Explore the Gapminder Dataset with Plotly Exp

About the Data: Data Source (https://www.gapminder.org/tools/#$state$time$v
chart-type=bubbles)

Task 1: Loading the Data

In [2]:
```
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import pandas as pd
import numpy as np
```

In [3]:
```
import plotly.express as px
#It is high level wrapper around plotly like seaborn is to matplotlib
from plotly.figure_factory import create_table
```

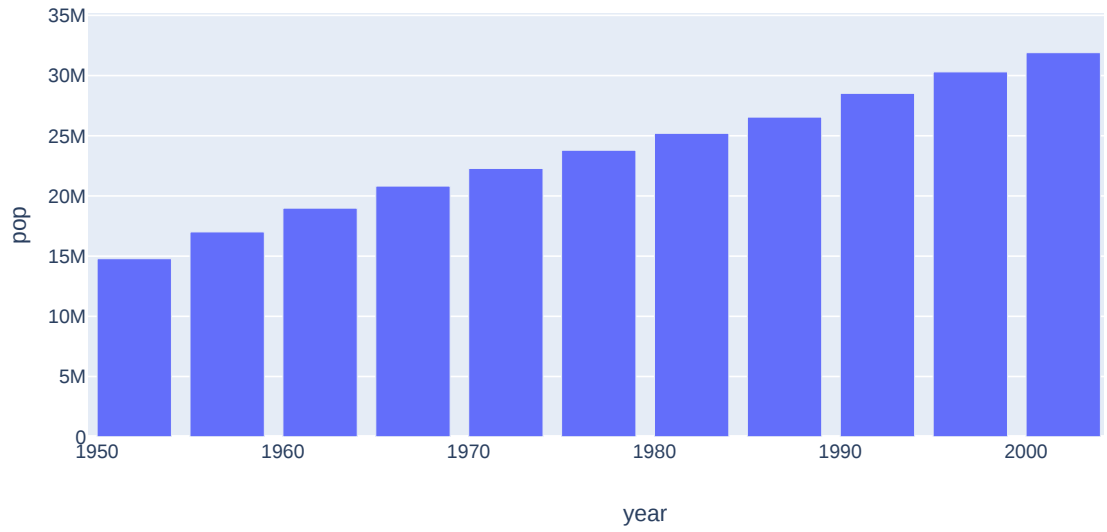
In [4]:
```
gapminder=px.data.gapminder()

table=create_table(gapminder.head(10))
py.iplot(table)
```

country	continent	year	lifeExp	pop	gdpPercap	iso_alpha
Afghanistan	Asia	1952	28.801	8425333	779.4453145	AFG
Afghanistan	Asia	1957	30.331999999999997	9240934	820.8530296	AFG
Afghanistan	Asia	1962	31.997	10267083	853.1007099999999	AFG
Afghanistan	Asia	1967	34.02	11537966	836.1971382	AFG
Afghanistan	Asia	1972	36.088	13079460	739.9811057999999	AFG
Afghanistan	Asia	1977	38.438	14880372	786.11336	AFG
Afghanistan	Asia	1982	39.854	12881816	978.0114388000001	AFG
Afghanistan	Asia	1987	40.821999999999996	13867957	852.3959447999999	AFG
Afghanistan	Asia	1992	41.674	16317921	649.3413952000001	AFG
Afghanistan	Asia	1997	41.763000000000005	22227415	635.341351	AFG

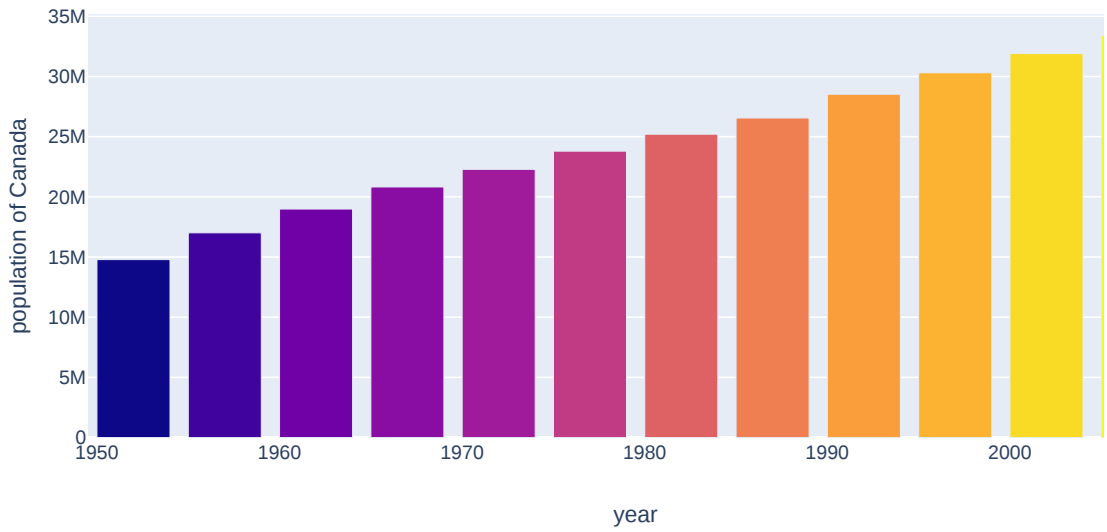
Task 2: Quick Visualizations with Custom Bar Charts

Note: If you are starting the notebook from this task, you can run cells from all in the kernel by going to the top menu and Kernel > Restart and Run All

In [5]:
```
data_canada=px.data.gapminder().query("country == 'Canada'")
fig=px.bar(data_canada, x='year', y='pop', height=400)
fig.show()
```



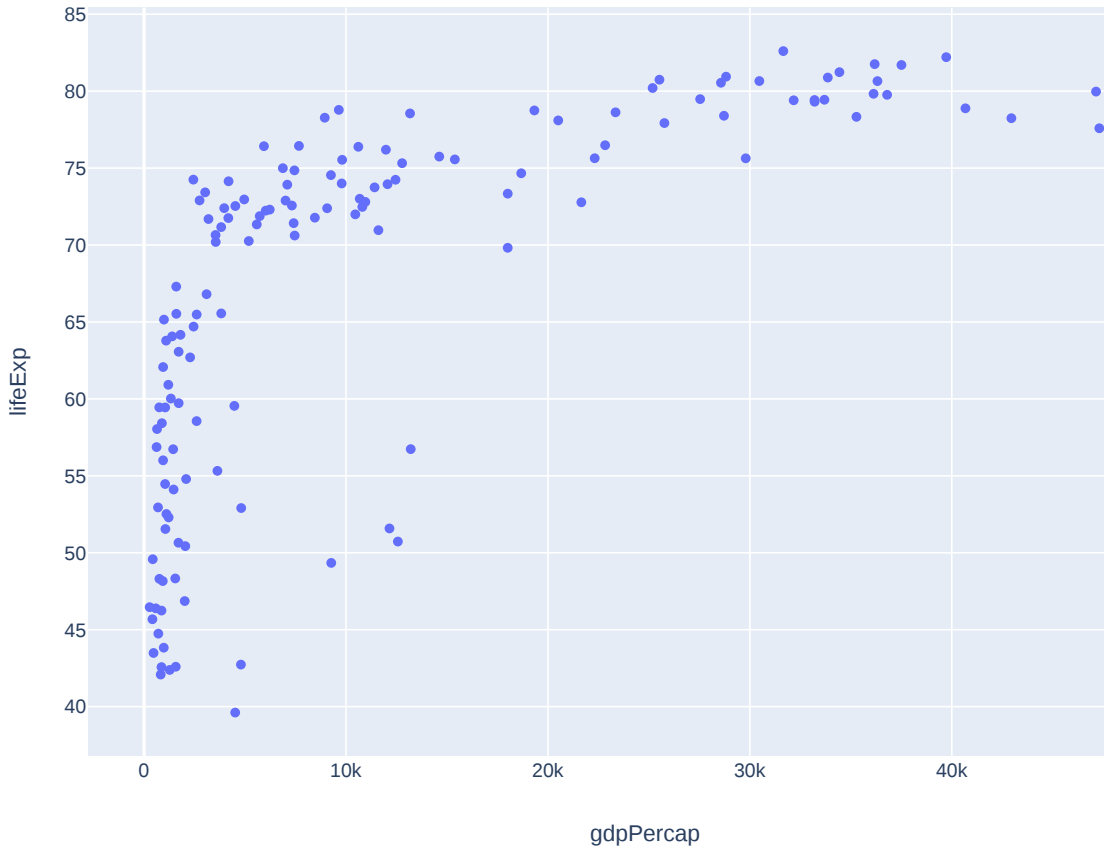
In [6]:
```
fig= px.bar(data_canada, x='year', y='pop',
            hover_data=['lifeExp', 'gdpPercap'], color='lifeExp',
            labels={'pop':'population of Canada'}, height=400)
fig.show()
```



Task 3: Plot Life Expectancy vs GDP per Capita

Note: If you are starting the notebook from this task, you can run cells from all in the kernel by going to the top menu and Kernel > Restart and Run All

In [7]:
```
gapminder2007= gapminder.query("year == 2007")

px.scatter(gapminder2007, x='gdpPercap', y='lifeExp')
```



In [8]:
```
gapminder2007.head()
```

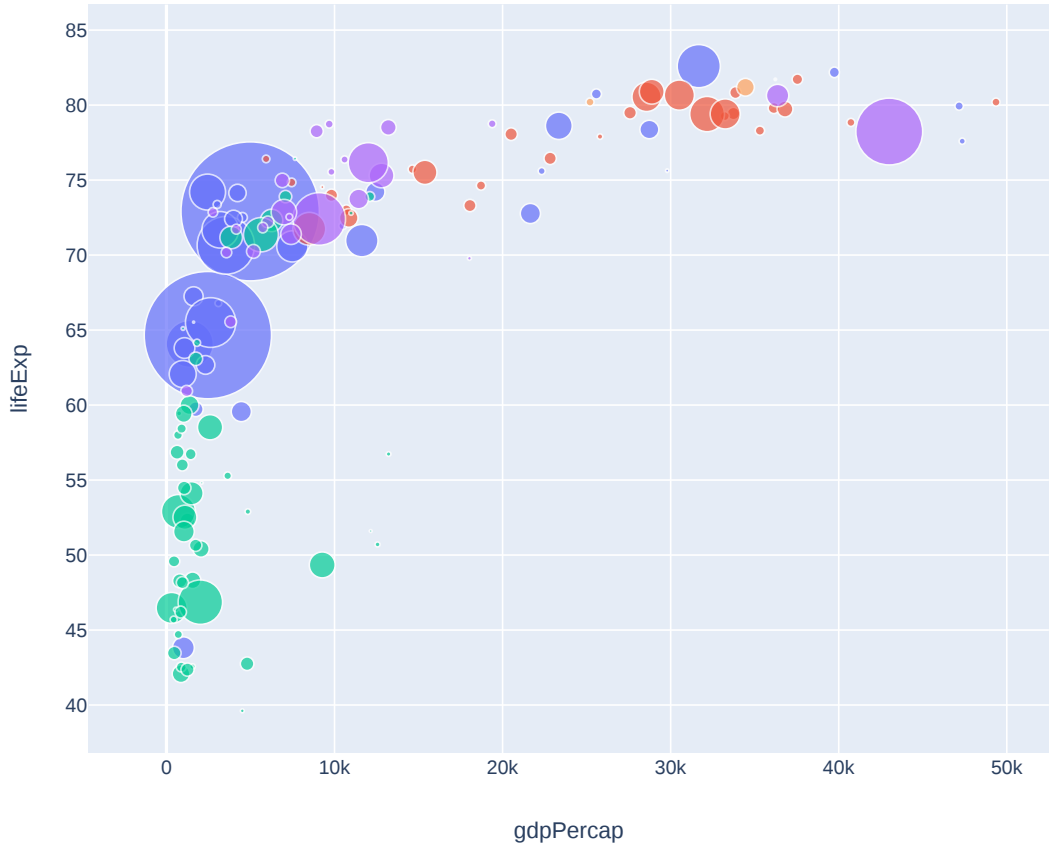
	country	continent	year	lifeExp	pop	gdpPercap	iso_alpha	iso_num
11	Afghanistan	Asia	2007	43.828	31889923	974.580338	AFG	4
23	Albania	Europe	2007	76.423	3600523	5937.029526	ALB	8
35	Algeria	Africa	2007	72.301	33333216	6223.367465	DZA	12
47	Angola	Africa	2007	42.731	12420476	4797.231267	AGO	24
59	Argentina	Americas	2007	75.320	40301927	12779.379640	ARG	32

In [9]:
```
px.scatter(gapminder2007, x="gdpPercap", y="lifeExp", color='continent')
```

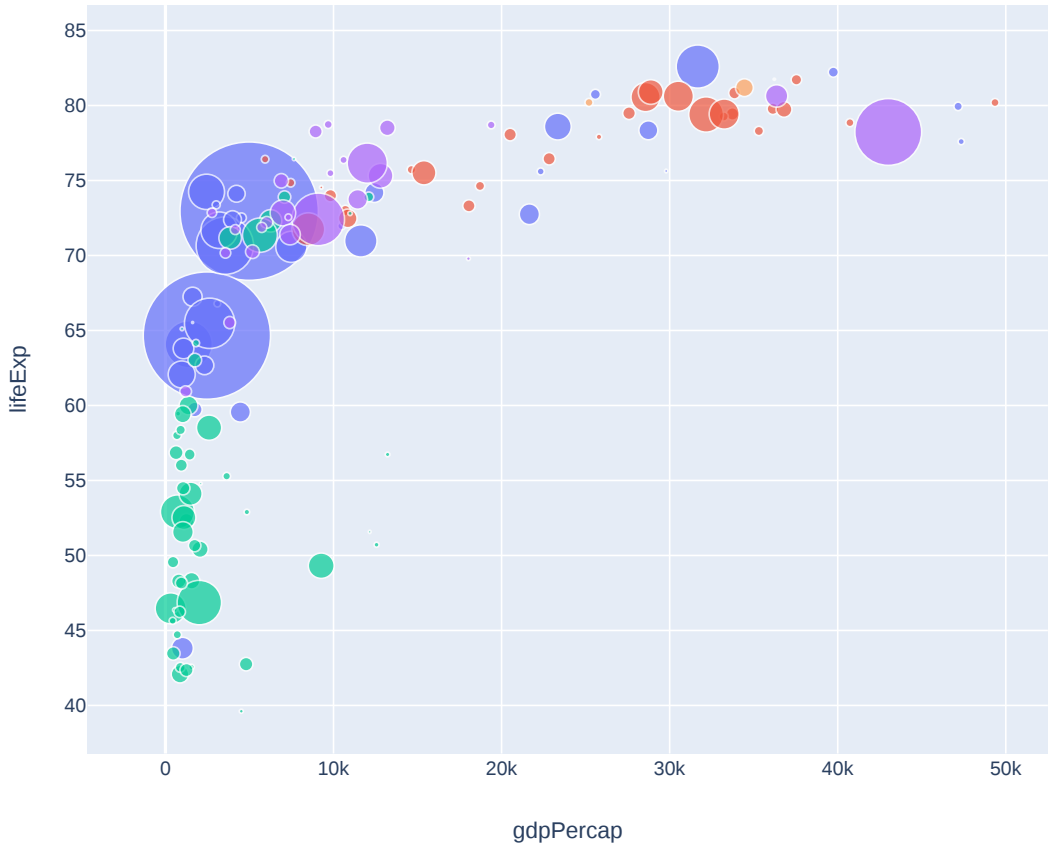
Task 4: Customize Interactive Bubble Charts

Note: If you are starting the notebook from this task, you can run cells from all in the kernel by going to the top menu and Kernel > Restart and Run All

In [10]:
```
px.scatter(gapminder2007, x="gdpPercap", y="lifeExp", color='continent',
           size="pop", size_max=60)
#each point is scaled by population of the country
```



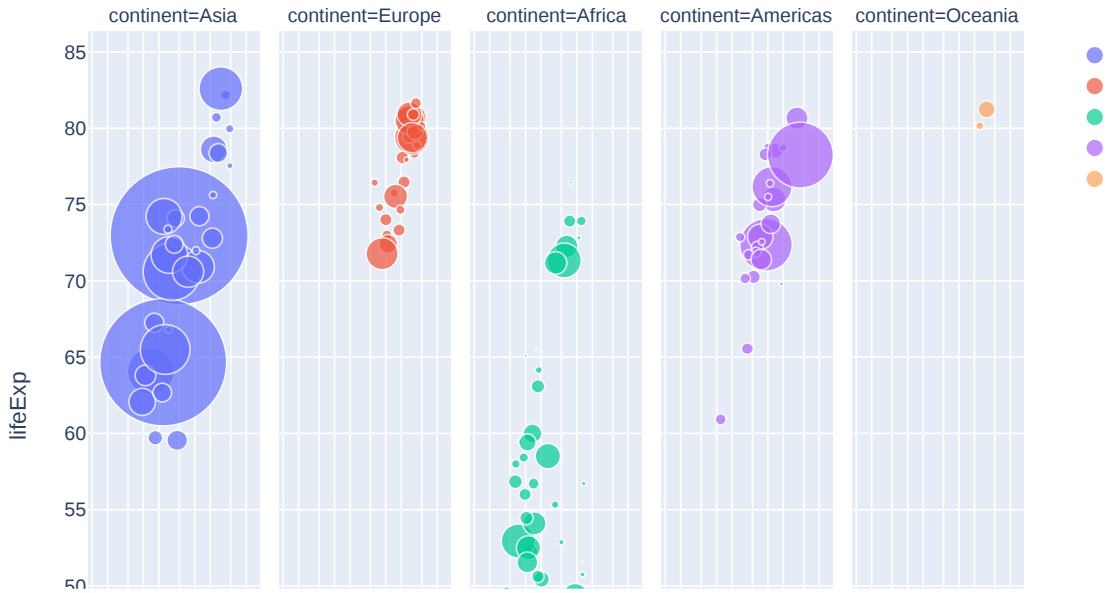
In [11]:
```
px.scatter(gapminder2007, x="gdpPercap", y="lifeExp", color="continent",
           size="pop", size_max=60, hover_name="country")
```



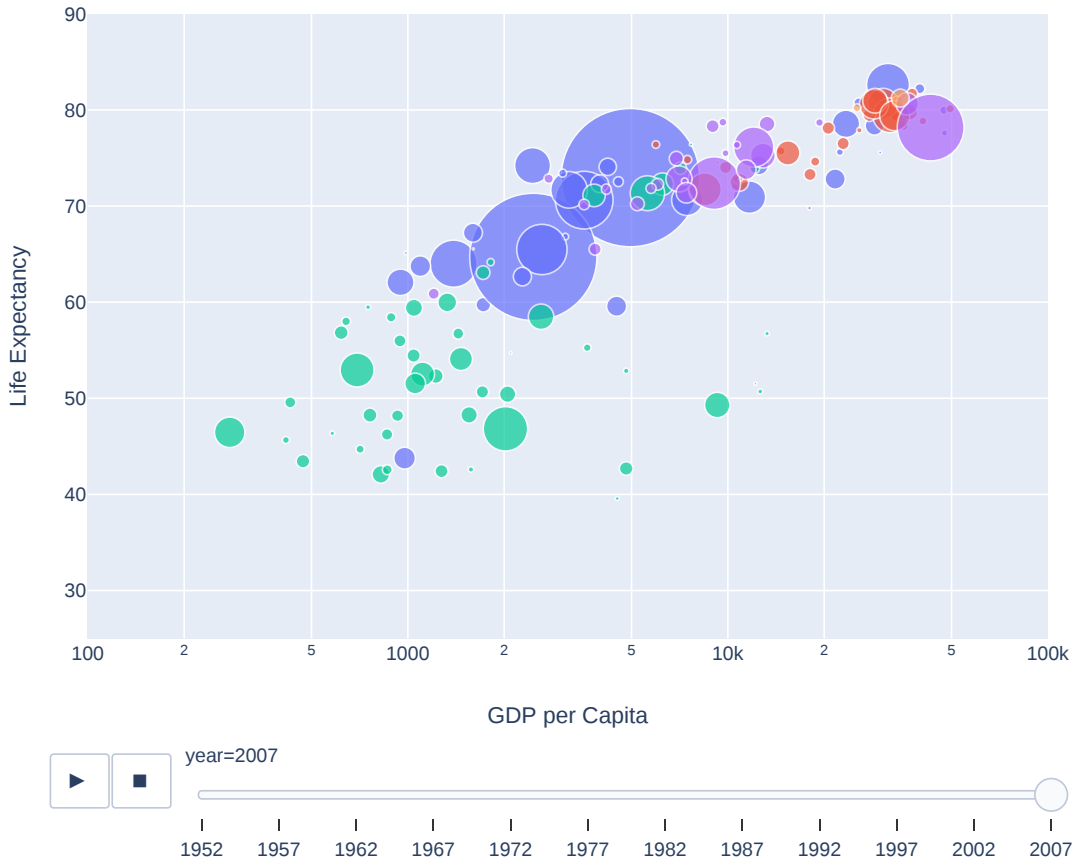
Task 5: Create Interactive Animations and Facet Plots

Note: If you are starting the notebook from this task, you can run cells from all in the kernel by going to the top menu and Kernel > Restart and Run All

In [12]:
```
px.scatter(gapminder2007, x="gdpPercap", y="lifeExp", color="continent",
           size="pop", size_max=60, hover_name="country", facet_col="continent",
           log_x=True)
```



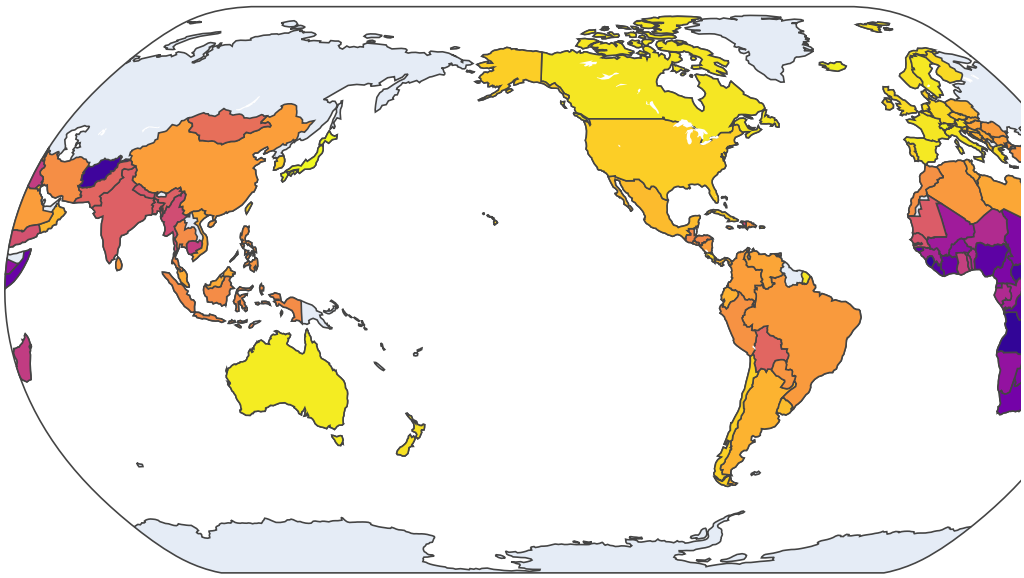
In [13]:
```
px.scatter(gapminder, x="gdpPercap", y="lifeExp", color="continent", size="pop",
           size_max=60, hover_name="country", animation_frame="year",
           animation_group="country", log_x=True, range_x=[100, 100000],
           range_y=[25, 90], labels=dict(pop="Population", gdpPercap="GDP per Cap
                                         lifeExp="Life Expectancy"))
```



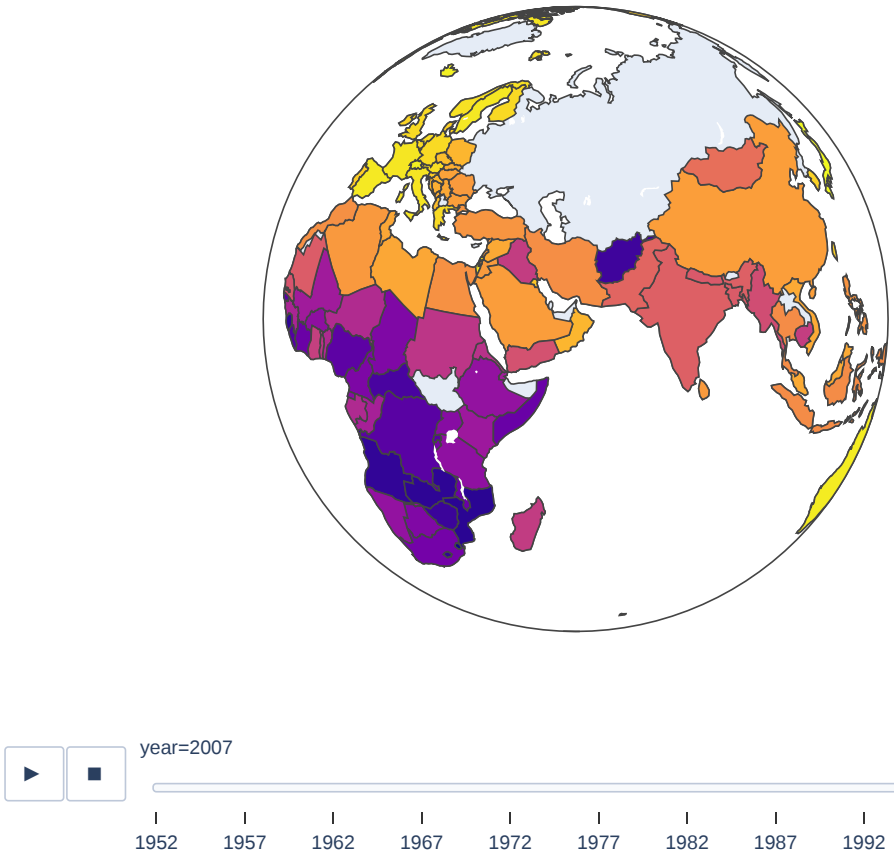
Task 6: Represent Geographic Data as Animated Maps

Note: If you are starting the notebook from this task, you can run cells from all in the kernel by going to the top menu and Kernel > Restart and Run All

In [14]:
```
px.choropleth(gapminder, locations="iso_alpha", color="lifeExp", hover_name="cou
              animation_frame="year", color_continuous_scale=px.colors.sequentia
              projection="natural earth")
```



In [15]:
```
px.choropleth(gapminder, locations="iso_alpha", color="lifeExp", hover_name="cou
              animation_frame="year", color_continuous_scale=px.colors.sequentia
              projection="orthographic")
```



Task 7: Interactive Line Plots and Area Plots

Note: If you are starting the notebook from this task, you can run cells from all in the kernel by going to the top menu and Kernel > Restart and Run All

In [ ]:

In [ ]:

In [ ]: