Multivariate Analysis Techniques
&
Methods for Missing Data

Master Thesis

Michael Spitieris

University of the Aegean

Department of Mathematics, Division of Statistics & Actuarial-Financial Mathematics

Committee:

Spyridon J. Hatjispyros
Dept. of Mathematics, University of the Aegean

Alex Karagrigoriou, *Thesis Supervisor*
Dept. of Mathematics, University of the Aegean

Stylianos Z. Xanthopoulos
Dept. of Mathematics, University of the Aegean

Acknowledgements

First and foremost, I would like to thank my advisor, Alex Karagrigoriou, for the suggestion of the topic! Without him this thesis would be impossible. I would like to thank him for his patience and his guidance. Working with him has been a great opportunity for expanding my knowledge and learning new things.

Also I would like to thank all the professors who taught me on undergraduate and postgraduate level.

I owe special thanks to my math school teacher Konstantinos Tsoukalas for the valuable advices during the school years.

I want to thank my friends Christos Merkatas and Kostas Kaloudis for their valuable advice during my postgraduate studies.

Finally, words alone cannot express the thanks I owe to my parents Lygeri Pitsiakou and Spiros Spitieris.

Abstract

The purpose of this Thesis is to illustrate Multivariate Analysis Techniques, and specifically intelligent clustering algorithms. We will describe methods for missing data and see how missing values affect the statistical procedures. At the end of this Thesis we will develop two new imputation methods, named Partition Means imputation and Partition Regression imputation.

In Chapter 1 we will give the notation of multivariate data, we will describe multivariate distributions such as multivariate Normal distribution, Wishart distribution and the Hotelling $\mathcal{T}^2$ distribution.

In Chapter 2 we will see three multivariate analysis techniques, Principal Component Analysis (PCA) which used for dimension reduction, Linear Discriminant Analysis (LDA) that is a supervised method used for classification and Clustering which belongs to the family of unsupervised methods.

In Chapter 3 we will briefly describe missing data methods. We will describe the mechanisms that generate missing data, Missing Completely at Random (MCAR), Missing at Random (MAR) and Not Missing at Random (NMAR). Furthermore we will describe the following methods: Complete case Analysis, Weighting Procedures, Imputation Methods and Model-Based methods. Specifically we will focus our interest on the imputation techniques.

In Chapter 4 we will illustrate three intelligent clustering methods which improve the accuracy of the final partition using external sources of information. These methods modify the K-means in a way such that the original algorithm accommodates a set of constraints. COP Kmeans accommodates a set of Hard constraints, SCOP Kmeans accommodates a set of Soft Constraints and KSC algorithm, a modified Kmeans which can deal with missing data. For the purposes of this Thesis, the code for the last algorithm has been developed in R.

In Chapter 5 we will develop two new imputation methods. These new approaches divide the data set into k homogeneous subsets, where each subset is treated as an individual data set. In each subset we will perform mean imputation and regression imputation.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Multivariate Statistical Analysis

Multivariate data consist of observations on several different variables (features) for a number of individuals or objects. In this chapter we will describe briefly summary statistics of multivariate data, multivariate distributions (Gaussian, Wishart, Hotelling), we will give the unbiased estimators of mean and variance and finally we will provide some simple tests of hypotheses.

1.1 Notation

We begin with an example of multivariate data.
Suppose we have observations for 4 students about their height, weight and age. A simple way to organize these data is in matrix form.

Student	Heigh in cm	Weight in kg	Age
1	167	67	21
2	178	75	22
3	162	52	20
4	190	85	21

Table 1.1: Multivariate example

In this example we have 4 observations (students) with 3 variables (Height,Weight and Age), in total we have $4 \times 3 = 12$ measurements. In general we denote the number of variables by p and the number of objects or individuals by n. Thus the data matrix consists of n rows and p columns and each element x_{ij} represents the j^{th} variable of the i^{th} observation ($i = 1, \ldots, n$, $j = 1, \ldots, p$) and will be denoted by X.

$$X = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix}$$

In this form the data matrix of the example is

$$X = \begin{bmatrix} 167 & 67 & 21 \\ 178 & 75 & 22 \\ 162 & 52 & 20 \\ 190 & 85 & 21 \end{bmatrix}.$$

The data matrix can be seen as n row vectors denoted by $\mathbf{x}_1^T$ to $\mathbf{x}^T$. Thus

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix},$$

where $\mathbf{x}_i^T$ denotes the transpose of $\mathbf{x}_i$ and $\mathbf{x}_i^T = (x_{i1}, x_{i2}, \ldots, x_{ip})$ is the i^{th} row of the data matrix.

1.2 Summary Statistics

The Mean Vector

In the univariate case if we have a sample $x_1, \ldots, x_n$ the sample mean is given by

$$\bar{x} = \frac{1}{n} \sum_{j=1}^{n} x_j.$$

In the multivariate case the sample mean is the vector

$$\bar{X} = \begin{bmatrix} \bar{\mathbf{x}}_1 \\ \bar{\mathbf{x}}_2 \\ \vdots \\ \bar{\mathbf{x}}_p \end{bmatrix},$$

where $\bar{\mathbf{x}}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}$, $j = 1, \ldots, p$ is the mean vector of the j^th variable (column).

Covariance Matrix

The sample Covariance between variables X_j and X_k is

$$s_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k),$$

and if $j = k$ then the covariance s_{jj} is the sample variance of the j^{th} variable.
In multivariate data we can define the **Covariance Matrix** which is a symmetric and positive definite (p.d.) matrix with diagonal elements the variances of variables and the other elements the covariances that correspond in each row and column

$$S = \begin{bmatrix} s_{11} & s_{12} & \ldots & s_{1p} \\ s_{21} & s_{22} & \ldots & s_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ s_{p1} & s_{p2} & \ldots & s_{pp} \end{bmatrix},$$

where $s_j^2 = s_{jj}$ is the sample variance of the variable X_j.

Correlation Matrix

The Correlation matrix is the matrix that contains as elements the Pearson correlation coefficients for each pair X_i and X_j of variables. Pearson correlation coefficient measures only the linear correlation. If we want to measure non linear correlation the Spearman coefficient is suitable for every form of

monotone correlation. Pearson correlation coefficient can be used only for quantitative data. The general form of the **Correlation Matrix** is

$$R = \begin{bmatrix} 1 & r_{12} & \dots & r_{1p} \\ r_{21} & 1 & \dots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \dots & 1 \end{bmatrix},$$

where $r_{ij} = \frac{s_{jk}}{s_j \cdot s_k} = \frac{s_{jk}}{\sqrt{s_{jj}^2}\sqrt{s_{jj}^2}}$, $j, k = 1, 2, \dots, p$ and $-1 \leq r_{ij} \leq 1$.

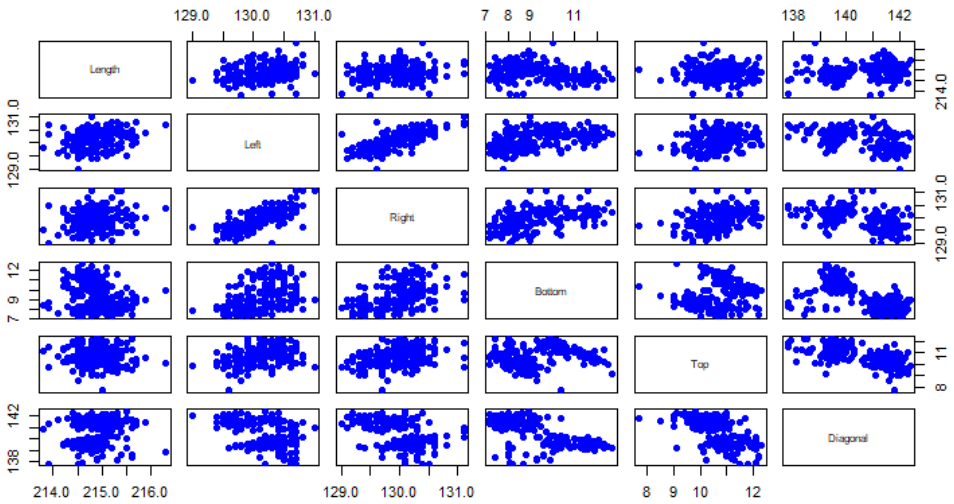
1.3 Graphical Representations

1.3.1 Scatter plots

Scatter plots are bivariate or trivariate graphs of plotted points that show the relationship between two or three data sets. In multivariate data analysis the number of dimensions (variables) is usually bigger than 3. A simple approach is to draw all the possible scatter plots for all pairs of variables. As example we will use the banknote data set[1].The data set contains six measurements made on 100 genuine and 100 counterfeit old-Swiss 1000–franc bank notes. The variables are

1. Status: the status of the banknote (genuine or counterfeit)
2. Length: Length of bill (mm)
3. Left: Width of left edge (mm)
4. Right: Width of right edge (mm)
5. Bottom: Bottom margin width (mm)
6. Top: Top margin width (mm)
7. Diagonal: Length of diagonal (mm)

Figure 1.1: Scatter plot of banknote data



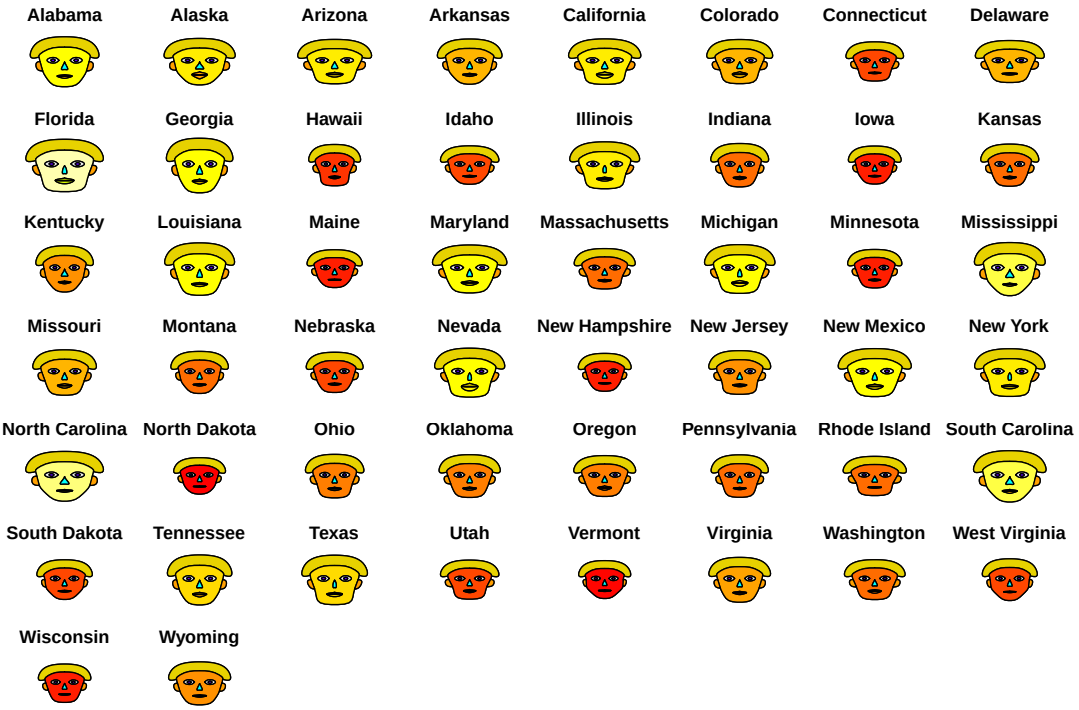
[1]Source Flury, B. and Riedwyl, H. (1988). Multivariate Statistics: A practical approach. London: Chapman & Hall, Tables 1.1 and 1.2, pp. 5-8)

In Figure 1.1 we see that the variables Left and Right are highly correlated.

1.3.2 Chernoff Faces

Chernoff faces display multivariate data in the shape of human face. The size of the individual parts (eyes, hair, nose etc) are assigned to certain variables. As example we will use crime data[2] The data set is consisted of 50 observations(states) and 4 variables (murder,assault,UrbaPop and Rape) of US crime rate by state.

Figure 1.2: Chernoff faces of crime data



where

1. height of face: murder
2. width of face: assault
3. structure of face: UrbanPop
4. height of mouth: Rape

We see that in the states of Florida or North Carolina the criminality is high.

[2]McNeil, D. R. (1977) Interactive Data Analysis. New York: Wiley.

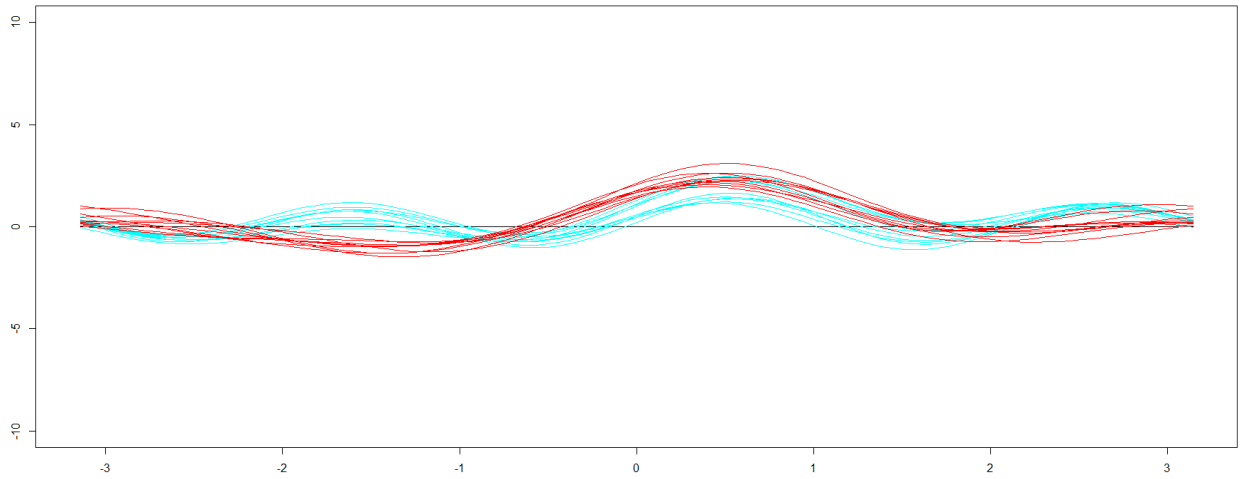
1.3.3 Andrews Curves

Each observation $x_i = (X_{i1}, X_{i2}, \ldots, X_{id})$ is an element of $\mathcal{R}^d$. To visualize them Andrews[3] defines a Fourier series

$$f_X(t) = X_1/\sqrt{2} + X_2 \sin t + X_3 \cos t + X_4 \sin(2t) + X_5 \cos(2t) + \ldots$$

and this function is plotted for $-\pi < t < \pi$. Andrews curves that are represented by functions close together suggest that the corresponding data points will also be close together. As example we will use the banknote data set, we will choose the observations 91 to 110 (the first 10 are genuine and the last 10 are counterfeit).

Figure 1.3: Andrews curves of banknote data.



1.4 Multivariate Distributions

One important concept in multivariate analysis is the idea of multivariate probability distributions. In the univariate case the interest is focused on the distribution of each random variable X_i separately. This can not be assumed in the multivariate case because of the dependence between them. To account such dependencies the joint probability density function (pdf) and the joint probability mass function (pmf) are used in the multivariate case. Consider the case of two random variables X and Y. Probabilities of events defined in terms of these variables can be obtained by operations involving the *cummulative distribution function* (cdf)

$$F(x, y) = P(X \leq x, Y \leq y),$$

in the case that $F(x, y)$ is absolutely continuous (the partial derivative exists almost everywhere)

$$f(x, y) = \frac{d^2 F(x, y)}{dxdy},$$

and

$$F(x, y) = \int_{-\infty}^{y} \int_{-\infty}^{x} f(u, v) du dv.$$

[3]Andrews, D. F. (1972). Plots of high-dimensional data. Biometrics, 125-136.

The nonnegative function $f(x, y)$ is called *joint density* of X and Y and the pair (X, Y) defines a random point in a plane.
For the case of p random variables, the joint probability function of $X_1, X_2, \ldots, X_p$ is $f(x_1, x_2, \ldots, x_p)$. The following distributions are common in the analysis of multivariate data and denoted by the joint pdf

- The **marginal** distribution of the random variable X_i is

$$f(x_i) = \int_{x_1} \int_{x_2} \ldots \int_{x_{i-1}} \int_{x_{i+1}} \ldots \int_{x_p} f(x_1, \ldots, x_p) dx_1 dx_2 \ldots dx_{i-1} dx_{i+1} \ldots dx_p.$$

  Generally the marginal distribution of $X_1, X_2, \ldots, X_m$, $m < p$ is the joint distribution of them and is given from

$$f(x_1, x_2, \ldots, x_m) = \int_{x_{m+1}} \int_{x_{m+2}} \ldots \int_{x_p} f(x_1, \ldots, x_p) dx_{m+1} \ldots dx_p.$$

- The **conditional probability** of the random variables X_1, X_3 given $X_2, X_4, \ldots X_p$ is defined as

$$f(x_1, x_3 | x_2, x_4, \ldots, x_p) = \frac{f(x_1, x_2, \ldots, x_p)}{f(x_2, x_4, \ldots, x_p)}.$$

- The **expected value** of the function $g(X_1, X_2, \ldots, X_p)$ is defined as

$$E[g(X_1, X_2, \ldots, X_p)] = \int_{x_1} \ldots \int_{x_p} g(X_1, X_2, \ldots, X_p) f(x_1, x_2, \ldots, x_p) dx_1 \ldots dx_p$$

Definition 1.4.1 *A Random Vector* $\mathbf{x}^T = (X_1, X_2, \ldots, X_p)$ *is a vector where all* X_i *are random variables.*

Suppose $X_{p\times 1}$ is a random vector

$$\mathbf{x} = \begin{bmatrix} X_1 \\ \vdots \\ X_p \end{bmatrix}$$

- the **expected value** of the random variable is

$$E(\mathbf{x}) = \boldsymbol{\mu} = \begin{bmatrix} E(X_1) \\ \vdots \\ E(X_p) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_p \end{bmatrix}$$

  and if $C_{m\times p}$ is a matrix and $b_{p\times 1}$ is a vector the expected value of $Y = C\mathbf{x} + b$ is

$$E(Y) = CE(\mathbf{x}) + b$$

- the **covariance matrix** of the random vector $\mathbf{x}$ is

$$Cov(\mathbf{x}) = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) & \ldots & Cov(X_1, X_p) \\ Cov(X_2, X_1) & Var(X_2) & \ldots & Cov(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_p, X_1) & Cov(X_p, X_2) & \ldots & Var(X_p) \end{bmatrix}.$$

  It is easy to see that the covariance matrix is symmetric($cov(X_i, X_j) = cov(X_j, X_i)$), and if $C_{m\times p}$ is a matrix and $b_{p\times 1}$ is a vector the covariance matrix of $Y = Cx + b$ is

$$Cov(Y) = Cov(\mathbf{Cx+b}) = \mathbf{C}Cov(\mathbf{x})\mathbf{C}^T$$

1.5 The Multivariate Normal Distribution

The most important multivariate probability distribution is the multivariate normal. If we write the p.d.f of univariate normal with mean μ and variance σ^2 as

$$f(x) = \{2\pi\sigma^2\}^{-1/2} \exp\{-\frac{1}{2}(x-\mu)\{\sigma^2\}^{-1}(x-\mu)\}$$

a plausible extension to p variates is

$$f(x) = |2\pi\mathbf{\Sigma}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\} \tag{1.1}$$

where $\boldsymbol{\mu}$ is the mean vector of $\mathbf{x}$ and $\mathbf{\Sigma}$ is the covariance matrix of x as defined in 1.4.

Definition 1.5.1 *The random vector x is said to have a p dimensional normal (or Gaussian) distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ if its pdf is given by (1.1). We write $\mathbf{x} \sim \mathcal{N}_p(\boldsymbol{\mu}, \mathbf{\Sigma})$*

Theorem 1.5.1 *Let $\mathbf{x} \sim \mathcal{N}_p(\boldsymbol{\mu}, \mathbf{\Sigma})$ and $\mathbf{a}_{p\times 1}$ a constant vector (not random variable) and $y = \mathbf{a}^T\mathbf{x}$, then*

$$y \sim \mathcal{N}(\mathbf{a}^T\boldsymbol{\mu}, \mathbf{a}^T\mathbf{\Sigma}\mathbf{a}) \tag{1.2}$$

A useful application of the theorem (1.5.1) is to find the distribution of the sample mean, from a sample where the values are not independent. Let $X_1, \ldots, X_n$ be the correlated observations where the vector

$$\mathbf{x} = \begin{bmatrix} X_1 & X_2 & \ldots & X_n \end{bmatrix}^T \sim \mathcal{N}_n(\boldsymbol{\mu}, \mathbf{\Sigma}).$$

We can define as $\mathbf{a}_{n\times 1}$ the vector

$$\mathbf{a}^T = \begin{bmatrix} \frac{1}{n} & \frac{1}{n} & \ldots & \frac{1}{n} \end{bmatrix}$$

consequently

$$\bar{x} = \frac{\sum_{i=1}^n X_i}{n} = \mathbf{a}^T\mathbf{x} \sim \mathcal{N}(\mathbf{a}^T\boldsymbol{\mu}, \mathbf{a}^T\mathbf{\Sigma}\mathbf{a})$$

where

$$\mathbf{a}^T\boldsymbol{\mu} = \frac{\sum_{i=1}^n \mu_i}{n}$$

and

$$\mathbf{a}^T\mathbf{\Sigma}\mathbf{a} = \frac{\sum_{i=1}^n \sum_{j=1}^n Cov(X_i, X_j)}{n^2}.$$

Therefore

$$\bar{x} \sim \mathcal{N}\left(\frac{\sum_{i=1}^n \mu_i}{n}, \frac{\sum_{i=1}^n \sum_{j=1}^n Cov(X_i, X_j)}{n^2}\right).$$

In the case of independent and identically distributed random variables X_i, $\bar{x} \sim \mathcal{N}(\mu, \sigma^2)$

Theorem 1.5.2 *Let $\mathbf{x} \sim \mathcal{N}_p(\boldsymbol{\mu}, \mathbf{\Sigma})$ and $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{c}$ where $\mathbf{A}$ is any $(q \times p)$ matrix and $\mathbf{c}$ $(p \times 1)$ vector, then*

$$\mathbf{y} \sim \mathcal{N}_q(A\boldsymbol{\mu} + c, A\mathbf{\Sigma}A^T) \tag{1.3}$$

Theorem 1.5.3 *Let* $\mathbf{x}$ *with pdf given by the equation (1.1), and*

$$\mathbf{y} = \mathbf{\Sigma}^{-1/2}(\mathbf{x} - \boldsymbol{\mu}) \tag{1.4}$$

($\mathbf{y}$ is the Mahalanobis Transformation) where $\mathbf{\Sigma}^{-1/2}$ is the square root of $\mathbf{\Sigma}^{-1}$. Then $y_1^T, y_2^T \ldots, y_p^T$ are independent $\mathcal{N}(0,1)$ variables.

Theorem 1.5.4 *Let* $\mathbf{x}$ *with pdf given by the equation (1.1), then*

$$(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \sim \chi_p^2. \tag{1.5}$$

Theorem 1.5.5 *Let* $\mathbf{x}$ *with pdf given by the equation (1.1), then all linear (non-trivial) combinations of the elements of* $\mathbf{x}$ *are univariate normal.*

Theorem 1.5.6 *If* $\mathbf{x} \sim \mathcal{N}_p(\boldsymbol{\mu}, \mathbf{\Sigma})$*, then* $\mathbf{Ax}$ *and* $\mathbf{Bx}$ *are independent if and only if* $\mathbf{A\Sigma B^T} = \mathbf{0}$.

Theorem 1.5.7 *Let* $\mathbf{x} \sim \mathcal{N}_p(\boldsymbol{\mu}, \mathbf{\Sigma})$ *and* $\mathbf{x}^T = (X_1, X_2, \ldots, X_p)$*. Suppose that we want to calculate the distribution of the first* q *variables, the vector of* $X_1, \ldots, X_q$ *is* $\mathbf{x} = (X_1, \ldots, X_q)$ $(q < p)$*.If* $\mathbf{y}_{q\times 1} = \mathbf{A}_{q\times p}\mathbf{x}_{p\times 1}$*, where* $\mathbf{A}_{q\times p} = [\mathbf{I}_{q\times q} \vdots \mathbf{0}_{q\times(p-q)}]$ *then from the theorem (1.5.2) can be derived that* $\mathbf{y} \sim \mathcal{N}_q(\mathbf{A}^T\boldsymbol{\mu}, \mathbf{A}^T\mathbf{\Sigma A})$.

Theorem 1.5.8 *Let* $\mathbf{x} \sim \mathcal{N}_p(\boldsymbol{\mu}, \mathbf{\Sigma})$*, the conditional distribution of* $\curvearrowleft_1$ *for a given* $\mathbf{x}_2$ *of is*

$$\mathbf{x_1}|\mathbf{x_2} \sim \mathcal{N}_\mathbf{q}(\boldsymbol{\mu_1} + \mathbf{\Sigma_{12}\Sigma_{22}^{-1}}(\mathbf{x_2} - \boldsymbol{\mu_2}), \mathbf{\Sigma_{11}} - \mathbf{\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}}) \tag{1.6}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}$*, q is the dimension of* $\mathbf{x}_1$ *vector,*

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \text{ and } \quad \mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{pmatrix}.$$

Estimation of Parameters

The unknown parameters of multivariate normal distribution is the mean vector $\boldsymbol{\mu}$ (vector $(p \times 1)$ and the $p \times p$ Covariance matrix $\mathbf{\Sigma}$. We have to estimate $p(p+1)/2$ parameters for the covariance matrix (because it is symmetric) and p for the mean vector. The problem of estimation is complicated due to the number of equations needed $(p(p+3)/2)$.
Assume that we have n random p-vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, iid (independent and identically distributed) as multivariate Normal vectors,

$$\mathbf{x}_i \sim \mathcal{N}_p(\boldsymbol{\mu}, \mathbf{\Sigma}), \quad i = 1, 2, \ldots, n,$$

where the parameters $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are both unknown. Using the method of *maximum likelihood* (ML) we derive that

$$\hat{\boldsymbol{\mu}} = \bar{\mathbf{X}} \tag{1.7}$$

and

$$\hat{\mathbf{\Sigma}} = S, \tag{1.8}$$

where S is the sample covariance matrix as defined in the Section 1.2. Note here, that the same results can be derived using the *method of moments*.

1.6 Wishart distribution

Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ be independent random $(p \times 1)$ vectors where $\mathbf{x}_i \sim \mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}), i = 1, 2, \ldots, n$. We say that the random and positive definite symmetric $(p \times p)$ matrix

$$\mathbf{W} = \sum_{i=1}^{n} X_i X_i^T, \tag{1.9}$$

has the *Wishart* distribution with n degrees of freedom and matrix $\boldsymbol{\Sigma}$. If $\mu_i = 0$ for all i, the *Wishart* distribution is termed *central*, otherwise *noncentral*. If $\mathbf{W}$ has a *Wishart* density we write

$$\mathbf{W} \sim \mathcal{W}_p(n, \boldsymbol{\Sigma}). \tag{1.10}$$

Some interesting points about the *Wishart* are

- when $p = 1$, $\mathcal{W}_1(n, \sigma^2)$ is identical to the $\sigma^2 \chi_n^2$ distribution
  Wishart which is denoted by $\mathcal{W}_1(n, \boldsymbol{\Sigma}, \boldsymbol{\mu})$ where $\boldsymbol{\mu} = \sum_{i=1}^{n} \mu_i$ is the *noncentrality parameter*
- $\mathbf{E}(\mathbf{W}) = n\boldsymbol{\Sigma}$
- The inverse of a matrix $\mathbf{W}$ that is distributed as *inverse Wishart* and is useful in Bayesian Statistics.
- The *Wishart* distribution is very important because it is the distribution of the sample covariance matrix
- if the mean values of vectors $\mathbf{x}_i$ are not equal to zero, we have the *noncentral*.

Theorem 1.6.1 *Suppose a random sample of independent random vectors* $\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_n$. *If* $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$, *then*

$$n\mathbf{S} = \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \sim \mathcal{W}_p(n-1, \boldsymbol{\Sigma}). \tag{1.11}$$

Theorem 1.6.2 *The sample mean vector* $\bar{\mathbf{x}}$ *and the sample covariance matrix* $\mathbf{S}$ *are independent.*

Properties of the Wishart Distribution

1. Let $\mathbf{W_i} \sim \mathcal{W}_\mathbf{p}(\mathbf{n_i}, \boldsymbol{\Sigma}), i = 1, 2, \ldots, m$, be independently distributed. Then
$$\sum_{\mathbf{i=1}}^{\mathbf{p}} \mathbf{W_i} \sim \mathcal{W}_\mathbf{p}(\sum_{\mathbf{i=1}}^{\mathbf{p}} \mathbf{n_i}, \boldsymbol{\Sigma})$$
.
2. Suppose $\mathbf{W} \sim \mathcal{W}_\mathbf{p}(\mathbf{n}, \boldsymbol{\Sigma})$ and $\mathbf{A}$ a $(q \times p)$ matrix of fixed constants with rank q. Then,
$$\mathbf{AWA^T} \sim \mathcal{W}_\mathbf{p}(\mathbf{n}, \mathbf{A\Sigma A^T})$$
.
3. Suppose $\mathbf{M} \sim \mathcal{W}(\mathbf{m}, \boldsymbol{\Sigma})$ and $\mathbf{c}$ is a $(p \times 1)$ vector, then
$$\frac{\mathbf{c^T M c}}{\boldsymbol{\sigma^2}} \sim \boldsymbol{\chi}^\mathbf{2}_\mathbf{m}$$
where $\sigma^2 = \mathbf{c^T \Sigma c}$.

1.7 The Hotelling $\mathcal{T}^2$ Distribution

Theorem 1.7.1 *Suppose a random vector* $\mathbf{x}$ *and a random matrix* $\mathbf{W}$ *which are distributed as* $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *and* $\mathcal{W}_p(m, \boldsymbol{\Sigma})$ *then the quantity*

$$(\mathbf{x} - \boldsymbol{\mu})^{\mathbf{T}} \mathbf{M}^{-\mathbf{1}} (\mathbf{x} - \boldsymbol{\mu}) \sim \mathcal{T}^{\mathbf{2}}(\mathbf{p}, \mathbf{m}) \tag{1.12}$$

This theorem is very important. Indeed, from previous theorems we know that

$$\bar{\mathbf{x}} \sim \mathcal{N}_p(\boldsymbol{\mu}, \frac{1}{n}\Sigma)$$

and

$$n\mathbf{S} = \sum_{\mathbf{i=1}}^{\mathbf{n}} (\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^{\mathbf{T}} \sim \mathcal{W}_{\mathbf{p}}(\mathbf{n} - \mathbf{1}, \boldsymbol{\Sigma})$$

so that from the above theorem the following results can be derived

$$\mathbf{T^2} = (\mathbf{n} - \mathbf{1})(\bar{\mathbf{x}} - \boldsymbol{\mu})^{\mathbf{T}} \boldsymbol{\Sigma}^{-\mathbf{1}} (\bar{\mathbf{x}} - \boldsymbol{\mu}) \sim \mathcal{T}^{\mathbf{2}}(\mathbf{p}, \mathbf{n} - \mathbf{1})$$

. Some useful properties are

- $\frac{\mathbf{m-p+1}}{\mathbf{mp}} \mathcal{T}^{\mathbf{2}}(\mathbf{p}, \mathbf{m}) \sim \mathbf{F}(\mathbf{p}, \mathbf{m} - \mathbf{p} + \mathbf{1})$
- $\mathcal{T}^2$ is invariant under linear transformations
- $\mathcal{T}^2$ is a generalization of the simple $\mathbf{t}$ distribution and if $p = 1$ then T^2 is $t = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$

1.8 Hypothesis Testing

The advantage of multivariate analysis is that we can create test of hypotheses for a variety of variables simultaneously making use of the information that can be drawn from the covariance. A simple approach is to create a test of hypothesis for each variable separately, which is inefficient. Suppose we have 4 variables and the test of hypothesis for each one separately has a level of significance 5% then the significance level of the multiple test will be $1 - 0.95^4$ which is equal to 19% therefore the probability of error is very large. An important problem in this simple approach is that we lose the information that the covariance can give us because for simplicity we have assumed independence.

1.8.1 Testing hypotheses on the multivariate normal vector

Multivariate Test with known Σ

Consider testing a null hypotheses $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$ against an alternative hypothesis $H_1 : \boldsymbol{\mu} \neq \boldsymbol{\mu}_0$. In this case the distribution of the likelihood ratio statistic under H_0 is:

$$W = n(\bar{\boldsymbol{x}} - \boldsymbol{\mu_0})^T \boldsymbol{\Sigma}^{-1} (\bar{\boldsymbol{x}} - \boldsymbol{\mu_0}) \sim \chi^2_p.$$

We reject H_0 at level α if

$$W > \chi^2_{p,\alpha}$$

Multivariate Test with Σ unknown

Consider again testing a null hypotheses $H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_0$ against an alternative hypothesis $H_1 : \boldsymbol{\mu} \neq \boldsymbol{\mu}_0$. In this case the statistic that is used is

$$\mathbf{T^2} = \mathbf{n}(\bar{\mathbf{x}} - \boldsymbol{\mu_0})^{\mathbf{T}}\mathbf{S}^{-\mathbf{1}}(\bar{\mathbf{x}} - \boldsymbol{\mu_0})$$

which follow a $T^2(p, n-1)$ distribution. We can use the transformation

$$F = \frac{n-p}{p(n-1)}T^2 \sim F_{p,n-p}$$

and we reject the H_0 at level α if

$$F > F_{p,n-p,\alpha}.$$

Multivariate Test for Equality of Mean Vectors when $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$

Consider testing a null hypotheses $H_0 : \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$ against an alternative hypothesis $H_1 : \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$. In this case we will consider the modified T^2 Hotelling's statistic

$$T^2 = (\bar{\mathbf{x}}_\mathbf{1} - \bar{\mathbf{x}}_\mathbf{2})^T \left\{ \frac{1}{n_1}\mathbf{S}_1 + \frac{1}{n_2}\mathbf{S}_2 \right\}^{-1} (\bar{\mathbf{x}}_\mathbf{1} - \bar{\mathbf{x}}_\mathbf{2}),$$

which for large samples is approximately χ^2_p distributed. We reject the H_0 at level α if

$$T^2 > \chi^2_{p,\alpha}$$

If the samples are small we can calculate the F transformation

$$F = \frac{n_1 + n_2 - p - 1}{p(n_1 + n_2 - 2)}T^2 \;\dot{\sim}\; F_{p,\nu}$$

where ν is given by

$$\frac{1}{\nu} = \sum_{i=1}^{2} \frac{1}{n_i - 1} \left\{ \frac{(\bar{\mathbf{x}}_\mathbf{1} - \bar{\mathbf{x}}_\mathbf{2})^T \mathbf{S}_T^{-1}(\frac{1}{n_i}\mathbf{S}_i)\mathbf{S}_T^{-1}(\bar{\mathbf{x}}_\mathbf{1} - \bar{\mathbf{x}}_\mathbf{2})}{T^2} \right\}^2$$

and

$$\mathbf{S_T} = \frac{1}{n_1}\mathbf{S_1} + \frac{1}{n_2}\mathbf{S}_2.$$

We reject H_0 at level α if

$$F > F_{p,\nu,\alpha}.$$

For further reading please see Seber (1984)

Testing Homogeneity of Covariance Matrices

Suppose that $\Sigma_i, \quad i = 1, 2, \ldots, k$ is the covariance matrix of the population i. We need to test the hypotheses
$H_0 : \Sigma_1 = \Sigma_2 = \ldots = \Sigma_k$ against
H_1 : at least two are different.
The test statistic is called **Box-M** and uses the statistic

$$\mathbf{M} = \boldsymbol{\phi} \sum_{\mathbf{i=1}}^{\mathbf{k}} [(\mathbf{n_i} - \mathbf{1})\mathbf{ln}(|\mathbf{S_i^{-1}S_{pooled}}|)]$$

where p is the number of variables, $\mathbf{S_{pooled}} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{k}}(\mathbf{n_i}-\mathbf{1})\mathbf{S_i}}{\mathbf{n}-\mathbf{k}}$
$\boldsymbol{\phi} = \mathbf{1} - \frac{\mathbf{2p^2}+\mathbf{3r}-\mathbf{1}}{\mathbf{6(p+1)(k-1)}} \sum_{\mathbf{i=1}}^{\mathbf{k}} \frac{\mathbf{1}}{\mathbf{(n_i-k)(n-k)}}$
$\mathbf{n} = \sum_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{n_i}$.
The asymptotic distribution of $\mathbf{M}$ under H_0 is $\boldsymbol{\chi}^2_{p(p+1)(k-1)/2}$.

Chapter 2

Multivariate Analysis Techniques

In this chapter we will illustrate three multivariate analysis techniques. The first is Principal Components Analysis (PCA) which is used for dimension reduction. The second method is Discriminant Analysis (DA) that is a supervised method used for classification problems. The third method belongs to the family of unsupervised methods for clustering of data. Specifically we will illustrate the K–means and Hierarchical clustering algorithms, both of which belong to the third class.

2.1 Principal Component Analysis

The Principal Components Analysis method is used to create linear combinations of the initial variables which are uncorrelated and explain the same amount of variation contained in the initial set of variables. Suppose that we have a random vector $\mathbf{x} = (X_1 \quad X_2 \quad \dots \quad X_p)$ with covariance matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1p} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \dots & \sigma_p^2 \end{pmatrix}.$$

Now consider the linear combinations

$$\begin{aligned} Y_1 &= a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\ Y_2 &= a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\ &\vdots \\ Y_p &= a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p. \end{aligned}$$

We can write these equations in matrix form as $\mathbf{Y} = \mathbf{AX}$ where $\mathbf{Y}, \mathbf{X}$ are $(p \times 1)$ vectors and $\mathbf{A}$ the $(p \times p)$ matrix where

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \dots & a_{pp} \end{pmatrix}.$$

The variance of the principal component $\mathbf{Y}_i$ is given by the equation

$$\text{var}(Y_i) = \sum_{k=1}^{p}\sum_{l=1}^{p} a_{ik}a_{il}\sigma_{kl} = \mathbf{a}_i^T \Sigma \mathbf{a}_i$$

where $\mathbf{a}_i = \begin{pmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{ip} \end{pmatrix}$.

The first principal component is the one with the largest variation and we write

$$\text{var}(Y_1) = \sum_{k=1}^{p}\sum_{l=1}^{p} a_{1k}a_{1l}\sigma_{kl} = \mathbf{a}_1^T\Sigma\mathbf{a}_1$$

with the constraint that

$$\mathbf{a}_1^T\mathbf{a}_1 = \sum_{j=1}^{p} a_{1j}^2 = 1$$

To determine the a_{1j}, $j = 1, \ldots p$ we will maximize the function

$$\mathbf{L}(\mathbf{a_1}) = \mathbf{a_1^T\Sigma a_1} - \boldsymbol{\lambda}(\mathbf{a_1^T a_1 - 1}),$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier.
Using derivatives to calculate the maximum we have

$$\frac{\partial\mathbf{L}(\mathbf{a_1})}{\partial\mathbf{a_1}} = \mathbf{2(\Sigma - \boldsymbol{\lambda} I)a_1 = 0}$$

consequently the maximum is given by the equation

$$\mathbf{\Sigma a_1} = \boldsymbol{\lambda}\mathbf{a_1}$$

which is the equation of eigenvectors of matrix $\mathbf{\Sigma}$ where $\boldsymbol{\lambda}$ is the eigenvalue. The variance of Y_1 will be equal to $\boldsymbol{\lambda}$ and since Y_1 is reputed to carry the largest variation, $\boldsymbol{\lambda}$ will be chosen to be the largest eigenvalue $\boldsymbol{\lambda}_1$ of $\mathbf{\Sigma}$. As a result $\mathbf{a}_1$ will be the associated eigenvector
Thus if

$$\boldsymbol{\lambda}_1 > \boldsymbol{\lambda}_2 > \ldots > \boldsymbol{\lambda}_{p-1} > \boldsymbol{\lambda}_p,$$

then the p^{th} principal component Y_p will carry variation equal to $\boldsymbol{\lambda}_p$ and the vector $\mathbf{a}_p$ will be the associated eigenvector.

Therefore

- The greatest eigenvalue and eigenvector correspond to the first principal component Y_1, the second greater eigenvalue corresponds to the second principal component etc.
- The variance of each principal component is equal to the corresponding eigenvalue $Var(Y_j) = \lambda_j$
- The principal components are uncorrelated thus the covariance matrix is a diagonal matrix with eigenvalues
- The total variance of principal components will be equal to the initial amount of variation
- The quantity $\frac{\lambda_j}{\sum_{i=1}^{p}\lambda_i}$ is the percentage of the total variation that is explained by the principal component j.
- The quantity $\frac{\sum_{j=1}^{k}\lambda_j}{\sum_{i=1}^{p}\lambda_i}$ is the percentage of the total variation that is explained by the first k principal components.

Note that if the variables are not measured in the same scale it is necessary to normalize the variables, and we use the correlation matrix instead.

PCA is a useful method

- In linear regression when the covariates are correlated we face the problem of collinearity where the OLS estimation is inconsistent.

- In graphical representations to multivariate data due to the high dimensionality are often quite complex. If the first few principal components explain a large part of the total variation we can achieve a good graphical representation, focusing exclusively on them.

- In data mining where we can compress the information from large data set into a reduced number of dimensions.

Principal Components Analysis Procedure

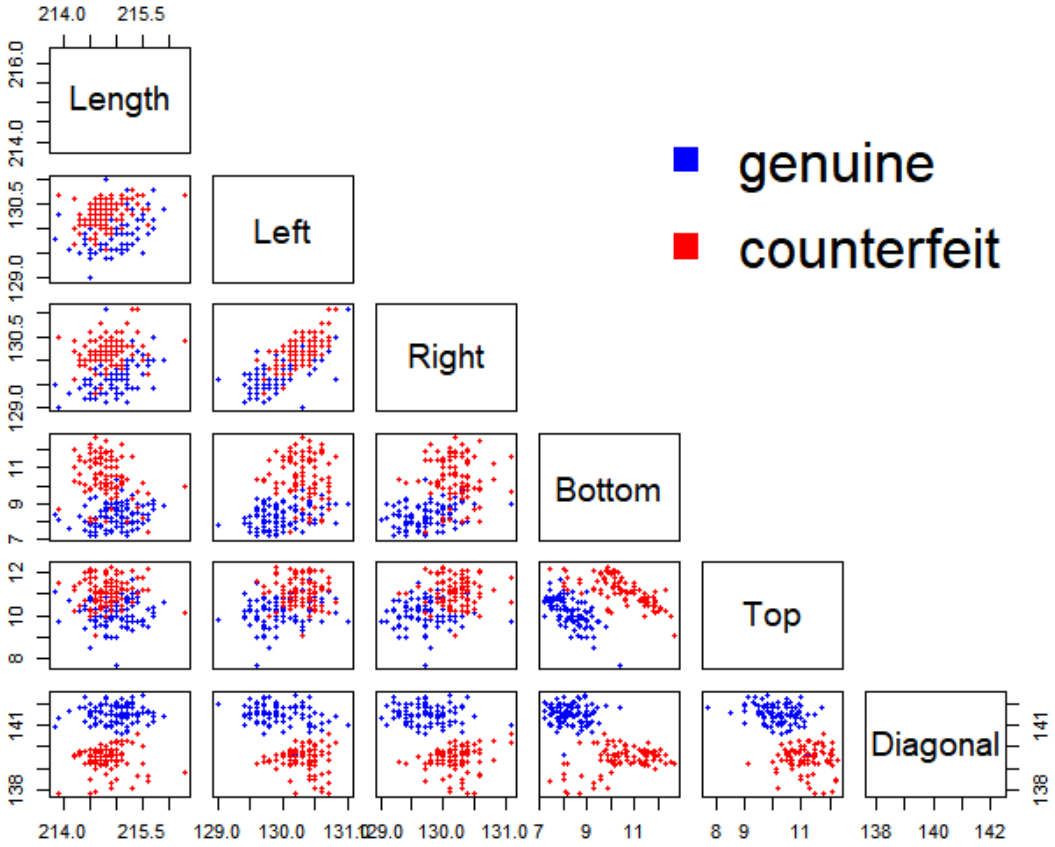
Step 1: The first step of the procedure is to check if the variables are correlated from the covariance matrix. If they are not it is not reasonable to continue the procedure. Variables that are not correlated with others are not useful to the analysis.
Step 2: If the units of measurement are not the same for all variables we use the correlation matrix.
Step 3: We calculate eigenvalues and eigenvectors of the matrix.
Step 4: Choose the number of components. This can be chosen by Kaiser's criterion, Scree plot, or to choose a significance proportion (e.g. 90%) of the total variation to be explained by the principal components.

2.1.1 Example of PCA

The Swiss Bank Notes data set[1] consists of 200 bank notes (100 genuine and 100 counterfeit) and the variables are

[1]Flury, B. and Riedwyl, H. (1988). Multivariate Statistics: A practical approach. London: Chapman & Hall, Tables 1.1 and 1.2, pp. 5-8.

Figure 2.1: Scatter Plot of Swiss Bank Notes



X_1: Length (Length of bill (mm))
X_2: Left (Width of left edge (mm))
X_3: Right (Width of right edge (mm))
X_4: Bottom (Bottom margin width (mm))
X_5: Top (Top margin width (mm))
X_6: Diagonal (Length of diagonal (mm))

In this data set the variables are on the same scale thus for the principal components procedure we will use the sample covariance matrix. From the scatter plot it's easy to see that the variables are correlated so it is reasonable to use the PCA method for dimension reduction. The eigenvalues are

$$\boldsymbol{\lambda} = (2.98530335, 0.93094242, 0.24219664, 0.19368545, 0.08478579, 0.03533710)$$

and the corresponding eigenvectors are the columns of the matrix

$$A = \begin{bmatrix} -0.04 & 0.01 & -0.33 & 0.56 & 0.75 & 0.10 \\ 0.11 & 0.07 & -0.26 & 0.46 & -0.35 & -0.77 \\ 0.14 & 0.07 & -0.34 & 0.42 & -0.53 & 0.63 \\ 0.77 & -0.56 & -0.22 & -0.19 & 0.10 & -0.02 \\ 0.20 & 0.66 & -0.56 & -0.45 & 0.10 & -0.03 \\ -0.58 & -0.49 & -0.59 & -0.26 & -0.08 & -0.05 \end{bmatrix}$$

From the matrix A we derive that the first two principal components are

$$Y_1 = -0.04X_1 + 0.11X_2 + 0.14X_3 + 0.77X_4 + 0.20X_5 - 0.58X_6$$
$$Y_2 = 0.01X_1 + 0.07X_2 + 0.07X_3 - 0.56X_4 + 0.66X_5 - 0.49X_6$$

which explain $87,57\%$ of the total variation (Table 2.1) and if we choose three principal components they will explain 92.98% of the total variation.

	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	1.7321	0.9673	0.4934	0.4412	0.2919	0.1885
Proportion of Variance	0.6675	0.2082	0.0542	0.0433	0.0190	0.0079
Cumulative Proportion	0.6675	0.8757	0.9298	0.9731	0.9921	1.0000

Table 2.1: Importance of components

The first principal component is essentially the difference between the Bottom and Diagonal and the second is the difference between the Top and the sum of Bottom and Diagonal.
There are many methods for choosing the number of principal components, some of them are:

- **Choosing the proportion of the total variation that is explained by the first q principal components** (e.g.80% or 90%)

- **Kaiser's Criterion**: Let λ, $i = 1, 2, \ldots, k$ be the eigenvalues of the matrix. We choose the eigenvalues that are greater than $\bar{\boldsymbol{\lambda}} = \sum_{\mathbf{i=1}}^{\mathbf{k}} \boldsymbol{\lambda}_\mathbf{i}/\mathbf{k}$. In this example $\bar{\boldsymbol{\lambda}} = \mathbf{0.75}$ thus we choose 2 components.

- **Scree Plot**: is a plot where the X axis denotes the number of components and the corresponding eigenvalues (variances) are represented at the Y-axis. The optimum number of components happens around the "elbow", where the graph becomes flat. If we use the scree plot method, the number of components will be three (see Figure 2.2)

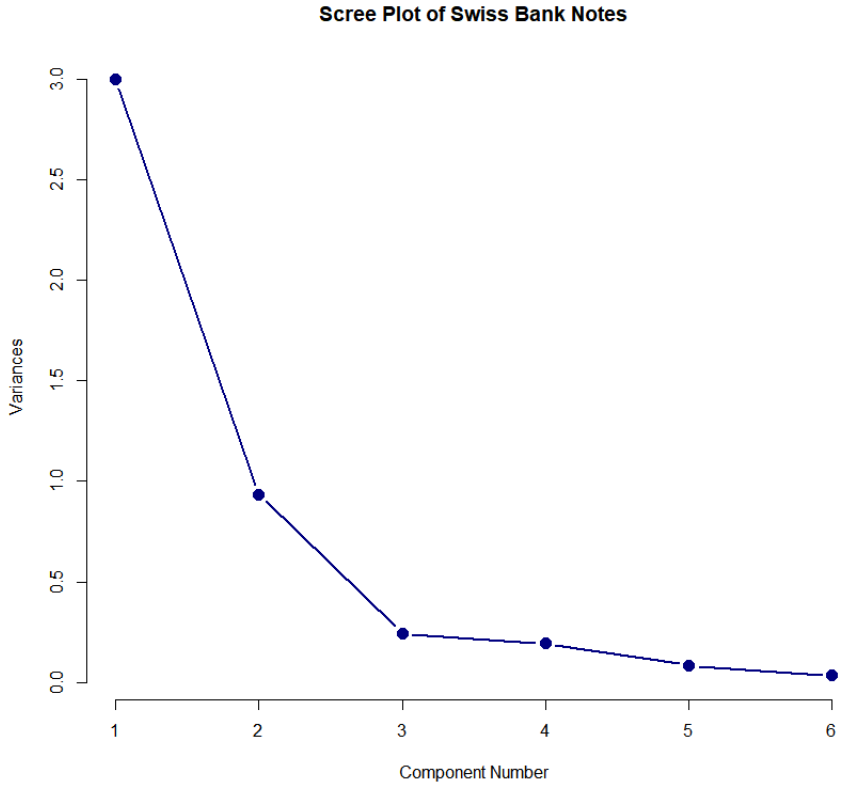


Figure 2.2: PCA Scree Plot

The next graph shows which of the initial variables are most correlated with Y_1 and Y_2

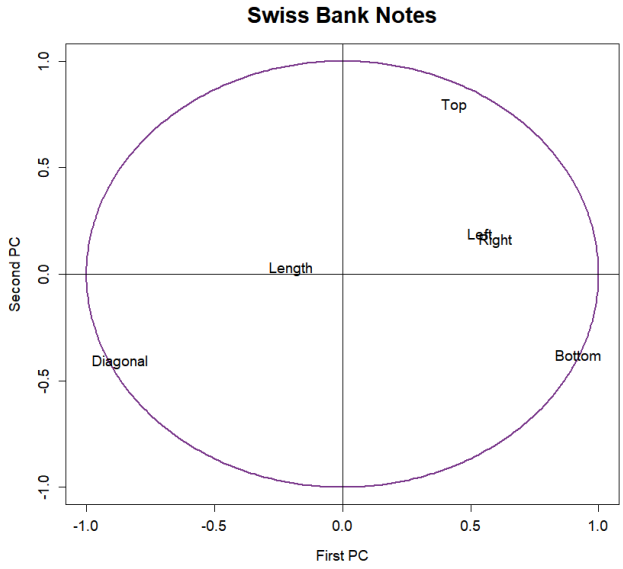


Figure 2.3: Correlation of initial variables

The next figure plots the estimated loadings of the first two principal components using arrows to indicate their directions.

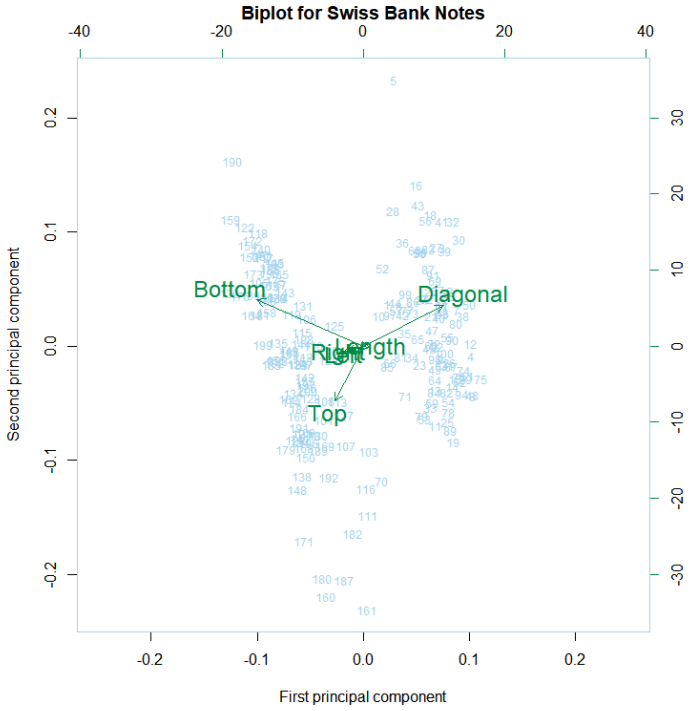


Figure 2.4: Biplot of the first two principal components

2.2 Discriminant Analysis

Suppose that we have two populations π_1 and π_2 with known distributions $\mathcal{N}(0,1)$ and $\mathcal{N}(1,2)$ respectively, and suppose that we have a new observation $x = 2$ and we need to classify x to one of these populations. A simple way to classify x is to calculate the likelihood of observation x under the two distributions and choose the one with the greater likelihood. By calculating $P(\pi_1|x) = f(2|\mu = 0, \sigma^2 = 1) = 0.07635476$ and $P(\pi_2|x) = f(2|\mu = 1, \sigma^2 = 2) = 0.1467627$ we classify the observation to the second population. We can generalize this for more than two populations and for multivariate vectors $\mathbf{x}$ by classifying each observation $\mathbf{x}$ to the population for which the value of $P(\pi_i|\mathbf{x})$ is greatest.
Using the Bayes theorem we derive that

$$P(\pi|\mathbf{x}) = \frac{P(\mathbf{x}|\pi_i)P(\pi_i)}{P(\mathbf{x})} = \frac{p_i f(\mathbf{x}|\pi_i)}{\sum_{j=1}^{k} p_j f(\mathbf{x}|\pi_j)}$$

where $P(\pi_i) = p_i$ and k is the number of populations.
Let $k = 2$, then we classify $\mathbf{x}$ to population 1 if

$$\frac{p_1 f(\mathbf{x}|\pi_1)}{p_2 f(\mathbf{x}|\pi_2)} > 1$$

and this can be written as

$$\frac{f(\mathbf{x}|\pi_1)}{f(\mathbf{x}|\pi_2)} > \frac{p_2}{p_1}.$$

Discriminant Analysis Procedure[2]

- **Step 1**: Collect training data
  We collect the training data that we actually know to which population each subject belongs

- **Step 2** : *PriorProbabilities*
  The prior probability π_i represents the expected portion of the community that belongs to population π_i. There are three choices:

  1. Arbitrary priors are selected according to the investigators beliefs regarding the relative population sizes with the constraint that

  $$\hat{p}_1 + \hat{p}_2 + \cdots + \hat{p}_k = 1$$

  2. Equal priors:

  $$\hat{p}_i = \frac{1}{k}$$

  3. Estimated priors:

  $$\hat{p}_i = \frac{n_i}{N}$$

  where n_i is the number of observations from population π_i in the training data and N is the total number of observations.

- **Step 3**: Use Barlett's test to test the homogenity of the populations
  Case 1: If the populations are homogeneous

  $$\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \cdots = \mathbf{\Sigma}_k = \mathbf{\Sigma}$$

  we use Linear Discriminant analysis.
  Case 2: If the populations are heterogeneous

  $$\mathbf{\Sigma}_i \neq \mathbf{\Sigma}_j \quad \text{for some} \quad i \neq j$$

  We do not discuss testing whether the means of the populations are different. If they are not, there is no case for DA.

- **Step 4**: Estimate the parameters of $f(\mathbf{X}|\pi_i)$. At this point we shall make the following assumptions

  1. The data from group i has μ_i.
  2. The data from group i covariance matrix Σ.
  3. The subjects are independently sampled.
  4. The data are multivariate normal distributed.

- **Step 5** Compute discriminant functions.

- **Step 6** Use cross validation to estimate misclassification probabilities.

- **Step 7** Classify observations with unknown group memberships.

[2] `https://onlinecourses.science.psu.edu/stat505/node/89`

2.2.1 Linear Discriminant Analysis

In Linear Discriminant Analysis (LDA) we assume that each population π_i distributed according to a multivariate normal distribution with mean μ_i and common covariance matrix Σ for all populations. We classify the observation $\mathbf{x}$ to the population with the largest $p_i f(\mathbf{x}|p_i)$ where

$$f(\mathbf{x}|\pi_i) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_i})^{\mathbf{T}}\mathbf{\Sigma^{-1}}(\mathbf{x}-\boldsymbol{\mu_i})\right]$$

Equivalently we can use the log transform of $p_i f(\mathbf{x}|\pi_i)$ which is $\log[p_i f(\mathbf{x}|p_i)]$.
The *Linear Score Function* is:

$$s_i^L(\mathbf{X}) = -\frac{1}{2}\boldsymbol{\mu_i^T}\mathbf{\Sigma^{-1}}\boldsymbol{\mu_i} + \boldsymbol{\mu_i^T}\mathbf{\Sigma^{-1}}\mathbf{x} + \log p_i = d_{i0} + \sum_{j=1}^{p} d_{ij}x_j + \log p_i$$

where

$$d_{i0} = -\frac{1}{2}\boldsymbol{\mu_i^T}\mathbf{\Sigma^{-1}}\boldsymbol{\mu_i}$$

$$d_{ij} = j\text{th element of } \boldsymbol{\mu}_i^T\mathbf{\Sigma}^{-1}$$

Linear Discriminant Function

$$d_i^L(\mathbf{x}) = -\frac{1}{2}\boldsymbol{\mu_i^T}\mathbf{\Sigma^{-1}}\boldsymbol{\mu_i} + \boldsymbol{\mu_i^T}\mathbf{\Sigma^{-1}}\mathbf{x} = d_{i0} + \sum_{j=1}^{p} d_{ij}x_j$$

To calculate $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ we use the estimators $\bar{\mathbf{x}}$ and $\mathbf{S}_p$ (pooled covariance matrix) respectively where

$$\mathbf{S}_p = \frac{\sum_{i=1}^{k}(n_i-1)\mathbf{S}_i}{\sum_{i=1}^{k}(n_i-1)}$$

to obtain the estimated linear score function:

$$\hat{s}_i^L(\mathbf{x}) = -\frac{1}{2}\bar{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{S}_\mathbf{p}^{-\mathbf{1}}\bar{\mathbf{x}}_\mathbf{i} + \bar{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{S}_\mathbf{p}^{-\mathbf{1}}\mathbf{x} + \log\hat{p}_i = \hat{d}_{i0} + \sum_{j=1}^{p}\hat{d}_{ij}x_j + \log p_i$$

where

$$\hat{d}_{i0} = -\frac{1}{2}\bar{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{S}_\mathbf{p}^{-\mathbf{1}}\bar{\mathbf{x}}_\mathbf{i}$$

and

$$\hat{d}_{ij} = jth \quad \text{element of} \quad \bar{\mathbf{x}}_\mathbf{i}^\mathbf{T}\mathbf{S}_\mathbf{p}^{-\mathbf{1}}.$$

Decision Rule: Classify the sample unit into the population that has the largest estimated linear score function.

Posterior Probabilities

The posterior probabilities measure the uncertainty regarding the classification of a unit from an unknown group. The posterior probability that an observation belongs to the population i is

$$p(\pi_i|\mathbf{x}) = \frac{\exp\hat{s}_i^L(\mathbf{x})}{\sum_{i=1}^{k}\exp\hat{s}_i^L\mathbf{x}}$$

Estimating Misclassification Probabilities

The uncertainty is a part of any statistical procedure. When we classify observations according to a decision rule there is always a possibility of misclassification. Below we describe three methods concerned with the estimation of the misclaffication probability.

Method 1.The confusion table describes how the discriminant function will classify each observation in the data set. In general, the confusion table takes the form:

Truth	1	2	...	k	Total
1	n_{11}	n_{12}	...	n_{1k}	$n_{1.}$
2	n_{21}	n_{22}	...	n_{2k}	$n_{2.}$
3	$\vdots$	$\vdots$	...	$\vdots$	$\vdots$
4	n_{k1}	n_{k2}	...	n_{kk}	$n_{k.}$
Total	$n_{.1}$	$n_{.2}$	...	$n_{.k}$	$n_{..}$

The sum of n_{ij}, $i \neq j$ is the number of misclassified observations, so that the misclassification probabilities can be estimated can be obtained by

$$\hat{p}(i|j) = \frac{\sum_{j=1}^{k} n_{ji}}{n_{i.}}$$

where $i \neq j$ and i is the number of column.

Method 2: Set Aside Method
Step 1: Randomly partition the observations into two "halves"
Step 2: Use one "half" to obtain the discriminant function.
Step 3: Use the discriminant function from Step 2 to classify all members of the second "half" of the data, from which the proportion of misclassified observations can be computed.
Advantage: This method yields unbiased estimates of the misclassification probabilities.
Problem: Does not make optimum use of the data, and so, estimated misclassification probabilities are not as precise as possible.

Method 3: Cross validation
Step 1: Delete one observation from the data.
Step 2: Use the remaining observations to compute a discriminant function.
Step 3: Use the discriminant function from Step 2 to classify the observation removed in Step 1. Steps 1-3 are repeated for all observations; compute the proportions of observations that are misclassified.

2.2.2 Quadratic Discriminant Analysis

Quadratic Discriminant analysis (QDA) is used for heterogeneous variance-covariance matrices:

$$\mathbf{\Sigma}_i \neq \mathbf{\Sigma}_j \quad \text{for some} \quad i \neq j$$

The quadratic discriminant Score function is

$$s_i^Q(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{\Sigma_i}| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_i})^{\mathbf{T}}\mathbf{\Sigma_i^{-1}}(\mathbf{x}-\boldsymbol{\mu_i}) + \log p_i$$

We use again as in LDA the estimations of the unknown quantities

$$s_i^Q(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{S_i}| - \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^{\mathbf{T}}\mathbf{S_i^{-1}}(\mathbf{x}-\bar{\mathbf{x}}) + \log p_i$$

Some other classification methods are

- Logistic regression
- Support Vector Machines
- Classification Trees
- Random Forest
- Neural Networks

2.2.3 Example of LDA

In this example we will illustrate the LDA method by using the Fisher's Iris data set[3]. This dataset consists of 150 observations(flowers) of three different species (setosa,versicolor,virginica) and four variables(Sepal Length, Sepal Width, Petal Length, Petal Width). The purpose of Discriminant Analysis is to create a decision rule for discriminating the species.

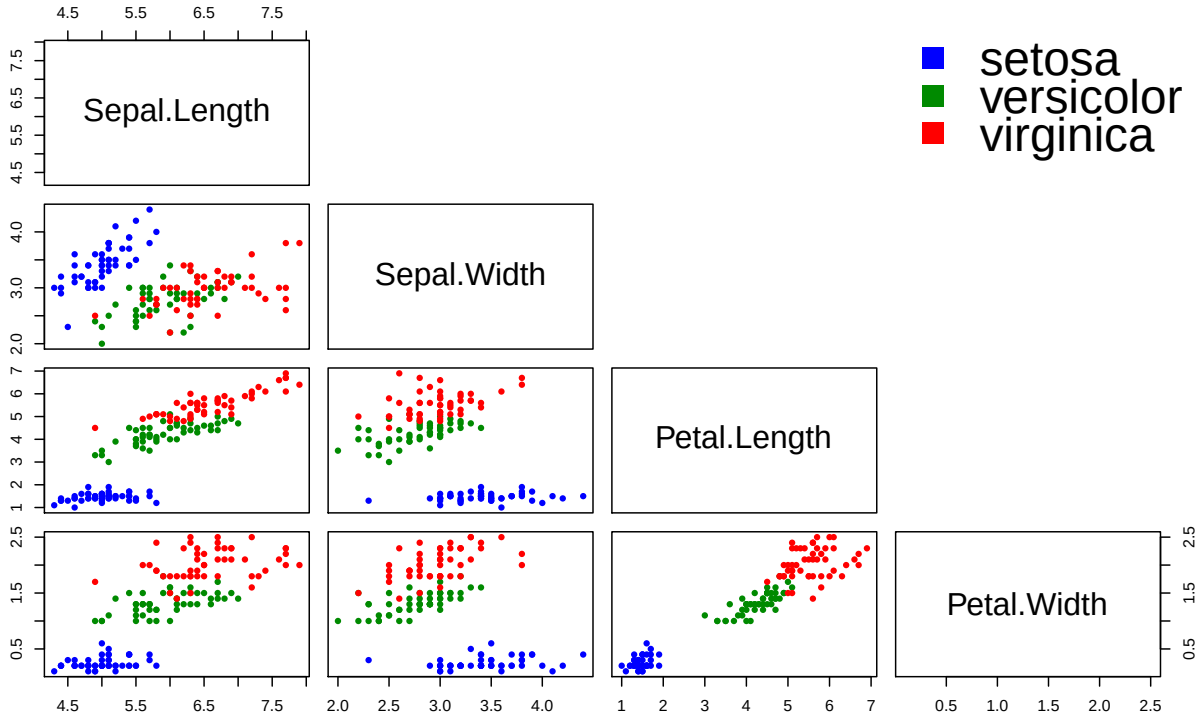


Figure 2.5: Scatter Plot of Iris

[3]Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7, Part II, 179–188.

The data were collected by Anderson, Ed.(1935). The irises of the Gaspe Peninsula, Bulletin of the American Iris Society, 59, 2–5.

From Table 2.2 we observe that the means of the species are different thus it is reasonable to use discriminant analysis for classification.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	5.01	3.43	1.46	0.25
versicolor	5.94	2.77	4.26	1.33
virginica	6.59	2.97	5.55	2.03

Table 2.2: Means of Iris Species

Using equal *prior probabilities* 1/3 the two Discriminant functions are :

$$Y_1 = 0.83 \times \text{Sepal.Length} + 1.53 \times \text{Sepal.Width} - 2.20 \times \text{Petal.Length} - 2.81 \times \text{Petal.Width}$$

$$Y_2 = 0.02 \times \text{Sepal.Length} + 2.16 \times \text{Sepal.Width} - 0.93 \times \text{Petal.Length} + 2.84 \times \text{Petal.Width}$$

In figure 2.6 we can see the data separation that was achieved using the Linear Discriminant Analysis method. In Figure 2.7 the red symbols are the misclassified for each pair of variables.

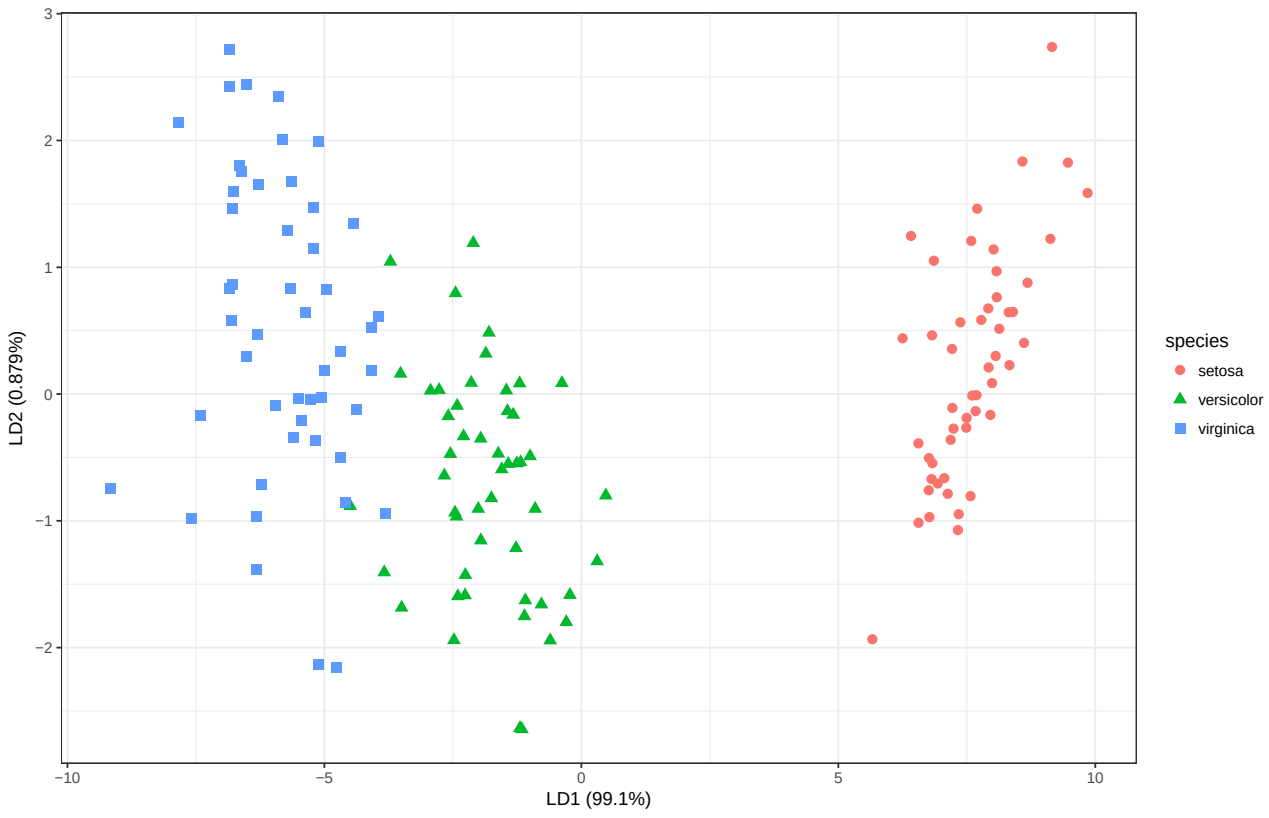


Figure 2.6: Data Separation

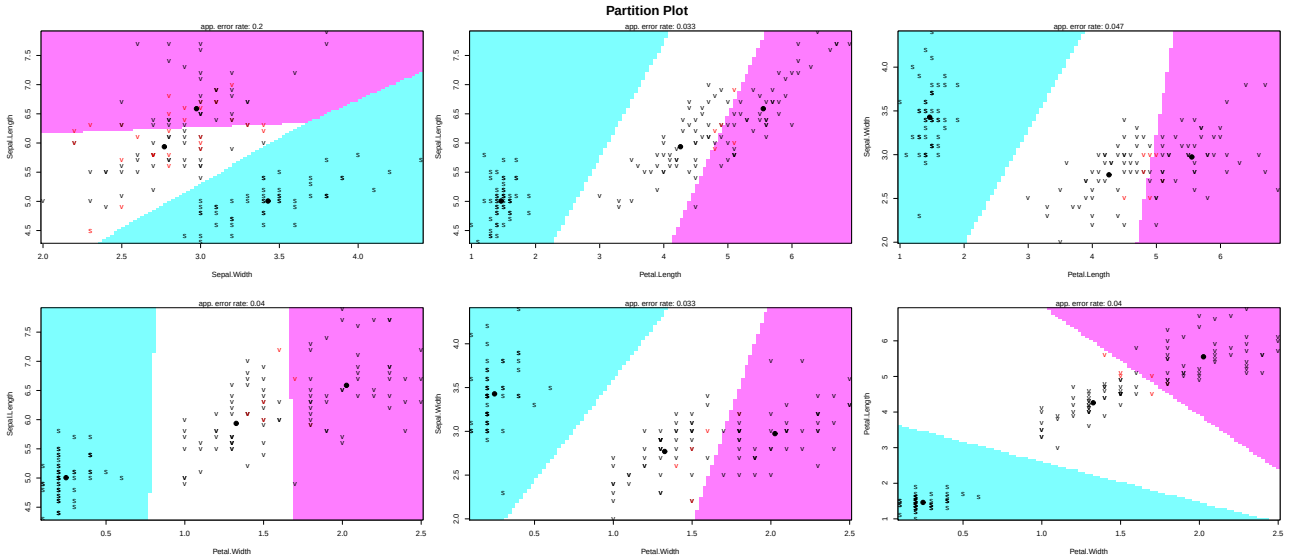


Figure 2.7: Partition Plots of Iris

The confusion matrix is:

Actual/Predicted	setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	48	2
virginica	0	1	49
total	50	49	51

Table 2.3: Confusion matrix of LDA

Here, we see that 50 of 50 (100%) setosa are expected to be correctly classified, 48 of 50 (96%) of versicolor were correctly classified and 49 of 50 (98%) of virginica were correctly classified.

2.3 Cluster Analysis

Clustering is an unsupervised method[4] of grouping observations, which are represented as vectors, in a way such that the observations in the same group (cluster) are more "similar" to each other than those in other groups. There are many clustering methods (algorithms), a common distinction is among **Hierarchical** and **Partitional** methods. More formally (Wagstaff, 2002) clustering algorithms seek to an organization P of a data set D that optimizes an objective function $f : P \to \mathbb{R}$. We will see later that these algorithms make use of a distance function $d : D \times D \to \mathbb{R}$, to measure the similarity or dissimilarity of two vectors.

2.3.1 Hierarchical Methods

There are two types of Hierarchical clustering methods, *agglomerative* and *divisive*

- **Agglomerative** algorithms (which are the most common used) start with each observation forming its own cluster, then clusters are successively merged, until a single cluster remains.

[4]Unsupervised methods aim to "learn" structures in the data. For example they aim to estimate the density function of data.

- **Divisive** algorithms start with all observations in a single cluster, which is divided to two clusters and successively each of these clusters is divided in two until each item is its own cluster.

The result of using Hierarchical clustering methods is the *dendrogram* that shows how the observations where successively merged or divided. As example we see in Figure 2.8 the dendrogram of iris data set.

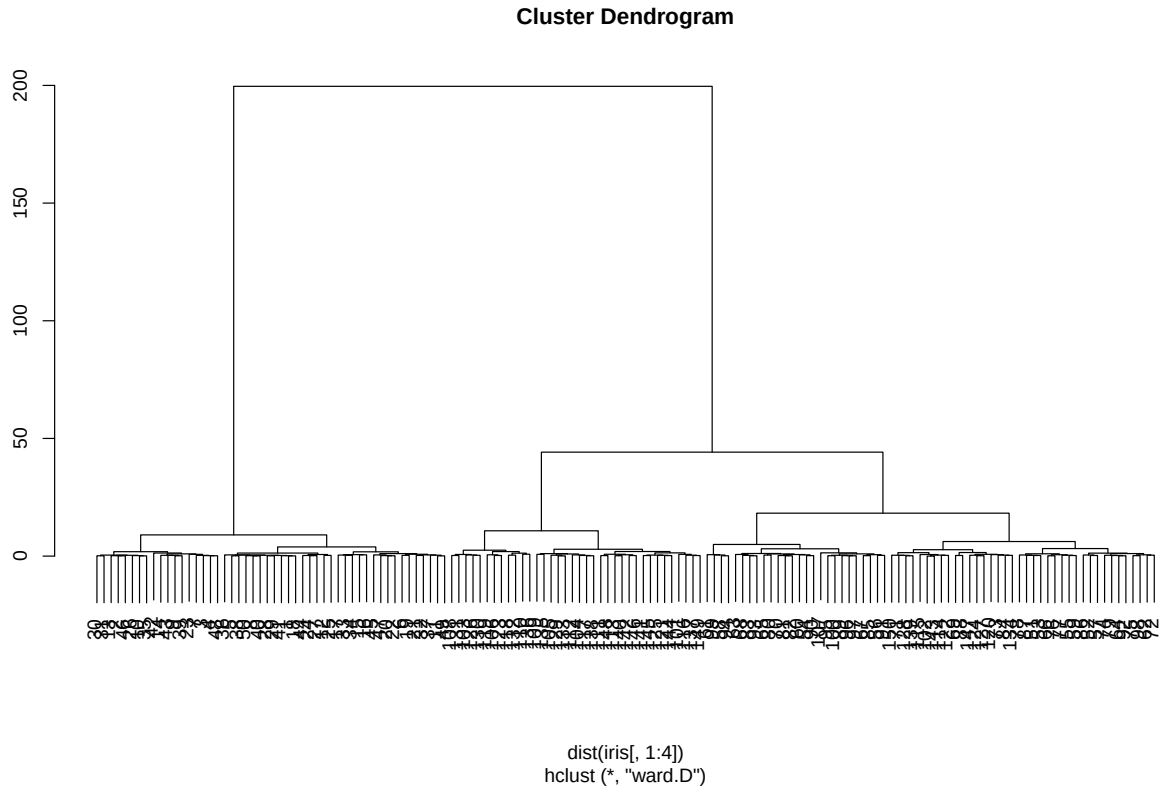


Figure 2.8: Dendrogram of Iris data set

Generally in clustering we do not have any information about the number of clusters that data can be divided. An easy way to choose the number of clusters is the dendrogram (e.g. in Figure 2.8 we can choose 3). To create the dendrogram we use *Measures of Association between observations* and *Measures of Association between Clusters*.

Measures of Association between observations:

- The most commonly used measure of association is the **Euclidean distance** which for p-dimensional vectors is defined as

$$d(\mathbf{X_i}, \mathbf{X_j}) = \sqrt{\sum_{k=1}^{p} (X_{ik} - X_{jk})^2} \tag{2.1}$$

  - The Euclidean distance is sensitive to the measurement units.A solution to this problem is to normalize the variables.
  - Variables with large absolute values can determine the distance between the observations.

- **Minkowski distance** is a generalization of the Euclidean distance

$$d(\mathbf{X_i}, \mathbf{X_j}) = \left[\sum_{k=1}^{p} |X_{ik} - X_{jk}|^q\right]^{1/q} \tag{2.2}$$

- **Manhattan distance** is defined as

$$d(\mathbf{X_i}, \mathbf{X_j}) = \sum_{k=1}^{p} |X_{ik} - X_{jk}| \tag{2.3}$$

  and it used in the case of outliers.

- **Canberra Metric** is defined as

$$d(\mathbf{X_i}, \mathbf{X_j}) = \sum_{k=1}^{p} \frac{|X_{ik} - X_{jk}|}{X_{ik} + X_{jk}} \tag{2.4}$$

- **Czekanowski Coefficient** is defined as

$$d(\mathbf{X_i}, \mathbf{X_j}) = 1 - \frac{2\sum_{k=1}^{p} \min(X_{ik}, X_{jk})}{\sum_{k=1}^{p}(X_{ik} + X_{jk})} \tag{2.5}$$

In general we can create measures of association which must satisfy the following properties[5]:

1. *Symmetry*
$$d(\mathbf{X_i}, \mathbf{X_j}) = d(\mathbf{X_j}, \mathbf{X_i})$$
2. *Positivity*
$$d(\mathbf{X_i}, \mathbf{X_j}) > 0 \quad \text{if} \quad \mathbf{X_i} \neq \mathbf{X_j}$$
3. *Identity*
$$d(\mathbf{X_i}, \mathbf{X_j}) = 0 \quad \text{if} \quad \mathbf{X_i} = \mathbf{X_j}$$
4. *Triangle inequality*
$$d(\mathbf{X_i}, \mathbf{X_k}) \leq d(\mathbf{X_i}, \mathbf{X_j}) + d(\mathbf{X_j}, \mathbf{X_k})$$

Measures of Association between Clusters or Linkage Methods[6]:

- **Single Linkage**: Is the distance between the closest members of two clusters

$$d_{12} = \min_{i,j} \; d(\mathbf{X}_i, \mathbf{Y}_j) \tag{2.6}$$

[5] https://onlinecourses.science.psu.edu/stat505/node/140
[6] https://onlinecourses.science.psu.edu/stat505/node/143

- **Complete Linkage**: Is the distance between the members that are furthest apart (most dissimilar)

$$d_{12} = \max_{i,j} \ d(\mathbf{X}_i, \mathbf{Y}_j) \tag{2.7}$$

- **Average Linkage**: This method involves looking at the distances between all pairs and averages all of these distances. This is also called UPGMA - Unweighted Pair Group Mean Averaging

$$d_{12} = \frac{1}{kl} \sum_{i=1}^{k} \sum_{j=1}^{l} d(\mathbf{X}_i, \mathbf{Y}_j), \tag{2.8}$$

  where k is the number of observations in cluster 1 and l is the number of observations in cluster 2.

- **Centroid Method**: This involves finding the mean vector location for each of the clusters and taking the distance between these two centroids.

$$d_{12} = d(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \tag{2.9}$$

- **Ward's Method**: This method does not directly define a measure of distance between two points or clusters. It is rather an ANOVA based approach. At each stage, those two clusters merge, which provides the smallest increase in the combined error sum of squares from one-way univariate ANOVAs that can be done for each variable with groups defined by the clusters at that stage of the process.

2.3.2 Partitioning Algorithms

Partitioning algorithms are algorithms that divide the hyperplane in such a way that every observation belongs to one cluster. In partitioning clustering algorithms the number of clusters is considered to be known. The most common used partitioning algorithm is the **K-means Algorithm**, an iterative procedure that begins with K initial centroids (usually randomly selected) and assigns each observation to the cluster where the observation has the minimum distance from its centroid. This procedure is repeated until convergence (the centroids can not change). K-means is a method that minimizes within-cluster variation.

The name K-means for this algorithm was first used by James MacQueen in 1967, though the idea existed already. The algorithm was first proposed by Stuart Lloyd in 1957 and E.W. Forgy who essentially have published the same method.
We will give the K-means algorithm that was found by James MacQueen[7]

General form of K-means Algorithm

1. Choose K initial centroids

2. Assign each observation to the cluster with the minimum distance from its center

3. Re–calculate the centroids of the new clusters

4. If the centroids do not change, stop the procedure, else go back to step 2

In step 1 the K initial centroids are randomly chosen (from the data) and at the assignment step (step 2) we use the Euclidean distance (formula (2.1)). Care though must be taken if the measurement units of the variables are not the same. If this is the case, we have to normalize them and then start the procedure.

Algorithm 1 K-means MacQueen (1967)

Randomly initialize K cluster centrers $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \ldots, \boldsymbol{\mu}_k$

ASSIGNMENT STEP
for $i \leftarrow 1$ to n **do**
  assign X_i to cluster k where minimizes the objective function
  $\|\mathbf{X_i} - \boldsymbol{\mu_k}\| = \sqrt{\sum_{j=1}^{p}(X_{ji} - \mu_k)^2}$
end for

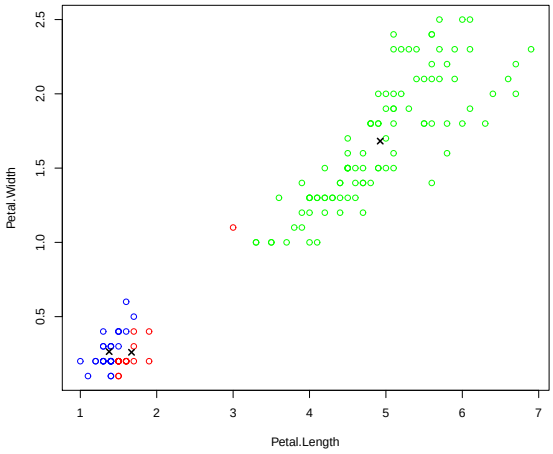
UPDATING CENTROIDS STEP
for $i \leftarrow 1$ to K **do**
  calculate $\boldsymbol{\mu}_k =$ **mean of observations assigned to cluster k**
end for

Iterate between Steps (1) and (2) until convergence
return partition $\{C_1, \ldots, C_k\}$

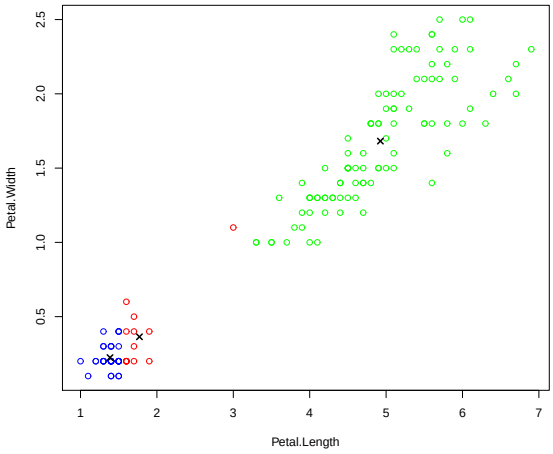
Now we will illustrate an example of K-means algorithm on the iris dataset which consists of 150 observations, using only the variables Petal Length and Petal Width, in order to have better visualization of how the species setosa is well separated from the versicolor and virginica species.

[7]MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281–297. MR 0214227. Zbl 0214.46201. Retrieved 2009-04-07.

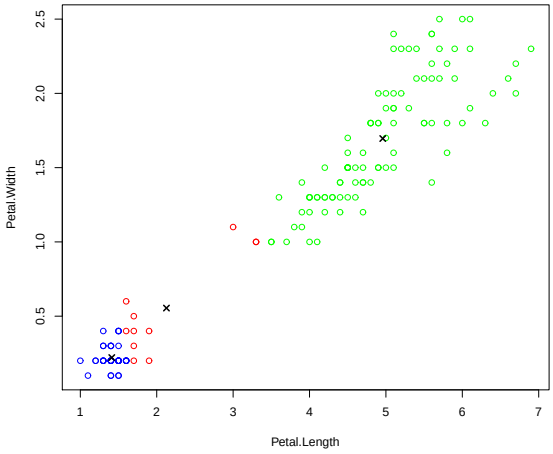
Figure 2.9: K-means Algorithm Procedure



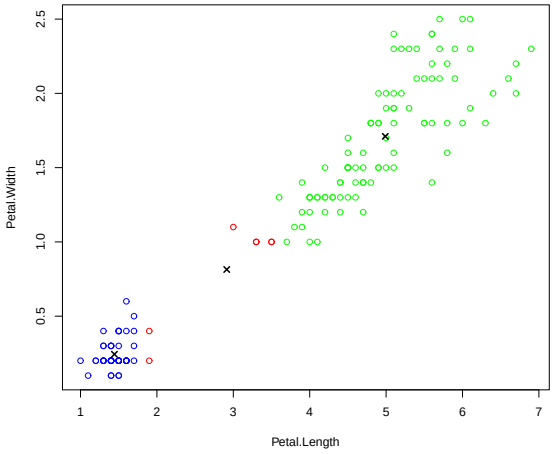
(a) Iteration Number = 1



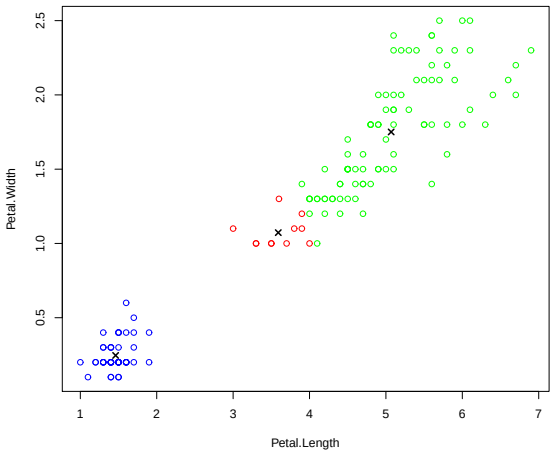
(b) Iteration Number = 2



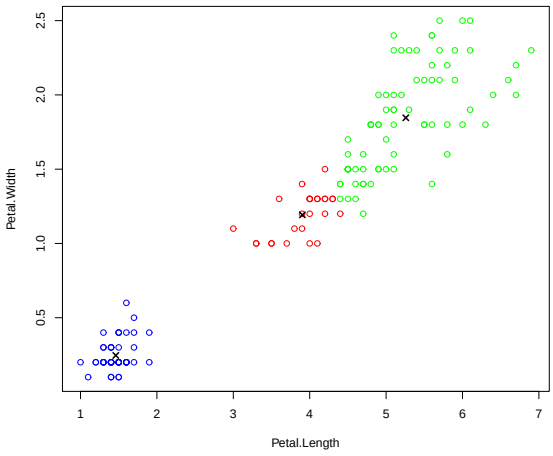
(c) Iteration Number = 3



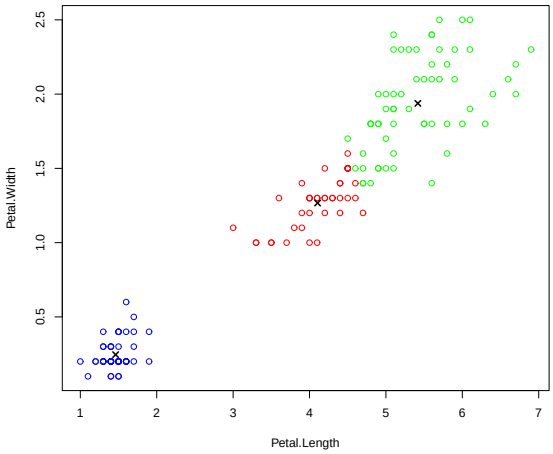
(d) Iteration Number = 4



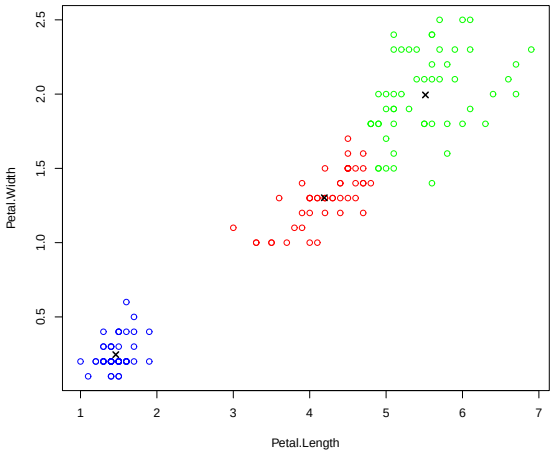
(e) Iteration Number = 5



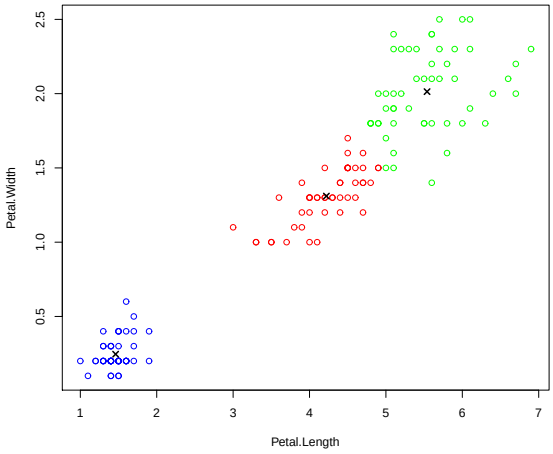
(f) Iteration Number = 6



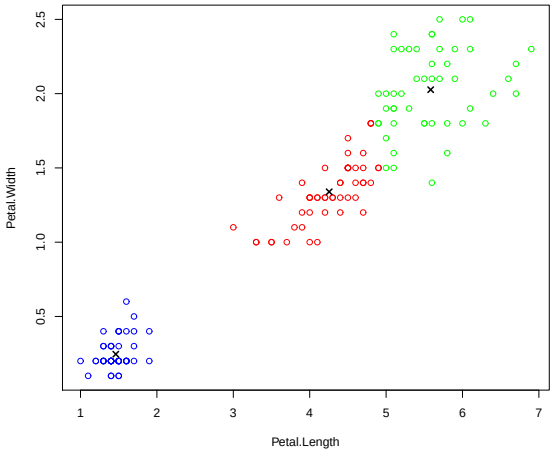
(g) Iteration Number = 7



(h) Iteration Number = 8



(i) Iteration Number = 9



(j) Iteration Number = 10

K-means is a fast algorithm that can deal with large data sets in contrast with Hierarchical methods. The final partition obtained from K-means is strongly dependent to the initial selection of the centroids, thus we have to run K-means several times to obtain a good partition of the data set. Another weakness of K-means is that we have to choose the number of clusters in advance. It is not always clear how to choose K, although as we will see below there have been proposed some empirical methods to obtain a good estimation of K.

Choosing the Optimal Number of K

The most common issue in partitioning algorithms is choosing the optimal number of clusters. A popular approach is to look the dendrogram but this approach is subjective and it is necessary to run an hierarchical method first. There is a variety of methods for choosing the number of clusters in partitioning algorithms. The most popular among these are the *elbow method*, the *silhouette method* and the *gap statistic method*.

Elbow method (Scree plot): generally the basic idea of K-means algorithm, is to minimize the total within cluster variation, or total within cluster sum of square (total WSS). The total WSS is plotted

as a function of the number of clusters. The elbow method suggests to choose the value of K such that the addition of another cluster does not improve further the total WSS. Sometimes the Elbow method is ambiguous and then we have to choose an other method.

Silhouette Analysis[8,9]: The silhouette Analysis measures how similar an object is to the cluster that is assigned to compared to other clusters. The **Silhouette plot** displays how close each point in one cluster is to points in neighboring clusters. The silhouette width S_i for each observation i is calculated as follows:

1. For each observation i, calculate the average dissimilarity a_i between i and all other points of the cluster which i belongs.

2. For all other clusters C, to which i does not belong, calculate the average dissimilarity $d(i, C)$ of i to all observations of C. The smallest of these $d(i, C)$ is defined as $b_i = \min_C d(i, c)$. The value of b_i is the dissimilarity between i and its closest cluster.

3. The silhouette width of observation i is given by the formula:

$$S_i = (b_i - a_i)/\max(a_i, b_i)$$

Observations with a large S_i(almost 1) are well clustered, observations with a small S_i lies between two clusters and observations with negative S_i are probably placed in the wrong cluster.

Gap statistic[10,11]: Gap statistic compares the pooled within cluster variation for different values of K with their expected values under null reference distribution of the data.

$$\text{Gap}_n(k) = E_n^*[\log(W_k)] - \log(W_k)$$

where $W_k = \frac{1}{2n_r} D_r$ and $D_r = \sum_{i,i' \in C_r} \sum_j (x_{ij} - x_{i'j})$ and E_n^* is the expectation under a sample of size n from the reference distribution.
The estimated $\hat{k}$ will be the value which maximizes $\text{Gap}_n(k)$, after taking the sampling distribution into account. This means that the clustering structure is far away from the random uniform distribution of points. The algorithm can be summarized in the following steps

1. Cluster the data for $k = 1, \ldots, k_{max}$ and compute the total within cluster variation W_k.

2. Generate U reference data sets with a uniform distribution. Cluster each data set for all the number of clusters $k = 1, \ldots, k_{max}$ and compute the total within cluster variation $W_u k$.

3. The $E_n^*[\log(W_k)]$ is $\frac{1}{B} \sum_{u=1}^{U} \log(W_{ku}^*)$. Compute Gap_k and the standard deviation of the statistics.

4. Choose the $\hat{k}$ as the smallest value of k such the gap statistic is one standard deviation of the gap at $K + 1$:

$$\text{Gap}_{(k)} \geq \text{Gap}_{(k+1)} - sd_{k+1}.$$

[8] Kassambara, A. (2017). Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning (Vol. 1). STHDA.

[9] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20, 53-65.

[10] Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2), 411-423.

[11] Kassambara, A. (2017). Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning (Vol. 1). STHDA.

2.3.3 Example of K-means

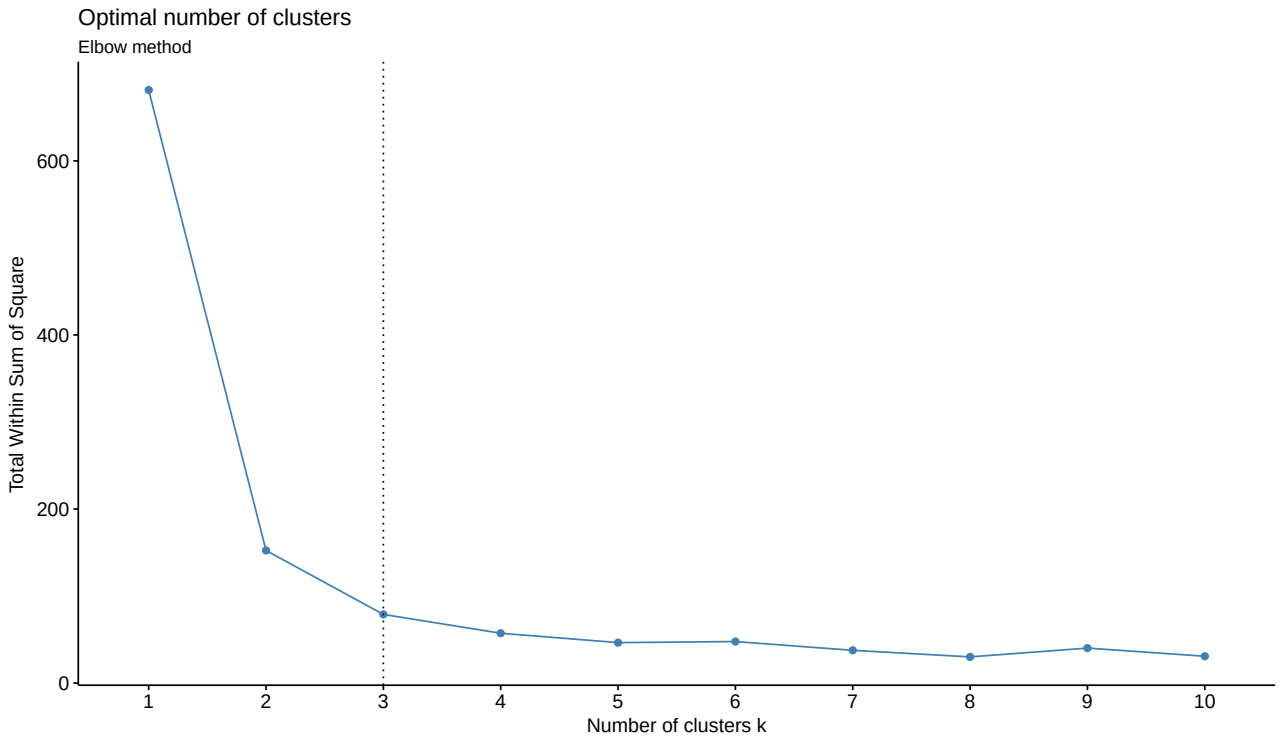
In this example we will illustrate the K-means algorithm in the iris dataset. We will not use the information of the (three) species but we will try to extract this information by clustering the data set and finding the optimum number of clusters by using the methods described above.
The data set is in this form

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.10	3.50	1.40	0.20
2	4.90	3.00	1.40	0.20
3	4.70	3.20	1.30	0.20
4	4.60	3.10	1.50	0.20

Table 2.4: Iris data set without labels

We run the K-means algorithm for $k = 1, 2 \ldots, 10$ and from the Elbow method we can choose k=3.

Figure 2.10: Elbow Method



We run the K-means algorithm 25 times with $k = 3$, in order to find the partition with the lowest within cluster variation. The three clusters consisted of 50, 38 and 62 observations respectively. The corresponding cluster means are given in Table 2.5.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.01	3.43	1.46	0.25
2	6.85	3.07	5.74	2.07
3	5.90	2.75	4.39	1.43

Table 2.5: Means of Clusters

As we can see from the confusion matrix given in Table 2.6 below, the setosa species belongs to the first cluster the versicolor belongs to the second cluster and the virginica to the third cluster.

	setosa	versicolor	virginica
1	50	0	0
2	0	2	36
3	0	48	14

Table 2.6: Confusion matrix of Clustering

The algorithm wrongly classified two observations to versicolor and fourteen to virginica. In chapter 4 we will illustrate clustering algorithms using background knowledge that can increase clustering accuracy.

The visualization of the clusters is not so easy due to the high number of the dimensions (variables). In section 2.1 we discussed how we can reduce the dimension of a data set by keeping a high proportion of the explained variance. In Figure 2.11 we plot the clusters as obtained by the K-means using the first two principal components

Figure 2.11: Clusters plotted using the first two pcas

2.3.4 Assumptions of K-means

[12] K-means considers two assumptions:

1. The clusters are spherical

2. The clusters are of similar size

Now imagine a data set that clusters can clearly be identified but K-means cannot correctly identify them. As an example we will create a dataset consisting of two non centric circles depicted in Figure 2.12.

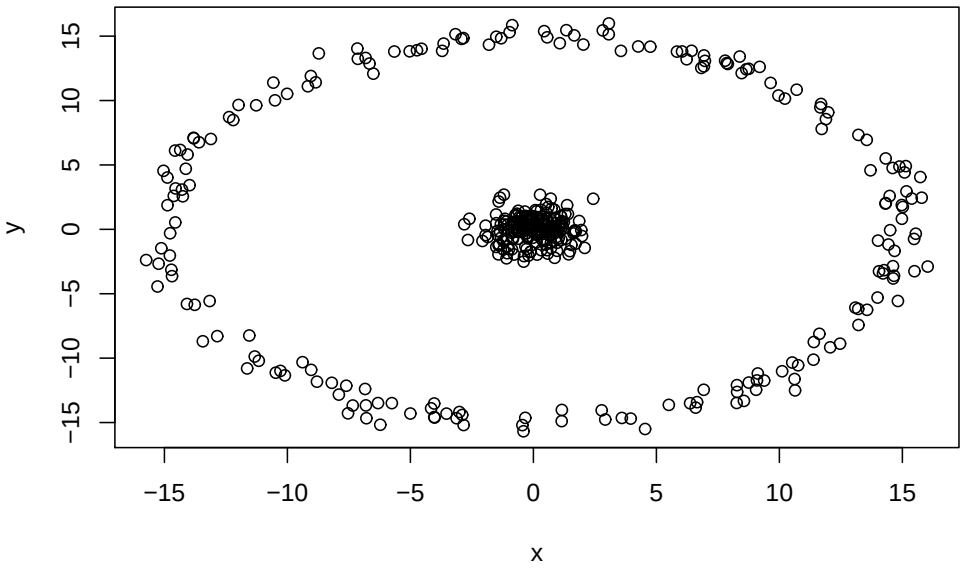


Figure 2.12: Data set of two non centric circles

It's easy to see that there are two clusters in the data set. We run k-means for $k = 2$ and the clustering isn't good.

[12] https://www.r-bloggers.com/exploring-assumptions-of-k-means-clustering-using-r/

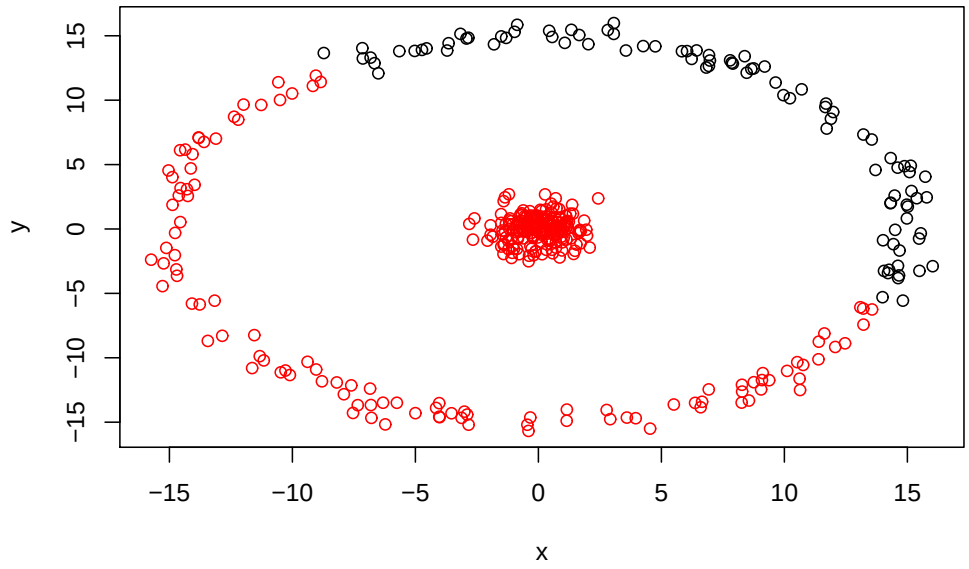


Figure 2.13: Inaccurate clustering

Maybe if use another clustering method (e.g. EM algorithm) the problem will be solved. A simple way to make K-means accurate in this example is to transform our data into polar coordinates and plot them.

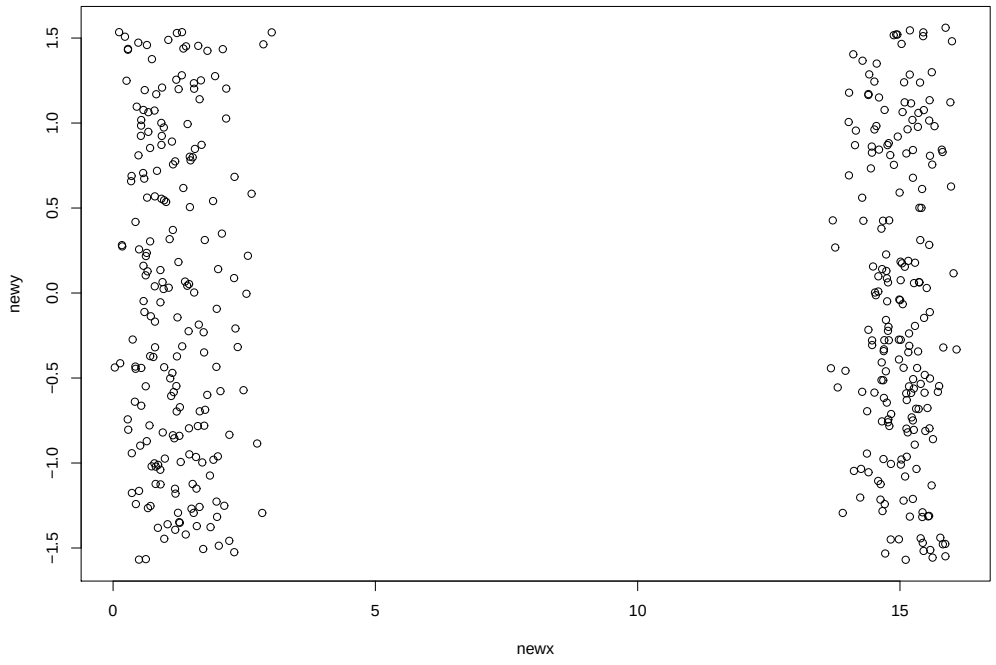


Figure 2.14: Transformed data

Now it is easier for the k-means to separate the data set in two clusters

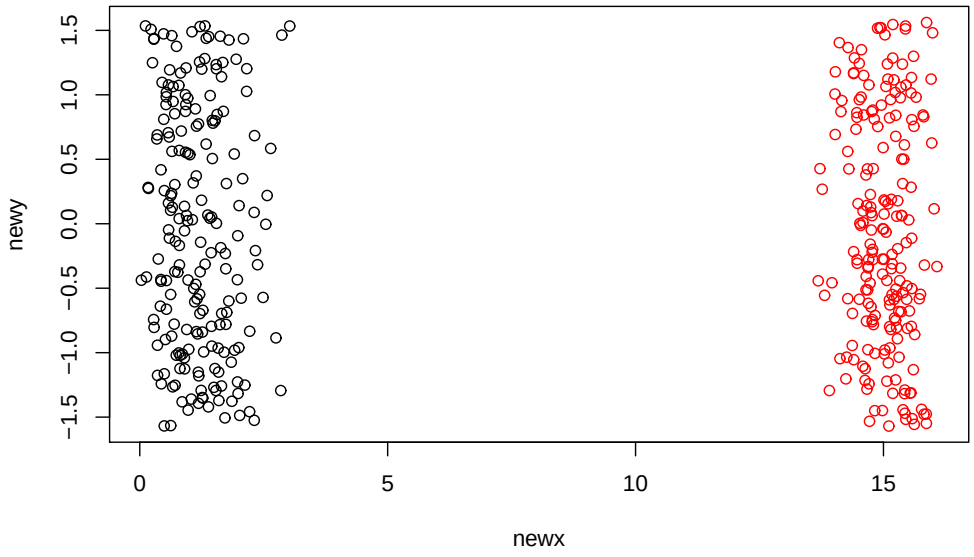


Figure 2.15: Clustering of transformed data

Now using the inverse transformation we can see how the initial data set is partitioned in two clusters

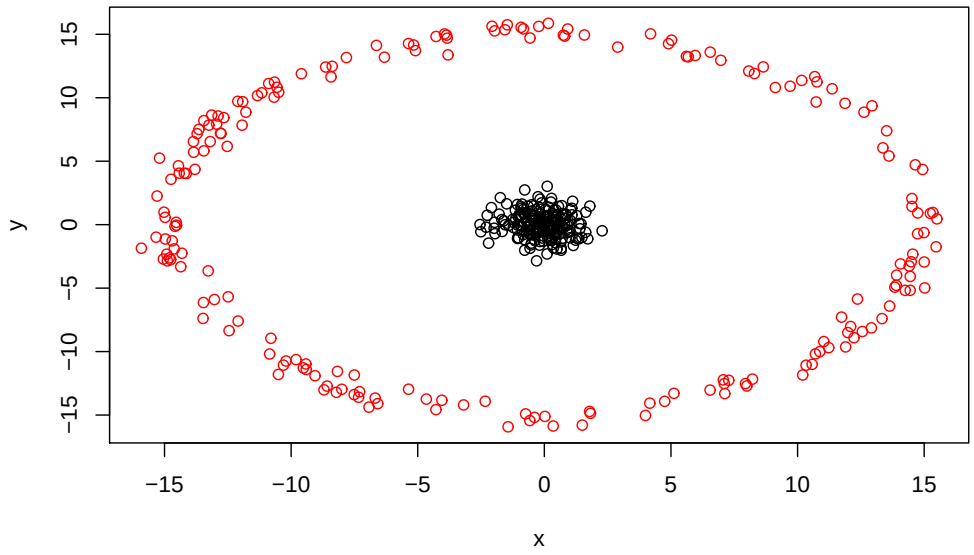


Figure 2.16: Clustering of spherical data set

Chapter 3

Missing Data

Missing data is a common problem that almost every statistician has to deal with. The two most common approaches is either to omit the observations with missing data or to replace them with estimated values. There are a lot of missing data methods, though their usefulness depends on the mechanisms that generate the missingness. This is a prerequisite chapter for the following two, thus we will not fully describe missing data methods, though we will only cover the necessary methods for quantitative variables. We will describe some of the simplest (historical) imputation methods such as mean imputation and two algorithmic methods, namely imputation using kNN (k Nearest Neighbors) and miss Forest (missing Forest).

Notation

X^{com}: the set of observations that are complete (without missing values) for all the variables.
X^{mis}: the set of observations with missing values .
$\mathbf{x}_i^{com}$: the i observation, where the indicator *com* shows that are complete for all the variables .
$\mathbf{x}_i^{mis}$: the i observation, where the indicator *mis* shows that is missing.
V_j^{com}: the j variable, where the indicator *com* shows that all elements are recorded.
V_j^{mis}: the j variable, where the indicator *mis* shows that some elements are missing.
x_{ij}^{com}: the ij element of X, where the indicator *com* shows that x_{ij} is present.
x_{ij}^{mis}: the ij element of X, where the indicator *mis* shows that x_{ij} is missing.

3.1 Mechanisms that generate missing data

As we have described in Chapter 1 multivariate data sets can be represented by rectangular matrices, with the rows representing the observations and the columns representing the variables. The data matrix is denoted by X, where each element of X is usually a real number, $x_{ij} \in \mathbb{R}$ and denotes the values of the j^{th} variable within i^{th} observation. A common problem is that for systematic or non systematic reasons some of the elements x_{ij} are not observed. These elements are called missing values or missing data. For the data set X with missing values we define the missing data indicator matrix $M = (m_{ij})$, where $m_{ij} = 1$ if the element x_{ij} is missing and $m_{ij} = 0$ if x_{ij} is present:

$$m_{ij} = \begin{cases} 1, & \text{if } x_{ij} \text{ is missing} \\ 0, & \text{if } x_{ij} \text{ present.} \end{cases} \tag{3.1}$$

The matrix M defines the pattern of missing data.

Table 3.1 is an example of a data set with missing values. The elements that are missing are x_{13}, x_{53} and x_{63} from the Variable 3 and $x_{14}, x_{34}, x_{44}, x_{64}$ and x_{104} from the Variable 4.

	Variable 1	Variable 2	Variable 3	Variable 4
1	5.10	3.50	·	·
2	4.90	3.00	1.40	0.20
3	4.70	3.20	1.30	·
4	4.60	3.10	1.50	·
5	7.00	3.20	·	1.40
6	6.40	3.20	·	·
7	6.90	3.10	4.90	1.50
8	6.30	3.30	6.00	2.50
9	5.80	2.70	5.10	1.90
10	7.10	3.00	5.90	·

Table 3.1: Data set with missing values

The corresponding indicator missing data matrix is

$$M = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Some missing data methods can be applied in every missing data pattern but others can be used only in specific missing data patterns. For further reading see Little & Rubin (2002, Chapter 1).

The most crucial question when we deal with missing data is if they occur randomly or in a systematic way. The important role of the mechanism that generates missing data was introduced in the theory of Rubin (1976), where the indicators of missing values are treated as random variables, thus we assign them a distribution. Following Little & Rubin (2002) the missing data mechanism is characterized by the conditional distribution of M given the data set X, $f(M|X, \phi)$, where ϕ are the unknown parameters of the distribution. The types of missing data are *missing completely at random* (MCAR), *missing at random* (MAR) and *missing not at random* (NMAR).

- **MCAR** Values in a data set are **Missing Completely at Random** if the reason(s) that missing values are generated does not depend on any of the values of the elements on the data X missing or observed. We write that

$$f(M|X, \phi) = f(M|\phi) \quad \forall X, \phi. \tag{3.2}$$

  Note that in the case of MCAR the full observed values are in effect a random sample of the data set.

- **MAR** Values in a data set are **Missing at Random** if they are not missing completely random, though their missingness is related only with the observed values (X^{com}) and not with the missing (X^{mis}), we write that

$$f(M|X, \phi) = f(M|X^{com}, \phi) \quad \forall X^{mis}, \phi. \tag{3.3}$$

- **NMAR** Values in a data set are **Not Missing at Random** if the distribution of M is related with values in X^{mis}.

Suppose that we have a data set of n observations where k observations are present and the rest $n-k$ are missing. A simple approach is to use only the observed data, thus we decrease the sample size from n to k. Suppose now we want to estimate the mean of the population using the observed subset of the data and our data are normally distributed. If the data are MCAR we can estimate the mean of data using the sample mean of the observed data, but if the data are NMAR the estimation will be biased. For example in Figure 3.1 if the missing values are all the values from 0 to 2 along x-axis, the distribution will not be symmetric anymore and the estimation will be biased downward.

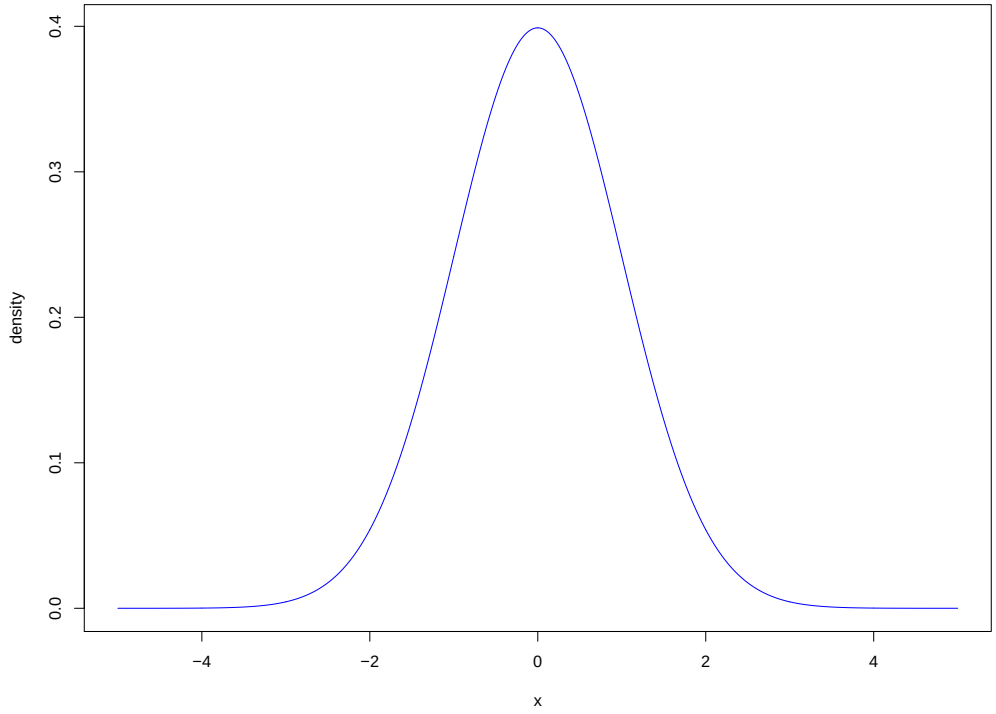


Figure 3.1: Univariate normal distribution

In the case of MCAR the probability that x_i, $\forall x_i$ is missing is equal to $Pr(m_i = 1|x_i, \phi) = 0.5$, $\forall x_i$ and in the case of NMAR the probability of x_i is equal to $Pr(m_i = 1|x_i, \phi) > 0.5$, if $x_i > 0$ and $Pr(m_i = 1|x_i, \phi) < 0.5$ if $x_i < 0$.

3.2 Methods for Missing Data

Following Little & Rubin (2002) missing data methods can be divided in the next four categories:

1. **Complete case Analysis**. In complete case analysis we use only the observations that are recorded for all the variables. This is a simple approach and sometimes is useful when a small amount of data is missing, though the analysis in usually inefficient and can lead in serious biases. Suppose we have the data set that described by the Table 3.1, the complete cases are only 4 (40%) of the data, thus we lose valuable information.

2. **Weighting Procedures**. Suppose that we want to estimate the mean of a population and we have a univariate sample of this population. Each observation i has been selected with probability π_i. The population mean can be estimated by the Horvitz-Thomson[1] estimator

$$\hat{\mu} = \frac{\sum \frac{x_i}{\pi_i}}{\sum \frac{1}{\pi_i}}. \tag{3.4}$$

In the case of missing values equation 3.4 can be written in the following form

$$\tilde{\mu} = \frac{\sum \frac{x_i}{\pi_i \hat{p}_i}}{\sum \frac{1}{\pi_i \hat{p}_i}}, \tag{3.5}$$

where $\hat{p}_i$ is an estimation of the probability of response for observation i.

3. **Imputation Methods**. Imputation methods fill in the missing data by estimated values. The most common used imputation methods are *hot deck* imputations where the missing values are replaced by recorded values in the data set. *Mean imputation* is the easiest approach where each missing value is replaced by the mean of the observed values for each variable. *Regression imputation*, where missing values are replaced by predicted values from a regression model. Other approaches are model-based imputation methods and multiple imputation methods where we impute the missing data set m times, where imputed values are drawn for a distribution, then we analyze each of the m completed data sets and at the end we integrate the analysis results into one final result.

4. **Model-Based Methods**. These methods define a model for the non missing data where the parameters of the model are estimated by e.g. the maximum likelihood. The analysis of the data is based on the likelihood of this model.

3.3 Imputation Methods

3.3.1 Hot Deck Imputation

In hot deck imputation we replace the missing values with values from "similar" presented observations. For example following Enders (2010) consider a population survey that some of the respondents refuse to disclose their income. The hot deck imputation method classifies the respondents into categories based on demographic characteristics such as gender and age. Then the missing values are replaced with random recorded draws from the income distribution of the observations that are classified in the same category.

3.3.2 Cold Deck Imputation

In cold deck imputation we replace missing values with items from an external source, e.g. following a previous example, values from previous population surveys or in a longitudinal setting the last recorded value.

3.3.3 Mean Imputation

Suppose we have the data set of Table 3.1, where variables 3 and 4 have missing values. In mean imputation we calculate the sample mean of each variable ($\bar{V}_i$) with missing values using only the recorded elements (X^{com}) and then we replace the missing values with the corresponding $\bar{V}_i$ value.

[1]Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. Journal of the American statistical Association, 47(260), 663-685.

For example in the Table the missing values in Variable 3 will be replaced by the value 4.35 and in variable 4 by the value 1.5. The problem with mean imputation is that decreases the variance of the data, the covariances and consequently the correlations. To fix this problem we can add to each imputed value an error term drawn randomly from a distribution with mean 0 and variance equal to the variance of the observed values, for example an error term e_i that $e \sim N(0, var_{com})$.

3.3.4 Regression Imputation

In regression imputation (sometimes referred as conditional mean imputation) the missing values are estimated from a regression equation. The basic idea is to use the information of the complete cases to estimate the missing values. We estimate a regression model only for the complete cases and then we use the regression equation to predict the missing values. Specifically we estimate a model for the **complete cases** given by the following equation:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi} + \epsilon_i, \tag{3.6}$$

and if case k is missing, is replaced by the following value:

$$\hat{y}_k = \hat{\beta}_0 + \hat{\beta}_1 x_{1k} + \hat{\beta}_2 x_{2k} + \ldots + \hat{\beta}_p x_{pk}. \tag{3.7}$$

3.3.5 Stochastic Regression Imputation

Stochastic Regression imputation uses regression equations to estimate missing values by adding a residual term. This residual term ϵ_k^* can be a random draw of sample residuals from the complete cases. Like simple regression imputation we estimate a model given by the equation (3.6) and the missing case k is replaced by the following value:

$$\hat{y}_k = \hat{\beta}_0 + \hat{\beta}_1 x_{1k} + \hat{\beta}_2 x_{2k} + \ldots + \hat{\beta}_p x_{pk} + \epsilon_k^*. \tag{3.8}$$

3.3.6 Predictive Mean Matching

In predictive mean matching we use the complete data to fit a regression equation and we predict the element k as before. The next step is to find for this variable the closest recorded value to the predicted value of k element and use it to replace the missing value. For example if in Table 3.1 the predicted value for the missing element x_{31} is 4.7 we replace this element with the value 4.9.

Note that for the last three methods all $x's$ for the estimated equations must present. In the case that some of the $x's$ do not present we can use **Chain Equations**. The idea of chain equations is to impute first the variable with the smallest amount of missing values using an regression model that is estimated from the existing complete cases. Then the imputed data are treated as complete data in order to impute the variable with the smallest amount of missing values as before. We repeat this procedure sequentially until all the variables are imputed.

All methods that described until now provide easy solutions for the user but they all seem to be inaccurate, except from the idea of chain equations that can be considered as an algorithmic procedure. The following two algorithmic multivariate procedures are referred as two of the most accurate imputation methods.

3.3.7 Imputation using kNN Algorithm

K Nearest Neighbors (kNN) is a multivariate non parametric method that can be used for both classification and regression. The idea of this method is to identify the k closest observations to the new

(input) observation using a distance measure (e.g. Euclidean distance in case of continuous variables) and classify the new observation to the majority. In kNN regression the algorithm returns the mean vector of the k nearest observations. The values of the nearest neighbors can also be weighted by e.g. a value inversely to the distance of each observation, $w_i = \frac{1}{dist_i}$ for each neighbor i.

The idea of **KNNimpute algorithm**[2] is to find the k most "similar" non missing observations, $\mathbf{x}_i^{com} \in X^{com}$, $i = 1, \ldots, k$, to each missing observation, $\mathbf{x}_j^{mis} \in X^{mis}$, and then impute each missing element (x_{ij}^{mis}) by a weighted average of the k corresponding elements of X^{com}. The KNNimpute algorithm uses the non missing observations, $\mathbf{x}_i^{com} \in X^{com}$, to identify the nearest neighbors, only for the recorded variables for each $\mathbf{x}_j^{mis} \in X^{mis}$ separately. Suppose that we want to impute the missing values for the Table 3.1 with $k = 2$ (using the 2 nearest neighbors), we will use the observations 2, 7, 8 and 9 ($\mathbf{x}_2, \mathbf{x}_7, \mathbf{x}_8$ and $\mathbf{x}_9$) to find them, due to the fact that they are non missing. For example if we want to impute the missing element $x_{5,3}^{mis}$ we will use the observations $\mathbf{x}_2, \mathbf{x}_7, \mathbf{x}_8$ and $\mathbf{x}_9$ for the variables V_1, V_2 and V_4 that are non missing for this observation.

Algorithm 2 KNNimpute

1: For all $\mathbf{x}_j^{mis} \in X^{mis}$. Compute the Euclidean distance between $\mathbf{x}_j^{mis}$ and all the $\mathbf{x}_i^{com} \in X^{com}$, using only the variables that **are not missing** for $\mathbf{x}_j^{mis}$.
2: Impute the missing values of each $\mathbf{x}_j^{mis}$ by the weighted average of the k closest corresponding elements.

Note that the weighted average is given by the equation

$$x_{ij}^{mis} = \frac{\sum_{k=1}^{K} w_k x_{kj}}{\sum_{k=1}^{K} w_k}, \tag{3.9}$$

where $w_k = \frac{1}{dist(\mathbf{x}_j^{mis}, \mathbf{x}_k)}$, computed only for the non missing variables, and $k = 1, \ldots, K$

Choice of k

For the choice[3] of k Hastie et al. (1999) suggest a simulation on the X^{obs} set. We use the full observed data by creating the same missing pattern (same proportion of missing observations and same proportion of missing elements), then we run the algorithm for several k and choose the one with the minimum error (difference between real and imputed values).

3.3.8 Imputation using Random Forest

Random Forest[4] like kNN is one of the most accurate (non parametric) multivariate methods that can be used for either classification and regression. As we have mentioned in the introduction of this Chapter we will describe methods only for continuous variables, thus we will use Random Forests only for regression.

[2] Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., ... & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. Bioinformatics, 17(6), 520-525.

[3] Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., & Botstein, D. (1999). Imputing missing data for gene expression arrays.

[4] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Regression Trees

Following James et al. (2014)[5] Regression Trees[6] divide the prediction space $(V_1, V_2, \ldots, V_p)$ into J high-dimensional rectangles where are denoted as regions $R_1, \ldots, R_J$. The prediction $\hat{y}_i$ for every observation $\mathbf{x}_i$ that $\mathbf{x}_i \in R_j$ is the mean of the values in R_j.

Bagging

Instead of having a single regression we can take repeated bootstrap[7, 8] samples of the data set. The idea is to generate N bootstrapped data sets (from the initial data set) to create N regression trees. For every regression tree we get a prediction at a point $\mathbf{x}$ as before, we denote this prediction as $\hat{f}^n(x)$ for each tree $n = 1, \ldots, N$. The the prediction of the point $\mathbf{x}$ is the average of all these predictions and is denoted as

$$\hat{f}_{bagging} = \frac{1}{N} \sum_{n=1}^{N} \hat{f}^n(x). \tag{3.10}$$

Bagging[9] is a method that improves the accuracy of the simple regression tree.

Random Forests

The idea of Random Forests is the same as Bagging but now the trees will be a little more uncorrelated. We build again a number of regression trees but now for a random selection of m, $m < n$ predictors (variables). Generally the number of predictors is approximately $\sqrt{p}$ where p is the number of variables. This method improves a little more the accuracy of bagging.

Miss Forest

Miss Forest[10] (miss is referred to missing values) is a non parametric imputation method that can be used either for continuous and categorical variables. The idea is to fit a random forest for the observed values and then make predictions for the missing values. More specifically the first step is to fill in the missing values with the mean of the corresponding variable (mean imputation), then we fit a random forest for the fully observed values (X^{obs}) and we predict the missing values by this model. We first impute the variables with the smaller amount of missing values and in each step the data matrix is updated. When all variables are imputed we have the imputed matrix for the first iteration of the algorithm and we denote it as X^{imp}_{new}. This imputation procedure is repeated until the stopping criterion is met. The stopping criterion is met when the difference between the new imputed data and the previous imputed data increase for the first time, where is defined as

$$\Delta_N = \frac{\sum_{j \in N} (X^{imp}_{new} - X^{imp}_{old})^2}{\sum_{j \in N} (X^{imp}_{new})^2}. \tag{3.11}$$

More details are given to the algorithm below.

[5] James, G., Witten, D., & Hastie, T. (2014). An Introduction to Statistical Learning: With Applications in R.

[6] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.

[7] is a method for sampling with replacement where the samples are the same size with the initial

[8] Efron, B., & Tibshirani, R. J. (1994). An introduction to the bootstrap. CRC press.

[9] Breiman, L. "Bagging predictors." Machine learning 24.2 (1996): 123-140.

[10] Stekhoven, D. J., & Bühlmann, P. (2011). MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics, 28(1), 112-118.

Algorithm 3 miss Forest

1: Fill in the missing values with the mean of the corresponding variable
2: Sort the indices of the variables in increasing amount of missing values and store them in a vector **k**
3: **While** not γ (the stopping criterion)
4: **for** s in **k**
5: Store previously imputed matrix to X_{old}^{imp}
6: Fit a random forest for the values in $Y^{obs} \sim X^{obs}$
7: Predict the values the values in Y^{mis} using X^{mis}
8: Update the imputed matrix X_{old}^{imp}, using the predicted Y^{mis}
9: **end for**
10: Update γ
11: **end while**
12: **return** the imputed matrix X^{imp}

Note that in miss Forest there is no parameter that need to be specified compared to kNN imputation algorithm.

3.4 Experimental Results

For the experimental results we use the Iris data set[11] (consists of 150 observations and 4 variables), which is a data set without missing values. We created randomly increasing proportion of missing values for one and two variables. We will compare four imputation methods, Mean imputation, Regression imputation, kNN imputation and miss Forest using the mean absolute error (MAE). Mean absolute error measures the average distance between the estimated values $\hat{y}_i$ (imputed values) and the original values y_i, thus will be used to evaluate the accuracy of the imputation method and is given from the following equation:

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}. \tag{3.12}$$

Following the notation of this chapter (3.12) can be written in the following form

$$MAE = \frac{\sum_{i=1}^{n} |\hat{x}_{ij} - x_{ij}|}{n}, \text{ for one missing variable } V_j^{mis}$$

and in the case of more than one missing variables we write

$$MAE = \frac{\sum_j \sum_{i=1}^{n} |\hat{x}_{ij} - x_{ij}|}{n}, \text{ for all indices } j \text{ where are } V_j^{mis}.$$

[11] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7, Part II, 179–188.

The data were collected by Anderson, E (1935). The irises of the Gaspe Peninsula, Bulletin of the American Iris Society, 59, 2–5.

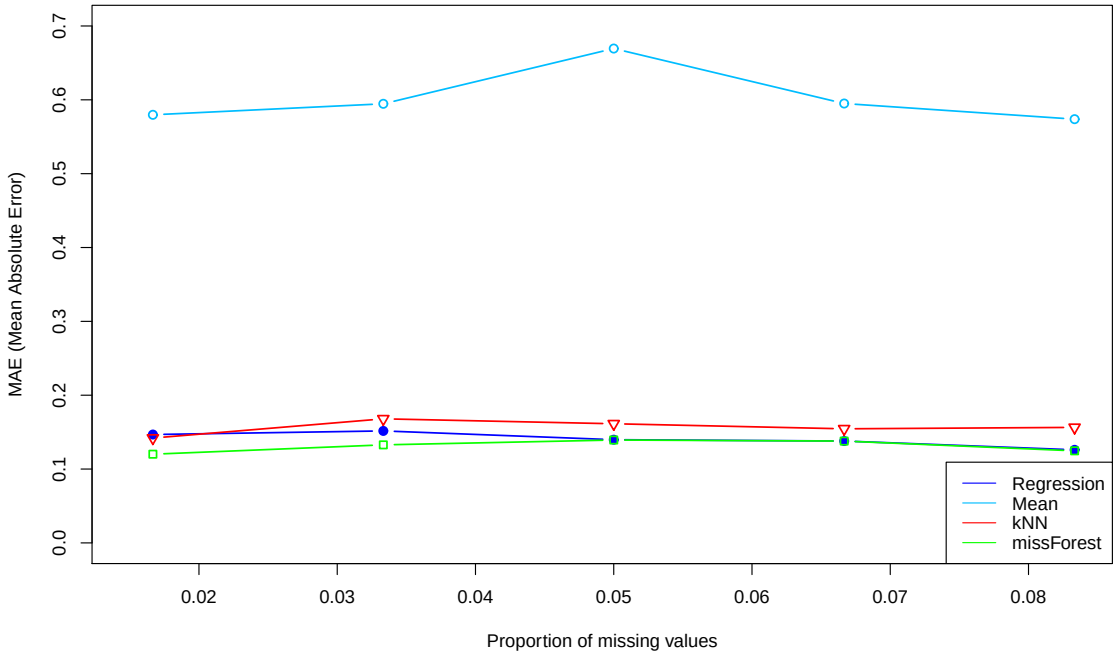


Figure 3.2: Imputation methods comparison for one missing variable

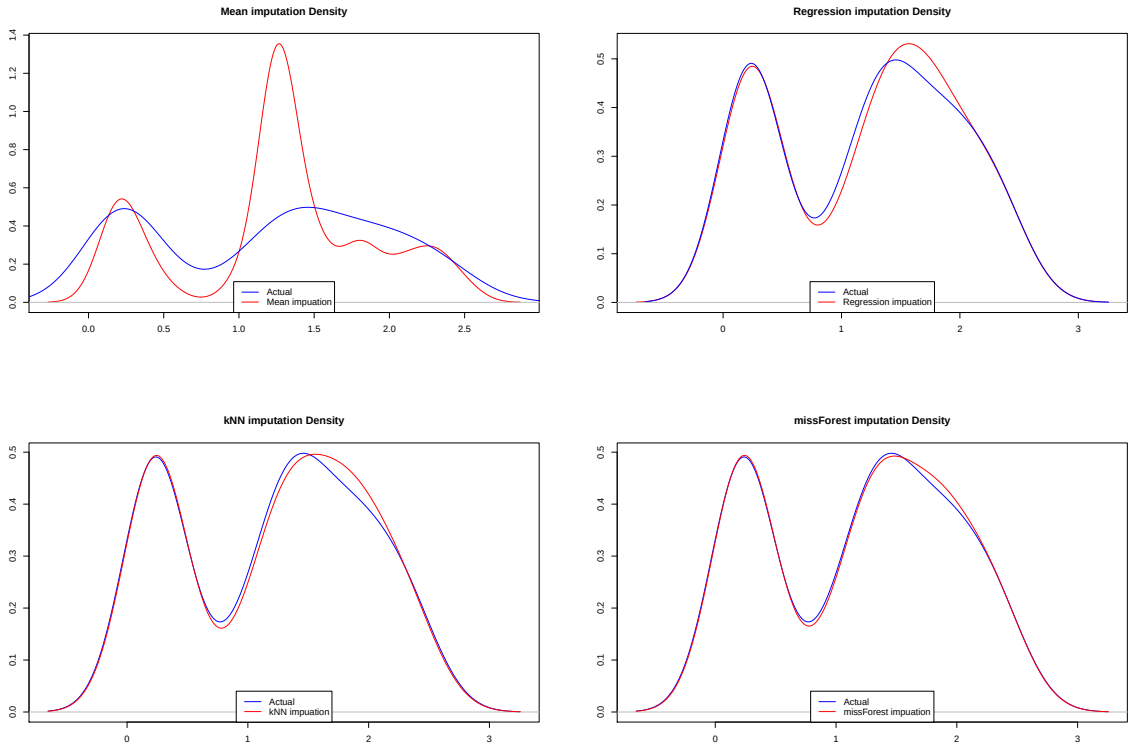


Figure 3.3: Densities of the imputed data

We created increasingly proportions of missing values for the variable *Petal Width*. In Figure 3.2 we see that the simplest imputation method, Mean imputation, has the lowest accuracy of all imputation methods. The other three imputation methods are very accurate with very low MAE values, furthermore miss Forest seems to be the most accurate. Note that the proportion of missing values (x-lab) is referred to the overall proportion (whole data set) and not to the missing variable. A better visual representation can give us the Figure 3.3 where depicts the density of the original data compared with the densities of the imputed data for each method separately.

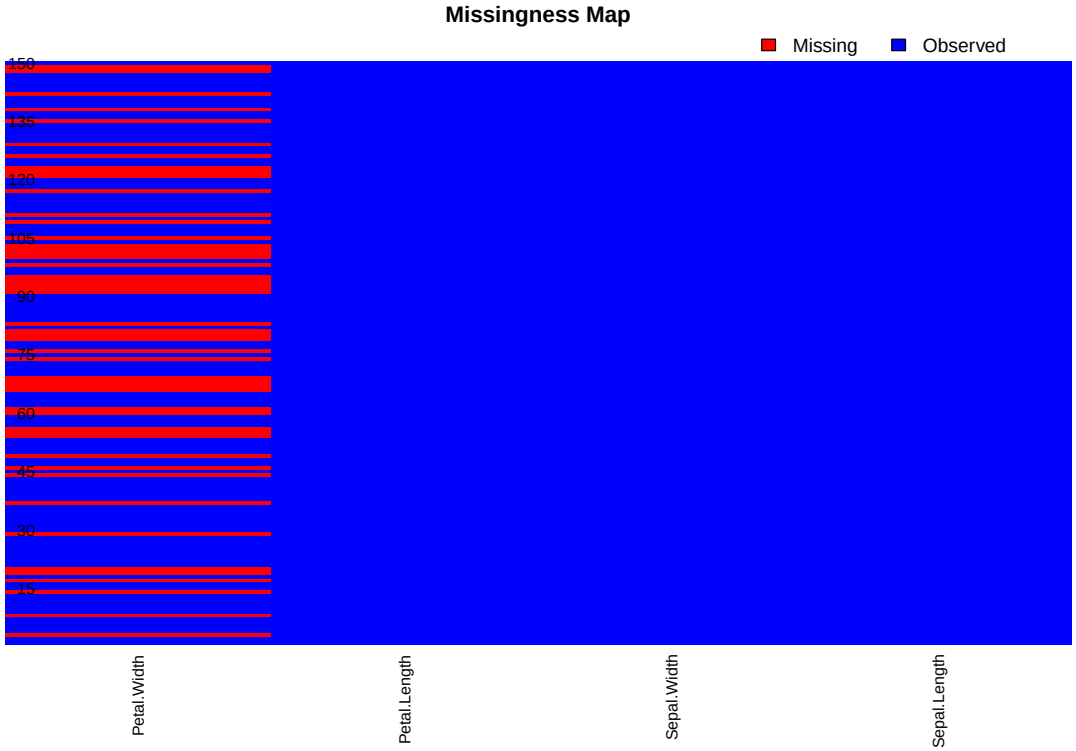


Figure 3.4: Missingness Pattern

Figure 3.3 is referred to 30% proportion of missing values for the variable Petal Width or 8.3% overall. The missingness pattern is given by Figure 3.4 where the values are MCAR due to the fact that they created randomly.

Two Missing Variables

As before we create randomly increasing proportions of missing values for two missing variables, *Sepal Length* and *Petal Length*.

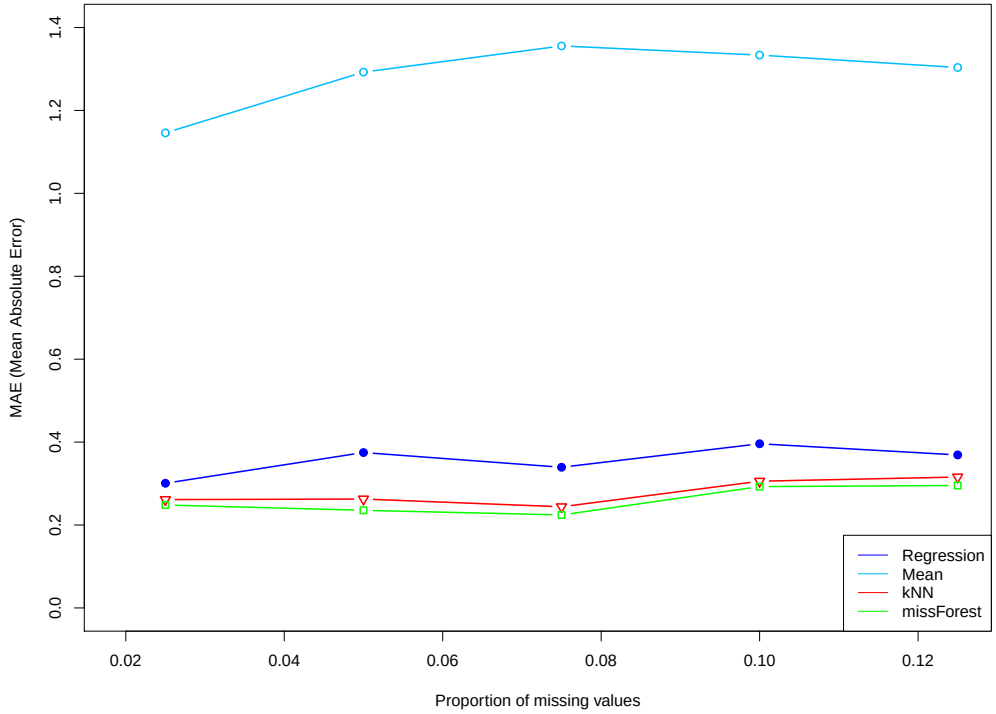


Figure 3.5: Imputation methods comparison for two missing variables

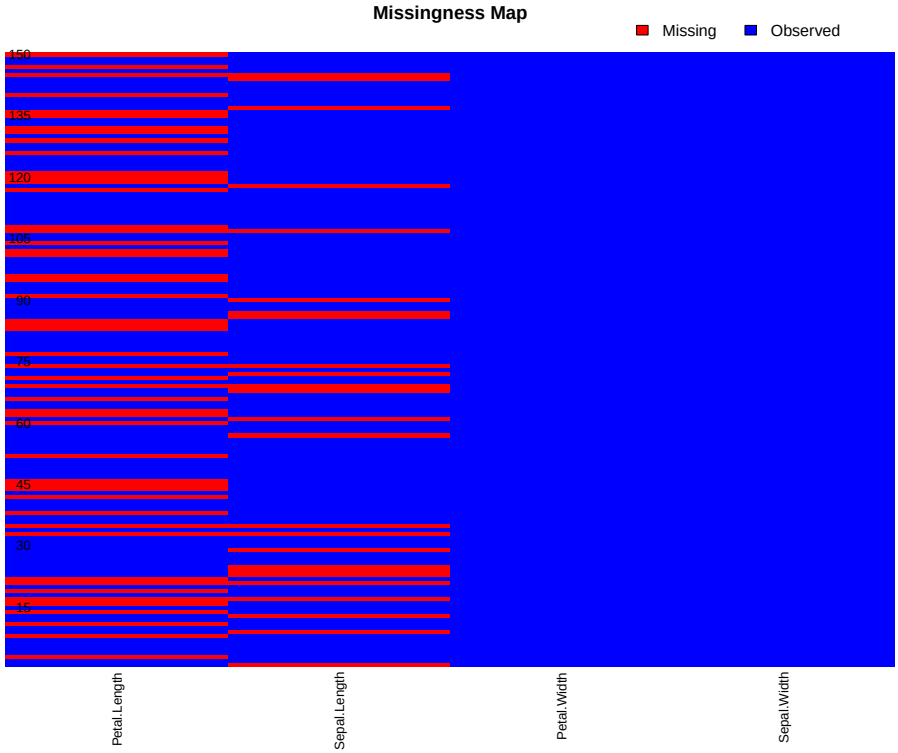


Figure 3.6: Missingness Pattern of two missing variables

The mean imputation is again by far the least accurate method and the miss Forest the most accurate with little difference from the kNN's imputation MAE. Regression imputation does not work well with more than one variables missing due to the fact that the regression model looses valuable information (uses only two of the variables to create the regression equation). Note that sequential regression imputation can perform better in cases of more than one variables are missing.

Figures 3.7 and 3.8 depict the densities of the original and imputed data for the corresponding missing variables. The proportion of the missing data is 12.5% and the missingness pattern is given by the Figure 3.6.

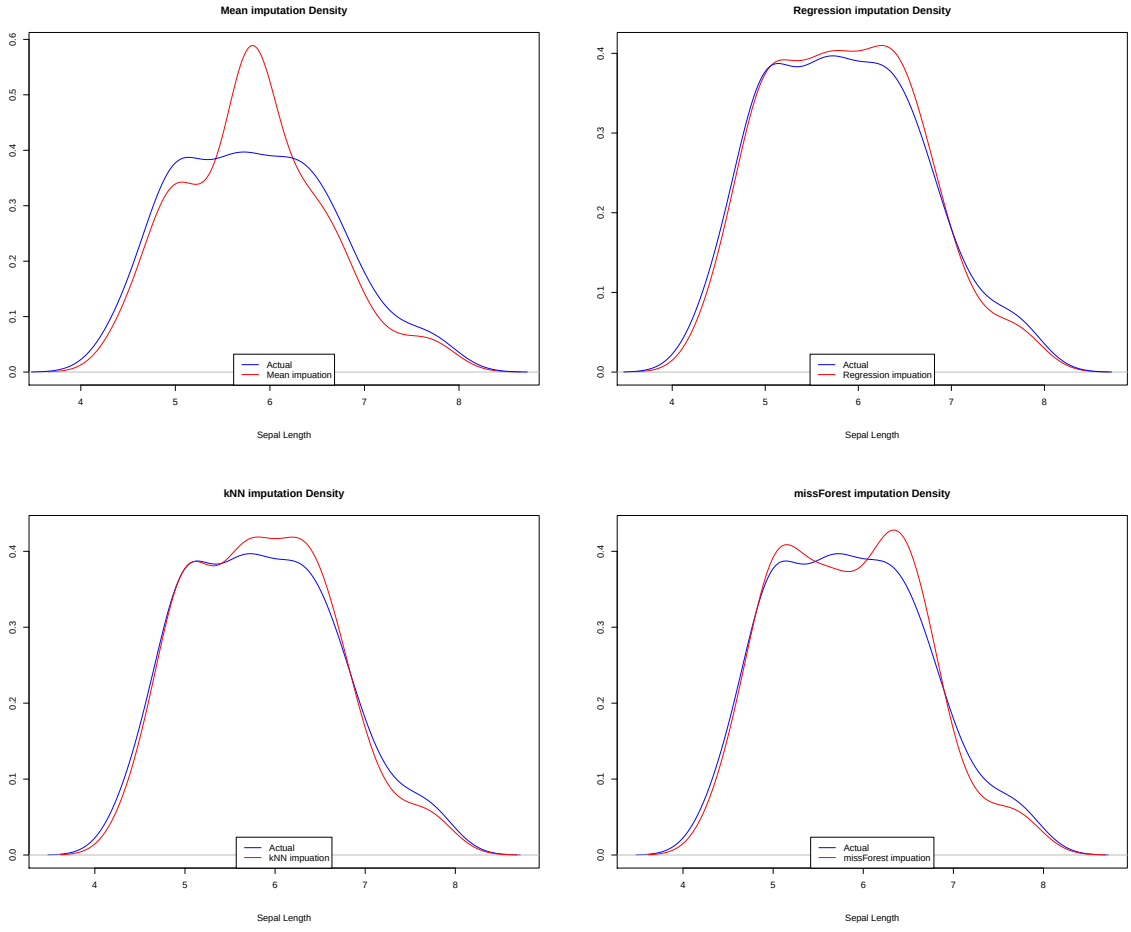


Figure 3.7: Densities for the variable Sepal Length of the imputed data

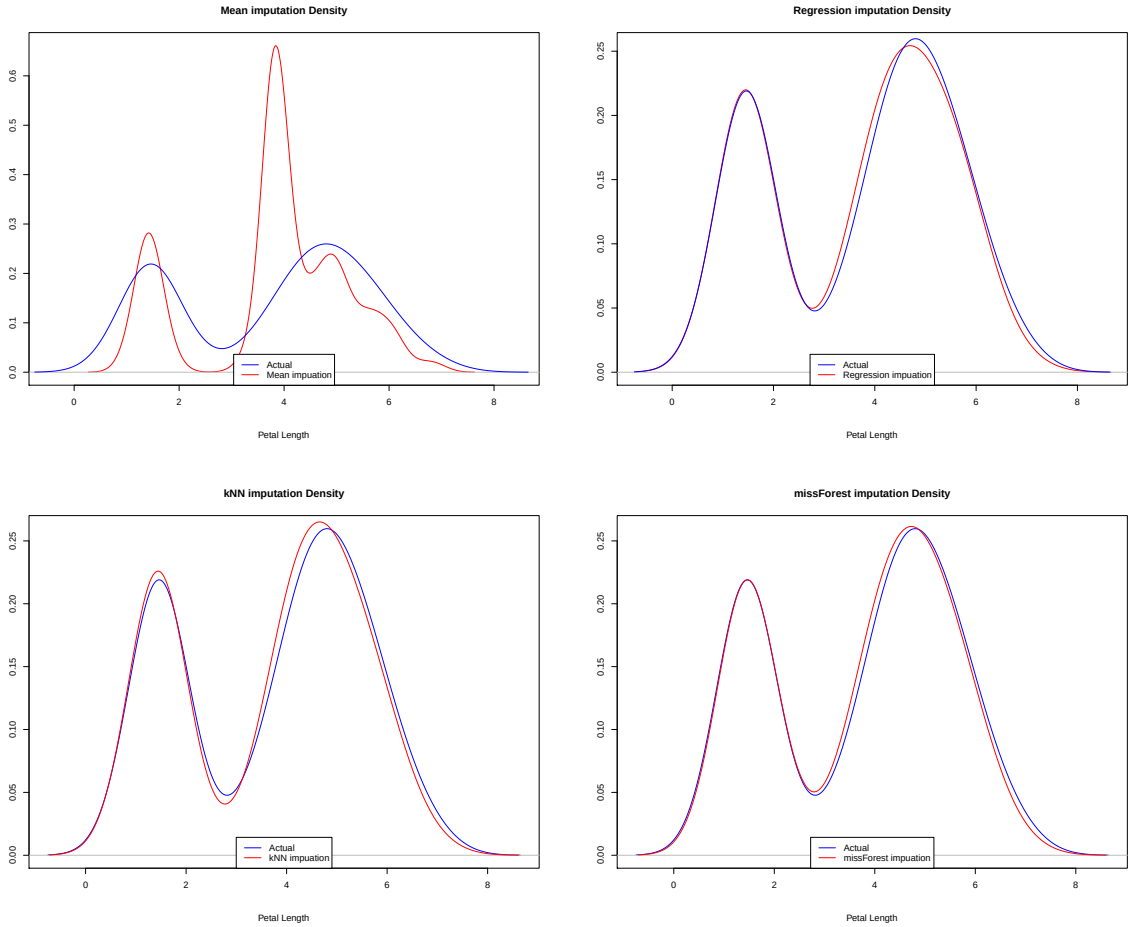


Figure 3.8: Densities for the variable Petal Length of the imputed data

In the this section we show that the most accurate methods are multivariate algorithmic procedures. In Chapter 5 we will illustrate a new (algorithmic) imputation method that uses the KSC algorithm, a clustering method that can deal with missing values, which will be described in Chapter 4. This new method improves the traditional methods, Mean imputation and Regression imputation.

Chapter 4

Intelligent Clustering

In this chapter we will illustrate three clustering algorithms that can improve the accuracy of the final Partition. The first two are modified K-means algorithms which can incorporate background knowledge from the data set expressed as constraints. The third algorithm again a modified K-means that can deal with missing values, without using imputation or marginalization though using the information that can be derived from the features (variables).

4.1 Incorporating Background Knowledge

The common distinction between supervised and unsupervised learning is that supervised algorithms recognize the information of the individual data labels. However there are a lot of forms that knowledge can take. In this type of clustering algorithms we would like to incorporate any form of information as a set of constraints. Suppose that we have an extended form of the data set that described in Chapter (1.1).

Student	Heigh in cm	Weight in kg	Age	Sport	Gender
Olivia	174	67	16	Basketball	F
Tom	178	75	17	Football	M
Helen	162	52	17	Volleyball	F
George	190	85	19	Football	M
...	...	...	...	...	

Table 4.1: Multivariate example extended

Each Student (observation) is described by a variety of features (variables), including Heigh in cm, Weight in kg, Age, Sport and Gender (F means female and M male). When we use clustering algorithms to derive knowledge from the data it is not possible to use all these variables. Consider that Olivia and George are siblings and must be in the same cluster. We cannot use a variable to capture that fact, because this requires a way to express the information of the relationship between the two observations (students).

In this chapter we will illustrate clustering algorithms where the knowledge can be expressed as a set of *instance level constraints*. We will introduce two forms of instance level constraints, *Hard constraints* and *Soft constraints*. Hard constraints are restrictions incorporated to the algorithm that must be satisfied in the final partition of the algorithm. Soft constraints are preferences (weights) about the final partition of the algorithm and they offer additional flexibility to the algorithm by modifying the objective function.

4.1.1 Constraints

The K-means algorithm is a clustering algorithm that minimizes the variance (objective function) of the final partition. When we modify K-means by incorporating a set of hard constraints that must be satisfied, we impose an additional requirement that the algorithm return partitions that fully satisfy the constraints. We use two types of constraints:

- A **must link** constraint specifies that two observations d_i and d_j must appear in the same output cluster. That is $class(d_i) = class(d_j)$, where $class(d)$ is the cluster that contains d. We will indicate that d_i **must-link** to d_j using the notation $d_i =_m d_j$. A set of must-link constraints defines an equivalence relation over $D \times D$ where D is the data set.
- A **cannot-link** constraint specifies that two observations must *not* be placed in the same cluster. A partition that satisfies a cannot-link constraint must have $class(d_i) \neq class(d_j)$, and we will indicate that d_i **cannot-link** to d_j with $d_i \neq_c d_j$. The relation defined by a set of cannot-link constraints in symmetric but not transitive.

$\forall i, j, k$: given		produce
$d_i =_m d_j$	$d_j =_m d_k$	$d_i =_m d_k$
$d_i =_m d_j$	$d_j \neq_c d_k$	$d_i \neq_c d_k$
$d_i \neq_c d_j$	$d_j =_m d_k$	$d_i \neq_c d_k$

Table 4.2: Hard constraints closure

If $d_i \neq_c d_j$ and $d_j \neq_c d_k$ we cannot say anything about the relationship between d_i and d_j, since cannot link relation is not transitive. Computing the closure in this way allows us to extend the cannot-link relation from each item to its equivalence class. In addition, any inconsistencies or conflicts between the constraints will be detected at this point.

Partial Labelled Data

Observation	Sepal Length	Sepal Width	Petal Length	Petal Width	Species (Labels)
1	5.10	3.50	1.40	0.20	setosa
2	4.90	3.00	1.40	0.20	
3	4.70	3.20	1.30	0.20	setosa
4	4.60	3.10	1.50	0.20	
⋮	⋮	⋮	⋮	⋮	
51	7.00	3.20	4.70	1.40	versicolor
52	6.40	3.20	4.50	1.50	versicolor
53	6.90	3.10	4.90	1.50	
54	5.50	2.30	4.00	1.30	versicolor
⋮	⋮	⋮	⋮	⋮	
147	6.30	2.50	5.00	1.90	
148	6.50	3.00	5.20	2.00	
149	6.20	3.40	5.40	2.30	
150	5.90	3.00	5.10	1.80	virginica

Table 4.3: Partially labelled iris data

In huge data sets we often face the problem of partially labelled data sets. Ordinary clustering algorithms do not make use of the additional information that the labels can give us, though they

just discard them. This valuable source of information must be incorporated to the output partition. Suppose that the Iris data set that consists of 150 observations and three species (setosa, virginica and versicolor) is partially labelled with only 70 labels be known. The usual K-means algorithm will not use the information that the labels can give us and the output partition can be summarized in Table (2.6).

Basu et al.[1](2002) suggested a Semi-supervised Clustering by Seeding. This clustering method uses the labelled data to select the initial centroids of the k-means algorithm. For each group of observations with the same label it calculates the mean and uses it as the initial cluster centroid. This is some kind of constraint that affects the final output (partition) of the algorithm. The approach of the constraint clustering algorithms that we will introduce in this chapter is to create *pairwise relations* between the known labels. For example if the iris data set was given from the partial labelled table 3.2 we know observations 1 and 3 must be in the same cluster, 51, 52 and 54 must be also in the same cluster but not in the cluster where are located the observations 1 and 2 and observation 150 cannot be in the same cluster with any of the observations 1, 3, 51, 52, 54. The transitive relation is created by the fact that if observations 1 and 2 must be together and 51 cannot be in the same cluster with observation 2, then 2 cannot be in the same cluster with 1. Formally we write that

$$(d_1, d_3) \in Con_{=} \quad \& \quad (d_1, d_{51}) \in Con_{\neq}.$$

Now suppose that we have the iris partial labelled data of Table 3.2 and the information that the data can be partitioned in three groups. Due to the fact that some labels are known, when we take the final partition of the clustering algorithm in each cluster we have representatives of each label (species), then the problem is a semi-supervised problem where we can classify each observation to the corresponding label (species) by labelling the whole cluster by a representative labelled observation. For example if the observations 1, 3 and 4 where in the same cluster at the final partition of the algorithm, then the observation 4 will be classified as setosa.

4.2 COP-KMEANS Algorithm

The COP-KMEANS[2] algorithm is a modified K-means designed to accommodate a set of constraints (must-link and cannot link). As described in Chapter 2.3.2 K-means begins with k initial (randomly selected) centroids and assigns the observation d_i to the cluster C that will minimize the total variance, then finds the mean of each cluster an iterates between these two steps until the centroids cannot change.

Incorporating must-link constraints. The must-link constraint is an indicator that two items must be placed in the same cluster. The modification for incorporating must-link constraints is a pre processing step where each group of observations that must be linked together is represented by a single observation, the mean of them. Now the reduced data set can be partitioned using the K-means algorithm. All observations that are represented by the mean of group will be placed in the cluster that hosts the representative of them (i.e. their mean). For example in table $3, 2$ observations 1 and 3 are represented by the $mean(d_1, d_3) = (d_1 + d_3)/2$.

Incorporating cannot-link constraints. In the constraint K-means algorithm the cannot link constraints indicate that two observations can not be in the same cluster. Therefore the modification is to prune a list of possible host clusters to eliminate any solution that would violate a constraint.

[1]Basu, S., Banerjee, A., & Mooney, R. (2002). Semi-supervised clustering by seeding. In In Proceedings of 19th International Conference on Machine Learning (ICML-2002.

[2]Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001, June). Constrained k-means clustering with background knowledge. In ICML (Vol. 1, pp. 577-584).

Algorithm 4 COP-KMEANS

1: Identify each group of items that must be linked together. Replace each group with a single item that is the mean of items in the group. Update $Con_{\neq}$ to be consistent with the reduced data.

2: Randomly initialize k cluster centrers (from the reduced data) $\mu_1, \mu_2 \ldots, \mu_k$

3: For each observation d in D, assign it to the cluster that will minimize the total variance, **such that** VIOLATE-CONSTRAINTS$(d, D, Con_{\neq})$ **is false**. **If no such cluster exists, halt(return**{})

4: Update each cluster centroid μ_i by averaging all of the points $d_j \in C_i$ that have been assigned to it.

5: **Iterate** between Steps (3) and (4) until convergence

6: **return** partition $\{C_1, \ldots, C_k\}$

VIOLATE-CONSTRAINTS(data point d, cluster C, cannot-link constraints $Con_{\neq}$)
1: For each $(d, d_{\neq}) \in Con_{\neq}$: If $d_{\neq} \in C$, return true.
2: Otherwise return false.

Any inconsistencies between the constraints should be identified in step 1 of the algorithm. For example if $d_i =_m d_j$ and $d_j \neq_c d_k$ then $d_i =_m d_k$ is an inconsistent constraint and must return a warning to the user. Furthermore the cannot link constraints can give better initial centroids to the algorithm which gives a faster convergence and a final partition with less total variation.

4.2.1 Evaluation of Clustering Accuracy

Rand index

In the following experimental results the clustering accuracy will be evaluated by the Rand index[3] which is a metric that calculates the agreement between two partitions P_1 and P_2. The partition P_1 will be the final partition of the clustering method and P_2 the known partition of the labelled data set. Let $D = \{d_1, d_2, \ldots, d_n\}$ be the data set, $P_1 = \{P_{1,1}, P_{1,2}, \ldots, P_{1,k}\}$ and $P_2 = \{P_{2,1}, P_{2,2}, \ldots, P_{2,k}\}$, where $P_{i,j}$, $i = 1, 2$ $j = 1, 2, \ldots, k$ is the cluster j(subset) of the partition i, $\cup_{j=1}^{k} P_{1,j} = \cup_{j=1}^{k} P_{2,j} = D$ and $\cap_{j=1}^{k} P_{1,j} = \emptyset = \cap_{j=1}^{k} P_{2,j}$. For every pair of observations d_i, d_j there are four possible outcomes:

1. the pair d_i, d_j is in the same subset in P_1 and in P_2, the number of all these pairs is denoted by **a**
2. the pair d_i, d_j is in different subset in P_1 and in P_2, the number of all these pairs is denoted by **b**
3. the pair d_i, d_j is in the same subset in P_1 but in different subset in P_2, the number of all these pairs is denoted by **c**
4. the pair d_i, d_j is in different subset in P_1 but in the same subset in P_2, the number of all these pairs is denoted by **d**.

The number of all possible pairs is $\binom{n}{2}$. Then the total agreement between P_1 and P_2 can be calculated by the following equation:

$$Rand(P_1, P_2) = \frac{a + b}{\binom{n}{2}.} \tag{4.1}$$

[3]Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association, 66(336), 846-850.

Adjusted Rand index
Let $D = \{d_1, d_2, \ldots, d_n\}$ be the data set, $P_1 = \{P_{1,1}, P_{1,2}, \ldots, P_{1,k}\}$ and $P_2 = \{P_{2,1}, P_{2,2}, \ldots, P_{2,k}\}$, where $P_{i,j}$, $i = 1, 2$ $j = 1, 2, \ldots, k.$, The four possible outcomes that we discussed in the Rand index can be summarized in a contingency table in the following form:

P_1 \ P_2	$P_{2,1}$	$P_{2,2}$	$\ldots$	$P_{2,k}$	$Sums$
$P_{2,1}$	n_{11}	n_{12}	$\ldots$	n_{1k}	$n_{1.}$
$P_{2,2}$	n_{21}	n_{22}	$\ldots$	n_{2k}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$P_{2,k}$	n_{k1}	n_{k2}	$\ldots$	n_{kk}	$n_{k.}$
$Sums$	$n_{.1}$	$n_{.2}$	$\ldots$	$n_{.k}$	

where n_{ij} is the number of observations $d \in D$ that appear in both subsets $P_{1,i}$ and $P_{2,j}$, we write that $n_{ij} = |X_i \cap X_j|$ and we denote as T the total agreement between the partitions P_1 and P_2 The Adjusted Rand index[4] is given by the following equation:

$$\text{Adj.Rand index} = \frac{\text{Index} - \text{Expected Index}}{\text{Maximum Index} - \text{Expected Index}} \tag{4.2}$$

or

$$\text{Adj.Rand index} = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}] - \sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}/\binom{n}{2}}. \tag{4.3}$$

The values of Rand index are between 0 and 1 and the values of Adjusted Rand index can be negative.Values close to 1 shows that the accuracy of the final partition is high.

4.2.2 Experimental Results of COP-KMEANS

For this example we will use the iris data set which consist of 150 observations (iris plants) and described by four variables. There are three distinct classes, iris virginica, iris setosa and iris versicolor. In our experiments the accuracy of the final partition has climbed from 88% to 93.3%, where we incorporated 205 constraints. Furthermore we will see the confusion matrices for a variety of constraints.

Incorporating fourteen constraints, we see in Table 3.3 that the algorithm wrongly classified four observations to versicolor and fifteen to virginica. The overall accuracy is 88%.

	setosa	versicolor	virginica
1	50	0	0
2	0	46	14
3	0	4	36

Table 4.4: Confusion matrix of iris with 14 constraints

Incorporating forty two constraints, we see in the Table 3.4 that the algorithm wrongly classified five observations to versicolor and ten to virginica and the overall accuracy is 90%.

Incorporating forty four constraints, we see in the Table 3.5 that the algorithm wrongly classified four observations to versicolor and seven to virginica and the overall accuracy is 91.3%.

Incorporating 205 constraints, we see in the Table 3.6 that the algorithm wrongly classified seven observations to versicolor and 3 to virginica and the overall accuracy has climbed to 93.3%.

[4]Hubert, L., & Arabie, P. (1985). Comparing partitions. Journal of classification, 2(1), 193-218.

	setosa	versicolor	virginica
1	0	45	10
2	50	0	0
3	0	5	40

Table 4.5: Confusion matrix of iris with 42 constraints

	setosa	versicolor	virginica
1	0	44	7
2	0	6	43
3	50	0	0

Table 4.6: Confusion matrix of iris with 44 constraints

Generally it is easy to see that the relation between the accuracy of the final partition and the number of constraints is not linear as you can see in the figure. Furthermore the accuracy of clustering does not always depend on the number of the constraints but in the accuracy of them.

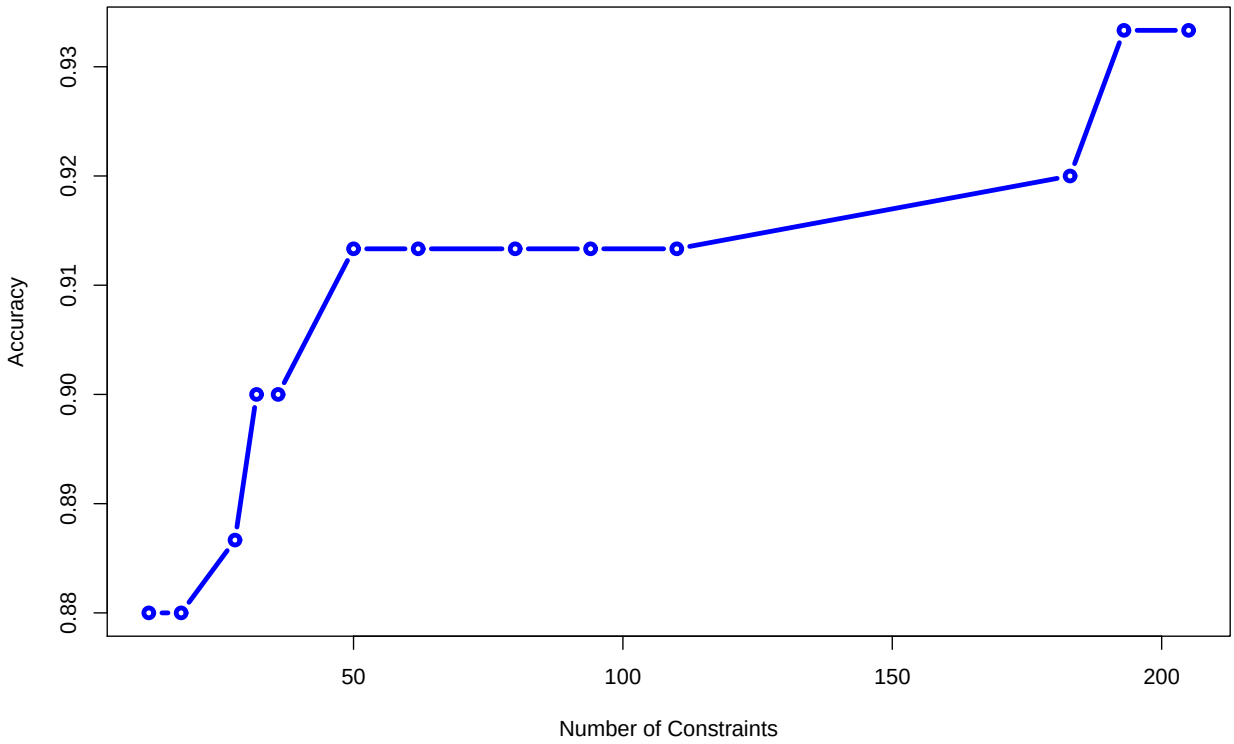


Figure 4.1: Accuracy of COP-KMEANS

Wagstaff et. al (2002) has conducted experiments in many UCI repository data sets and has shown that clustering accuracy steadily increases with the incorporation of constraints.

	setosa	versicolor	virginica
1	0	7	47
2	50	0	0
3	0	43	3

Table 4.7: Confusion matrix of iris with 205 constraints

4.3 Soft Constrained Version of COP-KMEANS

In Section 3.2 we described COP-KMEANS, a modified k-means algorithm that accommodates a set of hard constraints (must-link and cannot-link). SCOP-KMEANS[5] like COP-KMEANS is a modified k-means that incorporates a set of *soft constraints*.

4.3.1 Soft Constraints

In Section 3.1.1 we defined two kinds of hard constraints. Generally we can describe the cannot-link relation between two observations with the number -1, the must-link relation with the number 1 and if no information is given between two observations with the number 0. For soft constraints we augment each relationship (between two observations) with a *strength* factor s, that indicates how reliable the constraint and is denoted as $(\langle d_i, d_j, s\rangle)$. The value of factor s ranges from -1 to 1, where values close to 1 indicate a preference towards being grouped together and values close to -1 indicate preference against being grouped together. The constraint $\langle d_i, d_j, 1\rangle$ is equivalent to must-link constraint, $\langle d_i, d_j, -1\rangle$ is equivalent to cannot-link constraint and the constraint $\langle d_i, d_j, 0\rangle$ is a "don't care statement". In chapter 3.1.1 we defined the hard constraints closure as:

$\forall i, j, k$: given		produce
$d_i =_m d_j$	$d_j =_m d_k$	$d_i =_m d_k$
$d_i =_m d_j$	$d_j \neq_c d_k$	$d_i \neq_c d_k$
$d_i \neq_c d_j$	$d_j =_m d_k$	$d_i \neq_c d_k$

To extend this $d_i =_m d_j \;\; d_j \neq_c d_k$ produce $d_i \neq_c d_k$ relation for soft constraints we write that $\langle d_i, d_j, s_1\rangle \;\; \langle d_i, d_j, -s_2\rangle$ produce $\langle d_i, d_j, -\min(s_1, s_2)\rangle$. The soft constraints closure is defined by the table below:

$\forall i, j, k$: given		produce
$\langle d_i, d_j, s_1\rangle$	$\langle d_j, d_k, s_2\rangle$	$\langle d_i, d_k, \min(s_1, s_2)\rangle$
$\langle d_i, d_j, s_1\rangle$	$\langle d_j, d_k, -s_2\rangle$	$\langle d_i, d_k, -\min(s_1, s_2)\rangle$
$\langle d_i, d_j, -s_1\rangle$	$\langle d_j, d_k, s_2\rangle$	$\langle d_i, d_k, -\min(s_1, s_2)\rangle$

Table 4.8: Soft constraints closure

and when $\langle d_i, d_j, -s_1\rangle, \langle d_j, d_k, -s_2\rangle$ we cannot say anything about the relation between d_i and d_j. Soft constraints is incorporated to the k-means algorithm by modifying its objective function with real-valued penalty for violating constraints. We define CV as the maximum strength of the violated constraints, if any. The SCOP-KMEANS objective function combine variance (k-means objective function) with CV in the following way:

$$f(C_1 \dots C_k) = \frac{var}{1 - \text{CV}} \tag{4.4}$$

where *var* is the variance and CV is an estimation of the proportion of the maximum strength constraints that are violated, weighted by their strength.

[5]Wagstaff, K. L., & Cardie, C. (2002). Intelligent clustering with instance-level constraints. USA: Cornell University.

Algorithm 5 SCOP-KMEANS

1: Randomly initialize k cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \ldots, \boldsymbol{\mu}_k$

2: Assign each observation $d \in D$ to the cluster C_j which will minimize the objective function

$$f(C_1 \ldots C_k) = \frac{var}{1 - \text{CV}}$$

where var is the variance of the partition and CV is the value of constraint violation which calculates from $CV := \text{CONSTVIOL}(d, C_1 \ldots C_k, Pref)$

3: Update each cluster centroid $\boldsymbol{\mu}_i$ by averaging all of the points $d_j \in C_i$ that have been assigned to it.

4: **Iterate** between Steps (2) and (3) until convergence[6]

5: **return** partition $\{C_1, \ldots, C_k\}$

CONSTVIOL(data point d, partition $C_1 \ldots C_k$, preferences $Pref$)

1: Let $CV_{\max} := 0$, $nConst := 0$, $nViol := 0$

2: For each $\langle d, d', s \rangle \in Pref$:
If $|s| > CV_{\max}$,
  If $s > 0$ and $d.class \neq d'.class$, **then** $CV_{\max} := s$ and $nConst := nViol := 1$.
  Else if $s < 0$ and $d.class = d'.class$, **then** $CV_{\max} := s$ and $nConst := nViol := 1$.
Else if $|s| := CV_{\max}$,
  Increment $nConst$ by 1.
  If $s > 0$ and $d.class \neq d'.class$, **then** increment $nViol$ by 1.
  Else if $s < 0$ and $d.class = d'.class$, **then** increment $nViol$ by 1.

3: Return $CV_{\max} * \frac{nViol}{nConst}$.

4.4 Clustering with Missing Values

Missing values occur for a variety of reasons specifically in large data sets. The most common approaches of dealing with missing values is marginalization and imputation. Imputation methods fill in the missing values with estimated values, where the most common among them is the mean imputation. In marginalization we omit the variables with missing values or we discard the observations that contain missing values. Both approaches are limited to what they can achieve due to the fact that we loose valuable information. In this section we will illustrate a clustering method that can deal with missing values without using imputation or marginalization. In contrast this method uses the information that missing variables (variables with missing values) can give us.

4.4.1 KSC Algorithm

KSC[7] (K-means Soft Constraints) algorithm is a modified k-means algorithm that designed to accommodate a set of soft constraints (as described in section 3.3), where can handle missing values. In this method we will use the variables with missing values to create soft constraints. We divide the set of variables into V_o, the set of observed variables and V_m the set of variables with missing values,

[7] Wagstaff, K. (2004). Clustering with missing values: No imputation required. Classification, Clustering, and Data Mining Applications, 649-658.

also referred to as the set of constraining features. In this algorithm we perceive V_m only as a source of additional knowledge. As described in Chapter 3.3.1 a soft constraint between two observations is given as a triple: $\langle d_i, d_j, s\rangle$. We create a constraint $\langle d_i, d_j, s\rangle$ between the observations d_i, d_j with values from V_m, where s is the negative Euclidean distance

$$s = -\sqrt{\sum_{v \in V_m} (d_i.v - d_j.v)^2}. \tag{4.5}$$

We do not create constraints with observations that have missing values. The value of s is negative due to the fact that it shows the degree that two observations d_i, d_j should be separated. The modified objective function of KSC is

$$f := (1-w)\frac{Var}{Var_{\max}} + w\frac{CV_d}{CV_{\max}}, \tag{4.6}$$

$Var = \sum_{d \in D} dist(d, \mu_i)^2$ where $Var_{\max}$ is the total variance of the dataset (or the variance obtained by assigning all items in the same cluster) and it used to normalize the variance of the partition. CV is the sum of the squared strengths of violated constraints in the set of Soft Constraints (SC). $CV_{\max}$ is the sum of all squared constraints, which normalizes the quantity CV and w is a weighting factor that their values range from 0 to 1.

Algorithm 6 KSC (K-means Soft Constraints)

1: Randomly initialize k cluster centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \ldots, \boldsymbol{\mu}_k$.

2: For each observation d in D, assign it to the cluster that will minimize function

$$f := (1-w)\frac{dist(d, \boldsymbol{\mu}_i)^2}{Var_{\max}} + w\frac{CV_d}{CV_{\max}}$$

where CV_d is the sum of squared violated constraints in Soft Constraints set that involve d.

3: Update each cluster centroid $\boldsymbol{\mu}_i$ by averaging all of the points $d_j \in C_i$ that have been assigned to it.

4: **Iterate** between Steps (2) and (3) until convergence.

5: **return** partition $\{C_1, \ldots, C_k\}$.

4.4.2 Experimental Results

Iris data set[8] consists of 150 observations and 4 variables in 3 distinct classes (Species). We will create randomly missing values to the variables *Sepal Length* and *Petal Length*.

[8]Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7, Part II, 179–188.

The data were collected by Anderson, Ed.(1935). The irises of the Gaspe Peninsula, Bulletin of the American Iris Society, 59, 2–5.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	·	3.50	1.40	0.20
2	4.90	3.00	1.40	0.20
3	·	3.20	1.30	0.20
4	4.60	3.10	·	0.20
5	5.00	3.60	·	0.20
⋮	⋮	⋮	⋮	⋮

Table 4.9: Iris data set with missing values

The most common approaches while clustering with missing values is to fill in the missing values by using an imputation method or to use only the variables that do not contain missing values (marginalization method). We will create randomly increasing fractions of missing values and we will compare the clustering accuracy of these methods with KSC algorithm.

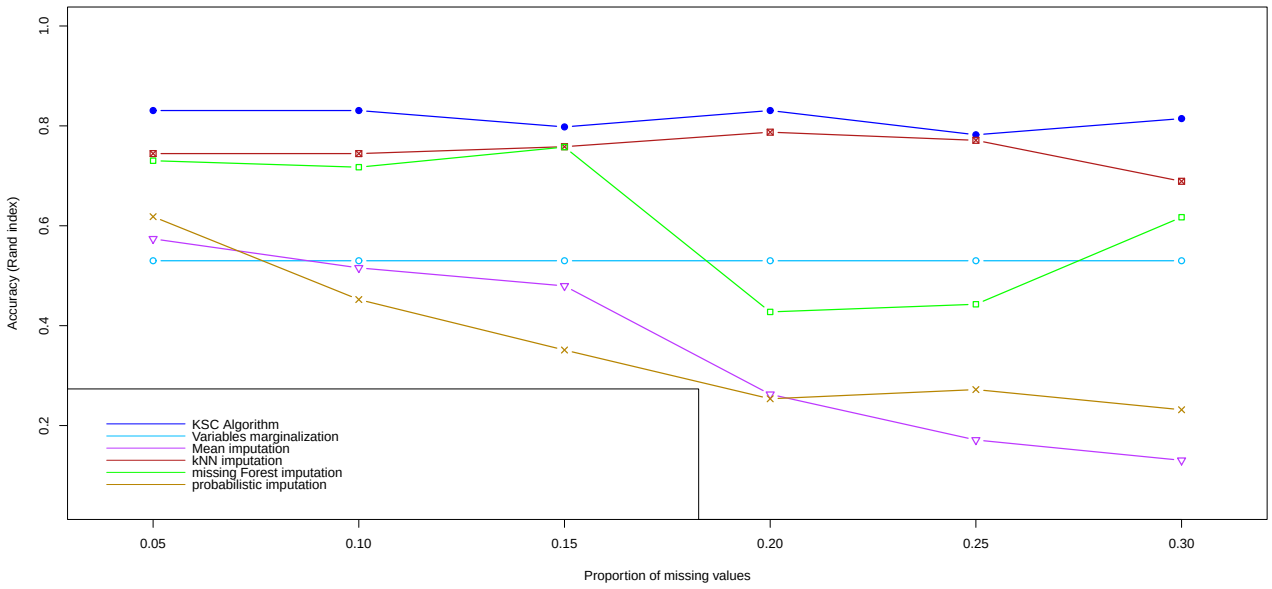


Figure 4.2: Comparison of KSC with imputation methods for two out of four missing variables

We run KSC-algorithm with $w = 0.5$ (proportional to the number of variables with missing values) and K-means algorithm (for imputed and marginalized data) for six different proportions of missing values in the data set. The accuracy of clustering while using the most common (simple) imputation methods, mean imputation and probabilistic imputation, is very low (about 60%) when only the 0.05% of the values is missing, and decreases rapidly while the proportion of the missing values is increasing. Variables marginalization method is the easiest approach (we discard variables with missing values), though it does not work well (accuracy about 55%) due to the fact that we loose valuable information. Missing Forest and k Nearest Neighbors are proposed as two of the most accurate imputation methods, though their accuracy depends on the number of missing values. In all proportions of missing values KSC-algorithm's accuracy is greater (about 85%) than the other methods and as we can see in figure 3.2 does not fall as the number of missing values increases.

4.4.3 Choice of w (weight)

Following Wagstaff et al. (2004) there is no general rule for selecting the best w value, though it can be estimated by clustering with several different values for w on a small labeled subset of the data. The general idea is to choose a weight proportional to the number of variables with missing values, though if we know that one or more variables contribute more to separation of the final partition, they should be given a larger weight.

We run KSC algorithm for all the possible combinations of two missing variables (variables with missing values), and for 99 different w (weight) values from 0.01 to 0.99. Values close to 0 give more weight to the non missing variables and values close to 1 give more weight to the missing variables. According to Figure 4.3, approximately in all the cases a weight equal to the proportion of the number ($w = 0.5$) of missing variables achieves a very accurate clustering outcome.

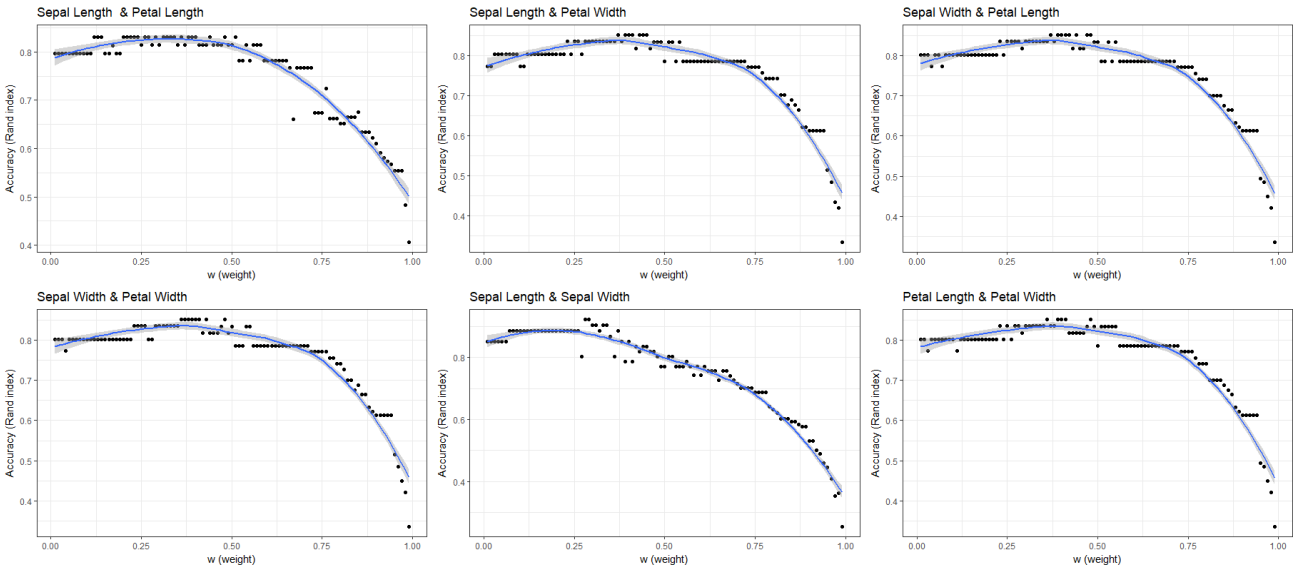


Figure 4.3: Weights comparison for two missing variables

In the diagonal of the matrix plot (Figure 4.4) we see the densities of species for each variable. We can derive that the greatest separation among species is achieved by the variable *Petal Length*, the second greatest separation is achieved by the variable *Petal Width*, the variable *Sepal Length* does not give us a very good separation and the variable *Sepal width* does not separate the species at all.

We run again KSC algorithm for 99 different w (weight) values from 0.01 to 0.99, but now for only one missing variable. In the case of two missing variables a weight proportional to the number of the missing variables can be a good choice. Now in the case of one missing variable the value of w depends on the separation ability of the missing variable.

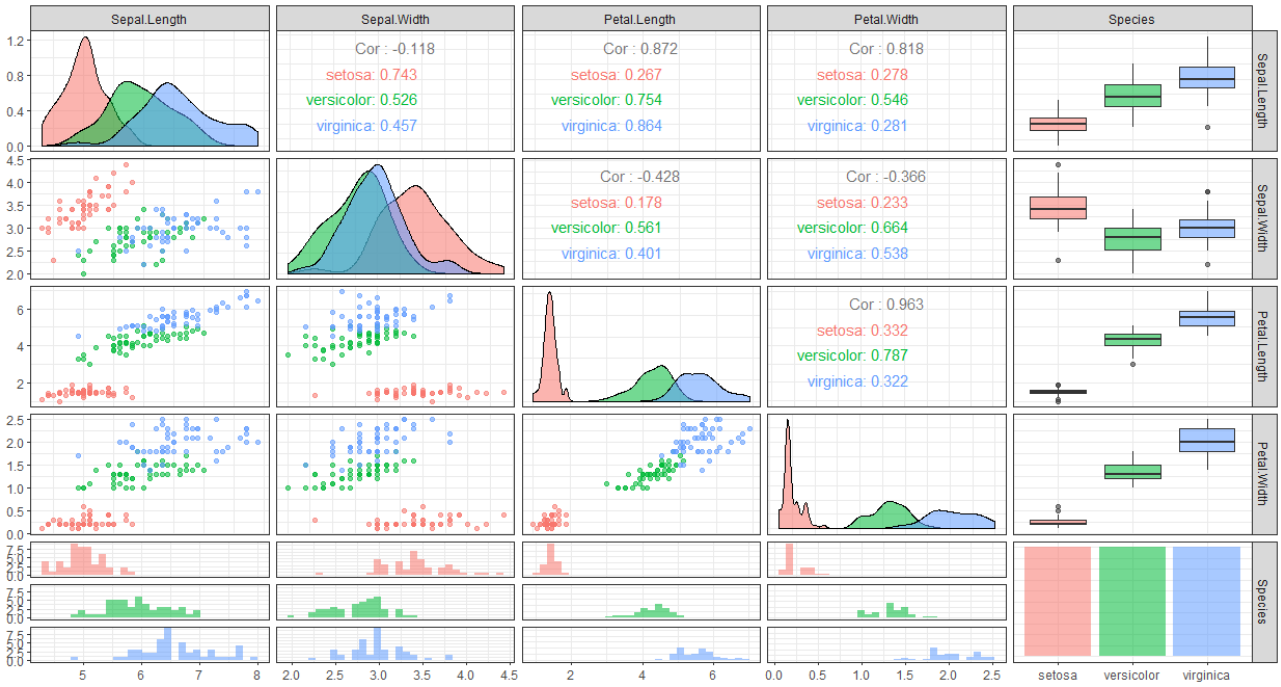


Figure 4.4: Iris summary matrix plot

As we mentioned before the variable *Sepal Width* does not separate the species at all. As we can see in Figure 4.5 the clustering accuracy is approximately the same with values of w from 0.01 to 0.6. When the missing variable is the *Petal Length* its easy to see that the clustering accuracy is very low. This happens because the variable Petal Length gives the greatest separation among the species, and if is missing we loose valuable information about the separation. In Figure 3.5 we can see that a value of w between 0.7 and 0.9 can increase the clustering accuracy.

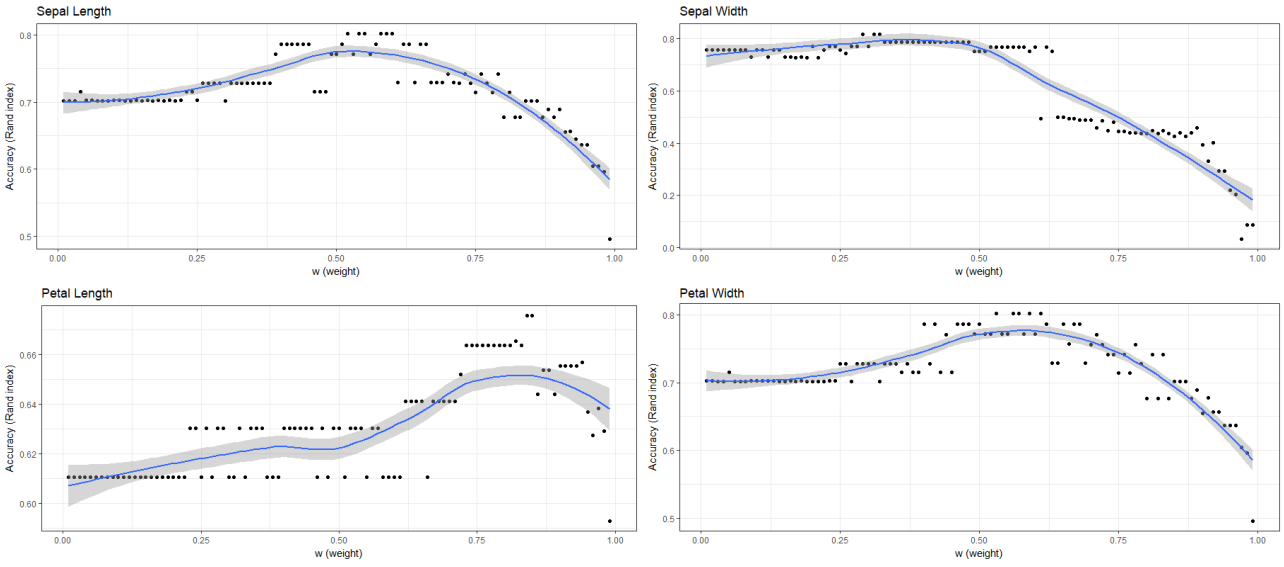


Figure 4.5: Weights comparison for one missing variable

Chapter 5

Partition Imputations

The main purpose of this work is to describe multivariate analysis techniques and specifically clustering methods. In Chapter 2 we have fully described K-means, the most popular clustering method. In Chapter 3 we briefly described missing data methods and specifically imputation methods. In Chapter 4 we described 3 intelligent clustering methods, COP-Kmeans, SCOP-Kmeans and KSC (Kmeans Soft Constraints) algorithm, where COP-Kmeans is a modified Kmeans that incorporates external sources of information as a set of hard constraints in order to improve the accuracy of the final partition. SCOP-Kmeans like COP-Kmeans is a modified Kmeans that incorporates a set of Soft constrains, and KSC is an algorithm that has a build in routine which uses the missing values as an external source of information, and improves the accuracy of the final partition compared with the traditional missing data methods (imputation, variables or observation marginalization). In this Chapter we will illustrate a new imputation method that was developed for the purposes of this thesis. This method is somehow a combination of KSC algorithm and two of the most traditional imputation methods, Mean imputation and Regression imputation.

5.1 APPROACH

Let $X_{n\times p}$ be a rectangular $n \times p$ data matrix where n denotes the number of observations and p denotes the number of variables. We assume that some elements x_{ij} of the the matrix are missing and denoted as x_{ij}^{mis},$i = 1, \ldots, n$, $j = 1, \ldots, p$, and at least one variable is complete for all elements. We denote the completely observed variable by V_j^{com} and if some elements are missing, V_j^{mis}. Furthermore if the observation $\mathbf{x}_i$ is complete for all variables is denoted by $\mathbf{x}_i^{com}$ and if some of them are missing is denoted by $\mathbf{x}_i^{mis}$. The set of complete observations is denoted by X^{com} and the set of missing observations is denoted by X^{mis}.
Clustering algorithms try to divide the data set into k groups in such way that observations in the same group are more "similar". Our approach is to divide the data set D into k homogeneous non overlapping subsets $D_1, \ldots, D_k$, $D_i \cap D_j = \emptyset$, $\cup_{i=1}^{k} D_i = D$, and for each subset D_i, $i = 1, \ldots, k$ use Mean imputation or Regression imputation and then we place the imputed values back to their initial positions in the data matrix. This partition of the data set can be achieved by the KSC algorithm (Wagstaff et al. 2004), which is a modified Kmeans that uses the missing values as an external source of information, without using imputation or marginalization (variables or observations), for more details see Section 4.4. The first method is called *Partition Mean imputation* and the second *Partition Regression imputation.*

5.2 Partition Mean imputation

In *Partition Mean imputation* we divide the subset as described before and in D_i for each variable with missing values we calculate the average of the recorded elements and we replace the missing elements with this value. As described in Section 2.3.2 the accuracy of the Kmeans algorithm depends on the choice of the initial centroids, that is usually random. In order to find the best partition we run Kmeans (usually) 20 times and we choose the partition with the smallest total within cluster variation, that is usually the most appeared. In KSC algorithm the objective function is modified, thus we choose the partition with the smallest total within cluster objective function value.

The first step to the algorithmic procedure of Partition Mean imputation is to run KSC 20-100 times with w proportional to the number of missing variables to find the best partition. Then each D_i is treated as an individual data set, and for $i = 1, \ldots, k$ we perform simple mean imputation. We identify the variables with missing values V_j^{mis} in each D_i, if they are any, we calculate the average of the observed values $\mu_j^{mis} = \sum_i x_{ij}^{com}/$(number of complete i) for the recorded i's and we replace the missing elements with these values. The last step is to place the imputed values back to their initial positions on the data matrix.

Algorithm 7 Partition Mean imputation

1: **Require:** $X_{n\times p}$ a data set matrix with at least one completed variable V_j^{com}
2: Run KSC-algorithm 20 times and identify the best partition $\{D_1, D_2, \ldots, D_k\}$
3: **for** $i \leftarrow 1$ to k **do**
4: Store the initial data matrix $X_{n\times p}$ positions for each missing element in D_i
5: Perform simple mean imputation in D_i
6: **end for**
7: Place the imputed data to their initial position in data matrix $X_{n\times p}$
8: **return** the imputed data matrix $X_{n\times p}^{imp}$

As discussed in Chapter 3 the problem with simple mean imputation is that reduces the variance of the imputed variable and consequently the covariance. When we impute each subset D_i with different values this problem is fixed. Furthermore if we want to create more "realistic" imputed values we can add an error term randomly drawn from a distribution with mean 0 and variance equal to the variance of the recorded values of the corresponding variable in D_i, for example an error term e where $e \sim \mathcal{N}(0, Var_{V_j^{mis}})$.

5.2.1 Experimental Results of Partition Mean imputation

For the experimental results we created various data sets consisted of three 3-dimensional normal distributions in the form of the Figure 5.1. We compare our method with two of the most accurate imputation methods that we have discussed in Chapter 3, namely kNNimpute and missForest, for four different numbers of observations. For evaluation method we use the mean absolute error (MAE).

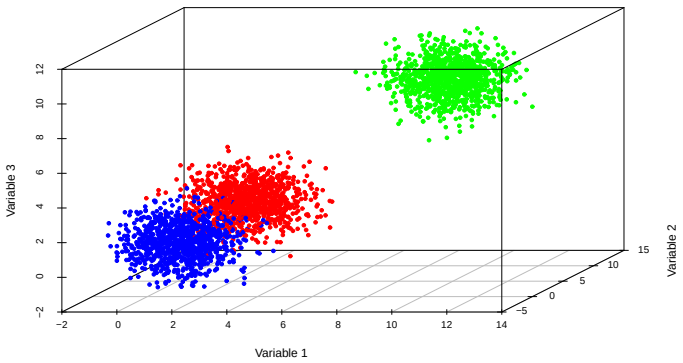


Figure 5.1: Experimental Data sets 3D scatter plot

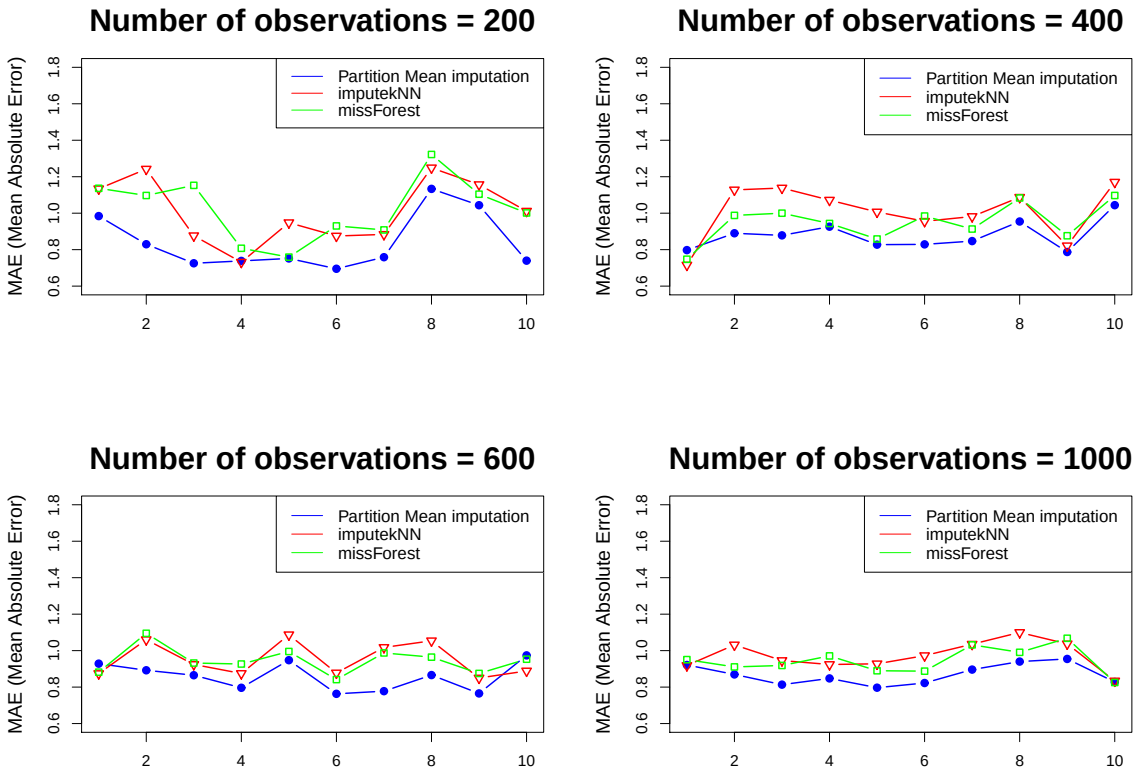


Figure 5.2: Comparison of Partition Mean imputation with kNNimpute and missForest

The data sets in Figure 5.2 have all 10% missing values only for one variable. In almost all cases Partition Mean imputation is the most accurate method. Furthermore we see that as the number of observations increases the accuracy of our method in comparison with the other two methods, increases too.

A common problem while clustering is that in many cases the optimal number of clusters is unknown. There are many methods for choosing the optimal k, though they are not always reliable. In Section 2.3.2 we have described three of the most popular, namely Elbow method, Silhouette

Analysis and Gap statistic. In order to adopt them we can use a simple imputation method to the initial data set. We found that the number of clusters is not an important issue for our method. We selected one from the data sets that compared in Figure 5.2 and we run the algorithm for various k's. The results are depicted in Figure 5.3.

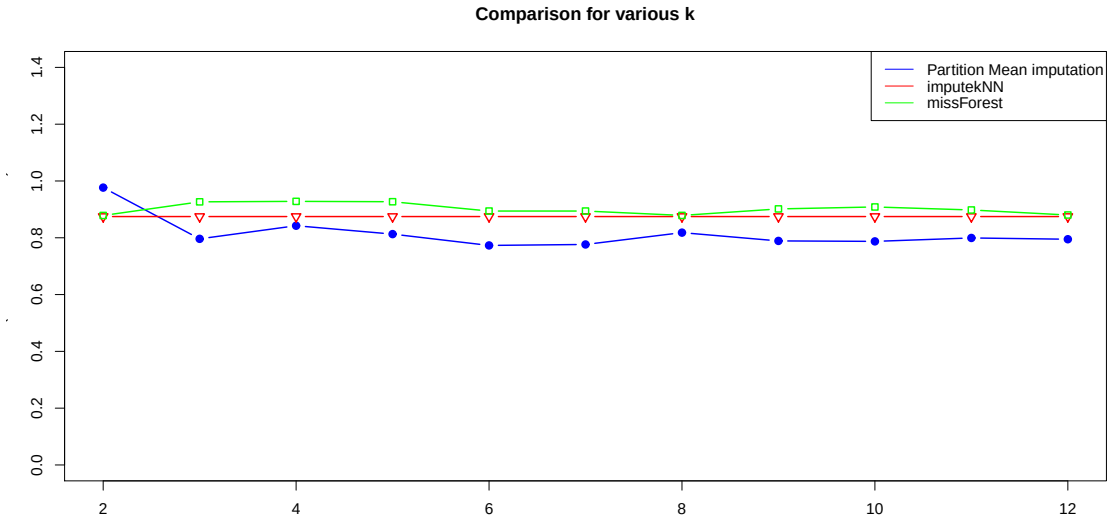


Figure 5.3: Robust to the choice of k

Note that for all values of k that are greater than the real number of clusters (3 clusters), this method performs better, except for the value 2. In the experiments we found that all the values greater than or equal to the real cluster number good choices.

5.3 Partition Regression imputation

Linear Regression imputation is one of the most popular methods with many variations. The classic form of this method is to estimate a Linear Regression model over the complete observations X^{com} and then predict the missing values X^{mis} with the estimated model, though imputing values drawn from a linear equation can increase the covariance of the data. In order to resolve this issue we can use stochastic regression imputation, which adds a error term drawn randomly from a normal distribution with mean 0 and variance equal to the residuals variance of the estimated model. A more sophisticated version of a Linear Regression imputation is to use chained equations.

Chained equation process[1]

1. The procedure begins with a simple imputation for every missing value in the data set. For each missing variable V_j^{mis} we replace the missing elements with random draws from the observed values. For each replaced value we store the position.

2. The observed values from the missing variable V_j^{mis} are regressed on the other variables in the imputed model.

3. The missing values of the missing variable V_j^{mis} are replaced with predictions from the estimated regression model.

[1]Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work?. International journal of methods in psychiatric research, 20(1), 40-49.

4. Steps 2 and 3 are repeated for each missing variable V_j^{mis}. The completion of this process for all missing variables consists an iteration. The iterations of this process are usually 5-10, with the imputations being updated at each cycle.

Suppose now that we have the two dimensional data set depicted in Figure 5.4. If we use Linear Regression imputation we see that the predictions will be far away from the real values. Our approach divides this data set into k data sets (here $k = 3$) and estimates the missing values with k different Linear Regressions. As we can see in Figure 5.5 we can divide the data set into 3 non overlapping data sets and estimate three different Linear Regression models. It's easy to see that this approach increases the accuracy of the Linear Regression imputation method.

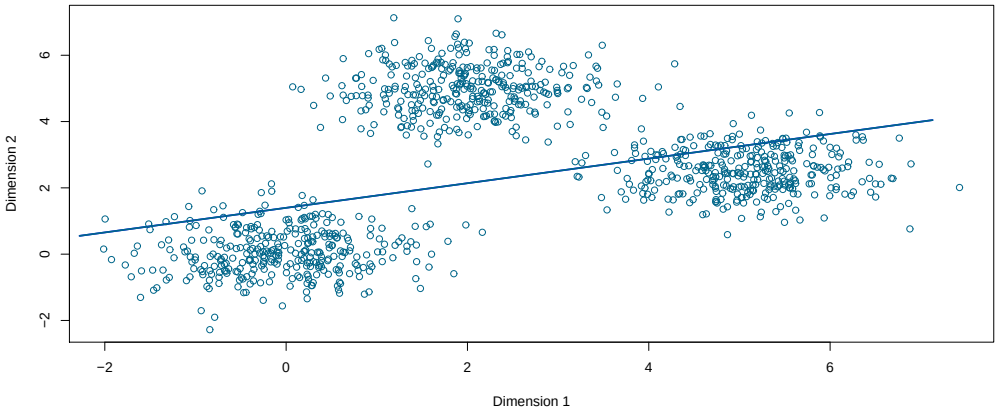


Figure 5.4: Whole data set Linear Regression

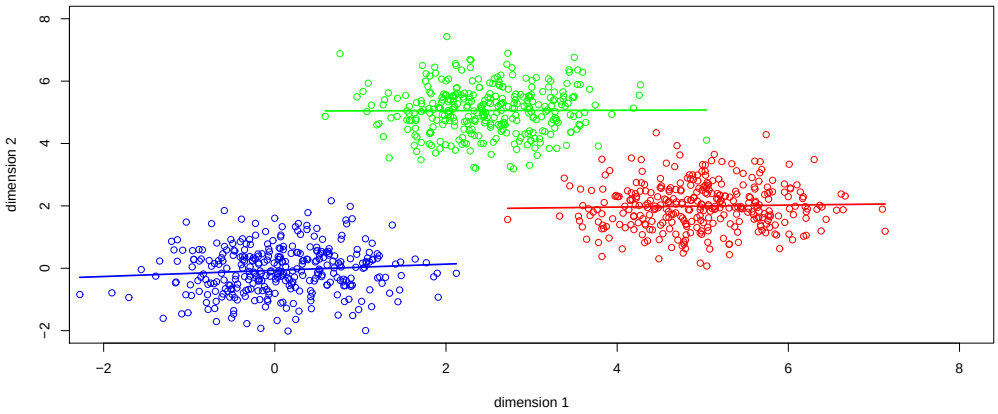


Figure 5.5: Subsets Linear Regression

This partition of the data set D into three non overlapping subsets D_1, D_2 and D_3 can be achieved by the KSC algorithm. The partition Regression imputation divides the data set D into k non overlapping subsets $D_1, D_2, \ldots D_k$, each subset D_i is treated as an individual data set where the missing values are imputed by a Linear Regression model, and specifically the chained equation approach of this method. The imputed values are placed back to their initial positions in the data matrix. This algorithmic procedure is described below.

Algorithm 8 Partition Regression imputation

1: **Require:** $X_{n\times p}$ a data set matrix with at least one completed variable V_j^{com}
2: Run KSC-algorithm 20 times and identify the best partition $\{D_1, D_2, \ldots, D_k\}$
3: **for** $i \leftarrow 1$ to k **do**
4: Store the initial data matrix $X_{n\times p}$ positions for each missing element in D_i
5: Perform chained equation Linear Regression imputation in D_i
6: **end for**
7: Place the imputed data to their initial position in data matrix $X_{n\times p}$
8: **return** the imputed data matrix $X_{n\times p}^{imp}$

Note that when we divide the data set into k parts we estimate k different linear regression equations, this approach can fix the problem of high correlated data. Furthermore we can add an error term as described above.

The significance of the Chained equation process is depicted in Figure 5.6

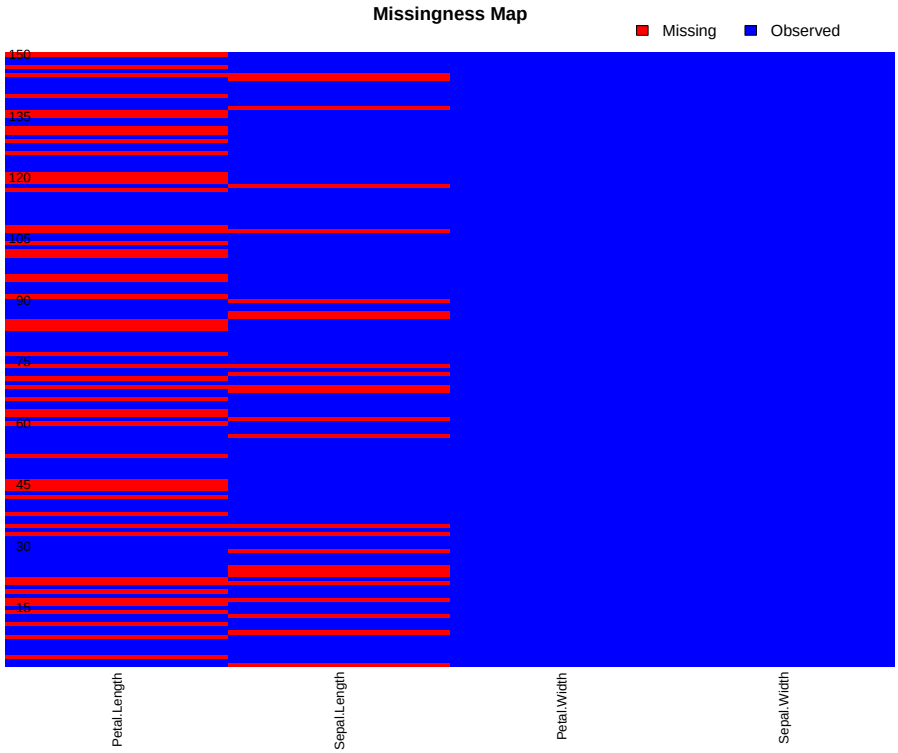


Figure 5.6: Missingness Pattern of two missing variables

Suppose that we have the missing pattern of the Figure 5.6. We see that the missing proportion observations is almost 80% of the data set. Simple Regression imputation uses only values from X^{com}, thus we will loose valuable information and consequently the Linear Regression model will be unreliable.

5.3.1 Experimental Results of Partition Regression imputation

For the experimental results we use the Iris data set[2] (consists of 150 observations and 4 variables). We created randomly increasing proportion of missing values for every set of two out of four variables involved. We compare Partition Regression imputation with kNN imputation and miss Forest using as evaluation method the mean absolute error (MAE). For the KSC algorithm we used a $k = 3$ and $w = 0.5$ (equal to the proportion of missing variables).

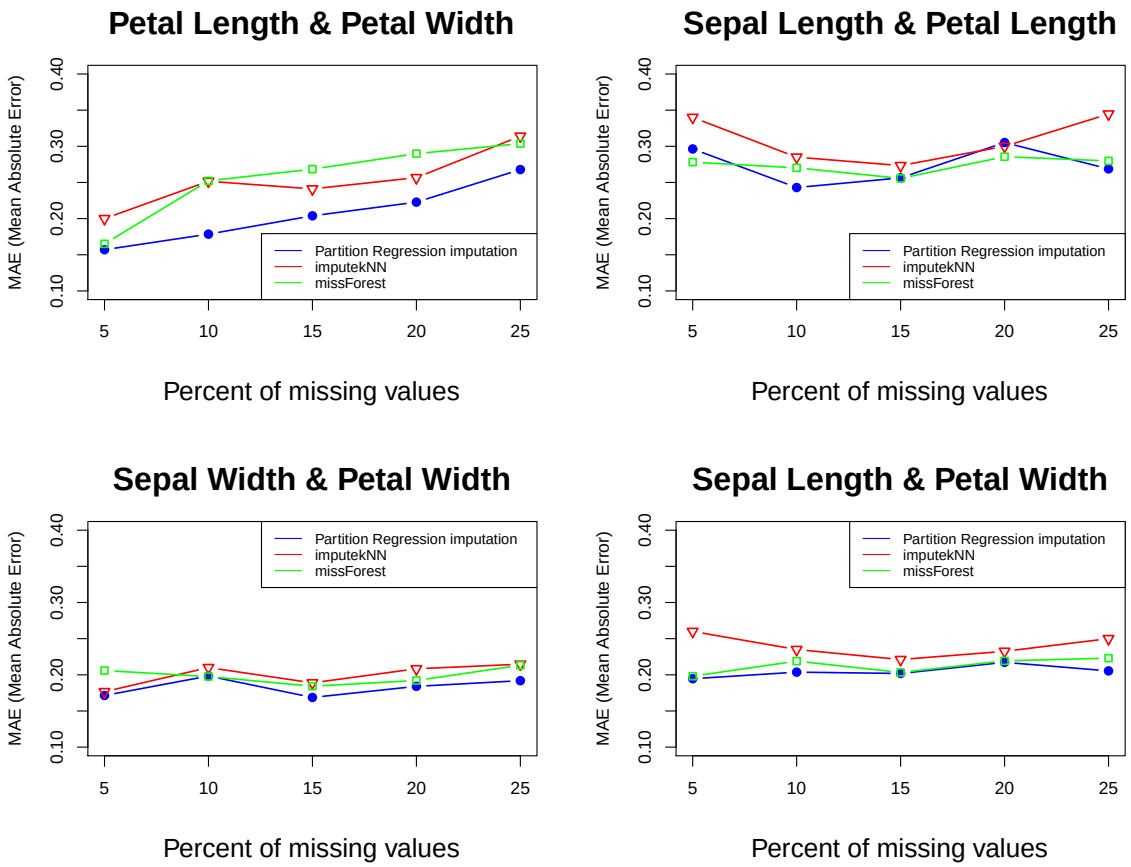


Figure 5.7: Comparison of Partition Regression imputation with kNNimpute and missForest for two missing variables

In Figure 5.7 we compare Partition Regression imputation method with impute kNN and miss Forest using as evaluation criterion the Mean Absolute Error. In the fist plot of the Figure where the missing variables are Petal Length and Petal Width we see that for all proportions of missing values our method is the most accurate. Furthermore for the other three plots in almost all the cases the MAE of our method is smaller and consequently the accuracy is greater.

In Figures 5.8 and 5.9 we see the densities of the original (blue) and the imputed data (red) for 10% proportion of missing values when both of the variables Petal Length and Petal Width are missing.

[2]Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7, Part II, 179–188.

The data were collected by Anderson, E. (1935). The irises of the Gaspe Peninsula, Bulletin of the American Iris Society, 59, 2–5.

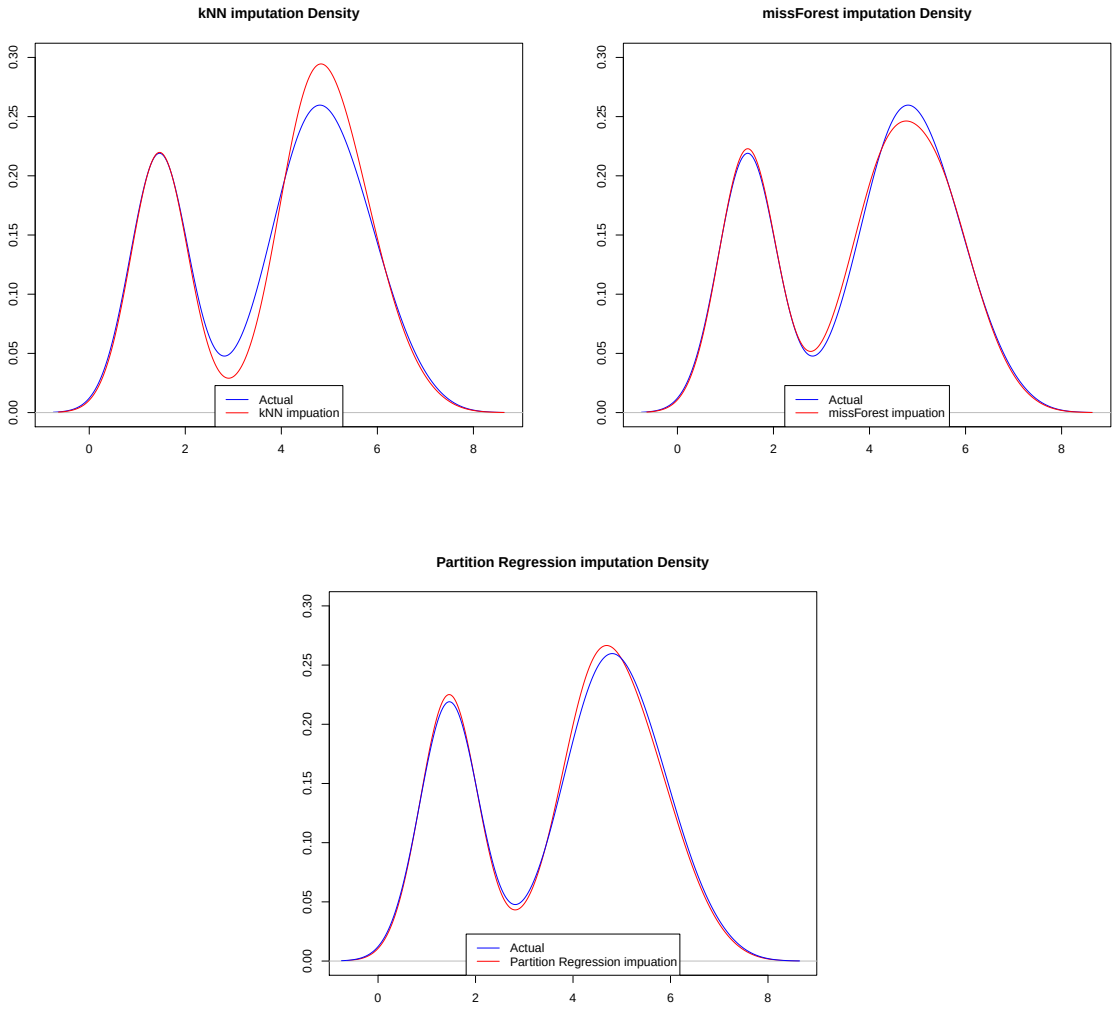


Figure 5.8: Densities of the imputed data for the variable Petal Length

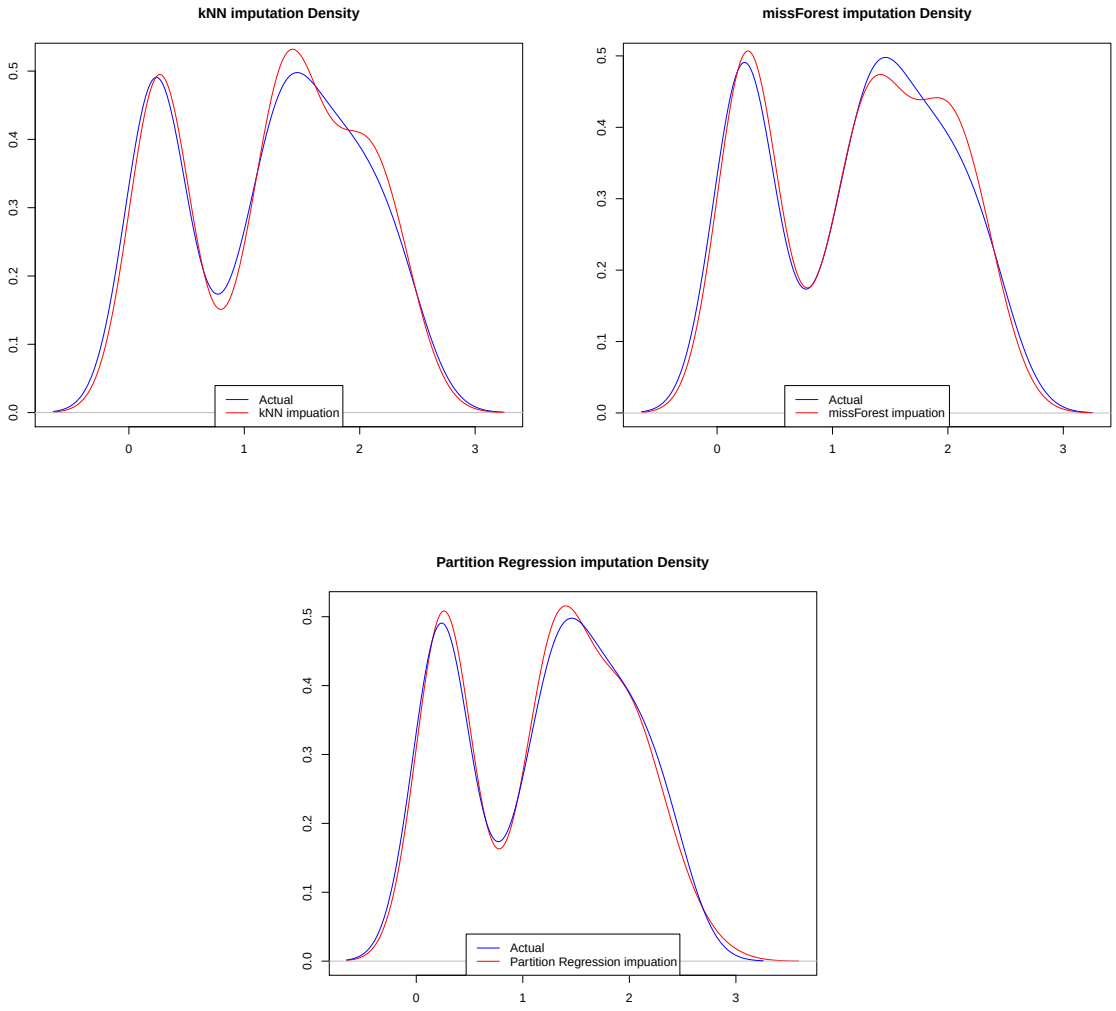


Figure 5.9: Densities of the imputed data for the variable Petal Width

5.4 Discussion

A question concerning the above methods is which method is the most accurate. We propose Partition Mean imputation in the case of clusters with smaller amount of data and Partition Regression imputation in the case of clusters with larger amount of data. For example suppose that we have a cluster with only 5 observations with missing values. It is not plausible to estimate with sufficient accuracy a linear regression model with such a small amount of data with missing values. On the other hand the idea of using the mean is closely related to the idea of imputekNN. Furthermore we choose to compare our methods with imputekNN and miss Forest due to the fact that all approaches to the problem of missing values are somewhat similar. The imputekNN method creates a region of the k closest neighbors of each missing observation and impute to each missing value a weighted mean of them. Random Forest estimates the mean of high dimensional rectangulars of 100 different regression trees and then impute the average of all these regions where the missing observation falls in. In Partition Mean imputation we create regions of observations (clusters) and we impute the mean for each missing variable, and in Partition Regression imputation in order to increase the accuracy, we use linear regression imputation.

Future Work

For future work we propose the use of more imputation methods for each D_i of the final partition of the KSC algorithm, such as predictive mean matching and the creation of an evaluation criterion for choosing the most accurate method.

Bibliography

[1] Theodore Wilbur Anderson. *An introduction to multivariate statistical analysis*, volume 2. Wiley New York, 1958.

[2] David F Andrews. Plots of high-dimensional data. *Biometrics*, pages 125–136, 1972.

[3] Donna Pauler Ankerst. *DANIEL ZELTERMAN, Applied Multivariate Statistics with R. Heidelberg: Springer*. Wiley Online Library, 2017.

[4] Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49, 2011.

[5] Sugato Basu, Arindam Banerjee, and Raymond Mooney. Semi-supervised clustering by seeding. In *In Proceedings of 19th International Conference on Machine Learning (ICML-2002*. Citeseer, 2002.

[6] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[7] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[8] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[9] Stef Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45(3), 2011.

[10] Christopher Chatfied and Alexander J Collins. *Introduction to multivariate analysis*. Springer, 2013.

[11] Καρλής Δημήτρης. *Πολυμεταβλητή Στατιστική Ανάλυση*. 2005.

[12] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[13] Craig K Enders. *Applied missing data analysis*. Guilford Press, 2010.

[14] Brian Everitt and Torsten Hothorn. *An introduction to applied multivariate analysis with R*. Springer Science & Business Media, 2011.

[15] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[16] Wolfgang Härdle and Léopold Simar. *Applied multivariate statistical analysis*. Springer Science & Business Media, 2007.

[17] Trevor Hastie, Robert Tibshirani, Gavin Sherlock, Michael Eisen, Patrick Brown, and David Botstein. Imputing missing data for gene expression arrays, 1999.

[18] Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

[19] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[20] Alan Julian Izenman. *Modern multivariate statistical techniques*, volume 1. Springer, 2008.

[21] Gareth James, Daniela Witten, and Trevor Hastie. An introduction to statistical learning: With applications in r., 2014.

[22] Alboukadel Kassambara. *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*, volume 1. STHDA, 2017.

[23] RJ Little. *A and Rubin, DB (1987) Statistical Analysis with Missing Data*. 85.

[24] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2002.

[25] James MacQueen et al. *Some methods for classification and analysis of multivariate observations*, volume 1. 1967.

[26] Kantilal V Mardia, John T Kent, and John M Bibby. *Multivariate analysis (probability and mathematical statistics)*. Academic Press London, 1980.

[27] Geert Molenberghs, Garrett Fitzmaurice, Michael G Kenward, Anastasios Tsiatis, and Geert Verbeke. *Handbook of missing data methodology*. CRC Press, 2014.

[28] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[29] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[30] Donald B Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

[31] Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2011.

[32] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[33] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[34] Stef Van Buuren. Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical methods in medical research*, 16(3):219–242, 2007.

[35] Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2012.

[36] Kiri Wagstaff. Clustering with missing values: No imputation required. *Classification, Clustering, and Data Mining Applications*, pages 649–658, 2004.

[37] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

[38] Kiri L Wagstaff and Victoria G Laidler. Making the most of missing values: Object clustering with partial data in astronomy. In *Astronomical Data Analysis Software and Systems XIV*, volume 347, page 172, 2005.

[39] Kiri Lou Wagstaff and Claire Cardie. *Intelligent clustering with instance-level constraints*. Cornell University USA, 2002.