

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 [23]: #Load data
df2 = px.data.gapminder()
df2.head()
```

Out[23]:

	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:

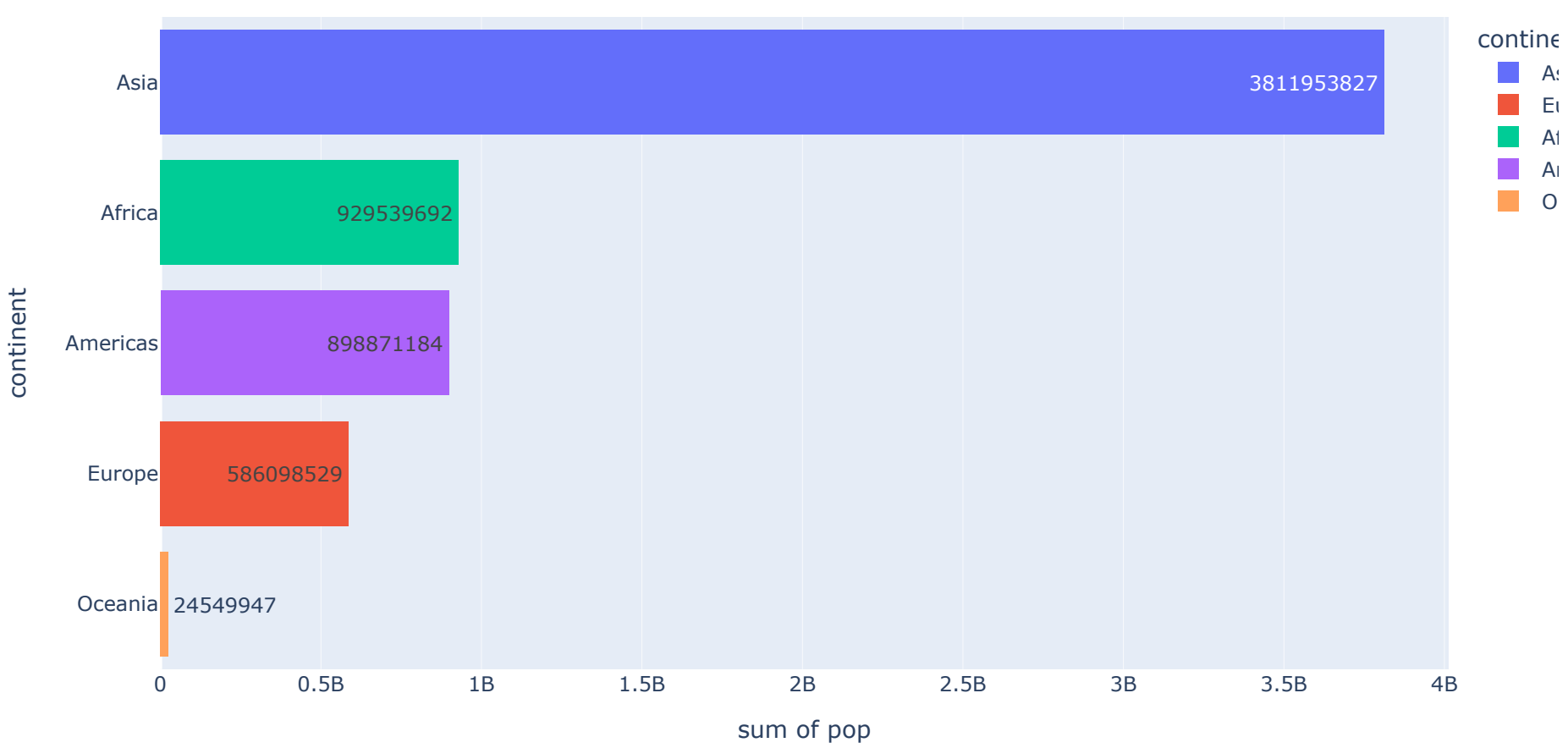
Recreate the barplot below that shows the population of different continents for the year 2007.

- Hints:
- Extract the 2007 year data from the dataframe. You have to process the data accordingly
 - use `plotly_bar` (<https://plotly.com/python-api-reference/generated/plotly.express.bar>)
 - Add different colors for different continents
 - Sort the order of the continent for the visualisation. Use `axis layout setting` (<https://plotly.com/python/reference/layout/xaxis/>)
 - Add text to each bar that represents the population

```
In [10]: df = px.data.gapminder().query('year == 2007')

fig = px.histogram(df, x="pop", y="continent", color="continent", text_auto="TRUE")
fig.update_yaxes(categoryorder='total ascending')

fig.show()
```



Question 2:

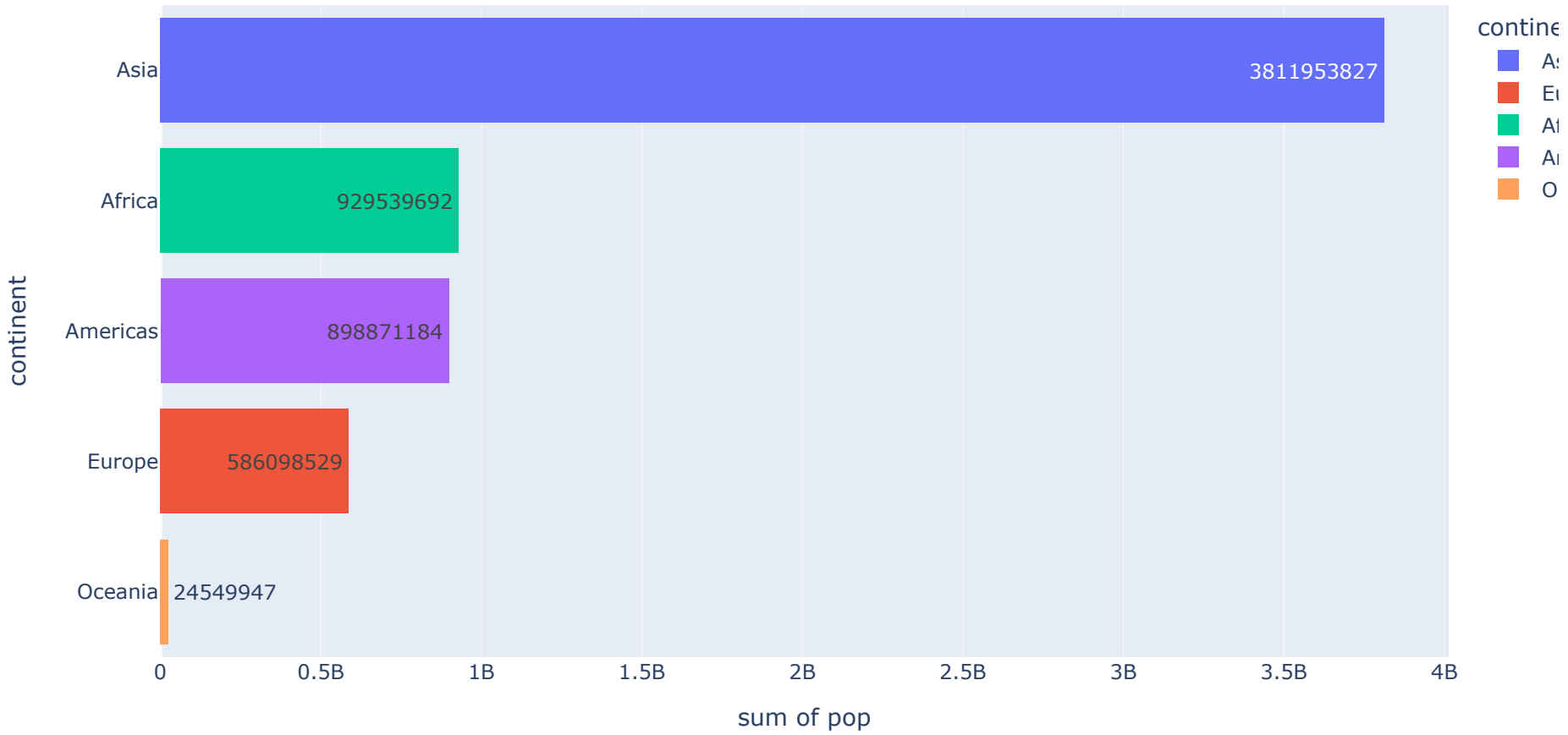
Sort the order of the continent for the visualisation

Hint: Use [axis layout setting \(https://plotly.com/python/reference/layout/xaxis/\)](https://plotly.com/python/reference/layout/xaxis/)

```
In [11]: df = px.data.gapminder().query('year == 2007')

fig = px.histogram(df, x="pop", y="continent", color="continent", text_auto="TRUE")
fig.update_yaxes(categoryorder='total ascending')

fig.show()
```



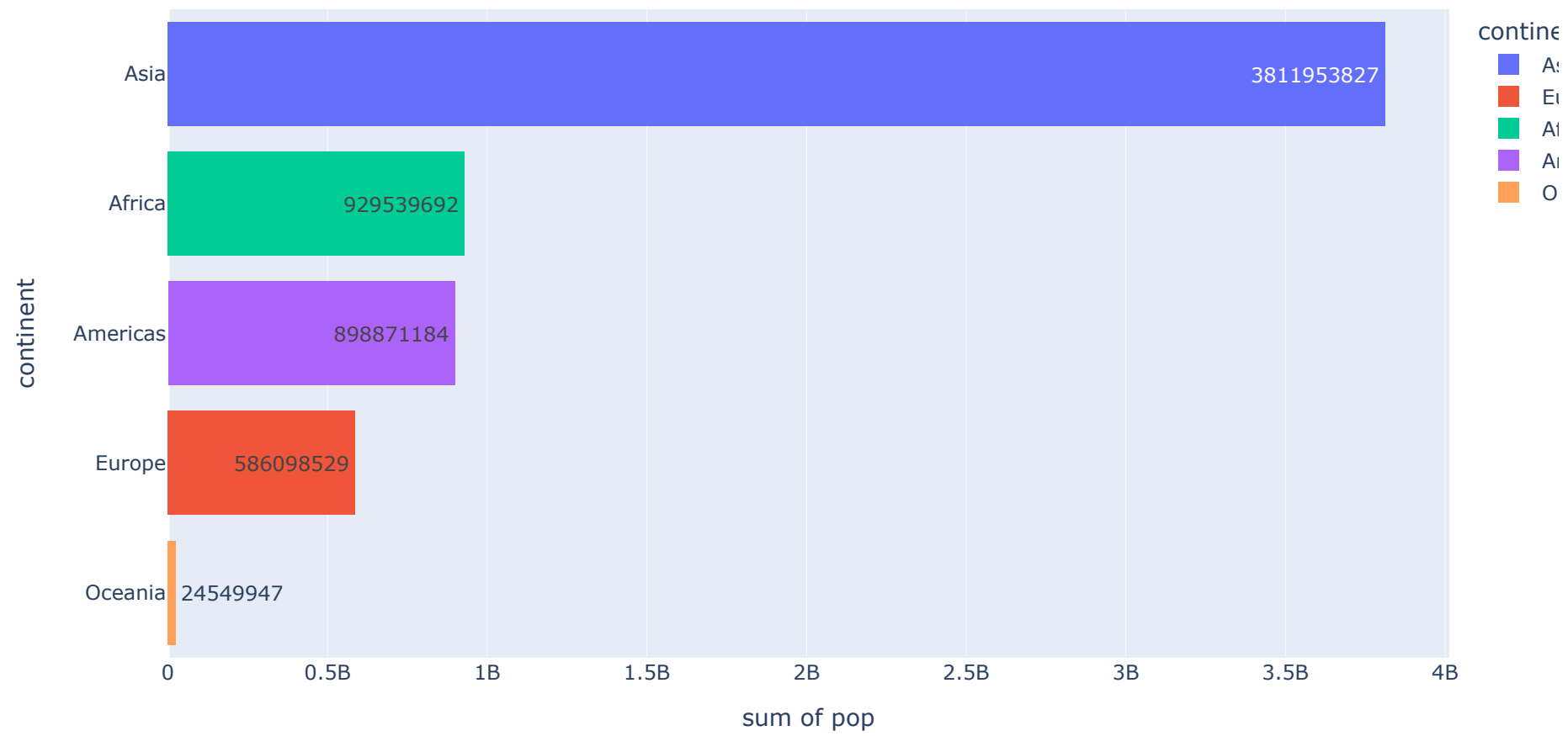
Question 3:

Add text to each bar that represents the population

```
In [12]: df = px.data.gapminder().query('year == 2007')

fig = px.histogram(df, x="pop", y="continent", color="continent", text_auto="TRUE")
fig.update_yaxes(categoryorder='total ascending')

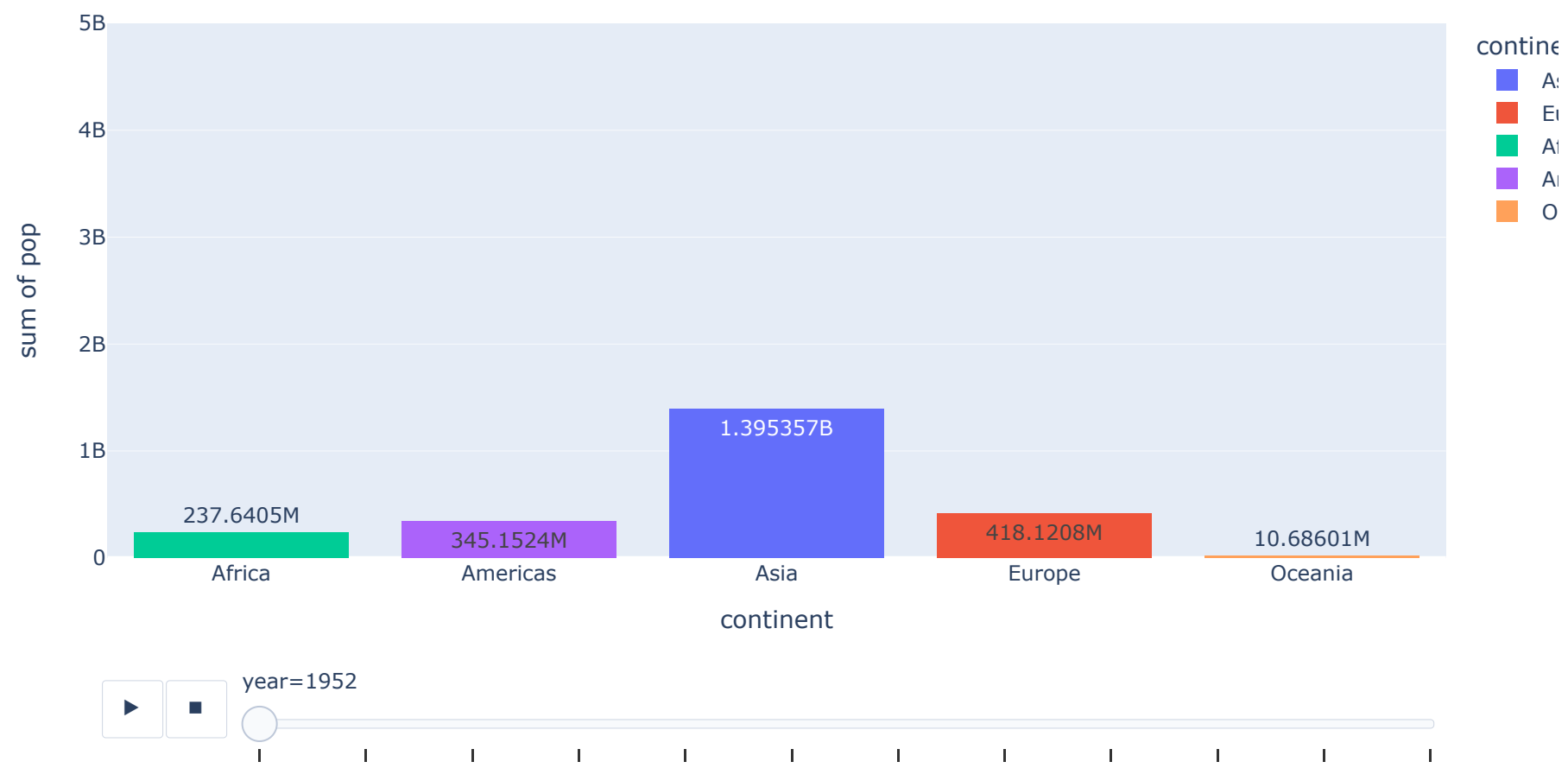
fig.show()
```



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 [26]: fig = px.histogram(df2, x='continent', y='pop', color='continent', text_auto=True, animation_frame="year", animation_group="continent")
fig.update_xaxes(categoryorder="category ascending")
fig.show()
```

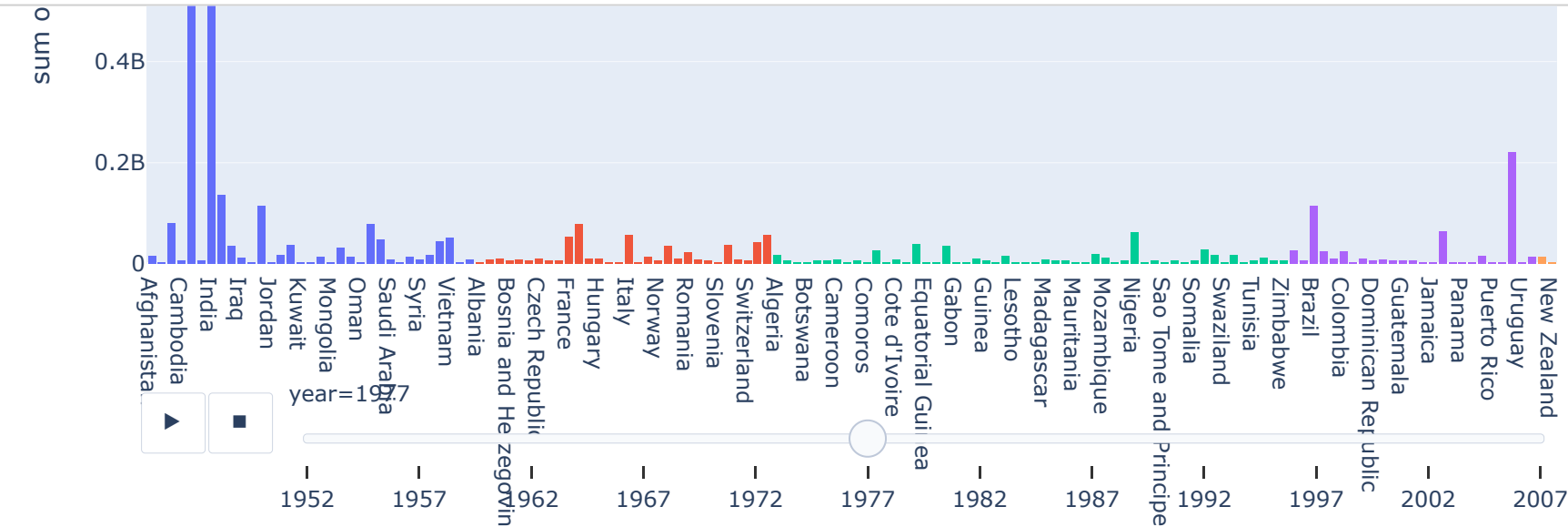


Question 5:

