الجمهورية الجزائرية الديمقراطية الشعبية

Ministry of Higher Education and Scientific Research
UNIVERSITY SAAD DAHLEB DE BLIDA
Faculty of Sciences
Mathematics Department

MASTER's THESIS
In mathematics

Option : Stochastic and Statistical Models

THEME :

Imputation of missing data and Inference by EM algorithm

Realised by
ABBACI Safaa & SAIDI Manel

Argued before the Jury:

TAMI Omar	University Blida 1	Chairmann
FRIHI Redhouane	University Blida 1	Examinator
RASSOUL Abdelaziz	ENSH of Blida	Advisor

July 2022

DEDICATIONS

To

To my homeland Algeria....

To The great martyrs and prisoners symbol of sacrifice...

To me.

Manel SAIDI

DEDICATIONS

This entire journey would not have been possible without the support of my family, professors, advisors, and friends.

Thank you to my family my father and my mother my only sister Ines and my brother Abd el basset for encouraging me all the time and this effort motivates me to pursue my dreams. I am especially grateful for the reason that my parents supported me cognitively and financially.I know they have always believed that I can do the best for myself. Thanks to my parents for guiding me to become the person I am today. and thanks for my cats Alex and zaatar because they make me smile all the times

I have to thank all the professors of the Mathematics Department with whom I have been working for the last years. Thank you for supporting my work and helping me to explore.

I am indebted to my mentor RASSOUL , who generously guided me to overcome all the difficulties. I thank him for him continued support and guidance throughout the time.

Thanks to my roommates and friends fadhila and rasha and ikram and asma and wided and nabila for their advice and attention. specialy for Manel SAIDI, They are the people who have been my pillar of strength all over my whole graduation journey. Your good wishes for me help me to achieve my dreams.

Safaa ABBACI

ACKNOWLEDGMENTS

First of all, we would like to thank ALLAH, the almighty and merciful gods, who gave me the will, the strength and the health to complete this dissertation. and Mister My great teacher and messenger Muhammad (may God bless him and grant him peace) who taught us the purpose of life

we express our deep gratitude and sincere thanks to our Professor RASSOUL Abdelaziz, for agreeing to direct this work, his advice, his corrections, his kindness and his patience as well as for the time he devoted created for the completion of this dissertation.

our sincere thanks to the members of the jury for their presence, for their careful study of this thesis, as well as for the remarks they made to us during of this defense in order to improve our work. These members are:

Mr. TAMI Omar, who did me the honor of presiding over the jury for the defense of my thesis, and

Mr. FRIHI Redouane who examines my dissertation.

I would like to thank my dear father and my dear mother for their support and encouragement throughout my academic career.

I would also like to express my gratitude to all the teachers who have contributed to my training throughout my academic career.

Finally, I would like to thank all the people who helped me directly or indirectly in the development this memory.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ملخص :

البيانات المفقودة هي مشكلة رئيسية في العديد من المشاكل التطبيقية. في هذا العمل، قمنا بدراسة البيانات المفقودة وأهداف التضمين المتعدد مقارنة بالتضمين الفردي. قمنا ايضا بدراسة الاستدلال الإحصائي لنموذج من البيانات المفقودة بناء على طريقة تقدير الاحتمال باستخدام خوارزمية تحقيق اقصى قدر للتوقع. أخيرًا، قدمنا معظم الوحدات اللازمة لاحتساب البيانات المفقودة وطبقنا خوارزمية EMلنموذج خليط غاوسي GMM.

الكلمات المفتاحية : خوارزمية اقصى قدر للتوقع، البيانات المفقودة، الاستدلال، طريقة تقدير الاحتمال

Abstract

Missing data is a major issue in many applied problems. in our work we examine data that are missing and . the aims of multiple imputation in comparison to single imputation, we also studying the statistical inference by likelihood Maximum method for sample with missing data with Maximization-Expectation algorithm. Finally, we present the mains packages for imputation of missing data, and we applied the EM algorithm for Mixture Gaussian model

Keywords: expectation maximization algorithm, missing data, imputation, maximum likelihood Method

Résumé

Les données manquantes sont un problème majeur dans de nombreux problèmes appliqués. Dans notre travail, nous avons examiné les données manquantes et les objectifs de l'imputation multiple par rapport à l'imputation simple. Nous avons également étudié l'inférence statistique en basant sur la méthode de la vraisemblance maximale (maximum-likelihood estimation) pour un exemple avec des données manquantes en utilisant l'algorithme Espérance-maximisation. Finalement, nous avons présenté la majorités des modules nécessaires pour l'imputation des données manquantes et nous avons appliqué l'algorithme EM pour le modèle de mélange Gaussien (Gaussian Mixture Model)

Mots-clés: algorithme espérance maximisation, données manquantes, imputation, vraisemblance maximale

GENERAL INTRODUCTION

In most situations, simple techniques for handling missing data (such as complete case analysis, overall mean imputation, and the missing-indicator method) produce biased results, whereas imputation techniques yield valid results without complicating the analysis once the imputations are carried out. Imputation techniques are based on the idea that any subject in a study sample can be replaced by a new randomly chosen subject from the same source population. Imputation of missing data on a variable is replacing that missing by a value that is drawn from an estimate of the distribution of this variable. In single imputation, only one estimate is used. In multiple imputation, various estimates are used, reflecting the uncertainty in the estimation of this distribution. Under the general conditions of so-called missing at random and missing completely at random, both single and multiple imputations result in unbiased estimates of study associations. But single imputation results in too small estimated standard errors, whereas multiple imputation results in correctly estimated standard errors and confidence intervals.Data that we plan to analyze are often incomplete. Study design strategies should ideally be set up to obtain complete data in the first place through questionnaire design, interviewer training, study protocol development, real-time data checking, or re-contacting participants to obtain complete data. When obtaining complete data is not feasible, proxy reports or the collection of characteristics associated with the missing values can help. Missing data can be categorized in multiple ways. Perhaps the most troubling are the data missing on entire observations (e.g., due to selection bias) or on entire variables that have been omitted from the study design. Somewhat more tractable, but still potentially problematic, are data missing on a subset of variables that are missing for a subset of the observations. In this case, it can be useful to label those observations without missing data as "complete cases" and those with some missing data as "partial cases." Ideally, we hope that the amount of missing data is limited, in which case we will rely less heavily on our assumptions about the pattern of missing data.

Missing data can bias study results because they distort the effect estimate of interest . Missing data are also problematic if they decrease the statistical power by effectively decreasing the sample size, or if they complicate comparisons across models that differ in both the analysis strategy and the number of included observations.Researchers usually address missing data by including in the analysis only complete cases those individuals who have no missing data in any of the variables required for that analysis. However, results of such analyses can be biased. Furthermore, the cumulative effect of missing data in several variables often leads to exclusion of a substantial proportion of the original sample, which in turn causes a substantial loss of precision and power. we used Expectation-Maximization (EM) algorithm, (EM) algorithm is a popular algorithm for obtaining maximum likelihood (ML) estimates. in cases of missing data, Here we present the main steps for exploring the EM algorithm for the factor analysis model. This algorithm extends a previously proposed EM algorithm to handle problems with missing data. It is simple to implement and is the most storage efficient among its competitors.the basic idea of the EM algorithm is to associate with the given incomplete-data problem, a complete-data problem for which ML estimation is computationally more tractable; for instance, the complete data problem chosen may yield a closed form solution to the maximum likelihood estimate (MLE) or may be amenable to MLE computation with a standard computer package. The methodology of the EM algorithm consists in reformulating the problem in terms of this more easily solved complete-data problem, establishing a relationship between the likelihoods of these two problems, and exploiting the simpler MLE computation of the complete-data problem in the M-step of the iterative computing algorithm

our work are present as follow:

- chapter 1

  We talk about causes of missing data,Ignorable Missing Value and the Missingness Mechanism and patterns and we give in this chapter diffrent methods for Handling Missing Data.

- chapter 2

  In this chapter we present the EM algorithm and his history, also his properties. and the relation between EM algorithm and Newton-Raphson method, we delve into the details E- and M-Steps of the regular exponential family and take an example of Censored Exponentially Distributed Survival Times,also Generalized Expectation-Maximization Algorithm(GEM), And where is used EM Algorithm by Newton-Raphson Step. Finally we show how to estimate the parameters of a bivariate normal distribution based on a sample from this distribution even when some of the data is missing

- chapter 3

  We give various examples of applications of the EM algorithm to the resolution of some problems of missing data, we have list of 5 R packages popularly known for missing value imputation. There might be more packages. But, we decided to focus on these ones. we have tried to explain the concepts in simplistic manner with practice examples of Imputation of missing data and Inference by EM algorithm in R. We complete this work with a general conclusion.

CHAPTER 1

TREATMENT OF THE PROBLEM OF MISSING-DATA

1.1 Missing data

In statistics, missing data, or missing values, occur when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data.

Missing data can occur because of non response: no information is provided for one or more items or for a whole unit ("subject"). Some items are more likely to generate a non response than others: for example items about private subjects such as income. Attrition is a type of missingness that can occur in longitudinal studies for instance studying development where a measurement is repeated after a certain period of time. Missingness occurs when participants drop out before the test ends and one or more measurements are missing.

Data often are missing in research in economics, sociology, and political science because governments or private entities choose not to, or fail to, report critical statistics, or because the information is not available. Sometimes missing values are caused by the researcher for example, when data collection is done improperly or mistakes are made in data entry.

These forms of missingness take different types, with differents impacts on the validity of conclusions from research: Missing completely at random, missing at random, and missing not at random. Missing data can be handled similarly as censored data.

Understanding the reasons why data are missing is important for handling the remaining data correctly. If values are missing completely at random, the data sample is likely still representative of the population. But if the values are missing systematically, analysis may be biased. For example, in a study of the relation between IQ and income, if participants with an above-average IQ tend to skip the question 'What is your salary?', analyses that do not take into account this missing at random (MAR pat-

tern (see below)) may falsely fail to find a positive association between IQ and salary. Because of these problems, methodologists routinely advise researchers to design studies to minimize the occurrence of missing values. Graphical models can be used to describe the missing data mechanism in detail.

1.2 Missing values

1.2.1 Introduction

The problem of missing values exists since the earliest attempts of exploiting data as a source of knowledge, as it lies intrinsically in the process of obtaining, recording, and preparation of the data itself. Clearly, (citing Gertrude Mary Cox) "The best thing to do with missing values is not to have any", but in the contemporary world, considering the increasingly growing amount of accessible data and demand in statistical justification this is not always the case, nay never. Main references on missing values include Schafer (1997) [1], Little and Rubin (1987, 2002) [2], van Buuren (2012) [3], Carpenter and Kenward (2013) [4] and (Gelman and Hill, 2007)[chp25][5]

Missing values occur for plenty of reasons: machines that fail, individuals who forget or do not want to answer to some questions of a questionnaire, damaged plants, etc. They are problematic since most statistical methods cannot be applied on a incomplete dataset. In this chapter we review the different types of missing data and statistical methods which allow their incorporation.

1.2.2 Ignorable missing values

Many statistical methods are based on estimating a parameter by maximizing the likelihood of the data. Assume that X has a density, parameterized by some parameter θ that we want to estimate, if X is Gaussian for instance we simply have $\theta = (\mu, \Sigma)$. Assume that M also has a density parameterized by another parameter ϕ for example the probability p of a Bernoulli distribution. In some cases, estimating θ from an incomplete data can be done in a very simple way by ignoring, or "skipping" the missing data, as detailed below.

We denote by $f(X, M|\theta, \phi)$ the joint density of the observed and missing entries and of the indicator of missingness conditioned on parameters θ and ϕ. In the context of maximum likelihood estimation, we maximize with respect to θ the marginal density of the observed data X_{OBS} and we have the missing data X_{MIS}

$$f(X_{\text{OBS}}, M|\theta, \phi) = \int f(X_{\text{OBS}}, X_{\text{MIS}}, M|\theta, \phi) dX_{\text{MIS}}.$$

If the data are MAR (or MCAR), the following factorization holds

$$f(X_{\text{OBS}}, X_{\text{MIS}}, M|\theta, \phi) = f(X_{\text{OBS}}, X_{\text{MIS}}|\theta) f(M|X_{\text{OBS}}, \phi).$$

Plugging this in the expression of the marginal density we obtain

$$f(X_{\text{OBS}}, M|\theta, \phi) = \int f(X_{\text{OBS}}, X_{\text{MIS}}|\theta) f(M|X_{\text{OBS}}, \phi) dX_{\text{MIS}},$$

$$f(X_{\text{OBS}}, M|\theta, \phi) = f(M|X_{\text{OBS}}, \phi) \int f(X_{\text{OBS}}, X_{\text{MIS}}|\theta) dX_{\text{MIS}},$$

$$f(X_{\text{OBS}}, M|\theta, \phi) = f(M|X_{\text{OBS}}, \phi) f(X_{\text{OBS}}|\theta) \tag{1.1}$$

If ϕ and θ are distinct (the joint parameter space of (θ, ϕ) is the product of the parameter space of θ and the parameter space of ϕ), as the term $f(M|X_{\text{OBS}}, \phi)$ is respect to θ, it is equivalent to maximize the likelihood $f(X_{\text{OBS}}|\theta)$, i.e. to ignore the missing data. It really means that when doing inference, i.e. to get the ML estimates for parameters from an incomplete set, one can "simply" maximizes the observed likelihood while ignoring the process that have generated missing values. Consequently, most of the methods used in practice relie on the assumption that the data are MAR.

1.3 Missing data mechanisms: MCAR, MAR, MNAR

There are several types of missing data, and explaining the reasons why part of the data is missing is crucial to perform inference or any kind of statistical analysis. Dealing with missing data boils down to considering that the observed data X_{OBS} is only a subset of a complete data model $X = (X_{\text{OBS}}, X_{\text{MIS}})$ which is not fully observable (i.e. X_{MIS} are the missing data). Assume $X = (X_1, \ldots, X_n)$; the missing values X_{MIS} are characterized by a set of indices $I_{\text{MIS}} \subset \{1, \ldots, n\}$ such that $X_{\text{MIS}} = \{X_i; i \in I_{\text{MIS}}\}$. We define the indicator of missingness $M \in \{0, 1\}^n$ such that $M_i = 1$ if $i \in I_{\text{MIS}}$ and $M_i = 0$ otherwise; M defines the of missingness. Both X and M are modeled as random variables with probability distributions $\mathbb{P}_X$ and $\mathbb{P}_M$ respectively. The different types of missing data refer to different dependence relationships between $X_{\text{OBS}}, X_{\text{MIS}}$ and M.

The observations are said to be Missing Completely At Random (MCAR) if the probability that an observation is missing is independent of the variables and observations in the dataset:

the probability that an observation is missing does not depend on $(X_{\text{OBS}}, X_{\text{MIS}})$. Formally:

$$\mathbb{P}_M(M|X_{\text{OBS}}, X_{\text{MIS}}) = \mathbb{P}_M(M). \tag{1.2}$$

The observations are said to be missing at random (MAR) if the probability that an

observation is missing only depends on the observed data X_{OBS}. Formally:

$$\mathbb{P}_M(M|X_{\text{OBS}}, X_{\text{MIS}}) = \mathbb{P}_M(M|X_{\text{OBS}}). \tag{1.3}$$

The observations are said to be Missing Not At Random (MNAR) in all other cases.

1.3.1 Missing Completely at Random (MCAR)

The MCAR case is observed when the possibility of a feature variable having missing data entries is independent of the feature variable itself or of any of the other feature variables within the data set. Essentially, this means that the missing data entry does not depend on the feature variable being considered or any of the other feature variables in the data set. This relationship is expressed mathematically as Little and Rubin (2014) [6]

$$\mathbb{P}_M(M|X_{\text{OBS}}, X_{\text{MIS}}) = \mathbb{P}_M(M)$$

where $M \in \{0, 1\}$ represents an indication of the missing value. $M = 1$ if Y is known and $M = 0$ if Y is unknown/missing. Y_o represents the observed values in Y while Y_m represents the missing values of Y. From **??**, the probability of a missing entry in a variable is not related to Y_o or Y_m. For instance, let us assume that in modelling software defects in relation to development time, if the missingness is in no way linked to the missing values of the rate of defects itself and at the same time not linked to the values of the development time, the data is said to be MCAR. Researchers have successfully addressed cases where the data is MCAR.Silva-Ramirez et al. (2011)[7] successfully applied multilayer perceptrons (MLPs) for missing data imputation in datasets with missing values. Other research work done on this mechanism could be found in Pigott (2001)[8], Nishanth and Ravi (2013)[9].

Example

We want to assess which are the main determinants of income (such as age). The MCAR assumption would be violated if people who did not report their income were, on average, younger than people who reported it. This can be tested by dividing the sample into those who did and did not report their income, and then testing a difference in mean age. If we fail to reject the null hypothesis, then we can conclude that the MCAR is mostly fulfilled (there could still be some relationship between missingness of Y and the values of Y).

1.3.2 Missing at random (MAR)

The MAR case is observed when the possibility of a specific feature variable having missing data entries is related to the other feature variables in the data set. However,this missing data does not depend on the feature variable itself. MAR means the missing data in the feature variable is conditional on any other feature variable in the data set but not on that being considered (Scheffer 2000)[10]. For example, consider a data set with two related variables, monthly expenditure and monthly income. Assume for instance that all high-income earners deny revealing their monthly expenditures while low-income earners do provide this information. This implies that in the data set, there is no monthly expenditure entry for high-income earners, while for low-income earners, the information is available. The missing monthly income entry is linked to the income earning level of the individual. This relationship can be expressed mathematically as Marwala (2009)[11]:

$$\mathbb{P}_M(M|X_{\text{OBS}}, X_{\text{MIS}}) = \mathbb{P}_M(M|X_{\text{OBS}}).$$

where $M \in \{0, 1\}$ is the missing data indicator, and $M = 1$, if Y is known, with $M = 0$. if Y is unknown/missing. Y_o represents the observed values in Y while Y_m represents the missing values of Y. Equation **??** indicates that the probability of a missing entry given an observable entry and a missing entry is equivalent to the probability of the missing entry given the observable entry only. the software defects might not be revealed because of a certain development time. Such a scenario points to the data being MAR. Several studies have been conducted in the literature where the missing data mechanism is MAR., for example Nelwamondo et al. (2007b)[12] performed a study to compare the performance of expectation maximization and a GA-optimized AANN and it was revealed that the AANN is a better method than the expectation maximization. Further research on this mechanism was performed in Garca-Laencina et al. (2009)[13], Poleto et al. (2011)[14], Liu and Brown (2013)[15]

The probability of missing data on Y is unrelated to the value of Y after controlling for other variables in the analysis (say X). Formally:

$$P(Y_{missing}|Y, X) = P(Y_{missing}|X).[16]$$

Example: The MAR assumption would be satisfied if the probability of missing data on income depended on a person's age, but within age group the probability of missing income was unrelated to income. However, this cannot be tested because we do not know the values of the missing data, thus, we cannot compare the values of those with and without missing data to see if they systematically differ on that variable.

1.3.3 Not missing at random (NMAR)

The third missing data mechanism is the missing not at random or non-ignorable case. The MNAR case is observed when the possibility of a feature variable having a missing data entry depends on the value of the feature variable itself irrespective of any alteration or modification to the values of other feature variables in the datasets (Allison 2000)[17]. In scenarios such as these, it is impossible to estimate the missing data by making use of the other feature variables in the dataset since the nature of the missing data is not random. MNAR is the most challenging missing data mechanism to model and these values are quite tough to estimate (Rubin 1978)[18]. Let us consider the same scenario described in the previous subsection. Assume for instance that some high-income earners do reveal their monthly expenditures while others refuse, and the same for low-income earners. Unlike the MAR mechanism, in this instance the missing entries in the monthly expenditure variable cannot be ignored because they are not directly linked to the income variable or any other variable. Models developed to estimate this kind of missing data are very often not biased. A probabilistic formulation of this mechanism is not easy because the data in the mechanism is neither MAR nor MCAR

Example 1:

we can imagine that patients with low blood pressure are more likely to have their blood pressure measured less frequently (the missing data for the variable "blood pressure" partially depends on the values of the blood pressure).

Example 2:

The NMAR assumption would be fulfilled if people with high income are less likely to report their income.

1.3.4 Missing by Natural Design (MBND)

This is a mechanism whereby the missing data occurs because it cannot be measured physically (Marwala 2009)[11]. It is impossible to measure these data entries; however, they are quite relevant in the data analysis procedure. Overcoming this problem requires that mathematical equations be formulated. This missing data mechanism mainly applies to mechanical engineering and natural science problems. Therefore, it will not be used in this thesis for the problem under consideration

1.4 Missing Data Patterns

The way in which missing data occurs can be grouped into three patterns given by Tables 1.1, 1.2, 1.3. Table 1.1 depicts a univariate pattern which is a scenario described by the presence of missing data in only one feature variable as seen in column I7. Table

1.2 depicts an arbitrary missing data pattern, which is a scenario whereby the missing data occurs in a distributed and random manner. The last pattern is the monotone missing data pattern which is shown in Table 1.3. This pattern is also referred to as a uniform pattern as it occurs in cases whereby the missing data can be present in more than one feature variable and, it is easy to understand and recognize (Ramoni and Sebastiani 2001)[19].

Table 1.1: Univariane missing data pattern

sample	11	12	13	14	15	16	17
1	0.38	0.18	0.20	0.19	0.75	0.67	0.96
2	0.69	0.11	0.08	0.41	0.65	0.63	?
3	0.17	0.79	0.66	0.53	0.95	0.43	?
4	0.19	0.24	0.15	0.91	0.46	0.82	?

Table 1.2: Arbitary missing data pattern

sample	11	12	13	14	15	16	17
1	0.38	?	0.20	0.19	0.75	0.67	0.96
2	0.69	0.11	0.08	0.41	?	0.63	0.04
3	0.17	0.79	?	0.53	0.95	0.43	0.054
4	?	0.24	0.15	0.91	0.46	0.82	?

Table 1.3: Monotone missing data pattern

sample	11	12	13	14	15	16	17
1	0.38	0.18	0.20	0.19	0.75	0.67	?
2	0.69	0.11	0.08	0.41	0.65	?	?
3	0.17	0.79	0.66	0.53	?	?	?
4	0.19	0.24	0.15	?	?	?	?

1.5 Methods for handling missing data

1.5.1 Deletion

The Deletion method is used when the probability of missing variable is same for all observations.

Example:

Respondents of data collection process decide that they will declare their earning after tossing a fair coin. If an head occurs, respondent declares his/her earnings & vice versa. Here each observation has equal chance of missing value.

Deletion can be performed in two types: List Wise Deletion and Pair Wise Deletion.

a) In list wise deletion, we delete observations where any of the variable is missing. Simplicity is one of the major advantage of this method, but this method reduces the power of model because it reduces the sample size. For simplicity we can say that, this method deletes the whole row of observations in which the data is missing.

b) In pair wise deletion, we perform analysis with all cases in which the variables of interest are present. Advantage of this method is, it keeps as many cases available for analysis. One of the disadvantage of this method, it uses different sample size for different variables.

Advantages: It can be used with any kind of statistical analysis and no special computational methods are required.

Limitations: It can exclude a large fraction of the original sample. For example, suppose a data set with 1,000 people and 20 variables. Each of the variables has missing data on 5 of the cases, then, you could expect to have complete data for only about 360 individuals, discarding the other 640.

It works well when the data are missing completely at random (MCAR), which rarely happens in reality (Nakai & Weiming, 2011) [20].

Table 1.4: List wise deletion

Gender	manpower	sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
.	26	259
M	32	297

Table 1.5: pair wise deletion

Gender	manpower	sales
M	25	343
F	.	280
M	33	332
M	.	272
F	25	.
M	29	326
.	26	259
M	32	297

1.5.2 Imputation methods

Substitute each missing value for a reasonable guess, and then carry out the analysis as if there were not missing values.

There are two main imputation techniques:

- **Marginal mean imputation:** Compute the mean of X using the non-missing values and use it to impute missing values of X. Limitations: It leads to biased estimates of variances and covariances and, generally, it should be avoided.

- **Conditional mean imputation:** Suppose we are estimating a regression model with multiple independent variables. One of them, X, has missing values. We select those cases with complete information and regress X on all the other independent variables. Then, we use the estimated equation to predict X for those cases it is missing.

  If the data are MCAR, least-squares coefficients are consistent (i.e. unbiased as the sample size increases) but they are not fully efficient (remember, efficiency is a measure of the optimality of an estimator. Essentially, a more efficient estimator, experiment or test needs fewer samples than a less efficient one to achieve a given performance). Estimating the model using weighted least squares or generalized least squares leads to better results (Graham, 2009[21] (Allison, 2001)[16] and (Briggs et al., 2003)[22].

1.5.3 Mean/ Mode/ Median Imputation

Imputation is a method to fill in the missing values with estimated ones. The objective is to employ known relationships that can be identified in the valid values of the data set to assist in estimating the missing values. Mean / Mode / Median imputation is one of the most frequently used methods. It consists of replacing the missing data for

a given attribute by the mean or median (quantitative attribute) or mode (qualitative attribute) of all known values of that variable. It can be of two types:

1. **Generalized Imputation:** In this case, we calculate the mean or median for all non missing values of that variable then replace missing value with mean or median. Like in above table, variable "Manpower" is missing so we take average of all non missing values of "Manpower" (28.33) and then replace missing value with it.

2. **Similar case Imputation:** In this case, we calculate average for gender "Male" (29.75) and "Female" (25) individually of non missing values then replace the missing value based on gender. For "Male", we will replace missing values of manpower with 29.75 and for "Female" with 25.

1.5.4 Prediction Model

Prediction model is one of the sophisticated method for handling missing data. Here, we create a predictive model to estimate values that will substitute the missing data. In this case, we divide our data set into two sets: One set with no missing values for the variable and another one with missing values. First data set become training data set of the model while second data set with missing values is test data set and variable with missing values is treated as target variable. Next, we create a model to predict target variable based on other attributes of the training data set and populate missing values of test data set.We can use regression, ANOVA, Logistic regression and various modeling technique to perform this. There are 2 drawbacks for this approach:

1. The model estimated values are usually more well-behaved than the true values

2. If there are no relationships with attributes in the data set and the attribute with missing values, then the model will not be precise for estimating missing values.

1.5.5 K-Nearest Neighbour(KNN) Imputation

In this method of imputation, the missing values of an attribute are imputed using the given number of attributes that are most similar to the attribute whose values are missing. The similarity of two attributes is determined using a distance function. It is also known to have certain advantage and disadvantages.

Advantages:

1. k-nearest neighbour can predict both qualitative & quantitative attributes

2. Creation of predictive model for each attribute with missing data is not required

3. Attributes with multiple missing values can be easily treated

4. Correlation structure of the data is taken into consideration

Disadvantage:

1. KNN algorithm is very time-consuming in analyzing large database It searches through all the dataset looking for the most similar instances.

2. Choice of k-value is very critical. Higher value of k would include attributes which are significantly different from what we need whereas lower value of k implies missing out of significant attributes.

1.5.6 Multiple Imputation

The imputed values are draws from a distribution, so they inherently contain some variation. Thus, multiple imputation (MI) solves the limitations of single imputation by introducing an additional form of error based on variation in the parameter estimates across the imputation, which is called "between imputation error". It replaces each missing item with two or more acceptable values, representing a distribution of possibilities (Allison, 2001)[16].

MI is a simulation-based procedure. Its purpose is not to re-create the individual missing values as close as possible to the true ones, but to handle missing data to achieve valid statistical inference (Schafer, 1997)[1]

It involves 3 steps:

a) Running an imputation model defined by the chosen variables to create imputed data sets. In other words, the missing values are filled in m times to generate m complete data sets. $m = 20$ is considered good enough. Correct model choices require considering:

   ★ Firstly, we should identify which are the variables with missing values.

   ★ Secondly, we should compute the proportion of missing values for each variable.

   ★ Thirdly, we should assess whether different missing value patterns exist in the data (SAS [23] helps us doing this) , and try to understand the nature of the missing values. Some key questions are:

      – Are there a lot of missing values for certain variables? (E.g. Sensitive question, data entry errors?)

      – Are there groups of subjects with very little information available? (E.g. Do they have something in common?)

– Which is the pattern of missingness? Monotone or arbitrary?

b) The m complete data sets are analyzed by using standard procedures

c) The parameter estimates from each imputed data set are combined to get a final set of parameter estimates.

Advantages:

It has the same optimal properties as ML, and it removes some of its limitations. Multiple imputation can be used with any kind of data and model with conventional software. When the data is MAR, multiple imputation can lead to consistent, asymptotically efficient, and asymptotically normal estimates.

Limitations:

It is a bit challenging to successfully use it. It produces different estimates (hopefully, only slightly different) every time you use it, which can lead to situations where different researchers get different numbers from the same data using the same method (Nakai & Weiming, 2011)[24], (Allison, 2001)[16].

1.5.7 Maximum Likelihood

We can use this method to get the variance-covariance matrix for the variables in the model based on all the available data points, and then use the obtained variance- covariance matrix to estimate our regression model (Schafer, 1997)[1].Compared to MI, MI requires many more decisions than ML (whether to use Markov Chain Monte Carlo (MCMC) method or the Fully Conditional Specification (FCS)[25], how many data sets to produce, how many iterations between data sets, what prior distribution to use-the default is Jeffreys-, etc.). On the other hand, ML is simpler as you only need to specify your model of interest and indicate that you want to use ML (SAS Institute, 2005)[26].

There are two main ML methods:

a) **Direct Maximum Likelihood:** It implies the direct maximization of the multivariate normal likelihood function for the assumed linear model.

Advantage: It gives efficient estimates with correct standard errors.

Limitations: It requires specialized software (it may be challenging and time consuming).

b) **Expectation-maximization algorithm:** It provides estimates of the means and covariance matrix, which can be used to get consistent estimates of the parameters of interest. It is based on an expectation step and a maximization step, which are repeated several times until maximum likelihood estimates are obtained. It requires a large sample size and that the data are missing at random (MAR).

Advantage: We can use SAS[23], since this is the default algorithm it employs for dealing with missing data with Maximum Likelihood.

Limitations: Only can be used for linear and log-linear models (there is neither theory nor software developed beyond them). (Allison, 2001)[16] (Graham, 2009)[21] (Enders & Bandalos, 2001)[27] and (Allison, 2003)[28].

1.5.8 Bayesian simulation methods

There are two main methods:

a) **Schafer algorithms:** It uses Bayesian iterative simulation methods to impute data sets assuming MAR. Precisely, it splits the multivariate missing problem into a series of univariate problems based on the assumed distribution of the multivariate missing variables (e.g. multivariate normal for continuous variables, multinomial loglinear for categorical variables). In other words, it uses an iterative algorithm that draws samples from a sequence of univariate regressions.

b) **Van Buuren algorithm:** It is a semi-parametric approach. The parametric part implies that each variable has a separate imputation model with a set of predictors that explain the missingness. The non-parametric part implies the specification of an appropriate form (e.g. linear), which depends on the kind of variables (Briggs et al., 2003)[22] (Kong et al., 1994)[29].

1.5.9 Hot deck imputation methods

It is used by the US Census Bureau[30]. This method completes a missing observation by selecting at random, with replacement, a value from those individuals who have matching observed values for other variables. In other words, a missing value is imputed based on an observed value that is closer in terms of distance.SAS macro developed by Lawrence Altmayer, of the U.S. Census Bureau[30]. Can be found in (Ahmed Kazi et al; 2009)[31]. (Briggs et al., 2003)[22]

CHAPTER 2

EXPECTATION MAXIMISATION ALGORITHM

2.1 Introduction

The EM algorithm is used to find (local) maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Typically these models involve latent variables in addition to unknown parameters and known data observations. That is, either missing values exist among the data, or the model can be formulated more simply by assuming the existence of further unobserved data points. For example, a mixture model can be described more simply by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component to which each data point belongs.

Finding a maximum likelihood solution typically requires taking the derivatives of the likelihood function with respect to all the unknown values, the parameters and the latent variables, and simultaneously solving the resulting equations. In statistical models with latent variables, this is usually impossible. Instead, the result is typically a set of interlocking equations in which the solution to the parameters requires the values of the latent variables and vice versa, but substituting one set of equations into the other produces an unsolvable equation.

The EM algorithm proceeds from the observation that there is a way to solve these two sets of equations numerically. One can simply pick arbitrary values for one of the two sets of unknowns, use them to estimate the second set, then use these new values to find a better estimate of the first set, and then keep alternating between the two until the resulting values both converge to fixed points. It's not obvious that this will work, but it can be proven in this context. Additionally, it can be proven that the derivative of the likelihood is (arbitrarily close to) zero at that point, which in turn means that the point is either a local maximum or a saddle point.
In general, multiple maxima may occur, with no guarantee that the global maximum

will be found. Some likelihoods also have singularities in them, i.e., nonsensical maxima. For example, one of the solutions that may be found by EM in a mixture model involves setting one of the components to have zero variance and the mean parameter for the same component to be equal to one of the data points.

2.2 history of the EM algorithm

The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster Nan Laird, and Donald Rubin[32]. They pointed out that the method had been "proposed many times in special circumstances" by earlier authors[33]. One of the earliest is the gene-counting method for estimating allele frequencies by Cedric Smith. Another was proposed by H.O. Hartley in 1958[34], and Hartley and Hocking in 1977[35], from which many of the ideas in the Dempster-Laird-Rubin paper originated[36]. Hartley's ideas[37] can be broadened to any grouped discrete distribution. A very detailed treatment of the EM method for exponential families was published by Rolf Sundberg in his thesis and several papers following his collaboration with Per Martin-Löf and Anders Martin-Löfd–Rubin paper in 1977[38] generalized the method and sketched a convergence analysis for a wider class of problems. The Dempster–Laird–Rubin[32] paper established the EM method as an important tool of statistical analysis.

The convergence analysis of the Dempster–Laird–Rubin algorithm[32] was flawed and a correct convergence analysis was published by C. F. Jeff Wu in 1983[39]. Wu's proof established the EM method's convergence outside of the exponential family, as claimed by Dempster–Laird–Rubin[32]

2.3 EM algorithm

We let Y be the random vector corresponding to the observed data y, having p.d.f. postulated as $g(y;\psi)$, where $\psi = (\psi_1,...,\psi_d)^T$ is a vector of unknown parameters with parameter space Ω.

The EM algorithm is a broadly applicable algorithm that provides an iterative procedure for computing MLE's in situations where, but for the absence of some additional data, ML estimation would be straightforward. Hence in this context, the observed data vector y is viewed as being incomplete and is regarded as an observable function of the so-called complete data. The notion of 'incomplete data' includes the conventional sense of missing data, but it also applies to situations where the complete data represent what would be available from some hypothetical experiment. In the latter case, the complete data may contain some variables that are never observable in a data sense. Within this framework, we let x denote the vector containing the

augmented or so-called complete data, and we let z denote the vector containing the additional data, referred to as the unobservable or missing data.

As will become evident from the many examples of the EM algorithm discussed in this book, even when a problem does not at first appear to be an incomplete-data one, computation of the MLE is often greatly facilitated by artificially formulating it to be as such. This is because the EM algorithm exploits the reduced complexity of ML estimation given the complete data. For many statistical problems the complete-data likelihood has a nice form.

We let $g_c(x;\psi)$ denote the p.d.f. of the random vector X corresponding to the complete data vector x. Then the complete data log likelihood function that could be formed for ψ if x were fully observable is given by

$$\log L_c(\psi) = \log g_c(x;\psi) \tag{2.1}$$

Formally, we have two samples spaces $\mathcal{X}$ and $\mathcal{Y}$ and a many to one mapping from$\mathcal{X}$ to $\mathcal{Y}$. Instead of observing the complete-data vector x in $\mathcal{X}$, we observe the incomplete data vector $y = y(x)$ in $\mathcal{Y}$. It follows that

$$g(y;\psi) = \int_{x(y)} g_c(x;\psi)dx \tag{2.2}$$

where $\mathcal{X}(y)$ is the subset of $\mathcal{X}$ determined by the equation $y = y(x)$.

The EM algorithm approaches the problem of solving the incomplete data likelihood equation

$$\partial \log L(\psi)/\partial\psi = 0 \tag{2.3}$$

indirectly by proceeding iteratively in terms of the complete data log likelihood function, $\log L_c(\psi)$. As it is unobservable, it is replaced by its conditional expectation given y, using the current fit for ψ.

More specifically, let $\psi^{(0)}$ be some initial value for ψ. Then on the first iteration, the E-step requires the calculation of

$$Q(\psi;\psi^{(0)}) = E_{\psi^{(0)}}\{\log L_c(\psi)|y\} \tag{2.4}$$

The M-step requires the maximization of $Q(\psi,\psi^{(0)})$ with respect to ψ over the parameter space Ω. That is, we choose $\psi^{(1)}$ such that

$$Q(\psi^{(1)};\psi^{(0)} \geq Q(\psi;\psi^{(0)}) \tag{2.5}$$

for all $\psi \in \Omega$.

The E- and M-steps are then carried out again, but this time with $\psi^{(0)}$ replaced by

the current fit . On the (k + 1)th iteration, the E- and M-steps are defined as follows:

E-Step. Calculate $Q(\psi;\psi^{(k)})$, where

$$Q(\psi;\psi^{(k)}) = E_{\psi^{(k)}}\{\log L_c(\psi)|y\}. \tag{2.6}$$

M-Step. Choose $\psi^{(k+1)}$ to be any value of $\psi \in \Omega$ that maximizes $Q(\psi;\psi^{(k)})$; that is,

$$Q(\psi^{(k+1)};\psi^{(k)} \geq Q(\psi;\psi^{(k)}) \tag{2.7}$$

for all $\psi \in \Omega$

The E- and M-steps are alternated repeatedly until the difference

$$L(\psi^{(k+1)}) - L(\psi^{(k)}) \tag{2.8}$$

changes by an arbitrarily small amount in the case of convergence of the sequence of likelihood values $\{L(\psi^{(k)})\}$. DLR show that the (incomplete-data) likelihood function $L(\psi)$ is not decreased after an EM iteration; that is,

$$L(\psi^{(k+1)}) \geq L(\psi^{(k)})$$

for k = 0,1,2, ... Hence convergence must be obtained with a sequence of likelihood values that are bounded above.

Another way of expressing 2.7 is to say that $\psi^{(k+1)}$ belongs to

$$\mathcal{M}(\psi^{(k)}) = \arg\max_{\psi} Q(\psi;\psi^{(k)}) \tag{2.9}$$

which is the set of points that maximize $Q(\psi;\psi^{(k)})$.

We see from the above that it is not necessary to specify the exact mapping from $\mathcal{X}$ to $\mathcal{Y}$, nor the corresponding representation of the incomplete-data density g in terms of the complete-data density g_c . All that is necessary is the specification of the complete data vector x and the conditional density of X given the observed data vector y. Specification of this conditional density is needed in order to carry out the E-step. As the choice of the complete data vector x is not unique, it is chosen for computational convenience with respect to carrying out the E- and M-steps. Consideration has been given to the choice of x so as to speed up the convergence of the corresponding EM algorithm.

As pointed out by a referee of the DLR paper, the use of the term "algorithm" to describe this procedure can be criticized, "because it does not specify the sequence of steps actually required to carry out a single E- or M-step." The EM algorithm is really a generic device. Hunter (2003)[40] goes so far as to suggest the usage "EM algorithms" or "an EM algorithm" because many different examples fall under the EM umbrella.

2.4 EM algorithm and Newton-Raphson method

The EM algorithm is an alternative to Newton–Raphson or the method of scoring for computing MLE in cases where the complications in calculating the MLE are due to incomplete observation and data are MAR, missing at random, with separate parameters for observation and the missing data mechanism, so the missing data mechanism can be ignored.

Data (X, Y) are the complete data whereas only incomplete data Y = y are observed. (Rubin uses Y = Y_{obs} and X =Y_{mis}).

The complete data log-likelihood is:

$$L(\psi) = \log L(\psi; x, y) = \log f(x, y; \psi) \tag{2.10}$$

The marginal log-likelihood or incomplete data log-likelihood is based on y alone and is equal to

$$L_y(\psi) = \log L(\psi; y) = \log f(y; \psi) \tag{2.11}$$

We wish to maximize l_y in ψ but l_y is typically quite unpleasant

$$L_y(\psi) = \log \int f(x, y; \psi) dx. \tag{2.12}$$

The EM algorithm is a method of maximizing the latter iteratively and alternates between two steps, one known as the E-step and one as the M-step, to be detailed below. We let ψ^* be and arbitrary but fixed value, typically the value of ψ at the current iteration.

The E-step calculates the expected complete data log-likelihood ratio q($\psi \mid \psi^*$):

$$\begin{aligned} q(\psi|\psi^*) &= E_{\psi^*}[\log \frac{f(X, y; \psi)}{f(X, y; \psi^*)} / Y = y] \\ &= \int \log \frac{f(X, y; \psi)}{f(X, y; \psi^*)} f(x/y; \psi^*) dx \end{aligned} \tag{2.13}$$

The M-step maximizes $q(\psi/\psi^*)$ in ψ for for fixed ψ^*,i.e calculates

$$\theta^{**} = \arg max(\psi; \psi^*)$$

After an E-step and subsequent M-step, the likelihood function has never decreased. The picture on the next overhead should show it all.

2.5 The properties of the EM algorithm

The EM algorithm has enjoyed wide popularity in many scientific fields from the 1970s onwards[41]. This is primarily due to easy implementation and stable convergence. As the previous paragraph illustrates, the EM algorithm is often easy to program and use. Although the algorithm may take many iterations to converge relative to other optimization routines (e.g., Newton–Raphson)[42], each iteration is often easy to program and quick to compute. Moreover, the EM algorithm is less sensitive to poor starting values, and can be easier to use with many parameters since the iterations necessarily remain in the parameter space and no second derivatives are required. Finally, the EM algorithm has the very important property that the objective function is increased at each iteration. That is, by the definition of f($\phi|\theta$),

$$\begin{aligned} \log \int f(\theta,\phi)d\phi &= \log f(\theta,\phi) - \log f(\phi/\theta) \\ &= Q(\theta/\theta^{(k)}) - \int \log f(\phi/\theta) f(\phi/\theta^{(k)} d\phi \end{aligned} \tag{2.14}$$

where the second equality follows by averaging over ϕ according to f($\phi/\theta^{(t)}$). Since the first term of 2.14 is maximized by $\theta^{(t+1)}$, and the second is minimized by $\theta^{(t)}$ (under the assumption that the support of f(ϕ/θ) does not depend on θ), one obtains

$$\log \int f(\theta^{(t+1)},\phi)d\phi \geq \log \int f(\theta^{(t)},\phi)d\phi \tag{2.15}$$

for t=0,1,.... This property not only contributes to the stability of the algorithm, but also is very valuable for diagnosing implementation errors.

Although the EM algorithm is not guaranteed to converge to even a local mode , this can easily be avoided in practice by using several "overdispersed" starting values. Running the EM algorithm with several starting values is also recommended because it can help one to find multiple local modes of $\int f(\theta,\phi)d\phi$, an important advantage for statistical analysis.

2.6 E- and M-Steps for the Regular Exponential Family

The complete-data p.d.f. $g_c(x;\psi)$ is from an exponential family if

$$g_c(x;\psi) = \exp\{a^T(\psi)t(x) - b(\psi) + c(x)\} \tag{2.16}$$

where the sufficient statistic t(x) is a k x 1 (k $\geq$ d) vector and a(ψ) is a k x 1 vector function of the d x 1 parameter vector ψ, and b(ψ) and c(x) are scalar functions. The parameter space Ω is a d-dimensional convex set such that 2.16 defines a p.d.f. for all

ψ in Ω that is,

$$\Omega = \{\psi : \int_x \exp\{a^T(\psi)t(x) + c(x)\}dx < \infty\} \tag{2.17}$$

If k = d and the Jacobian of $a(\psi)$ is of full rank, then $g_c(x;\psi)$ is said to be from a regular exponential family. The coefficient a(ψ) of the sufficient statistic t(x) in 2.16 is referred to as the natural or canonical parameter (vector). Thus if the complete-data p.d.f. $g_c(x;\psi)$ is from a regular exponential family in canonical form, then

$$g_c(x;\psi) = \exp\{\psi^T t(x) - b(\psi) + c(x)\} \tag{2.18}$$

The parameter ψ in 2.18 is unique up to an arbitrary nonsingular d x d linear transformation, as is the corresponding choice of t(x). The expectation of the sufficient statistic t(X) in 2.18 is given by

$$E_\psi\{t(X)\} = \frac{\partial b(\psi)}{\partial \psi} \tag{2.19}$$

Another property of the regular exponential family, which we shall use in a later section, is that the expected information matrix for the natural parameter vector equals the covariance matrix of the sufficient statistic t(X). Thus we have for the regular exponential family in the canonical form 2.18 that

$$cov_\psi t(X) = I_c(\psi) \tag{2.20}$$

where since the second derivatives of 2.20 do not depend on the data,

$$\begin{aligned} I_c(\psi) &= -\partial^2 \log L_c(\psi)/\partial\psi\partial\psi^T \\ &= \partial^2 b(\psi)/\partial\psi\partial\psi^T \end{aligned} \tag{2.21}$$

On taking the conditional expectation of $\log L_c(\psi)$ given y. we have from 2.18 that $Q(\psi;\psi^k)$ is given by, ignoring terms not involving ψ,

$$Q = (\psi;\psi^{(k)}) = \psi^T t^{(k)} - b(\psi) \tag{2.22}$$

where

$$t^{(k)} = E_{\psi(k)}\{t(X)/y\} \tag{2.23}$$

and where $\psi^{(k)}$ denotes the current tit for ψ.

On differentiation of 2.22 with respect to ψ and noting 2.19, it follows that the M-step requires $\psi^{(k+1)}$ to be chosen by solving the equation

$$E_\psi\{t(X)\} = t^{(k)} \tag{2.24}$$

If equation 2.24 can be solved for $\psi^{(k+1)}$ in Ω, then the solution is unique due to the well-known convexity property of minus the log likelihood of the regular exponential family. In cases where the equation is not solvable, the maximizer $\psi^{(k+1)n}$ of $L(\psi)$ lies on the boundary of Ω.

2.7 Censored Exponentially Distributed Survival Times

We suppose W is a non negative random variable having an exponential distribution with mean p Thus its probability density function is given by

$$f(\omega;\mu) = \mu^{-1}\exp(-\omega/\mu)I_{(O,\infty)}(\omega), (\mu > 0), \tag{2.25}$$

where the indicator function $I_{(o,\infty)}(w) = 1$ for $\mu > 0$ and is zero elsewhere. The distribution function is given by

$$F(\omega;\mu) = 1 - \exp(-\omega/\mu)I_{(O,\infty)}(\omega) \tag{2.26}$$

In survival or reliability analyses, a study to observe a random sample $W_1,...,W_n$, from 2.25 will generally be terminated in practice before all of these random variables are able to be observed. We let

$$y = (y_1^T,.....y_n^T)^T$$

denote the observed data, where

$$y_j = (c_j, \delta_j)^T$$

and $\delta = 0$ or 1 according as the observation W_j is censored or uncensored at $c_j (j = 1....,n)$. That is, if the observation W_j is uncensored, its realized value w_j is equal to c_j whereas if it is censored at c_j then w_j is some value greater than $c_j (j = 1,....n)$.

In this example, the unknown parameter vector ψ is a scalar, being equal to μ. We suppose now that the observations have been relabeled so that $W_1,...,W_r$ denote the r uncensored observations and $W_{r+1},...,W_n$the n- r censored observations. The log likelihood function for μ formed on the basis of y is given by

$$\log L(\mu) = -r\log\mu - \sum_{j=1}^{n} c_j/\mu, \tag{2.27}$$

In this case, the MLE of μ can be derived explicitly from equating the derivative of (2.27) to zero to give

$$\mu = \sum_{j=1}^{n} c_j, \tag{2.28}$$

Thus there is no need for the iterative computation of μ. But in this simple case, it

is instructive to demonstrate how the EM algorithm would work.

The complete-data vector x can be declared to be

$$x = (W_1, ..., W_n)^T = (W_1, ..., W_r, z^T)^T$$

where

$z = (W_{r+1}, ..., W_n)^T$

contains the unobservable realizations of the n - r censored random variables. In this example, the so-called unobservable or missing vector z is potentially observable in a data sense, as if the experiment were continued until each item failed, then there would be no censored observations.

The complete-data log likelihood is given by

$$\begin{aligned} \log L_c(\mu) &= \sum_{j=1}^{n} \log g_c(w_j; \mu) \\ &= -n \log \mu - \mu^{-1} \sum_{j=1}^{n} w_j \end{aligned} \tag{2.29}$$

It can be seen that $L_c(\mu)$ belongs to the regular exponential family. We shall proceed now without making explicit use of this property, but in the next section, we shall show how it can be exploited to simplify the implementation of the EM algorithm.

As $L_C(\mu)$ can be seen to be linear in the unobservable data $W_{r+1}, ..., W_n$ the calculation of $Q(\mu; \mu^{(k)}$ on the E-step (on the (k + 1) th iteration) simply requires each such W_j to be replaced by its conditional expectation given the observed data y, using the current fit $\mu^{(k)}$ for μ. By the lack of memory of the exponential distribution, the conditional distribution of $W_j - c_j$ given that $W_j > c_j$ is still exponential with mean μ. Equivalently, the conditional p.d.f of W_j given that it is greater than c_j is

$$\mu^{-1} \exp -(w_j - c_j)/\mu I_{(c_j, \infty)}(w_j), (\mu > 0) \tag{2.30}$$

From 2.30 we have that

$$\begin{aligned} E_{u^{(k}}(W_j/y) &= E_u(W_j/W_j > c_j) \\ &= c_j + E_{u^{(k)}}(W_j) \\ &= c_j + \mu^{(k)} \end{aligned} \tag{2.31}$$

for j=r+l, ... ,n.

we using 2.31 to take the current conditional expectation of the complete-data log

likelihood $\log L_c(\mu)$, we have that

$$\begin{aligned} Q(\mu;\mu^k) &= -n\log\mu - \mu^{-1}\sum_{j=1}^{n} c_j + \sum_{j=r+1}^{n}(c_j + (n-r)\mu^{(k)} \\ &= -\log(\mu) - \mu^{-1}\sum_{j=1}^{n} c_j + (n-r)\mu^{(k)} \end{aligned} \tag{2.32}$$

Concerning the M-step on the (k + 1)th iteration, it follows from 2.32 that the value of μ that maximizes $Q(\mu;\mu^{(k)})$ is given by the MLE of p that would be formed from the complete data, but with each unobservable w_j replaced by its current conditional expectation given by 2.31. Accordingly,

$$\begin{aligned} \mu^{(k+1)} &= \{\sum_{j=1}^{n} c_j + \sum_{j=r+1}^{n} E_{\mu^{(k)}}(W_j/y)\}/n \\ &= \{\sum_{j=1}^{n} c_j + \sum_{j=r+1}^{n}(c_j + \mu^{(k)})\}/n \\ &= \{\sum_{j=1}^{n} c_j + (n-r)\mu^{(k)}\}/n \end{aligned} \tag{2.33}$$

On putting $\mu^{(k+1)} = \mu^{(k)} = \mu^*$ in 2.33 and solving for μ^*, we have for $r < n$ that $\mu^* = \hat{\mu}$. That is, the EM sequence $\{\mu^{(k)}\}$ has the MLE $\hat{\mu}$ as its unique limit point, as k $\to \infty$ In order to demonstrate the rate of convergence of this sequence to $\hat{\mu}$, we can from 2.33 express $\mu^{(k+1)}$ in terms of the MLE $\hat{\mu}$ as

$$\begin{aligned} \mu^{(k+1)} &= \{r\hat{\mu} + (n-r)\mu^{(k)}\}/n \\ &= \hat{\mu} + n^{-1}(n-r)(\mu^{(k)} - \hat{\mu}) \end{aligned}$$

which gives

$$\mu^{(k+1)} - \hat{\mu} = (1 - r/n)(\mu^{(k)} - \hat{\mu}) \tag{2.34}$$

This establishes that $\mu^{(k)}$ converges to $\hat{\mu}$ as k $\to \infty$ co, provided r < n. It can be seen for this problem that each EM iteration is linear. We shall see later that in general the rate of convergence of the EM algorithm is essentially linear. The rate of convergence here is (1 - r/n), which is the proportion of censored observations in the observed sample. This proportion can be viewed as the missing information in the sample

It can be seen in this example that the complete-data distribution has the exponen-

tial family form 2.18 with natural parameter p-1 and sufficient statistic

$$t(X) = \sum_{j=1}^{n} W_j.$$

Hence the E-step requires the calculation of

$$\begin{aligned} t^{(k)} &= E_{\psi^{(k)}}\{t(x)/y\} \\ &= \sum_{j=1}^{n} c_j + \sum_{j=r+1}^{n} (c_j + \mu^{(k)}) \\ &= \sum_{j=1}^{n} c_j + (n-r)\mu^{(k)} \end{aligned}$$

from 2.31 The M-step then yields $\mu^{(k+l)}$ as the value of p that satisfies the equation

$$\begin{aligned} t^(k) &= E_{\mu}\{t(X)\} \\ &= n\mu \end{aligned}$$

This latter equation can be seen to be equivalent to 2.33, as derived by direct differentiation of the Q-function $Q(\psi;\psi^{(k)})$.

2.8 Generalized EM Algorithm

Often in practice, the solution to the M-step exists in closed form. In those instances where it does not, it may not be feasible to attempt to find the value of ψ that globally maximizes the function $Q(\psi;\psi^{(k)})$. For such situations, DLR defined a generalized EM algorithm (GEM algorithm)[39] for which the M-step requires $\psi^{(k+1)}$ to be chosen such that

$$Q(\psi^{(k+l)};\psi^{(k)}) \geq Q(\psi^{(k)};\psi^{(k)}) \tag{2.35}$$

holds. That is, one chooses $\psi^{(k+1)}$ to increase the Q-function $Q(\psi;\psi^{(k)})$ over its value at $\psi = \psi^{(k)}$ rather than to maximize it over all $\psi \in \Omega$ a. As to be shown in Section 3.3[43], the above condition on $\psi^{(k+l)}$ is sufficient to ensure that

$$L(\psi^{(k+1)}) \geq L(\psi^{(k)})$$

Hence the likelihood L(ψ) is not decreased after a GEM iteration, and so a GEM sequence of likelihood values must converge if bounded above. In Section 3.3[43], we shall discuss what specifications are needed on the process of increasing the Q-function in order to ensure that the limit of $\{L(\psi^{(k)})\}$ is a stationary value and that the

sequence of GEM iterates $\{\psi^{(k)}\}$ converges to a stationary point.

2.9 GEM Algorithm Based on One Newton-Raphson Step

In those situations where the global maximizer of the Q-function $Q(\psi;\psi^{(k)})$ does not exist in closed form, consideration may be given to using the Newton-Raphson procedure to iteratively compute $\psi^{(k+1)}$ on the M-step. As remarked above, it is not essential that $\psi^{(k+1)}$ actually maximizes the Q-function for the likelihood to be increased. We can use a GEM algorithm where $\psi^{(k+1)}$ need satisfy only 2.35, which is a sufficient condition to guarantee the monotonicity of the sequence of likelihood values $\{L(\psi^{(k)}\}$. In some instances, the limiting value $\psi^{(k+1)}$ of the Newton-Raphson method[44] may not be a global maximizer. But if condition 2.35 is confirmed to hold on each M-step, then at least the user knows that $\{\psi^{(k)}\}$ is a GEM sequence.

Following Wu (1983)[39] and Jargensen (1984)[45], Rai and Matthews (1993)[46] propose taking $\psi^{(k+1)}$ to be of the form

$$\psi^{(k+1)} = \psi^{(k)} + a^{(k)}\partial^{(k)} \tag{2.36}$$

where

$$\partial^{(k)} = -[\partial^2 Q(\psi;\psi^{(k)})/\partial\psi\partial\psi^T]^{-1}_{\psi=\psi^{(k)}}[\partial Q(\psi;\psi^{(k)})/\partial\psi]_{\psi=\psi^{(k)}} \tag{2.37}$$

and where $0 < a^{(k)} < 1$.

It can be seen that in the case of $a^{(k)} = 1$, 2.36 is the first iterate obtained when using the Newton-Raphson procedure to obtain a root of the equation

$$\partial Q(\psi;\psi^{(k)})/\partial\psi = 0$$

The idea is to choose $a^{(k)}$ so that 2.37 defines a GEM sequence, that is, so that 2.35 holds. It can be shown that

$$Q(\psi^{(k+1)};\psi^{(k)}) - Q(\psi^{(k)};\psi^{(k)}) = a^{(k)}S(y;\psi^{(k)})^T A^{(k)} S(y;\psi^{(k)}) \tag{2.38}$$

where

$$A^{(k)} = I_c^{-1}(\psi^{(k)};y)\{I_d - \frac{1}{2}a^{(k)}\hat{I}_c^{(k)}(y)I_c^{-1}(\psi^{(k)};y)\} \tag{2.39}$$

and where

$$\begin{aligned}\hat{I}_c^{(k)}(y) &= -[\partial^2 Q(\psi;\psi^{(k)})/\partial\psi\partial\psi^T]_{\psi=\hat{\psi}^{(k)}} \\ &= E(\psi^{(k)})\Big(I_c(\hat{\psi}^{(k)};X)/y\Big)\end{aligned}$$

and $\hat{\psi}^{(k)}$ is a point on the line segment from $\psi^{(k)}$ to $\psi^{(k+1)}$; I_d denotes the dxd iden-

tity matrix. Thus the left-hand side of 2.38 is nonnegative if the matrix $A^{(k)}$ is positive definite.

Typically in practice, $I_c(\psi^{(k)};y)$ is positive definite and so then we have a GEM sequence if the matrix

$$I_d - \frac{1}{2}a^{(k)}\hat{I}_c^{-1}(y)I_c^{-1}(\psi^{(k)};y) \tag{2.40}$$

is positive definite, which can be achieved by choosing the constant $a^{(k)}$ sufficiently small. Suppose that the sequence $\{\psi^{(k)}\}$ tends to some limit point as k$\rightarrow \infty$. Then it can be seen from 2.39 that, as $k \rightarrow \infty$, $a^{(k)} < 2$ will ensure that 2.40 holds.

The derivation of 2.38 is to be given in Section 4.12, where the use of this GEM algorithm in an attempt to reduce the computation on the M-step, is to be considered further.

2.10 EM Gradient Algorithm

The algorithm that uses one Newton-Raphson[47] step to approximate the M-step of the EM algorithm (that is, uses 2.36 with $a^{(k)} = 1$ is referred to by Lange (1995a)[48] as the EM gradient algorithm. It forms the basis of the quasi-Newton approach of Lange (1995b)[49] to speed up the convergence of the EM algorithm, as to be considered in Section 4.14[43]. But as pointed out by Lange (1995b)[49], it is an interesting algorithm in its own right, and is to be considered further in Section 4.13[43].

2.11 EM Mapping

Any instance of the EM (GEM) algorithm as described above implicitly defines a mapping $\psi \rightarrow M(\psi)$, from the parameter space off ψ, Ω, to itself such that

$$\psi^{(k+1)} = M(\psi^{(k)})(k = 0, 1, 2, 3, ...) \tag{2.41}$$

If $\psi^{(k)}$ converges to some point ψ^* and M(ψ) is continuous, then ψ^* must satisfy

$$\psi^* = M(\psi^*)$$

Thus ψ^* is a fixed point of the map M.

It is easy to show that if the MLE $\hat{\psi}$ of ψ is the unique global maximizer of the likelihood function, then it is a fixed point of the EM algorithm (although there is no guarantee that it is the only one). To see this, we note that the M-step of the EM algorithm (or a GEM algorithm) implies that

$$L(M(\hat{\psi}) \geq L(\hat{\psi}) \tag{2.42}$$

Thus M(ψ^*)= $\hat{\psi}$,as otherwise 2.42 would contradict the assertion that

$$L(\hat{\psi}) > L(\psi) \tag{2.43}$$

for all ψ (not equal to $\hat{\psi}$) $\in \Omega$.

2.12 EM algorithm for bivariate normal data with missing values

The purpose of this problem is to use the EM algorithm to estimate the mean of a bivariate normal dataset with missing entries in one of the two variables. We first generate synthetic data and then implement the EM algorithm to compute the estimator of the mean.

```
library(mvtnorm)
```

We consider a bivariate normal random variable $Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}$ and denote the mean vector and covariance matrix of its distribution $\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix}$: $Y \sim \mathcal{N}(\mu, \Sigma)$. We observe a sample of size n that contains some missing values in the variable Y_2, such that for some $r \leq n$, we observe (y_{i_1}, y_{i_2}) for $i = 1,...,r$ and y_{i1} for $i = r+1,...n$. The goal is to estimate the mean μ. We will compare two strategies:

1) direct computation of the maximum likelihood estimator and

2) estimation of the mean with the EM algorithm.

2.12.1 Data generation

Request 1:

Generate a bivariate normal sample of size 100 of mean $\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 5 \\ -1 \end{pmatrix}$ and covariance matrix $\begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} = \begin{pmatrix} 1.3 & 0.4 \\ 0.4 & 0.9 \end{pmatrix}$ containing 30% of missing values in the variable y_2.

```
set.seed(100)
```

```
n <- 100
r <- floor(n*0.3)
mu <- c(5, -1)
Sigma <- matrix(c(1.3, 0.4,0.4,0.9), nrow=2)
Y <- rmvnorm(n, mean=mu, sigma=Sigma)
```

```
missing_idx <-sample(100, r, replace = FALSE)
Y[missing_idx, 2] <- NA
```

2.12.2 Maximum likelihood estimator

We denote by $f_{1,2}(y_1, y_2; \mu, \Sigma)$, $f_1(y_1; \mu_1, \sigma_{11})$ and $f_{2|1}(y_2|y_1; \mu, \Sigma)$ the probability density functions of the joint (y_1, y_2), y_1 and $y_2|y_1$ respectively. The likelihood of the observed data can be written as

$$f_{1,2}(y_1, y_2; \mu, \Sigma) = \prod_{i=1}^{n} f_1(y_{i1}) \prod_{j=1}^{r} f_{2|1}(y_{j2}|y_{j1}),$$

and the log-ikelihood is written (up to an additional constant that does not appear in the maximization and that we therefore drop)

$$l(\mu, \Sigma|y_1, y_2) = -\frac{n}{2}\log(\sigma_{11}^2) - \frac{1}{2}\sum_{i=1}^{n}\frac{(y_{i1}-\mu_1)^2}{\sigma_{11}^2} - \frac{r}{2}\log((\sigma_{22} - \frac{\sigma_{12}^2}{\sigma_{11}})^2)$$
$$-\frac{1}{2}\sum_{i=1}^{r}\frac{(y_{i2}-\mu_2-\frac{\sigma_{12}}{\sigma_{11}}(y_{i1}-\mu_1))^2}{(\sigma_{22}-\frac{\sigma_{12}^2}{\sigma_{11}})^2}$$

We skip the computations and directly give the expression of the closed form maximum likelihood estimator of the mean:

$$\hat{\mu}_1 = n^{-1}\sum_{i=1}^{n} y_{i1}$$

$$\hat{\mu}_2 = \hat{\beta}_{20.1} + \hat{\beta}_{21.1}\hat{\mu}_1,$$

$$\hat{\beta}_{21.1} = s_{12}/s_{11}, \hat{\beta}_{20.1} = \bar{y}_2 - \hat{\beta}_{21.1}\bar{y}_1,$$

and

$$\bar{y}_j = r^{-1}\sum_{i=1}^{r} y_{ij}, j = 1, 2, \quad s_{jk} = r^{-1}\sum_{i=1}^{r}(y_{ij}-\bar{y}_j)(y_{ik}-\bar{y}_k), j, k = 1, 22$$

Request 2:

Compute the maximum likelihood estimates of μ_1 and μ_2.

```
  hat_mu1_ML <- (1/n)*sum(Y[,1])
bar_y1 <- mean(Y[setdiff(1:n,missing_idx), 1])
# mean(Y[!((1:n)%in%missing_idx), 1])
bar_y2 <-mean(Y[setdiff(1:n,missing_idx), 2])
s_11 <- mean((Y[setdiff(1:n,missing_idx),1]-bar_y1)^2)
s_22 <- mean((Y[setdiff(1:n,missing_idx),2]-bar_y2)^2)
```

```
s_12 <- mean((Y[setdiff(1:n,missing_idx),1]-bar_y1)*
             (Y[setdiff(1:n,missing_idx),2]-bar_y2))
hat_beta_21.1 <- s_12/s_11
hat_beta_20.1 <- bar_y2-hat_beta_21.1*bar_y1
hat_mu2_ML <- hat_beta_20.1+hat_beta_21.1*hat_mu1_ML
resML <- c(hat_mu1_ML=hat_mu1_ML,hat_mu2_ML=hat_mu2_ML
```

2.12.3 EM algorithm

In this simple setting, we have an explicit expression of the maximum likelihood estimator despite missing values. However, this is not always the case but it is possible to use an EM algorithm which allows to get the maximum likelihood estimators in the cases where data are missing.

