

```
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 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

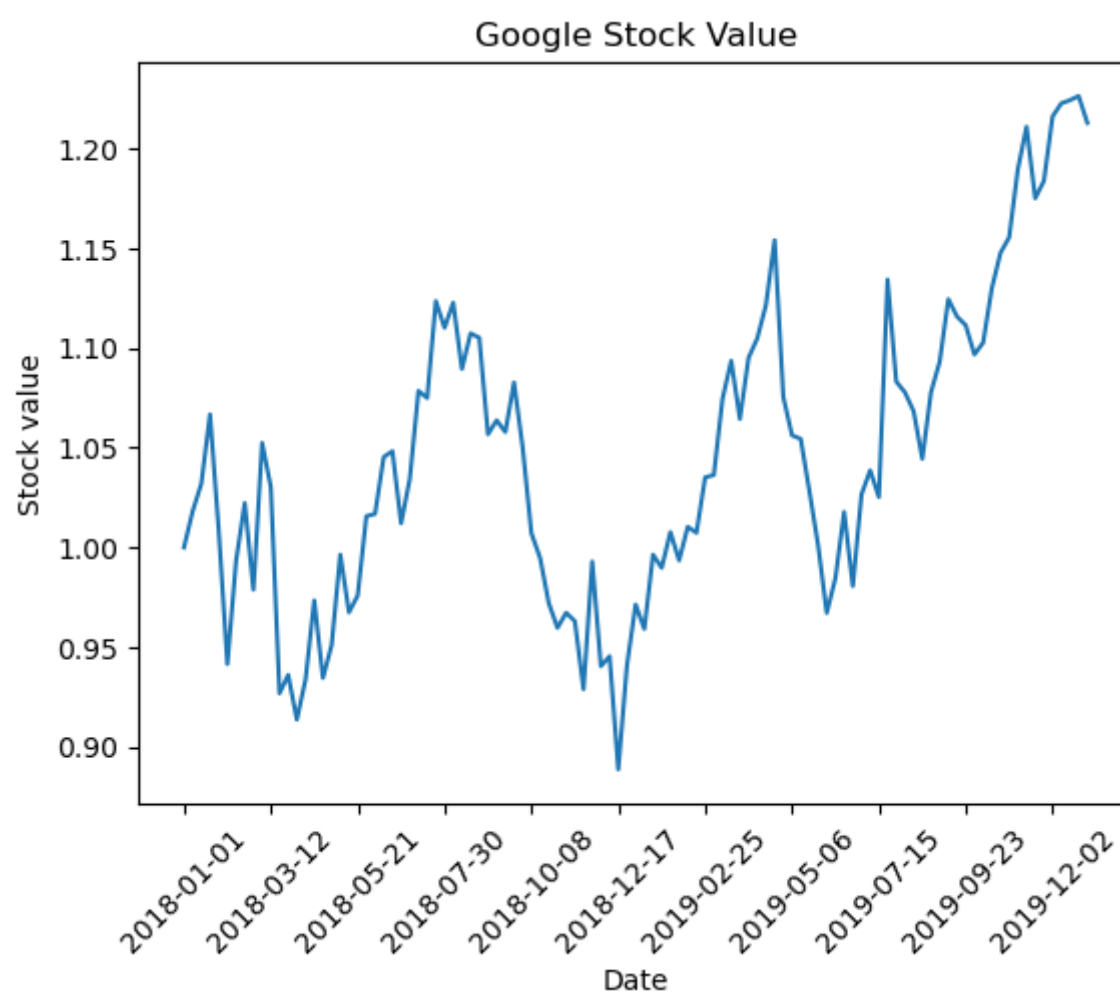
## Question 1:

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

```
In [25]: ► import numpy as np

x = stocks.date
y = stocks.GOOG

plt.xticks(rotation=45)
plt.xticks(np.arange(0, len(x)+1, 10))
plt.title("Google Stock Value")
plt.ylabel("Stock value")
plt.xlabel("Date")
plt.plot(x,y)
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 [33]: ► import numpy as np

x = stocks.date
y1 = stocks.GOOG
y2 = stocks.AAPL
y3 = stocks.FB

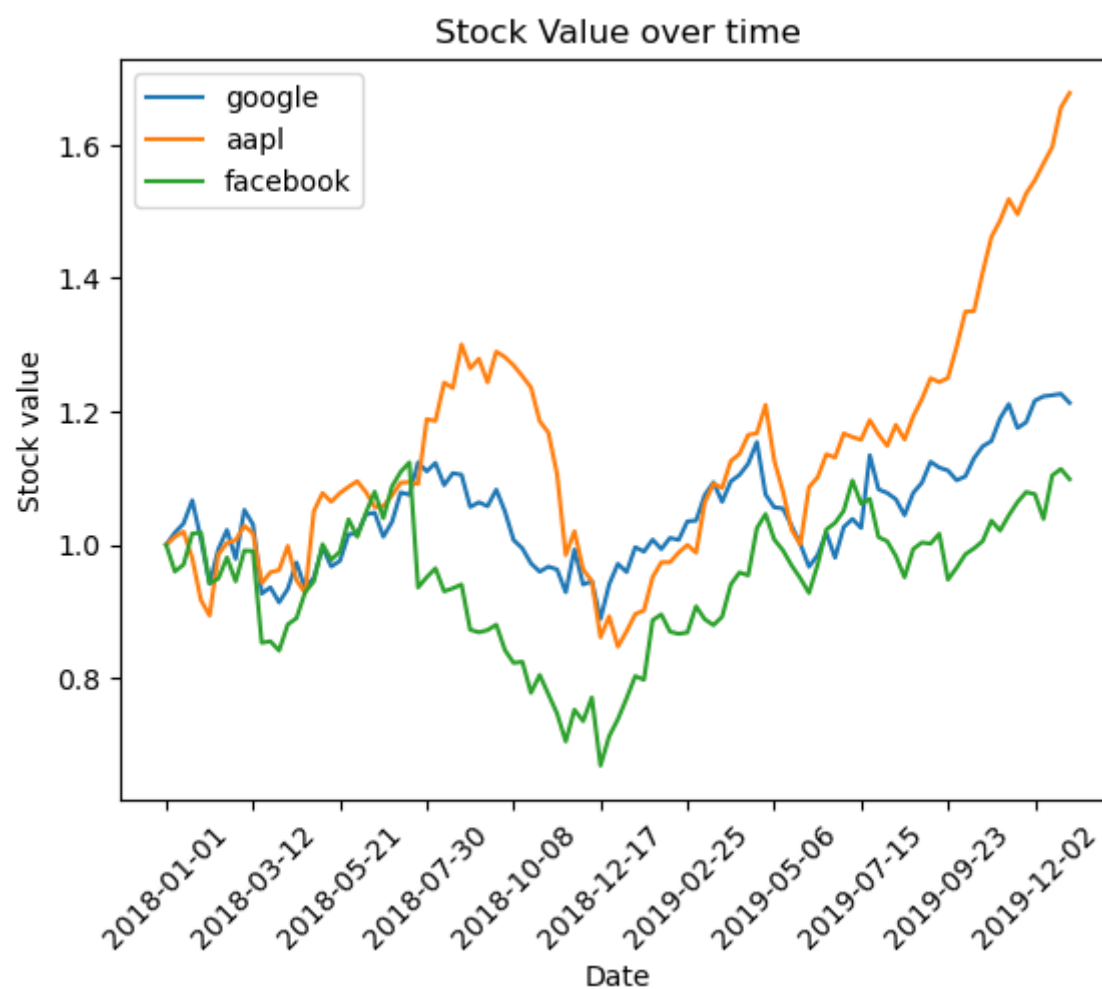
plt.xticks(rotation=45)
plt.xticks(np.arange(0, len(x)+1, 10))

plt.title("Stock Value over time")
plt.ylabel("Stock value")
plt.xlabel("Date")

plt.plot(x,y1, label="google")
plt.plot(x,y2, label="aapl")
plt.plot(x,y3, label="facebook")

plt.legend(["google", "aapl", "facebook"])
```

Out[33]: <matplotlib.legend.Legend at 0x2ce157a0820>



## Seaborn

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

```
In [34]: ► tips = sns.load_dataset('tips')
tips.head()
```

Out[34]:

	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

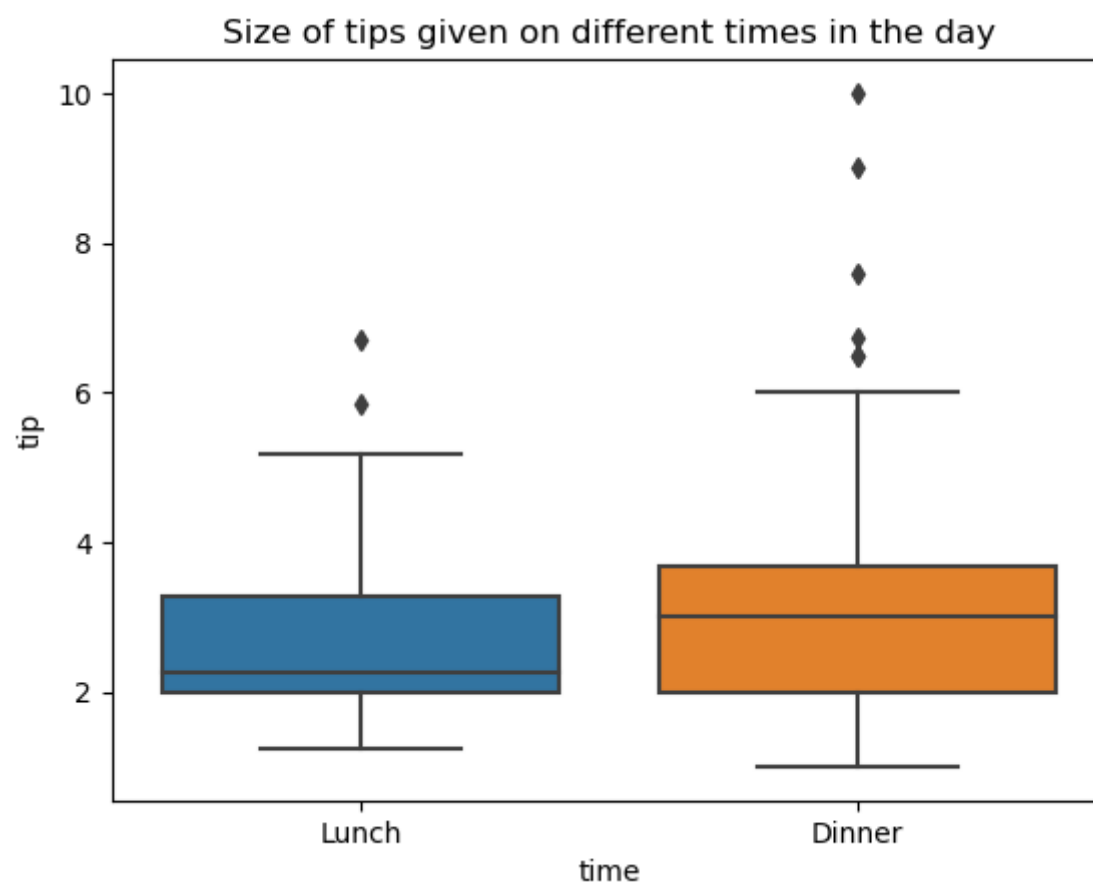
## Question 3:

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 [40]: #Question: What influence does the time of day have on the average tip given?  
  
sns.boxplot(x='time', y='tip', data=tips).set(title='Size of tips given on different times in the day')  
# fig.savefig('iris_petal_width_dist.png', dpi=200)  
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:*

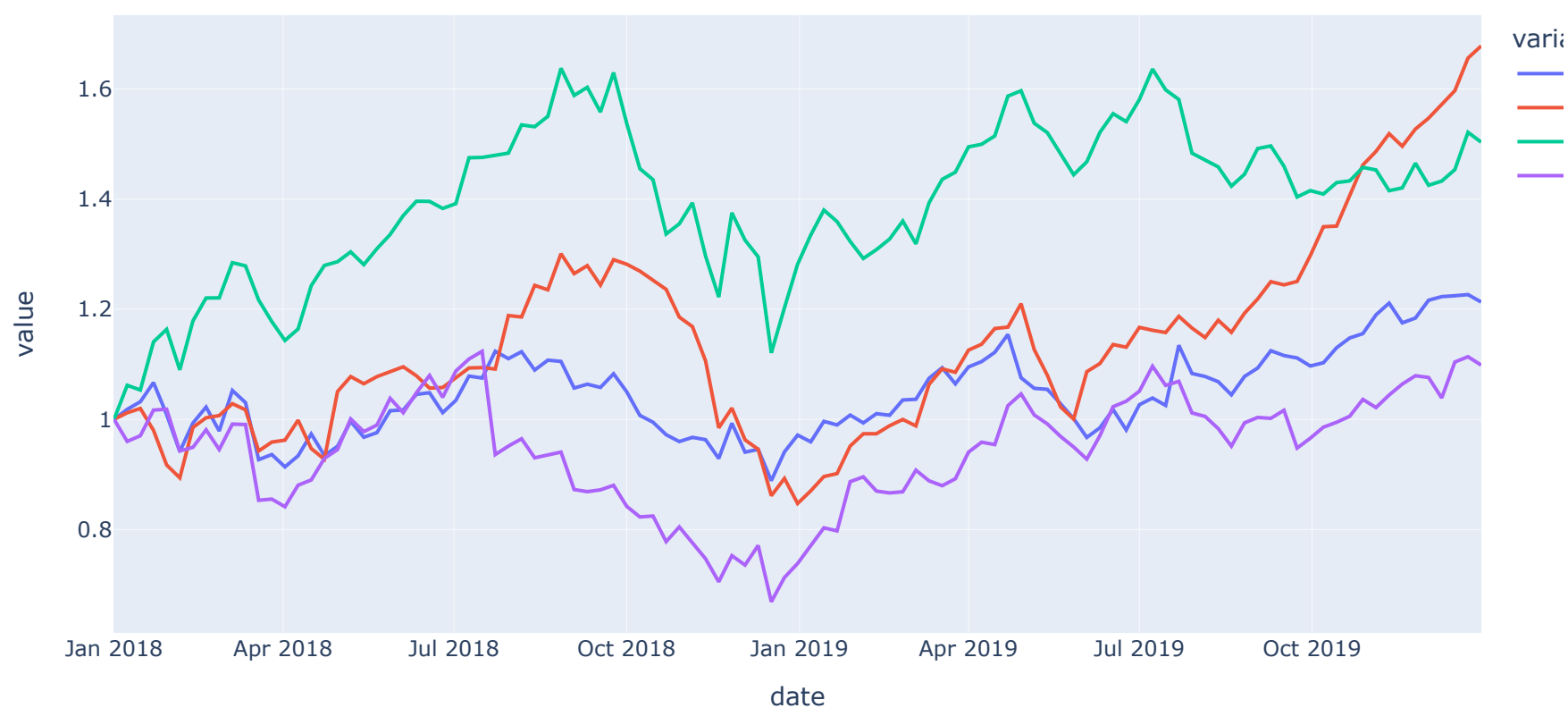
- Turn stocks dataframe into a structure that can be picked up easily with plotly express

```
In [4]: # YOUR CODE HERE
stocks = px.data.stocks()
stocks.head()

fig = px.line(stocks, x="date", y=["GOOG", "AAPL", "AMZN", "FB"], title='Stock value of different companies over time')

fig.show()
```

Stock value of different companies over time

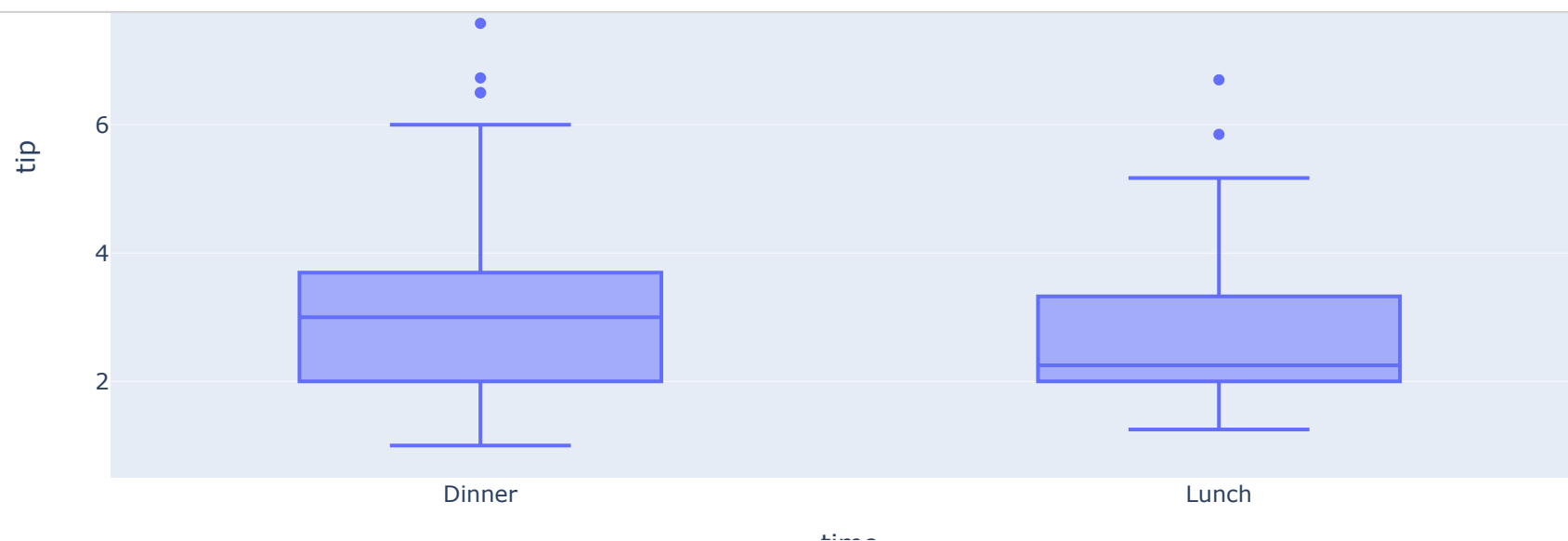


## The tips dataset

```
In [19]: tips = sns.load_dataset('tips')
tips.head()

fig = px.box(tips, x="time", y="tip", title='Stock value of different companies over time')

fig.show()
```



## ## Question 5:

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

*\*Hints:\**

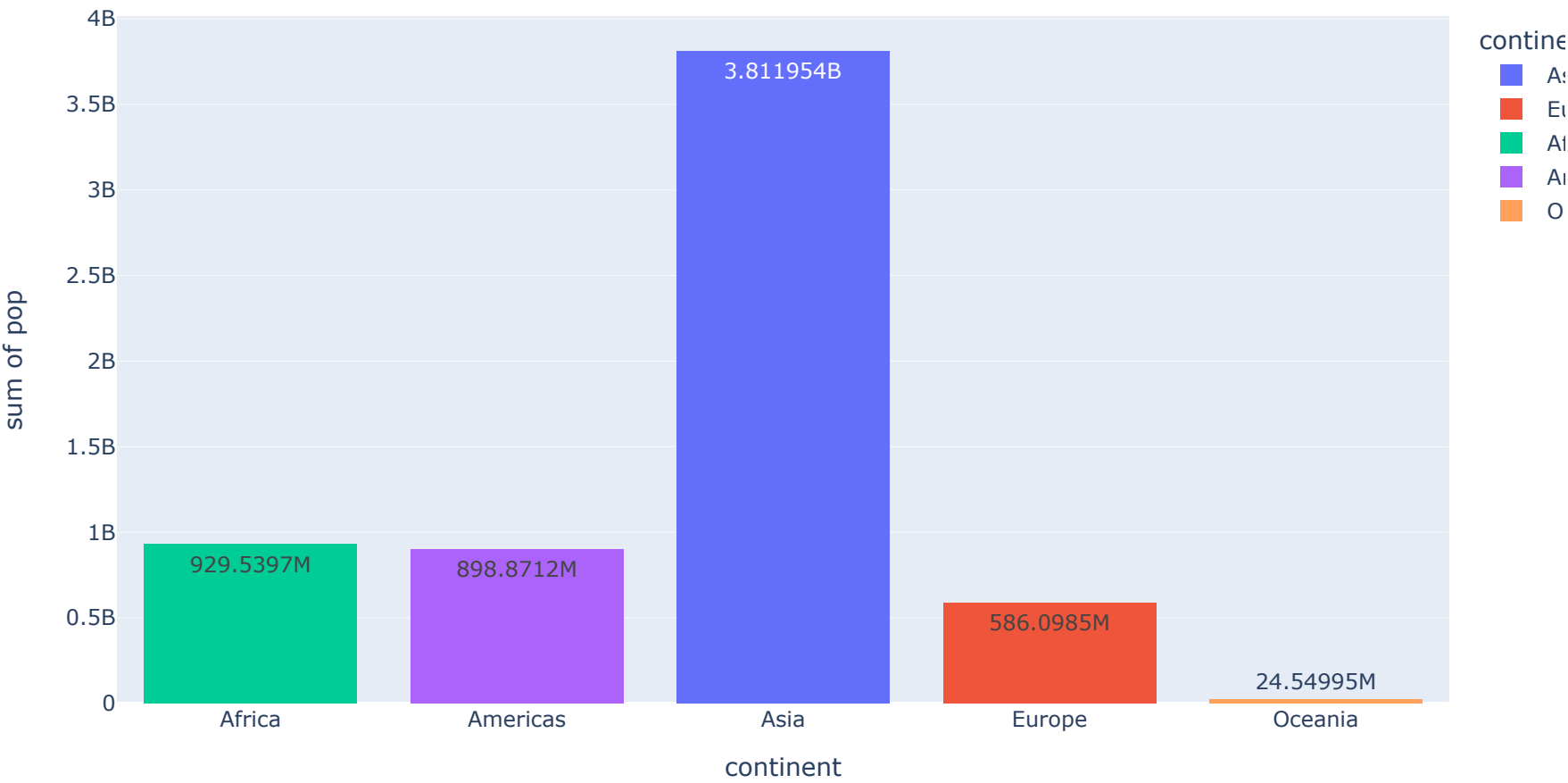
- Extract the 2007 year data from the dataframe. You have to process the data accordingly
- use `[plotly_bar](https://plotly.com/python-api-reference/generated/plotly.express.bar)`
- Add different colors for different continents
- Sort the order of the continent for the visualisation. Use `[axis layout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

```
In [17]: #Load data
df = px.data.gapminder().query("year == 2007")
df.head()

fig = px.histogram(df, x='continent', y='pop', color='continent', text_auto=True)

fig.update_xaxes(categoryorder="category ascending")

fig.show()
```



```
In [13]: # YOUR CODE HERE
```

```
In [ ]:
```