In [1]:

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

import matplotlib.pyplot as plt
```

In [2]:

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

Matplotlib

For this excercise, 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]:

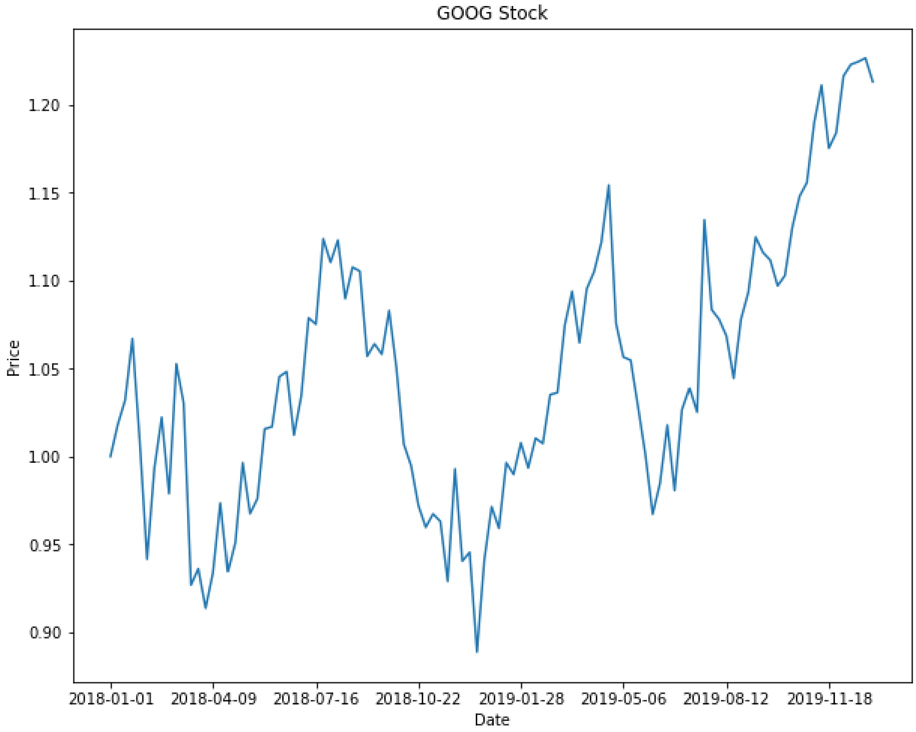
	date	GOOG	AAPL	AMZN	FB	NFLX	MSFT
0	2018-01-01	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	2018-01-08	1.018172	1.011943	1.061881	0.959968	1.053526	1.015988
2	2018-01-15	1.032008	1.019771	1.053240	0.970243	1.049860	1.020524
3	2018-01-22	1.066783	0.980057	1.140676	1.016858	1.307681	1.066561
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 [5]:

```
# YOUR CODE HERE

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

plt.plot(x, y)

ax = plt.gca()
from matplotlib.pyplot import MultipleLocator
x_major_locator=MultipleLocator(14)
ax.xaxis.set_major_locator(x_major_locator)


plt.title('GOOG Stock')
plt.xlabel('Date')
plt.ylabel('Price')
plt.rcParams['figure.figsize'] = (10.0, 8.0)

plt.show()
```

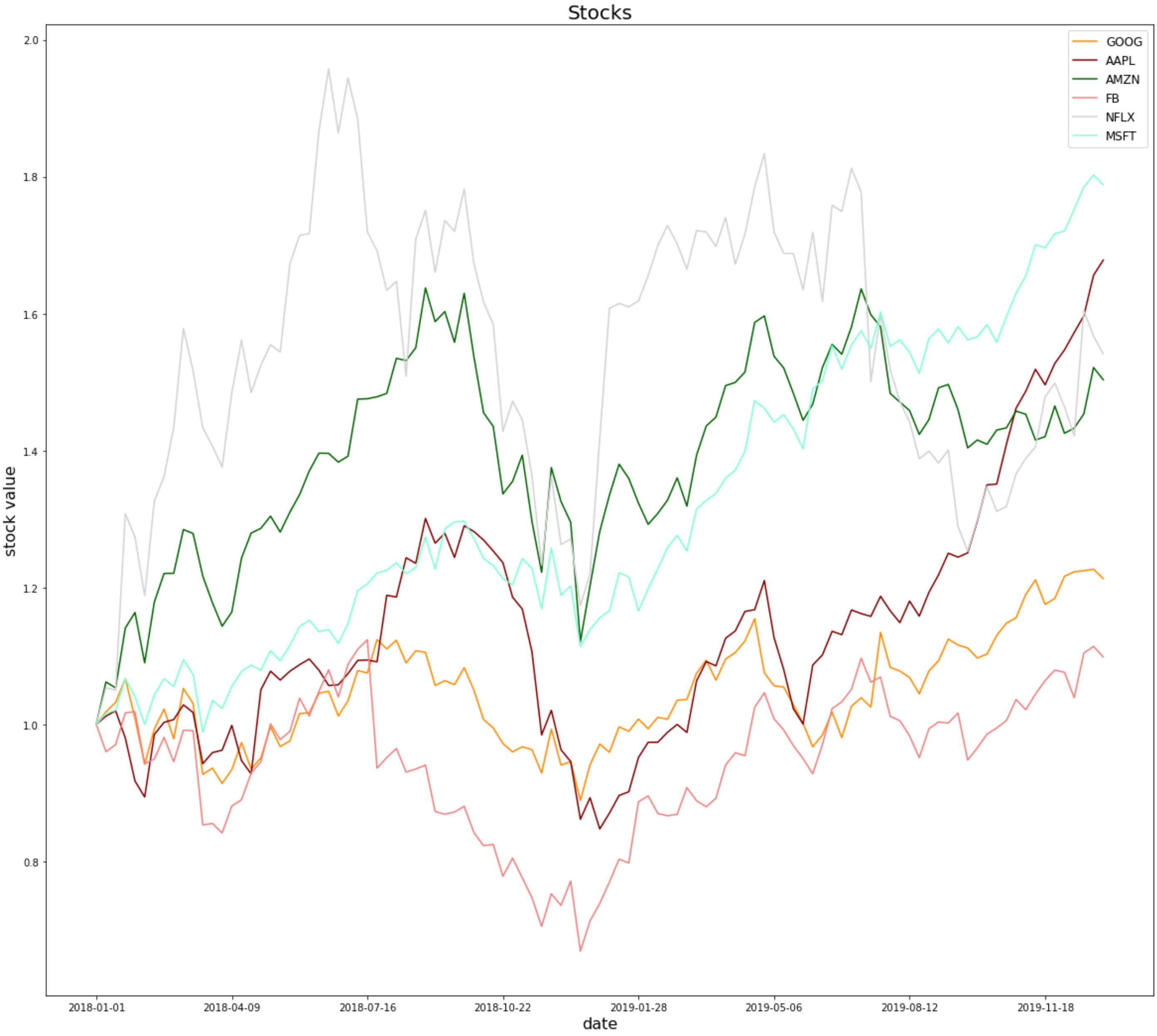


Question 2:

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 [6]:

```
# YOUR CODE HERE
plt.plot(x, stocks['GOOG'], 'darkorange', label = "GOOG")
plt.plot(x, stocks['AAPL'], 'maroon', label = "AAPL")
plt.plot(x, stocks['AMZN'], 'darkgreen', label = "AMZN")
plt.plot(x, stocks['FB'], 'lightcoral', label = "FB")
plt.plot(x, stocks['NFLX'], 'lightgray', label = "NFLX")
plt.plot(x, stocks['MSFT'], 'aquamarine', label = "MSFT")

ax = plt.gca()
from matplotlib.pyplot import MultipleLocator
x_major_locator=MultipleLocator(14)
ax.xaxis.set_major_locator(x_major_locator)


plt.title('Stocks', fontsize = 20)
plt.xlabel('date', fontsize = 16)
plt.ylabel('stock value', fontsize =16)

plt.rcParams['figure.figsize'] = (16.0, 12.0)

plt.legend(prop = {'size':12})
plt.show()
```



Seaborn

First, load the tips (https://github.com/mwaskom/seaborn-data/blob/master/tips.csv) 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:

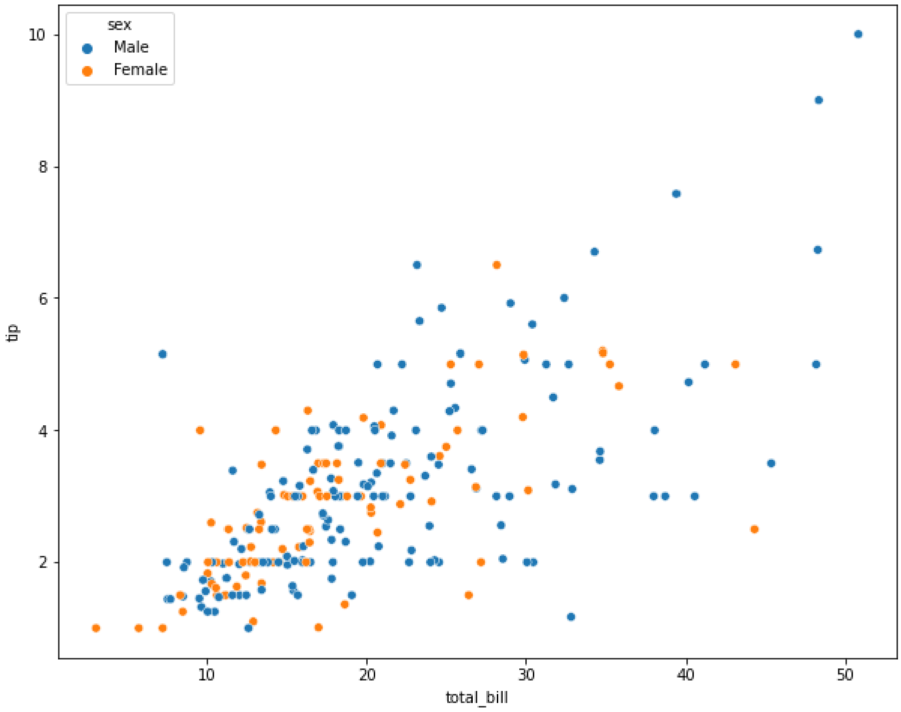
Let's explore this dataset. Pose a question and create a plot that support drawing answers for your question.

Question:

- What's the trend of the increase of tips while the value of total bill's increasing between different gender?

In [7]:

```
# YOUR CODE HERE
sns.scatterplot(x="total_bill", y="tip", data = tips, hue="sex")

