Project-1

SPRING 2024-25

Ezgi Sena Karabacak
Date: March 6, 2025

Question 1

In this question, we analyzed the Auto dataset and fit a linear regression model to it.

a) Linear Regression Model

The summary of the dataset can be seen below:

Column Name	Data Type
mpg	float64
cylinders	int64
displacement	float64
horsepower	object
weight	int64
acceleration	float64
year	int64
origin	int64
name	object

Table 1: Data Types of the Auto Dataset

We need to convert horsepower into numeric type for linear regression.
The summary of the model can be seen below:

	Coefficient	Std. Error	t-value	P-value
const	39.9359	0.717	55.660	0.000
horsepower	-0.1578	0.006	-24.489	0.000

Table 2: OLS Regression Results

Answers

1. There is a statistically significant relationship between *horsepower* and *mpg*. The p-value for the *horsepower* coefficient is **0.000**, which is very small, indicating a strong relationship. Furthermore, the F-statistic of **599.7** with a near-zero p-value confirms the overall significance of the model.

Statistic	**Value**
R-squared	0.606
Adj. R-squared	0.605
F-statistic	599.7
Prob (F-statistic)	7.03×10^{-81}
Log-Likelihood	-1178.7
AIC	2361
BIC	2369
Durbin-Watson	0.920
Omnibus	16.432
Prob (Omnibus)	0.000
Jarque-Bera (JB)	17.305
Prob (JB)	0.000175
Skew	0.492
Kurtosis	3.299
Cond. No.	322

Table 3: Additional Regression Statistics

2. The strength of the relationship is measured using the R-squared (R^2) value:

   $$R^2 = 0.606$$

   This means that **60.6% of the variance** in *mpg* is explained by *horsepower*, suggesting a **moderately strong** linear relationship.

3. The relationship is **negative** because the *horsepower* coefficient is:

   $$-0.1578$$

   This means that as **horsepower increases, mpg decreases**. This is expected as higher-horsepower cars are typically less fuel efficient.

4. Using the regression equation:

   $$\text{mpg} = 39.9359 - 0.1578 \times \text{horsepower}$$

   For *horsepower* = 98:

   $$\text{mpg} = 24.467$$

   The **95% Confidence Interval** (for the mean mpg) is:

   $$[23.973, 24.961]$$

   The **95% prediction interval** (for individual mpg values) is:

   $$[14.809, 34.125]$$

   This means:

   - The **true mean** *mpg* for cars with 98 horsepower is likely between **23.973 and 24.961**.
   - For an **individual car**, the *mpg* could vary between **14.809 and 34.125** due to natural variation.

b) Regression Plot

We can see the regression line aligns with the dataset.

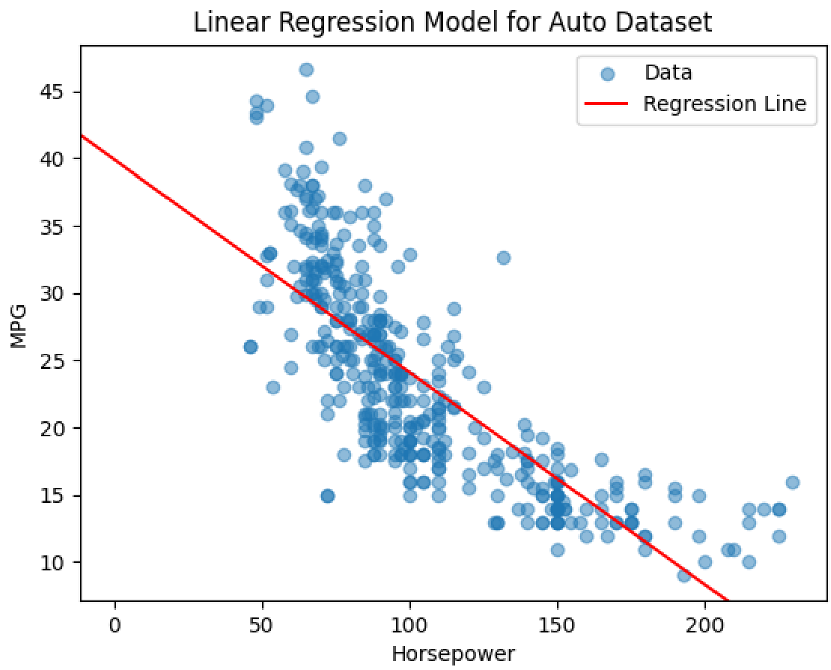

Figure 1: Scatter plot of Horsepower vs. MPG with regression line

c) Diagnostic Plots

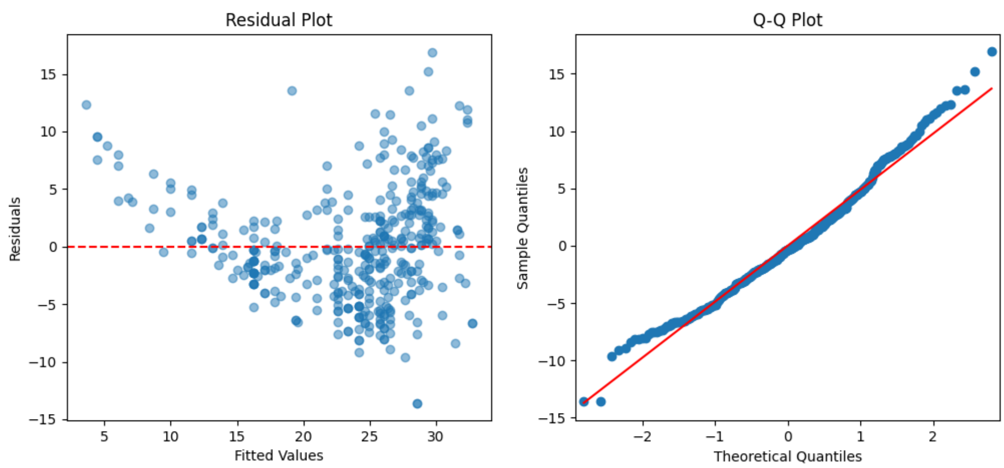

Figure 2: Residual plot (left) and Q-Q plot (right) for diagnostic analysis

In Figure 2, we can see the diagnostic plots for the model. From the residual plot, we see that the variance of residuals are not constant implying **heteroscedasticity**. Ideally, residuals should be randomly scattered around zero, but here, they **fan out** as fitted values increase. This suggests a **non-linear relationship** and a violation of the

homoscedasticity assumption. A polynomial regression model may provide a better fit. We can also confirm this from Figure 1 too.

Also, from them the Q-Q plot, we can say that most points align with the red reference line, but deviations at both ends indicate **non-normality**. The Q-Q plot checks if theiduals follow a normal distribution. Outliers may be affecting the model, leading to issues in inference tests such as confidence intervals and p-values.

Question 2

In this question, we analyzed the Auto dataset and fit a multiple linear regression model to it.

a) Scatter Plot Matrix

Scatter plot matrix for the Auto dataset can be seen in Figure 3. We removed the name column as it is not a numeric variable. We can see that the diagonal elements contain histograms of individual variables whereas the off-diagonal elements contain scatter plots comparing each pair of variables.

- **Negative Correlations:**
  - *mpg* vs *horsepower*
  - *mpg* vs *weight*
  - *mpg* vs *displacement*
- **Positive Correlations:**
  - *displacement* vs *horsepower*
  - *displacement* vs *weight*
- **Categorical Variables:**
  - *cylinders*, *origin*, and *year* appear as discrete dots.

b) Matrix of Correlations

We can see the correlation relationships between variables in Figure 4. Mpg and cylinders, displacement and horsepower have the highest correlations.

c) Multiple Regression Model Summary

Model summary can be seen in Table 4.

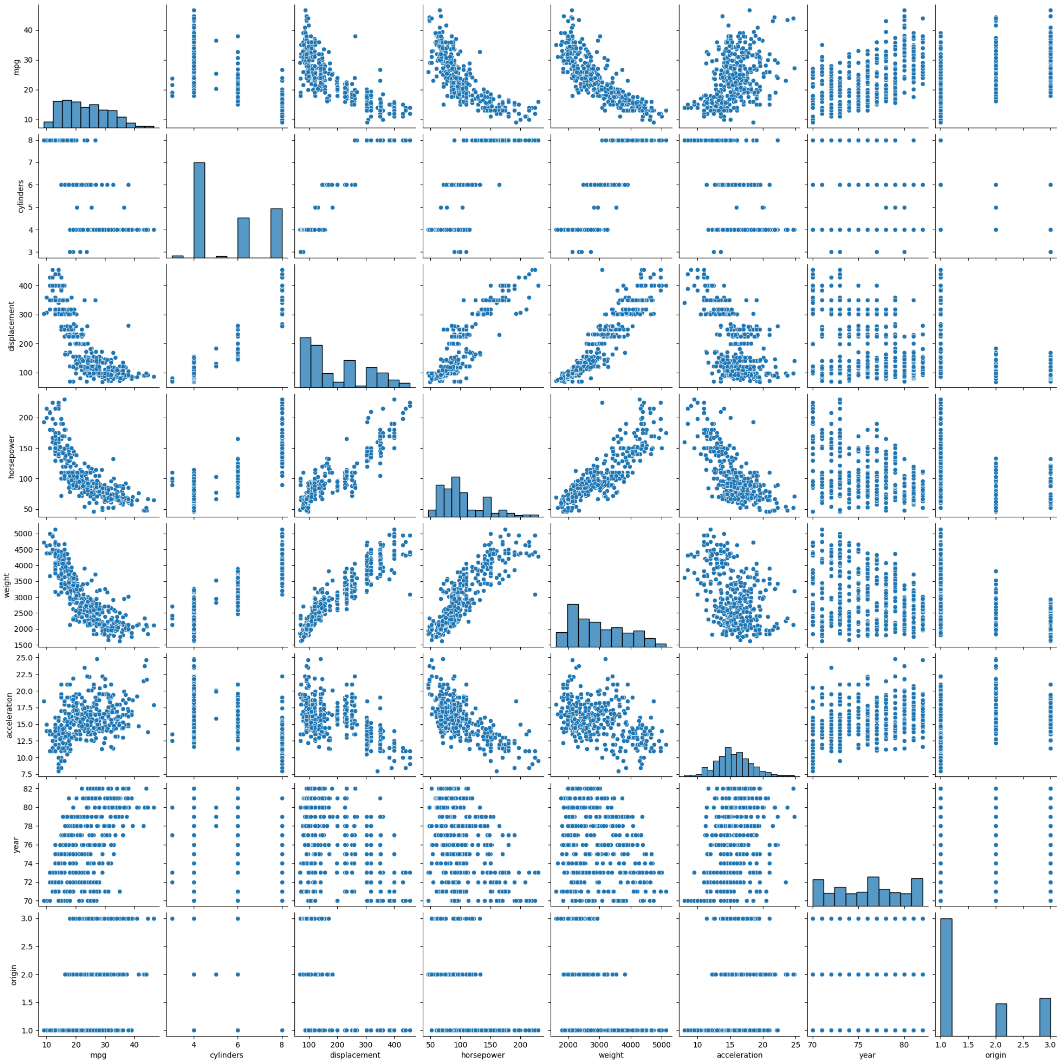

Figure 3: Scatterplot Matrix of the Auto Dataset

Answers

1. To determine if there is an overall relationship between the predictors and the response variable, we use the **ANOVA test**. The F-statistic value of **252.4** with a p-value of **2.04e-139** indicates that at least one predictor is significantly related to **mpg**. Since the p-value is extremely small (much less than 0.05), we **reject the null hypothesis**, confirming a statistically significant relationship between the predictors and **mpg**.

2. To identify which predictors have a statistically significant effect on **mpg**, we examine the **p-values** from both the regression and ANOVA results:

   - **Significant Predictors ($p < 0.05$ in both regression and ANOVA):**
     - **displacement** (ANOVA p = 1.53e-20, Regression p = 0.008)
     - **weight** (ANOVA p = 5.54e-19, Regression p = 0.000)
     - **year** (ANOVA p = 1.87e-39, Regression p = 0.000)
     - **origin** (ANOVA p = 4.66e-07, Regression p = 0.000)

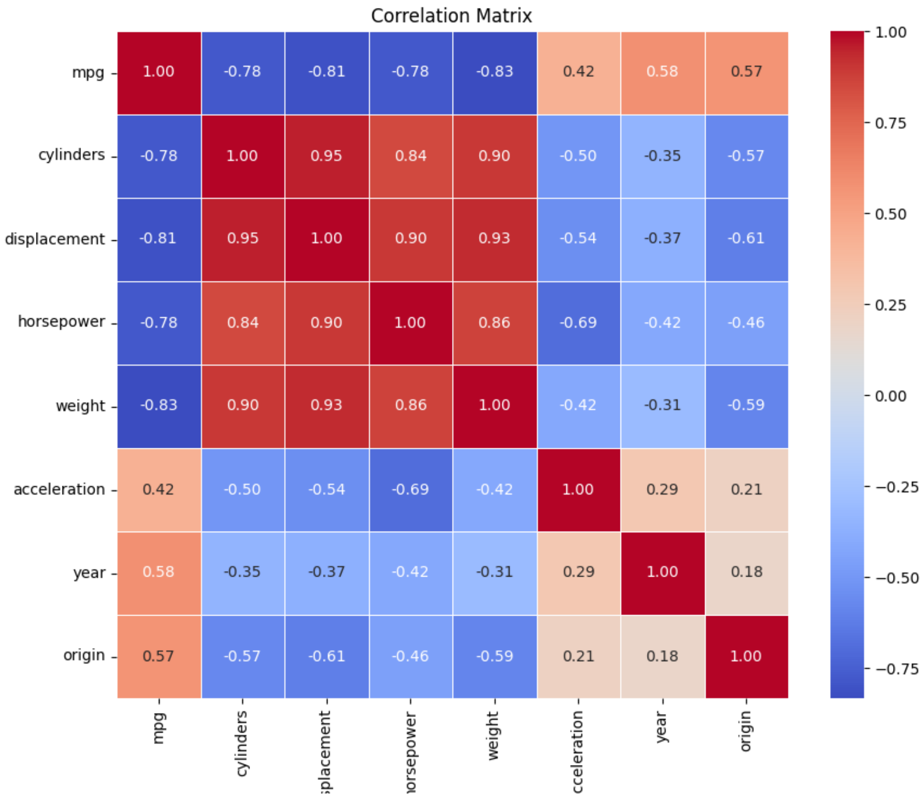

Figure 4: Correlation Matrix of the Auto Dataset

- **Non-Significant Predictors (p > 0.05 in ANOVA and Regression):**
  - **cylinders** (ANOVA p = 2.32e-125, Regression p = 0.128, but likely multicollinear)
  - **horsepower** (ANOVA p = 3.73e-09, Regression p = 0.220)
  - **acceleration** (ANOVA p = 0.768, Regression p = 0.415)

The **ANOVA results further confirm that displacement, weight, year, and origin have a significant impact on mpg**, while acceleration is clearly not significant. Cylinders have a very small ANOVA p-value but are not significant in regression, likely due to multicollinearity.

3. The coefficient for **year** is **0.7508**, which means that for each additional year, **mpg** increases by approximately **0.75 miles per gallon**, holding all other variables constant. The ANOVA test further reinforces the significance of this variable (**p = 1.87e-39**). This suggests that as vehicle model years increase, fuel efficiency tends to improve, possibly due to advancements in engine technology and efficiency regulations.

	Coefficient	Std. Error	t-value	P-value
const	-17.2184	4.644	-3.707	0.000
cylinders	-0.4934	0.323	-1.526	0.128
displacement	0.0199	0.008	2.647	0.008
horsepower	-0.0170	0.014	-1.230	0.220
weight	-0.0065	0.001	-9.929	0.000
acceleration	0.0806	0.099	0.815	0.415
year	0.7508	0.051	14.729	0.000
origin	1.4261	0.278	5.127	0.000

Table 4: OLS Regression Results for MPG as Response Variable

Statistic	Value
R-squared	0.821
Adj. R-squared	0.818
F-statistic	252.4
Prob (F-statistic)	2.04×10^{-139}
Log-Likelihood	-1023.5
AIC	2063
BIC	2095
Durbin-Watson	1.309
Omnibus	31.906
Prob (Omnibus)	0.000
Jarque-Bera (JB)	53.100
Prob (JB)	2.95×10^{-12}
Skew	0.529
Kurtosis	4.460
Cond. No.	8.59×10^{4}

Table 5: Regression Statistics

d) Diagnostic Plots

Residual Plot Analysis

From Figure 5, ew can see that residuals show a slight **curved pattern**, indicating possible **non-linearity**. Increasing spread in residuals suggests **heteroscedasticity** (non-constant variance) and some residuals are far from zero, indicating potential **outliers**.

From Q-Q Plot analysis, we can say that residuals mostly follow a normal distribution, but deviations at the extremes suggest **outliers**. The upper tail shows points deviating from normality, which indicaties possible **skewness**.

The leverage plot which can be seen in Figure 6, identifies **observation 13** as having **unusually high leverage**, meaning it has a significant impact on the regression model. This data point is far from the rest and could be an **influential outlier**, potentially distorting predictions.

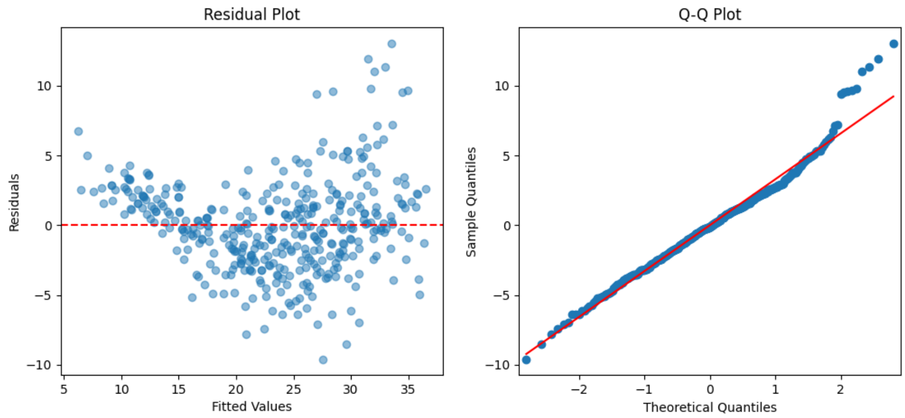

Figure 5: Diagnostic Plots for Multiple Linear Regression

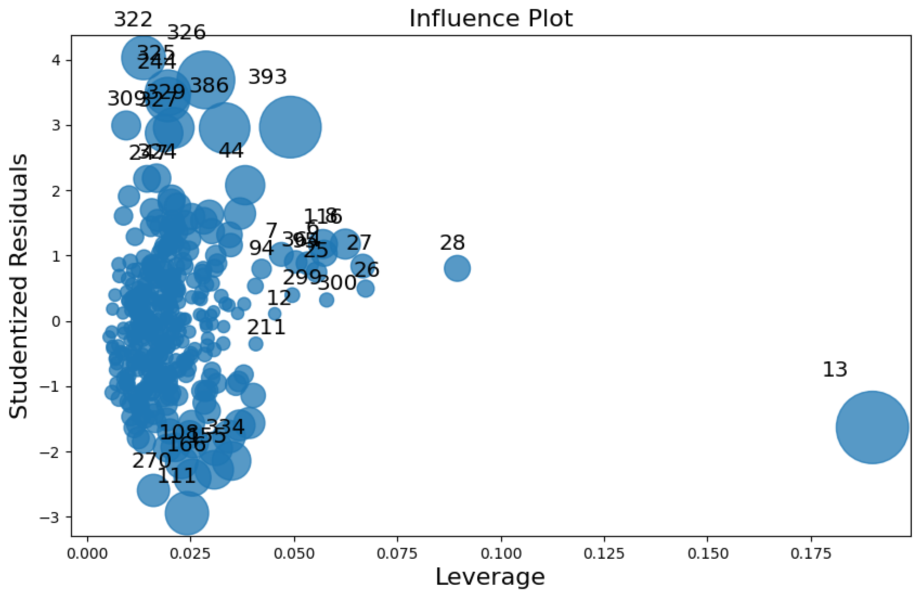

Figure 6: Leverage Plot for Multiple Linear Regression

e) Interaction Analysis

To analyze the effect of interaction terms, we added a new variable, horsepower * weight. Adding the interaction term **improves the model**, increasing R^2 from 0.821 to 0.862. The interaction effect is **highly significant** ($p = 0.000$), confirming that the relationship between *horsepower* and *mpg* depends on *weight*. AIC and BIC values are lower, indicating a **better model fit** and we can conclude that the effect of *horsepower* on *mpg* is stronger for heavier cars. We can see the new model analysis in Tables 6.

In Table 7, we compared previous and new model to see the effect of the newly added variable.

f) Transformations

To explore potential non-linear relationships, we applied the following transformations:

	Coefficient	Std. Error	t-value	P-value
const	2.8757	4.511	0.638	0.524
cylinders	-0.0296	0.288	-0.103	0.918
displacement	0.0059	0.007	0.881	0.379
horsepower	-0.2313	0.024	-9.791	0.000
weight	-0.0112	0.001	-15.393	0.000
acceleration	-0.0902	0.089	-1.019	0.309
year	0.7695	0.045	17.124	0.000
origin	0.8344	0.251	3.320	0.001
hp_weight	5.529e-05	5.23e-06	10.577	0.000

Table 6: OLS Regression Results with Interaction Term (‘horsepower $\times$ weight‘)

Statistic	Previous Model	With Interaction
R-squared	0.821	0.862
Adj. R-squared	0.818	0.859
F-statistic	252.4	298.6
AIC	2063	1964
BIC	2095	2000

Table 7: Comparison of Model Performance

- **Weight:** log(weight), $\sqrt{\text{weight}}$
- **Acceleration:** log(acceleration), acceleration^2, 1/acceleration
- **Displacement:** log(displacement), displacement^2

The results of the model can be seen in Table 8.

Statistic	With Transformations
R-squared	0.868
Adjusted R-squared	0.863
AIC	1959
BIC	2019

Table 8: Model Performance after Transformations

The model performance improved, with **$\mathbf{R^2}$ increasing to 0.868** and a lower AIC. Most transformations, including **log transformations of weight, acceleration, and displacement**, were **not significant** ($p > 0.05$), whereas **displacement squared (displacement^2) showed near significance** ($p = 0.055$), suggesting a potential non-linear effect. Overall, the model exhibits **high multicollinearity** (Condition Number = 9.69×10^8), indicating redundant predictors.

Python Code

Question 1

Listing 1: Linear Regression

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

# Load the Auto dataset
auto = pd.read_csv("Auto.csv")

#Horsepower is in object type
# print(auto.dtypes)

# Drop missing values if any
auto = auto.dropna()

# Convert 'horsepower' to numeric (if necessary)
auto["horsepower"] = pd.to_numeric(auto["horsepower"], errors
    ="coerce")

# Drop NaN values after conversion
auto = auto.dropna()

#After conversion horsepower is in float64 type
print(auto.dtypes)

# Define predictor (X) and response (y)
X = auto["horsepower"]
y = auto["mpg"]

# Add a constant for the intercept, without it model assumes
    it passes through the origin
X = sm.add_constant(X)

# Fit the regression model
model = sm.OLS(y, X).fit()

# Print the summary
print(model.summary())

# Part (a) - Answering the questions based on summary
# Predict mpg for horsepower of 98
hp_98 = np.array([[1, 98]])  # Constant term + horsepower
    value
predicted_mpg = model.predict(hp_98)[0]
```

```

# Compute confidence and prediction intervals
predictions = model.get_prediction(hp_98)
conf_int = predictions.conf_int(alpha=0.05)  # 95% confidence interval
pred_int = predictions.summary_frame(alpha=0.05)[["obs_ci_lower", "obs_ci_upper"]]

print(f"Predicted mpg for horsepower 98: {predicted_mpg}")
print(f"95% Confidence Interval: {conf_int}")
print(f"95% Prediction Interval: {pred_int}")

# Part (b) - Plotting the regression line
fig, ax = plt.subplots()
ax.scatter(auto["horsepower"], auto["mpg"], label="Data", alpha=0.5)
ax.axline((0, model.params["const"]), slope=model.params["horsepower"], color="red", label="Regression Line")
ax.set_xlabel("Horsepower")
ax.set_ylabel("MPG")
ax.legend()
ax.set_title("Linear Regression Model for Auto Dataset")
plt.savefig("img/regression.png")

# Part (c) - Diagnostic plots
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Residual plot
axes[0].scatter(model.fittedvalues, model.resid, alpha=0.5)
axes[0].axhline(0, color="red", linestyle="dashed")
axes[0].set_xlabel("Fitted Values")
axes[0].set_ylabel("Residuals")
axes[0].set_title("Residual Plot")

# Q-Q Plot for normality check
sm.qqplot(model.resid, line="s", ax=axes[1])
axes[1].set_title("Q-Q Plot")
plt.savefig("img/residual_qq.png")
```

Question 2

Listing 2: Multiple Linear Regression

```

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
import statsmodels.api as sm
from statsmodels.stats.anova import anova_lm
import statsmodels.formula.api as smf

# Load the dataset
auto = pd.read_csv("Auto.csv")

# Convert 'horsepower' to numeric
auto["horsepower"] = pd.to_numeric(auto["horsepower"], errors="coerce")

# Drop missing values
auto = auto.dropna()

# Drop the 'name' column as it is not a numeric predictor
auto = auto.drop(columns=["name"])

sns.pairplot(auto)
plt.savefig("img/scatterplot_matrix.png")

correlation_matrix = auto.corr()
print(correlation_matrix)

# Heatmap visualization
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)
plt.title("Correlation Matrix")
plt.savefig("img/correlation_matrix.png")

# Define predictors (all except 'mpg')
X = auto.drop(columns=["mpg"])
y = auto["mpg"]

# Add a constant for the intercept
X = sm.add_constant(X)

# Fit the multiple linear regression model
model = sm.OLS(y, X).fit()

# Print the regression summary
print(model.summary())


formula = "mpg ~ " + " + ".join(auto.drop(columns=["mpg"]).columns)
anova_model = smf.ols(formula=formula, data=auto).fit()
anova_results = anova_lm(anova_model)
```

```python
print(anova_results)

fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Residual plot
axes[0].scatter(model.fittedvalues, model.resid, alpha=0.5)
axes[0].axhline(0, color="red", linestyle="dashed")
axes[0].set_xlabel("Fitted Values")
axes[0].set_ylabel("Residuals")
axes[0].set_title("Residual Plot")

# Q-Q Plot for normality check
sm.qqplot(model.resid, line="s", ax=axes[1])
axes[1].set_title("Q-Q Plot")

plt.savefig("img/diagnostic_plots.png")

# Generate the leverage plot (influence plot)
fig, ax = plt.subplots(figsize=(10, 6))
sm.graphics.influence_plot(model, ax=ax, criterion="cooks")

# Save the leverage plot
plt.savefig("img/leverage_plot.png")

# Add interaction term: horsepower * weight
auto["hp_weight"] = auto["horsepower"] * auto["weight"]

# Refit the model with interaction
X_interact = auto.drop(columns=["mpg"])
X_interact = sm.add_constant(X_interact)
model_interact = sm.OLS(y, X_interact).fit()

# Print summary
print(model_interact.summary())

# Apply transformations
auto["log_horsepower"] = np.log(auto["horsepower"])
auto["sqrt_horsepower"] = np.sqrt(auto["horsepower"])
auto["sq_horsepower"] = auto["horsepower"] ** 2

# Fit model with transformations
X_trans = auto.drop(columns=["mpg"])
X_trans = sm.add_constant(X_trans)
model_trans = sm.OLS(y, X_trans).fit()

# Print summary
print(model_trans.summary())

```

```python
import numpy as np
import statsmodels.api as sm

# Apply transformations
auto["log_weight"] = np.log(auto["weight"])
auto["sqrt_weight"] = np.sqrt(auto["weight"])

auto["log_acceleration"] = np.log(auto["acceleration"])
auto["acceleration_squared"] = auto["acceleration"] ** 2
auto["inv_acceleration"] = 1 / auto["acceleration"]

auto["log_displacement"] = np.log(auto["displacement"])
auto["displacement_squared"] = auto["displacement"] ** 2


# Define predictors (including new transformed variables)
X_trans = auto[[
    "cylinders", "displacement", "horsepower", "weight", "acceleration", "year", "origin",
    "log_weight", "sqrt_weight",
    "log_acceleration", "acceleration_squared", "inv_acceleration",
    "log_displacement", "displacement_squared"
]]

# Add a constant for the intercept
X_trans = sm.add_constant(X_trans)

# Fit the model
model_trans = sm.OLS(auto["mpg"], X_trans).fit()

# Print summary
print(model_trans.summary())

# Extract key performance metrics
print(f"R-squared: {model_trans.rsquared:.3f}")
print(f"Adjusted R-squared: {model_trans.rsquared_adj:.3f}")
print(f"AIC: {model_trans.aic:.2f}")
print(f"BIC: {model_trans.bic:.2f}")
```