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:

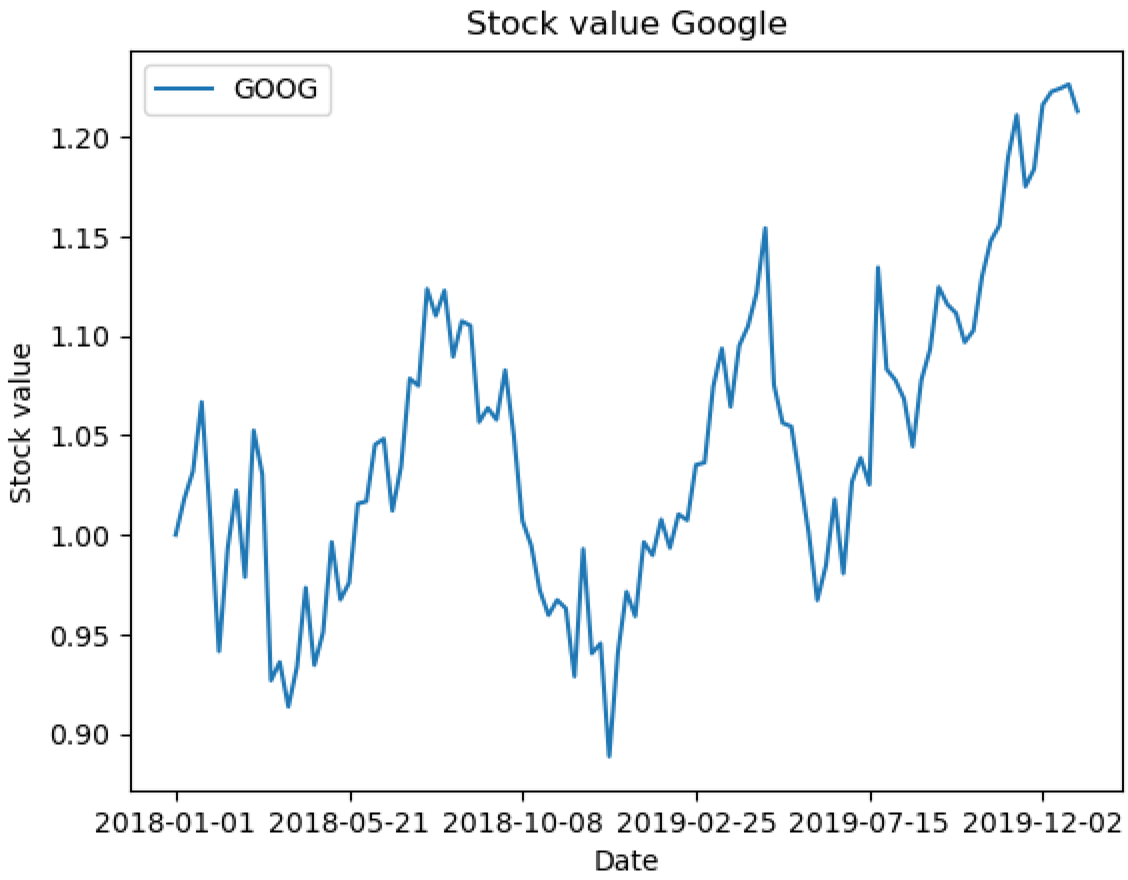
Select a stock and create a suitable plot for it. Make sure the plot is readable with relevant information, such as date, values.

In [4]:

```
stocks.plot(x='date', y='GOOG')
plt.xlabel('Date')
plt.ylabel('Stock value')
plt.title(label='Stock value Google',)
```

Out[4]:

```
Text(0.5, 1.0, 'Stock value Google')
```



Question 2:

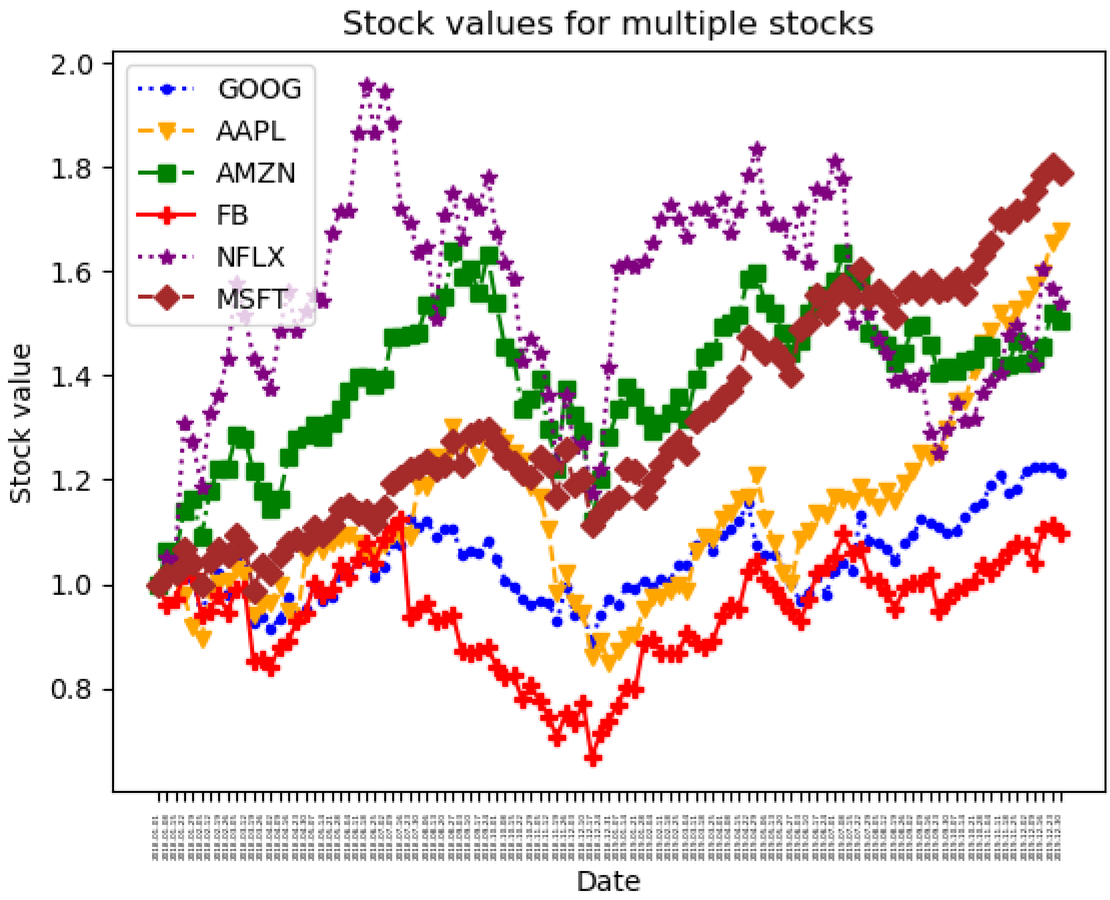
You've already plot data from one stock. It is possible to plot multiples of them to support comparison. To highlight different lines, customise line styles, markers, colors and include a legend to the plot.

In [5]:

```
plt.plot(stocks['date'],stocks['GOOG'], color='blue', linestyle='dotted', marker='.')
plt.plot(stocks['date'],stocks['AAPL'], color='orange', linestyle='dashed', marker='v')
plt.plot(stocks['date'],stocks['AMZN'], color='green', linestyle='dashdot', marker='s')
plt.plot(stocks['date'],stocks['FB'], color='red', linestyle='solid', marker='P')
plt.plot(stocks['date'],stocks['NFLX'], color='purple', linestyle='dotted',marker='*')
plt.plot(stocks['date'],stocks['MSFT'], color='brown', linestyle='dashed',marker='D')
plt.legend(['GOOG','AAPL','AMZN','FB','NFLX','MSFT'])
plt.xlabel('Date')
plt.xticks(rotation=90, size=3)
plt.ylabel('Stock value')
plt.title(label='Stock values for multiple stocks',)
```

Out[5]:

```
Text(0.5, 1.0, 'Stock values for multiple stocks')
```



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?

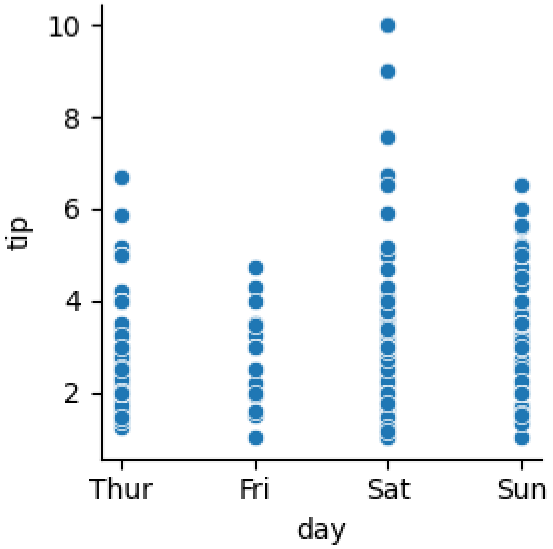
In [7]:

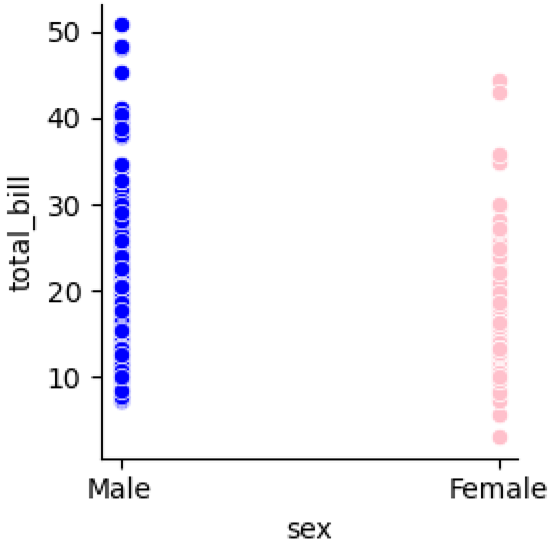
```
# Q: On which day is the highest tip given?
# A: Saturday.
```

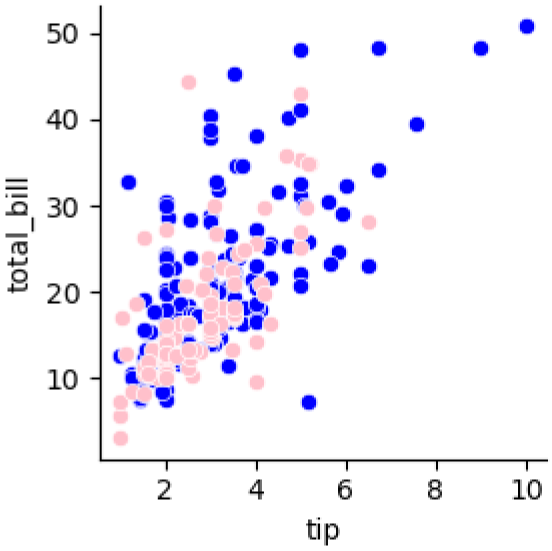
```
n=sns.FacetGrid(tips)
n.map(sns.scatterplot, 'day', 'tip')
plt.show()

# Q: Which sex has the lowest value found for the total bill?
# A: The women.
o=sns.FacetGrid(tips, hue='sex', palette=['blue','pink'])
o.map(sns.scatterplot, 'sex', 'total_bill')
plt.show()

#Q: Is there a stronger correlation between the total_bill and tip, for one gender?
#A: The correlation appears to be stronger for the female sex.
p=sns.FacetGrid(tips, hue='sex', palette=['blue','pink'])
p.map(sns.scatterplot, 'tip', 'total_bill')
plt.show()
```







Plotly Express

Question 4:

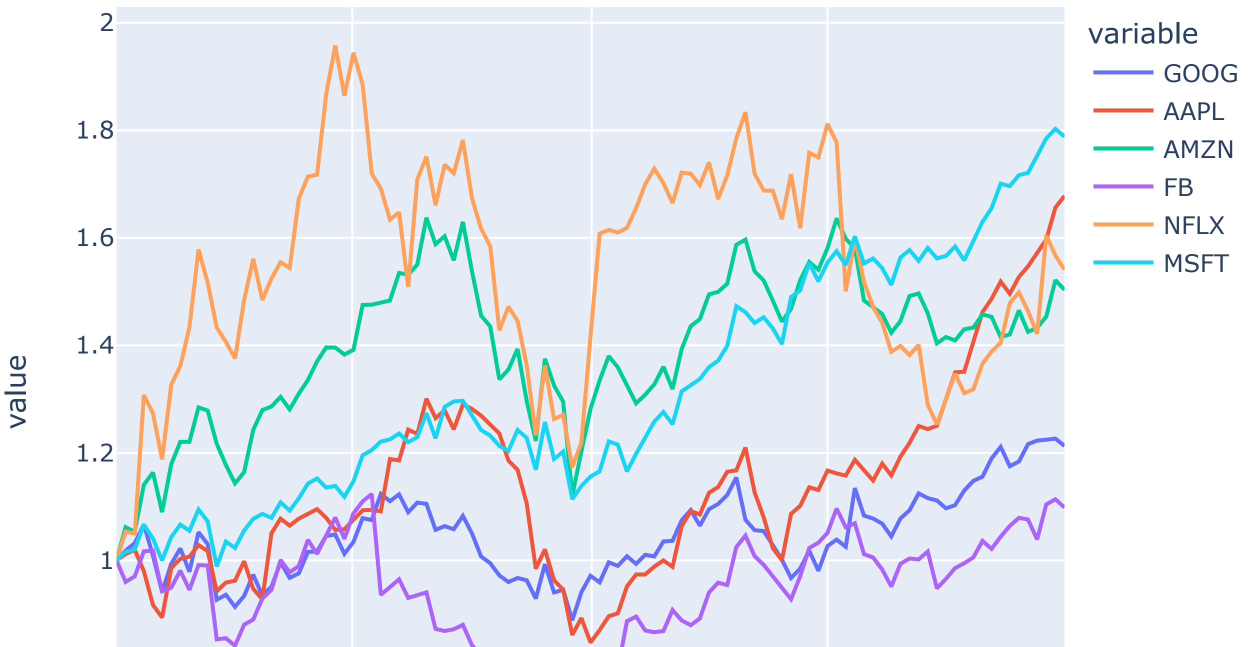
Redo the above exercises (challenges 2 & 3) with plotly express. Create diagrams which you can interact with.

The stocks dataset

Hints:

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

In [8]:
```
fig1 = px.line(stocks, x='date', y=stocks.keys())
fig1.show()
```

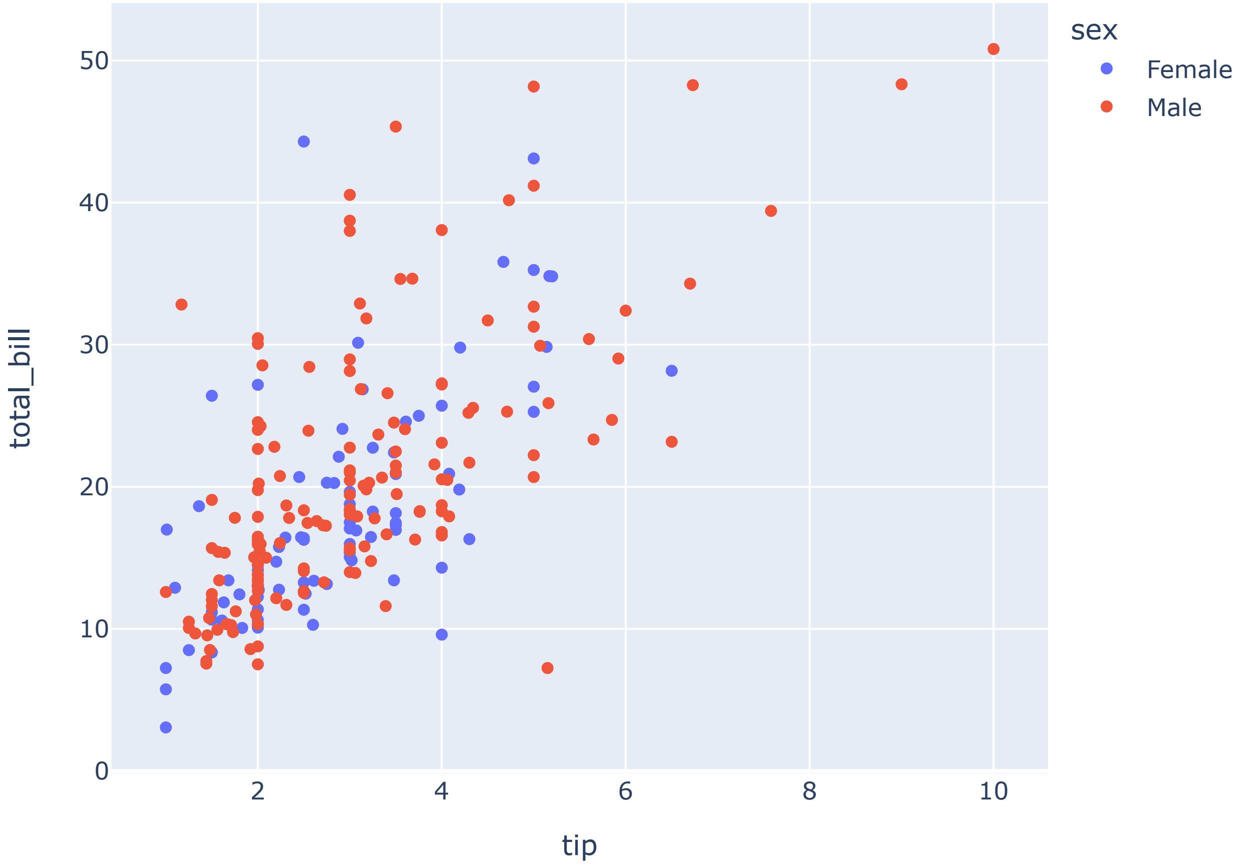


The tips dataset

In [10]:

```
#Q: Is there a stronger correlation between the total_bill and tip, for one gender?
#A: The correlation appears to be stronger for the female sex.

fig2 = px.scatter(tips, x='tip', y='total_bill', color='sex')
fig2.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

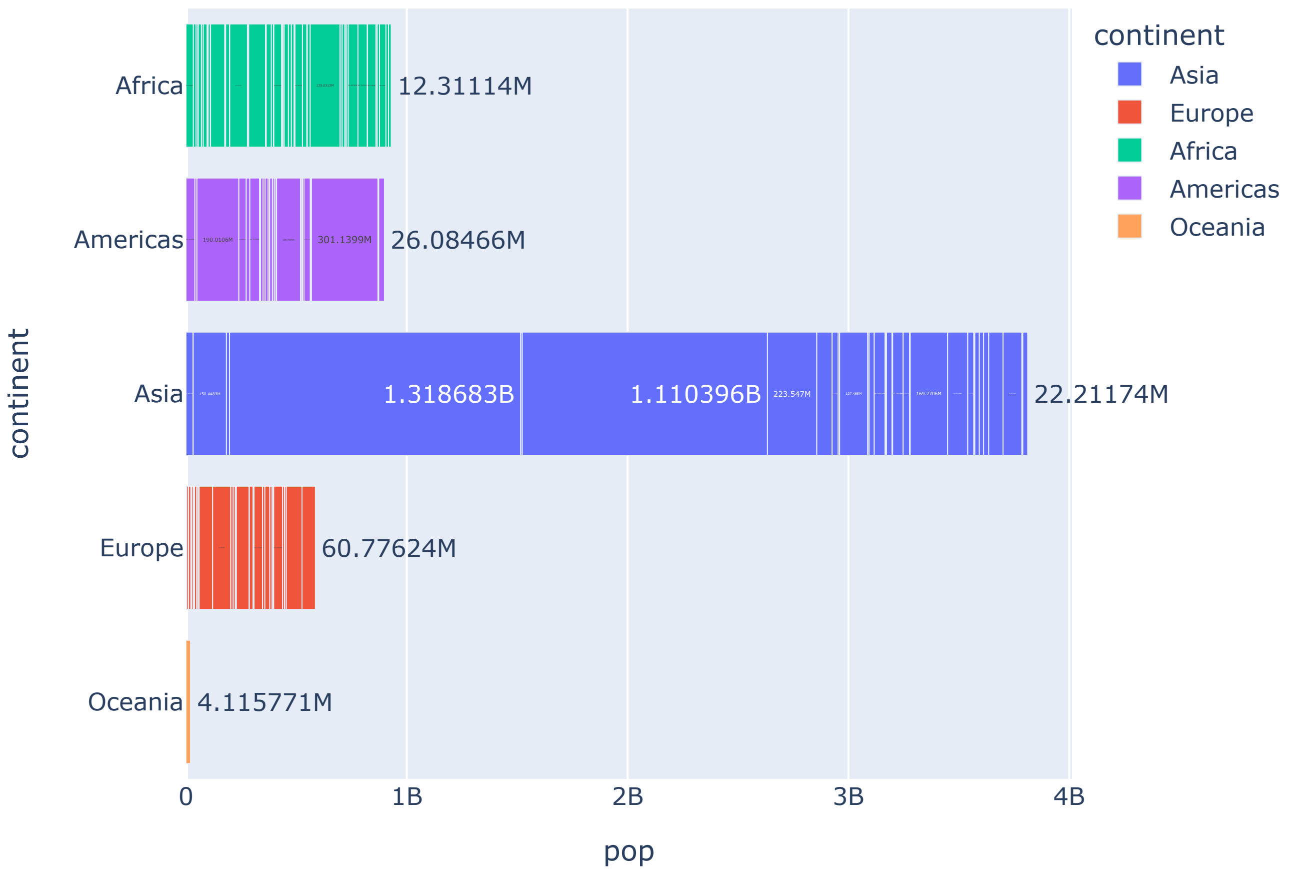
- Sort the order of the continent for the visualisation. Use axis layout setting
- Add text to each bar that represents the population

In [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 [11]:
```
df_2007 = df.query('year==2007')
fig3 = px.bar(df_2007, x='pop', y='continent', color='continent',
              orientation='h', text_auto=True)
fig3.update_yaxes(categoryorder='category descending')
fig3.update_traces(textfont_size=12, textangle=0, textposition='outside',
                   cliponaxis=False)
fig3.show()
```



In [ ]:
```
```