Lecture Notes/Tutorials by Dr. Bilgic @RIT

Visualization with Plotly

Topic: Python Plotly

Your Task:

1) Review the code, comments 2) Practice in a new notebook

Table of Contents

---

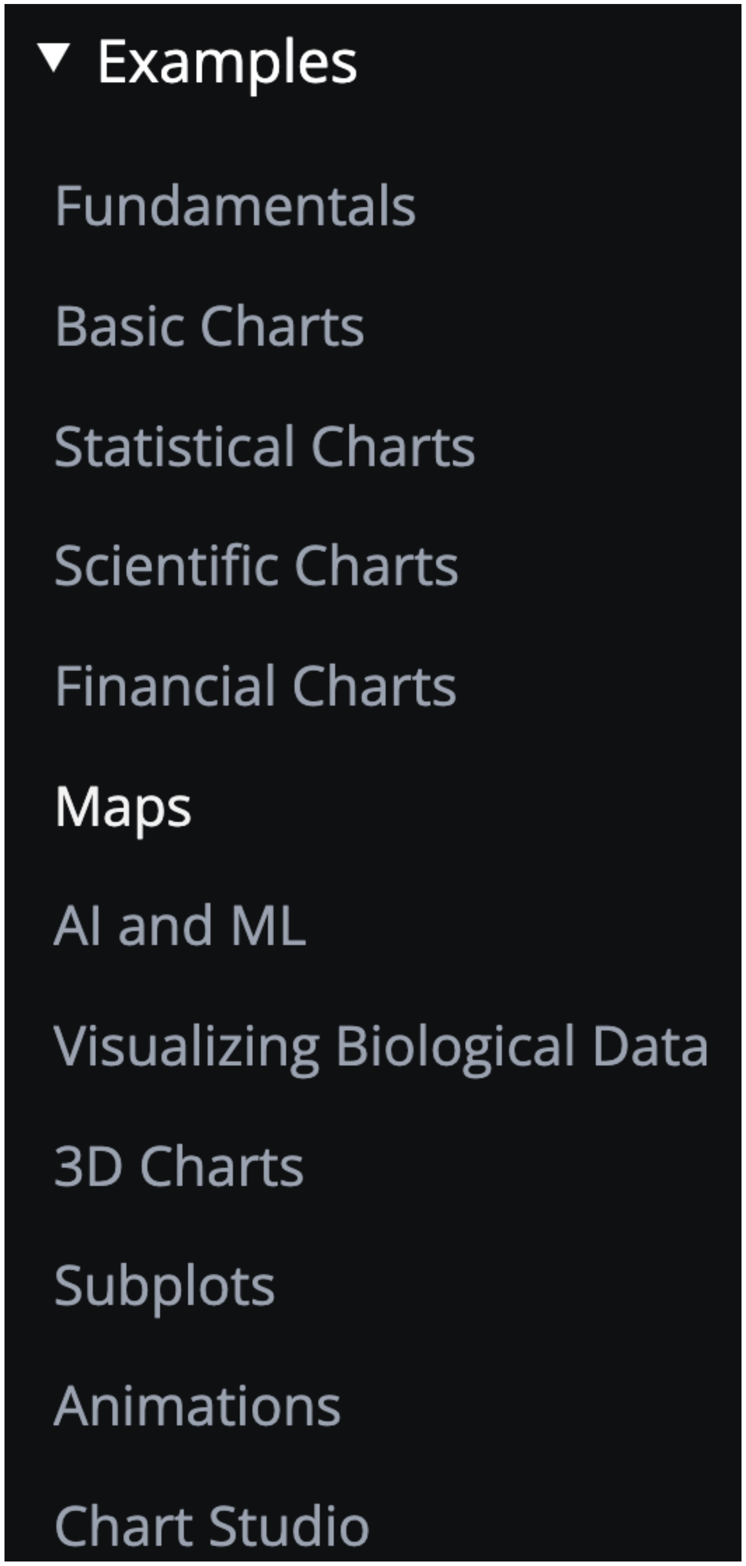
1. Plotly Pack Resources

- The plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.
- Built on top of the Plotly JavaScript library (plotly.js), plotly enables Python users to create beautiful interactive web-based visualizations that can be displayed in Jupyter notebooks, saved to standalone HTML files, or served as part of pure Python-built web applications using Dash. The plotly Python library is sometimes referred to as "plotly.py" to differentiate it from the JavaScript library.
- Start here first
- Graph objects: The figures created, manipulated and rendered by the plotly Python library are represented by tree-like data structures which are automatically serialized to JSON for rendering by the Plotly.js JavaScript library.
- Plotly Express: The plotly.express module (usually imported as px) contains functions that can create entire figures at once, and is referred to as Plotly Express or PX. Plotly Express is a built-in part of the plotly library, and is the recommended starting point for creating most common figures.

Plotly Express currently includes the following functions:

- **Basics**: scatter, line, area, bar, funnel, timeline
- **Part-of-Whole**: pie, sunburst, treemap, icicle, funnel_area
- **1D Distributions**: histogram, box, violin, strip, ecdf
- **2D Distributions**: density_heatmap, density_contour
- **Matrix or Image Input**: imshow
- **3-Dimensional**: scatter_3d, line_3d
- **Multidimensional**: scatter_matrix, parallel_coordinates, parallel_categories
- **Tile Maps**: scatter_mapbox, line_mapbox, choropleth_mapbox, density_mapbox
- **Outline Maps**: scatter_geo, line_geo, choropleth
- **Polar Charts**: scatter_polar, line_polar, bar_polar
- **Ternary Charts**: scatter_ternary, line_ternary

- Review the px's functions and features.
- Plotly Express more than 30 functions: Plotly Express provides more than 30 functions for creating different types of figures.
- Galery: Just a sampling of what can be done with Plotly Express


▼ Examples
Fundamentals
Basic Charts
Statistical Charts
Scientific Charts
Financial Charts
Maps
AI and ML
Visualizing Biological Data
3D Charts
Subplots
Animations
Chart Studio

- Let's install and practice plotly only in the coming s [redacted]

---

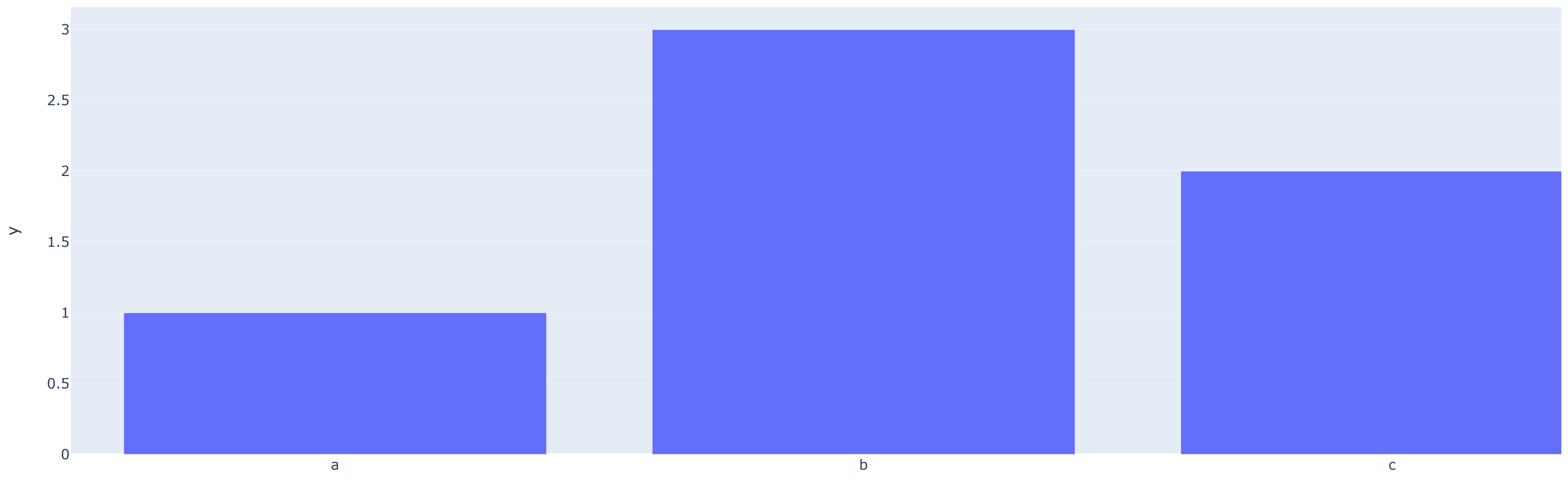
2. Install Plotly

- Run the following code on terminal/cmd:

  > conda install -c plotly plotly=5.11 #or any recent Python
  >
  > conda install -c plotly plotly=5.13.0

- Read JupyterLab Support with the link getting-started. Scroll down and make sure you install all using Terminal under your environm
- Use pip install plotly in case issues pop up. generally, then updating the conda and other packs, the issue is solved. Avoid use of pip pack if you can solve it with conda pack
- This package contains everything you need to write figures to standalone HTML files

In [1]:
```
# just a testing code with a trivial data
import plotly.express as px

fig = px.bar(x=["a", "b", "c"], y=[1, 3, 2])

fig.show()
```



---

3. Tutorial Instruction

During the tutorial after you reviewed the resources above, installed the packs and run the trivial code:

- First, watch Data Visualization with Plotly Express and Dash
- Then, start practicing some chunks from https://plotly.com/python/plotly-express/ (skip the dash codes chunk, we don't cover `dash` yet in this tutorial)

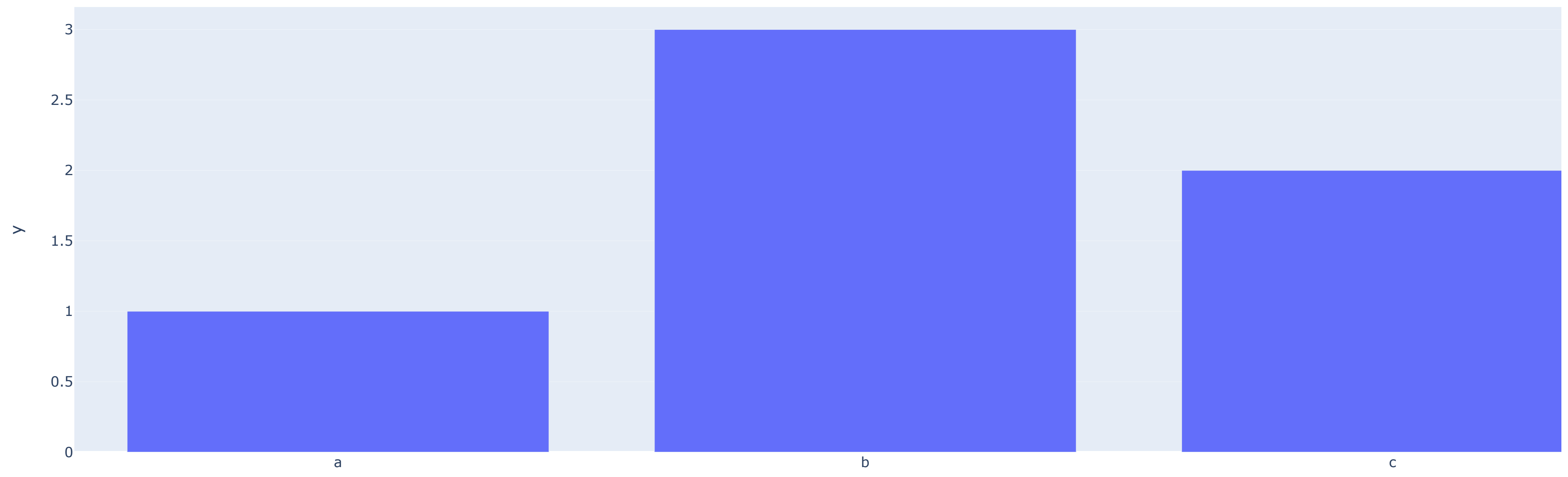
---

4. Plotly Practice

In [2]:
```
# import packs: no need to do again actually
import pandas as pd
import plotly.express as px
```

In [3]:
```
# a simple example: writing to html
#import plotly.express as px

fig = px.bar(x=["a", "b", "c"], y=[1, 3, 2])
#fig.show()
fig.write_html('first_figure.html', auto_open=True)
```

In [4]:
```
# a simple example: just showing here
import plotly.express as px

fig = px.bar(x=["a", "b", "c"], y=[1, 3, 2])
fig.show()
```

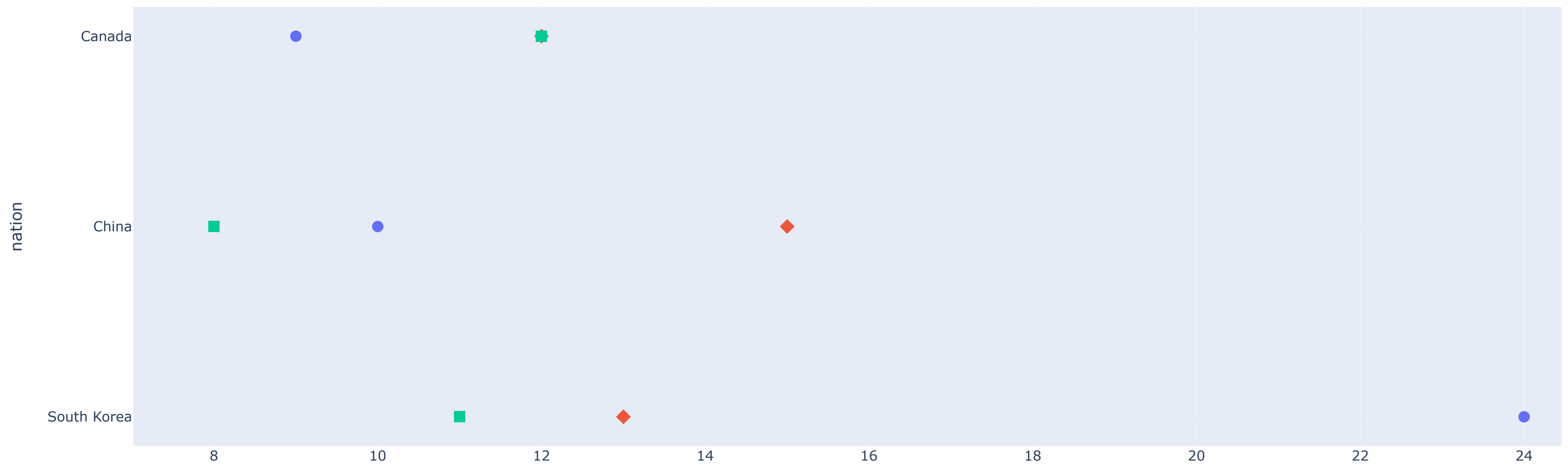


In this chart, the insight is ..

In [5]:
```
# or using FigureWidget objects: this is more flexible to use in web
import plotly.express as px

fig = px.bar(x=["a", "b", "c"], y=[1, 3, 2])

import plotly.graph_objects as go

fig_widget = go.FigureWidget(fig)

fig_widget
```

```
FigureWidget({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'x=%{x}<br>y=%{y}<extra>…
```

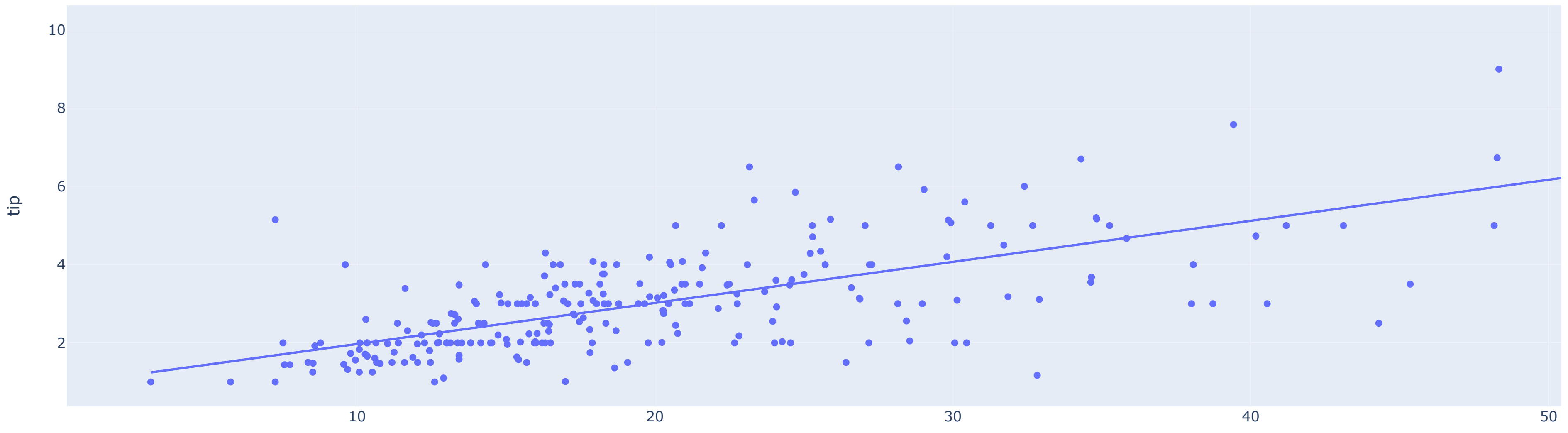
In [7]:
```
# basic dotplot

df = px.data.medals_long()

fig = px.scatter(df, y="nation", x="count",
                 color="medal",
                 symbol="medal")

fig.update_traces(marker_size=10)

fig.show()

# try this and open:
# fig.write_html('first_figure.html', auto_open=True)
```



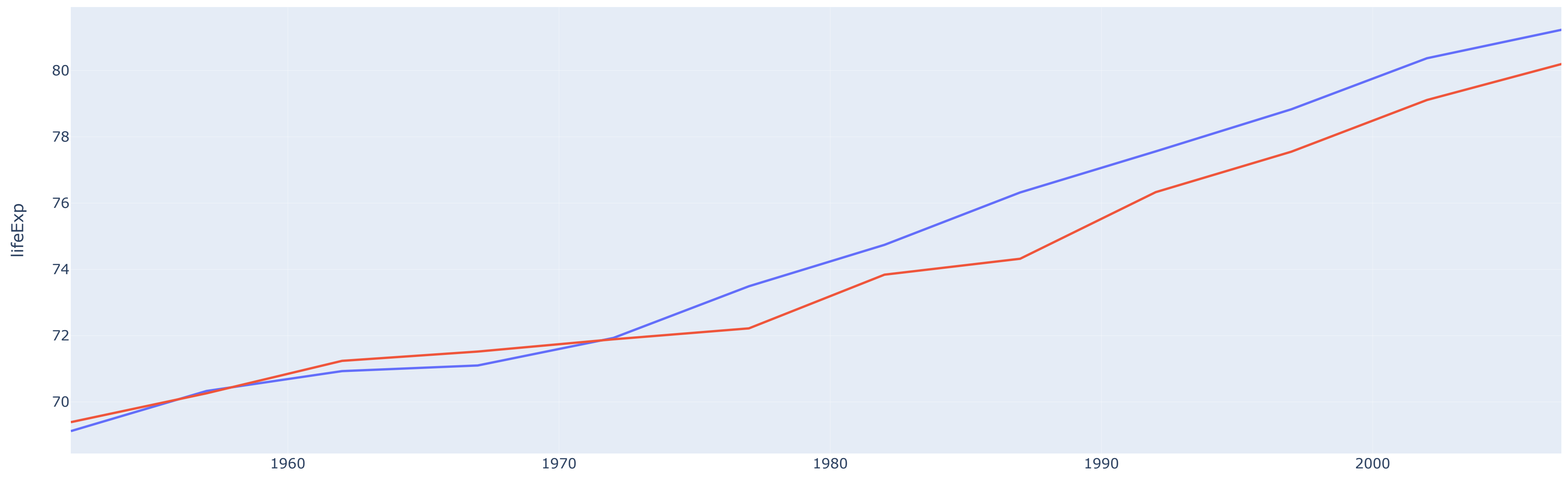
In [8]:
```
# scatterplot
import plotly.express as px

df = px.data.iris()

fig = px.scatter(df, x="sepal_width", y="sepal_length",
                 color='petal_length')

#fig.show()

fig.write_html('first_figure.html', auto_open=True)
```

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

df = px.data.tips()
fig = px.scatter(df, x="total_bill", y="tip",
                 trendline="ols", title='Fitted Line')
fig.show()
```

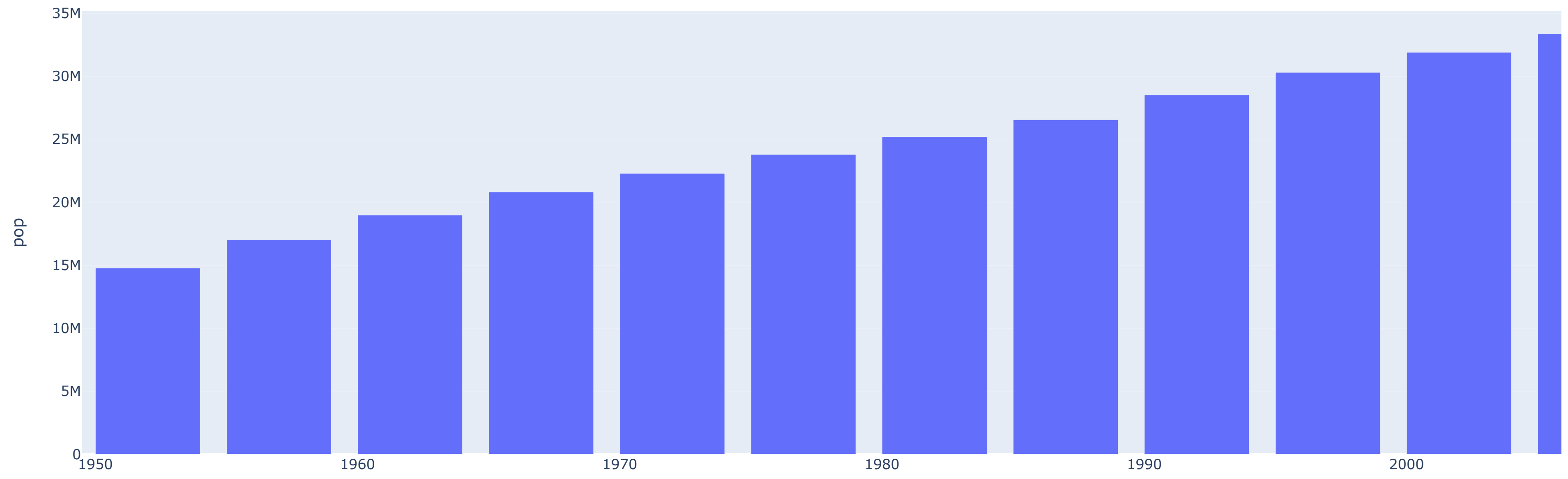
Fitted Line



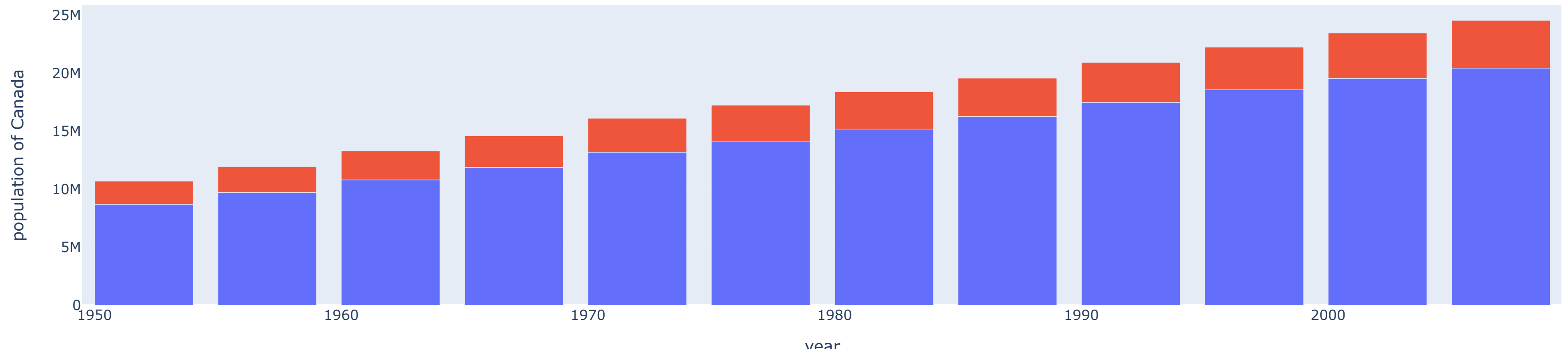
In [11]:
```
# line plot
import plotly.express as px

df = px.data.gapminder().query("continent=='Oceania'")

fig = px.line(df, x="year", y="lifeExp",
              color='country',
              hover_name='lifeExp')
fig.show()
```



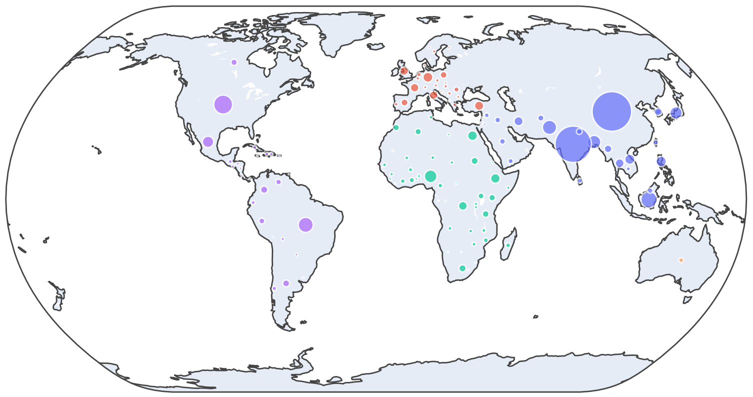
In [12]:
```
# bar chart-1
import plotly.express as px

data_canada = px.data.gapminder().query("country == 'Canada'")
fig = px.bar(data_canada, x='year', y='pop')
fig.show()
```



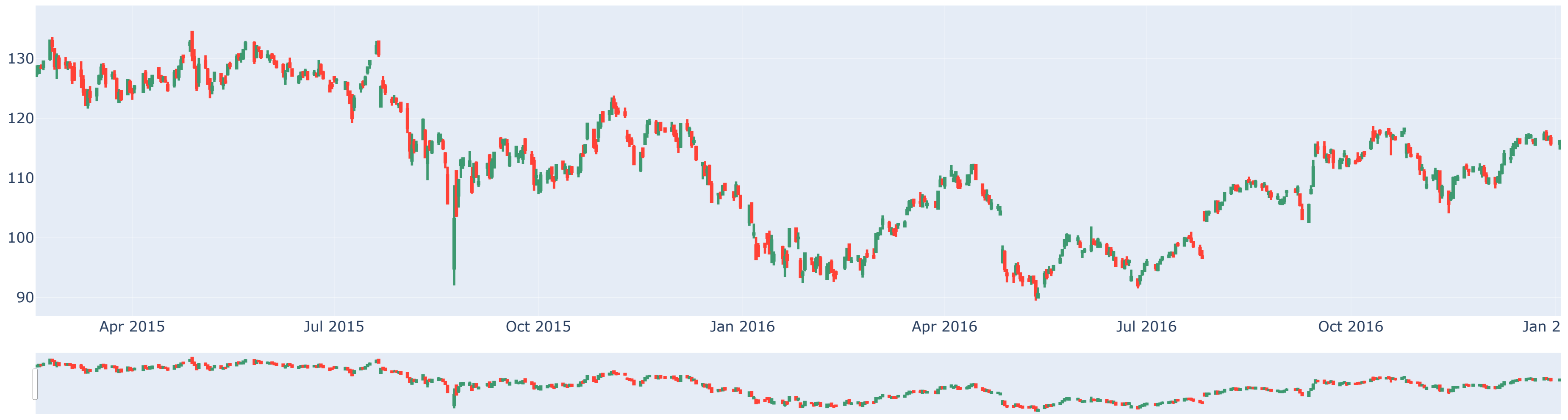
```
In [13]: # bar chart-2
         import plotly.express as px

         df = px.data.gapminder().query("continent == 'Oceania'")
         fig = px.bar(df, x='year', y='pop',
                      hover_data=['lifeExp', 'gdpPercap'], color='country',
                      labels={'pop':'population of Canada'}, height=400)
         fig.show()
```



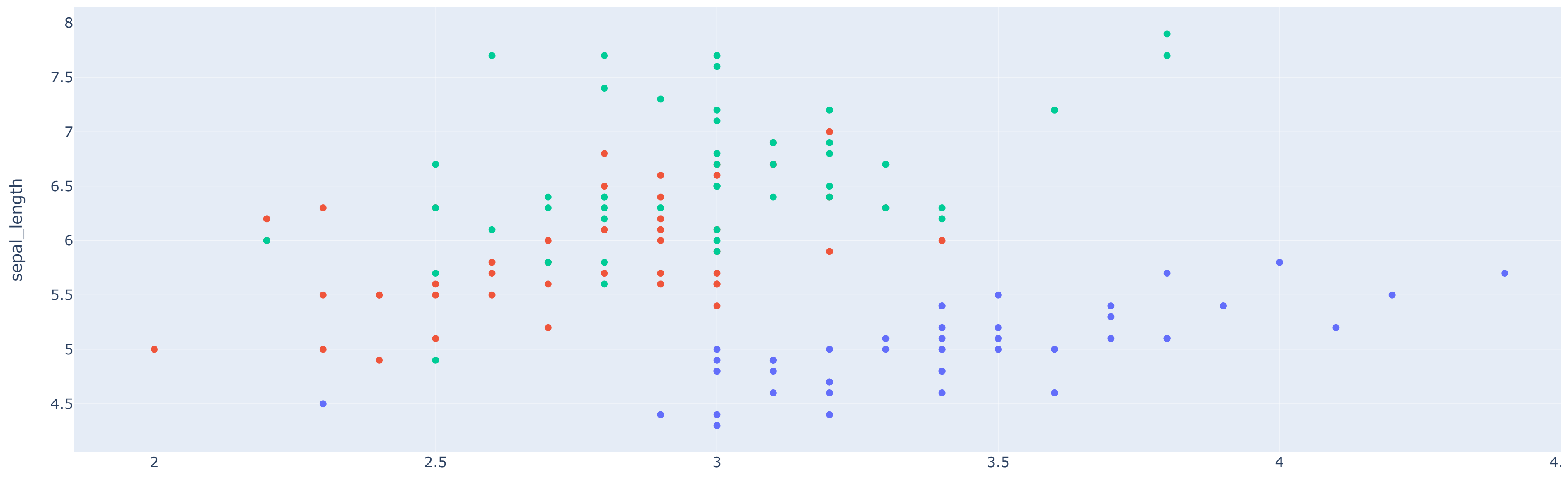
In [14]:
```python
# map
import plotly.express as px

df = px.data.gapminder().query("year==2007")
fig = px.scatter_geo(df, locations="iso_alpha", color="continent",
                     hover_name="country", size="pop",
                     projection="natural earth")
fig.show()
```



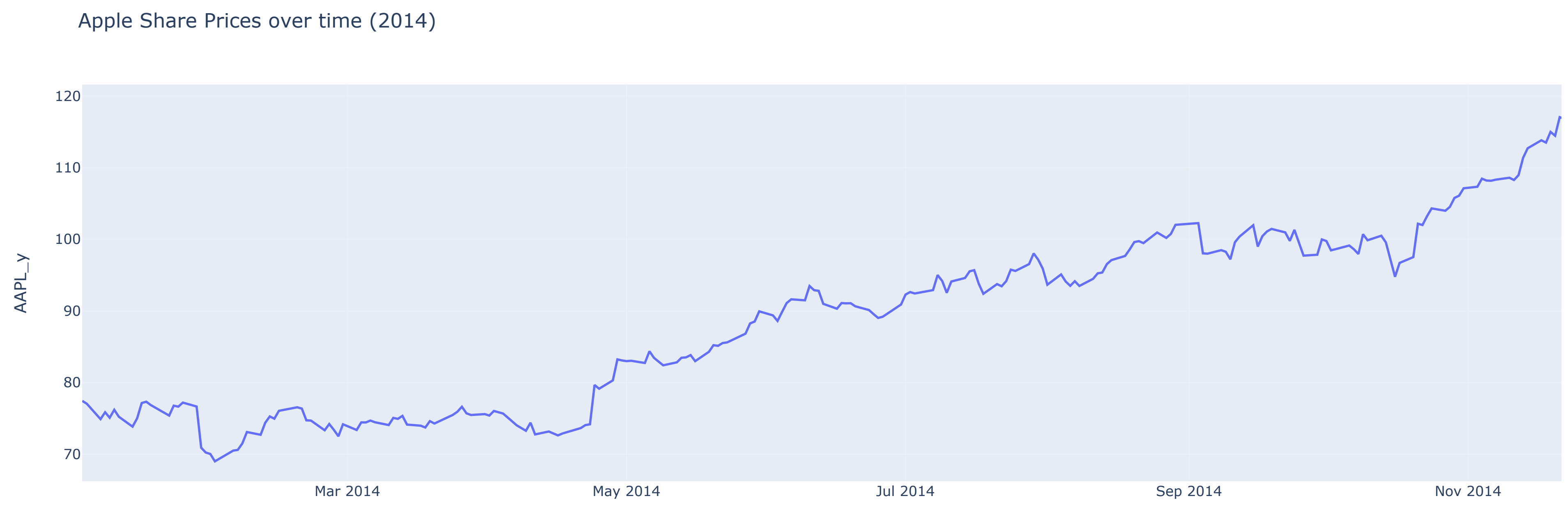
In [15]:
```python
# time series
import plotly.graph_objects as go

import pandas as pd
from datetime import datetime

df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-apple.csv')

fig = go.Figure(data=[go.Candlestick(x=df['Date'],
                open=df['AAPL.Open'],
                high=df['AAPL.High'],
                low=df['AAPL.Low'],
                close=df['AAPL.Close'])])

fig.show()
```



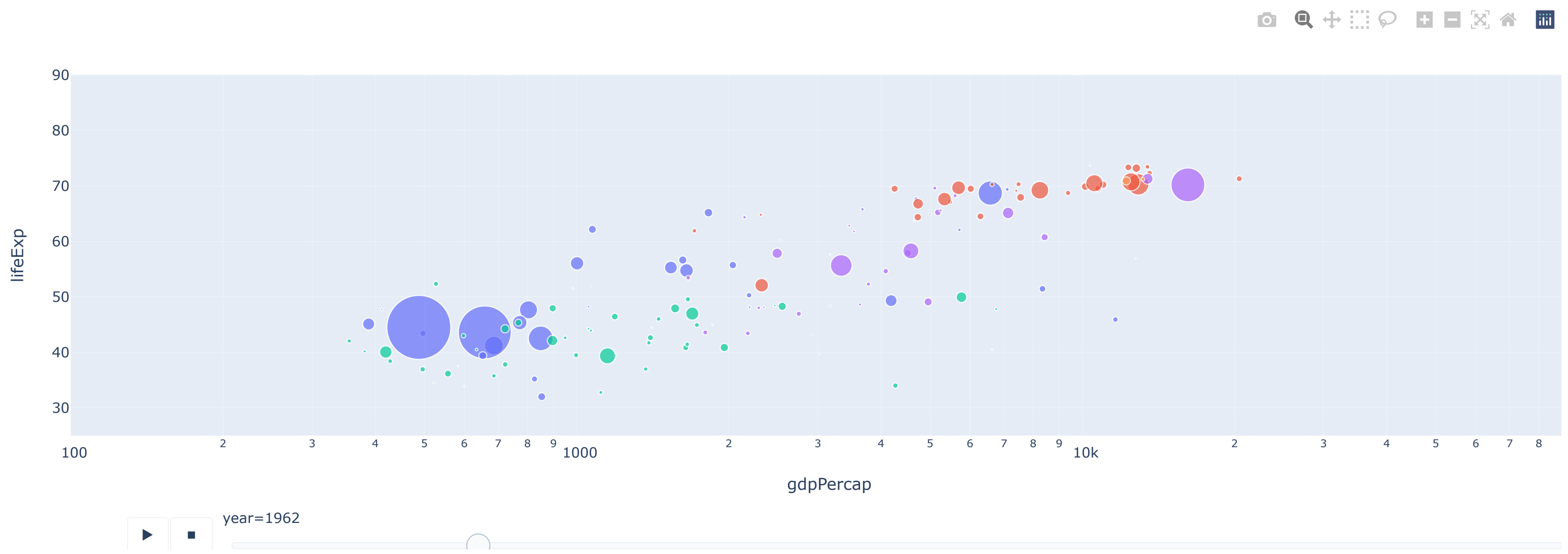
In [16]:
```
# a plot from pack
import plotly.express as px

df = px.data.iris()

fig = px.scatter(df, x="sepal_width", y="sepal_length",
                 color="species")
fig.show()
```



```
In [17]: # another example - web data

         df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/2014_apple_stock.csv')

         fig = px.line(df, x = 'AAPL_x', y = 'AAPL_y',
                       title='Apple Share Prices over time (2014)')
         fig.show()
```



In [18]:
```
# my favorite plot: multivariate, dynamic, interactive, free etc
# zoom in-out, play with features
# animated plot: how many variables can be employed? is it friendly?

import plotly.express as px

df = px.data.gapminder()

px.scatter(df, x="gdpPercap", y="lifeExp",
           animation_frame="year",
           animation_group="country",
           size="pop",
           color="continent",
           hover_name="country",
           log_x=True, size_max=55,
           range_x=[100,100000], range_y=[25,90])
```



Your Turn

- Practice more tools and code and include below.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

References

- Plotly
- Data sets and exports in Plotly, visit https://github.com/plotly/datasets
- Image export
