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 [3]:
```
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 [4]:
```
#Load data
df = px.data.gapminder()
df.head()
```

Out[4]:

	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:

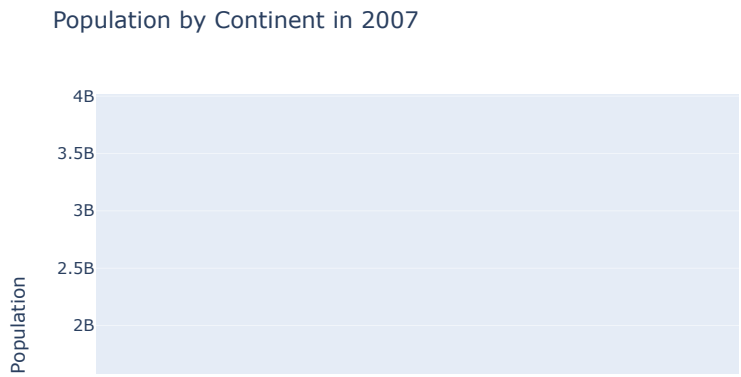
Recreate the barplot below that shows the population of different continents for the year 2007.

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 [2]:
```
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group data by continent and calculate total population
continent_pop = df_2007.groupby('continent')['pop'].sum().reset_index()

# Sort the data by population in descending order
continent_pop = continent_pop.sort_values(by='pop', ascending=False)

# Create the barplot
fig = px.bar(
    continent_pop,
    x='continent',
    y='pop',
    color='continent',  # Use different colors for each continent
    labels={'continent': 'Continent', 'pop': 'Population'},
    title='Population by Continent in 2007'
)

# Add text labels to each bar
fig.update_traces(texttemplate='%{text}', textposition='outside', text=con

# Customize the layout for better visualization
fig.update_layout(xaxis={'categoryorder': 'total ascending'})
```

In [2]:
```
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group data by continent and calculate total population
continent_pop = df_2007.groupby('continent')['pop'].sum().reset_index()

# Sort the data by population in descending order
continent_pop = continent_pop.sort_values(by='pop', ascending=False)

# Create the barplot
fig = px.bar(
    continent_pop,
    x='continent',
    y='pop',
    color='continent',  # Use different colors for each continent
    labels={'continent': 'Continent', 'pop': 'Population'},
    title='Population by Continent in 2007'
)

# Add text labels to each bar
fig.update_traces(texttemplate='%{text}', textposition='outside', text=con

# Customize the layout for better visualization
fig.update_layout(xaxis={'categoryorder': 'total ascending'})

# Show the plot
fig.show()
```



In [10]:
```
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group data by continent and calculate total population
continent_pop = df_2007.groupby('continent')['pop'].sum().reset_index()

# Sort the data by population in ascending order
continent_pop = continent_pop.sort_values(by='pop', ascending=True)  # Sor

# Create the barplot
fig = px.bar(
    continent_pop,
    x='continent',
    y='pop',
    color='continent',  # Use different colors for each continent
    labels={'continent': 'Continent', 'pop': 'Population'},
    title='Population by Continent in 2007'
)

# Remove text labels from each bar
fig.update_traces(text=[])

# Customize the layout to sort the bars by population in ascending order
fig.update_layout(xaxis={'categoryorder': 'total ascending'})
```

Question 2:

Sort the order of the continent for the visualisation

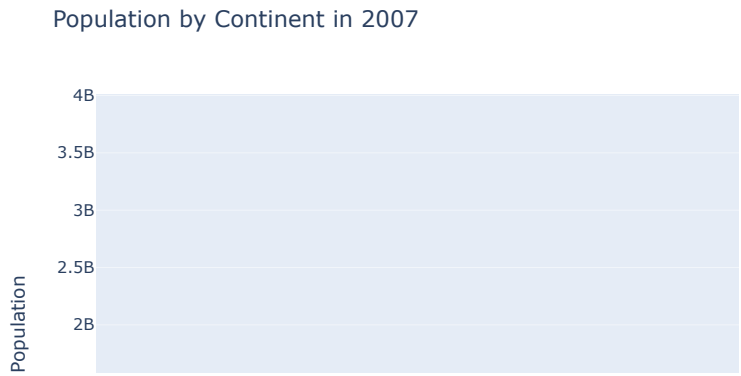
Hint: Use axis layout setting (https://plotly.com/python/reference/layout/xaxis/)

In [10]:
```
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group data by continent and calculate total population
continent_pop = df_2007.groupby('continent')['pop'].sum().reset_index()

# Sort the data by population in ascending order
continent_pop = continent_pop.sort_values(by='pop', ascending=True)  # Sor

# Create the barplot
fig = px.bar(
    continent_pop,
    x='continent',
    y='pop',
    color='continent',  # Use different colors for each continent
    labels={'continent': 'Continent', 'pop': 'Population'},
    title='Population by Continent in 2007'
)

# Remove text labels from each bar
fig.update_traces(text=[])

# Customize the layout to sort the bars by population in ascending order
fig.update_layout(xaxis={'categoryorder': 'total ascending'})

# Show the plot
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/)



In [4]:
```
# YOUR CODE HERE
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]

# Group data by continent and calculate total population
continent_pop = df_2007.groupby('continent')['pop'].sum().reset_index()

# Sort the data by population in ascending order
continent_pop = continent_pop.sort_values(by='pop', ascending=True)  # Sor

# Create the barplot
fig = px.bar(
    continent_pop,
    x='continent',
    y='pop',
    color='continent',  # Use different colors for each continent
    labels={'continent': 'Continent', 'pop': 'Population'},
    title='Population by Continent in 2007'
)

# Add text labels to each bar
fig.update_traces(text=continent_pop['pop'], textposition='outside')

# Customize the layout to sort the bars by population in ascending order
fig.update_layout(xaxis={'categoryorder': 'total ascending'})
```

Question 3:

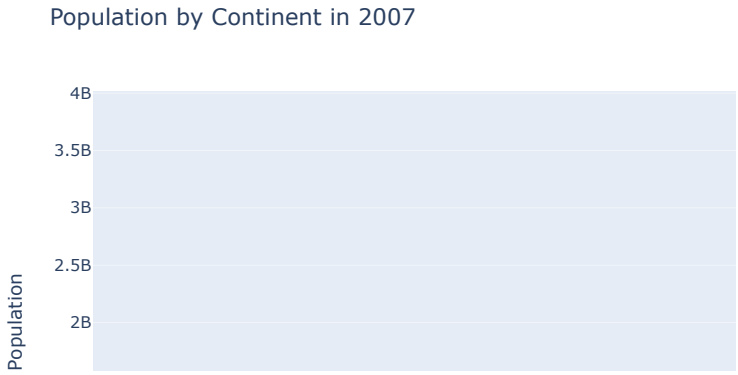
Add text to each bar that represents the population

In [4]:

```
# YOUR CODE HERE
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Filter data for the year 2007
df_2007 = df[df['year'] == 2007]
# Group data by continent and calculate total population
continent_pop = df_2007.groupby('continent')['pop'].sum().reset_index()

# Sort the data by population in ascending order
continent_pop = continent_pop.sort_values(by='pop', ascending=True)  # Sor

# Create the barplot
fig = px.bar(
    continent_pop,
    x='continent',
    y='pop',
    color='continent',  # Use different colors for each continent
    labels={'continent': 'Continent', 'pop': 'Population'},
    title='Population by Continent in 2007'
)

# Add text labels to each bar
fig.update_traces(text=continent_pop['pop'], textposition='outside')

# Customize the layout to sort the bars by population in ascending order
fig.update_layout(xaxis={'categoryorder': 'total ascending'})

# Show the plot
fig.show()
```

Question 3:

Add text to each bar that represents the population



In [12]:

```
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Create a barplot with animation for population by continent over the yea
fig = px.bar(
    x='continent',
    y='pop',
    color='continent',
    labels={'continent': 'Continent', 'pop': 'Population'},
    title='Population by Continent Over the Years',
    animation_frame='year',  # Specify the variable for animation
    range_y=[0, df['pop'].max()]  # Set the y-axis range
)

# Customize the layout
fig.update_layout(xaxis={'categoryorder': 'total ascending'})

# Remove vertical lines inside the bars
fig.update_traces(marker_line_width=0)

# Show the animated plot
fig.show()
```

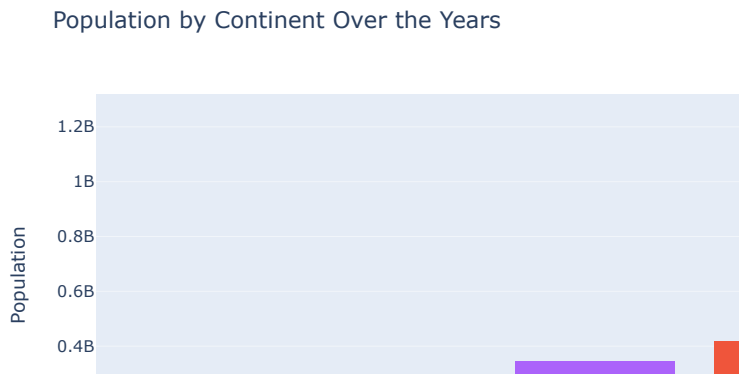
Question 4:

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

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

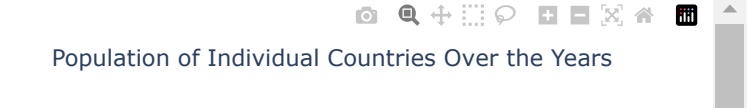
In [12]:
```
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Create a barplot with animation for population by continent over the yea
fig = px.bar(
    df,
    x='continent',
    y='pop',
    color='continent',
    labels={'continent': 'Continent', 'pop': 'Population'},
    title='Population by Continent Over the Years',
    animation_frame='year',  # Specify the variable for animation
    range_y=[0, df['pop'].max()]  # Set the y-axis range
)

# Customize the layout
fig.update_layout(xaxis={'categoryorder': 'total ascending'})

# Remove vertical lines inside the bars
fig.update_traces(marker_line_width=0)

# Show the animated plot
fig.show()
```



Question 5:

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

In [14]:
```
# YOUR CODE HERE
import plotly.express as px
# Load the Gapminder dataset
df = px.data.gapminder()

# Create a barplot with animation for population by individual countries o
fig = px.bar(
    df,
    x='country',
    y='pop',
    color='country',
    labels={'country': 'Country', 'pop': 'Population'},
    title='Population of Individual Countries Over the Years',
    animation_frame='year',  # Specify the variable for animation
    range_y=[0, df['pop'].max()]  # Set the y-axis range
)

# Customize the layout
fig.update_layout(xaxis={'categoryorder': 'total ascending'})

# Show the animated plot
fig.show()
```



Question 5:

In [14]:
```
# YOUR CODE HERE
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Create a barplot with animation for population by individual countries o
fig = px.bar(
    df,
    x='country',
    y='pop',
    color='country',
    labels={'country': 'Country', 'pop': 'Population'},
    title='Population of Individual Countries Over the Years',
    animation_frame='year',  # Specify the variable for animation
    range_y=[0, df['pop'].max()]  # Set the y-axis range
)

# Customize the layout
fig.update_layout(xaxis={'categoryorder': 'total ascending'})

# Show the animated plot
fig.show()
```

Instead of the country, let's 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 [15]:
```
# YOUR CODE HERE
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Create a barplot with animation for population by individual countries o
fig = px.bar(
    df,
    x='country',
    y='pop',
    color='country',
    labels={'country': 'Country', 'pop': 'Population'},
    title='Population of Individual Countries Over the Years',
    animation_frame='year',  # Specify the variable for animation
    range_y=[0, df['pop'].max()]  # Set the y-axis range
)

# Customize the layout
fig.update_layout(
    xaxis={'categoryorder': 'total ascending'},
    height=1000  # Set the height of the figure to 1000 pixels
)

# Show the animated plot
fig.show()
```

In [15]:
```
# YOUR CODE HERE
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Create a barplot with animation for population by individual countries o
fig = px.bar(
    df,
    x='country',
    y='pop',
    color='country',
    labels={'country': 'Country', 'pop': 'Population'},
    title='Population of Individual Countries Over the Years',
    animation_frame='year',  # Specify the variable for animation
    range_y=[0, df['pop'].max()]  # Set the y-axis range
)

# Customize the layout
fig.update_layout(
    xaxis={'categoryorder': 'total ascending'},
    height=1000  # Set the height of the figure to 1000 pixels
)

# Show the animated plot
fig.show()
```

Question 7:

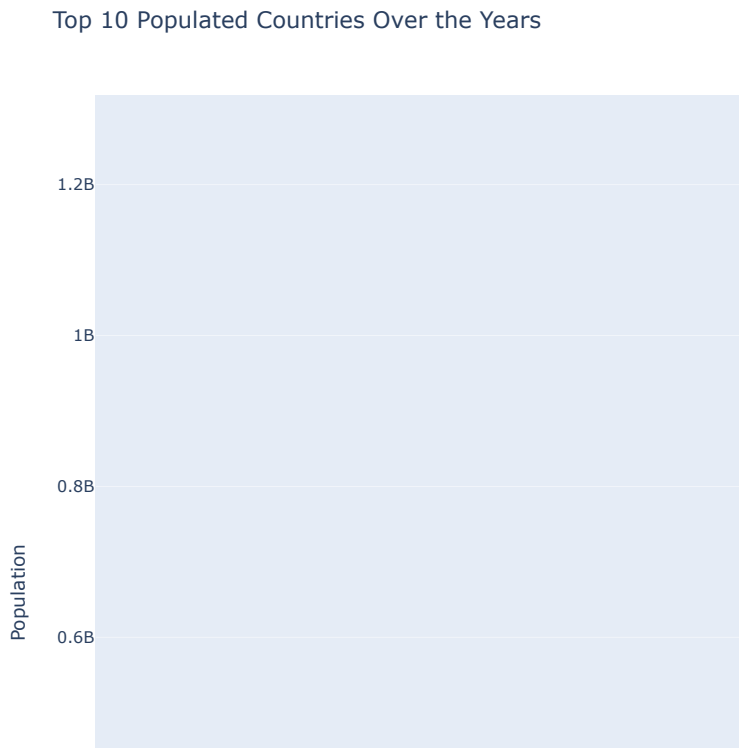
Show only the top 10 countries in the animation

Hint: Use the axis limit to set this.

In [17]:
```
# YOUR CODE HERE
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Create a barplot with animation for population by individual countries o
fig = px.bar(
    df,
    x='country',
    y='pop',
    color='country',
    labels={'country': 'Country', 'pop': 'Population'},
    title='Top 10 Populated Countries Over the Years',
    animation_frame='year',  # Specify the variable for animation
    range_y=[0, df['pop'].max()]  # Set the y-axis range
)

# Customize the layout
fig.update_layout(
    xaxis={'categoryorder': 'total ascending'},
    height=1000  # Set the height of the figure to 1000 pixels
)

# Show only the top 10 countries
fig.update_xaxes(categoryorder='total ascending', type='category', tickval

# Show the animated plot
fig.show()
```

In [17]:

```
# YOUR CODE HERE
import plotly.express as px

# Load the Gapminder dataset
df = px.data.gapminder()

# Create a barplot with animation for population by individual countries o
fig = px.bar(
    df,
    x='country',
    y='pop',
    color='country',
    labels={'country': 'Country', 'pop': 'Population'},
    title='Top 10 Populated Countries Over the Years',
    animation_frame='year',  # Specify the variable for animation
    range_y=[0, df['pop'].max()]  # Set the y-axis range
)

# Customize the layout
fig.update_layout(
    xaxis={'categoryorder': 'total ascending'},
    height=1000  # Set the height of the figure to 1000 pixels
)

# Show only the top 10 countries
fig.update_xaxes(categoryorder='total ascending', type='category', tickval

# Show the animated plot
fig.show()
```



0.6B