Project – 1

Prediction for house prices in King County, USA

by Linear Regression

By –

Divanshu Singh (8804523)

Predictive Analytics, Conestoga College, Doon Campus

(STAT8030) Multivariate Statistics

Bip Thapa

27th February 2023

Table of Contents

Introduction

Predicting house prices is a crucial task in the real estate industry, and linear regression is a popular method. In King County, USA, predicting house prices is particularly important due to the county's large population and significant economic activity. In recent years, there has been a growing interest in predicting house prices in King County using linear regression models, which utilize various features such as the number of bedrooms, square footage, floors, and other property characteristics. I have tried to apply all the advanced techniques to make accurate models to predict prices. The accuracy of these models has improved significantly with advanced techniques such as regularization, cross-validation, and feature selection. In this context, this paper aims to explore the effectiveness of linear regression models for predicting house prices in King County, USA, and to provide insights into the factors that affect the housing market in this region.

Gathering data

The "House Prices in King County, USA" dataset was obtained from Kaggle, and it is an excellent platform for data analytics and science enthusiasts. The dataset includes information on various attributes of houses, such as the number of bedrooms, bathrooms, square footage, floors, and price. Some unnecessary columns and outliers were removed to make the dataset more manageable and beneficial for our analysis. Removing irrelevant features from the dataset can help simplify the analysis process and provide more accurate insights into the factors influencing King County housing prices. By removing outliers, which are extreme values that can skew the analysis results, the dataset can be more reliable, resulting in more accurate predictions of house prices. With these adjustments, the "House Prices in King County, USA" dataset can provide valuable information for those interested in understanding the housing market in this region.

This dataset contains 12 columns with 21613 rows.

Here are all the columns with a description.

price column - denotes the price of each home sold.

bedrooms column – shows the number of bedrooms in a house.

bathrooms column – has the number of bathrooms in a house; the value of 0.5 indicates a half bathroom, typically consisting of a toilet and a sink but no shower or bathtub.

sqft_living column - represents the square footage of the apartment's interior living space.
sqft_lot column - indicates the square footage of the land space.

floors column - denotes the number of floors in each home.

waterfront column - is a dummy variable that indicates whether the apartment overlooks the waterfront or not.

view column - is an index ranging from 0 to 4, which rates the quality of the property's view from no to excellent.

condition column is an index that ranges from 1 to 5 and rates the apartment's condition from poor to very good.

grade column is an index ranging from 1 to 13, rating the building construction and design quality from falling short to high quality.

sqft_above column - indicates the square footage of the interior housing space above the ground level.

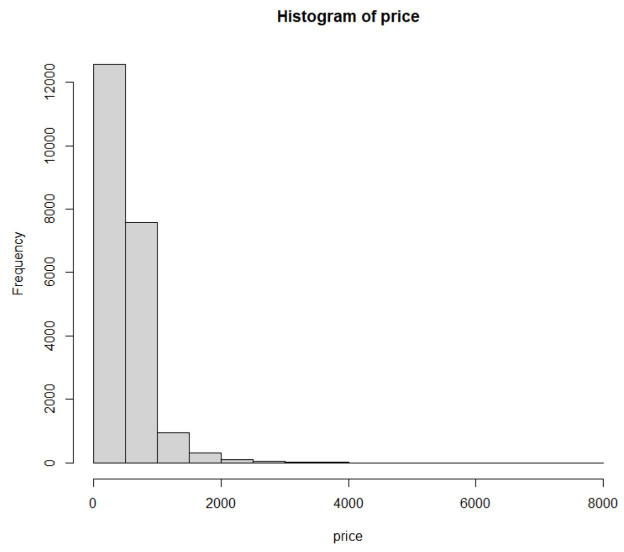
sqft_basement column - display the square foot internal interior housing space ground the ground level.

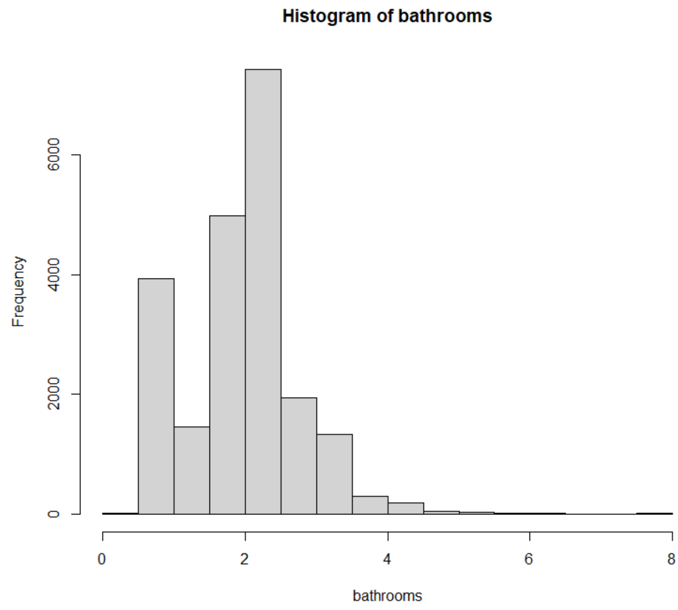
In this dataset, the price is the response variable, while the rest of the attributes are independent variables. The response variable is the variable that is being predicted or explained in the analysis. In contrast, the independent variables are the variables that are used to explain or predict the response variable. In this case, the price of the homes sold is the response variable. At the same time, attributes such as the number of bedrooms, bathrooms, square footage, floors, and other property characteristics serve as independent variables that can help explain or predict the sale price of a home.

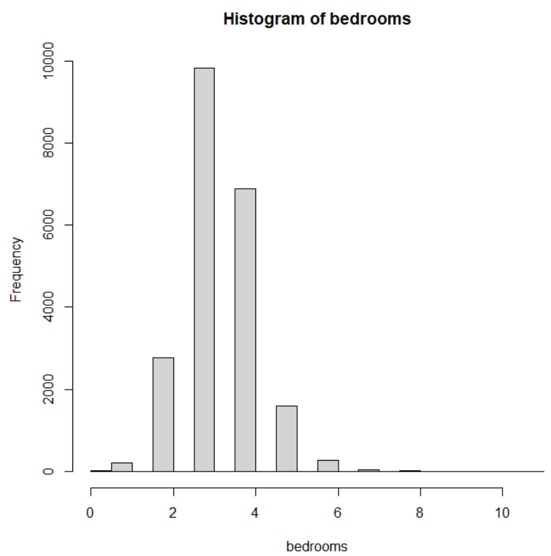
The price column in this dataset represents the sale price of each home, but the values have been given in hundreds of thousands and millions, making the numbers difficult to handle and interpret. Therefore, to make the data more manageable and easier to understand, the values in the price column have been divided by 1000. This transformation does not affect the relationship between the response and independent variables but makes the data more accessible for analysis. The new values in the price column represent the sale price of each home in thousands of dollars, which is easier to handle and interpret than the original values in hundreds of thousands and millions. This transformation simplifies the analysis process and allows for more straightforward comparisons and predictions based on the data.

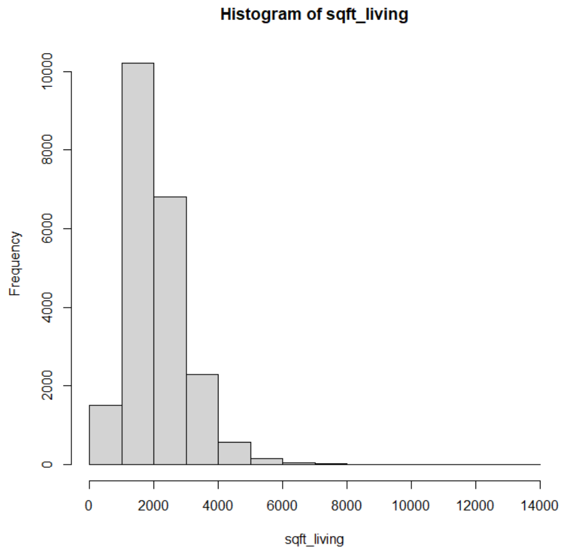
Frequency distribution of all the variables using histograms.

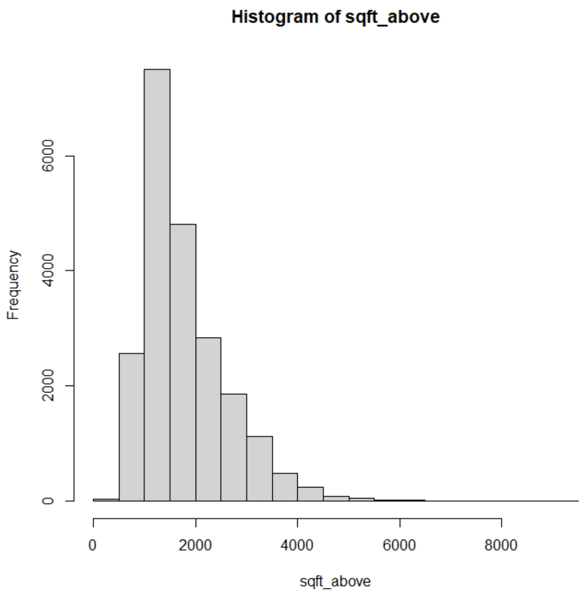
I have used hist() function in R to visualize the histograms for the frequency distribution of all the variables in this dataset.

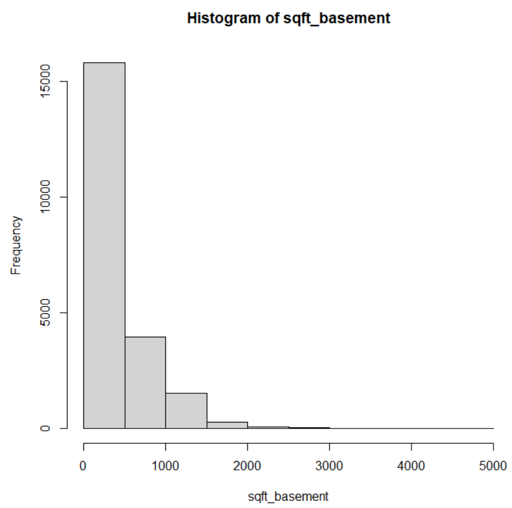

Histogram of price
Frequency
0
2000
4000
6000
8000
10000
12000
0
2000
4000
6000
8000
price

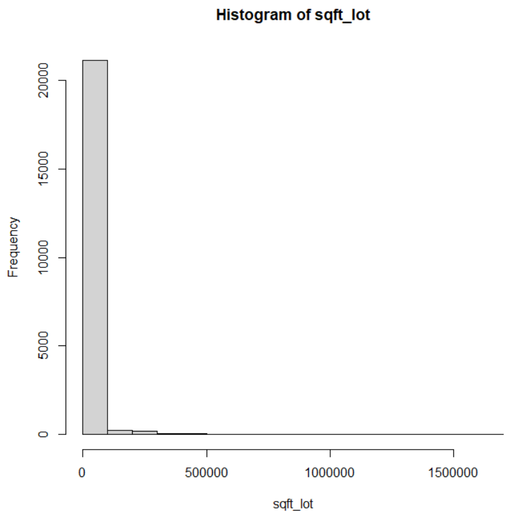

Histogram of bathrooms
Frequency
0
2000
4000
6000
0
2
4
6
8
bathrooms

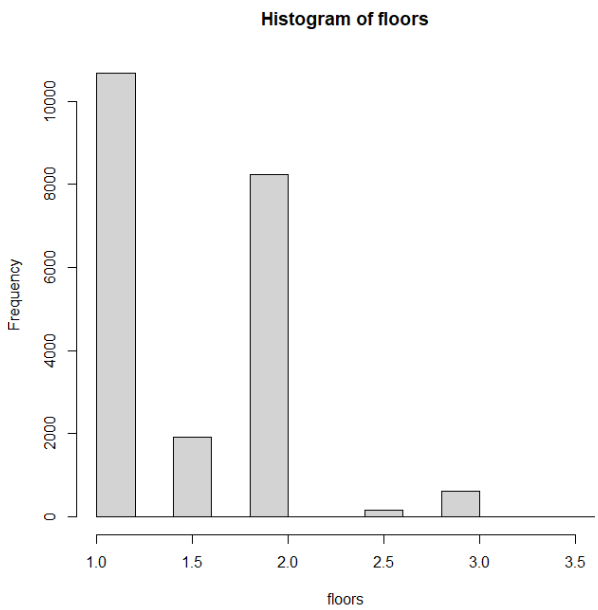

Histogram of bedrooms
Frequency
0
2000
4000
6000
8000
10000
0
2
4
6
8
10
bedrooms

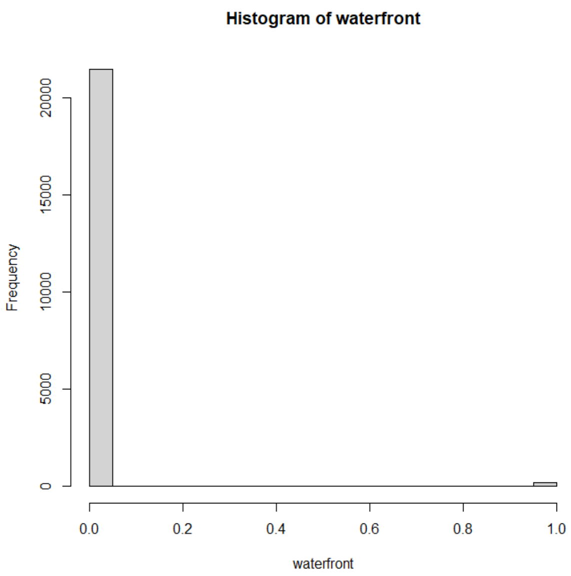

Histogram of sqft_living
Frequency
0
2000
4000
6000
8000
10000
0
2000
4000
6000
8000
10000
12000
14000
sqft_living

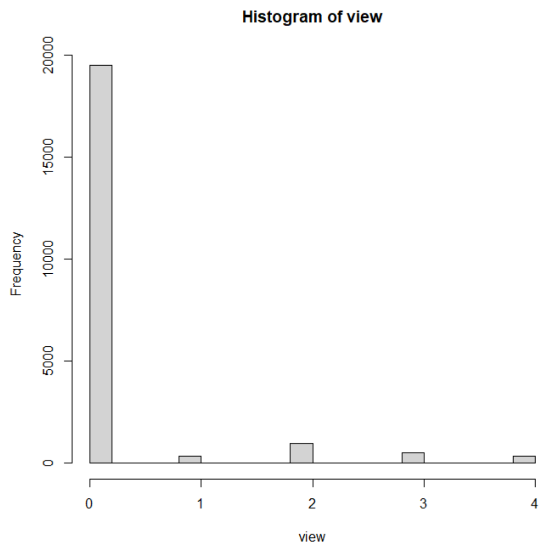

Histogram of sqft_above
Frequency
0
2000
4000
6000
0
2000
4000
6000
8000
sqft_above

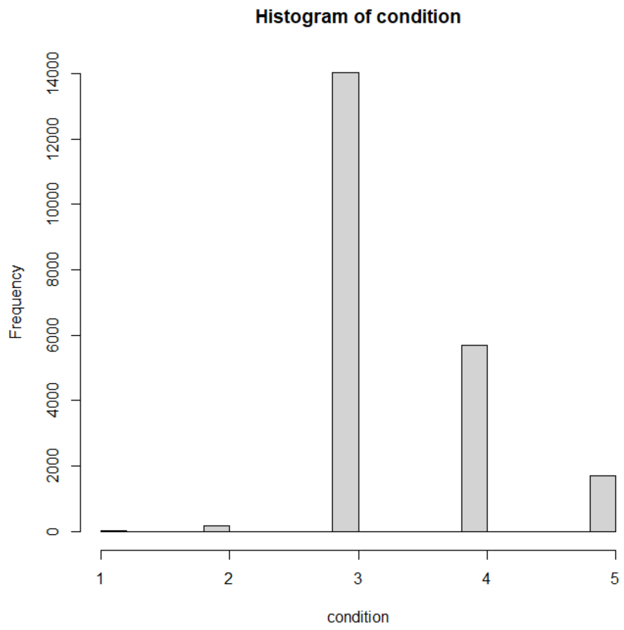

Histogram of sqft_basement
Frequency
0
5000
10000
15000
0
1000
2000
3000
4000
5000
sqft_basement

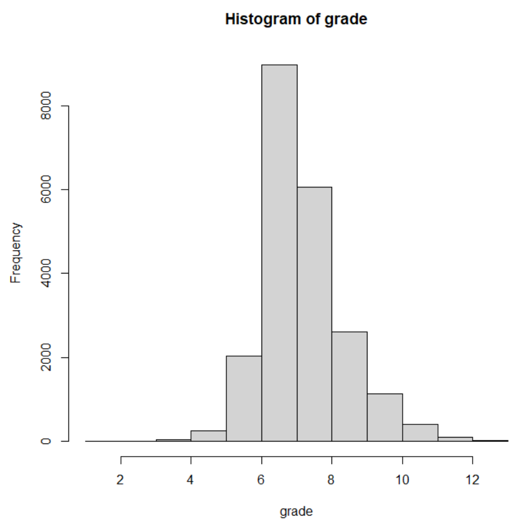

Histogram of sqft_lot
Frequency
0
5000
10000
15000
20000
0
500000
1000000
1500000
sqft_lot


Histogram of floors
Frequency
0
2000
4000
6000
8000
10000
1.0
1.5
2.0
2.5
3.0
3.5
floors


Histogram of waterfront
Frequency
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
waterfront


Histogram of view
Frequency
0
5000
10000
15000
20000
0
1
2
3
4
view


Histogram of condition
Frequency
0
2000
4000
6000
8000
10000
12000
14000
1
2
3
4
5
condition


Histogram of grade
Frequency
0
2000
4000
6000
8000
2
4
6
8
10
12
grade

Correlation

The histograms shown above display the frequency distribution of all the variables in the dataset. There is usually a positive correlation between the number of facilities and the price of a house, meaning that as the number of facilities increases, so does the price of the house, so does the price of the house. To determine if this pattern exists in our dataset, we must examine the covariance between each independent variable and the response variable. These covariance coefficients will indicate whether there is a correlation between the variables and, if so, the strength of that correlation. Here price is the response variable and other are the predictive variables.

Following is the table containing all the covariance coefficients –

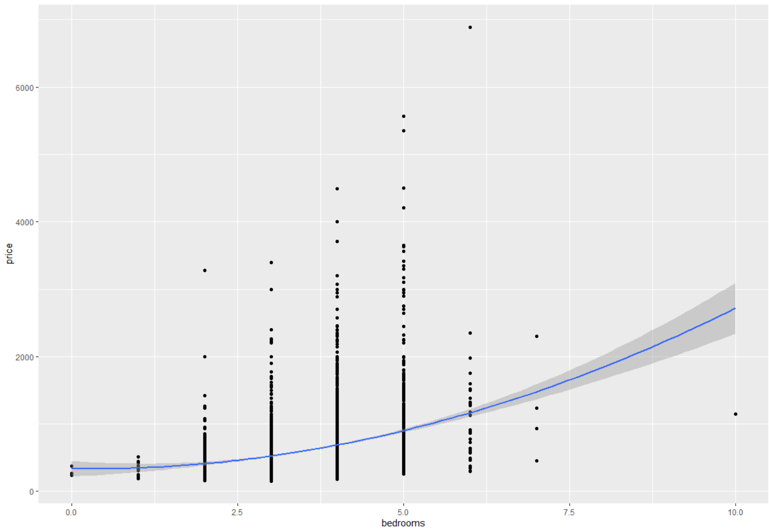
Parameters	**Correlation**
bedrooms	0.3437008
bathrooms	0.5946179
floors	-0.03726097
waterfront	0.2814419
view	0.3892226
condition	0.02040599
grade	0.7186924
sqft_above	0.6240518
sqft_basement	0.4138454
sqft_living	0.7129826
sqft_lot	0.155032

From the above table we can learn that except floors, all the values are positive which means there is a correlation between variables. Also, the covariance coefficient has a range of -1 to 1, with values close to zero indicating little or no correlation between the variables. A value closer to 1 indicates a strong positive correlation, while a value closer to -1 indicates a strong negative correlation.

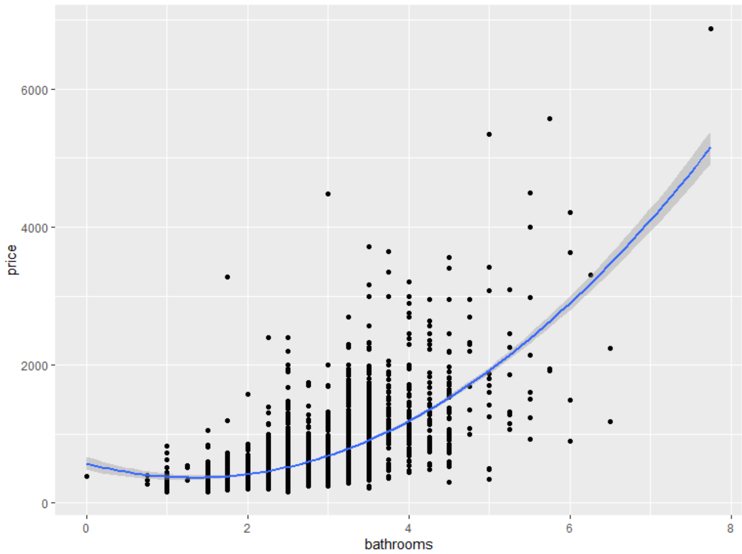
Here we can say that parameters – sqft_living, grade and sqft_above have the strongest correlation and sqft_lot, floors and condition have the weakest correlation.

I have also done a graphical representation of these correlation, using scatter charts. You can find them following –

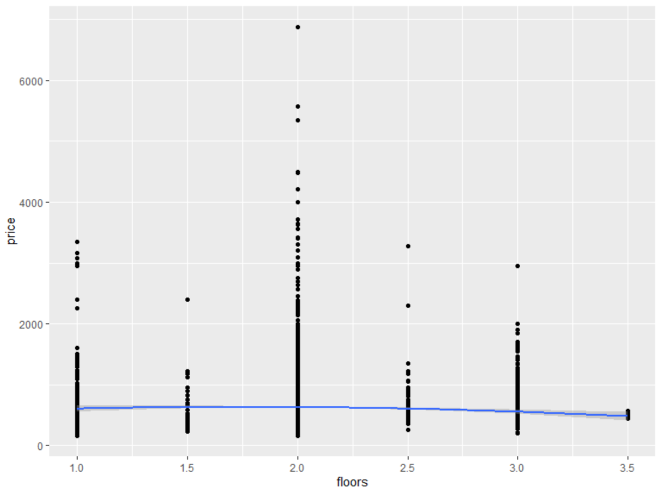
Price - Bedroom



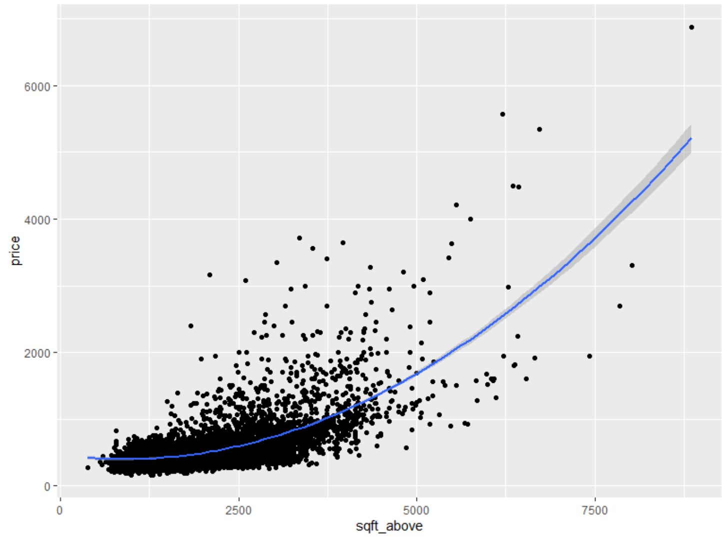
Price - Bathroom



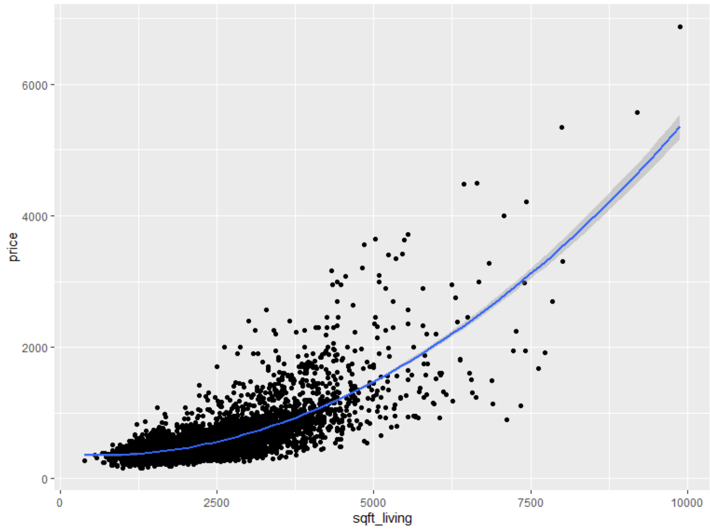
Price - floors



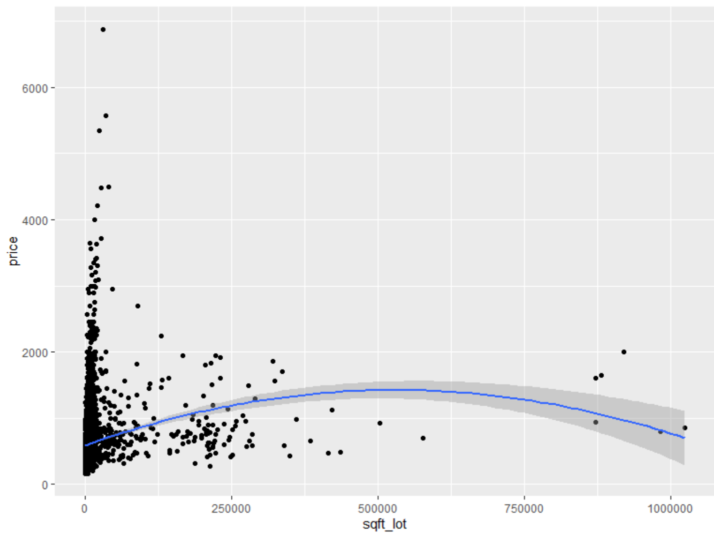
Price – sqft_above



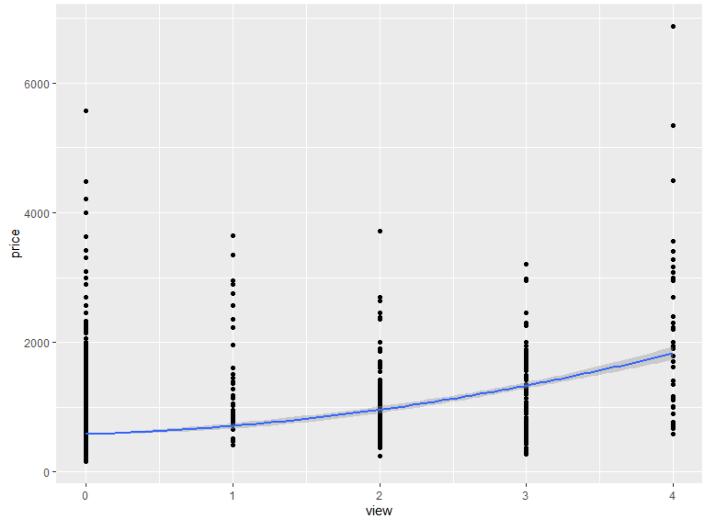
Price – sqft_living



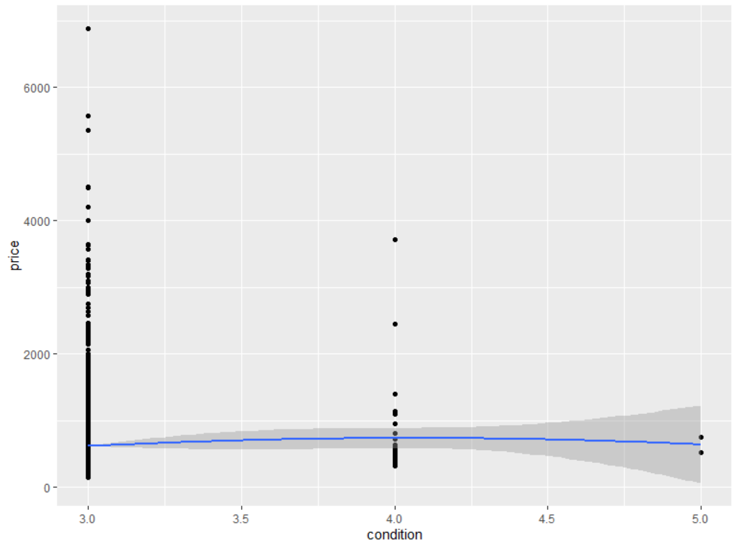
Price – sqft_lot



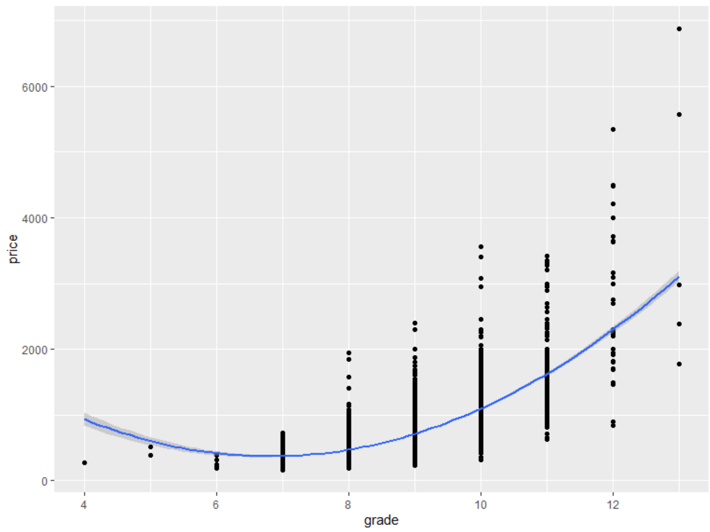
Price – view



Price – condition



Price – grade



Initial modeling

For this dataset, I have made three major models which would help us identify and help in finding the best one.

1. **First model** - I choose all the square feet variables in the dataset as it will give us a different perspective which will depict the relation between the square feet and the price of the houses.
2. **Second model** – For the second model, I tool all the variables that are related to the interior of the house. It will give us an understanding of the relationship of the price and the interior factor of the houses.
3. **Third model** – It is the model which contains all the variables of the dataset.

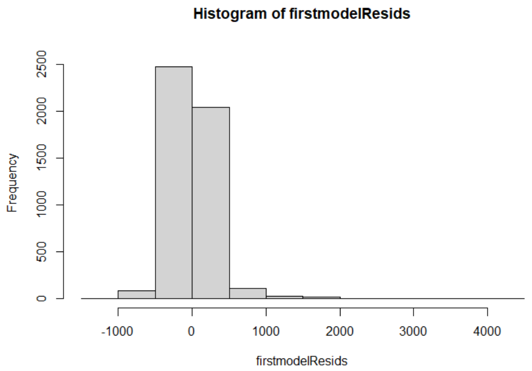
I have created a table to showcase all the models coefficients of the regression. Following are all the variables and coefficients. –

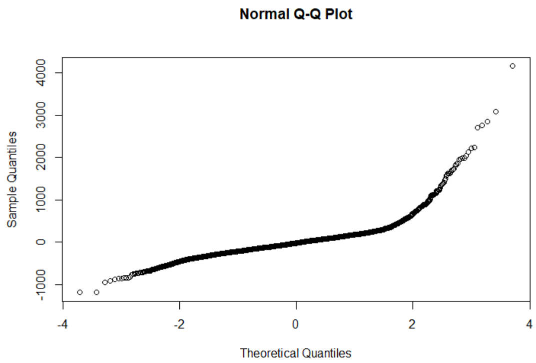
Model	**Variables (Response: price)**	**Coefficients of the Regression**
1.	**Predictors:** sqft_living, sqft_above, sqft_lot	(Intercept): -6.368873e+01 sqft_living: 4.248787e-01 sqft_above: -1.596033e-01 sqft_lot: -3.971134e-05
2.	**Predictors:** Bedrooms, bathrooms, floors, view	(Intercept): -314.364490 Bedrooms: 4.838902 Bathrooms: 346.669865 Floors: -19.272707 View: 161.529068
3.	**Predictors:** Bedrooms, bathrooms, floors, view, condition, grade, sqft_living, sqft_above, sqft_lot	(Intercept): -1.156679e+03 Bedrooms: -8.810762e+01 Bathrooms: 9.329996e+01 Floors: 4.312113e+01 View: 7.330982e+01 Condition: 6.092635e+01 Grade: 1.324256e+02 sqft_living: 2.620402e-01 sqft_above: -8.626652e-02 sqft_lot: -3.133347e-04

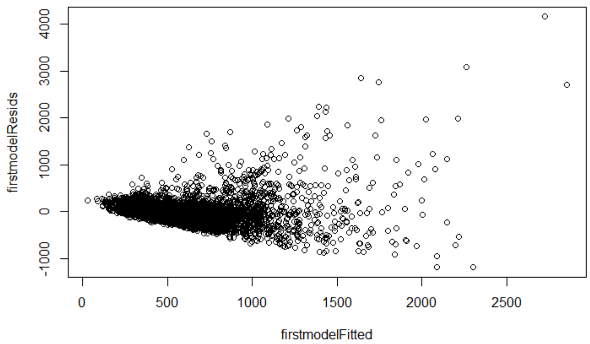
Diagnostics

Here I have used the Fitted vs Residuals graphs to identify the non – linearity, unequal error variances and outliers. This step gives us a better understanding of the models we created to predict the prices of the houses.

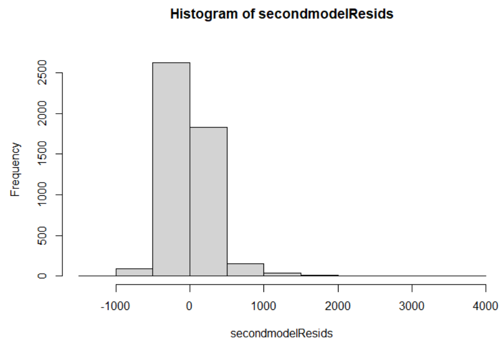
First model -

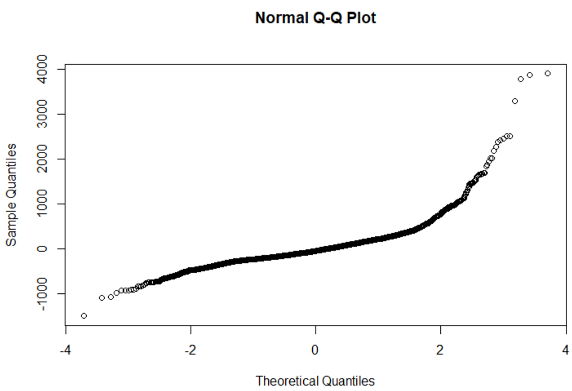


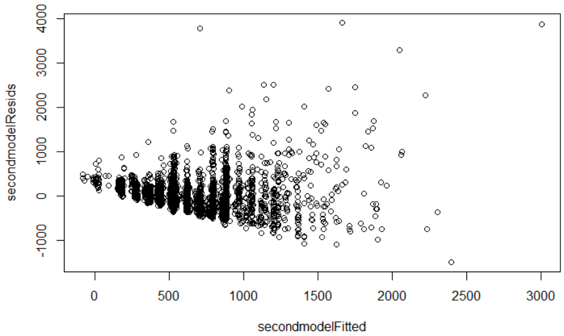




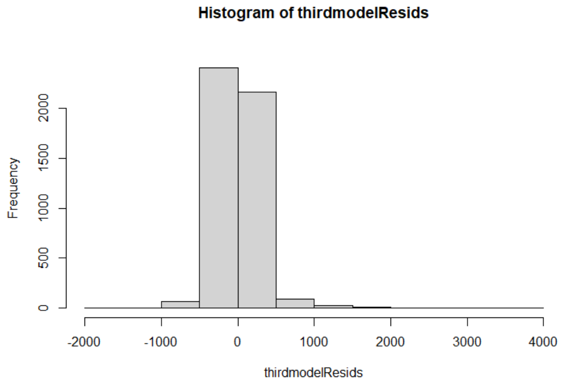
Second Model –

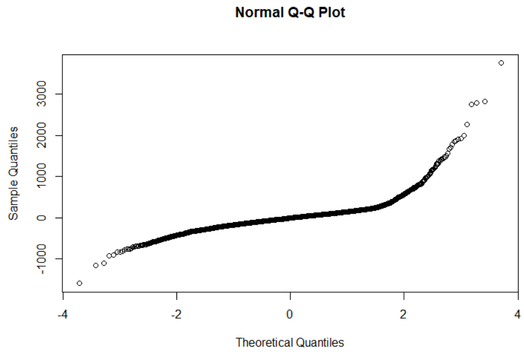


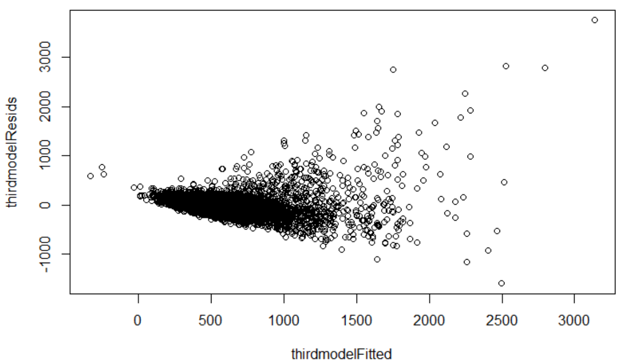




Third model –







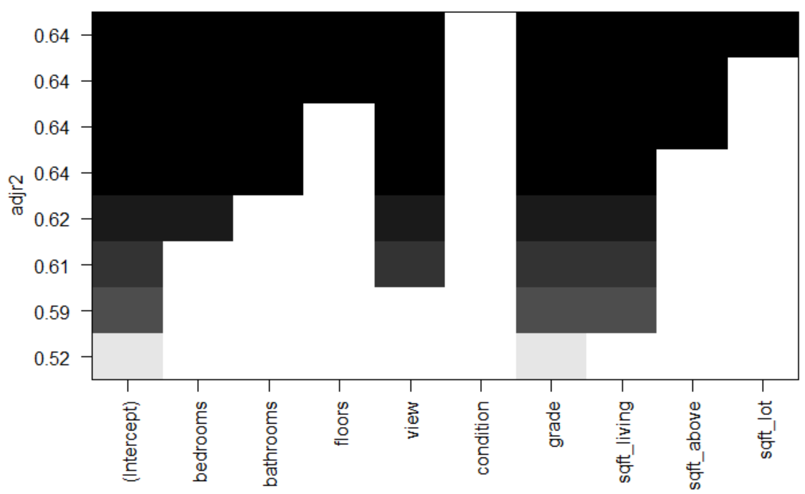
After checking the models, we can figure out that these are not the best models as they are not very much spread out and the QQ plot is not the idea one but lets check the Rsquare, RSME and MAE values before selecting.

I have placed all the three models values in a table to get a better understanding. –

Model	RSME	RSquared	MAE
1	290.1487	0.5260979	185.704
2	322.9214	0.4110043	213.2558
3	251.1471	0.646901	155.1098

The above table shoes that the best model out of these three is the third model with all the variables.

Lets try subset regression and AIC to compare and get the best models for this dataset and our prediction.



Now let's check the models by AIC.

```
Output -
Initial Model:
price ~ bedrooms + bathrooms + floors + view + condition + grade +
    sqft_living + sqft_above + sqft_lot

Final Model:
price ~ bedrooms + bathrooms + floors + view + condition + grade +
    sqft_living + sqft_above + sqft_lot
```

This method have given us two models – Initial and Final which both are the same to our third model. So, we are will select the third model which is the best one for this dataset.

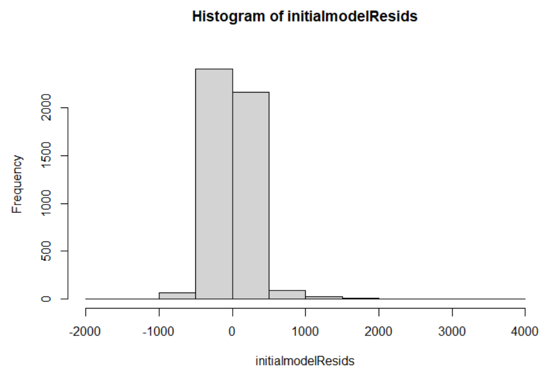
This model is not 100% accurate and it has some problems but in our case it is the most accurate model we can use to predict the prices of the houses.

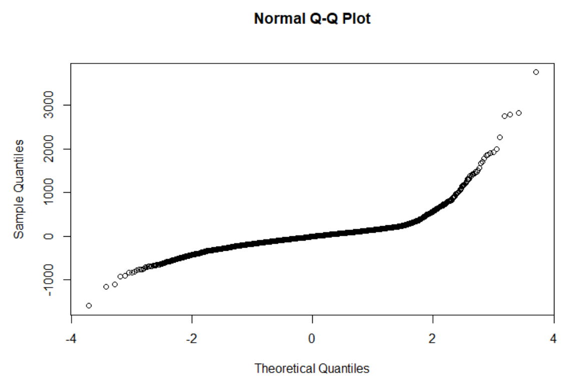
Model selection

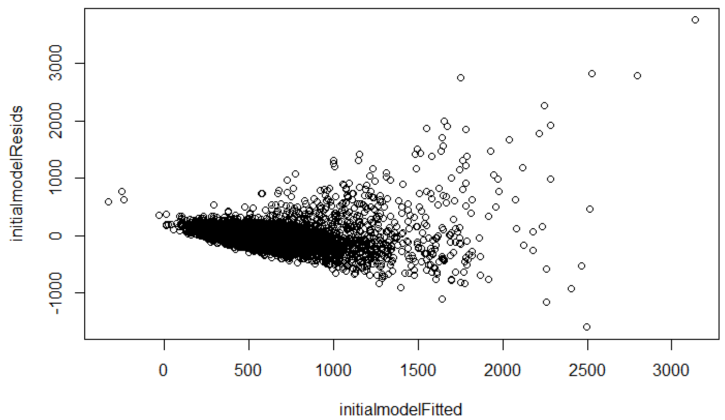
After experimenting with three different models containing square feet values, interior variables, and all variables, the most appropriate model for this dataset is the third one that includes all the variables. Upon performing the AIC analysis, we discovered that the model with the lowest AIC value was the same as our third model containing all variables.

Therefore, we will proceed to select the third model as the best fit for this dataset. Although the model may not be entirely accurate and may have some limitations, it is the most reliable option available for predicting house prices in our case.

Here are some diagnostics to finalize the model.







Here are the RSquare, RSME, MAE values for Final Model.

Model	RSME	RSquared	MAE
Final	252.7331	0.6421644	155.0578

Prediction

For our prediction, I have few variables values which will predict the prices of the houses accordingly.

Price	Bedroom	Bathrooms	Floors	View	Condition	Grade	Sqft_living	Sqft_above	Sqft_lot
1561.814	8	6	1	5	6	3	9500	9000	70000
1533.863	9	2	2	6	7	4	10000	9050	80000
2222.997	11	7	3	7	8	5	10500	9500	90000
2607.677	12	5	4	8	9	8	11000	10000	100000

Here price of the houses according to this specification will be $1561814.00, $1533863.00, $2222997.00, $2607677.00. The values that I got after the prediction model were divided by 1000 to simplify the calculation and convenience but now the actual values are multiplied by 1000.

Conclusion

In conclusion, this report aimed to predict house prices in King County, USA, using linear regression. The process involved testing and streamlining various models, which ultimately narrowed down to three models that yielded the desired prediction. Although the dataset had some limitations and required manipulation of columns and shortening, every step was taken to ensure the best possible results. After performing tests and analyzing the AIC models, the third model was the most promising, and it was used to develop a prediction model for various values of the variables. Despite the challenges, this report provides a valuable insight into predicting house prices in King County, USA, using linear regression.

There is always room for improvement in such analyses, and future studies could consider additional variables and techniques to refine the predictions further. However, this report provides valuable insights into predicting house prices in King County, USA, and the third model can be used as a reliable tool to make informed decisions regarding real estate investments. The results obtained from this study can help guide investors in making informed decisions and provide insights into the dynamics of the housing market in King County, USA.

