Python初級數據分析員證書

（六）數據分析 及可視化專案

13. 數據分析專案 Demo 19 – Life Expectancy

Review

- Statistics
- Hypothesis testing
- Algebra
- Linear regression
- Propositional logic
- Python
- R
- SQL

- Pandas, NumPy, SciPy
- Data Visualization, Matplotlib, Seaborn, Plotly
- Dashboard Visualization, Business Intelligence
- Storytelling

Chapter Summary

- Scenario
- Data Import
- Data Wrangling
- EDA
- Linear Regression Model
- Prediction

Scenario

The Global Health Observatory (GHO) data repository under World Health Organization (WHO) keeps track of the health status as well as many other related factors for all countries The data-sets are made available to public for the purpose of health data analysis. The data-set related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website and its corresponding economic data was collected from United Nation website.世界衛生組織（WHO）下屬的全球健康觀察站（GHO）數據存儲庫跟蹤所有國家的健康情況以及許多其他相關因素。數據集向公眾提供，用於健康數據分析。與 193 個國家 / 地區的預期壽命、健康因素相關的數據集是從同一個 WHO 數據儲存庫網站收集的，其相應的經濟數據是從聯合國網站收集的。

Data import

```
1  import pandas as pd
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import plotly.express as px
5  import seaborn as sns
6  import datetime as dt
7  from plotly.offline import init_notebook_mode
8  init_notebook_mode(connected=True)
9  import warnings
10 #warnings.filterwarnings('ignore')
11
12 pd.set_option('display.max_columns',None)
```

```
1  df=pd.read_csv('LifeExpectancy.csv')
2  df.head(5)
```

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	19.1	83	6.0	8.16
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	18.6	86	58.0	8.18
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	18.1	89	62.0	8.13
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	17.6	93	67.0	8.52
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	17.2	97	68.0	7.87

Overview all columns

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Country                          2938 non-null   object 
 1   Year                             2938 non-null   int64  
 2   Status                           2938 non-null   object 
 3   Life expectancy                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   infant deaths                    2938 non-null   int64  
 6   Alcohol                          2744 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64  
 10   BMI                             2904 non-null   float64
 11  under-five deaths                2938 non-null   int64  
 12  Polio                            2919 non-null   float64
 13  Total expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15   HIV/AIDS                        2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18   thinness  1-19 years            2904 non-null   float64
 19   thinness 5-9 years             2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

Check NaN and Null

```
1 df.isnull().sum()
```

```
Country                            0
Year                               0
Status                             0
Life expectancy                   10
Adult Mortality                   10
infant deaths                      0
Alcohol                          194
percentage expenditure             0
Hepatitis B                      553
Measles                            0
 BMI                              34
under-five deaths                  0
Polio                             19
Total expenditure                226
Diphtheria                        19
 HIV/AIDS                          0
GDP                              448
Population                       652
 thinness  1-19 years             34
 thinness 5-9 years               34
Income composition of resources  167
Schooling                        163
dtype: int64
```

Replacing the Null Values with mean values of the data

```
# Replacing the Null Values with mean values of the data
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan,strategy='mean',fill_value=None)
df['Life expectancy ']=imputer.fit_transform(df[['Life expectancy ']])
df['Adult Mortality']=imputer.fit_transform(df[['Adult Mortality']])
df['Alcohol']=imputer.fit_transform(df[['Alcohol']])
df['Hepatitis B']=imputer.fit_transform(df[['Hepatitis B']])
df[' BMI ']=imputer.fit_transform(df[[' BMI ']])
df['Polio']=imputer.fit_transform(df[['Polio']])
df['Total expenditure']=imputer.fit_transform(df[['Total expenditure']])
df['Diphtheria ']=imputer.fit_transform(df[['Diphtheria ']])
df['GDP']=imputer.fit_transform(df[['GDP']])
df['Population']=imputer.fit_transform(df[['Population']])
df[' thinness  1-19 years']=imputer.fit_transform(df[[' thinness  1-19 years']])
df[' thinness 5-9 years']=imputer.fit_transform(df[[' thinness 5-9 years']])
df['Income composition of resources']=imputer.fit_transform(df[['Income composition of resources']])
df['Schooling']=imputer.fit_transform(df[['Schooling']])

```

Data Wrangling

```
1 df.isnull().sum()
```

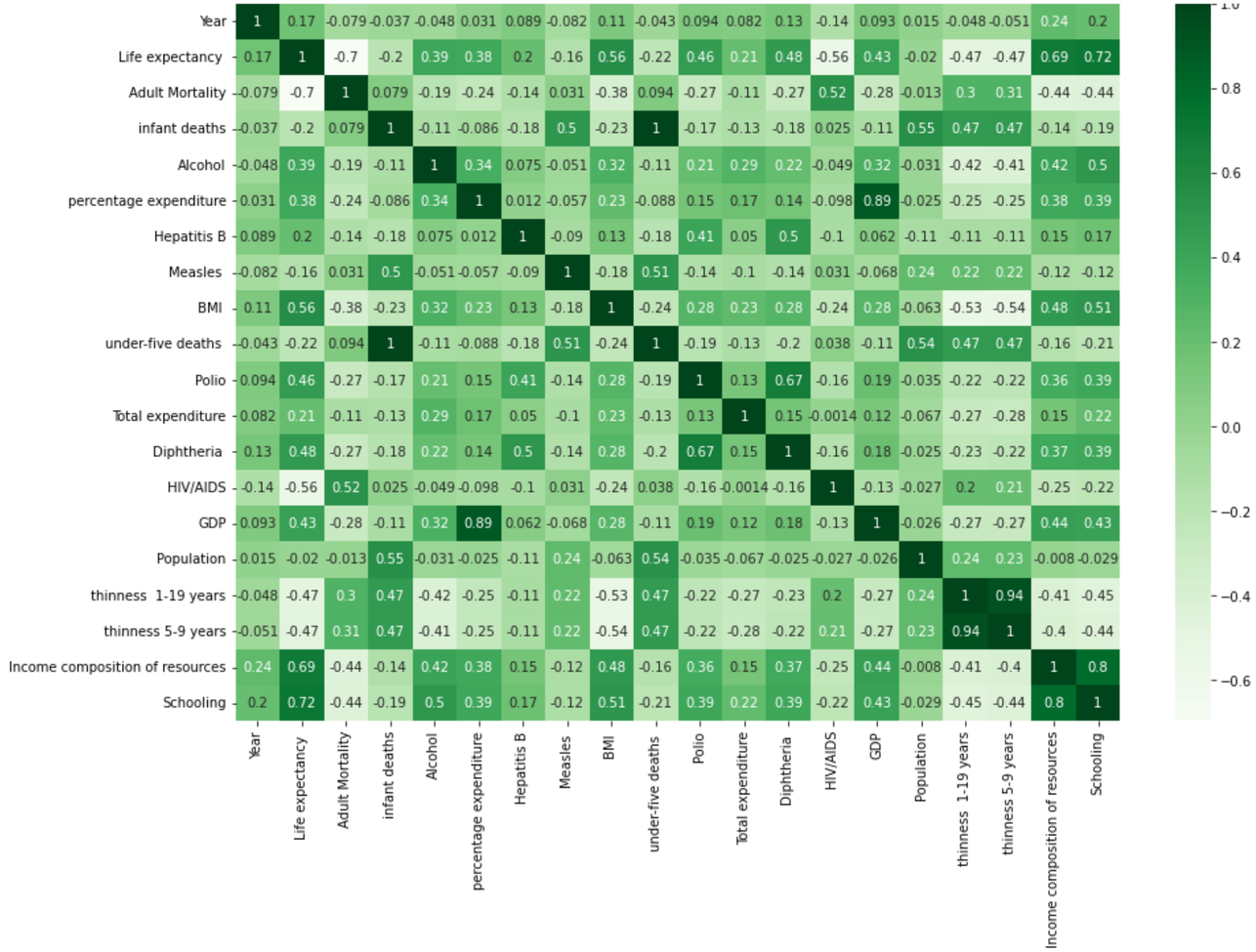
```
Country                          0
Year                             0
Status                           0
Life expectancy                  0
Adult Mortality                  0
infant deaths                    0
Alcohol                          0
percentage expenditure           0
Hepatitis B                      0
Measles                          0
 BMI                             0
under-five deaths                0
Polio                            0
Total expenditure                0
Diphtheria                       0
 HIV/AIDS                        0
GDP                              0
Population                       0
 thinness  1-19 years            0
 thinness 5-9 years              0
Income composition of resources  0
Schooling                        0
dtype: int64
```

```
1 df.describe()
```

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI
count	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000	2938.000000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461	2419.592240	38.321247
std	4.613841	9.507640	124.080302	117.926501	3.916288	1987.914858	22.586855	11467.272489	19.927677
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000
25%	2004.000000	63.200000	74.000000	0.000000	1.092500	4.685343	80.940461	0.000000	19.400000
50%	2008.000000	72.000000	144.000000	3.000000	4.160000	64.912906	87.000000	17.000000	43.000000
75%	2012.000000	75.600000	227.000000	22.000000	7.390000	441.534144	96.000000	360.250000	56.100000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000

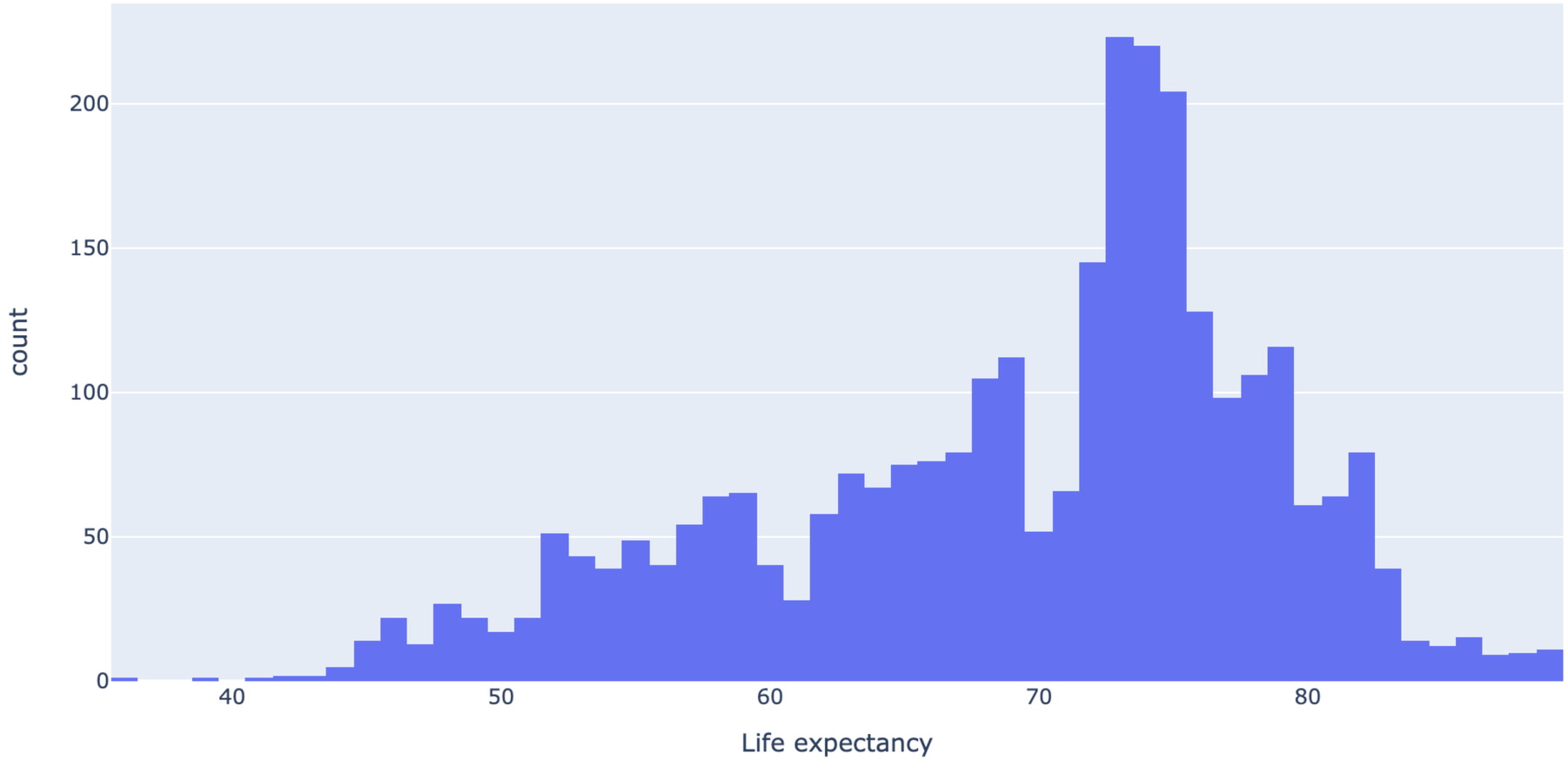
Heatmap (optout status & country)

```
1 plt.figure(figsize=(15,10))
2 sns.heatmap(df.drop(['Status','Country'],axis=1).corr(),annot=True,cmap='Greens')
3 plt.show()
```



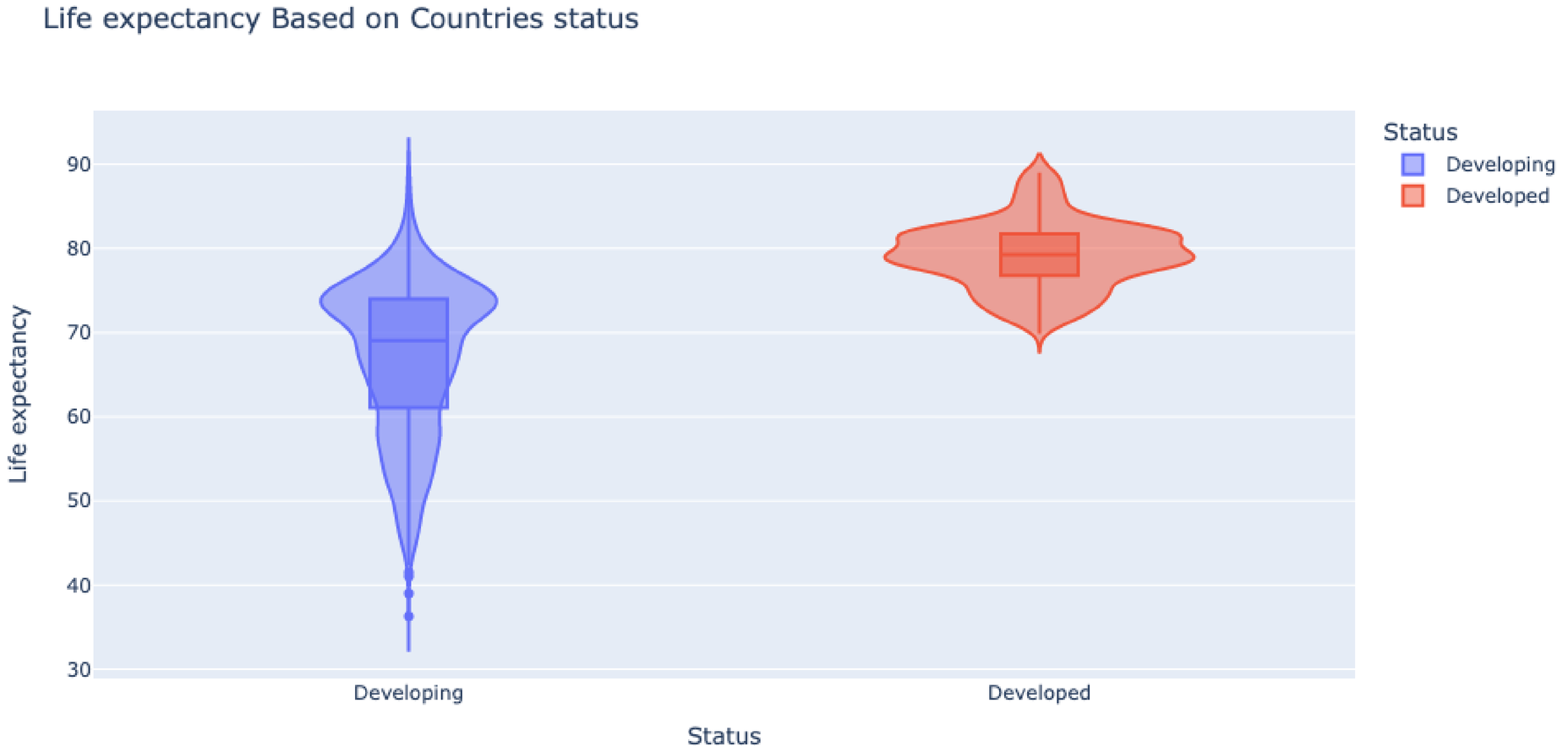
Life expectancy

```
1  fig=px.histogram(df,x='Life expectancy ')
2  fig.show()
```



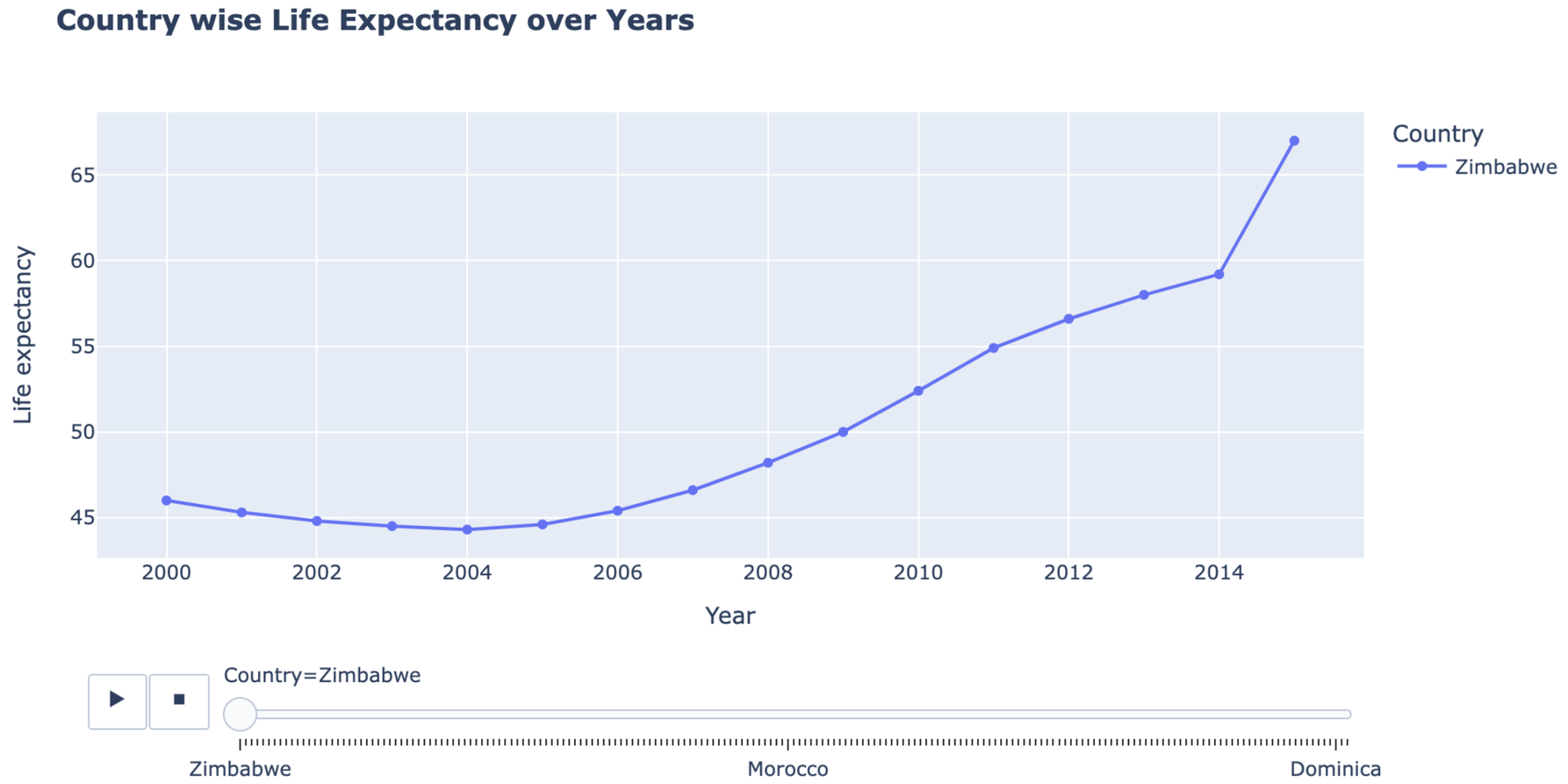
DEVELOPED COUNTRIES HAS MAXIMUM LIFE EXPENTANCY

```
1  fig=px.violin(df,x='Status',y='Life expectancy ',color='Status',
2                box=True,title='Life expectancy Based on Countries status')
3  fig.show()
```



Country wise Life Expectancy over Years

```
fig=px.line(df.sort_values(by='Year'),x='Year',y='Life expectancy ',
            animation_frame='Country',animation_group='Year',color='Country',
            markers=True,title='<b> Country wise Life Expectancy over Years')
fig.show()
```

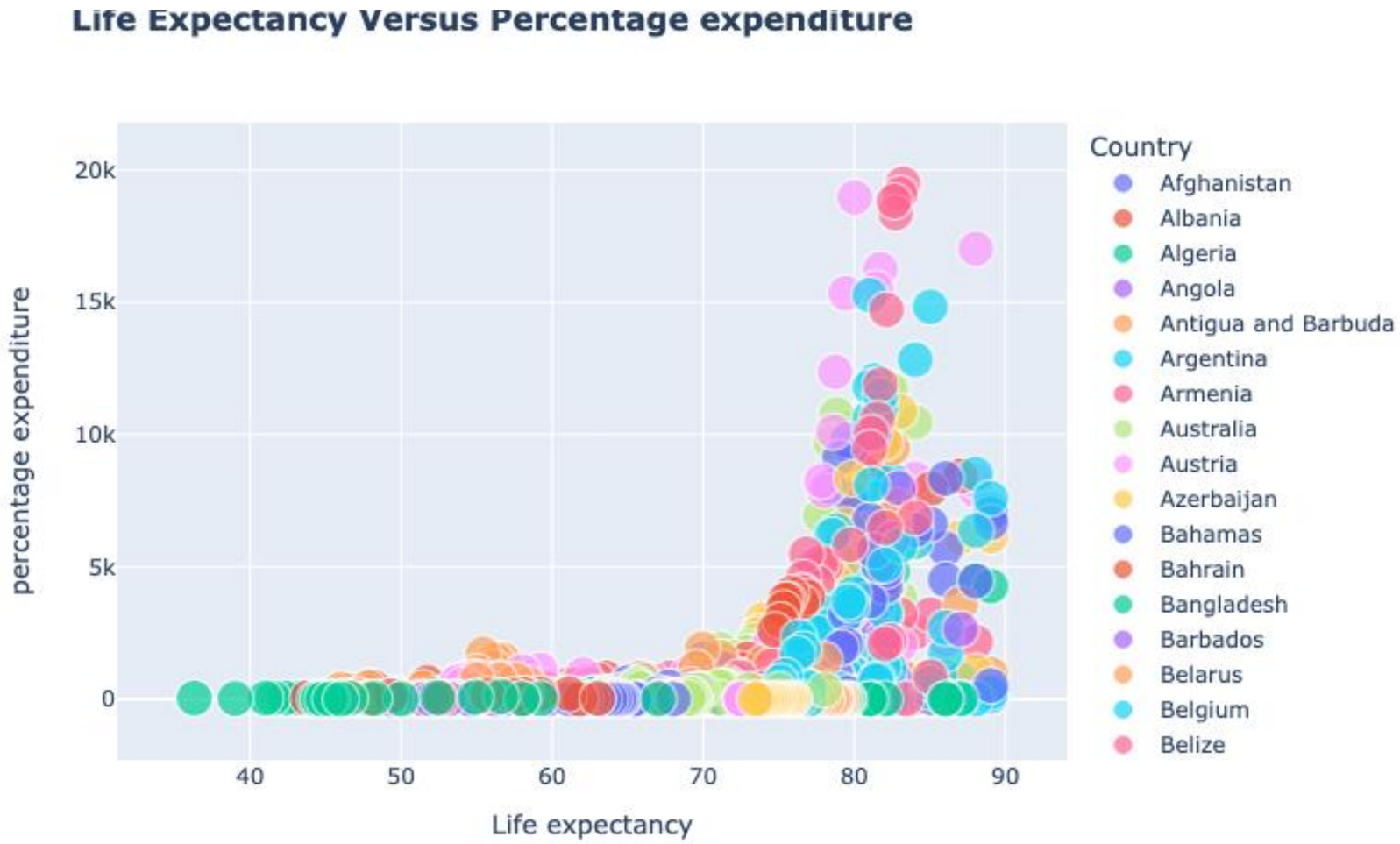


Life Expectancy Versus Adult Mortality

```
1  px.scatter(df,y='Adult Mortality',x='Life expectancy ',color='Country',
2             size='Life expectancy ', opacity=0.6,
3             title='<b> Life Expectancy Versus Adult Mortality')
```

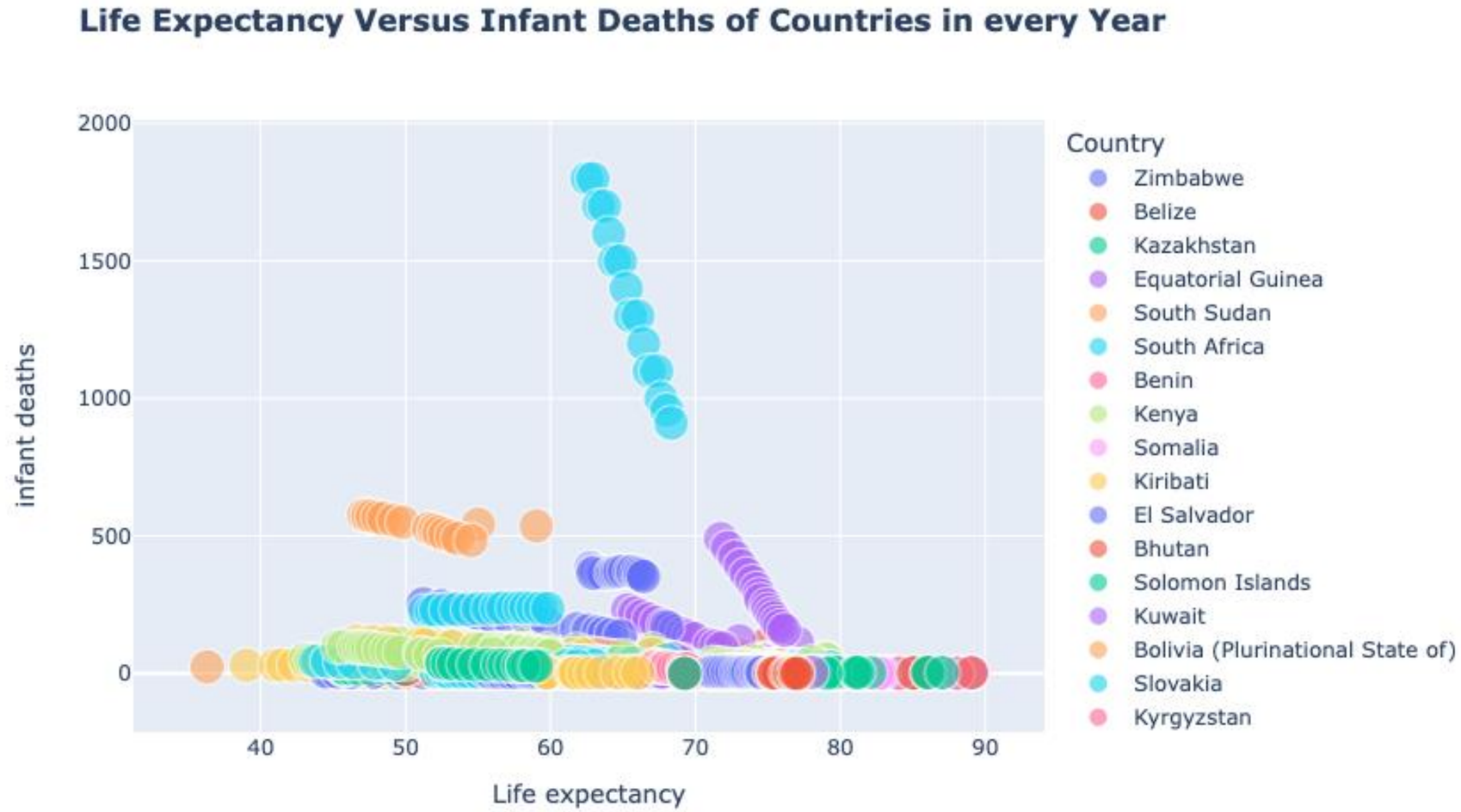
Life Expectancy Versus Percentage expenditure

```
px.scatter(df,x='Life expectancy ',y='percentage expenditure',color='Country',
           size='Year',title='<b> Life Expectancy Versus Percentage expenditure')
```



Life Expectancy Versus Infant Deaths of Countries in every Year

```
px.scatter(df.sort_values(by='Year'), y='infant deaths', x='Life expectancy ',
           size='Year',color='Country', opacity=0.6,
           title='<b>Life Expectancy Versus Infant Deaths of Countries in every Year')
```

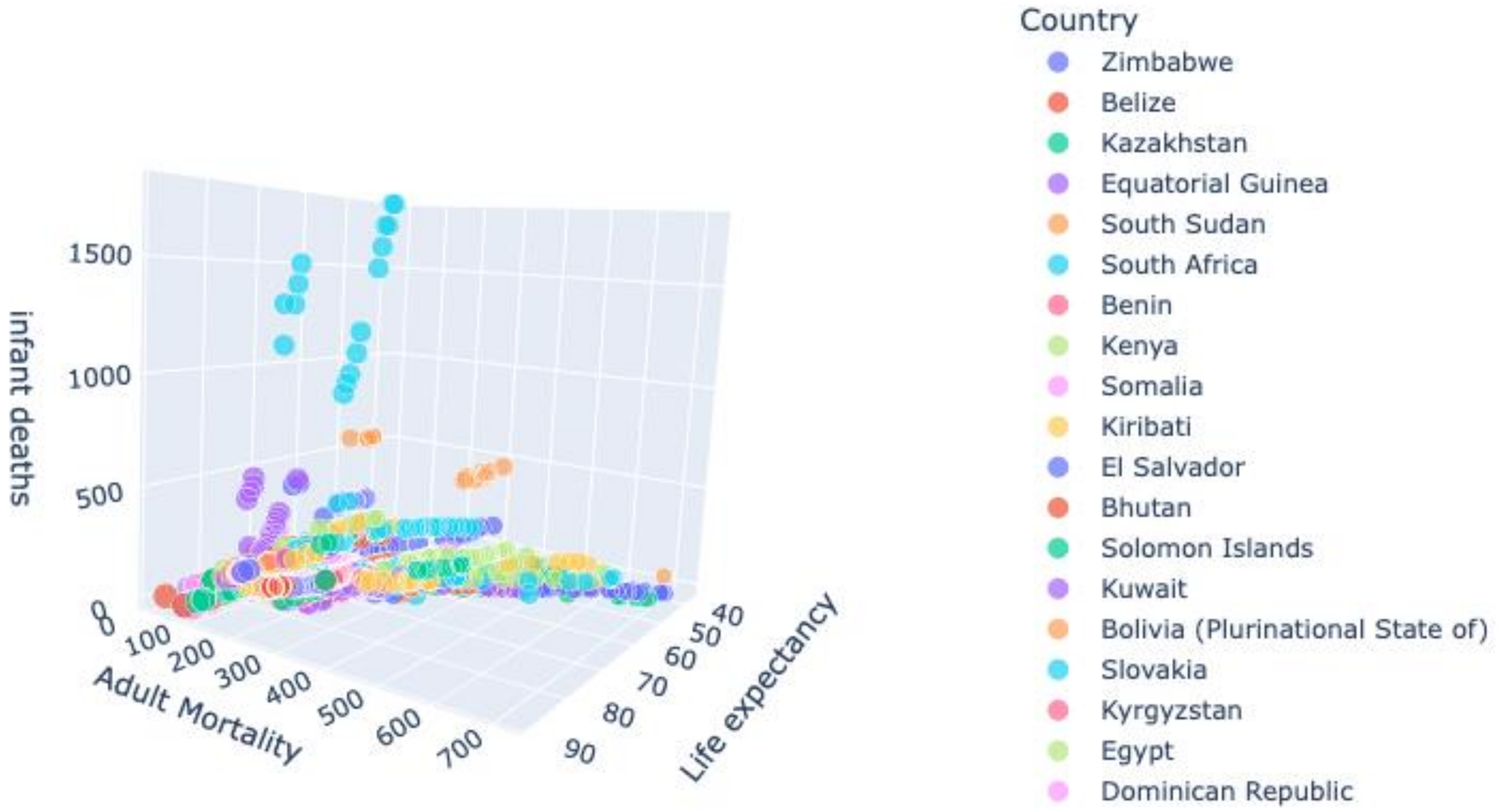


Life expectancy vs Schooling vs Total expenditure

```
px.scatter_3d(df.sort_values(by='Year'),y='Schooling',x='Life expectancy ',
              z='Total expenditure',color='Country',size='Total expenditure')
```

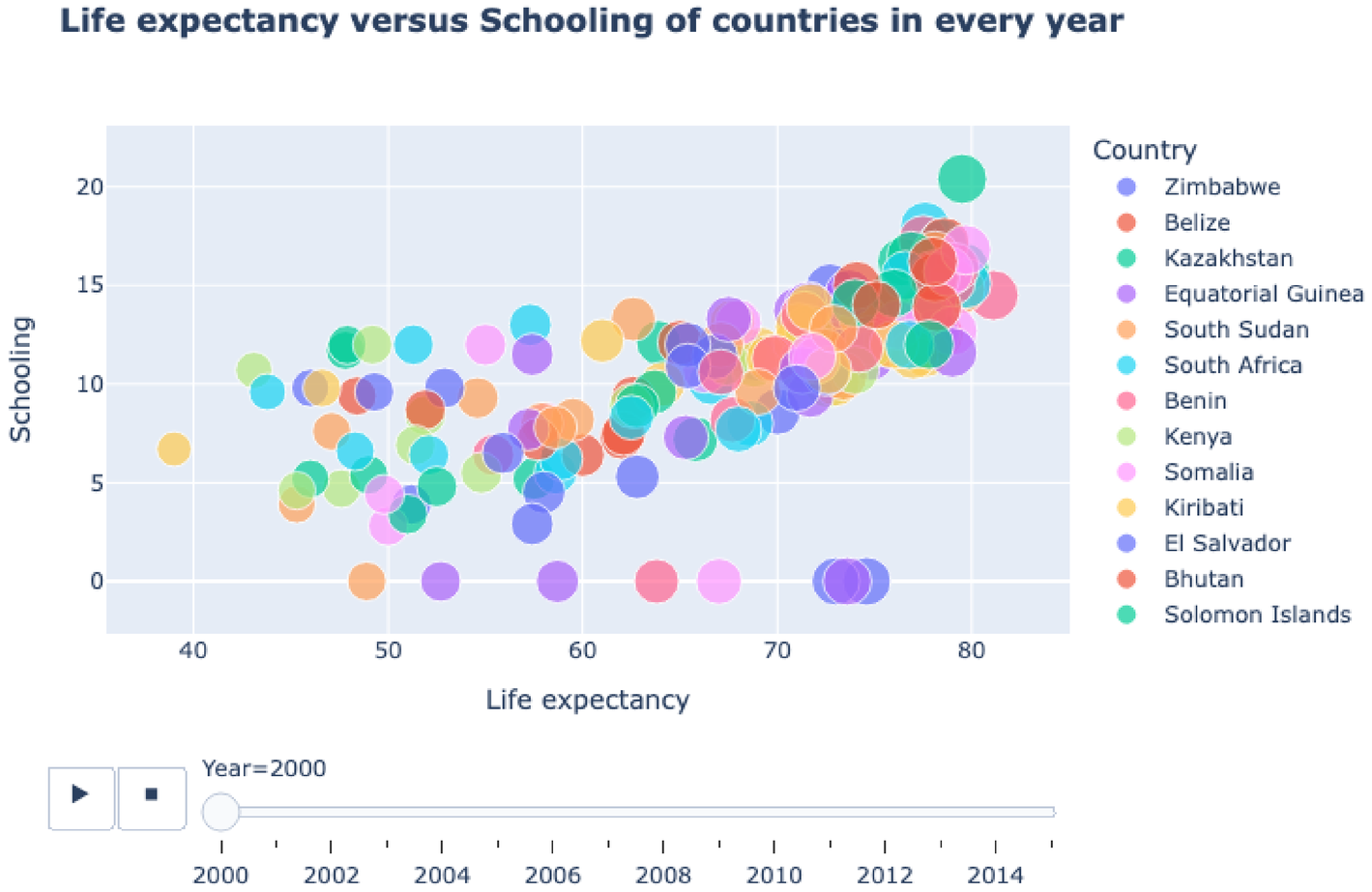
Life expectancy vs Adult Mortality vs infant deaths

```
1  px.scatter_3d(df.sort_values(by='Year'),y='Adult Mortality',
2                x='Life expectancy ',z='infant deaths',
3                size='Life expectancy ', color='Country')
```



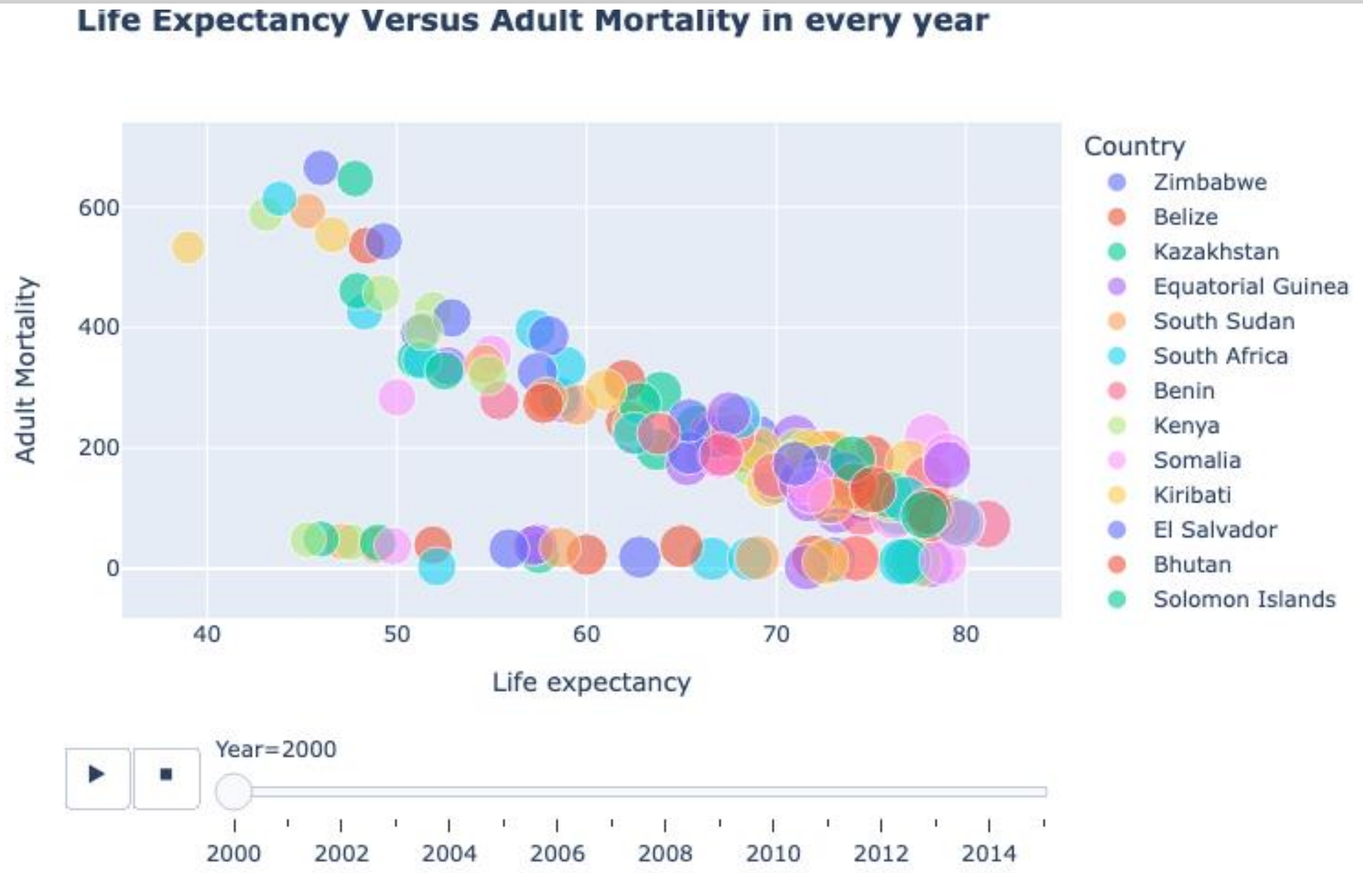
Life expectancy versus Schooling of countries in every year

```
1  px.scatter(df.sort_values(by='Year'),y='Schooling',x='Life expectancy ',
2             animation_frame='Year',animation_group='Country',
3             color='Country',size='Life expectancy ',
4             title='<b> Life expectancy versus Schooling of countries in every year')
```



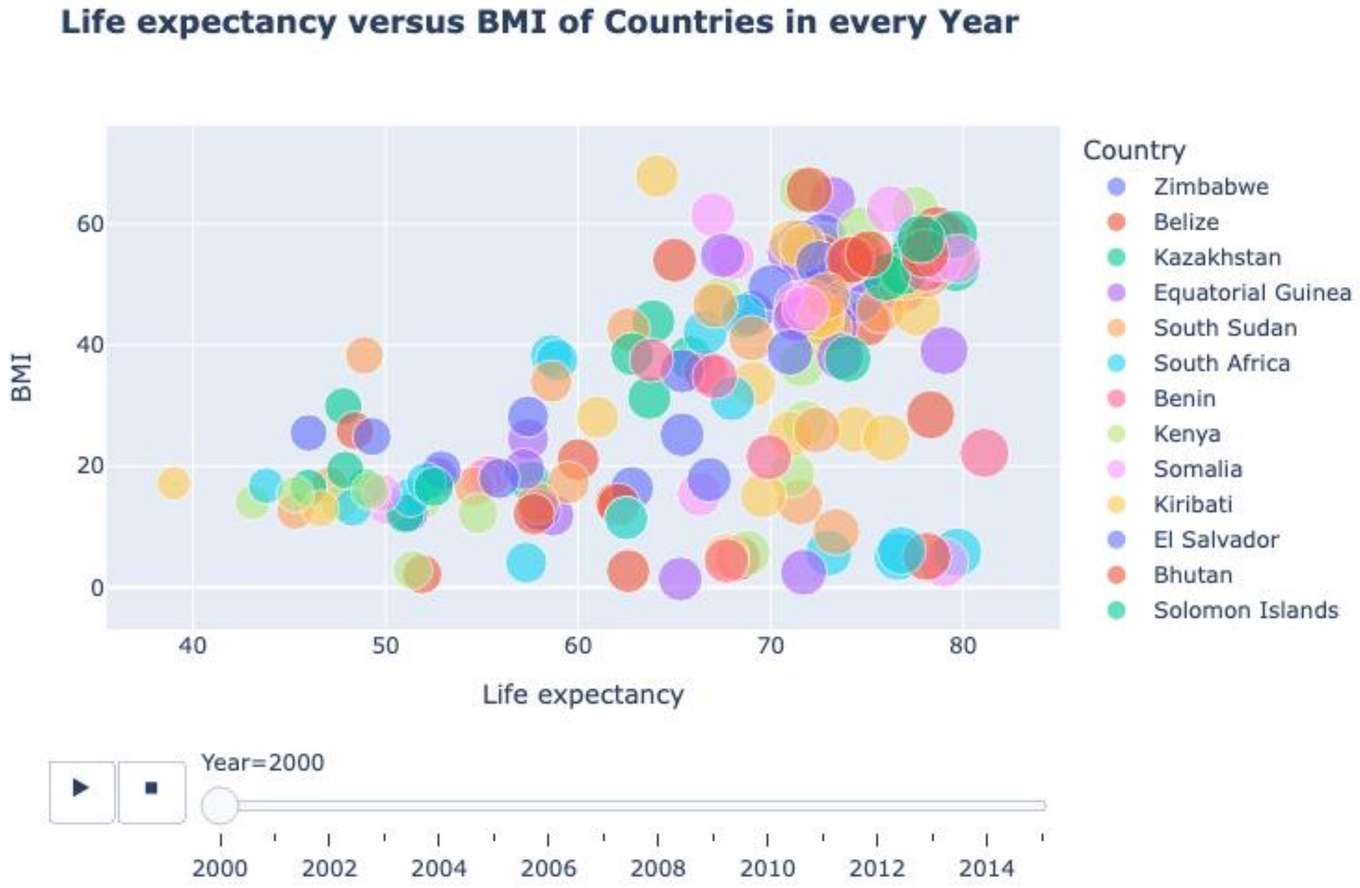
Life Expectancy Versus Adult Mortality in every year

```
px.scatter(df.sort_values(by='Year'),y='Adult Mortality',x='Life expectancy ',
           animation_frame='Year',animation_group='Country',color='Country',
           size='Life expectancy ',opacity=0.6,
           title='<b> Life Expectancy Versus Adult Mortality in every year')
```



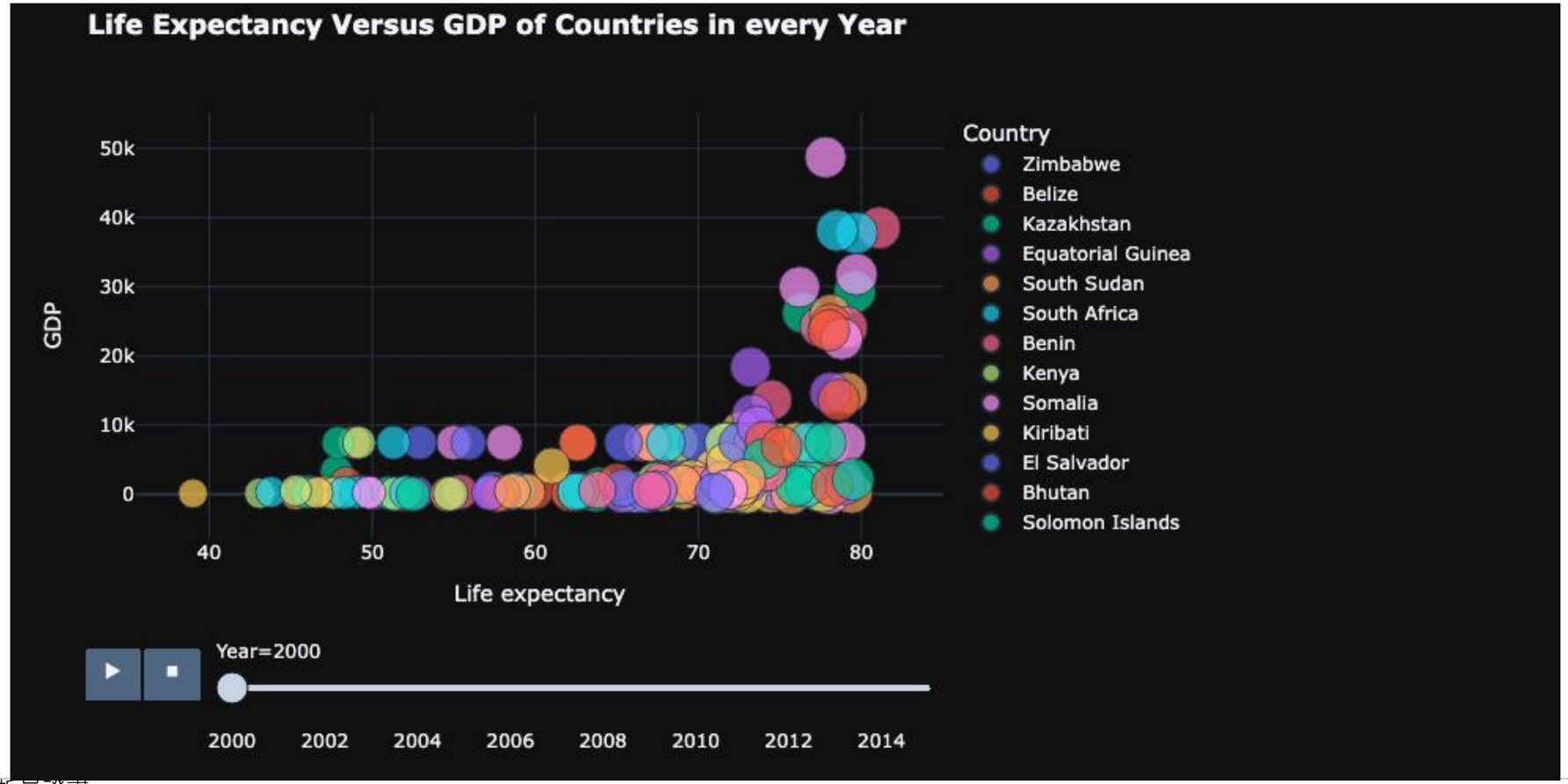
Life expectancy versus BMI of Countries in every Year

```
px.scatter(df.sort_values(by='Year'),y=' BMI ',x='Life expectancy ',
           animation_frame='Year',animation_group='Country',
           color='Country',size='Life expectancy ',opacity=0.6,
           title='<b> Life expectancy versus BMI of Countries in every Year')
```



Life Expectancy Versus GDP of Countries in every Year

```
px.scatter(df.sort_values(by='Year'),y='GDP',x='Life expectancy ',
           animation_frame='Year',animation_group='Country',
           template='plotly_dark',color='Country',size='Life expectancy ',
           title='<b>Life Expectancy Versus GDP of Countries in every Year')
```



Model bulding

```
1 df.describe(include='O')
```

	Country	Status
count	2938	2938
unique	193	2
top	Afghanistan	Developing
freq	16	2426

Label encoding the categoric columns to number.

```
1 df['Status'].unique()
```

```
array(['Developing', 'Developed'], dtype=object)
```

```
1 df['Country'].unique()
```

```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola',
       'Antigua and Barbuda', 'Argentina', 'Armenia', 'Australia',
       'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain', 'Bangladesh',
       'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bhutan',
       'Bolivia (Plurinational State of)', 'Bosnia and Herzegovina',
       'Botswana', 'Brazil', 'Brunei Darussalam', 'Bulgaria',
       'Burkina Faso', 'Burundi', "Côte d'Ivoire", 'Cabo Verde',
       'Cambodia', 'Cameroon', 'Canada', 'Central African Republic',
```

Label encoding

```
1  from sklearn.preprocessing import LabelEncoder
2  le = LabelEncoder()
3  # Country
4  le.fit(df.Country.drop_duplicates())
5  df.Country = le.transform(df.Country)
6  # Status
7  le.fit(df.Status.drop_duplicates())
8  df.Status = le.transform(df.Status)
```

Label encoding

```
1 df['Country'].unique()
```

```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
        13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
        26,  44,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,
        38,  39,  40,  41,  42,  43,  45,  46,  47,  48,  49,  50,  51,
        52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
        78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
        91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
       117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
       130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
       143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
       156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
       169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181,
       182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192])
```

```
1 df['Status'].unique()
```

```
array([1, 0])
```

Fit into model

```
1  from sklearn.linear_model import LinearRegression
2
3  linear_reg = LinearRegression()
4  x = df.drop('Life expectancy ',axis=1)
5  y = df['Life expectancy '].copy()
6
7  linear_reg.fit(x,y)
```

▼ LinearRegression

LinearRegression()

Predict with model

```
1 y_head = linear_reg.predict(x)
```

```
1 from sklearn import metrics
2 print("Mean Absolute Error: ", metrics.mean_absolute_error(y,y_head))
3 print("Mean Squared Error: ", metrics.mean_squared_error(y,y_head))
4 print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(y, y_head)))
```

```
Mean Absolute Error:  3.0085935023731882
Mean Squared Error:  16.22495639266941
Root Mean Squared Error:  4.028021399231813
```

Result

```
1 from sklearn.metrics import mean_squared_error, r2_score
2 print("Estimated accuracy: ", r2_score(y, linear_reg.predict(x)))
```

```
Estimated accuracy:  0.8204497893481657
```

Predict a single data

To predict new outcome, Life expectancy. We shall prepare df with the following columns, or direct key in the variable in numpy array.

```
1 x.columns
```

```
Index(['Country', 'Year', 'Status', 'Adult Mortality', 'infant deaths',
       'Alcohol', 'percentage expenditure', 'Hepatitis B', 'Measles ', ' BMI ',
       'under-five deaths ', 'Polio', 'Total expenditure', 'Diphtheria ',
       ' HIV/AIDS', 'GDP', 'Population', ' thinness  1-19 years',
       ' thinness 5-9 years', 'Income composition of resources', 'Schooling'],
      dtype='object')
```

```
1 x.iloc[200]
```

```
Country                          1.200000e+01
Year                             2.007000e+03
Status                           1.000000e+00
Adult Mortality                  1.510000e+02
infant deaths                    1.540000e+02
Alcohol                          1.000000e-02
percentage expenditure           4.636537e+01
Hepatitis B                      9.500000e+01
Measles                          2.924000e+03
 BMI                             1.350000e+01
under-five deaths                2.010000e+02
Polio                            9.600000e+01
Total expenditure                2.800000e+00
Diphtheria                       9.400000e+01
 HIV/AIDS                        1.000000e-01
GDP                              5.416515e+02
Population                       1.471392e+08
 thinness  1-19 years            1.950000e+01
 thinness 5-9 years              2.100000e+00
Income composition of resources  5.130000e-01
Schooling                        8.600000e+00
Name: 200, dtype: float64
```

Let’s use the example from index 200

Predict

The estimated Life expectancy is age 65.

```
1  linear_reg.predict(x.loc[x.index==200])
```

```
array([64.92600832])
```

Chapter Wrap UP

Inspiration

The data-set aims to answer the following key questions:

- Does various predicting factors which has been chosen initially really affect the Life expectancy?最初選擇的各種預測因素真的會影響預期壽命嗎？
- What are the predicting variables actually affecting the life expectancy?實際影響預期壽命的預測變數是什麼？
- Should a country having a lower life expectancy value(<65) increase its healthcare expenditure in order to improve its average lifespan?預期壽命較低（<65）的國家是否應該增加其醫療保健支出以提高其平均壽命？
- How does Infant and Adult mortality rates affect life expectancy?嬰兒和成人死亡率如何影響預期壽命？

Chapter Wrap UP

- Does Life Expectancy has positive or negative correlation with eating habits, lifestyle, exercise, smoking, drinking alcohol etc. Life Expectancy 與飲食習慣、生活方式、運動、吸煙、飲酒等。
- What is the impact of schooling on the lifespan of humans?
  學校教育對人類的壽命有什麼影響？
- Does Life Expectancy have positive or negative relationship with drinking alcohol? Life Expectancy 與飲酒有關？
- Do densely populated countries tend to have lower life expectancy?
  人口稠密的國家的預期壽命往往較低嗎？
- What is the impact of Immunization coverage on life Expectancy?
  免疫接種覆蓋率對預期壽命有什麼影響？

Reference & Resources

Sklearn Website:

https://scikit-learn.org/

Plotly Graph Objects:

https://plotly.com/python/graph-objects/

Seaborn:

https://seaborn.pydata.org/examples/index.html

Matplotlib:

https://matplotlib.org/