NATIONAL SCHOOL OF ENGINEERS OF TUNIS

Engineering Internship

Author:
Mayara ELATRACH

Supervisor:
Omar MEJRI

Subject: Building a regional statistics data warehouse on the municipal elections, and the realization of a Dashboard.

Acknowledgements

The internship opportunity I had with Data Expert was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to the MD of Data Expert who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

Contents

List of Figures

Chapter 1

General introduction

1.1 About Data Expert

Data Expert is a start-up founded by an engineer in statistics and information analysis, specialized in the world of data analysis.
Data Expert offers many services such as
Data Design
We help you design your data from several available sources: social sciences, finance and trading
Social Network Analysis
We provide a comprehensive analysis for any Social Network platform: Facebook, Twitter, Youtube
Survey Monitoring
You will be able to monitor your survey and obtain the data and statistics that make you construct data with good quality.
Smart Statistical Softwares
We provide easy solutions for specific statistical problems.
Data Analytics
We can make your data speak. **1**

1.2 Introduction to R

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly ATT, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS. **2**

1.3 The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- An effective data handling and storage facility, a suite of operators for calculations on arrays, in particular matrices **-** A large, coherent, integrated collection of intermediate tools for data analysis **-** Graphical facilities for data analysis and display either on-screen or on hardcopy **-** A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.
The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy. **2**

1.3.1 RStudio

RStudio is a free and open-source integrated development environment (IDE) for R, a programming language for statistical computing and graphics. RStudio was founded by JJ Allaire, creator of the programming language ColdFusion. Hadley Wickham is the Chief Scientist at RStudio.

RStudio is available in two editions: RStudio Desktop, where the program is run locally as a regular desktop application; and RStudio Server, which allows accessing RStudio using a web browser while it is running on a remote Linux server. Prepackaged distributions of RStudio Desktop are available for Windows, macOS, and Linux.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, macOS, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian, Ubuntu, Red Hat Linux, CentOS, openSUSE and SLES).

RStudio is partly written in the C++ programming language and uses the Qt framework for its graphical user interface. The bigger percentage of the code is written in Java, JavaScript is also amongst the languages used.

Work on RStudio started around December 2010, and the first public beta version (v0.92) was officially announced in February 2011. Version 1.0 was released on 1 November 2016. Version 1.1 was released on 9 October 2017.

In April 2018 it was announced RStudio will be providing operational and infrastructure support for Ursa Labs. Ursa Labs will focus on building a new data science runtime powered by Apache Arrow. **3**

1.3.2 R Shiny

Shiny is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge.
During my internship, I used shiny to build an app that helps to better visualize data and sort them however the user wants. **4**

Chapter 2

Data visualization

2.1 Packages used

2.1.1 Shinydashboard

Shinydashboard is a package that creates dashboards with 'Shiny'. This package provides a theme on top of 'Shiny', making it easy to create attractive dashboards.

2.1.2 Htmlwidgets

This package is a framework for creating HTML widgets that render in various contexts including the R console, 'R Markdown' documents, and 'Shiny' web applications.

2.1.3 D3TableFilter

D3TableFilter provides a powerful and flexible HTML table widget for use with the htmlwidgets R library. It allows R users to create HTML tables with functions for advanced filtering and sorting. It also allows for advanced formatting of table cells using D3 functions.

2.1.4 dplyr

A fast, consistent tool for working with data frame like objects, both in memory and out of memory. When working with data you must:

Figure out what you want to do.

Describe those tasks in the form of a computer program.

Execute the program.

The dplyr package makes these steps fast and easy:

By constraining your options, it helps you think about your data manipulation challenges.

It provides simple “verbs”, functions that correspond to the most common data manipulation tasks, to help you translate your thoughts into code.

It uses efficient backends, so you spend less time waiting for the computer.

2.1.5 DT

The R package DT provides an R interface to the JavaScript library DataTables. R data objects (matrices or data frames) can be displayed as tables on HTML pages, and DataTables provides filtering, pagination, sorting, and many other features in the tables. **5**

2.2 The shiny App

I developed a shiny app that allows the user to visualize data in 4 different ways.

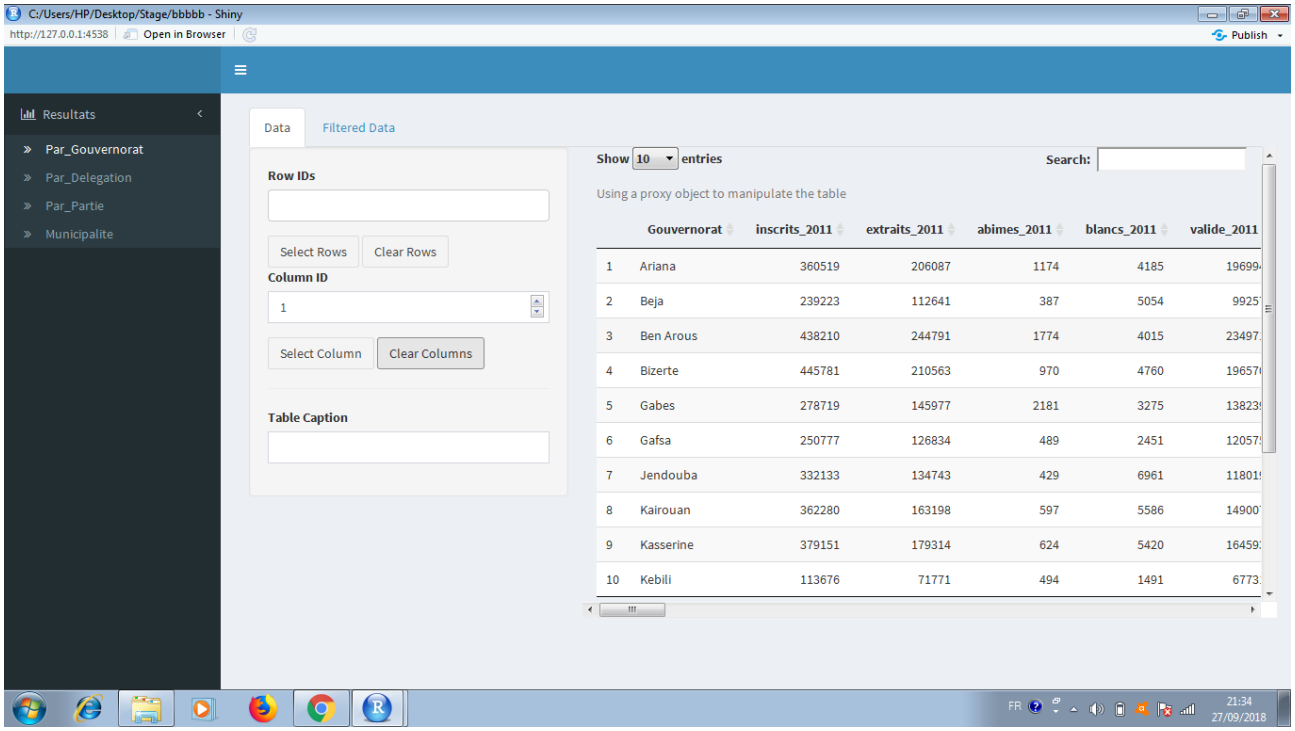

FIGURE 2.1: Data visualization using the shiny app

The user can see the data either by governorate, by delegation, by political party, or by municipality.
The app also gives access to the user to filter data by row and/or by column.

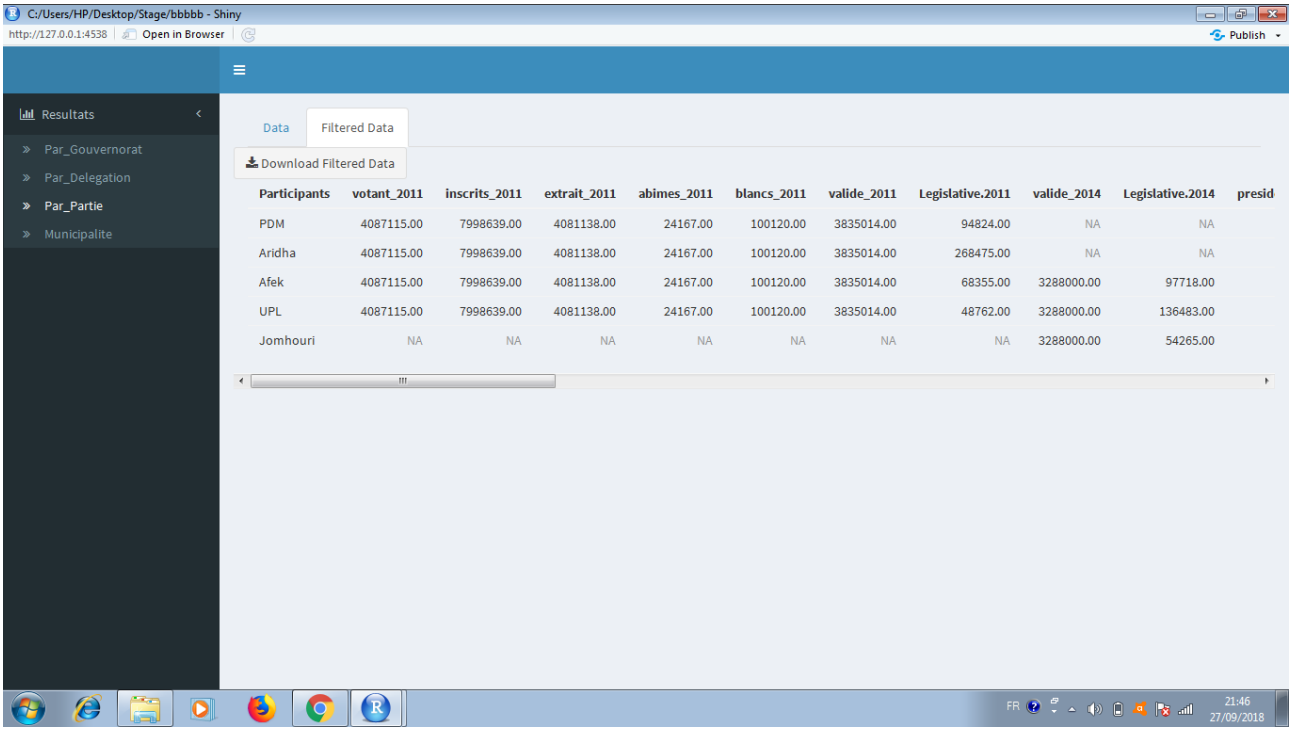

FIGURE 2.2: Filtered data visualization

When the user gets the filtered data they want, they can download the resulted data-set.

Chapter 3

Data Analysis

3.1 Elections of 2011

3.1.1 First Data-set composition

Our data-set is composed of 10 variables representing the elected parties for the 2011 elections. These variables are:
• CPR: Congress for the Republic
• PDP: progress's Democratic Party
• PDM: Democratic modernist pole
• Nahdha
• Takatol
• Afek Tounes
• Aridha
• UPL: Free patriotic union
• Moubadara
• Others
The individuals of the dataset represent the 24 governorates of Tunisia:
• Ariana
• Beja
• Ben Arous
• Bizerte
• Gabes
• Gafsa
• Jendouba
• Kairouan
• Kasserine
• Kebili
• Le Kef
• Mahdia
• Manouba
• Medenine
• Monastir
• Nabeul
• Sfax
• Sidi Bouzid
• Siliana

• Sousse
• Tataouine
• Tozeur
• Tunis
• Zaghouan
The table is filled with the number of votes each party got from the correspondent governorate. For example Takatol got 5937 votes in Kasserine.

3.1.2 PCA analysis

To analyze this dataset I chose the PCA method. Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components.

Before working on the data-set as it is, we need to scale it first. That means to subtract each component of the matrix from it's mean and divide it by it's standard deviation.

$$x_{ij} = \frac{x_{ij} - \bar{x_j}}{s_j}$$

This normalization is used to avoid scaling effects.Otherwise, variables with large variance would dominate the analysis and distort the results. This modification would not change the cloud of dots.
To center and reduce our data-set, I used the R code:

$$> gov11 = scale(gov11)$$

Now we can apply the PCA on the adjusted data-set by using this code:

$$> PCA(gov111)$$

R calculates the eigenvalues which carry the information conserved after the PCA transformation.

```
> PCA(gov111)$eig
        eigenvalue percentage of variance cumulative percentage of variance
comp 1  5.22821598             52.2821598                             52.28216
comp 2  1.95010855             19.5010855                             71.78325
comp 3  1.11486156             11.1486156                             82.93186
comp 4  0.58961153              5.8961153                             88.82798
comp 5  0.51123232              5.1123232                             93.94030
comp 6  0.28917142              2.8917142                             96.83201
comp 7  0.11715022              1.1715022                             98.00352
comp 8  0.08777155              0.8777155                             98.88123
comp 9  0.07978573              0.7978573                             99.67909
comp 10 0.03209115              0.3209115                            100.00000
```

FIGURE 3.1: Eigenvalues table

As we can see from this table, the projection into 2 principal axes conserves 71.78% of the information of the data-set. The first main axis contains 52.28% of the information.
We can also get the cloud of dots of the individuals with the same previous code.

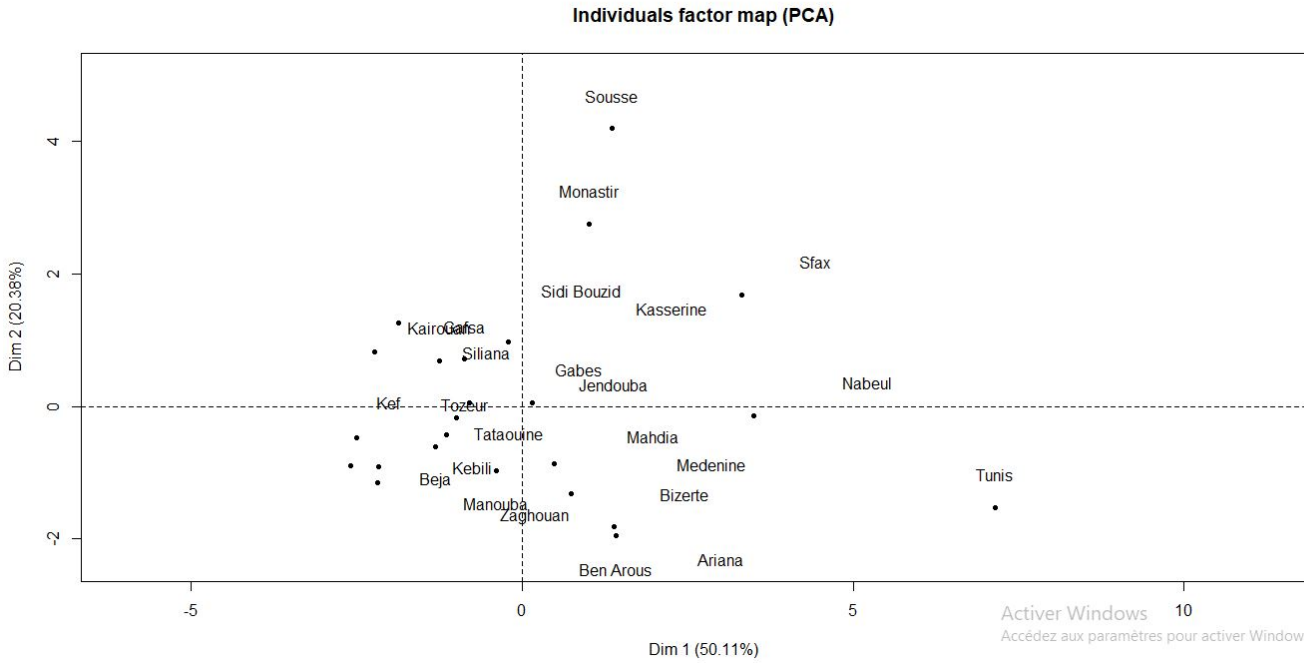

FIGURE 3.2: Cloud of dots of the individuals

Individuals such as Ariana and Ben Arous for example are close to each other meaning that the distance between them is smaller than the average distance between 2 individuals. Thus, those two governorates would behave the same and have the same voting pattern. So if a political party is popular in Ariana it would also be popular in Ben Arous as well.

The same R code allows us to visualize the variable factor map.

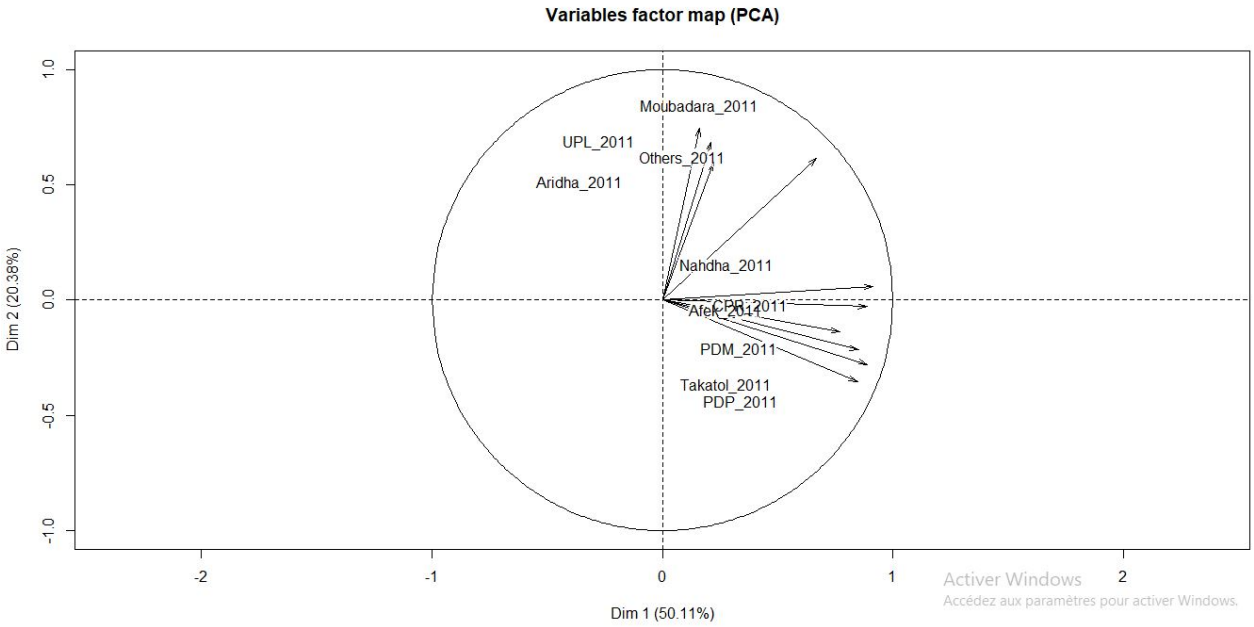

FIGURE 3.3: Circle of correlation between the variables

Variables such as Nahdha, CPR, Afek are more correlated with the first axis.So the first axis explains well their variability. Those variables contribute to the creation of the first axis. Variables such as UPL, Aridha and Moubadra are more correlated with the second axis. The latter explains best those variables. They contribute to the creation of the second axis.
The correlated individuals are parties who can be considered as the same. They would get similar votes from the individuals who are well represented meaning they are close to one of the axis and away from the center.

3.1.3 Second data-set composition

The variables of the data-set are still the 10 elected political parties. The individuals are however the delegations of Tunisia. The matrix contains 264 individuals.
The table is filled with the number of votes each party got in a specific delegation.

3.1.4 Ascendant hierarchical clustering

A cluster is a subset of data which are similar. Clustering (also called unsupervised learning) is the process of dividing a data-set into groups such that the members of each group are as similar as possible to one another, and different groups are as dissimilar as possible from one another. Clustering can uncover previously undetected relationships in a data-set.

Hierarchical clustering involves creating clusters that have a predetermined ordering from top to bottom.

In agglomerative or ascendant clustering method we assign each observation to its own cluster. Then, compute the similarity (e.g., distance) between each of the clusters and join the two most similar clusters. Finally, repeat steps 2 and 3 until there is only a single cluster left. The related algorithm is shown below. **6**

Given:

A set X of objects $\{x_1,...,x_n\}$

A distance function $dist(c_1,c_2)$

for $i = 1$ to n

$c_i = \{x_i\}$

end for

$C = \{c_1,...,c_n\}$

$l = n+1$

while C.size > 1 **do**

- (c_{min1},c_{min2}) = minimum $dist(c_i,c_j)$ for all c_i,c_j in C
- remove c_{min1} and c_{min2} from C
- add $\{c_{min1},c_{min2}\}$ to C
- $l = l + 1$

end while

FIGURE 3.4: Ascendant hierarchical clustering algorithm

To apply this method to the data-set, an R command is used.

$$> h = hclust(dist(d11), method = "ward")$$

After plotting h, we get the dendrogram.

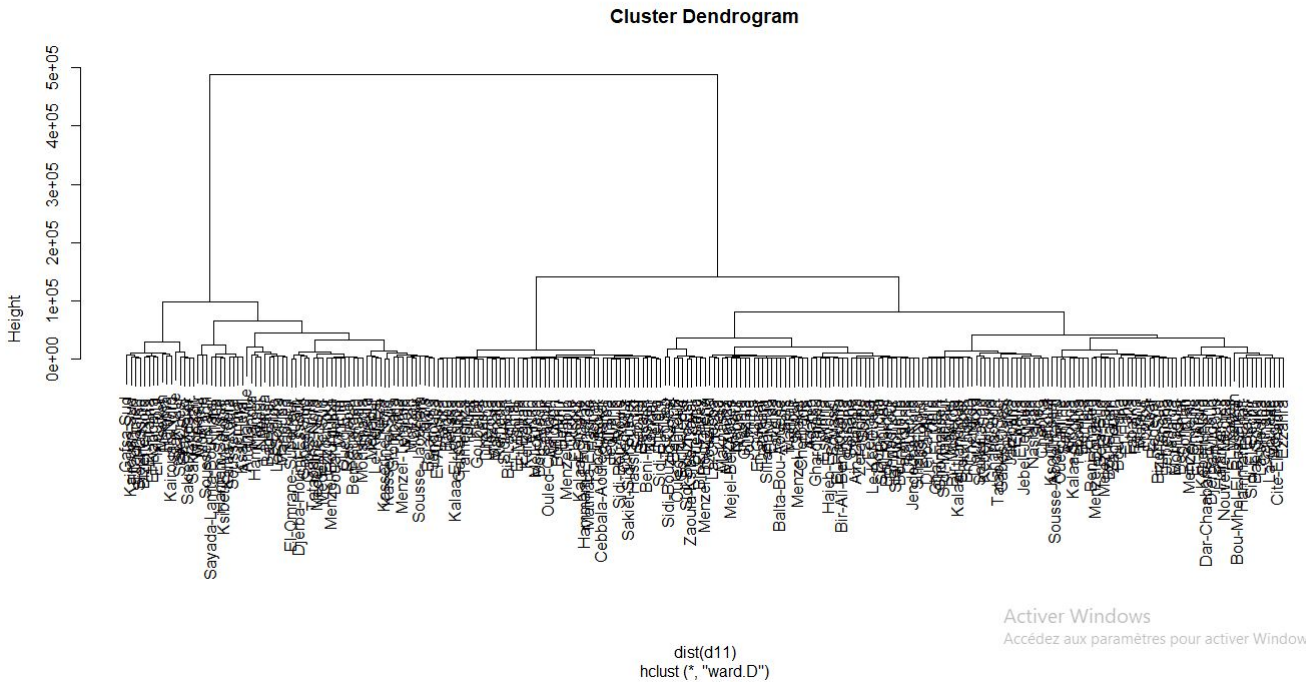

FIGURE 3.5: Cluster Dendrogram

If we choose to cut at the height $1e + 05$ then we can get 3 different clusters. Each cluster represents individuals that behave similarly. The delegations then have similar voting habits.

3.2 Legislative election 2014

3.2.1 First Data-set composition

The data-set is composed of 15 variables. Each variable represents the elected political party.
The individuals are the 24 governorates of Tunisia.
The matrix is then filled with the number of votes each party got from a specific governorate. For example CPR got 2424 votes in Ariana.

3.2.2 PCA analysis

To analyze this data-set I used the same PCA method. After scaling the data, we can visualize the cloud of dots of the individuals.

The projection into 2 principal axes conserves 62.21% of the information of our data-set. the first main axis contains 44.74% of the information while the second contains 17.47%.

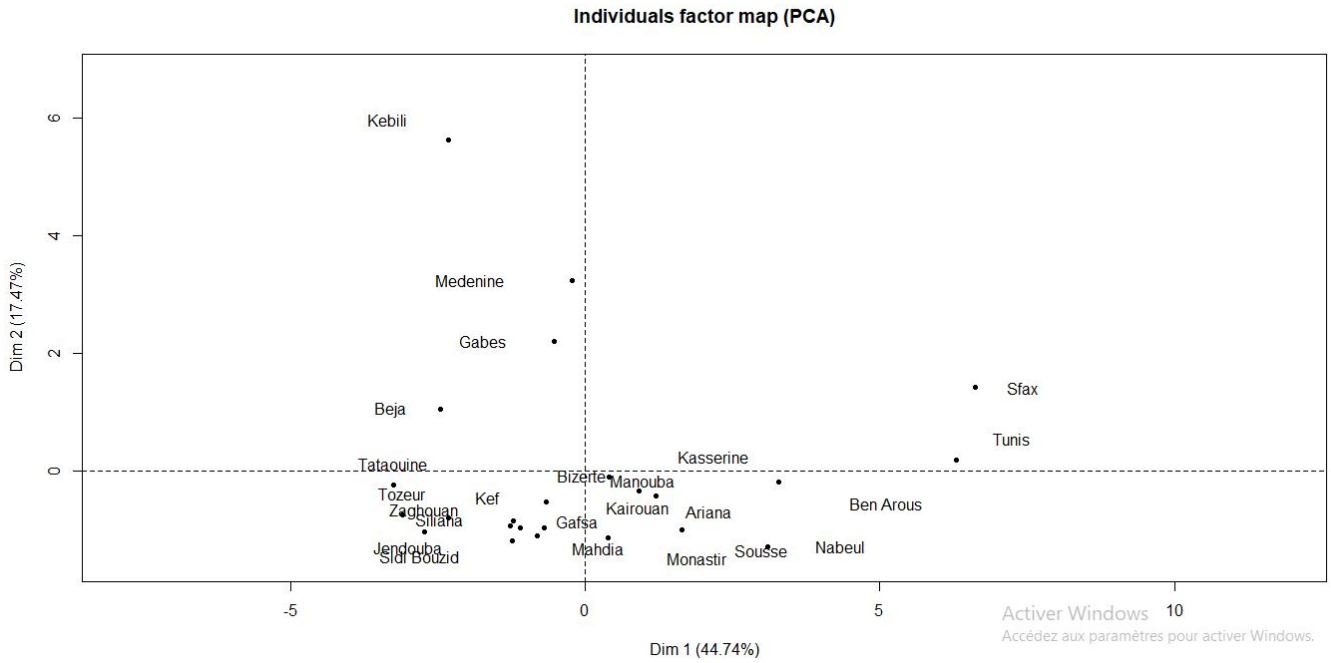

FIGURE 3.6: Cloud of dots between different individuals

Individuals such as Kef and Gafsa for example are close to each other meaning that the distance between them is smaller than the average distance between 2 individuals. Thus, those two governorates would behave the same and have the same voting pattern. So if a political party is popular in Kef it would also be popular in Gafsa as well.
We can also visualize the correlation circle.

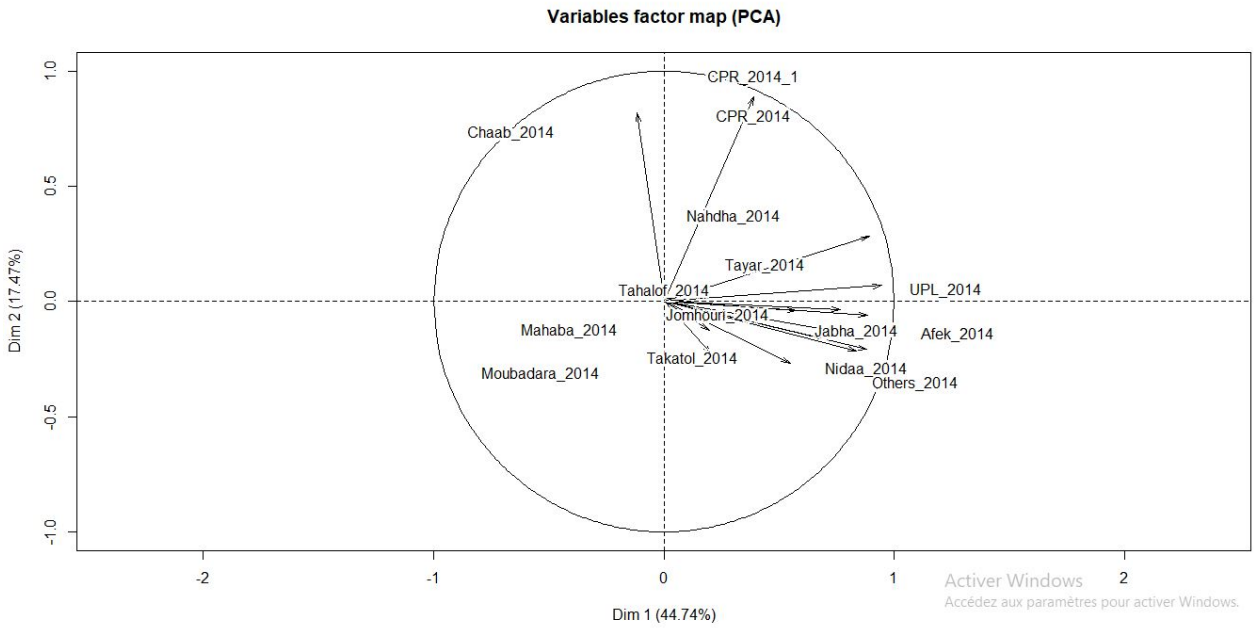

FIGURE 3.7: Circle of correlation between different variables

Variables such as Tahalof, Jomhouri, UPL are more correlated with the first axis.So the first axis explains well their variability. Those variables contribute to the creation of the first axis. Variables such as Chaab, CPR are more correlated with the second axis. The latter explains best those variables. They contribute to the creation of the second axis.
The correlated individuals are parties who can be considered as the same. They would get similar votes from the individuals who are well represented meaning they are close to one of the axis and away from the center.

