untitled1

September 8, 2024

[1]:
```
pip install pandas numpy scipy scikit-learn statsmodels
```

```
Requirement already satisfied: pandas in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (2.2.2)
Requirement already satisfied: numpy in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (2.1.1)
Requirement already satisfied: scipy in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages
(1.14.1)
Requirement already satisfied: scikit-learn in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (1.5.1)
Requirement already satisfied: statsmodels in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages
(0.14.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
pandas) (2024.1)
Requirement already satisfied: joblib>=1.2.0 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
scikit-learn) (3.5.0)
Requirement already satisfied: patsy>=0.5.6 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
statsmodels) (0.5.6)
Requirement already satisfied: packaging>=21.3 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
statsmodels) (24.1)
Requirement already satisfied: six in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
patsy>=0.5.6->statsmodels) (1.16.0)
```

Note: you may need to restart the kernel to use updated packages.

[2]:
```
import pandas as pd
from sklearn.datasets import load_diabetes

# Load the dataset
diabetes = load_diabetes()
df = pd.DataFrame(data=diabetes.data, columns=diabetes.feature_names)
df['target'] = diabetes.target

# Display the first few rows
print(df.head())
```

```
        age       sex       bmi        bp        s1        s2        s3  \
0  0.038076  0.050680  0.061696  0.021872 -0.044223 -0.034821 -0.043401
1 -0.001882 -0.044642 -0.051474 -0.026328 -0.008449 -0.019163  0.074412
2  0.085299  0.050680  0.044451 -0.005670 -0.045599 -0.034194 -0.032356
3 -0.089063 -0.044642 -0.011595 -0.036656  0.012191  0.024991 -0.036038
4  0.005383 -0.044642 -0.036385  0.021872  0.003935  0.015596  0.008142

         s4        s5        s6  target
0 -0.002592  0.019907 -0.017646   151.0
1 -0.039493 -0.068332 -0.092204    75.0
2 -0.002592  0.002861 -0.025930   141.0
3  0.034309  0.022688 -0.009362   206.0
4 -0.002592 -0.031988 -0.046641   135.0
```

[ ]:
```

```

[3]:
```
# Calculate basic descriptive statistics
print("Mean:\n", df.mean())
```

```
Mean:
 age      -1.444295e-18
sex       2.543215e-18
bmi      -2.255925e-16
bp       -4.854086e-17
s1       -1.428596e-17
s2        3.898811e-17
s3       -6.028360e-18
s4       -1.788100e-17
s5        9.243486e-17
s6        1.351770e-17
target    1.521335e+02
dtype: float64
```

[4]:
```
print("\nMedian:\n", df.median())
```

```
Median:
 age         0.005383
sex         -0.044642
bmi         -0.007284
bp          -0.005670
s1          -0.004321
s2          -0.003819
s3          -0.006584
s4          -0.002592
s5          -0.001947
s6          -0.001078
target     140.500000
dtype: float64
```

[5]:
```
print("\nMode:\n", df.mode().iloc[0])
```

```
Mode:
 age         0.016281
sex        -0.044642
bmi        -0.030996
bp         -0.040099
s1         -0.037344
s2         -0.001001
s3         -0.013948
s4         -0.039493
s5         -0.018114
s6          0.003064
target     72.000000
Name: 0, dtype: float64
```

[6]:
```
print("\nStandard Deviation:\n", df.std())
```

```
Standard Deviation:
 age         0.047619
sex         0.047619
bmi         0.047619
bp          0.047619
s1          0.047619
s2          0.047619
s3          0.047619
s4          0.047619
s5          0.047619
s6          0.047619
target     77.093005
dtype: float64
```

[7]:
```
print("\nVariance:\n", df.var())
```

```
Variance:
 age          0.002268
sex           0.002268
bmi           0.002268
bp            0.002268
s1            0.002268
s2            0.002268
s3            0.002268
s4            0.002268
s5            0.002268
s6            0.002268
target    5943.331348
dtype: float64
```

[8]:
```
print("\nRange:\n", df.max() - df.min())
```

```
Range:
 age          0.217952
sex          0.095322
bmi          0.260831
bp           0.244442
s1           0.280694
s2           0.314401
s3           0.283486
s4           0.261629
s5           0.259694
s6           0.273379
target     321.000000
dtype: float64
```

[9]:
```
print("\nSkewness:\n", df.skew())
```

```
Skewness:
 age       -0.231382
sex         0.127385
bmi         0.598148
bp          0.290658
s1          0.378108
s2          0.436592
s3          0.799255
s4          0.735374
s5          0.291754
s6          0.207917
```

```
target    0.440563
dtype: float64
```

[10]:
```
print("\nKurtosis:\n", df.kurt())
```

```
Kurtosis:
 age      -0.671224
sex       -1.992811
bmi        0.095094
bp        -0.532797
s1         0.232948
s2         0.601381
s3         0.981507
s4         0.444402
s5        -0.134367
s6         0.236917
target    -0.883057
dtype: float64
```

[ ]:
```

```

[11]:
```
from scipy import stats

# Example data: BMI values
bmi_values = df['bmi']

# Hypothetical population mean for BMI
population_mean = 0.05

# Perform one-sample t-test
t_stat, p_value = stats.ttest_1samp(bmi_values, population_mean)

print(f"T-Statistic: {t_stat}")
print(f"P-Value: {p_value}")
```

```
T-Statistic: -22.074985843710174
P-Value: 2.7634312235044638e-73
```

[12]:
```
import numpy as np
from scipy import stats

# Sample mean and standard error for BMI
sample_mean = np.mean(bmi_values)
standard_error = stats.sem(bmi_values)

# Compute 95% confidence interval for BMI
```

```
confidence_interval = stats.norm.interval(0.95, loc=sample_mean,␣
 ↪scale=standard_error)

print(f"95% Confidence Interval for BMI: {confidence_interval}")
```

```
95% Confidence Interval for BMI: (np.float64(-0.004439332370169141),
np.float64(0.0044393323701686915))
```

[13]:
```
import statsmodels.api as sm

# Define independent variable (add constant for intercept)
X = sm.add_constant(df['bmi'])

# Define dependent variable
y = df['target']

# Fit linear regression model
model = sm.OLS(y, X).fit()

# Print model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 target   R-squared:                       0.344
Model:                            OLS   Adj. R-squared:                  0.342
Method:                 Least Squares   F-statistic:                     230.7
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           3.47e-42
Time:                        23:31:57   Log-Likelihood:                -2454.0
No. Observations:                 442   AIC:                             4912.
Df Residuals:                     440   BIC:                             4920.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        152.1335      2.974     51.162      0.000     146.289     157.978
bmi          949.4353     62.515     15.187      0.000     826.570    1072.301
==============================================================================
Omnibus:                       11.674   Durbin-Watson:                   1.848
Prob(Omnibus):                  0.003   Jarque-Bera (JB):                7.310
Skew:                           0.156   Prob(JB):                       0.0259
Kurtosis:                       2.453   Cond. No.                         21.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

specified.

[14]:
```
import pandas as pd
data = pd.read_csv(r"Covid Data.csv")
data.head()
```

[14]:

	USMER	MEDICAL_UNIT	SEX	PATIENT_TYPE	DATE_DIED	INTUBED	PNEUMONIA	\
0	2	1	1	1	03/05/2020	97	1	
1	2	1	2	1	03/06/2020	97	1	
2	2	1	2	2	09/06/2020	1	2	
3	2	1	1	1	12/06/2020	97	2	
4	2	1	2	1	21/06/2020	97	2	

	AGE	PREGNANT	DIABETES	…	ASTHMA	INMSUPR	HIPERTENSION	OTHER_DISEASE	\
0	65	2	2	…	2	2	1	2	
1	72	97	2	…	2	2	1	2	
2	55	97	1	…	2	2	2	2	
3	53	2	2	…	2	2	2	2	
4	68	97	1	…	2	2	1	2	

	CARDIOVASCULAR	OBESITY	RENAL_CHRONIC	TOBACCO	CLASIFFICATION_FINAL	ICU
0	2	2	2	2	3	97
1	2	1	1	2	5	97
2	2	2	2	2	3	2
3	2	2	2	2	7	97
4	2	2	2	2	3	97

[5 rows x 21 columns]

[15]:
```
covid_data = data.copy()
```

[16]:
```
len(covid_data)
```

[16]: 1048575

[17]:
```
data_date_died = covid_data["DATE_DIED"]
del covid_data["DATE_DIED"]
```

[18]:
```
# Calculate basic descriptive statistics
print("Mean:\n", covid_data.mean())
print("\nMedian:\n", covid_data.median())
print("\nMode:\n", covid_data.mode().iloc[0])
print("\nStandard Deviation:\n", covid_data.std())
print("\nVariance:\n", covid_data.var())

# Additional descriptive statistics
print("\nRange:\n", covid_data.max() - covid_data.min())
```

```
print("\nSkewness:\n", covid_data.skew())
print("\nKurtosis:\n", covid_data.kurt())
```

```
Mean:
 USMER                   1.632194
MEDICAL_UNIT             8.980565
SEX                      1.499259
PATIENT_TYPE             1.190765
INTUBED                 79.522875
PNEUMONIA                3.346831
AGE                     41.794102
PREGNANT                49.765585
DIABETES                 2.186404
COPD                     2.260569
ASTHMA                   2.242626
INMSUPR                  2.298132
HIPERTENSION             2.128989
OTHER_DISEASE            2.435143
CARDIOVASCULAR           2.261810
OBESITY                  2.125176
RENAL_CHRONIC            2.257180
TOBACCO                  2.214333
CLASIFFICATION_FINAL     5.305653
ICU                     79.553974
dtype: float64

Median:
 USMER                   2.0
MEDICAL_UNIT            12.0
SEX                      1.0
PATIENT_TYPE             1.0
INTUBED                 97.0
PNEUMONIA                2.0
AGE                     40.0
PREGNANT                97.0
DIABETES                 2.0
COPD                     2.0
ASTHMA                   2.0
INMSUPR                  2.0
HIPERTENSION             2.0
OTHER_DISEASE            2.0
CARDIOVASCULAR           2.0
OBESITY                  2.0
RENAL_CHRONIC            2.0
TOBACCO                  2.0
CLASIFFICATION_FINAL     6.0
ICU                     97.0
```

```
dtype: float64

Mode:
 USMER                   2
MEDICAL_UNIT            12
SEX                      1
PATIENT_TYPE             1
INTUBED                 97
PNEUMONIA                2
AGE                     30
PREGNANT                97
DIABETES                 2
COPD                     2
ASTHMA                   2
INMSUPR                  2
HIPERTENSION             2
OTHER_DISEASE            2
CARDIOVASCULAR           2
OBESITY                  2
RENAL_CHRONIC            2
TOBACCO                  2
CLASIFFICATION_FINAL     7
ICU                     97
Name: 0, dtype: int64

Standard Deviation:
 USMER                   0.482208
MEDICAL_UNIT             3.723278
SEX                      0.500000
PATIENT_TYPE             0.392904
INTUBED                 36.868886
PNEUMONIA               11.912881
AGE                     16.907389
PREGNANT                47.510733
DIABETES                 5.424242
COPD                     5.132258
ASTHMA                   5.114089
INMSUPR                  5.462843
HIPERTENSION             5.236397
OTHER_DISEASE            6.646676
CARDIOVASCULAR           5.194850
OBESITY                  5.175445
RENAL_CHRONIC            5.135354
TOBACCO                  5.323097
CLASIFFICATION_FINAL     1.881165
ICU                     36.823073
dtype: float64
```

```
Variance:
 USMER                     0.232525
MEDICAL_UNIT              13.862797
SEX                        0.250000
PATIENT_TYPE               0.154374
INTUBED                 1359.314775
PNEUMONIA                141.916736
AGE                      285.859810
PREGNANT                2257.269723
DIABETES                   29.422399
COPD                       26.340073
ASTHMA                     26.153909
INMSUPR                    29.842656
HIPERTENSION               27.419855
OTHER_DISEASE              44.178296
CARDIOVASCULAR             26.986470
OBESITY                    26.785232
RENAL_CHRONIC              26.371859
TOBACCO                    28.335364
CLASIFFICATION_FINAL        3.538783
ICU                     1355.938731
dtype: float64

Range:
 USMER                     1
MEDICAL_UNIT              12
SEX                        1
PATIENT_TYPE               1
INTUBED                   98
PNEUMONIA                 98
AGE                      121
PREGNANT                  97
DIABETES                  97
COPD                      97
ASTHMA                    97
INMSUPR                   97
HIPERTENSION              97
OTHER_DISEASE             97
CARDIOVASCULAR            97
OBESITY                   97
RENAL_CHRONIC             97
TOBACCO                   97
CLASIFFICATION_FINAL       6
ICU                       98
dtype: int64

Skewness:
 USMER                    -0.548288
```

```
MEDICAL_UNIT            -0.515686
SEX                      0.002962
PATIENT_TYPE             1.574104
INTUBED                 -1.632934
PNEUMONIA                7.898181
AGE                      0.283560
PREGNANT                -0.011364
DIABETES                17.543677
COPD                    18.590718
ASTHMA                  18.649687
INMSUPR                 17.453713
HIPERTENSION            18.165618
OTHER_DISEASE           14.299725
CARDIOVASCULAR          18.361786
OBESITY                 18.380794
RENAL_CHRONIC           18.577582
TOBACCO                 17.891278
CLASIFFICATION_FINAL    -0.424923
ICU                     -1.634236
dtype: float64

Kurtosis:
 USMER                   -1.699384
MEDICAL_UNIT             -1.637617
SEX                      -1.999995
PATIENT_TYPE              0.477805
INTUBED                   0.666747
PNEUMONIA                60.436221
AGE                       0.064148
PREGNANT                 -1.999854
DIABETES                306.914929
COPD                    343.807287
ASTHMA                  346.212449
INMSUPR                 302.773404
HIPERTENSION            329.617021
OTHER_DISEASE           202.607826
CARDIOVASCULAR          335.406349
OBESITY                 337.534031
RENAL_CHRONIC           343.366457
TOBACCO                 318.961602
CLASIFFICATION_FINAL     -1.620642
ICU                       0.670949
dtype: float64
```

[19]:
```
df.columns
```

[19]:
```
Index(['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6',
       'target'],
      dtype='object')
```

[20]:
```
from scipy import stats

# Example data: BMI values
bmi_values = df['bmi']

# Hypothetical population mean for BMI
population_mean = 0.05

# Perform one-sample t-test
t_stat, p_value = stats.ttest_1samp(bmi_values, population_mean)

print(f"T-Statistic: {t_stat}")
print(f"P-Value: {p_value}")
```

```
T-Statistic: -22.074985843710174
P-Value: 2.7634312235044638e-73
```

[21]:
```
import numpy as np
from scipy import stats

# Sample mean and standard error for BMI
sample_mean = np.mean(bmi_values)
standard_error = stats.sem(bmi_values)

# Compute 95% confidence interval for BMI
confidence_interval = stats.norm.interval(0.95, loc=sample_mean,␣
 ↪scale=standard_error)

print(f"95% Confidence Interval for BMI: {confidence_interval}")
```

```
95% Confidence Interval for BMI: (np.float64(-0.004439332370169141),
np.float64(0.0044393323701686915))
```

[22]:
```
import statsmodels.api as sm

