PA 1: Exploratory Analysis over Covid19 Dataset

Student Details

Student Name and ID:

Notes: When submitting, fill your name and ID in this cell. Note that this is a markdown cell! Do not make any changes in the dataset file and do not rename any '.csv' files. Rename your submission file to **'yourLastName_Last4digitsofyourID_PA1.ipynb'** . Do not to forget to cite any external sources used by you.

- [-5 points] if this note/rule is not abided.

Assignment Details

In this assignment, you will conduct a guided exploration over covid 19 dataset. You will learn and use some of the most common exploration/aggregation/descriptive operations. This should also help you learn most of the key functionalities in Pandas.

You will also learn how to use visualization libraries to identify patterns in data that will help in your further data analysis. You will also explore most popular chart types and how to use different libraries and styles to make your visualizations more attractive.

Dataset Details

In this assignment, you will work on Covid 19 dataset. Specifically, you will work on covid.csv attached file with this project. The file covid.csv contains 35,156 rows and 10 columns. This dataset begins 01/22/2020, and runs upto 7/27/2020. It includes day to day country wise no. of cases which has County/State/Province level data. The columns of the data-set are:

- Date - The day on which cases have been reported / recorded.
- Country/Region - The country where these cases have been recorded.
- Confirmed Deaths - The no. of deaths
- Recovered - The no. of recovered cases.
- Active - The no. of active cases.
- New cases - The no. of new cases.
- New deaths - The no. of new deaths.
- New recovered - The no. of new recovered cases.
- WHO Region - WHO operated regions.

Required Python Packages

You will use the packages imported below in this assignment. Do NOT import any new packages without confirming with the TA.

In [20]:

```
# special IPython command to prepare the notebook for matplotlib
%matplotlib inline

#Array processing
import numpy as np
#Data analysis, wrangling and common exploratory operations
import pandas as pd
from pandas import Series, DataFrame

#For visualization. Matplotlib for basic viz and seaborn for more stylish figures
import matplotlib.pyplot as plt
import seaborn as sns
```

Reading Dataset

The Python code below reads the Federal Emergencies and Disasters dataset into a Pandas data frame with the name df_data. For this code to work, the file 'database.csv' must be in the same folder as this file.

In [21]:

```
#2.5 points for both questions below.

#read the csv file into a Pandas data frame
print (">>Task a: The csv file into a Pandas data frame: \n", None )

#return the last 5 rows of the dataset
print (">>Task b: The last 5 rows of the dataset are: \n", None )
```

```
>>Task a: The csv file into a Pandas data frame:
 None
>>Task b: The last 5 rows of the dataset are:
 None
```

Task 1: Statistical Exploratory Data Analysis

Let us start with getting know the dataset. Your first task will be to get some basic information by using Pandas features. For each task below, look for a Pandas function to do the task. Replace None in each task with your code.

In [22]:

```
# 2.5 points
#Task 1-a: Print the details of the data frame (information such as number of rows,colum
ns, name of columns, etc)
print (">>Task 1-a: Details of data frame are: \n", None )
```

```
>>Task 1-a: Details of data frame are:
 None
```

In [23]:

```python
#2.5 points
#Task 1-b: Find the number of rows and columns in the data frame.
num_rows = None
num_cols = None
print ("\n\n>>Task 1-b: Number of rows:%s and number of columns:%s \n" % (num_rows, num_cols))
```

```
>>Task 1-b: Number of rows:None and number of columns:None
```

In [24]:

```python
#7.5 points
#Task 1-c: Print the total cases confirmed, deaths, and recovered cases of each country with the given dataset.
#The below variable holds data for all the unique countries, and their total - confirmed, deaths, and recovered cases.
total_confirmed_deaths_recovered = None
print ("\n\n>>Task 1-c: Total cases confirmed, deaths, and recovered are \n", total_confirmed_deaths_recovered)
```

```
>>Task 1-c: Total cases confirmed, deaths, and recovered are
 None
```

In [25]:

```python
# 5 points for both questions below,

#Task 1-d-i: Print top 10 worst affected countries with confirmed cases.
top_confirmed_ten_countries  = None
print ("\n\n >>Task 1-d: \n", None)

#Task 1-d-ii: Print top 5 worst affected countries with death cases.
top_death_five_countries  = None
print ("\n\n >>Task 1-d: \n", None)
```

```
 >>Task 1-d:
 None


 >>Task 1-d:
 None
```

Task 2: Aggregation & Filtering & Rank

In this task, we will perform some very high level aggregation and filtering operations. Then, we will apply ranking on the results for some tasks. Pandas has a convenient and powerful syntax for aggregation, filtering, and ranking. DO NOT write a for loop. Pandas has built-in functions for all tasks.

In [26]:

```
# 8 points
#Task 2-a: Find out the countries that has had more than a total of 2 hundred thousand confirmed cases.
confirmed_greater_than_200K = None
print (">>Task 2-a:  The countries that has had more than a total of 2 hundred thousand confirmed cases are: \n%s" % (confirmed_greater_than_200K))

# 8 points
#Task 2-b: Find out the total number of confirmed, recovered, and death cases for each WHO region.
total_cases_whoregion = None
print ("\n\n>>Task 2-b: The total number of confirmed, recovered, and death cases for each WHO region are: \n%s" % (total_cases_whoregion))

# 7 points
#Task 2-c: Find out the top 5 poorly performing countries, in the order of confirmed, deaths, and recovered cases.
top5_poorlyperforming=None
print ("\n\n>>Task 2-c: The top 5 poorly performing countries, in the order of confirmed, deaths, and recovered cases.s are: \n%s" % (top5_poorlyperforming))

# 7 points
#Task 2-d: Find out the top 5 poorly performing WHO regions, in the order of confirmed, deaths, and recovered cases.
top5_disasters=None
print ("\n\n>>Task 2-d: The top 5 poorly performing WHO regions, in the order of confirmed, deaths, and recovered cases are: \n%s" % (top5_disasters))
```

```
>>Task 2-a:  The countries that has had more than a total of 2 hundred thousand confirmed cases are:
None


>>Task 2-b: The total number of confirmed, recovered, and death cases for each WHO region are:
None


>>Task 2-c: The top 5 poorly performing countries, in the order of confirmed, deaths, and recovered cases.s are:
None


>>Task 2-d: The top 5 poorly performing WHO regions, in the order of confirmed, deaths, and recovered cases are:
None
```

Task 3: Visualization

In this task, you will perform a number of visualization tasks to get some intuition about the data. Visualization is a key component of exploration. You can choose to use either Matplotlib or Seaborn for plotting. The default figures generated from Matplotlib might look a bit ugly. So you might want to try Seaborn to get better figures. Seaborn has a variety of styles. Feel free to experiment with them and choose the one you like. We have earmarked 10 points for the aesthetics of your visualizations.

In [27]:

```python
sns.set_style('whitegrid')
sns.set(font_scale = 1.3)

# 15 points
# Task 3-a: Plot the graph for the top 10 poorly performing countries, over 7 months of data provided
# Think of a way to nicely visualize all the countries.
#########################begin code for Task 3-a
##########################end code for Task 3-a

# 15 points
# Task 3-b: Plot a pie-chart for the top 3 poorly performing WHO regions, over 7 months of data provided
#########################begin code for Task 3-b
##########################end code for Task 3-b
```

Task 4: Interesting Information.

Find out an 'interesting' information from the dataset. Create a visualization for it and explain in a few lines your reasoning.

This task is worth 10 points. Your result will be judged based on the uniqueness and quality of your work (having a meaningful result and an aesthetic visualization).

In [31]:

```python
#########################begin code for Task 4

##########################end code for Task 4
```

Task 5: WEKA

You have to use WEKA,

- This task is worth 10 points.
- This task is different from the above tasks attached.
- You have to work on disaster.csv dataset attached in the same directory as the file.
- Find out how to convert the used dataset in weka format
- Convert it
- use the weka visualizer to produce some graphs.
- Write down observations, on the images you are going to attach with the submission.
- You must attach atleast 5 different observations alongwith your submission.

In this dataset, you will work on 63 years of Federal Disasters dataset. The file database.csv contains 46,184 rows and 14 columns. This dataset begins with the year 1953, and runs up to the year 2017. Each row corresponds to an emergency declared by the president due to a natural disaster all around the US. The columns of the data-set are:

- Declaration Number - Unique number for each emergency declared
- Declaration Type - Type of declaration
- Declaration Date - Date of declaration
- State - State affected
- County - County affected
- Disaster Type
- Disaster Title
- Start Date - The date event started
- End Date - The date event ended
- Close Date - End of Declaration
- Individual Assistance Program - Whether IAP was provided or not?
- Individuals & Households Program - Whether IHP was provided or not?
- Public Assistance Program - Whether PAP was provided or not?
- Hazard Mitigation Program - Whether HMP was provided or not?

In [30]:

```
#########################begin code for Task 5
# Goto weka tools and open disaster.csv file in arff viewer.
# If you find any problems in the dataset, you can pre-process data to avoid any such case as per your obersvations,
# and mention it here.
# and finally you can save it as arff file
# Write down observations, on the images you are going to attach with the submission.


### For submission, the arff file has to be submitted alongwith 5 graphical observations in a folder with extensions .png, .jpg
### or can provide a compiled set of images in a word or pdf.
#########################end code for Task 5
```