# LIFE EXPECTANCY ANALYSIS WITH PYTHON

**Name:** Gokulakrishnan K
**Branch:** B.Tech Artificial Intelligence and Data Science
**College:** B.S Abdur Rahman Crescent Institute Of Science And Technology

# Data Science Project on Life Expectancy Analysis

## INTRODUCTION:

Life expectancy serves as an essential metric to understand a country's overall health and wellbeing. This report aims to analyze the factors influencing life expectancy across various countries and to develop a predictive model to estimate life expectancy based on these factors. Life expectancy refers to the number of years a person is expected to live based on the statistical average. It depends on the geographical context of the area. Before the modernization of the world, life expectancy was around 30 years in all parts of the world. Life expectancy increased at the beginning of the 19th century but until there are the same countries while it remains low in the rest of the world.

## Life Expectancy Analysis with Python

Now let’s get started with the task of Life Expectancy Analysis with Python. I will start this task by importing the necessary Python libraries and the dataset:

```
import pandas as pd
from pandas import DataFrame
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from matplotlib import rcParams
import plotly.graph_objects as go
import plotly.express as px
from plotly.colors import n_colors
import numpy as np
import seaborn as sns
import pandas_profiling
%matplotlib inline
from matplotlib import rc
import scipy.stats
from scipy.stats.mstats import winsorize
life_expectancy = pd.read_csv("Life Expectancy Data.csv") #reading the file
life_expectancy.head()
```

```
life_expectancy.head()
```

| | Country | Year | Status | Life expectancy | Adult Mortality | infant deaths | Alcohol | percentage expenditure | Hepatitis B | Measles | ... | Polio |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Afghanistan | 2015 | Developing | 65.0 | 263.0 | 62 | 0.01 | 71.279624 | 65.0 | 1154 | ... | 6.0 |
| 1 | Afghanistan | 2014 | Developing | 59.9 | 271.0 | 64 | 0.01 | 73.523582 | 62.0 | 492 | ... | 58.0 |
| 2 | Afghanistan | 2013 | Developing | 59.9 | 268.0 | 66 | 0.01 | 73.219243 | 64.0 | 430 | ... | 62.0 |
| 3 | Afghanistan | 2012 | Developing | 59.5 | 272.0 | 69 | 0.01 | 78.184215 | 67.0 | 2787 | ... | 67.0 |
| 4 | Afghanistan | 2011 | Developing | 59.2 | 275.0 | 71 | 0.01 | 7.097109 | 68.0 | 3013 | ... | 68.0 |

5 rows × 22 columns

The dataset contains 22 columns

Now let's have a look at some statistics from the data by using the describe function of Pandas:

```
life_expectancy.describe()
```

| | Year | Life expectancy | Adult Mortality | infant deaths | Alcohol | percentage expenditure | Hepatitis B | Measles | BMI |
|---|---|---|---|---|---|---|---|---|---|
| count | 2938.000000 | 2928.000000 | 2928.000000 | 2938.000000 | 2744.000000 | 2938.000000 | 2385.000000 | 2938.000000 | 2904.000000 |
| mean | 2007.518720 | 69.224932 | 164.796448 | 30.303948 | 4.602861 | 738.251295 | 80.940461 | 2419.592240 | 38.321247 |
| std | 4.613841 | 9.523867 | 124.292079 | 117.926501 | 4.052413 | 1987.914858 | 25.070016 | 11467.272489 | 20.044034 |
| min | 2000.000000 | 36.300000 | 1.000000 | 0.000000 | 0.010000 | 0.000000 | 1.000000 | 0.000000 | 1.000000 |
| 25% | 2004.000000 | 63.100000 | 74.000000 | 0.000000 | 0.877500 | 4.685343 | 77.000000 | 0.000000 | 19.300000 |
| 50% | 2008.000000 | 72.100000 | 144.000000 | 3.000000 | 3.755000 | 64.912906 | 92.000000 | 17.000000 | 43.500000 |
| 75% | 2012.000000 | 75.700000 | 228.000000 | 22.000000 | 7.702500 | 441.534144 | 97.000000 | 360.250000 | 56.200000 |
| max | 2015.000000 | 89.000000 | 723.000000 | 1800.000000 | 17.870000 | 19479.911610 | 99.000000 | 212183.000000 | 87.300000 |

```
life_expectancy.columns
```

Index(['Country', 'Year', 'Status', 'Life expectancy ', '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')

