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 [152]:

```
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 [153]:

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

Out[153]:

	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

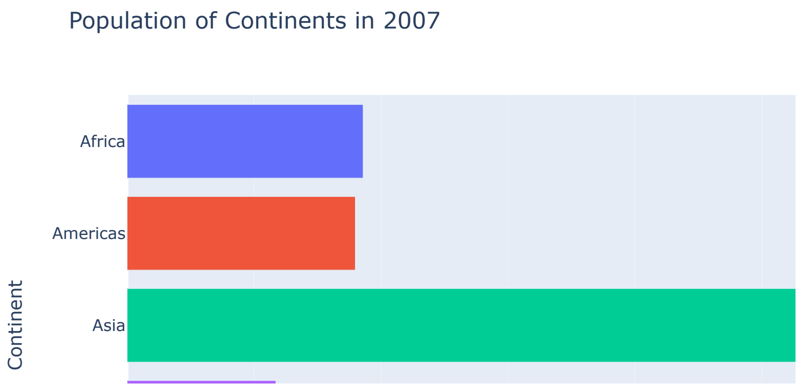
In [154]:

```

df = px.data.gapminder()

df_2007 = df[df['year'] == 2007]

population_by_continent = df_2007.groupby('continent')['pop'].sum().rese
fig = px.bar(
    population_by_continent,
    x='pop',
    y='continent',
    color='continent',
    labels={'pop': 'Population','continent': 'Continent'},
    title='Population of Continents in 2007'
)

fig.show()
```

Population of Continents in 2007
Africa

In [154]:

```
df = px.data.gapminder()

df_2007 = df[df['year'] == 2007]

population_by_continent = df_2007.groupby('continent')['pop'].sum().rese
fig = px.bar(
    population_by_continent,
    x='pop',
    y='continent',
    color='continent',
    labels={'pop': 'Population','continent': 'Continent'},
    title='Population of Continents in 2007'
)

fig.show()
```



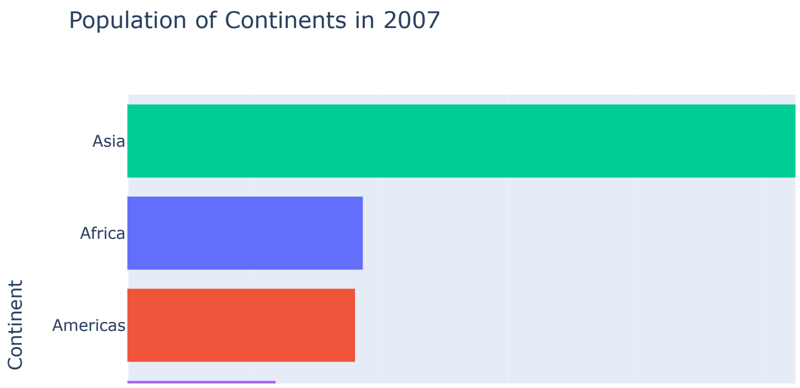
Question 2:

Sort the order of the continent for the visualisation

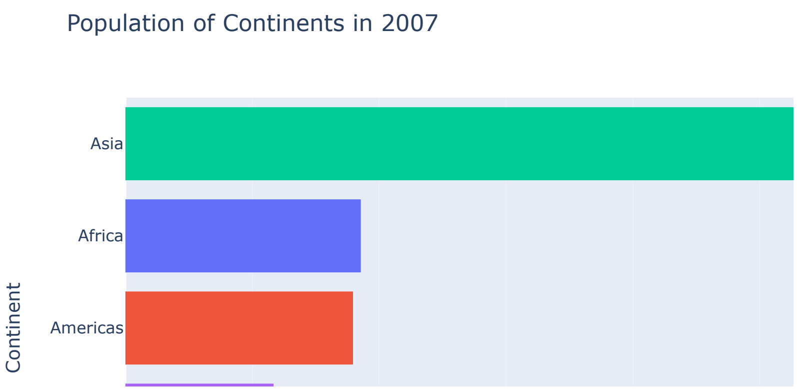
Hint: Use axis layout setting (https://plotly.com/python/reference/layout/xaxis/)

In [155]:

```
fig.update_yaxes(categoryorder='total ascending')

fig.show()
```



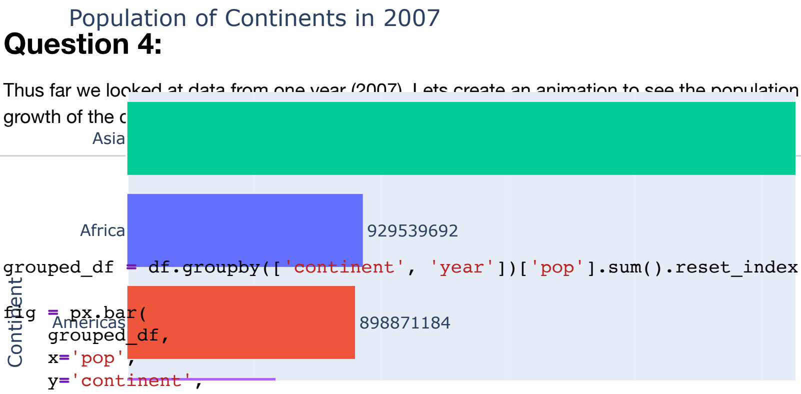
In [155]:
```
fig.update_yaxes(categoryorder='total ascending')

fig.show()
```



Question 3:

Add text to each bar that represents the population

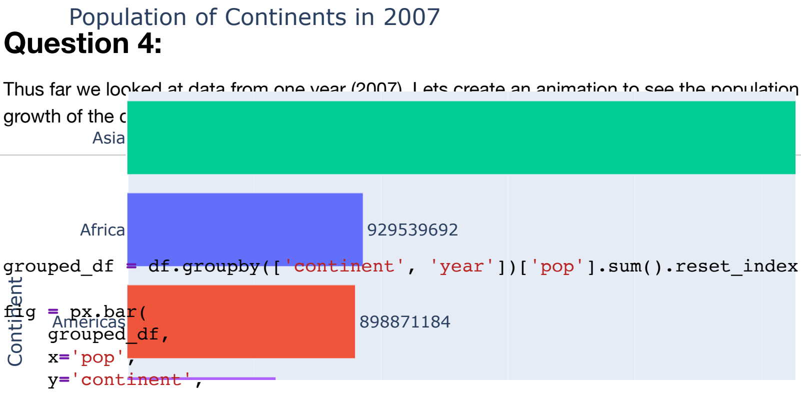
In [156]:
```
fig = px.bar(
    population_by_continent,
    x='pop',
    y='continent',
    color='continent',
    labels={'pop': 'Population','continent': 'Continent'},
    title='Population of Continents in 2007',
    text = 'pop'
)
fig.update_yaxes(categoryorder='total ascending')

fig.update_traces( textposition='outside')

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 c

In [157]:
```
grouped_df = df.groupby(['continent', 'year'])['pop'].sum().reset_index(

fig = px.bar(
    grouped_df,
    x='pop',
    y='continent',
    color='continent',
    labels={'pop': 'Population','continent': 'Continent'},
    animation_frame='year',
    title='Total Population of Continents Over the Years',
    text = 'pop'
)

fig.update_yaxes(categoryorder='total ascending')

fig.update_traces( textposition='outside')

fig.show()
```

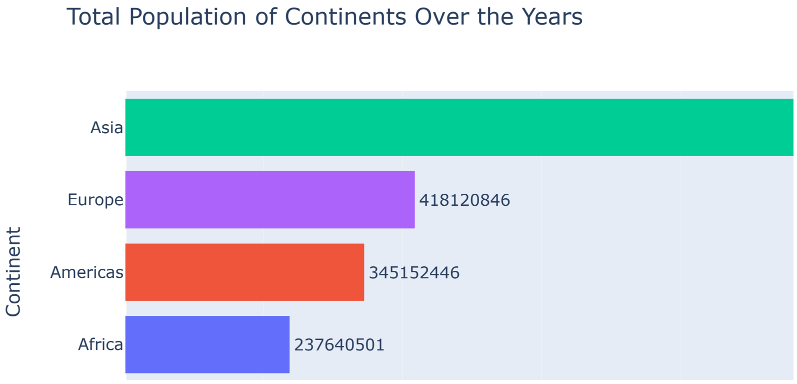


Question 4:

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

In [157]:

```
grouped_df = df.groupby(['continent', 'year'])['pop'].sum().reset_index(

fig = px.bar(
    grouped_df,
    x='pop',
    y='continent',
    color='continent',
    labels={'pop': 'Population','continent': 'Continent'},
    animation_frame='year',
    title='Total Population of Continents Over the Years',
    text = 'pop'
)

fig.update_yaxes(categoryorder='total ascending')

fig.update_traces( textposition='outside')

fig.show()
```

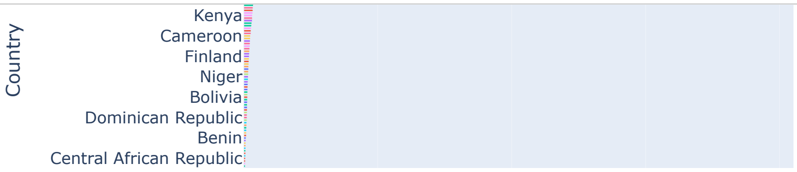


Question 5:

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

In [158]:

```
fig = px.bar(
    df,
    x='pop',
    y='country',
    color='country',
    labels={'pop': 'Population', 'country': 'Country'},    animation_fram
    title='Total Population of Countries Over the Years',
)

fig.update_yaxes(categoryorder='total ascending')


fig.show()
```

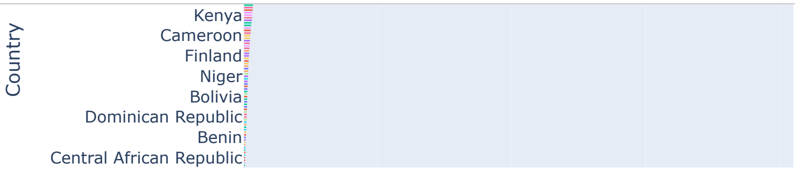


In [158]:

```
fig = px.bar(
    df,
    x='pop',
    y='country',
    color='country',
    labels={'pop': 'Population','country': 'Country'},     animation_fram
    title='Total Population of Countries Over the Years',
)

fig.update_yaxes(categoryorder='total ascending')


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

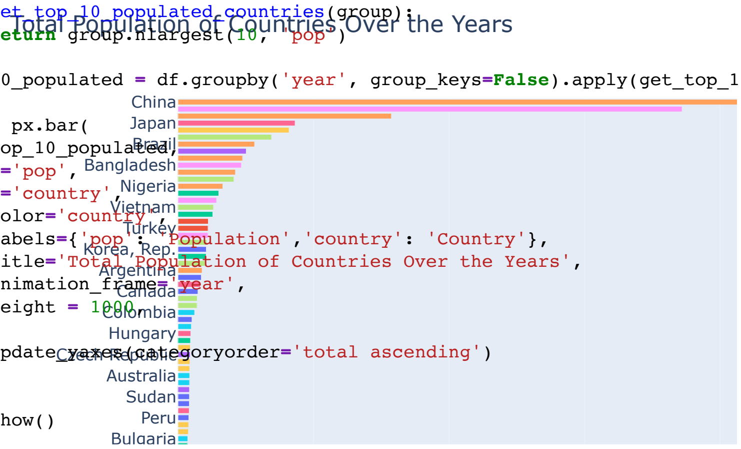


Question 6:

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

In [159]:

```
fig = px.bar(
    df,
    x='pop',
    y='country',
    color='country',
    labels={'pop': 'Population','country': 'Country'},
    title='Total Population of Countries Over the Years',
    animation_frame='year',
    height = 1000,
)

fig.update_yaxes(categoryorder='total ascending')


fig.show()
```



In [172]:

```
def get_top_10_populated_countries(group):
    return group.nlargest(10, 'pop')

top_10_populated = df.groupby('year', group_keys=False).apply(get_top_10_

fig = px.bar(
    top_10_populated,
    x='pop',
    y='country',
    color='country',
    labels={'pop': 'Population','country': 'Country'},
    title='Total Population of Countries Over the Years',
    animation_frame='year',
    height = 1000,
)
fig.update_yaxes(categoryorder='total ascending')


fig.show()
```

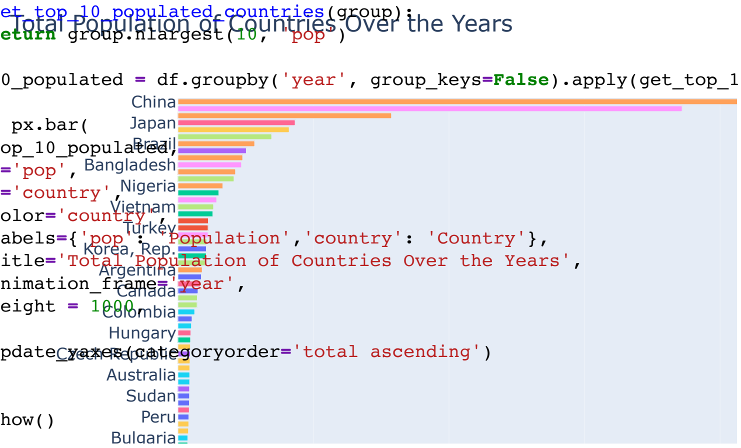
Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

In [172]:

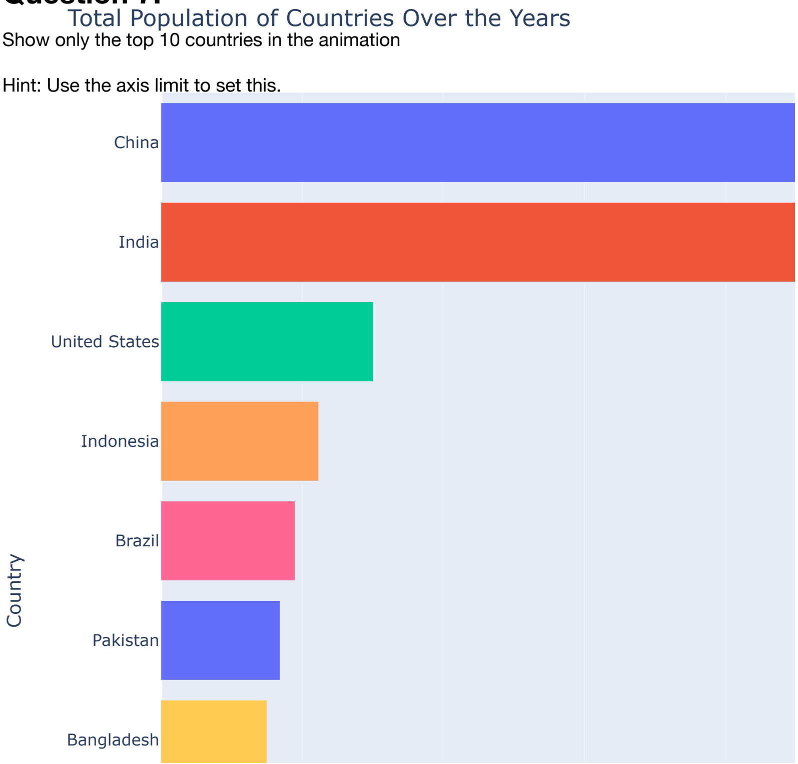
```
def get_top_10_populated_countries(group):
    return group.nlargest(10, 'pop')

top_10_populated = df.groupby('year', group_keys=False).apply(get_top_10

fig = px.bar(
    top_10_populated,
    x='pop',
    y='country',
    color='country',
    labels={'pop': 'Population','country': 'Country'},
    title='Total Population of Countries Over the Years',
    animation_frame='year',
    height = 1000
)
fig.update_yaxes(categoryorder='total ascending')

fig.show()
```



Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.



In [ ]: