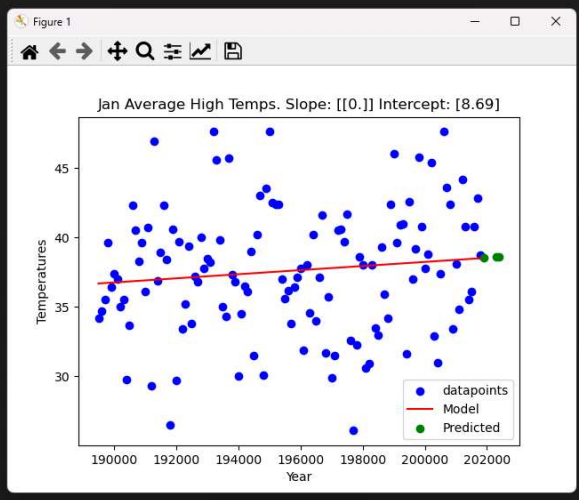
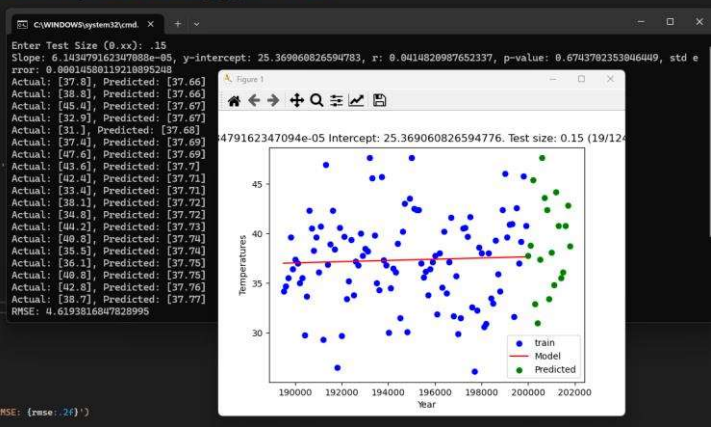
1.

```python
from matplotlib import pyplot as plt
from scipy import stats
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression

np.set_printoptions(precision=2)

reg = LinearRegression()

data = pd.read_csv(r"C:\Users\Yeyian PC\source\repos\MLCumulativeNotesandAssignemnts\MLCumulativeNotesandAssignemnts\HWs\HomeWork4\avgHigh_jan_1895-2018.csv")
pred_point = np.array([201901, 202301, 202401]).reshape(-1, 1)


X = np.array(data.iloc[:,0]).reshape(-1, 1)
y = np.array(data.iloc[:,1]).reshape(-1, 1)

reg.fit(X, y)

plt.scatter(X, y, color='b', label = 'datapoints')
plt.plot(X, reg.predict(X), color='r', label= 'Model')
plt.scatter(pred_point, reg.predict(pred_point), color='g', label= 'Predicted')
plt.legend(loc='lower right')
plt.title(f'Jan Average High Temps. Slope: {reg.coef_} Intercept: {reg.intercept_}')
plt.ylabel('Temperatures')
plt.xlabel('Year')
plt.show()
```



2.

```python
from matplotlib import pyplot as plt
from scipy import stats
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

np.set_printoptions(precision=2)

reg = LinearRegression()

data = pd.read_csv(r"C:\Users\Yeyian PC\source\repos\MLCumulativeNotesandAssignemnts\MLCumulativeNotesandAssignemnts\HWs\HomeWork4\avgHigh_jan_1895-2018.csv")
pred_point = np.array([201901, 202301, 202401]).reshape(-1, 1)


X = np.array(data.iloc[:,0]).reshape(-1, 1)
y = np.array(data.iloc[:,1]).reshape(-1, 1)

userin = float(input('Enter Test Size (0.xx): '))

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=userin, shuffle=False)

reg.fit(X_train, y_train)
predicted = np.array(reg.predict(X_test)).reshape(-1, 1)

slope, intercept, r, p, std_error = stats.linregress(X_train.flatten(), y_train.flatten())
print(f'Slope: {slope}, y-intercept: {intercept}, r: {r}, p-value: {p}, std error: {std_error}')

for i in range(len(X_test)):
    print(f'Actual: {y_test[i]}, Predicted: {predicted[i]}')



rmse = np.sqrt(np.mean((y_test - predicted) ** 2))


print(f'RMSE: {rmse}')

inter = float(reg.intercept_)
coeff = float(reg.coef_)


plt.scatter(X_train, y_train, color='b', label = 'train')
plt.plot(X_train, reg.predict(X_train), color='r', label= 'Model')
plt.scatter(X_test, y_test, color='g', label= 'Predicted')
plt.legend(loc='lower right')
plt.title(f'Slope: {coeff} Intercept: {inter}. Test size: {userin} ({len(X_test)}/{len(X)}), RMSE: {rmse:.2f}')
plt.ylabel('Temperatures')
plt.xlabel('Year')
plt.show()
```



3.

```python
from matplotlib import pyplot as plt
from scipy import stats
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

np.set_printoptions(precision=2)

reg = LinearRegression()

data = pd.read_csv(r"C:\Users\Yeyian PC\source\repos\MLCumulativeNotesandAssignemnts\MLCumulativeNotesandAssignemnts\HWs\HomeWork4\materials.csv")
pred_point = np.array([[32.1, 37.5, 128.95],
                       [36.9, 35.37, 130.03]])


X = np.array(data.loc[:,"Time":"Temperature"])
y = np.array(data.loc[:,"Strength"])

slope, intercept, r, p, std_error = stats.linregress(X[:,0], y)
print('Time vs Strength')
print(f'Slope: {slope}, r: {r}')

slope, intercept, r, p, std_error = stats.linregress(X[:,1], y)
print('Time vs Strength')
print(f'Slope: {slope}, r: {r}')

slope, intercept, r, p, std_error = stats.linregress(X[:,2], y)
print('Temperature vs Strength')
print(f'Slope: {slope}, r: {r}')


reg.fit(X,y)


for i in range(2):
    Ymodel = reg.intercept_
    for j in range(3):
        Ymodel = Ymodel + reg.coef_[j]*pred_point[i,j]
    print(f'Prediction for {pred_point[i,:]}: {Ymodel}')
```

```
C:\WINDOWS\system32\cmd.
Time vs Strength
Slope: 1.6540206374756767, r: 0.10235205191580073
Time vs Strength
Slope: 1.3852090083361166, r: 0.11209136865377828
Temperature vs Strength
Slope: -2.6753072861763023, r: -0.8488369669180945
Prediction for [ 32.1   37.5  128.95]: 267.82895127641154
Prediction for [ 36.9   35.37 130.03]: 263.441518393183
Press any key to continue . . .
```

4.

```python
from matplotlib import pyplot as plt
from scipy import stats
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

np.set_printoptions(precision=2)

reg = LinearRegression()

data = pd.read_csv(r"C:\Users\Yeyian PC\source\repos\MLCumulativeNotesandAssignemnts\MLCumulativeNotesandAssignemnts\HWs\HomeWork4\materials.csv")
pred_point = np.array([[32.1, 37.5, 128.95],
                       [36.9, 35.37, 130.03]])

X = np.array(data.loc[:,["Pressure","Temperature"]])
Pressure = X[:, 0]
Temperature = X[:, 1]

y = np.array(data.loc[:,"Strength"])

reg.fit(X,y)


X1, X2 = np.meshgrid(Pressure, Temperature)

Z = reg.intercept_ + reg.coef_[0]*X1 + reg.coef_[1]*X2

#3D plot
fig = plt.figure()
ax = plt.axes(projection = '3d')
ax.plot_wireframe(X1, X2, Z, color = 'blue')
#3D scattet plot (data points)
ax.scatter3D(Pressure, Temperature, y, c=y, cmap='Greens')
ax.set_title('3D Graph')
ax.set_xlabel('Pressure')
ax.set_zlabel('Strength')
ax.set_ylabel('Temperature')
plt.show()
```

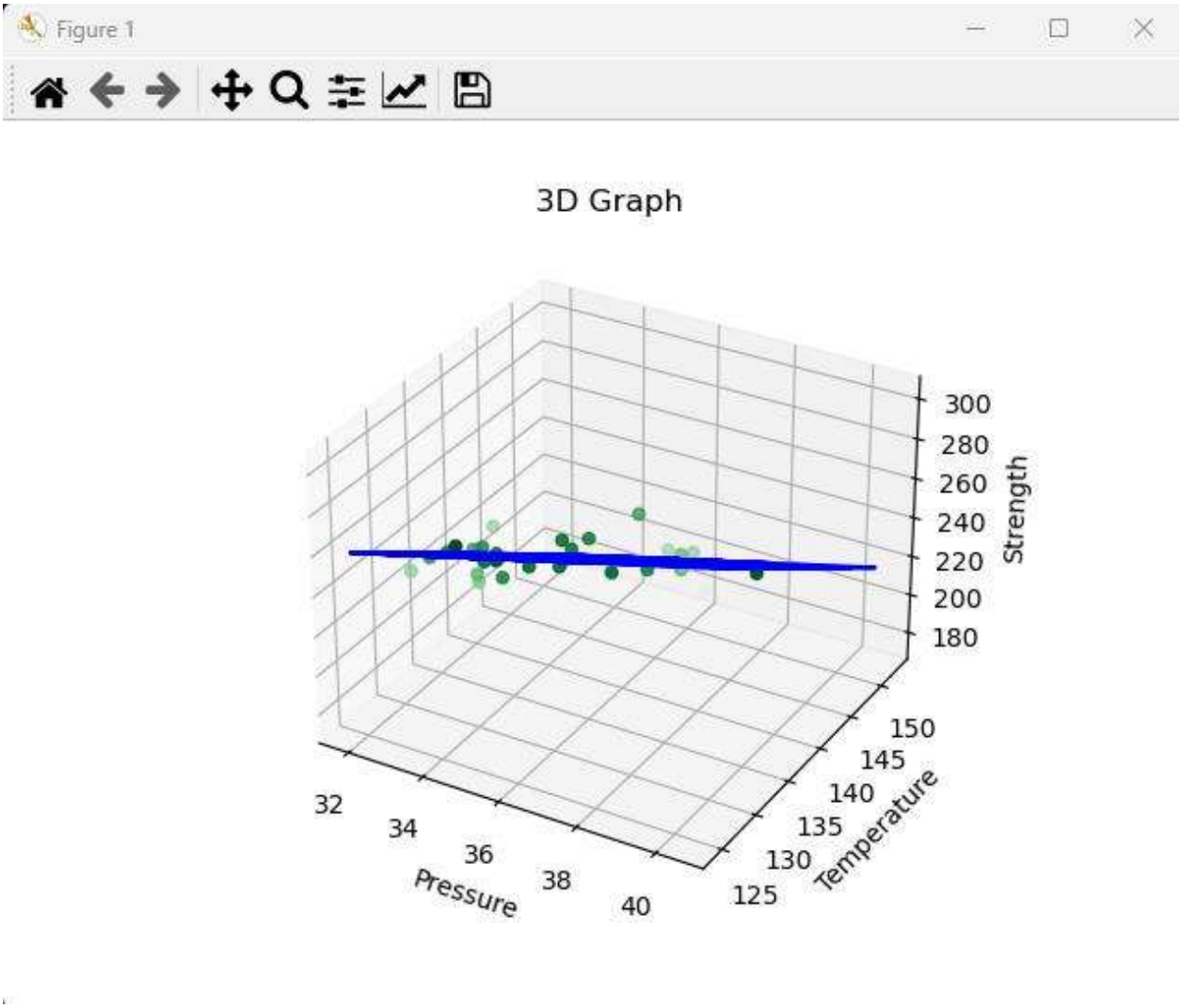


Figure 1
3D Graph
300
280
260
240
220
200
180
Strength
150
145
140
135
130
125
Temperature
32
34
36
38
40
Pressure

5.

```python
from matplotlib import pyplot as plt
from scipy import stats
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import RANSACRegressor

np.set_printoptions(precision=2)

data = pd.read_csv(r"C:\Users\Yeyian PC\source\repos\MLCumulativeNotesandAssignemnts\MLCumulativeNotesandAssignemnts\HWs\HomeWork4\materialsOutliers.csv")

data2 = data.copy()

X = np.array(data.loc[:,"Time":"Temperature"])
y = np.array(data.loc[:,"Strength"])

reg = LinearRegression()
reg.fit(X,y)
print("Before RANSAC:")
print(f'Coeff: {reg.coef_}') #print all coefficients
print(f'Y-intercept: {reg.intercept_}') #print all coefficients
Rsquared = reg.score(X,y)
print(f'R2: {Rsquared}') #print all coefficients

cumulative_mask = np.array([])
for i in range(X.shape[1]):
    ransac = RANSACRegressor(random_state=0, residual_threshold=15, stop_probability=1.00)
    ransac.fit(y.reshape(-1,1), X[:,i])
    inlier_mask = ransac.inlier_mask_
    if i == 0:
        cumulative_mask = inlier_mask
    else:
        cumulative_mask = cumulative_mask & inlier_mask



#use the combined inlier mask to filter the rows in the original df
data2 = data2[cumulative_mask]

X = np.array(data2.loc[:,"Time":"Temperature"])
y = np.array(data2.loc[:,"Strength"])

reg = LinearRegression()
reg.fit(X,y)
print("\nAfter RANSAC:")
print(f'Coeff: {reg.coef_}') #print all coefficients
print(f'Y-intercept: {reg.intercept_}') #print all coefficients
Rsquared = reg.score(X,y)
print(f'R2: {Rsquared}') #print all coefficients

```

```
C:\WINDOWS\system32\cmd.
Before RANSAC:
Coeff: [-0.04 -0.18 -0.37]
Y-intercept: 306.53593734084 83
R2: 0.15565291411636206

After RANSAC:
Coeff: [ 2.12  5.32 -3.02]
Y-intercept: 389.1659157434116
R2: 0.8741267827401 25
Press any key to continue . . .
```