Correlation and Regression:

Can one predict the market prices of the houses based on the basic information about the houses?

CS50: Formal Analysis

Minerva Schools at KGI

Milana Stetsenko

February 2020

Introduction:

This dataset contains house sale prices for King County, WA. It includes the homes sold from May 2014 to May 2015 by harlfoxem (Harlfoxem). The research question chosen for this dataset is: How can I best predict the market price of the house, knowing the trends of the previous sales, not to be tricked by overpricing of realtors? I will use python, Data Science learning materials, and OpenIntro (Diez) to conduct the most precise research and to automate the findings.

Dataset:

I retrieved the dataset from Kaggle. The variables chosen for the analysis are the ones that most often can be found on the real estate websites, so the regular person without the need to obtain any additional information can estimate the price of the house. The initial predictor variables are quantitative discrete: a number of bedrooms, number of bathrooms; quantitative continuous: square footage of the house, squared footage of the basement, and the year the house was built; and qualitative categorical: the condition of the house. The response variable is the price (continuous quantitative). It is important to identify the types of the variable to properly interpret the regression line (for categorical variable).[1] No cleaning for the data was needed.

Methods:

Defining the variables

First, I will compute the descriptive statistics (computation can be found in Appendix A).

	price	bedrooms	bathrooms	sqft_living	sqft_basement	condition	yr_built
count	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000
mean	5.400881e+05	3.370842	2.114757	2079.899736	291.509045	3.409430	1971.005136
std	3.671272e+05	0.930062	0.770163	918.440897	442.575043	0.650743	29.373411
min	7.500000e+04	0.000000	0.000000	290.000000	0.000000	1.000000	1900.000000
25%	3.219500e+05	3.000000	1.750000	1427.000000	0.000000	3.000000	1951.000000
50%	4.500000e+05	3.000000	2.250000	1910.000000	0.000000	3.000000	1975.000000
75%	6.450000e+05	4.000000	2.500000	2550.000000	560.000000	4.000000	1997.000000
max	7.700000e+06	33.000000	8.000000	13540.000000	4820.000000	5.000000	2015.000000

Figure 1. The descriptive statistics table for the predictor and response variables of the analysis. This shows that the data has no significant errors when computing the values of data.

[1] #variables: The variables are described and classified. One needs to know whether the variable is a predictor or a response, quantitative or qualitative to interpret the plots properly because the graph would look differently.

Then, I computed a heatmap to see the distribution of correlations between all the variables to identify the multicollinearity (Appendix B).

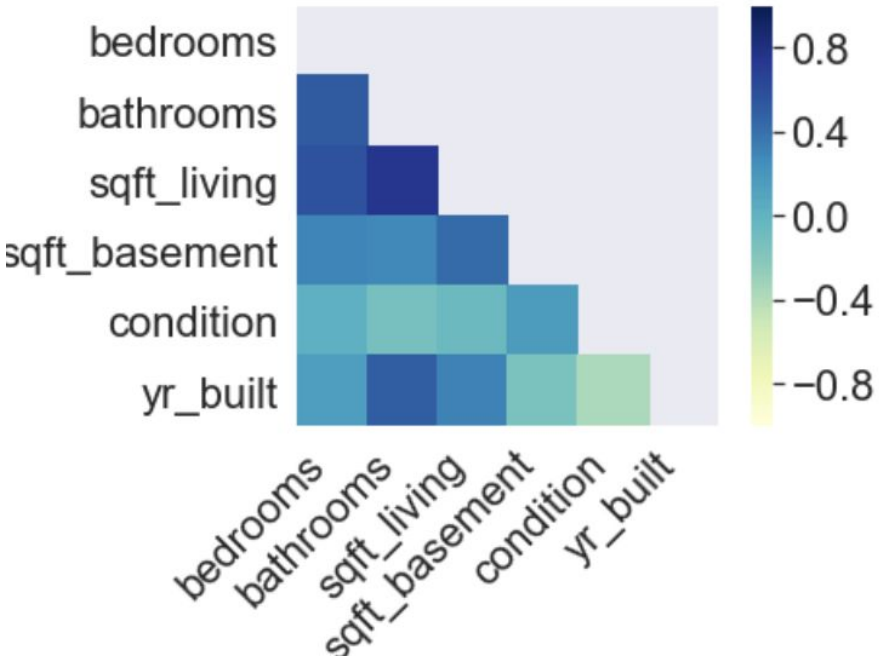


Figure 2. The heatmap plot showing the correlation between the input variables. From the plot, one can see the stronger correlation between footage and number of bathrooms, the age and number of bedrooms…. The heatmap is time-efficient showing all the correlation and helps identify potential elimination instead of plotting all the scatterplots[2]

From the heatmap (Appendix B), I can see that sqft_living is strongly correlated to the number of bathrooms, and yr_built is strongly correlated to the condition of the house. Sqft_living also has a strong correlation with the number of bedrooms and yr_built with the number of bathrooms. Further on, I will use this information to conduct the correlation plots and to eliminate these variables. Using both number bedrooms and bathrooms seem redundant to the plot because of multicollinearity.

The R -squared value for y_built vs sqft_living is 0.101 (Appendix B), showing there is no multicollinearity, so I want to leave both of the variables in the model.

The R-squared value for a number of bathrooms against bedrooms is 0.27, therefore, Pearson's r is =0.52 - it is a moderate positive correlation, so I will need to take out one of the

[2] #dataviz:The heatmap is used to identify multicollinearity. The caption explains the model, giving insights about the potential high Pearson's R values which are further explored. The scale of the map is adjusted, the redundant top part of the map is reduced to easen the comprehension.

variables. (Further examining the data, I learned that the number of bathrooms does not improve the R squared value of the multiple regression, therefore I eliminate this variable).

Checking the conditions for multiple regression

Checkpoint: 7 independent variables: 'bedrooms', 'sqft_living', ‘yr_built, 'sqft_basement’, ‘bedrooms’, ‘condition’, ‘bathrooms’.

According to OpenIntro (Diez, p. 387), before computing the multiple regression, I need to check the assumptions: the normality of the residuals, the constant variability of the residuals, independence of residuals, and the linear correlation of each of the variable to the response variable.

Linear correlation to the response variable:

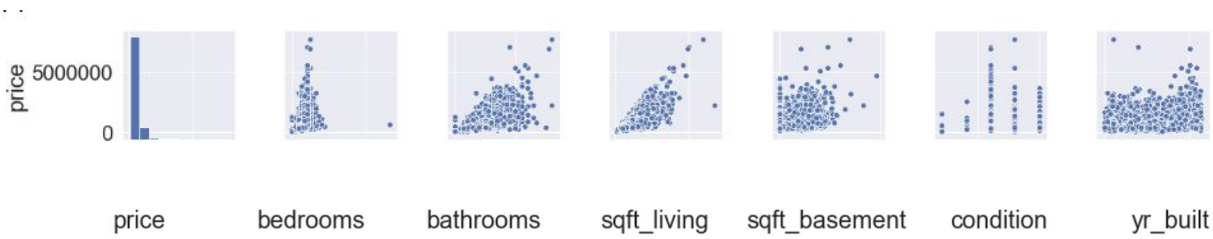


Figure 3: The part of the pair plot outputting the correlation plots of all the predictor variables against the response variable. Each of the plots represents a predictor variable plotted against the response variable - price.

Looking at the pair plot (top row prices against the rest of the predictor variables), (Appendix C) (Seaborn Pydata, n.d), a number of bedrooms against prices are not linear (Appendix C), also the residuals (average distance of the factual data from the model’s prediction) have the reverse hyperbolic pattern, most of the residuals are concentrated below with many outliers on top.

Sqft_living against prices is not ideal (Appendix C), because the data is somewhat heteroscedastic, but I will proceed with skepticism because the overall trend is linear, R squared is 0.49, and the residual distribution is normal.[3]

Sqft_basement against prices is not ideal either because it contains many 0 values (Appendix C). However, I will need to proceed with backward selection to make sure this variable contributes to the R-squared.

[3] #distributions: when checking the linear correlation for the linear regression, the normality of the residuals is the condition to be satisfied. The distribution is plotted and properly explained, relevant to the goal of the analysis. This confirmation allows to proceed with this variable.

Condition vs prices does not show any correlation at all (and R-square is 0), so I can take it out of the model. (Appendix C)[4]

The normality of the residuals:

Checkpoint: I have 4 remaining independent variables: 'bedrooms','sqft_living', 'sqft_basement', 'yr_built'.

As I can see from the normal probability plot (Appendix D), the residuals are not normal, not satisfying the condition for the model (the tails are too heavy). Hence, it is not reliable to use the model to predict the prices of the houses for sale. All the combinations of the predictor variable produce the unfit QQ plot. For the purpose of this assignment, I will continue inferencing from the model to perform the required steps of the assignment.

Constant Variance: The condition is not satisfied, either, the plot is not football-shaped, the data is heteroscedastic (Appendix D).

Independence: I do not know how the data was collected, the data was collected for a certain period of time (from May to May), therefore, there is no assurance that it is independent.

Multiple regression:

Backward elimination:

Using the backward elimination method (Diaz, p.379), I first computed all the variables in the regression model (Appendix D). I want to compare the R squared value of the equation with and without sqft_basement, because it might repeat the patterns of the sqft_living but is not shown on the feet map since not all of the houses have the basement. The R squared value is 0.507 without the basement variable dropping only by 0.01 points. This is an insignificant drop and it simplifies our model. Therefore, I will take out this variable.[5]

Checkpoint: 3 predictor variables: footage of the house, year the house was built in, and the number of bedrooms.

---

[4] #correlation:the linear correlation is used to check one of the conditions for multivariate regression.The R values are interpreted and the graph is interpreted to judge whether there is multicollinearity. Also, using the correlation is relevant to the goal of the analysis - we need to prove that there is a linear correlation between each of the predictor variable and a response variable.

[5] #regression: both linear and multiple regression models are used to select the variables for multivariate regression model. The conditions for the model are checked and are not satisfied. Therefore, the inference from the model is not reliable. The practical significance of the R-squared value is interpreted and the value of the model is explained, however with the reservation that the model did not satisfy the conditions and cannot be reliable.

Both of the P-values of the slopes are 0.00, confirming the idea that the variables are statistically significant to the model, which is consistent with the fact that if we take out any one of the two variables out of the model, the R-squared value will drop significantly. The practical significance: the R-squared value is 0.507. One can explain 50.7% of variance using these variables when predicting the price of the home based on the footage and the number of bedrooms.[6]

Confidence intervals.

Further on, I will construct confidence intervals for the slope of the regression line.

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.507
Model:	OLS	**Adj. R-squared:**	0.507
Method:	Least Squares	**F-statistic:**	1.110e+04
Date:	Sat, 25 Jan 2020	**Prob (F-statistic):**	0.00
Time:	19:16:24	**Log-Likelihood:**	-2.9997e+05
No. Observations:	21613	**AIC:**	5.999e+05
Df Residuals:	21610	**BIC:**	6.000e+05
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	7.947e+04	6604.764	12.032	0.000	6.65e+04	9.24e+04
bedrooms	-5.707e+04	2308.223	-24.723	0.000	-6.16e+04	-5.25e+04
sqft_living	313.9487	2.337	134.314	0.000	309.367	318.530

Figure 4: The output of the multiple regression function, showing all the data necessary to make inferences about the model and justifying practical significance - the R-squared value.

From this report, I can derive the 95% confidence interval:

For the number of bedrooms: [-6.16e+04;-5.25e+04].

For the footage of the house: [309.367;318.530].

It means that if the data collection was repeated many times, 95% of the confidence intervals created would contain the population mean.[7]

---

[6] #significance: p-value is checked to interpret statistical significance; practical significance - the R-squared value is introduced and explained - of how much variance of the prices of the housing can be explained by footage of the house and the number of bedrooms. However, it is addressed with skepticism because of the unsatisfied assumptions.

[7] #confidenceintervals: a population parameter is estimated based on the sample point estimate and interpreted. The 95% interval is chosen because it is specific enough. The computational software is used

Results and Conclusions

I have obtained the prediction model based on its footage and the number of bedrooms. The population data is the whole state of Washington, however, the results might differ based on the area, because the King county includes Seattle - a more densely populated area, implying that the prices might be higher. One need to be cautious to avoid hasty generalization. The coefficient might be introduced as a measure of adapting to the demand on the houses or the location of the houses for it to be generalized to other areas. This needs to be further hypothesized and researched. The model does not prove that the prices rise/drop is caused by the number of bedrooms and the footage, only that it is correlated, however, it seems like a logical explanation. To prove the causation, one needs to conduct further research.[8] The model, however, cannot be relied on: the conditions for conducting linear regression were not fulfilled - the induction is weak.[9] If no other model for predicting the model exists and the need for the model is sufficient, one may proceed with caution. Also, the model does not account for the extrapolation, and if the houses are out of price range of those examined in the report, then one must once again proceed with caution with the absence of better models.

---

to calculate the value and I have checked it using the by-hand calculation, but given the world limit, I could not have put it in the text

[8] #correlation: correlation does not imply causation - the concpet is explained but stated that it might be causation. It is not right to draw this conclusion at this stage, that is why more knowledge and evidence is required to make such a claim. One could hypothesize and make a new study on causation which would be useful to predict the price once building the houses.

[9] #induction: The strength and the type of induction is identified and justified: the house prices depend on the area and density of the population. And what changes need to be made for the induction to be made stronger: introduction of the coefficient as an example. Inferences of the regression are made with caution because the induction is weak.

Appendix A

Importing the libraries and the dataset. Computing descriptive statistics of the dataset.

```
# Import useful packages
import pandas
pandas.set_option('max_rows', 10)
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import statsmodels.api as statsmodels # useful stats package with regression functions
import seaborn as sns # very nice plotting package

# style settings
sns.set(color_codes=True, font_scale = 2)
sns.set_style("darkgrid")

# import and print data
#data = pandas.read_csv("soil_observations.csv") # requires file to be loaded in the directory
data = pd.read_csv("kc_house_data.csv")
data = pd.DataFrame(data)
data
```

```
analysis.describe()
```

	price	bedrooms	bathrooms	sqft_living	sqft_basement	condition	yr_built
count	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000
mean	5.400881e+05	3.370842	2.114757	2079.899736	291.509045	3.409430	1971.005136
std	3.671272e+05	0.930062	0.770163	918.440897	442.575043	0.650743	29.373411
min	7.500000e+04	0.000000	0.000000	290.000000	0.000000	1.000000	1900.000000
25%	3.219500e+05	3.000000	1.750000	1427.000000	0.000000	3.000000	1951.000000
50%	4.500000e+05	3.000000	2.250000	1910.000000	0.000000	3.000000	1975.000000
75%	6.450000e+05	4.000000	2.500000	2550.000000	560.000000	4.000000	1997.000000
max	7.700000e+06	33.000000	8.000000	13540.000000	4820.000000	5.000000	2015.000000

Appendix B

Plotting the heatmap to identify possible multicollinearity.

```
#plotting a heatmap to define the correlations
#definfing the reight potential variables
analysis1 = data[['bedrooms', 'bathrooms','sqft_living', 'sqft_basement', 'condition', 'yr_built']]
analysis = data[['price','bedrooms', 'bathrooms','sqft_living', 'sqft_basement', 'condition', 'yr_built']]
#reducing the half of the data - the top triangle fo the graph to take away the redundancy
mask = np.zeros(corr.shape, dtype=bool)
mask[np.triu_indices(len(mask))] = True

#plotting the heatmap
corr = analysis1.corr()
ax = sns.heatmap(corr, vmin=-1, vmax=1, center=0, cmap="YlGnBu", mask = mask, square=True)
ax.set_xticklabels(
    ax.get_xticklabels(),
    rotation=45,
    horizontalalignment='right');
```

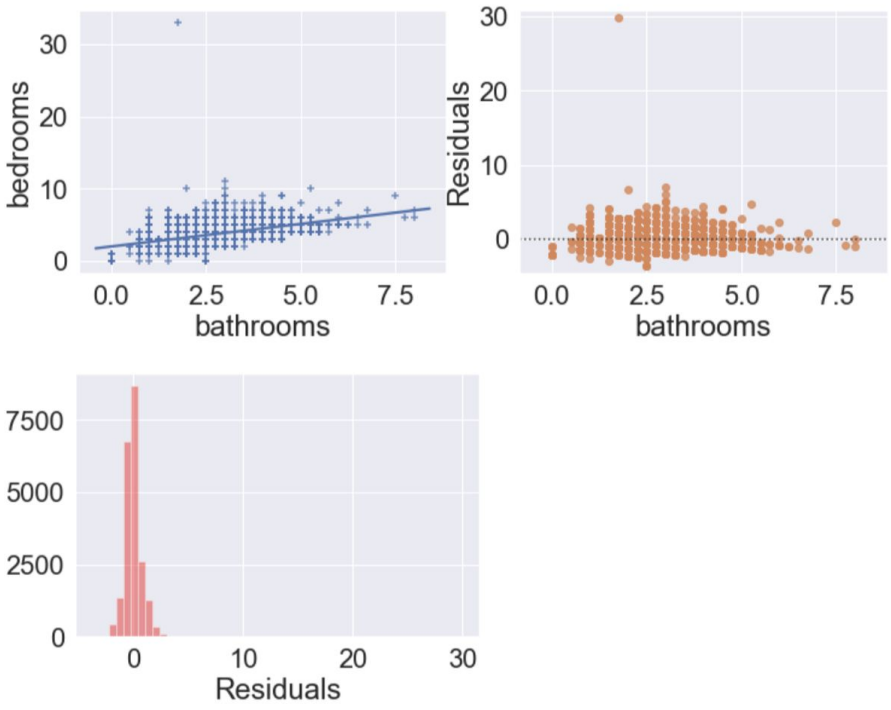
The code for the function for the simple regression.

```
#simple regression
def simple_regression(column_x, column_y):
    #this function computes the R-squared value, the regression line equation, and plots
    #dataviz crucial for confirming the conditions for the OSL
    #this dataviz is the normality and the constant variability of the residuals


    # fit the regression line
    X = statsmodels.add_constant(data[column_x])
    Y = analysis[column_y]
    regressionmodel = statsmodels.OLS(Y,X).fit() #ordinary least squares

    # necessary paramateres to construct the regression line, rounded to 2 decimal
    Rsquared = round(regressionmodel.rsquared,2)
    slope = round(regressionmodel.params[1],2)
    intercept = round(regressionmodel.params[0],2)

    # plotting the resifuals, and the scatterplot to check the conditions:
    fig, (ax1, ax2) = plt.subplots(ncols=2, sharex=True, figsize=(12,4))
    sns.regplot(x=column_x, y=column_y, data=data, marker="+", ax=ax1) # scatter plot for linearity
    sns.residplot(x=column_x, y=column_y, data=data, ax=ax2) # residual plot for constant variance
    ax2.set(ylabel='Residuals')
    ax2.set_ylim(min(regressionmodel.resid)-1,max(regressionmodel.resid)+1)
    plt.figure() # histogram for the normality of the residuals
    sns.distplot(regressionmodel.resid, kde=False, axlabel='Residuals', color='red') # histogram

    # outputing the necessary values.
    print("R-squared = ",Rsquared)
    print("Regression equation: "+column_y+" = ",slope,"* "+column_x+" + ",intercept)
```

Identifying multicollinearity in the number of bedrooms against bathrooms.

```
simple_regression ('bathrooms', 'bedrooms')
```

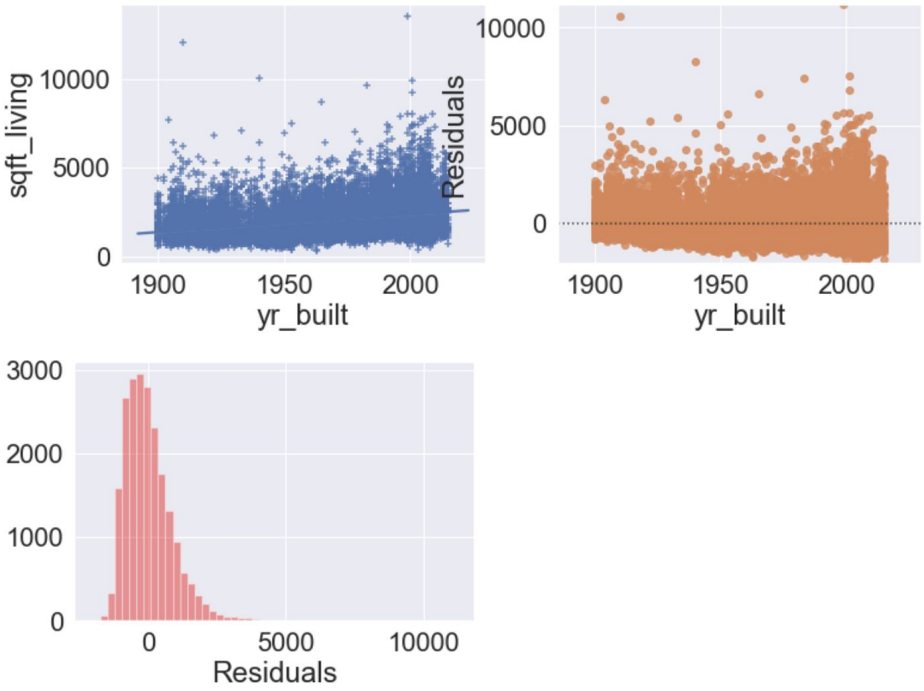
```
R-squared =  0.27
Regression equation: bedrooms =  0.62 * bathrooms +  2.05
```



Identifying multicollinearity in the year of building against the footage of the living area

```
simple_regression ('yr_built', 'sqft_living')
```

```
R-squared =  0.1
Regression equation: sqft_living =  9.94 * yr_built +  -17521.1
```



Appendix C

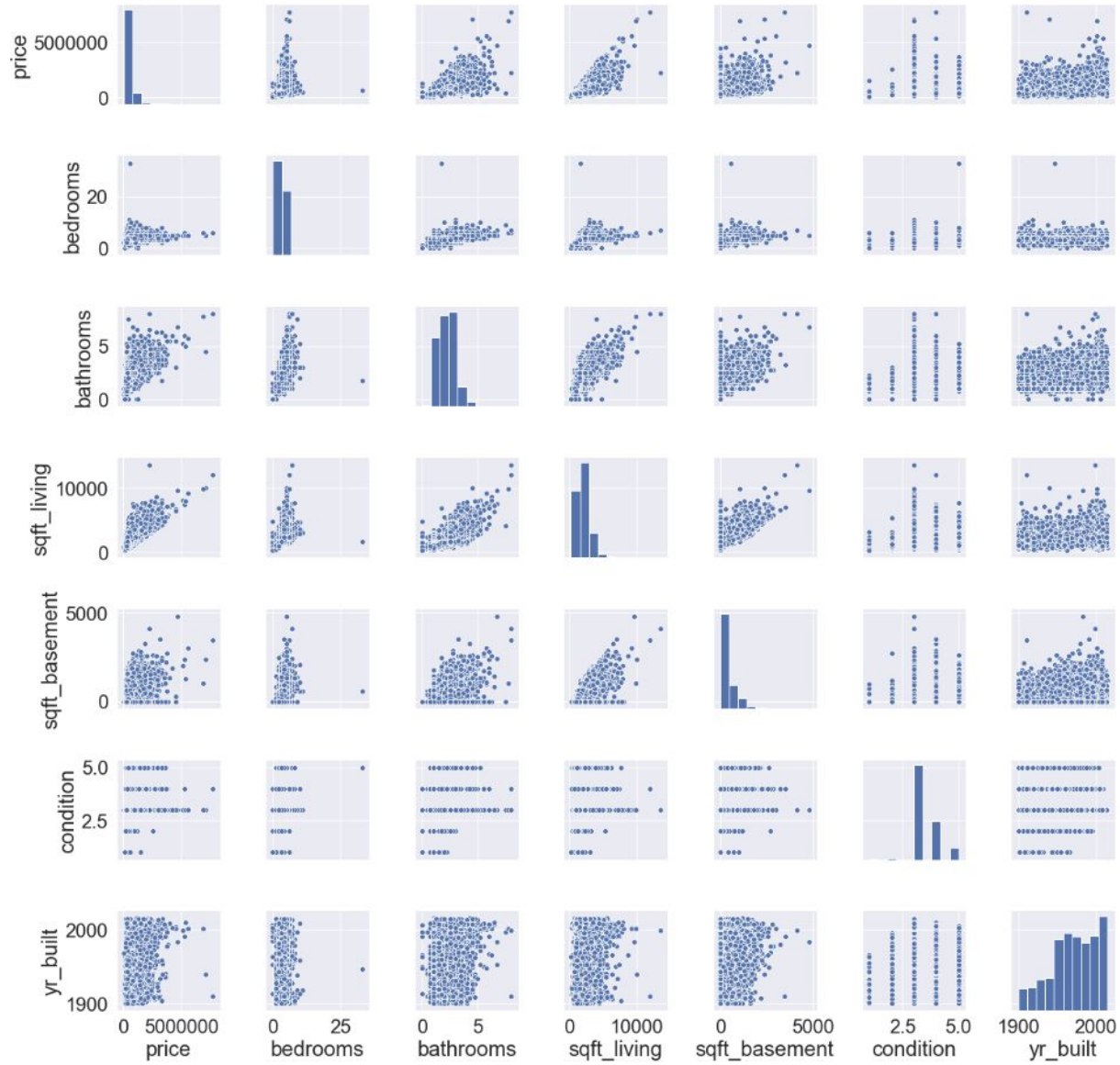
Checking the conditions for multiple regression:

1. The linear relationship with the response variable

   Pair plot 1: the year the house was built against the prices

```
#pairplot cretaes scatterplots for all the variables we are investigating.
#It is useful to prove the conditions for the OLS
analysis = data[['price','bedrooms', 'bathrooms','sqft_living', 'sqft_basement', 'conditic
sns.pairplot(analysis)

plt.plot()
```

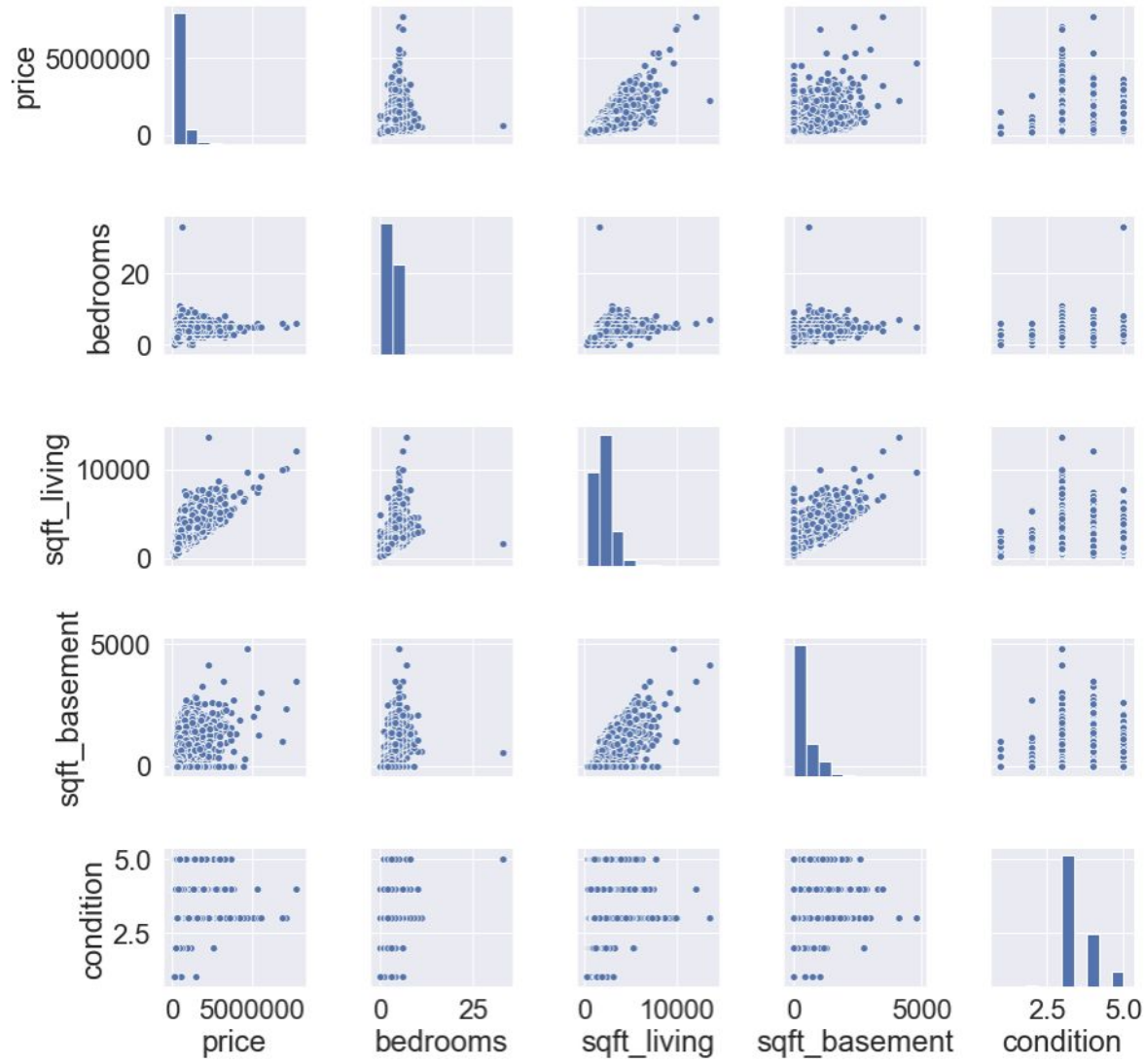
[ ]



Pair plot 2: without the bathrooms variable and without the year the house was built in.

```
#paorplotting to identify other non-linear correlations
analysis3 = data[['price','bedrooms','sqft_living', 'sqft_basement', 'condition']]
sns.pairplot(analysis3)

plt.plot()
```

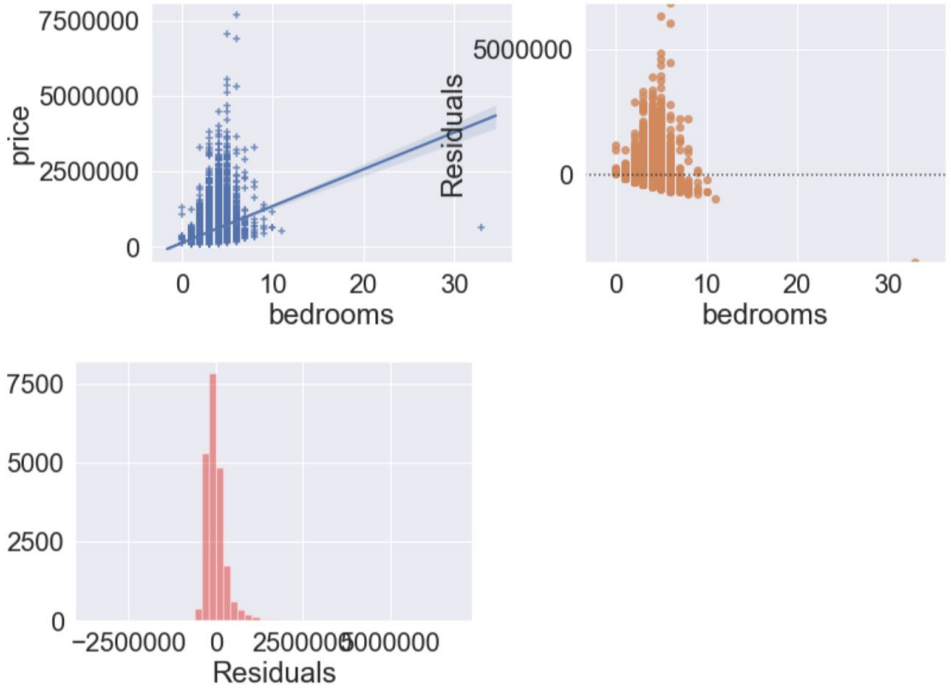
```
[]
```



Checking the linearity of the relationship of each of the graph:

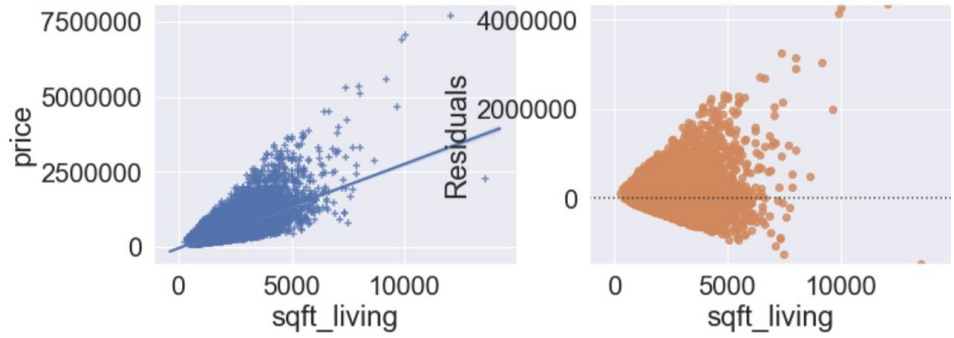
```
#checking the linearity of correlation conditions for OLS using the simple_regression function
simple_regression ('bedrooms', 'price')
```

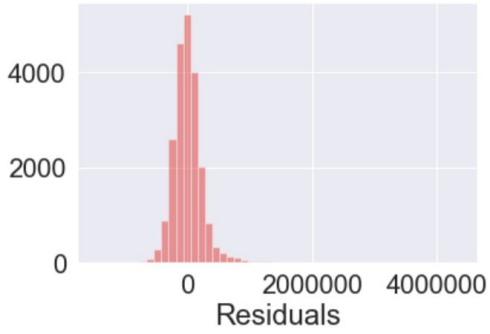
```
R-squared =  0.1
Regression equation: price =  121716.13 * bedrooms +  129802.36
```




```
#checking the linearity of correlation conditions for OLS using the simple_regression function
simple_regression ('sqft_living', 'price')
```

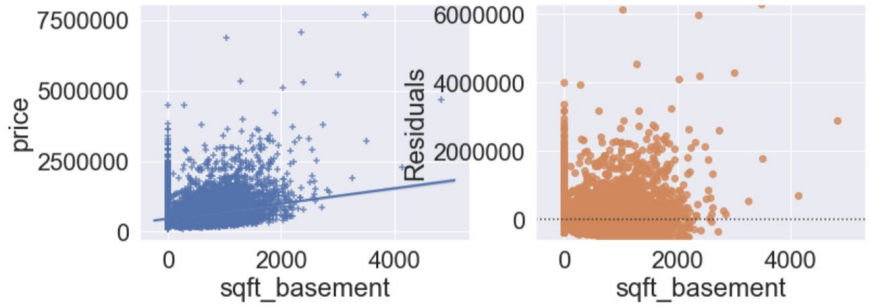
```
R-squared =  0.49
Regression equation: price =  280.62 * sqft_living +  -43580.74
```

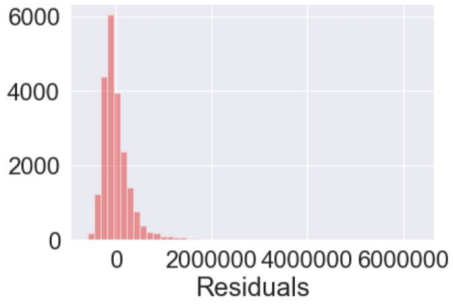




```
#checking the linearity of correlation conditions for OLS using the simple_regression function
simple_regression ('sqft_basement', 'price')
```

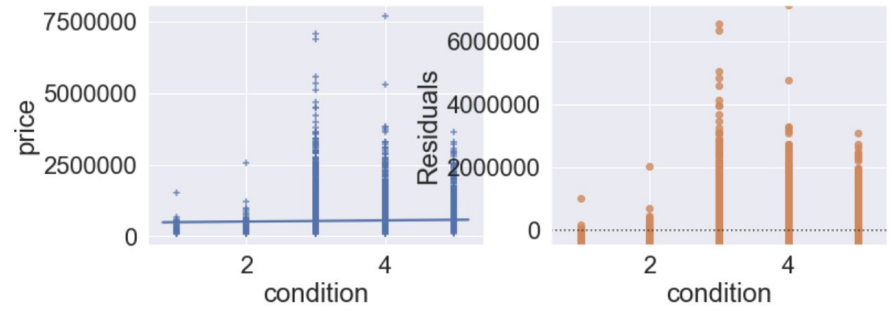
```
R-squared =  0.1
Regression equation: price =  268.61 * sqft_basement +  461784.85
```

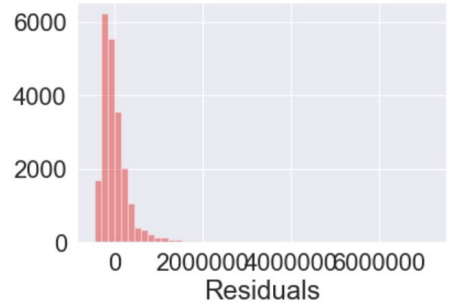





```
#checking the linearity of correlation conditions for OLS using the simple_regression function
simple_regression ('condition', 'price')
```

```
R-squared =  0.0
Regression equation: price =  20514.09 * condition +  470146.8
```





Appendix D

Multiple Regression Model

Function:

```
def mult_regression(column_x, column_y):
    ''' this function checks two conditions for building the model using the graphsand outputs the regression
    model with all the necessary stats for inferencing the results of the model.'''

    # If there is only one predictor variable, plot the regression line
    if len(column_x)==1:
        plt.figure()
        sns.regplot(x=column_x[0], y=column_y, data=data, marker="+",fit_reg=True,color='green')

    # define predictors X and response Y:
    X = analysis[column_x]
    X = statsmodels.add_constant(X)
    Y = analysis[column_y]

    # construct model:
    global regressionmodel
    regressionmodel = statsmodels.OLS(Y,X).fit() # OLS = "ordinary least squares"

    # residual plot:
    plt.figure()
    residualplot = sns.residplot(x=regressionmodel.predict(), y=regressionmodel.resid, color='blue')
    residualplot.set(xlabel='Fitted values for '+column_y, ylabel='Residuals')
    residualplot.set_title('Residuals vs Fitted values',fontweight='bold',fontsize=14)

    # Normal Probability plot:
    qqplot = statsmodels.qqplot(regressionmodel.resid,fit=True,line='45')
    qqplot.suptitle("Normal Probability (\"QQ\") Plot for Residuals",fontweight='bold',fontsize=14, color = 'red')
```

With the basement footage:

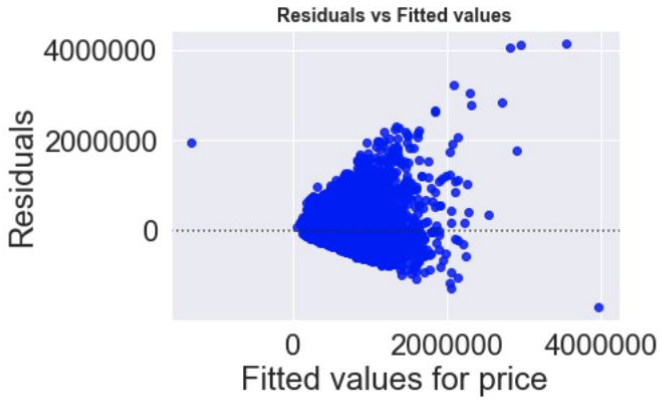
```
#building the model with footage of the basement
mult_regression(['bedrooms','sqft_living', 'sqft_basement'],'price')
regressionmodel.summary()
```

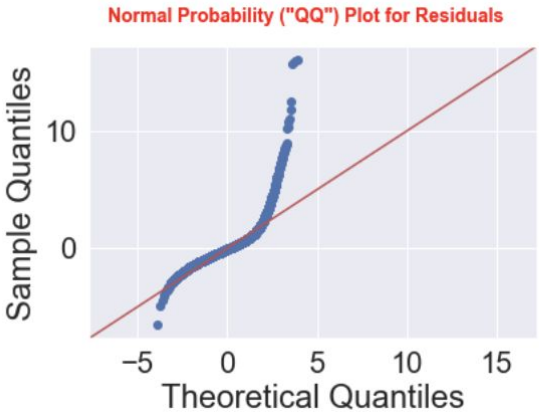
OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.508
Model:	OLS	**Adj. R-squared:**	0.508
Method:	Least Squares	**F-statistic:**	7426.
Date:	Sat, 25 Jan 2020	**Prob (F-statistic):**	0.00
Time:	19:13:57	**Log-Likelihood:**	-2.9995e+05
No. Observations:	21613	**AIC:**	5.999e+05
Df Residuals:	21609	**BIC:**	5.999e+05
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	8.547e+04	6673.321	12.807	0.000	7.24e+04	9.85e+04
bedrooms	-5.806e+04	2312.155	-25.111	0.000	-6.26e+04	-5.35e+04
sqft_living	308.9346	2.478	124.668	0.000	304.077	313.792
sqft_basement	26.6854	4.409	6.053	0.000	18.044	35.327

Omnibus:	14369.246	**Durbin-Watson:**	1.985
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	489010.614
Skew:	2.718	**Prob(JB):**	0.00
Kurtosis:	25.660	**Cond. No.**	9.07e+03





Without the basement footage:

```
]: #building the model with footage of the basement
   mult_regression(['bedrooms','sqft_living'],'price')
   regressionmodel.summary()
```

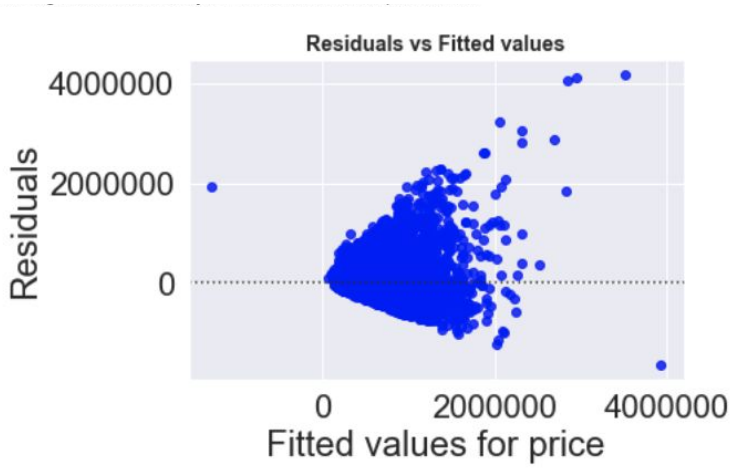
]:

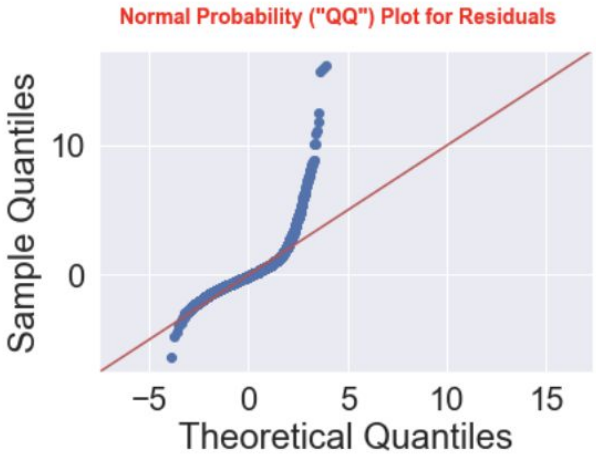
OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.507
Model:	OLS	**Adj. R-squared:**	0.507
Method:	Least Squares	**F-statistic:**	1.110e+04
Date:	Sat, 25 Jan 2020	**Prob (F-statistic):**	0.00
Time:	19:16:24	**Log-Likelihood:**	-2.9997e+05
No. Observations:	21613	**AIC:**	5.999e+05
Df Residuals:	21610	**BIC:**	6.000e+05
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	7.947e+04	6604.764	12.032	0.000	6.65e+04	9.24e+04
bedrooms	-5.707e+04	2308.223	-24.723	0.000	-6.16e+04	-5.25e+04
sqft_living	313.9487	2.337	134.314	0.000	309.367	318.530

Omnibus:	14423.033	**Durbin-Watson:**	1.986
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	492253.321
Skew:	2.732	**Prob(JB):**	0.00
Kurtosis:	25.732	**Cond. No.**	8.87e+03


Residuals vs Fitted values
4000000
2000000
0
Residuals
0
2000000
4000000
Fitted values for price


Normal Probability ("QQ") Plot for Residuals
Sample Quantiles
10
0
−5
0
5
10
15
Theoretical Quantiles

Reference

Diez, D. M., Barr, C. D., & Çetinkaya-Rundel Mine. (2016). OpenIntro statistics. United States: publisher not identified. Retrieved from https://drive.google.com/file/d/0B-DHaDEbiOGkc1RycUtIcUtIelE/view

Harlfoxem. House Sales in King County, USA. Retrieved from https://www.kaggle.com/harlfoxem/housesalesprediction/data

Seaborn Heatmap. (n.d.) Retrieved from: https://seaborn.pydata.org/generated/seaborn.heatmap.html