HARVARD EXTENSION SCHOOL

EXT CSCI E-106 Model Data Class Group Project

John Hur Kacper Lewtak German Paredes Lauren Briese Michael Lefkoe
Rajesh Jain

12 December 2023

Abstract

This is the location for your abstract.
It must consist of two paragraphs.

Contents

House Sales in King County, USA data to be used in the Final Project

Variable	Description
id	**Unique ID for each home sold (it is not a predictor)**
date	*Date of the home sale*
price	*Price of each home sold*
bedrooms	*Number of bedrooms*
bathrooms	*Number of bathrooms, where ".5" accounts for a bathroom with a toilet but no shower*
sqft_living	*Square footage of the apartment interior living space*
sqft_lot	*Square footage of the land space*
floors	*Number of floors*
waterfront	*A dummy variable for whether the apartment was overlooking the waterfront or not*
view	*An index from 0 to 4 of how good the view of the property was*
condition	*An index from 1 to 5 on the condition of the apartment,*
grade	*An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 has a high-quality level of construction and design.*
sqft_above	*The square footage of the interior housing space that is above ground level*
sqft_basement	*The square footage of the interior housing space that is below ground level*
yr_built	*The year the house was initially built*
yr_renovated	*The year of the house's last renovation*
zipcode	*What zipcode area the house is in*
lat	*Latitude*
long	*Longitude*
sqft_living15	*The square footage of interior housing living space for the nearest 15 neighbors*
sqft_lot15	*The square footage of the land lots of the nearest 15 neighbors*

Instructions:

0. Join a team with your fellow students with appropriate size (Four Students total)
1. Load and Review the dataset named "KC_House_Sales'csv
2. Create the train data set which contains 70% of the data and use set.seed (1023). The remaining 30% will be your test data set.
3. Investigate the data and combine the level of categorical variables if needed and drop variables as needed. For example, you can drop id, Latitude, Longitude, etc.
4. Build a regression model to predict price.
5. Create scatter plots and a correlation matrix for the train data set. Interpret the possible relationship between the response.
6. Build the best multiple linear models by using the stepwise selection method. Compare the performance of the best two linear models.
7. Make sure that model assumption(s) are checked for the final model. Apply remedy measures (transformation, etc.) that helps satisfy the assumptions.
8. Investigate unequal variances and multicollinearity. If necessary, apply remedial methods (WLS, Ridge, Elastic Net, Lasso, etc.).
9. Build an alternative model based on one of the following approaches to predict price: regression tree, NN, or SVM. Check the applicable model assumptions. Explore using a logistic regression.
10. Use the test data set to assess the model performances from above.
11. Based on the performances on both train and test data sets, determine your primary (champion) model and the other model which would be your benchmark model.
12. Create a model development document that describes the model following this template, input the name of the authors, Harvard IDs, the name of the Group, all of your code and calculations, etc..:

Due Date: December 18th, 2023 at 11:59 pm EST

Notes No typographical errors, grammar mistakes, or misspelled words, use English language All tables need to be numbered and describe their content in the body of the document All figures/graphs need to be numbered and describe their content All results must be accurate and clearly explained for a casual reviewer to fully understand their purpose and impact Submit both the RMD markdown file and PDF with the sections with appropriate explanations. A more formal document in Word can be used in place of the pdf file but must include all appropriate explanations.

Executive Summary

This section will describe the model usage, your conclusions and any regulatory and internal requirements. In a real world scneario, this section is for senior management who do not need to know the details. They need to know high level (the purpose of the model, limitations of the model and any issues).

I. Introduction (5 points)

This section needs to introduce the reader to the problem to be resolved, the purpose, and the scope of the statistical testing applied. What you are doing with your prediction? What is the purpose of the model? What methods were trained on the data, how large is the test sample, and how did you build the model?

II. Description of the data and quality (15 points)

Here you need to review your data, the statistical test applied to understand the predictors and the response and how are they correlated. Extensive graph analysis is recommended. Is the data continuous, or categorical, do any transformation needed? Do you need dummies?

```
# read in data
kc_house_sales = read.csv('KC_House_Sales.csv')

# get number of N/A values in each column
colSums(is.na(kc_house_sales))
```

```
##            id          date         price      bedrooms     bathrooms
##             0             0             0             0             0
##   sqft_living      sqft_lot        floors    waterfront          view
##             0             0             0             0             0
##     condition         grade    sqft_above sqft_basement      yr_built
##             0             0             0             0             0
##  yr_renovated       zipcode           lat          long sqft_living15
##             0             0             0             0             0
##    sqft_lot15
##             0
```

```
# get column names, data types, and first few values for each column
str(kc_house_sales)
```

```
## 'data.frame':    21613 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ price        : chr  " $221,900.00 " " $538,000.00 " " $180,000.00 " " $604,000.00 " ...
##  $ 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 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

The data to be analyzed in this report is a set of housing sales data from King County, USA in the 2014-2015 time frame. The original data set contains 21 columns with 21613 observations.

```
kc_house_sales = subset(kc_house_sales, select = -c(id, lat, long, sqft_basement))
```

We dropped the following columns prior to developing our models:
* id: This is not a predictor or relevant to the sales data beyond being an identifier for each row.
* lat: This level of detail is not necessary, and the zipcode column also approximates location.
* long: This level of detail is not necessary, and the zipcode column also approximates location.
* sqft_basement: This column is equal to sqft_living - sqft_above, so it does not provide new information and is not necessary.

```
# convert the price to a numeric column
kc_house_sales$price = as.numeric(gsub('\\$|,','',kc_house_sales$price))
```

We updated the price column to numeric as it is the response variable which we built the models to predict.

```r
# convert zip code to characters so it will create dummy variables
kc_house_sales$zipcode = as.character(kc_house_sales$zipcode)

# convert date to just years and months, there are 13 unique values
kc_house_sales$date = substr(kc_house_sales$date, start = 1, stop = 6)
```

We treated the following columns as categorical based on their data types and converted the data and zip code columns accordingly:
* date: Although they appear to be numeric, date should be treated as categorical data. Date was grouped by month to avoid having hundred of individual variables.
* zipcode: Although they appears to be numbers, zip codes are not numeric and instead should be treated as categorical data.
* waterfront: As mentioned in the description provided with this data set, this is a dummy variable to indicate whether the property overlooks the water.

```r
# get correlation of numeric data, rounded for ease of review)
kc_house_sales_numeric = subset(kc_house_sales, select = -c(date, waterfront, zipcode))
round(cor(kc_house_sales_numeric),3)
```

```
##               price bedrooms bathrooms sqft_living sqft_lot floors   view
## price         1.000    0.308     0.525       0.702    0.090  0.257  0.397
## bedrooms      0.308    1.000     0.516       0.577    0.032  0.175  0.080
## bathrooms     0.525    0.516     1.000       0.755    0.088  0.501  0.188
## sqft_living   0.702    0.577     0.755       1.000    0.173  0.354  0.285
## sqft_lot      0.090    0.032     0.088       0.173    1.000 -0.005  0.075
## floors        0.257    0.175     0.501       0.354   -0.005  1.000  0.029
## view          0.397    0.080     0.188       0.285    0.075  0.029  1.000
## condition     0.036    0.028    -0.125      -0.059   -0.009 -0.264  0.046
## grade         0.667    0.357     0.665       0.763    0.114  0.458  0.251
## sqft_above    0.606    0.478     0.685       0.877    0.184  0.524  0.168
## yr_built      0.054    0.154     0.506       0.318    0.053  0.489 -0.053
## yr_renovated  0.126    0.019     0.051       0.055    0.008  0.006  0.104
## sqft_living15 0.585    0.392     0.569       0.756    0.145  0.280  0.280
## sqft_lot15    0.082    0.029     0.087       0.183    0.719 -0.011  0.073
##               condition  grade sqft_above yr_built yr_renovated sqft_living15
## price             0.036  0.667      0.606    0.054        0.126         0.585
## bedrooms          0.028  0.357      0.478    0.154        0.019         0.392
## bathrooms        -0.125  0.665      0.685    0.506        0.051         0.569
## sqft_living      -0.059  0.763      0.877    0.318        0.055         0.756
## sqft_lot         -0.009  0.114      0.184    0.053        0.008         0.145
## floors           -0.264  0.458      0.524    0.489        0.006         0.280
## view              0.046  0.251      0.168   -0.053        0.104         0.280
## condition         1.000 -0.145     -0.158   -0.361       -0.061        -0.093
## grade            -0.145  1.000      0.756    0.447        0.014         0.713
## sqft_above       -0.158  0.756      1.000    0.424        0.023         0.732
## yr_built         -0.361  0.447      0.424    1.000       -0.225         0.326
## yr_renovated     -0.061  0.014      0.023   -0.225        1.000        -0.003
## sqft_living15    -0.093  0.713      0.732    0.326       -0.003         1.000
## sqft_lot15       -0.003  0.119      0.194    0.071        0.008         0.183
##               sqft_lot15
## price              0.082
## bedrooms           0.029
## bathrooms          0.087
## sqft_living        0.183
## sqft_lot           0.719
## floors            -0.011
## view               0.073
## condition         -0.003
## grade              0.119
## sqft_above         0.194
## yr_built           0.071
```

```
## yr_renovated       0.008
## sqft_living15      0.183
## sqft_lot15         1.000
```

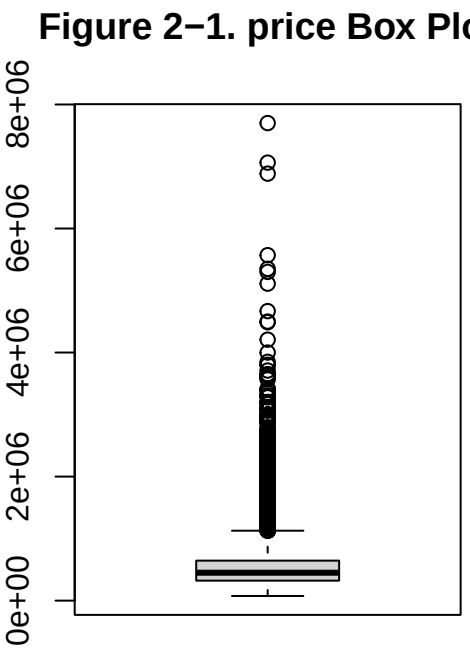
Looking at the correlation matrix above for the numeric columns in the data set, price appears to be most correlated with the fields sqft_living (0.702), grade (0.667), and sqft_above (0.606). Note that the categorical variables date, waterfront, and zipcode are not included in the correlation matrix.

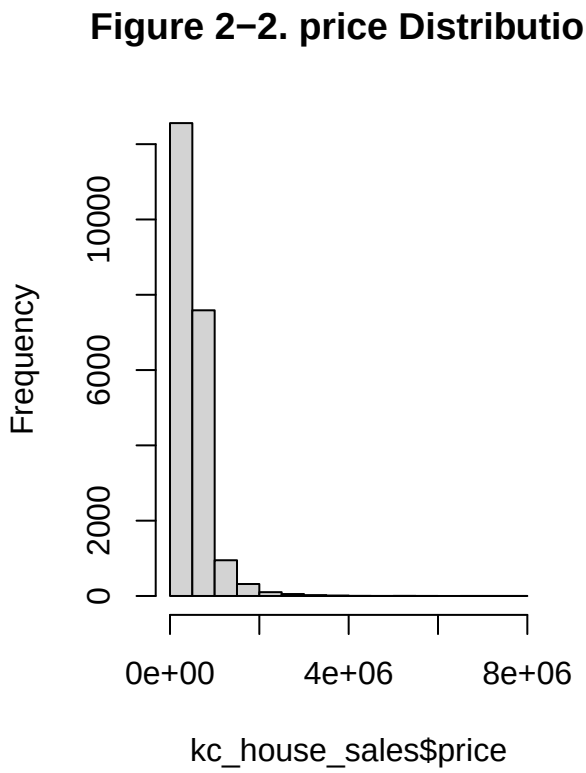
The sqft_living and sqft_above variables have a high correlation; it may be worth including only one of these in the model. For houses without basements, the sqft_living and sqft_above values should be equal.

The following sections describe the response variable (price) and the predictor variables to be analyzed for inclusion or exclusion from the data models.

Price

```
# price box plot and histogram
par(mfrow=c(1,2))
boxplot(kc_house_sales$price, main = "Figure 2-1. price Box Plot")
hist(kc_house_sales$price, main = "Figure 2-2. price Distribution")
```






```
mean(kc_house_sales$price)
```

```
## [1] 540088.1
```

```
median(kc_house_sales$price)
```

```
## [1] 450000
```

We updated the price column to a numeric column since it could not be analyzed with the character type. The median sales price for the observations was $450,000, and the average was $540,088.10. Per Figure 2-1, it appears as though there are many outliers in the data on the higher end of the price range. Figure 2-2 confirms this assumption, showing that the distribution of sales prices has a much higher frequency at the lower end of the range.

Bedrooms

```
# bedrooms box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
```

```
boxplot(kc_house_sales$bedrooms, main = "Figure 2-3. bedrooms Box Plot")
hist(kc_house_sales$bedrooms, main = "Figure 2-4. bedrooms  Distribution")
```

Figure 2−3. bedrooms Box Plot

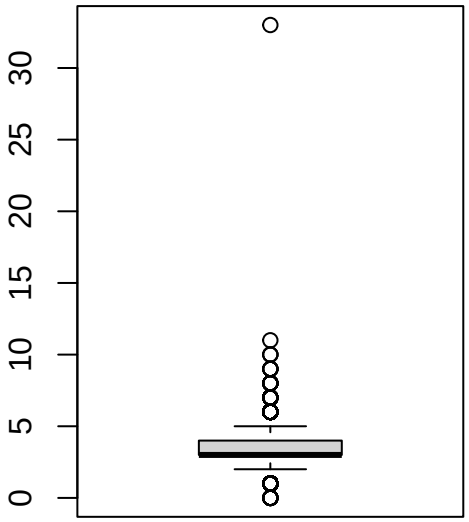
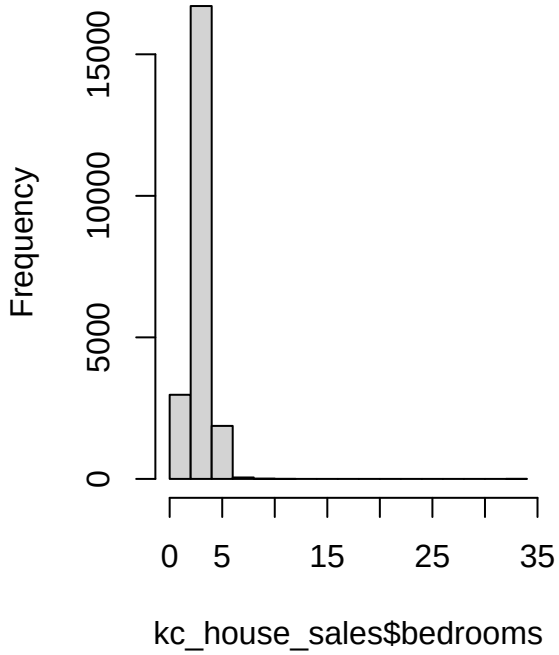
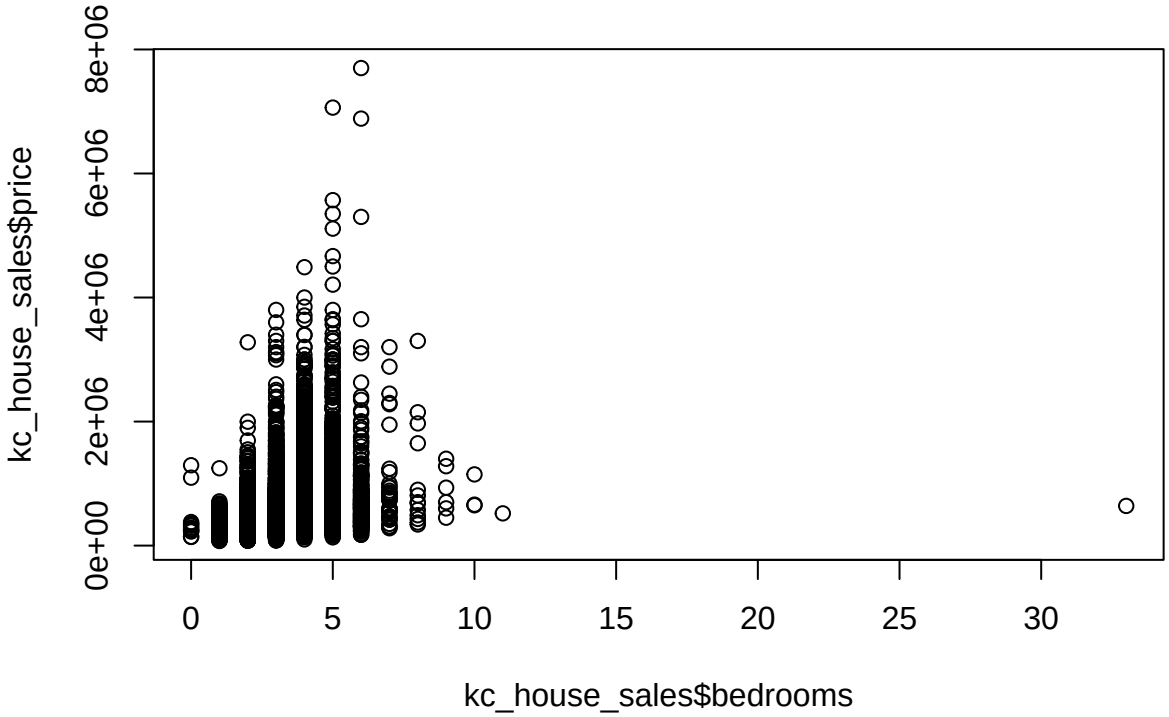


Figure 2−4. bedrooms Distributi




```
par(mfrow=c(1,1))
plot(kc_house_sales$bedrooms, kc_house_sales$price,  main = "Figure 2-5. bedrooms Scatter Plot")
```

Figure 2−5. bedrooms Scatter Plot



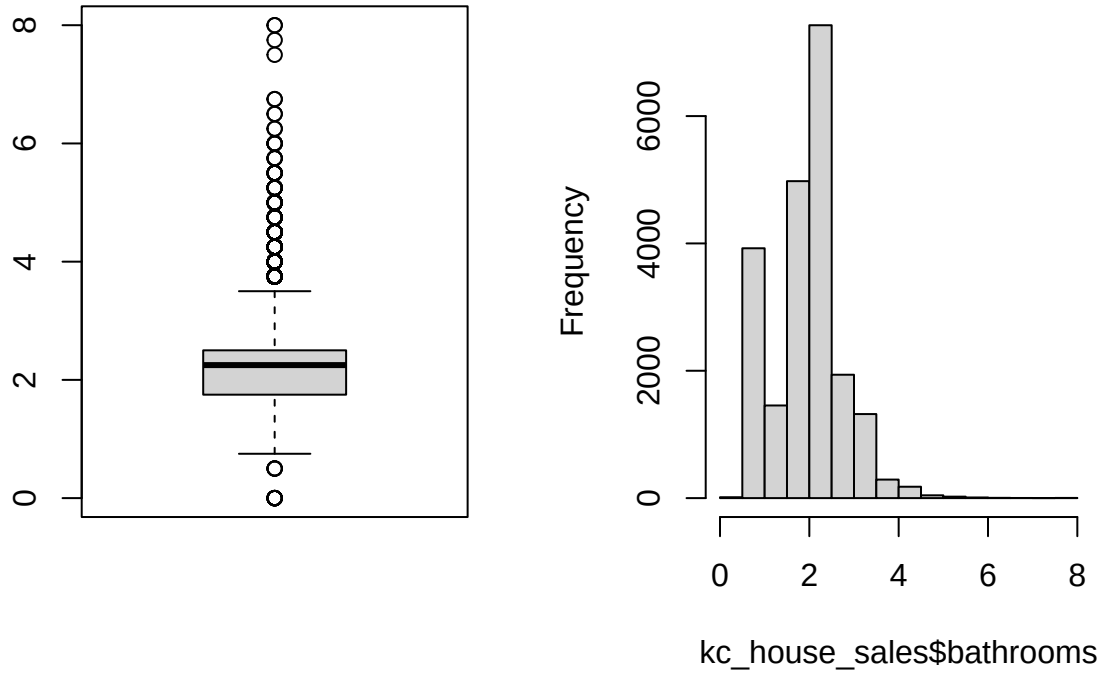
Looking at the number of bedrooms and the box plot in Figure 2-3, there are outliers on the lower and upper ends of the range of bedroom values, with one observation having more than 30 bedrooms. Per Figure 2-4, it appears as though the vast majority of houses have between two and four bedrooms. It would be interesting to know the state of the "houses" with 0 bedrooms–are these studio apartments, cabins, plots of bare land, houses that have been gutted or otherwise have no interiors and thus no bedrooms, commercial properties accidentally included in the data set, or is there another explanation?

Figure 2-5 displays what appears to be a curvilinear relationship between the number of bedrooms and the house price, increasing until reaching five bedrooms and then falling again.

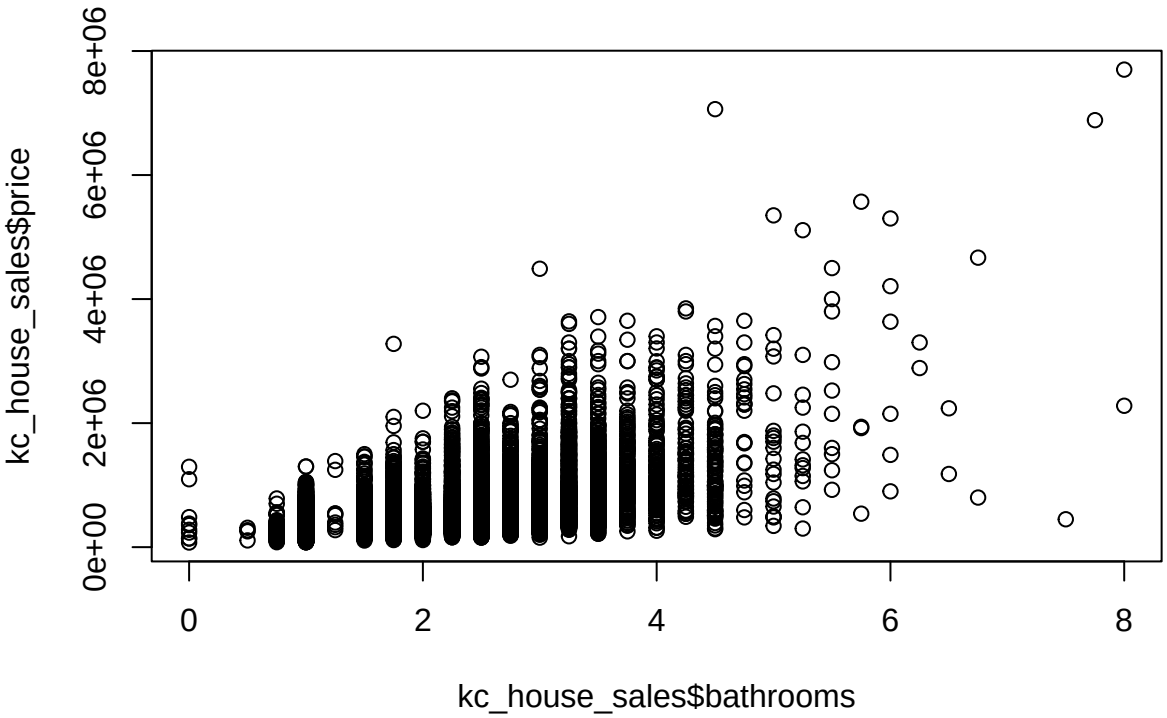
Bathrooms

