X=bos.PTRATIO.values; y=bos.PRICE.values;

# add intercept

X = sm.add\_constant(X)

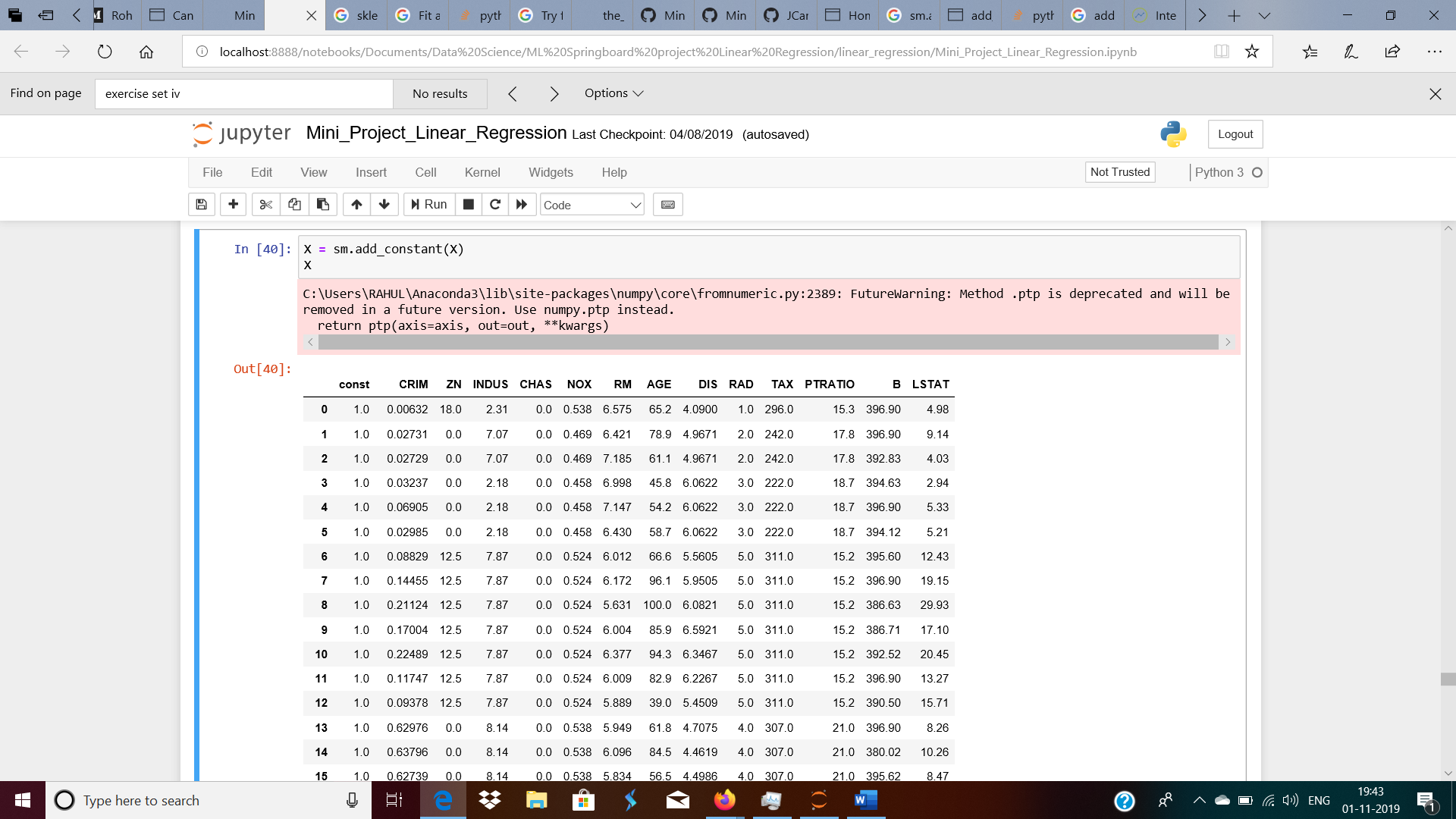
# fit model

model = sm.OLS(y, X); results = model.fit()

results.summary2()

The add\_constant() function adds an array of one. The original array with a constant (column of ones) as the first or last column.

An intercept is not included by default and should be added by the user



Interpreting the Intercept in a Regression Model

*by* Karen Grace-Martin

The intercept (often labeled the constant) is the expected mean value of Y when all X=0.

Start with a regression equation with one predictor, X.

If X sometimes equals 0, the intercept is simply the expected mean value of Y at that value.

If X never equals 0, then the intercept has no intrinsic meaning. In scientific research, the purpose of a regression model is to understand the relationship between predictors and the response.  If so, and if X never = 0, there is no interest in the intercept. It doesn’t tell you anything about the relationship between X and Y.

You do need it to calculate predicted values, though.  In market research, there is usually more interest in prediction, so the intercept is more important here.

When X never equals 0 is one reason for centering X. If you re-scale X so that the mean or some other meaningful value = 0 (just subtract a constant from X), now the intercept has a meaning. It’s the mean value of Y at the chosen value of X.

If you have dummy variables in  your model, though, the intercept has more meaning.  Dummy coded variables have values of 0 for the reference group and 1 for the comparison group. Since the intercept is the expected mean value when X=0, it is the mean value only for the reference group (when all other X=0).

This is especially important to consider when the dummy coded predictor is included in an interaction term.  Say for example that X1 is a continuous variable centered at its mean.  X2 is a dummy coded predictor, and the model contains an interaction term for X1\*X2.

The B value for the intercept is the mean value of X1 only for the reference group.  The mean value of X1 for the comparison group is the intercept plus the coefficient for X2.

It’s hard to give an example because it really depends on how X1 and X2 are coded. So I put together 6 situations in this follow up:

### [**How to Interpret the Intercept in 6 Linear Regression Examples**](https://www.theanalysisfactor.com/interpret-the-intercept/)

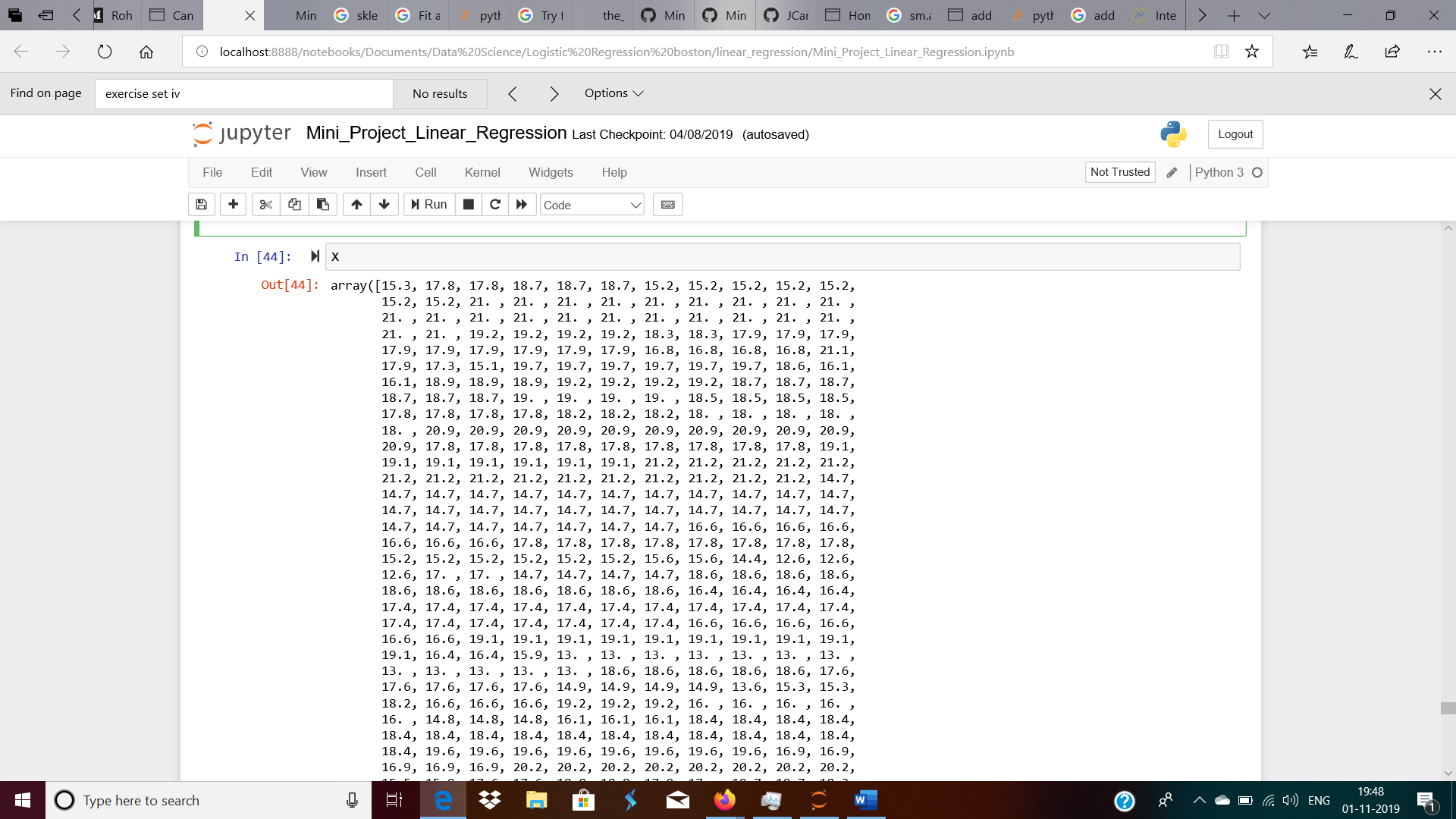
On the other hand if we just include the value of columns.

X=bos.PTRATIO.values; y=bos.PRICE.values;

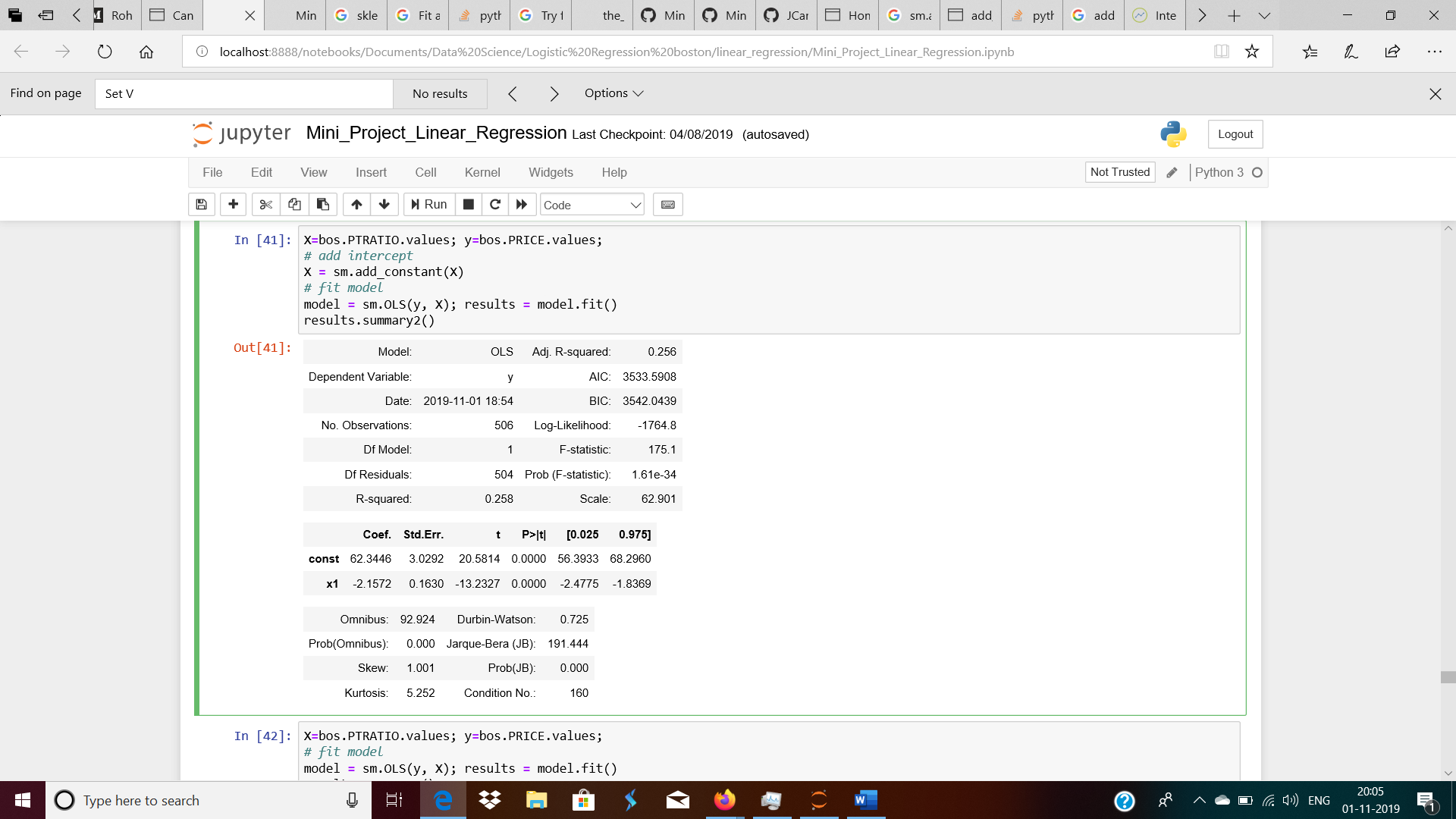
# fit model

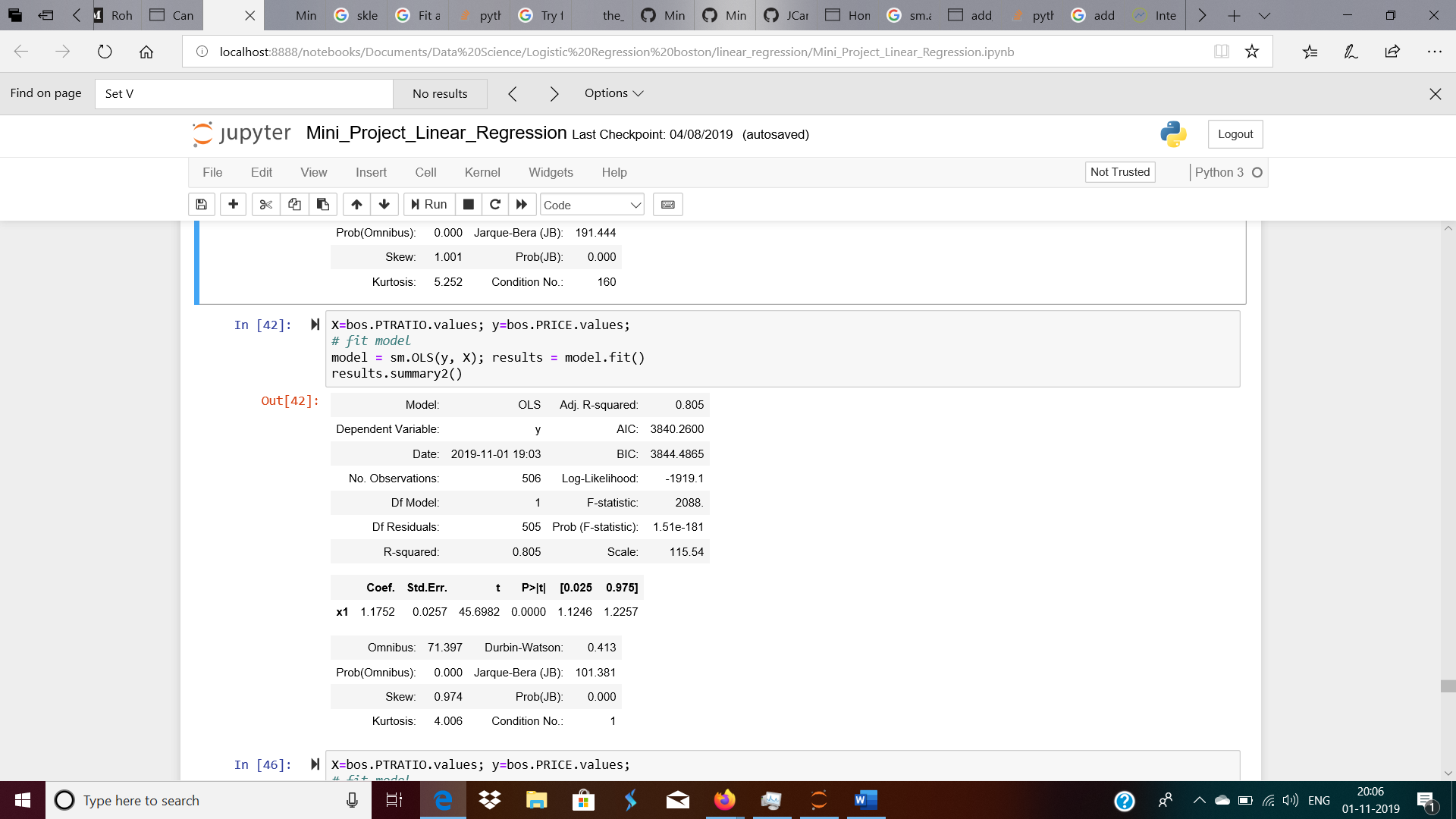
model = sm.OLS(y, X); results = model.fit()

results.summary2()



Even the results are different.





## Part 4: Comparing Models[¶](https://render.githubusercontent.com/view/ipynb?commit=627e349a1c650805ebce426ba14560e89cdecf60&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f676973742f4a43617264656e617352647a2f35346131353165613663303638643661346536663764343564636437623337382f7261772f363237653334396131633635303830356562636534323662613134353630653839636465636636302f4d696e695f50726f6a6563745f4c696e6561725f52656772657373696f6e5f536f6c7574696f6e2e6970796e62&nwo=JCardenasRdz%2F54a151ea6c068d6a4e6f7d45dcd7b378&path=Mini_Project_Linear_Regression_Solution.ipynb&repository_id=44921486&repository_type=Gist#Part-4:-Comparing-Models)

During modeling, there will be times when we want to compare models to see which one is more predictive or fits the data better. There are many ways to compare models, but we will focus on two.

### The -Statistic Revisited[¶](https://render.githubusercontent.com/view/ipynb?commit=627e349a1c650805ebce426ba14560e89cdecf60&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f676973742f4a43617264656e617352647a2f35346131353165613663303638643661346536663764343564636437623337382f7261772f363237653334396131633635303830356562636534323662613134353630653839636465636636302f4d696e695f50726f6a6563745f4c696e6561725f52656772657373696f6e5f536f6c7574696f6e2e6970796e62&nwo=JCardenasRdz%2F54a151ea6c068d6a4e6f7d45dcd7b378&path=Mini_Project_Linear_Regression_Solution.ipynb&repository_id=44921486&repository_type=Gist#The-%24F%24-Statistic-Revisited)

The -statistic can also be used to compare two nested models, that is, two models trained on the same dataset where one of the models contains a subset of the variables of the other model. The full model contains variables and the reduced model contains a subset of these variables. This allows us to add additional variables to a base model and then test if adding the variables helped the model fit.

where where is the number of variables in model .

### Akaike Information Criterion (AIC)[¶](https://render.githubusercontent.com/view/ipynb?commit=627e349a1c650805ebce426ba14560e89cdecf60&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f676973742f4a43617264656e617352647a2f35346131353165613663303638643661346536663764343564636437623337382f7261772f363237653334396131633635303830356562636534323662613134353630653839636465636636302f4d696e695f50726f6a6563745f4c696e6561725f52656772657373696f6e5f536f6c7574696f6e2e6970796e62&nwo=JCardenasRdz%2F54a151ea6c068d6a4e6f7d45dcd7b378&path=Mini_Project_Linear_Regression_Solution.ipynb&repository_id=44921486&repository_type=Gist#Akaike-Information-Criterion-(AIC))

Another statistic for comparing two models is AIC, which is based on the likelihood function and takes into account the number of variables in the model.

where is the likelihood of the model. AIC is meaningless in the absolute sense, and is only meaningful when compared to AIC values from other models. Lower values of AIC indicate better fitting models.

statsmodels provides the AIC in its output.

### Part 4 Checkup Exercises

**Exercise:** Find another variable (or two) to add to the model we built in Part 3. Compute the -test comparing the two models as well as the AIC. Which model is better?

In [247]:

res1 = ols('PRICE ~ CRIM + RM + PTRATIO',bos).fit()

print(res1.aic)

print(res1.f\_pvalue)

res2 = ols('PRICE ~ CRIM + RM + PTRATIO + AGE + RAD + LSTAT',bos).fit()

print(res2.aic)

print(res2.f\_pvalue)

3233.10027446

1.08999376748e-97

3110.57429901

1.35309219564e-121

**The second model is better**

## Part 5: Evaluating the Model via Model Assumptions and Other Issues[¶](https://render.githubusercontent.com/view/ipynb?commit=627e349a1c650805ebce426ba14560e89cdecf60&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f676973742f4a43617264656e617352647a2f35346131353165613663303638643661346536663764343564636437623337382f7261772f363237653334396131633635303830356562636534323662613134353630653839636465636636302f4d696e695f50726f6a6563745f4c696e6561725f52656772657373696f6e5f536f6c7574696f6e2e6970796e62&nwo=JCardenasRdz%2F54a151ea6c068d6a4e6f7d45dcd7b378&path=Mini_Project_Linear_Regression_Solution.ipynb&repository_id=44921486&repository_type=Gist#Part-5:-Evaluating-the-Model-via-Model-Assumptions-and-Other-Issues)

Linear regression makes several assumptions. It is always best to check that these assumptions are valid after fitting a linear regression model.

* \*\*Linearity\*\*. The dependent variable is a linear combination of the regression coefficients and the independent variables . This can be verified with a scatterplot of each vs. and plotting correlations among . Nonlinearity can sometimes be resolved by [transforming](https://onlinecourses.science.psu.edu/stat501/node/318) one or more independent variables, the dependent variable, or both. In other cases, a [generalized linear model](https://en.wikipedia.org/wiki/Generalized\_linear\_model) or a [nonlinear model](https://en.wikipedia.org/wiki/Nonlinear\_regression) may be warranted.
* \*\*Constant standard deviation\*\*. The SD of the dependent variable should be constant for different values of X. We can check this by plotting each against and verifying that there is no "funnel" shape showing data points fanning out as increases or decreases. Some techniques for dealing with non-constant variance include weighted least squares (WLS), [robust standard errors](https://en.wikipedia.org/wiki/Heteroscedasticity-consistent\_standard\_errors), or variance stabilizing transformations.
* \*\*Normal distribution for errors\*\*. The term we discussed at the beginning are assumed to be normally distributed. This can be verified with a fitted values vs. residuals plot and verifying that there is no pattern, and with a quantile plot. Sometimes the distributions of responses may not be normally distributed at any given value of . e.g. skewed positively or negatively.
* \*\*Independent errors\*\*. The observations are assumed to be obtained independently.
  + e.g. Observations across time may be correlated

There are some other issues that are important investigate with linear regression models.

* \*\*Correlated Predictors:\*\* Care should be taken to make sure that the independent variables in a regression model are not too highly correlated. Correlated predictors typically do not majorly affect prediction, but do inflate standard errors of coefficients making interpretation unreliable. Common solutions are dropping the least important variables involved in the correlations, using regularlization, or, when many predictors are highly correlated, considering a dimension reduction technique such as principal component analysis (PCA).
* \*\*Influential Points:\*\* Data points that have undue influence on the regression model. These points can be high leverage points or outliers. Such points are typically removed and the regression model rerun.

### Part 5 Checkup Exercises

Take the reduced model from Part 3 to answer the following exercises. Take a look at [this blog post](http://mpastell.com/2013/04/19/python\_regression/) for more information on using statsmodels to construct these plots.

**Exercise:** Construct a fitted values versus residuals plot. What does the plot tell you? Are there any violations of the model assumptions?

**Exercise:** Construct a quantile plot of the residuals. What does the plot tell you?

**Exercise:** What are some advantages and disadvantages of the fitted vs. residual and quantile plot compared to each other?

**Exercise:** Identify any outliers (if any) in your model and write a story describing what these outliers might represent.

**Exercise:** Construct a leverage plot and identify high leverage points in the model. Write a story explaining possible reasons for the high leverage points.

**Exercise:** Remove the outliers and high leverage points from your model and run the regression again. How do the results change?

In [248]:

# Your turn.

import statsmodels.formula.api as sm

model = sm.ols(formula='PRICE ~ CRIM + RM + PTRATIO', data=bos)

fitted = model.fit()

x=bos.PRICE.values;

r=fitted.resid

plt.figure(); plt.plot(x,r,'o'); plt.xlabel("predicted price"); plt.ylabel("residual")

import scipy.stats as stats

plt.figure();

stats.probplot(r, dist="norm", plot=plt)

plt.title("Normal Q-Q plot")

plt.show()

# find houses for which the predicted prices is less than 50

index50=x<50

# plot those houses

plt.figure(); plt.title("Scatter for price <50")

plt.figure(); plt.plot(x[index50],r[index50],'or');

plt.figure();

stats.probplot(r[index50], dist="norm", plot=plt)

plt.title("Normal Q-Q plot")

plt.show()

1. Exercise: Construct a fitted values versus residuals plot. What does the plot tell you? Are there any violations of the model assumptions?  
   The residuals are not distributed randomly distributed
2. Exercise: Construct a quantile plot of the residuals. What does the plot tell you?  
   That the residuals don't follow a normal distribution
3. Exercise: What are some advantages and disadvantages of the fitted vs. residual and quantile plot compared to each other?  
   The qq plot makes the interpretation much easier, just follow the line. The first plot can tells us where the data deviation from the assumption
4. Exercise: Identify any outliers (if any) in your model and write a story describing what these outliers might represent. The houses for which a price of 50 is predicted are the outliers. these houses migh contain informaiton that is not handled property by the model
5. Exercise: Construct a leverage plot and identify high leverage points in the model. Write a story explaining possible reasons for the high leverage points. I created the leverage plot, but it is easier to see what is going on if I estimate the influeance of each point using cook's distance. It is hard to know the reason, but it makes it easier to eliminate these points. See next.
6. Exercise: Remove the outliers and high leverage points from your model and run the regression again. How do the results change?  
   the adjusted r-square improved to 0.740

In [249]:

from statsmodels.graphics.regressionplots import \*

plot\_leverage\_resid2(fitted)

influence\_plot(fitted)

plt.figure(); plt.hist(x[0:360])

Out[249]:

(array([ 28., 62., 105., 62., 31., 26., 18., 5., 8., 15.]),

array([ 11.8 , 15.62, 19.44, 23.26, 27.08, 30.9 , 34.72, 38.54,

42.36, 46.18, 50. ]),

<a list of 10 Patch objects>)

In [250]:

influence = fitted.get\_influence()

#c is the distance and p is p-value

(c, p) = influence.cooks\_distance

#plt.stem(np.arange(len(c)), c, markerfmt=",")

plt.subplot(1,2,1); plt.plot(c,'o')

plt.subplot(1,2,2); plt.plot(c,'o'); plt.ylim(ymin=0,ymax=0.05)

Out[250]:

(0, 0.05)

In [251]:

index\_lev= c<=.01

bos\_clean=bos[index\_lev];

import statsmodels.formula.api as sm

model\_clean = sm.ols(formula='PRICE ~ CRIM + RM + PTRATIO', data=bos\_clean)

fitted\_clean = model\_clean.fit()

print(fitted\_clean.summary2())

Results: Ordinary least squares

==================================================================

Model: OLS Adj. R-squared: 0.740

Dependent Variable: PRICE AIC: 2686.4195

Date: 2017-02-13 16:59 BIC: 2703.0812

No. Observations: 476 Log-Likelihood: -1339.2

Df Model: 3 F-statistic: 452.7

Df Residuals: 472 Prob (F-statistic): 1.92e-138

R-squared: 0.742 Scale: 16.403

-------------------------------------------------------------------

Coef. Std.Err. t P>|t| [0.025 0.975]

-------------------------------------------------------------------

Intercept -10.0726 3.0429 -3.3102 0.0010 -16.0520 -4.0932

CRIM -0.3191 0.0313 -10.1775 0.0000 -0.3807 -0.2575

RM 8.0605 0.3248 24.8174 0.0000 7.4223 8.6987

PTRATIO -0.9557 0.0948 -10.0854 0.0000 -1.1419 -0.7695

------------------------------------------------------------------

Omnibus: 8.175 Durbin-Watson: 1.012

Prob(Omnibus): 0.017 Jarque-Bera (JB): 11.338

Skew: 0.133 Prob(JB): 0.003

Kurtosis: 3.708 Condition No.: 327

==================================================================

Value\_counts() – pandas

From === import \*