MSDS696\_Practicum\_II\_Project\_Markdown\_File

Odes Mullins

12/15/2019

## Practicum II Project Douglas County Home Sales Prices Evaluation and Prediction

### Load required libraries

library(readr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

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

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

library(maps)  
library(rgdal)

## Loading required package: sp

## rgdal: version: 1.4-8, (SVN revision 845)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 2.2.3, released 2017/11/20  
## Path to GDAL shared files: C:/Users/Owner/Documents/R/win-library/3.5/rgdal/gdal  
## GDAL binary built with GEOS: TRUE   
## Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ\_VERSION: 493]  
## Path to PROJ.4 shared files: C:/Users/Owner/Documents/R/win-library/3.5/rgdal/proj  
## Linking to sp version: 1.3-2

library(ggplot2)  
library(scales)

##   
## Attaching package: 'scales'

## The following object is masked from 'package:readr':  
##   
## col\_factor

library(tidyverse)

## -- Attaching packages ------------------------------------------------- tidyverse 1.3.0 --

## v tibble 2.1.3 v stringr 1.4.0  
## v tidyr 1.0.0 v forcats 0.4.0  
## v purrr 0.3.3

## -- Conflicts ---------------------------------------------------- tidyverse\_conflicts() --  
## x scales::col\_factor() masks readr::col\_factor()  
## x purrr::discard() masks scales::discard()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x purrr::map() masks maps::map()

library(tidyquant)

## Loading required package: lubridate

##   
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':  
##   
## date

## Loading required package: PerformanceAnalytics

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'xts'

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

##   
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':  
##   
## legend

## Loading required package: quantmod

## Loading required package: TTR

## Version 0.4-0 included new data defaults. See ?getSymbols.

## == Need to Learn tidyquant? ==============================================================  
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio Optimization with tidyquant!  
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>

library(GGally)

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':  
##   
## nasa

library(car)

## 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

library(skimr)  
library(class)  
library(gmodels)  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(ranger)  
library(DMwR)

## Loading required package: grid

library(gbm)

## Loaded gbm 2.1.5

library(xgboost)

##   
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':  
##   
## slice

library(e1071)

##   
## Attaching package: 'e1071'

## The following objects are masked from 'package:PerformanceAnalytics':  
##   
## kurtosis, skewness

### Fix number display, set seed, and set the working directory

options(scipen = 999)  
set.seed(123)  
setwd('C:\\Users\\Owner\\Documents\\MSDS696 Class Info\\Class\_Assignments\\Practicum\_II')

### Load the Douglas County Home Sales Report dataset with the column names and evaluate it

dcSales <- data.frame(read\_csv('SalesInformationReport0101201412312018.csv',col\_names = TRUE), check.names=FALSE)

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## `State Parcel #` = col\_double(),  
## `Situs Zip Code` = col\_double(),  
## `Reception #` = col\_double(),  
## `Sale Price` = col\_double(),  
## Acres = col\_double()  
## )

## See spec(...) for full column specifications.

head(dcSales)

## State Parcel # Account # Owner Name  
## 1 235304104002 R0000553 KEVIN CASHMAN & KIM CASHMAN  
## 2 222933405002 R0000625 JENNIFER M BRADY & MICHAEL L WEICKUM  
## 3 235304107010 R0000684 STUART M HUNGSBERG & KRISTINE A HUNGSBERG  
## 4 222932003001 R0000895 GARRY BRAUER TRUST  
## 5 222932002011 R0001118 J CALVIN DOWNS & MELAINE V DOWNS  
## 6 235304110013 R0001177 REBECCA CONNET & DARIN CONNET  
## Owner Street Address Owner Street Address 2 Owner City Owner State  
## 1 7927 ELM ST null LOUVIERS CO  
## 2 7994 MAIN ST null LOUVIERS CO  
## 3 7819 VALLEY VIEW ST null LOUVIERS CO  
## 4 8312 W TRAIL SOUTH DR null LITTLETON CO  
## 5 7583 W TRAIL SOUTH DR null LITTLETON CO  
## 6 PO BOX 14 null LOUVIERS CO  
## Owner Zip Code Situs Street Address Situs City Situs Zip Code  
## 1 80131 7927 ELM ST LOUVIERS 80131  
## 2 80131 7994 MAIN ST LOUVIERS 80131  
## 3 80131 7819 VALLEY VIEW ST LOUVIERS 80131  
## 4 80125-9514 8312 W TRAIL SOUTH DR LITTLETON 80125  
## 5 80125-9511 7583 W TRAIL SOUTH DR LITTLETON 80125  
## 6 80131-0014 6405 FOURTH ST LOUVIERS 80131  
## Account Type Deed Type Reception # Sale Date Sale Price  
## 1 Residential Warranty Deed Joint 2016036192 5/27/2016 298600  
## 2 Residential Warranty Deed Joint 2014058571 10/6/2014 235000  
## 3 Residential Warranty Deed Joint 2018021226 4/10/2018 554100  
## 4 Residential Warranty Deed 2016093954 12/20/2016 0  
## 5 Residential Warranty Deed Joint 2015043873 6/19/2015 634900  
## 6 Residential Warranty Deed Joint 2014029143 6/3/2014 326000  
## Grantor  
## 1 DAVID W DULAIGH & JANET P DULAIGH  
## 2 JOCELYN MERTENS  
## 3 WESLEY A RIBER & JULIE L RIBER  
## 4 GARRY L BRAUER  
## 5 MCCORMICK REVOCABLE TRUST  
## 6 PATRICK WOODWARD & DELBERT WOODWARD  
## Grantee Book Page Acres  
## 1 KEVIN CASHMAN & KIM CASHMAN null null 0.253  
## 2 JENNIFER M BRADY & MICHAEL L WEICKUM null null 0.227  
## 3 STUART M HUNGSBERG & KRISTINE A HUNGSBERG null null 0.386  
## 4 GARRY BRAUER TRUST null null 4.010  
## 5 J CALVIN DOWNS & MELAINE V DOWNS null null 4.780  
## 6 REBECCA CONNET & DARIN CONNET null null 0.556

str(dcSales)

## 'data.frame': 31587 obs. of 21 variables:  
## $ State Parcel # : num 235304104002 222933405002 235304107010 222932003001 222932002011 ...  
## $ Account # : chr "R0000553" "R0000625" "R0000684" "R0000895" ...  
## $ Owner Name : chr "KEVIN CASHMAN & KIM CASHMAN" "JENNIFER M BRADY & MICHAEL L WEICKUM" "STUART M HUNGSBERG & KRISTINE A HUNGSBERG" "GARRY BRAUER TRUST" ...  
## $ Owner Street Address : chr "7927 ELM ST" "7994 MAIN ST" "7819 VALLEY VIEW ST" "8312 W TRAIL SOUTH DR" ...  
## $ Owner Street Address 2: chr "null" "null" "null" "null" ...  
## $ Owner City : chr "LOUVIERS" "LOUVIERS" "LOUVIERS" "LITTLETON" ...  
## $ Owner State : chr "CO" "CO" "CO" "CO" ...  
## $ Owner Zip Code : chr "80131" "80131" "80131" "80125-9514" ...  
## $ Situs Street Address : chr "7927 ELM ST" "7994 MAIN ST" "7819 VALLEY VIEW ST" "8312 W TRAIL SOUTH DR" ...  
## $ Situs City : chr "LOUVIERS" "LOUVIERS" "LOUVIERS" "LITTLETON" ...  
## $ Situs Zip Code : num 80131 80131 80131 80125 80125 ...  
## $ Account Type : chr "Residential" "Residential" "Residential" "Residential" ...  
## $ Deed Type : chr "Warranty Deed Joint" "Warranty Deed Joint" "Warranty Deed Joint" "Warranty Deed" ...  
## $ Reception # : num 2016036192 2014058571 2018021226 2016093954 2015043873 ...  
## $ Sale Date : chr "5/27/2016" "10/6/2014" "4/10/2018" "12/20/2016" ...  
## $ Sale Price : num 298600 235000 554100 0 634900 ...  
## $ Grantor : chr "DAVID W DULAIGH & JANET P DULAIGH" "JOCELYN MERTENS" "WESLEY A RIBER & JULIE L RIBER" "GARRY L BRAUER" ...  
## $ Grantee : chr "KEVIN CASHMAN & KIM CASHMAN" "JENNIFER M BRADY & MICHAEL L WEICKUM" "STUART M HUNGSBERG & KRISTINE A HUNGSBERG" "GARRY BRAUER TRUST" ...  
## $ Book : chr "null" "null" "null" "null" ...  
## $ Page : chr "null" "null" "null" "null" ...  
## $ Acres : num 0.253 0.227 0.386 4.01 4.78 ...

dim(dcSales)

## [1] 31587 21

glimpse(dcSales)

## Observations: 31,587  
## Variables: 21  
## $ `State Parcel #` <dbl> 235304104002, 222933405002, 235304107...  
## $ `Account #` <chr> "R0000553", "R0000625", "R0000684", "...  
## $ `Owner Name` <chr> "KEVIN CASHMAN & KIM CASHMAN", "JENNI...  
## $ `Owner Street Address` <chr> "7927 ELM ST", "7994 MAIN ST", "7819 ...  
## $ `Owner Street Address 2` <chr> "null", "null", "null", "null", "null...  
## $ `Owner City` <chr> "LOUVIERS", "LOUVIERS", "LOUVIERS", "...  
## $ `Owner State` <chr> "CO", "CO", "CO", "CO", "CO", "CO", "...  
## $ `Owner Zip Code` <chr> "80131", "80131", "80131", "80125-951...  
## $ `Situs Street Address` <chr> "7927 ELM ST", "7994 MAIN ST", "7819 ...  
## $ `Situs City` <chr> "LOUVIERS", "LOUVIERS", "LOUVIERS", "...  
## $ `Situs Zip Code` <dbl> 80131, 80131, 80131, 80125, 80125, 80...  
## $ `Account Type` <chr> "Residential", "Residential", "Reside...  
## $ `Deed Type` <chr> "Warranty Deed Joint", "Warranty Deed...  
## $ `Reception #` <dbl> 2016036192, 2014058571, 2018021226, 2...  
## $ `Sale Date` <chr> "5/27/2016", "10/6/2014", "4/10/2018"...  
## $ `Sale Price` <dbl> 298600, 235000, 554100, 0, 634900, 32...  
## $ Grantor <chr> "DAVID W DULAIGH & JANET P DULAIGH", ...  
## $ Grantee <chr> "KEVIN CASHMAN & KIM CASHMAN", "JENNI...  
## $ Book <chr> "null", "null", "null", "null", "null...  
## $ Page <chr> "null", "null", "null", "null", "null...  
## $ Acres <dbl> 0.253, 0.227, 0.386, 4.010, 4.780, 0....

### Clean the sales dataset

dcSalesClean <- subset(dcSales[order(dcSales$`Account #`, dcSales$`Sale Date`),], `Account Type` == "Residential")  
   
dcSalesClean <- subset(dcSalesClean, `Sale Price` > 149999)

### Load the Douglas County Property Improvement Segments Report dataset with the column names and evaluate it

dcPisr <- data.frame(read\_csv('PropertyImprovementSegmentsReport0101201412312018.csv',col\_names = TRUE), check.names=FALSE)

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## `State Parcel #` = col\_double(),  
## `Building #` = col\_double(),  
## `Completion Percentage` = col\_double(),  
## `Improvement SF` = col\_double(),  
## `Year Built` = col\_double(),  
## Stories = col\_double(),  
## `Bedroom Count` = col\_double(),  
## `Bathroom Count` = col\_double(),  
## `Built As SF` = col\_double()  
## )

## See spec(...) for full column specifications.

## Warning: 2 parsing failures.  
## row col expected actual file  
## 26735 Bedroom Count a double null 'PropertyImprovementSegmentsReport0101201412312018.csv'  
## 26735 Bathroom Count a double null 'PropertyImprovementSegmentsReport0101201412312018.csv'

head(dcPisr)

## State Parcel # Account # Owner Name  
## 1 235304104002 R0000553 KEVIN CASHMAN & KIM CASHMAN  
## 2 222933405002 R0000625 JENNIFER M BRADY & MICHAEL L WEICKUM  
## 3 235304107010 R0000684 STUART M HUNGSBERG & KRISTINE A HUNGSBERG  
## 4 222932003001 R0000895 GARRY BRAUER TRUST  
## 5 222932002011 R0001118 J CALVIN DOWNS & MELAINE V DOWNS  
## 6 235304110013 R0001177 REBECCA CONNET & DARIN CONNET  
## Owner Street Address Owner Street Address 2 Owner City Owner State  
## 1 7927 ELM ST null LOUVIERS CO  
## 2 7994 MAIN ST null LOUVIERS CO  
## 3 7819 VALLEY VIEW ST null LOUVIERS CO  
## 4 8312 W TRAIL SOUTH DR null LITTLETON CO  
## 5 7583 W TRAIL SOUTH DR null LITTLETON CO  
## 6 PO BOX 14 null LOUVIERS CO  
## Owner Zip Code Property Type Building # Quality Completion Percentage  
## 1 80131 Residential 1 Average 1  
## 2 80131 Residential 1 Average 1  
## 3 80131 Residential 1 Average 1  
## 4 80125-9514 Residential 1 Average 1  
## 5 80125-9511 Residential 1 Average 1  
## 6 80131-0014 Residential 1 Average 1  
## Improvement SF Style Year Built Stories Bedroom Count  
## 1 1288 1 - Ranch 1 Story 1966 1 2  
## 2 1484 1 - Ranch 1 Story 1908 1 2  
## 3 2858 5 - 1 1/2 Story Fin 1911 2 4  
## 4 2200 1 - Ranch 1 Story 1972 1 3  
## 5 2678 8 - 2 Story 1976 2 3  
## 6 1462 5 - 1 1/2 Story Fin 1972 2 4  
## Bathroom Count Built As SF Roof Type Interior Type Exterior Type  
## 1 1 1288 Gable Drywall Frame Masonry Veneer  
## 2 2 1484 Gable Drywall Frame Siding  
## 3 2 2858 Gable Drywall Frame Siding  
## 4 2 2200 null Drywall Frame Masonry Veneer  
## 5 3 2678 Gable Drywall Frame Masonry Veneer  
## 6 2 1462 Gable Drywall Frame Stucco  
## Heat Type Roof Material Type Floor Type  
## 1 Hot Water Baseboard Composition Shingle Allowance  
## 2 Central Air to Air Composition Shingle Allowance  
## 3 Central Air to Air Composition Shingle Allowance  
## 4 Forced Air Wood Shake Allowance  
## 5 Central Air to Air Composition Shingle Allowance  
## 6 Central Air to Air Composition Shingle Allowance

str(dcPisr)

## 'data.frame': 26749 obs. of 25 variables:  
## $ State Parcel # : num 235304104002 222933405002 235304107010 222932003001 222932002011 ...  
## $ Account # : chr "R0000553" "R0000625" "R0000684" "R0000895" ...  
## $ Owner Name : chr "KEVIN CASHMAN & KIM CASHMAN" "JENNIFER M BRADY & MICHAEL L WEICKUM" "STUART M HUNGSBERG & KRISTINE A HUNGSBERG" "GARRY BRAUER TRUST" ...  
## $ Owner Street Address : chr "7927 ELM ST" "7994 MAIN ST" "7819 VALLEY VIEW ST" "8312 W TRAIL SOUTH DR" ...  
## $ Owner Street Address 2: chr "null" "null" "null" "null" ...  
## $ Owner City : chr "LOUVIERS" "LOUVIERS" "LOUVIERS" "LITTLETON" ...  
## $ Owner State : chr "CO" "CO" "CO" "CO" ...  
## $ Owner Zip Code : chr "80131" "80131" "80131" "80125-9514" ...  
## $ Property Type : chr "Residential" "Residential" "Residential" "Residential" ...  
## $ Building # : num 1 1 1 1 1 1 1 1 2 1 ...  
## $ Quality : chr "Average" "Average" "Average" "Average" ...  
## $ Completion Percentage : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ Improvement SF : num 1288 1484 2858 2200 2678 ...  
## $ Style : chr "1 - Ranch 1 Story" "1 - Ranch 1 Story" "5 - 1 1/2 Story Fin" "1 - Ranch 1 Story" ...  
## $ Year Built : num 1966 1908 1911 1972 1976 ...  
## $ Stories : num 1 1 2 1 2 2 1 2 1 2 ...  
## $ Bedroom Count : num 2 2 4 3 3 4 3 1 2 3 ...  
## $ Bathroom Count : num 1 2 2 2 3 2 3 2 2 3 ...  
## $ Built As SF : num 1288 1484 2858 2200 2678 ...  
## $ Roof Type : chr "Gable" "Gable" "Gable" "null" ...  
## $ Interior Type : chr "Drywall" "Drywall" "Drywall" "Drywall" ...  
## $ Exterior Type : chr "Frame Masonry Veneer" "Frame Siding" "Frame Siding" "Frame Masonry Veneer" ...  
## $ Heat Type : chr "Hot Water Baseboard" "Central Air to Air" "Central Air to Air" "Forced Air" ...  
## $ Roof Material Type : chr "Composition Shingle" "Composition Shingle" "Composition Shingle" "Wood Shake" ...  
## $ Floor Type : chr "Allowance" "Allowance" "Allowance" "Allowance" ...

dim(dcPisr)

## [1] 26749 25

glimpse(dcPisr)

## Observations: 26,749  
## Variables: 25  
## $ `State Parcel #` <dbl> 235304104002, 222933405002, 235304107...  
## $ `Account #` <chr> "R0000553", "R0000625", "R0000684", "...  
## $ `Owner Name` <chr> "KEVIN CASHMAN & KIM CASHMAN", "JENNI...  
## $ `Owner Street Address` <chr> "7927 ELM ST", "7994 MAIN ST", "7819 ...  
## $ `Owner Street Address 2` <chr> "null", "null", "null", "null", "null...  
## $ `Owner City` <chr> "LOUVIERS", "LOUVIERS", "LOUVIERS", "...  
## $ `Owner State` <chr> "CO", "CO", "CO", "CO", "CO", "CO", "...  
## $ `Owner Zip Code` <chr> "80131", "80131", "80131", "80125-951...  
## $ `Property Type` <chr> "Residential", "Residential", "Reside...  
## $ `Building #` <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1...  
## $ Quality <chr> "Average", "Average", "Average", "Ave...  
## $ `Completion Percentage` <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...  
## $ `Improvement SF` <dbl> 1288, 1484, 2858, 2200, 2678, 1462, 2...  
## $ Style <chr> "1 - Ranch 1 Story", "1 - Ranch 1 Sto...  
## $ `Year Built` <dbl> 1966, 1908, 1911, 1972, 1976, 1972, 1...  
## $ Stories <dbl> 1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 1, 1...  
## $ `Bedroom Count` <dbl> 2, 2, 4, 3, 3, 4, 3, 1, 2, 3, 3, 3, 2...  
## $ `Bathroom Count` <dbl> 1, 2, 2, 2, 3, 2, 3, 2, 2, 3, 2, 1, 1...  
## $ `Built As SF` <dbl> 1288, 1484, 2858, 2200, 2678, 1462, 2...  
## $ `Roof Type` <chr> "Gable", "Gable", "Gable", "null", "G...  
## $ `Interior Type` <chr> "Drywall", "Drywall", "Drywall", "Dry...  
## $ `Exterior Type` <chr> "Frame Masonry Veneer", "Frame Siding...  
## $ `Heat Type` <chr> "Hot Water Baseboard", "Central Air t...  
## $ `Roof Material Type` <chr> "Composition Shingle", "Composition S...  
## $ `Floor Type` <chr> "Allowance", "Allowance", "Allowance"...

### Clean the improvement segments dataset

dcPisrClean <- subset(dcPisr[order(dcPisr$`Account #`),], `Property Type` == "Residential")  
   
dcPisrDup1 = !duplicated(dcPisr[,1:2])  
dcPisrDup2 = !duplicated(apply(dcPisr[,1:2], 2, rev))  
dcPisrClean <- dcPisr[dcPisrDup1 & rev(dcPisrDup2), ]  
   
dcPisrClean$Style[dcPisrClean$Style == '1 - Ranch 1 Story'] <- 1  
dcPisrClean$Style[dcPisrClean$Style == '11 - A Frame'] <- 11  
dcPisrClean$Style[dcPisrClean$Style == '25 - 3 Story'] <- 25  
dcPisrClean$Style[dcPisrClean$Style == '5 - 1 1/2 Story Fin'] <- 5  
dcPisrClean$Style[dcPisrClean$Style == '8 - 2 Story'] <- 8  
dcPisrClean$Style[dcPisrClean$Style == '9 - 2 1/2 Story'] <- 9  
   
dcPisrClean$Style <- as.double(dcPisrClean$Style)

### Load the Douglas County Building Summary Report dataset with the column names and evaluate it

dcBuildSum <- data.frame(read\_csv('BuildingSummaryReport0101201412312018.csv',col\_names = TRUE), check.names=FALSE)

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## `State Parcel #` = col\_double(),  
## `Building #` = col\_double(),  
## `Completion Percentage` = col\_double(),  
## `Improvmnt SF` = col\_double(),  
## `Garage SF` = col\_double(),  
## `Basement SF` = col\_double(),  
## `Finished Basement SF` = col\_double(),  
## `Total Porch SF` = col\_double()  
## )

## See spec(...) for full column specifications.

head(dcBuildSum)

## State Parcel # Account # Owner Name  
## 1 235304104002 R0000553 KEVIN CASHMAN & KIM CASHMAN  
## 2 222933405002 R0000625 JENNIFER M BRADY & MICHAEL L WEICKUM  
## 3 235304107010 R0000684 STUART M HUNGSBERG & KRISTINE A HUNGSBERG  
## 4 222932003001 R0000895 GARRY BRAUER TRUST  
## 5 222932002011 R0001118 J CALVIN DOWNS & MELAINE V DOWNS  
## 6 235304110013 R0001177 REBECCA CONNET & DARIN CONNET  
## Owner Street Address Owner Street Address 2 Owner City Owner State  
## 1 7927 ELM ST null LOUVIERS CO  
## 2 7994 MAIN ST null LOUVIERS CO  
## 3 7819 VALLEY VIEW ST null LOUVIERS CO  
## 4 8312 W TRAIL SOUTH DR null LITTLETON CO  
## 5 7583 W TRAIL SOUTH DR null LITTLETON CO  
## 6 PO BOX 14 null LOUVIERS CO  
## Owner Zip Code Property Type Building # Quality Completion Percentage  
## 1 80131 Residential 1 Average 1  
## 2 80131 Residential 1 Average 1  
## 3 80131 Residential 1 Average 1  
## 4 80125-9514 Residential 1 Average 1  
## 5 80125-9511 Residential 1 Average 1  
## 6 80131-0014 Residential 1 Average 1  
## Improvmnt SF Unit Type Fireplace Count Garage SF Basement SF  
## 1 1288 null null 400 912  
## 2 1484 null null 0 0  
## 3 2858 null wood : 1 1541 520  
## 4 2200 null wood : 1 1464 2043  
## 5 2678 null wood : 1 520 1721  
## 6 1462 null null 900 988  
## Finished Basement SF Unfinished Basement SF Finished Basement %  
## 1 867 45 0.9506578947368421  
## 2 0 undefined undefined  
## 3 0 520 0  
## 4 1295 748 0.6338717572197748  
## 5 0 1721 0  
## 6 476 512 0.4817813765182186  
## Unfinished Basement % Walkout Basement Total Porch SF  
## 1 0.049342105263157895 false 228  
## 2 undefined null 696  
## 3 1 false 436  
## 4 0.36612824278022515 false 0  
## 5 1 false 508  
## 6 0.5182186234817814 false 436

str(dcBuildSum)

## 'data.frame': 26749 obs. of 23 variables:  
## $ State Parcel # : num 235304104002 222933405002 235304107010 222932003001 222932002011 ...  
## $ Account # : chr "R0000553" "R0000625" "R0000684" "R0000895" ...  
## $ Owner Name : chr "KEVIN CASHMAN & KIM CASHMAN" "JENNIFER M BRADY & MICHAEL L WEICKUM" "STUART M HUNGSBERG & KRISTINE A HUNGSBERG" "GARRY BRAUER TRUST" ...  
## $ Owner Street Address : chr "7927 ELM ST" "7994 MAIN ST" "7819 VALLEY VIEW ST" "8312 W TRAIL SOUTH DR" ...  
## $ Owner Street Address 2: chr "null" "null" "null" "null" ...  
## $ Owner City : chr "LOUVIERS" "LOUVIERS" "LOUVIERS" "LITTLETON" ...  
## $ Owner State : chr "CO" "CO" "CO" "CO" ...  
## $ Owner Zip Code : chr "80131" "80131" "80131" "80125-9514" ...  
## $ Property Type : chr "Residential" "Residential" "Residential" "Residential" ...  
## $ Building # : num 1 1 1 1 1 1 1 1 2 1 ...  
## $ Quality : chr "Average" "Average" "Average" "Average" ...  
## $ Completion Percentage : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ Improvmnt SF : num 1288 1484 2858 2200 2678 ...  
## $ Unit Type : chr "null" "null" "null" "null" ...  
## $ Fireplace Count : chr "null" "null" "wood : 1" "wood : 1" ...  
## $ Garage SF : num 400 0 1541 1464 520 ...  
## $ Basement SF : num 912 0 520 2043 1721 ...  
## $ Finished Basement SF : num 867 0 0 1295 0 ...  
## $ Unfinished Basement SF: chr "45" "undefined" "520" "748" ...  
## $ Finished Basement % : chr "0.9506578947368421" "undefined" "0" "0.6338717572197748" ...  
## $ Unfinished Basement % : chr "0.049342105263157895" "undefined" "1" "0.36612824278022515" ...  
## $ Walkout Basement : chr "false" "null" "false" "false" ...  
## $ Total Porch SF : num 228 696 436 0 508 436 12 767 715 759 ...

dim(dcBuildSum)

## [1] 26749 23

glimpse(dcBuildSum)

## Observations: 26,749  
## Variables: 23  
## $ `State Parcel #` <dbl> 235304104002, 222933405002, 235304107...  
## $ `Account #` <chr> "R0000553", "R0000625", "R0000684", "...  
## $ `Owner Name` <chr> "KEVIN CASHMAN & KIM CASHMAN", "JENNI...  
## $ `Owner Street Address` <chr> "7927 ELM ST", "7994 MAIN ST", "7819 ...  
## $ `Owner Street Address 2` <chr> "null", "null", "null", "null", "null...  
## $ `Owner City` <chr> "LOUVIERS", "LOUVIERS", "LOUVIERS", "...  
## $ `Owner State` <chr> "CO", "CO", "CO", "CO", "CO", "CO", "...  
## $ `Owner Zip Code` <chr> "80131", "80131", "80131", "80125-951...  
## $ `Property Type` <chr> "Residential", "Residential", "Reside...  
## $ `Building #` <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1...  
## $ Quality <chr> "Average", "Average", "Average", "Ave...  
## $ `Completion Percentage` <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...  
## $ `Improvmnt SF` <dbl> 1288, 1484, 2858, 2200, 2678, 1462, 2...  
## $ `Unit Type` <chr> "null", "null", "null", "null", "null...  
## $ `Fireplace Count` <chr> "null", "null", "wood : 1", "wood : 1...  
## $ `Garage SF` <dbl> 400, 0, 1541, 1464, 520, 900, 1044, 0...  
## $ `Basement SF` <dbl> 912, 0, 520, 2043, 1721, 988, 0, 1055...  
## $ `Finished Basement SF` <dbl> 867, 0, 0, 1295, 0, 476, 0, 1002, 141...  
## $ `Unfinished Basement SF` <chr> "45", "undefined", "520", "748", "172...  
## $ `Finished Basement %` <chr> "0.9506578947368421", "undefined", "0...  
## $ `Unfinished Basement %` <chr> "0.049342105263157895", "undefined", ...  
## $ `Walkout Basement` <chr> "false", "null", "false", "false", "f...  
## $ `Total Porch SF` <dbl> 228, 696, 436, 0, 508, 436, 12, 767, ...

### Clean the building summary dataset

dcBsClean <- subset(dcBuildSum[order(dcBuildSum$`Account #`),], `Property Type` == "Residential")  
   
dcBsDup1 = !duplicated(dcBuildSum[,1:2])  
dcBsDup2 = !duplicated(apply(dcBuildSum[,1:2], 2, rev))  
dcBsClean <- dcBuildSum[dcBsDup1 & rev(dcBsDup2), ]  
   
dcBsClean$Quality[dcBsClean$Quality == 'Low'] <- 0  
dcBsClean$Quality[dcBsClean$Quality == 'Fair'] <- 1  
dcBsClean$Quality[dcBsClean$Quality == 'Average'] <- 2  
dcBsClean$Quality[dcBsClean$Quality == 'Good'] <- 3  
dcBsClean$Quality[dcBsClean$Quality == 'Very Good'] <- 4  
dcBsClean$Quality[dcBsClean$Quality == 'Excellent'] <- 5  
dcBsClean$Quality <- as.double(dcBsClean$Quality)  
   
dcBsClean$`Walkout Basement`[dcBsClean$`Walkout Basement` != "true"] <- 0  
dcBsClean$`Walkout Basement`[dcBsClean$`Walkout Basement` == "true"] <- 1  
dcBsClean$`Walkout Basement` <- as.double(dcBsClean$`Walkout Basement`)

### Add sale year and location columns to the sales dataset

dcSalesClean$`Sale Date` <- as.Date(dcSalesClean$`Sale Date`,format="%m/%d/%Y")  
dcSalesClean[, "Sale Year"] <- format(dcSalesClean[,"Sale Date"], "%Y")  
dcSalesClean$`Sale Year` <- as.double(dcSalesClean$`Sale Year`)  
   
#dcSalesClean <- dcSalesClean %>%   
# mutate(Location = paste0(`Account #`, ", en-US, ", `Situs Street Address`, ", ", `Situs City`, ", CO ", `Situs Zip Code`))  
   
dcSalesClean$Location <- paste0(dcSalesClean$`Situs Street Address`, ", ", dcSalesClean$`Situs City`, ", CO ", dcSalesClean$`Situs Zip Code`)  
   
glimpse(dcSalesClean$`Sale Year`)

## num [1:28947] 2016 2014 2018 2015 2014 ...

### Remove columns from the three datasets which are not being used for evaluations

dcSalesClean2 <- dcSalesClean  
dcSalesClean2 <- dcSalesClean2[, -c(17:20)]  
dcSalesClean2 <- dcSalesClean2[, -c(12:14)]  
dcSalesClean2 <- dcSalesClean2[, -c(4:8)]  
glimpse(dcSalesClean2)

