Lab assignment 5 Florentin Dutruel 5011507

September 19, 2022

[1]:

```
import pandas as pd
import seaborn as sns
import plotly.express as px
import numpy as np

import matplotlib.pyplot as plt
```

[2]:

```
import plotly.io as pio
pio.renderers.default = "plotly_mimetype+notebook"
```

1 Matplotlib

For this excercise, we have written the following code to load the stock dataset built into plotly express.

[3]:

```
stocks = px.data.stocks()
stocks.head()
```

[3]:

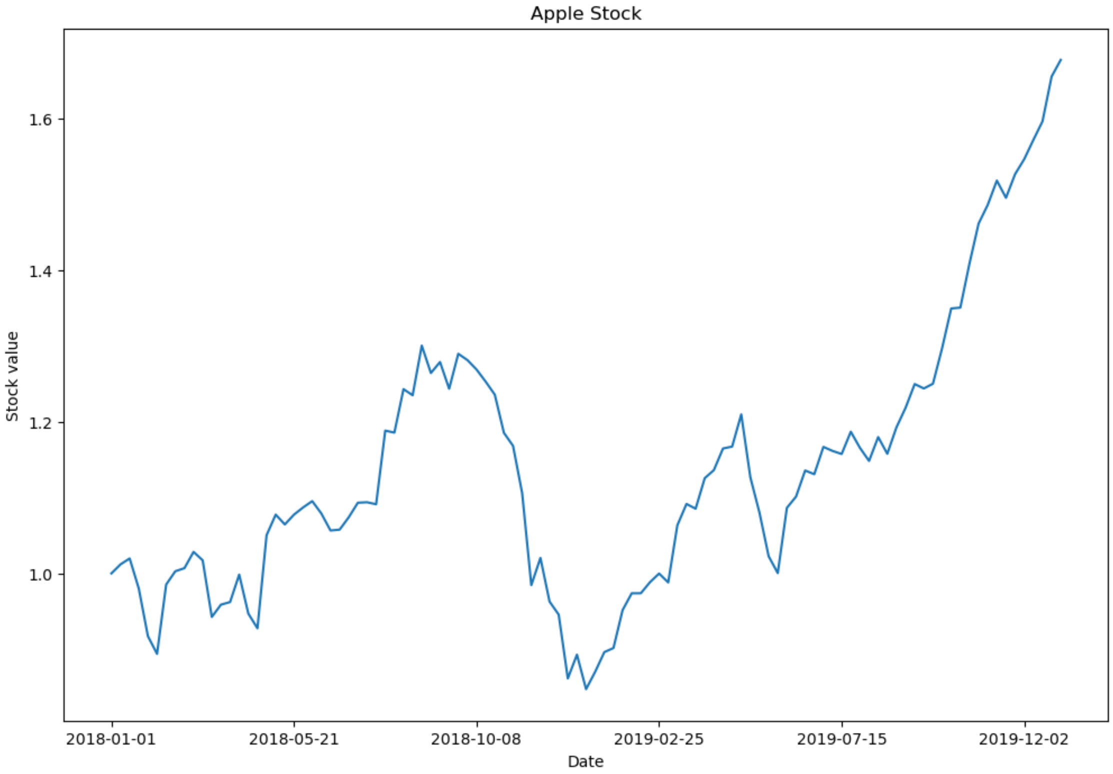
```
         date      GOOG      AAPL      AMZN        FB      NFLX      MSFT
0  2018-01-01  1.000000  1.000000  1.000000  1.000000  1.000000  1.000000
1  2018-01-08  1.018172  1.011943  1.061881  0.959968  1.053526  1.015988
2  2018-01-15  1.032008  1.019771  1.053240  0.970243  1.049860  1.020524
3  2018-01-22  1.066783  0.980057  1.140676  1.016858  1.307681  1.066561
4  2018-01-29  1.008773  0.917143  1.163374  1.018357  1.273537  1.040708
```

1.1 Question 1:

Select a stock and create a suitable plot for it. Make sure the plot is readable with relevant information, such as date, values.

[4]:

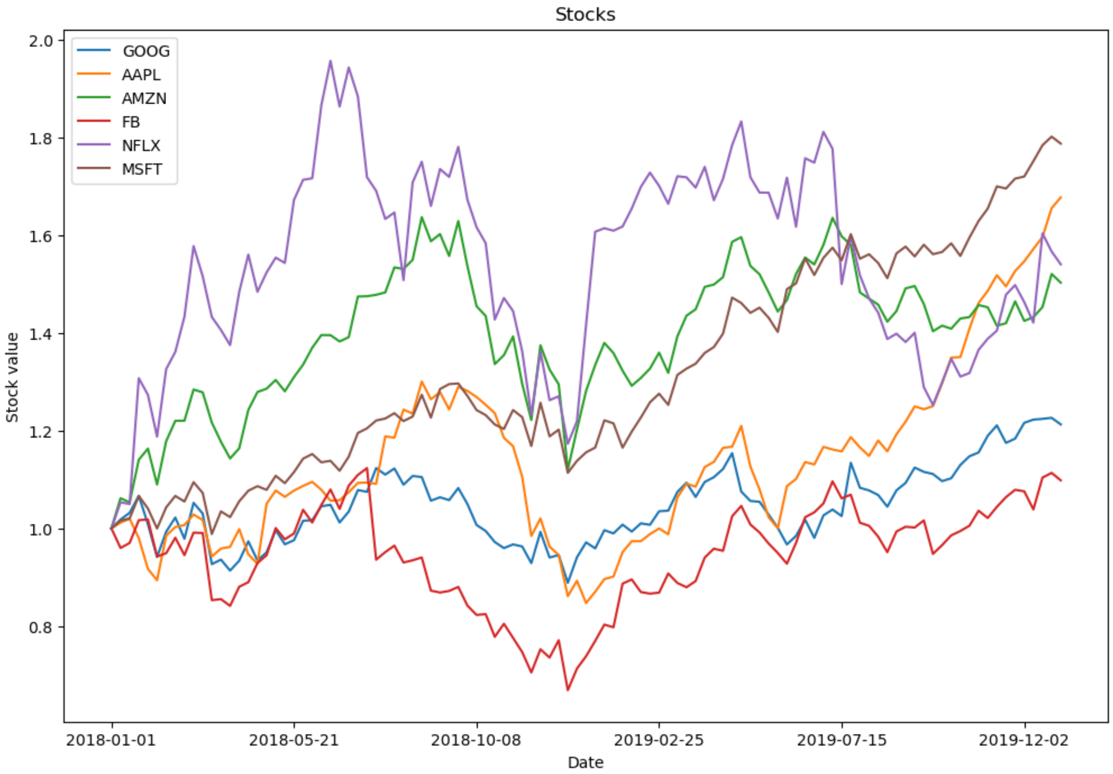
```
# YOUR CODE HERE
plt.figure(figsize=(12,8))
plt.plot(stocks["date"], stocks["AAPL"])
plt.xlabel("Date")
plt.ylabel("Stock value")
plt.title("Apple Stock")
plt.xticks(stocks["date"].array[::20]);
```



1.2 Question 2:

You've already plot data from one stock. It is possible to plot multiples of them to support comparison.
To highlight different lines, customise line styles, markers, colors and include a legend to the plot.

[5]:
```
# YOUR CODE HERE
plt.figure(figsize=(12,8))
plt.plot(stocks["date"], stocks["GOOG"], label="GOOG")
plt.plot(stocks["date"], stocks["AAPL"], label="AAPL")
plt.plot(stocks["date"], stocks["AMZN"], label="AMZN")
plt.plot(stocks["date"], stocks["FB"], label="FB")
plt.plot(stocks["date"], stocks["NFLX"], label="NFLX")
plt.plot(stocks["date"], stocks["MSFT"], label="MSFT")
plt.xlabel("Date")
plt.ylabel("Stock value")
plt.title("Stocks")
plt.legend()
plt.xticks(stocks["date"].array[::20]);
```



2 Seaborn

First, load the tips dataset

[6]:
```
tips = sns.load_dataset('tips')
tips.head()
```

[6]:
```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

2.1 Question 3:

Let's explore this dataset. Pose a question and create a plot that support drawing answers for your question.

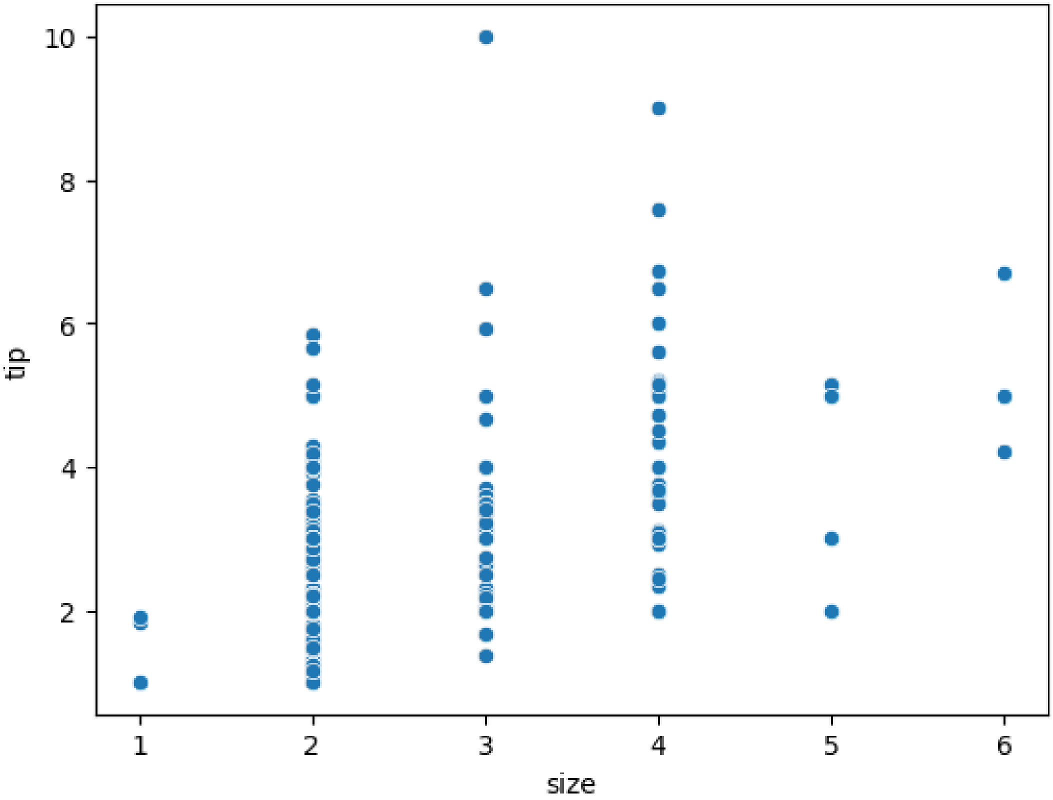
Some possible questions: - Are there differences between male and female when it comes to giving tips? - What attribute correlate the most with tip?

[7]:

```
# YOUR CODE HERE
# Do group size have a positive correlation with tips?
sns.scatterplot(data=tips, x="size", y="tip")
corr = tips["size"].corr(tips["tip"])
print(f"Correlation between group size and tips are {corr:.2f}")

# Conclusion there is a possitive correlation between the group size and tip␣
 ↪size but it's rather weak 0.5. Preferably we need more data about larger␣
 ↪group sizes.
```

```
Correlation between group size and tips are 0.49
```



3 Plotly Express

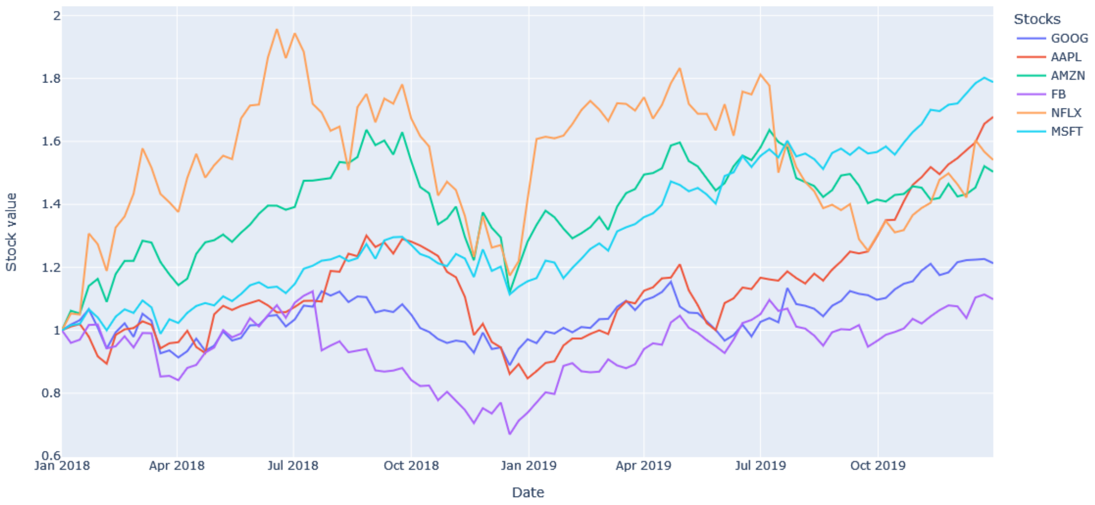
3.1 Question 4:

Redo the above exercises (challenges 2 & 3) with plotly express. Create diagrams which you can interact with.

3.1.1 The stocks dataset

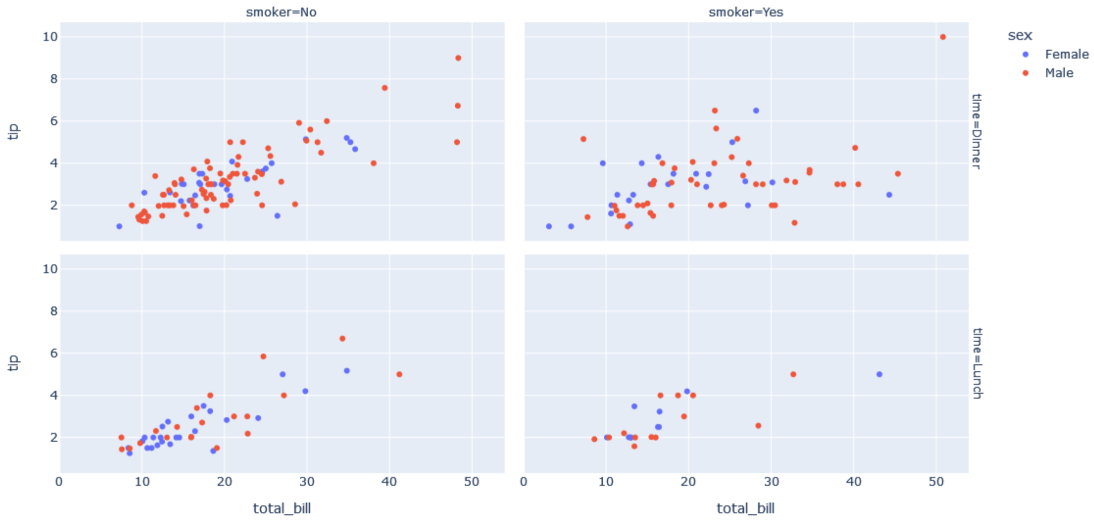
Hints: - Turn stocks dataframe into a structure that can be picked up easily with plotly express

[8]:
```
# YOUR CODE HERE
px.line(stocks, x="date", y=["GOOG", "AAPL", "AMZN", "FB", "NFLX", "MSFT"],␣
↪labels={"date": "Date", "value": "Stock value", "variable": "Stocks"},␣
↪height=600)
```



3.1.2 The tips dataset

[9]:
```
# YOUR CODE HERE
px.scatter(tips, x="total_bill", y="tip", color="sex", facet_col="smoker",␣
↪facet_row="time", height=600)
```



3.2 Question 5:

Recreate the barplot below that shows the population of different continents for the year 2007.

Hints:

- Extract the 2007 year data from the dataframe. You have to process the data accordingly
- use plotly bar
- Add different colors for different continents
- Sort the order of the continent for the visualisation. Use axis layout setting
- Add text to each bar that represents the population

[10]:

```
#load data
df = px.data.gapminder()
df.head()
```

[10]:

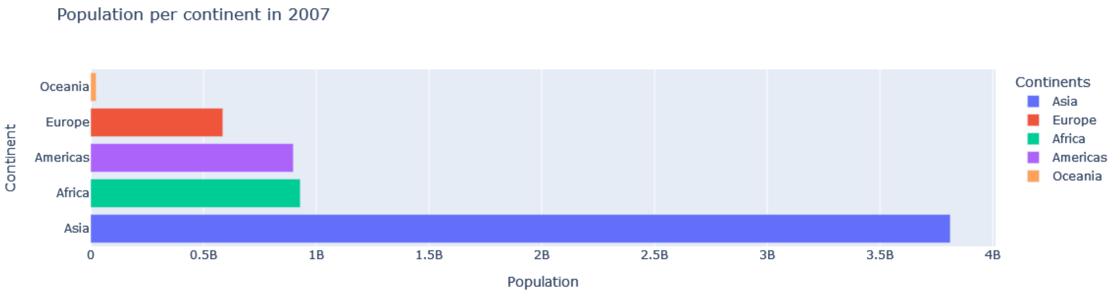
```
       country continent  year  lifeExp       pop   gdpPercap iso_alpha  \
0  Afghanistan      Asia  1952   28.801   8425333  779.445314       AFG
1  Afghanistan      Asia  1957   30.332   9240934  820.853030       AFG
2  Afghanistan      Asia  1962   31.997  10267083  853.100710       AFG
3  Afghanistan      Asia  1967   34.020  11537966  836.197138       AFG
4  Afghanistan      Asia  1972   36.088  13079460  739.981106       AFG

   iso_num
0        4
1        4
2        4
3        4
4        4
```

[11]:

```
# YOUR CODE HERE
data = df[df["year"] == 2007]
cont = data["continent"].unique()
pop = np.zeros(len(cont))
for i in range(len(cont)):
    pop[i] = data[data["continent"] == cont[i]]["pop"].sum()


fig = px.bar(x=pop, y=cont, color=cont, title="Population per continent in␣
 ↪2007", labels={"x": "Population", "y": "Continent", "color": "Continents"})
fig.update_yaxes(categoryorder="total descending")
fig.show()
```


Population per continent in 2007
Oceania
Europe
Americas
Africa
Asia
Continent
0
0.5B
1B
1.5B
2B
2.5B
3B
3.5B
4B
Population
Continents
Asia
Europe
Africa
Americas
Oceania

[ ]: