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	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

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]:
```
df2007 = df.query('year==2007')
df2007new = df2007.groupby('continent').sum()
fig = px.bar(df2007new,
             x = 'pop',
             y = df2007new.index,
             orientation = 'h',
             color = df2007new.index,)

fig.show()
```

1.2 Question 2:

Sort the order of the continent for the visualisation

Hint: Use axis layout setting

[4]:
```
df2007 = df.query('year==2007')
df2007new = df2007.groupby('continent').sum()
fig = px.bar(df2007new,
             x = 'pop',
             y = df2007new.index,
             orientation = 'h',
             color = df2007new.index)
fig.update_yaxes(categoryorder = 'total ascending')
fig.show()
```

1.3 Question 3:

Add text to each bar that represents the population

[64]:
```
df2007 = df.query('year==2007')
df2007new = df2007.groupby('continent').sum()
df2007new['pop'].round(decimals = -7)
fig = px.bar(df2007new,
             x = 'pop',
             y = df2007new.index,
             orientation = 'h',
             color = df2007new.index,
             text_auto = True)
fig.update_yaxes(categoryorder = 'total ascending')
fig.update_traces(textposition = 'outside')

fig.show()
```

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

[35]:
```
df = px.data.gapminder()
fig = px.histogram(df, y="continent", x="pop",
             orientation = 'h',
             animation_frame="year", animation_group="country",
             color="continent", hover_name="country",
             range_x=[0,4000000000])
fig.update_layout(barmode='stack', yaxis={'categoryorder':'sum ascending'})
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

[46]:
```
df = px.data.gapminder()
fig = px.bar(df, y="country", x="pop", orientation =␣
 ↪'h',animation_frame="year", animation_group="country", color="country",␣
 ↪hover_name="country",
        range_x=[0,1300000000])

fig["layout"].pop("updatemenus") # optional, drop animation buttons
fig.update_layout(barmode='stack', yaxis={'categoryorder':'total ascending'})
```

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

[47]:
```
df = px.data.gapminder()
fig = px.bar(df, y="country", x="pop", orientation =␣
 ↪'h',animation_frame="year", animation_group="country", color="country",␣
 ↪hover_name="country",
        range_x=[0,1300000000], height = 1000)

fig["layout"].pop("updatemenus") # optional, drop animation buttons
fig.update_layout(barmode='stack', yaxis={'categoryorder':'total ascending'})
```

1.7 Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

[56]:
```
df = px.data.gapminder()
fig = px.bar(df, y="country", x="pop", orientation =␣
 ↪'h',animation_frame="year", animation_group="country", color="country",␣
 ↪hover_name="country",
       range_x=[0,1300000000], range_y=[132,141])

fig["layout"].pop("updatemenus") # optional, drop animation buttons
fig.update_layout(barmode='stack', yaxis={'categoryorder':'total ascending'})
```

[ ]:
```

```