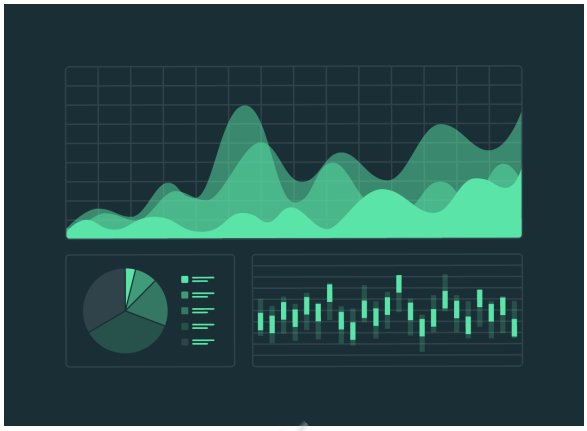
PRO-C112

Capstone class : Data Story -1



What is our GOAL for this MODULE?

The goal of this module is to learn about data visualization.

What did we ACHIEVE in the class TODAY?

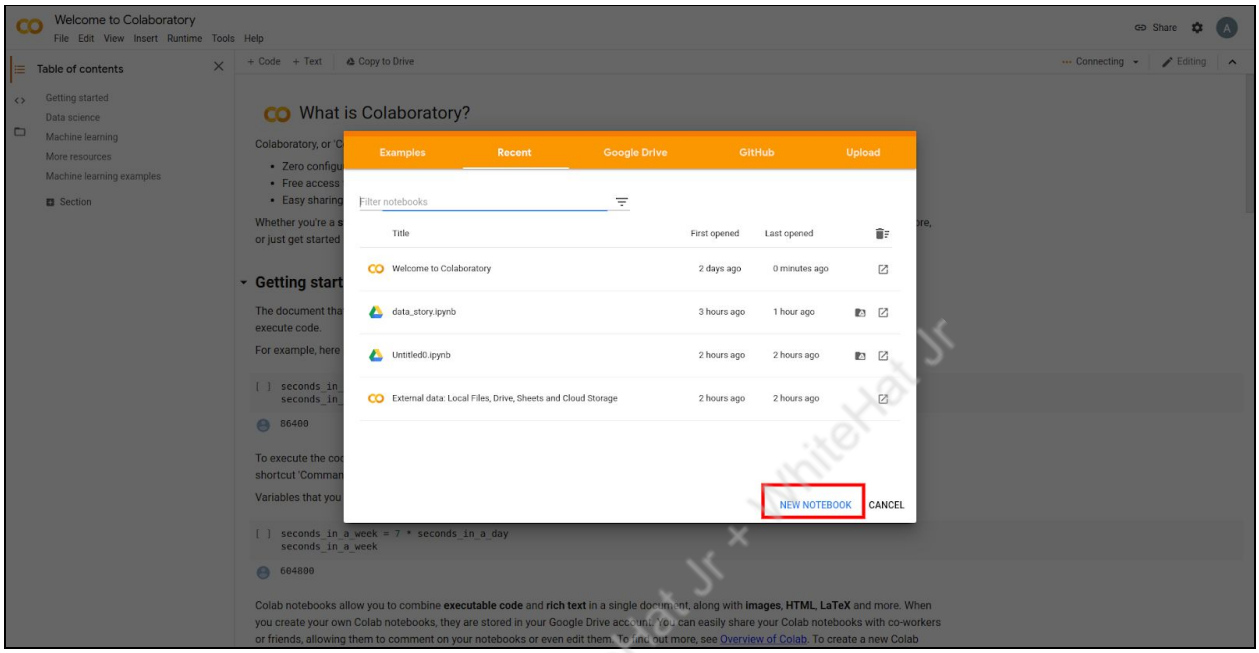
We started to create a data story from data of people who were reminded to save money and people who weren't reminded to save money to create a narrative to help convey the meaning of the data.

Which CONCEPTS/CODING BLOCKS did we cover today?

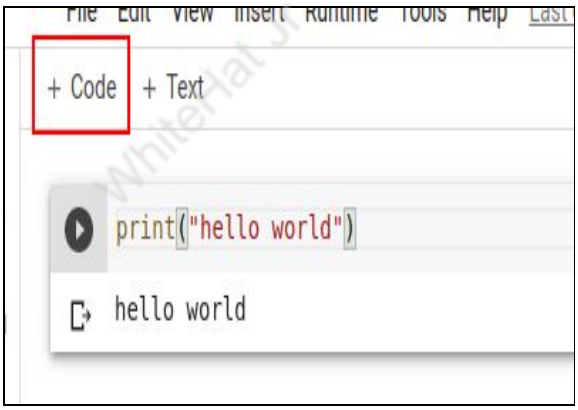
- Revised mean, median, mode, and standard deviation.
- Revised finding correlation and plotting graphs.
- Google colab

How did we DO the activities?

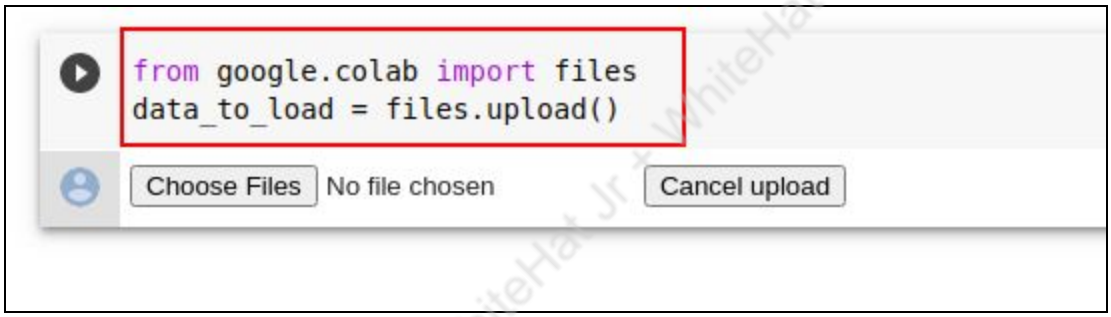
1. We learned about the usage of google colab.



- Learned to write code and text in the colab.



- Saw how to upload files on colab.



- We also saw how to plot a graph on colab.

2. We started to write the data story.
3. We imported the pandas, statistics and plotly.express libraries.

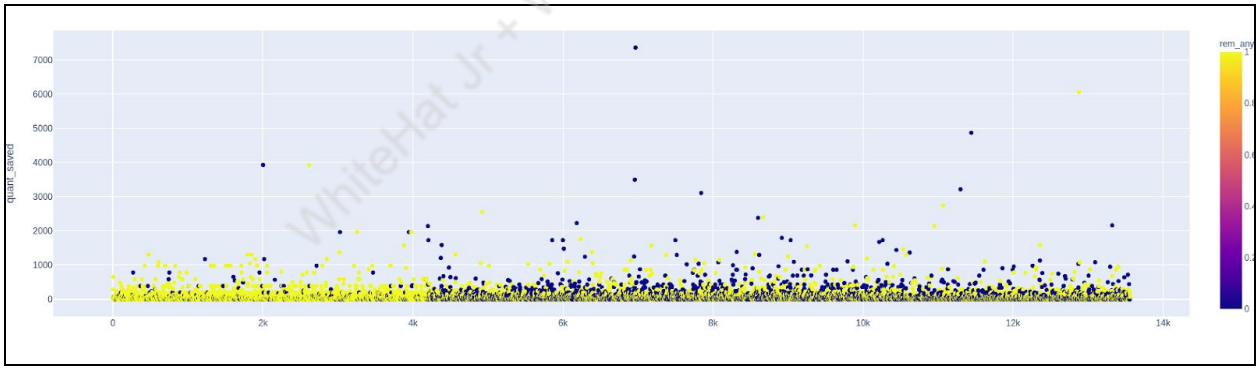
```
[ ] #Importing the important modules
```

```
[ ] import pandas as pd
    import statistics
    import plotly.express as px
```

4. We then uploaded the data file and plotted it on the scatter plot.

```
[ ] #Uploading the csv
    from google.colab import files
    data_to_load = files.upload()

    #Plotting the graph
    df = pd.read_csv("savings_data_final.csv")
    fig = px.scatter(df, y="quant_saved", color="rem_any")
    fig.show()
```



5. We calculated and plotted a graph with the number of people who were reminded and who weren't.

```
import csv

with open('savings_data_final.csv', newline="") as f:
  reader = csv.reader(f)
  savings_data = list(reader)

savings_data.pop(0)

#Finding total number of people and number of people who were reminded
total_entries = len(savings_data)
total_people_given_reminder = 0
for data in savings_data:
  if int(data[3]) == 1:
    total_people_given_reminder += 1

import plotly.graph_objects as go

fig = go.Figure(go.Bar(x=["Reminded", "Not Reminded"], y=[total_people_given_reminder, (total_entries - total_people_given_reminder)]))

fig.show()
```

6. Then we found the mean, median and mode of the entire savings data.

```
[ ] #Mean, median and mode of savings
    all_savings = []
    for data in savings_data:
      all_savings.append(float(data[0]))

    print(f"Mean of savings - {statistics.mean(all_savings)}")
    print(f"Median of savings - {statistics.median(all_savings)}")
    print(f"Mode of savings - {statistics.mode(all_savings)}")
```

```
Mean of savings - 85.32780331328739
Median of savings - 39.2724
Mode of savings - 0.0
```

7. Then we found the mean, median and mode of the data of people who were reminded and who weren't.

data_story.ipynb

File Edit View Insert Runtime Tools Help Last saved at 14:39

+ Code + Text

```
[ ] #Mean, median and mode of savings
    reminded_savings = []
    not_reminded_savings = []
    for data in savings_data:
      if int(data[3]) == 1:
        reminded_savings.append(float(data[0]))
      else:
        not_reminded_savings.append(float(data[0]))

    print("Results for people who were reminded to save")
    print(f"Mean of savings - {statistics.mean(reminded_savings)}")
    print(f"Median of savings - {statistics.median(reminded_savings)}")
    print(f"Mode of savings - {statistics.mode(reminded_savings)}")
    #To add new lines
    print("\n\n")
    print("Results for people who were not reminded to save")
    print(f"Mean of savings - {statistics.mean(not_reminded_savings)}")
    print(f"Median of savings - {statistics.median(not_reminded_savings)}")
    print(f"Mode of savings - {statistics.mode(not_reminded_savings)}")
```

```
Results for people who were reminded to save
Mean of savings - 82.56940908263547
Median of savings - 39.2724
Mode of savings - 0.0


Results for people who were not reminded to save
Mean of savings - 89.67095185226523
Median of savings - 36.0475
Mode of savings - 0.0
```

Again, there seems to be massive differences between the mean, median and modes of savings of people who were both reminded and not reminded. We can conclude this with the same explaination as above.

Let's first calculate the standard deviation of the data.

```
[ ] #Standard Deviation
    print(f"Standard deviation of all the data -> {statistics.stdev(all_savings)}")
```

8. Then we calculated the standard deviation of all those data.

```
[ ] #Standard Deviation
    print(f"Standard deviation of all the data -> {statistics.stdev(all_savings)}")
    print(f"Standard deviation of people who were reminded -> {statistics.stdev(reminded_savings)}")
    print(f"Standard deviation of people who were not reminded -> {statistics.stdev(not_reminded_savings)}")

    Standard deviation of all the data -> 196.75453011909315
    Standard deviation of people who were reminded -> 173.24866414440817
    Standard deviation of people who were not reminded -> 228.875050299707
```

9. Then we found the correlation between age and the saved money.

```
[ ] import numpy as np

    age = []
    savings = []
    for data in savings_data:
      if float(data[5]) != 0:
        age.append(float(data[5]))
        savings.append(float(data[0]))

    correlation = np.corrcoef(age, savings)
    print(f"Correlation between the age of the person and their savings is - {correlation[0,1]}")

    Correlation between the age of the person and their savings is - 0.03663447975985462
```

We concluded that the data is not correlated and mean of the data are also significantly far.

What's NEXT?

In the next class, we will learn more about the IQR and find the z score of the data.

EXTEND YOUR KNOWLEDGE

You can experiment with other data from https://www.kaggle.com/ and calculate the mean, median and mode to find correlation.