```
# bathrooms box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$bathrooms, main = "Figure 2-6. bathrooms Box Plot")
hist(kc_house_sales$bathrooms, main = "Figure 2-7. bathrooms Distribution")
```

Figure 2−6. bathrooms Box Plot **Figure 2−7. bathrooms Distributi**




```
par(mfrow=c(1,1))
plot(kc_house_sales$bathrooms, kc_house_sales$price,  main = "Figure 2-8. bathrooms Scatter Plot")
```



The bathroom predictor variable presented confusion upon first glance. Assuming the data is for residential real estate in the King County district, it seems odd that it would be possible for a house to have less than one full bathroom, yet there were multiple observations with less than 1.0 bathroom (and some with 0 bathrooms or fractions of a bathroom ending in .25). Upon researching the definitions used for bathroom terminology in the real estate industry [2], it appears as though each major component of a bathroom–toilet, sink, shower, and tub–counts as 0.25. It might have been more interesting for the bathroom data to be split into two columns: full_bathrooms and partial_bathrooms since, for example, if a house has 2 bathrooms according to the data set, it is unclear whether that means two full bathrooms, one full bathroom and two half bathrooms (toilet and sink), or another configuration. As for the houses with less than 1.0 bathroom, it would be interesting to know the reason–is the house unfinished? Is it a tiny house with just a composting toilet in the bathroom?

Overall, per Figure 2-6 and Figure 2-7, most houses have around 2 bathrooms, with outliers at the lower and upper end of the ranges. Per Figure 2-8, it appears as though there is a linear relationship between number of bathrooms and price of the house.

Square Footage - Lot

```
# sqft_lot box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$sqft_lot, main = "Figure 2-9. sqft_lot Box Plot")
hist(kc_house_sales$sqft_lot, main = "Figure 2-10. sqft_lot Distribution")
```

Figure 2−9. sqft_lot Box Plot

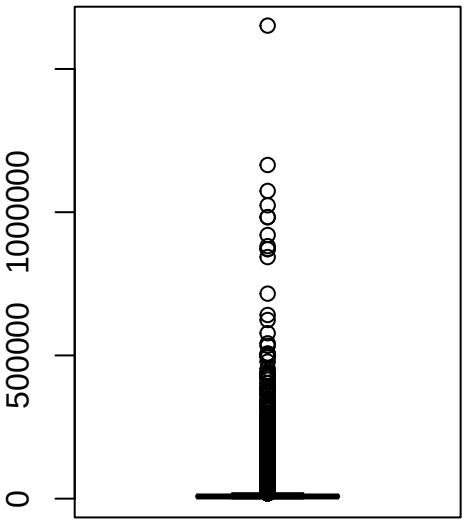
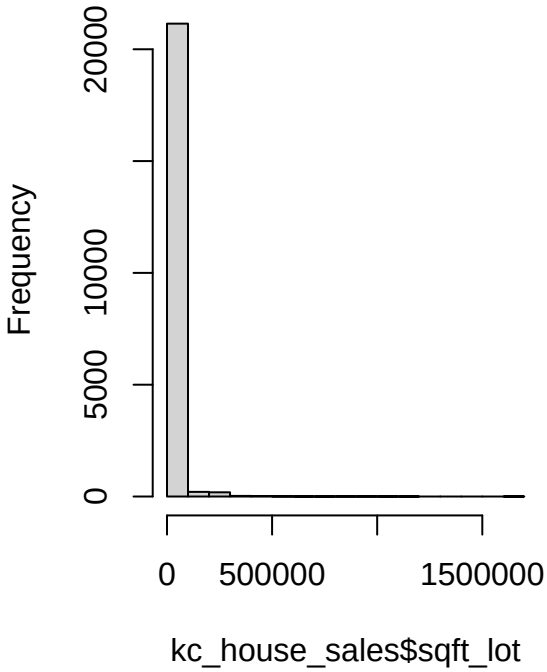
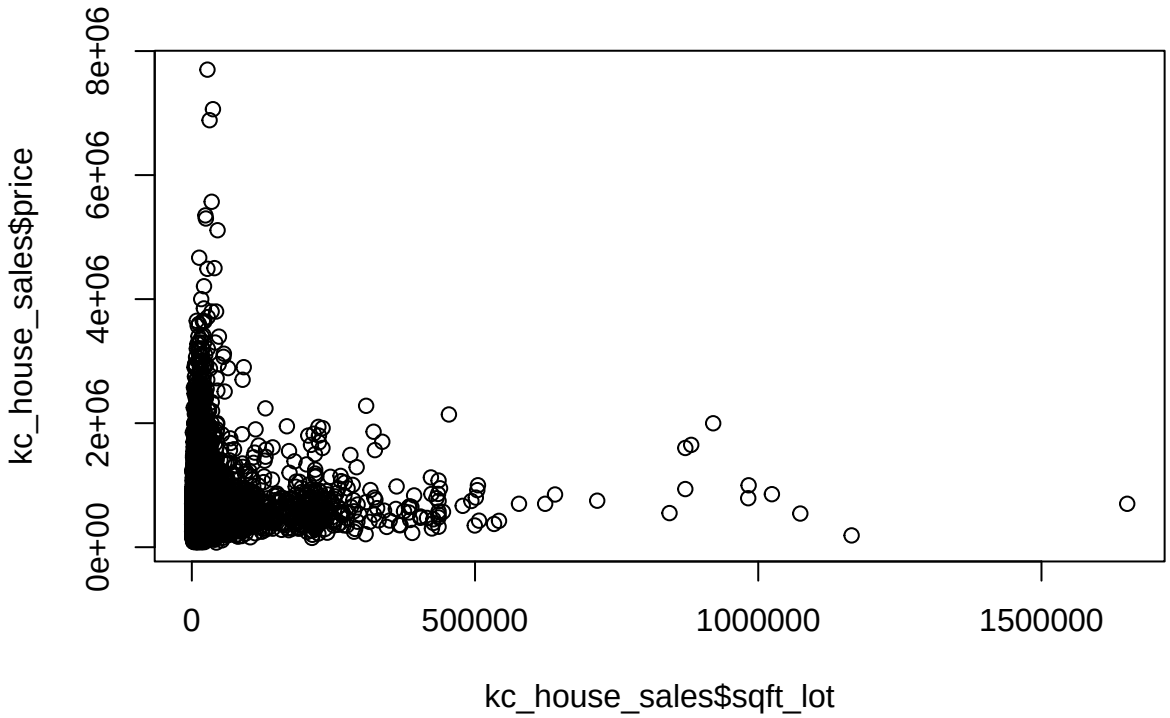


Figure 2−10. sqft_lot Distributio




```
par(mfrow=c(1,1))
plot(kc_house_sales$sqft_lot, kc_house_sales$price,  main = "Figure 2-11. sqft_lot Scatter Plot")
```

Figure 2−11. sqft_lot Scatter Plot



Per Figure 2-9 and Figure 2-10 showing the box plot and distribution of the sqft_lot it appears as though the lot size is very concentrated around the lower end of the spectrum, with a few extreme outliers. According to the scatter plot in Figure 2-11, it looks like there is somewhat of a negative linear relationship between lot size and house price.

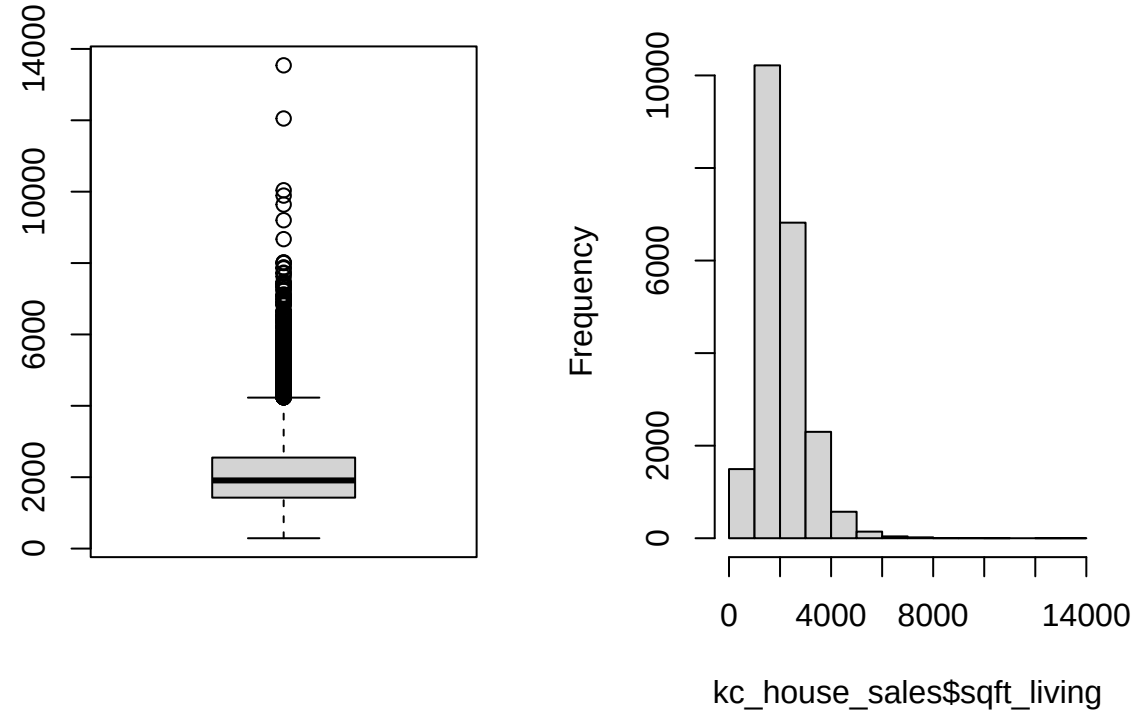
Looking at some of the individual data points, it seems as though some of the lot sizes might be impossible considering the square footage of the living area. For example, the row with ID 9828702895 has a sqft_lot size of 520, a sqft_living of 2420, no basement (sqft_above is also 2420), and only 1.5 floors. With only 1.5 floors and no basement, it does not seem possible to have a square footage of 2420 sqft on a lot size of 520 sqft, considering that the lot size typically includes the house's footprint [3]. Unfortunately, without further clarification, it is not possible to exclude data points based on this rationale. The "square

footage of the land space" definition provided with the data set could mean the land excluding the house's footprint.

Square Footage - Living

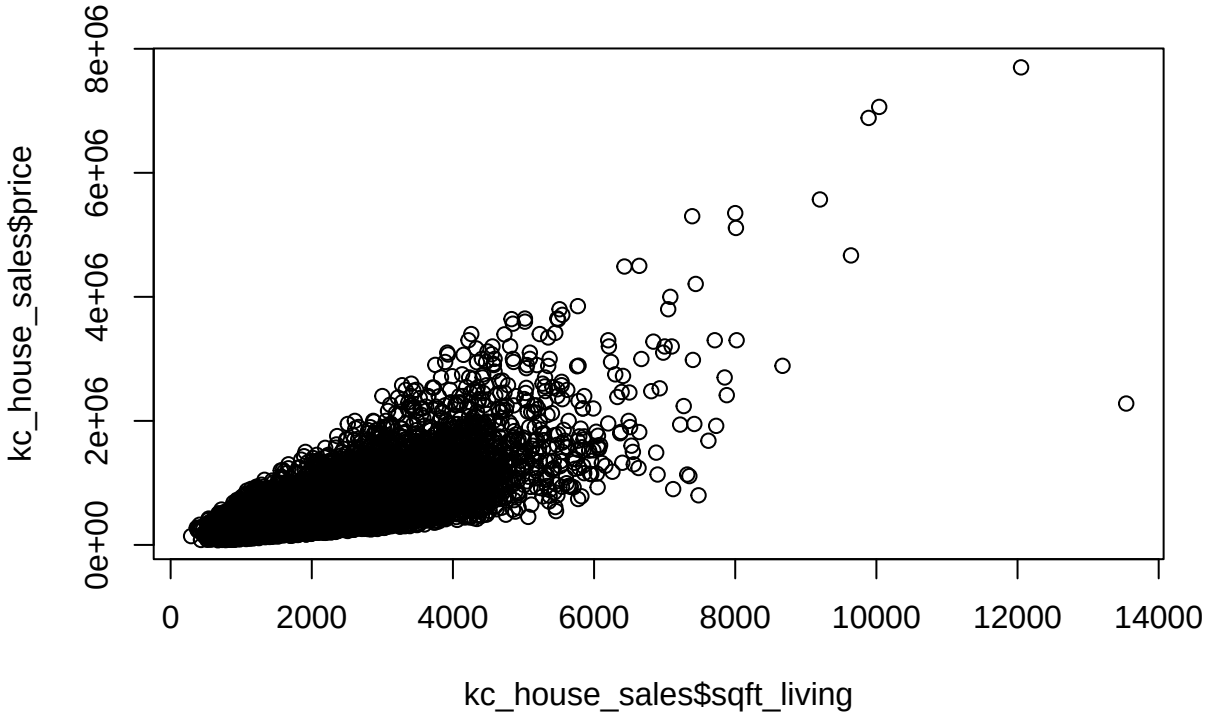
```
# sqft_living box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$sqft_living, main = "Figure 2-12. sqft_living Box Plot")
hist(kc_house_sales$sqft_living, main = "Figure 2-13. Distribution of sqft_living")
```

Figure 2−12. sqft_living Box Plo **Figure 2−13. Distribution of sqft_liv**




```
par(mfrow=c(1,1))
plot(kc_house_sales$sqft_living, kc_house_sales$price,  main = "Figure 2-14. sqft_living Scatter Plot")
```

Figure 2−14. sqft_living Scatter Plot



Per Figure 2-12 and 2-13, the median square footage of living space appears to be around 2000 sqft, with outliers present at the higher end of the range.

Per Figure 2-14, there appears to be a linear relationship between square feet of living space and price, with price increasing as the square footage increases.

Square Footage - Above Ground

```
# sqft_above box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$sqft_above, main = "Figure 2-15. sqft_above Box Plot")
hist(kc_house_sales$sqft_above, main = "Figure 2-16. sqft_above Distribution")
```

Figure 2−15. sqft_above Box Plc

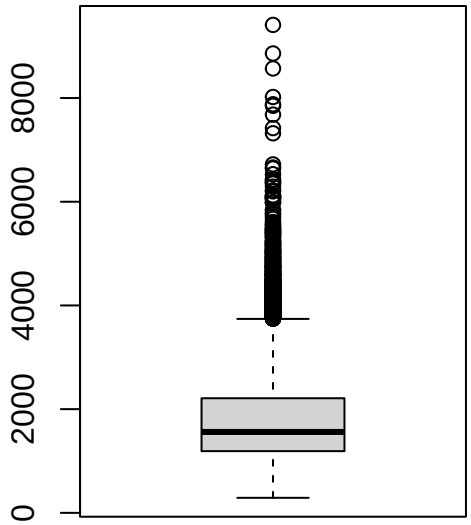
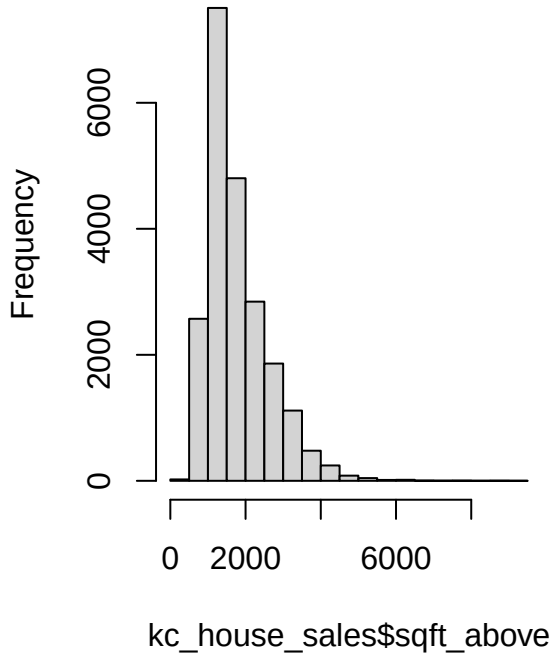
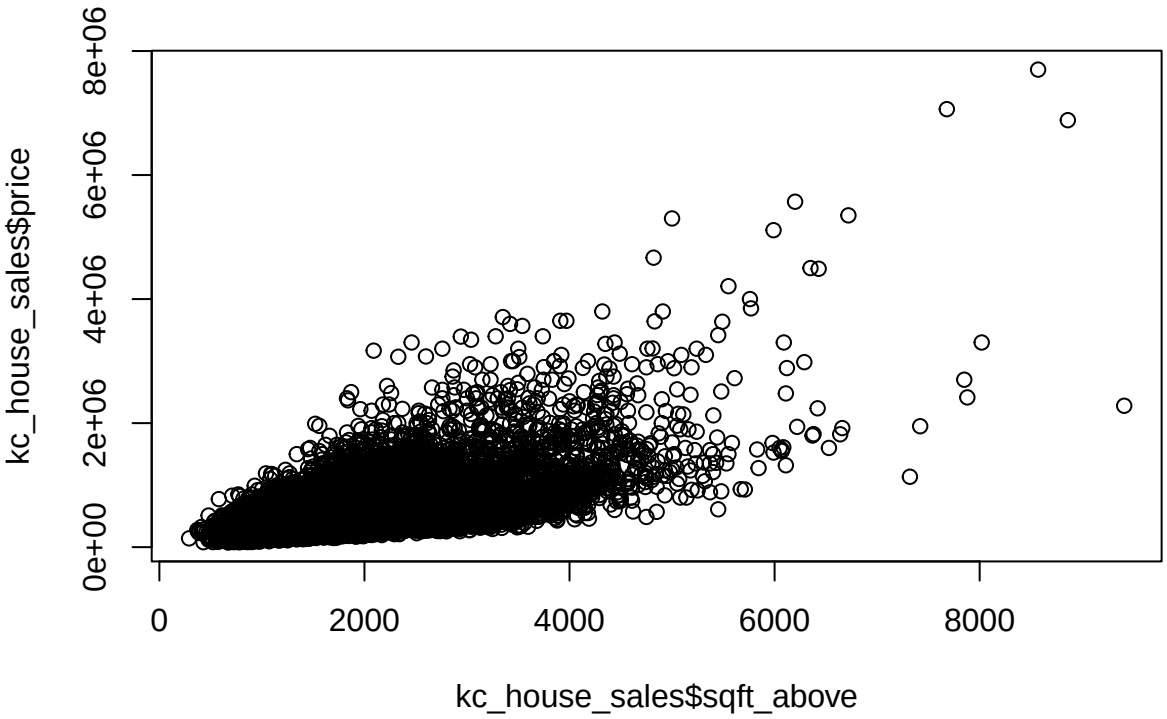


Figure 2−16. sqft_above Distributi




```
par(mfrow=c(1,1))
plot(kc_house_sales$sqft_above, kc_house_sales$price,  main = "Figure 2-17. sqft_above Scatter Plot")
```



Per Figure 2-15 and 2-16, the median square footage above ground appears to be slightly less than 2000 sqft, with outliers present at the higher end of the range.

Per Figure 2-17, there appears to be a linear relationship between square feet of living space and price, with price increasing as the square footage increases.

Floors

```
# floors box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$floors, main = "Figure 2-18. floors Box Plot")
hist(kc_house_sales$floors, main = "Figure 2-19. floors Distribution")
```

Figure 2−18. floors Box Plot

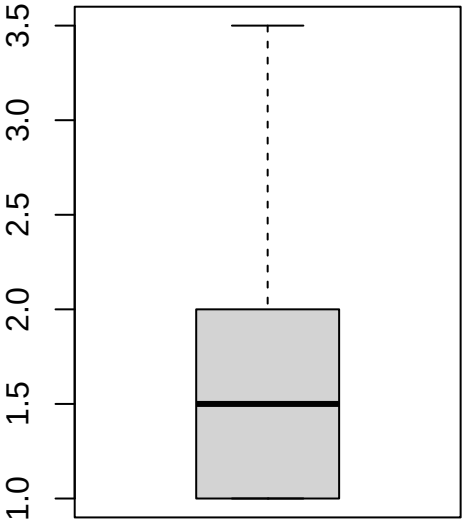
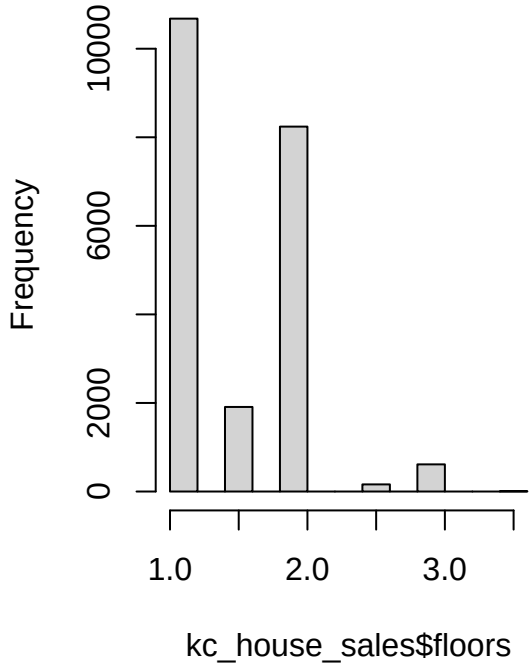
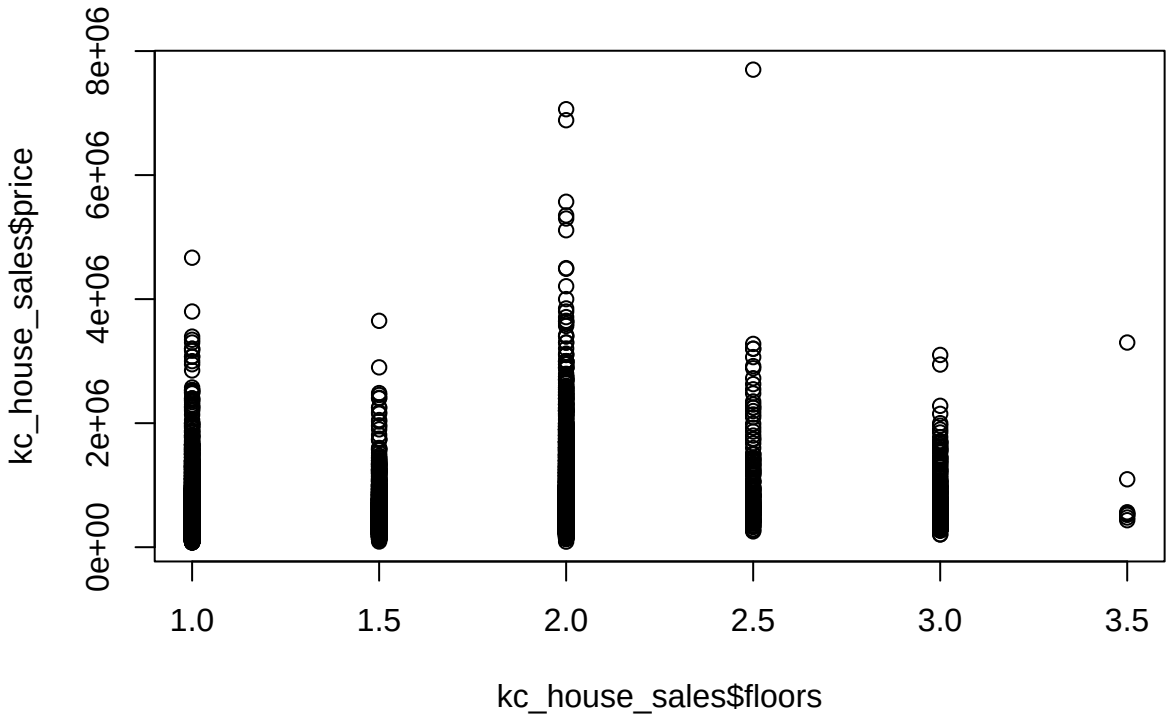


Figure 2−19. floors Distribution




```
par(mfrow=c(1,1))
plot(kc_house_sales$floors, kc_house_sales$price,  main = "Figure 2-20. floors Scatter Plot")
```

Figure 2−20. floors Scatter Plot

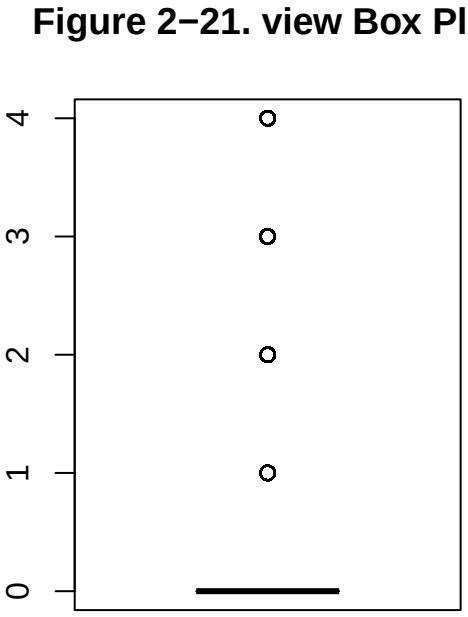


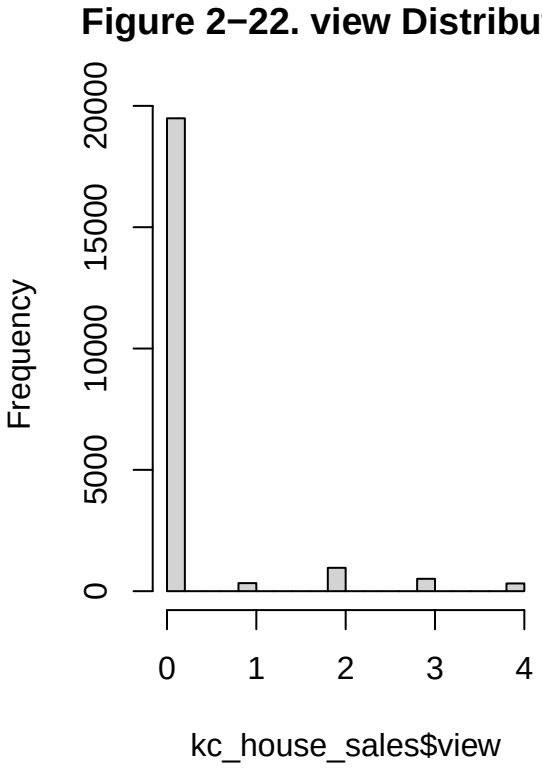
Per Figures 2-18, the median house has 1.5 floors, and there are no outliers in the data set. Although the median is 1.5 floors, looking at the distribution in Figure 2-19, most houses have either 1 floor or 2 floors.

Looking at Figure 2-20, there appears to be a nonlinear relationship between nnumber of floors and price. Houses with 1 floor or 2 floors appear to have a wider price range with higher maximum prices than houses with a fractional floor or those with 3 foors.

View

```
# view box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$view, main = "Figure 2-21. view Box Plot")
hist(kc_house_sales$view, main = "Figure 2-22. view Distribution")
```






```
par(mfrow=c(1,1))
plot(kc_house_sales$view, kc_house_sales$price,  main = "Figure 2-23. view Scatter Plot")
```

Looking at Figure 2-21 and Figure 2-22, most houses do not have a view; according to the box plot, any house with a view appears to be an outlier. Per Figure 2-23, the price does not seem to increase or decrease depending on the house's view.

Condition

```
# condition box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$condition, main = "Figure 2-24. condition Box Plot")
hist(kc_house_sales$condition, main = "Figure 2-25. condition Distribution")
```

Figure 2−24. condition Box Plot

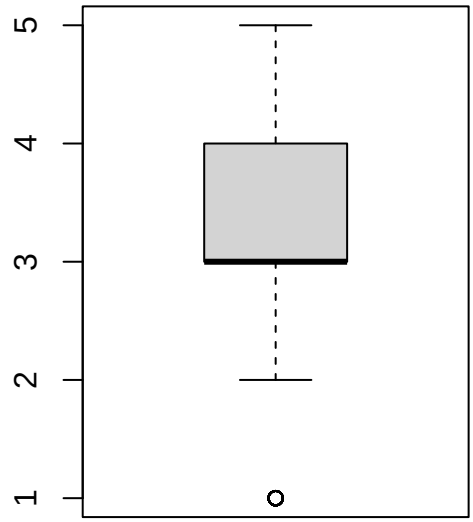
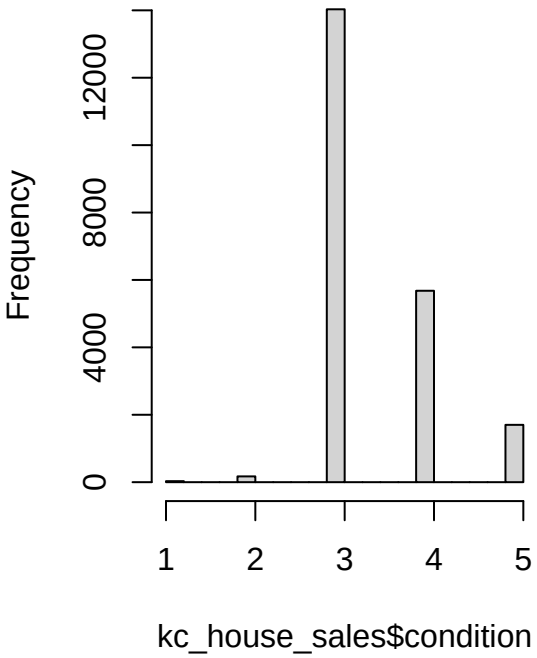
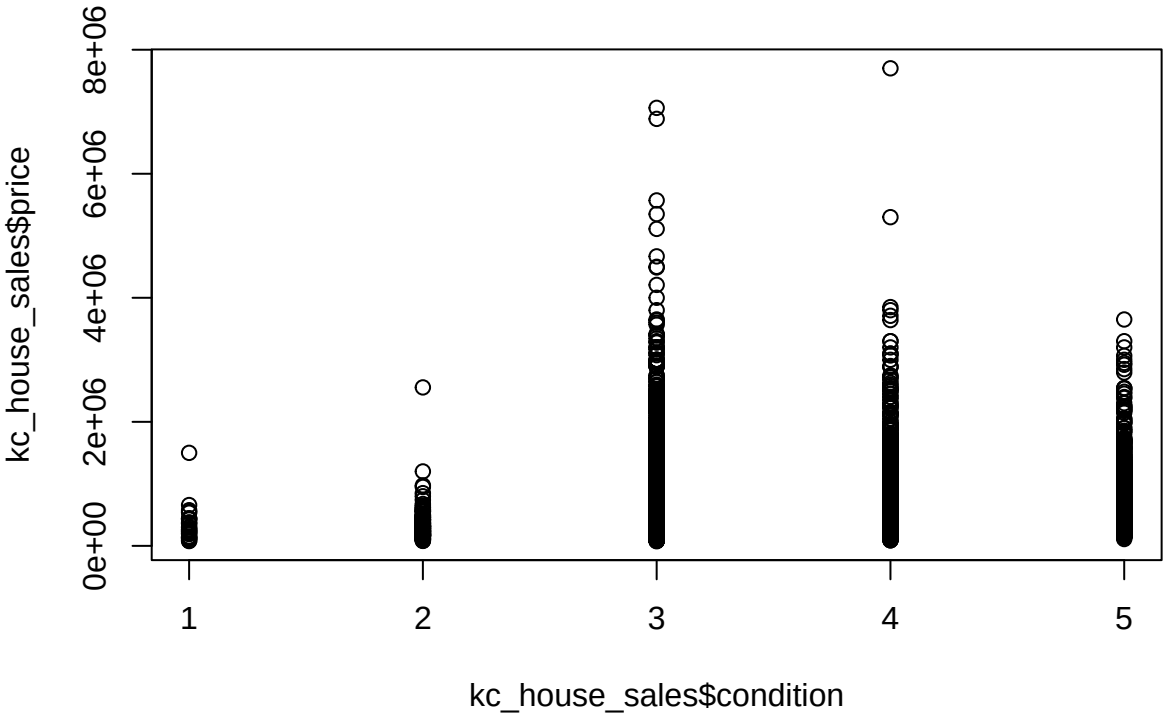


Figure 2−25. condition Distributic




```
par(mfrow=c(1,1))
plot(kc_house_sales$condition, kc_house_sales$price,  main = "Figure 2-26. condition Scatter Plot")
```

Figure 2−26. condition Scatter Plot



The box plot in Figure 2-24 indicates that the median house has a condition of 3, which is average, and houses with a condition of 1 are outliers. Figure 2-25 with the distribution again shows that the vast majority of houses have a condition of

3. Interestingly, in the scatter plot in Figure 2-26, it appears as though if the condition is at least 3, there is not much of an impact on housing price, but below 3 and the price decreases.

Grade

```
# grade box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$grade, main = "Figure 2-27. grade Box Plot")
hist(kc_house_sales$grade, main = "Figure 2-28. grade Distribution")
```

Figure 2−27. grade Box Plot

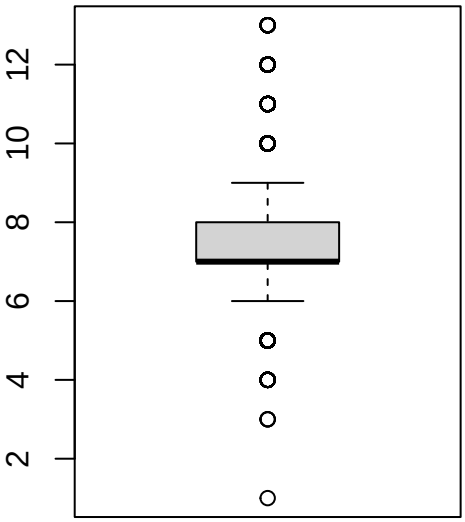
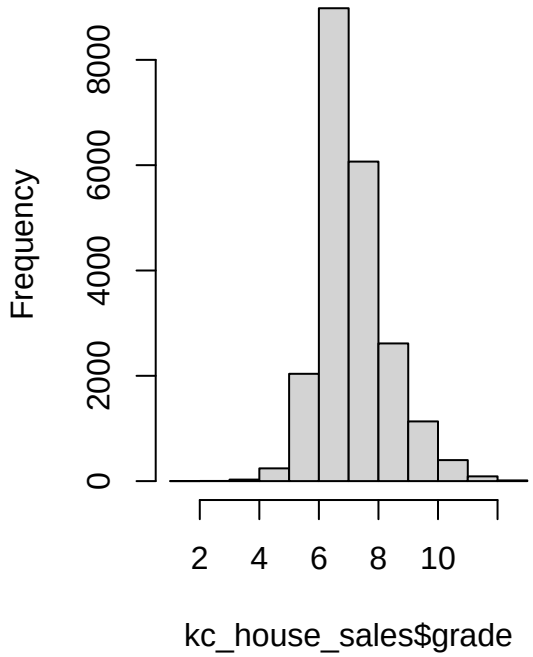
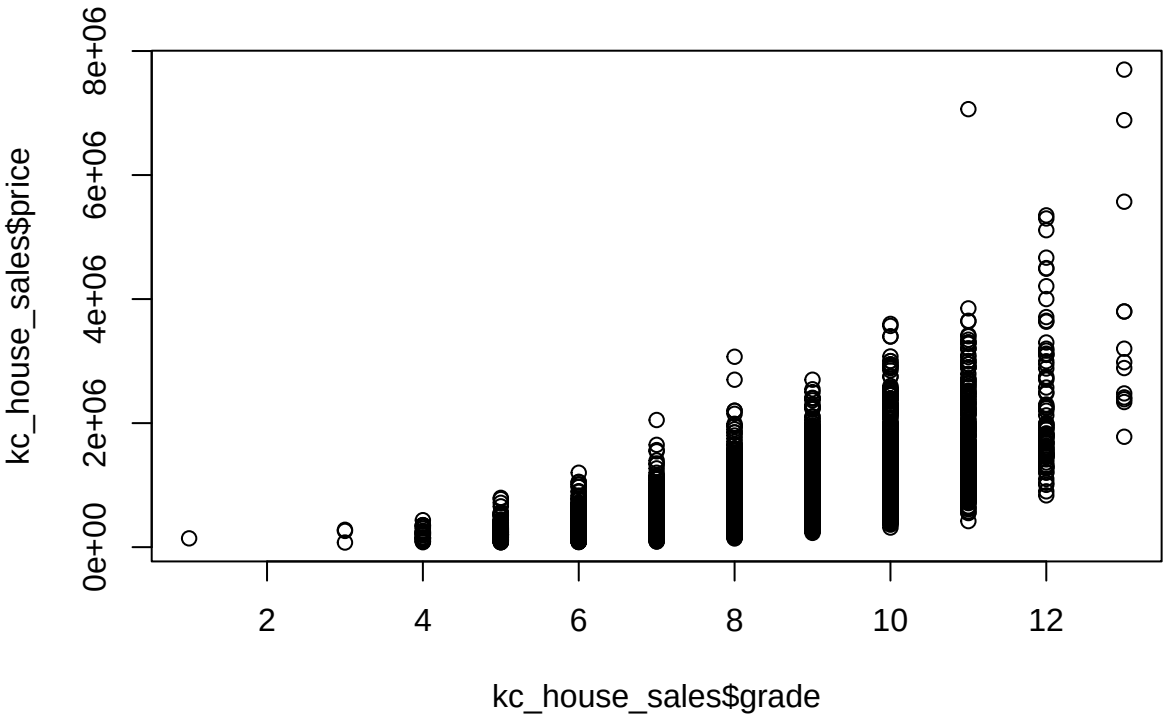


Figure 2−28. grade Distribution




```
par(mfrow=c(1,1))
plot(kc_house_sales$grade, kc_house_sales$price,  main = "Figure 2-29. grade Scatter Plot")
```

Figure 2−29. grade Scatter Plot

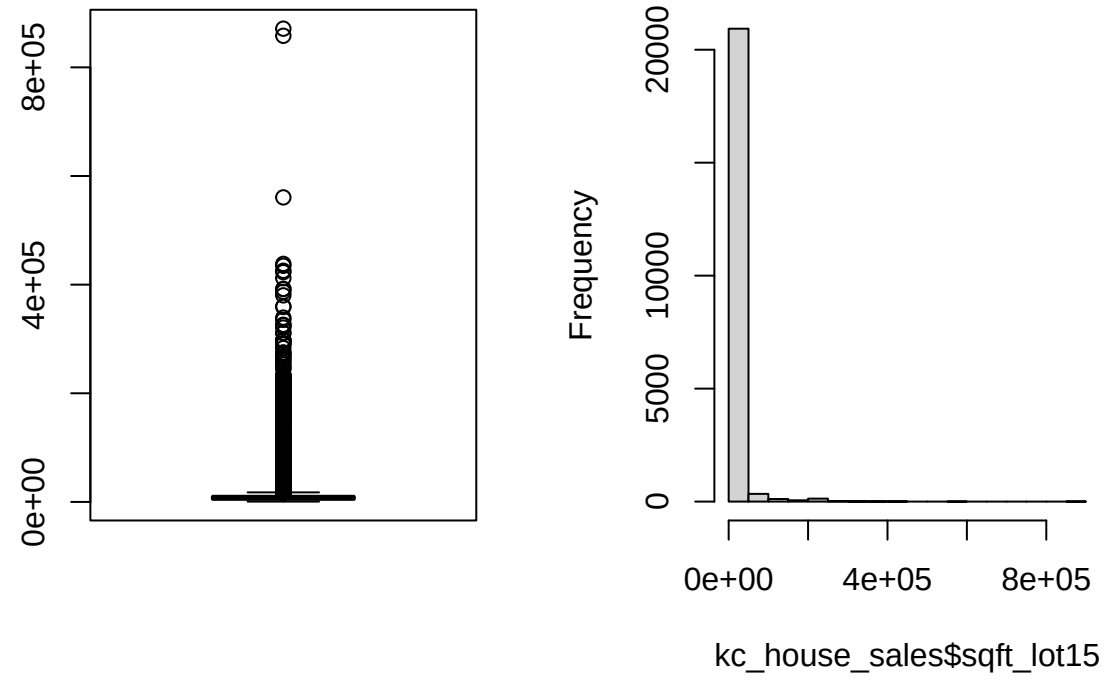


According to the box plot and histogram in Figure 2-27 and 2-28, the median grade appears to be 7, which is an "average" quality, and outliers are any grade outside the range of 6 to 9. The scatter plot in Figure 2-29 indicates that grade appears to have a linear relationship with prices, with the price increasing as the grade of the house increases.

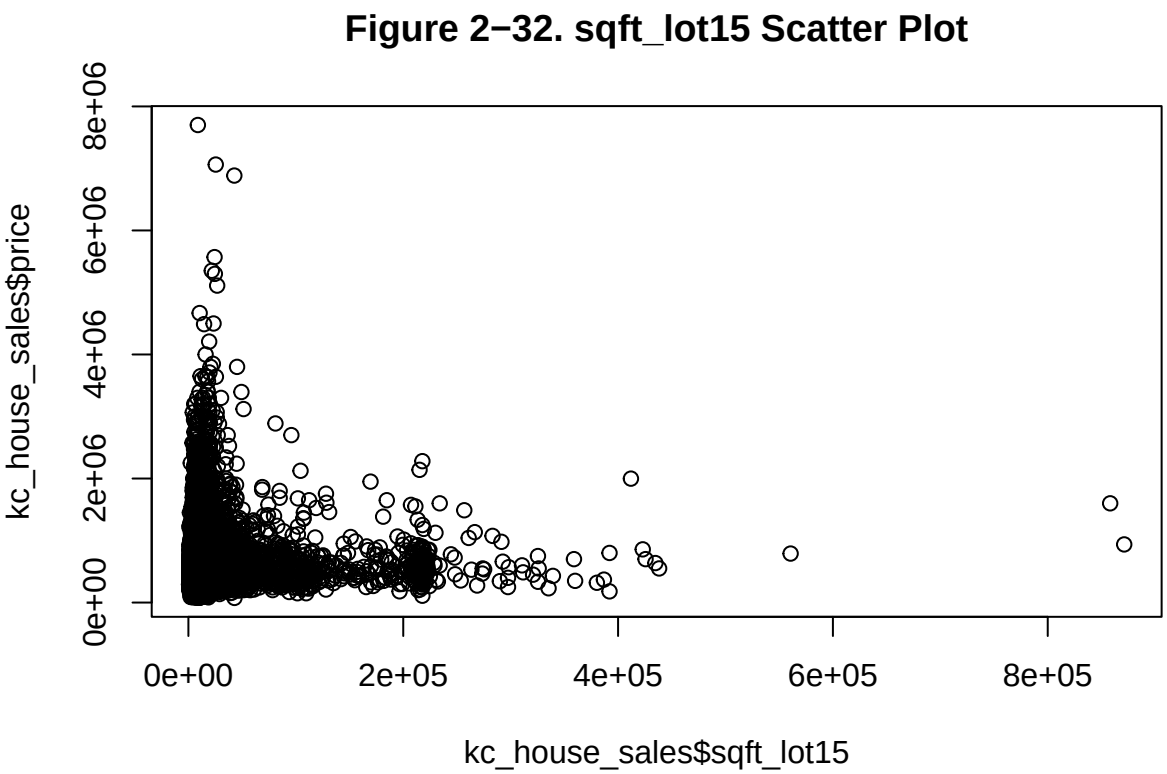
Neighbor Square Footage - Lot

```
# sqft_lot15 box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$sqft_lot15, main = "Figure 2-30. sqft_lot15 Box Plot")
hist(kc_house_sales$sqft_lot15, main = "Figure 2-31. sqft_lot15 Distribution")
```

Figure 2−30. sqft_lot15 Box Plo **Figure 2−31. sqft_lot15 Distributi**



```
par(mfrow=c(1,1))
plot(kc_house_sales$sqft_lot15, kc_house_sales$price,  main = "Figure 2-32. sqft_lot15 Scatter Plot")
```

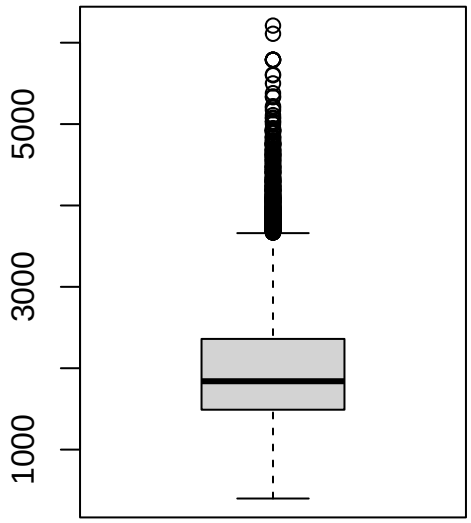


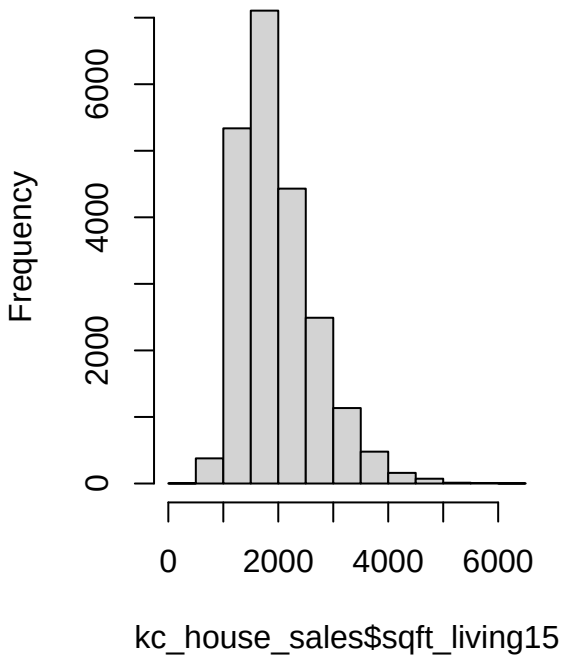
According to the box plot and histogram in Figure 2-30 and 2-31 TBD. The scatter plot in Figure 2-32 appears to show somewhat of a negative linear relationship between price and the square footage of lot size of the nearest 15 neighbors. Perhaps houses closer to the city center, which likely have smaller lot sizes, are worth more.

Neighbor Square Footage - Living

```
# sqft_living15 box plot, histogram, scatter plot vs. price
par(mfrow=c(1,2))
boxplot(kc_house_sales$sqft_living15, main = "Figure 2-33. sqft_living15 Box Plot")
hist(kc_house_sales$sqft_living15, main = "Figure 2-34. sqft_living15 Distribution")
```

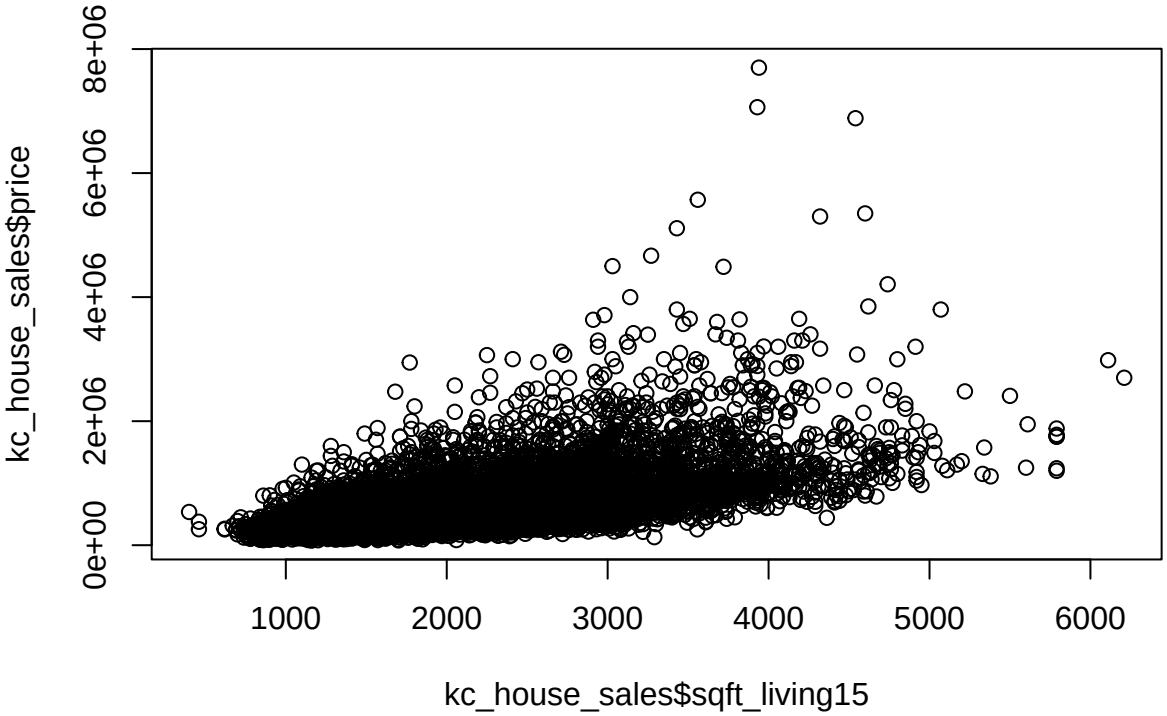
Figure 2−33. sqft_living15 Box Pl Figure 2−34. sqft_living15 Distribut






```
par(mfrow=c(1,1))
plot(kc_house_sales$sqft_living15, kc_house_sales$price,  main = "Figure 2-35. sqft_living15 Scatter Plot")
```

Figure 2−35. sqft_living15 Scatter Plot



According to the box plot and histogram in Figure 2-33 and 2-34 TBD. The scatter plot in Figure 2-35 shows a linear relationship between the square footage of living space of the 15 nearest neighbors and price.

III. Model Development Process (15 points)

Build a regression model to predict price. And of course, create the train data set which contains 70% of the data and use set.seed (1023). The remaining 30% will be your test data set. Investigate the data and combine the level of categorical variables if needed and drop variables. For example, you can drop id, Latitude, Longitude, etc.

IV. Model Performance Testing (15 points)

Use the test data set to assess the model performances. Here, build the best multiple linear models by using the stepwise both ways selection method. Compare the performance of the best two linear models. Make sure that model assumption(s) are checked for the final linear model. Apply remedy measures (transformation, etc.) that helps satisfy the assumptions. In particular you must deeply investigate unequal variances and multicollinearity. If necessary, apply remedial methods (WLS, Ridge, Elastic Net, Lasso, etc.).

V. Challenger Models (15 points)

Build an alternative model based on one of the following approaches to predict price: regression tree, NN, or SVM. Explore using a logistic regression. Check the applicable model assumptions. Apply in-sample and out-of-sample testing, backtesting and review the comparative goodness of fit of the candidate models. Describe step by step your procedure to get to the best model and why you believe it is fit for purpose.

```
############
#Data Prep
############
#Read data
sales_data <- read.csv("/cloud/project/Class Group Project/KC_House_Sales.csv")

#Remove columns that are not needed
sales_data <- subset(sales_data, select = -c(id, date, lat, long, sqft_basement))

#Clean up price column
sales_data$price = gsub("[$,]", "", sales_data$price)
sales_data$price = as.numeric(sales_data$price)

#Change ZipCode to factor
sales_data$zipcode = as.factor(sales_data$zipcode)

#Create Train and Test dfs
set.seed(1023)
n <- dim(sales_data)[1]
IND <- sample(c(1:n),n*0.7)
train_data <- sales_data[IND,]
test_data <- sales_data[-IND,]
```

```
#Create Regression Tree
library(rpart)
reg_tree <- rpart(price ~ ., data = train_data, cp=0.001) #using cp of 0.001 improved the model.
reg_tree
```

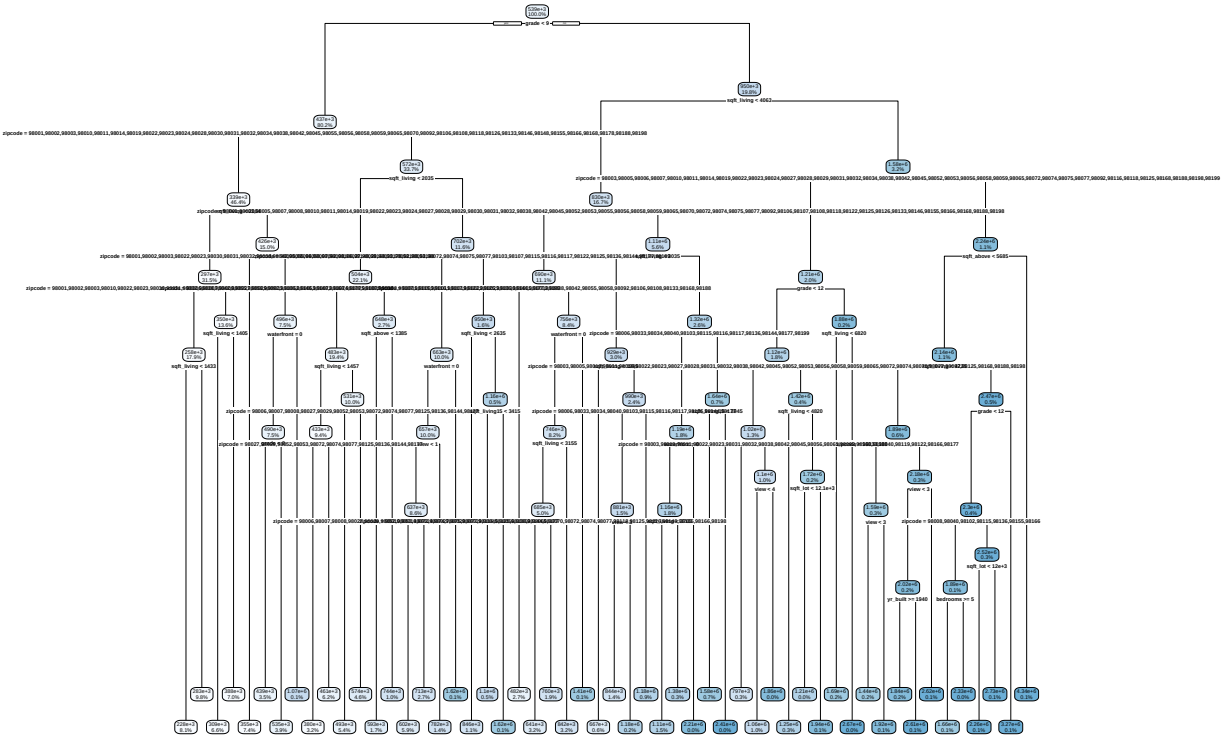
```
## n= 15129
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##   1) root 15129 1.978906e+15  538713.5
##     2) grade< 8.5 12131 4.601037e+14  437022.5
##       4) zipcode=98001,98002,98003,98010,98011,98014,98019,98022,98023,98024,98028,98030,98031,98032,98034,
##         8) sqft_living< 2004 4760 3.973822e+13  297358.7
##          16) zipcode=98001,98002,98003,98010,98022,98023,98030,98031,98032,98038,98042,98055,98058,98092,98
##            32) sqft_living< 1432.5 1232 4.825540e+12  227613.9 *
##            33) sqft_living>=1432.5 1478 5.366830e+12  282682.3 *
##          17) zipcode=98011,98014,98019,98024,98028,98034,98045,98056,98059,98065,98070,98108,98118,98126,98
##            34) sqft_living< 1405 991 5.988624e+12  308826.8 *
##            35) sqft_living>=1405 1059 8.367327e+12  388248.5 *
##         9) sqft_living>=2004 2265 4.287936e+13  425881.6
##          18) zipcode=98001,98002,98003,98022,98023,98030,98031,98032,98038,98042,98055,98058,98092,98106,98
##          19) zipcode=98010,98011,98014,98019,98024,98028,98034,98045,98056,98059,98065,98070,98108,98118,98
##            38) waterfront< 0.5 1128 1.537720e+13  489581.3
##              76) grade< 7.5 533 4.154025e+12  438864.0 *
##              77) grade>=7.5 595 8.624019e+12  535013.8 *
##            39) waterfront>=0.5 13 2.908849e+12 1069881.0 *
##       5) zipcode=98004,98005,98006,98007,98008,98027,98029,98033,98039,98040,98052,98053,98072,98074,98075,
##        10) sqft_living< 2035 3344 6.232319e+13  503509.4
##          20) zipcode=98006,98007,98008,98027,98029,98033,98052,98053,98072,98074,98075,98077,98103,98107,98
##            40) sqft_living< 1456.5 1422 1.569806e+13  432953.7
```

```
##             80) zipcode=98027,98029,98052,98053,98072,98074,98077,98125,98136,98144,98177 491 2.980394e+12
##             81) zipcode=98006,98007,98008,98033,98075,98103,98107,98115,98116,98117,98122,98199 931 1.0645
##           41) sqft_living>=1456.5 1512 1.821779e+13  530796.0
##             82) zipcode=98006,98007,98008,98027,98029,98052,98053,98072,98074,98077,98125,98136,98144,9817
##             83) zipcode=98033,98075,98103,98107,98115,98116,98117,98122,98199 701 8.546364e+12  574020.0 *
##         21) zipcode=98004,98005,98039,98040,98102,98105,98109,98112,98119 410 1.169149e+13  647589.4
##           42) sqft_above< 1385 261 5.040823e+12  592674.1 *
##           43) sqft_above>=1385 149 4.484836e+12  743783.4 *
##       11) sqft_living>=2035 1762 8.310552e+13  702460.3
##         22) zipcode=98005,98006,98007,98008,98027,98029,98033,98052,98053,98072,98074,98075,98077,98103,98
##           44) waterfront< 0.5 1510 3.552729e+13  657086.2
##             88) view< 0.5 1300 2.541033e+13  636929.3
##              176) zipcode=98006,98007,98008,98027,98029,98052,98053,98072,98074,98075,98077,98116,98125,98
##              177) zipcode=98005,98033,98103,98107,98115,98117,98122,98199 411 9.533065e+12  712783.3 *
##             89) view>=0.5 210 6.319024e+12  781866.7 *
##           45) waterfront>=0.5 9 1.352884e+12 1619556.0 *
##         23) zipcode=98004,98039,98040,98102,98105,98109,98112,98119 243 2.060299e+13  950448.0
##           46) sqft_living< 2635 161 5.302402e+12  845865.2 *
##           47) sqft_living>=2635 82 1.008217e+13 1155787.0
##             94) sqft_living15< 3415 73 6.810131e+12 1098205.0 *
##             95) sqft_living15>=3415 9 1.066707e+12 1622844.0 *
##     3) grade>=8.5 2998 8.857481e+14  950192.7
##       6) sqft_living< 4062.5 2519 3.159003e+14  830227.4
##         12) zipcode=98001,98003,98005,98007,98008,98010,98011,98014,98019,98022,98023,98024,98027,98028,9802
##           24) zipcode=98001,98003,98011,98019,98022,98023,98030,98031,98032,98038,98042,98055,98058,98092,98
##           25) zipcode=98005,98007,98008,98010,98014,98024,98027,98028,98029,98045,98052,98053,98056,98059,98
##             50) waterfront< 0.5 1248 3.511145e+13  746044.1
##              100) sqft_living< 3155 764 1.538568e+13  685187.8
##                200) zipcode=98010,98014,98024,98027,98028,98029,98045,98056,98059,98065,98070,98072,98074,98
##                201) zipcode=98005,98007,98008,98052,98053,98075,98107,98122 282 5.632550e+12  759948.6 *
##              101) sqft_living>=3155 484 1.242996e+13  842106.5 *
##             51) waterfront>=0.5 19 7.643001e+12 1407666.0 *
##         13) zipcode=98004,98006,98033,98034,98039,98040,98102,98103,98105,98109,98112,98115,98116,98117,9811
##           26) sqft_living< 3035 455 3.759955e+13  929355.2
##             52) sqft_living< 1995 86 2.224901e+12  667063.4 *
##             53) sqft_living>=1995 369 2.807918e+13  990485.5
##              106) zipcode=98006,98033,98034,98040,98103,98115,98116,98117,98136,98144,98177 232 9.961196e+12
##                212) view< 2.5 206 5.706803e+12  843802.0 *
##                213) view>=2.5 26 1.700702e+12 1176391.0 *
##              107) zipcode=98004,98039,98102,98105,98109,98112,98119,98199 137 1.063779e+13 1175765.0 *
##           27) sqft_living>=3035 391 6.799087e+13 1316688.0
##             54) zipcode=98006,98033,98034,98040,98103,98115,98116,98117,98136,98144,98177,98199 278 3.437794
##              108) waterfront< 0.5 271 2.611058e+13 1160745.0
##                216) sqft_living< 3765 221 1.693384e+13 1111687.0 *
##                217) sqft_living>=3765 50 6.293992e+12 1377580.0 *
##              109) waterfront>=0.5 7 7.569029e+11 2209857.0 *
##             55) zipcode=98004,98039,98102,98105,98109,98112,98119,98178 113 1.747446e+13 1635347.0
##              110) sqft_living15< 3945 106 1.190305e+13 1583865.0 *
##              111) sqft_living15>=3945 7 1.036240e+12 2414929.0 *
##       7) sqft_living>=4062.5 479 3.429471e+14 1581075.0
##         14) zipcode=98003,98005,98006,98007,98010,98011,98014,98019,98022,98023,98024,98027,98028,98029,9803
##           28) grade< 11.5 269 4.551755e+13 1117753.0
##             56) zipcode=98003,98005,98010,98011,98019,98022,98023,98027,98028,98031,98032,98038,98042,98045,
##              112) zipcode=98003,98010,98011,98022,98023,98031,98032,98038,98042,98045,98056,98065,98092,9816
##              113) zipcode=98005,98019,98027,98028,98052,98053,98058,98059,98072,98074,98075,98077,98125,9819
##                226) view< 3.5 144 6.761045e+12 1061039.0 *
##                227) view>=3.5 7 5.006705e+12 1857921.0 *
##             57) zipcode=98006,98007,98014,98024,98029,98034,98116,98118,98199 66 1.601028e+13 1415593.0
##              114) sqft_living< 4820 43 3.858485e+12 1254541.0 *
##              115) sqft_living>=4820 23 8.951305e+12 1716690.0
```

```
##                 230) sqft_lot< 12119 7 5.097725e+11 1214500.0 *
##                 231) sqft_lot>=12119 16 5.903823e+12 1936398.0 *
##             29) grade>=11.5 37 1.832120e+13 1875497.0
##               58) sqft_living< 6820 30 6.028977e+12 1690450.0 *
##               59) sqft_living>=6820 7 6.862354e+12 2668556.0 *
##           15) zipcode=98004,98008,98033,98039,98040,98102,98105,98107,98109,98112,98115,98119,98122,98136,9814
##             30) sqft_above< 5685 165 8.726940e+13 2136728.0
##               60) sqft_living< 4725 95 3.264685e+13 1891728.0
##                 120) zipcode=98033,98040,98119,98122,98166,98177 47 1.068381e+13 1594429.0
##                   240) view< 2.5 32 4.252696e+12 1440304.0 *
##                   241) view>=2.5 15 4.049322e+12 1923230.0 *
##                 121) zipcode=98004,98008,98039,98102,98107,98109,98112,98115,98144 48 1.374125e+13 2182833.0
##                   242) view< 2.5 35 7.950692e+12 2019136.0
##                     484) yr_built>=1940 27 2.822467e+12 1843731.0 *
##                     485) yr_built< 1940 8 1.493915e+12 2611125.0 *
##                   243) view>=2.5 13 2.327568e+12 2623558.0 *
##               61) sqft_living>=4725 70 4.118131e+13 2469227.0
##                 122) grade< 11.5 58 2.245954e+13 2303523.0
##                   244) zipcode=98008,98040,98102,98115,98136,98155,98166 20 6.097079e+12 1893073.0
##                     488) bedrooms>=4.5 13 1.767188e+12 1658808.0 *
##                     489) bedrooms< 4.5 7 2.291483e+12 2328137.0 *
##                   245) zipcode=98004,98033,98039,98105,98109,98112,98119,98144 38 1.121971e+13 2519550.0
##                     490) sqft_lot< 12041.5 17 1.894113e+12 2256029.0 *
##                     491) sqft_lot>=12041.5 21 7.189391e+12 2732876.0 *
##                 123) grade>=11.5 12 9.431973e+12 3270125.0 *
##             31) sqft_above>=5685 8 1.912630e+13 4338250.0 *
```

```
#plot tree
library(rpart.plot)
rpart.plot(reg_tree, digits = 3)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```



```
#evaluating model performance
############################
#OUT OF SAMPLE TEST (Test Data)
############################
```

```r
#Predict values using test data
price_predict <- predict(reg_tree, test_data)
summary(price_predict) #summary of predicted values
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  227614  354507  460676  540073  601861 4338250
```

```r
summary(test_data$price) #summary of actual values
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   81000  325000  450000  543295  648000 7062500
```

```r
cor(price_predict, test_data$price)
```

```
## [1] 0.8846356
```

```r
#Measuring performance with the mean absolute error
MAE <- function(actual, predicted) {mean(abs(actual - predicted))}

#The MAE for our predictions is then:
MAE(test_data$price, price_predict)
```

```
## [1] 97456.69
```

```r
#Measuring performance with the SSE
SSE <- function(actual, predicted) {sum((actual - predicted)^2)}
SSE(test_data$price, price_predict)
```

```
## [1] 2.031268e+14
```

```r
#Measuring performance with the R-square
R2 <- function(actual, predicted) {sum((actual - predicted)^2)/((length(actual)-1)*var(actual))}
1 - R2(test_data$price, price_predict)
```

```
## [1] 0.7824997
```

```r
#evaluating model performance
############################
#IN SAMPLE TEST (Train Data)
############################

price_predict <- predict(reg_tree, train_data)
summary(price_predict) #summary of predicted values
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  227614  354507  460676  538714  601861 4338250
```

```r
summary(train_data$price) #summary of actual values
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   75000  320000  450000  538714  641250 7700000
```

```r
cor(price_predict, train_data$price)
```

```
## [1] 0.915282
```

```r
#Measuring performance with the mean absolute error
MAE <- function(actual, predicted) {mean(abs(actual - predicted))}

#The MAE for our predictions is then:
MAE(train_data$price, price_predict)
```

```
## [1] 90145.88
```

```r
#Measuring performance with the SSE
SSE <- function(actual, predicted) {sum((actual - predicted)^2)}
SSE(train_data$price, price_predict)
```

```
## [1] 3.210949e+14
```

```
#Measuring performance with the R-square
R2 <- function(actual, predicted) {sum((actual - predicted)^2)/((length(actual)-1)*var(actual))}
1 - R2(train_data$price, price_predict)
```

```
## [1] 0.8377412
```

Model Results

- **Out-Of-Sample Results**
  - Correlation: 0.8846356
  - MAE: 97456.69
  - SSE: 2.031268e+14
  - R-Squared: 0.7824997
- **In-Sample Results**
  - Correlation: 0.915282
  - MAE: 90145.88
  - SSE: 3.210949e+14
  - R-Squared: 0.8377412

Steps to create the Regression Tree

- Load the data and remove the unnecessary variables (same as the ones removed in the regression models)
- Split df into test and train datasets
- Create a Regression Tree to predict Price while using the train dataset, and a cp of 0.001
- Perform out-of-sample and in-sample tests for the model
- For both tests:
- `Get the predicted values using the test and train datasets respectively`
- `Get the summary of the predicted values and compare it againts the summary of the actual values in the Te`
- `Calculate the correlation between the predicted and actual values`
- `Calculate the MAE, SSE, and R-squared of the model`
- Once all three metrics have been calculated, compare the results with the results of the previous regression models.

VI. Model Limitation and Assumptions (15 points)

Based on the performances on both train and test data sets, determine your primary (champion) model and the other model which would be your benchmark model. Validate your models using the test sample. Do the residuals look normal? Does it matter given your technique? How is the prediction performance using Pseudo R^2, SSE, RMSE? Benchmark the model against alternatives. How good is the relative fit? Are there any serious violations of the model assumptions? Has the model had issues or limitations that the user must know? (Which assumptions are needed to support the Champion model?)

VII. Ongoing Model Monitoring Plan (5 points)

How would you picture the model needing to be monitored, which quantitative thresholds and triggers would you set to decide when the model needs to be replaced? What are the assumptions that the model must comply with for its continuous use?

VIII. Conclusion (5 points)

Summarize your results here. What is the best model for the data and why?

Bibliography (7 points)

[1] Board of Governors of the Federal Reserve System. *SR Letter 11-7: Supervisory Guidance on Model Risk Management.* 2011. https://www.federalreserve.gov/supervisionreg/srletters/sr1107a1.pdf.

[2] Yuko, Elizabeth. *The Actual Difference Between a Half, 3/4, and Full Bathroom.* April 10, 2012. https://lifehacker.com/the-actual-difference-between-a-half-3-4-and-full-bat-1848773483.

[3] Pacheco, Kaitlyn. *The 2022 U.S. Lot Size Index.* Augut 5, 2022. https://www.angi.com/articles/lot-size-index.htm.

Appendix (3 points)

Please add any additional supporting graphs, plots and data analysis.