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.

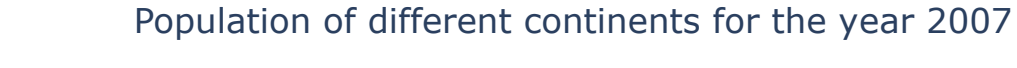
Hints:

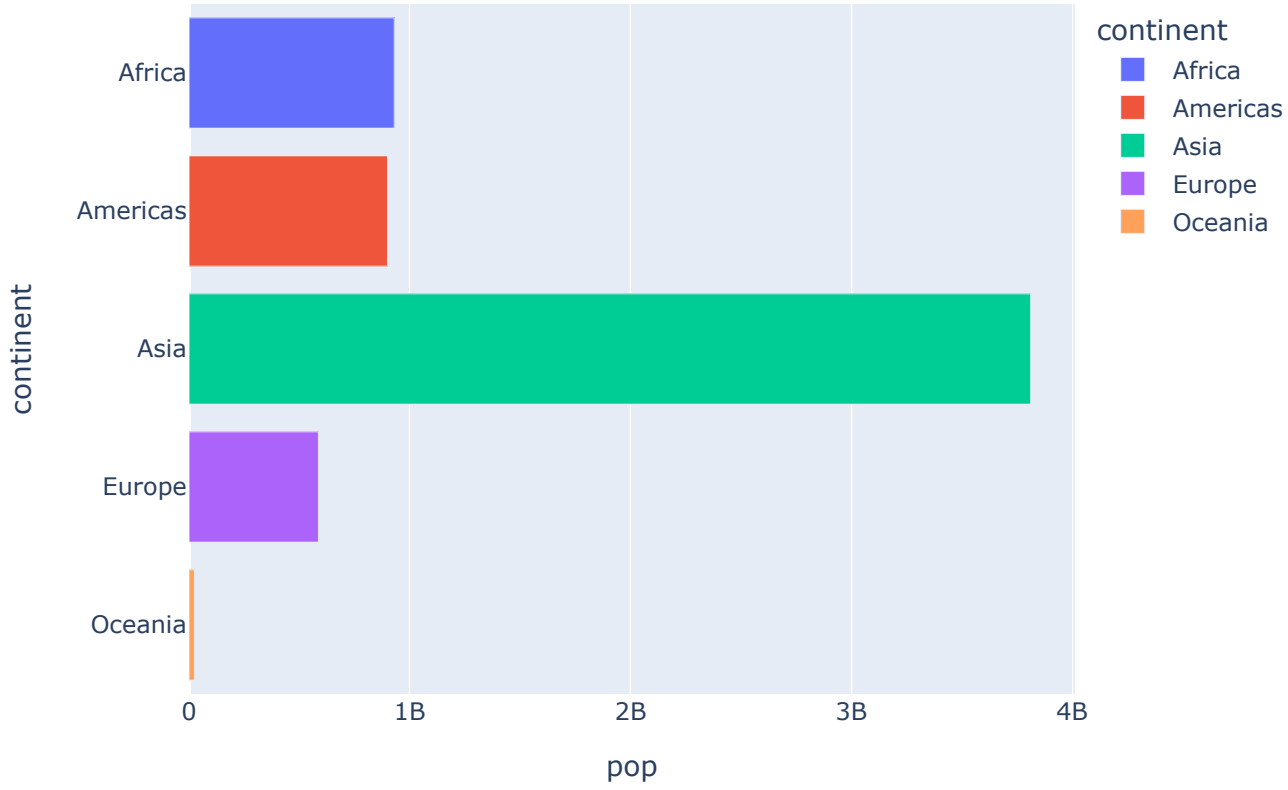
- Extract the 2007 year data from the dataframe. You have to process the data accordingly
- use plotly bar
- Add different colors for different continents
- Sort the order of the continent for the visualisation. Use axis layout setting
- Add text to each bar that represents the population

In [3]:
```
df_2007 = df.query('year==2007')
df_2007_new = df_2007.groupby('continent').sum()

fig = px.bar(df_2007_new, x='pop', y=df_2007_new.index, color=df_2007_new.index,
             title="Population of different continents for the year 2007",
             )

fig.show()
```





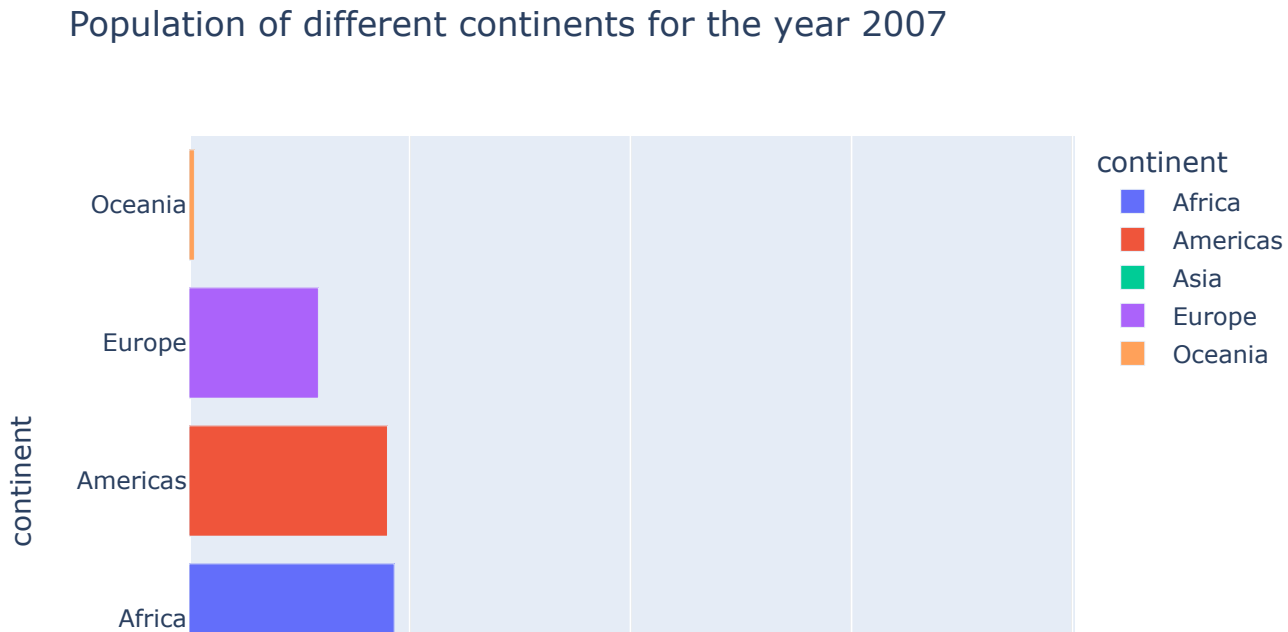
Question 2:

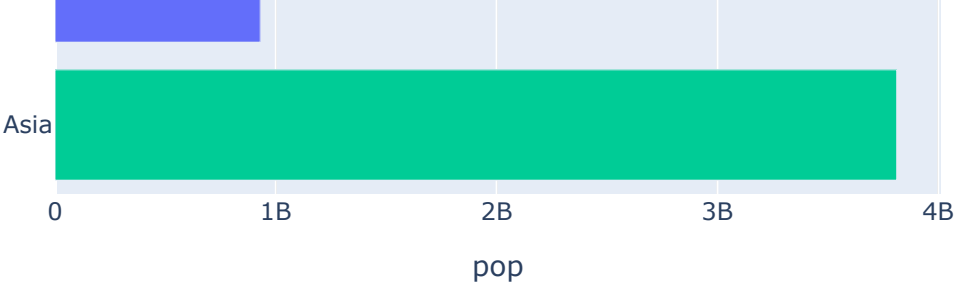
Sort the order of the continent for the visualisation

Hint: Use axis layout setting

In [4]:
```
fig = px.bar(df_2007_new, x='pop', y=df_2007_new.index, color=df_2007_new.index,
             title="Population of different continents for the year 2007",
            )
fig.update_layout(yaxis={'categoryorder':'total descending'})


fig.show()
```



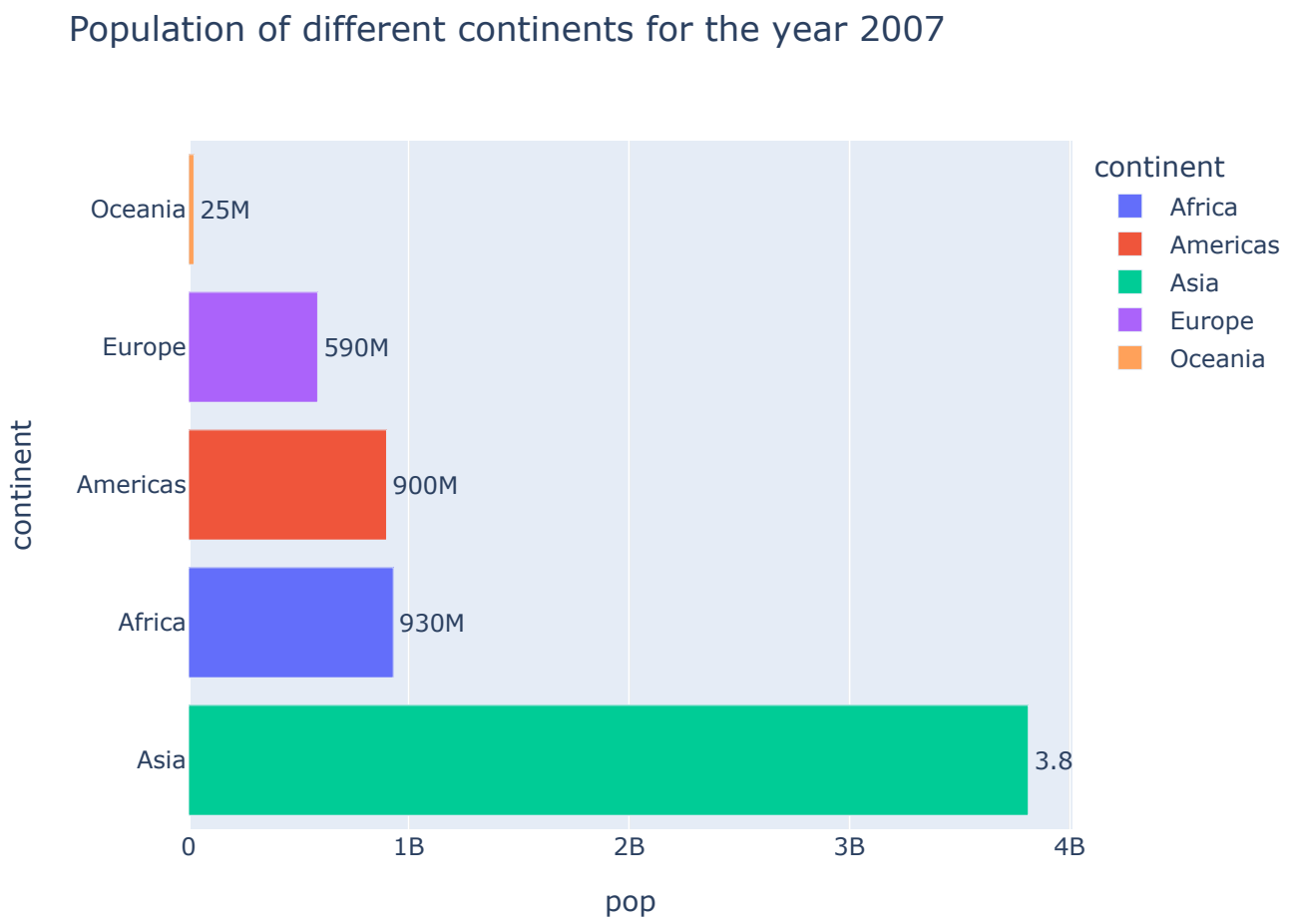


Question 3:

Add text to each bar that represents the population

In [5]:
```
fig = px.bar(df_2007_new, x='pop', y=df_2007_new.index, color=df_2007_new.index,
             text='pop', title="Population of different continents for the year 2007",
             text_auto='.2s')
fig.update_layout(yaxis={'categoryorder':'total descending'})
fig.update_traces(textposition='outside')

fig.show()
```



