LabSession_AdvancedViz

October 3, 2022

1 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

[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"
```

[2]:

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

[2]:

```
       country continent  year  lifeExp       pop   gdpPercap iso_alpha  \
0  Afghanistan      Asia  1952   28.801   8425333  779.445314       AFG
1  Afghanistan      Asia  1957   30.332   9240934  820.853030       AFG
2  Afghanistan      Asia  1962   31.997  10267083  853.100710       AFG
3  Afghanistan      Asia  1967   34.020  11537966  836.197138       AFG
4  Afghanistan      Asia  1972   36.088  13079460  739.981106       AFG

   iso_num
0        4
1        4
2        4
3        4
4        4
```

1.1 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

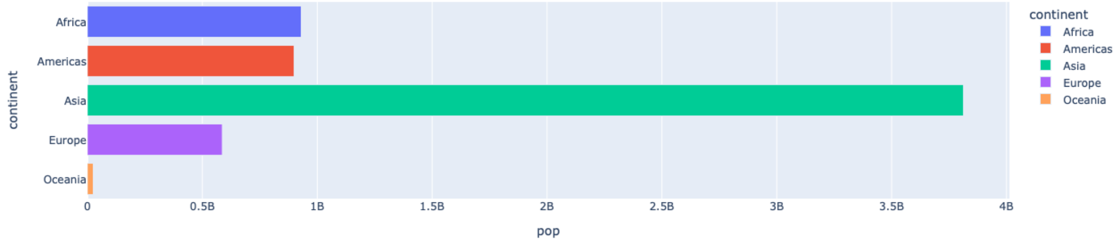
- Sort the order of the continent for the visualisation. Use axis layout setting
- Add text to each bar that represents the population

[3]:

```
# YOUR CODE HERE

pop_2007 = df[df['year'] == 2007]
pop_2007_continent = pop_2007.groupby('continent').sum() #.reindex()
# pop_2007_continent = pop_2007_continent.sort_values('pop', ascending=False)
pop_2007_continent = pop_2007_continent.reset_index()
pop_2007_continent.head()

px.bar(
    data_frame=pop_2007_continent,
    x='pop',
    y='continent',
    color='continent',
    orientation='h',
)
```



[4]:

```
pop_2007 = df[df['year'] == 2007]
pop_2007_continent = pop_2007.groupby('continent').sum().reindex()
# pop_2007_continent = pop_2007_continent.sort_values('pop', ascending=False)
pop_2007_continent = pop_2007_continent.reset_index()
# pop_2007_continent.head()
```

1.2 Question 2:

Sort the order of the continent for the visualisation

Hint: Use axis layout setting

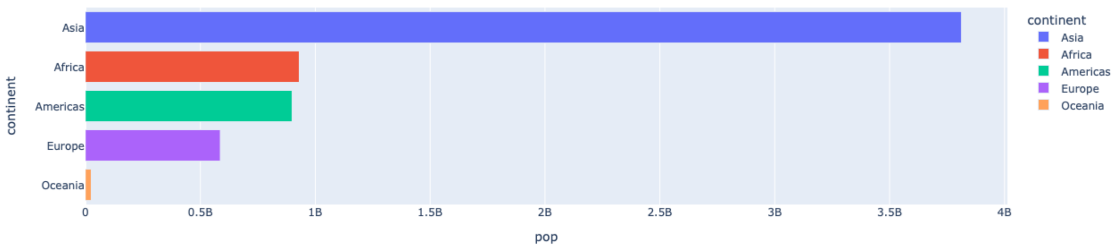
[5]:

```
# YOUR CODE HERE

pop_2007 = df[df['year'] == 2007]
pop_2007_continent = pop_2007.groupby('continent').sum().reindex()
pop_2007_continent = pop_2007_continent.sort_values('pop', ascending=False)
```

```
pop_2007_continent = pop_2007_continent.reset_index()
# pop_2007_continent.head()

px.bar(
    data_frame=pop_2007_continent,
    x='pop',
    y='continent',
    color='continent',
    orientation='h',
)
```



1.3 Question 3:

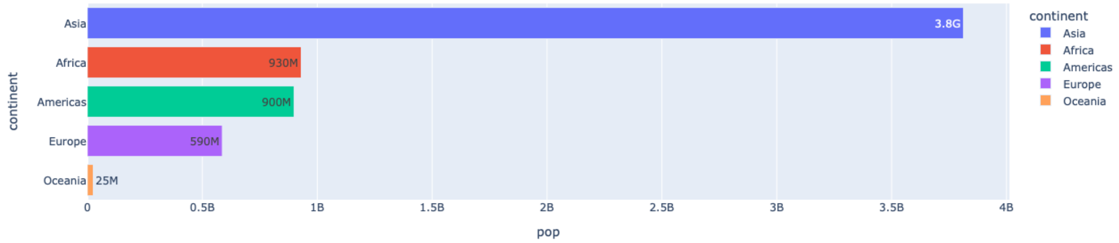
Add text to each bar that represents the population

[6]:

```
# YOUR CODE HERE

pop_2007 = df[df['year'] == 2007]
pop_2007_continent = pop_2007.groupby('continent').sum().reindex()
pop_2007_continent = pop_2007_continent.sort_values('pop', ascending=False)
pop_2007_continent = pop_2007_continent.reset_index()
# pop_2007_continent.head()

px.bar(
    data_frame=pop_2007_continent,
    x='pop',
    y='continent',
    color='continent',
    orientation='h',
    # text_auto=True,
    text_auto='.2s',
    # text='pop',
)
```



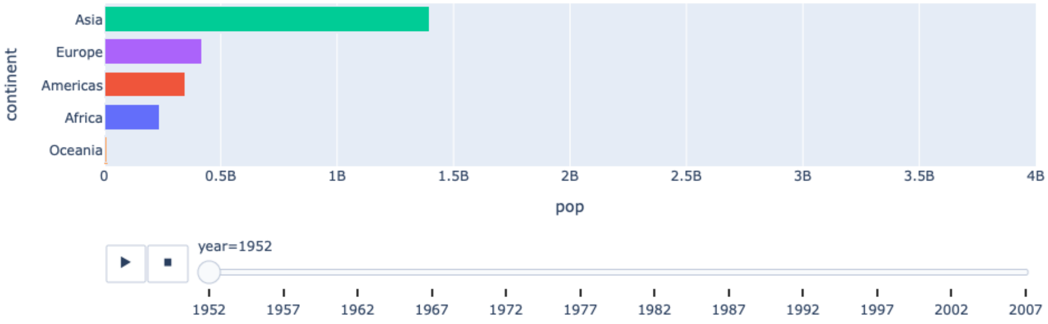
1.4 Question 4:

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

[13]:

```
pop = df.groupby(['year', 'continent']).sum().reindex()
# pop = pop.sort_values('pop', ascending=False)
pop = pop.reset_index()
pop.head()

fig = px.bar(
    data_frame=pop,
    x='pop',
    y='continent',
    color='continent',
    animation_frame="year",
    animation_group="continent",
    orientation='h',
    range_x=[0,4_000_000_000]
)
fig = fig.update_layout(barmode='stack',yaxis={'categoryorder':'total␣
  ↪ascending'},showlegend=False)
fig.show()
```

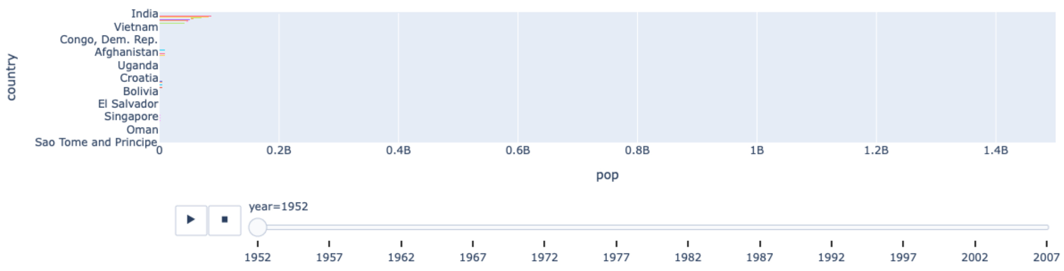


1.5 Question 5:

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

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

fig = px.bar(
    data_frame=pop,
    x='pop',
    y='country',
    color='country',
    animation_frame='year',
    animation_group='country',
    orientation='h',
    range_x=[0,1_500_000_000]
)
fig = fig.update_layout(barmode='stack',yaxis={'categoryorder':'total␣
  ↪ascending'},showlegend=False)
fig.show()
```



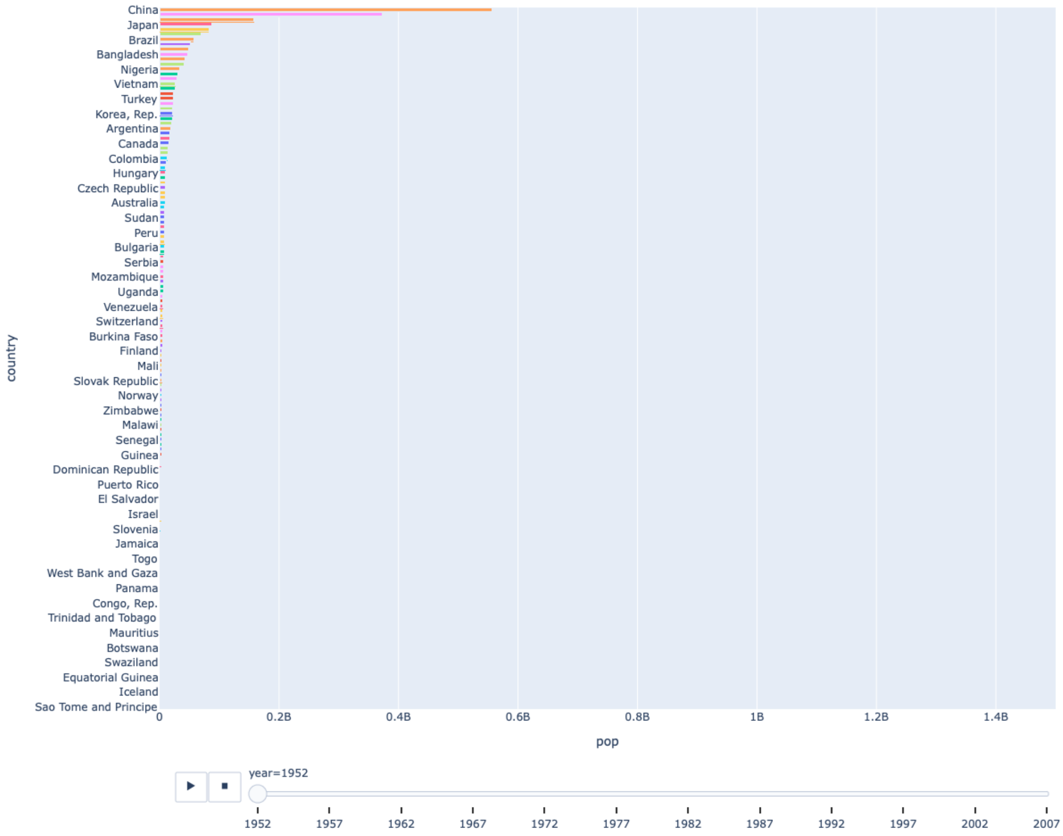
1.6 Question 6:

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

[12]:
```
# YOUR CODE HERE

# pop = df.sort_values('year', ascending=True)
pop = df.groupby(['year','country'])['pop'].sum().reset_index()

fig = px.bar(
    data_frame=pop,
    x='pop',
    y='country',
    color='country',
    animation_frame='year',
    # sort_values='year',
    animation_group='country',
    orientation='h',
    range_x=[0,1_500_000_000],
    height=1000,
)
fig = fig.update_layout(barmode='stack',yaxis={'categoryorder':'total ascending'},showlegend=False)
fig.show()
```



1.7 Question 7:

Show only the top 10 countries in the animation

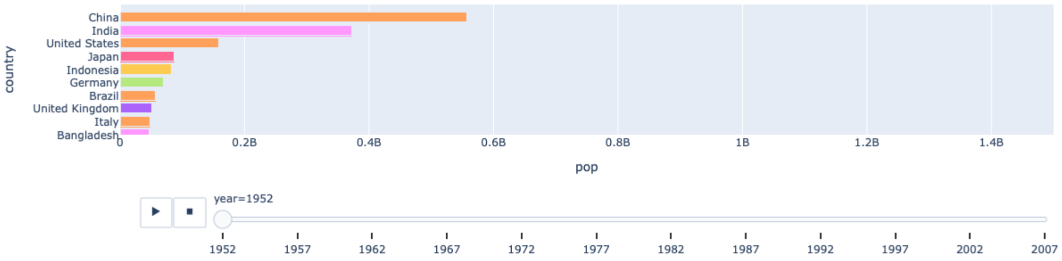
Hint: Use the axis limit to set this.

[10]:

```
# YOUR CODE HERE

pop = df.groupby(['year','country'])['pop'].sum().reset_index()

fig = px.bar(
    data_frame=pop,
    x='pop',
    y='country',
    color='country',
    animation_frame='year',
    animation_group='country',
    orientation='h',
    range_x=[0,1_500_000_000],
    range_y=[132,142],
    height=300,
)
fig = fig.update_layout(barmode='stack',yaxis={'categoryorder':'total ascending'},showlegend=False)
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