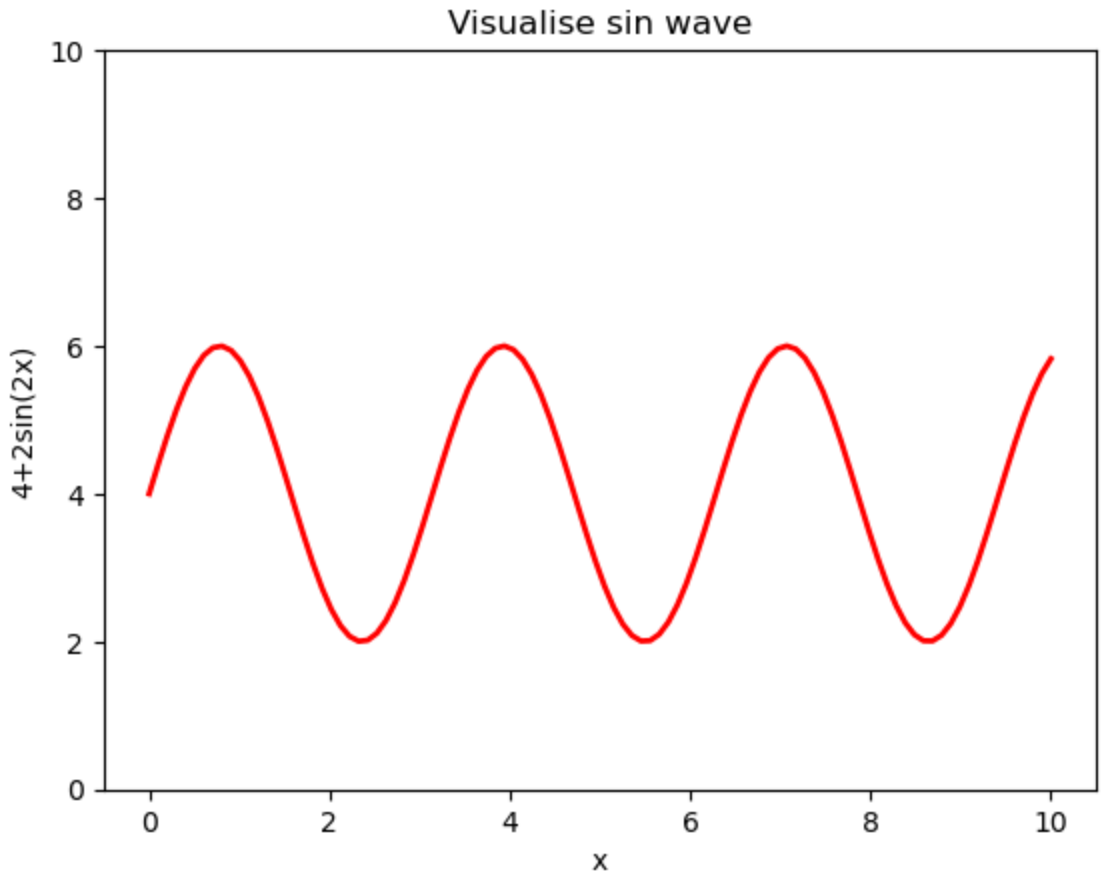
In [34]:
```
import pandas as pd
import seaborn as sns
import plotly.express as px
import numpy as np

import matplotlib.pyplot as plt
```

In [2]:
```
import plotly.io as pio
pio.renderers.default = "plotly_mimetype+notebook"
```

In [33]:
```
nmb = [1, -2, 3, -4]

res = [ele for ele in nmb if ele > 0]
dif = max(res)-min(res)

print('The difference between highest and lowest number is', dif)
```

```
The difference between highest and lowest number is 2
```

In [57]:
```
x = np.linspace(0, 10, 100)
y= 4 + 2 * np.sin(2 * x)

fig, ax = plt.subplots()
ax.plot(x, y, linewidth=2.0, color='red')
plt.title('Visualise sin wave')
ax.set_ylabel('4+2sin(2x)')
ax.set_xlabel('x')
plt.yticks([0, 2, 4, 6, 8, 10])
plt.show()
```



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()
```

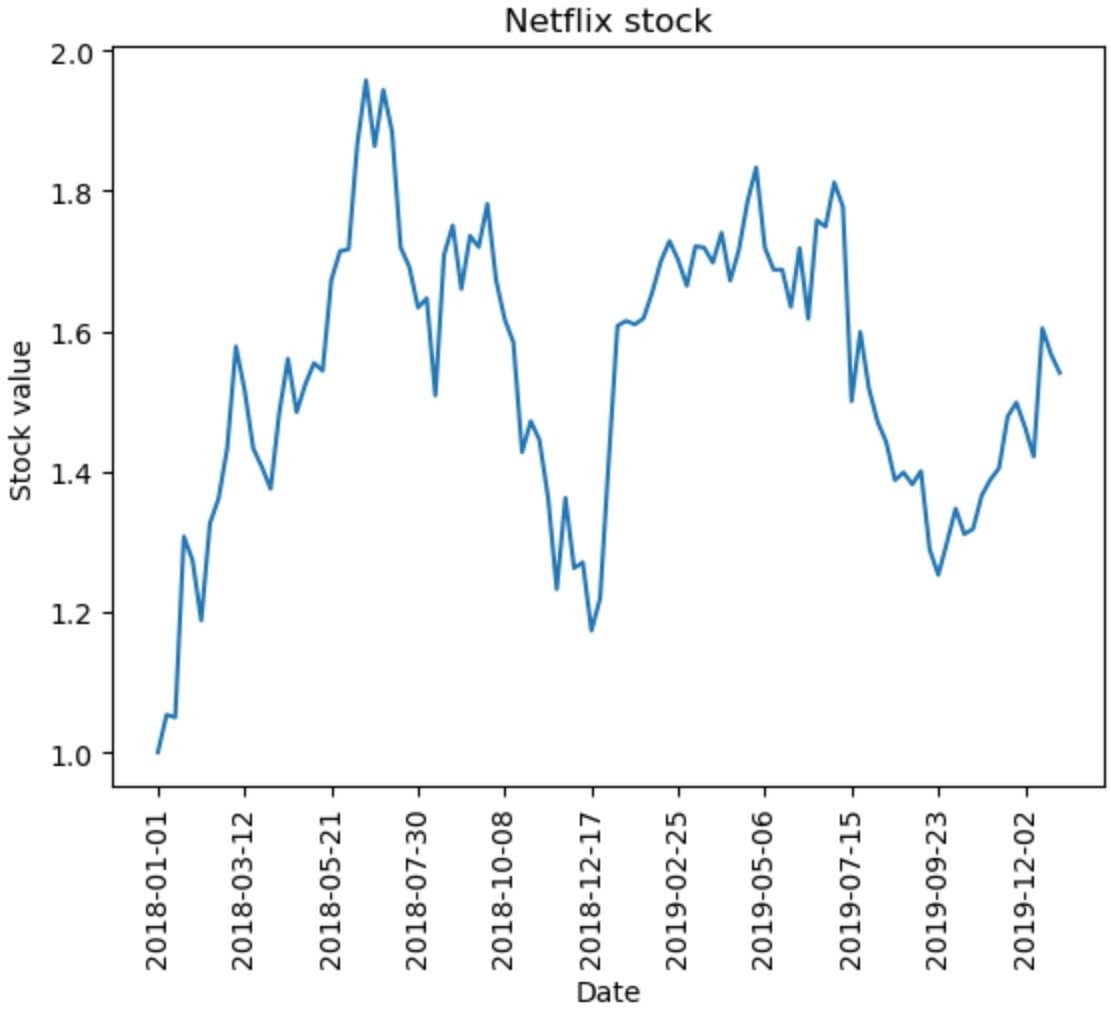
Out[3]:

	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.

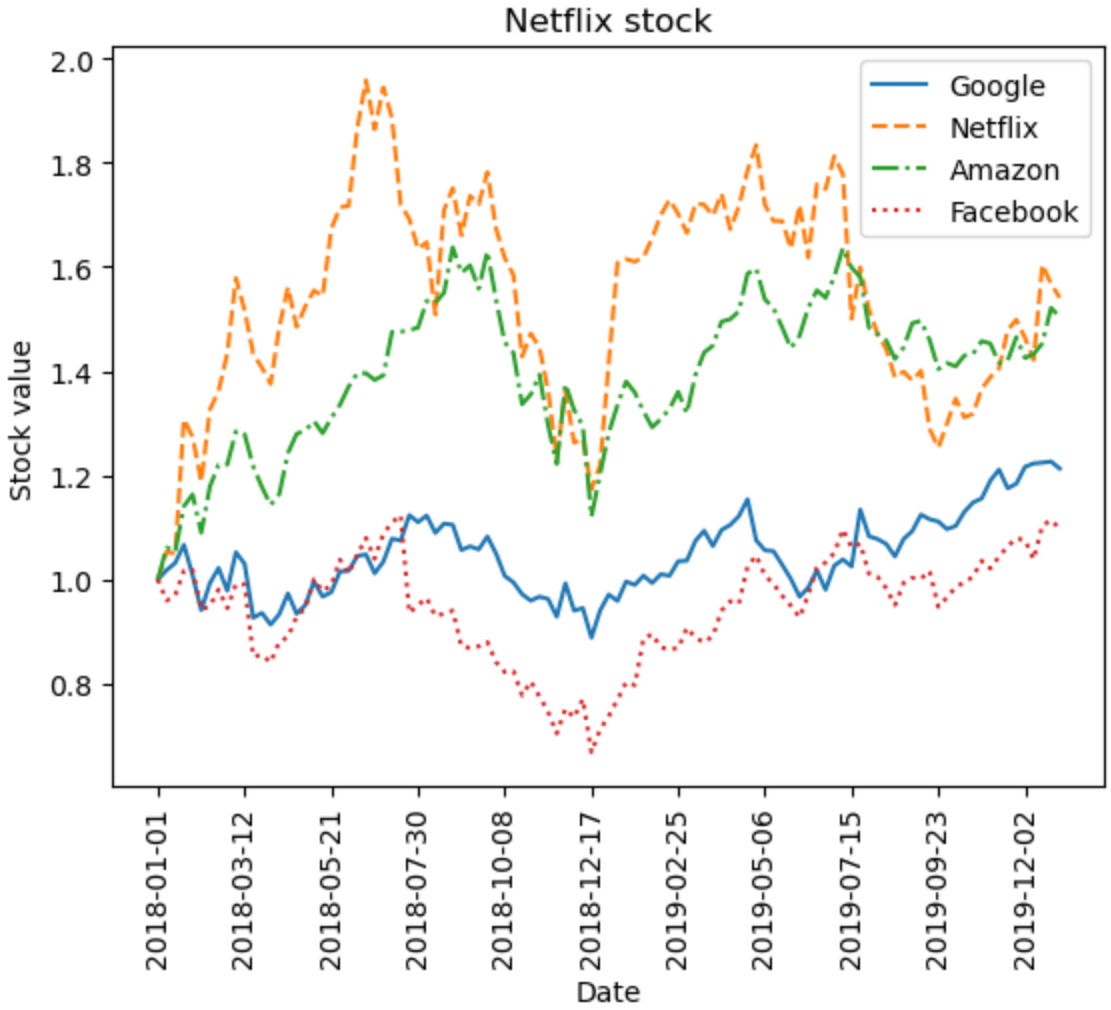
In [4]:
```
# Plot Netflix stock
plt.plot(stocks.date.values, stocks.NFLX.values)

# Make plot on x-axis readable
plt.xticks(rotation='vertical')
plt.xticks(range(0, stocks.shape[0], 10))

# Add titles and labels
plt.title('Netflix stock')
plt.xlabel('Date')
plt.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 [5]:
```
# Plot more than one stock (not all)
plt.plot(stocks.date.values, stocks.GOOG.values, label = "Google", linestyle="-")
plt.plot(stocks.date.values, stocks.NFLX.values, label = "Netflix", linestyle="--")
plt.plot(stocks.date.values, stocks.AMZN.values, label = "Amazon", linestyle="-.")
plt.plot(stocks.date.values, stocks.FB.values, label = "Facebook", linestyle=":")

# Make plot on x-axis readable
plt.xticks(rotation='vertical')
plt.xticks(range(0, stocks.shape[0], 10))

# Add titles, labels and legend
plt.title('Netflix stock')
plt.xlabel('Date')
plt.ylabel('Stock value')
plt.legend()

plt.show()
```



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

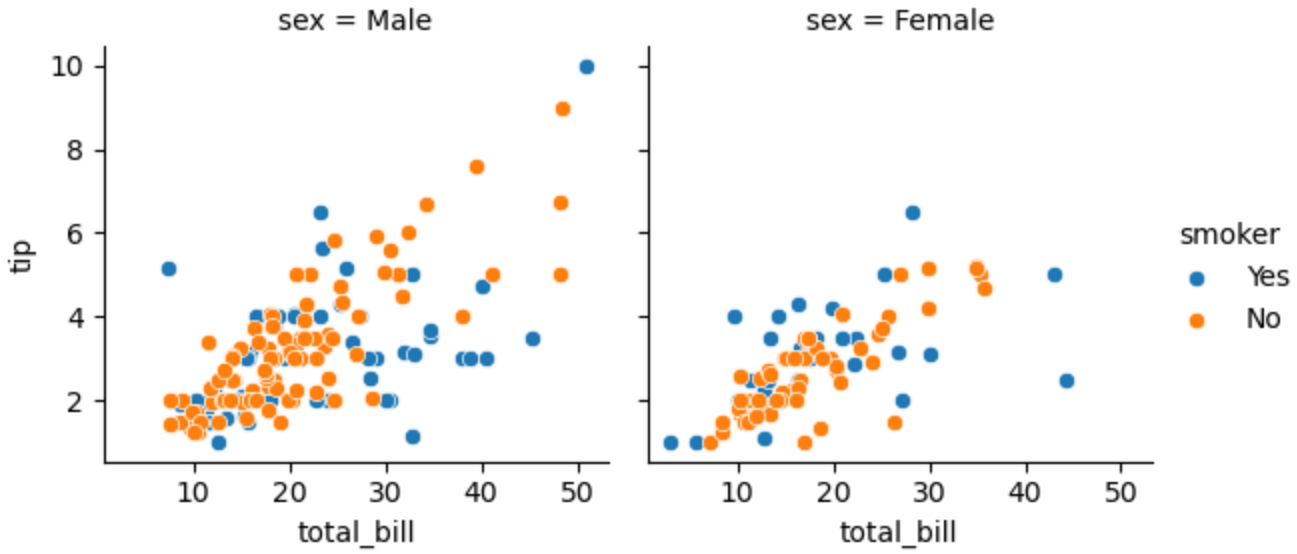
Question 3:

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

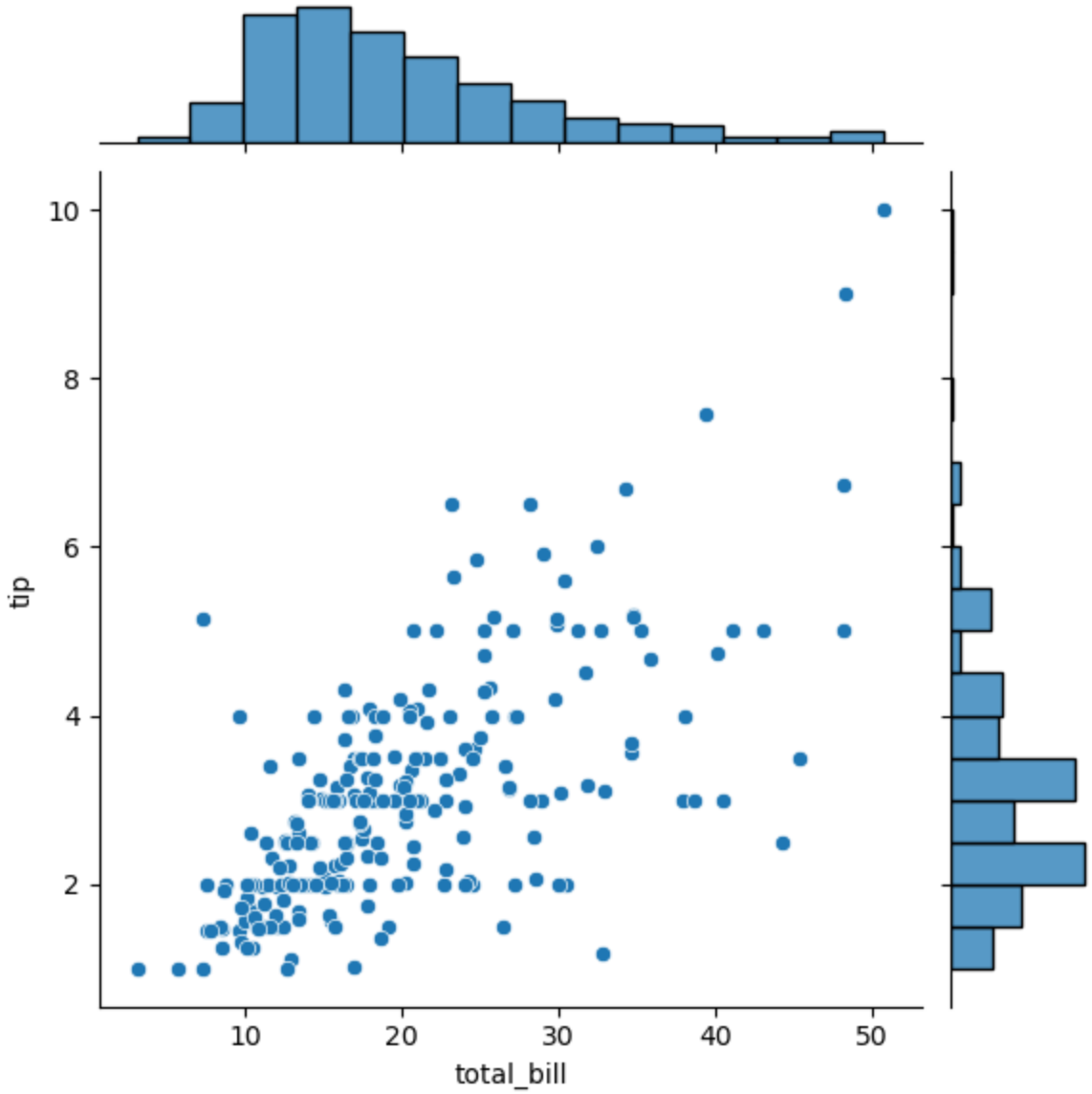
Some possible questions:

