King County Housing Data

Presented By:

Rosario Fabian
Ashay Kargaonkar
Dharun Selvan

Table of Contents:

INTRODUCTION

The aim of this project is to identify which variables explain the house price in this specific city King County, Washington State, USA.

The King County is a county located in the U.S. state of Washington. The population was 2,233,163 in the 2018 census estimate, making it the most populous county in Washington, and the 12th-most populous in the United States. This is the reason we selected the king county's data housing dataset.

The dataset

The dataset consisted of historic data of houses sold between May 2014 to May 2015. We will predict the price of those houses in King County in 2020 and make people understand which factors are responsible for the price of property. The dataset was obtained from Kaggle. The Original dataset consisted of 21 variables and 21613 observations.
In order to complete our model used the following software in this project:

1. SAS 9.4
   For Creating models.
2. IBM SPSS SAS and Statistics 26
   For Pre-processing, Data Exploration.
3. MS EXCEL 2019
   For Pre-processing.
4. Jupyter Notebook
   For pre-processing using Python V3.7 with third party libraries.
5. Visual Studio 2019
   For creating application in Visual Basic using C sharp programming language that predicts the price of the house based on user input.
6. M.S. Word 2019
   For Report.

Description of variables:

1. ID:
   This variable gives the house's ID to recognize each observation.
2. Date:
   This variable tells when this observation was made. The value of date ranges from 1900 to 2015.
3. Price:
   This variable tells the current price of the house. This is a dependent variable who's value we are trying to predict.
4. Bedrooms:
   This variable tells the number of bedrooms that are present in a house. The value of bedrooms ranges from 0 to 11.
5. Bathrooms:
   This variable tells the number of bathrooms that are present in a house. The value of bedrooms ranges from 0 to 8.
6. Sqft_Living:
   This variable gives the carpet area of the house.
7. Sqft_Lot:
   This variable gives the total area of the house, which includes Sqft_Living
8. Floors:
   This variable tells the number of floors that are present in a house. The value of floors ranges from 1-3.5.
9. Waterfront:
   This variable tells if there is a waterbody near a house. It can be a lake, river etc. If the house has a waterfront view, then the value is 1 else it is 0.
10. View:
    This variable tells us if there is a view nearby like mountain, historical monument, etcetera. The value of View ranges from 0 to 4, where 0 means no views and 4 means that there are 4 views surrounding the house.
11. Condition:
    This variable tells the overall condition of the house. The value of condition ranges from 0 to 5, where 0 is the worst condition and 5 means the best condition.
12. Grade:
    This variable is the evaluation of construction materials and level of craftmanship used to build the houses. The value of grade ranges from 0 to 13, where 0 is a bad grade and 13 is the best.
13. Sqft_above:

This variable tells the total area excluding the basement area. It's the area above the basement.

14. Sqft_basement:
    This variable tells only the area of basement and ignores the rest of the area. Ignored area might include the areas of the floors above the basement or the area of terrace.
15. Yr_built:
    This variable tells the year of when the house was built. The value of Yr_built ranges from year 1900 to 2015
16. Yr_renovate:
    This variable tells the year when the house was renovated.
17. Zipcode:
    This variable tells the zip code of where the house is situated.
18. Lat:
    This variable gives the latitude of the house location.
19. Long:
    This variable gives the longitude of the house location.
20. Sqft_living15:
    This variable gives the average house square footage of 15 closest houses.
21. Sqft_lot15:
    This variable gives the average of the lot square footage of the 15 closest house.

Pre-Processing

Multiple tools were used to do data pre-processing. Those tools are IBM, SAS SPSS Statistics 26, Microsoft Excel 2019 and Jupiter Notebook (Python 3.7).

Missing Values Analysis

```
In [1]: import pandas as pd;
        import numpy as np;

In [2]: data=pd.read_csv(r'C:\Users\dharu\Documents\Studies\DADR\Project\final2.csv',header=0,delimiter=',')

In [3]: data
```

Out[3]:

	no	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	...	yr_renovated_d	log_sqft_lot	price_t	sqft_living
0	0	221900	3	1.00	1180	5650	1.0	0	0	3	...	0	8.639411	221900	1180
1	1	538000	3	2.25	2570	7242	2.0	0	0	3	...	1	8.887653	538000	2570
2	2	180000	2	1.00	770	10000	1.0	0	0	3	...	0	9.210340	180000	770
3	3	604000	4	3.00	1960	5000	1.0	0	0	5	...	0	8.517193	604000	1960
4	4	510000	3	2.00	1680	8080	1.0	0	0	3	...	0	8.997147	510000	1680
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
21607	21608	360000	3	2.50	1530	1131	3.0	0	0	3	...	0	7.030857	360000	1530
21608	21609	400000	4	2.50	2310	5813	2.0	0	0	3	...	0	8.667852	400000	2310
21609	21610	402101	2	0.75	1020	1350	2.0	0	0	3	...	0	7.207860	402101	1020
21610	21611	400000	3	2.50	1600	2388	2.0	0	0	3	...	0	7.778211	400000	1600
21611	21612	325000	2	0.75	1020	1076	2.0	0	0	3	...	0	6.981006	325000	1020

21612 rows × 33 columns

The above snapshot shows how we imported the file is Jupiter Notebook.

```
In [23]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 17 columns):
id               21613 non-null int64
date             21613 non-null object
price            21613 non-null float64
bedrooms         21613 non-null int64
bathrooms        21613 non-null float64
sqft_living      21613 non-null int64
sqft_lot         21613 non-null int64
floors           21613 non-null float64
waterfront       21613 non-null int64
view             21613 non-null int64
condition        21613 non-null int64
grade            21613 non-null int64
sqft_above       21611 non-null float64
sqft_basement    21613 non-null int64
yr_built         21613 non-null int64
yr_renovated     21613 non-null int64
zipcode          21613 non-null int64
dtypes: float64(4), int64(12), object(1)
memory usage: 2.8+ MB
```

Then we checked the data types of each variables.

```
In [24]: data.isnull().sum()
```

```
Out[24]: id              0
         date            0
         price           0
         bedrooms        0
         bathrooms       0
         sqft_living     0
         sqft_lot        0
         floors          0
         waterfront      0
         view            0
         condition       0
         grade           0
         sqft_above      2
         sqft_basement   0
         yr_built        0
         yr_renovated    0
         zipcode         0
         dtype: int64
```

The above snapshot shows the variable **sqft_above** have 2 missing value.

```
In [25]: mean_sqft_ab=int(data.sqft_above.mean())
```

```
In [26]: mean_sqft_ab
```

```
Out[26]: 1788
```

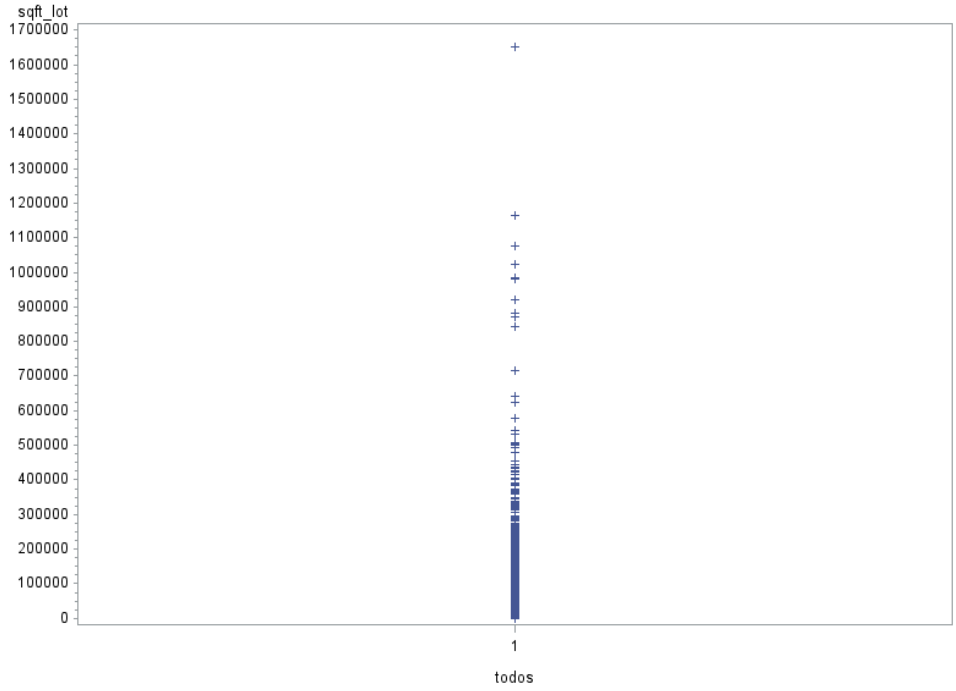
```
In [27]: data.sqft_above.fillna(mean_sqft_ab,axis=0,inplace=True)
```

So, we planned to fill those missing value with different kinds of methodologies, means, boixplot anaylis

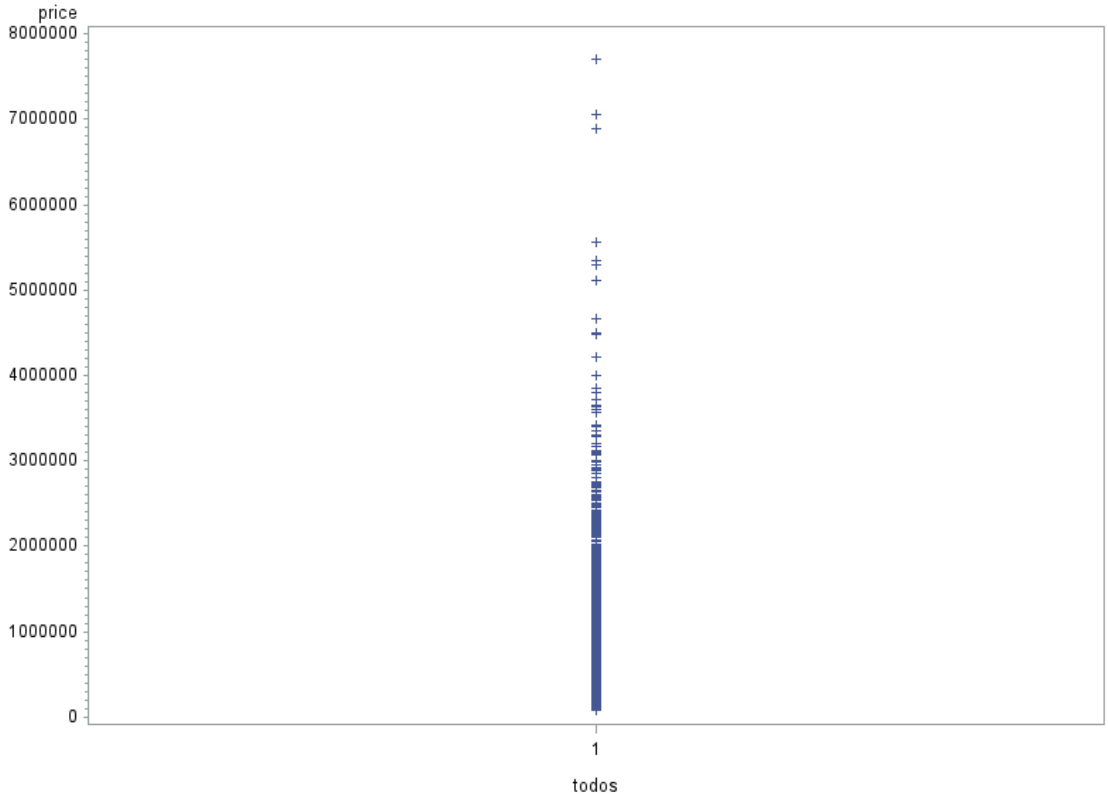
Analysis of Outliers

It was made in SAS

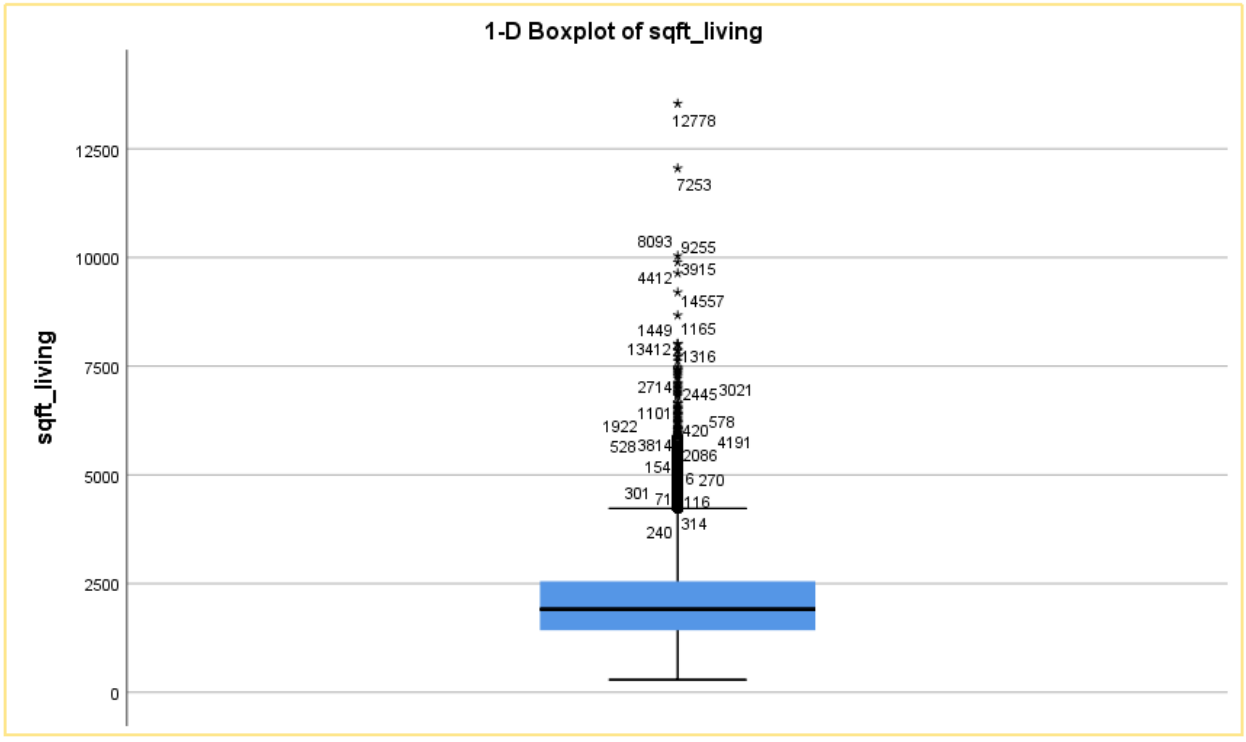
Variable: Size of buildings

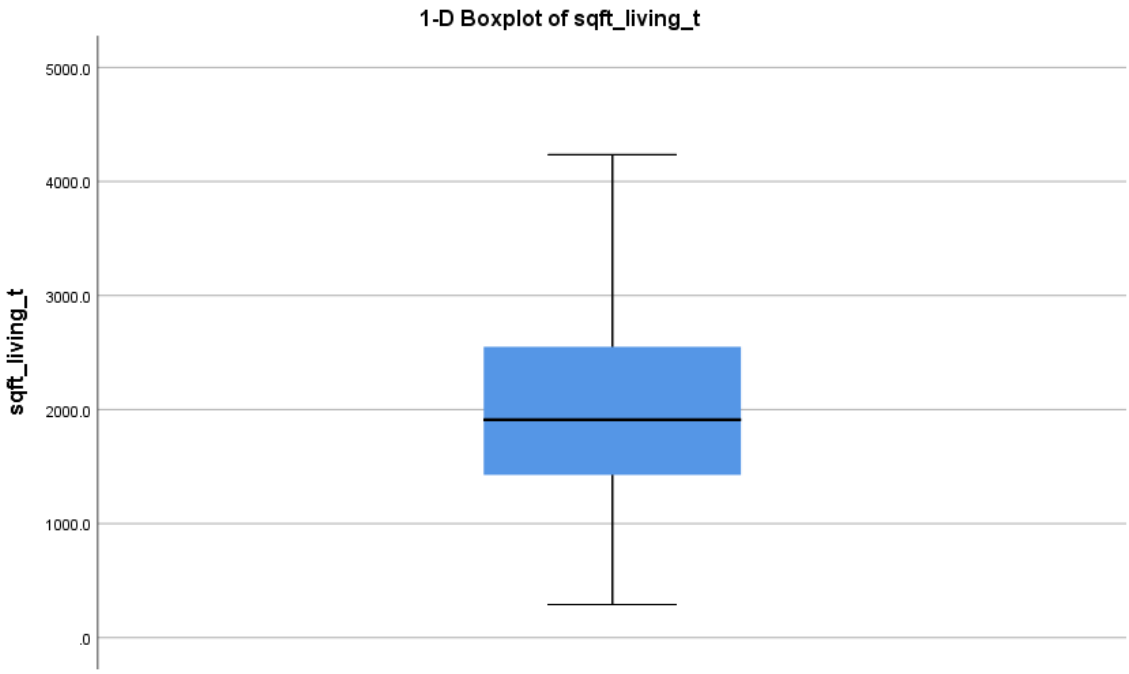


Variable: Price

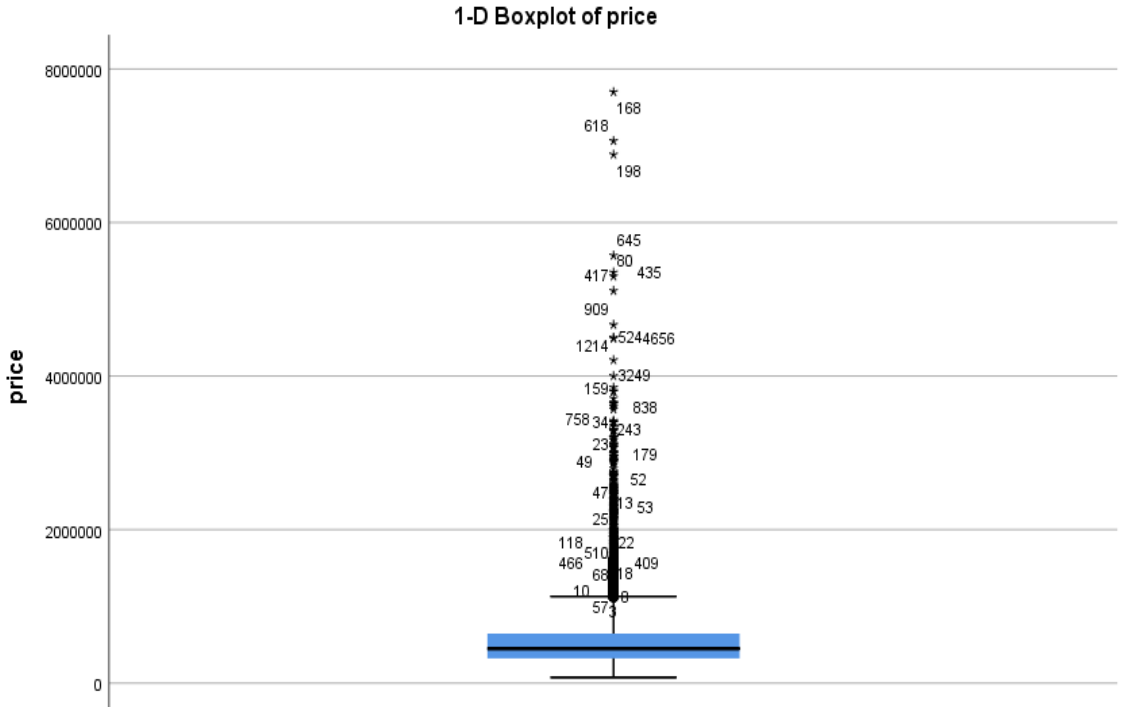

price
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
1
todos

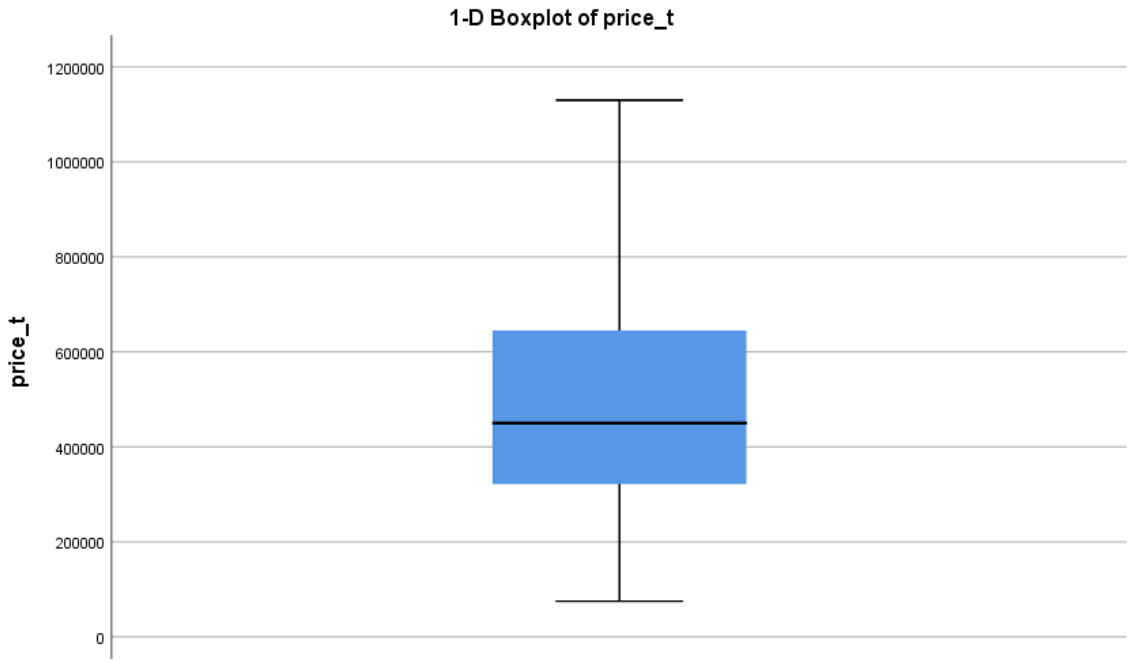
Now In spss,


1-D Boxplot of sqft_living
sqft_living
12500
10000
7500
5000
2500
0
12778
7253
8093
9255
3915
4412
14557
1449
1165
1316
2445
3021
1922
578
420
4191
2086
270
116
240
314

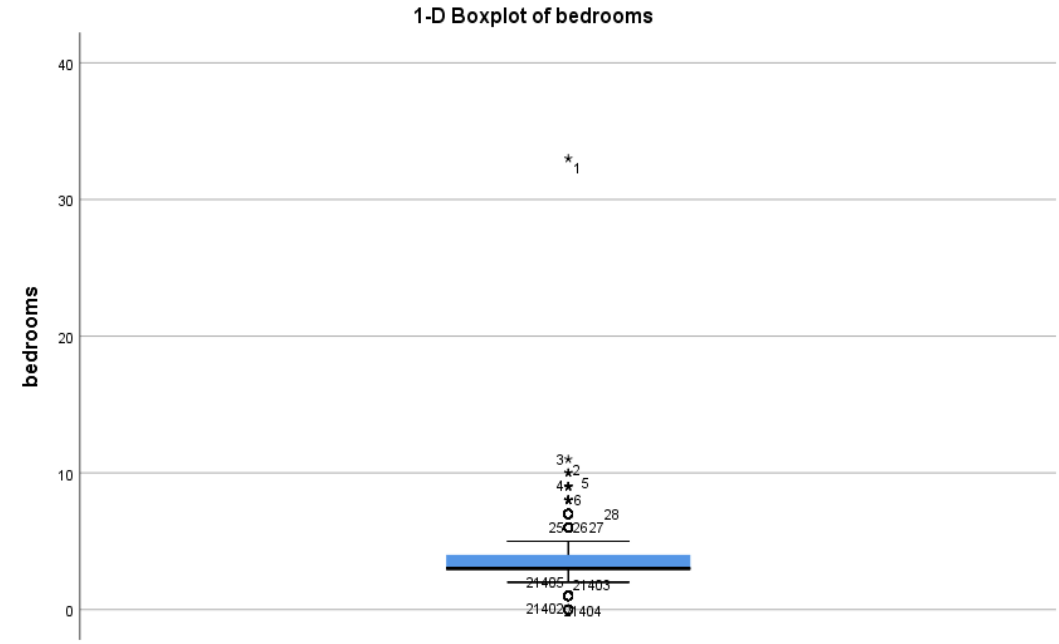


The above snapshot of **sqft_living_t** shows that there are no outliers after transformation.

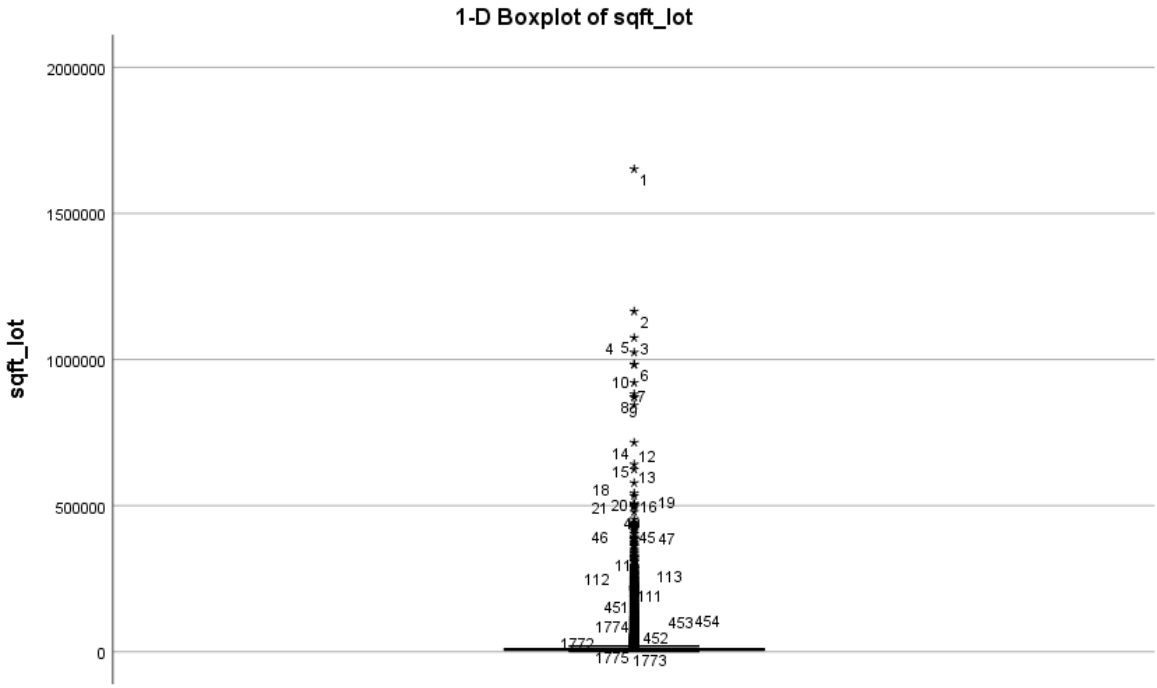




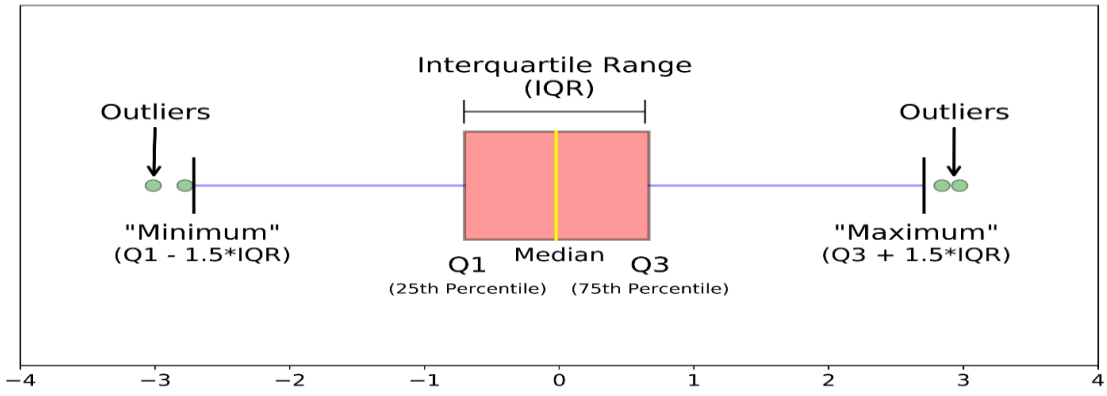
The above snapshot of **price_t** shows that there are no outliers after t ransformation.
Then we analysed the outliers for bedrooms.



Then we looked for outliers in **sqft_lot.**



Decision: In four variables, we replaced outliers following the next formula. In other cases the specific observations were eliminated. The qualitative variables did not have these problems.



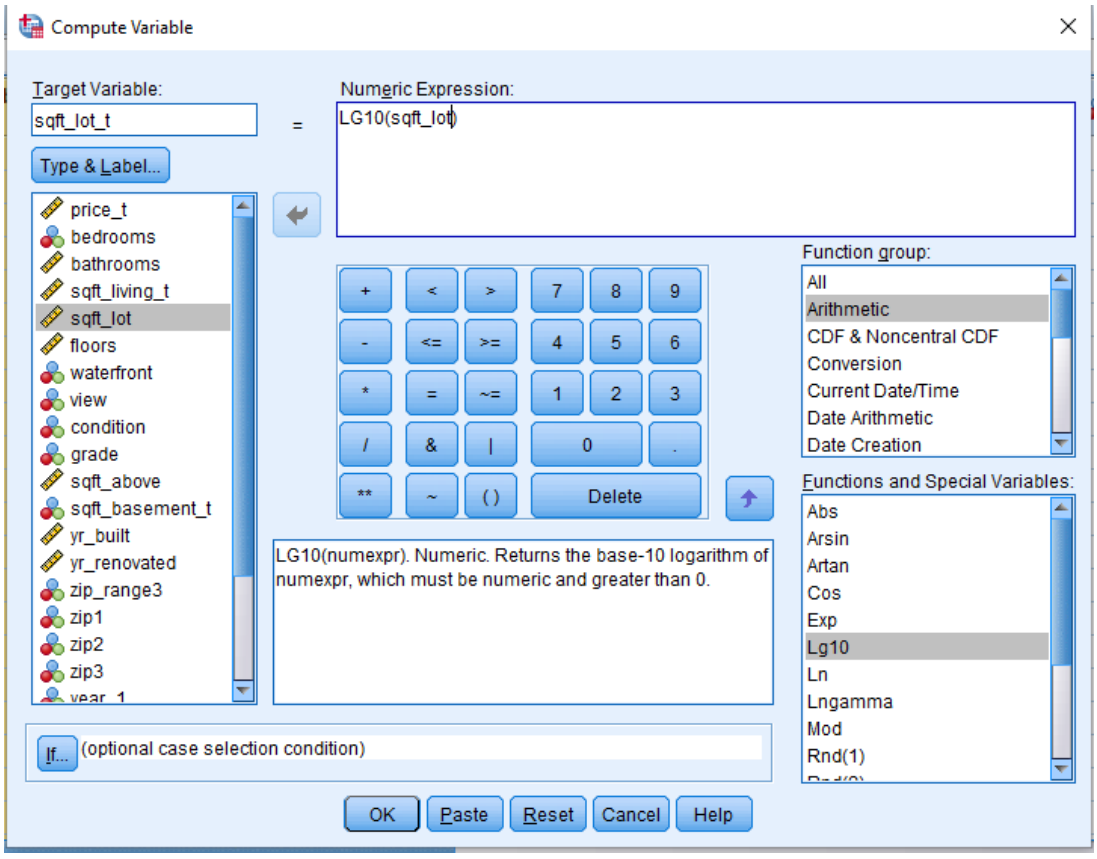
Main changes:

Price: The variable was transformed the maximum value from $770000000 to $1129574 which is calculated using the same formula and saved it in a new variable called **price_t**.

Size of building : The boxplot shows that there is many outliers. We found that we had almost 2500+ outliers. So we planned to do log transformation to the it.

Bedrooms: We deleted only one row that has 33 bedrooms and planned to keep all other outliers.

It was made by SPSS



These dataset do not have string values, therefore we did not have string – processing.

Exploration

After the first – Preprocessing, we started to analyze each variable again in SAS, we check main statistics values and distributions in order to understand better the dataset and then creating new variables.

Variable Size

N	20279	**Sum Weights**	20279
Mean	1951.22087	**Sum Observations**	39568808
Std Deviation	732.860753	**Variance**	537084.883
Skewness	0.58368018	**Kurtosis**	-0.1358301
Uncorrected SS	8.80985E10	**Corrected SS**	1.0891E10
Coeff Variation	37.5590875	**Std Error Mean**	5.1463367

Basic Statistical Measures

Location		**Variability**	
Mean	1951.221	**Std Deviation**	732.86075
Median	1850.000	**Variance**	537085
Mode	1300.000	**Range**	3940
		Interquartile Range	1020

Tests for Location: Mu0=0

Test		**Statistic**		**p Value**
Student's t	**t**	379.1475	**Pr > |t|**	<.0001
Sign	**M**	10139.5	**Pr >= |M|**	<.0001
Signed Rank	**S**	1.0281E8	**Pr >= |S|**	<.0001

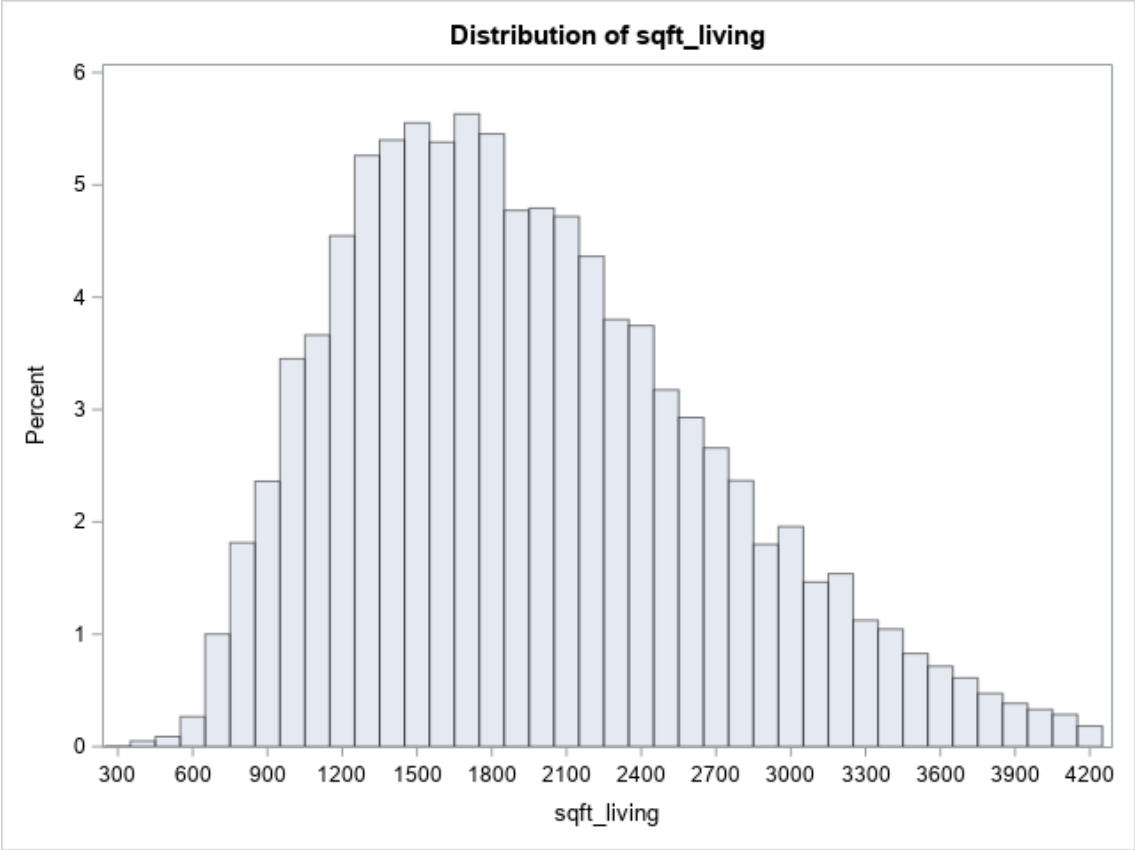
Quantiles (Definition 5)

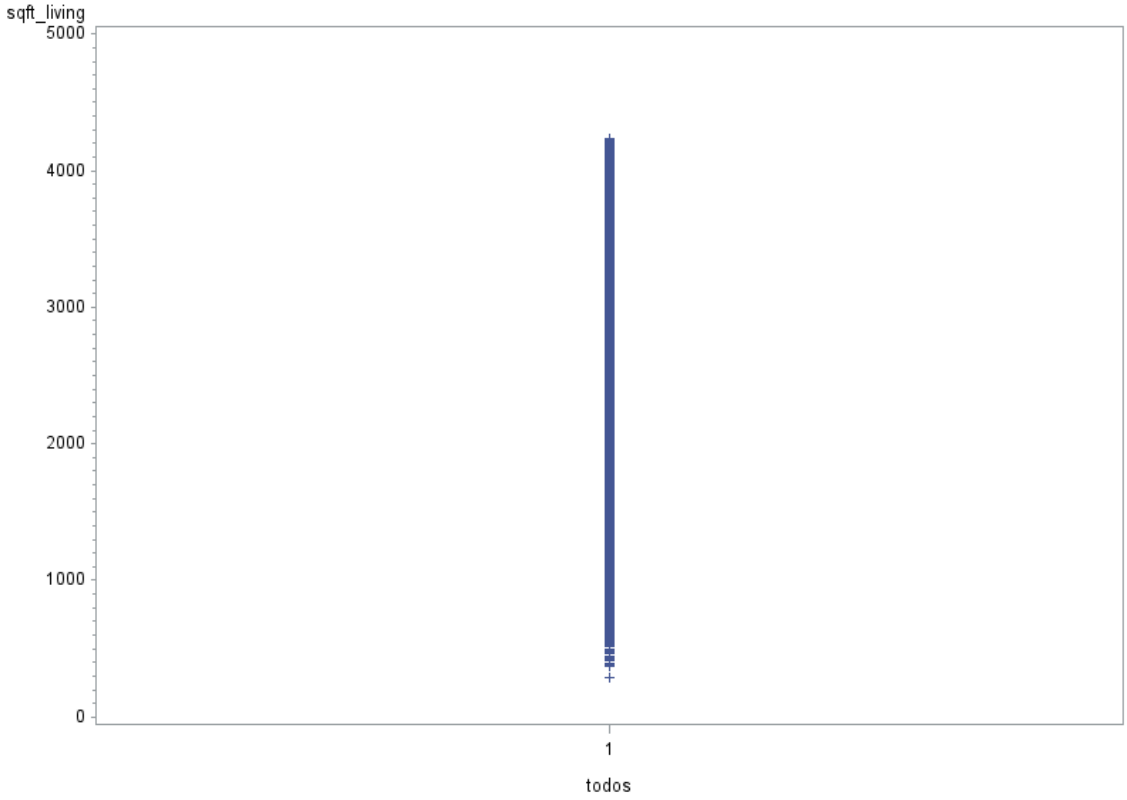
Level	Quantile
100% Max	4230
99%	3890
95%	3330
90%	3000
75% Q3	2420
50% Median	1850
25% Q1	1400
10%	1070
5%	920
1%	720
0% Min	290

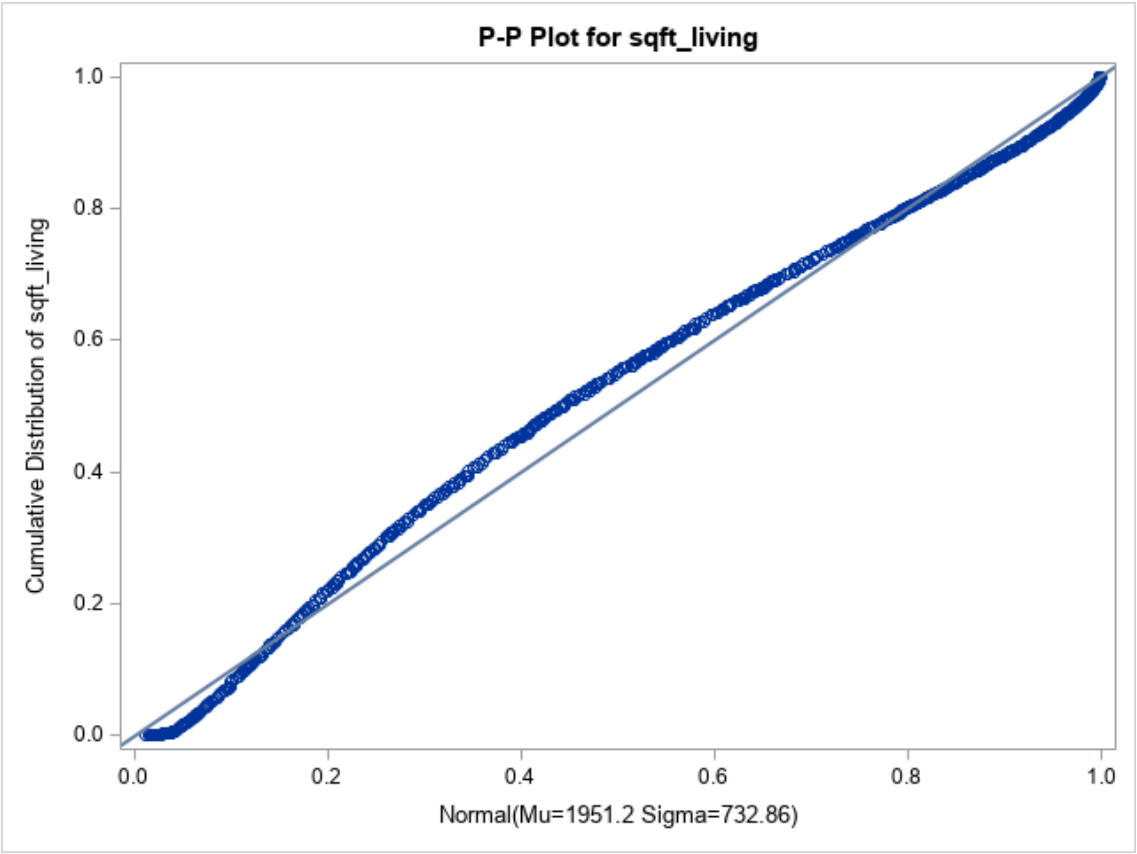
Extreme Observations

Lowest		Highest	
Value	**Obs**	**Value**	**Obs**
290	20279	4230	2
370	20278	4230	3
380	20277	4230	4
384	20276	4230	5
390	20275	4230	6

Variable xxxx




sqft_living
5000
4000
3000
2000
1000
0
1
todos


P-P Plot for sqft_living
Cumulative Distribution of sqft_living
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Normal(Mu=1951.2 Sigma=732.86)

The UNIVARIATE Procedure
Variable: sqft_lot

Moments

N	21613	**Sum Weights**	21613
Mean	15106.9676	**Sum Observations**	326506890
Std Deviation	41420.5115	**Variance**	1715658774
Skewness	13.060019	**Kurtosis**	285.07782
Uncorrected SS	4.20113E13	**Corrected SS**	3.70788E13
Coeff Variation	274.181508	**Std Error Mean**	281.746112

Basic Statistical Measures

Location		**Variability**	
Mean	15106.97	**Std Deviation**	41421
Median	7618.00	**Variance**	1715658774
Mode	5000.00	**Range**	1650839
		Interquartile Range	5648

Tests for Location: Mu0=0

Test		**Statistic**	**p Value**	
Student's t	**t**	53.61908	**Pr > |t|**	<.0001
Sign	**M**	10806.5	**Pr >= |M|**	<.0001
Signed Rank	**S**	1.1679E8	**Pr >= |S|**	<.0001

Quantiles (Definition 5)

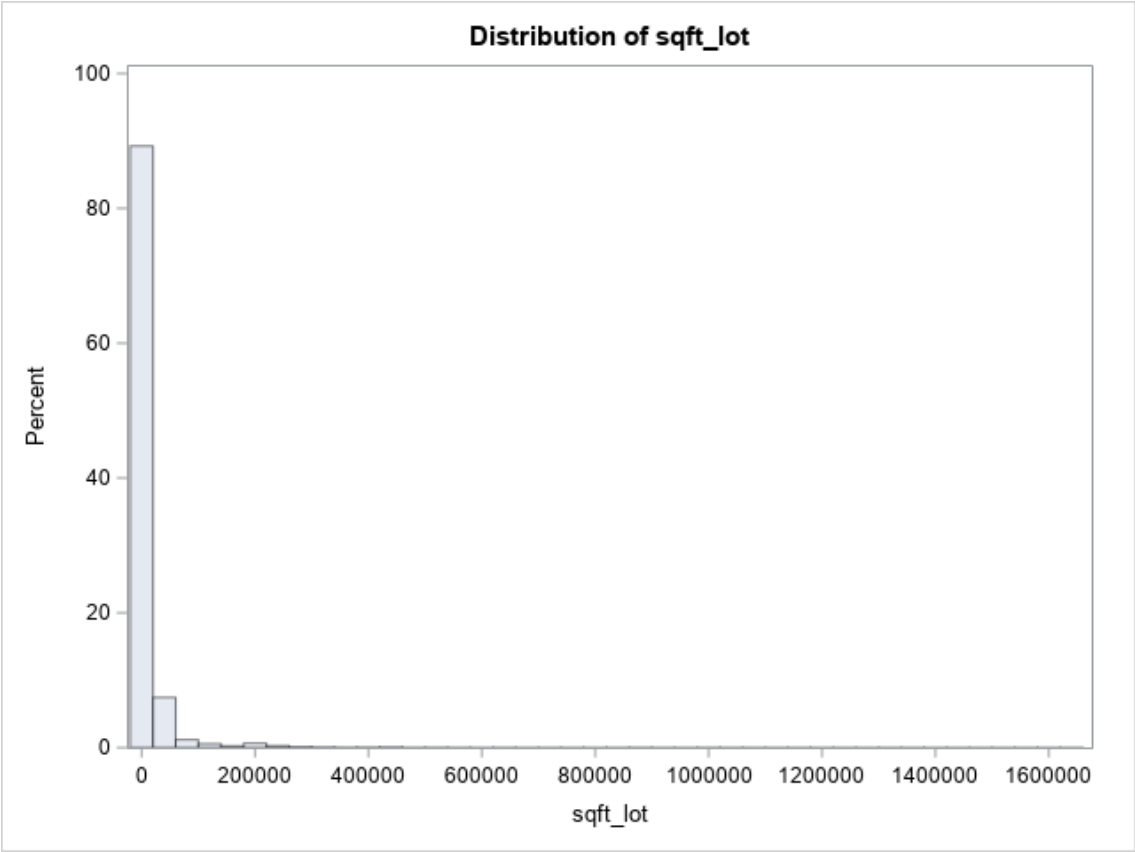
Level	**Quantile**
100% Max	1651359
99%	213008
95%	43350
90%	21399
75% Q3	10688

Quantiles (Definition 5)

Level	Quantile
50% Median	7618
25% Q1	5040
10%	3322
5%	1800
1%	1013
0% Min	520

Extreme Observations

Lowest		Highest	
Value	**Obs**	**Value**	**Obs**
520	15744	982998	3950
572	5827	1024068	7770
600	7590	1074218	7648
609	3453	1164794	17320
635	20605	1651359	1720



Creating the new variables:

Variable Zipcode: This variable is like a small cluster of city, we divide in 4 zones according to zipcode.

a. Binning was done by selecting Visual Binning option in Analyse menu. 3 cuts were given with equal width.

```
* Visual Binning.
*zipcode.
RECODE  zipcode (MISSING=COPY) (LO THRU 98033=1) (LO THRU 98065=2) (LO THRU 98118=3) (LO THRU HI=4)
    (ELSE=SYSMIS) INTO zip_range3.
VARIABLE LABELS  zip_range3 'zipcode (Binned)'.
FORMATS  zip_range3 (F5.0).
VALUE LABELS  zip_range3 1 '' 2 '' 3 '' 4 ''.
VARIABLE LEVEL  zip_range3 (ORDINAL).
EXECUTE.
```

b. Dummy variables are created using selecting Decode Variables into Different Variable option in Analyse menu. 3 dummy variables are created as **zip1, zip2, zip3.**

Variable Year : It is is a date variable, it was changed in order to have number of years of building.

- **year_1** for buildings built from 2013 to 2015.
- **Year_2** for buildings built from 2003 to 2012. It was done using SPSS and it's log is shown below.

```
DATASET NAME DataSet4 WINDOW=FRONT.
SORT CASES BY yr_built (A).
SORT CASES BY yr_built (D).
RECODE yr_built (2015=1) (2014=1) (2013=1) INTO year_1.
VARIABLE LABELS  year_1 'year_1'.
EXECUTE.
RECODE yr_built (2003 thru 2012=1) (ELSE=0) INTO year_2.
VARIABLE LABELS  year_2 'year_2'.
EXECUTE.
```

Dummy : Variable View: It is converted into dummy variable as **view_t**. 0 means no views and 1 means the house has view(s).

Variable Renovated: It is a good nice smart variable, that we created.

In original data set, **yr_renovated** has 0 as the home that hasn't renovated. With that variable we created 2 new variables as **yr_renovated_t** and **yr_renovated_d.**

- **yr_renovated_t** is the age of the building from renovation. This was done in Ms Excel using the formula =2020-yr_renovated. (Only for those renovated houses. Remaining will be zero.)
- **yr_renovated_d** is a dummy variable. 0 means it's not renovated and 1 means it's renovated. This was done using SPSS.

yr_renovated	yr_renovated_age	yr_renovated_d
1994	21	1
2011	4	1
2014	1	1
2002	13	1
2000	15	1
2002	13	1
1982	33	1
2005	10	1
2014	1	1
2014	1	1
1992	23	1
1989	26	1
1999	16	1
1997	18	1
2006	9	1
1987	28	1
1995	20	1
2003	12	1

The above screenshot shows the output for yr_renovation.

Then using **sqft_basement** variable, we created a new dummy variable **sqft_basement_t**. 0 means no basement and 1 means the house has basement. Screenshot with the converted dummy variable along with original value is shown below. This was also done usind SPSS's Decode variable into different value is used.

sqft_basement	sqft_basement_t
0	0
180	1
0	0
0	0
0	0
0	0
0	0
340	1
390	1
1130	1
0	0
0	0
500	1
0	0
0	0
750	1
1510	1
0	0

More variables:

We created a new variable called **sqft_diff** using **sqft_built** and **sqft_lot**. It is the total difference between total area of the lot and total area of the building. It was done using excel and the formula is =sqft_lot-sqft_built.

yr_built	yr_age
1998	22
1940	80
2014	6
1964	56
1986	34
1914	106
1968	52
2004	16
2008	12
1960	60
1948	72
1965	55
1984	36
1990	30
1944	76
1966	54
1990	30
1963	57
2008	12

We also created a new variable called **yr_age.** It is the total age of the building from built. It was calculated using the formula =2020-yr_built.

sqft_living	sqft_lot	sqft_diff
2480	10800	8320
1430	5200	3770
2870	4712	1842
1860	7700	5840
1250	7029	5779
1480	6500	5020
970	8874	7904
1400	1657	257
1500	1119	-381
2260	12500	10240
1590	11222	9632
2340	12443	10103
1810	7350	5540
1946	17786	15840
790	7153	6363
2250	8970	6720
4230	21455	17225
1170	9848	8678

After these analysis, the new variables were created and also preprocessing again in order to have a clean dataset

Variable	Description
Number of years of building	This variable was a year in dataset, now it is a number (number of years)
How old the building was renovated	Count the number of years it was renovated, but for news building, it is 0
Logathm size	Logarithm function commands this relation with price.
Segmentation geographic	Created cluster of places according to the zipcode
Dummies	New scores because there are many groups
Years of building	How many years has the building
Years of renovation	How many yearswas renovated
Zip code	Geographic Zone
View	With or not view

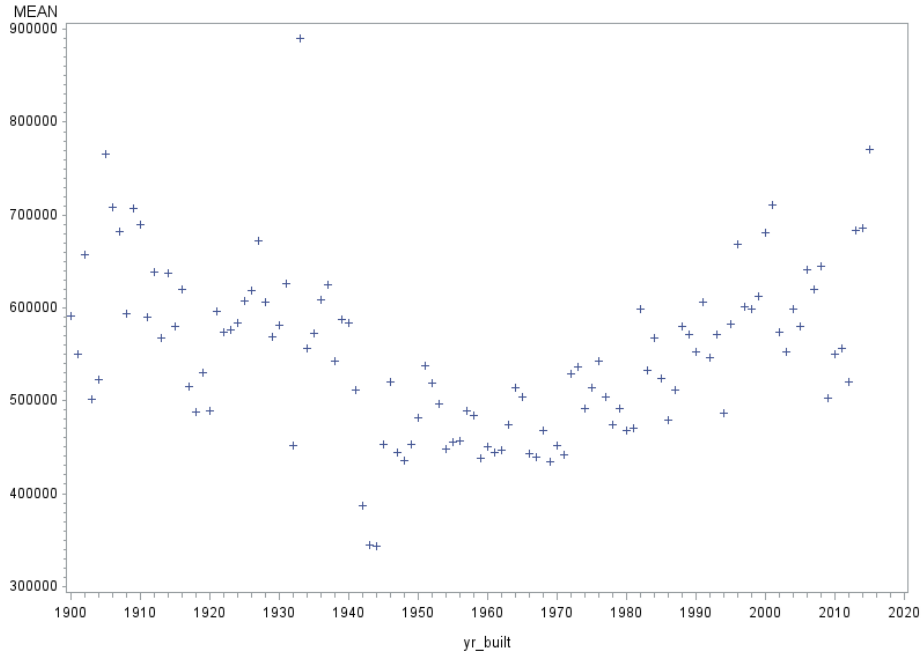
Analysis Bivariate:

We analyze the relation function between price and all independent variables in order to find tendency between them.

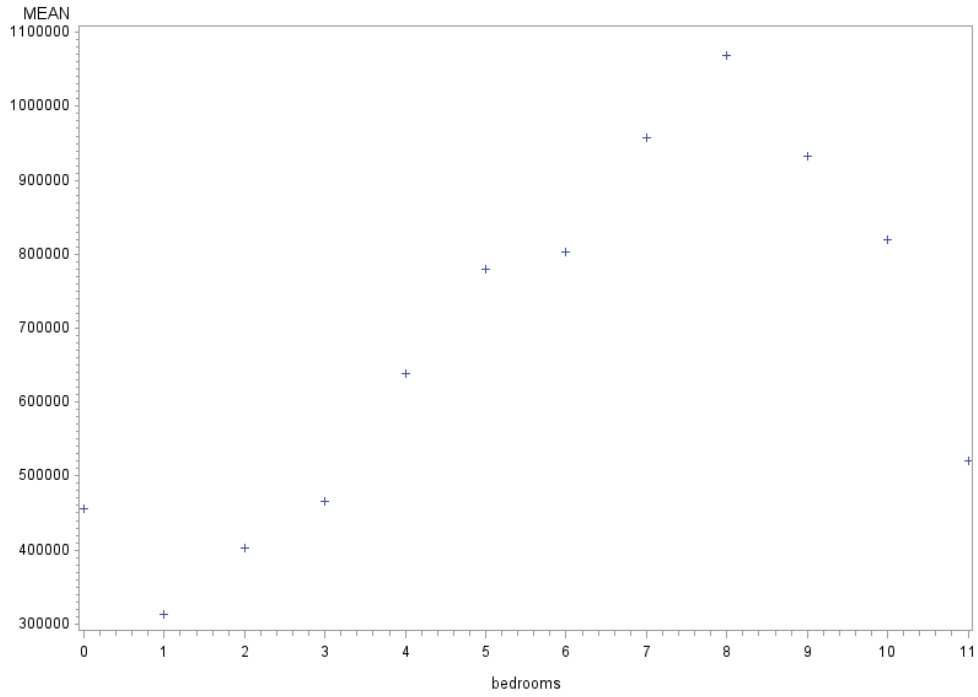
Variable:

x: Years of building

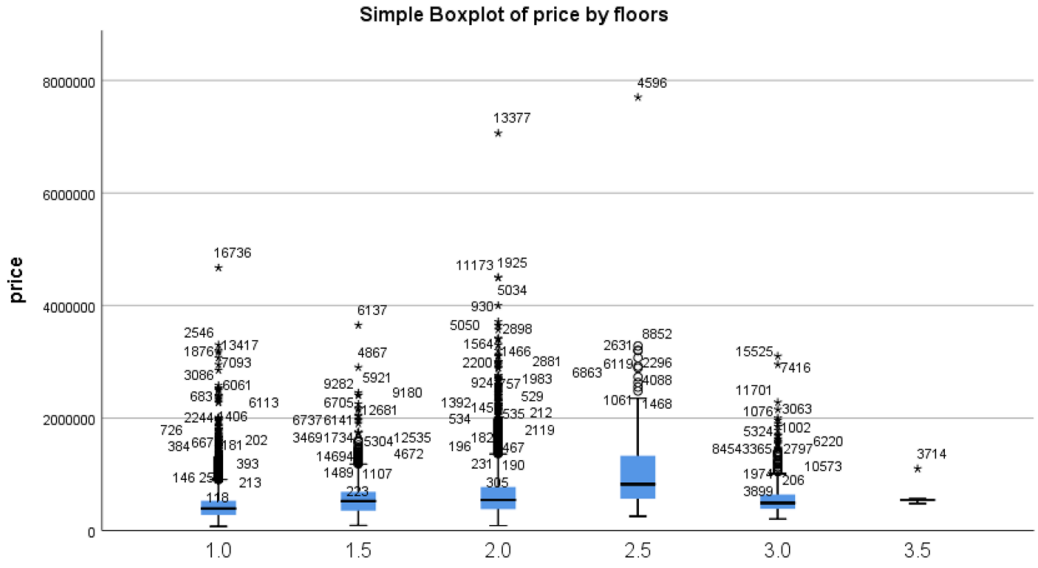
Y : Price



We though that new houses were more expensive than older houses. However, it is not true there is something that affect this simple relation. It is renovation, when the houses were renovated, this effect help to increase the prices of houses. These interaction was considered in our fixing modeling



In this plot is evident there is a change of slope. Therefore, we consider to model with the methodology of piecewise in order to changes to direction of slope during modeling process



Modeling

After the exploration and creating new variables, we try different kind of regression models. First, we started to fit with simples variables, then we try more complicated models and finally we choose the most convenient in order to explain the price of houses in King County city.

Model: Multiple Regression

Dataset: 17829 observations and 30 variables.

Step 1: Try simple models

Simple Models

Model: MODEL1
Dependent Variable: price

Number of Observations Read	17289
Number of Observations Used	17289

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	1.248038E15	3.120094E14	5512.40	<.0001
Error	17284	9.782982E14	56601375387		
Corrected Total	17288	2.226336E15			

Root MSE	237910	**R-Square**	0.5606
Dependent Mean	540198	**Adj R-Sq**	0.5605
Coeff Var	44.04138		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	-36424	8068.27221	-4.51	<.0001
yr_age	**1**	2102.98350	66.06902	31.83	<.0001
bedrooms	**1**	-55100	2490.07882	-22.13	<.0001
sqft_living	**1**	313.76280	2.68605	116.81	<.0001
view_t	**1**	171872	6463.54792	26.59	<.0001

This Model has R-SQUARE 0.5606 and this is good, and all variables are significant. However, we decided to improve this score.

Polynomial second order

The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read	17289
Number of Observations Used	17289

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	6.462679E13	3.23134E13	258.39	<.0001
Error	17286	2.161709E15	1.250555E11		
Corrected Total	17288	2.226336E15			

Root MSE	353632	**R-Square**	0.0290
Dependent Mean	540198	**Adj R-Sq**	0.0289
Coeff Var	65.46343		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	676222	6928.88101	97.59	<.0001
yr_age	**1**	-7134.17898	313.96409	-22.72	<.0001
YR_AGE_CUADRA	**1**	63.54092	2.95005	21.54	<.0001

It is a simple model with a variable, we can see R-square is 3%, it means that just 3% of variability is explained for the model. So we start to use more variables.

Model with dummy

Model 1 dummy

The variable view_1 has values 0 and 1, it means that the building has view or does have view.

The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read	17289
Number of Observations Used	17289

Analysis of Variance

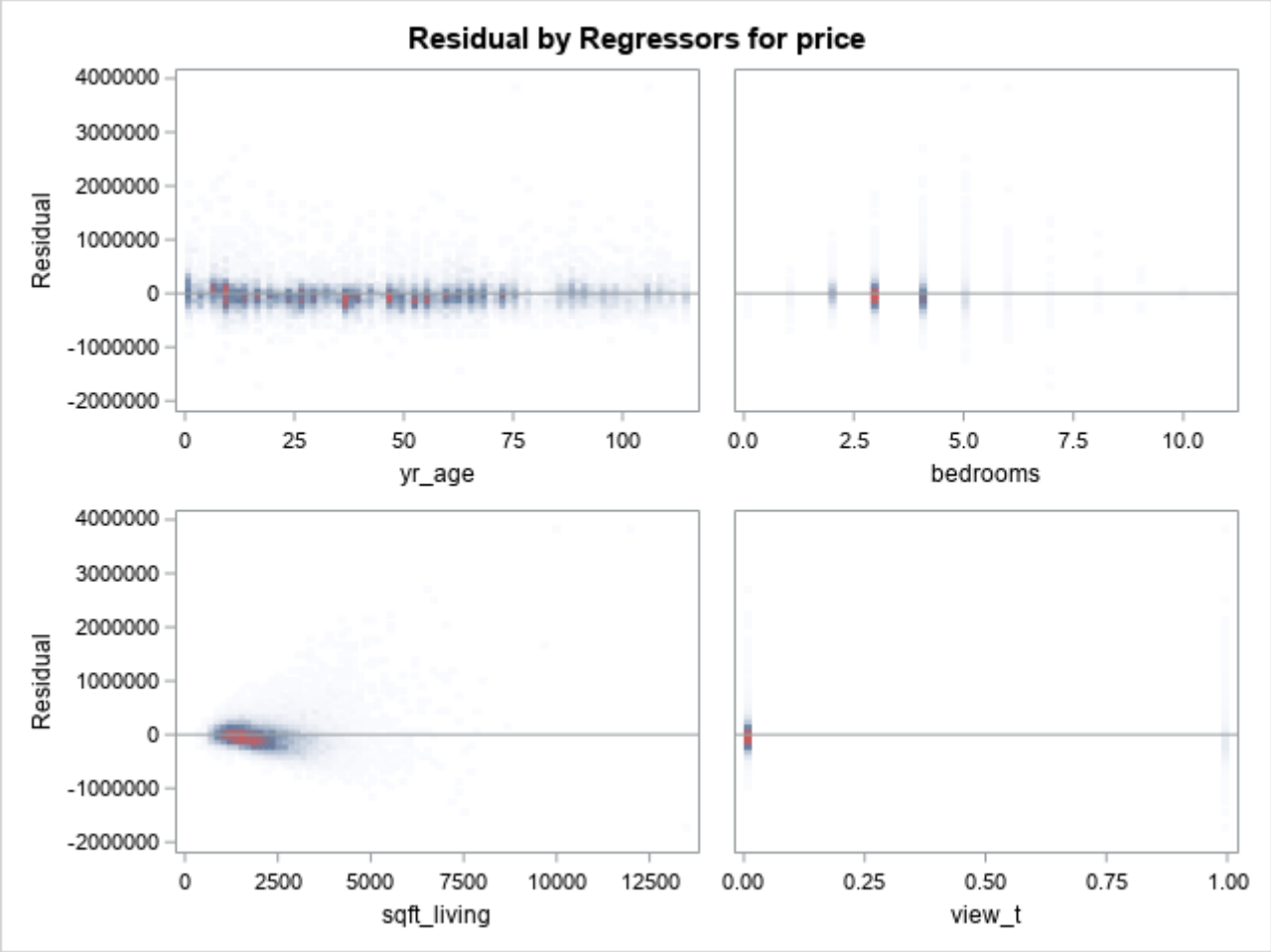
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	1.248038E15	3.120094E14	5512.40	<.0001
Error	17284	9.782982E14	56601375387		
Corrected Total	17288	2.226336E15			

Root MSE	237910	**R-Square**	0.5606
Dependent Mean	540198	**Adj R-Sq**	0.5605
Coeff Var	44.04138		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	-36424	8068.27221	-4.51	<.0001
yr_age	**1**	2102.98350	66.06902	31.83	<.0001
bedrooms	**1**	-55100	2490.07882	-22.13	<.0001
sqft_living	**1**	313.76280	2.68605	116.81	<.0001
view_t	**1**	171872	6463.54792	26.59	<.0001

The REG Procedure
Model: MODEL1
Dependent Variable: price



Model 2: With dummy

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read	17289
Number of Observations Used	17289

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	1.451891E15	1.319901E14	2944.55	<.0001
Error	17277	7.744444E14	44825167587		
Corrected Total	17288	2.226336E15			

Root MSE	211720	**R-Square**	0.6521
Dependent Mean	540198	**Adj R-Sq**	0.6519
Coeff Var	39.19299		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	-1025703	18642	-55.02	<.0001
yr_age	**1**	1309.24690	219.93626	5.95	<.0001
YR_AGE_CUADRA	**1**	17.87659	1.90813	9.37	<.0001
view_t	**1**	108985	5995.66014	18.18	<.0001
sqft_living	**1**	145.61303	3.33616	43.65	<.0001
grade	**1**	129505	2306.58776	56.15	<.0001
condition	**1**	20361	2731.64910	7.45	<.0001
waterfront	**1**	615313	19556	31.46	<.0001

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
bathrooms	**1**	30377	3604.14830	8.43	<.0001
zip1	**1**	5724.35899	4866.57751	1.18	0.2395
zip2	**1**	-1451.45161	5086.09906	-0.29	0.7754
zip3	**1**	46859	4689.06386	9.99	<.0001

This model has a high R-Square = 0.6521, however 2 dummies are not significant.

With Effect Piecewise and Second Polynomial second order

The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read 17289

Number of Observations Used 17289

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	2.9003E14	7.250749E13	647.22	<.0001
Error	17284	1.936306E15	1.120288E11		
Corrected Total	17288	2.226336E15			

Root MSE	334707	**R-Square**	0.1303
Dependent Mean	540198	**Adj R-Sq**	0.1301

Coeff Var	61.96012

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	216897	12165	17.83	<.0001
yr_age	**1**	-6643.72216	297.37947	-22.34	<.0001
YR_AGE_CUADRA	**1**	64.82010	2.79284	23.21	<.0001
bedrooms	**1**	128939	2876.94771	44.82	<.0001
X2STAR	**1**	-340622	66481	-5.12	<.0001

This model has a Rsquare = 13% , only 13 % of variability of price can be explaineded. However, when evaluating how the parameters are going, these are significant. We are going for a right path. We need to add more variables to the model.

Model with variables discretas

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read	17289
Number of Observations Used	17289

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	1.373367E15	3.433416E14	6957.25	<.0001
Error	17284	8.529693E14	49350223141		
Corrected Total	17288	2.226336E15			

Root MSE	222149	**R-Square**	0.6169
Dependent Mean	540198	**Adj R-Sq**	0.6168
Coeff Var	41.12368		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	-960260	17344	-55.37	<.0001
yr_age	**1**	3636.68928	64.41144	56.46	<.0001
bedrooms	**1**	-40147	2342.87746	-17.14	<.0001
sqft_living	**1**	200.60401	3.33950	60.07	<.0001
grade	**1**	138081	2385.47512	57.88	<.0001

We can see the R square is 61%, It is good. However we want to increase more this R square.

More models:

The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read	17289
Number of Observations Used	17289

Analysis of Variance

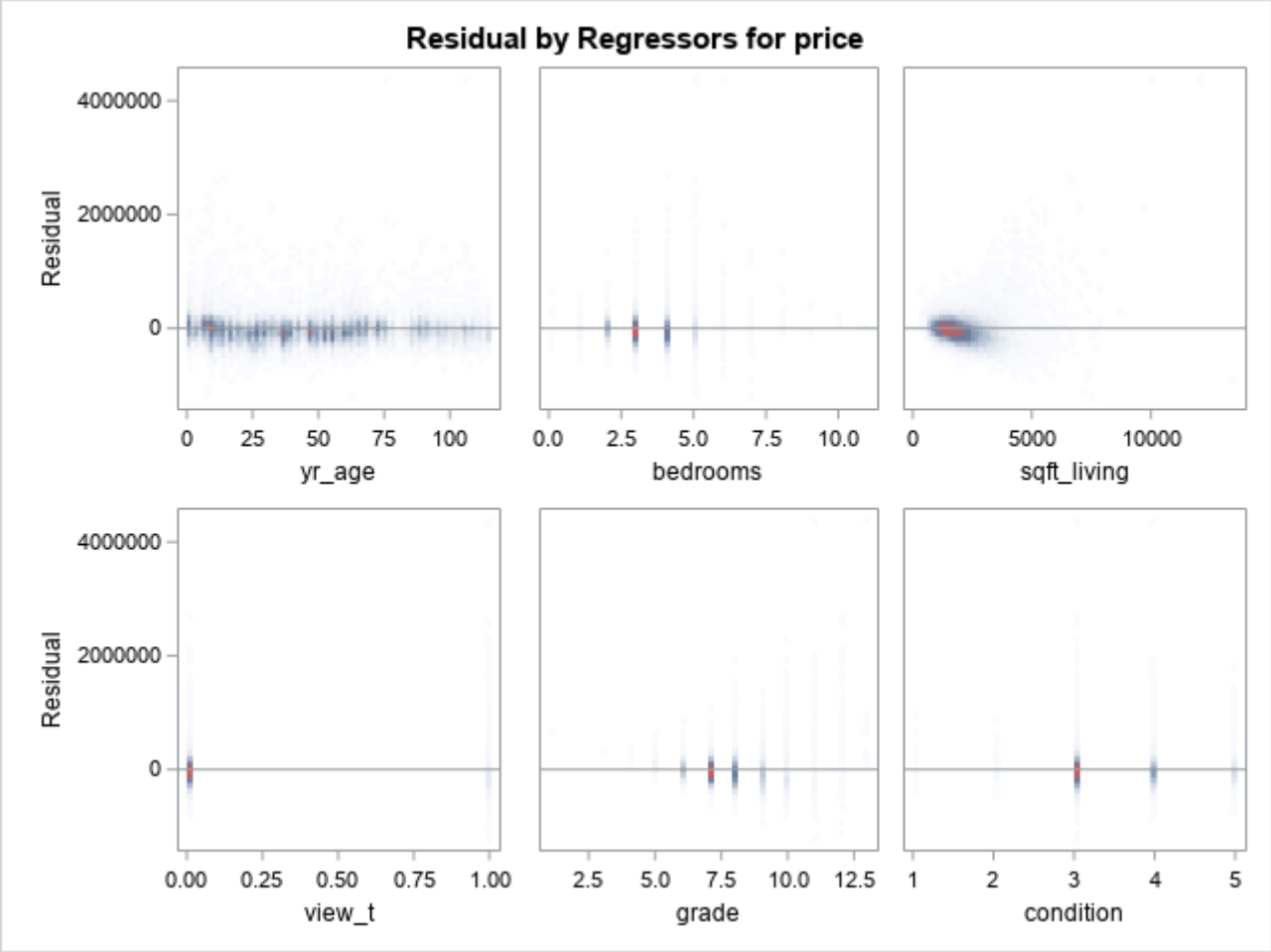
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	1.40064E15	2.3344E14	4885.95	<.0001
Error	17282	8.256957E14	47777784295		
Corrected Total	17288	2.226336E15			

Root MSE	218581	**R-Square**	0.6291
Dependent Mean	540198	**Adj R-Sq**	0.6290
Coeff Var	40.46322		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	-959831	19095	-50.27	<.0001
yr_age	**1**	3208.63184	68.39118	46.92	<.0001
bedrooms	**1**	-36178	2318.83635	-15.60	<.0001
sqft_living	**1**	187.61688	3.33085	56.33	<.0001
view_t	**1**	137330	5969.79571	23.00	<.0001
grade	**1**	132995	2360.47574	56.34	<.0001
condition	**1**	16922	2751.59016	6.15	<.0001

The REG Procedure
Model: MODEL1
Dependent Variable: price



The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read 17289

Number of Observations Used 17289

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	1.400743E15	2.001061E14	4188.55	<.0001

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Error	17281	8.25593E14	47774606298		
Corrected Total	17288	2.226336E15			

Root MSE	218574	**R-Square**	0.6292
Dependent Mean	540198	**Adj R-Sq**	0.6290
Coeff Var	40.46187		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	-958854	19106	-50.19	<.0001
yr_age	**1**	3205.80086	68.41615	46.86	<.0001
bedrooms	**1**	-36719	2347.90724	-15.64	<.0001
sqft_living	**1**	187.79476	3.33295	56.34	<.0001
view_t	**1**	137276	5969.71342	23.00	<.0001
grade	**1**	133021	2360.46330	56.35	<.0001
condition	**1**	17026	2752.41204	6.19	<.0001
X2STAR	**1**	63767	43493	1.47	0.1426

Model

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read	17289
Number of Observations Used	17289

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	9	1.417992E15	1.575546E14	3367.86	<.0001
Error	17279	8.083441E14	46781878785		
Corrected Total	17288	2.226336E15			

Root MSE	216291	**R-Square**	0.6369
Dependent Mean	540198	**Adj R-Sq**	0.6367
Coeff Var	40.03928		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	-563697	24610	-22.91	<.0001
yr_age	**1**	2958.41317	227.04863	13.03	<.0001
YR_AGE_CUADRA	**1**	6.01530	2.40531	2.50	0.0124
bedrooms	**1**	-36822	2334.53414	-15.77	<.0001
X2STAR	**1**	35541	43064	0.83	0.4092
INT_YR_AGE_YR_RENOVATED	**1**	-1.46466	1.54881	-0.95	0.3443
log_sqft_lot	**1**	-37314	2196.10788	-16.99	<.0001
view_t	**1**	139306	5936.38969	23.47	<.0001
sqft_living	**1**	206.96865	3.51317	58.91	<.0001
grade	**1**	127334	2356.48246	54.04	<.0001

Decision:

After evaluating different models, they have among R- square 0.55 and 0.64 with 5 and 8 variables, however we can get models with more R- square, but these had many variables and no all were significants . Therefore, we decided to select among the next models.

	Model 1	Model 2	Model 3	Model 4	Model 5
Age	X	X	X	X	X
Renovation					X
Size	X	X	X	X	X
Bedrooms	X	X	X	X	X
View					X
Condition	X				
Grade	X	X	X		X
Zone		X	X		
Floors			X	X	
Living					X
Water front				X	
Basement				X	
Bathroom	X				
Rsquare	**0.54**	**0.58**	**0.58**	**0.62**	**0.64**

Final Model

The SAS System

The REG Procedure
Model: MODEL1
Dependent Variable: price

Number of Observations Read 17289

Number of Observations Used 17289

Analysis of Variance

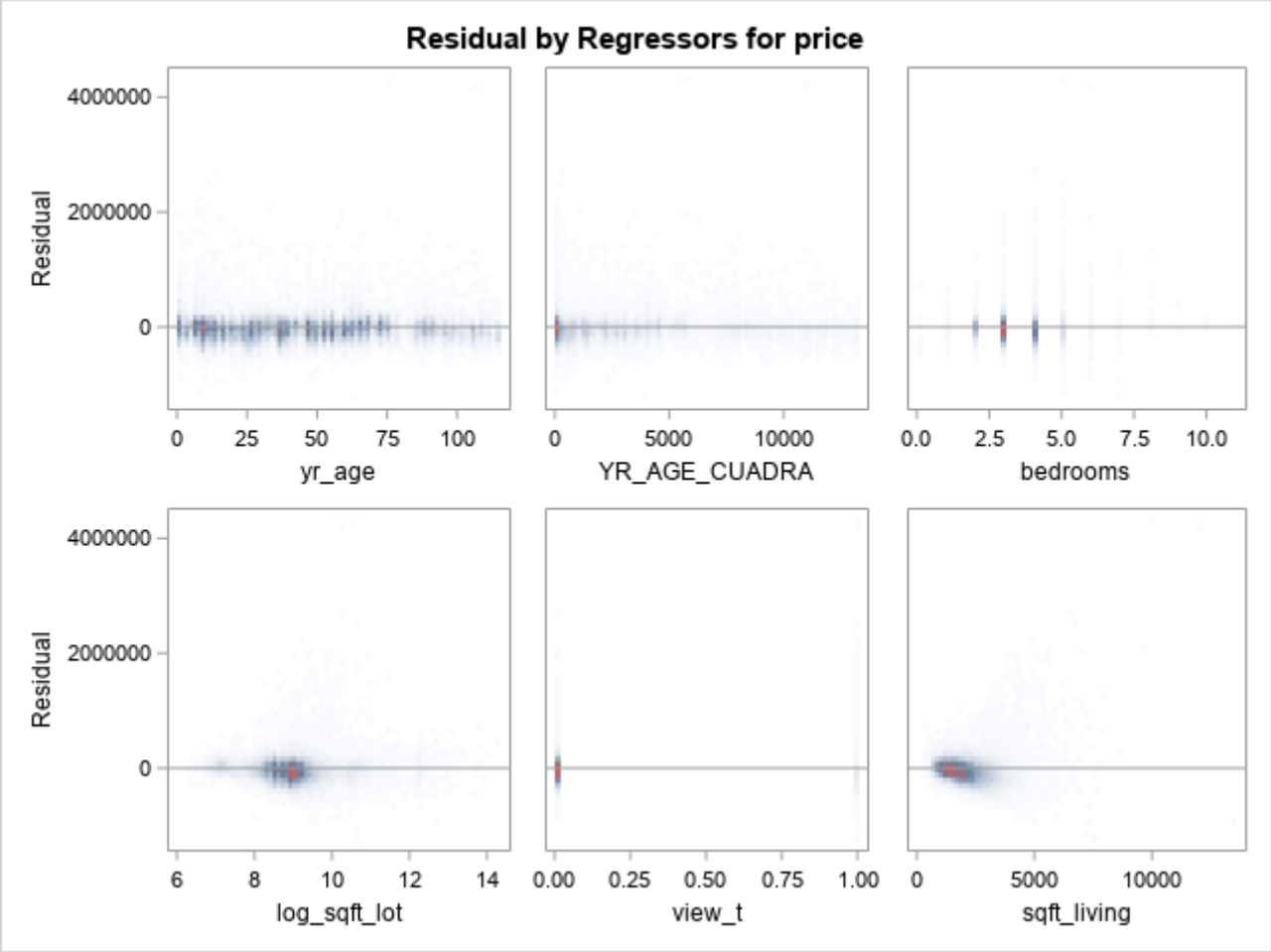
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	8	1.42073E15	1.775912E14	3809.28	<.0001
Error	17280	8.056062E14	46620730218		
Corrected Total	17288	2.226336E15			

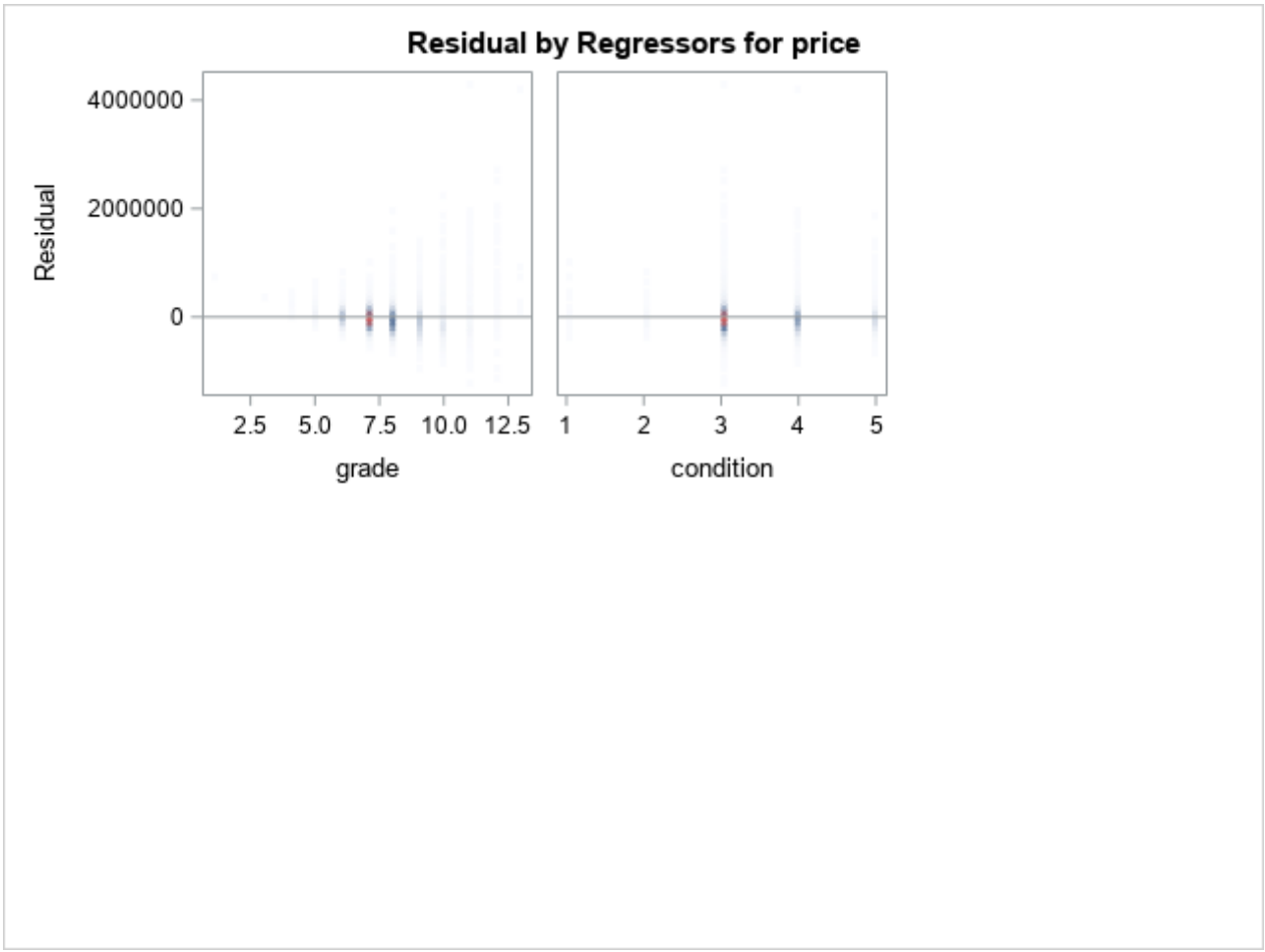
Root MSE	215918	**R-Square**	0.6481
Dependent Mean	540198	**Adj R-Sq**	0.6380
Coeff Var	39.97026		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	**1**	-626609	25774	-24.31	<.0001
yr_age	**1**	2485.28618	234.02762	10.62	<.0001
YR_AGE_CUADRA	**1**	7.57086	2.07243	3.65	0.0003
bedrooms	**1**	-37284	2303.93989	-16.18	<.0001
log_sqft_lot	**1**	-36862	2193.08433	-16.81	<.0001
view_t	**1**	139629	5924.45505	23.57	<.0001
sqft_living	**1**	205.77015	3.50692	58.68	<.0001
grade	**1**	127655	2349.51988	54.33	<.0001
condition	**1**	21449	2760.94775	7.77	<.0001

The REG Procedure
Model: MODEL1
Dependent Variable: price




Residual by Regressors for price
Residual
4000000
2000000
0
2.5
5.0
7.5
10.0
12.5
grade
1
2
3
4
5
condition

Model

$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 \text{LOG} x_4 + \beta_6 x_5 + \beta_7 x_6 + \beta_8 x_7$

Fitting the model

$\hat{Y} = -626609 + 2485.28 x_1 + 7.57 x_1^2 - 37284 x_2 + -36862 x_3 + 139629 x_4 + 205.77 x_5 + 127655 x_6 + 21449 x_7$

Testing Using Usefulness of the model

Ho : B1= B2 = 0

H1 : At least one of the coefficients is nonzero

Test Statistic :F = 3809.28

p-value = 0.0001

Conclusion: since α = 0.05 exceeds the observed significance level, p=0.0053, the data provide strong evidence that at least one of the model coefficients is nonzero. The overall model appears to be statistical usefull for predicting earnings.

Testing B Parameters Coefficients in the Multiple Regression

For each Bi in our model:

Ho : Bi = 0

Ho : B4 $\neq$ 0

Since p value is 0.001 and α = 0.01, so p value is much smaller than 0.05, we reject Ho. Therefore, The parameter Bi is different 0

H1 : Bi $\neq$ 0

Test statistics: t

Since p values are 0.0001 and α = 0.05 (sas output), so p value is much smaller than 0.05, we have evidence not reject Ho. Thus, the parameter Bi is different than 0

All B-Parameters are significant, because p- values are smaller than 0.05.

Interpretation of Coefficients Betas of the model

Bo is -626609, It is impractical, because the price can not be negative. The prices of the houses are also positive.

B1 is 2484.5, B1 is the slope and this case it is positive, it is the mean of price of houses E(y) will increase 2484.5 for every unit increase in age of building when the other variables is held fixed.

B2 is 7.57, it is positive, so the mean of price of houses E(Y) will increase 7.57 for every unit increase in the square of age of building when the other variables is held fixed.

B3 is -37284, it is negative, so the mean of price of houses E(Y) will decrease -37284 for every unit increase in bedrooms when the other variables is held fixed.

B4 is -36862, it is negative, so the mean of price of houses E(Y) will decrease 36862 for every unit increase in logarithm the all first lote floor when the other variables is held fixed.

B5 is 139629, it is positive, so the mean of price of houses E(Y) will increase 139629 for houses with view, houses with not view d o not have this effect. when the other variables is held fixed.

B6 is 205.77, it is positive, so the mean of price of houses E(Y) will increase 205.77 for every unit increase in size of km when the other variables is held fixed.

B7 is 127655, it is positive, so the mean of price of houses E(Y) will increase 127655 for every unit increase in grade when the other variables is held fixed.

B8 is 21449, it is positive, so the mean of price of houses E(Y) will increase 21499 for every unit increase in condition when the other variables is held fixed.

Multicollinearity

The SAS System

The CORR Procedure

4 Variables: yr_age bedrooms log_sqft_lot sqft_living

Simple Statistics

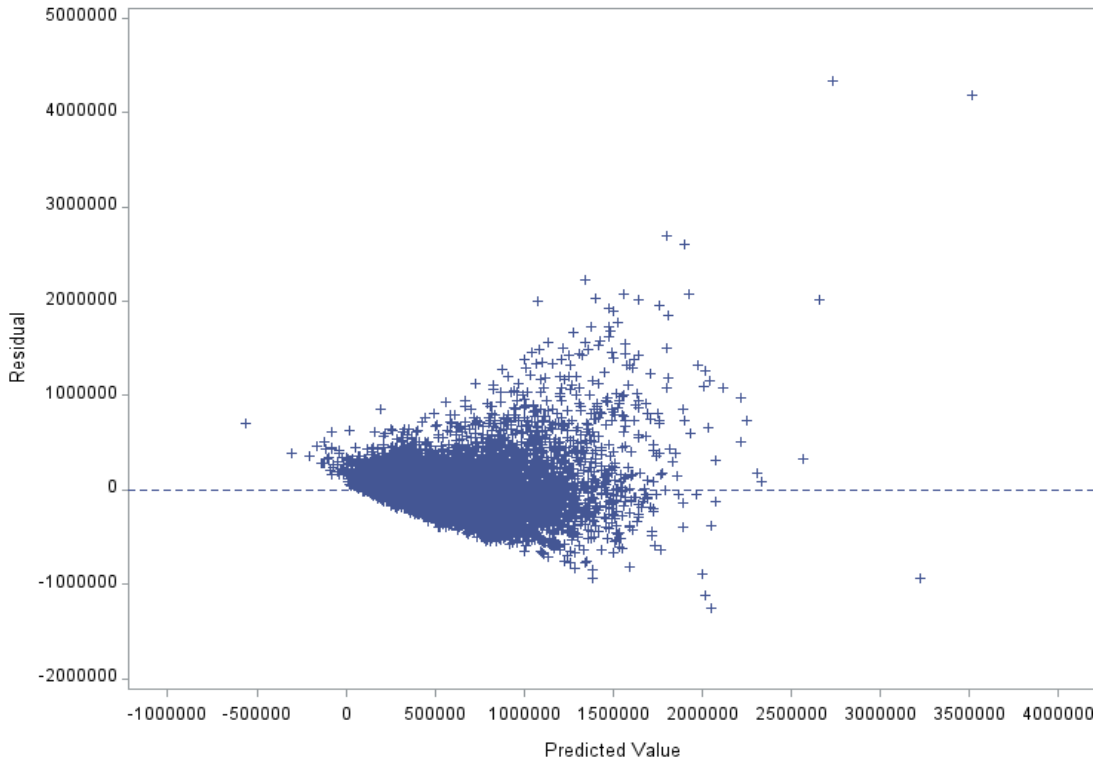
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
yr_age	17289	43.98161	29.37772	760398	0	115.00000
bedrooms	17289	3.36943	0.90641	58254	0	11.00000
log_sqft_lot	17289	8.98934	0.90077	155417	6.25383	14.31711
sqft_living	17289	2081	917.10323	35984111	290.00000	13540

Pearson Correlation Coefficients, N = 17289
Prob > |r| under H0: Rho=0

	yr_age	bedrooms	log_sqft_lot	sqft_living
yr_age	1.00000	-0.15372	0.00436	-0.32070
		<.0001	0.5662	<.0001
bedrooms	-0.15372	1.00000	0.19314	0.59024
	<.0001		<.0001	<.0001
log_sqft_lot	0.00436	0.19314	1.00000	0.34820
	0.5662	<.0001		<.0001
sqft_living	-0.32070	0.59024	0.34820	1.00000
	<.0001	<.0001	<.0001	

Our variables do not contribute with redundant information, these are not strong correlated with each other. It is common to find correlation. However, we try to control this effect that could give not real values and relation among the variables.

Residual Analysis



The SAS System

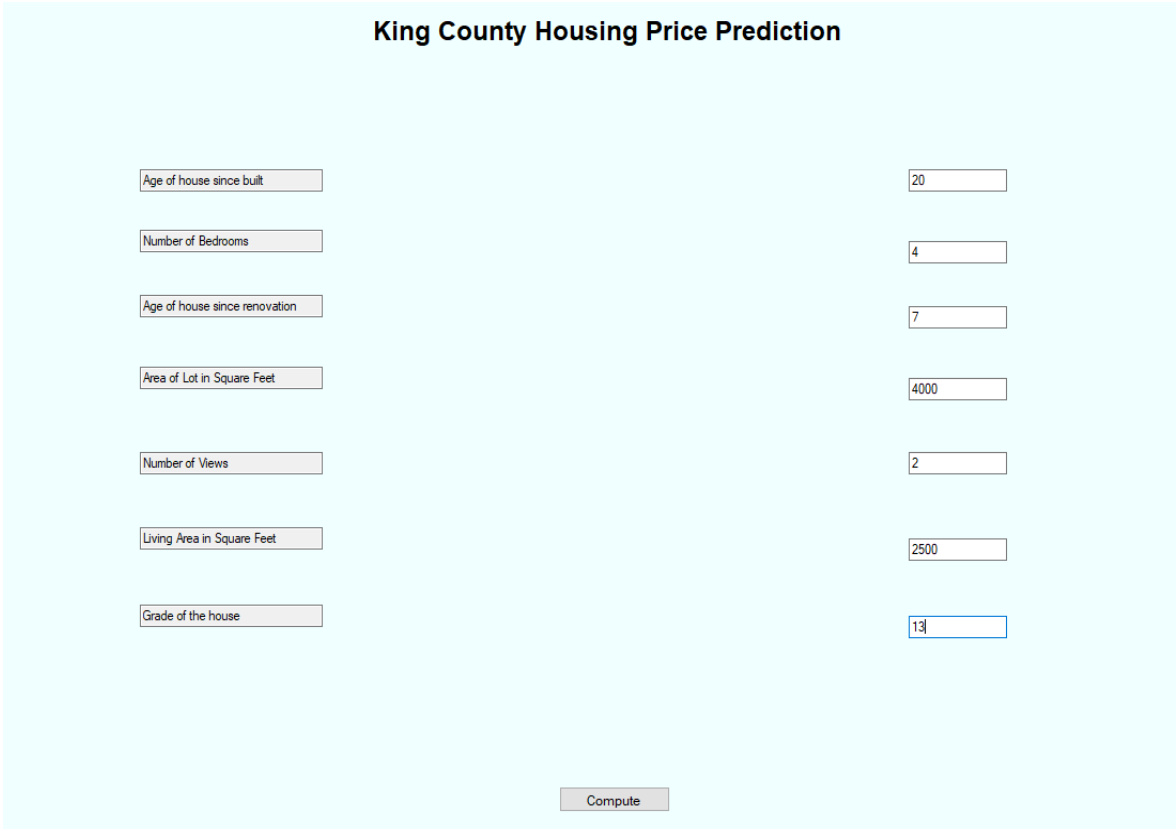
The REG Procedure
Model: MODEL1
Dependent Variable: price

Durbin-Watson D	2.011
Number of Observations	17289
1st Order Autocorrelation	-0.005

By visual inspection, the residual plot indicates that the values are around the cero and It does not have a patron or tendency. So the variances satisfy the property of homoscedastic. However, after of quickly research durbin Watson, because we can not use table (book) or table website (n=200), we have 1700 observations, we considered that it is a border. So we do not consider this statistic in this evaluation, just plot observation.

IMPLEMENTATION

By getting the equation of the best model for our data we can now predict the price of houses. For demonstration we have made a Windows Form Application which outputs the value of the price depending upon the value of independent variables. Image of the Application is displayed below.



From the image we can see that the textboxes are provided to enter relevant data and a compute button. After the values of independent variables are entered, we press the compute button. After pressing this button, a Message Box appears displaying the Price Range value. The price range is between 0.9*Predicted_Value and 1.1* Predicted_Value.

The values to be entered are:

1. Age of house since built
2. Number of bedrooms
3. Age of house since renovation
4. Area of lot in square feet.

5. Number of views
6. Living area in square feet
7. Grade of the house

For making this application we had used Visual Studio 2019 and the language of code is C#. Below Images displays the code of the program.

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    var betaAge = 8643.41071;
    var betaAgeSquare = -55.69137;
    var betaBedroom = -36286;
    var betaX2Star = 38793;
    var betaYearRenovateAge = -6133.55030;
    var betaInt_Yr_AgeYr_Renovated = 64.61776;
    var betaLogSqftLot = -36169;
    var betaView = 137908;
    var betaSqftLiving = 204.91411;
    var betaGrade = 127454;
    var intercept = -566374;
```

The above code block shows assigning the beta values of our final model to the variables.

```
age = float.Parse(textBox1.Text);
bedrooms = float.Parse(textBox11.Text);
yr_renovate_age = float.Parse(textBox9.Text);
sqft_feet_lot = Math.Log(Int32.Parse(textBox7.Text));
view = Int32.Parse(textBox5.Text);
sqft_living = Int32.Parse(textBox3.Text);
grade = Int32.Parse(textBox16.Text);

if (age < 0) {

    MessageBox.Show("Enter the correct age value");
}

if (bedrooms < 0) {
    MessageBox.Show("Enter the correct number of bedrooms value");
}

if (yr_renovate_age == 0) {
    yr_renovate_age = age;
}

if (sqft_feet_lot < 0) {
    MessageBox.Show("Enter the correct Area of Lot in Square Feet");
}

if (view != 0) {
    view = 1;
}

if(view < 0)
{
```

```
    }

    if (sqft_feet_lot < 0) {
        MessageBox.Show("Enter the correct Area of Lot in Square Feet");
    }

    if (view != 0) {
        view = 1;
    }

    if(view < 0)
    {
        MessageBox.Show("Enter the correct Number of Views Value");
    }

    if (sqft_living < 0) {
        MessageBox.Show("Enter the correct Area of Living in Square Feet");
    }

    if (grade < 0 || grade > 13) {
        MessageBox.Show("Enter the correct Grade");
    }

    if (bedrooms >= 8) {
        X2Star = bedrooms - 8;
    }

    else
    {
        X2Star = 0;
    }
```

The above two snapshots shows how we created the text boxes and what should be the error message if someone gives an incorrect value.

```
price = intercept + betaAge*age - betaAgeSquare*age*age + betaBedroom*bedrooms + betaX2Star*X2Star + betaYearRenovateAge*yr_renovate_age +
        betaInt_Yr_AgeYr_Renovated * age * yr_renovate_age +
        betaLogSqftLot * sqft_feet_lot + betaView * view + betaSqftLiving * sqft_living + betaGrade * grade;
```

The above code shows the calculation based on our final model.

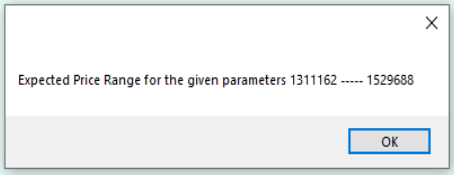
```
MessageBox.Show("Expected Price Range for the given parameters " +price_min + " ----- " +price_max);
```

The above snapshot shows the code for displaying the output block.

Compute button on the output window applies the equation which is displayed on Page No. xx and calculate the price range values. Image below displays the message box which is displaying price range values.

King County Housing Price Prediction

Age of house since built	20
Number of Bedrooms	4
Age of house since renovation	7
Area of Lot in Square Feet	4000
Number of Views	2
Living Area in Square Feet	
Grade of the house	



Compute

Conclusion

- Our first datasets were not used because these needed another kind of models and this course is strictly Regression, so we had to look for new datasets during the course. Then we had different datasets, but finally we worked together only with one dataset.

- Have a good dataset is very important in order to make a great a model, in this case the dataset was easy to manage, it does not have string values, just numbers. Therefore, it makes the process of preprocessing more friendly and quickly, no too much preprocessing but smart exploration.

- Doing a good exploration is crucial in order to understand the dataset and create useful variables. For example, how many years ago it was renovated, also dummies like geographical dummy.

- We tried and did models with the new concepts that were taught in class as piecewise regression, dummies and also polynomials

- Nobody in this team knows about housing business. However, during this modeling we learnt quickly how this business works and finally we could create a friendly model that can be understand and used.

- All HomeWorks are very important in order to reinforce specific concepts, no matter if we practiced with small bases, it was useful in our project.

- The powerful software SAS and SPSS command these project, so it is very important to manage different softwares.

- The application is a great tool in order to show our model.

Program

```
PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE YR_AGE_CUADRA BEDROOMS X2STAR YR_RENOVATED_AGE
INT_YR_AGE_YR_RENOVATED;
RUN;

PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE YR_AGE_CUADRA BEDROOMS X2STAR YR_RENOVATED_AGE
INT_YR_AGE_YR_RENOVATED SQFT_LIVING;
RUN;


PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE YR_AGE_CUADRA BEDROOMS X2STAR YR_RENOVATED_AGE
INT_YR_AGE_YR_RENOVATED SQFT_LIVING;
RUN;


PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE YR_AGE_CUADRA BEDROOMS X2STAR YR_RENOVATED_AGE
INT_YR_AGE_YR_RENOVATED SQFT_LIVING VIEW_T;
RUN;
*/mORE MODELS  NO CUADRADO**/


PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  BEDROOMS SQFT_LIVING VIEW_T;
RUN;

/* Modles with no square and qauality**/

PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  BEDROOMS SQFT_LIVING VIEW_T grade condition;
RUN;

/* Modles with no square and qauality and piecewaise in bedrrom **/

PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  BEDROOMS SQFT_LIVING VIEW_T grade condition
x2star ;
RUN;

/* Modles with no square and qauality and piecewaise in bedrrom **/

PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  yr_age_cuadra bedrooms x2star
int_yr_age_yr_renovated log_sqft_lot view_t sqft_living grade;
RUN;



/* Modles with no square and qauality and piecewaise in bedrrom **/

PROC REG DATA= FINAL_MB_T_3;
```

```
MODEL PRICE = YR_AGE  yr_age_cuadra bedrooms x2star
int_yr_age_yr_renovated log_sqft_lot view_t sqft_living grade ;
RUN;


/* Modles with no square and qauality and piecewaise in bedrrom **/

PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  yr_age_cuadra bedrooms x2star
int_yr_age_yr_renovated log_sqft_lot view_t sqft_living grade;
RUN;









 PROC REG DATA=HOUSING_F_2;
  MODEL PRICE = YR_BUILT YR_BUILT2;
  RUN;

PROC REG DATA= FINAL_V3;
MODEL PRICE = YR_AGE YR_AGE_CUADRA BEDROOMS X2STAR;
RUN;

PROC REG DATA= FINAL_V3;
MODEL PRICE = YR_AGE YR_AGE_CUADRA BEDROOMS X2STAR YR_RENOVATED_AGE
YR_AGE*YR_RENOVATED_AGE;
RUN;




PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  yr_age_cuadra bedrooms x2star
int_yr_age_yr_renovated log_sqft_lot view_t sqft_living grade;
RUN;




PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  BEDROOMS SQFT_LIVING VIEW_T grade condition;
RUN;



PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  BEDROOMS SQFT_LIVING VIEW_T grade condition
x2star ;
RUN;

/* Modles with no square */
```

```
PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  yr_age_cuadra bedrooms x2star
int_yr_age_yr_renovated log_sqft_lot view_t sqft_living grade;
RUN;
```

```
PROC REG DATA= FINAL_MB_T_3;
MODEL PRICE = YR_AGE  yr_age_cuadra bedrooms x2star
int_yr_age_yr_renovated log_sqft_lot view_t sqft_living grade;
```

```
DATASET NAME DataSet4 WINDOW=FRONT.
SORT CASES BY yr_built (A).
SORT CASES BY yr_built (D).
RECODE yr_built (2015=1) (2014=1) (2013=1) INTO year_1.
VARIABLE LABELS  year_1 'year_1'.
EXECUTE.
RECODE yr_built (2003 thru 2012=1) (ELSE=0) INTO year_2.
VARIABLE LABELS  year_2 'year_2'.
EXECUTE.
```

```
from sklearn.model_selection import train_test_split
final_mb_t,final_mb_v=train_test_split(data,test_size=0.2,random_state=33)
```

```
In [25]: mean_sqft_ab=int(data.sqft_above.mean())

In [26]: mean_sqft_ab
Out[26]: 1788

In [27]: data.sqft_above.fillna(mean_sqft_ab,axis=0,inplace=True)
```