University of the Aegean

Doctoral Thesis

Mathematical Modelling of Categorical Data with Actuarial and Financial Applications

---

Author:
Christos Meselidis

Supervisor:
Prof. Alex Karagrigoriou

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Department of Statistics and Actuarial—Financial Mathematics

Samos, 18 October 2022

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις, που προβλέπονται από τις διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι: Είμαι ο αποκλειστικός συγγραφέας της υποβληθείσας Διδακτορικής Διατριβής με τίτλο «Mathematical Modelling of Categorical Data with Actuarial and Financial Applications». Η συγκεκριμένη Διδακτορική Διατριβή είναι πρωτότυπη και εκπονήθηκε αποκλειστικά για την απόκτηση του Διδακτορικού διπλώματος του Τμήματος. Κάθε βοήθεια, την οποία είχα για την προετοιμασία της, αναγνωρίζεται πλήρως και αναφέρεται επακριβώς στην εργασία. Επίσης, επακριβώς αναφέρω στην εργασία τις πηγές, τις οποίες χρησιμοποίησα, και μνημονεύω επώνυμα τα δεδομένα ή τις ιδέες που αποτελούν προϊόν πνευματικής ιδιοκτησίας άλλων, ακόμη κι εάν η συμπερίληψή τους στην παρούσα εργασία υπήρξε έμμεση ή παραφρασμένη. Γενικότερα, βεβαιώνω ότι κατά την εκπόνηση της Διδακτορικής Διατριβής έχω τηρήσει απαρέγκλιτα όσα ο νόμος ορίζει περί διανοητικής ιδιοκτησίας και έχω συμμορφωθεί πλήρως με τα προβλεπόμενα στο νόμο περί προστασίας προσωπικών δεδομένων και τις αρχές Ακαδημαϊκής Δεοντολογίας.

University of the Aegean

Doctoral Thesis

Mathematical Modelling of Categorical Data with Actuarial and Financial Applications

---

Author:
Christos Meselidis

Supervisor:
Prof. Alex Karagrigoriou

Seven-member Evaluation Committee

Batsidis Apostolos
Associate Professor, University of Ioannina

Halidias Nikolaos
Professor, University of the Aegean

Karagrigoriou Alexandros (*Supervisor*)
Professor, University of the Aegean

Nastou Panagiotis
Assistant Professor, University of the Aegean

Psarrakos Georgios
Associate Professor, University of Piraeus

Tachtsis Eleftherios (*Three-member Evaluation Committee*)
Professor, University of the Aegean

Tzavelas Georgios (*Three-member Evaluation Committee*)
Associate Professor, University of Piraeus

Dedicated to my mother.

"If . . .
If you can keep your head when all about you
Are losing theirs and blaming it on you,
If you can trust yourself when all men doubt you,
But make allowance for their doubting too;

If you can wait and not be tired by waiting,
Or being lied about, don't deal in lies,
Or being hated, don't give way to hating,
And yet don't look too good, nor talk too wise:

If you can dream – and not make dreams your master;
If you can think – and not make thoughts your aim;
If you can meet with Triumph and Disaster
And treat those two impostors just the same;

If you can bear to hear the truth you've spoken
Twisted by knaves to make a trap for fools,
Or watch the things you gave your life to broken,
And stoop and build 'em up with wornout tools:

If you can make one heap of all your winnings
And risk it on one turn of pitch-and-toss,
And lose, and start again at your beginnings
And never breathe a word about your loss;

If you can force your heart and nerve and sinew
To serve your turn long after they are gone,
And so hold on when there is nothing in you
Except the Will which says to them: 'Hold on!'

If you can talk with crowds and keep your virtue,
Or walk with kings – nor lose the common touch,
If neither foes nor loving friends can hurt you,
If all men count with you, but none too much;

If you can fill the unforgiving minute
With sixty seconds' worth of distance run –
Yours is the Earth and everything that's in it,
And – which is more – you'll be a Man my son!."

Rudyard Kipling

Acknowledgements

First and foremost I would like to pay my deepest gratitude to professor Alexandros KARAGRIGORIOU, who is the supervisor of this thesis. His solid character and wide spirit, along with the fact that he was working tirelessly with profound enthusiasm while simultaneously being so approachable to his students, inspired me in every step of the process.

I would like also to express my appreciation to the members of the evaluation committee: Associate Professor Apostolos BATSIDIS, Professor Nikolaos HALIDIAS, Assistant Professor Panagiotis NASTOU, Associate Professor Georgios PSARRAKOS, Professor Eleftherios TACHTSIS and Associate Professor Georgios TZAVELAS. Their useful comments have improved significantly the overall quality of this thesis.

Especially, I would like to thank professor Apostolos BATSIDIS of the University of Ioannina whose comments, in reference to the comparative studies undertaken in our work, made during a conference, lead to the derivation of solid conclusions and improved the quality of this thesis. Furthermore, I would like to thank professor Aylin ALIN of Dokuz Eylul University, who directed our attention to penalized measures, leading to the writing of one of the chapters of this thesis.

I should also thank my fellow student Kimon NTOTSIS, where after a long discussion regarding the visualization of the results, my view in respect to that matter got widen.

Finally, I need to express my gratitude and unbounded love to my wife Christina POPORI, who stands by my side all those years, embracing the sacrifices that come along with such a decision.

Contents

List of Figures

List of Tables

List of Abbreviations

AIC	**A**kaike **I**nformation **C**riterion
ARE	**A**symptotic **R**elative **E**fficiency
BAN	**B**est **A**symptotically **N**ormal
BHHJ	**B**asu **H**arris **H**jort and **J**ones
CDF	**C**omulative **D**istribution **F**unction
CR	**C**ressie **R**ead
CS	**C**hi **S**quare
DIC	**D**ivergence **I**nformation **C**riterion
DITS	**D**ouble **I**ndex **T**est **S**tatistic
ECDF	**E**mpirical **C**omulative **D**istribution **F**unction
edf	**e**mpirical **d**istribution **f**unction
FT	**F**reeman **T**ukey
gof	**g**oodness$-$**o**f$-$**f**it
iff	if and only if
IFT	**I**mplicit **F**unction **T**heorem
IQR	**I**nterquartile **R**ange
LR	**L**ikelihood **R**atio
M(Φ,α)PDE	**M**inimum (Φ,α) **P**ower **D**ivergence **E**stimator
M(Φ,α)PDF	**M**odified (Φ,α) **P**ower **D**ivergence **F**amily
MCRPDE	**M**inimum **C**ressie **R**ead **P**ower **D**ivergence **E**stimator
MCS	**M**odified **C**hi **S**quare
MDI	**M**inimum **D**iscrimination **I**nformation
MDIC	**M**odified **D**ivergence **I**nformation **C**riterion
MDITS	**M**odified **D**ouble **I**ndexed **T**est **S**tatistic
MLE	**M**aximum **L**ikelihood **E**stimator
MM(Φ,α)PDE	**M**inimum **M**odified (Φ,α) **P**ower **D**ivergence **E**stimator
MPφDE	**M**inimum **P**enalized φ **D**ivergence **E**stimator
MSE	**M**ean **S**quared **E**rror
MSNC	**M**odified **S**kew **N**ormal **C**auchy
MφDE	**M**inimum φ **D**ivergence **E**stimator
ORMSE	**O**verall **R**oot **M**ean **S**quared **E**rror
PIC	**P**seudodistance **I**nformation **C**riterion
PφDTS	**P**enalized φ **D**ivergence **T**est **S**tatistic
r.v.	**r**andom **v**ariable
rMD	**r**estricted **M**inimum **D**ivergence
rMDE	**r**estricted **M**inimum **D**ivergence **E**stimator
RMSE	**R**oot **M**ean **S**quared **E**rror
ROC	**R**eceiver **O**perating **C**haracteristic
φDTS	φ **D**ivergence **T**est **S**tatistic

UNIVERSITY OF THE AEGEAN

Abstract

Faculty of Science

Department of Statistics and Actuarial−Financial Mathematics

Doctor of Philosophy

Mathematical Modelling of Categorical Data with Actuarial and Financial Applications

by Christos MESELIDIS

The main focus of this thesis is on model−based inferential statistics for contingency tables. The leading role in the search for an optimal modelling approach is played by divergence measures. Divergence measures can be used in statistical inference for estimating purposes, in the construction of test statistics for test−of−fit or in statistical modeling for the construction of model selection criteria. At the same time, in most scientific fields, the interest lies in statistical analysis of multivariate categorical data. A standard way for testing purposes is via the traditional Pearson's chi−squared goodness−of−fit test statistic or by the likelihood ratio test statistic, where the unknown parameters are estimated through the maximum likelihood method. However, this procedure can lead to poor results in many circumstances and it is possible to acquire better results by considering general families of test statistics, as well as of general families of estimators. Under the set−up of contaminated data, regarding the estimation procedure, these general families of divergence measures impose a compromise between robustness and efficiency. Furthermore, in the situation where zero frequency cells occur, a modified version of these general families can lead to robust estimators that are more efficient than the classical ones. On the other hand, under the contaminated data regime, such general families can produce more stable test statistics regarding the size and the power of the test. In this thesis, we study and exploit the general $(\Phi, \alpha)-$power divergence family and prominent sub−families of it, like the BHHJ family of measures. Rigorous asymptotic results, regarding the estimators and the associated test statistics, are provided while their applicability is presented through extensive simulation studies. Note that, regarding the aforementioned methodology, there is a straightforward connection with the actuarial science, mainly associated with the collective risk model of general insurance businesses and the modelling of frequency and severity of claims, and in general with financial mathematics.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Περίληψη

Σχολή Θετικών Επιστημών
Τμήμα Στατιστικής και Αναλογιστικών–Χρηματοοικονομικών Μαθηματικών

Διδακτορικό

Μαθηματική Μοντελοποίηση Κατηγορικών Δεδομένων με Εφαρμογές στα Αναλογιστικά και Χρηματοοικονομικά

του Χρήστου ΜΕΣΕΛΙΔΗ

Το κύριο ενδιαφέρον αυτής της διατριβής εστιάζεται στη στατιστική συμπερασματολογία μοντέλων που αφορούν πίνακες συνάφειας. Σημαντικό ρόλο στην αναζήτηση βέλτιστου μοντέλου παίζουν τα μέτρα απόκλισης (divergence). Τα divergence μέτρα μπορούν να χρησιμοποιηθούν για σκοπούς εκτίμησης παραμέτρων, για την κατασκευή στατιστικών ελέγχων καλής προσαρμογής ή στη στατιστική μοντελοποίηση για την κατασκευή κριτηρίων επιλογής μοντέλου. Την ίδια στιγμή, στα περισσότερα επιστημονικά πεδία, το ενδιαφέρον εστιάζεται στη στατιστική ανάλυση κατηγορικών δεδομένων. Ο συνήθης τρόπος εκτίμησης του μοντέλου γίνεται με τον παραδοσιακό Pearson chi–squared έλεγχο καλής προσαρμογής ή με τον likelihood ratio έλεγχο καλής προσαρμογής και οι άγνωστες παράμετροι εκτιμώνται με την μέθοδο της μέγιστης πιθανοφάνειας. Ωστόσο σε αρκετές περιπτώσεις αυτό μπορεί να μην οδηγήσει στα βέλτιστα αποτελέσματα, και είναι δυνατό να έχουμε καλύτερα αποτελέσματα θεωρώντας πιο γενικές οικογένειες στατιστικών ελέγχων, καθώς επίσης και πιο γενικές οικογένειες εκτιμητών. Υπό το πρίσμα νοθευμένων (contaminated) δεδομένων, αυτές οι γενικές οικογένειες divergence μέτρων οδηγούν σε "συμβιβασμό" μεταξύ της ανθεκτικότητας (robustness) και της αποδοτικότητας (efficiency) των εκτιμητών. Επιπλέον, στην περίπτωση εμφάνισης κελιών μηδενικής συχνότητας, μία τροποποιημένη (modified) εκδοχή αυτών των γενικών οικογενειών μπορεί να οδηγήσει σε ανθεκτικούς (robust) εκτιμητές οι οποίοι είναι πιο αποδοτικοί (efficient) από ότι οι κλασικοί εκτιμητές. Από την άλλη μεριά, στην περίπτωση των contaminated δεδομένων, τέτοιες γενικές οικογένειες δύνανται να παράγουν πιο ευσταθείς (stable) στατιστικούς ελέγχους, όσον αφορά το σφάλμα τύπου I (ή μέγεθος) (size) και την ισχύ (power) του ελέγχου. Σε αυτή τη διατριβή, μελετάμε και εκμεταλλευόμαστε τη γενική (Φ, α)–power divergence οικογένεια μέτρων απόκλισης και κάποιες προεξέχουσες υπό-οικογένειες αυτής, όπως τη BHHJ οικογένεια μέτρων. Αναφορικά με τους εκτιμητές και τους παραγόμενους στατιστικούς ελέγχους παρέχονται αυστηρά ασυμπτωτικά αποτελέσματα, ενώ η πρακτική εφαρμογή αυτών παρουσιάζεται μέσα από εκτενείς μελέτες προσομοίωσης. Σημειώνεται ότι η ανωτέρω μεθοδολογία, είναι άμεσα συνδεδεμένη με την αναλογιστική επιστήμη και μεταξύ άλλων σχετίζεται με το συλλογικό μοντέλο κινδύνου που απαντάται στις γενικές ασφαλίσεις και τη μοντελοποίηση της συχνότητας (frequency) και της σφοδρότητας (severity) των απαιτήσεων (claims) και γενικά στα οικονομικά μαθηματικά.

Συγκεκριμένα, το κεφάλαιο 2 πραγματεύεται ζητήματα εκτίμησης παραμέτρων και ελέγχων στατιστικών υποθέσεων, σε πολυωνυμικούς πληθυσμούς. Οι εκτιμητές, καθώς και οι αντίστοιχες στατιστικές συναρτήσεις που προτείνονται, βασίζονται στη γενική (Φ, α) οικογένεια divergence μέτρων. Έτσι, για την εκτίμηση των παραμέτρων, ορίζεται η οικογένεια των minimum (Φ, α)–power divergence εκτιμητών, ενώ δίνονται αυστηρά

θεωρητικά αποτελέσματα σχετικά με το ανάπτυγμα του εκτιμητή γύρω από την πραγματική τιμή της άγνωστης παραμέτρου (decomposition) και την ασυμπτωτική κατανομή (asymptotic distribution) αυτού. Στη συνέχεια, για τον έλεγχο σύνθετων μηδενικών στατιστικών υποθέσεων, προτείνεται η γενική οικογένεια double index $(\Phi, \alpha)-$power divergence στατιστικών η οποία εξαρτάται από δύο δείκτες (indices) οι τιμές των οποίων διαδραματίζουν καθοριστικό ρόλο στην αποτελεσματικότητα και την ακρίβεια της προτεινόμενης μεθοδολογίας. Παράλληλα, δίνονται αυστηρές αποδείξεις αναφορικά με την ασυμπτωτική κατανομή αυτής της οικογένειας στατιστικών, τόσο κάτω από τη μηδενική υπόθεση όσο και κάτω από εναλλακτικές υποθέσεις οι οποίες τείνουν προς τη μηδενική καθώς αυξάνει το μέγεθος του δείγματος (γνωστές ως contiguous alternatives) καθώς και την ισχύ (power) του ελέγχου. Το κεφάλαιο 2 ολοκληρώνεται με τρείς εκτενείς μελέτες προσομοίωσης. Η πρώτη εξ αυτών, στοχεύει στην επαλήθευση των θεωρητικών αποτελεσμάτων αναφορικά με την ασυμπτωτική κατανομή του προτεινόμενου στατιστικού και το κατά πόσο αυτή περιγράφει ορθώς την εμπειρική κατανομή του στατιστικού σε πεπερασμένα (μεγάλα) δείγματα. Επιπλέον, αναδεικνύεται η επίδραση που έχουν οι δείκτες, από τους οποίους εξαρτάται το στατιστικό, πάνω στην εμπειρική κατανομή αυτού. Η δεύτερη μελέτη εστιάζει στη σύγκριση, ως προς το σφάλμα τύπου I (size) και την ισχύ (power) του ελέγχου, μεταξύ μελών της προτεινόμενης οικογένειας στατιστικών και στατιστικών που είθισται να χρησιμοποιούνται κατά κόρον και τα οποία αναφέρονται στην παρούσα διατριβή ως "κλασικά". Η τρίτη μελέτη αφορά τη σύγκριση εκτιμητών, ως προς την αποδοτικότητα (efficiency) και την ανθεκτικότητα (robustness), καθώς και στατιστικών, ως προς το σφάλμα τύπου I (size) και την ισχύ (power), υπό το πρίσμα νοθευμένων κατανομών (contaminated distributions).

Το κεφάλαιο 3 εστιάζεται σε πίνακες συνάφειας (contingency tables) με έμφαση στη δεσμευμένη ανεξαρτησία (conditional independence) των μεταβλητών. Ορίζεται η οικογένεια των restricted minimum $(\Phi, \alpha)-$power divergence εκτιμητών καθώς και η παραγόμενη οικογένεια των double index $(\Phi, \alpha)-$power divergence στατιστικών, ενώ παράλληλα δίνονται θεωρητικά αποτελέσματα ανάλογα με αυτά του κεφαλαίου 2. Η μελέτη προσομοίωσης αφορά την περίπτωση της δεσμευμένης ανεξαρτησίας σε $2 \times 2 \times 2$ πίνακες συνάφειας και υποδεικνύει ότι μέλη της προτεινόμενης οικογένειας στατιστικών συμπεριφέρονται, ως προς το σφάλμα τύπου I (size) και την ισχύ (power) του ελέγχου, τόσο καλά ή ακόμη καλύτερα σε σχέση με τα "κλασικά" στατιστικά.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η περίπτωση όπου κατηγορίες (ή κελιά) του πολυωνυμικού μοντέλου εμφανίζουν μηδενικές παρατηρούμενες συχνότητες. Σε τέτοιες περιπτώσεις έχει παρατηρηθεί ότι τροποποιημένες εκδοχές divergence μέτρων, στις οποίες το βάρος που δίνεται στις κατηγορίες που εμφανίζουν μηδενικές παρατηρούμενες συχνότητες είναι πλέον μία μεταβλητή παράμετρος και όχι σταθερά, οδηγούν σε ανθεκτικούς (robust) εκτιμητές που είναι πιο αποδοτικοί (efficient) καθώς επίσης και σε πιο ευσταθείς (stable) στατιστικές συναρτήσεις. Έτσι, στο κεφάλαιο 4, ορίζεται για πρώτη φορά η modified $(\Phi, \alpha)-$power divergence οικογένεια μέτρων, η modified minimum $(\Phi, \alpha)-$power divergence οικογένεια εκτιμητών καθώς και η modified double index οικογένεια στατιστικών συναρτήσεων. Δίνονται θεωρητικά αποτελέσματα αναφορικά με το ανάπτυγμα του εκτιμητή γύρω από την τιμή της παραμέτρου που ελαχιστοποιεί την απόσταση μεταξύ των παρατηρούμενων σχετικών συχνοτήτων και του παραμετρικού μοντέλου που έχει θεωρηθεί (decomposition) καθώς και την ασυμπτωτική κατανομή (asymptotic distribution) αυτού, κάτω από μη ορθώς καθορισμένα (misspecified) μοντέλα. Επιπλέον, δίνεται η ασυμπτωτική κατανομή της προτεινόμενης οικογένειας των τροποποιημένων στατιστικών, όταν τα υποτιθέμενο μοντέλο έχει ορθώς καθοριστεί. Το κεφάλαιο 4 ολοκληρώνεται με μία μελέτη προσομοίωσης, όπου εξετάζεται η αποδοτικότητα (efficiency) και η ανθεκτικότητα (robustness), μελών της προτεινόμενης οικογένειας εκτιμητών καθώς παρουσιάζεται και ο ρόλος που διαδραματίζει η παράμετρος που σχετίζεται με τις κατηγορίες που εμφανίζουν μηδενικές παρατηρούμενες συχνότητες. Επιπλέον, εξετάζεται η συμπεριφορά μελών της προτεινόμενης οικογένειας στατιστικών ως προς στο σφάλμα τύπου I (size) τόσο χωρίς όσο και στην παρουσία νοθευμένων δεδομένων (contaminated data) της κατανομής. Η μελέτη υποδεικνύει ότι, μέλη της προτεινόμενης οικογένειας στατιστικών είναι πιο ευσταθή (stable) υπό την παρουσία δεδομένων νόθευσης της κατανομής.

Το κεφάλαιο 5, αποσκοπεί στην ανάδειξη της πρακτικής εφαρμογής της μεθοδολογίας που αναπτύχθηκε στα προηγούμενα κεφάλαια. Συγκεκριμένα, στρέφει το ενδιαφέρον στο συλλογικό μοντέλο κινδύνου, το οποίο χρησιμοποιείται κατά κόρον στις γενικές ασφαλίσεις, κατά τη διαδικασία εκτίμησης του ύψους των συνολικών απαιτήσεων. Έτσι, η χρησιμότητα των μεθόδων που αναπτύχθηκαν, παρουσιάζεται μέσω ενός παραδείγματος το οποίο εστιάζει στη μοντελοποίηση της συχνότητας και της σφοδρότητας των απαιτήσεων.

Το κεφάλαιο 6, παρουσιάζει ανοιχτά προβλήματα που χρήζουν περαιτέρω διερεύνησης και αποτελούν θέματα μελλοντικής έρευνας.

Τέλος, στο παράρτημα παρουσιάζεται ο κώδικας των μελετών προσομοίωσης.

Chapter 1

Introduction

The concept of distance or divergence is known since at least the time of Pearson, who, in 1900, considered the classical goodness$-$of$-$fit (gof) problem by considering the distance between observed and expected frequencies. The problem of statistical inference based on measures of divergence, for both discrete and discretized continuous distributions, have been in the center of attention for the last 100+ years. The classical set$-$up is the one considered by Pearson, where a hypothesized $m-$dimensional multinomial distribution, say $Multi(N, p_1, \ldots, p_m)$ is examined as being the underlying distributional mechanism for producing a given sample of size N. The problem can be extended to examine the homogeneity (in terms of the distributional mechanisms) among two independent samples or the independence among two population characteristics. In all such problems we are dealing with cross tabulations or crosstabs (or contingency tables). Problems of such nature appear, frequently, in a great variety of fields including biosciences, socio$-$economic and political sciences, actuarial science, finance, business, accounting and marketing. The need to establish, for instance, whether the mechanisms producing two phenomena are the same or not is vital for altering economic policies, preventing socio$-$economic crises or enforcing the same economic or financial decisions to groups with similar underlying mechanisms (e.g., retaining the insurance premium in case of similarity or having different premiums in case of diversity). The intense engagement of many authors with the measures of divergence, demonstrates the significant role that they play in statistical inference. To this regard, we provide references to the fields where divergence measures have been applied as well as to their uses in statistical inference. Divergence measures can be applied in various fields such as

- signal processing and pattern recognition (Bekara et al. 2006),
- speech recognition (Qiao & Minematsu, 2010),
- clustering (Fischer, 2010),
- bioscience (Dubrova et al., 2002; Znaor et al., 2003),
- finance and economics (Toma & Leoni-Aubin, 2014; Merková, 2015),
- insurance science (Geenens & Simar, 2010),
- socioeconomics (Bartolucci & Scaccia, 2004),
- analysis of contingency tables (Alin & Kurt, 2008; Kateri, 2018),
- multivariate data analysis (Vassiliou et al., 2004; Koutras et al., 2008),

- model selection (Akaike, 1973; Cavanaugh, 2004; Shang & Cavanaugh, 2008; Mattheou et al., 2009; Mantalos et al., 2010; Toma, 2014),
- parameter estimation (Alin & Kurt, 2008; Toma, 2008, 2009; Jiménez-Gamero et al., 2011; Toma & Leoni-Aubin, 2014; Neath et al., 2015; Jiménez-Gamero & Batsidis, 2017),
- goodness−of−fit tests (Zografos et al., 1990; Toma & Leoni-Aubin, 2010; Stehlík et al., 2014; Batsidis et al., 2014, 2016a; Stehlík et al., 2018), etc.

In this thesis, emphasis will be given to actuarial and financial applications and in particular to the collective risk model and the modelling of frequency and severity of claims and the evaluation of the expected cost of total losses.

It is important to note that divergence measures play a pivotal role also in statistical inference in continuous settings. Indeed, for example, in Salicru et al. (1994) the authors investigate the multivariate normal case while in a recent work (Contreras-Reyes et al., 2021), the modified skew−normal−Cauchy (MSNC) distribution is considered, against normality.

Measures of information are powerful statistical tools directly related to statistical inference, including robustness, with diverse applicability (see e.g. Papaioannou, 2004; Basu et al., 2011; Ghosh et al., 2013). Indeed, on one hand they can be used for estimation purposes with classical example the well−known maximum likelihood estimator which is the result of the implementation of the famous Kullback−Leibler measure. On the other hand, measures are applicable in tests of fit to quantify the degree of agreement between the distribution of an observed random sample and a theoretical, hypothesized distribution. The problem of goodness−of−fit (gof) to any distribution, is frequently treated by partitioning the range of data in a number of disjoint intervals. In all cases, a test statistic is compared against a known critical value to accept or reject the hypothesis that the sample is from the postulated distribution. Over the years, numerous nonparametric gof methods including the chi−squared test and various empirical distribution function (edf) tests (see the monograph by D'Agostino & Stephens (1986) and references therein), have been developed. At the same time measures of entropy, divergence and information are quite popular in goodness−of−fit tests. Several measures have been suggested to reflect the fact that some probability distributions are closer together than others. Many of the currently used tests, such as the likelihood ratio, the chi−squared, the score and Wald tests are defined in terms of appropriate measures.

Although, in special cases, physical data generating mechanisms suggest a good probability model, in general, we do not have sufficient knowledge of the machinery generating the data to convert it into a probability distribution. A way to resolve the issue of model identification is to estimate the underlying mechanism by approaches based on information or divergence measures, which are used for measuring the closeness or the distance between each candidate model and the true but unknown model.

In this chapter the focus is placed on measures of divergence and distance providing an overview of their properties and their uses in statistical inference. The most known measures are briefly presented in Section 1.1, while some of their properties are provided in Section 1.2. Model selection criteria and how they are derived via divergence measures are discussed in Section 1.3, while in Section 1.4 we provide the basic theoretical results about goodness−of−fit tests for multinomial distributions, which tests are based on divergence measures. Section 1.5 concludes with the scope of this thesis.

1.1 Divergence measures

Let f and g be two probability density functions which may depend on an unknown parameter of fixed finite dimension. For historical reasons we present first Shannon's Entropy (Shannon, 1948) given by

$$I^S(X) \equiv I^S(f) = -\int f(z) \log f(z) dz = E_f[-\log f(X)],$$

where X is a random variable with density function $f(x)$. The development of the concept of entropy started in the 19th century in the field of physics and in particular in describing thermodynamics processes, but the development of the statistical description of entropy by Boltzmann led to a strong resistance by many. Shannon's entropy was introduced and used during World War II in Communication Engineering. Shannon derived the discrete version of $I^S(f)$, where f is a probability mass function and named it *entropy* because of its similarity with thermodynamics entropy. The continuous version was defined by analogy and it is called *differential entropy* (see Cover & Thomas (2006) for further details). For a finite number of points, Shannon's entropy measures the expected information of a signal provided without noise from a source X with density $f(x)$ and is related to Kullback−Leibler divergence (Kullback & Leibler, 1951) through the following expression:

$$I^S(f) = I^S(h) - I_X^{KL}(f,h)$$

where h is the density of the uniform distribution while I_X^{KL} is the Kullback−Leibler divergence between two densities f and g which is given in the sequel (see relation (1.2)).

Many generalizations of Shannon's Entropy were hereupon introduced. Rényi (1961) entropy as extended by Liese & Vajda (1987) is given by

$$I^{R_{lv},a}(X) \equiv I^{R_{lv},a}(f) = \frac{1}{a(a-1)} \ln E_f\left(f(X)^{a-1}\right), \ a \neq 0, 1.$$

For more details about entropy measures the reader is referred to Mathai & Rathie (1975) and Nadarajah & Zografos (2003).

1.1.1 $\varphi-$divergences

The $\varphi-$divergence family of measures (Csiszár, 1963; Ali & Silvey, 1966) is defined by

$$I_X^{\varphi}(f,g) = \int g(z) \varphi\left(\frac{f(z)}{g(z)}\right) dz \tag{1.1}$$

where φ is a convex function in $[0,\infty)$ such that $0\varphi(u/0) = \lim_{x\to\infty}[\varphi(x)/x]$ and $0\varphi(0/0) = 0$. Many of the well−known measures of divergence can be considered as particular cases of the $\varphi-$divergence measure. For instance, Pearson's chi−squared divergence (also known as Kagan's divergence; Kagan (1963)) which is given by

$$I_X^{Ka}(f,g) = \frac{1}{2}\int g(z)\left(1-\frac{f(z)}{g(z)}\right)^2 dz$$

can be derived from (1.1) for $\varphi(u) \doteq \varphi_{Ka}(u) = (1-u)^2/2$. Hellinger distance is another important divergence measure that can be derived from (1.1) for $\varphi(u) = 2(\sqrt{u}-1)$ and is given by

$$I_X^{He}(f,g) = \left(\int (\sqrt{f(z)}-\sqrt{g(z)})^2 dz\right)^{1/2}.$$

Note that, Hellinger distance is the square root of Matusita's divergence (Matusita, 1967). For $\varphi(u) \doteq \varphi_{KL}(u) = u\log u + 1 - u$, the $\varphi-$divergence measure yields the Kullback−Leibler divergence (Kullback & Leibler, 1951) given by

$$I_X^{KL}(f,g) = \int f(z)\log\left(\frac{f(z)}{g(z)}\right) dz \tag{1.2}$$

and for $\varphi(u) \doteq \varphi_{CR}(u) = (u^{\lambda+1} - u - \lambda(u-1))/(\lambda(\lambda+1))$, $\lambda \neq 0, -1$ yields the family of power divergences introduced by Cressie & Read (1984), which is given by

$$I_X^{CR}(f,g) = \frac{1}{\lambda(\lambda+1)}\int f(z)\left[\left(\frac{f(z)}{g(z)}\right)^{\lambda} - 1\right] dz, \quad \lambda \in \mathbb{R} \tag{1.3}$$

where for $\lambda = 0, -1$ is defined by continuity. Note that for $\lambda \to 0$ the Kullback−Leibler divergence is obtained, while for $\lambda = 1$, (1.3) reduces to the Pearson's chi−squared divergence.

1.1.2 $\alpha-$divergences

Another class of divergence measures is the class of $\alpha-$divergences. The additive and nonadditive directed divergences of order α were introduced in the 1960s and the 1970s (Rényi, 1961; Csiszár, 1963; Rathie & Kannappan, 1972). The so−called order α information measure of Rényi (1961) is given by

$$I_X^R(f,g) = \frac{1}{\alpha-1}\log\left(\int f^{\alpha}(z)g^{1-\alpha}(z)\ dz\right), \quad \alpha > 0, \ \alpha \neq 1.$$

It should be noted that for $\alpha \to 1$ the above measure becomes the Kullback−Leibler divergence.

One of the most recently proposed measures is the family of pseudodistances of order $\alpha > 0$ (see Jones et al., 2001; Broniatowski et al., 2012) which is defined by

$$\begin{aligned} I_X^P(f,g) = {} & \frac{1}{\alpha+1}\log\left(\int g^{1+\alpha}(z)dz\right) + \frac{1}{\alpha(\alpha+1)}\log\left(\int f^{1+\alpha}(z)dz\right) \\ & - \frac{1}{\alpha}\log\left(\int f(z)g^{\alpha}(z)dz\right) \end{aligned} \tag{1.4}$$

where for α tending to 0 reduces to the Kullback−Leibler divergence $I_X^{KL}(f,g)$.

The BHHJ (Basu, Harris, Hjort and Jones) power divergence (Basu et al., 1998) between f and g is another measure indexed by a positive parameter α and defined as

$$I_X^{BHHJ}(f,g) \\ = \int \left\{ g^{1+\alpha}(z) - \left(1+\frac{1}{\alpha}\right) f(z) g^{\alpha}(z) + \frac{1}{\alpha} f^{1+\alpha}(z) \right\} dz, \quad \alpha > 0. \tag{1.5}$$

Note that for $\alpha \to 0$ the BHHJ measure reduces to the Kullback−Leibler divergence and for $\alpha = 1$ to the square of the L_2 distance between f and g. Mattheou et al. (2009) proposed a general class of divergence measures, the $(\Phi, \alpha)-$power divergence family given by

$$I_\Phi^\alpha(f,g) = \begin{cases} \displaystyle\int g^{1+\alpha}(z) \Phi\left(\frac{f(z)}{g(z)}\right) dz; & \alpha > 0 \\ \displaystyle\int g(z) \Phi\left(\frac{f(z)}{g(z)}\right) dz; & \alpha = 0 \end{cases} \tag{1.6}$$

where Φ is a convex function such that $0\Phi(u/0) = \lim_{x\to\infty}[\Phi(x)/x]$ and $0\Phi(0/0) = 0$. For $\Phi(u) = 1 - (1+\alpha^{-1})u + \alpha^{-1}u^{1+\alpha}$ (1.6) reduces to the BHHJ power divergence. The $(\Phi, \alpha)-$measure is a broad class of measures that covers also the Cressie and Read power divergence family and the $\varphi-$divergence family. Indeed, for $\alpha = 0$ if we take $\Phi = \varphi$ then (1.6) coincides with (1.1), while for $\Phi = \varphi_3$ the $(\Phi, \alpha)-$measure reduces to (1.3).

1.1.3 Bregman divergences

Finally, the Bregman divergence (Bregman, 1967) between the probability density functions f and g is given by

$$I_X^{Breg}(f,g) = \int \Big(\phi(f(z)) - \phi(g(z)) - \phi'(g(z))(f(z) - g(z))\Big) dz$$

for any differentiable convex function ϕ with $\phi(0) = \lim_{u\to 0}\phi(u) \in (-\infty, \infty)$. Observe that for $\phi(u) = u \log u$ the Bregman divergence reduces to the Kullback−Leibler divergence and for $\phi(u) = \alpha^{-1}(u^{\alpha+1} - u)$, $\alpha > 0$ reduces to the BHHJ power divergence.

1.1.4 Discrete divergences

The above divergence measures can be defined also for discrete settings. Let $\mathbf{p} = (p_1, \ldots, p_m)^\top$ and $\mathbf{q} = (q_1, \ldots, q_m)^\top$ be two discrete finite probability distributions, where the superscript $\top$ denotes the transpose of a vector or a matrix, while the subscript m denotes the dimension of the probability vector of the associated multinomial model and is related to the number of categories. Then the discrete version of the $\varphi-$divergence measure is given by

$$d_\varphi(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^m q_i \varphi\left(\frac{p_i}{q_i}\right)$$

while the discrete version of the $(\Phi, \alpha)-$power divergence measure is given by

$$d_{\Phi}^{\alpha}(\mathbf{p},\mathbf{q}) = \begin{cases} \sum_{i=1}^{m} q_i^{1+\alpha}\Phi\left(\frac{p_i}{q_i}\right); & \alpha > 0 \\ \sum_{i=1}^{m} q_i\Phi\left(\frac{p_i}{q_i}\right); & \alpha = 0. \end{cases} \tag{1.7}$$

Other measures of divergence can be defined in a similar fashion. Examples of ϕ functions are given in Arndt (2001) and Pardo (2006). For more details on divergence measures see Cavanaugh (2004), Toma (2009), and Toma & Broniatowski (2011). Specifically for robust inference based on divergence measures see Basu et al. (2011) and a paper by Patra et al. (2013) on the power divergence and the density power divergence families. The descritized version of measures has been given considerable attention over the years with some representative works being by Zografos et al. (1986) and Papaioannou et al. (1994).

1.2 Properties of divergence measures

The measures of divergence are not formal distance functions. Any bivariate function $I_X(\cdot,\cdot)$ that satisfies the non−negativity property, namely $I_X(\cdot,\cdot) \geq 0$ with equality if and only if its two arguments are equal can be possibly used as a measure of divergence. Note that, formal distance functions, i.e metrics, are additionally symmetric and satisfy the triangle inequality. Several properties of divergence measures have been investigated over the years and some of them are presented in the sequel.

$\varphi-$divergences are invariant with respect to common changes of variables. Also, generally, the divergences are not symmetric. Indeed, $I^{\varphi}(f,g)$ and $I^{\varphi}(g,f)$ are in general, different. They are equal when the convex function φ verifies the condition $\varphi(x) - x\varphi(1/x) = c(x-1)$, where c is a constant (Liese & Vajda, 1987). Also, $I_X^{\varphi}(f,g) = 0$ if and only if the two arguments are equal, provided that the function $x \to \varphi(x)$ is strictly convex in a neighborhood of $x = 1$. Otherwise, only "$f = g$ implies $I_X^{\varphi}(f,g) = 0$" occurs.

The order−preserving property has been introduced by Shiva et al. (1973) and shows that the relation between the amount of information contained in a r.v. (random variable) X_1 and that contained in another r.v. X_2 remains intact irrespective of the measure of divergence used. In particular, if the superscripts 1 and 2 represent two different measures of divergence then

$$I_{X_1}^1(f_1,g_1) \leq I_{X_2}^1(f_2,g_2) \Leftrightarrow I_{X_1}^2(f_1,g_1) \leq I_{X_2}^2(f_2,g_2).$$

The limiting property which refers to the $\mu-$almost everywhere convergence states that a sequence of probability density functions f_N converges to a probability density function f if and only if the corresponding measure of divergence tends to zero, i.e.

$$f_N \to f \text{ iff } I_X(f_N,f) \to 0.$$

Note that the $\mu-$almost everywhere convergence is a weakened version of pointwise convergence according to which, for a measure space X, a sequence $g_N(x)$ tends to $g(x)$ for all $x \in Y$, a measurable subset of X such that, $\mu(X|Y) = 0$ (see Vonta et al., 2012).

Another property is the Ali−Silvey property, where if $f(x, \theta)$ (or simply f_θ) has the monotone likelihood ratio property in x then

$$\theta_1 < \theta_2 < \theta_3 \Rightarrow I_X(f_{\theta_1}, f_{\theta_2}) < I_X(f_{\theta_1}, f_{\theta_3}).$$

Note that a family of density functions f_θ has the monotone likelihood ratio property in x if for any $\theta_1 < \theta_2$, $f_{\theta_2}/f_{\theta_1}$ is a non−decreasing function of x.

Recently Vonta et al. (2012) proved the limiting property and the order preserving property for the $(\Phi, \alpha)-$power divergence family of measures (1.6). As they state, the order preserving property holds for (1.6) provided that the parameter α belongs to a specific interval, which is different for each measure. They also showed the quadratic convergence of the $(\Phi, \alpha)-$measure. In other words they showed that the discretized version of the $(\Phi, \alpha)-$measure converges quadratically to the theoretical $(\Phi, \alpha)-$measure.

1.3 Model selection criteria

A model selection criterion can be considered as an approximately unbiased estimator of the expected overall discrepancy, a nonnegative quantity that measures the distance or divergence between the true unknown model and a fitted approximating model. If the value of the criterion is small then the approximated model is good.

Let $\mathbf{Y} = (Y_1, \ldots, Y_N)^\top$ be a random vector from the distribution g (the true model) and f_θ a candidate model from a parametric family of models $\{f_\theta\}$, indexed by an unknown $p-$dimensional parameter $\theta \in \Theta \subseteq \mathbb{R}^p$. Akaike (1973), proposed the Akaike Information Criterion (AIC) which is based on the Kullback−Leibler measure of divergence between the true and the candidate models. For a realization $\mathbf{y} = (y_1, \ldots, y_N)^\top$ of $\mathbf{Y} = (Y_1, \ldots, Y_N)^\top$ the AIC is given by

$$AIC = -2l(\hat{\theta}_{MLE}; \mathbf{y}) + 2p,$$

where $l(\theta; \mathbf{y})$ is the log−likelihood and $\hat{\theta}_{MLE}$ is the maximum likelihood estimator (MLE) of the unknown parameter, i.e.

$$l(\hat{\theta}_{MLE}; \mathbf{y}) = \sum_{i=1}^{n} \log(f_{\hat{\theta}_{MLE}}(y_i)) = \max_{\theta \in \Theta} l(\theta; \mathbf{y}).$$

Applying the same methodology used for AIC, Mattheou et al. (2009) developed the Divergence Information Criterion (DIC) which is based on the quantity

$$W_\theta = \int \left\{ f_\theta^{1+\alpha}(z) - (1 + \alpha^{-1}) g(z) f_\theta^\alpha(z) \right\} dz, \quad \alpha > 0. \tag{1.8}$$

Note that (1.8) is the same as the BHHJ power divergence $I_X^{BHHJ}(g, f_\theta)$, see relation (1.5), without the last term that remains constant irrespectively of the model f_θ used. The expected overall discrepancy, which is the target theoretical quantity for which Mattheou et al. (2009) provided an asymptotically unbiased estimator, is given by

$$E(W_{\hat{\theta}}) = E\Big(W_\theta | \theta = \hat{\theta} \Big)$$

where $\hat{\theta}$ is any consistent and asymptotically normal estimator of θ, under the assumption that f_θ is the true model. This quantity can be viewed as the average distance between g and f_θ up to a constant. For a set of observations $y_1, \dots, y_N$ the resulting criterion is given by

$$DIC = N\, Q_{\hat{\theta}} + (2\pi)^{-\alpha/2}(1+\alpha)^{2+p/2}p$$

where

$$Q_{\hat{\theta}} = \int f_{\hat{\theta}}^{1+\alpha}(z)dz - (1+\alpha^{-1})\frac{1}{N}\sum_{i=1}^{N} f_{\hat{\theta}}^{\alpha}(y_i).$$

Because of the fact that the integral $\int f_{\hat{\theta}}^{1+\alpha}(z)dz$ in the first part of $Q_{\hat{\theta}}$, for α relatively close to 0, is approximately constant for the various candidate models, Mantalos et al. (2010) proposed a modified version of DIC, the Modified Divergence Information Criterion (MDIC) which is given by

$$MDIC = N\, MQ_{\hat{\theta}} + (2\pi)^{-\alpha/2}(1+\alpha)^{2+p/2}p,$$

where

$$MQ_{\hat{\theta}} = -(1+\alpha^{-1})\frac{1}{N}\sum_{i=1}^{N} f_{\hat{\theta}}^{\alpha}(y_i)$$

and the expected overall discrepancy up to a constant, is given by

$$E(V_{\hat{\theta}}) = E\Big(V_\theta \big| \theta = \hat{\theta}\Big) \;\text{ with }\; V_\theta = -(1+\alpha^{-1})\int g(z) f_\theta^{\alpha}(z)dz.$$

Recently, Toma et al. (2020) proposed a new model selection criterion, the so−called Pseudodistance Information Criterion (PIC). For this new criterion the family of pseudodistances of order $\alpha > 0$ was considered. The quantity that has been used for the construction of the criterion is given by

$$M_\theta = \frac{1}{\alpha+1}\log\left(\int f_\theta^{1+\alpha}(z)dz\right) - \frac{1}{\alpha}\log\left(\int g(z) f_\theta^{\alpha}(z)dz\right).$$

Observe that M_θ is the same as the pseudodistance $I_X^P(g, f_\theta)$, given in (1.4), without the middle term which is constant irrespectively of the model f_θ used. The expected overall discrepancy is given by $E(M_{\hat{\theta}_N}) = E(M_\theta|\theta = \hat{\theta}_N)$, where $\hat{\theta}_N$ is the minimum pseudodistance estimator (Broniatowski et al., 2012) defined as

$$\hat{\theta}_N = \arg\inf_{\theta\in\Theta}\left\{\frac{1}{\alpha+1}\log\left(\int f_\theta^{1+\alpha}(z)dz\right) - \frac{1}{\alpha}\log\left(\frac{1}{N}\sum_{i=1}^{N} f_\theta^{\alpha}(y_i)\right)\right\}.$$

The resulting criterion, based on the index α and the dimension p of θ, for the univariate normal case (Toma et al., 2020) is given by

$$PIC = N\, Q^*_{\hat{\theta}_N} + \frac{(\alpha+1)^2}{(2\alpha+1)^{3/2}}p + \frac{1}{2\alpha}\left(1 - \frac{\alpha+1}{\sqrt{2\alpha+1}}\right),$$

where

$$Q^*_{\hat{\theta}_N} = \frac{1}{\alpha+1}\log\left(\int f_{\hat{\theta}_N}^{1+\alpha}(z)dz\right) - \frac{1}{\alpha}\log\left(\frac{1}{N}\sum_{i=1}^{N} f_{\hat{\theta}_N}^{\alpha}(y_i))\right).$$

1.4 Goodness−of−fit tests

1.4.1 Simple null hypothesis

Let $\mathbf{X} = (X_1, \ldots, X_m)^\top$ be a random vector from the multinomial distribution, $\mathbf{X} \sim M(N, \mathbf{p})$, with parameters N and $\mathbf{p} = (p_1, \ldots, p_m)^\top$. For a realization $\mathbf{x} = (x_1, \ldots, x_m)^\top$ of $\mathbf{X} = (X_1, \ldots, X_m)^\top$ we have that $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_m)^\top$ with $\hat{p}_i = x_i/N, i = 1, \ldots, m$ is the nonparametric maximum likelihood estimator of the true probabilities p_i, $i = 1, \ldots, m$ and $N = \sum_{i=1}^m x_i$. If we want to test the simple null hypothesis that the data $(x_1, \ldots, x_m)^\top$ come from a known multinomial distribution $M(N, \mathbf{p}_0)$, where $\mathbf{p}_0 = (p_{10}, \ldots, p_{m0})^\top$ is the vector of hypothesized probabilities, namely

$$H_0 : \mathbf{p} = \mathbf{p}_0 \tag{1.9}$$

the most commonly used test statistics are Pearson's or chi−squared test statistic, given by

$$X^2 = \sum_{i=1}^m \frac{(x_i - Np_{i0})^2}{Np_{i0}} \tag{1.10}$$

and the likelihood ratio test statistic given by

$$G^2 = 2\sum_{i=1}^m x_i \log\left(\frac{x_i}{Np_{i0}}\right). \tag{1.11}$$

In the domain of hypothesis testing, divergence measures have a pivotal role. Traditionally, for testing hypotheses, we use the chi−squared and the likelihood ratio tests mentioned above, but Kupperman (Kupperman, 1957, 1958) proposed a new ground−breaking way. He established a test statistic based on the Kullback−Leibler divergence which is asymptotically distributed as a chi−squared random variable. The connection between divergence measure theory and hypothesis testing, continued with several alternative approaches. Indeed, Simon (1973) connected the likelihood ratio test statistic for hypothesis testing under the exponential distribution family with the Kullback−Leibler divergence. Nayak (Nayak, 1983, 1985) used entropy measures, while Basu et al. (2010) used the Hellinger distance for the construction of statistical tests for discrete populations.

Most of the above test statistics are special cases of the family of power divergence test statistics introduced by Cressie & Read (1984) which is given by

$$T_N^\lambda(\hat{\mathbf{p}}, \mathbf{p}_0) = \frac{2N}{\lambda(\lambda+1)} \sum_{i=1}^m \hat{p}_i \left[\left(\frac{\hat{p}_i}{p_{i0}}\right)^\lambda - 1\right], \quad \lambda \in \mathbb{R} \setminus \{0, -1\}, \tag{1.12}$$

where specific values of λ correspond to well known test statistics. Particularly for $\lambda = 1$ and for $\lambda \to 0$ the chi−squared and the likelihood ratio test statistics are obtained respectively.

Recently, Mattheou & Karagrigoriou (2010) proposed the more general (Φ, α)−power divergence family of test statistics, for testing the simple null hypothesis (1.9), which is defined by

$$T_\Phi^\alpha(\hat{\mathbf{p}}, \mathbf{p}_0) = \frac{2N}{\Phi''(1)} \sum_{i=1}^m p_{i0}^{1+\alpha} \Phi\left(\frac{\hat{p}_i}{p_{i0}}\right) \tag{1.13}$$

for any convex function Φ such that $\Phi(1) = \Phi'(1) = 0$ and $\Phi''(1) > 0$. Observe that for $\alpha = 0$ and Φ equal to φ_{Ka}, φ_{KL}, and φ_{CR} the $(\Phi, \alpha)-$power divergence test statistic reduces to chi−squared, likelihood ratio and power divergence test statistics, respectively. Note also that for $\alpha = 0$ and Φ as in (1.1) the test statistic (1.13) reduces to the traditional tests of fit based on the $\varphi-$divergence family of measures.

Under the simple null hypothesis (1.9) the asymptotic distribution of the tests (1.10), (1.11), (1.12) and the $\varphi-$divergence test statistic is chi−squared with $m-1$ degrees of freedom ($\mathcal{X}^2_{m-1}$). These results were established by Pearson (1900), Neyman & Pearson (1928), Cressie & Read (1984) and Zografos et al. (1990), respectively. Mattheou & Karagrigoriou (2010) proved that under (1.9) the $(\Phi, \alpha)-$power divergence test statistic is asymptotically (in distribution) equal to $c\mathcal{X}^2_{m-1}$, where c is between the $\min_i p^{\alpha}_{i0}$ and $\max_i p^{\alpha}_{i0}$, with the most common value used in practice being the middle value: $c = 0.5(\min_i p^{\alpha}_{i0} + \max_i p^{\alpha}_{i0})$ (see Mattheou & Karagrigoriou (2010, Theorem 2)).

For the fixed alternative hypothesis $H_0 : p_i = p_{i0} \ \ vs \ \ H_1 : p_i = p_{ib}$, $i = 1, ..., m$, Mattheou & Karagrigoriou (2010) proved that the power of the test, using the test statistic (1.13) with $p_{i0} = p_{ib}$, is asymptotically equal to

$$\gamma_a = Pr\left(Z \geq \frac{\Phi''(1)\, c\mathcal{X}^2_{m-1,1-\alpha} - 2Nd^{a}_{\Phi}(\mathbf{p}_b, \mathbf{p}_0)}{2\sqrt{N}\sigma_a}\right),$$

where Z is a standard Normal random variable, $\mathcal{X}^2_{m-1,1-\alpha}$ is the $(1-\alpha)$ percentile of the $\mathcal{X}^2_{m-1}$ distribution, which means that it satisfies the relation $Pr(\mathcal{X}^2_{m-1} \leq \mathcal{X}^2_{m-1,1-\alpha})$ $= 1 - \alpha$ and

$$\sigma^2_a = \sum_{j=1}^{m} p_{jb}\left[p^{a}_{j0}\Phi'\left(\frac{p_{jb}}{p_{j0}}\right)\right]^2 - \left[\sum_{j=1}^{m} p_{jb}p^{a}_{j0}\Phi'\left(\frac{p_{jb}}{p_{j0}}\right)\right]^2.$$

It is known that it is not always possible to determine the asymptotic distribution under any alternative. The asymptotic distribution under contiguous alternatives is provided in the sequel. Suppose that the simple null hypothesis indicates that $p_i = p_{i0}$, $i = 1, 2, \ldots, m$ when in fact it is $p_i = p_{ib}$, $\forall i$. As it is well known, if p_{i0} and p_{ib} are fixed then as N tends to infinity, the power of the test tends to 1. In order to examine the situation when the power is not close to 1, we must make it continually harder for the test as N increases. This can be done by allowing the alternative hypothesis steadily closer to the null hypothesis. As a result we define a sequence of alternative hypotheses as follows

$$H_{1,N} : \mathbf{p} = \mathbf{p}_N = \mathbf{p}_0 + \mathbf{d}/\sqrt{N}, \tag{1.14}$$

where $\mathbf{p}_N = (p_{1N}, \ldots, p_{mN})'$ and $\mathbf{d} = (d_1, \ldots, d_m)'$ is a fixed vector such that $\sum_{i=1}^{m} d_i$ $= 0$. This hypothesis is known as *Pitman transition alternative* or *Pitman (local) alternative* or *local contiguous alternative* to the null hypothesis $H_0 : \mathbf{p} = \mathbf{p_0}$ (McManus, 1991). Observe that as N tends to infinity the local contiguous alternative converges to the null hypothesis at the rate $O(N^{-1/2})$.

Mattheou & Karagrigoriou (2010) proved that, under the contiguous alternative hypotheses given in (1.14), the asymptotic distribution of the $(\Phi, \alpha)-$power divergence test statistic $T^{\alpha}_{\Phi}(\hat{\mathbf{p}}, \mathbf{p}_0)$ is a non−central chi−squared times c with $m-1$ degrees of freedom, $c\mathcal{X}_{m-1}(\delta)$, where $c = 0.5\big(\min_i p^a_{i0} + \max_i p^a_{i0}\big)$ and noncentrality parameter $\delta = \sum_{i=1}^m d_i^2/p_{i0}$.

Due to the above results the power of the test under the fixed alternative hypothesis $H_1 : p_i = p_{ib}$ and the local contiguous alternative hypotheses (1.14) can be easily obtained. For the case of the local contiguous alternative hypotheses, the power is given by

$$\gamma_N = Pr(T^{\alpha}_{\Phi}(\hat{\mathbf{p}}, \mathbf{p}_0) > \mathcal{X}^2_{m-1,1-\alpha} | p_i = p_{iN}, i = 1, ..., m) = Pr(\mathcal{X}^2_{m-1}(\delta) > \mathcal{X}^2_{m-1,1-\alpha}).$$

1.4.2 Composite null hypothesis

Although the above simple null hypothesis appears frequently in practice, it is common to test the composite null hypothesis that the unknown distribution belongs to a known parametric family

$$\mathcal{P} = \left\{\mathbf{p}(\boldsymbol{\theta}) = (p_1(\boldsymbol{\theta}), \ldots, p_m(\boldsymbol{\theta}))^\top : \boldsymbol{\theta} = (\theta_1, \ldots, \theta_s)^\top \in \Theta\right\}$$

with $\Theta \subseteq \mathbb{R}^s$ $(s < m-1)$. Note that $\mathcal{P} \subset \Delta_m = \{(p_1, \ldots, p_m)^\top : p_i > 0, i = 1, \ldots, m, \sum_{i=1}^m p_i = 1\}$. Pearson encountered this problem in the well known chi−squared test statistic and suggested the use of a consistent estimator for the unknown parameter. He further claimed that the asymptotic distribution of the resulting test statistic, under the null hypothesis, is a chi−squared random variable with $m-1$ degrees of freedom. Later, for the same test, Fisher (1924) established that the correct distribution has $m-1-s$ degrees of freedom, when the method of estimation is the maximum likelihood for either discretized or grouped data. The result was later discussed by Neyman (1949) and recently by Menéndez et al. (2001) and Pardo (2006, p.258). In this case, since the null distribution depends on the unknown parameter $\boldsymbol{\theta}$, a consistent estimator of $\boldsymbol{\theta}$ is required.

Now consider a random vector $\mathbf{X} = (X_1, \ldots, X_m)^\top$ from the multinomial distribution $M(N, \mathbf{p})$, where $\mathbf{p} = (p_1, \ldots, p_m)^\top$ is the vector of true probabilities and N is the number of the experiments. The vector $\mathbf{p}$ is assumed to be unknown but belonging to the family $\mathcal{P}$. In other words, the true value $\boldsymbol{\theta}_0$ of the parameter $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_s)^\top$ is assumed to be unknown. For a realization $(x_1, \ldots, x_m)^\top$ of $\mathbf{X} = (X_1, \ldots, X_m)^\top$ we have $N = \sum_{i=1}^m x_i$, $x_i \geq 0$. Then $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_m)^\top$ with $\hat{p}_i = x_i/N$ is the nonparametric maximum likelihood estimator of the true probabilities $p_i, i = 1, \ldots, m$.

In order to estimate the unknown parameter $\boldsymbol{\theta}$ one could use the maximum likelihood method, where in the multinomial setting it suffices to find a value $\hat{\boldsymbol{\theta}}$ over Θ that maximizes $\sum_{i=1}^m \hat{p}_i \log p_i(\boldsymbol{\theta})$. Then $\hat{\boldsymbol{\theta}}$ is the point maximum likelihood estimator (MLE) of the parameter $\boldsymbol{\theta}$. Optimal estimating approaches, like the maximum likelihood estimation, are available in the literature (e.g. Papaioannou et al., 2007). Equivalently, the MLE can be defined as the argument that minimizes the Kullback−Leibler divergence:

$$\hat{\boldsymbol{\theta}} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_{\mathrm{KL}}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})).$$

Cressie & Read (1984) proposed the minimum power divergence estimator

$$\hat{\boldsymbol{\theta}}^{\lambda}_{CR} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d^{\lambda}_{CR}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$$

and studied its properties. As they argue (Read & Cressie, 1988, p.30) this estimator is a natural generalization of the MLE. Furthermore, Morales et al. (1995) considered the more general minimum $\varphi-$divergence estimator (MφDE), defined as

$$\hat{\boldsymbol{\theta}}_{\varphi} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_{\varphi}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})). \tag{1.15}$$

In the sequel we assume that the model is correct, so that there exists a value $\boldsymbol{\theta}_0$ which leads to the true probabilities that generate the observations. Under the regularity conditions introduced by Birch (1964) and the assumption that the function $\mathbf{p}(\boldsymbol{\theta})$ has continuous second partial derivatives in a neighbourhood of $\boldsymbol{\theta}_0$, it holds

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\varphi} = \boldsymbol{\theta}_0 &+ \left(\mathbf{A}(\boldsymbol{\theta}_0)^{\top}\mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^{\top} diag\left(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}\right)\left(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\right)\\ &+ o(\left\|\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\right\|)\end{aligned} \tag{1.16}$$

and

$$\sqrt{N}(\hat{\boldsymbol{\theta}}_{\varphi} - \boldsymbol{\theta}_0) \xrightarrow[N\to\infty]{L} \mathrm{N}_s\left(\mathbf{0}, \left(\mathbf{A}(\boldsymbol{\theta}_0)^{\top}\mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1}\right) \tag{1.17}$$

where

$$\mathbf{A}(\boldsymbol{\theta}_0) = diag\left(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}\right)\left(\frac{\partial p_i(\boldsymbol{\theta}_0)}{\partial \theta_j}\right)_{\substack{i=1,\dots,m\\ j=1,\dots,s}},$$

and $\left(\mathbf{A}(\boldsymbol{\theta}_0)^{\top}\mathbf{A}(\boldsymbol{\theta}_0)\right)$ is the Fisher information matrix associated with the multinomial model. Note that, for a generic vector $\mathbf{v} = (v_1, \dots, v_m)^{\top}$, $diag(\mathbf{v})$ denotes the $m \times m$ diagonal matrix whose (i,i) element is v_i, $i = 1, \dots, m$.

Also note that, the minimum $\varphi-$divergence estimator under misspecified models, i.e. when it does not exist a value $\boldsymbol{\theta}_0$ which leads to the true probabilities of the multinomial model, has been studied by Jiménez-Gamero et al. (2011).

For testing the composite null hypothesis

$$H_0\colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0) \text{ vs. } H_1\colon \mathbf{p} \neq \mathbf{p}(\boldsymbol{\theta}_0),\ \boldsymbol{\theta}_0 \in \Theta \subseteq \mathbb{R}^s \tag{1.18}$$

the family of $\varphi-$divergence test statistics (φDTS) can be used, were the unknown parameter $\boldsymbol{\theta}_0$ is estimated by the minimum $\varphi-$divergence estimator given in (1.15), for which we have that (Menéndez et al., 2001)

$$T_{\varphi_1}\left(\hat{\boldsymbol{\theta}}_{\varphi_2}\right) = \frac{2N}{\varphi_1''(1)} d_{\varphi_1}\left(\hat{\mathbf{p}}, \mathbf{p}\left(\hat{\boldsymbol{\theta}}_{\varphi_2}\right)\right) \xrightarrow[n\to\infty]{L} \mathcal{X}^2_{m-1-s}. \tag{1.19}$$

where φ_1, φ_2 as in (1.1). Observe that functions φ_1, φ_2 are not necessarily the same. This is a general family of test statistics that includes as particular case the notable family of power$-$divergence test statistics, which has been introduced by Cressie & Read (1984) and is based on the measure given in (1.3). This family of test statistics is given by

$$T_{CR}^{\lambda_1}\left(\hat{\boldsymbol{\theta}}_{CR}^{\lambda_2}\right) = \frac{2N}{\lambda_1\left(\lambda_1+1\right)}\sum_{i=1}^{m}\hat{p}_i\left(\left(\frac{\hat{p}_i}{p_i(\hat{\boldsymbol{\theta}}_{CR}^{\lambda_2})}\right)^{\lambda_1} - 1\right), \tag{1.20}$$

where $\lambda \neq -1, 0$, $-\infty < \lambda < \infty$.

Particular values of $\lambda(=\lambda_1=\lambda_2)$ in (1.20) correspond to well known test statistics: chi−squared test statistic ($\lambda=1$), likelihood ratio test statistic ($\lambda\to 0$), Freeman−Tukey test statistic ($\lambda=-1/2$), minimum discrimination information (or modified likelihood ratio) test statistic (Gokhale & Kullback, 1978; Kullback, 2006) ($\lambda\to -1$), modified chi−squared test statistic (Neyman, 1949) ($\lambda=-2$) and Cressie−Read test statistic ($\lambda=2/3$).

Menéndez et al. (2001) proved that under the alternative hypothesis $H_1 : p_i = p_{ib}$, $i=1,...,m$ and the assumption that

$$\sqrt{N}\left((\hat{\mathbf{p}},\mathbf{p}(\hat{\boldsymbol{\theta}}_{\varphi_2})) - (\mathbf{p}_b,\mathbf{p}(\boldsymbol{\theta}_1))\right) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{0},\boldsymbol{\Sigma}^*)$$

where $\boldsymbol{\theta}_1 = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_{\varphi_2}(\mathbf{p}_b,\mathbf{p}(\boldsymbol{\theta}))$,

$$\boldsymbol{\Sigma}^* = \begin{pmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12}\\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{pmatrix},\ \boldsymbol{\Sigma}_{11} = diag(\mathbf{p}_b) - \mathbf{p}_b\mathbf{p}_b^{\top}\ \mathit{and}\ \boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21},$$

the asymptotic power of the test using the $\varphi-$divergence test statistic (1.19) is equal to

$$\gamma_{\varphi_1} = Pr\left(Z \geq (2\sqrt{N}\sigma_{\varphi})^{-1}\left(\varphi_1''(1)\chi^2_{m-1-s,1-a} - 2Nd_{\varphi_1}(\mathbf{p}_b,\mathbf{p}(\boldsymbol{\theta}_1))\right)\right).$$

Note that in this case $\mathbf{p}(\boldsymbol{\theta}_1) \neq \mathbf{p}_b$ and $\hat{\mathbf{p}} \xrightarrow{p} \mathbf{p}_b$ as well as $\mathbf{p}(\hat{\boldsymbol{\theta}}_{\varphi_2}) \xrightarrow{p} \mathbf{p}(\boldsymbol{\theta}_1)$. Moreover, they proved that the asymptotic distribution of the test statistic (1.19) under the contiguous alternative hypothesis

$$H_{1,N}\colon \mathbf{p} = \mathbf{p}_N = \mathbf{p}(\boldsymbol{\theta}_0) + \mathbf{d}/\sqrt{N} \tag{1.21}$$

is a noncentral chi−squared with $m-1-s$ degrees of freedom and noncentrality parameter δ given by

$$\delta = \mathbf{d}^{\top} diag\left(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}\right)\left(I - \mathbf{A}(\boldsymbol{\theta}_0)\left(\mathbf{A}(\boldsymbol{\theta}_0)^{\top}\mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^{\top}\right) diag\left(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}\right)\mathbf{d},$$

where I is the identity matrix. Since in this case the distribution of the $\varphi-$divergence test statistic (1.19) does not depend on the form of the functions φ_1, φ_2, the power of the test is given by $Pr(\mathcal{X}^2_{m-1-s}(\delta) > \mathcal{X}^2_{m-1-s,1-\alpha})$.

Furthermore, if we suppose that the parameter $\boldsymbol{\theta}$ is constrained by ν functions, $f_1(\boldsymbol{\theta})$, $\ldots$, $f_{\nu}(\boldsymbol{\theta})$ such that $f_k(\boldsymbol{\theta}) = 0$, $k = 1,\ldots,\nu$ and $\nu < s < m-1$ we can derive the so called restricted minimum divergence estimators. Pardo et al. (2002) proposed the restricted minimum $\varphi-$divergence estimator

$$\hat{\boldsymbol{\theta}}^{r}_{\varphi} = \arg \inf_{\boldsymbol{\theta}\in\Theta\subseteq\mathbb{R}^s\colon\ f_k(\boldsymbol{\theta})=0,k=1,\ldots,\nu} d_{\varphi}(\hat{\mathbf{p}},\mathbf{p}(\boldsymbol{\theta})). \tag{1.22}$$

By further assuming that each constraint function $f_k(\boldsymbol{\theta})$ has continuous second partial derivatives, while the $\nu\times s$ matrix $\mathbf{Q}(\boldsymbol{\theta}_0) = (\partial f_k(\boldsymbol{\theta}_0)/\partial\theta_j)$, $k = 1,\ldots,\nu$, $j = 1\ldots,s$ is of full rank, Pardo et al. (2002) proved that

$$\begin{aligned}\hat{\boldsymbol{\theta}}^{r}_{\varphi} = \boldsymbol{\theta}_0 &+ \mathbf{H}^*(\boldsymbol{\theta}_0)\left(\mathbf{A}(\boldsymbol{\theta}_0)^{\top}\mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^{\top} diag\left(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}\right)\left(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\right)\\ &+ o(\|\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\|)\end{aligned} \tag{1.23}$$

and

$$\sqrt{N}(\hat{\boldsymbol{\theta}}^r_{\varphi} - \boldsymbol{\theta}_0) \xrightarrow[N\to\infty]{L} \mathrm{N}_s\left(\mathbf{0}, \mathbf{H}^*(\boldsymbol{\theta}_0)\left(\mathbf{A}(\boldsymbol{\theta}_0)^\top \mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1} \mathbf{H}^*(\boldsymbol{\theta}_0)^\top\right), \tag{1.24}$$

where

$$\mathbf{H}^*(\boldsymbol{\theta}_0) = I - \left(\mathbf{A}(\boldsymbol{\theta}_0)^\top \mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1} \mathbf{Q}(\boldsymbol{\theta}_0)^\top \left(\mathbf{Q}(\boldsymbol{\theta}_0)\left(\mathbf{A}(\boldsymbol{\theta}_0)^\top \mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1} \mathbf{Q}(\boldsymbol{\theta}_0)^\top\right)^{-1} \mathbf{Q}(\boldsymbol{\theta}_0).$$

In addition they showed that

$$T_{\varphi_1}\left(\hat{\boldsymbol{\theta}}^r_{\varphi_2}\right) = \frac{2N}{\varphi_1''(1)} d_{\varphi_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^r_{\varphi_2})\right) \xrightarrow[n\to\infty]{L} \mathcal{X}^2_{m-1-s+\nu}.$$

For further references, please refer to Read & Cressie (1988); Alin & Kurt (2008); Toma (2009); Jiménez-Gamero et al. (2011); Kim & Lee (2014); Neath et al. (2015); Ghosh (2016); Jiménez-Gamero & Batsidis (2017); Basu et al. (2020).

Case of continuous r.v.
The above procedures can be applied in discretized versions of continuous random variables. Indeed, let $\mathbf{Y} = (Y_1, \dots, Y_N)^\top$ be a random vector from the continuous distribution F. We can test the composite null hypothesis, $H_0\colon F = F_{\boldsymbol{\theta}_0}$, that the distribution function F is a member of a parametric family $\{F_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\Theta}$, $\Theta \subseteq \mathbb{R}^s$ by partitioning the data−range into m disjoint intervals and testing the hypothesis $H_0\colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0)$ about the parameter−vector of the resulting multinomial distribution, for some unknown $\boldsymbol{\theta}_0 \in \Theta \subseteq \mathbb{R}^s$ and $s < m - 1$.

Let $P = \{E_i\}_{i=1,\dots,m}$ be a partition of the real line $\mathbb{R}$ into m intervals. The probabilities of the intervals E_i, $i = 1, \dots, m$ depend on the unknown parameter $\boldsymbol{\theta}_0$, in such a way that $p_i(\boldsymbol{\theta}_0) = \Pr_{F_{\boldsymbol{\theta}_0}}(E_i) = \int_{E_i} dF_{\boldsymbol{\theta}_0}$. Let $x_i = \sum_{j=1}^N I_{E_i}(Y_j)$ where

$$I_{E_i}(Y_j) = \begin{cases} 1, & \text{if } Y_j \in E_i \\ 0, & \text{otherwise} \end{cases},$$

$\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_m)^\top$ with $\hat{p}_i = x_i/N$, $i = 1, \dots, m$ be the nonparametric maximum likelihood estimator of the true probability of the E_i interval and $\sum_{i=1}^m x_i = N$.

We close this section with a discussion about the number m of classes required in tests of fit. The partition of the data range is a delicate matter since it is frequently associated with loss of information. For a thorough investigation on the issue, the interested reader is referred to the works by Ferentinos & Papaioannou (1979, 1983). The problem of determining the optimum number of classes has a long history. Mann & Wald (1942), Cochran (1952), Dahiya & Gurland (1973) and Harrison (1985) proposed similar techniques for the case of the chi−squared test which though arrive at diverse conclusions with recommendations ranging from 3 to 24 classes. In almost all these cases, the optimum number of classes depends on various factors like the type of the alternative hypothesis, the "distance" between the alternative hypothesis and the data, the minimum power to be achieved, the significance level of the test and the sample size. One should be always aware of the danger associated with the use of too many classes in cases where the observations are spread too thinly over the data range. The power of the test together with the size are the key factors in deciding if the increase of the number of classes is useful. It should be noted that even in cases, where the number of classes is allowed to increase as the sample size increases, specific assumptions should be imposed in order to secure satisfactory asymptotic results. One such assumption is the well−known sparseness assumption according to which for $m = m(N)$, $\lim_{N\to\infty} N/m = c < \infty$ (see Holst, 1972, Dale, 1986 and Read & Cressie, 1988). Typically, for classes at most equal to 10, the results are quite satisfactory.

1.4.3 Contingency table analysis

Divergence measures play also a prominent role in contingency table analysis. If we consider the case of three categorical variables, say X, Y and Z with corresponding I, J and K categories, then the probability mass of a realization of a randomly selected subject is denoted by $p_{ijk}(\boldsymbol{\theta}) = Pr(X = i, Y = j, Z = k) > 0$, where here and in what follows $i = 1, \ldots, I$, $j = 1, \ldots, J$, $k = 1, \ldots, K$ unless otherwise stated. The associated probability vector is given as $\mathbf{p}(\boldsymbol{\theta}) = \{p_{ijk}(\boldsymbol{\theta})\}$, where

$$p_{ijk}(\boldsymbol{\theta}) = \begin{cases} \theta_{ijk}, & (i,j,k) \neq (I,J,K) \\ 1 - \sum\limits_{\substack{i=1 \\ (i,j,k)\neq(I,J,K)}}^{I} \sum\limits_{j=1}^{J} \sum\limits_{k=1}^{K} \theta_{ijk}, & (i,j,k) = (I,J,K) \end{cases}$$

and the parameter space is $\Theta = \{\theta_{ijk}, (i,j,k) \neq (I,J,K)\}$. The sample estimator of $p_{ijk}(\boldsymbol{\theta})$ is $\hat{p}_{ijk} = n_{ijk}/N$, where n_{ijk} is the frequency of the corresponding (i,j,k) cell.

In this set−up, at least eight models of great importance of the associated three−way contingency table arise (see Christensen, 1997). These models are:

(a) Complete independence,

  (a_1) X, Y and Z are all independent.

(b) One variable independent of the other two,

  (b_1) X, independent of Y and Z, but Y and Z are not necessarily independent.

  (b_2) Y, independent of X and Z, but X and Z are not necessarily independent.

  (b_3) Z, independent of X and Y, but X and Y are not necessarily independent.

(c) Conditional independence,

  (c_1) Given a value k of Z, X and Y are independent.

  (c_2) Given a value j of Y, X and Z are independent.

  (c_3) Given a value i of X, Y and Z are independent.

(d) Additional hypotheses,

  (d_1) No second−order interaction, also known as no three−factor interaction.

Each of these models impose specific constraints on the unknown parameter $\boldsymbol{\theta}_0$, thus the methodology regarding the restricted minimum divergence estimators and the associated test statistics can be incorporated in order to proceed with the statistical inference. In this case, the formulation of the restricted minimum φ−divergence (rMD) estimator and the φ−divergence test statistic are given below

$$T_{\varphi_1}\left(\hat{\boldsymbol{\theta}}^r_{\varphi_2}\right) = \frac{2N}{\varphi_1''(1)} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} p_{ijk}(\hat{\boldsymbol{\theta}}^r_{\varphi_2})\varphi_1\left(\frac{\hat{p}_{ijk}}{p_{ijk}(\hat{\boldsymbol{\theta}}^r_{\varphi_2})}\right), \tag{1.25}$$

where $\hat{p}_{ijk}$ as above and the rMD estimator as

$$\hat{\boldsymbol{\theta}}^r_{\varphi_2} = \arg \inf_{\{\boldsymbol{\theta}\in\Theta\subseteq\mathbb{R}^s:\ f_k(\boldsymbol{\theta})=0, k=1,\ldots,\nu\}} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} p_{ijk}(\boldsymbol{\theta})\varphi_2\left(\frac{\hat{p}_{ijk}}{p_{ijk}(\boldsymbol{\theta})}\right). \tag{1.26}$$

For a comprehensive study regarding the use of divergence measures in three−way contingency tables we refer the interested reader to Pardo (2010), while for two−way contingency tables to Pardo (2006, Ch.8).

1.4.4 Penalized divergence measures

Members of the more general families of minimum divergence estimators, introduced in subsection 1.4.2, can perform better in terms of robustnees, but rather poor, compared to the *MLE*, in terms of efficiency, when the sample size is small. One reason for this poor behaviour is the fact that these robust estimators attach disproportional large weight on empty cells (see e.g. Harris & Basu, 1994; Basu et al., 1996, 1997). In order to illustrate this trade−off between robustness and efficiency, caused by the empty cells, Basu & Basu (1998) based on an equivalent form of the family of power divergences (1.3), proposed the following decomposition

$$\begin{aligned} {}^*d^{\lambda}_{CR}(\hat{\mathbf{p}},\mathbf{p}(\boldsymbol{\theta})) &= \sum_{i:\hat{p}_i>0}\left\{\frac{\hat{p}_i}{\lambda(\lambda+1)}\left[\left(\frac{\hat{p}_i}{p_i(\boldsymbol{\theta})}\right)^{\lambda}-1\right]+\frac{p_i(\boldsymbol{\theta})-\hat{p}_i}{\lambda+1}\right\} \\ &+\frac{1}{\lambda+1}\sum_{i:\hat{p}_i=0} p_i(\boldsymbol{\theta}). \end{aligned} \tag{1.27}$$

Note that the second component in (1.27) refers to the contribution of the empty cells, which can become very large for values of λ close to -1. For the resolution of this issue these authors considered a penalization by exchanging the weight of the empty cells from $(\lambda+1)^{-1}$ to 1, which is the weight imposed to the empty cells by the Kullback−Leibler divergence. Later Alin (2007) considered as weight a tuning parameter h and studied the behaviour of the resulting estimators and test statistics in various frameworks (see also, Alin & Kurt, 2008; Alin, 2008, 2012). Thus, the penalized family of power divergences has the following form

$$\begin{aligned} d^{\lambda}_{CR,h}(\mathbf{p},\mathbf{q}) &= \sum_{i:\hat{p}_i>0}\left\{\frac{\hat{p}_i}{\lambda(\lambda+1)}\left[\left(\frac{\hat{p}_i}{p_i(\boldsymbol{\theta})}\right)^{\lambda}-1\right]+\frac{p_i(\boldsymbol{\theta})-\hat{p}_i}{\lambda+1}\right\} \\ &+h\sum_{i:\hat{p}_i=0} p_i(\boldsymbol{\theta}). \end{aligned} \tag{1.28}$$

The minimum penalized power divergence estimator is defined as

$$\hat{\boldsymbol{\theta}}^{\lambda}_{CR,h} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d^{\lambda}_{CR,h}(\hat{\mathbf{p}},\mathbf{p}(\boldsymbol{\theta}))$$

and the associated test statistic as

$$T^{\lambda_1}_{CR,h_1}\left(\hat{\boldsymbol{\theta}}^{\lambda_2}_{CR,h_2}\right) = 2Nd^{\lambda_1}_{CR,h_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^{\lambda_2}_{CR,h_2})\right).$$

Note that, asymptotically the probability of observing empty cells goes to zero, thus the asymptotic distribution of the estimators and the test statistics remain intact under the penalization.

Alba-Fernández et al. (2018) considered the $\varphi-$divergence family of measures under model misspecification. In this framework, $\mathbf{X} = (X_1, \ldots, X_m)^\top \sim M(N, \mathbf{p})$, where $\mathbf{p} = (p_1, \ldots, p_m)^\top \in \Delta_{0m} = \{(p_1, \ldots, p_m)^\top : p_i \geq 0, i = 1, \ldots, m, \sum_{i=1}^{m} p_i = 1\}$, in other words some cell frequencies may be equal to zero even for large sample sizes. Let $\mathbf{p} = (p_1, \ldots, p_m)^\top \in \Delta_{0m}$ and $\mathbf{q} = (q_1, \ldots, q_m)^\top \in \Delta_m$, the $\varphi-$divergence family can be written in the form

$$d_\varphi(\mathbf{p}, \mathbf{q}) = \sum_{i:p_i>0} q_i \varphi\left(\frac{p_i}{q_i}\right) + \varphi(0) \sum_{i:p_i=0} q_i.$$

The penalized $\varphi-$divergence family is derived from the above expression by exchanging the fixed weight $\varphi(0)$ regarding the empty cells, with the tuning parameter h and is given as

$$d_{\varphi,h}(\mathbf{p}, \mathbf{q}) = \sum_{i:p_i>0} q_i \varphi\left(\frac{p_i}{q_i}\right) + h \sum_{i:p_i=0} q_i.$$

The minimum penalized $\varphi-$divergence estimator (MPφDE) is defined as

$$\hat{\boldsymbol{\theta}}_{\varphi,h} = \arg \inf_{\boldsymbol{\theta}\in\Theta} d_{\varphi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$$

and the associated penalized $\varphi-$divergence test statistic (PφDTS) as

$$T_{\varphi_1,h_1}\left(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2}\right) = \frac{2N}{\varphi_1''(1)} d_{\varphi_1,h_1}\left(\hat{\mathbf{p}}, \mathbf{p}\left(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2}\right)\right).$$

Under specific assumptions, the authors provided asymptotic results for the case of model misspecification, i.e. when $\mathbf{p} = (p_1, \ldots, p_m)^\top \in \Delta_{0m}$. Specifically, they proved that the MPφDE ($\hat{\boldsymbol{\theta}}_{\varphi,h}$) is consistent for $\boldsymbol{\vartheta}_0$, where in this case $\boldsymbol{\vartheta}_0 = \arg\inf_{\boldsymbol{\theta}\in\Theta}$ $d_{\varphi,h}$ $(\mathbf{p},$ $\mathbf{p}(\boldsymbol{\theta}))$ and that $\sqrt{N}(\hat{\boldsymbol{\theta}}_{\varphi,h} - \boldsymbol{\vartheta}_0)$ is asymptotically normal with mean 0. In addition to that, asymptotic results regarding the $d_{\varphi_1,h_1}\left(\hat{\mathbf{p}}, \mathbf{p}\left(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2}\right)\right)$ are provided. Under the assumption that the parametric model is correctly specified i.e. when $\mathbf{p}$ $= (p_1, \ldots, p_m)^\top \in \mathcal{P}$ the following simplifications are obtained

$$\sqrt{N}(\hat{\boldsymbol{\theta}}_{\varphi,h} - \boldsymbol{\theta}_0) \xrightarrow[N\to\infty]{L} \mathrm{N}_s\left(\mathbf{0}, \left(\mathbf{A}(\boldsymbol{\theta}_0)^\top \mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1}\right)$$

and

$$T_{\varphi_1,h_1}\left(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2}\right) = \frac{2N}{\varphi_1''(1)} d_{\varphi_1,h_1}\left(\hat{\mathbf{p}}, \mathbf{p}\left(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2}\right)\right) \xrightarrow[N\to\infty]{L} \mathcal{X}^2_{m-1-s}.$$

In other words, when the hypothesized parametric model holds, the asymptotic results of the MPφDE and PφDTS are the same with those of the MφDE and φDTS. These results coincide with those provided by Mandal et al. (2010).

In order to achieve a better approximation to the underlying distribution of the PφDTS, after the observation that the asymptotic $\mathcal{X}^2$ approximation is rather poor, Alba-Fernández et al. (2018), triggered by the fact that the MPφDE always converges to a well−defined limit, proposed the bootstrap approximation. The null bootstrap distribution of $T_{\varphi_1,h_1}\left(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2}\right)$ is the conditional distribution of

$$T^*_{\varphi_1,h_1}\left(\hat{\boldsymbol{\theta}}^*_{\varphi_2,h_2}\right) = \frac{2N}{\varphi_1''(1)} d_{\varphi_1,h_1}\left(\hat{\mathbf{p}}^*, \mathbf{p}(\hat{\boldsymbol{\theta}}^*_{\varphi_2,h_2})\right)$$

given $(X_1,\ldots,X_m)$, where $\hat{\mathbf{p}}^*$ is defined as $\hat{\mathbf{p}}$ with $(X_1,\ldots,X_m)$ replaced by $(X_1^*, \ldots, X_m^*) \sim M(N,\mathbf{p}(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2}))$ and $\hat{\boldsymbol{\theta}}^*_{\varphi_2,h_2} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_{\varphi_2,h_2}(\hat{\mathbf{p}}^*, \mathbf{p}(\boldsymbol{\theta}))$. By denoting the bootstrap conditional probability law, given $(X_1,\ldots,X_m)$, as Pr_* and letting $\mathbf{X} \sim M(N,\mathbf{p})$ with $\mathbf{p} \in \Delta_m$ we have that

$$\sup_x \left| Pr_*(T^*_{\varphi_1,h_1}\left(\hat{\boldsymbol{\theta}}^*_{\varphi_2,h_2}\right) \leq x) - Pr(Y \leq x) \right| \xrightarrow[N\to\infty]{P} 0,$$

where $Y \sim \mathcal{X}^2_{m-1-s}$. The proposed bootstrap $p-$value algorithm is comprised from the following steps :

1. Calculate the observed value of the test statistic based on the available data $(X_1,\ldots,X_m)$, $T^{obs} \equiv T_{\varphi_1,h_1}(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2})$, where $\hat{\boldsymbol{\theta}}_{\varphi_2,h_2} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_{\varphi_2,h_2}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$.
2. Generate B bootstrap samples $(X_1^{*b},\ldots,X_m^{*b})^\top \sim M(N,\mathbf{p}(\hat{\boldsymbol{\theta}}_{\varphi_2,h_2}))$, $b = 1,\ldots,B$.
3. Calculate the statistic for each bootstrap sample obtaining $T^{*b} \equiv T^{*b}_{\varphi_1,h_1}(\hat{\boldsymbol{\theta}}^{*b}_{\varphi_2,h_2})$, $b = 1,\ldots,B$.
4. Approximate the $p-$value by means of the expressions

   $$\hat{p}_{boot} = \frac{\text{card}\left\{b : T^{*b} \geq T^{obs}\right\}}{B}$$

   where card$\{A\}$ denotes the number of elements in A.
5. Reject the null hypothesis if $\hat{p}_{boot} \leq \alpha$ (nominal size).

1.5 Scope of this thesis

In this thesis we study and exploit the general $(\Phi,\alpha)-$power divergence family and prominent sub−families of it, like the BHHJ family of measures. Rigorous asymptotical results, regarding the estimators and the associated test statistics, are provided while their applicability is presented through extensive simulation studies, covering

- in Chapter 2 the case of multinomial populations,
- in Chapter 3 the case of conditional independence in cross tabulations,
- in Chapter 4 the case of zero frequency cells and,
- in Chapter 5 the case of the collective risk model in insurance science.

In particular, in Chapter 2 the minimum $(\Phi,\alpha)-$power divergence estimator is defined, while its decomposition and asymptotic distribution are provided. Furthermore, the associated general family of double index $(\Phi,\alpha)-$power divergence test statistics is proposed, while its asymptotic distribution, both under the null hypothesis and the contiguous alternatives, along with the power of the test, are provided. Finally, the asymptotic approximation to the underlying distribution of the test statistic is verified via simulation studies. Additionally, the undertaken simulation studies reveal that, members of the proposed family of test statistics can be considered as worthy counterparts to the classical ones.

In Chapter 3, the focus is placed on cross tabulations with emphasis on the conditional independence. The restricted minimum $(\Phi, \alpha)-$power divergence estimator is defined for estimation of the parameters under constraints. Once more, the associated double index family of test statistics is proposed, for testing procedures. Theoretical results analogous to those of Chapter 2 are provided. The undertaken simulation study deals with the case of conditional independence in $2 \times 2 \times 2$ contingency tables and reveals that members of the proposed family of test statistics, perform as well as or even better, in terms of size and power, than the classical ones.

In Chapter 4, the modified $(\Phi, \alpha)-$power divergence family along with the modified minimum $(\Phi, \alpha)-$power divergence estimator and the modified double index $(\Phi, \alpha)-$power divergence test statistic are defined for the first time. Theoretical results, regarding the decomposition and the asymptotic distribution of the estimator under misspecified models, are provided. The asymptotic distribution of the proposed family of test statistics, when the postulated model is correctly specified, is provided as well. The undertaken simulation study, examines the efficiency and the robustness of the proposed estimator, as well as, the performance of the proposed test statistic in terms of size, not only under the regular case but also under the presence of contaminating data. The study indicates that, members of the proposed family of test statistics are more stable in terms of size, when contaminating data are present.

Finally, Chapter 5 is devoted to the collective risk model, which is a technique of paramount importance, in general insurance industry, for the evaluation of the expected cost of total losses in a period. In this chapter, the applicability of the discussed methodology is presented through a paradigm that focuses on the modelling of the frequency and the severity of claims.

The syntax for the various simulation studies appears in the Appendix C

Chapter 2

Statistical Inference for Multinomial Populations based on a Double Index Family of Test Statistics

The present chapter deals with parameter estimation and hypothesis testing for multinomial populations based on a general class of measures, namely the $(\Phi, \alpha)-$ power divergence family. In particular, the general family of double index $(\Phi, \alpha)-$ test statistics that involves two indices is proposed, the values of which playing a key role in the effectiveness and accuracy of the proposed methodology. The asymptotic properties of the associated estimators are examined together with the power of the test and the asymptotic distribution under the contiguous alternatives. Finally, we explore through extensive simulations, the effect of the shape of the multinomial distribution on the performance of the proposed test. The theoretical results of this chapter along with the simulation study regarding the verification of the asymptotic results (see Subsection 2.4.1), have been published in Meselidis & Karagrigoriou (2020)[1]. The remaining two simulation studies, regarding the comparison of the test statistics and the examination under contaminated data (see Subsections 2.4.2 and 2.4.3), have been published in Karagrigoriou & Meselidis (2020)[2] and Meselidis et al. (2022)[3], respectively.

2.1 Introduction

Divergence measures are used as indices of similarity or dissimilarity between populations. They are also used either to measure mutual information concerning two variables, to perform tests and to estimate parameters, or to construct selection criteria. Measures of divergence or discrepancy are used as a way to evaluate the distance (divergence) between any two populations or functions.

---

[1] Meselidis, C. and Karagrigoriou, A. (2020). Statistical Inference for Multinomial Populations based on a Double Index Family of Test Statistics, *Journal of Statistical Computation and Simulation*, 90:10, 1773−1792, DOI: 10.1080/00949655.2020.1747069

[2] Karagrigoriou, A. and Meselidis, C. (2020). Inferential Statistics Based on Measures of Information and Divergence. In: *Statistical Topics and Stochastic Models for Dependent Data with Applications* (eds V.S. Barbu and N. Vergne). John Wiley & Sons, Ltd. Chap. 10, 215−234. DOI: 10.1002/9781119779421.ch10

[3] Meselidis, C., Karagrigoriou, A. and Papaioannou T. (2022). Data Analysis based on Entropies and Measures. In: *Data Analysis and Related Applications: Theory and Practice.* (eds K. N. Zafeiris et al.) Vol. 1. John Wiley & Sons, Ltd. Chap. 18. URL: https://www.iste.co.uk/book.php?id=1927

As mentioned in Chapter 1 (see, Sections 1.3 and 1.4) Mattheou & Karagrigoriou (2010) proposed a general class of divergence measures, the $(\Phi, \alpha)-$power divergence family. Let $\mathbf{p} = (p_1, \ldots, p_m)^\top$ and $\mathbf{q} = (q_1, \ldots, q_m)^\top$ be two discrete finite probability distributions, the $(\Phi, \alpha)-$power divergence family in the discrete case is given in (1.7) and repeated here for easy access

$$d_\Phi^\alpha(\mathbf{p}, \mathbf{q}) = \begin{cases} \sum_{i=1}^m q_i^{1+\alpha} \Phi\left(\frac{p_i}{q_i}\right); & \alpha > 0 \\ \sum_{i=1}^m q_i \Phi\left(\frac{p_i}{q_i}\right); & \alpha = 0, \end{cases} \tag{2.1}$$

$\Phi \in F$, where F is the class of all convex functions $\Phi(u), u \geq 0$ such that $\Phi(1) = 0, \Phi'(1) = 0$ and $\Phi''(1) > 0$, with $0\Phi(u/0) = \lim_{x\to\infty}[\Phi(x)/x], u > 0$ and $0\Phi(0/0) = 0$. If

$$\Phi(u) = 1 - \left(1 + \frac{1}{\alpha}\right) u + \frac{u^{1+\alpha}}{\alpha} \tag{2.2}$$

expression (2.1) reduces to the BHHJ$-$power divergence (Basu et al., 1998), which is the discrete version of (1.5) on page 5 , given by

$$d_{\mathrm{BHHJ}}^\alpha(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^m q_i^{1+\alpha} - \left(1 + \frac{1}{\alpha}\right) \sum_{i=1}^m p_i q_i^\alpha + \frac{1}{\alpha} \sum_{i=1}^m p_i^{1+\alpha}.$$

The most well known family of divergence measures is the $\varphi-$divergence family (Csiszár, 1963; Ali & Silvey, 1966) also known as Csiszar's $\varphi-$divergence, defined by (see, Section 1.1.1)

$$d_\varphi(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^m q_i \varphi\left(\frac{p_i}{q_i}\right) \tag{2.3}$$

where $\varphi(u), u \geq 0$ is a convex function such that $\varphi(1) = 0$, $\varphi'(1) = 0$ and $\varphi''(1) > 0$, with $0\varphi(0/0) = 0$ and $0\varphi(u/0) = \lim_{x\to\infty}[\varphi(x)/x], u > 0$. Observe that the $\varphi-$divergence family can be derived from (2.1) for $\Phi(u) = \varphi(u)$ and $\alpha = 0$.

Other important divergence measures that can be derived from (2.1) for $\alpha = 0$ are the Kullback$-$Leibler (KL) divergence $d_{\mathrm{KL}}(\mathbf{p}, \mathbf{q})$ (Kullback & Leibler, 1951) for $\Phi(u) = u \log u + 1 - u$, Pearson's chi$-$squared divergence $d_{\mathrm{P}}(\mathbf{p}, \mathbf{q})$ for $\Phi(u) = 1/2(u-1)^2$ (also known as Kagan's divergence; Kagan (1963)) and Matusita's divergence $d_{\mathrm{M}}(\mathbf{p}, \mathbf{q})$ (Matusita, 1967) for $\Phi(u) = (1 - \sqrt{u})^2$. If

$$\Phi(u) = \frac{(u^{\lambda+1} - u - \lambda(u-1))}{\lambda(\lambda+1)}, \lambda \neq 0, -1$$

and $\alpha = 0$, the $(\Phi, \alpha)-$power divergence family reduces to the discrete version of Cressie and Read power divergence (Cressie & Read, 1984), see (1.3), defined as

$$d_{\mathrm{CR}}^\lambda(\mathbf{p}, \mathbf{q}) = \frac{1}{\lambda(\lambda+1)} \sum_{i=1}^m p_i \left[\left(\frac{p_i}{q_i}\right)^\lambda - 1\right]; -\infty < \lambda < \infty,\ \lambda \notin \{0, -1\}.$$

We note that for $\lambda \to 0$ the KL$-$divergence is obtained and for $\lambda = 1$ the Pearson's chi$-$squared divergence is obtained.

Let $\mathbf{X} = (X_1, \ldots, X_m)^\top$ be a random vector from the multinomial distribution, $\mathbf{X} = (X_1, \ldots, X_m)^\top \sim M(N, \mathbf{p}(\boldsymbol{\theta}_0))$, where $\mathbf{p}(\boldsymbol{\theta}_0) = (p_1(\boldsymbol{\theta}_0), \ldots, p_m(\boldsymbol{\theta}_0))^\top$ is the vector of true probabilities and N is the number of the experiments. The vector $\mathbf{p}(\boldsymbol{\theta}_0)$ is assumed to be unknown but belonging to a known family

$$\mathcal{P} = \left\{\mathbf{p}(\boldsymbol{\theta}) = (p_1(\boldsymbol{\theta}), \ldots, p_m(\boldsymbol{\theta}))^\top : \boldsymbol{\theta} = (\theta_1, \ldots, \theta_s) \in \Theta\right\}$$

with $\Theta \subseteq \mathbb{R}^s$ $(s < m-1)$. For a realization $(x_1, \ldots, x_m)^\top$ of $\mathbf{X} = (X_1, \ldots, X_m)^\top$ we have $N = \sum_{i=1}^m x_i$, $x_i \geq 0$ and $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_m)^\top$ with $\hat{p}_i = x_i/N$ be the nonparametric maximum likelihood estimator of the true probabilities $p_i(\boldsymbol{\theta}_0), i = 1, \ldots, m$.

In order to estimate the unknown parameter $\boldsymbol{\theta}$ one could use the maximum likelihood method, where it suffices to find a value $\hat{\boldsymbol{\theta}}$ over Θ that maximizes the quantity, $\sum_{i=1}^m \hat{p}_i \log p_i(\boldsymbol{\theta})$. Then $\hat{\boldsymbol{\theta}}$ is the point maximum likelihood estimator (MLE) of the unknown parameter $\boldsymbol{\theta}$. Equivalently, the MLE can be defined as $\hat{\boldsymbol{\theta}} = \arg\inf_{\boldsymbol{\theta}\in\Theta}$ $d_{\mathrm{KL}}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$. Cressie & Read (1984) proposed the minimum power divergence estimator $\hat{\boldsymbol{\theta}}_\lambda = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_{\mathrm{CR}}^\lambda(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$ and studied its properties. As they argue (Read & Cressie, 1988, p.30) this estimator is a natural generalization of the MLE. Furthermore, Morales et al. (1995) considered the more general minimum $\varphi-$divergence estimator, which is defined as $\hat{\boldsymbol{\theta}}_\varphi = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_\varphi(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$ and Pardo et al. (2002) the restricted minimum $\varphi-$divergence estimator, defined as $\hat{\boldsymbol{\theta}}_\varphi^r = \arg\inf_{\{\boldsymbol{\theta}\in\Theta:\, f_k(\boldsymbol{\theta})=0, k=1,\ldots,\nu\}}$ $d_\varphi(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$ where the parameter $\boldsymbol{\theta}$ is constrained by the functions $f_1(\boldsymbol{\theta}) = 0, \ldots, f_\nu(\boldsymbol{\theta}) = 0$.

In this thesis for testing the composite null hypothesis

$$H_0 \colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0),\ \boldsymbol{\theta}_0 \in \Theta \subseteq \mathbb{R}^s \tag{2.4}$$

we propose the new double index $(\Phi, \alpha)-$test statistic (DITS) which is associated with the (Φ, α)-power divergence family and is given by

$$T_{\Phi_1}^{\alpha_1}\left(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}\right) = \frac{2N}{\Phi_1''(1)} d_{\Phi_1}^{\alpha_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})\right) \tag{2.5}$$

where $\Phi_1, \Phi_2 \in F$ and $\hat{\boldsymbol{\theta}}_\Phi^\alpha$ is the minimum $(\Phi, \alpha)-$power divergence estimator which is going to be defined in the sequel. This general family of test statistics involves two indices and two functions that are not necessarily equal and this feature provides greater flexibility in terms of statistical inference as it will be evident in the following sections. The motivation of this proposal comes from the fact that often the asymptotic distribution fails to represent fully and accurately the underlying distribution of the test statistic.

The results show that the double index family, with proper choices of the two indices, succeeds in ensuring for both the equiprobable as well as skewed models, a very high accuracy of the asymptotic distribution with, at the same time, a Type I error as close as possible to the nominal level.

Some important properties of the recently proposed $(\Phi, \alpha)-$power divergence measure have been studied by Vonta et al. (2012) and the development of a test statistic, which is a special case of (2.5), for goodness−of−fit tests under the simple null hypothesis is the main subject of Mattheou & Karagrigoriou (2010).

In this chapter we consider the $(\Phi,\alpha)-$power divergence family for the purpose of estimation and hypothesis testing and in Section 2.2 we define and study the asymptotic properties of the minimum $(\Phi,\alpha)-$power divergence estimator $\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha}$, which can be seen as a more general class of estimators. In Section 2.3 we investigate the implementation of the proposed methodology in goodness$-$of$-$fit tests under the composite null hypothesis as well as under the contiguous alternatives. Simulations results are presented in Section 2.4 and in Section 2.5 we sum up with some conclusions.

2.2 Minimum $(\Phi,\alpha)-$power divergence estimator

In the sequel, following an approach similar to Cox (Cox, 1984) we provide the form of the estimator $\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha}$ and establish in Theorem 2.3 its asymptotic distribution.

Definition 2.1. *Let* $\mathbf{X} = (X_1,\dots,X_m)^\top$ *be a random vector from the multinomial distribution* $\mathbf{X} \sim M(N,\mathbf{p}(\boldsymbol{\theta}_0))$. *The minimum* $(\Phi,\alpha)-$*power divergence estimator of* $\boldsymbol{\theta}_0$ *is any* $\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha} \in \Theta \subseteq \mathbb{R}^s$ *with* $\Phi \in F$ *which satisfies the condition*

$$\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_{\Phi}^{\alpha}(\hat{\mathbf{p}},\mathbf{p}(\boldsymbol{\theta})). \tag{2.6}$$

We continue by assuming that the model is correct, so that $\boldsymbol{\pi} = \mathbf{p}(\boldsymbol{\theta}_0)$ and the unknown parameter $\boldsymbol{\theta}_0$ satisfies the regularity conditions introduced by Birch (1964). These regularity conditions are the following:

1. The point $\boldsymbol{\theta}_0$ is an interior point of Θ.
2. $\pi_i = p_i(\boldsymbol{\theta}_0) > 0$ for $i = 1,\dots,m$.
3. The mapping $\mathbf{p}(\boldsymbol{\theta})\colon \Theta \to \mathcal{P}$ is totally differentiable at $\boldsymbol{\theta}_0$ so that the partial derivatives of $p_i(\boldsymbol{\theta}_0)$ with respect to each θ_j exist at $\boldsymbol{\theta}_0$ and $\mathbf{p}(\boldsymbol{\theta})$ has a linear approximation at $\boldsymbol{\theta}_0$ given by
$$p_i(\boldsymbol{\theta}) = p_i(\boldsymbol{\theta}_0) + \sum_{j=1}^{s}(\theta_j - \theta_{0j})\frac{\partial p_i(\boldsymbol{\theta}_0)}{\partial\theta_j} + o(\|\boldsymbol{\theta}-\boldsymbol{\theta}_0\|),\ i = 1,\dots,m$$
as $\boldsymbol{\theta} \to \boldsymbol{\theta}_0$.
4. The Jacobian matrix
$$\mathbf{J}(\boldsymbol{\theta}_0) = \left(\frac{\partial\mathbf{p}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = \left(\frac{\partial p_i(\boldsymbol{\theta}_0)}{\partial\theta_j}\right)_{\substack{i=1,\dots,m\\ j=1,\dots,s}}$$
is of full rank.
5. The mapping inverse to $\boldsymbol{\theta} \to \mathbf{p}(\boldsymbol{\theta})$ exists and is continuous at $\boldsymbol{\theta}_0$.
6. The mapping $\mathbf{p}\colon \Theta \to \mathcal{P}$ is continuous at every point $\boldsymbol{\theta} \in \Theta$.

Note that, generalized results under misspecified models, i.e. when it does not exist a value $\boldsymbol{\theta}_0$ which leads to the true probabilities of the multinomial model, are given in Chapter 4. In order to obtain a decomposition of $\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha}$ we are going to use the Implicit Function Theorem in the form presented below (see e.g. Cox (1984)).

Theorem 2.1 (Implicit Function Theorem)**.** *Let* $\mathbf{F}\colon \mathbb{R}^{m+s} \to \mathbb{R}^s$ *be continuously differentiable in an open set* $D \subset \mathbb{R}^{m+s}$ *containing the point* $(\mathbf{x}_0 = (x_{01},\ldots,x_{0m})^\top, \mathbf{y}_0 = (y_{01},\ldots,y_{0s})^\top)$ *for which* $\mathbf{F}(\mathbf{x}_0,\mathbf{y}_0) = 0$*. Suppose that the matrix of partial derivatives* $\partial\mathbf{F}/\partial\mathbf{y}$ *is nonsingular at* $(\mathbf{x}_0,\mathbf{y}_0)$*. Then there exists a neighbourhood* U *of* $(\mathbf{x}_0,\mathbf{y}_0)$ *such that the matrix* $\partial\mathbf{F}/\partial\mathbf{y}$ *is nonsingular for all* $(\mathbf{x},\mathbf{y}) \in U$*, an open set* $A \subset \mathbb{R}^m$ *containing* $\mathbf{x}_0$ *and a unique continuously differentiable function* $\mathbf{g} = (g_1,\ldots,g_s)\colon A \to \mathbb{R}^s$ *such that* $\{(\mathbf{x},\mathbf{y}) \in U\colon \mathbf{F}(\mathbf{x},\mathbf{y}) = 0\} = \{(\mathbf{x},\mathbf{g}(\mathbf{x}))\colon \mathbf{x} \in A\}$*.*

Theorem 2.2. *Under the Birch regularity conditions and assuming that the function* $\mathbf{p}(\boldsymbol{\theta})$ *has continuous second partial derivatives in a neighbourhood of* $\boldsymbol{\theta}_0$*, it holds*

$$\begin{aligned}\hat{\boldsymbol{\theta}}_\Phi^\alpha = \boldsymbol{\theta}_0 + \left(\mathbf{B}(\boldsymbol{\theta}_0)^\top\mathbf{B}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2}) diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0))\\ + o(\|\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\|)\end{aligned} \tag{2.7}$$

where $\hat{\boldsymbol{\theta}}_\Phi^\alpha$ *is unique in a neighbourhood of* $\boldsymbol{\theta}_0$ *and*

$$\mathbf{B}(\boldsymbol{\theta}_0) = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2})\mathbf{A}(\boldsymbol{\theta}_0),$$

while the last element of $\mathbf{B}(\boldsymbol{\theta}_\mathbf{0})$ *is given by*

$$\mathbf{A}(\boldsymbol{\theta}_0) = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})\mathbf{J}(\boldsymbol{\theta}_0).$$

Proof. Following the typical approach for similar results in the literature, let l^m be the interior of the unit $m-$dimensional cube. Let V be a neighbourhood of $\boldsymbol{\theta}_0$ on which $\mathbf{p}\colon \Theta \to \mathcal{P} \subset l^m$ has continuous second partial derivatives. Let

$$\mathbf{F} = (F_1,\ldots,F_s)\colon l^m \times V \to \mathbb{R}^s$$

with

$$F_j(p_1,\ldots,p_m,\theta_1,\ldots,\theta_s) = \frac{\partial d_\Phi^\alpha(\mathbf{p},\mathbf{p}(\boldsymbol{\theta}))}{\partial\theta_j},\ j = 1,\ldots,s.$$

It holds that

$$F_j(\pi_1,\ldots,\pi_m,\theta_{01},\ldots,\theta_{0s}) = 0,\ j = 1,\ldots,s$$

and the matrix

$$\frac{\partial\mathbf{F}}{\partial\boldsymbol{\theta}} = \left(\frac{\partial F_j}{\partial\theta_k}\right)_{\substack{j=1,\ldots,s\\k=1,\ldots,s}} = \Phi''(1)\mathbf{B}(\boldsymbol{\theta})^\top\mathbf{B}(\boldsymbol{\theta})$$

is nonsingular and positive definite at $(\mathbf{p},\boldsymbol{\theta}) = (\boldsymbol{\pi},\boldsymbol{\theta}_0)$.

Applying the Implicit Function Theorem there exists a neighbourhood U of $(\boldsymbol{\pi},\boldsymbol{\theta}_0)$ such that the matrix $\partial\mathbf{F}/\partial\boldsymbol{\theta}$ is nonsingular and a unique differentiable function $\boldsymbol{\theta}^*$: $A \subset l^m \to \mathbb{R}^s$ such that $\boldsymbol{\pi} \in A$ and $\{(\mathbf{p},\boldsymbol{\theta}) \in U\colon \mathbf{F}(\mathbf{p},\boldsymbol{\theta}) = 0\} = \{(\mathbf{p},\boldsymbol{\theta}^*(\mathbf{p}))\colon \mathbf{p} \in A\}$ and $\boldsymbol{\theta}^*(\boldsymbol{\pi}) = \boldsymbol{\theta}_0$. By the chain rule and for $\mathbf{p} = \boldsymbol{\pi}$ we get

$$\frac{\partial\mathbf{F}}{\partial\boldsymbol{\pi}} + \frac{\partial\mathbf{F}}{\partial\boldsymbol{\theta}_0}\frac{\partial\boldsymbol{\theta}_0}{\partial\boldsymbol{\pi}} = 0.$$

Then

$$\mathbf{W}(\boldsymbol{\theta}_0) \equiv \frac{\partial\boldsymbol{\theta}_0}{\partial\boldsymbol{\pi}} = \left(\mathbf{B}(\boldsymbol{\theta}_0)^\top\mathbf{B}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2}) diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}) \tag{2.8}$$

since

$$\frac{\partial \mathbf{F}}{\partial \boldsymbol{\pi}} = -\Phi''(1)\mathbf{A}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha}).$$

The Taylor expansion of $\boldsymbol{\theta}^*(\mathbf{p})$ around $\boldsymbol{\pi}$ along with (2.8) and for $\boldsymbol{\theta}^*(\boldsymbol{\pi}) = \boldsymbol{\theta}_0$ gives

$$\begin{aligned}\boldsymbol{\theta}^*(\mathbf{p}) = \boldsymbol{\theta}_0 + \Big(\mathbf{B}(\boldsymbol{\theta}_0)^\top\mathbf{B}(\boldsymbol{\theta}_0)\Big)^{-1}\mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2})diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\mathbf{p}-\mathbf{p}(\boldsymbol{\theta}_0))\\ + o(\|\mathbf{p}-\mathbf{p}(\boldsymbol{\theta}_0)\|).\end{aligned}$$

Since $\hat{\mathbf{p}} \xrightarrow{p} \boldsymbol{\pi}$ eventually $\hat{\mathbf{p}} \in A$ and then $\boldsymbol{\theta}^*(\hat{\mathbf{p}})$ is the unique solution of the system of equations

$$\frac{\partial d_{\Phi}^{\alpha}\left(\hat{\mathbf{p}}, \mathbf{p}\left(\boldsymbol{\theta}^*(\hat{\mathbf{p}})\right)\right)}{\partial \theta_j} = 0,\ j = 1, \ldots, s$$

as well $(\hat{\mathbf{p}}, \boldsymbol{\theta}^*(\hat{\mathbf{p}})) \in U$, from where we can conclude that $\boldsymbol{\theta}^*(\hat{\mathbf{p}})$ is the minimum $(\Phi, \alpha)-$power divergence estimator $\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha}$ given in (2.7). ☐

The asymptotic distribution of the minimum $(\Phi, \alpha)-$power divergence estimator is established in the theorem below.

Theorem 2.3. *Under the assumptions of Theorem 2.2, it holds*

$$\sqrt{N}(\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha} - \boldsymbol{\theta}_0) \xrightarrow[N\to\infty]{L} \mathrm{N}_s(\mathbf{0}, \mathbf{W}(\boldsymbol{\theta}_0)\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}\mathbf{W}(\boldsymbol{\theta}_0)^\top)$$

where $\mathbf{W}(\boldsymbol{\theta}_0)$ *is given in (2.8),* $\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)} = diag(\mathbf{p}(\boldsymbol{\theta}_0)) - \mathbf{p}(\boldsymbol{\theta}_0)\mathbf{p}(\boldsymbol{\theta}_0)^\top$ *and* $\mathrm{N}(\cdot,\cdot)$ *is the* $s-$*dimensional Normal distribution.*

Proof. The result follows from Theorem 2.2 and the fact that by the Central Limit Theorem we have that:

$$\sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}). \tag{2.9}$$

☐

Remark 2.1. *It should be noted that in general, both the decomposition of the minimum* $(\Phi, \alpha)-$*power divergence estimator given in (2.7) and the asymptotic variance−covariance matrix of* $\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha}$ *given in Theorem 2.3, depend on the index* α. *At the same time, notice that under the equiprobable model,* $diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha}) = m^{-\alpha}I_{m\times m}$, *while for* $\alpha = 0$, $diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha}) = I_{m\times m}$. *Thus, for these two special settings, the decomposition of the minimum* $(\Phi, \alpha)-$*power divergence estimator takes a simplified form that does not involve the index* α*, namely*

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\Phi} = \boldsymbol{\theta}_0 + \Big(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0)\Big)^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\hat{\mathbf{p}}-\mathbf{p}(\boldsymbol{\theta}_0))\\ + o(\|\hat{\mathbf{p}}-\mathbf{p}(\boldsymbol{\theta}_0)\|).\end{aligned} \tag{2.10}$$

Note that this simplified form coincides with the decomposition of the minimum $\varphi-$*divergence estimator proposed in Morales et al. (1995) and is given by* $\hat{\boldsymbol{\theta}}_{\varphi} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d_{\varphi}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$. *This estimator can be derived from the minimum* $(\Phi, \alpha)-$*power divergence estimator if we choose* $\alpha = 0$ *and* $\Phi = \varphi$. *Hence, the minimum* $(\Phi, \alpha)-$*power divergence estimators with index* α*, can be thought of forming a general family of estimators which includes as special cases, the estimators associated with the two settings that do not involve* α*, namely, the* $\varphi-$*divergence of Morales et al. (1995) and the equiprobable model.*

Remark 2.2 (Best Asymptotically Normal (BAN) Estimators). *An estimator that satisfies the following asymptotic expansion*

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_0 + \mathbf{C}(\boldsymbol{\theta}_0) diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)) + o_p(N^{-1/2}),$$

with $\mathbf{C}(\boldsymbol{\theta}_0) = \left(\mathbf{A}(\boldsymbol{\theta}_0)^\top \mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1} \mathbf{A}(\boldsymbol{\theta}_0)^\top$ *and* $\mathbf{A}(\boldsymbol{\theta}_0) = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})\mathbf{J}(\boldsymbol{\theta}_0)$, *is called best asymptotically normal (BAN) (see for instance Cressie & Read (1984)). BAN estimators have three important properties: (a) They are consistent, i.e. as* $N \to \infty$ *the estimator converges to the true value of the estimated parameter; (b) They are asymptotically normally distributed; (c) They are asymptotically efficient, i.e. as* $N \to \infty$ *no other estimator has smaller variance.*

Remark 2.3. *It should be also noted that the minimum* $(\Phi, \alpha)-$*power divergence estimator given in (2.7) is not in general, Best Asymptotically Normal (BAN) . However, best asymptotic normality is achieved in the two special cases mentioned in Remark 2.1, namely, when* $\alpha = 0$ *and when the true model* $\mathbf{p}(\boldsymbol{\theta}_0)$ *is the equiprobable. Indeed, in both these cases, the decomposition of the minimum* $(\Phi, \alpha)-$*power divergence estimator takes the simplified form given in (2.10) with the variance−covariance matrix of Theorem 2.3 reduced to*

$$\left(\mathbf{A}(\boldsymbol{\theta}_0)^\top \mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1} \tag{2.11}$$

which is the Fisher information matrix associated with the multinomial model. Note further that for values of α *close to zero (as is the case in most applications) the minimum* $(\Phi, \alpha)-$*power divergence estimator and the variance−covariance matrix mentioned above are approximately equal to (2.10) and (2.11), respectively, and therefore the resulting estimators are approximately BAN. In other words, irrespectively of the underlying modelling mechanism (equiprobable or not), at least approximately BAN estimators are ensured provided that sufficiently small values of the index* α *are chosen. This feature provides the statistician with the flexibility to choose between classical BAN estimators (for the case of equiprobable model or when* $\alpha = 0$*) and approximately BAN estimators (for nonnecessarily equiprobable models) derived through the proposed methodology with index values in the range* $(0, 0.5)$*.*

Remark 2.4. *Note that the present work combines under a single umbrella all measures of divergence and the associated estimators previously investigated. In other words, cases discussed in the past can be considered as special cases of the general family of measures explored in this work. At the same time the double index family as it will be verified by an extensive simulation study (Section 2.4.1) allows, for proper choices of* α_1 *and* α_2*, a high accuracy of the asymptotic distribution of the associated test statistic for both equiprobable and skewed multinomial models.*

2.3 Statistical inference

The main object of this section is to provide the asymptotic distribution of the double index $(\Phi, \alpha)-$test statistic (2.5) under the composite null hypothesis (2.4). In order to do so, we will prove first the asymptotic distribution of the test statistic

$$T_\varphi\left(\hat{\boldsymbol{\theta}}_\Phi^\alpha\right) = \frac{2N}{\varphi''(1)} d_\varphi\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}_\Phi^\alpha)\right) \tag{2.12}$$

where $\hat{\boldsymbol{\theta}}_\Phi^\alpha$ is the minimum $(\Phi, \alpha)-$power divergence estimator (M(Φ, α)PDE) and φ as in (2.3). Furthermore, for establishing the results the following Lemma is required.

Lemma 2.4. *(Ferguson, 1996, p.57). Suppose that* $\mathbf{X} = (X_1, \ldots, X_m)^\top \sim \mathrm{N_m}(\mathbf{0}, \mathbf{\Sigma})$ *then* $\mathbf{X}^\top \mathbf{X} \sim \chi^2_m$*, if and only if* $\mathbf{\Sigma}$ *is a projection matrix of rank m, where* χ^2_m *is the chi−squared distribution with m degrees of freedom.*

Theorem 2.5. *Under the assumptions of Theorem 2.2, the composite null hypothesis (2.4) and assuming further that the true model* $\mathbf{p}(\boldsymbol{\theta}_0)$ *is the equiprobable, we have*

$$T_\varphi\left(\hat{\boldsymbol{\theta}}^\alpha_\Phi\right) \xrightarrow[N\to\infty]{L} \chi^2_{m-1-s}\,.$$

Proof. From Birch's third regularity condition, we have that

$$\mathbf{p}(\hat{\boldsymbol{\theta}}^\alpha_\Phi) = \mathbf{p}(\boldsymbol{\theta}_0) + \mathbf{J}(\boldsymbol{\theta}_0)(\hat{\boldsymbol{\theta}}^\alpha_\Phi - \boldsymbol{\theta}_0) + o(\|\hat{\boldsymbol{\theta}}^\alpha_\Phi - \boldsymbol{\theta}_0\|).$$

By Theorem 2.3 and relation (2.9) we have that

$$\mathbf{p}(\hat{\boldsymbol{\theta}}^\alpha_\Phi) - \mathbf{p}(\boldsymbol{\theta}_0) = \mathbf{M}(\boldsymbol{\theta}_0)(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)) + o_p(N^{-1/2}),$$

where

$$\mathbf{M}(\boldsymbol{\theta}_0) = \mathbf{J}(\boldsymbol{\theta}_0)\mathbf{W}(\boldsymbol{\theta}_0) \tag{2.13}$$

and

$$\sqrt{N}(\mathbf{p}(\hat{\boldsymbol{\theta}}^\alpha_\Phi) - \mathbf{p}(\boldsymbol{\theta}_0)) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{0}, \mathbf{M}(\boldsymbol{\theta}_0)\mathbf{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}\mathbf{M}(\boldsymbol{\theta}_0)^\top). \tag{2.14}$$

Then,

$$\sqrt{N}\begin{pmatrix} \hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0) \\ \mathbf{p}(\hat{\boldsymbol{\theta}}^\alpha_\Phi) - \mathbf{p}(\boldsymbol{\theta}_0) \end{pmatrix} \xrightarrow[N\to\infty]{L} \mathrm{N}\left(\mathbf{0}, \begin{pmatrix} I \\ \mathbf{M}(\boldsymbol{\theta}_0) \end{pmatrix} \mathbf{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}(I, \mathbf{M}(\boldsymbol{\theta}_0)^\top)\right)$$

and

$$\sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}(\hat{\boldsymbol{\theta}}^\alpha_\Phi)) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{0}, \mathbf{L}(\boldsymbol{\theta}_0))$$

with

$$\mathbf{L}(\boldsymbol{\theta}_0) = \mathbf{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)} - \mathbf{M}(\boldsymbol{\theta}_0)\mathbf{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)} - \mathbf{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}\mathbf{M}(\boldsymbol{\theta}_0)^\top + \mathbf{M}(\boldsymbol{\theta}_0)\mathbf{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}\mathbf{M}(\boldsymbol{\theta}_0)^\top. \tag{2.15}$$

A second order Taylor expansion of $d_\varphi\left(\mathbf{p}, \mathbf{q}\right)$ around $(\mathbf{p}(\boldsymbol{\theta}_0), \mathbf{p}(\boldsymbol{\theta}_0))$ yields

$$\frac{2N}{\varphi''(1)} d_\varphi\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^\alpha_\Phi)\right) = \sum_{i=1}^m \frac{N}{p_i(\boldsymbol{\theta}_0)}\left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}^\alpha_\Phi)\right)^2 + o_p(1) = \mathbf{X}^\top\mathbf{X} + o_p(1).$$

where

$$\mathbf{X} = \sqrt{N} diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\hat{\mathbf{p}} - \mathbf{p}(\hat{\boldsymbol{\theta}}^\alpha_\Phi)) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{0}, \mathbf{T}(\boldsymbol{\theta}_0)).$$

Under the assumption of the equiprobable true model, $\mathbf{T}(\boldsymbol{\theta}_0)$ obtained via (2.15) is a projection matrix with rank

$$\begin{aligned} rank(\mathbf{T}(\boldsymbol{\theta}_0)) &= m - 1 - trace(diag(\mathbf{p}(\boldsymbol{\theta}_0)^\alpha)\mathbf{A}(\boldsymbol{\theta}_0)\left(\mathbf{B}(\boldsymbol{\theta}_0)^\top\mathbf{B}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^\top) \\ &= m - 1 - s. \end{aligned}$$

Then the result follows from Lemma 2.4. □

Remark 2.5. *When* $\alpha = 0$ *the result of Theorem 2.5 holds true for every model of true probabilities* $\mathbf{p}(\boldsymbol{\theta}_0)$ *and not only for the equiprobable one. In that sense when* $\alpha \to 0$ *the* *above result holds approximately true and indeed, as it will be evident in the simulation studies of Section 2.4.1, the asymptotic distribution of the test statistic (2.12) for values of* α *close to zero (e.g.* $\alpha = 0.1$*) is practically a chi−squared distribution, even when the true model is far from the equiprobable.*

2.3.1 Asymptotic distribution of the double index $(\Phi, \alpha)-$test statistic

Now we proceed to investigate the asymptotic distribution of the test statistic (2.5).

Theorem 2.6. *Under the assumptions of Theorem 2.5 we have*

$$T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}) \xrightarrow[N\to\infty]{L} c\,\chi^2_{m-1-s}$$

where $c = m^{-\alpha_1}$.

Proof. A second order Taylor expansion of $d_{\Phi_1}^{\alpha_1}(\mathbf{p}, \mathbf{q})$ around $(\mathbf{p}(\boldsymbol{\theta}_0), \mathbf{p}(\boldsymbol{\theta}_0))$ is given by

$$\begin{aligned} d_{\Phi_1}^{\alpha_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})\right) =& \frac{\Phi_1''(1)}{2}\sum_{i=1}^{m}\frac{p_i(\boldsymbol{\theta}_0)^{\alpha_1}}{p_i(\boldsymbol{\theta}_0)}\left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})\right)^2 + o(\|\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\|^2) \\ &+ o(\|\mathbf{p}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}) - \mathbf{p}(\boldsymbol{\theta}_0)\|^2). \end{aligned}$$

By (2.9) and (2.14) we have that $No(\|\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\|^2) = o_p(1)$ and $No(\|\mathbf{p}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}) - \mathbf{p}(\boldsymbol{\theta}_0)\|^2) = o_p(1)$, so that

$$T_{\Phi_1}^{\alpha_1}\left(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}\right) = \sum_{i=1}^{m}\frac{Np_i(\boldsymbol{\theta}_0)^{\alpha_1}}{p_i(\boldsymbol{\theta}_0)}\left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})\right)^2 + o_p(1).$$

The result follows from Theorem 2.5. □

Remark 2.6. *When the true model is not the equiprobable but* $\alpha_2 = 0$ *or the value of* α_2 *is close to zero, we can reformulate Theorem 2 of Mattheou & Karagrigoriou (2010) and set* $Y = \sum_{i=1}^m \frac{N}{p_i(\boldsymbol{\theta}_0)}(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}))^2$ *in order to have*

$$(\min_i p_i(\boldsymbol{\theta}_0)^{\alpha_1})Y \prec_{st} \sum_{i=1}^{m}\frac{Np_i(\boldsymbol{\theta}_0)^{\alpha_1}}{p_i(\boldsymbol{\theta}_0)}(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}))^2 \prec_{st} (\max_i p_i(\boldsymbol{\theta}_0)^{\alpha_1})Y,$$

where $\prec_{st}$ *is the symbol for usual stochastic order. Then the asymptotic distribution of the test statistic* $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})$ *is estimated to be approximately* $c\chi^2_{m-1-s}$*, where*

$$c = 0.5(\min_i p_i(\boldsymbol{\theta}_0)^{\alpha_1} + \max_i p_i(\boldsymbol{\theta}_0)^{\alpha_1}). \tag{2.16}$$

It is easy to see that when the true model is the equiprobable, the value of c in (2.16) coincides with the one given in the statement of Theorem 2.6. Note that, for the real r.v. X and Y the term $X \prec_{st} Y$ *is equivalent to* $Pr(X > x) \leq Pr(Y > x)$, $x \in \mathbb{R}$.

Remark 2.7. *At this point it is important to clarify the role of each of the indices* α_1 *and* α_2*. Index* α_1 *affects the distribution of the test statistic. Thus, for values of* α_1 *close to zero the asymptotic distribution of the test statistic (2.5), under the equiprobable model, coincides with the chi−squared distribution (with the appropriate degrees of freedom), while for larger values of* α_1 *the asymptotic distribution of (2.5) coincides with c times the chi−squared distribution.*

On the other hand, when the true model is not the equiprobable, as α_2 deviates from 0 the asymptotic distribution fails to describe properly the underlying distribution of the test statistic (2.5). As it will later be clear from the simulation studies, when the value of α_2 is close to zero then the asymptotic distribution of the test statistic is approximately c times the chi−squared distribution, even when the true model is far from the equiprobable.

Also the simulation studies reveal that for small departures from the equiprobable model, the asymptotic distribution of (2.5) coincides with c times the chi−squared distribution, even for values of α_2 not close to zero.

We close this section by furnishing the power of the test statistic (2.5) under the alternative hypothesis $H_1 \colon \mathbf{p} = \mathbf{q}$. For this purpose, we assume that the joint asymptotic distribution of $\hat{\mathbf{p}}$ and $\mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2})$ is normal, given by

$$\sqrt{N}\left((\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2})) - (\mathbf{q}, \mathbf{p}(\boldsymbol{\theta}_1))\right) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}^*), \tag{2.17}$$

where $\boldsymbol{\theta}_1 = \arg\inf_{\boldsymbol{\theta}\in\Theta} d^{\alpha_2}_{\Phi_2}(\mathbf{q}, \mathbf{p}(\boldsymbol{\theta}))$,

$$\boldsymbol{\Sigma}^* = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \ \boldsymbol{\Sigma}_{11} = diag(\mathbf{q}) - \mathbf{q}\mathbf{q}^\top and\ \boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}.$$

We note that in this case $\mathbf{p}(\boldsymbol{\theta}_1) \neq \mathbf{q}$ and $\hat{\mathbf{p}} \xrightarrow{p} \mathbf{q}$ as well as $\mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2}) \xrightarrow{p} \mathbf{p}(\boldsymbol{\theta}_1)$ (see e.g. Pardo (2006, p.266)).

Theorem 2.7. *Under the alternative hypothesis $H_1 \colon \mathbf{p} = \mathbf{q}$ and the assumption (2.17), the asymptotic power of the test given in (2.5) is equal to*

$$\gamma_{\alpha_1} = Pr\left(Z \geq (2\sqrt{N}\sigma_{\alpha_1})^{-1}(\Phi_1''(1)c\chi^2_{m-1-s,a} - 2Nd^{\alpha_1}_{\Phi_1}(\mathbf{q}, \mathbf{p}(\boldsymbol{\theta}_1)))\right)$$

where Z is the standard normal random variable, $\chi^2_{m-1-s,1-a}$ is the $(1-a)$ percentile of the χ^2_{m-1-s} distribution, c as in (2.16) and

$$\begin{aligned}\sigma^2_{\alpha_1} = {} & \left(\frac{\partial}{\partial \mathbf{u}} d^{\alpha_1}_{\Phi_1}(\mathbf{u}, \mathbf{p}(\boldsymbol{\theta}_1))\right)_{\mathbf{u}=\mathbf{q}} \boldsymbol{\Sigma}_{11} \left(\frac{\partial}{\partial \mathbf{u}} d^{\alpha_1}_{\Phi_1}(\mathbf{u}, \mathbf{p}(\boldsymbol{\theta}_1))\right)^\top_{\mathbf{u}=\mathbf{q}} \\ & + 2\left(\frac{\partial}{\partial \mathbf{u}} d^{\alpha_1}_{\Phi_1}(\mathbf{u}, \mathbf{p}(\boldsymbol{\theta}_1))\right)_{\mathbf{u}=\mathbf{q}} \boldsymbol{\Sigma}_{12} \left(\frac{\partial}{\partial \mathbf{t}} d^{\alpha_1}_{\Phi_1}(\mathbf{q}, \mathbf{t})\right)^\top_{\mathbf{t}=\mathbf{p}(\boldsymbol{\theta}_1)} \\ & + \left(\frac{\partial}{\partial \mathbf{t}} d^{\alpha_1}_{\Phi_1}(\mathbf{q}, \mathbf{t})\right)_{\mathbf{t}=\mathbf{p}(\boldsymbol{\theta}_1)} \boldsymbol{\Sigma}_{22} \left(\frac{\partial}{\partial \mathbf{t}} d^{\alpha_1}_{\Phi_1}(\mathbf{q}, \mathbf{t})\right)^\top_{\mathbf{t}=\mathbf{p}(\boldsymbol{\theta}_1)}.\end{aligned}$$

Proof. A first order Taylor expansion of the $(\Phi_1, \alpha_1)-$power divergence measure around $(\mathbf{q}, \mathbf{p}(\boldsymbol{\theta}_1))$ is given by

$$\begin{aligned}d^{\alpha_1}_{\Phi_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2})\right) = {} & d^{\alpha_1}_{\Phi_1}(\mathbf{q}, \mathbf{p}(\boldsymbol{\theta}_1)) + \left(\frac{\partial}{\partial \mathbf{u}} d^{\alpha_1}_{\Phi_1}(\mathbf{u}, \mathbf{p}(\boldsymbol{\theta}_1))\right)_{\mathbf{u}=\mathbf{q}} (\hat{\mathbf{p}} - \mathbf{q}) \\ & + \left(\frac{\partial}{\partial \mathbf{t}} d^{\alpha_1}_{\Phi_1}(\mathbf{q}, \mathbf{t})\right)_{\mathbf{t}=\mathbf{p}(\boldsymbol{\theta}_1)} \left(\mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2}) - \mathbf{p}(\boldsymbol{\theta}_1)\right) + o_p(N^{-1/2})\end{aligned}$$

since we have assumed that (2.17) holds under $H_1 \colon \mathbf{p} = \mathbf{q}$. The result follows immediately. $\square$

2.3.2 Asymptotic theory under contiguous alternatives

In this section we provide the asymptotic distribution of the proposed test statistic under a sequence of alternative hypotheses of the form

$$H_{1,N} \colon \mathbf{p} = \mathbf{p}_N = \mathbf{p}(\boldsymbol{\theta}_0) + \frac{\mathbf{d}}{\sqrt{N}} \tag{2.18}$$

which is known as the contiguous alternative (Neyman, 1937; McManus, 1991) to the null hypothesis $H_0 \colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0)$, $\boldsymbol{\theta}_0 \in \Theta \subseteq \mathbb{R}^s$ unknown, where $\mathbf{d} = (d_1, \ldots, d_m)^\top$ is a fixed vector such that $\sum_{i=1}^m d_i = 0$. The following lemma is required in order to establish in Theorem 2.9 that the asymptotic distribution of the test statistic (2.5), under the contiguous alternative (2.18), is a non−central chi−squared distribution.

Lemma 2.8. *(Ferguson, 1996, p.63) Let* $\mathbf{X} = (X_1, \ldots, X_m)^\top \sim \mathrm{N}(\boldsymbol{\delta}, \boldsymbol{\Sigma})$*. If* $\boldsymbol{\Sigma}$ *is a projection matrix of rank m and* $\boldsymbol{\Sigma}\boldsymbol{\delta} = \boldsymbol{\delta}$*, then* $\mathbf{X}^\top\mathbf{X} \sim \chi^2_m(\boldsymbol{\delta}^\top\boldsymbol{\delta})$ *where* $\chi^2_m(\boldsymbol{\delta}^\top\boldsymbol{\delta})$ *is the non−central chi−squared distribution with m degrees of freedom and non−centrality parameter* $\boldsymbol{\delta}^\top\boldsymbol{\delta}$*.*

Theorem 2.9. *Under the assumptions of Theorem 2.2, the local contiguous alternative hypothesis (2.18) and assuming further that the true model* $\mathbf{p}(\boldsymbol{\theta}_0)$ *is the equiprobable, the asymptotic distribution of the test statistic given in (2.5) is* $c\chi^2_{m-1-s}(\boldsymbol{\xi}^\top\boldsymbol{\xi})$ *with* $\boldsymbol{\xi} = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})\,(I - \mathbf{M}(\boldsymbol{\theta}_0))\,\mathbf{d}$*,* $\mathbf{M}(\boldsymbol{\theta}_0)$ *as in (2.13) and* $c = m^{-\alpha_1}$*.*

Proof. Under the contiguous alternative hypotheses (2.18) we have

$$\sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}_N) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}) \text{ and } \sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{d}, \boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)})$$

so that

$$\sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2})) \xrightarrow[N\to\infty]{L} \mathrm{N}((I - \mathbf{M}(\boldsymbol{\theta}_0))\mathbf{d}, \mathbf{L}(\boldsymbol{\theta}_0)).$$

Also

$$T_\varphi(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2}) = \sum_{i=1}^m \frac{N}{p_i(\boldsymbol{\theta}_0)} \left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2})\right)^2 + o_p(1) = \mathbf{X}^\top\mathbf{X} + o_p(1),$$

where

$$\mathbf{X} = \sqrt{N} diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})\,(\hat{\mathbf{p}} - \mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2})) \xrightarrow[N\to\infty]{L} \mathrm{N}(\boldsymbol{\xi}, \mathbf{T}(\boldsymbol{\theta}_0)),$$

with $\mathbf{T}(\boldsymbol{\theta}_0)$ as in the proof of Theorem 2.5 and $\boldsymbol{\xi} = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(I - \mathbf{M}(\boldsymbol{\theta}_0))\mathbf{d}$. Then it is easy to see that under the equiprobable model, $\mathbf{T}(\boldsymbol{\theta}_0)\boldsymbol{\xi} = \boldsymbol{\xi}$. Indeed, under the equiprobable model, we have that $(\mathbf{B}(\boldsymbol{\theta}_0)^\top\mathbf{B}(\boldsymbol{\theta}_0))^{-1} = (1/m)^{-1}(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0))^{-1}$ and $\mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha_2/2}) = \mathbf{A}(\boldsymbol{\theta}_0)^\top(1/m)^{\alpha_2}$ thus

$$\boldsymbol{\xi} = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})\mathbf{d} - \mathbf{A}(\boldsymbol{\theta}_0)(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0))^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})\mathbf{d}.$$

The result follows from the fact that $\sqrt{\mathbf{p}(\boldsymbol{\theta}_0)^\top} diag(\mathbf{p}(\boldsymbol{\theta}_0)^{1/2})\mathbf{d} = 0$ and $\sqrt{\mathbf{p}(\boldsymbol{\theta}_0)^\top}$ $\mathbf{A}(\boldsymbol{\theta}_0) = 0$. Then, from Lemma (2.8) we conclude that

$$\sum_{i=1}^m \frac{N}{p_i(\boldsymbol{\theta}_0)} \left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2})\right)^2 \xrightarrow[N\to\infty]{L} \chi^2_{m-1-s}(\boldsymbol{\xi}^\top\boldsymbol{\xi}).$$

Finally, by using the same arguments as in Theorem 2.6, the distribution of the test statistic (2.5) under (2.18) is $c\chi^2_{m-1-s}(\boldsymbol{\xi}^\top\boldsymbol{\xi})$ with $c = m^{-\alpha_1}$. □

Note that although in general, the non−centrality parameter depends on the index α_2, under the equiprobable model, it is independent of α_2 and is given by $\boldsymbol{\xi}^\top\boldsymbol{\xi} = \mathbf{d}^\top diag\big(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}\big)\big(I - \mathbf{A}(\boldsymbol{\theta}_0)\left(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^\top\big)diag\big(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}\big)\mathbf{d}$. Since, under the equiprobable model, the distribution of the above theorem does not depend on the form of the function Φ or the indices α_1 and α_2, the power of the test under the contiguous alternative hypotheses is given by $\Pr(\chi^2_{m-1-s}(\boldsymbol{\xi}^\top\boldsymbol{\xi}) \geq \chi^2_{m-1-s,a})$. In other words, although in general, there is no equivalence in terms of Pitman Asymptotic Relative Efficiency (ARE), when the true model is the equiprobable, the asymptotic power functions for all members of the double index $(\Phi, \alpha)-$ test statistic under the contiguous alternatives coincide, which in turn, implies that they are equivalent in terms of Pitman ARE. Note that, the Pitman ARE concerns two test statistics, which are based on sample sizes N_1 and N_2. Under the condition that the size and the power of these two test statistics are respectively the same, considering every time a new contiguous alternative, the $\lim_{N_1\to\infty}(N_2/N_1)$, under the equiprobable model, is equal to 1. For more details about Pitman ARE see Pitman (1979, Ch.7).

Remark 2.8. *Following the lines of Remark 2.6, when the true model is not the equiprobable but* $\alpha_2 = 0$ *or the value of* α_2 *is close to zero, the asymptotic distribution of the test statistic given in (2.5) under the contiguous alternative hypotheses (2.18) is estimated to be approximately* $c\chi^2_{m-1-s}(\boldsymbol{\xi}^\top\boldsymbol{\xi})$ *with c as in (2.16).*

2.4 Simulation studies

For the simulation studies presented in this section, we used the R software (R Core Team, 2016), while for the optimization the **optimise** function from the base package **stats** is implemented. For the manipulation of the data we used the package **dplyr** (Wickham et al., 2022) and for the visualization of the results the package **ggplot2** (Wickham, 2016). The syntax is provided in Appendices C.1, C.2 and C.3.

2.4.1 Asymptotic results verification

The purpose of this simulation study, is to explore the behaviour and the accuracy of the asymptotic distribution of the test statistics (2.12) and (2.5) not only under the equiprobable model but also for nonnecessarily equiprobable (i.e. skewed) models. Furthermore, the proposed simulations are used to investigate the correctness of the size of the test. In order to study the behaviour and the accuracy of the asymptotic distribution of the test statistics (2.12) and (2.5) we will consider the following model of probabilities:

$$\begin{aligned}\mathbf{p}(\theta) &= (p_1(\theta), p_2(\theta), p_3(\theta), p_4(\theta))^\top \\ &= \left(\frac{\theta}{2}, \theta\left(1 - \frac{\theta}{2}\right) - \frac{1}{8}, \frac{1-\theta}{2}, \frac{(1-\theta)^2}{2} + \frac{1}{8}\right)^\top\end{aligned}$$

with $\theta \in (1 - \sqrt{3}/2, 1)$. Six values of θ are going to be used, namely $\theta = 0.5, 0.6, 0.7, 0.8, 0.9$ and 0.999. This model has been considered due to the fact that it provides the ability to study the behaviour of the proposed test statistic under both equiprobable and not equiprobable models. Observe that for $\theta = 0.5$ we get the equiprobable model and as θ deviates from 0.5 the model deviates from the equiprobable one. Although at first sight, one may get the impression that the results of Theorems 2.5 and 2.6 are valid only under the equiprobable model, this simulation study reveals that these results are still valid for a wide range of multinomial models by choosing properly the values of the indices α_1 and α_2. The multinomial model used in this simulation study allows the examination of the behaviour of the test statistic not only for the equiprobable model ($\theta = 0.5$) but also for skewed models that deviate from the equiprobable one (right skewed for $\theta \to 1$ and left skewed for $\theta \to 1 - \sqrt{3}/2$).

For the comparison between the asymptotic cumulative distribution function (CDF) of Theorems 2.5 and 2.6 and the empirical cumulative distribution function (ECDF) of (2.12) and (2.5), we generated 10000 samples from each multinomial model with sample size of 10000 observations. Test statistic (2.12) is applied for $\varphi(u) = 1/2(u-1)^2$ and Φ as in (2.2), while the proposed double index $(\Phi, \alpha)-$test statistic is applied for Φ_1 and Φ_2 as in (2.2).

The ECDF of the test statistic (2.12) is presented in Figures 2.1 and 2.2 for two values of the index α (0.1 and 5), together with the CDF of the chi$-$squared distribution with two degrees of freedom (χ_2^2). As expected, for values of α close to zero (e.g. $\alpha = 0.1$) the distribution of (2.12) coincides with the asymptotic distribution χ_2^2 even when the true model is far from the equiprobable. On the other hand, for larger values of α (e.g. $\alpha = 5$) as the true model deviates from the equiprobable, the asymptotic distribution fails to describe properly the distribution of the test statistic. Also, we note that for small departures from the equiprobable model (e.g. $\theta = 0.6$), the distribution of the test statistic is almost identical to the asymptotic distribution χ_2^2 even for values of α not close to zero (e.g. $\alpha = 5$). The ECDF of the proposed double index $(\Phi, \alpha)-$test statistic given in (2.5) is presented in Figures 2.3 through 2.8 for three different values of the index α_1, namely $\alpha_1 = 0.01,\ 0.5,\ 3$ and two values of $\alpha_2 = 0.1,\ 5$ together with the chi$-$squared distribution and the c times chi$-$squared distribution with c as in (2.16). From the figures, we conclude the following:

- As expected (see Figures 2.3 & 2.4) for $\alpha_1 = 0.01$ the value of c in (2.16) is close to 1 and therefore $c\chi_2^2 \simeq \chi_2^2$.
- For α_1 small (the results for $\alpha_1 = 0.01$ appear in Figures 2.3 & 2.4) the distribution of the test statistic in (2.5)
    - coincides with the asymptotic distribution $c\chi_2^2 \simeq \chi_2^2$ as long as α_2 is close to zero ($\alpha_2 = 0.1$ in Figure 2.3) for every model, and
    - it is satisfactorily close to the asymptotic distribution $c\chi_2^2$ for larger values of α_2 for medium size deviations from the equiprobable model ($0.3 < \theta < 0.8$).
- Almost identical results arise for medium values of α_1 ($\alpha_1 = 0.5$ in Figures 2.5 & 2.6). Indeed the distribution in (2.5) coincides with $c\chi_2^2$
    - for small values of α_2 ($\alpha_2 = 0.1$ in Figure 2.5) for all models considered,
    - as well as for larger values of α_2 ($\alpha_2 = 5$ in Figure 2.6) as long the departure from the equiprobable model is moderate ($0.3 < \theta < 0.8$).
- The same behaviour is observed for large values of α_1 ($\alpha_1 = 3$ in Figures 2.7 & 2.8) especially when α_2 is relatively close to zero ($\alpha_2 = 0.1$ in Figure 2.7).

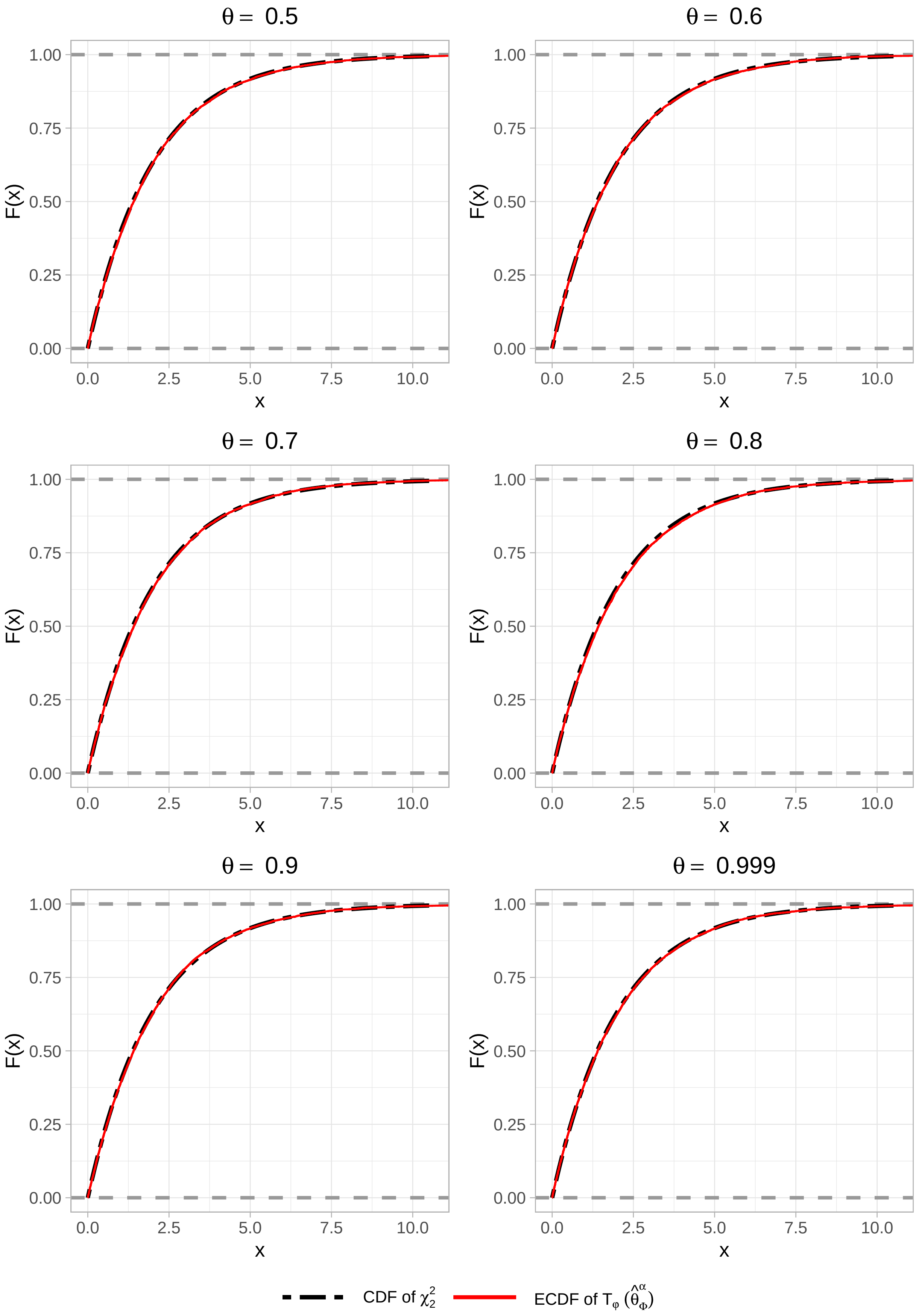


Figure 2.1: Comparison between the ECDF of $T_\varphi\left(\hat{\boldsymbol{\theta}}^\alpha_\Phi\right)$ and the CDF of χ^2_2, $\varphi(u) = (u-1)^2/2$, $\Phi(u)$ as in (2.2), $\alpha = 0.1$.

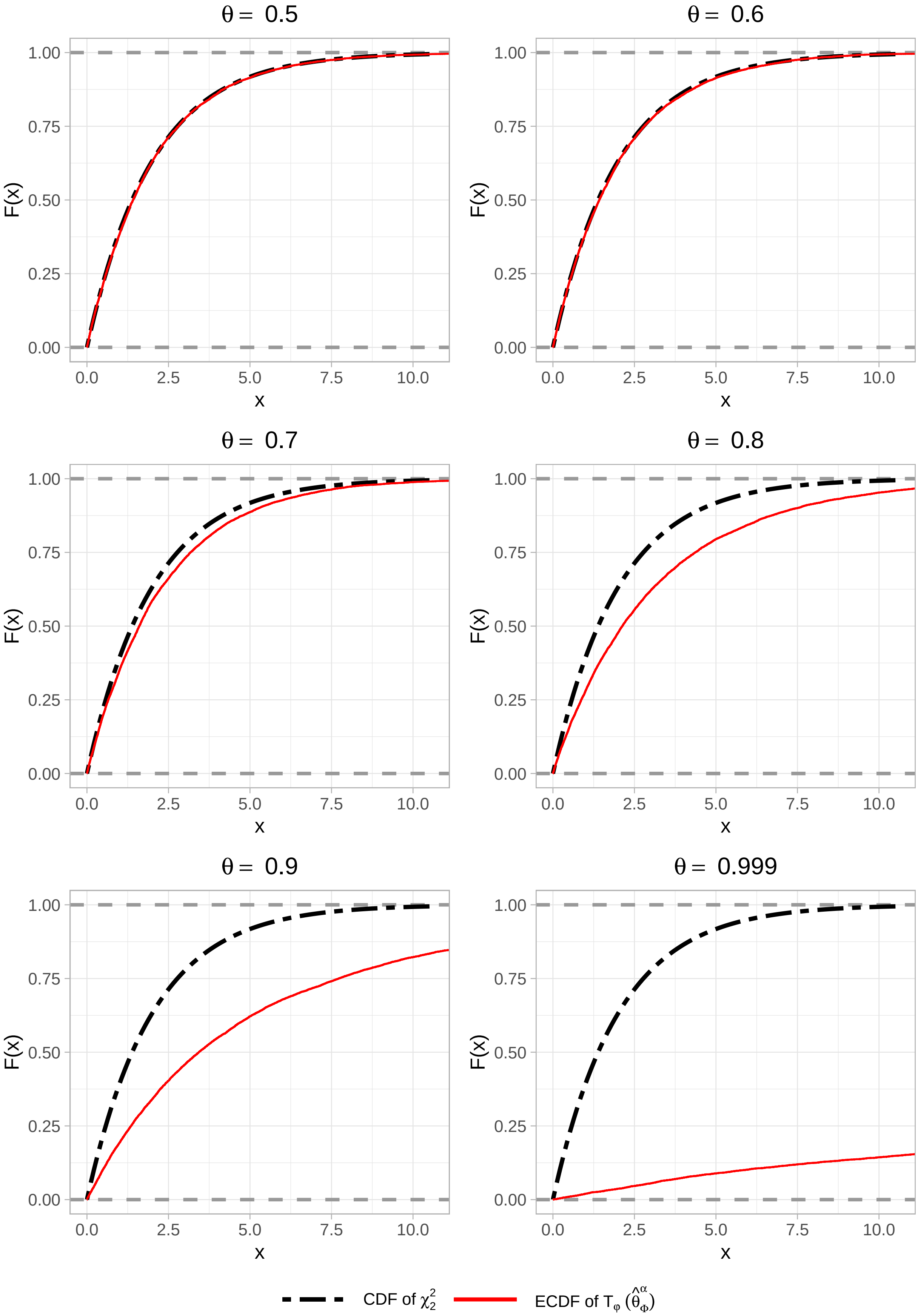


Figure 2.2: Comparison between the ECDF of $T_\varphi\left(\hat{\theta}^\alpha_\Phi\right)$ and the CDF of χ^2_2, $\varphi(u) = (u-1)^2/2$, $\Phi(u)$ as in (2.2), $\alpha = 5$.

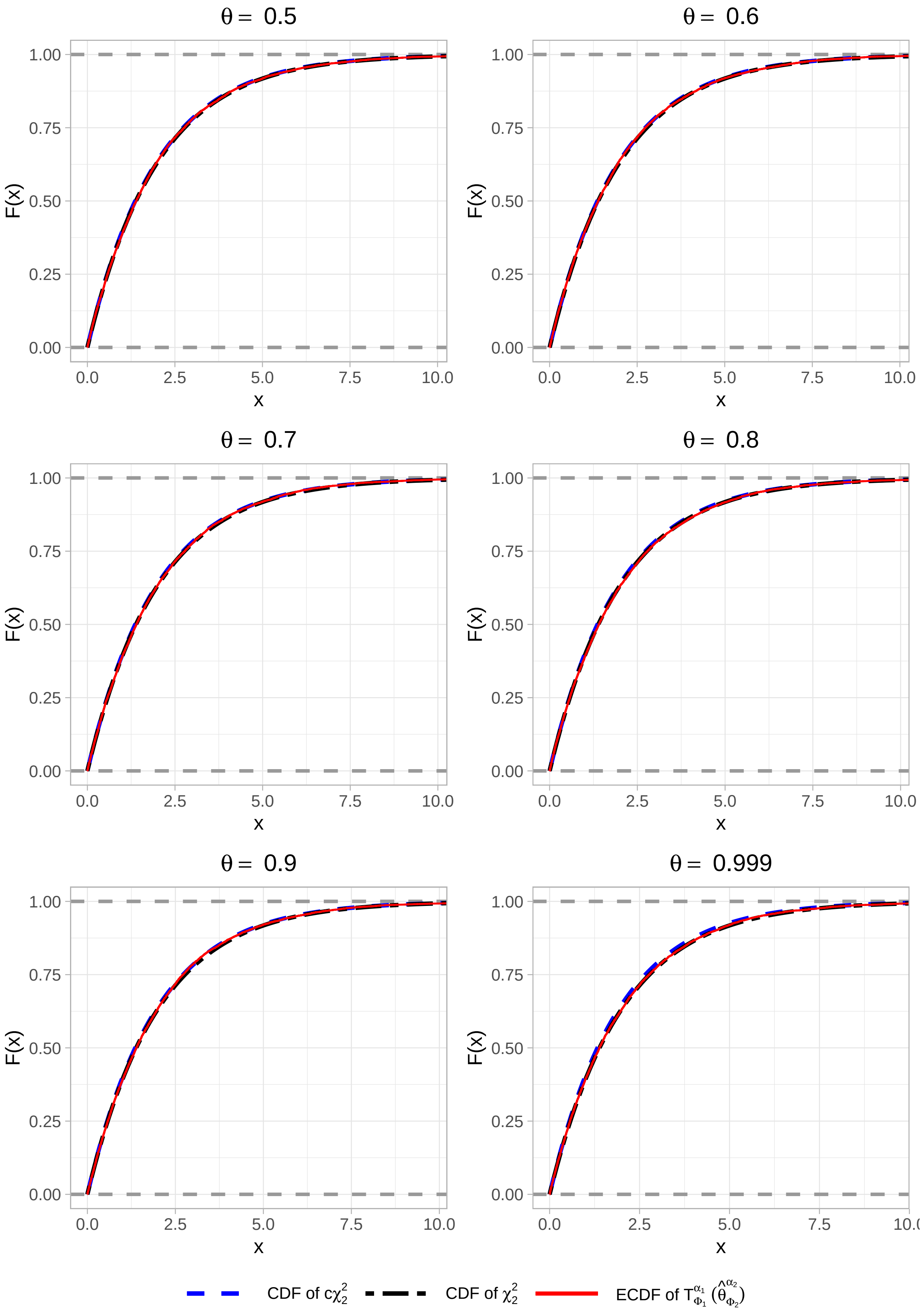


Figure 2.3: Comparison between the ECDF of $T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2})$ and the CDF of χ^2_2 and $c\chi^2_2$, $\Phi_1(u)$, $\Phi_2(u)$ as in (2.2), c as in (2.16), $\alpha_1 = 0.01$, $\alpha_2 = 0.1$.

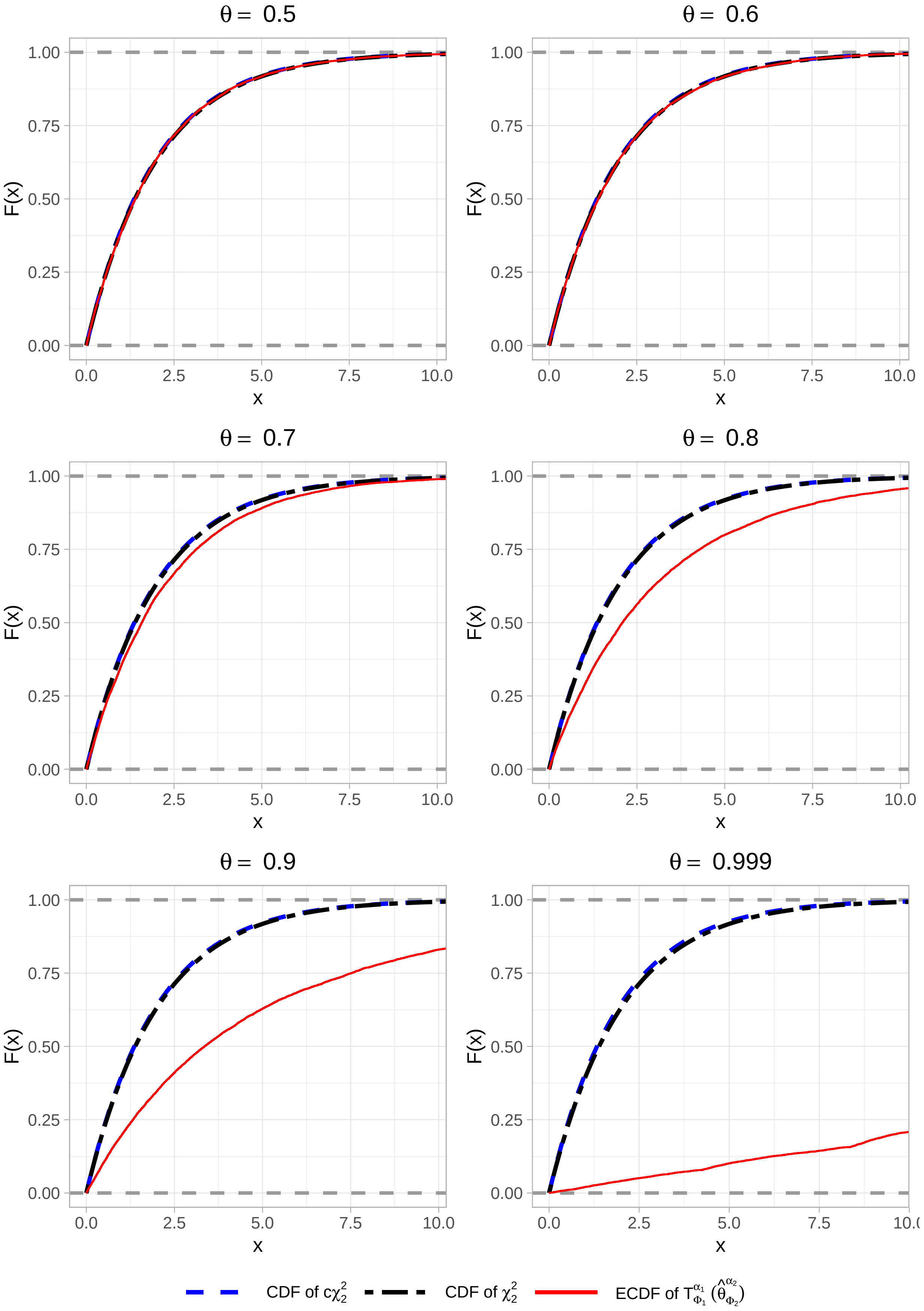


Figure 2.4: Comparison between the ECDF of $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})$ and the CDF of χ_2^2 and $c\chi_2^2$, $\Phi_1(u)$, $\Phi_2(u)$ as in (2.2), c as in (2.16), $\alpha_1 = 0.01$, $\alpha_2 = 5$.

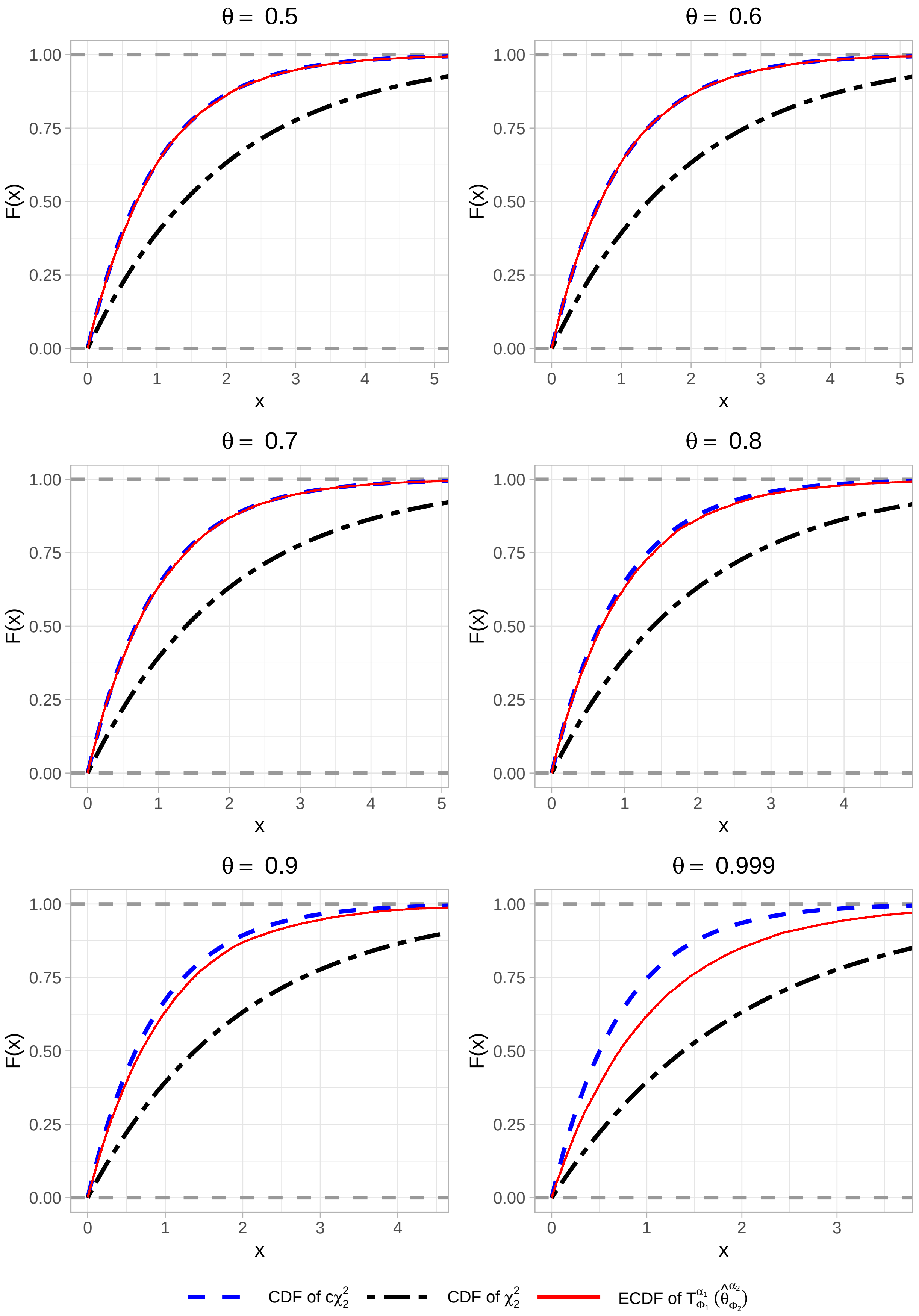


Figure 2.5: Comparison between the ECDF of $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})$ and the CDF of χ_2^2 and $c\chi_2^2$, $\Phi_1(u)$, $\Phi_2(u)$ as in (2.2), c as in (2.16), $\alpha_1 = 0.5, \alpha_2 = 0.1$.

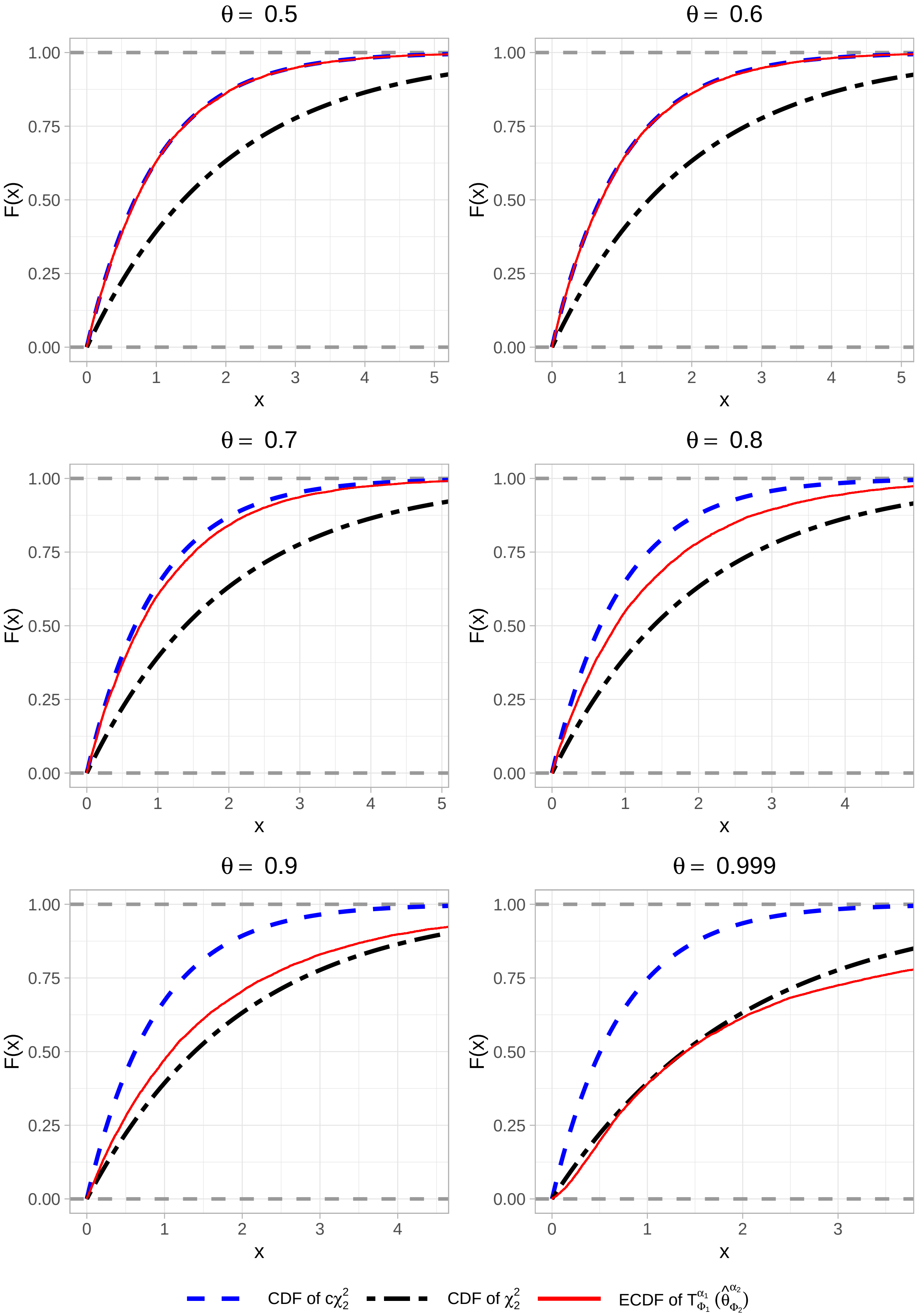


Figure 2.6: Comparison between the ECDF of $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})$ and the CDF of χ_2^2 and $c\chi_2^2$, $\Phi_1(u)$, $\Phi_2(u)$ as in (2.2), c as in (2.16), $\alpha_1 = 0.5$ & $\alpha_2 = 5$.

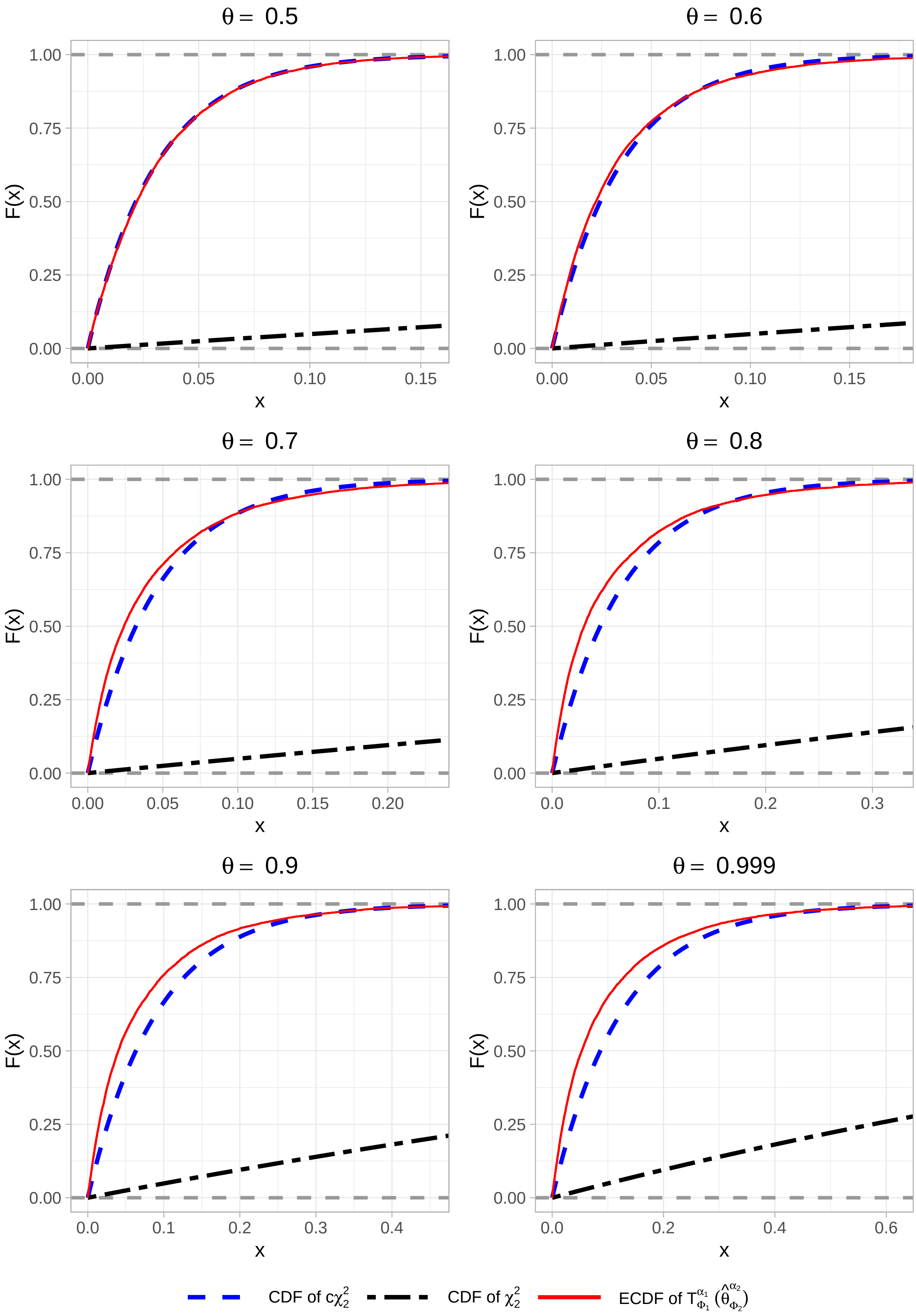


Figure 2.7: Comparison between the ECDF of $T_{\Phi_1}^{\alpha_1}(\hat{\theta}_{\Phi_2}^{\alpha_2})$ and the CDF of χ_2^2 and $c\chi_2^2$, $\Phi_1(u)$, $\Phi_2(u)$ as in (2.2), c as in (2.16), $\alpha_1 = 3$ & $\alpha_2 = 0.1$.

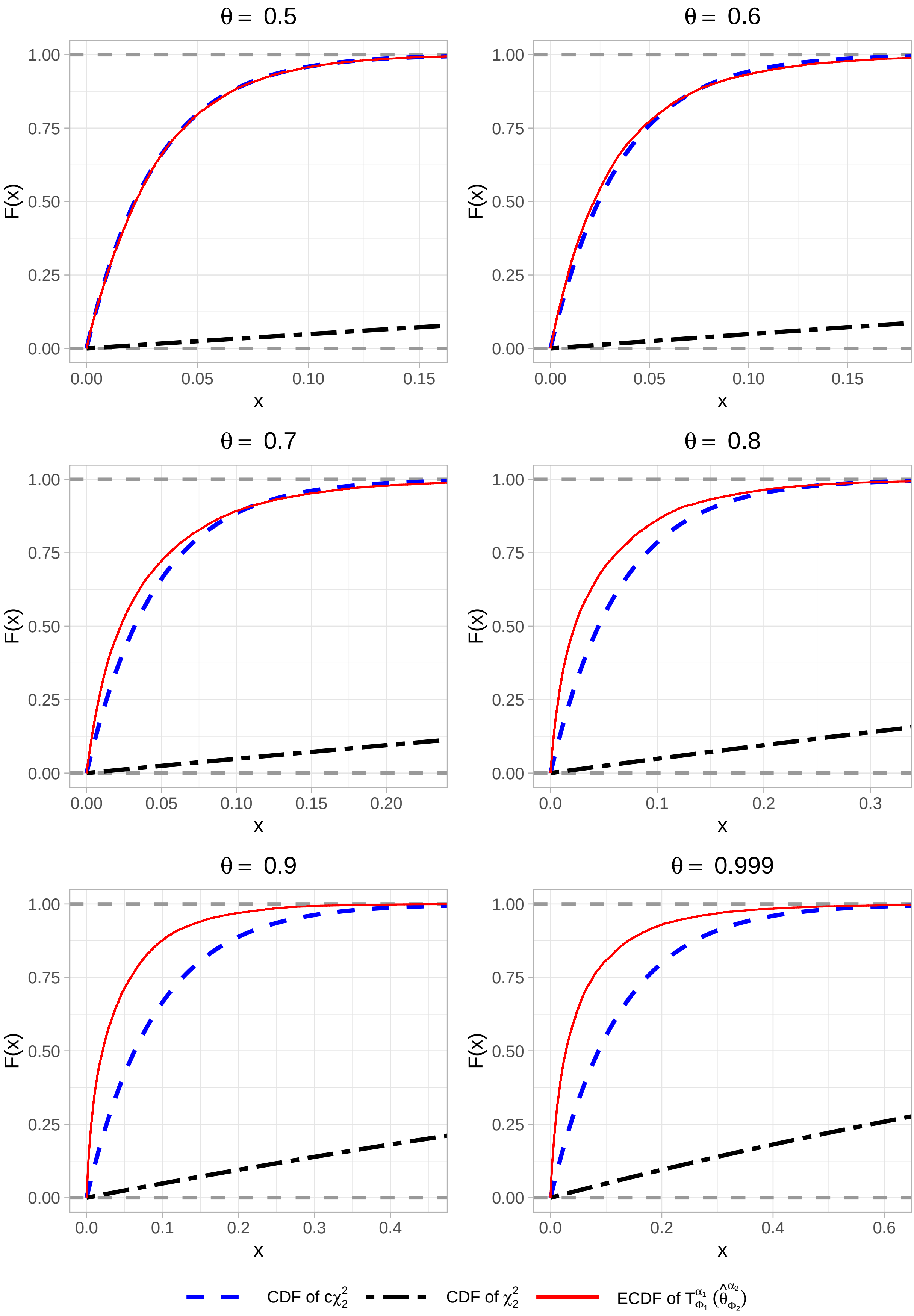


Figure 2.8: Comparison between the ECDF of $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})$ and the CDF of χ_2^2 and $c\chi_2^2$, $\Phi_1(u)$, $\Phi_2(u)$ as in (2.2), c as in (2.16), $\alpha_1 = 3$ & $\alpha_2 = 5$.

Remark 2.9. *Note that in this chapter we propose a general family of test statistics for testing the composite null hypothesis (2.4) and establishing its asymptotic theory. The primary purpose of the simulation study undertaken, is to explore the behaviour and the accuracy of the asymptotic distribution of the test statistics (2.12) and (2.5) not only under the equiprobable model but also for nonnecessarily equiprobable (i.e. skewed) models. In addition, we have included the first results regarding the power of the proposed test (end of Section 2.3.1) and the asymptotic distribution under contiguous alternatives (Section 2.3.2). These results set the basis for investigating, in the next level, the capabilities of the proposed tests in terms of power. Our intention is to evaluate the Bahadur slope of the proposed test as compared with various competing tests available in the literature and for various alternatives. Based on the findings of this simulation study, we claim that the efficiency of the proposed test, as expressed by the Bahadur slope, is expected to be close to the one of similar tests for the equiprobable model as well as for the cases where the index α is close to 0. The power behaviour of the test for models that deviate significantly from the equiprobable model remains to be seen. Thus, the power performance and in particular the Bahadur slope, which is considered to be a natural definition of efficiency relying on a limiting function of the p−value of the test under various alternatives (see Nikitin, 1995; Bahadur, 1967, 1971), remains an open problem of future research.*

Table 2.1: Type I error (0.05 significance level) calculations (%) based on 100000 simulations from each multinomial model with sample size 10000. Comparison of the double index $(\Phi, \alpha)-$test statistic for $\alpha_1 = 0.01, 0.05, 0.1, 0.5, 3$ and $\alpha_2 = 0.01, 0.05, 0.1, 0.5, 5$.

(a) $\theta = 0.5$

	α_1				
α_2	0.01	0.05	0.10	0.50	3.00
0.01	5.03	5.03	5.03	5.03	5.03
0.05	5.03	5.03	5.03	5.03	5.03
0.10	5.03	5.03	5.03	5.03	5.03
0.50	5.03	5.03	5.03	5.03	5.03
5.00	5.03	5.03	5.03	5.03	5.03

(b) $\theta = 0.6$

	α_1				
α_2	0.01	0.05	0.10	0.50	3.00
0.01	5.02	5.02	5.03	5.03	5.87
0.05	5.02	5.02	5.03	5.03	5.87
0.10	5.02	5.02	5.03	5.03	5.87
0.50	5.02	5.03	5.04	5.03	5.86
5.00	5.21	5.20	5.21	5.15	5.84

(c) $\theta = 0.7$

	α_1				
α_2	0.01	0.05	0.10	0.50	3.00
0.01	5.07	5.07	5.09	5.21	6.14
0.05	5.07	5.07	5.09	5.21	6.12
0.10	5.07	5.07	5.09	5.20	6.10
0.50	5.10	5.10	5.11	5.17	5.96
5.00	7.25	7.21	7.14	6.75	5.64

(d) $\theta = 0.8$

	α_1				
α_2	0.01	0.05	0.10	0.50	3.00
0.01	5.04	5.11	5.20	5.81	5.44
0.05	5.05	5.11	5.20	5.79	5.39
0.10	5.05	5.11	5.20	5.76	5.32
0.50	5.26	5.30	5.37	5.65	4.80
5.00	14.75	14.51	14.19	11.42	3.60

(e) $\theta = 0.9$

	α_1				
α_2	0.01	0.05	0.10	0.50	3.00
0.01	5.04	5.31	5.60	7.04	4.52
0.05	5.05	5.31	5.60	6.97	4.43
0.10	5.06	5.31	5.59	6.91	4.32
0.50	5.84	5.96	6.13	6.55	3.30
5.00	31.85	31.05	30.02	20.21	1.11

(f) $\theta = 0.999$

	α_1				
α_2	0.01	0.05	0.10	0.50	3.00
0.01	5.37	7.05	8.98	12.44	4.03
0.05	5.37	7.05	8.98	12.44	4.03
0.10	5.37	7.05	8.97	12.44	4.02
0.50	8.94	10.10	11.44	12.24	3.63
5.00	88.14	87.17	83.95	36.11	1.82

Before closing this subsection, the Type I error is investigated for the proposed double index $(\Phi, \alpha)-$test statistic given in (2.5). The analysis is based on five values of α_1 and α_2, namely for $\alpha_1 = 0.01,\ 0.05,\ 0.1,\ 0.5,\ 3$ and $\alpha_2 = 0.01,\ 0.05,\ 0.1,\ 0.5,\ 5$. In order to have a better insight we generated 100000 samples from each multinomial model with sample size of 10000 observations. Again, the proposed double index $(\Phi, \alpha)-$test statistic is applied for Φ_1 and Φ_2 as in (2.2). The critical values that have been used are the asymptotic critical values with a 0.05 level of significance based on the asymptotic $c\chi^2_2$ distribution with c as in (2.16). Results are presented in Table 2.1 where we record the proportion of times that the null hypothesis is rejected. We observe that, under the equiprobable model ($\theta = 0.5$), the Type I error is very close to the nominal level irrespectively of the values of the indices α_1 and α_2. Adequate results, for each value of the indices, are obtained also for small departures from the equiprobable model. Further, we can see that as we deviate from the equiprobable model, the results are very good for small values of α_1 and α_2 with the best results obtained when additionally α_1 is smaller than α_2.

Taking into consideration the simulation results for the accuracy of the asymptotic distribution and the Type I error, we conclude that the double index $(\Phi, \alpha)-$family is more effective if α_1 is relatively small and α_2 is chosen to be larger than α_1. The value of α_2 is irrelevant for models close to the equiprobable model, but for models far away from the equiprobable, a small value of α_2 ensures very high accuracy. The results show that it is always possible to choose an ideal pair of indices to ensure the correctness of the underlying distribution with a nominal Type I error for equiprobable as well as skewed models.

2.4.2 Comparison of test statistics

Let $\mathbf{Y} = (Y_1, \ldots, Y_N)^\top$ be a random vector from the distribution F. We can test the composite null hypothesis, $H_0\colon F = F_{\boldsymbol{\theta}_0}$, that the distribution function F is a member of a parametric family $\{F_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$, $\Theta \subseteq \mathbb{R}^s$ by partitioning the data$-$range into m disjoint intervals and testing the hypothesis $H_0\colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0)$ about the parameter$-$vector of the resulting multinomial distribution, for some unknown $\boldsymbol{\theta}_0 \in \Theta \subseteq \mathbb{R}^s$ and $s < m - 1$.

Let $P = \{E_i\}_{i=1,\ldots,m}$ be a partition of the real line $\mathbb{R}$ into m intervals. The probabilities of the intervals E_i, $i = 1, \ldots, m$ depend on the unknown parameter $\boldsymbol{\theta}_0$, in such a way that $p_i(\boldsymbol{\theta}_0) = \Pr_{F_{\boldsymbol{\theta}_0}}(E_i) = \int_{E_i} dF_{\boldsymbol{\theta}_0}$. Let $x_i = \sum_{j=1}^N I_{E_i}(Y_j)$, i = 1,…,m where

$$I_{E_i}(Y_j) = \begin{cases} 1, & \text{if } Y_j \in E_i \\ 0, & \text{otherwise} \end{cases},$$

with $\sum_{i=1}^m x_i = N$ and $\hat{p}_i$ be the nonparametric MLE of the true probablity of the E_i interval, defined at the beginning of this chapter.

In order to understand the behaviour of the double index $(\Phi, \alpha)-$test statistic given in (2.5) we compare it with three tests based on the $\varphi-$divergence family of measures, namely the tests of fit based on the Kullback$-$Leibler measure $(G^2(\hat{\boldsymbol{\theta}}))$, the Pearson's chi$-$squared measure $(X^2(\hat{\boldsymbol{\theta}}_{X^2}))$ and the Cressie and Read measure with $\lambda = 2/3$ $(CR(\hat{\boldsymbol{\theta}}^{\lambda}_{\mathrm{CR}}))$. Note that 2/3 is the value that Cressie & Read (1984) proposed for λ as a competing alternative to the $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ and $G^2(\hat{\boldsymbol{\theta}})$. Observe that for every gof test statistic the estimator of the unknown parameter is based on the same measure, although this is not mandatory in general. The proposed test $(T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2}))$ is applied for $\Phi_1(u) = 1 - (1 + \alpha_1^{-1})u + \alpha_1^{-1}u^{1+\alpha_1}$ and $\Phi_2(u) = 1 - (1 + \alpha_2^{-1})u + \alpha_2^{-1}u^{1+\alpha_2}$ and three different values of α_1 and α_2, namely $\alpha_1, \alpha_2 = 0.01, 0.10$, and 0.50.

For the null distribution we focus on the Gamma distribution where the shape parameter is taken to be equal to 0.44, 1 and 2. These values of the shape parameter correspond to skewness equal to 3, 2 and 1.414, respectively. Distributions with the same skewness values have been used as possible alternatives in each case, namely the Weibull (Wei), Log−normal (LN) and Inverse Gaussian (IG). The scale parameter is taken to be equal to 1 in all cases, since the Gamma distribution is scale invariant. Note that the same distributions have been used previously by Vonta et al. (2012) in order to examine the performance of the test statistic (1.13) for the simple null hypothesis (1.9) as well as by Koutrouvelis & Karagrigoriou (2012) for the analysis of the Inverse Gaussian distribution. This set−up has been chosen in order to examine whether the test statistics under consideration are able to distinguish between distributions with the same skewness. Likewise, this set−up can be extended to the case where the alternative models are defined via their excess kurtosis in reference to the kurtosis of the exponential distribution, which can be useful for practical purposes in actuarial science, where the discrimination of distributions with fatter tails is of great importance.

For the partition of the data−range, we use $m = 3$ intervals and we consider four models for the resulting multinomial distribution, namely the equiprobable $(1/3, 1/3, 1/3)$, symmetric $(0.2, 0.6, 0.2)$, left skewed $(0.1, 0.5, 0.4)$ and right skewed $(0.4, 0.5, 0.1)$ model.

In order to investigate the power for various alternatives and the size of the proposed test, we generated 10000 samples from each distribution with sample sizes $N = 25$, 50, 75, 100, 500. The results for every sample size N are provided at the Appendix A. Here we report only the results for $N = 25$ (Table 2.2) and the models

(a) equiprobable $(1/3, 1/3, 1/3)$ and

(b) left skewed $(0.1, 0.5, 0.4)$

as well as for $N = 50$ (Table 2.3) and the models

(c) symmetric $(0.2, 0.6, 0.2)$ and

(d) right skewed $(0.4, 0.5, 0.1)$.

Note that, in terms of size, the test statistic with the best performance is considered to be the one whose empirical size is closer to the nominal 0.05 one. Test statistics with empirical size greater than 0.05 are considered as liberal, while those with empirical size smaller than 0.05 as conservative. Nevertheless, Dale's (Dale, 1986) criterion (see Subsection 2.4.3, where it is used for the first time in this thesis) could be utilized in order to derive conclusions regarding the closeness of the empirical size to the nominal one. According to the criterion, the empirical size is considered to be close to the nominal 0.05 if it belongs to the range $[0.0357, 0.0695]$.

In terms of power, the best behaving test statistic among all competing tests, is considered the one with the largest power, even if it is much smaller than 1. In every case, the largest power is depicted in bold in the tables.

In particular, in Table 2.2 we observe that for the equiprobable model, $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ and $CR(\hat{\boldsymbol{\theta}}_{\mathrm{CR}}^{\lambda})$ have the best performance among all competing tests for every alternative, with the first being slightly better. Also, we observe that $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ and $G^2(\hat{\boldsymbol{\theta}})$ have

Table 2.2: Size (S) and powers (%) for the equiprobable and left skewed models, $N = 25$. Higher power is depicted in bold.

Shape parameter	$T_{\Phi_1}^{0.10}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$	$G^2(\hat{\boldsymbol{\theta}})$	$X^2(\hat{\boldsymbol{\theta}}_{X^2})$	$CR(\hat{\boldsymbol{\theta}}_{\text{CR}}^{\lambda})$
Equiprobable model (1/3, 1/3, 1/3)						
(S) Gamma(0.44)	4.74	4.30	4.93	4.83	6.12	5.63
Wei(0.77)	12.00	11.89	12.27	12.01	**12.43**	12.21
LN(0.729)	13.23	11.88	13.44	13.44	**17.44**	16.00
IG(1)	12.74	11.54	12.96	12.97	**16.75**	15.48
(S) Gamma(1)	4.88	4.38	4.94	4.93	6.21	5.72
Wei(1)	5.12	4.72	5.25	5.19	**6.70**	6.27
LN(0.551)	36.61	33.92	37.06	37.09	**44.69**	42.33
IG(2.25)	31.67	29.14	32.05	32.05	**39.24**	36.91
(S) Gamma(2)	4.71	4.36	4.84	4.78	6.14	5.63
Wei(1.259)	9.14	8.15	9.31	9.29	**12.33**	11.19
LN(0.423)	69.06	66.46	69.48	69.50	**77.25**	74.84
IG(4.5)	64.53	61.56	65.11	65.13	**72.93**	70.47
Left skewed model (0.1, 0.5, 0.4)						
(S) Gamma(0.44)	3.99	4.06	5.74	3.84	4.58	4.33
Wei(0.77)	10.55	10.53	**16.17**	10.27	11.56	11.14
LN(0.729)	25.44	24.50	26.03	25.61	**27.68**	26.99
IG(1)	31.59	30.54	32.24	31.75	**34.40**	33.41
(S) Gamma(1)	3.95	3.90	5.61	3.90	4.65	4.41
Wei(1)	4.01	3.85	**5.84**	3.97	4.75	4.49
LN(0.551)	64.73	63.18	65.51	64.99	**67.63**	66.79
IG(2.25)	63.68	62.08	64.48	63.93	**66.54**	65.74
(S) Gamma(2)	3.97	3.58	5.73	3.91	4.74	4.49
Wei(1.259)	7.76	7.41	8.18	7.79	**8.80**	8.35
LN(0.423)	94.44	93.98	94.67	94.53	**95.38**	95.16
IG(4.5)	93.45	92.86	93.65	93.48	**94.48**	94.07

a similar behaviour. In terms of size (denoted by (S)) the best results are obtained for $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ while $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ and $CR(\hat{\boldsymbol{\theta}}_{\text{CR}}^{\lambda})$ appear to be liberal. Similar results (not shown here) are obtained for bigger sample sizes ($N \geq 50$), with the exception that against the Weibull alternative (Wei(0.77)) the best results are obtained for $T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$ and $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ with the first being slightly better.

For the left skewed model and the Weibull alternative, $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ has the best performance in most cases. For the remaining alternatives the best results are obtained for $X^2(\hat{\boldsymbol{\theta}}_{X^2})$, while $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ and $G^2(\hat{\boldsymbol{\theta}})$ have a similar behaviour with the first being slightly better. As the sample size N increases (results not shown here) $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ performs better than $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ and $CR(\hat{\boldsymbol{\theta}}_{\text{CR}}^{\lambda})$ for every alternative considered. It is interesting to note that for alternatives like Inverse Gaussian and Log−normal the best results when $N \geq 50$ are obtained for $T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$. In terms of size, when $N = 25$ the best results are obtained for $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ while for bigger sample sizes the best results are obtained for $G^2(\hat{\boldsymbol{\theta}})$ in most cases. The test statistic $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ appears to be liberal for every sample size.

Table 2.3 provides the results for the symmetric and right skewed models with sample size $N = 50$. The results reveal that for the symmetric model in most cases

Table 2.3: Size (S) and powers (%) for the symmetric and right skewed models, $N = 50$. Higher power is depicted in bold.

Shape parameter	$T_{\Phi_1}^{0.10}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$	$G^2(\hat{\boldsymbol{\theta}})$	$X^2(\hat{\boldsymbol{\theta}}_{X^2})$	$CR(\hat{\boldsymbol{\theta}}_{\text{CR}}^{\lambda})$
Symmetric model (0.2, 0.6, 0.2)						
(S) Gamma(0.44)	5.33	5.08	5.59	5.46	5.73	5.64
Wei(0.77)	28.66	25.01	30.41	29.37	**31.84**	31.03
LN(0.729)	46.77	**47.20**	46.72	46.71	46.59	46.59
IG(1)	42.82	**43.28**	42.74	42.72	42.42	42.43
(S) Gamma(1)	5.16	4.93	5.30	5.19	5.45	5.34
Wei(1)	5.15	4.81	5.34	5.22	**5.55**	5.41
LN(0.551)	97.12	**97.20**	97.11	97.11	97.08	97.06
IG(2.25)	94.34	**94.54**	94.34	94.34	94.32	94.32
(S) Gamma(2)	5.21	4.89	5.37	5.26	5.54	5.45
Wei(1.259)	29.13	**29.48**	29.05	29.04	28.89	28.89
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	99.98	99.98	99.98	99.98	99.98	99.98
Right skewed model (0.4, 0.5, 0.1)						
(S) Gamma(0.44)	4.61	3.66	5.34	4.95	5.19	5.10
Wei(0.77)	30.67	21.64	35.41	32.07	**37.44**	35.96
LN(0.729)	22.36	21.04	**23.20**	22.69	21.30	21.75
IG(1)	15.52	14.21	**16.46**	15.94	14.86	15.20
(S) Gamma(1)	4.95	4.22	5.72	5.23	5.37	5.34
Wei(1)	5.05	3.84	**5.84**	5.33	5.28	5.36
LN(0.551)	82.16	80.78	**83.02**	82.51	80.82	81.40
IG(2.25)	73.18	71.73	**74.15**	73.57	71.82	72.31
(S) Gamma(2)	5.33	4.36	6.20	5.58	5.63	5.66
Wei(1.259)	32.07	30.53	**33.23**	32.55	30.58	31.35
LN(0.423)	99.73	99.69	**99.73**	99.73	99.70	99.70
IG(4.5)	99.49	99.40	**99.56**	99.51	99.40	99.44

$T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$ has the best performance among all competing tests, not only in terms of power but also in terms of size. Test statistics $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ and $CR(\hat{\boldsymbol{\theta}}_{\text{CR}}^{\lambda})$ have an edge against the Weibull alternative (Wei(0.77)), but they appear to be liberal. As the sample size increases, although no test is clearly better than the others, $T_{\Phi_1}^{0.10}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$ has a size quite close to the nominal level, while $T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$ appears to have slightly higher power in most cases as compared to all tests examined.

For the right skewed model in most cases $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ performs better in terms of power than all other competing tests, but it appears to be liberal. The $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ test statistic appears to have a slight edge against the Weibull alternative (Wei(0.77)). Similar results are obtained for bigger sample sizes.

In conclusion, for every sample size and under the right skewed model $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ appears to be the best choice among all tests considered, while for the symmetric model $T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$ should be preferred. For these two models $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ appears to have a slight edge against the Weibull (Wei(0.77)) alternative. Although for small sample sizes and the left skewed model $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ appears to have the best performance among all tests in most cases, when the sample size increases the best choice for every alternative is the $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$, while for alternatives like the Inverse Gaussian and the Log−normal the best choice is $T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$. Finally, for the equiprobable model, $X^2(\hat{\boldsymbol{\theta}}_{X^2})$ has the best performance among all tests in most cases, but as the sample size increases $T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$ has slightly better power against the Weibull (Wei(0.77)) alternative.

It should also be noted that, since the empirical size of each test statistic is different from the targeted nominal one, in order to derive solid conclusions regarding the best test statistic in terms of power, an adjustment is mandatory. To this regard, the method proposed by Lloyd (2005), which deploys the Receiver Operating Characteristic (ROC) curves could be used. For further details see Subsection 2.4.3.

2.4.3 Examination under contaminated data

The problem of contingency tables or cross tabulations and their statistical analysis based on measures of divergence, always attracts the attention of researchers with a plethora of important contributions (see e.g. Kateri & Papaioannou, 1997; Kateri et al., 1996; Kateri & Papaioannou, 1996). Such problems though, are often associated with two serious issues that frequently appear in practice and considerably affect both estimating and testing procedures, namely censoring and contamination often encountered among other fields, in survival analysis (see, Sachlas & Papaioannou, 2014; Vonta & Karagrigoriou, 2010; Basu et al., 2006). For an extensive overview of such issues and their handling, refer to Tsairidis et al. (1996) and Tsairidis et al. (2001). A thorough study of the $(\Phi, \alpha)-$power divergence family and the associated estimators and test statistics under the censoring regime remains an open problem of future research. The emphasis in this section is on contamination. More specifically, in order to attain a better insight of the behaviour of the proposed divergences used both for estimation and testing purposes we proceed further with a simulation study. The null hypothesis considered focuses on the Gamma distribution with shape parameter equal to 1, denoted by $\Gamma(1)$. On the other hand, as alternative hypotheses we have used Gamma distributions with shape parameters equal to 1.5, 4.0 and 10.0 denoted by $\Gamma(1.5)$, $\Gamma(4)$ and $\Gamma(10)$, respectively. In every case the scale parameter is chosen to be equal to 1 due to the fact that the Gamma distribution is scale invariant.

The study is implemented not only for the regular case but also for cases where the data set is contaminated. In this regard we define ϵ as the contamination level with $\epsilon \in [0, 1]$. Thus the data generating distribution has the form $(1-\epsilon)\Gamma_d + \epsilon\Gamma_c$ where Γ_d is the dominant and Γ_c the contaminant Gamma distribution. Note that the used contamination level is taken to be equal to 0.075. Thus, for the examination of estimators and test statistics in terms of size of the test (α) we contaminate the null distribution with observations from the alternative hypotheses and vice versa for the examination of tests in terms of power (γ). Furthermore, for the implementation, we have considered a large sample size, $N = 200$ and 100000 repetitions of the experiment, while for the partition of the data range we use $\lceil\sqrt{200}\rceil = 15$ equiprobable intervals, where the $\lceil\cdot\rceil$ operator returns the least integer which is greater than or equal to its argument.

As classical minimum divergence estimators we use those that can be derived from the Cressie−Read family for $\lambda = -2, -1, -1/2,\ 0,\ 2/3$, and 1 which are known as the minimum modified chi−squared $(\hat{\theta}_{MCS})$, discrimination information $(\hat{\theta}_{MDI})$, Freeman−Tukey $(\hat{\theta}_{FT})$, likelihood ratio $(\hat{\theta}_{LR})$, Cressie−Read $(\hat{\theta}_{CR})$ and chi−squared

($\hat{\theta}_{CS}$) estimators. On the other hand, the proposed BHHJ family ($\hat{\theta}_{(\Phi_2,\alpha_2)}$) of estimators is applied for thirteen values of the parameter $\alpha_2 = 10^{-7}, 0.01, 0.05, 0.10$ ... (0.10) ... 1.00 and Φ_2 as in (2.2) with $\alpha = \alpha_2$. Furthermore, we have included in our analysis not only the L_2-distance estimator ($\hat{\theta}_{L_2}$), which along with the likelihood ratio serve as benchmark estimators (divergences in general) since they are equivalent with the BHHJ family for $\alpha = 1$ and $\alpha \to 0$, respectively, but also the maximum likelihood estimator ($\hat{\theta}_{MLE}$) based on the ungrouped data.

In reference to the test statistics we proceed in a similar manner and retrieve from the Cressie−Read family the classical modified chi−squared $MCS(\hat{\theta}_{MCS})$, minimum discrimination information $MDI(\hat{\theta}_{MDI})$, Freeman−Tukey $FT(\hat{\theta}_{FT})$, likelihood ratio $LR(\hat{\theta}_{LR})$, Cressie−Read $CR(\hat{\theta}_{CR})$ and Pearson's chi−squared $CS(\hat{\theta}_{CS})$ test statistics along with the proposed $T_{\Phi_1}^{\alpha_1}\left(\hat{\theta}_{(\Phi_2,\alpha_2)}\right)$, for $\alpha_1 = 10^{-7}$, 0.01, 0.05, 0.10...(0.10)...1.00 and Φ_1 as in (2.2) with $\alpha = \alpha_1$.

The examination of the behaviour of the minimum divergence estimators is based on the mean squared error (MSE) given by

$$MSE_{\hat{\theta}} = \frac{1}{10^5}\sum_{l=1}^{10^5}\left(\hat{\theta}_l^I - \theta_0\right)^2$$

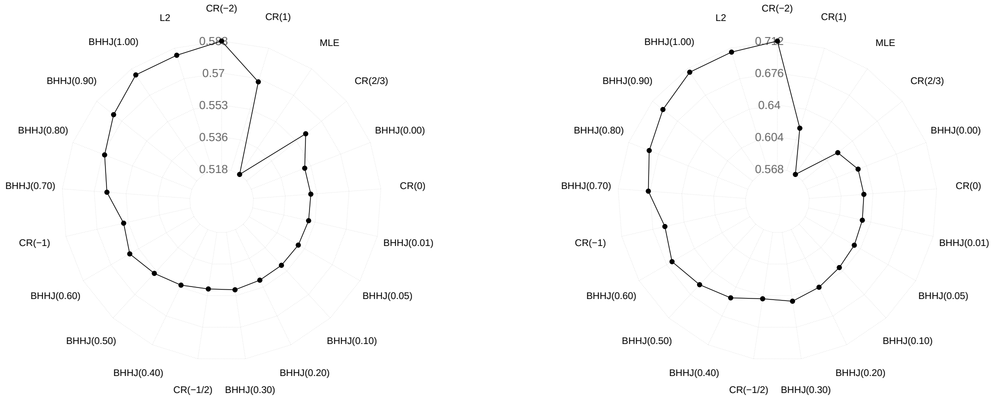
with θ_l^I being the minimum divergence estimator based on any I divergence for the l^{th} sample.

Figure 2.9 presents the MSE for the the four cases which are associated to no contamination and contamination from the three alternative distributions. The minimum divergence estimators are displayed in ascending order following a counter clockwise direction according to the case where the contaminant distribution lies far form the null, i.e. when the data are generated from $0.925\Gamma(1) + 0.075\Gamma(10)$.

The results given in Figure 2.9 indicate that in terms of MSE, estimators that can be derived from the Cressie−Read family with $\lambda \geq 0$ along with those that can be derived from the BHHJ family with small values of α_2 have better performance for the no contamination case and when the contaminant distribution is close to the null (Figures 2.9a & 2.9b). Note that in these two cases ($\hat{\theta}_{MLE}$) has the best performance among all competing estimators. On the contrary, when the contaminant distribution departs further from the null (Figures 2.9c & 2.9d), estimators from the BHHJ family with larger values of α_2 and those from the Cressie−Read family with negative values of λ appear to behave better, while the worst results arise for the $\hat{\theta}_{MLE}$. In addition, Figure 2.9 reveals the robustness aspect of the BHHJ and the Cressie−Read estimators, since it is apparent that in the presence of contamination the larger the value of the index α_2 and the smaller the value of the parameter λ the smaller the MSE.

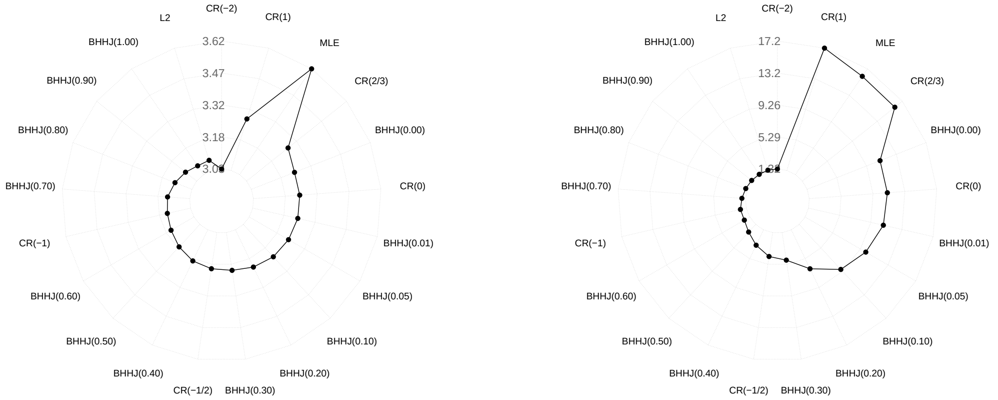
Finally, note that in every case the MSE of $\hat{\theta}_{(\Phi_2,\alpha_2)}$ lies between the MSEs of the $\hat{\theta}_{LR}$ and the $\hat{\theta}_{L_2}$. We should state here that for presentation purposes the MSE has been multiplied by 100. For more information about robust estimation for grouped data refer to (Basu et al., 1997; Victoria-Feser & Ronchetti, 1997; Lin & He, 2006; Toma & Broniatowski, 2011), while for the mathematical connection of the BHHJ and Cressie− Read families to Patra et al. (2013).

Under the set−up of this study we have $m = 15$ probabilities of the multinomial model and $k = 1$ unknown parameter to estimate, thus the critical values used are the asymptotic critical values based on the asymptotic distribution $c\chi^2_{13}$ with $c = 0.5\big(\min_i p_{i0}^{\alpha_1}(\hat{\theta}_{(\Phi_2,\alpha_2)}) + \max_i p_{i0}^{\alpha_1}(\hat{\theta}_{(\Phi_2,\alpha_2)})\big)$ for the BHHJ family of test statistics, and the χ^2_{13} for the classical test statistics that can be derived from the Cressie−Read family, with nominal level equal to 0.05.



(a) $\Gamma(1)$

(b) $0.925\Gamma(1) + 0.075\Gamma(1.5)$

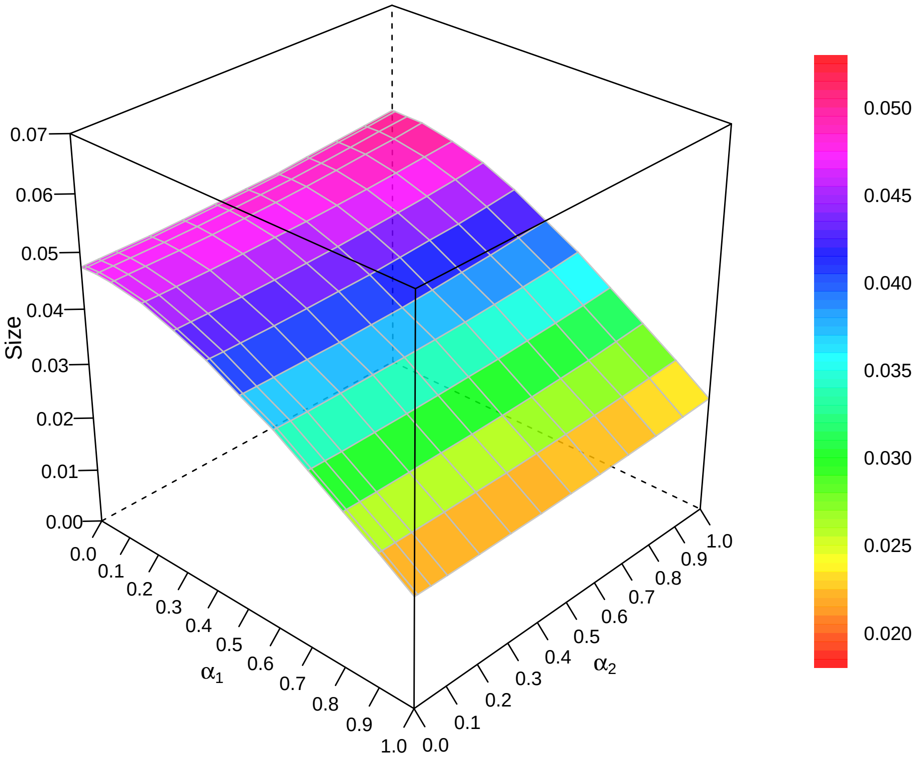


(c) $0.925\Gamma(1) + 0.075\Gamma(4)$

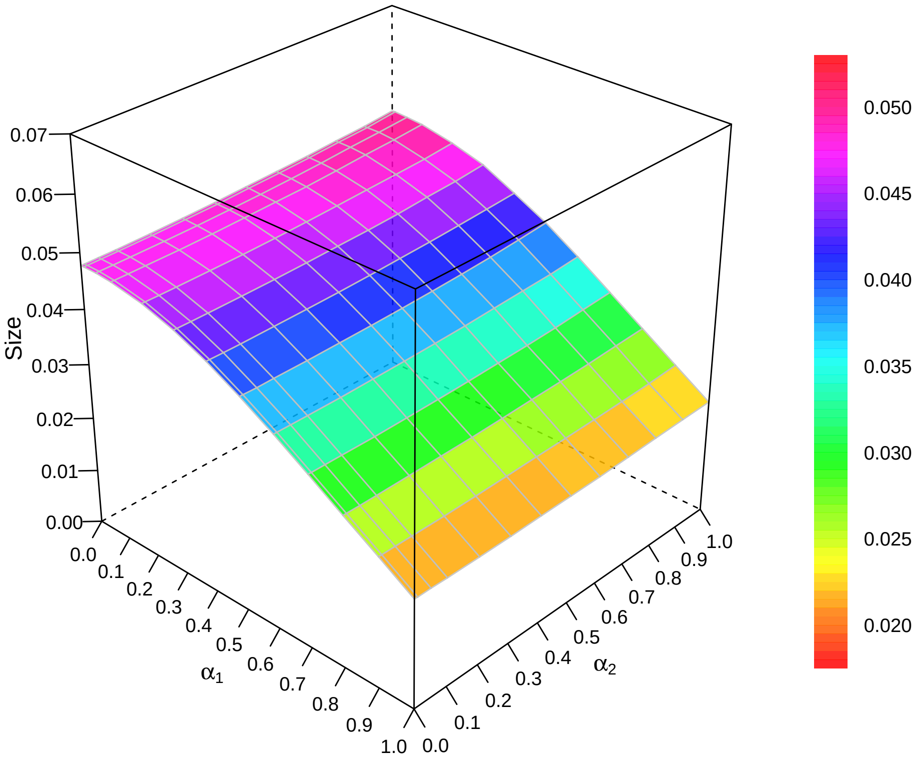
(d) $0.925\Gamma(1) + 0.075\Gamma(10)$

Figure 2.9: MSE ($\times 100$) for the four cases of contamination regarding the tests that can be derived both from the BHHJ and Cressie−Read families.

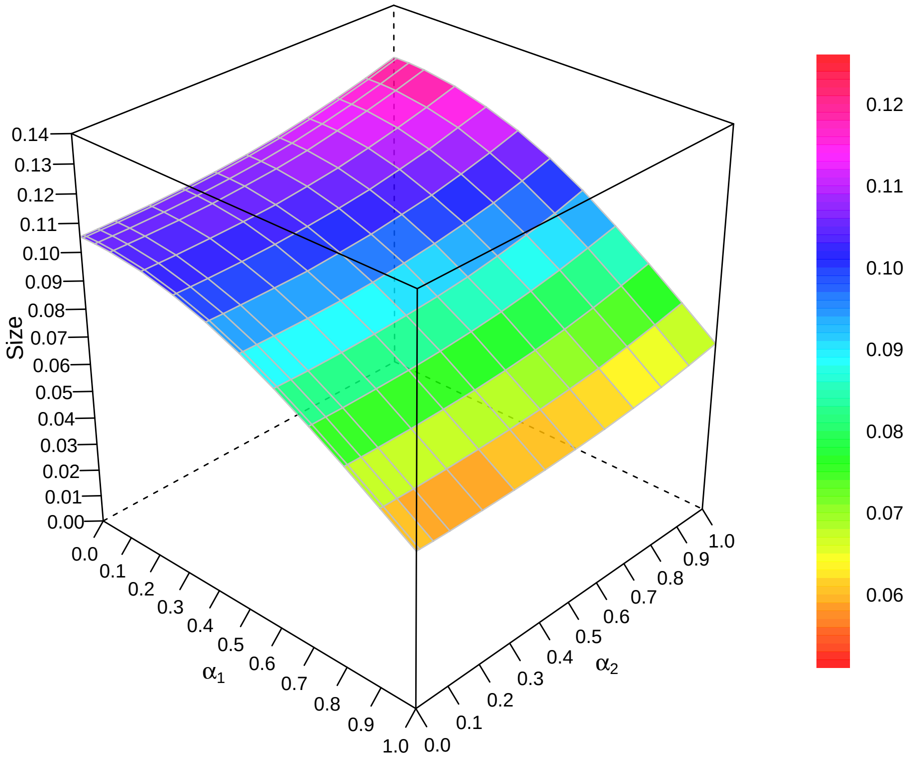
In Figure 2.10 we examine under the four aforementioned cases the behaviour of the BHHJ test statistics in terms of size for various values of the indices α_1 and α_2, while in Table 2.4 the behaviour of the classical tests is presented. In general we can see that as the index α_1 increases the size decreases, while as the index α_2 increases the size increases as well. Furthermore, we can observe that in the case where the contaminant distribution lies far from the null (Figure 2.10d) the size becomes very large, indicating the disastrous effect imposed from the contaminant distribution to all BHHJ test statistics. This disastrous effect is also apparent in the classical test statistics. In the case where the contaminant distribution is the $\Gamma(4)$ (Figure 2.10c) the BHHJ family of tests discounts the effect of contamination for values of $\alpha_1 \geq 0.8$, while the classical tests are largely affected by the contamination once again. Finally, for the no contamination and contamination from the $\Gamma(1.5)$, we can derive the following conclusions about the behaviour of the tests.



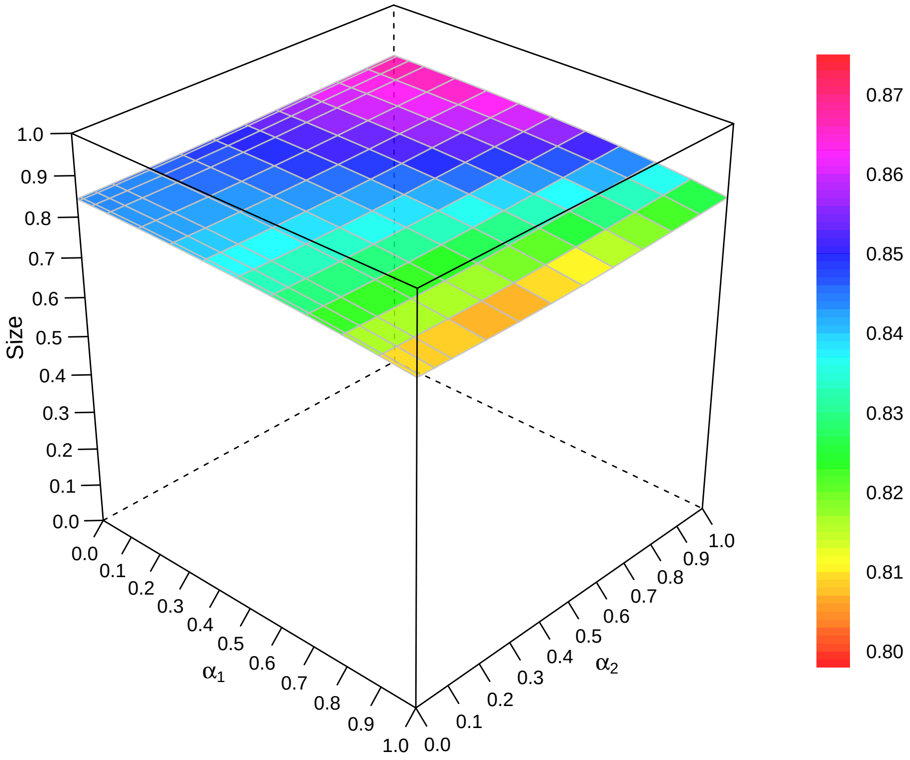
(a) $\Gamma(1)$



(b) $0.925\Gamma(1) + 0.075\Gamma(1.5)$



(c) $0.925\Gamma(1) + 0.075\Gamma(4)$



(d) $0.925\Gamma(1) + 0.075\Gamma(10)$

Figure 2.10: Size for the four contamination cases regarding the tests that can be derived from the BHHJ family.

Table 2.4: Size (%) for the four contamination cases regarding the classical tests that can be derived from the Cressie–Read family.

Data distribution	*FT*	*CR*	*CS*	*LR*	*MDI*	*MCS*
$\Gamma(1)$	4.028	6.538	7.841	4.744	3.783	4.263
$0.925\Gamma(1) + 0.075\Gamma(1.5)$	3.943	6.759	8.195	4.775	3.659	4.072
$0.925\Gamma(1) + 0.075\Gamma(4)$	9.546	12.762	14.392	10.539	9.063	9.521
$0.925\Gamma(1) + 0.075\Gamma(10)$	83.558	85.420	85.953	85.953	82.183	68.850

Regarding the BHHJ family (Figures 2.10a & 2.10b), we can observe that the larger the value of α_1 the more conservative the test is, while the best performance appears for $\alpha_1 \leq 0.10$ and $\alpha_2 \geq 0.50$. With respect to the classical tests, $MDI(\hat{\theta}_{MDI})$, $FT(\hat{\theta}_{FT})$ and $MCS(\hat{\theta}_{MCS})$ appear to be conservative, while $CS(\hat{\theta}_{CS})$ and $CR(\hat{\theta}_{CR})$ appear to be liberal. Note that in terms of size $LR(\hat{\theta}_{LR})$ appears to have the best performance among all classical test statistics.

In terms of power, results are presented in Figure 2.11 and Table 2.5 for the BHHJ and classical tests, respectively. Note that we only present results that are associated with the $\Gamma(1.5)$ alternative since in every other case the power reaches the highest level 1 for all tests. As a general conclusion we can state that the contamination affects the performance of all tests by notably downgrading their power. Concerning the BHHJ tests, the best results appear for small values of α_1 and large values of α_2, while the classical modified chi–squared test statistic, $MCS(\hat{\theta}_{MCS})$, has the best performance among all classical tests.

Table 2.5: Power (%) for the no contamination & contamination from $\Gamma(1)$ cases regarding the classical tests that can be derived from the Cressie–Read family.

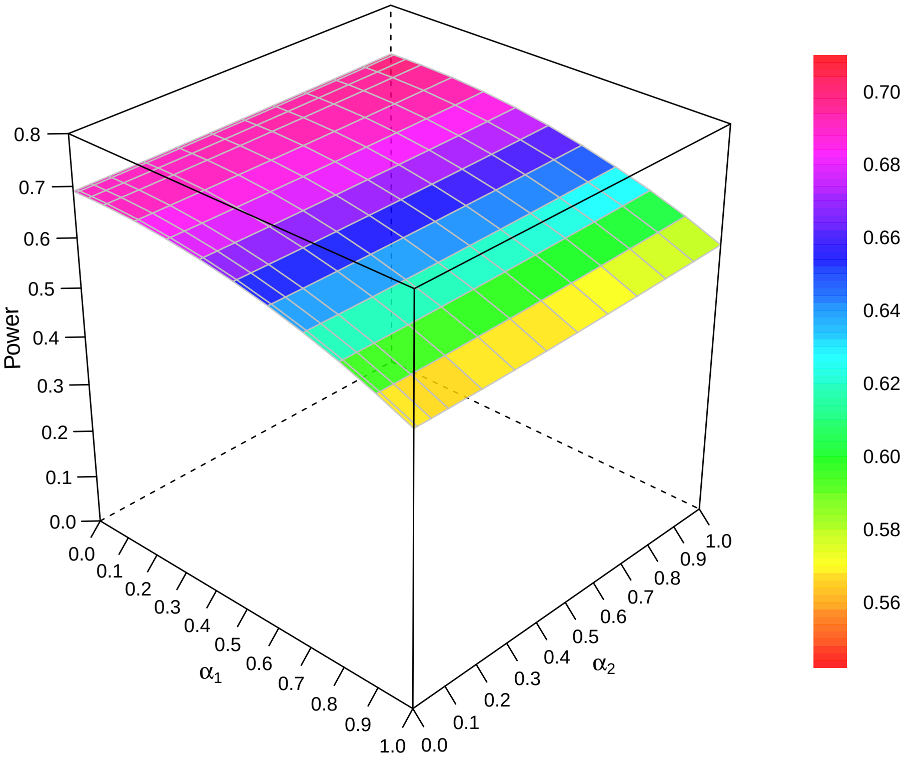
Data distribution	*FT*	*CR*	*CS*	*LR*	*MDI*	*MCS*
$\Gamma(1.5)$	69.412	69.911	70.887	69.061	70.568	74.808
$0.925\Gamma(1.5) + 0.075\Gamma(1)$	55.308	57.515	59.049	55.571	56.224	60.448

Based on the preceding analysis, we proceed further with the comparison of the tests. Beyond the classical ones we choose the following four test statistics from the BHHJ family $T1 \equiv T^{0.05}_{\Phi_1}(\hat{\theta}_{(\Phi_2,0.90)})$, $T2 \equiv T^{0.30}_{\Phi_1}(\hat{\theta}_{(\Phi_2,0.30)})$, $T3 \equiv T^{0.60}_{\Phi_1}(\hat{\theta}_{(\Phi_2,0.60)})$ and $T4 \equiv T^{0.90}_{\Phi_1}(\hat{\theta}_{(\Phi_2,0.30)})$. In order to derive solid conclusions about the behaviour of the test statistics in terms of size we consider Dale's criterion (Dale, 1986) which involves the following inequality

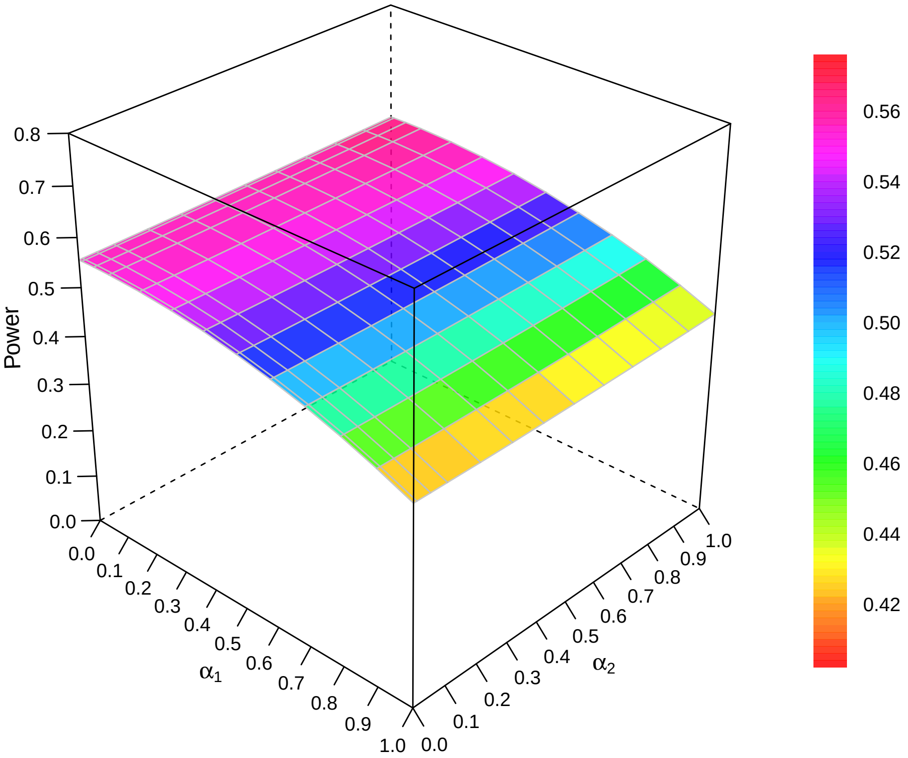
$$|\mathrm{logit}(1-\hat{\alpha}_n) - \mathrm{logit}(1-\alpha)| \leq d, \tag{2.19}$$

where $\mathrm{logit}(p) = \log(p/(1-p))$, while $\hat{\alpha}_n$ and α are the exact simulated and nominal sizes, respectively. When (2.19) is satisfied with $d = 0.35$ the exact simulated (empirical) size is considered to be close to the nominal size.

For $\alpha = 0.05$ the exact simulated size is close to the nominal if $\hat{\alpha}_n \in [0.0357, 0.0695]$. This criterion has been used previously by Pardo (2010) and Batsidis et al. (2016b). We apply the criterion not only for $\alpha = 0.05$, but also for a range of nominal sizes that are of interest, namely $\alpha \in [0, 0.1]$.



(a) $\Gamma(1.5)$



(b) $0.925\Gamma(1.5) + 0.075\Gamma(1)$

Figure 2.11: Power for the no contamination & contamination from $\Gamma(1)$ cases regarding the tests that can be derived from the BHHJ family.

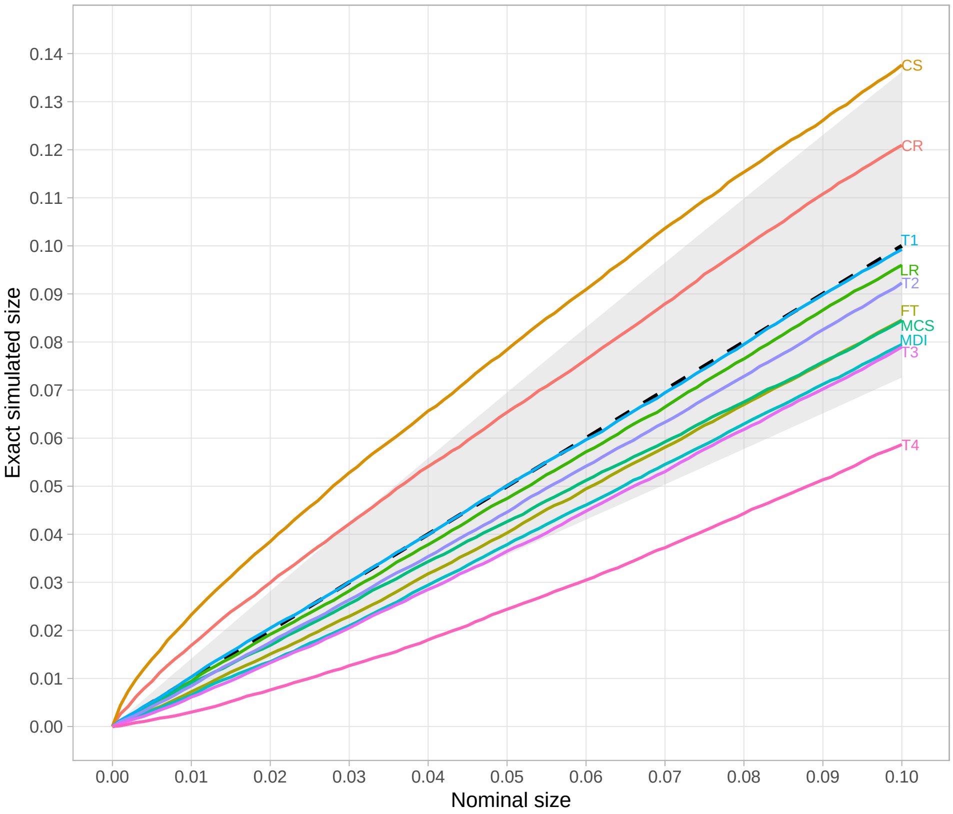
Results are presented in Figure 2.12 where the dashed line refers to the situation where the exact simulated equals the nominal size, thus lines that lie above this reference line refer to liberal, while those that lie below to conservative test statistics. Furthermore, the gray area that is depicted in Figure 2.12 refers to Dale's criterion, thus lines that lie into this area satisfy the criterion. From Figures 2.12a & 2.12b we observe that in the no contamination case and when the contaminant distribution is close to the null, besides CS and $T4$ every other test satisfies Dale's criterion. On a more granular level, we observe that the CR test statistic satisfies the criterion only for nominal sizes $\alpha \geq 0.03$. For the case where the contaminant distribution is the $\Gamma(4)$ we can see that the only test that resist the contamination and satisfies the criterion is the $T4$. One conclusion that can be derived from Figure 2.12d, is that even though every test fails to satisfy Dale's criterion, MCS appears to be notable resistant to the contamination, in relation to all other tests, especially for small nominal sizes.

Apparently, the actual size of each test differs from the targeted nominal one, thus in order to proceed further with the comparison of the tests in terms of power we have to make an adjustment. We follow the method proposed in Lloyd (2005) which involves the so called receiver operating characteristic (ROC) curves. In particular, let $G(t) = Pr(T \geq t)$ be the survivor function of a general test statistic T, and $c = \inf\{t : G(t) \leq \alpha\}$ be the critical value, then ROC curves can be formulated by plotting the power $G_1(c)$ against the size $G_0(c)$ for various values of the critical value c. Note that with $G_0(t)$ we denote the distribution of the test statistic under the null hypothesis and with $G_1(t)$ under the alternative.

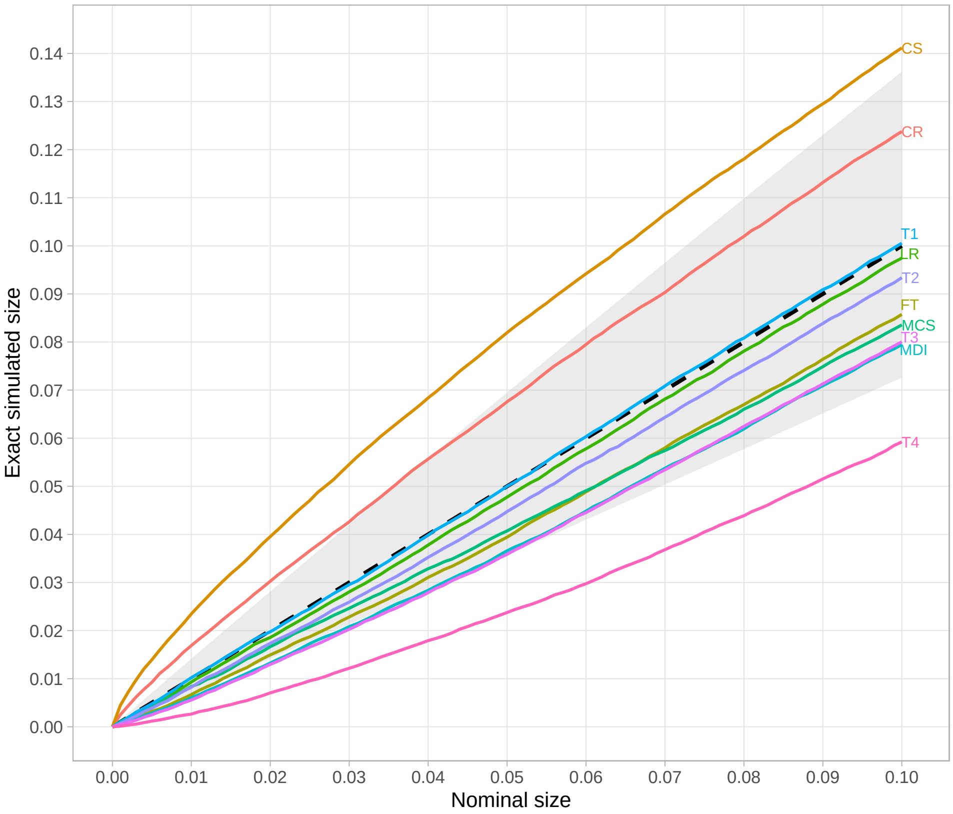
Results are presented in Figure 2.13, from where we can observe that under the adjustment the test statistics have similar behaviour in terms of power for both cases of no contamination and contamination from the $\Gamma(1)$, with the performance being downsized in the latter case. Note also, that results under the adjustment differs from those of the preceding analysis. In particular, we can see that even though from Figure 2.11 we derived the conclusion that the best results arise for small values of α_1 and large values of α_2 in the no contamination case, T_1 has the worst performance among all the BHHJ tests under the adjustment in size. Similar conclusions can be derived for the classical tests. For example, CS and CR have the worst performance among the classical tests under the adjustment, although the results in Table 2.5 indicate the opposite. This behaviour is explained from the fact that the power of the test is highly affected from its liberality or not, making the adjustment in size mandatory before proceeding to the comparison.

In addition, taking into account the results of Figure 2.13, we focus our interest in the following four tests, two from each family, $T4$, MCS, $T3$, and MDI, which appear to have the best performance in terms of power. Note that, even though MDI and $T3$ follow closely each other in terms of size, $T3$ appears to perform better in terms of power. Additionally, we can see that the performance of $T3$ in terms of power follows closely the performance of MCS and especially when the alternative distribution is contaminated from the null. Although, $T4$ appears to have the best performance among all competing tests in terms of power, we should only consider it when the null distribution is contaminated from a distribution which is neither far nor close to the null, since in every other case the exact simulated size fails to satisfy Dale's criterion.

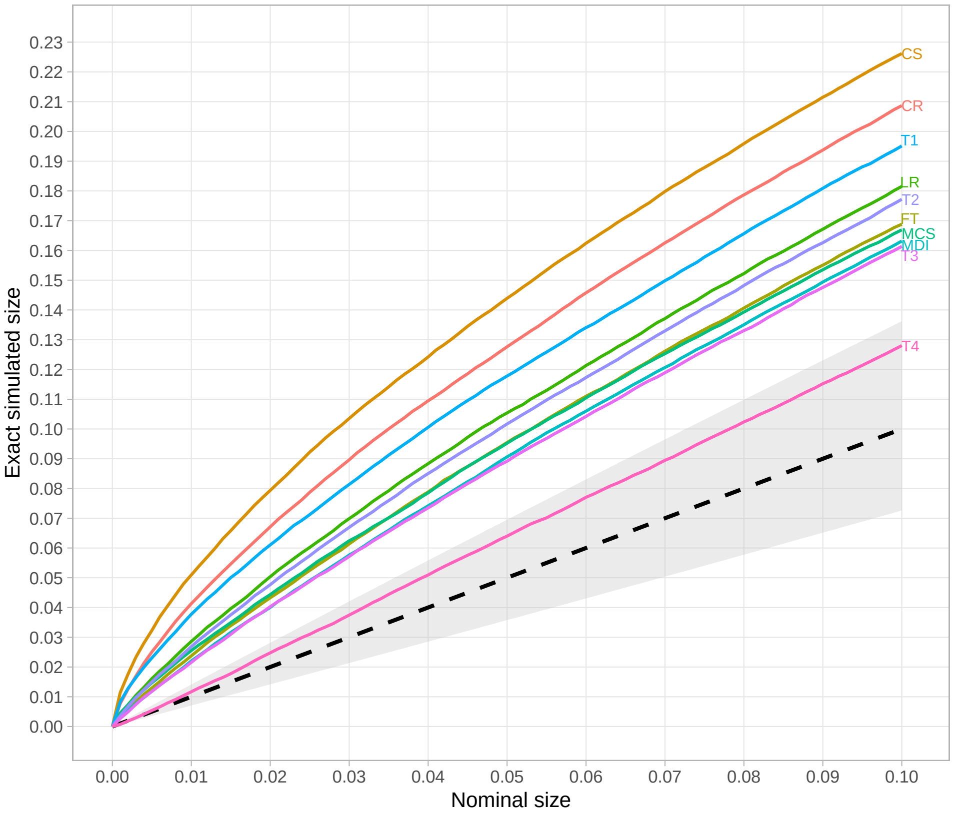
Based on the conducted analysis with regard to the two families of estimators and test statistics, namely the BHHJ and the Cressie−Read families, we can state the following remarks.



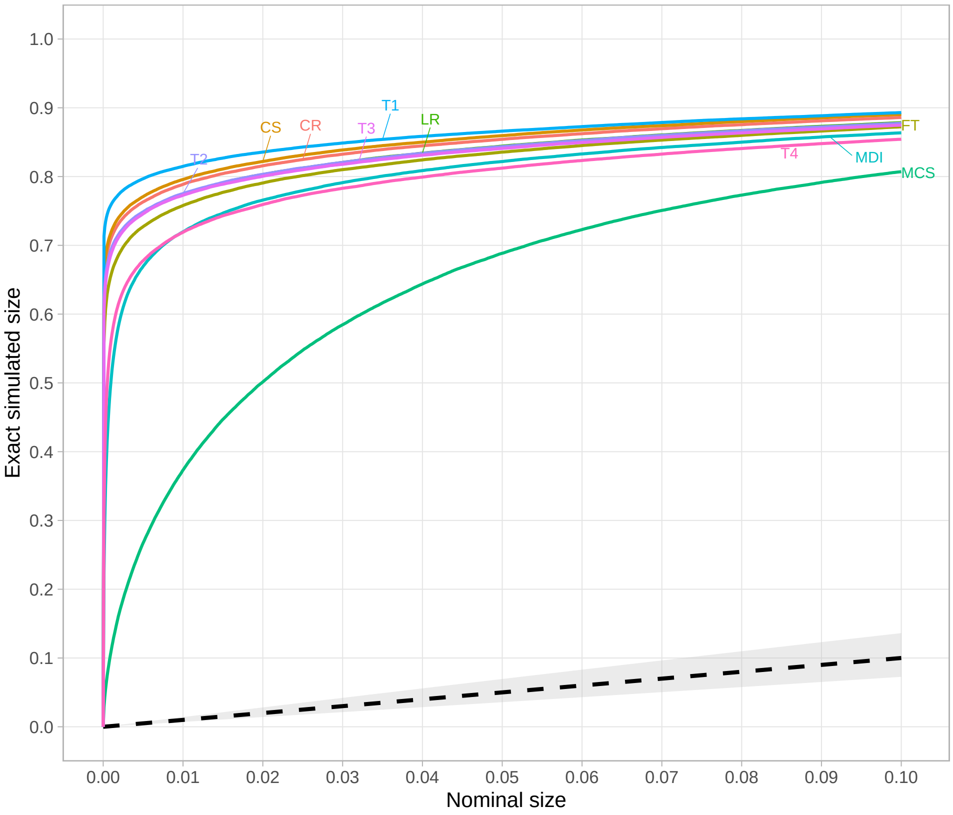
(a) $\Gamma(1)$



(b) $0.925\Gamma(1) + 0.075\Gamma(1.5)$

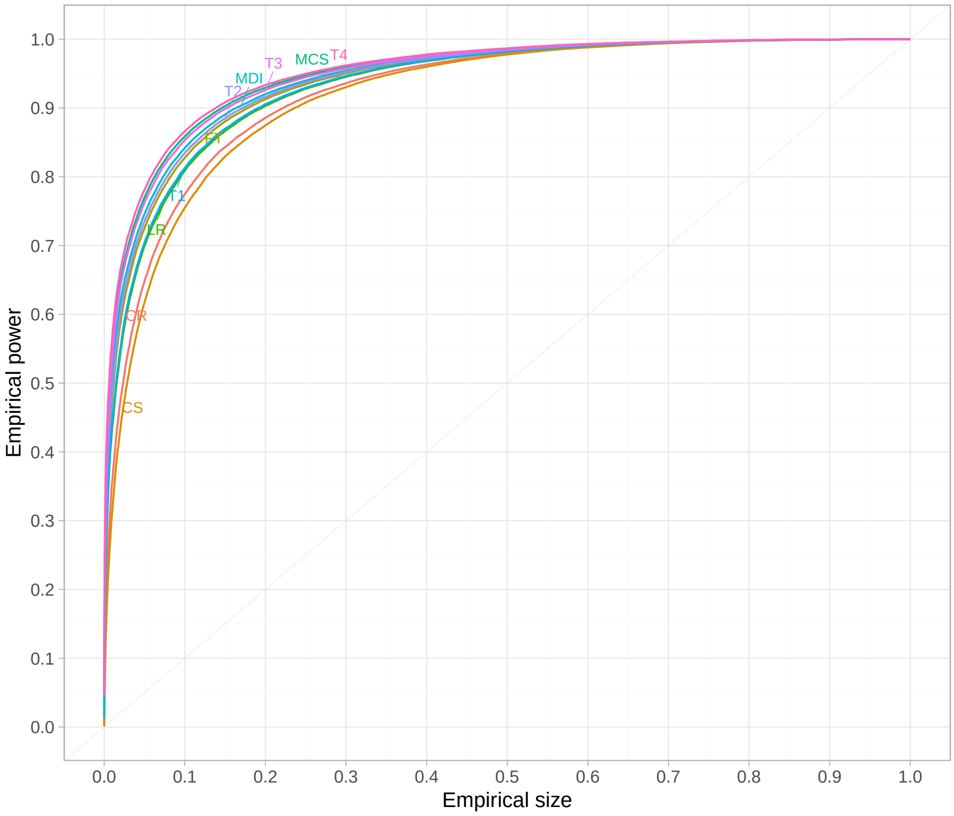


(c) $0.925\Gamma(1) + 0.075\Gamma(4)$

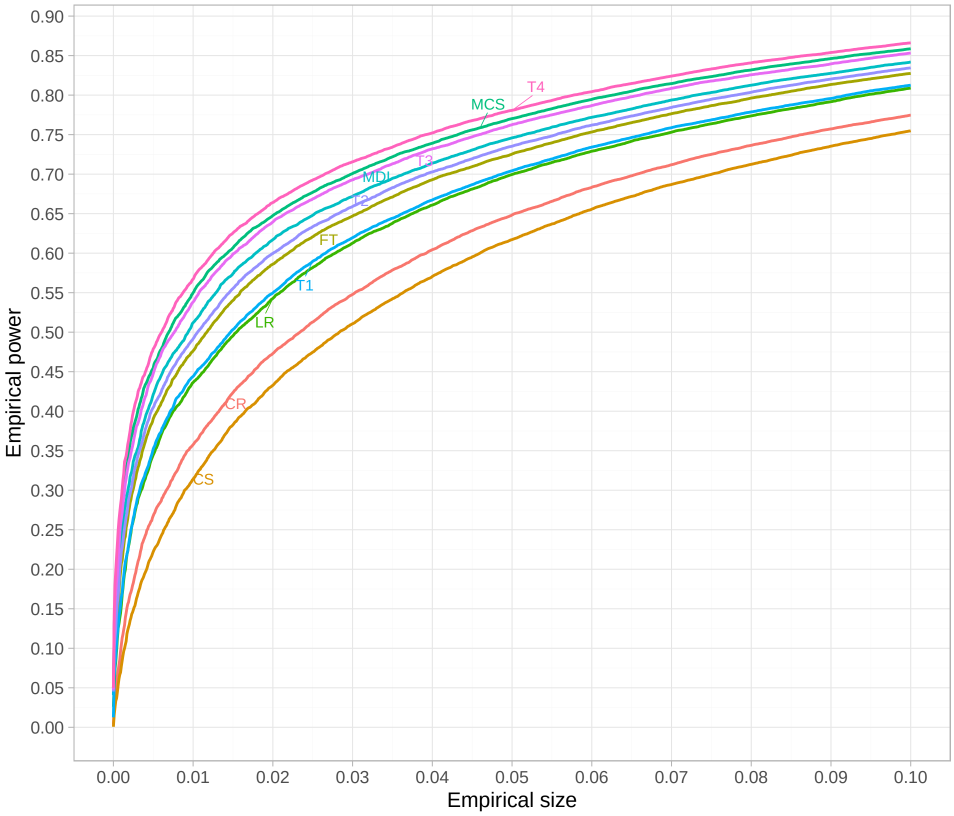


(d) $0.925\Gamma(1) + 0.075\Gamma(10)$

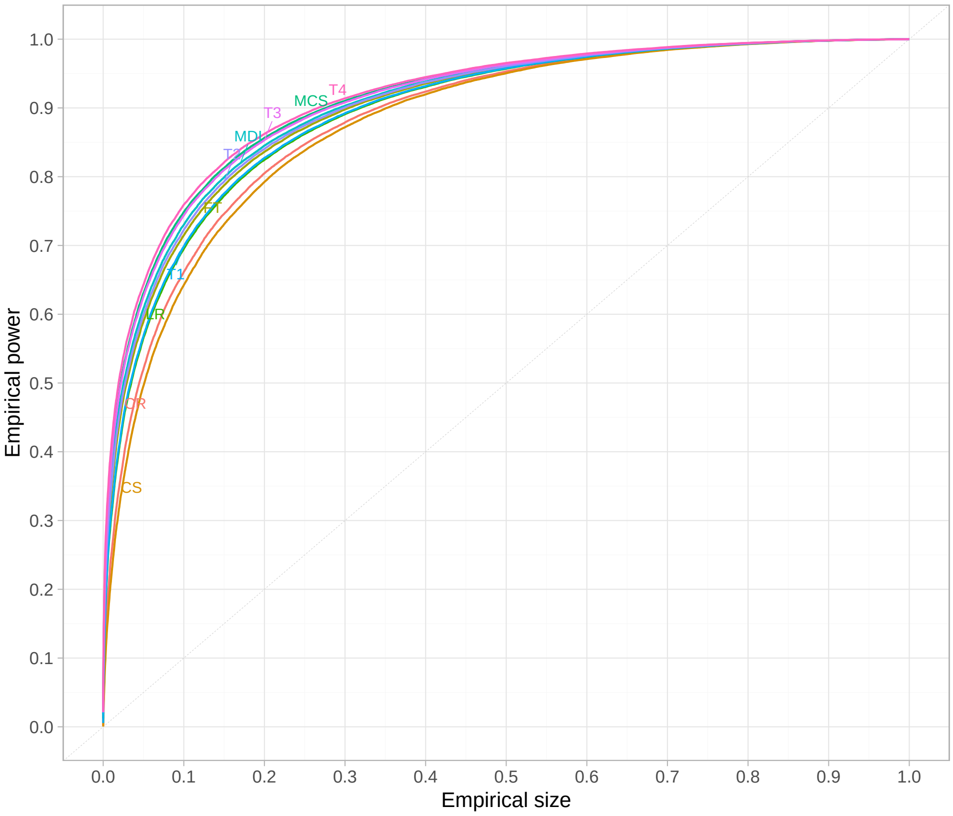
Figure 2.12: Exact simulated sizes against nominal sizes for the four cases of contamination. Gray area depicts the range of exact simulated sizes in which Dale's criterion is satisfied.



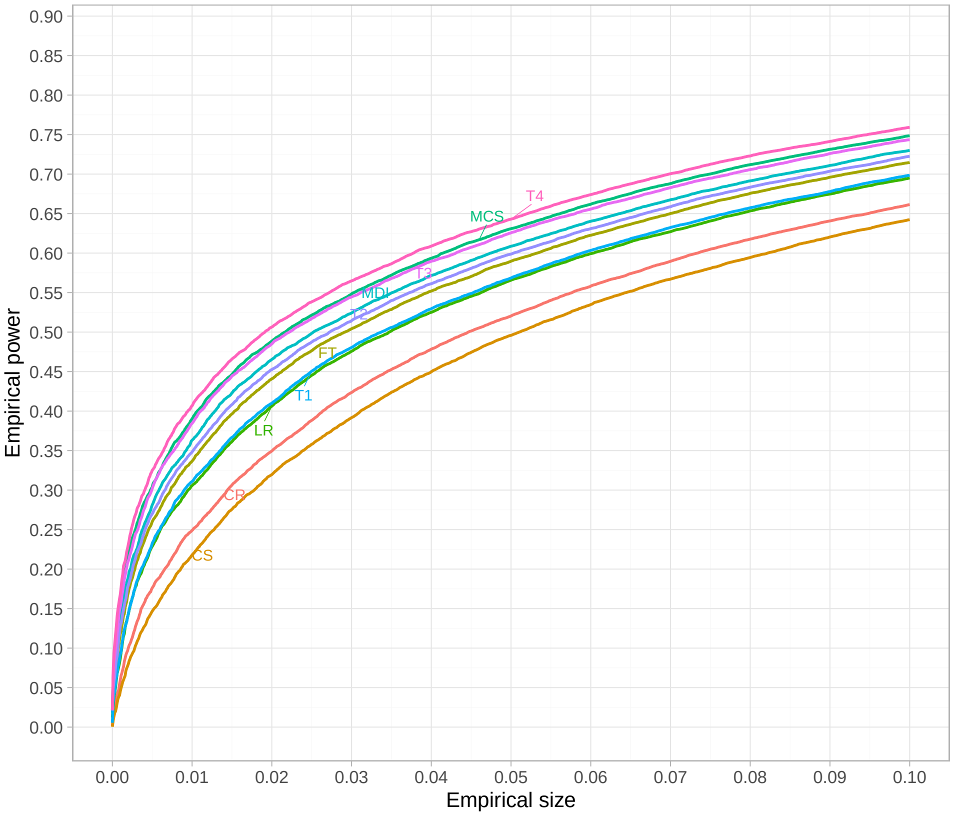
(a) $\Gamma(1.5)$



(b) $\Gamma(1.5)$ magnified



(c) $0.925\Gamma(1.5) + 0.075\Gamma(1)$



(d) $0.925\Gamma(1.5) + 0.075\Gamma(1)$ magnified

Figure 2.13: (a),(c): Empirical ROC curves for the no contamination & contamination from $\Gamma(1)$ cases. (b),(d): The same curves magnified over a relevant range of empirical sizes.

For estimation purposes, under contamination, the best estimators arise for large values of the index α_2 and small negative values of the parameter λ, while the opposite is true when there is no contamination. In relation to testing procedures, when the null distribution is not contaminated or is contaminated from a distribution that is close to it, the best test statistics from the BHHJ family arise for values of the indices α_1 and α_2 close to 0.50, say between 0.40 and 0.60, while the most prominent members of the Cressie−Read family arise for values of $\lambda \in [-2, -1]$. In the case where the contaminant distribution lies neither too close nor too far from the null, only test statistics that are members of the BHHJ family with large values of α_1 near 0.90 and moderate values of α_2 near 0.30 are appropriate choices.

2.5 Conclusions

In this chapter we propose a general family of test statistics for testing the composite null hypothesis (2.4) based on the the $(\Phi, \alpha)-$power divergence family, where the unknown parameter $\boldsymbol{\theta}$ is estimated by the general minimum $(\Phi, \alpha)-$power divergence estimator given in (2.6).

This general family of test statistics involves two indices, namely α_1 and α_2 and two functions Φ that are not necessarily equal. Under the equiprobable model, independently of the values of these indices, the asymptotic distribution of the proposed test statistic is equal to c times the chi−squared distribution, with c as in (2.16). In the case where the true model is not the equiprobable the resemblance of the asymptotic distribution of the test statistic to the c times chi−squared distribution is achieved for small values of α_1 and α_2, (α_1, $\alpha_2 < 0.5$) and in terms of the Type I error the best results are achieved when additionally the index α_1 is smaller than the index α_2. For such values of the two indices the performance of the proposed methodology is considered to be extremely satisfactory in all cases, irrespectively of the true probabilities of the underlying multinomial distribution. Moreover, a comparison between members of the proposed family of test statistics and the classical ones took place. In this case we have considered not only the equiprobable model but also the symmetric, left skewed and right skewed ones, under some alternatives with same skewness values. Results indicate that members of the proposed family of test statistics can be thought of as worthy counterparts to the classical ones. Finally, we have examined the behaviour of the proposed family of estimators and test statistics under the framework of contamination. In this case, results revealed that the best estimators arise for large values of the index α_2 when the data are contaminated, while the opposite is true under no contamination schemes. Regarding the test statistic, we can conclude that members of the proposed family have a slight edge in relation to the classical ones, mostly in cases where the contaminant distribution lies neither too close nor too far from the null.

Chapter 3

Contingency Table Analysis and Inference via Double Index Measures

In this chapter, we focus on a general family of measures of divergence for estimation and testing with emphasis on conditional independence in cross tabulations. For this purpose, a restricted minimum divergence estimator is used for the estimation of parameters under constraints and a new double index (dual) divergence test statistic is introduced and thoroughly examined. The associated asymptotic theory is provided and the advantages and practical implications are explored via simulation studies. The main results presented in this chapter has been published in Meselidis & Karagrigoriou (2022)[1].

3.1 Introduction

Let us consider the general case of two m-dimensional multinomial distributions for which each probability depends on an s-dimensional unknown parameter, say $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_s)^\top$. Assume that the underlying true distribution of an m-dimensional multinomial random variable with N experiments, is

$$\mathbf{X} = (X_1, \ldots, X_m)^\top \sim Multi\big(N, \mathbf{p} = (p_1, \ldots, p_m)^\top\big)$$

where $\mathbf{p}$ is, in general, unknown, belonging to the parametric family

$$\mathcal{P} = \Big\{\mathbf{p}(\boldsymbol{\theta}) = (p_1(\boldsymbol{\theta}), \ldots, p_m(\boldsymbol{\theta}))^\top : \boldsymbol{\theta} = (\theta_1, \ldots, \theta_s)^\top \in \Theta \subseteq \mathbb{R}^s\Big\}. \tag{3.1}$$

The sample estimate $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_m)^\top$ of $\mathbf{p}$ is easily obtained by $\hat{p}_i = x_i/N$, where x_i is the observed frequency for the i-th category (or class).

Divergence measures can be used for estimating purposes by minimizing the associated measure. The classical estimating technique is the one obtained from (1.7) for $\alpha = 0$ and $\Phi(u) = \Phi_{KL}(u) = u \log u + 1 - u$. Then, the resulting KL minimization is equivalent to the classical maximization of the likelihood producing the well−known Maximum Likelihood Estimator (MLE, see Pardo, 2006, Section 5.2). In general, the minimization, with respect to the parameter of interest, of the divergence measure, gives rise to the corresponding minimum divergence estimator (see,

---

[1] Meselidis, C. and Karagrigoriou, A. (2022). Contingency Table Analysis and Inference via Double Index Measures, *Entropy*, 24.4, p. 477, DOI: 10.3390/e24040477

e.g., Cressie & Read, 1984; Morales et al., 1995; Meselidis & Karagrigoriou, 2020 and Chapter 2).

Consider the hypothesis

$$H_0 \colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0) \; vs. \; H_1 \colon \mathbf{p} \neq \mathbf{p}(\boldsymbol{\theta}_0), \; \boldsymbol{\theta}_0 = (\theta_{01}, \ldots, \theta_{0s})^\top \in \Theta \subseteq \mathbb{R}^s, \tag{3.2}$$

where $\mathbf{p}$ is the vector of the true but unknown probabilities of the underlying distribution and $\mathbf{p}(\boldsymbol{\theta}_0)$ the vector of the corresponding probabilities of the hypothesized distribution which is unknown and falls within the family of $\mathcal{P}$ with the unknown parameters satisfying in general, certain constraints, e.g., of the form $c(\boldsymbol{\theta}) = 0$, under which the estimation of the parameter will be performed. The purpose of this chapter is twofold: having as a reference the divergence measure given in (1.7), we will first propose a general double index divergence class of measures and make inference regarding the parameter estimators involved. Then, we proceed with the hypothesis problem with the emphasis given to the concept of conditional independence. As in Chapter 2, the innovative idea proposed in this chapter is the duality in choosing among the members of the general class of divergences, one for estimating and one for testing purposes which may not be necessarily, the same. In that sense, we propose a double index divergence test statistic offering the greatest possible range of options, both for the convex function Φ and the indicator value $\alpha > 0$.

Thus, the estimation problem can be examined considering expression (1.7) using a function $\Phi_2 \in F$ and an indicator $\alpha_2 > 0$:

$$d_{\Phi_2}^{\alpha_2}(\mathbf{p}, \mathbf{p}(\boldsymbol{\theta})) = \sum_{i=1}^{m} p_i^{1+\alpha_2}(\boldsymbol{\theta}) \Phi_2 \left(\frac{p_i}{p_i(\boldsymbol{\theta})} \right) \tag{3.3}$$

the minimization of which with respect to the unknown parameter, will produce the restricted minimum (Φ_2, α_2) divergence (rMD) estimator

$$\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)} = \arg \inf_{\boldsymbol{\theta} \in \Theta : c(\boldsymbol{\theta})=0} d_{\Phi_2}^{\alpha_2}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})) \tag{3.4}$$

for some constraints $c(\boldsymbol{\theta}) = 0$. Observe that the unknown vector of underlying probabilities has been replaced by the vector of the corresponding sample frequencies $\hat{\mathbf{p}}$. Then, the testing problem will be based on

$$d_{\Phi_1}^{\alpha_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})\right) = \sum_{i=1}^{m} p_i^{1+\alpha_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}) \Phi_1 \left(\frac{\hat{p}_i}{p_i(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})} \right) \tag{3.5}$$

where $\Phi_1(\cdot)$ and α_1 may be different from the corresponding quantities used for the estimation problem in (3.3). Finally, the duality of the proposed methodology surfaces when the testing problem is explored via the dual divergence test statistic formulated on the basis of the double-α-double-Φ divergence given by

$$d_{\Phi_1}^{\alpha_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})\right) \tag{3.6}$$

where $\Phi_1, \Phi_2 \in F$ and $\alpha_1, \alpha_2 > 0$.

The remaining parts of this chapter are: Section 3.2 presents the formal definition and the asymptotic properties of the rMD estimator (rMDE). Section 3.3 deals with the general testing problem with the use of rMDE. The associated set−up for the case of three−way contingency tables is developed in Section 3.4 with a simulation section emphasizing on the conditional independence of three random variables. We close this chapter with some conclusions.

3.2 Restricted minimum $(\Phi, \alpha)-$power divergence estimator

In what follows, we will provide the formal definition and the expansion of the rMD estimator and prove its asymptotic normality. The assumptions required for establishing the results of this section for the rMD estimator under constraints, are provided below:

Assumption 1.

(A_0) *$f_1(\boldsymbol{\theta}), \ldots, f_\nu(\boldsymbol{\theta})$ are the constraint functions on the s-dimensional parameter $\boldsymbol{\theta}$, $f_k(\boldsymbol{\theta}) = 0$, $k = 1, \ldots, \nu$ and $\nu < s < m-1$;*

(A_1) *There exists a value $\boldsymbol{\theta}_0 \in \Theta$, such that $\mathbf{X} = (X_1, \ldots, X_m)^\top \sim Multi\,(N, \mathbf{p}(\boldsymbol{\theta}_0))$;*

(A_2) *Each constraint function $f_k(\boldsymbol{\theta})$ has continuous second partial derivatives;*

(A_3) *The $\nu \times s$ and $m \times s$ matrices*

$$\mathbf{Q}(\boldsymbol{\theta}_0) = \left(\frac{\partial f_k(\boldsymbol{\theta}_0)}{\partial \theta_j}\right)_{\substack{k=1,\ldots,\nu \\ j=1,\ldots,s}} \quad and \quad \mathbf{J}(\boldsymbol{\theta}_0) = \left(\frac{\partial p_i(\boldsymbol{\theta}_0)}{\partial \theta_j}\right)_{\substack{i=1,\ldots,m \\ j=1,\ldots,s}}$$

are of full rank;

(A_4) *$\mathbf{p}(\boldsymbol{\theta})$ has continuous second partial derivatives in a neighbourhood of $\boldsymbol{\theta}_0$;*

(A_5) *$\boldsymbol{\theta}_0$ satisfies the Birch regularity conditions (see Section 2.2 and Birch (1964)).*

Definition 3.1. *Under assumptions (A_0)–(A_3) the rMD estimator of $\boldsymbol{\theta}_0$ is any vector in Θ, such that*

$$\hat{\boldsymbol{\theta}}^r_{(\Phi,\alpha)} = \arg\inf_{\{\boldsymbol{\theta}\in\Theta\subseteq\mathbb{R}^s:\ f_k(\boldsymbol{\theta})=0, k=1,\ldots,\nu\}} d^\alpha_\Phi(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})). \tag{3.7}$$

In order to derive the decomposition of $\hat{\boldsymbol{\theta}}^r_{(\Phi,\alpha)}$ the Implicit Function Theorem (IFT) is exploited according to which if a function has an invertible derivative at a point then itself is invertible in a neighbourhood of this point but it cannot be expressed in closed form (see e.g. Theorem 2.1).

Theorem 3.1. *Under Assumptions (A_0)–(A_5), the rMD estimator of $\boldsymbol{\theta}_0$ is such that*

$$\begin{aligned}\hat{\boldsymbol{\theta}}^r_{(\Phi,\alpha)} = \boldsymbol{\theta}_0 + \mathbf{H}(\boldsymbol{\theta}_0)\left(\mathbf{B}(\boldsymbol{\theta}_0)^\top\mathbf{B}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2})diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)) \\ + o(\|\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\|),\end{aligned} \tag{3.8}$$

where $\hat{\boldsymbol{\theta}}^r_{(\Phi,\alpha)}$ is unique in a neighbourhood of $\boldsymbol{\theta}_0$ and

$$\mathbf{H}(\boldsymbol{\theta}_0) = I - \left(\mathbf{B}(\boldsymbol{\theta}_0)^\top\mathbf{B}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{Q}(\boldsymbol{\theta}_0)^\top\left(\mathbf{Q}(\boldsymbol{\theta}_0)\left(\mathbf{B}(\boldsymbol{\theta}_0)^\top\mathbf{B}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{Q}(\boldsymbol{\theta}_0)^\top\right)^{-1}\mathbf{Q}(\boldsymbol{\theta}_0),$$

$$\mathbf{B}(\boldsymbol{\theta}_0) = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2})\mathbf{A}(\boldsymbol{\theta}_0),\ while\ \mathbf{A}(\boldsymbol{\theta}_0) = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})\mathbf{J}(\boldsymbol{\theta}_0).$$

Proof. Following the lines of the proof of Theorem 2.2 given in Chapter 2, let V be a neighbourhood of $\boldsymbol{\theta}_0$ on which $\mathbf{p}(\cdot): \Theta \to \mathcal{P} \subset l^m$ has continuous second partial derivatives, where l^m is the interior of the unit cube of dimension m. Let

$$\mathbf{F} = (F_1, \dots, F_{\nu+s})\colon l^m \times \mathbb{R}^{\nu+s} \to \mathbb{R}^{\nu+s}$$

with

$$F_j(\mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = \begin{cases} f_j(\boldsymbol{\theta}), & j = 1, \dots, \nu \\ \dfrac{\partial d^{\alpha}_{\Phi}(\mathbf{p}, \mathbf{p}(\boldsymbol{\theta}))}{\partial \theta_{j-\nu}} + \sum\limits_{k=1}^{\nu} \lambda_k \dfrac{\partial f_k(\boldsymbol{\theta})}{\partial \theta_{j-\nu}}, & j = \nu+1, \dots, \nu+s, \end{cases}$$

where $(\mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = (p_1, \dots, p_m, \lambda_1, \dots, \lambda_\nu, \theta_1, \dots, \theta_s)$ and λ_k, $k = 1, \dots, \nu$ are the coefficients of the constraints.

It holds that

$$F_j(p_1(\boldsymbol{\theta}_0), \dots, p_m(\boldsymbol{\theta}_0), 0, \dots, 0, \theta_{01}, \dots, \theta_{0s}) = 0,\ j = 1, \dots, \nu+s,$$

and by denoting $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_{\nu+s}) = (\lambda_1, \dots, \lambda_\nu, \theta_1, \dots, \theta_s)$, the matrix

$$\frac{\partial \mathbf{F}}{\partial \boldsymbol{\gamma}} = \left(\frac{\partial F_j}{\partial \gamma_k}\right)_{\substack{j=1,\dots,\nu+s \\ k=1,\dots,\nu+s}} = \begin{pmatrix} \mathbf{0}_{\nu\times\nu} & \mathbf{Q}(\boldsymbol{\theta}_0) \\ \mathbf{Q}(\boldsymbol{\theta}_0)^\top & \Phi''(1)\mathbf{B}(\boldsymbol{\theta})^\top \mathbf{B}(\boldsymbol{\theta}) \end{pmatrix}$$

is nonsingular at $(\mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{\theta}) = (\mathbf{p}(\boldsymbol{\theta}_0), \boldsymbol{\gamma}_0) = (p_1(\boldsymbol{\theta}_0), \dots, p_m(\boldsymbol{\theta}_0), 0, \dots, 0, \theta_{01}, \dots, \theta_{0s})$ with $\boldsymbol{\gamma}_0 = (\mathbf{0}_\nu, \boldsymbol{\theta}_0)$.

Using the IFT a neighbourhood U of $(\mathbf{p}(\boldsymbol{\theta}_0), \boldsymbol{\gamma}_0)$ exists, such that $\partial \mathbf{F}/\partial \boldsymbol{\gamma}$ is nonsingular and a unique differentiable function $\boldsymbol{\gamma}^* = (\boldsymbol{\lambda}^*, \boldsymbol{\theta}^*) : A \subset l^m \to \mathbb{R}^{\nu+s}$, such that $\mathbf{p}(\boldsymbol{\theta}_0) \in A$ and $\{(\mathbf{p}, \boldsymbol{\gamma}) \in U \colon \mathbf{F}(\mathbf{p}, \boldsymbol{\gamma}) = 0\} = \{(\mathbf{p}, \boldsymbol{\gamma}^*(\mathbf{p}))\colon \mathbf{p} \in A\}$ and $\boldsymbol{\gamma}^*(\mathbf{p}(\boldsymbol{\theta}_0)) = (\boldsymbol{\lambda}^*(\mathbf{p}(\boldsymbol{\theta}_0)), \boldsymbol{\theta}^*(\mathbf{p}(\boldsymbol{\theta}_0))) = \boldsymbol{\gamma}_0$. By the chain rule and for $\mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0)$ we obtain

$$\frac{\partial \mathbf{F}}{\partial \mathbf{p}(\boldsymbol{\theta}_0)} + \frac{\partial \mathbf{F}}{\partial \boldsymbol{\gamma}_0} \frac{\partial \boldsymbol{\gamma}_0}{\partial \mathbf{p}(\boldsymbol{\theta}_0)} = 0.$$

Then

$$\frac{\partial \boldsymbol{\theta}_0}{\partial \mathbf{p}(\boldsymbol{\theta}_0)} = \begin{pmatrix} \mathbf{E}(\boldsymbol{\theta}_0) \\ \mathbf{W}(\boldsymbol{\theta}_0) \end{pmatrix},$$

where

$$\mathbf{E}(\boldsymbol{\theta}_0) = \Phi''(1) \left(\mathbf{Q}(\boldsymbol{\theta}_0) \left(\mathbf{B}(\boldsymbol{\theta}_0)^\top \mathbf{B}(\boldsymbol{\theta}_0) \right)^{-1} \mathbf{Q}(\boldsymbol{\theta}_0)^\top \right)^{-1} \times$$
$$\times\, \mathbf{Q}(\boldsymbol{\theta}_0) \left(\mathbf{B}(\boldsymbol{\theta}_0)^\top \mathbf{B}(\boldsymbol{\theta}_0) \right)^{-1} \mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2}) diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})$$

and

$$\mathbf{W}(\boldsymbol{\theta}_0) = \mathbf{H}(\boldsymbol{\theta}_0) \left(\mathbf{B}(\boldsymbol{\theta}_0)^\top \mathbf{B}(\boldsymbol{\theta}_0) \right)^{-1} \mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2}) diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}) \qquad (3.9)$$

since

$$\frac{\partial \mathbf{F}}{\partial \mathbf{p}(\boldsymbol{\theta}_0)} = \begin{pmatrix} \mathbf{0}_{\nu\times m} \\ -\Phi''(1)\mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2}) diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}) \end{pmatrix}.$$

Expanding $\boldsymbol{\theta}^*(\mathbf{p})$ around $\mathbf{p}(\boldsymbol{\theta}_0)$ and using (3.9) gives, for $\boldsymbol{\theta}^*(\mathbf{p}(\boldsymbol{\theta}_0)) = \boldsymbol{\theta}_0$,

$$\begin{aligned}\boldsymbol{\theta}^*(\mathbf{p}) = \boldsymbol{\theta}_0 + \mathbf{H}(\boldsymbol{\theta}_0)\left(\mathbf{B}(\boldsymbol{\theta}_0)^\top \mathbf{B}(\boldsymbol{\theta}_0)\right)^{-1} \mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha/2}) \times \\ \times\, diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)) + o(\|\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)\|).\end{aligned}$$

Since $\hat{\mathbf{p}} \xrightarrow{p} \mathbf{p}(\boldsymbol{\theta}_0)$ eventually $\hat{\mathbf{p}} \in A$ and then $\gamma^*(\hat{\mathbf{p}}) = (\boldsymbol{\lambda}^*(\hat{\mathbf{p}}), \boldsymbol{\theta}^*(\hat{\mathbf{p}}))$ is the unique solution of the system

$$\begin{aligned} &f_k(\boldsymbol{\theta}) = 0, && k = 1, \dots, \nu \\ &\frac{\partial d_\Phi^\alpha(\mathbf{p}, \mathbf{p}(\boldsymbol{\theta}))}{\partial \theta_j} + \sum_{k=1}^{\nu} \lambda_k \frac{\partial f_k(\boldsymbol{\theta})}{\partial \theta_j} = 0, && j = 1, \dots, s \end{aligned}$$

and $(\hat{\mathbf{p}}, \gamma^*(\hat{\mathbf{p}})) \in U$. Hence, $\boldsymbol{\theta}^*(\hat{\mathbf{p}})$ coincides with rMDE $\hat{\boldsymbol{\theta}}^r_{(\Phi,\alpha)}$ given in (3.8). $\square$

The theorem below establishes the asymptotic normality of rMDE which is a straightforward extension of Theorem 2.4 (Meselidis & Karagrigoriou, 2020) since by the Central Limit Theorem we know that

$$\sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)) \xrightarrow[N\to\infty]{L} \mathrm{N}_m(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}) \tag{3.10}$$

with the asymptotic covariance matrix $\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}$ given by $diag(\mathbf{p}(\boldsymbol{\theta}_0)) - \mathbf{p}(\boldsymbol{\theta}_0)\mathbf{p}(\boldsymbol{\theta}_0)^\top$.

Theorem 3.2. *Under Assumptions (A_0)–(A_5), by (3.10) and for $\mathbf{W}(\boldsymbol{\theta}_0)$ given in (3.9), the asymptotic distribution of rMDE is the s-dimensional Normal distribution given by*

$$\sqrt{N}(\hat{\boldsymbol{\theta}}^r_{(\Phi,\alpha)} - \boldsymbol{\theta}_0) \xrightarrow[N\to\infty]{L} \mathrm{N}_s(\mathbf{0}, \mathbf{W}(\boldsymbol{\theta}_0)\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}\mathbf{W}(\boldsymbol{\theta}_0)^\top).$$

Remark 3.1. *The proposed class of estimators forms a family of estimators that goes beyond the indicator α since it is easy to see that estimators obtained for the Csiszar's φ family are given for $\alpha = 0$ in (1.7) and also the standard equiprobable model.*

3.3 Statistical inference

In this section, we introduce the double index divergence test statistic

$$T_{\Phi_1}^{\alpha_1}\left(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}\right) = \frac{2N}{\Phi_1''(1)} d_{\Phi_1}^{\alpha_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})\right) \tag{3.11}$$

with $\Phi_1, \Phi_2 \in F$ and $\alpha_1, \alpha_2 > 0$ and make the additional assumptions by which we focus on the Csiszar's family of measures for testing purposes (the notation φ is used for clarity) and the equiprobable model:

Assumption 2.

(A_6) $p_i = 1/m, \forall i$

(A_7) $\Phi_1 = \varphi, \alpha_1 = 0$.

The theorem below provides the asymptotic distribution of (3.11) under Assumptions (A_0)–(A_7). Assumption (A_7) will be later relaxed and a general asymptotic result will be presented in Subsection 3.3.1 . A discussion about Assumption (A_6) will also be made in the sequel.

Theorem 3.3. *Under Assumptions (A_0)–(A_7) and for the hypothesis in (3.2) we have*

$$T_{\varphi}^{0}\left(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)}\right)=\frac{2N}{\varphi''(1)}d_{\varphi}\left(\hat{\mathbf{p}},\mathbf{p}(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)})\right)\xrightarrow[N\to\infty]{L}\chi^2_{m-1-s+\nu}$$

with $\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)}$ given in (3.8).

Proof. From Birch's third regularity condition (see Section 2.2), we have that

$$\mathbf{p}(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)})=\mathbf{p}(\boldsymbol{\theta}_0)+\mathbf{J}(\boldsymbol{\theta}_0)(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)}-\boldsymbol{\theta}_0)+o(\|\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)}-\boldsymbol{\theta}_0\|)$$

which by Theorem 3.2, expression (3.10), and for $\mathbf{M}(\boldsymbol{\theta}_0)=\mathbf{J}(\boldsymbol{\theta}_0)\mathbf{W}(\boldsymbol{\theta}_0)$ reduces to

$$\mathbf{p}(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)})-\mathbf{p}(\boldsymbol{\theta}_0)=\mathbf{M}(\boldsymbol{\theta}_0)(\hat{\mathbf{p}}-\mathbf{p}(\boldsymbol{\theta}_0))+o_p(N^{-1/2})$$

which implies that

$$\sqrt{N}(\mathbf{p}(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)})-\mathbf{p}(\boldsymbol{\theta}_0))\xrightarrow[N\to\infty]{L}\mathrm{N}(\mathbf{0},\mathbf{M}(\boldsymbol{\theta}_0)\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}\mathbf{M}(\boldsymbol{\theta}_0)^{\top}). \tag{3.12}$$

Combining the above we obtain

$$\sqrt{N}\begin{pmatrix}\hat{\mathbf{p}}-\mathbf{p}(\boldsymbol{\theta}_0)\\ \mathbf{p}(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)})-\mathbf{p}(\boldsymbol{\theta}_0)\end{pmatrix}\xrightarrow[N\to\infty]{L}\mathrm{N}\left(\mathbf{0},\begin{pmatrix}I\\ \mathbf{M}(\boldsymbol{\theta}_0)\end{pmatrix}\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}(I,\mathbf{M}(\boldsymbol{\theta}_0)^{\top})\right)$$

and

$$\sqrt{N}(\hat{\mathbf{p}}-\mathbf{p}(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)}))\xrightarrow[N\to\infty]{L}\mathrm{N}(\mathbf{0},\mathbf{L}(\boldsymbol{\theta}_0)),$$

where

$$\mathbf{L}(\boldsymbol{\theta}_0)=\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}-\mathbf{M}(\boldsymbol{\theta}_0)\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}-\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}\mathbf{M}(\boldsymbol{\theta}_0)^{\top}+\mathbf{M}(\boldsymbol{\theta}_0)\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}\mathbf{M}(\boldsymbol{\theta}_0)^{\top}. \tag{3.13}$$

The expansion of $d_{\varphi}(\mathbf{p},\mathbf{q})$ around $(\mathbf{p}(\boldsymbol{\theta}_0),\mathbf{p}(\boldsymbol{\theta}_0))$ yields

$$T_{\varphi}^{0}\left(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)}\right)=\sum_{i=1}^{m}\frac{N}{p_i(\boldsymbol{\theta}_0)}\left(\hat{p}_i-p_i(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)})\right)^2+o_p(1)=\mathbf{X}^{\top}\mathbf{X}+o_p(1),$$

where

$$\mathbf{X}=\sqrt{N}diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\hat{\mathbf{p}}-\mathbf{p}(\hat{\boldsymbol{\theta}}^{r}_{(\Phi_2,\alpha_2)}))\xrightarrow[N\to\infty]{L}\mathrm{N}(\mathbf{0},\mathbf{T}(\boldsymbol{\theta}_0)).$$

Then, under (A_6), $\mathbf{T}(\boldsymbol{\theta}_0)$ (see (3.13)) is a projection matrix of rank $m-1-s+\nu$. Indeed, under (A_6), we have that

- $diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha_2})=m^{-\alpha_2}I_{m\times m}$,
- $\mathbf{H}(\boldsymbol{\theta}_0)=\mathbf{H}^*(\boldsymbol{\theta}_0)$, with $\mathbf{H}^*(\boldsymbol{\theta}_0)$ as is given for relation (1.24),
- $\mathbf{B}(\boldsymbol{\theta}_0)=m^{-\alpha_2/2}\,\mathbf{A}(\boldsymbol{\theta}_0)$,
- $\left(\mathbf{B}(\boldsymbol{\theta}_0)^{\top}\mathbf{B}(\boldsymbol{\theta}_0)\right)^{-1}=m^{\alpha_2}\left(\mathbf{A}(\boldsymbol{\theta}_0)^{\top}\mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1}$

and thus

$$\begin{aligned}\mathbf{T}(\boldsymbol{\theta}_0) = I_{m\times m} - \mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}))^\top - \mathbf{A}(\boldsymbol{\theta}_0)\left(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0)\right)^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^\top \\ + \mathbf{A}(\boldsymbol{\theta}_0)(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0))^{-1}\mathbf{Q}(\boldsymbol{\theta}_0)^\top\left(\mathbf{Q}(\boldsymbol{\theta}_0)(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0))^{-1}\mathbf{Q}(\boldsymbol{\theta}_0)^\top\right)^{-1}\mathbf{Q}(\boldsymbol{\theta}_0)\times \\ \times(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0))^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^\top.\end{aligned}$$

Also

$$\begin{aligned}trace(\mathbf{T}(\boldsymbol{\theta}_0)) &= trace(I_{m\times m}) - trace\left(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2}))^\top\right) \\ &\quad - trace\left((\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0))^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0)\right) \\ &\quad + trace\Bigg(\left(\mathbf{Q}(\boldsymbol{\theta}_0)(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0))^{-1}\mathbf{Q}(\boldsymbol{\theta}_0)^\top\right)^{-1}\times \\ &\qquad\qquad \times\mathbf{Q}(\boldsymbol{\theta}_0)(\mathbf{A}(\boldsymbol{\theta}_0)^\top\mathbf{A}(\boldsymbol{\theta}_0))^{-1}\mathbf{A}(\boldsymbol{\theta}_0)\mathbf{Q}(\boldsymbol{\theta}_0)^\top\Bigg) \\ &= m-1-s-\nu.\end{aligned}$$

Then, the result follows from the fact (see e.g. Lemma 2.4 or Ferguson (1996, p. 57)) that $\mathbf{X}^\top\mathbf{X}$ has a chi-squared distribution with degrees of freedom equal to the rank of the covariance matrix of the random vector $\mathbf{X}$ as long as it is a projection matrix.

Note that, the asymptotic variance $\mathbf{T}(\boldsymbol{\theta}_0)$ can be easily estimated since the unknown theoretical quantities p_i, $i = 1,\ldots,m$, can be substituted from the relevant sample equivalents. As a result, the computational complexity of the estimator of the asymptotic variance is negligible. □

Remark 3.2. *Relaxation of Assumption (A_6): Arguing as in Meselidis & Karagrigoriou (2020), when the true model is not the equiprobable the result of Theorem 3.3 holds true as long as $\alpha_2 = 0$ and approximately true when $\alpha_2 \to 0$.*

3.3.1 Asymptotic Theory of the Dual Divergence Test Statistic

Having established the two main results of this Chapter, namely the decomposition of the proposed restricted estimator (Theorem 3.1) together with its asymptotic properties (Theorem 3.2), as well as the asymptotic distribution of the associated test statistic under the class of Csiszar φ-functions (Theorem 3.3) we continue below extended in a natural way the results of Chapter 2 for the dual divergence test statistic. The extensions presented in this section are considered vital due to their practical implication on cross tabulations discussed in Section 3.4. In what follows we adopt the following notation:

$$c = m^{-\alpha_1},\ p_{(1)}^{\alpha_1} = \min_{i\in\{1,..m\}} p_i(\boldsymbol{\theta}_0)^{\alpha_1},\ p_{(m)}^{\alpha_1} = \max_{i\in\{1,..m\}} p_i(\boldsymbol{\theta}_0)^{\alpha_1}, k = m-1-s+\nu.$$

Theorem 3.4. *Under Assumptions (A_0)–(A_6) we have*

$$T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}) \xrightarrow[N\to\infty]{L} c\chi^2_k.$$

Proof. A second order Taylor expansion of $d_{\Phi_1}^{\alpha_1}(\mathbf{p},\mathbf{q})$ around $(\mathbf{p}(\boldsymbol{\theta}_0),\mathbf{p}(\boldsymbol{\theta}_0))$ is given by

$$d_{\Phi_1}^{\alpha_1}\left(\hat{\mathbf{p}},\mathbf{p}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})\right) = \frac{\Phi_1''(1)}{2}\sum_{i=1}^{m}\frac{p_i(\boldsymbol{\theta}_0)^{\alpha_1}}{p_i(\boldsymbol{\theta}_0)}\left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})\right)^2 + o\big(\|\hat{\mathbf{p}}-\mathbf{p}(\boldsymbol{\theta}_0)\|^2\big) + o\big(\|\mathbf{p}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})-\mathbf{p}(\boldsymbol{\theta}_0)\|^2\big).$$

By (3.10) and (3.12) we have that $No(\|\hat{\mathbf{p}}-\mathbf{p}(\boldsymbol{\theta}_0)\|^2) = o_p(1)$ and $No(\|\mathbf{p}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}) - \mathbf{p}(\boldsymbol{\theta}_0)\|^2) = o_p(1)$, so that

$$T_{\Phi_1}^{\alpha_1}\left(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}\right) = \sum_{i=1}^{m}\frac{Np_i(\boldsymbol{\theta}_0)^{\alpha_1}}{p_i(\boldsymbol{\theta}_0)}\left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})\right)^2 + o_p(1).$$

The result follows from Theorem 3.3. □

Remark 3.3. *Consider the case where Assumption* (A_6) *is relaxed. Then, the asymptotic distribution of the test statistic* $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ *is estimated to be approximately* $c\chi^2_k$ *where*

$$c = \frac{p_{(1)}^{\alpha_1} + p_{(m)}^{\alpha_1}}{2} \tag{3.14}$$

as long as $\alpha_2 = 0$ *or* $\alpha_2 \to 0$. *For further elaboration of this remark we refer to Remarks 2.6 and 2.7.*

Remark 3.4. *Observe that if* $\alpha_1 \to 0$ *then* $c \to 1$ *and the asymptotic distribution becomes* χ^2_k, *while for* α_1 *away from* 0 *the distribution is proportional to* χ^2_k *with proportionality index* $c \neq 1$. *However, for not equiprobable models these statements hold true as long as* α_2 *is close to zero.*

Consider now the hypothesis with contiguous alternatives (McManus, 1991; Neyman, 1937)

$$H_0\colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0) \text{ vs. } H_{1,N}\colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0) + \frac{\mathbf{d}}{\sqrt{N}}, \tag{3.15}$$

where $\mathbf{d}$ is an m-dimensional vector of known real values with components d_i satisfying the assumption $\sum_{i=1}^m d_i = 0$.

Theorem 3.5. *Under Assumptions* (A_0)–(A_6) *and for the hypothesis (3.15) we have*

$$T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}) \xrightarrow[N\to\infty]{L} c\chi^2_k(\boldsymbol{\xi}^\top\boldsymbol{\xi})$$

which represents a non-central chi-squared distribution with k degrees of freedom and non-centrality parameter $\boldsymbol{\xi}^\top\boldsymbol{\xi}$ *for which* $\boldsymbol{\xi} = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(I - \mathbf{J}(\boldsymbol{\theta}_0)\mathbf{W}(\boldsymbol{\theta}_0))\mathbf{d}$.

Proof. Under the contiguous alternative hypotheses (3.15) we have

$$\sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}_N) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{0},\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)}) \text{ and } \sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}(\boldsymbol{\theta}_0)) \xrightarrow[N\to\infty]{L} \mathrm{N}(\mathbf{d},\boldsymbol{\Sigma}_{\mathbf{p}(\boldsymbol{\theta}_0)})$$

so that

$$\sqrt{N}(\hat{\mathbf{p}} - \mathbf{p}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})) \xrightarrow[N\to\infty]{L} \mathrm{N}\big((I - \mathbf{M}(\boldsymbol{\theta}_0))\mathbf{d},\mathbf{L}(\boldsymbol{\theta}_0)\big).$$

Also

$$T_\varphi(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}) = \sum_{i=1}^{m} \frac{N}{p_i(\boldsymbol{\theta}_0)} \left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})\right)^2 + o_p(1) = \mathbf{X}^\top \mathbf{X} + o_p(1)$$

where

$$\mathbf{X} = \sqrt{N} diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(\hat{\mathbf{p}} - \mathbf{p}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})) \xrightarrow[N\to\infty]{L} \mathrm{N}(\boldsymbol{\xi}, \mathbf{T}(\boldsymbol{\theta}_0))$$

with $\mathbf{T}(\boldsymbol{\theta}_0)$ as in the proof of Theorem 3.3 and $\boldsymbol{\xi} = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{-1/2})(I - \mathbf{M}(\boldsymbol{\theta}_0))\mathbf{d}$. Then, it is easy to see that, under the assumption (A_6) at the beginning of Section 3.3, $\mathbf{T}(\boldsymbol{\theta}_0)\boldsymbol{\xi} = \boldsymbol{\xi}$. Indeed, under the equiprobable model, we have that $(\mathbf{B}(\boldsymbol{\theta}_0)^\top \mathbf{B}(\boldsymbol{\theta}_0))^{-1}$ $= (1/m)^{-1}(\mathbf{A}(\boldsymbol{\theta}_0)^\top \mathbf{A}(\boldsymbol{\theta}_0))^{-1}$ and $\mathbf{B}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{\alpha_2/2}) = \mathbf{A}(\boldsymbol{\theta}_0)^\top (1/m)^{\alpha_2}$ thus

$$\boldsymbol{\xi} = diag(\mathbf{p}(\boldsymbol{\theta}_0)^{1/2})\mathbf{d} - \mathbf{A}(\boldsymbol{\theta}_0)\mathbf{H}^*(\boldsymbol{\theta}_0)(\mathbf{A}(\boldsymbol{\theta}_0)^\top \mathbf{A}(\boldsymbol{\theta}_0))^{-1}\mathbf{A}(\boldsymbol{\theta}_0)^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{1/2})\mathbf{d}.$$

The result follows from the fact that $\sqrt{\mathbf{p}(\boldsymbol{\theta}_0)}^\top diag(\mathbf{p}(\boldsymbol{\theta}_0)^{1/2})\mathbf{d} = 0$ and $\sqrt{\mathbf{p}(\boldsymbol{\theta}_0)}^\top$ $\mathbf{A}(\boldsymbol{\theta}_0) = 0$. Then from Lemma 2.8 we conclude that

$$\sum_{i=1}^{m} \frac{N}{p_i(\boldsymbol{\theta}_0)} \left(\hat{p}_i - p_i(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})\right)^2 \xrightarrow[N\to\infty]{L} \chi^2_{m-1-s+\nu}(\boldsymbol{\xi}^\top \boldsymbol{\xi}).$$

Finally, by using the same arguments as in Theorem 3.4, the distribution of the test statistic (3.11) under (3.15) is $c\chi^2_{m-1-s+\nu}(\boldsymbol{\xi}^\top \boldsymbol{\xi})$ with $c = m^{-\alpha_1}$. □

Remark 3.5. *Observe that under Assumption* (A_6) *(*$p_i = 1/m$*) the asymptotic distribution is independent of* Φ, α_1 *and* α_2*. As a result, the associated power of the test is* $\Pr(\chi^2_k(\boldsymbol{\xi}^\top \boldsymbol{\xi}) \geq \chi^2_{k,a})$ *where a the* $100(1-a)\%$ *percentile of the distribution. If assumption* A_6 *is relaxed then the distribution is approximately non-central chi-squared with proportionality index* $c = \frac{p_{(1)}^{\alpha_1} + p_{(m)}^{\alpha_1}}{2}$*.*

3.4 Cross tabulations and dual divergence test statistic

In this section, we try to take advantage of the methodology proposed earlier for the analysis of cross tabulations. In particular we focus on the case of three categorical variables, say X, Y, and Z with corresponding, I, J, and K categories. Then, assume that the probability mass of a realization of a randomly selected subject is denoted by $p_{ijk}(\boldsymbol{\theta}) = Pr(X = i, Y = j, Z = k) > 0$, where here and in what follows $i = 1, \ldots, I$, $j = 1, \ldots, J$, $k = 1, \ldots, K$ unless otherwise stated. The associated probability vector is given as $\mathbf{p}(\boldsymbol{\theta}) = \{p_{ijk}(\boldsymbol{\theta})\}$, where

$$p_{ijk}(\boldsymbol{\theta}) = \begin{cases} \theta_{ijk}, & (i,j,k) \neq (I,J,K) \\ 1 - \sum\limits_{\substack{i=1 \\ (i,j,k)\neq(I,J,K)}}^{I} \sum\limits_{j=1}^{J} \sum\limits_{k=1}^{K} \theta_{ijk}, & (i,j,k) = (I,J,K) \end{cases}$$

and the parameter space as $\Theta = \{\theta_{ijk}, (i,j,k) \neq (I,J,K)\}$. The sample estimator of $p_{ijk}(\boldsymbol{\theta})$ is $\hat{p}_{ijk} = n_{ijk}/N$, where n_{ijk} is the frequency of the corresponding (i,j,k) cell. Note that, even though this set−up has been given previously in Section 1.4.3, we provide it here, once more, for easy access to the reader.

In this set−up the dual divergence test statistics is given as

$$T_{\Phi_1}^{\alpha_1}\left(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}\right) = \frac{2N}{\Phi_1''(1)}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} p_{ijk}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})^{1+\alpha}\Phi_1\left(\frac{\hat{p}_{ijk}}{p_{ijk}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})}\right), \tag{3.16}$$

where $\hat{p}_{ijk}$ as above and the rMD estimator as

$$\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)} = \arg \inf_{\{\boldsymbol{\theta}\in\Theta\subseteq\mathbb{R}^s:\ f_k(\boldsymbol{\theta})=0,k=1,\dots,\nu\}} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} p_{ijk}(\boldsymbol{\theta})^{1+\alpha_2}\Phi_2\left(\frac{\hat{p}_{ijk}}{p_{ijk}(\boldsymbol{\theta})}\right). \tag{3.17}$$

For $\alpha_1, \alpha_2 = 0$ and special cases of the functions Φ_1 and Φ_2, classical restricted minimum divergence estimators and associated test statistics can be derived from (3.17) and (3.16), respectively. For example, for $\alpha_1,\ \alpha_2 = 0$, and $\Phi_1,\ \Phi_2 = \Phi_{KL}$ the likelihood ratio test statistic with the restricted maximum likelihood estimator $(G^2(\hat{\boldsymbol{\theta}}^r))$ can be derived, while for $\Phi_1,\ \Phi_2 = \Phi_\lambda$ and $\lambda = 1$ we obtain the chi−squared test statistic with the restricted minimum chi−squared estimator $(X^2(\hat{\boldsymbol{\theta}}^r_{X^2}))$, where $\Phi_\lambda(u) = (u^{\lambda+1} - u - \lambda(u-1))/(\lambda(\lambda+1)),\ \lambda \neq 0, -1$. For $\Phi_1,\ \Phi_2 = \Phi_\lambda$ and $\lambda = 2/3$ the dual divergence test statistic reduces to the power divergence test statistic with the restricted minimum power divergence estimator $(CR(\hat{\boldsymbol{\theta}}^r_{CR}))$, whereas for $\lambda = -1/2$ reduces to the Freeman–Tukey test statistic with the restricted minimum Freeman–Tukey estimator $(FT(\hat{\boldsymbol{\theta}}^r_{FT}))$.

The hypothesis of conditional independence between X, Y, and Z is given for any triplet i, j, k by

$$H_0 : p_{ijk}(\boldsymbol{\theta}_0) = \frac{p_{i*k}(\boldsymbol{\theta}_0)p_{*jk}(\boldsymbol{\theta}_0)}{p_{**k}(\boldsymbol{\theta}_0)},\ \theta_0 \in \Theta \text{ unknown},$$

where

$$p_{i*k}(\boldsymbol{\theta}_0) = \sum_{j=1}^{J} p_{ijk}(\boldsymbol{\theta}_0),\ \ p_{*jk}(\boldsymbol{\theta}_0) = \sum_{i=1}^{I} p_{ijk}(\boldsymbol{\theta}_0)\ \ \textit{and}\ \ p_{**k}(\boldsymbol{\theta}_0) = \sum_{i=1}^{I}\sum_{j=1}^{J} p_{ijk}(\boldsymbol{\theta}_0).$$

Under the $(I-1)(J-1)K$ constraint functions

$$f_{ijk}(\boldsymbol{\theta}) = p_{11k}(\boldsymbol{\theta})p_{ijk}(\boldsymbol{\theta}) - p_{1jk}(\boldsymbol{\theta})p_{i1k}(\boldsymbol{\theta}) = 0$$

$i = 2,\dots,I,\ j = 2,\dots,J,\ k = 1,\dots,K$ the above H_0 hypothesis with $\boldsymbol{\theta}_0$ unknown, becomes

$$H_0 : \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0),\ \text{for } \boldsymbol{\theta}_0 \in \Theta_0,$$

where $\Theta_0 = \{\boldsymbol{\theta} \in \Theta : f_{ijk}(\boldsymbol{\theta}) = 0,\ i = 2,\dots,I,\ j = 2,\dots,J,\ k = 1,\dots,K\}$.

Remark 3.6. *For practical purposes, the choice of the values of the indices is motivated by the work of Basu et al. (1998) where, in an attempt to achieve a compromise between robustness and efficiency of estimators, they recommended the use of small values in the* $(0,1)$ *region. In the following subsection, our analysis will reconfirm their findings since as it will be seen, values of both indices close to 0 (than to one 1) will be found to be associated with a good performance not only in terms of estimation but also in terms of goodness−of−fit as it will be reflected in the size and the power of the test.*

3.4.1 Simulation study

In this simulation study, we use the rMD estimator and the associated dual divergence test statistic for the analysis of cross tabulations. Specifically, we are going to compare in terms of size and power classical tests with those that can be derived through the proposed methodology, for the problem of conditional independence of three random variables in contingency tables. We test the hypothesis of conditional independence for a $2 \times 2 \times 2$ contingency table, thus in this case we have $m = 8$ probabilities of the multinomial model, $s = 7$ unknown parameters to estimate and two constraint functions ($\nu = 2$) which are given by

$$f_{221}(\boldsymbol{\theta}) = \theta_{111}\theta_{221} - \theta_{121}\theta_{211} \; and \; f_{222}(\boldsymbol{\theta}) = \theta_{112}\Big(1 - \sum_{\substack{i=1\\(i,j,k)\neq(2,2,2)}}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2}\theta_{ijk}\Big) - \theta_{122}\theta_{212}.$$

For a better understanding of the behaviour of the dual divergence test statistic given in (3.16) we compare it with the four classical goodness−of−fit test statistics mentioned earlier in Section 3.4, namely with the $G^2(\hat{\boldsymbol{\theta}}^r)$, $X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$, $CR(\hat{\boldsymbol{\theta}}^r_{CR})$ and $FT(\hat{\boldsymbol{\theta}}^r_{FT})$. The proposed test $T^{\alpha_1}_{\Phi_1}\left(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)}\right)$ is applied for $\Phi_1 = \Phi_{\alpha_1}$, $\Phi_2 = \Phi_{\alpha_2}$ and six different values of α_1 and α_2, 10^{-7}, 0.01, 0.05, 0.10, 0.50, and 1.50, where $\Phi_\alpha(u) = 1 - (1+\alpha^{-1})u + \alpha^{-1}u^{1+\alpha}$. Note that, the critical values used in this simulation study, are the asymptotic critical values based on the asymptotic distribution $c\chi^2_2$ with c as in (3.14) for the double index family of test statistics, and the χ^2_2 for the classical test statistics. For the analysis we used 100,000 simulations and sample sizes equal to $N = 20, 25$ (small sample sizes) and $N = 40, 45$ (moderate sample sizes).

In this study, we have used the model previously considered by Pardo (2010) given by

$$\begin{aligned} p_{111} &= \pi_{111} - \pi_{111}w & p_{211} &= \pi_{211} + \pi_{222}w - \pi_{111}w \\ p_{112} &= \pi_{112} + \pi_{111}w - \pi_{222}w & p_{212} &= \pi_{212} + \pi_{111}w - \pi_{222}w \\ p_{121} &= \pi_{121} + \pi_{222}w & p_{221} &= \pi_{221} + \pi_{222}w - \pi_{111}w \\ p_{122} &= \pi_{122} + \pi_{111}w & p_{222} &= \pi_{222} - \pi_{222}w \end{aligned}$$

where $0 \leq w < 1$ and $\pi_{ijk} = p_{i**} \times p_{*j*} \times p_{**k}$ $i,j,k = 1,2$ with

$$\begin{array}{llll} \pi_{111} = 0.036254 & \pi_{112} = 0.164994 & \pi_{121} = 0.092809 & \pi_{122} = 0.133645 \\ \pi_{211} = 0.092809 & \pi_{212} = 0.133645 & \pi_{221} = 0.237591 & \pi_{222} = 0.108253. \end{array}$$

For $w = 0$ we take the model under the null hypothesis of conditional independence, while for values $w \neq 0$ we take the models under the alternative hypotheses. We considered the following values of $w = 0.00, 0.30, 0.60$, and 0.90. Note that the larger the value of w the more we deviate from the null model. For the simulation study, we used the R software (R Core Team, 2016), while for the constrained optimization the **auglag** function from the **nloptr** package (Johnson, 2014). The syntax is provided in Appendix C.4.

From Table 3.1, we can observe that in terms of size the performance of the test statistic $T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ is adequate for values of $\alpha_1, \alpha_2 \leq 0.5$ both for small and moderate sample sizes. In addition, we can see that for $\alpha_1 \leq 0.10$, $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ appears to be liberal, while for $\alpha_1 \geq 0.5$ appears to be conservative. We also note that the size becomes smaller as α_1 and α_2 increase with $\alpha_1 \geq \alpha_2$. Table 3.2 provides the size of the classical gof test statistics from where we can observe that $CR(\hat{\boldsymbol{\theta}}^r_{CR})$ has the best performance among all competing tests for every sample size. In contrast, $FT(\hat{\boldsymbol{\theta}}^r_{FT})$ has the worst performance among all competing tests and appears to be very liberal. Furthermore, $X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$ appears to be conservative, while $G^2(\hat{\boldsymbol{\theta}}^r)$ appears to be liberal. Note that for $\alpha_1 \in [0.01, 0.5]$ and $\alpha_2 \leq 0.10$, $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ behaves better than the $G^2(\hat{\boldsymbol{\theta}}^r)$ test statistic and its performance is quite close to the performance of the $X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$.

Table 3.1: Size ($w = 0.00$) calculations (%) of the $T_{\Phi_1}^{\alpha_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ test statistic for sample sizes $N = 20, 25, 40, 45$. Sizes that satisfy Dale's criterion are presented in bold.

	α_2											
α_1	10^{-7}	0.01	0.05	0.10	0.50	1.50	10^{-7}	0.01	0.05	0.10	0.50	1.50
	$n = 20$						$n = 25$					
10^{-7}	8.256	8.257	8.260	8.263	9.216	13.856	7.863	7.865	7.878	7.920	8.927	13.192
0.01	8.207	8.206	8.209	8.224	9.224	13.623	7.753	7.754	7.763	7.817	8.797	12.930
0.05	7.896	7.849	7.879	7.886	8.719	12.916	7.340	7.334	7.327	7.350	8.313	12.277
0.10	7.403	7.404	7.378	7.356	8.046	11.994	**6.965**	**6.959**	**6.940**	**6.934**	7.675	11.364
0.50	**3.873**	**3.850**	**3.769**	**3.612**	3.023	4.050	**3.857**	**3.819**	**3.722**	**3.604**	3.191	4.304
1.50	0.920	0.893	0.807	0.758	0.509	0.202	1.046	1.019	0.948	0.885	0.602	0.203
	$n = 40$						$n = 45$					
10^{-7}	7.016	7.016	7.027	7.055	7.887	11.362	**6.858**	**6.858**	**6.870**	**6.908**	7.732	11.099
0.01	6.933	6.933	6.940	6.957	7.778	11.183	**6.760**	**6.760**	**6.770**	**6.805**	7.601	10.941
0.05	**6.590**	**6.589**	**6.580**	**6.593**	7.342	10.505	**6.427**	**6.422**	**6.415**	**6.426**	7.153	10.340
0.10	**6.246**	**6.239**	**6.228**	**6.222**	6.794	9.758	**6.082**	**6.070**	**6.053**	**6.043**	6.612	9.586
0.50	**3.854**	**3.832**	**3.762**	**3.661**	3.367	4.362	**3.813**	**3.789**	**3.716**	**3.635**	3.331	4.269
1.50	1.172	1.160	1.115	1.066	0.760	0.383	1.183	1.170	1.119	1.068	0.773	0.437

In order to examine the closeness of the estimated (true) size ($\hat{\alpha}_n$) to the nominal size $\alpha = 0.05$ we consider the criterion given by Dale (Dale, 1986), for further details see Subsection 2.4.3. Note that in this situation the estimated (true) size is close to the nominal one if $\hat{\alpha}_n \in [0.0357, 0.0695]$ and is presented in Tables 3.1 and 3.2 in bold.

Regarding the proposed test we can see that for small sample sizes the estimated (true) size is close to the nominal for $\alpha_1 \in [0.10, 0.50]$ and $\alpha_2 \leq 0.10$, while for moderate sample sizes for $\alpha_1 \in [10^{-7}, 0.50]$ and $\alpha_2 \leq 0.10$. With reference to the classical gof test statistics we can observe that the size of the $CR(\hat{\boldsymbol{\theta}}^r_{CR})$ is close to the nominal for every sample size, whereas the size of $G^2(\hat{\boldsymbol{\theta}}^r)$ and $X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$ is close only for moderate sample sizes. Finally, we note that the estimated (true) size of $FT(\hat{\boldsymbol{\theta}}^r_{FT})$ fails to be close to the nominal both for small and moderate sample sizes.

In Tables 3.3–3.5, we provide the results regarding the power of the proposed family of test statistics for the three alternatives and sample sizes $N = 20, 25, 40, 45$, while Table 3.2 provides the results regarding the power of the classical gof test statistics. The performance tends to be better as we deviate from the null model and as the sample size increases both for the classical and the proposed tests.

Table 3.2: Size ($w = 0.00$) and power ($w = 0.30, 0.60, 0.90$) calculations (%) for the classical gof test statistics. Sizes that satisfy Dale's criterion are presented in bold.

Sample size	FT	G^2	CR	X^2	FT	G^2	CR	X^2
	$w = 0.00$				$w = 0.30$			
$N = 20$	14.715	8.261	**4.219**	3.140	18.366	9.072	4.200	2.966
$N = 25$	13.664	7.865	**4.333**	3.477	19.674	9.846	4.783	3.646
$N = 40$	11.154	7.016	**4.722**	**4.059**	21.920	12.192	6.935	5.548
$N = 45$	10.787	**6.858**	**4.703**	**4.082**	22.467	12.992	7.471	6.081
	$w = 0.40$				$w = 0.45$			
$N = 20$	29.707	14.936	7.096	4.910	47.859	26.721	13.789	9.704
$N = 25$	35.768	18.966	9.469	7.118	62.810	38.023	20.147	15.296
$N = 40$	48.366	31.513	18.780	15.030	85.773	69.599	47.644	39.481
$N = 45$	50.821	35.381	22.367	18.217	89.108	76.685	57.000	48.451

Table 3.3: Power ($w = 0.30$) calculations (%) of the $T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ test statistic for sample sizes $N = 20, 25, 40, 45$.

	α_2											
α_1	10^{-7}	0.01	0.05	0.10	0.50	1.50	10^{-7}	0.01	0.05	0.10	0.50	1.50
	$N = 20$						$N = 25$					
10^{-7}	9.073	9.072	9.071	9.076	9.993	15.062	9.846	9.846	9.868	9.895	10.924	15.729
0.01	8.990	8.989	8.988	9.006	9.948	14.724	9.630	9.630	9.651	9.727	10.712	15.343
0.05	8.350	8.278	8.340	8.357	9.231	13.819	9.033	9.008	8.990	9.022	9.876	14.332
0.10	7.694	7.696	7.626	7.616	8.273	12.656	8.225	8.216	8.194	8.188	8.890	13.111
0.50	3.751	3.717	3.607	3.418	2.889	4.199	3.797	3.761	3.656	3.581	3.252	4.620
1.50	0.793	0.764	0.676	0.630	0.415	0.163	0.820	0.810	0.756	0.718	0.479	0.158
	$N = 40$						$N = 45$					
10^{-7}	12.192	12.193	12.207	12.231	13.142	17.775	12.992	12.992	13.003	13.052	14.014	18.490
0.01	11.935	11.934	11.942	11.979	12.853	17.387	12.724	12.724	12.730	12.764	13.721	18.148
0.05	11.075	11.075	11.069	11.074	11.844	16.046	11.799	11.786	11.760	11.768	12.628	16.815
0.10	10.072	10.060	10.039	10.022	10.565	14.549	10.747	10.729	10.688	10.669	11.218	15.183
0.50	4.863	4.842	4.743	4.648	4.342	5.815	5.214	5.179	5.078	4.977	4.648	6.116
1.50	0.979	0.970	0.928	0.890	0.662	0.379	1.032	1.019	0.978	0.928	0.693	0.412

Table 3.4: Power ($w = 0.60$) calculations (%) of the $T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ test statistic for sample sizes $N = 20, 25, 40, 45$.

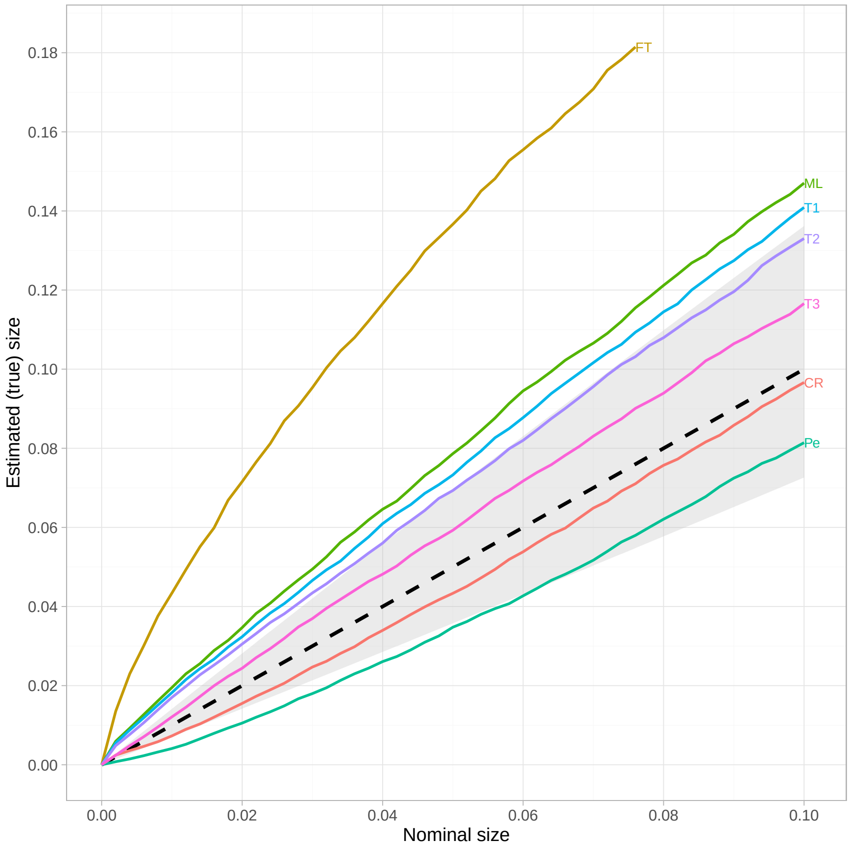
	α_2											
α_1	10^{-7}	0.01	0.05	0.10	0.50	1.50	10^{-7}	0.01	0.05	0.10	0.50	1.50
	$N = 20$						$N = 25$					
10^{-7}	14.928	14.937	14.932	14.944	16.186	22.900	18.965	18.964	19.004	19.042	20.607	27.684
0.01	14.807	14.813	14.808	14.833	16.117	22.486	18.565	18.564	18.598	18.702	20.235	27.069
0.05	13.711	13.583	13.726	13.735	14.939	21.143	17.436	17.383	17.360	17.422	18.733	25.365
0.10	12.612	12.619	12.529	12.525	13.217	19.545	15.794	15.767	15.743	15.726	16.869	23.368
0.50	6.088	5.994	5.811	5.416	4.553	6.403	6.879	6.821	6.656	6.473	5.912	8.458
1.50	1.118	1.077	0.944	0.889	0.553	0.215	1.275	1.240	1.152	1.081	0.729	0.260
	$N = 40$						$N = 45$					
10^{-7}	31.513	31.518	31.533	31.608	33.469	40.799	35.381	35.381	35.404	35.465	37.411	44.556
0.01	30.904	30.903	30.925	30.999	32.868	40.221	34.848	34.845	34.863	34.941	36.744	43.942
0.05	28.949	28.946	28.938	28.956	30.509	37.756	32.727	32.716	32.697	32.715	34.310	41.510
0.10	26.504	26.485	26.434	26.398	27.631	34.747	30.146	30.110	30.051	30.014	31.289	38.456
0.50	11.949	11.867	11.598	11.409	10.830	14.703	14.052	13.966	13.632	13.321	12.731	16.901
1.50	1.797	1.761	1.692	1.578	1.142	0.716	1.973	1.945	1.870	1.776	1.295	0.838

Table 3.5: Power ($w = 0.90$) calculations (%) of the $T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ test statistic for sample sizes $N = 20, 25, 40, 45$.

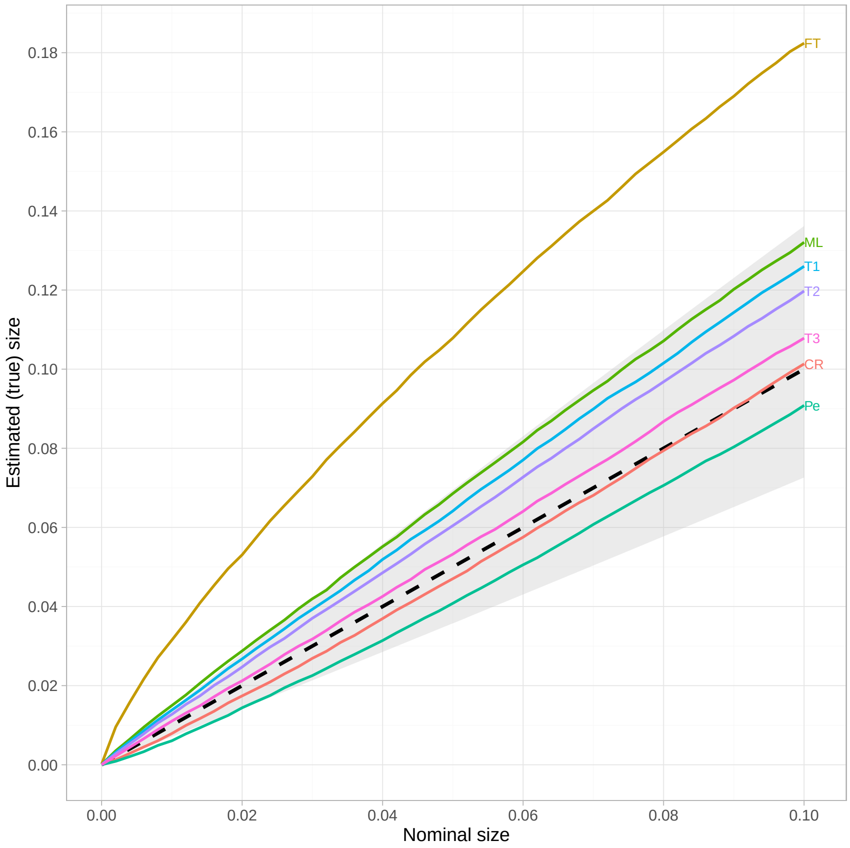
	α_2											
α_1	10^{-7}	0.01	0.05	0.10	0.50	1.50	10^{-7}	0.01	0.05	0.10	0.50	1.50
	$N = 20$						$N = 25$					
10^{-7}	26.712	26.710	26.707	26.711	28.495	37.924	38.017	38.016	38.132	38.191	40.982	50.954
0.01	26.589	26.586	26.585	26.613	28.718	37.421	37.365	37.364	37.456	37.645	40.482	50.206
0.05	25.437	25.267	25.531	25.502	27.170	35.979	35.674	35.559	35.526	35.643	38.260	48.187
0.10	24.287	24.284	24.232	24.172	24.868	33.946	33.014	32.939	32.867	32.854	35.184	45.569
0.50	12.003	11.780	11.424	10.772	8.807	11.665	14.353	14.226	13.870	13.560	12.312	16.886
1.50	1.731	1.662	1.489	1.422	0.904	0.298	2.268	2.226	2.026	1.916	1.387	0.506
	$N = 40$						$N = 45$					
10^{-7}	69.599	69.605	69.637	69.755	72.196	79.363	76.685	76.685	76.731	76.805	78.802	84.683
0.01	68.923	68.923	68.954	69.049	71.518	79.003	76.177	76.173	76.192	76.264	78.143	84.344
0.05	66.310	66.309	66.306	66.365	68.576	77.069	73.760	73.745	73.732	73.766	75.748	82.751
0.10	62.500	62.455	62.372	62.343	64.660	74.161	70.295	70.264	70.144	70.131	72.172	80.319
0.50	30.094	29.904	29.349	28.848	27.895	36.902	36.612	36.465	35.792	35.073	34.056	43.732
1.50	3.748	3.678	3.472	3.210	2.269	1.562	4.349	4.274	4.017	3.747	2.665	1.927

As general comments regarding the behaviour of the proposed and the classical gof test statistics in terms of power we state that the best results for the $T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ are obtained for small values of α_1 in the range $(0, 0.1]$ and large values of α_2 with $\alpha_1 \leq \alpha_2$. Note that, although in terms of power results become better as α_2 increases in terms of size, these are adequate only for $\alpha_2 \leq 0.5$. In addition, we can observe that the performance of $T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ is better than the $CR(\hat{\boldsymbol{\theta}}^r_{CR})$ and $X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$ for every alternative and every sample size for $\alpha_1 \leq 0.1$ and $\alpha_2 \leq 0.5$ and slightly better than $G^2(\hat{\boldsymbol{\theta}}^r)$ for small values of α_1 and large values of α_2, for example for $\alpha_1 = 0.01$ and $\alpha_2 = 0.50$. Furthermore, we can observe that for $\alpha_1 = 0.1$ and $\alpha_2 \leq 0.1$ the size of the test is better than the size of the $G^2(\hat{\boldsymbol{\theta}}^r)$ and slightly worst from the size of the $CR(\hat{\boldsymbol{\theta}}^r_{CR})$ and $X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$ test statistics, while its power is quite better than the power of the $CR(\hat{\boldsymbol{\theta}}^r_{CR})$ and $X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$ and slightly worst than the $G^2(\hat{\boldsymbol{\theta}}^r)$. Additionally, we can see, that as α_1 and α_2 tend to 0, the behaviour of the $T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,\alpha_2)})$ test statistic coincides with the $G^2(\hat{\boldsymbol{\theta}}^r)$ test both in terms of size and power as it was expected.

In order to attain a better insight about the behaviour of the test statistics, we apply Dale's criterion, not only for the nominal size $\alpha = 0.05$, but also for a range of nominal sizes that are of interest. Based on the previous analysis, beside the classical tests, we will focus our interest on the $T^{0.05}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,0.05)})$, $T^{0.10}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,0.10)})$, and $T^{0.20}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,0.20)})$. The following simplified notation is used in every figure, $\mathsf{FT} \equiv FT(\hat{\boldsymbol{\theta}}^r_{FT})$, $\mathsf{ML} \equiv G^2(\hat{\boldsymbol{\theta}}^r)$, $\mathsf{CR} \equiv CR(\hat{\boldsymbol{\theta}}^r_{CR})$, $\mathsf{Pe} \equiv X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$, $\mathsf{T1} \equiv T^{0.05}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,0.05)})$, $\mathsf{T2} \equiv T^{0.10}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,0.10)})$, and $\mathsf{T3} = T^{0.20}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,0.20)})$. From Figure 3.1a, we can see that for small sample sizes $(N = 25)$ $T^{0.20}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,0.20)})$ and $CR(\hat{\boldsymbol{\theta}}^r_{CR})$ satisfy Dale's criterion for every nominal size, while $T^{0.10}_{\Phi_1}(\hat{\boldsymbol{\theta}}^r_{(\Phi_2,0.10)})$ and $X^2(\hat{\boldsymbol{\theta}}^r_{X^2})$ for nominal sizes greater than 0.03 and 0.06, respectively. Note that the dashed line in Figure 3.1 denotes the situation in which the estimated (true) size equals to the nominal size and thus lines that lie above this reference line refer to liberal tests, while those that lie below to conservative ones. On the other hand, for moderate sample sizes $(N = 45)$ all chosen test statistics satisfy Dale's criterion except $FT(\hat{\boldsymbol{\theta}}^r_{FT})$.

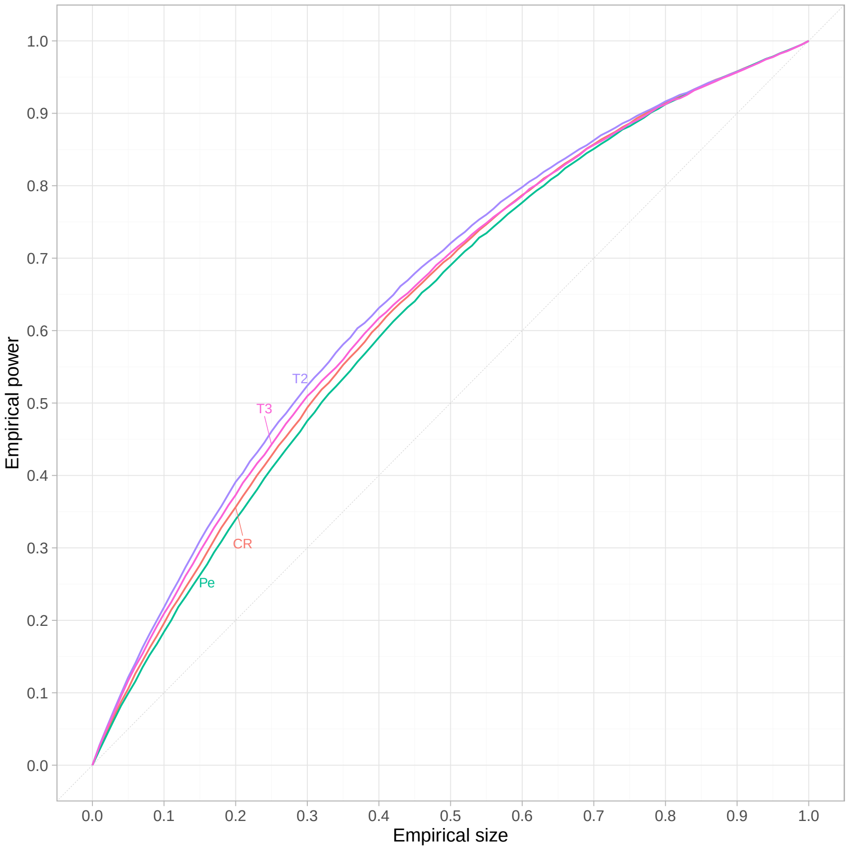


(a)

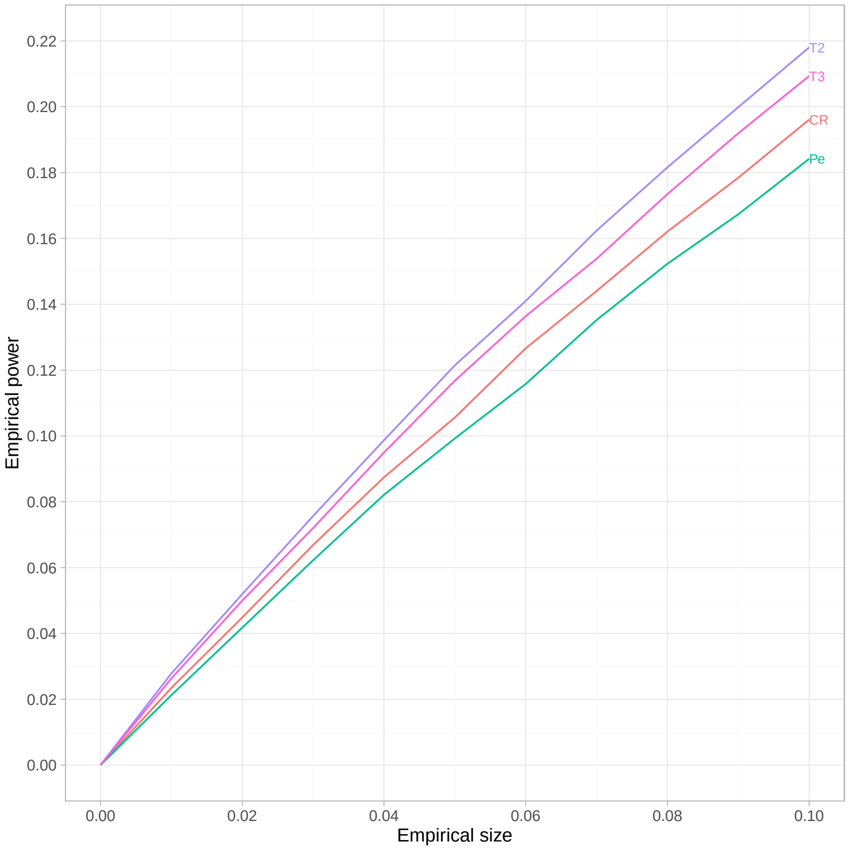


(b)

Figure 3.1: Estimated (true) sizes against nominal sizes. The shaded area refers to Dale's criterion. (a) $N = 25$. (b) $N = 45$.

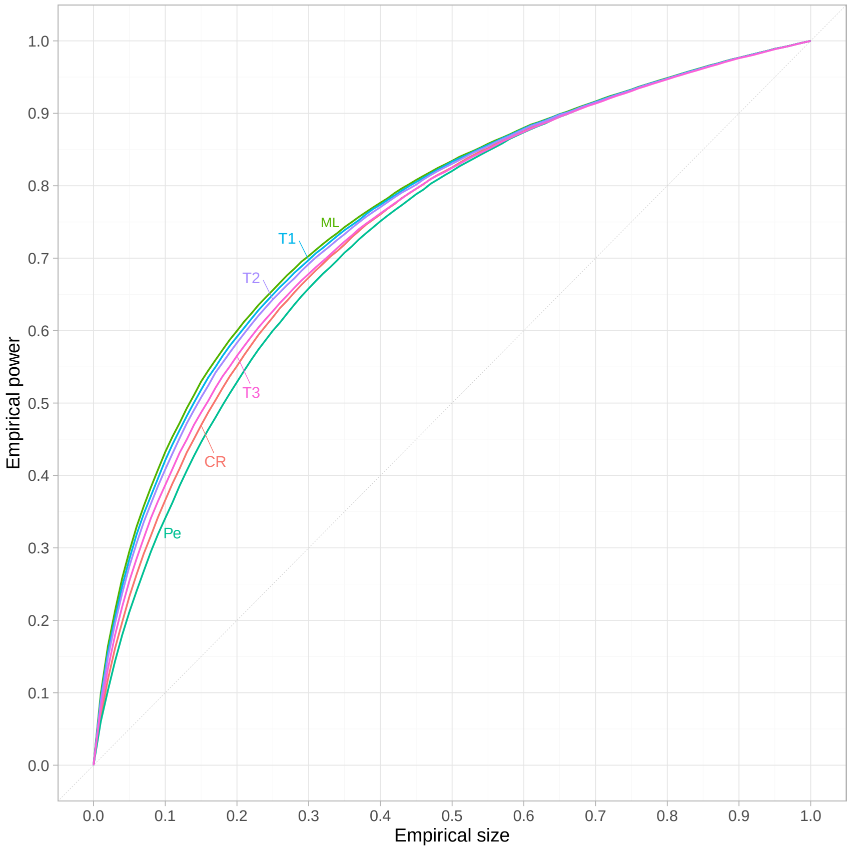


(a)

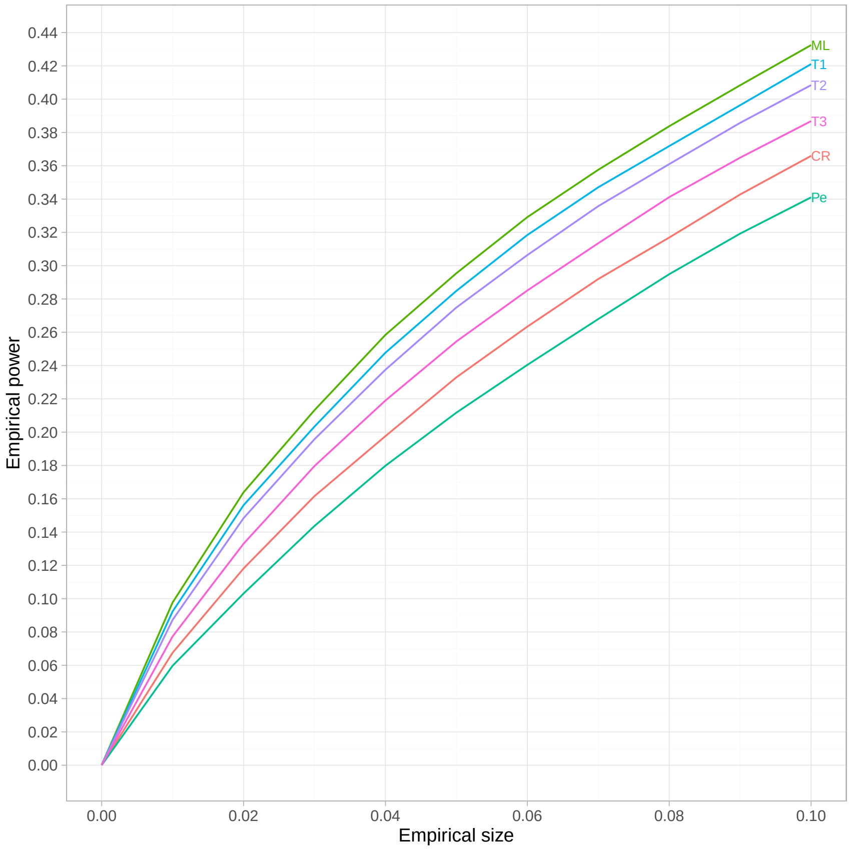


(b)

Figure 3.2: (a) Empirical ROC curves for $N = 25$. (b) The same curves magnified over a relevant range of empirical sizes.



(a)



(b)

Figure 3.3: (a) Empirical ROC curves for $N = 45$. (b) The same curves magnified over a relevant range of empirical sizes.

Taking into account the fact that the actual size of each test differs from the targeted nominal size, we have to make an adjustment in order to proceed further with the comparison of the tests in terms of power. We focus our interest in those tests that satisfy Dale's criterion and follow the method proposed in Lloyd (2005) which involves the so-called receiver operating characteristic (ROC) curves, for further details see Subsection 2.4.3.

Since results are similar for every alternative we restrict ourselves to $w = 0.60$ which refers to an alternative that is neither too close nor too far from the null. For small sample sizes $(N = 25)$ results are presented in Figure 3.2, where we can see that the proposed test is superior from the classical gof test statistics in terms of power. However, for moderate sample sizes $(N = 45)$ we can observe in Figure 3.3 that $G^2(\hat{\boldsymbol{\theta}}^r)$ has the best performance among all competing tests followed by the proposed test-of-fit.

From the conducted analysis we conclude that regarding the proposed test there is a trade off between size and power for different choices of the indices α_1 and α_2. In particular, we can see that as α_1 increases the size becomes smaller in the expense of smaller power, while as α_2 increases the power becomes better and the test more liberal. In conclusion, we could state that for values of α_1 and α_2 in the range $(0.05, 0.25)$ the resulting test statistic provides a fair balance between size and power, which makes it an attractive alternative to the classical gof test statistics, where for small sample sizes larger values of the indices are preferable, whereas for moderate sample sizes, smaller ones are recommended.

Note that, even though in this subsection the focus is placed on the conditional independence, every other model presented in Subsection 1.4.3 is of great importance, both from theoretical and practical perspective. Thus, the behaviour of the rMDE and the associated DITS under the remaining models of Subsection 1.4.3 will be examined in a future work.

3.5 Conclusions

In this chapter, a general divergence family of test statistics is presented for hypothesis testing problems as in (3.2), under constraints. For estimating purposes, we introduce, discuss and use the rMD (restricted minimum divergence) estimator presented in (3.7). The proposed double index (dual) divergence test statistic involves two pairs of elements, namely (Φ_2, α_2) to be used for the estimation problem and (Φ_1, α_1) to be used for the testing problem. The duality refers to the fact that the two pairs may or may not be the same providing the researcher with the greatest possible flexibility.

The asymptotic distribution of the dual divergence test statistic is found to be proportional to the chi-squared distribution irrespectively of the nature of the multinomial model, as long as the values of the two indicators involved are relative close to zero (less than 0.5). Such values are known to provide a satisfactory balance between efficiency and robustness (see, for instance, Basu et al. (1998) or Mattheou & Karagrigoriou (2010)).

The methodology developed in this chapter can be used in the analysis of contingency tables which is applicable in various scientific fields: biosciences, such as genetics (Dubrova et al., 2002) and epidemiology (Znaor et al., 2003); finance, such as the evaluation of investment effectiveness or business performance (Merková, 2015);

insurance science (Geenens & Simar, 2010); or socioeconomics (Bartolucci & Scaccia, 2004). This chapter concludes with a comparative simulation study between classical test statistics and members of the proposed family, where the focus is placed on the conditional independence of three random variables. Results indicate that, by selecting wisely the values of the α_1 and α_2 indices, we can derive a test statistic that can be thought of as a powerful and reliable alternative to the classical gof test statistics especially for small sample sizes.

Chapter 4

The Modified $(\Phi, \alpha)-$Power Divergence Family

In this chapter we focus on the situations where zero frequency cells occur. The modified $(\Phi, \alpha)-$power divergence family (M(Φ, α)PDF) is defined for the first time. Furthermore, we define the minimum modified $(\Phi, \alpha)-$power divergence estimator (MM(Φ, α)PDE) and the associated modified double indexed (dual) test statistic (MDITS). Asymptotic results are obtained under two situations, namely under model misspecification and when the hypothesized parametric model is correctly specified. The behaviour of the proposed family of estimators and test statistics under small and moderate sample sizes is examined, through an extensive simulation study. This chapter has been submitted for publication and is under evaluation.

The rest of this chapter is organized as follows. In Section 4.1 the M(Φ, α)PDF, MM(Φ, α)PDE and the MDITS are defined. Section 4.2 deals with the asymptotic results of the proposed family of estimators and test statiscs. Finally in Section 4.3 an extensive simulation study regarding the behaviour of the MM(Φ, α)PDE and the MDITS takes place.

4.1 Introduction

Following the set$-$up of Alba-Fernández et al. (2018), for a generic probability vector $\mathbf{p}^* = (p_1^*, \dots, p_m^*)$ of length m let $\Delta_{0m} = \{(p_1^*, \dots, p_m^*)^\top : p_i^* \geq 0, i = 1, \dots, m, \sum_{i=1}^m p_i^* = 1\}$ and $\Delta_m = \{(p_1^*, \dots, p_m^*)^\top : p_i^* > 0, i = 1, \dots, m, \sum_{i=1}^m p_i^* = 1\}$. Now let $\mathbf{p} = (p_1, \dots, p_m)^\top \in \Delta_{0m}$ and $\mathbf{q} = (q_1, \dots, q_m)^\top \in \Delta_m$. Then the $(\Phi, \alpha)-$power divergence family given in (1.7) can be expressed in the following form

$$d_\Phi^\alpha(\mathbf{p}, \mathbf{q}) = \sum_{i:p_i>0} q_i^{1+\alpha} \Phi\left(\frac{p_i}{q_i}\right) + \Phi(0) \sum_{i:p_i=0} q_i^{1+\alpha}, \ \alpha > 0, \ \Phi \in F \tag{4.1}$$

which incorporates two components, the second of which is related to the zero frequency cells.

The modified $(\Phi, \alpha)-$power divergence family (M(Φ, α)PDF) is defined as

$$d_{\Phi,h}^\alpha(\mathbf{p}, \mathbf{q}) = \sum_{i:p_i>0} q_i^{1+\alpha} \Phi\left(\frac{p_i}{q_i}\right) + h \sum_{i:p_i=0} q_i^{1+\alpha}, \ \alpha, h > 0, \ \Phi \in F. \tag{4.2}$$

Note that the M(Φ, α)PDF is derived from (4.1) by replacing the fixed value $\Phi(0)$ with the tuning parameter $h > 0$. Also note that we have used the term *modified* rather than *penalized* based on the fact that this family down$-$weights in advance the effect of the empty cells through the index α in relation to the Kullback-Leibler divergence which can be derived for $\alpha = 0$.

Now let $\mathbf{X} = (X_1, \ldots, X_m)^\top$ be a random vector from a multinomial distribution with parameters N and $\mathbf{p}, \mathbf{X} \sim M(N, \mathbf{p})$, and $\mathcal{P} = \{\mathbf{p}(\boldsymbol{\theta}) = (p_1(\boldsymbol{\theta}), \ldots, p_m(\boldsymbol{\theta}))^\top : \boldsymbol{\theta} = (\theta_1, \ldots, \theta_s)^\top \in \Theta\} \subset \Delta_m$ a parametric family, with $\Theta \subseteq \mathbb{R}^s$ $(s < m-1)$. If $\mathbf{p} \in \mathcal{P}$ then there exist a value $\boldsymbol{\theta}_0 \in \Theta$ which leads to the true probabilities of the multinomial model and implies that $d^\alpha_{\Phi,h}(\mathbf{p}, \mathbf{p}(\boldsymbol{\theta}_0)) = 0$. On the other hand, if $\mathbf{p} \in \Delta_{0m}$ we define as $\boldsymbol{\vartheta}_0 \in \Theta$ the value which minimizes the M(Φ, α)PDF between the probability vector of the multinomial model and the parametric family, i.e. $\boldsymbol{\vartheta}_0 = \arg\inf_{\boldsymbol{\theta}\in\Theta} d^\alpha_{\Phi,h}(\mathbf{p}, \mathbf{p}(\boldsymbol{\theta}))$. The sample estimate $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_m)^\top$ of $\mathbf{p}$ is easily obtained by $\hat{p}_i = x_i/N$, where x_i is the observed frequency for the i-th category (or class). Note that, when the postulated model $\mathcal{P}$ has been correctly specified then $\boldsymbol{\vartheta}_0 \equiv \boldsymbol{\theta}_0$.

The minimum modified $(\Phi, \alpha)-$power divergence estimator (MM(Φ, α)PDE) is defined as

$$\hat{\boldsymbol{\theta}}^\alpha_{\Phi,h} = \arg\inf_{\boldsymbol{\theta}\in\Theta} d^\alpha_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})). \tag{4.3}$$

For testing the composite null hypothesis

$$H_0 \colon \mathbf{p} = \mathbf{p}(\boldsymbol{\theta}_0) \ vs. \ H_1 \colon \mathbf{p} \neq \mathbf{p}(\boldsymbol{\theta}_0), \ \boldsymbol{\theta}_0 = (\theta_{01}, \ldots, \theta_{0s})^\top \in \Theta \subseteq \mathbb{R}^s \tag{4.4}$$

where $\boldsymbol{\theta}_0$ is unknown, we introduce the modified double index divergence test statistic (MDITS)

$$T^{\alpha_1}_{\Phi_1,h_1}\left(\hat{\boldsymbol{\theta}}^{\alpha_2}_{(\Phi_2,h_2)}\right) = \frac{2N}{\Phi_1''(1)} d^{\alpha_1}_{\Phi_1,h_1}\left(\hat{\mathbf{p}}, \mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{(\Phi_2,h_2)})\right) \tag{4.5}$$

with $\Phi_1, \Phi_2 \in F$ and $\alpha_1, \alpha_2, h_1, h_2 > 0$.

Note that this family of estimators and test statistics is a general family that includes as special cases for α_1 and α_2 equal to zero and specific forms of the Φ_1 and Φ_2 functions the penalized $\varphi-$divergence family and the penalized power$-$divergence family of estimators and test statistics.

4.2 Asymptotic results

This section is devoted to provide rigorous results regarding the decomposition and the asymptotic distribution of the MM(Φ, α)PDE as well as the asymptotic distribution of the MDITS. To this scope the following assumptions should be taken into consideration.

Assumption 3.

$(A_{4.1})$ $\mathbf{p}(\boldsymbol{\theta})$ *has continuous second partial derivatives in the interior of* Θ.

$(A_{4.2})$ *The minimum of* $d^\alpha_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$ *exists and is unique at* $\boldsymbol{\vartheta}_0$.

$(A_{4.3})$ *Without loss of generality the components of* $\mathbf{p}$ *are sorted such as* $\mathbf{p} = (\mathbf{p}^+, \mathbf{p}^0)$ *where* $\mathbf{p}^+ = (p_1, \ldots, p_r)$ *and* $\mathbf{p}^0 = (p_{r+1}, \ldots, p_m)$ *for some* $1 < r \leq m$ *with* $p_i > 0$ *for* $i = 1, \ldots, r$ *and* $p_i = 0$ *for* $i = r+1, \ldots, m$.

$(A_{4.4})$ $\Delta_{0m}(\Phi, \mathcal{P}, h) = \{\mathbf{p} \in \Delta_{0m}$ *such that Assumption* $(A_{4.2})$ *holds*$\}$.

Note that under Assumption $(A_{4.3})$ we have that

$$d^{\alpha}_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})) = d^{\alpha+}_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})) + hI(r < m), \tag{4.6}$$

where

$$d^{\alpha+}_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})) = \sum_{i=1}^{r} p_i(\boldsymbol{\theta})^{\alpha+1} \Phi_h \left(\frac{p_i}{p_i(\boldsymbol{\theta})} \right) \tag{4.7}$$

with $\Phi_h(x) = \Phi(x) - hI(r < m)$ and $I(r < m)$ equal to 1 if $r < m$ and 0 otherwise. Observe that, from assumption $(A_{4.3})$, when $r = m$ then $p_i > 0$ for every $i = 1, \dots m$. Considering the expression (4.6) it is clear that the value $\boldsymbol{\vartheta}_0$ which minimizes the $d^{\alpha}_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$ minimizes also the $d^{\alpha+}_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta}))$, in other words

$$\arg \inf_{\boldsymbol{\theta} \in \Theta} d^{\alpha}_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})) = \arg \inf_{\boldsymbol{\theta} \in \Theta} d^{\alpha+}_{\Phi,h}(\hat{\mathbf{p}}, \mathbf{p}(\boldsymbol{\theta})).$$

The decomposition of the MM(Φ, α)PDE is derived through the utilization of the Implicit Function Theorem and is given in the theorem below.

Theorem 4.1. *Under the Assumptions $(A_{4.1})$ to $(A_{4.4})$, the MM(Φ, α)PDE of $\boldsymbol{\vartheta}_0$ is such that*

$$\hat{\boldsymbol{\theta}}^{\alpha}_{\Phi,h} = \boldsymbol{\vartheta}_0 + \mathbf{J_F}(\boldsymbol{\vartheta}_0)^{-1} \mathbf{D}(\boldsymbol{\vartheta}_0)^{\top} diag(\mathbf{e})(\hat{\mathbf{p}}^{+} - \mathbf{p}^{+}) + o(\|\hat{\mathbf{p}}^{+} - \mathbf{p}^{+}\|) \tag{4.8}$$

where the $(i, j)^{th}$ element of the $s \times s$ matrix $\mathbf{J_F}$ is

$$\left(\mathbf{J_F}\right)_{i,j} = \sum_{l=1}^{r} \frac{\partial}{\partial \vartheta_i \partial \vartheta_j} p_l(\boldsymbol{\vartheta}_0) v_l(\boldsymbol{\vartheta}_0) + \sum_{l=1}^{r} \frac{\partial}{\partial \vartheta_j} p_l(\boldsymbol{\vartheta}_0) \frac{\partial}{\partial \vartheta_i} p_l(\boldsymbol{\vartheta}_0) w_l(\boldsymbol{\vartheta}_0),\ i, j = 1, \dots, s \tag{4.9}$$

with

$$v_l(\boldsymbol{\vartheta}_0) = (1 + \alpha) p_l(\boldsymbol{\vartheta}_0)^{\alpha} \left[\Phi \left(\frac{p_l}{p_l(\boldsymbol{\vartheta}_0)} \right) - hI(r < m) \right] - p_l(\boldsymbol{\vartheta}_0)^{\alpha - 1} \Phi' \left(\frac{p_l}{p_l(\boldsymbol{\vartheta}_0)} \right) p_l \tag{4.10}$$

and

$$\begin{aligned} w_l(\boldsymbol{\vartheta}_0) = {} & \alpha(1 + \alpha) p_l(\boldsymbol{\vartheta}_0)^{\alpha - 1} \left[\Phi \left(\frac{p_l}{p_l(\boldsymbol{\vartheta}_0)} \right) - hI(r < m) \right] \\ & - 2\alpha p_l(\boldsymbol{\vartheta}_0)^{\alpha - 2} \Phi' \left(\frac{p_l}{p_l(\boldsymbol{\vartheta}_0)} \right) p_l + p_l(\boldsymbol{\vartheta}_0)^{\alpha - 3} \Phi'' \left(\frac{p_l}{p_l(\boldsymbol{\vartheta}_0)} \right) p_l^2. \end{aligned} \tag{4.11}$$

The $(i, j)^{th}$ element of the $r \times s$ matrix $\mathbf{D}$ is $(\mathbf{D})_{i,j} = \frac{\partial}{\partial \vartheta_j} p_i(\boldsymbol{\vartheta}_0)$ $i = 1, \dots, r$, $j = 1, \dots, s$, while the i^{th} element of the $r \times 1$ vector $\mathbf{e} = (e_1, \dots, e_r)^{\top}$ is $e_i = -\ \alpha\ p_i(\boldsymbol{\vartheta}_0)^{\alpha - 1}\ \Phi'(\frac{p_i}{p_i(\boldsymbol{\vartheta}_0)})$ $+\ pi(\boldsymbol{\vartheta}_0)^{\alpha - 2}\ \Phi''(\frac{p_i}{p_i(\boldsymbol{\vartheta}_0)})\ p_i$, $i = 1, \dots, r$.

Proof. Let V be a neighbourhood of $\boldsymbol{\vartheta}_0$ on which $\mathbf{p} \colon \Theta \to \mathcal{P} \subset \Delta_m$ has continuous second partial derivatives. Let

$$\mathbf{F} = (F_1, \dots, F_s) \colon \Delta_{0m}(\Phi, \mathcal{P}, h) \times V \to \mathbb{R}^s$$

with

$$F_j(p_1^*, \dots, p_r^*, \underbrace{0, \dots, 0}_{m-r}, \theta_1, \dots, \theta_s) = \frac{\partial d^{\alpha+}_{\Phi,h}(\mathbf{p}^*, \mathbf{p}(\boldsymbol{\theta}))}{\partial \theta_j},\ j = 1, \dots, s.$$

From Assumption $(A_{4.2})$ it holds that

$$F_j(p_1, \ldots, p_r, \underbrace{0, \ldots, 0}_{m-r}, \theta_{01}, \ldots, \theta_{0s}) = \sum_{l=1}^{r} \frac{\partial p_l(\boldsymbol{\vartheta}_0)}{\partial \theta_j} v_l(\boldsymbol{\vartheta}_0) = 0,\ j = 1, \ldots, s$$

and that the matrix

$$\mathbf{J_F} = \left(\frac{\partial F_j}{\partial \theta_k} \right)_{\substack{j=1,\ldots,s \\ k=1,\ldots,s}}$$

is nonsingular and positive definite at $(\mathbf{p}^+, \boldsymbol{\vartheta}_0)$ with $v_l(\boldsymbol{\vartheta}_0)$ and $\mathbf{J_F}$ as in (4.10) and (4.9) respectively.

Applying the Implicit Function Theorem, there exist a neighbourhood U of $(\mathbf{p}^+, \boldsymbol{\vartheta}_0)$ such that the matrix $\mathbf{J_F}$ is nonsingular and a unique differentiable function $\boldsymbol{\theta}^* : A \subset \Delta_{0m}(\Phi, \mathcal{P}, h) \to \mathbb{R}^s$ such that $\mathbf{p}^{*+} \in A$ and $\{(\mathbf{p}^{*+}, \boldsymbol{\theta}) \in U\colon \mathbf{F}(\mathbf{p}^{*+}, \boldsymbol{\theta}) = 0\} = \{(\mathbf{p}^+, \boldsymbol{\theta}^*(\mathbf{p}^{*+})) : \mathbf{p}^{*+} \in A\}$ and $\boldsymbol{\theta}^*(\mathbf{p}^+) = \boldsymbol{\vartheta}_0$.

By the chain rule and for $\mathbf{p}^* = \mathbf{p}$ we get

$$\frac{\partial \mathbf{F}}{\partial \mathbf{p}^+} + \frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}_0} \frac{\partial \boldsymbol{\theta}_0}{\partial \mathbf{p}^+} = 0.$$

Then

$$\frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}_0} = \mathbf{J_F}(\boldsymbol{\vartheta}_0)^{-1} \mathbf{D}(\boldsymbol{\vartheta}_0)^\top diag(\mathbf{e}) \tag{4.12}$$

since

$$\frac{\partial \mathbf{F}}{\partial \boldsymbol{\theta}_0} = -\mathbf{D}(\boldsymbol{\vartheta}_0)^\top diag(\mathbf{e})$$

with $\mathbf{D}$ and $\mathbf{e}$ as given in the statement of Theorem 4.1.

The Taylor expansion of $\boldsymbol{\theta}^*(\mathbf{p}^{*+})$ around $\mathbf{p}^+$ along with (4.12) and for $\boldsymbol{\theta}^*(\mathbf{p}) = \boldsymbol{\vartheta}_0$ gives

$$\boldsymbol{\theta}^*(\mathbf{p}^{*+}) = \boldsymbol{\vartheta}_0 + \mathbf{J_F}(\boldsymbol{\vartheta}_0)^{-1} \mathbf{D}(\boldsymbol{\vartheta}_0)^\top diag(\mathbf{e})(\mathbf{p}^{*+} - \mathbf{p}^+) + o(\|\mathbf{p}^{*+} - \mathbf{p}^+\|).$$

Since $\hat{\mathbf{p}}^+ \xrightarrow{p} \mathbf{p}^+$ eventually $\hat{\mathbf{p}}^+ \in A$ and then $\boldsymbol{\theta}^*(\hat{\mathbf{p}}^+)$ is the unique solution of the system of equations

$$\frac{\partial d_{\Phi,h}^{\alpha+}\Big(\hat{\mathbf{p}}, \mathbf{p}\big(\boldsymbol{\theta}^*(\hat{\mathbf{p}})\big)\Big)}{\partial \vartheta_j} = 0,\ j = 1, \ldots, s,$$

as well as $(\hat{\mathbf{p}}^+, \boldsymbol{\theta}^*(\hat{\mathbf{p}}^+)) \in U$, from where we can conclude that $\boldsymbol{\theta}^*(\hat{\mathbf{p}}^+)$ is the MM(Φ, α) PDE given in (4.6) □

Taking into account the fact that from the Central Limit Theorem we know that

$$\sqrt{N}(\hat{\mathbf{p}}^+ - \mathbf{p}^+) \xrightarrow[N\to\infty]{L} \mathrm{N}_r(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{p}^+}). \tag{4.13}$$

where $\boldsymbol{\Sigma}_{\mathbf{p}^+} = diag(\mathbf{p}^+) - \mathbf{p}^+(\mathbf{p}^+)^\top$, the theorem below, which is a straightforward extension of Theorem 2.3, establishes the asymptotic distribution of the MM(Φ, α)PDE.

Theorem 4.2. *Under the Assumptions $(A_{4.1})$ to $(A_{4.4})$ the asymptotic distribution of the MM(Φ,α)PDE is the s-dimensional Normal distribution given by*

$$\sqrt{N}(\hat{\boldsymbol{\theta}}^{\alpha}_{\Phi,h} - \boldsymbol{\vartheta}_0) \xrightarrow[N\to\infty]{L} \mathrm{N}_s\big(\mathbf{0}, \boldsymbol{\Sigma}^*\big), \tag{4.14}$$

where

$$\boldsymbol{\Sigma}^* = \mathbf{J}_{\mathbf{F}}(\boldsymbol{\vartheta}_0)^{-1}\mathbf{D}(\boldsymbol{\vartheta}_0)^{\top} diag(\mathbf{e})\boldsymbol{\Sigma}_{\mathbf{p}^+} diag(\mathbf{e})\mathbf{D}(\boldsymbol{\vartheta}_0)\big(\mathbf{J}_{\mathbf{F}}(\boldsymbol{\vartheta}_0)^{-1}\big)^{\top}.$$

Proof. The result follows from Theorem 4.1 and relation (4.13). □

Remark 4.1. *Note that, when $\alpha = 0$ the statements of the above theorems coincide with those of Alba-Fernández et al. (2018). On the other hand, when $\mathbf{p} \in \mathcal{P}$ we derive as particular case, the associated statements given in Chapter 2.*

Under the assumption that the multinomial model is the equiprobable one and that the postulated model is correctly specified, following the lines of Mandal & Basu (2011), the theorem below establishes that the asymptotic distribution of the MDITS is a $c\mathcal{X}^2_{m-1-s}$ with $c = m^{-\alpha_1}$.

Theorem 4.3. *Under the Birch regularity condition, the Assumption $(A_{4.1})$ and assuming further that $\mathbf{p} \in \mathcal{P}$ with $p_i = 1/m$ for every $i = 1, \ldots, m$ we have*

$$T^{\alpha_1}_{\Phi_1,h_1}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}) \xrightarrow[N\to\infty]{L} c\mathcal{X}^2_{m-1-s} \tag{4.15}$$

where $c = m^{-\alpha_1}$.

Proof. Note that when $\mathbf{p} \in \mathcal{P}$ we have that $\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2} \equiv \hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}$, then from Theorem 2.6 we know that

$$T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}) \xrightarrow[N\to\infty]{L} c\mathcal{X}^2_{m-1-s}$$

with $c = m^{-\alpha_1}$. In order to establish the result, it is sufficient to show that

$$R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}) = o_p(1),$$

where

$$R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}) = T^{\alpha_1}_{\Phi_1}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}) - T^{\alpha_1}_{\Phi_1,h_1}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}).$$

We have

$$R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}) = \frac{2N}{\Phi_1''(1)}(\Phi_1(0) - h)\sum_{i=1}^{m} p_i(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})^{1+\alpha_1} I(\hat{p}_i = 0).$$

Then

$$\big|R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})\big| = \frac{2N}{\Phi_1''(1)}\big|\Phi_1(0) - h\big|\sum_{i=1}^{m} p_i(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})^{1+\alpha_1} I(\hat{p}_i = 0)$$

which is equivalent to

$$\big|R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})\big| \leq \frac{2N}{\Phi_1''(1)}\big|\Phi_1(0) - h\big|\sum_{i=1}^{m} I(\hat{p}_i = 0).$$

Taking expectation we have

$$E\Big(\big|R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})\big|\Big) \leq \frac{2N}{\Phi_1''(1)}|\Phi_1(0)-h| \sum_{i=1}^{m} E\Big(I(\hat{p}_i=0)\Big)$$
$$= \frac{2N}{\Phi_1''(1)}|\Phi_1(0)-h| \sum_{i=1}^{m} \Big(1-p_i(\boldsymbol{\theta}_0)\Big)^N. \tag{4.16}$$

As m is finite and $p_i(\boldsymbol{\theta}_0) > 0$ for every $i = 1, \ldots, m$ we get from (4.16)

$$E\Big(\big|R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})\big|\Big) \xrightarrow[N\to\infty]{} 0.$$

Then from Markov's inequality, for every $\epsilon > 0$, we have that

$$Pr\Big(\big|R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})\big| > \epsilon\Big) \leq \frac{E\Big(\big|R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})\big|\Big)}{\epsilon} \xrightarrow[N\to\infty]{} 0.$$

Thus,

$$R_N(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2}) \xrightarrow[N\to\infty]{p} 0$$

and the desired result is established. ☐

Remark 4.2. *Note that, for not equiprobable models the asymptotic distribution of the test statistic is approximated with c as in (2.16).*

4.3 Simulation study

The aim of this simulation study is to investigate the behaviour of the MM(Φ, α)PDE and the MDITS for small and moderate sample sizes as well as to clarify the role of the tuning parameter h regarding the proposed estimating and testing procedures. As data generating mechanism we have considered the following four models, (M_1) truncated geometric model, (M_2) truncated Poisson model, (M_3) truncated negative binomial model, and (M_4) truncated binomial model. The used definition of truncation, is similar to the one given in Bishop et al. (2007), where N independent and identically distributed variates are observed, where the observations are truncated at $k = 3$. Let X_1, X_2, X_3 and X_4 be the number of observations taking on the values 0, 1, 2 and 3 *or more*, respectively. In other words, instead of allowing the r.v. to take each distinct value $\{0, 1, 2, 3, \ldots\}$, the r.v. is truncated so that $X = \{0, 1, 2, \geq 3\}$. Then, in every case, under the null hypothesized model, the random vector is derived from the following multinomial model $(X_1, X_2, X_3, X_4)^\top \sim M(N, \mathbf{p}^{M_i}(\theta_0^{M_i}))$, $i = 1, \ldots, 4$ where

$$\begin{aligned}
\mathbf{p}^{M_1}(\theta_0^{M_1}) &= (p_1^{M_1}(\theta_0^{M_1}), p_2^{M_1}(\theta_0^{M_1}), p_3^{M_1}(\theta_0^{M_1}), p_4^{M_1}(\theta_0^{M_1}))\\
&= (\theta_0, \theta_0(1-\theta_0), \theta_0(1-\theta_0)^2, (1-\theta_0)^3) \text{ and } \theta_0^{M_1} = 0.9,\\
\mathbf{p}^{M_2}(\theta_0^{M_2}) &= (p_2^{M_2}(\theta_0^{M_2}), p_2^{M_2}(\theta_0^{M_2}), p_3^{M_2}(\theta_0^{M_2}), p_4^{M_2}(\theta_0^{M_2}))\\
&= (e^{-\theta_0}, \theta_0 e^{-\theta_0}, \theta_0^2 e^{-\theta_0}/2, 1-e^{-\theta_0}-\theta_0 e^{-\theta_0}-\theta_0^2 e^{-\theta_0}/2) \text{ and } \theta_0^{M_2} = 0.5,\\
\mathbf{p}^{M_3}(\theta_0^{M_3}) &= (p_1^{M_3}(\theta_0^{M_3}), p_2^{M_3}(\theta_0^{M_3}), p_3^{M_3}(\theta_0^{M_3}), p_4^{M_3}(\theta_0^{M_3}))\\
&= ((1-\theta_0)^2, 2\theta_0(1-\theta_0)^2, 3\theta_0^2(1-\theta_0)^2, 1-(1-\theta_0)^2((1+\theta_0)^2+2\theta_0^2))
\end{aligned}$$

and $\theta_0^{M_3} = 0.1$,

$$\begin{aligned}\mathbf{p}^{M_4}(\theta_0^{M_4}) &= (p_1^{M_4}(\theta_0^{M_4}), p_2^{M_4}(\theta_0^{M_4}), p_3^{M_4}(\theta_0^{M_4}), p_4^{M_4}(\theta_0^{M_4})) \\ &= ((1-\theta_0)^3, 3\theta_0(1-\theta_0)^2, 3\theta_0^2(1-\theta_0), \theta_0^3) \text{ and } \theta_0^{M_4} = 0.2.\end{aligned}$$

Note that for notational simplicity we use θ_0 rather than $\theta_0^{M_i}$ in the resulting multinomial model. The associated probability mass functions for which the above models are observed, respectively are

$$\begin{aligned}Pr(X=k) &= (1-\theta)^k\theta,\ k=0,1,2,3,\dots \text{ and } \theta \in (0,1), \\ Pr(X=k) &= \frac{\theta^k e^{-\theta}}{k!},\ k=0,1,2,3,\dots \text{ and } \theta > 0, \\ Pr(X=k) &= \binom{k+r-1}{k}(1-\theta)^r\theta^k,\ r=2, k=0,1,2,3,\dots \text{ and } \theta \in (0,1), \\ Pr(X=k) &= \binom{n}{k}\theta^k(1-\theta)^{n-k},\ n=2, k=0,1,2,3,\dots \text{ and } \theta \in (0,1).\end{aligned}$$

A similar set−up has been used among others by Basu & Basu (1998) and Mandal et al. (2010).

As sample sizes we have used N = 15, 30, 60, 120, 240. Moreover, in order to examine the robustness of the proposed estimators and test statistics we have used the following contaminated model

$$^{*}\mathbf{p}^{M_i} = \Big((1-\frac{\epsilon}{\sqrt{N}})p_1^{M_i}, (1-\frac{\epsilon}{\sqrt{N}})p_2^{M_i}, (1-\frac{\epsilon}{\sqrt{N}})p_3^{M_i}, (1-\frac{\epsilon}{\sqrt{N}})p_4^{M_i} + \frac{\epsilon}{\sqrt{N}}\Big).$$

where the contamination rate is taken as a function of N. We based this option on the contaminated model used in Toma & Broniatowski (2011) and from the fact that fixed amounts of contamination have greater impact in larger samples (see Basu et al., 2010). For every sample size N, as ϵ, we have used the value $\epsilon = 0.1\sqrt{15}$ which is associated with the highest contamination rate, i.e. 0.1, in the case when $N = 15$. For $N = 30$, 60, 120 and 240, the contamination rate is 0.071, 0.05, 0.035 and 0.025, respectively. Each experiment is repeated 10^4 times.

Regarding the examination of the MM(Φ,α)PDE, $\hat{\theta}^{\alpha_2}_{\Phi_2,h_2}$ is applied for $\Phi_2(u) = 1 - (1+\alpha_2^{-1})u + \alpha_2^{-1}u^{1+\alpha_2}$, which leads to the BHHJ−divergence measure, for $\alpha_2 = 0.01$, 0.05, 0.10… (0.10) …1.00 and $h_2 = 0.0 : 0.5 : 80.0$. Note that in this case $\Phi_2(0) = 1$ indicating that the ordinary M(Φ,α)PDE are derived from the MM(Φ,α)PDE for $h_2 = 1$. Our conclusions are based on the root mean squared error ($RMSE$) which is given as,

$$RMSE_{\hat{\theta}^{\alpha_2}_{\Phi_2,h_2}} = \sqrt{\frac{1}{10^4}\sum_{l=1}^{10^4}\left(\hat{\theta}^{\alpha_2}_{(\Phi_2,h_2),l} - \theta_0\right)^2}$$

which is related to the efficiency and the robustness of the estimator when the data are generated from the null models with no contamination data and models under contamination, respectively. Note that in general, it is not computationally intense to calculate the $RMSE$ for one specific member of the MM(Φ,α)PDE family. In this simulation study, a considerable amount of computational time is needed, in order for the results to be derived, since $12 \times 161 = 1932$ combinations of the α_2 index and the h_2 tuning parameter are considered.

For the examination of the size of the MDITS we have considered as $\Phi_1(u) = 1 - (1 + \alpha_1^{-1})u + \alpha_1^{-1}u^{1+\alpha_1}$ and $\Phi_2(u) = 1 - (1 + \alpha_2^{-1})u + \alpha_2^{-1}u^{1+\alpha_2}$, with α_1, $\alpha_2 = 0.01$, 0.05, 0.10 : 0.10 : 1.00 and h_1, $h_2 = 1$, 5, 15, 40. Note that, the selection of the values of the parameters h_1 and h_2 is based on the results regarding the examination of the MM(Φ, α)PDE. The critical values used in this simulation study, are the asymptotic critical values based on the asymptotic distribution $c\chi^2_2$ with $c = \min_i(p_i(\hat{\theta}))/2 + \max_i(p_i(\hat{\theta}))/2$, $i = 1, \ldots, 4$. As nominal level we have considered the 0.05.

For this simulation study, we used the R software (R Core Team, 2016), while for the optimization the **optimise** function from the base package **stats** is implemented. For the manipulation of the data, we used the packages **dplyr** (Wickham et al., 2022) and **data.table** (Dowle & Srinivasan, 2022). Furthermore, for the visualization of the results we used the package **ggplot2** (Wickham, 2016) while we took advantage of the package **future.apply** (Bengtsson, 2021) to activate parallel computing in order to compensate the computational burden. The syntax is provided in Appendix C.5.

The percentage of zero frequency cells occurrence, for the four data generating models, under no contamination and under contamination is presented in Tables 4.1 and 4.2, respectively. In this section, only results regarding the truncated geometric model (M_1) are presented. Results for the remaining models can be found in the Appendix B.

In relation to the efficiency of the MM(Φ, α)PDE, results are presented in Figures 4.1a to 4.5a. To begin with, as we expected (see, Basu et al. (1998)), we can observe that the ordinary BHHJ$-$divergence estimators ($h_2 = 1$) appear to be more efficient as the value of α_2 increases, for every sample size. Moreover, we can observe that the *RMSE* is a convex function of the tuning parameter h_2, where as h_2 increases an improvement in the efficiency is achieved, regardless the value of the index α_2.

Table 4.1: (%) occurrence of zero cell frequencies for the four data generating models under no contamination.

N	X_1	X_2	X_3	X_4	X_1	X_2	X_3	X_4
	Truncated Geometric				Truncated Poisson			
15	0.00	24.83	86.99	98.29	0.00	0.31	31.65	80.85
30	0.00	5.93	76.01	96.90	0.00	0.00	9.35	64.57
60	0.00	0.32	58.01	94.17	0.00	0.00	0.90	42.43
120	0.00	0.00	34.49	88.41	0.00	0.00	0.00	17.72
240	0.00	0.00	11.25	78.57	0.00	0.00	0.00	3.04
	Truncated Negative Binomial				Truncated Binomial			
15	0.00	7.09	68.83	94.57	0.00	0.02	22.58	88.32
30	0.00	0.50	48.92	89.34	0.00	0.00	5.08	78.44
60	0.00	0.00	23.42	80.08	0.00	0.00	0.17	62.13
120	0.00	0.00	5.65	64.15	0.00	0.00	0.00	37.45
240	0.00	0.00	0.25	41.05	0.00	0.00	0.00	14.34

In particular, we can see that specific values of the tuning parameter h_2 lead to almost the same *RMSE* of the estimators, at least for small sample sizes. For example, when

Table 4.2: (%) occurrence of zero frequency cells for the four data generating models under contamination.

N	X_1	X_2	X_3	X_4	X_1	X_2	X_3	X_4
	Truncated Geometric				Truncated Poisson			
15	0.00	28.02	88.70	20.93	0.00	0.75	34.84	16.91
30	0.00	7.26	77.48	10.97	0.00	0.00	11.63	7.43
60	0.00	0.38	59.70	4.54	0.00	0.00	1.19	1.82
120	0.00	0.00	35.54	0.98	0.00	0.00	0.00	0.19
240	0.00	0.00	11.84	0.16	0.00	0.00	0.00	0.00
	Truncated Negative Binomial				Truncated Binomial			
15	0.00	9.33	71.72	20.03	0.01	0.16	25.43	16.28
30	0.00	0.61	50.29	10.20	0.00	0.00	6.26	8.37
60	0.00	0.00	25.07	3.56	0.00	0.00	0.38	2.72
120	0.00	0.00	5.64	0.89	0.00	0.00	0.00	0.42
240	0.00	0.00	0.25	0.10	0.00	0.00	0.00	0.02

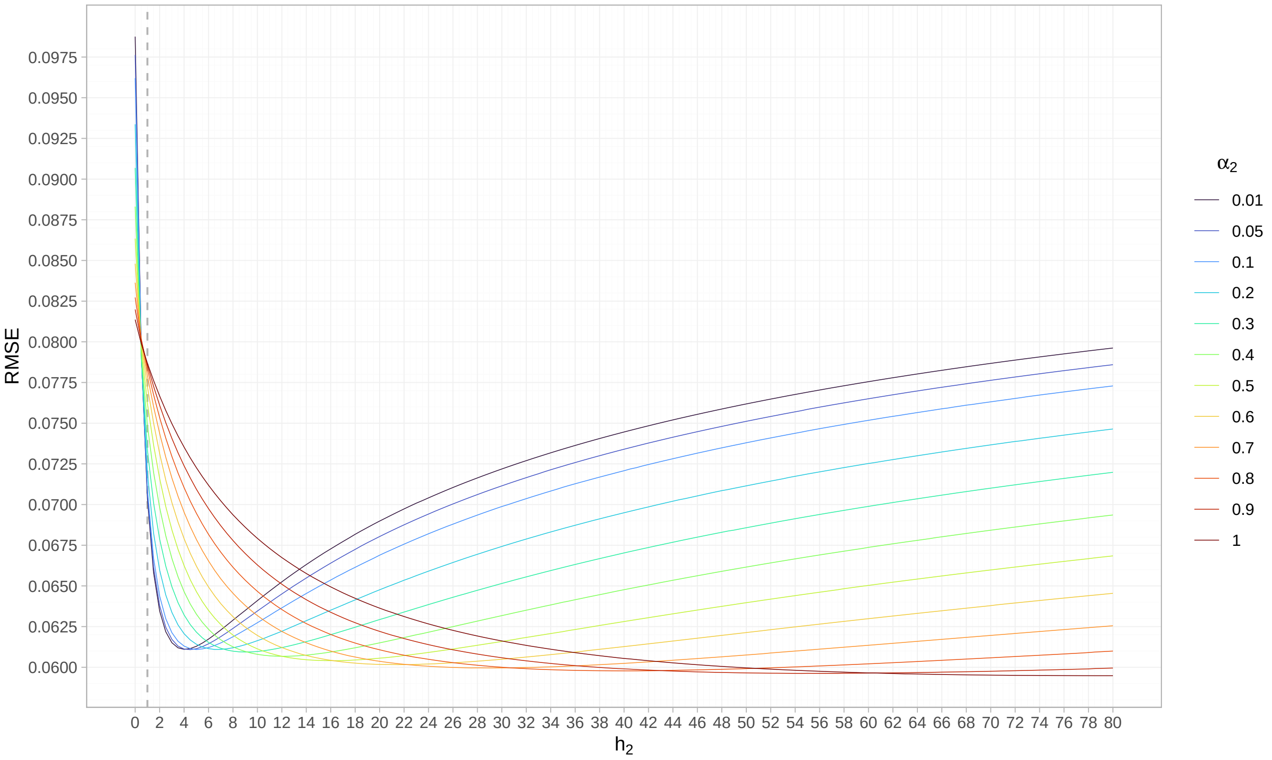
$N = 15$ it is apparent that the minimum $RMSE$ for small values of the index α_2 is achieved for h_2 around 5, for moderate values of α_2 around 15 and for large values, around 40. Similar results can be observed for $N = 30$, while as the size of the sample increases we can see that the improvement in the efficiency is less significant. Note that, the larger the sample size, the smaller the percentage of zero frequency cell occurrence is (Tables 4.1 and 4.2).

With respect to the robustness of the MM(Φ, α)PDE, results are presented in Figures 4.1b to 4.5b. As we can observe, estimators corresponding to larger value of the index α_2 tend to be more robust for every value of the tuning parameter h_2 and every sample size. Once again, there is an improvement in the robustness of the estimators as h_2 increases, which is more significant for small sample sizes.

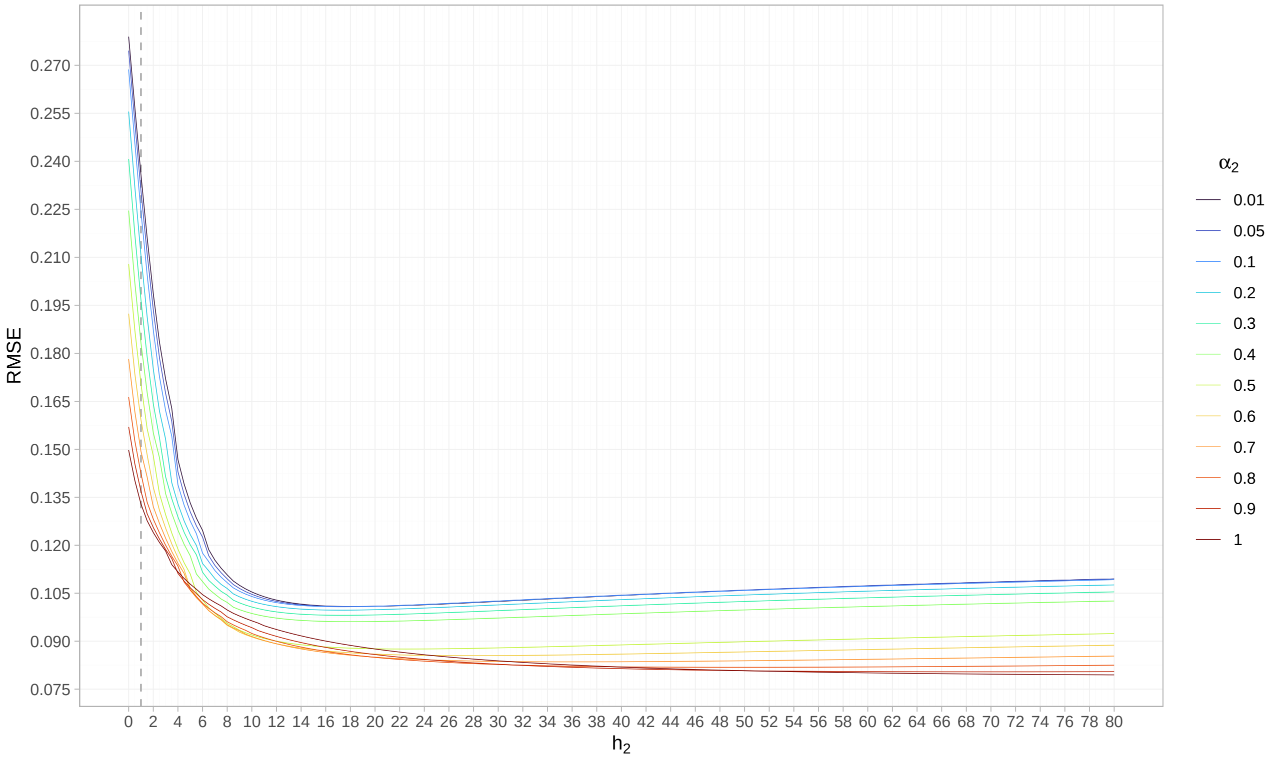
In order to derive conclusions regarding the size of the MDITS, Dale's (Dale, 1986) criterion is exploited (see, Subsection 2.4.3), where for 0.05 nominal level the exact simulated size is considered to be close to the nominal if it belongs to the range $[0.0357, 0.0695]$.

Figures 4.6 to 4.10, usually referred as *tile graphs*, present results regarding the size of the test statistic for the two cases under no contamination and under contamination. In these figures the exact simulated size is presented for every combination of the two indices α_1, α_2 and the two tuning parameters h_1 and h_2. Green tiles represent combinations of the indices and tuning parameters that lead to test statistics which satisfy Dale's criterion. Cooler colors represent conservative test statistics, while wormer colors liberal ones. As we can observe, in most of the cases for every sample size Dale's criterion is not satisfied both for the cases with no contamination and under contamination.

These results are in line with those of Alba-Fernández et al. (2018). Thus, in order to achieve a better approximation to the underlying distribution of the MDITS we are going to exploit to bootstrap p-value which can be derived from the following algorithm.

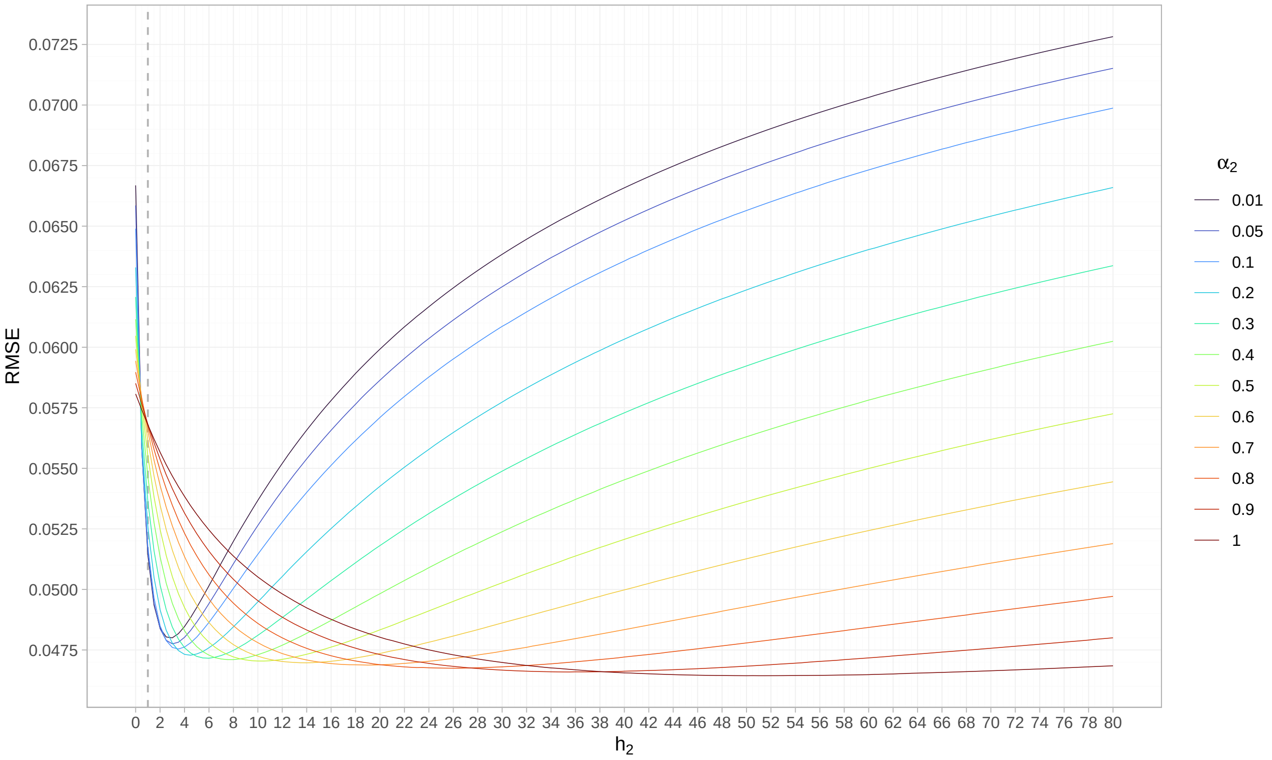


(a)

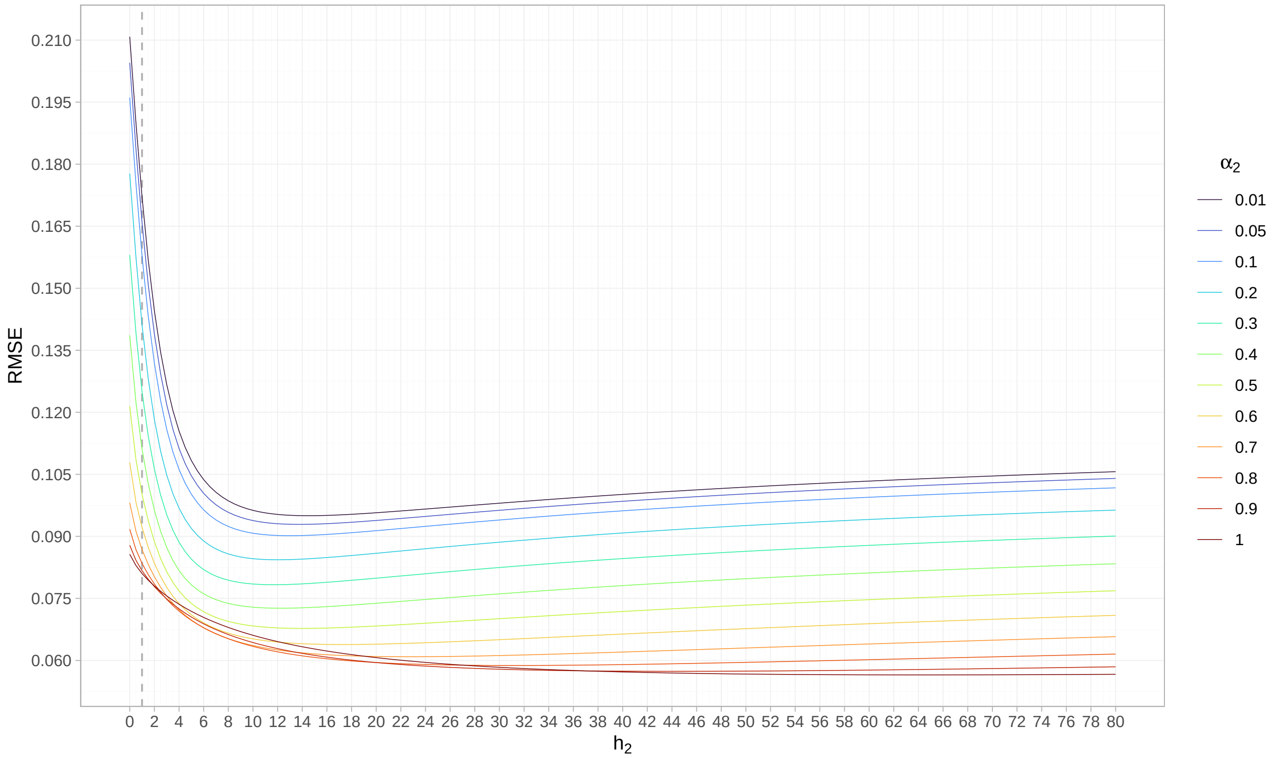


(b)

Figure 4.1: Root mean squared error for the MM(Φ, α)PDE. $N = 15$.
(a) Under no contamination. (b) Under contamination.

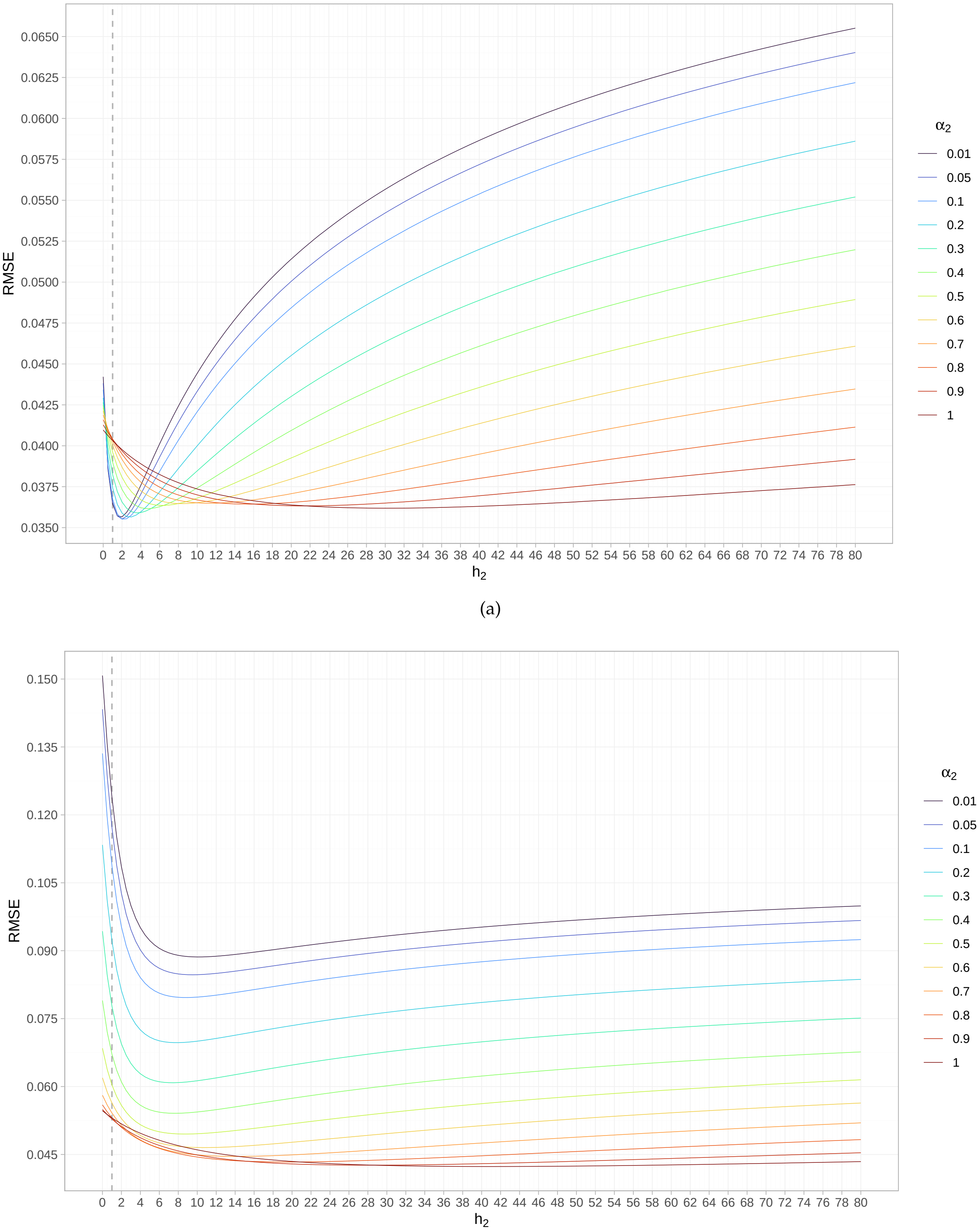


(a)



(b)

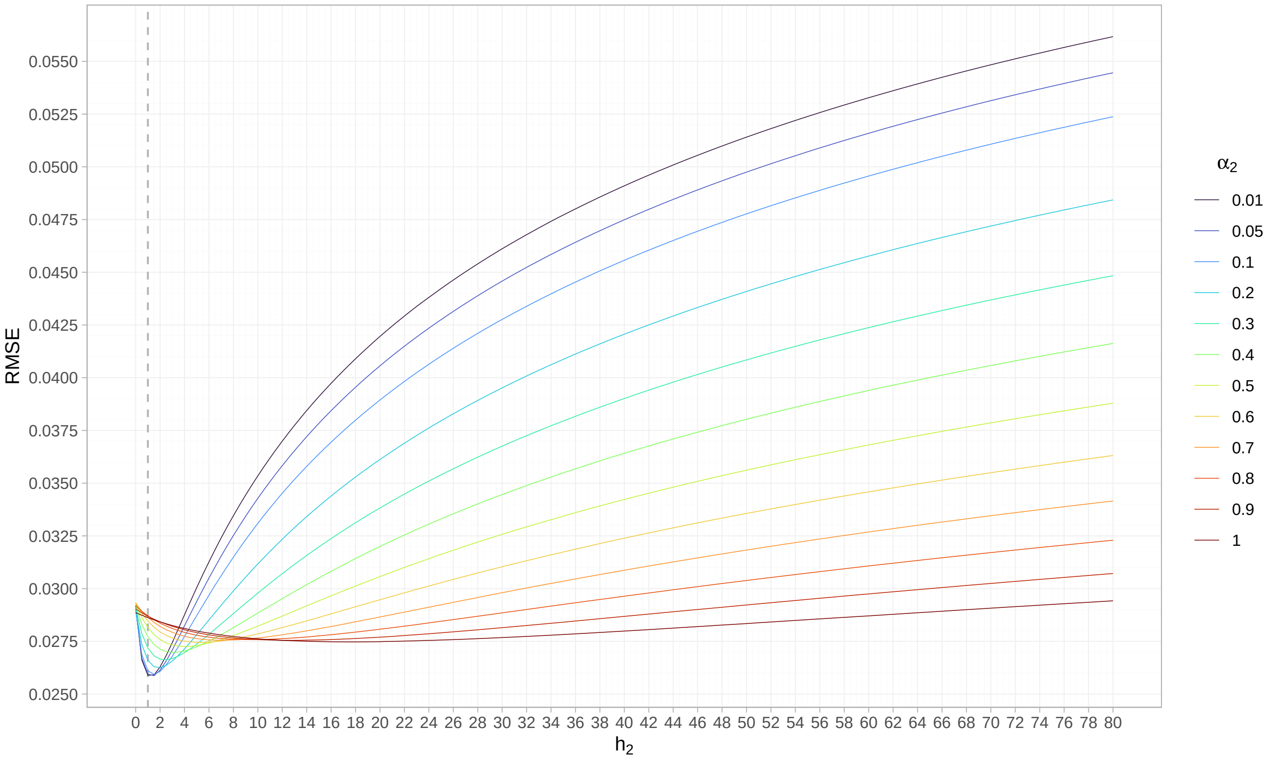
Figure 4.2: Root mean squared error for the MM(Φ, α)PDE. $N = 30$.
(a) Under no contamination. (b) Under contamination.



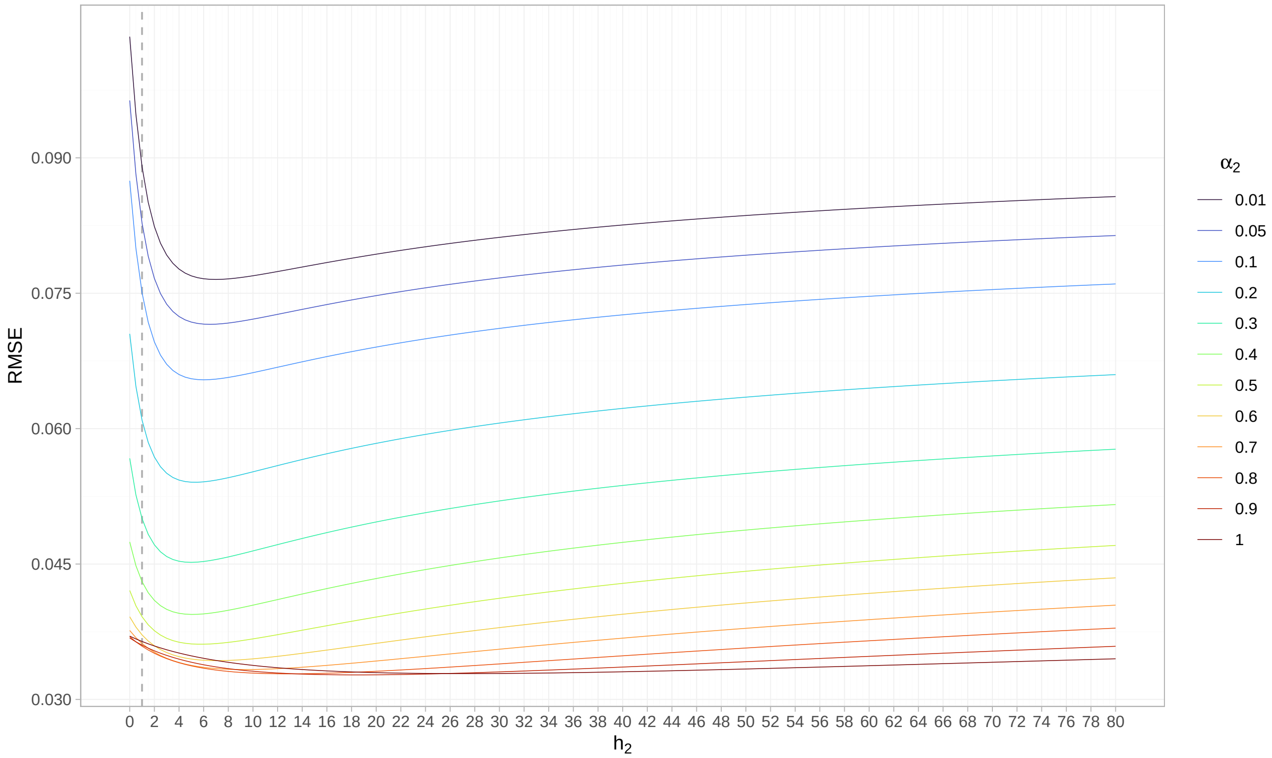
(a)

(b)

Figure 4.3: Root mean squared error for the MM(Φ,α)PDE. $N = 60$.
(a) Under no contamination. (b) Under contamination.

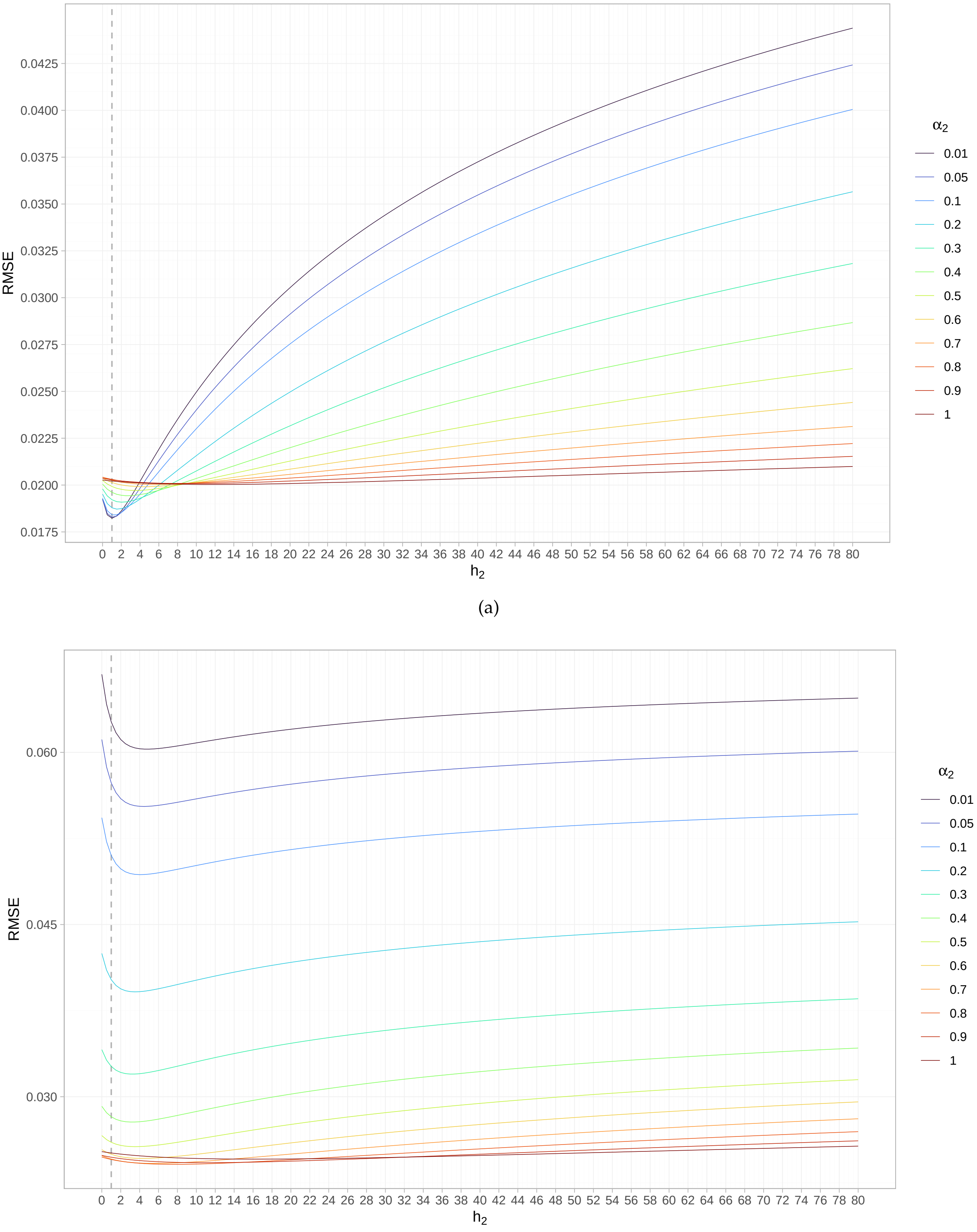


(a)



(b)

Figure 4.4: Root mean squared error for the MM(Φ, α)PDE. $N = 120$.
(a) Under no contamination. (b) Under contamination.

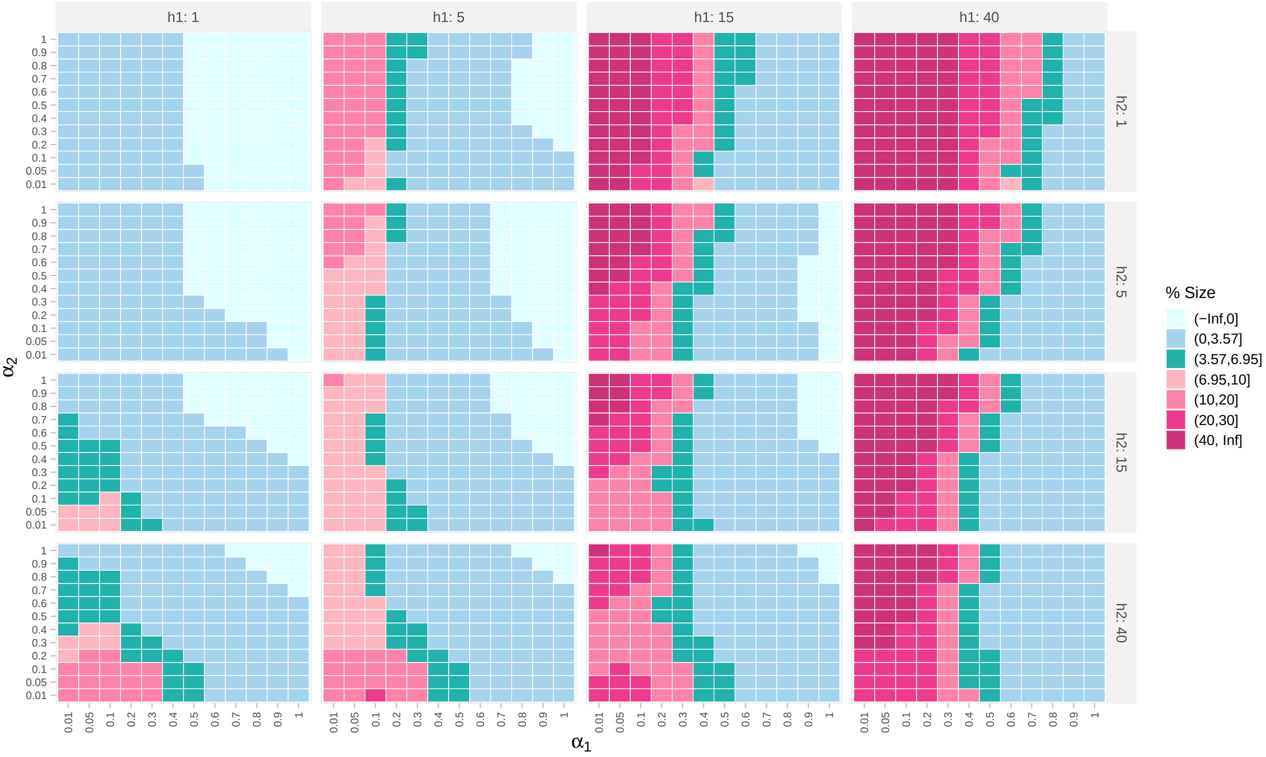


(a)

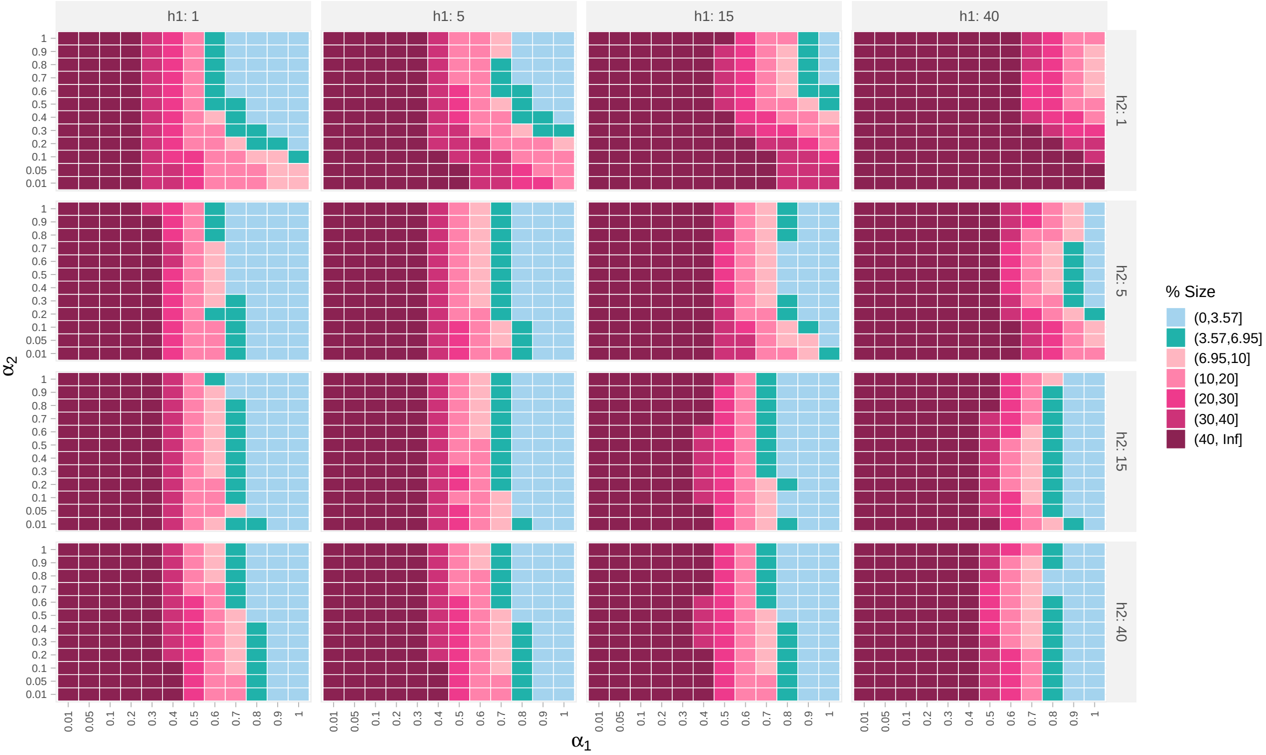
(b)

Figure 4.5: Root mean squared error for the MM(Φ, α)PDE. $N = 240$.
(a) Under no contamination. (b) Under contamination.

Figure 4.6: (%) size for the MDITS. $N = 15$. (a) Under no contamination. (b) Under contamination.



(a)



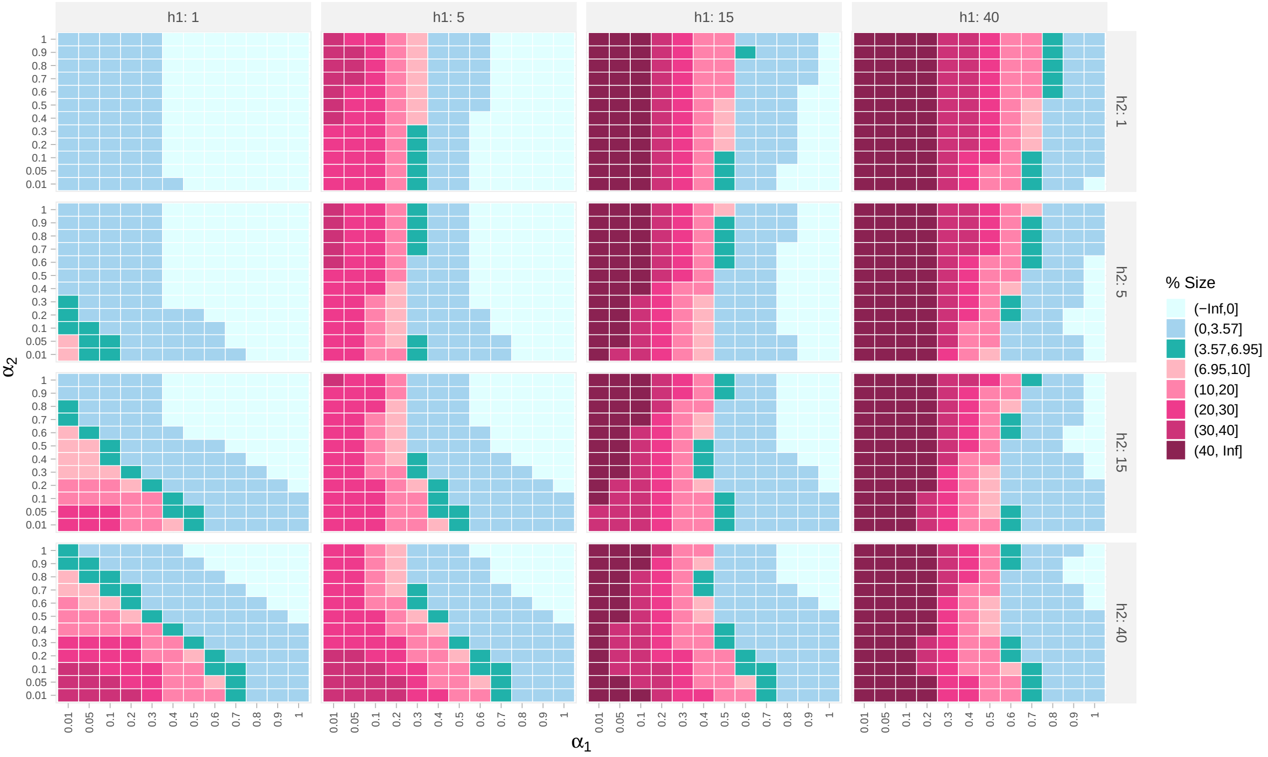
(b)

Figure 4.7: (%) size for the MDITS. $N = 30$. (a) Under no contamination. (b) Under contamination.

(a)

(b)

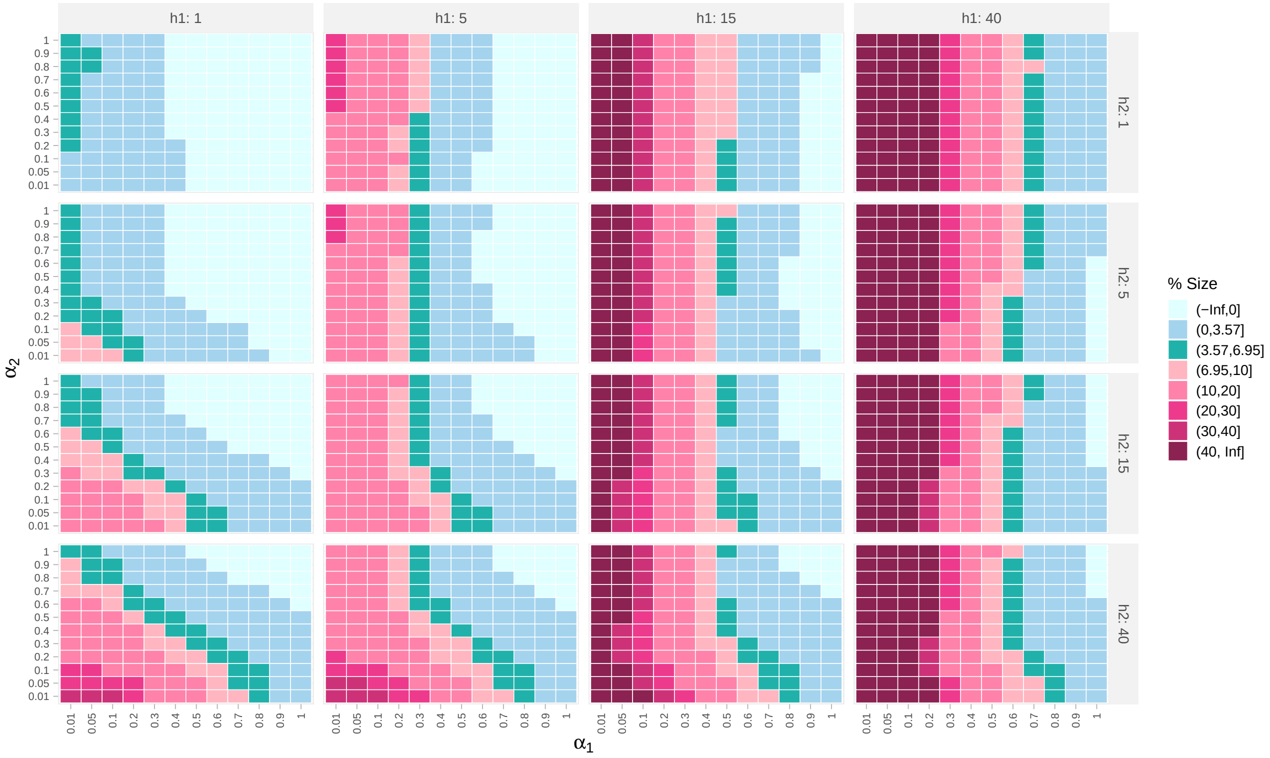
Figure 4.8: (%) size for the MDITS. $N = 60$. (a) Under no contamination. (b) Under contamination.



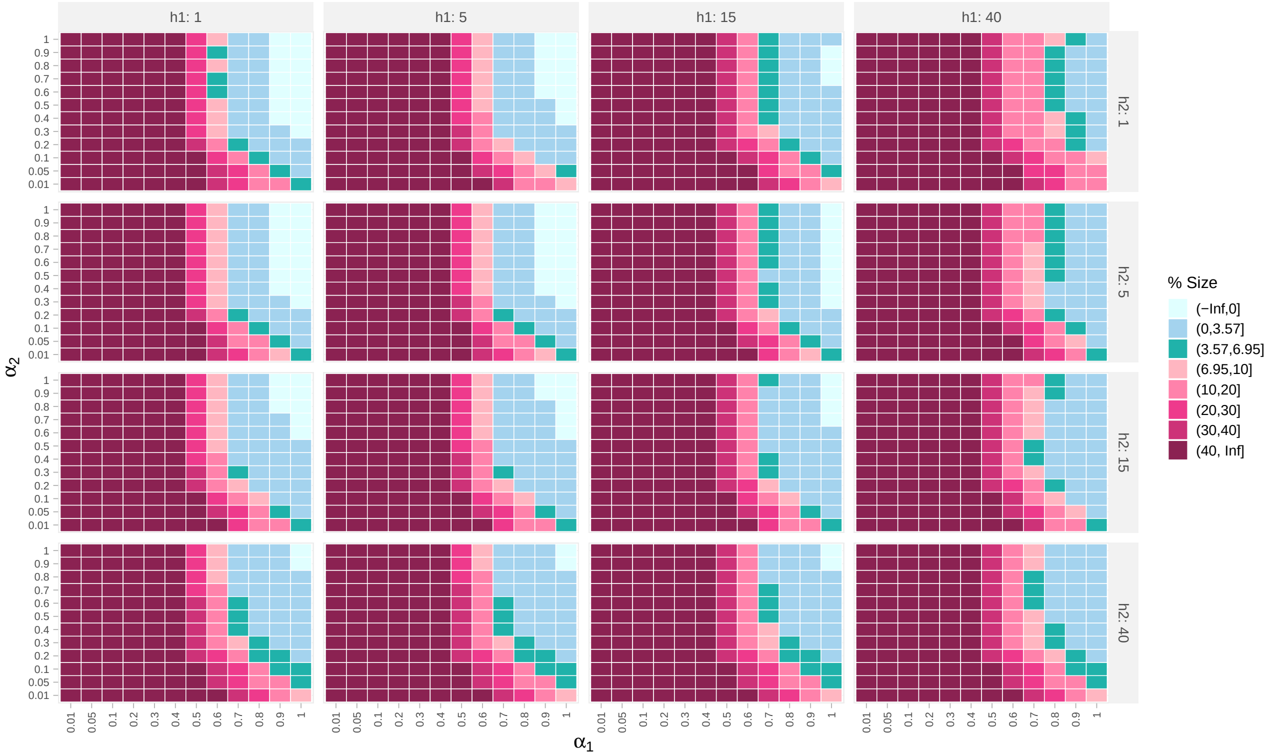
(a)

(b)

Figure 4.9: (%) size for the MDITS. $N = 120$. (a) Under no contamination. (b) Under contamination.



(a)



(b)

Figure 4.10: (%) size for the MDITS. $N = 240$. (a) Under no contamination. (b) Under contamination.

1. Calculate the observed value of the test statistic based on the available data $(X_1, \ldots, X_4)$, $T^{obs} \equiv T^{\alpha_1}_{\Phi_1,h_1}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})$.
2. Generate B bootstrap samples $(X_1^{*b}, \ldots, X_4^{*b})^\top \sim M(N, \mathbf{p}(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2,h_2})), b = 1, \ldots, B$.
3. Calculate the statistic for each bootstrap sample obtaining $T^{*b} \equiv T^{\alpha_1 * b}_{\Phi_1,h_1}(\hat{\boldsymbol{\theta}}^{\alpha_2 * b}_{\Phi_2,h_2})$, $b = 1, \ldots, B$.
4. Approximate the p-value by means of the expressions

$$\hat{p}_{boot} = \frac{\text{card}\left\{b : T^{*b} \geq T^{obs}\right\}}{B}$$

   where card$\{A\}$ denotes the number of elements in A.
5. Reject the null hypothesis if $\hat{p}_{boot} \leq \alpha$ (nominal size).

In order to relieve the computational burden this algorithm is applied for α_1, $\alpha_2 =$ 0.01, 0.10, 0.30, 0.60 and h_1, $h_2 = 1, 5, 15$. The sample sizes considered are N = 15, 30 , 60, 120 and 240, while B = 200 bootstrap samples have been taken into account. Results are presented in Figures 4.11 to 4.15.

To begin with, regarding the case where no contaminating data are present, we can see that as the sample size increases the observed size of each member of the MDITS family considered here gets closer to the nominal 0.05. Furthermore, we can see that in general the larger the value of the tuning parameter h_2, the more liberal the associated test is, especially for smaller values of the α_2 index. In particular, we can observe that for small sample sizes $N = 15$ the best test statistics in terms of size arise for values of $h_2 \leq 5$ and values of the α_2 index around 0.6. As the sample size increases, the best test statistics are observed for $h_1 = 1$ and $h_2 = 5$ with large values of the index α_2. In other words, as the tuning parameter h_2 increases larger values of the index α_2 are preferable.

Under contamination, we can observe that, although Dale's criterion is not satisfied, there exists a region in which test statistics appear to be more stable in terms of size. In this study, this region occurs for $h_1 = 15$, $h_2 \leq 5$, $\alpha_1 < 0.3$ where once again as h_2 increases larger values of the α_2 index are preferable. A closer look at the results indicates that the observed size of the test statistics belonging to the aforementioned region gets closer to the nominal 0.05 as the sample size increases. On the other hand, we can observe that the observed size of the test statistics that do not belong to this region tends to 1 as the sample size increases. Note that, the larger the sample size, the closer to the null model the contaminated model is. This fact indicates that there are two groups of test statistics in the MDITS family, those that are very sensitive to small deviations from the null model and those which can be considered as more stable in terms of size. The choice of which group is going to be used should be based on the perception of what these deviations refer to. If these deviations are considered as contamination, then more stable test statistics should be used. On the other hand, if the practitioner has the conviction that deviations comprise a pattern which should be incorporated into the model describing the generating mechanism, then more sensitive test statistics should be used.

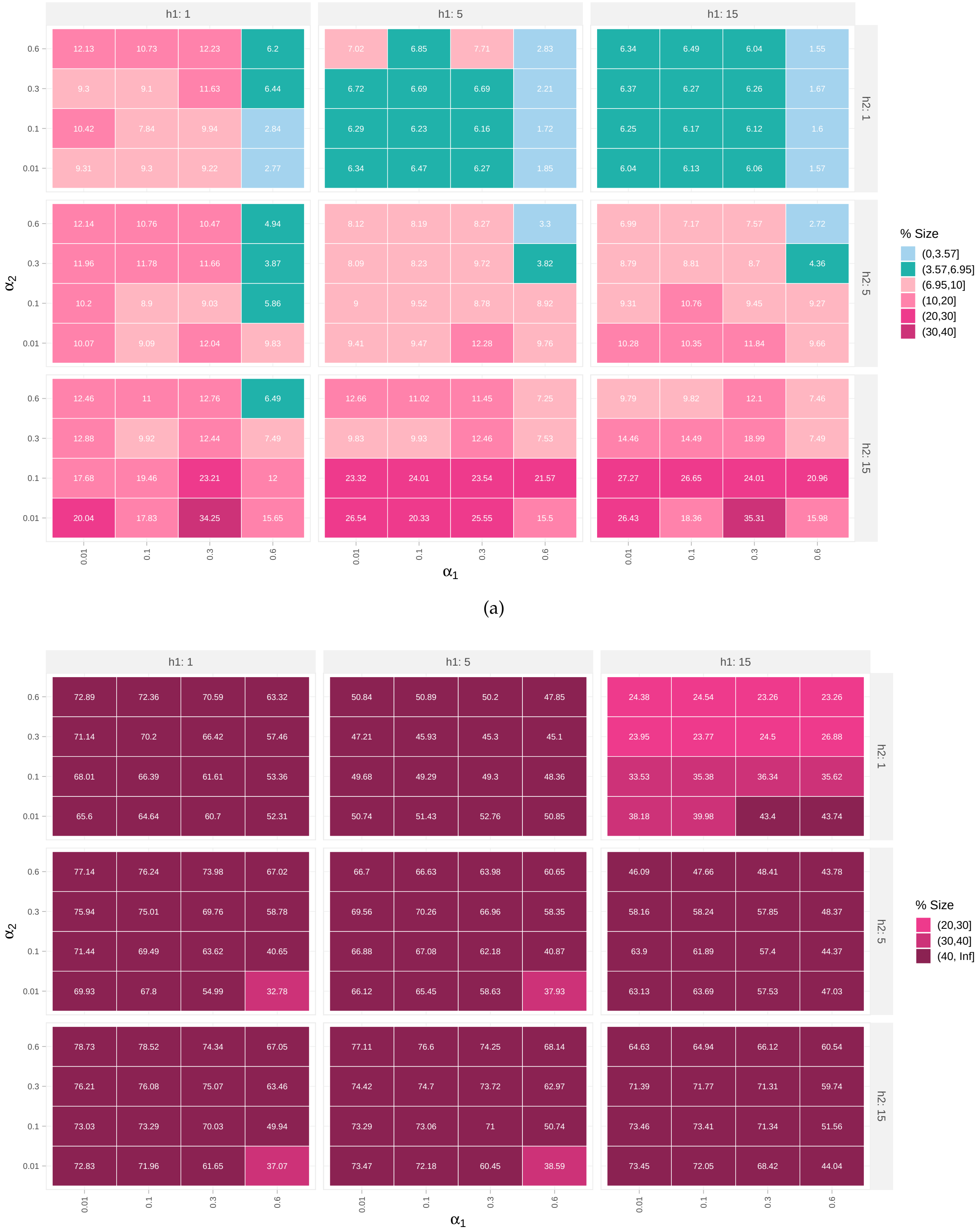
		h1: 1				h1: 5				h1: 15			
	α_2 \ α_1	0.01	0.1	0.3	0.6	0.01	0.1	0.3	0.6	0.01	0.1	0.3	0.6
h2: 1	0.6	25.39	25.25	25.43	4.82	24.95	24.94	24.9	4.34	24.96	24.88	24.89	4.35
	0.3	25.27	25.18	25.15	4.46	24.88	24.86	24.86	4.34	24.87	24.87	24.86	4.32
	0.1	24.49	24.5	24.9	2.58	24.48	24.49	24.86	3.85	24.45	24.39	24.4	3.71
	0.01	24.9	23.93	24.9	4.1	24.86	24.16	24.87	4.33	24.87	24.24	24.84	4.31
h2: 5	0.6	25.48	25.35	25.36	4.53	25.33	25.33	25.19	4.55	24.96	25.02	25.13	4.4
	0.3	25.36	25.41	25.63	4.92	25.46	25.41	25.2	4.91	25.04	25.76	26.13	6.19
	0.1	25.38	25.33	25.7	9.05	25.18	25.21	25.83	4.97	25.19	25.4	25.93	7.63
	0.01	25.36	25.14	28.42	7.84	25.32	25.18	26.11	5.35	25.5	25.16	26.03	5.3
h2: 15	0.6	25.7	25.63	26.03	7.44	25.69	25.61	25.94	7.28	25.32	25.85	28.01	7.67
	0.3	25.92	25.85	26.89	14.47	25.87	25.88	27.98	14.36	26.87	26.82	30.19	14.41
	0.1	33.6	31.44	29.47	8.67	33.62	33.6	29.28	8.64	32.79	33.33	31.75	8.48
	0.01	38.35	27.76	44.05	17.67	28.9	29.08	44.28	10.32	35.51	27.88	44.03	10.32

(a)

		h1: 1				h1: 5				h1: 15			
	α_2 \ α_1	0.01	0.1	0.3	0.6	0.01	0.1	0.3	0.6	0.01	0.1	0.3	0.6
h2: 1	0.6	57.85	56.68	54.72	44.13	39.11	38.54	40.25	33.6	29.73	29.87	29.91	25.58
	0.3	53.49	51.98	48.77	39.01	37.48	36.13	35.72	32.04	31.23	31.58	31.84	27.82
	0.1	45.95	47.69	43.89	35.32	37.06	38.19	38.29	34.49	32.42	33.1	32.71	28.63
	0.01	44.73	44.27	43.23	34.23	39.76	39.93	40.53	35.46	33.62	33.39	34.24	30.31
h2: 5	0.6	61.98	63.59	63.33	50.88	57.56	59.78	57.59	47.93	48.31	50.6	44.65	39.49
	0.3	63.34	64.43	62.17	44.11	62.84	64.63	56.92	44.87	57.5	58.23	57.31	46.76
	0.1	61.11	59.51	51.54	32.59	59.36	58.3	54.31	34.98	58.9	57.16	56.8	42.96
	0.01	56.75	52.91	45.91	29.46	57.15	56.6	52.56	31.64	57.55	57.8	57.01	46.26
h2: 15	0.6	67.4	69.33	69.15	61.33	66.61	66.91	67.3	55.11	60.58	64.44	69.73	55.69
	0.3	72.08	73.79	71.15	56.03	72.33	72.98	71.22	55.93	74.83	74.54	75.86	58.32
	0.1	76.3	73.76	67.29	46.68	76.87	74.72	67.27	47.08	75.68	74.75	73.65	47.74
	0.01	75.13	72.56	63.47	44.61	72.91	73.22	67.35	43.43	74.81	72.6	69.25	43.49

(b)

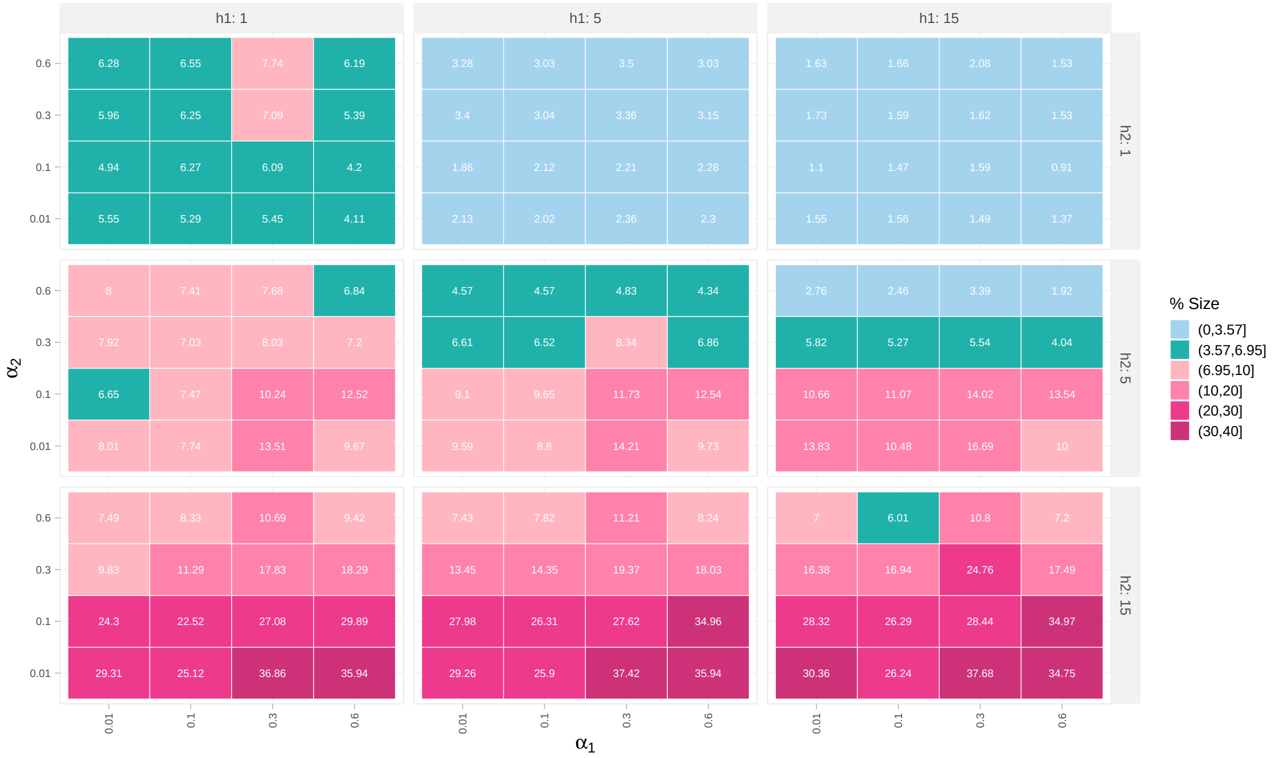
Figure 4.11: (%) size for the MDITS based on the bootstrap algorithm. $N = 15$. (a) Under no contamination. (b) Under contamination.



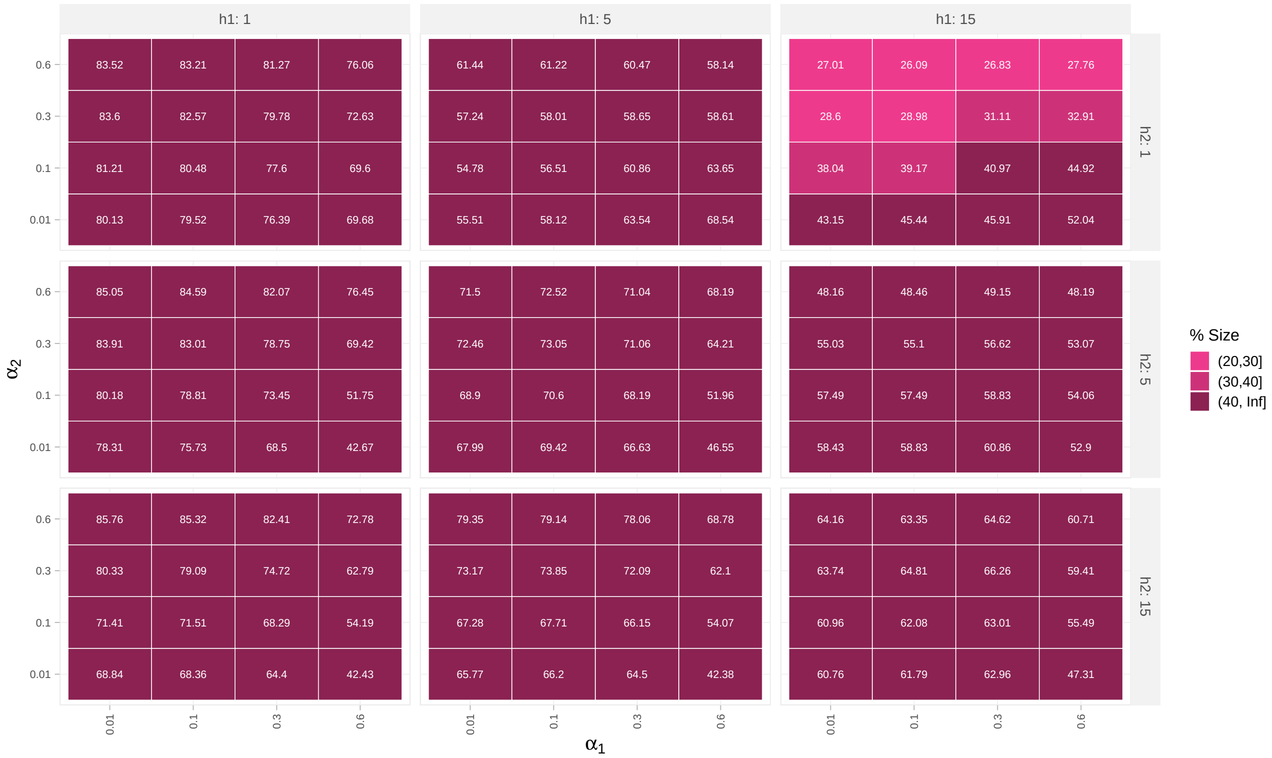
(a)

(b)

Figure 4.12: (%) size for the MDITS based on the bootstrap algorithm. $N = 30$. (a) Under no contamination. (b) Under contamination.



(a)

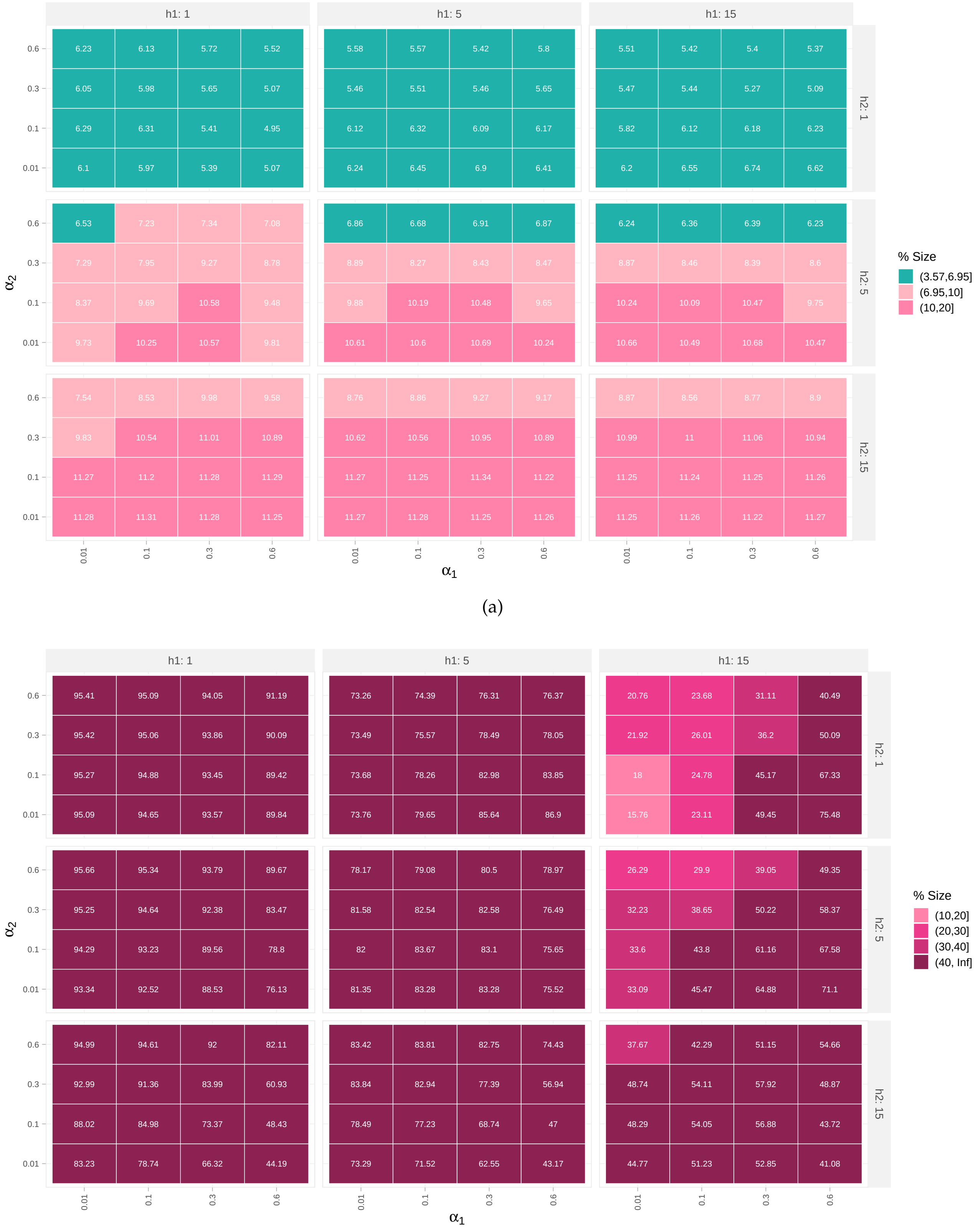


(b)

Figure 4.13: (%) size for the MDITS based on the bootstrap algorithm. $N = 60$. (a) Under no contamination. (b) Under contamination.

(b)

Figure 4.14: (%) size for the MDITS based on the bootstrap algorithm. $N = 120$. (a) Under no contamination. (b) Under contamination.



		h1: 1				h1: 5				h1: 15			
	α_2 \ α_1	0.01	0.1	0.3	0.6	0.01	0.1	0.3	0.6	0.01	0.1	0.3	0.6
h2: 1	0.6	6.23	6.13	5.72	5.52	5.58	5.57	5.42	5.8	5.51	5.42	5.4	5.37
	0.3	6.05	5.98	5.65	5.07	5.46	5.51	5.46	5.65	5.47	5.44	5.27	5.09
	0.1	6.29	6.31	5.41	4.95	6.12	6.32	6.09	6.17	5.82	6.12	6.18	6.23
	0.01	6.1	5.97	5.39	5.07	6.24	6.45	6.9	6.41	6.2	6.55	6.74	6.62
h2: 5	0.6	6.53	7.23	7.34	7.08	6.86	6.68	6.91	6.87	6.24	6.36	6.39	6.23
	0.3	7.29	7.95	9.27	8.78	8.89	8.27	8.43	8.47	8.87	8.46	8.39	8.6
	0.1	8.37	9.69	10.58	9.48	9.88	10.19	10.48	9.65	10.24	10.09	10.47	9.75
	0.01	9.73	10.25	10.57	9.81	10.61	10.6	10.69	10.24	10.66	10.49	10.68	10.47
h2: 15	0.6	7.54	8.53	9.98	9.58	8.76	8.86	9.27	9.17	8.87	8.56	8.77	8.9
	0.3	9.83	10.54	11.01	10.89	10.62	10.56	10.95	10.89	10.99	11	11.06	10.94
	0.1	11.27	11.2	11.28	11.29	11.27	11.25	11.34	11.22	11.25	11.24	11.25	11.26
	0.01	11.28	11.31	11.28	11.25	11.27	11.28	11.25	11.26	11.25	11.26	11.22	11.27

(a)

		h1: 1				h1: 5				h1: 15			
	α_2 \ α_1	0.01	0.1	0.3	0.6	0.01	0.1	0.3	0.6	0.01	0.1	0.3	0.6
h2: 1	0.6	95.41	95.09	94.05	91.19	73.26	74.39	76.31	76.37	20.76	23.68	31.11	40.49
	0.3	95.42	95.06	93.86	90.09	73.49	75.57	78.49	78.05	21.92	26.01	36.2	50.09
	0.1	95.27	94.88	93.45	89.42	73.68	78.26	82.98	83.85	18	24.78	45.17	67.33
	0.01	95.09	94.65	93.57	89.84	73.76	79.65	85.64	86.9	15.76	23.11	49.45	75.48
h2: 5	0.6	95.66	95.34	93.79	89.67	78.17	79.08	80.5	78.97	26.29	29.9	39.05	49.35
	0.3	95.25	94.64	92.38	83.47	81.58	82.54	82.58	76.49	32.23	38.65	50.22	58.37
	0.1	94.29	93.23	89.56	78.8	82	83.67	83.1	75.65	33.6	43.8	61.16	67.58
	0.01	93.34	92.52	88.53	76.13	81.35	83.28	83.28	75.52	33.09	45.47	64.88	71.1
h2: 15	0.6	94.99	94.61	92	82.11	83.42	83.81	82.75	74.43	37.67	42.29	51.15	54.66
	0.3	92.99	91.36	83.99	60.93	83.84	82.94	77.39	56.94	48.74	54.11	57.92	48.87
	0.1	88.02	84.98	73.37	48.43	78.49	77.23	68.74	47	48.29	54.05	56.88	43.72
	0.01	83.23	78.74	66.32	44.19	73.29	71.52	62.55	43.17	44.77	51.23	52.85	41.08

(b)

Figure 4.15: (%) size for the MDITS based on the bootstrap algorithm. $N = 240$. (a) Under no contamination. (b) Under contamination.

4.4 Conclusions

In this chapter, the modified $(\Phi, \alpha)-$power divergence family is defined for the first time. This family, beyond the index α, incorporates a tuning parameter h which determines the weight imposed on the zero frequency cells. Rigorous theoretical results regarding the decomposition and the asymptotic distribution of the associated modified minimum $(\Phi, \alpha)-$power divergence family of estimators (MM(Φ, α)PDE) are provided, under the framework of misspecified models. As particular case, when the postulated model is correctly specified the decomposition of the MM(Φ, α)PDE and its asymptotic distribution coincide with those of the ordinary minimum $(\Phi, \alpha)-$power divergence family of estimators. Furthermore, the asymptotic distribution of the associated modified double indexed test statistics (MDITS) is provided as well, under the assumption that the postulated model is correctly specified. An exhaustive simulation study reveals that through the proposed methodology, robust estimators with improved efficiency may be obtained. Moreover, members of the MDITS family appear to be more stable in terms of size under contamination schemes.

By replacing the density function with the probability generating function, a probability generating function$-$based divergence is obtained, for parameter estimation. A similar type of divergence, could be based on the weighted squared distance between two probability generating functions, in a fashion similar to Anderson$-$ Darling or Crammer$-$von Mises distance. These type of problems have been recently discussed in Jiménez-Gamero & Batsidis (2017) and Tay et al. (2019) and could be investigated under the scenario discussed in this chapter. The use, in our case, of a tuning parameter and an index, could be useful in an attempt to investigate the impact of zero frequency cells and contaminated data or outliers.

Chapter 5

Application to General Insurance

An important task of a general insurance company is to define the appropriate premium of an insurance contract (policy). The premium may be divided into two categories, namely the technical premium and the commercial one. Consumers (policy holders) are exposed only to the commercial premium, which is clearly defined through a strategic process of the company and among others takes into account the market competition, the targeted market share and other commercial considerations. Not appropriate evaluation of the technical premium may lead to catastrophic decisions, thus a well defined technical premium is of crucial importance for the company. In its simplest form, the technical premium is comprised from the cost of the risk and further loadings regarding the operating expenses of the company, as well as a margin of profit. The operating expenses and the margin of profit can deterministically be defined and they may be expressed as rate over the technical premium. Careful consideration should be placed on the evaluation of the cost of the risk which is referring to the expected aggregate losses of a period, for example a year, and is a stochastic amount.

In non−life businesses (general insurance) the most commonly used technique to evaluate the expected cost of total losses in a period (say a year) is the collective risk model. This model places the focus to the losses. Specifically, we assume that there is going to be a number of N independent and identically distributed losses in each period, where each loss, say X, generates a claim amount X_j, $j = 1, \ldots, N$. Usually, in order to describe the mechanism that generates the number N of claims (frequency) we use the Poisson and the Negative Binomial models, where their main practical difference is the variance/mean ratio, which is 1 for the former and greater than 1 for the latter. The claim amounts (severity) X_j, $j = 1, \ldots, N$ are commonly modelled through the Lognormal, Gamma and Pareto distributions and considered to be independent and identically distributed. Furthermore, we assume that N and X are independent random variables. The collective risk model dictates that the total losses for each period will be $S = \sum_{j=1}^{N} X_j$ while the expected amount of losses in a given period is $E(S) = E(N)E(X)$ and the volatility of S around $E(S)$ is given as $Var(S) = Var(X)E(N) + E(X^2)Var(N)$.

An approximate and very popular method to estimate the aggregate distribution of the losses is to use the Monte Carlo simulation. This method is very popular, mainly, due to its flexibility and ease on implementation. The implementation of this method consists of randomly choosing a number N of losses, based on a frequency distribution, and then for each of these losses a claim amount X_j, $j = 1, \ldots, N$, based on a severity distribution. Then, by applying the aforementioned procedure a large number of times, we can derive conclusions about the percentiles of the distribution of the total losses $F_S(s) = Pr(S \leq s)$ (see Parodi, 2014; Klugman et al., 2008).

As it is apparent, defining models which describe appropriately the generating mechanism of the number (frequency) and the amount (severity) of losses is of paramount importance in deriving the appropriate distribution of the total losses in which the premium is going to be based. The methodology developed in Chapters 2 and 4 can firmly be applied in order to achieve this goal. To this regard, in the sequel of this chapter, we present a paradigm where the frequency and the severity models are estimated through the MM(Φ, α)PDE, while the goodness−of−fit test is based on the MDITS. Moreover, we include in our analysis the classical estimators and test statistics derived from the chi−squared and the Kullback−Leibler divergences.

5.1 Frequency distribution examination

As claim number generating mechanism we consider the Poisson distribution with rate parameter $\theta_0 = 10^3$ and suppose that we have 15 observations at hand, i.e. $N_1, \ldots, N_{15} \sim Poiss(10^3)$. The choice of the number of observations is justified from the fact that in real case scenarios the fitting of the frequency distribution is based on a small number of experience periods (Parodi, 2014, Ch. 14). Based on those 15 observations we are going to define the associated multinomial model with $\lceil \sqrt{15} \rceil = 4$ categories as following

$$
\begin{aligned}
p_1 &= Pr(N \leq n_1), \\
p_k &= Pr(N \leq n_k) - Pr(N \leq n_{k-1}),\ k = 2, 3, \\
p_4 &= 1 - Pr(N \leq n_3),
\end{aligned}
$$

where $N \sim Poiss(\theta)$ and $n_k = m + ks$, $k = 1, 2, 3$, with $m = \min_j(N_j : j = 1, \ldots, 15)$ and $s = (\max_j(N_j : j = 1, \ldots, 15) - \min_j(N_j : j = 1, \ldots, 15))/4$. Note that in this case the resulting multinomial model is not fixed and is based on the observed data range. Furthermore, the non−parametric estimator of the true probabilities of the multinomial model is defined as

$$
\begin{aligned}
\hat{p}_1 &= \frac{1}{15} \sum_{j=1}^{15} I(N_j \leq n_1), \\
\hat{p}_k &= \frac{1}{15} \sum_{j=1}^{15} I(N_j \leq n_k) - \sum_{i=1}^{k-1} \hat{p}_k,\ k = 2, 3, \\
\hat{p}_4 &= 1 - \sum_{i=1}^{3} \hat{p}_i,
\end{aligned}
$$

where $I(Z \leq z)$, which equals 1 if $Z \leq z$ and 0 otherwise, is the indicator function.

For the estimation of the unknown parameter, the MM(Φ, α)PDE is used with $\Phi_2(u) = 1 - (1 - \alpha_2^{-1})u + \alpha_2^{-1} u^{1+\alpha_2}$, $\alpha = 0.01, 0.10, 0.25, 0.50, 0.75$ and $h = 0 \ldots (1) \ldots 10$, while we are going to consider also the minimum chi−squared (CS) and Kullback−Leibler (KL) estimators, which can be derived from the MCRPDE for $\lambda = 1$ and $\lambda = 0$, respectively. The experiment is repeated 10^4 times and conclusions are based on the root mean squared error ($RMSE$), which is given as

$$RMSE_{\hat{\theta}_l^E} = \sqrt{\frac{1}{10^4}\sum_{l=1}^{10^4}(\hat{\theta}_l^E - \theta_0)^2}$$

where $\hat{\theta}_l^E$ is the estimation of the unknown parameter for the l^{th} sample regarding the E minimum divergence estimator.

Note that, in Chapter 2, instead of the $RMSE$ we used the MSE. The main difference of these two accuracy measures is that, the former is measured in the same units as the targeted value and thus it is easier to interpret it. In every case, both measures can be used without any differentiation on the conclusions, regarding the efficiency and the robustness of the compared estimators. In this thesis, we use alternatively either one of those two accuracy measures, aiming exclusively at better visualization of the results.

Results are presented in Figure 5.1, from where we can observe that the tuning parameter does not improve significantly the efficiency of the estimators. This result is directly connected to the number of times, zero frequency cells occur. Note that, in this case the percentage of zero frequency cell occurrence is 0.00, 0.10, 0.12 and 0.00 for each of the four cells, respectively. As indicated from the simulation study of Subsection 4.3, the tuning parameter is expected to improve considerably the efficiency, mainly in cases where we systematically observe zero frequency cells. In most other cases, a relatively limited improvement is expected to take place.

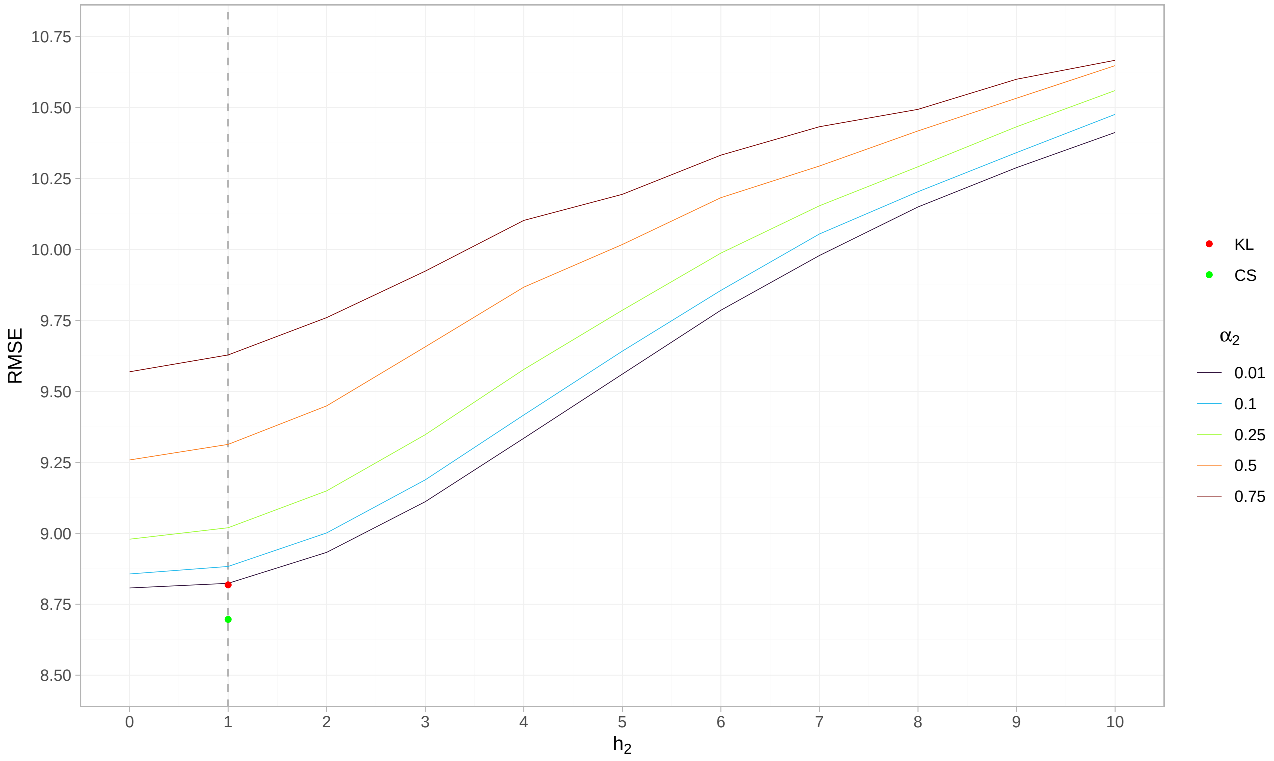


Figure 5.1: Root mean squared error for the MM(Φ_2, α_2)PDE along with the minimum chi−squared (CS) and Kullback−Leibler (KL) estimators.

In addition, we can see that the minimum Kullback−Leibler estimator coincides, as expected, with the ordinary M(Φ_2, α_2)PDE for $\alpha_2 = 0.01$. Finally, it is apparent that the minimum chi−squared estimator has the best behaviour in terms of efficiency among all used estimators.

In relation to the goodness−of−fit tests, we use the MDITS with $\Phi_1(u) = 1-(1-\alpha_1^{-1})u+\alpha_1^{-1}u^{1+\alpha_1}$, $\Phi_2(u) = 1-(1-\alpha_2^{-1})u+\alpha_2^{-1}u^{1+\alpha_2}$, $\alpha_1, \alpha_2 = 0.01, 0.10, 0.25, 0.50, 0.75$. As tuning parameter we consider only the value $h_1, h_2 = 1$, since no improvement in the efficiency of the estimators is imposed by this parameter. In other words, the family of ordinary DITS is taken into consideration. Moreover, we consider the chi−squared and the Kullback−Leibler test statistics in our analysis. The examination of the goodness−of−fit tests is based on the size of the test. Regarding the DITS family, results are presented in Figure 5.2, while Table 5.1 contains information about the size of the CS and KL test statistics.

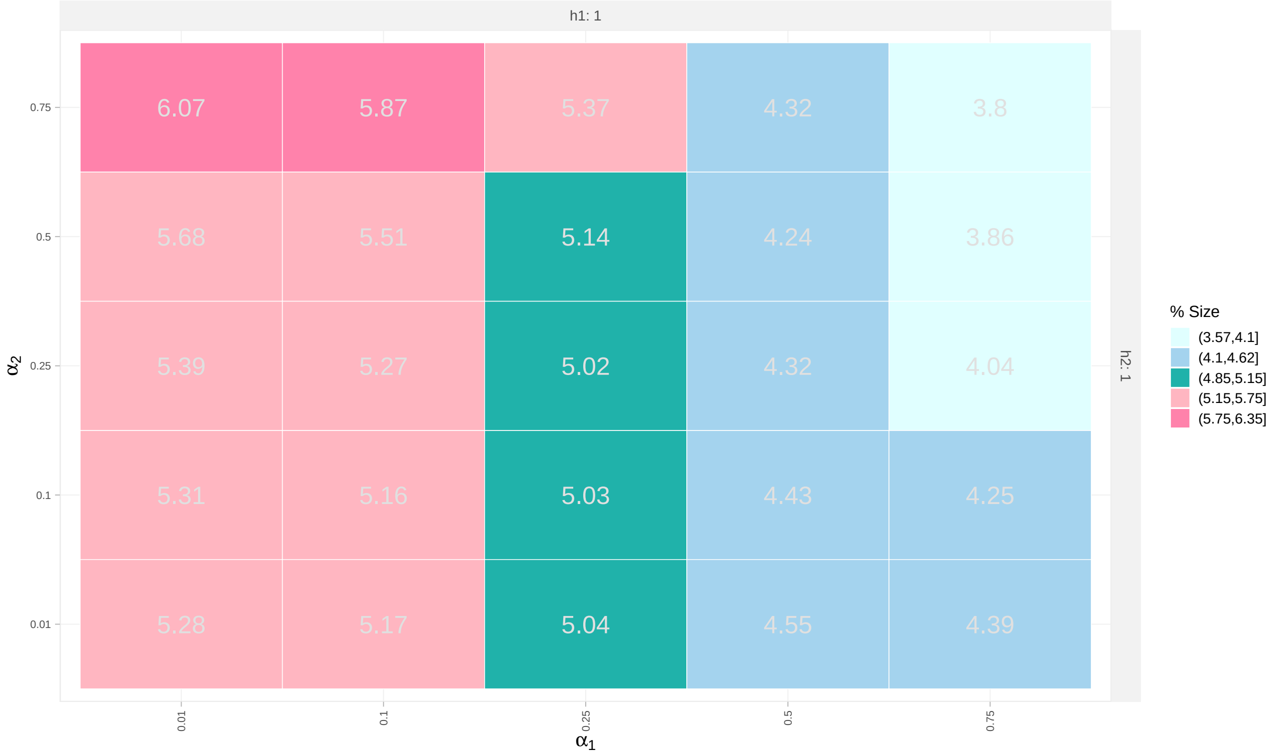


Figure 5.2: (%) size for the DITS.

Table 5.1: (%) size for the chi−squared (CS) and Kullback−Leibler (KL) test statistics.

CS	KL
4.43	5.29

As we can observe, even though some test statistics appear to have liberal and others conservative performance, all the test statistics under consideration satisfy Dale's criterion given in Subsection 2.4.3. Members of the DITS family with $\alpha_2 = 0.25$ appear to have the best performance in terms of size.

5.2 Severity distribution examination

Regarding the severity of each claim, we consider as generating mechanism the *lognormal* distribution with location parameter $\theta_0^1 = 6$ and scale parameter $\theta_0^2 = 1$ and 0.5% contamination from a *generalized Pareto* distribution, with location parameter 15000, scale parameter 1200 and shape parameter 1.5. Suppose further, that we have 10^3 observations. The number of observations equals the mean of the *Poisson* distribution used above to generate the frequency data. The associated multinomial model consists of $\lceil\sqrt{1000}\rceil = 32$ categories and is given as

$$\begin{aligned} p_1 &= Pr(X \leq x_1), \\ p_k &= Pr(X \leq x_k) - Pr(X \leq x_{k-1}),\ k = 2, \dots, 31, \\ p_{32} &= 1 - Pr(X \leq x_{31}), \end{aligned}$$

where $X \sim LN(\theta_1, \theta_2)$ and $x_k = ks, k = 1, \dots, 31$ and s is the interquartile range (IQR) of the observed data. Thus, the resulting multinomial model is a rather data based model than a fixed one. The non−parametric estimator of the true probabilities of the multinomial model is defined as

$$\begin{aligned} \hat{p}_1 &= \frac{1}{10^3} \sum_{j=1}^{10^3} I(X_j \leq x_1), \\ \hat{p}_k &= \frac{1}{10^3} \sum_{j=1}^{10^3} I(X_j \leq x_k) - \sum_{i=1}^{k-1} \hat{p}_k,\ k = 2, \dots 31, \\ \hat{p}_{32} &= 1 - \sum_{i=1}^{31} \hat{p}_i. \end{aligned}$$

Once more, for the estimation of the unknown parameter, the MM(Φ, α)PDE is used, with $\Phi_2(u) = 1 - (1 - \alpha_2^{-1})u + \alpha_2^{-1}u^{1+\alpha_2}$, $\alpha = 0.01, 0.10, 0.25, 0.50, 0.75$ and $h = 0 \dots (1) \dots 10$, while we are going to consider also the minimum CS and KL estimators which can be derived from the MCRPDE for $\lambda = 1$ and $\lambda = 0$, respectively. The experiment is repeated 10^4 times and conclusions are based on the overall root mean squared error ($ORMSE$), which is given as

$$ORMSE_{{}^E\hat{\boldsymbol{\theta}}_l} = \sqrt{\frac{1}{10^4} \sum_{l=1}^{10^4} \left\{ ({}^E\hat{\theta}_l^1 - \theta_0^1)^2 + ({}^E\hat{\theta}_l^2 - \theta_0^2)^2 \right\}}$$

where ${}^E\hat{\boldsymbol{\theta}}_l$ is the estimation of the unknown parameter for the l^{th} sample regarding the E minimum divergence estimator.

Results are presented in Figure 5.3. As we can observe the ordinary M(Φ_2, α_2)PDE (h = 1) appear to be more robust for larger values of the index α_2. Furthermore, we can see that the KL estimator coincides with the M(Φ_2, α_2)PDE when $\alpha_2 = 0.01$, while the CS estimator has the worst behaviour in terms of robustness. In addition, we can observe that the tuning parameter h imposes an improvement in the robustness of the estimators, especially for those that are derived for small values of the index α_2. In particular we can see that M(Φ_2, α_2)PDE with $\alpha_2 = 0.01$ appear to have the best performance among all the considered estimators when h_2 is around 5.

In relation to the goodness−of−fit, we use the MDITS with $\Phi_1(u) = 1 - (1 -$

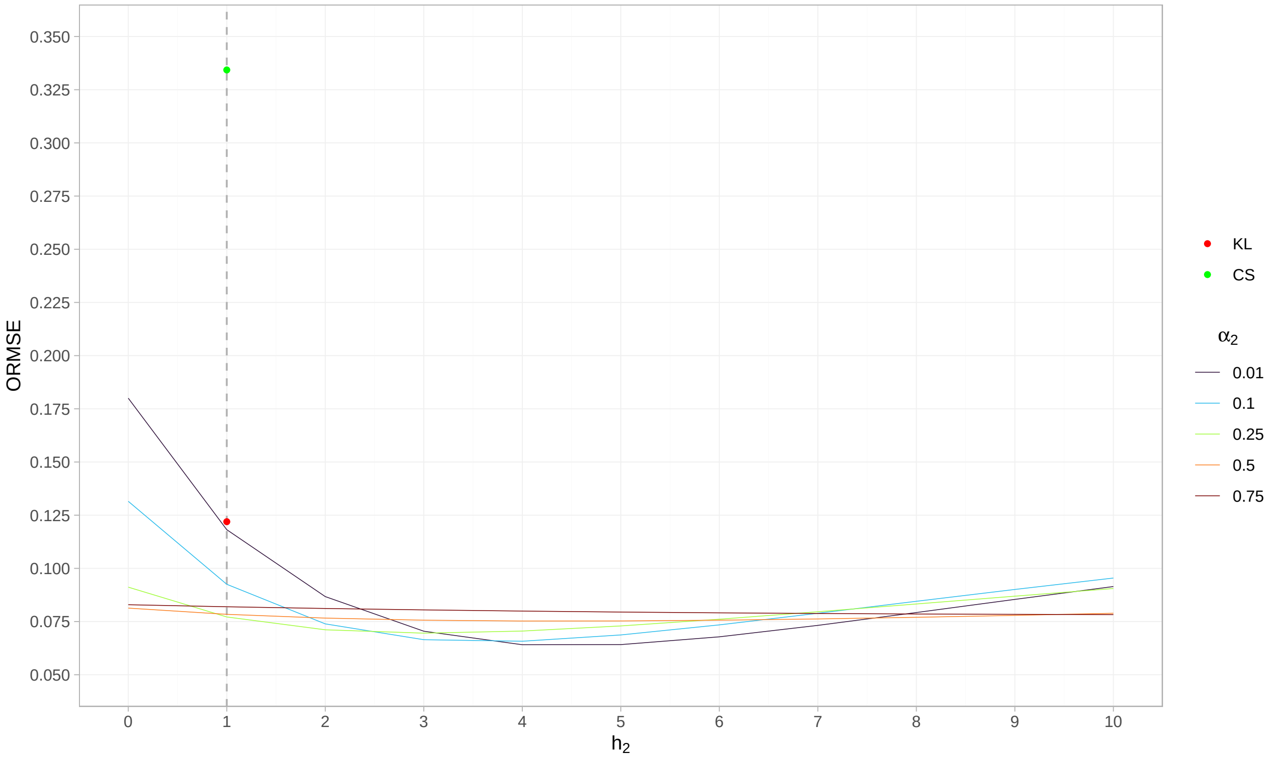
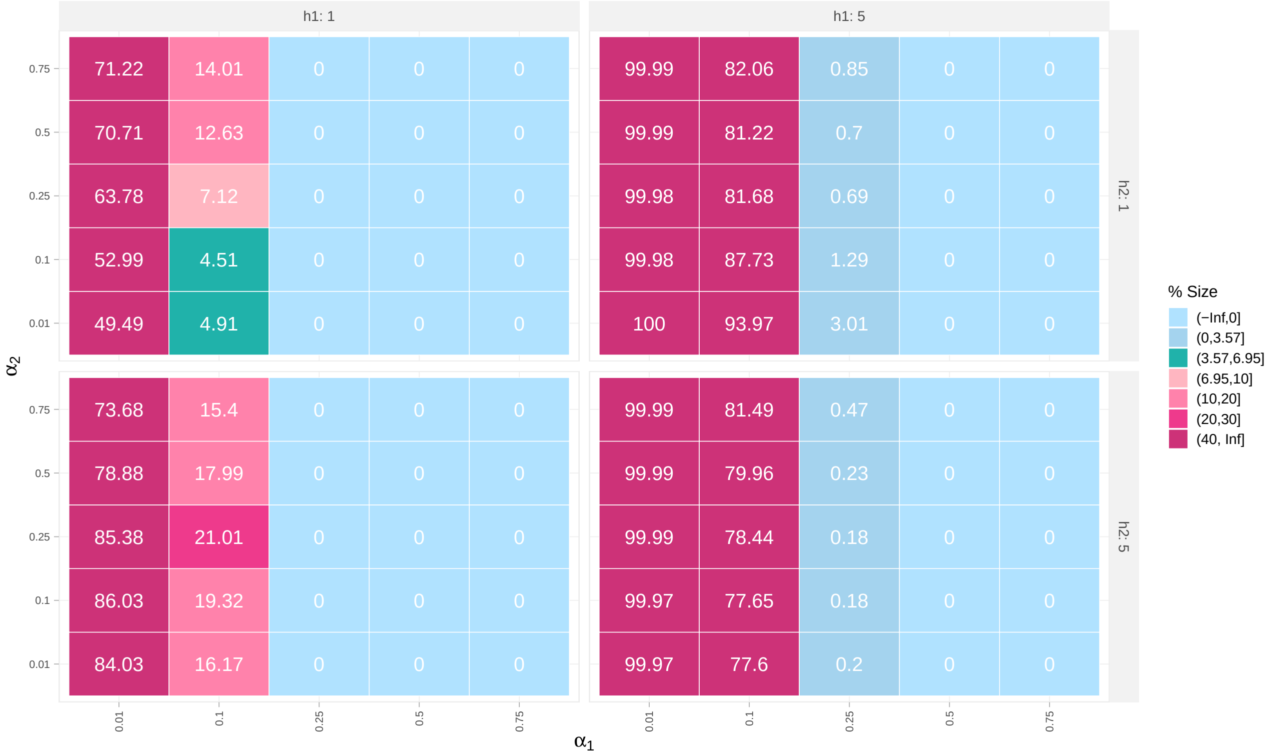


Figure 5.3: Overal root mean squared error for the MM(Φ_2, α_2)PDE along with the minimum chi−squared (CS) and Kullback−Leibler (KL) estimators.



		h1: 1					h1: 5				
	α_2 \ α_1	0.01	0.1	0.25	0.5	0.75	0.01	0.1	0.25	0.5	0.75
h2: 1	0.75	71.22	14.01	0	0	0	99.99	82.06	0.85	0	0
h2: 1	0.5	70.71	12.63	0	0	0	99.99	81.22	0.7	0	0
h2: 1	0.25	63.78	7.12	0	0	0	99.98	81.68	0.69	0	0
h2: 1	0.1	52.99	4.51	0	0	0	99.98	87.73	1.29	0	0
h2: 1	0.01	49.49	4.91	0	0	0	100	93.97	3.01	0	0
h2: 5	0.75	73.68	15.4	0	0	0	99.99	81.49	0.47	0	0
h2: 5	0.5	78.88	17.99	0	0	0	99.99	79.96	0.23	0	0
h2: 5	0.25	85.38	21.01	0	0	0	99.99	78.44	0.18	0	0
h2: 5	0.1	86.03	19.32	0	0	0	99.97	77.65	0.18	0	0
h2: 5	0.01	84.03	16.17	0	0	0	99.97	77.6	0.2	0	0

Figure 5.4: (%) size for the MDITS.

Table 5.2: (%) size for the chi−squared (CS) and Kullback−Leibler (KL) test statistics.

CS	*KL*
92.37	54.7

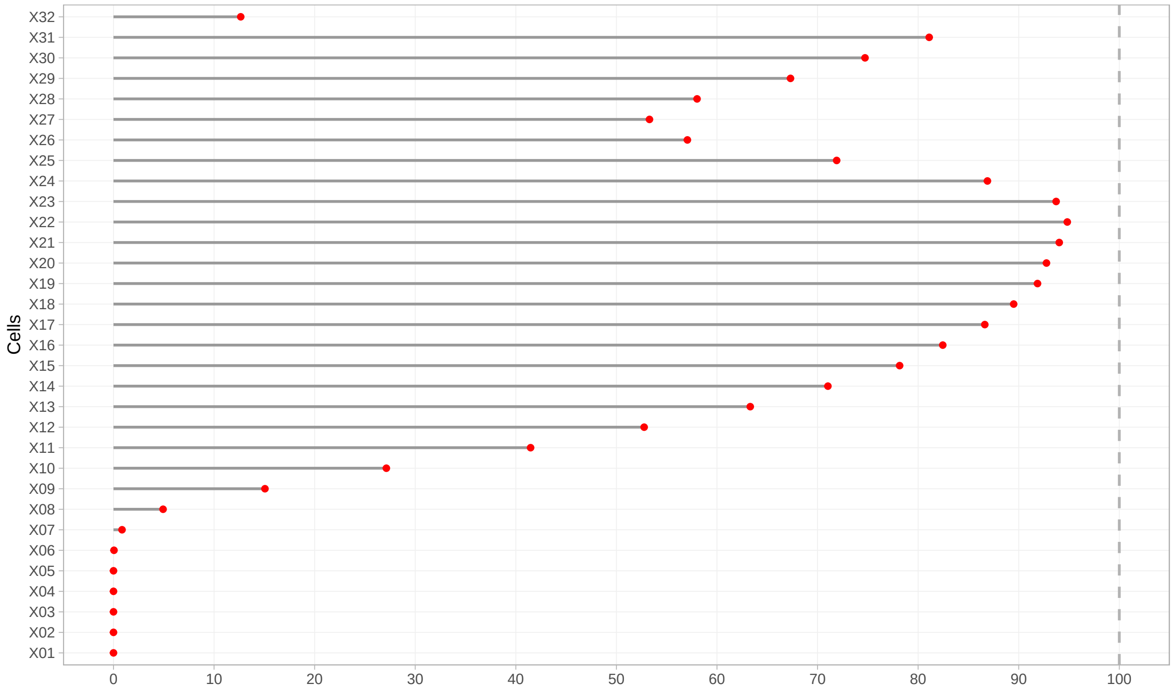


Figure 5.5: (%) zero frequency cell occurrence.

$\alpha_1^{-1})u + \alpha_1^{-1}u^{1+\alpha_1}$, $\Phi_2(u) = 1 - (1 - \alpha_2^{-1})u + \alpha_2^{-1}u^{1+\alpha_2}$, $\alpha_1, \alpha_2 = 0.01, 0.10, 0.25, 0.50$, 0.75 and $h_1, h_2 = 15, 5$. Moreover, we consider the CS and KL test statistics in our analysis. The examination of the goodness−of−fit tests is based on the size of the test and results are presented in Figure 5.4 and Table 5.2 regarding the MDITS and CS and KL respectively. Results indicate that the asymptotic $\mathcal{X}^2$ approximation is rather poor and alternative approximations, like the bootstrap mentioned in Chapter 4, should be considered. Note that the poor approximation of the $\mathcal{X}^2$ is highly associated with the fact that in this case the percentage of zero frequency cell occurrence was very high in many of cells (see Figure 5.5).

Note that for the conducted simulation studies of this chapter, we used the R software (R Core Team, 2016), while for the optimization the **optimise** function from the base package **stats**. For the manipulation of the data we used the packages **dplyr** (Wickham et al., 2022) and **data.table** (Dowle & Srinivasan, 2022). Furthermore, for the visualization of the results we used the package **ggplot2** (Wickham, 2016) while we took advantage of the package **future.apply** (Bengtsson, 2021) to activate parallel computing in order to compensate the computational burden. The syntax is provided in Appendices C.6 and C.7.

5.3 Conclusions

In this chapter the focus is placed to the collective risk model, which is used for the evaluation of the expected cost of total losses in a period (say year) for non−life lines of businesses (general insurance). In particular, we present through a paradigm how the modified minimum $(\Phi, \alpha)-$power diverge family of estimators MM(Φ, α)PDE and the modified double index family of test statistics (MDITS) can be utilized in order to model the frequency (number of losses in a period) and the severity (amount of losses) of a company's portfolio. The implementation appears to be straightforward, while the flexibility of those estimators and test statistics may lead to better approximations of the underlying model, something that is of paramount importance to the practitioner and the insurance industry.

Chapter 6

Future Work

6.1 Censored data

The existence of censored data in survival analysis and reliability theory is very common, while the determination of an appropriate model under such schemes is of great importance. Our intention is to study the estimators and test statistics that are associated with the $(\Phi, \alpha)-$power divergence family under the framework of censored data.

In particular, for the formulation of the problem one possible solution is to follow the lines of Vonta & Karagrigoriou (2014), who have studied Csiszar's family of measures under this framework. Let suppose that subjects are observed during a time interval $[0, \tau]$, which is partitioned into k subintervals of the form $E_i = (\tau_{i-1}, \tau_i]$, $i = 1, \ldots, k$, with $0 = \tau_0 < \tau_1 < \cdots < \tau_k = \tau$. Let n_i be the number of subjects at risk at the beginning of the i^{th} interval, while d_i is the number of failures during the i^{th} interval. Let further, $h_i^c = 1 - h_i$ and h_i be the survival and hazard rate for the i^{th} interval respectively. For each i, let $\mathcal{F}_i = \sigma\{d_1, \ldots, d_{i-1}, n_1, \ldots, n_i\}$ be the $\sigma-$field generated by $d_1, \ldots, d_{i-1}, n_1, \ldots, n_i$. Then, conditionally on $\mathcal{F}_i$, the random variables d_i, $i = 1, \ldots, k$ are independent and specifically, the number of failures d_i follows a binomial distribution

$$Pr(d_i = j|\mathcal{F}_i) = \binom{n_i}{j} h_i^j (1 - h_i)^{n_i - j},\ j = 0, \ldots, n_i \tag{6.1}$$

Suppose further that there are N subjects with failure and censoring time denoted by X_i and C_i, $i = 1, \ldots, N$ respectively, with F being the distribution function of X_i. The data consist of pairs (T_i, δ_i), where $T_i = min(X_i, C_i)$ and $\delta_i = I(T_i \leq C_i), i = 1, \ldots, N$. Under the assumption that the censoring occurs only at τ_i equation (6.1) holds with

$$h_i = \frac{F(\tau_i) - F(\tau_{i-1})}{1 - F(\tau_{i-1})},\ i = 1, \ldots, k,$$

which is the hazard rate for the discrete case.

For testing the composite null hypothesis

$H_0 : F = F_0(\boldsymbol{\theta})$ or equivalenty $H_0 : \mathbf{h} = (h_1, \ldots, h_k)^\top = \mathbf{h}_0(\boldsymbol{\theta}) = (h_{10}(\boldsymbol{\theta}), \ldots, h_{k0}(\boldsymbol{\theta}))^\top$

we propose the following family of test statistics

$$\begin{aligned} T_{\Phi_1}^{\alpha_1}(\mathbf{d},\mathbf{n},\mathbf{h}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})) = \frac{2}{\Phi_1''(1)}\sum_{i=1}^{k} n_i\Bigg\{ & h_{i0}^{1+\alpha_1}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})\Phi_1\left(\frac{d_i/n_i}{h_{i0}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})}\right)+ \\ & h_{i0}^{c\;1+\alpha_1}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})\Phi_1\left(\frac{d_i/n_i}{h_{i0}^{c}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2})}\right)\Bigg\}, \end{aligned}$$

where $\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}$ is the associated minimum $(\Phi_2,\alpha_2)-$power divergence estimator given by

$$\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2} = \arg\inf_{\boldsymbol{\theta}\in\Theta}\sum_{i=1}^{k} n_i\left\{h_{i0}^{1+\alpha_2}(\boldsymbol{\theta})\Phi_2\left(\frac{d_i/n_i}{h_{i0}(\boldsymbol{\theta})}\right)+h_{i0}^{c\;1+\alpha_2}(\boldsymbol{\theta})\Phi_2\left(\frac{d_i/n_i}{h_{i0}^{c}(\boldsymbol{\theta})}\right)\right\}.$$

Our quest is to find the asymptotic distribution of the estimator and the test statistic and then, through an extensive simulation study, to compare the behaviour of members of this family with those that can be derived from the Csiszar's family.

6.2 Selection procedures of optimum indices

Another issue of great importance is the selection of optimal indices, that are incorporated on families of estimators and test statistics, based on a data driven or a theoretical procedure.

Regarding the calibration of the index α, of the minimum BHHJ$-$divergence estimator, which governs the trade-off between efficiency and robustness, Warwick & Jones (2005) in order to determine the optimal index value proposed a data driven procedure, which minimizes an empirical measure of the mean squared error of the estimator. Recently, Sugasawa & Yonekura (2021) proposed a selection criterion based on an asymptotic approximation of the Hyvarinen score applied to an unnormalized model defined by robust divergence. Furthermore, Mandal et al. (2010) proposed a data driven procedure in order to determine the optimal value of the tuning parameter h of the penalized estimators. This procedure is based on the minimization of the error in estimation based on Rao's (Rao, 1963) expression from the estimated bias in multinomial model for the BHHJ family.

In this thesis, general families of test statistics have been considered that provide flexibility, through their indices and tuning parameters, resulting in improved performance. However, the construction of a procedure that determines the optimal values of those indices and tuning parameters is of paramount importance and to the best of our knowledge there is a void in the literature.

Bibliography

Akaike, H. (1973). Information theory and an extension of maximum likelihood principle. In V. B. N. Petrov, & F. Csaki (Eds.), *Proceedings of the 2nd International Symposium on Information Theory* (pp. 267–281). Budapest: Akademiai Kiado.

Alba-Fernández, M. V., Jiménez-Gamero, M. D., & Ariza-López, F. J. (2018). Minimum penalized ϕ-divergence estimation under model misspecification. *Entropy (Basel, Switzerland)*, *20*(5), 329. doi:`10.3390/e20050329`.

Ali, S. M., & Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, *28*(1), 131–142. doi:`10.1111/j.2517-6161.1966.tb00626.x`.

Alin, A. (2007). A note on penalized power-divergence test statistics. *International Journal of Mathematical and Computational Sciences*, *1*(3), 175 – 180. URL: `https://publications.waset.org/vol/3`.

Alin, A. (2008). Comparison of ordinary and penalized power-divergence test statistics for small and moderate samples in three-way contingency tables via simulation. *Communications in Statistics - Simulation and Computation*, *37*(8), 1593–1602. doi:`10.1080/03610910802050894`.

Alin, A. (2012). Penalized power-divergence test statistics for testing linear-by-linear association. *Journal of Statistical Computation and Simulation*, *82*(2), 227–239. doi:`10.1080/00949655.2011.565418`.

Alin, A., & Kurt, S. (2008). Ordinary and penalized minimum power - divergence estimators in two-way contingency tables. *Computational Statistics*, *23*(3), 455–468. doi:`10.1007/s00180-007-0088-2`.

Arndt, C. (2001). *Information Measures*. Springer Berlin, Heidelberg. doi:`10.1007/978-3-642-56669-1`.

Auguie, B. (2017). *gridExtra: Miscellaneous Functions for "Grid" Graphics*. URL: `https://CRAN.R-project.org/package=gridExtra`.

Bader, B., & Yan, J. (2020). *eva: Extreme Value Analysis with Goodness-of-Fit Testing*. URL: `https://cran.r-project.org/web/packages/eva/index.html`.

Bahadur, R. R. (1967). Rates of convergence of estimates and test statistics. *The Annals of Mathematical Statistics*, *38*(2), 303–324. URL: `https://www.jstor.org/stable/2239145`.

Bahadur, R. R. (1971). *Some Limit Theorems in Statistics*. Society for Industrial and Applied Mathematics. doi:`10.1137/1.9781611970630.ch1`.

Bartolucci, F., & Scaccia, L. (2004). Testing for positive association in contingency tables with fixed margins. *Computational Statistics & Data Analysis*, *47*(1), 195–210. doi:`10.1016/j.csda.2003.10.019`.

Basu, A., & Basu, S. (1998). Penalized minimum disparity methods for multinomial models. *Statistica Sinica*, *8*(3), 841–860. URL: `https://www.jstor.org/stable/24306467`.

Basu, A., Basu, S., & Chaudhuri, G. (1997). Robust minimum divergence procedures for count data models. *Sankhyā: The Indian Journal of Statistics*, *59*(1), 11–27. URL: `http://www.jstor.org/stable/25052977`.

Basu, A., Ghosh, A., Mandal, A., Martin, N., & Pardo, L. (2020). Robust wald-type tests in GLM with random design based on minimum density power divergence estimators. *Statistical Methods & Applications*, . doi:`10.1007/s10260-020-00544-4`.

Basu, A., Harris, I. R., & Basu, S. (1996). Tests of hypotheses in discrete models based on the penalized hellinger distance. *Statistics & Probability Letters*, *27*(4), 367–373. doi:`10.1016/0167-7152(95)00101-8`.

Basu, A., Harris, I. R., Hjort, N. L., & Jones, M. C. (1998). Robust and efficient estimation by minimising a density power divergence. *Biometrika*, *85*(3), 549–559. doi:`10.1093/biomet/85.3.549`.

Basu, A., Mandal, A., & Pardo, L. (2010). Hypothesis testing for two discrete populations based on the hellinger distance. *Statistics & Probability Letters*, *80*(3), 206 – 214. doi:`10.1016/j.spl.2009.10.008`.

Basu, A., Shioya, H., & Park, C. (2011). *Statistical Inference: The Minimum Distance Approach*. Chapman and Hall/CRC. doi:`10.1201/b10956`.

Basu, S., Basu, A., & Jones, M. C. (2006). Robust and efficient parametric estimation for censored survival data. *Annals of the Institute of Statistical Mathematics*, *58*, 341–355. doi:`10.1007/s10463-005-0004-x`.

Batsidis, A., Economou, P., & Tzavelas, G. (2016a). Tests of fit for a lognormal distribution. *Journal of Statistical Computation and Simulation*, *86*(2), 215–235. doi:`10.1080/00949655.2014.1003138`.

Batsidis, A., Martin, N., Pardo, L., & Zografos, K. (2014). A necessary power divergence-type family of tests for testing elliptical symmetry. *Journal of Statistical Computation and Simulation*, *84*(1), 57–83. doi:`10.1080/00949655.2012.694437`.

Batsidis, A., Martin, N., Pardo, L., & Zografos, K. (2016b). φ-divergence based procedure for parametric change-point problems. *Methodology and Computing in Applied Probability*, *18*(1), 21–35. doi:`10.1007/s11009-014-9398-3`.

Bekara, M., Knockaert, L., Seghouane, A.-K., & Fleury, G. (2006). A model selection approach to signal denoising using kullback's symmetric divergence. *Signal Processing*, *86*(7), 1400–1409. doi:`10.1016/j.sigpro.2005.03.023`.

Bengtsson, H. (2021). A unifying framework for parallel and distributed processing in r using futures. *The R Journal*, *13*(2), 208–227. doi:`10.32614/RJ-2021-048`.

Birch, M. W. (1964). A new proof of the pearson-fisher theorem. *The Annals of Mathematical Statistics*, *35*(2), 817–824. doi:`10.1214/aoms/1177703581`.

Bishop, Y. M. M., Fienberg, S. E., & Holland, P. W. (2007). *Discrete Multivariate Analysis, Theory and Practice*. Springer New York, NY. doi:`10.1007/978-0-387-72806-3` Originally published by MIT Press, 1975.

Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, *7*(3), 200–217. doi:`10.1016/0041-5553(67)90040-7`.

Broniatowski, M., Toma, A., & Vajda, I. (2012). Decomposable pseudodistances and applications in statistical estimation. *Journal of Statistical Planning and Inference*, *142*(9), 2574–2585. doi:`10.1016/j.jspi.2012.03.019`.

Cavanaugh, J. E. (2004). Criteria for linear model selection based on kullback's symmetric divergence. *Australian & New Zealand Journal of Statistics*, *46*(2), 257–274. doi:`10.1111/j.1467-842X.2004.00328.x`.

Christensen, R. (1997). *Log−Linear Models and Logistic Regression*. (2nd ed.). Springer New York, NY. doi:`10.1007/b97647`.

Cochran, W. G. (1952). The χ^2 test of goodness of fit. *The Annals of Mathematical Statistics*, *23*(3), 315–345. doi:`10.1214/aoms/1177729380`.

Contreras-Reyes, J. E., Kahrari, F., & Cortés, D. D. (2021). On the modified skew-normal-cauchy distribution: Properties, inference and applications. *Communications in Statistics - Theory and Methods*, *50*(15), 3615–3631. doi:`10.1080/03610926.2019.1708942`.

Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory*. John Wiley & Sons, Inc. doi:`10.1002/047174882X`.

Cox, C. (1984). An elementary introduction to maximum likelihood estimation for multinomial models: Birch's theorem and the delta method. *The American Statistician*, *38*(4), 283–287. doi:`10.1080/00031305.1984.10483226`.

Cressie, N., & Read, T. R. C. (1984). Multinomial goodness-of-fit tests. *Journal of the Royal Statistical Society. Series B (Methodological)*, *46*(3), 440–464. doi:`10.1111/j.2517-6161.1984.tb01318.x`.

Csiszár, I. (1963). Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten. *Magyer Tud. Akad. Mat. Kutato Int. Koezl.*, *8*, 85–108.

Dahiya, R. C., & Gurland, J. (1973). How many classes in the Pearson chi-square test? *Journal of the American Statistical Association*, *68*(343), 707–712. doi:`10.1080/01621459.1973.10481409`.

Dale, J. R. (1986). Asymptotic normality of goodness-of-fit statistics for sparse product multinomials. *Journal of the Royal Statistical Society. Series B (Methodological)*, *48*(1), 48–59. doi:`10.1111/j.2517-6161.1986.tb01389.x`.

Dowle, M., & Srinivasan, A. (2022). *data.table: Extension of 'data.frame'*. URL: `https://r-datatable.com`.

Dragulescu, A., & Arendt, C. (2020). *xlsx: Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files*. URL: `https://CRAN.R-project.org/package=xlsx`.

Dubrova, Y. E., Grant, G., Chumak, A. A., Stezhka, V. A., & Karakasian, A. N. (2002). Elevated minisatellite mutation rate in the post-chernobyl families from ukraine. *The American Journal of Human Genetics*, *71*(4), 801–809. doi:`10.1086/342729`.

D'Agostino, R. B., & Stephens, M. A. (1986). *Goodness-of-Fit Techniques*. Routledge. doi:`10.1201/9780203753064`.

Ferentinos, K., & Papaioannou, T. (1979). Loss of information due to groupings. *Transactions of the Eighth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes, Volume C*, 87–94.

Ferentinos, K., & Papaioannou, T. (1983). Convexity of measures of information and loss of information due to grouping of observations. *Journal of Combinatorics, Information & System Sciences*, *8*(4), 286–294.

Ferguson, T. S. (1996). *A Course in Large Sample Theory*. Boca Raton: Chapman and Hall. doi:`10.1201/9781315136288`.

Fischer, A. (2010). Quantization and clustering with bregman divergences. *Journal of Multivariate Analysis*, *101*(9), 2207–2221. doi:`10.1016/j.jmva.2010.05.008`.

Fisher, R. A. (1924). The conditions under which $\mathcal{X}^2$ measures the discrepancy between observation and hypothesis. *Journal of the Royal Statistical Society*, *87*(3), 442–450. URL: `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2397-2335.1924.tb01691.x`.

Garnier, S., Ross, N., Rudis, R., Camargo, A. P., Sciaini, M., & Scherer, C. (2021). *viridis - Colorblind-Friendly Color Maps for R*. doi:`10.5281/zenodo.4679424`.

Geenens, G., & Simar, L. (2010). Nonparametric tests for conditional independence in two-way contingency tables. *Journal of Multivariate Analysis*, *101*(4), 765–788. doi:`10.1016/j.jmva.2009.12.017`.

Ghosh, A. (2016). Divergence based robust estimation of the tail index through an exponential regression model. *Statistical Methods & Applications*, *26*, 181–213. doi:`10.1007/s10260-016-0364-9`.

Ghosh, A., Maji, A., & Basu, A. (2013). Robust inference based on divergences in reliability systems. In I. Frenkel, A. Karagrigoriou, A. Lisnianski, & A. Kleyner (Eds.), *Applied Reliability Engineering and Risk Analysis* chapter 21. (pp. 290–307). John Wiley & Sons, Ltd. doi:`10.1002/9781118701881.ch21`.

Giner, G., & Smyth, G. K. (2016). statmod: probability calculations for the inverse gaussian distribution. *R Journal*, *8*(1), 339–351.

Gokhale, D. V., & Kullback, S. (1978). *The Information in Contingency Tables* volume 23. M. Dekker.

Harris, I. R., & Basu, A. (1994). Hellinger distance as a penalized log likelihood. *Communications in Statistics - Simulation and Computation*, *23*(4), 1097–1113. doi:`10.1080/03610919408813219`.

Harrison, R. H. (1985). Choosing the optimum number of classes in the chi-square test for arbitrary power levels. *Sankhyā: The Indian Journal of Statistics, Series B*, *47*(3), 319–324. URL: `https://www.jstor.org/stable/25052408`.

Hocking, T. D. (2021). *directlabels: Direct Labels for Multicolor Plots*. URL: `https://CRAN.R-project.org/package=directlabels`.

Holst, L. (1972). Asymptotic normality and efficiency for certain goodness-of-fit tests. *Biometrika*, *59*(1), 137–145. doi:`10.1093/biomet/59.1.137`.

Jiménez-Gamero, M. D., & Batsidis, A. (2017). Minimum distance estimators for count data based on the probability generating function with applications. *Metrika*, *80*(5), 503–545. doi:`10.1007/s00184-017-0614-3`.

Jiménez-Gamero, M. D., Pino-Mejías, R., Alba-Fernández, V., & Moreno-Rebollo, J. L. (2011). Minimum ϕ-divergence estimation in misspecified multinomial models. *Computational Statistics & Data Analysis*, *55*(12), 3365 – 3378. doi:`10.1016/j.csda.2011.06.035`.

Johnson, S. G. (2014). *The NLopt Nonlinear-optimization Package*. URL: `https://cran.r-project.org/web/packages/nloptr/nloptr.pdf`.

Jones, M. C., Hjort, N. L., Harris, I. R., & Basu, A. (2001). A comparison of related density-based minimum divergence estimators. *Biometrika*, *88*(3), 865–873. doi:`10.1093/biomet/88.3.865`.

Kagan, A. M. (1963). On the theory of Fisher's amount of information. *Soviet Mathematics. Doklady*, *4*, 991–993.

Karagrigoriou, A., & Meselidis, C. (2020). Inferential statistics based on measures of information and divergence. In *Statistical Topics and Stochastic Models for Dependent Data with Applications* chapter 10. (pp. 215–234). John Wiley & Sons, Ltd. doi:`10.1002/9781119779421.ch10`.

Kassambara, A. (2020). *ggpubr: 'ggplot2' Based Publication Ready Plots*. URL: `https://CRAN.R-project.org/package=ggpubr`.

Kateri, M. (2018). ϕ-divergence in contingency table analysis. *Entropy*, *20*(324). doi:`10.3390/e20050324`.

Kateri, M., & Papaioannou, T. (1996). Symmetry and asymmetry models for rectangular tables. *Biometrical Journal*, *38*(2), 203–220. doi:`10.1002/bimj.4710380208`.

Kateri, M., & Papaioannou, T. (1997). Asymmetry models for contingency tables. *Journal of the American Statistical Association*, *92*(439), 1124–1131. doi:`10.2307/2965577`.

Kateri, M., Papaioannou, T., & Ahmad, R. (1996). New association models for the analysis of sets of two–way contingency tables. *Statistica Applicata*, *8*(3), 537–551.

Kim, B., & Lee, S. (2014). Minimum density power divergence estimator for covariance matrix based on skew t distribution. *Statistical Methods & Applications*, *23*, 565–575. doi:`10.1007/s10260-014-0284-5`.

Klugman, S. A., Panjer, H. H., & Willmot, G. E. (2008). *Loss Models: From Data to Decisions*. (3rd ed.). John Wiley & Sons, Inc. doi:`10.1002/9780470391341`.

Koutras, M. V., Maravelakis, P. E., & Bersimis, S. (2008). Techniques for controlling bivariate grouped observations. *Journal of Multivariate Analysis*, *99*(7), 1474–1488. doi:`10.1016/j.jmva.2008.02.032`.

Koutrouvelis, I. A., & Karagrigoriou, A. (2012). Cumulant plots and goodness-of-fit tests for the inverse gaussian distribution. *Journal of Statistical Computation and Simulation*, *82*(3), 343–358. doi:`10.1080/00949655.2010.531018`.

Kullback, S. (2006). Minimum discrimination information (MDI) estimation. In *Encyclopedia of Statistical Sciences*. John Wiley & Sons, Ltd. doi:`10.1002/0471667196.ess1642.pub2`.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86. doi:10.1214/aoms/1177729694.

Kupperman, M. (1957). *Further Application of Information Theory to Multivariate Analysis and Statistical Inference*. Ph.D. thesis George Washington University.

Kupperman, M. (1958). Probabilities of hypotheses and information-statistics in sampling from exponential-class populations. *The Annals of Mathematical Statistics*, *29*(2), 571–575. doi:10.1214/aoms/1177706633.

Liese, F., & Vajda, I. (1987). *Convex Statistical Distances* volume 95. Teubner-Texte sur Mathematik, Leibzig.

Lin, N., & He, X. (2006). Robust and efficient estimation under data grouping. *Biometrika*, *93*(1), 99–112. doi:10.1093/biomet/93.1.99.

Lloyd, C. J. (2005). Estimating test power adjusted for size. *Journal of Statistical Computation and Simulation*, *75*(11), 921–933. doi:10.1080/00949650412331321160.

Mandal, A., & Basu, A. (2011). Multinomial goodness-of-fit tests under inlier modification. *Electronic Journal of Statistics*, *5*, 1846 – 1875. doi:10.1214/11-EJS658.

Mandal, A., Basu, A., & Pardo, L. (2010). Minimum disparity inference and the empty cell penalty: Asymptotic results. *Sankhya A*, *72*(2), 376–406. doi:10.1007/s13171-010-0022-5.

Mann, H. B., & Wald, A. (1942). On the choice of the number of class intervals in the application of the chi square test. *The Annals of Mathematical Statistics*, *13*(3), 306–317. doi:10.1214/aoms/1177731569.

Mantalos, P., Mattheou, K., & Karagrigoriou, A. (2010). Forecasting ARMA models: A comparative study of information criteria focusing on MDIC. *Journal of Statistical Computation and Simulation*, *80*(1), 61–73. doi:https://doi.org/10.1080/00949650802464137.

Mathai, A. M., & Rathie, P. N. (1975). *Basic Concepts in Information Theory and Statistics: Axiomatic Foundations and Applications*. Wiley Eastern Pvt.Ltd.

Mattheou, K., & Karagrigoriou, A. (2010). A new family of divergence measures for tests of fit. *Australian & New Zealand Journal of Statistics*, *52*(2), 187–200. doi:10.1111/j.1467-842X.2010.00574.x.

Mattheou, K., Lee, S., & Karagrigoriou, A. (2009). A model selection criterion based on the BHHJ measure of divergence. *Journal of Statistical Planning and Inference*, *139*(2), 228–235. doi:10.1016/j.jspi.2008.04.022.

Matusita, K. (1967). On the notion of affinity of several distributions and some of its applications. *Annals of the Institute of Statistical Mathematics*, *19*, 181–192. doi:10.1007/BF02911675.

McManus, D. A. (1991). Who invented local power analysis? *Econometric Theory*, *7*(2), 265–268. doi:10.1017/S026646660000445X.

Menéndez, M., Morales, D., Pardo, L., & Vajda, I. (2001). Approximations to powers of φ-disparity goodness-of-fit tests. *Communications in Statistics - Theory and Methods*, *30*(1), 105–134. doi:10.1081/STA-100001562.

Merková, M. (2015). Use of investment controlling and its impact into business performance. *Procedia Economics and Finance*, *34*, 608–614. doi:`10.1016/S2212-5671(15)01675-5`.

Meschiari, S. (2022). *latex2exp: Use LaTeX Expressions in Plots*. URL: `https://CRAN.R-project.org/package=latex2exp`.

Meselidis, C., & Karagrigoriou, A. (2020). Statistical inference for multinomial populations based on a double index family of test statistics. *Journal of Statistical Computation and Simulation*, *90*(10), 1773–1792. doi:`10.1080/00949655.2020.1747069`.

Meselidis, C., & Karagrigoriou, A. (2022). Contingency table analysis and inference via double index measures. *Entropy*, *24*(4), 477. doi:`10.3390/e24040477`.

Meselidis, C., Karagrigoriou, A., & Papaioannou, T. (2022). Data analysis based on entropies and measures. In K. N. Zafeiris, C. H. Skiadas, Y. Dimotikalis, A. Karagrigoriou, & C. Kragrigoriou-Vonta (Eds.), *Data Analysis and Related Applications: Theory and Practice* chapter 18. John Wiley & Sons, Ltd volume 1. URL: `https://www.iste.co.uk/book.php?id=1927`.

Morales, D., Pardo, L., & Vajda, I. (1995). Asymptotic divergence of estimates of discrete distributions. *Journal of Statistical Planning and Inference*, *48*(3), 347–369. doi:`10.1016/0378-3758(95)00013-Y`.

Müller, K., & Wickham, H. (2022). *tibble: Simple Data Frames*. URL: `https://tibble.tidyverse.org/`.

Nadarajah, S., & Zografos, K. (2003). Formulas for rényi information and related measures for univariate distributions. *Information Sciences*, *155*(1), 119–138. doi:`10.1016/S0020-0255(03)00156-7`.

Nakazawa, M. (2022). *fmsb: Functions for Medical Statistics Book with some Demographic Data*. URL: `https://CRAN.R-project.org/package=fmsb`.

Nayak, T. K. (1983). *Applications of Entropy Functions in Measurement and Analysis of Diversity*. Ph.D. thesis University of Pittsburgh.

Nayak, T. K. (1985). On diversity measures based on entropy functions. *Communications in Statistics - Theory and Methods*, *14*(1), 203–215. doi:`https://doi.org/10.1080/03610928508828905`.

Neath, A. A., Cavanaugh, J. E., & Weyhaupt, A. G. (2015). Model evaluation, discrepancy function estimation, and social choice theory. *Computational Statistics*, *30*(1), 231–249. doi:`10.1007/s00180-014-0532-z`.

Neyman, J. (1937). ″Smooth″test for goodness of fit. *Scandinavian Actuarial Journal*, *1937*(3-4), 149–199. doi:`10.1080/03461238.1937.10404821`.

Neyman, J. (1949). Contribution to the theory of the $\mathcal{X}^2$ test. In *Proceedings of the* 1^{st} *Berkeley Symposium on Mathematical Statistics and Probability* (pp. 239–273). University of California Press, Berkeley.

Neyman, J., & Pearson, E. S. (1928). On the use and interpretation of certain test criteria for purposes of statistical inference: Part i. *Biometrika*, *20A*(1/2), 175–240. doi:`10.1093/biomet/20A.1-2.175`.

Nikitin, Y. (1995). *Asymptotic Efficiency of Nonparametric Tests*. Cambridge University Press. doi:`10.1017/CBO9780511530081`.

Papaioannou, T. (2004). Measures of information. In *Encyclopedia of Statistical Sciences*. John Wiley & Sons, Ltd. doi:`10.1002/0471667196.ess1594`.

Papaioannou, T., Ferentinos, K., Menéndez, M. L., & Salicru, M. (1994). Discretization of (h, ϑ)-divergences. *Information Sciences*, *77*(3), 351–358. doi:`10.1016/0020-0255(94)90008-6`.

Papaioannou, T., Ferentinos, K., & Tsairidis, C. (2007). Some information theoretic ideas useful in statistical inference. *Methodology and Computing in Applied Probability*, *77*(9), 307–323. doi:`10.1007/s11009-007-9017-7`.

Pardo, J., Pardo, L., & Zografos, K. (2002). Minimum φ-divergence estimators with constraints in multinomial populations. *Journal of Statistical Planning and Inference*, *104*(1), 221–237. doi:`10.1016/S0378-3758(01)00113-6`.

Pardo, J. A. (2010). An approach to multiway contingency tables based on φ-divergence test statistics. *Journal of Multivariate Analysis*, *101*(10), 2305–2319. doi:`10.1016/j.jmva.2010.06.003`.

Pardo, L. (2006). *Statistical Inference Based on Divergence Measures*. Chapman and Hall/CRC. doi:`10.1201/9781420034813`.

Parodi, P. (2014). *Pricing in General Insurance*. Chapman and Hall/CRC. doi:`10.1201/b17525`.

Patra, S., Maji, A., Basu, A., & Pardo, L. (2013). The power divergence and the density power divergence families: the mathematical connection. *Sankhya B*, *75*, 16–28. doi:`10.1007/s13571-012-0050-3`.

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosopy Magazine*, *50*, 157–172.

Pitman, E. (1979). *Some Basic Theory for Statistical Inference*. New York: Wiley. doi:`10.1201/9781351076777`.

Qiao, Y., & Minematsu, N. (2010). A study on invariance of f-divergence and its application to speech recognition. *IEEE Transactions on Signal Processing*, *58*(7), 3884–3890. doi:`10.1109/TSP.2010.2047340`.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing Vienna, Austria. URL: `https://www.R-project.org/`.

Rao, C. R. (1963). Criteria of estimation in large samples. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, *25*(2), 189–206. URL: `http://www.jstor.org/stable/25049263`.

Rathie, P. N., & Kannappan, P. (1972). A directed-divergence function of type β. *Information and Control*, *20*(1), 38–45. doi:`10.1016/S0019-9958(72)90260-4`.

Read, T. R. C., & Cressie, N. A. C. (1988). *Goodness-of-Fit Statistics for Discrete Multivariate Data*. New York: Springer-Verlag. doi:`10.1007/978-1-4612-4578-0`.

Rudis, B. (2020). *hrbrthemes: Additional Themes, Theme Components and Utilities for 'ggplot2'*. URL: `https://CRAN.R-project.org/package=hrbrthemes`.

Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics* (pp. 547–561). Berkeley, Calif.: University of California Press.

Sachlas, A., & Papaioannou, T. (2014). Residual and past entropy in actuarial science and survival models. *Methodology and Computing in Applied Probability*, *16*, 79–99. doi:`10.1007/s11009-012-9300-0`.

Salicru, M., Morales, D., Menendez, M. L., & Pardo, L. (1994). On the applications of divergence type measures in testing statistical hypotheses. *Journal of Multivariate Analysis*, *51*(2), 372–391. doi:`10.1006/jmva.1994.1068`.

Sarkar, D. (2008). *Lattice: Multivariate Data Visualization with R*. New York: Springer. URL: `http://lmdvr.r-forge.r-project.org`.

Shang, J., & Cavanaugh, J. E. (2008). Bootstrap variants of the akaike information criterion for mixed model selection. *Computational Statistics & Data Analysis*, *52*(4), 2004–2021. doi:`10.1016/j.csda.2007.06.019`.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423. doi:`10.1002/j.1538-7305.1948.tb01338.x`.

Shiva, S. G. S., Ahmed, N. U., & Georganas, N. D. (1973). Order preserving measures of information. *Journal of Applied Probability*, *10*(3), 666–670. doi:`10.2307/3212786`.

Simon, G. (1973). Additivity of information in exponential family probability laws. *Journal of the American Statistical Association*, *68*(342), 478–482. doi:`10.1080/01621459.1973.10482459`.

Slowikowski, K. (2021). *ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'*. URL: `https://CRAN.R-project.org/package=ggrepel`.

Stehlík, M., Economou, P., Papić, L., Aronov, J., Nicolis, O., Antoch, J., Cézová, E., & Kisel'ák, J. (2018). Statistical testing of availability for mining technological systems with air quality constraints. *Applied Stochastic Models in Business and Industry*, *34*(3), 278–292. doi:`10.1002/asmb.2337`.

Stehlík, M., Střelec, L., & Thulin, M. (2014). On robust testing for normality in chemometrics. *Chemometrics and Intelligent Laboratory Systems*, *130*, 98–108. doi:`10.1016/j.chemolab.2013.10.010`.

Sugasawa, S., & Yonekura, S. (2021). On selection criteria for the tuning parameter in robust divergence. *Entropy*, *23*(9). doi:`10.3390/e23091147`.

Tay, S. Y., Ng, C. M., & Ong, S. H. (2019). Parameter estimation by minimizing a probability generating function-based power divergence. *Communications in Statistics - Simulation and Computation*, *48*(10), 2898–2912. doi:`10.1080/03610918.2018.1468462`.

Toma, A. (2008). Minimum hellinger distance estimators for multivariate distributions from the johnson system. *Journal of Statistical Planning and Inference*, *138*(3), 803 – 816. doi:`10.1016/j.jspi.2007.05.033`.

Toma, A. (2009). Optimal robust m-estimators using divergences. *Statistics & Probability Letters*, *79*(1), 1–5. doi:`10.1016/j.spl.2008.04.011`.

Toma, A. (2014). Model selection criteria using divergences. *Entropy*, *16*(5), 2686–2698. doi:`10.3390/e16052686`.

Toma, A., & Broniatowski, M. (2011). Dual divergence estimators and tests: Robustness results. *Journal of Multivariate Analysis*, *102*(1), 20–36. doi:`10.1016/j.jmva.2010.07.010`.

Toma, A., Karagrigoriou, A., & Trentou, P. (2020). Robust model selection criteria based on pseudodistances. *Entropy (Basel, Switzerland)*, *22*(3). doi:`10.3390/e22030304`.

Toma, A., & Leoni-Aubin, S. (2010). Robust tests based on dual divergence estimators and saddlepoint approximations. *Journal of Multivariate Analysis*, *101*(5), 1143–1155. doi:`10.1016/j.jmva.2009.11.001`.

Toma, A., & Leoni-Aubin, S. (2014). Minimum pseudodistance estimators and applications to portfolio optimization. In V. G. J. R. Bozeman, & C. H. Skiadas (Eds.), *New Perspectives on Stochastic Modeling and Data Analysis* (pp. 79–86). Athens, Greece: ISAST International.

Tsairidis, C., Ferentinos, K., & Papaioannou, T. (1996). Information and random censoring. *Information Sciences*, *92*(1), 159–174. doi:`10.1016/0020-0255(96)00055-2`.

Tsairidis, C., Zografos, K., & Ferentinos, K. (2001). Information in quantal response data and random censoring. *Annals of the Institute of Statistical Mathematics*, *53*(3), 528–542. doi:`10.1023/A:1014621229708`.

Urbanek, S. (2021). *rJava: Low-Level R to Java Interface*. URL: `https://CRAN.R-project.org/package=rJava`.

Vassiliou, A., Tambouratzis, D. G., Koutras, M. V., & Bersimis, S. (2004). A new similarity measure and its use in determining the number of clusters in a multivariate data set. *Communications in Statistics - Theory and Methods*, *33*(7), 1643–1666. doi:`10.1081/STA-120037266`.

Victoria-Feser, M., & Ronchetti, E. (1997). Robust estimation for grouped data. *Journal of the American Statistical Association*, *92*(437), 333–340. doi:`10.1080/01621459.1997.10473631`.

Vonta, F., & Karagrigoriou, A. (2010). Generalized measures of divergence in survival analysis and reliability. *Journal of Applied Probability*, *47*(1), 216–234. doi:`10.1239/jap/1269610827`.

Vonta, F., Mattheou, K., & Karagrigoriou, A. (2012). On properties of the (ϕ, α)-power divergence family with applications in goodness of fit tests. *Methodology and Computing in Applied Probability*, *14*(2), 335–356. doi:`10.1007/s11009-010-9205-8`.

Vonta, I., & Karagrigoriou, A. (2014). Goodness-of-fit tests via ϕ-measures of divergence for censored data. *Journal of Statistical Computation and Simulation*, *84*(5), 946–963. doi:`10.1080/00949655.2012.733396`.

Warwick, J., & Jones, M. C. (2005). Choosing a robustness tuning parameter. *Journal of Statistical Computation and Simulation*, *75*(7), 581–588. doi:`10.1080/00949650412331299120`.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. URL: `https://ggplot2.tidyverse.org`.

Wickham, H. (2018). *scales: Scale Functions for Visualization*. URL: `https://CRAN.R-project.org/package=scales`.

Wickham, H., François, R., Henry, L., & Müller, K. (2022). *dplyr: A Grammar of Data Manipulation*. URL: `https://dplyr.tidyverse.org`.

Xiao, N. (2018). *ggsci: Scientific Journal and Sci-Fi Themed Color Palettes for 'ggplot2'*. URL: `https://CRAN.R-project.org/package=ggsci`.

Znaor, A., Brennan, P., Gajalakshmi, V., Mathew, A., Shanta, V., Varghese, C., & Boffetta, P. (2003). Independent and combined effects of tobacco smoking, chewing and alcohol drinking on the risk of oral, pharyngeal and esophageal cancers in indian men. *International Journal of Cancer*, *105*(5), 681–686. doi:`10.1002/ijc.11114`.

Zografos, K., Ferentinos, K., & Papaioannou, T. (1986). Discrete approximations to the csiszár, renyi, and fisher measures of information. *Canadian Journal of Statistics*, *14*(4), 355–366. doi:`10.2307/3315194`.

Zografos, K., Ferentinos, K., & Papaioannou, T. (1990). Divergence statistics: Sampling properties and multinomial goodness of fit and divergence tests. *Communications in Statistics - Theory and Methods*, *19*(5), 1785–1802. doi:`10.1080/03610929008830290`.

Appendix A

Further Results Regarding the Comparison of Test Statistics

In this appendix, all results regarding the comparison of the test statistics of Section 2.4.2 are presented.

Table A.1: Size (S) and powers (%) for the **Equiprobable (1/3, 1/3, 1/3)** model.

Shape parameter	$T_{\Phi_1}^{0.10}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$	$G^2(\hat{\boldsymbol{\theta}})$	$X^2(\hat{\boldsymbol{\theta}}_{X^2})$	$CR(\hat{\boldsymbol{\theta}}_{\mathrm{CR}}^{\lambda})$
			$N = 25$			
(S) Gamma(0.44)	4.74	4.30	4.93	4.83	6.12	5.63
Wei(0.77)	12.00	11.89	12.27	12.01	12.43	12.21
LN(0.729)	13.23	11.88	13.44	13.44	17.44	16.00
IG(1)	12.74	11.54	12.96	12.97	16.75	15.48
(S) Gamma(1)	4.88	4.38	4.94	4.93	6.21	5.72
Wei(1)	5.12	4.72	5.25	5.19	6.70	6.27
LN(0.551)	36.61	33.92	37.06	37.09	44.69	42.33
IG(2.25)	31.67	29.14	32.05	32.05	39.24	36.91
(S) Gamma(2)	4.71	4.36	4.84	4.78	6.14	5.63
Wei(1.259)	9.14	8.15	9.31	9.29	12.33	11.19
LN(0.423)	69.06	66.46	69.48	69.50	77.25	74.84
IG(4.5)	64.53	61.56	65.11	65.13	72.93	70.47
			$N = 50$			
(S) Gamma(0.44)	4.31	4.16	4.37	4.34	4.85	4.70
Wei(0.77)	21.46	21.54	21.52	21.42	20.87	21.02
LN(0.729)	18.67	17.40	18.87	18.88	22.09	20.92
IG(1)	17.61	16.54	17.87	17.87	21.19	20.20
(S) Gamma(1)	4.14	3.98	4.20	4.17	4.59	4.41
Wei(1)	4.47	4.33	4.50	4.49	4.95	4.76
LN(0.551)	63.94	61.89	64.32	64.34	69.05	67.67
IG(2.25)	56.15	54.11	56.50	56.50	61.32	59.63
(S) Gamma(2)	4.29	4.10	4.33	4.31	4.76	4.59
Wei(1.259)	9.66	8.94	9.79	9.79	11.62	11.00
LN(0.423)	95.83	95.20	95.92	95.93	97.12	96.78
IG(4.5)	93.65	92.94	93.85	93.87	95.47	94.94
			$N = 75$			
(S) Gamma(0.44)	4.77	4.64	4.81	4.80	5.25	5.12
Wei(0.77)	23.16	23.28	23.17	23.06	22.50	22.68
LN(0.729)	35.50	34.53	35.75	35.75	38.74	37.73
IG(1)	32.97	32.15	33.10	33.10	35.88	35.10
(S) Gamma(1)	4.64	4.45	4.65	4.65	5.18	4.99
Wei(1)	4.87	4.75	4.92	4.91	5.30	5.14
LN(0.551)	89.71	89.13	89.82	89.82	91.37	90.96
IG(2.25)	84.08	83.26	84.19	84.19	86.29	85.55
(S) Gamma(2)	5.09	4.94	5.09	5.11	5.60	5.37
Wei(1.259)	18.62	17.91	18.80	18.80	20.95	20.09
LN(0.423)	99.86	99.86	99.86	99.86	99.86	99.86
IG(4.5)	99.83	99.80	99.83	99.83	99.85	99.84
			$N = 100$			
(S) Gamma(0.44)	4.91	4.90	4.93	4.90	5.22	5.07
Wei(0.77)	31.36	31.64	31.36	31.29	30.70	30.85
LN(0.729)	42.01	41.26	42.14	42.14	45.39	44.19
IG(1)	37.45	36.65	37.59	37.59	40.35	39.37
(S) Gamma(1)	5.11	5.01	5.14	5.14	5.40	5.32
Wei(1)	4.59	4.48	4.61	4.58	4.87	4.85
LN(0.551)	95.62	95.36	95.65	95.65	96.41	96.17
IG(2.25)	91.42	91.06	91.46	91.46	92.65	92.27
(S) Gamma(2)	5.08	4.98	5.13	5.12	5.27	5.18
Wei(1.259)	22.04	21.48	22.22	22.22	24.46	23.66
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	99.99	99.99	99.99	99.99	100.00	100.00
			$N = 500$			
(S) Gamma(0.44)	4.61	4.57	4.61	4.61	4.66	4.62
Wei(0.77)	88.79	88.81	88.79	88.79	88.51	88.61
LN(0.729)	98.39	98.34	98.41	98.41	98.49	98.45
IG(1)	96.04	96.00	96.05	96.05	96.15	96.13
(S) Gamma(1)	4.62	4.63	4.62	4.61	4.77	4.72
Wei(1)	4.66	4.64	4.67	4.67	4.77	4.75
LN(0.551)	100.00	100.00	100.00	100.00	100.00	100.00
IG(2.25)	100.00	100.00	100.00	100.00	100.00	100.00
(S) Gamma(2)	4.71	4.68	4.71	4.71	4.78	4.79
Wei(1.259)	80.85	80.65	80.92	80.92	81.62	81.39
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00

Table A.2: Size (S) and powers (%) for the **Symmetric (0.2, 0.6, 0.2)** model.

Shape parameter	$T_{\Phi_1}^{0.10}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$	$G^2(\hat{\boldsymbol{\theta}})$	$X^2(\hat{\boldsymbol{\theta}}_{X^2})$	$CR(\hat{\boldsymbol{\theta}}_{\mathrm{CR}}^{\lambda})$
			$N = 25$			
(S) Gamma(0.44)	5.47	5.33	5.66	5.43	6.08	5.83
Wei(0.77)	13.99	10.75	16.05	14.68	17.39	16.52
LN(0.729)	29.69	30.66	29.48	29.44	30.10	29.85
IG(1)	27.88	28.73	27.71	27.65	28.30	28.08
(S) Gamma(1)	5.53	5.51	5.69	5.50	6.09	5.88
Wei(1)	5.37	5.36	5.60	5.43	5.92	5.78
LN(0.551)	77.33	78.33	77.14	77.14	77.78	77.53
IG(2.25)	71.24	72.30	70.98	70.94	71.70	71.35
(S) Gamma(2)	5.05	5.05	5.28	5.07	5.73	5.54
Wei(1.259)	18.85	19.69	18.72	18.71	19.18	18.94
LN(0.423)	98.69	98.82	98.68	98.68	98.78	98.72
IG(4.5)	98.29	98.48	98.23	98.22	98.37	98.31
			$N = 50$			
(S) Gamma(0.44)	5.33	5.08	5.59	5.46	5.73	5.64
Wei(0.77)	28.66	25.01	30.41	29.37	31.84	31.03
LN(0.729)	46.77	47.20	46.72	46.71	46.59	46.59
IG(1)	42.82	43.28	42.74	42.72	42.42	42.43
(S) Gamma(1)	5.16	4.93	5.30	5.19	5.45	5.34
Wei(1)	5.15	4.81	5.34	5.22	5.55	5.41
LN(0.551)	97.12	97.20	97.11	97.11	97.08	97.06
IG(2.25)	94.34	94.54	94.34	94.34	94.32	94.32
(S) Gamma(2)	5.21	4.89	5.37	5.26	5.54	5.45
Wei(1.259)	29.13	29.48	29.05	29.04	28.89	28.89
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	99.98	99.98	99.98	99.98	99.98	99.98
			$N = 75$			
(S) Gamma(0.44)	4.95	4.62	5.13	5.04	5.23	5.19
Wei(0.77)	40.70	36.65	42.34	41.52	43.55	43.07
LN(0.729)	60.78	60.95	60.80	60.80	60.47	60.54
IG(1)	55.64	55.82	55.64	55.64	55.23	55.36
(S) Gamma(1)	5.04	4.86	5.20	5.12	5.29	5.21
Wei(1)	4.99	4.62	5.16	5.08	5.31	5.23
LN(0.551)	99.70	99.70	99.70	99.70	99.68	99.70
IG(2.25)	99.03	99.05	99.03	99.03	98.98	98.99
(S) Gamma(2)	5.19	4.82	5.34	5.28	5.58	5.45
Wei(1.259)	39.19	39.28	39.19	39.19	38.74	38.83
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00
			$N = 100$			
(S) Gamma(0.44)	5.18	4.82	5.39	5.25	5.49	5.42
Wei(0.77)	52.19	48.66	53.50	52.96	54.55	54.01
LN(0.729)	72.87	72.78	72.88	72.88	72.64	72.68
IG(1)	65.88	65.76	65.90	65.90	65.50	65.62
(S) Gamma(1)	5.19	4.78	5.35	5.31	5.42	5.35
Wei(1)	4.93	4.67	5.08	5.05	5.24	5.14
LN(0.551)	99.98	99.98	99.98	99.98	99.98	99.98
IG(2.25)	99.90	99.90	99.90	99.90	99.90	99.90
(S) Gamma(2)	5.10	4.76	5.27	5.23	5.34	5.30
Wei(1.259)	48.98	48.92	49.01	49.02	48.52	48.66
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00
			$N = 500$			
(S) Gamma(0.44)	4.42	3.92	4.56	4.53	4.56	4.59
Wei(0.77)	99.29	99.19	99.34	99.32	99.36	99.36
LN(0.729)	99.99	99.99	99.99	99.99	99.99	99.99
IG(1)	99.84	99.81	99.84	99.84	99.84	99.84
(S) Gamma(1)	5.22	4.66	5.31	5.29	5.22	5.29
Wei(1)	5.06	4.56	5.25	5.24	5.28	5.29
LN(0.551)	100.00	100.00	100.00	100.00	100.00	100.00
IG(2.25)	100.00	100.00	100.00	100.00	100.00	100.00
(S) Gamma(2)	5.15	4.67	5.34	5.34	5.33	5.34
Wei(1.259)	99.11	99.06	99.12	99.12	99.10	99.11
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00

Table A.3: Size (S) and powers (%) for the **Left Skewed (0.1, 0.5, 0.4)** model.

Shape parameter	$T^{0.10}_{\Phi_1}(\hat{\boldsymbol{\theta}}^{0.01}_{\Phi_2})$	$T^{0.50}_{\Phi_1}(\hat{\boldsymbol{\theta}}^{0.01}_{\Phi_2})$	$T^{0.01}_{\Phi_1}(\hat{\boldsymbol{\theta}}^{0.50}_{\Phi_2})$	$G^2(\hat{\boldsymbol{\theta}})$	$X^2(\hat{\boldsymbol{\theta}}_{X^2})$	$CR(\hat{\boldsymbol{\theta}}^{\lambda}_{\text{CR}})$
$N = 25$						
(S) Gamma(0.44)	3.99	4.06	5.74	3.84	4.58	4.33
Wei(0.77)	10.55	10.53	16.17	10.27	11.56	11.14
LN(0.729)	25.44	24.50	26.03	25.61	27.68	26.99
IG(1)	31.59	30.54	32.24	31.75	34.40	33.41
(S) Gamma(1)	3.95	3.90	5.61	3.90	4.65	4.41
Wei(1)	4.01	3.85	5.84	3.97	4.75	4.49
LN(0.551)	64.73	63.18	65.51	64.99	67.63	66.79
IG(2.25)	63.68	62.08	64.48	63.93	66.54	65.74
(S) Gamma(2)	3.97	3.58	5.73	3.91	4.74	4.49
Wei(1.259)	7.76	7.41	8.18	7.79	8.80	8.35
LN(0.423)	94.44	93.98	94.67	94.53	95.38	95.16
IG(4.5)	93.45	92.86	93.65	93.48	94.48	94.07
$N = 50$						
(S) Gamma(0.44)	5.02	5.75	6.33	4.76	5.13	5.03
Wei(0.77)	14.00	14.24	19.48	13.69	14.97	14.58
LN(0.729)	61.96	63.01	62.70	61.57	61.75	61.70
IG(1)	72.44	73.58	73.09	72.12	72.35	72.30
(S) Gamma(1)	5.46	5.79	6.71	5.31	5.61	5.55
Wei(1)	5.35	5.60	6.65	5.12	5.46	5.37
LN(0.551)	97.84	98.00	97.98	97.78	97.81	97.79
IG(2.25)	97.13	97.39	97.32	97.11	97.13	97.13
(S) Gamma(2)	5.05	5.17	6.12	4.94	5.15	5.06
Wei(1.259)	18.46	19.54	19.07	18.26	18.40	18.35
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	99.99	99.99	99.99	99.99	99.99	99.99
$N = 75$						
(S) Gamma(0.44)	4.45	5.21	5.70	4.24	4.45	4.34
Wei(0.77)	23.32	24.24	29.18	22.67	23.89	23.57
LN(0.729)	73.86	75.54	74.59	73.42	73.51	73.47
IG(1)	84.14	85.62	84.71	83.80	83.94	83.89
(S) Gamma(1)	4.63	5.19	5.94	4.41	4.65	4.59
Wei(1)	4.74	5.18	5.98	4.55	4.72	4.68
LN(0.551)	99.71	99.75	99.74	99.70	99.70	99.70
IG(2.25)	99.69	99.74	99.72	99.68	99.68	99.68
(S) Gamma(2)	4.45	4.68	5.94	4.38	4.67	4.56
Wei(1.259)	18.05	19.81	18.69	17.56	17.67	17.62
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00
$N = 100$						
(S) Gamma(0.44)	5.30	6.08	6.31	4.99	5.13	5.10
Wei(0.77)	24.92	25.99	30.88	24.53	25.59	25.12
LN(0.729)	90.26	91.28	90.91	89.94	89.78	89.81
IG(1)	95.23	95.86	95.43	95.09	95.00	95.04
(S) Gamma(1)	5.14	5.82	6.24	4.98	5.13	5.08
Wei(1)	5.24	5.84	6.45	5.06	5.25	5.16
LN(0.551)	99.98	99.98	99.98	99.98	99.98	99.98
IG(2.25)	99.98	99.98	99.98	99.98	99.98	99.98
(S) Gamma(2)	5.00	5.35	6.06	4.89	5.09	5.02
Wei(1.259)	28.06	30.38	28.92	27.48	27.36	27.42
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00
$N = 500$						
(S) Gamma(0.44)	5.31	6.43	6.24	4.96	4.96	4.95
Wei(0.77)	84.36	85.46	86.67	83.81	84.15	84.06
LN(0.729)	100.00	100.00	100.00	100.00	100.00	100.00
IG(1)	100.00	100.00	100.00	100.00	100.00	100.00
(S) Gamma(1)	5.28	6.08	6.26	5.03	5.05	5.03
Wei(1)	5.21	6.06	6.35	4.89	4.95	4.90
LN(0.551)	100.00	100.00	100.00	100.00	100.00	100.00
IG(2.25)	100.00	100.00	100.00	100.00	100.00	100.00
(S) Gamma(2)	5.44	6.05	6.65	5.21	5.26	5.24
Wei(1.259)	84.54	86.25	85.60	83.94	83.75	83.82
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00

Table A.4: Size (S) and powers (%) for the **Right Skewed (0.4, 0.5, 0.1)** model.

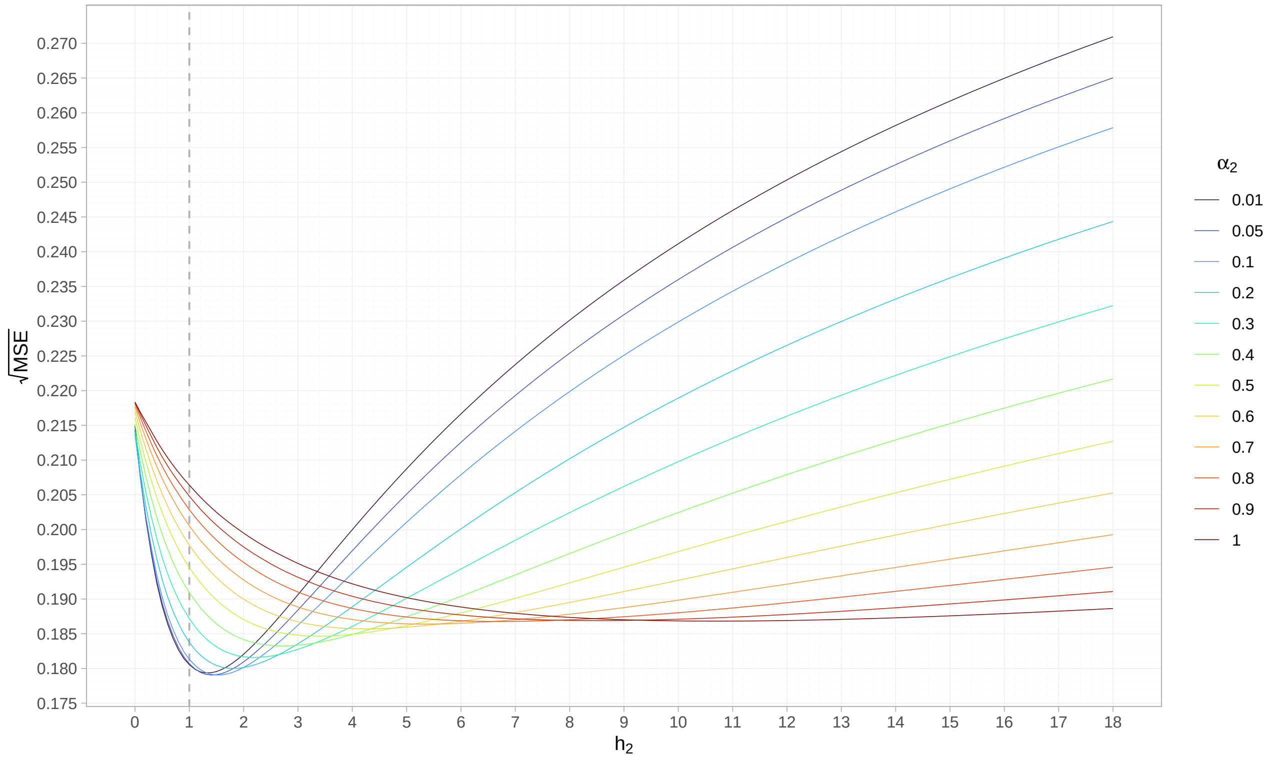
Shape parameter	$T_{\Phi_1}^{0.10}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.50}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.01})$	$T_{\Phi_1}^{0.01}(\hat{\boldsymbol{\theta}}_{\Phi_2}^{0.50})$	$G^2(\hat{\boldsymbol{\theta}})$	$X^2(\hat{\boldsymbol{\theta}}_{X^2})$	$CR(\hat{\boldsymbol{\theta}}_{\mathrm{CR}}^{\lambda})$
$N = 25$						
(S) Gamma(0.44)	3.53	2.13	4.53	3.97	5.33	4.94
Wei(0.77)	22.99	15.25	27.41	24.52	30.09	28.55
LN(0.729)	9.97	8.55	10.84	10.47	10.82	10.74
IG(1)	8.40	6.76	9.18	8.76	9.37	9.19
(S) Gamma(1)	3.49	2.31	4.36	3.76	4.93	4.54
Wei(1)	4.20	2.67	5.02	4.46	5.64	5.22
LN(0.551)	42.88	39.57	43.88	43.72	43.59	43.62
IG(2.25)	34.04	30.95	34.96	34.77	34.49	34.53
(S) Gamma(2)	3.46	2.47	4.66	3.79	4.88	4.62
Wei(1.259)	11.65	10.26	12.06	12.01	11.99	12.00
LN(0.423)	84.57	82.27	85.31	85.18	85.12	85.12
IG(4.5)	81.09	78.37	81.93	81.73	81.63	81.66
$N = 50$						
(S) Gamma(0.44)	4.61	3.66	5.34	4.95	5.19	5.10
Wei(0.77)	30.67	21.64	35.41	32.07	37.44	35.96
LN(0.729)	22.36	21.04	23.20	22.69	21.30	21.75
IG(1)	15.52	14.21	16.46	15.94	14.86	15.20
(S) Gamma(1)	4.95	4.22	5.72	5.23	5.37	5.34
Wei(1)	5.05	3.84	5.84	5.33	5.28	5.36
LN(0.551)	82.16	80.78	83.02	82.51	80.82	81.40
IG(2.25)	73.18	71.73	74.15	73.57	71.82	72.31
(S) Gamma(2)	5.33	4.36	6.20	5.58	5.63	5.66
Wei(1.259)	32.07	30.53	33.23	32.55	30.58	31.35
LN(0.423)	99.73	99.69	99.73	99.73	99.70	99.70
IG(4.5)	99.49	99.40	99.56	99.51	99.40	99.44
$N = 75$						
(S) Gamma(0.44)	4.34	3.02	5.15	4.62	5.11	4.98
Wei(0.77)	52.79	45.09	57.59	54.21	58.52	57.24
LN(0.729)	21.62	20.83	22.47	21.80	20.78	21.27
IG(1)	13.50	12.54	14.40	13.75	13.07	13.30
(S) Gamma(1)	4.52	3.57	5.45	4.77	5.08	5.00
Wei(1)	4.45	3.31	5.30	4.73	5.03	4.93
LN(0.551)	91.42	90.79	91.91	91.57	90.65	90.93
IG(2.25)	82.79	81.69	83.62	83.04	81.58	81.99
(S) Gamma(2)	4.49	3.69	5.55	4.69	5.16	5.03
Wei(1.259)	35.30	33.83	36.58	35.74	33.63	34.31
LN(0.423)	99.99	99.99	99.99	99.99	99.99	99.99
IG(4.5)	99.98	99.98	99.98	99.98	99.98	99.98
$N = 100$						
(S) Gamma(0.44)	4.85	3.85	5.62	5.15	5.16	5.09
Wei(0.77)	57.96	50.00	62.28	59.33	62.83	61.80
LN(0.729)	31.08	30.15	32.55	31.43	29.84	30.42
IG(1)	18.78	17.89	19.84	19.06	18.16	18.45
(S) Gamma(1)	5.14	4.39	6.07	5.41	5.44	5.39
Wei(1)	4.89	4.16	5.50	5.07	5.07	5.11
LN(0.551)	98.11	97.96	98.23	98.13	97.88	97.99
IG(2.25)	93.75	93.34	94.33	93.89	93.22	93.42
(S) Gamma(2)	5.28	4.50	5.99	5.43	5.47	5.48
Wei(1.259)	53.91	52.30	55.61	54.36	51.88	52.75
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00
$N = 500$						
(S) Gamma(0.44)	4.61	3.55	5.67	4.97	5.06	5.08
Wei(0.77)	99.81	99.78	99.86	99.82	99.83	99.83
LN(0.729)	80.56	79.44	82.13	80.95	80.14	80.46
IG(1)	45.04	43.14	47.19	45.59	44.55	44.96
(S) Gamma(1)	4.61	3.76	5.51	4.80	4.81	4.80
Wei(1)	4.93	3.81	5.94	5.11	5.18	5.18
LN(0.551)	100.00	100.00	100.00	100.00	100.00	100.00
IG(2.25)	100.00	100.00	100.00	100.00	100.00	100.00
(S) Gamma(2)	4.60	4.02	5.61	4.76	4.79	4.76
Wei(1.259)	99.53	99.45	99.63	99.57	99.49	99.50
LN(0.423)	100.00	100.00	100.00	100.00	100.00	100.00
IG(4.5)	100.00	100.00	100.00	100.00	100.00	100.00

Appendix B

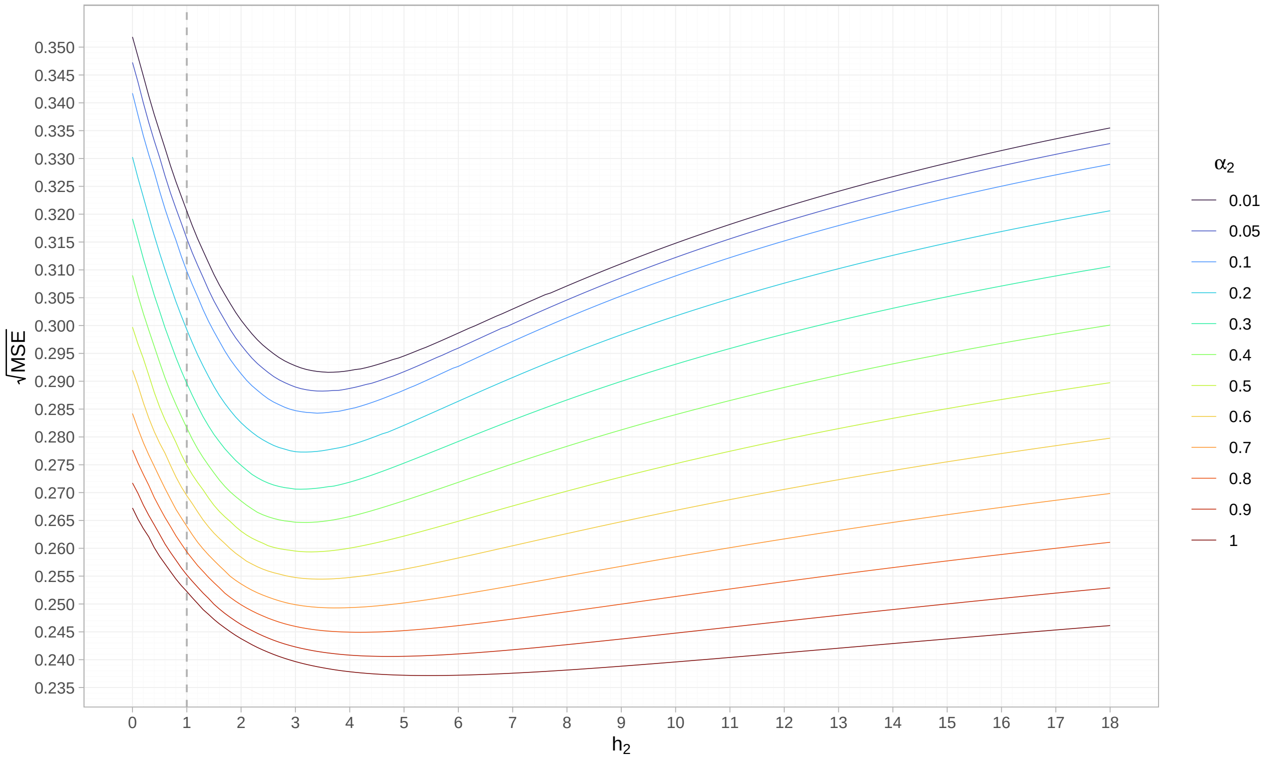
Further Results Regarding the MM(Φ, α)PDE

In this appendix results regarding the three remaining models mentioned in Section 4.3 are presented.

B.1 Truncated Poisson model

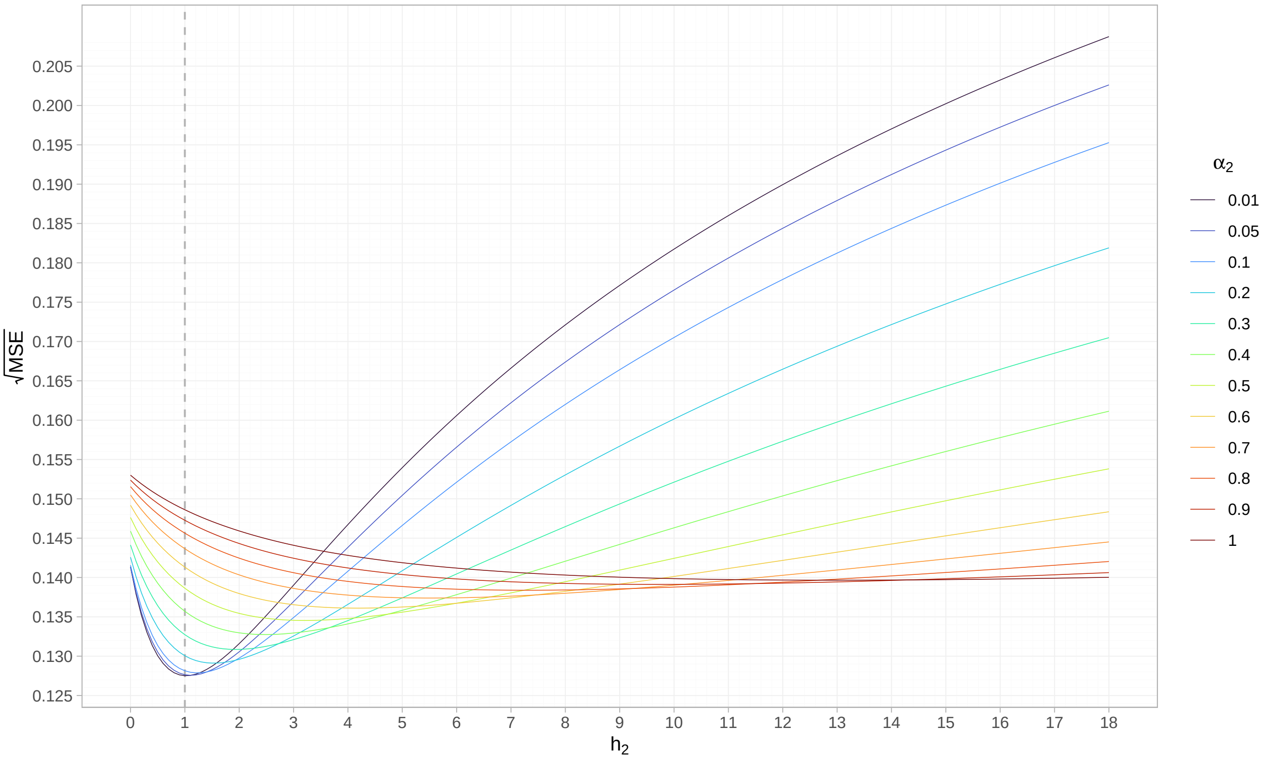


(a)

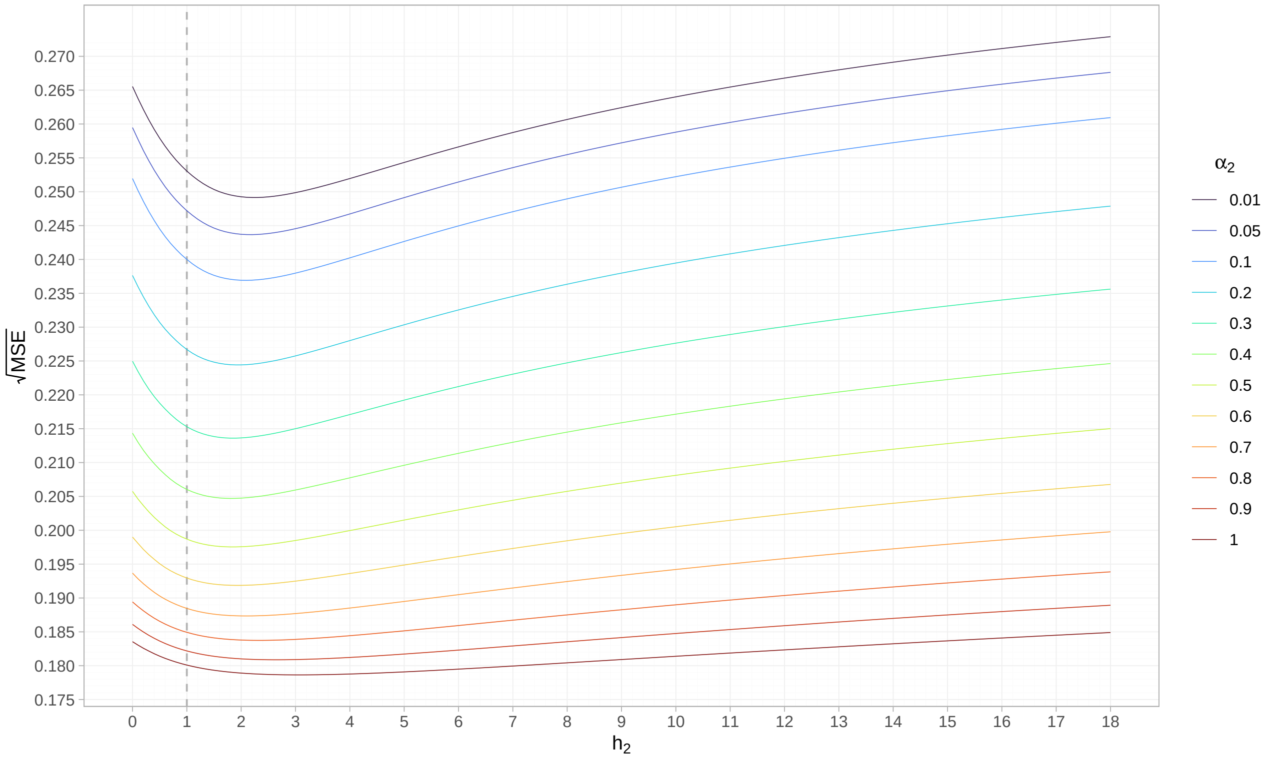


(b)

Figure B.1: Root mean squared error for the MM(Φ, α)PDE. $N = 15$.
(a) Under no contamination. (b) Under contamination.

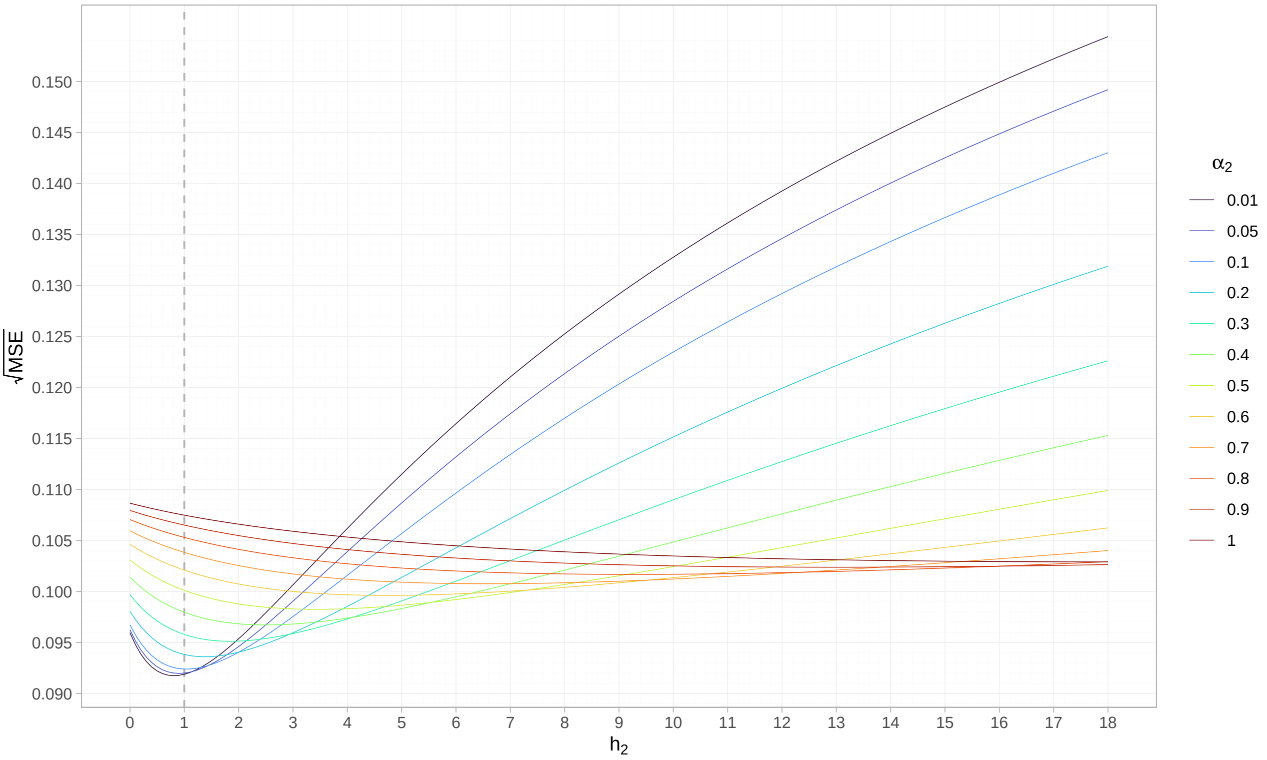


(a)

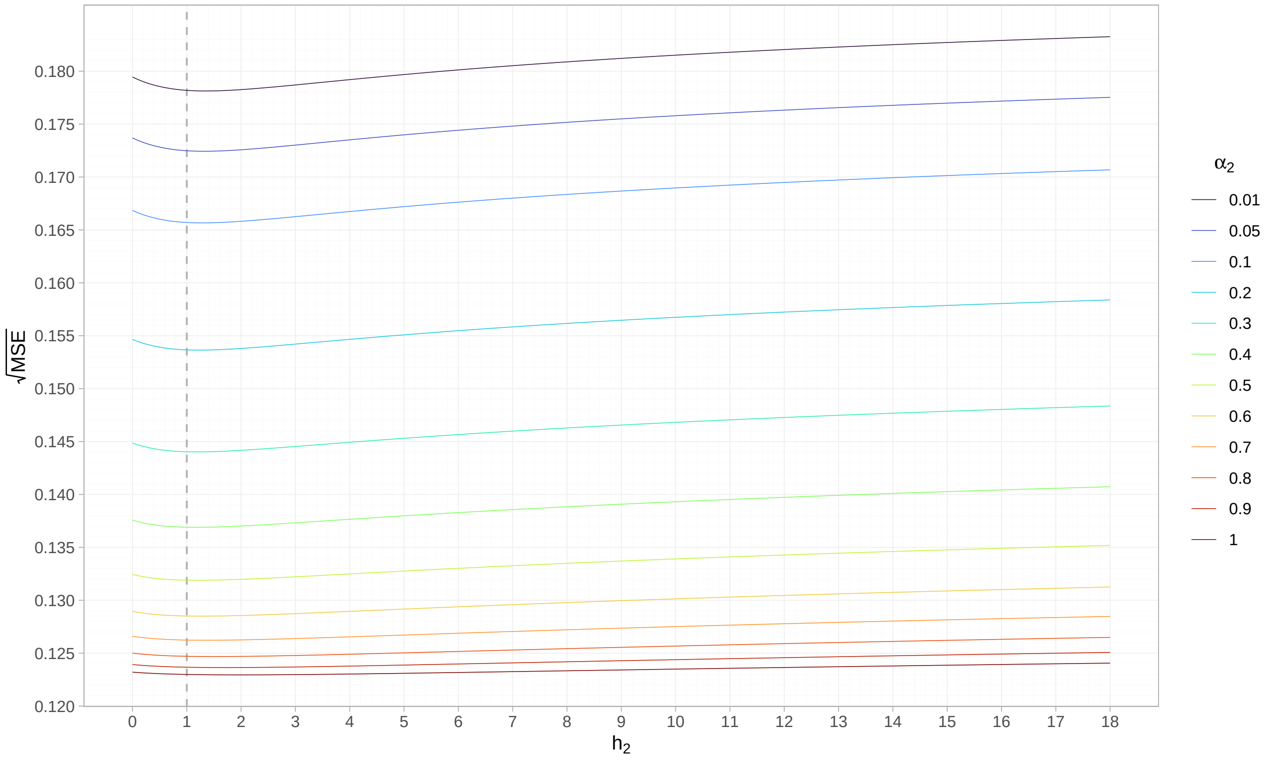


(b)

Figure B.2: Root mean squared error for the MM(Φ, α)PDE. $N = 30$. (a) Under no contamination. (b) Under contamination.

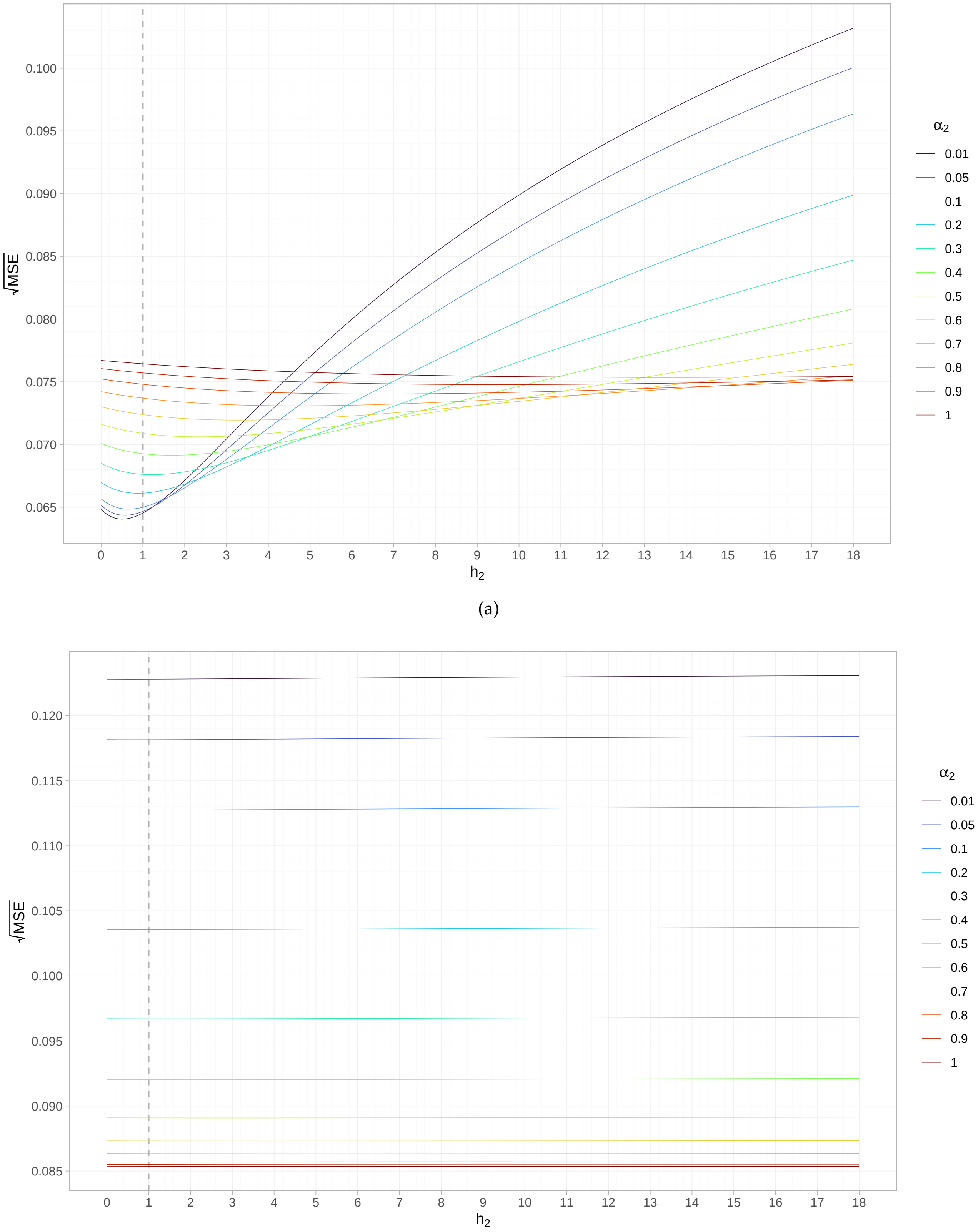


(a)



(b)

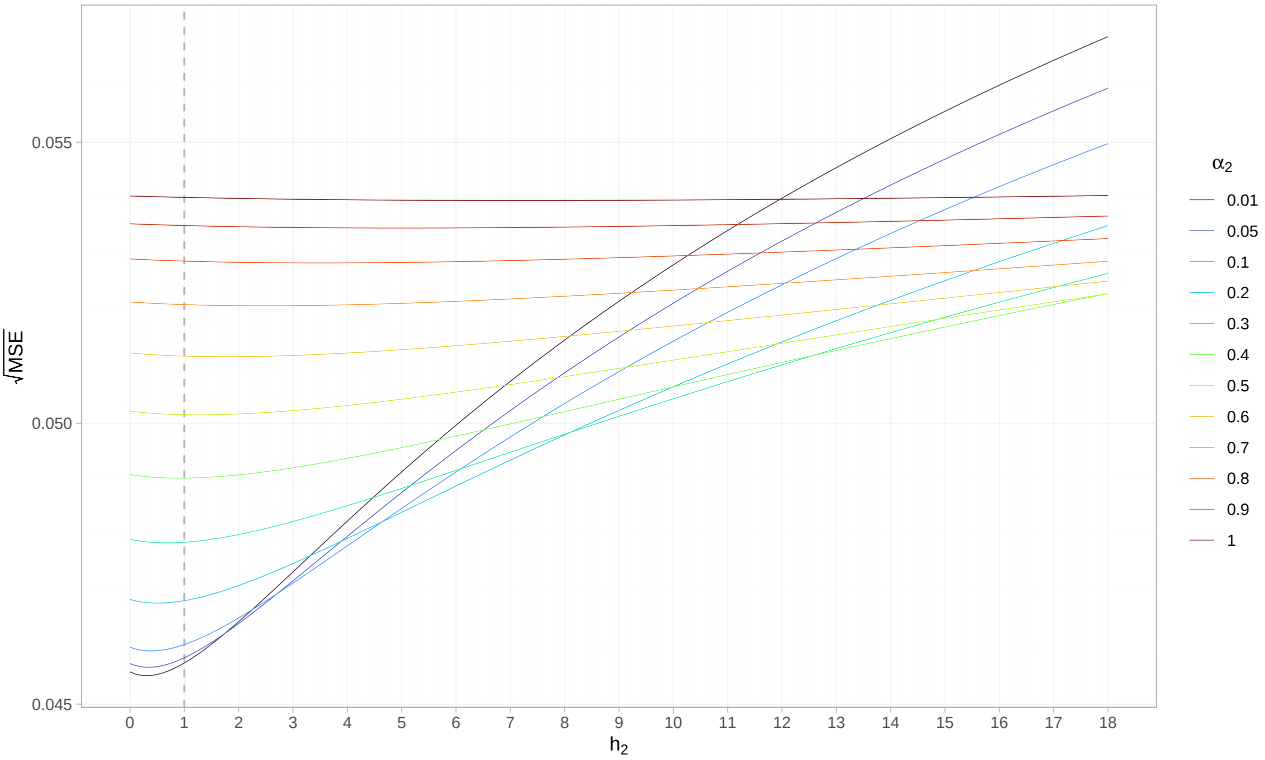
Figure B.3: Root mean squared error for the MM(Φ, α)PDE. $N = 60$.
(a) Under no contamination. (b) Under contamination.



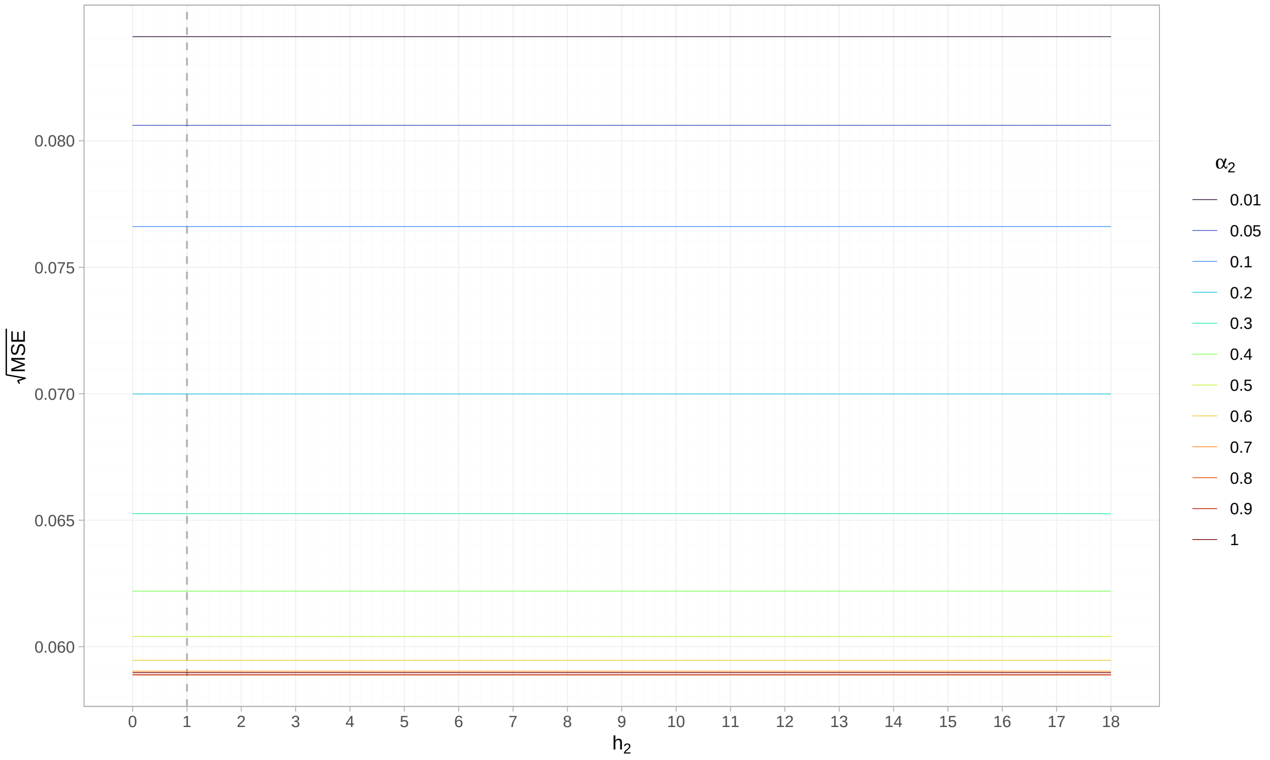
(a)

(b)

Figure B.4: Root mean squared error for the MM(Φ, α)PDE. $N = 120$.
(a) Under no contamination. (b) Under contamination.



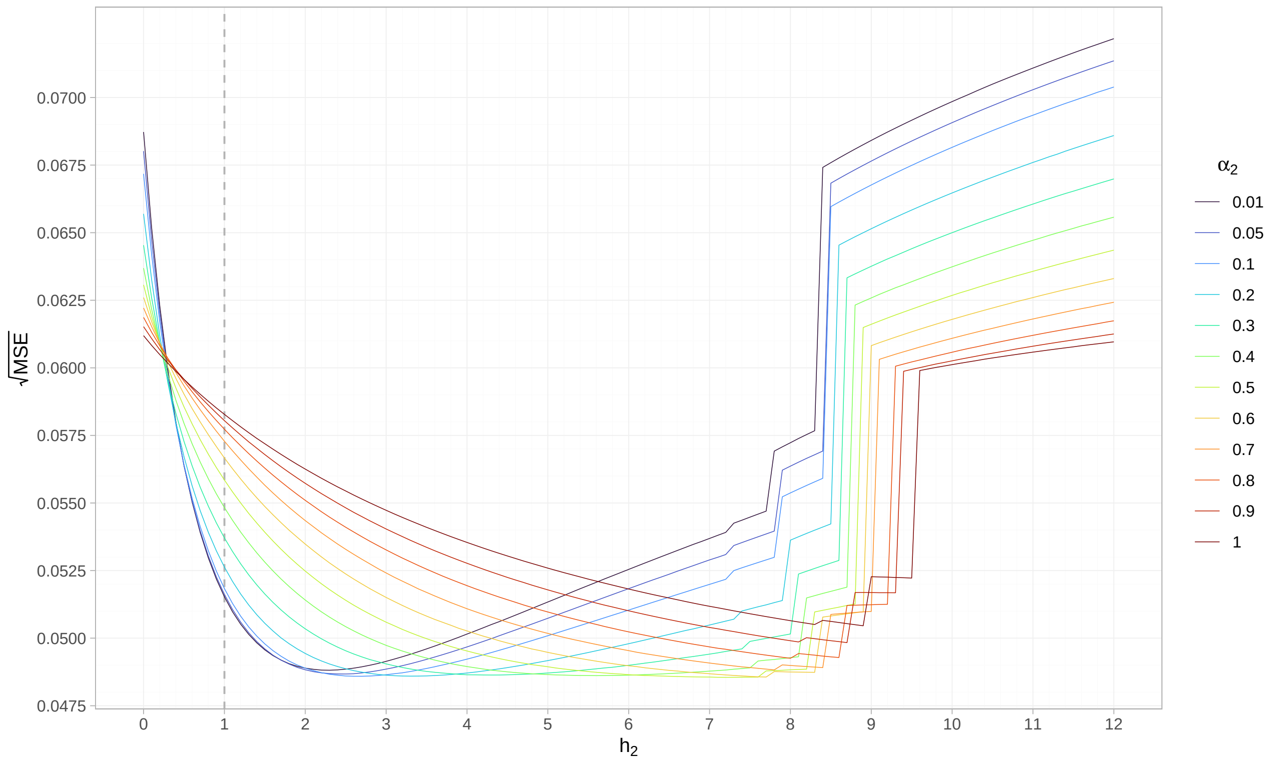
(a)



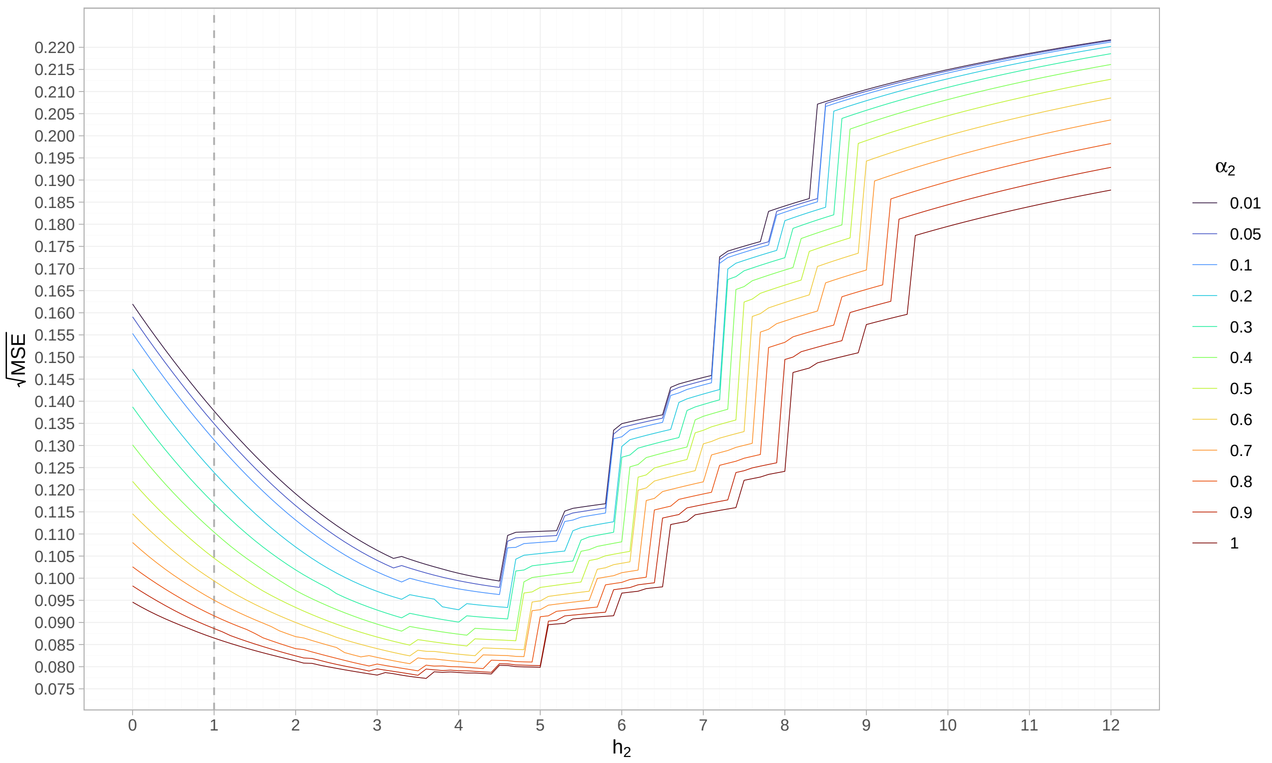
(b)

Figure B.5: Root mean squared error for the MM(Φ, α)PDE. $N = 240$.
(a) Under no contamination. (b) Under contamination.

B.2 Truncated Negative Binomial model

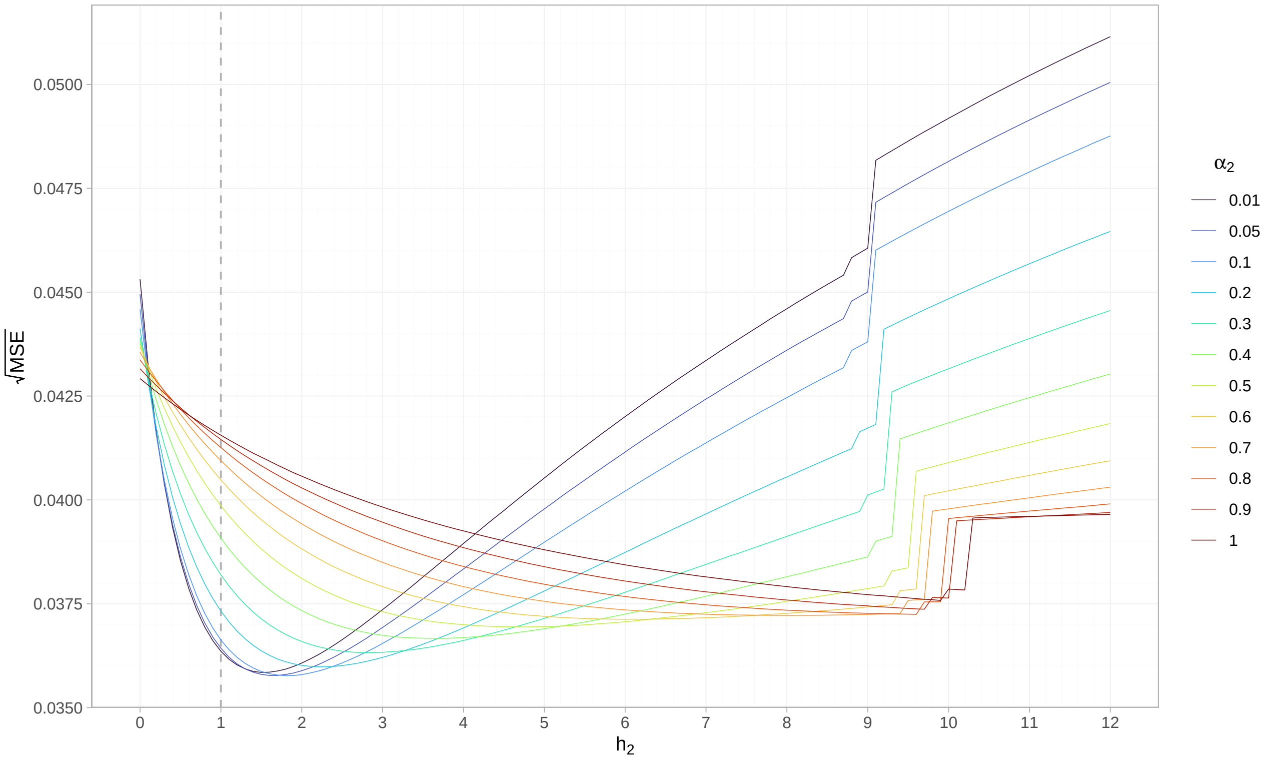


(a)

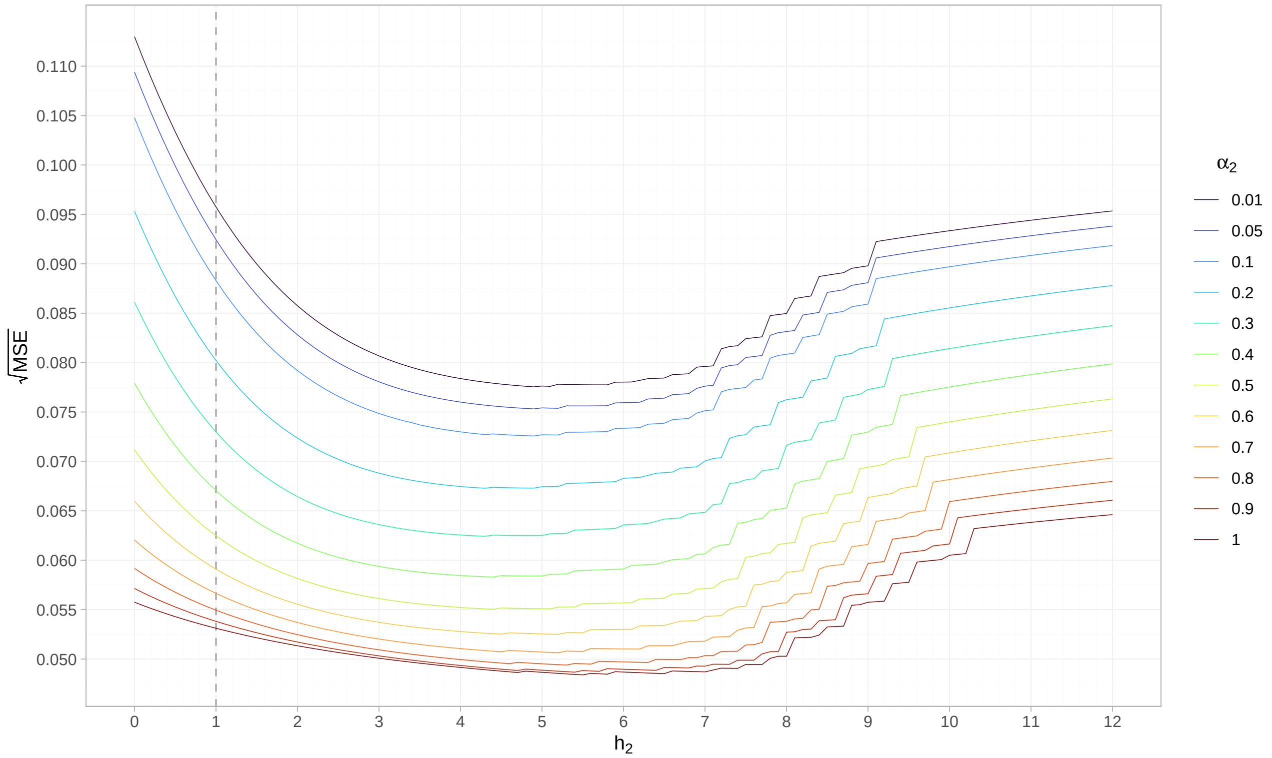


(b)

Figure B.6: Root mean squared error for the MM(Φ, α)PDE. $N = 15$.
(a) Under no contamination. (b) Under contamination.

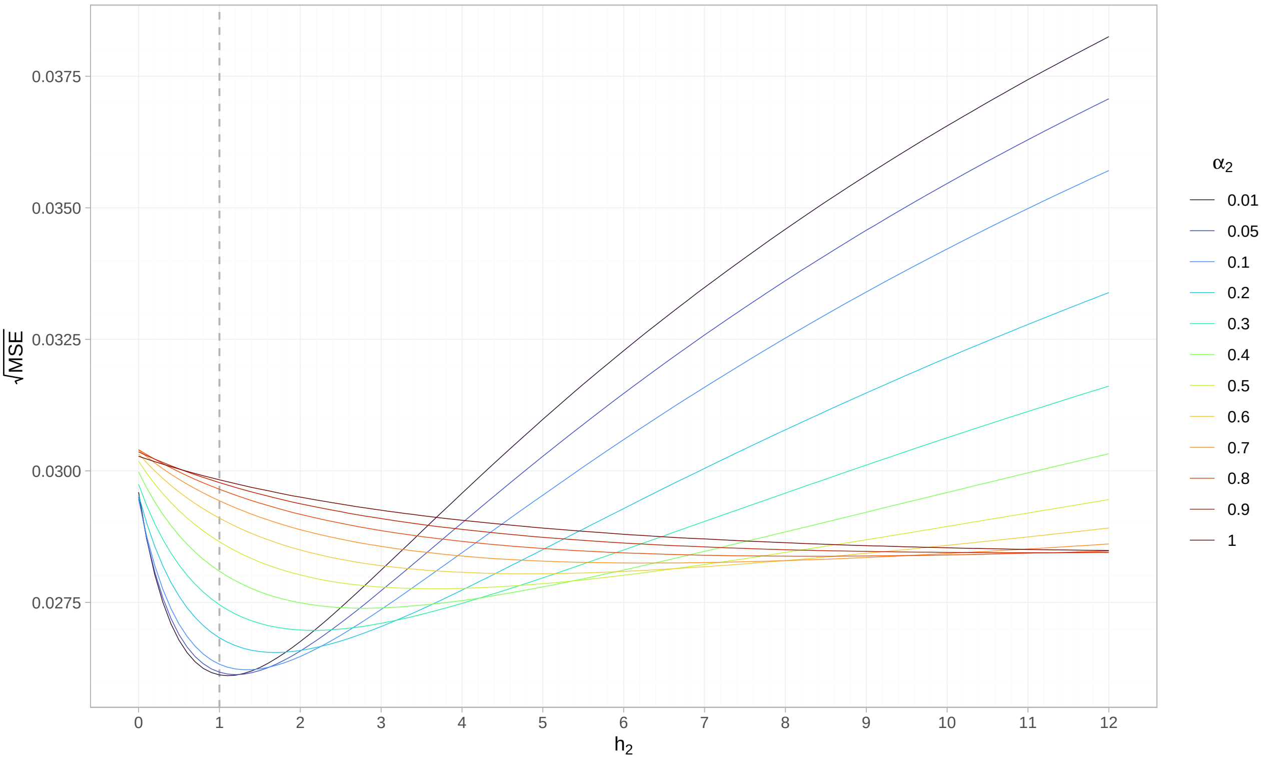


(a)

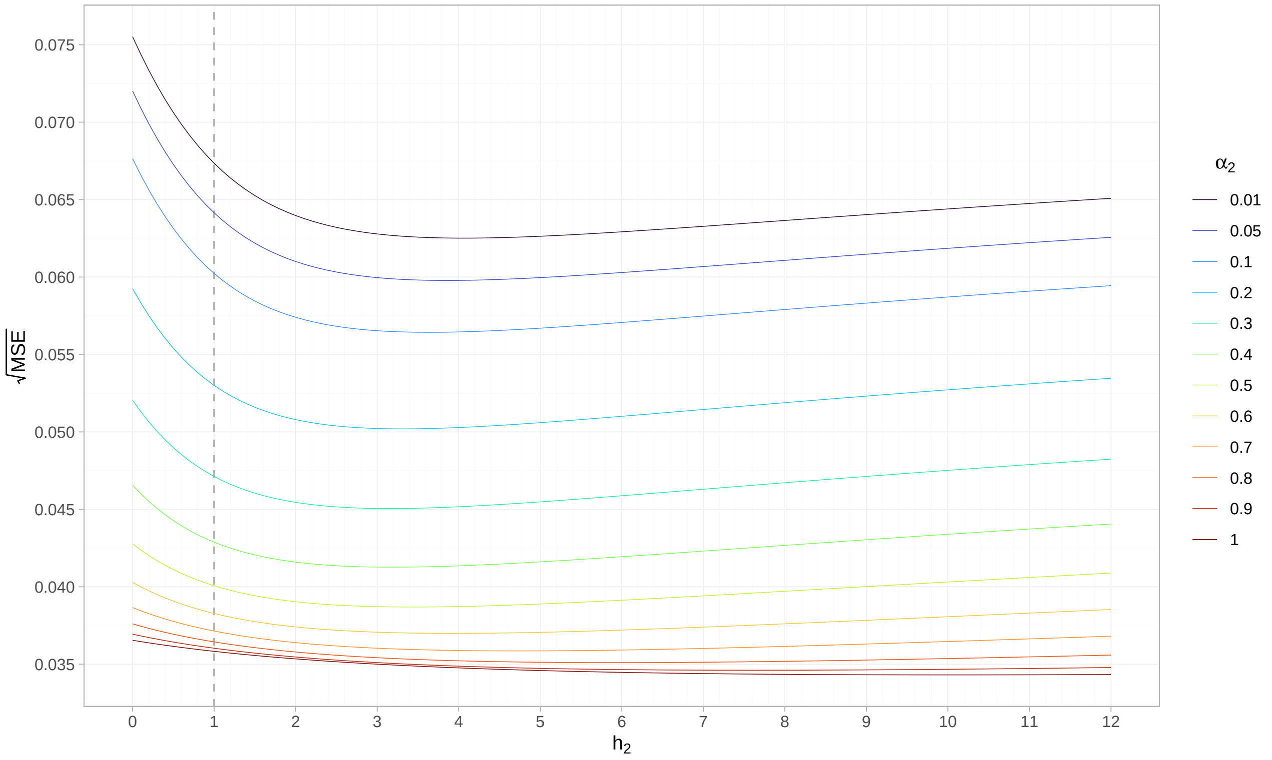


(b)

Figure B.7: Root mean squared error for the MM(Φ, α)PDE. $N = 30$.
(a) Under no contamination. (b) Under contamination.

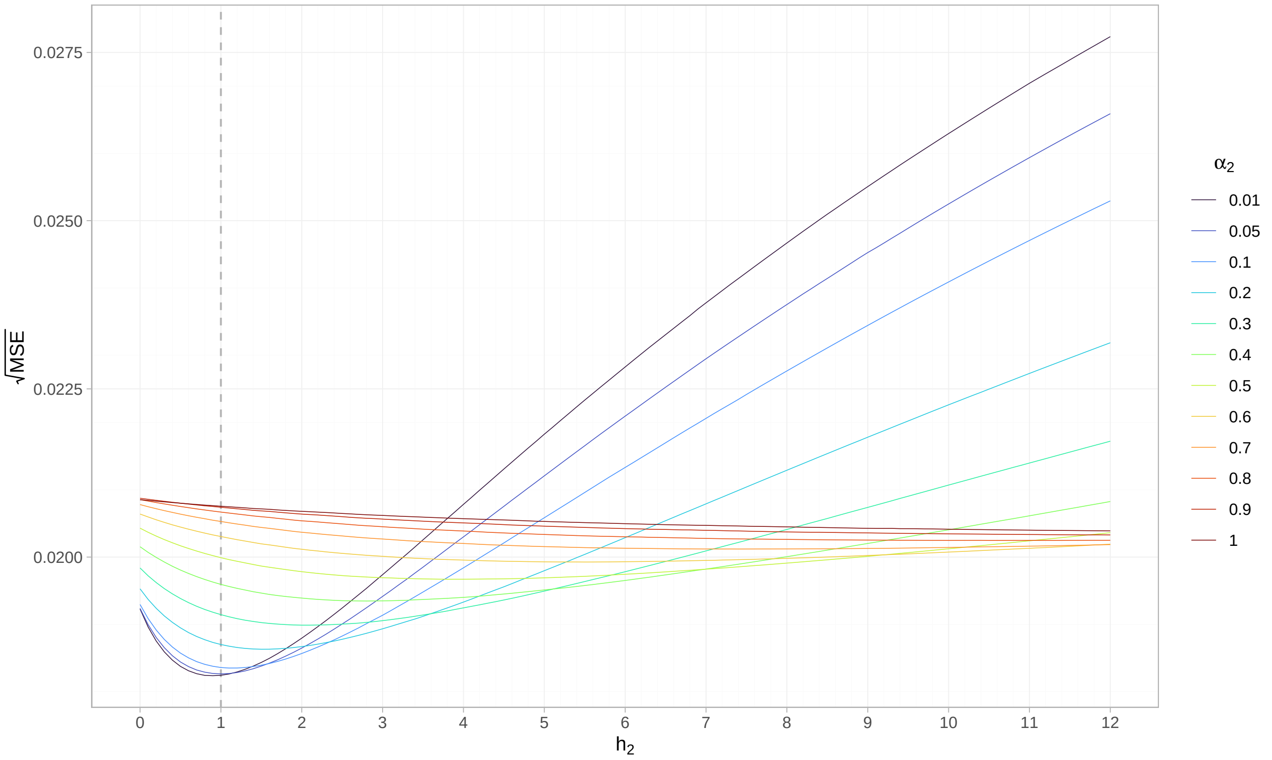


(a)

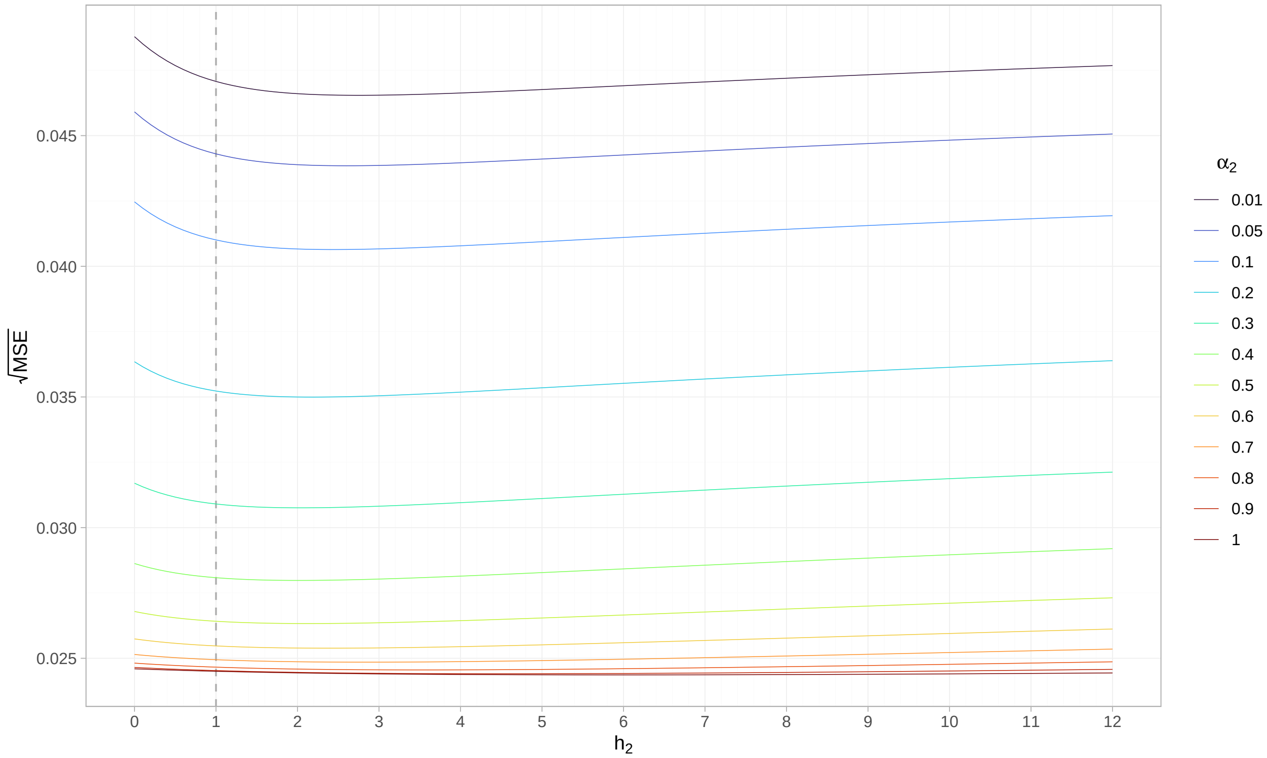


(b)

Figure B.8: Root mean squared error for the MM(Φ, α)PDE. $N = 60$.
(a) Under no contamination. (b) Under contamination.

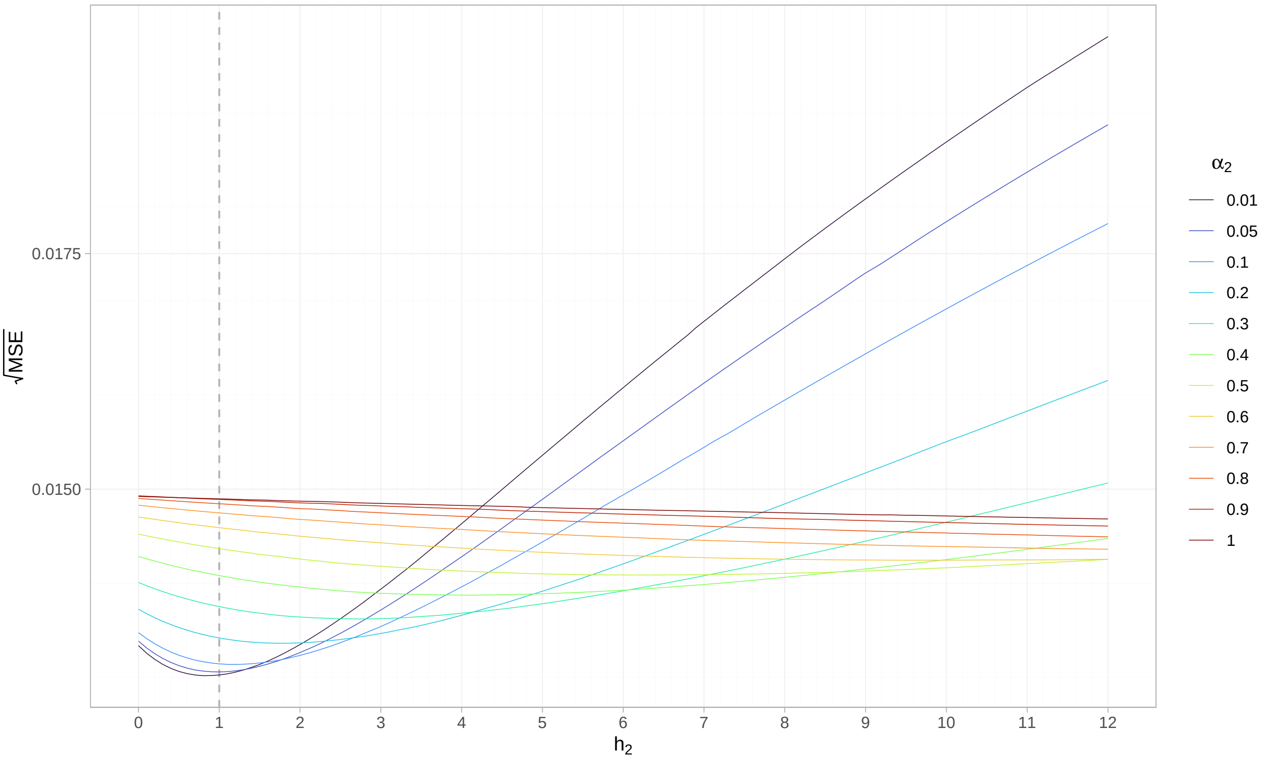


(a)

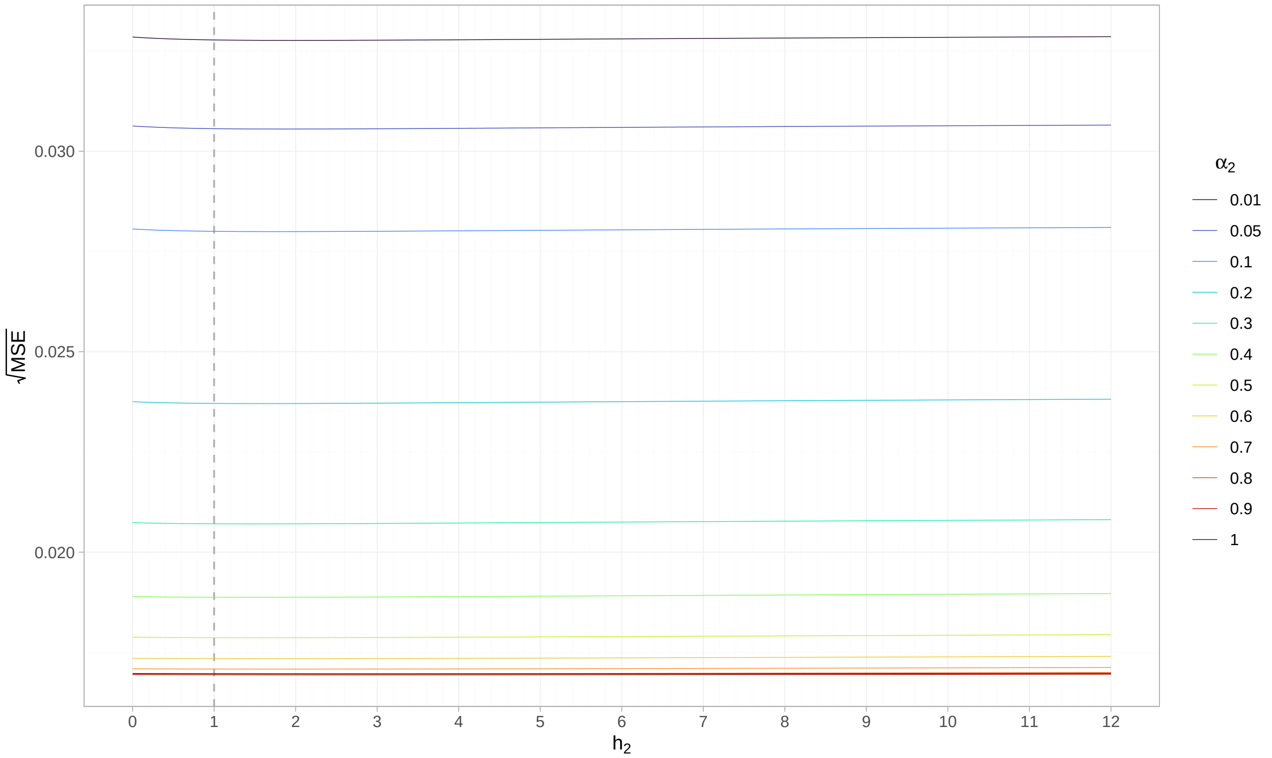


(b)

Figure B.9: Root mean squared error for the MM(Φ, α)PDE. $N = 120$.
(a) Under no contamination. (b) Under contamination.



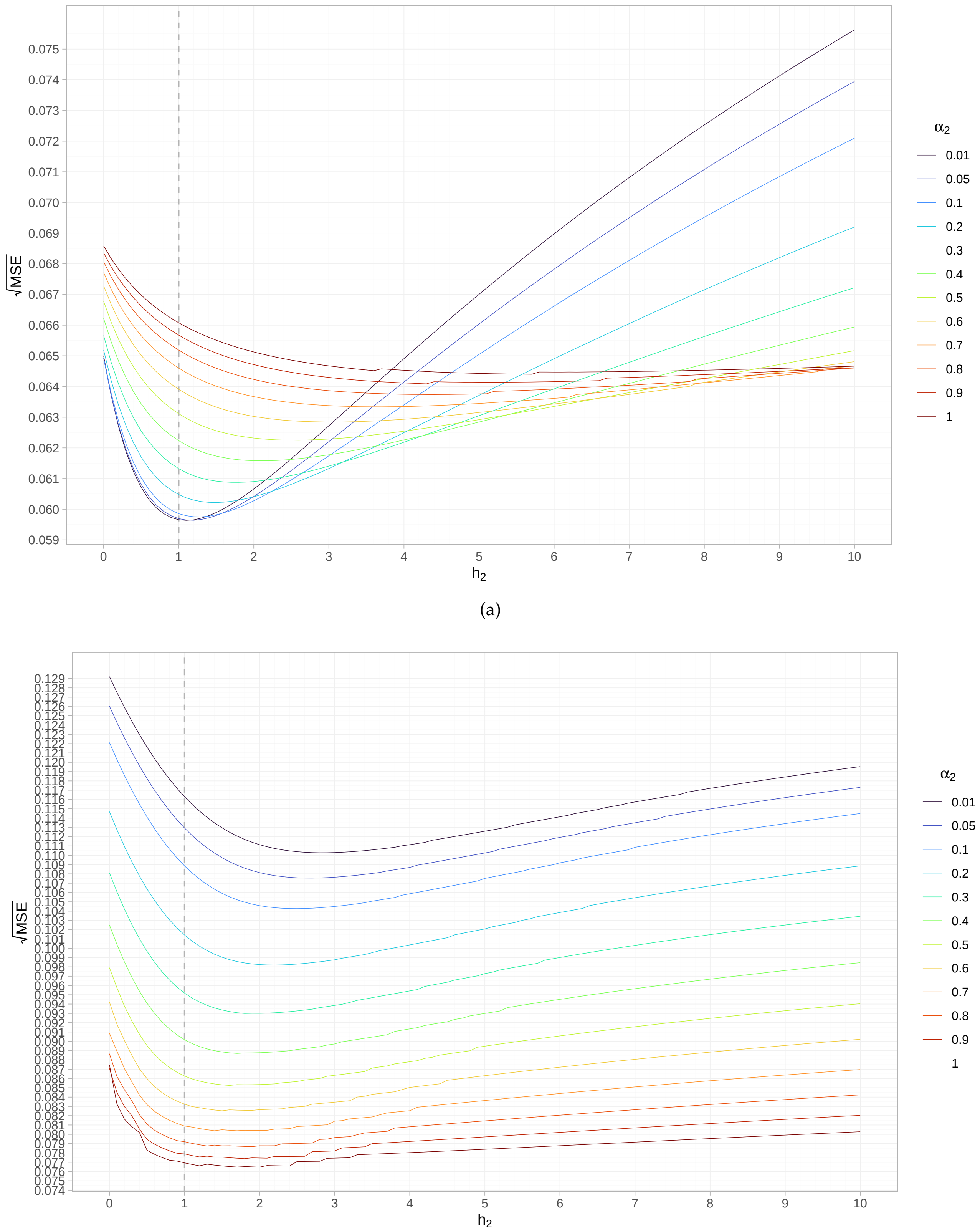
(a)



(b)

Figure B.10: Root mean squared error for the MM(Φ, α)PDE. $N = 240$.
(a) Under no contamination. (b) Under contamination.

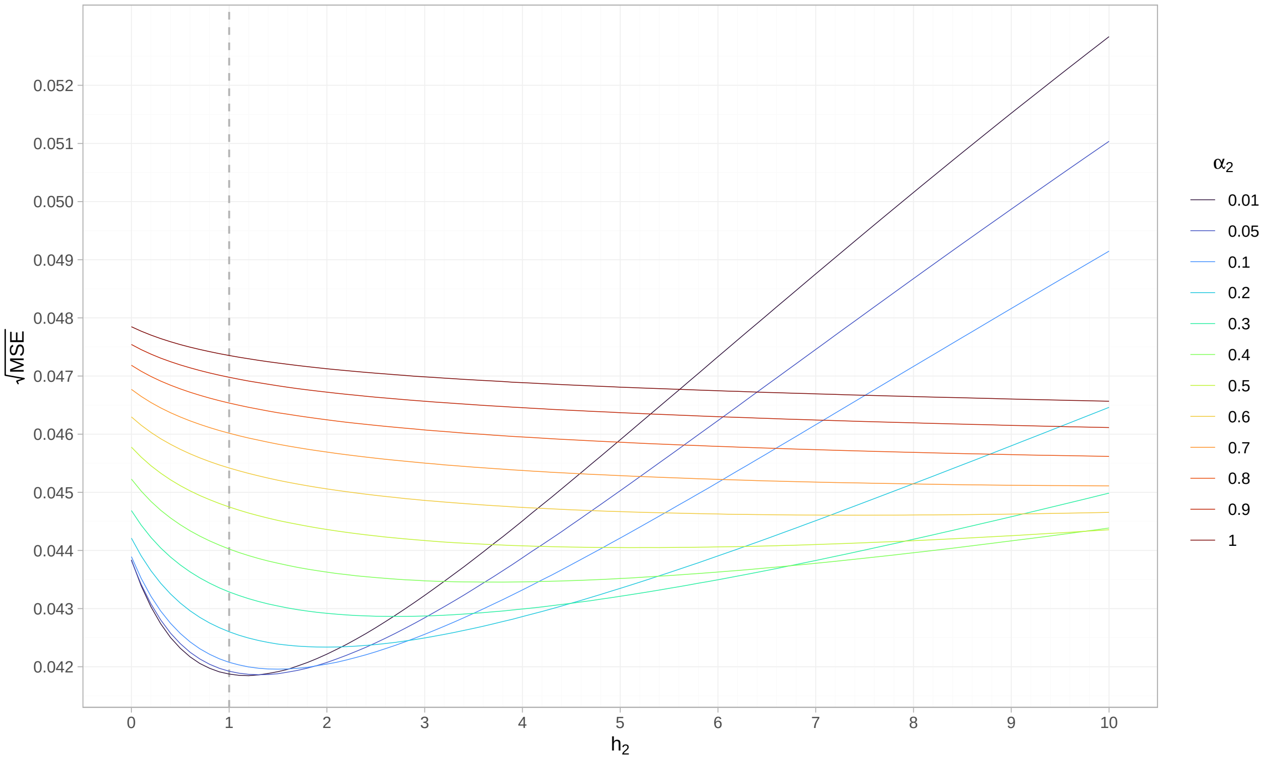
B.3 Truncated Binomial model



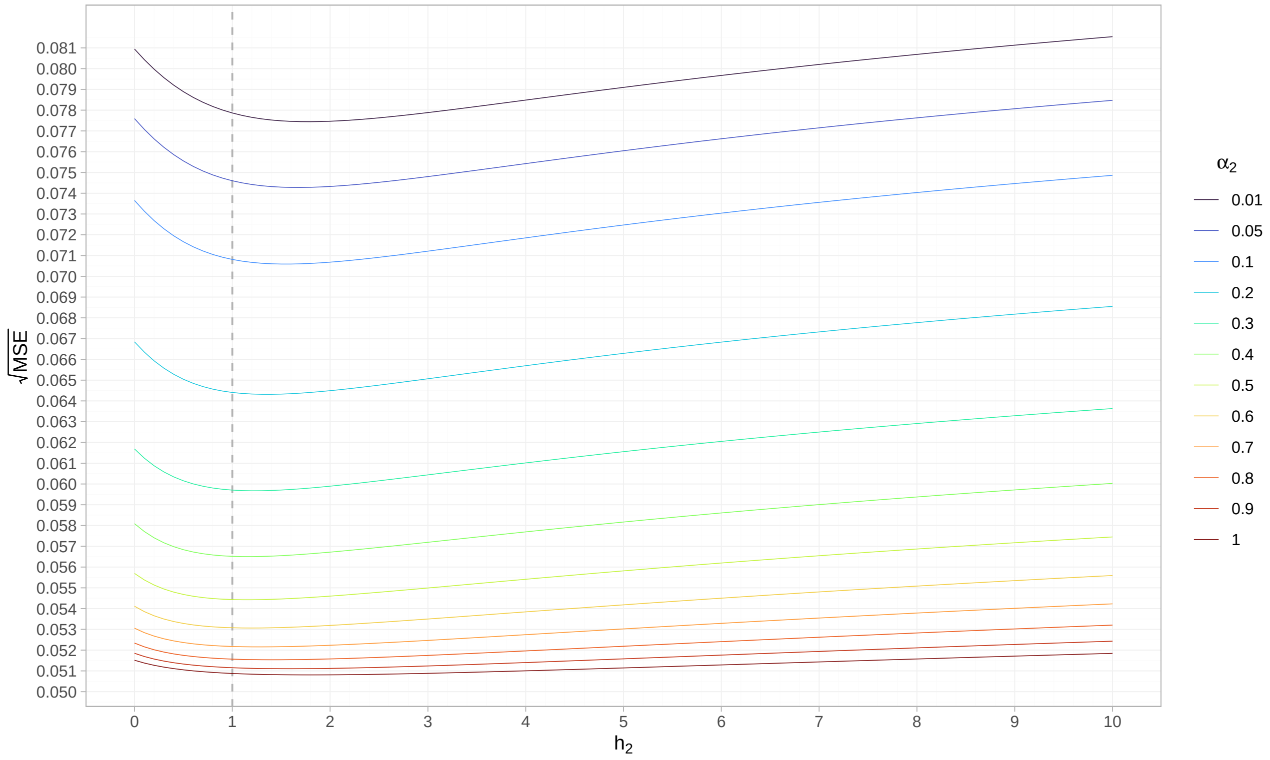
(a)

(b)

Figure B.11: Root mean squared error for the MM(Φ, α)PDE. $N = 15$.
(a) Under no contamination. (b) Under contamination.

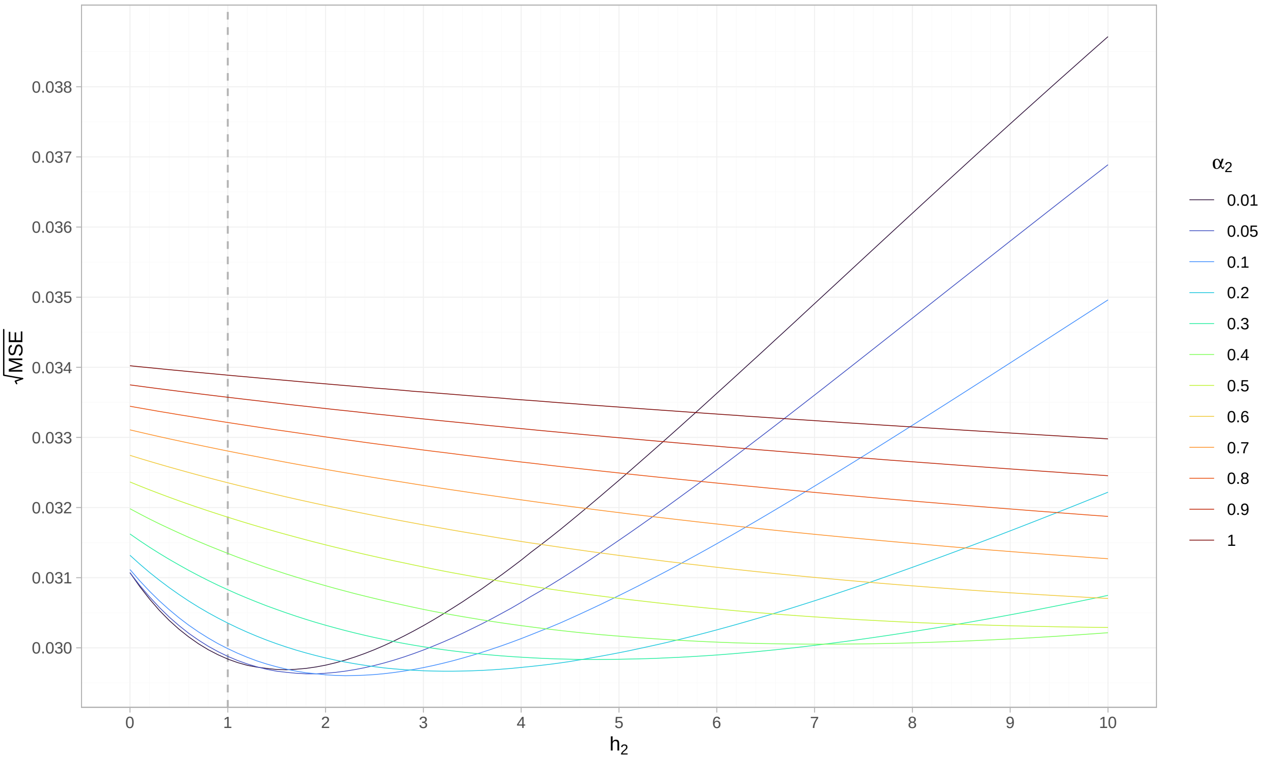


(a)

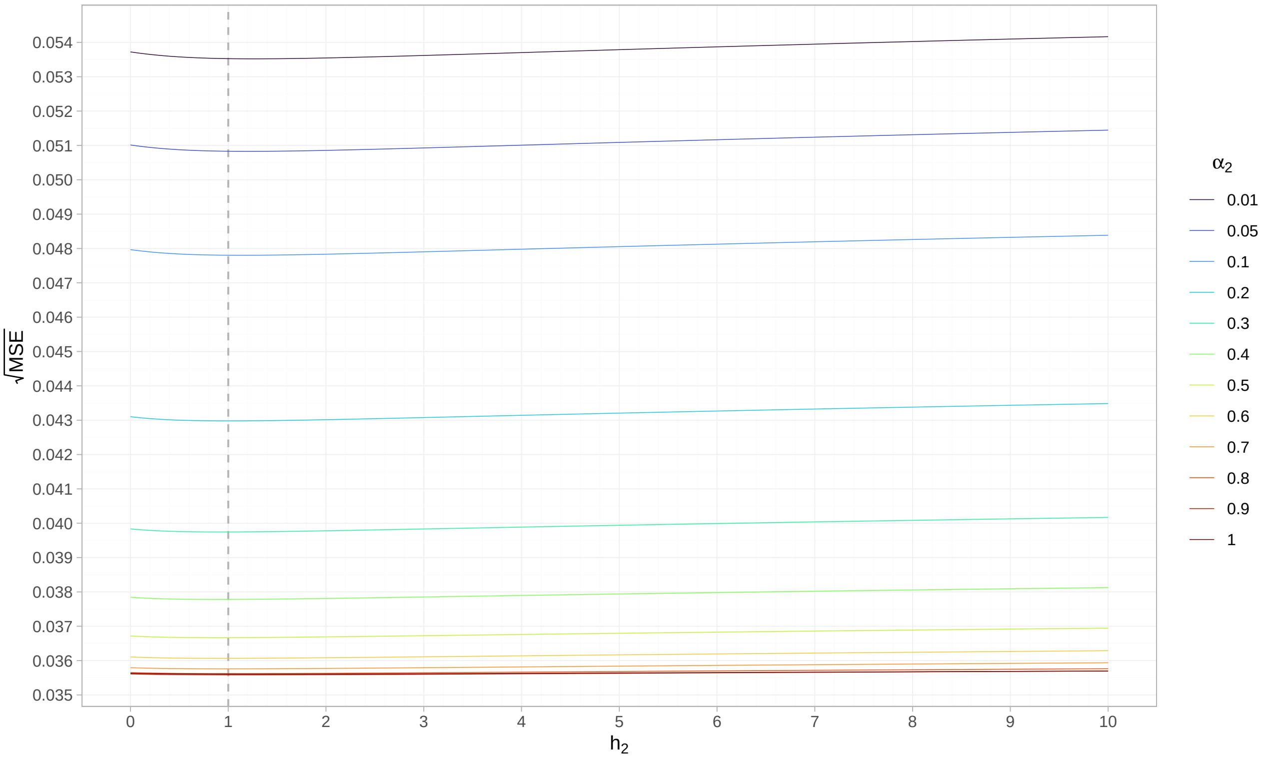


(b)

Figure B.12: Root mean squared error for the MM(Φ, α)PDE. $N = 30$.
(a) Under no contamination. (b) Under contamination.

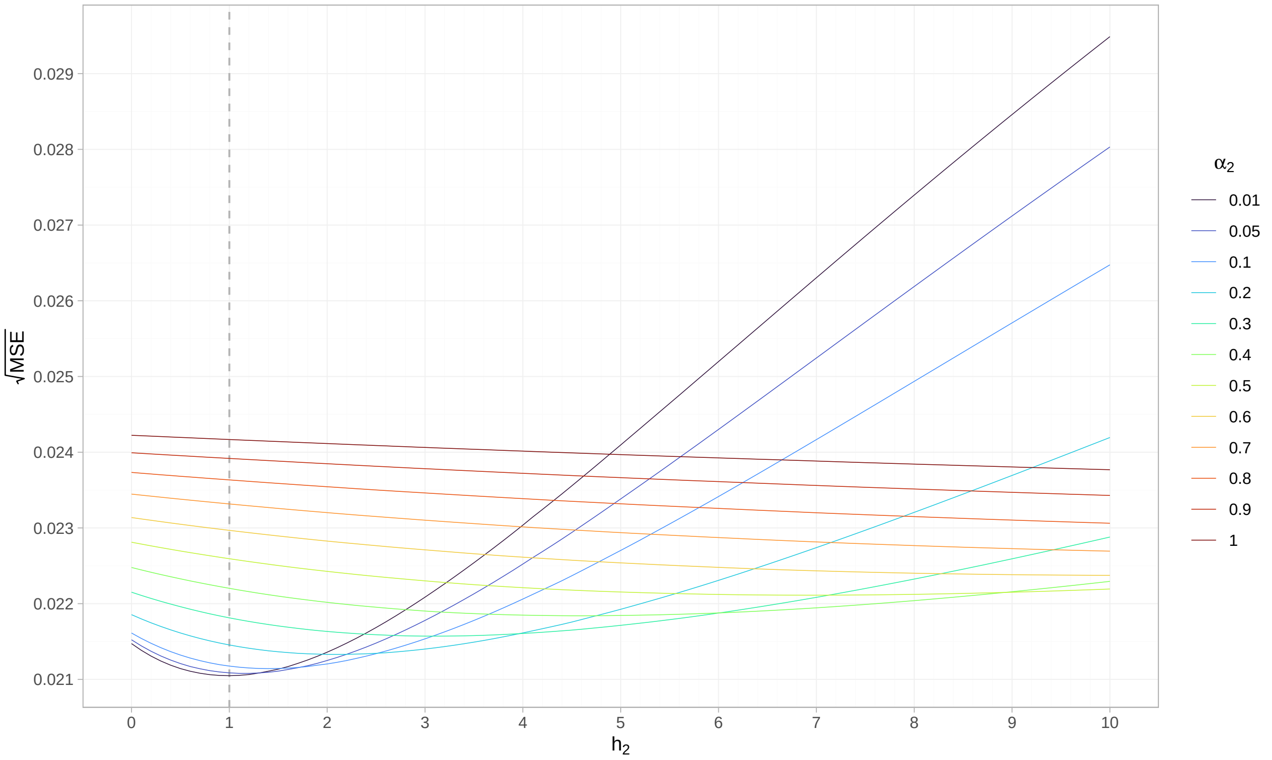


(a)

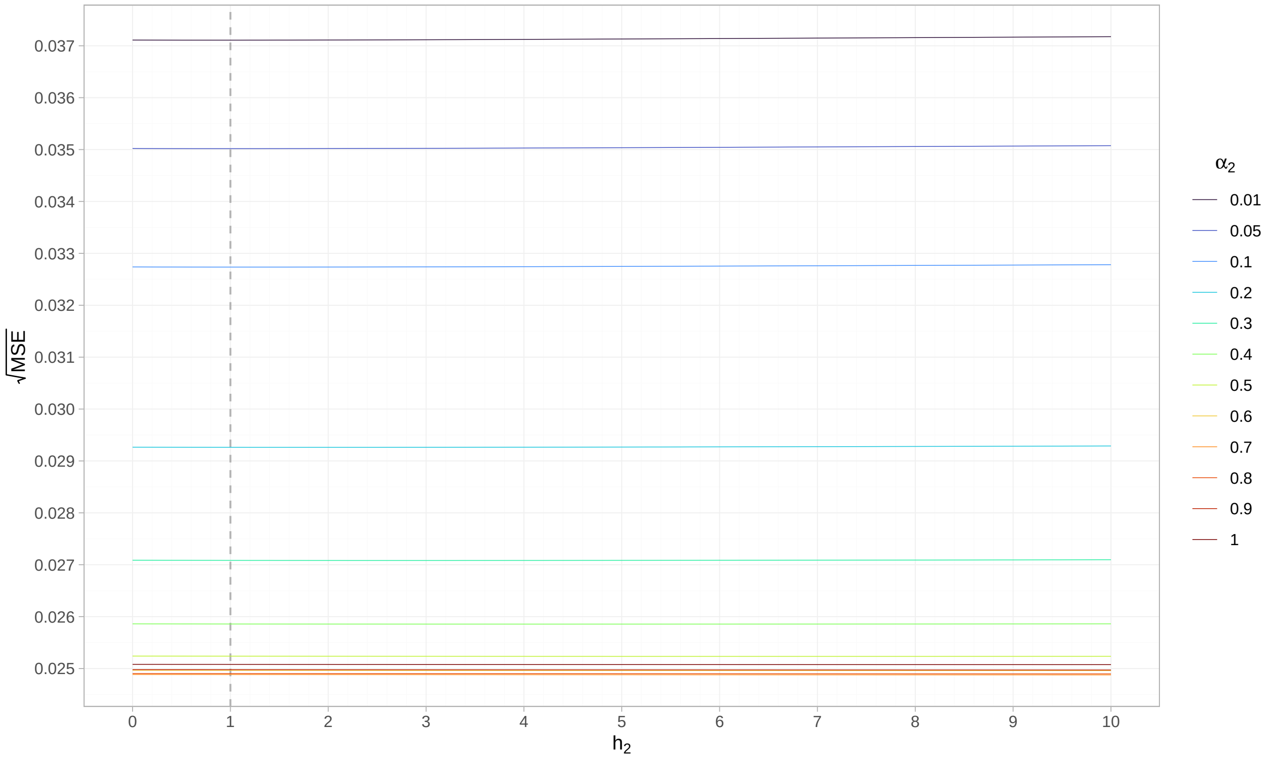


(b)

Figure B.13: Root mean squared error for the MM(Φ, α)PDE. $N = 60$.
(a) Under no contamination. (b) Under contamination.

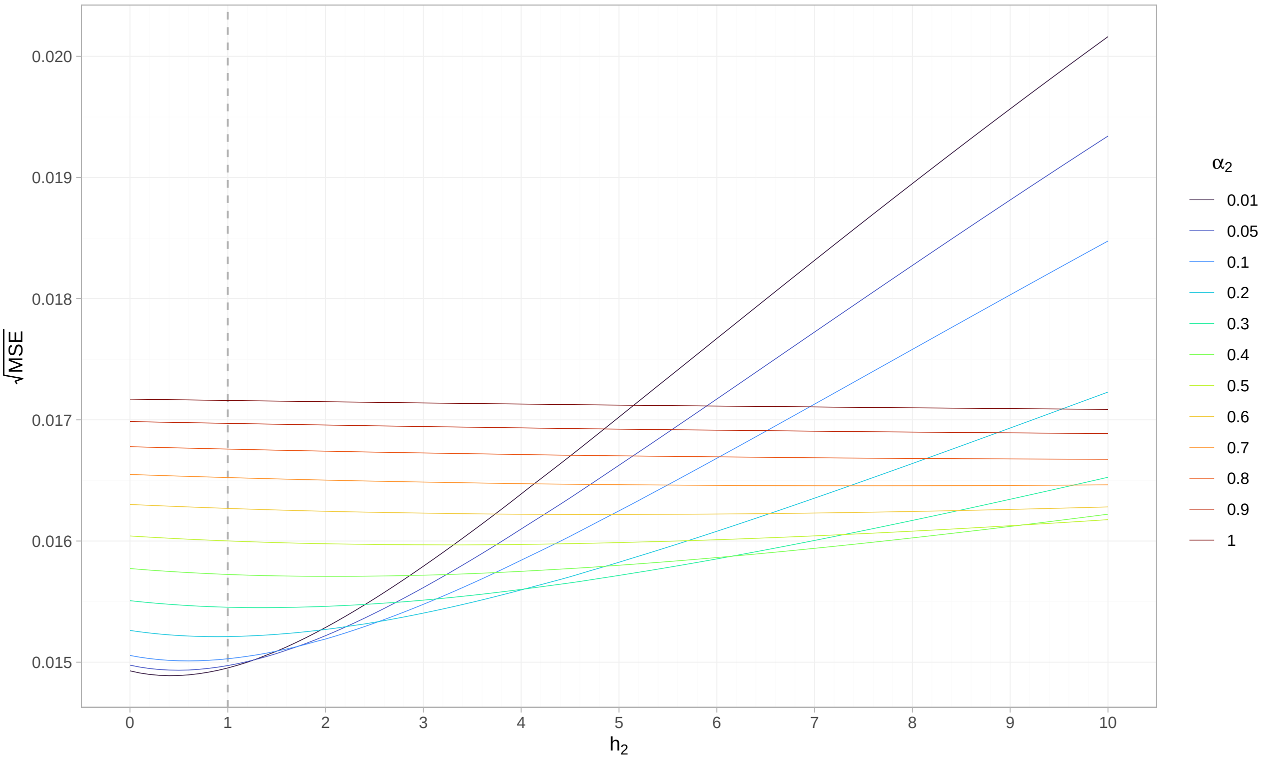


(a)

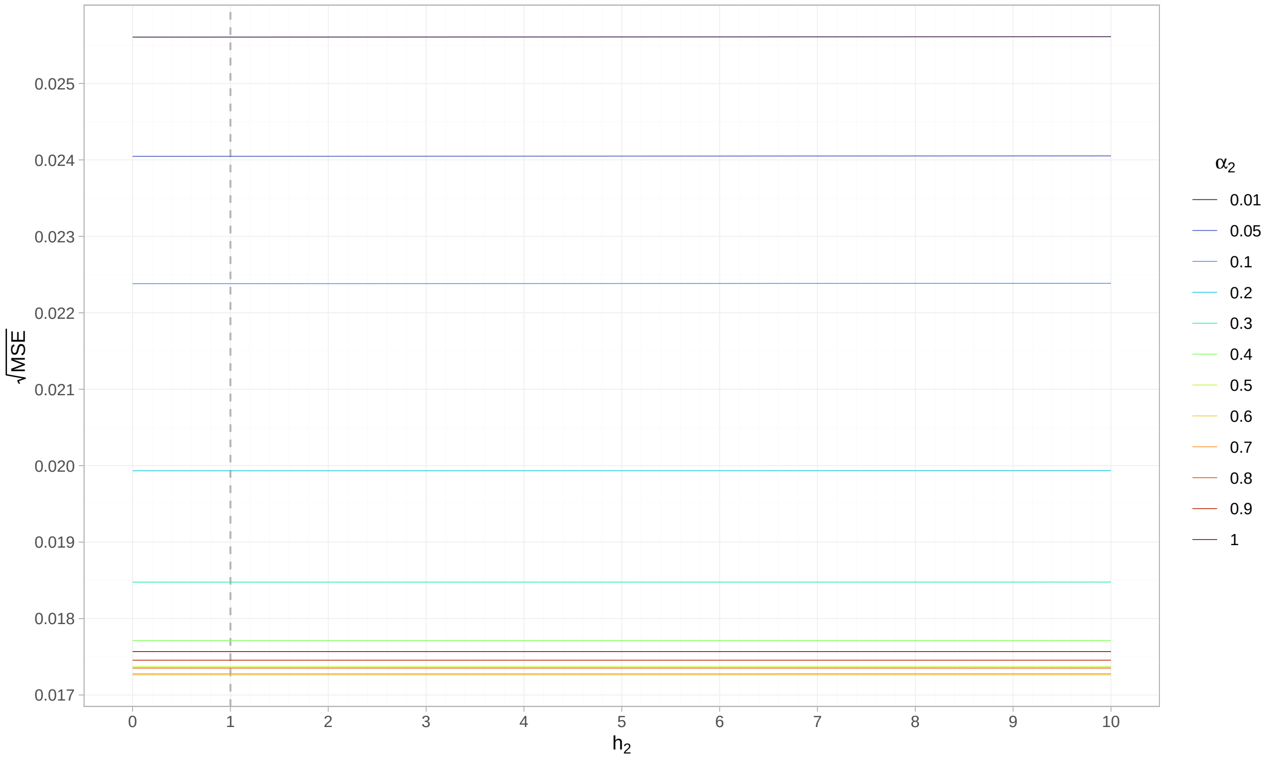


(b)

Figure B.14: Root mean squared error for the MM(Φ, α)PDE. $N = 120$.
(a) Under no contamination. (b) Under contamination.



(a)



(b)

Figure B.15: Root mean squared error for the MM(Φ,α)PDE. $N = 240$.
(a) Under no contamination. (b) Under contamination.

Appendix C

Simulation Platform

In this appendix, the syntax regarding the simulations studies undertaken in this thesis is presented. For each simulation study, the R software (R Core Team, 2016) is used. All packages that have been used are presented in the following lines. Thus, for the manipulation of the data, the packages **data.table** (Dowle & Srinivasan, 2022), **dplyr** (Wickham et al., 2022), and **tibble** (Müller & Wickham, 2022) are used, while for the visualization of the results the packages **ggplot2** (Wickham, 2016), **ggpubr** (Kassambara, 2020), **gridExtra** (Auguie, 2017), **lattice** (Sarkar, 2008), **ggsci** (Xiao, 2018), **hrbrthemes** (Rudis, 2020), **directlabels** (Hocking, 2021), **ggrepel** (Slowikowski, 2021), **scales** (Wickham, 2018), **fmsb** (Nakazawa, 2022), **latex2exp** (Meschiari, 2022) and **viridis** (Garnier et al., 2021) are used. In order to save the results into an *.xlsx* file, the packages **rJava** (Urbanek, 2021) and **xlsx** (Dragulescu & Arendt, 2020) are used, while in order to activate parallel computing the package **future.apply** (Bengtsson, 2021) is used. In order to generate random sample data from the inverse Gaussian distribution and the generalized Pareto distribution the packages **statmod** (Giner & Smyth, 2016) and **eva** (Bader & Yan, 2020) are used respectively. Finally, for the optimization the **optimise** function from the base package **stats** is implemented while for the constrained optimization the **auglag** function from the **nloptr** package (Johnson, 2014) is implemented.

C.1 Asymptotic results verification

C.1.1 ECDF of $T_{\varphi}\left(\hat{\boldsymbol{\theta}}^{\alpha}_{\Phi}\right)$ and $T^{\alpha_1}_{\Phi_1}\left(\hat{\boldsymbol{\theta}}^{\alpha_2}_{\Phi_2}\right)$

```r
# Used packages
library(data.table)
library(future.apply)
plan(multisession)  #

n <- 10000 # Sample size.
N <- 10000 # Number of simulated samples.

u_vect <- c(0.5, 0.6, 0.7, 0.8, 0.9, 0.999) # Value of the parameter.

# NULL MODEL DEFINITION
hypvector <- function(v){
  z <- c(v / 2
         , v * (1 - v / 2) - 1 / 8
         , (1 - v) / 2
         , (1 - v) ^ 2 / 2 + 1 / 8
  )
  return(z)
}

# Definition of the Divergence Measures

```

```
# Power Divergence Measure for the Estimation of the unknown parameter
CR_Divergence_Function <- function(u, lamda, probvector) {
  if (lamda == 0) {
    vect <- probvector * log(probvector)
    vect[vect == 'NaN'] <- 0
    z <- sum(vect) - sum(probvector * log(hypvector(v=u)))
  } else if (lamda == -1) {
    z <- sum(hypvector(v=u)
             * (log(hypvector(v=u)) - log(probvector)))
  } else {
    z <- (1 / (lamda * (lamda + 1))) * (sum((probvector ^ (lamda + 1)
    ) / (hypvector(v=u) ^ lamda)) - 1)
  }
  return(z)
}

# BHHJ Divergence Measure for the Estimation of the unknown parameter
BHHJ_Divergence_Function <- function(u, alpha, probvector) {
  z <- sum(hypvector(v = u) ^ (1 + alpha)) -
    (1 + 1 / alpha) * sum(probvector * hypvector(v = u) ^ alpha) +
    (1 / alpha) * sum(probvector ^ (1 + alpha))
  return(z)
}

# Definition of the Test Statistics

# Cressie-Read Test Statistic
CR_Test_Statistic <- function(u1, lamda1, probvector1, n) {
  if (lamda1 == 0) {
    z <-
      2 * n * CR_Divergence_Function(u = u1,
                                     lamda = lamda1,
                                     probvector = probvector1)
  } else if (lamda1 == -1) {
    z <-
      2 * n * CR_Divergence_Function(u = u1,
                                     lamda = lamda1,
                                     probvector = probvector1)
  } else {
    z <-
      2 * n * CR_Divergence_Function(u = u1,
                                     lamda = lamda1,
                                     probvector = probvector1)
  }
  return(z)
}

# BHHJ Test Statistic
BHHJ_Test_Statistic <- function(u1, alpha1, probvector1, n) {
  z <-
    ((2 * n) / (1 + alpha1)) * BHHJ_Divergence_Function(u = u1,
                                                        alpha = alpha1,
                                                        probvector = probvector1)
  return(z)
}

# Implementation for the CR test statistics with the BHHJ estimator.
# ----------------------------------------------------------------
a2_vect <- c(0.01, 0.05, seq(from = 0.1, to = 1.0, by = 0.1), 3, 5)
a2 <- rep(a2_vect, each = N)

for (j in u_vect){
  u <- j
  print(u)
  # Sample generation
  set.seed(seed = 1)
  relfreq <- setNames(as.data.table(t(rmultinom(n = N
                                                , size = n
                                                , prob = hypvector(v=u)
  )
  )/n
  )
```

```
  , c("phat_1","phat_2","phat_3","phat_4")) # Each line represents a vector of relative frequencies.


  # Data table generation
  Data <- data.table(n, N, u, a2, relfreq) # relfreq table is repeated as many times as the length of a2

  # Estimation of the unknown parameter
  Data[, MDE := future_apply(Data, 1, function(i){
    optimise(
      BHHJ_Divergence_Function
      ,interval = c(0.134,1)
      ,alpha = i[["a2"]]
      ,probvector = c(i[["phat_1"]]
                      ,i[["phat_2"]]
                      ,i[["phat_3"]]
                      ,i[["phat_4"]]))$minimum
  })]

  # Test statistic's value calculation
  Data[, ChiSqTestStat := future_apply(Data, 1, function(i){
    CR_Test_Statistic(u1 = i[["MDE"]]
                        , lamda1 = 1
                        , probvector1 = c(i[["phat_1"]]
                                          ,i[["phat_2"]]
                                          ,i[["phat_3"]]
                                          ,i[["phat_4"]])
                        , n = i[["n"]])
  }
  )
  ]

  # Save results in a .csv file
  fwrite(Data, paste("CR_MDE_and_TS_u"
                     , u
                     , "N", N
                     , "n", n
                     , ".csv"
                     , sep = "_"))
}

# Load results for each parameter value u
dt_1 <- fread("CR_MDE_and_TS_u_0.5_N_10000_n_10000_.csv")
dt_2 <- fread("CR_MDE_and_TS_u_0.6_N_10000_n_10000_.csv")
dt_3 <- fread("CR_MDE_and_TS_u_0.7_N_10000_n_10000_.csv")
dt_4 <- fread("CR_MDE_and_TS_u_0.8_N_10000_n_10000_.csv")
dt_5 <- fread("CR_MDE_and_TS_u_0.9_N_10000_n_10000_.csv")
dt_6 <- fread("CR_MDE_and_TS_u_0.999_N_10000_n_10000_.csv")

# Combine results for each parameter value u
Total_dt <- rbind(dt_1
                  , dt_2
                  , dt_3
                  , dt_4
                  , dt_5
                  , dt_6
)

# Save results for every parameter value u in one .csv file
fwrite(Total_dt
       ,paste("CR_MDE_and_TS_Total"
              , "N", N
              , "n", n
              , ".csv"
              , sep = "_")
       )

# Implementation for the BHHJ test statistics with the BHHJ estimator.
# --------------------------------------------------------------------
a1_vect <- c(0.01, 0.05, seq(from = 0.1, to = 1.0, by = 0.1), 3, 5)
a2_vect <- c(0.01, 0.05, seq(from = 0.1, to = 1.0, by = 0.1), 3, 5)
```

```

# For the combination of a1_vect with th a2_vect for each sample
a2 <- rep(a2_vect, each = length(a1_vect) * N)
a1 <- rep(a1_vect, each = N, times = length(a2_vect))

for (j in u_vect){
  u <- j
  print(u)
  # Data Sample Generation (Frequencies & Relative Frequencies)
  set.seed(seed = 1)
  # The sample
  relfreq <- setNames(as.data.table(t(rmultinom(n = N
                                                , size = n
                                                , prob = hypvector(v=u)
  )
  )/n
  )
  , c("phat_1","phat_2","phat_3","phat_4")) # Each line represents a vector of relative frequencies.

  # Data Table Generation
  Data <- data.table(n, N, u, a2, a1, relfreq)

  Data[, MDE := future_apply(Data, 1, function(i){
    optimise(
      BHHJ_Divergence_Function
      ,interval = c(0.134,1)
      ,alpha = i[["a2"]]
      ,probvector = c(i[["phat_1"]]
                      ,i[["phat_2"]]
                      ,i[["phat_3"]]
                      ,i[["phat_4"]]))$minimum
  })]

  Data[, BHHJTestStat := future_apply(Data, 1, function(i){
    BHHJ_Test_Statistic(u1 = i[["MDE"]]
                        , alpha1 = i[["a1"]]
                        , probvector1 = c(i[["phat_1"]]
                                          ,i[["phat_2"]]
                                          ,i[["phat_3"]]
                                          ,i[["phat_4"]])
                        , n = i[["n"]])
  }
  )
  ]

  Data[, c_param := future_apply(Data, 1, function(i){
    0.5 * ((max(hypvector(v = i[["u"]]))) ^ i[["a1"]] +
             (min(hypvector(v = i[["u"]]))) ^ i[["a1"]])
  }
  )
  ]

  fwrite(Data, paste("BHHJ_MDE_and_TS_u"
                     , u
                     , "N", N
                     , "n", n
                     , ".csv"
                     , sep = "_"))
}

# Load results for each parameter value u
dt_1 <- fread("BHHJ_MDE_and_TS_u_0.5_N_10000_n_10000_.csv")
dt_2 <- fread("BHHJ_MDE_and_TS_u_0.6_N_10000_n_10000_.csv")
dt_3 <- fread("BHHJ_MDE_and_TS_u_0.7_N_10000_n_10000_.csv")
dt_4 <- fread("BHHJ_MDE_and_TS_u_0.8_N_10000_n_10000_.csv")
dt_5 <- fread("BHHJ_MDE_and_TS_u_0.9_N_10000_n_10000_.csv")
dt_6 <- fread("BHHJ_MDE_and_TS_u_0.999_N_10000_n_10000_.csv")

# Combine results for each parameter value u
Total_dt <- rbind(dt_1
                  , dt_2
```

```
                  , dt_3
                  , dt_4
                  , dt_5
                  , dt_6
)

# Save results for every parameter value u in one .csv file
fwrite(Total_dt
       ,paste("BHHJ_MDE_and_TS_Total"
              , "N", N
              , "n", n
              , ".csv"
              , sep = "_")
       )
```

For the production of the associated plots, the following code was implemented. It is given only for the case regarding $T_{\Phi_1}^{\alpha_1}\left(\hat{\boldsymbol{\theta}}_{\Phi_2}^{\alpha_2}\right)$. For the case regarding $T_{\varphi}\left(\hat{\boldsymbol{\theta}}_{\Phi}^{\alpha}\right)$ the code can accordingly be modified.

```
library(data.table)
library(dplyr)
library(ggplot2)
library(latex2exp)
library(gridExtra)
library(ggpubr)

hypvector <- function(v){
  z <- c(v / 2
           , v * (1 - v / 2) - 1 / 8
           , (1 - v) / 2
           , (1 - v) ^ 2 / 2 + 1 / 8
  )
  return(z)
}

Data_Table <- fread("BHHJ_MDE_and_TS_Total_N_10000_n_10000_.csv")

a1_vect <- c(0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 3, 5)
a2_vect <- c(0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 3, 5)
u_vect <- c(0.5, 0.6, 0.7, 0.8, 0.9, 0.999)

for (i in a1_vect){
  for(j in a2_vect){
    l <- 0
    for(k in u_vect){
      print(c(i, j, k))
      l <- l + 1
      c <- (min(hypvector(v = k))^i + max(hypvector(v = k))^i)/2

      assign(paste("p", l, sep = "_")
             , ggplot() +
               geom_hline(yintercept = 1, color = "grey60", linetype = 2, size = 0.9) +
               geom_hline(yintercept = 0, color = "grey60", linetype = 2, size = 0.9) +
               theme_light() +
               theme(panel.grid = element_line(color = "gray90")
                     ,panel.grid.minor = element_line("gray90")
               ) +
               stat_function(fun = pgamma # CDF of Gamma with shape = df/2 and scale = 2*0.5*(max(Pi) + min(Pi))
                             , args = list(
                               shape = 2 / 2
                               , scale = 2 * c
                             )
                             , size = 1.1
                             , xlim = c(0.
                                        , qgamma(p = 0.995
```

```
                                                 , shape = 2 /2
                                                 , scale = 2 * c
                                        )
                             )
                             , aes(colour = "a"
                                   , linetype = "a")
               ) +
               stat_function(fun = pchisq
                             , args = list(
                               df = 2
                             )
                             , size = 1.1
                             , xlim = c(0.
                                        , qgamma(p = 0.995
                                                 , shape = 2 /2
                                                 , scale = 2 * c
                                        )
                             )
                             , aes(colour = "b"
                                   , linetype = "b")
               ) +
               stat_ecdf(
                 data = data.frame(x = Data_Table[u == as.numeric(k)
                                                  & a2 == as.numeric(j)
                                                  & a1 == as.numeric(i)
                                                  , BHHJTestStat ])
                 , aes(x = x
                       , colour = "c"
                       , linetype = "c")
                 #   , col = "red"
                 , pad = FALSE
               ) +
               coord_cartesian(xlim = c(0,  qgamma(p = 0.993
                                                   , shape = 2 /2
                                                   , scale = 2 * c
               )
               )
               )+
               scale_y_continuous(name = "F(x)") +
               scale_linetype_manual(""
                                     , values = c("a" = 2
                                                  ,"b" = 6
                                                  ,"c" = 1)
                                     , labels = c(TeX("CDF of $c\\chi^2_2$")
                                                  , TeX("CDF of $\\chi^2_2$")
                                                  , TeX("ECDF of $T_{\\Phi_1}^{\\alpha_1}$ $(\\hat{\\theta}_{\\Phi_2}^{\\alpha_2})$")
                                     )
               ) + # We define the names of the groups here, because we are using latex expressions
               scale_colour_manual(""
                                   , values = c("a" = "blue"
                                                ,"b" = "black"
                                                ,"c" = "red")
                                   , labels = c(TeX("CDF of $c\\chi^2_2$")
                                                , TeX("CDF of $\\chi^2_2$")
                                                , TeX("ECDF of $T_{\\Phi_1}^{\\alpha_1}$ $(\\hat{\\theta}_{\\Phi_2}^{\\alpha_2})$")
                                   )
               ) + # We define the names of the groups here, because we are using latex expressions
               ggtitle(TeX(paste("$\\theta = $", k))) +
               theme(plot.title = element_text(hjust = 0.5)
                     ,legend.key.width = unit(1.5,"cm")) +
               guides(linetype = guide_legend(
                 override.aes = list(size = c(1.1, 1.1, 1))))
             )
    }
    pdf(file = paste("a1", i, "a2", j, ".pdf", sep = "_" ),
        width = 7,
```

```
        height = 10,
        onefile = FALSE)

    print(ggarrange(p_1, p_2, p_3, p_4, p_5, p_6
                    , ncol = 2
                    , nrow = 3
                    , common.legend = TRUE
                    , legend="bottom")
          )

    dev.off()
  }
}
```

C.1.2 Type I error of $T^{\alpha_1}_{\Phi_1}\left(\hat{\theta}^{\alpha_2}_{\Phi_2}\right)$

```
# Packages
library("rJava")
library("xlsx")

u <- 0.5 # Value of the parameter theta. Repeat for u = 0.6, 0.7, 0.8, 0.9 and 0.99

# The alpha parameter for the BHHJ Test Statistic
a1_vector <- c(0.01, 0.05, 0.1, 0.5, 3)

# The alpha parameter for the BHHJ Minimum Distance Estimator
a2_vector <- c(0.01, 0.05, 0.1, 0.5, 5)

n <- 10000 # Sample size.
N <- 100000 # Number of simulated samples.

# Definition of the divergence measure (in order to use it for the parameter
    estimation and the test statistic respectivly):
BHHJ_Divergence_Function <- function(u, alpha, probvector) {
    z <- sum(c(u / 2
               , u * (1 - u / 2) - 1 / 8
               , (1 - u) / 2
               , (1 - u) ^ 2 / 2 + 1 / 8
               ) ^ (1 + alpha)
             )
        - (1 + 1 / alpha) * sum(
                probvector * (c(u / 2
                                , u * (1 - u / 2) - 1 / 8
                                , (1 - u) / 2
                                , (1 - u) ^ 2 / 2 + 1 / 8) ^ alpha)
                                )
        + (1 / alpha) * sum(probvector ^ (1 + alpha))
    return(z)
}

# Vector of parameters of Multinomial Distribution
multpar <- c(u / 2
             , u * (1 - u / 2) - 1 / 8
             , (1 - u) / 2
             , (1 - u) ^ 2 / 2 + 1 / 8)

# Vector of Type I Error for the 5 values of a1
TypeIErrorVector <- rep(NA, length(a1_vector))
TypeIErrorComparisonTable <- NULL

for (l in 1:length(a2_vector)) {
    a2 <- a2_vector[l]
    for (k in 1:length(a1_vector)) {
        a1 <- a1_vector[k]
        c1 <- 0.5 * (max(multpar ^ a1) + min(multpar ^ a1))
        f <- (1 + a1)

```

```
        # Definition of the (Phi, alpha) - power divergence test statistic (it
            depends on the parameter a1)
        BHHJ_Test_Statistic <- function(u1, alpha1, probvector1) {
            z <- ((2 * n) / f) * BHHJ_Divergence_Function(u = u1, alpha = alpha1,
                probvector = probvector1)
            return(z)
        }

        # Computation of the MDE and the Test Statistic:
        BHHJ_Test_Statistic_VECTOR <- rep(NA, N)

        for (i in 1:N) {
            set.seed(i)
            # The Sample.
            mult_obs <- rmultinom(n = 1, size = n, prob = multpar)
            # Vector of Relative Frequencies
            relfreq <- as.vector(mult_obs / n)

            # Minimum Divergence Estimation for each one of the N samples:
            BHHJ_MDE <- optimise(BHHJ_Divergence_Function
                                 , interval = c(0.134, 1)
                                 , alpha = a2
                                 , probvector = relfreq)$minimum

            BHHJ_Test_Statistic_VECTOR[i] <-
                 BHHJ_Test_Statistic(u1 = BHHJ_MDE
                                     , alpha1 = a1
                                     , probvector1 = relfreq)

        }

        # Number of times that we reject the null hypothesis
        RejectionOfNullHypothesis <- BHHJ_Test_Statistic_VECTOR[BHHJ_Test_Statistic_
            VECTOR > c1 * qchisq(0.95, df = 2)]

        # The proportion of times that we reject the null hypotesis. Each column of
            the vector corresponds to a different value of a1
        TypeIErrorVector[k] <- length(RejectionOfNullHypothesis) / length(BHHJ_Test_
            Statistic_VECTOR)
    }

    temp <- TypeIErrorVector

    # Each line of the table below corresponds to a different value of a2.
    TypeIErrorComparisonTable <- rbind(TypeIErrorComparisonTable, temp)
}

paste('theta', '=' , u)
rownames(TypeIErrorComparisonTable) <-
  c('a2=0.01', 'a2=0.05', 'a2=0.1', 'a2=0.5', 'a2=5')
colnames(TypeIErrorComparisonTable) <-
  c('a1=0.01', 'a1=0.05', 'a1=0.1', 'a1=0.5', 'a1=3')
TypeIErrorComparisonTable <- round(TypeIErrorComparisonTable * 100, digits =2)

TypeIErrorComparisonTable

# Write the table in xlsx format
write.xlsx(TypeIErrorComparisonTable, file = "TypeIError.xlsx", sheetName = "Theta
    0,5", col.names = TRUE, row.names = TRUE)
```

C.2 Comparison of test statistics

C.2.1 Gamma(0.44) under the equiprobable model

```
# Packages
library(statmod)
library(rJava)
```

```
library(xlsx)

N <- 10000 # Number of simulated samples.
n <- 25 # Number of observations. Repeat for n = 50, 75, 100, 500

# Definition of the divergence functions (in order to use them for the parameter estimation and the test statistics respectivly):
BHHJ_Divergence_Function <- function(u, alpha, probvector) {
    z <- sum(c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u)) ^ (1 + alpha)) -
            (1 + 1 / alpha) * sum(probvector * c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u)) ^ alpha) +
                (1 / alpha) * sum(probvector ^ (1 + alpha))
    return(z)
}

KL_Divergence_Function <- function(u, probvector) {
    vect <- probvector * log(probvector)
    vect[vect == 'NaN'] <- 0
    z <- sum(vect) - sum(probvector * log(c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u))))
    return(z)
}

Pearsons_Divergence_Function <- function(u, probvector) {
    z <- (1 / 2) * sum(((probvector - c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u))) ^ 2)
              / c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u)))
    return(z)
}

CR_Divergence_Function <- function(u, lamda, probvector) {
    if (lamda == 0) {
        vect <- probvector * log(probvector)
        vect[vect == 'NaN'] <- 0
        z <- sum(vect) - sum(probvector * log(c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u))))
    } else if (lamda == -1) {
        z <- sum(c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u))
             * (log(c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u))) - log(probvector)))
    } else {
        z <- (1 / (lamda * (lamda + 1))) * (sum((probvector ^ (lamda + 1)) / (c(pgamma(quant[1], u), pgamma(quant[2], u) - pgamma(quant[1], u), 1 - pgamma(quant[2], u))) ^ lamda) - 1)
    }
    return(z)
}

# Definition of the Test Statistics
BHHJ_Test_Statistic <- function(u1, alpha1, probvector1) {
    z <- ((2 * n) /(1 + alpha1)) * BHHJ_Divergence_Function(u = u1, alpha = alpha1, probvector = probvector1)
    return(z)
}

KL_Test_Statistic <- function(u1, probvector1) {
    z <- 2 * n * KL_Divergence_Function(u = u1, probvector = probvector1)
    return(z)
}

Pearsons_Test_Statistic <- function(u1, probvector1) {
    z <- 2*n * Pearsons_Divergence_Function(u = u1, probvector = probvector1)
    return(z)
}

CR_Test_Statistic <- function(u1, lamda1, probvector1) {
    if (lamda1 == 0) {
        z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector = probvector1)
```

```
    } else if (lamda1 == -1) {
        z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector = probvector1)
    } else {
        z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector = probvector1)
    }
    return(z)
}

power_table <- NULL
for (l in 1:4) {
    # Computation of the MDE and the Test Statistic:
    BHHJ_0.01_0.01_Test_Statistic_VECTOR <- rep(NA, N)
    BHHJ_0.01_0.10_Test_Statistic_VECTOR <- rep(NA, N)
    BHHJ_0.10_0.01_Test_Statistic_VECTOR <- rep(NA, N)
    BHHJ_0.10_0.10_Test_Statistic_VECTOR <- rep(NA, N)
    BHHJ_0.50_0.01_Test_Statistic_VECTOR <- rep(NA, N)
    BHHJ_0.01_0.50_Test_Statistic_VECTOR <- rep(NA, N)
    BHHJ_0.00001_0.00001_Test_Statistic_VECTOR <- rep(NA, N)

    KL_Test_Statistic_VECTOR <- rep(NA, N)
    Pearsons_Test_Statistic_VECTOR <- rep(NA, N)
    CR_Test_Statistic_VECTOR <- rep(NA, N)

    RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.01 <- rep(NA, N)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.10 <- rep(NA, N)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.10_0.01 <- rep(NA, N)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.10_0.10 <- rep(NA, N)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.50_0.01 <- rep(NA, N)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.50 <- rep(NA, N)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.00001_0.00001 <- rep(NA, N)

    for (i in 1:N) {
        set.seed(i)
        # The Sample
        if (l == 1) {
            Obs <- rgamma(n = n, shape = 0.44, scale = 1)
        } else if (l == 2) {
            Obs <- rweibull(n=n, shape = 0.77, scale = 1)
        } else if (l == 3) {
            Obs <- rlnorm(n = n, meanlog = 0, sdlog = 0.729)
        } else {
            Obs <- rinvgauss(n = n, mean = 1, shape = 1)
        }
        # The 1/3 and 2/3 quantiles (for these quantiles we get the model (1/3, 1/3, 1/3))
        quant <- quantile(x = Obs, c(1/3, 2/3), names = FALSE)
        # Observed frequencies:
        n1 <- length(Obs[Obs <= quant[1]])
        n2 <- length(Obs[Obs <= quant[2]]) - length(Obs[Obs <= quant[1]])
        n3 <- n - n1 - n2

        obsfreq <- c(n1, n2, n3) # Vector of observed frequencies
        relfreq <- obsfreq / n # Vector of relative frequencies.

        # Minimum Divergence Estimation for each one of the N samples:
        BHHJ_MDE_0.01 <- optimise(BHHJ_Divergence_Function, interval = c(0, 10), alpha = 0.01, probvector = relfreq)$minimum
        BHHJ_MDE_0.10 <- optimise(BHHJ_Divergence_Function, interval = c(0, 10), alpha = 0.10, probvector = relfreq)$minimum
        BHHJ_MDE_0.50 <- optimise(BHHJ_Divergence_Function, interval = c(0, 10), alpha = 0.50, probvector = relfreq)$minimum
        BHHJ_MDE_0.00001 <- optimise(BHHJ_Divergence_Function, interval = c(0, 10), alpha = 0.00001, probvector = relfreq)$minimum

        KL_MDE <- optimise(KL_Divergence_Function, interval = c(0, 10), probvector = relfreq)$minimum
        Pearsons_MDE <- optimise(Pearsons_Divergence_Function, interval = c(0, 10), probvector = relfreq)$minimum
        CR_MDE <- optimise(CR_Divergence_Function, interval = c(0, 10), lamda = 2/3, probvector = relfreq)$minimum
```

```r

    # We use the following lines in order to compute c.
        multpar_0.01 <- c(pgamma(quant[1], BHHJ_MDE_0.01), pgamma(quant[2], BHHJ_MDE_0.01) - pgamma(quant[1], BHHJ_MDE_0.01), 1 - pgamma(quant[2], BHHJ_MDE_0.01))
        multpar_0.10 <- c(pgamma(quant[1], BHHJ_MDE_0.10), pgamma(quant[2], BHHJ_MDE_0.10) - pgamma(quant[1], BHHJ_MDE_0.10), 1 - pgamma(quant[2], BHHJ_MDE_0.10))
        multpar_0.50 <- c(pgamma(quant[1], BHHJ_MDE_0.50), pgamma(quant[2], BHHJ_MDE_0.50) - pgamma(quant[1], BHHJ_MDE_0.50), 1 - pgamma(quant[2], BHHJ_MDE_0.50))
        multpar_0.00001 <- c(pgamma(quant[1], BHHJ_MDE_0.00001), pgamma(quant[2], BHHJ_MDE_0.00001) - pgamma(quant[1], BHHJ_MDE_0.00001), 1 - pgamma(quant[2], BHHJ_MDE_0.00001))

    # We need a parameter c for each combination of a1 and a2
        c_0.01_0.01 <- 0.5 * (max(multpar_0.01 ^ 0.01) + min(multpar_0.01 ^ 0.01))
        c_0.01_0.10 <- 0.5 * (max(multpar_0.10 ^ 0.01) + min(multpar_0.10 ^ 0.01))
        c_0.10_0.01 <- 0.5 * (max(multpar_0.01 ^ 0.10) + min(multpar_0.01 ^ 0.10))
        c_0.10_0.10 <- 0.5 * (max(multpar_0.10 ^ 0.10) + min(multpar_0.10 ^ 0.10))
        c_0.50_0.01 <- 0.5 * (max(multpar_0.01 ^ 0.50) + min(multpar_0.01 ^ 0.50))
        c_0.01_0.50 <- 0.5 * (max(multpar_0.50 ^ 0.01) + min(multpar_0.50 ^ 0.01))
        c_0.00001_0.00001 <- 0.5 * (max(multpar_0.00001 ^ 0.00001) + min(multpar_0.00001 ^ 0.00001))

    #(BHHJ_a1_a2 = BHHJ_0.01_0.01)
        BHHJ_0.01_0.01_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.01, alpha1 = 0.01, probvector1 = relfreq)
        #(BHHJ_a1_a2 = BHHJ_0.01_0.10)
        BHHJ_0.01_0.10_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.10, alpha1 = 0.01, probvector1 = relfreq)
        #(BHHJ_a1_a2 = BHHJ_0.10_0.01)
        BHHJ_0.10_0.01_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.01, alpha1 = 0.10, probvector1 = relfreq)
        #(BHHJ_a1_a2 = BHHJ_0.10_0.10)
        BHHJ_0.10_0.10_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.10, alpha1 = 0.10, probvector1 = relfreq)
        #(BHHJ_a1_a2 = BHHJ_0.50_0.01)
        BHHJ_0.50_0.01_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.01, alpha1 = 0.50, probvector1 = relfreq)
        #(BHHJ_a1_a2 = BHHJ_0.01_0.50)
        BHHJ_0.01_0.50_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.50, alpha1 = 0.01, probvector1 = relfreq)
        #(BHHJ_a1_a2 = BHHJ_0.0001_0.0001)
        BHHJ_0.00001_0.00001_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.00001, alpha1 = 0.00001, probvector1 = relfreq)

        KL_Test_Statistic_VECTOR[i] <- KL_Test_Statistic(u1 = KL_MDE, probvector1 = relfreq)
        Pearsons_Test_Statistic_VECTOR[i] <- Pearsons_Test_Statistic(u1 = Pearsons_MDE, probvector1 = relfreq)
        CR_Test_Statistic_VECTOR[i] <- CR_Test_Statistic(u1 = CR_MDE, lamda1 = 2/3, probvector1 = relfreq)

        # In each loop we implement a hypothesis test, if we reject the null hypothesis the value TRUE is stored in the vector. We work in this way because the value of c is immediately related with the quantiles and for each sample the quantiles may be different.
        RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.01[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.01, alpha1 = 0.01, probvector1 = relfreq) > c_0.01_0.01 * qchisq(0.95, df = 1)
        RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.10[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.10, alpha1 = 0.01, probvector1 = relfreq) > c_0.01_0.10 * qchisq(0.95, df = 1)
        RejectionOfNullHypothesis_BHHJ_VECTOR_0.10_0.01[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.01, alpha1 = 0.10, probvector1 = relfreq) > c_0.10_0.01 * qchisq(0.95, df = 1)
        RejectionOfNullHypothesis_BHHJ_VECTOR_0.10_0.10[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.10, alpha1 = 0.10, probvector1 = relfreq) > c_0.10_0.10 * qchisq(0.95, df = 1)
```

```r
165         RejectionOfNullHypothesis_BHHJ_VECTOR_0.50_0.01[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.01, alpha1 = 0.50, probvector1 = relfreq) > c_0.50_0.01 * qchisq(0.95, df = 1)
166         RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.50[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.50, alpha1 = 0.01, probvector1 = relfreq) > c_0.01_0.50 * qchisq(0.95, df = 1)
167         RejectionOfNullHypothesis_BHHJ_VECTOR_0.00001_0.00001[i] <- BHHJ_Test_Statistic(u1 = BHHJ_MDE_0.00001, alpha1 = 0.00001, probvector1 = relfreq) > c_0.00001_0.00001 * qchisq(0.95, df = 1)
168     }
169     # The number of times that we reject the null hypothesis is given below. Here the decision rule is dependent only on the value of the test statistic, so we implement the test in
170     # an aggregate way.
171     RejectionOfNullHypothesis_KL <- KL_Test_Statistic_VECTOR[KL_Test_Statistic_VECTOR > qchisq(0.95, df = 1)]
172     RejectionOfNullHypothesis_Pearsons <- Pearsons_Test_Statistic_VECTOR[Pearsons_Test_Statistic_VECTOR > qchisq(0.95, df = 1)]
173     RejectionOfNullHypothesis_CR <- CR_Test_Statistic_VECTOR[CR_Test_Statistic_VECTOR > qchisq(0.95, df = 1)]
174 
175     # The proportion of times that we reject the null hypotesis.
176     power_BHHJ_0.01_0.01 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.01) / N
177     power_BHHJ_0.01_0.10 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.10) / N
178     power_BHHJ_0.10_0.01 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.10_0.01) / N
179     power_BHHJ_0.10_0.10 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.10_0.10) / N
180     power_BHHJ_0.50_0.01 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.50_0.01) / N
181     power_BHHJ_0.01_0.50 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.01_0.50) / N
182     power_BHHJ_0.00001_0.00001 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.00001_0.00001) / N
183 
184     power_KL <- length(RejectionOfNullHypothesis_KL) / N
185     power_Pearsons <- length(RejectionOfNullHypothesis_Pearsons) / N
186     power_CR <- length(RejectionOfNullHypothesis_CR) / N
187 
188     power_vector <- c(power_BHHJ_0.01_0.01, power_BHHJ_0.01_0.10, power_BHHJ_0.10_0.01, power_BHHJ_0.10_0.10, power_BHHJ_0.50_0.01, power_BHHJ_0.01_0.50, power_BHHJ_0.00001_0.00001, power_KL, power_Pearsons, power_CR)
189 
190     # Each line of the table below corresponds to a different value of the shape parameter.
191     power_table <- rbind(power_table, power_vector)
192 }
193 
194 paste('Model : (1/3, 1/3, 1/3)')
195 paste(expression(Sample_Size), '=', n)
196 paste(expression(Simulations), '=', N)
197 paste('(S) is the size')
198 
199 rownames(power_table) <- c('gamma(shape = 0.44, scale = 1)(S)', 'weibull(shape = 0.77, scale = 1)', 'lnorm(meanlog = 0, sdlog = 0.729)', 'invgauss(mean = 1, shape = 1)')
200 colnames(power_table) <- c(' BHHJ_0.01_0.01 ', ' BHHJ_0.01_0.10 ', ' BHHJ_0.10_0.01 ', ' BHHJ_0.10_0.10 ', ' BHHJ_0.50_0.01 ', ' BHHJ_0.01_0.50 ', ' BHHJ_0.00001_0.00001 ', ' Kullback-Leibler ', ' Pearsons ', ' Cressie Read 2/3 ')
201 power_table <- round(power_table * 100, digits = 2)
202 power_table
203 
204 # Write the table in xlsx format
205 write.xlsx(power_table, file = "Gamma(0,44).xlsx", sheetName = "n 25", col.names = TRUE, row.names = TRUE)
```

Note that for the **symmetric** model, the above code can be applied by exchanging the line 107 with the following line of code,

```r
1 quant <- quantile(x = Obs, c(0.2, 0.8), names = FALSE) # The 0.2 and 0.8 quantiles (for this quantiles we get the model (0.2, 0.6, 0.2))
```

for the **left skewed** model with,

```r
quant <- quantile(x = Obs, c(0.1, 0.6), names = FALSE) # The 0.1 and 0.6 quantiles (for this quantiles we get the model (0.1, 0.5, 0.4))
```

and for the **right skewed** with,

```r
quant <- quantile(x = Obs, c(0.4, 0.9), names = FALSE) # The 0.4 and 0.9 quantiles (for this quantiles we get the model (0.4, 0.5, 0.1))
```

For the remaining null models and the associated alternatives, the above code can be used by appropriately altering the lines 97 to 105.

C.3 Examination under contaminated data

C.3.1 Efficiency and robustness examination

```r
N = 100000 # Number of simulated samples.
n <- 200  # Number of observations.
exp_par <- 1 # Parameter of Exponential distribution. Exp(1) and Gamma(1) coincide.
cont_rate <- 0 # 0 for no contamination, 0.075 for 7.5% contamination

gamma_shape <- 1.5 # 1.5, 4 and 10.

# Null model definition
hypvector <- function(v){
  z <- c(1-exp(-v*quant[1]),
         exp(-v*quant[1]) - exp(-v*quant[2]),
         exp(-v*quant[2]) - exp(-v*quant[3]),
         exp(-v*quant[3]) - exp(-v*quant[4]),
         exp(-v*quant[4]) - exp(-v*quant[5]),
         exp(-v*quant[5]) - exp(-v*quant[6]),
         exp(-v*quant[6]) - exp(-v*quant[7]),
         exp(-v*quant[7]) - exp(-v*quant[8]),
         exp(-v*quant[8]) - exp(-v*quant[9]),
         exp(-v*quant[9]) - exp(-v*quant[10]),
         exp(-v*quant[10]) - exp(-v*quant[11]),
         exp(-v*quant[11]) - exp(-v*quant[12]),
         exp(-v*quant[12]) - exp(-v*quant[13]),
         exp(-v*quant[13]) - exp(-v*quant[14]),
         exp(-v*quant[14]))
  return(z)
}

# Definintion of Divergence Measures

CR_Divergence_Function <- function(u, lamda, probvector) {
  if (lamda == 0) {
    vect <- probvector * log(probvector)
    vect[vect == 'NaN'] <- 0
    z <- sum(vect) - sum(probvector * log(hypvector(v=u)))
  } else if (lamda == -1) {
    z <- sum(hypvector(v=u)
           * (log(hypvector(v=u)) - log(probvector)))
  } else {
    z <- (1 / (lamda * (lamda + 1))) * (sum((probvector ^ (lamda + 1)) / (hypvector(v=u)) ^ lamda) - 1)
  }
  return(z)
}

BHHJ_Divergence_Function <- function(u, alpha = a2, probvector = relfreq) {
  z <- sum(hypvector(v=u) ^ (1 + alpha)) -
    (1 + 1 / alpha) * sum(probvector * hypvector(v=u) ^ alpha) +
```

```
    (1 / alpha) * sum(probvector ^ (1 + alpha))
  return(z)
}

L2_Distance_Function <- function(u, probvector){
  z <- sqrt(sum((probvector - hypvector(v=u))^2 ))
  return(z)
}

# Definition of the Log - Likelihood function:
Log_Likelihood_Function <- function(u, obs){
  z <- length(obs)*log(u) - u * sum(obs)
  return(z)
}

# Computation of the MDE:

KL_MDE_VECTOR <- rep(NA, N)
Pe_MDE_VECTOR <- rep(NA, N)
CR_MDE_VECTOR <- rep(NA, N)
FT_MDE_VECTOR <- rep(NA, N)
MD_MDE_VECTOR <- rep(NA, N) # Minimum discrimination information statistic
MCH_MDE_VECTOR <- rep(NA, N) # Modified chi squared statistic

L2_MDE_VECTOR <- rep(NA, N)

BHHJ_MDE_0.00_VECTOR <- rep(NA, N)
BHHJ_MDE_0.01_VECTOR <- rep(NA, N)
BHHJ_MDE_0.05_VECTOR <- rep(NA, N)
BHHJ_MDE_0.10_VECTOR <- rep(NA, N)
BHHJ_MDE_0.20_VECTOR <- rep(NA, N)
BHHJ_MDE_0.30_VECTOR <- rep(NA, N)
BHHJ_MDE_0.40_VECTOR <- rep(NA, N)
BHHJ_MDE_0.50_VECTOR <- rep(NA, N)
BHHJ_MDE_0.60_VECTOR <- rep(NA, N)
BHHJ_MDE_0.70_VECTOR <- rep(NA, N)
BHHJ_MDE_0.80_VECTOR <- rep(NA, N)
BHHJ_MDE_0.90_VECTOR <- rep(NA, N)
BHHJ_MDE_1.00_VECTOR <- rep(NA, N)

MLE_VECTOR <- rep(NA, N)

ExpObs <- rep(NA, n)

for (i in 1 : N){
  set.seed(i)
  cont_vect <- runif(n) <= cont_rate
  # The Sample.
  ExpObs <- cont_vect * rgamma(n = n, shape = gamma_shape, rate = exp_par) + (1 -
      cont_vect) * rexp(n = n, rate = exp_par)
  # The quantiles
  quant <- quantile(x = ExpObs, c(1/15,
                                  2/15,
                                  3/15,
                                  4/15,
                                  5/15,
                                  6/15,
                                  7/15,
                                  8/15,
                                  9/15,
                                  10/15,
                                  11/15,
                                  12/15,
                                  13/15,
                                  14/15
                                  ),
                    names = FALSE)
  # Observed frequencies:
  n1 <- length(ExpObs[ExpObs<=quant[1]])
  n2 <- length(ExpObs[ExpObs<=quant[2]]) - length(ExpObs[ExpObs<=quant[1]])
  n3 <- length(ExpObs[ExpObs<=quant[3]]) - length(ExpObs[ExpObs<=quant[2]])
  n4 <- length(ExpObs[ExpObs<=quant[4]]) - length(ExpObs[ExpObs<=quant[3]])
```

```r
  n5 <- length(ExpObs[ExpObs<=quant[5]]) - length(ExpObs[ExpObs<=quant[4]])
  n6 <- length(ExpObs[ExpObs<=quant[6]]) - length(ExpObs[ExpObs<=quant[5]])
  n7 <- length(ExpObs[ExpObs<=quant[7]]) - length(ExpObs[ExpObs<=quant[6]])
  n8 <- length(ExpObs[ExpObs<=quant[8]]) - length(ExpObs[ExpObs<=quant[7]])
  n9 <- length(ExpObs[ExpObs<=quant[9]]) - length(ExpObs[ExpObs<=quant[8]])
  n10 <- length(ExpObs[ExpObs<=quant[10]]) - length(ExpObs[ExpObs<=quant[9]])
  n11 <- length(ExpObs[ExpObs<=quant[11]]) - length(ExpObs[ExpObs<=quant[10]])
  n12 <- length(ExpObs[ExpObs<=quant[12]]) - length(ExpObs[ExpObs<=quant[11]])
  n13 <- length(ExpObs[ExpObs<=quant[13]]) - length(ExpObs[ExpObs<=quant[12]])
  n14 <- length(ExpObs[ExpObs<=quant[14]]) - length(ExpObs[ExpObs<=quant[13]])
  n15 <- n - n1 - n2 - n3 - n4 - n5 - n6 - n7 - n8 - n9 - n10 - n11 - n12 - n13 - n14

  # Vector of observed frequencies
  obsfreq <- c(n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15)

  # Vector of relative frequencies.
  relfreq <- obsfreq/n

  # Minimum Divergence Estimation for each one of the N samples:
  CR_MDE_VECTOR[i] <- optimise(CR_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), lamda = 2/3, probvector = relfreq)$minimum
  KL_MDE_VECTOR[i] <- optimise(CR_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), lamda = 0, probvector = relfreq)$minimum
  Pe_MDE_VECTOR[i] <- optimise(CR_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), lamda = 1, probvector = relfreq)$minimum
  FT_MDE_VECTOR[i] <- optimise(CR_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), lamda = -1/2, probvector = relfreq)$minimum
  MD_MDE_VECTOR[i] <- optimise(CR_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), lamda = -1, probvector = relfreq)$minimum
  MCH_MDE_VECTOR[i] <- optimise(CR_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), lamda = -2, probvector = relfreq)$minimum

  BHHJ_MDE_0.00_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = (10^-7), probvector = relfreq)$minimum
  BHHJ_MDE_0.01_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.01, probvector = relfreq)$minimum
  BHHJ_MDE_0.05_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.05, probvector = relfreq)$minimum
  BHHJ_MDE_0.10_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.10, probvector = relfreq)$minimum
  BHHJ_MDE_0.20_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.20, probvector = relfreq)$minimum
  BHHJ_MDE_0.30_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.30, probvector = relfreq)$minimum
  BHHJ_MDE_0.40_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.40, probvector = relfreq)$minimum
  BHHJ_MDE_0.50_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.50, probvector = relfreq)$minimum
  BHHJ_MDE_0.60_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.60, probvector = relfreq)$minimum
  BHHJ_MDE_0.70_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.70, probvector = relfreq)$minimum
  BHHJ_MDE_0.80_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.80, probvector = relfreq)$minimum
  BHHJ_MDE_0.90_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 0.90, probvector = relfreq)$minimum
  BHHJ_MDE_1.00_VECTOR[i]  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), alpha = 1.00, probvector = relfreq)$minimum

  L2_MDE_VECTOR[i] <- optimise(L2_Distance_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), probvector = relfreq)$minimum

  # Maximum Likelihood Estimation for each one of the N samples :
  MLE_VECTOR[i] <- optimise(Log_Likelihood_Function, interval = c(0, (length(ExpObs)/sum(ExpObs))+10), obs = ExpObs, maximum = T)$maximum

  print(i)

}

```

```
# Mean value of the  Minimum Divergence Estimations we used and the MLE.

CR_MDE_MEAN <- mean(CR_MDE_VECTOR)
KL_MDE_MEAN <- mean(KL_MDE_VECTOR)
Pe_MDE_MEAN <- mean(Pe_MDE_VECTOR)
FT_MDE_MEAN <- mean(FT_MDE_VECTOR)
MD_MDE_MEAN <- mean(MD_MDE_VECTOR)
MCH_MDE_MEAN <- mean(MCH_MDE_VECTOR)

BHHJ_MDE_0.00_MEAN <- mean(BHHJ_MDE_0.00_VECTOR)
BHHJ_MDE_0.01_MEAN <- mean(BHHJ_MDE_0.01_VECTOR)
BHHJ_MDE_0.05_MEAN <- mean(BHHJ_MDE_0.05_VECTOR)
BHHJ_MDE_0.10_MEAN <- mean(BHHJ_MDE_0.10_VECTOR)
BHHJ_MDE_0.20_MEAN <- mean(BHHJ_MDE_0.20_VECTOR)
BHHJ_MDE_0.30_MEAN <- mean(BHHJ_MDE_0.30_VECTOR)
BHHJ_MDE_0.40_MEAN <- mean(BHHJ_MDE_0.40_VECTOR)
BHHJ_MDE_0.50_MEAN <- mean(BHHJ_MDE_0.50_VECTOR)
BHHJ_MDE_0.60_MEAN <- mean(BHHJ_MDE_0.60_VECTOR)
BHHJ_MDE_0.70_MEAN <- mean(BHHJ_MDE_0.70_VECTOR)
BHHJ_MDE_0.80_MEAN <- mean(BHHJ_MDE_0.80_VECTOR)
BHHJ_MDE_0.90_MEAN <- mean(BHHJ_MDE_0.90_VECTOR)
BHHJ_MDE_1.00_MEAN <- mean(BHHJ_MDE_1.00_VECTOR)

L2_MDE_MEAN <- mean(L2_MDE_VECTOR)

MLE_MEAN <- mean(MLE_VECTOR)

# SD value of the Minimum Divergence Estimators we used and the MLE.

CR_MDE_sd <- sd(KL_MDE_VECTOR)
KL_MDE_sd <- sd(KL_MDE_VECTOR)
Pe_MDE_sd <- sd(KL_MDE_VECTOR)
FT_MDE_sd <- sd(KL_MDE_VECTOR)
MD_MDE_sd <- sd(MD_MDE_VECTOR)
MCH_MDE_sd <- sd(MCH_MDE_VECTOR)

BHHJ_MDE_0.00_sd <- sd(BHHJ_MDE_0.00_VECTOR)
BHHJ_MDE_0.01_sd <- sd(BHHJ_MDE_0.01_VECTOR)
BHHJ_MDE_0.05_sd <- sd(BHHJ_MDE_0.05_VECTOR)
BHHJ_MDE_0.10_sd <- sd(BHHJ_MDE_0.10_VECTOR)
BHHJ_MDE_0.20_sd <- sd(BHHJ_MDE_0.20_VECTOR)
BHHJ_MDE_0.30_sd <- sd(BHHJ_MDE_0.30_VECTOR)
BHHJ_MDE_0.40_sd <- sd(BHHJ_MDE_0.40_VECTOR)
BHHJ_MDE_0.50_sd <- sd(BHHJ_MDE_0.50_VECTOR)
BHHJ_MDE_0.60_sd <- sd(BHHJ_MDE_0.60_VECTOR)
BHHJ_MDE_0.70_sd <- sd(BHHJ_MDE_0.70_VECTOR)
BHHJ_MDE_0.80_sd <- sd(BHHJ_MDE_0.80_VECTOR)
BHHJ_MDE_0.90_sd <- sd(BHHJ_MDE_0.90_VECTOR)
BHHJ_MDE_1.00_sd <- sd(BHHJ_MDE_1.00_VECTOR)

L2_MDE_sd   <- sd(L2_MDE_VECTOR)

MLE_sd <- sd(MLE_VECTOR)

# Mean Square Error of the Minimum Divergence Estimators and the MLE.

CR_MDE_MSE <- mean((CR_MDE_VECTOR - exp_par) ^ 2)
KL_MDE_MSE <- mean((KL_MDE_VECTOR - exp_par) ^ 2)
Pe_MDE_MSE <- mean((Pe_MDE_VECTOR - exp_par) ^ 2)
FT_MDE_MSE <- mean((FT_MDE_VECTOR - exp_par) ^ 2)
MD_MDE_MSE <- mean((MD_MDE_VECTOR - exp_par) ^ 2)
MCH_MDE_MSE <- mean((MCH_MDE_VECTOR - exp_par) ^ 2)


BHHJ_MDE_0.00_MSE <- mean((BHHJ_MDE_0.00_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.01_MSE <- mean((BHHJ_MDE_0.01_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.05_MSE <- mean((BHHJ_MDE_0.05_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.10_MSE <- mean((BHHJ_MDE_0.10_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.20_MSE <- mean((BHHJ_MDE_0.20_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.30_MSE <- mean((BHHJ_MDE_0.30_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.40_MSE <- mean((BHHJ_MDE_0.40_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.50_MSE <- mean((BHHJ_MDE_0.50_VECTOR - exp_par) ^ 2)
```

```
BHHJ_MDE_0.60_MSE <- mean((BHHJ_MDE_0.60_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.70_MSE <- mean((BHHJ_MDE_0.70_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.80_MSE <- mean((BHHJ_MDE_0.80_VECTOR - exp_par) ^ 2)
BHHJ_MDE_0.90_MSE <- mean((BHHJ_MDE_0.90_VECTOR - exp_par) ^ 2)
BHHJ_MDE_1.00_MSE <- mean((BHHJ_MDE_1.00_VECTOR - exp_par) ^ 2)

L2_MDE_MSE <- mean((L2_MDE_VECTOR- exp_par)^2)

MLE_MSE <- mean((MLE_VECTOR-exp_par)^2)

# Result storage

Mean_Of_N_Estimations <- c(CR_MDE_MEAN
                            ,KL_MDE_MEAN
                            ,Pe_MDE_MEAN
                            ,FT_MDE_MEAN
                            ,MD_MDE_MEAN
                            ,MCH_MDE_MEAN
                            ,BHHJ_MDE_0.00_MEAN
                            ,BHHJ_MDE_0.01_MEAN
                            ,BHHJ_MDE_0.05_MEAN
                            ,BHHJ_MDE_0.10_MEAN
                            ,BHHJ_MDE_0.20_MEAN
                            ,BHHJ_MDE_0.30_MEAN
                            ,BHHJ_MDE_0.40_MEAN
                            ,BHHJ_MDE_0.50_MEAN
                            ,BHHJ_MDE_0.60_MEAN
                            ,BHHJ_MDE_0.70_MEAN
                            ,BHHJ_MDE_0.80_MEAN
                            ,BHHJ_MDE_0.90_MEAN
                            ,BHHJ_MDE_1.00_MEAN
                            ,L2_MDE_MEAN
                            ,MLE_MEAN)

SD_Of_N_Estimations <- c(CR_MDE_sd
                          ,KL_MDE_sd
                          ,Pe_MDE_sd
                          ,FT_MDE_sd
                          ,MD_MDE_sd
                          ,MCH_MDE_sd
                          ,BHHJ_MDE_0.00_sd
                          ,BHHJ_MDE_0.01_sd
                          ,BHHJ_MDE_0.05_sd
                          ,BHHJ_MDE_0.10_sd
                          ,BHHJ_MDE_0.20_sd
                          ,BHHJ_MDE_0.30_sd
                          ,BHHJ_MDE_0.40_sd
                          ,BHHJ_MDE_0.50_sd
                          ,BHHJ_MDE_0.60_sd
                          ,BHHJ_MDE_0.70_sd
                          ,BHHJ_MDE_0.80_sd
                          ,BHHJ_MDE_0.90_sd
                          ,BHHJ_MDE_1.00_sd
                          ,L2_MDE_sd
                          ,MLE_sd)


Mean_Square_Error <- c(CR_MDE_MSE
                        ,KL_MDE_MSE
                        ,Pe_MDE_MSE
                        ,FT_MDE_MSE
                        ,MD_MDE_MSE
                        ,MCH_MDE_MSE
                        ,BHHJ_MDE_0.00_MSE
                        ,BHHJ_MDE_0.01_MSE
                        ,BHHJ_MDE_0.05_MSE
                        ,BHHJ_MDE_0.10_MSE
                        ,BHHJ_MDE_0.20_MSE
                        ,BHHJ_MDE_0.30_MSE
                        ,BHHJ_MDE_0.40_MSE
                        ,BHHJ_MDE_0.50_MSE
                        ,BHHJ_MDE_0.60_MSE
```

```
311                         ,BHHJ_MDE_0.70_MSE
312                         ,BHHJ_MDE_0.80_MSE
313                         ,BHHJ_MDE_0.90_MSE
314                         ,BHHJ_MDE_1.00_MSE
315                         ,L2_MDE_MSE
316                         ,MLE_MSE)
317
318 Minimum_Divergence_Estimator <- c("CR"
319                                   ,"KL"
320                                   ,"Pe"
321                                   ,"FT"
322                                   ,"MD"
323                                   ,"MCH"
324                                   ,"BHHJ_0.00"
325                                   ,"BHHJ_0.01"
326                                   ,"BHHJ_0.05"
327                                   ,"BHHJ_0.10"
328                                   ,"BHHJ_0.20"
329                                   ,"BHHJ_0.30"
330                                   ,"BHHJ_0.40"
331                                   ,"BHHJ_0.50"
332                                   ,"BHHJ_0.60"
333                                   ,"BHHJ_0.70"
334                                   ,"BHHJ_0.80"
335                                   ,"BHHJ_0.90"
336                                   ,"BHHJ_1.00"
337                                   ,"L2"
338                                   ,"MLE")
339
340 dt_Mean_SD_MSE_n_200_cp_0.00 <- data.frame(Mean_Of_N_Estimations, SD_Of_N_Estimations, Mean_Square_Error, stringsAsFactors = FALSE, row.names = Minimum_Divergence_Estimator)
341 dt_Mean_SD_MSE_n_200_cp_0.00[order(Mean_Square_Error), ]
342
343 dt_MSE_n_200_cp_0.00 <- t(data.frame(Mean_Square_Error, stringsAsFactors = FALSE, row.names = Minimum_Divergence_Estimator))
344 dt_MSE_n_200_cp_0.00[ ,order(Mean_Square_Error)]
345
346 save(dt_Mean_SD_MSE_n_200_cp_0.00
347      ,file = "dt_Mean_SD_MSE_n_200_cp_0.00.RData"
348 )
349
350 save(dt_MSE_n_200_cp_0.00
351      ,file = "dt_MSE_n_200_cp_0.00.RData"
352 )
```

Note that, in the above code, the object *cont_rate* in line 4 controls the existence or not of contamination data. Furthermore, the object *gamma_shape* in line 6 defines the model from which those contamination data are drown.

For the production of the associated plots, the following code was implemented. We present only the implementation regarding the no contamination case. The remaining cases can be implemented in a similar way.

```
library(fmsb)
sing_dig <- 3 # significant digits used in the signif function.

load(file = "dt_MSE_n_200_cp_0.00.RData")
load(file = "dt_MSE_n_200_cp_0.075_gamma_shape_1.5.RData")
load(file = "dt_MSE_n_200_cp_0.075_gamma_shape_4.0.RData")
load(file = "dt_MSE_n_200_cp_0.075_gamma_shape_10.0.RData")

dt_MSE_n_200_cp_0.00[, order(dt_MSE_n_200_cp_0.075_gamma_shape_10.0)]
dt_MSE_n_200_cp_0.075_gamma_shape_1.5[, order(dt_MSE_n_200_cp_0.075_gamma_shape_10.0)]
dt_MSE_n_200_cp_0.075_gamma_shape_4.0[, order(dt_MSE_n_200_cp_0.075_gamma_shape_10.0)]
```

```r
dt_MSE_n_200_cp_0.075_gamma_shape_10.0[, order(dt_MSE_n_200_cp_0.075_gamma_shape_10.0)]

# Names are defined based on the gamma_hape_10.00 where distances are in ascending order in accordance to the MSE.
names_gamma_hape_10.00 <- c(
  "CR(-2)"
  ,"L2"
  ,"BHHJ(1.00)"
  ,"BHHJ(0.90)"
  ,"BHHJ(0.80)"
  ,"BHHJ(0.70)"
  ,"CR(-1)"
  ,"BHHJ(0.60)"
  ,"BHHJ(0.50)"
  ,"BHHJ(0.40)"
  ,"CR(-1/2)"
  ,"BHHJ(0.30)"
  ,"BHHJ(0.20)"
  ,"BHHJ(0.10)"
  ,"BHHJ(0.05)"
  ,"BHHJ(0.01)"
  ,"CR(0)"
  ,"BHHJ(0.00)"
  ,"CR(2/3)"
  ,"MLE"
  ,"CR(1)"
)

morder <- order(dt_MSE_n_200_cp_0.075_gamma_shape_10.0)
length(dt_MSE_n_200_cp_0.00)

# contamination level 0.00
df_cp_0.00 <- as.data.frame(
  rbind(
    max = rep(max(dt_MSE_n_200_cp_0.00), length(dt_MSE_n_200_cp_0.00))
    , mmin = rep(min(dt_MSE_n_200_cp_0.00), length(dt_MSE_n_200_cp_0.00))
    , cp_0.00 = dt_MSE_n_200_cp_0.00[morder]
  )
)
colnames(df_cp_0.00) <- names_gamma_hape_10.00

min_mse_cp_0.00 <- signif(min(dt_MSE_n_200_cp_0.00) * 100, sing_dig )
max_mse_cp_0.00 <- signif(max(dt_MSE_n_200_cp_0.00) * 100, sing_dig )
diff_cp_0.00 <- max_mse_cp_0.00 - min_mse_cp_0.00
v1_cp_0.00 <- signif(min_mse_cp_0.00 + 1 * diff_cp_0.00/4, sing_dig )
v2_cp_0.00 <- signif(min_mse_cp_0.00 + 2 * diff_cp_0.00/4, sing_dig )
v3_cp_0.00 <- signif(min_mse_cp_0.00 + 3 * diff_cp_0.00/4, sing_dig )

pdf(file = "Radar_graph_cp_000.pdf", width = 7, height = 6)

par(mar = c(0,0,0,0))

radarchart(df_cp_0.00
           , axistype = 4
           , seg = 4 # number of segments for each axis
           , cglcol = "gray80"
           , axislabcol = "gray40"
           , caxislabels = c(min_mse_cp_0.00, v1_cp_0.00, v2_cp_0.00, v3_cp_0.00, max_mse_cp_0.00)
           , cglwd = 0.5
           ,vlcex = 0.8
)

dev.off()
```

C.3.2 Size and power examination

For reasons of space economy, the following code is given only for $\alpha_2 = 10^{-7}$. For the remaining values of the α_2 index, the code can be adjusted accordingly.

```r
N = 100000 # Number of simulated samples.
n <- 200  # Number of observations.
exp_par <- 1    # Parameter of Exponential distribution function.
cont_rate <- 0.00
gamma_shape <- 1  # 1.5, 4, 10

hypvector <- function(v){
  z <- c(1-exp(-v*quant[1]),
         exp(-v*quant[1]) - exp(-v*quant[2]),
         exp(-v*quant[2]) - exp(-v*quant[3]),
         exp(-v*quant[3]) - exp(-v*quant[4]),
         exp(-v*quant[4]) - exp(-v*quant[5]),
         exp(-v*quant[5]) - exp(-v*quant[6]),
         exp(-v*quant[6]) - exp(-v*quant[7]),
         exp(-v*quant[7]) - exp(-v*quant[8]),
         exp(-v*quant[8]) - exp(-v*quant[9]),
         exp(-v*quant[9]) - exp(-v*quant[10]),
         exp(-v*quant[10]) - exp(-v*quant[11]),
         exp(-v*quant[11]) - exp(-v*quant[12]),
         exp(-v*quant[12]) - exp(-v*quant[13]),
         exp(-v*quant[13]) - exp(-v*quant[14]),
         exp(-v*quant[14]))
  return(z)
}

# Definintion of Divergence Measures for the Estimation of the unknown parameter

CR_Divergence_Function <- function(u, lamda, probvector) {
  if (lamda == 0) {
    vect <- probvector * log(probvector)
    vect[vect == 'NaN'] <- 0
    z <- sum(vect) - sum(probvector * log(hypvector(v=u)))
  } else if (lamda == -1) {
    z <- sum(hypvector(v=u)
             * (log(hypvector(v=u)) - log(probvector)))
  } else {
    z <- (1 / (lamda * (lamda + 1))) * (sum((probvector ^ (lamda + 1)) / (hypvector(v=u)) ^ lamda) - 1)
  }
  return(z)
}

BHHJ_Divergence_Function <- function(u, alpha = a2, probvector = relfreq) {
  z <- sum(hypvector(v=u) ^ (1 + alpha)) -
    (1 + 1 / alpha) * sum(probvector * hypvector(v=u) ^ alpha) +
    (1 / alpha) * sum(probvector ^ (1 + alpha))
  return(z)
}

# Definintion of the Test Statistics

CR_Test_Statistic <- function(u1, lamda1, probvector1) {
  if (lamda1 == 0) {
    z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector = probvector1)
  } else if (lamda1 == -1) {
    z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector = probvector1)
  } else {
    z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector = probvector1)
  }
  return(z)
}

BHHJ_Test_Statistic <- function(u1, alpha1, probvector1) {
```

```
  z <- ((2 * n) / (1 + alpha1)) * BHHJ_Divergence_Function(u = u1, alpha = alpha1, probvector = probvector1)
  return(z)
}

FT_vector <- rep(NA, N)
KL_vector <- rep(NA, N)
CR_vector <- rep(NA, N)
Pe_vector <- rep(NA, N)
MD_vector <- rep(NA, N) # Minimum discrimination information statistic
MCH_vector <- rep(NA, N) # Modified chi squared statistic

BHHJ_0.00_0.00_vector <- rep(NA, N)
BHHJ_0.01_0.00_vector <- rep(NA, N)
BHHJ_0.05_0.00_vector <- rep(NA, N)
BHHJ_0.10_0.00_vector <- rep(NA, N)
BHHJ_0.20_0.00_vector <- rep(NA, N)
BHHJ_0.30_0.00_vector <- rep(NA, N)
BHHJ_0.40_0.00_vector <- rep(NA, N)
BHHJ_0.50_0.00_vector <- rep(NA, N)
BHHJ_0.60_0.00_vector <- rep(NA, N)
BHHJ_0.70_0.00_vector <- rep(NA, N)
BHHJ_0.80_0.00_vector <- rep(NA, N)
BHHJ_0.90_0.00_vector <- rep(NA, N)
BHHJ_1.00_0.00_vector <- rep(NA, N)

c_0.00_0.00_vector <- rep(NA, N)
c_0.01_0.00_vector <- rep(NA, N)
c_0.05_0.00_vector <- rep(NA, N)
c_0.10_0.00_vector <- rep(NA, N)
c_0.20_0.00_vector <- rep(NA, N)
c_0.30_0.00_vector <- rep(NA, N)
c_0.40_0.00_vector <- rep(NA, N)
c_0.50_0.00_vector <- rep(NA, N)
c_0.60_0.00_vector <- rep(NA, N)
c_0.70_0.00_vector <- rep(NA, N)
c_0.80_0.00_vector <- rep(NA, N)
c_0.90_0.00_vector <- rep(NA, N)
c_1.00_0.00_vector <- rep(NA, N)

for (i in 1 : N){
  set.seed(i)
  cont_vect <- runif(n) <= cont_rate

  # The Sample.
  Obs <- cont_vect *  rgamma(n = n, shape = gamma_shape, rate = exp_par) + (1 - cont_vect) * rexp(n = n, rate = exp_par)

  # The quantiles
  quant <- quantile(x = Obs, c(1/15,
                                  2/15,
                                  3/15,
                                  4/15,
                                  5/15,
                                  6/15,
                                  7/15,
                                  8/15,
                                  9/15,
                                  10/15,
                                  11/15,
                                  12/15,
                                  13/15,
                                  14/15
  ),
  names = FALSE)

  n1 <- length(Obs[Obs<=quant[1]])
  n2 <- length(Obs[Obs<=quant[2]]) - length(Obs[Obs<=quant[1]])
  n3 <- length(Obs[Obs<=quant[3]]) - length(Obs[Obs<=quant[2]])
  n4 <- length(Obs[Obs<=quant[4]]) - length(Obs[Obs<=quant[3]])
  n5 <- length(Obs[Obs<=quant[5]]) - length(Obs[Obs<=quant[4]])
  n6 <- length(Obs[Obs<=quant[6]]) - length(Obs[Obs<=quant[5]])
```

```r
n7 <- length(Obs[Obs<=quant[7]]) - length(Obs[Obs<=quant[6]])
n8 <- length(Obs[Obs<=quant[8]]) - length(Obs[Obs<=quant[7]])
n9 <- length(Obs[Obs<=quant[9]]) - length(Obs[Obs<=quant[8]])
n10 <- length(Obs[Obs<=quant[10]]) - length(Obs[Obs<=quant[9]])
n11 <- length(Obs[Obs<=quant[11]]) - length(Obs[Obs<=quant[10]])
n12 <- length(Obs[Obs<=quant[12]]) - length(Obs[Obs<=quant[11]])
n13 <- length(Obs[Obs<=quant[13]]) - length(Obs[Obs<=quant[12]])
n14 <- length(Obs[Obs<=quant[14]]) - length(Obs[Obs<=quant[13]])
n15 <- n - n1 - n2 - n3 - n4 - n5 - n6 - n7 - n8 - n9 - n10 - n11 - n12 - n13 - n14

# Vector of observed frequencies
obsfreq <- c(n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15)

# Vector of relative frequencies.
relfreq <- obsfreq/n

# Minimum Divergence Estimation for each one of the N samples:
optim_CR <- optimise(CR_Divergence_Function, interval = c(0, (length(Obs)/sum(Obs))+10), lamda = 2/3, probvector = relfreq)$minimum
optim_KL <- optimise(CR_Divergence_Function, interval = c(0, (length(Obs)/sum(Obs))+10), lamda = 0, probvector = relfreq)$minimum
optim_Pe <- optimise(CR_Divergence_Function, interval = c(0, (length(Obs)/sum(Obs))+10), lamda = 1, probvector = relfreq)$minimum
optim_FT <- optimise(CR_Divergence_Function, interval = c(0, (length(Obs)/sum(Obs))+10), lamda = -1/2, probvector = relfreq)$minimum
optim_MD <- optimise(CR_Divergence_Function, interval = c(0, (length(Obs)/sum(Obs))+10), lamda = -1, probvector = relfreq)$minimum
optim_MCH <- optimise(CR_Divergence_Function, interval = c(0, (length(Obs)/sum(Obs))+10), lamda = -2, probvector = relfreq)$minimum

optim_BHHJ_0.00  <- optimise(BHHJ_Divergence_Function, interval = c(0, (length(Obs)/sum(Obs))+10), alpha = (10^-7), probvector = relfreq)$minimum


c_0.00_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.0000001) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.0000001))
c_0.01_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.01) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.01))
c_0.05_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.05) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.05))
c_0.10_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.10) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.10))
c_0.20_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.20) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.20))
c_0.30_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.30) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.30))
c_0.40_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.40) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.40))
c_0.50_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.50) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.50))
c_0.60_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.60) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.60))
c_0.70_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.70) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.70))
c_0.80_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.80) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.80))
c_0.90_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 0.90) + min(hypvector(v = optim_BHHJ_0.00) ^ 0.90))
c_1.00_0.00_vector[i] <- 0.5 * (max(hypvector(v = optim_BHHJ_0.00) ^ 1.00) + min(hypvector(v = optim_BHHJ_0.00) ^ 1.00))


BHHJ_0.00_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.0000001, probvector1 = relfreq)
BHHJ_0.01_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.01, probvector1 = relfreq)
BHHJ_0.05_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.05, probvector1 = relfreq)
BHHJ_0.10_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.10, probvector1 = relfreq)
```

```
  BHHJ_0.20_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.20, probvector1 = relfreq)
  BHHJ_0.30_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.30, probvector1 = relfreq)
  BHHJ_0.40_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.40, probvector1 = relfreq)
  BHHJ_0.50_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.50, probvector1 = relfreq)
  BHHJ_0.60_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.60, probvector1 = relfreq)
  BHHJ_0.70_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.70, probvector1 = relfreq)
  BHHJ_0.80_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.80, probvector1 = relfreq)
  BHHJ_0.90_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 0.90, probvector1 = relfreq)
  BHHJ_1.00_0.00_vector[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.00, alpha1 = 1.00, probvector1 = relfreq)

  FT_vector[i] <- CR_Test_Statistic(u1 = optim_FT, lamda1 = -0.5, probvector1 = relfreq)
  KL_vector[i] <- CR_Test_Statistic(u1 = optim_KL, lamda1 =  0.0, probvector1 = relfreq)
  CR_vector[i] <- CR_Test_Statistic(u1 = optim_CR, lamda1 =  2/3, probvector1 = relfreq)
  Pe_vector[i] <- CR_Test_Statistic(u1 = optim_Pe, lamda1 =  1.0, probvector1 = relfreq)
  MD_vector[i] <- CR_Test_Statistic(u1 = optim_MD, lamda1 =  -1.0, probvector1 = relfreq)
  MCH_vector[i] <- CR_Test_Statistic(u1 = optim_MCH, lamda1 = -2.0, probvector1 = relfreq)

  print(i)
}

df_Exp_Model_TS_Size_value_n_200_cp_0.00 <- data.frame(
  FT_vector
  ,KL_vector
  ,CR_vector
  ,Pe_vector
  ,MD_vector
  ,MCH_vector

  ,BHHJ_0.00_0.00_vector
  ,BHHJ_0.01_0.00_vector
  ,BHHJ_0.05_0.00_vector
  ,BHHJ_0.10_0.00_vector
  ,BHHJ_0.20_0.00_vector
  ,BHHJ_0.30_0.00_vector
  ,BHHJ_0.40_0.00_vector
  ,BHHJ_0.50_0.00_vector
  ,BHHJ_0.60_0.00_vector
  ,BHHJ_0.70_0.00_vector
  ,BHHJ_0.80_0.00_vector
  ,BHHJ_0.90_0.00_vector
  ,BHHJ_1.00_0.00_vector


  ,c_0.00_0.00_vector
  ,c_0.01_0.00_vector
  ,c_0.05_0.00_vector
  ,c_0.10_0.00_vector
  ,c_0.20_0.00_vector
  ,c_0.30_0.00_vector
  ,c_0.40_0.00_vector
  ,c_0.50_0.00_vector
  ,c_0.60_0.00_vector
  ,c_0.70_0.00_vector
  ,c_0.80_0.00_vector
  ,c_0.90_0.00_vector
  ,c_1.00_0.00_vector
 )
```

```

BHHJ_0.00_0.00_size <- sum(BHHJ_0.00_0.00_vector > c_0.00_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.01_0.00_size <- sum(BHHJ_0.01_0.00_vector > c_0.01_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.05_0.00_size <- sum(BHHJ_0.05_0.00_vector > c_0.05_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.10_0.00_size <- sum(BHHJ_0.10_0.00_vector > c_0.10_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.20_0.00_size <- sum(BHHJ_0.20_0.00_vector > c_0.20_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.30_0.00_size <- sum(BHHJ_0.30_0.00_vector > c_0.30_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.40_0.00_size <- sum(BHHJ_0.40_0.00_vector > c_0.40_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.50_0.00_size <- sum(BHHJ_0.50_0.00_vector > c_0.50_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.60_0.00_size <- sum(BHHJ_0.60_0.00_vector > c_0.60_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.70_0.00_size <- sum(BHHJ_0.70_0.00_vector > c_0.70_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.80_0.00_size <- sum(BHHJ_0.80_0.00_vector > c_0.80_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_0.90_0.00_size <- sum(BHHJ_0.90_0.00_vector > c_0.90_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N
BHHJ_1.00_0.00_size <- sum(BHHJ_1.00_0.00_vector > c_1.00_0.00_vector * qchisq(p
     =0.05, df = 13, lower.tail = FALSE))/N

FT_null_Cum_Fun <- ecdf(FT_vector)
CR_null_Cum_Fun <- ecdf(CR_vector)
Pe_null_Cum_Fun <- ecdf(Pe_vector)
KL_null_Cum_Fun <- ecdf(KL_vector)
MD_null_Cum_Fun <- ecdf(MD_vector)
MCH_null_Cum_Fun <- ecdf(MCH_vector)

chi_sq_crit_val = qchisq(p=0.05, df = 13, lower.tail = FALSE)

FT_size <- 1 - FT_null_Cum_Fun(chi_sq_crit_val)
CR_size <- 1 - CR_null_Cum_Fun(chi_sq_crit_val)
Pe_size <- 1 - Pe_null_Cum_Fun(chi_sq_crit_val)
KL_size <- 1 - KL_null_Cum_Fun(chi_sq_crit_val)
MD_size <- 1 - MD_null_Cum_Fun(chi_sq_crit_val)
MCH_size <- 1 - MCH_null_Cum_Fun(chi_sq_crit_val)

Exp_Model_Size_a2_0.00 <- c(
  BHHJ_0.00_0.00_size
  ,BHHJ_0.01_0.00_size
  ,BHHJ_0.05_0.00_size
  ,BHHJ_0.10_0.00_size
  ,BHHJ_0.20_0.00_size
  ,BHHJ_0.30_0.00_size
  ,BHHJ_0.40_0.00_size
  ,BHHJ_0.50_0.00_size
  ,BHHJ_0.60_0.00_size
  ,BHHJ_0.70_0.00_size
  ,BHHJ_0.80_0.00_size
  ,BHHJ_0.90_0.00_size
  ,BHHJ_1.00_0.00_size
)

Exp_Model_Size_Classical <- c(
  FT_size
  ,CR_size
  ,Pe_size
  ,KL_size
  ,MD_size
  ,MCH_size
)

df_Exp_Model_Size_BHHJ_n_200_cp_0.00 <- data.frame(
  a2_0.00 = Exp_Model_Size_a2_0.00
)
```

```
295
296 row.names(df_Exp_Model_Size_BHHJ_n_200_cp_0.00) <- c(
297   "a1_0.00"
298   ,"a1_0.01"
299   ,"a1_0.05"
300   ,"a1_0.10"
301   ,"a1_0.20"
302   ,"a1_0.30"
303   ,"a1_0.40"
304   ,"a1_0.50"
305   ,"a1_0.60"
306   ,"a1_0.70"
307   ,"a1_0.80"
308   ,"a1_0.90"
309   ,"a1_1.00"
310
311 )
312
313 df_Exp_Model_Size_Classical_n_200_cp_0.00 <- data.frame(
314   FT = FT_size
315   ,CR = CR_size
316   ,Pe = Pe_size
317   ,KL = KL_size
318   ,MD = MD_size
319   ,MCH = MCH_size
320 )
321
322 save(df_Exp_Model_TS_Size_value_n_200_cp_0.00
323      ,file = "Exp_Model_TS_Size_value_n_200_cp_0.00.RData"
324 )
325
326 save(df_Exp_Model_Size_BHHJ_n_200_cp_0.00
327      ,file = "df_Exp_Model_Size_BHHJ_n_200_cp_0.00.RData"
328 )
329
330 save(df_Exp_Model_Size_Classical_n_200_cp_0.00
331      ,file = "df_Exp_Model_Size_Classical_n_200_cp_0.00.RData"
332 )
```

The code for the examination of the power is similar with the one given above. The only difference is that the line 107 is exchanged with the following code,

```
Obs <- (1 - cont_vect) * rgamma(n = n, shape = gamma_shape, rate = exp_par) + cont
    _vect * rexp(n = n, rate = exp_par)
```

Note that in this case the *gamma_shape* object given in line 5 should take a value that leads to the appropriate alternative model.

For the associated surface graphs, the following code was implemented. We present only the implementation regarding the no contamination case under the null hypothesis. The remaining cases can be implemented in a similar way.

```
library(lattice)

load("df_Exp_Model_Size_BHHJ_n_200_cp_0.00.RData")

df_Exp_Model_Size_BHHJ_n_200_cp_0.00

xval <- c(0.00, 0.01, 0.05, seq(from = 0.10, to = 1.00, by = 0.10))
yval <- c(0.00, 0.01, 0.05, seq(from = 0.10, to = 1.00, by = 0.10))
g <- expand.grid(x = xval, y = yval)

g$z <- unlist(df_Exp_Model_Size_BHHJ_n_200_cp_0.00)

pdf(file = "Surface_Chart_Size_cp_000.pdf", width = 7, height = 6)
```

```

wireframe(z ~ x * y
          , data = g
          , panel = lattice.getOption("panel.wireframe")
          , pretty = T
          #, panel.aspect = 0.99
          #, aspect = c(1, 1)
          #, groups = gr
          , scales = list(arrows = FALSE
                          , col="black"
                          #, font=10
                          , distance = c(0.7, 0.7, 0.7) # defines the distance of the labels from the axis
                          , x = list( tick.number = 10)
                          , y = list( tick.number = 10)
                          , z = list( tick.number = 8)
                        )
          , drape = TRUE
          #, shade = TRUE
          , col = "grey"
          , colorkey =list(
                            TRUE
                           , heigh = 0.65
                           , axis.line = list(min = 5)
                           #, labels = list(
                           #                label = c("a", "b")
                           #               , at = c(4, 5)
                           #  )
                           ) # the heigh of the colorkey as fraction
          , col.regions = rainbow(
                                  100
                                  , s = 1
                                  , v = 1
                                  , start = 0
                                  , end = max(1, 100 - 1) / 100
                                  , alpha = 1
                                  ) # sets the color of the surface
          #, alpha.regions = 0.85
          , par.settings = list(
                                axis.line = list(col = "transparent") # removes the panel grid
                                , regions = list(alpha = 0.85) # adds transparency
                                , layout.heights = list(
                                            top.padding = 1
                                           , main.key.padding = 0
                                           , key.axis.padding = -5
                                           #  , axis.xlab.padding = 0
                                           , xlab.key.padding = 0
                                           , key.sub.padding = 0
                                           , bottom.padding = -7
                               )
                               , layout.widths = list(
                                            left.padding = 0
                                            , key.ylab.padding = 0
                                            , ylab.axis.padding = -10
                                            , axis.key.padding = -5
                                            , right.padding = -5
                               )
                                )
          , zoom = 0.85
          , screen = list(z = -43, x = -60, y = 0) # Changes the angle
          #, light.source = c(0,0,0)
          , xlab = list(expression(alpha[1]), rot = 0)
          , ylab = list(expression(alpha[2]), rot = -0)
          , zlab = list("Size",rot=92)
          #, main = "0% Contamination"
          , zlim = c(0,0.07)
)
dev.off()
```

C.3.3 Empirical against nominal size graphs

The following code refers to the case of no contamination under the null model. The remaining cases can similarly be implemented.

```
load("Exp_Model_TS_Size_value_n_200_cp_0.00.RData")

df_Exp_Model_TS_Size_value_n_200_cp_0.00
colnames(df_Exp_Model_TS_Size_value_n_200_cp_0.00)

FT_vector <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"FT_vector"]
KL_vector <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"KL_vector"]
CR_vector <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"CR_vector"]
Pe_vector <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"Pe_vector"]
MD_vector <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"MD_vector"]
MCH_vector <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"MCH_vector"]

BHHJ_0.05_0.05_vector <-
   df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"BHHJ_0.05_0.05_vector"]
c_0.05_0.05_vector <-
   df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"c_0.05_0.05_vector"]

BHHJ_0.05_0.90_vector <-
   df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"BHHJ_0.05_0.90_vector"]
c_0.05_0.90_vector <-
   df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"c_0.05_0.90_vector"]

BHHJ_0.30_0.30_vector <-
   df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"BHHJ_0.30_0.30_vector"]
c_0.30_0.30_vector <-
   df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"c_0.30_0.30_vector"]

BHHJ_0.90_0.30_vector <-
   df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"BHHJ_0.90_0.30_vector"]
c_0.90_0.30_vector <-
   df_Exp_Model_TS_Size_value_n_200_cp_0.00[,"c_0.90_0.30_vector"]

N <- 100000
size <- seq(from = 0.00000, to = 0.10, by = 0.001)

#    Dale's Criterion :
d<-0.35 # d = 0.35 --> Close to the nominal sizes
dl.vector <- rep(NA, length(size))
du.vector <- rep(NA, length(size))

for(i in 1: length(size)){
  dl.vector[i] <- 1/(1+(exp(d + log((1-size[i])/size[i]))))
  du.vector[i] <- 1/(1+(exp(-d + log((1-size[i])/size[i]))))
}

# Nominal - Simulated Size plot

BHHJ_0.05_0.05_size <- rep(x=NA, times = length(size))
BHHJ_0.05_0.90_size <- rep(x=NA, times = length(size))
BHHJ_0.30_0.30_size <- rep(x=NA, times = length(size))
BHHJ_0.90_0.30_size <- rep(x=NA, times = length(size))


for(i in 1 : length(size)){
  BHHJ_0.05_0.05_size[i] <- sum(BHHJ_0.05_0.05_vector > c_0.05_0.05_vector * qchisq(
      p=size[i], df = 13, lower.tail = FALSE))/N
  BHHJ_0.05_0.90_size[i] <- sum(BHHJ_0.05_0.90_vector > c_0.05_0.90_vector * qchisq(
      p=size[i], df = 13, lower.tail = FALSE))/N
  BHHJ_0.30_0.30_size[i] <- sum(BHHJ_0.30_0.30_vector > c_0.30_0.30_vector * qchisq(
      p=size[i], df = 13, lower.tail = FALSE))/N
  BHHJ_0.90_0.30_size[i] <- sum(BHHJ_0.90_0.30_vector > c_0.90_0.30_vector * qchisq(
      p=size[i], df = 13, lower.tail = FALSE))/N
}


FT_null_Cum_Fun <- ecdf(FT_vector)
CR_null_Cum_Fun <- ecdf(CR_vector)
```

```
Pe_null_Cum_Fun <- ecdf(Pe_vector)
KL_null_Cum_Fun <- ecdf(KL_vector)
MD_null_Cum_Fun <- ecdf(MD_vector)
MCH_null_Cum_Fun <- ecdf(MCH_vector)


chi_sq_crit_val = qchisq(p=size, df = 13, lower.tail = FALSE)

FT_size <- 1 - FT_null_Cum_Fun(chi_sq_crit_val)
CR_size <- 1 - CR_null_Cum_Fun(chi_sq_crit_val)
Pe_size <- 1 - Pe_null_Cum_Fun(chi_sq_crit_val)
KL_size <- 1 - KL_null_Cum_Fun(chi_sq_crit_val)
MD_size <- 1 - MD_null_Cum_Fun(chi_sq_crit_val)
MCH_size <- 1 - MCH_null_Cum_Fun(chi_sq_crit_val)

opar <- par(no.readonly = T) # save current graphical parameters

par(mar = c(3.5, 4.1, 1.5, 1.5)) # Change the margins of the plot

library(ggplot2)
library(hrbrthemes)
library(directlabels)
library(dplyr)
library(ggrepel)
library(scales)
hrbrthemes::import_roboto_condensed()

mydata <- data.frame(size, dl.vector, du.vector)
nom_size <- rep(size, times = 10)
emp_size <- c(FT_size
              , CR_size
              , Pe_size
              , KL_size
              , MD_size
              , MCH_size
              , BHHJ_0.05_0.05_size
              , BHHJ_0.05_0.90_size
              , BHHJ_0.30_0.30_size
              , BHHJ_0.90_0.30_size
              )

test_stat <- c(rep("FT", times = length(FT_size))
               , rep("CR", times = length(CR_size))
               , rep("CS", times = length(Pe_size))
               , rep("LR", times = length(MD_size))
               , rep("MDI", times = length(MCH_size))
               , rep("MCS", times = length(KL_size))
               , rep("T1", times = length(KL_size))
               , rep("T2", times = length(KL_size))
               , rep("T3", times = length(KL_size))
               , rep("T4", times = length(KL_size))
               )

mydata_1 <- data.frame(test_stat, nom_size, emp_size)

pdf(file = "Nominal_Simulated_n_200_cp_000.pdf", width = 7, height = 6)

ggplot(data = mydata_1) +
  geom_ribbon(data = mydata, aes(x= size, ymin = dl.vector, ymax = du.vector), fill = "grey", alpha = 0.3) +
  geom_line(data=mydata, aes(x=size, y=size), color = "black", size = 1, linetype = "dashed") +
  geom_line(aes(x=nom_size, y=emp_size, color = test_stat), size = 0.75) +
 # geom_point(aes(x=nom_size, y=emp_size, color = test_stat, shape = test_stat), size = 1)+
 # geom_dl(aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat),
 #         method = list(dl.trans(size = size + 1.5), "last.points", cex = 0.7)) + # for labeling the lines.
  theme_light() +
  theme(panel.grid = element_line(color = "gray90"),
        panel.grid.minor = element_line("gray97"),
        legend.position = "none"
```

```
      ) +
  scale_y_continuous(minor_breaks = seq(0 , 0.14, 0.01), breaks = seq(0, 0.14, 0.01), limits = c(0, 0.143)) +
  scale_x_continuous(minor_breaks = seq(0 , 0.10, 0.01), breaks = seq(0, 0.10, 0.01), limits = c(0, 0.101)) +
  labs(x= "Nominal size", y = "Exact simulated size") +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "CS"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat, segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
                  nudge_y = 0.000,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "CR"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat, segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
                  nudge_y = 0.000,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "T1"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat, segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
                  nudge_y = 0.002,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "T2"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat, segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
                  nudge_y = 0.002,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "T3"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat, segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
                  nudge_y = -0.001,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "T4"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat, segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
                  nudge_y = 0.000,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "LR"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat, segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
```

```
                  nudge_y = -0.001,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "FT"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat,
                      segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
                  nudge_y = 0.002,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "MCS"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat,
                      segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.002,
                  nudge_y = -0.001,
                  cex = 2.8
                  # na.rm = TRUE
  ) +
  geom_text_repel(data = subset(mydata_1, nom_size == 0.1 & test_stat == "MDI"),
                  aes(x=nom_size, y=emp_size,label = test_stat, color = test_stat,
                      segment.size = 0.2),
                  box.padding = -0.1,
                  point.padding = -0.1,
                  nudge_x = 0.001,
                  nudge_y = 0.000,
                  cex = 2.8
                  # na.rm = TRUE
  )
dev.off()
```

C.3.4 ROC curves

The following code refers to the case of no contaminating data when the alternative model is the gamma with shape parameter 1.5. The remaining cases can similarly be implemented.

```
load("Exp_Model_TS_Size_value_n_200_cp_0.00.RData")
load("Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5.RData")

df_Exp_Model_TS_Size_value_n_200_cp_0.00
df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5

names(df_Exp_Model_TS_Size_value_n_200_cp_0.00)
names(df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5)

FT_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"FT_vector"]
KL_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"KL_vector"]
CR_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"CR_vector"]
Pe_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"Pe_vector"]
MD_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"MD_vector"]
MCH_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"MCH_vector"]
BHHJ_0.05_0.05_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"BHHJ
    _0.05_0.05_vector"]
BHHJ_0.05_0.90_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"BHHJ
    _0.05_0.90_vector"]
BHHJ_0.30_0.30_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"BHHJ
    _0.30_0.30_vector"]
BHHJ_0.90_0.30_vector_Size <- df_Exp_Model_TS_Size_value_n_200_cp_0.00[ ,"BHHJ
    _0.90_0.30_vector"]

FT_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"FT_
    vector"]
```

```
KL_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"KL_vector"]
CR_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"CR_vector"]
Pe_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"Pe_vector"]
MD_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"MD_vector"]
MCH_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"MCH_vector"]
BHHJ_0.05_0.05_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"BHHJ_0.05_0.05_vector"]
BHHJ_0.05_0.90_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"BHHJ_0.05_0.90_vector"]
BHHJ_0.30_0.30_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"BHHJ_0.30_0.30_vector"]
BHHJ_0.90_0.30_vector_Power <- df_Exp_Model_TS_Power_value_n_200_cp_0.000_gamma_shape_1.5[ ,"BHHJ_0.90_0.30_vector"]

size <- seq(from = 0.0, to = 1, by = 0.0001)

# Exact critical values.
FT_crit_val <- quantile(x = FT_vector_Size, probs = 1-size)
KL_crit_val <- quantile(x = KL_vector_Size, probs = 1-size)
CR_crit_val <- quantile(x = CR_vector_Size, probs = 1-size)
Pe_crit_val <- quantile(x = Pe_vector_Size, probs = 1-size)
MD_crit_val <- quantile(x = MD_vector_Size, probs = 1-size)
MCH_crit_val <- quantile(x = MCH_vector_Size, probs = 1-size)

BHHJ_0.05_0.05_crit_val<- quantile(x = BHHJ_0.05_0.05_vector_Size, probs = 1 - size)
BHHJ_0.05_0.90_crit_val<- quantile(x = BHHJ_0.05_0.90_vector_Size, probs = 1 - size)
BHHJ_0.30_0.30_crit_val<- quantile(x = BHHJ_0.30_0.30_vector_Size, probs = 1 - size)
BHHJ_0.90_0.30_crit_val<- quantile(x = BHHJ_0.90_0.30_vector_Size, probs = 1 - size)

# ECDF of the Test Statistic under the alternative hypothesis.

FT_Cum_Fun <- ecdf(FT_vector_Power)
KL_Cum_Fun <- ecdf(KL_vector_Power)
CR_Cum_Fun <- ecdf(CR_vector_Power)
Pe_Cum_Fun <- ecdf(Pe_vector_Power)
MD_Cum_Fun <- ecdf(MD_vector_Power)
MCH_Cum_Fun <- ecdf(MCH_vector_Power)

BHHJ_0.05_0.05_Cum_Fun <- ecdf(BHHJ_0.05_0.05_vector_Power)
BHHJ_0.05_0.90_Cum_Fun <- ecdf(BHHJ_0.05_0.90_vector_Power)
BHHJ_0.30_0.30_Cum_Fun <- ecdf(BHHJ_0.30_0.30_vector_Power)
BHHJ_0.90_0.30_Cum_Fun <- ecdf(BHHJ_0.90_0.30_vector_Power)

# Exact power of the Test Statistic.

FT_power <- 1 - FT_Cum_Fun(FT_crit_val)
KL_power <- 1 - KL_Cum_Fun(KL_crit_val)
CR_power <- 1 - CR_Cum_Fun(CR_crit_val)
Pe_power <- 1 - Pe_Cum_Fun(Pe_crit_val)
MD_power <- 1 - MD_Cum_Fun(MD_crit_val)
MCH_power <- 1 - MCH_Cum_Fun(MCH_crit_val)

BHHJ_0.05_0.05_power <- 1 - BHHJ_0.05_0.05_Cum_Fun(BHHJ_0.05_0.05_crit_val)
BHHJ_0.05_0.90_power <- 1 - BHHJ_0.05_0.90_Cum_Fun(BHHJ_0.05_0.90_crit_val)
BHHJ_0.30_0.30_power <- 1 - BHHJ_0.30_0.30_Cum_Fun(BHHJ_0.30_0.30_crit_val)
BHHJ_0.90_0.30_power <- 1 - BHHJ_0.90_0.30_Cum_Fun(BHHJ_0.90_0.30_crit_val)

opar <- par(no.readonly = T) # save current graphical parameters

par(mar = c(3.5, 4.1, 1.5, 1.5)) # Change the margins of the plot (first -> bottom margin)

emp_size <- rep(size, times = 10)
emp_power <- c(FT_power
               , CR_power
               , Pe_power
               , KL_power
```

```
               , MD_power
               , MCH_power
               , BHHJ_0.05_0.05_power
               , BHHJ_0.05_0.90_power
               , BHHJ_0.30_0.30_power
               , BHHJ_0.90_0.30_power
               )

test_stat <- c(rep("FT", times = length(size))
               , rep("CR", times = length(size))
               , rep("CS", times = length(size))
               , rep("LR", times = length(size))
               , rep("MDI", times = length(size))
               , rep("MCS", times = length(size))
               , rep("T1", times = length(size))
               , rep("T2", times = length(size))
               , rep("T3", times = length(size))
               , rep("T4", times = length(size))
               )

size_power_data <- data.frame(test_stat, emp_size, emp_power)

library("ggplot2")
library(dplyr)
library(hrbrthemes)
library(grid)
library(directlabels)
library(ggrepel)
library("scales")
hrbrthemes::import_roboto_condensed()

hex_codes1 <- hue_pal()(7) # Identify hex codes
hex_codes1

mycolors <- c("#F8766D", "#C49A00", "#53B400", "#00C094", "#00B6EB", "#A58AFF", "#FB61D7")
mycolors[c(2,3,6,7)]


pdf(file = "Size_Power_n_200_cp_0000_gs_015.pdf", width = 7, height = 6)

ggplot(data = size_power_data, ) +
        geom_abline(intercept=0,slope=1,colour="gray80", linetype = "dashed", size = 0.1)+
        #geom_line(data=size_power_data, aes(x=emp_size, y=emp_size), color = "black", size = 0.5, linetype = "dashed") +
        geom_line(aes(x=emp_size, y=emp_power, color = test_stat), size = 0.5) +
        # geom_point(aes(x=nom_size, y=emp_size, color = test_stat, shape = test_stat), size = 1)+
        geom_text_repel(data = subset(size_power_data, emp_size == 0.150 & test_stat == "FT"),
                         aes(x=emp_size, y=emp_power,label = test_stat, color = test_stat, segment.size = 0.2),
                         box.padding = -0.1,
                         point.padding = -0.1,
                         nudge_x = 0.01,
                         nudge_y = 0.05,
                        cex = 2.8
                         # na.rm = TRUE
        ) +
        geom_text_repel(data = subset(size_power_data, emp_size == 0.05 & test_stat == "CR"),
                        aes(x=emp_size, y=emp_power,label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
                        nudge_x = -0.01,
                        nudge_y = -0.05,
                        cex = 2.8
                        # na.rm = TRUE
        ) +
```

```
        geom_text_repel(data = subset(size_power_data, emp_size == 0.025 & test_stat == "CS"),
                        aes(x=emp_size, y=emp_power, label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
                        nudge_x = 0.01,
                        nudge_y = -0.01,
                        cex = 2.8
                        # na.rm = TRUE
        ) +
        geom_text_repel(data = subset(size_power_data, emp_size == 0.075 & test_stat == "LR"),
                        aes(x=emp_size, y=emp_power, label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
                        nudge_x = -0.01,
                        nudge_y = -0.04,
                        cex = 2.8
                        # na.rm = TRUE
        ) +
        geom_text_repel(data = subset(size_power_data, emp_size == 0.200 & test_stat == "MDI"),
                        aes(x=emp_size, y=emp_power, label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
                        nudge_x = 0.01,
                        nudge_y = 0.04,
                        cex = 2.8
                        # na.rm = TRUE
        ) +
        geom_text_repel(data = subset(size_power_data, emp_size == 0.248 & test_stat == "MCS"),
                        aes(x=emp_size, y=emp_power, label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
                        nudge_x = 0.01,
                        nudge_y = 0.025,
                        cex = 2.8
                        # na.rm = TRUE
        ) +
        geom_text_repel(data = subset(size_power_data, emp_size == 0.10 & test_stat == "T1"),
                        aes(x=emp_size, y=emp_power, label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
                        nudge_x = -0.01,
                        nudge_y = -0.04,
                        cex = 2.8
                        # na.rm = TRUE
        ) +
        geom_text_repel(data = subset(size_power_data, emp_size == 0.125 & test_stat == "T2"),
                        aes(x=emp_size, y=emp_power, label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
                        nudge_x = -0.01,
                        nudge_y = -0.03,
                        cex = 2.8
                        # na.rm = TRUE
        ) +
        geom_text_repel(data = subset(size_power_data, emp_size == 0.170 & test_stat == "T3"),
                        aes(x=emp_size, y=emp_power, label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
```

```
                        nudge_x = 0.01,
                        nudge_y = 0.04,
                        cex = 2.8
                        # na.rm = TRUE
        ) +
        geom_text_repel(data = subset(size_power_data, emp_size == 0.281 & test_stat == "T4"),
                        aes(x=emp_size, y=emp_power,label = test_stat, color = test_stat, segment.size = 0.2),
                        box.padding = -0.1,
                        point.padding = -0.1,
                        nudge_x = 0.01,
                        nudge_y = 0.02,
                        cex = 2.8
                        # na.rm = TRUE
        ) +

        #geom_dl(aes(x=emp_size, y=emp_power,label = test_stat, color = test_stat),
        #        method = list(dl.trans(size = emp_size + 0.5), "last.points", cex = 0.7)) +
        theme_light() +
        theme(panel.grid = element_line(color = "gray90"),
              panel.grid.minor = element_line("gray97"),
              legend.position = "none"
        ) +
        scale_y_continuous(minor_breaks = seq(0 , 1, 0.05), breaks = seq(0, 1, 0.1), limits = c(NA, 1)) +
        scale_x_continuous(minor_breaks = seq(0 , 1, 0.05), breaks = seq(0, 1, 0.1), limits = c(NA, 1)) +
        #scale_color_manual(values = mycolors[c(1,4,6,7)]) + # manually defines the color pallet.
        labs(x= "Empirical size", y = "Empirical power")

dev.off()
```

C.4 Conditional independence

For reasons of space economy, the following code is given only for $\alpha_1 = 10^{-7}$. For the remaining values of the α_1 index, the code can be adjusted accordingly.

```
library(nloptr)

n = 40 #20 #25, #40, # 45
N = 100000

q1 <- 0.036254
q2 <- 0.164994
q3 <- 0.092809
q4 <- 0.133645
q5 <- 0.092809
q6 <- 0.133645
q7 <- 0.237591
q8 <- 0.108253


# For w = 0 we get the Conditional Independent Models
# while for w <> 0 we get the Alternative Models
# 0 <= w < 1
w <- 0.90 #0.00, 0.15, 0.30, 0.45, 0.60, 0.75, 0.90

# Multinomial Parameters
p1 <- q1 - q1 * w
p2 <- q2 + q1 * w - q8 * w
p3 <- q3 + q8 * w
p4 <- q4 + q1 * w
p5 <- q5 + q8 * w - q1 * w
p6 <- q6 + q1 * w - q8 * w
```

```r
p7 <- q7 + q8 * w - q1 * w
p8 <- q8 - q8 * w

CI_Model <- c(p1, p2, p3, p4, p5, p6, p7, p8)

# Definintion of Divergence Measures for the Estimation of the unknown parameter

CR_Divergence_Function <- function(u, lamda, probvector) {
    if (lamda == 0) {
        vect <- probvector * log(probvector)
        vect[vect == 'NaN'] <- 0
        z <- sum(vect) - sum(probvector * log(c(u[1], u[2], u[3], u[4], u[5], u[6],
            u[7], u[8])))
    } else if (lamda == -1) {
        z <- sum(c(u[1], u[2], u[3], u[4], u[5], u[6], u[7], u[8])
             * (log(c(u[1], u[2], u[3], u[4], u[5], u[6], u[7], u[8])) - log(
                probvector)))
    } else {
        z <- (1 / (lamda * (lamda + 1))) * (sum((probvector ^ (lamda + 1)) / (c(u
            [1], u[2], u[3], u[4], u[5], u[6], u[7], u[8])) ^ lamda) - 1)
    }
    return(z)
}

BHHJ_Divergence_Function <- function(u, alpha = a2, probvector = relfreq) {
    z <- sum(c(u[1], u[2], u[3], u[4], u[5], u[6], u[7], u[8]) ^ (1 + alpha)) -
        (1 + 1 / alpha) * sum(probvector * c(u[1], u[2], u[3], u[4], u[5], u[6], u
            [7], u[8]) ^ alpha) +
        (1 / alpha) * sum(probvector ^ (1 + alpha))
    return(z)
}


# Definintion of the Test Statistics
CR_Test_Statistic <- function(u1, lamda1, probvector1) {
    if (lamda1 == 0) {
        z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector =
            probvector1)
    } else if (lamda1 == -1) {
        z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector =
            probvector1)
    } else {
        z <- 2 * n * CR_Divergence_Function(u = u1, lamda = lamda1, probvector =
            probvector1)
    }
    return(z)
}

BHHJ_Test_Statistic <- function(u1, alpha1, probvector1) {
    z <- ((2 * n) / (1 + alpha1)) * BHHJ_Divergence_Function(u = u1, alpha = alpha1,
         probvector = probvector1)
    return(z)
}

# The constraint function
nloptr_constfunc <- function(u) {
    g1 <- u[1] * u[7] - u[3] * u[5]
    g2 <- u[2] * u[8] - u[4] * u[6]
    g3 <- u[1] + u[2] + u[3] + u[4] + u[5] + u[6] + u[7] + u[8] - 1
    return(c(g1, g2, g3))
}

inval <- c(0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5) # Initial values

lowbound <- rep(0.00001, 8)
uppbound <- rep(0.99999, 8)

lamda_1_vector_m0.5 <- rep(NA, N)
lamda_1_vector_0.0  <- rep(NA, N)
lamda_1_vector_CR   <- rep(NA, N)
lamda_1_vector_1.0  <- rep(NA, N)

```

```r
BHHJ_0.0000001_0.0000001_Test_Statistic_VECTOR <- rep(NA, N)
BHHJ_0.0000001_0.01_Test_Statistic_VECTOR <- rep(NA, N)
BHHJ_0.0000001_0.05_Test_Statistic_VECTOR <- rep(NA, N)
BHHJ_0.0000001_0.1_Test_Statistic_VECTOR <- rep(NA, N)
BHHJ_0.0000001_0.5_Test_Statistic_VECTOR <- rep(NA, N)
BHHJ_0.0000001_1.5_Test_Statistic_VECTOR <- rep(NA, N)

RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.0000001 <- rep(NA, N)
RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.01 <- rep(NA, N)
RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.05 <- rep(NA, N)
RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.1 <- rep(NA, N)
RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.5 <- rep(NA, N)
RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_1.5 <- rep(NA, N)


for (i in 1:N) {
    set.seed(i)
    Obs <- as.vector(rmultinom(n = 1, size = n, prob = CI_Model) / n)

    # Minimum Divergence Estimation for each one of the N samples
    optim_FT <- auglag(x0 = inval, fn = CR_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), lamda = -1/2, probvector = Obs)$par
    optim_ML <- auglag(x0 = inval, fn = CR_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), lamda = 0, probvector = Obs)$par
    optim_CR <- auglag(x0 = inval, fn = CR_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), lamda = 2/3, probvector = Obs)$par
    optim_Pe <- auglag(x0 = inval, fn = CR_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), lamda = 1, probvector = Obs)$par

    optim_BHHJ_0.0000001 <- auglag(x0 = inval, fn = BHHJ_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), alpha = 0.0000001, probvector = Obs)$par
    optim_BHHJ_0.01 <- auglag(x0 = inval, fn = BHHJ_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), alpha = 0.01, probvector = Obs)$par
    optim_BHHJ_0.05 <- auglag(x0 = inval, fn = BHHJ_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), alpha = 0.05, probvector = Obs)$par
    optim_BHHJ_0.1 <- auglag(x0 = inval, fn = BHHJ_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), alpha = 0.1, probvector = Obs)$par
    optim_BHHJ_0.5 <- auglag(x0 = inval, fn = BHHJ_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), alpha = 0.5, probvector = Obs)$par
    optim_BHHJ_1.5 <- auglag(x0 = inval, fn = BHHJ_Divergence_Function, lower = lowbound, upper = uppbound, heq = nloptr_constfunc, localsolver = c("SLSQP"), alpha = 1.5, probvector = Obs)$par

    # We need a parameter c for each combination of a1 and a2
    c_0.0000001_0.0000001 = 0.5 * (max(optim_BHHJ_0.0000001 ^ 0.0000001) + min(optim_BHHJ_0.0000001 ^ 0.0000001))
    c_0.0000001_0.01 = 0.5 * (max(optim_BHHJ_0.01 ^ 0.0000001) + min(optim_BHHJ_0.01 ^ 0.0000001))
    c_0.0000001_0.05 = 0.5 * (max(optim_BHHJ_0.05 ^ 0.0000001) + min(optim_BHHJ_0.05 ^ 0.0000001))
    c_0.0000001_0.1 = 0.5 * (max(optim_BHHJ_0.1 ^ 0.0000001) + min(optim_BHHJ_0.1 ^ 0.0000001))
    c_0.0000001_0.5 = 0.5 * (max(optim_BHHJ_0.5 ^ 0.0000001) + min(optim_BHHJ_0.5 ^ 0.0000001))
    c_0.0000001_1.5 = 0.5 * (max(optim_BHHJ_1.5 ^ 0.0000001) + min(optim_BHHJ_1.5 ^ 0.0000001))


    BHHJ_0.0000001_0.0000001_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.0000001, alpha1 = 0.0000001, probvector1 = Obs)
    BHHJ_0.0000001_0.01_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.01, alpha1 = 0.0000001, probvector1 = Obs)
```

```
    BHHJ_0.0000001_0.05_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.05, alpha1 = 0.0000001, probvector1 = Obs)
    BHHJ_0.0000001_0.1_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.1, alpha1 = 0.0000001, probvector1 = Obs)
    BHHJ_0.0000001_0.5_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.5, alpha1 = 0.0000001, probvector1 = Obs)
    BHHJ_0.0000001_1.5_Test_Statistic_VECTOR[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_1.5, alpha1 = 0.0000001, probvector1 = Obs)


    # For each sample we implement a hypothesis test, if we reject the null hypothesis the value TRUE is stored in the vector. We work in this way because the value of c is immediately related with the quantiles and for each sample the quantiles may be different.
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.0000001[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.0000001, alpha1 = 0.0000001, probvector1 = Obs) > c_0.0000001_0.0000001 * qchisq(0.95, df = 2)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.01[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.01, alpha1 = 0.0000001, probvector1 = Obs) > c_0.0000001_0.01 * qchisq(0.95, df = 2)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.05[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.05, alpha1 = 0.0000001, probvector1 = Obs) > c_0.0000001_0.05 * qchisq(0.95, df = 2)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.1[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.1, alpha1 = 0.0000001, probvector1 = Obs) > c_0.0000001_0.1 * qchisq(0.95, df = 2)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.5[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_0.5, alpha1 = 0.0000001, probvector1 = Obs) > c_0.0000001_0.5 * qchisq(0.95, df = 2)
    RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_1.5[i] <- BHHJ_Test_Statistic(u1 = optim_BHHJ_1.5, alpha1 = 0.0000001, probvector1 = Obs) > c_0.0000001_1.5 * qchisq(0.95, df = 2)


    lamda_1_vector_m0.5[i] <- CR_Test_Statistic(u1=optim_FT, lamda1 = -0.5, probvector1=Obs)
    lamda_1_vector_0.0[i]  <- CR_Test_Statistic(u1=optim_ML, lamda1 =  0.0, probvector1=Obs)
    lamda_1_vector_CR[i]   <- CR_Test_Statistic(u1=optim_CR, lamda1 =  2/3, probvector1=Obs)
    lamda_1_vector_1.0[i]  <- CR_Test_Statistic(u1=optim_Pe, lamda1 =  1.0, probvector1=Obs)

    print(i)
}

RejectionOfNullHypothesis_lamda_1_m0.5 <- lamda_1_vector_m0.5[lamda_1_vector_m0.5 > qchisq(0.95, df = 2)]
RejectionOfNullHypothesis_lamda_1_0.0  <- lamda_1_vector_0.0[lamda_1_vector_0.0 > qchisq(0.95, df = 2)]
RejectionOfNullHypothesis_lamda_1_CR <- lamda_1_vector_CR[lamda_1_vector_CR > qchisq(0.95, df = 2)]
RejectionOfNullHypothesis_lamda_1_1.0  <- lamda_1_vector_1.0[lamda_1_vector_1.0 > qchisq(0.95, df = 2)]

Power_FT <- length(RejectionOfNullHypothesis_lamda_1_m0.5)/ N
Power_ML <- length(RejectionOfNullHypothesis_lamda_1_0.0)  / N
Power_CR <- length(RejectionOfNullHypothesis_lamda_1_CR)   / N
Power_Pe <- length(RejectionOfNullHypothesis_lamda_1_1.0)  / N

Power_BHHJ_0.0000001_0.0000001 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.0000001) / N
Power_BHHJ_0.0000001_0.01 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.01) / N
Power_BHHJ_0.0000001_0.05 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.05) / N
Power_BHHJ_0.0000001_0.1 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.1) / N
Power_BHHJ_0.0000001_0.5 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_0.5) / N
Power_BHHJ_0.0000001_1.5 <- sum(RejectionOfNullHypothesis_BHHJ_VECTOR_0.0000001_1.5) / N
```

```r


Power.Vector <-round(c(
    Power_FT,
    Power_ML,
    Power_CR,
    Power_Pe,
    Power_BHHJ_0.0000001_0.0000001,
    Power_BHHJ_0.0000001_0.01,
    Power_BHHJ_0.0000001_0.05,
    Power_BHHJ_0.0000001_0.1,
    Power_BHHJ_0.0000001_0.5,
    Power_BHHJ_0.0000001_1.5
)*100,digits=4)

Test.Statistics <- c(
    "Power_FT",
    "Power_ML",
    "Power_CR",
    "Power_Pe",
    "BHHJ_0.0000001_0.0000001",
    "BHHJ_0.0000001_0.01",
    "BHHJ_0.0000001_0.05",
    "BHHJ_0.0000001_0.1",
    "BHHJ_0.0000001_0.5",
    "BHHJ_0.0000001_1.5"
)

results <- data.frame(abs_dist, Power.Vector, stringsAsFactors = FALSE, row.names =
    Test.Statistics)
```

The code for the empirical against the nominal size graphs and the ROC curves is similar with the one presented in Subsections C.3.3 and C.3.4, respectively.

C.5 Zero frequency cells

```r
library(tibble)
library(dplyr)
library(data.table)
library(future.apply)
plan(multisession) # Activates parallel computation.

N = 10000 # Number of simulated samples.

# Definition of the Modified Divergence Measures
# Penalized BHHJ divergence
P_BHHJ_Divergence_Function <- function(u, alpha, probvector, w) {
  boolvect <- probvector > 0
  u1 <- hypvector(v=u)[boolvect]
  u2 <- hypvector(v=u)[!boolvect]
  probvector1 <- probvector[boolvect]
  z <- sum(u1 ^ (1 + alpha)) -
    (1 + 1 / alpha) * sum(probvector1 * u1 ^ alpha) +
    (1 / alpha) * sum(probvector1 ^ (1 + alpha)) +
    w * sum(u2 ^ (1 + alpha))
  return(z)
}

# Definition of the Modified Test Statistics
# Modified BHHJ Test Statistic
P_BHHJ_Test_Statistic <- function(u1, alpha1, probvector1, w1, n) {
  z <-
    ((2 * n) / (1 + alpha1)) * P_BHHJ_Divergence_Function(
      u = u1,
      alpha = alpha1,
      probvector = probvector1,
      w = w1
```

```
    )
  return(z)
}


# Parameters Definition for the composition of the Data Table

a2_vect <- c(0.01, 0.05, seq(0.1, 1, 0.1)) # a2 values.

# MDE & Squared Error Calculation

epsilon <- 0.1 * sqrt(15) # For n = 15 this value of epsilon leads to 10% contamination rate.

for (model in (c(#"01_Geometric"
                 #, "02_Poisson"
                 #, "03_Negative_Binom"
                  "04_Binomial"
                 )
               )
){
  print(model)
  if (model == "01_Geometric"){ # GEOMETRIC MODEL
    u <- 0.9                    # theta parameter
    # NULL MODEL DEFINITION
    hypvector <- function(v){
      z <- c((v),
             (v*(1-v)),
             (v*(1-v)^2),
             ((1-v)^3))
      return(z)
    }
    # CONTAMINATED MODEL DEFINITION
    cont_hypvector <- function(v){
      z <- c((1-cont_rate) * (v),
             (1-cont_rate) * (v*(1-v)),
             (1-cont_rate) * (v*(1-v)^2),
             (1-cont_rate) * ((1-v)^3) + cont_rate)
      return(z)
    }
    h2_vect <- seq(0, 80, 0.5) # h2 values.

    a2 <- rep(a2_vect, each = length(h2_vect) * N)
    h2 <- rep(h2_vect, each = N, times = length(a2_vect))

    } else if (model == "02_Poisson"){ # POISSON MODEL
      u <- 0.5 # theta parameter
      # NULL MODEL DEFINITION
      hypvector <- function(v){
        z <- c((exp(-v)),
               (v * exp(-v)),
               ((v^2 * exp(-v))/2),
               (1 - (exp(-v)) - (v * exp(-v)) - ((v^2 * exp(-v))/2)))
        return(z)
      }
      # CONTAMINATED MODEL DEFINITION
      cont_hypvector <- function(v){
        z <- c((1-cont_rate) * (exp(-v)),
               (1-cont_rate) * (v * exp(-v)),
               (1-cont_rate) * ((v^2 * exp(-v))/2),
               (1-cont_rate) * (
                 1 - (
                   exp(-v)) - (v * exp(-v)) - ((v^2 * exp(-v))/2)) + cont_rate)
        return(z)
      }

      h2_vect <- seq(0, 18, 0.1) # h2 values.

      a2 <- rep(a2_vect, each = length(h2_vect) * N)
      h2 <- rep(h2_vect, each = N, times = length(a2_vect))

    } else if (model == "03_Negative_Binom"){ # NEGATIVE BINOMIAL MODEL
```

```
      u <- 0.1 # theta parameter
      # NULL MODEL DEFINITION
      hypvector <- function(v){
        z <- c(((1-v)^2),
               (2*v*(1-v)^2),
               (3*v^2*(1-v)^2),
               (1-((1-v)^2*((1+v)^2+2*v^2))))
        return(z)
      }
      # CONTAMINATED MODEL DEFINITION
      cont_hypvector <- function(v){
        z <- c((1-cont_rate) * ((1-v)^2),
               (1-cont_rate) * (2*v*(1-v)^2),
               (1-cont_rate) * (3*v^2*(1-v)^2),
               (1-cont_rate) * (1-((1-v)^2*((1+v)^2+2*v^2)))+ cont_rate)
        return(z)
      }
      h2_vect <- seq(0, 12, 0.1) # h2 values.

      a2 <- rep(a2_vect, each = length(h2_vect) * N)
      h2 <- rep(h2_vect, each = N, times = length(a2_vect))

    } else if (model == "04_Binomial"){ # BINOMIAL MODEL
      u <- 0.2 # theta parameter

      # NULL MODEL DEFINITION
      hypvector <- function(v){
        z <- c(((1-v)^3),
               (3*v*(1-v)^2),
               (3*v^2*(1-v)),
               (v^3))
        return(z)
      }
      # CONTAMINATED MODEL DEFINITION
      cont_hypvector <- function(v){
        z <- c((1-cont_rate) * ((1-v)^3),
               (1-cont_rate) * (3*v*(1-v)^2),
               (1-cont_rate) * (3*v^2*(1-v)),
               (1-cont_rate) * (v^3) + cont_rate)
        return(z)
      }
      h2_vect <- seq(0, 10, 0.1) # h2 values.

      a2 <- rep(a2_vect, each = length(h2_vect) * N)
      h2 <- rep(h2_vect, each = N, times = length(a2_vect))
    }

  for (n in c(15, 30, 60, 120, 240)){
    for (cont_rate in c(0, epsilon/sqrt(n))){
      print(c(n, cont_rate))

      # Data Sample Generation (Frequencies & Relative Frequencies)
      set.seed(seed = 1)
      # The sample
      mult_obs <- if(cont_rate == 0) t(rmultinom(n=N
                                                 , size = n
                                                 , prob = hypvector(v=u)
      )
      ) else t(rmultinom(n=N
                         , size = n
                         , prob = cont_hypvector(v=u)
      )
      )

      relfreq <- setNames(as.data.table(mult_obs/n)
                          , c("phat_1","phat_2","phat_3","phat_4")) # Each line represents a vector of relative frequencies.
      #---------------------------------------------------------

      # Data Table Generation
      MDE_Data <- data.table(n, N, cont_rate, u, a2, h2, relfreq) # relfreq table is repeated as many times as the length of a2 or h2
```

```
     #-------------------------------------------------------

     MDE_Data[, MDE := future_apply(MDE_Data, 1, function(i){
       optimise(
         P_BHHJ_Divergence_Function
         ,interval = c(0,1)
         ,alpha = i[["a2"]]
         ,w = i[["h2"]]
         ,probvector = c(i[["phat_1"]]
                         ,i[["phat_2"]]
                         ,i[["phat_3"]]
                         ,i[["phat_4"]]))$minimum
     })][, Squared_Error := (u - MDE)^2]

     fwrite(MDE_Data, paste(model
                            , "MDE"
                            , "Simulations", N
                            , "SampleSize", n
                            , "ContRate", cont_rate
                            , "ThetaParamet", u
                            , ".csv"
                            , sep = "_")
     )

   }
 }
 # TEST STATISTIC CALCULATION PROCEDURE
 #-------------------------------------

h1_vect <- c(1, 5, 15, 40)
a1_vect <- c(0.01, 0.05, seq(0.1, 1, 0.1))

for (n in c(15, 30, 60, 120, 240)){
   for (cont_rate in c(0, epsilon/sqrt(n))){
     print(c(n, cont_rate))
     if (model == "01_Geometric"){ # GEOMETRIC MODEL
       MDE_Data <- fread(paste(model
                               , "MDE"
                               , "Simulations", N
                               , "SampleSize", n
                               , "ContRate", cont_rate
                               , "ThetaParamet", 0.9
                               , ".csv"
                               , sep = "_")
                         )

     } else if (model == "02_Poisson"){ # POISSON MODEL
       MDE_Data <- fread(paste(model
                               , "MDE"
                               , "Simulations", N
                               , "SampleSize", n
                               , "ContRate", cont_rate
                               , "ThetaParamet", 0.5
                               , ".csv"
                               , sep = "_")
                         )

     } else if (model == "03_Negative_Binom"){ # NEGATIVE BINOMIAL MODEL
       MDE_Data <- fread(paste(model
                               , "MDE"
                               , "Simulations", N
                               , "SampleSize", n
                               , "ContRate", cont_rate
                               , "ThetaParamet", 0.1
                               , ".csv"
                               , sep = "_")
                         )

     } else if (model == "04_Binomial"){ # BINOMIAL MODEL
       MDE_Data <- fread(paste(model
                               , "MDE"
                               , "Simulations", N
```

```
                             , "SampleSize", n
                             , "ContRate", cont_rate
                             , "ThetaParamet", 0.2
                             , ".csv"
                             , sep = "_")
                       )

    }
    u <- MDE_Data[1, u]

    # Extract only the rows for which h2 is in the set of h1_vect:
    MDE_Data_extract <- MDE_Data[h2 %in% h1_vect]

    # count the rows of the data.table in which we are going to work
    row_count <- nrow(MDE_Data_extract)

    # Test Statistic Data Table Generation
    # create a table where the MDE for a2 and h2 is repeated
    # as many times as a1_vect times h1_vect
    a1 <- rep(a1_vect, each = length(h1_vect) * row_count)
    h1 <- rep(h1_vect, each = row_count, times = length(a1_vect))

    # Keep track of the calculations through the index.
    index <- 1:length(a1)

    # Data table Generation.
    Test_Stat_Data <- data.table(index, a1, h1, MDE_Data_extract)

    rm(list = c("MDE_Data"
                ,"MDE_Data_extract")
       )

    # Definition of the number of rows for which the slice is going to be implemented
    mslice <- nrow(Test_Stat_Data)/10

    # Calculation of the Test Statistic and the C parameter for each block of data
    for (mi in (1:10)){
      Test_Stat_Data[
        (((mi-1)*mslice)+1) : (mi * mslice)
        , c_param := future_apply(
          Test_Stat_Data[
            (((mi-1)*mslice)+1) : (mi * mslice)]
          , 1
          , function(i){
            0.5 * ((max(hypvector(v = i[["MDE"]]))) ^ i[["a1"]] +
                     (min(hypvector(v = i[["MDE"]]))) ^ i[["a1"]])
          }
        )
      ][
        (((mi-1)*mslice)+1) : (mi * mslice)
        , TestStat := future_apply(
          Test_Stat_Data[
            (((mi-1)*mslice)+1) : (mi * mslice)]
          , 1
          , function(i){
            P_BHHJ_Test_Statistic(u1 = i[["MDE"]]
                                  , alpha1 = i[["a1"]]
                                  , w1 = i[["h1"]]
                                  , probvector1 = c(i[["phat_1"]]
                                                    ,i[["phat_2"]]
                                                    ,i[["phat_3"]]
                                                    ,i[["phat_4"]])
                                  , n = i[["n"]])
          }
        )
      ]
      print(mi)
    }

    fwrite(Test_Stat_Data, paste(model
                          , "TestStatData"
```

```
                              , "Simulations", N
                              , "SampleSize", n
                              , "ContRate", cont_rate
                              , "ThetaParamet", u
                              , ".csv"
                              , sep = "_")
             )
      rm(list = c("Test_Stat_Data"))
    }
  }
}
```

C.5.1 Efficiency and robustness examination

```
library(ggplot2)
library(viridis)
library(ggsci)
library(dplyr)
library(latex2exp)
library(xlsx)
library(rJava)
library(data.table)
library(future.apply)

# Data uploading
# ------------------------ No Contamination Data ------------------------
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_15_ContRate_0_ThetaParamet_0.9_.csv")
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_30_ContRate_0_ThetaParamet_0.9_.csv")
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_60_ContRate_0_ThetaParamet_0.9_.csv")
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_120_ContRate_0_ThetaParamet_0.9_.csv")
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_240_ContRate_0_ThetaParamet_0.9_.csv")

# -------------------------- Contaminated Data ------------------------
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_15_ContRate_0.1_ThetaParamet_0.9_.csv")
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_30_ContRate_0.0707106781186548_ThetaParamet_0.9_.csv")
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_60_ContRate_0.05_ThetaParamet_0.9_.csv")
# MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_120_ContRate_0.0353553390593274_ThetaParamet_0.9_.csv")
 MDE_Data_DF <- fread("01_Geometric_MDE_Simulations_10000_SampleSize_240_ContRate_0.025_ThetaParamet_0.9_.csv")

# MSE calculation
df_results <- MDE_Data_DF %>%
  group_by(n, u, cont_rate, a2, h2) %>%
  summarise(samples = max(N)
            , avg_MDE = mean(MDE)
            , MSE = mean(Squared_Error)
            , .groups = 'drop')


sample_size <- min(df_results[["n"]])
cont_rate <- min(df_results[["cont_rate"]])
samples <- max(df_results[["samples"]])
theta <- max(df_results[["u"]])

df_results[["MSE"]] <- sqrt(df_results[["MSE"]])

mmax <- max(df_results[["MSE"]])
mmin <- min(df_results[["MSE"]])


```

```r
mydata <- df_results

# get the line for which the minimum is achieved!
minMSEh2 <- mydata %>%
  group_by(a2) %>%
  filter(MSE == min(MSE)) %>%
  select(h2)

minMSEh2[["h2"]]


h2_a2_MSE_data <- mydata %>%
  group_by(a2) %>%
  filter(h2 %in% c(1, minMSEh2[["h2"]]))

fwrite(h2_a2_MSE_data
       ,file = paste("Geometric_MSE"
                     ,"h2_values_min_occurrence"
                     ,"SampleSize", sample_size
                     ,"ContRate", cont_rate
                     ,"Samples", samples
                     ,"theta", theta
                     ,".csv"
                     , sep = "_")
       )

# Zero frequency cells, percentage occurrence calculation.
zero_cell <- MDE_Data_DF %>%
  group_by(n) %>%
  filter(a2 == 0.01 & h2 == 1) %>%
  select(n, phat_1, phat_2, phat_3, phat_4) %>%
  summarise(p1 = sum(phat_1==0) / nrow(MDE_Data_DF %>%
                                         filter(a2 == 0.01 & h2 == 1))
            ,p2 = sum(phat_2==0) / nrow(MDE_Data_DF %>%
                                          filter(a2 == 0.01 & h2 == 1))
            ,p3 = sum(phat_3==0) / nrow(MDE_Data_DF %>%
                                          filter(a2 == 0.01 & h2 == 1))
            ,p4 = sum(phat_4==0) / nrow(MDE_Data_DF %>%
                                          filter(a2 == 0.01 & h2 == 1))
            ,.groups = 'drop')

zero_cell

fwrite(zero_cell
       ,file = paste("Geometric"
                     ,"zero_cell_occurrence"
                     ,"SampleSize", sample_size
                     ,"ContRate", cont_rate
                     ,"Samples", samples
                     ,"theta", theta
                     ,".csv"
                     , sep = "_")
)

# Graph construction
pdf(file = paste("Geometric_MSE"
                 , "SampleSize", sample_size
                 , "ContRate", cont_rate
                 , "Samples", samples
                 , "theta", theta
                 , ".pdf", sep = "_")
    , width = 10, height = 6)

ggplot(data = mydata,aes(x=h2
                         , y=MSE
                         , group = a2
                         , color = factor(a2)
                         )) +
  geom_vline(xintercept = 1, color = "grey70", linetype = 2) +
  geom_line(alpha = 1, size = 0.2) +
  theme_light() +
  theme(panel.grid = element_line(color = "gray94")
```

```
        ,panel.grid.minor = element_line("gray98")
        #,legend.position = "none"
  ) +
  # scale_y_continuous(minor_breaks = seq(0, mmax, 0.001), breaks = seq(0, mmax, 0.0025), limits = c(mmin, mmax)) + # For the No contamination case
  scale_y_continuous(minor_breaks = seq(0, mmax, 0.0075), breaks = seq(0, mmax, 0.015), limits = c(mmin, mmax)) + # For the contamination case
  scale_x_continuous(minor_breaks = seq(0, 80, 0.5), breaks = seq(0, 80, 2), limits = c(0, 80)) +
  #scale_color_grey()
  #scale_color_npg()
  scale_color_viridis(discrete = TRUE, option = "turbo")+
  # scale_color_manual(values = c("azure4"
  #                     ,"aquamarine2"
  #                     ,"cadetblue3"
  #                     ,"chocolate2"
  #                     ,"blue"
  #                     ,"brown"
  #                     ,"yellow"
  #                     ,"purple"
  #                     ,"chartreuse"
  #                     ,"magenta"
  #                     ,"pink"
  #                     ,"red"
  #                     ,"black"))
  labs(x = TeX("$h_2$"), y = TeX("RMSE"), colour = TeX("$\\alpha_2$")) + # colour refers to the title of the legend
  theme(legend.title.align = 0.5) # to align in the center the title of the legned

dev.off()
```

C.5.2 Size examination

```
library(dplyr)
library(plotly)
library(viridis)
library(latex2exp)
library(xlsx)
library(rJava)
library(data.table)

# Data uploading
# ---------------------------- NO CONTAMINATION -------------------------
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_15_ContRate_0_ThetaParamet_0.9_.csv")
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_30_ContRate_0_ThetaParamet_0.9_.csv")
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_60_ContRate_0_ThetaParamet_0.9_.csv")
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_120_ContRate_0_ThetaParamet_0.9_.csv")
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_240_ContRate_0_ThetaParamet_0.9_.csv")

# ---------------------------- WITH CONTAMINATION ----------------------
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_15_ContRate_0.1_ThetaParamet_0.9_.csv")
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_30_ContRate_0.0707106781186548_ThetaParamet_0.9_.csv")
 TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_60_ContRate_0.05_ThetaParamet_0.9_.csv")
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_120_ContRate_0.0353553390593274_ThetaParamet_0.9_.csv")
# TestStat_Data_prep_DF <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_240_ContRate_0.025_ThetaParamet_0.9_.csv")

TestStat_Data_prep_DF

n <- TestStat_Data_prep_DF %>%
```

```r
  summarise(min(n))

cont_rate <- TestStat_Data_prep_DF %>%
  summarise(min(cont_rate))

mydata <- TestStat_Data_prep_DF %>%
  #select(-one_of(extra_cols)) %>%
  mutate(H0Rejection = if_else(TestStat >= c_param * qchisq(p=0.05
                                                             , df = 2
                                                             , lower.tail = FALSE)
                                 , 1, 0)) %>%
  group_by(a1, a2, h1, h2) %>%
  summarize(RejectionTimes = sum(H0Rejection)
            , RejectionPercentage = (sum(H0Rejection)/10000)*100)


if (cont_rate == 0){
  pdf(file =  paste("TestStat_Size_No_Cont_Geometric_NomLev_0.05_n"
                    , n, "N_10000.pdf"
                    , sep = "_"),
      width = 10, height = 6)
} else {
  pdf(file =  paste("TestStat_Size_With_Cont_Geometric_NomLev_0.05_n"
                    , n
                    , "N_10000.pdf"
                    , sep = "_"),
      width = 10, height = 6)
}

ggplot(mydata
       , aes(x = factor(a1)
             , y = factor(a2)
             , fill = cut(RejectionPercentage
                          , c(-Inf ,0, 3.57, 6.95, 10, 20, 30, 40, Inf)))) +
  geom_tile(size = 0.2, color = "white") +  # the size and the color refer to the line between the tiles
  #geom_text(aes(label = RejectionPercentage), color = "black", size = 2)+
  #coord_fixed()+
  facet_grid(h2~h1, labeller = label_both) + # lines ~ columns
  theme_light() +
  theme(panel.grid = element_line(color = "gray94")
        ,panel.grid.minor = element_line("gray98")
        #,legend.position = "none"
  ) +
  #scale_y_continuous(breaks = c(0.01, 0.05, seq(0.1, 1, 0.1)), limits = c(0, 1.05)) +
  #scale_x_continuous(breaks = c(0.01, 0.05, seq(0.1, 1, 0.1)), limits = c(0, 1.05)) +
  #scale_fill_viridis(discrete = TRUE, option = "inferno", direction = -1) + # direction = -1 changes the direction of the color scale
  scale_fill_manual(values = c(#"lightcyan1"
                                "lightskyblue2"
                                ,"lightseagreen"
                                ,"lightpink"
                                ,"palevioletred1"
                                ,"violetred2"
                                ,"violetred3"
                                ,"violetred4"))+
  labs(x = TeX("$\\alpha_1$"), y = TeX("$\\alpha_2$"), fill = TeX("% Size")) + # colour refers to the title of the legend
  theme(legend.title.align = 0.0 ) + # to align in the center the title of the legned
  theme(legend.key.size = unit(0.4, 'cm'))+ # change legend key size (the size of the tiles in the legend)
  theme(legend.title = element_text(size = 9)) + # change the legend title fond size
  theme(legend.text = element_text(size = 8)) + # change the legend text fond size
  theme(axis.text = element_text(size = 6)) + # change the fond size of the axes
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) + # rotation of the x axis points
  theme(strip.text = element_text(size = 8, color = "gray30")) + # change the font size and the color of the facets
```

```
  theme(strip.background =element_rect(fill="gray95")) + # change the background color of the facets
  theme(panel.border = element_rect(color = "gray93")) # change the color of the outer border of the graphs


dev.off()
```

C.5.3 Bootstrap p−value

```
library(tibble)
library(dplyr)
library(data.table)
library(future.apply)
plan(multisession)

hypvector <- function(v){
  z <- c((v),
         (v*(1-v)),
         (v*(1-v)^2),
         ((1-v)^3))
  return(z)
}

# Penalized BHHJ divergence
P_BHHJ_Divergence_Function <- function(u, alpha, probvector, w) {
  boolvect <- probvector > 0
  u1 <- hypvector(v = u)[boolvect]
  u2 <- hypvector(v = u)[!boolvect]
  probvector1 <- probvector[boolvect]
  z <- sum(u1 ^ (1 + alpha)) -
    (1 + 1 / alpha) * sum(probvector1 * u1 ^ alpha) +
    (1 / alpha) * sum(probvector1 ^ (1 + alpha)) +
    w * sum(u2 ^ (1 + alpha))
  return(z)
}

# Modified BHHJ Test Statistic
P_BHHJ_Test_Statistic <- function(u1, alpha1, probvector1, w1, n) {
  z <-
    ((2 * n) / (1 + alpha1)) * P_BHHJ_Divergence_Function(
      u = u1,
      alpha = alpha1,
      probvector = probvector1,
      w = w1
    )
  return(z)
}

a1_vect <- c(0.01, 0.1, 0.3, 0.6)
a2_vect <- c(0.01, 0.1, 0.3, 0.6)
h1_vect <- c(1, 5, 15)
h2_vect <- c(1, 5, 15)

trial_table <- fread("01_Geometric_TestStatData_Simulations_10000_SampleSize_240_ContRate_0.025_ThetaParamet_0.9_.csv")

# Retain only the important columns
trial_table_1 <- trial_table[ , c("phat_1"
                                  , "phat_2"
                                  , "phat_3"
                                  , "phat_4"
                                  , "Squared_Error"
                                  , "c_param") := NULL][   a1 %in% a1_vect
                                                           & a2 %in% a2_vect
                                                           & h1 %in% h1_vect
                                                           & h2 %in% h2_vect, ]

rm(list = "trial_table")
```

```

# Bootstrap samples
B <- 200

for (i in a1_vect){
  for (j in a2_vect){
    for(k in h1_vect){
      for(l in h2_vect){

        test_1 <- trial_table_1[a1 == i
                                & a2 == j
                                & h1 == k
                                & h2 == l, ]

        # Repeat each line of a data table a number of times
        test_1 <- test_1[rep(1:.N, each = B)]

        # Create the random vector for each line
        boot_rel_freq <- setNames(
          as.data.table(
            t(future_apply(
              test_1[]
              , 1
              , function(i) {
                rmultinom(n = 1
                          , size = i[["n"]]
                          , prob = hypvector(i[["MDE"]])
                ) / i[["n"]]
              }
              , future.seed = TRUE
            )
            )
          )
          ,
          c("b_phat_1", "b_phat_2", "b_phat_3", "b_phat_4")
        )


        test_1 <- cbind(
          test_1
          , boot_rel_freq
        )

        rm(list = "boot_rel_freq")

        # Calculate the bootstrap minimum divergence estimator
        test_1[, boot_MDE := future_apply(
          test_1
          ,1
          ,function(i){
            optimise(
              P_BHHJ_Divergence_Function
              ,interval = c(0,1)
              ,alpha = i[["a2"]]
              ,w = i[["h2"]]
              ,probvector = c(i[["b_phat_1"]]
                              ,i[["b_phat_2"]]
                              ,i[["b_phat_3"]]
                              ,i[["b_phat_4"]]))$minimum
          }
        )
        ]

        # Calculate the bootstrap P_BHHJ_Test

        test_1[, boot_TS := future_apply(
          test_1
          ,1
          ,function(i){
            P_BHHJ_Test_Statistic(u1 = i[["boot_MDE"]]
                                  ,alpha1 = i[["a1"]]
                                  ,w1 = i[["h1"]]
```

```
                                   ,probvector1 = c(i[["b_phat_1"]]
                                                    ,i[["b_phat_2"]]
                                                    ,i[["b_phat_3"]]
                                                    ,i[["b_phat_4"]])
                                   ,n = i[["n"]]
            )
          }
        )
        ]

        # Check if boot_TS >= Observed TS (TestStat)
        test_1[, check_TS := future_apply(test_1
                                           , 1
                                           , function(i) {
                                             i[["boot_TS"]] >= i[["TestStat"]]
                                           })]

        # Calculate the bootstrap p_value
        test_1[, boot_p_value := sum(check_TS)/.N, by = index]

        # Create a table with the index and the boot p_value as columns (p_value by index)
        boot_pvalue <- as.data.table(unique(test_1[, .(index, boot_p_value)], by = "index"))

        # Append the boot p_value to the initial table
        vv <- as.vector(boot_pvalue[,index])

        bpv <- as.vector(boot_pvalue[,boot_p_value])

        trial_table_1[index %in% vv
                      , boot_pvalue := bpv]

        rm(list = c("test_1", "boot_pvalue"))

        print(c(i,j,k,l))
      }
    }
  }
 )
}

fwrite(trial_table_1, file = paste(
  "01_Geometric_boot_pvalue"
  , "u", min(trial_table_1[,u])
  , "n", min(trial_table_1[,n])
  , "N", min(trial_table_1[,N])
  , "B", B
  , "cont_rate", min(trial_table_1[,cont_rate])
  , ".csv"
  , sep = "_"
))
```

C.6 Frequency distribution examination

```
library(dplyr)
library(data.table)
library(future.apply)
plan(multisession)


#                  Count Data Model Parameter Estimation
#                          Poisson Distribution

hypvector <- function(p, x){
  z <- c(
    ppois(q = x[1], lambda = p)
    ,ppois(q = x[2], lambda = p) - ppois(q = x[1], lambda = p)
```

```r
    ,ppois(q = x[3], lambda = p) - ppois(q = x[2], lambda = p)
    ,1 - ppois(q = x[3], lambda = p)

  )
  return(z)
}

# ---      Modified BHHJ divergence     ---
P_BHHJ_Divergence_Function <- function(u, y, alpha, probvector, w) {
  boolvect <- probvector > 0
  u1 <- hypvector(p=u, x=y)[boolvect]
  u2 <- hypvector(p=u, x=y)[!boolvect]
  probvector1 <- probvector[boolvect]
  z <- sum(u1 ^ (1 + alpha)) -
    (1 + 1 / alpha) * sum(probvector1 * u1 ^ alpha) +
    (1 / alpha) * sum(probvector1 ^ (1 + alpha)) +
   w * sum(u2 ^ (1 + alpha))
  return(z)
}

# ---     Modified BHHJ Test Statistic    ---
P_BHHJ_Test_Statistic <- function(u1, y1, alpha1, probvector1, w1, n) {
  z <-
    ((2 * n) / (1 + alpha1)) * P_BHHJ_Divergence_Function(
      u = u1,
      y = y1,
      alpha = alpha1,
      probvector = probvector1,
     w = w1
    )
  return(z)
}

# ---      Cressie-Read divergence     ---
CR_Divergence_Function <- function(u, y, lamda, probvector) {
  if (lamda == 0) {
    vect <- probvector * log(probvector)
    vect[vect == 'NaN'] <- 0
    z <- sum(vect) - sum(probvector * log(hypvector(p=u, x=y)))
  } else if (lamda == -1) {
    z <- sum(hypvector(p=u, x=y)
             * (log(hypvector(p=u, x=y)) - log(probvector)))
  } else {
    z <- (1 / (lamda * (lamda + 1))) * (sum((
      probvector ^ (lamda + 1)) / (hypvector(p=u, x=y)) ^ lamda) - 1)
  }
  return(z)
}

# ---   Cressie-Read Test Statistic   ---
CR_Test_Statistic <- function(u1, y1, lamda1, probvector1, n) {
  if (lamda1 == 0) {
    z <- 2 * n * CR_Divergence_Function(u = u1, y = y1, lamda = lamda1, probvector = probvector1)
  } else if (lamda1 == -1) {
    z <- 2 * n * CR_Divergence_Function(u = u1, y = y1, lamda = lamda1, probvector = probvector1)
  } else {
    z <- 2 * n * CR_Divergence_Function(u = u1, y = y1, lamda = lamda1, probvector = probvector1)
  }
  return(z)
}


# ---------------------- Data Table Generation -----------------------

N <- 10000 # Repetitions of the Experiment
n <- 15    # Size of the sample
u <- 1000  # Poisson rate

mmin <- rep(NA, times = N)
```

```r
mmax <- rep(NA, times = N)
mstep <- rep(NA, times = N)

phat_1 <- rep(NA, times = N)
phat_2 <- rep(NA, times = N)
phat_3 <- rep(NA, times = N)
phat_4 <- rep(NA, times = N)


for (i in 1 : N){
  set.seed(seed = i)
  obs <- rpois(n = n, lambda = u)
  mmin[i] <- min(obs)
  mmax[i] <- max(obs)
  mstep[i] <- (max(obs) - min(obs))/4
  phat_1[i] <- sum(obs <= min(obs) + 1 * (max(obs) - min(obs))/4)  / length(obs)
  phat_2[i] <-
    (sum(obs <= min(obs) +  2 * (max(obs) - min(obs)) / 4)
     - sum(obs <= min(obs) +  1 * (max(obs) - min(obs)) /
             4)) / length(obs)
  phat_3[i] <-
    (sum(obs <= min(obs) +  3 * (max(obs) - min(obs)) / 4)
     - sum(obs <= min(obs) +  2 * (max(obs) - min(obs)) /
             4))  / length(obs)
  phat_4[i] <- sum(obs > min(obs) +  3 * (max(obs) - min(obs))/4)  / length(obs)
}

Pois_Data_Table <- data.table(index = 1:N
                              , mmin
                              , mmax
                              , mstep
                              , phat_1
                              , phat_2
                              , phat_3
                              , phat_4
                              , u)

Pois_Data_Table[, sum(phat_1
                      , phat_2
                      , phat_3
                      , phat_4) , by = index]



#    Parameters Definition for the composition of the Data Table
# ------------------------------------------------------------
# The h tuning parameter may have no impact on the estimation and the test statistic, since no empty cell may occur!.
# Before proceed further we should check the results regarding the empty cells.

zero_cell <- Pois_Data_Table[,
                             .(
                               phat_1
                               , phat_2
                               , phat_3
                               , phat_4
                               )
                             ][,
                               .(
                                 sum(phat_1 == 0)/.N
                                 , sum(phat_2 == 0)/.N
                                 , sum(phat_3 == 0)/.N
                                 , sum(phat_4 == 0)/.N
                               )
                             ]
zero_cell

# Minimum Modified BHHJ Divergence Estimator
h2_vect <- seq(0, 10, 1) # h2 values.
a2_vect <- c(0.01, 0.1, 0.25, 0.50, 0.75) # a2 values.

a2 <- rep(a2_vect, each = length(h2_vect) * N)
```

```r
h2 <- rep(h2_vect, each = N, times = length(a2_vect))

MDE_Data <- data.table(a2, h2, n, Pois_Data_Table)

MDE_Data[, MDE := future_apply(MDE_Data, 1, function(i){
  optimise(
    P_BHHJ_Divergence_Function
    ,interval = c(800,1200)
    ,y = c(i[["mmin"]] + i[["mstep"]]
           , i[["mmin"]] + 2 * i[["mstep"]]
           , i[["mmin"]] + 3 * i[["mstep"]])
    ,alpha = i[["a2"]]
    ,w = i[["h2"]]
    ,probvector = c(i[["phat_1"]]
                    ,i[["phat_2"]]
                    ,i[["phat_3"]]
                    ,i[["phat_4"]]))$minimum
})][, Squared_Error := (u - MDE)^2]


fwrite(x = zero_cell
       , file = "Poisson_Model_Zero_Count.csv")

fwrite(x = MDE_Data
       , file = "Poisson_Model_MDE.csv")


# Minimum CR Divergence Estimator
l <- c(rep(1, times = N), rep(0, times = N))

MDE_Data_CR <- data.table(l, n, Pois_Data_Table)

rm(list = "Pois_Data_Table")

MDE_Data_CR[, MDE := future_apply(MDE_Data_CR, 1, function(i){
  optimise(
    CR_Divergence_Function
    ,interval = c(500,2000)
    ,y = c(i[["mmin"]] + i[["mstep"]]
           , i[["mmin"]] + 2 * i[["mstep"]]
           , i[["mmin"]] + 3 * i[["mstep"]])
    ,lamda = i[["l"]]
    ,probvector = c(i[["phat_1"]]
                    ,i[["phat_2"]]
                    ,i[["phat_3"]]
                    ,i[["phat_4"]]))$minimum
})][, Squared_Error := (u - MDE)^2]

fwrite(x = MDE_Data_CR
       , file = "Poisson_Model_MDE_CR.csv")

# Test Statistic Calculation
#---------------------------------------------

# Modified BHHJ Test Statistic
MDE_Data <- fread("Poisson_Model_MDE.csv")

MDE_Data <- MDE_Data[h2 == 1]

# count the rows of the data.table in which we are going to work
row_count <- nrow(MDE_Data)


h1_vect <- 1 #seq(0, 10, 1) # h1 values.
a1_vect <- c(0.01, 0.1, 0.25, 0.50, 0.75) # a1 values.

# Test Statistic Data Table Generation
# create a table where the MDE for a2 and h2 is repeated
# as many times as a1_vect times h1_vect
a1 <- rep(a1_vect, each = length(h1_vect) * row_count)
h1 <- rep(h1_vect, each = row_count, times = length(a1_vect))

```

```
Test_Stat_Data <- data.table(a1,h1,MDE_Data)

rm(list = "MDE_Data")

# Definition of the number of rows for which the slice is going to be implemented
mslice <- nrow(Test_Stat_Data)/1

for (mi in (1:1)){
  Test_Stat_Data[
    (((mi-1)*mslice)+1) : (mi * mslice)
    , c_param := future_apply(Test_Stat_Data[
      (((mi-1)*mslice)+1) : (mi * mslice)
    ], 1, function(i) {
    0.5 * ((max(hypvector(p = i[["MDE"]]
                          , x =  c(i[["mmin"]] + i[["mstep"]]
                                   , i[["mmin"]] + 2 * i[["mstep"]]
                                   , i[["mmin"]] + 3 * i[["mstep"]])))) ^ i[["a1"]]
                                       +
             (min(hypvector(p = i[["MDE"]]
                            , x =  c(i[["mmin"]] + i[["mstep"]]
                                     , i[["mmin"]] + 2 * i[["mstep"]]
                                     , i[["mmin"]] + 3 * i[["mstep"]])))) ^ i[["a1"
                                         ]])
  })]

  Test_Stat_Data[
    (((mi-1)*mslice)+1) : (mi * mslice)
    , TestStat := future_apply(Test_Stat_Data[
      (((mi-1)*mslice)+1) : (mi * mslice)
    ], 1, function(i) {
    P_BHHJ_Test_Statistic(u1 = i[["MDE"]]
                          , alpha1 = i[["a1"]]
                          , y1 = c(i[["mmin"]] + i[["mstep"]]
                                   , i[["mmin"]] + 2 * i[["mstep"]]
                                   , i[["mmin"]] + 3 * i[["mstep"]])
                          , w1 = i[["h1"]]
                          , probvector1 = c(i[["phat_1"]]
                                            ,i[["phat_2"]]
                                            ,i[["phat_3"]]
                                            ,i[["phat_4"]])
                          , n = i[["n"]])
  })]

}

fwrite(x = Test_Stat_Data,
       file = "Poisson_Model_TS.csv")

# CR Test Statistic
MDE_Data_CR <- fread("Poisson_Model_MDE_CR.csv")

# count the rows of the data.table in which we are going to work
row_count <- nrow(MDE_Data_CR)

Test_Stat_Data_CR <- data.table(MDE_Data_CR)

rm(list = "MDE_Data_CR")

# Definition of the number of rows for which the slice is going to be implemented
mslice <- nrow(Test_Stat_Data_CR)/1

for (mi in (1:1)){
  Test_Stat_Data_CR[
    (((mi-1)*mslice)+1) : (mi * mslice)
    , TestStat := future_apply(Test_Stat_Data_CR[
      (((mi-1)*mslice)+1) : (mi * mslice)
    ], 1, function(i) {
      CR_Test_Statistic(u1 = i[["MDE"]]
                          , lamda1 = i[["l"]]
                          , y1 = c(i[["mmin"]] + i[["mstep"]]
                                   , i[["mmin"]] + 2 * i[["mstep"]]
                                   , i[["mmin"]] + 3 * i[["mstep"]])
```

```r
                          , probvector1 = c(i[["phat_1"]]
                                            ,i[["phat_2"]]
                                            ,i[["phat_3"]]
                                            ,i[["phat_4"]])
                          , n = i[["n"]])
  })]

}

fwrite(x = Test_Stat_Data_CR,
       file = "Poisson_Model_TS_CR.csv")

mydata <- Test_Stat_Data_CR %>%
  #select(-one_of(extra_cols)) %>%
  mutate(H0Rejection = if_else(TestStat >= qchisq(p=0.05, df = 2, lower.tail = FALSE), 1, 0)) %>%
  group_by(1) %>%
  summarize(RejectionTimes = sum(H0Rejection), RejectionPercentage = (sum(H0Rejection)/10000)*100)

fwrite(x = mydata,
       file = "Poisson_Model_TS_Size_CR.csv")
```

C.7 Severity distribution examination

```r
library(data.table)
library(future.apply)
plan(multisession)
library(eva)

# data range generation
obs <- c(rlnorm(n = 995, meanlog = 6, sdlog = 1)
         , rgpd(n = 5,loc =  15000 , scale = 1200, shape = 1.5))

# number of bins
bins <- ceiling(sqrt(length(obs)))

hypvector <- function(p, x){
  z <- c(
    plnorm(q = x[1], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[2], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[1], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[3], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[2], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[4], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[3], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[5], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[4], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[6], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[5], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[7], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[6], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[8], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[7], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[9], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[8], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[10], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[9], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[11], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[10], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[12], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[11], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[13], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[12], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[14], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[13], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[15], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[14], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[16], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[15], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[17], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[16], meanlog = p[1], sdlog = p[2])
```

```
    ,plnorm(q = x[18], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[17], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[19], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[18], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[20], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[19], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[21], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[20], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[22], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[21], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[23], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[22], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[24], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[23], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[25], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[24], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[26], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[25], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[27], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[26], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[28], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[27], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[29], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[28], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[30], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[29], meanlog = p[1], sdlog = p[2])
    ,plnorm(q = x[31], meanlog = p[1], sdlog = p[2]) - plnorm(q = x[30], meanlog = p[1], sdlog = p[2])
    ,1 - plnorm(q = x[31], meanlog = p[1], sdlog = p[2])

  )
  return(z)
}

# ---                 Penalized BHHJ divergence                  ---
P_BHHJ_Divergence_Function <- function(u, y, alpha, probvector, w) {
  boolvect <- probvector > 0
  u1 <- hypvector(p=u, x=y)[boolvect]
  u2 <- hypvector(p=u, x=y)[!boolvect]
  probvector1 <- probvector[boolvect]
  z <- sum(u1 ^ (1 + alpha)) -
    (1 + 1 / alpha) * sum(probvector1 * u1 ^ alpha) +
    (1 / alpha) * sum(probvector1 ^ (1 + alpha)) +
   w * sum(u2 ^ (1 + alpha))
  return(z)
}

# ---              Modified BHHJ Test Statistic                    ---
P_BHHJ_Test_Statistic <- function(u1, y1, alpha1, probvector1, w1, n) {
  z <-
    ((2 * n) / (1 + alpha1)) * P_BHHJ_Divergence_Function(
      u = u1,
      y = y1,
      alpha = alpha1,
      probvector = probvector1,
     w = w1
    )
  return(z)
}

# ---     Cressie-Read divergence    ---
CR_Divergence_Function <- function(u, y, lamda, probvector) {
  if (lamda == 0) {
    vect <- probvector * log(probvector)
    vect[vect == 'NaN'] <- 0
    z <- sum(vect) - sum(probvector * log(hypvector(p=u, x=y)))
  } else if (lamda == -1) {
    z <- sum(hypvector(p=u, x=y)
              * (log(hypvector(p=u, x=y)) - log(probvector)))
  } else {
    z <- (1 / (lamda * (lamda + 1))) * (sum((
      probvector ^ (lamda + 1)) / (hypvector(p=u, x=y)) ^ lamda) - 1)
```

```r
  }
  return(z)
}

# ---   Cressie-Read Test Statistic   ---
CR_Test_Statistic <- function(u1, y1, lamda1, probvector1, n) {
  if (lamda1 == 0) {
    z <- 2 * n * CR_Divergence_Function(u = u1, y = y1, lamda = lamda1, probvector = probvector1)
  } else if (lamda1 == -1) {
    z <- 2 * n * CR_Divergence_Function(u = u1, y = y1, lamda = lamda1, probvector = probvector1)
  } else {
    z <- 2 * n * CR_Divergence_Function(u = u1, y = y1, lamda = lamda1, probvector = probvector1)
  }
  return(z)
}


# ------------------------------------------------------------------
# ---------------------- Data Table Generation ----------------------

N <- 10^4 # Repetitions of the Experiment
n <- 10^3 # Size of the sample
mu <- 6   # menanlog parameter
sd <- 1   # sdlog parameter

mstep <- rep(NA, times = N)

phat_1 <- rep(NA, times = N)
phat_2 <- rep(NA, times = N)
phat_3 <- rep(NA, times = N)
phat_4 <- rep(NA, times = N)
phat_5 <- rep(NA, times = N)
phat_6 <- rep(NA, times = N)
phat_7 <- rep(NA, times = N)
phat_8 <- rep(NA, times = N)
phat_9 <- rep(NA, times = N)
phat_10 <- rep(NA, times = N)
phat_11 <- rep(NA, times = N)
phat_12 <- rep(NA, times = N)
phat_13 <- rep(NA, times = N)
phat_14 <- rep(NA, times = N)
phat_15 <- rep(NA, times = N)
phat_16 <- rep(NA, times = N)
phat_17 <- rep(NA, times = N)
phat_18 <- rep(NA, times = N)
phat_19 <- rep(NA, times = N)
phat_20 <- rep(NA, times = N)
phat_21 <- rep(NA, times = N)
phat_22 <- rep(NA, times = N)
phat_23 <- rep(NA, times = N)
phat_24 <- rep(NA, times = N)
phat_25 <- rep(NA, times = N)
phat_26 <- rep(NA, times = N)
phat_27 <- rep(NA, times = N)
phat_28 <- rep(NA, times = N)
phat_29 <- rep(NA, times = N)
phat_30 <- rep(NA, times = N)
phat_31 <- rep(NA, times = N)
phat_32 <- rep(NA, times = N)

for (i in 1 : N){
  set.seed(seed = i)
  obs <- c(rlnorm(n = 995, meanlog = mu, sdlog = sd)
           , rgpd(n = 5,loc =  15000 , scale = 1200, shape = 1.5))
  # step for the construction of the bins
  step <- ceiling(quantile(obs, 0.75)[[1]] - quantile(obs, 0.25)[[1]])
  # step vector for the construction of the bins
  mstep[i] <- step
  phat_1[i] <- sum(obs <= step) / n
```

```
  phat_2[i] <- (sum(obs <= 2 * step) - sum(obs <= 1 * step)) / n
  phat_3[i] <- (sum(obs <= 3 * step) - sum(obs <= 2 * step)) / n
  phat_4[i] <- (sum(obs <= 4 * step) - sum(obs <= 3 * step)) / n
  phat_5[i] <- (sum(obs <= 5 * step) - sum(obs <= 4 * step)) / n
  phat_6[i] <- (sum(obs <= 6 * step) - sum(obs <= 5 * step)) / n
  phat_7[i] <- (sum(obs <= 7 * step) - sum(obs <= 6 * step)) / n
  phat_8[i] <- (sum(obs <= 8 * step) - sum(obs <= 7 * step)) / n
  phat_9[i] <- (sum(obs <= 9 * step) - sum(obs <= 8 * step)) / n
  phat_10[i] <- (sum(obs <= 10 * step) - sum(obs <= 9 * step)) / n
  phat_11[i] <- (sum(obs <= 11 * step) - sum(obs <= 10 * step)) / n
  phat_12[i] <- (sum(obs <= 12 * step) - sum(obs <= 11 * step)) / n
  phat_13[i] <- (sum(obs <= 13 * step) - sum(obs <= 12 * step)) / n
  phat_14[i] <- (sum(obs <= 14 * step) - sum(obs <= 13 * step)) / n
  phat_15[i] <- (sum(obs <= 15 * step) - sum(obs <= 14 * step)) / n
  phat_16[i] <- (sum(obs <= 16 * step) - sum(obs <= 15 * step)) / n
  phat_17[i] <- (sum(obs <= 17 * step) - sum(obs <= 16 * step)) / n
  phat_18[i] <- (sum(obs <= 18 * step) - sum(obs <= 17 * step)) / n
  phat_19[i] <- (sum(obs <= 19 * step) - sum(obs <= 18 * step)) / n
  phat_20[i] <- (sum(obs <= 20 * step) - sum(obs <= 19 * step)) / n
  phat_21[i] <- (sum(obs <= 21 * step) - sum(obs <= 20 * step)) / n
  phat_22[i] <- (sum(obs <= 22 * step) - sum(obs <= 21 * step)) / n
  phat_23[i] <- (sum(obs <= 23 * step) - sum(obs <= 22 * step)) / n
  phat_24[i] <- (sum(obs <= 24 * step) - sum(obs <= 23 * step)) / n
  phat_25[i] <- (sum(obs <= 25 * step) - sum(obs <= 24 * step)) / n
  phat_26[i] <- (sum(obs <= 26 * step) - sum(obs <= 25 * step)) / n
  phat_27[i] <- (sum(obs <= 27 * step) - sum(obs <= 26 * step)) / n
  phat_28[i] <- (sum(obs <= 28 * step) - sum(obs <= 27 * step)) / n
  phat_29[i] <- (sum(obs <= 29 * step) - sum(obs <= 28 * step)) / n
  phat_30[i] <- (sum(obs <= 30 * step) - sum(obs <= 29 * step)) / n
  phat_31[i] <- (sum(obs <= 31 * step) - sum(obs <= 30 * step)) / n
  phat_32[i] <- sum(obs > 31 * step) / n
}

LogNor_Data_Table <- data.table(index = 1:N
                               , mu
                               , sd
                               , mstep
                               , phat_1
                               , phat_2
                               , phat_3
                               , phat_4
                               , phat_5
                               , phat_6
                               , phat_7
                               , phat_8
                               , phat_9
                               , phat_10
                               , phat_11
                               , phat_12
                               , phat_13
                               , phat_14
                               , phat_15
                               , phat_16
                               , phat_17
                               , phat_18
                               , phat_19
                               , phat_20
                               , phat_21
                               , phat_22
                               , phat_23
                               , phat_24
                               , phat_25
                               , phat_26
                               , phat_27
                               , phat_28
                               , phat_29
                               , phat_30
                               , phat_31
                               , phat_32
                               )


```

```
LogNor_Data_Table[,
                  sum(c(phat_1
                        , phat_2
                        , phat_3
                        , phat_4
                        , phat_5
                        , phat_6
                        , phat_7
                        , phat_8
                        , phat_9
                        , phat_10
                        , phat_11
                        , phat_12
                        , phat_13
                        , phat_14
                        , phat_15
                        , phat_16
                        , phat_17
                        , phat_18
                        , phat_19
                        , phat_20
                        , phat_21
                        , phat_22
                        , phat_23
                        , phat_24
                        , phat_25
                        , phat_26
                        , phat_27
                        , phat_28
                        , phat_29
                        , phat_30
                        , phat_31
                        , phat_32)),
                  by = index]

zero_cell <- LogNor_Data_Table[,
                              .(phat_1
                                , phat_2
                                , phat_3
                                , phat_4
                                , phat_5
                                , phat_6
                                , phat_7
                                , phat_8
                                , phat_9
                                , phat_10
                                , phat_11
                                , phat_12
                                , phat_13
                                , phat_14
                                , phat_15
                                , phat_16
                                , phat_17
                                , phat_18
                                , phat_19
                                , phat_20
                                , phat_21
                                , phat_22
                                , phat_23
                                , phat_24
                                , phat_25
                                , phat_26
                                , phat_27
                                , phat_28
                                , phat_29
                                , phat_30
                                , phat_31
                                , phat_32)][,
                                           .(
                                            sum(phat_1 == 0) / .N
                                            , sum(phat_2 == 0) / .N
                                            , sum(phat_3 == 0) / .N
```

```
                                            , sum(phat_4 == 0) / .N
                                            , sum(phat_5 == 0) / .N
                                            , sum(phat_6 == 0) / .N
                                            , sum(phat_7 == 0) / .N
                                            , sum(phat_8 == 0) / .N
                                            , sum(phat_9 == 0) / .N
                                            , sum(phat_10 == 0) / .N
                                            , sum(phat_11 == 0) / .N
                                            , sum(phat_12 == 0) / .N
                                            , sum(phat_13 == 0) / .N
                                            , sum(phat_14 == 0) / .N
                                            , sum(phat_15 == 0) / .N
                                            , sum(phat_16 == 0) / .N
                                            , sum(phat_17 == 0) / .N
                                            , sum(phat_18 == 0) / .N
                                            , sum(phat_19 == 0) / .N
                                            , sum(phat_20 == 0) / .N
                                            , sum(phat_21 == 0) / .N
                                            , sum(phat_22 == 0) / .N
                                            , sum(phat_23 == 0) / .N
                                            , sum(phat_24 == 0) / .N
                                            , sum(phat_25 == 0) / .N
                                            , sum(phat_26 == 0) / .N
                                            , sum(phat_27 == 0) / .N
                                            , sum(phat_28 == 0) / .N
                                            , sum(phat_29 == 0) / .N
                                            , sum(phat_30 == 0) / .N
                                            , sum(phat_31 == 0) / .N
                                            , sum(phat_32 == 0) / .N
                                            )
                                          ]
zero_cell

fwrite(x = zero_cell,
       file = "Severity_Model_Zero_Cell.csv")



#        Parameters Definition for the composition of the Data Table
#----------------------------------------------------------------

h2_vect <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10) # h2 values.
a2_vect <- c(0.01, 0.1, 0.25, 0.50, 0.75)      # a2 values.

a2 <- rep(a2_vect, each = length(h2_vect) * N)
h2 <- rep(h2_vect, each = N, times = length(a2_vect))

MDE_Data <- data.table(a2, h2, n, LogNor_Data_Table)

# Minimum Divergence Estimation

# Minimum Modified BHHJ Estimation
MDE_Data[, opt_vect := future_apply(MDE_Data, 1, function(i){
  as.list(optim(par = c(4, 1.5)
        ,P_BHHJ_Divergence_Function
        , y = c(1 * i[["mstep"]]
                , 2 * i[["mstep"]]
                , 3 * i[["mstep"]]
                , 4 * i[["mstep"]]
                , 5 * i[["mstep"]]
                , 6 * i[["mstep"]]
                , 7 * i[["mstep"]]
                , 8 * i[["mstep"]]
                , 9 * i[["mstep"]]
                , 10 * i[["mstep"]]
                , 11 * i[["mstep"]]
                , 12 * i[["mstep"]]
                , 13 * i[["mstep"]]
                , 14 * i[["mstep"]]
                , 15 * i[["mstep"]]
                , 16 * i[["mstep"]]
                , 17 * i[["mstep"]]
```

```
                 , 18 * i[["mstep"]]
                 , 19 * i[["mstep"]]
                 , 20 * i[["mstep"]]
                 , 21 * i[["mstep"]]
                 , 22 * i[["mstep"]]
                 , 23 * i[["mstep"]]
                 , 24 * i[["mstep"]]
                 , 25 * i[["mstep"]]
                 , 26 * i[["mstep"]]
                 , 27 * i[["mstep"]]
                 , 28 * i[["mstep"]]
                 , 29 * i[["mstep"]]
                 , 30 * i[["mstep"]]
                 , 31 * i[["mstep"]]
        )
        , alpha = i[["a2"]]
        , w = i[["h2"]]
        , probvector = c(i[["phat_1"]]
                          , i[["phat_2"]]
                          , i[["phat_3"]]
                          , i[["phat_4"]]
                          , i[["phat_5"]]
                          , i[["phat_6"]]
                          , i[["phat_7"]]
                          , i[["phat_8"]]
                          , i[["phat_9"]]
                          , i[["phat_10"]]
                          , i[["phat_11"]]
                          , i[["phat_12"]]
                          , i[["phat_13"]]
                          , i[["phat_14"]]
                          , i[["phat_15"]]
                          , i[["phat_16"]]
                          , i[["phat_17"]]
                          , i[["phat_18"]]
                          , i[["phat_19"]]
                          , i[["phat_20"]]
                          , i[["phat_21"]]
                          , i[["phat_22"]]
                          , i[["phat_23"]]
                          , i[["phat_24"]]
                          , i[["phat_25"]]
                          , i[["phat_26"]]
                          , i[["phat_27"]]
                          , i[["phat_28"]]
                          , i[["phat_29"]]
                          , i[["phat_30"]]
                          , i[["phat_31"]]
                          , i[["phat_32"]]
        )
        #,lower = c(5, 0.8)
        #,upper = c(7, 1.2)

  )$par)
})]


MDE_Data[, meanlog := future_apply(MDE_Data, 1, function(i){
  i[["opt_vect"]][[1]]
})]

MDE_Data[, sdlog := future_apply(MDE_Data, 1, function(i){
  i[["opt_vect"]][[2]]
})]

MDE_Data[, c("muSE", "sdSE") := .((meanlog - mu)^2
                                  ,(sdlog - sd)^2 )]

MDE_Data[, overallSE := muSE + sdSE]

fwrite(x = MDE_Data[, opt_vect := NULL]
       , file = "Severity_Model_MDE.csv")
```

```

# Minimum CR Estimation
l <- c(rep(1, times = N), rep(0, times = N))

MDE_Data_CR <- data.table(l, n, LogNor_Data_Table)

MDE_Data_CR[, opt_vect := future_apply(MDE_Data_CR, 1, function(i){
  as.list(optim(par = c(4, 1.5)
        ,CR_Divergence_Function
        , y = c(1 * i[["mstep"]]
                , 2 * i[["mstep"]]
                , 3 * i[["mstep"]]
                , 4 * i[["mstep"]]
                , 5 * i[["mstep"]]
                , 6 * i[["mstep"]]
                , 7 * i[["mstep"]]
                , 8 * i[["mstep"]]
                , 9 * i[["mstep"]]
                , 10 * i[["mstep"]]
                , 11 * i[["mstep"]]
                , 12 * i[["mstep"]]
                , 13 * i[["mstep"]]
                , 14 * i[["mstep"]]
                , 15 * i[["mstep"]]
                , 16 * i[["mstep"]]
                , 17 * i[["mstep"]]
                , 18 * i[["mstep"]]
                , 19 * i[["mstep"]]
                , 20 * i[["mstep"]]
                , 21 * i[["mstep"]]
                , 22 * i[["mstep"]]
                , 23 * i[["mstep"]]
                , 24 * i[["mstep"]]
                , 25 * i[["mstep"]]
                , 26 * i[["mstep"]]
                , 27 * i[["mstep"]]
                , 28 * i[["mstep"]]
                , 29 * i[["mstep"]]
                , 30 * i[["mstep"]]
                , 31 * i[["mstep"]]
        )
        , lamda = i[["l"]]
        , probvector = c(i[["phat_1"]]
                         , i[["phat_2"]]
                         , i[["phat_3"]]
                         , i[["phat_4"]]
                         , i[["phat_5"]]
                         , i[["phat_6"]]
                         , i[["phat_7"]]
                         , i[["phat_8"]]
                         , i[["phat_9"]]
                         , i[["phat_10"]]
                         , i[["phat_11"]]
                         , i[["phat_12"]]
                         , i[["phat_13"]]
                         , i[["phat_14"]]
                         , i[["phat_15"]]
                         , i[["phat_16"]]
                         , i[["phat_17"]]
                         , i[["phat_18"]]
                         , i[["phat_19"]]
                         , i[["phat_20"]]
                         , i[["phat_21"]]
                         , i[["phat_22"]]
                         , i[["phat_23"]]
                         , i[["phat_24"]]
                         , i[["phat_25"]]
                         , i[["phat_26"]]
                         , i[["phat_27"]]
                         , i[["phat_28"]]
                         , i[["phat_29"]]
                         , i[["phat_30"]]
```

```
                         , i[["phat_31"]]
                         , i[["phat_32"]]
        )
        #,lower = c(5, 0.8)
        #,upper = c(7, 1.2)

  )$par)
})]


MDE_Data_CR[, meanlog := future_apply(MDE_Data_CR, 1, function(i){
  i[["opt_vect"]][[1]]
})]

MDE_Data_CR[, sdlog := future_apply(MDE_Data_CR, 1, function(i){
  i[["opt_vect"]][[2]]
})]

MDE_Data_CR[, c("muSE", "sdSE") := .((meanlog - mu)^2
                                  ,(sdlog - sd)^2 )]

MDE_Data_CR[, overallSE := muSE + sdSE]

fwrite(x = MDE_Data_CR[, opt_vect := NULL]
       , file = "Severity_Model_MDE_CR.csv")


# Test Statistic Calculation

# Modified BHHJ Test Statistic
MDE_Data <- fread("Severity_Model_MDE.csv")
MDE_Data <- MDE_Data[h2 %in% c(1,5)]

# count the rows of the data.table in which we are going to work
row_count <- nrow(MDE_Data)


h1_vect <- c(1,5) # h1 values.
a1_vect <- c(0.01, 0.1, 0.25, 0.50, 0.75) # a1 values.

# Test Statistic Data Table Generation
# create a table where the MDE for a2 and h2 is repeated
# as many times as a1_vect times h1_vect
a1 <- rep(a1_vect, each = length(h1_vect) * row_count)
h1 <- rep(h1_vect, each = row_count, times = length(a1_vect))

Test_Stat_Data <- data.table(a1,h1,MDE_Data)

rm(list = "MDE_Data")

# Definition of the number of rows for which the slice is going to be implemented
mslice <- nrow(Test_Stat_Data)/100

for (mi in (1:100)){
  print(mi)
  Test_Stat_Data[
    (((mi-1)*mslice)+1) : (mi * mslice)
    , c_param := future_apply(Test_Stat_Data[
      (((mi-1)*mslice)+1) : (mi * mslice)
    ], 1, function(i) {
    0.5 * ((max(hypvector(p = c(i[["meanlog"]], i[["sdlog"]])
                          , x =  c(1 * i[["mstep"]]
                                   , 2 * i[["mstep"]]
                                   , 3 * i[["mstep"]]
                                   , 4 * i[["mstep"]]
                                   , 5 * i[["mstep"]]
                                   , 6 * i[["mstep"]]
                                   , 7 * i[["mstep"]]
                                   , 8 * i[["mstep"]]
                                   , 9 * i[["mstep"]]
                                   , 10 * i[["mstep"]]
                                   , 11 * i[["mstep"]]
```

```
                                 , 12 * i[["mstep"]]
                                 , 13 * i[["mstep"]]
                                 , 14 * i[["mstep"]]
                                 , 15 * i[["mstep"]]
                                 , 16 * i[["mstep"]]
                                 , 17 * i[["mstep"]]
                                 , 18 * i[["mstep"]]
                                 , 19 * i[["mstep"]]
                                 , 20 * i[["mstep"]]
                                 , 21 * i[["mstep"]]
                                 , 22 * i[["mstep"]]
                                 , 23 * i[["mstep"]]
                                 , 24 * i[["mstep"]]
                                 , 25 * i[["mstep"]]
                                 , 26 * i[["mstep"]]
                                 , 27 * i[["mstep"]]
                                 , 28 * i[["mstep"]]
                                 , 29 * i[["mstep"]]
                                 , 30 * i[["mstep"]]
                                 , 31* i[["mstep"]]
                        )))) ^ i[["a1"]] +
           (min(hypvector(p = c(i[["meanlog"]], i[["sdlog"]])
                          , x =  c(1 * i[["mstep"]]
                                   , 2 * i[["mstep"]]
                                   , 3 * i[["mstep"]]
                                   , 4 * i[["mstep"]]
                                   , 5 * i[["mstep"]]
                                   , 6 * i[["mstep"]]
                                   , 7 * i[["mstep"]]
                                   , 8 * i[["mstep"]]
                                   , 9 * i[["mstep"]]
                                   , 10 * i[["mstep"]]
                                   , 11 * i[["mstep"]]
                                   , 12 * i[["mstep"]]
                                   , 13 * i[["mstep"]]
                                   , 14 * i[["mstep"]]
                                   , 15 * i[["mstep"]]
                                   , 16 * i[["mstep"]]
                                   , 17 * i[["mstep"]]
                                   , 18 * i[["mstep"]]
                                   , 19 * i[["mstep"]]
                                   , 20 * i[["mstep"]]
                                   , 21 * i[["mstep"]]
                                   , 22 * i[["mstep"]]
                                   , 23 * i[["mstep"]]
                                   , 24 * i[["mstep"]]
                                   , 25 * i[["mstep"]]
                                   , 26 * i[["mstep"]]
                                   , 27 * i[["mstep"]]
                                   , 28 * i[["mstep"]]
                                   , 29 * i[["mstep"]]
                                   , 30 * i[["mstep"]]
                                   , 31* i[["mstep"]]
                          )))) ^ i[["a1"]])
})]

Test_Stat_Data[
  (((mi-1)*mslice)+1) : (mi * mslice),
  TestStat := future_apply(Test_Stat_Data[
    (((mi-1)*mslice)+1) : (mi * mslice)
  ], 1, function(i) {
  P_BHHJ_Test_Statistic(u1 = c(i[["meanlog"]], i[["sdlog"]])
                        , alpha1 = i[["a1"]]
                        , y1 = c(1 * i[["mstep"]]
                                 , 2 * i[["mstep"]]
                                 , 3 * i[["mstep"]]
                                 , 4 * i[["mstep"]]
                                 , 5 * i[["mstep"]]
                                 , 6 * i[["mstep"]]
                                 , 7 * i[["mstep"]]
                                 , 8 * i[["mstep"]]
                                 , 9 * i[["mstep"]]
```

```
                                    , 10 * i[["mstep"]]
                                    , 11 * i[["mstep"]]
                                    , 12 * i[["mstep"]]
                                    , 13 * i[["mstep"]]
                                    , 14 * i[["mstep"]]
                                    , 15 * i[["mstep"]]
                                    , 16 * i[["mstep"]]
                                    , 17 * i[["mstep"]]
                                    , 18 * i[["mstep"]]
                                    , 19 * i[["mstep"]]
                                    , 20 * i[["mstep"]]
                                    , 21 * i[["mstep"]]
                                    , 22 * i[["mstep"]]
                                    , 23 * i[["mstep"]]
                                    , 24 * i[["mstep"]]
                                    , 25 * i[["mstep"]]
                                    , 26 * i[["mstep"]]
                                    , 27 * i[["mstep"]]
                                    , 28 * i[["mstep"]]
                                    , 29 * i[["mstep"]]
                                    , 30 * i[["mstep"]]
                                    , 31* i[["mstep"]]
                          )
                          , w1 = i[["h1"]]
                          , probvector1 = c(i[["phat_1"]]
                                             , i[["phat_2"]]
                                             , i[["phat_3"]]
                                             , i[["phat_4"]]
                                             , i[["phat_5"]]
                                             , i[["phat_6"]]
                                             , i[["phat_7"]]
                                             , i[["phat_8"]]
                                             , i[["phat_9"]]
                                             , i[["phat_10"]]
                                             , i[["phat_11"]]
                                             , i[["phat_12"]]
                                             , i[["phat_13"]]
                                             , i[["phat_14"]]
                                             , i[["phat_15"]]
                                             , i[["phat_16"]]
                                             , i[["phat_17"]]
                                             , i[["phat_18"]]
                                             , i[["phat_19"]]
                                             , i[["phat_20"]]
                                             , i[["phat_21"]]
                                             , i[["phat_22"]]
                                             , i[["phat_23"]]
                                             , i[["phat_24"]]
                                             , i[["phat_25"]]
                                             , i[["phat_26"]]
                                             , i[["phat_27"]]
                                             , i[["phat_28"]]
                                             , i[["phat_29"]]
                                             , i[["phat_30"]]
                                             , i[["phat_31"]]
                                             , i[["phat_32"]]
                          )
                          , n = i[["n"]])
  })]

}

Test_Stat_Data[index == 1]

fwrite(x = Test_Stat_Data
       , file = "Severity_Model_TS.csv")


# CR Test Statistic
MDE_Data_CR <- fread("Severity_Model_MDE_CR.csv")

# count the rows of the data.table in which we are going to work
```

```r
row_count <- nrow(MDE_Data_CR)

Test_Stat_Data_CR <- data.table(MDE_Data_CR)

rm(list = "MDE_Data_CR")

# Definition of the number of rows for which the slice is going to be implemented
mslice <- nrow(Test_Stat_Data_CR)/1

for (mi in (1:1)){
  print(mi)
  Test_Stat_Data_CR[
    (((mi-1)*mslice)+1) : (mi * mslice),
    TestStat := future_apply(Test_Stat_Data_CR[
      (((mi-1)*mslice)+1) : (mi * mslice)
    ], 1, function(i) {
      CR_Test_Statistic(u1 = c(i[["meanlog"]], i[["sdlog"]])
                         , lamda1 = i[["l"]]
                         , y1 = c(1 * i[["mstep"]]
                                  , 2 * i[["mstep"]]
                                  , 3 * i[["mstep"]]
                                  , 4 * i[["mstep"]]
                                  , 5 * i[["mstep"]]
                                  , 6 * i[["mstep"]]
                                  , 7 * i[["mstep"]]
                                  , 8 * i[["mstep"]]
                                  , 9 * i[["mstep"]]
                                  , 10 * i[["mstep"]]
                                  , 11 * i[["mstep"]]
                                  , 12 * i[["mstep"]]
                                  , 13 * i[["mstep"]]
                                  , 14 * i[["mstep"]]
                                  , 15 * i[["mstep"]]
                                  , 16 * i[["mstep"]]
                                  , 17 * i[["mstep"]]
                                  , 18 * i[["mstep"]]
                                  , 19 * i[["mstep"]]
                                  , 20 * i[["mstep"]]
                                  , 21 * i[["mstep"]]
                                  , 22 * i[["mstep"]]
                                  , 23 * i[["mstep"]]
                                  , 24 * i[["mstep"]]
                                  , 25 * i[["mstep"]]
                                  , 26 * i[["mstep"]]
                                  , 27 * i[["mstep"]]
                                  , 28 * i[["mstep"]]
                                  , 29 * i[["mstep"]]
                                  , 30 * i[["mstep"]]
                                  , 31* i[["mstep"]]
                         )
                         , probvector1 = c(i[["phat_1"]]
                                           , i[["phat_2"]]
                                           , i[["phat_3"]]
                                           , i[["phat_4"]]
                                           , i[["phat_5"]]
                                           , i[["phat_6"]]
                                           , i[["phat_7"]]
                                           , i[["phat_8"]]
                                           , i[["phat_9"]]
                                           , i[["phat_10"]]
                                           , i[["phat_11"]]
                                           , i[["phat_12"]]
                                           , i[["phat_13"]]
                                           , i[["phat_14"]]
                                           , i[["phat_15"]]
                                           , i[["phat_16"]]
                                           , i[["phat_17"]]
                                           , i[["phat_18"]]
                                           , i[["phat_19"]]
                                           , i[["phat_20"]]
                                           , i[["phat_21"]]
                                           , i[["phat_22"]]
```

```
                                            , i[["phat_23"]]
                                            , i[["phat_24"]]
                                            , i[["phat_25"]]
                                            , i[["phat_26"]]
                                            , i[["phat_27"]]
                                            , i[["phat_28"]]
                                            , i[["phat_29"]]
                                            , i[["phat_30"]]
                                            , i[["phat_31"]]
                                            , i[["phat_32"]]
                          )
                          , n = i[["n"]])
  })]
}

fwrite(x = Test_Stat_Data_CR
       , file = "Severity_Model_TS_CR.csv")

mydata <- Test_Stat_Data_CR %>%
  #select(-one_of(extra_cols)) %>%
  mutate(H0Rejection = if_else(TestStat >= qchisq(p=0.05, df = 29, lower.tail = FALSE), 1, 0)) %>%
  group_by(1) %>%
  summarize(RejectionTimes = sum(H0Rejection), RejectionPercentage = (sum(H0Rejection)/10000)*100)

fwrite(x = mydata,
       file = "Severity_Model_TS_Size_CR.csv")
```