LIFE EXPECTANCY BY WHO

About Dataset

Context

This gives motivation to resolve a regression model based on mixed effects model and multiple linear regression while considering data from a period of 2000 to 2015 for all the countries. Important immunization like Hepatitis B, Polio and Diphtheria will also be considered. In a nutshell, this study will focus on immunization factors, mortality factors, economic factors, social factors and other health related factors as well. Since the observations this dataset are based on different countries, it will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy. This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.

Content

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-set related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website. Among all categories of health-related factors only those critical factors were chosen which are more representative. It has been observed that in the past 15 years , there has been a huge development in health sector resulting in improvement of human mortality rates especially in the developing nations in comparison to the past 30 years. Therefore, in this project we have considered data from year 2000-2015 for 193 . The final dataset consists of 22 Columns and 2938 rows which meant 20 predicting variables. All predicting variables was then divided into several broad categories:Immunization related factors, Mortality factors, Economical factors and Social factors.

Importing Necessary Libraries

In [78]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
from sklearn.preprocessing import LabelEncoder,StandardScaler
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import GradientBoostingRegressor,RandomForestRegressor,AdaBoostRegr
from sklearn.svm import LinearSVR
```

In [79]:

```
df=pd.read_csv("./Life Expectancy Data (1).csv")
df.head()
```

Out[79]:

	Country	Year	Status	Life_expectancy	Adult_ Mortality	infant_deaths	Alcohol	percentag
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	

5 rows × 22 columns

EXPLORATORY DATA ANALYSIS

In [80]:

```
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                   2771 non-null   float64
 21  Schooling                2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

In [4]:

```
df.describe()
```

Out[4]:

	Year	Life_expectancy	Adult_ Mortality	infant_deaths	Alcohol	percentage_exp
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	293
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	73
std	4.613841	9.523867	124.292079	117.926501	4.052413	198
min	2000.000000	36.300000	1.000000	0.000000	0.010000	
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	6
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	44
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	1947

In [5]:

```
df.shape
```

Out[5]:

```
(2938, 22)
```

In [6]:

```
df.duplicated().sum()
```

Out[6]:

```
0
```

In [7]:

```
df.isnull().mean()*100
```

Out[7]:

```
Country                          0.000000
Year                             0.000000
Status                           0.000000
Life_expectancy                  0.340368
Adult_ Mortality                 0.340368
infant_deaths                    0.000000
Alcohol                          6.603131
percentage_expenditure           0.000000
Hepatitis_B                     18.822328
Measles                          0.000000
 BMI                             1.157250
under_five deaths                0.000000
Polio                            0.646698
Total_expenditure                7.692308
Diphtheria                       0.646698
 HIV_AIDS                        0.000000
GDP                             15.248468
Population                      22.191967
 thinness _1-19 years            1.157250
 thinness 5-9 years              1.157250
Income                           5.684139
Schooling                        5.547992
dtype: float64
```

In [8]:

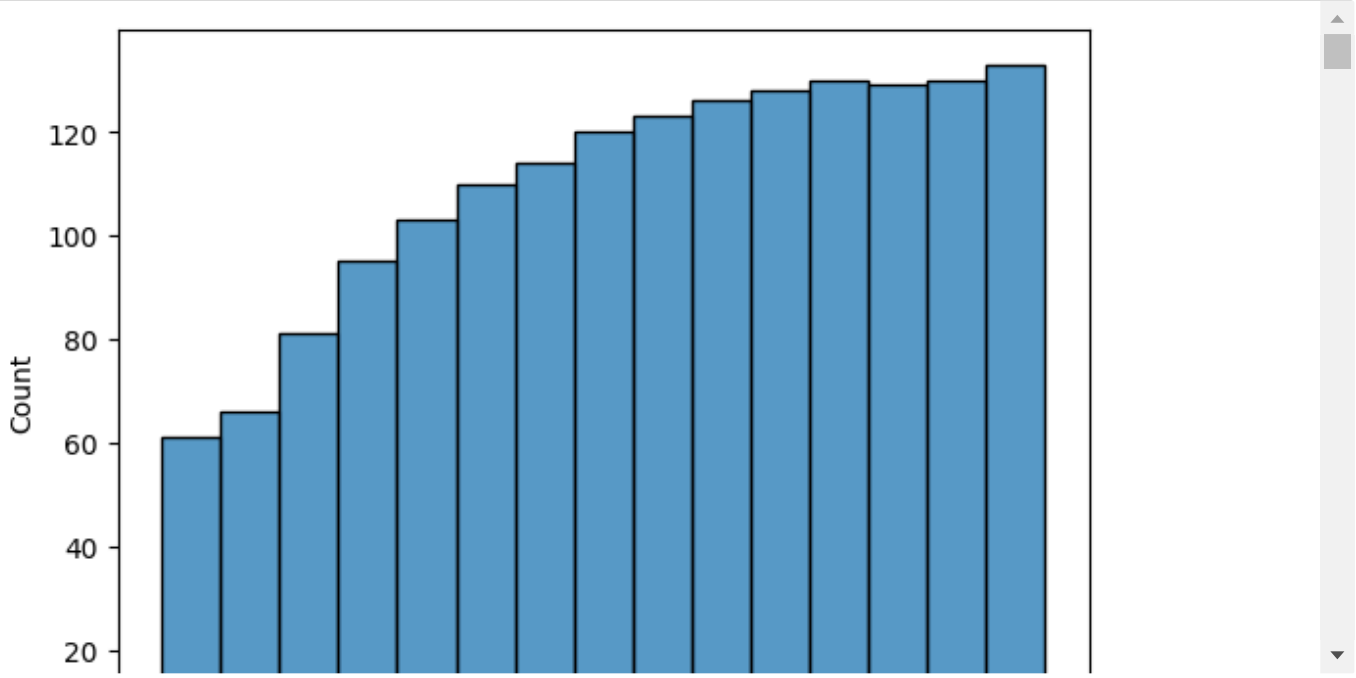
```
df.dropna(inplace=True)
```

Univariate Analysis

Histogram

In [9]:

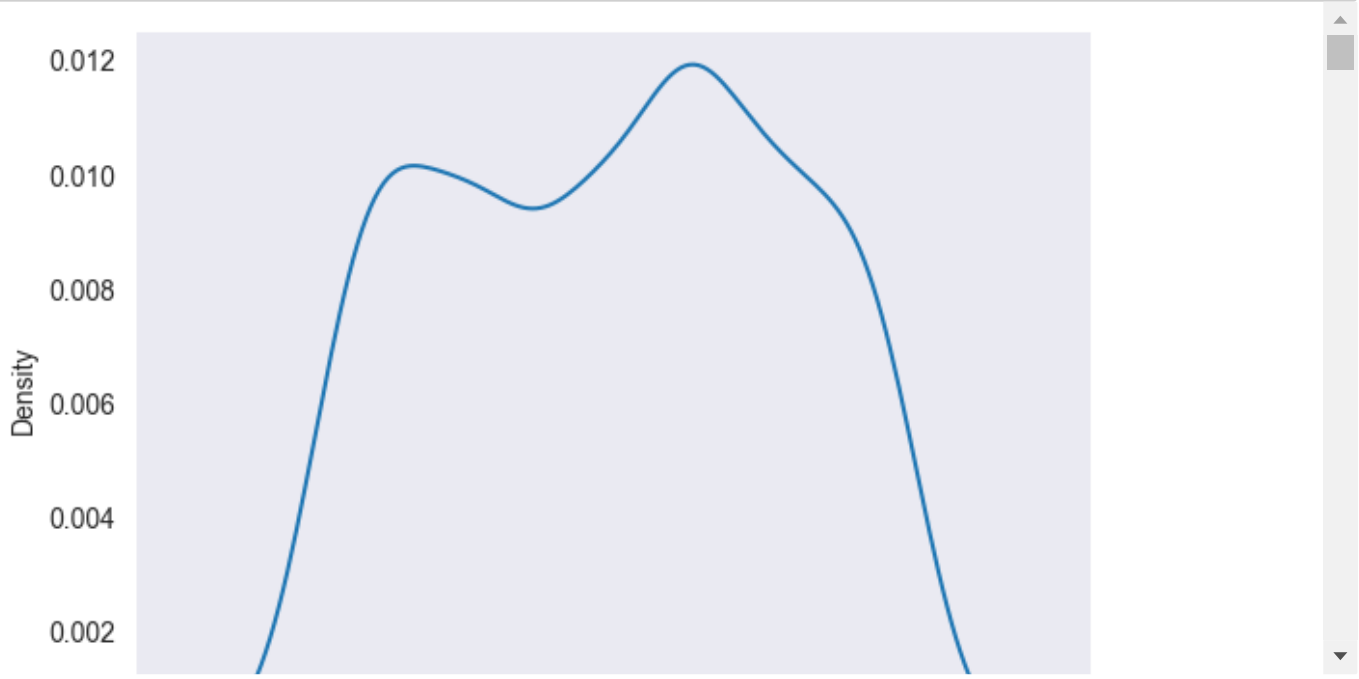
```
for i in df.columns:
    if df[i].dtypes!="object":
        sns.histplot(x=df[i])
        plt.show()
```



KDE Plot

In [40]:

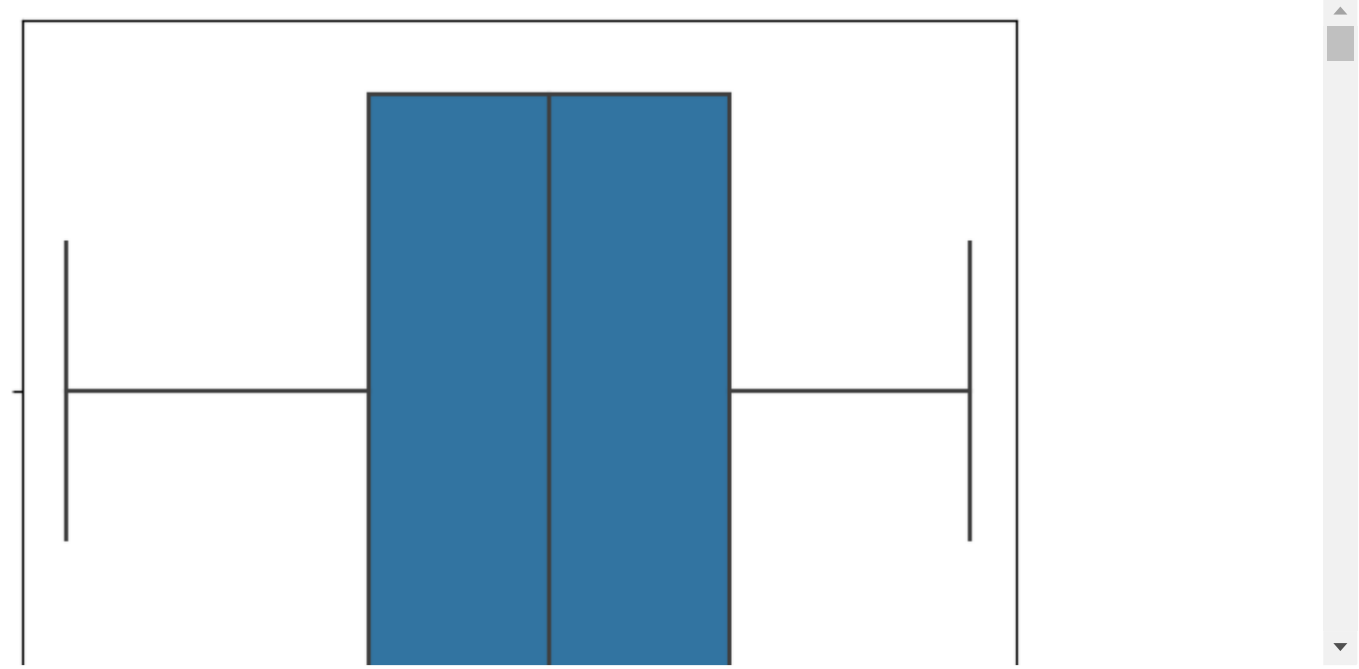
```
for i in df.columns:
    if df[i].dtypes!="object":
        sns.kdeplot(x=df[i])
        plt.show()
```



Box Plot

In [11]:

```
for i in df.columns:
    if df[i].dtypes!="object":
        sns.boxplot(x=df[i])
        plt.show()
```



Outlier Treatment

In [41]:

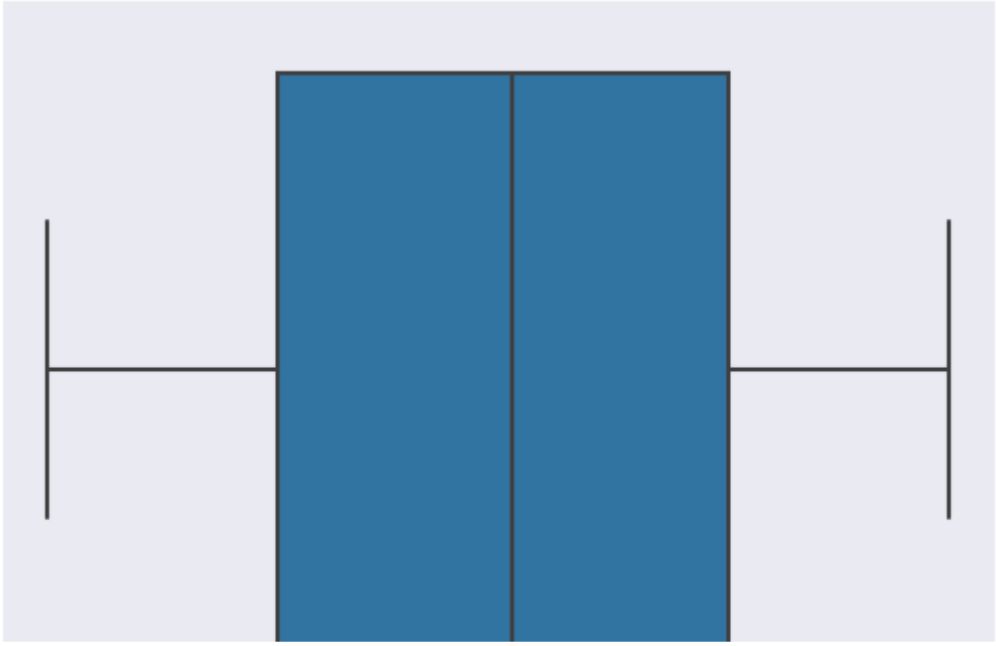
```
def outlier_limit(col):
    Q3,Q1=np.nanpercentile(col,[75,25])
    IQR=Q3-Q1
    UL=Q3+1.5*IQR
    LL=Q1-1.5*IQR
    return UL,LL
```

In [42]:

```
for column in df.columns:
    if df[column].dtype !="object":
        UL,LL=outlier_limit(df[column])
        df[column]=np.where((df[column]>UL)|(df[column]<LL),np.nan,df[column])
```

In [43]:

```
for i in df.columns:
    if df[i].dtypes!="object":
        sns.boxplot(x=df[i])
        plt.show()
```



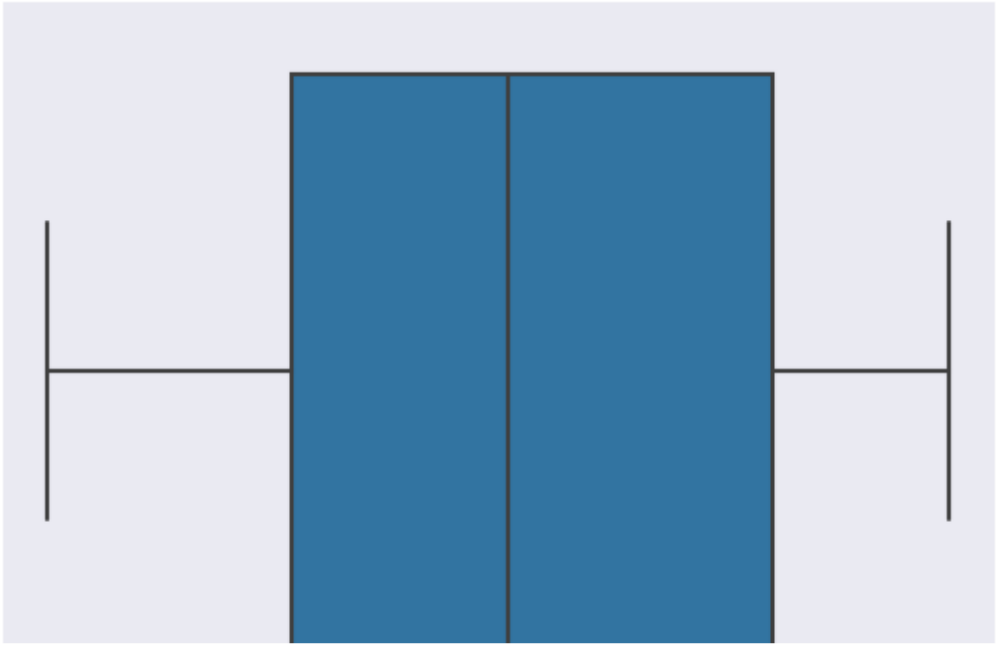
In [44]:

```
df.dropna(inplace=True)
```

Plotting the Box Plot after the Outlier Treatment

In [46]:

```
for i in df.columns:
    if df[i].dtypes!="object":
        sns.boxplot(x=df[i])
        plt.show()
```



In [48]:

```
df.dropna(inplace=True)
```

Multivariate Analysis

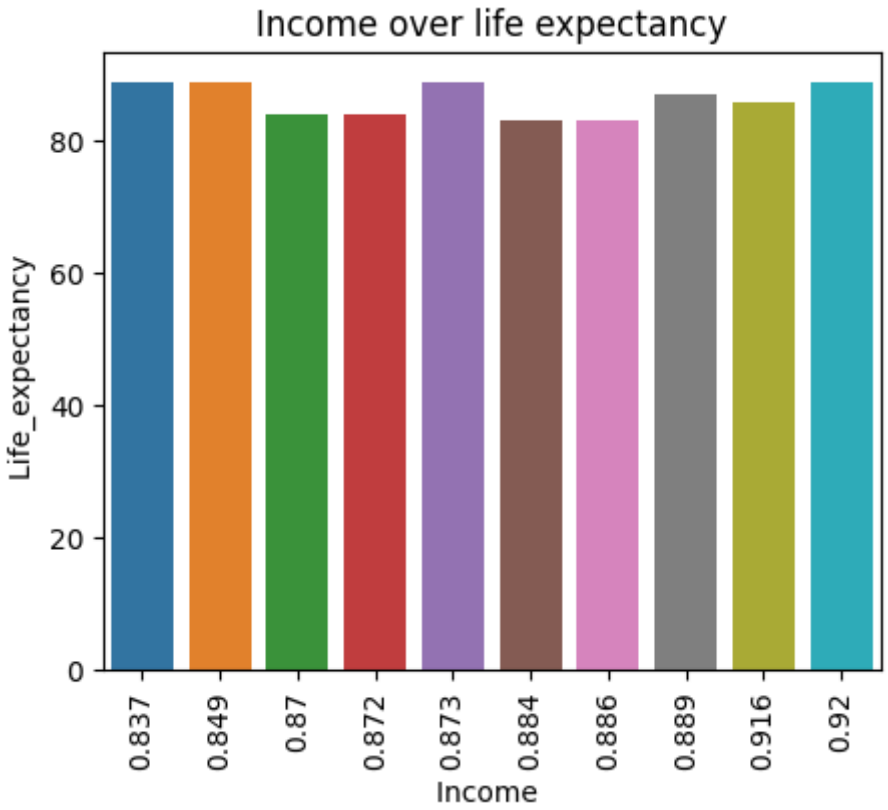
Bar Plot

In [17]:

```
plt.figure(figsize=(5,4))
sns.barplot(x='Income ',y='Life_expectancy ',data=df.sort_values(by='Life_expectancy ',as
plt.xticks(rotation=90)
plt.title('Income over life expectancy')
```

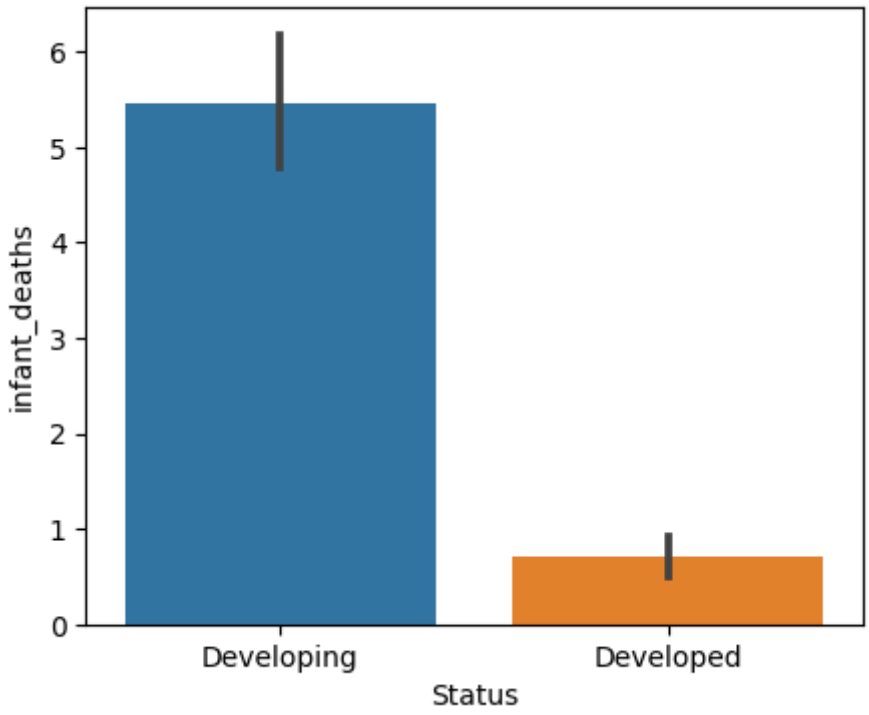
Out[17]:

```
Text(0.5, 1.0, 'Income over life expectancy')
```



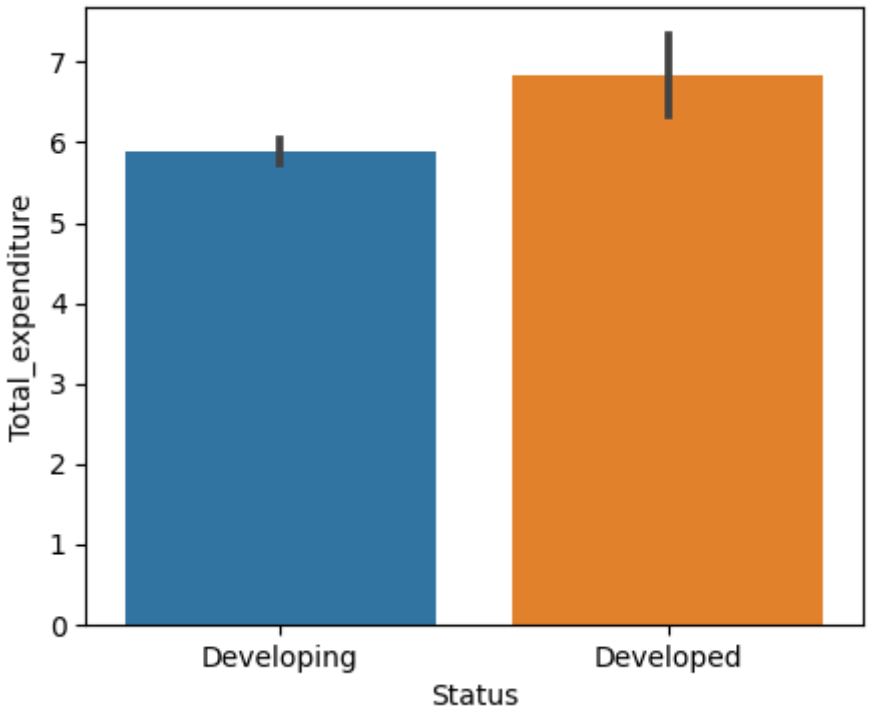
In [18]:

```
plt.figure(figsize=(5,4))
sns.barplot(x='Status',y='infant_deaths',data=df.sort_values(by='Status',ascending=False)
plt.show()
```



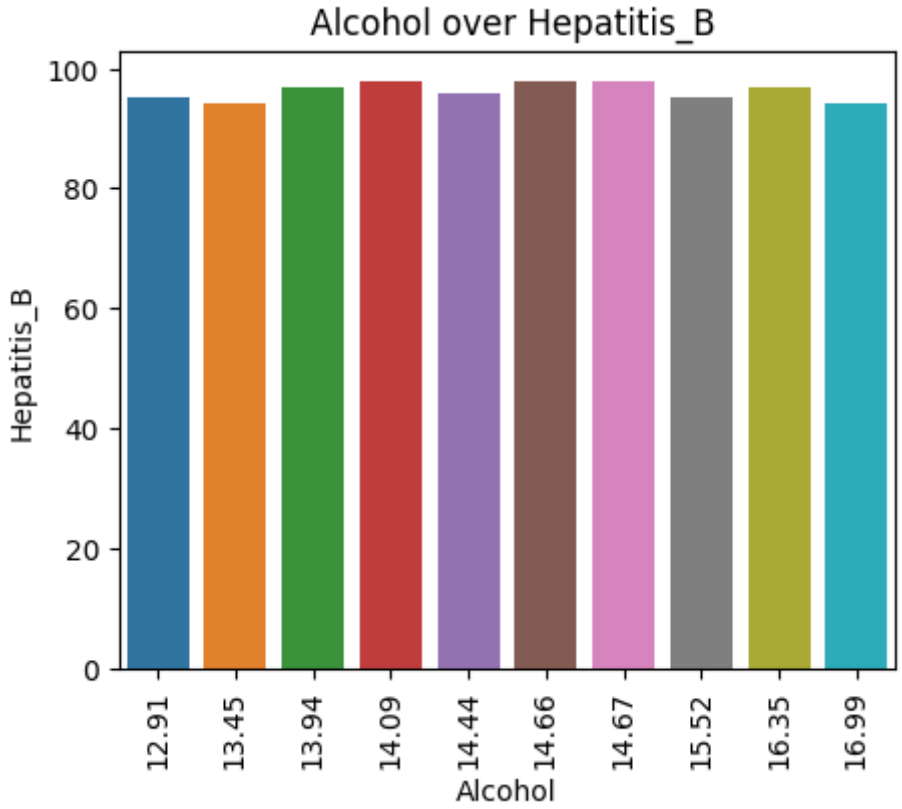
In [19]:

```
plt.figure(figsize=(5,4))
sns.barplot(x='Status',y='Total_expenditure',data=df.sort_values(by='Status',ascending=Fa
plt.show()
```



In [20]:

```
plt.figure(figsize=(5,4))
sns.barplot(x='Alcohol',y='Hepatitis_B',data=df.sort_values(by='Alcohol',ascending=False))
plt.title(' Alcohol over Hepatitis_B')
plt.xticks(rotation=90)
plt.show()
```

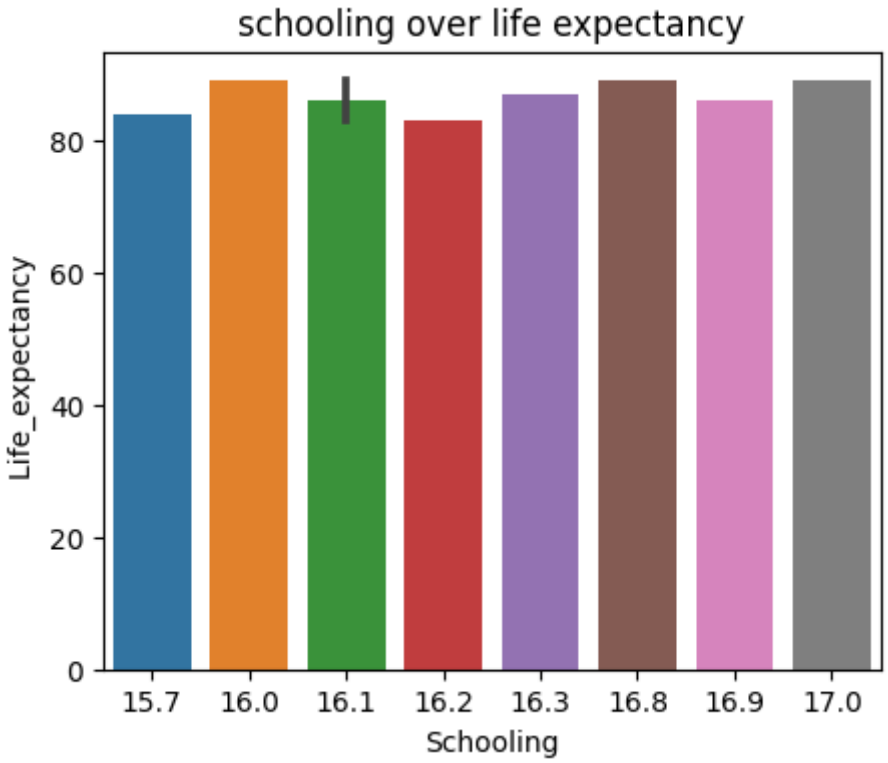


In [21]:

```
plt.figure(figsize=(5,4))
sns.barplot(x='Schooling',y='Life_expectancy ',data=df.sort_values(by='Life_expectancy ',
plt.title('schooling over life expectancy')
```

Out[21]:

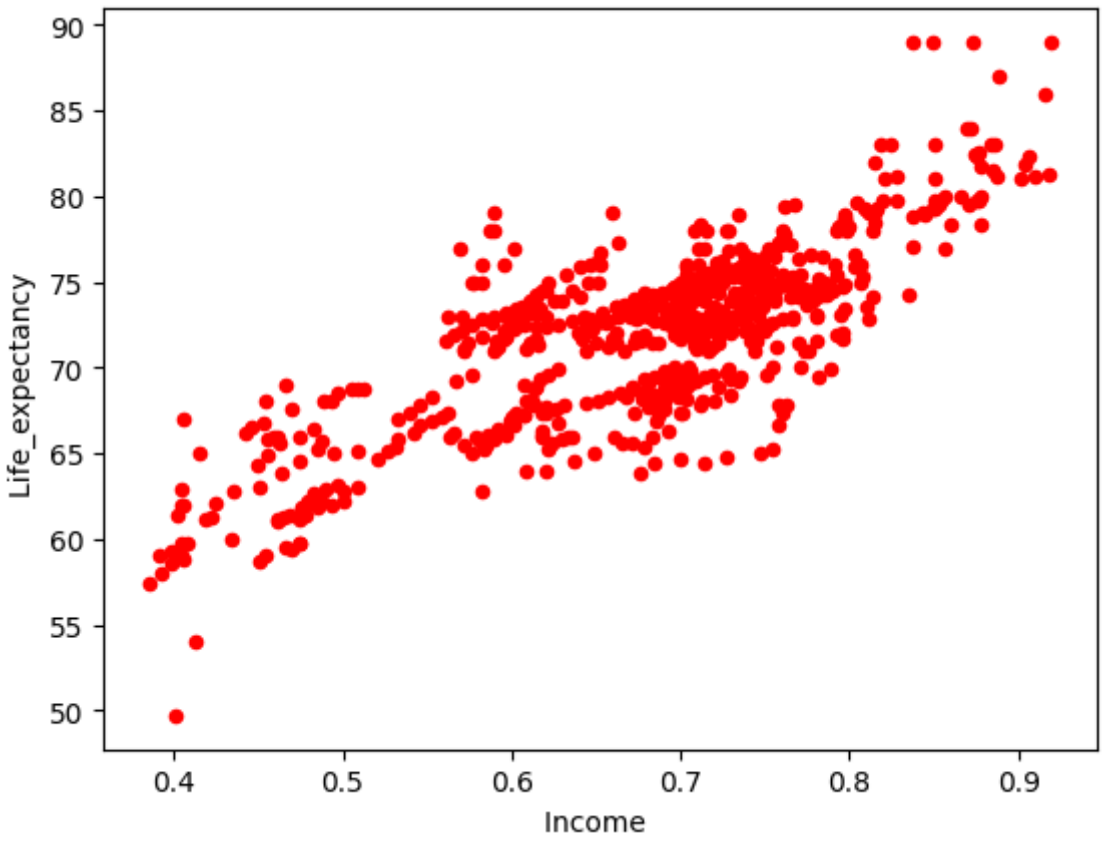
```
Text(0.5, 1.0, 'schooling over life expectancy')
```



Scatter Plot

In [22]:

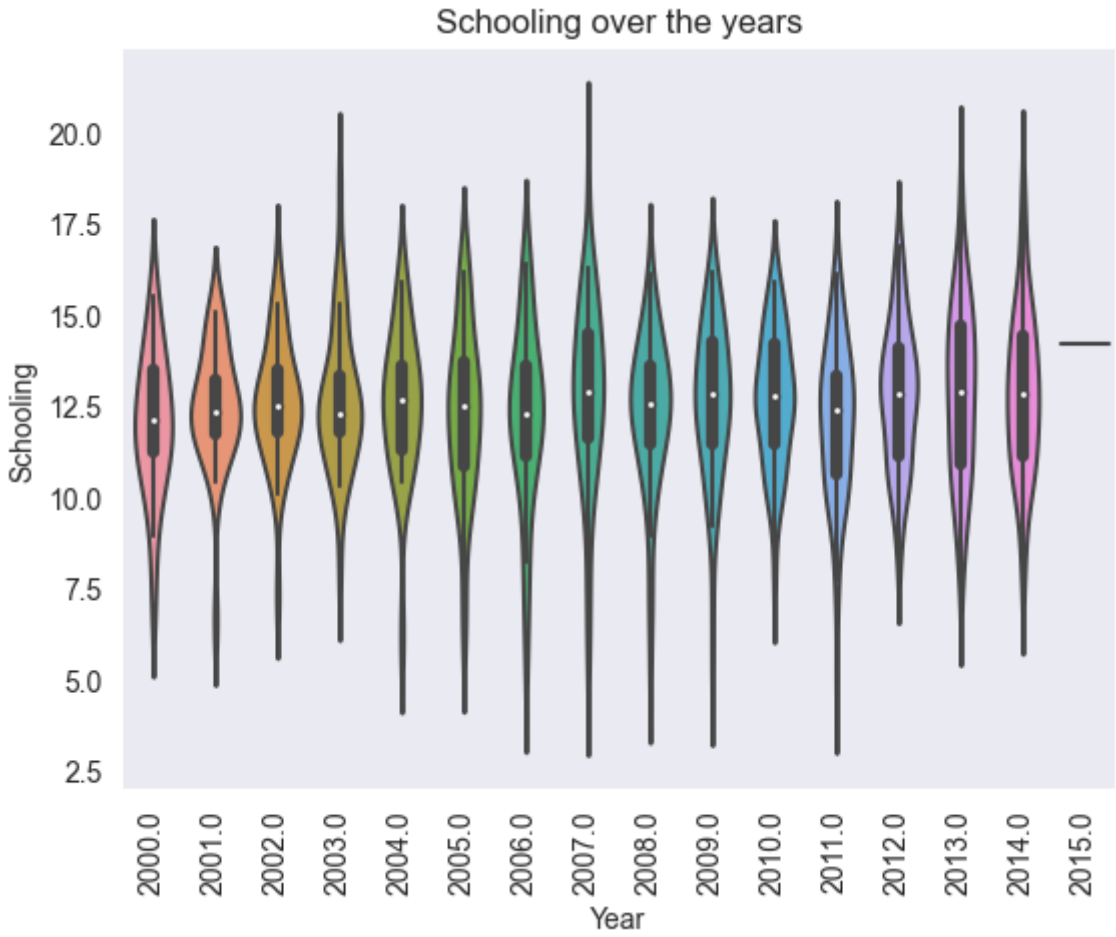
```
df.plot.scatter(x="Income ",y='Life_expectancy ',color="red");
```



Violin Plot

In [23]:

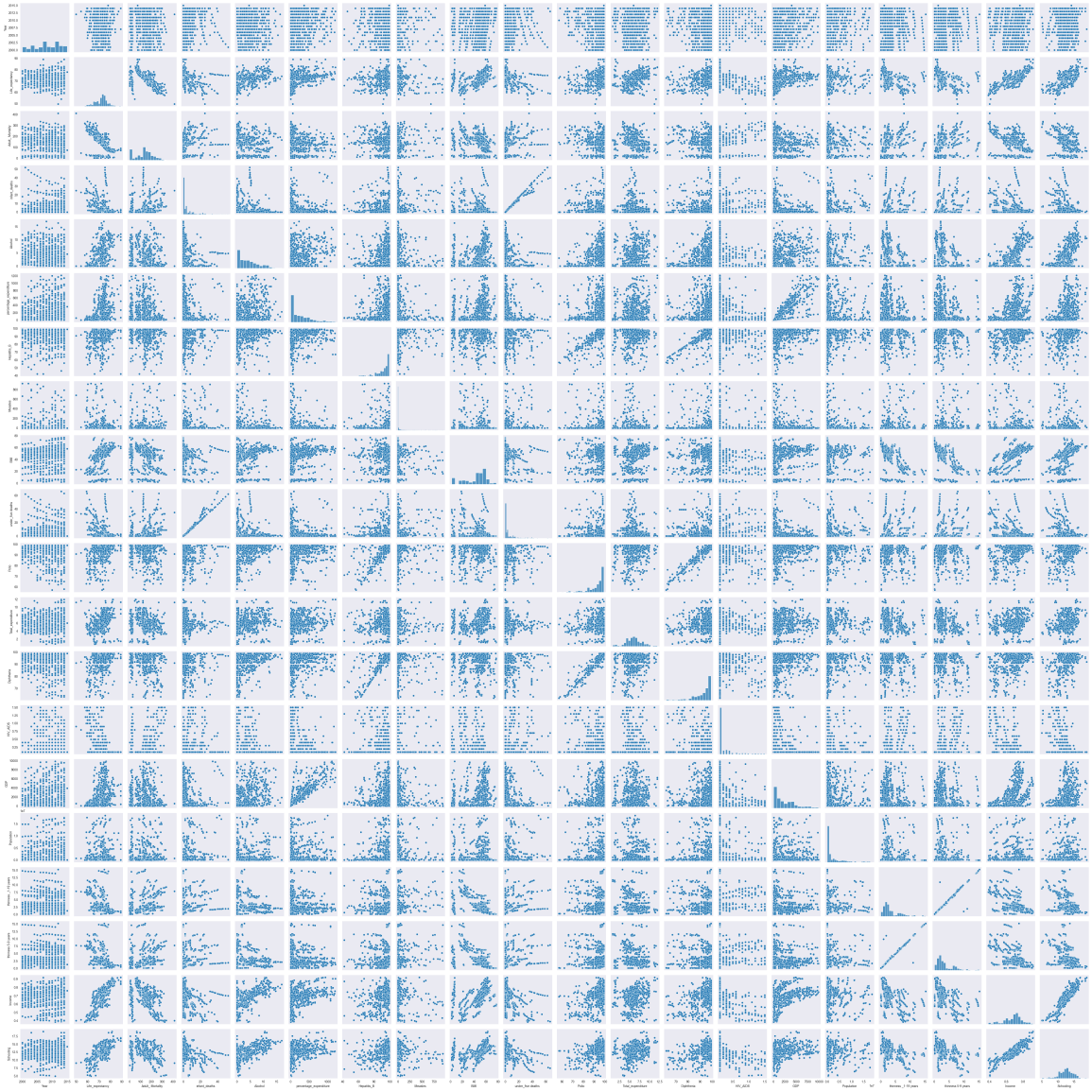
```
sns.set_style('dark')
sns.violinplot(x='Year',y='Schooling',figsize=(30,20),data=df.sort_values(by='Schooling',
plt.title('Schooling over the years')
plt.xticks(rotation=90)
plt.show()
```



Pair Plot

In [24]:

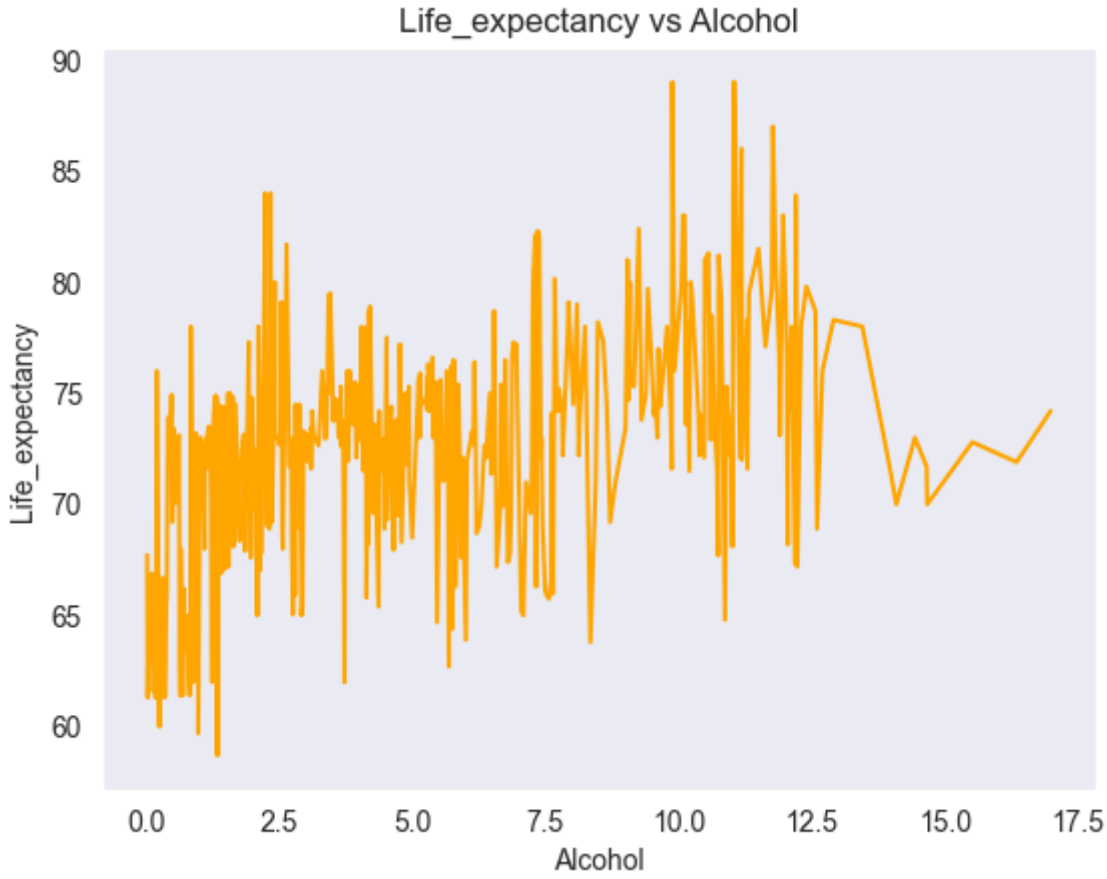
```
sns.pairplot(data=df)
plt.show()
```



Line Plot

In [25]:

```
sns.lineplot(data=df,x='Alcohol',y='Life_expectancy ',color="orange",ci=None)
plt.title('Life_expectancy vs Alcohol' );
```



Label Encoding the Catagorical Columns

In [49]:

```
le=LabelEncoder()
df["Country"]=le.fit_transform(df["Country"])
df['Status']=le.fit_transform(df['Status'])
df
```

Out[49]:

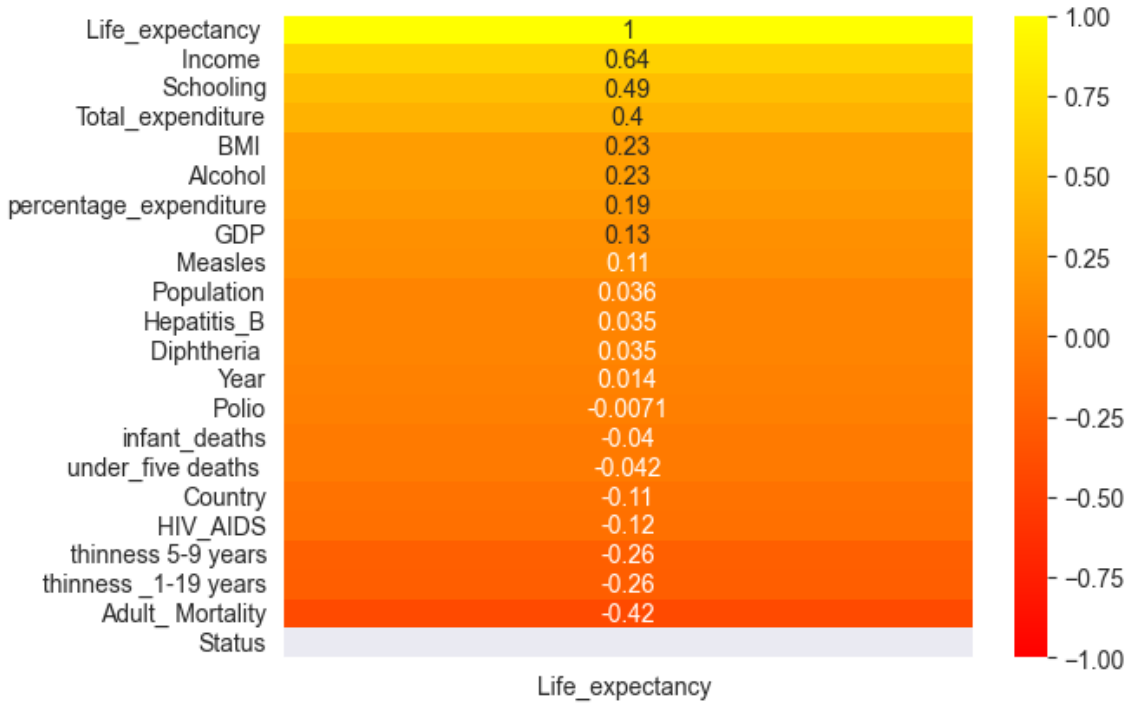
	Country	Year	Status	Life_expectancy	Adult_ Mortality	infant_deaths	Alcohol	percentage_
16	0	2015.0	0	77.8	74.0	0.0	4.60	
17	0	2014.0	0	77.5	8.0	0.0	4.51	
18	0	2013.0	0	77.2	84.0	0.0	4.76	
19	0	2012.0	0	76.9	86.0	0.0	5.14	
20	0	2011.0	0	76.6	88.0	0.0	5.37	
...	...	...	...	...	...	...	...	
2817	45	2008.0	0	76.4	119.0	1.0	6.76	
2818	45	2007.0	0	75.4	124.0	1.0	6.67	
2822	45	2003.0	0	75.4	121.0	1.0	5.11	
2823	45	2002.0	0	75.4	124.0	1.0	5.86	
2824	45	2001.0	0	75.2	123.0	1.0	6.48	

305 rows × 22 columns

Checking the Correlation with the Target 'Life_expectancy'

In [50]:

```
sns.heatmap(df.corr()[["Life_expectancy "]].sort_values(by="Life_expectancy ",ascending=F
```



In [28]:

```
df.columns
```

Out[28]:

```
Index(['Country', 'Year', 'Status', 'Life_expectancy ', 'Adult_ Mortalit
y',
       'infant_deaths', 'Alcohol', 'percentage_expenditure', 'Hepatitis_
B',
       'Measles', ' BMI ', 'under_five deaths ', 'Polio', 'Total_expenditu
re',
       'Diphtheria ', ' HIV_AIDS', 'GDP', 'Population',
       ' thinness _1-19 years', ' thinness 5-9 years', 'Income ', 'Schooli
ng'],
      dtype='object')
```

Lets take only the columns with good correlation with the target

In [51]:

```
x=df[['Income ', 'Schooling', 'Alcohol',"Total_expenditure"]].values
```

In [52]:

```
y=df["Life_expectancy "].values
```

Feature Scaling

In [53]:

```
ss=StandardScaler()
x=ss.fit_transform(x)
```

Model Building

In [81]:

```
models = {
    "LinearSVR":LinearSVR(),
    "DecisionTreeRegressor":DecisionTreeRegressor(),
    "GradientBoostingRegressor":GradientBoostingRegressor(),
    "AdaBoostRegressor":AdaBoostRegressor(),
    "RandomForestRegressor":RandomForestRegressor()
}
```

In [82]:

```
for name, model in models.items():
    scores = cross_val_score(model, x,y, scoring="neg_mean_squared_error",cv=10,n_jobs=-1
    print("cross validation model : {}".format(name))
    rmse = np.sqrt(-scores)
    rmse_average = np.mean(rmse)
    print("AVERAGE RMSE: ",rmse_average)
    print("*"*100)
```

```
cross validation model : LinearSVR
AVERAGE RMSE:  3.323705882082934
****************************************************************************************************
cross validation model : DecisionTreeRegressor
AVERAGE RMSE:  3.9756479717929993
****************************************************************************************************
cross validation model : GradientBoostingRegressor
AVERAGE RMSE:  3.2663963624925514
****************************************************************************************************
cross validation model : AdaBoostRegressor
AVERAGE RMSE:  3.2523302006395953
****************************************************************************************************
cross validation model : RandomForestRegressor
AVERAGE RMSE:  3.1704662268883714
****************************************************************************************************
```

Selecting RandomForestRegressor as it is having Best Metrics

In [76]:

```
model =RandomForestRegressor()
```

In [77]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=20)
```

In [70]:

```
model.fit(x_train,y_train)
```

Out[70]:

```
RandomForestRegressor()
```

In [71]:

```
model.score(x,y)
```

Out[71]:

```
0.8966741791474415
```

In [72]:

```
y_pred = model.predict(x)
OUTPUT = pd.DataFrame(zip(y,y_pred), columns=("ACTUAL", "PREDICTION"),
dtype=float)
OUTPUT.head()
```

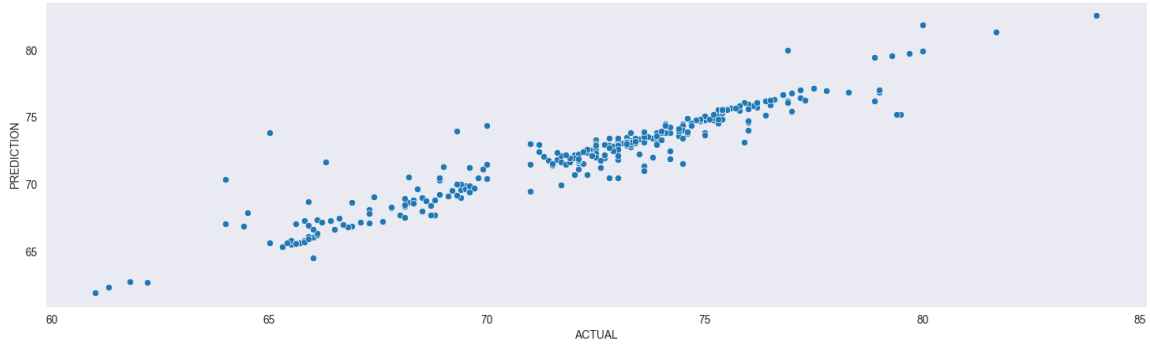
Out[72]:

	ACTUAL	PREDICTION
0	77.8	76.954
1	77.5	77.132
2	77.2	77.020
3	76.9	76.224
4	76.6	76.306

Scatter Plot

In [73]:

```
plt.figure(figsize=(18,5))
sns.scatterplot(data=OUTPUT,x="ACTUAL",y="PREDICTION");
```



In [75]:

```
model.score(x,y)*100
```

Out[75]:

89.66741791474415

Conclusion

This project helps to suggest a country which area should be given importance in order to efficiently improve the life expectancy of its population. Life Expectancy have negative correlation with eating habits, lifestyle, exercise, smoking, drinking alcohol etc. The impact of schooling on the lifespan of humans. Life Expectancy have negative relationship with drinking alcohol populated countries tend to have lower life expectancy. Impact of Immunization coverage on life Expectancy.

In [ ]:

```

```