ASSIGNMENT 8.2.3

Siddhartha Bhaumik

2022-05-14

Add Citations

- R for Everyone
- Discovering Statistics Using R

Load libraries

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:pastecs':
##
##     first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v readr   2.1.2
## v tidyr   1.2.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract() masks pastecs::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::first()   masks pastecs::first()
## x dplyr::lag()     masks stats::lag()
## x dplyr::last()    masks pastecs::last()

## corrplot 0.92 loaded

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:purrr':
##
##     some

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## Load the `data/week-7-housing.xlsx`
housing_df <- readxl::read_excel("/Users/siddharthabhaumik/Documents/GitHub/dsc520/data/week-7-housing.

# Get the structure of the data frame
str(housing_df)
```

```
## tibble [12,865 x 24] (S3: tbl_df/tbl/data.frame)
##  $ Sale Date               : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...
##  $ Sale Price              : num [1:12865] 698000 649990 572500 420000 369900 ...
##  $ sale_reason             : num [1:12865] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12865] 3 3 3 3 3 15 3 3 3 3 ...
##  $ sale_warning            : chr [1:12865] NA NA NA NA ...
##  $ sitetype                : chr [1:12865] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12865] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE
##  $ zip5                    : num [1:12865] 98052 98052 98052 98052 98052 ...
##  $ ctyname                 : chr [1:12865] "REDMOND" "REDMOND" NA "REDMOND" ...
##  $ postalctyn              : chr [1:12865] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:12865] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:12865] 47.7 47.7 47.7 47.6 47.7 ...
##  $ building_grade          : num [1:12865] 9 9 8 8 7 7 10 10 9 8 ...
##  $ square_feet_total_living: num [1:12865] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
##  $ bedrooms                : num [1:12865] 4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num [1:12865] 2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num [1:12865] 1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num [1:12865] 0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num [1:12865] 2003 2006 1987 1968 1980 ...
##  $ year_renovated          : num [1:12865] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:12865] "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num [1:12865] 6635 5570 8444 9600 7526 ...
##  $ prop_type               : chr [1:12865] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12865] 2 2 2 2 2 2 2 2 2 2 ...
```

```
# Get summary of data in Data Frame
summary(housing_df)
```

```
##    Sale Date                     Sale Price       sale_reason
##  Min.   :2006-01-03 00:00:00   Min.   :    698   Min.   : 0.00
##  1st Qu.:2008-07-07 00:00:00   1st Qu.: 460000   1st Qu.: 1.00
##  Median :2011-11-17 00:00:00   Median : 593000   Median : 1.00
##  Mean   :2011-07-28 15:07:32   Mean   : 660738   Mean   : 1.55
##  3rd Qu.:2014-06-05 00:00:00   3rd Qu.: 750000   3rd Qu.: 1.00
##  Max.   :2016-12-16 00:00:00   Max.   :4400000   Max.   :19.00
##  sale_instrument  sale_warning         sitetype          addr_full
##  Min.   : 0.000   Length:12865       Length:12865       Length:12865
##  1st Qu.: 3.000   Class :character   Class :character   Class :character
##  Median : 3.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 3.678
##  3rd Qu.: 3.000
##  Max.   :27.000
##       zip5         ctyname           postalctyn             lon
##  Min.   :98052   Length:12865       Length:12865       Min.   :-122.2
##  1st Qu.:98052   Class :character   Class :character   1st Qu.:-122.1
##  Median :98052   Mode  :character   Mode  :character   Median :-122.1
##  Mean   :98053                                         Mean   :-122.1
```

```
##  3rd Qu.:98053                                        3rd Qu.:-122.0
##  Max.   :98074                                        Max.   :-121.9
##       lat        building_grade  square_feet_total_living    bedrooms
##  Min.   :47.46   Min.   : 2.00   Min.   :  240            Min.   : 0.000
##  1st Qu.:47.67   1st Qu.: 8.00   1st Qu.: 1820            1st Qu.: 3.000
##  Median :47.69   Median : 8.00   Median : 2420            Median : 4.000
##  Mean   :47.68   Mean   : 8.24   Mean   : 2540            Mean   : 3.479
##  3rd Qu.:47.70   3rd Qu.: 9.00   3rd Qu.: 3110            3rd Qu.: 4.000
##  Max.   :47.73   Max.   :13.00   Max.   :13540            Max.   :11.000
##  bath_full_count  bath_half_count  bath_3qtr_count   year_built
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.000   Min.   :1900
##  1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1979
##  Median : 2.000   Median :1.0000   Median :0.000   Median :1998
##  Mean   : 1.798   Mean   :0.6134   Mean   :0.494   Mean   :1993
##  3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:2007
##  Max.   :23.000   Max.   :8.0000   Max.   :8.000   Max.   :2016
##  year_renovated    current_zoning       sq_ft_lot        prop_type
##  Min.   :   0.00   Length:12865       Min.   :    785   Length:12865
##  1st Qu.:   0.00   Class :character   1st Qu.:   5355   Class :character
##  Median :   0.00   Mode  :character   Median :   7965   Mode  :character
##  Mean   :  26.24                      Mean   :  22229
##  3rd Qu.:   0.00                      3rd Qu.:  12632
##  Max.   :2016.00                      Max.   :1631322
##   present_use
##  Min.   :  0.000
##  1st Qu.:  2.000
##  Median :  2.000
##  Mean   :  6.598
##  3rd Qu.:  2.000
##  Max.   :300.000
```

```
# Viewing sample data
head(housing_df)
```

```
## # A tibble: 6 x 24
##   `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning
##   <dttm>                     <dbl>       <dbl>           <dbl> <chr>
## 1 2006-01-03 00:00:00       698000           1               3 <NA>
## 2 2006-01-03 00:00:00       649990           1               3 <NA>
## 3 2006-01-03 00:00:00       572500           1               3 <NA>
## 4 2006-01-03 00:00:00       420000           1               3 <NA>
## 5 2006-01-03 00:00:00       369900           1               3 15
## 6 2006-01-03 00:00:00       184667           1              15 18 51
## # ... with 19 more variables: sitetype <chr>, addr_full <chr>, zip5 <dbl>,
## #   ctyname <chr>, postalctyn <chr>, lon <dbl>, lat <dbl>,
## #   building_grade <dbl>, square_feet_total_living <dbl>, bedrooms <dbl>,
## #   bath_full_count <dbl>, bath_half_count <dbl>, bath_3qtr_count <dbl>,
## #   year_built <dbl>, year_renovated <dbl>, current_zoning <chr>,
## #   sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>
```

```
## 3.a.i) Any issues in the housing dataset which needs cleanup?
# Yes, I see 'ctyname' column has many missing values like 'NA'.
housing_upd_df <- housing_df %>% filter(`Sale Date` > "1999-12-31") %>% filter(year_built > "2000") %>%

# checking dimensions
```

```
dim(housing_upd_df)
```

```
## [1] 2653   24
```

```
# Viewing sample data
head(housing_upd_df)
```

```
## # A tibble: 6 x 24
##   `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning
##   <dttm>                     <dbl>       <dbl>           <dbl> <chr>
## 1 2006-01-03 00:00:00       698000           1               3 <NA>
## 2 2006-01-03 00:00:00       649990           1               3 <NA>
## 3 2006-01-04 00:00:00       526787           1               3 <NA>
## 4 2006-01-05 00:00:00       507950           1               3 <NA>
## 5 2006-01-06 00:00:00       589950           1               3 <NA>
## 6 2006-01-12 00:00:00       717390           1               3 <NA>
## # ... with 19 more variables: sitetype <chr>, addr_full <chr>, zip5 <dbl>,
## #   ctyname <chr>, postalctyn <chr>, lon <dbl>, lat <dbl>,
## #   building_grade <dbl>, square_feet_total_living <dbl>, bedrooms <dbl>,
## #   bath_full_count <dbl>, bath_half_count <dbl>, bath_3qtr_count <dbl>,
## #   year_built <dbl>, year_renovated <dbl>, current_zoning <chr>,
## #   sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>
```

```
## 3.b.i) Explain any transformations or modifications you made to the dataset
# I have removed rows with missing 'ctyname' i.e. 'ctyname' as 'NA'. This will make my dataset cleaner
# Also, I have excluded rows with 'Sale Date' and 'year built' older than Jan 2000 as I am only interes

## 3.b.ii) Create two variables; one that will contain the variables Sale Price and Square Foot of Lot
## (same variables used from previous assignment on simple regression) and one that will contain Sale P
## Explain the basis for your additional predictor selections.

## Fit a linear model using the `sq_ft_lot` variable as the predictor and `Sale Price` as the outcome
price_per_sq_ft_df <-  lm(`Sale Price` ~ sq_ft_lot, data = housing_upd_df)

price_per_sq_ft_df2 <-  lm(`Sale Price` ~ sq_ft_lot + year_built + square_feet_total_living + zip5 + be

# Few additional predictors which are important for a home buyer and have significant weight on the hou
# location i.e. zip5, year_built, number of bedrooms, total living area, total lot square feet and zone

## 3.b.iii) Execute a summary() function on two variables defined in the previous step to compare the m
## What are the R2 and Adjusted R2 statistics? Explain what these results tell you about the overall mo
## Did the inclusion of the additional predictors help explain any large variations found in Sale Price

  summary(price_per_sq_ft_df)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot, data = housing_upd_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1796871  -166556   -81240    31496  3611344
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 6.811e+05  1.581e+04   43.08   <2e-16 ***
## sq_ft_lot   2.077e+01  2.055e+00   10.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 477400 on 2651 degrees of freedom
## Multiple R-squared:  0.03711,    Adjusted R-squared:  0.03675
## F-statistic: 102.2 on 1 and 2651 DF,  p-value: < 2.2e-16
```

```
# Multiple R-squared:  0.03711, Adjusted R-squared:  0.03675
# R-squared and Adjusted R-squared values are very low indicating that the predictor or independent var

summary(price_per_sq_ft_df2)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot + year_built + square_feet_total_living +
##     zip5 + bedrooms + current_zoning, data = housing_upd_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1139054  -134213   -42744    35934  3660721
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.954e+08  4.283e+08  -0.456   0.6484
## sq_ft_lot                  7.587e+00  2.402e+00   3.158   0.0016 **
## year_built                 1.507e+04  2.297e+03   6.559 6.50e-11 ***
## square_feet_total_living   1.738e+02  1.524e+01  11.407  < 2e-16 ***
## zip5                       1.685e+03  4.362e+03   0.386   0.6993
## bedrooms                   1.745e+04  1.495e+04   1.167   0.2433
## current_zoningR1          -7.248e+04  1.199e+05  -0.604   0.5457
## current_zoningR12          9.351e+04  7.240e+04   1.292   0.1966
## current_zoningR18         -2.624e+04  1.189e+05  -0.221   0.8254
## current_zoningR3           1.224e+05  1.344e+05   0.910   0.3627
## current_zoningR4           6.272e+04  6.750e+04   0.929   0.3529
## current_zoningR4/C         4.722e+04  8.322e+04   0.567   0.5705
## current_zoningR5          -2.032e+04  7.036e+04  -0.289   0.7727
## current_zoningR6          -9.407e+03  7.303e+04  -0.129   0.8975
## current_zoningR6/C         5.134e+05  1.130e+05   4.543 5.81e-06 ***
## current_zoningR8           7.612e+05  1.059e+05   7.186 8.65e-13 ***
## current_zoningRA5          3.462e+05  4.461e+05   0.776   0.4379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 440400 on 2636 degrees of freedom
## Multiple R-squared:  0.1851, Adjusted R-squared:  0.1802
## F-statistic: 37.43 on 16 and 2636 DF,  p-value: < 2.2e-16
```

```
# Multiple R-squared:  0.1851,  Adjusted R-squared:  0.1802
# Addition of more number of predictors significantly increased the R-squared and Adjusted R-squared va

## 3.b.iv)   Considering the parameters of the multiple regression model you have created. What are the

lm.beta(price_per_sq_ft_df)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot, data = housing_upd_df)
##
## Standardized Coefficients::
## (Intercept)   sq_ft_lot
##   0.0000000   0.1926369
```

```
lm.beta(price_per_sq_ft_df2)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot + year_built + square_feet_total_living +
##     zip5 + bedrooms + current_zoning, data = housing_upd_df)
##
## Standardized Coefficients::
##              (Intercept)                sq_ft_lot                year_built
##              0.000000000              0.070369524              0.126110744
## square_feet_total_living                     zip5                 bedrooms
##              0.280154198              0.007650186              0.026400597
##         current_zoningR1        current_zoningR12        current_zoningR18
##             -0.014683501              0.050296735             -0.004549188
##         current_zoningR3         current_zoningR4       current_zoningR4/C
##              0.018867605              0.061947322              0.016804917
##         current_zoningR5         current_zoningR6       current_zoningR6/C
##             -0.013701515             -0.005000006              0.099952406
##         current_zoningR8        current_zoningRA5
##              0.162760296              0.013817658
```

```
summary(lm.beta(price_per_sq_ft_df2))
```

```
##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot + year_built + square_feet_total_living +
##     zip5 + bedrooms + current_zoning, data = housing_upd_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1139054  -134213   -42744    35934  3660721
##
## Coefficients:
##                             Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)              -1.954e+08    0.000e+00  4.283e+08  -0.456   0.6484
## sq_ft_lot                 7.587e+00    7.037e-02  2.402e+00   3.158   0.0016
## year_built                1.507e+04    1.261e-01  2.297e+03   6.559 6.50e-11
## square_feet_total_living  1.738e+02    2.802e-01  1.524e+01  11.407  < 2e-16
## zip5                      1.685e+03    7.650e-03  4.362e+03   0.386   0.6993
## bedrooms                  1.745e+04    2.640e-02  1.495e+04   1.167   0.2433
## current_zoningR1         -7.248e+04   -1.468e-02  1.199e+05  -0.604   0.5457
## current_zoningR12         9.351e+04    5.030e-02  7.240e+04   1.292   0.1966
## current_zoningR18        -2.624e+04   -4.549e-03  1.189e+05  -0.221   0.8254
## current_zoningR3          1.224e+05    1.887e-02  1.344e+05   0.910   0.3627
## current_zoningR4          6.272e+04    6.195e-02  6.750e+04   0.929   0.3529
## current_zoningR4/C        4.722e+04    1.681e-02  8.322e+04   0.567   0.5705
## current_zoningR5         -2.032e+04   -1.370e-02  7.036e+04  -0.289   0.7727
```

```
## current_zoningR6         -9.407e+03   -5.000e-03  7.303e+04  -0.129   0.8975
## current_zoningR6/C        5.134e+05    9.995e-02  1.130e+05   4.543 5.81e-06
## current_zoningR8          7.612e+05    1.628e-01  1.059e+05   7.186 8.65e-13
## current_zoningRA5         3.462e+05    1.382e-02  4.461e+05   0.776   0.4379
##
## (Intercept)
## sq_ft_lot                **
## year_built               ***
## square_feet_total_living ***
## zip5
## bedrooms
## current_zoningR1
## current_zoningR12
## current_zoningR18
## current_zoningR3
## current_zoningR4
## current_zoningR4/C
## current_zoningR5
## current_zoningR6
## current_zoningR6/C       ***
## current_zoningR8         ***
## current_zoningRA5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 440400 on 2636 degrees of freedom
## Multiple R-squared:  0.1851, Adjusted R-squared:  0.1802
## F-statistic: 37.43 on 16 and 2636 DF,  p-value: < 2.2e-16
```

```
  ## 3.b.v)  Calculate the confidence intervals for the parameters in your model and explain what the r
```

```
# Calculate the mean and standard error
l.model1 <- lm( `Sale Price` ~ sq_ft_lot, housing_upd_df)

l.model2 <- lm( `Sale Price` ~ year_built, housing_upd_df)

l.model3 <- lm( `Sale Price` ~ square_feet_total_living, housing_upd_df)

l.model4 <- lm( `Sale Price` ~ zip5, housing_upd_df)

l.model5 <- lm( `Sale Price` ~ bedrooms, housing_upd_df)

l.model6 <- lm( `Sale Price` ~ current_zoning, housing_upd_df)

# Calculate the confidence interval
confint(l.model1, level=0.95)
```

```
##                   2.5 %       97.5 %
## (Intercept) 650053.6871 712056.98636
## sq_ft_lot       16.7397     24.79776
```

```
confint(l.model2, level=0.95)
```

```
##                    2.5 %       97.5 %
## (Intercept) -48903467.05 -30888124.41
```

```
## year_built      15782.02     24751.42
```

```
confint(l.model3, level=0.95)
```

```
##                                2.5 %      97.5 %
## (Intercept)              71153.3988 208718.4399
## square_feet_total_living   200.1819    244.3093
```

```
confint(l.model4, level=0.95)
```

```
##                    2.5 %       97.5 %
## (Intercept) -1.16619e+09 478389606.23
## zip5        -4.87066e+03     11901.83
```

```
confint(l.model5, level=0.95)
```

```
##                 2.5 % 97.5 %
## (Intercept)  82534.78 279024
## bedrooms    134840.17 183695
```

```
confint(l.model6, level=0.95)
```

```
##                          2.5 %    97.5 %
## (Intercept)         356700.182  615385.7
## current_zoningR1    150188.925  592463.7
## current_zoningR12    -2802.869  287005.9
## current_zoningR18  -376951.206  116018.0
## current_zoningR3    350636.501  889133.3
## current_zoningR4    230878.659  493351.8
## current_zoningR4/C  239348.671  567558.9
## current_zoningR5    114958.471  392831.2
## current_zoningR6     23465.632  314044.4
## current_zoningR6/C  665212.655 1119449.4
## current_zoningR8    908305.340 1335264.8
## current_zoningRA5    39265.175 1886649.0
```

```
## 3.b.vi)  Assess the improvement of the new model compared to your original model (simple regression m
## whether this change is significant by performing an analysis of variance.
anova(price_per_sq_ft_df,price_per_sq_ft_df2)
```

```
## Analysis of Variance Table
##
## Model 1: `Sale Price` ~ sq_ft_lot
## Model 2: `Sale Price` ~ sq_ft_lot + year_built + square_feet_total_living +
##     zip5 + bedrooms + current_zoning
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1   2651 6.0410e+14
## 2   2636 5.1123e+14 15 9.2871e+13 31.924 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 3.b.vii)  Perform case wise diagnostics to identify outliers and/or influential cases, storing each i
## in a dataframe assigned to a unique variable name.

#price_per_sq_ft_df2 <-  lm(`Sale Price` ~ sq_ft_lot + year_built + square_feet_total_living + zip5 + b
# Finding outliers for all of my predictors

# Predictor sq_ft_lot
```

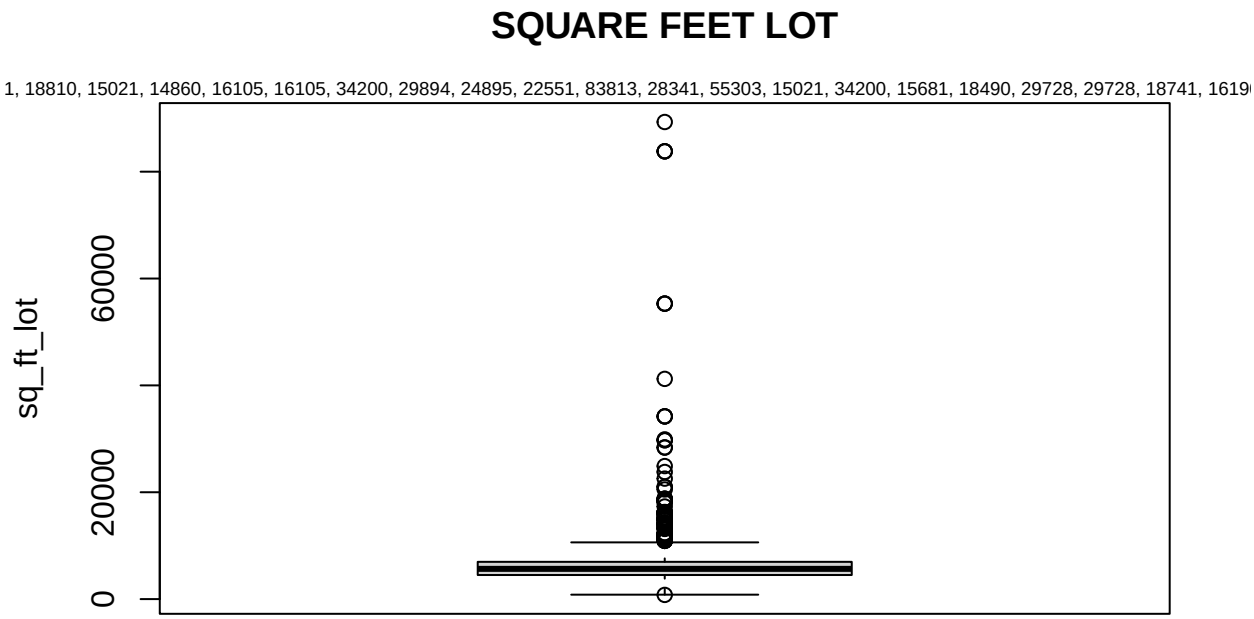
```
summary(housing_upd_df$sq_ft_lot)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     785    4505    5659    6234    6971   89298
```

```
sq_ft_outlier_values <- boxplot.stats(housing_upd_df$sq_ft_lot,coef=3)$out  # outlier values.

# Display Outliers
print(sq_ft_outlier_values)
```

```
##  [1] 34200 55303 18297 18485 15021 18045 21010 18741 16264 16252 17328 28408
## [13] 89298 15011 18810 15021 14860 16105 16105 34200 29894 24895 22551 83813
## [25] 28341 55303 15021 34200 15681 18490 29728 29728 18741 16190 21010 83813
## [37] 83813 14710 55303 14380 18741 16305 18490 15858 15095 41217 15368 16252
## [49] 23787 20520
```

```
#Plot sq_ft_lot
boxplot(housing_upd_df$sq_ft_lot, main="SQUARE FEET LOT", ylab = "sq_ft_lot")

#Plot sq_ft_lot with outliers
mtext(paste("Outliers: ", paste(sq_ft_outlier_values, collapse=", ")), cex=0.6)
```




```
# Predictor year_built
summary(housing_upd_df$year_built)
```

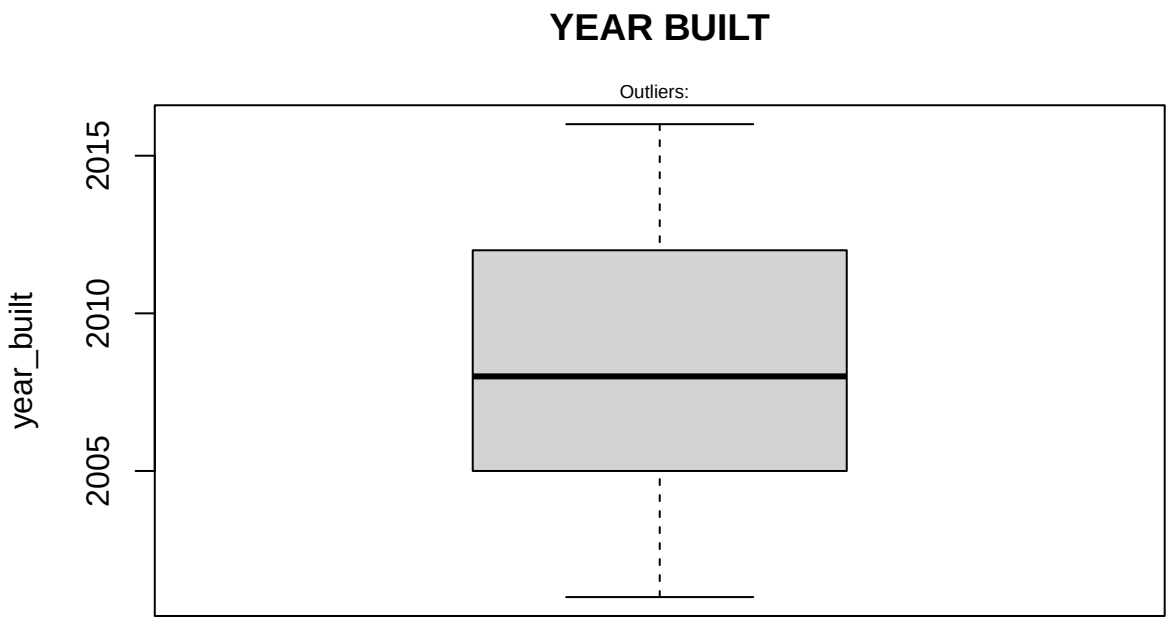
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2001    2005    2008    2009    2012    2016
```

```
yr_outlier_values <- boxplot.stats(housing_upd_df$year_built,coef=3)$out  # outlier values.

# Display Outliers
print(yr_outlier_values)
```

```
## numeric(0)
```

```
#Plot year_built
boxplot(housing_upd_df$year_built, main="YEAR BUILT", ylab = "year_built")
```

```
#Plot year_built with outliers
mtext(paste("Outliers: ", paste(yr_outlier_values, collapse=", ")), cex=0.6)
```



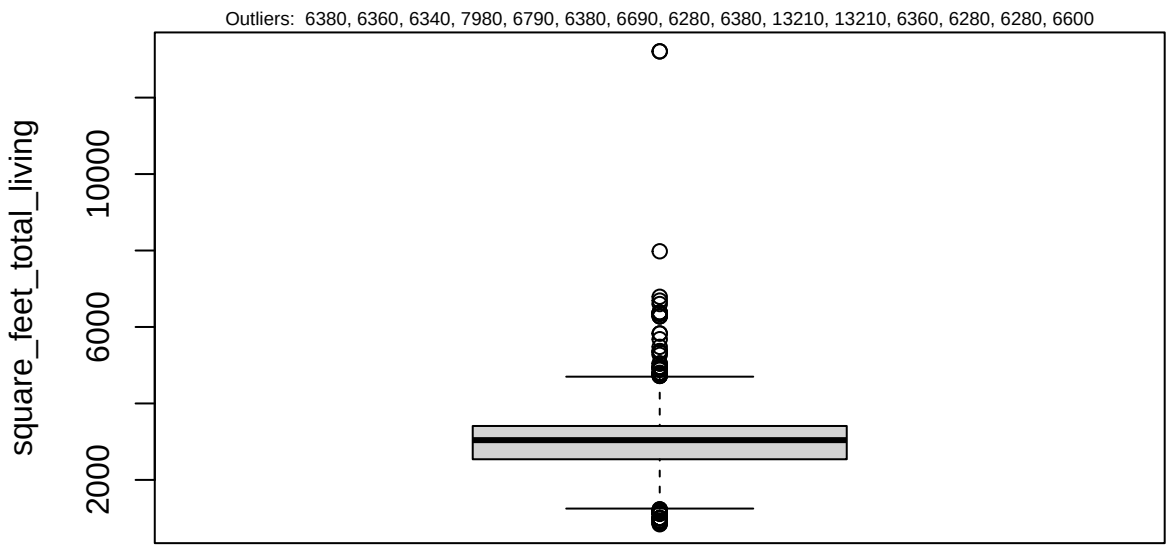

```
# Predictor square_feet_total_living
summary(housing_upd_df$square_feet_total_living)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     840    2540    3040    3017    3410   13210
```

```
sq_feet_outlier_values <- boxplot.stats(housing_upd_df$square_feet_total_living,coef=3)$out  # outlier

# Display Outliers
print(sq_feet_outlier_values)
```

```
##  [1]  6380  6360  6340  7980  6790  6380  6690  6280  6380 13210 13210  6360
## [13]  6280  6280  6600
```

```
#Plot square_feet_total_living
boxplot(housing_upd_df$square_feet_total_living, main="square_feet_total_living", ylab = "square_feet_t

#Plot square_feet_total_living with outliers
mtext(paste("Outliers: ", paste(sq_feet_outlier_values, collapse=", ")), cex=0.6)
```




```
# Predictor bedrooms
summary(housing_upd_df$bedrooms)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   4.000   4.000   3.954   4.000  11.000
```

```
bedr_outlier_values <- boxplot.stats(housing_upd_df$bedrooms,coef=10)$out  # outlier values.

# Display Outliers
print(bedr_outlier_values)
```

```
##    [1]  3  3  3  3  5  2  3  5  5  3  3  3  3  5  5  5  5  3  3  3  3  3  5  3
##   [25]  3  5  3  5  3  3  5  5  3  5  5  3  5  5  5  3  5  5  3  3  3  6  3  3
##   [49]  3  5  5  3  3  3  3  5  5  3  5  3  3  5  3  5  5  3  3  5  5  3  5  6
##   [73]  5  5  5  5  5  5  5  3  3  3  3  5  5  3  3  5  3  3  5  3  3  3  5  5
##   [97]  5  5  5  5  3  5  5  5  5  3  5  3  5  5  5  5  5  3  5  3  5  5  1  5
##  [121]  3  3  3  3  5  5  3  3  5  5  5  5  5  3  3  5  5  3  3  3  5  6  5  5
##  [145]  5  3  3  5  5  3  3  3  5  3  5  5  5  5  3  3  3  3  5  3  3  3  3  3
##  [169]  3  3  5  3  3  3  3  3  5  5  3  5  3  3 10  3  3  3  3  3  3  3  5  3
##  [193]  3  5  3  3  3  3  3  5  3  3  3  5  3  5  3  3  3  5  3  5  3  2  3 11
##  [217]  5  5  5  5  5  3  2  3  3  5  5  3  3  3  5  3  3  3  3  3  6  7  3  3
##  [241]  3  3  5  5  3  3  5  3  5  3  5  3  3  5  5  3  3  5  3  5  5  3  5  3
##  [265]  5  5  3  3  3  3  5  5  3  3  5  5  5  5  5  5  3  3  2  5  3  3  3  3
##  [289]  3  6  5  5  3  5  3  3  3  5  3  3  5  3  3  3  5  5  2  3  5  5  3  3
##  [313]  2  2  5  5  5  3  5  5  3  5  2  3  3  5  3  5  3  6  2  3  5  3  3  5
##  [337]  3  3  6  5  5  2  3  3  3  3  3  2  3  3  3  2  3  3  2  5  3  5  3  3
##  [361]  5  3  2  2  2  2  3  3  5  3  5  5  3  2  2  2  3  3  3  2  2  2  2  5
##  [385]  2  2  5  2  2  3  2  5  3  2  3  3  5  5  5  3  3  3  2  3  2  5  2  2
##  [409]  5  2  2  3  5  5  3  5  3  2  3  5  5  5  3  3  3  3  3  2  3  3  3  2
##  [433]  5  5  3  3  3  5  2  3  3  3  3  3  3  5  3  3  5  5  5  5  3  5  3  3
##  [457]  3  5  3  3  5  3  5  5  3  5  5  5  5  3  5  5  5  3  3  3  5  3  5  3
##  [481]  3  5  5  5  5  3  2  5  3  5  3  3  3  3  3  3  3  3  3  3  5  3  5  5  5
##  [505]  5  5  3  3  3  3  5  3  5  3  5  3  5  3  2  3  6  3  5  1  3  5  5  3
##  [529]  5  3  5  5  5  5  5  5  3  5  5  5  5  5  3  3  6  5  5  5  5  3  5  5
##  [553]  5  5  5  5  5  5  5  5  3  5  3  5  5  5  5  5  3  3  5  5  5  5  3  5
```

```
##  [577]  3  3  3  3  5  5  3  5  5  5  5  5  5  3  3  5  0  3  3  5  5  5  3  3
##  [601]  5  5  6  5  3  3  5  3  5  3  5  3  5  3  5  5  5  5  5  3  3  5  5  5
##  [625]  5  5  5  5  5  5  3  5  5  3  3  5  3  5  5  5  3  5  5  5  3  3  5  5
##  [649]  5  5  3  5  5  5  3  3  5  3  3  5  3  5  5  6  3  5  3  3  5  5  5  5
##  [673]  3  3  3  3  3  3  3  3  3  5  5  3  5  5  5  5  5  3  3  5  5  5  5  3
##  [697]  3  5  3  3  5  5  3  5  5  3  5  3  0  3  5  3  5  3  3  3  3  2  5  5
##  [721]  5  5  3  2  5  3  5  3  5  5  3  2  3  5  3  5  3  5  5  5  2  5  5  3
##  [745]  5  5  3  5  5  5  3  5  5  3  5  2  3  5  3  2  5  5  2  5  5  3  5  5
##  [769]  3  5  5  3  5  5  3  3  3  5  3  3  3  5  3  3  3  3  3  5  3  5  5  5
##  [793]  3  5  3  2  5  3  5  5  3  3  5  5  3  3  5  5  5  3  3  3  5  5  5  5
##  [817]  5  5  3  5  3  2  5  2  5  5  3  2  5  5  3  5  5  3  5  2  5  5  5  3
##  [841]  3  3  5  2  5  5  5  3  2  3  3  3  3  5  3  3  5  6  3  6  5  5  3  5
##  [865]  3  6  5  3  5  3  5  5  5  5  3  5  5  3  3  3  3  3  5  5  5  5  5  5
##  [889]  3  3  3  5  3  3  3  3  3  5  5  3  3  3  3  5  3  3  3  3  5  3  5  3
##  [913]  3  5  5  5  5  5  5  3  5  3  3  5  5  6  5  3  3  3  3  5  6  5  5  3
##  [937]  5  3  3  5  3  3  3  3  3  5  5  3  3  0  5  3  3  3  2  3  3  6  2  3
##  [961]  2  5  3  5  3  5  3  2  3  2  5  5  5  3  3  3  3  3  3  5  3  0  3  3
##  [985]  3  3  2  2  3  5  5  5  5  3  3  5  2  3  3  5  3  3  3  3  5  3  3  5
## [1009]  3  3  5  5  3  2  5  5  3  3  5  5  5  3  5  5  5  5  3  3  3
```

```
# Too many outliers or there is something wrong,

#Plot bedrooms
boxplot(housing_upd_df$bedrooms, main="BEDROOMS", ylab = "bedrooms")
```


```
# Predictor current_zoning
#summary(housing_upd_df$current_zoning)

#zone_outlier_values <- boxplot.stats(housing_upd_df$current_zoning,coef=3)$out  # outlier values.
# non-numeric argument to binary operator

# Display Outliers
#print(zone_outlier_values)
# Too many outliers or there is something wrong,
```

```
#Plot current_zoning
#boxplot(housing_upd_df$current_zoning, main="current_zoning", ylab = "current_zoning")


## 3.b.viii)  Calculate the standardized residuals using the appropriate command, specifying those that
## storing the results of large residuals in a variable you create.

# Model#1
l.modeli <- lm( `Sale Price` ~ sq_ft_lot, housing_upd_df)

#calculate the standardized residuals
standard_res_modeli <- rstandard(l.modeli)

print(head(standard_res_modeli))
```

```
##          1          2          3          4          5          6
## -0.2532223 -0.3074716 -0.4384667 -0.4975968 -0.3970284 -0.1745195
```

```
#column bind standardized residuals back to original data frame
final_data_modeli <- cbind(housing_upd_df, standard_res_modeli)

#sort standardized residuals descending
head(final_data_modeli[order(-standard_res_modeli),])
```

```
##       Sale Date Sale Price sale_reason sale_instrument sale_warning sitetype
## 1365 2011-11-17    4380542           1              22        11 45       R1
## 1364 2011-11-17    4380542           1              22        11 45       R1
## 1368 2011-11-17    4380542           1              22        11 45       R1
## 1360 2011-11-17    4380542           1              22        11 45       R1
## 1361 2011-11-17    4380542           1              22        11 45       R1
## 1357 2011-11-17    4380542           1              22        11 45       R1
##               addr_full  zip5 ctyname postalctyn       lon      lat
## 1365  11719 171ST PL NE 98052 REDMOND    REDMOND -122.1125 47.70568
## 1364  11902 171ST PL NE 98052 REDMOND    REDMOND -122.1120 47.70651
## 1368 16906 NE 118TH WAY 98052 REDMOND    REDMOND -122.1146 47.70631
## 1360 16944 NE 118TH WAY 98052 REDMOND    REDMOND -122.1138 47.70624
## 1361  16909 NE 120TH ST 98052 REDMOND    REDMOND -122.1145 47.70694
## 1357  11818 171ST PL NE 98052 REDMOND    REDMOND -122.1119 47.70639
##      building_grade square_feet_total_living bedrooms bath_full_count
## 1365              8                     2440        3               2
## 1364              8                     2550        4               2
## 1368              8                     2960        4               3
## 1360              8                     3200        5               2
## 1361              8                     3200        5               2
## 1357              8                     2450        4               2
##      bath_half_count bath_3qtr_count year_built year_renovated current_zoning
## 1365               1               0       2011             0             R4
## 1364               1               0       2010             0             R4
## 1368               0               1       2012             0             R4
## 1360               1               0       2010             0             R4
## 1361               1               0       2012             0             R4
## 1357               1               0       2010             0             R4
##      sq_ft_lot prop_type present_use standard_res_modeli
## 1365      4244         R           2            7.566908
## 1364      4368         R           2            7.561478
```

```
## 1368      4451         R           2            7.557845
## 1360      4584         R           2            7.552025
## 1361      4681         R           2            7.547782
## 1357      4749         R           2            7.544809
```

```
# Model#2
l.modelii <- lm( `Sale Price` ~ year_built, housing_upd_df)

# calculate the standardized residuals
standard_res_modelii <- rstandard(l.modelii)

print(head(standard_res_modelii))
```

```
##           1           2           3           4           5           6
## -0.00094224 -0.22796076 -0.44275494 -0.48205869 -0.31096417 -0.08733911
```

```
#column bind standardized residuals back to original data frame
final_data_modelii <- cbind(housing_upd_df, standard_res_modelii)

#sort standardized residuals descending
head(final_data_modelii[order(-standard_res_modelii),])
```

```
##       Sale Date Sale Price sale_reason sale_instrument sale_warning sitetype
## 1357 2011-11-17    4380542           1              22        11 45       R1
## 1360 2011-11-17    4380542           1              22        11 45       R1
## 1364 2011-11-17    4380542           1              22        11 45       R1
## 1365 2011-11-17    4380542           1              22        11 45       R1
## 1366 2011-11-17    4380542           1              22        11 45       R1
## 1356 2011-11-17    4380542           1              22        11 45       R1
##               addr_full  zip5 ctyname postalctyn       lon      lat
## 1357  11818 171ST PL NE 98052 REDMOND    REDMOND -122.1119 47.70639
## 1360 16944 NE 118TH WAY 98052 REDMOND    REDMOND -122.1138 47.70624
## 1364  11902 171ST PL NE 98052 REDMOND    REDMOND -122.1120 47.70651
## 1365  11719 171ST PL NE 98052 REDMOND    REDMOND -122.1125 47.70568
## 1366 16955 NE 118TH WAY 98052 REDMOND    REDMOND -122.1135 47.70579
## 1356  17137 NE 120TH ST 98052 REDMOND    REDMOND -122.1113 47.70674
##      building_grade square_feet_total_living bedrooms bath_full_count
## 1357              8                     2450        4               2
## 1360              8                     3200        5               2
## 1364              8                     2550        4               2
## 1365              8                     2440        3               2
## 1366              8                     3160        4               2
## 1356              8                     3290        4               2
##      bath_half_count bath_3qtr_count year_built year_renovated current_zoning
## 1357               1               0       2010              0            R4
## 1360               1               0       2010              0            R4
## 1364               1               0       2010              0            R4
## 1365               1               0       2011              0            R4
## 1366               1               0       2011              0            R4
## 1356               1               0       2012              0            R4
##      sq_ft_lot prop_type present_use standard_res_modelii
## 1357      4749         R           2             7.385876
## 1360      4584         R           2             7.385876
## 1364      4368         R           2             7.385876
## 1365      4244         R           2             7.343923
## 1366      5778         R           2             7.343923
```

```
## 1356      6712         R           2             7.302133
```

```
# Model#3
l.modeliii <- lm( `Sale Price` ~ square_feet_total_living, housing_upd_df)

# calculate the standardized residuals
standard_res_modeliii <- rstandard(l.modeliii)

print(head(standard_res_modeliii))
```

```
##          1          2          3          4          5          6
## -0.1463219 -0.2863021 -0.3618745 -0.4033589 -0.2668148 -0.2406550
```

```
#column bind standardized residuals back to original data frame
final_data_modeliii <- cbind(housing_upd_df, standard_res_modeliii)

#sort standardized residuals descending
head(final_data_modeliii[order(-standard_res_modeliii),])
```

```
##       Sale Date Sale Price sale_reason sale_instrument sale_warning sitetype
## 1365 2011-11-17    4380542           1              22        11 45       R1
## 1357 2011-11-17    4380542           1              22        11 45       R1
## 1364 2011-11-17    4380542           1              22        11 45       R1
## 1358 2011-11-17    4380542           1              22        11 45       R1
## 1363 2011-11-17    4380542           1              22        11 45       R1
## 1368 2011-11-17    4380542           1              22        11 45       R1
##              addr_full  zip5 ctyname postalctyn       lon      lat
## 1365  11719 171ST PL NE 98052 REDMOND    REDMOND -122.1125 47.70568
## 1357  11818 171ST PL NE 98052 REDMOND    REDMOND -122.1119 47.70639
## 1364  11902 171ST PL NE 98052 REDMOND    REDMOND -122.1120 47.70651
## 1358 17011 NE 118TH WAY 98052 REDMOND    REDMOND -122.1134 47.70580
## 1363  17136 NE 120TH ST 98052 REDMOND    REDMOND -122.1112 47.70716
## 1368 16906 NE 118TH WAY 98052 REDMOND    REDMOND -122.1146 47.70631
##      building_grade square_feet_total_living bedrooms bath_full_count
## 1365              8                     2440        3               2
## 1357              8                     2450        4               2
## 1364              8                     2550        4               2
## 1358              8                     2750        4               2
## 1363              8                     2810        4               2
## 1368              8                     2960        4               3
##      bath_half_count bath_3qtr_count year_built year_renovated current_zoning
## 1365               1               0       2011              0            R4
## 1357               1               0       2010              0            R4
## 1364               1               0       2010              0            R4
## 1358               1               0       2012              0            R4
## 1363               1               0       2012              0            R4
## 1368               0               1       2012              0            R4
##      sq_ft_lot prop_type present_use standard_res_modeliii
## 1365      4244         R           2              8.144869
## 1357      4749         R           2              8.139945
## 1364      4368         R           2              8.090743
## 1358      5816         R           2              7.992495
## 1363     13289         R           2              7.963061
## 1368      4451         R           2              7.889553
```

```
# Model#4
l.modeliv <- lm( `Sale Price` ~ zip5, housing_upd_df)

# calculate the standardized residuals
standard_res_modeliv <- rstandard(l.modeliv)

print(head(standard_res_modeliv))
```

```
##          1          2          3          4          5          6
## -0.2297639 -0.3284850 -0.5818226 -0.6205564 -0.4519430 -0.1898930
```

```
#column bind standardized residuals back to original data frame
final_data_modeliv <- cbind(housing_upd_df, standard_res_modeliv)

#sort standardized residuals descending
head(final_data_modeliv[order(-standard_res_modeliv),])
```

```
##       Sale Date Sale Price sale_reason sale_instrument sale_warning sitetype
## 1356 2011-11-17    4380542           1              22        11 45       R1
## 1357 2011-11-17    4380542           1              22        11 45       R1
## 1358 2011-11-17    4380542           1              22        11 45       R1
## 1359 2011-11-17    4380542           1              22        11 45       R1
## 1360 2011-11-17    4380542           1              22        11 45       R1
## 1361 2011-11-17    4380542           1              22        11 45       R1
##               addr_full  zip5 ctyname postalctyn       lon      lat
## 1356  17137 NE 120TH ST 98052 REDMOND    REDMOND -122.1113 47.70674
## 1357  11818 171ST PL NE 98052 REDMOND    REDMOND -122.1119 47.70639
## 1358 17011 NE 118TH WAY 98052 REDMOND    REDMOND -122.1134 47.70580
## 1359 16943 NE 118TH WAY 98052 REDMOND    REDMOND -122.1138 47.70579
## 1360 16944 NE 118TH WAY 98052 REDMOND    REDMOND -122.1138 47.70624
## 1361  16909 NE 120TH ST 98052 REDMOND    REDMOND -122.1145 47.70694
##      building_grade square_feet_total_living bedrooms bath_full_count
## 1356              8                     3290        4               2
## 1357              8                     2450        4               2
## 1358              8                     2750        4               2
## 1359              8                     3010        4               2
## 1360              8                     3200        5               2
## 1361              8                     3200        5               2
##      bath_half_count bath_3qtr_count year_built year_renovated current_zoning
## 1356               1               0       2012             0            R4
## 1357               1               0       2010             0            R4
## 1358               1               0       2012             0            R4
## 1359               0               1       2012             0            R4
## 1360               1               0       2010             0            R4
## 1361               1               0       2012             0            R4
##      sq_ft_lot prop_type present_use standard_res_modeliv
## 1356      6712         R           2             7.342506
## 1357      4749         R           2             7.342506
## 1358      5816         R           2             7.342506
## 1359      8908         R           2             7.342506
## 1360      4584         R           2             7.342506
## 1361      4681         R           2             7.342506
```

```
# Model#5
l.modelv <- lm( `Sale Price` ~ bedrooms, housing_upd_df)
```

```
# calculate the standardized residuals
standard_res_modelv <- rstandard(l.modelv)

print(head(standard_res_modelv))
```

```
##          1          2          3          4          5          6
## -0.2538937 -0.3555997 -0.2792876 -0.3192051 -0.4827904  0.1246197
```

```
#column bind standardized residuals back to original data frame
final_data_modelv <- cbind(housing_upd_df, standard_res_modelv)

#sort standardized residuals descending
head(final_data_modelv[order(-standard_res_modelv),])
```

```
##       Sale Date Sale Price sale_reason sale_instrument sale_warning sitetype
## 1365 2011-11-17    4380542           1              22        11 45       R1
## 1356 2011-11-17    4380542           1              22        11 45       R1
## 1357 2011-11-17    4380542           1              22        11 45       R1
## 1358 2011-11-17    4380542           1              22        11 45       R1
## 1359 2011-11-17    4380542           1              22        11 45       R1
## 1363 2011-11-17    4380542           1              22        11 45       R1
##               addr_full  zip5 ctyname postalctyn       lon      lat
## 1365  11719 171ST PL NE 98052 REDMOND    REDMOND -122.1125 47.70568
## 1356  17137 NE 120TH ST 98052 REDMOND    REDMOND -122.1113 47.70674
## 1357  11818 171ST PL NE 98052 REDMOND    REDMOND -122.1119 47.70639
## 1358 17011 NE 118TH WAY 98052 REDMOND    REDMOND -122.1134 47.70580
## 1359 16943 NE 118TH WAY 98052 REDMOND    REDMOND -122.1138 47.70579
## 1363  17136 NE 120TH ST 98052 REDMOND    REDMOND -122.1112 47.70716
##      building_grade square_feet_total_living bedrooms bath_full_count
## 1365              8                     2440        3               2
## 1356              8                     3290        4               2
## 1357              8                     2450        4               2
## 1358              8                     2750        4               2
## 1359              8                     3010        4               2
## 1363              8                     2810        4               2
##      bath_half_count bath_3qtr_count year_built year_renovated current_zoning
## 1365               1               0       2011              0             R4
## 1356               1               0       2012              0             R4
## 1357               1               0       2010              0             R4
## 1358               1               0       2012              0             R4
## 1359               0               1       2012              0             R4
## 1363               1               0       2012              0             R4
##      sq_ft_lot prop_type present_use standard_res_modelv
## 1365      4244         R           2            7.887215
## 1356      6712         R           2            7.547323
## 1357      4749         R           2            7.547323
## 1358      5816         R           2            7.547323
## 1359      8908         R           2            7.547323
## 1363     13289         R           2            7.547323
```

```
# Model#6
l.modelvi <- lm( `Sale Price` ~ current_zoning, housing_upd_df)

# calculate the standardized residuals
standard_res_modelvi <- rstandard(l.modelvi)
```

```
print(head(standard_res_modelvi))
```

```
##          1          2          3          4          5          6
## -0.3220307 -0.4249934 -0.4576951 -0.4981434 -0.5537557 -0.2804467
```

```
#column bind standardized residuals back to original data frame
final_data_modelvi <- cbind(housing_upd_df, standard_res_modelvi)

#sort standardized residuals descending
head(final_data_modelvi[order(-standard_res_modelvi),])
```

```
##       Sale Date Sale Price sale_reason sale_instrument sale_warning sitetype
## 1356 2011-11-17    4380542           1              22        11 45       R1
## 1357 2011-11-17    4380542           1              22        11 45       R1
## 1358 2011-11-17    4380542           1              22        11 45       R1
## 1359 2011-11-17    4380542           1              22        11 45       R1
## 1360 2011-11-17    4380542           1              22        11 45       R1
## 1361 2011-11-17    4380542           1              22        11 45       R1
##               addr_full  zip5 ctyname postalctyn       lon      lat
## 1356  17137 NE 120TH ST 98052 REDMOND    REDMOND -122.1113 47.70674
## 1357  11818 171ST PL NE 98052 REDMOND    REDMOND -122.1119 47.70639
## 1358 17011 NE 118TH WAY 98052 REDMOND    REDMOND -122.1134 47.70580
## 1359 16943 NE 118TH WAY 98052 REDMOND    REDMOND -122.1138 47.70579
## 1360 16944 NE 118TH WAY 98052 REDMOND    REDMOND -122.1138 47.70624
## 1361  16909 NE 120TH ST 98052 REDMOND    REDMOND -122.1145 47.70694
##      building_grade square_feet_total_living bedrooms bath_full_count
## 1356              8                     3290        4               2
## 1357              8                     2450        4               2
## 1358              8                     2750        4               2
## 1359              8                     3010        4               2
## 1360              8                     3200        5               2
## 1361              8                     3200        5               2
##      bath_half_count bath_3qtr_count year_built year_renovated current_zoning
## 1356               1               0       2012              0            R4
## 1357               1               0       2010              0            R4
## 1358               1               0       2012              0            R4
## 1359               0               1       2012              0            R4
## 1360               1               0       2010              0            R4
## 1361               1               0       2012              0            R4
##      sq_ft_lot prop_type present_use standard_res_modelvi
## 1356      6712         R           2             7.575585
## 1357      4749         R           2             7.575585
## 1358      5816         R           2             7.575585
## 1359      8908         R           2             7.575585
## 1360      4584         R           2             7.575585
## 1361      4681         R           2             7.575585
```

```
## 3.b.ix)  Use the appropriate function to show the sum of large residuals.

sum(l.model1$residuals^2)
```

```
## [1] 6.040964e+14
```

```
sum(l.model2$residuals^2)
```

```
## [1] 6.093294e+14
```

```
sum(l.model3$residuals^2)
```

```
## [1] 5.46896e+14
```

```
sum(l.model4$residuals^2)
```

```
## [1] 6.272179e+14
```

```
sum(l.model5$residuals^2)
```

```
## [1] 5.90942e+14
```

```
sum(l.model6$residuals^2)
```

```
## [1] 5.745504e+14
```

```
## 3.b.x)  Which specific variables have large residuals (only cases that evaluate as TRUE)?
# All variables have a large residuals but Model4 and Model2 have the highest.

## 3.b.xi)  Investigate further by calculating the leverage, cooks distance, and covariance rations. Co
summary(price_per_sq_ft_df2)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot + year_built + square_feet_total_living +
##     zip5 + bedrooms + current_zoning, data = housing_upd_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1139054  -134213   -42744    35934  3660721
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.954e+08  4.283e+08  -0.456   0.6484
## sq_ft_lot                  7.587e+00  2.402e+00   3.158   0.0016 **
## year_built                 1.507e+04  2.297e+03   6.559 6.50e-11 ***
## square_feet_total_living   1.738e+02  1.524e+01  11.407  < 2e-16 ***
## zip5                       1.685e+03  4.362e+03   0.386   0.6993
## bedrooms                   1.745e+04  1.495e+04   1.167   0.2433
## current_zoningR1          -7.248e+04  1.199e+05  -0.604   0.5457
## current_zoningR12          9.351e+04  7.240e+04   1.292   0.1966
## current_zoningR18         -2.624e+04  1.189e+05  -0.221   0.8254
## current_zoningR3           1.224e+05  1.344e+05   0.910   0.3627
## current_zoningR4           6.272e+04  6.750e+04   0.929   0.3529
## current_zoningR4/C         4.722e+04  8.322e+04   0.567   0.5705
## current_zoningR5          -2.032e+04  7.036e+04  -0.289   0.7727
## current_zoningR6          -9.407e+03  7.303e+04  -0.129   0.8975
## current_zoningR6/C         5.134e+05  1.130e+05   4.543 5.81e-06 ***
## current_zoningR8           7.612e+05  1.059e+05   7.186 8.65e-13 ***
## current_zoningRA5          3.462e+05  4.461e+05   0.776   0.4379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 440400 on 2636 degrees of freedom
## Multiple R-squared:  0.1851, Adjusted R-squared:  0.1802
## F-statistic: 37.43 on 16 and 2636 DF,  p-value: < 2.2e-16
```

```r
#calculate leverage for each observation in the model
hats1 <- as.data.frame(hatvalues(price_per_sq_ft_df2))

#display leverage stats for each observation
head(hats1)
```

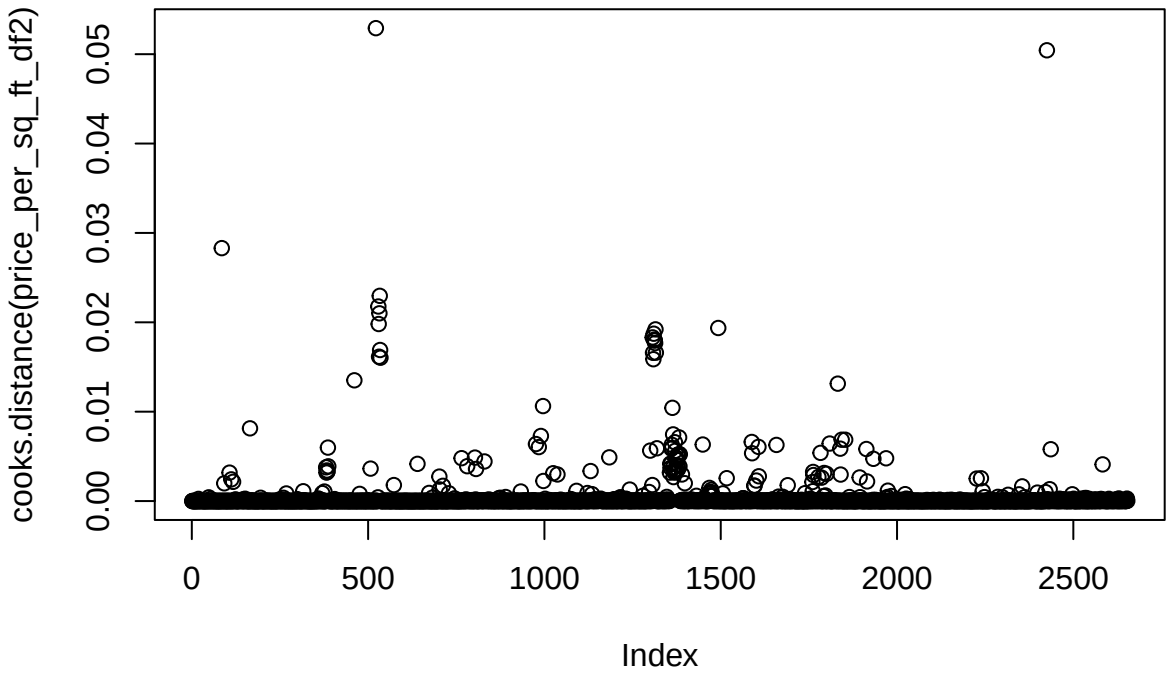
```
##   hatvalues(price_per_sq_ft_df2)
## 1                   0.0016623099
## 2                   0.0009042801
## 3                   0.0043628768
## 4                   0.0042947167
## 5                   0.0013181693
## 6                   0.0021811043
```

```r
#sort observations by leverage, descending
head(hats1[order(-hats1['hatvalues(price_per_sq_ft_df2)']), ])
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

```
## [1] 1.0000000 0.2051158 0.1356392 0.1356392 0.1356392 0.1034383
```

```r
# We can see that the largest leverage value is 1.0. Since this isn't greater than 2, we know that none

# Cooks distance
plot(cooks.distance(price_per_sq_ft_df2))
```




```r
# covariance ratio's
cov(housing_upd_df$`Sale Price`,housing_upd_df$sq_ft_lot, method = "pearson")
```

```
## [1] 422692406
```

```r
cov(housing_upd_df$`Sale Price`,housing_upd_df$year_built, method = "pearson")
```

```
## [1] 335798.4
```

```r
cov(housing_upd_df$`Sale Price`,housing_upd_df$square_feet_total_living, method = "pearson")
```

```
## [1] 136549639
```

```r
cov(housing_upd_df$`Sale Price`,housing_upd_df$zip5, method = "pearson")
```

```
## [1] 17147.09
```

```r
## 3.b.xii)  Perform the necessary calculations to assess the assumption of independence and state if t

chisq.test(table(housing_upd_df$`Sale Price`,housing_upd_df$sq_ft_lot))
```

```
## Warning in chisq.test(table(housing_upd_df$`Sale Price`,
## housing_upd_df$sq_ft_lot)): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(housing_upd_df$`Sale Price`, housing_upd_df$sq_ft_lot)
## X-squared = 2173950, df = 2064120, p-value < 2.2e-16
```

```r
chisq.test(table(housing_upd_df$`Sale Price`,housing_upd_df$year_built))
```

```
## Warning in chisq.test(table(housing_upd_df$`Sale Price`,
## housing_upd_df$year_built)): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(housing_upd_df$`Sale Price`, housing_upd_df$year_built)
## X-squared = 22066, df = 20040, p-value < 2.2e-16
```

```r
chisq.test(table(housing_upd_df$`Sale Price`,housing_upd_df$square_feet_total_living))
```

```
## Warning in chisq.test(table(housing_upd_df$`Sale Price`,
## housing_upd_df$square_feet_total_living)): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(housing_upd_df$`Sale Price`, housing_upd_df$square_feet_total_living)
## X-squared = 493693, df = 450232, p-value < 2.2e-16
```

```r
chisq.test(table(housing_upd_df$`Sale Price`,housing_upd_df$bedrooms))
```

```
## Warning in chisq.test(table(housing_upd_df$`Sale Price`,
## housing_upd_df$bedrooms)): Chi-squared approximation may be incorrect
```

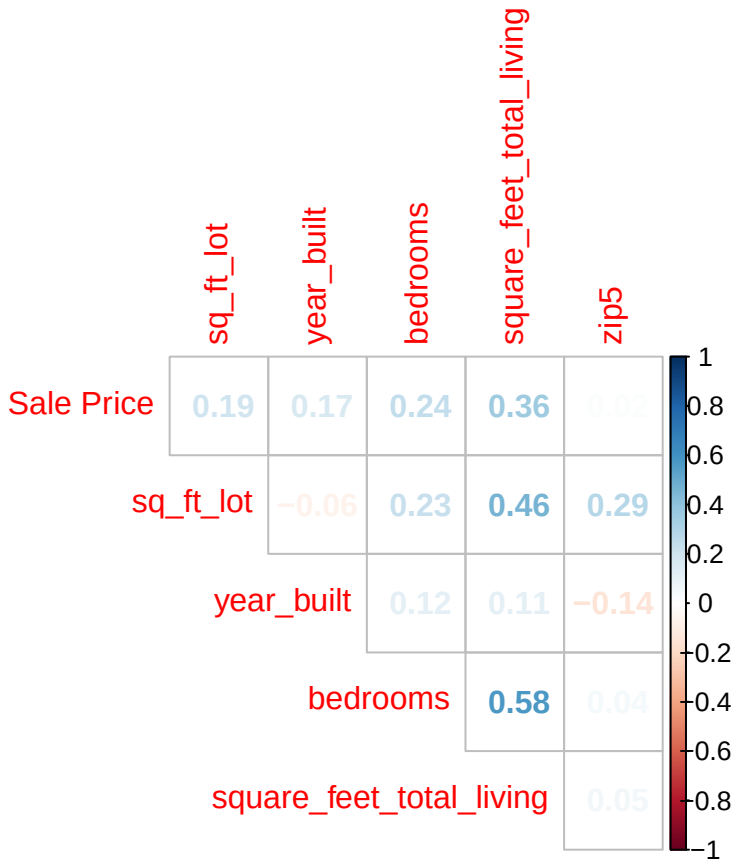
```
##
##  Pearson's Chi-squared test
##
## data:  table(housing_upd_df$`Sale Price`, housing_upd_df$bedrooms)
## X-squared = 14701, df = 12024, p-value < 2.2e-16
```

```r
chisq.test(table(housing_upd_df$`Sale Price`,housing_upd_df$zip5))
```

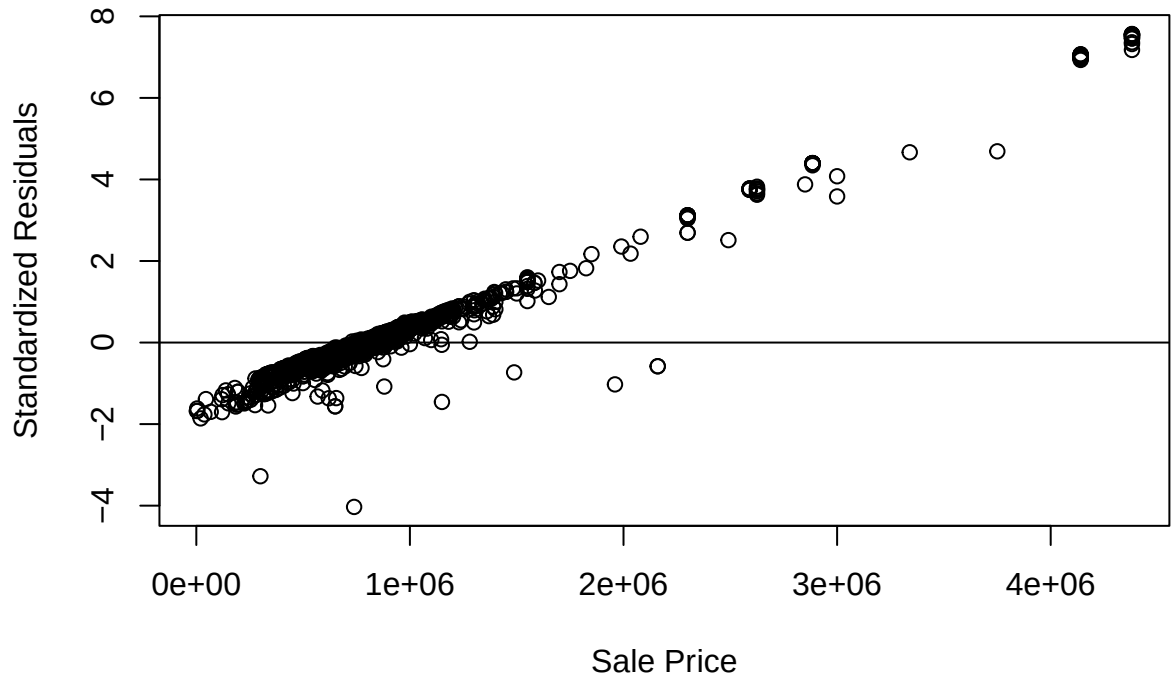
```
## Warning in chisq.test(table(housing_upd_df$`Sale Price`, housing_upd_df$zip5)):
## Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(housing_upd_df$`Sale Price`, housing_upd_df$zip5)
## X-squared = 1611, df = 1336, p-value = 2.788e-07
```

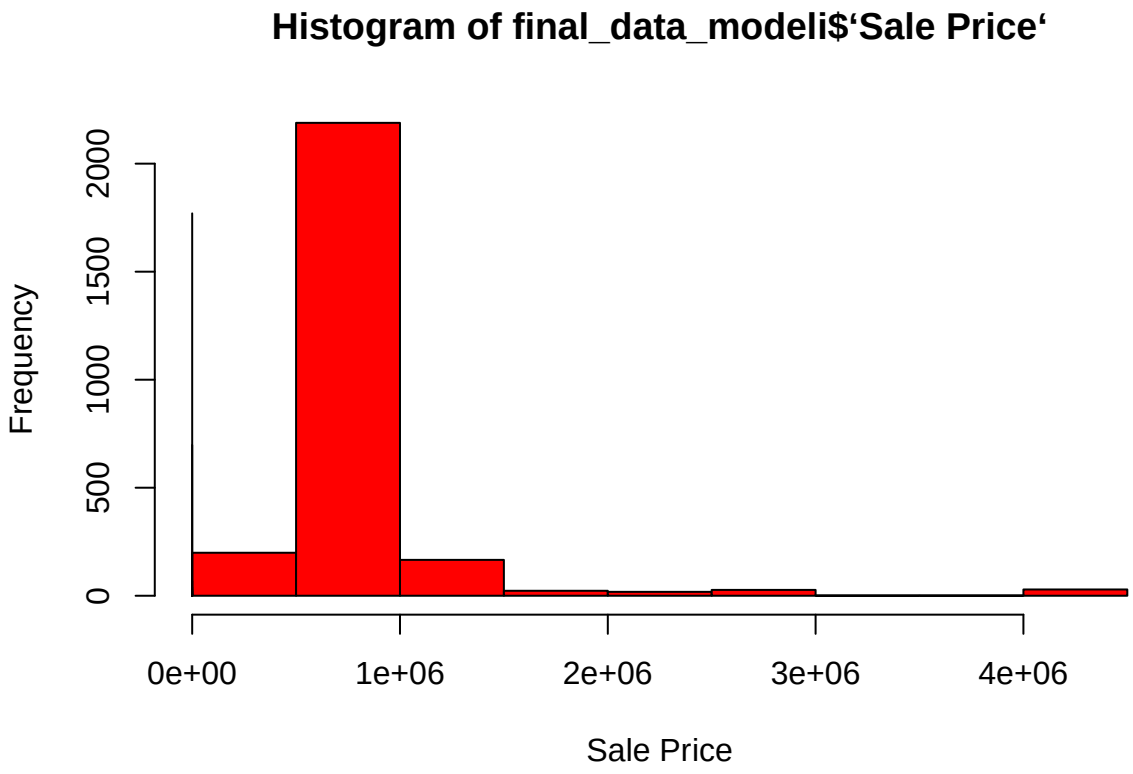
```
# Since "Chi-squared approximation may be incorrect" appears, it means that the smallest expected frequ

## 3.b.xiii)  Perform the necessary calculations to assess the assumption of no multicollinearity and s

housing_cor_df <- housing_upd_df %>% select(`Sale Price` ,sq_ft_lot, year_built,bedrooms,square_feet_to

corrplot(cor(housing_cor_df), method = "number", type = "upper", diag = FALSE)
```



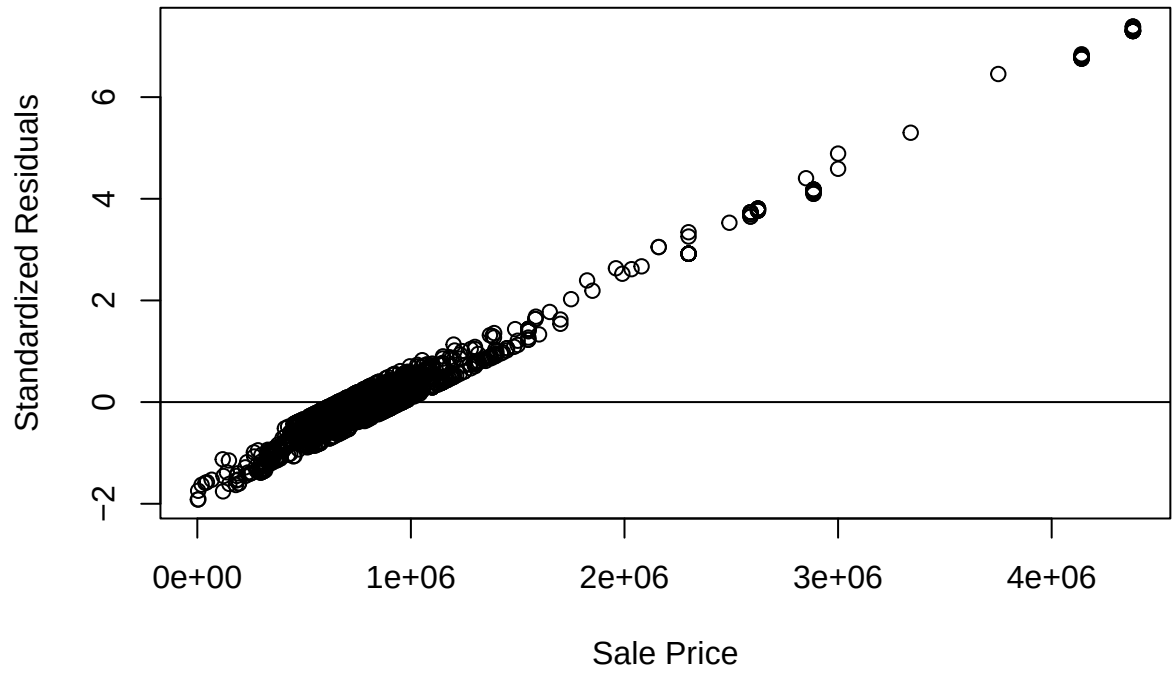

```
## 3.b.xiv)  Visually check the assumptions related to the residuals using the plot() and hist() functi
## Summarize what each graph is informing you of and if any anomalies are present.

#plot predictor variable1 vs. standardized residuals
plot(final_data_modeli$`Sale Price`, standard_res_modeli, ylab='Standardized Residuals', xlab='Sale Pri
#add horizontal line at 0
abline(0, 0)
```



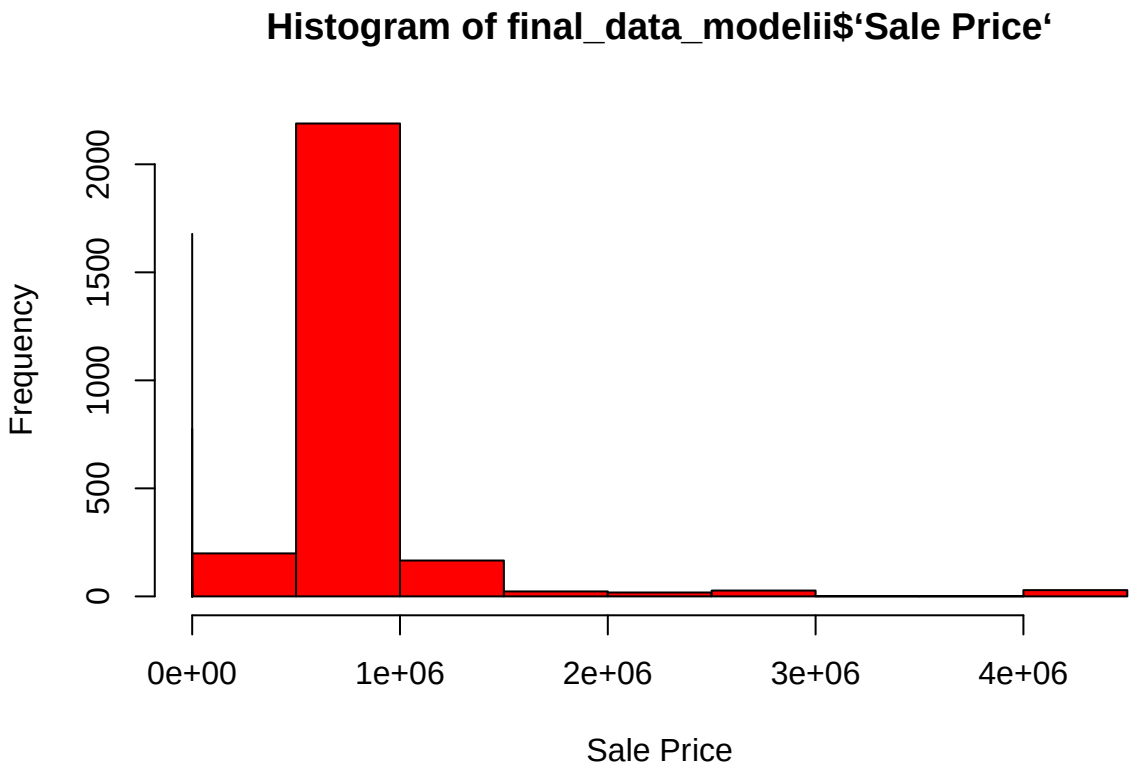

```
#Using hist() function
hist(final_data_modeli$`Sale Price`, col = "red", xlab = "Sale Price")
hist(standard_res_modeli,col= "blue", add = TRUE)
```



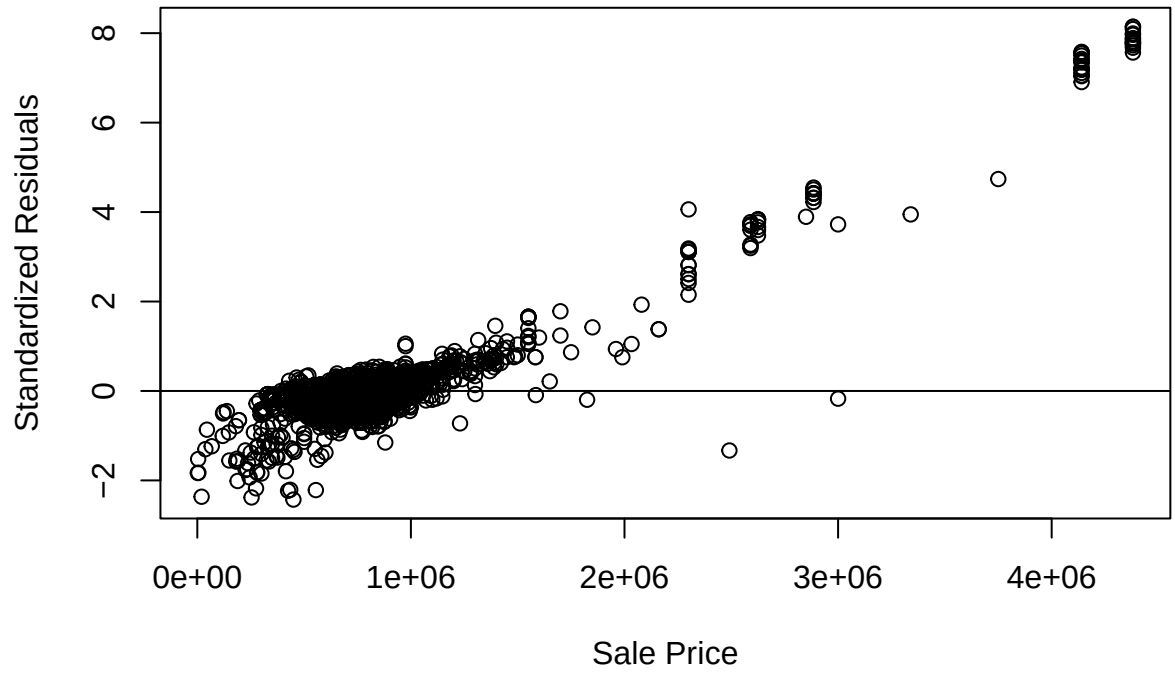

```
#plot predictor variable2 vs. standardized residuals
plot(final_data_modelii$`Sale Price`, standard_res_modelii, ylab='Standardized Residuals', xlab='Sale Pr
#add horizontal line at 0
abline(0, 0)
```



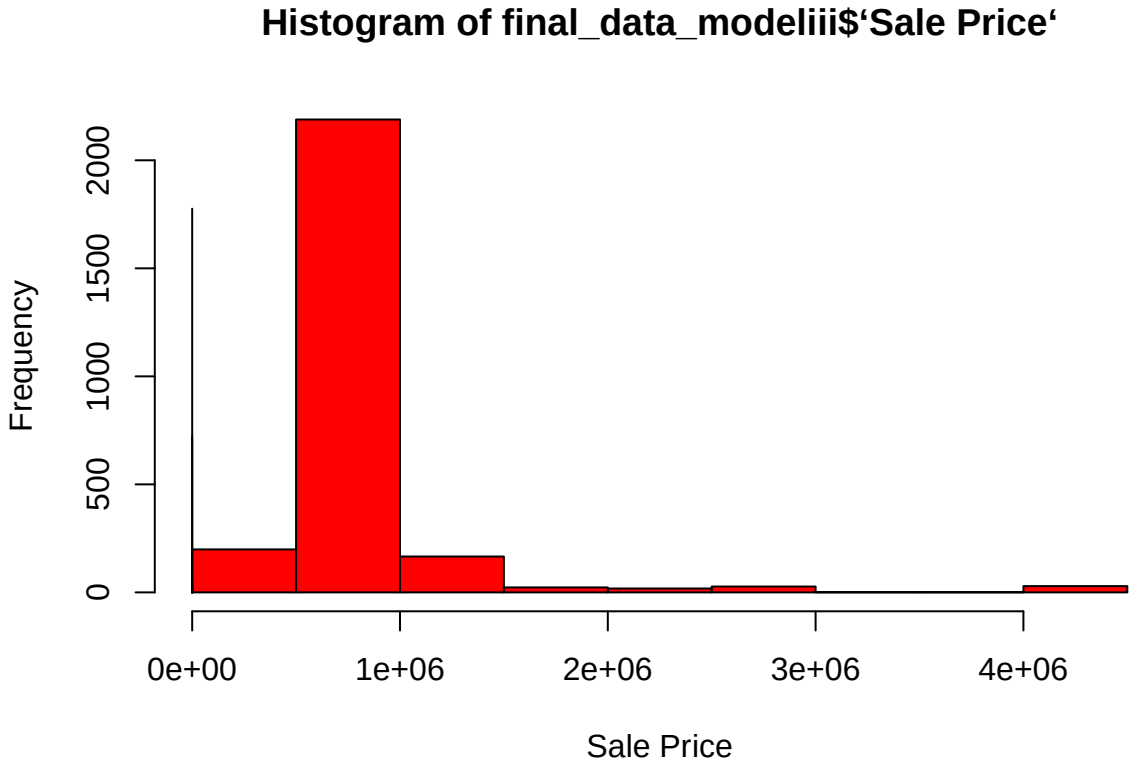

```
#Using hist() function
hist(final_data_modelii$`Sale Price`, col = "red", xlab = "Sale Price")
hist(standard_res_modelii,col= "blue", add = TRUE)
```



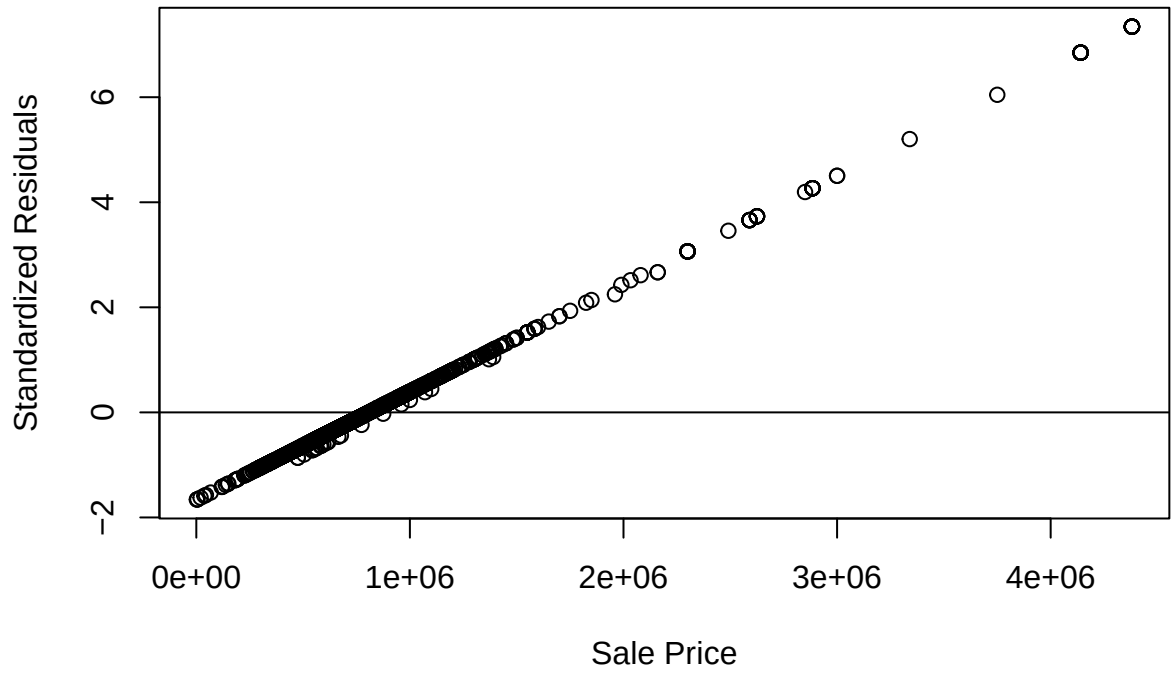

```
#plot predictor variable3 vs. standardized residuals
plot(final_data_modeliii$`Sale Price`, standard_res_modeliii, ylab='Standardized Residuals', xlab='Sale
#add horizontal line at 0
abline(0, 0)
```



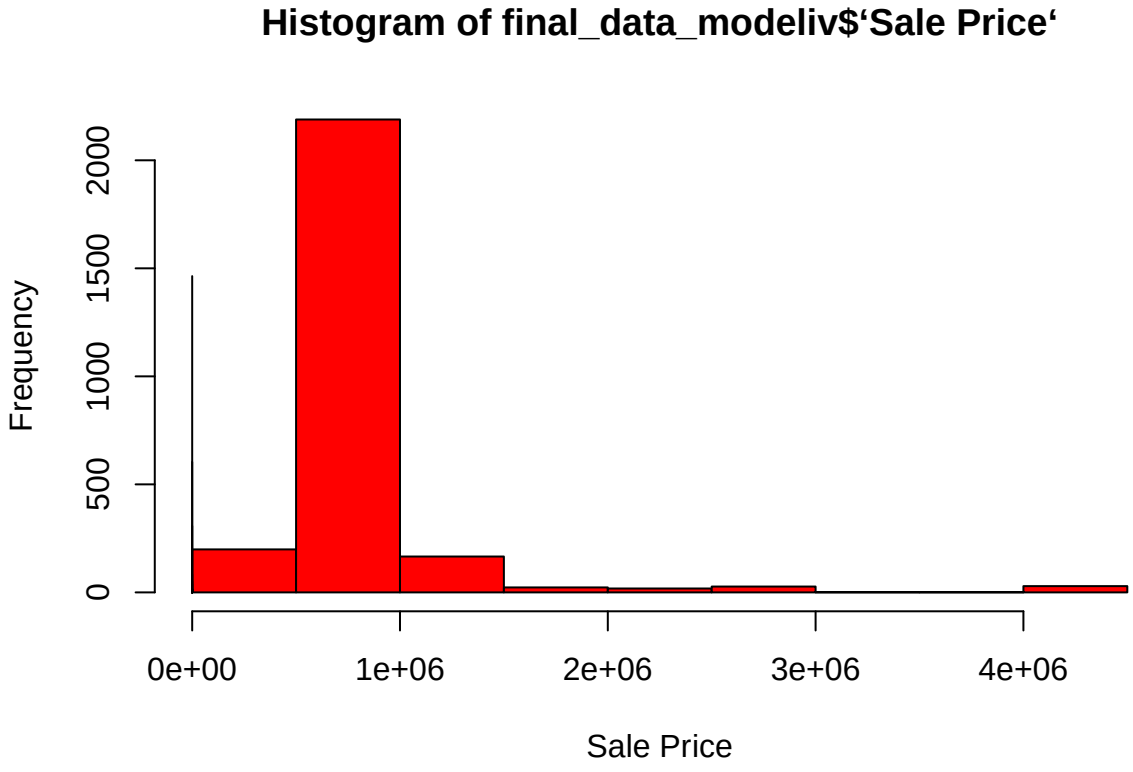

```
#Using hist() function
hist(final_data_modeliii$`Sale Price`, col = "red", xlab = "Sale Price")
hist(standard_res_modeliii,col= "blue", add = TRUE)
```



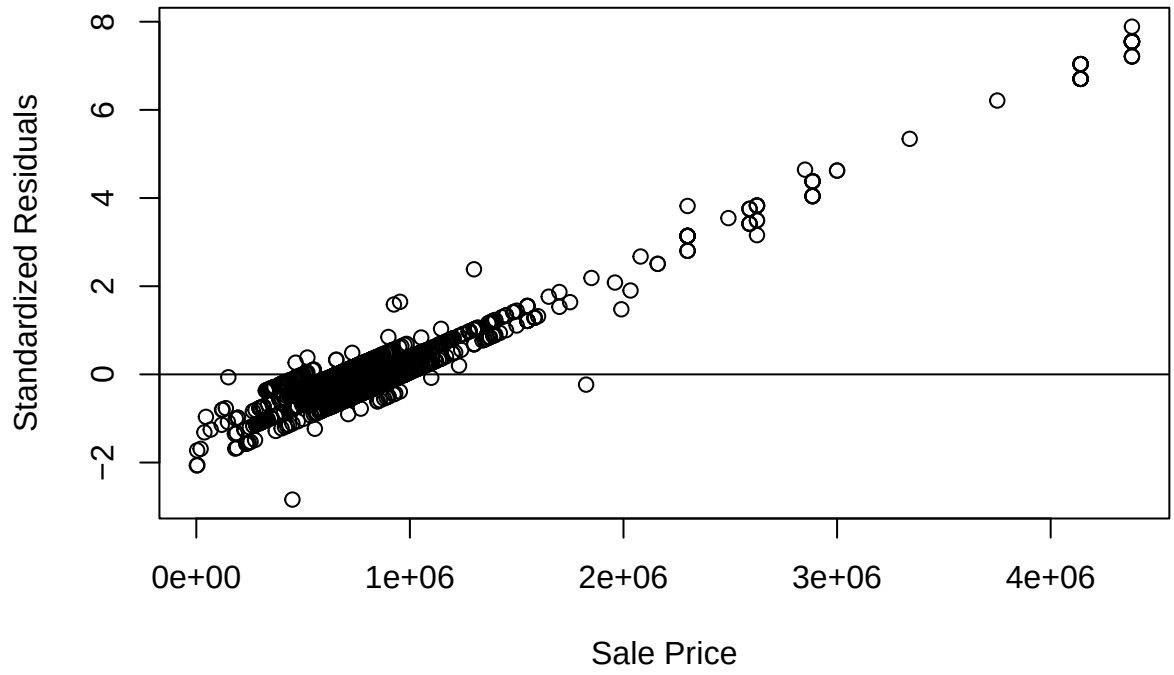

```
#plot predictor variable4 vs. standardized residuals
plot(final_data_modeliv$`Sale Price`, standard_res_modeliv, ylab='Standardized Residuals', xlab='Sale P
#add horizontal line at 0
abline(0, 0)
```



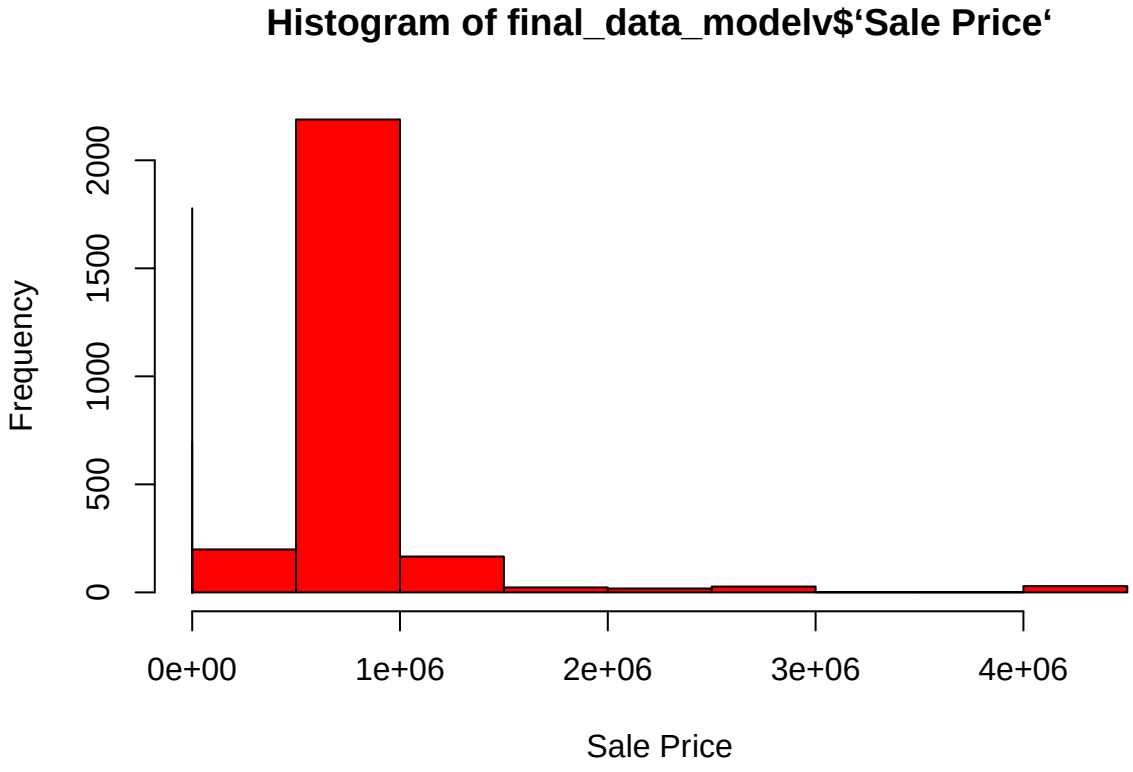

```
#Using hist() function
hist(final_data_modeliv$`Sale Price`, col = "red", xlab = "Sale Price")
hist(standard_res_modeliv,col= "blue", add = TRUE)
```



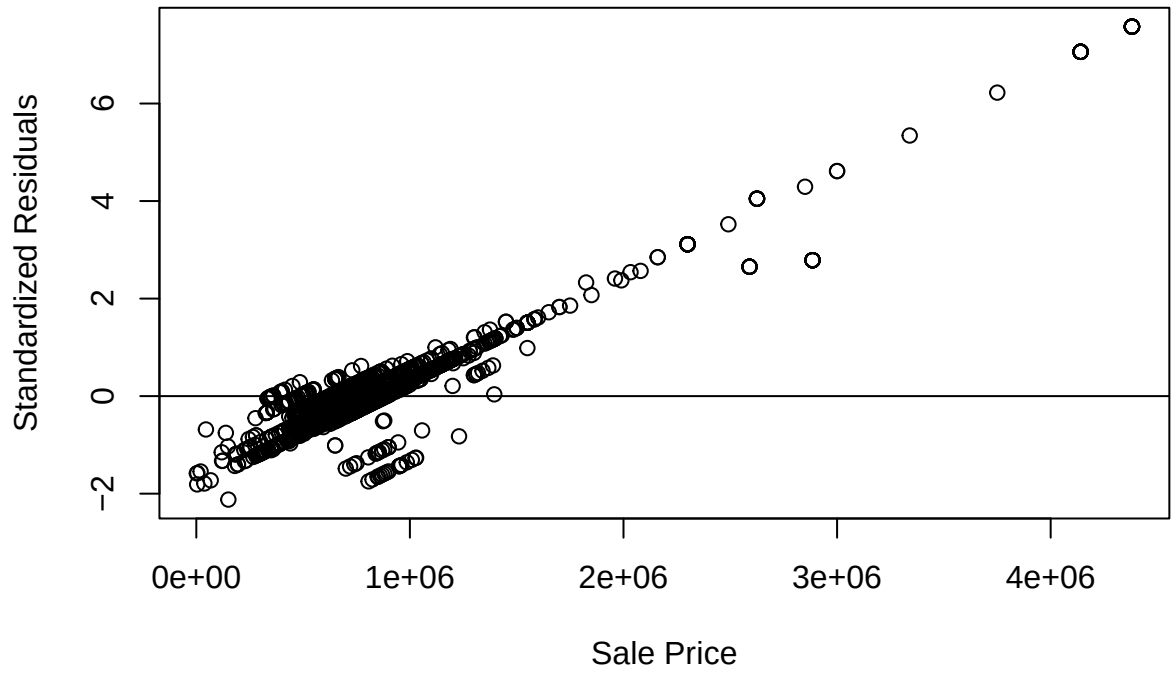

```
#plot predictor variable5 vs. standardized residuals
plot(final_data_modelv$`Sale Price`, standard_res_modelv, ylab='Standardized Residuals', xlab='Sale Pri
#add horizontal line at 0
abline(0, 0)
```



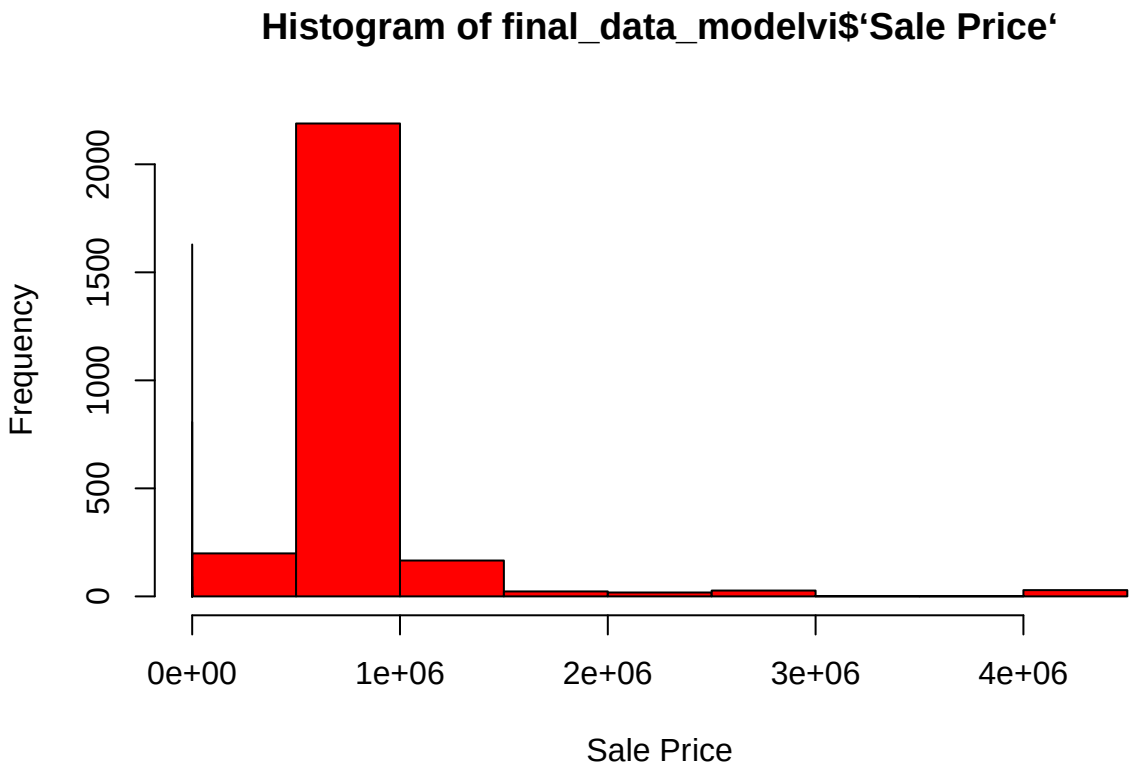

```
#Using hist() function
hist(final_data_modelv$`Sale Price`, col = "red", xlab = "Sale Price")
hist(standard_res_modelv,col= "blue", add = TRUE)
```




```
#plot predictor variable6 vs. standardized residuals
plot(final_data_modelvi$`Sale Price`, standard_res_modelvi, ylab='Standardized Residuals', xlab='Sale P
#add horizontal line at 0
abline(0, 0)
```




```
#Using hist() function
hist(final_data_modelvi$`Sale Price`, col = "red", xlab = "Sale Price")
hist(standard_res_modelvi,col= "blue", add = TRUE)
```




```
## 3.b.xv)  Overall, is this regression model unbiased?
## If an unbiased regression model, what does this tell us about the sample vs. the entire population m
# Yes, the model seems to be unbiased based on the estimated values but different types of sample popul
```