Assignment #2

2. Carefully explain the differences between the KNN classifier and KNN regression methods.

The KNN classifier predicts categorical classes based on the majority vote among its k nearest neighbors, suitable for qualitative responses. In contrast, KNN regression estimates continuous values for quantitative variables by averaging neighboring target values, fitting numerical predictions.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns; sns.set()
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
import statsmodels.formula.api as smf
from statsmodels.stats.outliers_influence import OLSInfluence
import patsy
from scipy import stats
from sklearn import datasets
from IPython.display import display, HTML
```

9. This question involves the use of multiple linear regression on the Auto data set.

a. Produce a scatterplot matrix which includes all of the variables in the data set.

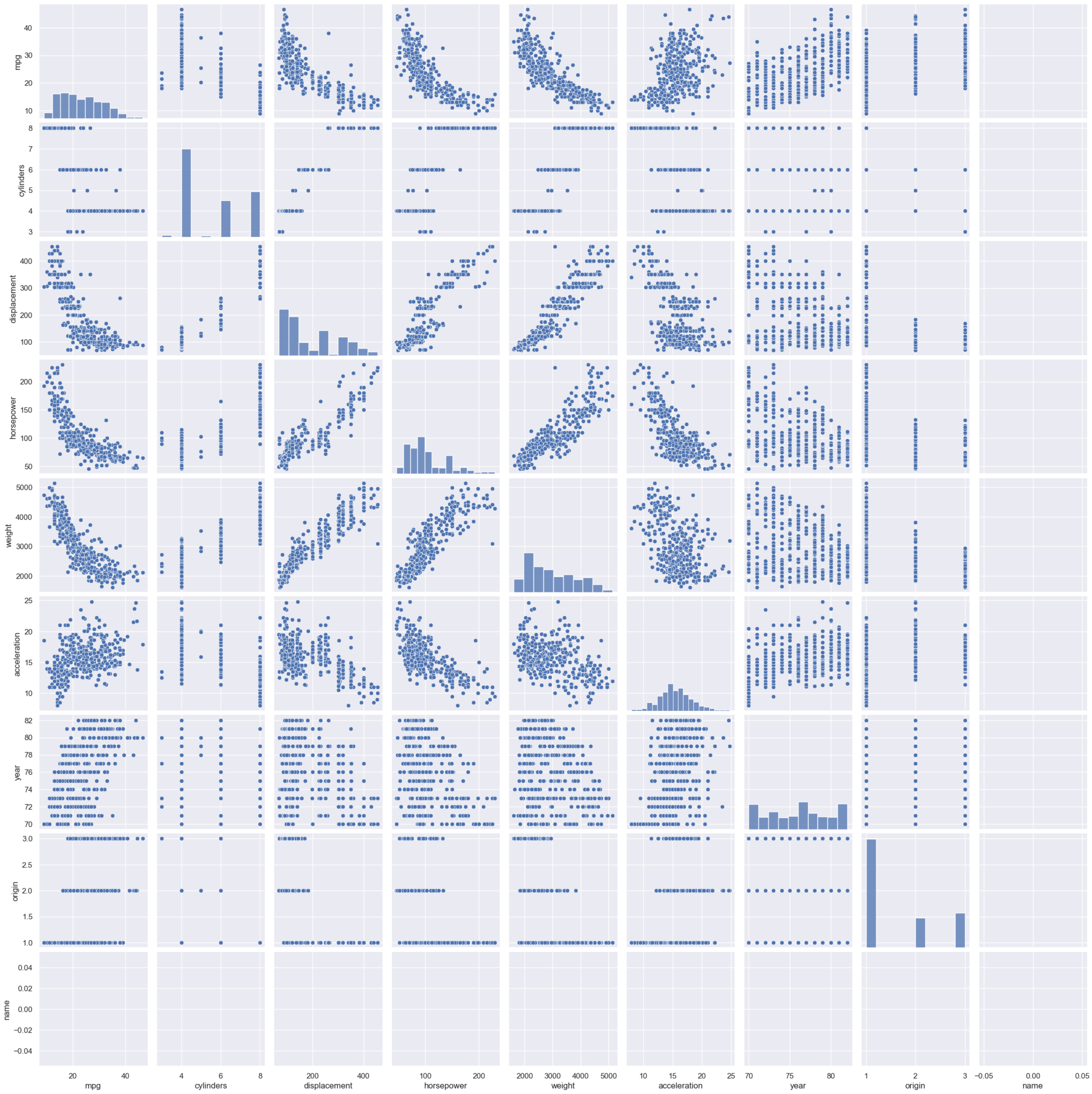
In [2]:
```
auto = pd.read_csv("/Users/kenziekenz/Desktop/Auto.csv")
auto = auto.drop(auto[auto.values == '?'].index)
auto = auto.reset_index()
auto = auto.apply(pd.to_numeric, errors='coerce')
```

In [3]:
```
datatypes = {'quant': ['mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'year'],
             'qual': ['origin', 'name']}

quants = auto[datatypes['quant']].astype(np.float_)

auto = pd.concat([quants, auto[datatypes['qual']]], axis=1)
```

In [4]:
```
sns.pairplot(auto)
```

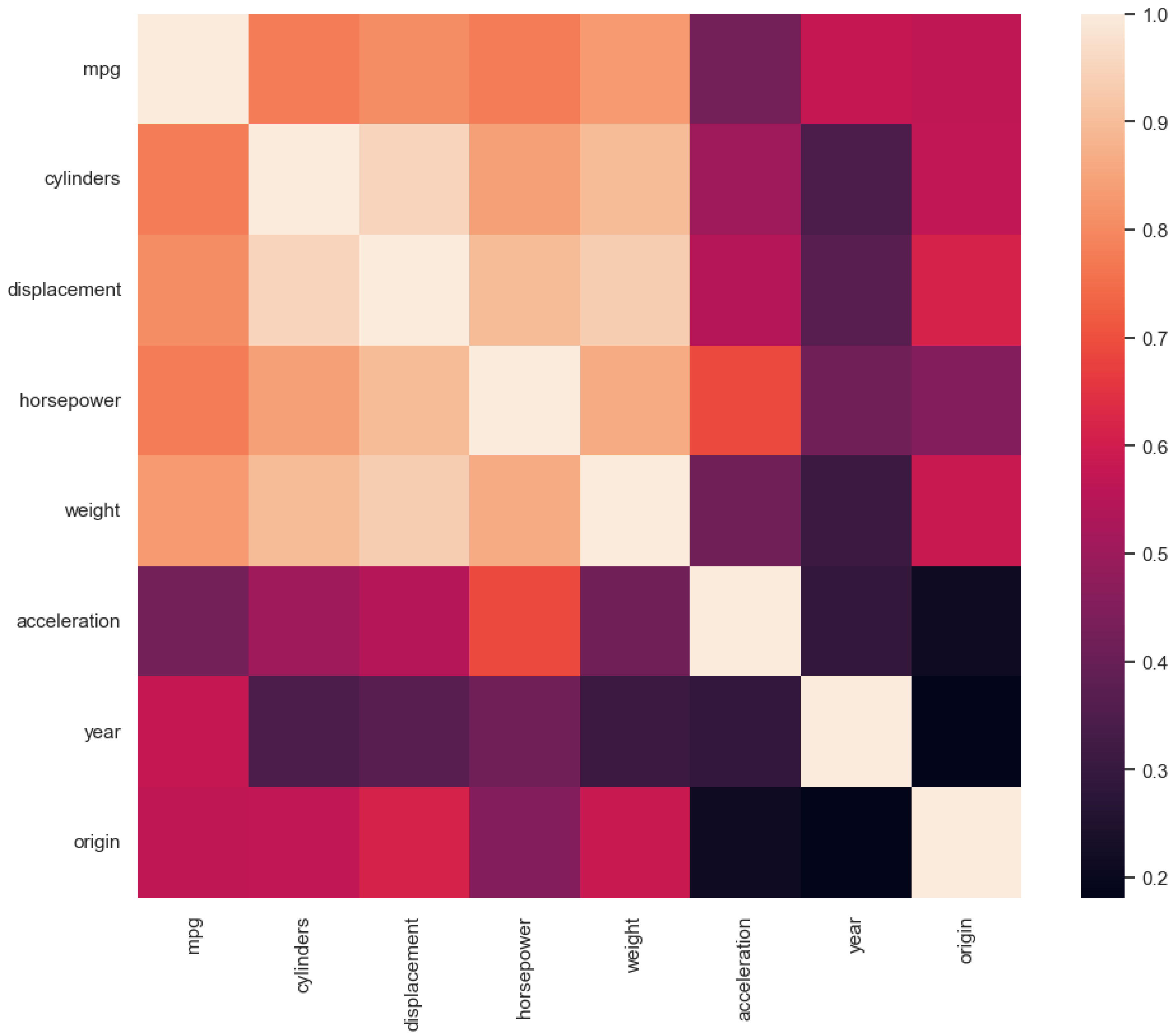
```
/Users/kenziekenz/anaconda3/lib/python3.11/site-packages/seaborn/axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[4]:
```
<seaborn.axisgrid.PairGrid at 0x168025850>
```



b. Compute the matrix of correlations between the variables using the DataFrame.corr() method.

In [5]:
```
if 'name' in auto.columns:
    auto = auto.drop(columns=['name'])

corr_matrix = auto.corr().abs()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corr_matrix, vmax=1, square=True)
plt.xticks(rotation=90)
plt.yticks(rotation=0);
```



c. Use the sm.OLS() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summarize() function to print the results. Comment on the output. For instance:

In [6]:
```
formula = 'mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + C(origin)'
y, X = patsy.dmatrices(formula, data=auto, return_type='dataframe')

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

model.pvalues[model.pvalues < 0.05].sort_values()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.824
Model:                            OLS   Adj. R-squared:                  0.821
Method:                 Least Squares   F-statistic:                     224.5
Date:                Tue, 20 Feb 2024   Prob (F-statistic):          1.79e-139
Time:                        22:06:47   Log-Likelihood:                -1020.5
No. Observations:                 392   AIC:                             2059.
Df Residuals:                     383   BIC:                             2095.
Df Model:                           8
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept        -17.9546      4.677     -3.839      0.000     -27.150      -8.759
C(origin)[T.2]     2.6300      0.566      4.643      0.000       1.516       3.744
C(origin)[T.3]     2.8532      0.553      5.162      0.000       1.766       3.940
cylinders         -0.4897      0.321     -1.524      0.128      -1.121       0.142
displacement       0.0240      0.008      3.133      0.002       0.009       0.039
horsepower        -0.0182      0.014     -1.326      0.185      -0.045       0.009
weight            -0.0067      0.001    -10.243      0.000      -0.008      -0.005
acceleration       0.0791      0.098      0.805      0.421      -0.114       0.272
year               0.7770      0.052     15.005      0.000       0.675       0.879
==============================================================================
Omnibus:                       23.395   Durbin-Watson:                   1.291
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               34.452
Skew:                           0.444   Prob(JB):                     3.30e-08
Kurtosis:                       4.150   Cond. No.                     8.70e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.7e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Out[6]:

```
year              2.332943e-40
weight            6.375633e-22
C(origin)[T.3]    3.933208e-07
C(origin)[T.2]    4.720373e-06
Intercept         1.445124e-04
displacement      1.862685e-03
dtype: float64
```

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

In [7]:

```python
formula = 'mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + C(origin)'

model = ols(formula, data=auto).fit()

anova_table = anova_lm(model)

print(anova_table)
```

```
                 df       sum_sq      mean_sq           F         PR(>F)
C(origin)       2.0  7904.291038  3952.145519  361.483198   6.397133e-89
cylinders       1.0  7067.298967  7067.298967  646.410872   3.035644e-84
displacement    1.0   793.509247   793.509247   72.578365   3.723110e-16
horsepower      1.0   584.192513   584.192513   53.433199   1.572821e-12
weight          1.0   819.505257   819.505257   74.956091   1.356065e-16
acceleration    1.0     1.166009     1.166009    0.106649   7.441703e-01
year            1.0  2461.638760  2461.638760  225.153919   2.332943e-40
Residual      383.0  4187.391678    10.933138         NaN            NaN
```

There is a relationship between most predictors and the response variable, except for "acceleration" which doesn't appear to have a significant relationship with "mpg."

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

Predictors with statistically significant relationships to the response variable "mpg" ($p < 0.05$) are: cylinders, displacement, horsepower, weight, year, C

iii. What does the coefficient for the year variable suggest?

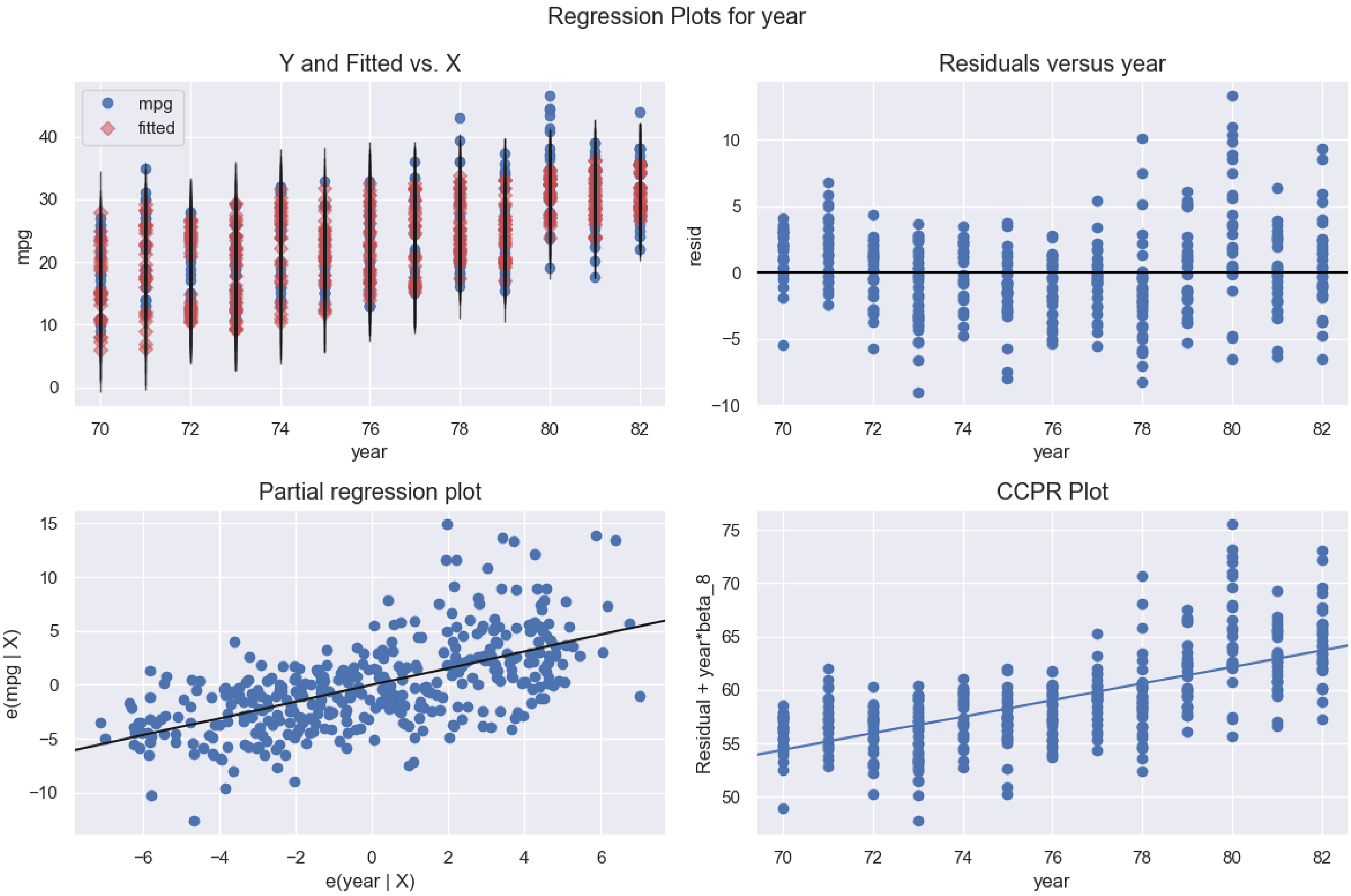
Based on the regression analysis, there is evidence to suggest that there is a statistically significant and positive relationship between the model year of a car and its fuel efficiency, measured by miles per gallon.The coeffecient of 0.7770 means that for each one-unit increase in the "year", the "mpg" tends to increase by approximately 0.7770 units.

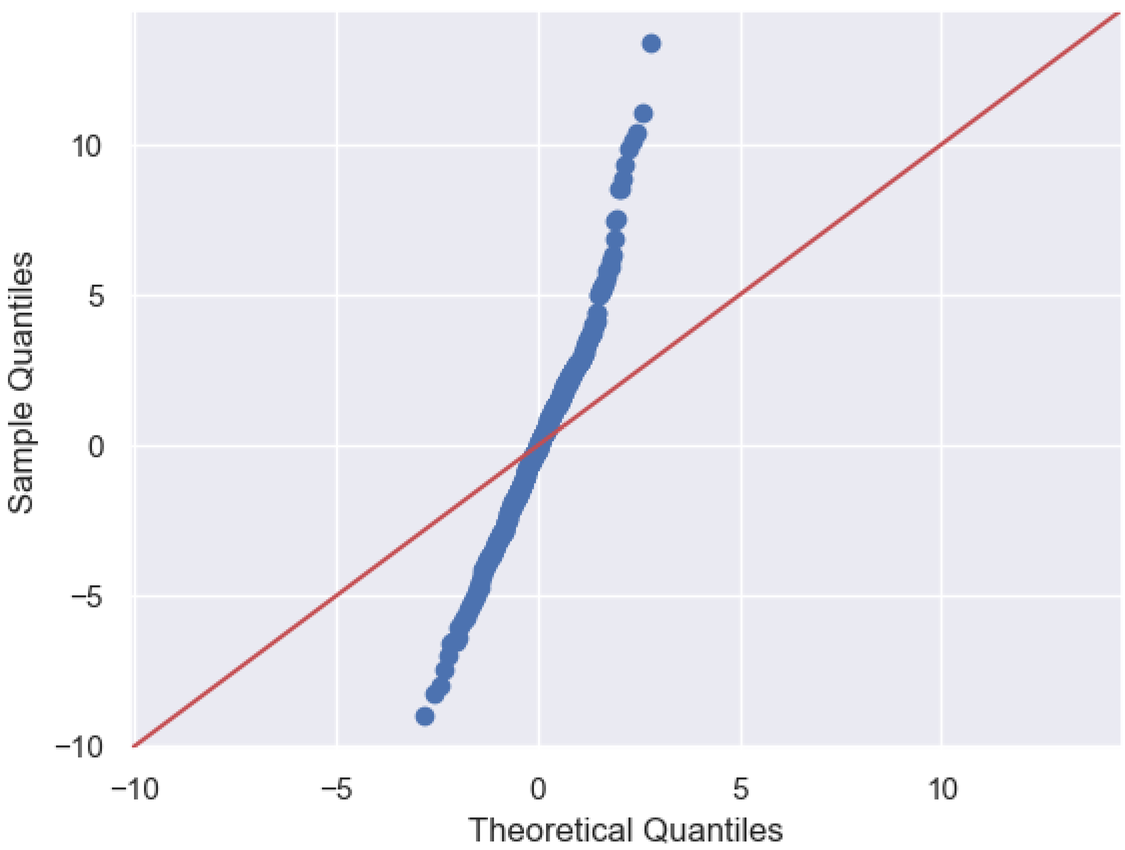
d. Produce some of diagnostic plots of the linear regression fit as described in the lab. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

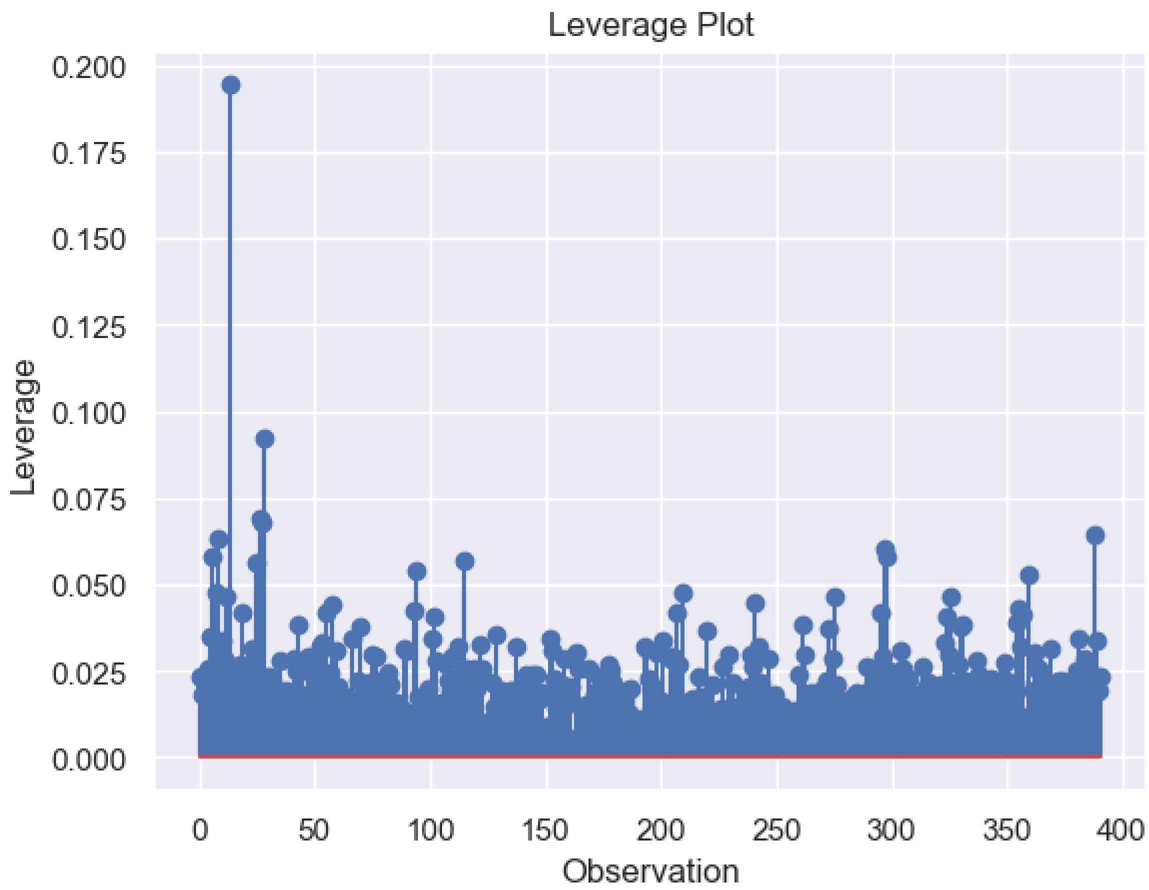
In [8]:

```python
sm.graphics.plot_regress_exog(model, 'year', fig=plt.figure(figsize=(12, 8)))
plt.show()

residuals = model.resid
fig = sm.qqplot(residuals, line='45')
plt.show()
```

```
influence = OLSInfluence(model)
leverage = influence.hat_matrix_diag
plt.stem(leverage)
plt.xlabel('Observation')
plt.ylabel('Leverage')
plt.title('Leverage Plot')
plt.show()
```







There doesn't seem to be any large outliers and the leverage plot seems to show a value that is significantly larger than the others.

e. Fit some models with interactions as described in the lab. Do any interactions appear to be statistically significant?

In [9]:
```
formula_with_interactions = 'mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + C(origin) +
model_with_interactions = ols(formula_with_interactions, data=auto).fit()

print(model_with_interactions.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.869
Model:                            OLS   Adj. R-squared:                  0.866
Method:                 Least Squares   F-statistic:                     252.8
Date:                Tue, 20 Feb 2024   Prob (F-statistic):          2.23e-161
Time:                        22:06:47   Log-Likelihood:                -962.78
No. Observations:                 392   AIC:                             1948.
Df Residuals:                     381   BIC:                             1991.
Df Model:                          10
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
Intercept           -34.6204      8.052     -4.299      0.000     -50.453     -18.788
C(origin)[T.2]        1.9919      0.495      4.021      0.000       1.018       2.966
C(origin)[T.3]        1.4288      0.497      2.878      0.004       0.453       2.405
cylinders            -4.0749      0.569     -7.161      0.000      -5.194      -2.956
displacement          0.2286      0.034      6.701      0.000       0.162       0.296
horsepower           -0.0435      0.012     -3.599      0.000      -0.067      -0.020
weight               -0.0138      0.001    -12.728      0.000      -0.016      -0.012
acceleration          0.1177      0.085      1.384      0.167      -0.050       0.285
year                  1.2892      0.089     14.418      0.000       1.113       1.465
cylinders:weight      0.0013      0.000      8.334      0.000       0.001       0.002
displacement:year    -0.0029      0.000     -6.248      0.000      -0.004      -0.002
==============================================================================
Omnibus:                       41.489   Durbin-Watson:                   1.570
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               95.397
Skew:                           0.550   Prob(JB):                     1.93e-21
Kurtosis:                       5.152   Cond. No.                     1.47e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.47e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

The interaction between "cylinders" and "weight" and the interaction between "displacement" and "year" appear to be statistically significant and should be considered when interpreting the relationship between predictors and the response variable "mpg."

f. Try a few different transformations of the variables, such as log(X), √X, X2. Comment on your findings.

In [10]:
```
auto['log_displacement'] = np.log(auto['displacement'])
auto['sqrt_weight'] = np.sqrt(auto['weight'])
auto['weight_squared'] = auto['weight'] ** 2

transformed_formula = 'mpg ~ cylinders + log_displacement + horsepower + sqrt_weight + acceleration + year + C(origin)
model_transformed = ols(transformed_formula, data=auto).fit()

print(model_transformed.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.835
Model:                            OLS   Adj. R-squared:                  0.831
Method:                 Least Squares   F-statistic:                     241.9
Date:                Tue, 20 Feb 2024   Prob (F-statistic):          1.26e-144
Time:                        22:06:47   Log-Likelihood:                -1008.3
No. Observations:                 392   AIC:                             2035.
Df Residuals:                     383   BIC:                             2070.
Df Model:                           8
Covariance Type:            nonrobust
====================================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
Intercept            7.5877      6.492      1.169      0.243      -5.178      20.353
C(origin)[T.2]       1.3237      0.589      2.249      0.025       0.166       2.481
C(origin)[T.3]       1.3600      0.587      2.316      0.021       0.206       2.515
cylinders            0.6239      0.299      2.084      0.038       0.035       1.212
log_displacement    -2.6186      1.487     -1.762      0.079      -5.541       0.304
horsepower           0.0024      0.012      0.191      0.848      -0.022       0.027
sqrt_weight         -0.6449      0.078     -8.229      0.000      -0.799      -0.491
acceleration         0.0686      0.096      0.715      0.475      -0.120       0.257
year                 0.7752      0.050     15.447      0.000       0.676       0.874
==============================================================================
Omnibus:                       34.744   Durbin-Watson:                   1.279
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               67.358
Skew:                           0.516   Prob(JB):                     2.36e-15
Kurtosis:                       4.749   Cond. No.                     5.88e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.88e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

The R-squared value of 0.835 indicates that approximately 83.5% of the variance in the response variable (mpg) is explained by the predictors in the model. The predictors "cylinders," "sqrt_weight," and "year" have p-values less than 0.05, indicating that they are statistically significant in predicting mpg. The predictors "log_displacement," "horsepower," and "acceleration" have p-values greater than 0.05, suggesting that they may not be statistically significant predictors in this model.

10. This question should be answered using the Carseats data set.

a. Fit a multiple regression model to predict Sales using Price, Urban, and US.

In [11]:

```
carseats = pd.read_csv('/Users/kenziekenz/Desktop/Carseats.csv')

datatypes = {'quant': ['Sales', 'CompPrice', 'Income', 'Advertising', 'Population', 'Price', 'Age', 'Education'],
             'qual': ['ShelveLoc', 'Urban', 'US']}

quants = carseats[datatypes['quant']].astype(np.float_)
carseats_df = pd.concat([quants, carseats[datatypes['qual']]], axis=1)

carseats_df.head()
```

Out[11]:

	Sales	CompPrice	Income	Advertising	Population	Price	Age	Education	ShelveLoc	Urban	US
0	9.50	138.0	73.0	11.0	276.0	120.0	42.0	17.0	Bad	Yes	Yes
1	11.22	111.0	48.0	16.0	260.0	83.0	65.0	10.0	Good	Yes	Yes
2	10.06	113.0	35.0	10.0	269.0	80.0	59.0	12.0	Medium	Yes	Yes
3	7.40	117.0	100.0	4.0	466.0	97.0	55.0	14.0	Medium	Yes	Yes
4	4.15	141.0	64.0	3.0	340.0	128.0	38.0	13.0	Bad	Yes	No

In [12]:

```
import statsmodels.api as sm

f = 'Sales ~ Price + C(Urban) + C(US)'
y, X = patsy.dmatrices(f, carseats_df, return_type='dataframe')

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

y_pred = model.predict(X)
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.239
Model:                            OLS   Adj. R-squared:                  0.234
Method:                 Least Squares   F-statistic:                     41.52
Date:                Tue, 20 Feb 2024   Prob (F-statistic):           2.39e-23
Time:                        22:06:47   Log-Likelihood:                -927.66
No. Observations:                 400   AIC:                             1863.
Df Residuals:                     396   BIC:                             1879.
Df Model:                           3                                         
Covariance Type:            nonrobust                                         
===================================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
Intercept          13.0435      0.651     20.036      0.000      11.764      14.323
C(Urban)[T.Yes]    -0.0219      0.272     -0.081      0.936      -0.556       0.512
C(US)[T.Yes]        1.2006      0.259      4.635      0.000       0.691       1.710
Price              -0.0545      0.005    -10.389      0.000      -0.065      -0.044
==============================================================================
Omnibus:                        0.676   Durbin-Watson:                   1.912
Prob(Omnibus):                  0.713   Jarque-Bera (JB):                0.758
Skew:                           0.093   Prob(JB):                        0.684
Kurtosis:                       2.897   Cond. No.                         628.
==============================================================================

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

b. Provide an interpretation of each coefficient in the model. Becareful—some of the variables in the model are qualitative!

As the Price of car seats increases, Sales tend to decrease. On average, for every 1000 increase in Price, Sales decrease by approximately 54.50 dollars.

There appears to be a statistically significant association between Sales and the location of the store. Specifically, car seats sold in the United States are expected to achieve an average sale price approximately $1,200 higher than those sold elsewhere.

However, we found no substantial relationship between Sales and whether the store is situated in an urban or rural area.

c. Write out the model in equation form, being careful to handle the qualitative variables properly.

$\hat{Y}$ = 13.045 + (-0.0219 *Urban) + (1.2006* US) + (-0.0545 * Price)

d. For which of the predictors can you reject the null hypothesis H0: βj =0?

Price and US

e. On the basis of your response to the previous question, fit a smaller model that only uses the predictors for which there is evidence of association with the outcome.

In [13]:

```python
import statsmodels.api as sm
import patsy

f = 'Sales ~ Price + C(US)'
y, X = patsy.dmatrices(f, carseats_df, return_type='dataframe')

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

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.239
Model:                            OLS   Adj. R-squared:                  0.235
Method:                 Least Squares   F-statistic:                     62.43
Date:                Tue, 20 Feb 2024   Prob (F-statistic):           2.66e-24
Time:                        22:06:47   Log-Likelihood:                -927.66
No. Observations:                 400   AIC:                             1861.
Df Residuals:                     397   BIC:                             1873.
Df Model:                           2                                         
Covariance Type:            nonrobust                                         
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       13.0308      0.631     20.652      0.000      11.790      14.271
C(US)[T.Yes]     1.1996      0.258      4.641      0.000       0.692       1.708
Price           -0.0545      0.005    -10.416      0.000      -0.065      -0.044
==============================================================================
Omnibus:                        0.666   Durbin-Watson:                   1.912
Prob(Omnibus):                  0.717   Jarque-Bera (JB):                0.749
Skew:                           0.092   Prob(JB):                        0.688
Kurtosis:                       2.895   Cond. No.                         607.
==============================================================================

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

f. How well do the models in (a) and (e) fit the data?

In [14]:
```
import matplotlib.pyplot as plt

plt.scatter(y_pred, model.resid)
plt.axhline(y=0, color='r', linestyle='-')
plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.show()
```

The plot shows a slight pattern, suggesting that our linear model fits the data reasonably well. Additionally, the residuals (the differences between predicted and actual values) seem to follow a normal distribution, and there are no signs of unequal variability (heteroscedasticity) in the data.

g. Using the model from (e), obtain 95% confidence intervals for the coefficient(s).

In [15]:
```
conf_inter_95 = model.conf_int(alpha=0.05)
conf_inter_95.rename(index=str, columns={0: "min.", 1: "max.",})
```

Out[15]:

	min.	max.
Intercept	11.79032	14.271265
C(US)[T.Yes]	0.69152	1.707766
Price	-0.06476	-0.044195

h. Is there evidence of outliers or high leverage observations in the model from (e)?

None of the observations surpass the threshold for outliers set at +/-3 for studentized residuals. However, a couple of observations come close to this threshold.

12. This problem involves simple linear regression without an intercept.

a. Recall that the coefficient estimate β^ for the linear regression of Y onto X without an intercept is given by (3.38). Under what circumstance is the coefficient estimate for the regression of X onto Y the same as the coefficient estimate for the regression of Y onto X?

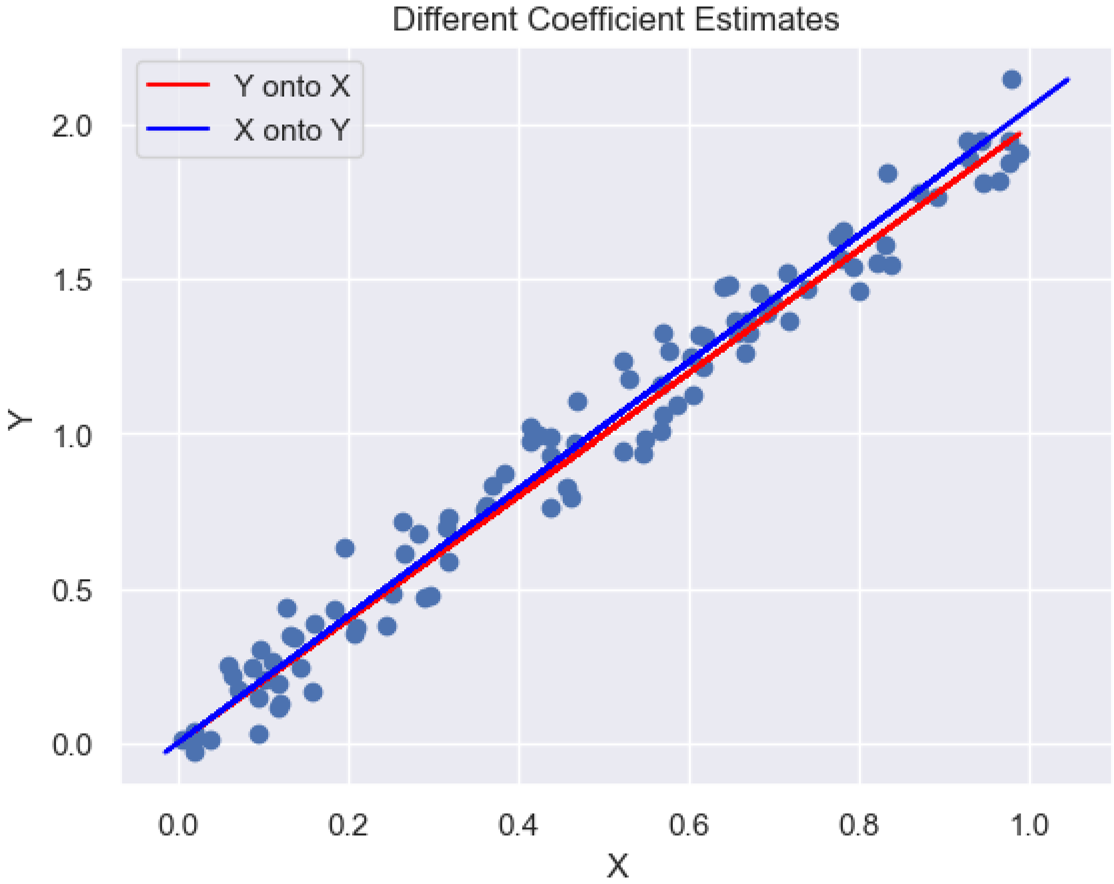
The coefficient estimate for the regression of X onto Y is the same as the coefficient estimate for the regression of Y onto X when the variables X and Y are perfectly correlated. When there is a perfect linear relationship between X and Y, the slopes of the regression lines will be identical.

b. Generate an example in Python with n = 100 observations in which the coefficient estimate for the regression of X onto Y is different from the coefficient estimate for the regression of Y onto X.

In [16]:
```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(0)
X = np.random.rand(100)
Y = 2*X + np.random.normal(0, 0.1, 100)

coeff_Y_on_X = np.cov(X, Y, ddof=0)[0, 1] / np.var(X)
coeff_X_on_Y = np.cov(X, Y, ddof=0)[1, 0] / np.var(Y)
```

```
print("Coefficient estimate for Y onto X:", coeff_Y_on_X)
print("Coefficient estimate for X onto Y:", coeff_X_on_Y)

plt.scatter(X, Y)
plt.plot(X, coeff_Y_on_X*X, color='red', label='Y onto X')
plt.plot(coeff_X_on_Y*Y, Y, color='blue', label='X onto Y')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.title('Different Coefficient Estimates')
plt.show()
```

```
Coefficient estimate for Y onto X: 1.993693502140204
Coefficient estimate for X onto Y: 0.48695376206001306
```

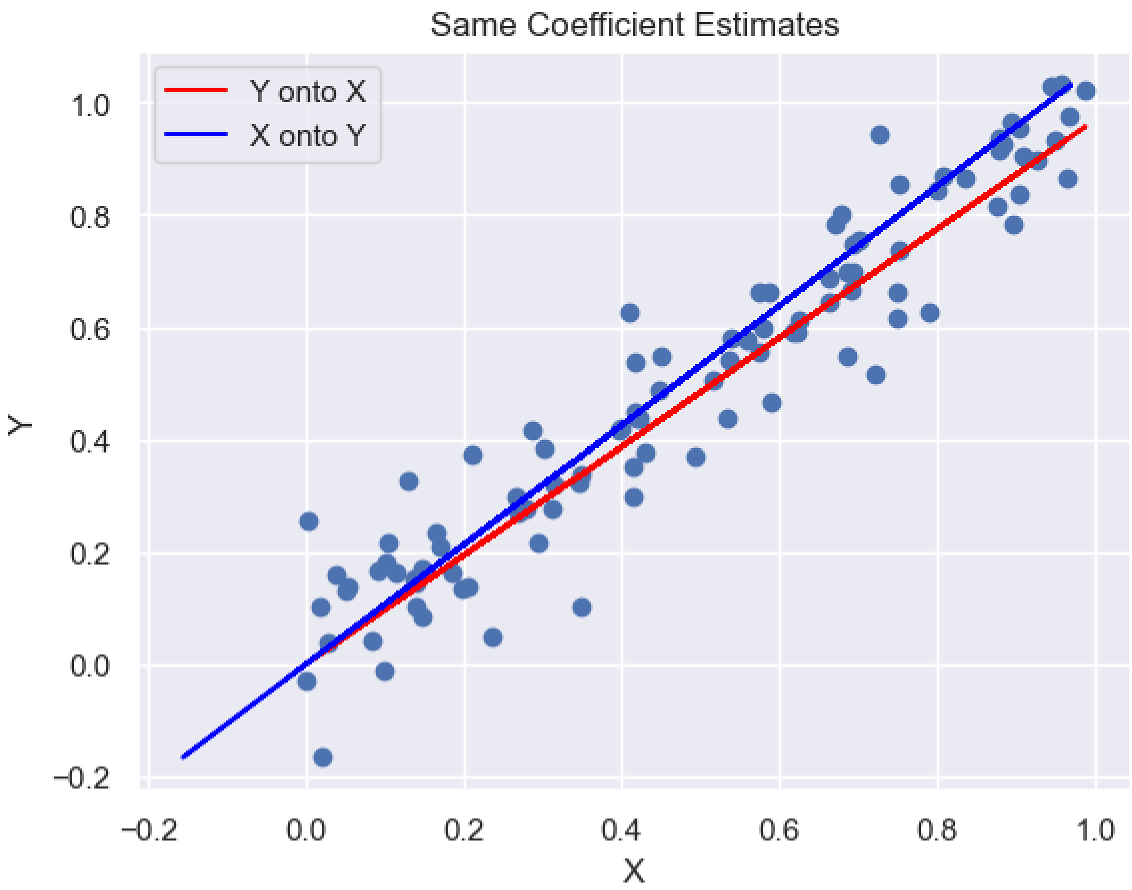


c. Generate an example in Python with n = 100 observations in which the coefficient estimate for the regression of X onto Y is the same as the coefficient estimate for the regression of Y onto X.

In [17]:

```
np.random.seed(1)
X = np.random.rand(100)
Y = X + np.random.normal(0, 0.1, 100)

coeff_Y_on_X = np.cov(X, Y, ddof=0)[0, 1] / np.var(X)
coeff_X_on_Y = np.cov(X, Y, ddof=0)[1, 0] / np.var(Y)

print("Coefficient estimate for Y onto X:", coeff_Y_on_X)
print("Coefficient estimate for X onto Y:", coeff_X_on_Y)

plt.scatter(X, Y)
plt.plot(X, coeff_Y_on_X*X, color='red', label='Y onto X')
plt.plot(coeff_X_on_Y*Y, Y, color='blue', label='X onto Y')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.title('Same Coefficient Estimates')
plt.show()
```

```
Coefficient estimate for Y onto X: 0.9684925087655322
Coefficient estimate for X onto Y: 0.9400540128203652
```


Same Coefficient Estimates
Y onto X
X onto Y
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Y
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
X