Google Colab Notebook link:
https://drive.google.com/file/d/1D9enKTYf0P3i-oSwTG-OGb-90T3tCV42/view?usp=sharing

Homework 1 (Total 100 points)

Homework 1

Q1. Load and examine the `Auto.csv` dataset from the Blackboard course site. **(20 points total)**

1. Should you drop any variable from regression analysis and why? (5 points)
2. Which variables should be treated as numeric and which as categorical? Explain why. (5 points)

FYI column definitions (from https://cran.r-project.org/web/packages/ISLR/ISLR.pdf):

- mpg: miles per gallon (The outcome, or y, variable)
- cylinders: Number of cylinders between 4 and 8
- displacement: Engine displacement (cu. inches)
- horsepower: Engine horsepower
- weight: Vehicle weight (lbs.)
- acceleration: Time to accelerate from 0 to 60 mph (sec.)
- year: Model year (modulo 100)
- origin: Origin of car (1. American, 2. European, 3. Japanese)
- name: Vehicle name

In [31]:

```
from google.colab import drive
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

drive.mount('/content/drive')
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/BA810/Data/Auto.
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount,
call drive.mount("/content/drive", force_remount=True).
```

In [32]:

```
df.head()
```

Out[32]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	nam
0	18.0	8	307.0	130	3504	12.0	70	1	chevrol chevel malib
1	15.0	8	350.0	165	3693	11.5	70	1	buic skyla 32
2	18.0	8	318.0	150	3436	11.0	70	1	plymou satelli
3	16.0	8	304.0	150	3433	12.0	70	1	am rebel s
4	17.0	8	302.0	140	3449	10.5	70	1	for torir

In [33]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 397 entries, 0 to 396
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           397 non-null    float64
 1   cylinders     397 non-null    int64  
 2   displacement  397 non-null    float64
 3   horsepower    397 non-null    object 
 4   weight        397 non-null    int64  
 5   acceleration  397 non-null    float64
 6   year          397 non-null    int64  
 7   origin        397 non-null    int64  
 8   name          397 non-null    object 
dtypes: float64(3), int64(4), object(2)
memory usage: 28.0+ KB
```

In [34]:

```
# Noticed that horsepower is object and wanted to change it to float

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

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 397 entries, 0 to 396
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           397 non-null    float64
 1   cylinders     397 non-null    int64  
 2   displacement  397 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        397 non-null    int64  
 5   acceleration  397 non-null    float64
 6   year          397 non-null    int64  
 7   origin        397 non-null    int64  
 8   name          397 non-null    object 
dtypes: float64(4), int64(4), object(1)
memory usage: 28.0+ KB
```

In [35]:
```
# Checking name column unique values
len(df['name'].unique())
```

Out[35]:
```
304
```

In [36]:
```
# Since it has many unique values which will not add to the regression mo
df.drop(columns=['name'], inplace=True)
```

In [37]:
```
# To ensure that name column was actually dropped
df.head()
```

Out[37]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin
0	18.0	8	307.0	130.0	3504	12.0	70	1
1	15.0	8	350.0	165.0	3693	11.5	70	1
2	18.0	8	318.0	150.0	3436	11.0	70	1
3	16.0	8	304.0	150.0	3433	12.0	70	1
4	17.0	8	302.0	140.0	3449	10.5	70	1

A1.

1. I've dropped name column from regression because it consists of many unique values that will not contribute to the regression model
2.
`Numeric Variables :`
it's continous changing variables which differ from one to another, such as:

- mpg
- displacement
- horsepower
- weight
- acceleration

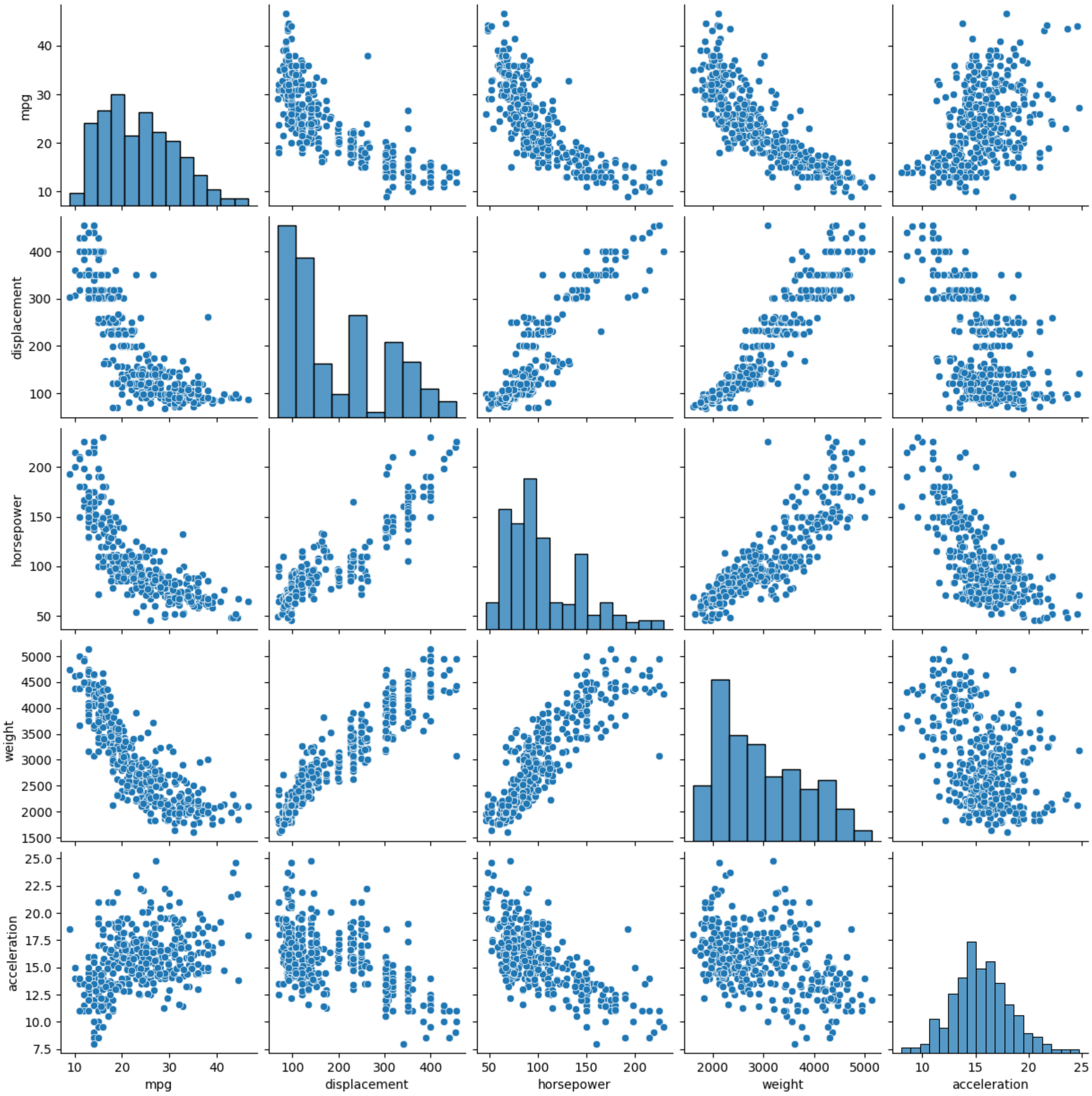
`Categorical Variables :`
the value it represent falls into certain categories even though some of them contain numeric values it's still within fixed categories, such as:

- cylinders
- year
- origin

Q2. Scatter and explore. **(20 points total)**

1. Plot all the pairwise scatter plots and histograms for the numeric features. (10 points)
2. Discuss two interesting relationships that you notice. (10 points)

In [38]:
```
numeric_variables = ['mpg', 'displacement', 'horsepower', 'weight', 'acce

