CSCI6612 - Visual Analytics

Assignment 3

Fall 2023

Due on: 24 October 2023, 23:59 ADT

Tirumala Vinjamuri Abhinav Acharya

ab806657@dal.ca B00929073

Dugar Arihant

ar968345@dal.ca B00917961

QUESTION 1

In this question, you study the Dash tutorial, and you will experiment with some basic visualization algorithms that you will embed within Dash. You will use the Gapminder dataset. Make use of code examples available in the Dash tutorial and Plotly documentation, and clearly reference the URLs of the classes / methods you reused.

Dashboard Creation with Dash

Using Dash, you will create a dashboard that displays multiple interactive coordinated visualizations of the dataset. Your general exploration objective is to understand whether and how life expectancy correlates with population, GDPperCapita, and continent, over the time period covered by the dataset.

Components of the Dashboard

The components of your dashboard will be visualizations selected from the above examples, plus any other visualizations that you consider helpful for the exploration objective stated. Interesting exploration objectives to consider are:

1. The evolution of the gap between high, middle and low income countries over time. The gap can be based on life expectancy or GDP per capita.
2. The slope of the linear dependency between life expectancy and GDP per capita over time.
3. How the GDP distribution per capita changes over time.
4. What is the world GDP and world life expectancy over time as a reference level. The world GDP per capita / life expectancy is a weighted average of GDP per capita / life expectancy over the world countries, where the weight is the population of each country.

Instead of a focus on continents, you may focus on the income level defined based on GDP per capita in US$ as:

- Low-income (0-1000), lower middle income (1000-4000),
- Upper middle income (4000-13500), and
- High income countries (13500+).

In the visualizations that do not explicitly represent GDP per capita you can use different colour for the different income levels. Countries near income level boundaries may switch income levels over time, which can form the basis of an exploration objective.

1. Load the simplified Gapminder dataset, and convert the loaded data to a pandas dataframe, as it plays well with Plotly Express.

In [1]:

```
import plotly.express as px
import pandas as pd
import numpy as np
data = px.data.gapminder()

print(type(data))
```

```
<class 'pandas.core.frame.DataFrame'>
```

The loaded dataset is already in a type of `Pandas DataFrame`. So there's no need of converting it to a dataframe again.

In [2]:

```
display(data.describe())
```

	year	lifeExp	pop	gdpPercap	iso_num
count	1704.00000	1704.000000	1.704000e+03	1704.000000	1704.000000
mean	1979.50000	59.474439	2.960121e+07	7215.327081	425.880282
std	17.26533	12.917107	1.061579e+08	9857.454543	248.305709
min	1952.00000	23.599000	6.001100e+04	241.165876	4.000000
25%	1965.75000	48.198000	2.793664e+06	1202.060309	208.000000
50%	1979.50000	60.712500	7.023596e+06	3531.846989	410.000000
75%	1993.25000	70.845500	1.958522e+07	9325.462346	638.000000
max	2007.00000	82.603000	1.318683e+09	113523.132900	894.000000

2. Explain your specific exploration objective.

Answer:

1. Life Expectancy vs. GDP per Capita (Graph 1)

- Analyze the relationship between life expectancy and GDP per capita over time.
- Observe how different continents or countries vary in terms of this relationship.
- Explore how color coding by income group or life expectancy affects the visualization.

2. Life Expectancy vs. Continents (Graph 2)

- Understand how life expectancy varies across different continents.
- Observe changes in life expectancy for different continents over time.
- Investigate any outliers or unique trends.

3. Life Expectancy vs. Population (Graph 3)

- Examine how life expectancy correlates with population for different continents or countries.
- Investigate if there are trends regarding the impact of population on life expectancy.
- Investigate any outliers or unique trends.

4. Countries' Evolution (Graph 4)

- Explore the evolution of life expectancy or GDP per capita for individual countries within a chosen continent.
- Observe differences between countries in the same continent.
- Analyze trends for specific countries over time.

5. Slope of Linear Dependency (Graph 5)

- Calculate and interpret the slope of the linear relationship between life expectancy and GDP per capita over time.
- Assess how the strength and direction of this relationship change over the years.

6. GDP Per Capita Changes Over Time (Graph 6)

- Analyze how GDP per capita changes over time for continents. Also analyse the same for all the countries in a continent
- Investigate the difference in growth trends between continents.

7. World GDP and Life Expectancy (Graph 7)

- Observe the world GDP and life expectancy over time as reference levels.
- Analyze whether these reference levels are improving or declining over time.
- Evaluate the relationship between world GDP and life expectancy.

8. Country's Key Performance Analytics (Graph 8)

- Select a country's key performance attributes such as life expectancy, population, or GDP per capita.
- Analyze the performance of specific countries over time.
- Identify any interesting trends or patterns for individual countries.

9. Geographical Representation of Growth (Graph 9)

- Explore geographical representations of key indicators, such as life expectancy, GDP per capita, or population.
- Assess how these indicators change over time across countries.
- Utilize color coding to add additional dimensions to the visualizations.

3. Describe the visualizations included in the dashboard you design for your exploration objective. Justify your choice of visualizations.

Answer:

Graph 1: Life Expectancy vs. GDP per Capita

- **Visualization Type**: Scatter Plot
- **Objective**: To understand the relationship between life expectancy and GDP per capita over time. This graph allows users to observe trends, correlations, and variations across continents or countries.
- **Justification**: Scatter plots are effective for visualizing the relationship between two continuous variables. The use of color coding (by continent, income group, or life expectancy) adds depth to the analysis, enabling users to explore multiple dimensions within the same graph. The animation by year provides a dynamic view of how these relationships change over time.

Graph 2: Life Expectancy vs. Continents

- **Visualization Type**: Scatter Plot
- **Objective**: To observe and compare life expectancy across different continents over time. This graph helps in identifying continent-specific trends and variations.
- **Justification**: Similar to Graph 1, a scatter plot is used for its effectiveness in comparing two continuous variables. By plotting life expectancy against continents, users can quickly identify trends and differences across major geographical regions, while animation by year provides a temporal view of changes.

Graph 3: Life Expectancy vs. Population

- **Visualization Type**: Scatter Plot
- **Objective**: To explore how life expectancy correlates with population size for different continents or countries. This graph aids in understanding whether there is any significant impact of population on life expectancy.
- **Justification**: Scatter plots are well-suited for examining the relationship between two numerical variables. The size of the data points and color coding enhance the visual exploration, while animation by year provides a temporal view of changes.

Graph 4: Countries' Evolution

- **Visualization Type**: Bar Chart
- **Objective**: To analyze the evolution of life expectancy or GDP per capita for individual countries within a selected continent. The objective is to compare the performance of countries within the same region.
- **Justification**: Bar charts are useful for comparing values across different categories (countries in this case). The use of animation by year and color coding allows users to track changes over time and discern differences between countries.

Graph 5: Slope of Linear Dependency

- **Visualization Type**: Scatter Plot with Regression Line
- **Objective**: To calculate and visualize the slope of the linear relationship between life expectancy and GDP per capita over time. The goal is to assess the strength and direction of this relationship.
- **Justification**: Scatter plots with regression lines are suitable for assessing the strength and direction of linear relationships. They provide a clear visualization of the trend, and the trendline aids in quantifying the relationship. Moreover, Log transformation on the trendline made sure that the trendline aligns/ overlaps with the majority of the data points.

Graph 6: GDP Per Capita Changes Over Time

- **Visualization Type**: Scatter Plot
- **Objective**: To observe how GDP per capita changes over time for continents and countries. Users can explore the differences in growth trends across geographical regions.
- **Justification**: Lines are ideal for showing trends over time, so we connect the scattered points on the graph area, and scatter plots are used for individual country-level analysis. The choice of a connected line between scatter points allows users to focus on aggregated trends, while the second scatter plots help dive deeper into country-level data and on hover it provides a more detailed description.

Graph 7: World GDP and Life Expectancy

- **Visualization Type**: Scatter Plot (with the option to flip axes) and Line Plot (For connecting the scatter points)
- **Objective**: To visualize world GDP and world life expectancy over time as reference levels. Users can assess changes in these reference levels and their relationship.
- **Justification**: A scatter plot is chosen to visualize two continuous variables. The ability to flip axes allows users to switch their focus between world GDP and life expectancy, providing flexibility in exploration.

Graph 8: Country's Key Performance Analytics

- **Visualization Type**: Area Chart
- **Objective**: To select and analyze the performance of specific countries over time, focusing on key attributes like life expectancy, population, or GDP per capita.
- **Justification**: Area charts are suitable for comparing the performance of multiple countries over time. Users can select the attribute of interest (y-axis) and track changes in performance.

Graph 9: Geographical Representation of Growth

- **Visualization Type**: Choropleth Map
- **Objective**: To explore geographical representations of key indicators (life expectancy, GDP per capita, or population) over time and assess their changes across countries. Users can use color coding to add additional dimensions using radio buttons.
- **Justification**: Choropleth maps are effective for visualizing spatial data. Users can select the data to be displayed on the y-axis and choose how to color code the map to provide a geographical context.

4. Explain the user interactions included in your dashboard. Justify your choice of interactions.

Answer:

1. Radio Buttons (Graph 1, Graph 3, and Graph 4)

- **User Interaction**: Users can select between different data views using radio buttons, such as choosing to visualize data by "Continents" or "Countries."
- **Justification**: Radio buttons offer a simple and intuitive way to toggle between data views, allowing users to focus on either continent-level or country-level analysis. This interactivity helps users control the scope of their exploration.

2. Dropdown Menus (Graph 1, Graph 3, and Graph 4)

- **User Interaction**: Users can select a specific continent using dropdown menus to see data related to that continent.
- **Justification**: Dropdown menus enable users to filter and narrow down data to a specific continent of interest. This is valuable for drilling down into regional insights or making targeted comparisons.

3. Color Coding Radio Buttons (Graph 1 and Graph 9)

- **User Interaction**: Users can select how the data points are color-coded, whether by "Continents," "Countries," "Income Group," or "Life Expectancy Group."
- **Justification**: Color coding adds an additional dimension to the visualization, facilitating the differentiation and comparison of data points. Users can choose the aspect they want to emphasize, depending on their analysis goals.

4. Button to Flip Axes (Graph 7)

- **User Interaction**: Users can click a button to flip the axes of the scatter plot in Graph 7 between "World GDP" and "World Life Expectancy."
- **Justification**: Flipping the axes of the scatter plot provides users with flexibility to switch their focus between GDP and life expectancy as the dependent variable. This option accommodates different exploration objectives.

5. Graph Hover (All Graphs)

- **User Interaction**: Users can hover over data points on the graph to access additional information such as country names or continent names.
- **Justification**: Hovering over data points offers an easy way to access specific details without cluttering the visualization. Users can quickly identify data points of interest and retrieve relevant context.

6. Graph Hover (Graph 6)

- **User Interaction**: Users can hover over data points on the graph to dynamically generate additional views/ graphs.
- **Justification**: Hovering over data points dynamically updates another graph which visualizes granularly for countries. Users can quickly identify data points of interest and retrieve relevant context.

7. Dropdown Menu for Selecting the Y-Axis (Graph 8)

- **User Interaction**: Users can select the attribute (life expectancy, population, or GDP per capita) to be displayed on the y-axis of the area chart.
- **Justification**: The dropdown menu empowers users to choose the specific attribute they want to focus on for the area chart, making it a personalized exploration tool.

8. Radio Buttons for Selecting Color Coding (Graph 9)

- **User Interaction**: Users can choose how the choropleth map should be color-coded, whether by "default," "Income Group," or "Life Expectancy."
- **Justification**: Color coding in choropleth maps helps convey additional information about the data. Users can choose the aspect that is most relevant to their exploration goals.

These user interactions were chosen to strike a balance between flexibility and simplicity. They allow users to customize their exploration while keeping the interface intuitive and easy to navigate. The choice of interactions is aligned with the specific objectives of each graph, enhancing the overall user experience and enabling more meaningful data analysis.

In [3]:

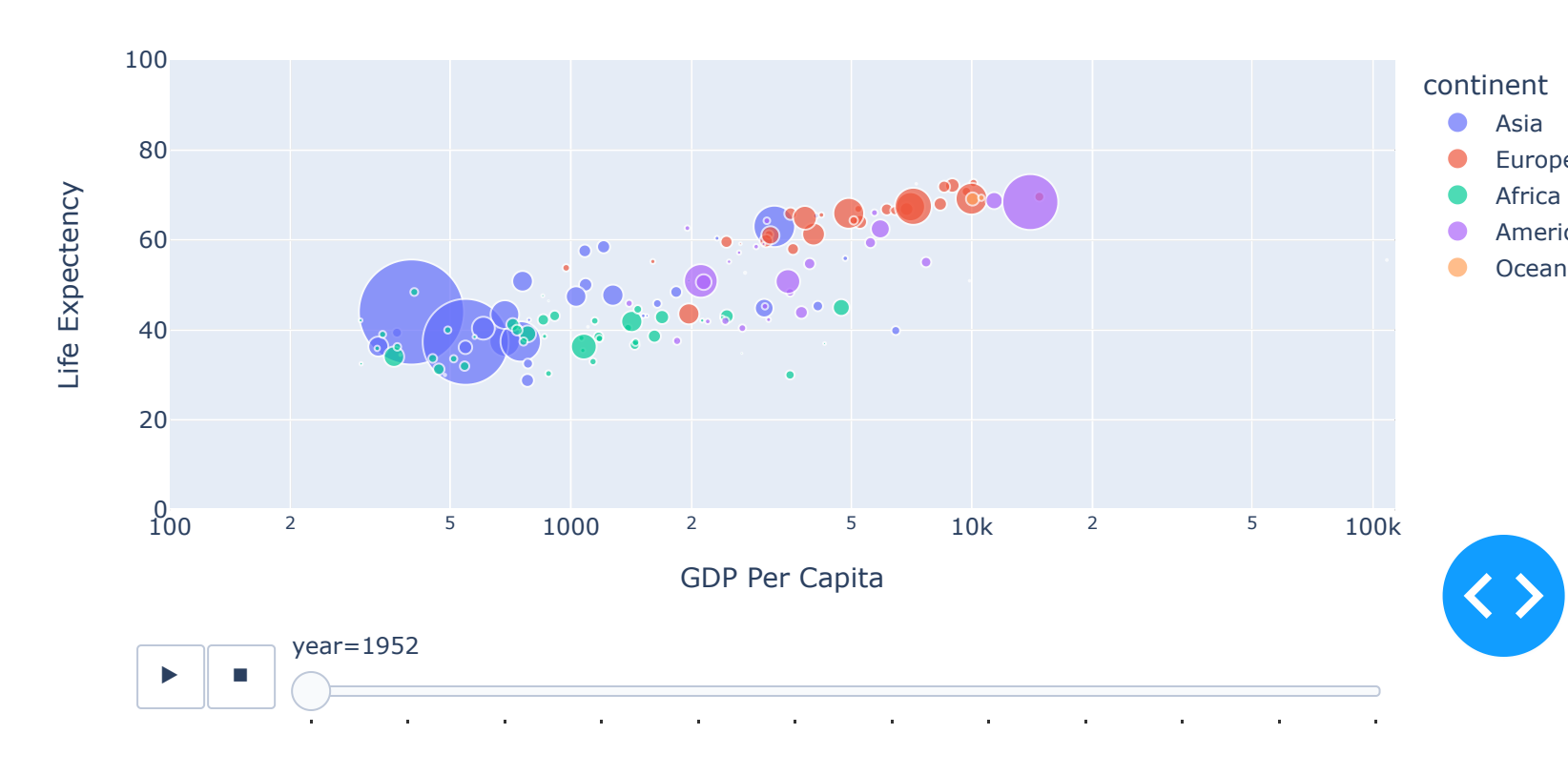
```
# Code of your first dashboard here
import plotly.express as px
import dash
from dash import dcc
from dash import html
from dash.dependencies import Input, Output, State
import plotly.express as px
import pandas as pd
import numpy as np

# Load dataset
data = px.data.gapminder()

data['IncomeGroup'] = pd.cut(data['gdpPercap'],
                        bins=[0, 1000, 4000, 13500, float('inf')],
                        labels=['Low (0 - 1000)', 'Lower middle (1001 - 4000)', 'Upper middle (4001 - 13500)', 'High (1
data['LifeExpectancyGroup'] = pd.cut(data['lifeExp'],
                                bins=[0, 40, 55, 70, 101],
                                labels=['Low (0 - 40)', 'Lower middle (41 - 55)', 'Upper middle (56 - 70)', 'High (71 -


app = dash.Dash(__name__)

@app.callback(
    [Output('g1-continent-dropdown', 'style'),
     Output('g1-dropdown-header', 'style'),
     Output('g1-dropdown-header', 'children'),
     Output('g3-continent-dropdown', 'style'),
     Output('g3-dropdown-header', 'style'),
     Output('g3-dropdown-header', 'children')],
    [Input('g1-data-radio-value', 'value'),
     Input('g3-radio-value', 'value')]
)
def toggle_dropdown(g1_radio_value, g3_radio_value):
    def handle_visibility(radio_value):
        if radio_value == "Countries":
            return {'display': 'block', 'margin-right':'4%'}, {'display': 'block'}, "Select a continent to see all of i
        else:
            return {'display': 'none'}, {'display': 'none'}, ""

    return handle_visibility(g1_radio_value) + handle_visibility(g3_radio_value)

@app.callback(
    Output('life-exp-vs-gdp', 'figure'),
    [Input('g1-continent-dropdown', 'value'), Input('g1-data-radio-value', 'value'), Input('g1-color-code-radio-value',
)
def update_graph(continent_value, data_radio_value, color_code_radio_value):
    # Because of the order of if blocks, the color code wont update when we select data_radio_type.
    # But user has the full freedom to choose their color acc to color_code_radio_value
    if data_radio_value == 'Countries':
        dff = data[data['continent'] == continent_value]
        color_word = "country"
    else:
        dff = data
        color_word = "continent"

    if color_code_radio_value == 'Continents':
        color_word = "continent"
    elif color_code_radio_value == 'Countries':
        color_word = 'country'
    elif color_code_radio_value == 'Income Group':
        color_word = 'IncomeGroup'
    else:
        color_word = 'LifeExpectancyGroup'

    fig = px.scatter(dff, x="gdpPercap", y="lifeExp",
                     size="pop", color=color_word, hover_name='country',
                     log_x=True, size_max=60, animation_frame='year',
                     animation_group="country", range_x=[100, dff['gdpPercap'].max()],
                     range_y=[0,100], labels=dict(pop="Population",
                     gdpPercap="GDP Per Capita", lifeExp="Life Expectency", IncomeGroup="Income Group", LifeExpectancyG

    return fig

def update_graph_continent():
    fig = px.scatter(data, x="continent", y="lifeExp",
                     size="pop", color="continent", hover_name="country", size_max=60, animation_frame='year',
                     animation_group="country", range_y=[0,100], labels=dict(lifeExp="Life Expectancy", IncomeGroup="In

    return fig

@app.callback(
    Output('life-exp-vs-population', 'figure'),
    [Input('g3-continent-dropdown', 'value'), Input('g3-radio-value', 'value')]
)
def update_graph_population(continent_value, radio_value):
    if radio_value == 'Countries':
        dff = data[data['continent'] == continent_value]
        color_word = "country"
        hover_word = "continent"
    else:
        dff = data
        hover_word = "country"
        color_word = "continent"

    fig = px.scatter(dff, x="pop", y="lifeExp",
                     size="pop", color=color_word, hover_name=hover_word,
                     log_x=True, size_max=60, animation_frame='year',
                     animation_group="country", range_y=[0,100],range_x=[100000,2000000000], labels=dict(pop="Populatio
                     lifeExp="Life Expectency"))

    return fig

# The evolution of the gap between high, middle and low income countries over time. The gap can be based on life expect
@app.callback(
        Output('countries-evolution', 'figure'),
        [Input('g4-continent-dropdown', 'value'),
         Input('g4-radio-value', 'value')]
)
def country_evolution(continent_value, radio_value):
    fig = px.bar(data[data['continent']==continent_value], x='country', y='lifeExp' if radio_value == "Life Expectency"
                 color='LifeExpectancyGroup' if radio_value == "Life Expectency" else 'IncomeGroup', animation_frame="y

    max_value = data['lifeExp' if radio_value == "Life Expectency" else 'gdpPercap'].max()
    fig.update_layout(yaxis=dict(range=[0, max_value]))
    return fig

# The slope of the linear dependency between life expectancy and GDP per capita over time.
def gdp_linear_regression():
    fig = px.scatter(data, x='gdpPercap', y='lifeExp', trendline='ols', trendline_options=dict(log_x=True), animation_f
    return fig


# How GDP per capita changes over time for continents
@app.callback(
    Output('gdp-over-time-1', 'figure'),
    [Input('g6-1-radio-value', 'value')]
)
def update_gdp_overtime_1(y_type):
    dff = data.groupby(['continent', 'year']).agg({"gdpPercap":"mean"}).reset_index()
    fig = px.scatter(dff, x="year", y="gdpPercap", color="continent", hover_name='continent', labels=dict(gdpPercap="GD
    fig.update_traces(mode="lines+markers")
    fig.update_yaxes(type=y_type)
    fig.update_layout(hovermode='closest')
    return fig

# How GDP per capita changes over time for countries
@app.callback(
    Output('gdp-over-time-2', 'figure'),
    [Input('gdp-over-time-1', 'hoverData'),
     Input('g6-2-radio-value', 'value'),
    ]
)
def update_gdp_overtime_2(hoverData, y_type):
    dff = data[data['continent']==hoverData['points'][0]['hovertext']]
    fig = px.scatter(dff, x="year", y="gdpPercap", color="country", hover_name='country', labels=dict(gdpPercap="GDP pe
    fig.update_traces(mode="lines+markers")
    fig.update_yaxes(type=y_type)
    fig.update_layout(hovermode='closest')
    return fig

# World GDP/ Life expectancy as a reference level
@app.callback(
        Output('world-gdp-life-exp', 'figure'),
        [Input('g7-radio-value', 'value'),
         Input('g7-button', 'n_clicks')
        ]
)
def world_gdp_life(radio_value, n_clicks):

    dff = data.groupby('year').apply(lambda x: pd.Series({
    'world_gdp': np.average(x['gdpPercap'], weights=x['pop']),
    'world_life_exp': np.average(x['lifeExp'], weights=x['pop'])
    }))

    dff['IncomeGroup'] = pd.cut(dff['world_gdp'],
                            bins=[1000, 4000, 7000, float('inf')],
                            labels=['Lower middle (1000 - 4000)', 'Upper middle (4001 - 7000)', 'High (7000+)'])
    dff['LifeExpectancyGroup'] = pd.cut(dff['world_life_exp'],
                                    bins=[40, 50, 60, 101],
                                    labels=['Lower middle (40 - 50)', 'Upper middle (51 - 60)', 'High (60+)'])
    dff.reset_index(inplace=True)

    if n_clicks is None or n_clicks % 2 == 0:
        fig = px.scatter(dff, x='world_life_exp', y='world_gdp', color='IncomeGroup',
                     labels={"IncomeGroup":"Income Group", "world_gdp":"World GDP Per Capita (Weighted Average)",
                             "world_life_exp":"World Life Expectancy (Weighted Average)", "LifeExpectancyGroup":"Life E
        line = px.line(dff, x='world_life_exp', y='world_gdp').data[0]
    else:
        fig = px.scatter(dff, x='world_gdp', y='world_life_exp', color='LifeExpectancyGroup',
                     labels={"IncomeGroup":"Income Group", "world_gdp":"World GDP Per Capita (Weighted Average)",
                             "world_life_exp":"World Life Expectancy (Weighted Average)", "LifeExpectancyGroup":"Life E
        line = px.line(dff, y='world_life_exp', x='world_gdp').data[0]

    fig.update_xaxes(type='linear' if radio_value == 'linear' else 'log')
    fig.update_yaxes(type='linear' if radio_value == 'linear' else 'log')

    fig.add_trace(line)

    return fig

@app.callback(
    Output("country-key-performance", "figure"),
    Input("y-axis", "value"),
)
def country_key_performance(selected_yaxis):
    if selected_yaxis == 'lifeExp':
        yaxis_label = 'Life Expectancy'
    elif selected_yaxis == 'pop':
        yaxis_label = 'Population'
    elif selected_yaxis == 'gdpPercap':
        yaxis_label = 'GDP per Capita'
    countries = data.country.drop_duplicates().sample(n=10, random_state=42)
    dff = data[data.country.isin(countries)]
    fig = px.area(dff, x="year", y=selected_yaxis, color="continent", line_group="country")
    fig.update_layout(yaxis_title=yaxis_label, xaxis_title="Year")
    return fig

@app.callback(
    Output("graph", "figure"),
    [
        Input("datay", "value"),
        Input('g9-color-code-radio-value', 'value')
    ]
)
def map_box(selected_yaxis, colour_code_radio_value):

    if colour_code_radio_value == 'Income Group':
        color_word = 'IncomeGroup'
    elif colour_code_radio_value == 'Life Expectancy':
        color_word = 'LifeExpectancyGroup'
    else:
        color_word = selected_yaxis

    # Create a choropleth map
    fig = px.choropleth(
        data,
        locations='iso_alpha',  # Country codes or names
        color=color_word,        # Values to visualize
        color_continuous_scale='Viridis',  # Color scale
        projection='natural earth',  # Specify the map projection
        animation_frame='year',
        hover_name='country',
        labels=dict(lifeExp="Life Expectancy", IncomeGroup="Income Group", pop= "Population", gdpPercap="GDP per Capita
    )

    # Customize the layout if needed
    fig.update_geos(
        showcoastlines=True, coastlinecolor="Black",
        showland=True, landcolor="white",
        showocean=True, oceancolor="lightblue",
    )
    fig.update_layout()
    return fig

# Define app layout
app.layout = html.Div([
    html.H1(children="ASSIGNMENT - 3", style={"text-align":"center"}),

    # Graph-1
    html.Div([
        html.H3(children="Life Expectency vs GDP per capita graph", style={"margin-left":"4%"}),
        html.Div([
            html.Div([
                html.H4(children="View this graph for "),
                dcc.RadioItems(id="g1-data-radio-value", options=[{'label': i, 'value': i} for i in ['Continents', 'Cou
            ], style={"display":"flex", "flex-direction":"row", "align-items":"center"}
            ),
            html.Div([
                html.H4(children="Color code acc. to "),
                dcc.RadioItems(id="g1-color-code-radio-value", options=[{'label': i, 'value': i} for i in ['Continents'
            ], style={"display":"flex", "flex-direction":"row", "align-items":"center"}
            ),

            html.H4(id='g1-dropdown-header', children="", style={'display': 'none'}),
            dcc.Dropdown(
                id='g1-continent-dropdown',
                options=[{'label': i, 'value': i} for i in data['continent'].unique()], value="Asia",
                style={'display': 'none', 'margin-right':'55%'}
            )],
            style={"margin-left":"4%"}
        ),
        dcc.Graph(id='life-exp-vs-gdp'),
    ]),

    # Graph-2
    html.Div([
        html.H3(children="Life Expectency vs Continents", style={"margin-left":"4%"}),
        dcc.Graph(id='life-exp-vs-continent', figure=update_graph_continent()),
    ]),

    # Graph-3
    html.Div([
        html.H3(children="Life Expectency vs Population", style={"margin-left":"4%"}),
        html.Div([
            html.H4(children="View this graph for "),
            dcc.RadioItems(id="g3-radio-value", options=[{'label': i, 'value': i} for i in ['Continents', 'Countries']]
            html.H4(id='g3-dropdown-header', children="", style={'display': 'none'}),
            dcc.Dropdown(
                id='g3-continent-dropdown',
                options=[{'label': i, 'value': i} for i in data['continent'].unique()],
                style={'display': 'none'}
            ),
        ],
        style={"margin-left":"4%"}
        ),
        dcc.Graph(id='life-exp-vs-population'),
    ]),

    # Graph-4
    html.Div([
        html.H3(children="Countries' Evolution", style={"margin-left":"4%"}),
        html.Div([
                html.H4(id='g4-dropdown-header', children="View this graph for ", style={'margin-right':'4%'}),
                dcc.Dropdown(
                    id='g4-continent-dropdown',
                    options=[{'label': i, 'value': i} for i in data['continent'].unique()], value="Asia",
                    style={'margin-right':'4%'}
                ),
                dcc.RadioItems(id="g4-radio-value", options=[{'label': i, 'value': i} for i in ['Life Expectency', 'GDP
            ],
        style={"margin-left":"4%"}
        ),
        dcc.Graph(id='countries-evolution'),
    ]),

    # Graph-5
    html.Div([
        html.H3(children="The slope of the linear dependency between life expectancy and GDP per capita over time", sty
        dcc.Graph(id="gdp-linear-regression", figure=gdp_linear_regression())

    ]),

    # Graph-6
    html.Div([
        html.Div([
            html.H3(children="How GDP per capita changes over time", style={"margin-left":"4%"}),
            html.H3(children="GDP growth for continents", style={"margin-left":"4%"}),
            html.Div([
                dcc.RadioItems(id="g6-1-radio-value", options=[{'label': i, 'value': i} for i in ['linear', 'log']], va
            ],
            style={"margin-left":"4%"}
            ),
            dcc.Graph(id='gdp-over-time-1', hoverData={'points': [{'hovertext': 'Asia'}]}),
        ], style={"width": "49%"}),
        html.Div([
            html.Div([
                html.H4(id='g6-2-dropdown-header', children="Hover over a point in the continent scatter plot or select
                dcc.RadioItems(id="g6-2-radio-value", options=[{'label': i, 'value': i} for i in ['linear', 'log']], va
            ],
            style={"margin-left":"4%"}
            ),
            dcc.Graph(id='gdp-over-time-2'),
        ], style={"width": "49%", "margin-top":"5px"}),
    ], style={"display":"flex"}),

    # Graph-7
    html.Div([
        html.H3(children="World GDP and world life expectancy over time as a reference level.", style={"margin-left":"4
        html.Button(id="g7-button", children="Flip axes", style={"margin-left":"4%"}),
        dcc.RadioItems(id="g7-radio-value", options=[{'label': i, 'value': i} for i in ['linear', 'log']], value='linea
        dcc.Graph(id='world-gdp-life-exp'),
    ]),

    #Graph-8
    html.Div(
    [
        html.H4("Country's key performance analytics"),
        html.P("Select data on y-axis:"),
        dcc.Dropdown(
            id="y-axis",
            options=[
                {'label': 'Life Expectancy', 'value': 'lifeExp'},
                {'label': 'Population', 'value': 'pop'},
                {'label': 'GDP per Capita', 'value': 'gdpPercap'}
            ],
            value="gdpPercap",
        ),
        dcc.Graph(id="country-key-performance"),
    ]),

    #Graph-9
    html.Div([
        html.H4('Geographical representation of growth'),
        html.P("Select data on y-axis:"),
        dcc.Dropdown(
            id="datay",
            options=[
                {'label': 'Life Expectancy', 'value': 'lifeExp'},
                {'label': 'Population', 'value': 'pop'},
                {'label': 'GDP per Capita', 'value': 'gdpPercap'},
            ],
            value="gdpPercap",
        ),
        html.Div([
                html.H4(children="Color code acc. to "),
                dcc.RadioItems(id="g9-color-code-radio-value", options=[{'label': i, 'value': i} for i in ['default', '
            ], style={"display":"flex", "flex-direction":"row", "align-items":"center"}
        ),
        dcc.Graph(id="graph"),
    ]),
])

if __name__ == '__main__':
    app.run_server(debug=True, dev_tools_hot_reload=True)
```

ASSIGNMENT - 3

Life Expectency vs GDP per capita graph

View this graph for ◉ Continents ○ Countries

Color code acc. to ◉ Continents ○ Countries ○ Income Group ○ Life Expectancy



5. Create a short video (3-5min) to demo the use of your dashboard for accomplishing your exploration objective. You should upload the mp4 file to your OneDrive folder and share it with Aman and me. You can record the video using the Camera app on windows, or you can record it on Teams by creating a meeting with you as participant, share your demo screen in the meeting and record the demo. Any other way you find to record the demo is acceptable.

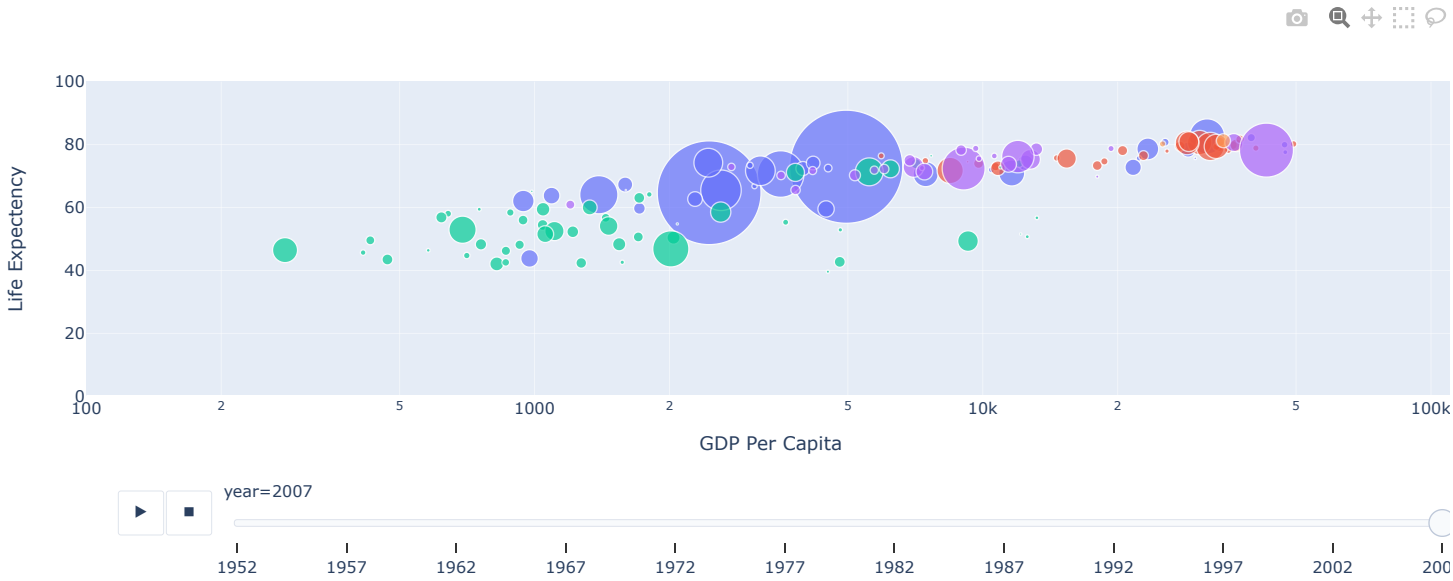
Answer: https://tinyurl.com/ypm9w4fa

ASSIGNMENT - 3

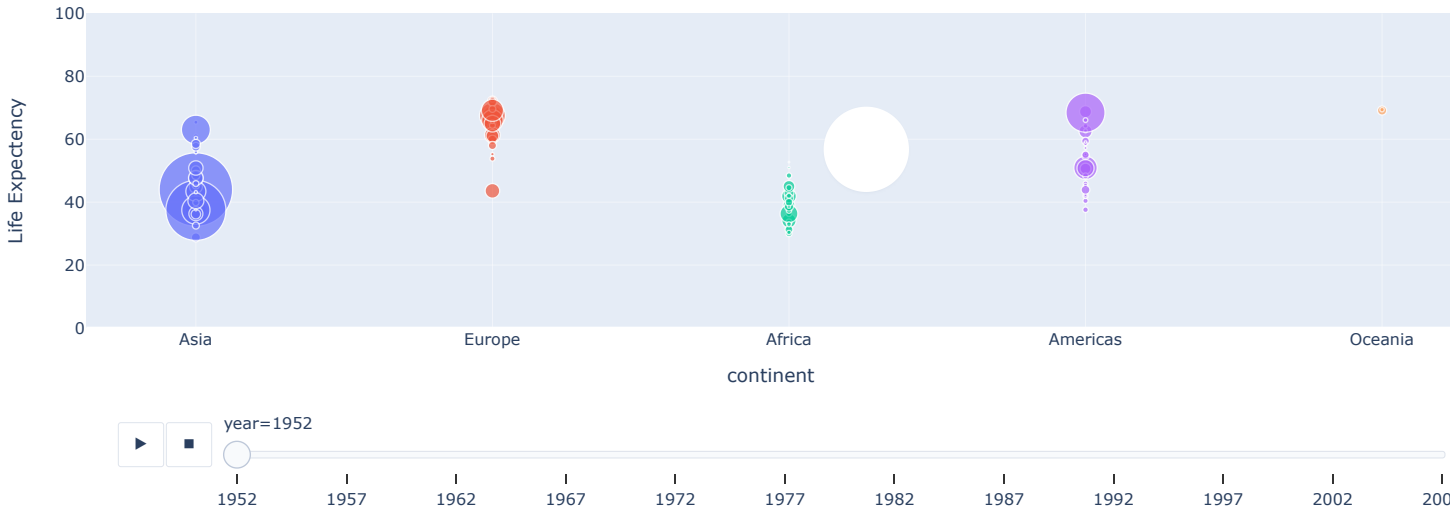
Life Expectency vs GDP per capita graph

View this graph for ◉ Continents ○ Countries

Color code acc. to ◉ Continents ○ Countries ○ Income Group ○ Life Expectancy



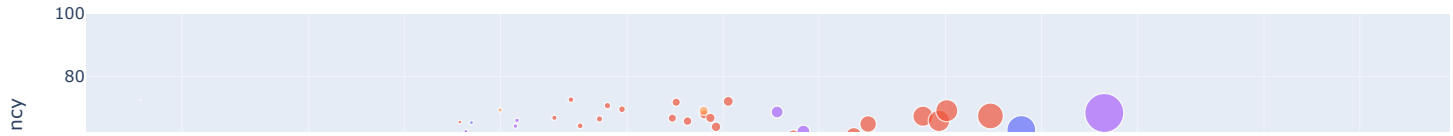
Life Expectency vs Continents

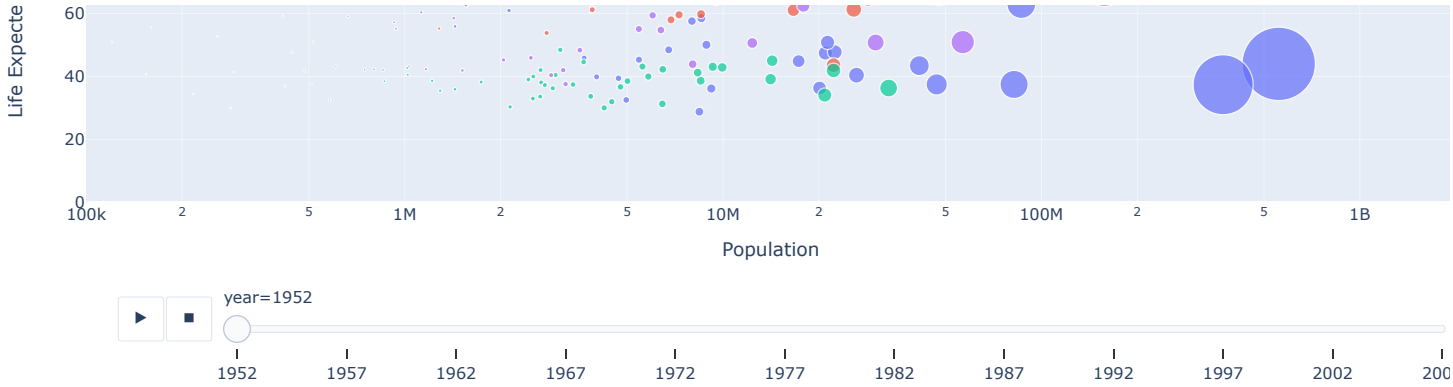


Life Expectency vs Population

View this graph for

◉ Continents ○ Countries



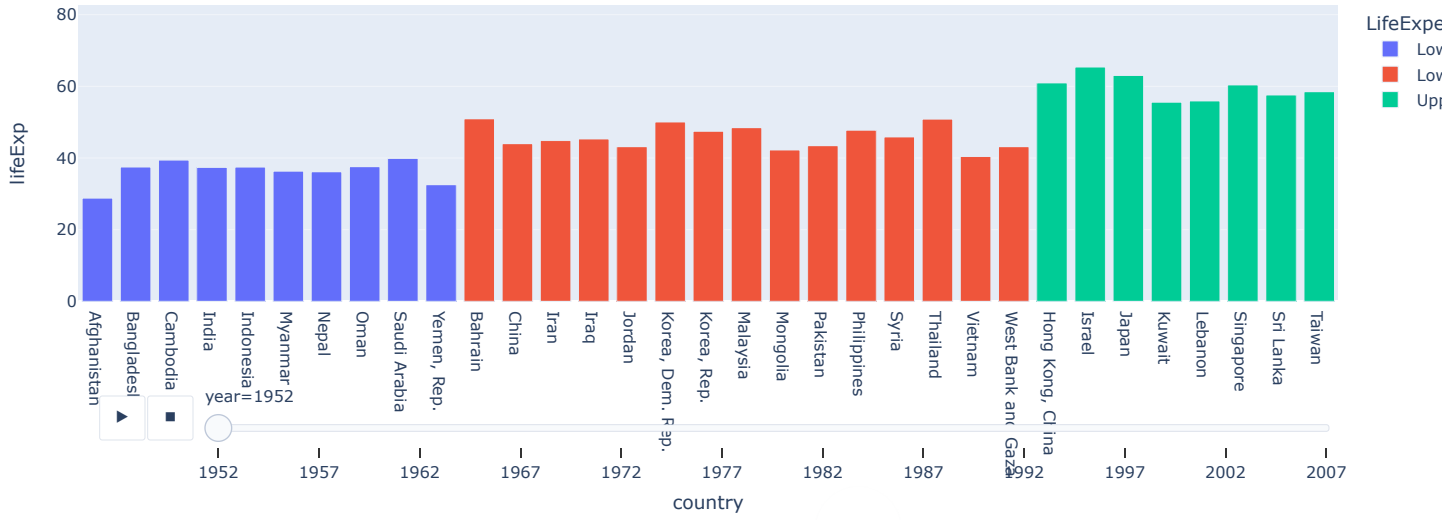


Countries' Evolution

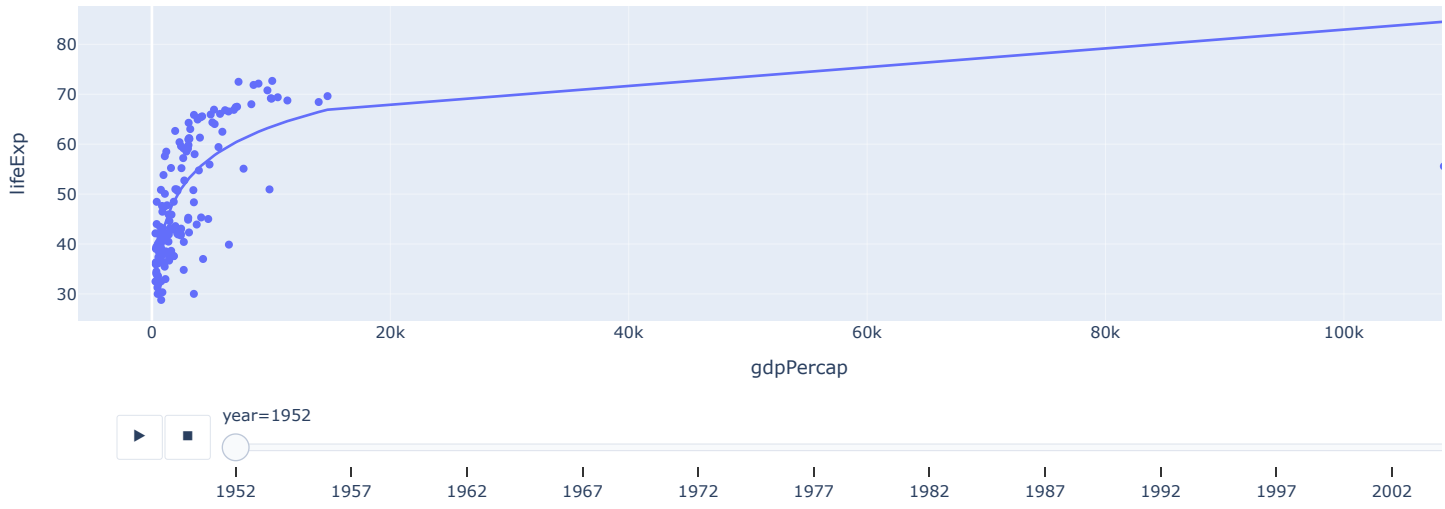
View this graph for

Asia

Life Expectency GDP Per Capita



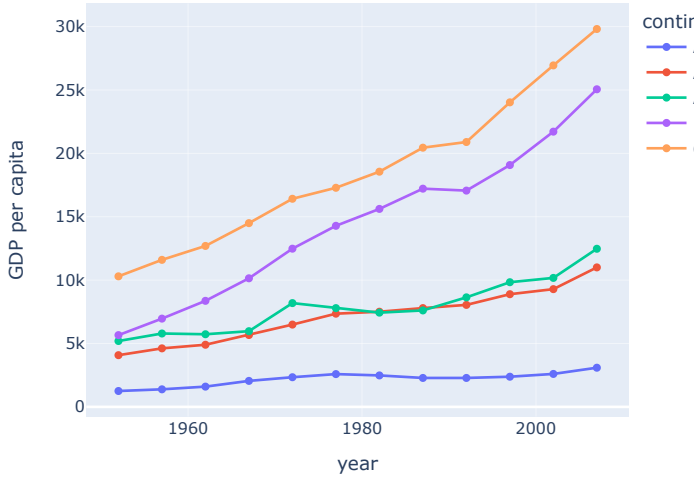
The slope of the linear dependency between life expectancy and GDP per ca... over time



How GDP per capita changes over time

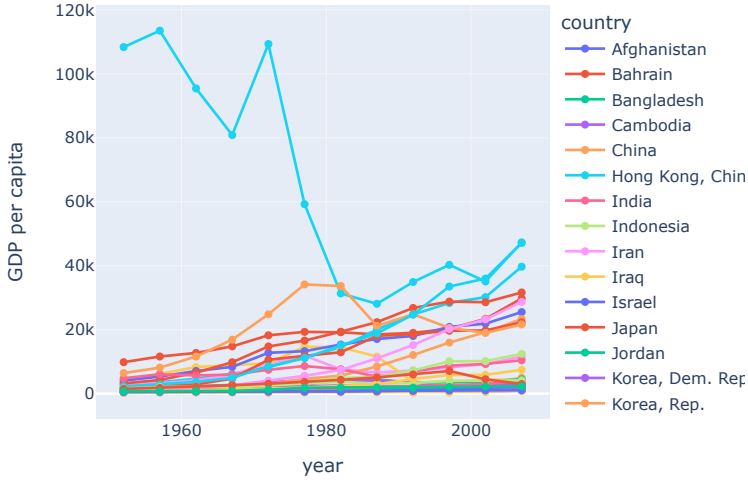
GDP growth for continents

linear log



Hover over a point in the continent scatter plot or select a continent to see all of its countries' growth over time

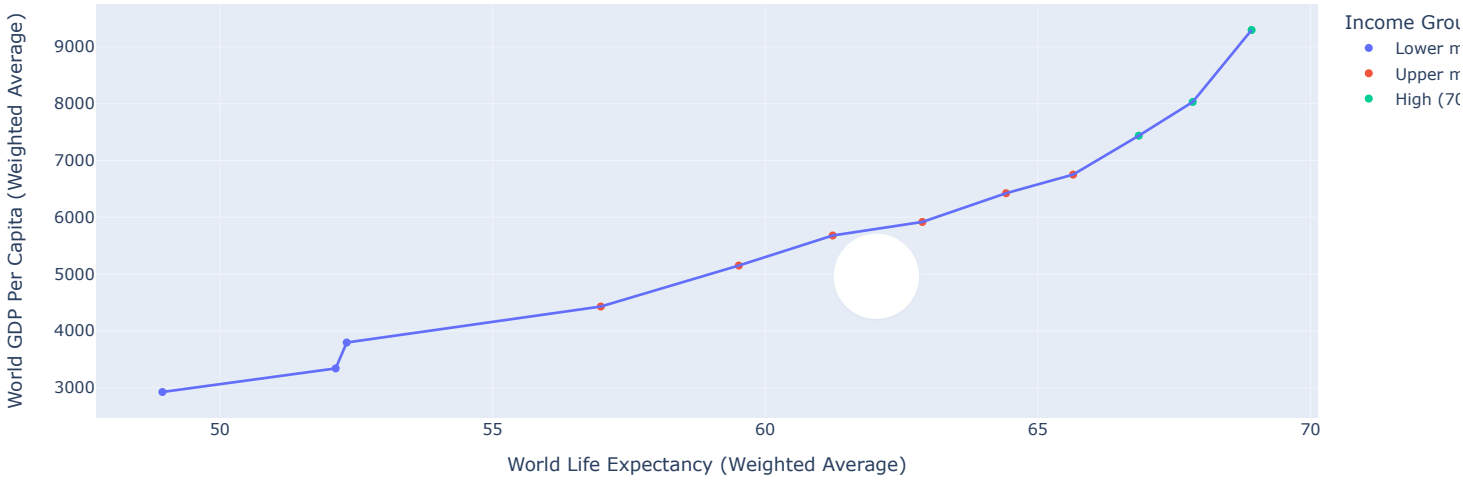
r log



World GDP and world life expectancy over time as a reference level.

Flip axes

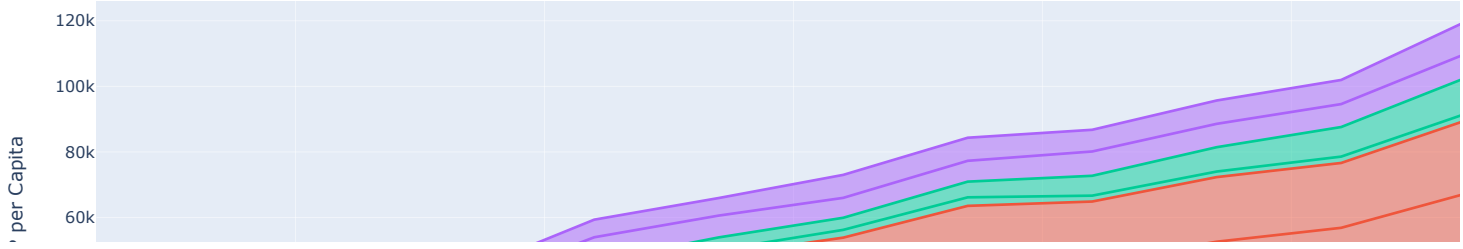
linear log

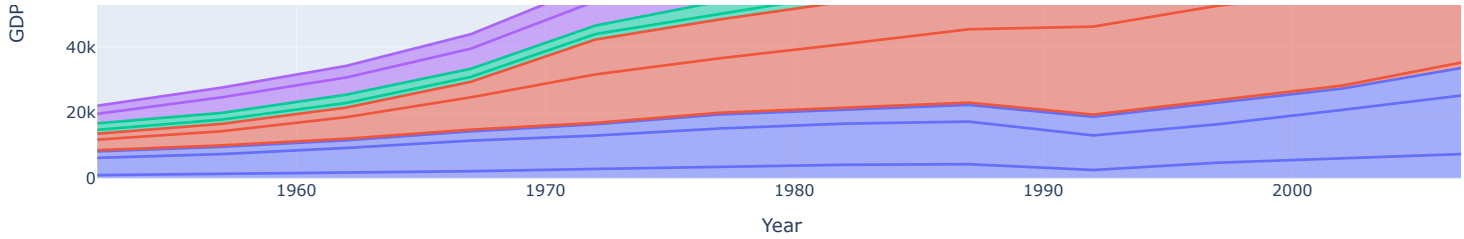


Country's key performance analytics

Select data on y-axis:

GDP per Capita





Geographical representation of growth

Select data on y-axis:

GDP per Capita

Color code acc. to default Income Group Life Expectancy