## 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.

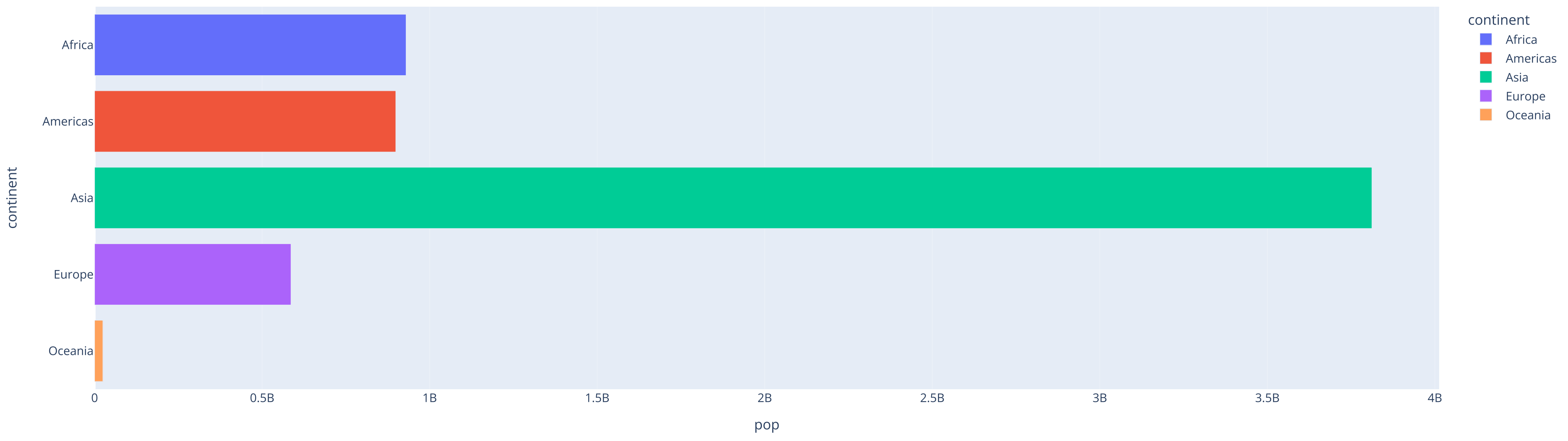
*Hints:*

- 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
df_2007 = df[df['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, labels=df_2007_new.index, orientation='h')
fig.show()
```

```
C:\Users\senha\AppData\Local\Temp\ipykernel_153148\176774304.py:3: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.sum is deprecated.  In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.

C:\Users\senha\anaconda3\Lib\site-packages\plotly\express\_core.py:137: FutureWarning:

Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy array before indexing instead.
```



## Question 2:

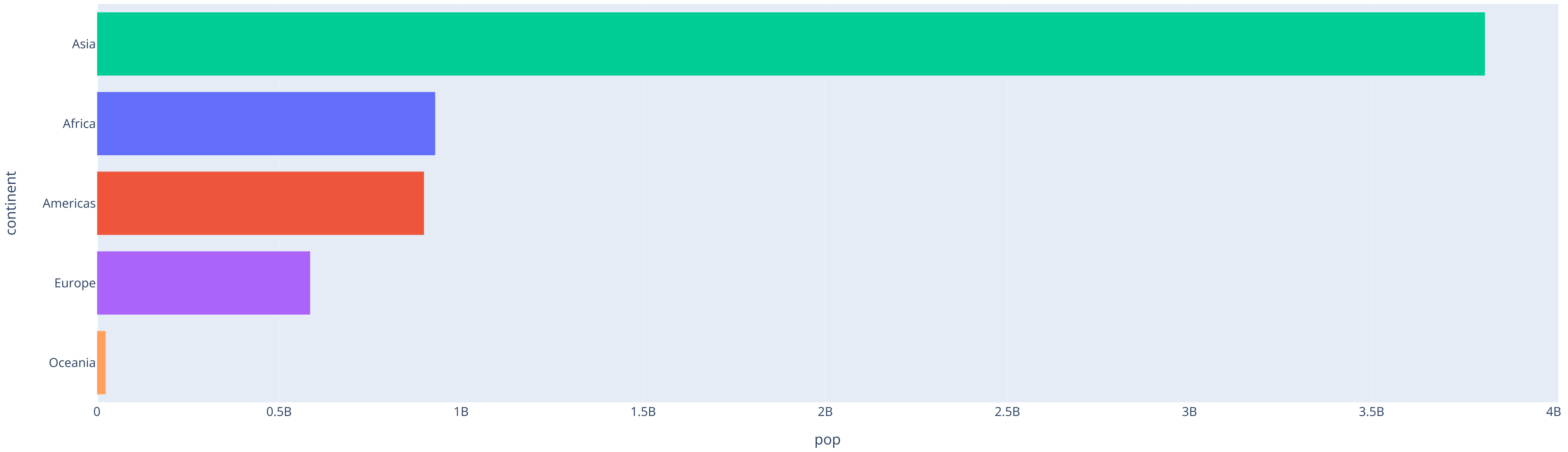
Sort the order of the continent for the visualisation

Hint: Use axis layout setting

In [4]:

```
df_2007 = df[df['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, labels=df_2007_new.index, orientation='h')
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.show()
```

```
C:\Users\senha\AppData\Local\Temp\ipykernel_153148\2998428336.py:2: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.sum is deprecated.  In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.

C:\Users\senha\anaconda3\Lib\site-packages\plotly\express\_core.py:137: FutureWarning:

Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy array before indexing instead.
```



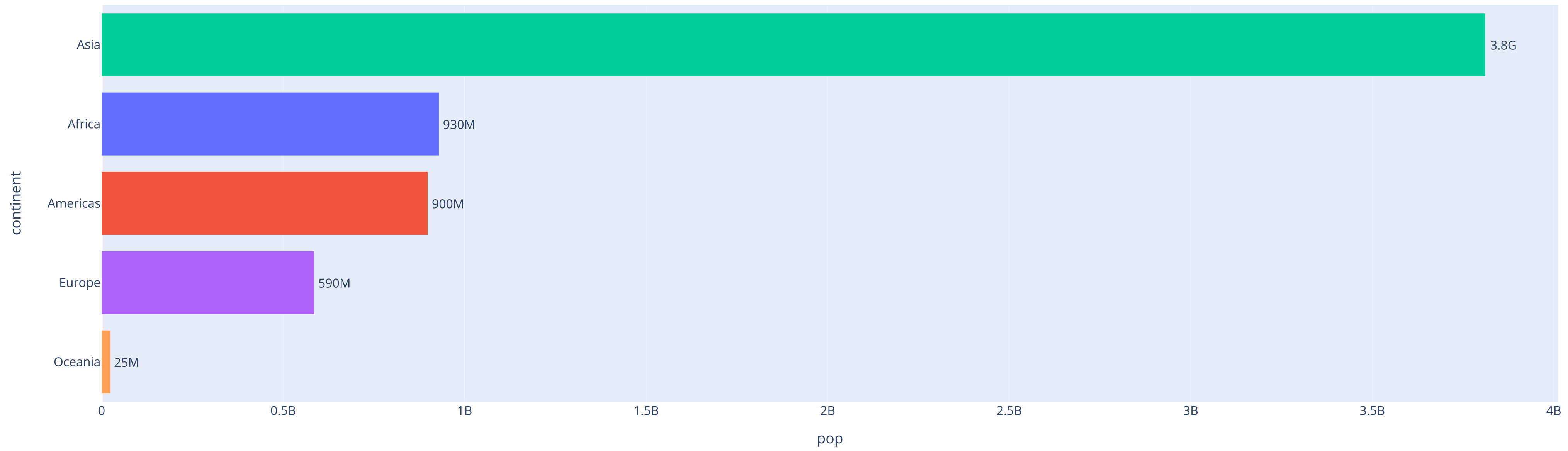
## Question 3:

Add text to each bar that represents the population

In [5]:

```
df_2007 = df[df['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, labels=df_2007_new.index, orientation='h',text_auto='.2s')
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=False)
fig.show()
```

```
C:\Users\senha\AppData\Local\Temp\ipykernel_153148\4188970760.py:2: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.sum is deprecated.  In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.

C:\Users\senha\anaconda3\Lib\site-packages\plotly\express\_core.py:137: FutureWarning:

Support for multi-dimensional indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy array before indexing instead.
```

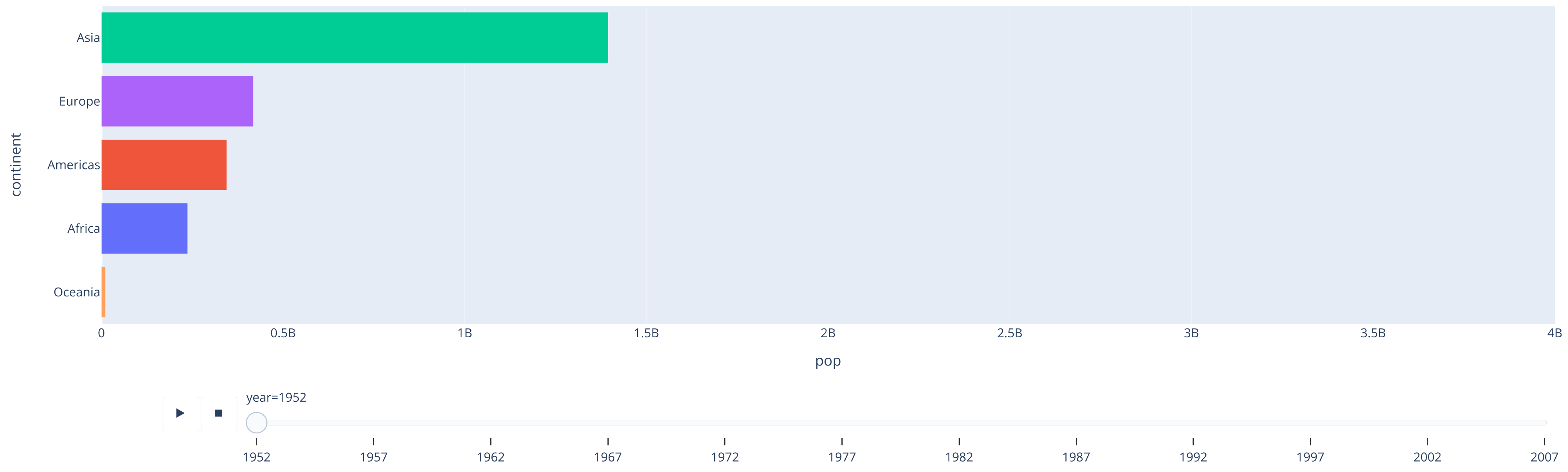


## Question 4:

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", orientation='h',
        animation_frame="year", animation_group="continent", range_x=[0,4000000000])
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=False)
fig.show()
```

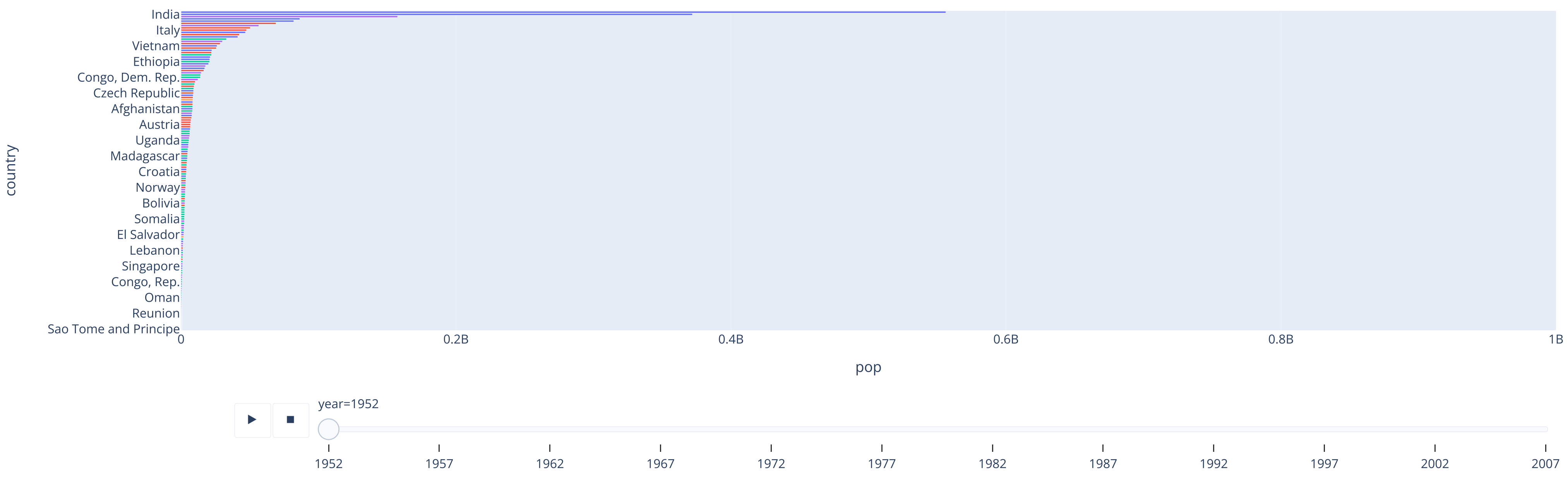
```
C:\Users\senha\AppData\Local\Temp\ipykernel_153148\3444603529.py:1: FutureWarning:

The default value of numeric_only in DataFrameGroupBy.sum is deprecated.  In a future version, numeric_only will default to False. Either specify numeric_only or select only columns which should be valid for the function.
```



## 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]:
```
# YOUR CODE HERE
fig = px.bar(df, x="pop", y="country", color="continent", orientation='h',
        animation_frame="year", animation_group="country", range_x=[0,1000000000])
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=False)
fig.show()
```

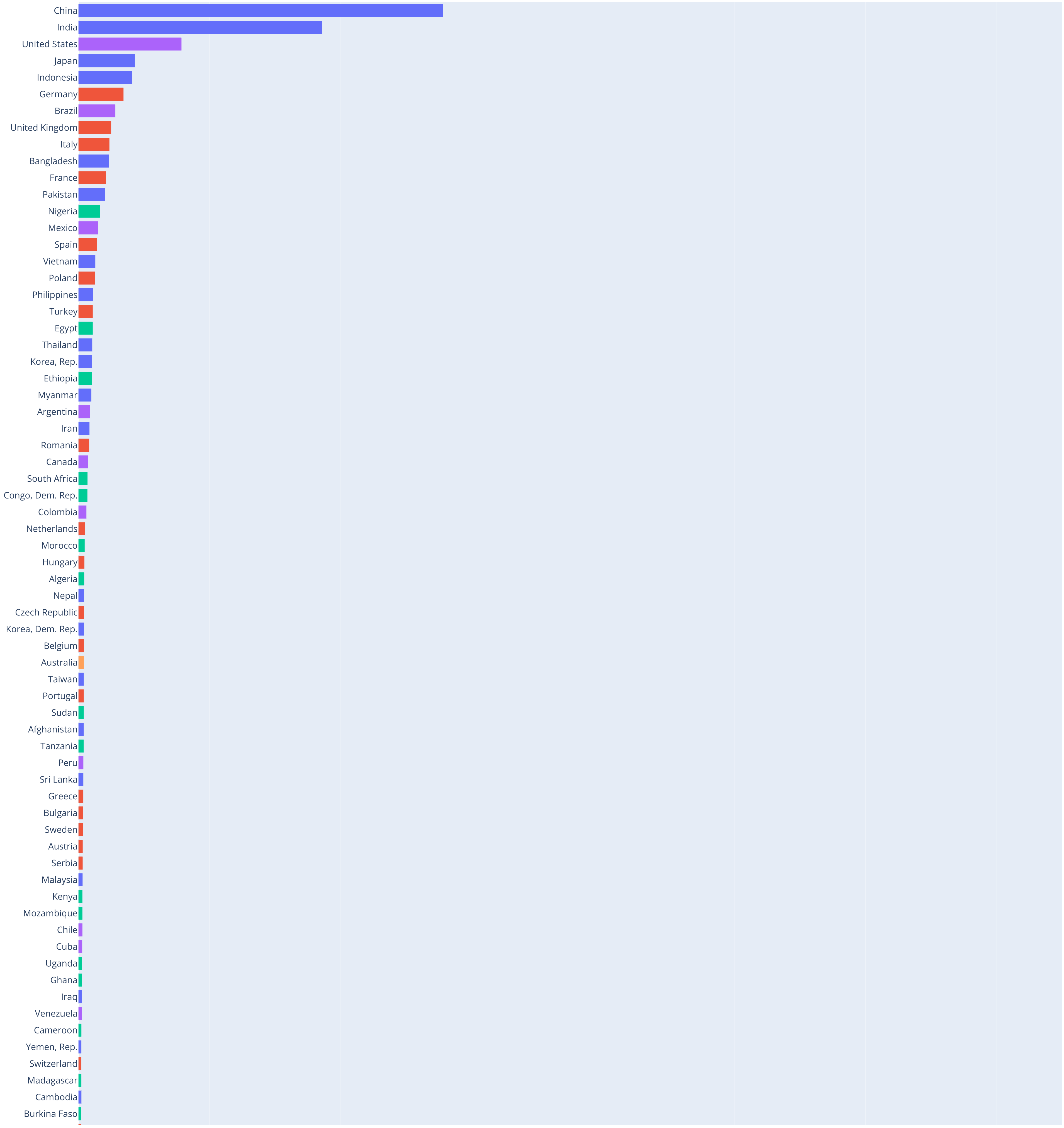


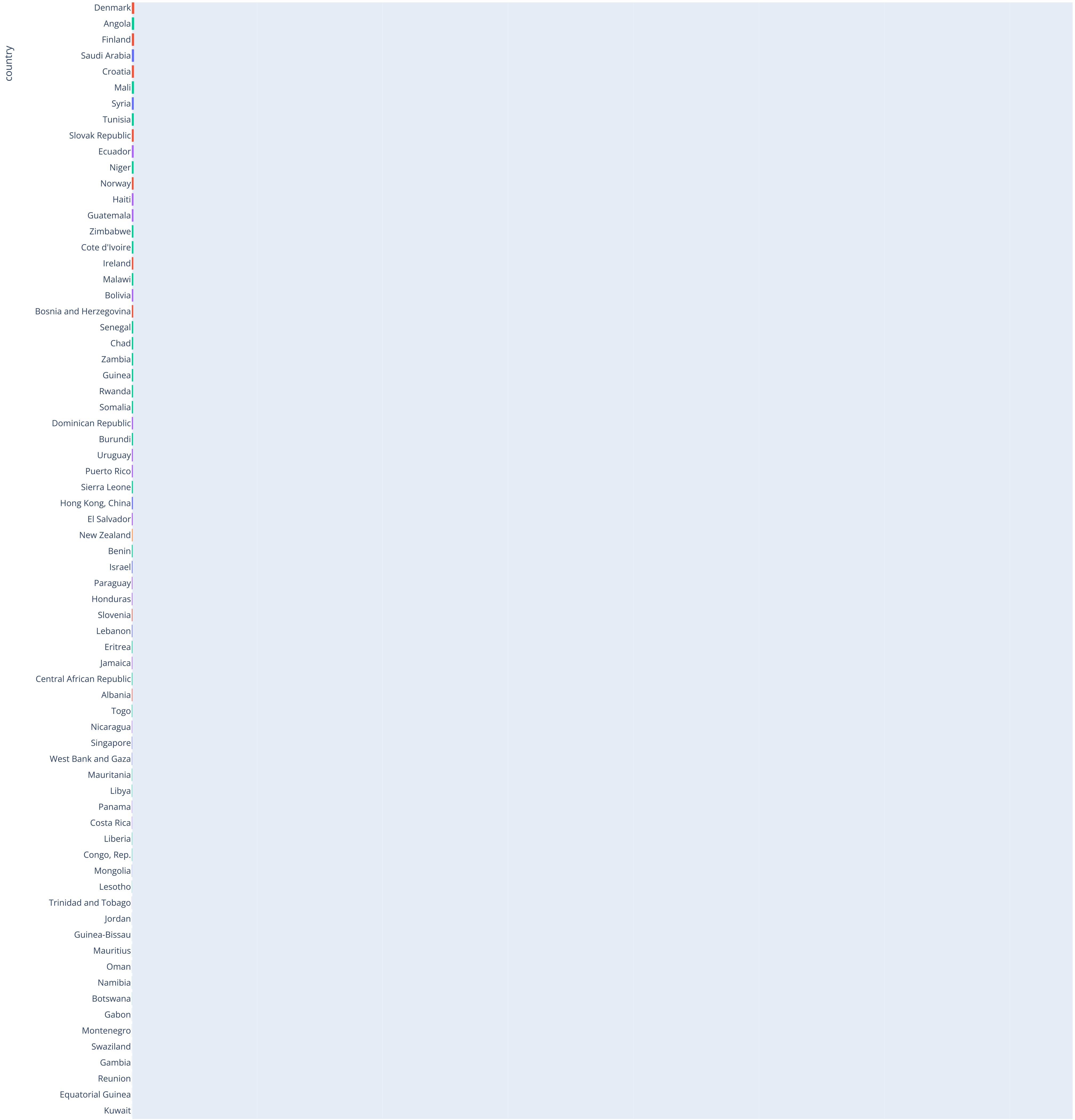
## Question 6:

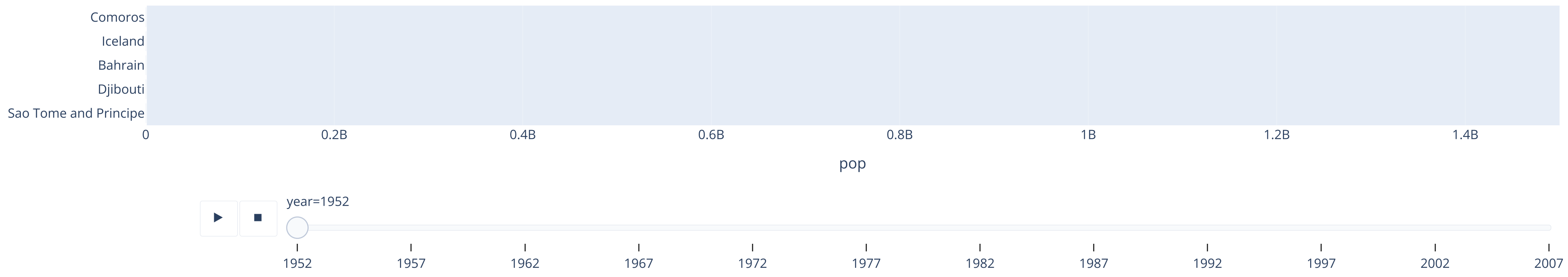
Clean up the country animation. Set the height size of the figure to 1000 to have a better view of the animation

In [8]:

```
# YOUR CODE HERE

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

fig.update_layout(height=len(df) * 2  )
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=False)
fig.show()
```


China
India
United States
Japan
Indonesia
Germany
Brazil
United Kingdom
Italy
Bangladesh
France
Pakistan
Nigeria
Mexico
Spain
Vietnam
Poland
Philippines
Turkey
Egypt
Thailand
Korea, Rep.
Ethiopia
Myanmar
Argentina
Iran
Romania
Canada
South Africa
Congo, Dem. Rep.
Colombia
Netherlands
Morocco
Hungary
Algeria
Nepal
Czech Republic
Korea, Dem. Rep.
Belgium
Australia
Taiwan
Portugal
Sudan
Afghanistan
Tanzania
Peru
Sri Lanka
Greece
Bulgaria
Sweden
Austria
Serbia
Malaysia
Kenya
Mozambique
Chile
Cuba
Uganda
Ghana
Iraq
Venezuela
Cameroon
Yemen, Rep.
Switzerland
Madagascar
Cambodia
Burkina Faso


country
Denmark
Angola
Finland
Saudi Arabia
Croatia
Mali
Syria
Tunisia
Slovak Republic
Ecuador
Niger
Norway
Haiti
Guatemala
Zimbabwe
Cote d'Ivoire
Ireland
Malawi
Bolivia
Bosnia and Herzegovina
Senegal
Chad
Zambia
Guinea
Rwanda
Somalia
Dominican Republic
Burundi
Uruguay
Puerto Rico
Sierra Leone
Hong Kong, China
El Salvador
New Zealand
Benin
Israel
Paraguay
Honduras
Slovenia
Lebanon
Eritrea
Jamaica
Central African Republic
Albania
Togo
Nicaragua
Singapore
West Bank and Gaza
Mauritania
Libya
Panama
Costa Rica
Liberia
Congo, Rep.
Mongolia
Lesotho
Trinidad and Tobago
Jordan
Guinea-Bissau
Mauritius
Oman
Namibia
Botswana
Gabon
Montenegro
Swaziland
Gambia
Reunion
Equatorial Guinea
Kuwait

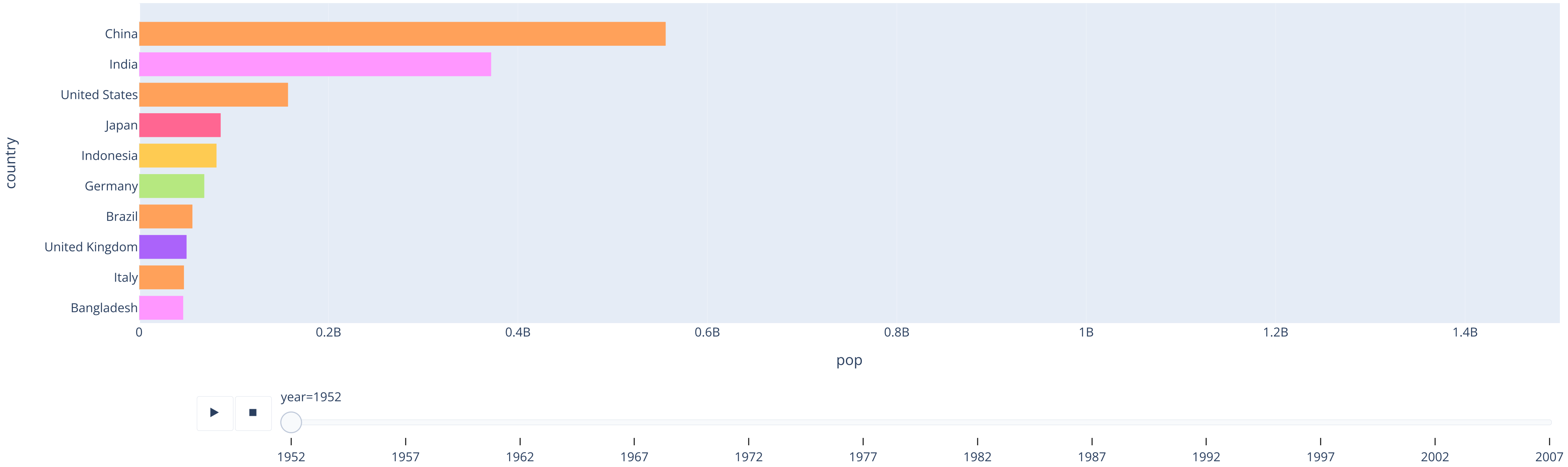


## Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

In [11]:
```
# YOUR CODE HERE
df_pop = df.groupby(['year','country'])['pop'].sum()
df_pop = df_pop.to_frame()
df_pop = df_pop.reset_index()
country_count = df.groupby(['country'])['pop'].sum()

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

# fig.update_layout(height=len(df) * 2  )
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=False)
fig.update_yaxes(range=(len(country_count)-10.5, len(country_count)))
fig.show()
```


China
India
United States
Japan
Indonesia
Germany
Brazil
United Kingdom
Italy
Bangladesh
country
0.2B
0.4B
0.6B
0.8B
1B
1.2B
1.4B
pop
year=1952