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 [2]:

```
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 [33]:

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

Out[33]:

	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:

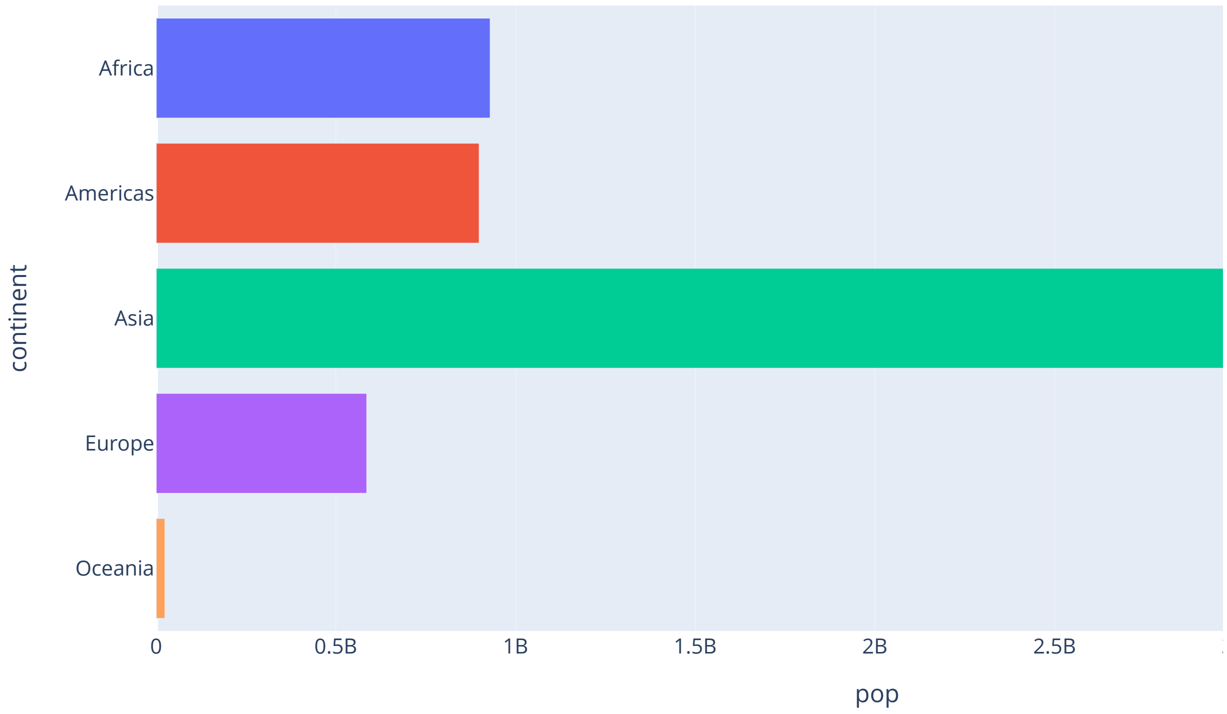
- Extract the 2007 year data from the dataframe. You have to process the data accordingly
- use plotly bar
- Add different colors for different continents
- Sort the order of the continent for the visualisation. Use axis layout setting
- Add text to each bar that represents the population

In [10]:

```
df_2007 = df[df['year'] == 2007]
df_2007_adapted = df_2007.groupby('continent').sum(numeric_only=True)
fig = px.bar(df_2007_adapted, x='pop', y = df_2007_adapted.index, color = df_2007_adapted.index)
fig.update_layout(showlegend=False)

fig.update_layout(autosize=False,
    width=1000,
    height=500,)

