PRACTICAL 9

Name:	Harsh Shah	**Semester:**	VII	**Division:**	6
Roll No.:	21BCP359	**Date:**	01-10-24	**Batch:**	G11

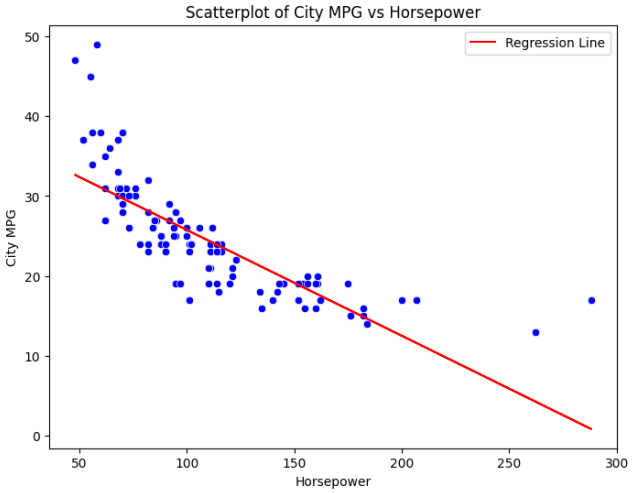
Code

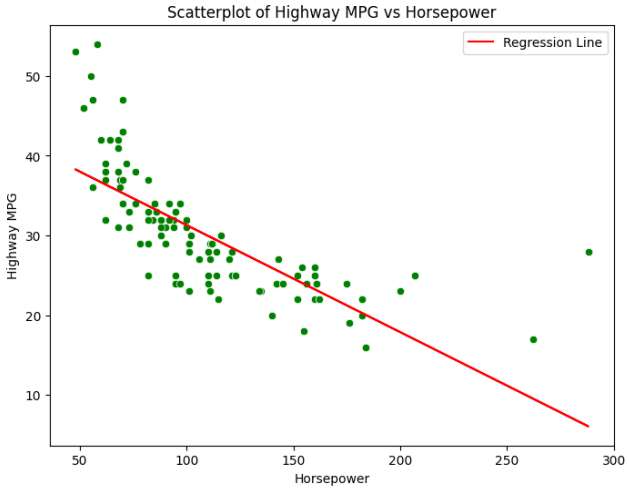
```
import pandas as pd
import statsmodels.api as sm

df = pd.read_csv('CarPrice_Assignment.csv')

X = df['horsepower']
y = df['citympg']

X = sm.add_constant(X)

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

print(model_citympg.summary())

X_highway = df['horsepower']
y_highway = df['highwaympg']

X_highway = sm.add_constant(X_highway)

# Create the regression model
model_highway = sm.OLS(y_highway, X_highway).fit()

print(model_highway.summary())


import matplotlib.pyplot as plt
import seaborn as sns

# Scatterplot for citympg vs. horsepower
plt.figure(figsize=(8, 6))
sns.scatterplot(x=df['horsepower'], y=df['citympg'], color='blue')
plt.plot(df['horsepower'], model_citympg.predict(sm.add_constant(df['horsepower'])), color='red',
label='Regression Line')
plt.title('Scatterplot of City MPG vs Horsepower')
plt.xlabel('Horsepower')
plt.ylabel('City MPG')
plt.legend()
plt.show()

# Scatterplot for highwaympg vs. horsepower
plt.figure(figsize=(8, 6))
sns.scatterplot(x=df['horsepower'], y=df['highwaympg'], color='green')
```

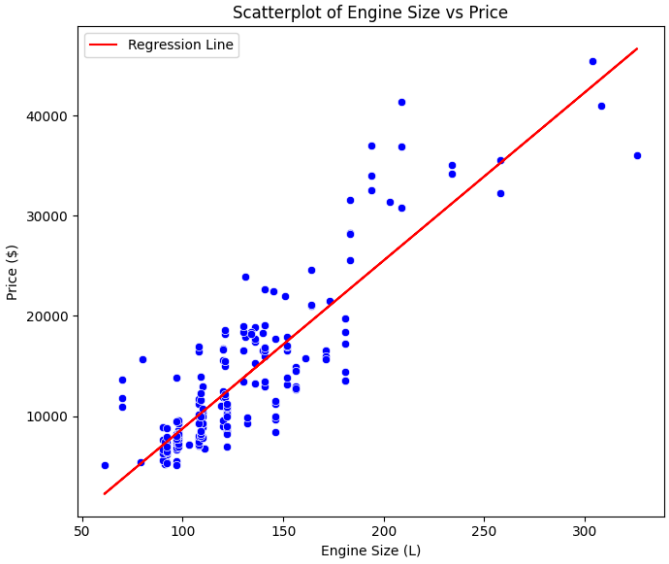
```
plt.plot(df['horsepower'], model_highway.predict(sm.add_constant(df['horsepower'])), color='red', label='Regression Line')
plt.title('Scatterplot of Highway MPG vs Horsepower')
plt.xlabel('Horsepower')
plt.ylabel('Highway MPG')
plt.legend()
plt.show()


# Regression Model 1: price as dependent and citympg as independent variable
X_citympg = sm.add_constant(df['citympg'])
model_price_citympg = sm.OLS(df['price'], X_citympg).fit()

# Display model statistics for price vs. citympg
print("Model 1: price vs citympg")
print(model_price_citympg.summary())

# Regression Model 2: price as dependent and highwaympg as independent variable
X_highwaympg = sm.add_constant(df['highwaympg'])  # Add constant term
model_price_highwaympg = sm.OLS(df['price'], X_highwaympg).fit()

# Display model statistics for price vs. highwaympg
print("\nModel 2: price vs highwaympg")
print(model_price_highwaympg.summary())



# Model 1: Regression of 'price' on 'enginesize'
X_enginesize = df[['enginesize']]
y_price = df['price']

# Add a constant (intercept)
X_enginesize = sm.add_constant(X_enginesize)

# Create and fit the regression model
model_enginesize = sm.OLS(y_price, X_enginesize).fit()

# Output the model summary
print(model_enginesize.summary())


# Model 2: Regression of 'price' on 'curbweight'
X_curbweight = df[['curbweight']]  # Independent variable
y_price = df['price']  # Dependent variable

# Add a constant (intercept)
X_curbweight = sm.add_constant(X_curbweight)

# Create and fit the regression model
model_curbweight = sm.OLS(y_price, X_curbweight).fit()

# Output the model summary
```

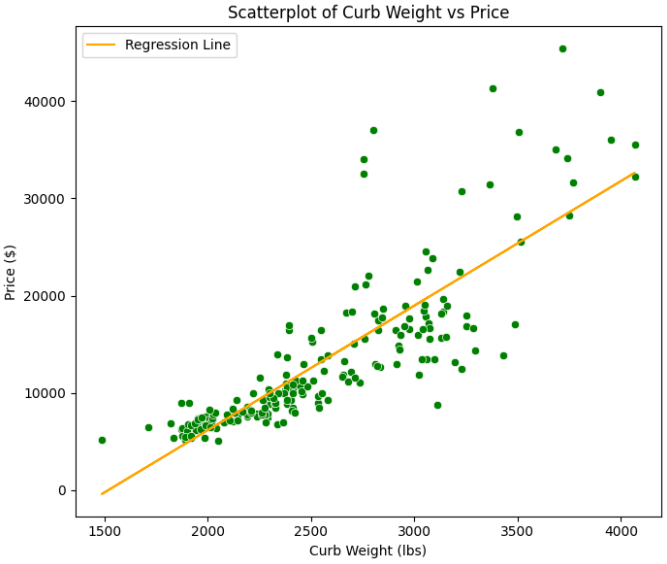
```
print(model_curbweight.summary())


# Scatterplot for Engine Size vs. Price
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.scatterplot(x=df['enginesize'], y=df['price'], color='blue')
plt.plot(df['enginesize'], model_enginesize.predict(sm.add_constant(df['enginesize'])), color='red', label='Regression Line')
plt.title('Scatterplot of Engine Size vs Price')
plt.xlabel('Engine Size (L)')
plt.ylabel('Price ($)')
plt.legend()

# Scatterplot for Curb Weight vs. Price
plt.subplot(1, 2, 2)
sns.scatterplot(x=df['curbweight'], y=df['price'], color='green')
plt.plot(df['curbweight'], model_curbweight.predict(sm.add_constant(df['curbweight'])), color='orange', label='Regression Line')
plt.title('Scatterplot of Curb Weight vs Price')
plt.xlabel('Curb Weight (lbs)')
plt.ylabel('Price ($)')
plt.legend()

# Show the plots
plt.tight_layout()
plt.show()


import pandas as pd
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor

df = pd.read_csv('CarPrice_Assignment.csv')

# Select numeric variables except 'citympg' and 'highwaympg'
numeric_df = df.drop(['price', 'citympg', 'highwaympg'], axis=1).select_dtypes(include=[float, int])

independent_vars = sm.add_constant(numeric_df)

# Variance Inflation Factor (VIF) calculation
vif_data = pd.DataFrame()
vif_data['Feature'] = independent_vars.columns
vif_data['VIF'] = [variance_inflation_factor(independent_vars.values, i) for i in range(independent_vars.shape[1])]

print(vif_data)
```

Output


Scatterplot of City MPG vs Horsepower
Regression Line
City MPG
Horsepower


Scatterplot of Highway MPG vs Horsepower
Regression Line
Highway MPG
Horsepower


Scatterplot of Engine Size vs Price
Regression Line
Price ($)
Engine Size (L)


Scatterplot of Curb Weight vs Price
Regression Line
Price ($)
Curb Weight (lbs)