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

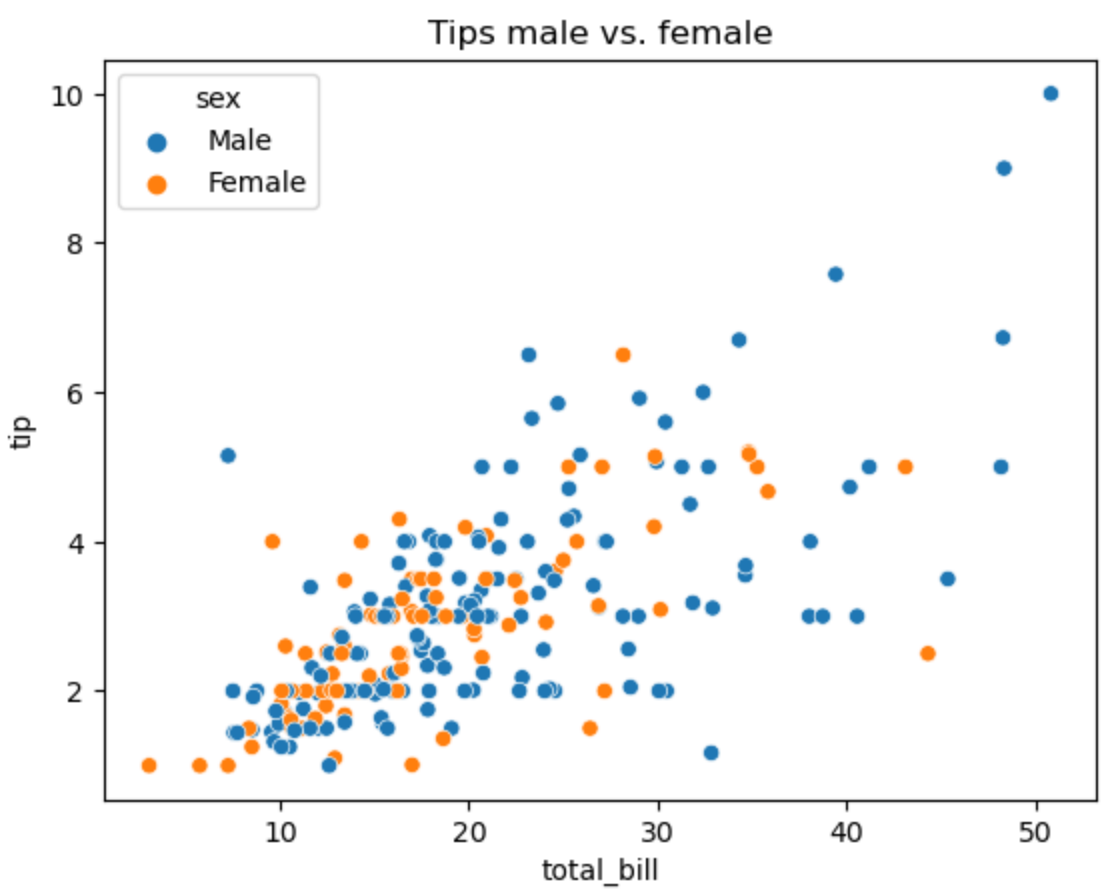
In [7]:
```
g = sns.FacetGrid(tips, col='sex', hue='smoker')
g.map(sns.scatterplot, 'total_bill', 'tip')
g.add_legend()

plt.show()
```



In [8]:
```
sns.jointplot(x='total_bill', y='tip', data=tips)
plt.show()
```



In [9]:
```
sns.scatterplot(x='total_bill', y='tip', data=tips, hue='sex').set(title='Tips male vs.
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:

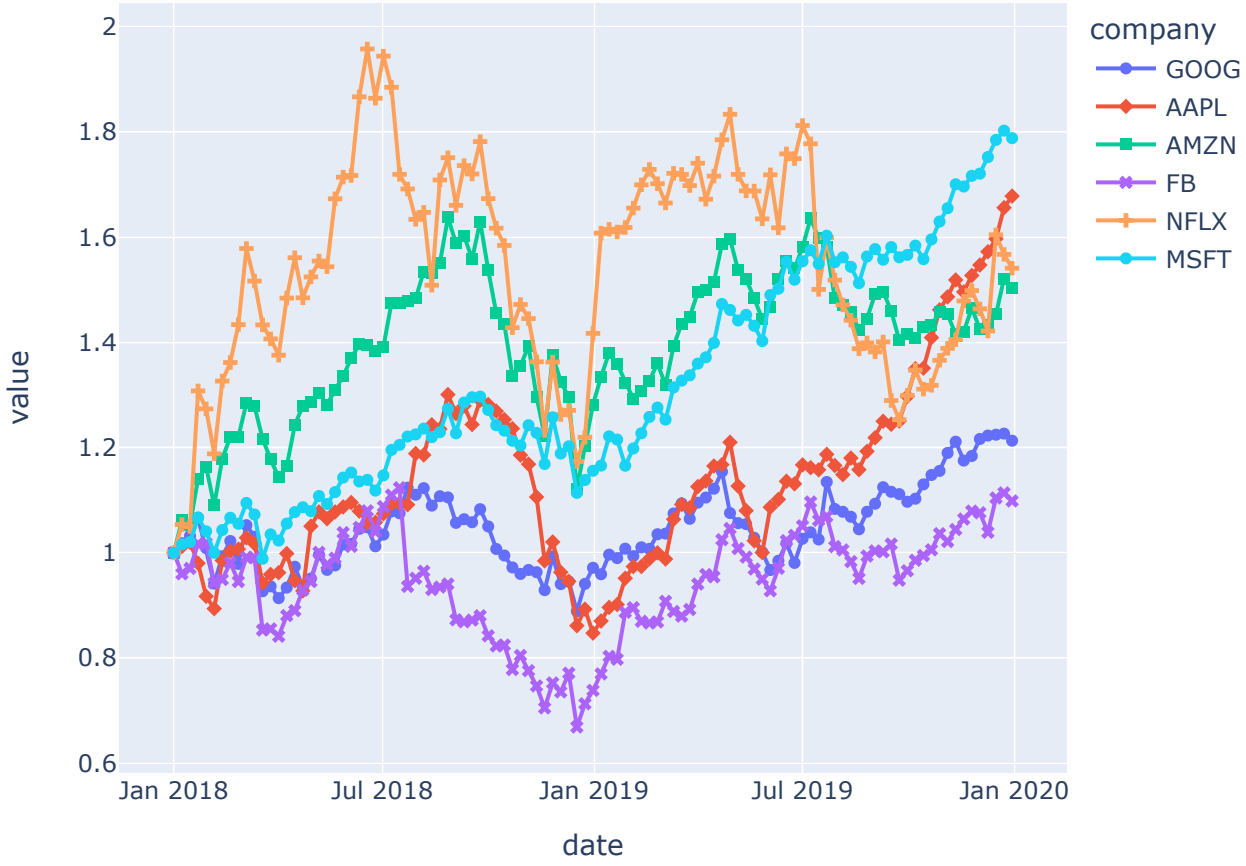
- Turn stocks dataframe into a structure that can be picked up easily with plotly express

In [10]:
```
df_stocks = stocks.melt(id_vars=['date'], var_name='company')
df_stocks.head()
```

Out[10]:

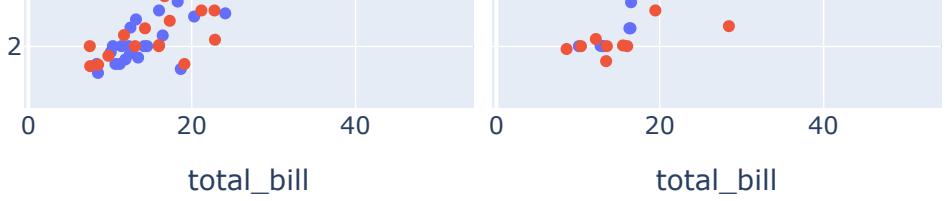
	date	company	value
0	2018-01-01	GOOG	1.000000
1	2018-01-08	GOOG	1.018172
2	2018-01-15	GOOG	1.032008
3	2018-01-22	GOOG	1.066783
4	2018-01-29	GOOG	1.008773

In [11]:
```
px.line(df_stocks, 'date', 'value', color='company', symbol='company')
```



The tips dataset

In [12]:
```
fig = px.scatter(tips, x='total_bill', y='tip', color='sex', facet_col='smoker',
                 facet_row='time')
fig.show()
```



Question 5:

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

Hints:

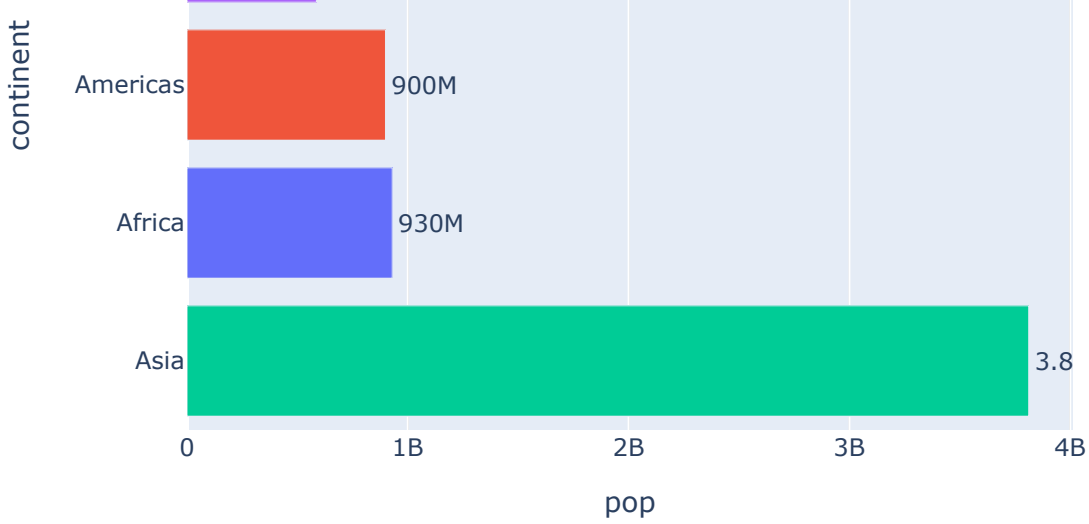
- Extract the 2007 year data from the dataframe. You have to process the data accordingly
- use plotly bar
- Add different colors for different continents
- Sort the order of the continent for the visualisation. Use axis layout setting
- Add text to each bar that represents the population

In [13]:
```
#load data
df = px.data.gapminder()
df.head()
```

Out[13]:

	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 [14]:
```
df_2007 = df.query('year==2007')
df_2007_new = df_2007.groupby('continent').sum()
fig = px.bar(df_2007_new, x='pop', y=df_2007_new.index, color=df_2007_new.index,
             text='pop', title="Population of different continents for the year 2007",
             text_auto='.2s')
fig.update_layout(yaxis={'categoryorder':'total descending'})
fig.update_traces(textposition='outside')

fig.show()
```


continent
Americas
900M
Africa
930M
Asia
3.8
0
1B
2B
3B
4B
pop