Lab Assignment – 1

Implementing Multilinear and linear regression

Code:-

```
import pandas as pd
import numpy as np
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
from sklearn.model_selection import train_test_split
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

```
# Load data and check data
mydata = pd.read_csv("self noise.csv")
print(mydata.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1503 entries, 0 to 1502
Data columns (total 6 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Frquency(Hz)         1503 non-null   int64  
 1   Angle_of_Attack      1503 non-null   float64
 2   Chord_Length         1503 non-null   float64
 3   Free_stream_velocity  1503 non-null   float64
 4   Displacement         1503 non-null   float64
 5   Sound_pressure_level  1503 non-null   float64
dtypes: float64(5), int64(1)
memory usage: 70.6 KB
None
```

```
mydata.head()
```

	Frquency(Hz)	Angle_of_Attack	Chord_Length	Free_stream_velocity	Displacement	Sound_pressure_level
0	800	0.0	0.3048	71.3	0.002663	126.201
1	1000	0.0	0.3048	71.3	0.002663	125.201
2	1250	0.0	0.3048	71.3	0.002663	125.951
3	1600	0.0	0.3048	71.3	0.002663	127.591
4	2000	0.0	0.3048	71.3	0.002663	127.461

```
#checking missing values
missing_values = mydata.isna().sum()
missing_values
```

```
Frquency(Hz)            0
Angle_of_Attack         0
Chord_Length            0
Free_stream_velocity    0
Displacement            0
Sound_pressure_level    0
dtype: int64
```

```python
# Checking correlation matrix
correlation_matrix = mydata.corr()
print(correlation_matrix)
```

```
                       Frquency(Hz)  Angle_of_Attack  Chord_Length  \
Frquency(Hz)               1.000000        -0.272765     -0.003661   
Angle_of_Attack           -0.272765         1.000000     -0.504868   
Chord_Length              -0.003661        -0.504868      1.000000   
Free_stream_velocity       0.133664         0.058760      0.003787   
Displacement              -0.230107         0.753394     -0.220842   
Sound_pressure_level      -0.390711        -0.156108     -0.236162   

                       Free_stream_velocity  Displacement  Sound_pressure_level  
Frquency(Hz)                       0.133664     -0.230107             -0.390711  
Angle_of_Attack                    0.058760      0.753394             -0.156108  
Chord_Length                       0.003787     -0.220842             -0.236162  
Free_stream_velocity               1.000000     -0.003974              0.125103  
Displacement                      -0.003974      1.000000             -0.312670  
Sound_pressure_level               0.125103     -0.312670              1.000000  
```

```python
X = mydata.drop(columns=['Sound_pressure_level'])
```

```python
X = sm.add_constant(X)
```

```python
y = mydata['Sound_pressure_level']
```

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
model = sm.OLS(y_train, X_train).fit()
```

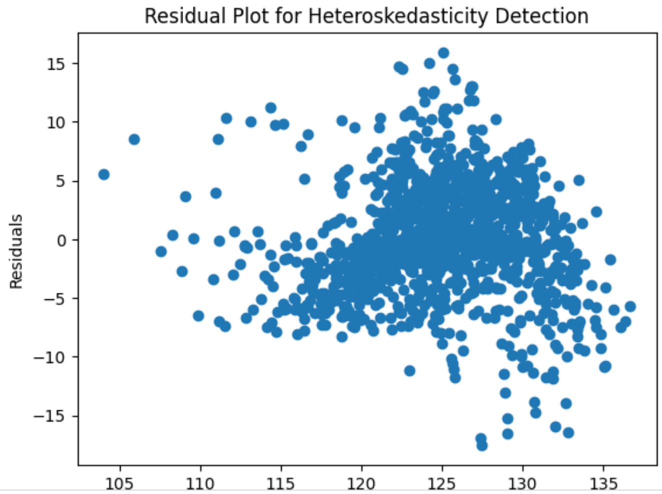
```python
y_pred = model.predict(X_test)
```

```python
r_squared = r2_score(y_test, y_pred)
adj_r_squared = model.rsquared_adj
f_statistic = model.fvalue
p_value = model.f_pvalue
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
mae = mean_absolute_error(y_test, y_pred)
```

```python
# Print the results
print(f"R-squared: {r_squared}")
print(f"Adjusted R-squared: {adj_r_squared}")
print(f"F-statistics: {f_statistic}, p-value: {p_value}")
print(f"MSE: {mse}")
print(f"RMSE: {rmse}")
print(f"MAE: {mae}")
```

```
R-squared: 0.5582979754896895
Adjusted R-squared: 0.5013716488971148
F-statistics: 242.5215055435907, p-value: 6.080414968968652e-179
MSE: 22.128643318249228
RMSE: 4.704109194975094
MAE: 3.6724145641747024
```

```python
residuals = model.resid
plt.scatter(model.predict(), residuals)
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot for Heteroskedasticity Detection')
plt.show()
```



Interpretation:-

The model explains approximately 55.8% of the variability in the dependent variable. The F-statistic is highly significant ($p < 0.001$), indicating that the model as a whole is significant. The mean squared error (MSE) is 22.13, and the root mean squared error (RMSE) is 4.70, suggesting the model's predictions are reasonably close to the actual values on average. The mean absolute error (MAE) is 3.67, indicating the average absolute difference between predicted and actual values. Overall, the model demonstrates good explanatory power and predictive accuracy, though further validation may be necessary.