LinearRegressionApplied

November 29, 2025

1 Linear Regression Applied

1.1 Import the Data

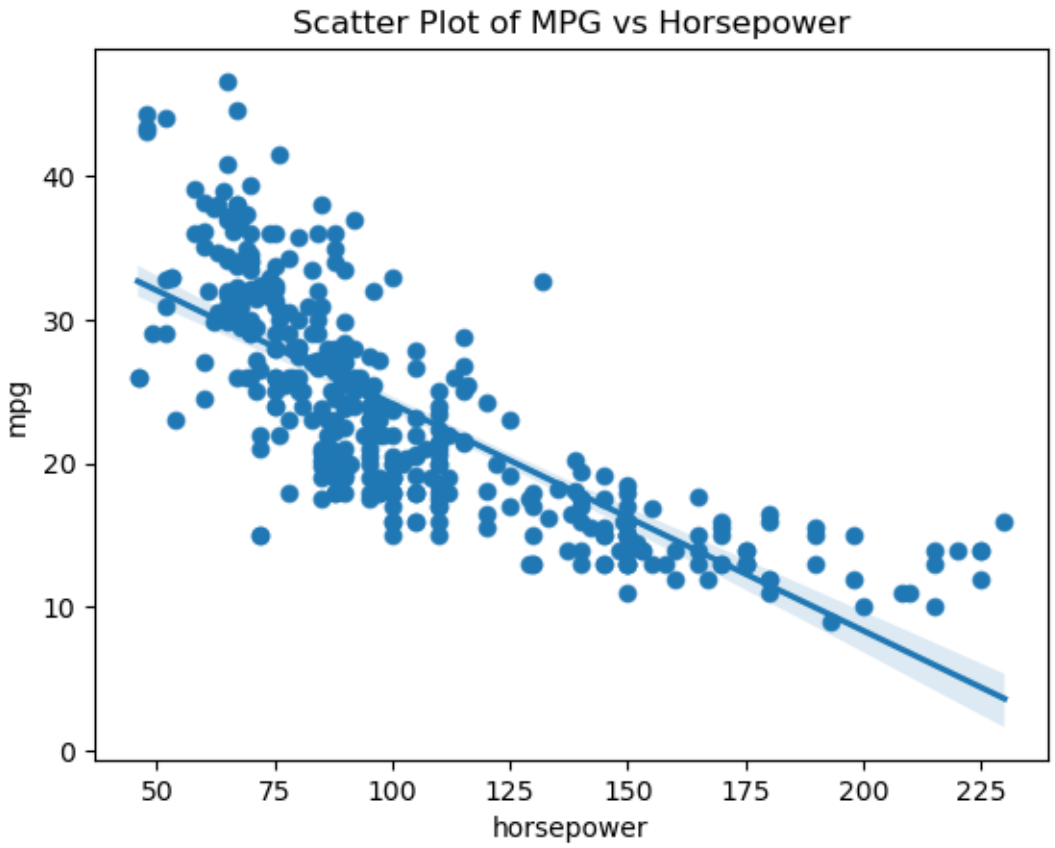
[21]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from math import log
from sklearn import linear_model
import statsmodels.api as sm
import seaborn as sns
from statsmodels.stats.anova import anova_lm
from statsmodels.formula.api import ols



Autodata = pd.read_csv('Auto.csv')
print(Autodata)
```

```
      mpg  cylinders  displacement horsepower  weight  acceleration  year  \
0    18.0          8         307.0        130    3504          12.0    70
1    15.0          8         350.0        165    3693          11.5    70
2    18.0          8         318.0        150    3436          11.0    70
3    16.0          8         304.0        150    3433          12.0    70
4    17.0          8         302.0        140    3449          10.5    70
..    …          …           …          …      …             …     …
392  27.0          4         140.0         86    2790          15.6    82
393  44.0          4          97.0         52    2130          24.6    82
394  32.0          4         135.0         84    2295          11.6    82
395  28.0          4         120.0         79    2625          18.6    82
396  31.0          4         119.0         82    2720          19.4    82

     origin                       name
0         1  chevrolet chevelle malibu
1         1          buick skylark 320
2         1         plymouth satellite
3         1              amc rebel sst
```

```
4         1                ford torino
..      …                        …
392       1            ford mustang gl
393       2                  vw pickup
394       1              dodge rampage
395       1                ford ranger
396       1                 chevy s-10

[397 rows x 9 columns]
```

1.2 Part A: Graphs

[6]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
from math import log
from sklearn import linear_model
import statsmodels.api as sm


Autodata = pd.read_csv('Auto.csv')

df = pd.DataFrame(Autodata)

df['horsepower'] = pd.to_numeric(df['horsepower'], errors='coerce')
df = df.dropna(subset=['horsepower'])


sns.scatterplot(data=df, x='horsepower', y='mpg')
plt.title('Scatter Plot of MPG vs Horsepower')
plt.xlabel('Horsepower')
plt.ylabel('MPG')

x = df['horsepower']
y = df['mpg']
X = sm.add_constant(x)


regression = sm.OLS(y,X).fit() #makes the actual regression
sns.regplot(x='horsepower', y='mpg', data=df) #simply visualises it
plt.show()
print(regression.summary())
```



OLS Regression Results

Dep. Variable:	mpg	R-squared:	0.606
Model:	OLS	Adj. R-squared:	0.605
Method:	Least Squares	F-statistic:	599.7
Date:	Sat, 29 Nov 2025	Prob (F-statistic):	7.03e-81
Time:	17:18:41	Log-Likelihood:	-1178.7
No. Observations:	392	AIC:	2361.
Df Residuals:	390	BIC:	2369.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	39.9359	0.717	55.660	0.000	38.525	41.347
horsepower	-0.1578	0.006	-24.489	0.000	-0.171	-0.145

Omnibus:	16.432	Durbin-Watson:	0.920
Prob(Omnibus):	0.000	Jarque-Bera (JB):	17.305
Skew:	0.492	Prob(JB):	0.000175

```
Kurtosis:                       3.299   Cond. No.                         322.
==============================================================================

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

1.3 Questions:

i. Is there a relationship between the predictor and the response?

There is infact a relationship between the predictor and the response. In this specific example the predictor is the Horsepower, and the response in the MPG (miles per gallon). Now if we think logically and critically without the datasets, regression, or the summary, there is obviously going to be a relationship between the two, because the higher the horsepower, the most fuel it consumes and the lower the miles per gallon. However in this exercise we want to look at one specific point, the P> |t| value in this summary.

The P-Value is the value that takes random points on the graph to see if it disproves the Null hypothesis. The null hypothesis, is the hypothesis that the predictor and response do not have a relationship with eachother. Now this value clearly needs to be smaller than 0.05. This value is chosen because it aligns with the 95% Internval of Confidence in this, meaning that there is a 5% chance that it is just a random outlier that doesn't have anything to do with the rest of the results.

Our P-value is 0.000, which is smaller than 0.05, thus there is a relationship between the 2 values. THe 0.000 is rounded down from an insanely small number, and even by looking at the graph we can see that there is clearly a trend going down as the horsepower increases.

ii. How strong is the relationship between the predictor and the response?

R-Squared is one of the most useful pieces of information about any data set that you can have, and this along with P-value can tell you about the relationship of data. R-Squared is used to see the correlation between 2 variables, and it sees how well it follows the regression line. It runs between 0-1, with the closer to 1 it gets, the stronger the relationship and the accuracy of the data in regards to the regression line. The R-Squared is all about prediction, how closely the data follows the trend identified by the regression, which brings us to our value. Our R-squared value is 0.606, which put into simple terms says that the horsepower explains 60.6% the MPG, and that 60.6% is clustered around the line of regression.

iii. Is the relationship between the predictor and the response positive or negative?

Now if we look at the formula of linear regression, we get y = mx + b. Now there are definitions for each of these values in this specific dataset. - X is the predictor, independent variable, or horsepower - Y is the response, dependent variable, or the MPG - B is the constant value applied to this dataset, which explains what happens to the MPG when horsepower is 0 - M is the slope, which is also known as the coefficient to the dependen variable

The M value in this case controls the trend of the regression line. Having a value of -0.1578, this means that the slope of the trend is a negative value. This also matches the practical reading of the data, which also says that as horsepower increases, MPG will decrease,

iv. What is the predicted mpg associated with a horsepower of 98? What are the

associated 95 % confidence and prediction intervals?

To find the predicted MPG at 98 horsepower you simply plug it into the x-value. Your work would look like:

$39.9359 + (98 \times -0.1578) = 24.47$ And therefore, for a horsepower of 98, yours estimated MPG would be 24.47.

The associated 95% confidece and prediction intervals at 98 would be calculated by running the code:

- ci_predict = [1, 98]
- prediction = regression.get_prediction(ci_predict)
- print(prediction.summary_frame(alpha=0.05))

Which is listed below. Now reading off of this, you can see that The mean matches the predicted values above. The mean_ci_lower and mean_ci_upper are the confidence intervals, and the obs_ci_lower and obs_ci_upper are the prediction intervals. This means that we are 95% confident that the MPG for 98 horsepower are between these 2 values, and that one specific car is between these 2 values in the predicted range.

1.3.1 Confidence and Prediction Intervals

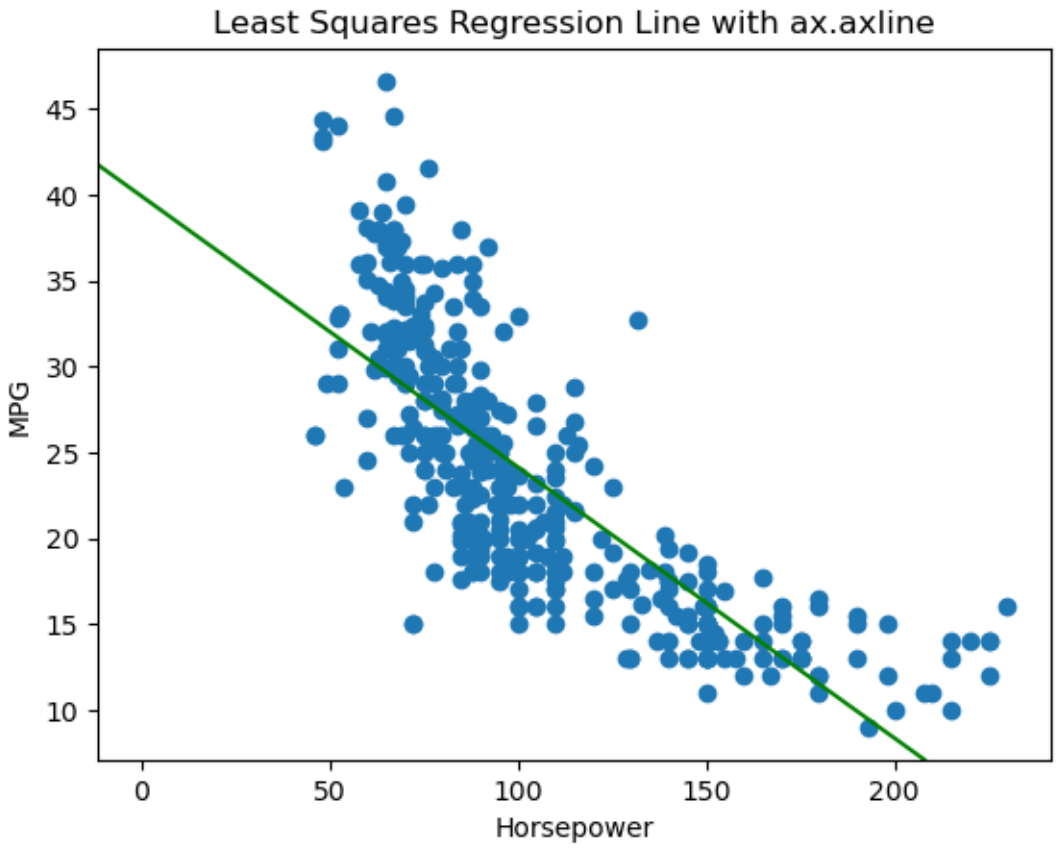
[7]:
```
ci_predict = [1, 98]
prediction = regression.get_prediction(ci_predict)
print(prediction.summary_frame(alpha=0.05).T)
```

```
                     0
mean         24.467077
mean_se       0.251262
mean_ci_lower 23.973079
mean_ci_upper 24.961075
obs_ci_lower  14.809396
obs_ci_upper  34.124758
```

1.3.2 (b)

[8]:
```
fix, ax = plt.subplots()
ax.scatter(df['horsepower'], df['mpg'])
intrcpt = regression.params.iloc[0]
slope1 = regression.params.iloc[1]
ax.axline((0, intrcpt), slope=slope1, color='green')
ax.set_xlabel("Horsepower")
ax.set_ylabel("MPG")
ax.set_title("Least Squares Regression Line with ax.axline")
```

[8]:
```
Text(0.5, 1.0, 'Least Squares Regression Line with ax.axline')
```

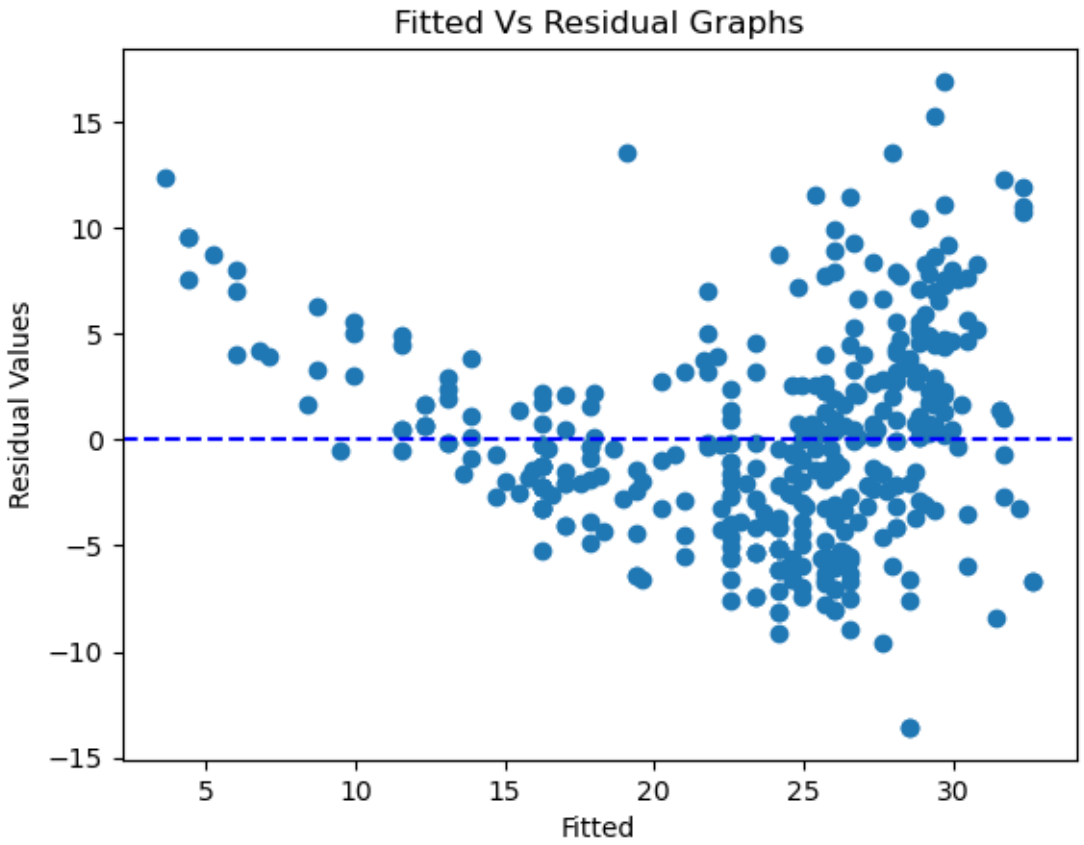


1.3.3 (c)

Fitted vs Residual

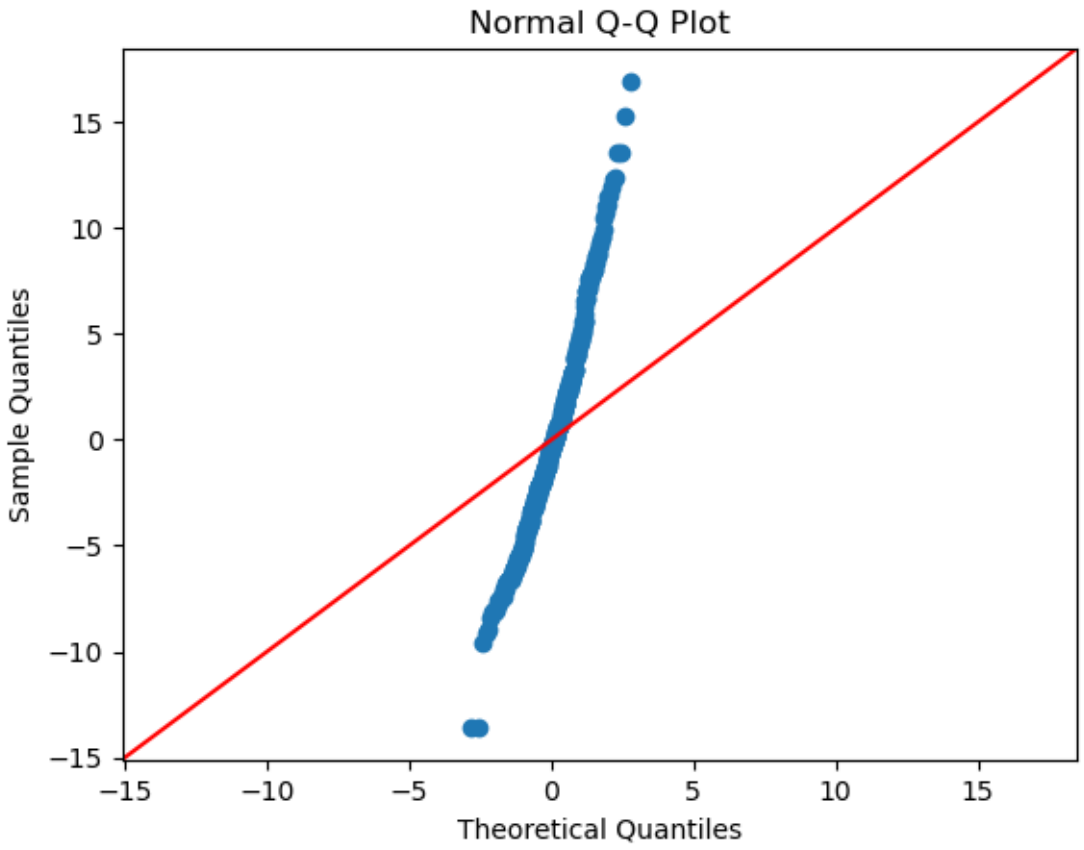
[9]:
```
fittedvalues = regression.fittedvalues
residual = regression.resid

plt.scatter(fittedvalues,residual)
plt.axhline(0, color='blue', linestyle='--')
plt.xlabel('Fitted')
plt.ylabel('Residual Values')
plt.title('Fitted Vs Residual Graphs')
```

[9]:
```
Text(0.5, 1.0, 'Fitted Vs Residual Graphs')
```



Q-Q Plots

[10]:
```
sm.qqplot(residual, line='45')
plt.title("Normal Q-Q Plot")
plt.show()
```



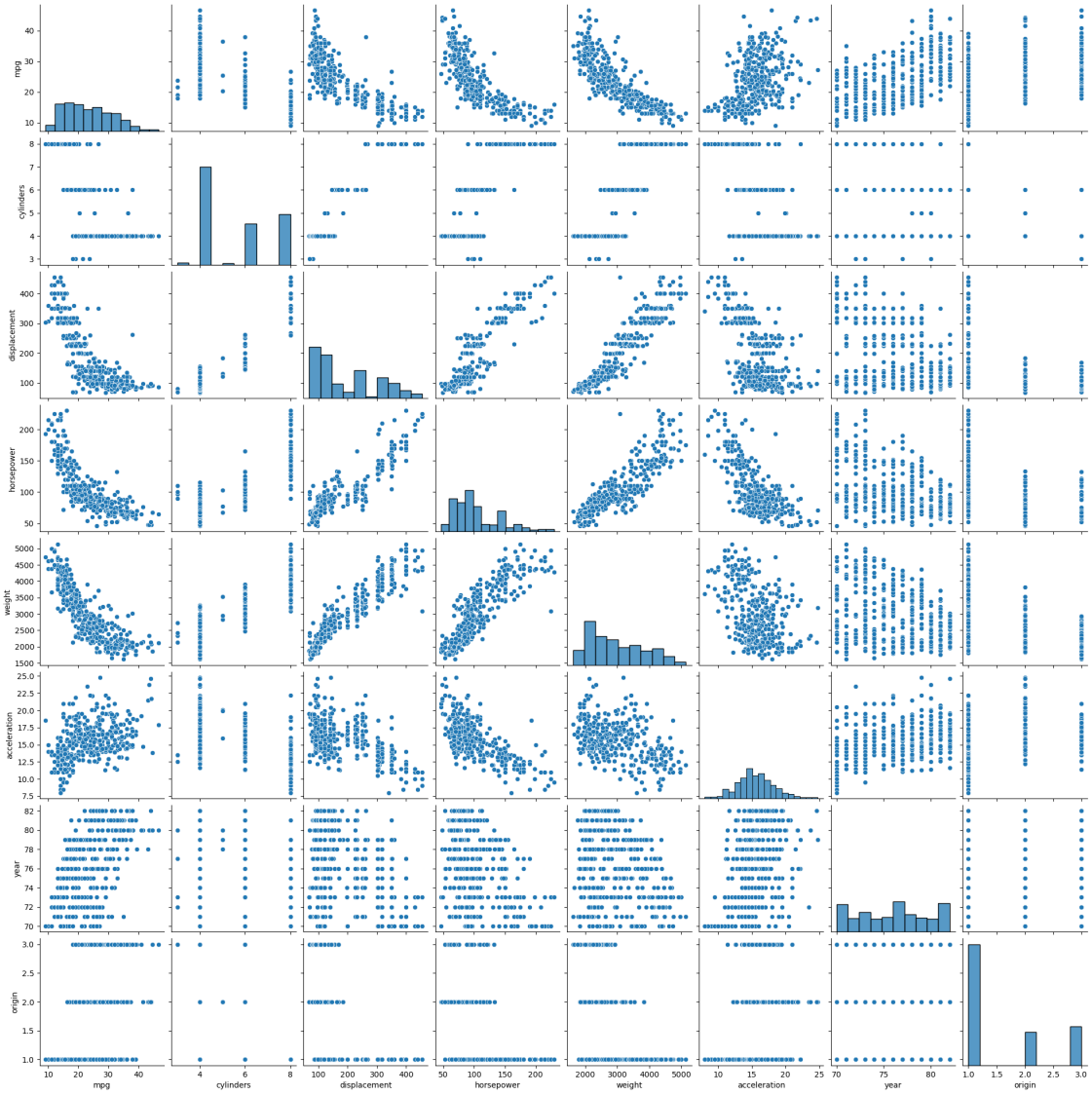
Comments

1. Residuals vs Fitted: We can clearly see a curve on the graph, facing upwards. This shows us that maybe linear regression isn't the best way to model this data, and that we need to use classificationsto do it better, which is the next chapter.
2. Q-Q Plot We can clearly see that the 45 degree line is not being followed, which shows us that this data is not plotted exactly as how it should be, which raises concerns about whenther this type of regression is correct for this dataset.

1.4 Part B:

(a)

[11]:
```
sns.pairplot(df.drop(columns=["name"]))
plt.show()
```



(b)

[12]:
```
corr_matrix = df.drop(columns=["name"]).corr()
print(corr_matrix)
```

```
                    mpg  cylinders  displacement  horsepower    weight  \
mpg            1.000000  -0.777618     -0.805127   -0.778427 -0.832244
cylinders     -0.777618   1.000000      0.950823    0.842983  0.897527
displacement  -0.805127   0.950823      1.000000    0.897257  0.932994
horsepower    -0.778427   0.842983      0.897257    1.000000  0.864538
weight        -0.832244   0.897527      0.932994    0.864538  1.000000
acceleration   0.423329  -0.504683     -0.543800   -0.689196 -0.416839
year           0.580541  -0.345647     -0.369855   -0.416361 -0.309120
origin         0.565209  -0.568932     -0.614535   -0.455171 -0.585005
```

```
              acceleration      year    origin
mpg               0.423329  0.580541  0.565209
cylinders        -0.504683 -0.345647 -0.568932
displacement     -0.543800 -0.369855 -0.614535
horsepower       -0.689196 -0.416361 -0.455171
weight           -0.416839 -0.309120 -0.585005
acceleration      1.000000  0.290316  0.212746
year              0.290316  1.000000  0.181528
origin            0.212746  0.181528  1.000000
```

1.4.1 Multiple Linear Regression

[28]:
```
X = df.drop(columns=["mpg", "name"])
y = df["mpg"]

X = sm.add_constant(X)

model = sm.OLS(y, X).fit()
print(model.summary())

formula = 'mpg ~ cylinders + displacement + horsepower + weight + acceleration␣
 ↪+ year + origin'
model = ols(formula, data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print()
print( '----------Anova Summary----------')
print(anova_table)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.821
Model:                            OLS   Adj. R-squared:                  0.818
Method:                 Least Squares   F-statistic:                     252.4
Date:                Sat, 29 Nov 2025   Prob (F-statistic):          2.04e-139
Time:                        18:01:26   Log-Likelihood:                -1023.5
No. Observations:                 392   AIC:                             2063.
Df Residuals:                     384   BIC:                             2095.
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -17.2184      4.644     -3.707      0.000     -26.350      -8.087
cylinders     -0.4934      0.323     -1.526      0.128      -1.129       0.142
displacement   0.0199      0.008      2.647      0.008       0.005       0.035
horsepower    -0.0170      0.014     -1.230      0.220      -0.044       0.010
weight        -0.0065      0.001     -9.929      0.000      -0.008      -0.005
```

```
acceleration     0.0806      0.099      0.815      0.415      -0.114       0.275
year             0.7508      0.051     14.729      0.000       0.651       0.851
origin           1.4261      0.278      5.127      0.000       0.879       1.973
==============================================================================
Omnibus:                       31.906   Durbin-Watson:                   1.309
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               53.100
Skew:                           0.529   Prob(JB):                     2.95e-12
Kurtosis:                       4.460   Cond. No.                     8.59e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 8.59e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
----------Anova Summary----------
                   sum_sq     df           F        PR(>F)
cylinders       25.791491    1.0    2.329125  1.277965e-01
displacement    77.612668    1.0    7.008884  8.444649e-03
horsepower      16.739754    1.0    1.511699  2.196328e-01
weight        1091.631693    1.0   98.580813  7.874953e-21
acceleration     7.358417    1.0    0.664509  4.154780e-01
year          2402.249906    1.0  216.937408  3.055983e-39
origin         291.134494    1.0   26.291171  4.665681e-07
Residual      4252.212530  384.0         NaN           NaN
```

1.5 Questions 2:

i. Is there a relationship between the predictors and the response? Use the anova_lm() function from statsmodels to answer this question.

An annova table is a table used to see the relationship and significance of the predictors and responses in a data set, being short for Analysis of Variance Table. Our predictor is the MPG(miles per gallon). In this table there are 4 parts of it in the summary:

- Sum of Squares: how much variation in the response each factor explains
- Degrees of Freedom: How many values can change freely before the rest are fixed
- F - Statistic: ratio of explained variance to unexplained variance
- PR/P - Value: probability that the factor has no real effect(<0.5 is better and more significant)

On this specific summary we need to look at the F - Values and the F - Statistic. These 2 ratios are the most significant to seeing the relationship between the predictors and the responses. The F-values are very important, the larger the F-Value, the more we know about the variance in the responses and we can see how it correlates with the predictor. If the predictor can explain more of the data, it is a clear sign of whether it is significant or not. However this is not the only statistic that we have to look at. A more accurate predictor that we use is the P-Value.

With the P-Value, it is the opposite. The P-Value needs to be below 0.05 in order for it to be

deemed significant. It uses the F-Statistic, and using probability it deems whether the category significant by chance or not. The lower the p-value, the less percent of a chance the F-Value is random and happened by chance, such as if the P-Vaue is 0.01, then there is a 1% chance the F-Value is false.

Based on these 2: There are 4 ones that seem to have a significant relationship, and 3 that don't.

Significant Predictors ($p < 0.05$) - displacement ($p = 0.008$) - weight (p 0) - year (p 0) - origin ($p = 4.67e\text{-}07$)

Not Significant Predictors (p 0.05) - cylinders ($p = 0.128$) - horsepower ($p = 0.22$) - acceleration ($p = 0.415$)

ii. Which predictors appear to have a statistically significant relationship to the response?

Based on these 2: There are 4 ones that seem to have a significant relationship, and 3 that don't.

Significant Predictors ($p < 0.05$) - displacement ($p = 0.008$) - weight (p 0) - year (p 0) - origin ($p = 4.67e\text{-}07$)

Not Significant Predictors (p 0.05) - cylinders ($p = 0.128$) - horsepower ($p = 0.22$) - acceleration ($p = 0.415$)

iii. What does the coefficient for the year variable suggest? The Coefficient on the year variable tells use that the yea is always increasing, which makes sense because time cannot go backwards, and that it is moving at a linear scale, the coeficient of the year being around 0.7508. On the Anova chart, the t-statistic (14.729) is very high,therefore indicating a strong effect on MPG as well. Using common knowledge we can also predict that the newer a car, the more MPG that it will have, and it having a positive trend refelcts this quite well.