

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA, INDORE



Project Report

On

Life Expectancy Data

Submitted By

- 1. Divyam Pithawa [19100BTCSAII05259]**
- 2. Mradul Agrawal [19100BTCSAII05269]**
- 3. Rudraksh Thorat [19100BTCSAII05280]**
- 4. Sarthak Nahar [19100BTCSAII05283]**
- 5. Shivam Sharma [19100BTCSAII05286]**

Under the Guidance of

Mr. Sachin Garg

SHRI VAISHNAV INSTITUTE OF INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JULY 2021-JANUARY 2021

INTRODUCTION

Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that affect of immunization and human development index was not taken into account in the past. Also, some of the past research was done considering multiple linear regression based on data set of one year for all the countries. Hence, this gives motivation to resolve both the factors stated previously by formulating 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 project relies on accuracy of data. 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. 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 countries for further analysis. The individual data files have been merged together into a single data-set. On initial visual inspection of the data showed some missing values. As the data-sets were from WHO, we found no evident errors. Missing data was handled in R software by using Missmap command. The result indicated that most

of the missing data was for population, Hepatitis B and GDP. The missing data were from less known countries like Vanuatu, Tonga, Togo, Cabo Verde etc. Finding all data for these countries was difficult and hence, it was decided that we exclude these countries from the final model data-set. The final merged file(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.

Acknowledgements

The data was collected from WHO and United Nations website with the help of Deeksha Russell and Duan Wang.

Public Dataset available at below website

<https://www.kaggle.com/kumarajarshi/life-expectancy-who>

Tools Used

Jupyter Notebook.

DataSet (Screenshot) : -

Life Expectancy Data.csv - Excel

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard Font Alignment Number Styles Cells Editing

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

H1 percentage expenditure

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Country	Year	Status	Life expectancy	Adult Mortality	Infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphth
1	Afghanistan	2015	Developing	65	263	62	0.01	71.27962362	65	1154	19.1	83	6	8.16	
2	Afghanistan	2014	Developing	59.9	271	64	0.01	73.52358168	62	492	18.6	86	58	8.18	
3	Afghanistan	2013	Developing	59.9	268	66	0.01	73.21924272	64	430	18.1	89	62	8.13	
4	Afghanistan	2012	Developing	59.5	272	69	0.01	78.1842153	67	2787	17.6	93	67	8.52	
5	Afghanistan	2011	Developing	59.2	275	71	0.01	70.097108703	68	3013	17.2	97	68	7.87	
6	Afghanistan	2010	Developing	58.8	279	74	0.01	79.67936736	66	1989	16.7	102	66	9.2	
7	Afghanistan	2009	Developing	58.6	281	77	0.01	56.76221682	63	2861	16.2	106	63	9.42	
8	Afghanistan	2008	Developing	58.1	287	80	0.03	25.87392536	64	1599	15.7	110	64	8.33	
9	Afghanistan	2007	Developing	57.5	295	82	0.02	10.91015598	63	1141	15.2	113	63	6.73	
10	Afghanistan	2006	Developing	57.3	295	84	0.03	17.17151751	64	1990	14.7	116	58	7.43	
11	Afghanistan	2005	Developing	57.3	291	85	0.02	1.388647732	66	1296	14.2	118	58	8.7	
12	Afghanistan	2004	Developing	57	293	87	0.02	15.29606643	67	466	13.8	120	5	8.79	
13	Afghanistan	2003	Developing	56.7	295	87	0.01	11.08905273	65	798	13.4	122	41	8.82	
14	Afghanistan	2002	Developing	56.2	3	88	0.01	16.88735091	64	2486	13	122	36	7.76	
15	Afghanistan	2001	Developing	55.3	316	88	0.01	10.5747282	63	8762	12.6	122	35	7.8	
16	Afghanistan	2000	Developing	54.8	321	88	0.01	10.42496	62	6532	12.2	122	24	8.2	
17	Albania	2015	Developing	77.8	74	0	4.6	364.9752287	99	0	58	0	99	6	
18	Albania	2014	Developing	77.5	8	0	4.51	428.7490668	98	0	57.2	1	98	5.88	
19	Albania	2013	Developing	77.2	84	0	4.76	430.8769785	99	0	56.5	1	99	5.66	
20	Albania	2012	Developing	76.0	86	0	5.14	413.423563	99	0	55.8	1	99	5.50	

Ready Average: 738.2512955 Count: 2939 Sum: 2168982.306

Type here to search

Life Expectancy Data.csv - Excel

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Clipboard Font Alignment Number Styles Cells Editing

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

O1 Diphtheria

	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	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	Schooling
1	65	1154	19.1	83	6	8.16	65	0.1	584.2592	33736494	17.2	17.3	0.479	10.1
2	62	492	18.6	86	58	8.18	62	0.1	612.6965	327582	17.5	17.5	0.476	10
3	64	430	18.1	89	62	8.13	64	0.1	631.745	31731688	17.7	17.7	0.47	9.9
4	67	2787	17.6	93	67	8.52	67	0.1	669.959	3696958	17.9	18	0.463	9.8
5	68	3013	17.2	97	68	7.87	68	0.1	63.53723	2978599	18.2	18.2	0.454	9.5
6	66	1989	16.7	102	66	9.2	66	0.1	553.3289	2883167	18.4	18.4	0.448	9.2
7	63	2861	16.2	106	63	9.42	63	0.1	445.8933	284331	18.6	18.7	0.434	8.9
8	64	1599	15.7	110	64	8.33	64	0.1	373.3611	2729431	18.8	18.9	0.433	8.7
9	63	1141	15.2	113	63	6.73	63	0.1	369.8358	26616792	19	19.1	0.415	8.4
10	64	1990	14.7	116	58	7.43	58	0.1	272.5638	2589345	19.2	19.3	0.405	8.1
11	66	1296	14.2	118	58	8.7	58	0.1	25.29413	257798	19.3	19.5	0.396	7.9
12	67	466	13.8	120	5	8.79	5	0.1	219.1414	24118979	19.5	19.7	0.381	6.8
13	65	798	13.4	122	41	8.82	41	0.1	198.7285	2364851	19.7	19.9	0.373	6.5
14	64	2486	13	122	36	7.76	36	0.1	187.846	21979923	19.9	2.2	0.341	6.2
15	63	8762	12.6	122	35	7.8	33	0.1	117.497	2966463	2.1	2.4	0.34	5.9
16	62	6532	12.2	122	24	8.2	24	0.1	114.56	293756	2.3	2.5	0.338	5.5
17	99	0	58	0	99	6	99	0.1	3954.228	28873	1.2	1.3	0.762	14.2
18	98	0	57.2	1	98	5.88	98	0.1	4575.764	288914	1.2	1.3	0.761	14.2
19	99	0	56.5	1	99	5.66	99	0.1	4414.723	289592	1.3	1.4	0.759	14.2
20	99	0	55.0	1	99	5.50	99	0.1	4243.514	28443	1.3	1.4	0.753	14.2

Edit Average: 82.32408359 Count: 2920 Sum: 240304

Type here to search

About Columns of Dataset: -

1. Country - Country
2. Year - Year
3. Status - Developed or Developing status
4. Life expectancy - Life Expectancy in age
5. Adult Mortality - Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)
6. Infant Deaths - Number of Infant Deaths per 1000 population
7. Alcohol - Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)
8. Percentage Expenditure - Expenditure on health as a percentage of Gross Domestic Product per capita(%)
9. Hepatitis B - Hepatitis B (HepB) immunization coverage among 1-year-olds (%)
- 10.Measles - Number of reported cases per 1000 population
- 11.BMI - Average Body Mass Index of entire population
- 12.Under-Five Deaths - Number of under-five deaths per 1000 population
- 13.Polio - Polio (Pol3) immunization coverage among 1-year-olds (%)
- 14.Total Expenditure - General government expenditure on health as a percentage of total government expenditure (%)
- 15.Diphtheria - Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)
- 16.HIV/AIDS - Deaths per 1 000 live births HIV/AIDS (0-4 years)
- 17.GDP - Gross Domestic Product per capita (in USD)
- 18.Population - Population of the country
- 19.Thinness 1-19 Years - Prevalence of thinness among children and adolescents for Age 10 to 19 (%)
- 20.Thinness 5-9 Years - Prevalence of thinness among children for Age 5 to 9(%)
- 21.Income Composition of Resources - Human Development Index in terms of income composition of resources (index ranging from 0 to 1)
- 22.Schooling - Number of years of Schooling(years)

Problem Statement are: -

Que 1. Finding top 10 Countries with maximum number of Life Expectancy.

Que 2. Find the Correlation between GDP and Life Expectancy.

Que 3. Which Disease affect Life Expectancy the most.

Que 4. Which Country's Life Expectancy increased the most in 15 years.

Que 5. Top Five Country with Highest Adult Mortality.

Que 6. Draw a Graph b/w Income Composition and Population.

Que 7. In which Year the Total Expenditure is Highest in India.

Que 8. What are the predicting variables that actually affecting the Life Expectancy?

Que 9. Should a country having a lower life expectancy value (<65) increase its healthcare expenditure in order to improve its average lifespan?

Que 10. How does Infant and Adult mortality rates affect Life Expectancy?

Que 11. Does Life Expectancy have positive or negative correlation with Status of the Country, drinking alcohol, BMI, GDP etc.

Que 12. What is the impact of schooling on the lifespan of humans?

Que 13. Do densely populated countries tend to have lower life expectancy?

Que 14. Draw a Line Graph of Year wise GDP of India.

Que 15. Which Country infant die more often.

Que 16. In which Year the Under Five Death is Highest.

Que 17. Draw a Graph between Year and Schooling.

Que 18. Country which consumed most alcohol.

Que 19. Countries whose government shows awareness about their health condition (more expenditure on health issues).

Que 20. In which year India shows healthiest Body Mass Index (BMI).

Que 21. Top 5 countries whose infant mortality rate is lowest.

Que 22. Countries whose literacy rate is best.

Que 23. Countries with lowest literacy rate.

Que 24. In which Year the Total Expenditure is Lowest in India.

Que 25. In which year the Income Composition of Resources is Highest in India.

Que 26. How much the GDP has increased in the last 15 years in India.

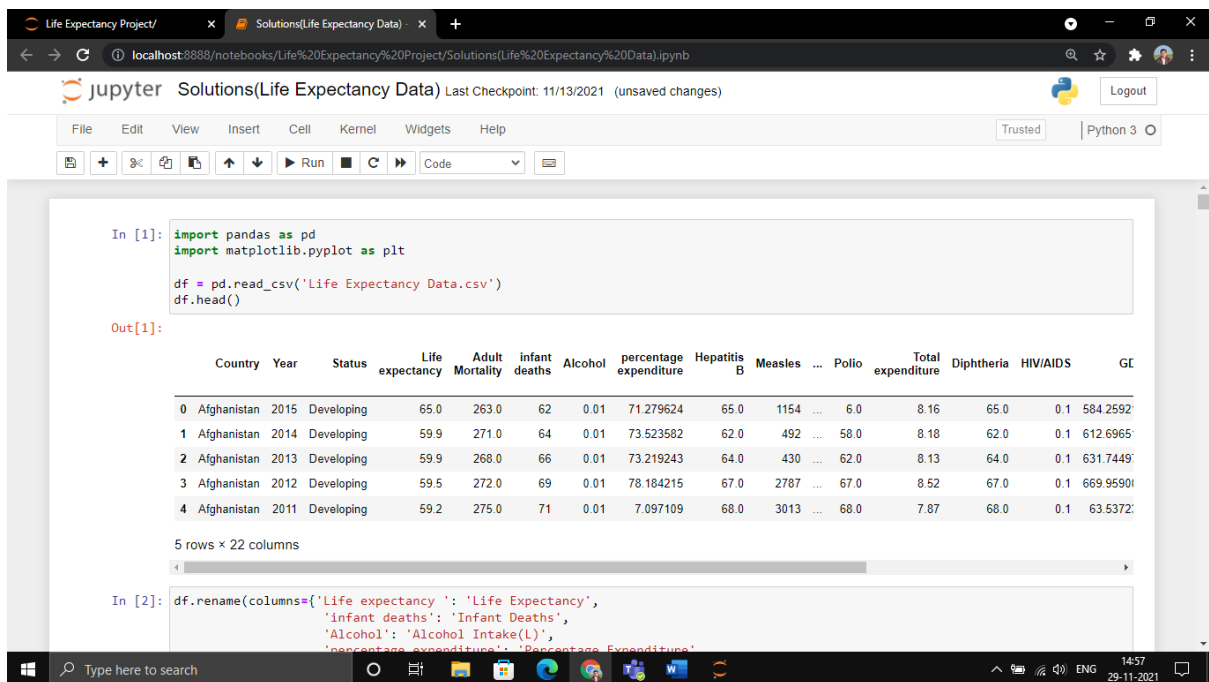
Que 27. How many countries are Developed till the year 2015.

Que 28. List of Countries which has Developed in the last 15 Years.

Que 29. Draw a Graph of Year wise Schooling in India.

Que 30. Draw a Graph of Life Expectancy of different Countries.

Step 1. Importing Data set to a Python Code



The screenshot shows a Jupyter Notebook interface with the title 'Solutions(Life Expectancy Data)'. The code in the first cell imports pandas and matplotlib, reads a CSV file named 'Life Expectancy Data.csv', and displays the first five rows of the resulting DataFrame. The output shows a table with 22 columns: Country, Year, Status, Life expectancy, Adult Mortality, infant deaths, Alcohol, percentage expenditure, Hepatitis B, Measles, Polio, Total expenditure, Diphtheria, HIV/AIDS, and GC. The data is for Afghanistan from 2011 to 2015.

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('Life Expectancy Data.csv')
df.head()
```

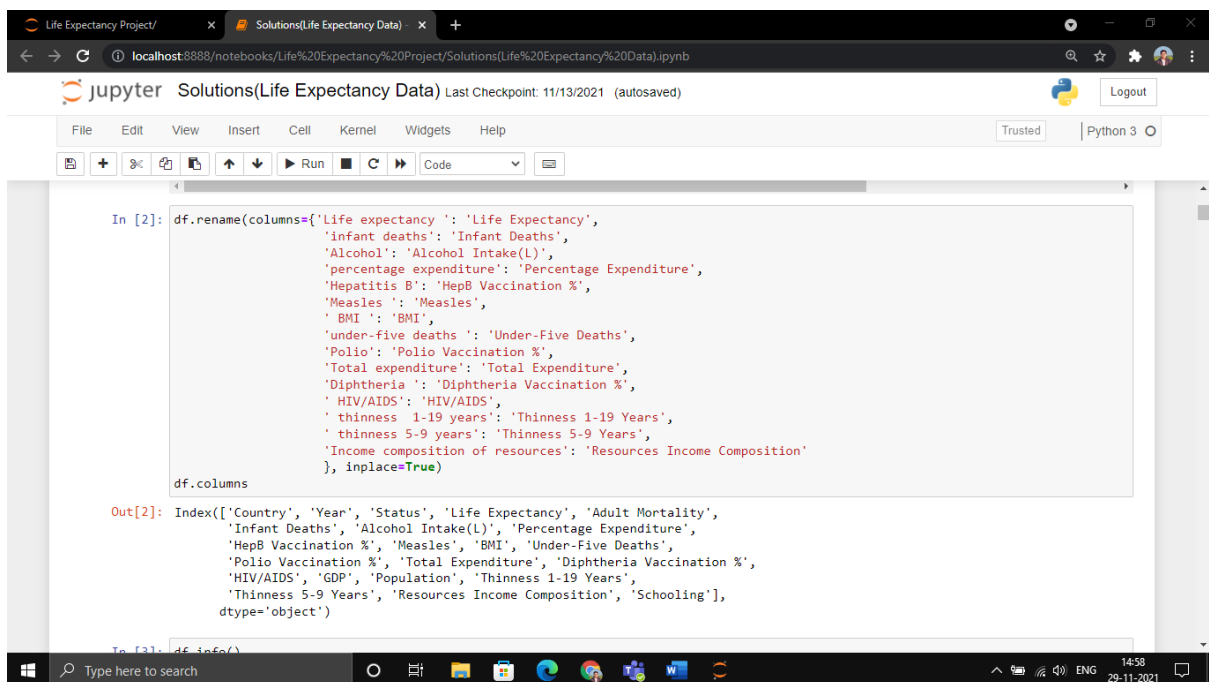
Out[1]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	Polio	Total expenditure	Diphtheria	HIV/AIDS	GC
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	6.0	8.16	65.0	0.1	584.2592
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	58.0	8.18	62.0	0.1	612.6965
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	62.0	8.13	64.0	0.1	631.7449
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	67.0	8.52	67.0	0.1	669.9590
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	68.0	7.87	68.0	0.1	63.5372

5 rows x 22 columns

```
In [2]: df.rename(columns={'Life expectancy': 'Life Expectancy',
                        'infant deaths': 'Infant Deaths',
                        'Alcohol': 'Alcohol Intake(L)',
                        'percentage expenditure': 'Percentage Expenditure'
                        })
```

Step 2. Renaming Columns of a Data Set



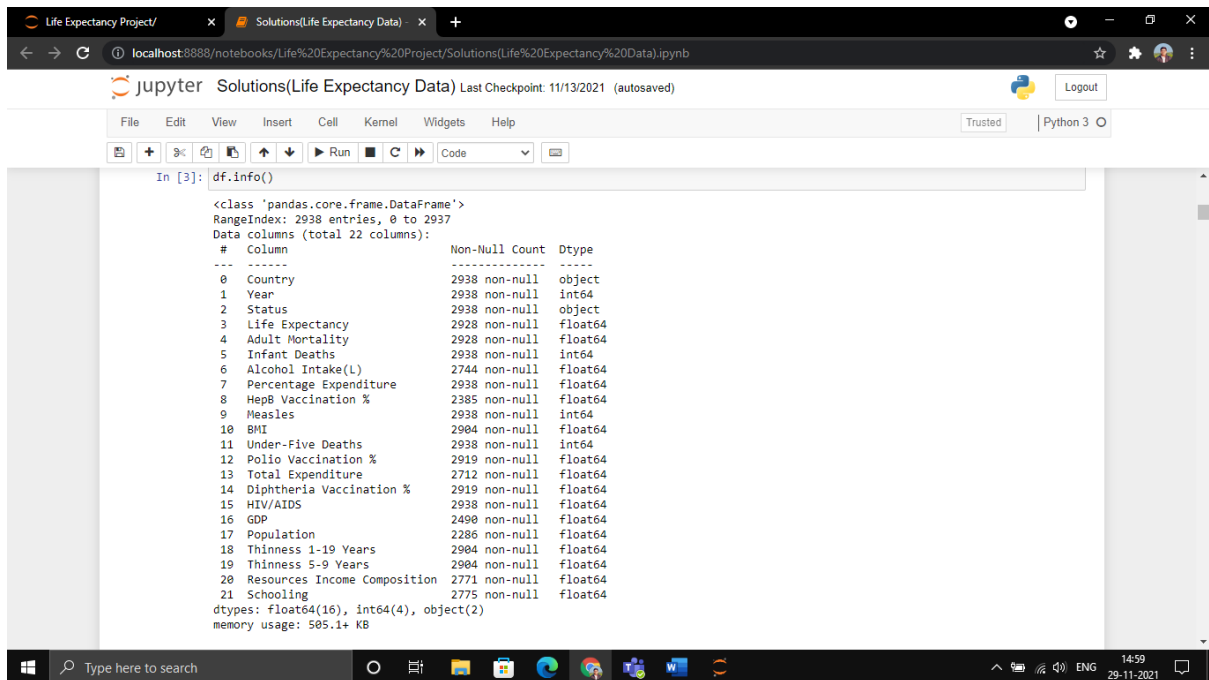
The screenshot shows the same Jupyter Notebook with the second cell. The code renames the columns of the DataFrame to be more descriptive. The output shows the new column names for the first five rows of the DataFrame.

```
In [2]: df.rename(columns={'Life expectancy': 'Life Expectancy',
                        'infant deaths': 'Infant Deaths',
                        'Alcohol': 'Alcohol Intake(L)',
                        'percentage expenditure': 'Percentage Expenditure',
                        'Hepatitis B': 'HepB Vaccination %',
                        'Measles': 'Measles',
                        'BMI': 'BMI',
                        'under-five deaths': 'Under-Five Deaths',
                        'Polio': 'Polio Vaccination %',
                        'Total expenditure': 'Total Expenditure',
                        'Diphtheria': 'Diphtheria Vaccination %',
                        'HIV/AIDS': 'HIV/AIDS',
                        'thinness 1-19 years': 'Thinness 1-19 Years',
                        'thinness 5-9 years': 'Thinness 5-9 Years',
                        'Income composition of resources': 'Resources Income Composition'
                        }, inplace=True)

df.columns
```

Out[2]: Index(['Country', 'Year', 'Status', 'Life Expectancy', 'Adult Mortality', 'Infant Deaths', 'Alcohol Intake(L)', 'Percentage Expenditure', 'HepB Vaccination %', 'Measles', 'BMI', 'Under-Five Deaths', 'Polio Vaccination %', 'Total Expenditure', 'Diphtheria Vaccination %', 'HIV/AIDS', 'GDP', 'Population', 'Thinness 1-19 Years', 'Thinness 5-9 Years', 'Resources Income Composition', 'Schooling'], dtype='object')

Step 3. Finding info about Data Set



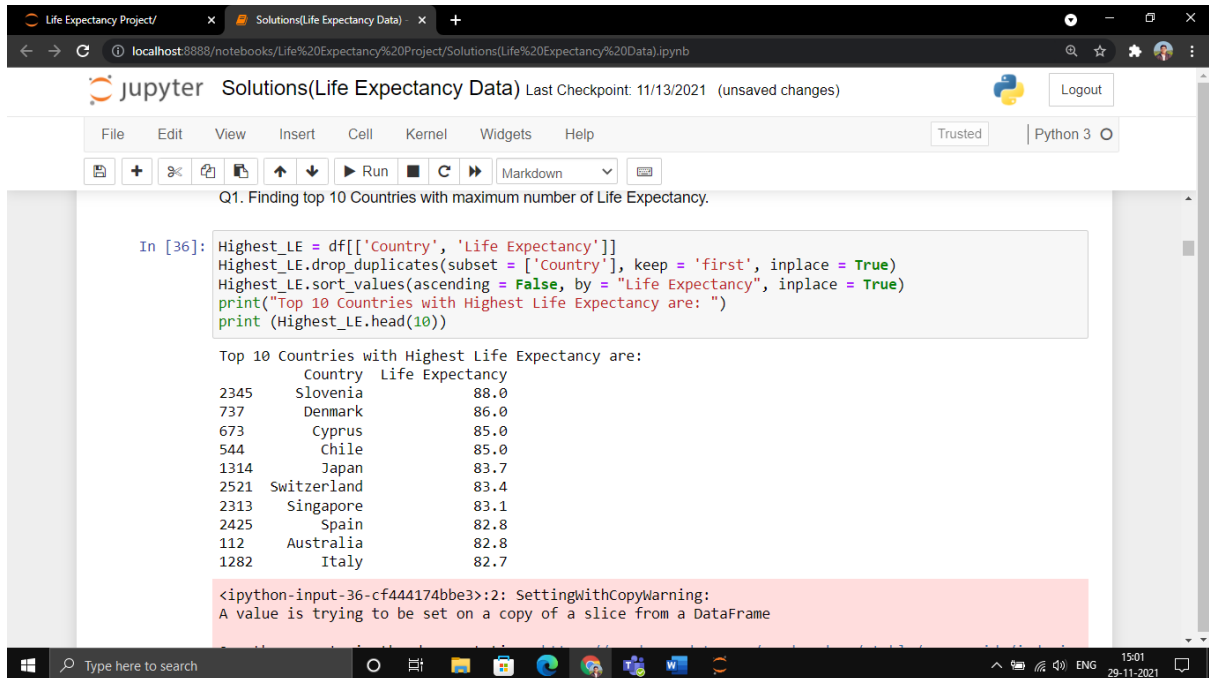
The screenshot shows a Jupyter Notebook interface with a single code cell. The code cell contains the command `df.info()`. The output of this command is displayed below the code cell. It shows the class of the data frame, the range of the index, and a summary of the columns including their names, non-null counts, and data types. The data types are listed as `float64(16)`, `int64(4)`, and `object(2)`. The memory usage is reported as 505.1+ KB.

```
In [3]: 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 Intake(L)                         2744 non-null   float64
7   Percentage Expenditure                    2938 non-null   float64
8   HepB Vaccination %                       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 Vaccination %                      2919 non-null   float64
13  Total Expenditure                         2712 non-null   float64
14  Diphtheria Vaccination %                 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  Resources Income Composition              2771 non-null   float64
21  Schooling                                2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

Solutions of Problem Statement: -

Que 1. Finding top 10 Countries with maximum number of Life Expectancy.



The screenshot shows a Jupyter Notebook interface with the title "Solutions(Life Expectancy Data)". The notebook contains a single code cell with the following Python code:

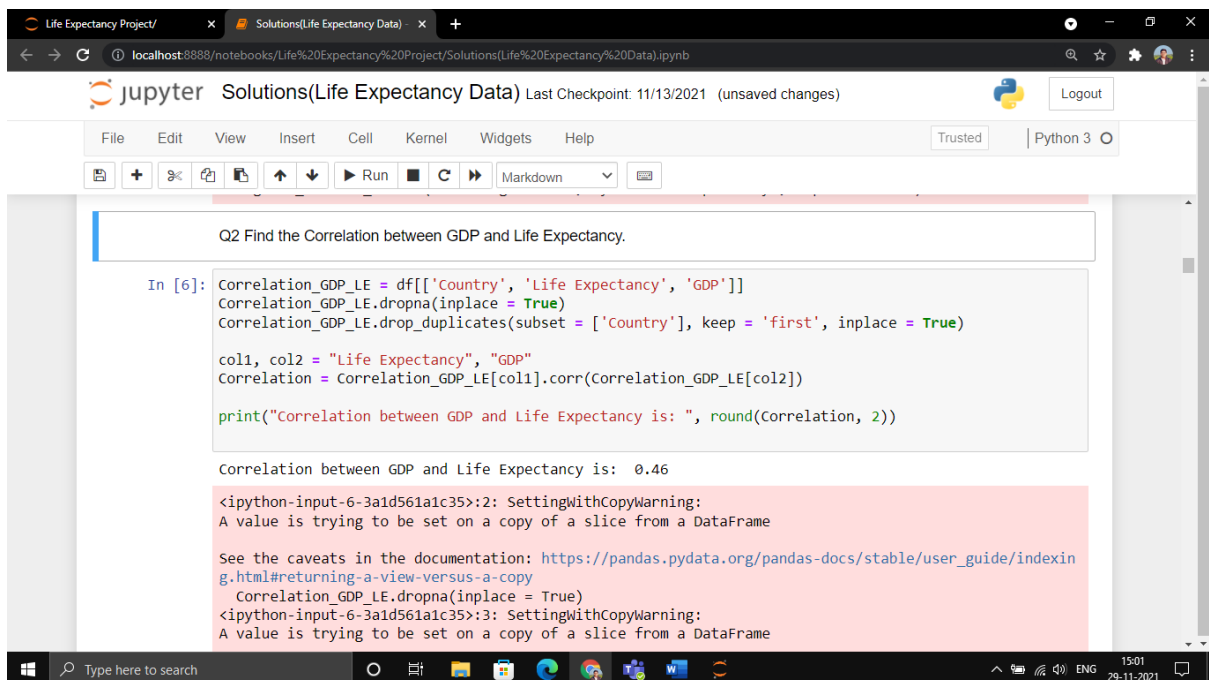
```
In [36]: Highest_LE = df[['Country', 'Life Expectancy']]
Highest_LE.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
Highest_LE.sort_values(ascending = False, by = "Life Expectancy", inplace = True)
print("Top 10 Countries with Highest Life Expectancy are: ")
print (Highest_LE.head(10))
```

The output of the code is a table showing the top 10 countries with the highest life expectancy:

	Country	Life Expectancy
2345	Slovenia	88.0
737	Denmark	86.0
673	Cyprus	85.0
544	Chile	85.0
1314	Japan	83.7
2521	Switzerland	83.4
2313	Singapore	83.1
2425	Spain	82.8
112	Australia	82.8
1282	Italy	82.7

Below the table, there is a warning message: "<ipython-input-36-cf444174bbe3>:2: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame".

Que 2. Find the Correlation between GDP and Life Expectancy.



The screenshot shows a Jupyter Notebook interface with the title "Solutions(Life Expectancy Data)". The notebook contains a single code cell with the following Python code:

```
In [6]: Correlation_GDP_LE = df[['Country', 'Life Expectancy', 'GDP']]
Correlation_GDP_LE.dropna(inplace = True)
Correlation_GDP_LE.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)

col1, col2 = "Life Expectancy", "GDP"
Correlation = Correlation_GDP_LE[col1].corr(Correlation_GDP_LE[col2])

print("Correlation between GDP and Life Expectancy is: ", round(Correlation, 2))
```

The output of the code is:

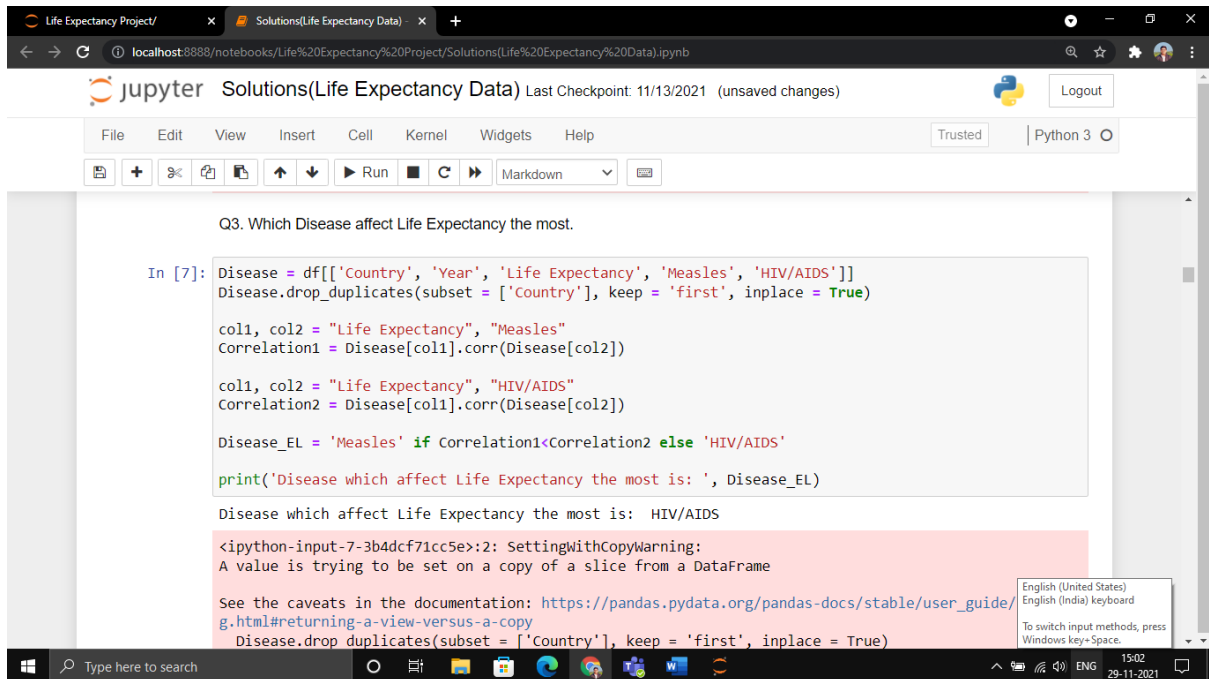
```
Correlation between GDP and Life Expectancy is: 0.46
```

Below the output, there is a warning message: "<ipython-input-6-3a1d561a1c35>:2: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame".

Below the warning message, there is a link to the pandas documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy.

Below the link, there is another warning message: "<ipython-input-6-3a1d561a1c35>:3: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame".

Que 3. Which Disease affect Life Expectancy the most.



A screenshot of a Jupyter Notebook interface. The browser tab is 'Solutions(Life Expectancy Data)'. The URL is 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook title is 'Solutions(Life Expectancy Data)' with a 'Last Checkpoint: 11/13/2021 (unsaved changes)' status. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and output viewing. The code cell contains the following Python code:

```
Q3. Which Disease affect Life Expectancy the most.

In [7]: Disease = df[['Country', 'Year', 'Life Expectancy', 'Measles', 'HIV/AIDS']]
Disease.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)

col1, col2 = "Life Expectancy", "Measles"
Correlation1 = Disease[col1].corr(Disease[col2])

col1, col2 = "Life Expectancy", "HIV/AIDS"
Correlation2 = Disease[col1].corr(Disease[col2])

Disease_EL = 'Measles' if Correlation1 < Correlation2 else 'HIV/AIDS'

print('Disease which affect Life Expectancy the most is: ', Disease_EL)

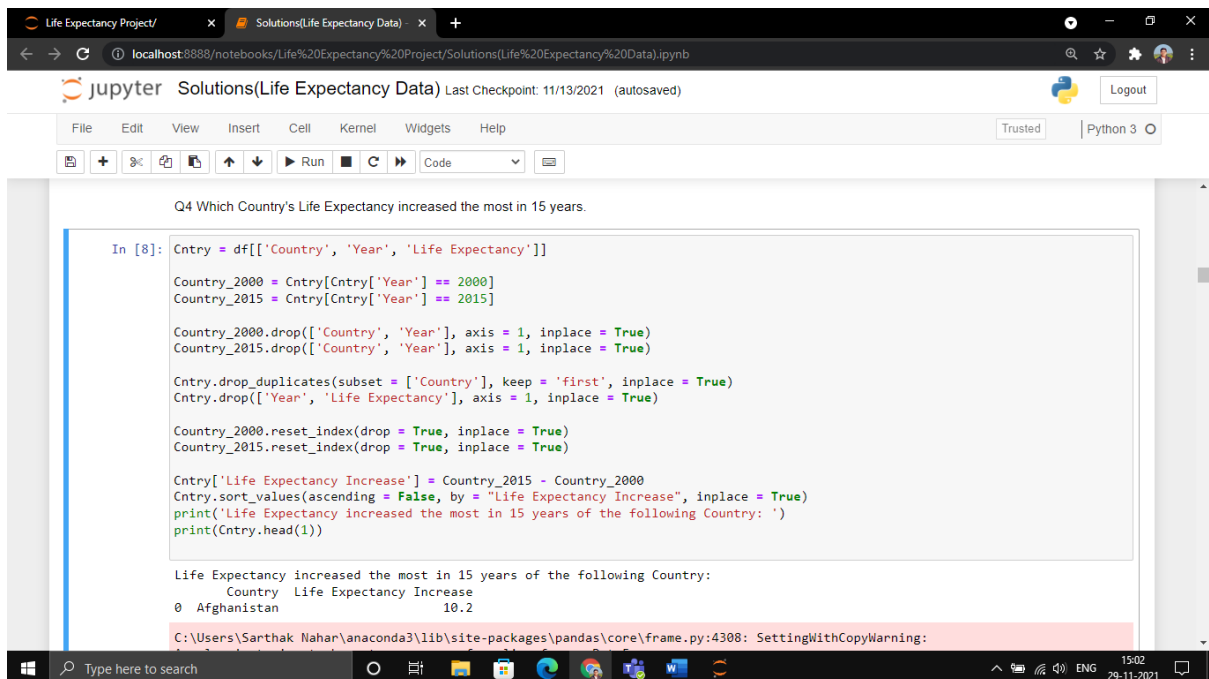
Disease which affect Life Expectancy the most is: HIV/AIDS

<ipython-input-7-3b4dcf71cc5e>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
g.html#returning-a-view-versus-a-copy
Disease.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
```

The output shows 'Disease which affect Life Expectancy the most is: HIV/AIDS'. A 'SettingWithCopyWarning' is displayed, along with a link to the pandas documentation. The Windows taskbar at the bottom shows the search bar and system clock (15:02, 29-11-2021).

Que 4. Which Country's Life Expectancy increased the most in 15 years.



A screenshot of a Jupyter Notebook interface. The browser tab is 'Solutions(Life Expectancy Data)'. The URL is 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook title is 'Solutions(Life Expectancy Data)' with a 'Last Checkpoint: 11/13/2021 (autosaved)' status. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and output viewing. The code cell contains the following Python code:

```
Q4 Which Country's Life Expectancy increased the most in 15 years.

In [8]: Cntry = df[['Country', 'Year', 'Life Expectancy']]

Country_2000 = Cntry[Cntry['Year'] == 2000]
Country_2015 = Cntry[Cntry['Year'] == 2015]

Country_2000.drop(['Country', 'Year'], axis = 1, inplace = True)
Country_2015.drop(['Country', 'Year'], axis = 1, inplace = True)

Cntry.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
Cntry.drop(['Year', 'Life Expectancy'], axis = 1, inplace = True)

Country_2000.reset_index(drop = True, inplace = True)
Country_2015.reset_index(drop = True, inplace = True)

Cntry['Life Expectancy Increase'] = Country_2015 - Country_2000
Cntry.sort_values(ascending = False, by = "Life Expectancy Increase", inplace = True)
print('Life Expectancy increased the most in 15 years of the following Country: ')
print(Cntry.head(1))

Life Expectancy increased the most in 15 years of the following Country:
   Country  Life Expectancy Increase
0  Afghanistan                    10.2

C:\Users\Sarthak Nahar\anaconda3\lib\site-packages\pandas\core\frame.py:4308: SettingWithCopyWarning:
```

The output shows 'Life Expectancy increased the most in 15 years of the following Country:' followed by a table with one row: Afghanistan with a Life Expectancy Increase of 10.2. A 'SettingWithCopyWarning' is displayed at the bottom. The Windows taskbar at the bottom shows the search bar and system clock (15:02, 29-11-2021).

Que 5. Top Five Country with Highest Adult Mortality.

```
Life Expectancy Project/ Solutions(Life Expectancy Data)
localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb
jupyter Solutions(Life Expectancy Data) Last Checkpoint: 11/13/2021 (autosaved)
Python 3

Q5. Top Five Country with Highest Adult Mortality.

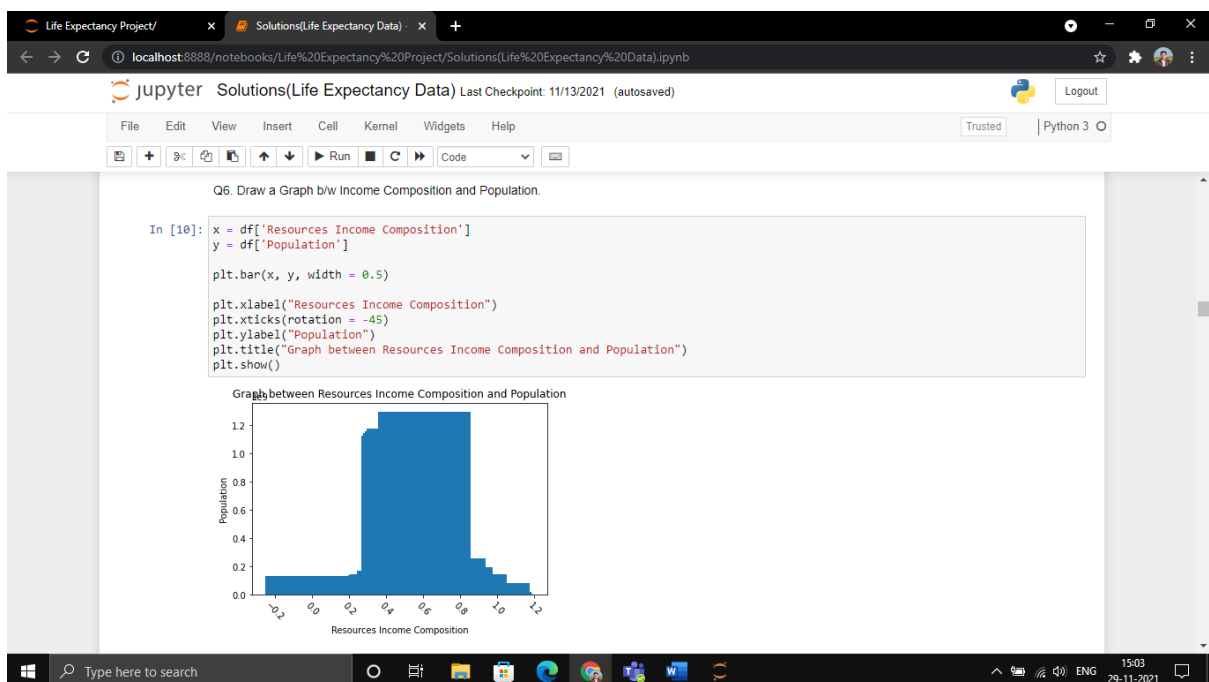
In [9]: Adult_Mortality = df[['Country', 'Adult Mortality']]
Adult_Mortality.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)

Adult_Mortality.sort_values(ascending = False, by = 'Adult Mortality', inplace = True)
print('Country with the Highest Adult Mortality is: ')
print(Adult_Mortality.head(5))

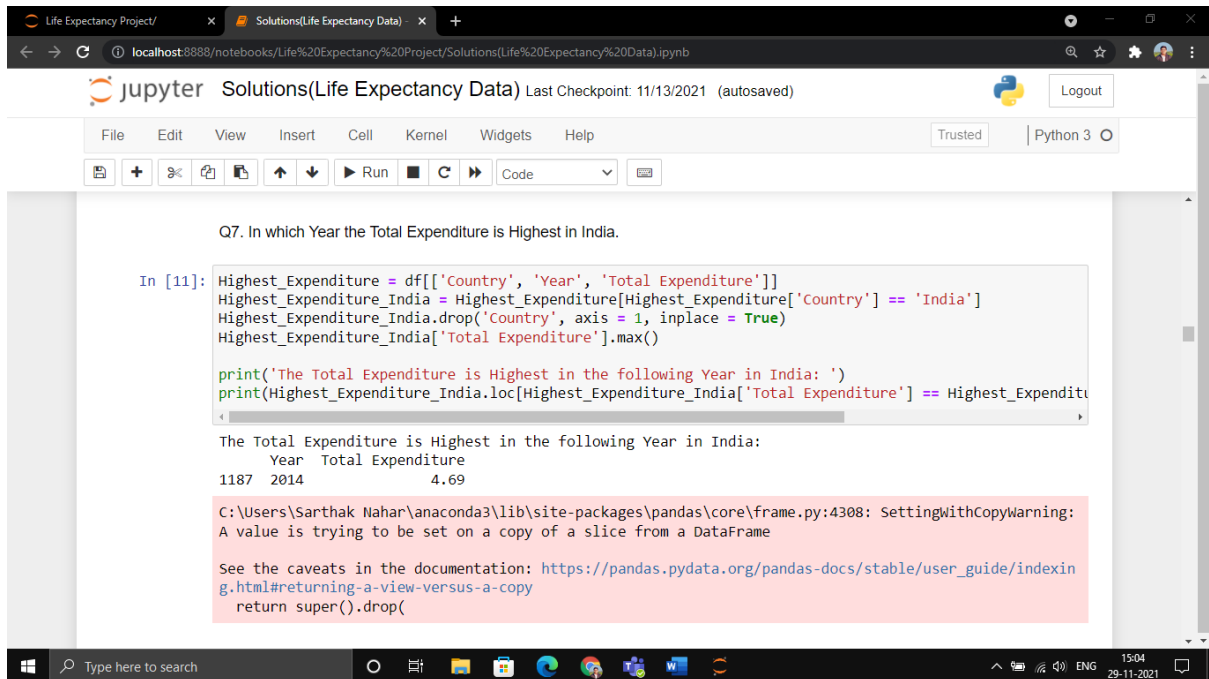
Country with the Highest Adult Mortality is:
Country Adult Mortality
1474 Lesotho 484.0
2297 Sierra Leone 413.0
432 Côte d'Ivoire 397.0
512 Central African Republic 397.0
2489 Swaziland 373.0

<ipython-input-9-2e7a8722fe26>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexin
g.html#returning-a-view-versus-a-copy
Adult_Mortality.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
```

Que 6. Draw a Graph b/w Income Composition and Population.



Que 7. In which Year the Total Expenditure is Highest in India.



```
Q7. In which Year the Total Expenditure is Highest in India.
```

```
In [11]: Highest_Expenditure = df[['Country', 'Year', 'Total Expenditure']]
Highest_Expenditure_India = Highest_Expenditure[Highest_Expenditure['Country'] == 'India']
Highest_Expenditure_India.drop('Country', axis = 1, inplace = True)
Highest_Expenditure_India['Total Expenditure'].max()

print('The Total Expenditure is Highest in the following Year in India: ')
print(Highest_Expenditure_India.loc[Highest_Expenditure_India['Total Expenditure'] == Highest_Expenditure_India['Total Expenditure'].max()])
```

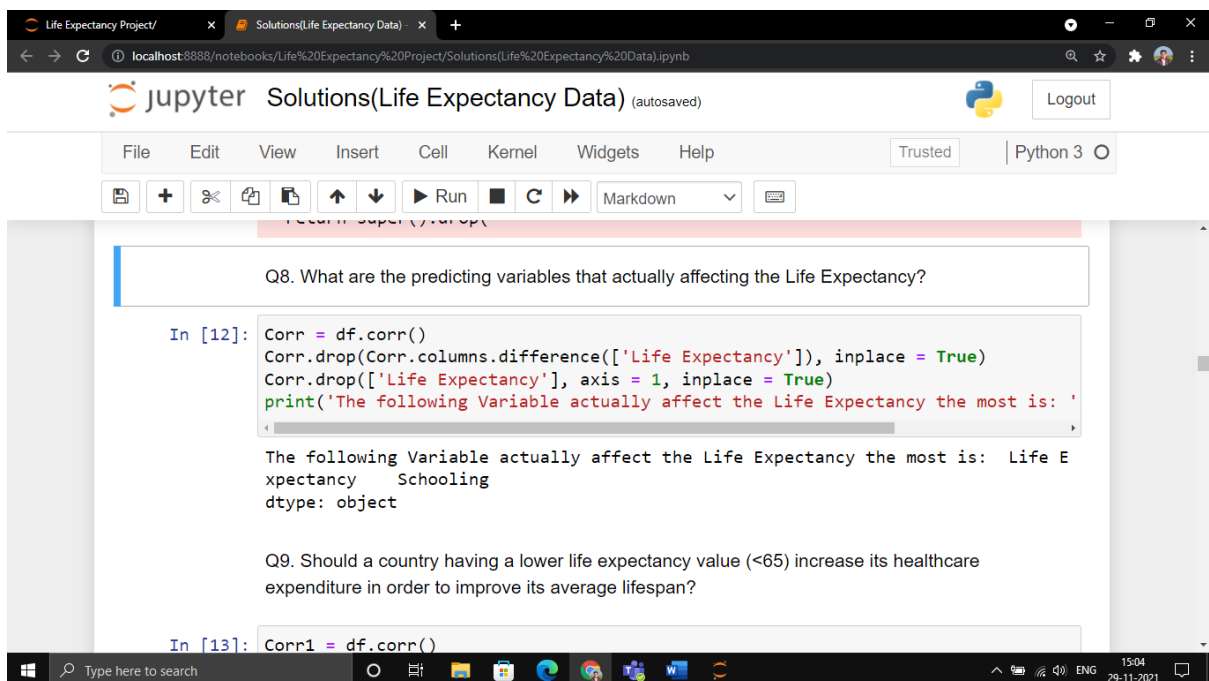
```
The Total Expenditure is Highest in the following Year in India:
Year  Total Expenditure
1187   2014              4.69
```

C:\Users\Sarthak Nahar\anaconda3\lib\site-packages\pandas\core\frame.py:4308: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
return super().drop()
```

Que 8. What are the predicting variables that actually affecting the Life Expectancy?



```
Q8. What are the predicting variables that actually affecting the Life Expectancy?
```

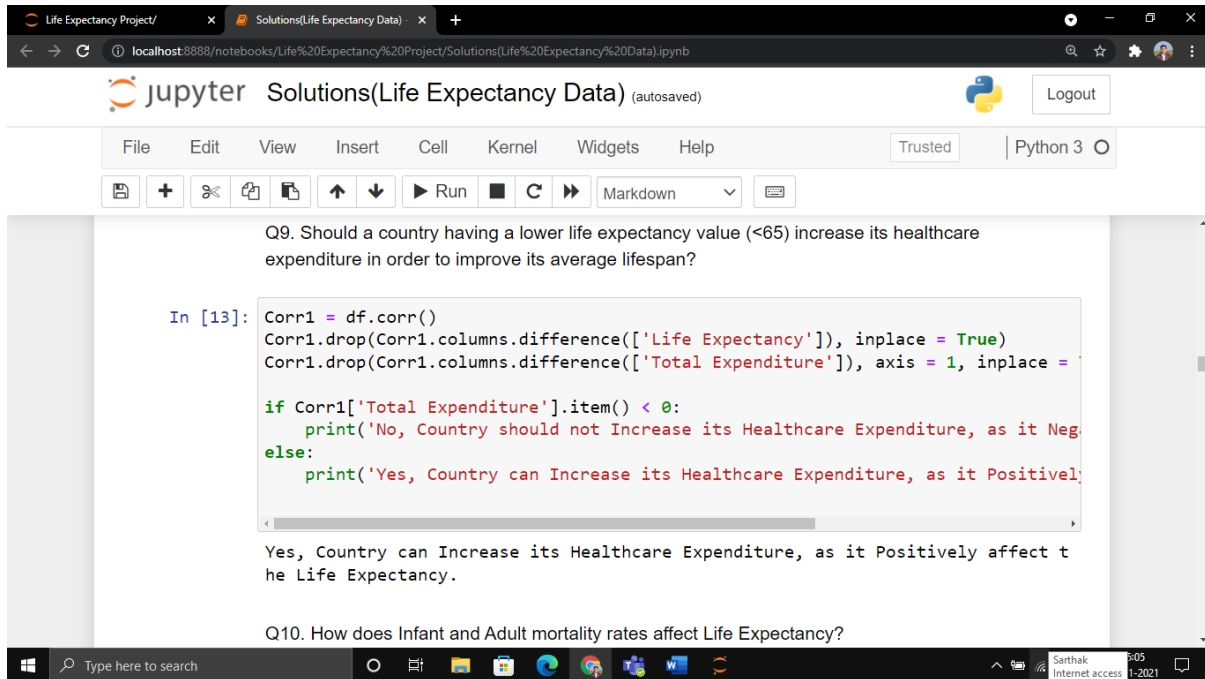
```
In [12]: Corr = df.corr()
Corr.drop(Corr.columns.difference(['Life Expectancy']), inplace = True)
Corr.drop(['Life Expectancy'], axis = 1, inplace = True)
print('The following Variable actually affect the Life Expectancy the most is: ', Corr.sort_values(by='Life Expectancy', ascending=False).iloc[0,0])
```

```
The following Variable actually affect the Life Expectancy the most is: Life Expectancy
dtype: object
```

Q9. Should a country having a lower life expectancy value (<65) increase its healthcare expenditure in order to improve its average lifespan?

```
In [13]: Corr1 = df.corr()
```

Que 9. Should a country having a lower life expectancy value (<65) increase its healthcare expenditure in order to improve its average lifespan?



The screenshot shows a Jupyter Notebook interface with the title "Solutions(Life Expectancy Data) (autosaved)". The notebook contains a code cell for Question 9. The code calculates the correlation between 'Life Expectancy' and 'Total Expenditure' and prints a message based on the result. The output shows a positive correlation, leading to the conclusion that healthcare expenditure should be increased.

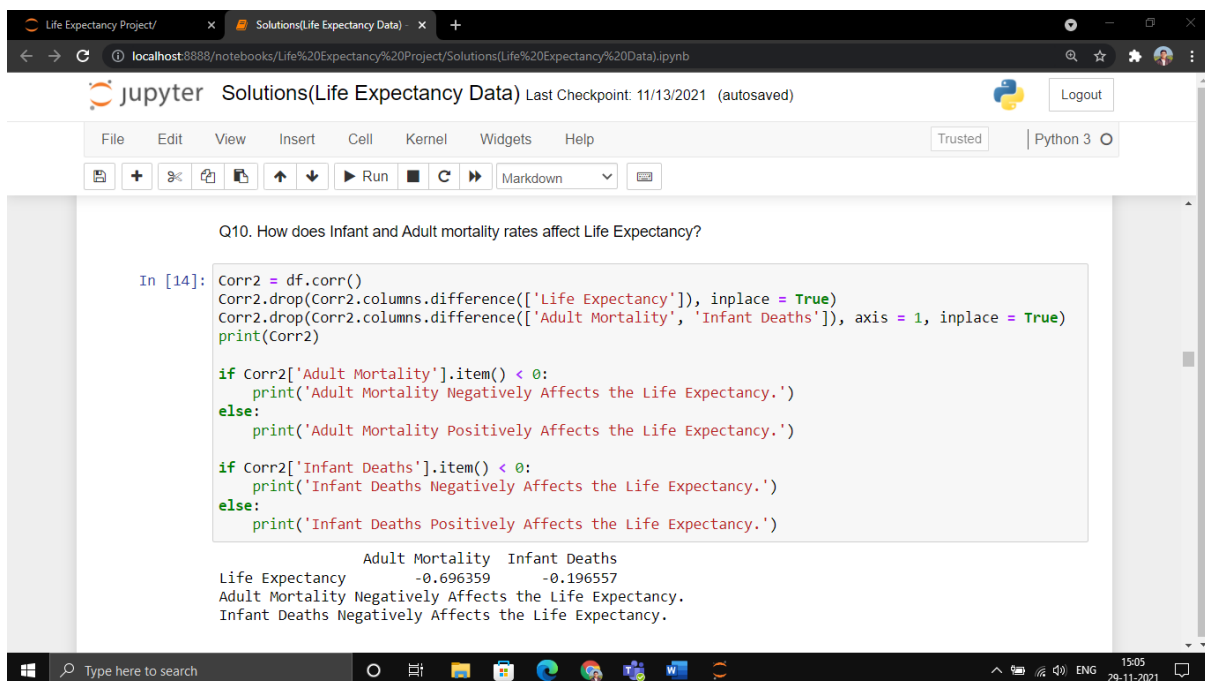
```
In [13]: Corr1 = df.corr()
Corr1.drop(Corr1.columns.difference(['Life Expectancy']), inplace = True)
Corr1.drop(Corr1.columns.difference(['Total Expenditure']), axis = 1, inplace = True)

if Corr1['Total Expenditure'].item() < 0:
    print('No, Country should not Increase its Healthcare Expenditure, as it Negatively affects the Life Expectancy.')
else:
    print('Yes, Country can Increase its Healthcare Expenditure, as it Positively affects the Life Expectancy.')

Yes, Country can Increase its Healthcare Expenditure, as it Positively affects the Life Expectancy.
```

Q9. Should a country having a lower life expectancy value (<65) increase its healthcare expenditure in order to improve its average lifespan?

Que 10. How does Infant and Adult mortality rates affect Life Expectancy?



The screenshot shows a Jupyter Notebook interface with the title "Solutions(Life Expectancy Data) Last Checkpoint: 11/13/2021 (autosaved)". The notebook contains a code cell for Question 10. The code calculates the correlation between 'Life Expectancy' and both 'Adult Mortality' and 'Infant Deaths', and prints messages based on the results. The output shows that both adult and infant mortality rates negatively affect life expectancy.

```
In [14]: Corr2 = df.corr()
Corr2.drop(Corr2.columns.difference(['Life Expectancy']), inplace = True)
Corr2.drop(Corr2.columns.difference(['Adult Mortality', 'Infant Deaths']), axis = 1, inplace = True)
print(Corr2)

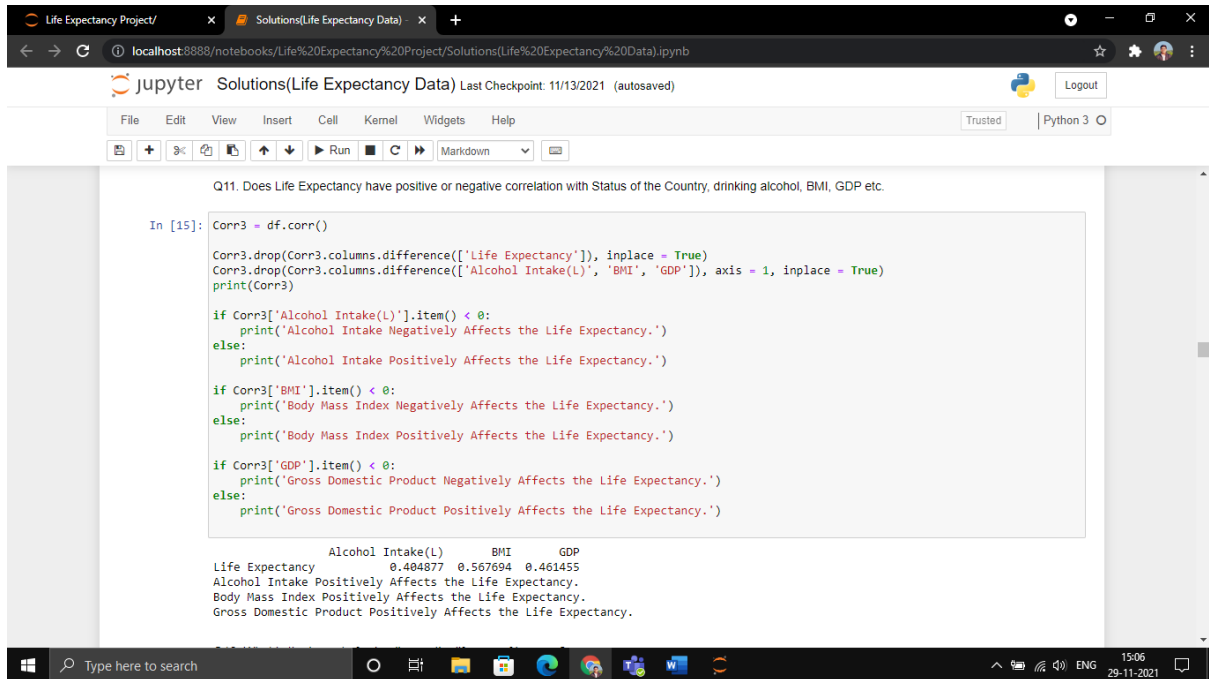
if Corr2['Adult Mortality'].item() < 0:
    print('Adult Mortality Negatively Affects the Life Expectancy.')
else:
    print('Adult Mortality Positively Affects the Life Expectancy.')

if Corr2['Infant Deaths'].item() < 0:
    print('Infant Deaths Negatively Affects the Life Expectancy.')
else:
    print('Infant Deaths Positively Affects the Life Expectancy.')

Adult Mortality  Infant Deaths
Life Expectancy -0.696359  -0.196557
Adult Mortality Negatively Affects the Life Expectancy.
Infant Deaths Negatively Affects the Life Expectancy.
```

Q10. How does Infant and Adult mortality rates affect Life Expectancy?

Que 11. Does Life Expectancy have positive or negative correlation with Status of the Country, drinking alcohol, BMI, GDP etc.



```
Q11. Does Life Expectancy have positive or negative correlation with Status of the Country, drinking alcohol, BMI, GDP etc.
```

```
In [15]: Corr3 = df.corr()

Corr3.drop(Corr3.columns.difference(['Life Expectancy']), inplace = True)
Corr3.drop(Corr3.columns.difference(['Alcohol Intake(L)', 'BMI', 'GDP']), axis = 1, inplace = True)
print(Corr3)

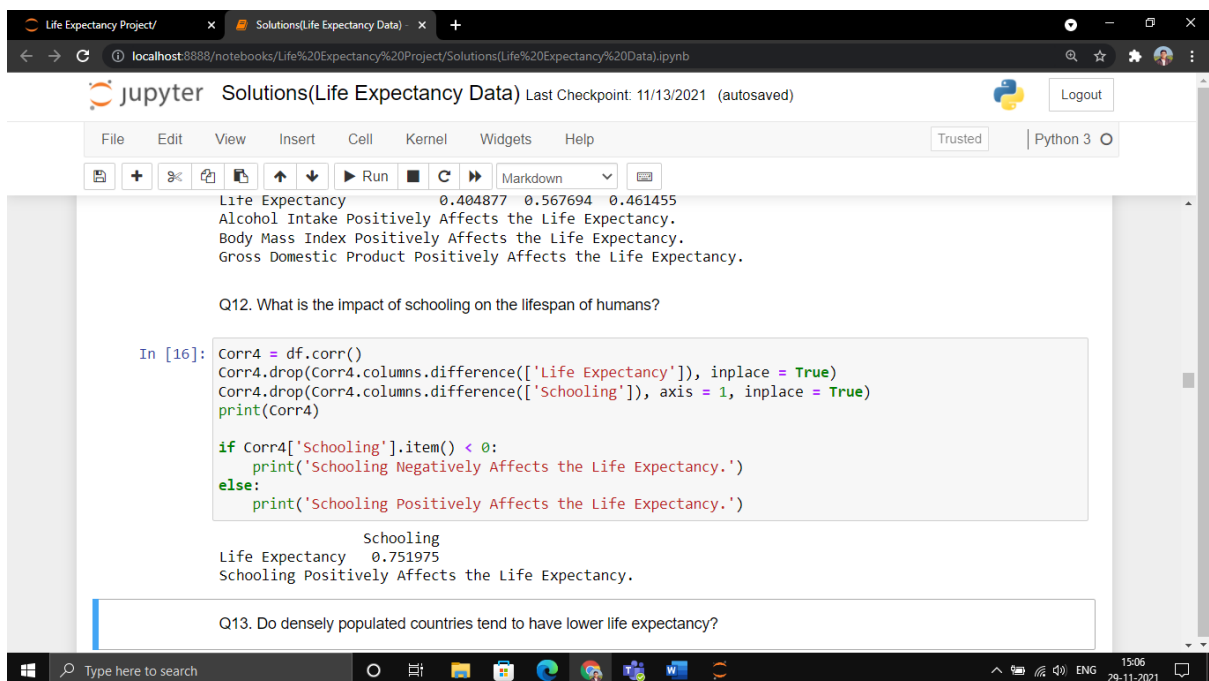
if Corr3['Alcohol Intake(L)'].item() < 0:
    print('Alcohol Intake Negatively Affects the Life Expectancy.')
else:
    print('Alcohol Intake Positively Affects the Life Expectancy.')

if Corr3['BMI'].item() < 0:
    print('Body Mass Index Negatively Affects the Life Expectancy.')
else:
    print('Body Mass Index Positively Affects the Life Expectancy.')

if Corr3['GDP'].item() < 0:
    print('Gross Domestic Product Negatively Affects the Life Expectancy.')
else:
    print('Gross Domestic Product Positively Affects the Life Expectancy.')

Alcohol Intake(L)    BMI    GDP
Life Expectancy    0.404877  0.567694  0.461455
Alcohol Intake Positively Affects the Life Expectancy.
Body Mass Index Positively Affects the Life Expectancy.
Gross Domestic Product Positively Affects the Life Expectancy.
```

Que 12. What is the impact of schooling on the lifespan of humans?



```
Life Expectancy    0.404877  0.567694  0.461455
Alcohol Intake Positively Affects the Life Expectancy.
Body Mass Index Positively Affects the Life Expectancy.
Gross Domestic Product Positively Affects the Life Expectancy.
```

```
Q12. What is the impact of schooling on the lifespan of humans?
```

```
In [16]: Corr4 = df.corr()
Corr4.drop(Corr4.columns.difference(['Life Expectancy']), inplace = True)
Corr4.drop(Corr4.columns.difference(['Schooling']), axis = 1, inplace = True)
print(Corr4)

if Corr4['Schooling'].item() < 0:
    print('Schooling Negatively Affects the Life Expectancy.')
else:
    print('Schooling Positively Affects the Life Expectancy.')

Schooling
Life Expectancy    0.751975
Schooling Positively Affects the Life Expectancy.
```

```
Q13. Do densely populated countries tend to have lower life expectancy?
```

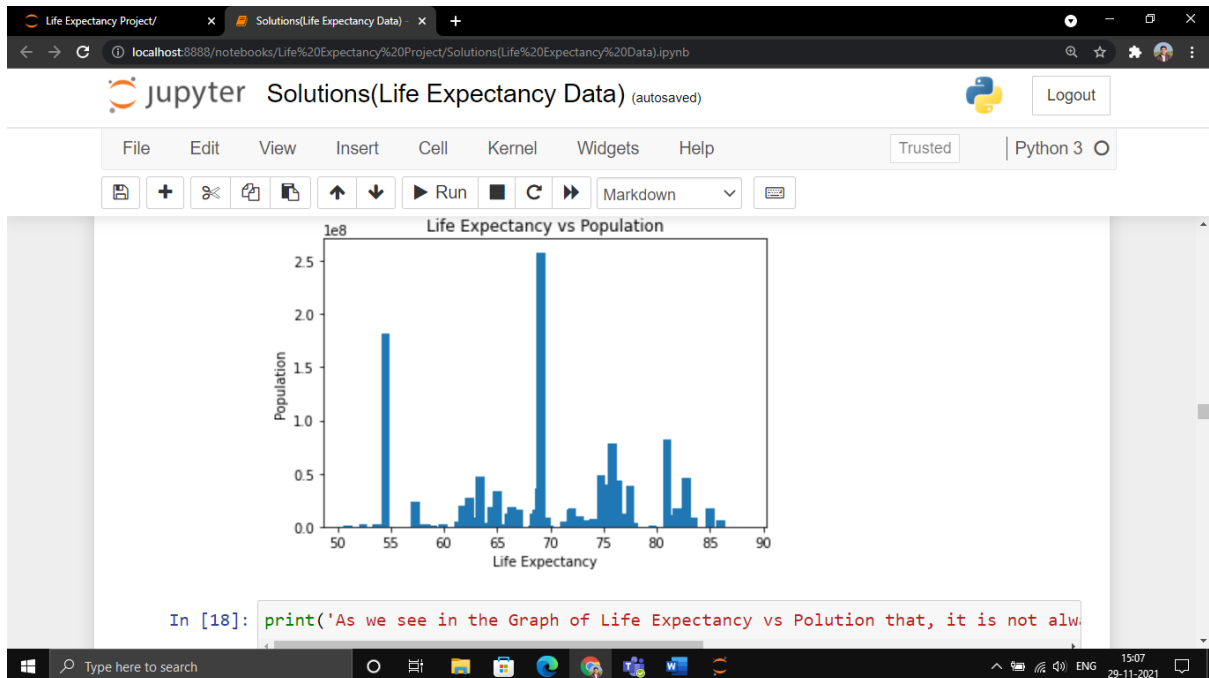
Que 13. Do densely populated countries tend to have lower life expectancy?

```
Life Expectancy Project/ Solutions(Life Expectancy Data) - x +
localhost8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb
jupyter Solutions(Life Expectancy Data) (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [17]: df_Country = df[['Country', 'Life Expectancy', 'Population']]
df_Country.dropna(inplace = True)
df_Country.drop_duplicates(subset = 'Country', keep = 'first', inplace = True)

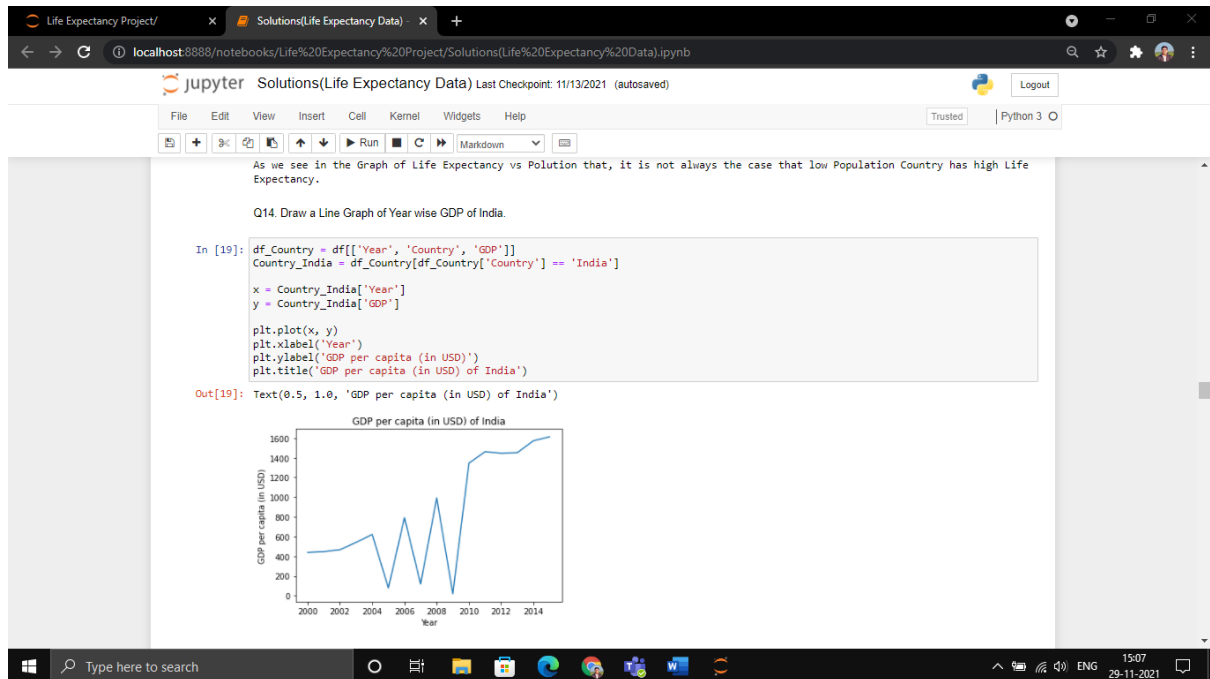
x = df_Country['Life Expectancy']
y = df_Country['Population']

plt.bar(x, y)
plt.xlabel('Life Expectancy')
plt.ylabel('Population')
plt.title('Life Expectancy vs Population')

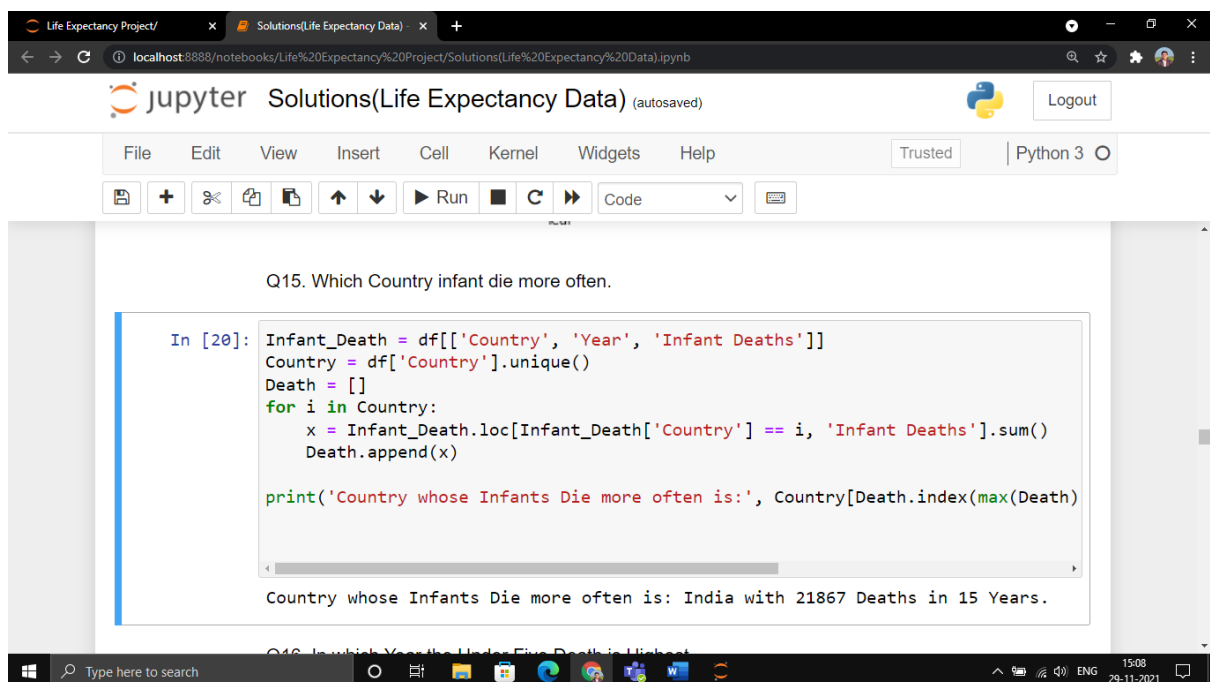
<ipython-input-17-aa8d38d8a6cf>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
```



Que 14. Draw a Line Graph of Year wise GDP of India.



Que 15. Which Country infant die more often.



Que 16. In which Year the Under Five Death is Highest.

Life Expectancy Project/ Solutions(Life Expectancy Data) - x +

localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb

jupyter Solutions(Life Expectancy Data) (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Country whose Infants Die more often is: India with 21867 Deaths in 15 Years.

Q16. In which Year the Under Five Death is Highest.

```
In [21]: Infant_Death = df[['Country', 'Year', 'Infant Deaths']]
Year = df['Year'].unique()
Y = []
for i in Year:
    x = Infant_Death.loc[Infant_Death['Year'] == i, 'Infant Deaths'].sum()
    Y.append(x)

print('Year in which the Infants Die most is:', Year[Y.index(max(Y))])
```

Year in which the Infants Die most is: 2000

Q17. Draw a Graph between Year and Schooling.

Type here to search 15:09 29-11-2021

Que 17. Draw a Graph between Year and Schooling.

Life Expectancy Project/ Solutions(Life Expectancy Data) - x +

localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb

jupyter Solutions(Life Expectancy Data) Last Checkpoint: 11/13/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

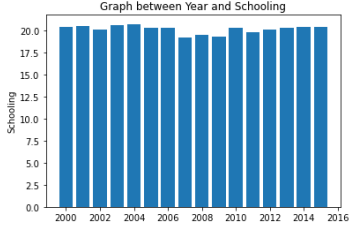
Q17. Draw a Graph between Year and Schooling.

```
In [22]: x = df['Year']
y = df['Schooling']

plt.bar(x, y)

plt.xlabel("Year")
plt.ylabel("Schooling")
plt.title("Graph between Year and Schooling")
plt.show()
```

Graph between Year and Schooling

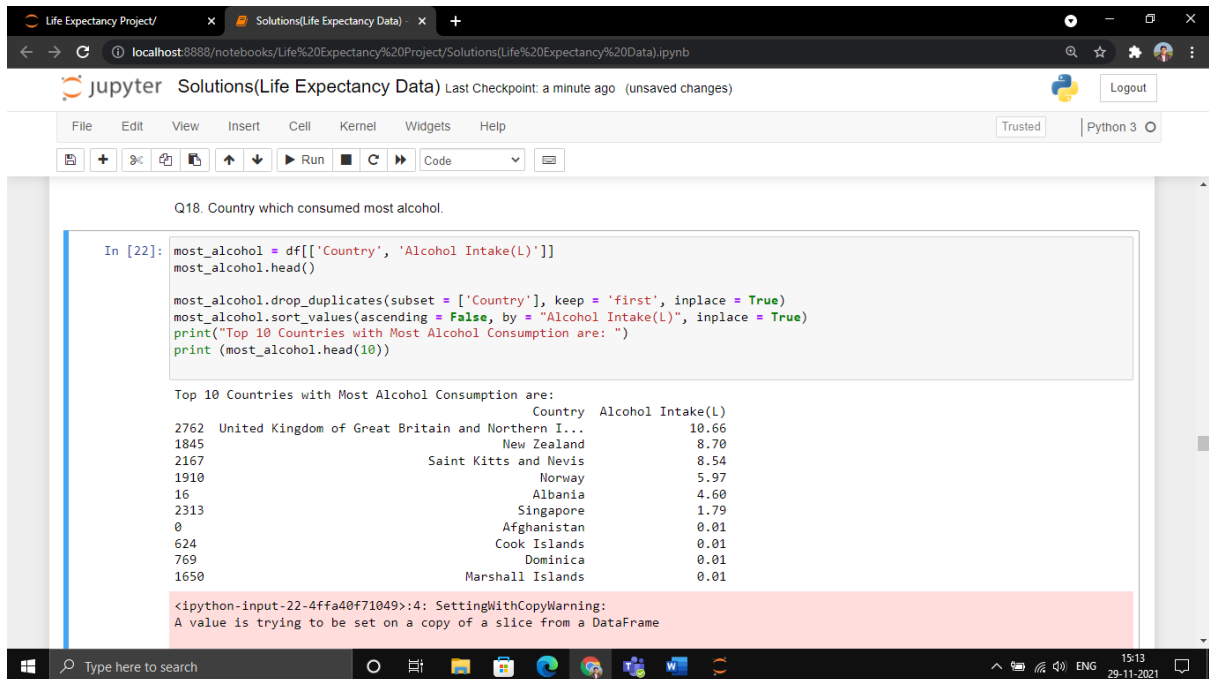


Year	Schooling
2000	19.5
2001	19.5
2002	19.5
2003	19.5
2004	19.5
2005	19.5
2006	19.5
2007	19.5
2008	19.5
2009	19.5
2010	19.5
2011	19.5
2012	19.5
2013	19.5
2014	19.5
2015	19.5
2016	19.5

Q18. Country which consumed most alcohol.

Type here to search 15:09 29-11-2021

Que 18. Country which consumed most alcohol.



A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook title is 'Solutions(Life Expectancy Data)'. The code cell contains the following Python code:

```
In [22]: most_alcohol = df[['Country', 'Alcohol Intake(L)']]
most_alcohol.head()

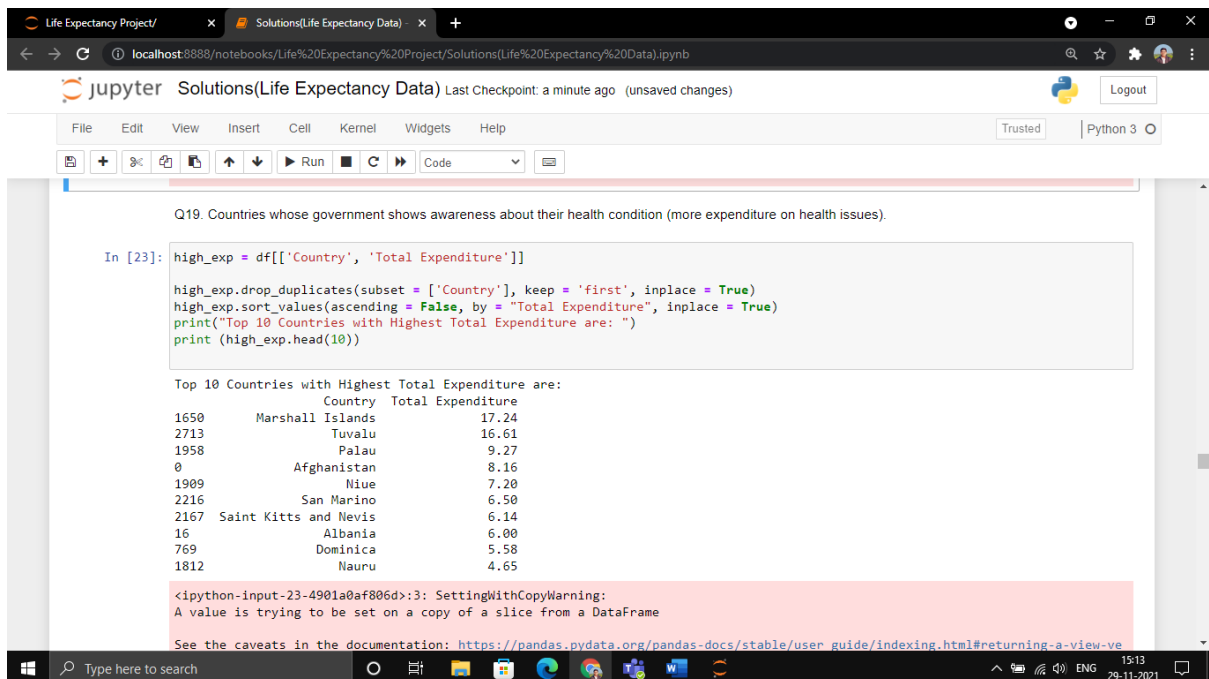
most_alcohol.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
most_alcohol.sort_values(ascending = False, by = "Alcohol Intake(L)", inplace = True)
print("Top 10 Countries with Most Alcohol Consumption are: ")
print(most_alcohol.head(10))
```

The output shows the top 10 countries with the highest alcohol consumption:

	Country	Alcohol Intake(L)
2762	United Kingdom of Great Britain and Northern I...	10.66
1845	New Zealand	8.70
2167	Saint Kitts and Nevis	8.54
1910	Norway	5.97
16	Albania	4.60
2313	Singapore	1.79
0	Afghanistan	0.01
624	Cook Islands	0.01
769	Dominica	0.01
1650	Marshall Islands	0.01

A warning message is displayed at the bottom: '<ipython-input-22-4ffa40f71049>:4: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame'.

Que 19. Countries whose government shows awareness about their health condition (more expenditure on health issues).



A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook title is 'Solutions(Life Expectancy Data)'. The code cell contains the following Python code:

```
In [23]: high_exp = df[['Country', 'Total Expenditure']]

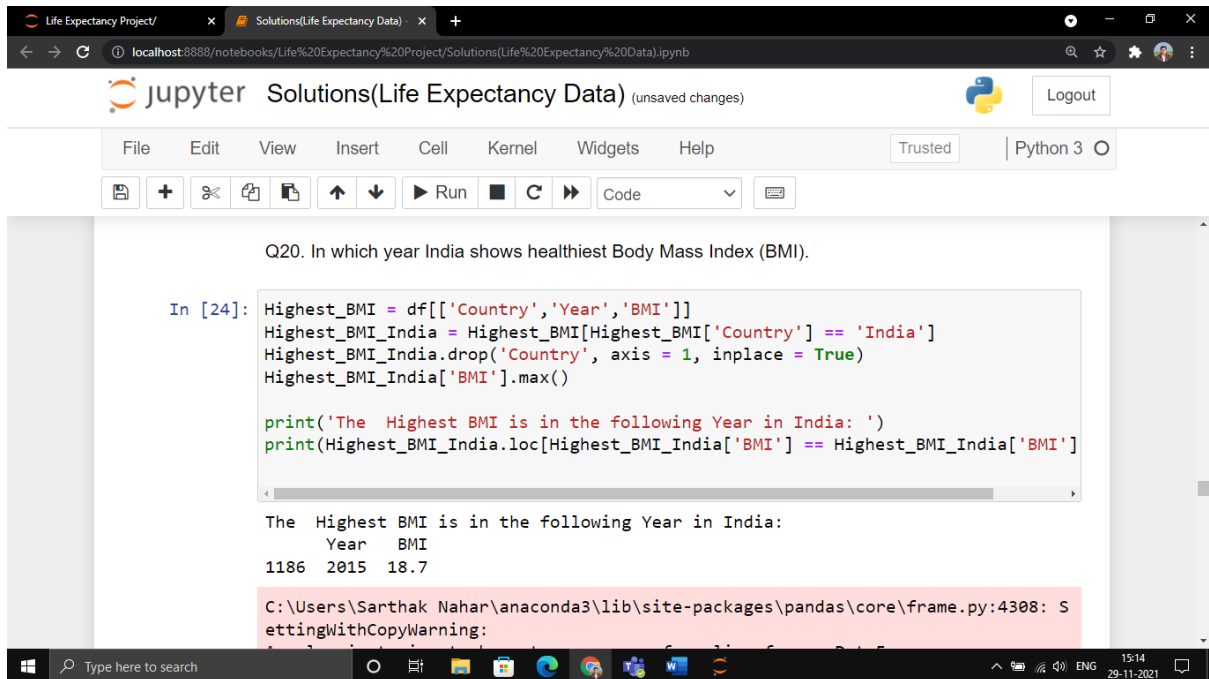
high_exp.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
high_exp.sort_values(ascending = False, by = "Total Expenditure", inplace = True)
print("Top 10 Countries with Highest Total Expenditure are: ")
print(high_exp.head(10))
```

The output shows the top 10 countries with the highest total health expenditure:

	Country	Total Expenditure
1650	Marshall Islands	17.24
2713	Tuvalu	16.61
1958	Palau	9.27
0	Afghanistan	8.16
1909	Niue	7.20
2216	San Marino	6.50
2167	Saint Kitts and Nevis	6.14
16	Albania	6.00
769	Dominica	5.58
1812	Nauru	4.65

A warning message is displayed at the bottom: '<ipython-input-23-4901a0af806d>:3: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame'. Below the warning, a link is provided: 'See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-vs-returning-a-copy'.

Que 20. In which year India shows healthiest Body Mass Index (BMI).

A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook title is 'Solutions(Life Expectancy Data) (unsaved changes)'. The code cell contains the following Python code:

```
In [24]: Highest_BMI = df[['Country', 'Year', 'BMI']]
Highest_BMI_India = Highest_BMI[Highest_BMI['Country'] == 'India']
Highest_BMI_India.drop('Country', axis = 1, inplace = True)
Highest_BMI_India['BMI'].max()

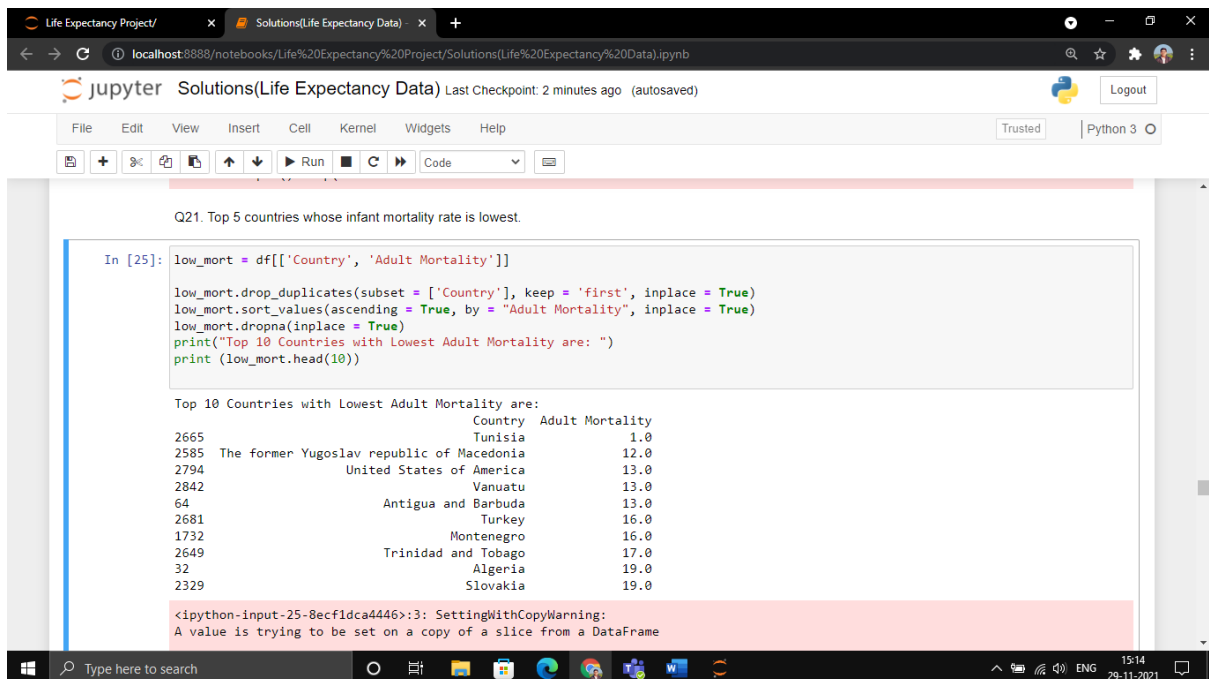
print('The Highest BMI is in the following Year in India: ')
print(Highest_BMI_India.loc[Highest_BMI_India['BMI'] == Highest_BMI_India['BMI']])
```

The output shows:

```
The Highest BMI is in the following Year in India:
Year BMI
1186 2015 18.7
```

A pink error message is visible at the bottom: 'C:\Users\Sarthak Nahar\anaconda3\lib\site-packages\pandas\core\frame.py:4308: SettingWithCopyWarning:'. The Windows taskbar at the bottom shows the date as 29-11-2021.

Que 21. Top 5 countries whose infant mortality rate is lowest.

A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook title is 'Solutions(Life Expectancy Data) Last Checkpoint: 2 minutes ago (autosaved)'. The code cell contains the following Python code:

```
In [25]: low_mort = df[['Country', 'Adult Mortality']]

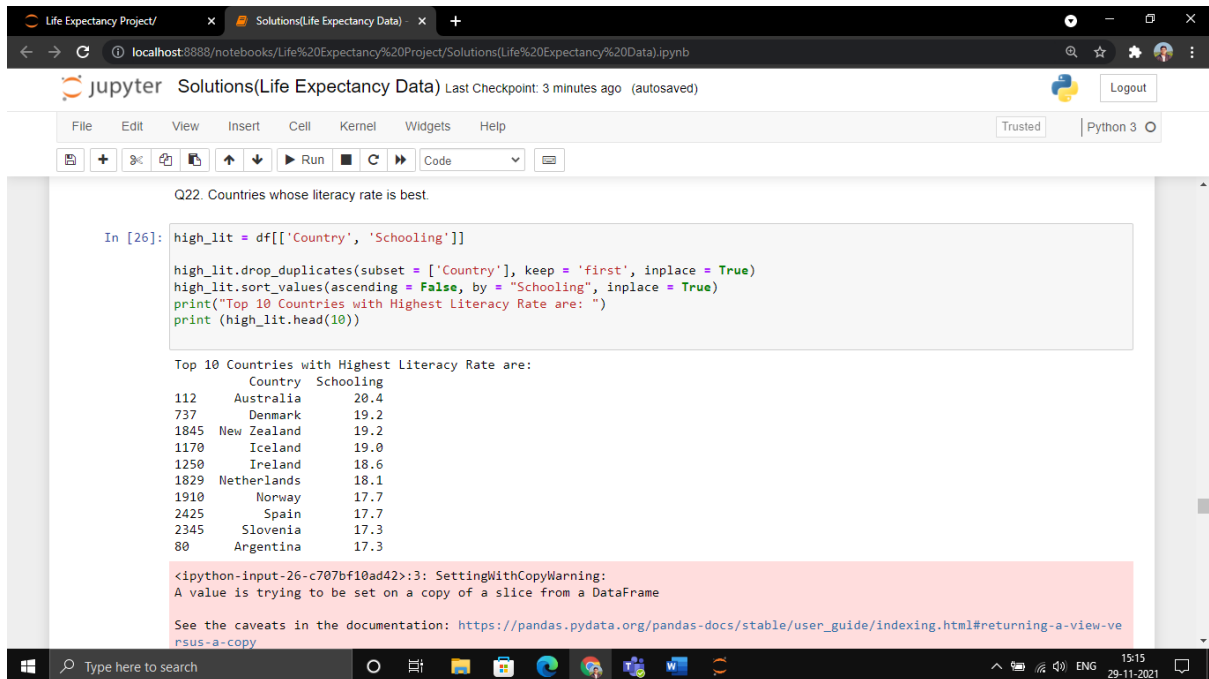
low_mort.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
low_mort.sort_values(ascending = True, by = "Adult Mortality", inplace = True)
low_mort.dropna(inplace = True)
print("Top 10 Countries with Lowest Adult Mortality are: ")
print(low_mort.head(10))
```

The output shows a table of the top 10 countries with the lowest adult mortality rates:

```
Top 10 Countries with Lowest Adult Mortality are:
Country Adult Mortality
2665 Tunisia 1.0
2585 The former Yugoslav republic of Macedonia 12.0
2794 United States of America 13.0
2842 Vanuatu 13.0
64 Antigua and Barbuda 13.0
2681 Turkey 16.0
1732 Montenegro 16.0
2649 Trinidad and Tobago 17.0
32 Algeria 19.0
2329 Slovakia 19.0
```

A pink error message is visible at the bottom: '<ipython-input-25-0ecf1dca4446>:3: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame'. The Windows taskbar at the bottom shows the date as 29-11-2021.

Que 22. Countries whose literacy rate is best.



A screenshot of a Jupyter Notebook interface. The browser tab is 'Life Expectancy Project/' and the notebook is 'Solutions(Life Expectancy Data)'. The URL is 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook has a 'Trusted' badge and 'Python 3' selected. The code cell contains the following Python code:

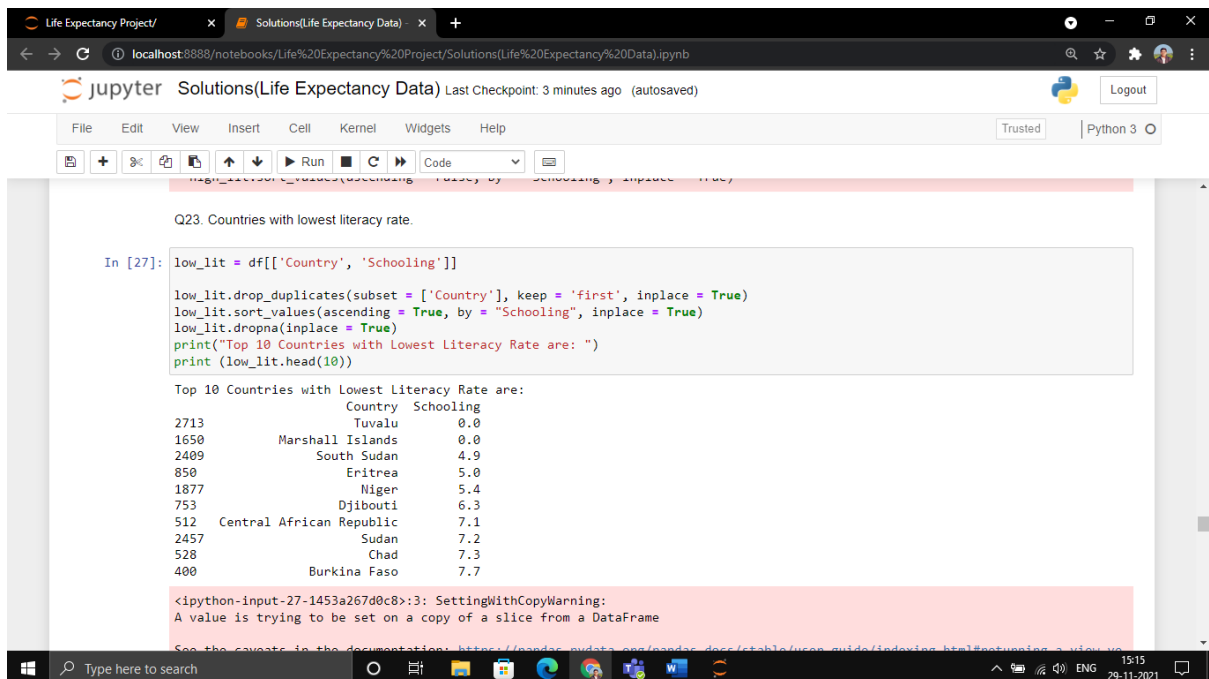
```
In [26]: high_lit = df[['Country', 'Schooling']]
high_lit.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
high_lit.sort_values(ascending = False, by = "Schooling", inplace = True)
print("Top 10 Countries with Highest Literacy Rate are: ")
print (high_lit.head(10))
```

The output shows the top 10 countries with the highest literacy rate:

	Country	Schooling
112	Australia	20.4
737	Denmark	19.2
1845	New Zealand	19.2
1170	Iceland	19.0
1250	Ireland	18.6
1829	Netherlands	18.1
1910	Norway	17.7
2425	Spain	17.7
2345	Slovenia	17.3
80	Argentina	17.3

A warning message is displayed at the bottom: '<ipython-input-26-c707bf10ad42>:3: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame'. A link to the pandas documentation is provided: 'See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy'.

Que 23. Countries with lowest literacy rate.



A screenshot of a Jupyter Notebook interface, similar to the one above. The code cell contains the following Python code:

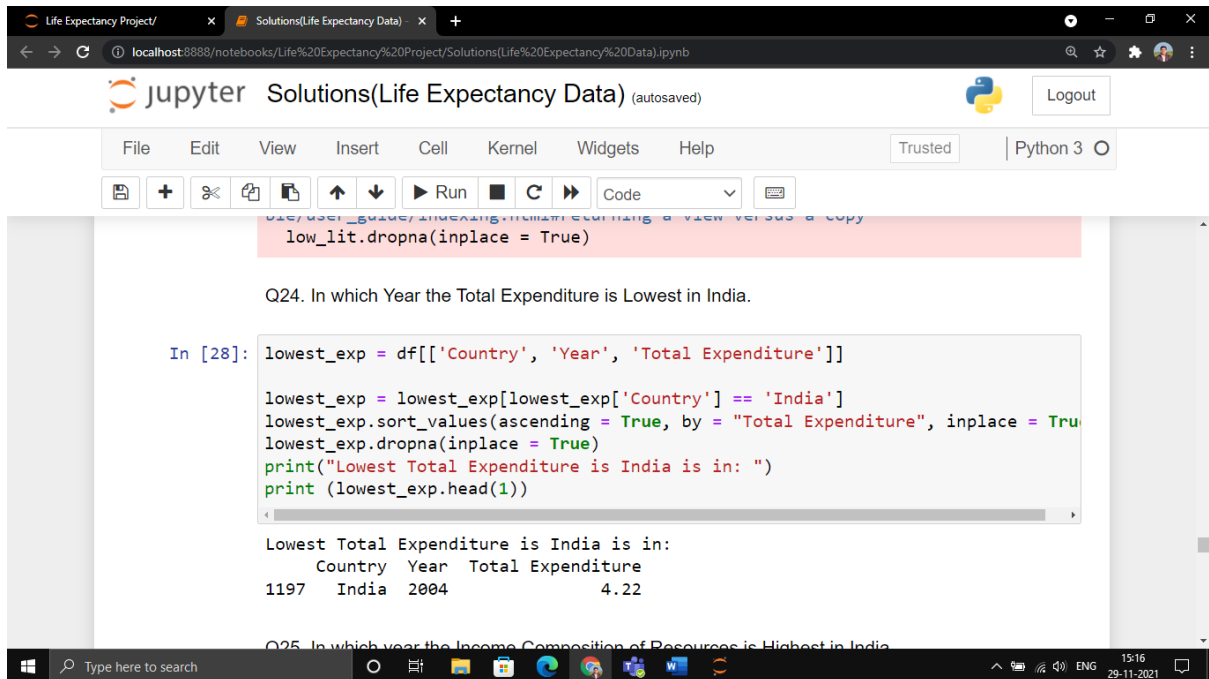
```
In [27]: low_lit = df[['Country', 'Schooling']]
low_lit.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
low_lit.sort_values(ascending = True, by = "Schooling", inplace = True)
low_lit.dropna(inplace = True)
print("Top 10 Countries with Lowest Literacy Rate are: ")
print (low_lit.head(10))
```

The output shows the top 10 countries with the lowest literacy rate:

	Country	Schooling
2713	Tuvalu	0.0
1650	Marshall Islands	0.0
2409	South Sudan	4.9
850	Eritrea	5.0
1877	Niger	5.4
753	Djibouti	6.3
512	Central African Republic	7.1
2457	Sudan	7.2
528	Chad	7.3
400	Burkina Faso	7.7

A warning message is displayed at the bottom: '<ipython-input-27-1453a267d0c8>:3: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame'. A link to the pandas documentation is provided: 'See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy'.

Que 24. In which Year the Total Expenditure is Lowest in India.



A screenshot of a Jupyter Notebook interface. The browser address bar shows 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook title is 'Solutions(Life Expectancy Data) (autosaved)'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar shows icons for saving, adding, deleting, and running cells. The code cell contains the following Python code:

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

Q24. In which Year the Total Expenditure is Lowest in India.

```
In [28]: lowest_exp = df[['Country', 'Year', 'Total Expenditure']]

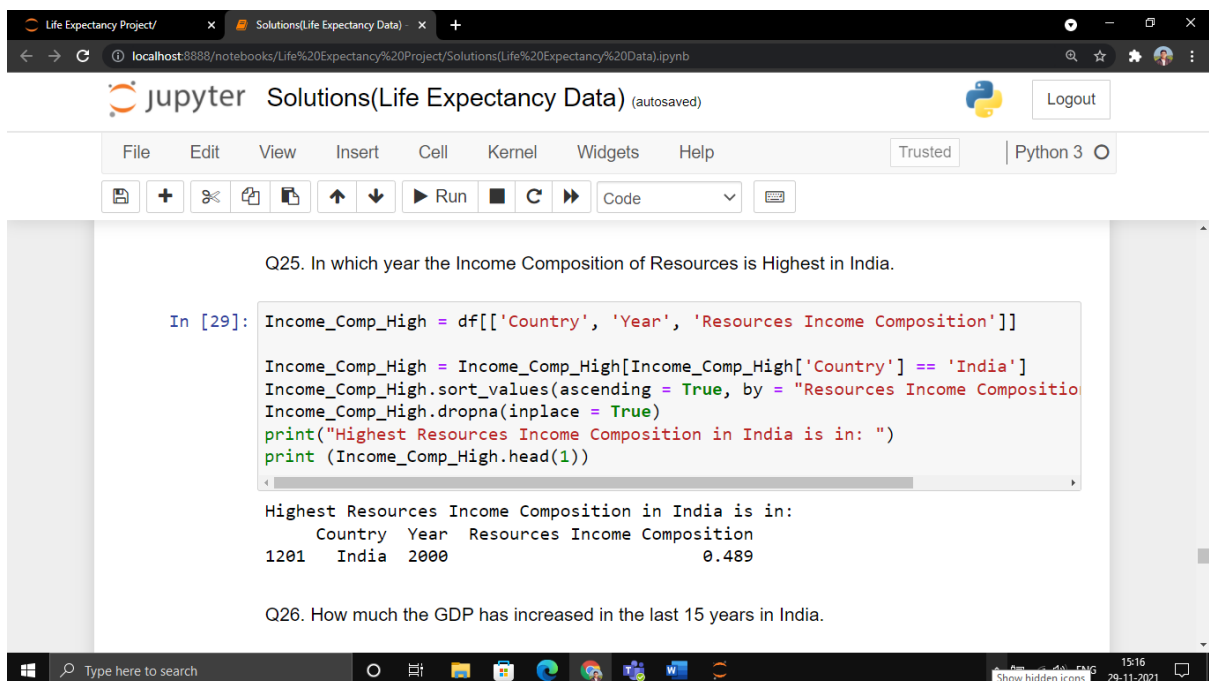
lowest_exp = lowest_exp[lowest_exp['Country'] == 'India']
lowest_exp.sort_values(ascending = True, by = "Total Expenditure", inplace = True)
lowest_exp.dropna(inplace = True)
print("Lowest Total Expenditure is India is in: ")
print (lowest_exp.head(1))
```

Lowest Total Expenditure is India is in:

Country	Year	Total Expenditure	
1197	India	2004	4.22

Q25. In which year the Income Composition of Resources is Highest in India.

Que 25. In which year the Income Composition of Resources is Highest in India.



A screenshot of a Jupyter Notebook interface, similar to the one above. The browser address bar shows 'localhost:8888/notebooks/Life%20Expectancy%20Project/Solutions(Life%20Expectancy%20Data).ipynb'. The notebook title is 'Solutions(Life Expectancy Data) (autosaved)'. The menu bar and toolbar are the same. The code cell contains the following Python code:

```
Income_Comp_High = df[['Country', 'Year', 'Resources Income Composition']]

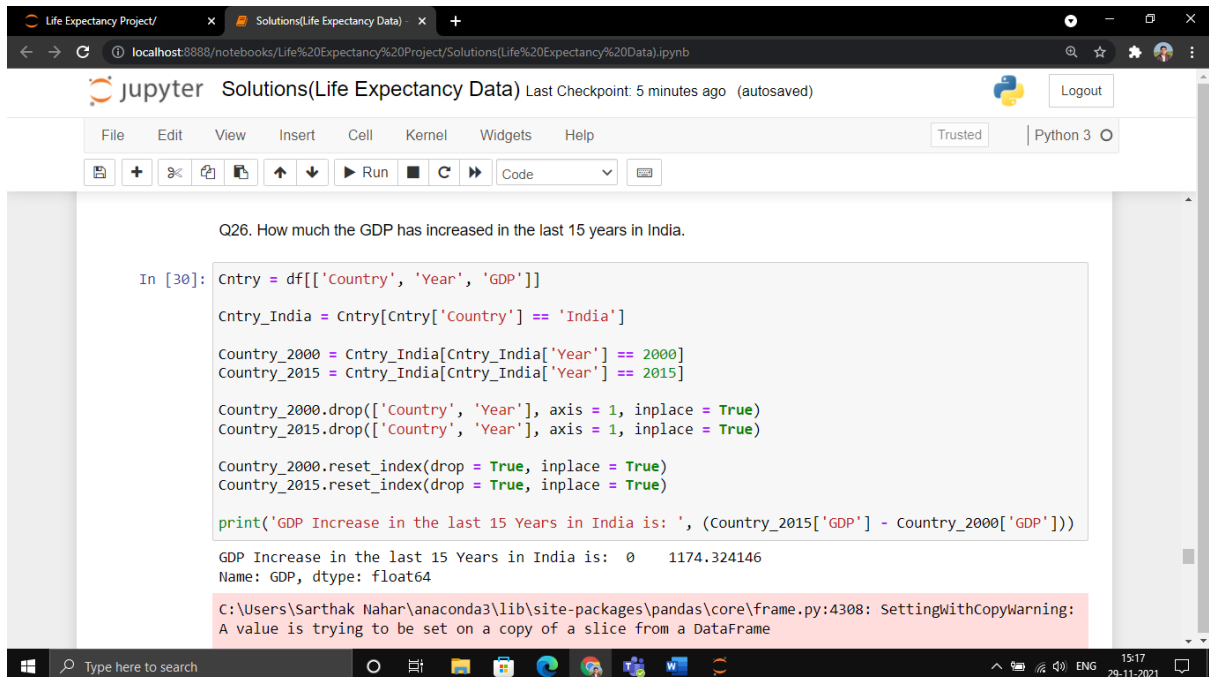
Income_Comp_High = Income_Comp_High[Income_Comp_High['Country'] == 'India']
Income_Comp_High.sort_values(ascending = True, by = "Resources Income Composition", inplace = True)
Income_Comp_High.dropna(inplace = True)
print("Highest Resources Income Composition in India is in: ")
print (Income_Comp_High.head(1))
```

Highest Resources Income Composition in India is in:

Country	Year	Resources Income Composition	
1201	India	2000	0.489

Q26. How much the GDP has increased in the last 15 years in India.

Que 26. How much the GDP has increased in the last 15 years in India.



```
Q26. How much the GDP has increased in the last 15 years in India.

In [30]: Cntry = df[['Country', 'Year', 'GDP']]
Cntry_India = Cntry[Cntry['Country'] == 'India']
Country_2000 = Cntry_India[Cntry_India['Year'] == 2000]
Country_2015 = Cntry_India[Cntry_India['Year'] == 2015]

Country_2000.drop(['Country', 'Year'], axis = 1, inplace = True)
Country_2015.drop(['Country', 'Year'], axis = 1, inplace = True)

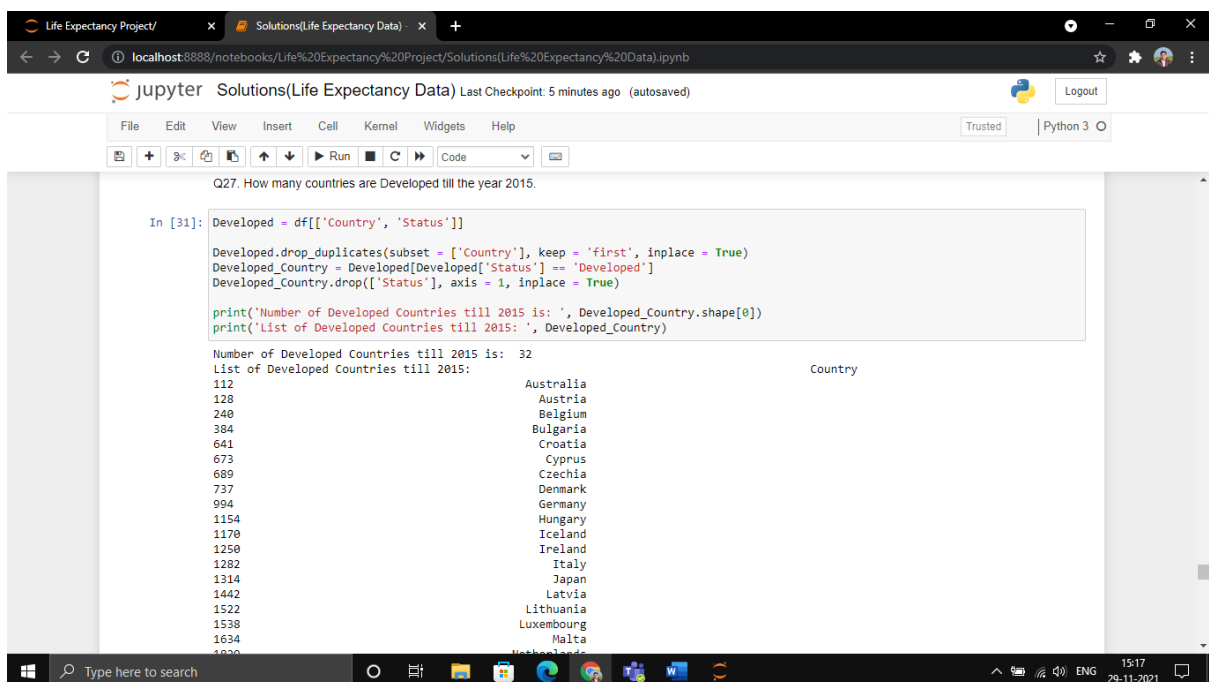
Country_2000.reset_index(drop = True, inplace = True)
Country_2015.reset_index(drop = True, inplace = True)

print('GDP Increase in the last 15 Years in India is: ', (Country_2015['GDP'] - Country_2000['GDP']))

GDP Increase in the last 15 Years in India is: 0    1174.324146
Name: GDP, dtype: float64

C:\Users\Sarthak Nahar\anaconda3\lib\site-packages\pandas\core\frame.py:4308: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

Que 27. How many countries are Developed till the year 2015.

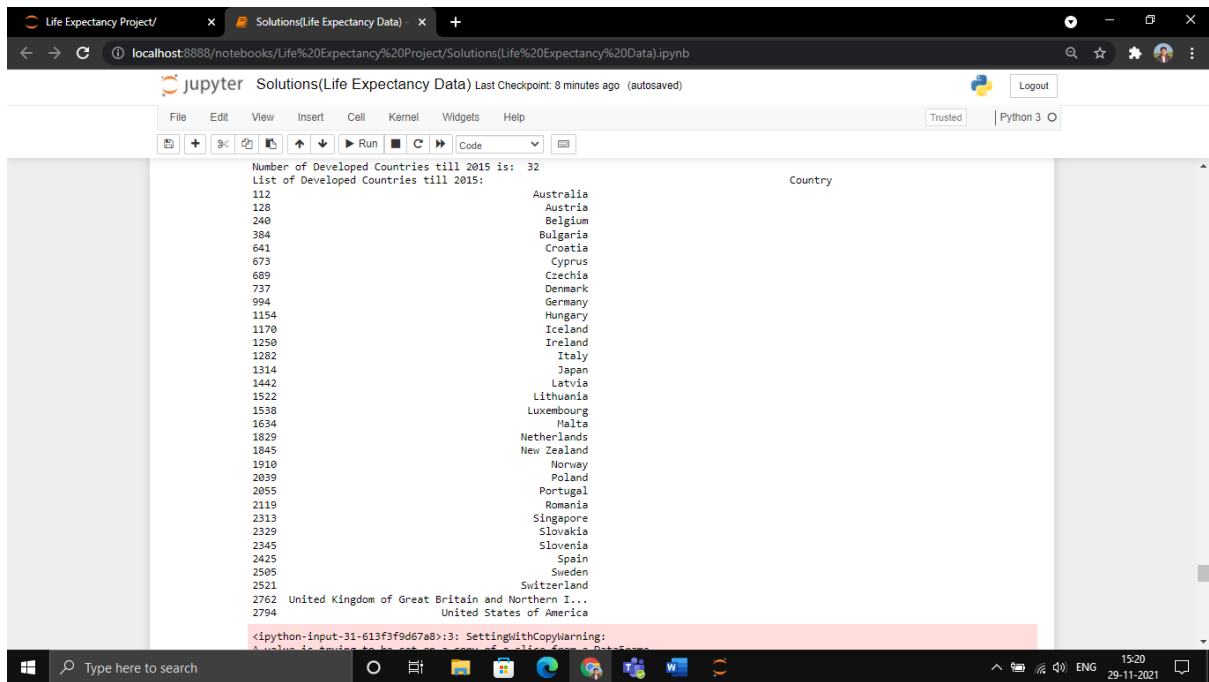


```
Q27. How many countries are Developed till the year 2015.

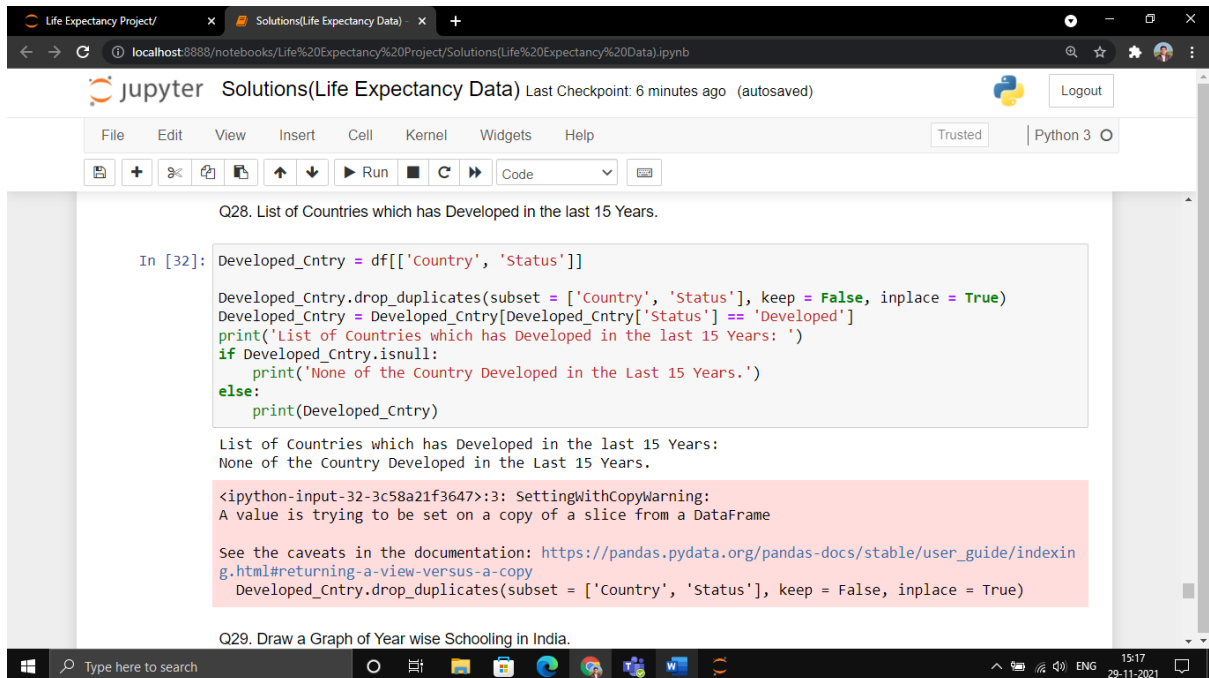
In [31]: Developed = df[['Country', 'Status']]
Developed.drop_duplicates(subset = ['Country'], keep = 'first', inplace = True)
Developed_Country = Developed[Developed['Status'] == 'Developed']
Developed_Country.drop(['Status'], axis = 1, inplace = True)

print('Number of Developed Countries till 2015 is: ', Developed_Country.shape[0])
print('List of Developed Countries till 2015: ', Developed_Country)

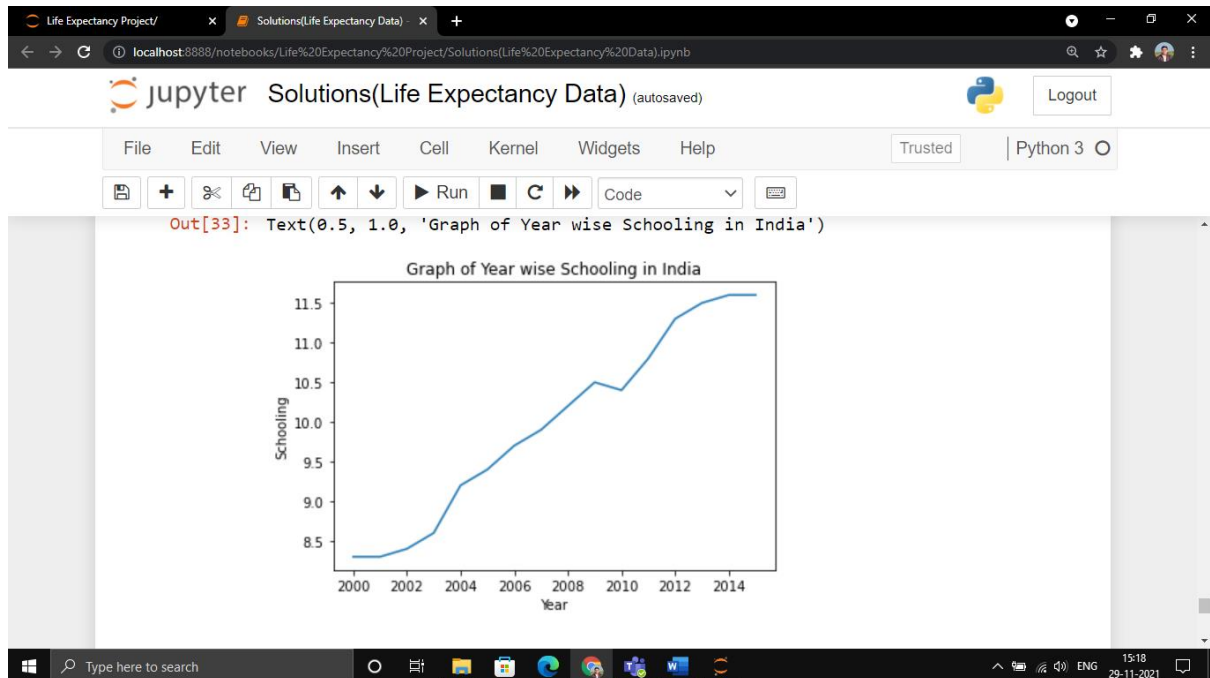
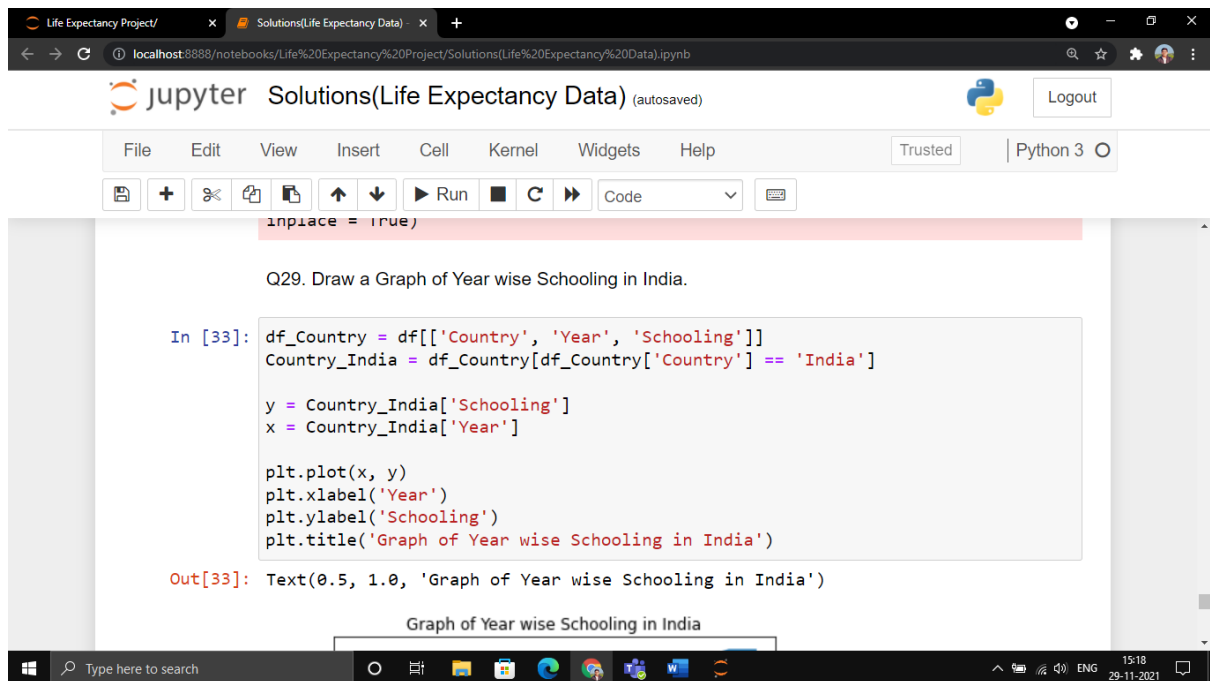
Number of Developed Countries till 2015 is: 32
List of Developed Countries till 2015:
Country
112      Australia
128      Austria
240      Belgium
384      Bulgaria
641      Croatia
673      Cyprus
689      Czechia
737      Denmark
994      Germany
1154     Hungary
1170     Iceland
1250     Ireland
1282      Italy
1314      Japan
1442      Latvia
1522     Lithuania
1538     Luxembourg
1634      Malta
1692     Netherlands
```



Que 28. List of Countries which has Developed in the last 15 Years.



Que 29. Draw a Graph of Year wise Schooling in India.



Que 30. Draw a Graph of Life Expectancy of different Countries.