References

1. Dataset page –

   https://www.kaggle.com/code/lashkingl/eda-kc-house-data/data

2. Conestoga d2l website –

   https://conestoga.desire2learn.com/d2l/home/687144

3. Data cleaning and manipulation –

   https://r4ds.had.co.nz/wrangle-intro.html

Code –

Project.R

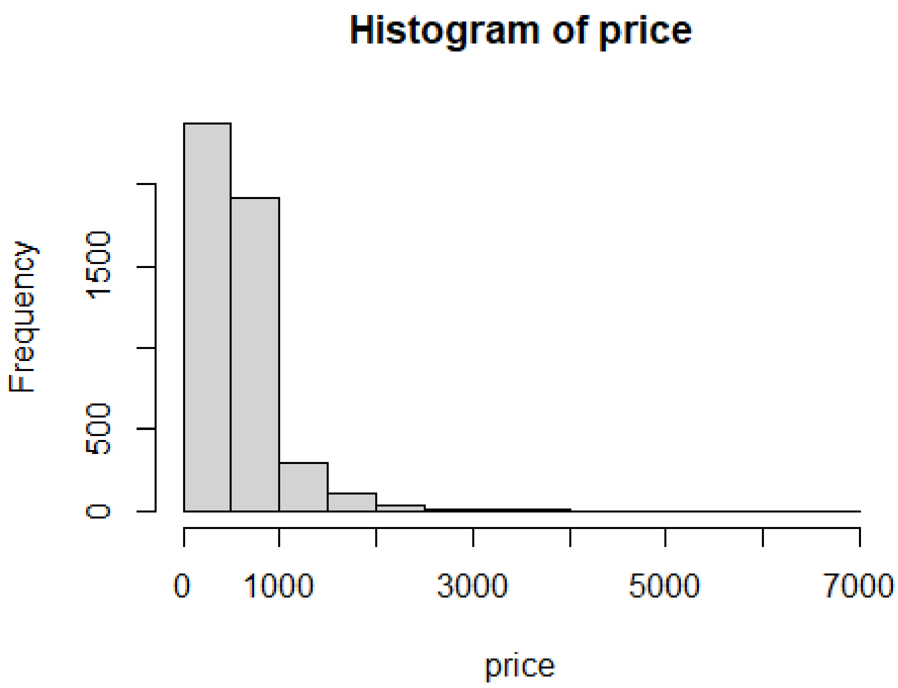
Divanshu Singh

2023-02-27

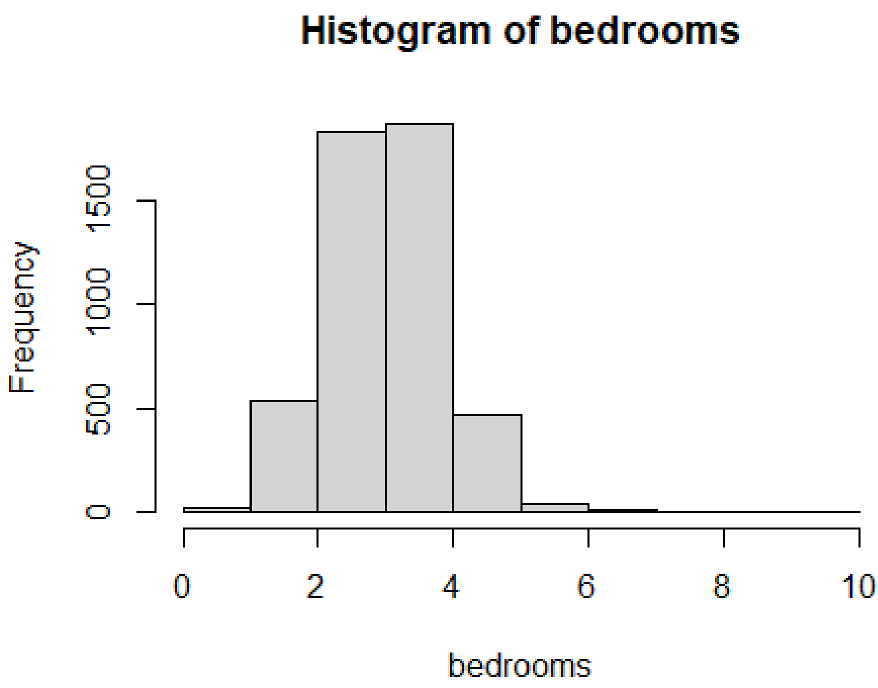
```
# Importing Libraries
library(readxl)
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tid
yverse 1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   1.0.1
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.5.0
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse
_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()

library(ISLR2)
library(stargazer)

##
## Please cite as:
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and S
ummary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=star
gazer

library(caret)

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift

library(leaps)
library(MASS)

##
## Attaching package: 'MASS'
##
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
##
## The following object is masked from 'package:dplyr':
##
##     select

# Importing Dataset
housing_data <- read_excel("D:/Predictive Analytics/Multivariate Sta
tistics/Project 1/King county, USA housing data.xlsx")
head(housing_data)

## # A tibble: 6 × 11
##   yr_built price bedrooms bathrooms sqft_li…¹ sqft_…² floors  vie
w condi…³ grade
##      <dbl> <dbl>    <dbl>     <dbl>     <dbl>   <dbl>  <dbl> <dbl
>   <dbl> <dbl>
## 1     2001 1225         4       4.5      5420  101930      1
0       3    11
## 2     2003  323         3       2.5      1890    6560      2
0       3     7
## 3     2005  719         4       2.5      2570    7173      2
0       3     8
## 4     2003  580.        3       2.5      2320    3980      2
0       3     8
## 5     2005  280         2       1.5      1190    1265      3
0       3     7
## 6     2000  625         4       2.5      2570    5520      2
0       3     9
## # … with 1 more variable: sqft_above <dbl>, and abbreviated varia
ble names
## #   ¹sqft_living, ²sqft_lot, ³condition

# Removing /Checking for Null Values
house_data <- na.omit(housing_data)
head(house_data)

## # A tibble: 6 × 11
##   yr_built price bedrooms bathrooms sqft_li…¹ sqft_…² floors  vie
w condi…³ grade
##      <dbl> <dbl>    <dbl>     <dbl>     <dbl>   <dbl>  <dbl> <dbl
>   <dbl> <dbl>
## 1     2001 1225         4       4.5      5420  101930      1
0       3    11
## 2     2003  323         3       2.5      1890    6560      2
0       3     7
## 3     2005  719         4       2.5      2570    7173      2
0       3     8
## 4     2003  580.        3       2.5      2320    3980      2
```

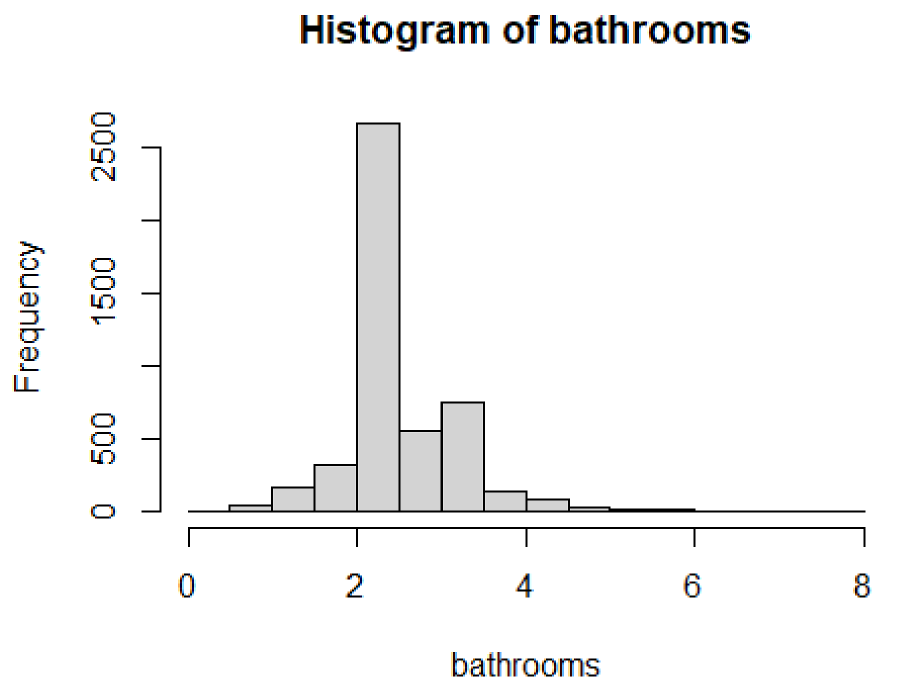
```
0       3     8
## 5     2005  280         2       1.5       1190    1265      3
0       3     7
## 6     2000  625         4       2.5       2570    5520      2
0       3     9
## # … with 1 more variable: sqft_above <dbl>, and abbreviated varia
ble names
## #   ¹sqft_living, ²sqft_lot, ³condition

# Summary of the dataset
summary(house_data)

##     yr_built        price           bedrooms        bathrooms
##  Min.   :2000   Min.   : 155.0   Min.   : 0.000   Min.   :0.000
##  1st Qu.:2004   1st Qu.: 375.0   1st Qu.: 3.000   1st Qu.:2.500
##  Median :2006   Median : 503.0   Median : 3.000   Median :2.500
##  Mean   :2007   Mean   : 618.4   Mean   : 3.495   Mean   :2.678
##  3rd Qu.:2010   3rd Qu.: 720.0   3rd Qu.: 4.000   3rd Qu.:3.000
##  Max.   :2015   Max.   :6885.0   Max.   :10.000   Max.   :7.750
##   sqft_living      sqft_lot           floors           view
##  Min.   : 384   Min.   :     572   Min.   :1.000   Min.   :0.0000
##  1st Qu.:1640   1st Qu.:    2500   1st Qu.:2.000   1st Qu.:0.0000
##  Median :2340   Median :    5000   Median :2.000   Median :0.0000
##  Mean   :2471   Mean   :   12239   Mean   :2.055   Mean   :0.1681
##  3rd Qu.:3080   3rd Qu.:    7236   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :9890   Max.   :1024068   Max.   :3.500   Max.   :4.0000
##    condition         grade          sqft_above
##  Min.   :3.000   Min.   : 4.000   Min.   : 384
##  1st Qu.:3.000   1st Qu.: 8.000   1st Qu.:1550
##  Median :3.000   Median : 8.000   Median :2240
##  Mean   :3.007   Mean   : 8.336   Mean   :2302
##  3rd Qu.:3.000   3rd Qu.: 9.000   3rd Qu.:2908
##  Max.   :5.000   Max.   :13.000   Max.   :8860

# Histogram for
attach(house_data)
hist(price)
```



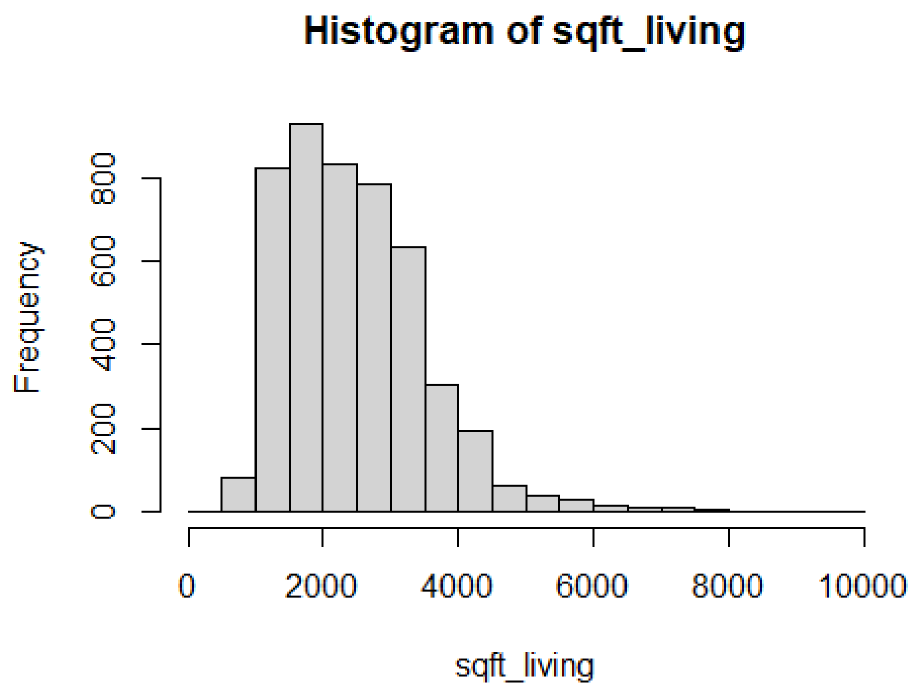

```
hist(bedrooms)
```



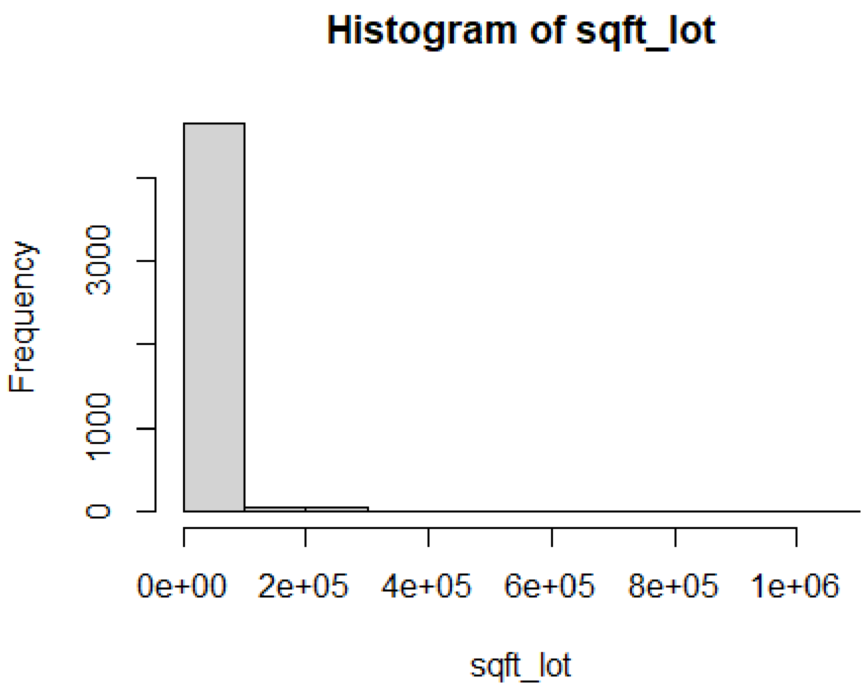

```
hist(bathrooms)
```



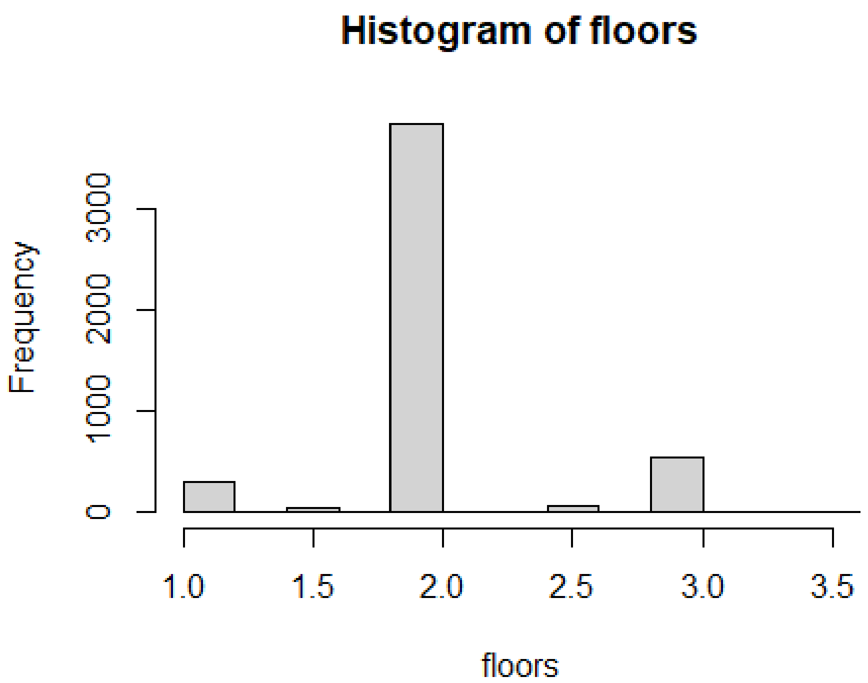

```
hist(sqft_living)
```



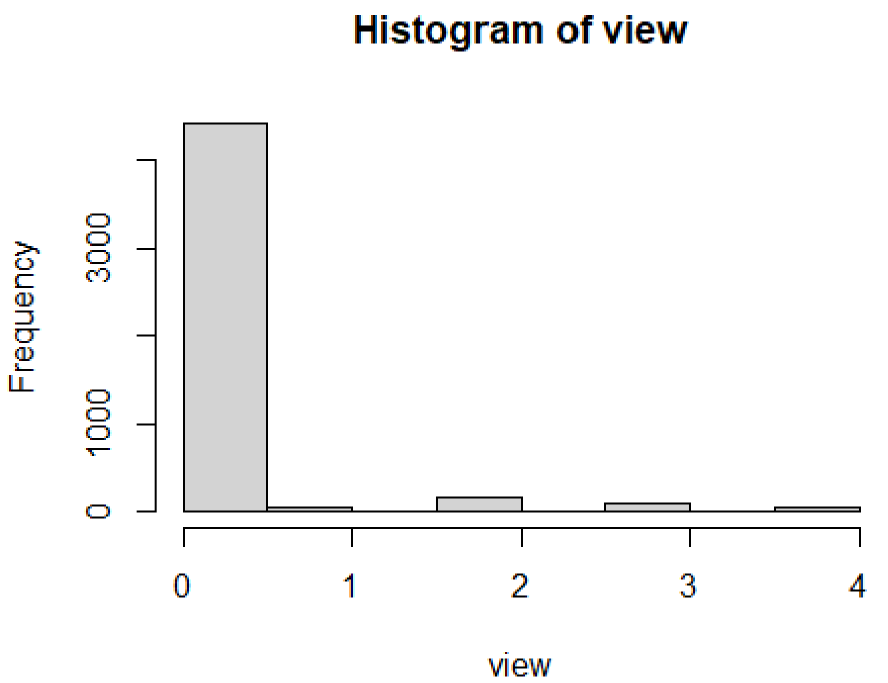

```
hist(sqft_lot)
```



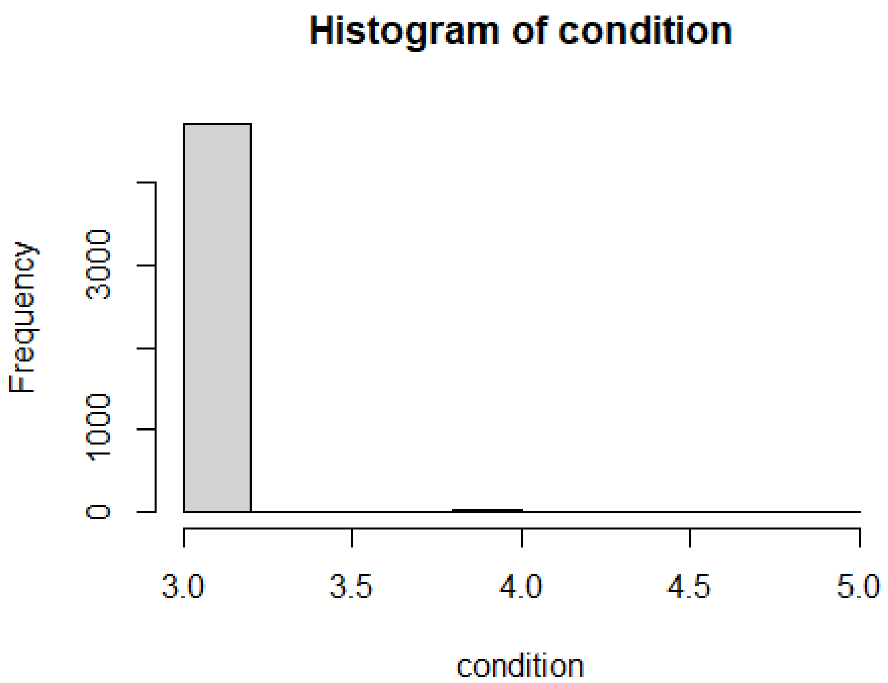

```
hist(floors)
```



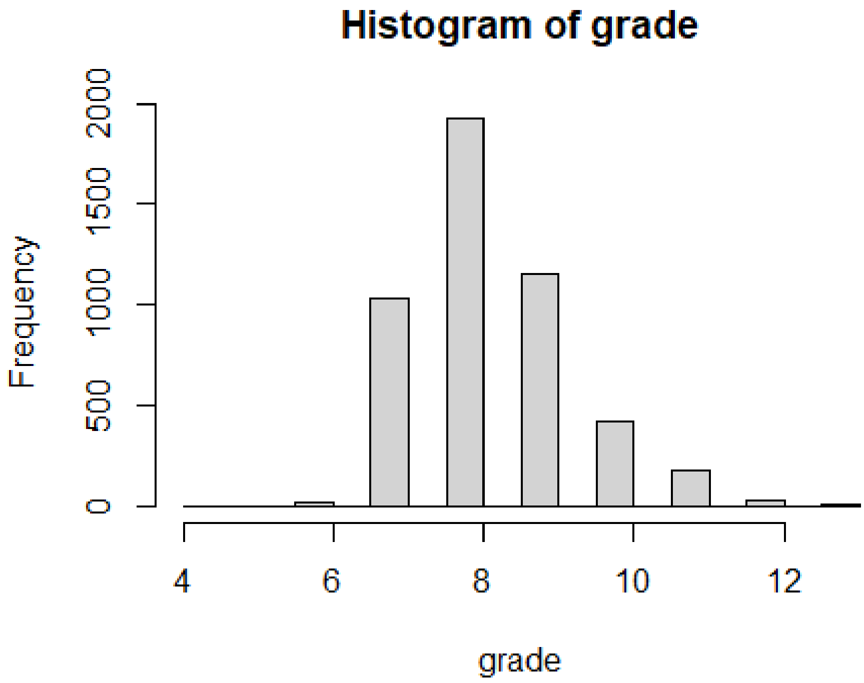

```
hist(view)
```



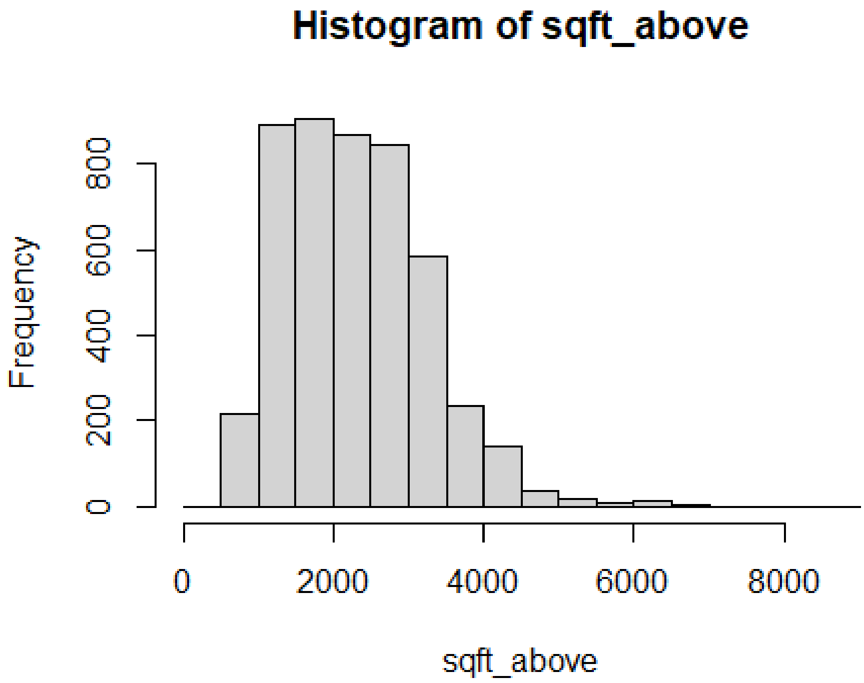

```
hist(condition)
```




```
hist(grade)
```

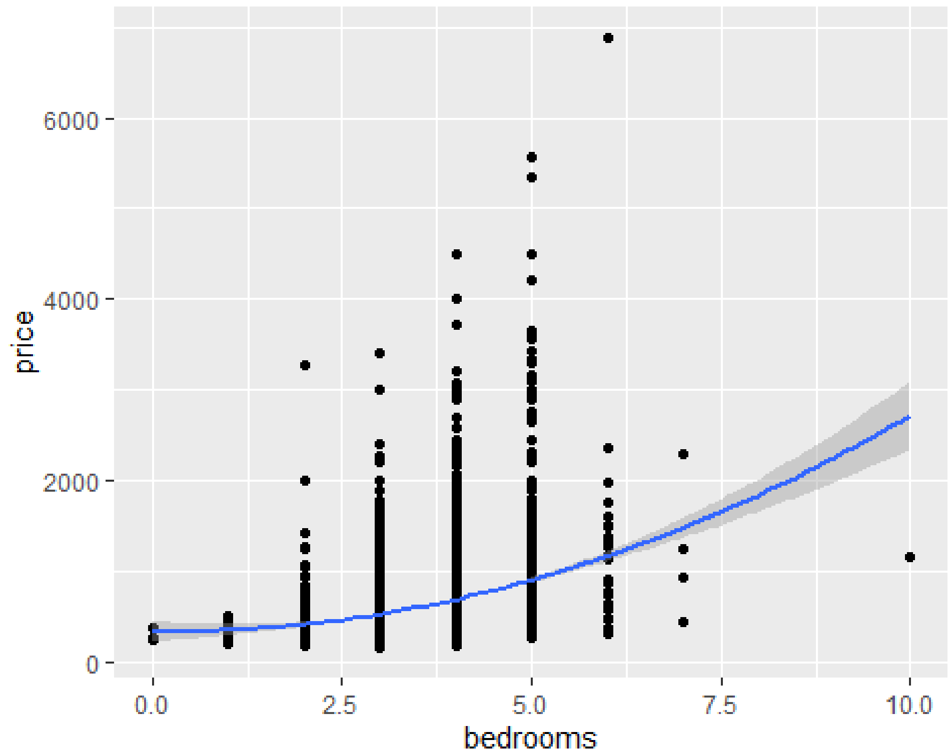



```
hist(sqft_above)
```

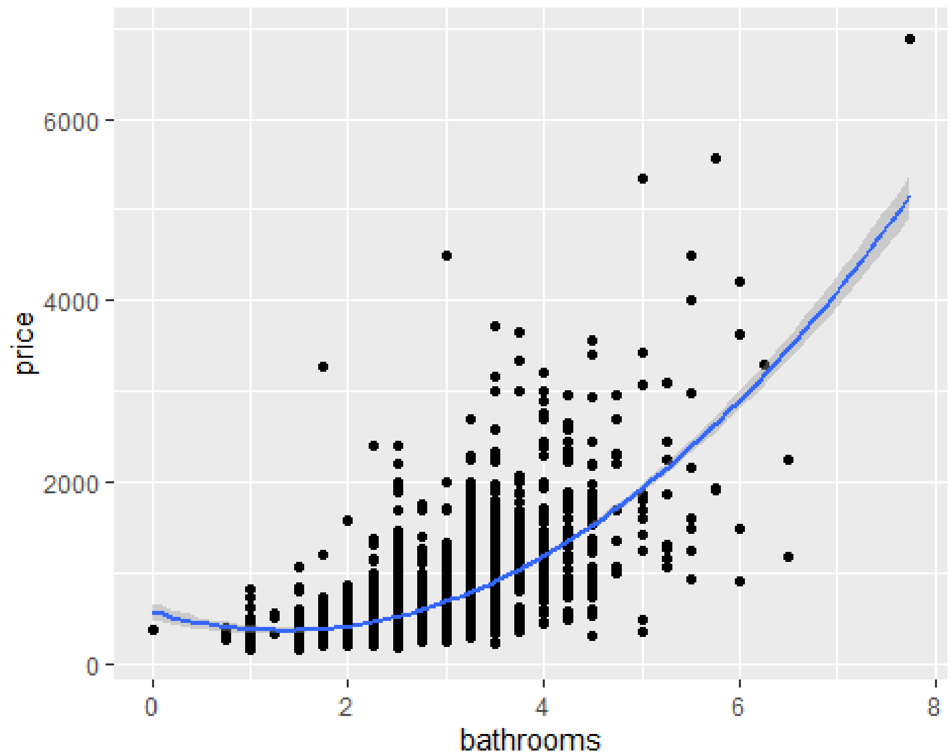



```
# Correlation
cor(price, bedrooms)

## [1] 0.3437008
```

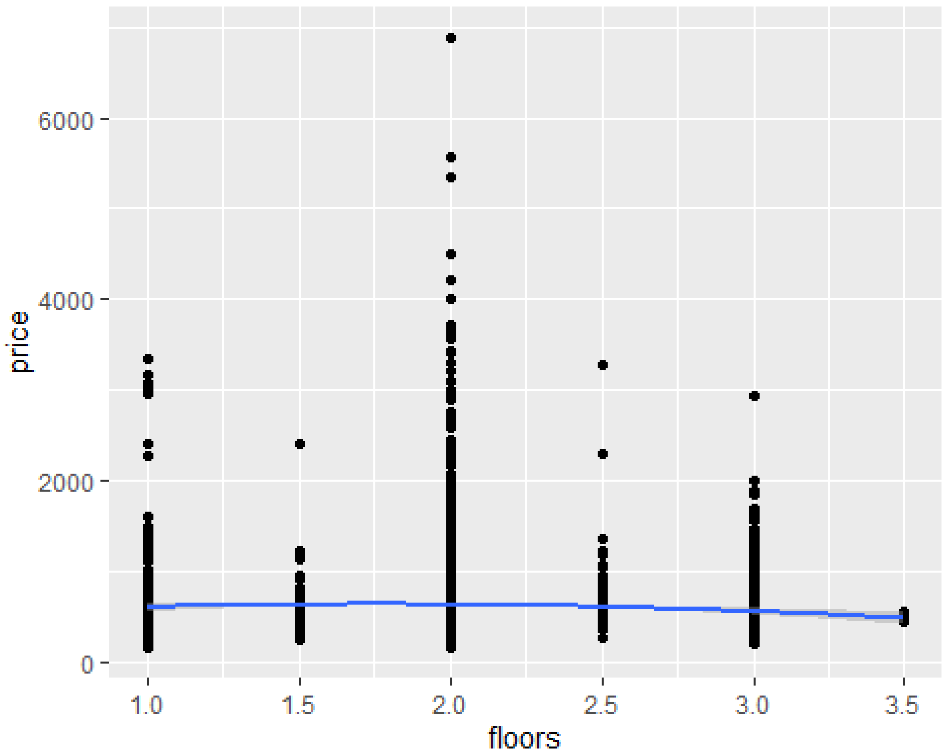
```
cor(price, bathrooms)

## [1] 0.5946179

cor(price, sqft_living)

## [1] 0.7129826

cor(price, sqft_lot)

## [1] 0.155032

cor(price, floors)

## [1] -0.03726097

cor(price, view)

## [1] 0.3892226

cor(price, condition)

## [1] 0.02040599

cor(price, grade)

## [1] 0.7186924

cor(price, sqft_above)

## [1] 0.6240518

# GGplot
house_data %>% ggplot(aes(x = bedrooms, y = price)) + geom_point() +
geom_smooth(method="lm", formula = "y ~ x + I(x^2)")
```



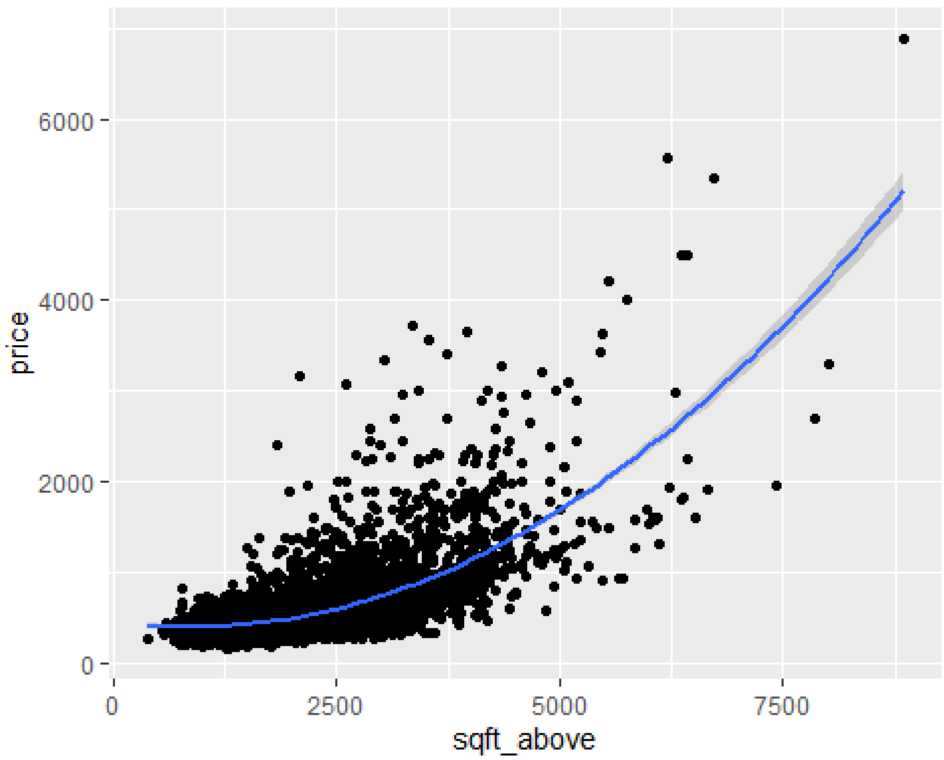

```
house_data %>% ggplot(aes(x = bathrooms, y = price)) + geom_point()
+ geom_smooth(method="lm", formula = "y ~ x + I(x^2)")
```



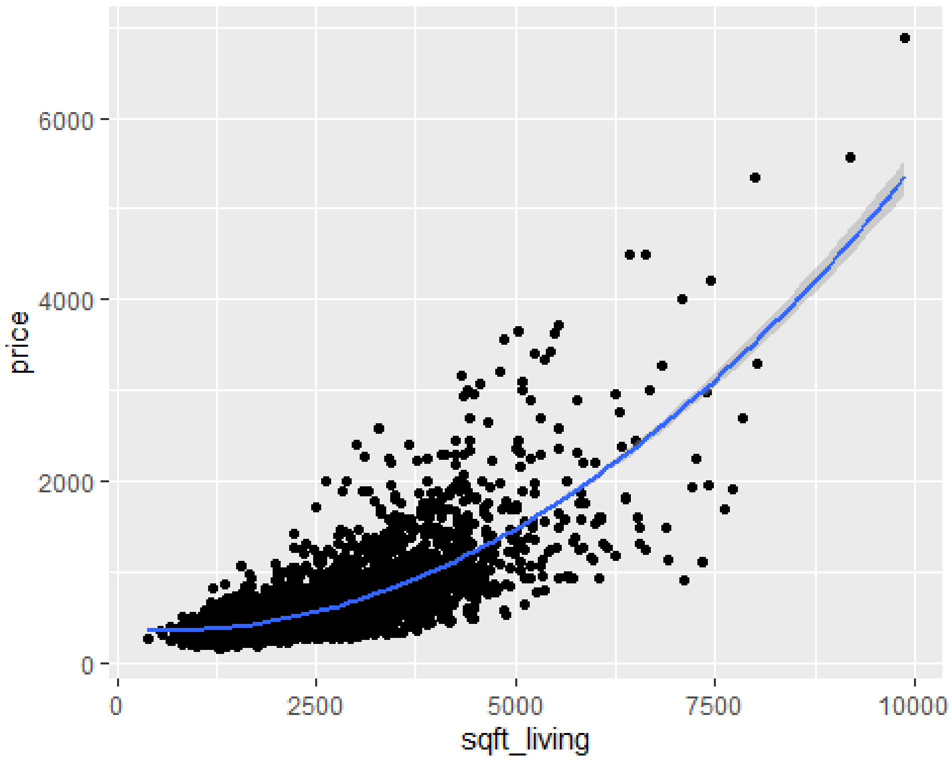

```
house_data %>% ggplot(aes(x = floors, y = price)) + geom_point() + g
eom_smooth(method="lm", formula = "y ~ x + I(x^2)")
```



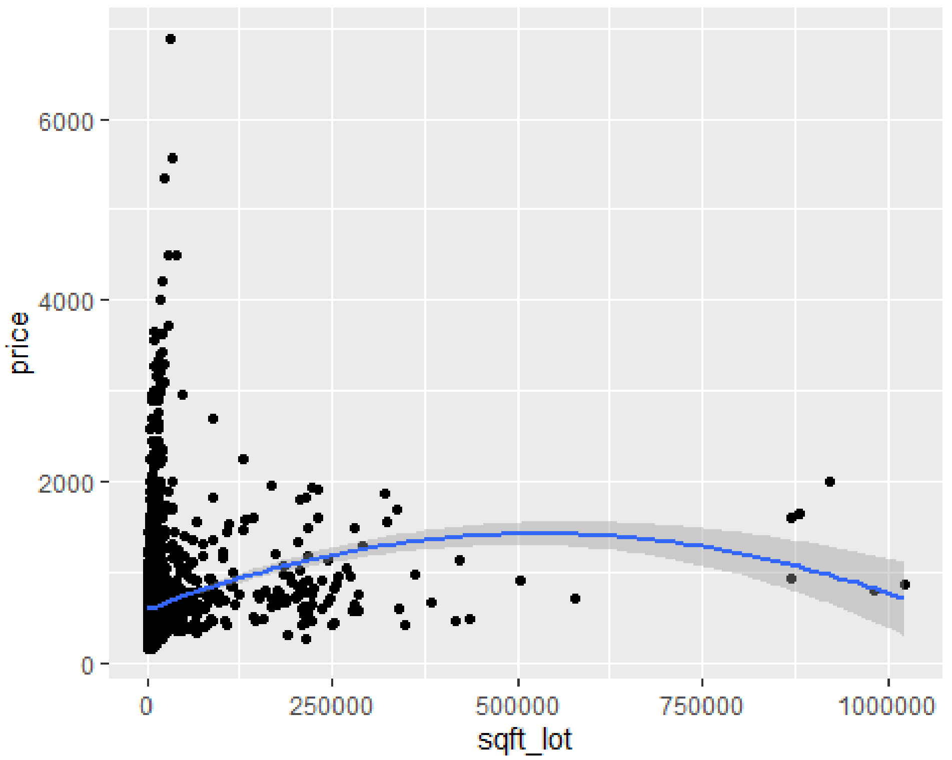

```
house_data %>% ggplot(aes(x = sqft_above, y = price)) + geom_point()
+ geom_smooth(method="lm", formula = "y ~ x + I(x^2)")
```



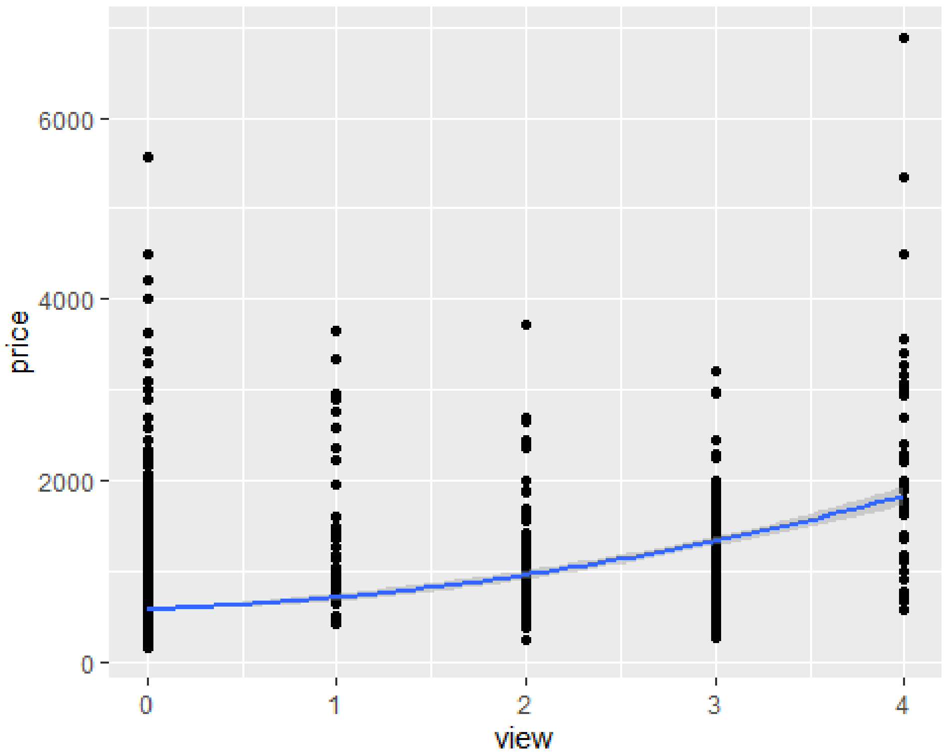

```
house_data %>% ggplot(aes(x = sqft_living, y = price)) + geom_point(
) + geom_smooth(method="lm", formula = "y ~ x + I(x^2)")
```



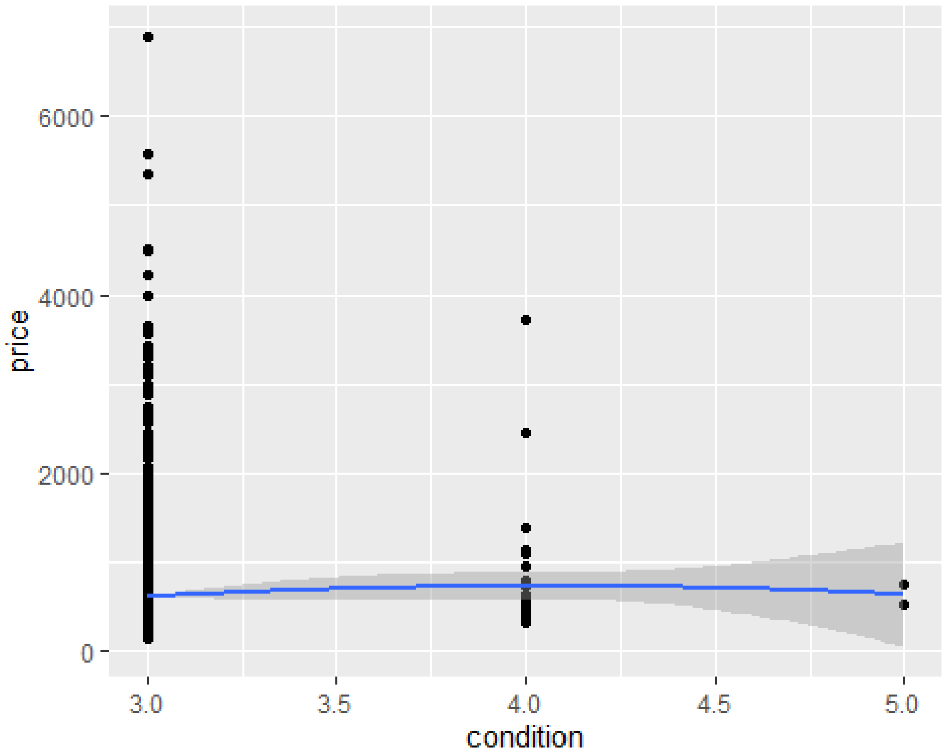

```
house_data %>% ggplot(aes(x = sqft_lot, y = price)) + geom_point() +
geom_smooth(method="lm", formula = "y ~ x + I(x^2)")
```



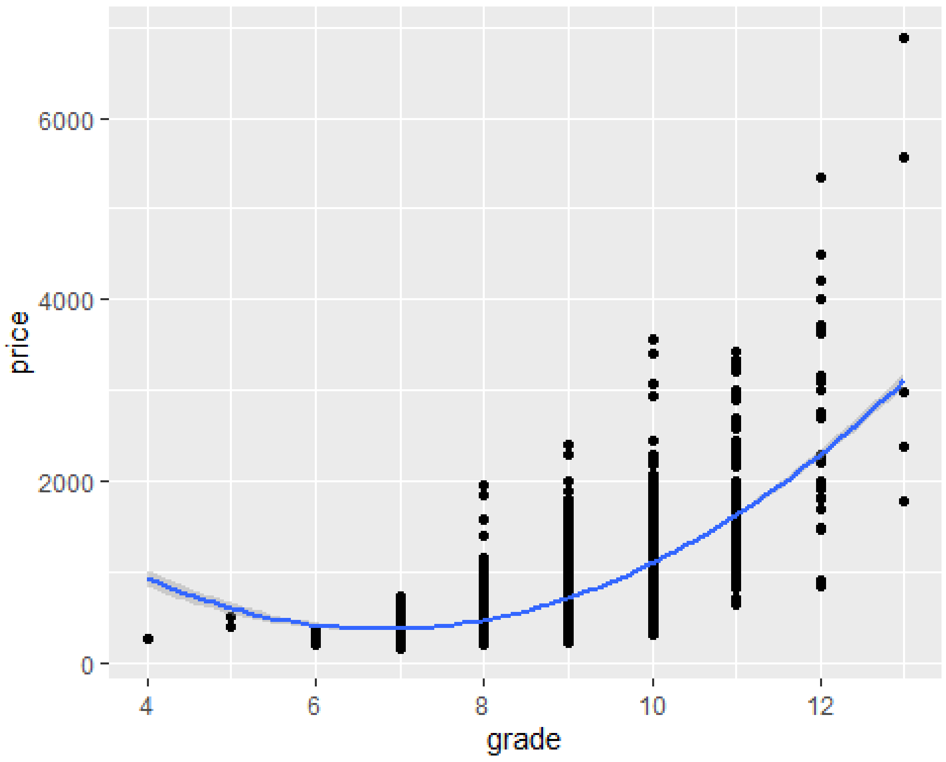

```
house_data %>% ggplot(aes(x = view, y = price)) + geom_point() + geo
m_smooth(method="lm", formula = "y ~ x + I(x^2)")
```




```
house_data %>% ggplot(aes(x = condition, y = price)) + geom_point()
+ geom_smooth(method="lm", formula = "y ~ x + I(x^2)")
```

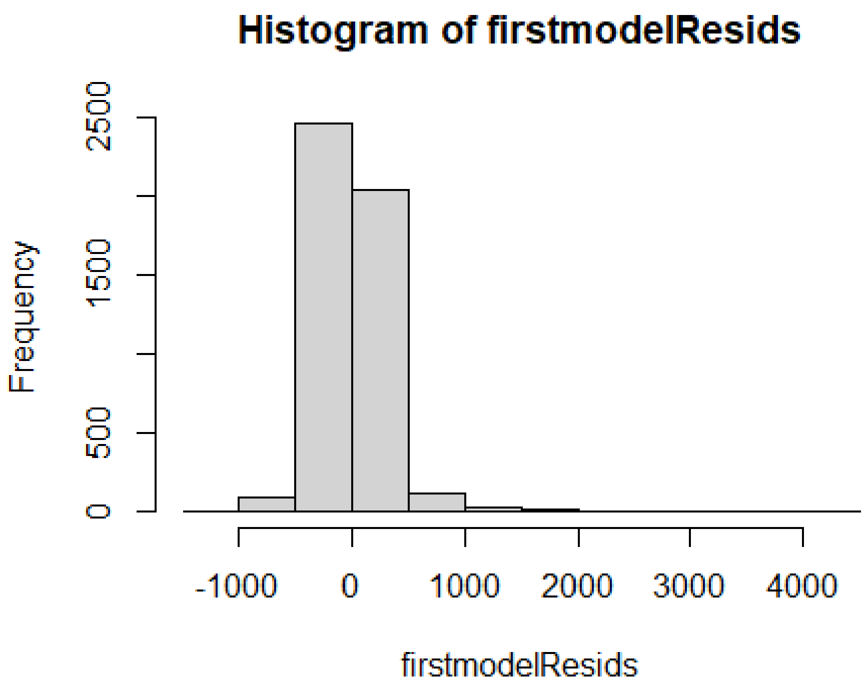



```
house_data %>% ggplot(aes(x = grade, y = price)) + geom_point() + ge
om_smooth(method="lm", formula = "y ~ x + I(x^2)")
```

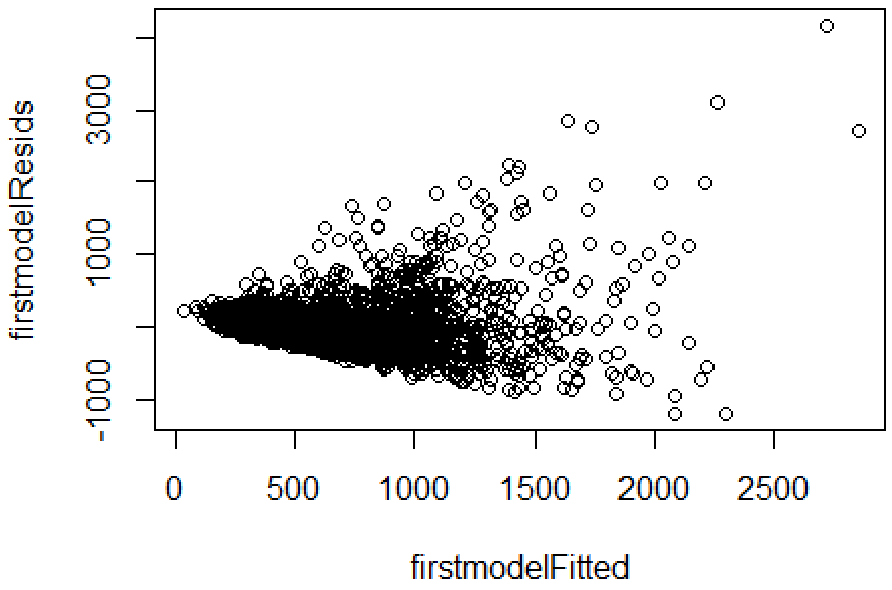



```
# First Model with all the Square feet independent variables
first_model <- lm(price ~ sqft_living + sqft_above + sqft_lot, data
= house_data)

coef(first_model)

##   (Intercept)   sqft_living    sqft_above      sqft_lot
## -6.368873e+01  4.248787e-01 -1.596033e-01 -3.971134e-05

summary(first_model)

##
## Call:
## lm(formula = price ~ sqft_living + sqft_above + sqft_lot, data =
house_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1190.6  -150.8   -21.5   111.1  4162.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.369e+01  1.122e+01  -5.677 1.45e-08 ***
## sqft_living  4.249e-01  1.158e-02  36.675  < 2e-16 ***
## sqft_above  -1.596e-01  1.273e-02 -12.536  < 2e-16 ***
## sqft_lot    -3.971e-05  9.187e-05  -0.432    0.666
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

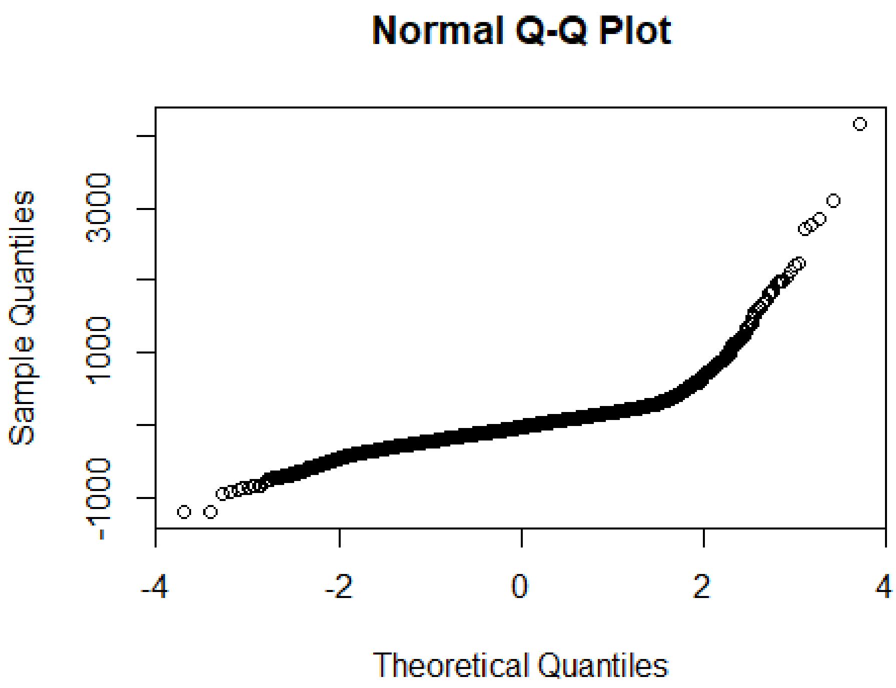
```
## Residual standard error: 291.5 on 4755 degrees of freedom
## Multiple R-squared:  0.5242, Adjusted R-squared:  0.5239
## F-statistic:  1746 on 3 and 4755 DF,  p-value: < 2.2e-16

firstmodelResids <- first_model$residuals
firstmodelFitted <- first_model$fitted.values

hist(firstmodelResids)
```




```
plot(firstmodelFitted, firstmodelResids)
```




```
qqnorm(firstmodelResids)
```

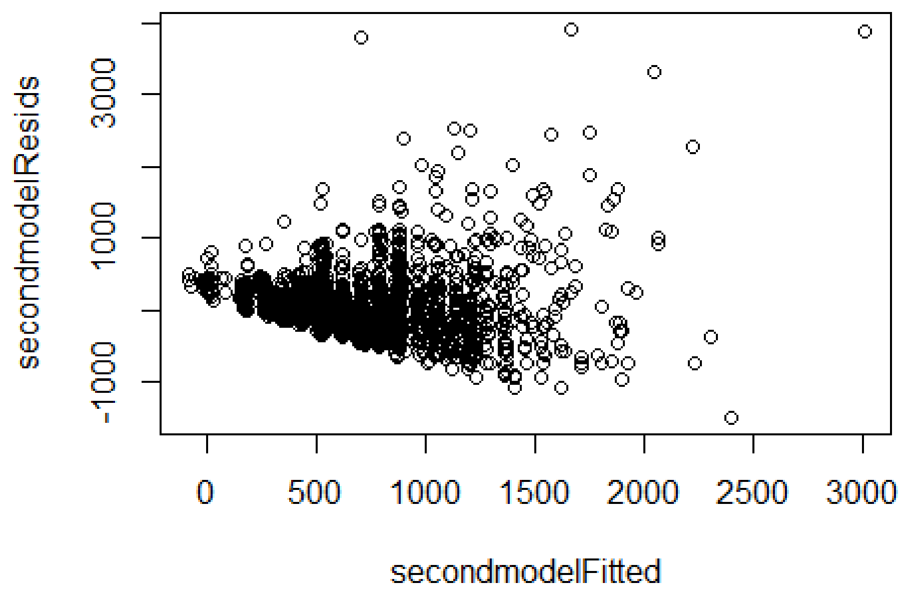



```
# Training the first model
firstCVModel <- train(
  form = price ~ sqft_living + sqft_above + sqft_lot,
  data = house_data,
  method = "lm",
```

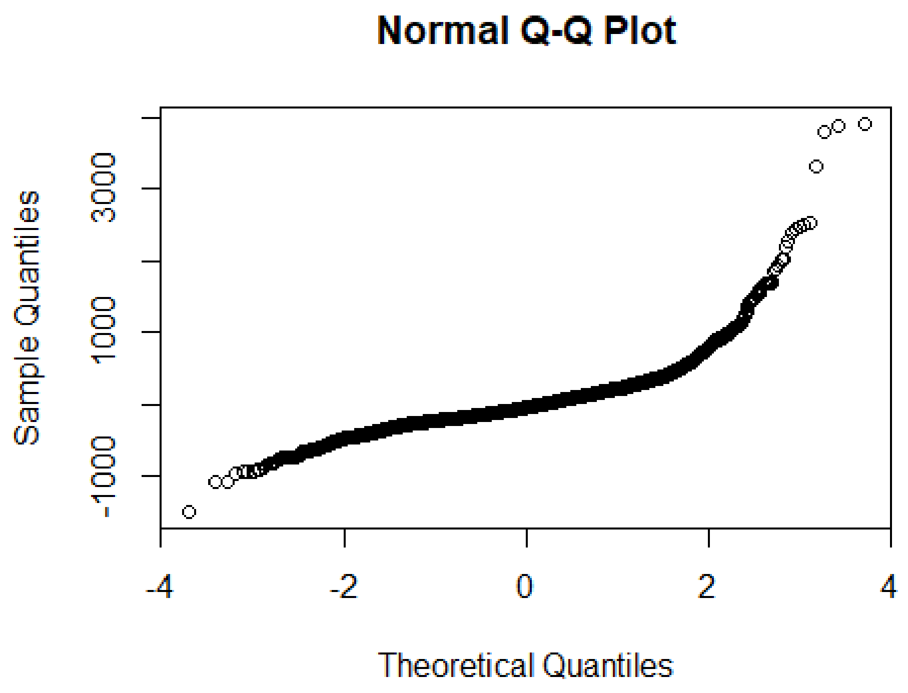
```
  trControl = trainControl(method = "cv", number = 10)
)
firstCVModel

## Linear Regression
##
## 4759 samples
##    3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4284, 4282, 4284, 4283, 4284, 4283, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   290.9796  0.5252852  185.835
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

# Second model with all the interior variables
second_model <- lm(price ~ bedrooms + bathrooms + floors + view, dat
a = house_data)
coef(second_model)

## (Intercept)    bedrooms   bathrooms      floors        view
## -314.364490    4.838902  346.669865  -19.272707  161.529068

summary(second_model)

##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + floors + view, data =
house_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1497.4  -181.5   -45.6   130.0  3905.4
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -314.364     33.233  -9.459   <2e-16 ***
## bedrooms       4.839      6.582   0.735   0.4623
## bathrooms    346.670      9.238  37.528   <2e-16 ***
## floors       -19.273     11.227  -1.717   0.0861 .
## view         161.529      7.437  21.721   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 323.9 on 4754 degrees of freedom
## Multiple R-squared:  0.4125, Adjusted R-squared:  0.412
## F-statistic: 834.3 on 4 and 4754 DF,  p-value: < 2.2e-16
```

```
secondmodelResids <- second_model$residuals
secondmodelFitted <- second_model$fitted.values

hist(secondmodelResids)
```


```
plot(secondmodelFitted, secondmodelResids)
```

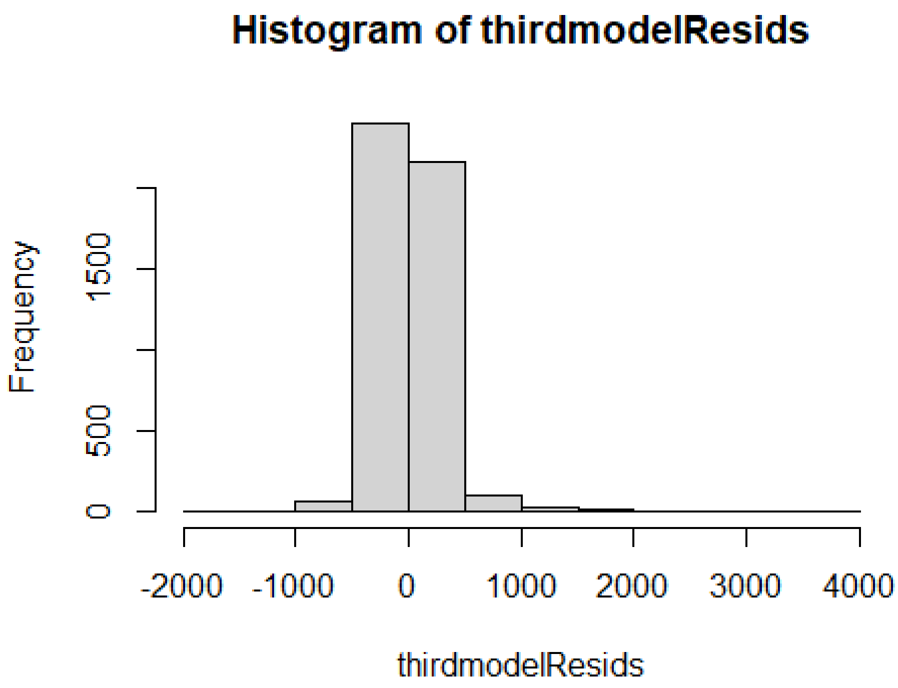


```
qqnorm(secondmodelResids)
```

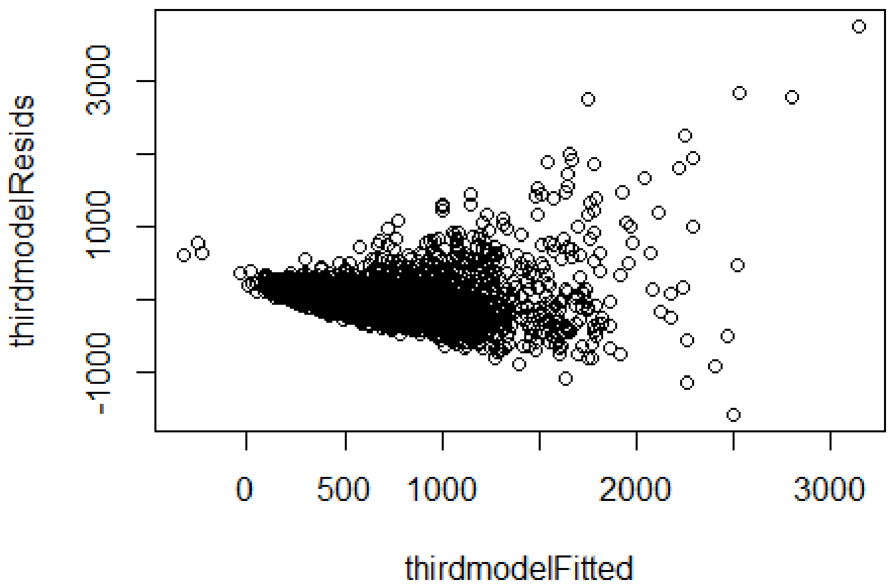



```
# Training the second model
secondCVModel <- train(
  form = price ~ bedrooms + bathrooms + floors + view,
  data = house_data,
  method = "lm",
  trControl = trainControl(method = "cv", number = 10)
)
secondCVModel

## Linear Regression
##
## 4759 samples
##    4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4283, 4283, 4283, 4284, 4283, 4284, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   323.3085  0.4128688  213.5115
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

# Third Model
third_model <- lm(price ~ bedrooms + bathrooms + floors + view + con
dition + grade + sqft_living + sqft_above + sqft_lot, data = house_d
```

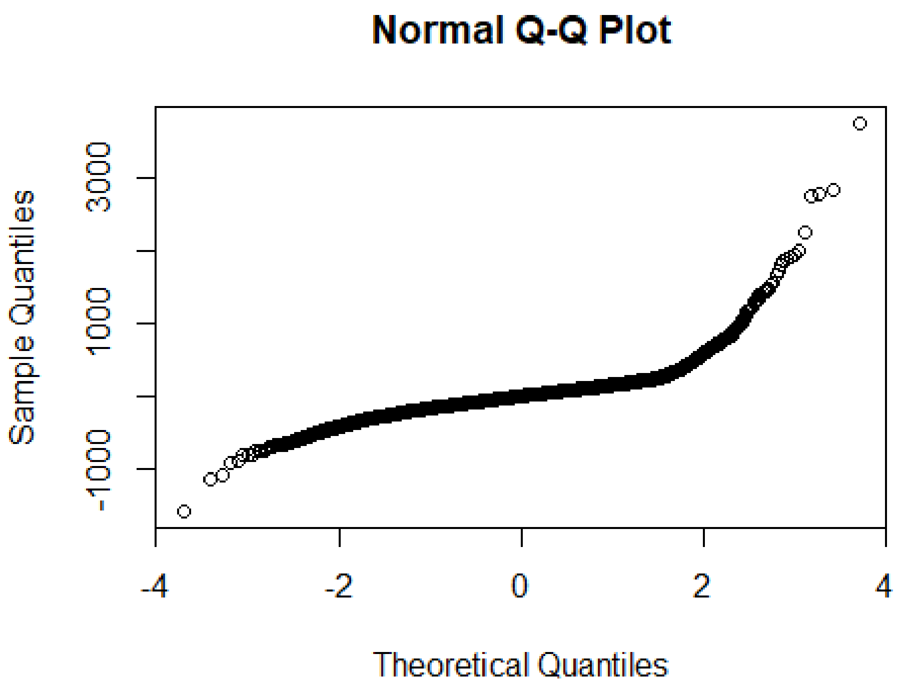
```
ata)
coef(third_model)

##   (Intercept)      bedrooms     bathrooms        floors
view
## -1.156679e+03 -8.810762e+01  9.329996e+01  4.312113e+01  7.330982
e+01
##     condition         grade   sqft_living    sqft_above      sqft
_lot
##  6.092635e+01  1.324256e+02  2.620402e-01 -8.626652e-02 -3.133347
e-04

summary(third_model)

##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + floors + view + condi
tion +
##     grade + sqft_living + sqft_above + sqft_lot, data = house_dat
a)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1594.2  -118.9    -7.5    91.3  3746.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.157e+03  1.314e+02  -8.800  < 2e-16 ***
## bedrooms    -8.811e+01  6.003e+00 -14.677  < 2e-16 ***
## bathrooms    9.330e+01  8.724e+00  10.695  < 2e-16 ***
## floors       4.312e+01  9.138e+00   4.719 2.44e-06 ***
## view         7.331e+01  6.083e+00  12.051  < 2e-16 ***
## condition    6.093e+01  4.115e+01   1.481 0.138743
## grade        1.324e+02  5.113e+00  25.898  < 2e-16 ***
## sqft_living  2.620e-01  1.303e-02  20.107  < 2e-16 ***
## sqft_above  -8.627e-02  1.171e-02  -7.364 2.09e-13 ***
## sqft_lot    -3.133e-04  8.067e-05  -3.884 0.000104 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 252.1 on 4749 degrees of freedom
## Multiple R-squared:  0.6445, Adjusted R-squared:  0.6438
## F-statistic: 956.6 on 9 and 4749 DF,  p-value: < 2.2e-16

thirdmodelResids <- third_model$residuals
thirdmodelFitted <- third_model$fitted.values

hist(thirdmodelResids)
```




```
plot(thirdmodelFitted, thirdmodelResids)
```

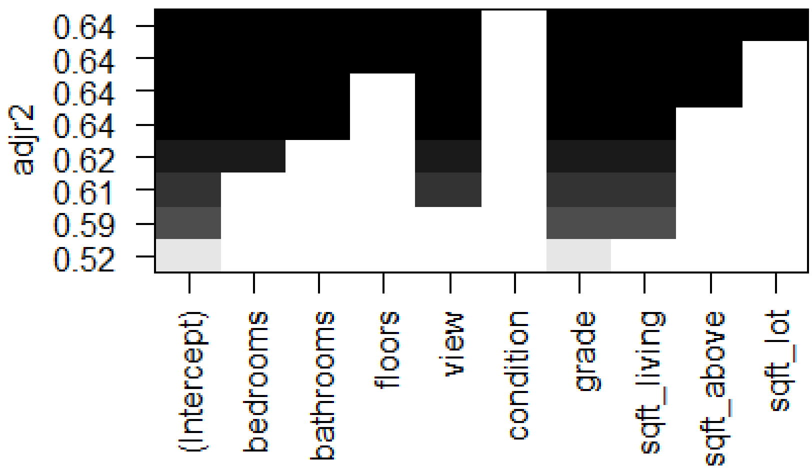



```
qqnorm(thirdmodelResids)
```



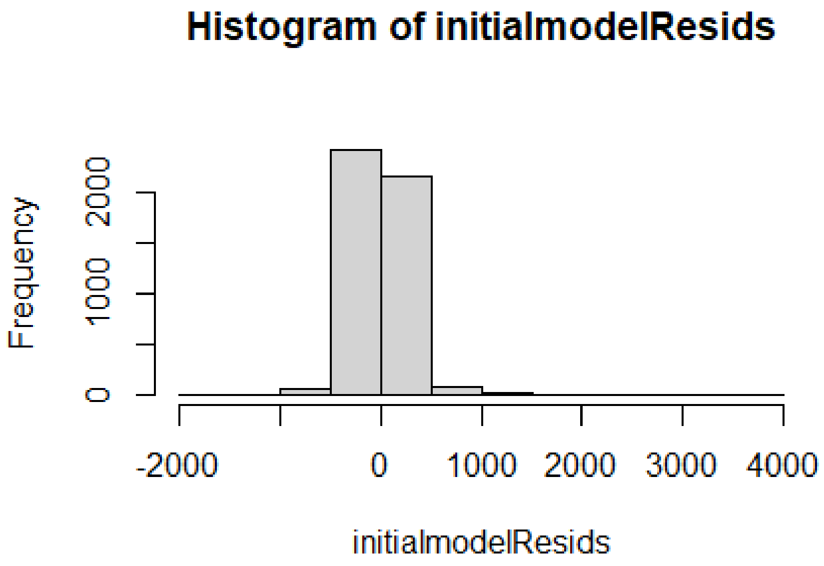

```
# Training the Third model
thirdCVModel <- train(
  form = price ~ bedrooms + bathrooms + floors + view + condition +
grade + sqft_living + sqft_above + sqft_lot,
  data = house_data,
  method = "lm",
  trControl = trainControl(method = "cv", number = 10)
)
thirdCVModel

## Linear Regression
##
## 4759 samples
##    9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4284, 4282, 4284, 4283, 4284, 4283, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   251.8467  0.6427208  154.8474
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

# Using sub-setting
subsetmodel <- regsubsets(price ~ bedrooms + bathrooms + floors + vi
ew + condition + grade + sqft_living + sqft_above + sqft_lot, data =
```

```
house_data)
plot(subsetmodel, scale = "adjr2")
```

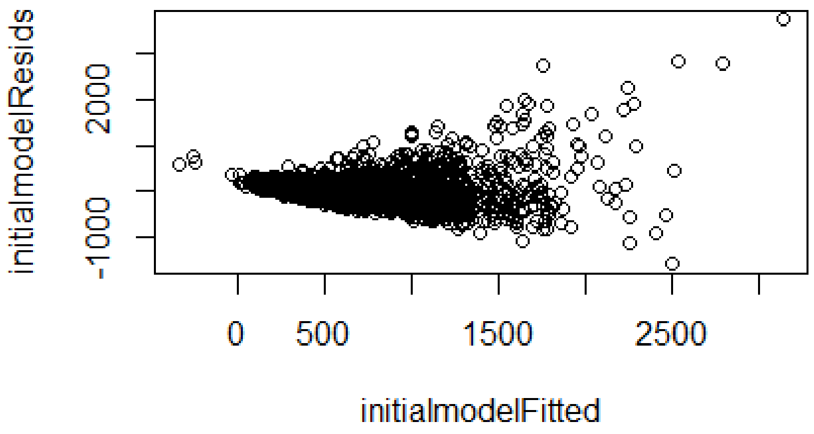



```
# AIC best model
AIC <- lm(price ~ bedrooms + bathrooms + floors + view + condition +
grade + sqft_living + sqft_above + sqft_lot, data = house_data)
step <- stepAIC(AIC, trace = FALSE)
step$anova

## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## price ~ bedrooms + bathrooms + floors + view + condition + grade
+
##     sqft_living + sqft_above + sqft_lot
##
## Final Model:
## price ~ bedrooms + bathrooms + floors + view + condition + grade
+
##     sqft_living + sqft_above + sqft_lot
##
##
##   Step Df Deviance Resid. Df Resid. Dev      AIC
## 1                       4749  301866742 52643.61

# Initial Model:
initial_model <- lm(price ~ bedrooms + bathrooms + floors + view + c
ondition + grade + sqft_living + sqft_above + sqft_lot, data = house
```

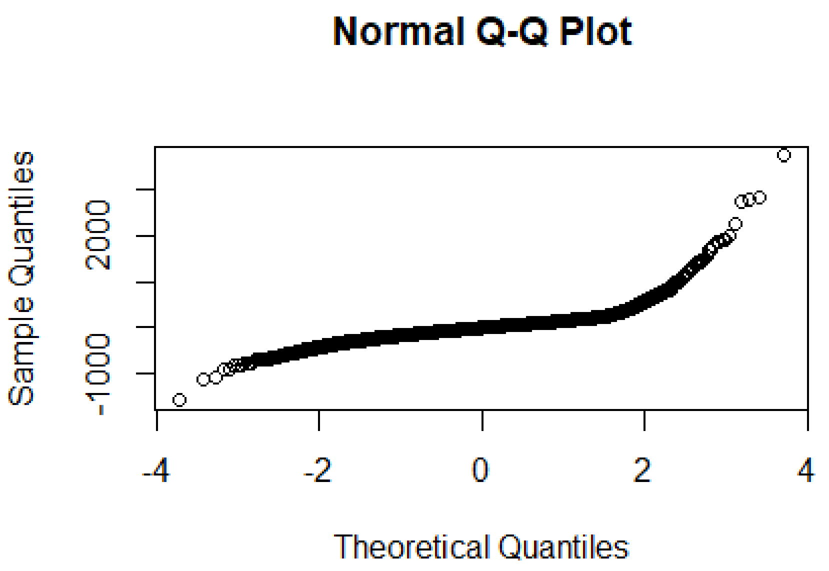
```
_data)
coef(initial_model)

##   (Intercept)      bedrooms     bathrooms        floors
view
## -1.156679e+03 -8.810762e+01  9.329996e+01  4.312113e+01  7.330982
e+01
##     condition         grade   sqft_living    sqft_above      sqft
_lot
##  6.092635e+01  1.324256e+02  2.620402e-01 -8.626652e-02 -3.133347
e-04

summary(initial_model)

##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + floors + view + condi
tion +
##     grade + sqft_living + sqft_above + sqft_lot, data = house_dat
a)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1594.2  -118.9    -7.5    91.3  3746.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.157e+03  1.314e+02  -8.800  < 2e-16 ***
## bedrooms    -8.811e+01  6.003e+00 -14.677  < 2e-16 ***
## bathrooms    9.330e+01  8.724e+00  10.695  < 2e-16 ***
## floors       4.312e+01  9.138e+00   4.719 2.44e-06 ***
## view         7.331e+01  6.083e+00  12.051  < 2e-16 ***
## condition    6.093e+01  4.115e+01   1.481 0.138743
## grade        1.324e+02  5.113e+00  25.898  < 2e-16 ***
## sqft_living  2.620e-01  1.303e-02  20.107  < 2e-16 ***
## sqft_above  -8.627e-02  1.171e-02  -7.364 2.09e-13 ***
## sqft_lot    -3.133e-04  8.067e-05  -3.884 0.000104 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 252.1 on 4749 degrees of freedom
## Multiple R-squared:  0.6445, Adjusted R-squared:  0.6438
## F-statistic: 956.6 on 9 and 4749 DF,  p-value: < 2.2e-16

initialmodelResids <- initial_model$residuals
initialmodelFitted <- initial_model$fitted.values

hist(initialmodelResids)
```




```
plot(initialmodelFitted, initialmodelResids)
```




```
qqnorm(initialmodelResids)
```




```
# Training Initial model
InitialCVModel <- train(
  form = price ~ bedrooms + bathrooms + floors + view + condition +
grade + sqft_living + sqft_above + sqft_lot,
  data = house_data,
  method = "lm",
  trControl = trainControl(method = "cv", number = 10)
)
InitialCVModel

## Linear Regression
##
## 4759 samples
##    9 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4283, 4284, 4283, 4284, 4282, 4284, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   251.7113  0.6465939  155.0141
##
## Tuning parameter 'intercept' was held constant at a value of TRUE

# Prediction using Initial model
pricePrediction<- data.frame(bedrooms = c(8, 9, 11, 12), bathrooms =
c(6,2,7,5), floors = c(1,2,3,4), view = c(5,6,7,8), condition = c(6,
```

```
7, 8, 9),
                              grade = c(3, 4, 5, 8), sqft_living = c(
9500, 10000, 10500, 11000), sqft_above = c(9000, 9050, 9500, 10000),
                              sqft_lot = c(70000, 80000, 90000, 10000
0))
predict(initial_model, pricePrediction)
```

```
##        1        2        3        4
## 1561.814 1533.863 2222.997 2607.677
```