Sankeerth Sai Shabad

Introduction to Data Science Project-3

03/01/2022

PURPOSE OF THE PROBLEM:

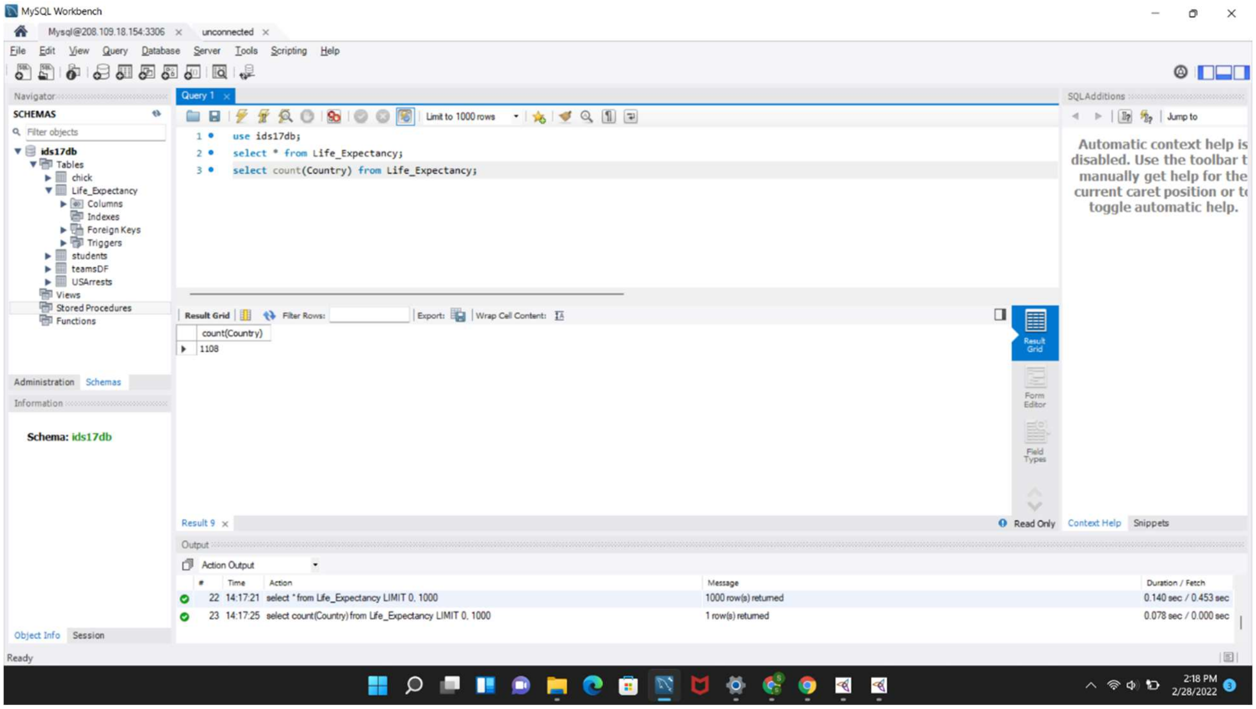
- To perform data cleaning and find and insert data by using data cleaning techniques and necessary steps to ensure the dataset is cleansed and performing SQL queries using MySQL Workbench.
- Importing cleansed MySQL files into data frames in Jupyter notebook in Google colab. And to perform and plot the linear regression model and independent models that have a high correlation with correlation analysis
- To provide the equation of each model and compute their R2 and MSE.

Methodology:

- COLLECTION OF DATA: All the data is collected from the dataset with their values from https://github.com/SankeerthShabad/IDS/blob/main/HW3/Life_Expectancy.csv
- OPERATIONS: Addressing missing values by using mean and performing required queries by using SQL in MySQL Workbench. And Importing cleansed data into Jupyter notebook in Google colab. And to perform and plot linear regression, correlation analysis by using Python.
- OBSERVATIONS: To find the values by comparing and effects caused by life expectancy and mortality rates. Negative correlation for life expectancy by social and economic factors. And the impact of schooling on the lifespan of humans and to observe the performance of the models.

Results:

- SQL: After performing SQL queries for data cleaning and operations required, using MySQL Workbench the following outputs are observed.

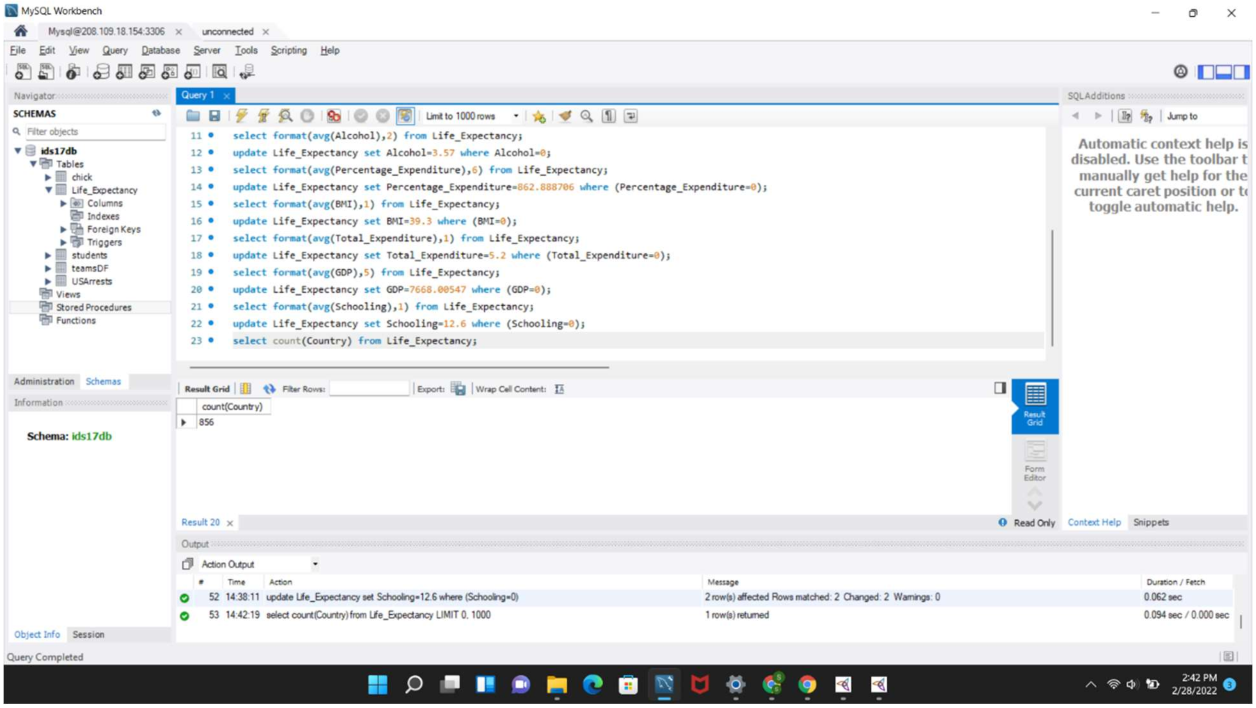


```
use ids17db;
select * from Life_Expectancy;
select count(Country) from Life_Expectancy;
```

```
count(Country)
1108
```

```
#   Time      Action                                                  Message                 Duration / Fetch
22  14:17:21  select * from Life_Expectancy LIMIT 0, 1000             1000 row(s) returned    0.140 sec / 0.453 sec
23  14:17:25  select count(Country) from Life_Expectancy LIMIT 0, 1000  1 row(s) returned     0.078 sec / 0.000 sec
```

Fig: Total number of country counts

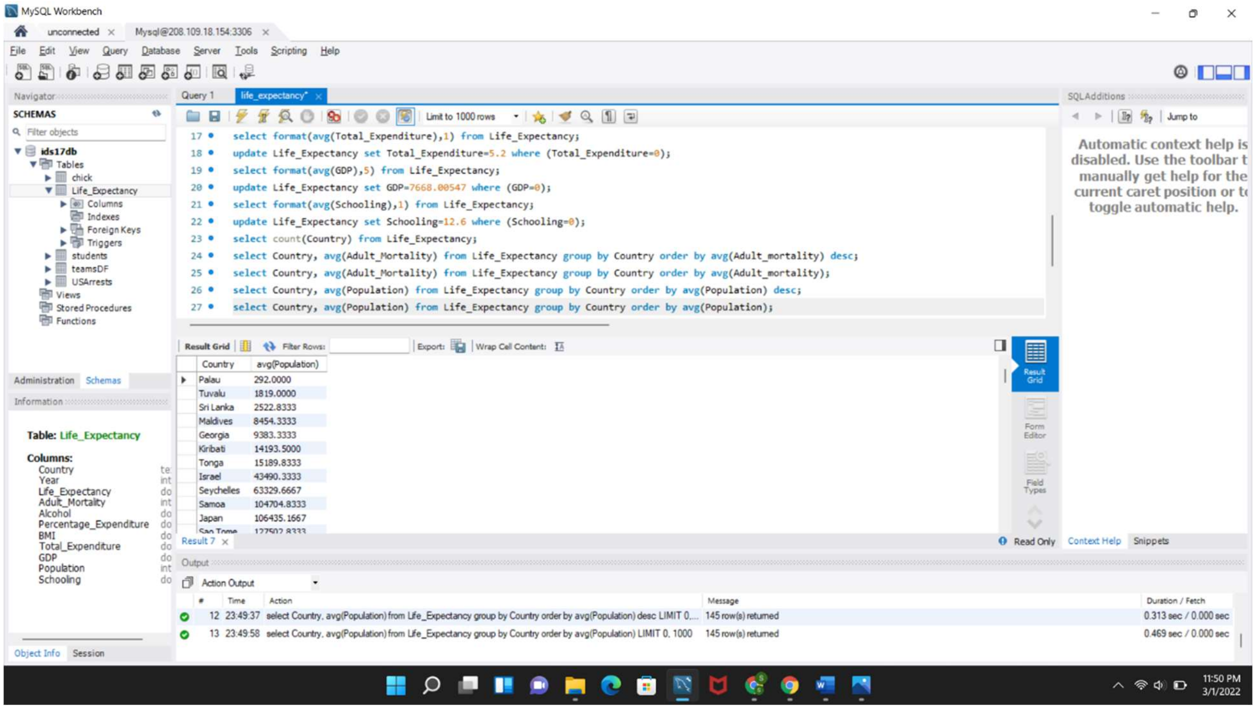


```
select format(avg(Alcohol),2) from Life_Expectancy;
update Life_Expectancy set Alcohol=3.57 where Alcohol=0;
select format(avg(Percentage_Expenditure),6) from Life_Expectancy;
update Life_Expectancy set Percentage_Expenditure=862.888706 where (Percentage_Expenditure=0);
select format(avg(BMI),1) from Life_Expectancy;
update Life_Expectancy set BMI=39.3 where (BMI=0);
select format(avg(Total_Expenditure),1) from Life_Expectancy;
update Life_Expectancy set Total_Expenditure=5.2 where (Total_Expenditure=0);
select format(avg(GDP),5) from Life_Expectancy;
update Life_Expectancy set GDP=7668.00547 where (GDP=0);
select format(avg(Schooling),1) from Life_Expectancy;
update Life_Expectancy set Schooling=12.6 where (Schooling=0);
select count(Country) from Life_Expectancy;
```

```
count(Country)
856
```

```
#   Time      Action                                                      Message                                                 Duration / Fetch
52  14:38:11  update Life_Expectancy set Schooling=12.6 where (Schooling=0)  2 row(s) affected Rows matched: 2  Changed: 2  Warnings: 0   0.062 sec
53  14:42:19  select count(Country) from Life_Expectancy LIMIT 0, 1000    1 row(s) returned                                       0.094 sec / 0.000 sec
```

Fig: Total count of countries after performing data cleaning techniques



Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

Table: Life_Expectancy

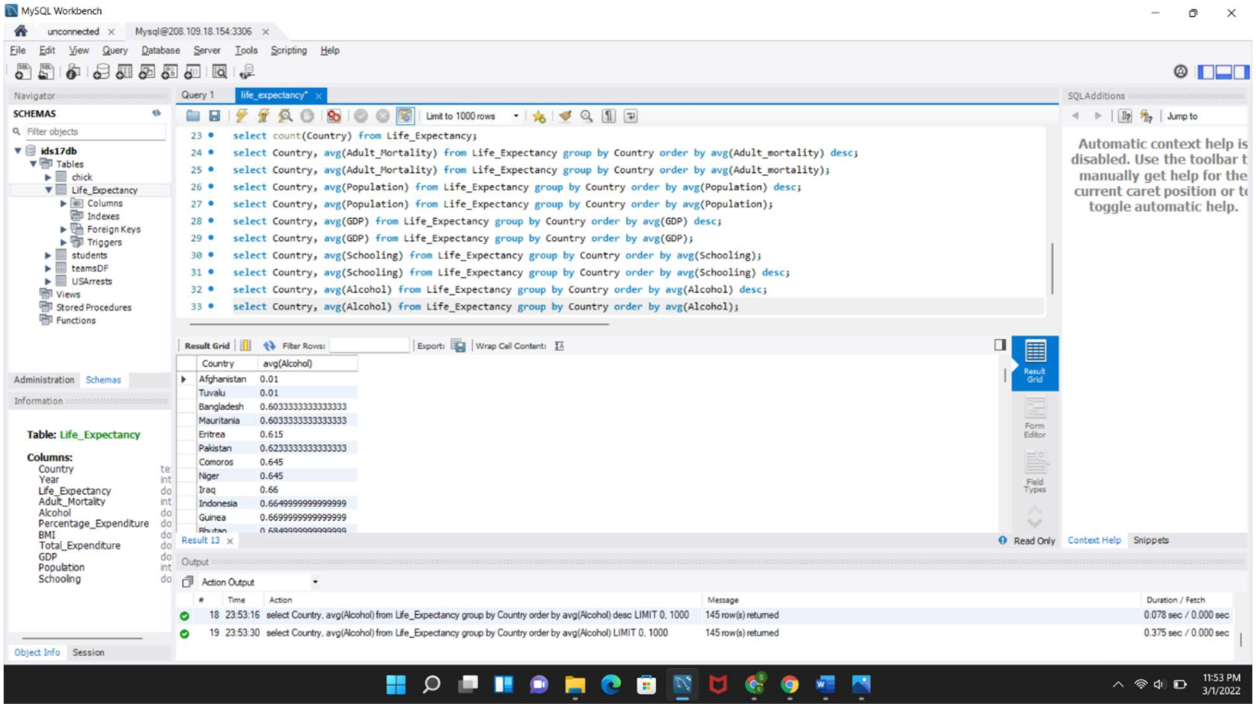
Columns:
Country te
Year int
Life_Expectancy do
Adult_Mortality int
Alcohol do
Percentage_Expenditure do
BMI do
Total_Expenditure do
GDP do
Population int
Schooling do

```
select format(avg(Total_Expenditure),1) from Life_Expectancy;
update Life_Expectancy set Total_Expenditure=5.2 where (Total_Expenditure=0);
select format(avg(GDP),5) from Life_Expectancy;
update Life_Expectancy set GDP=7668.00547 where (GDP=0);
select format(avg(Schooling),1) from Life_Expectancy;
update Life_Expectancy set Schooling=12.6 where (Schooling=0);
select count(Country) from Life_Expectancy;
select Country, avg(Adult_Mortality) from Life_Expectancy group by Country order by avg(Adult_mortality) desc;
select Country, avg(Adult_Mortality) from Life_Expectancy group by Country order by avg(Adult_mortality);
select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population) desc;
select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population);
```

Country	avg(Population)
Palau	292.0000
Tuvalu	1819.0000
Sri Lanka	2522.8333
Maldives	8454.3333
Georgia	9383.3333
Kiribati	14193.5000
Tonga	15189.8333
Israel	43490.3333
Seychelles	63329.6667
Samoa	104704.8333
Japan	106435.1667
Sao Tome	127502.8333

```
#   Time      Action                                                                                       Message              Duration / Fetch
12  23:49:37  select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population) desc LIMIT 0,...  145 row(s) returned  0.313 sec / 0.000 sec
13  23:49:58  select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population) LIMIT 0, 1000   145 row(s) returned  0.469 sec / 0.000 sec
```

Fig: List of countries with highest average mortality rates



Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

Table: Life_Expectancy

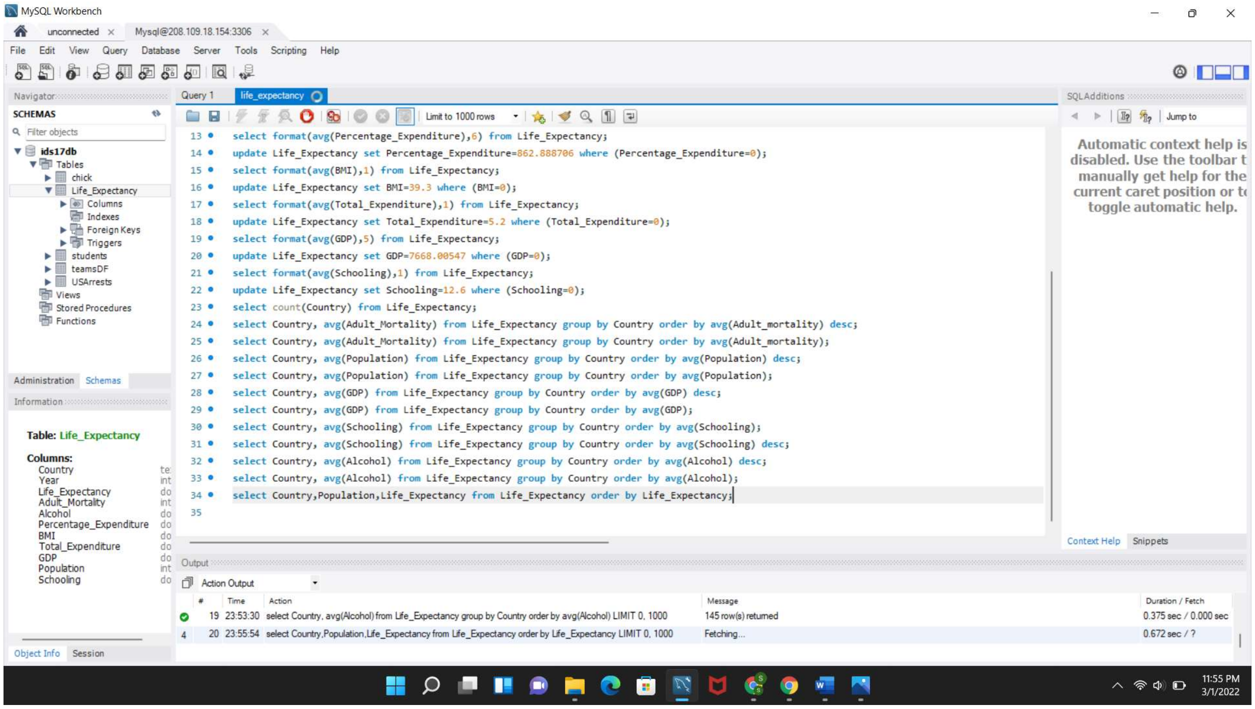
Columns:
Country te
Year int
Life_Expectancy do
Adult_Mortality int
Alcohol do
Percentage_Expenditure do
BMI do
Total_Expenditure do
GDP do
Population int
Schooling do

```
select count(Country) from Life_Expectancy;
select Country, avg(Adult_Mortality) from Life_Expectancy group by Country order by avg(Adult_mortality) desc;
select Country, avg(Adult_Mortality) from Life_Expectancy group by Country order by avg(Adult_mortality);
select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population) desc;
select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population);
select Country, avg(GDP) from Life_Expectancy group by Country order by avg(GDP) desc;
select Country, avg(GDP) from Life_Expectancy group by Country order by avg(GDP);
select Country, avg(Schooling) from Life_Expectancy group by Country order by avg(Schooling);
select Country, avg(Schooling) from Life_Expectancy group by Country order by avg(Schooling) desc;
select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol) desc;
select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol);
```

Country	avg(Alcohol)
Afghanistan	0.01
Tuvalu	0.01
Bangladesh	0.6033333333333333
Mauritania	0.6033333333333333
Eritrea	0.615
Pakistan	0.6233333333333333
Comoros	0.645
Niger	0.645
Iraq	0.66
Indonesia	0.6649999999999999
Guinea	0.6699999999999999
Bhutan	0.6849999999999999

```
#   Time      Action                                                                                      Message              Duration / Fetch
18  23:53:16  select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol) desc LIMIT 0, 1000  145 row(s) returned  0.078 sec / 0.000 sec
19  23:53:30  select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol) LIMIT 0, 1000       145 row(s) returned  0.375 sec / 0.000 sec
```

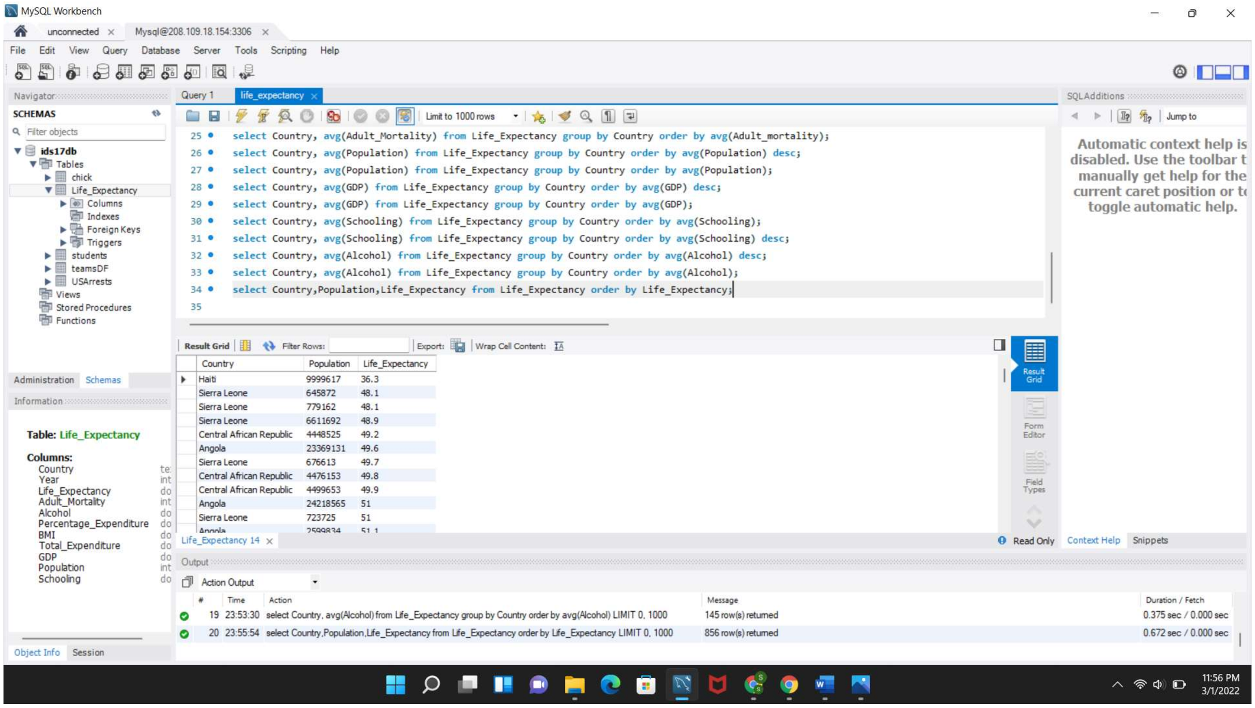
Fig: List of countries with Lowest average mortality rates



```
select format(avg(Percentage_Expenditure),6) from Life_Expectancy;
update Life_Expectancy set Percentage_Expenditure=862.888706 where (Percentage_Expenditure=0);
select format(avg(BMI),1) from Life_Expectancy;
update Life_Expectancy set BMI=39.3 where (BMI=0);
select format(avg(Total_Expenditure),1) from Life_Expectancy;
update Life_Expectancy set Total_Expenditure=5.2 where (Total_Expenditure=0);
select format(avg(GDP),5) from Life_Expectancy;
update Life_Expectancy set GDP=7668.00547 where (GDP=0);
select format(avg(Schooling),1) from Life_Expectancy;
update Life_Expectancy set Schooling=12.6 where (Schooling=0);
select count(Country) from Life_Expectancy;
select Country, avg(Adult_Mortality) from Life_Expectancy group by Country order by avg(Adult_mortality) desc;
select Country, avg(Adult_Mortality) from Life_Expectancy group by Country order by avg(Adult_mortality);
select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population) desc;
select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population);
select Country, avg(GDP) from Life_Expectancy group by Country order by avg(GDP) desc;
select Country, avg(GDP) from Life_Expectancy group by Country order by avg(GDP);
select Country, avg(Schooling) from Life_Expectancy group by Country order by avg(Schooling);
select Country, avg(Schooling) from Life_Expectancy group by Country order by avg(Schooling) desc;
select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol) desc;
select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol);
select Country,Population,Life_Expectancy from Life_Expectancy order by Life_Expectancy;
```

```
   #   Time      Action                                                                                                   Message             Duration / Fetch
   19  23:53:30  select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol) LIMIT 0, 1000  145 row(s) returned  0.375 sec / 0.000 sec
4  20  23:55:54  select Country,Population,Life_Expectancy from Life_Expectancy order by Life_Expectancy LIMIT 0, 1000     Fetching...          0.672 sec / ?
```

Fig: Queries for the highest and lowest average for Countries and years



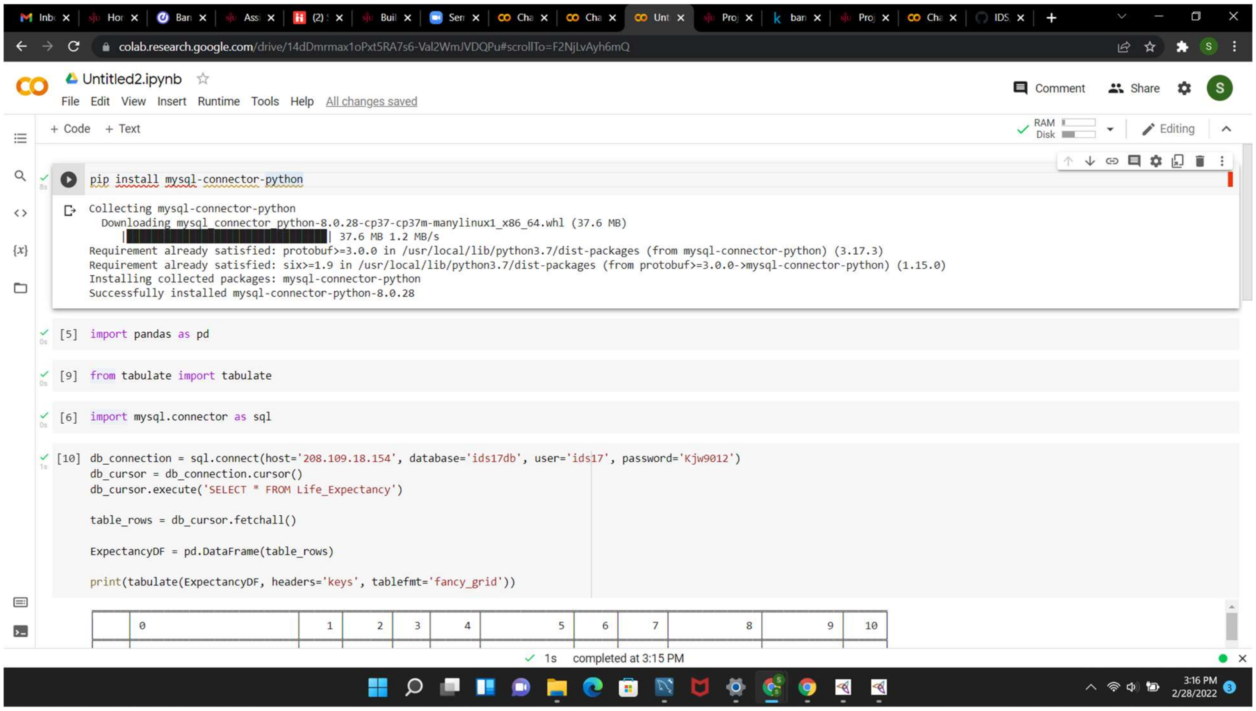
```
select Country, avg(Adult_Mortality) from Life_Expectancy group by Country order by avg(Adult_mortality);
select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population) desc;
select Country, avg(Population) from Life_Expectancy group by Country order by avg(Population);
select Country, avg(GDP) from Life_Expectancy group by Country order by avg(GDP) desc;
select Country, avg(GDP) from Life_Expectancy group by Country order by avg(GDP);
select Country, avg(Schooling) from Life_Expectancy group by Country order by avg(Schooling);
select Country, avg(Schooling) from Life_Expectancy group by Country order by avg(Schooling) desc;
select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol) desc;
select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol);
select Country,Population,Life_Expectancy from Life_Expectancy order by Life_Expectancy;
```

Country	Population	Life_Expectancy
Haiti	9999617	36.3
Sierra Leone	645872	48.1
Sierra Leone	779162	48.1
Sierra Leone	6611692	48.9
Central African Republic	4448525	49.2
Angola	23369131	49.6
Sierra Leone	676613	49.7
Central African Republic	4476153	49.8
Central African Republic	4499653	49.9
Angola	24218565	51
Sierra Leone	723725	51
Angola	2599834	51.1

```
#   Time      Action                                                                                                   Message             Duration / Fetch
19  23:53:30  select Country, avg(Alcohol) from Life_Expectancy group by Country order by avg(Alcohol) LIMIT 0, 1000  145 row(s) returned  0.375 sec / 0.000 sec
20  23:55:54  select Country,Population,Life_Expectancy from Life_Expectancy order by Life_Expectancy LIMIT 0, 1000     856 row(s) returned  0.672 sec / 0.000 sec
```

Fig: Comparing life expectancy for densely populated countries

PYTHON: After Importing cleansed MySQL files into data frames in Jupyter notebook in Google colab the plots, linear regression model, and independent models that have a high correlation with correlation analysis and computed R2 and MSE.



```
pip install mysql-connector-python
```

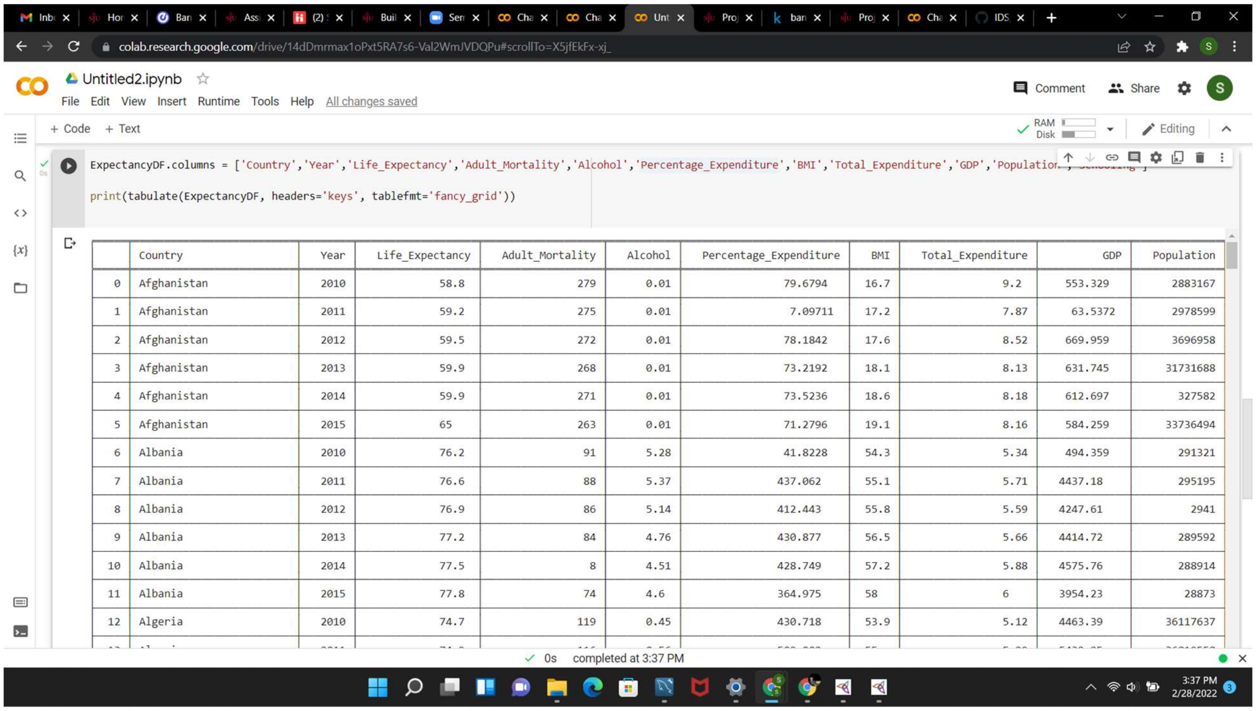
```
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.0.28-cp37-cp37m-manylinux1_x86_64.whl (37.6 MB)
     |████████████████████████████████| 37.6 MB 1.2 MB/s
Requirement already satisfied: protobuf>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from mysql-connector-python) (3.17.3)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.7/dist-packages (from protobuf>=3.0.0->mysql-connector-python) (1.15.0)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-8.0.28
```

```
[5]  import pandas as pd

[9]  from tabulate import tabulate

[6]  import mysql.connector as sql

[10] db_connection = sql.connect(host='208.109.18.154', database='ids17db', user='ids17', password='Kjw9012')
     db_cursor = db_connection.cursor()
     db_cursor.execute('SELECT * FROM Life_Expectancy')

     table_rows = db_cursor.fetchall()

     ExpectancyDF = pd.DataFrame(table_rows)

     print(tabulate(ExpectancyDF, headers='keys', tablefmt='fancy_grid'))
```

```
│    │ 0                    │   1 │   2 │  3 │   4 │        5 │   6 │    7 │         8 │        9 │   10 │
```

✓ 1s completed at 3:15 PM

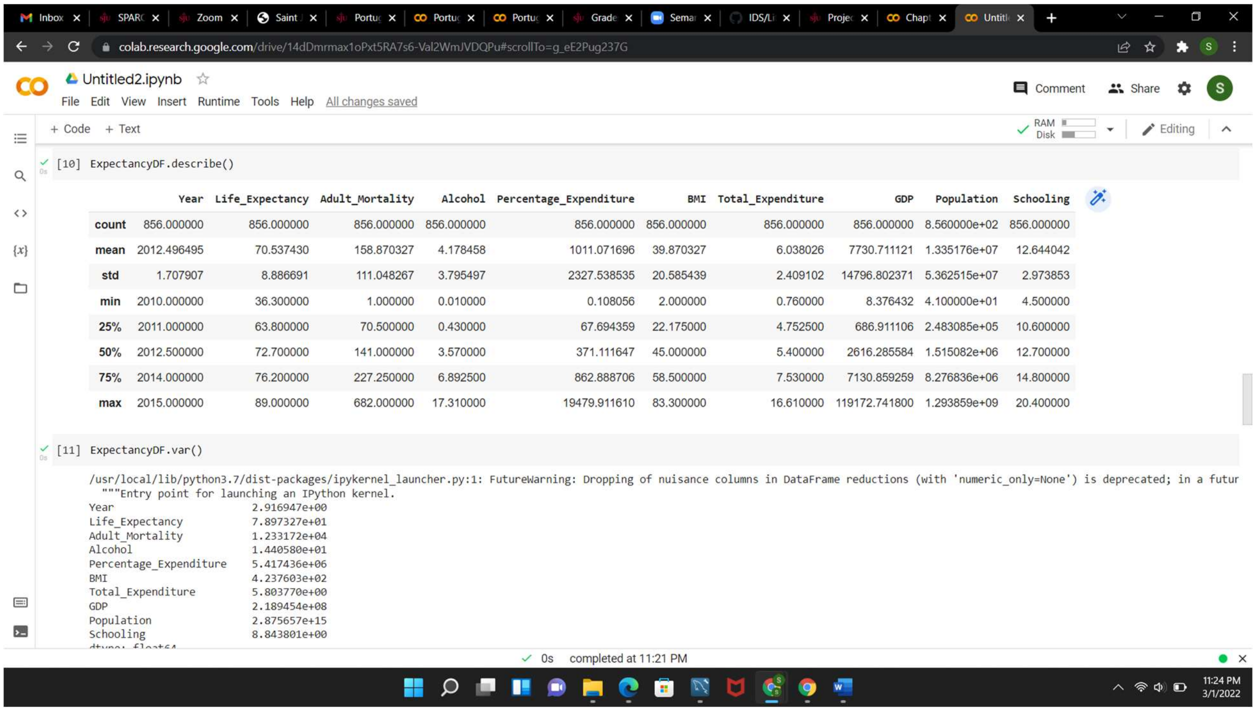
Fig: Importing cleansed data in Jupyter



```
ExpectancyDF.columns = ['Country','Year','Life_Expectancy','Adult_Mortality','Alcohol','Percentage_Expenditure','BMI','Total_Expenditure','GDP','Population

print(tabulate(ExpectancyDF, headers='keys', tablefmt='fancy_grid'))
```

```
│    │ Country      │   Year │   Life_Expectancy │   Adult_Mortality │   Alcohol │   Percentage_Expenditure │   BMI │   Total_Expenditure │      GDP │   Population │
│  0 │ Afghanistan  │   2010 │              58.8 │               279 │      0.01 │                  79.6794 │  16.7 │                9.2  │  553.329 │      2883167 │
│  1 │ Afghanistan  │   2011 │              59.2 │               275 │      0.01 │                  7.09711 │  17.2 │                7.87 │  63.5372 │      2978599 │
│  2 │ Afghanistan  │   2012 │              59.5 │               272 │      0.01 │                  78.1842 │  17.6 │                8.52 │  669.959 │      3696958 │
│  3 │ Afghanistan  │   2013 │              59.9 │               268 │      0.01 │                  73.2192 │  18.1 │                8.13 │  631.745 │     31731688 │
│  4 │ Afghanistan  │   2014 │              59.9 │               271 │      0.01 │                  73.5236 │  18.6 │                8.18 │  612.697 │       327582 │
│  5 │ Afghanistan  │   2015 │              65   │               263 │      0.01 │                  71.2796 │  19.1 │                8.16 │  584.259 │     33736494 │
│  6 │ Albania      │   2010 │              76.2 │                91 │      5.28 │                  41.8228 │  54.3 │                5.34 │  494.359 │       291321 │
│  7 │ Albania      │   2011 │              76.6 │                88 │      5.37 │                 437.062  │  55.1 │                5.71 │ 4437.18  │       295195 │
│  8 │ Albania      │   2012 │              76.9 │                86 │      5.14 │                 412.443  │  55.8 │                5.59 │ 4247.61  │         2941 │
│  9 │ Albania      │   2013 │              77.2 │                84 │      4.76 │                 430.877  │  56.5 │                5.66 │ 4414.72  │       289592 │
│ 10 │ Albania      │   2014 │              77.5 │                 8 │      4.51 │                 428.749  │  57.2 │                5.88 │ 4575.76  │       288914 │
│ 11 │ Albania      │   2015 │              77.8 │                74 │      4.6  │                 364.975  │  58   │                6    │ 3954.23  │        28873 │
│ 12 │ Algeria      │   2010 │              74.7 │               119 │      0.45 │                 430.718  │  53.9 │                5.12 │ 4463.39  │     36117637 │
```

✓ 0s completed at 3:37 PM



Inbox
SPARC
Zoom
Saint
Portu
Portu
Portu
Grade
Semar
IDS/L
Projec
Chapt
Untitle
colab.research.google.com/drive/14dDmrmax1oPxt5RA7s6-Val2WmJVDQPu#scrollTo=g_eE2Pug237G
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing

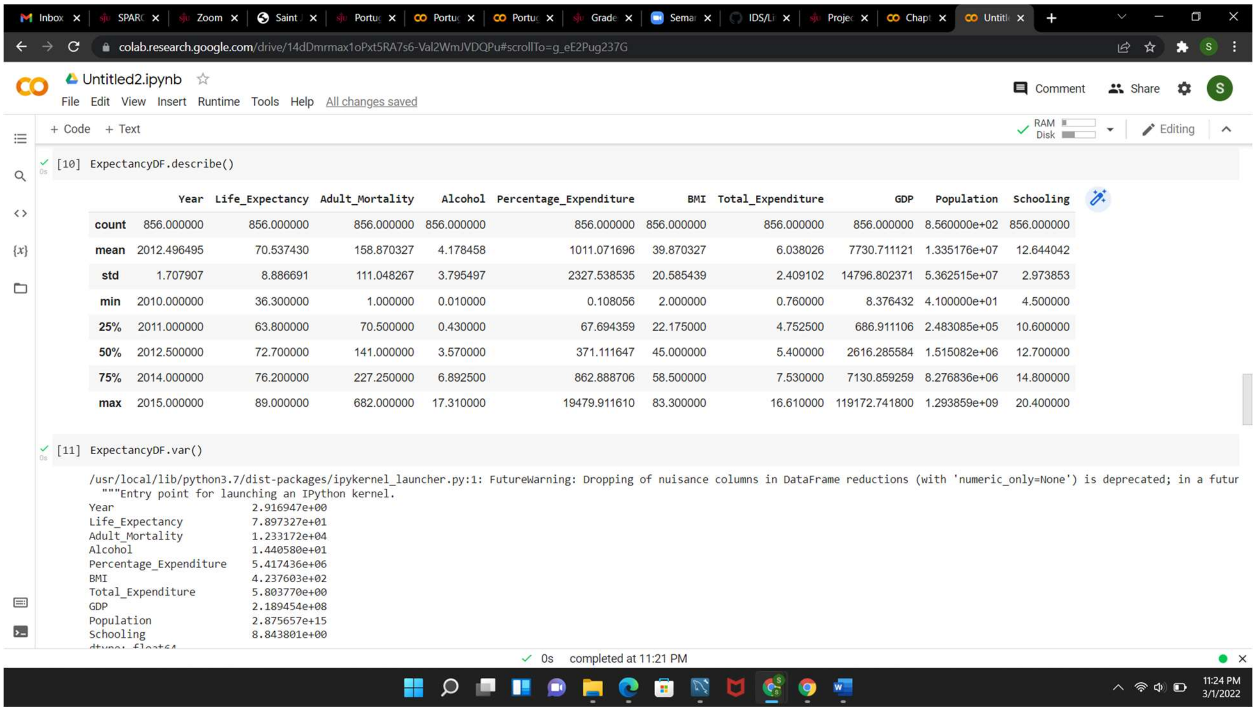
```
[10] ExpectancyDF.describe()
```

	Year	Life_Expectancy	Adult_Mortality	Alcohol	Percentage_Expenditure	BMI	Total_Expenditure	GDP	Population	Schooling
count	856.000000	856.000000	856.000000	856.000000	856.000000	856.000000	856.000000	856.000000	8.560000e+02	856.000000
mean	2012.496495	70.537430	158.870327	4.178458	1011.071696	39.870327	6.038026	7730.711121	1.335176e+07	12.644042
std	1.707907	8.886691	111.048267	3.795497	2327.538535	20.585439	2.409102	14796.802371	5.362515e+07	2.973853
min	2010.000000	36.300000	1.000000	0.010000	0.108056	2.000000	0.760000	8.376432	4.100000e+01	4.500000
25%	2011.000000	63.800000	70.500000	0.430000	67.694359	22.175000	4.752500	686.911106	2.483085e+05	10.600000
50%	2012.500000	72.700000	141.000000	3.570000	371.111647	45.000000	5.400000	2616.285584	1.515082e+06	12.700000
75%	2014.000000	76.200000	227.250000	6.892500	862.888706	58.500000	7.530000	7130.859259	8.276836e+06	14.800000
max	2015.000000	89.000000	682.000000	17.310000	19479.911610	83.300000	16.610000	119172.741800	1.293859e+09	20.400000

```
[11] ExpectancyDF.var()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a futur
  """Entry point for launching an IPython kernel.
Year                      2.916947e+00
Life_Expectancy           7.897327e+01
Adult_Mortality           1.233172e+04
Alcohol                   1.440580e+01
Percentage_Expenditure    5.417436e+06
BMI                       4.237603e+02
Total_Expenditure         5.803770e+00
GDP                       2.189454e+08
Population                2.875657e+15
Schooling                 8.843801e+00
dtype: float64
```

Fig: calculating count, mean, std, min by using describe().



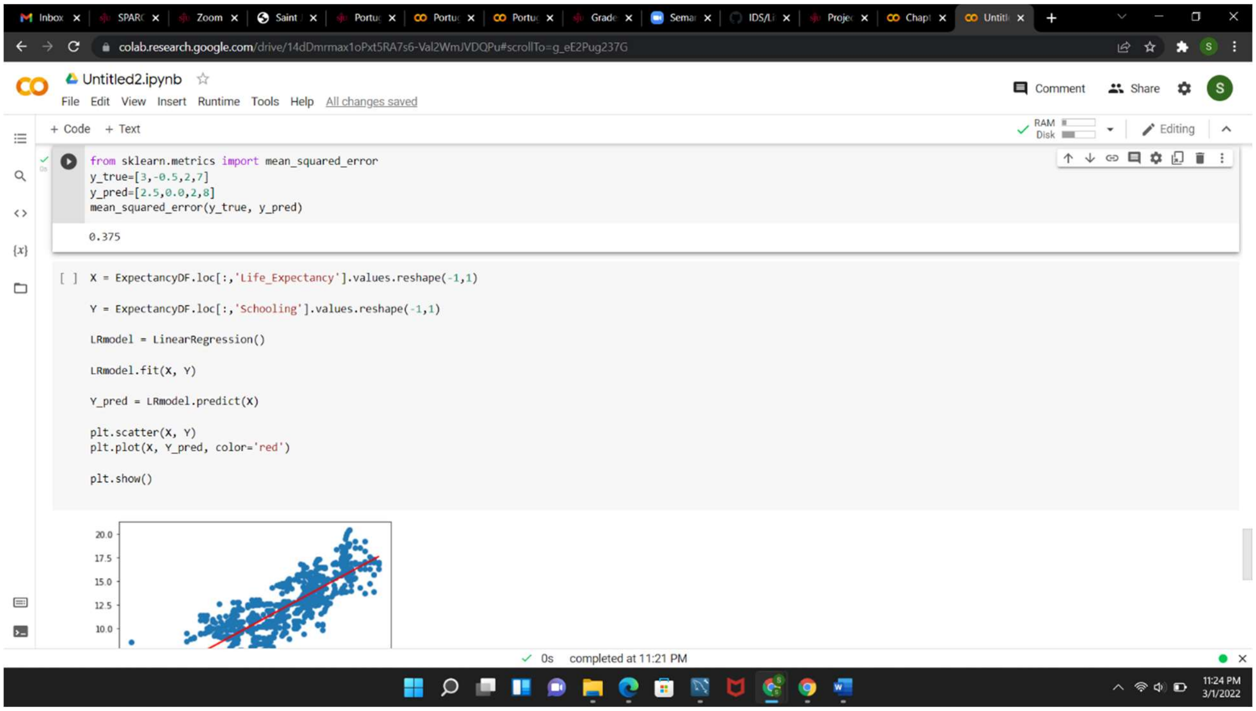
```
[10] ExpectancyDF.describe()
```

	Year	Life_Expectancy	Adult_Mortality	Alcohol	Percentage_Expenditure	BMI	Total_Expenditure	GDP	Population	Schooling
count	856.000000	856.000000	856.000000	856.000000	856.000000	856.000000	856.000000	856.000000	8.560000e+02	856.000000
mean	2012.496495	70.537430	158.870327	4.178458	1011.071696	39.870327	6.038026	7730.711121	1.335176e+07	12.644042
std	1.707907	8.886691	111.048267	3.795497	2327.538535	20.585439	2.409102	14796.802371	5.362515e+07	2.973853
min	2010.000000	36.300000	1.000000	0.010000	0.108056	2.000000	0.760000	8.376432	4.100000e+01	4.500000
25%	2011.000000	63.800000	70.500000	0.430000	67.694359	22.175000	4.752500	686.911106	2.483085e+05	10.600000
50%	2012.500000	72.700000	141.000000	3.570000	371.111647	45.000000	5.400000	2616.285584	1.515082e+06	12.700000
75%	2014.000000	76.200000	227.250000	6.892500	862.888706	58.500000	7.530000	7130.859259	8.276836e+06	14.800000
max	2015.000000	89.000000	682.000000	17.310000	19479.911610	83.300000	16.610000	119172.741800	1.293859e+09	20.400000

```
[11] ExpectancyDF.var()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a futur
  """Entry point for launching an IPython kernel.
Year                      2.916947e+00
Life_Expectancy           7.897327e+01
Adult_Mortality           1.233172e+04
Alcohol                   1.440580e+01
Percentage_Expenditure    5.417436e+06
BMI                       4.237603e+02
Total_Expenditure         5.803770e+00
GDP                       2.189454e+08
Population                2.875657e+15
Schooling                 8.843801e+00
dtype: float64
```

Fig: variance of expectancy

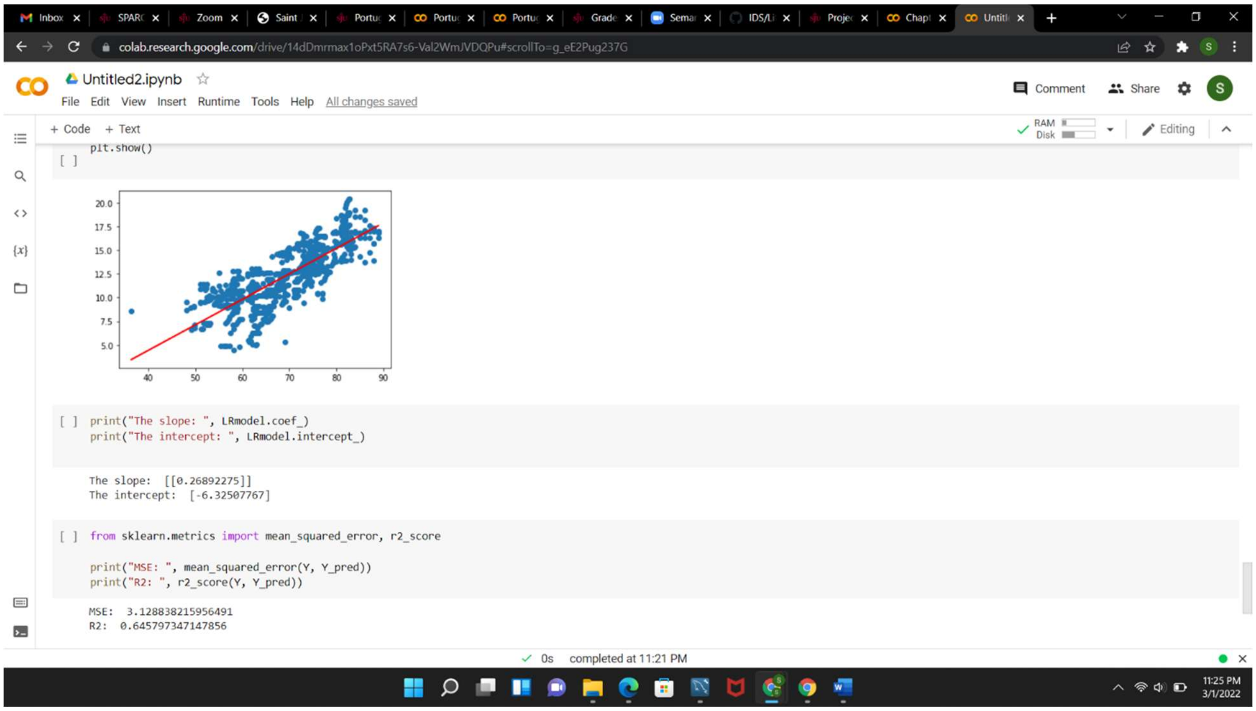

colab.research.google.com/drive/14dDmrmax1oPxt5RA7s6-Val2WmJVDQPu#scrollTo=g_eE2Pug237G
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
RAM Disk
Editing
20.0
17.5
15.0
12.5
10.0
0s completed at 11:21 PM
11:24 PM 3/1/2022

```
from sklearn.metrics import mean_squared_error
y_true=[3,-0.5,2,7]
y_pred=[2.5,0.0,2,8]
mean_squared_error(y_true, y_pred)
```

```
0.375
```

```
[ ] X = ExpectancyDF.loc[:,'Life_Expectancy'].values.reshape(-1,1)

    Y = ExpectancyDF.loc[:,'Schooling'].values.reshape(-1,1)

    LRmodel = LinearRegression()

    LRmodel.fit(X, Y)

    Y_pred = LRmodel.predict(X)

    plt.scatter(X, Y)
    plt.plot(X, Y_pred, color='red')

    plt.show()
```

Fig: plotting linear regression model


colab.research.google.com/drive/14dDmrmax1oPxt5RA7s6-Val2WmJVDQPu#scrollTo=g_eE2Pug237G
Untitled2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Comment Share
RAM Disk
Editing
20.0
17.5
15.0
12.5
10.0
7.5
5.0
40
50
60
70
80
90
0s completed at 11:21 PM
11:25 PM 3/1/2022

```
[ ] plt.show()
```

```
[ ] print("The slope: ", LRmodel.coef_)
    print("The intercept: ", LRmodel.intercept_)
```

```
The slope:  [[0.26892275]]
The intercept:  [-6.32507767]
```

```
[ ] from sklearn.metrics import mean_squared_error, r2_score

    print("MSE: ", mean_squared_error(Y, Y_pred))
    print("R2: ", r2_score(Y, Y_pred))
```

```
MSE:  3.128838215956491
R2:  0.645797347147856
```

Fig: Printing slope, intercept, MSE, R2 score

Conclusion: Several linear regression models for life expectancy as found in correlation analysis and I computed R2 and MSE values. After summarizing Regression model performs best.