

```
In [1]: import pandas as pd
import seaborn as sns
import plotly.express as px
import numpy as np

import matplotlib.pyplot as plt
```

```
In [2]: import plotly.io as pio
pio.renderers.default = "plotly_mimetype+notebook"
```

Matplotlib

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

```
In [3]: stocks = px.data.stocks()
stocks.head()
```

```
Out[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

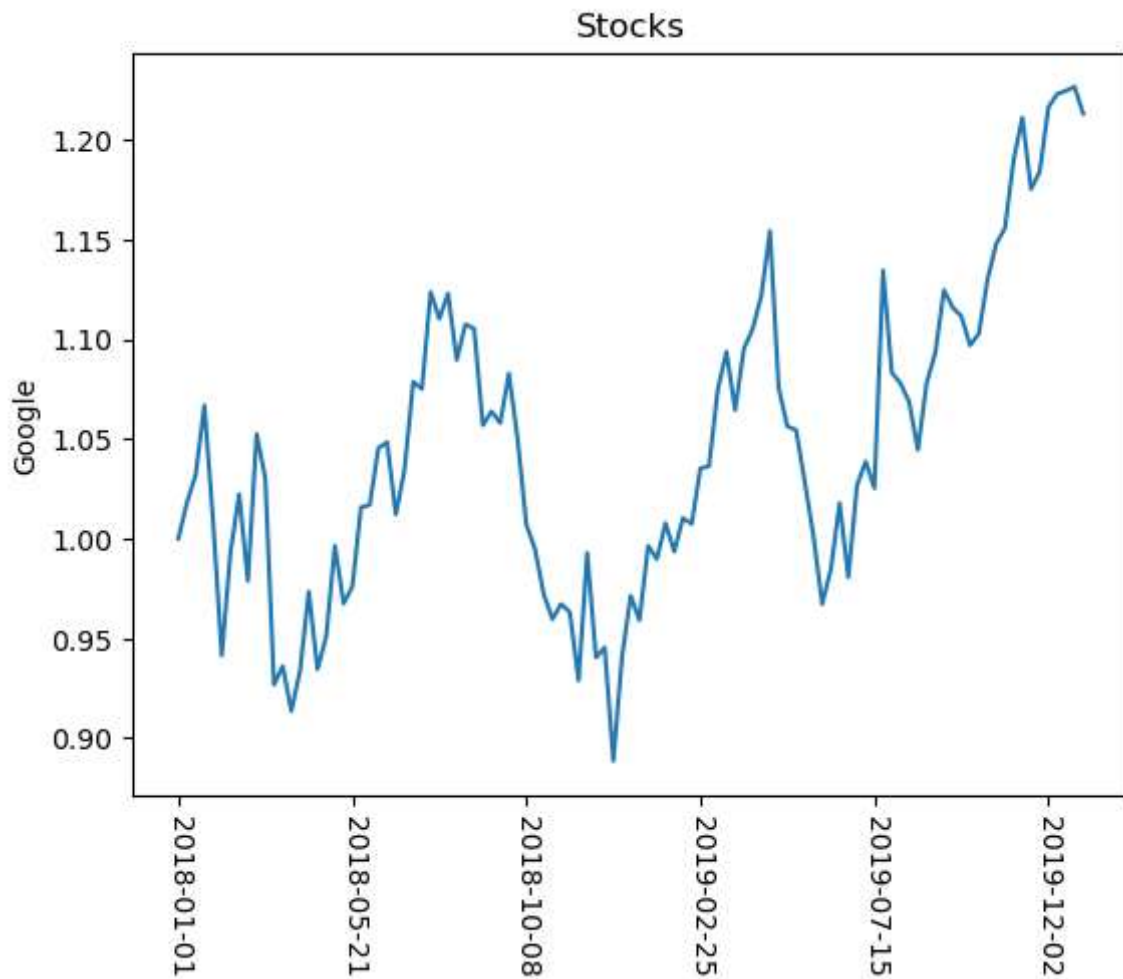
Question 1:

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

```
In [4]: # YOUR CODE HERE

x = stocks['date']
y = stocks['GOOG']

fig, ax = plt.subplots()
ax.plot(x,y)
ax.set_title('Stocks')
ax.set_ylabel('Google')
plt.xticks(np.arange(0, len(x)+1, step=20), rotation=-90)
plt.show()
```



Question 2:

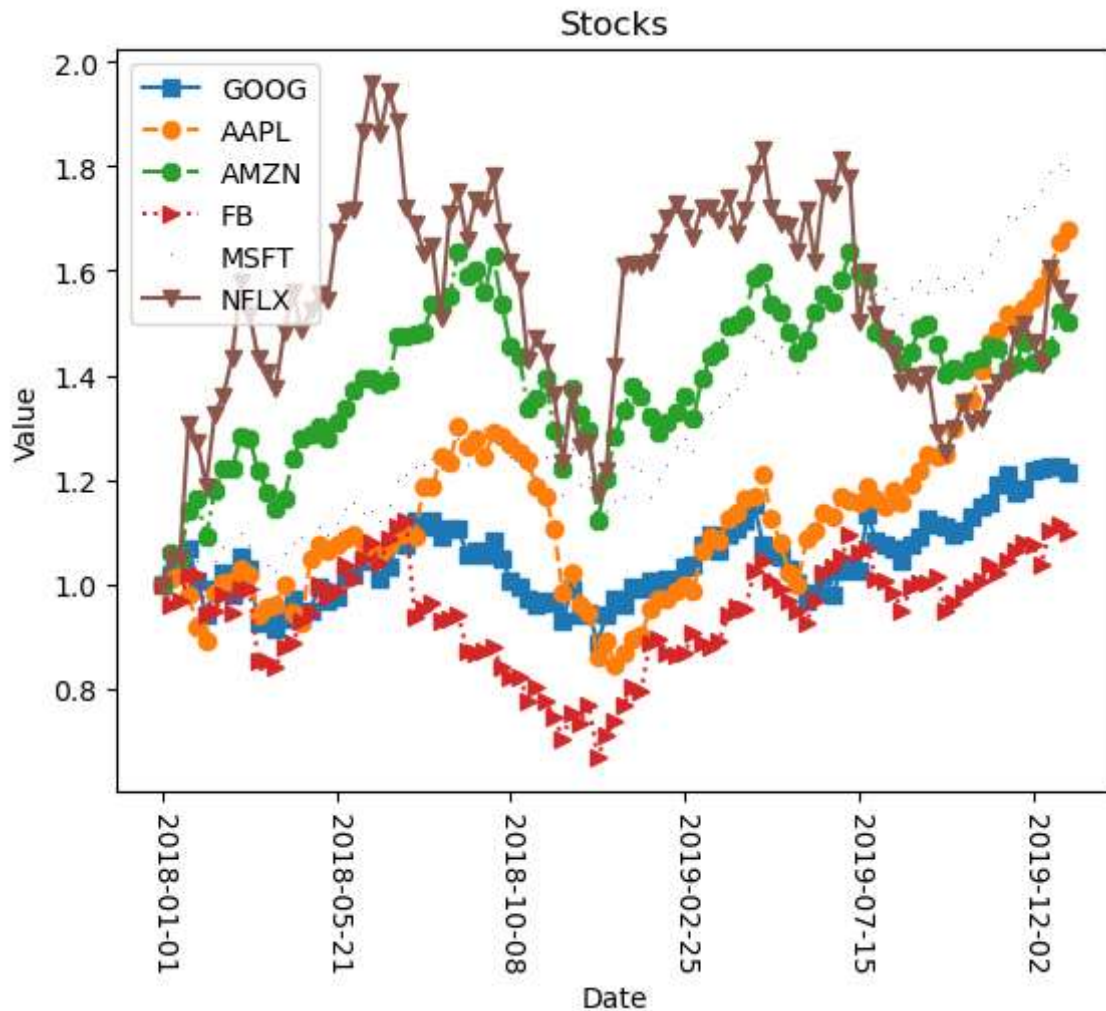
You've already plot data from one stock. It is possible to plot multiples of them to support comparison.

To highlight different lines, customise line styles, markers, colors and include a legend to the plot.

```
In [5]: # YOUR CODE HERE
x = stocks['date']
ya = stocks['GOOG']
yb = stocks['AAPL']
yc = stocks['AMZN']
yd = stocks['FB']
ye = stocks['MSFT']
yf = stocks['NFLX']

fig, ax = plt.subplots()
ax.plot(x, ya, '-s', label = 'GOOG')
ax.plot(x, yb, '--o', label = 'AAPL')
ax.plot(x, yc, '-.8', label = 'AMZN')
ax.plot(x, yd, ':>', label = 'FB')
ax.plot(x, ye, ',', label = 'MSFT')
ax.plot(x, yf, '-v', label = 'NFLX')
ax.set_title('Stocks')
```

```
plt.xticks(np.arange(0, len(x)+1, step=20), rotation = -90)
ax.set_xlabel('Date')
ax.set_ylabel('Value')
plt.legend(loc="upper left")
plt.show()
```



Seaborn

First, load the `tips` dataset

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

```
Out[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

Question 3:

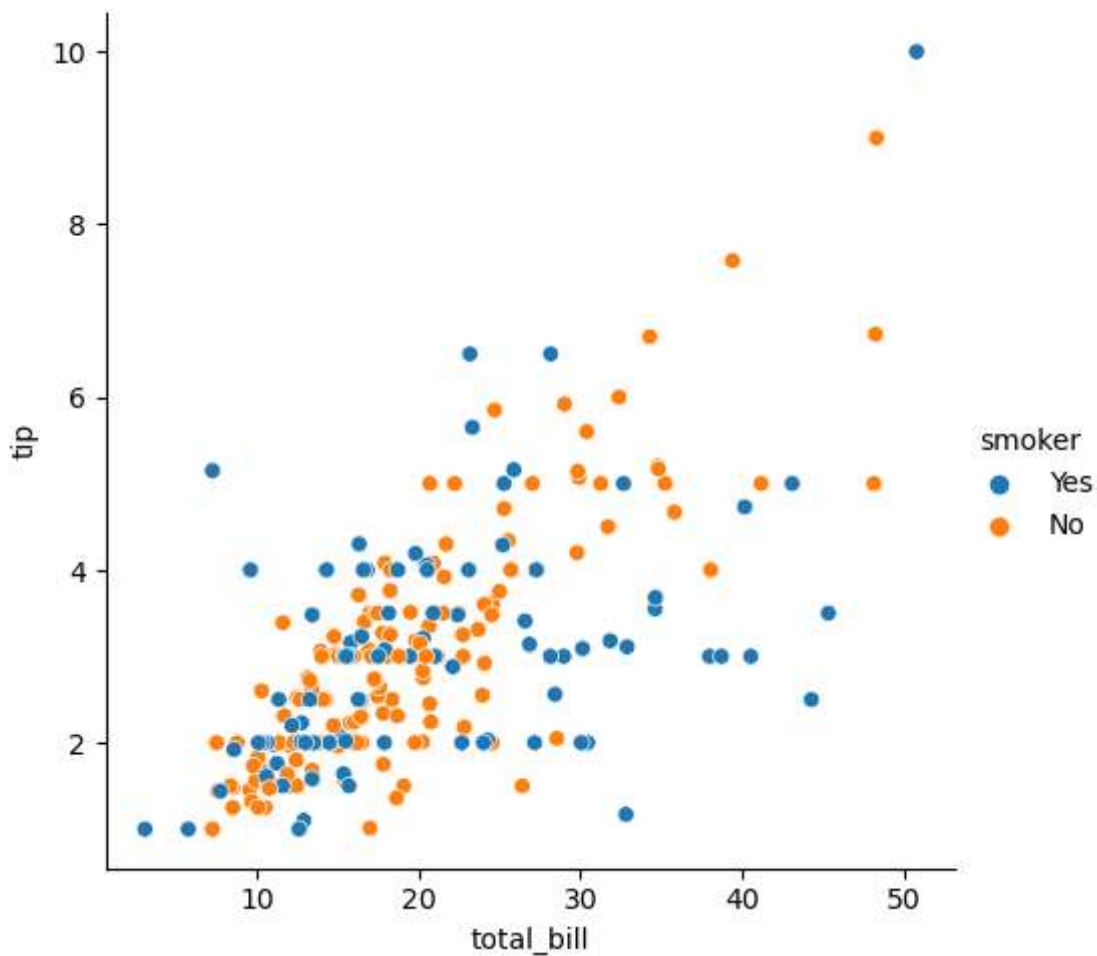
Let's explore this dataset. Pose a question and create a plot that support drawing answers for your question.

Some possible questions:

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

```
In [7]: # YOUR CODE HERE
#What is the contribution of the total bill on the amount of tip?
sns.relplot(x='total_bill', y='tip', data=tips, hue='smoker')
plt.plot()
```

Out[7]: []



Plotly Express

Question 4:

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

The stocks dataset

Hints:

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

```
In [30]: # YOUR CODE HERE
df = px.data.stocks()
fig = px.line(df, x="date", y=["GOOG", 'AMZN', 'AAPL', 'FB', 'MSFT', 'NFLX'], markers=
fig.show()
```

The tips dataset

```
In [9]: # YOUR CODE HERE
df = px.data.tips()
fig = px.scatter(df, x='total_bill', y='tip', color = 'smoker')

fig.show()
```

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

```
In [10]: #Load data
df1 = px.data.gapminder()
df2 = df1[df1['year']==2007]
df2.head()
```

Out[10]:

	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	4
23	Albania	Europe	2007	76.423	3600523	5937.029526	ALB	8
35	Algeria	Africa	2007	72.301	33333216	6223.367465	DZA	12
47	Angola	Africa	2007	42.731	12420476	4797.231267	AGO	24
59	Argentina	Americas	2007	75.320	40301927	12779.379640	ARG	32

In [18]:

```
# YOUR CODE HERE
contlist=df2.groupby('continent').sum()
print(contlist)
fig=px.bar(contlist,'pop', color = ['Africa', 'Americas', 'Asia', 'Europe', 'Oceania'])

fig.update_yaxes(categoryorder= "total ascending")
fig.show()
```

	year	lifeExp	pop	gdpPercap	iso_num
continent					
Africa	104364	2849.914	929539692	160629.695446	23859
Americas	50175	1840.203	898871184	275075.790634	9843
Asia	66231	2334.040	3811953827	411609.886714	13354
Europe	60210	2329.458	586098529	751634.449078	12829
Oceania	4014	161.439	24549947	59620.376550	590

In []: