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

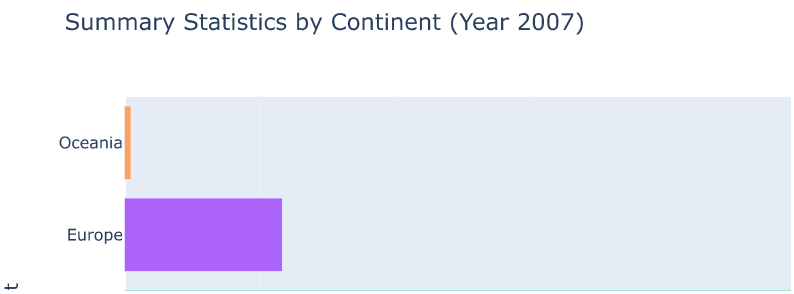
Question 1:

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

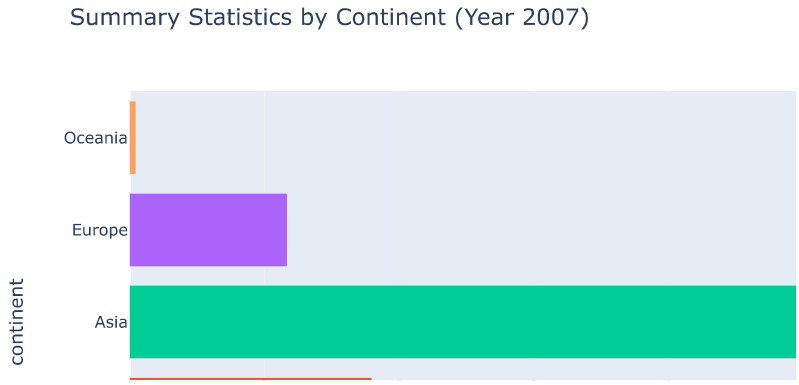
Hints:

- Extract the 2007 year data from the dataframe. You have to process the data accordingly
- use plotly bar (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 [3]:
```
# Filter the DataFrame to select data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns
df_sum = df_2007.groupby('continent').sum()
df_sum['pop']=df_sum['pop']/1e9
# Create a bar chart using Plotly Express
fig = px.bar(df_sum, y=df_sum.index, x=df_sum['pop'],
             labels={'pop': 'Total Population'},
             title='Summary Statistics by Continent (Year 2007)'
            )
colors = px.colors.qualitative.Plotly[:len(df_sum)]
fig.update_traces(marker_color=colors)
fig.show()
```



In [3]:
```
# Filter the DataFrame to select data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns
df_sum = df_2007.groupby('continent').sum()
df_sum['pop']=df_sum['pop']/1e9
# Create a bar chart using Plotly Express
fig = px.bar(df_sum, y=df_sum.index, x=df_sum['pop'],
             labels={'pop': 'Total Population'},
             title='Summary Statistics by Continent (Year 2007)'
            )
colors = px.colors.qualitative.Plotly[:len(df_sum)]
fig.update_traces(marker_color=colors)
fig.show()
```



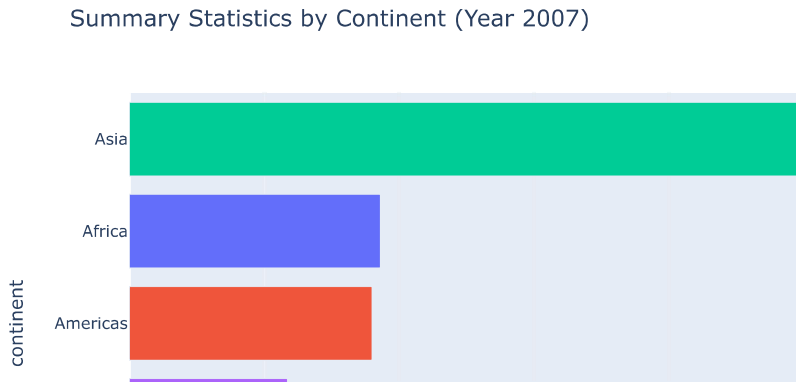
Question 2:

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 [4]:
```
# Filter the DataFrame to select data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns
df_sum = df_2007.groupby('continent').sum()
df_sum['pop']=df_sum['pop']/1e9
# Create a bar chart using Plotly Express
fig = px.bar(df_sum, y=df_sum.index, x=df_sum['pop'],
             labels={'pop': 'Total Population'},
             title='Summary Statistics by Continent (Year 2007)'
            )

# Customize the layout of the chart: hide the legend
fig.update_layout(showlegend=False)

# Update the layout for the y-axis to order categories by total population in ascending
fig.update_yaxes(categoryorder='total ascending')

# Customize the text labels on the bars: format with two decimal places and position th
colors = px.colors.qualitative.Plotly[:len(df_sum)]
fig.update_traces(marker_color=colors)

# Display the resulting chart
fig.show()
```

In [4]:
```
# Filter the DataFrame to select data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns
df_sum = df_2007.groupby('continent').sum()
df_sum['pop']=df_sum['pop']/1e9
# Create a bar chart using Plotly Express
fig = px.bar(df_sum, y=df_sum.index, x=df_sum['pop'],
             labels={'pop': 'Total Population'},
             title='Summary Statistics by Continent (Year 2007)'
            )

# Customize the layout of the chart: hide the legend
fig.update_layout(showlegend=False)

# Update the layout for the y-axis to order categories by total population in ascending
fig.update_yaxes(categoryorder='total ascending')

# Customize the text labels on the bars: format with two decimal places and position th
colors = px.colors.qualitative.Plotly[:len(df_sum)]
fig.update_traces(marker_color=colors)

# Display the resulting chart
fig.show()
```



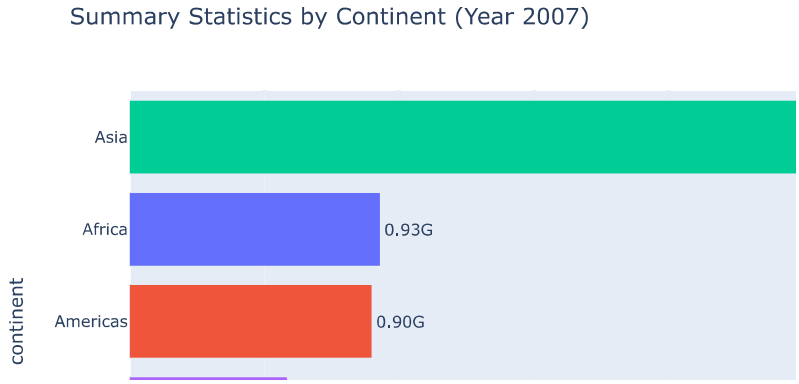
Question 3:

Add text to each bar that represents the population

In [5]:
```
# Filter the DataFrame to select data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns
df_sum = df_2007.groupby('continent').sum()
df_sum['pop']=df_sum['pop']/1e9
# Create a bar chart using Plotly Express
fig = px.bar(df_sum, y=df_sum.index, x=df_sum['pop'],
             labels={'pop': 'Total Population'},
             title='Summary Statistics by Continent (Year 2007)'
            )

# Customize the layout of the chart: hide the legend
fig.update_layout(showlegend=False)

# Update the layout for the y-axis to order categories by total population in ascending
fig.update_yaxes(categoryorder='total ascending')

# Customize the text labels on the bars: format with two decimal places and position th
colors = px.colors.qualitative.Plotly[:len(df_sum)]
fig.update_traces(text=df_sum['pop'],
                  textposition='outside',
                  texttemplate='%{text:.2f}G',
                  marker_color=colors)

# Display the resulting chart
fig.show()
```

Add text to each bar that represents the population

In [5]:
```
# Filter the DataFrame to select data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group the filtered data by continent and calculate the sum of numeric columns
df_sum = df_2007.groupby('continent').sum()
df_sum['pop']=df_sum['pop']/1e9
# Create a bar chart using Plotly Express
fig = px.bar(df_sum, y=df_sum.index, x=df_sum['pop'],
             labels={'pop': 'Total Population'},
             title='Summary Statistics by Continent (Year 2007)'
            )

# Customize the layout of the chart: hide the legend
fig.update_layout(showlegend=False)

# Update the layout for the y-axis to order categories by total population in ascending
fig.update_yaxes(categoryorder='total ascending')

# Customize the text labels on the bars: format with two decimal places and position th
colors = px.colors.qualitative.Plotly[:len(df_sum)]
fig.update_traces(text=df_sum['pop'],
                  textposition='outside',
                  texttemplate='%{text:.2f}G',
                  marker_color=colors)

# Display the resulting chart
fig.show()
```



Question 4:

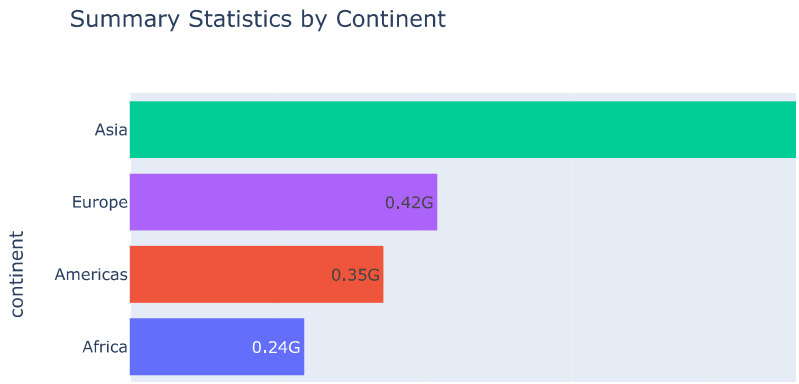
Thus far we looked at data from one year (2007). Lets create an animation to see the population growth of the continents through the years.

In [6]:
```
# Filter the DataFrame to select data for the years 1952 to 2007
df_filtered = df[(df['year'] >= 1952) & (df['year'] <= 2007)]
# Calculate the sum of numeric columns grouped by continent and year
df_sums = df_filtered.groupby(['continent', 'year']).sum().reset_index()
df_sums['pop'] = df_sums['pop'] / 1e9

# Create a bar chart using Plotly Express
fig = px.bar(
    df_sums,
    x=df_sums['pop'],
    y='continent',
    labels={'pop': 'Total Population'},
    title='Summary Statistics by Continent',
    animation_group="continent",
    animation_frame="year",
    color="continent",
    text='pop'
)

# Update layout
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(
    texttemplate='%{text:.2f}G',
)

# Display the resulting chart
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]:
```
# Filter the DataFrame to select data for the years 1952 to 2007
df_filtered = df[(df['year'] >= 1952) & (df['year'] <= 2007)]

# Calculate the sum of numeric columns grouped by continent and year
df_sums = df_filtered.groupby(['continent', 'year']).sum().reset_index()
df_sums['pop'] = df_sums['pop'] / 1e9

# Create a bar chart using Plotly Express
fig = px.bar(
    df_sums,
    x=df_sums['pop'],
    y='continent',
    labels={'pop': 'Total Population'},
    title='Summary Statistics by Continent',
    animation_group="continent",
    animation_frame="year",
    color="continent",
    text='pop'
)

# Update layout
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(
    texttemplate='%{text:.2f}G',
)

# Display the resulting chart
fig.show()
```



Question 5:

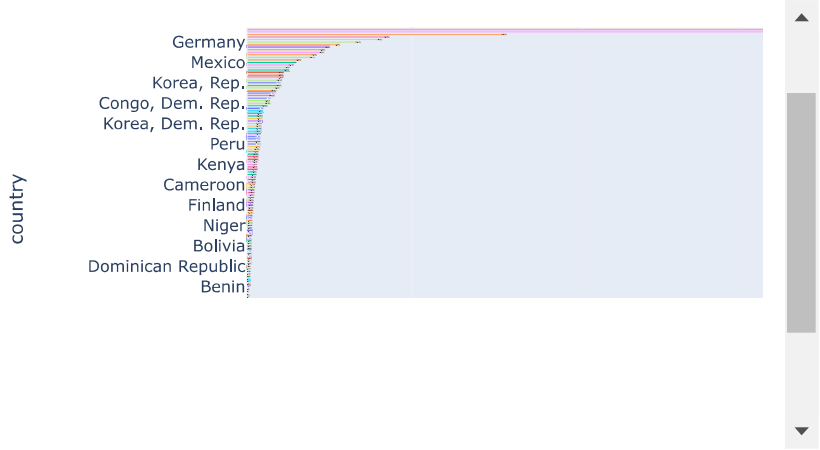
Instead of the continents, lets look at individual countries. Create an animation that shows the population growth of the countries through the years

In [7]:
```
# Filter the DataFrame to select data for the years 1952 to 2007
df_filtered = df[(df['year'] >= 1952) & (df['year'] <= 2007)]

# Calculate the sum of numeric columns grouped by continent and year
df_sums = df_filtered.groupby(['country', 'year']).sum().reset_index()
df_sums['pop'] = df_sums['pop'] / 1e9

# Create a bar chart using Plotly Express
fig = px.bar(
    df_sums,
    x=df_sums['pop'],
    y='country',
    labels={'pop': 'Total Population'},
    title='Summary Statistics by Continent',
    animation_group="country",
    animation_frame="year",
    color="country",
    text='pop'
)

# Update layout
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(
    texttemplate='%{text:.2f}G',
)

# Display the resulting chart
fig.show()
```

In [7]:
```
# Filter the DataFrame to select data for the years 1952 to 2007
df_filtered = df[(df['year'] >= 1952) & (df['year'] <= 2007)]

# Calculate the sum of numeric columns grouped by continent and year
df_sums = df_filtered.groupby(['country', 'year']).sum().reset_index()
df_sums['pop'] = df_sums['pop'] / 1e9

# Create a bar chart using Plotly Express
fig = px.bar(
    df_sums,
    x=df_sums['pop'],
    y='country',
    labels={'pop': 'Total Population'},
    title='Summary Statistics by Continent',
    animation_group="country",
    animation_frame="year",
    color="country",
    text='pop'
)

# Update layout
fig.update_layout(showlegend=False)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(
    texttemplate='%{text:.2f}G',
)

# Display the resulting chart
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

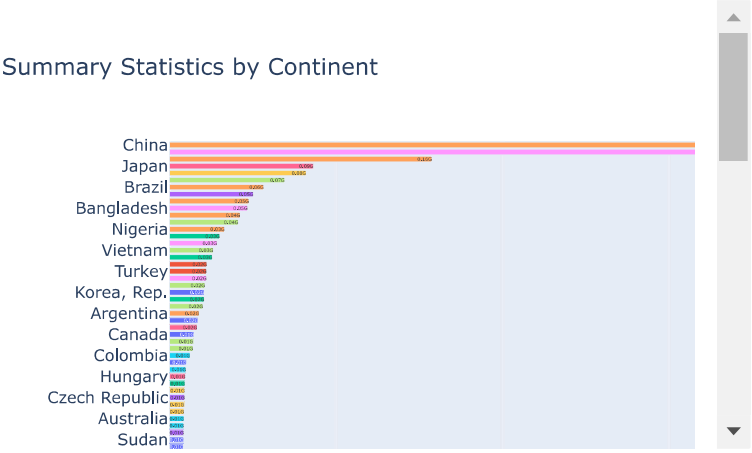


Question 6:

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

In [8]:
```
# Filter the DataFrame to select data for the years 1952 to 2007
df_filtered = df[(df['year'] >= 1952) & (df['year'] <= 2007)]

# Calculate the sum of numeric columns grouped by continent and year
df_sums = df_filtered.groupby(['country', 'year']).sum().reset_index()
df_sums['pop'] = df_sums['pop'] / 1e9

# Create a bar chart using Plotly Express
fig = px.bar(
    df_sums,
    x=df_sums['pop'],
    y='country',
    labels={'pop': 'Total Population'},
    title='Summary Statistics by Continent',
    animation_group="country",
    animation_frame="year",
    color="country",
    text='pop'
)

# Update layout
fig.update_layout(showlegend=False,height=1000)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(
    texttemplate='%{text:.2f}G',
)

# Display the resulting chart
fig.show()
```

In [8]:
```
# Filter the DataFrame to select data for the years 1952 to 2007
df_filtered = df[(df['year'] >= 1952) & (df['year'] <= 2007)]

# Calculate the sum of numeric columns grouped by continent and year
df_sums = df_filtered.groupby(['country', 'year']).sum().reset_index()
df_sums['pop'] = df_sums['pop'] / 1e9

# Create a bar chart using Plotly Express
fig = px.bar(
    df_sums,
    x=df_sums['pop'],
    y='country',
    labels={'pop': 'Total Population'},
    title='Summary Statistics by Continent',
    animation_group="country",
    animation_frame="year",
    color="country",
    text='pop'
)

# Update layout
fig.update_layout(showlegend=False,height=1000)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(
    texttemplate='%{text:.2f}G',
)

# Display the resulting chart
fig.show()
```



Question 7:

Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

In [9]:
```
#load data
df = px.data.gapminder()
df.head()
# Filter the DataFrame to select data for the years 1952 to 2007
df_filtered = df[(df['year'] >= 1952) & (df['year'] <= 2007)]

# Calculate the sum of numeric columns grouped by continent and year
df_sums = df_filtered.groupby(['country', 'year']).sum().reset_index()
df_sums['pop'] = df_sums['pop'] / 1e9
df_top_10 = df_sums.groupby('year').apply(lambda x: x.nlargest(10, 'pop')).reset_index
df_sums= df_top_10
# Create a bar chart using Plotly Express
fig = px.bar(
    df_sums,
    x=df_sums['pop'],
    y='country',
    labels={'pop': 'Total Population'},
    title='Summary Statistics by Continent',
    animation_group="country",
    animation_frame="year",
    color="country",
    text='pop'
)

# Update layout
fig.update_layout(showlegend=False,
                  height=500)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(
    texttemplate='%{text:.2f}G',
```

In [9]:
```
#load data
df = px.data.gapminder()
df.head()
# Filter the DataFrame to select data for the years 1952 to 2007
df_filtered = df[(df['year'] >= 1952) & (df['year'] <= 2007)]

# Calculate the sum of numeric columns grouped by continent and year
df_sums = df_filtered.groupby(['country', 'year']).sum().reset_index()
df_sums['pop'] = df_sums['pop'] / 1e9
df_top_10 = df_sums.groupby('year').apply(lambda x: x.nlargest(10, 'pop')).reset_index
df_sums= df_top_10
# Create a bar chart using Plotly Express
fig = px.bar(
    df_sums,
    x=df_sums['pop'],
    y='country',
    labels={'pop': 'Total Population'},
    title='Summary Statistics by Continent',
    animation_group="country",
    animation_frame="year",
    color="country",
    text='pop'
)

# Update layout
fig.update_layout(showlegend=False,
                  height=500)
fig.update_yaxes(categoryorder='total ascending')
fig.update_traces(
    texttemplate='%{text:.2f}G',
)

# Display the resulting chart
fig.show()
```