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 [14]:
```
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 [100…
```
#load data
df = px.data.gapminder()
df.head()
```

Out[100]:

	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

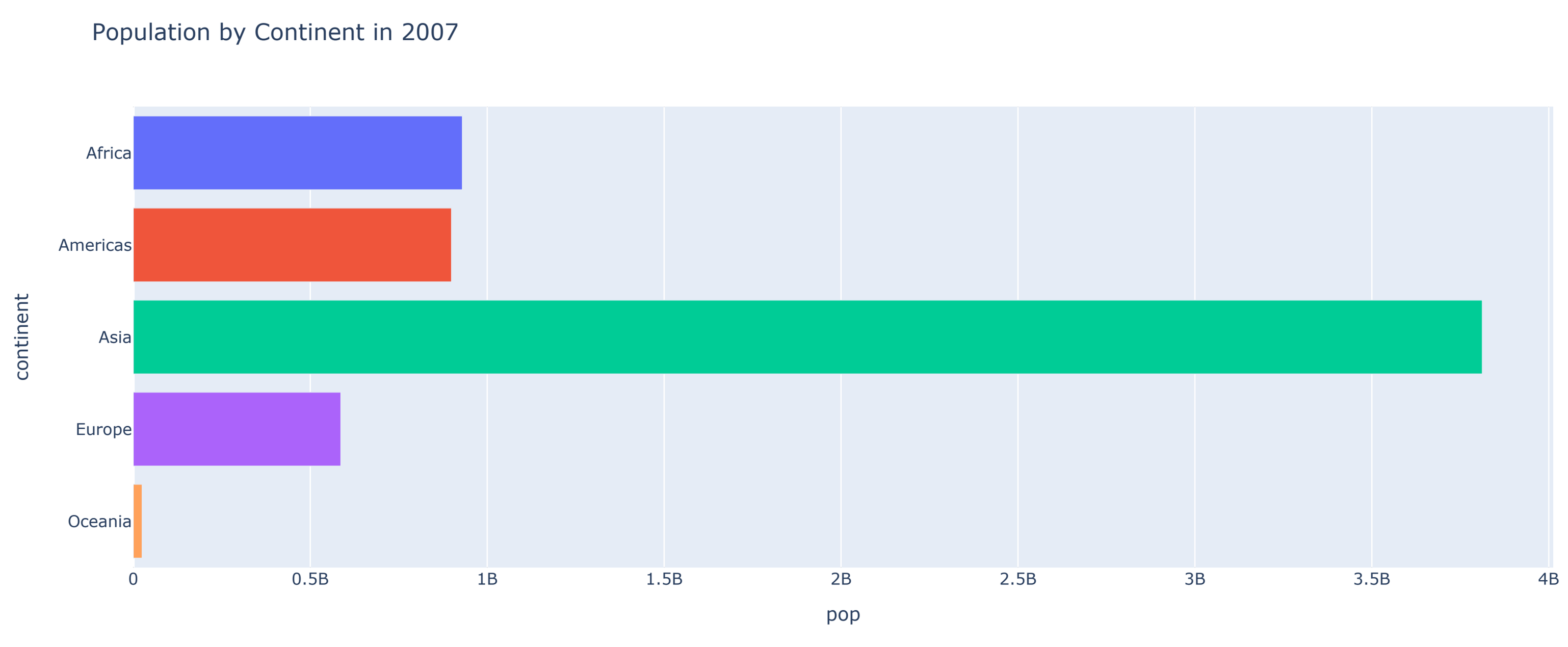
Question 1:

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 [101…
```
# YOUR CODE HERE

# Filter the DataFrame to select data for the year 2007

filtered_df = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns

grouped_df = filtered_df.groupby('continent')['pop'].sum().reset_index()

# Create a bar chart using Plotly Express

fig = px.bar(grouped_df, x='pop', y='continent', orientation='h',
    title='Population by Continent in 2007', color='continent'
)

# Customize the layout of the chart: hide the legend

fig.update_layout(showlegend=False)