The EM algorithm consists in maximizing the "observed likelihood"

$$l(\mu,\Sigma|y_1,y_2) = -\frac{n}{2}\log(\sigma_{11}^2) - \frac{1}{2}\sum_{i=1}^{n}\frac{(y_{i1}-\mu_1)^2}{\sigma_{11}^2} - \frac{r}{2}\log((\sigma_{22}-\frac{\sigma_{12}^2}{\sigma_{11}})^2)$$

$$-\frac{1}{2}\sum_{i=1}^{r}\frac{(y_{i2}-\mu_2-\frac{\sigma_{12}}{\sigma_{11}}(y_{i1}-\mu_1))^2}{(\sigma_{22}-\frac{\sigma_{12}^2}{\sigma_{11}})^2},$$

through successive maximization of the "complete likelihood" (if we had observed all n realizations of y_1 and y_2). Maximizing the complete likelihood

$$l_c(\mu,\Sigma|y_1,y_2) = -\frac{n}{2}\log(\det(\Sigma)) - \frac{1}{2}\sum_{i=1}^{n}(y_{i1}-\mu_1)^T\Sigma^{-1}(y_{i1}-\mu_1)$$

would be straightforward if we had all the observations. However elements of this likelihood are not available. Consequently, we replace them by the conditional expectation given observed data and the parameters of the current iteration. These two steps of computation of the conditional expectation (E-step) and maximization of the completed likelihood (M step) are repeated until convergence.

The update formulas for the E and M steps are the following

E step:

The sufficient statistics of the likelihood are:

$$s_1 = \sum_{i=1}^{n} y_{i1}, \quad s_2 = \sum_{i=1}^{n} y_{i2}, \quad s_{11} = \sum_{i=1}^{n} y_{i1}^2, \quad s_{22} = \sum_{i=1}^{n} y_{i2}^2, \quad s_{12} = \sum_{i=1}^{n} y_{i1}y_{i2}.$$

Since some values of y_2 are not available, we fill in the sufficient statistics with:

$$E[y_{i2}|y_{i1},\mu,\Sigma] = \beta_{20.1} + \beta_{21.1}y_{i1}$$

$$E[y_{i2}^2|y_{i1},\mu,\Sigma] = (\beta_{20.1} + \beta_{21.1}y_{i1})^2 + \sigma_{22.1}$$

$$E[y_{i2}y_{i2}|y_{i1},\mu,\Sigma] = (\beta_{20.1} + \beta_{21.1}y_{i1})y_{i1}.$$

M step:

The M step consists in computing the maximum likelihood estimates as usual. Given $s_1, s_2, s_{11}, s_{22},$ and s_{12}, update $\hat{\mu}$ and $\hat{\sigma}$ with

$$\hat{\mu}_1 = s_1/n,\ \hat{\mu}_2 = s_2/n,$$

$$\hat{\sigma}_1 = s_{11}/n - \hat{\mu}_1^2,\ \hat{\sigma}_2 = s_{22}/n - \hat{\mu}_2^2,\ \hat{\sigma}_{12} = s_{12}/n - \hat{\mu}_1\hat{\mu}_2$$

Note that s_1, s_{11}, $\hat{\mu}_1$ and $\hat{\sigma}_1$ are constant accross iterations since we do not have missing values on y_1.

Request 3:

Write two functions called Estep and Mstep that respectively perform the E step and the M step. The Estep function can take as an input μ and Σ. Then, you can compute $\beta_{21.1} = \sigma_{12}/\sigma_{11}$, $\beta_{20.1} = \mu_2 - \beta_{21.1}\mu_1$, and $\sigma_{22.1} = \sigma_{22} - \sigma_{12}^2/\sigma_{11}$ and update the sufficient statistics s_{ij}.

The Mstep function consists in updating the update the μ and Σ given the s_{ij}.

```
Estep=function(Y, mu, Sigma, missing_idx)
{
n=nrow(Y)
sigma_22.1=Sigma[2,2]-Sigma[1,2]^2/Sigma[1,1]
beta_21.1=Sigma[1,2]/Sigma[1,1]
beta_20.1=mu[2]-beta_21.1*mu[1]
E_y2=rep(0, n)
E_y2[missing_idx]=rep(beta_20.1, length(missing_idx))+beta_21.1*Y[missing_idx,1]
E_y2[setdiff(1:n, missing_idx)]=Y[setdiff(1:n, missing_idx),2]
E_y1=Y[,1]
E_y2_y2=rep(0, n)
E_y2_y2[missing_idx]=E_y2[missing_idx]^2+rep(sigma_22.1, length(missing_idx))
E_y2_y2[setdiff(1:n, missing_idx)]=E_y2[setdiff(1:n, missing_idx)]^2
E_y1_y1=Y[,1]^2
E_y1_y2=rep(0, n)
E_y1_y2=E_y2*E_y1
return(structure(list(s1=sum(E_y1), s2=sum(E_y2), s11=sum(E_y1_y1),
                          s22=sum(E_y2_y2), s12=sum(E_y1_y2))))
```

```
}

Mstep=function(Y, s1, s2, s11, s22, s12)
{
n=nrow(Y)
mu1=s1/n
mu2=s2/n
sigma1=s11/n-mu1^2
sigma2=s22/n-mu2^2
sigma12=s12/n-mu1*mu2
mu=c(mu1,mu2)
Sigma=matrix(c(sigma1, sigma12,sigma12,sigma2), nrow=2)
return(structure(list(mu=mu, Sigma=Sigma)))
}
```

Question:

How could we initialize the algorithm ?

request 4:

Implement a function called initEM that returns initial values for $\hat{\mu}$ and $\hat{\Sigma}$.

```
  initEM=function(Y, missing_idx)
{n=nrow(Y)
r=n-length(missing_idx)
mu1=mean(Y[,1])
mu2=mean(Y[,2], na.rm=T)
sigma1=mean(Y[,1]^2)-mu1^2
sigma2=mean(Y[,2]^2, na.rm=T)-mu2^2
sigma12=mean(Y[,1]*Y[,2], na.rm=T)-mu1*mu2
mu=c(mu1,mu2)
Sigma=matrix(c(sigma1, sigma12,sigma12,sigma2), nrow=2)
return(structure(list(mu=mu, Sigma=Sigma)))
}
```

request 5:

Implement the EM algorithm over 15 iterations and plot the value of $\left\|\mu - \hat{\mu}\right\|^2$ over iterations. Comment your results briefly.

```
 init=initEM(Y, missing_idx)
hat_mu=init$mu
hat_Sigma=init$Sigma
```

```
error_mu=rep(0,50)
for(i in 1:50)
{
error_mu[i]=sqrt(sum((hat_mu-mu)^2))
# E step
E=Estep(Y, hat_mu, hat_Sigma, missing_idx)
s1=E$s1
s11=E$s11
s2=E$s2
s22=E$s22
s12=E$s12
M=Mstep(Y, s1, s2, s11, s22, s12)
hat_mu=M$mu
print(hat_mu)
hat_Sigma=M$Sigma
}
plot(error_mu)
```

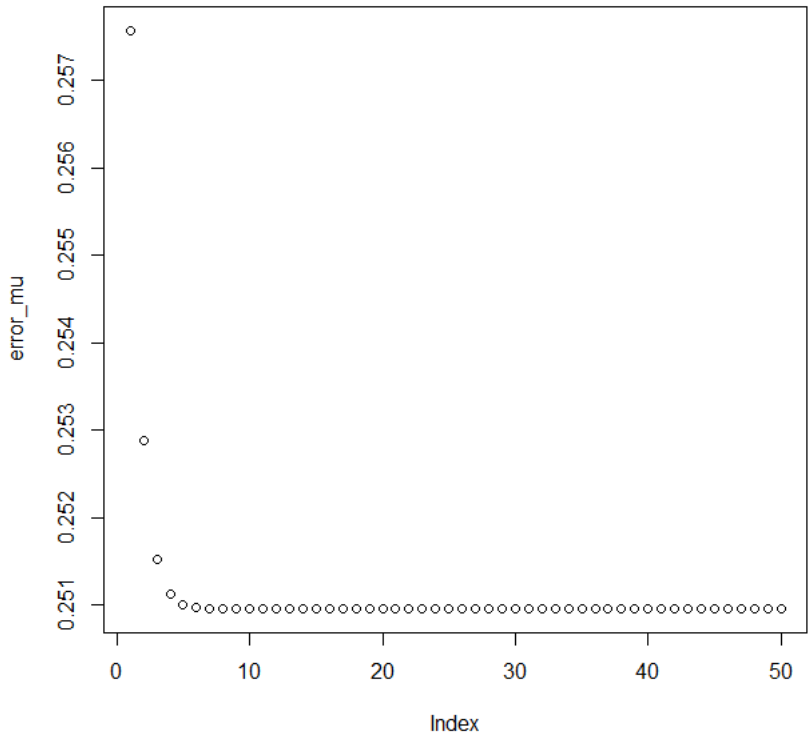

Figure 2.1: error mu

request 6: Check that the EM estimator μ is equal to the maximum likelihood

estimator.

finally we get

$$\hat{\mu} = \begin{pmatrix} \hat{\mu_1} \\ \hat{\mu_2} \end{pmatrix} = \begin{pmatrix} 4.8505341 \\ -0.7984072 \end{pmatrix}$$

CHAPTER 3

SIMULATIONS AND APPLICATIONS

3.1 Two recommended methods: EM / Multiple imputation

Under the classical missing at random mechanism (MAR) assumption, the parameters can thus be estimated by maximizing the observed likelihood. To do so, it is possible to use an Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977)[32] as detailled in the next paragraph - The standard error of the parameters can be estimated using a supplemented Expectation-Maximization (SEM) algorithm (Meng and Rubin, 1991)[50]. This is the first main startegy to do inference with missing values. In fact, it consists in adapting the statistical analysis (the estimation process) so that it can be applied on an incomplete data set. It is tailored to a specific statistical method but it has two drawbacks:

1. it can be difficult to establish(EM algorithm can involve integral not easy to compute)

2. a specific algorithm has to be derived for each statistical method that we would like to apply.

This is why the second strategy, namely multiple imputation (MI) (Rubin, 1987, Little and Rubin, 1987, 2002)[51] seems to have taken the lead. The principle of MI consists in predicting M different values for each missing value, which leads to M imputed data sets. The variability across the imputations reflects the variance of the prediction of each missing entry. Then, MI consists in performing the statistical analysis on each imputed data set to estimate the parameter θ and consists of combining the results $(\theta_m)_{1\leq m\leq M}$ to provide a unique estimation for θ and for its associated variability using Rubin's rules (Rubin, 1987)[52]. This ensures that the variance of the estimator is not

underestimated and thus good coverage properties. What is important is that the aim of both approaches is to estimate as well as possible the parameters and their variance despite missing values, i.e. taking into account the supplementary variability due to missing values. The goal is not to impute the entries as accurately as possible.

3.2 Expectation Maximization algorithm

In the case where we are interested in estimating some unknown parameter $\theta \in \mathbb{R}^d$ characterizing the model (such as μ and Σ in the Gaussian example), the Expectation Maximization (EM) algorithm (Dempster et al. 1977) [32] can be used when the joint distribution of the missing data X_{MIS} and the observed data X_{OBS} is explicit. For all $\theta \in \mathbb{R}^d$ let f_θ be the probability density function of $(X_{\text{OBS}}, X_{\text{MIS}})$ with respect to a given reference measure μ. The EM algorithm aims at iteratively maximizing the likelihood of the observations, i.e. the probability density function of the observations, where y refers to X_{OBS} and x to X_{MIS}:

$$L_\theta(y) = \int f_\theta(x, y)\lambda(\mathrm{d}x).$$

As this quantity cannot be computed explicitly in general cases, the EM algorithm relies on the surrogate intermediate quantity:

$$Q(\theta, \theta') = \mathbb{E}_{\theta'}[\log f_\theta(X_{\text{OBS}}, X_{\text{MIS}})|X_{\text{OBS}}],$$

where $\mathbb{E}_{\theta'}$ is the expectation under the law of the model parameterized by θ'. The following crucial property motivates the EM algorithm: for all θ, θ',

$$\log L_\theta(Y) - \log L_{\theta'}(Y) \geq Q(\theta, \theta') - Q(\theta', \theta').$$

Therefore, any value θ such that $Q(\theta, \theta')$ is greater than the reference value $Q(\theta', \theta')$ increases the log likelihood of the observations. Based on this inequality, the EM algorithm produces iteratively a sequence of parameter estimates $(\theta_p)_{p\geq 0}$. Each iteration is decomposed into two steps:

$$\text{E-step: compute} \quad \theta \mapsto Q(\theta, \theta_p), \text{M-step: set} \quad \theta_p \in \arg\max_\theta Q(\theta, \theta_p).$$

The practical interest of this algorithm can be assessed only in cases where $Q(\theta, \theta_p)$ can be computed (or estimated) with a reasonable computational cost (see for instance the special case where f_θ belongs to the exponential family) and when $\theta \mapsto Q(\theta, \theta_p)$ can be maximized (at least numerically).

3.3 Conditional distributions in the Gaussian case

Assume first that the complete data (X, Y) has a multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$. The parameters μ and Σ may be estimated using maximum likelihood based procedures for incomplete data models such as the Expectation Maximization algorithm detailed above. Then, the conditional distribution of the missing data X_{MIS} given the observations X_{OBS} can be derived using Schur complements. If $\Sigma_{\text{MIS}} \in \mathbb{R}^{m\times m}$ is the covariance of X_{MIS}, $\Sigma_{\text{OBS}} \in \mathbb{R}^{p\times p}$ is the covariance of X_{OBS} and $C_{\text{MIS,OBS}} \in \mathbb{R}^{m\times p}$ is the covariance matrix between X_{MIS} and X_{OBS} then Σ is given by:

$$\Sigma = \begin{pmatrix} \Sigma_{\text{MIS}} & C_{\text{MIS,OBS}} \\ C'_{\text{MIS,OBS}} & \Sigma_{\text{OBS}} \end{pmatrix}.$$

Conditionally on X_{OBS}, X_{MIS} has a normal distribution with covariance matrix $\Sigma_{X_{\text{MIS}}|X_{\text{OBS}}}$ given by the Schur complement of Σ_{OBS} in Σ:

$$\Sigma_{\text{MIS|OBS}} = \Sigma_{\text{MIS}} - C_{\text{MIS,OBS}}\Sigma_{\text{OBS}}^{-1}C'_{\text{MIS,OBS}}.$$

Note also that the mean $\mu_{\text{MIS|OBS}}$ of the distribution of X_{MIS} given X_{OBS} is:

$$\mu_{\text{MIS|OBS}} = \mathbb{E}[X_{\text{MIS}}] + C_{\text{MIS,OBS}}\Sigma_{\text{OBS}}^{-1}(X_{\text{OBS}} - \mathbb{E}[X_{\text{OBS}}]).$$

In R, we can estimate the mean and covariance matrix with EM and then impute missing values with the previous formulae with:

```
library(norm)
pre <- prelim.norm(as.matrix(don))
thetahat <- em.norm(pre)
```

so we get the iterations of EM:

1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...21...22...23...24...

```
getparam.norm(pre,thetahat)
```

Table 3.1: Values of μ

	[,1]	[,2]	[,3]	[,4]	[,5]]
[1,]	2.001076e+07	9.074487e+01	1.814555e+01	2.124103e+01	2.247745e+01

	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]
[1,]	4.882849e+00	4.893095e	4.735842e	-1.174023e	-1.625259e	-1.654785e

Table 3.2: Values of σ

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	12024.62747	-701.50386	-84.3314880	-53.173896	-52.334905	30.031060
[2,]	-701.50386	799.97281	57.7080103	87.833518	98.556827	-41.753582
[3,]	-84.33149	57.70801	9.5010995	10.746927	11.626659	-2.449068
[4,]	-53.17390	87.83352	10.7469268	15.920412	16.752532	-4.568836
[5,]	-52.33490	98.55683	11.6266593	16.752532	20.435781	-5.534268
[6,]	30.03106	-41.75358	-2.4490682	-4.568836	-5.534268	6.042519
[7,]	52.92496	-43.73738	-2.6048132	-5.323420	-6.238921	4.102274
[8,]	46.69286	-32.08293	-1.9842688	-3.486077	-5.632869	2.959156
[9,]	-20.57684	38.06663	1.2604022	4.163887	5.160051	-2.895176
[10,]	13.24623	35.80373	1.0361158	2.812798	4.171018	-3.659578
[11,]	42.35594	28.35067	0.1733737	1.934071	2.826757	-2.948763

	[,7]	[,8]	[,9]	[,10]	[,11]
[1,]	52.924960	46.692859	-20.576838	13.246228	42.3559379
[2,]	-43.737382	-32.082933	38.066629	35.803727	28.3506673
[3,]	-2.604813	-1.984269	1.260402	1.036116	0.1733737
[4,]	-5.323420	-3.486077	4.163887	2.812798	1.9340706
[5,]	-6.238921	-5.632869	5.160051	4.171018	2.8267575
[6,]	4.102274	2.959156	-2.895176	-3.659578	-2.9487626
[7,]	5.057694	3.670474	-2.888070	-3.122882	-2.5286569
[8,]	3.670474	5.304180	-2.407587	-2.567052	-2.2318395
[9,]	-2.888070	-2.407587	6.734501	5.553650	4.7461342
[10,]	-3.122882	-2.567052	5.553650	7.964083	6.4291709
[11,]	-2.528657	-2.231839	4.746134	6.429171	7.5462185

```
imp <- imp.norm(pre,thetahat,don)
```

3.3.1 Bootstrap

The bootstrap method is another way to estimate unknown parameters characterizing the statistical model: confidence intervals, estimation of the standard error etc. It is used when the unknown quantity to be estimated can be written as a functional of the unknwon distribution function f of interest. For instance, in the case of incomplete data models, the bootstrap method is a solution to estimate any quantity which can be expressed as a functional of the unknown conditional distribution π of the latent data X_{MIS} given the observations.

Assume that $(X_i)_{1\leq i\leq n}$ are i.i.d. with common unknown distribution function f and let $\theta \in \mathbb{R}^d$ be any parameter characterizing f. The parameter θ is estimated by $\hat{\theta}(X_1,\ldots,X_n)$. Then, the variance of the estimator is given by:

$$\mathbb{V}_f[\hat{\theta}(X_1,\ldots,X_n)] = \mathbb{E}_f\left[\left(\hat{\theta}(X_1,\ldots,X_n) - \mathbb{E}_f[\hat{\theta}(X_1,\ldots,X_n)]\right)^2\right],$$

where $\mathbb{E}_f$ is the expectation under the law of $(X_1,\ldots,X_n)$. The bootstrap estimator of

$\mathbb{V}_f[\hat{\theta}(X_1,\dots,X_n)]$ is obtained then by replacing the unknwon distribution function f in this expression by its empirical estimate given, for any x, by:

$$f_n(x) = \frac{1}{n}\sum_{i=1}^{n} 1_{(-\infty,x]}(X_i).$$

For any integrable function h,

$$\mathbb{E}_{f_n}[h(Z)] = \frac{1}{n}\sum_{i=1}^{n} h(X_i).$$

Replacing f by f_n can lead to highly involved estimates but in some common situations the bootstrap estimate of $\mathbb{V}_f[\hat{\theta}(X_1,\dots,X_n)]$ can be derived easily.

For instance, assume that $\hat{\theta}(X_1,\dots,X_n) = \bar{X}_n$ is the empirical estimate of the mean of f.

Then

$$\mathbb{V}_f[\hat{\theta}(X_1,\dots,X_n)] = \mathbb{V}_f[\bar{X}_n] = \frac{1}{n}\left(\mathbb{E}_f[X_1^2] - \mathbb{E}_f[X_1]^2\right).$$

Therefore, the bootstrap estimator of $\mathbb{V}_f[\hat{\theta}(X_1,\dots,X_n)]$ is given by

$$\mathbb{V}_{f_n}[\hat{\theta}(X_1,\dots,X_n)] = \frac{1}{n}\left(\mathbb{E}_{f_n}[X_1^2] - \mathbb{E}_{f_n}[X_1]^2\right) = \frac{1}{n}\left(\frac{1}{n}\sum_{i=1}^{n} X_i^2 - \bar{X}_n^2\right).$$

3.4 Gibbs sampling

In the case where the complete data $(X_{\text{OBS}}, X_{\text{MIS}})$ is not assumed to be a Gaussian vector, we may be interested in estimating or sampling from the (usually unknwon) conditional distribution π of the missing data X given the observations Y. A widely spread technique to do so is to use Markov Chain Monte Carlo (MCMC) methods[53] which naturally provide simulation based methods which have been successfully applied to many disciplines such as signal processing, biology, target tracking etc... One of the main objectives of these MCMC algorithms is to produce a Markov chain $(\xi^i)_{i\geq 0}$ targetting the unknown target distribution π. Using ergodic theory for Markov chains, it is expected for instance that $N^{-1}\sum_{i=1}^{N} f(\xi^i)$ is a good estimate of

$$\int f(x)\pi(\mathrm{d}x) = \mathbb{E}[f(X_{\text{MIS}})|X_{\text{OBS}}]$$

for a large class of functions f.

Many MCMC algorithms have been developed to sample the chain $(\xi^i)_{i\geq 1}$, this section details the popular Gibbs sampler which may be used when the conditional distribution of each latent variable given all the other variables has a simple form. Assume

that the missing data may be decomposed into several components $(X_1,\dots,X_m)$ and for all $1 \leq k \leq m$ let π_{-k} be the conditional distribution of X_k given the observations and the other missing data. Then, starting with any initial state $\xi^1 = (\xi_1^1,\dots,\xi_m^1)$, for all $i \geq 1$, conditionally on ξ^i, the Gibss sampler samples ξ^{i+1} as follows. For all $1 \leq k \leq m$, ξ_k^{i+1} is sampled accodind to $\pi_{-k}(\cdot|\xi_1^{i+1},\dots,\xi_{k-1}^{i+1},\xi_{k+1}^{i},\xi_m^{i})$. All components of the new state ξ^{i+1} are sampled iteratively according to the conditional distribution given the observations and the other components. A nice feature of this conditional sampler is that at each iteration $i \geq 1$, every component update $1 \leq k \leq m$ (which produces ξ_k^{i+1}) is reversible with respect to π which implies that the Markov kernel associated with each component update admits π as a stationary probability distribution. The Gibbs sampler convergence may be established for the complete update of all components at each iteration and several procedures have been proposed to combine the individual moves (not necessarily with a systematic update of each component in a row).

3.5 R Packages used for imputing missing values

Overview Learn the methods to impute missing values in R for data cleaning and exploration Understand how to use packages like amelia, missForest, hmisc, mi and mice which use bootstrap sampling and predictive modeling

Introduction Missing values are considered to be the first obstacle in predictive modeling. Hence, it's important to master the methods to overcome them. Though, some machine learning algorithms claim to treat them intrinsically, but who knows how good it happens inside the 'black box'.

The choice of method to impute missing values, largely influences the model's predictive ability. In most statistical analysis methods, listwise deletion is the default method used to impute missing values. But, it not as good since it leads to information loss.

Do you know R has robust packages for missing value imputations?

Yes! R Users have something to cheer about. We are endowed with some incredible R packages for missing values imputation. These packages arrive with some inbuilt functions and a simple syntax to impute missing data at once. Some packages are known best working with continuous variables and others for categorical. With this article, you can make a better decision choose the best suited package.

In this article, I've listed 5 R packages popularly known for missing value imputation. There might be more packages. But, I decided to focus on these ones. I've tried to explain the concepts in simplistic manner with practice examples in R.

Loading Image India's Largest Data Science Hiring Event

missing values imputation, powerful R packages Tutorial on 5 Powerful R Packages used for imputing missing values

List of mains R Packages:

- MICE
- Amelia
- missForest
- Hmisc
- mi

3.5.1 MICE package

MICE (Multivariate Imputation via Chained Equations) is one of the commonly used package by R users. Creating multiple imputations as compared to a single imputation (such as mean) takes care of uncertainty in missing values.

MICE assumes that the missing data are Missing at Random (MAR), which means that the probability that a value is missing depends only on observed value and can be predicted using them. It imputes data on a variable by variable basis by specifying an imputation model per variable.

For example:

Suppose we have X1, X2....Xk variables. If X1 has missing values, then it will be regressed on other variables X2 to Xk. The missing values in X1 will be then replaced by predictive values obtained Similarly, if X2 has missing values, then X1, X3 to Xk variables will be used in prediction model as independent variables. Later, missing values will be replaced with predicted values.

By default, linear regression is used to predict continuous missing values. Logistic regression is used for categorical missing values. Once this cycle is complete, multiple data sets are generated. These data sets differ only in imputed missing values. Generally, it's considered to be a good practice to build models on these data sets separately and combining their results.

Precisely, the methods used by this package are:

- PMM (Predictive Mean Matching) – For numeric variables
- logreg(Logistic Regression) – For Binary Variables(with 2 levels)
- polyreg(Bayesian polytomous regression) – For Factor Variables (>= 2 levels)
- Proportional odds model (ordered, >= 2 levels)

3.5.2 Amelia

This package (Amelia II) is named after Amelia Earhart, the first female aviator to fly solo across the Atlantic Ocean. History says, she got mysteriously disappeared (missing) while flying over the pacific ocean in 1937, hence this package was named to solve missing value problems.

This package also performs multiple imputation (generate imputed data sets) to deal with missing values. Multiple imputation helps to reduce bias and increase efficiency. It is enabled with bootstrap based EMB algorithm which makes it faster and robust to impute many variables including cross sectional, time series data etc. Also, it is enabled with parallel imputation feature using multicore CPUs.

It makes the following assumptions:

All variables in a data set have Multivariate Normal Distribution (MVN). It uses means and covariances to summarize data. Missing data is random in nature (Missing at Random) It works this way. First, it takes m bootstrap samples and applies EMB algorithm to each sample. The m estimates of mean and variances will be different. Finally, the first set of estimates are used to impute first set of missing values using regression, then second set of estimates are used for second set and so on.

On comparing with MICE, MVN lags on some crucial aspects such as:

MICE imputes data on variable by variable basis whereas MVN uses a joint modeling approach based on multivariate normal distribution. MICE is capable of handling different types of variables whereas the variables in MVN need to be normally distributed or transformed to approximate normality. Also, MICE can manage imputation of variables defined on a subset of data whereas MVN cannot. Hence, this package works best when data has multivariable normal distribution. If not, transformation is to be done to bring data close to normality.

3.5.3 missForest

As the name suggests, missForest is an implementation of random forest algorithm. It's a non parametric imputation method applicable to various variable types.

So, what's a non parametric method ?

Non-parametric method does not make explicit assumptions about functional form of f (any arbitary function). Instead, it tries to estimate f such that it can be as close to the data points without seeming impractical.

How does it work ?

In simple words, it builds a random forest model for each variable. Then it uses the model to predict missing values in the variable with the help of observed values.

It yield OOB (out of bag) imputation error estimate. Moreover, it provides high level of control on imputation process. It has options to return OOB separately (for

each variable) instead of aggregating over the whole data matrix. This helps to look more closely as to how accurately the model has imputed values for each variable.

3.5.4 Hmisc

Hmisc is a multiple purpose package useful for data analysis, high – level graphics, imputing missing values, advanced table making, model fitting & diagnostics (linear regression, logistic regression and cox regression) etc. Amidst, the wide range of functions contained in this package, it offers 2 powerful functions for imputing missing values. These are impute() and aregImpute(). Though, it also has transcan() function, but aregImpute() is better to use.

impute() function simply imputes missing value using user defined statistical method (mean, max, mean). It's default is median. On the other hand, aregImpute() allows mean imputation using additive regression, bootstrapping, and predictive mean matching.

In bootstrapping, different bootstrap resamples are used for each of multiple imputations. Then, a flexible additive model (non parametric regression method) is fitted on samples taken with replacements from original data and missing values (acts as dependent variable) are predicted using non-missing values (independent variable).

Then, it uses predictive mean matching (default) to impute missing values. Predictive mean matching works well for continuous and categorical (binary & multi-level) without the need for computing residuals and maximum likelihood fit.

Here are some important highlights of this package:

It assumes linearity in the variables being predicted. Fisher's optimum scoring method is used for predicting categorical variables.

3.5.5 mi

mi (Multiple imputation with diagnostics) package provides several features for dealing with missing values. Like other packages, it also builds multiple imputation models to approximate missing values. And, uses predictive mean matching method.

Though, I've already explained predictive mean matching (pmm) above, but if you haven't understood yet, here's a simpler version: For each observation in a variable with missing value, we find observation (from available values) with the closest predictive mean to that variable. The observed value from this "match" is then used as imputed value.

Below are some unique characteristics of this package:

It allows graphical diagnostics of imputation models and convergence of imputation process. It uses bayesian version of regression models to handle issue of separation. Imputation model specification is similar to regression output in R It automati-

cally detects irregularities in data such as high collinearity among variables. Also, it adds noise to imputation process to solve the problem of additive constraints.

3.6 Application

3.6.1 Lecture questions

When you suggest methods to deal with missing values to users, the recurrent question is "What is the percentage of missing values that I can have in my data set, is 50% too much but 20% OK?" What is your answer to this question?

- the answer:

  The percentage of missing values is not the only thing which is important. If the variables are highly correlated, we can predict the missing values precisely even with a high fraction of missing values. On the contrary, if the data set is very noisy to begin with, even a small fraction of missing values can be troublesome. Multiple imputation can always be performed and enables to measure precisely the variability of the predictions, which evaluates how much we can trust the results obtained from a (very) incomplete dataset.

Explain the aims of multiple imputation in comparison to single imputation.

- the answer:

  Single imputation leads to underestimating the variability of the parameters estimators because it does not account for the variability due to missing values. Multiple imputation aims at providing an estimation of the parameters of interest as well as their variability, taking into account the variability due to missing values.

3.6.2 Continuous data with missing values-Regression with missing data via Multiple Imputation

First of all you will need to install the following packages

```
install.packages("VIM")
install.packages("missMDA")
install.packages("Amelia")
```

Air pollution is currently one of the most serious public health worries worldwide. Many epidemiological studies have proved the influence that some chemical compounds, such as sulphur dioxide (SO2), nitrogen dioxide (NO2), ozone (O3) can have on our health. Associations set up to monitor air quality are active all over the

world to measure the concentration of these pollutants. They also keep a record of meteorological conditions such as temperature, cloud cover, wind, etc.

We have at our disposal 112 observations collected during the summer of 2001 in Rennes. The variables available are

- maxO3 (maximum daily ozone)
- maxO3v (maximum daily ozone the previous day)
- T12 (temperature at midday)
- T9
- T15 (Temp at 3pm)
- Vx12 (projection of the wind speed vector on the east-west axis at midday)
- Vx9 and Vx15 as well as the Nebulosity (cloud) Ne9, Ne12, Ne15

Here the final aim is to analyse the relationship between the maximum daily ozone (maxO3) level and the other meteorological variables. To do so we will perform regression to explain maxO3 in function of all the other variables. This data is incomplete (there are missing values). Indeed, it occurs frenquently to have machines that fail one day, leading to some information not recorded. We will therefore perform regression via multiple imputation.

Import the data:

```
data.ozo<-read.table("data/ozone.xlsx",header=TRUE,sep=",",row.names=1)
WindDirection <- ozo[,12]
don <- ozo[,1:11]   #### keep the continuous variables
summary(don)
```

Table 3.3: Description parameters for all variables of ozone dataset

	Min	1st Qu	Median	Mean	3rd Qu	Max	NA's
maxO3	42.00	71.00	81.50	91.24	108.25	166.00	16
t9	11.30	16.00	17.70	18.22	19.90	25.30	37
T12	14.30	18.60	20.40	21.46	23.60	33.50	33
T15	14.90	18.90	21.40	22.41	25.65	35.50	37
Ne9	0.000	3.000	5.000	4.987	7.000	8.000	34
Ne12	0.000	4.000	5.000	4.986	7.000	8.000	42
Ne15	0.00	3.00	5.00	4.60	6.25	8.00	32
Vx9	-7.8785	-3.0000	-0.8671	-1.0958	0.6919	5.1962	18
Vx12	-7.8785	-3.6941	-1.9284	-1.6853	-0.1302	6.5778	10
Vx15	-9.000	-3.759	-1.710	-1.830	0.000	3.830	21
maxO3v	42.00	70.00	82.50	89.39	101.00	166.00	12

The head of the ozone dataset are given by the following commande

```
head(don)
```

Table 3.4: Head values of the ozone dataset

	maxO3	T9	T12	T15	Ne9	Ne12	Ne15	Vx9	Vx12	Vx15	maxO3v
0601	87	15.6	18.5	NA	4	4	8	0.69	-1.71	-0.69	84
0602	82	NA	NA	NA	5	5	7	-4.33	-4.00	-3.00	87
0603	92	15.3	17.6	19.5	2	NA	NA	2.95	NA	0.52	82
0604	114	16.2	19.7	NA	1	1	0	NA	0.35	-0.17	92
0605	94	NA	20.5	20.4	NA	NA	NA	-0.50	-2.95	-4.33	114
0606	80	17.7	19.8	18.3	6	NA	7	-5.64	-5.00	-6.00	94

```
dim(don)
```

dimension are 112 row and 11 column

Load the libraries.

```
library(VIM)
library(FactoMineR)
library(missMDA)
```

When could it be a good idea to delete rows or columns with missing values to work with a complete data set?

```
dim(na.omit(don))
```

dimension are 13 row and 11 column

First, we perfom some descriptive statistics (how many missing? how many variables, individuals with missing?) and try to inspect and vizualize the pattern of missing entries and get hints on the mechanism that generated the missingness.

For this purpose, we use the R package VIM (Visualization and Imputation of Missing Values - Mathias Templ) as well as Multiple Correspondence Analysis (FactoMineR package).

The package VIM provides tools for the visualization of missing or imputed values, which can be used for exploring the data and the structure of the missing or imputed values. Depending on this structure, they may help to identify the mechanism generating the missing values or errors, which may have happened in the imputation process. You should install the package VIM, then you can check the documentation by executing

VIM: The VIM function aggr calculates and plots the amount of missing entries in each variables and in some combinations of variables (that tend to be missing simultaneously).

```
res<-summary(aggr(don, sortVar=TRUE))combinations
```

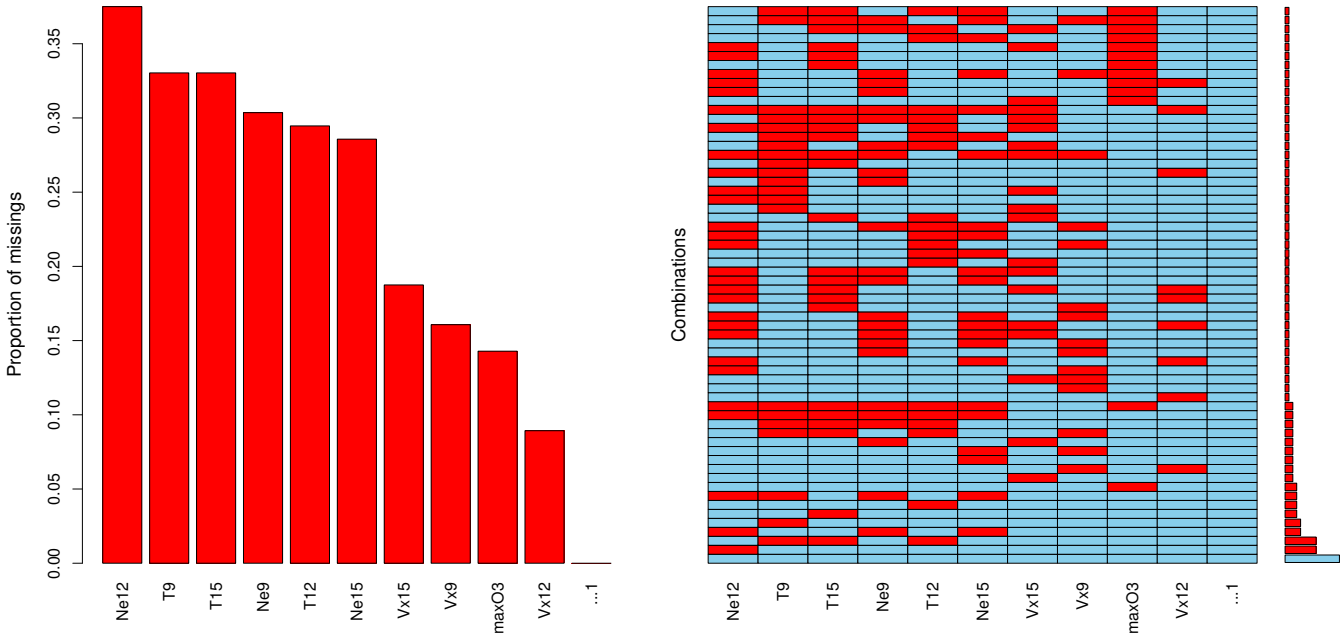

Figure 3.1: the number of missing

Table 3.5: Variables sorted by number of missings

Count	Ne12	T9	T15	Ne9	T12
Variable	0.37500000	0.33035714	0.33035714	0.30357143	0.29464286 5
Ne15	Vx15	Vx9	maxO3	maxO3v	Vx12
0.28571429	0.18750000	0.16071429	0.14285714	0.10714286	0.08928571

```
head(res[rev(order(res[,2])),])
```

Table 3.6: the combination

	Combinations	Count	Percent
1	0:0:0:0:0:0:0:0:0:0:0:0:0	13	11.607143
45	0:1:1:1:0:0:0:0:0:0:0:0:0	7	6.250000
10	0:0:0:0:0:1:0:0:0:0:0:0:0	5	4.464286
35	0:1:0:0:0:0:0:0:0:0:0:0:0	4	3.571429
41	0:1:0:0:1:1:1:0:0:0:0:0:0	3	2.678571
28	0:0:1:0:0:0:0:0:0:0:0:0:0	3	2.678571

We can see that the combination which is the most frequent is the one where all the variables are observed (13 values). Then, the second one is the one where T9, T12 and T15 are simultaneously missing (7 rows) (1 is missing, 0 is observed there is a 1 for

the second, third and fourth variables). The graph on the right panel represents the pattern, with blue for observed and red for missing.

The VIM function matrixplot creates a matrix plot in which all cells of a data matrix are visualized by rectangles. Available data is coded according to a continuous color scheme (gray scale), while missing/imputed data is visualized by a clearly distinguishable color (red).

If you use Rstudio the plot is not interactive (thus the warnings), but if you use R directly, you can click on a column of your choice, this will result in sorting the rows in the decreasing order of the values of this column. This is useful to check if there is an association between the value of a variable and the missingness of another one.

```
matrixplot(don, sortby = 2)
```

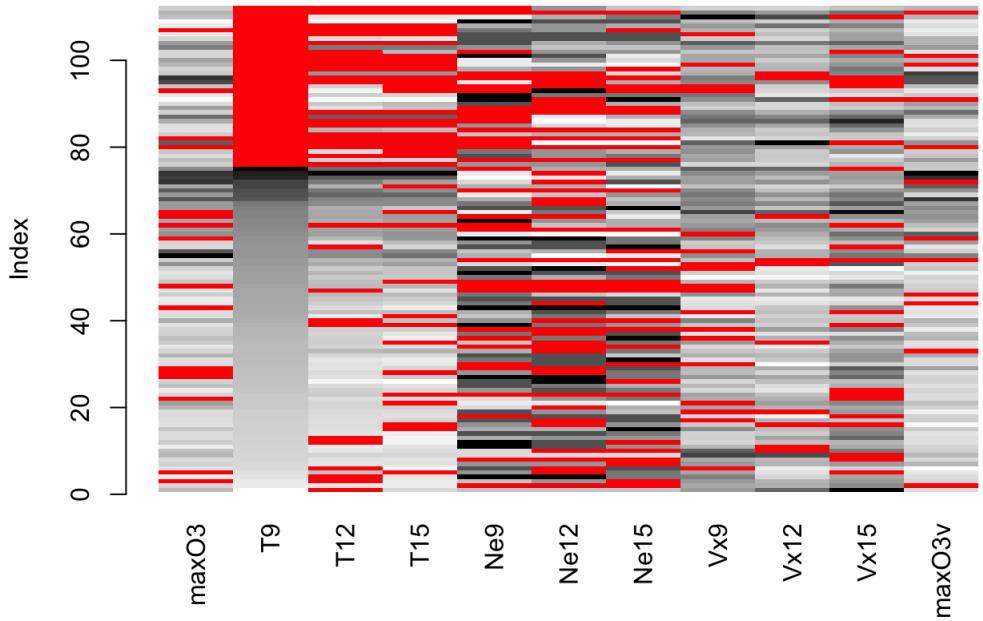

Figure 3.2: additional information on the missing values

Q2 Do you observe any associations between the missing entries ? When values are missing on a variable does it correspond to small or large values on another one ?

The VIM function marginplot creates a scatterplot with additional information on the missing values. If you plot the variables (x,y), the points with no missing values are represented as in a standard scatterplot. The points for which x (resp. y) is missing are represented in red along the y (resp. x) axis. In addition, boxplots of the x and y variables are represented along the axes with and without missing values (in red all variables x where y is missing, in blue all variables x where y is observed).

```
marginplot(don[,c("T9","maxO3")])
```

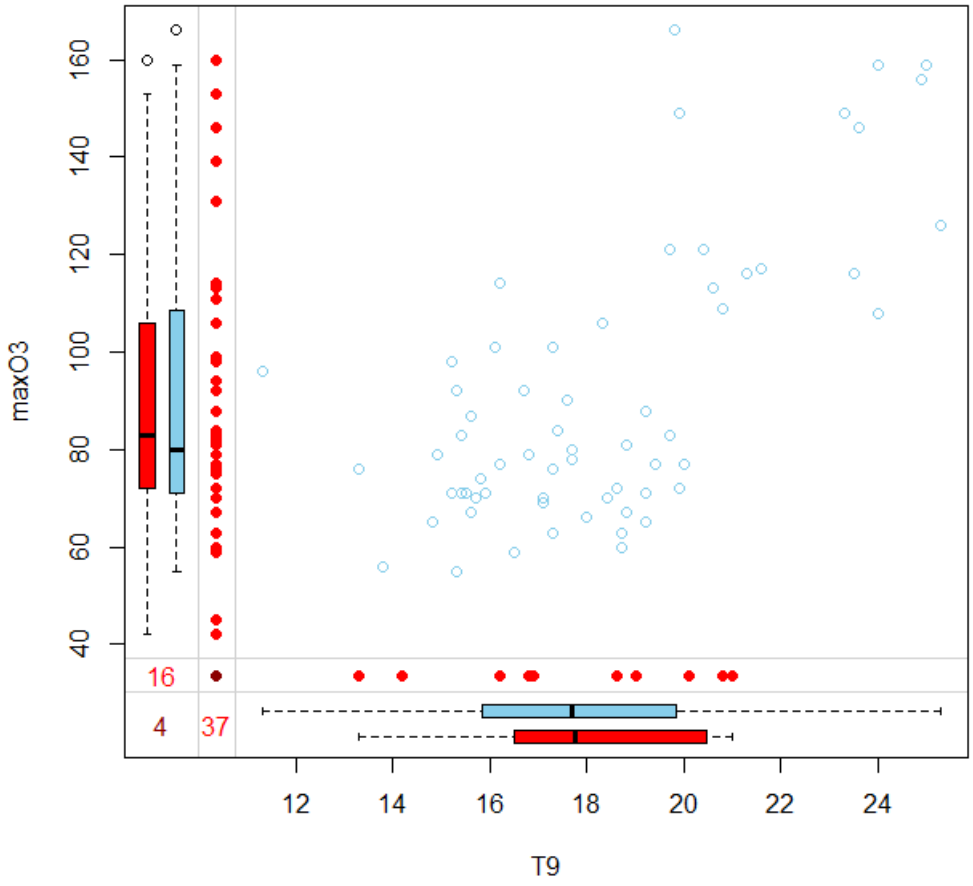

Figure 3.3: scatterplot with additional information on the missing values

We can see that the distribution of T9 is the same when maxO3 is oberved and when maxO3 is missing. If the two boxplots (red and blue) would have been very different it would imply that when maxO3 is missing the values of T9 can be very high or very low which lead to suspect the MAR hypothesis.

Create a categorical dataset with "o" when the value of the cell is observed and "m" when it is missing, and with the same row and column names as in the original data. Then, you can perform Multiple Correspondence Analysis to visualize the association with the MCA function.

```
?MCA
data_miss<-data.frame(is.na(don))
data_miss<-apply(X=data_miss, FUN=function(x) if(x) "m" else "o", MARGIN=c(1,2))
res.mca<-MCA(data_miss, graph = F)
plot(res.mca,invis="ind", title="MCA graph of the categories", cex=0.5)
```

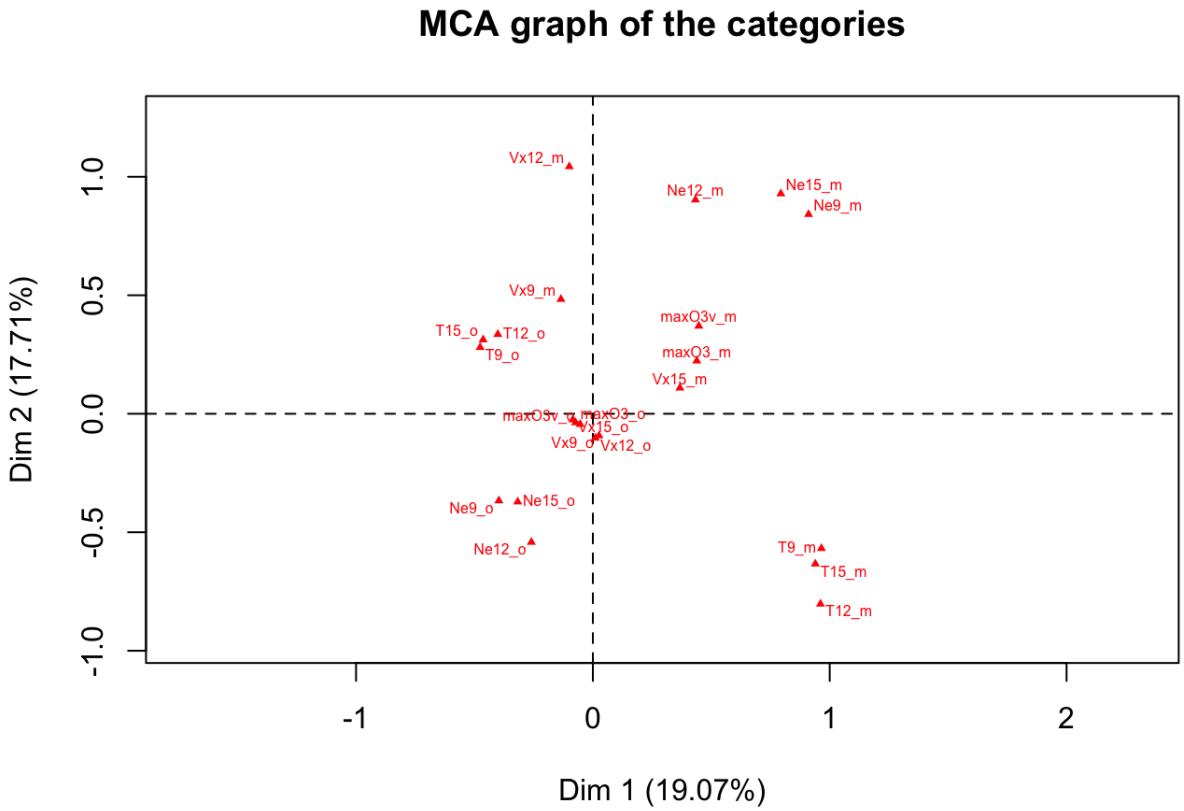

Figure 3.4: Outputs of the PCA function graph of individuals

Then, before modeling the data, we perform a PCA with missing values to explore the correlation between variables. Use the R package missMDA dedicated to perform principal components methods with missing values and to impute data with PC methods.

Determine the number of components ncp to keep using the $estim_ncpPCA$ function. Perform PCA with missing values using the imputePCA function and ncp components. Then plot the correlation circle.

```
$?estim_ncpPCA$
$?imputePCA$
```

Could you guess how cross-validation is performed to select the number of components? Then, to run the regression with missing values, we use Multiple Imputation. We impute the data either assuming 1) a Gaussian distribution (library Amelia) or 2) a PCA based model (library missMDA). Note that there are two ways to impute either using a Joint Modeling (one joint probabilitisc model for the variables all together) or a Condional Modeling (one model per variable) approach. We refer to the references given in the slides for more details.

We use the R package Amelia. We generate 100 imputed data sets with the amelia method:

library(Amelia): Loading required package: Rcpp

Amelia II: Multiple Imputation
(Version 1.8.0, built: 2021-05-26)
Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell
Refer to http://gking.harvard.edu/amelia/ for more information
-

```
?amelia
res.amelia <- amelia(don, m=100)
```

– Imputation 1–
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
– Imputation 2 –
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
– Imputation 3 –
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
– Imputation 4 –
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
– Imputation 5 –
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
...ect

```
  #names(res.amelia$imputations)
#res.amelia$imputations$imp1# the first imputed data set
```

then

```
library(mice)
imp.mice <- mice(don, m=100, defaultMethod="norm.boot")
# the variability of the parameters is obtained
```

the first imputed data set. Now generate 100 imputed data sets with the MIPCA method and 2 components. Store the result in a variable called res.MIPCA.

```
?MIPCA\\
?plot.MIPCA\\
res.MIPCA<-MIPCA(don, ncp=2, nboot=100) # MI with PCA using 2 dimensions
```

We will inspect the imputed values created to know if the imputation method should require more investigation or if we can continue and analyze the data. A common practice consists in comparing the distribution of the imputed values and of the observed values.

Check the compare.density function and apply it to compare the distributions of the T12 variable.

?compare.density

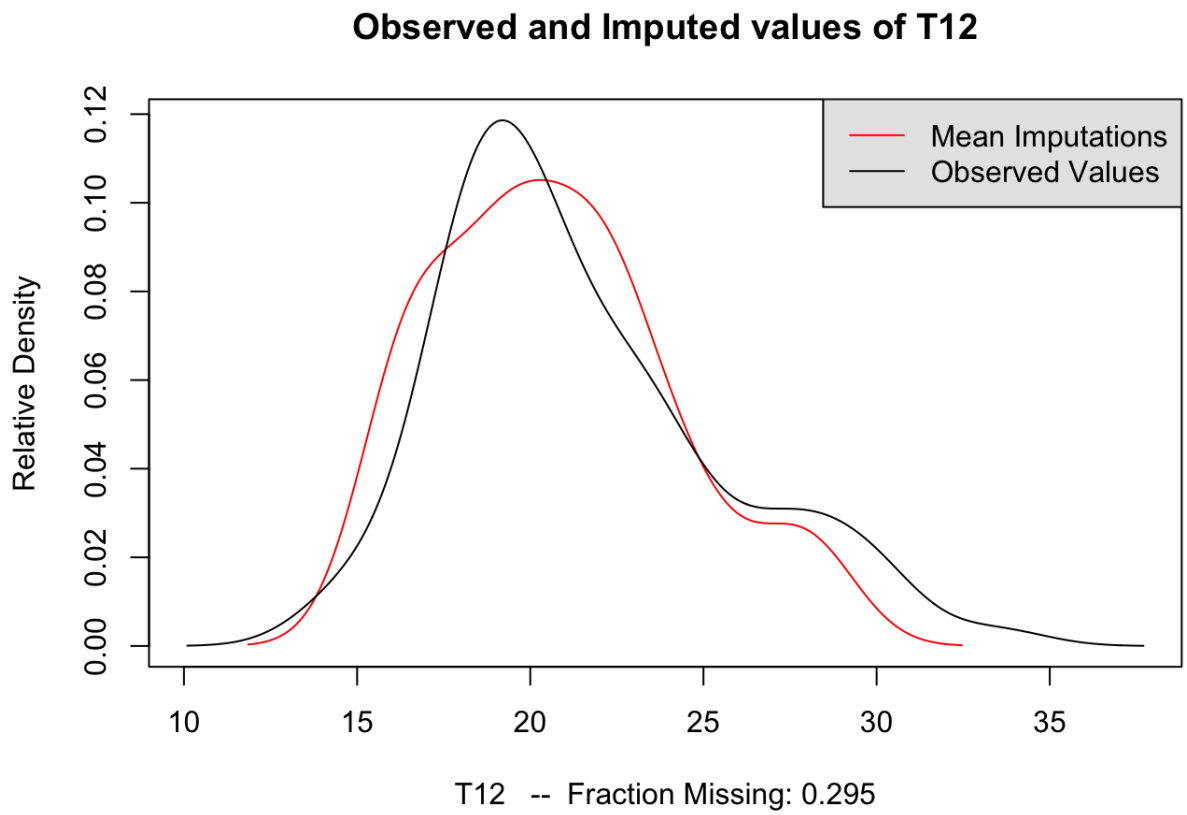

Figure 3.5: Imputation multiple T-12 Faction Missing

The quality of imputation can also be assessed with cross-validation using the overimpute function. Each observed value is deleted and for each one 100 values are predicted (using the same MI method) and the mean and 90% intervals are computed for

these 100 values. Then, we inspect whether the observed value falls within the obtained interval. On the graph, the y=x line is plotted (where the imputations should fall if they were perfect), as well as the mean (dots) and intervals (lines) for each value. Around ninety percent of these confidence intervals should contain the y = x line, which means that the true observed value falls within this range. The color of the line (as coded in the legend) represents the fraction of missing observations in the pattern of missingness for that observation (ex: blue=0-2 missing entries).

```
?overimpute
overimpute(res.amelia, var = "maxO3")
```

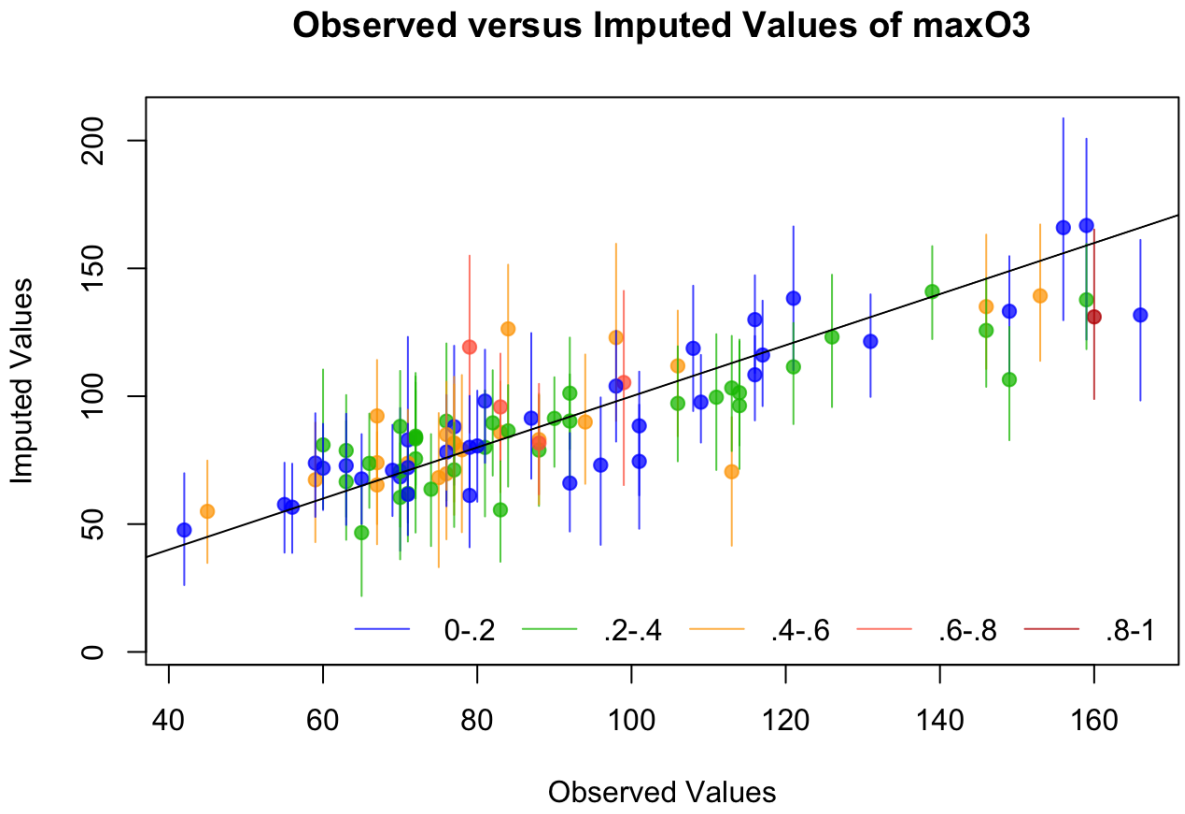

Figure 3.6: Imputation multiple Observed Values

Comment the quality of the imputation.

We can also examine the variability by projecting as supplementary tables the imputed data sets on the PCA configuration (plot the results of MI with PCA).

```
plot(res.MIPCA,choice= "ind.supp")
```

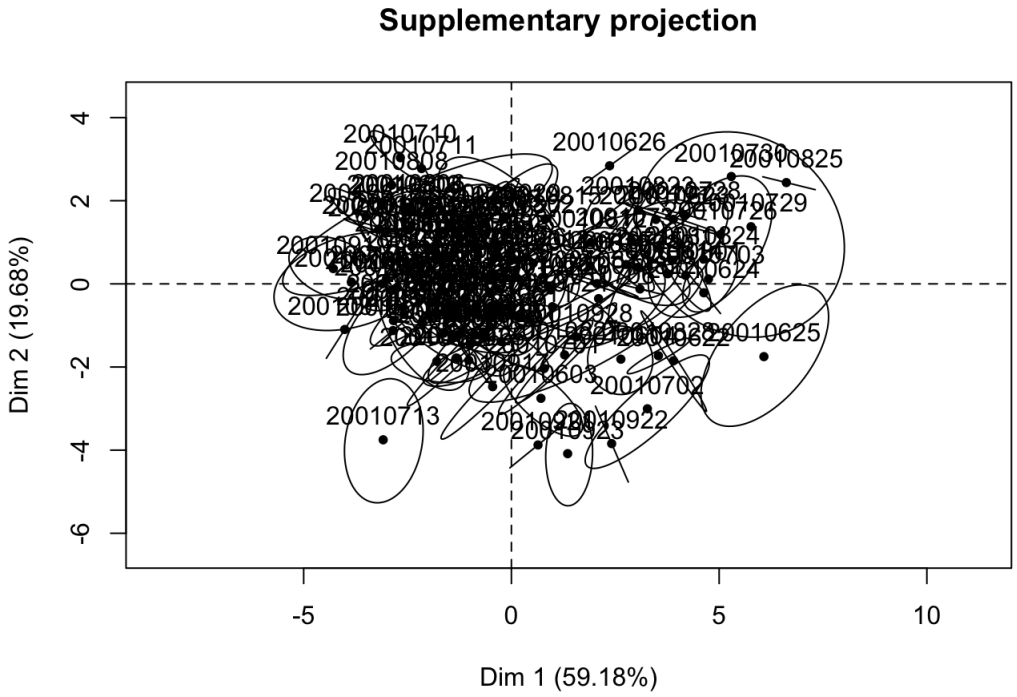

Figure 3.7: supplementary projection

```
plot(res.MIPCA,choice= "var")
```

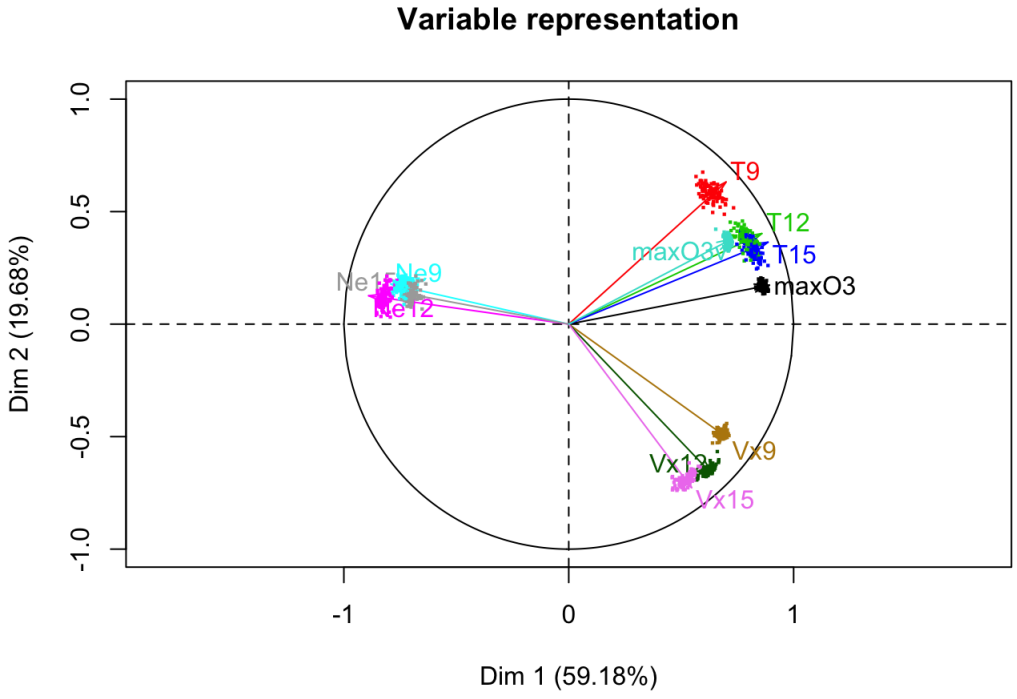

Figure 3.8: Outputs of the PCA function correlation circle

Apply a regression model on each imputed data set of the amelia method. Hint: a regression with several variables can be performed as follows

`lm(formula="maxO3 ~ T9+T12", data =don)'.

You can also use the function with Amelia package

```
resamelia <- lapply(res.amelia$imputations, as.data.frame)
# A regression on each imputed data-set
fitamelia<-lapply(resamelia, lm,
formula="maxO3~T9+T12+T15+Ne9+Ne12+Ne15+Vx9+Vx12+Vx15+maxO3v")
fitamelia <- lapply(resamelia, with,
lm(maxO3 ~T9+T12+T15+Ne9+Ne12+Ne15+Vx9+Vx12+Vx15+maxO3v))
```

Now do the same with the imputed data-sets of the MIPCA method. The package mice (Multiple Imputation by Chained Equations) allows to aggregate the results from some simple models.

library(mice)

Loading required package: lattice

?mice

pool is a function from mice to aggregate the results according to Rubin's rule

?pool

Aggregate the results of Regression with Multiple Imputation according to Rubin's rule (slide "Multiple imputation") for MI with amelia with the pool function from the mice package.

```
poolamelia<-pool(as.mira(fitamelia))
summary(poolamelia)
```

Table 3.7: the results of Regression with Multiple Imputation and Amelia package

	estimate	std.error	statistic	df	p.value
Intercept	19.3008802	20.4269267	0.9448744	9.156416	0.36334317
T9	1.0365941	3.7064439	0.2796735	3.725997	0.78448590
T12	1.3936132	3.1734209	0.4391517	4.852282	0.66834788
T15	0.6400731	1.9088331	0.3353217	6.137191	0.74316648
Ne9	-1.2425404	1.6724030	-0.7429671	5.934270	0.47178381
Ne12	-2.7872454	2.9442641	-0.9466696	5.307437	0.36246397
Ne15	0.8739977	1.3634798	0.6410052	12.011132	0.53355926
Vx9	0.8666599	1.6405449	0.5282756	7.015473	0.60693100
Vx12	0.4887195	2.1687298	0.2253482	6.209202	0.82549695
Vx15	0.4964256	1.4392222	0.3449263	9.386397	0.73611455
maxO3v	0.2981957	0.1203265	2.4782219	7.689351	0.02903622

Now do the same with the MIPCA results.

```
poolMIPCA<-pool(as.mira(fitMIPCA))
summary(poolMIPCA)
```

Table 3.8: the results of Regression with Multiple Imputation and MIPCA package

	estimate	std.error	statistic	df	p.value
Intercept	12.7547158	18.44444620	0.6915207	62.65724	0.491533365
T9	1.0124513	1.32664563	0.7631663	43.94939	0.447937301
T12	1.5481541	1.02177779	1.5151573	53.57541	0.134250639
T15	0.8270436	0.90466203	0.9142018	56.48618	0.363759801
Ne9	-1.0669031	1.18182976	-0.9027553	56.20932	0.369762308
Ne12	-1.8046512	1.56165049	-1.1556051	50.39130	0.251785274
Ne15	0.3818986	1.16884956	0.3267303	62.74103	0.744850124
Vx9	0.8234073	1.06833916	0.7707358	69.73461	0.443466007
Vx12	0.9935546	1.14848906	0.8650971	69.76246	0.389950634
Vx15	0.2081217	1.13987925	0.1825822	65.20298	0.855655377
maxO3v	0.2502536	0.09206577	2.7182051	67.67098	0.008273753

Write a function that removes the variables with the largest p-values step by step (each time a variable is removed the regression model is performed again) until all variables are significant.

```
don2 <- don
reg <- lm(maxO3 ~. , data = don2)
while(any(summary(reg)$coeff[-1, 4]>0.05)){
don2 <- don2[,!(colnames(don2)%in%names
(which.max(summary(reg)$coeff[-1, 4])))]
reg <- lm(maxO3 ~. , data = don2)}
```

We combine the results and performed the regression with missing values

```
# Submodel to compare
fitMIPCA<-lapply(res.MIPCA,lm, formula="maxO3~ T12+Ne9+Vx12+maxO3v")
poolMIPCA<-pool(as.mira(fitMIPCA))
summary(poolMIPCA)
```

Table 3.9: Coefficients of regression with MIPCA package

	estimate	std.error	statistic	df	p.value
(Intercept)	9.4746829	14.26052896	0.6643991	66.39501	0.5085254775
T12	2.9522183	0.63368128	4.6588378	63.63925	0.0000139317
Ne9	-1.8307128	1.07180284	-1.7080686	61.45186	0.0918675454
Vx12	1.7929235	0.74833086	2.3958968	69.73607	0.0191404878
maxO3v	0.3286382	0.08415626	3.9050956	73.09463	0.0002079746

```
    #lm.mice.out <- with(imp.mice, lm(maxO3 ~ T12+Ne9+Vx12+maxO3v))
#pool.mice <- pool(lm.mice.out)
#summary(pool.mice)
fitamelia<-lapply(resamelia,lm, formula="maxO3~ T12+Ne9+Vx12+maxO3v")
poolamelia<-pool(as.mira(fitamelia))
summary(poolamelia)
```

Table 3.10: Coefficients of regression Amelia

	estimate	std.error	statistic	df	p.value
(Intercept)	6.3208449	14.98158031	0.4219078	11.699053	6.767519e-01
T12	3.1149265	0.58410855	5.3327869	20.590983	1.653628e-05
Ne9	-2.0510525	1.52007385	-1.3493111	5.446113	1.895097e-01
Vx12	1.6882470	0.88284691	1.9122761	9.195620	6.753819e-02
maxO3v	0.3400338	0.07501389	4.5329447	24.632555	1.289599e-04

3.7 Gaussian Mixture Models Explained

This model is a soft probabilistic clustering model that allows us to describe the membership of points to a set of clusters using a mixture of Gaussian densities. It is a soft classification (in contrast to a hard one) because it assigns probabilities of belonging to a specific class instead of a definitive choice.

In essence, each observation will belong to every class but with different probabilities.

In the world of Machine Learning, we can distinguish two main areas: Supervised and unsupervised learning. The main difference between both lies in the nature of the data as well as the approaches used to deal with it.

Clustering is an unsupervised learning problem where we intend to find clusters of points in our data-set that share some common characteristics.

Let's suppose we have a data-set that looks like this:

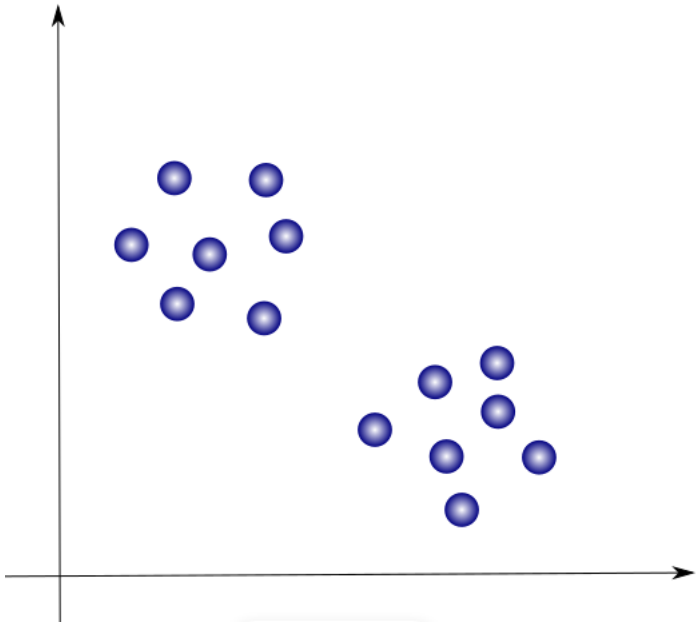

Figure 3.9: data-set

Our job is to find sets of points that appear close together. In this case, we can clearly identify two clusters of points which we will colour blue and red, respectively:

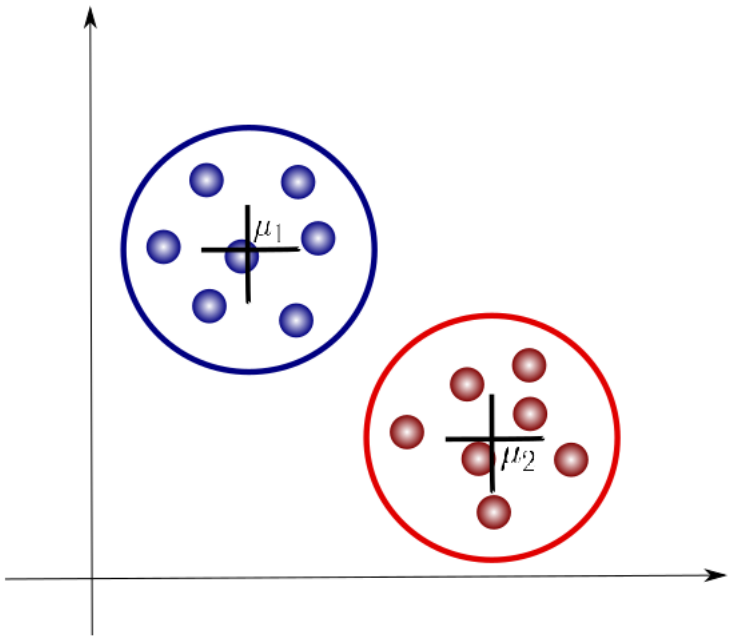

Figure 3.10: clustering data-set

Definitions

A Gaussian Mixture is a function that is comprised of several Gaussian; each identified by $k \in 1,\ldots,K$, where K is the number of clusters of our data-set. Each Gaussian k in

the mixture is comprised of the following parameters:

• A mean μ that defines its centre.

• A covariance $\sum$ that defines its width. This would be equivalent to the dimensions of an ellipsoid in a multivariate scenario.

• A mixing probability π that defines how big or small the Gaussian function will be.

Let us now illustrate these parameters graphically:

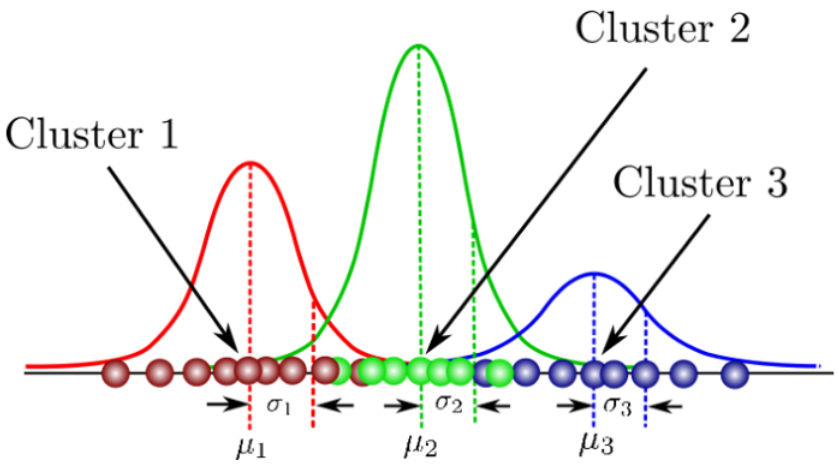

Figure 3.11: the clustering of three Gaussian functions

Here, we can see that there are three Gaussian functions, hence $K = 3$. Each Gaussian explains the data contained in each of the three clusters available. The mixing coefficients are themselves probabilities and must meet this condition:

$$\sum_{k=1}^{K} \pi_k = 1 \tag{3.1}$$

Now how do we determine the optimal values for these parameters? To achieve this we must ensure that each Gaussian fits the data points belonging to each cluster. This is exactly what maximum likelihood does.

In general, the Gaussian density function is given by:

$$N(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}|\sum|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{3.2}$$

Where x represents our data points, D is the number of dimensions of each data point. μ and Σ are the mean and covariance, respectively. If we have a dataset comprised of $N = 1000$ three-dimensional points ($D = 3$), then x will be a 1000×3 matrix. μ will be a 1×3 vector, and Σ will be a 3×3 matrix. For later purposes, we will also find it useful to take the log of this equation, which is given by:

$$\ln N(x|\mu,\Sigma) = -\frac{D}{2}\ln 2\pi - \frac{1}{2}\ln\sum -\frac{1}{2}(x-\mu)^T(\Sigma)^{-1}(x-\mu) \tag{3.3}$$

If we differentiate this equation with respect to the mean and covariance and then equate it to zero, then we will be able to find the optimal values for these parameters, and the solutions will correspond to the Maximum Likelihood Estimates (MLE) for this setting. However, because we are dealing with not just one, but many Gaussian's, things will get a bit complicated when time comes for us to find the parameters for the whole mixture.

3.7.1 Application 1

The galaxies data in the MASS package (Venables and Ripley, 2002)[40] is a frequently used example for Gaussian mixture models. It contains the velocities of 82 galaxies from a redshift survey in the Corona Borealis region. Clustering of galaxy velocities reveals information about the large scale structure of the universe.

```
library(MASS)
data(galaxies)
X = galaxies / 1000
```

The Mclust function from the mclust package (Fraley et al., 2012)[54] is used to fit Gaussian mixture models. The code below fits a model with $G = 4$ components to the galaxies data, allowing the variances to be unequal (model="V").

```
library(mclust, quietly=TRUE)
fit = Mclust(X, G=4, model="V")
summary(fit)
```

figure 3.12 shows the resulting density plot.

```
plot(fit, what="density", main="", xlab="Velocity (Mm/s)")
rug(X)
```

Section 6.2 of Drton and Plummer (2017)[55] considers singular BIC for Gaussian mixture models using the galaxies data set as an example. Singularities occur when two mixture components coincide (i.e. they have the same mean and variance) or on the boundary of the parameter space where the prior probability of a mixture component is zero.

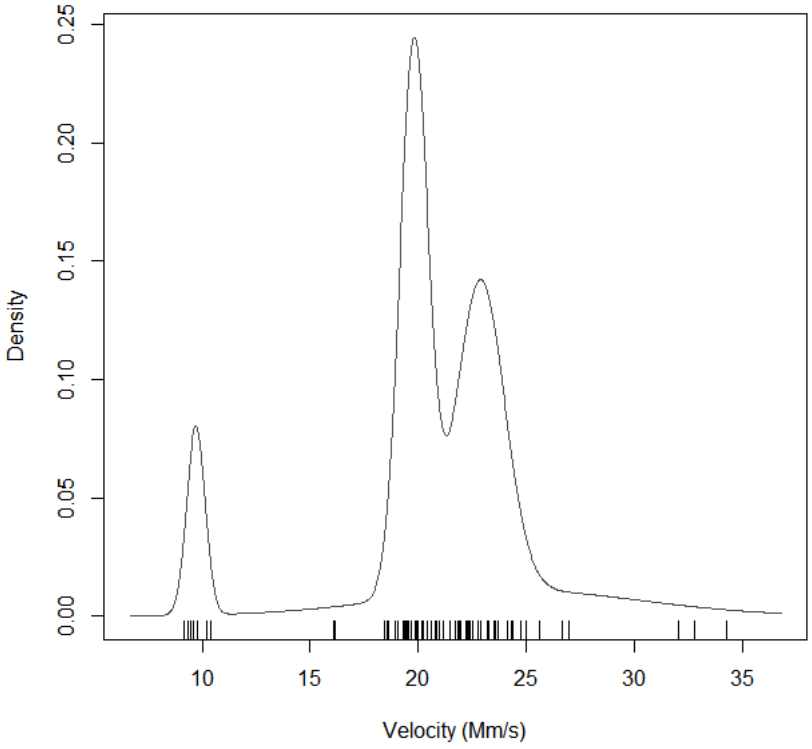

Figure 3.12: Density estimate for galaxies data from a 4-component mixture model

The GaussianMixtures() function creates an object representing a family of mixture models up to a specified maximum number of components (maxNumComponents=10 in this example). The phi parameter controls the penalty to be used for sBIC (See below) and the restarts parameter determines the number of times each model is fitted starting from randomly chosen starting points. Due to the multi-modal likelihood surface for mixture models, multiple restarts are used to find a good (local) maximum.

```
library(sBIC)
gMix=GaussianMixtures(maxNumComponents=10, phi=1, restarts=100)
```

Learning coefficients are known exactly for Gaussian mixtures with known and equal variances, but this model is rarely applied in practice. For unequal variances, the learning coefficients are unknown, but upper bounds are given by Drton and Plummer (2017, equation 6.11)[55]. These bounds are implemented by setting the penalty parameter phi=1 in the GaussianMixtures() function. We refer to the singular BIC using these approximate penalties as sBIC1. It is calculated by supplying the data X and the model set gMix to the sBIC() function. The RNG seed is set for reproducibility, due to the random restarts.

```
set.seed(1234)
```

```
m = sBIC(X, gMix)
print(m)
```

figure 3.12 compares BIC with sBIC1. Both criteria have been standardized so that the value for the 1-component model is 0. This figures reproduces Figure 7 of Drton and Plummer (2017)[32]. The reproduction is not exact because, in the interests of speed, we have reduced the number of restarts from 5000 to 100. This mainly affects the models with larger number of components.

```
matplot(
cbind(m$BIC - m$BIC[1], m$sBIC - m$sBIC[1]),
pch = c(1, 3),
col = "black",
xlab = "Number of components",
ylab = expression(BIC - BIC(M[1])),
las=1, xaxt="n"
)
axis(1, at = 1:10)
legend("topleft",
c(expression(BIC), expression(bar(sBIC)[1])),
pch = c(1, 3),
y.intersp = 1.2)
```

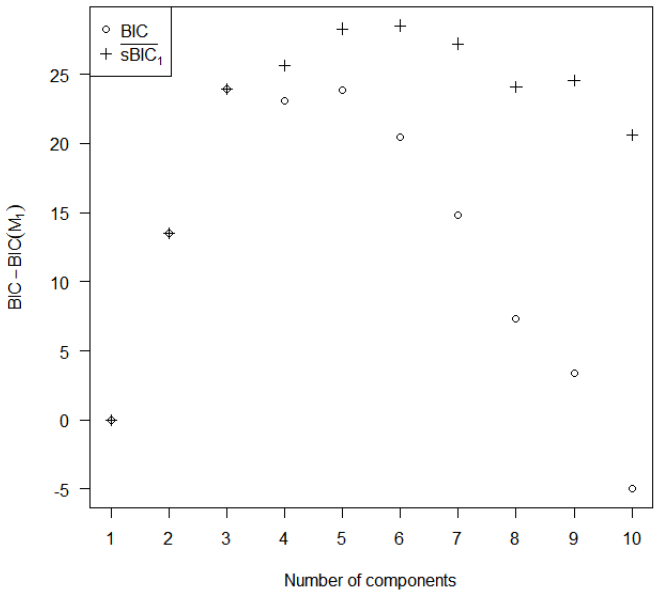

Figure 3.13: Comparison of singular BIC with BIC for choosing the number of components in the galaxies data

The BIC and singular BIC results for the galaxies data can be compared with the posterior probabilities for the number of components derived by Richardson and Green (1997, Table 1)[56] using reversible jump MCMC. Since Richardson and Green (1997) [57] consider up to 14 components, we truncate the distribution up to 10 components and renormalize

```
post.MCMC = c(0.000, 0.000, 0.061, 0.128, 0.182, 0.199, 0.160,
0.109, 0.071, 0.040, 0.023, 0.013, 0.006, 0.003)[1:10]
post.MCMC = post.MCMC / sum(post.MCMC)
```

The posterior probabilities from BIC and sBIC1 are derived by exponentiating and then renormalizing using the helper function postBIC().

```
postBIC <- function(BIC) {
prob <- exp(BIC - max(BIC))
prob/sum(prob) }
normalizedProbs=rbind("BIC"=postBIC(m$BIC),
"sBIC1"=postBIC(m$sBIC), "MCMC"=post.MCMC)
```

3.14 compares the posterior densities from the three approaches. This reproduces figure 8 from Drton and Plummer (2017) [55].

```
barplot(
normalizedProbs,
beside = TRUE,
col = c("white","grey","black"),
legend = c(expression(BIC), expression(bar(sBIC)[1]), expression(MCMC)),
xlab = "Number of components",
ylab = "Probability",
args.legend = list(y.intersp = 1.2),
names.arg = 1:10
)
```

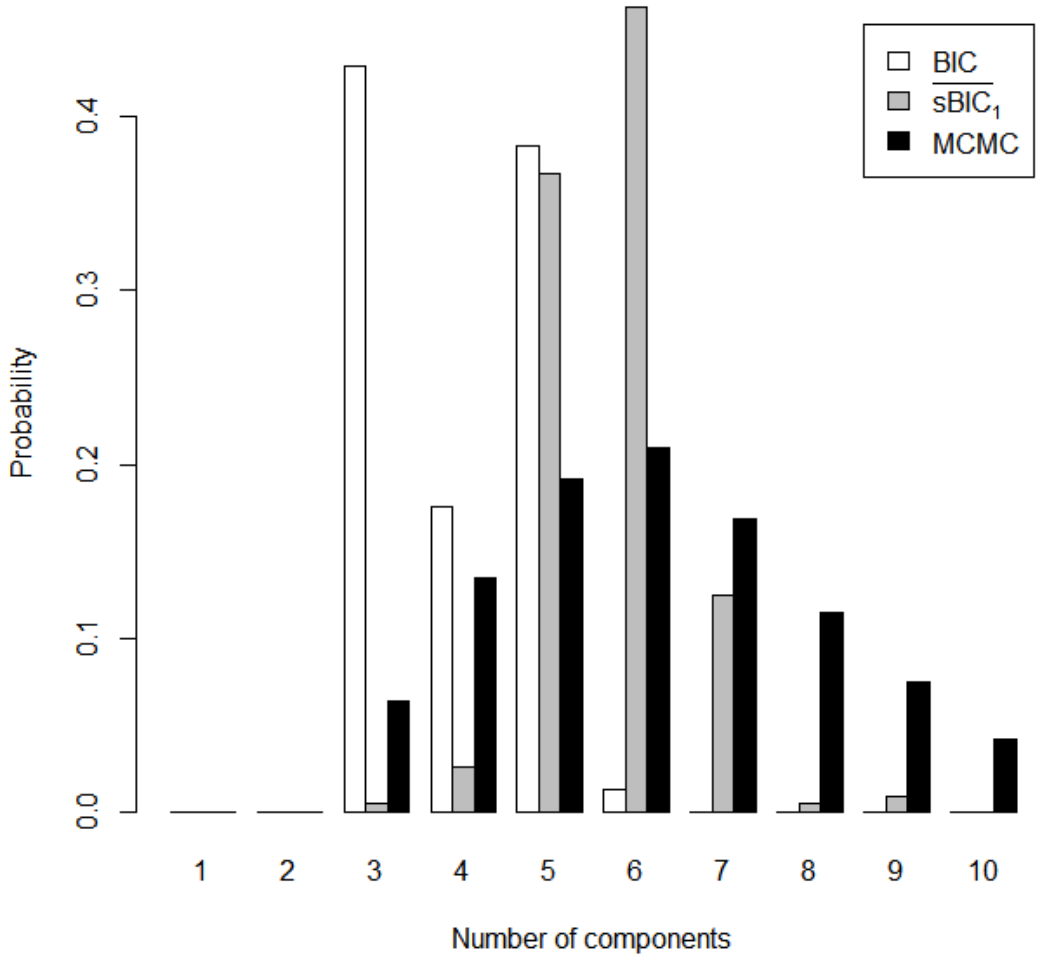

Figure 3.14: Posterior distribution of the number of components in a Gaussian mixture model with unequal variances applied to the galaxies data

3.7.2 Application 2

In this example, we will assume our mixture components are fully specified Gaussian distributions (i.e the means and variances are known), and we are interested in finding the maximum likelihood estimates of the π_k's.

Assume we have $K = 2$ components, so that:

$$\begin{aligned} X_i|Z_i &= 0 \sim N(5, 1.5) \\ X_i|Z_i &= 1 \sim N(10, 2) \end{aligned}$$

The true mixture proportions will be $P(Z_i = 0) = 0.25$ and $P(Z_i = 1) = 0.75$. First we simulate data from this mixture model:

```
# mixture components
mu.true    = c(5, 10)
sigma.true = c(1.5, 2)
# determine Z_i
Z = rbinom(500, 1, 0.75)
# sample from mixture model
X <- rnorm(10000, mean=mu.true[Z+1], sd=sigma.true[Z+1])
hist(X,breaks=15)
```

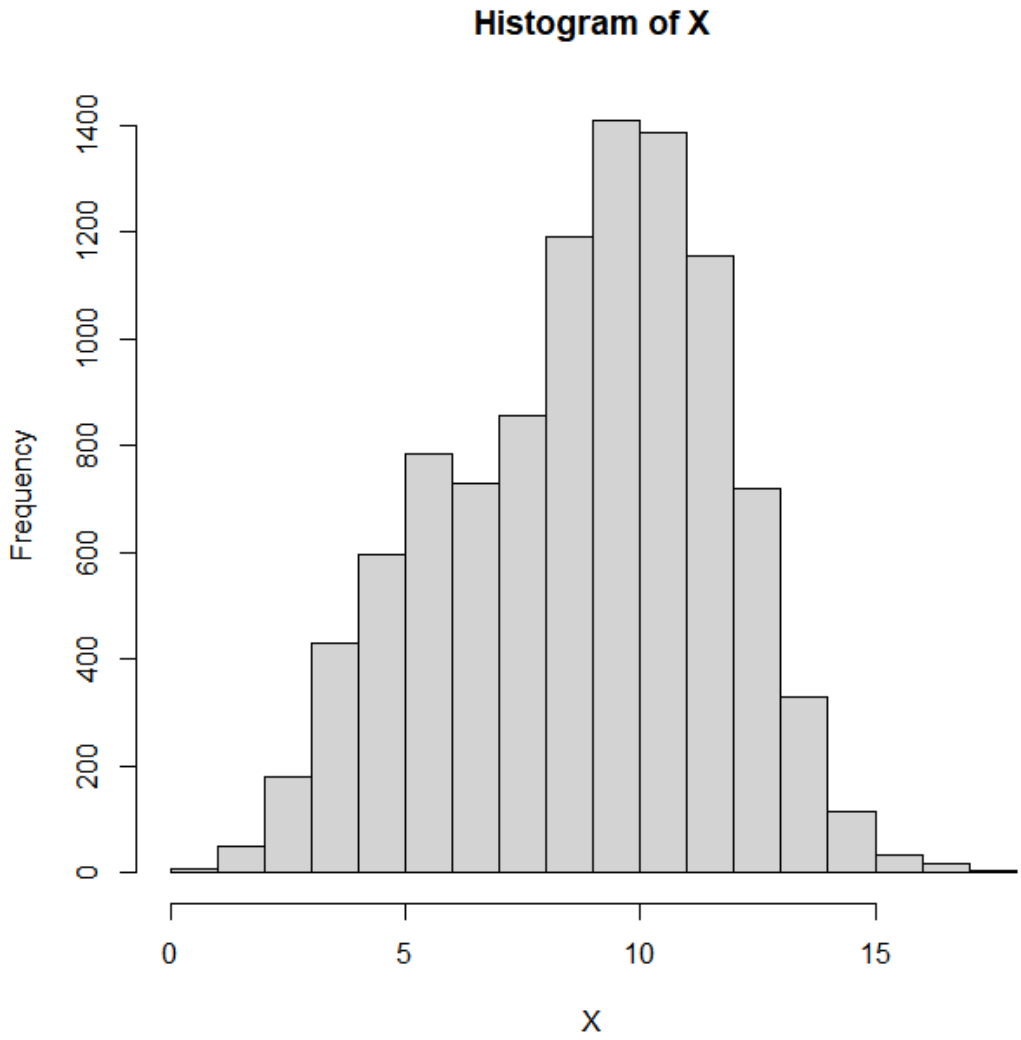

Figure 3.15: Histogram of X

Now we write a function to compute the log-likelihood for the incomplete data, assuming the parameters are known. This will be used to determine convergence:

$$l(\theta) = \sum_{i=1}^{n} \log(\sum_{k=1}^{2} \pi_k N(x_i; \mu_k, \sigma_k^2)) \tag{3.4}$$

```
compute.log.lik <- function(L, w) {
L[,1] = L[,1]*w[1]
L[,2] = L[,2]*w[2]
return(sum(log(rowSums(L))))}
```

Since the mixture components are fully specified, for each sample Xi we can compute the likelihood $P(X_i|Z_i = 0)$ and $P(X_i|Z_i = 1)$. We store these values in the columns of L:

```
L = matrix(NA, nrow=length(X), ncol= 2)
L[, 1] = dnorm(X, mean=mu.true[1], sd = sigma.true[1])
L[, 2] = dnorm(X, mean=mu.true[2], sd = sigma.true[2])
```

Finally, we implement the E and M step in the EM.iter function below. The mixture.EM function is the driver which checks for convergence by computing the log-likelihoods at each step.

```
mixture.EM <- function(w.init, L) {
w.curr <- w.init
# store log-likehoods for each iteration
log_liks <- c()
ll<-compute.log.lik(L, w.curr)
log_liks <- c(log_liks, ll)
delta.ll <- 1
while(delta.ll > 1e-5) {
    w.curr   <- EM.iter(w.curr, L)
    ll       <- compute.log.lik(L, w.curr)
    log_liks <- c(log_liks, ll)
    delta.ll <- log_liks[length(log_liks)]  - log_liks[length(log_liks)-
1]  }
  return(list(w.curr, log_liks)) }
EM.iter <- function(w.curr, L, ...) {
    # E-step: compute E_{Z|X,w0}[I(Z_i = k)]
  z_ik <- L
  for(i in seq_len(ncol(L))) {
    z_ik[,i] <- w.curr[i]*z_ik[,i]  }
  z_ik<-z_ik / rowSums(z_ik)
  # M-step
  w.next   <- colSums(z_ik)/sum(z_ik)
  return(w.next)    }

#perform EM
```

```
ee <- mixture.EM(w.init=c(0.5,0.5), L)
print(paste("Estimate=(", round(ee[[1]][1],2), ",",
round(ee[[1]][2],2), ")", sep=""))
```

Estimate of μ et σ are given by $(0.2748159, 0.7251841)$.
Finally, we inspect the evolution of the log-likelihood and note that it is strictly increases:

```
plot(ee[[2]], ylab='incomplete log-likelihood', xlab='iteration')
```

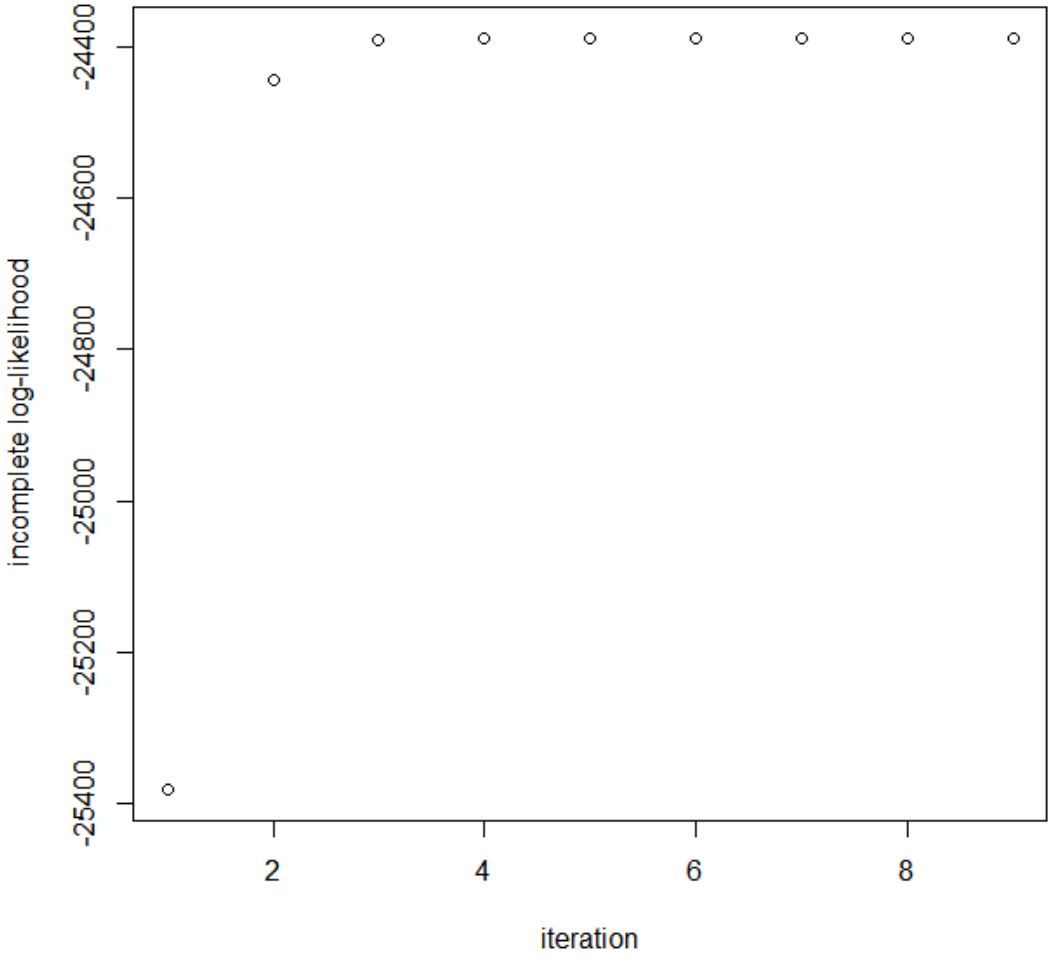

Figure 3.16: Iteration of EM algorithm

CONCLUSION

The impact of missing data on quantitative research can be serious, leading to biased estimates of parameters, loss of information, decreased statistical power, increased standard errors, and weakened generalizability of findings. In this paper, we discussed and we demonstrated expectation-maximization algorithm, applied to a real-world data set. Results were contrasted with those obtained from the complete data set , We give various examples of applications of the EM algorithm to the resolution of some problems of missing data. We used with some incredible R packages for missing values imputation. These packages arrive with some inbuilt functions and a simple syntax to impute missing data at once. Some packages are known best working with continuous variables and others for categorical.

BIBLIOGRAPHY

[1] Joseph L Schafer. *Analysis of incomplete multivariate data*. CRC press, 1997.

[2] Judi Scheffer. Dealing with missing data. 2002.

[3] Shinichi Nakagawa. Missing data: mechanisms, methods and messages. *Ecological statistics: Contemporary theory and application*, pages 81–105, 2015.

[4] James R Carpenter, James H Roger, and Michael G Kenward. Analysis of longitudinal trials with protocol deviation: a framework for relevant, accessible assumptions, and inference via multiple imputation. *Journal of biopharmaceutical statistics*, 23(6):1352–1371, 2013.

[5] Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[6] Roderick J A. Little and Donald B Rubin. Incomplete data. *Wiley StatsRef: Statistics Reference Online*, 2014.

[7] Esther-Lydia Silva-Ramírez, Rafael Pino-Mejías, Manuel López-Coello, and María-Dolores Cubiles-de-la Vega. Missing value imputation on missing completely at random data using multilayer perceptrons. *Neural Networks*, 24(1):121–129, 2011.

[8] Therese D Pigott. A review of methods for missing data. *Educational research and evaluation*, 7(4):353–383, 2001.

[9] Kancherla Jonah Nishanth and Vadlamani Ravi. A computational intelligence based online data imputation method: An application for banking. *Journal of Information Processing Systems*, 9(4):633–650, 2013.

[10] Barbara K Scheffer and M Gaie Rubenfeld. A consensus statement on critical thinking in nursing, 2000.

[11] Tshilidzi Marwala. *Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques: Knowledge Optimization Techniques*. IGI Global, 2009.

[12] Fulufhelo V Nelwamondo, Shakir Mohamed, and Tshilidzi Marwala. Missing data: A comparison of neural network and expectation maximization techniques. *Current Science*, pages 1514–1521, 2007.

[13] Elena M Hernández-Pereira, Diego Álvarez-Estévez, and Vicente Moret-Bonillo. Automatic classification of respiratory patterns involving missing data imputation techniques. *Biosystems Engineering*, 138:65–76, 2015.

[14] Steven C Wofsy. Hiaper pole-to-pole observations (hippo): fine-grained, global-scale measurements of climatically important atmospheric gases and aerosols. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 369(1943):2073–2086, 2011.

[15] Cameron W Brennan, Roel GW Verhaak, Aaron McKenna, Benito Campos, Houtan Noushmehr, Sofie R Salama, Siyuan Zheng, Debyani Chakravarty, J Zachary Sanborn, Samuel H Berman, et al. The somatic genomic landscape of glioblastoma. *Cell*, 155(2):462–477, 2013.

[16] Paul D Allison. *Missing data*. Sage publications, 2001.

[17] Truett Allison, Aina Puce, and Gregory McCarthy. Social perception from visual cues: role of the sts region. *Trends in cognitive sciences*, 4(7):267–278, 2000.

[18] Donald B Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pages 34–58, 1978.

[19] Marco Ramoni and Paola Sebastiani. Robust bayes classifiers. *Artificial Intelligence*, 125(1-2):209–226, 2001.

[20] Michikazu Nakai and Weiming Ke. Review of the methods for handling missing data in longitudinal data analysis. *International Journal of Mathematical Analysis*, 5(1):1–13, 2011.

[21] John W Graham et al. Missing data analysis: Making it work in the real world. *Annual review of psychology*, 60(1):549–576, 2009.

[22] Richard Bauman, Charles L Briggs, Charles S Briggs, et al. *Voices of modernity: Language ideologies and the politics of inequality*. Number 21. Cambridge University Press, 2003.

[23] Marina Soley-Bori. Dealing with missing data: Key assumptions and methods for applied analysis. *Boston University*, 4(1):19, 2013.

[24] Michikazu Nakai and Weiming Ke. Review of the methods for handling missing data in longitudinal data analysis. *International Journal of Mathematical Analysis*, 5(1):1–13, 2011.

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

[26] Donna Spiegelman and Ellen Hertzmark. Easy sas calculations for risk or prevalence ratios and differences. *American journal of epidemiology*, 162(3):199–200, 2005.

[27] Craig K Enders and Deborah L Bandalos. The relative performance of full information maximum likelihood estimation for missing data in structural equation models. *Structural equation modeling*, 8(3):430–457, 2001.

[28] Paul D Allison. Missing data techniques for structural equation modeling. *Journal of abnormal psychology*, 112(4):545, 2003.

[29] MA Patterson, S-C Kong, GJ Hampson, and Rolf D Reitz. Modeling the effects of fuel injection characteristics on diesel engine soot and nox emissions. *SAE transactions*, pages 836–852, 1994.

[30] John M Abowd. The us census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, 2018.

[31] Ning Cao, Shiyong Li, Zhaoqing Wang, Kazi Mokim Ahmed, Michael E Degnan, Ming Fan, Joseph R Dynlacht, and Jian Jian Li. Nf-κb-mediated her2 overexpression in radiation-adaptive resistance. *Radiation research*, 171(1):9–21, 2009.

[32] 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)*, 39(1):1–22, 1977.

[33] William Léchelle, Fabrizio Gotti, and Philippe Langlais. Wire57: A fine-grained benchmark for open information extraction. *arXiv preprint arXiv:1809.08962*, 2018.

[34] Herman O Hartley. Maximum likelihood estimation from incomplete data. *Biometrics*, 14(2):174–194, 1958.

[35] Jae-On Kim and James Curry. The treatment of missing data in multivariate analysis. *Sociological Methods & Research*, 6(2):215–240, 1977.

[36] JE Gentle. The em algorithm and extensions. *Biometrics*, 54(1):395, 1998.

[37] William Hatherell. 'words and things': Locke, hartley and the associationist context for the preface to lyrical ballads. *Romanticism*, 12(3):223–235, 2006.

[38] Wiki How English. Expectation–maximization algorithm.

[39] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

[40] William N Venables and Brian D Ripley. Random and mixed effects. In *Modern applied statistics with S*, pages 271–300. Springer, 2002.

[41] Paul Van Mele. A historical review of research on the weaver ant oecophylla in biological control. *Agricultural and forest entomology*, 10(1):13–22, 2008.

[42] Liyun Rao, Renjie He, Youhua Wang, Weili Yan, Jing Bai, and Datian Ye. An efficient improvement of modified newton-raphson algorithm for electrical impedance tomography. *IEEE transactions on magnetics*, 35(3):1562–1565, 1999.

[43] Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[44] Saba Akram and Quarrat Ul Ann. Newton raphson method. *International Journal of Scientific & Engineering Research*, 6(7):1748–1752, 2015.

[45] Lauritz B Holm-Nielsen and Peter M Jargensen. Passiflora tryphostemmatoides andits allies.

[46] SN Rai and DE Matthews. Improving the em algorithm. *Biometrics*, pages 587–591, 1993.

[47] Anthony YC Kuk and Yuk W Cheng. The monte carlo newton-raphson algorithm. *Journal of Statistical Computation and Simulation*, 59(3):233–250, 1997.

[48] Kenneth Lange. A gradient algorithm locally equivalent to the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(2):425–437, 1995.

[49] Mortaza Jamshidian and Robert I Jennrich. Acceleration of the em algorithm by using quasi-newton methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):569–587, 1997.

[50] Xiao-Li Meng and Donald B Rubin. Using em to obtain asymptotic variance-covariance matrices: The sem algorithm. *Journal of the American Statistical Association*, 86(416):899–909, 1991.

[51] Patricia A Patrician. Multiple imputation for missing data. *Research in nursing & health*, 25(1):76–84, 2002.

[52] Donald B Rubin. An overview of multiple imputation. In *Proceedings of the survey research methods section of the American statistical association*, pages 79–84. Citeseer, 1988.

[53] Chain Monte Carlo. Markov chain monte carlo and gibbs sampling. *Lecture notes for EEB*, 581:540, 2004.

[54] Chris Fraley, Adrian E Raftery, T Brendan Murphy, and Luca Scrucca. mclust version 4 for r: normal mixture modeling for model-based clustering, classification, and density estimation. Technical report, Technical report, 2012.

[55] Mathias Drton and Martyn Plummer. A bayesian information criterion for singular models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2):323–380, 2017.

[56] Marjon van Slegtenhorst, Ronald de Hoogt, Caroline Hermans, Mark Nellist, Bart Janssen, Senno Verhoef, Dick Lindhout, Ans van den Ouweland, Dicky Halley, Janet Young, et al. Identification of the tuberous sclerosis gene tsc1 on chromosome 9q34. *Science*, 277(5327):805–808, 1997.

[57] Sylvia Richardson and Peter J Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792, 1997.