

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 [ ]: 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 [ ]: #Load data
df = px.data.gapminder()
df.head()
```

```
Out[ ]:      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](#)
- 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

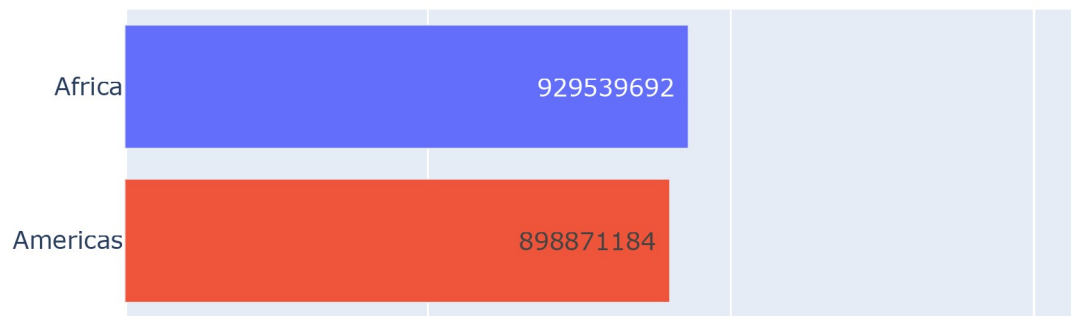
```
In [ ]: # extracting the data from 2007
df_2007 = df[
    (df['year'] == 2007)
]

# Labels of the continents
continents_2007 = df_2007.groupby('continent', as_index=False)

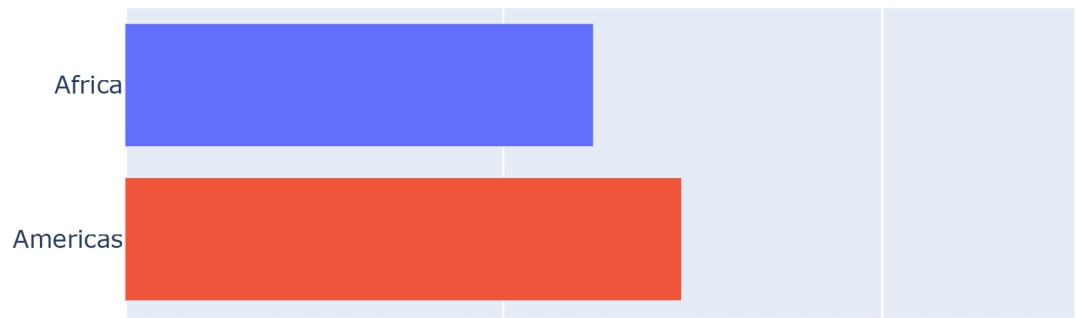
# Create a bar chart using Plotly Express
fig = px.bar(continents_2007.sum(), x="pop", y="continent", orientation='h', color=

fig.update_layout(showlegend=False)

# Using the layout of the example plot
fig.update_layout(yaxis = {'categoryorder': 'category descending'})
fig.update_layout(yaxis = dict(title = 'continent'))
fig.update_layout(xaxis = dict(title = 'pop'))
```



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



Question 2:

Sort the order of the continent for the visualisation

Hint: Use [axis layout setting](#)

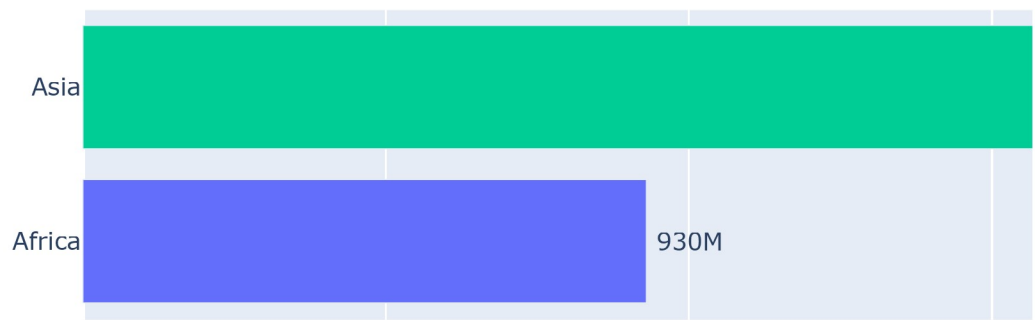
```
In [ ]: # YOUR CODE HERE
fig.update_layout(yaxis = {'categoryorder':'total ascending'})
```



Question 3:

Add text to each bar that represents the population

```
In [ ]: fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')
```



In []: *# YOUR CODE HERE*



Question 4:

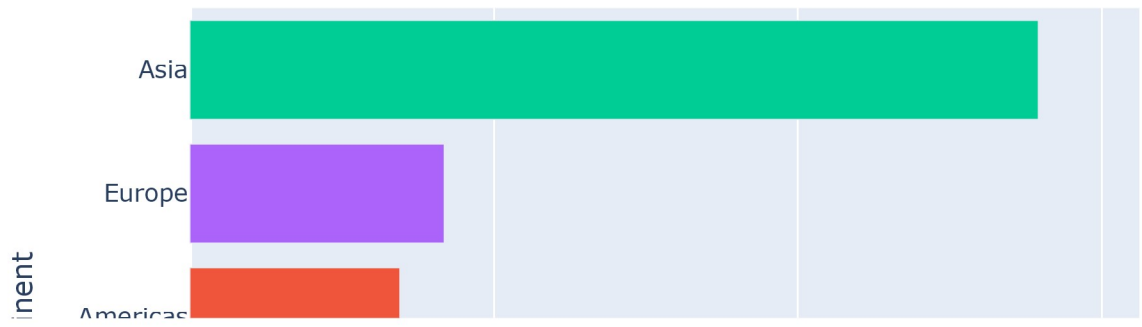
Thus far we looked at data from one year (2007). Lets create an animation to see the population growth of the continents through the years

```
In [ ]: # Filtering of the dataset
df_continents = df.groupby(['continent', 'year'], as_index=False)
df_continents.sum()

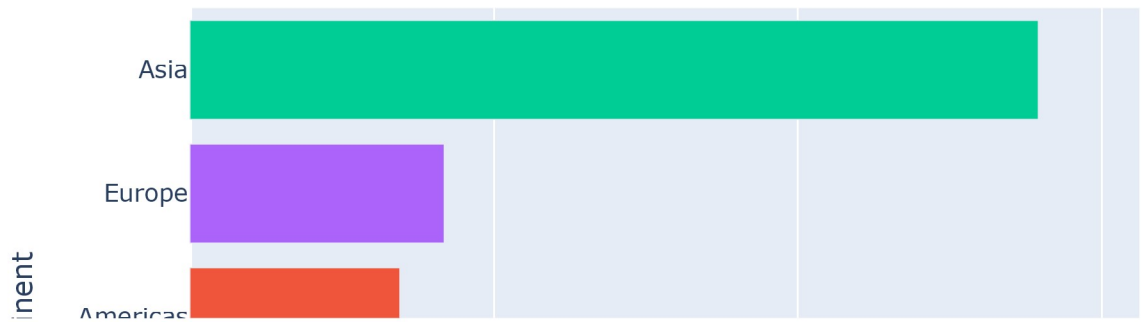
# Plot
fig = px.bar(df_continents.sum(), x="pop", y="continent", orientation='h',
             color="continent", animation_frame="year", animation_group="contine",
             range_x=[0, 4000000000])

fig.update_layout(showlegend = False)
fig.update_layout(yaxis = {'categoryorder': 'total ascending'})

fig.show()
```



In []: # YOUR CODE HERE

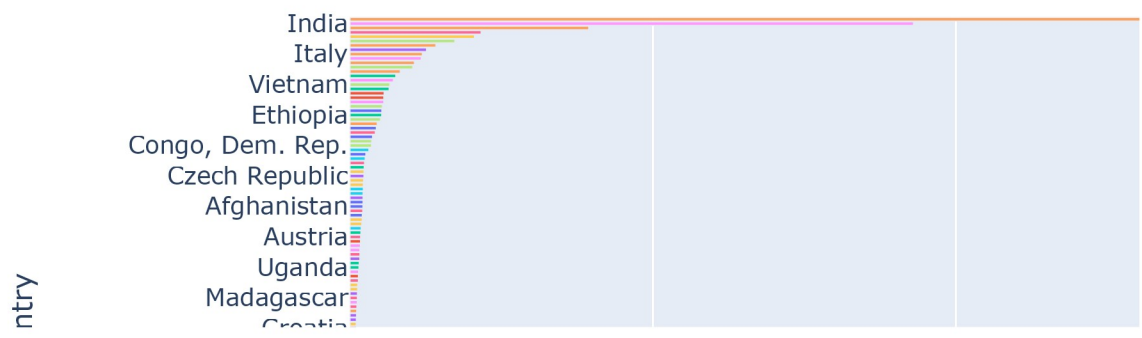


Question 5:

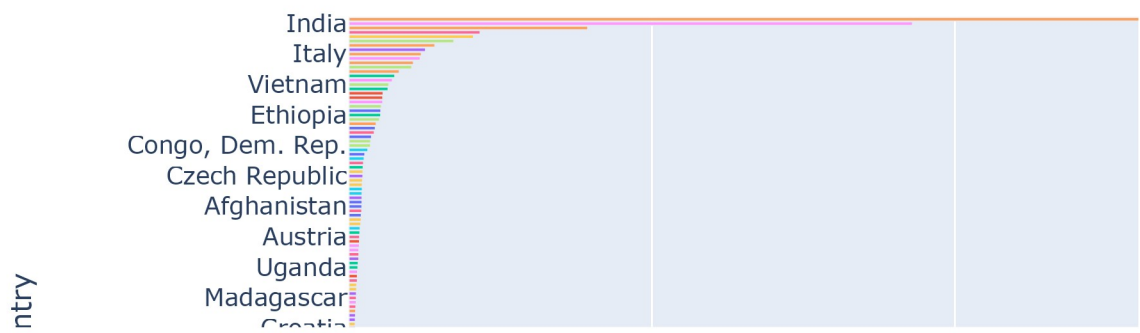
Instead of the continents, lets look at individual countries. Create an animation that shows the population growth of the countries through the years

```
In [ ]: fig = px.bar(df,x="pop", y="country", orientation='h',
                    color="country", animation_frame="year",range_x = [0, 1500000000])

fig.update_layout(showlegend = False)
fig.update_layout(yaxis = {'categoryorder':'total ascending'})
fig.show()
```

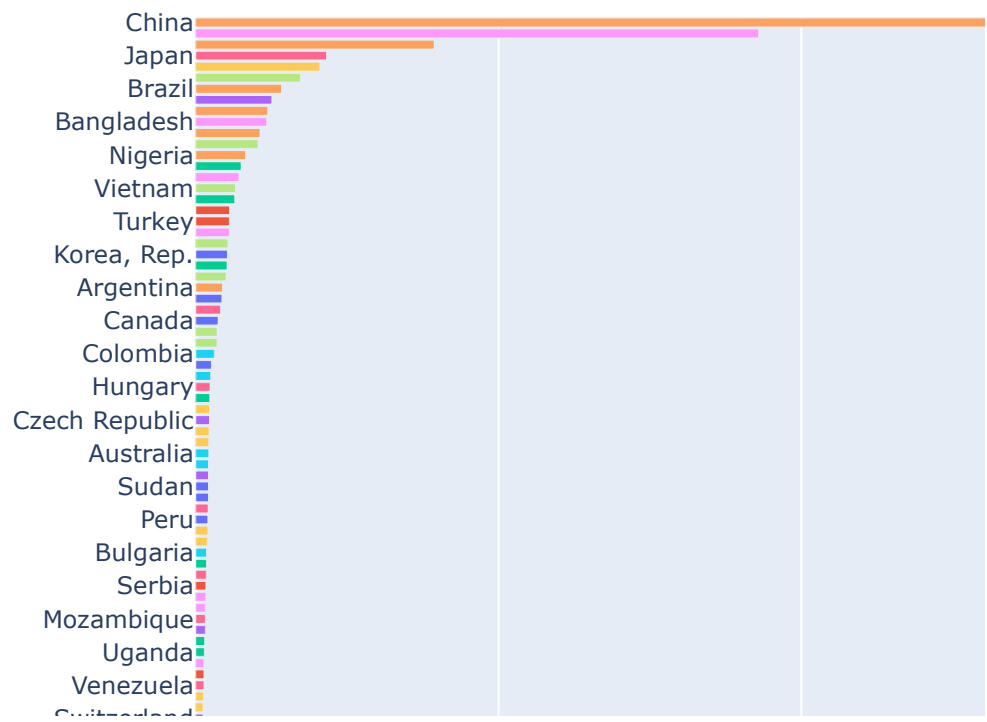
In []: # YOUR CODE HERE



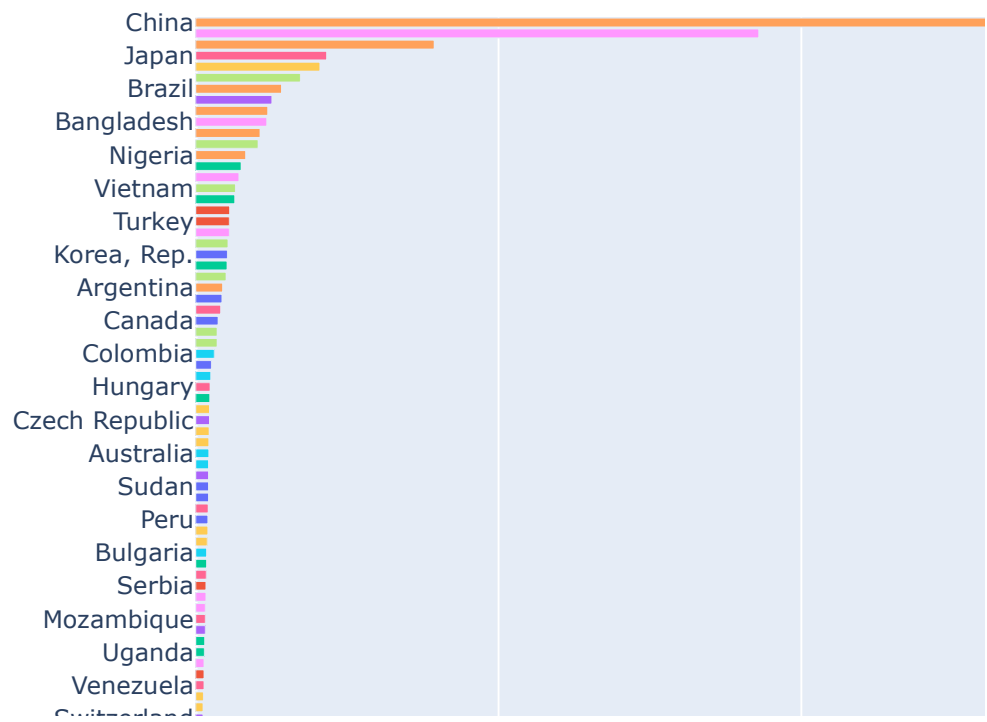
Question 6:

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

```
In [ ]: fig.update_layout(height = 1000)
```



In []: *# YOUR CODE HERE*

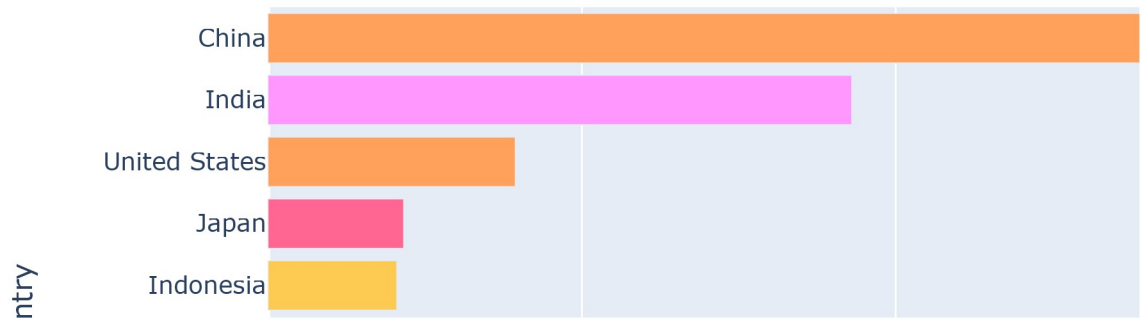


Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

```
In [ ]: range_countries = len(df['country'].unique())  
fig.update_yaxes(range=[range_countries-10.5, range_countries-0.5])  
fig.update_layout(height = None)
```



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

