CS 4342 Assignment #2

Ivan Martinovic

**Conceptual and Theoretical Questions**

1. Describe the null hypotheses to which the p-values given in the below Table correspond. Explain what conclusions you can draw based on these p-values. Your explanation should be phrased in terms of sales, TV, radio, and newspaper, rather than in terms of the coefficients of the linear model.



Intercept: Null hypothesis is that the Intercept = 0, i.e. when we don't invest any money in any kind of advertising, our sales are zero. The p-value is a measurement of the plausibility that the null hypothesis is true. Since the p-value is extremely small, i.e. there is a very small probability that the null hypothesis is true, we reject the null hypothesis, in favor of the alternative hypothesis: Intercept > 0 or Intercept < 0, i.e. the sales are non-zero when there is no advertising spending (they should be positive, because sales cannot be negative).

TV: Null hypothesis is that the TV coefficient = 0, i.e. investment in TV advertising has no effect on the total sales. Since p-value is again extremely small, we reject the null-hypothesis in favor of the alternative: TV coefficient > 0 or < 0; i.e. the sales are affected (either positively or negatively) by increased TV advertising.

radio: Null hypothesis is that the radio coefficient = 0, i.e. investment in radio advertising has no effect on the total sales. Since p-value is again extremely small, we reject the null-hypothesis in favor of the alternative: radio coefficient > 0 or < 0; i.e. the sales are affected (either positively or negatively) by increased radio advertising.

newspaper: Null hypothesis is that the newspaper coefficient = 0, i.e. investment in newspaper advertising has no effect on the total sales. In this case the p-value is very high, indicating that the plausibility of the null hypothesis is strong. Hence we do not reject the null-hypothesis.

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

The KNN classifier is used in classification problems, for predicting to which class the response variable belongs to. The KNN classifier determines the K nearest neighbors for given predictors, and then assigns the class to the response variable which is found in the highest number of the K selected neighbors. For example: our response variable is whether a fruit is an orange or a lemon, and the predictors are width, height and color. Say for a certain color, width and height, the KNN classifier (with K = 5) finds that, 2 neighbors are oranges and 3 are lemons. It will therefore assign the predicted value to the lemons class.

The KNN regression method is used in estimation problems, for predicting the value of the response variable. The KNN regression method determines the K nearest neighbors for given predictors, and then assigns the value to the response variable, which is an average of the response variables of all K selected neighbors. For example: our response variable is the length of a fruit, and our predictors are width, color and class (orange or lemon). Say for a lemon, with certain width and color, the KNN regression method (K=5) finds that the values of the response variables of the 5 closest neighbors are 5.0, 5.2, 5.4, 5.6, and 5.8. It will therefore predict that the length of the lemon is: avg(5.0, 5.2, 5.4, 5.6, 5.8) = 5.4 for the given predictors.

3. Suppose we have a data set with five predictors, X1 = GPA, X2 = IQ, X3 = Gender (1 for Female and 0 for Male), X4 = Interaction between GPA and IQ, and X5 = Interaction between GPA and Gender. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model, and get

(a) Which answer is correct, and why?

1. For a fixed value of IQ and GPA, males earn more on average than females.
2. For a fixed value of IQ and GPA, females earn more on average than males.
3. For a fixed value of IQ and GPA, males earn more on average than females provided that the GPA is high enough.
4. For a fixed value of IQ and GPA, females earn more on average than males provided that the GPA is high enough.

iii. is correct.

Write the equation out:

If two people have the same IQ and GPA, then terms X1, X2 and X4 can be replaced by a constant C, leaving only X3 and X5 as the variables which change the value of the salary. Now salary becomes:

Assume . If a person is male then and salary is just:

However, if the person if female then and salary is:

Here we can see that the salary for females is higher for GPA's under 3.5; and the salary for males is higher for GPA's over 3.5.

(b) Predict the salary of a female with IQ of 110 and a GPA of 4.0.

Assuming :

(c) True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is very little evidence of an interaction effect. Justify your answer.

To conclude whether there is little evidence of an interaction effect we need to calculate the p-value for the null hypothesis. In this case we do not have the data to do so.

I will try to provide a more simplistic approach in determining whether the interaction effect exists. Let’s see how much the value of the Salary is impacted as a percentage when we include or exclude X4.

The minimum value X4 can achieve is 0, and it occurs either when the IQ is 0 or the GPA is 0 or both. In either case adding the value will not affect the value of the salary, therefore the impact of X4 expressed as a percentage is 0.

The maximum value X4 can achieve is 800, and it occurs when the IQ is 200 (assuming this is the maximum one can achieve) and the GPA is 4.0. In this case . The Salary then becomes:

In both cases removing X4 will affect the Salary by 5.74% and 5.95% respectively. This is quite a noticeable bump (especially in someone's salary), and I would say that this is some evidence (although not so strong) of an interaction effect.

4. I collect a set of data (n = 100 observations) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e.

(a) Suppose that the true relationship between X and Y is linear, i.e. 𝑌 = 𝛽0 + 𝛽1 𝑋 + . Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

For a training data set (regardless of the true relationship), we expect the more flexible model to have a lower RSS, since a more flexible model can always fit training data better than an inflexible model. Since the cubic regression is more flexible we would expect its RSS to be smaller.

(b) Answer (a) using test rather than training RSS.

We expect RSS for test data to be lower for the model which better reflects the true relationship. Since in this case the true relationship is linear, we expect the linear regression model to have a lower RSS.

(c) Suppose that the true relationship between X and Y is not linear, but we don’t know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell? Justify your answer.

Just as in part (a), for a training data set (regardless of the true relationship), we expect the more flexible model to have a lower RSS, because a more flexible model can always fit training data better than an inflexible model. Since the cubic regression is more flexible we would expect its RSS to be smaller.

(d) Answer (c) using test rather than training RSS.

Just as in part (b), we expect RSS for test data to be lower for the model which better reflects the true relationship. Since now we do not know how far away a relationship is from being linear, we do not have enough information to say whether a linear or a cubic regression model would have lower RSS for test data.

5. Consider the fitted values that result from performing linear regression without an intercept. In this setting, the ith fitted value takes the form:

where:

Show that we can write

Using algebraic manipulations:

Where:

What is ?

Note: We interpret this result by saying that the fitted values from linear regression are linear combinations of the response variables.

is the projection of our xi on our normalized bases of our training predictor variables.

When we expand the equation for we get:

Which is a linear combination of the response variables, where coefficients are , …

6. Using equation (3.4) – shown below, argue that in the case of simple linear regression, the least squares line always passes through the point (𝑥̅, 𝑦̅).

(3.4.a)

(3.5.b)

In simple linear regression:

We have to show that when then

Suppose:

Substitute for :

Now substitute for :

With some simple algebraic manipulations we arrive at:

q.e.d.

**Applied Questions**

1. This question involves the use of simple linear regression on the Auto data set.

(a) Perform a simple linear regression with mpg as the response and horsepower as the predictor and answer the following questions:



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

Yes, there seems to a relationship between mpg and horsepower. The p value is very small, indicating that the plausibility of the null hypothesis is very low i.e. we reject the null hypothesis that there is no relationship between mpg and horsepower.

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

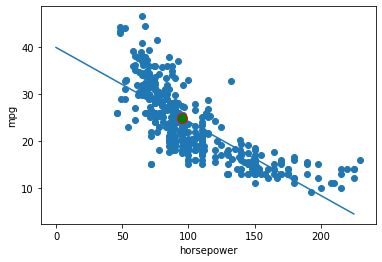
The absolute value of the relationship is 0.1578

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

The relationship is negative (slope is negative). The mpg decreases as horsepower increases.

1. What is the predicted mpg associated with a horsepower of 95?

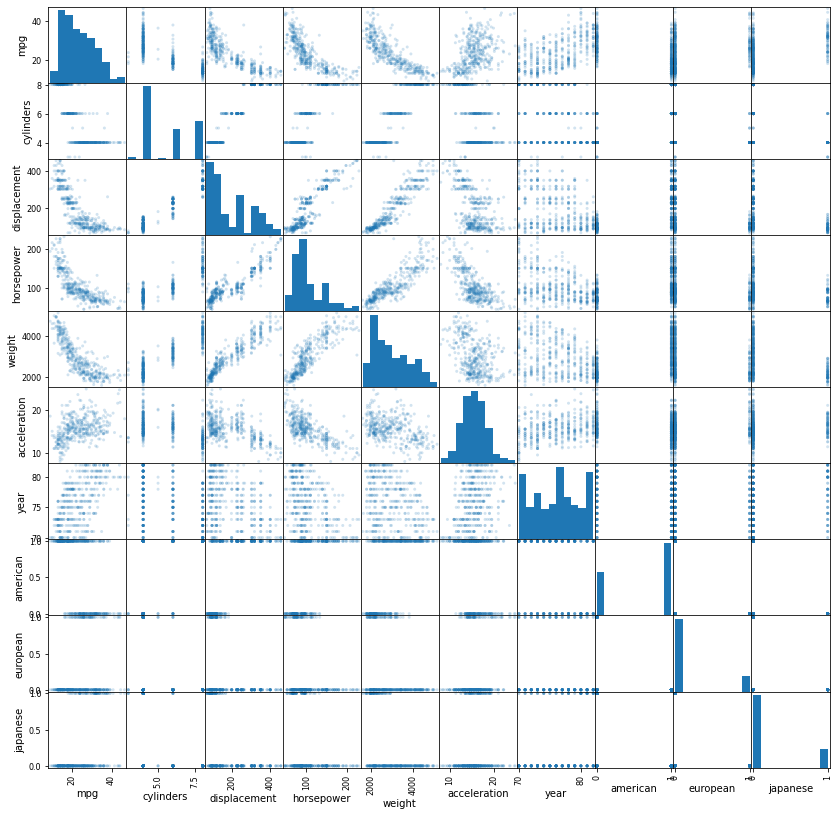
The predicted mpg is 24.94061135

(b) Plot the response and the predictor along with the predicted line.

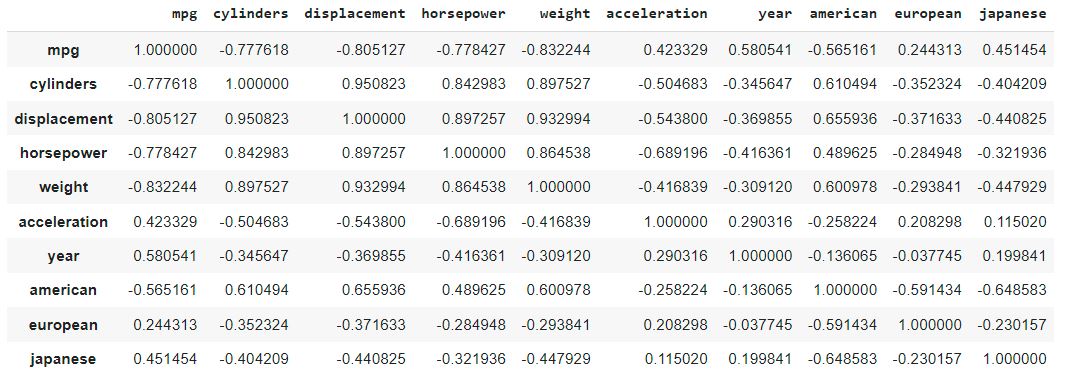
2. This question involves the use of multiple linear regression on the Auto data set.

I created 3 dummy variables for origin: 'american' which is 1 when origin = 1, 0 otherwise; 'european' which is 1 when origin = 2, 0 otherwise; and 'japanese' which is 1 when origin = 3, 0 otherwise.

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

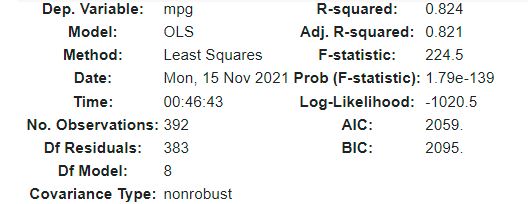


(b) Compute the matrix of correlations between the variables. You will need to exclude the name variable which is qualitative.



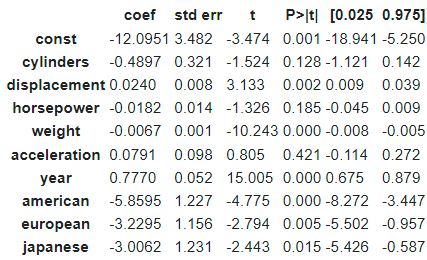
(c) Perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Examine the results, and comment on the output. For instance:

1. Is there a relationship between the predictors and the response?



The F-statistic is large and the p-value for the F-statistic is small, meaning that we reject the null hypothesis that all coefficients are 0. We there say that there exists a relationship between some of the predictors and the response.

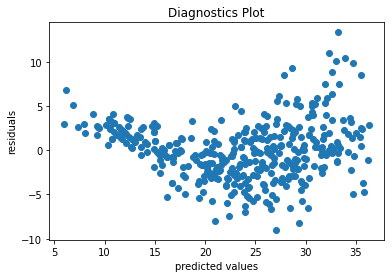
1. Which predictors appear to have a statistically significant relationship to the response?

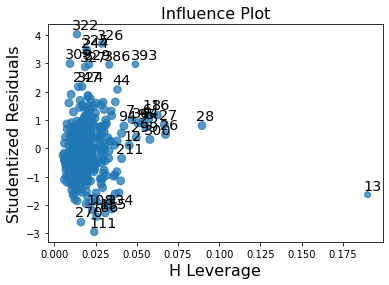


The predictors with a statistically significant (p-value < 0.01) relationship to the response are constant term, displacement, weight, year and the three dummy vairables. In other words, for these predictors we reject the null hypothesis.

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

Since the coefficient is positive, it suggests that newer cars have a better mpg ratio, meaning that they are more fuel efficient.

(d) Produce diagnostic plots of the linear regression fit. 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?



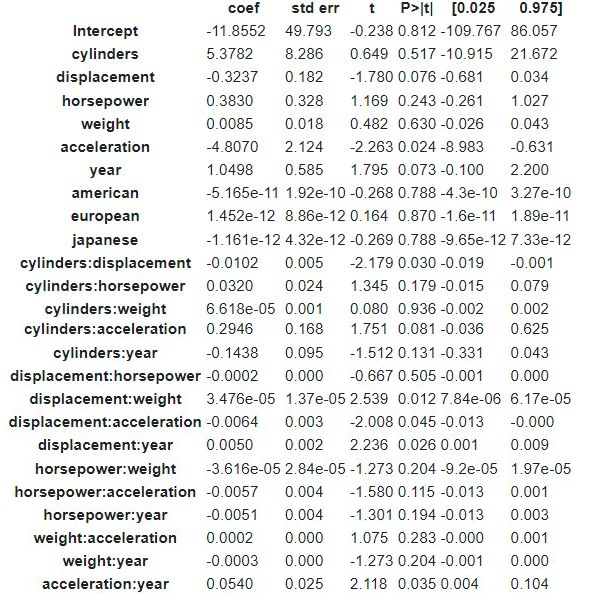
The residuals exhibit a clear U-shape, which provides a strong indication of non-linearity in the data. The variance in the error terms also seems to increase as the predicted value increases.

From the influence plot, we can see that quite a few points have a studentized residual over 3 in absolute value, which makes them outliers, with largest one being data point 322.

Two observations seem to stand out as having large leverage. Namely point 28 and point 13.

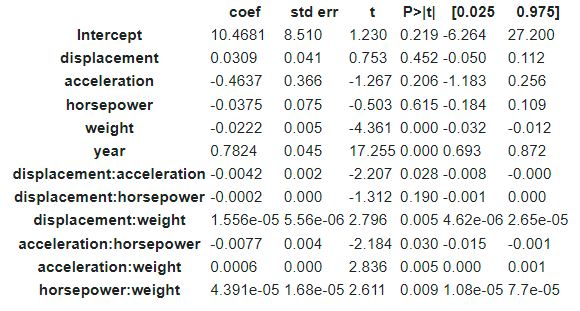
(e) i) Fit linear regression models with predictors and interaction terms. Do any interactions appear to be statistically significant? NOTE: there are two part e's for this question!!!

Initially I tried fitting as many predictors and interactions to obtain the following model:



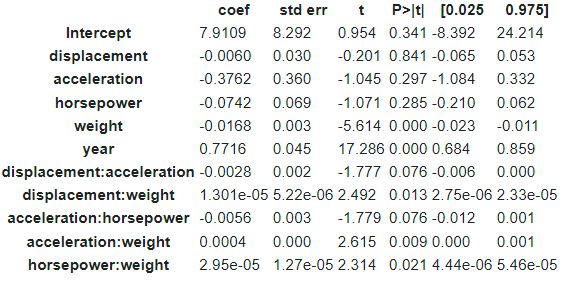
Assuming a significance level of 0.05, we have that acceleration, cylinders\*displacement, displacement\*acceleration, displacement\*year and acceleration\*year are of statistical significance.

After thinking about the problem for a bit, to me it would make sense that displacement, acceleration, horsepower and year as well as their interactions (excluding those with year) would have the greatest statistical significance. Here are the results of fitting such a model:



Here we see (assuming statistical significance of 0.05) that weight, year, and all interactions except displacement\*horsepower.

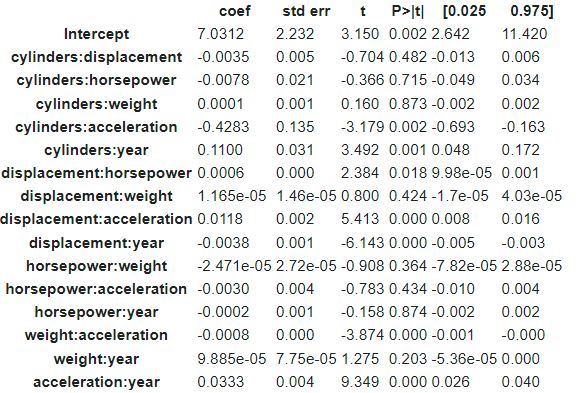
Going even further I removed this singular interaction with no statistical significance to obtain my final model:



Here the end result is that the same predictors are of statistical significance, however interaction between displacement and acceleration and that of acceleration and horsepower lost its statistical significance.

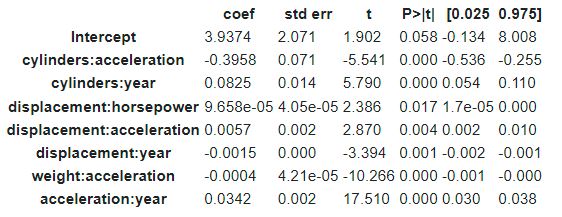
e) ii) Fit linear regression models with only interaction terms. Do any interactions appear to be statistically significant?

Initially I tried fitting as many interactions as possible to obtain the following model:



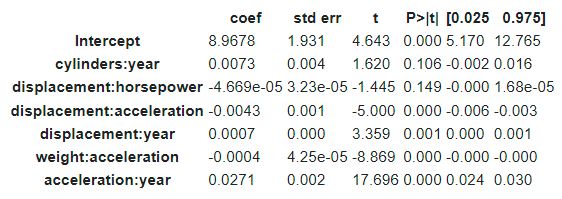
Assuming again a significance level of 0.05, we have that the intercept, cylinders\*acceleration, cylinders\*year, displacement\*horsepower, displacement\*acceleration, displacement\*year, weight\*acceleration and acceleration\*year are of statistical significance.