# Display the resulting chart
fig.show()
```

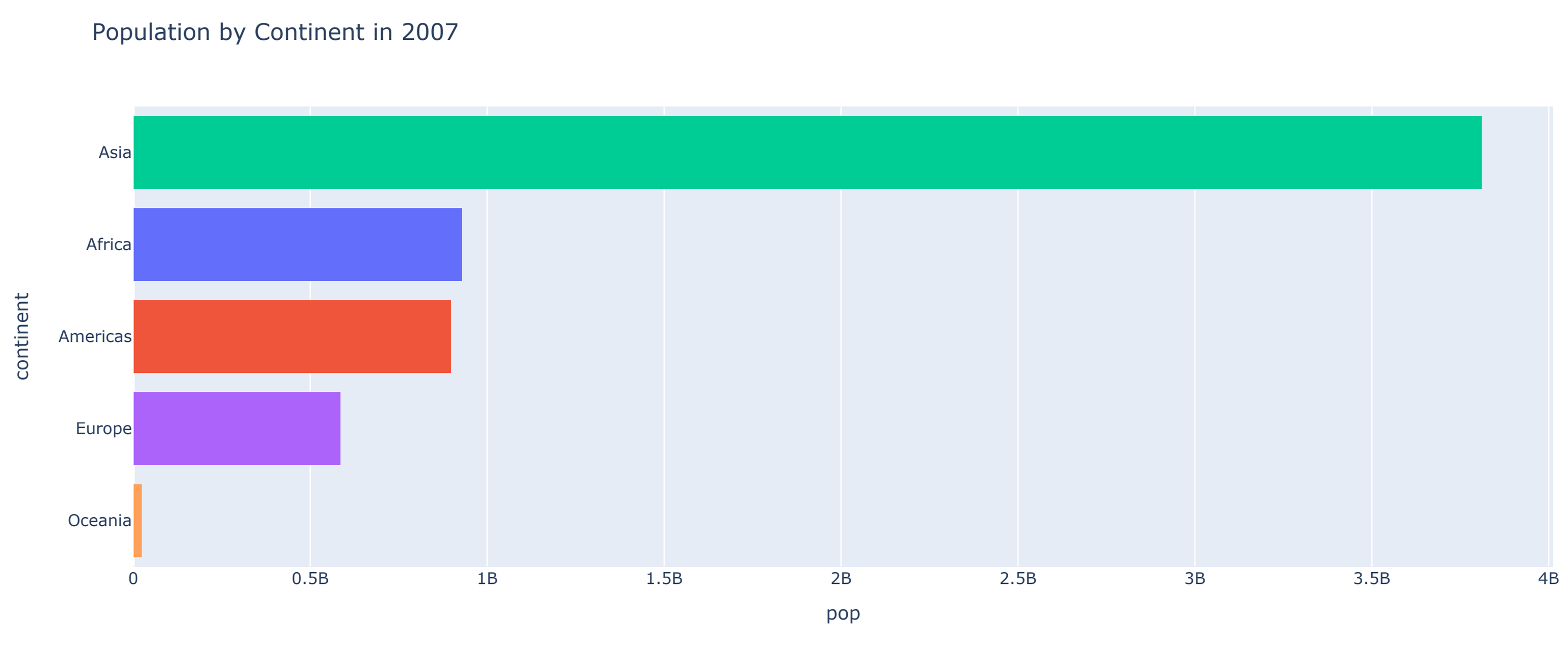


Question 2:

Sort the order of the continent for the visualisation

Hint: Use axis layout setting

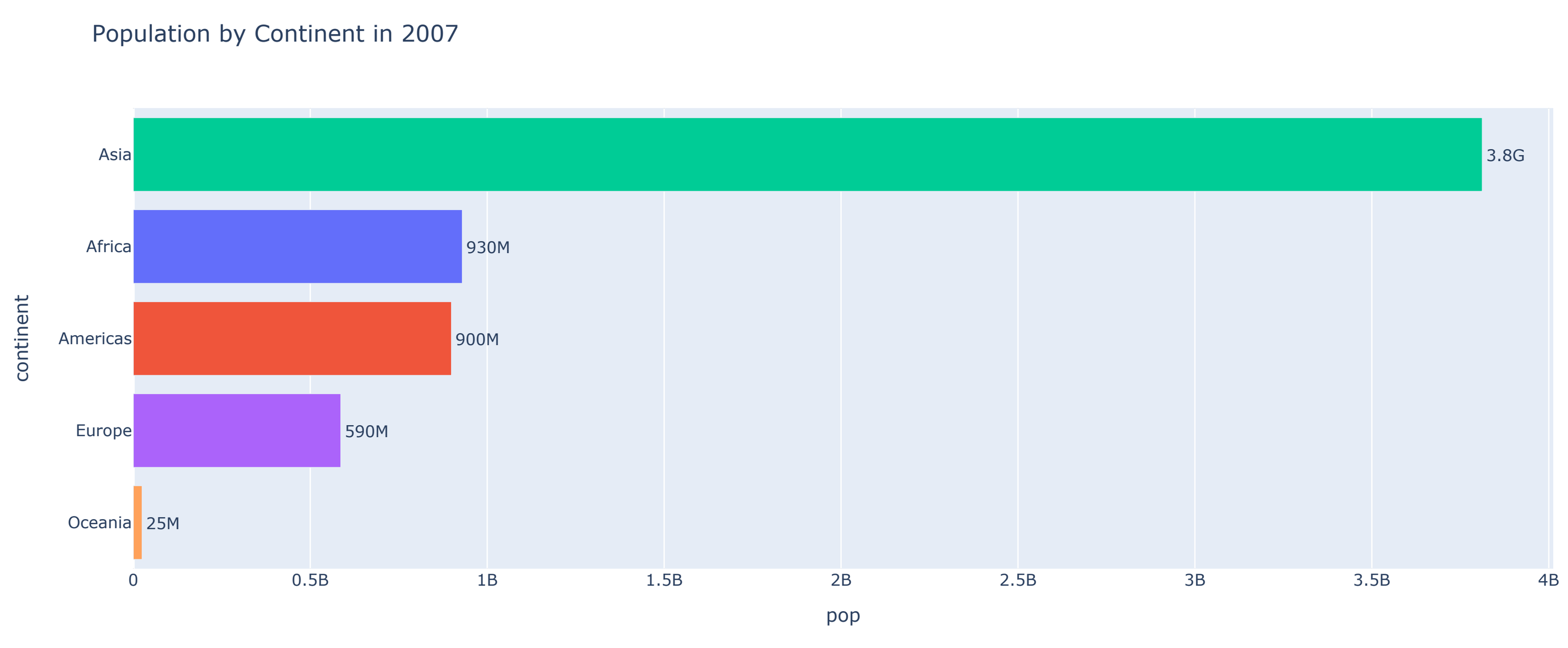
In [102…
```
# YOUR CODE HERE

# Filter the DataFrame to select data for the year 2007

filtered_df = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns

grouped_df = filtered_df.groupby('continent')['pop'].sum().reset_index()

# Create a bar chart using Plotly Express

fig = px.bar(grouped_df, x='pop', y='continent', orientation='h',
    title='Population by Continent in 2007', color='continent'
)

# Customize the layout of the chart: hide the legend

fig.update_layout(showlegend=False)

# Update the layout for the y-axis to order categories by total population in ascending order

fig.update_yaxes(categoryorder='total ascending')

# Display the resulting chart
fig.show()
```



Question 3:

Add text to each bar that represents the population

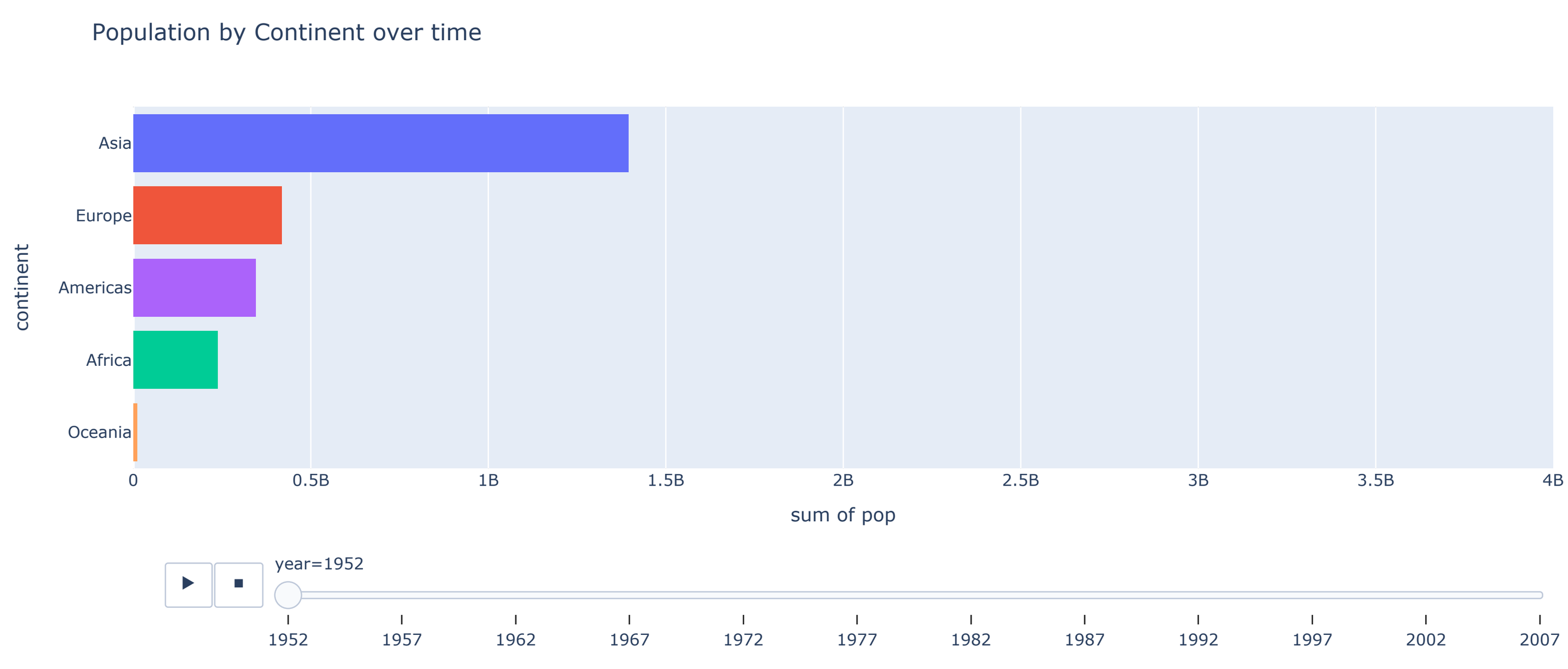
In [103…
```
# YOUR CODE HERE

# Filter the DataFrame to select data for the year 2007

filtered_df = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns

grouped_df = filtered_df.groupby('continent')['pop'].sum().reset_index()

# Create a bar chart using Plotly Express

fig = px.bar(grouped_df, x='pop', y='continent', orientation='h',
    title='Population by Continent in 2007', color='continent'
)

# Customize the layout of the chart: hide the legend

fig.update_layout(showlegend=False)

# Update the layout for the y-axis to order categories by total population in ascending order

fig.update_yaxes(categoryorder='total ascending')

# Customize the text labels on the bars: format with two decimal places and position them outside the bars
fig.update_traces(texttemplate='%{x:.2s}', textposition='outside')

# Display the resulting chart
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

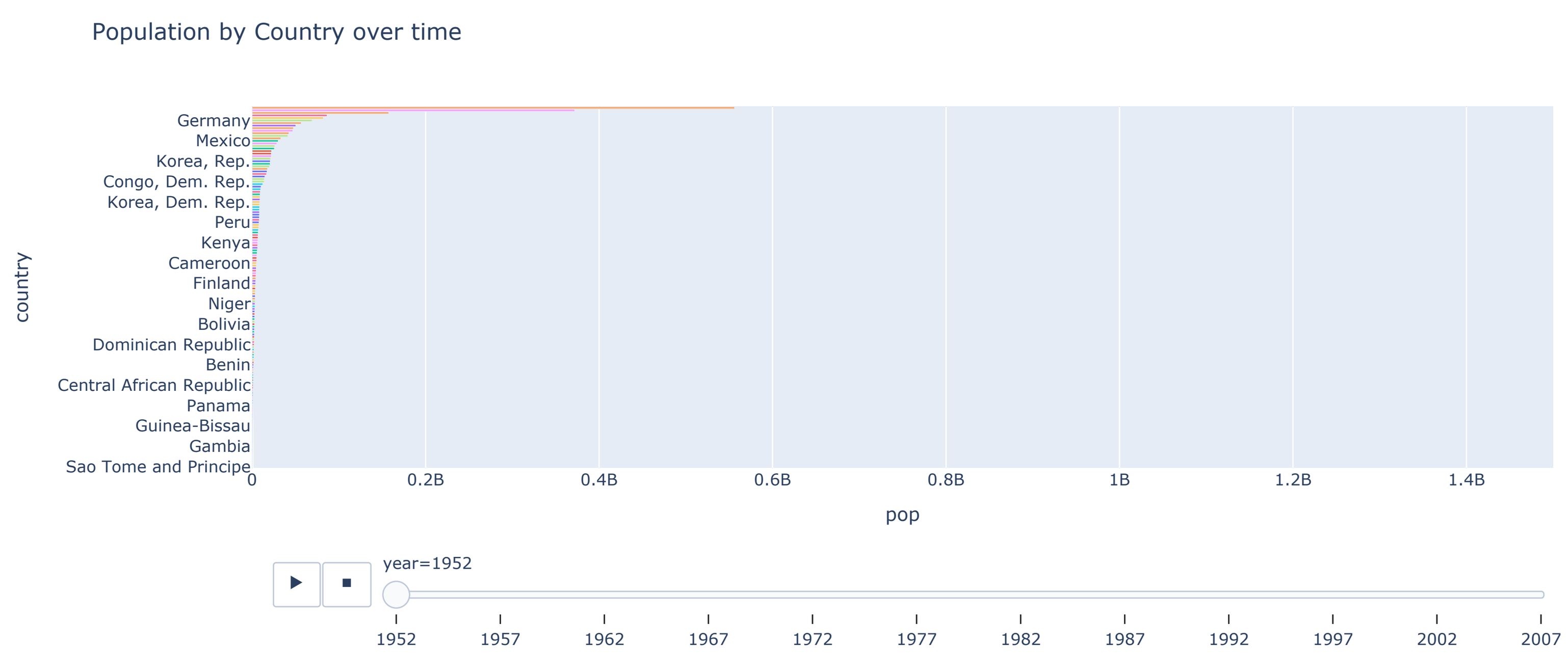
In [148…
```
# YOUR CODE HERE

# Create animation plot
animated_fig = px.histogram(df, x='pop', y='continent',
                     color='continent', animation_frame='year',
                     title='Population by Continent over time',
                     range_x=[0,4000000000] #set 4 billion on the x axis

)

# Customize the layout of the chart: hide the legend

animated_fig.update_layout(showlegend=False)

# Update the layout for the y-axis to order categories by total population in ascending order

animated_fig.update_yaxes(categoryorder='total ascending')

# Display the resulting chart
animated_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

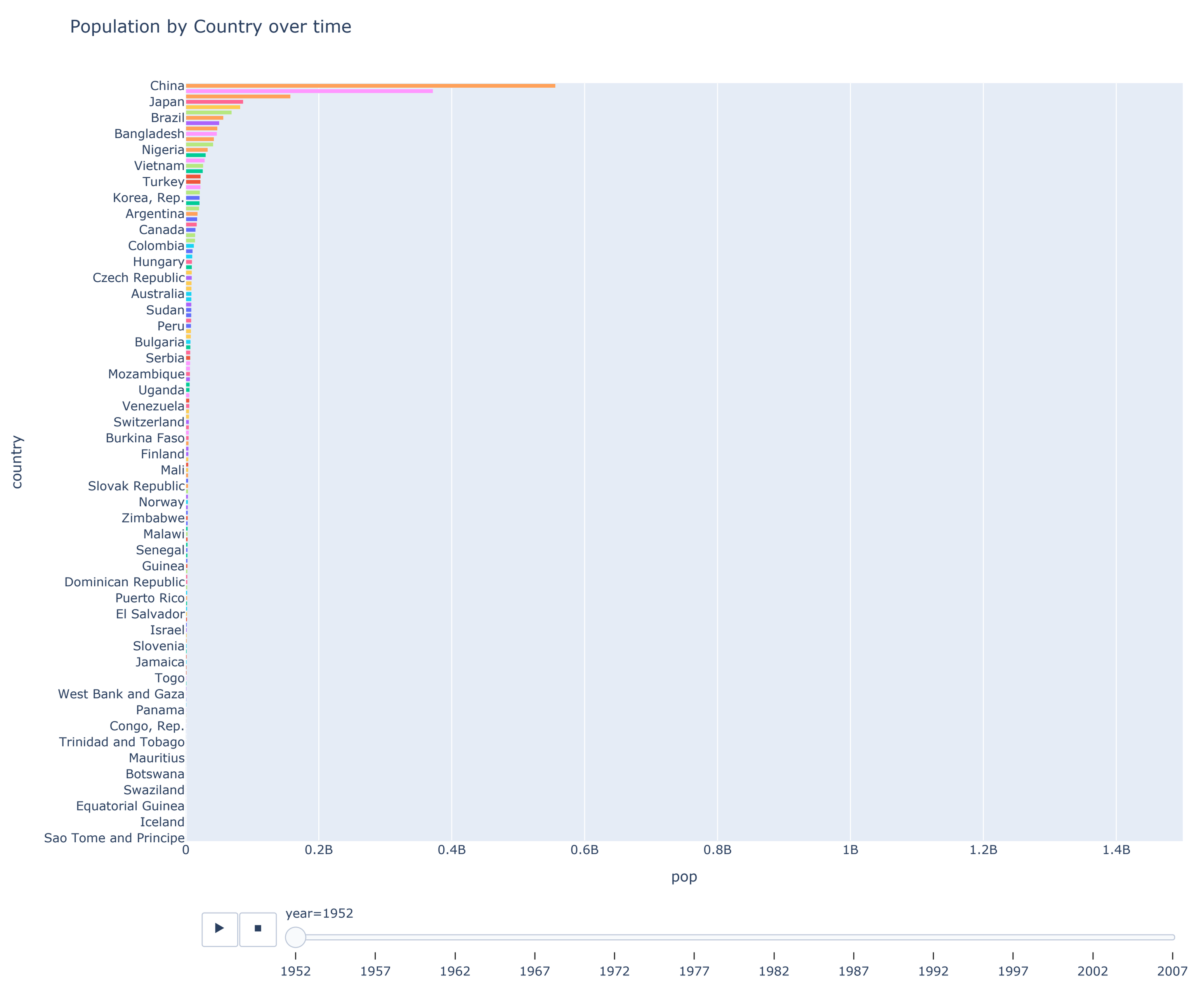
In [126…
```
# YOUR CODE HERE

# Create animation plot
animated_fig = px.bar(df, x='pop', y='country',
                     color='country', animation_frame='year',
                     title='Population by Country over time',
                     range_x=[0,1500000000] #set to 1.5 billion on x axis
)

# Customize the layout of the chart: hide the legend

animated_fig.update_layout(showlegend=False)

# Update the layout for the y-axis to order categories by total population in ascending order

animated_fig.update_yaxes(categoryorder='total ascending')

# Display the resulting chart
animated_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 [127…
```
# YOUR CODE HERE

# Create animation plot
animated_fig = px.bar(df, x='pop', y='country',
                     color='country', animation_frame='year',
                     title='Population by Country over time',
                     range_x=[0,1500000000]  #set to 1.5 billion on x axis
)

# Customize the layout of the chart: hide the legend

animated_fig.update_layout(showlegend=False)

#Update the height size of the figure to 1000

animated_fig.update_layout(height=1000)

# Update the layout for the y-axis to order categories by total population in ascending order

animated_fig.update_yaxes(categoryorder='total ascending')

# Display the resulting chart
animated_fig.show()
```



Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

In [141…
```
# YOUR CODE HERE

# Create animation plot
animated_fig = px.bar(df, x='pop', y='country',
                         color='country', animation_frame='year',
                         title='Population by Country over time',
                         range_x=[0, 1500000000],  #set to 1.5 billion on x axis
                         range_y=[131.5, 141.5] #get 10 countries on the y axis
)

# Customize the layout of the chart: hide the legend

animated_fig.update_layout(showlegend=False)

#Update the height size of the figure to 1000

animated_fig.update_layout(height=550)

# Update the layout for the y-axis to order categories by total population in ascending order

animated_fig.update_yaxes(categoryorder='total ascending')

# Display the resulting chart
animated_fig.show()
```