So there are only two categorical variables in the data which are country and status. Now let's change the names of all the columns to make them look uniform:

```
life_expectancy.rename(columns = {" BMI " :"BMI",
                                  "Life expectancy ": "Life_expectancy",
                                  "Adult Mortality":"Adult_mortality",
                                  "infant deaths":"Infant_deaths",
                                  "percentage expenditure":"Percentage_expenditure",
                                  "Hepatitis B":"HepatitisB",
                                  "Measles ":"Measles",
                                  "under-five deaths ": "Under_five_deaths",
```

```
                                        "Total expenditure":"Total_expenditure",
                                        "Diphtheria ": "Diphtheria",
                                        " thinness  1-19 years":"Thinness_1-19_years",
                                        " thinness 5-9 years":"Thinness_5-9_years",
                                        " HIV/AIDS":"HIV/AIDS",
                                        "Income composition of resources":"Income_composition_of_resources"}, inplace = True)
```

## Data Cleaning:

Now let's move further on the task of Life Expectancy analysis by looking at the null values in the dataset:

```
life_expectancy.info()
#   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   HepatitisB                       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
```

The columns that we found with null values are:

1. Life_expectancy
2. Adult_mortality
3. Alcohol
4. Hepatitis B
5. BMI
6. Polio

7. Total_expenditure
8. Diphtheria
9. GDP
10. Population
11. Thinness_1-19_years
12. Thinness_5-9_years
13. Income_composition_of_resources
14. Schooling

So there are so many columns with the null values. Now let’s have a look at how many null values all these columns are having:

```
   print(life_expectancy.isnull().sum())
Country                            0
Year                               0
Status                             0
Life_expectancy                   10
Adult_mortality                   10
Infant_deaths                      0
Alcohol                          194
Percentage_expenditure             0
HepatitisB                       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
```

There are many columns with null values, but the number of missing values is not large enough to remove the columns. So imputing missing values would be a good idea. We also know that all columns with missing values are numeric continuous variables.

Filling in the missing values with a central tendency average would not be a good idea due to the outliers. We can also fill it with the median:

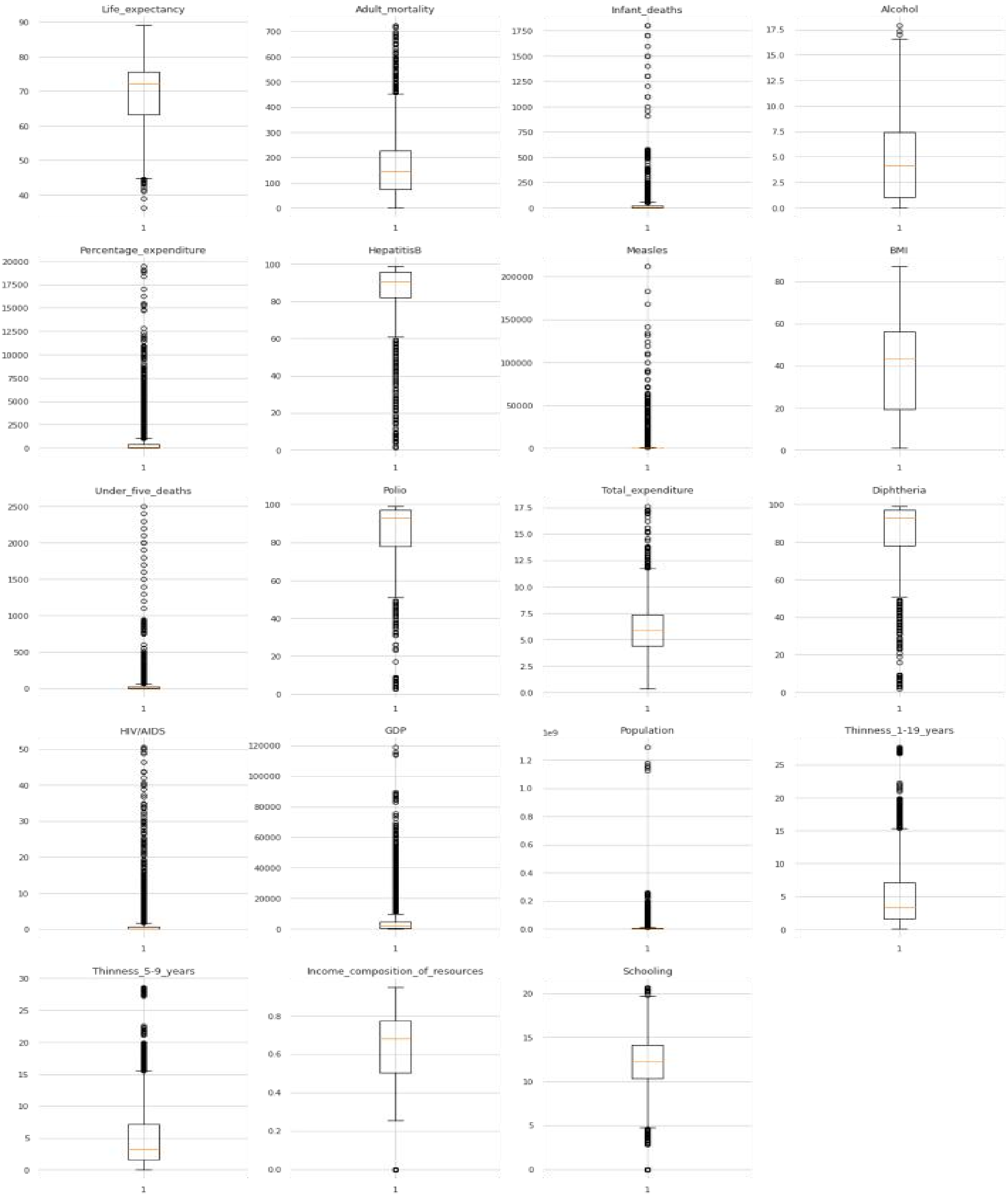
```
life_expectancy.reset_index(inplace=True)
   life_expectancy.groupby('Country').apply(lambda group: group.interpolate(method=
   'linear'))
   imputed_data = []
```

```
for year in list(life_expectancy.Year.unique()):
    year_data = life_expectancy[life_expectancy.Year == year].copy()
    for col in list(year_data.columns)[4:]:
        year_data[col] = year_data[col].fillna(year_data[col].dropna().median()).copy()
    imputed_data.append(year_data)
life_expectancy = pd.concat(imputed_data).copy()
```

## Removing Outliers:

The next step in the task of Life Expectancy analysis is to deal with outliers, let's have a look at the outliers and then we will see how we can deal with the outliers:

```
col_dict =
{'Life_expectancy':1,'Adult_mortality':2,'Infant_deaths':3,'Alcohol':4,'Percentage_expenditure':5,'HepatitisB':6,'Measles':7,'BMI':8,'Under_five_deaths':9,'Polio':10,'Total_expenditure':11,'Diphtheria':12,'HIV/AIDS':13,'GDP':14,'Population':15,'Thinness_1-19_years':16,'Thinness_5-9_years':17,'Income_composition_of_resources':18,'Schooling':19}
# Detect outliers in each variable using box plots.
fig = plt.figure(figsize=(20,30))
for variable,i in col_dict.items():
                    plt.subplot(5,4,i)
                    plt.boxplot(life_expectancy[variable])
                    plt.title(variable)
                    plt.grid(True)
plt.show()
```




Infant_Deaths represents several infant deaths per 1,000 population. That is why the number beyond 1000 is unrealistic. We will therefore remove them as outliers. The same is true for measles and deaths under five, as both are a number per 1,000 population.

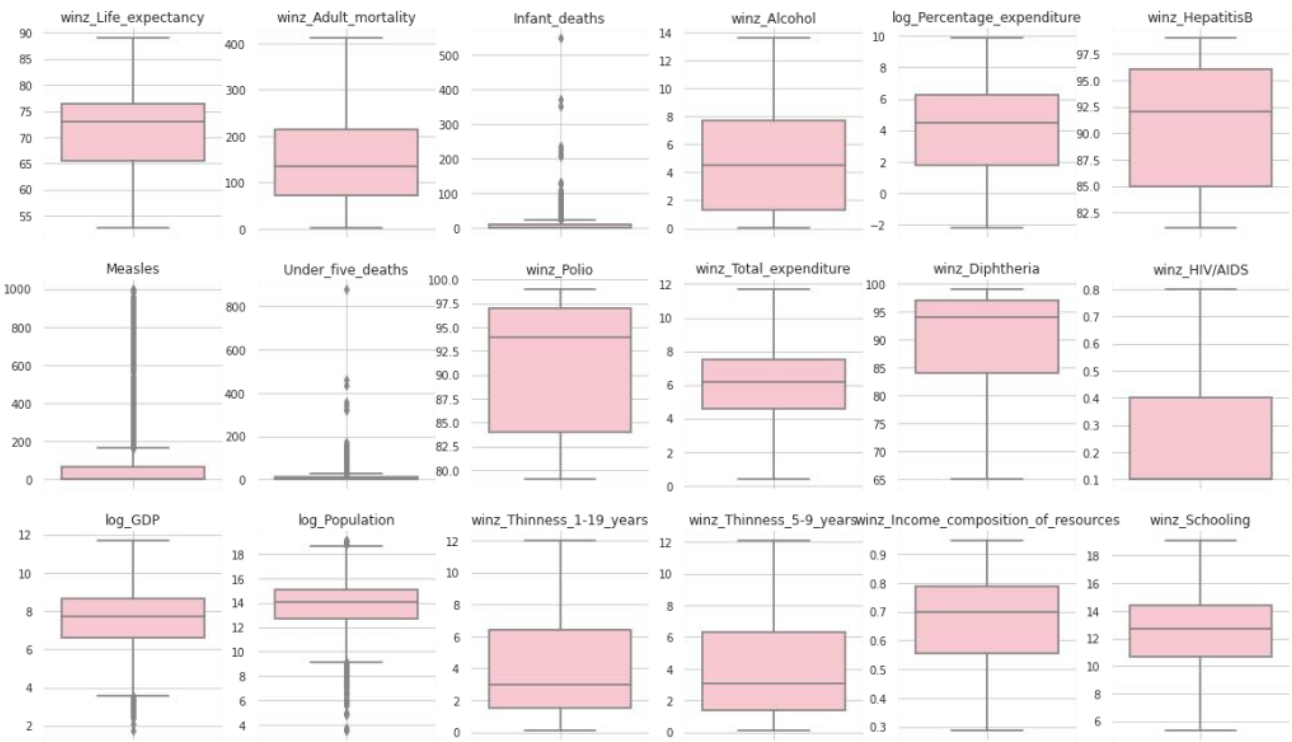
As we can see, some countries spend up to 20,000% of their GDP on health. Most countries spend less than 2,500% of their GDP on health. Since the values are very important in the Expenditure_Percentage, GDP, and Population columns, it is better to take a logarithmic value or use winsorization if necessary.

The BMI values are very unrealistic because the value plus 40 is considered extreme obesity. The median is over 40 and some countries have an average of around 60 which is not possible. We can delete this whole column.

As almost all other columns have outliers, we can use winsorization:

```
life_expectancy = life_expectancy[life_expectancy['Infant_deaths'] < 1001]
life_expectancy = life_expectancy[life_expectancy['Measles'] < 1001]
life_expectancy = life_expectancy[life_expectancy['Under_five_deaths'] < 1001]

life_expectancy.drop(['BMI'], axis=1, inplace=True)
life_expectancy['log_Percentage_expenditure'] =
np.log(life_expectancy['Percentage_expenditure'])
life_expectancy['log_Population'] = np.log(life_expectancy['Population'])
life_expectancy['log_GDP'] = np.log(life_expectancy['GDP'])
life_expectancy = life_expectancy.replace([np.inf, -np.inf], 0)
life_expectancy['log_Percentage_expenditure']

life_expectancy['winz_Life_expectancy'] = winsorize(life_expectancy['Life_expectancy'],
(0.05,0))
life_expectancy['winz_Adult_mortality'] = winsorize(life_expectancy['Adult_mortality'],
(0,0.04))
life_expectancy['winz_Alcohol'] = winsorize(life_expectancy['Alcohol'], (0.0,0.01))
life_expectancy['winz_HepatitisB'] = winsorize(life_expectancy['HepatitisB'],
(0.20,0.0))
life_expectancy['winz_Polio'] = winsorize(life_expectancy['Polio'], (0.20,0.0))
life_expectancy['winz_Total_expenditure'] =
winsorize(life_expectancy['Total_expenditure'], (0.0,0.02))
life_expectancy['winz_Diphtheria'] = winsorize(life_expectancy['Diphtheria'],
(0.11,0.0))
life_expectancy['winz_HIV/AIDS'] = winsorize(life_expectancy['HIV/AIDS'], (0.0,0.21))
life_expectancy['winz_Thinness_1-19_years'] = winsorize(life_expectancy['Thinness_1-
19_years'], (0.0,0.04))
life_expectancy['winz_Thinness_5-9_years'] = winsorize(life_expectancy['Thinness_5-
9_years'], (0.0,0.04))
life_expectancy['winz_Income_composition_of_resources'] =
winsorize(life_expectancy['Income_composition_of_resources'], (0.05,0.0))
life_expectancy['winz_Schooling'] = winsorize(life_expectancy['Schooling'], (0.03,0.01))

col_dict_winz =
{'winz_Life_expectancy':1,'winz_Adult_mortality':2,'Infant_deaths':3,'winz_Alcohol':4,

'log_Percentage_expenditure':5,'winz_HepatitisB':6,'Measles':7,'Under_five_deaths':8,'w
inz_Polio':9,

'winz_Total_expenditure':10,'winz_Diphtheria':11,'winz_HIV/AIDS':12,'log_GDP':13,'log_P
opulation':14,
            'winz_Thinness_1-19_years':15,'winz_Thinness_5-
9_years':16,'winz_Income_composition_of_resources':17,
            'winz_Schooling':18}



fig = plt.figure(figsize=(20,20))
for variable,i in col_dict_winz.items():
                     plt.subplot(5,6,i)
```

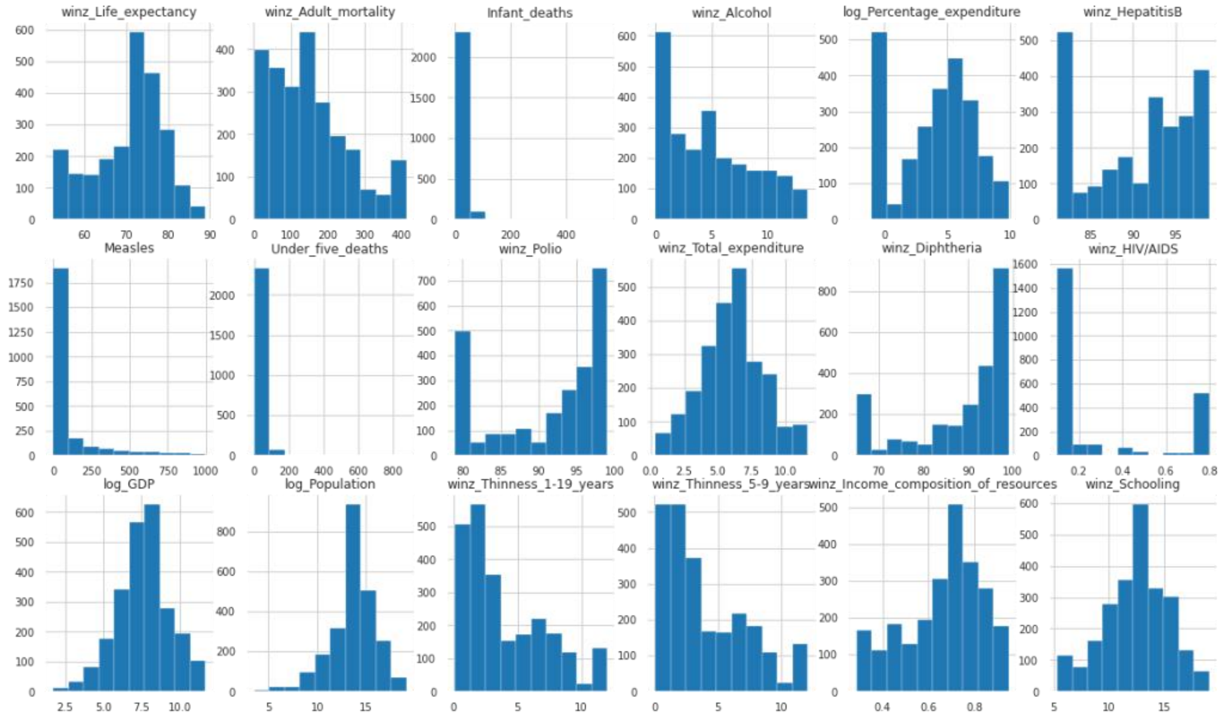
```
                    sns.boxplot(y = life_expectancy[variable], color = "pink")
                    plt.title(variable)
                    plt.ylabel('')

                    plt.grid(True)

plt.show()
```



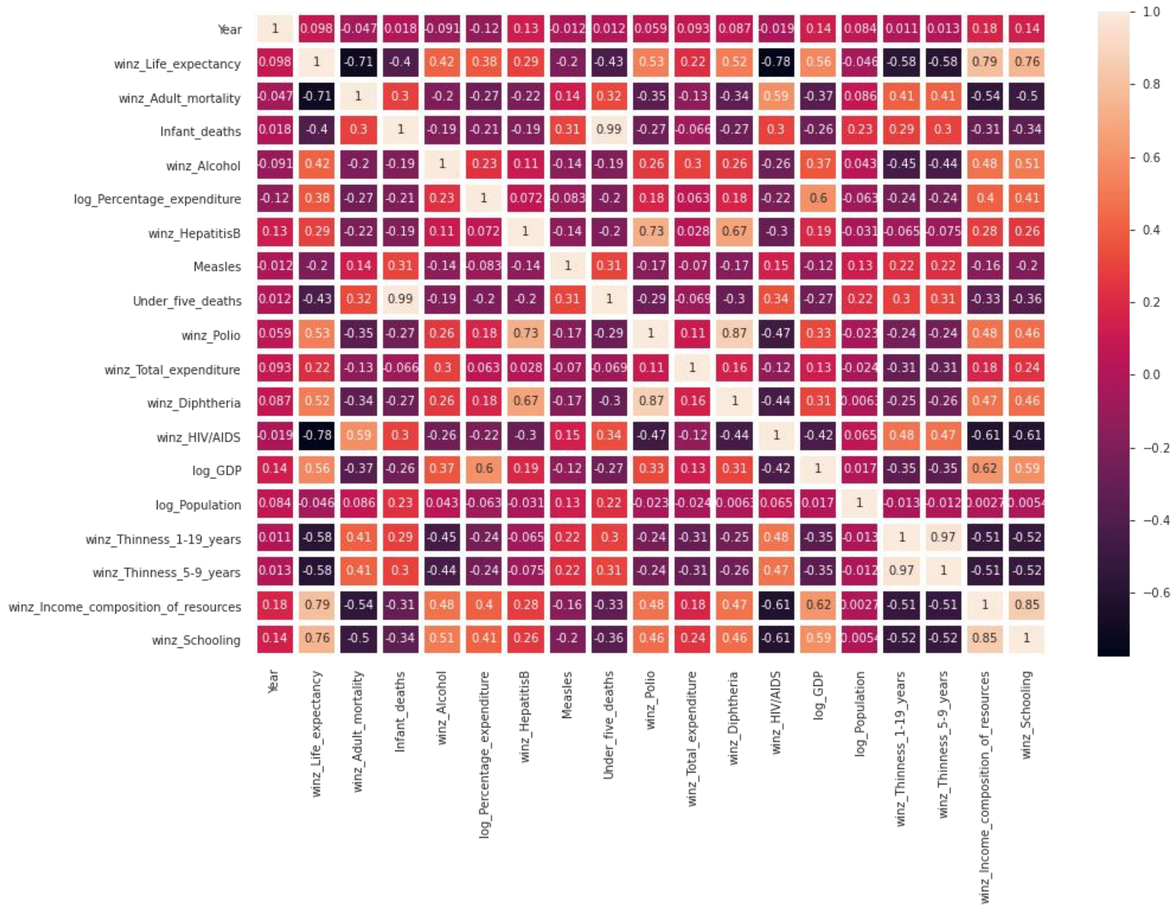

## Life Expectancy Analysis

Now we have done all the data cleaning and we also have removed all the outliers in the dataset. Now let's see move forward with the task of Life Expectancy Analysis. Let's start by exploring the data and looking at the correlation:

```
fig = plt.figure(figsize=(20,20))
for variable,i in col_dict_winz.items():
                    plt.subplot(5,6,i)
                    plt.hist(life_expectancy[variable])
                    plt.title(variable)
                    plt.ylabel('')

                    plt.grid(True)

plt.show()
```




```
life_exp = life_expectancy[['Year', 'Country',
'Status','winz_Life_expectancy','winz_Adult_mortality','Infant_deaths','winz_Alcohol',

'log_Percentage_expenditure','winz_HepatitisB','Measles','Under_five_deaths','winz_Polio',
'winz_Total_expenditure','winz_Diphtheria','winz_HIV/AIDS','log_GDP','log_Population',
'winz_Thinness_1-19_years','winz_Thinness_5-
9_years','winz_Income_composition_of_resources',
               'winz_Schooling']]
   plt.figure(figsize=(15,10))
   sns.heatmap(life_exp.corr(), annot =True, linewidths = 4)
```



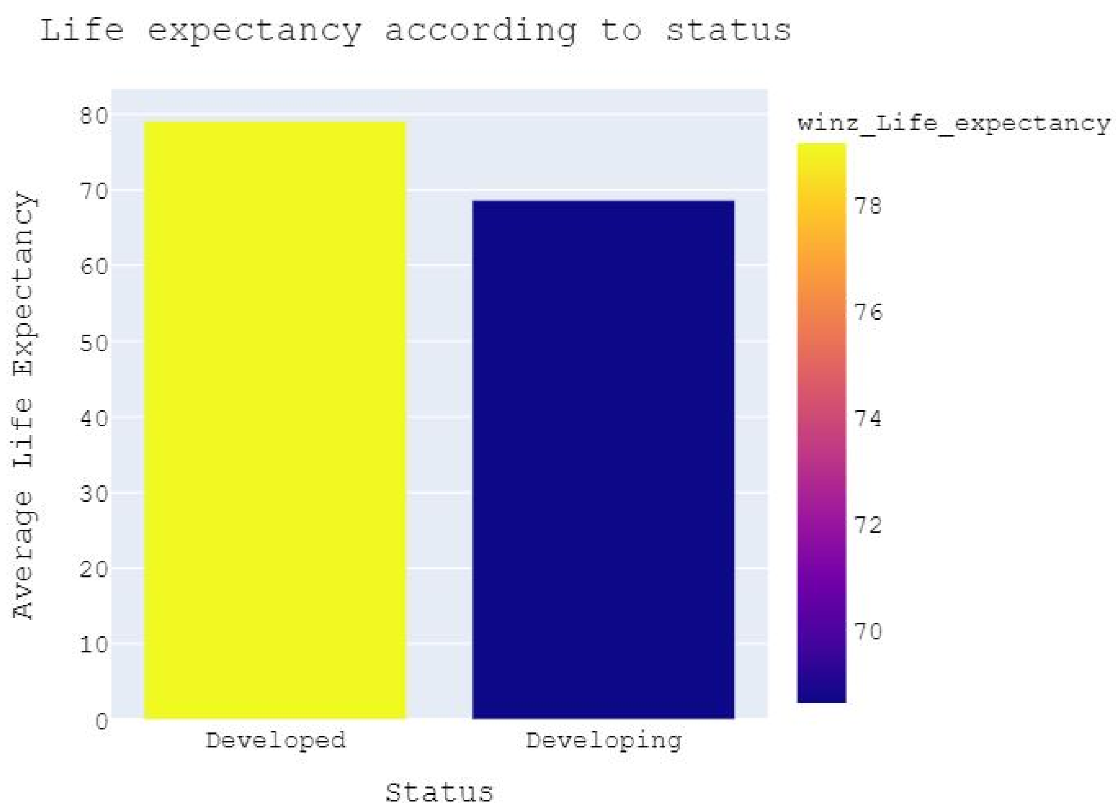

Observations from the above correlation:

- Adult_mortality has a negative relationship with education, the composition of resource income, and a positive relationship with HIV / AIDS.
- Infant_deaths and Under_five_deaths have a strong positive relationship.
- Schooling and alcohol have a positive relationship.
- Percentage expenditure has a positive relationship with education, the composition of resource income, GDP and life expectancy.
- hepatitis B has a strong positive relationship with polio and diphtheria.
- Polio also has a strong positive relationship with diphtheria, hepatitis B, and life expectancy.
- Diphtheria has a strong positive relationship with polio and life expectancy.

As we can see from the heat map, Life_expectancy has a positive relationship with education, resource income composition, GDP, diphtheria, polio, and percentage spending. Life_expectancy has a negative relationship with Adult_mortality, Thinness_1-19_years, Thinness_5-9_years, HIV / AIDS, Under_five_deaths, and Infant_deaths. Let's explore them in detail to conclude the task of life expectancy analysis:

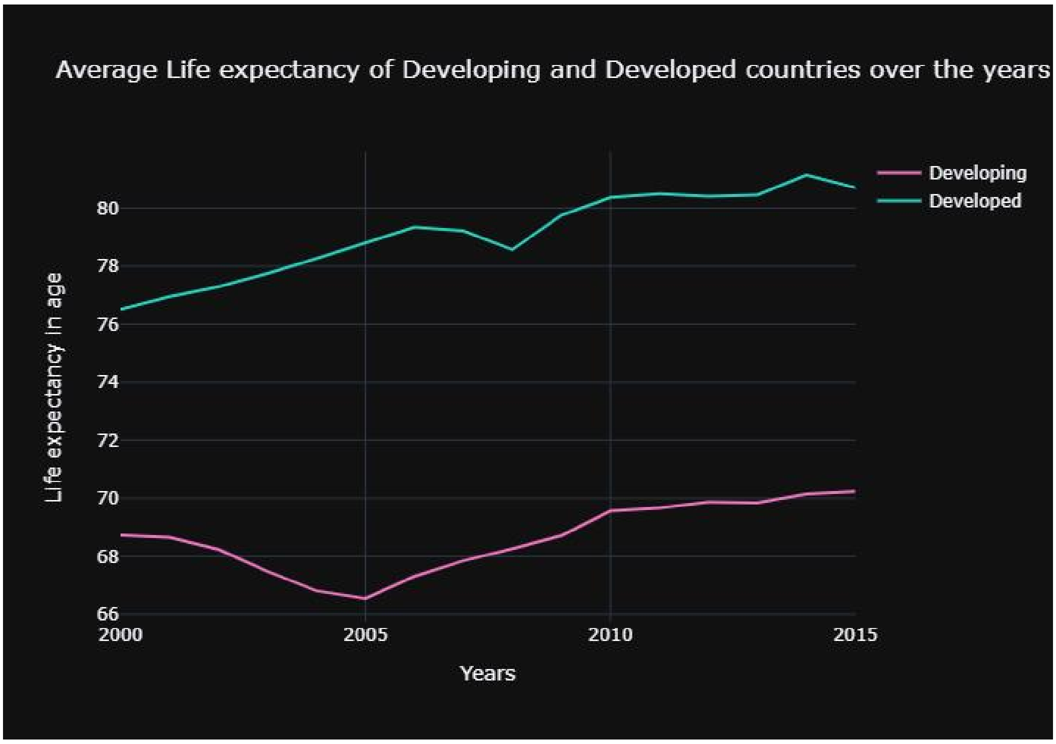
```
status_life_exp =
life_expectancy.groupby(by=['Status']).mean().reset_index().sort_values('winz_Life_expe
ctancy',ascending=False).reset_index(drop=True)
plt.figure(figsize=(20,10))

fig = px.bar(status_life_exp,  x='Status',
y='winz_Life_expectancy',color='winz_Life_expectancy')

fig.update_layout(
        title="Life expectancy according to status",
```

```
        xaxis_title="Status",
        yaxis_title="Average Life Expectancy",
        font=dict(
            family="Courier New",
            size=16,
            color="black"
        )
    )
fig.show()
```




```
life_year = life_expectancy.groupby(by = ['Year', 'Status']).mean().reset_index()
Developed = life_year.loc[life_year['Status'] == 'Developed',:]
Developing = life_year.loc[life_year['Status'] == 'Developing',:]
fig1 = go.Figure()
for template in ["plotly_dark"]:
    fig1.add_trace(go.Scatter(x=Developing['Year'],
y=Developing['winz_Life_expectancy'],
                    mode='lines',
                    name='Developing',
                    marker_color='#f075c2'))
    fig1.add_trace(go.Scatter(x=Developed['Year'], y=Developed['winz_Life_expectancy'],
                    mode='lines',
                    name='Developed',
                    marker_color='#28d2c2'))
    fig1.update_layout(
    height=500,
    xaxis_title="Years",
```

```
    yaxis_title='Life expectancy in age',
    title_text='Average Life expectancy of Developing and Developed countries over the years',
    template=template)
fig1.show()
```




We can see from the two graphs above that developed countries have more life expectancy than in developing countries.

## CONCLUTION:

Through our extensive data analysis project on life expectancy conducted using Python, several significant conclusions can be drawn:

- **Multifaceted Determinants**: Life expectancy isn't influenced by a single factor but is the culmination of a myriad of interplaying factors. Social, economic, and healthcare-related parameters often overlap in their impacts on life expectancy.

- **Economic Prosperity as a Key Indicator**: GDP per capita showcased a strong correlation with life expectancy. Nations with more robust economies tend to have better healthcare, nutrition, and education, which contribute to a higher life expectancy.

- **Healthcare Accessibility and Quality**: Countries with better healthcare infrastructure, lower infant mortality rates, and fewer epidemic diseases showed higher life expectancies, indicating the importance of robust healthcare systems.

- **Educational Attainment:** A higher average number of years of schooling for populations was directly correlated with a rise in life expectancy, showcasing the role of education in health awareness and socio-economic upliftment.

- **Prevalence of Diseases:** Certain diseases, such as HIV/AIDS, significantly reduce life expectancy in affected regions. Our analysis underlined the importance of combating these diseases to enhance global life expectancy.

- **Predictive Modeling:** Using machine learning techniques in Python, we developed a predictive model with reasonable accuracy. This model can assist policymakers in estimating the impact of their decisions on life expectancy.

- **Role of Python:** Python's versatile libraries like Pandas for data manipulation, Matplotlib and Seaborn for visualization, and Scikit-learn for modeling were instrumental in the in-depth analysis of the data.

In sum, life expectancy serves as an essential metric reflecting a country's overall health, socio-economic conditions, and the quality of life of its inhabitants. While improving life expectancy is a complex task that requires multifaceted strategies, data-driven insights, such as those gleaned from our Python-based analysis, can guide effective interventions and policies.