# Define independent variable (add constant for intercept)
X = sm.add_constant(df['bmi'])

# Define dependent variable
y = df['target']

# Fit linear regression model
model = sm.OLS(y, X).fit()
```

```
# Print model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 target   R-squared:                       0.344
Model:                            OLS   Adj. R-squared:                  0.342
Method:                 Least Squares   F-statistic:                     230.7
Date:                Sun, 08 Sep 2024   Prob (F-statistic):           3.47e-42
Time:                        23:32:01   Log-Likelihood:                -2454.0
No. Observations:                 442   AIC:                             4912.
Df Residuals:                     440   BIC:                             4920.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        152.1335      2.974     51.162      0.000     146.289     157.978
bmi          949.4353     62.515     15.187      0.000     826.570    1072.301
==============================================================================
Omnibus:                       11.674   Durbin-Watson:                   1.848
Prob(Omnibus):                  0.003   Jarque-Bera (JB):                7.310
Skew:                           0.156   Prob(JB):                       0.0259
Kurtosis:                       2.453   Cond. No.                         21.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[23]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (4.53.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (1.4.7)
Requirement already satisfied: numpy>=1.23 in
```

```
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (2.1.1)
Requirement already satisfied: packaging>=20.0 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[24]:
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages
(0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
seaborn) (2.1.1)
Requirement already satisfied: pandas>=1.2 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
seaborn) (2.2.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
seaborn) (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.3.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (4.53.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.4.7)
Requirement already satisfied: packaging>=20.0 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
```

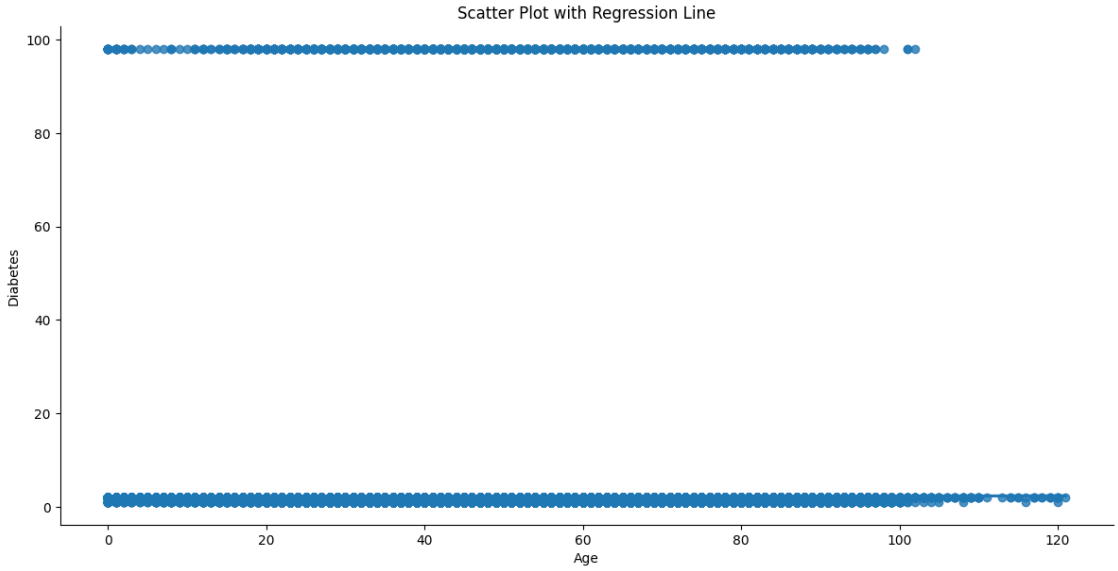
```
Requirement already satisfied: pillow>=8 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: six>=1.5 in
c:\users\pedag\appdata\local\programs\python\python312\lib\site-packages (from
python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

[25]:
```python
import seaborn as sns
import matplotlib.pyplot as plt

# Example: Assuming 'x_variable' and 'y_variable' are column names in the␣
 ↪dataset
sns.lmplot(x='AGE', y='DIABETES', data=covid_data, aspect=2, height=6)

plt.title('Scatter Plot with Regression Line')
plt.xlabel('Age')
plt.ylabel('Diabetes')

plt.show()
```


Scatter Plot with Regression Line
Diabetes
Age
0
20
40
60
80
100
120