

```
In [103... import pandas as pd
import seaborn as sns
import plotly.express as px

import matplotlib.pyplot as plt

In [104... 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 [105... stocks = px.data.stocks()
stocks.head()
```

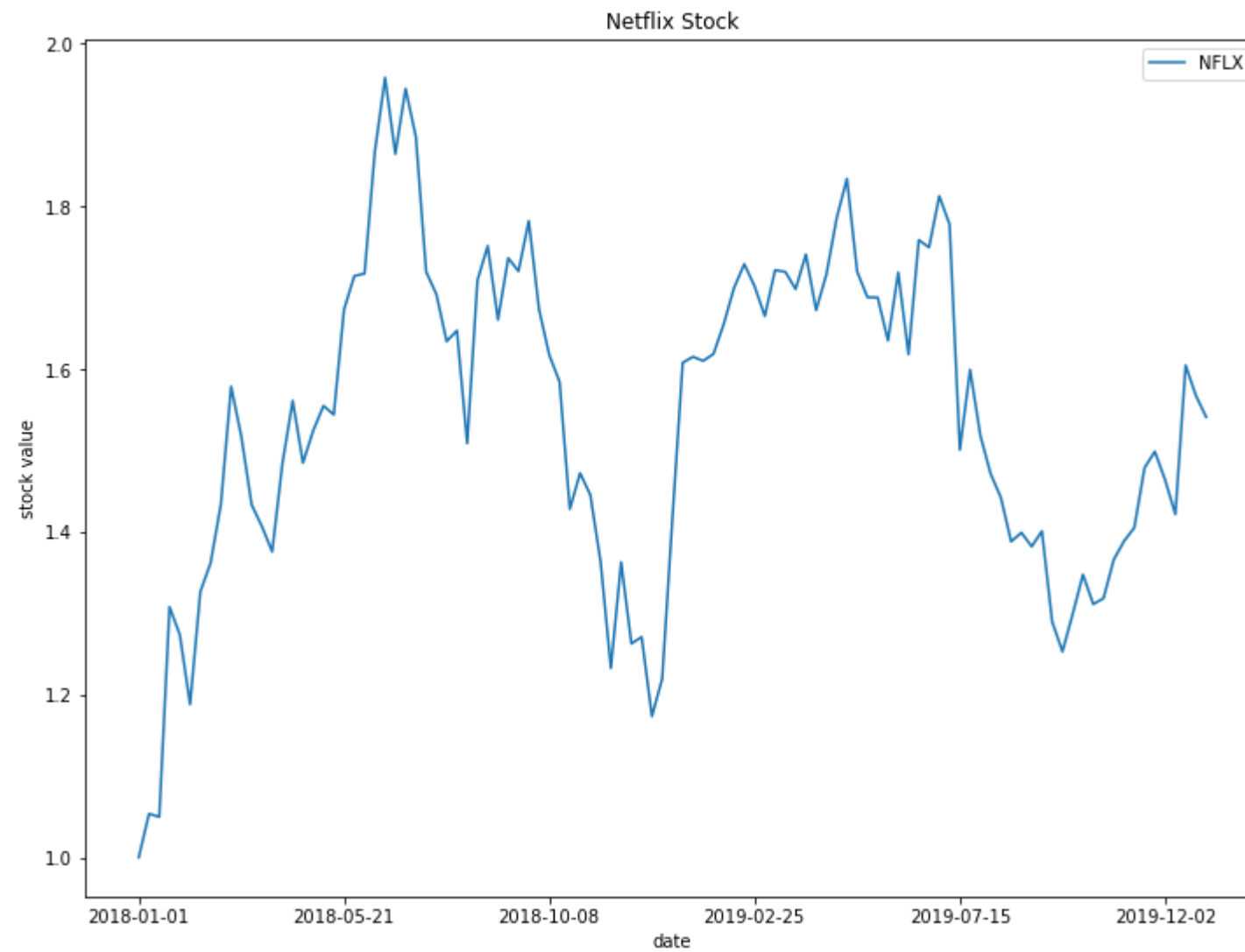
Out[105]:

	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.

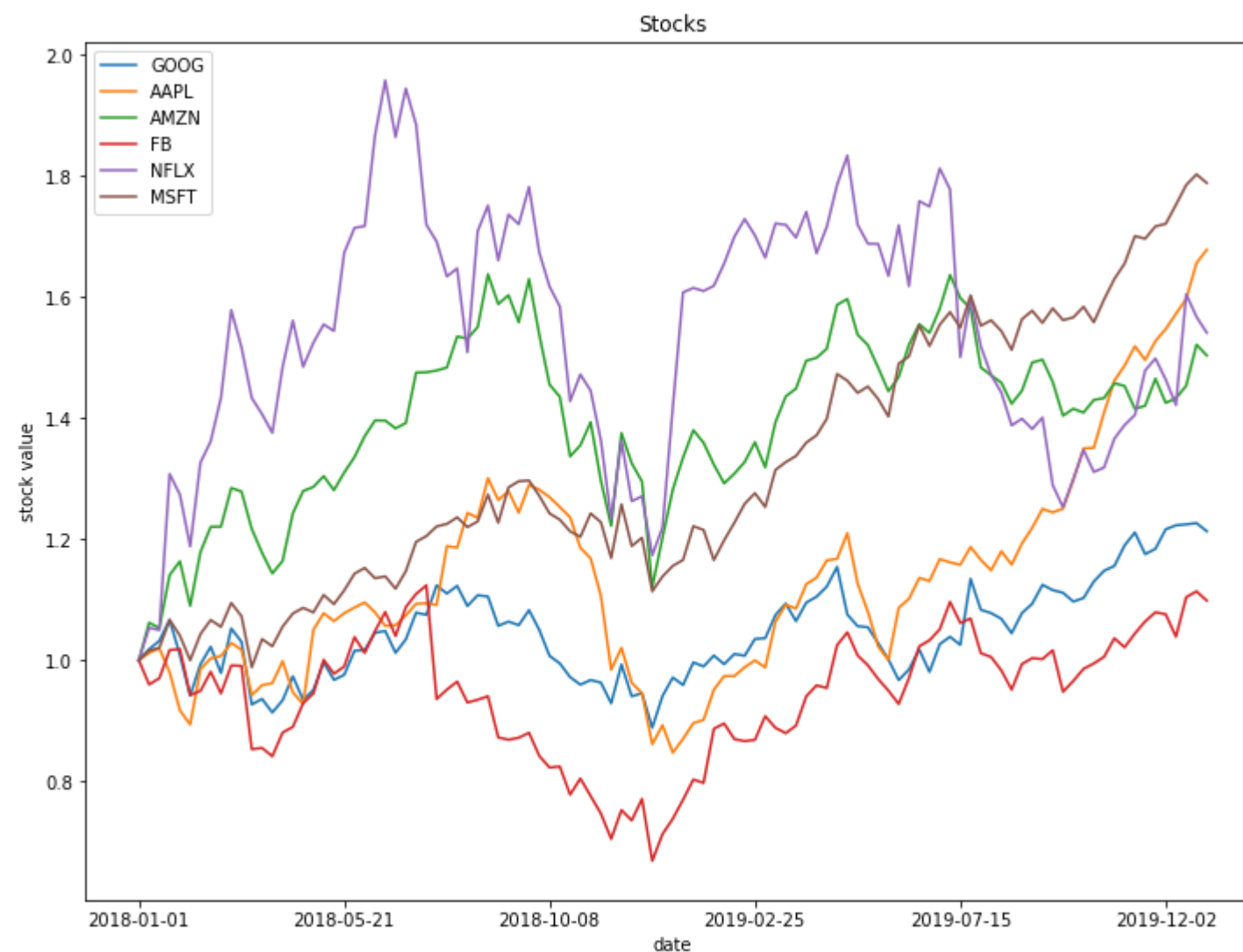
```
In [106... ax = stocks.plot(x='date', y='NFLX', figsize=(12, 9))
ax.set_title('Netflix Stock')
ax.set_xlabel('date')
ax.set_ylabel('stock value')
plt.show()
```



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 [107... ax = stocks.plot(x='date', figsize=(12,9))
ax.set_title('Stocks')
ax.set_xlabel('date')
ax.set_ylabel('stock value')
plt.show()
```



Seaborn

First, load the `tips` dataset

```
In [108]: tips = sns.load_dataset('tips')
tips.head()
```

```
Out[108]:
```

	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:

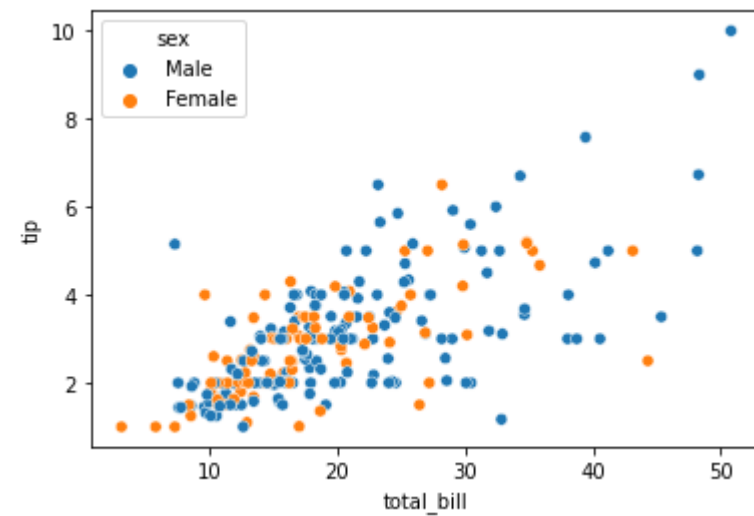
Let's explore this dataset. Pose a question and create a plot that support drawing answers for your question.

Some possible questions:

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

In [109... *#Question: Are there differences between male and female when it comes to giving tips?*

```
p = sns.scatterplot(x='total_bill', y='tip', data=tips, hue='sex')
plt.show()
```



Plotly Express

Question 4:

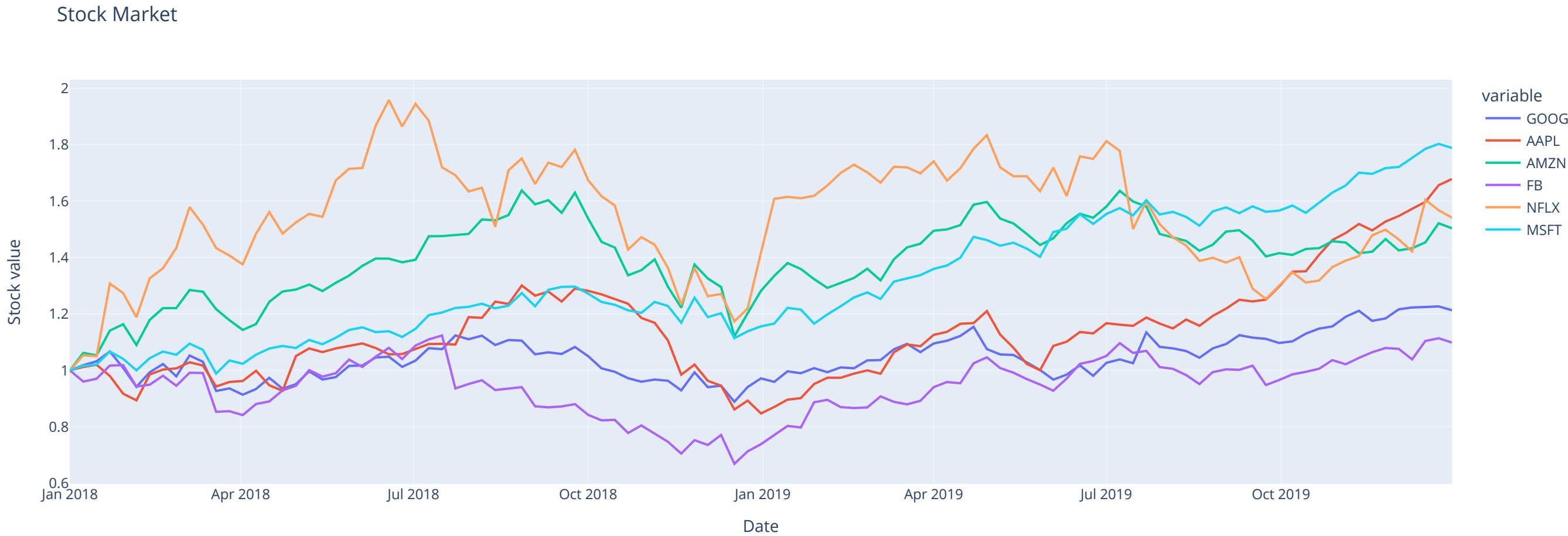
Redo the above exercises (challenges 2 & 3) with plotly express. Create diagrams which you can interact with.

The stocks dataset

Hints:

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

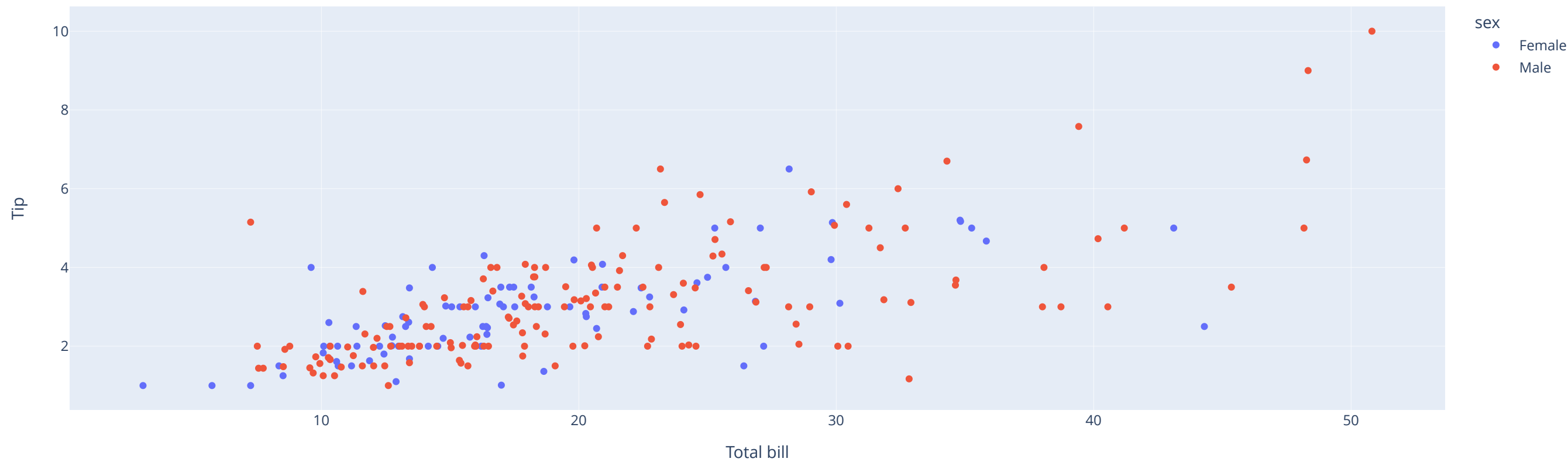
```
In [110... df = px.data.stocks()
fig = px.line(df, x='date', y=df.columns, title='Stock Market')
fig.update_layout(xaxis_title='Date', yaxis_title='Stock value')
fig.show()
```



The tips dataset

```
In [111... df = px.data.tips()
fig = px.scatter(df, x='total_bill', y='tip', title='Difference between male and female with giving tips', color='sex')
fig.update_layout(xaxis_title='Total bill', yaxis_title='Tip')
fig.show()
```

Difference between male and female with giving tips



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 [112... #Load data
df = px.data.gapminder()
df.head()
```

Out[112]:

	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 [113...

```
df_2007 = df.query('year==2007')
dfnew = df_2007.groupby('continent').sum()
fig = px.bar(dfnew, y='pop', title='Population per continent', color=dfnew.index, text_auto=True)
fig.update_xaxes(categoryorder="total ascending")
fig.update_layout(xaxis_title='Continents', yaxis_title='Population')
fig.show()
```