fig.show()
```



Question 2:

Sort the order of the continent for the visualisation

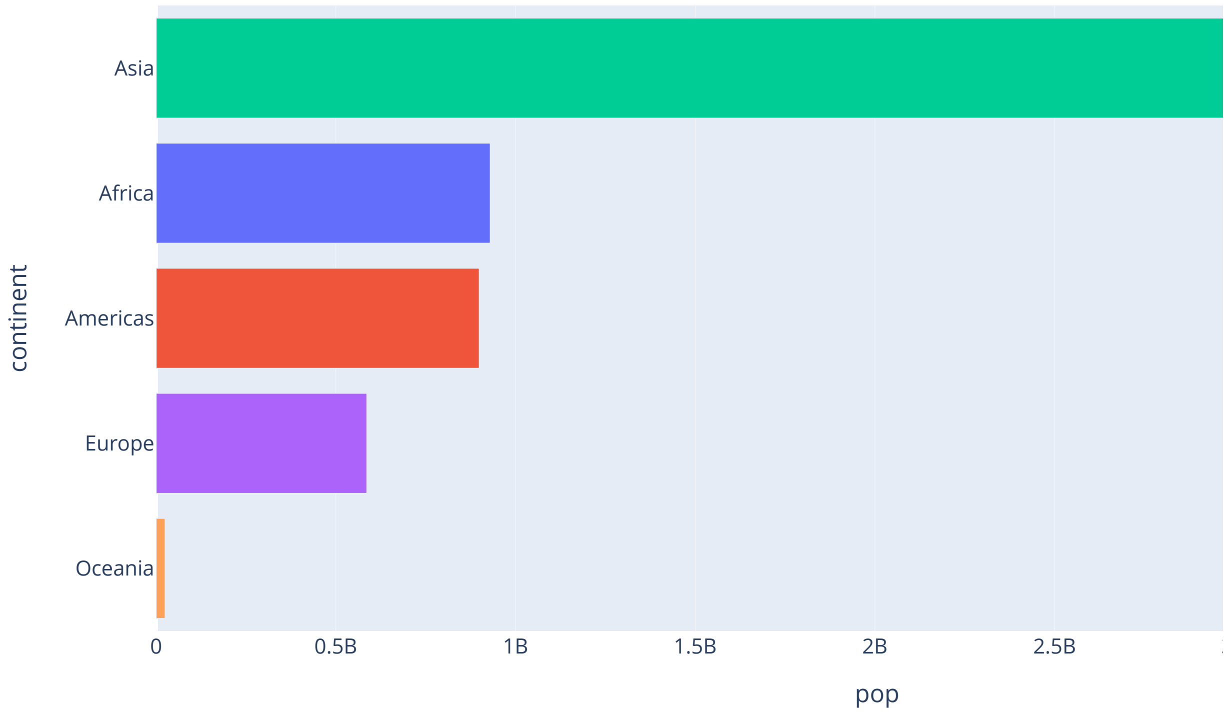
Hint: Use axis layout setting

In [9]:

```
df_2007 = df[df['year'] == 2007]
df_2007_adapted = df_2007.groupby('continent').sum(numeric_only=True)
fig = px.bar(df_2007_adapted, x='pop', y = df_2007_adapted.index, color = df_2007_adapted.index)
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder = "total ascending")

fig.update_layout(autosize=False,
    width=1000,
    height=500,)

fig.show()
```



Question 3:

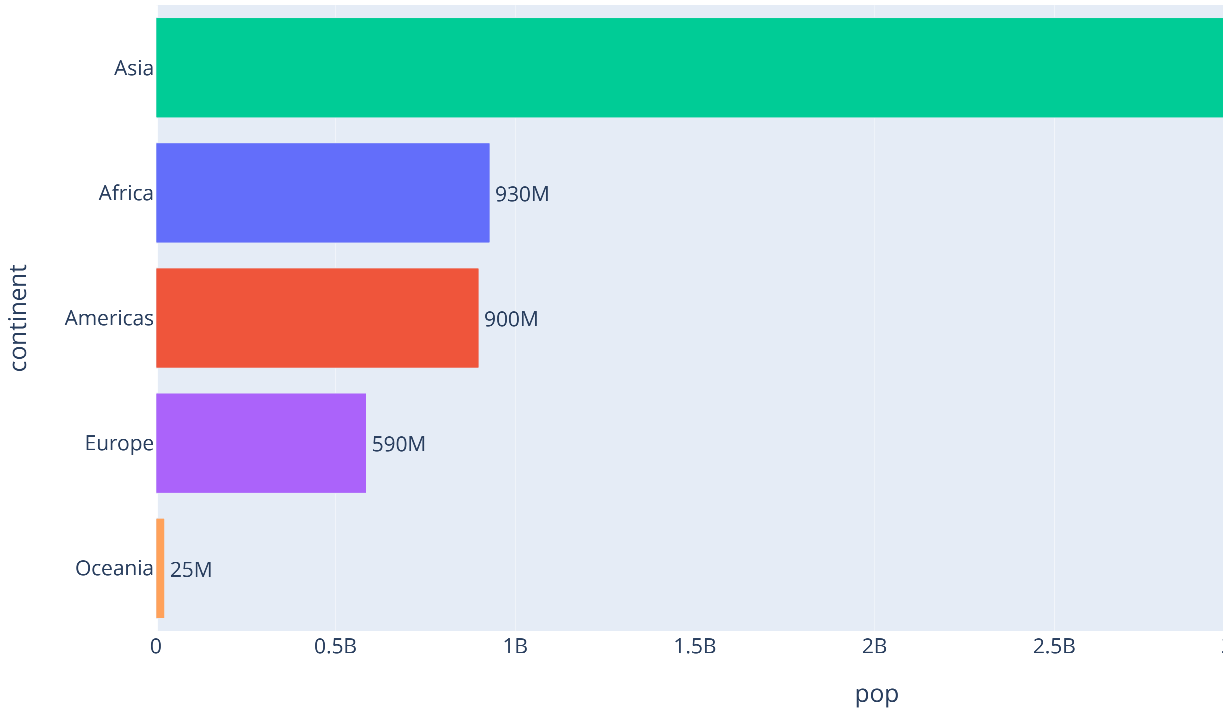
Add text to each bar that represents the population

In [15]:

```
df_2007 = df[df['year'] == 2007]
df_2007_adapted = df_2007.groupby('continent').sum(numeric_only=True)
fig = px.bar(df_2007_adapted, x='pop', y = df_2007_adapted.index, text = 'pop', color = df_2007_ad
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder = "total ascending")
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')

fig.update_layout(autosize=False,
    width=1000,
    height=500,)

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 [46]:

```python
df_pop_in_year = df.groupby(['year', 'continent']).sum(numeric_only=True).reset_index()
df_pop_in_year.head()
```

Out[46]:

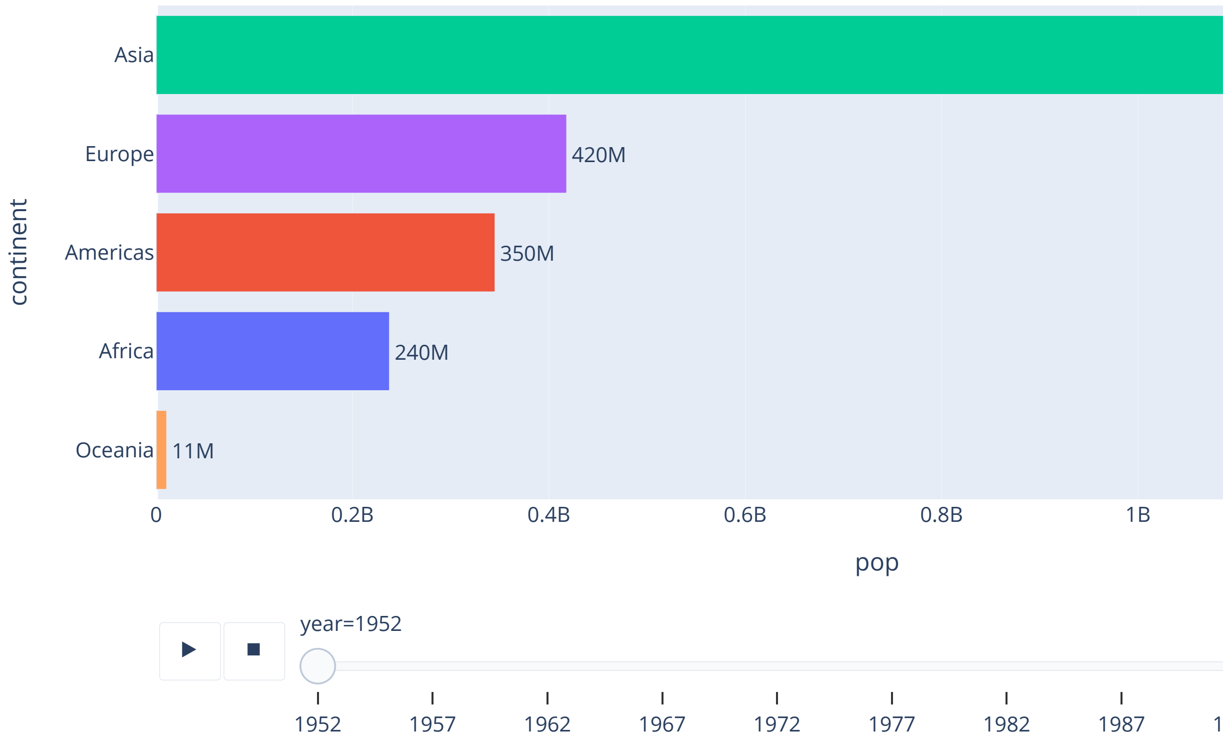
	year	continent	lifeExp	pop	gdpPercap	iso_num
0	1952	Africa	2035.046	237640501	65133.768223	23859
1	1952	Americas	1331.996	345152446	101976.563805	9843
2	1952	Asia	1528.375	1395357351	171450.972133	13354
3	1952	Europe	1932.255	418120846	169831.723043	12829
4	1952	Oceania	138.510	10686006	20596.171300	590

In [49]:

```python
fig = px.bar(df_pop_in_year, x='pop', y='continent', orientation='h', text = 'pop',
             color='continent', animation_frame='year', animation_group='continent')
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder = "total ascending")
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')

fig.update_layout(autosize=False,
    width=1000,
    height=500,)

fig.show()
```



Question 5:

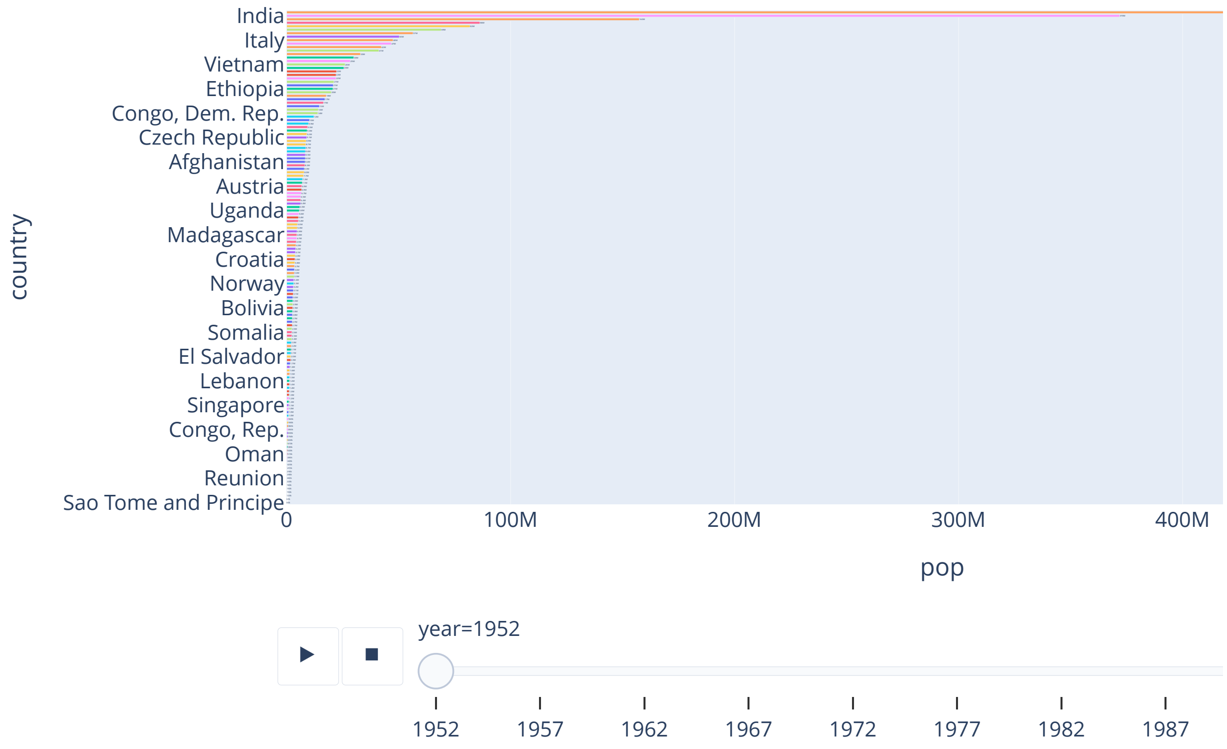
Instead of the continents, lets look at individual countries. Create an animation that shows the population growth of the countries through the years

In [52]:

```
df_pop_in_year = df.groupby(['year', 'country']).sum(numeric_only=True).reset_index()
fig = px.bar(df_pop_in_year, x='pop', y='country', orientation='h', text = 'pop',
             color='country', animation_frame='year', animation_group='country')
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder = "total ascending")
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')

fig.update_layout(autosize=False,
    width=1000,
    height=500,)

fig.show()
```



Question 6:

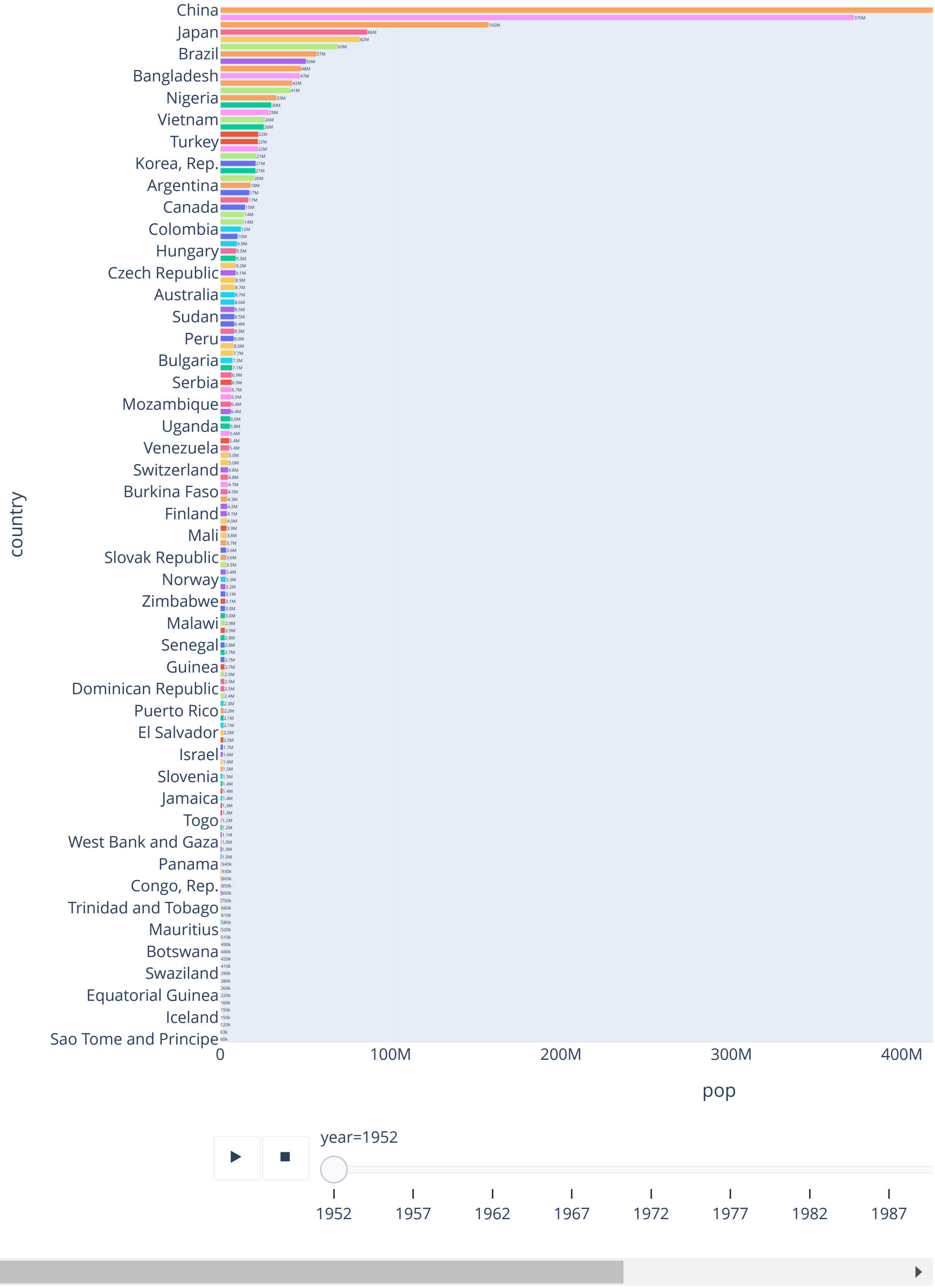
Clean up the country animation. Set the height size of the figure to 1000 to have a better view of the animation

In [53]:

```
df_pop_in_year = df.groupby(['year', 'country']).sum(numeric_only=True).reset_index()
fig = px.bar(df_pop_in_year, x='pop', y='country', orientation='h', text = 'pop',
             color='country', animation_frame='year', animation_group='country')
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder = "total ascending")
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')

fig.update_layout(autosize=False,
    width=1000,
    height=1000,)

fig.show()
```



Question 7:

Show only the top 10 countries in the animation

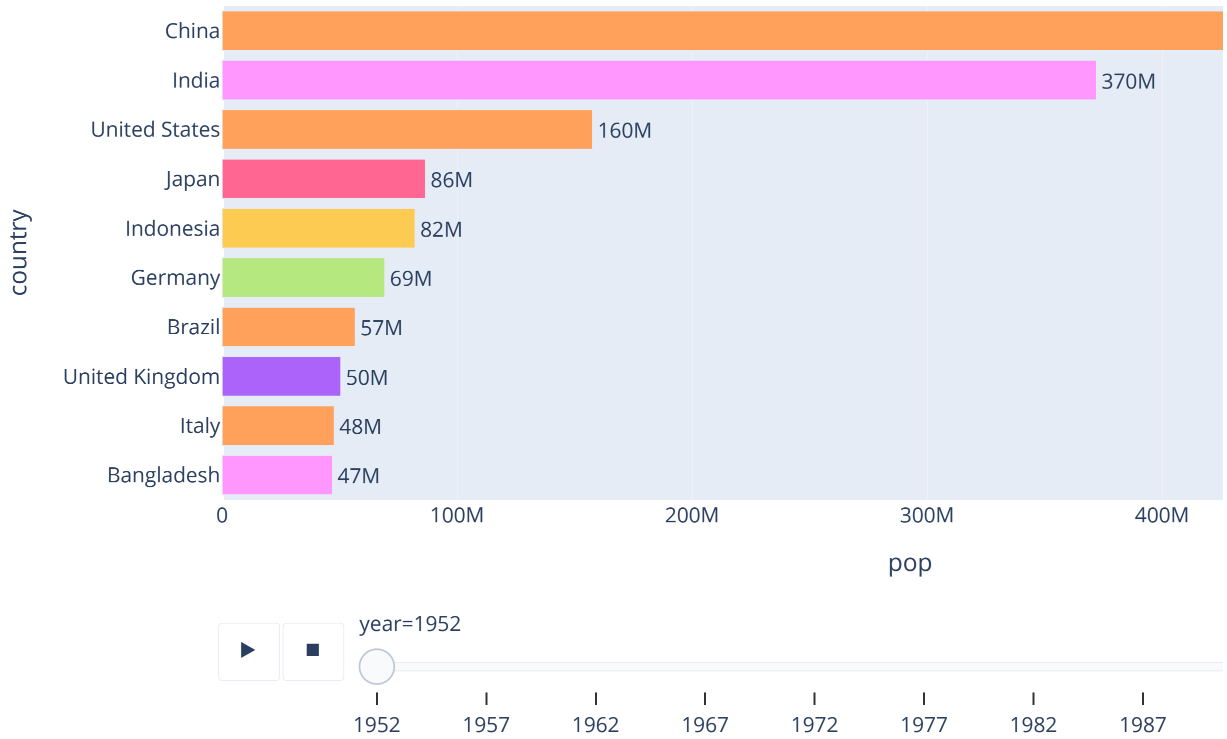
Hint: Use the axis limit to set this.

In [67]:

```
df_pop_in_year = df.groupby(['year', 'country']).sum(numeric_only=True).reset_index()
fig = px.bar(df_pop_in_year, x='pop', y='country', orientation='h', text = 'pop',
             color='country', animation_frame='year', animation_group='country')
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder = "total descending", range=[9.5,-0.5])
fig.update_traces(texttemplate='%{text:.2s}', textposition='outside')

fig.update_layout(autosize=False,
    width=1000,
    height=500,)

fig.show()
```



In [ ]: