# LIFE EXPECTANCY ANALYSIS WITH PYTHON

**Name:** Kirthigaa. S

**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

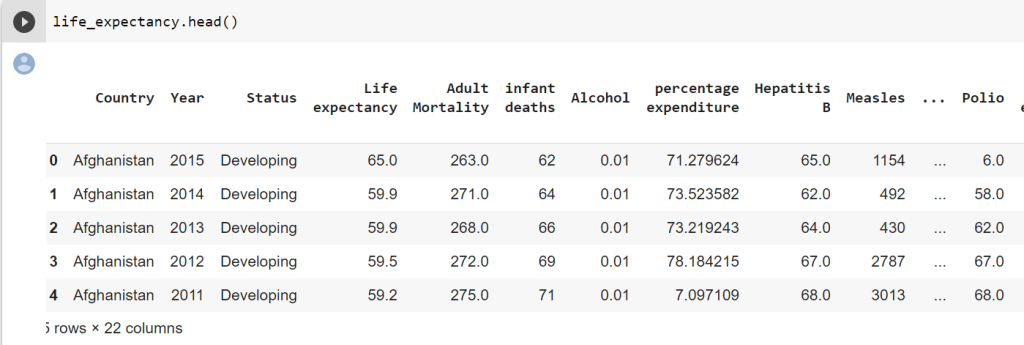
from matplotlib import rc

import scipy.stats

from scipy.stats.mstats import winsorize

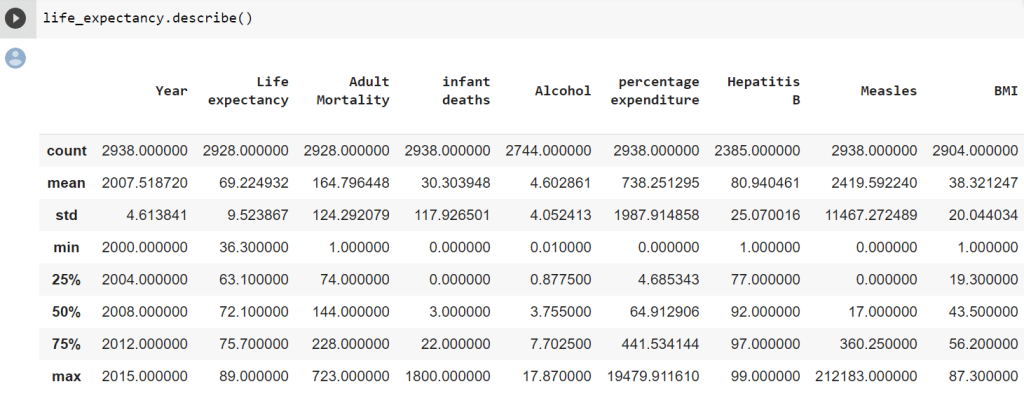
life\_expectancy = pd.read\_csv("Life Expectancy Data.csv") #reading the file

life\_expectancy.head()



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.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:

1

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():

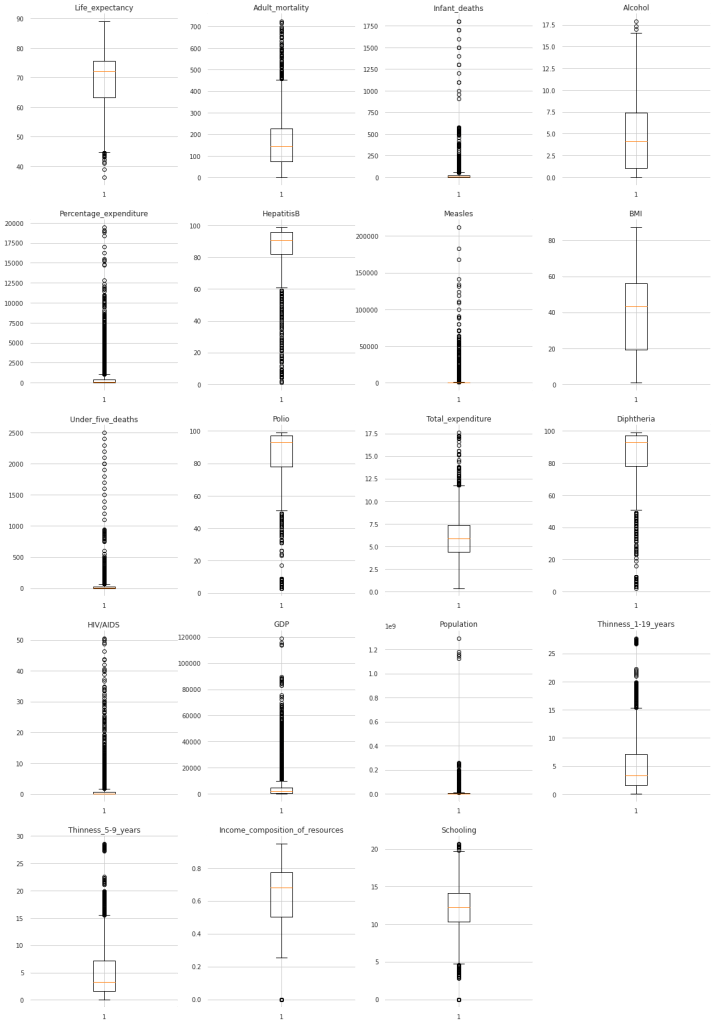
plt.subplot(5,4,i)

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,'winz\_Polio':9,

'winz\_Total\_expenditure':10,'winz\_Diphtheria':11,'winz\_HIV/AIDS':12,'log\_GDP':13,'log\_Population':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)

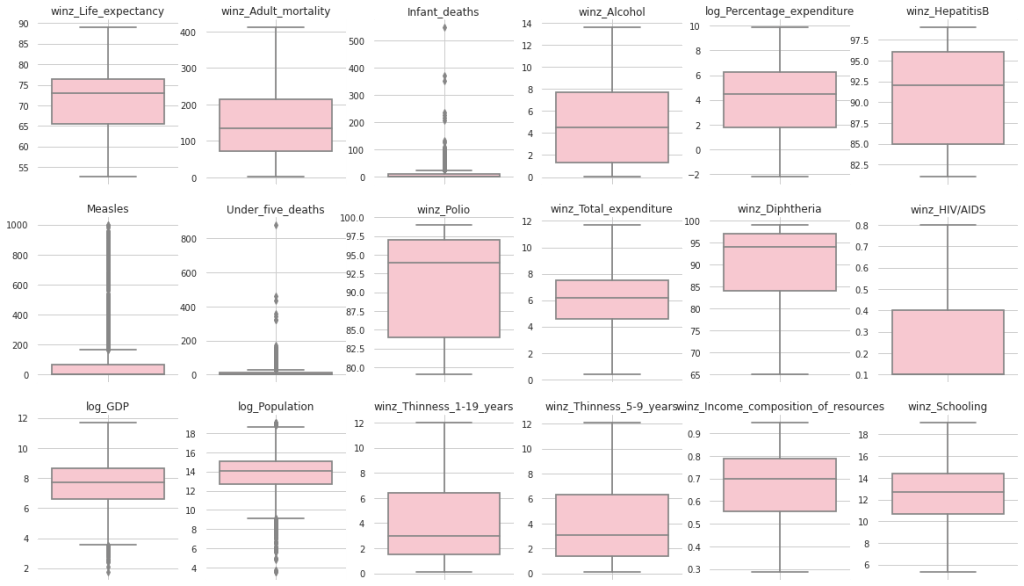
sns.boxplot(y = life\_expectancy[variable], color = "pink")

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)

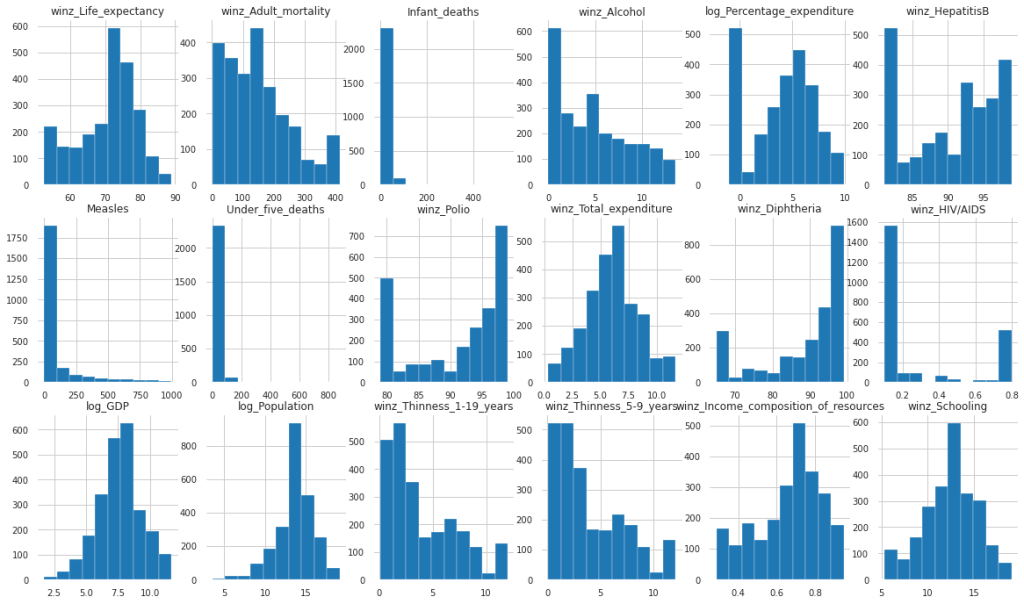
plt.hist(life\_expectancy[variable])

plt.title(variable)

plt.ylabel('')

plt.grid(True)

plt.show()



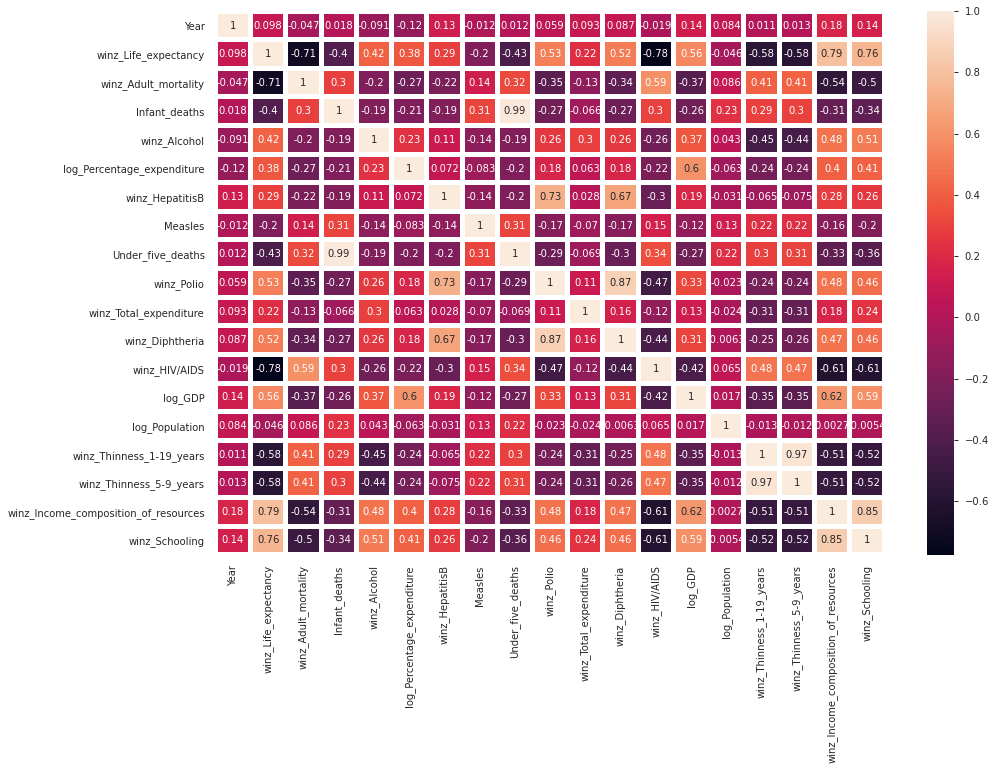
life\_exp = life\_expectancy[['Year', 'Country', 'Status','winz\_Life\_expectancy','winz\_Adult\_mortality','Infant\_deaths','winz\_Alcohol',

'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\_expectancy',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",

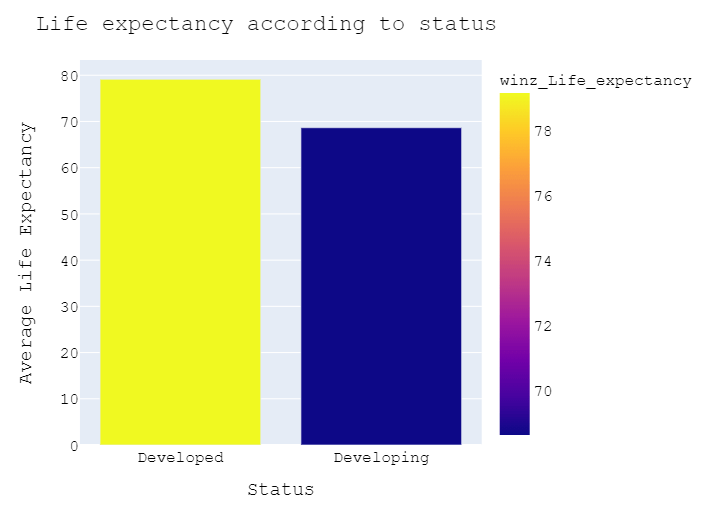
size=16,

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,

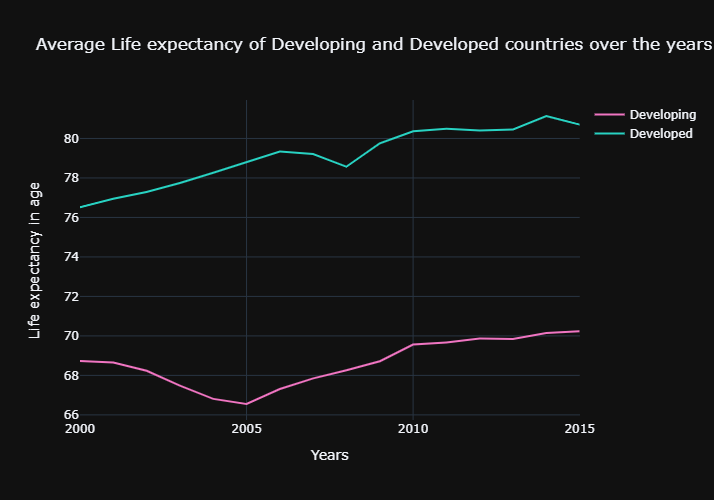
xaxis\_title="Years",

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.