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:

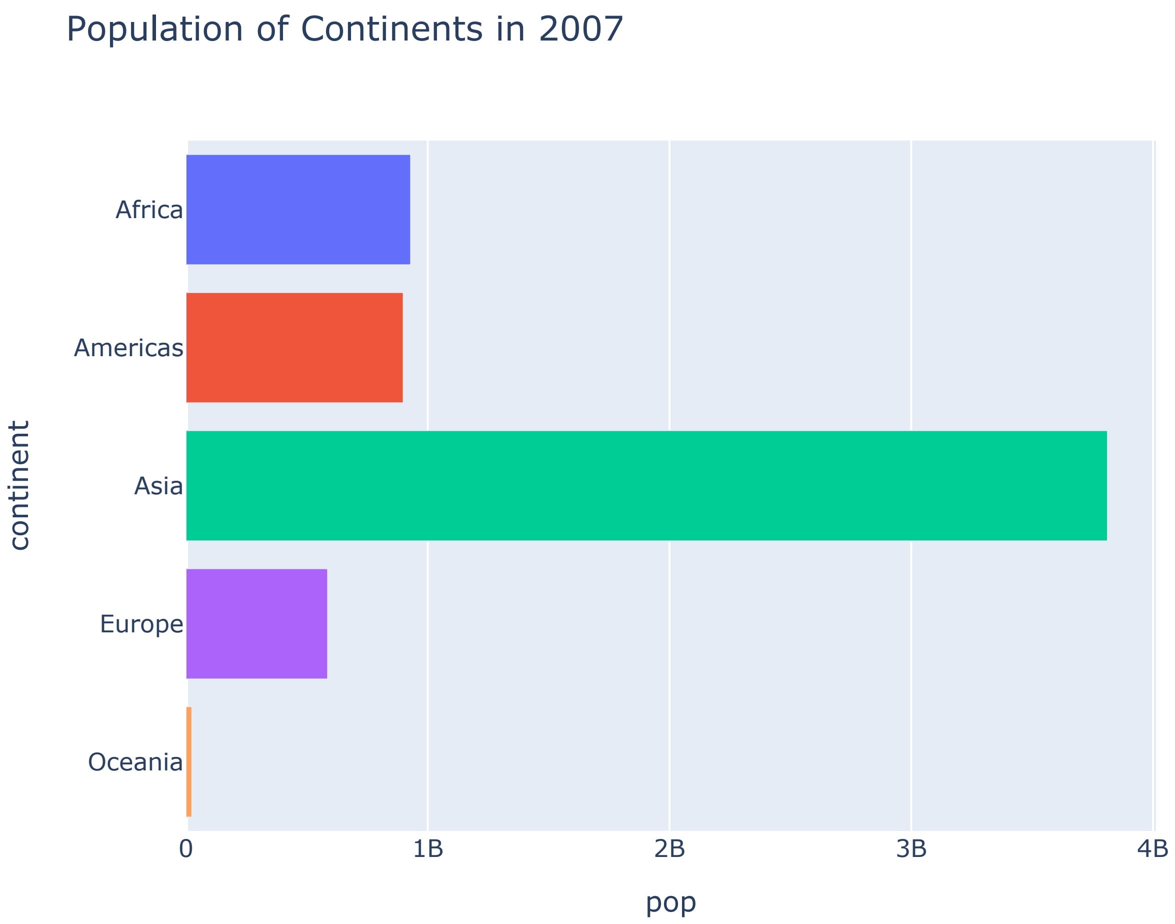
Recreate the barplot below that shows the population of different continents for the year 2007.

Hints:

- Extract the 2007 year data from the dataframe. You have to process the data accordingly
- use plotly bar
- 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 [3]:

```
# YOUR CODE HERE
# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]

# Sort the data by continent and population
df_2007_sorted = df_2007.sort_values(by=['continent', 'pop']).groupby('continent').sum()


# Create the barplot
fig = px.bar(
    df_2007_sorted,
    x = 'pop',
    y = 'continent',
    color= 'continent',  # Different colors for each continent
    title= 'Population of Continents in 2007'
)

# Customize layout of the figure
fig.update_layout(
    showlegend=False,   # Hide legend
)

# Show the plot
fig.show()
```



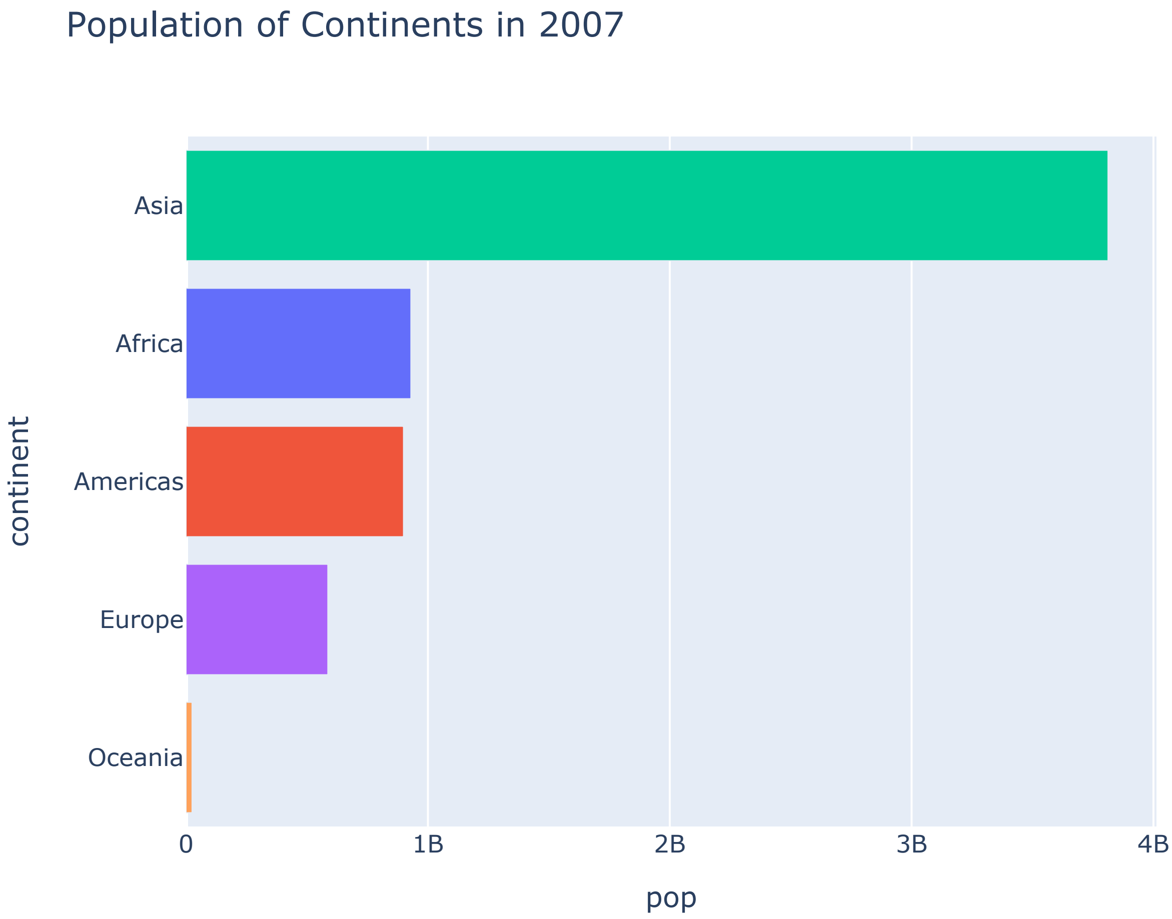
Question 2:

Sort the order of the continent for the visualisation

Hint: Use axis layout setting

In [4]:

```
# YOUR CODE HERE
# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]

# Sort the data by continent and population
df_2007_sorted = df_2007.sort_values(by=['continent', 'pop']).groupby('continent').sum()


# Create the barplot
fig = px.bar(
    df_2007_sorted,
    x = 'pop',
    y = 'continent',
    color= 'continent',  # Different colors for each continent
    title= 'Population of Continents in 2007'
)

# Customize layout of the figure
fig.update_layout(
    showlegend=False,  # Hide legend,
    yaxis={'categoryorder': 'total ascending'}   # Sort the order of the continent for t
)


# Show the plot
fig.show()
```



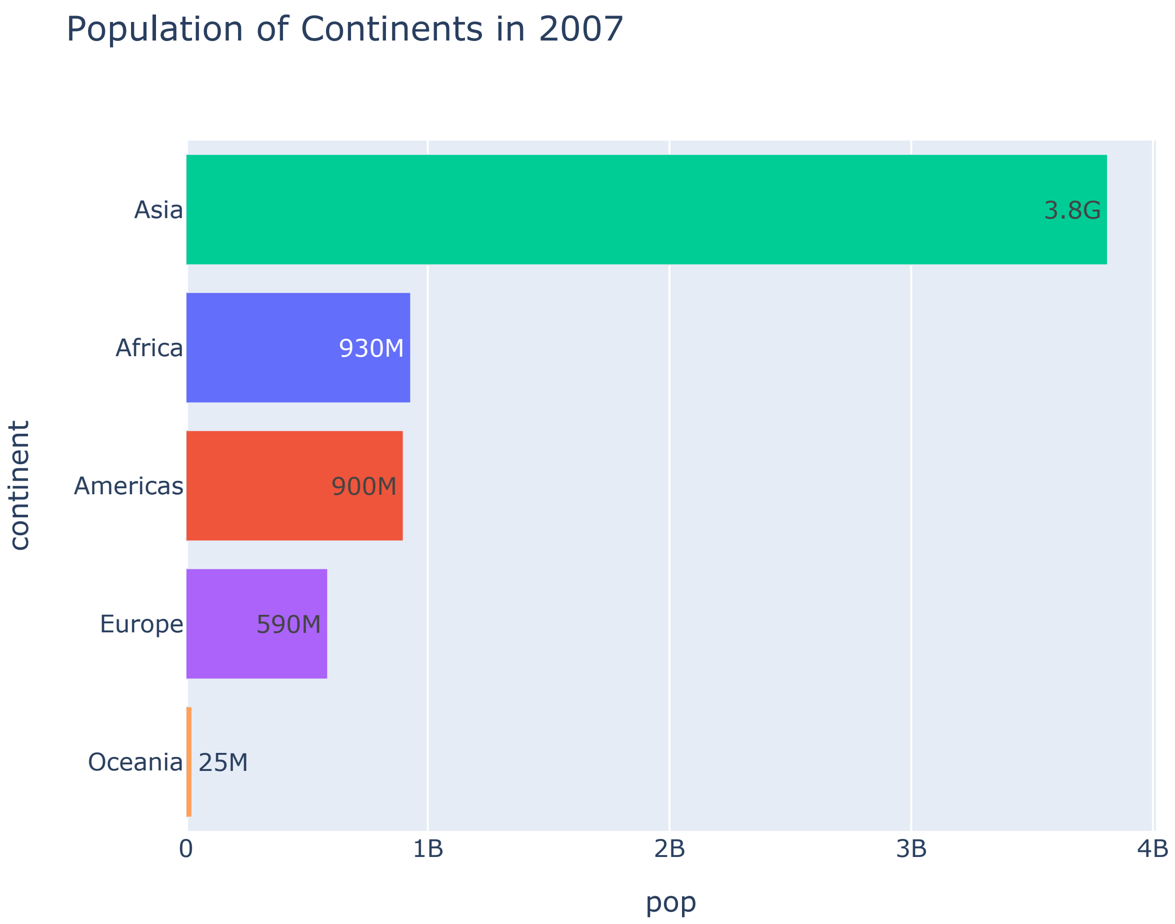
Question 3:

Add text to each bar that represents the population

In [26]:

```
# YOUR CODE HERE
# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]

# Sort the data by continent and population
df_2007_sorted = df_2007.sort_values(by=['continent', 'pop']).groupby('continent').sum()


# Create the barplot
```

```
fig = px.bar(
    df_2007_sorted,
    x = 'pop',
    y = 'continent',
    color= 'continent',  # Different colors for each continent
    text_auto='.2s',
    title= 'Population of Continents in 2007'
)

# Customize layout of the figure
fig.update_layout(
    showlegend=False,  # Hide legend,
    yaxis={'categoryorder': 'total ascending'},   # Sort the order of the continent for
)


# Show the plot
fig.show()
```

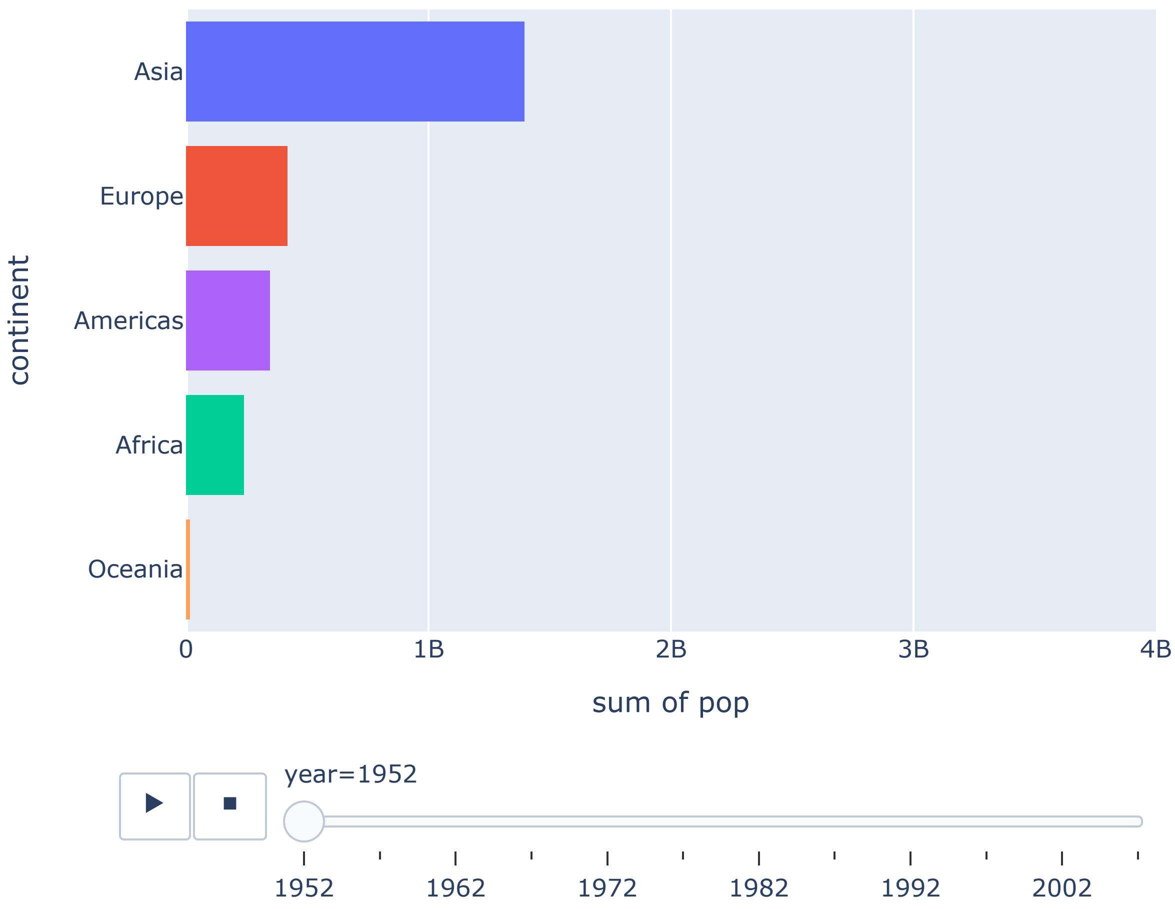


Question 4:

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

# Sort the data by continent and population
df_sorted = df.sort_values(by=['continent', 'pop'])
```

```python
# Create the barplot
animated_fig = px.histogram(
    df_sorted,
    x = 'pop',
    y = 'continent',
    color= 'continent',   # Different colors for each continent
    title= 'Population of Continents over time'
)

# Create animation
animated_fig = px.histogram(df, x="pop", y="continent", color="continent",
        animation_frame="year", range_x=[0,4000000000])

# Customize layout of the figure
animated_fig.update_layout(
    showlegend=False,   # Hide legend,
    yaxis={'categoryorder': 'total ascending'},    # Sort the order of the continent for
)

# Show figure
animated_fig.show()
```

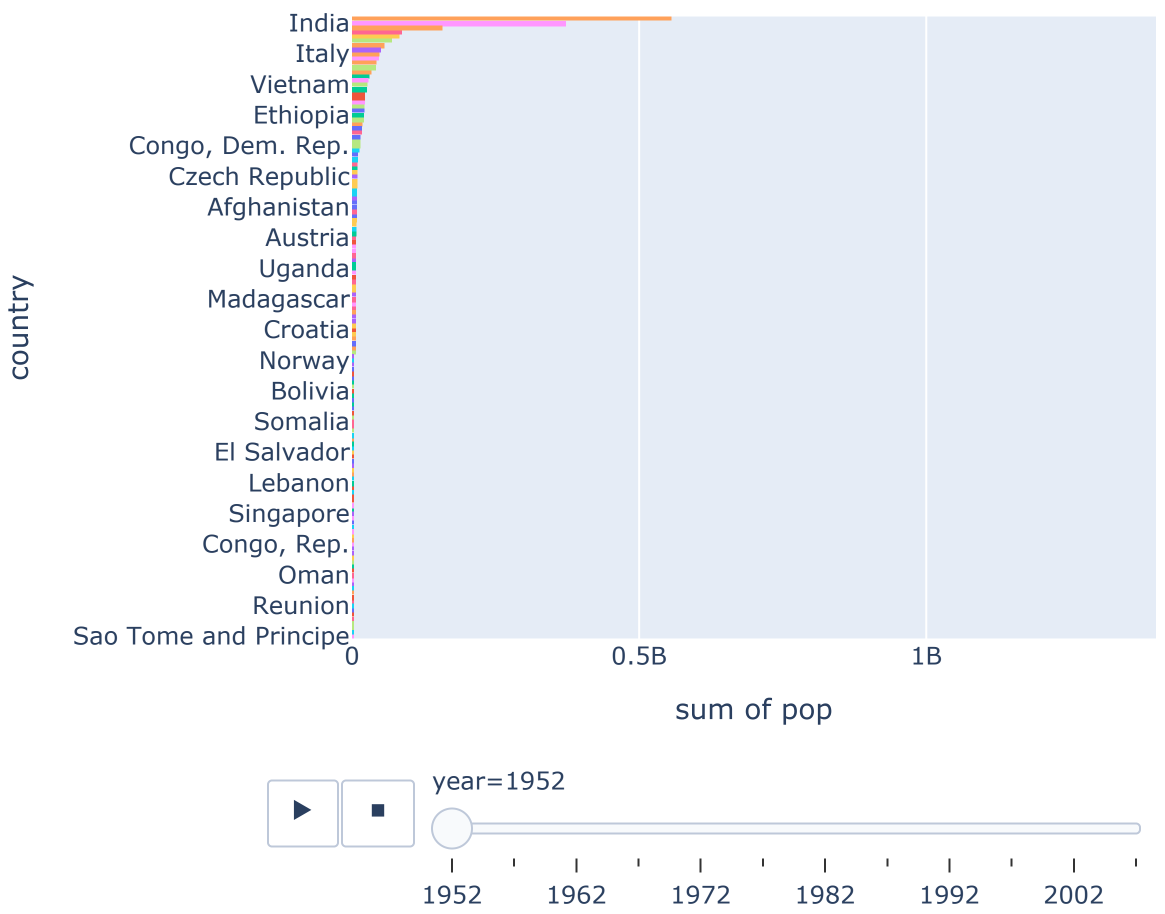


Question 5:

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

```
In [16]: # YOUR CODE HERE
```

```
# Create animation
animated_fig = px.histogram(df, x="pop", y="country", color="country",
        animation_frame="year", range_x=[0,1400000000])

# Customize layout of the figure
animated_fig.update_layout(
        showlegend=False,  # Hide legend,
        yaxis={'categoryorder': 'total ascending'},   # Sort the order of the continent
        title= 'Population of Countries over time'
)

# Show figure
animated_fig.show()
```

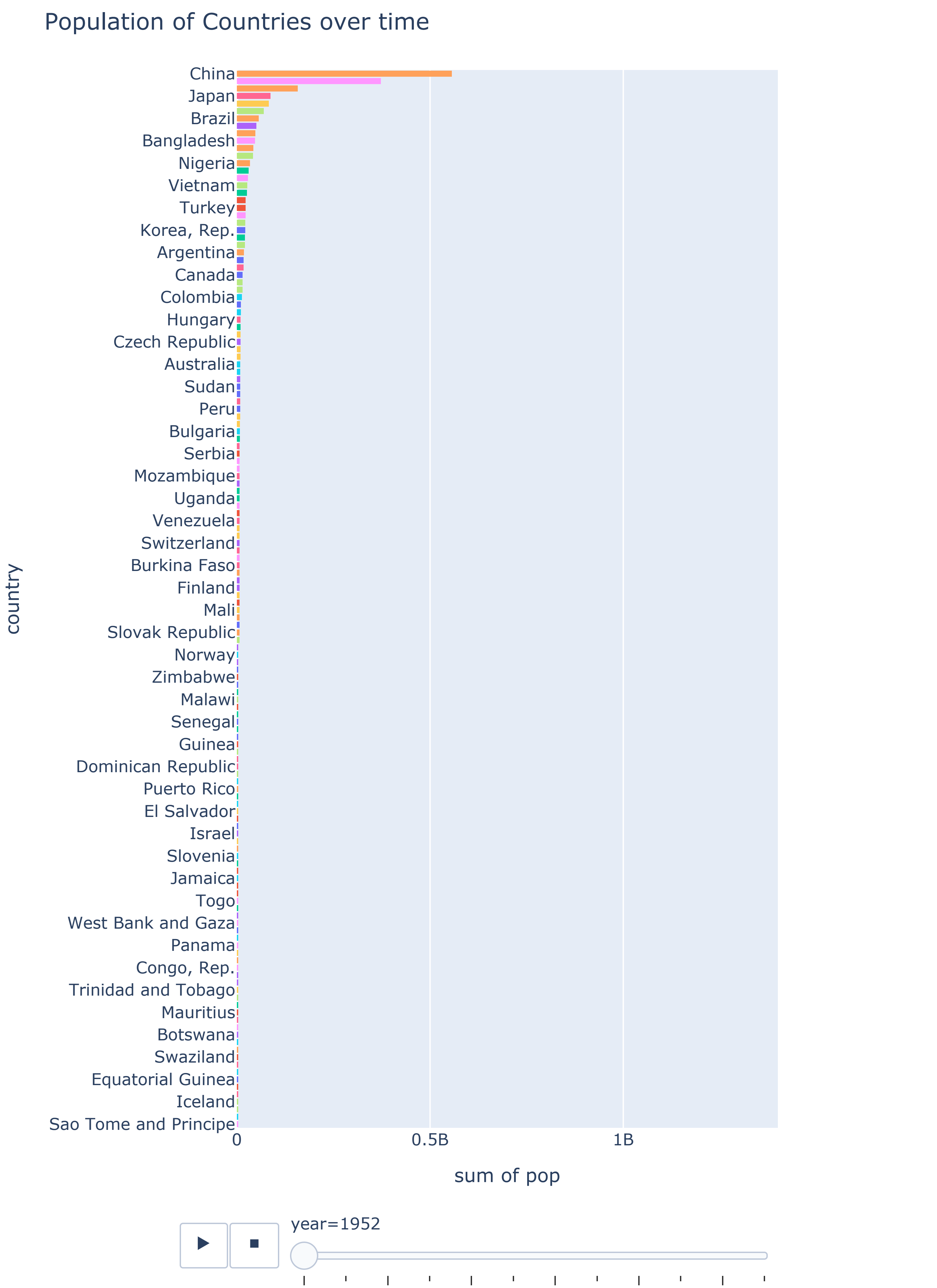


Question 6:

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

In [20]:

```
# YOUR CODE HERE
# Create animation
animated_fig = px.histogram(df, x="pop", y="country", color="country",
        animation_frame="year", range_x=[0,1400000000])

# Customize layout of the figure
animated_fig.update_layout(
    showlegend=False,  # Hide legend,
    yaxis={'categoryorder': 'total ascending'},   # Sort the order of the continent for
    height=1000,
    title= 'Population of Countries over time'
)

# Show figure
animated_fig.show()
```



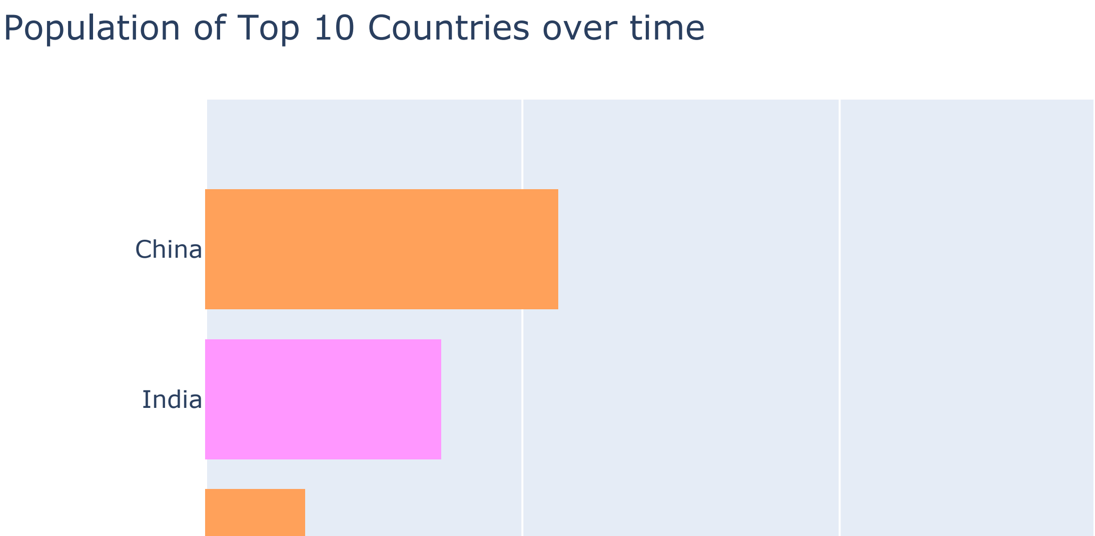
Question 7:

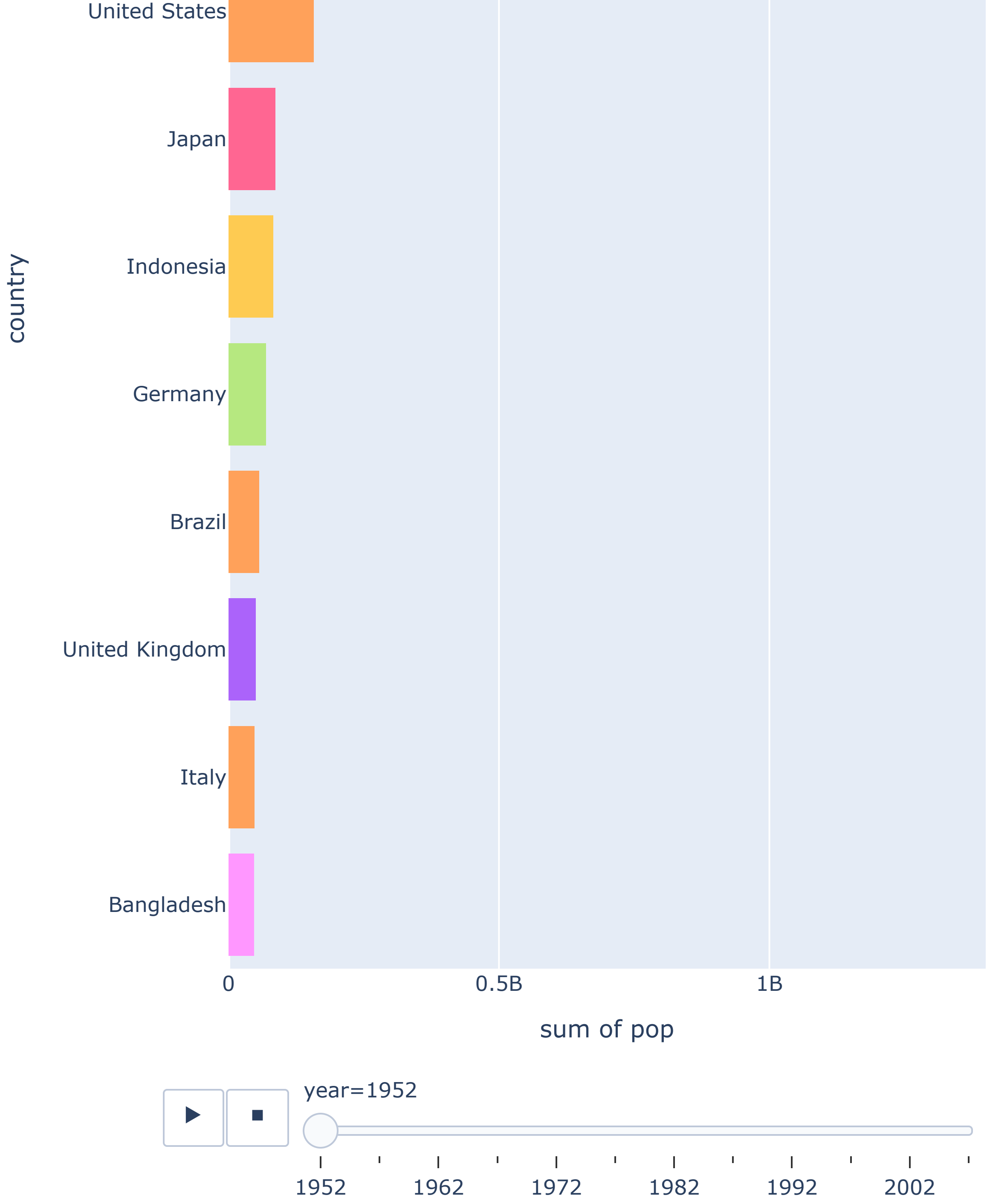
Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

In [25]:

```
# YOUR CODE HERE

# # Sort the data by continent and population
# df_sorted = df.sort_values(by=['country', 'pop'])


# # Create the barplot
# animated_fig = px.histogram(
#     df_sorted,
#     x = 'pop',
#     y = 'country',
#     color= 'country',  # Different colors for each continent
#     title= 'Population of Top 10 Countries over time'
# )

# Create animation
animated_fig = px.histogram(df,
                            x="pop",
                            y="country",
                            color="country",
                            animation_frame="year",
                            range_x=[0,1400000000],
                            range_y=[131.5,142]

)

# Customize layout of the figure
animated_fig.update_layout(
    showlegend=False,
    yaxis={'categoryorder': 'total ascending'},
    height=1000,
    title='Population of Top 10 Countries over time'
)

# Show figure
animated_fig.show()
```




United States
Japan
Indonesia
Germany
Brazil
United Kingdom
Italy
Bangladesh
country
0
0.5B
1B
sum of pop
▶
■
year=1952
1952
1962
1972
1982
1992
2002

In [ ]: