# House Sales Data Analysis in King County, WA

Christopher Wong

Created: 2023-04-25 Updated: 2025-01-18

Online property companies use machine learning techniques to provide house valuations. This project aims to predict house sales in King County, Washington State, USA using Multiple Linear Regression (MLR). The dataset includes historical records of houses sold between May 2014 and May 2015. This project was created by Christopher Wong, for Professor You Liang (course MTH404).

## DATA

The data for this project was sourced from the Kaggle dataset titled "KC_Housesales_Data". The dataset can be accessed at: https://www.kaggle.com/swathiachath/kc-housesales-data

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
```

```
## Warning: package 'readr' was built under R version 4.4.2
```

```
## Warning: package 'purrr' was built under R version 4.4.2
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
## Warning: package 'stringr' was built under R version 4.4.2
```

```
## Warning: package 'forcats' was built under R version 4.4.2
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

```
## ── Attaching core tidyverse packages ──────────────────────────────────
tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.4     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
```

```
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.4.2

## corrplot 0.95 loaded

library(lubridate)
library(readr)
library(caTools)

## Warning: package 'caTools' was built under R version 4.4.2

library(GGally)

## Warning: package 'GGally' was built under R version 4.4.2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(caret)

## Warning: package 'caret' was built under R version 4.4.2

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

library(leaps)

## Warning: package 'leaps' was built under R version 4.4.2

house_data <- read_csv("kc_house_data.csv")

## Rows: 21597 Columns: 21
## ── Column specification
──────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): date
## dbl (20): id, price, bedrooms, bathrooms, sqft_living, sqft_lot, floors,
wat...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```r
# Display the first few rows and structure of the data
head(house_data)
```

```
## # A tibble: 6 × 21
##         id date   price bedrooms bathrooms sqft_living sqft_lot floors
waterfront
##      <dbl> <chr>  <dbl>    <dbl>     <dbl>       <dbl>    <dbl>  <dbl>
<dbl>
## 1  7.13e9 10/1… 2.22e5        3      1           1180     5650      1
0
## 2  6.41e9 12/9… 5.38e5        3      2.25        2570     7242      2
0
## 3  5.63e9 2/25… 1.80e5        2      1            770    10000      1
0
## 4  2.49e9 12/9… 6.04e5        4      3           1960     5000      1
0
## 5  1.95e9 2/18… 5.10e5        3      2           1680     8080      1
0
## 6  7.24e9 5/12… 1.23e6        4      4.5         5420   101930      1
0
## # ℹ 12 more variables: view <dbl>, condition <dbl>, grade <dbl>,
## #   sqft_above <dbl>, sqft_basement <dbl>, yr_built <dbl>, yr_renovated
<dbl>,
## #   zipcode <dbl>, lat <dbl>, long <dbl>, sqft_living15 <dbl>, sqft_lot15
<dbl>
```

```r
str(house_data)
```

```
## spc_tbl_ [21,597 × 21] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id           : num [1:21597] 7.13e+09 6.41e+09 5.63e+09 2.49e+09
1.95e+09 ...
##  $ date         : chr [1:21597] "10/13/2014" "12/9/2014" "2/25/2015"
"12/9/2014" ...
##  $ price        : num [1:21597] 221900 538000 180000 604000 510000 ...
##  $ bedrooms     : num [1:21597] 3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num [1:21597] 1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : num [1:21597] 1180 2570 770 1960 1680 ...
##  $ sqft_lot     : num [1:21597] 5650 7242 10000 5000 8080 ...
##  $ floors       : num [1:21597] 1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : num [1:21597] 0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : num [1:21597] 0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : num [1:21597] 3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : num [1:21597] 7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : num [1:21597] 1180 2170 770 1050 1680 ...
##  $ sqft_basement: num [1:21597] 0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : num [1:21597] 1955 1951 1933 1965 1987 ...
##  $ yr_renovated : num [1:21597] 0 1991 0 0 0 ...
##  $ zipcode      : num [1:21597] 98178 98125 98028 98136 98074 ...
##  $ lat          : num [1:21597] 47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num [1:21597] -122 -122 -122 -122 -122 ...
```

```
##  $ sqft_living15: num [1:21597] 1340 1690 2720 1360 1800 ...
##  $ sqft_lot15   : num [1:21597] 5650 7639 8062 5000 7503 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   date = col_character(),
##   ..   price = col_double(),
##   ..   bedrooms = col_double(),
##   ..   bathrooms = col_double(),
##   ..   sqft_living = col_double(),
##   ..   sqft_lot = col_double(),
##   ..   floors = col_double(),
##   ..   waterfront = col_double(),
##   ..   view = col_double(),
##   ..   condition = col_double(),
##   ..   grade = col_double(),
##   ..   sqft_above = col_double(),
##   ..   sqft_basement = col_double(),
##   ..   yr_built = col_double(),
##   ..   yr_renovated = col_double(),
##   ..   zipcode = col_double(),
##   ..   lat = col_double(),
##   ..   long = col_double(),
##   ..   sqft_living15 = col_double(),
##   ..   sqft_lot15 = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(house_data)
```

```
##        id                 date               price            bedrooms
##  Min.   :1.000e+06   Length:21597       Min.   :  78000   Min.   : 1.000
##  1st Qu.:2.123e+09   Class :character   1st Qu.: 322000   1st Qu.: 3.000
##  Median :3.905e+09   Mode  :character   Median : 450000   Median : 3.000
##  Mean   :4.580e+09                      Mean   : 540297   Mean   : 3.373
##  3rd Qu.:7.309e+09                      3rd Qu.: 645000   3rd Qu.: 4.000
##  Max.   :9.900e+09                      Max.   :7700000   Max.   :33.000
##    bathrooms      sqft_living       sqft_lot           floors
##  Min.   :0.500   Min.   :  370   Min.   :    520   Min.   :1.000
##  1st Qu.:1.750   1st Qu.: 1430   1st Qu.:   5040   1st Qu.:1.000
##  Median :2.250   Median : 1910   Median :   7618   Median :1.500
##  Mean   :2.116   Mean   : 2080   Mean   :  15099   Mean   :1.494
##  3rd Qu.:2.500   3rd Qu.: 2550   3rd Qu.:  10685   3rd Qu.:2.000
##  Max.   :8.000   Max.   :13540   Max.   :1651359   Max.   :3.500
##    waterfront            view          condition        grade
##  Min.   :0.000000   Min.   :0.0000   Min.   :1.00   Min.   : 3.000
##  1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:3.00   1st Qu.: 7.000
##  Median :0.000000   Median :0.0000   Median :3.00   Median : 7.000
##  Mean   :0.007547   Mean   :0.2343   Mean   :3.41   Mean   : 7.658
##  3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:4.00   3rd Qu.: 8.000
```

```
##  Max.   :1.000000   Max.   :4.0000   Max.   :5.00   Max.   :13.000
##    sqft_above   sqft_basement       yr_built     yr_renovated
##  Min.   : 370   Min.   :   0.0   Min.   :1900   Min.   :   0.00
##  1st Qu.:1190   1st Qu.:   0.0   1st Qu.:1951   1st Qu.:   0.00
##  Median :1560   Median :   0.0   Median :1975   Median :   0.00
##  Mean   :1789   Mean   : 291.7   Mean   :1971   Mean   :  84.46
##  3rd Qu.:2210   3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:   0.00
##  Max.   :9410   Max.   :4820.0   Max.   :2015   Max.   :2015.00
##     zipcode           lat             long        sqft_living15
##  Min.   :98001   Min.   :47.16   Min.   :-122.5   Min.   : 399
##  1st Qu.:98033   1st Qu.:47.47   1st Qu.:-122.3   1st Qu.:1490
##  Median :98065   Median :47.57   Median :-122.2   Median :1840
##  Mean   :98078   Mean   :47.56   Mean   :-122.2   Mean   :1987
##  3rd Qu.:98118   3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2360
##  Max.   :98199   Max.   :47.78   Max.   :-121.3   Max.   :6210
##    sqft_lot15
##  Min.   :   651
##  1st Qu.:  5100
##  Median :  7620
##  Mean   : 12758
##  3rd Qu.: 10083
##  Max.   :871200
```

The dataset includes 21 independent variables such as bedrooms, sqft_living, view, and grade, with the dependent variable being price. It comprises 21,597 observations.

```
# Check for missing values
na_values <- data.frame(no_of_na_values = colSums(is.na(house_data)))
head(na_values, 21)
```

```
##               no_of_na_values
## 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                  0
## sqft_basement               0
## yr_built                    0
## yr_renovated                0
## zipcode                     0
## lat                         0
## long                        0
```

```
## sqft_living15               0
## sqft_lot15                  0
```

There are no missing values in this data.

## EXPLORATORY DATA ANALYSIS ON THE TRAINING DATA

Now we modify the data slightly and add two new columns for better understanding. Price might depend on the age of the house and the number of times it has been renovated. Therefore, we extract the age and renovation count of each house from our training data.

```
# Modify the data to add age and renovation status
date_sale <- mdy(house_data$date)
house_data$sale_date_year <- as.integer(year(date_sale))
house_data$age <- house_data$sale_date_year - house_data$yr_built

house_data$reno <- ifelse(house_data$yr_renovated == 0, 0, 1)
house_data$reno <- as.factor(house_data$reno)
```

### Training and test data

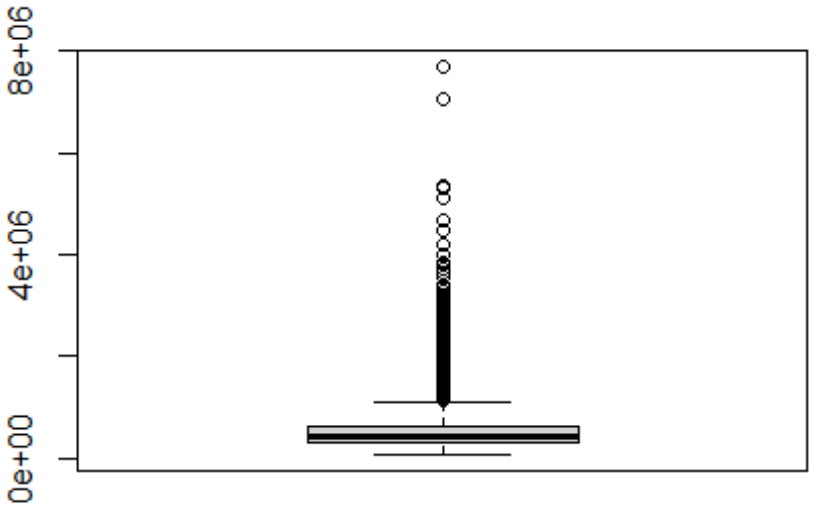
We divide the data to be the training set (80%) and test set (20%). If we have already the training and test set provided, then we do not need to add this step.

```
set.seed(500827260)

# Get the number of rows in the dataset
n <- nrow(house_data)

# Calculate the number of test samples (20% of the data)
ntest <- trunc(0.2 * n)

# Randomly sample indices for the test set
testid <- sample(1:n, ntest)

# Split the data into training and test sets
train_data <- house_data[-testid, ]
test_data <- house_data[testid, ]
```

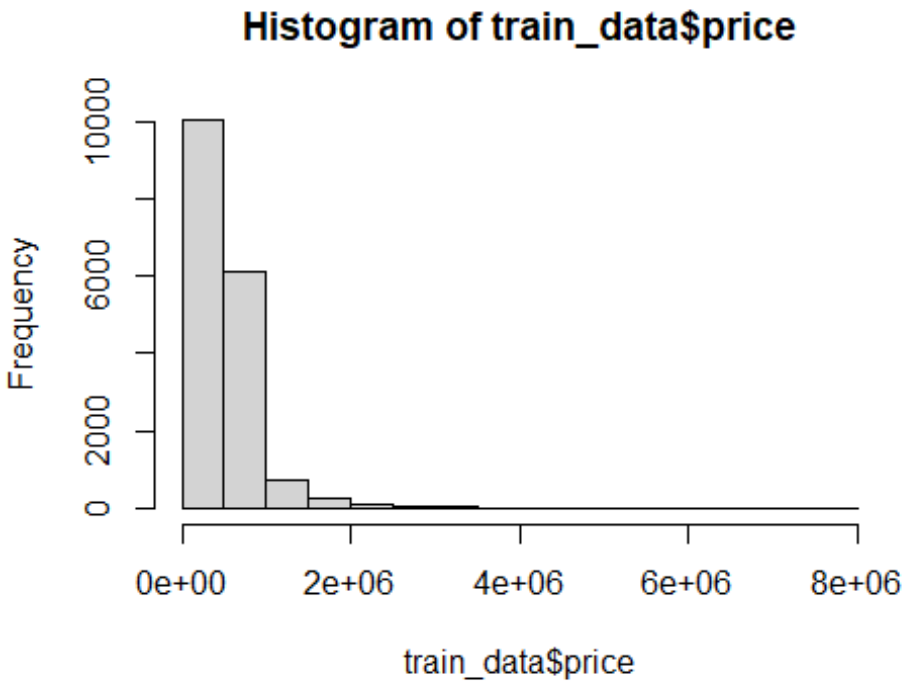
### Check the response variable

The price is skewed to the right with several very high prices.

```
boxplot(train_data$price)
```




```r
hist(train_data$price)
```



## Determining the association between variables.

We take out the correlation plot (corrplot) to understand the association of the dependent variable (price) with the independent variables.

```
# Extract relevant columns for correlation analysis
cor_data <- data.frame(train_data[, 3:21])
# Calculate/display the correlation matrix
correlation <- cor(cor_data)
correlation
```

```
##                    price     bedrooms   bathrooms sqft_living
sqft_lot
## price          1.00000000  0.306077990  0.51903533  0.70011874
0.088939498
## bedrooms       0.30607799  1.000000000  0.50936216  0.57463700
0.034448882
## bathrooms      0.51903533  0.509362162  1.00000000  0.75160682
0.086502204
## sqft_living    0.70011874  0.574636999  0.75160682  1.00000000
0.172952011
## sqft_lot       0.08893950  0.034448882  0.08650220  0.17295201
1.000000000
## floors         0.25161088  0.170354578  0.50276724  0.34891671
-0.005977938
## waterfront     0.28277826 -0.006113970  0.06695185  0.11011122
0.024810840
## view           0.40306286  0.078695888  0.18356416  0.28776393
0.076969959
## condition      0.03161977  0.027814949 -0.13246512 -0.06040568
-0.006574401
## grade          0.66648266  0.356335695  0.66338991  0.76239481
0.113819516
## sqft_above     0.60026762  0.476004602  0.68416377  0.87546345
0.182922145
## sqft_basement  0.33186412  0.303366049  0.28242436  0.44041722
0.017637077
## yr_built       0.05627200  0.154861177  0.51159332  0.31718189
0.049922766
## yr_renovated   0.13453372  0.017731384  0.04957508  0.05823968
0.010115012
## zipcode       -0.06011189 -0.148847183 -0.20530046 -0.20312836
-0.130853386
## lat            0.30489411 -0.007234058  0.02280074  0.05418708
-0.087744983
## long           0.02205016  0.129090398  0.22220669  0.23936565
0.230778328
## sqft_living15  0.58491740  0.389300891  0.56480778  0.75741531
0.149343375
## sqft_lot15     0.08138600  0.030529652  0.08563335  0.18387244
```

```
0.706972433
##                     floors  waterfront         view    condition
grade
## price          0.251610879  0.28277826  0.403062857  0.031619766
0.66648266
## bedrooms       0.170354578 -0.00611397  0.078695888  0.027814949
0.35633570
## bathrooms      0.502767245  0.06695185  0.183564156 -0.132465124
0.66338991
## sqft_living    0.348916706  0.11011122  0.287763934 -0.060405676
0.76239481
## sqft_lot      -0.005977938  0.02481084  0.076969959 -0.006574401
0.11381952
## floors         1.000000000  0.02527030  0.025936262 -0.267699127
0.45735869
## waterfront     0.025270296  1.00000000  0.413469577  0.019458215
0.09089819
## view           0.025936262  0.41346958  1.000000000  0.041003987
0.25534838
## condition     -0.267699127  0.01945822  0.041003987  1.000000000
-0.15038244
## grade          0.457358687  0.09089819  0.255348379 -0.150382442
1.00000000
## sqft_above     0.520478123  0.07822323  0.169153474 -0.158890104
0.75553266
## sqft_basement -0.245761327  0.08225864  0.280503514  0.170324875
0.17213915
## yr_built       0.494982268 -0.02398992 -0.052646270 -0.359951001
0.44996371
## yr_renovated   0.001275129  0.09919937  0.111684723 -0.066416235
0.01626279
## zipcode       -0.056690845  0.02641660  0.071849232  0.000790564
-0.18889936
## lat            0.049081704 -0.01090550  0.006353295 -0.013466213
0.11326987
## long           0.123129357 -0.04010604 -0.074786158 -0.106765135
0.19771543
## sqft_living15  0.274070466  0.09367597  0.290851588 -0.097286967
0.71338212
## sqft_lot15    -0.012680664  0.02991400  0.072275383 -0.002548374
0.12093126
##                 sqft_above sqft_basement    yr_built  yr_renovated
zipcode
## price          0.600267621    0.33186412  0.05627200  0.1345337199
-0.060111885
## bedrooms       0.476004602    0.30336605  0.15486118  0.0177313844
-0.148847183
## bathrooms      0.684163772    0.28242436  0.51159332  0.0495750815
-0.205300456
## sqft_living    0.875463453    0.44041722  0.31718189  0.0582396818
```

```
-0.203128361
## sqft_lot       0.182922145    0.01763708  0.04992277  0.0101150119
-0.130853386
## floors         0.520478123   -0.24576133  0.49498227  0.0012751290
-0.056690845
## waterfront     0.078223225    0.08225864 -0.02398992  0.0991993709
0.026416604
## view           0.169153474    0.28050351 -0.05264627  0.1116847228
0.071849232
## condition     -0.158890104    0.17032488 -0.35995100 -0.0664162346
0.000790564
## grade          0.755532657    0.17213915  0.44996371  0.0162627895
-0.188899359
## sqft_above     1.000000000   -0.04832022  0.42498824  0.0250415006
-0.266294813
## sqft_basement -0.048320219    1.00000000 -0.13395878  0.0738479609
0.074876404
## yr_built       0.424988241   -0.13395878  1.00000000 -0.2237978474
-0.343239906
## yr_renovated   0.025041501    0.07384796 -0.22379785  1.0000000000
0.057968532
## zipcode       -0.266294813    0.07487640 -0.34323991  0.0579685323
1.000000000
## lat           -0.002468024    0.11657643 -0.14493479  0.0253187033
0.265373818
## long           0.345720740   -0.14753157  0.40557131 -0.0640832477
-0.565867503
## sqft_living15  0.733405253    0.20295362  0.32458673  0.0006588159
-0.283406163
## sqft_lot15     0.193891611    0.01982908  0.07015339  0.0122694464
-0.149605112
##                        lat        long sqft_living15   sqft_lot15
## price          0.304894109  0.02205016  0.5849174024  0.081386003
## bedrooms      -0.007234058  0.12909040  0.3893008912  0.030529652
## bathrooms      0.022800741  0.22220669  0.5648077846  0.085633351
## sqft_living    0.054187081  0.23936565  0.7574153100  0.183872438
## sqft_lot      -0.087744983  0.23077833  0.1493433753  0.706972433
## floors         0.049081704  0.12312936  0.2740704663 -0.012680664
## waterfront    -0.010905497 -0.04010604  0.0936759690  0.029914000
## view           0.006353295 -0.07478616  0.2908515878  0.072275383
## condition     -0.013466213 -0.10676513 -0.0972869666 -0.002548374
## grade          0.113269875  0.19771543  0.7133821211  0.120931255
## sqft_above    -0.002468024  0.34572074  0.7334052534  0.193891611
## sqft_basement  0.116576427 -0.14753157  0.2029536163  0.019829076
## yr_built      -0.144934794  0.40557131  0.3245867290  0.070153389
## yr_renovated   0.025318703 -0.06408325  0.0006588159  0.012269446
## zipcode        0.265373818 -0.56586750 -0.2834061626 -0.149605112
## lat            1.000000000 -0.13429243  0.0470454133 -0.086315437
## long          -0.134292426  1.00000000  0.3349122556  0.258025384
```

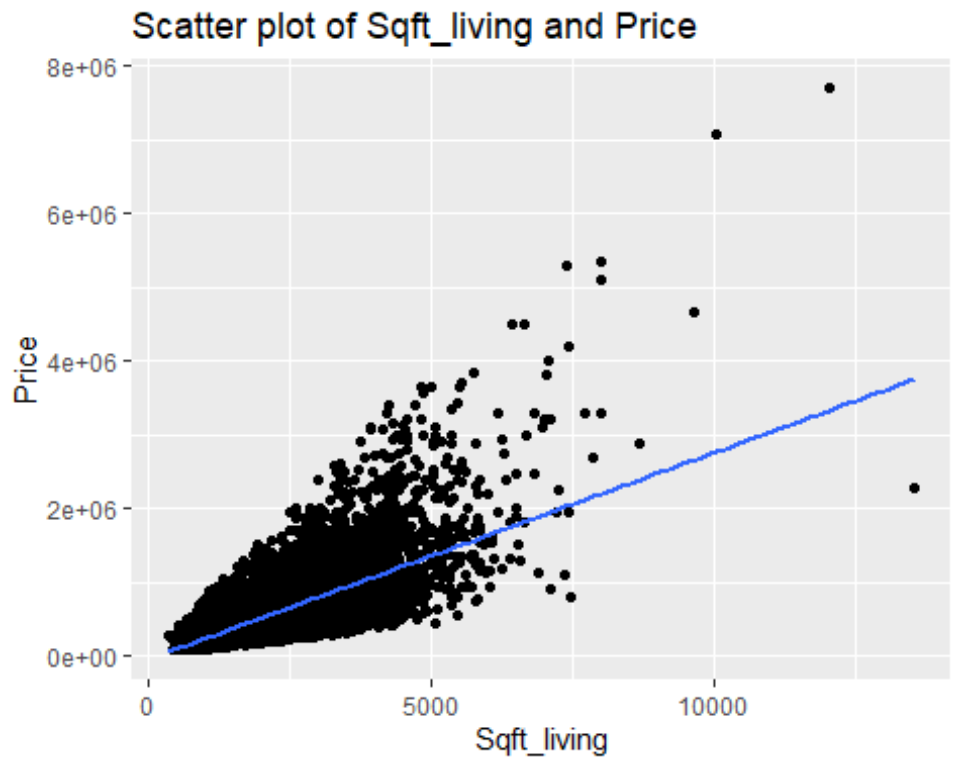
```
## sqft_living15  0.047045413  0.33491226  1.0000000000  0.189558735
## sqft_lot15    -0.086315437  0.25802538  0.1895587354  1.000000000
```

```
# Setup and plot the correlation matrix
par(mfrow = c(1, 1))
corrplot(correlation, method = "color")
```


Price is strongly positively correlated with bathroom, Sqft_living, grade, sqft_above, sqft_living15. We use scatterplot and boxplot to visualize the relationship between price and some predictors.

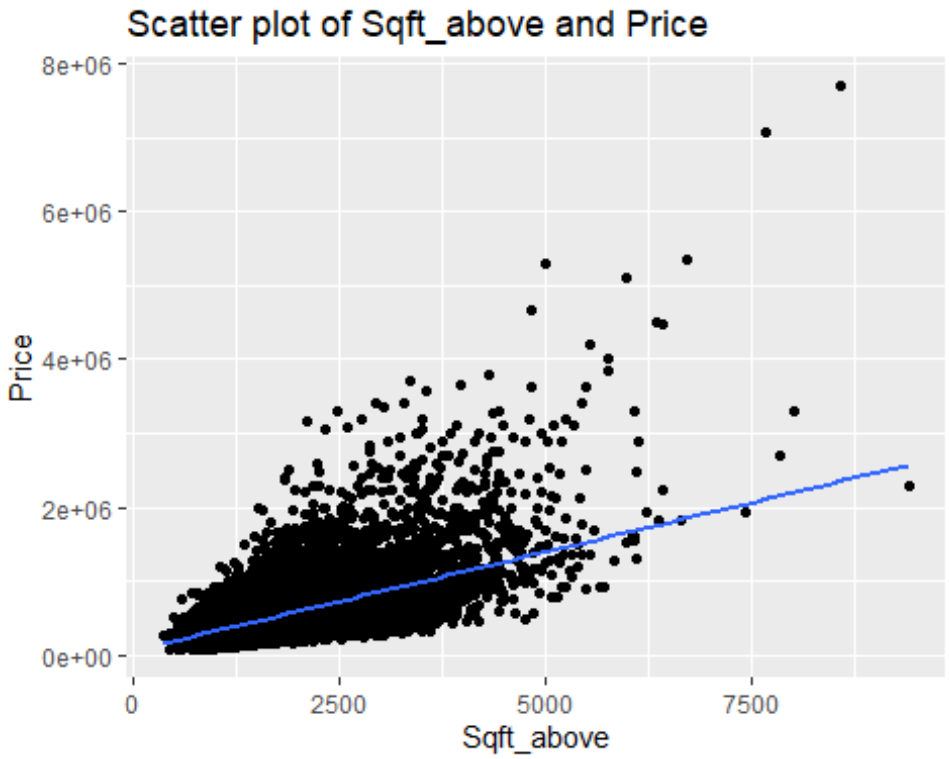
```
# Scatter plot of Sqft_living and Price
ggplot(data = train_data, aes(x = sqft_living, y = price)) +
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Scatter plot of Sqft_living and Price",
       x = "Sqft_living",
       y = "Price")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



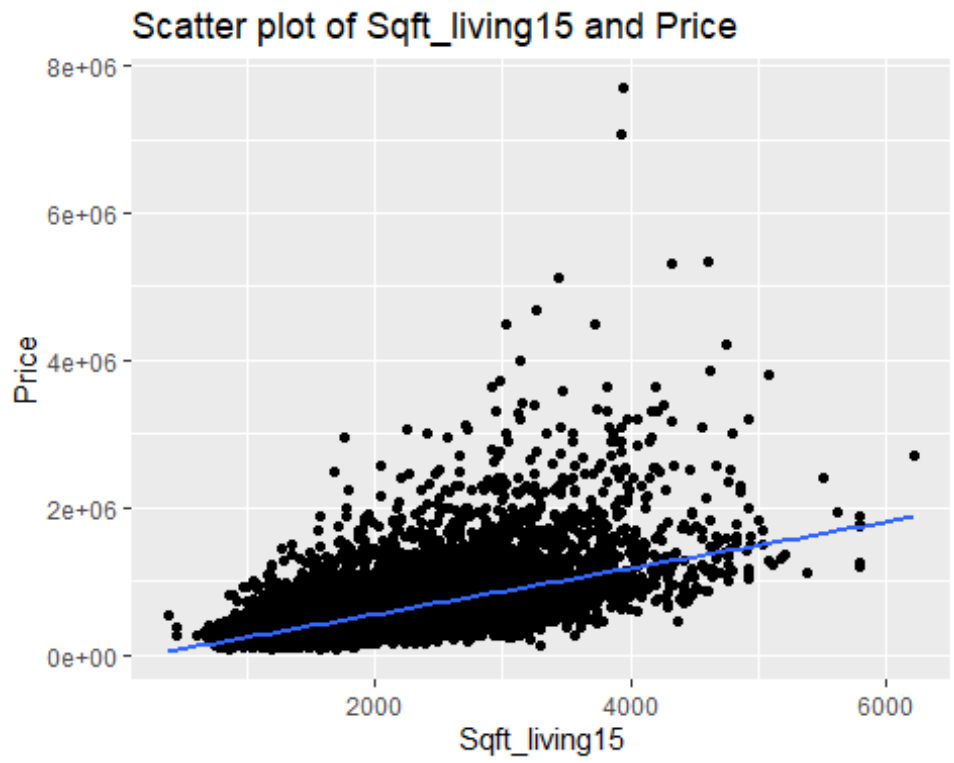

```
# Scatter plot of Sqft_above and Price
ggplot(data = train_data, aes(x = sqft_above, y = price)) +
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Scatter plot of Sqft_above and Price",
       x = "Sqft_above",
       y = "Price")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

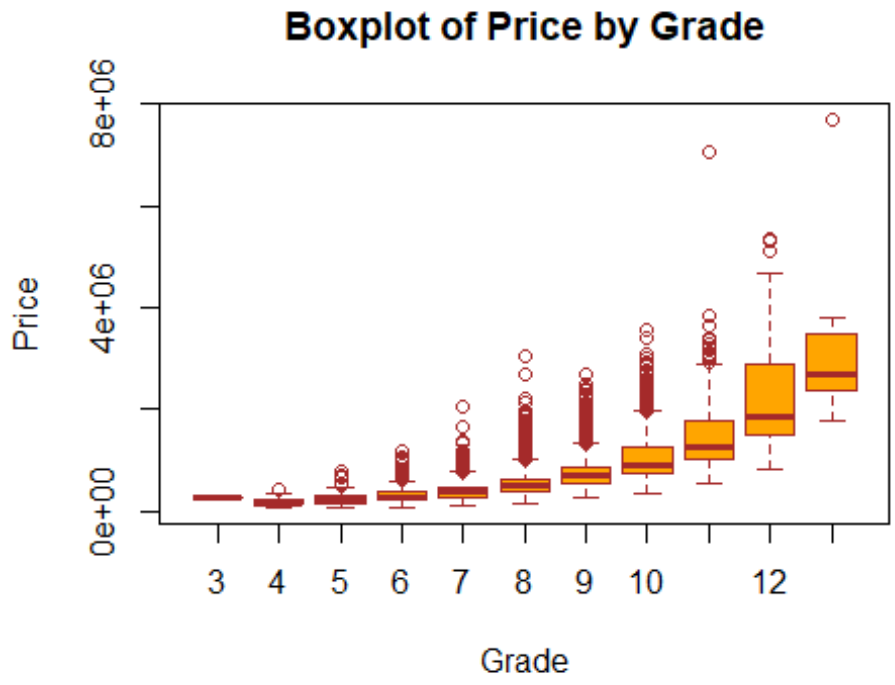



```
# Scatter plot of Sqft_living15 and Price
ggplot(data = train_data, aes(x = sqft_living15, y = price)) +
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Scatter plot of Sqft_living15 and Price",
       x = "Sqft_living15",
       y = "Price")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

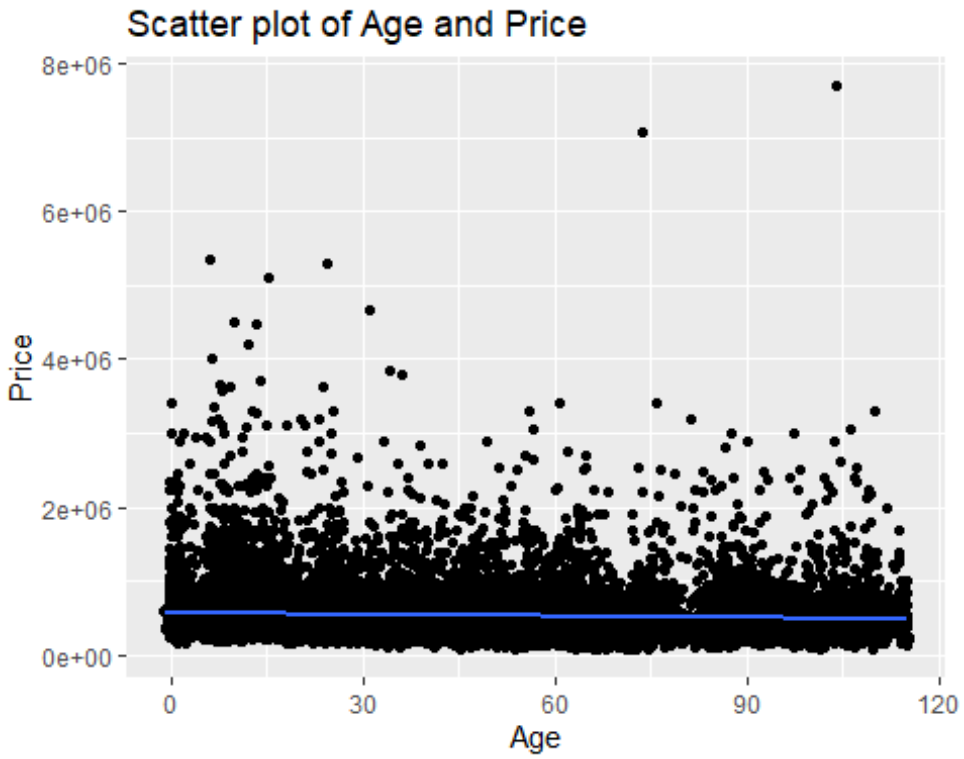



```r
# Boxplot of Price by Grade
boxplot(price ~ grade,
        data = train_data,
        main = "Boxplot of Price by Grade",
        xlab = "Grade",
        ylab = "Price",
        col = "orange",
        border = "brown")
```

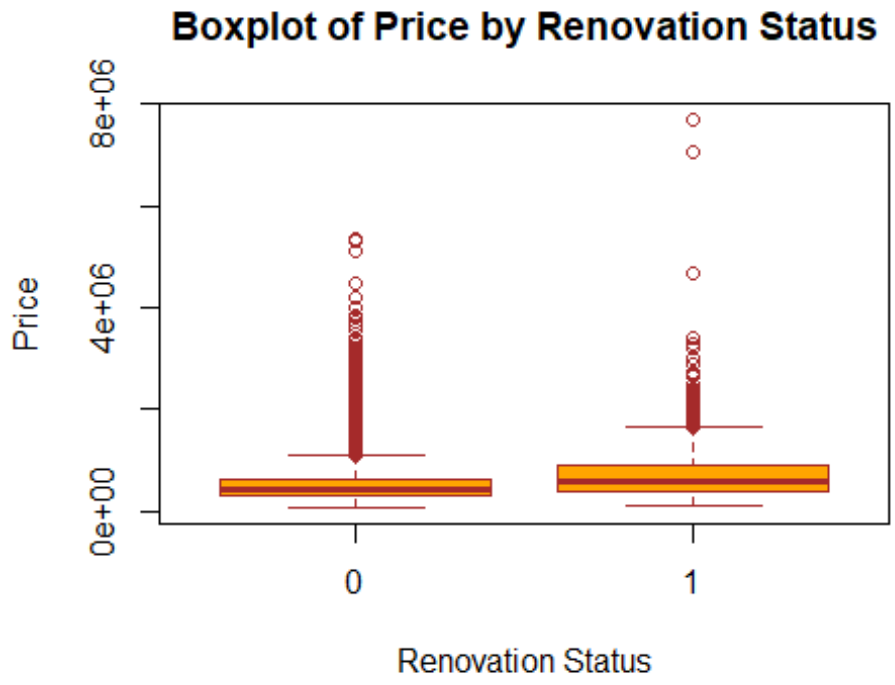



Check the new added variables:

```
# Scatter plot of Age and Price
ggplot(data = train_data, aes(x = age, y = price)) +
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Scatter plot of Age and Price",
       x = "Age",
       y = "Price")

## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Boxplot of Price by Renovation Status
boxplot(price ~ reno,
        data = train_data,
        main = "Boxplot of Price by Renovation Status",
        xlab = "Renovation Status",
        ylab = "Price",
        col = "orange",
        border = "brown")
```




## Removing outlier could be optional

We see that we have a significantly large number of outliers.

Treating or altering the outlier/extreme values in genuine observations is not a standard operating procedure. However, it is essential to understand their impact on our predictive models.

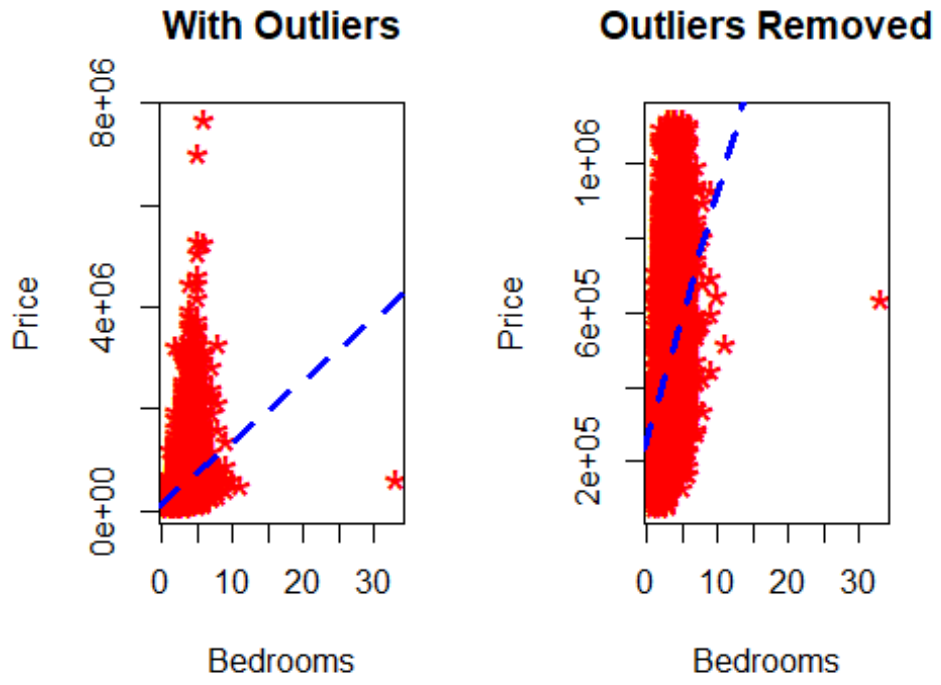
To better understand the implications of outliers better, we should compare the fit of a simple linear regression model on the dataset with and without outliers. For this we first extract outliers from the data and then obtain the data without the outliers.

```
# Identify and remove outliers
outliers <- boxplot(train_data$price, plot = FALSE)$out
outliers_data <- train_data[which(train_data$price %in% outliers), ]
train_data1 <- train_data[-which(train_data$price %in% outliers), ]
```

Now plot Now we plot the data with and without outliers.

```
par(mfrow = c(1, 2))

# Plot of original data with outliers
plot(train_data$bedrooms, train_data$price,
     main = "With Outliers",
     xlab = "Bedrooms",
     ylab = "Price",
     pch = "*",
```

```
     col = "red",
     cex = 2)
abline(lm(price ~ bedrooms, data = train_data),
       col = "blue",
       lwd = 3,
       lty = 2)

# Plot of data without outliers
plot(train_data1$bedrooms, train_data1$price,
     main = "Outliers Removed",
     xlab = "Bedrooms",
     ylab = "Price",
     pch = "*",
     col = "red",
     cex = 2)
abline(lm(price ~ bedrooms, data = train_data1),
       col = "blue",
       lwd = 3,
       lty = 2)
```




## MODELING

We first use the entire data.

```
# Prepare the training data by removing unnecessary columns
# and converting some columns to factors
```

```r
train_data.m <- train_data[, -c(1, 2, 15, 16, 17, 22)] %>%
  mutate(waterfront = as.factor(waterfront),
         view = as.factor(view),
         condition = as.factor(condition),
         reno = as.factor(reno))

# Check the structure of the data
str(train_data.m)
```

```
## tibble [17,278 × 18] (S3: tbl_df/tbl/data.frame)
##  $ price        : num [1:17278] 221900 538000 180000 510000 1230000 ...
##  $ bedrooms     : num [1:17278] 3 3 2 3 4 3 3 3 3 2 ...
##  $ bathrooms    : num [1:17278] 1 2.25 1 2 4.5 2.25 1 2.5 2.5 1 ...
##  $ sqft_living  : num [1:17278] 1180 2570 770 1680 5420 ...
##  $ sqft_lot     : num [1:17278] 5650 7242 10000 8080 101930 ...
##  $ floors       : num [1:17278] 1 2 1 1 1 2 1 2 1 1 ...
##  $ waterfront   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ view         : Factor w/ 5 levels "0","1","2","3",..: 1 1 1 1 1 1 1 1 1
1 ...
##  $ condition    : Factor w/ 5 levels "1","2","3","4",..: 3 3 3 3 3 3 3 3 3
4 ...
##  $ grade        : num [1:17278] 7 7 6 8 11 7 7 7 8 7 ...
##  $ sqft_above   : num [1:17278] 1180 2170 770 1680 3890 ...
##  $ sqft_basement: num [1:17278] 0 400 0 0 1530 0 730 0 1700 300 ...
##  $ lat          : num [1:17278] 47.5 47.7 47.7 47.6 47.7 ...
##  $ long         : num [1:17278] -122 -122 -122 -122 -122 ...
##  $ sqft_living15: num [1:17278] 1340 1690 2720 1800 4760 ...
##  $ sqft_lot15   : num [1:17278] 5650 7639 8062 7503 101930 ...
##  $ age          : num [1:17278] 59 63 82 28 13 19 55 12 50 72 ...
##  $ reno         : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
```

```r
# Check the number of columns
ncol(train_data.m)
```

```
## [1] 18
```

```r
# Fit the model
model.full <- lm (formula = price ~ ., data = train_data.m)
# Display the model summary
summary(model.full)
```

```
##
## Call:
## lm(formula = price ~ ., data = train_data.m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1205541   -99237    -9262    76784  4393501
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   -4.013e+07  1.711e+06 -23.447  < 2e-16 ***
## bedrooms      -3.262e+04  2.114e+03 -15.428  < 2e-16 ***
## bathrooms      3.989e+04  3.661e+03  10.896  < 2e-16 ***
## sqft_living    1.474e+02  4.916e+00  29.980  < 2e-16 ***
## sqft_lot       1.192e-01  5.223e-02   2.283   0.0224 *
## floors         2.982e+03  4.033e+03   0.739   0.4596
## waterfront1    5.268e+05  2.182e+04  24.143  < 2e-16 ***
## view1          1.158e+05  1.247e+04   9.286  < 2e-16 ***
## view2          6.193e+04  7.669e+03   8.075 7.22e-16 ***
## view3          1.261e+05  1.048e+04  12.035  < 2e-16 ***
## view4          2.918e+05  1.656e+04  17.619  < 2e-16 ***
## condition2    -1.272e+04  4.554e+04  -0.279   0.7799
## condition3    -2.483e+04  4.238e+04  -0.586   0.5580
## condition4     6.739e+03  4.238e+04   0.159   0.8737
## condition5     4.148e+04  4.263e+04   0.973   0.3306
## grade          9.994e+04  2.427e+03  41.177  < 2e-16 ***
## sqft_above     2.643e+01  4.911e+00   5.381 7.50e-08 ***
## sqft_basement         NA         NA      NA       NA
## lat            5.518e+05  1.177e+04  46.888  < 2e-16 ***
## long          -1.077e+05  1.337e+04  -8.059 8.20e-16 ***
## sqft_living15  2.660e+01  3.898e+00   6.824 9.13e-12 ***
## sqft_lot15    -3.861e-01  8.005e-02  -4.824 1.42e-06 ***
## age            2.409e+03  8.185e+01  29.432  < 2e-16 ***
## reno1          5.513e+04  8.226e+03   6.702 2.12e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202100 on 17255 degrees of freedom
## Multiple R-squared:  0.6967, Adjusted R-squared:  0.6963
## F-statistic:  1802 on 22 and 17255 DF,  p-value: < 2.2e-16

# Get the model coefficients
models <- regsubsets(price ~ ., data = train_data.m, nvmax = 23)

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in =
## force.in, : 1 linear dependencies found

## Reordering variables and trying again:

summary(models)

## Subset selection object
## Call: regsubsets.formula(price ~ ., data = train_data.m, nvmax = 23)
## 23 Variables  (and intercept)
##               Forced in Forced out
## bedrooms          FALSE      FALSE
## bathrooms         FALSE      FALSE
## sqft_living       FALSE      FALSE
## sqft_lot          FALSE      FALSE
## floors            FALSE      FALSE
```

```
## waterfront1       FALSE      FALSE
## view1             FALSE      FALSE
## view2             FALSE      FALSE
## view3             FALSE      FALSE
## view4             FALSE      FALSE
## condition2        FALSE      FALSE
## condition3        FALSE      FALSE
## condition4        FALSE      FALSE
## condition5        FALSE      FALSE
## grade             FALSE      FALSE
## sqft_above        FALSE      FALSE
## lat               FALSE      FALSE
## long              FALSE      FALSE
## sqft_living15     FALSE      FALSE
## sqft_lot15        FALSE      FALSE
## age               FALSE      FALSE
## reno1             FALSE      FALSE
## sqft_basement     FALSE      FALSE
## 1 subsets of each size up to 22
## Selection Algorithm: exhaustive
##           bedrooms bathrooms sqft_living sqft_lot floors waterfront1 view1
## 1  ( 1 )  " "      " "       "*"         " "      " "    " "         " "
## 2  ( 1 )  " "      " "       "*"         " "      " "    " "         " "
## 3  ( 1 )  " "      " "       "*"         " "      " "    "*"         " "
## 4  ( 1 )  " "      " "       "*"         " "      " "    " "         " "
## 5  ( 1 )  " "      " "       "*"         " "      " "    "*"         " "
## 6  ( 1 )  " "      " "       "*"         " "      " "    "*"         " "
## 7  ( 1 )  "*"      " "       "*"         " "      " "    "*"         " "
## 8  ( 1 )  "*"      "*"       "*"         " "      " "    "*"         " "
## 9  ( 1 )  "*"      "*"       "*"         " "      " "    "*"         " "
## 10  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"
## 11  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"
## 12  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"
## 13  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"
## 14  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"
## 15  ( 1 ) "*"      "*"       "*"         " "      " "    "*"         "*"
## 16  ( 1 ) "*"      "*"       " "         " "      " "    "*"         "*"
## 17  ( 1 ) "*"      "*"       " "         " "      " "    "*"         "*"
## 18  ( 1 ) "*"      "*"       " "         " "      " "    "*"         "*"
## 19  ( 1 ) "*"      "*"       "*"         "*"      " "    "*"         "*"
## 20  ( 1 ) "*"      "*"       "*"         "*"      " "    "*"         "*"
## 21  ( 1 ) "*"      "*"       "*"         "*"      "*"    "*"         "*"
## 22  ( 1 ) "*"      "*"       "*"         "*"      "*"    "*"         "*"
##           view2 view3 view4 condition2 condition3 condition4 condition5
grade
## 1  ( 1 )  " "   " "   " "   " "        " "        " "        " "        "
"
## 2  ( 1 )  " "   " "   " "   " "        " "        " "        " "        "
"
## 3  ( 1 )  " "   " "   " "   " "        " "        " "        " "        "
```

```
"
## 4  ( 1 )  " "   " "   " "   " "        " "        " "        " "
"*"
## 5  ( 1 )  " "   " "   " "   " "        " "        " "        " "
"*"
## 6  ( 1 )  " "   " "   "*"   " "        " "        " "        " "
"*"
## 7  ( 1 )  " "   " "   "*"   " "        " "        " "        " "
"*"
## 8  ( 1 )  " "   " "   "*"   " "        " "        " "        " "
"*"
## 9  ( 1 )  " "   "*"   "*"   " "        " "        " "        " "
"*"
## 10  ( 1 ) " "   "*"   "*"   " "        " "        " "        " "
"*"
## 11  ( 1 ) "*"   "*"   "*"   " "        " "        " "        " "
"*"
## 12  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 13  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 14  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 15  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 16  ( 1 ) "*"   "*"   "*"   " "        "*"        " "        " "
"*"
## 17  ( 1 ) "*"   "*"   "*"   " "        " "        "*"        "*"
"*"
## 18  ( 1 ) "*"   "*"   "*"   " "        " "        "*"        "*"
"*"
## 19  ( 1 ) "*"   "*"   "*"   " "        " "        "*"        "*"
"*"
## 20  ( 1 ) "*"   "*"   "*"   "*"        "*"        " "        "*"
"*"
## 21  ( 1 ) "*"   "*"   "*"   "*"        "*"        " "        "*"
"*"
## 22  ( 1 ) "*"   "*"   "*"   "*"        "*"        "*"        "*"
"*"
##           sqft_above sqft_basement lat long sqft_living15 sqft_lot15 age
reno1
## 1  ( 1 )  " "        " "           " " " "  " "           " "        " " "
"
## 2  ( 1 )  " "        " "           "*" " "  " "           " "        " " "
"
## 3  ( 1 )  " "        " "           "*" " "  " "           " "        " " "
"
## 4  ( 1 )  " "        " "           "*" " "  " "           " "        "*" "
"
## 5  ( 1 )  " "        " "           "*" " "  " "           " "        "*" "
```

```
"
## 6  ( 1 )  " "         " "           "*" " "  " "           " "         "*" "
"
## 7  ( 1 )  " "         " "           "*" " "  " "           " "         "*" "
"
## 8  ( 1 )  " "         " "           "*" " "  " "           " "         "*" "
"
## 9  ( 1 )  " "         " "           "*" " "  " "           " "         "*" "
"
## 10  ( 1 ) " "         " "           "*" " "  " "           " "         "*" "
"
## 11  ( 1 ) " "         " "           "*" " "  " "           " "         "*" "
"
## 12  ( 1 ) " "         " "           "*" " "  " "           " "         "*" "
"
## 13  ( 1 ) " "         " "           "*" "*"  " "           " "         "*" "
"
## 14  ( 1 ) "*"         " "           "*" "*"  " "           " "         "*" "
"
## 15  ( 1 ) " "         " "           "*" "*"  "*"           " "         "*"
"*"
## 16  ( 1 ) "*"         "*"           "*" "*"  "*"           " "         "*"
"*"
## 17  ( 1 ) "*"         "*"           "*" "*"  "*"           " "         "*"
"*"
## 18  ( 1 ) "*"         "*"           "*" "*"  "*"           "*"         "*"
"*"
## 19  ( 1 ) " "         "*"           "*" "*"  "*"           "*"         "*"
"*"
## 20  ( 1 ) "*"         " "           "*" "*"  "*"           "*"         "*"
"*"
## 21  ( 1 ) "*"         " "           "*" "*"  "*"           "*"         "*"
"*"
## 22  ( 1 ) "*"         " "           "*" "*"  "*"           "*"         "*"
"*"

# Get the best model based on Adjusted R-squared, Cp, and BIC
res.sum <- summary(models)
data.frame(
  Adj.R2 = which.max(res.sum$adjr2),
  CP = which.min(res.sum$cp),
  BIC = which.min(res.sum$bic)
)

##   Adj.R2 CP BIC
## 1     19 19  18

# id: model id
# object: regsubsets object
# data: data used to fit regsubsets
```

```r
# outcome: outcome variable
get_model_formula <- function(id, object, outcome) {
  # get models data
  models <- summary(object)$which[id, -1]
  # Get model predictors
  predictors <- names(which(models == TRUE))
  predictors <- paste(predictors, collapse = "+")
  # Build model formula
  as.formula(paste0(outcome, "~", predictors))
}

get_model_formula(21, models, "Price")
```

```
## Price ~ bedrooms + bathrooms + sqft_living + sqft_lot + floors +
##     waterfront1 + view1 + view2 + view3 + view4 + condition2 +
##     condition3 + condition5 + grade + sqft_above + lat + long +
##     sqft_living15 + sqft_lot15 + age + reno1
## <environment: 0x000001915b045e48>
```

```r
# # Fit the linear regression model 1 with selected predictors
model1 <- lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + floors +
            waterfront + view + condition + grade + sqft_basement + lat +
            long + sqft_living15 + sqft_lot15 + age + reno,
            data = train_data.m)
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     lat + long + sqft_living15 + sqft_lot15 + age + reno, data =
train_data.m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1205541   -99237    -9262    76784  4393501
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.013e+07  1.711e+06 -23.447  < 2e-16 ***
## bedrooms      -3.262e+04  2.114e+03 -15.428  < 2e-16 ***
## bathrooms      3.989e+04  3.661e+03  10.896  < 2e-16 ***
## sqft_living    1.738e+02  4.137e+00  42.019  < 2e-16 ***
## sqft_lot       1.192e-01  5.223e-02   2.283   0.0224 *
## floors         2.982e+03  4.033e+03   0.739   0.4596
## waterfront1    5.268e+05  2.182e+04  24.143  < 2e-16 ***
## view1          1.158e+05  1.247e+04   9.286  < 2e-16 ***
## view2          6.193e+04  7.669e+03   8.075 7.22e-16 ***
## view3          1.261e+05  1.048e+04  12.035  < 2e-16 ***
## view4          2.918e+05  1.656e+04  17.619  < 2e-16 ***
```

```
## condition2    -1.272e+04  4.554e+04  -0.279   0.7799
## condition3    -2.483e+04  4.238e+04  -0.586   0.5580
## condition4     6.739e+03  4.238e+04   0.159   0.8737
## condition5     4.148e+04  4.263e+04   0.973   0.3306
## grade          9.994e+04  2.427e+03  41.177  < 2e-16 ***
## sqft_basement -2.643e+01  4.911e+00  -5.381 7.50e-08 ***
## lat            5.518e+05  1.177e+04  46.888  < 2e-16 ***
## long          -1.077e+05  1.337e+04  -8.059 8.20e-16 ***
## sqft_living15  2.660e+01  3.898e+00   6.824 9.13e-12 ***
## sqft_lot15    -3.861e-01  8.005e-02  -4.824 1.42e-06 ***
## age            2.409e+03  8.185e+01  29.432  < 2e-16 ***
## reno1          5.513e+04  8.226e+03   6.702 2.12e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202100 on 17255 degrees of freedom
## Multiple R-squared:  0.6967, Adjusted R-squared:  0.6963
## F-statistic:  1802 on 22 and 17255 DF,  p-value: < 2.2e-16
```

```
# Fit the linear regression model 2 with selected predictors
model2 <- lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + floors +
             waterfront + view + condition + grade + sqft_basement + lat +
             long + sqft_living15 + sqft_lot15 + age + reno +
             bathrooms * grade + grade * sqft_living15 + grade * sqft_lot15 +
             lat * long,
             data = train_data.m)
summary(model2)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_basement +
##     lat + long + sqft_living15 + sqft_lot15 + age + reno + bathrooms *
##     grade + grade * sqft_living15 + grade * sqft_lot15 + lat *
##     long, data = train_data.m)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1891128   -91109    -7415    73422  3912013
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.985e+09  4.713e+08  12.700  < 2e-16 ***
## bedrooms            -1.393e+04  2.041e+03  -6.825 9.08e-12 ***
## bathrooms           -4.107e+05  1.346e+04 -30.505  < 2e-16 ***
## sqft_living          1.236e+02  4.106e+00  30.092  < 2e-16 ***
## sqft_lot             7.988e-02  4.940e-02   1.617 0.105924
## floors               1.466e+04  3.858e+03   3.800 0.000145 ***
## waterfront1          5.409e+05  2.061e+04  26.242  < 2e-16 ***
## view1                1.115e+05  1.177e+04   9.475  < 2e-16 ***
```

```
## view2                 5.465e+04  7.238e+03   7.549 4.59e-14 ***
## view3                 1.105e+05  9.890e+03  11.175  < 2e-16 ***
## view4                 2.417e+05  1.565e+04  15.440  < 2e-16 ***
## condition2            1.975e+04  4.295e+04   0.460 0.645615
## condition3            4.129e+04  4.000e+04   1.032 0.301965
## condition4            8.319e+04  4.001e+04   2.080 0.037582 *
## condition5            1.205e+05  4.025e+04   2.994 0.002762 **
## grade                -3.041e+04  4.151e+03  -7.326 2.48e-13 ***
## sqft_basement         5.339e-01  4.708e+00   0.113 0.909713
## lat                  -1.261e+08  9.913e+06 -12.725  < 2e-16 ***
## long                  4.919e+07  3.856e+06  12.757  < 2e-16 ***
## sqft_living15         1.727e+01  1.600e+01   1.080 0.280371
## sqft_lot15            2.157e+00  2.938e-01   7.342 2.20e-13 ***
## age                   1.743e+03  7.886e+01  22.101  < 2e-16 ***
## reno1                 7.797e+04  7.780e+03  10.023  < 2e-16 ***
## bathrooms:grade       5.804e+04  1.655e+03  35.063  < 2e-16 ***
## grade:sqft_living15   2.633e+00  1.870e+00   1.408 0.159251
## grade:sqft_lot15     -3.024e-01  3.458e-02  -8.747  < 2e-16 ***
## lat:long             -1.037e+06  8.112e+04 -12.782  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 190500 on 17251 degrees of freedom
## Multiple R-squared:  0.7307, Adjusted R-squared:  0.7303
## F-statistic:  1800 on 26 and 17251 DF,  p-value: < 2.2e-16
```

## PREDICTION ON THE TEST DATA

```
# Preparing the test data
test_data.m <- test_data[, -c(1, 2, 15, 16, 17, 22)] %>%
  mutate(waterfront = as.factor(waterfront),
         view = as.factor(view),
         condition = as.factor(condition),
         reno = as.factor(reno))

# Predict the house prices on the test data using model2
pred_test <- predict(newdata = test_data.m, model2)

# Create a data frame to compare actual and predicted prices
tally_table_1 <- data.frame(actual = test_data.m$price, predicted =
pred_test)

mean(abs(test_data.m$price - pred_test))

## [1] 118663.6
```

## Compare with one-layer forward neural network

```
# Create the model matrix for the predictors and scale the data
x <- model.matrix(price ~ . - 1, data = train_data.m) %>% scale()
y <- train_data.m$price
```

The neural network has a similar test error to the multiple linear regression model. However, with additional time, we could tune the parameters to improve the performance of the neural network.