3.2.3 Second data-set composition

The variables of the data-set are still the 15 elected political parties. The individuals are however the delegations of Tunisia. The matrix contains 264 individuals.
The table is filled with the number of votes each party got in a specific delegation.

3.2.4 Ascendant hierarchical clustering

In this part I applied the same R code as the previous section. By doing so I plotted a dendrogram.

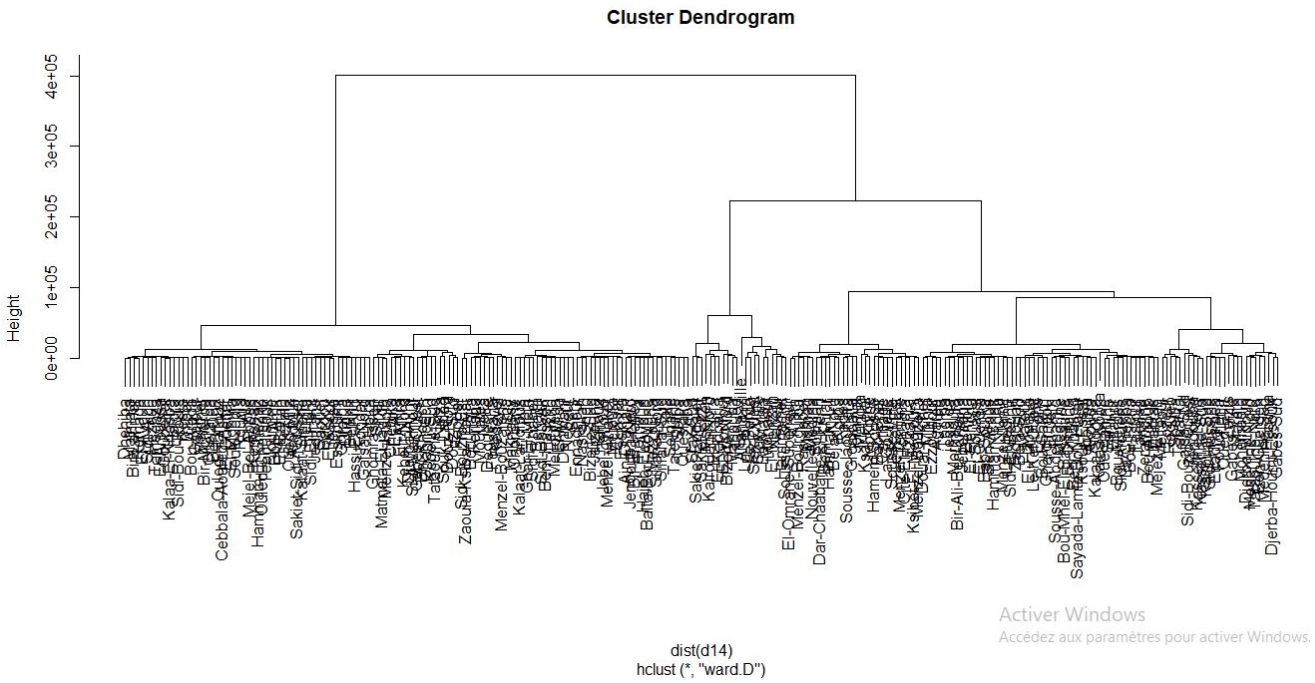

FIGURE 3.8: Cluster Dendrogram

If we choose to cut at the height $1e + 05$ then we can get 3 different clusters. Each cluster represents individuals that behave similarly. The delegations then have similar voting habits.

3.3 Conclusion

During my internship I had the chance to learn more about the programming language R and work with it to arrange the different data-sets. I learned how to manipulate to package Shiny and develop a web app using it.
In addition, I had the chance to apply data analysis methods such as PCA (Principal Component Analysis) and AHC(Ascendant Hierarchical Clustering) to extract information from the data I had.

Bibliography

[1] https://data-expert.net/

[2] https://www.r-project.org/

[3] https://www.rstudio.com/

[4] https://shiny.rstudio.com/

[5] https://www.rdocumentation.org/packages/document/versions/3.0.1

[6] https://www.saedsayad.com/clustering-hierarchical.htm