plt.show()
```



Plotly Express

Question 4:

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

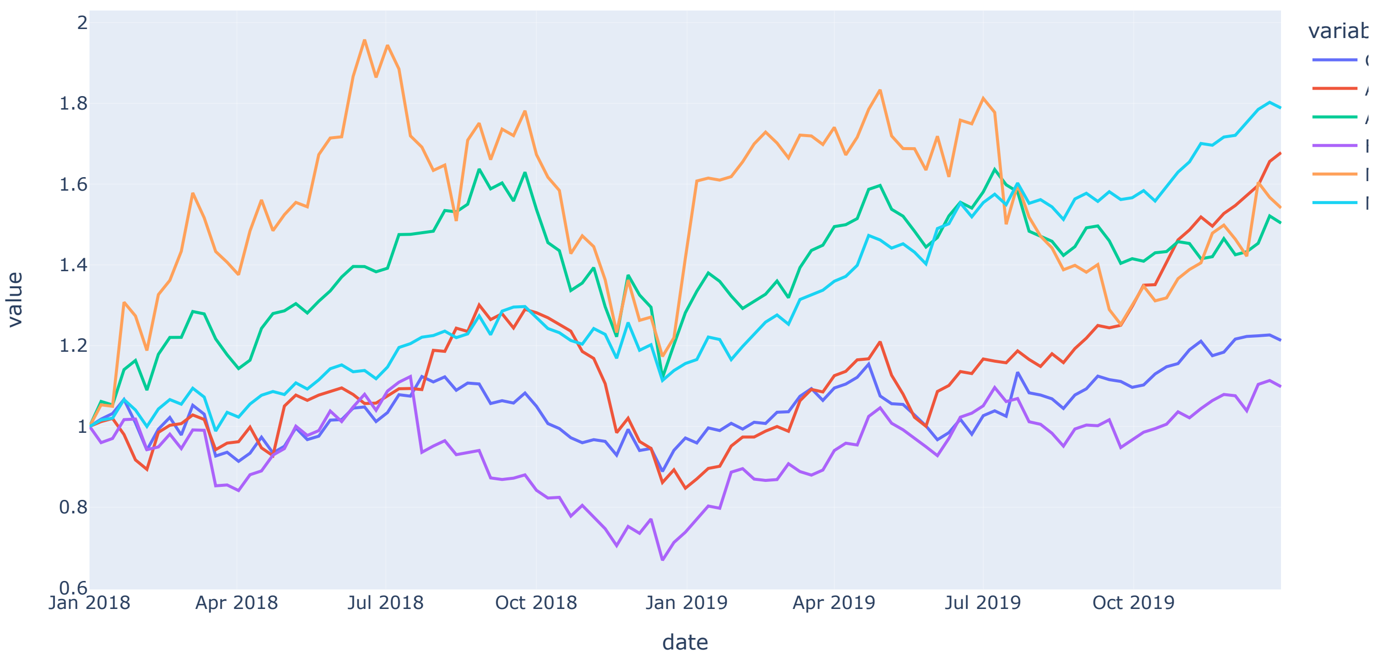
The stocks dataset

Hints:

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

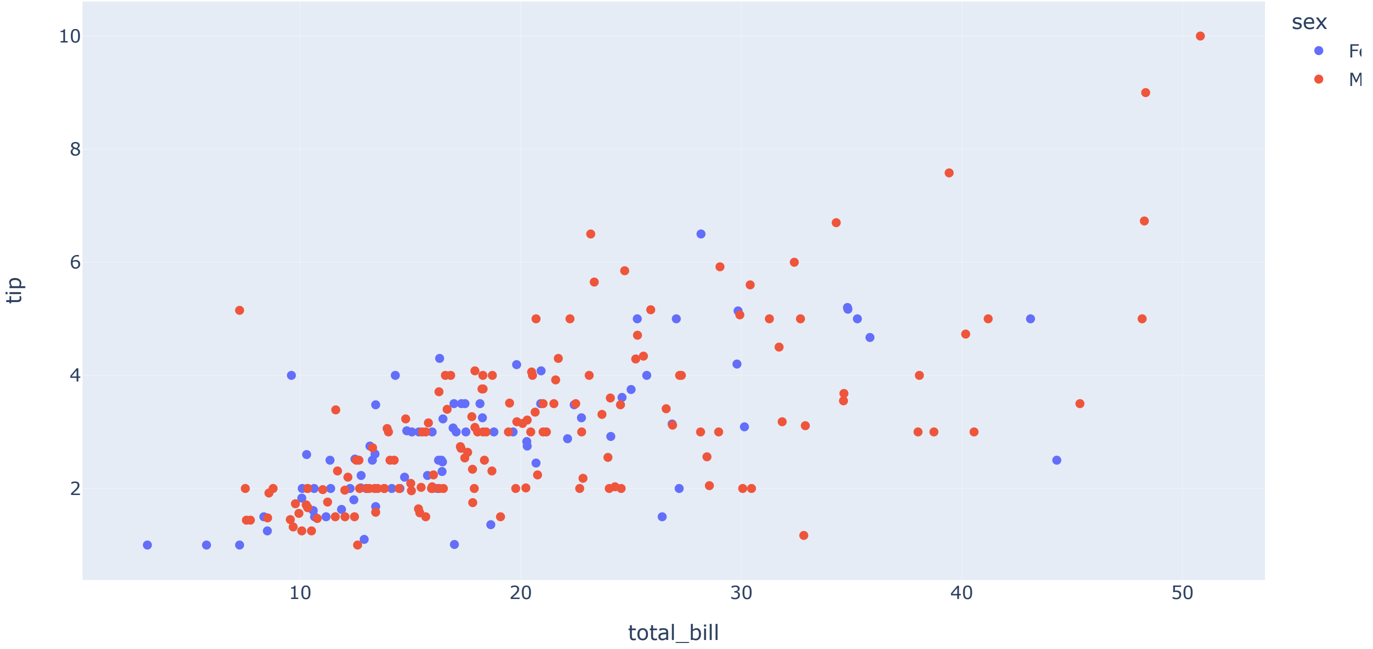
In [8]:

```
fig = px.line(stocks, x=stocks.date, y=["GOOG", "AAPL", "AMZN", "FB", "NFLX", "MSFT"])

fig.show()
```



The tips dataset

In [10]:
```
# YOUR CODE HERE
fig = px.scatter(tips, x=tips.total_bill, y=tips.tip, color=tips.sex)

fig.show()
```



Question 5:

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

Hints:

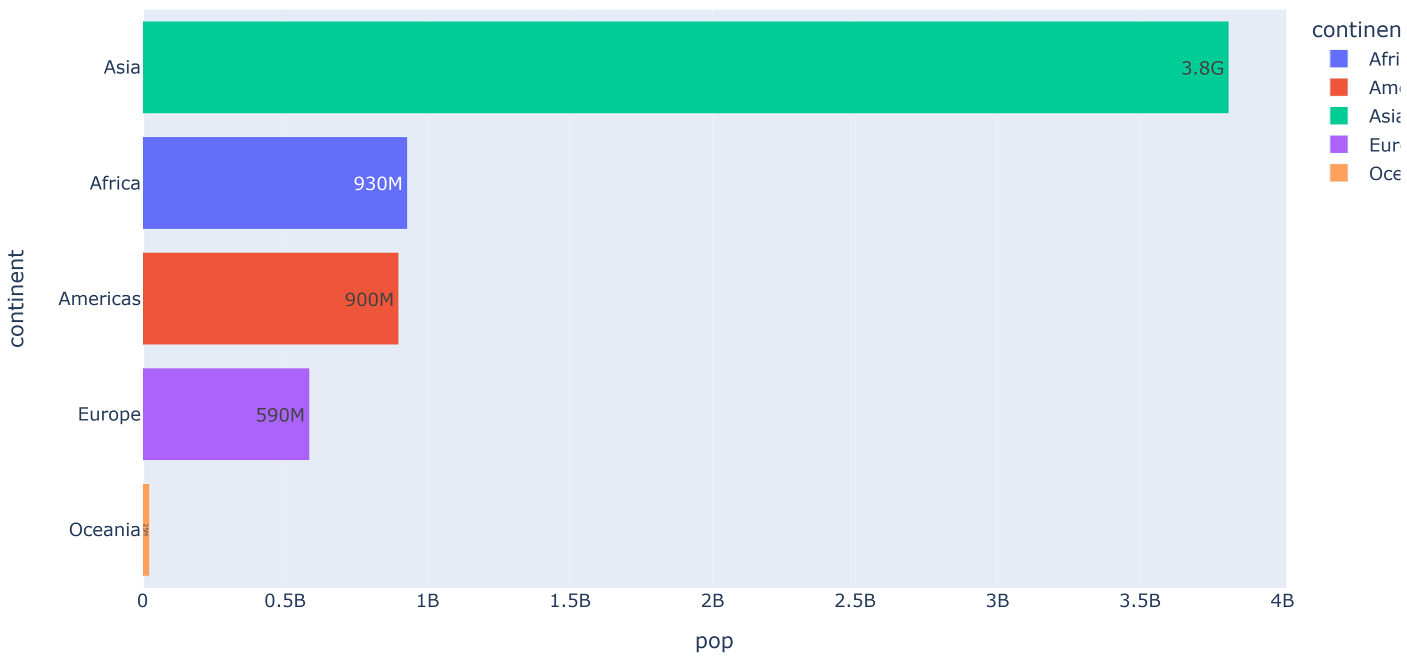
- Extract the 2007 year data from the dataframe. You have to process the data accordingly
- use plotly bar (https://plotly.com/python-api-reference/generated/plotly.express.bar)
- Add different colors for different continents
- Sort the order of the continent for the visualisation. Use axis layout setting (https://plotly.com/python/reference/layout/xaxis/)
- Add text to each bar that represents the population

In [11]:
```
#load data
df = px.data.gapminder()
df.head()
```

Out[11]:

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

```
In [12]: # YOUR CODE HERE
         df_2007 = df.query('year==2007')
         dx = df_2007.groupby('continent').sum()
         fig = px.bar(dx, x="pop", y=dx.index, color = dx.index, orientation='h', text='pop')

         fig.update_traces(texttemplate='%{text:.2s}', textposition='inside')
         fig.update_yaxes(categoryorder="sum ascending" )

         fig.show()
```


Asia
3.8G
Africa
930M
Americas
900M
Europe
590M
Oceania
continent
0.5B
1B
1.5B
2B
2.5B
3B
3.5B
4B
pop
continen
Afri
Am
Asia
Eur
Oce