Data Wrangling

The main objective of the project is to capture the survival rate based on the geographical location and gender for every year occurred due to cancer disease. Also to infer the type of the cancer that caused the deaths and the rate of death for each cancer type. I started by searching the data set and visualizations currently available for cancer and found that there is too much of data available but visualized poorly. The visualization had too many parameters to comprehend by a lay man. There are a lot of data variable available for a study on cancer. I wanted to start with the geographic interactive visualization having variables based on year and gender. I want to expand this project in a more granular way by categorizing each cancer, expenses incurred for the treatment, mortality rate based on the stage at which cancer was diagnosed, the number of successful diagnosis of breast cancer due to initiatives of the government such as the Obama Care.

# **Data searching on the web.**

Google searched for the cancer survival data and found multiple data sources clearly explaining the cancer survival for 5 years. And found the very relevant data under American cancer society for **Breast cancer** please see the data sources under (<https://www.cancer.org/cancer/breast-cancer/understanding-a-breast-cancer-diagnosis/breast-cancer-survival-rates.html)>

Diving further inside this link found many insightful data and one which was very interest to my project is the survival rate under the link <https://cancerstatisticscenter.cancer.org/#/>

Later selected the 5 year survival rate data from the link: <https://cancerstatisticscenter.cancer.org/module/cNgHqCms> and downloaded the relevant excel file which is placed under my github location for reference: <https://github.com/Narendrakumarg1728/Data_analysis_and_visualization/tree/master/Final_project/data>

Data wrangling was easy as I had done similar data wrangling on similar data on a huge data set using xrld package on the WHO data which is explained in details in later section [ICD-10.](#_ICD10-_data.)

Also fond very useful info on the mortality rate of cancer under “**GLOBOCAN”** project which is cancer project research by World Health Organization on 2012 data set. Please see the data source under <http://globocan.iarc.fr/old/FactSheets/cancers/breast-new.asp>

Found huge data set containing mortality data set of the whole world with clear indication of cause of the death under the WHO site pointing the data of International Classification of data 2016. [Mortality datasetby WHO](http://www.who.int/healthinfo/statistics/mortality_rawdata/en/)

# Data Processing.

I have used python package **xlrd** for more details on xlrd please see <http://xlrd.readthedocs.io/en/latest/> to know how to use xlrd package.

Data processing in xlrd is very easy and used the methods like sheet\_by\_name and rows to process the data.

The downloaded data was in the excel format with name **“SurvivalByStage.xlsx”** which for processing the data have placed it under “../data/” folder and the python code **“virtualize.py”** for the project is placed under “../Final\_project/”

Later divided the data from the excel to the “**list”** data structure format having separate list for “**Cancer type**”, “**Survival Rate**”, **Stages:** “Stage1”, “Stage2”, “Stage3” and “Stage4”, **“New Cases**” and “**Deaths”** for further plotting of the data in alphabetical order and for classification of the data on the graph.

The difficult part to process the data was due to the large size of the excel data from the ICD1-0 data. Processing the data on the python used to take more than 24 hours to complete. Please see the section [ICD10](#_Inputs_and_Output) for more details.

The ICD-10 data is compressed and placed under **“../data/Morticd10\_part1.zip”** asthe datasize was 126MB which is difficult to move it to github. Please un compress the file before executing the script.

# Visualization

Used python **Bokeh** **package** for visualization, please see the bokeh project documentation <http://bokeh.pydata.org/en/latest/> for complete documentation.

In Bokeh package used the **circle graph** for visualization, which takes x and y axis data and plots the data on the graph, it looks similar to the bar charts when we first look into it but it’s much more useful providing us more options to visualize data clearly for example we can have multiple data points on a single y axis rather than having a single data as in the bar graph.

Another example is the use of the radius of the circle to be used visualize the variation in the data.

Also the hover tool in the circle graph can be used present many other data set which will help to visualize the data clearly, for example we have used the hover tool to reflect whether the data set is a new cancer incident or if the data representing is a death incidence. And used the hover tool to differentiate and visualize the different stages of the cancer.

To color differentiate the plot based on the survival rate, had to provide the color coding for each data plotted this feature was not available in circle graph, which has a single colour for all the x and y axis in a graph. To hack this had to plot multiple data on the graph based on the stages and for few had to pass the color of the plots based on the survival rate data.

# **ICD10- data.**

## **Inputs and Output for the ICD-10 processing**

1. Google search for “International classification of diseases WHO mortality data”
2. You will be able to find the link of World Health Organization website with mortality data of all the diseases from ICD [International Classification of Diseases].
3. Through web scraping as described in the below section, fetch all the links that have downloadable file by running the webscraping.py file. This file also has other website links that were found during the search.
4. From the links and the information obtained after web scraping, the files Morticd10\_part1 and Morticd10\_part2 have been identified as relevant to the project.
5. For illustration purpose, download the data of ICD10 from the page whose link is ICD-10
6. Unzip the downloaded file and specify the file format as csv or xlsx or xls by renaming.
7. Run the datawrangling.py file to obtain the required data in a dictionary format as described below in the data wrangling section.
8. All the above mentioned code and data files as well as the outputs have been uploaded in my GitHub link.
9. Other interesting data set found is from [Kaggle Wisconsin website](https://www.kaggle.com/uciml/breast-cancer-wisconsin-data) related to breast cancer.

Web Scraping

Used Beautiful soup library for web scraping. Few of the websites have a downloadable link for the data required for the project. The web page has numerous links but since the interest is only with the links that have downloadable, using beautiful soup I am scraping all the links in the web page which have downloadable files. I am able to achieve this by first identifying all the anchor tags and in turn checking if it belongs to “link\_media” class and then scrape the link. Along with the link I am scraping the info of the downloadable file i.e the metadata of the file in the output.

Data Wrangling

The World Health Organization [WHO] releases mortality data of all the diseases across the world. The downloaded data through web scraping results in multiple huge excel files with a lot more data than required for the project. Since the project is based on the cancer disease, this necessitates me to traverse these huge files and scrape only cancer data. The data that can be captured is the year, country, number of deaths, number of male deaths and number of female deaths for each country and year. Below steps have been taken to capture this data:

1. Open the excel using xlrd library
2. Traverse through each row in the sheet
3. Read the cause from each row and move forward only if the cause is Cancer else go to next row
4. Read Country, year, the total number of deaths, number of deaths for male and female
5. The country is listed in codes in the datasheet and hence needs to be converted to country name,  which is listed in another datasheet called the country\_codes.
6. A new function is defined to get the country name by passing the country code as the parameter  to the function.
7. Country\_code file is opened using the xlrd library again and traversed through each row.
8. Compare the country code received and the one in the row, if they are the same return country  name.
9. Once all the required data is captured in the row in the required format, the dictionary data  structure is constructed.
10. Since I am planning to present the mortality data for each country for each year, am using a three  level dictionary as my data structure. The primary key is the year, secondary key is the country and the keys at the third level are the total number of deaths, male deaths and the female deaths. To achieve this I have used the defaultdict from the collections library. The output is this data structure.

Problems faced

1. The downloaded file has 2 million rows of data and it was recommended not to open in excel, the recommended tool was “STATA”, which is a paid software.
2. Hence I went ahead and opened it in excel to capture this huge data, as a result my script datawrangling.py takes 12 hours to run successfully and give me the required data in the dictionary data structure.
3. Am planning to use the SQL database to upload and manipulate the data for visualization to overcome this problem.