

Rome, 21 February 2020

Visual Analytics – Final Assignment

Group formed by:

Andrea Bellia (1586420)

Manuel Manzara (1876891)

**SARS-CoV-2 EPIDEMIC VISUALIZATION**

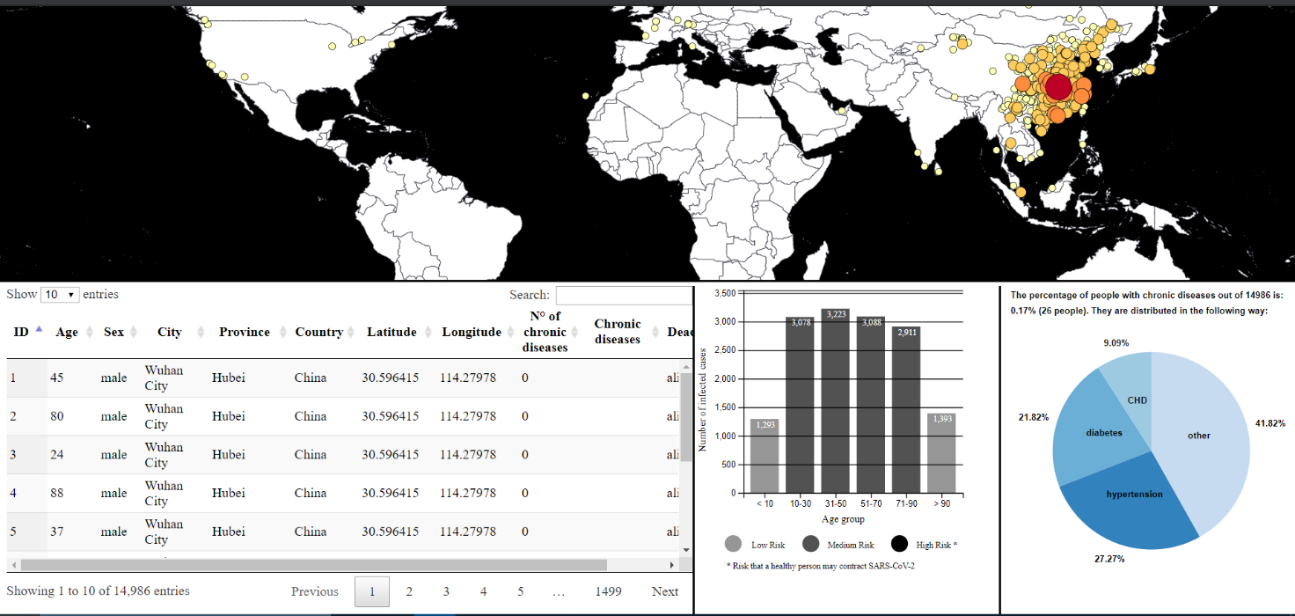
# Abstract

SARS-CoV-2 of 2019-2020, is an epidemic initially identified around the end of December 2019 in the city of Wuhan, the capital of the Chinese province of Hubei, in a group of people with symptoms of pneumonia of unknown cause. Although the methods of transmission of the virus are not yet fully understood, its contagiousness among humans has been confirmed. An infected person can infect another individual even by coughing or sneezing within a radius of about 1-2 meters. Due to this rapid spread of the virus, we decided to create an application that would allow us to understand how the virus is expanding in the world and which information we can rely on to understand if a specific person is at risk of contracting the virus.

Therefore, thanks to "**SARS-CoV-2 EPIDEMIC VISUALIZATION**" we will be able to understand which categories have been classified as most at risk, which and how many cases have been confirmed until now, what is the mortality rate and much more.

# Introduction

The application we developed is mainly composed of 4 views. Each of them will be updated based on user interaction showing specific information about infected people in that specific area of the world. It therefore appears to the user this way.



User interactions with each view will be explained in more detail in the specific view section of the paper. From this first screen, however, we can understand what information we have most focused on to represent the data. In particular, within the first views, which we call "**Map View**", we find the world map in which circles of different size and color will be shown based on the number of cases that have been found in that specific area until the last update. of the dataset. Hovering over each of these circles some information will be shown to the user, and clicking on one of them, all the data shown in the other views of the application will be updated instantly based on the information related to that circle.

In the second half of the screen we see therefore, 3 different views which are respectively named "**Attribute View**", "**Histogram View**" and "**Pie Chart View**". Proceeding in order: the first of these views will have the task of showing the user the information of all the people who have been infected in that area. With the second view we can understand in which age group the victims of the virus for that specific area are classified and passing the mouse some information on the mortality rate will be shown to the user. We also note that the bars have a different color based on the risk that an uninfected person would have to contract the virus. This classification was possible thanks to a pre-processing of the data with the PCA (Principal Component Analysis) method. Finally, the last view represents a pie chart showing which chronic diseases were already present in some of the people infected with the virus and in what percentage. When the user clicks on any slice of the pie, he will be able to understand what is the average age of the victims of the virus who had that chronic disease, since the bar representing that age group will be flashed in the Histogram view.

# Dataset & Pre-processing

Obviously, being the contagion of the virus a very current topic nowadays, it was not easy at all, to find a dataset that had truthful and correctly formed information. So precisely for this reason we had to carry out a preprocessing of data to give input to our application understandable data. However, we took our dataset from the following link:

<https://github.com/beoutbreakprepared/nCoV2019/tree/master/dataset_archive>

Since in this dataset many attributes had N/A values, we wrote a Python program able to correctly filter and format all the records of the csv taken as input. The result of this preprocessing work was a single csv file containing all the information of the victims of the virus both in the province of Hubei and in the rest of the world. Among all the attributes that have been provided in the csv for each victim, we have chosen to focus only on 11 of them, which are the following:

* Dead\_Alive
* ID

* Age
* Sex
* City
* Province
* Country
* Latitude
* Longitude
* Number\_chroDiseases
* Chronic\_diseases

It is also important to stress the large amount of data that our application can manage. In fact, in this first version we have loaded 14.986 entries which for 11 attributes have an AS value equal to 164.846. Precisely for this reason we notice a slight slowdown in the loading of the other 3 views when the application starts.

# Description of the app & Views

Now that we have given a general description of what the application looks like to the user and what its possible interactions are, let's see in more detail what are all the features and information that each view makes available to the user. Let's start with the first view, the Map View, then continuing the analysis for all the others.

# Map View

The Map view is perhaps the most important one as well as the largest in our application, as it allows the user to understand the exact geographic position of all the cases of the SARS-CoV-2 virus that have been found until now. The map can be zoomed by the user and moved to any part of the world he wants to analyze. In addition, 4 important user interactions have been added in this view. In particular, hovering over any country, a label containing the name of the country on which the user is located will be shown and when clicking on any country, the resolution of the map will be adapted so as to show in the foreground only the clicked-on nation.



The other 2 interactions with the user are shown instead when he decides to hover with the mouse on one of the circles highlighted on the map or to click on one of them. In particular, the mouse over the single circle will show the city, province and nation for that area as well as the number of infected cases. While at the click of the user, the other views will be updated showing only the information relating to that number of cases. As concerns the circles showed on the map, they are divided by size and color, based on the number of infected cases in that specific area. We find 5 different shades of orange from the lighter one to almost an intense red which want to indicate the following categories: {< 10 cases, 10-100 cases, 100-500 cases, 500-1000 cases, > 1000 cases}. The different grades have been chosen by the site:

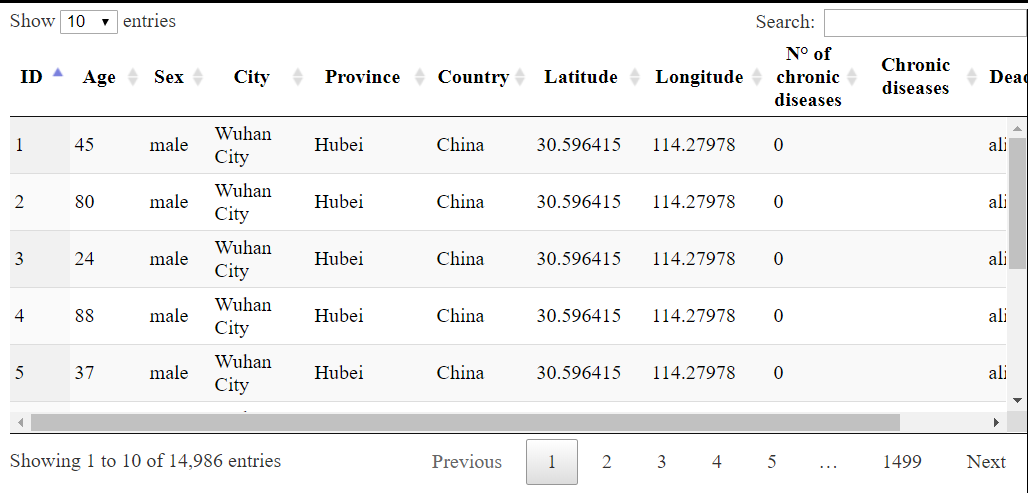
<http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3>

# 

# Attributes View

In this view, we decided to show all 11 attributes for each person inside the circle selected by the user. We note that the table was created in a way that allows the user to choose how many entries to show at one time. In addition, there is a possibility to sort the records of each attribute in ascending or descending order, and to do a research based on any one of those attributes.

## 

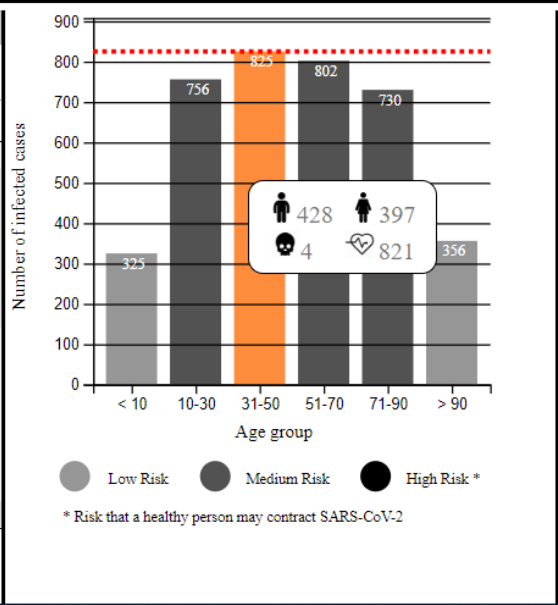


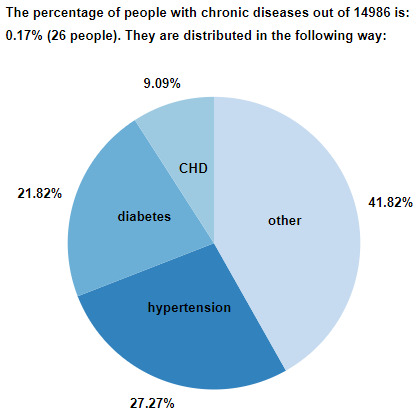
# Histogram View

The Histogram View allows the user to understand in which age group all infected people in the world are grouped and subsequently only those related to the circle selected by the user. In this view 2 types of user interactions have been implemented. In particular we can see that when the user passes the mouse over one of the bars of the histogram, it will change colour to make the user understand that it has been selected, and will show a tooltip that provides information related to what the number of males and females for that age is, as well as the number of people still alive or already deceased. The other interaction with the user will be shown to him when he interacts with the "Pie Chart View". Another very important aspect that was considered in the creation of the histogram was that of dynamically assigning a different colour to the bars according to 3 types of risk: low, medium and high. It has been calculated thanks to the PCA (Principal Component Analysis) method, which will be explained in detail in the following paragraphs, and allows the user to understand what is the risk that a "not yet infected person" would have to contract the virus, divided into bands of age.

# Pie Chart View

The last view is a pie chart made always through the d3.js. Here we have shown what the distribution of chronic diseases was in infected people. Initially, the distribution of the main chronic diseases throughout the world population is loaded (with a frequency greater than 9% otherwise the pie chart would become illegible due to the numerous diseases). Subsequently, upon clicking on one of the circles, the diseases are updated and consequently also their percentages. All diseases with a frequency of less than 9% were channelled into the other slice as they were irrelevant for statistical purposes. Another feature that we have implemented on this chart is the onClick on the cake slices. In fact, by clicking on a specific slice of cake, which will correspond to a disease, it will be possible to see, by flashing on the histogram, which age group the average of people with that disease belongs to. This function has allowed us, for example, to understand that almost all people with chronic diseases are in the age group between 71 and 90 years. An image of the histogram and of the pie chart are below:





# PCA (Principal Component Analysis)

Since our work started from a dataset made up exclusively of infected people, we asked ourselves what the risk was of contracting the virus for a healthy person. To do this we decided to apply PCA to the data that we considered fundamental for the calculation of the risk, i.e. age, gender and number of chronic diseases. In addition to these, there was obviously a need to assign to each infected person, based on the attributes chosen, a value of the risk (from 0 to 2) that person had of contracting the virus when he was still healthy. To assign the risk, we observed the histogram of our application and immediately noticed that, unlike what one might think, children and the elderly are the categories with the lowest risk of contagion, while the other age groups have about the same number of people within them. Another thing that we have always noticed from the histogram is that women and men are affected in equal measure, while as regards chronic diseases (given in our opinion very important and also reported by the news) it is not possible to extrapolate much data as the dataset in our opinion has not been fully filled and therefore many data are missing. We then wrote an additional Python script to build the dataset we needed, ordered according to the risk we assigned, and we input it to another Python program for the analysis of the main components (PCA).

The first operation was the normalization of data via StandardScaler, then through the PCA function of the sklearn library we calculated PCA. Through fit\_transform we made the fitting of parameters on the training set x, and we also obtained a transformed x′. At this point we can see, through the matplotlib library, the graphs of PCA and its clustered version.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

From here we can see how the medium and low risks represent most cases (almost a single cluster), while the high risk is composed of a few very scattered cases as it is highly dependent on the number of chronic diseases of the individual.

# Problems and limitations

During the realization of this project the problems are certainly not missing. Since the whole project was based on a new library that none of us had ever used before, it was a big problem at the beginning for the realization of the map and the relative positioning of the points on it. Another issue was browser compatibility. We noticed some different behaviours between the various browsers, therefore most of the tests were performed on Chrome. Furthermore, another big problem was the structure of the dataset, as many important data were missing, and we were forced to create Python program for better data formatting and in some cases we had to insert random data in order to produce acceptable results.

Finding the right application to PCA was also not easy, as the dataset has very few numerical data that can hardly be clustered for useful purposes. Nonetheless, thanks to the d3 online documentation, we still managed to accomplish almost everything we set out to do.

# Conclusions and future work

In conclusion, we can say that our tool can find multiple applications ranging from pure statistics to medicine. An important note is the fact that in any case you do not have to be an epidemiologist to be able to use and interpret the results of the application as they are very simple, direct and above all visual.

During the development of “SARS-CoV-2 EPIDEMIC VISUALIZATION” we noticed possible improvements that could be implemented in the future to allow the user to collect as much information as possible about the virus. First, let's start by saying that this tool could be used to observe the expansion of any virus in the world, not only for the one we have considered.

It would be interesting to add the possibility of monitoring the expansion of the virus over time, allowing the user to monitor the monthly or annual trend of the disease by creating specific charts.

Another very important aspect could be to add a section in which, given the virus you want to observe, it can be compared with other viruses that have already been analysed in the system previously, trying to find some common aspects that facilitate its study.

Finally, it would be interesting to add the possibility that a user receives a notification when a new csv file with the updated data is available from the server, and this upload could be carried out automatically within our application.

# References

[1] <https://d3js.org/>

[2] <https://jquery.com/>

[3] <https://datatables.net/>

[4] <https://it.wikipedia.org/wiki/Coronavirus>

[5] [http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3](http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3 )

[6] <https://bl.ocks.org/d3noob/96b74d0bd6d11427dd797892551a103c>

[7] <https://bl.ocks.org/farazshuja/e2cb52828c080ba85da5458e2304a61f>

[8] <https://bl.ocks.org/andybarefoot/765c937c8599ef540e1e0b394ca89dc5>

[9] <https://github.com/beoutbreakprepared/nCoV2019/tree/master/dataset_archive>

[10] <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>