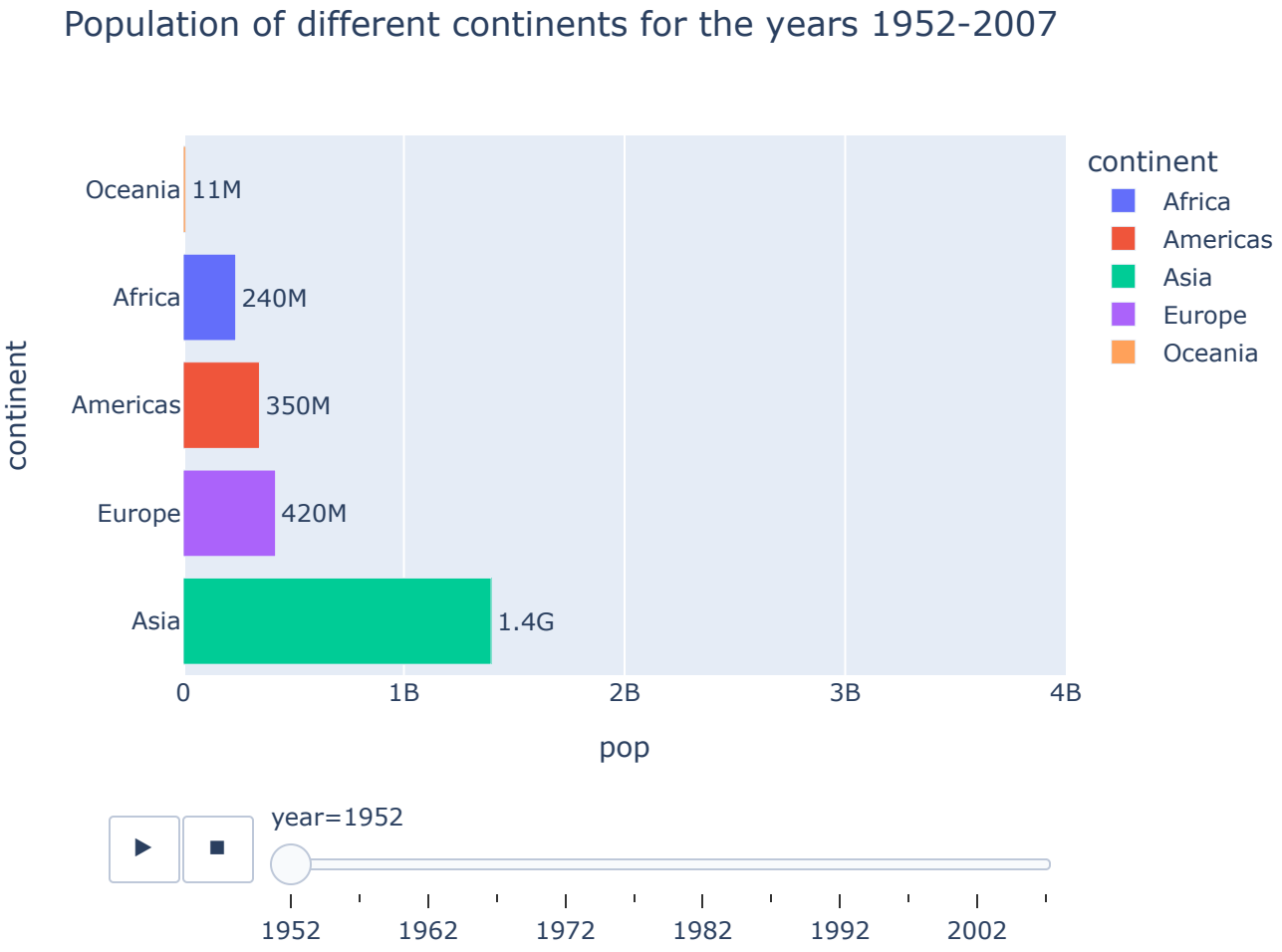
Question 4:

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

In [6]:
```
df_new = df.groupby(['year', 'continent']).sum().reset_index()

fig = px.bar(df_new, x='pop', y='continent', color='continent', animation_frame='year',
             text='pop', title="Population of different continents for the years 1952-20
             text_auto='.2s')
fig.update_layout(xaxis_range=[0,4000000000], yaxis={'categoryorder':'total descending'}
fig.update_traces(textposition='outside')

fig.show()
```



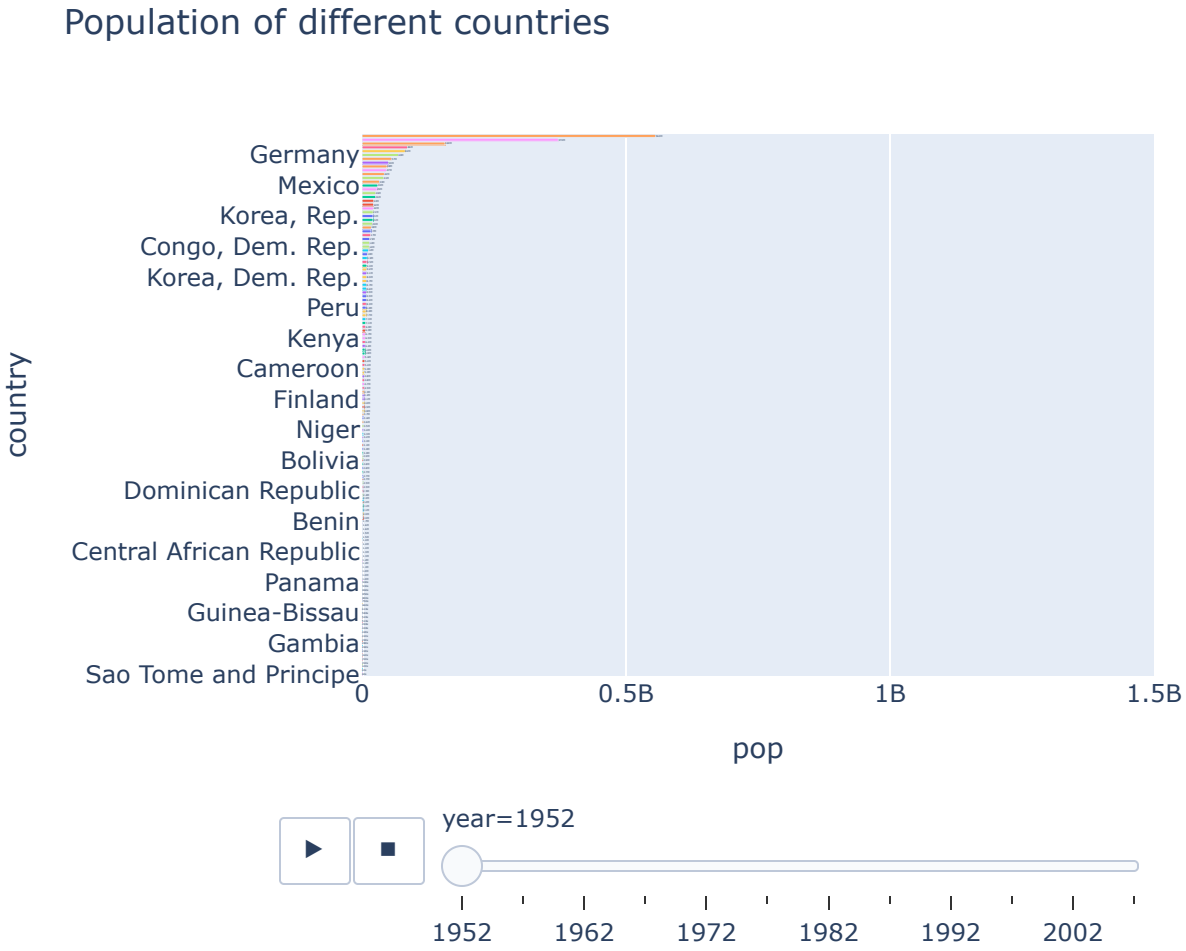
Question 5:

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

In [7]:
```
df_countries = df.groupby(["year", "country"]).sum().reset_index()

fig = px.bar(df_countries, x='pop', y='country', color='country', animation_frame='year'
             text='pop', title='Population of different countries',
             text_auto='.2s')
fig.update_layout(xaxis_range=[0,1500000000], yaxis={'categoryorder':'total ascending'},
fig.update_traces(textposition='outside')

fig.show()
```

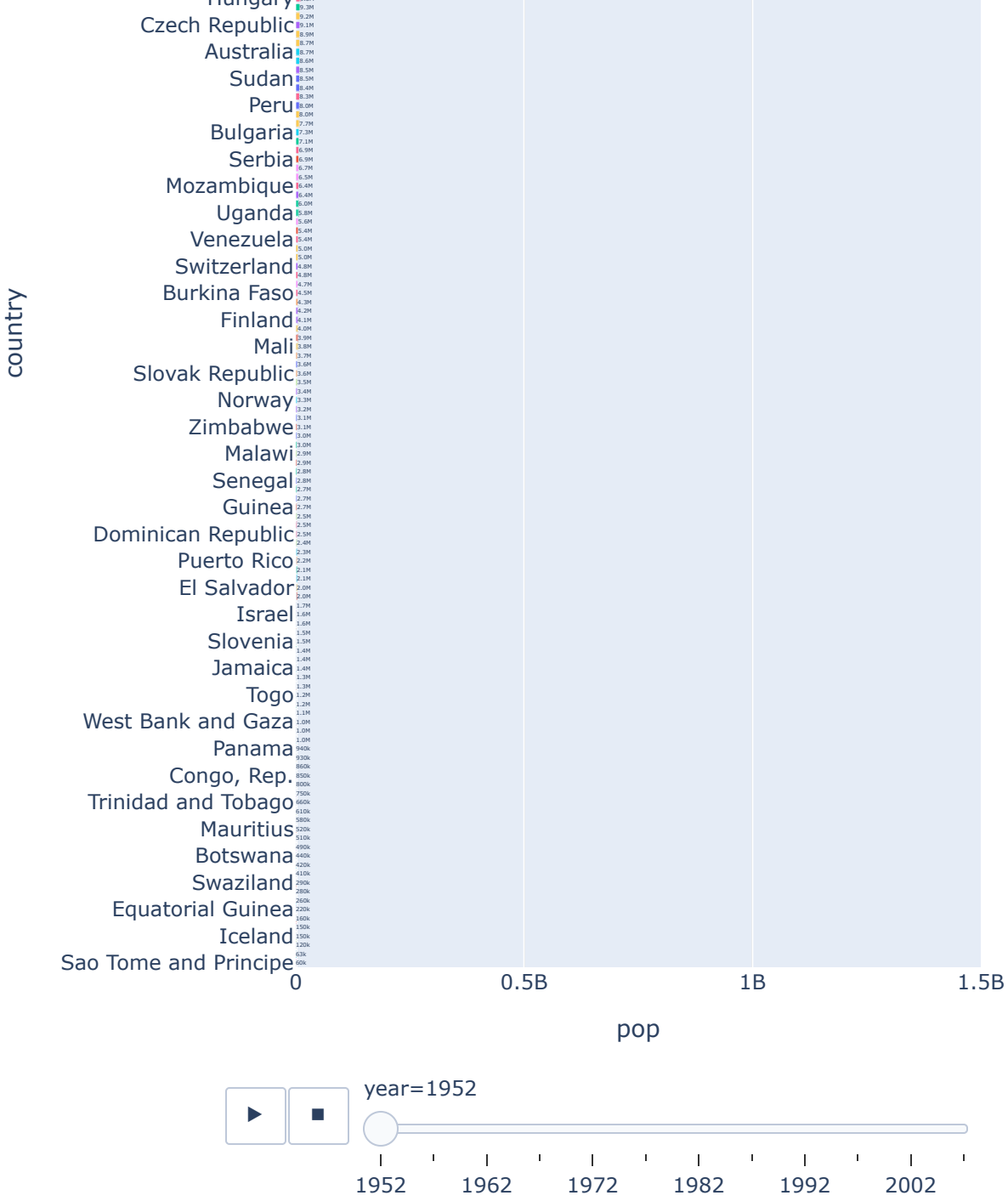


Question 6:

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

In [8]:
```
fig = px.bar(df_countries, x='pop', y='country', color='country', animation_frame='year'
             text='pop', title='Population of different countries',
             text_auto='.2s', height=1000)
fig.update_layout(xaxis_range=[0,1500000000], yaxis={'categoryorder':'total ascending'},
fig.update_traces(textposition='outside')

fig.show()
```



Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

In [9]:
```
df.groupby(["country"]).sum()
# so there are 141 countries, top 10 is country 132-141 if dataset is acending
```

Out[9]:

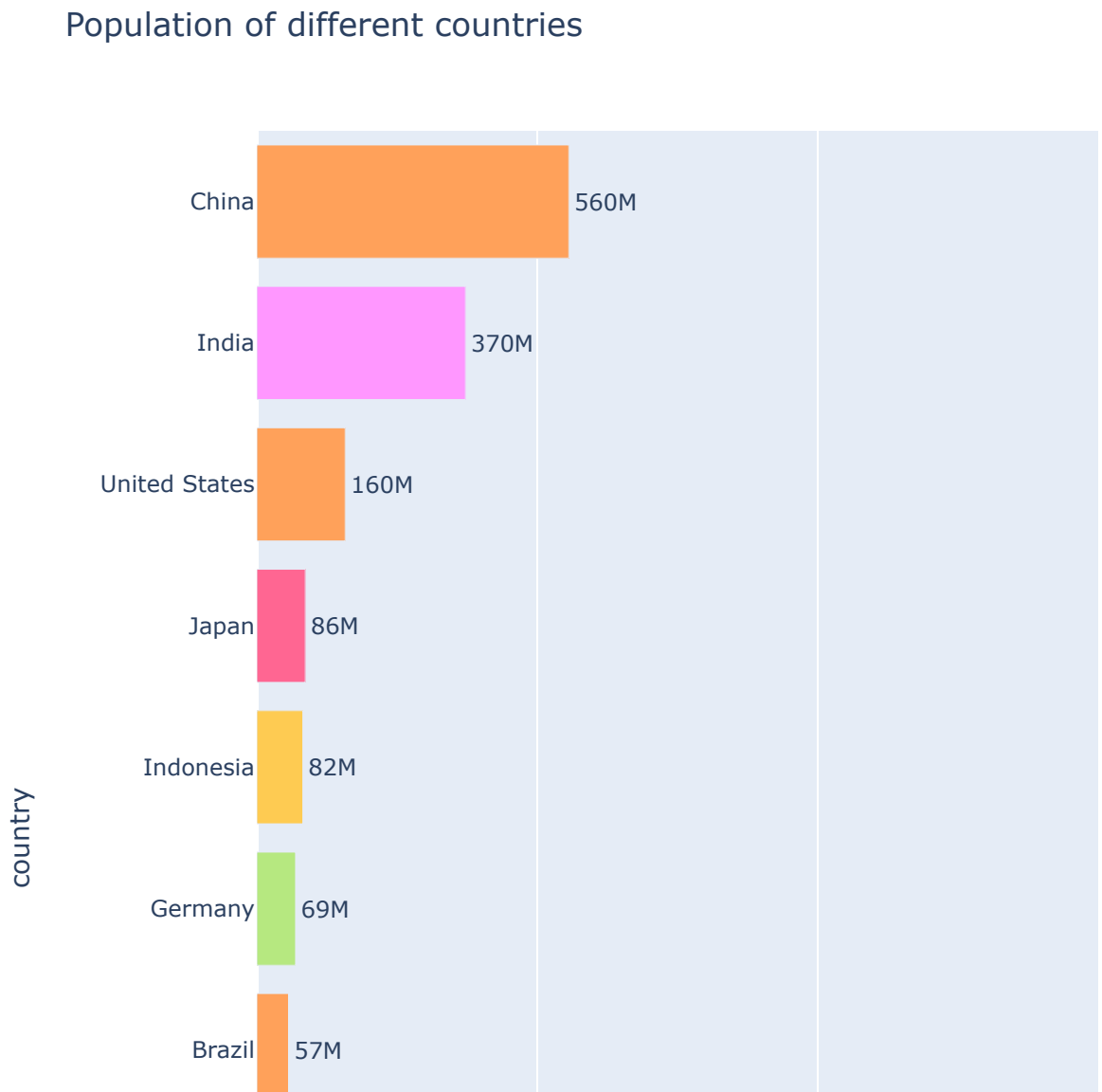
country	year	lifeExp	pop	gdpPercap	iso_num
Afghanistan	23754	449.746	189884585	9632.095181	48
Albania	23754	821.195	30962990	39064.399592	96

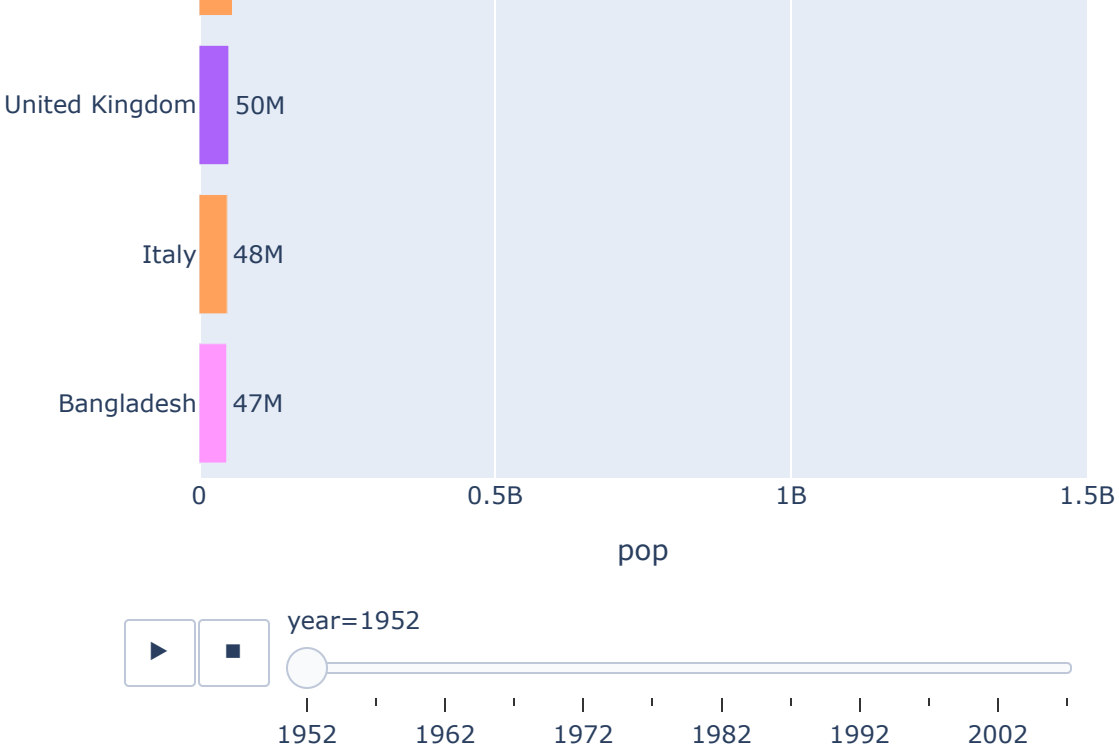
Algeria	23754	708.362	238504874	53112.311678	144
Angola	23754	454.602	87712681	43285.206346	288
Argentina	23754	828.725	343226879	107466.645392	384
...	...	...	...	...	...
Vietnam	23754	689.754	654822851	12212.551382	8448
West Bank and Gaza	23754	723.944	22183278	45119.961375	3300
Yemen, Rep.	23754	561.365	130118302	18831.296066	10644
Zambia	23754	551.956	76245658	16298.392908	10728
Zimbabwe	23754	631.958	91703593	7630.296508	8592

142 rows × 5 columns

In [10]:
```
fig = px.bar(df_countries, x='pop', y='country', color='country', animation_frame='year'
             text='pop', title='Population of different countries',
             text_auto='.2s', height=1000)
fig.update_layout(xaxis_range=[0,1500000000], yaxis={'categoryorder':'total ascending'},
fig.update_traces(textposition='outside')
fig.update_yaxes(range=(131.5, 141.5))

fig.show()
```




United Kingdom
50M
Italy
48M
Bangladesh
47M
0
0.5B
1B
1.5B
pop

year=1952
1952
1962
1972
1982
1992
2002

In [ ]: