Applied Data Analysis – Autumn 2017

School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne

Report #01 – Panda and Data Wrangling

Task 1 : Compiling Ebola Data :

We divided the Task into many subtasks to have better visibility and easier understanding of the whole task (1.1 Import Data, 1.2 Indexing Data, 1.3 Data cleaning, 1.4 Calculating means, 1.5 Merging).

1.1 Import Data :

In this part, we imported the data of each country and created a Dataframe for each country separately. We did this following a stackoverflow thread that allowed us to easily concatenate the files one by one.

1.2 Indexing Data :

Here we set the index to the needed fields, to make cleaning easier.
For Liberia and Sierra Leone, we parsed the date into a single type (datetime) as some date were in string and others in datetime. We didn't need to do this for Guinea because all the date were in one single type (string).
(a voir qqch index.is_unique)

1.3 Data Cleaning :

Then we cleaned the Data. First idea: it seems to make sense to replace NaN values by 0's. Then we could easily proceed to merge columns, especially for the Sierra Leone.

1.4 Calculating Means :

In this part we computed the means for the needed fields of every country. We took the fields that represented best new cases (suspects, probable and confirmed) and death (the new registered). The idea was to calculate each mean, then sum the new value for each report over roughly a month and divide the result by the day-span between the first and

last report. For example, if the data begin the 15 June, we took values until 15 July (roughly one month).
However, for December in Liberia, we had some issues between the beginning and the end of the month (there were less death at the end than at the beginning). Furthermore we had good values from June to November for Liberia and that's why in the last Dataframe we have only 0 for the month (December) for this country.
For the Sierra Leone, you can see that we have a "date_death_mean" from the previous month. This is because the values for the death are cumulative. So we needed to subtract by the mean of the previous month to get the value of a given month. However the data seems corrupted too, because they just change month after month. So if we had taken a date and extracted a value from the following day it will be exactly the same as the mean of the month. This is why we took only one day to do our mean of the previous month.

1.5 Merging :

In the end we merged all month of each country then merged all country into one single Dataframe.

Task 2 : RNA Sequences:

The trick here was to merge all the data into one dataframe. To do this we added a column with the index of the csv file (MIDX, the X is for the index of the csv file)

Then when we had all in one Dataframe we needed to import the metadata and merge them together. To do this in a simpler manner, we renamed the column of the metadata Dataframe to have the merge done on columns that have the same name.

Task 3 : Class War in Titanic:

3.1 :

We described the data and categorised those that could be: pclass, survived, sex, sibsp, parch, fare

3.2 :

We first counted the passenger in each class and ploted them into a barchart

Same for the embarkation point, sex of passengers and the group age of passengers. For the last part we cut the data age of passenger and group them to have a good Dataframe for finally plotting them.

3.3 :

We extracted the number of passenger for each floor and added the floor ' S ' to those who didn't have a floor (NA value) to count them too. We then created a Dataframe to visualise them and to plot them in a pie chart.

3.4 :

We extracted the data for each class and all passengers in each class. Secondly, for each class, we subtracted the passengers that survived from all passengers to have only the ones that died. Then we created a Dataframe of the survivors and died individuals and plot them into a pie chart. (3 pie chart because 3 different classes).

3.5 :

For this task, we needed to find the number of passenger that survived on each floor and if they were female or male. We created a Dataframe with the values and plot them into a bar and a pie chart to have the best visibility and understanding of the data.

3.6 :

The final task is a little bit tricky. First we extracted the young from the old (we found that 28 was the middle to create two equally populated age). Then extracted on which floor they were. We then extracted their sex to distinguish the female from the male. We finally divided this number by the number of person that survived to have the fraction that our data represent. At the end, we did a Dataframe with this data to see the percent of each field that survived.

Kevin Kappel 226850

Isaac Leimgruber 236908

Charles Thiebaut 238935