COVID-19: FINAL PROJECT REPORT

Presented in the report, our team took extensive steps of collecting, analyzing & comparing/contrasting multiple data types to offer the best insight possible on the condition of COVID-19 in Texas, USA

Bipin Lama,
Hemant Sharma,
Bryce Stuber,
Pablo Khadka

Data Analysis

For the purpose of this project, our team chose to combine two different datasets in regard to the current pandemic caused by Codiv-19, with the state of Texas as the center of our focus. To carefully analyze the data reported for the hospital capacity in Texas we compared data sets **Texas Hospital Capacity vs. US-States.**

Upon analysis of the data sets we came to the conclusion that there's not much in common between the datasets, except for the numbers reported on both datasets based on their **similar dates**. Thus, we chose to compare/contrast the data on the basis of numbers reported by date. Though we understand that there aren't many similarities between the datasets, we believe the comparisons we have presented in our scatterplot charts will shine a light on the condition of Covid-19 in Texas as compared to other states.

Data Cleaning & Merging

For the code itself, our team chose to use Pandas as our choice of collecting, cleaning & merging the data. To start off the code, our team chose to **import Pandas & matplotlib.pyplot** for graphing the dataset charts.

From there we downloaded the CSV files & store them into the same folder to set the file path. Upon importing the CSV files into the code, we chose to check each file's data types & the first/last dataset values.

While running the code, we ran into few issues due to missing data such as blank values in Total_Beds_Available & ICU_Beds_Occupied in the Texas_Hospital_Capacity file, To improve the data & avoid any errors due to missing data, we decided to use the .**dropna function**.

Additionally, to make sure the dataset being used is proper, we chose to use **.head & .tail** functions to check the type of data being printed from each file as presented in part of the code below.

```
#Import Pandas along with the mathplotting functions
import pandas as pd
import matplotlib.pyplot

#Import, define & read the csv file path for the first dataset.
df= pd.read_csv(r"/home/isqsdac/Desktop/workspace/final_project/data_covid/Texas_Hospital_Capacity.csv")
#Prints the general info about characteristics in file.
df.info()

#Counts & prints the first 20 rows of the file for us to analyze the type of data.
df.head(20)

#Handles & drops any rows with missing values.
```

```
print("------HANDLE MISSING VALUES--------")
df.dropna(inplace=True)
df.info()

#Import & read the csv file path for the second dataset.
df1= pd.read_csv(r"/home/isqsdac/Desktop/workspace/final_project/data_covid/US-States.csv")
#Prints the ending 20 rows of the file characteristics.
df1.tail(20)
df1.info()
```

For the purpose of merging the two datasets & presenting values only for the state of Texas, our team chose to use the **.merge** function, and we merged the data on the similar data type of **dates with an inner join**. Once the data has been merged, if there were still any missing values which may have been included upon the merge, we used the **.dropna** function again. Furthermore, to view what the merged dataset types will look like, we used the **.info function**, so that prior to the plotting of the graphs, we have an understanding of the data to compare as presented in the next part of the code below.

```
#Prints the beginning statistics for the state of Texas from the second file being used
#for the data merge.
df1.head()
df1_texas = df1[df1['state']=='Texas']
df1_texas

#Merges the data in the two files & prints the HEADER for each rows' data.
print("Code to merge the two datasets for a comprehensible scatterplot chart.")

#Code will be merged on the basis of dates provided in both datasets with an inner join.
df_date_merge=pd.merge(df, df1_texas, on = "date", how = "inner")
df_date_merge.dropna()

#Presents us with an insight on the data type upon dataset merging.
df_date_merge.info()
```

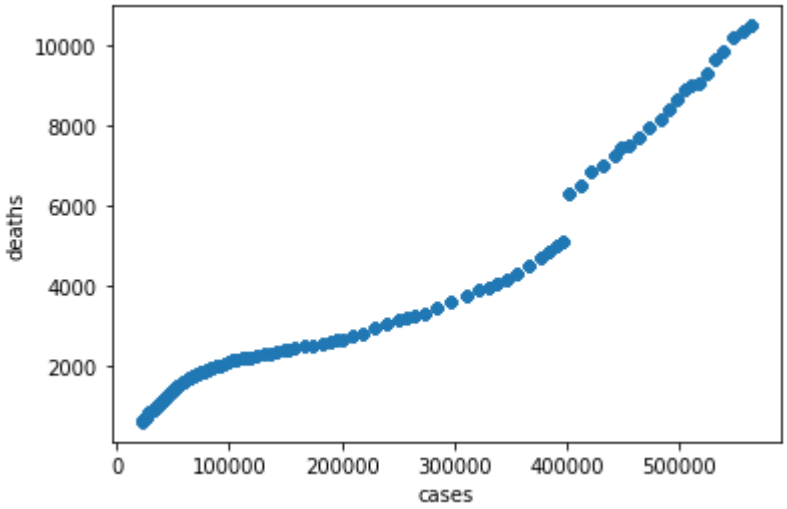
Visualizations

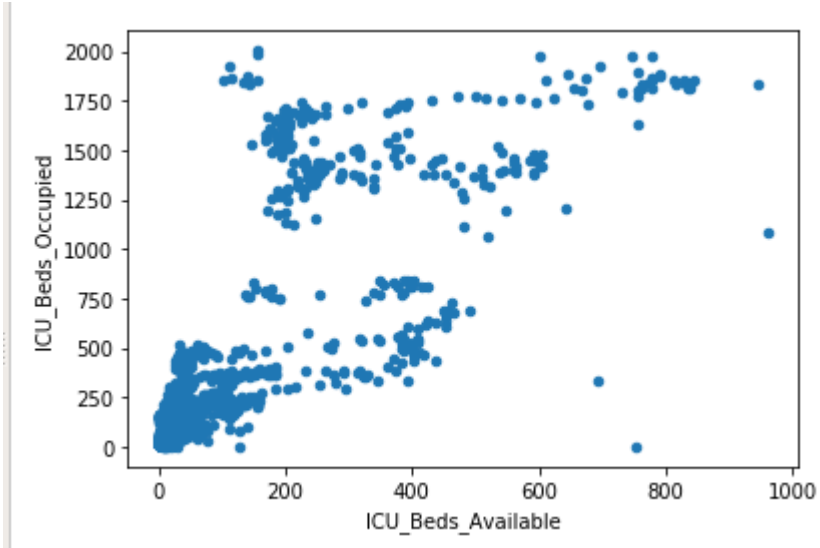
For presenting the visualizations within pandas, we used the scatterplot graphs. As presented in the code below, once we merged our data, we used the **.plot.scatter function** to show the correlation between relevant Covid-19 data such as “Cases vs. Deaths”, “Total_Occupied_Beds vs. Total_Available_Beds” & “ICU_Beds_Available vs. ICU_Beds_Occupied”. Attached below are the sample of image outputs generated by our code.

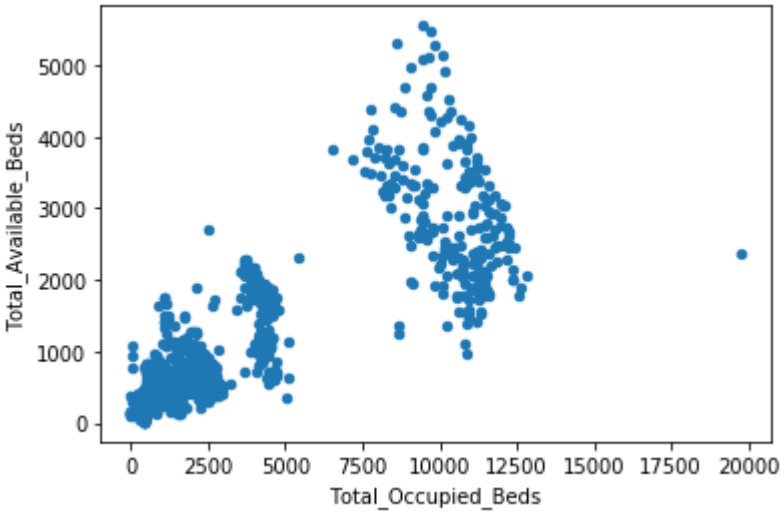
#Presents a positively correlating scatterplot between #of cases vs. #of deaths.
df_date_merge.plot.scatter(x='cases',y='deaths')

#Presents an unconventional scatterplot with regards to Total Beds Occupied vs. Total Beds Available.
df_date_merge.plot.scatter(x='Total_Occupied_Beds',y='Total_Available_Beds')

#Presents an unconventional scatterplot with regards to ICU Beds Available vs. ICU Beds Occupied.
df_date_merge.plot.scatter(x='ICU_Beds_Available',y='ICU_Beds_Occupied')







Flow-diagram