

1.

```
1 from sklearn.preprocessing import StandardScaler
2 from matplotlib import pyplot as plt
3 from scipy import stats
4 import numpy as np
5 import pandas as pd
6 from sklearn.linear_model import LinearRegression
7
8 data = pd.read_csv('vehicles.csv')
9 X= np.array(data.loc[:, 'cyl:'])
10 y = np.array(data.loc[:, 'mpg'])
11 data = data.drop(['make', 'mpg'], axis =1) #dropping columns
12 #column names are not included in the data.columns that is called later
13
14 #print(X)
15
16 xScaledData = StandardScaler()
17 xScaled = xScaledData.fit_transform(X)
18
19 reg = LinearRegression().fit(xScaled, y)#between scaledX and y
20
21 print(f'Weighted Coeff (scaled X): {reg.coef_}')
22 print(f'Intercept: {reg.intercept_}')
23
24 temp = abs(reg.coef_)
25 fiveBest = data.columns[temp.argsort()[-5:]]
26 print(f'\nBest Coefficients: {fiveBest}')
27
28
29 dataPoint = xScaledData.transform([[6, 163, 111, 3.9, 2.77, 16.45, 0, 1, 4, 4]])
30 pred = reg.predict(dataPoint)
```

C:\WINDOWS\system32\cmd. x + v

Weighted Coeff (scaled X): [-0.19588953 1.626723 -1.44967936 0.4142235 -3.57801491 1.444047
0.15763526 1.23776483 0.47595069 -0.31702919]
Intercept: 20.090625000000003

Best Coefficients: Index(['am', 'qsec', 'hp', 'disp', 'wt'], dtype='object')
Press any key to continue . . .

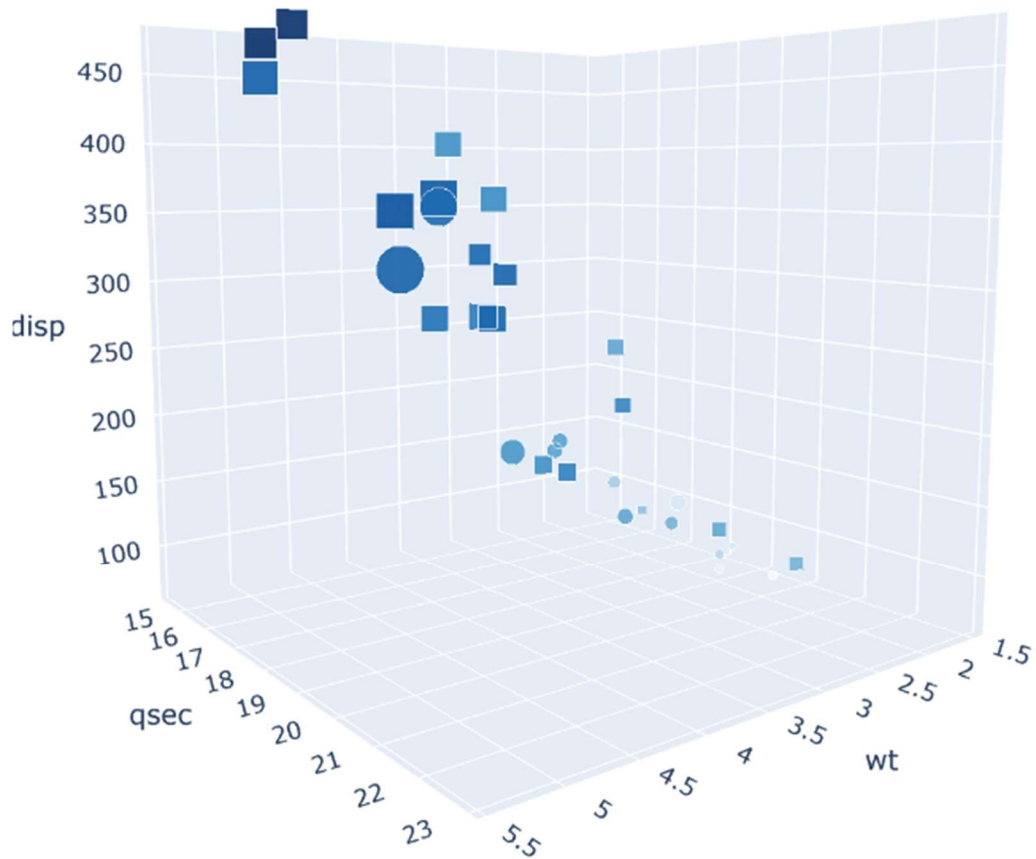
2.

```
2 import plotly
3 import plotly.graph_objs as go
4 from sklearn.preprocessing import StandardScaler
5 from matplotlib import pyplot as plt
6 from scipy import stats
7 import numpy as np
8 import pandas as pd
9 from sklearn.linear_model import LinearRegression
10
11 data = pd.read_csv('vehicles.csv')
12 X= np.array(data.loc[:, 'cyl:'])
13 y_mpg = np.array(data.loc[:, 'mpg'])
14 data = data.drop(['make', 'mpg'], axis =1) #dropping columns make and mpg from data so their
15 #column names are not included in the data.columns that is called later
16
17 print(X)
18
19 xScaledData = StandardScaler()
20 xScaled = xScaledData.fit_transform(X)
21
22 reg = LinearRegression().fit(xScaled, y_mpg)#between scaledX and y
23
24 print(f'Weighted Coeff (scaled X): {reg.coef_}')
25 print(f'Intercept: {reg.intercept_}')
26
27 temp = abs(reg.coef_)
28 fiveBestColIndex = temp.argsort()[-5:]
29
30 bestX = data.iloc[:, fiveBestColIndex]
31 print(bestX)
32
33 markersize = data['hp']/12
34 markercolor = y_mpg
35 markershape = data['am'].map({1: 'circle', 0: 'square'})
36 # ['am', 'qsec', 'hp', 'disp', 'wt']
37
```

```

37
38 #Make Plotly figure
39 fig1 = go.Scatter3d(x=data['wt'],
40                    y=data['qsec'],
41                    z=data['disp'],
42                    marker=dict(size=markersize,
43                                color=markercolor,
44                                symbol=markershape,
45                                opacity=0.9,
46                                reversescale=True,
47                                colorscale='Blues'),
48                    line=dict(width=0.02),
49                    mode='markers')
50
51 #Make Plot.ly Layout
52 mylayout = go.Layout(scene=dict(xaxis=dict(title="wt"),
53                                     yaxis=dict(title="qsec"),
54                                     zaxis=dict(title="disp")),)
55
56 #Plot and save html
57 plotly.offline.plot({"data": [fig1],
58                     "layout": mylayout},
59                     auto_open=True,
60                     filename=("6DPlot.html"))

```



3.

```
1 from sklearn.preprocessing import StandardScaler
2 from matplotlib import pyplot as plt
3 from scipy import stats
4 import numpy as np
5 import pandas as pd
6 from sklearn.linear_model import LinearRegression
7
8 data = pd.read_csv('vehicles.csv')
9 X = np.array(data.loc[:, 'cyl':])
10 y = np.array(data.loc[:, 'mpg'])
11 data = data.drop(['make', 'mpg'], axis=1) #dropping columns make and mpg from data so their
12                                         #column names are not included in the data.columns that is called later
13
14 #print(X)
15
16 xScaledData = StandardScaler()
17 xScaled = xScaledData.fit_transform(X)
18
19 reg = LinearRegression().fit(xScaled, y) #between scaledX and y
20
21
22
23 #problem did not specify to use the best coefficients found before so i used all of them
24 print(f'Weighted Coeff (scaled X): {reg.coef_}')
25 print(f'Intercept: {reg.intercept_}')
26
27 dataPoint = xScaledData.transform([[6, 163, 111, 3.9, 2.77, 16.45, 0, 1, 4, 4]])
28 pred = reg.predict(dataPoint)
29
30 print(f'\nPredicted mpg: {pred}')
```

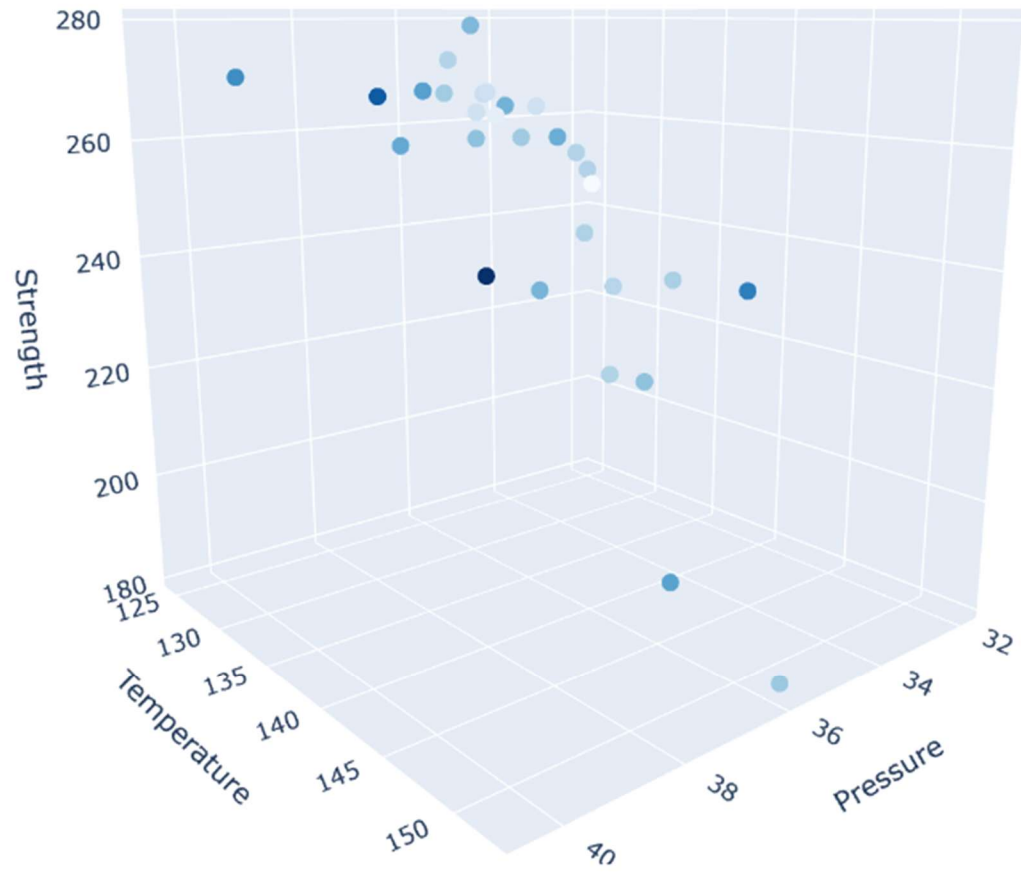
C:\WINDOWS\system32\cmd. × + ▾

Weighted Coeff (scaled X): [-0.19588953 1.626723 -1.44967936 0.4142235 -3.57801491 1.444047
0.15763526 1.23776483 0.47595069 -0.31702919]
Intercept: 20.090625000000003

Best Coefficients: Index(['am', 'qsec', 'hp', 'disp', 'wt'], dtype='object')

Predicted mpg: [22.05252337]
Press any key to continue . . .

4.



5.