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

```
In [31]: fig = px.histogram(df2, x='country', y='pop', color='continent', text_auto=False, animation_frame="year", range_y=[0,500000000])
#fig.update_xaxes(categoryorder="category ascending")
```

fig.show()

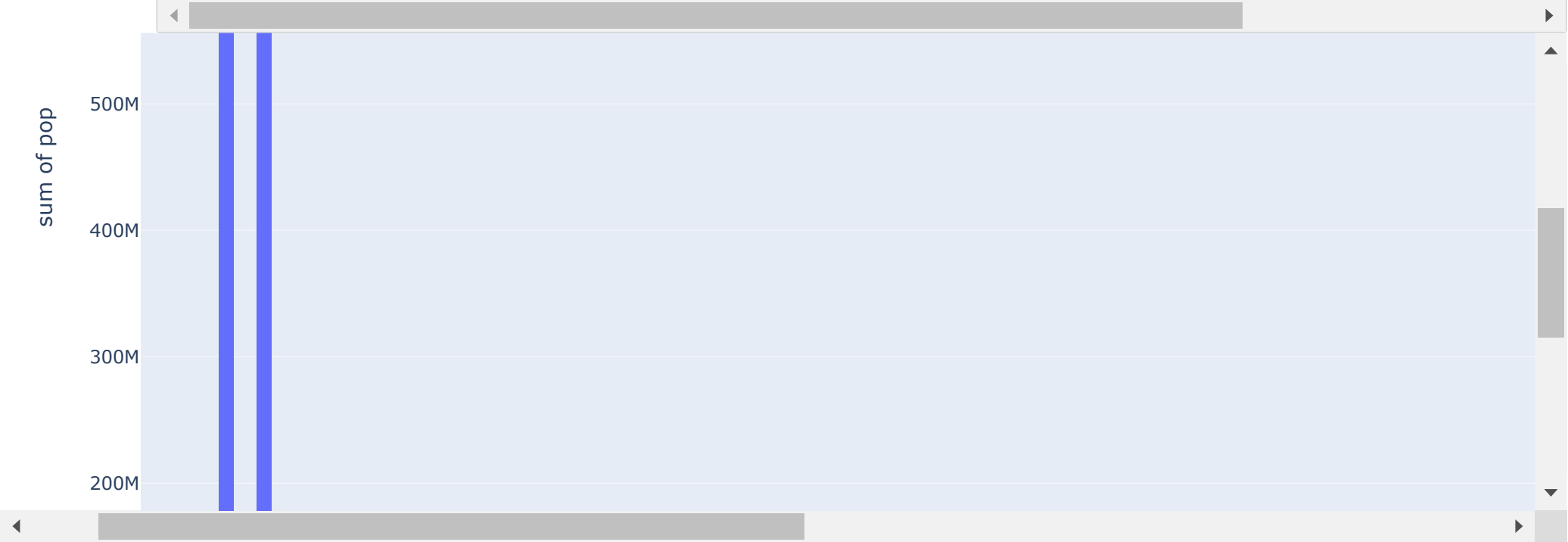


Question 6:

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

```
In [40]: fig = px.histogram(df2, x='country', y='pop', color='continent', text_auto=False, animation_frame="year", range_y=[0,900000000])
#fig.update_xaxes(categoryorder="category ascending")
```

fig.show()

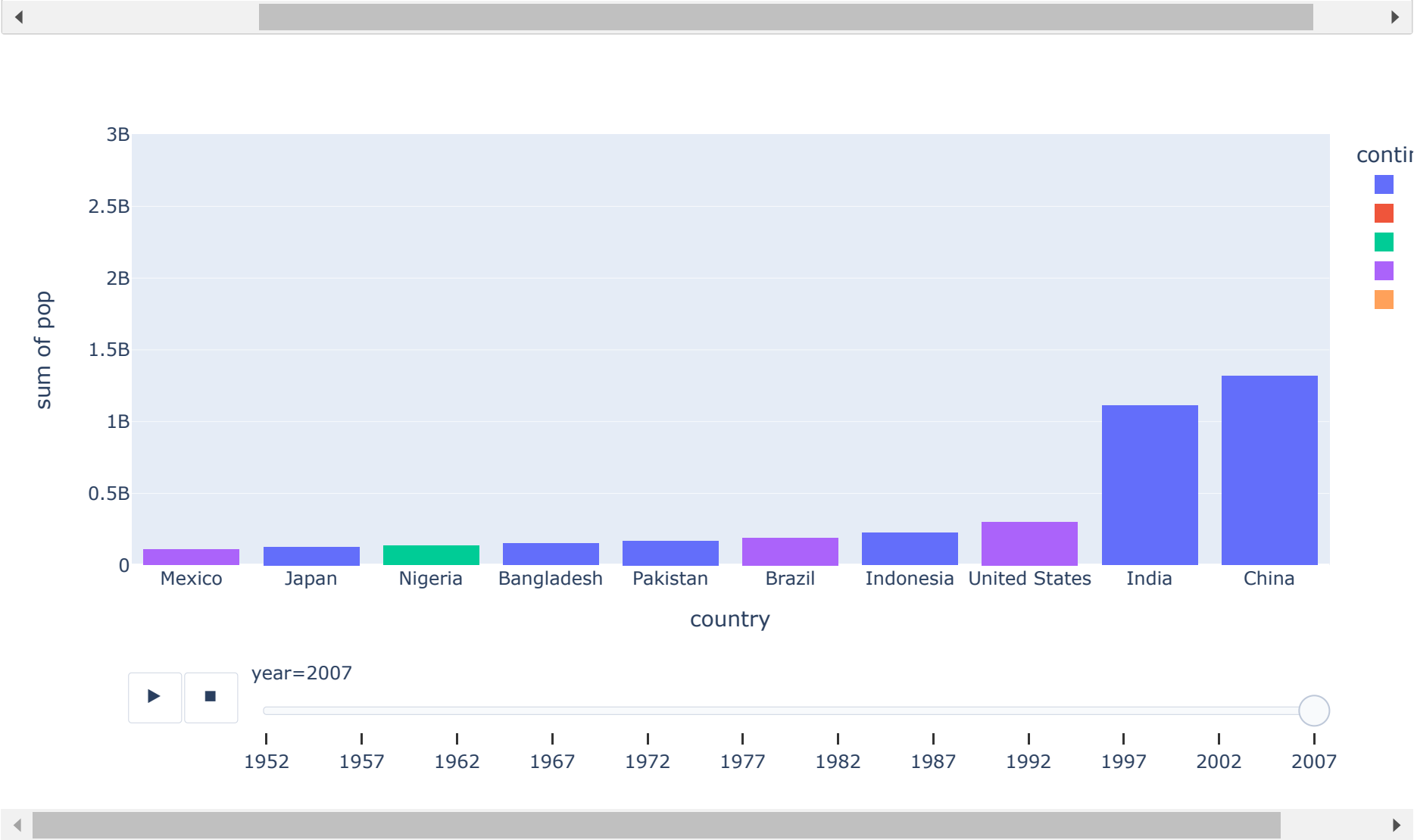


Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

```
In [55]: = 'country', y='pop', color='continent', text_auto=False, animation_frame="year", range_y=[0,3000000000], height=500, width=31.5, 141.5))
order="total ascending")
```



```
In [ ]:
```