Submission

Put the ipynb file and html file in the github branch you created in the last assignment and submit the link to the commit in brightspace

In [1]:

```
from plotly.offline import init_notebook_mode
import plotly.io as pio
import plotly.express as px

init_notebook_mode(connected=True)
pio.renderers.default = "plotly_mimetype+notebook"
```

In [2]:

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

Out[2]:

	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

Question 1:

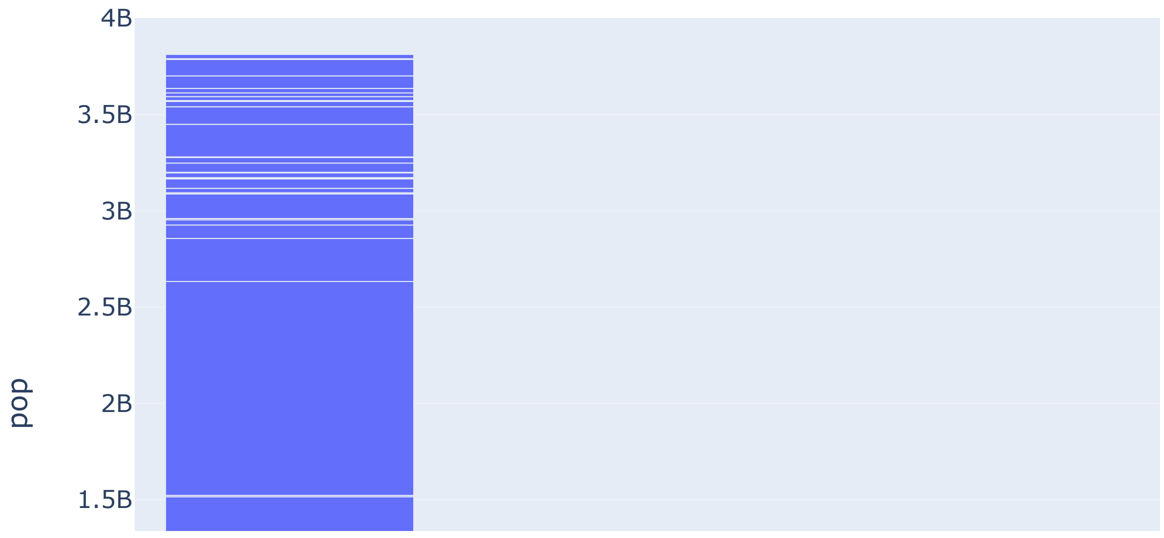
Recreate the barplot below that shows the population of different continents for the year 2007.

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 [3]:

```
# YOUR CODE HERE
'''create the barplot that shows the population of different continents for only the year 2
'''only use the rows from dataset data where year is equal to 2007'''
df_2007 = df[df['year'] == 2007]
fig = px.bar(df_2007, x="continent", y="pop", color="continent", animation_frame="year", an
fig.show()
```



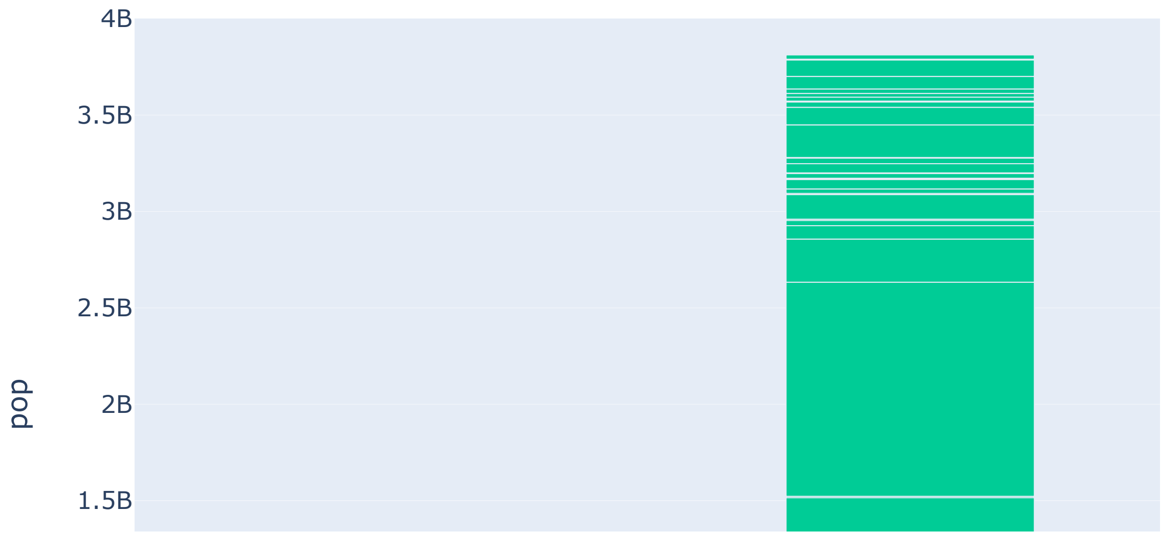
Question 2:

Sort the order of the continent for the visualisation

Hint: Use axis layout setting (https://plotly.com/python/reference/layout/xaxis/)

In [4]:

```
# YOUR CODE HERE
'''Sort the order of the continent for the visualisation'''
fig = px.bar(df_2007, x="continent", y="pop", color="continent", animation_frame="year", an
fig.show()
```

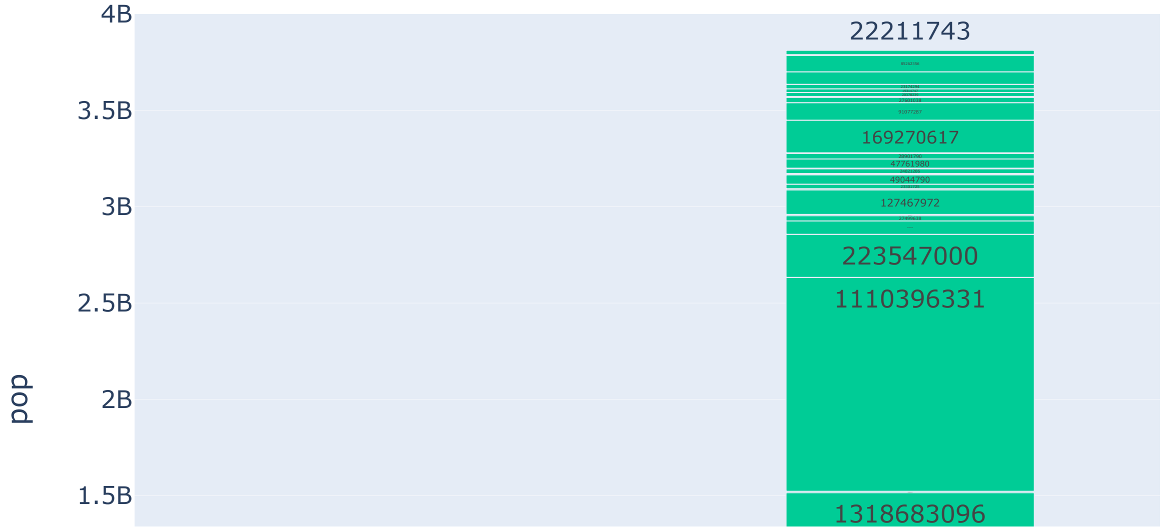


Question 3:

Add text to each bar that represents the population

```
In [5]:
```

```
# YOUR CODE HERE
'''Add text to each bar that represents the population'''
fig = px.bar(df_2007, x="continent", y="pop", color="continent", animation_frame="year", an
fig.show()
```

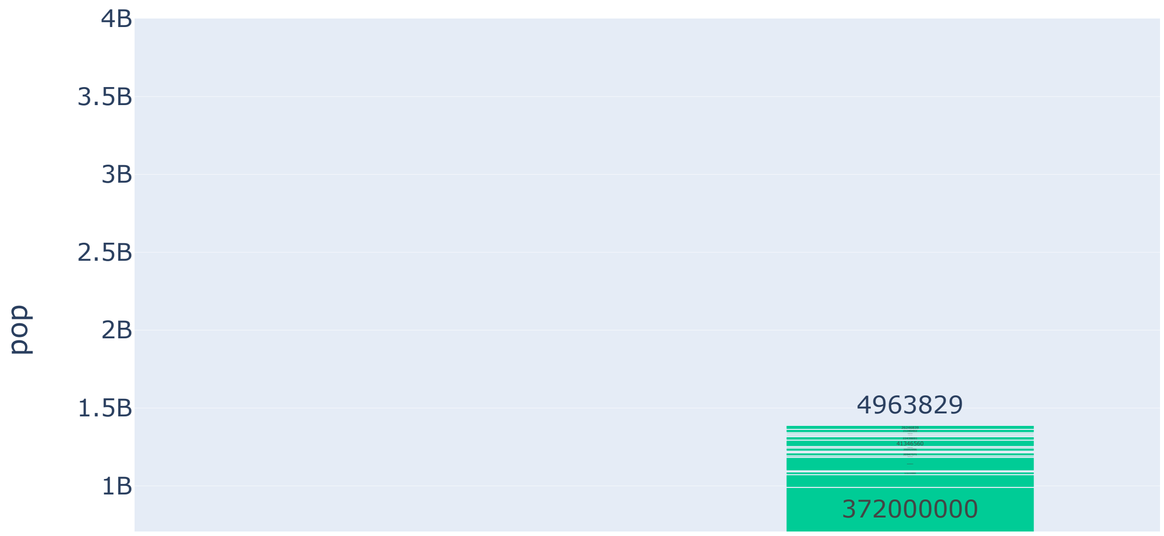


Question 4:

Thus far we looked at data from one year (2007). Lets create an animation to see the population growth of the continents through the years

In [6]:

```
# YOUR CODE HERE
'''Lets create an animation to see the population growth of the continents through the year
fig = px.bar(df, x="continent", y="pop", color="continent", animation_frame="year", animati
fig.show()
```

