Multivariate Analysis: Imputing missing data and detecting outliers

Pol Serra i Lidón
Polytechnic University of Catalonia
FIB, MIRI, Multivariate Analysis
Barcelona, Spain
Email: pol_serra_lidon@outlook.es

Abstract—**This work about Multivariate Analysis is focused on the concepts of imputing missing data and detection of outliers in a data set. The Russet data set, that is used as an example, tries to find the relation between political instability of countries and the economical and agricultural inequality, through the collected data of 47 countries on the period after the Second World War (1945-1962). Missing data is imputed to the data set, and later on outliers are obtained.**

I. Introduction

This work about Multivariative Analysis introduces the concepts of missing data, data imputation, and outlier detection. The work is organized as follows:
Section II presents the imputing of missing data problem and discusses some solutions. Between them, the k-NN imputation is the one selected to treat the data.
Section III explains the outlier detection problematic, and presents three different detection methods. Section IV introduces briefly some outlier treatment methods commonly used, while section V presents the code used to impute the data through the k-NN algorithm, and detect outliers with the Mahalobis distance later on. Finally, conclusions are drawn in Section V.

II. Imputing missing data

Firstly, data is imported to the RStudio environment, substituing the missing blank values with an NA value. This is performed by executing in console the following codeline in the RStudio console:
>russet <- read.table("Russetineqdata.txt", header=TRUE, sep="\t")
After importing the Russet data set to RStudio and prompting the data, it is firstly observed that there ara some NA values in the RENT column, corresponding to the percentage of farmers not cultivating their own land missing on the Russet data $.txt$ file. Checked in detail, there are three NA values, corresponding to the contries named Australie, Nicaragua, and Peru. Also, there is an NA value in the column ECKS, that measure the number of conflicts during the period, and that corresponds to the country labeled Norvege. To find which columns (called variables so on) contain missing values, the R function $summary()$ is of great help. Following subsections explain some imputation methods, with special attention to the kNN imputation method used in this work.

A. Simple imputation

The most simple way of imputing data is the simple imputation. The procedure is as follows: the indexs of the missing values of a variable are obtained, and then all of their missing values are filled with a single value, that is obtained through a function that computes this new value by observing the complete cases of that variable. The number imputed will be the same in all the cases. and will depend on the function that is used to compute it. Among others, some examples of simple imputation are:

- *Mean imputation*, when the mean value of the complete cases of the same column is computed, and later imputed in the missing values;
- *Median imputation*, when the median value of the complete cases of the same column is

computed, and later imputed in the missing values.

In our case, using the R codeline:

```
russetmean<-russet
russetmean$Rent[is.na(russetmean$Rent)]<-
mean(russetmean$Rent[!is.na(russetmean$Rent)])
```

the mean of the complete cases of the column RENT is computed to be equal to 21.74, and imputed to the NA values of the Russet data set.

B. Simple linear regression imputation

The linear regression imputation is based on fitting a linear regression model between the desired variable with values to imputate, and the variable with higher correlation with it. Let X be the desired variable with values to imputate. The simple linear regression imputation method is based on the following steps:

- Check the most correlated variable with X.
- Create an indicator variable, that monitor the cases that are complete or incomplete.
- Fit a linear regresion model: The desired column with the most correlated variable.
- For each row with the indicator variable showing incomplete, imput the value obtained with the regression model.

This steps can be easily fullfiled with the usege of the R functions for computing the correlation and a linear regression model of X and Y, $cor(data, use = ”complete.obs”)$ and $lm(x\ y, data = data)$.

C. k-NN imputation

The K Nearest Neighbors (k-NN) algorithm, is a non-parametric method used for classification and regression. Used for classification, it outputs a class membership by the mojority of votes of the k nearest neighbours. In our case study, we are interested in the k-NN regression algorithm for imputation, that otuputs the average of the values of the k nearest neighbours, to be imputed to a missing value.
It is worth mention that RStudio offers packages (VIM and $MDwR$) to import both functions. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. It is considered among the simplest of all machine learning algorithms. To compute the k-NN regression for the $Rent$ variable in the treated Russet data set, the following code could be executed in the RStudio console:

```
>russetkNN <- kNN(russet$Rent,
variable=”Rent”,k=4)
```

It is noticialbe that the k-NN function creates new variables to check if the original columns have missing values. This new variables are erased in this work for easier manipulation of the data set.

D. Chained equations imputation

Multivariate Imputation by Chained Equations (MICE) is based on the approach of imputation by a sequence of small steps, each of which may require diagnostic checking. The process is developed as the following:

- Start filling in the missing data with values at random
- For every variablewith missing values: impute the missing values of th evariable from the predicted values of the regression of the current variable with the remaining ones
- Iterate the above procedure till the convergence

The R code is as simple:

```
>impute_data <- mice(data,m=1)
>imputed_data <- complete(imputed_data)
```

E. Random Forests imputation

The imputation by Random Forests (IRF), is a non parametric method of imputaton, implemented step-by-step as:

- Start filling in the missing data with values at random
- For every variable with missing values: impute the missing values of the variable from the predicted values from the random forest of the individuals with the current variable as response using the remaining ones as predictors.
- Iterate the above procedure till the convergence. When convergence is reached, output the OOB error.

The R code to implement the random Forest algorithm is as following:

```
>impute_data <- missForest(data)
>imputed_data <- imputed_data$ximp
```

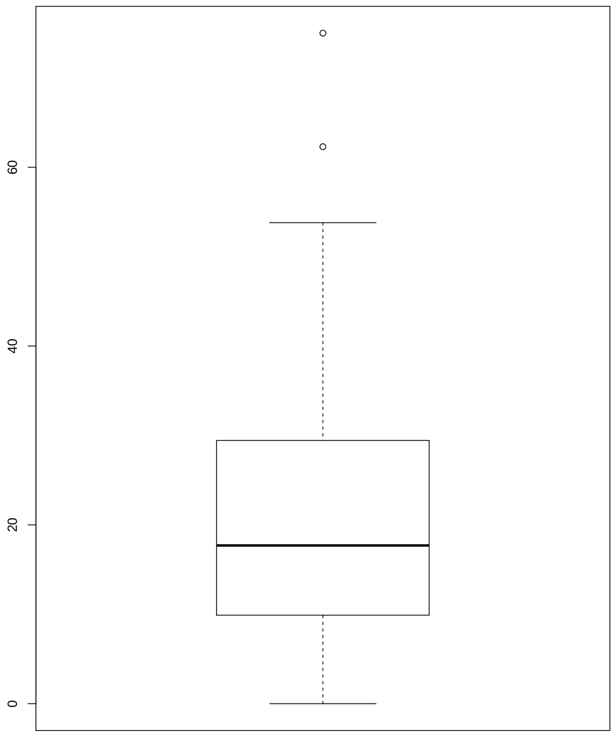

Fig. 1: Boxplot of the RENT variable of the imputed data with the 5NN default algorithm

III. Outliers

A. Detecting outliers

An outlier is an observation which deviates so much from the other observations, and may be considered to be generated by a different mechanism. In fact, outlying data may be or a very unlikely event for a current generating mechanism, or a set of data following a different generating mechanism. With the boxplot function in R, it is possible to graphically distinguish the outliers of a variable.

Considering the Interquartile Range (IQR) as the difference between Q3 and Q1, all the values surpassing the threshold:

$$(Q1 - 1.5 * IQR)$$

could be declared an outlier. If the data follows a normal distribution, $Prob(XQ3 + 1.5IQR) = 0.003488302$.

As a graphic example, some outlying values of the variable RENT, that measures the % of farmers not cultivating their own land, can observed in Figure 1, corresponding to the countries labeled Belgique and Irak. Although, having an outlier in one variable does imply that all that observation of the country is an outlier itself.

B. Detecting outliers with the Mahalanobis distance

As stated before, univariate detection of outliers does not imply multivariate detection. Then, in order to address such problematic, detection of outliers can be based on computing the Mahalanobis distances to the central point of data, by means of an iterative algorithm, involveing the mean of variables (G), and the matrix of variances (V). The Mahalanobis distance accounts for the variance of each variable and the covariance between variables. Geometrically, it does this by transforming the data into standardized uncorrelated data, and computing the ordinary Euclidean distance for the transformed data. In this way, the Mahalanobis distance is like a univariate z-score: it provides a way to measure distances that takes into account the scale of the data. The process to calculate the Mahalanobis distance is as follows:

- *1)* Compute the Mahalanobis distances fo all i points;
- *2)* Rank the Mahalanobis distances and retain the individuals with lower distances;
- *3)* Update G and V till convergence.

After the iterative process is finished, the final ”robustified” Mahalanobis distances with the initial Mahalanobis distances can be plotted to detect outliers.

C. Detecting outliers with Local Outlier Factor (LOF)

Briefly, LOF is an algorithm for identifying outliers within a density-based approach. The algorithm compares the distances of the neighorhood of a point and the maximum distanes of the neighborhood of the neighbors of the points. This process provides an outlying value per individual, and if is greater than 1, the individual is suggested to be an outlier.

IV. Outlier treatment

Once an outlier is detected, several options are there to deal with it. Even a deep outlier treatment scapes of the scope of this homework, some solutions are presented. Among others, outliers could be processed by:

- Eliminating all the outliers (may not be the best solution);
- Weighting the individuals inversely to outlying degree individuals, to diminish their importance;
- Applying a windsorizing approach, and transforming the outliers to the maximum value inside the quartile interval.
- Treating the outliers as missing data.

V. R script to find multivariate outliers and detect them

In order to detect the outliers once the missing data have been imputed, the following R script have been used:

```
>russet<-read.table(”russet.txt”,header=TRUE,
sep=”\t”)
> data1 <- kNN(russet,variable=c(”Rent”,”ecks”),k=
> russetimp <- subset(data1,select=Gini:Death)
>finale_data<-Moutlier(russetimp,quantile=0.975,
plot=TRUE))
```

The resulting ”$robust$” Mahalanobius distance of the Russet data set can be observed in Figure 2.

From the 47 countries observed, at most 25% of them could be assumed to be potential outliers. In our Russet data set case, can be treated as outliers the observations number 1, 5, 9, 11, 20, 22, 24, 30, 36, 41, and 46, that correspond to the countries named in French: Argentine, Bolivie, Colombie, Guatemala, Honduras, Irak, Nicaragua, Philippines, Sud-Vietnam, and Venezuela.

If the country with the highest Mahalanobis distance is checked (Cuba, 4537.44), it is observed that it presents an outstandingly high number of violent conflicts conflicts between 1945 and 1961 (100), and a very high number of deaths during demonstrations between 1950 and 1962 (2900). Then, Cuba is considered to be the outlier country in the data set. We could consider that the time of great tension lived in Cuba in the 50s decade leads the country data to be considered as an outlier. Also, Sud-Vietnam presents huge numbers in that sense (50 and 1000), being the second observation with highest Mahalanobian distance (1563.69).

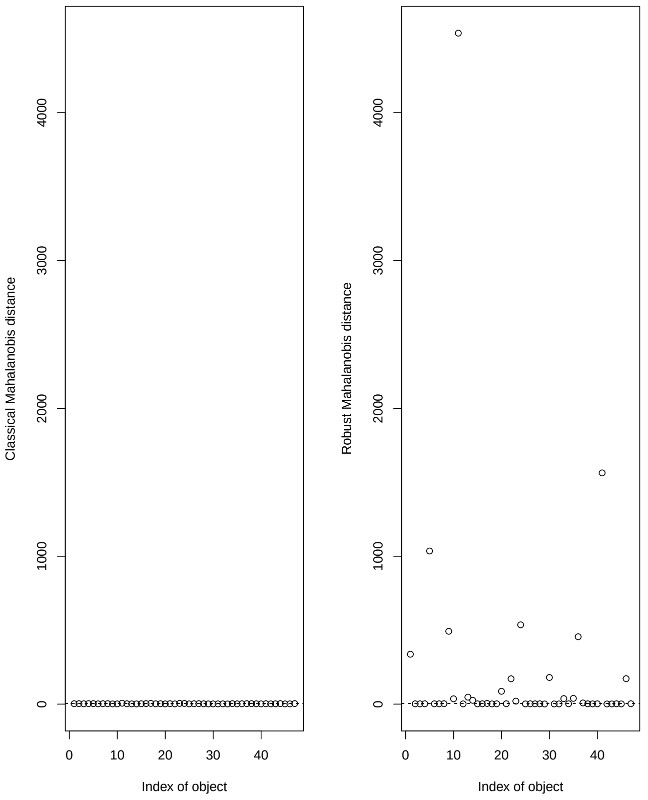

Fig. 2: Mahalanobis distance for the imputed (with k-NN algorithm) Russet data set.

VI. Conclusions

During this work, some imputational methods were defined, includeing simple imputation, kNN algorithm, chained equations and random Forests.

Furthermore, some outlier detection techniques were presented, like graphic boxplots, Mahalanobis distance detecion, or the LOF algorithm.

Besides, a brief R script is presented to detect outliers from a dataset, and some results are discussed to check its correctness, highlighting the most significant outliers.