Going further I removed all interactions with no statistical significance to obtain the following model:



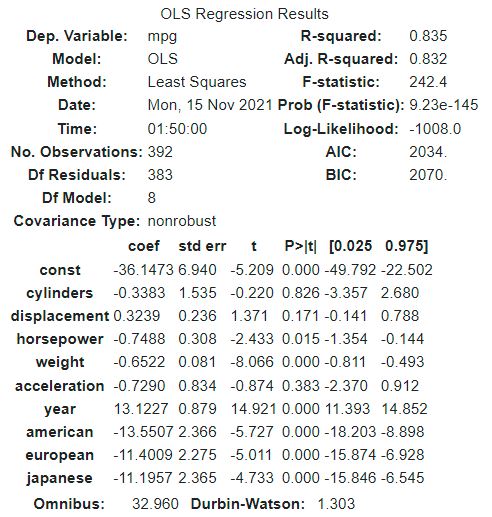
Here we see (assuming statistical significance of 0.05) that only the interaction between cylinders and acceleration is not of statistical significance.

For the final model I have therefore removed this interaction to obtain:



Here the end result is that all interactions, except those between cylinders and year and displacement and horsepower are of statistical significance.

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



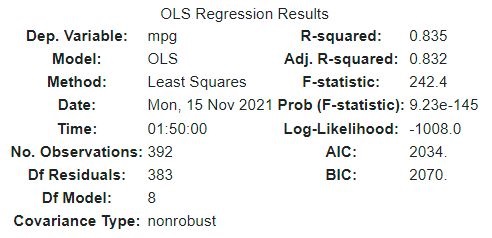
X2:

From the top table we see that R-squared is decently high at 0.803, meaning that our regression explains a fair amount of variance in the data.

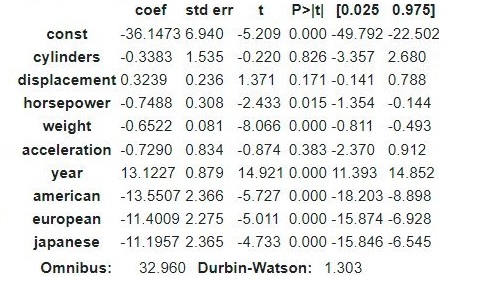
The F-statistic is high and the probability (p-value) for that F-statistic is low which indicates that the null hypothesis (i.e. that there is no relationship) is rejected. In other words there is a relationship between at least one predictor and the response.

From the bottom table we see that (assuming a significance level of 0.05), all predictors are of statistical significance except for the constant term and the horsepower squared.

:

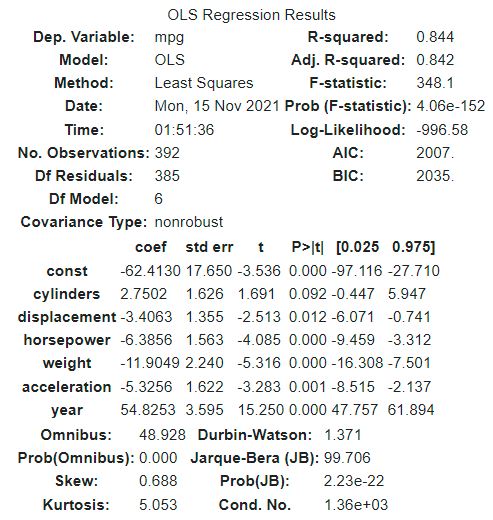
From the top table we see that the R-squared is even higher than for X2, implying that this model explains even more of the variance in the data.

The F-statistic is high and the probability (p-value) for that F-statistic is low which indicates that the null hypothesis is rejected. In other words there is a relationship between at least one predictor and the response.

From the bottom table we see that (assuming a significance level of 0.05), all predictors are of statistical significance, except sqrt(cylinders), sqrt(displacement) and sqrt(acceleration).

log(X):

Note: dummy variables had to be excluded (log(0) = NaN)

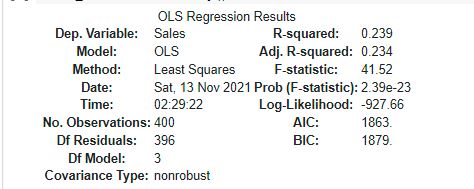
From the top table we see that the R-squared is highest of all models, implying that this model explains the variance in the data the best.

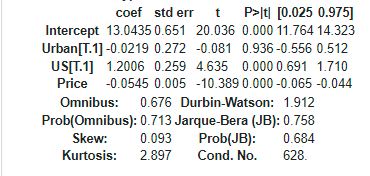
The F-statistic is high and the probability (p-value) for that F-statistic is low which indicates that the null hypothesis is rejected. In other words there is a relationship between at least one predictor and the response.

From the bottom table we see that (assuming a significance level of 0.05), all predictors are of statistical significance, except for the number of cylinders.

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

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





(b) Provide an interpretation of each coefficient in the model. Be careful—some of the variables in the model are qualitative!

The intercept coefficient is 13.0435 implying that when a car seat price is 0, the car seat is not urban or American it the sales are 13.0435.

The Urban coefficient is -0.0219, which implies that when a car seat is urban its sales are 0.0219 units lower.

The US coefficient is 1.2006, which implies that when a car seat is from the US its sales are 1.2006 units higher.

The price coefficient is -0.0545, which implies that the car seat's sales decrease by 0.0545 units for every unit increase in price.

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

Where is the intercept term, is the price and is its corresponding constant; is a dummy variable indicating whether a car seat is urban (1 if it is, 0 if not) price and is its corresponding constant; is a dummy variable indicating whether a car seat is from the US (1 if it is, 0 if not) price and is its corresponding constant.

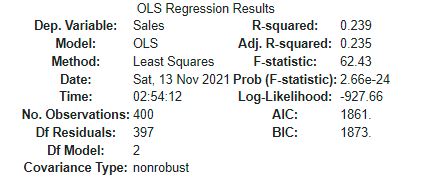
(d) For which of the predictors can you reject the null hypothesis H0 : βj = 0?

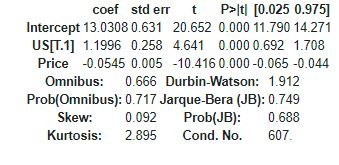
The only predictor with non-zero and quite high p-value is the Urban predictor. This is the only predictor for which we should not reject the null hypothesis.

For all other predictors we may reject the null hypothesis.

(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.

Here we remove the Urban predictor. Here is the outcome:





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

The R2 both models is 0.239 which implies that neither model explains the variance in the data very well.

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

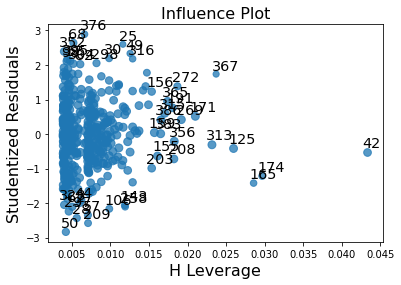
These are given above in the last column:

Intercept: (11.790, 14.271)

US: (0.692, 1.708)

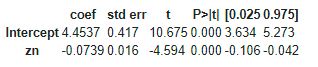
Price: (-0.065, -0.044)

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



From the influence plot shown, we have no data points with studentized residuals greater than 3 in absolute value (meaning there are not outliers). However we see quite a few points with high leverage. Say our cutoff for high leverage is a leverage statistic above 0.025. This then implies that the high leverage observations are 313, 367, 125, 165, 174 and 42

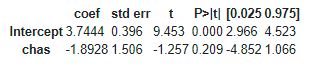
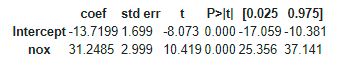
4. This problem involves the Boston data set, which we saw in the previous HW. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

(a) For each predictor, fit a simple linear regression model to predict the response. Describe your results. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

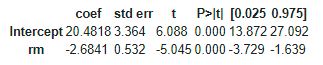
p-value<0.01 -> zn is statistically significant



p-value<0.01 -> indus is statistically significant

p-value>0.01 -> chas is not statistically significant

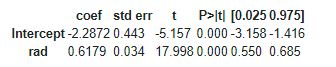
p-value<0.01 -> nox is statistically significant

p-value<0.01 -> rm is statistically significant

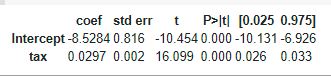


p-value<0.01 -> age is statistically significant

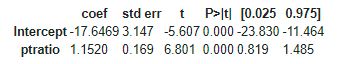
p-value < 0.01 -> dis is statistically significant



p-value < 0.01 -> rad is statistically significant



p-value < 0.01 -> tax is statistically significant

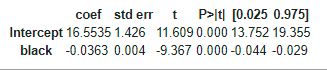


p-value < 0.01 -> ptratio is statistically significant

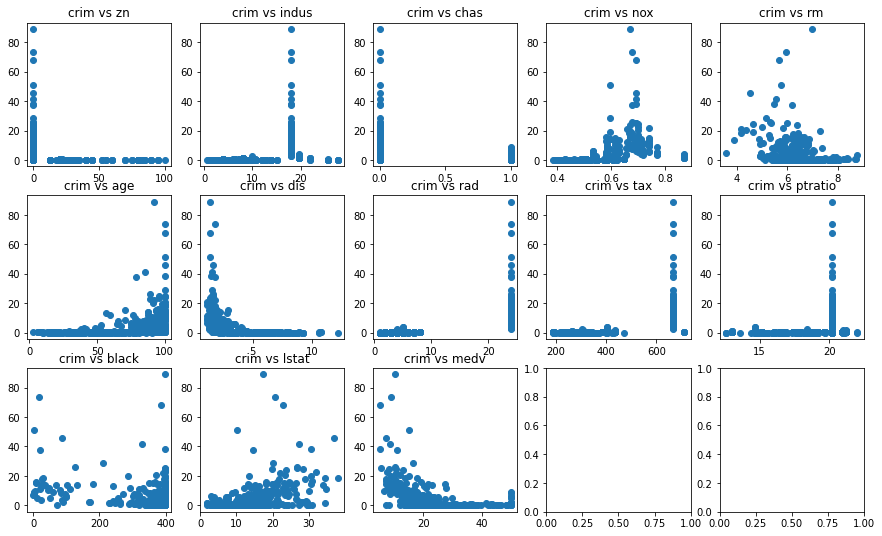


p-value <0.01 -> medv is statistically significant

p-value < 0.01 -> lstat is statistically significant

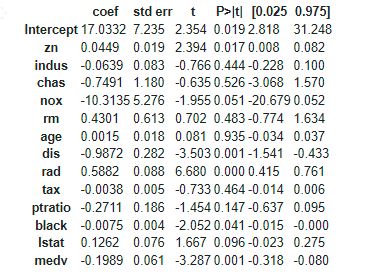


p-value < 0.01 -> black is statistically significant

For all predictors which we deemed statistically significant, we see that as the x-value goes from zero to non-zero we see a change in the response value.

For the chas predictor which we deemed statistically insignificant, when x is zero, the value for the predictor takes on a wide range of values, however when it takes the value of 1 the response goes to 0, which implies that the chas coefficient is likely zero.

(b) Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis H0 : βj = 0?

Rejecting null hypothesis for p-values under 0.01:

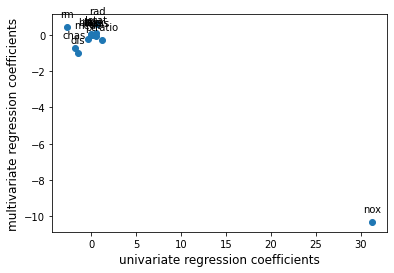
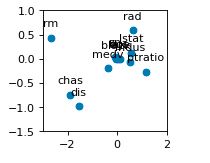
dis -> reject null

rad -> reject null

medv ->reject null

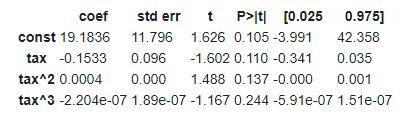
This implies that the only statistically significant predictors are dis, rad and medv.

(c) How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.



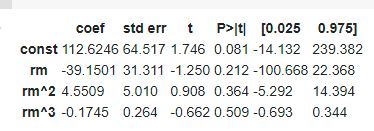
We can see that the univariate regression predicts a large positive coefficient for nox, but the multivariate regression predicts a large negative coefficient. On the right I have zoomed in on the remaining coefficients. rm, ptratio and indus seem to be the only coefficients for which the univariate and multivariate regressions disagree in sign. For the remaining predictors the values are about the same for both regressions. Exception to these are chas and dis, for which the univariate regression predicts a stronger relationship than the multivariate relationship.

(d) Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form

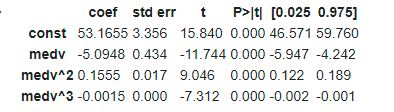


For tax p-value for all coefficients is greater than 0.01, implying that there is no evidence of non-linear association.

For rm p-value for all coefficients is greater than 0.01, implying that there is no evidence of non-linear association.

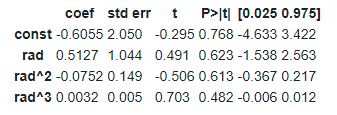


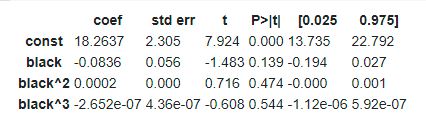
For rad p-value for all coefficients is greater than 0.01, implying that there is no evidence of non-linear association.



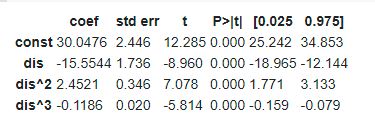
For medv p-value for all coefficients is less than 0.01, implying that there is evidence of non-linear association.



For lstat p-value for all coefficients is greater than 0.01, implying that there is no evidence of non-linear association.

For black p-value is less than 0.01 only for the constant coefficient, implying that there is no evidence of non-linear association.

For dis p-value for all coefficients is less than 0.01, implying that there is evidence of non-linear association.



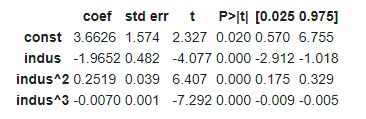
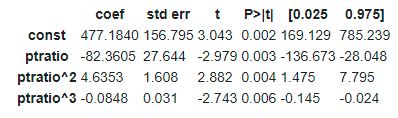
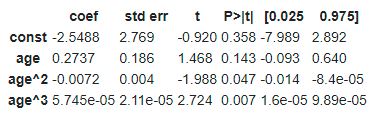
For nox p-value for all coefficients is less than 0.01, implying that there is evidence of non-linear association.

For age p-value for the cubic coefficient is less than 0.01, implying that there is evidence of non-linear association.

For indus p-value for all coefficients is less than 0.01, implying that there is evidence of non-linear association.

For ptratio p-value for all coefficients is less than 0.01, implying that there is evidence of non-linear association.

**Code Section**

Questions 1&2:

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

from matplotlib.pyplot import figure

import statsmodels.api as sm

from statsmodels.formula.api import ols

from statsmodels.stats.outliers\_influence import OLSInfluence

import statsmodels.formula.api as smf

!pip install pandas

import pandas as pd

from google.colab import data\_table

data\_table.disable\_dataframe\_formatter()

data = pd.read\_csv('Auto.csv', na\_values='?').dropna()

#Question1

'''

hp = pd.to\_numeric(data[data['horsepower'] != '?']['horsepower'])

mpg = pd.to\_numeric(data[data['horsepower'] != '?']['mpg'])

plt.scatter(hp, mpg)

hp = sm.add\_constant(hp)

model = sm.OLS(mpg, hp)

results = model.fit()

results.summary().tables[1]

print(results.params)

print("The predicted value of mpg for a 95 horsepower car is: ")

print(results.predict([1, 95]))

plt.plot([0, 225], [results.params[0],  results.params[0] + 225 \* results.params[1]])

plt.plot(95, results.predict([1, 95]), marker="o", markersize = 10, markeredgecolor="red", markerfacecolor="green")

plt.xlabel('horsepower')

plt.ylabel('mpg')

'''

#Question2

data = data.loc[:, data.columns!='name']

data['american'] = 0

data['european'] = 0

data['japanese'] = 0

data.loc[(data.origin == 1), 'american']= 1

data.loc[(data.origin == 2), 'european']= 1

data.loc[(data.origin == 3), 'japanese']= 1

data =  data.loc[:, data.columns!='origin']

'''

#a)

pd.plotting.scatter\_matrix(data, alpha=0.2, figsize=(14,14))

#b)

corrM = data.corr()

corrM

#c

predictors = data.loc[:, data.columns!='mpg']

predictors['american'] = 0

predictors['european'] = 0

predictors['japanese'] = 0

predictors.loc[(predictors.origin == 1), 'american']= 1

predictors.loc[(predictors.origin == 2), 'european']= 1

predictors.loc[(predictors.origin == 3), 'japanese']= 1

predictors =  predictors.loc[:, predictors.columns!='origin']

response = data['mpg']

predictors = sm.add\_constant(predictors)

model = sm.OLS(response, predictors.astype(float))

results = model.fit()

print(results.params)

results.summary()

#d)

predictions = results.predict(predictors)

residuals = response - predictions;

#residual plot

plt.title('Diagnostics Plot')

plt.xlabel('predicted values')

plt.ylabel('residuals')

plt.scatter(predictions, residuals)

figure(figsize=(16, 16), dpi=80)

sm.graphics.influence\_plot(results, size=6)

#e1) plot interactions

model\_all\_interactions = smf.ols(formula='mpg ~ cylinders + displacement + horsepower + weight + acceleration + year + american + european + japanese + cylinders:displacement + cylinders:horsepower + cylinders:weight + cylinders:acceleration + cylinders:year + displacement:horsepower + displacement:weight + displacement:acceleration + displacement:year + horsepower:weight + horsepower:acceleration + horsepower:year + weight:acceleration + weight:year + acceleration:year', data=data).fit()

model\_all\_interactions.summary().tables[1]

model\_some\_interactions\_1 = smf.ols(formula='mpg ~ displacement + acceleration + horsepower + weight + year + displacement:acceleration + displacement:horsepower + displacement:weight + acceleration:horsepower + acceleration:weight + horsepower:weight', data=data).fit()

model\_some\_interactions\_1.summary().tables[1]

model\_some\_interactions\_2 = smf.ols(formula='mpg ~ displacement + acceleration + horsepower + weight + year + displacement:acceleration + displacement:weight + acceleration:horsepower + acceleration:weight + horsepower:weight', data=data).fit()

model\_some\_interactions\_2.summary().tables[1]

'''

#e2) plot interactions

model\_all\_interactions = smf.ols(formula='mpg ~ cylinders:displacement + cylinders:horsepower + cylinders:weight + cylinders:acceleration + cylinders:year + displacement:horsepower + displacement:weight + displacement:acceleration + displacement:year + horsepower:weight + horsepower:acceleration + horsepower:year + weight:acceleration + weight:year + acceleration:year', data=data).fit()

model\_all\_interactions.summary().tables[1]

model\_some\_interactions\_1 = smf.ols(formula='mpg ~ cylinders:acceleration + cylinders:year + displacement:horsepower + displacement:acceleration + displacement:year + weight:acceleration + acceleration:year', data=data).fit()

model\_some\_interactions\_1.summary().tables[1]

model\_some\_interactions\_2 = smf.ols(formula='mpg ~ cylinders:year + displacement:horsepower + displacement:acceleration + displacement:year + weight:acceleration + acceleration:year', data=data).fit()

model\_some\_interactions\_2.summary().tables[1]

'''

#f)

predictors =  data.loc[:, data.columns!='mpg']

response = data['mpg']

predictorsSquared = sm.add\_constant(np.power(predictors.astype(float), 2))

predictorsSquareRoot = sm.add\_constant(np.power(predictors.astype(float), 0.5))

resultsSquared = sm.OLS(response, predictorsSquared).fit()

resultsSquareRoot = sm.OLS(response, predictorsSquareRoot).fit()

predictors =  predictors.loc[:, predictors.columns!='american']

predictors =  predictors.loc[:, predictors.columns!='european']

predictors =  predictors.loc[:, predictors.columns!='japanese']

predictorsLog = sm.add\_constant(np.log(predictors.astype(float)))

resultsLog = sm.OLS(response, predictorsLog).fit()

resultsSquared.summary()

resultsSquareRoot.summary()

resultsLog.summary()

'''

plt.show()

Question 3

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

from matplotlib.pyplot import figure

import statsmodels.api as sm

from statsmodels.formula.api import ols

from statsmodels.stats.outliers\_influence import OLSInfluence

import statsmodels.formula.api as smf

!pip install pandas

import pandas as pd

from google.colab import data\_table

data\_table.disable\_dataframe\_formatter()

data = pd.read\_csv('Carseats.csv', na\_values='?').dropna()

#Question3

#a)

data.loc[(data.Urban == 'Yes'),'Urban']= 1

data.loc[(data.Urban == 'No'),'Urban']= 0

data.loc[(data.US == 'Yes'),'US']= 1

data.loc[(data.US == 'No'),'US']= 0

model = smf.ols(formula='Sales ~ Price + Urban + US', data=data).fit()

#e)

model\_noUrban = smf.ols(formula='Sales ~ Price + US', data=data).fit()

#g)

#h)

sm.graphics.influence\_plot(model\_noUrban, size=6)

plt.show()

Question 4

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

from matplotlib.pyplot import figure

import statsmodels.api as sm

from statsmodels.formula.api import ols

from statsmodels.stats.outliers\_influence import OLSInfluence

import statsmodels.formula.api as smf

!pip install pandas

import pandas as pd

from google.colab import data\_table

data\_table.disable\_dataframe\_formatter()

data = pd.read\_csv('Boston.csv', na\_values='?').dropna()

crim = data['crim']

predictors =  data.loc[:, data.columns!='crim']

zn = data['zn']

indus = data['indus']

chas = data['chas']

nox = data['nox']

rm = data['rm']

age = data['age']

dis = data['dis']

rad = data['rad']

tax = data['tax']

ptratio = data['ptratio']

black = data['black']

lstat = data['lstat']

medv = data['medv']

#a)

model\_zn = smf.ols(formula='crim ~ zn', data=data).fit()

model\_indus = smf.ols(formula='crim ~ indus', data=data).fit()

model\_chas = smf.ols(formula='crim ~ chas', data=data).fit()

model\_nox = smf.ols(formula='crim ~ nox', data=data).fit()

model\_rm = smf.ols(formula='crim ~ rm', data=data).fit()

model\_age = smf.ols(formula='crim ~ age', data=data).fit()

model\_dis = smf.ols(formula='crim ~ dis', data=data).fit()

model\_rad = smf.ols(formula='crim ~ rad', data=data).fit()

model\_tax = smf.ols(formula='crim ~ tax', data=data).fit()

model\_ptratio = smf.ols(formula='crim ~ ptratio', data=data).fit()

model\_black = smf.ols(formula='crim ~ black', data=data).fit()

model\_lstat = smf.ols(formula='crim ~ lstat', data=data).fit()

model\_medv = smf.ols(formula='crim ~ medv', data=data).fit()

model\_zn.summary().tables[1]

model\_indus.summary().tables[1]

model\_chas.summary().tables[1]

model\_nox.summary().tables[1]

model\_rm.summary().tables[1]

model\_age.summary().tables[1]

model\_dis.summary().tables[1]

model\_rad.summary().tables[1]

model\_tax.summary().tables[1]

model\_ptratio.summary().tables[1]

model\_black.summary().tables[1]

model\_lstat.summary().tables[1]

model\_medv.summary().tables[1]

fig, axs = plt.subplots(3, 5)

axs[0, 0].set\_title('crim vs zn')

axs[0, 0].scatter(zn, crim)

axs[0, 1].set\_title('crim vs indus')

axs[0, 1].scatter(indus, crim)

axs[0, 2].set\_title('crim vs chas')

axs[0, 2].scatter(chas, crim)

axs[0, 3].set\_title('crim vs nox')

axs[0, 3].scatter(nox, crim)

axs[0, 4].set\_title('crim vs rm')

axs[0, 4].scatter(rm, crim)

axs[1, 0].set\_title('crim vs age')

axs[1, 0].scatter(age, crim)

axs[1, 1].set\_title('crim vs dis')

axs[1, 1].scatter(dis, crim)

axs[1, 2].set\_title('crim vs rad')

axs[1, 2].scatter(rad, crim)

axs[1, 3].set\_title('crim vs tax')

axs[1, 3].scatter(tax, crim)

axs[1, 4].set\_title('crim vs ptratio')

axs[1, 4].scatter(ptratio, crim)

axs[2, 0].set\_title('crim vs black')

axs[2, 0].scatter(black, crim)

axs[2, 1].set\_title('crim vs lstat')

axs[2, 1].scatter(lstat, crim)

axs[2, 2].set\_title('crim vs medv')

axs[2, 2].scatter(medv, crim)

fig.set\_figheight(9)

fig.set\_figwidth(15)

'''

#b)

model = smf.ols(formula='crim ~ zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + black + lstat + medv', data=data).fit()

model.summary().tables[1]

#c)

X = [model\_zn.params[1], model\_indus.params[1], model\_chas.params[1], model\_nox.params[1], model\_rm.params[1], model\_age.params[1], model\_dis.params[1], model\_rad.params[1], model\_tax.params[1], model\_ptratio.params[1], model\_black.params[1], model\_lstat.params[1], model\_medv.params[1]]

Y = [model.params[1], model.params[2], model.params[3], model.params[4], model.params[5], model.params[6], model.params[7], model.params[8], model.params[9], model.params[10], model.params[11], model.params[12], model.params[13]]

#figure(figsize=(2, 2), dpi=80)

#plt.xlim([-3, 2])

#plt.ylim([-1.5, 1])

plt.xlabel('univariate regression coefficients', fontsize=12)

plt.ylabel('multivariate regression coefficients', fontsize=12)

plt.scatter(X,Y)

plt.annotate('zn', (X[0],Y[0]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('indus', (X[1],Y[1]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('chas', (X[2],Y[2]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('nox', (X[3],Y[3]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('rm', (X[4],Y[4]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('age', (X[5],Y[5]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('dis', (X[6],Y[6]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('rad', (X[7],Y[7]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('tax', (X[8],Y[8]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('ptratio', (X[9],Y[9]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('black', (X[10],Y[10]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('lstat', (X[11],Y[11]),  textcoords="offset points",  xytext=(0,10), ha='center')

plt.annotate('medv', (X[12],Y[12]),  textcoords="offset points",  xytext=(0,10), ha='center')

#d)

zn\_cubic = pd.DataFrame(columns = ['zn', 'zn^2', 'zn^3'])

zn\_cubic['zn'] = zn

zn\_cubic['zn^2'] = zn\*zn

zn\_cubic['zn^3'] = zn\*zn\*zn

zn\_cubic = sm.add\_constant(zn\_cubic)

model\_zn\_cube = sm.OLS(crim, zn\_cubic).fit()

indus\_cubic = pd.DataFrame(columns = ['indus', 'indus^2', 'indus^3'])

indus\_cubic['indus'] = indus

indus\_cubic['indus^2'] = indus\*indus

indus\_cubic['indus^3'] = indus\*indus\*indus

indus\_cubic = sm.add\_constant(indus\_cubic)

model\_indus\_cube = sm.OLS(crim, indus\_cubic).fit()

chas\_cubic = pd.DataFrame(columns = ['chas', 'chas^2', 'chas^3'])

chas\_cubic['chas'] = chas

chas\_cubic['chas^2'] = chas\*chas

chas\_cubic['chas^3'] = chas\*chas\*chas

chas\_cubic = sm.add\_constant(chas\_cubic)

model\_chas\_cube = sm.OLS(crim, chas\_cubic).fit()

model\_chas\_cube.summary().tables[1]

nox\_cubic = pd.DataFrame(columns = ['nox', 'nox^2', 'nox^3'])

nox\_cubic['nox'] = nox

nox\_cubic['nox^2'] = nox\*nox

nox\_cubic['nox^3'] = nox\*nox\*nox

nox\_cubic = sm.add\_constant(nox\_cubic)

model\_nox\_cube = sm.OLS(crim, nox\_cubic).fit()

rm\_cubic = pd.DataFrame(columns = ['rm', 'rm^2', 'rm^3'])

rm\_cubic['rm'] = rm

rm\_cubic['rm^2'] = rm\*rm

rm\_cubic['rm^3'] = rm\*rm\*rm

rm\_cubic = sm.add\_constant(rm\_cubic)

model\_rm\_cube = sm.OLS(crim, rm\_cubic).fit()

age\_cubic = pd.DataFrame(columns = ['age', 'age^2', 'age^3'])

age\_cubic['age'] = age

age\_cubic['age^2'] = age\*age

age\_cubic['age^3'] = age\*age\*age

age\_cubic = sm.add\_constant(age\_cubic)

model\_age\_cube = sm.OLS(crim, age\_cubic).fit()

dis\_cubic = pd.DataFrame(columns = ['dis', 'dis^2', 'dis^3'])

dis\_cubic['dis'] = dis

dis\_cubic['dis^2'] = dis\*dis

dis\_cubic['dis^3'] = dis\*dis\*dis

dis\_cubic = sm.add\_constant(dis\_cubic)

model\_dis\_cube = sm.OLS(crim, dis\_cubic).fit()

rad\_cubic = pd.DataFrame(columns = ['rad', 'rad^2', 'rad^3'])

rad\_cubic['rad'] = rad

rad\_cubic['rad^2'] = rad\*rad

rad\_cubic['rad^3'] = rad\*rad\*rad

rad\_cubic = sm.add\_constant(rad\_cubic)

model\_rad\_cube = sm.OLS(crim, rad\_cubic).fit()

tax\_cubic = pd.DataFrame(columns = ['tax', 'tax^2', 'tax^3'])

tax\_cubic['tax'] = tax

tax\_cubic['tax^2'] = tax\*tax

tax\_cubic['tax^3'] = tax\*tax\*tax

tax\_cubic = sm.add\_constant(tax\_cubic)

model\_tax\_cube = sm.OLS(crim, tax\_cubic).fit()

ptratio\_cubic = pd.DataFrame(columns = ['ptratio', 'ptratio^2', 'ptratio^3'])

ptratio\_cubic['ptratio'] = ptratio

ptratio\_cubic['ptratio^2'] = ptratio\*ptratio

ptratio\_cubic['ptratio^3'] = ptratio\*ptratio\*ptratio

ptratio\_cubic = sm.add\_constant(ptratio\_cubic)

model\_ptratio\_cube = sm.OLS(crim, ptratio\_cubic).fit()

black\_cubic = pd.DataFrame(columns = ['black', 'black^2', 'black^3'])

black\_cubic['black'] = black

black\_cubic['black^2'] = black\*black

black\_cubic['black^3'] = black\*black\*black

black\_cubic = sm.add\_constant(black\_cubic)

model\_black\_cube = sm.OLS(crim, black\_cubic).fit()

lstat\_cubic = pd.DataFrame(columns = ['lstat', 'lstat^2', 'lstat^3'])

lstat\_cubic['lstat'] = lstat

lstat\_cubic['lstat^2'] = lstat\*lstat

lstat\_cubic['lstat^3'] = lstat\*lstat\*lstat

lstat\_cubic = sm.add\_constant(lstat\_cubic)

model\_lstat\_cube = sm.OLS(crim, lstat\_cubic).fit()

medv\_cubic = pd.DataFrame(columns = ['medv', 'medv^2', 'medv^3'])

medv\_cubic['medv'] = medv

medv\_cubic['medv^2'] = medv\*medv

medv\_cubic['medv^3'] = medv\*medv\*medv

medv\_cubic = sm.add\_constant(medv\_cubic)

model\_medv\_cube = sm.OLS(crim, medv\_cubic).fit()

model\_zn\_cube.summary().tables[1]

model\_indus\_cube.summary().tables[1]

model\_nox\_cube.summary().tables[1]

model\_rm\_cube.summary().tables[1]

model\_age\_cube.summary().tables[1]

model\_dis\_cube.summary().tables[1]

model\_rad\_cube.summary().tables[1]

model\_tax\_cube.summary().tables[1]

model\_ptratio\_cube.summary().tables[1]

model\_black\_cube.summary().tables[1]

model\_lstat\_cube.summary().tables[1]

model\_medv\_cube.summary().tables[1]

'''

plt.show()