In [26]:
```
import pandas as pd
import seaborn as sns
import plotly.express as px

import matplotlib.pyplot as plt
```

In [27]:
```
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 [28]:
```
stocks = px.data.stocks()
stocks.head()
```

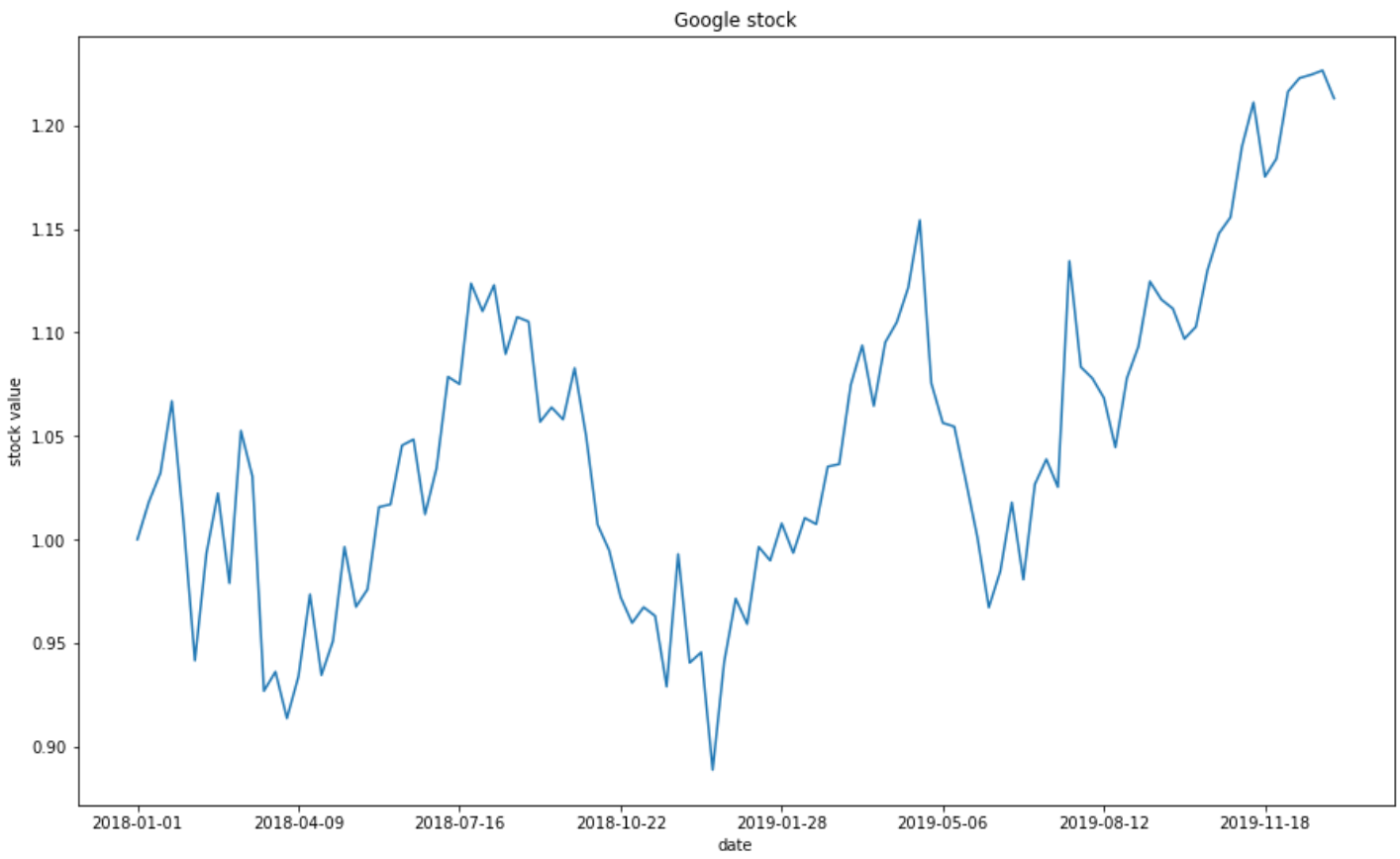
Out[28]:

	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

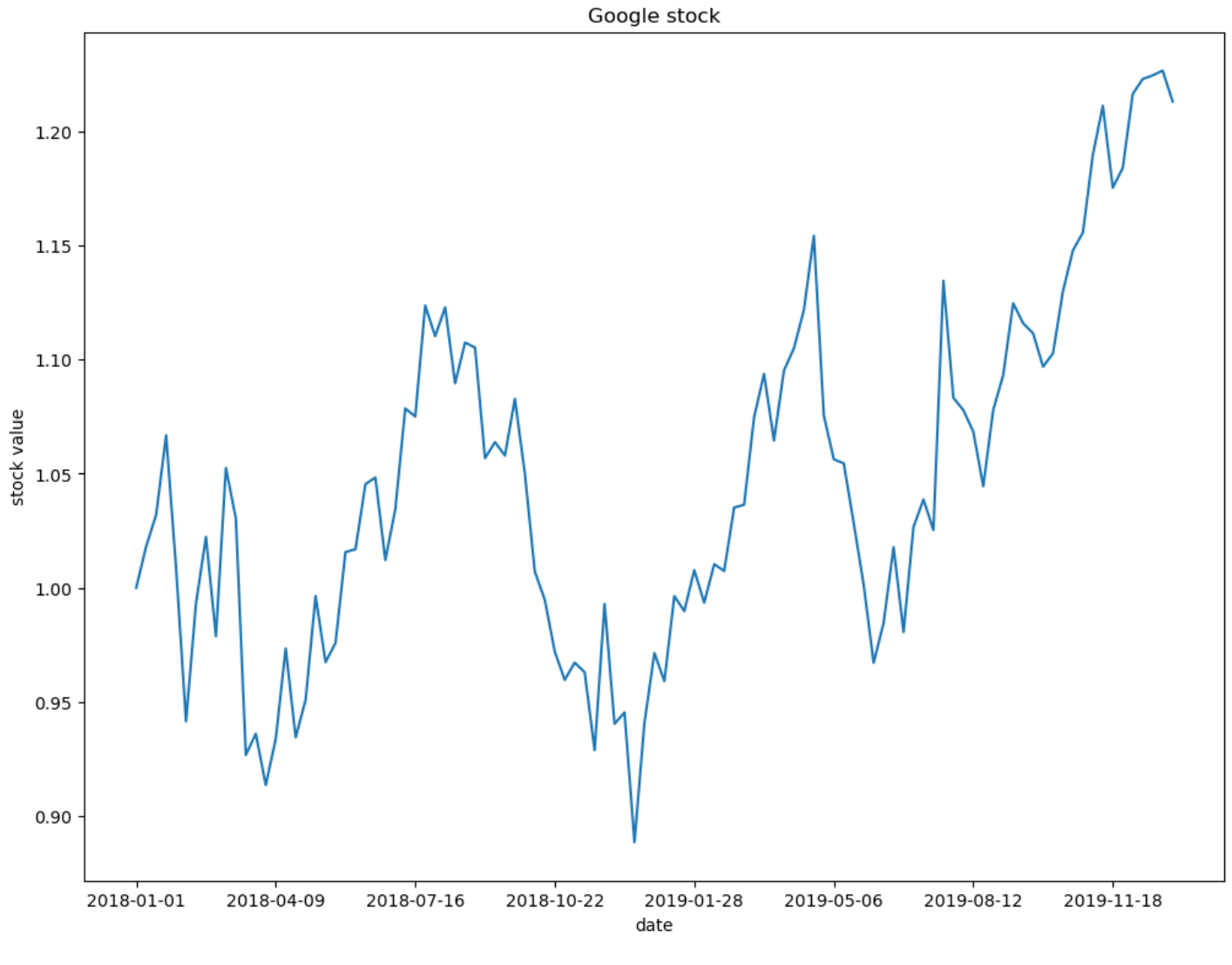
Question 1:

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

In [29]:
```
import matplotlib.ticker as ticker
x = stocks['date']
y = stocks['GOOG']
fig, ax = plt.subplots(figsize=(15,9))
ax.xaxis.set_major_locator(ticker.MultipleLocator(14))
ax.plot(x,y)
# set title
ax.set_title('Google stock')
# horizontal axis
ax.set_xlabel('date')
# vertical axis
ax.set_ylabel('stock value')
plt.show()
```



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



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 [38]:

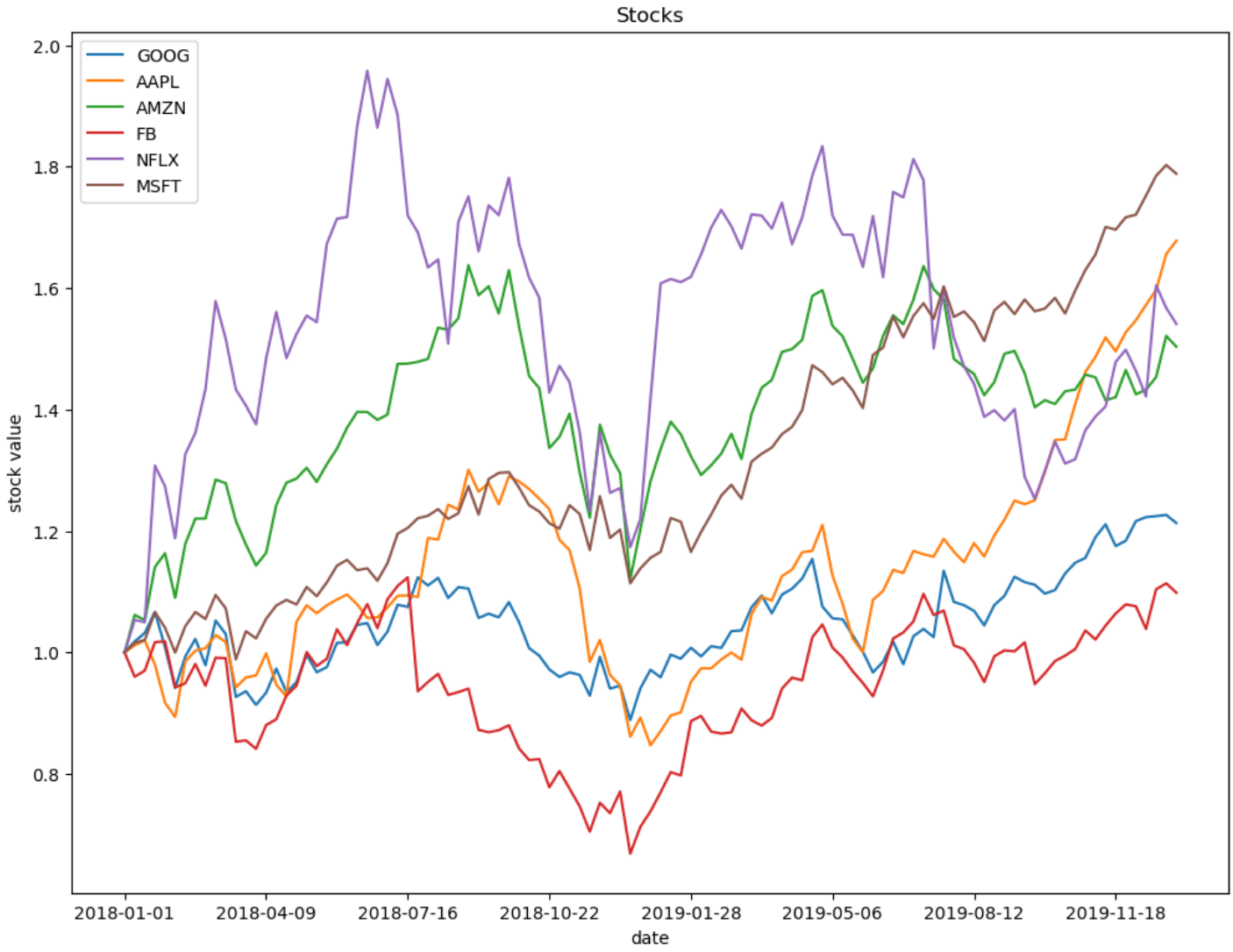
```
x = stocks['date']
y = stocks['GOOG']
z=stocks['AAPL']
c=stocks['AMZN']
d=stocks['FB']
e=stocks['NFLX']
w=stocks['MSFT']

fig, ax = plt.subplots(figsize=(15,9))
ax.xaxis.set_major_locator(ticker.MultipleLocator(14))
ax.plot(x,y, color='green',label='GOOG')
ax.plot(x,z,label='AAPL')
ax.plot(x,c,label='AMZN')
ax.plot(x,d,label='FB')
ax.plot(x,e,label='NFLX')
ax.plot(x,w,label='MSFT')

# set title
ax.set_title('Google stock')
# horizontal axis
ax.set_xlabel('date')
# vertical axis
ax.set_ylabel('stock value')
ax.legend()
plt.show()
```

In [5]:

```
# YOUR CODE HERE
```



Seaborn

First, load the tips dataset

In [39]:
```
tips = sns.load_dataset('tips')
tips.head()
```

Out[39]:

	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:

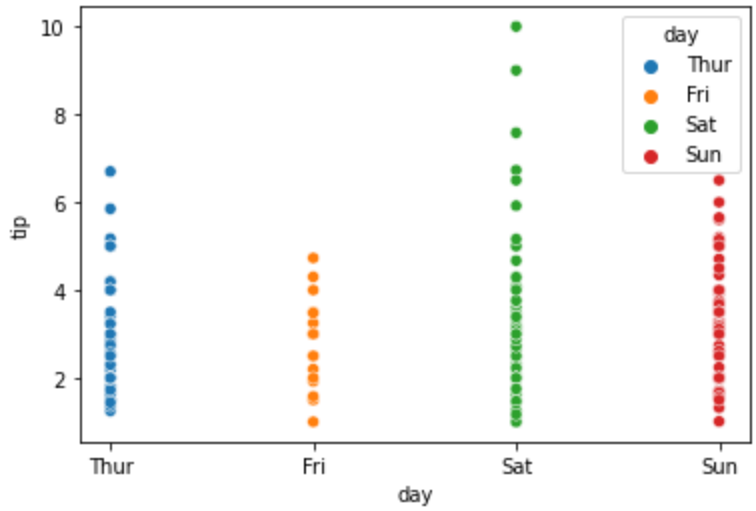
Let's explore this dataset. Pose a question and create a plot that support drawing answers for your question.

Some possible questions:

- Are there differences between male and female when it comes to giving tips?

- What attribute correlate the most with tip?

In [96]:
```
#Are there differences between day of weeks when it comes to giving tips?
# day
sns.scatterplot(x='day', y='tip', data=tips,hue='day')
plt.show()
```



Yes, tipping is more on weekends than on weekdays.

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 [63]:
```
# YOUR CODE HERE
df = px.data.stocks()
fig = px.line(df, x="date", y=["GOOG","AAPL","FB","AMZN","NFLX","MSFT"])

fig.show()
```

The tips dataset

In [97]:

```
# YOUR CODE HERE
fig = px.scatter(tips, x="day", y=["tip"],color='day')

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 [12]:
```
#load data
df = px.data.gapminder()
df.head()
```

Out[12]:

	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 [93]:
```
# YOUR CODE HERE
df = px.data.gapminder()
df_2007 = df.query('year==2007')
df_new = df_2007.groupby('continent').sum()
df_new.sort_values("pop",ascending=False,inplace=True)
#fig.update_xaxes(categoryorder='array', categoryarray= ['Asia','Africa','Americas','Europ
fig = px.bar(df_new, x="pop", y=df_new.index, color=df_new.index, orientation='h')


fig.show()
```

In [85]:

continent	year	lifeExp	pop	gdpPercap	iso_num
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 [90]:

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

In [ ]: