# House Prices

*Nick Hopewell*

*November 11, 2017*

This dataset includes the sale prices of houses in King County, Washington between May 2014 and May 2015. Seattle is the largest city within King County and is the seat of the county. Kings County is the thirteethn most populaous county in America.

**All the code will be displayed for grading purposes with expection to the scatter plots in the final section because the code for these is very lengthy.**

```
#  Load required packages
library(ggplot2)
library(lubridate)
library(gridExtra)
library(corrplot)
library(psych)
library(Hmisc)
library(dplyr)
library(MASS)
library(car)
library(pastecs)
library(gridExtra)
```

# Data Inspection and Preprocessing

**Read in the data file**

```
#  Set working directory
setwd("C:\\Users\\nicho\\Desktop\\Assignments\\Statistical Modelling assignments\\Data analysis
 asignment")

#  Read csv file and save in object 'priceData'
priceData<-read.csv("house_prices.csv", header = TRUE)
```

**Print out first few rows**

```
#View(priceData)

head(priceData)
```

```
##           id            date   price bedrooms bathrooms sqft_living sqft_lot
## 1 7129300520 20141013T000000  221900        3      1.00        1180     5650
## 2 6414100192 20141209T000000  538000        3      2.25        2570     7242
## 3 5631500400 20150225T000000  180000        2      1.00         770    10000
## 4 2487200875 20141209T000000  604000        4      3.00        1960     5000
## 5 1954400510 20150218T000000  510000        3      2.00        1680     8080
## 6 7237550310 20140512T000000 1225000        4      4.50        5420   101930
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      1          0    0         3     7       1180             0     1955
## 2      2          0    0         3     7       2170           400     1951
## 3      1          0    0         3     6        770             0     1933
## 4      1          0    0         5     7       1050           910     1965
## 5      1          0    0         3     8       1680             0     1987
## 6      1          0    0         3    11       3890          1530     2001
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1            0   98178 47.5112 -122.257          1340       5650
## 2         1991   98125 47.7210 -122.319          1690       7639
## 3            0   98028 47.7379 -122.233          2720       8062
## 4            0   98136 47.5208 -122.393          1360       5000
## 5            0   98074 47.6168 -122.045          1800       7503
## 6            0   98053 47.6561 -122.005          4760     101930
```

Below are descriptions of the attributes in this data set:

- **id**: The id of the house sold
- **date**: The date the house was sold
- **price**: The price the house was sold for - target
- **bedrooms**: Number of bedrooms in the house
- **bathrooms**: Number of bathrooms in the house
- **sqft_living**: Sqaure footage of living space (house)
- **sqrt_lot**: Sqaure footage of the lot
- **floors**: Number of floors in the house
- **waterfront**: Whether the house has a waterfront view
- **view**: Number of viewings of the house
- **condition**: The overall condition of the house
- **grade**: grade of the house based on King County grading conditions
- **sqrt_above**: square footage of house above ground (excluding basement)
- **sqft_basement**: square footage of the basement
- **yr_built**: The year the house was build
- **yr_renovated**: The year the house was renovated (if it was renovated)
- **zipcode**: Zipcode of house
- **lat**: Latitude of house
- **long**: Longitude of house
- **sqrt_living15**: Living area square footage in 2015 (possibly implies renovations and also may impact lot size)
- **sqrt_lot15**: Lot sqaurefootage in 2015 (possibly implies renovations)

### Minor Preprocessing

The 'date' attribute is not in a proper format and must be fixed with lubridate:

```
#  fix date
priceData$date <- gsub("T000000", "", priceData$date)
priceData$date <- ymd(priceData$date)
head(priceData$date)
```

```
## [1] "2014-10-13" "2014-12-09" "2015-02-25" "2014-12-09" "2015-02-18"
## [6] "2014-05-12"
```

In addition, simply knowing the house was renovated is no very informative. If one house were renovated in 1989 and another were renovated in 1990 does knowing this fact allow a better prediction of house prices? Likely not. In addition, dated renovations likely do not add nearly as much value (if any depending on how dated) when compared to newer renovations. For this reason, its more informative to know whether a house has very recent renovations, relatively new renovations, dated renovations, or no renovations at all. Therefore, the yr_renovated attribute will be divded into these catagories and renamed to simply'renovations' to match the semantics of this altered atribute.

- 4 = Very new renovation (less than 2 yrs)
- 3 = New renovation (last 5 yrs)
- 2 = Quite dated renovation (last 10 years)
- 1 = Very dated renovation (more than 10 years old)
- 0 = no renovation

I also thought it might be interesting to see if whether a house having a basement made much of a difference in terms of the sale price of the house. To see this I first will have to add a new attribute to the dataset based on the existing attribute 'sqft_basement' called 'basement' which is binary and simply indicated whether a house has a basement or not.

Finally, some attributes are not interesting to the current analysis, including house id, geographical, and zipcode attributes. A subset which does not include these attributes will be retained for anaylsis.

```
# change year renovated to a categorical attribute based on years since renovation occured
priceData$yr_renovated <-  ifelse(priceData$yr_renovated >= 2014, 4,
                              ifelse(priceData$yr_renovated >= 2010, 3,
                                    ifelse(priceData$yr_renovated >= 2005, 2,
                                          ifelse(priceData$yr_renovated > 0, 1, 0)
                 )
       )
)


#  name does not fit anymore so change it
priceData <- rename(priceData, renovations = yr_renovated)


#  does the house have a basement?
priceData <- mutate(priceData,
               basement = ifelse(priceData$sqft_basement >1, 1, 0))


#  retain only interesting variables
priceData <- dplyr::select(priceData, date:renovations, basement)

# this is what the data now looks like:
head(priceData)
```

```
##         date   price bedrooms bathrooms sqft_living sqft_lot floors waterfront
## 1 2014-10-13  221900        3      1.00        1180     5650      1          0
## 2 2014-12-09  538000        3      2.25        2570     7242      2          0
## 3 2015-02-25  180000        2      1.00         770    10000      1          0
## 4 2014-12-09  604000        4      3.00        1960     5000      1          0
## 5 2015-02-18  510000        3      2.00        1680     8080      1          0
## 6 2014-05-12 1225000        4      4.50        5420   101930      1          0
##   view condition grade sqft_above sqft_basement yr_built renovations basement
## 1    0         3     7       1180             0     1955           0        0
## 2    0         3     7       2170           400     1951           1        1
## 3    0         3     6        770             0     1933           0        0
## 4    0         5     7       1050           910     1965           0        1
## 5    0         3     8       1680             0     1987           0        0
## 6    0         3    11       3890          1530     2001           0        1
```

# Exploratory analysis of the data set

Summary statistics, grouped summaries, and correlations.

```
#  Examine the structure of the data set
str(priceData)
```

```
## 'data.frame':    21613 obs. of  16 variables:
##  $ date         : Date, format: "2014-10-13" "2014-12-09" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ renovations  : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ basement     : num  0 1 0 1 0 1 0 0 1 0 ...
```

Can see that the date is now a date data type and the rest are integers and real numbers.

```
#  Get summary statistics (min, max, median, mean, 1st and 3rd quartiles) for each variable
summary(priceData)
```

```
##       date                price            bedrooms        bathrooms
##  Min.   :2014-05-02   Min.   :  75000   Min.   : 0.000   Min.   :0.000
##  1st Qu.:2014-07-22   1st Qu.: 321950   1st Qu.: 3.000   1st Qu.:1.750
##  Median :2014-10-16   Median : 450000   Median : 3.000   Median :2.250
##  Mean   :2014-10-29   Mean   : 540088   Mean   : 3.371   Mean   :2.115
##  3rd Qu.:2015-02-17   3rd Qu.: 645000   3rd Qu.: 4.000   3rd Qu.:2.500
##  Max.   :2015-05-27   Max.   :7700000   Max.   :33.000   Max.   :8.000
##   sqft_living       sqft_lot           floors        waterfront
##  Min.   :  290   Min.   :    520   Min.   :1.000   Min.   :0.000000
##  1st Qu.: 1427   1st Qu.:   5040   1st Qu.:1.000   1st Qu.:0.000000
##  Median : 1910   Median :   7618   Median :1.500   Median :0.000000
##  Mean   : 2080   Mean   :  15107   Mean   :1.494   Mean   :0.007542
##  3rd Qu.: 2550   3rd Qu.:  10688   3rd Qu.:2.000   3rd Qu.:0.000000
##  Max.   :13540   Max.   :1651359   Max.   :3.500   Max.   :1.000000
##       view          condition         grade          sqft_above
##  Min.   :0.0000   Min.   :1.000   Min.   : 1.000   Min.   : 290
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.: 7.000   1st Qu.:1190
##  Median :0.0000   Median :3.000   Median : 7.000   Median :1560
##  Mean   :0.2343   Mean   :3.409   Mean   : 7.657   Mean   :1788
##  3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.: 8.000   3rd Qu.:2210
##  Max.   :4.0000   Max.   :5.000   Max.   :13.000   Max.   :9410
##  sqft_basement       yr_built     renovations         basement
##  Min.   :   0.0   Min.   :1900   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:   0.0   1st Qu.:1951   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :   0.0   Median :1975   Median :0.00000   Median :0.0000
##  Mean   : 291.5   Mean   :1971   Mean   :0.07065   Mean   :0.3927
##  3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :4820.0   Max.   :2015   Max.   :4.00000   Max.   :1.0000
```

The lowest house price in the data set is $75,000 while the highest is $7,700,000. The average house price in the data set is $540,088. Most houses have 3 or more bedrooms and 2 or more bathrooms while one house has as many as 33 bedrooms and 8 bathrooms. The amount of living space is 2080 square feet but one house is as small as 290 square feet while another is as large as 13,540 square feet. In terms of outdoor space, the average for these data is 15,107 square feet but this number is being pulled up by a property with a massive yard at 1,651,359 square feet; the median lot space is 7,618 square feet. This number is way too high for average outdoor lot space so I assume many of these houses sold were country side homes. The earliest a house was built was 1900 while the median year built is 1975. It also appears that most houses have had at least some renovations done at some point in there history.

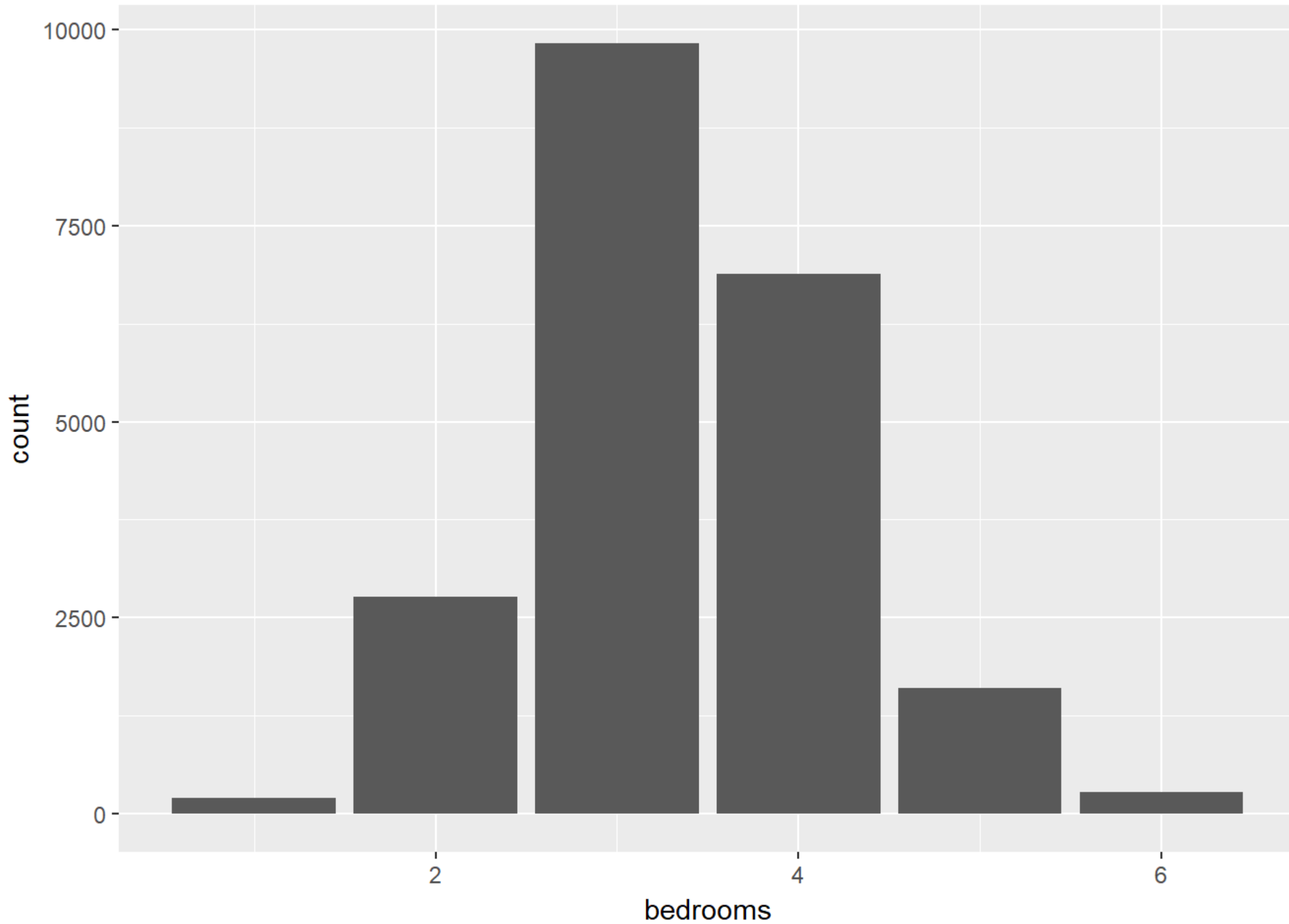
Below is a count of the number of houses which include each number of bedrooms.

```
priceData %>%
  count(bedrooms)
```

```
## # A tibble: 13 x 2
##    bedrooms     n
##       <int> <int>
##  1        0    13
##  2        1   199
##  3        2  2760
##  4        3  9824
##  5        4  6882
##  6        5  1601
##  7        6   272
##  8        7    38
##  9        8    13
## 10        9     6
## 11       10     3
## 12       11     1
## 13       33     1
```

Here are the number of bedrooms, restricted to 1 through 7 bedrooms, as a barchart.

```
num_bedrooms <- filter(priceData, bedrooms >0 & bedrooms < 7)

ggplot(data = num_bedrooms) +
  geom_bar(mapping = aes(x = bedrooms))
```




Most houses have 3-4 bedrooms.

Average house price grouped by number of bedrooms:

```
priceData %>%
  group_by(bedrooms) %>%
  summarise(mean(price))
```

```
## # A tibble: 13 x 2
##    bedrooms `mean(price)`
##       <int>         <dbl>
##  1        0      409503.8
##  2        1      317642.9
##  3        2      401372.7
##  4        3      466232.1
##  5        4      635419.5
##  6        5      786599.8
##  7        6      825520.6
##  8        7      951184.7
##  9        8     1105076.9
## 10        9      893999.8
## 11       10      819333.3
## 12       11      520000.0
## 13       33      640000.0
```

There is an apparent linear increase in average house prices for each additional bedroom, up to about 8 bedrooms.

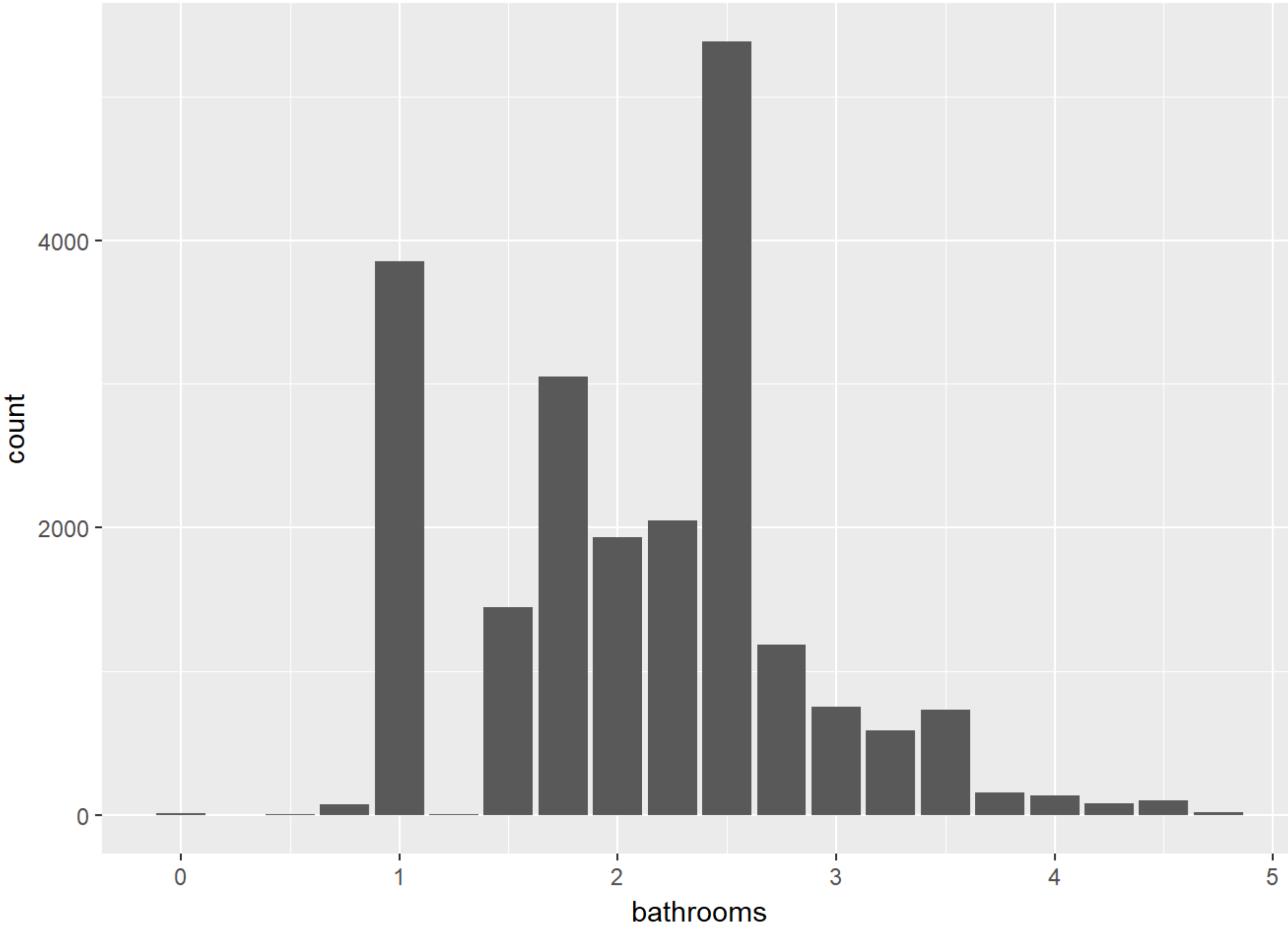
Below is a count of the number of houses which include each number of bathrooms

```
priceData %>%
  count(bathrooms)
```

```
## # A tibble: 30 x 2
##    bathrooms     n
##        <dbl> <int>
##  1      0.00    10
##  2      0.50     4
##  3      0.75    72
##  4      1.00  3852
##  5      1.25     9
##  6      1.50  1446
##  7      1.75  3048
##  8      2.00  1930
##  9      2.25  2047
## 10      2.50  5380
## # ... with 20 more rows
```

Below is a display of the number of bathrooms, restricted to less than 5 bathrooms, as a barchart.

```
num_bathrooms <- filter(priceData, bathrooms < 5)

ggplot(data = num_bathrooms) +
  geom_bar(mapping = aes(x = bathrooms))
```




It appears that most houses have 1, 1.75, or 2.5 bathrooms.

Average price grouped by number of bathrooms:

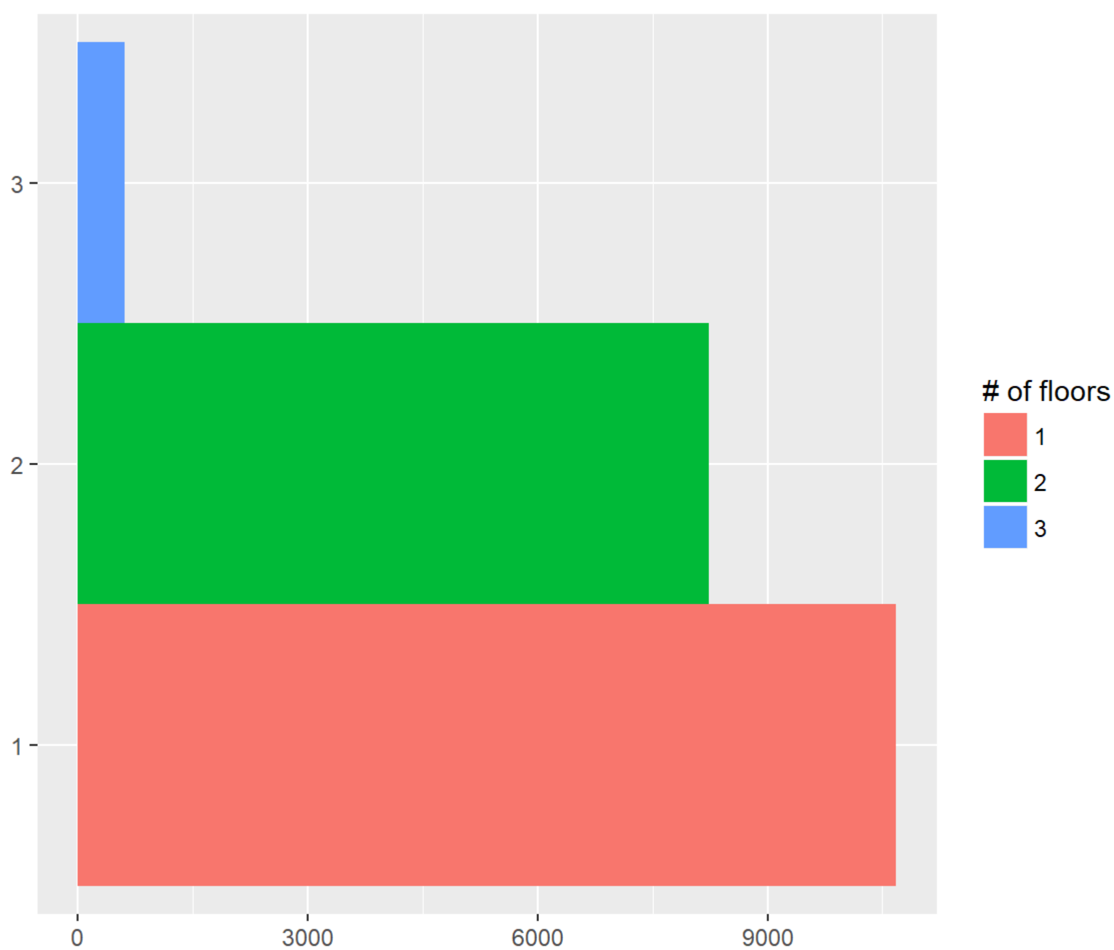
```
priceData %>%
  group_by(bathrooms) %>%
  summarise(mean(price))
```

```
## # A tibble: 30 x 2
##    bathrooms `mean(price)`
##        <dbl>         <dbl>
##  1      0.00      448160.0
##  2      0.50      237375.0
##  3      0.75      294520.9
##  4      1.00      347041.2
##  5      1.25      621216.7
##  6      1.50      409322.2
##  7      1.75      454896.1
##  8      2.00      457889.7
##  9      2.25      533676.8
## 10      2.50      553596.5
## # ... with 20 more rows
```

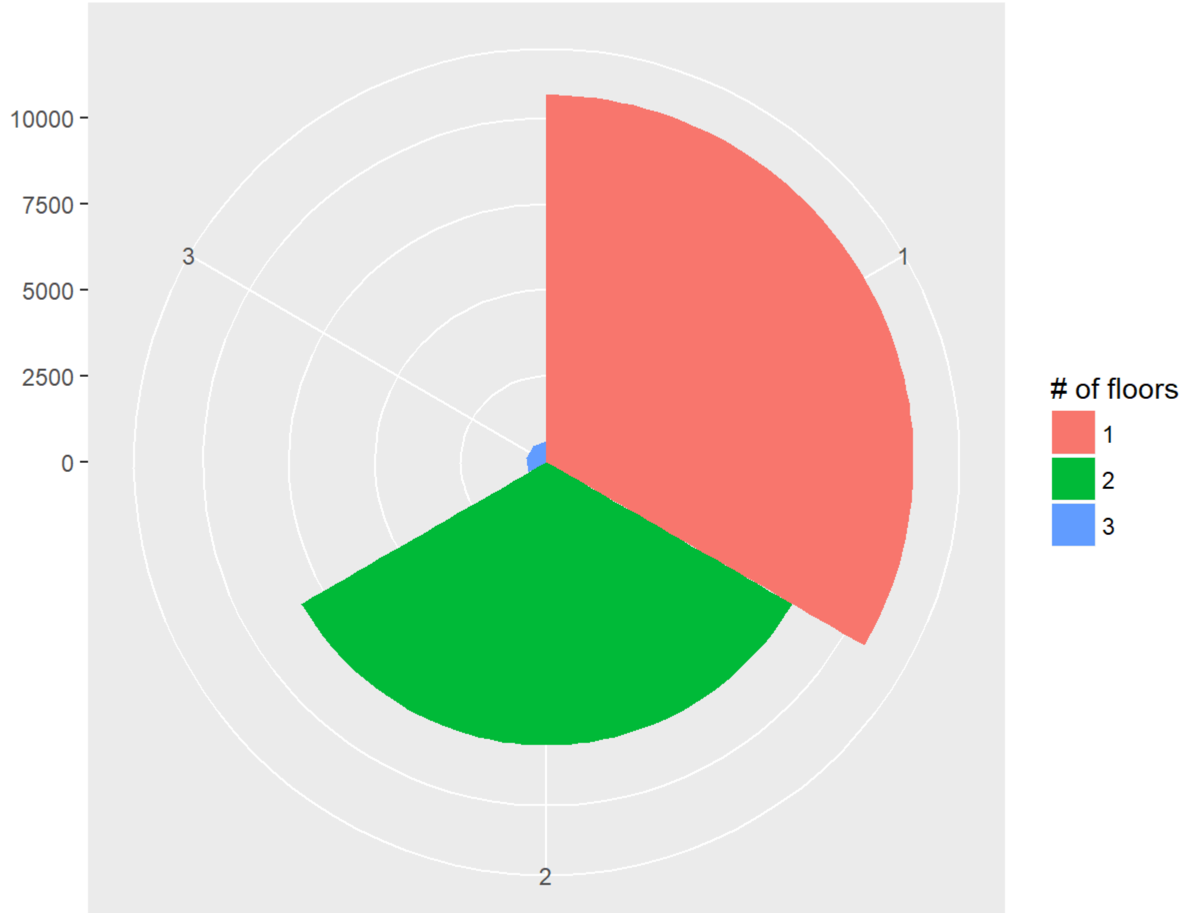
There appears to be a gradual increase in average of house prices with 2, 3, and 4 bathrooms but this pattern is not true for other numbers of bathrooms. This would likely be due to the type of housing unit which would include other numbers of bathrooms; condos and small houses in downtown centers may only have 1 bathroom but be considerably more expensive than larger homes in more rural areas which have more bathrooms.

Below is a different way to visualize a barchart. A count of the number of floors for each house in the data set (restricted to 1,2,3 or 4 floors) is displayed below. The bars have been coloured according to the number of floors and the x and y axis have been flipped.

```
num_floors <- filter(priceData, floors == 1 | floors == 2 | floors == 3 | floors == 4)


bar <- ggplot(data = num_floors) +
  geom_bar(
    mapping = aes(x = as.factor(floors), fill = as.factor(floors)),
    show.legend = T,
    width = 1
  ) +
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL, fill = "# of floors")

bar + coord_flip()
```



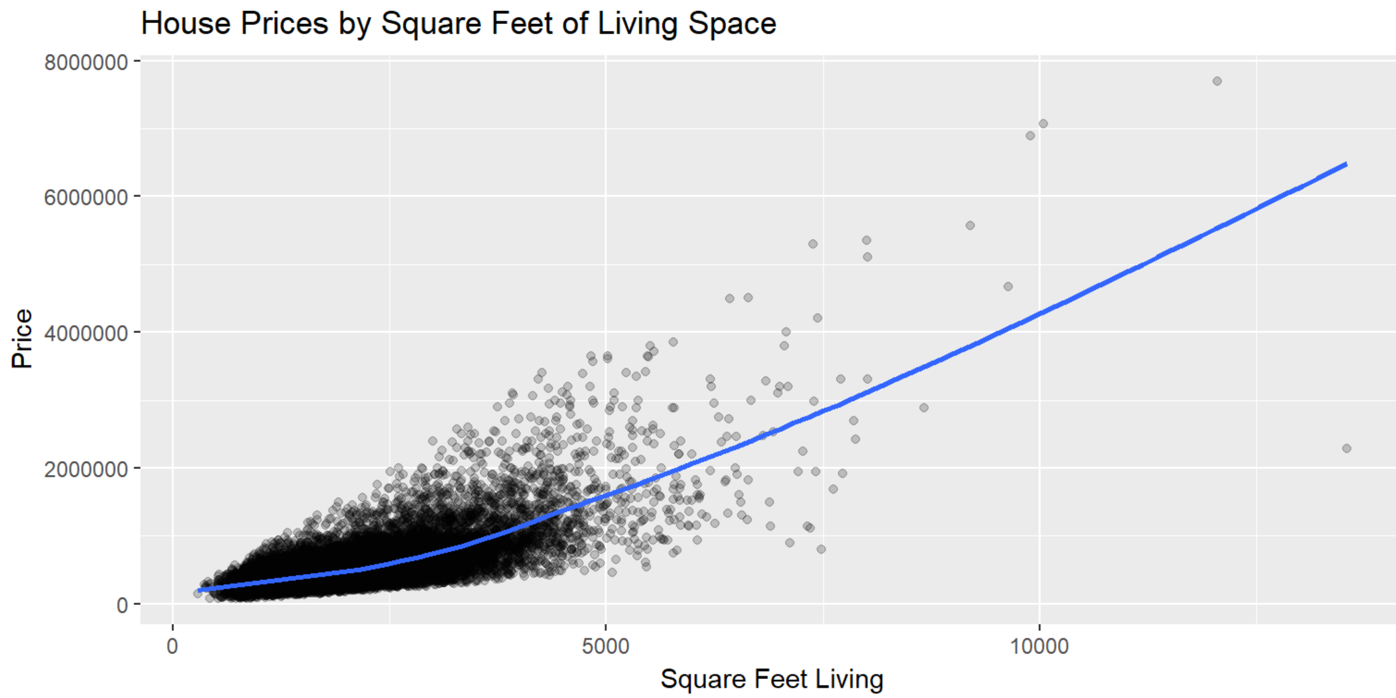
```
bar + coord_polar()
```




The second plot is just another way to visualize the information, this time as a polar plot. It appears that the majority of houses have only 1 floor, followed by 2 floors, and very few have 3 floors. There was not a single house in this data set which had 4 floors.

Below is a scatterplot of house prices in terms of square feet of living space

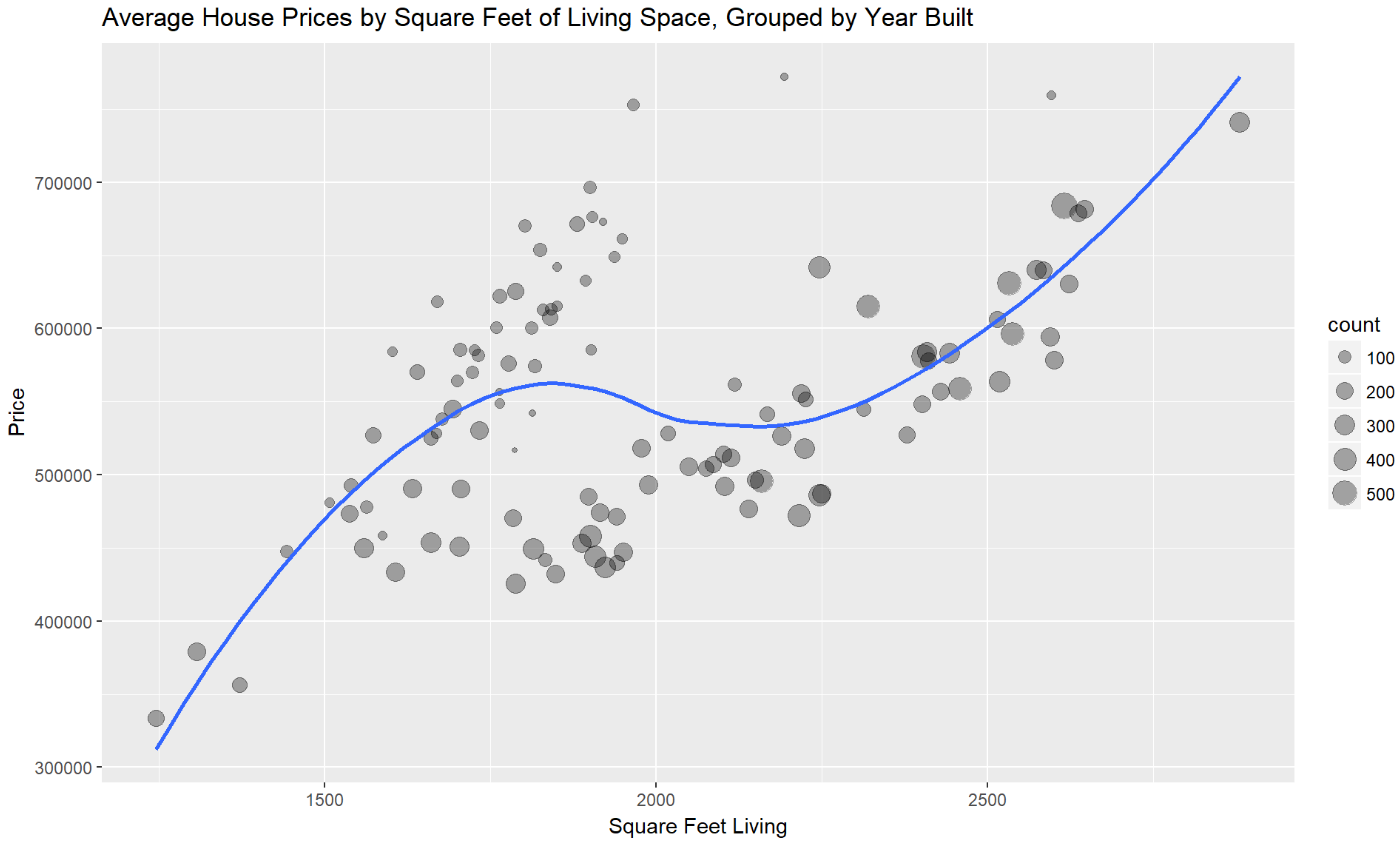
```
options(scipen = 999)

ggplot(data = priceData, mapping = aes(x = sqft_living, y = price)) +
  geom_point(alpha = 1/5) +
  geom_smooth(se = FALSE) +
  ggtitle("House Prices by Square Feet of Living Space") +
  xlab("Square Feet Living") +
  ylab("Price")
```




A very clear trend is present here: as the size of the house increases so does the price. Most houses are 800 to 4000 square feet inside.

The plot below is more complex, it displays the average price of the house plotted against square feet of living space where data have been grouped by year build and displayed as counts of varrying sizes depending on the number of houses in any one group. Furthermore, the data has been filtered to not display any points where there are less than 20 houses within a group.

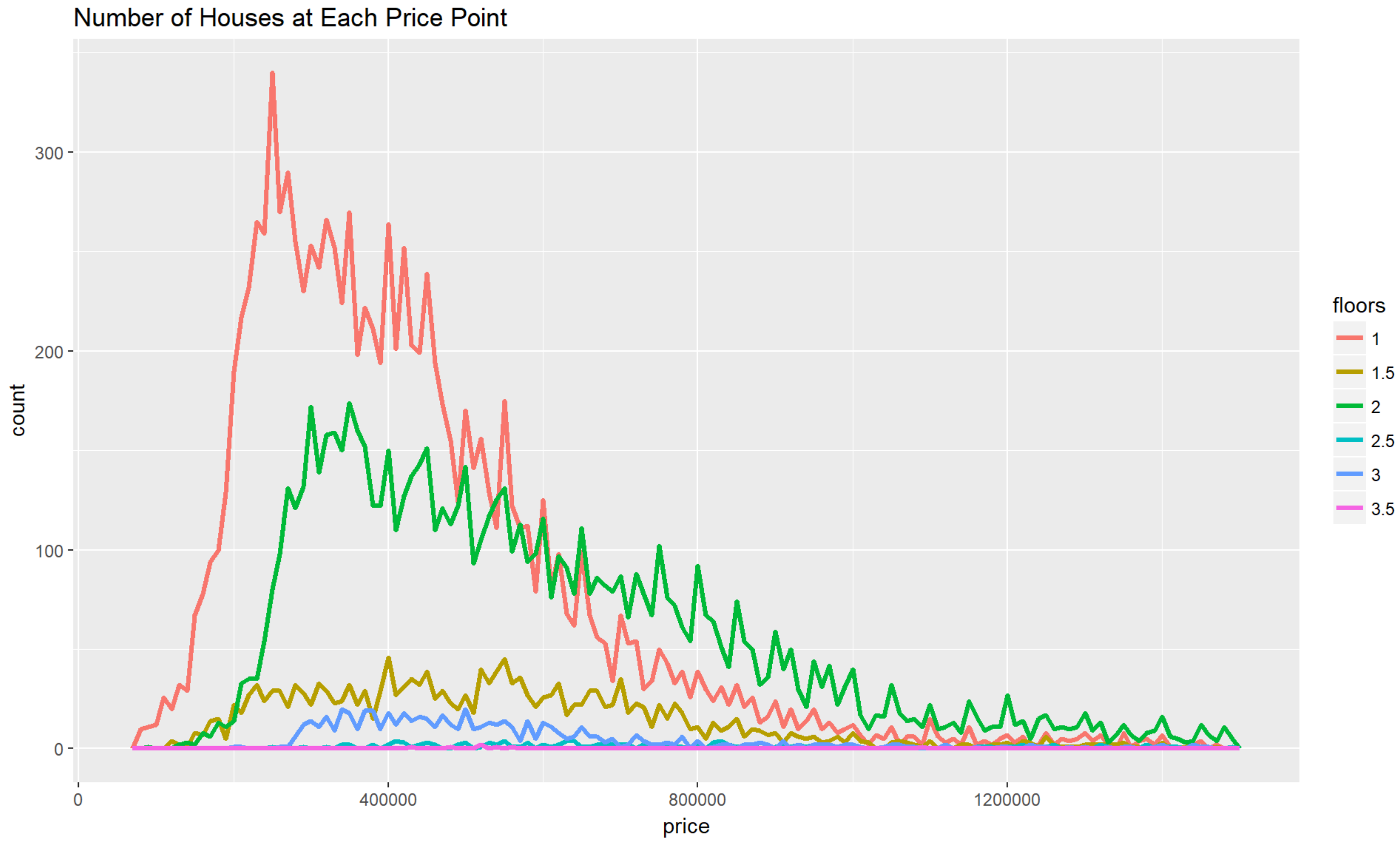
```
price_by_sqftliving <- priceData %>%
  group_by(yr_built) %>%
  summarise(
    count = n(),
    sqftlive = mean(sqft_living, na.rm = TRUE),
    pricelive = mean(price, na.rm = TRUE)
  ) %>%
  filter(count > 20)



ggplot(data = price_by_sqftliving, mapping = aes(x = sqftlive, y = pricelive)) +
  geom_point(aes(size = count), alpha = 1/3) +
  geom_smooth(se = FALSE)  +
  ggtitle("Average House Prices by Square Feet of Living Space, Grouped by Year Built") +
  xlab("Square Feet Living") +
  ylab("Price")
```




The same trend is seen in this plot as the previous plot with additional information. Many groups of houses with similar squarefootage and prices are not visibile. There appears to be two smaller trends present, possibly showing smaller city houses and larger, more rural houses.

Below is a plot of the number of houses at each price point, grouped by number of floors of the house.

```
normal_houses <- filter(priceData, price < 1500000)

normal_houses$floors <- factor(normal_houses$floors)


ggplot(data = normal_houses, aes(x = price)) +
  geom_freqpoly(mapping = aes(colour = floors), binwidth = 10000, size = 1.2) +
  ggtitle("Number of Houses at Each Price Point")
```

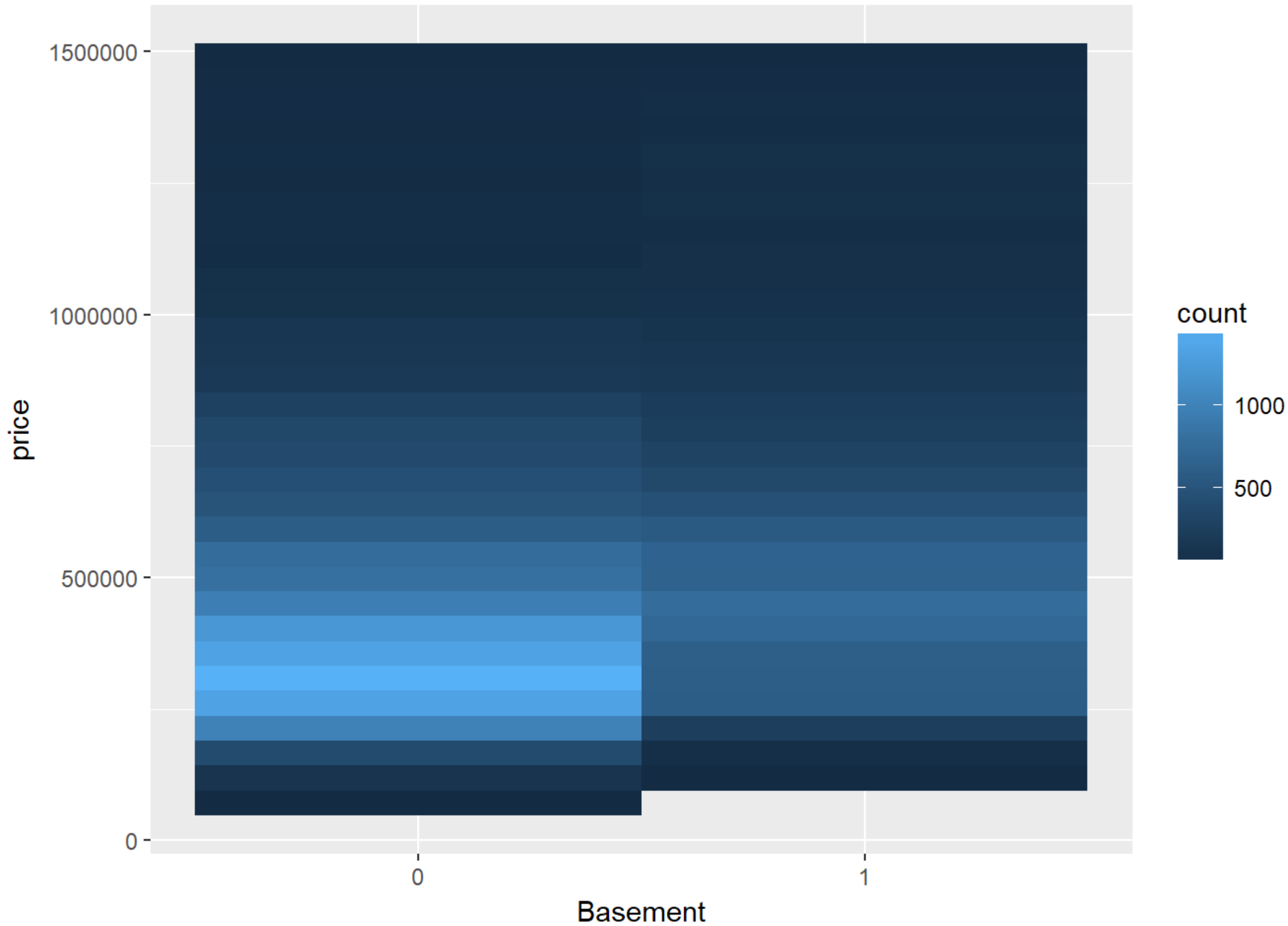



Notice the spikes in the plots at equal interval which reflect pricing strategies implemented by realators. This trend is most apparent in houses with the most common number of floors, 1 and 2 floors. The trend in pricing is less clear for houses with a less common number of floors perhaps because these houses are priced differently as there might be fewer comps in the local market.

br />

Below is a heatmap of the counts of each house price for houses with and without basements

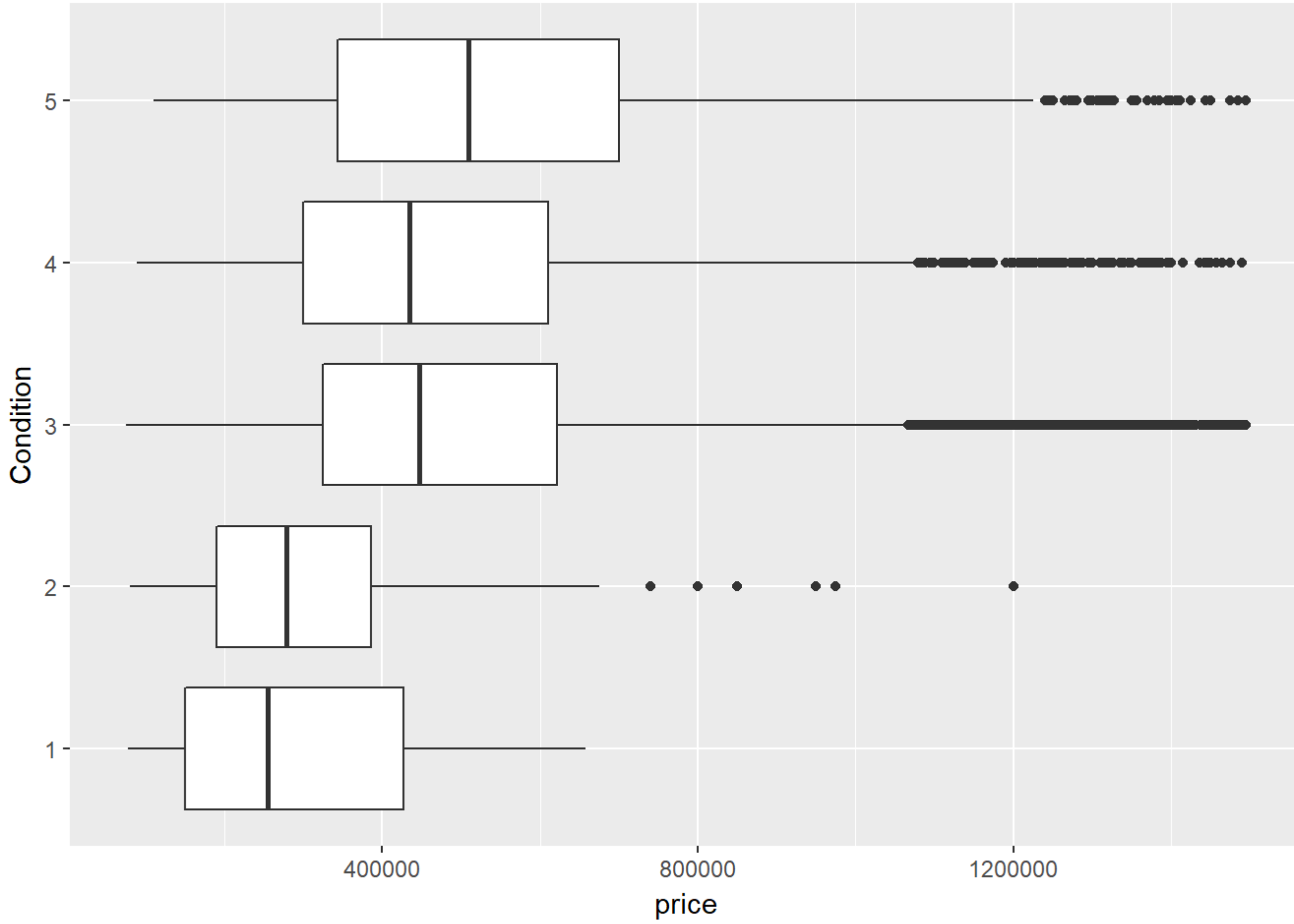
```
ggplot(data = normal_houses) +
  geom_bin2d(mapping = aes(x = as.factor(basement), y = price)) +
  xlab("Basement")
```




This plot shows that most houses at lower prices do not have basements while houses which go for the highest prices more often have basements than do not.

Below is a side-by-side boxplot of the condition of houses plotted against house price.

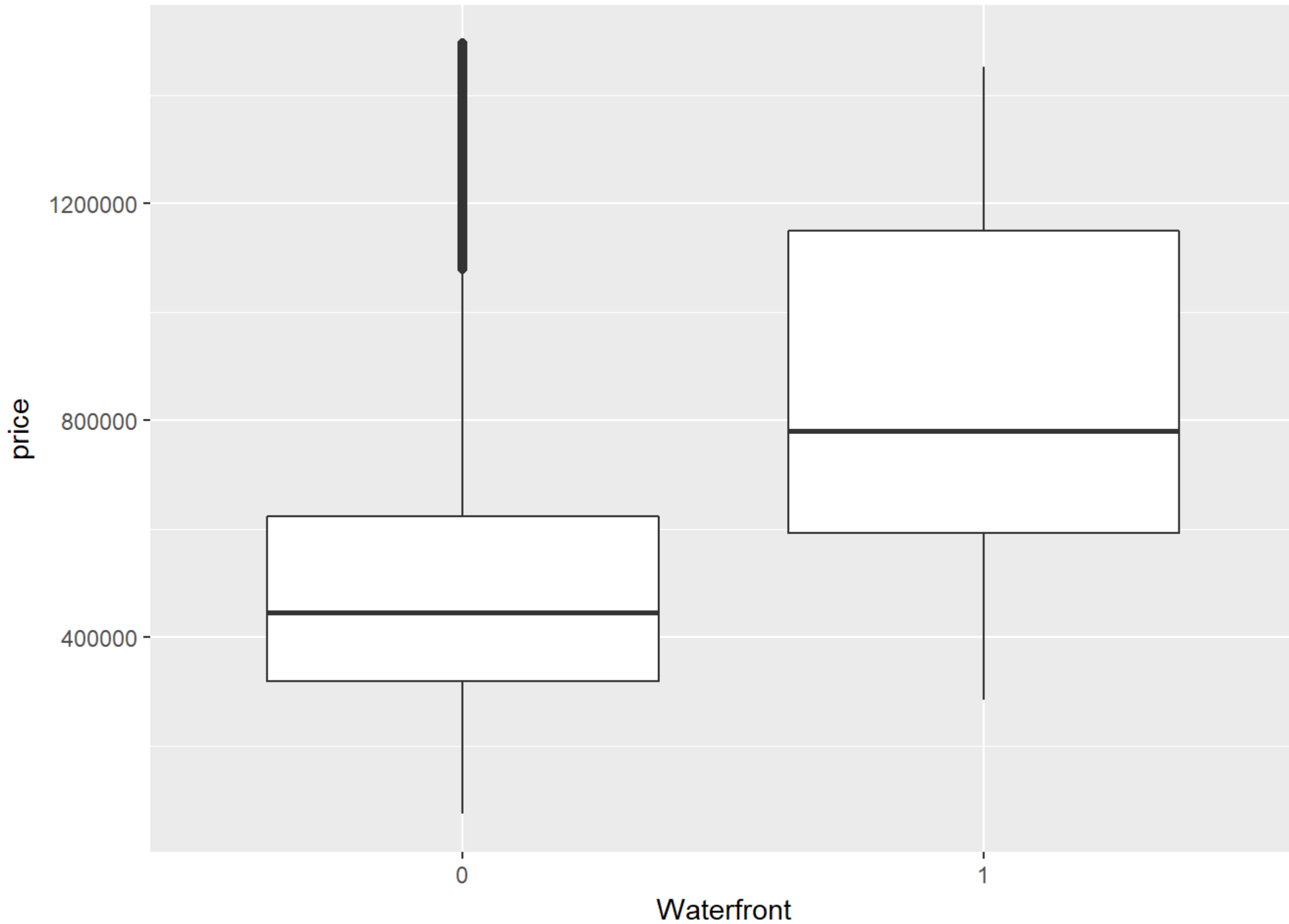
```
ggplot(data = normal_houses, mapping = aes(x = as.factor(condition), y = price)) +
  geom_boxplot() +
  coord_flip() +
  xlab("Condition")
```




As seen above, houses in poor conditions sell for the leastm with very few outliers where such houses sell for higher than expected prices. That being said, houses in average to great condition sell for similar prices overall. Houses in great condition sell for the most overall.

Below is a boxplot of the prices of houses which do and do not have waterfront properties ```

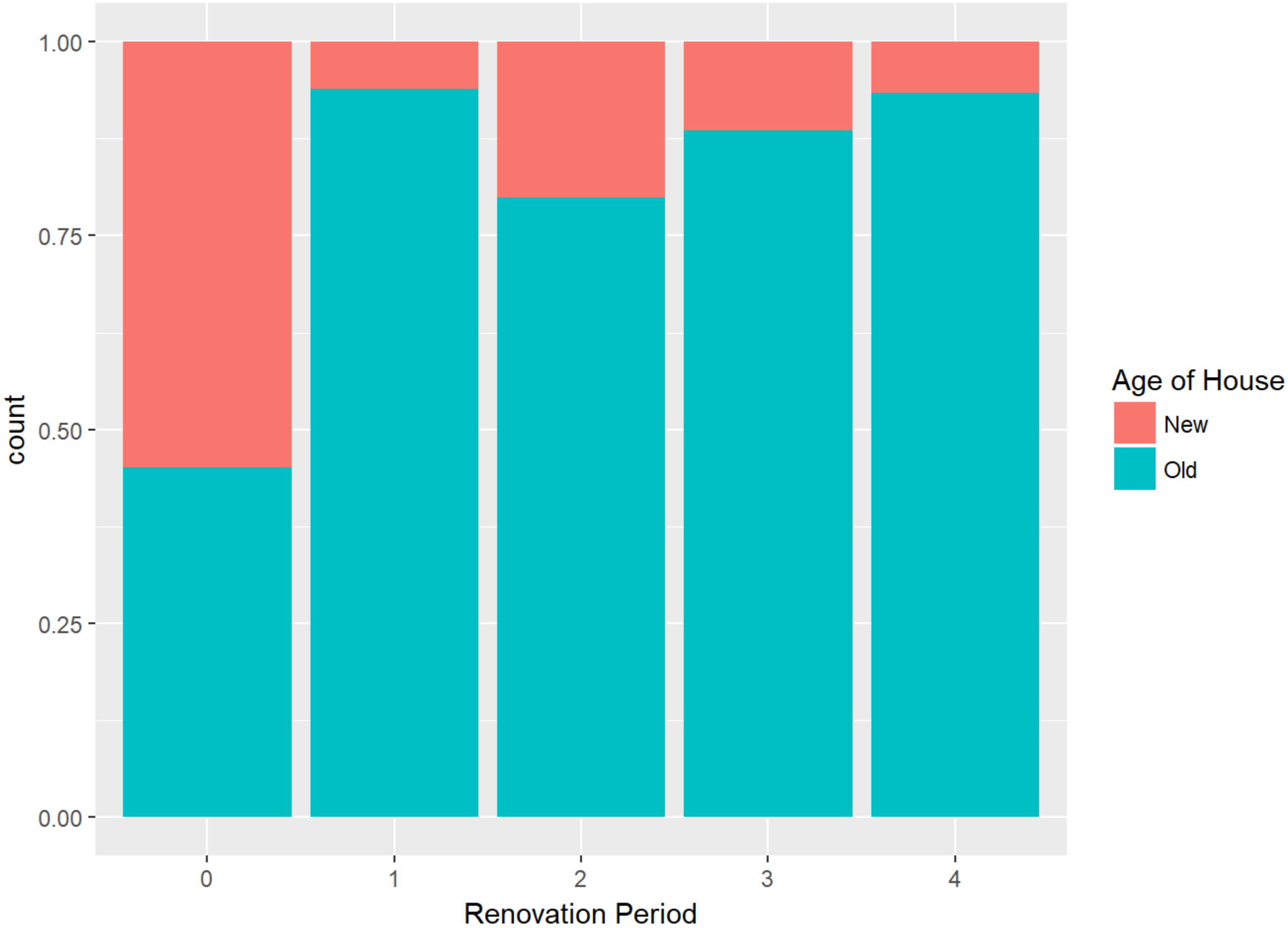
```
ggplot(data = normal_houses, mapping = aes(x = as.factor(waterfront), y = price)) +
  geom_boxplot() +
  xlab("Waterfront")
```




It seems that houses with waterfront properties sell for more on average but a fair number of houses that do not waterfront properties sell for as much or more than houses that do.

Below is a barchart of the proportions of houses which have had renovations, coloured to show old and new houses.

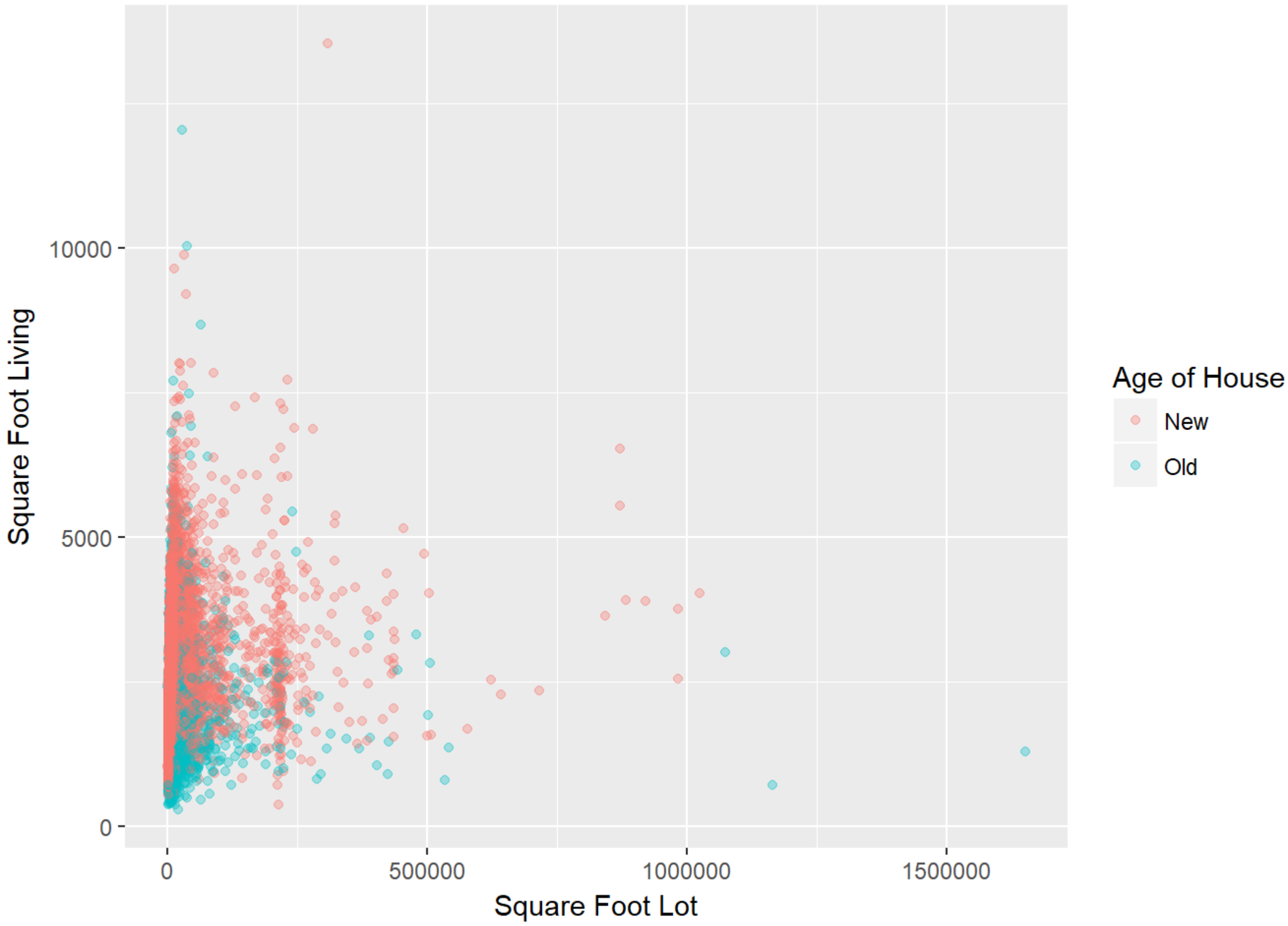
```
priceData2 <- mutate(priceData, new_old = ifelse(yr_built > 1970, "New", "Old"))

ggplot(data = priceData2) +
  geom_bar(mapping = aes(x = as.factor(renovations), fill = as.factor(new_old)), position = "fil
l") +
  labs(x = "Renovation Period", fill = "Age of House")
```




Not surprisingly, more recent renovations were done on older houses while most houses which have not had some form of renovations are newer houses.

The next plot is a scatter plot of the square feet of living space plotted against the square feet of lot space. Data points are coloured to show whether they are new or old homes. It appears that more newer houses are being built to be large but it is hard to tell due to the size of the x and y axses. The second plot is zoomed in to capture the majority of the data such that trends are more noticible.

```
p <- ggplot(data = priceData2) +
  geom_point(mapping = aes(x = sqft_lot, y = sqft_living, color = new_old), alpha = 1/3) +
  labs(x = "Square Foot Lot", y = "Square Foot Living", color = "Age of House")


p
```




```
p + coord_cartesian(xlim = c(0,25000),ylim= c(0,7500))
```


As seen above, more newer homes are being built to be larger in terms of indoor space but smaller in terms of outdoor space. This likely has to do with population increases which occur slowly overtime or perhaps the development of different economic pressures which have made land space more costly.

## Correlations

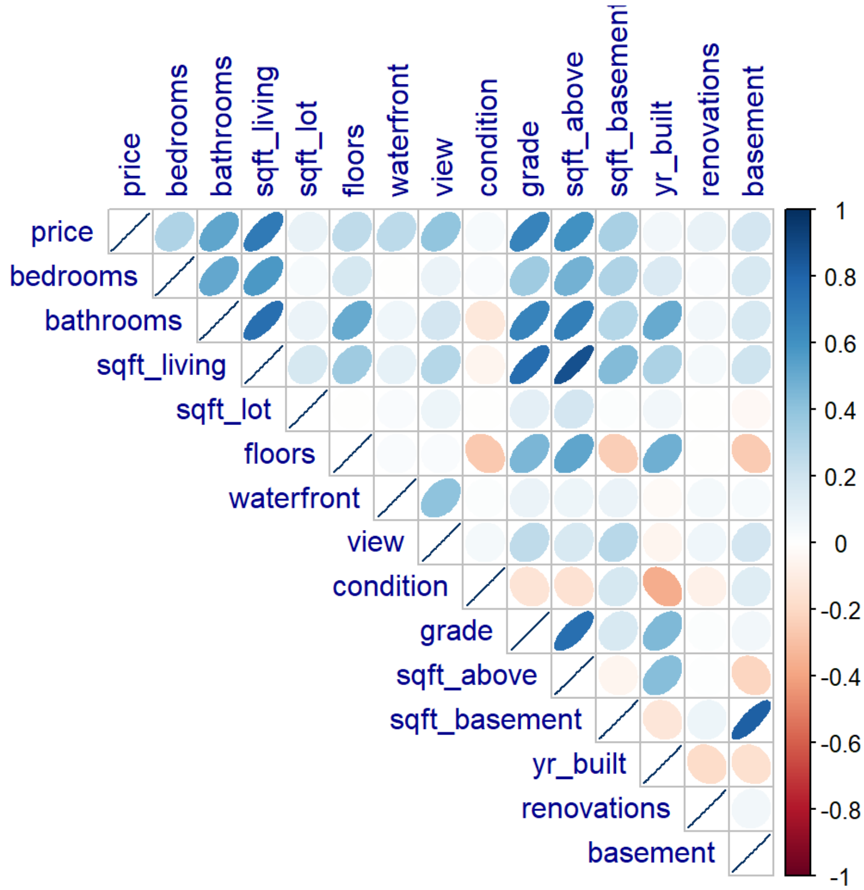
Next, I want to visualize an r-matrix of the bivariate correlations between each of the variables.

```
# begin with correlations
priceSub <- dplyr::select(priceData, -date)

#  the correlation with the basement variable (binary) is a point-biseral correlation
cors <- round(cor(priceSub, use = "complete.obs", method = "pearson"), 3)


# rcorr includes pvalues
cors2 <- rcorr(as.matrix(priceSub))
```

```
#  these can be combined and flattened into a nice table

flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
  )
}



corMat <- rcorr(as.matrix(priceData[,-c(1)]))
#  filter output to only show correlations with price
filter(flattenCorrMatrix(corMat$r, corMat$P), row == 'price')
```

```
##      row        column        cor                      p
## 1  price      bedrooms 0.30834961 0.00000000000000000000
## 2  price     bathrooms 0.52513748 0.00000000000000000000
## 3  price   sqft_living 0.70203507 0.00000000000000000000
## 4  price      sqft_lot 0.08966086 0.00000000000000000000
## 5  price        floors 0.25679389 0.00000000000000000000
## 6  price    waterfront 0.26636943 0.00000000000000000000
## 7  price          view 0.39729348 0.00000000000000000000
## 8  price     condition 0.03636179 0.00000008935666118148З
## 9  price         grade 0.66743428 0.00000000000000000000
## 10 price    sqft_above 0.60556728 0.00000000000000000000
## 11 price sqft_basement 0.32381603 0.00000000000000000000
## 12 price      yr_built 0.05401153 0.00000000000001776357
## 13 price   renovations 0.09796116 0.00000000000000000000
## 14 price      basement 0.18023008 0.00000000000000000000
```

```
corrplot(cors, method = "ellipse", type ='upper', tl.col = "darkblue")
```




It appears that square foot of living space is not strongly related to housing prices. This makes sense because the most expensive homes are either those inside city-centers (with very small living areas) or those in the country with large lots. Year built is also not storngly related to the price of the house price which again makes sense as a new and old homes can be worth a lot or be very economically priced. Whether the house had renovations or not also is not correlated to price because, as previously mentioned, most houses in this data set have had renovations at some point or another. Something interesting to consider is that the square feet of the basemet is moderately related to house price but the fact that the house has a basement or not is not realted. This is likely because the sqft of the basement is correlated with the sqft of the house which is highly correlated with the price of the house. The number of bathrooms and grade of the property are also highly correlated with house price.

Another concern when looking at this r-matrix is the near perfect collinearity between a few of the variables. I will make sure to not include a subset of variables which includes these pairs to avoid multicollinearity being an issue.

# Stepwise Multiple Regression - Forward selection:

I want to determine which variables are good predictors of house prices. Stepwise regression will be used to determine significant predictors. Variables will be added to the model in a forward direction and only retained if the AIC is lowered as result of them being added.

First, I need to get rid of some variables I do not want to include in the regression analysis. I need to get rid of a couple categorical variables which have more than 2 levels as well as one of the variables which duplicates informationof another variable (basement and sqft_basement). I also need to get rid of sqft_above due to it being alsomost perfectly linearly related to sqft_living (due to the two measuring almost the same thing).

```
priceData <- dplyr::select(priceData, price:sqft_lot, waterfront, basement)
```

Next, I will define the basic model as only containing the intercept. This will be the initial state. The complete model will contain all variables are predictors.

```
basicModel<-lm(price ~ 1, data=priceData)


forwardModel <- stepAIC(basicModel, scope=list(lower=~1, upper = ~ bedrooms + bathrooms +sqft_li
ving +
                                               sqft_lot + waterfront +  basement),
                          direction="forward", trace = 0)


forwardModel$anova
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## price ~ 1
## 
## Final Model:
## price ~ sqft_living + waterfront + bedrooms + sqft_lot + basement + 
##     bathrooms
## 
## 
##            Step Df         Deviance Resid. Df       Resid. Dev      AIC
## 1                                       21612 2912916761921299 553875.8
## 2 + sqft_living  1 1435640399598809     21611 1477276362322490 539203.5
## 3  + waterfront  1  110238185400763     21610 1367038176921727 537529.4
## 4    + bedrooms  1   30630523480111     21609 1336407653441616 537041.6
## 5    + sqft_lot  1    5062087384297     21608 1331345566057319 536961.6
## 6    + basement  1    4108420762289     21607 1327237145295030 536896.8
## 7   + bathrooms  1     312591973992     21606 1326924553321038 536893.7
```

```
summary(forwardModel)
```

```
##
## Call:
## lm(formula = price ~ sqft_living + waterfront + bedrooms + sqft_lot +
##     basement + bathrooms, data = priceData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1444544  -138408   -19212   102875  4269828
##
## Coefficients:
##                  Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)  69792.17435   6672.36639  10.460 < 0.0000000000000002 ***
## sqft_living    298.65416      3.05051  97.903 < 0.0000000000000002 ***
## waterfront  790597.52262  19661.57218  40.210 < 0.0000000000000002 ***
## bedrooms    -53123.32494   2260.69481 -23.499 < 0.0000000000000002 ***
## sqft_lot        -0.34803      0.04163  -8.361 < 0.0000000000000002 ***
## basement     28945.49434   3541.28650   8.174 0.000000000000000315 ***
## bathrooms     7624.68030   3379.62611   2.256               0.0241 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 247800 on 21606 degrees of freedom
## Multiple R-squared:  0.5445, Adjusted R-squared:  0.5443
## F-statistic:  4304 on 6 and 21606 DF,  p-value: < 0.00000000000000022
```

The final model for predicting house prices includes sqft_living, waterfront, bedrooms, renovations, sqft_lot, and basement. This means that house prices are best predicted by the square feet of living space and lot space, the number of bedrooms, whether the househas a waterfront view or not, whether the house has had renovations or not, and whether or not the house has a basement.

In conclusion:

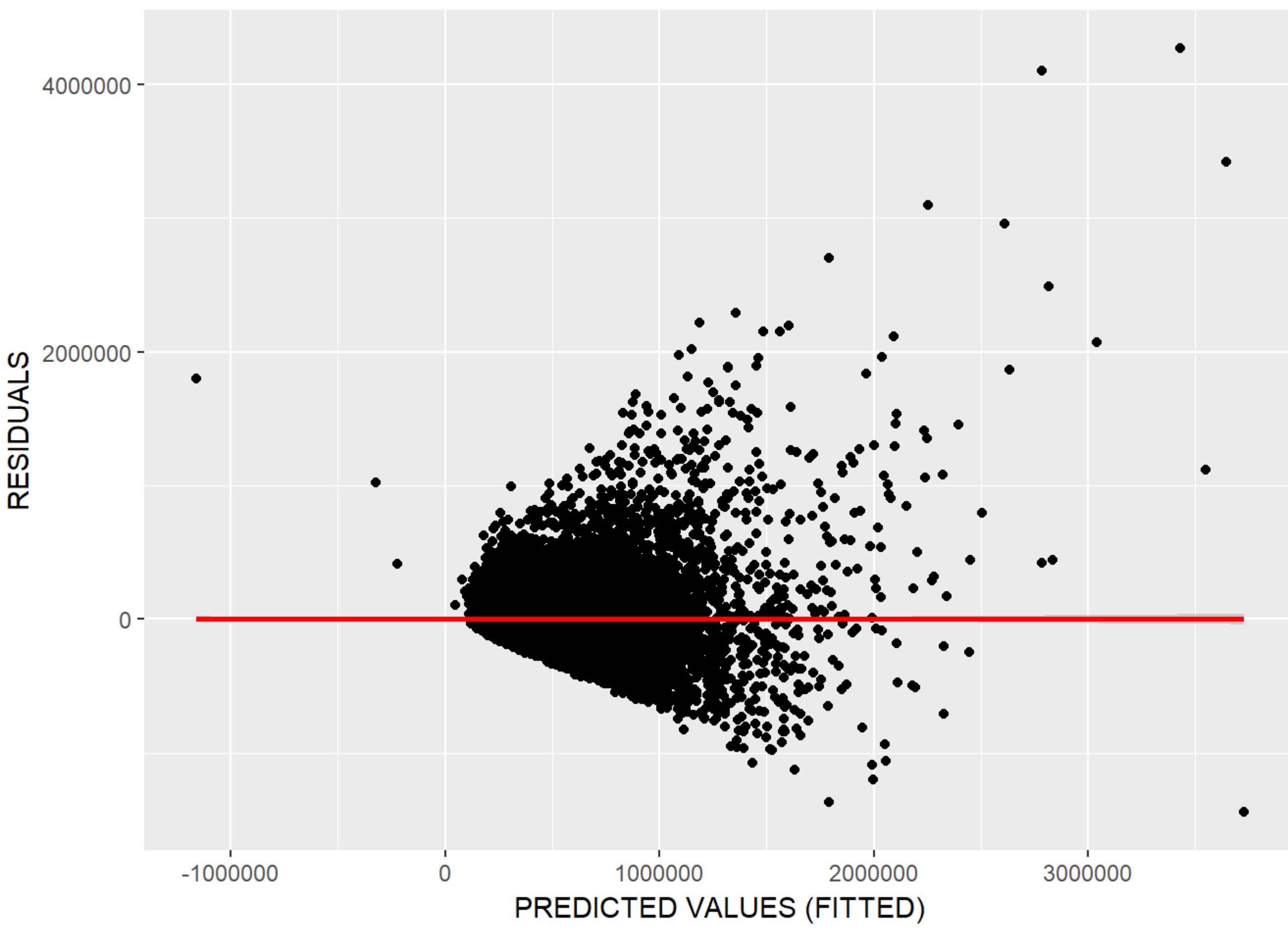
A forward-selection multiple regression model was built to determine the variables which best predicted house prices. Ten variables were included in the regression model; these predictors account for approximately 54.4% of the variation in house prices. For these data, $F(6, 21606) = 4304, p < .00001$. Therefore, the regression model, including the ten independant variables, is significantly better for predicting house prices compared to if we simply used a basic model to predict house prices (the mean/intercept). In other words, the fit of the full model is significantly better than the fit of the basic model.

# Checking Assumptions

To draw conclusions about a population based on any regression analysis, one must check whether a number are met. Some of these assumptions cannot be checked until after the regression model has run.

- **Homoscedasticity**

```
# the final model after forward selection:
finalModel<- lm(formula = price ~ sqft_living + waterfront + bedrooms + sqft_lot +
    basement + bathrooms, data = priceData)


# get predicted and residual values based on the regression model:
priceData$predicted = round(predict(finalModel),3)
priceData$residuals = round(resid(finalModel),3)

# plot predicted and residual values:
scatter <- ggplot(priceData, aes(predicted, residuals))
scatter + geom_point() +
  geom_smooth(method = "lm", colour = "Red", se = T) +
  labs(x = "PREDICTED VALUES (FITTED)", y = "RESIDUALS")
```




It appears that this assumption has been violated. The variability of the error is not roughly constant across the regression line. This is seen in a very noticible fanning-out of the residuals. This can be confirmed with a Breusch-Pagan test; this test is a chi-squared test for independance which determines whether the variance of the errors of regression is dependent on the values of the independent variables.

```
ncvTest(finalModel)
```

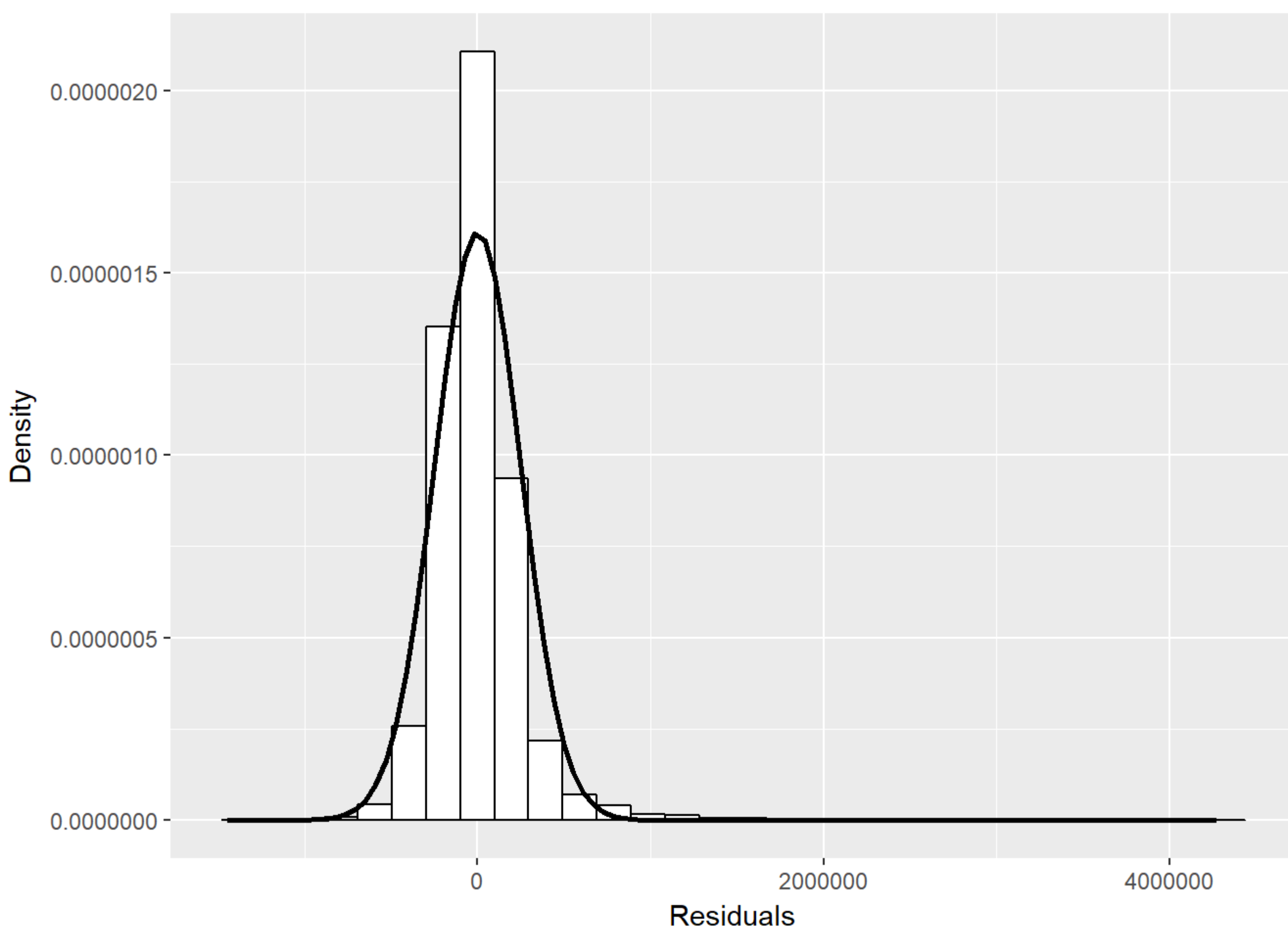
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 30999.46    Df = 1     p = 0
```

The nul-hypothesis of this test are that variances are not dependant on the values of the independant variables and therefore homoscedastic. The alternative hypothesis is that they are dependant and thus, heteroscedastic. This test has confirmed that the errors are certainly heteroscedastic and therefore this assumption is violated.

- **Normally Distributed Errors**

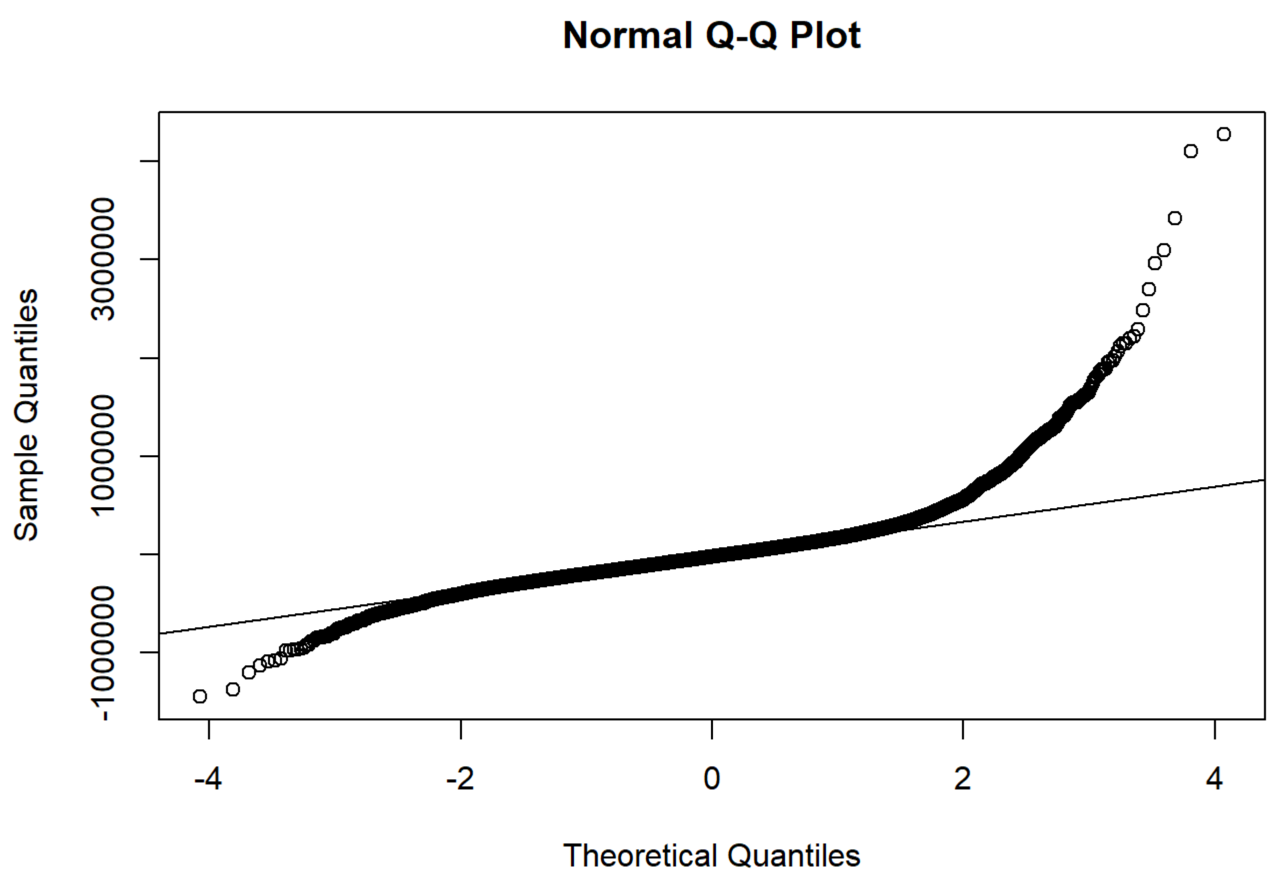
Histogram:

```
aSaleData.errors <- ggplot(priceData, aes(residuals)) +
  labs(legend.position = "none") +
  geom_histogram(aes(y=..density..), colour="black", fill="white") +
  stat_function(fun = dnorm,
                args = list(mean = mean(priceData$residuals, na.rm = TRUE),
                            sd = sd(priceData$residuals, na.rm = TRUE)),
                colour = "black", size = 1) +
  labs(x="Residuals", y = "Density")

aSaleData.errors
```



QQplot

```
qqplot.residuals <- qqnorm(priceData$residuals,
                           xlab = "Theoretical Quantiles",
                           ylab = "Sample Quantiles")
qqline(priceData$residuals)
```




It is clear that the residuals are not normally distributed. A major reason why this is so is because there appears to be one or a few instances where the predicted house price was very, very different from the actual house price, resulting in a massive residual value(s).

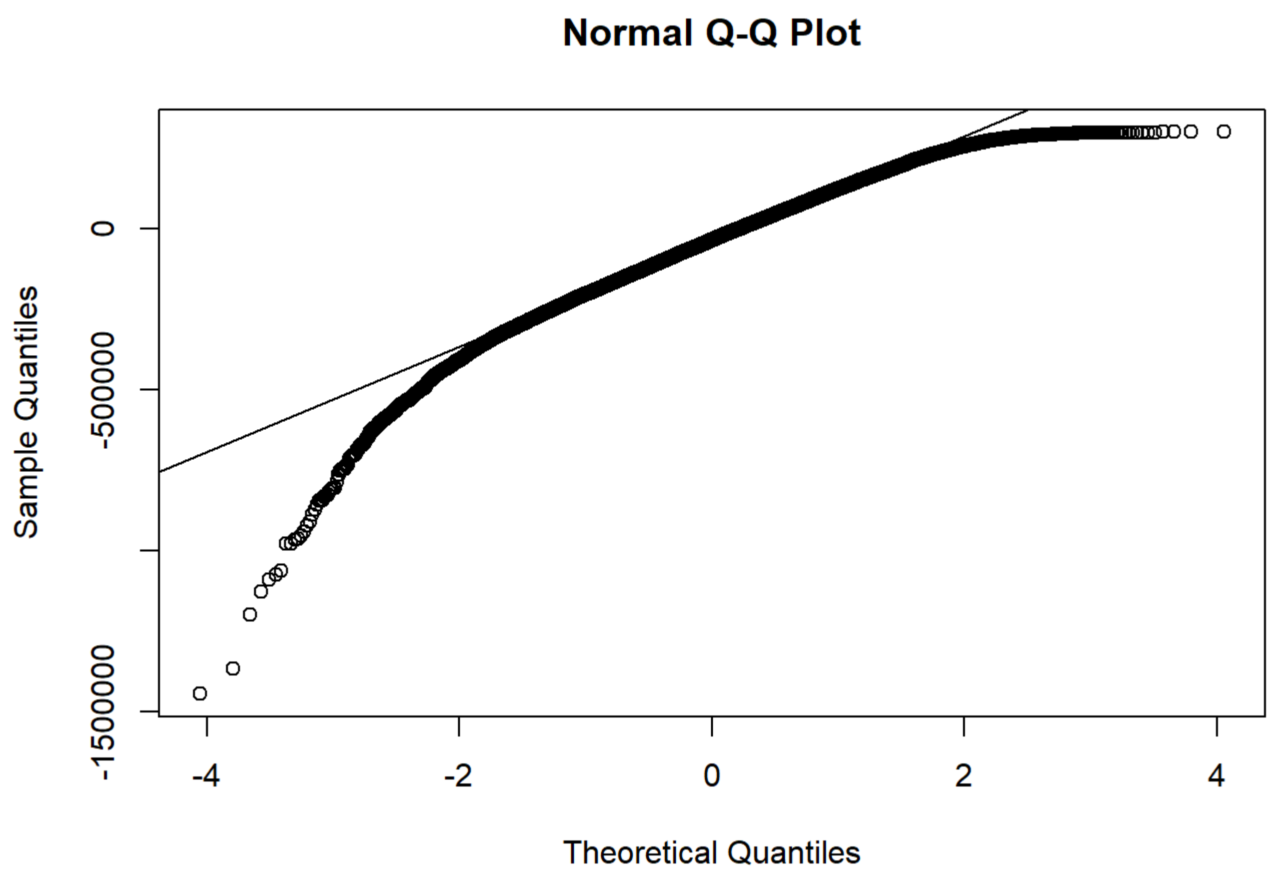
Let's look at the residual values in descending order:

```
head(arrange(priceData, desc(residuals)), 10)
```

```
##      price bedrooms bathrooms sqft_living sqft_lot waterfront basement
## 1  7700000        6      8.00       12050    27600          0        1
## 2  6885000        6      7.75        9890    31374          0        1
## 3  7062500        5      4.50       10040    37325          1        1
## 4  5350000        5      5.00        8000    23985          0        1
## 5  5570000        5      5.75        9200    35069          0        1
## 6  4489000        4      3.00        6430    27517          0        0
## 7  5300000        6      6.00        7390    24829          1        1
## 8  3650000        5      3.75        5020     8694          0        1
## 9  3400000        4      4.00        4260    11765          0        1
## 10 3800000        3      4.25        5510    35000          0        1
##    predicted residuals
## 1    3430172   4269828
## 2    2781860   4103140
## 3    3643527   3418973
## 4    2252130   3097870
## 5    2612376   2957624
## 6    1790943   2698057
## 7    2814756   2485244
## 8    1357932   2292068
## 9    1184915   2215085
## 10   1605176   2194824
```

As seen above, a few of the residual values are much larger than the others, with one being incredibly large in comparison to the rest. To visualize the influence these of few residual values, below is another qqplot of the residuals without those very large points:

```
priceData$residuals<-ifelse(priceData$residuals > 300000, NA, priceData$residuals)


priceData.residuals <- qqnorm(priceData$residuals,
                           xlab = "Theoretical Quantiles",
                           ylab = "Sample Quantiles")

qqline(priceData$residuals)
```

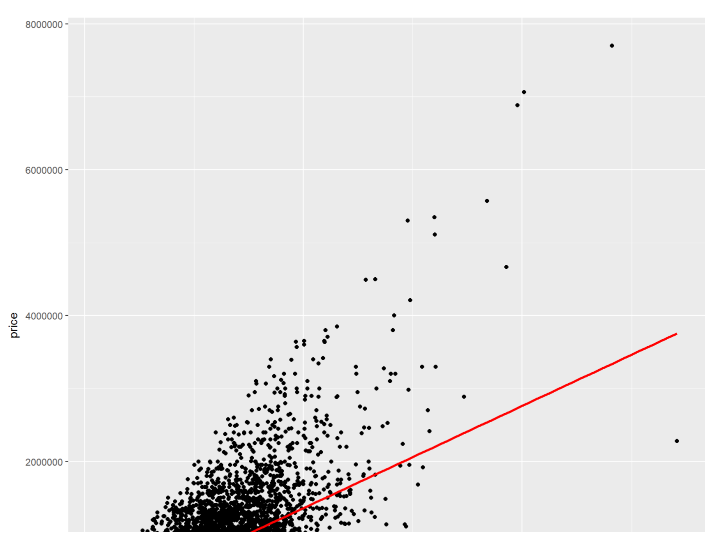


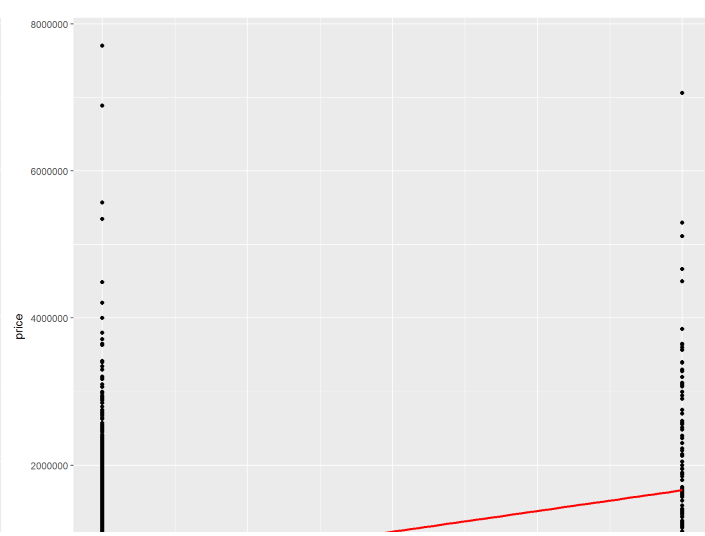

Because the qqplot is now pulled up towards the top left corner, that leads me to believe that the model produced a lot of very large negative residual values. Because the residuals are simply the actual house price minus the predicted house price, many large negative residuals would mean the model frequently over-estimated the price of houses.

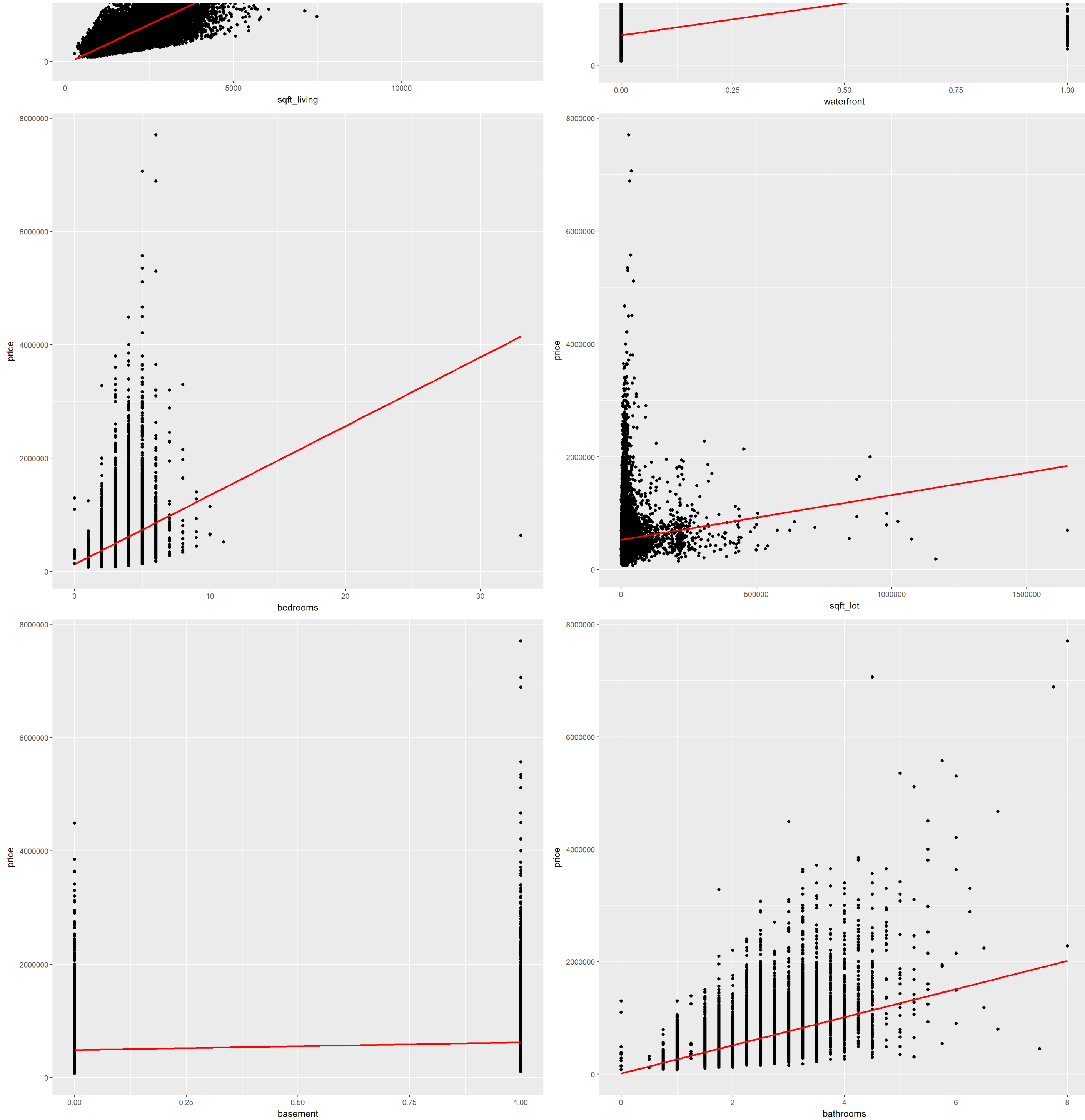
In conclusion, this assumption is violated.

- **Linearity between outcome and predictor variables:**

Scatterplots






There seems to be a considerable floor effect apparent in the sqaure footage of the lot sizes. This is because a lot of inner city properties do not have lots. Overall, linearity between outcome and predictor variables is not a concern.

- **Independant Observations:** The observations are independant as the data comes from a random sample of house prices which comprise less than 10% of the total number of house prices on the market.

In conclusion, these data do not meet the conditions required for inferring conclusions beyond this sample due to unequal variance of residuals as well as residuals not being normally distributed.