Project: Regression Modeling with the Boston Housing Dataset

Introduction

In this lab, you'll apply the regression analysis and diagnostics techniques covered in this section to the "Boston Housing" dataset. You performed a detailed EDA for this dataset earlier on, and hopefully, you more or less recall how this data is structured! In this lab, you'll use some of the features in this dataset to create a linear model to predict the house price!

Objectives

You will be able to:

- Perform a linear regression using statsmodels
- Determine if a particular set of data exhibits the assumptions of linear regression
- Evaluate a linear regression model by using statistical performance metrics pertaining to overall model and specific parameters
- Use the coefficient of determination to determine model performance
- Interpret the parameters of a simple linear regression model in relation to what they signify for specific data

Let's get started

Import necessary libraries and load 'BostonHousing.csv' as a pandas dataframe

In [1]:
```
# Your code here
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('ggplot')
boston = pd.read_csv('BostonHousing.csv')
```

The columns in the Boston housing data represent the dependent and independent variables. The dependent variable here is the median house value `MEDV` . The description of the other variables is available on KAGGLE (https://www.kaggle.com/c/boston-housing).

Inspect the columns of the dataset and comment on type of variables present

In [2]:
```
# Your code here
boston.head()
```

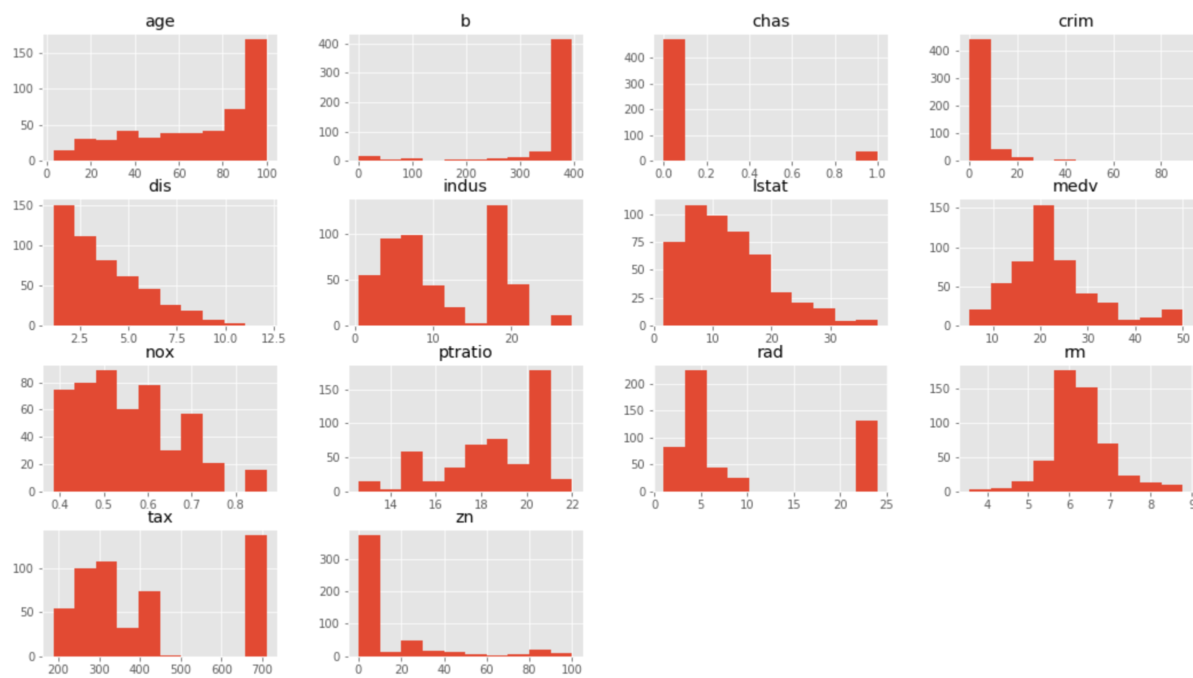
Out[2]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

In [3]:
```
# Record your observations here
# The dataset mostly contains continuous variables
# chas and rad are the only two categorical variables
# there are no null and missing values
```

Create histograms for all variables in the dataset and comment on their shape (uniform or not?)

In [4]:
```
# Your code here
boston.hist(figsize=(18,10) );
```



In [5]:
```
# You observations here
# We see lot of skewness and kurtosis in most variables e.g. dis, age
# Some variables have outliers at extreme tails
# the target variables looks good with some outliers in the right tail
```

Based on this, we preselected some features for you which appear to be more 'normal' than others.

Create a new dataset with `['crim', 'dis', 'rm', 'zn', 'age', 'medv']`

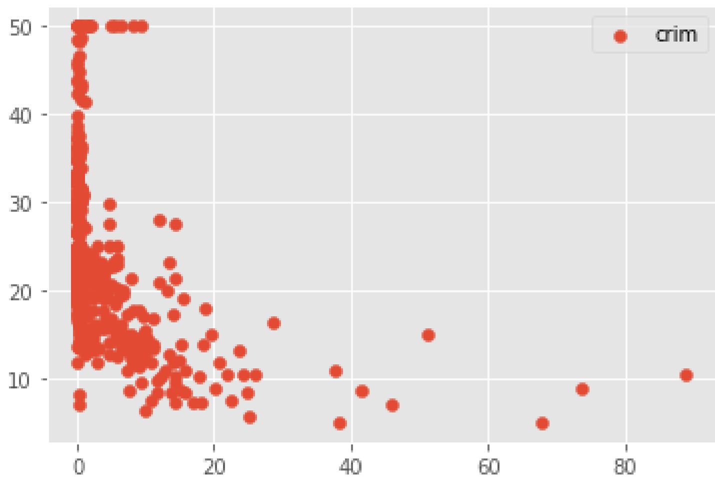
In [6]:
```
# Your code here
data = boston[['crim', 'dis', 'rm', 'zn', 'age', 'medv']].copy()
data.head()
```

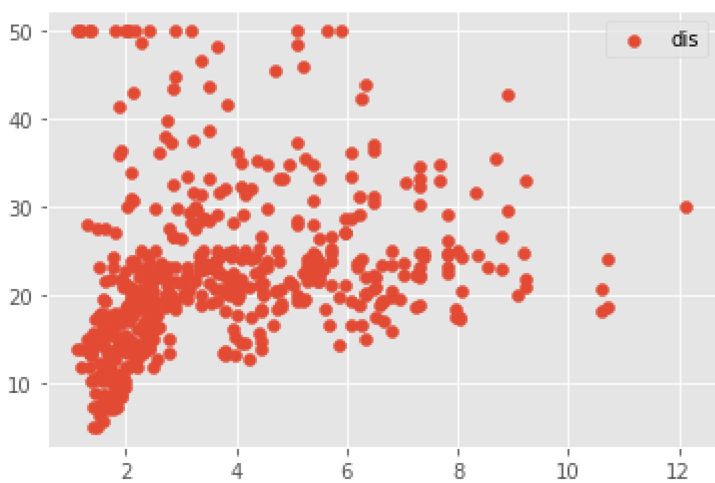
Out[6]:

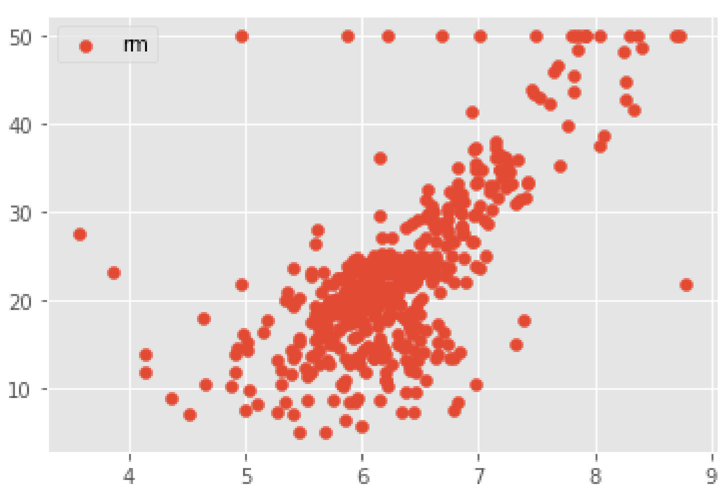
	crim	dis	rm	zn	age	medv
0	0.00632	4.0900	6.575	18.0	65.2	24.0
1	0.02731	4.9671	6.421	0.0	78.9	21.6
2	0.02729	4.9671	7.185	0.0	61.1	34.7
3	0.03237	6.0622	6.998	0.0	45.8	33.4
4	0.06905	6.0622	7.147	0.0	54.2	36.2

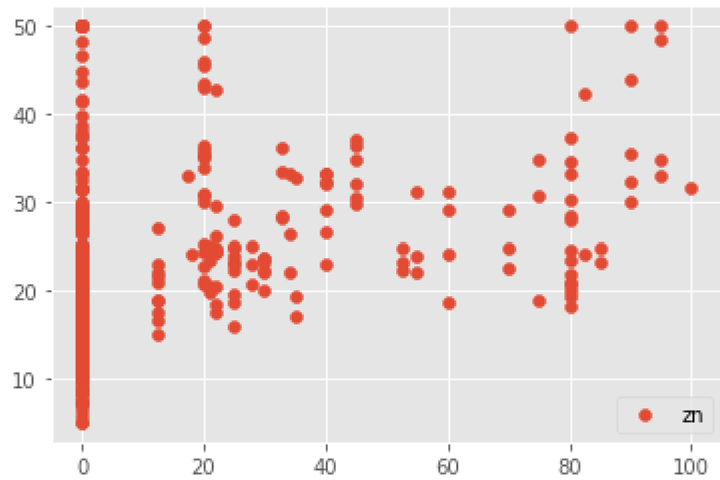
Check the linearity assumption for all chosen features with target variable using scatter plots

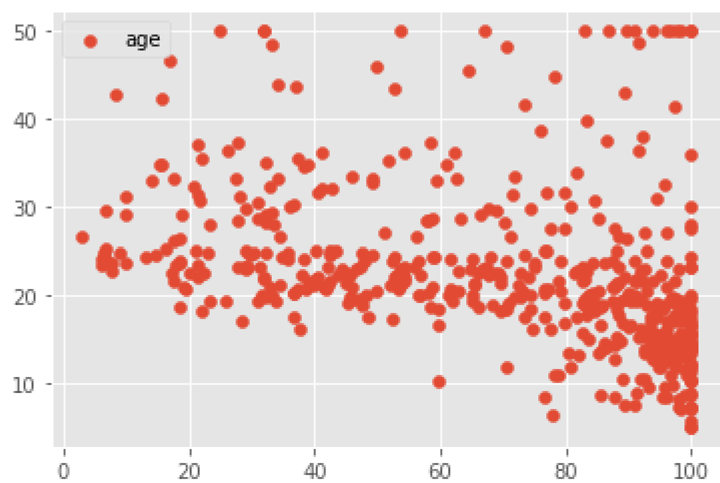
In [7]:
```
# Your code here
for column in ['crim', 'dis', 'rm', 'zn', 'age']:
    plt.scatter(data[column], data.medv, label=column)
    plt.legend()
    plt.show()
```


crim
50
40
30
20
10
0
20
40
60
80


dis
50
40
30
20
10
2
4
6
8
10
12


rm
50
40
30
20
10
4
5
6
7
8
9





In [8]:
```
# Your observations here
# Crim variable's linearity seemd a bit unclear as the values are too close to
each other and generally very small
# There is SOME linearity apparent in some variables although the variance alo
ng the y-axis is a bit unpredictable for some values
# Some outliers are present in almost all cases
# Data probably needs more normalization and pre-processing to "Clean it up"
```

Clearly, your data needs a lot of preprocessing to improve the results. This key behind a Kaggle competition is to process the data in such a way that you can identify the relationships and make predictions in the best possible way. For now, we'll use the dataset untouched and just move on with the regression. The assumptions are not *exactly* all fulfilled, but they still hold to a level that we can move on.

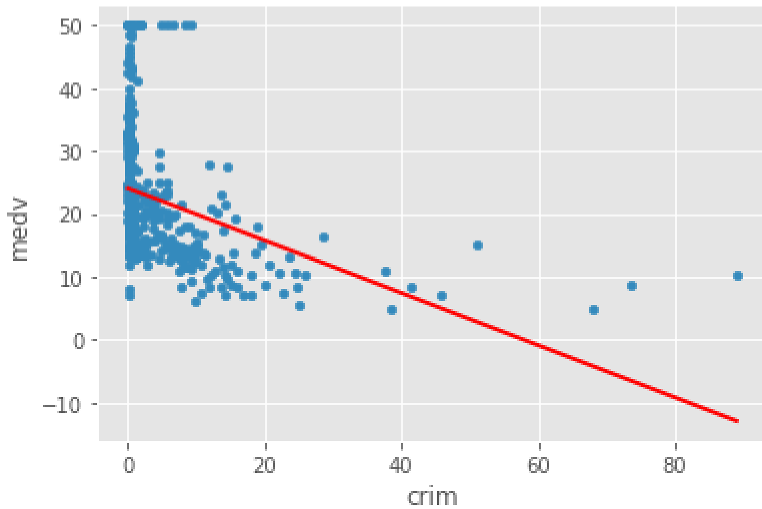
Let's do Regression

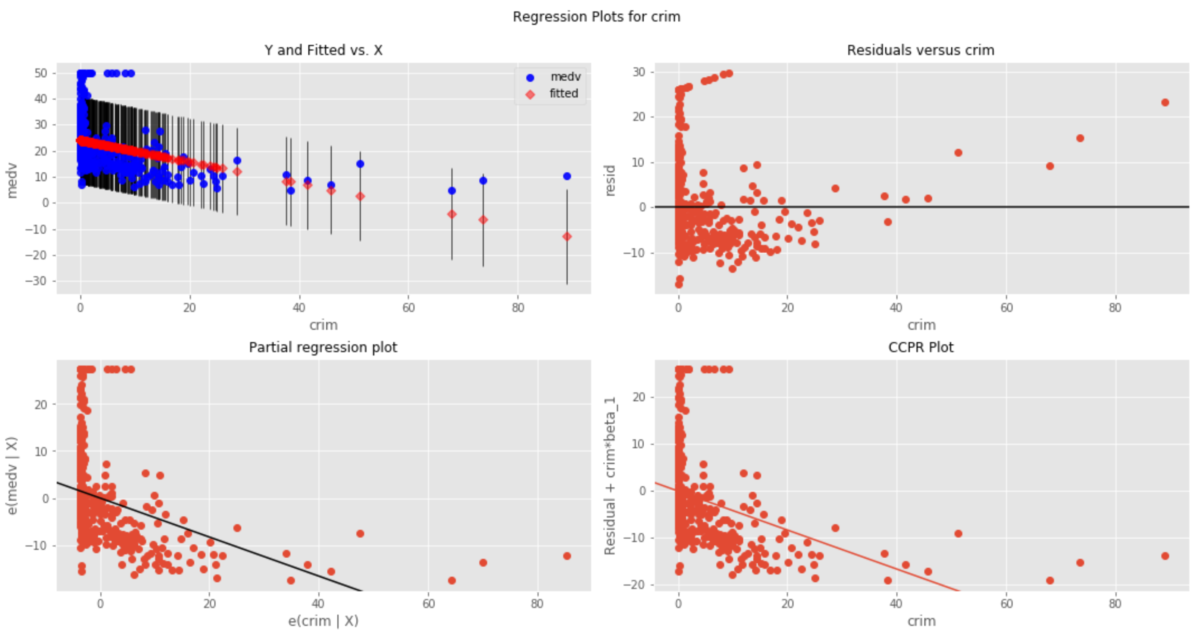
Now, let's perform a number of simple regression experiments between the chosen independent variables and the dependent variable (price). You'll do this in a loop and in every iteration, you should pick one of the independent variables. Perform the following steps:

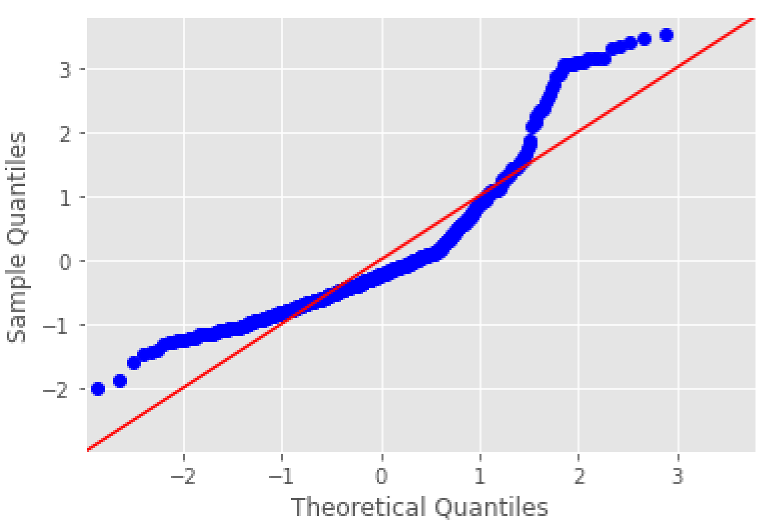
- Run a simple OLS regression between independent and dependent variables
- Plot a regression line on the scatter plots
- Plot the residuals using `sm.graphics.plot_regress_exog()`
- Plot a Q-Q plot for regression residuals normality test
- Store following values in array for each iteration:
  - Independent Variable
  - r_squared'
  - intercept'
  - 'slope'
  - 'p-value'
  - 'normality (JB)'
- Comment on each output

In [9]:
```
# Your code here
# import libraries
import statsmodels.api as sm
import statsmodels.formula.api as smf
import scipy.stats as stats
import statsmodels.stats.api as sms



results = [['ind_var', 'r_squared', 'intercept', 'slope', 'p-value', 'normalit
y (JB)' ]]
for idx, val in enumerate(['crim', 'dis', 'rm', 'zn', 'age']):
    print ("Boston Housing DataSet - Regression Analysis and Diagnostics for f
ormula: medv~" + val)
    print ("-------------------------------------------------------------------
-------------------")

    f = 'medv~' + val
#
    model = smf.ols(formula=f, data=data).fit()

    X_new = pd.DataFrame({val: [data[val].min(), data[val].max()]});
    preds = model.predict(X_new)
    data.plot(kind='scatter', x=val, y='medv');
    plt.plot(X_new, preds, c='red', linewidth=2);
    plt.show()
    fig = plt.figure(figsize=(15,8))
    fig = sm.graphics.plot_regress_exog(model, val, fig=fig)
    fig = sm.graphics.qqplot(model.resid, dist=stats.norm, line='45', fit=True
,   )
    plt.show()

    results.append([val, model.rsquared, model.params[0], model.params[1], mod
el.pvalues[1], sms.jarque_bera(model.resid)[0] ])
    #input("Press Enter to continue...")
```

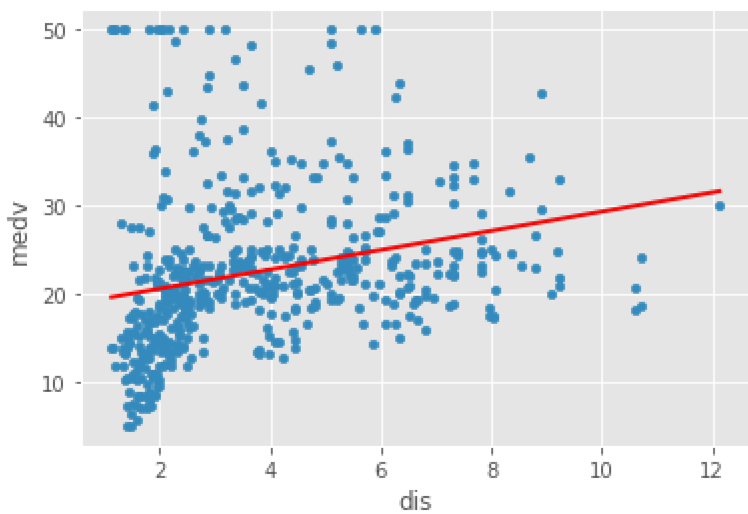
```
Boston Housing DataSet - Regression Analysis and Diagnostics for formula: med
v~crim
---------------------------------------------------------------------------
--------
```

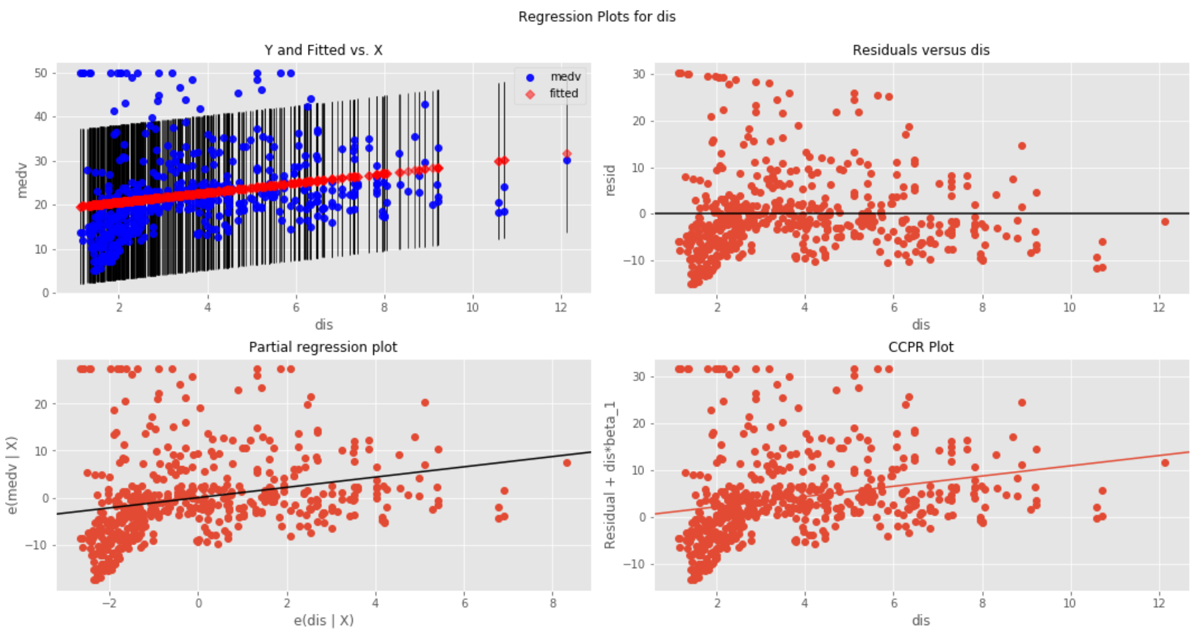


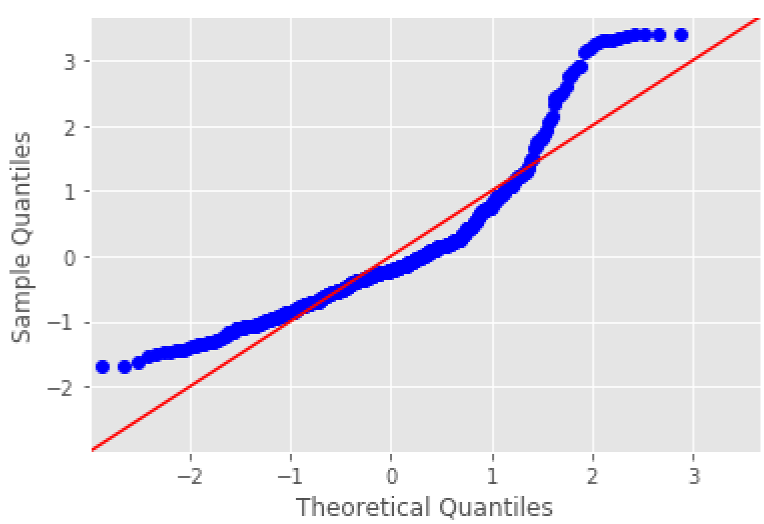




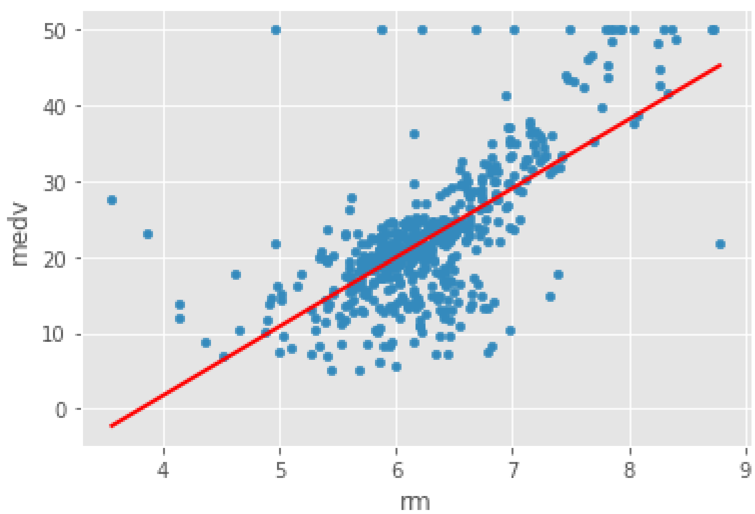
```
Boston Housing DataSet - Regression Analysis and Diagnostics for formula: med
v~dis
---------------------------------------------------------------------------
--------
```

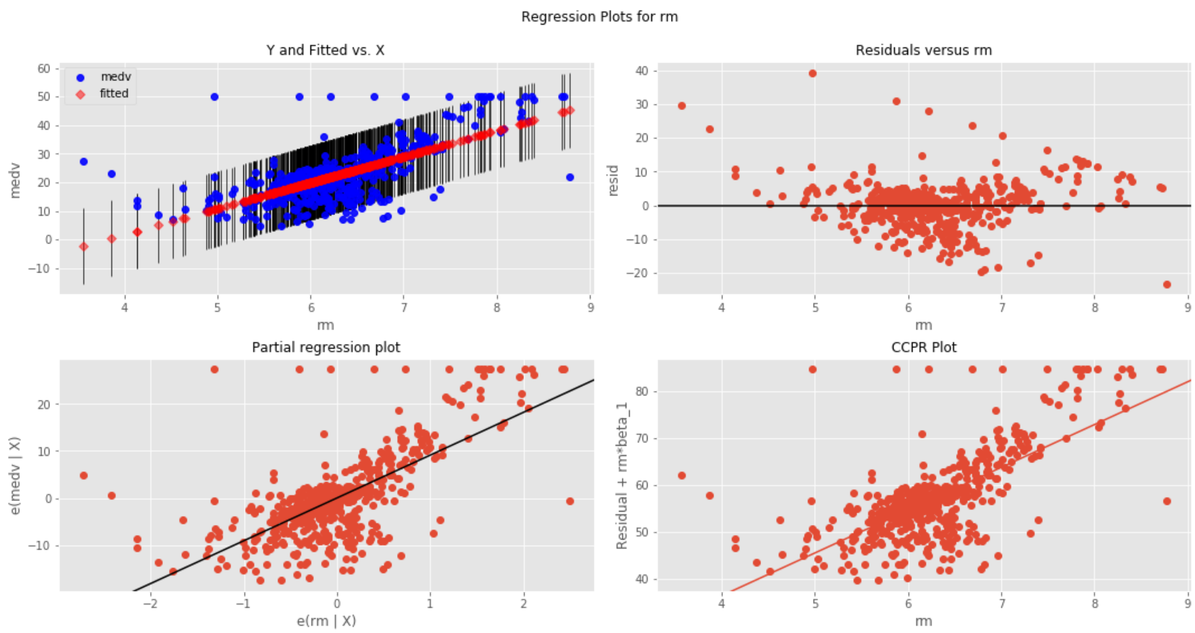


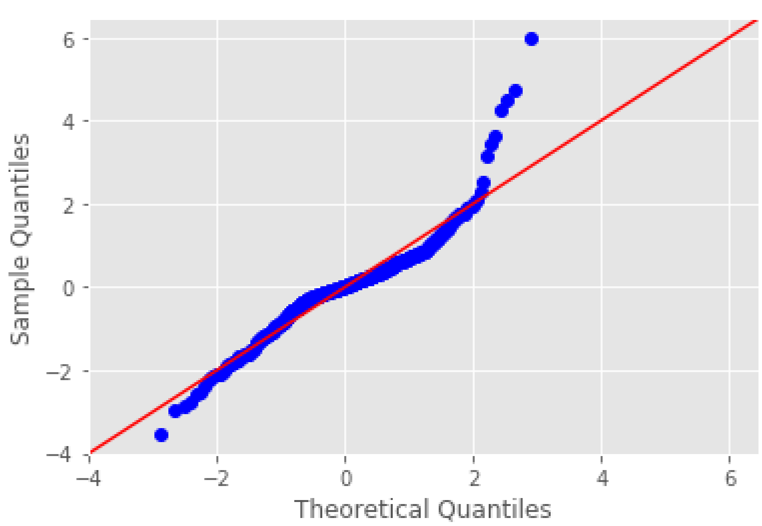




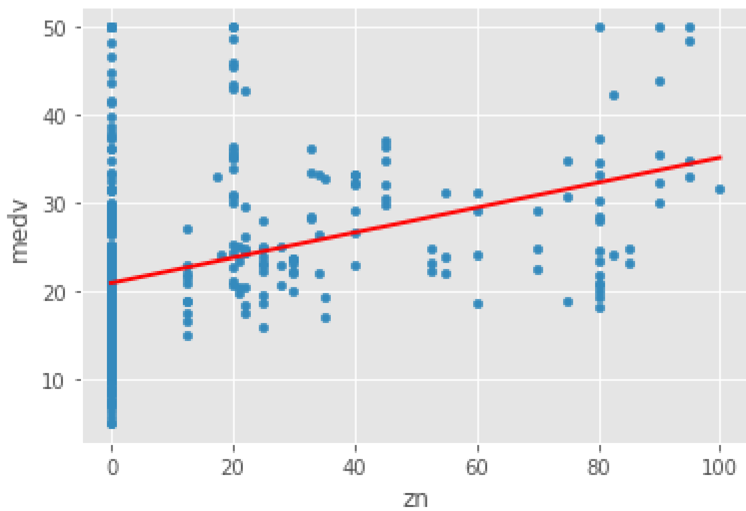
```
Boston Housing DataSet - Regression Analysis and Diagnostics for formula: med
v~rm
--------------------------------------------------------------------------------
--------
```

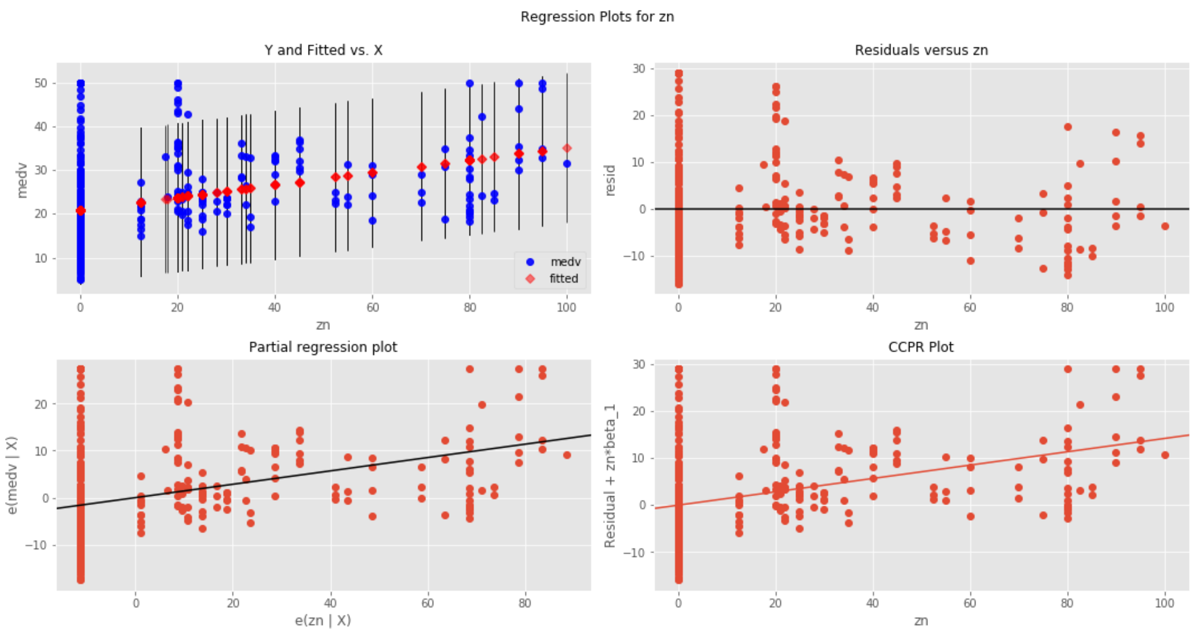


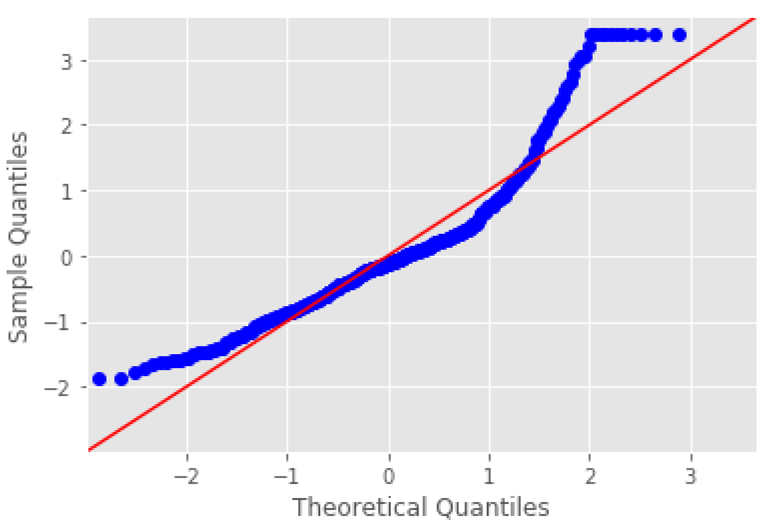




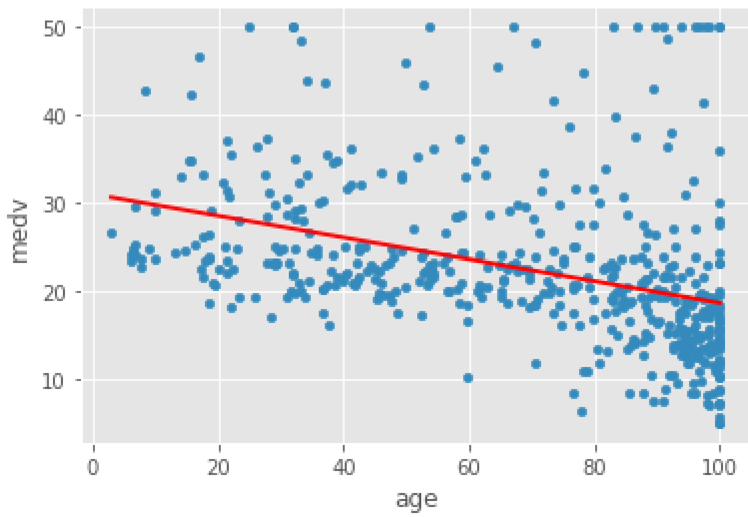
```
Boston Housing DataSet - Regression Analysis and Diagnostics for formula: med
v~zn
-------------------------------------------------------------------------------
--------
```

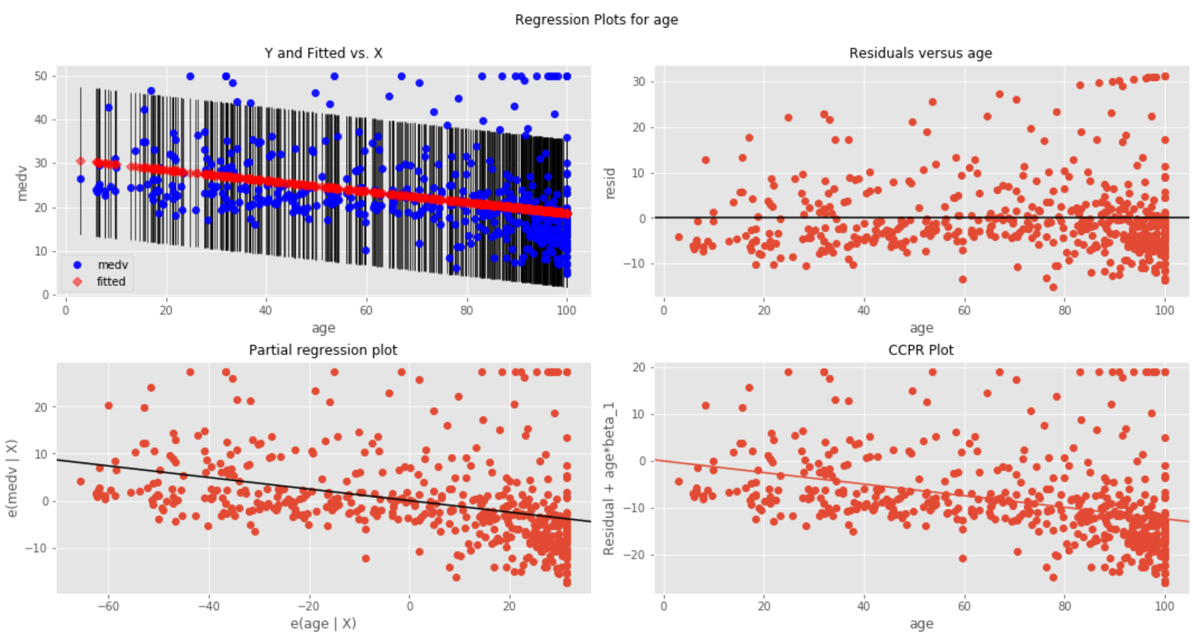


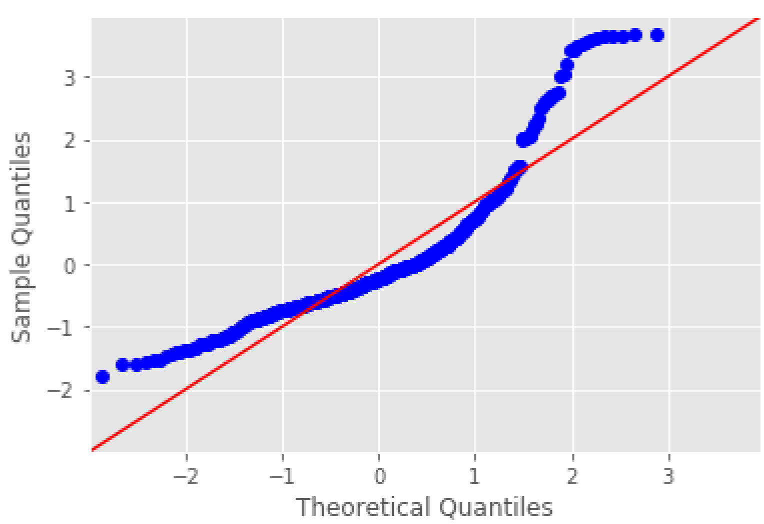




```
Boston Housing DataSet - Regression Analysis and Diagnostics for formula: med
v~age
-----------------------------------------------------------------------------
--------
```







In [10]:
```
pd.DataFrame(results)
```

Out[10]:

	0	1	2	3	4	5
0	ind_var	r_squared	intercept	slope	p-value	normality (JB)
1	crim	0.15078	24.0331	-0.41519	1.17399e-19	295.404
2	dis	0.0624644	18.3901	1.09161	1.20661e-08	305.104
3	rm	0.483525	-34.6706	9.10211	2.48723e-74	612.449
4	zn	0.129921	20.9176	0.14214	5.71358e-17	262.387
5	age	0.142095	30.9787	-0.123163	1.56998e-18	456.983

In [11]:
```
#Your observations here

# We can do a detailed analysis of each exsperiment and eloborate in detail
# Here we shall show a summary of selected observations

# Crime has a negative relationship with price i.e. less crime > higher price
 and vice vera
# Crime does not show any clear signs heteroscedasticity
# Crime has a low r-squared so not such a good fit
# Residuals are not normally distributed (needs log normalization that we'll s
ee in next section)

# A positive relationship exists between dis and medv
# dis residual plots show some signs of heteroscadasticity as cone shaped resi
duals
# normality is still questionable

# rm shows a strong positive relationship
# rm residuals show no signs of heteroscdasticity however some outliers are pr
esent
# rm qqplot shows a long right tail which hurts normality

# zn variable scatter shows a lot of variance along the y-axis and hence gives
a very low r-squared value
# no clear heteroscedasticity in residuals
# normality through Q-Q plots and JB is far from perfect

# age has a negative relatioship with prices i.e. young people > expensive hou
ses :o
# Some obvious heteroscadasticity and normality is questionable.
```

Clearly, the results are not very reliable. The best R-Squared is witnessed with `rm` , so in this analysis, this is our best predictor.

How can you improve these results?

1. Preprocessing

This is where the preprocessing of data comes in. Dealing with outliers, normalizing data, scaling values etc. can help regression analysis get more meaningful results from the given data.

1. Advanced Analytical Methods

Simple regression is a very basic analysis technique and trying to fit a straight line solution to complex analytical questions may prove to be very inefficient. Later on, you'll explore multiple regression where you can use multiple features **at once** to define a relationship with the outcome. You'll also look at some preprocessing and data simplification techniques and revisit the Boston dataset with an improved toolkit.

Level up - Optional

Apply some data wrangling skills that you have learned in the previous section to pre-process the set of independent variables we chose above. You can start off with outliers and think of a way to deal with them. See how it affects the goodness of fit.

Summary

In this lab, you applied your skills learned so far on a new data set. You looked at the outcome of your analysis and realized that the data might need some preprocessing to see a clear improvement in the results. You'll pick this back up later on, after learning about more preprocessing techniques and advanced modeling techniques.