sns.pairplot(df[numeric_variables])
plt.show()
```



A2.

- **mpg vs. weight:**
  There's a clear negative correlation between these two variables. As the weight of the vehicle increases, the mpg (fuel efficiency) tends to decrease. This relationship is consistent with the expectation that heavier vehicles generally consume more fuel than lighter ones.

- **horsepower vs. mpg:**
  There's a negative correlation, cars with higher horsepower generally have lower fuel efficiency.

These relationships highlight important considerations in vehicle design and marketing. Fuel efficiency (mpg) is a crucial factor for many consumers, and these plots underscore the trade-offs between power (as indicated by weight and horsepower) and fuel economy.

Q3. Compute the correlation matrix among the numeric variables. Discuss one interesting correlation. (5+5=**10 points total)**

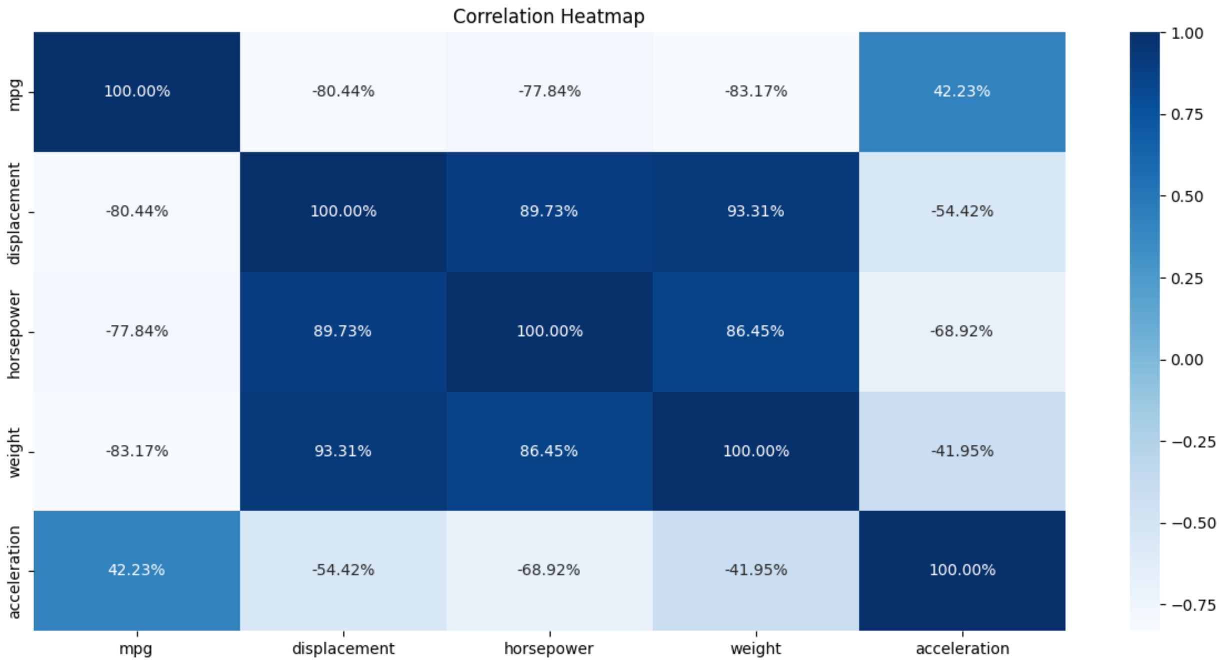
In [39]:

```
# To create the correlation matrix for numeric variables

corr_matrix = df[numeric_variables].corr(numeric_only=True)
print(corr_matrix)
```

```
                   mpg  displacement  horsepower    weight  acceleration
mpg           1.000000     -0.804443   -0.778427 -0.831739      0.422297
displacement -0.804443      1.000000    0.897257  0.933104     -0.544162
horsepower   -0.778427      0.897257    1.000000  0.864538     -0.689196
weight       -0.831739      0.933104    0.864538  1.000000     -0.419502
acceleration  0.422297     -0.544162   -0.689196 -0.419502      1.000000
```

In [40]:

```
# To visualize it in heatmap for better reading

plt.figure(figsize=(15, 7))
sns.heatmap(df[numeric_variables].corr(numeric_only=True), annot=True, cm
plt.title('Correlation Heatmap')
plt.show()
```



A3.

As shown in the correlation matrix and heatmap, the correlation between mpg and weight is -83.17%, indicating a strong negative relationship. This means as the weight of the car increases, its mpg tends to decrease. This is one of the strongest negative correlations in the matrix, highlighting the significant trade-off between the weight of a vehicle and its fuel efficiency.

Q4. Use `statsmodels` to regress mpg on all other variables. Note you can tell `ols()` to treat a variable as categorical by enclosing the variable in `C()`. (10 points) **(15 points total)**

1. Interpret the significant effects. (5 points)
2. Which variables don't have a significant effect? Provide potential explanation for one surprising non-effect. (5 points)
3. Discuss the difference in results when you treat `year` as a categorical vs a numeric variable. (5 points)

In [41]:
```
# Treating year as categorical

mpg_model_c = smf.ols('mpg ~ displacement + horsepower + weight + acceler

print(mpg_model_c.summary())
```

```
                            OLS Regression Results
==============================================================================
=====
Dep. Variable:                    mpg   R-squared:
0.874
```

```
Model:                            OLS   Adj. R-squared:                  
0.867
Method:                 Least Squares   F-statistic:                     
116.8
Date:                Thu, 02 Nov 2023   Prob (F-statistic):          2.64
e-151
Time:                        01:22:28   Log-Likelihood:                -9
54.59
No. Observations:                 392   AIC:                             
1955.
Df Residuals:                     369   BIC:                             
2047.
Df Model:                          22                                         
Covariance Type:            nonrobust                                         
==================================================================
============
                        coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------
------------
Intercept            30.9168      2.361     13.095      0.000      26.274
35.559
C(cylinders)[T.4]     6.9399      1.537      4.516      0.000       3.918
9.962
C(cylinders)[T.5]     6.6377      2.337      2.840      0.005       2.042
11.234
C(cylinders)[T.6]     4.2973      1.706      2.519      0.012       0.943
7.652
C(cylinders)[T.8]     6.3668      1.969      3.234      0.001       2.495
10.238
C(year)[T.71]         0.9104      0.816      1.116      0.265      -0.693
2.514
C(year)[T.72]        -0.4903      0.804     -0.610      0.542      -2.071
1.090
C(year)[T.73]        -0.5529      0.721     -0.766      0.444      -1.972
0.866
C(year)[T.74]         1.2420      0.855      1.453      0.147      -0.439
2.923
C(year)[T.75]         0.8704      0.837      1.039      0.299      -0.776
2.517
C(year)[T.76]         1.4967      0.802      1.866      0.063      -0.080
3.074
C(year)[T.77]         2.9987      0.820      3.657      0.000       1.386
4.611
C(year)[T.78]         2.9738      0.779      3.816      0.000       1.442
4.506
C(year)[T.79]         4.8962      0.825      5.936      0.000       3.274
6.518
C(year)[T.80]         9.0589      0.875     10.351      0.000       7.338
10.780
C(year)[T.81]         6.4582      0.864      7.477      0.000       4.760
8.157
C(year)[T.82]         7.8376      0.849      9.228      0.000       6.167
9.508
C(origin)[T.2]        1.6933      0.516      3.280      0.001       0.678
2.708
C(origin)[T.3]        2.2929      0.497      4.616      0.000       1.316
3.270
displacement          0.0118      0.007      1.745      0.082      -0.001
0.025
horsepower           -0.0392      0.013     -3.010      0.003      -0.065
```

```
-0.014
weight              -0.0052      0.001     -8.300      0.000      -0.006
-0.004
acceleration         0.0036      0.087      0.042      0.967      -0.167
0.174
==========================================================================
=====
Omnibus:                       32.560   Durbin-Watson:
1.574
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                5
5.829
Skew:                           0.528   Prob(JB):                      7.5
3e-13
Kurtosis:                       4.518   Cond. No.                      7.9
5e+04
==========================================================================
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 7.95e+04. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

In [42]:

```python
# Treating year as numeric

mpg_model_n = smf.ols('mpg ~ displacement + horsepower + weight + acceler

print(mpg_model_n.summary())
```

```
                            OLS Regression Results
==========================================================================
=====
Dep. Variable:                    mpg   R-squared:
0.847
Model:                            OLS   Adj. R-squared:
0.842
Method:                 Least Squares   F-statistic:
191.1
Date:                Thu, 02 Nov 2023   Prob (F-statistic):           2.39
e-147
Time:                        01:22:28   Log-Likelihood:                 -9
93.35
No. Observations:                 392   AIC:
2011.
Df Residuals:                     380   BIC:
2058.
Df Model:                          11
Covariance Type:            nonrobust
==========================================================================
============
                        coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------
------------
Intercept           -22.0801      4.541     -4.862      0.000     -31.009
-13.151
C(cylinders)[T.4]     6.7218      1.654      4.064      0.000       3.470
9.974
C(cylinders)[T.5]     7.0784      2.516      2.813      0.005       2.131
```

```
12.026
C(cylinders)[T.6]     3.3512      1.824      1.837      0.067      -0.236
6.938
C(cylinders)[T.8]     5.0992      2.109      2.418      0.016       0.953
9.246
C(origin)[T.2]        1.7640      0.551      3.200      0.001       0.680
2.848
C(origin)[T.3]        2.6172      0.527      4.964      0.000       1.581
3.654
displacement          0.0187      0.007      2.590      0.010       0.005
0.033
horsepower           -0.0349      0.013     -2.639      0.009      -0.061
-0.009
weight               -0.0058      0.001     -9.154      0.000      -0.007
-0.005
acceleration          0.0260      0.093      0.279      0.780      -0.157
0.209
year                  0.7370      0.049     15.064      0.000       0.641
0.833
==========================================================================
=====
Omnibus:                       45.781   Durbin-Watson:
1.336
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                 8
5.634
Skew:                           0.677   Prob(JB):                       2.5
4e-19
Kurtosis:                       4.846   Cond. No.                       9.3
2e+04
==========================================================================
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 9.32e+04. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

A4.

1. **Significant Effects:**
   If the p-value is less than 0.05, the variable is considered to have a significant effect.

- Cars with 4, 5, 6, and 8 cylinders have higher mpg.
- Cars from years 77 to 82 have significantly higher mpg.
- Cars of origin 2 and 3 have higher mpg than origin 1.
- Heavier cars have lower mpg.

1. **Non-significant Variables:**

- Displacement doesn't significantly predict mpg.
- Many horsepower levels don't significantly impact mpg.
- Acceleration does not show a significant effect on mpg in this model.

1. **What will happen if we treated Year as Categorical vs Numeric?**

- Treating year as categorical captures unique year effects.
- As numeric, it would assume a linear mpg change over years.
- The categorical treatment captures potential non-linear patterns but adds more parameters.

Q5. From the above regression model in Q4, include two way interactions between a numeric and categorical variable in three different regression models (three separate models in total). Do any of them appear significant? Discuss the results. **(15 points total)**

In [43]:

```
# Model 1: Interaction between weight (numeric) and origin (categorical):
model1 = smf.ols('mpg ~ weight * C(origin)', data=df).fit()
print(model1.summary())
```

```
                            OLS Regression Results
==========================================================================
=====
Dep. Variable:                    mpg   R-squared:
0.706
Model:                            OLS   Adj. R-squared:
0.703
Method:                 Least Squares   F-statistic:
188.1
Date:                Thu, 02 Nov 2023   Prob (F-statistic):          1.14
e-101
Time:                        01:22:28   Log-Likelihood:                -1
136.4
No. Observations:                 397   AIC:
2285.
Df Residuals:                     391   BIC:
2309.
Df Model:                           5
Covariance Type:            nonrobust
==========================================================================
================
                            coef    std err          t      P>|t|      [
0.025      0.975]
--------------------------------------------------------------------------
----------------
Intercept                42.9846      1.179     36.465      0.000      4
0.667      45.302
C(origin)[T.2]            2.3912      2.847      0.840      0.401      -
3.206       7.988
C(origin)[T.3]           11.2755      3.583      3.147      0.002
4.231      18.320
weight                   -0.0068      0.000    -19.973      0.000      -
0.007      -0.006
weight:C(origin)[T.2]    -0.0004      0.001     -0.365      0.715      -
0.003       0.002
weight:C(origin)[T.3]    -0.0039      0.002     -2.527      0.012      -
0.007      -0.001
==========================================================================
=====
Omnibus:                       42.084   Durbin-Watson:
0.819
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               6
1.346
Skew:                           0.720   Prob(JB):                     4.7
8e-14
Kurtosis:                       4.278   Cond. No.                     5.3
6e+04
==========================================================================
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 5.36e+04. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

Observation:

- `R-squared Value:`
  0.706 or 70.6% of the variance in mpg.
- `Significant Variables:`
  weight & origin (3) & their interactions with each other were significant predictors.

In [44]:
```
# Model 2: Interaction between acceleration (numeric) and cylinders (cate
model2 = smf.ols('mpg ~ acceleration * C(cylinders)', data=df).fit()
print(model2.summary())
```

```
                            OLS Regression Results
==========================================================================
=====
Dep. Variable:                    mpg   R-squared:
0.653
Model:                            OLS   Adj. R-squared:
0.645
Method:                 Least Squares   F-statistic:
81.07
Date:                Thu, 02 Nov 2023   Prob (F-statistic):           1.8
3e-83
Time:                        01:22:28   Log-Likelihood:                -1
169.3
No. Observations:                 397   AIC:
2359.
Df Residuals:                     387   BIC:
2398.
Df Model:                           9
Covariance Type:            nonrobust
==========================================================================
=========================
                                    coef    std err          t      P>|t
|      [0.025      0.975]
--------------------------------------------------------------------------
-------------------------
Intercept                        76.2000     71.340      1.068      0.28
6     -64.063     216.463
C(cylinders)[T.4]               -50.2580     71.378     -0.704      0.48
2    -190.595      90.078
C(cylinders)[T.5]               -95.0958     75.950     -1.252      0.21
1    -244.421      54.230
C(cylinders)[T.6]               -45.8103     71.461     -0.641      0.52
2    -186.310      94.689
C(cylinders)[T.8]               -66.5424     71.392     -0.932      0.35
2    -206.907      73.823
acceleration                     -4.2000      5.381     -0.780      0.43
6     -14.780       6.380
acceleration:C(cylinders)[T.4]    4.4036      5.383      0.818      0.41
4      -6.180      14.987
acceleration:C(cylinders)[T.5]    6.6828      5.558      1.202      0.23
0      -4.245      17.611
acceleration:C(cylinders)[T.6]    3.5603      5.387      0.661      0.50
9      -7.032      14.152
acceleration:C(cylinders)[T.8]    4.6095      5.385      0.856      0.39
3      -5.979      15.198
==========================================================================
```

```
=====
Omnibus:                       40.959   Durbin-Watson:                   
1.024
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                6
2.131
Skew:                           0.687   Prob(JB):                      3.2
2e-14
Kurtosis:                       4.368   Cond. No.                      1.3
1e+04
==========================================================================
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 1.31e+04. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

Observation:

- `R-squared Value:`

  0.653 or 65.3% of the variance in mpg.
- `Significant Variables:`

  No segnificat variables in this model.
- `Concerns:`

  Adjusted R-squared is lower than R-squared, hinting at potential overfitting.

In [45]:
```
# Model 3: Interaction between displacement (numeric) and year (categoric
model3 = smf.ols('mpg ~ displacement * C(year)', data=df).fit()
print(model3.summary())
```

```
                            OLS Regression Results                        
==========================================================================
=====
Dep. Variable:                    mpg   R-squared:                       
0.818
Model:                            OLS   Adj. R-squared:                  
0.805
Method:                 Least Squares   F-statistic:                     
66.56
Date:                Thu, 02 Nov 2023   Prob (F-statistic):          9.82
e-121
Time:                        01:22:28   Log-Likelihood:                -1
041.8
No. Observations:                 397   AIC:                             
2136.
Df Residuals:                     371   BIC:                             
2239.
Df Model:                          25                                     
Covariance Type:            nonrobust                                     
==========================================================================
=====================
                              coef    std err          t      P>|t|      
[0.025      0.975]
--------------------------------------------------------------------------
---------------------
Intercept                  27.8717      1.609     17.325      0.000      
```

```
24.708      31.035
C(year)[T.71]                   4.6350      2.116      2.190      0.029
0.473       8.797
C(year)[T.72]                  -0.5125      2.095     -0.245      0.807
-4.631       3.606
C(year)[T.73]                  -1.9690      2.061     -0.955      0.340
-6.021       2.083
C(year)[T.74]                   6.2381      2.146      2.907      0.004
2.018      10.458
C(year)[T.75]                   2.3236      2.290      1.015      0.311
-2.179       6.827
C(year)[T.76]                   4.4264      2.127      2.081      0.038
0.244       8.609
C(year)[T.77]                   5.8099      2.099      2.768      0.006
1.683       9.937
C(year)[T.78]                   8.1828      2.187      3.742      0.000
3.883      12.483
C(year)[T.79]                  10.1203      2.227      4.544      0.000
5.741      14.499
C(year)[T.80]                  21.7052      2.831      7.667      0.000
16.139      27.272
C(year)[T.81]                  12.1169      2.299      5.271      0.000
7.596      16.637
C(year)[T.82]                  10.9740      2.691      4.077      0.000
5.682      16.266
displacement                   -0.0362      0.005     -6.900      0.000
-0.046      -0.026
displacement:C(year)[T.71]    -0.0175      0.008     -2.242      0.026
-0.033      -0.002
displacement:C(year)[T.72]    -0.0034      0.008     -0.454      0.650
-0.018       0.011
displacement:C(year)[T.73]     0.0019      0.007      0.276      0.783
-0.012       0.016
displacement:C(year)[T.74]    -0.0302      0.009     -3.360      0.001
-0.048      -0.013
displacement:C(year)[T.75]    -0.0121      0.009     -1.348      0.179
-0.030       0.006
displacement:C(year)[T.76]    -0.0180      0.008     -2.188      0.029
-0.034      -0.002
displacement:C(year)[T.77]    -0.0177      0.008     -2.184      0.030
-0.034      -0.002
displacement:C(year)[T.78]    -0.0313      0.009     -3.364      0.001
-0.050      -0.013
displacement:C(year)[T.79]    -0.0262      0.009     -3.061      0.002
-0.043      -0.009
displacement:C(year)[T.80]    -0.1009      0.020     -5.038      0.000
-0.140      -0.062
displacement:C(year)[T.81]    -0.0352      0.012     -2.849      0.005
-0.059      -0.011
displacement:C(year)[T.82]    -0.0172      0.017     -1.018      0.309
-0.051       0.016
==========================================================================
=====
Omnibus:                        27.551   Durbin-Watson:
1.630
Prob(Omnibus):                   0.000   Jarque-Bera (JB):                 9
9.287
Skew:                            0.086   Prob(JB):                       2.7
5e-22
Kurtosis:                        5.444   Cond. No.                       7.9
2e+03
```

```
==========================================================================
=====

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 7.92e+03. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

Observation:

- `R-squared Value:`
  0.818 or 81.8% of the variance in mpg. Which is the highest among the three
- `Significant Variables:`
  displacement & the following years (71, 74, 76-82) & their interactions with each other were significant predictors.
- `Concerns:`
  This model has the highest R-squared value, but has many coefficients, which might make it prone to overfitting. Also, some coefficients are not statistically significant.

Discussing the Results of 3 models:

The third model is the best at predicting mpg. It uses important information from the first two models and looks deeper into how year and displacement work together. However, there might be some issues with the data being too closely related, which could be a problem.
Even though it's the most detailed model, we need to be careful when understanding what it tells us.

Q6. Measure the in-sample and out of sample R^2 of the model specified in Q4.1 using 80% data for training and 20% data for testing. **(10 points total)**

In [46]:

```python
train, test = train_test_split(df, test_size = 0.20, random_state=23)

est = smf.ols('mpg ~ displacement + horsepower + weight + acceleration +
print('In sample R-square: {:.4f}'.format(est.rsquared))

predictions = est.predict(test)
print('Out of sample R-square: {:.4f}'.format(r2_score(test.mpg, predicti
```

```
In sample R-square: 0.8744
Out of sample R-square: 0.8651
```

Q7. Collaboration statement **(10 points total)**

Who did you discuss while answering this homework (whether to get or to provide help)? What questions/topics did you discuss? Did you use any generative AI tool, such as ChatGPT? If so, provide your prompts.

Note: No penalty for either side. While getting help in figuring out how to solve is OK, all answers should be produced by you.

If you did not collaborate with anyone simply declare so.

Sample answer:

1. I discussed with Hazel for this homework. I needed her help with submission system and as a sounding board for the reasonableness of this homework.
2. I used ChatGPT with the following prompt to understand how to measure out of sample R^2:

   > provide prompt here Then I wrote the code that is submitted in the assignment.

3. I did not discuss with anyone or get any help from any generative AI tool.

- I had issue with numerical variable being treated as categorical "horsepower". I've discussed this issue with Fahad and he advised me to check its type and I've found out that it was object and that's why I faced these error. Then I went and changed the dtype to float, and it's working now.
- I've used ChatGPT to explain to me how to interpret the $R2$ outcome.
- I've used the code shared in Slack by Boyuan Chen to be able to expert a clean PDF file of the notebook.

In [ ]:

```
# Code to export notebook to pdf

! jupyter nbconvert --to html /content/Sulaiman_Alhomoud_HW1.ipynb
```