Kaggle Dataset: House Sales in King County

Steps of Data Exploration and Preparation

Variable Identification

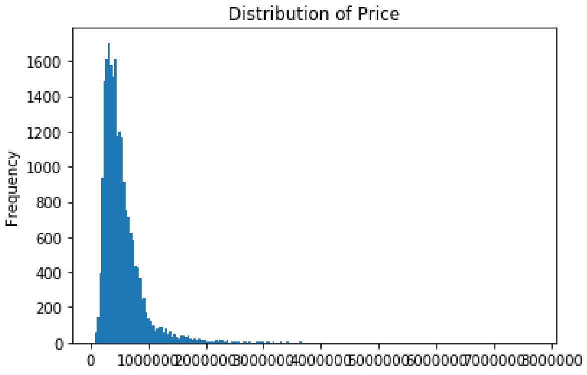
Dependent Variable: **price**
Independent Variables: The table below shows the independent variables with their datatype.
For my analysis I have changed the datatype of Date variable to numeric for uniformity. And Variables having more than 20 categories are considered as Numeric.

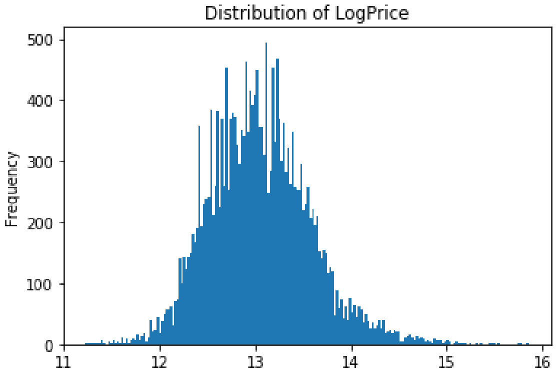
Variables	Datatype	Variable	datatype	Variable	Datatype
id	Numeric	waterfront	Category	Yr_renovated	Numeric
date	Numeric	view	Category	zipcode	Numeric
bedrooms	Category	condition	Category	lat	Numeric
bathrooms	Category	grade	Category	long	Numeric
sqft_living	Numeric	sqft_above	Numeric	sqft_living15	Numeric
sqft_lot	Numeric	Sqft_basement	Numeric	Sqft_lot5	Numeric
floors	Category	Yr_built	Numeric		

Univariate Analysis

At this stage, we explore variables one by one. Method to perform univariate analysis depends on whether the variable type is categorical or continuous.

Continuous Variables: - In case of continuous variables, we need to understand the central tendency and spread of the variable. I used the histogram plot to analyze the distribution of numerical variables. For variables having Skewed distribution, I have used log transformation to have approx. normal distribution.





Numerical Variables having skewed distribution are price, sqft_living, sqft_lot, sqft_above, sqft_living15, sqft_lot15.

Bi-variate Analysis

Bi-variate Analysis finds out the relationship between two variables. Here, we look for association and disassociation between variables at a pre-defined significance level. Here I have used the correlation between two numerical variables as a factor to include/exclude a variable in the model.

For variables having high correlation, I have used one of them in the final model on basis of P-values which will be discussed later.

sqft_living, sqft_above: 0.88

sqft_living, sqft_living15: 0.76

sqft_living, bathrooms: 0.75

sqft_bathrooms, basement: 0.75

Missing Value Treatment

Missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behavior and relationship with other variables correctly. It can lead to wrong prediction or classification.

Fortunately, this data set does not have any missing values.

Feature Selection

I have used statsmodels.formula.api library in python to build the model.

Firstly, I removed the Date and ID since they are insignificant to the model.

Then I performed regression analysis with all 19 variables which resulted in R^2 of .695 and adjusted R^2 as 0.695.

Then I removed the numerical variables having multicollinearity sqft_above, sqft_basement, sqft_living15, basement and bathroom.

Later I included bathroom in the model as R^2 had decreased on its removal.

Then I used Log transformation on the variables having skewed distribution (mentioned above).

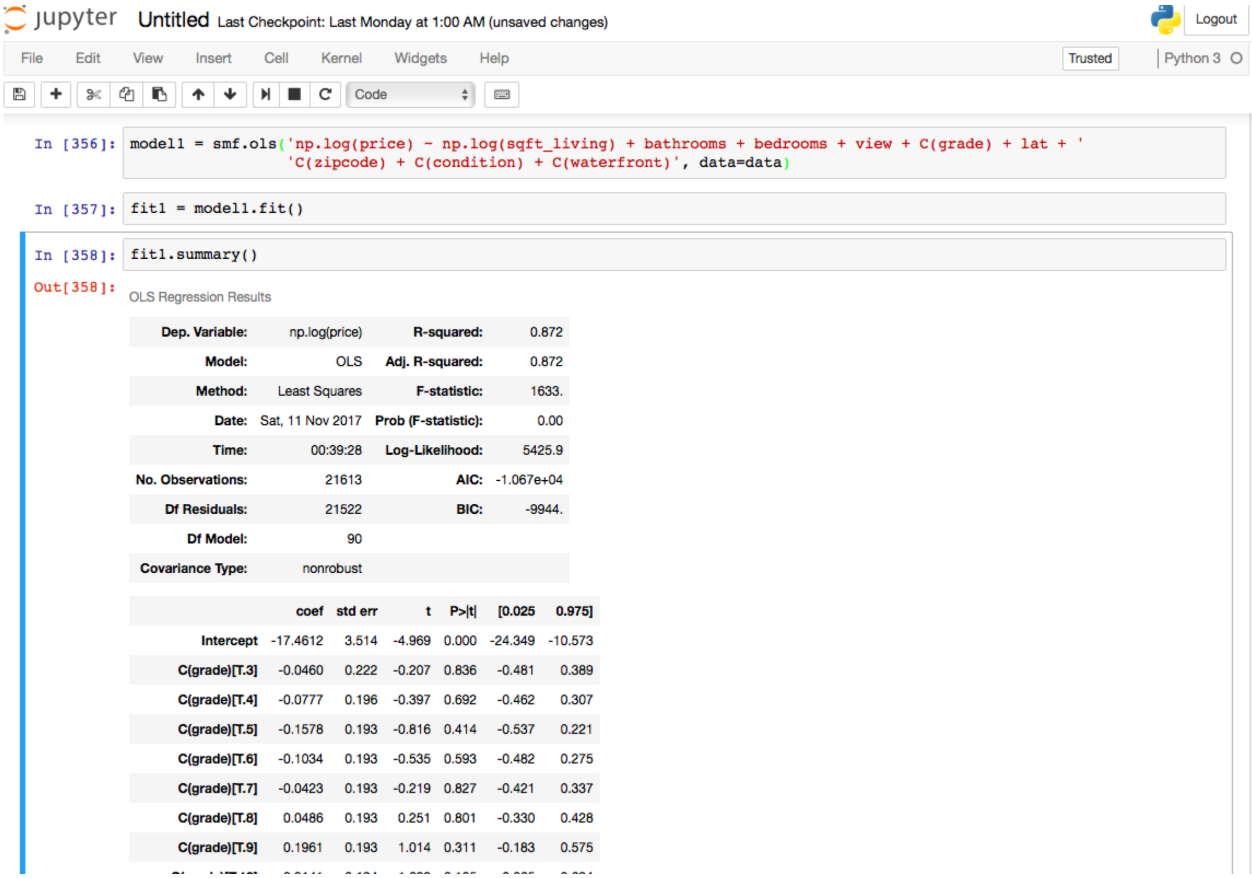
P-value:

I removed all the variables which were having a p-values higher than the significance level of 0.05, which means they are statistically insignificant to the model.

Confidence Interval:

I removed all the variables having a value of zero lying in their confidence interval, which signifies that the mean of variances of the variable is same and we cannot reject the null hypothesis that all variances in the variables are same.

My final model has nine variables sqft_living, bathrooms, bedrooms, view, grade, lat, zipcode, condition and waterfront. I have used log transformation on Price and sqft_living and got the final R^2 value of 0.872 and adjusted R^2 of 0.872.



```
In [356]: model1 = smf.ols('np.log(price) ~ np.log(sqft_living) + bathrooms + bedrooms + view + C(grade) + lat + '
                           'C(zipcode) + C(condition) + C(waterfront)', data=data)

In [357]: fit1 = model1.fit()

In [358]: fit1.summary()

Out[358]:
```

OLS Regression Results

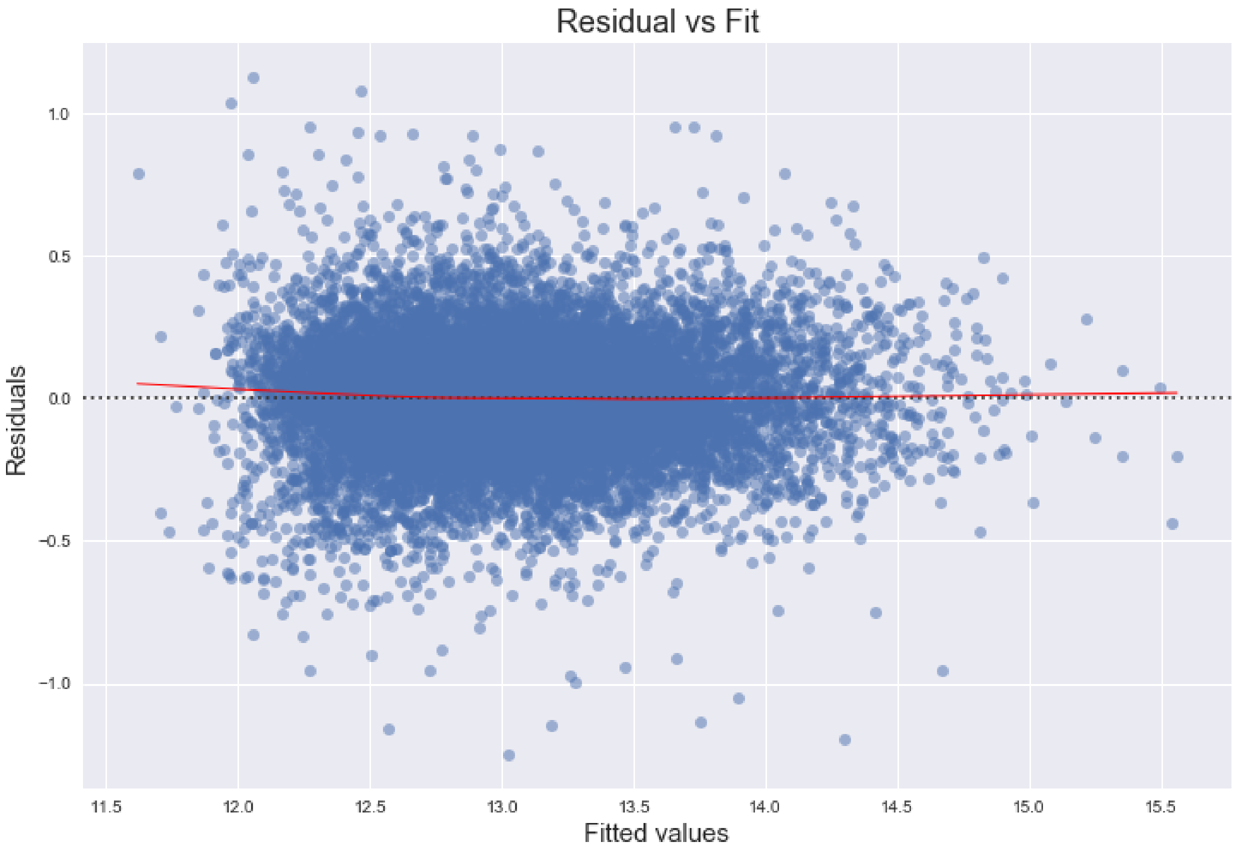
Dep. Variable:	np.log(price)	**R-squared:**	0.872
Model:	OLS	**Adj. R-squared:**	0.872
Method:	Least Squares	**F-statistic:**	1633.
Date:	Sat, 11 Nov 2017	**Prob (F-statistic):**	0.00
Time:	00:39:28	**Log-Likelihood:**	5425.9
No. Observations:	21613	**AIC:**	-1.067e+04
Df Residuals:	21522	**BIC:**	-9944.
Df Model:	90		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-17.4612	3.514	-4.969	0.000	-24.349	-10.573
C(grade)[T.3]	-0.0460	0.222	-0.207	0.836	-0.481	0.389
C(grade)[T.4]	-0.0777	0.196	-0.397	0.692	-0.462	0.307
C(grade)[T.5]	-0.1578	0.193	-0.816	0.414	-0.537	0.221
C(grade)[T.6]	-0.1034	0.193	-0.535	0.593	-0.482	0.275
C(grade)[T.7]	-0.0423	0.193	-0.219	0.827	-0.421	0.337
C(grade)[T.8]	0.0486	0.193	0.251	0.801	-0.330	0.428
C(grade)[T.9]	0.1961	0.193	1.014	0.311	-0.183	0.575

This model can explain approximately 87.2% variability of house prices.

Residual plot¶

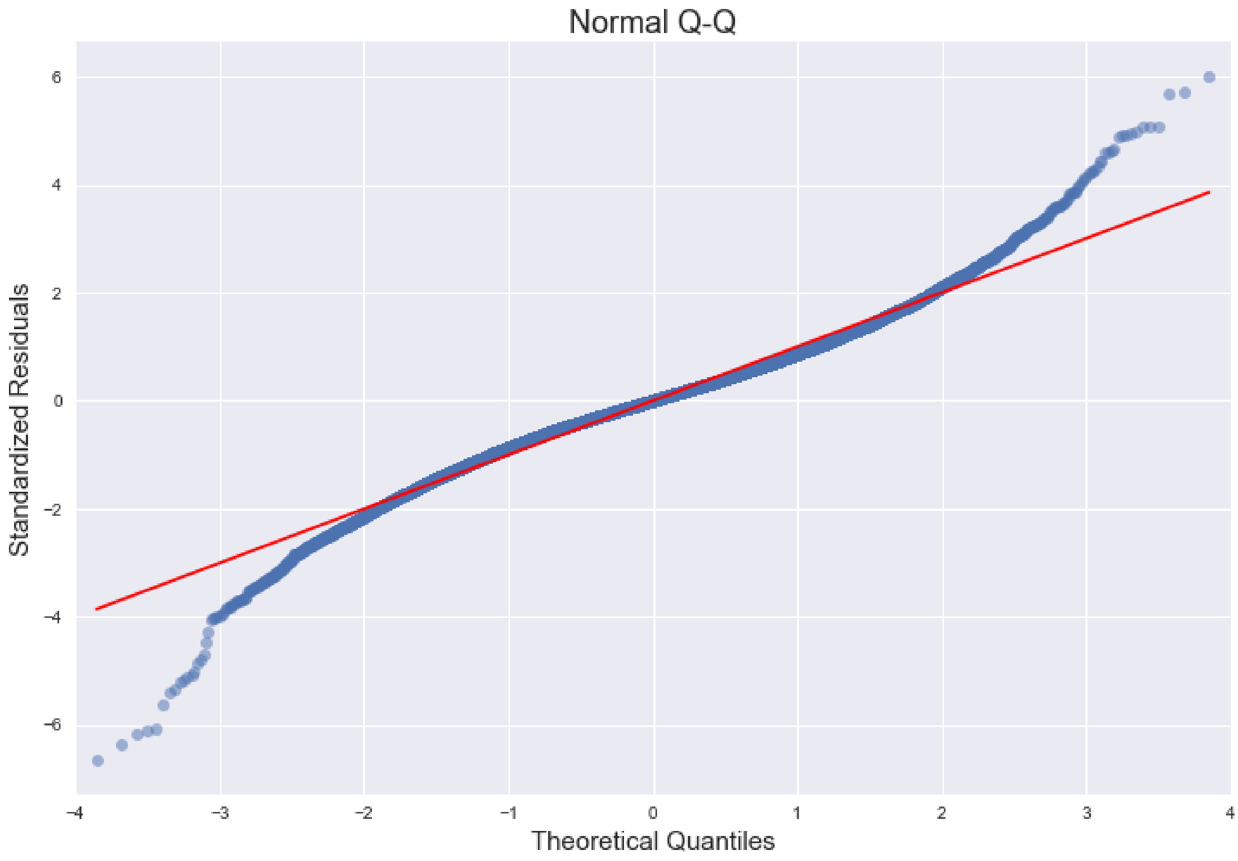
This is a scatterplot of fitted values against residuals, with a “locally weighted scatterplot smoothing (lowess)” regression line showing any apparent trend. This one can be easily plotted using seaborn residplot with fitted values as x parameter, and the dependent variable as y. lowess=True makes sure the lowess regression line is drawn. Additional parameters are passed to underlying matplotlib scatter and line functions using scatter_kws and line_kws, also titles and labels are set using matplotlib methods.



2. QQ plot

This one shows how well the distribution of residuals fit the normal distribution. This plots the standardized (z-score) residuals against the theoretical normal quantiles. For this, I’m using ProbPlot and its qqplot method from statsmodels graphics API. statsmodels actually has a qqplot method that we can use directly.

If a set of observation are approximately normally distributed, the normal Q-Q plot would result in approximately straight line



Fat tails Interpretation

The plot shows a dataset with “fat tails,” meaning that compared to the normal distribution there is more data located at the extremes of the distribution and less data in the center of the distribution. In terms of quantiles this means that the first quantile is much less than the first theoretical quantile and the last quantile is greater than the last theoretical quantile.

In other words, larger values are larger (more extreme) than the expected values and the smaller values(at the bottom) are smaller than expected.