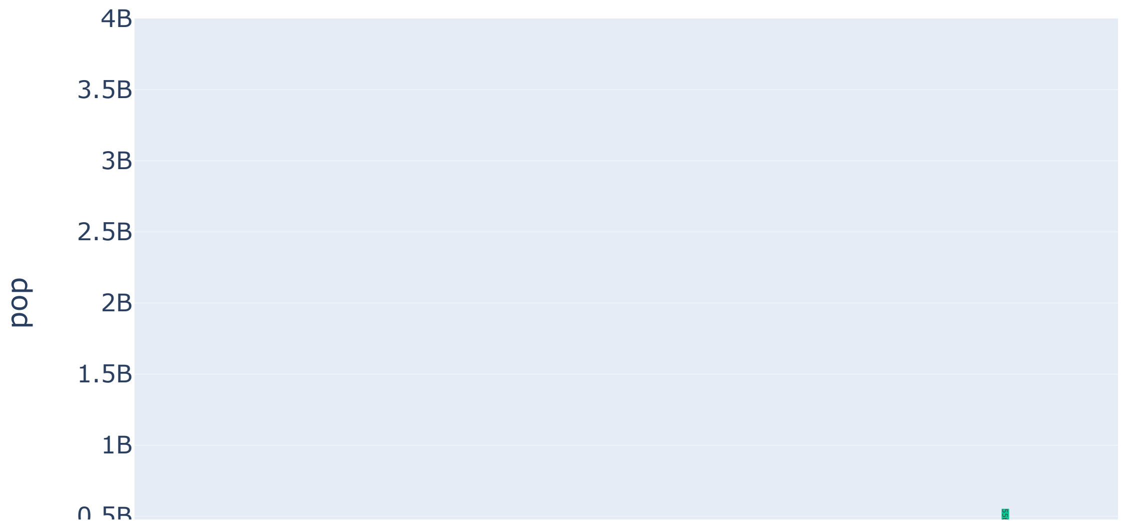


Question 5:

Instead of the continents, lets look at individual countries. Create an animation that shows the population growth of the countries through the years

In [7]:

```
# YOUR CODE HERE
'''Create an animation that shows the population growth of the countries through the years'
fig = px.bar(df, x="country", y="pop", color="continent", animation_frame="year", animation
fig.show()
```

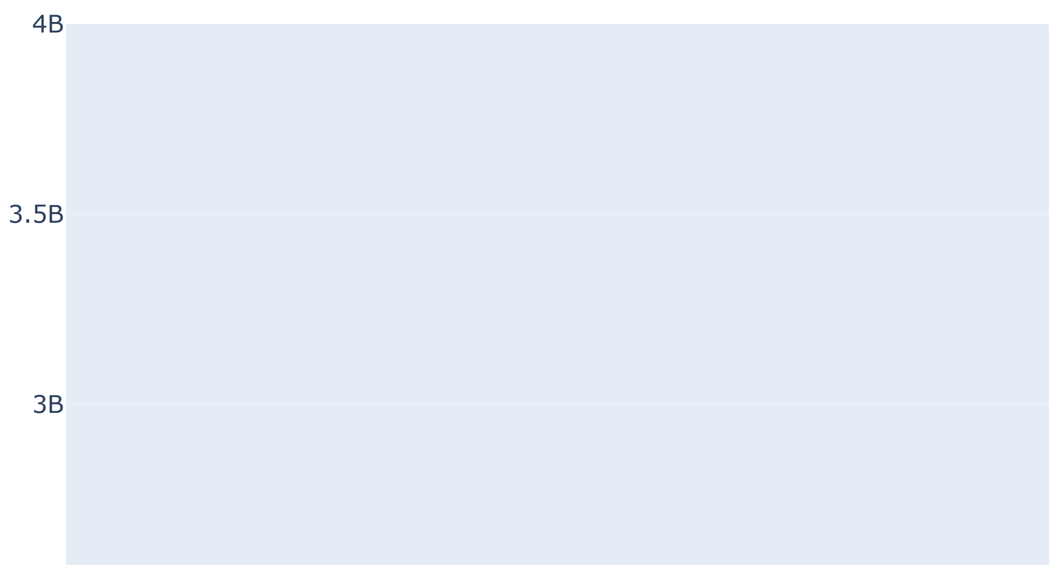


Question 6:

Clean up the country animation. Set the height size of the figure to 1000 to have a better view of the animation

In [8]:

```
'''Clean up the country animation. Set the height size of the figure to 1000 to have a bett
fig = px.bar(df, x="country", y="pop", color="continent", animation_frame="year", animation
fig.show()
```



Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

In [9]:

```
# YOUR CODE HERE
'''caculate the top 10 countries in terms of population in the year 2007'''
df_2007 = df[df['year'] == 2007]
df_2007 = df_2007.sort_values(by=['pop'], ascending=False)
df_2007 = df_2007.head(10)
df_2007

'''now create a animation that only the top 10 countries in terms of population in the year
df_2007=df_2007['country'].unique()

if 'country' in df:
    df = df[df['country'].isin(df_2007)]


fig = px.bar(df, x="country", y="pop", color="continent", animation_frame="year", animation
fig.show()
```

In [ ]: