In [1]:

```
import pandas as pd
import seaborn as sns
import plotly.express as px

import matplotlib.pyplot as plt
```

```
C:\Users\linde\anaconda3\envs\TIL6022\lib\site-packages\scipy\__init__.py:146: UserWarni
ng: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected
version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

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]:

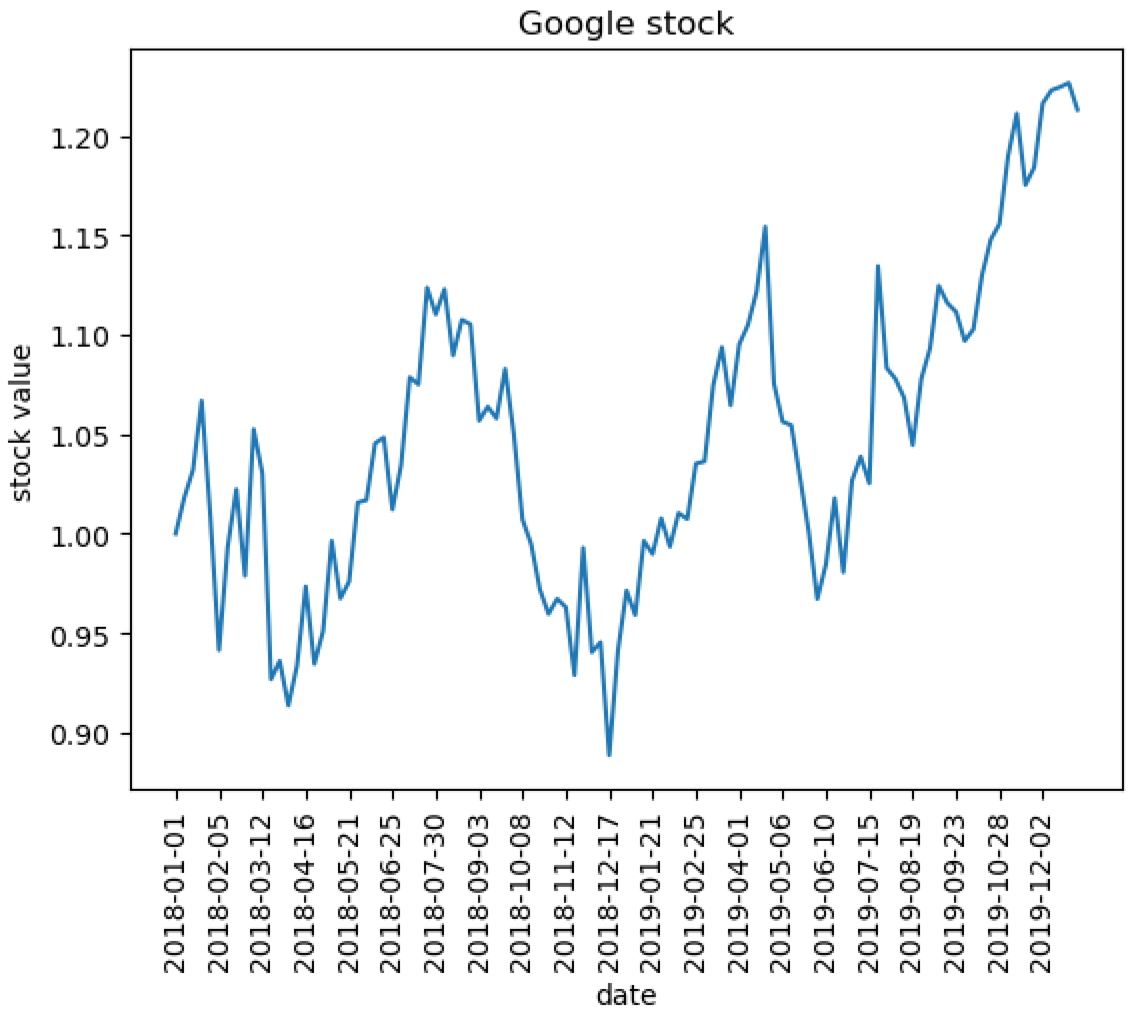
	date	GOOG	AAPL	AMZN	FB	NFLX	MSFT
0	2018-01-01	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1	2018-01-08	1.018172	1.011943	1.061881	0.959968	1.053526	1.015988
2	2018-01-15	1.032008	1.019771	1.053240	0.970243	1.049860	1.020524
3	2018-01-22	1.066783	0.980057	1.140676	1.016858	1.307681	1.066561
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]:

```
# YOUR CODE HERE
import numpy as np
x = stocks['date']
y = stocks['GOOG']
plt.plot(x,y)
plt.title('Google stock')
plt.xlabel('date')
plt.ylabel('stock value')
ticks = list(stocks['date'])
plt.xticks([ticks[i] for i in range(len(ticks)) if i % 5 == 0], rotation = 90)
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.

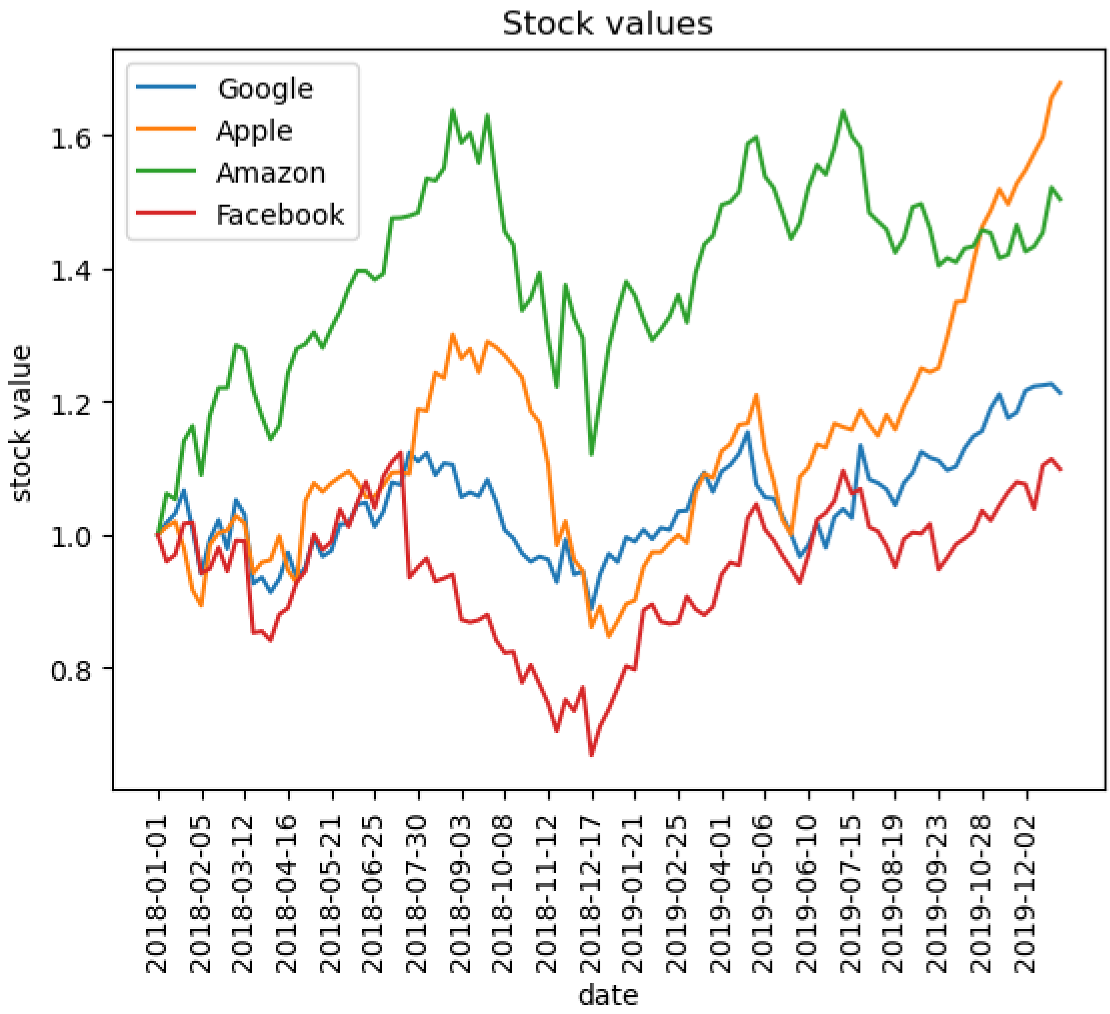
In [5]:

```
# YOUR CODE HERE
fig, ax = plt.subplots()
x = stocks.date
y1 = stocks.GOOG
y2 = stocks.AAPL
y3 = stocks.AMZN
y4 = stocks.FB

plt.plot(x, y1, label='Google')
plt.plot(x, y2, label='Apple')
plt.plot(x, y3, label='Amazon')
plt.plot(x, y4, label='Facebook')

plt.title('Stock values')
plt.xlabel('date')
plt.ylabel('stock value')

ticks = list(stocks['date'])
plt.xticks([ticks[i] for i in range(len(ticks)) if i % 5 == 0], rotation = 90)
plt.legend(loc='best')
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

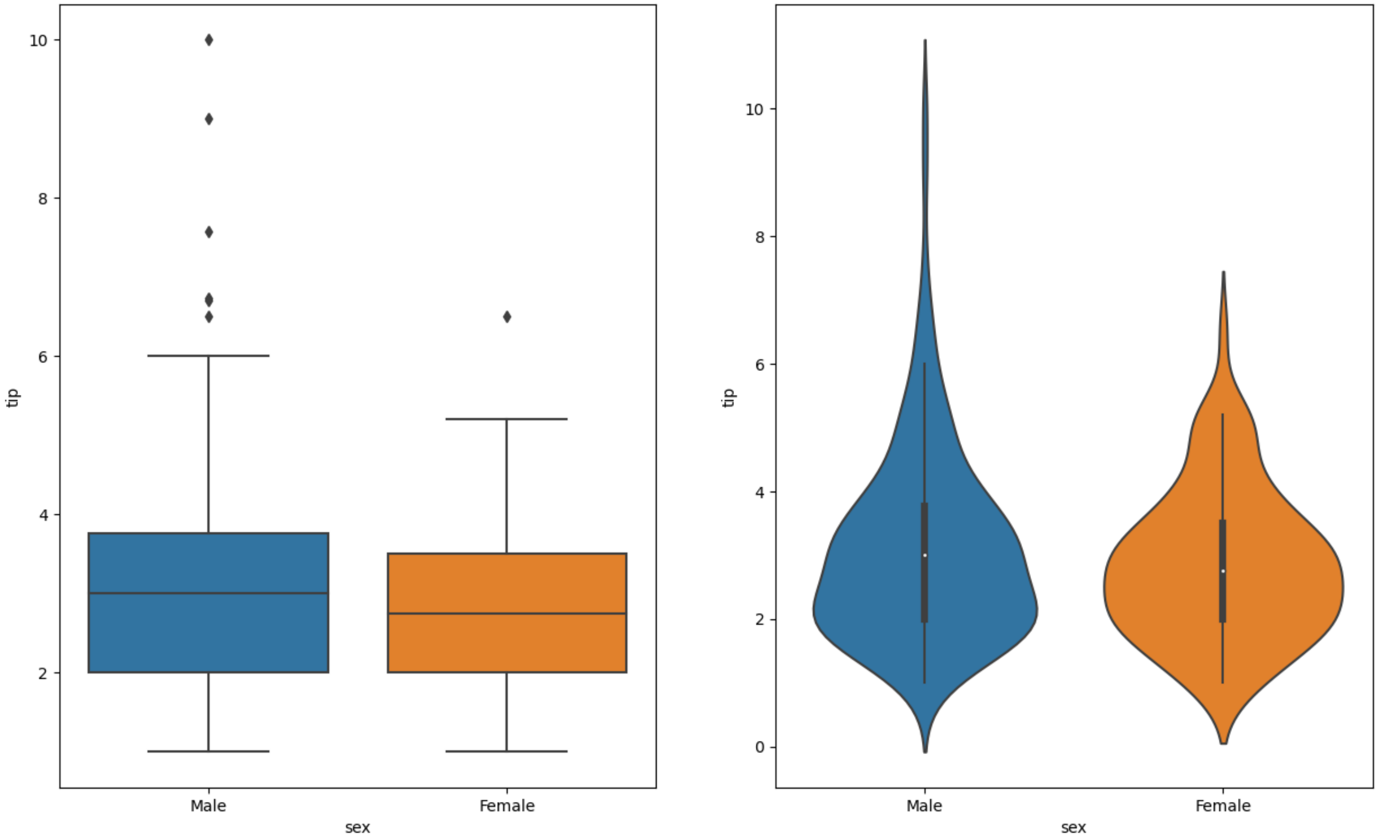
Question 3:

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]:
```
# YOUR CODE HERE
# Question: Are there differences between male and female when it comes to giving tips?
fig, ax = plt.subplots(ncols=2, figsize=(15,9))
sns.boxplot(x='sex', y='tip', data=tips, ax=ax[0])
sns.violinplot(x='sex', y='tip', data=tips, ax=ax[1])
plt.show()
```



Plotly Express

Question 4:

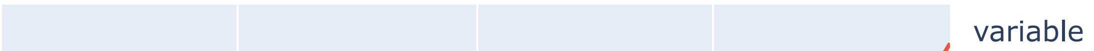
Redo the above exercises (challenges 2 & 3) with plotly express. Create diagrams which you can interact with.

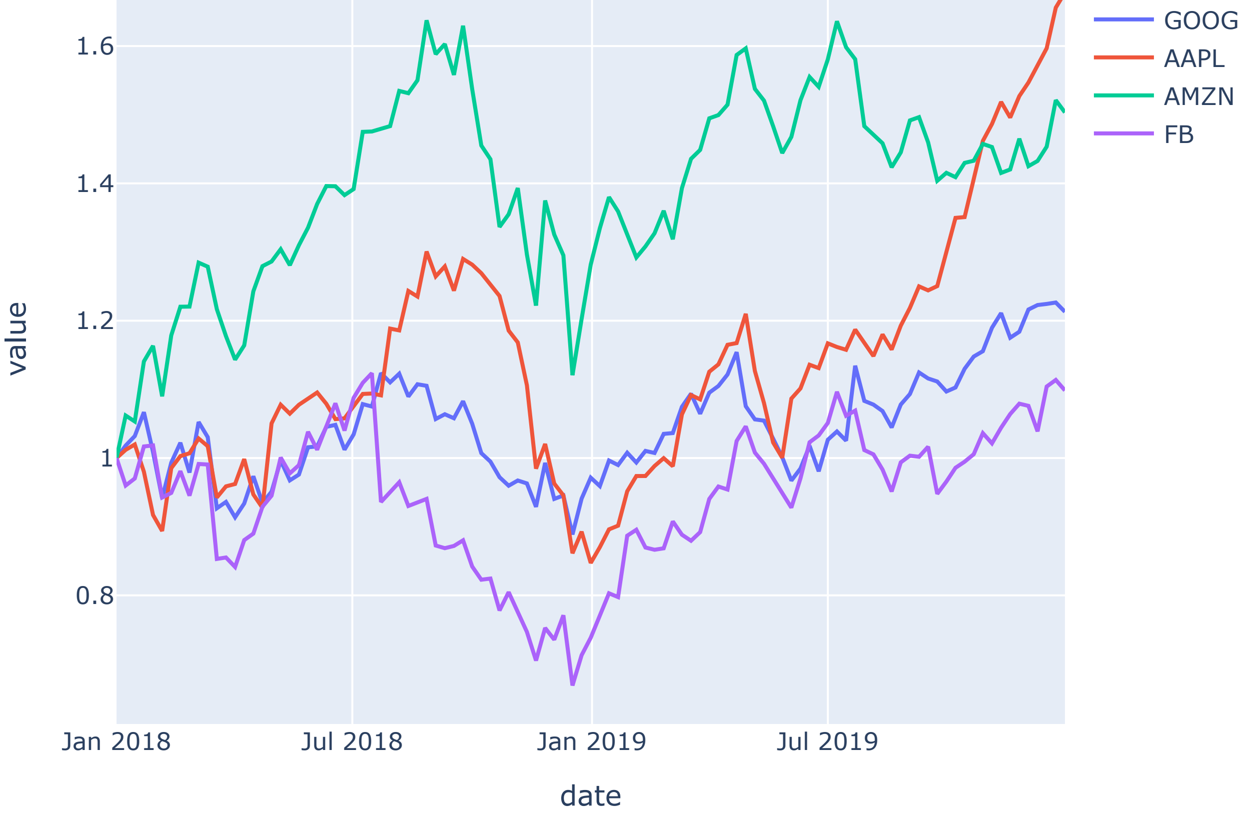
The stocks dataset

Hints:

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

In [8]:
```
# YOUR CODE HERE
df = px.data.stocks()
fig = px.line(df, x='date', y=['GOOG', 'AAPL', 'AMZN', 'FB'])
fig.show()
```

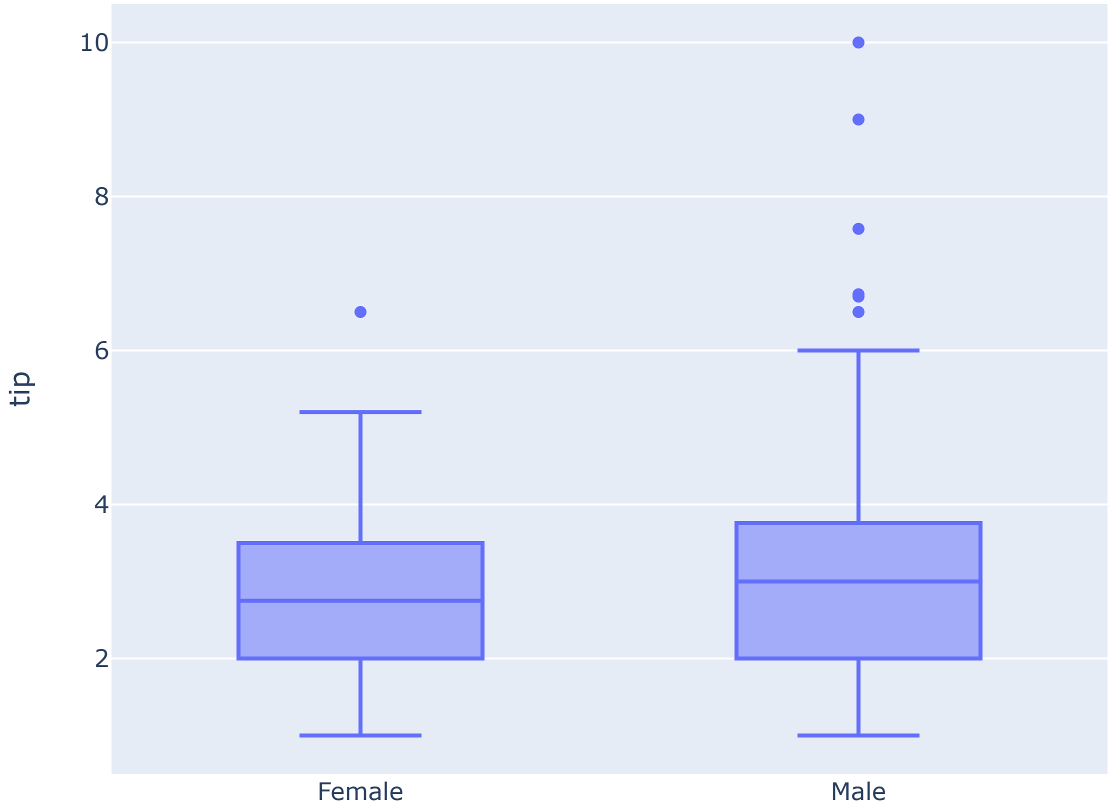




The tips dataset

In [9]:

```
# YOUR CODE HERE
df = px.data.tips()
fig = px.box(df, x='sex', y='tip')
fig.show()
```



sex

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 [10]:

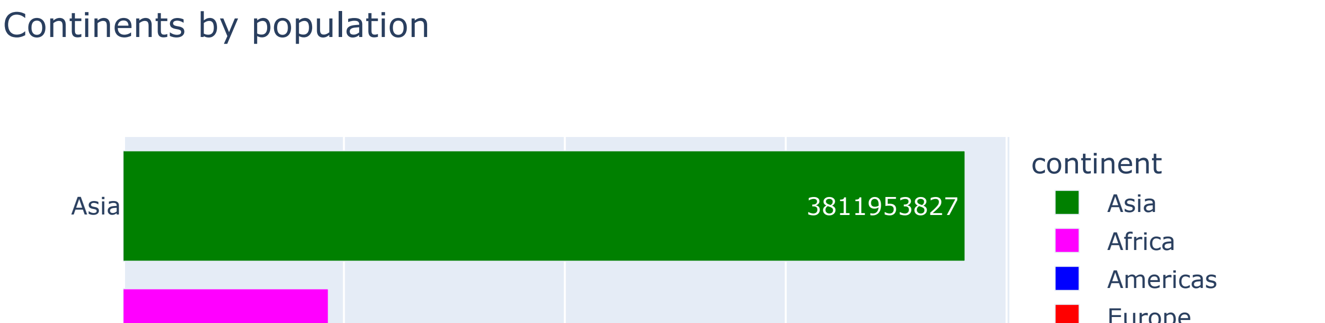
```
#load data
df = px.data.gapminder()
df.head()
```

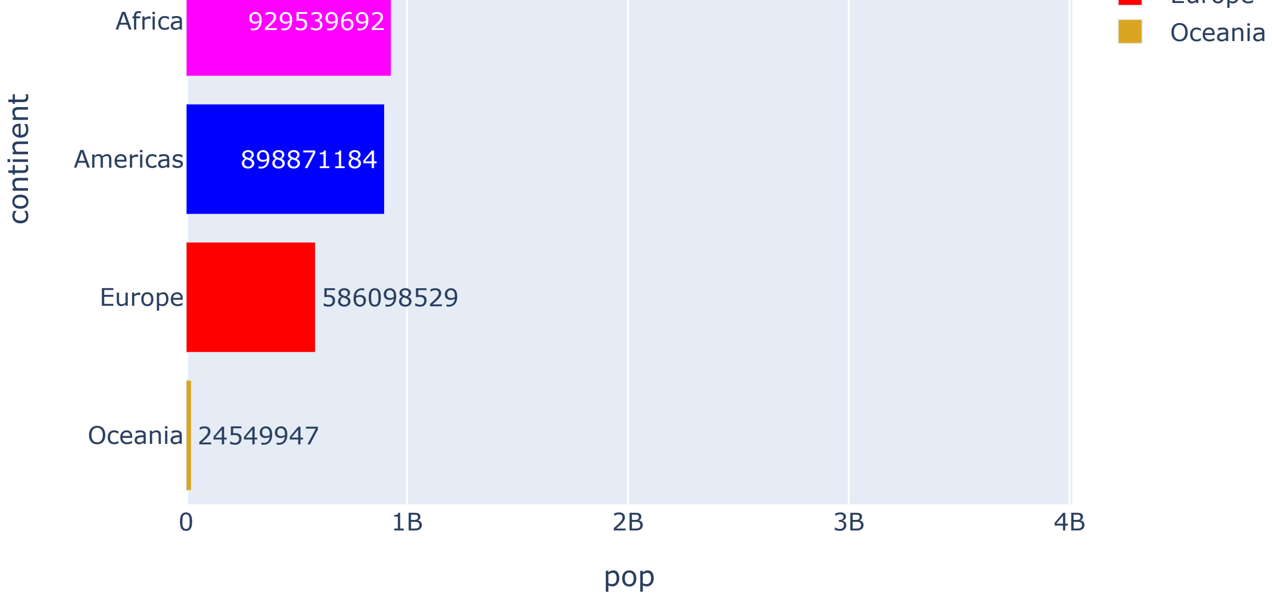
Out[10]:

	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]:

```
# YOUR CODE HERE
df = px.data.gapminder()
df_2007 = df.query('year==2007')
df_2007_new = df_2007.groupby('continent').sum()
df_2007_new = df_2007_new.reset_index()
fig = px.bar(df_2007_new, y = 'continent', x = 'pop', color = 'continent', orientation =
             color_discrete_map={
                 "Europe": "red",
                 "Asia": "green",
                 "Americas": "blue",
                 "Oceania": "goldenrod",
                 "Africa": "magenta"},
             category_orders={'continent': ["Asia", "Africa", "Americas", "Europe", "Oce
             title="Continents by population"
            )
fig.show()
```




Africa
929539692
Americas
898871184
Europe
586098529
Oceania
24549947
continent
0
1B
2B
3B
4B
pop
Europe
Oceania