## Observations: 28,947  
## Variables: 11  
## $ `State Parcel #` <dbl> 235304104002, 222933405002, 23530410701...  
## $ `Account #` <chr> "R0000553", "R0000625", "R0000684", "R0...  
## $ `Owner Name` <chr> "KEVIN CASHMAN & KIM CASHMAN", "JENNIFE...  
## $ `Situs Street Address` <chr> "7927 ELM ST", "7994 MAIN ST", "7819 VA...  
## $ `Situs City` <chr> "LOUVIERS", "LOUVIERS", "LOUVIERS", "LI...  
## $ `Situs Zip Code` <dbl> 80131, 80131, 80131, 80125, 80131, 8012...  
## $ `Sale Date` <date> 2016-05-27, 2014-10-06, 2018-04-10, 20...  
## $ `Sale Price` <dbl> 298600, 235000, 554100, 634900, 326000,...  
## $ Acres <dbl> 0.253, 0.227, 0.386, 4.780, 0.556, 1.00...  
## $ `Sale Year` <dbl> 2016, 2014, 2018, 2015, 2014, 2014, 201...  
## $ Location <chr> "7927 ELM ST, LOUVIERS, CO 80131", "799...

dcPisrClean2 <- dcPisrClean  
dcPisrClean2 <- dcPisrClean2[, -c(19:25)]  
dcPisrClean2 <- dcPisrClean2[, -c(4:13)]  
glimpse(dcPisrClean2)

## Observations: 26,635  
## Variables: 8  
## $ `State Parcel #` <dbl> 235304104002, 222933405002, 235304107010, 222...  
## $ `Account #` <chr> "R0000553", "R0000625", "R0000684", "R0000895...  
## $ `Owner Name` <chr> "KEVIN CASHMAN & KIM CASHMAN", "JENNIFER M BR...  
## $ Style <dbl> 1, 1, 5, 1, 8, 5, 1, 8, 1, 8, 1, 1, 1, 8, 1, ...  
## $ `Year Built` <dbl> 1966, 1908, 1911, 1972, 1976, 1972, 1970, 190...  
## $ Stories <dbl> 1, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, ...  
## $ `Bedroom Count` <dbl> 2, 2, 4, 3, 3, 4, 3, 1, 2, 3, 3, 3, 2, 3, 4, ...  
## $ `Bathroom Count` <dbl> 1, 2, 2, 2, 3, 2, 3, 2, 2, 3, 2, 1, 1, 3, 2, ...

dcBsClean2 <- dcBsClean  
dcBsClean2 <- dcBsClean2[, -c(19:21)]  
dcBsClean2 <- dcBsClean2[, -c(14:15)]  
dcBsClean2 <- dcBsClean2[, -c(12)]  
dcBsClean2 <- dcBsClean2[, -c(4:10)]  
glimpse(dcBsClean2)

## Observations: 26,635  
## Variables: 10  
## $ `State Parcel #` <dbl> 235304104002, 222933405002, 23530410701...  
## $ `Account #` <chr> "R0000553", "R0000625", "R0000684", "R0...  
## $ `Owner Name` <chr> "KEVIN CASHMAN & KIM CASHMAN", "JENNIFE...  
## $ Quality <dbl> 2, 2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, ...  
## $ `Improvmnt SF` <dbl> 1288, 1484, 2858, 2200, 2678, 1462, 293...  
## $ `Garage SF` <dbl> 400, 0, 1541, 1464, 520, 900, 1044, 0, ...  
## $ `Basement SF` <dbl> 912, 0, 520, 2043, 1721, 988, 0, 1055, ...  
## $ `Finished Basement SF` <dbl> 867, 0, 0, 1295, 0, 476, 0, 1002, 1415,...  
## $ `Walkout Basement` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...  
## $ `Total Porch SF` <dbl> 228, 696, 436, 0, 508, 436, 12, 767, 71...

### Merge the three datasets into one dataset for evaluations and remove bad data

dcSalesMerge <- merge(dcSalesClean2,dcPisrClean2,by=c("State Parcel #","Account #", "Owner Name"))  
   
dcSalesMerge <- merge(dcSalesMerge,dcBsClean2,by=c("State Parcel #","Account #", "Owner Name"))  
   
dcSalesMerge <- subset(dcSalesMerge, !(`Sale Year` < `Year Built`))  
dcSalesMerge <- dcSalesMerge %>% distinct()  
   
sum(is.na(dcSalesMerge))

## [1] 9

dcSalesMerge <- na.omit(dcSalesMerge)  
sum(is.na(dcSalesMerge))

## [1] 0

dcSalesMerge <- subset(dcSalesMerge, !(`Situs City` == "ELBERT"))

### Below is the code used to extract property address Latitude and Longitude but I only ran once since it runs a long time and there is a limit to usage

### I am commenting the commands out but leaving the code for future use. There were 7 errors so I included the code below to fix records as well.

### options(BingMapsKey=‘Really long key’)

### x<-lapply(dcSalesClean5$location[i], geocode, service = “bing”, returntype = “coordinates”)

### Loop through the addesses to get the latitude and longitude of each address and add columns Lat and Lon to the dcSalesClean data frame

### The file was written out before. There is a key to extract the Latitude and Longitude so I only ran this process once and I am reading it back in to merge with the data.

### for (i in 1:nrow(dcSalesClean5))

### {

### coord <- geocode(dcSalesMerge$Location[i], service = “bing”, returntype=“coordinates”)

### dcSalesMerge$lat[i] <- as.numeric(coord[1])

### dcSalesMerge$lon[i] <- as.numeric(coord[2])

### }

### write.csv(dcSalesMerge, “SalesInformationFinal.csv”, row.names=FALSE)

### dcSalesFix <- data.frame(read\_csv(‘SalesInformationFix.csv’,col\_names = TRUE), check.names=FALSE)

### dim(dcSalesFix)

### Loop through the addesses to get the latitude and longitude of each address and add columns Lat and Lon to the dcSalesFix data frame

### for (i in 1:nrow(dcSalesFix))

### {

### coord <- geocode(dcSalesFix$Location[i], service = “bing”, returntype=“coordinates”)

### dcSalesFix$lat[i] <- as.numeric(coord[1])

### dcSalesFix$lon[i] <- as.numeric(coord[2])

### }

### write.csv(dcSalesFix, “SalesInformationFixed.csv”, row.names=FALSE)

### Read saved file and extract Latitude and Longitude coordinates from previous saved file and combine with data from merged file.

dcSalesFinalCoord <- data.frame(read\_csv('SalesInformationFinal.csv',col\_names = TRUE), check.names=FALSE)

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## `Account #` = col\_character(),  
## `Owner Name` = col\_character(),  
## `Situs Street Address` = col\_character(),  
## `Situs City` = col\_character(),  
## `Sale Date` = col\_character(),  
## Location = col\_character()  
## )

## See spec(...) for full column specifications.

dim(dcSalesFinalCoord)

## [1] 28523 25

dcSalesFinalCoord <- dcSalesFinalCoord[, -c(11:23)]  
dcSalesFinalCoord$`Sale Date` <- as.Date(dcSalesFinalCoord$`Sale Date`,format="%m/%d/%Y")  
dcSalesFinal <- merge(dcSalesMerge,dcSalesFinalCoord,by=c("State Parcel #","Account #", "Owner Name", "Situs Street Address", "Situs City","Situs Zip Code","Sale Date","Sale Price","Acres","Sale Year"))  
dim(dcSalesFinal)

## [1] 28523 25

str(dcSalesFinal)

## 'data.frame': 28523 obs. of 25 variables:  
## $ State Parcel # : num 222724000006 222724002007 222724002007 222724003004 222724007003 ...  
## $ Account # : chr "R0157016" "R0002794" "R0002794" "R0002250" ...  
## $ Owner Name : chr "KIM C CARLSON & EVAN D ELA" "MARCUS A MCDERMETT" "MARCUS A MCDERMETT" "BRITTON L JAMES & ASHLEY JAMES" ...  
## $ Situs Street Address: chr "10500 WILDLIFE WAY" "10852 SOLAR DR" "10852 SOLAR DR" "9499 WAGONWHEEL DR" ...  
## $ Situs City : chr "LITTLETON" "LITTLETON" "LITTLETON" "LITTLETON" ...  
## $ Situs Zip Code : num 80125 80125 80125 80125 80125 ...  
## $ Sale Date : Date, format: "2017-08-29" "2017-03-15" ...  
## $ Sale Price : num 1540000 520000 605000 655000 406000 640000 480000 370000 699000 460000 ...  
## $ Acres : num 7.2 1 1 1 1.01 ...  
## $ Sale Year : num 2017 2017 2018 2016 2014 ...  
## $ Location : chr "10500 WILDLIFE WAY, LITTLETON, CO 80125" "10852 SOLAR DR, LITTLETON, CO 80125" "10852 SOLAR DR, LITTLETON, CO 80125" "9499 WAGONWHEEL DR, LITTLETON, CO 80125" ...  
## $ Style : num 1 1 1 1 1 8 1 1 1 1 ...  
## $ Year Built : num 1997 1978 1978 1950 1970 ...  
## $ Stories : num 1 1 1 1 1 2 1 1 1 1 ...  
## $ Bedroom Count : num 3 1 1 3 3 3 2 2 3 1 ...  
## $ Bathroom Count : num 3 2 2 1 3 3 2 2 2 3 ...  
## $ Quality : num 4 2 2 2 3 3 2 2 2 2 ...  
## $ Improvmnt SF : num 3497 1619 1619 1194 2932 ...  
## $ Garage SF : num 1043 0 0 992 1044 ...  
## $ Basement SF : num 2477 1167 1167 858 0 ...  
## $ Finished Basement SF: num 2353 1108 1108 795 0 ...  
## $ Walkout Basement : num 1 0 0 0 0 0 0 0 0 1 ...  
## $ Total Porch SF : num 105 1058 1058 902 12 ...  
## $ lat : num 39.5 39.5 39.5 39.5 39.5 ...  
## $ long : num -105 -105 -105 -105 -105 ...

write.csv(dcSalesFinal, "SalesFinalMerged.csv", row.names=FALSE)

### Use FIPS to create Douglas County Map and plot Douglas County sales data

### Extract the map coordinates from the Census Bureau and unzip the information in the directory before using. I commented this command as well since it needs to be run once.

### <https://www2.census.gov/geo/tiger/GENZ2010/gz_2010_08_060_00_500k.zip>

### I used the coordinates to map all of Colorado then I limited to only Douglas County and removed division lines from within the county since this map showed county districts.

shape\_file <- "gz\_2010\_08\_060\_00\_500k"  
shape\_file\_dir <- shape\_file  
   
raw\_tract <- readOGR(dsn = shape\_file\_dir, layer = shape\_file)

## OGR data source with driver: ESRI Shapefile   
## Source: "C:\Users\Owner\Documents\MSDS696 Class Info\Class\_Assignments\Practicum\_II\gz\_2010\_08\_060\_00\_500k", layer: "gz\_2010\_08\_060\_00\_500k"  
## with 209 features  
## It has 7 fields

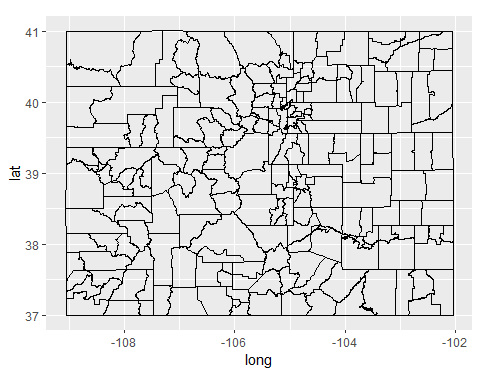
class(raw\_tract)

## [1] "SpatialPolygonsDataFrame"  
## attr(,"package")  
## [1] "sp"

tract <- fortify(raw\_tract, region="GEO\_ID")  
head(tract)

## long lat order hole piece id  
## 1 -104.7523 40.00047 1 FALSE 1 0600000US0800190399  
## 2 -104.7307 40.00051 2 FALSE 1 0600000US0800190399  
## 3 -104.7253 40.00051 3 FALSE 1 0600000US0800190399  
## 4 -104.7162 40.00051 4 FALSE 1 0600000US0800190399  
## 5 -104.7148 40.00051 5 FALSE 1 0600000US0800190399  
## 6 -104.7136 40.00051 6 FALSE 1 0600000US0800190399  
## group  
## 1 0600000US0800190399.1  
## 2 0600000US0800190399.1  
## 3 0600000US0800190399.1  
## 4 0600000US0800190399.1  
## 5 0600000US0800190399.1  
## 6 0600000US0800190399.1

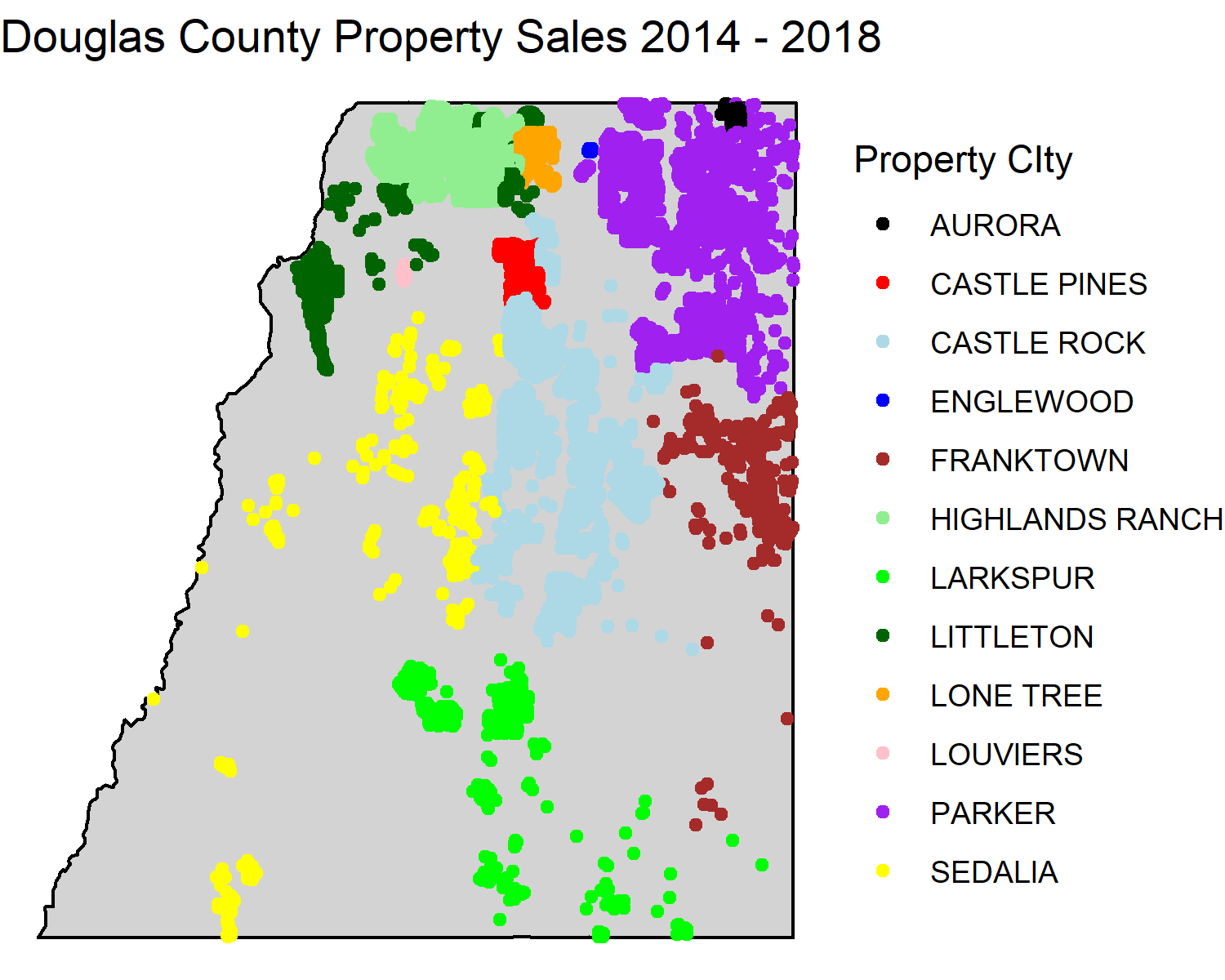
outline\_layer <- geom\_polygon(aes(long, lat, group = group),  
 fill = NA, col = "black", size = 0.2)  
   
ggplot(tract) + outline\_layer + coord\_quickmap()



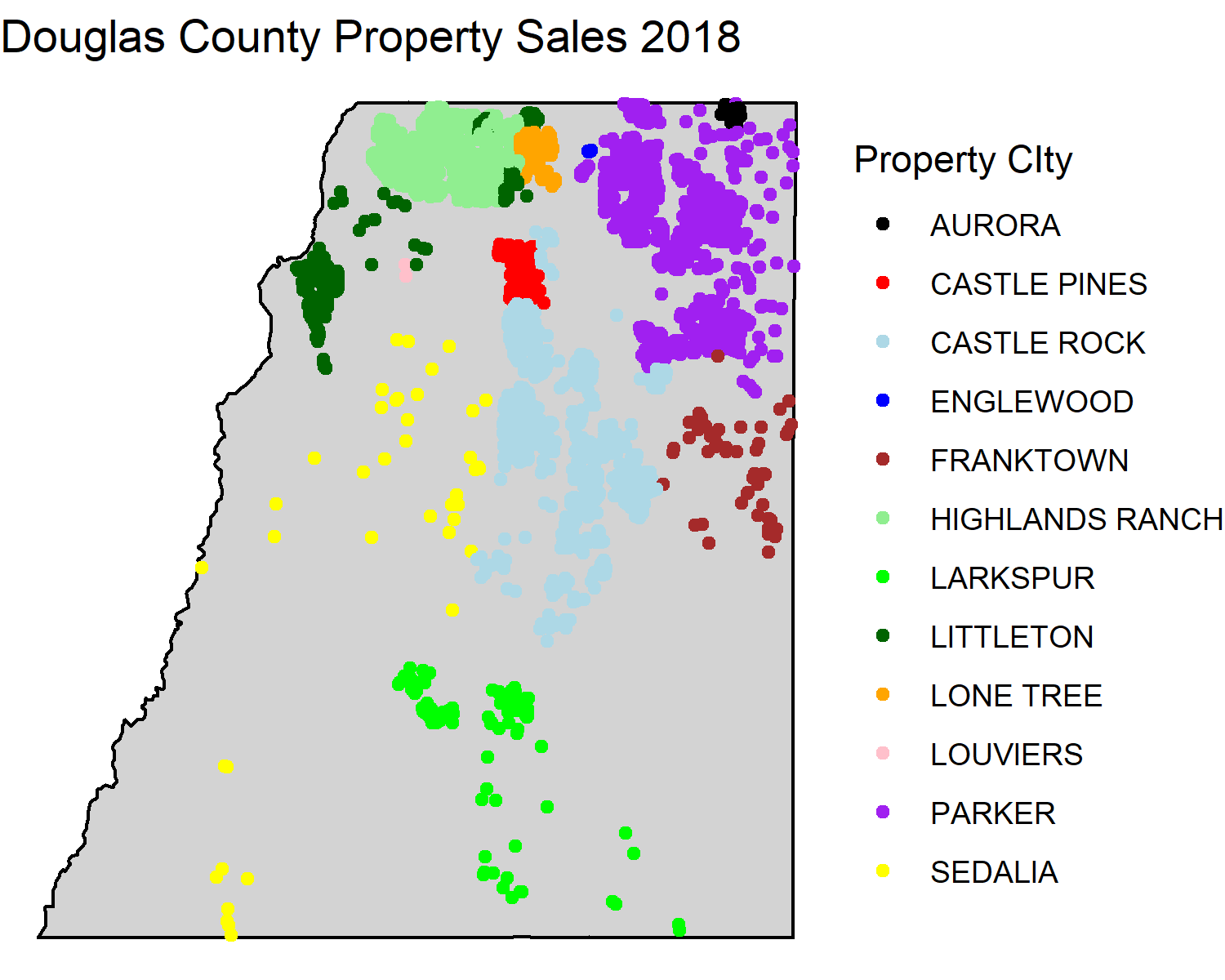
county.fips.table <-  
 mutate(maps::county.fips, county = sub(":.\*", "", polyname))  
   
getCountyFIPS <- function(county, state = "colorado") {  
 key <- paste(tolower(state), tolower(county), sep = ",")  
 idx <- match(key, county.fips.table$county)  
 county.fips.table$fips[idx]  
}  
   
county\_data <- function(tract, county, state = "colorado") {  
 fips <- getCountyFIPS(county, state)  
 pkey <- paste0("0600000US0", fips)  
 filter(tract, grepl(pkey, id))  
}  
   
dcTract <- county\_data(tract, "douglas")  
   
dcTract <- dcTract[ -c(1:92,103:277,292:397,447:651), ]  
   
for (i in 1:nrow(dcTract))  
{  
 dcTract$order[i][dcTract$order[i] >11164 & dcTract$order[i] < 11175] <- (dcTract$order[i] - 1000)  
 dcTract$order[i][dcTract$order[i] >11349 & dcTract$order[i] < 11364] <- (dcTract$order[i] - 2000)  
 dcTract$order[i][dcTract$order[i] >11469 & dcTract$order[i] < 11509] <- (dcTract$order[i] - 3000)  
 dcTract$order[i][dcTract$order[i] >11508 & dcTract$order[i] < 11519] <- (dcTract$order[i] - 4000)  
}  
   
dcTract <- dcTract[order(dcTract$order),]

### Plot map with all five years of sales data for review then plot only 2018 for comparison.

ggplot(dcTract, aes(x=long, y=lat)) +  
 geom\_polygon(fill = "lightgrey", color = "black") +   
 geom\_point(data = dcSalesFinal, aes(x=long, y=lat, group=as.factor(`Situs City`), color = as.factor(`Situs City`))) +   
 theme\_void() +   
 scale\_color\_manual(name = "Property CIty", values = c("black", "red", "lightblue", "blue", "brown", "lightgreen", "green", "darkgreen", "orange", "pink", "purple", "yellow")) +   
 guides(alpha = guide\_legend(title.position = "top", title.hjust = 0.5)) +  
 labs(title="Douglas County Property Sales 2014 - 2018")

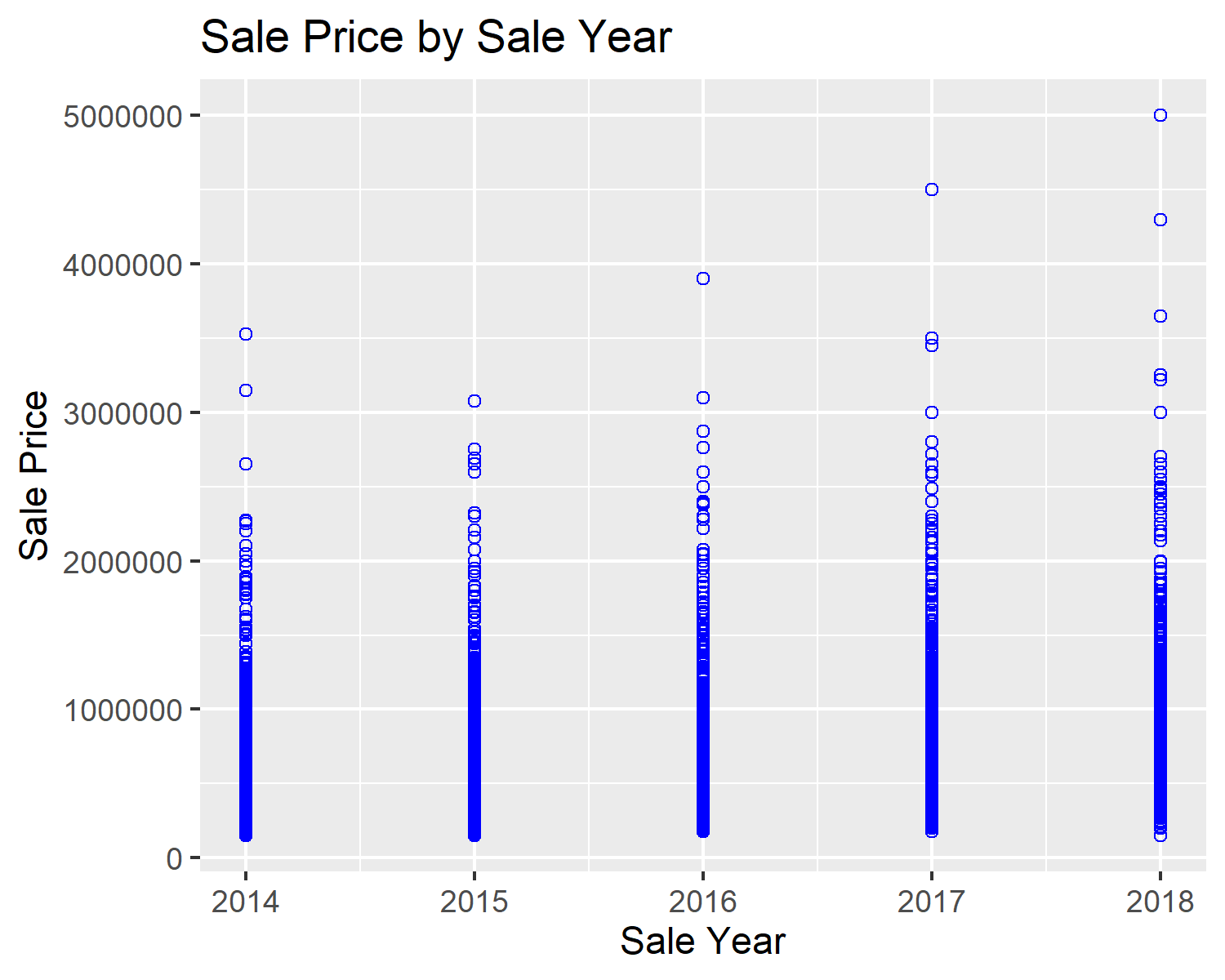


dcSalesFinal2018 <- subset(dcSalesFinal, (`Sale Year` == 2018))  
   
ggplot(dcTract, aes(x=long, y=lat)) +  
 geom\_polygon(fill = "lightgrey", color = "black") +   
 geom\_point(data = dcSalesFinal2018, aes(x=long, y=lat, group=as.factor(`Situs City`), color = as.factor(`Situs City`))) +   
 theme\_void() +   
 scale\_color\_manual(name = "Property CIty", values = c("black", "red", "lightblue", "blue", "brown", "lightgreen", "green", "darkgreen", "orange", "pink", "purple", "yellow")) +   
 guides(alpha = guide\_legend(title.position = "top", title.hjust = 0.5)) +  
 labs(title="Douglas County Property Sales 2018")

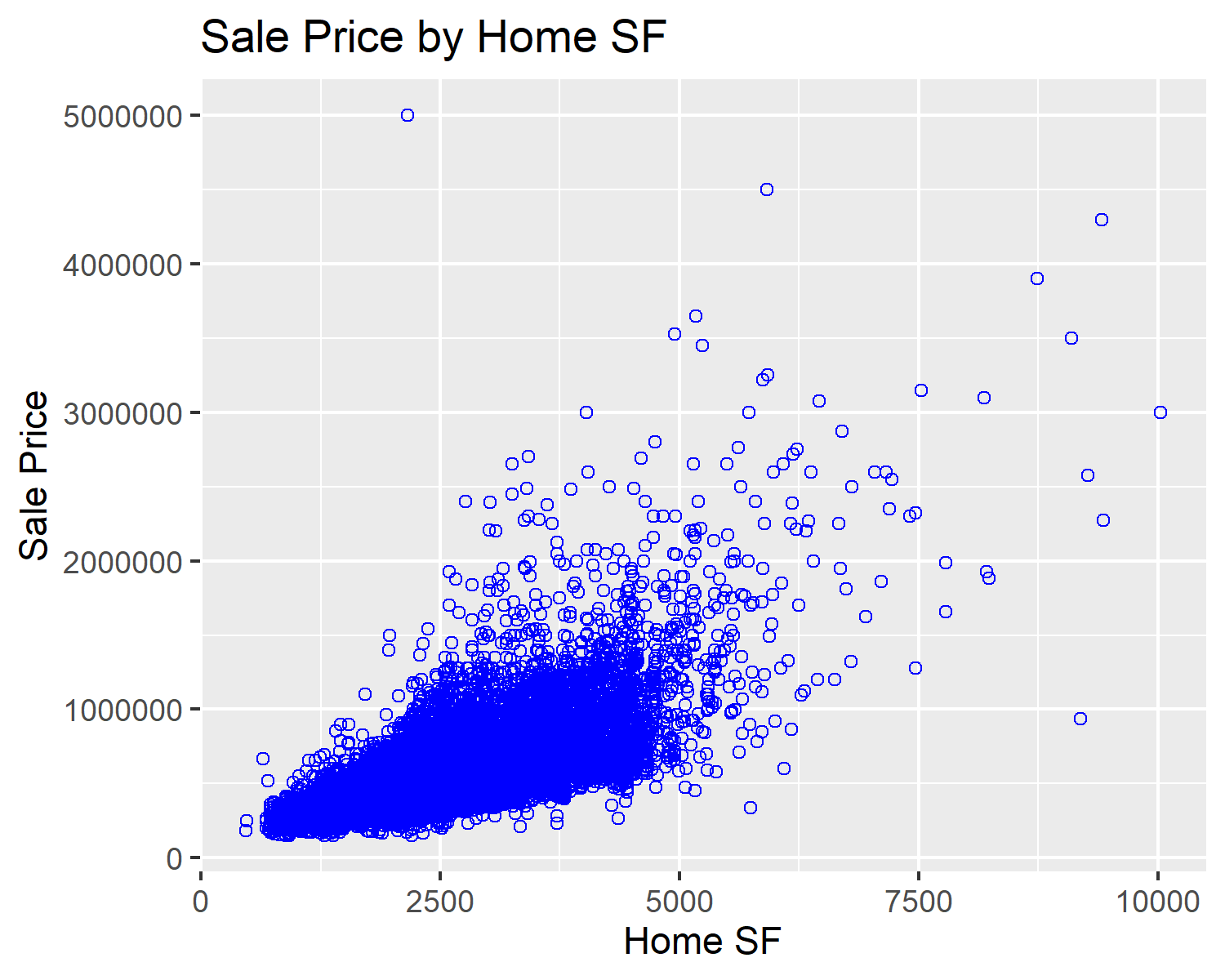


### Plot comparisons of several of the variables for comparison.

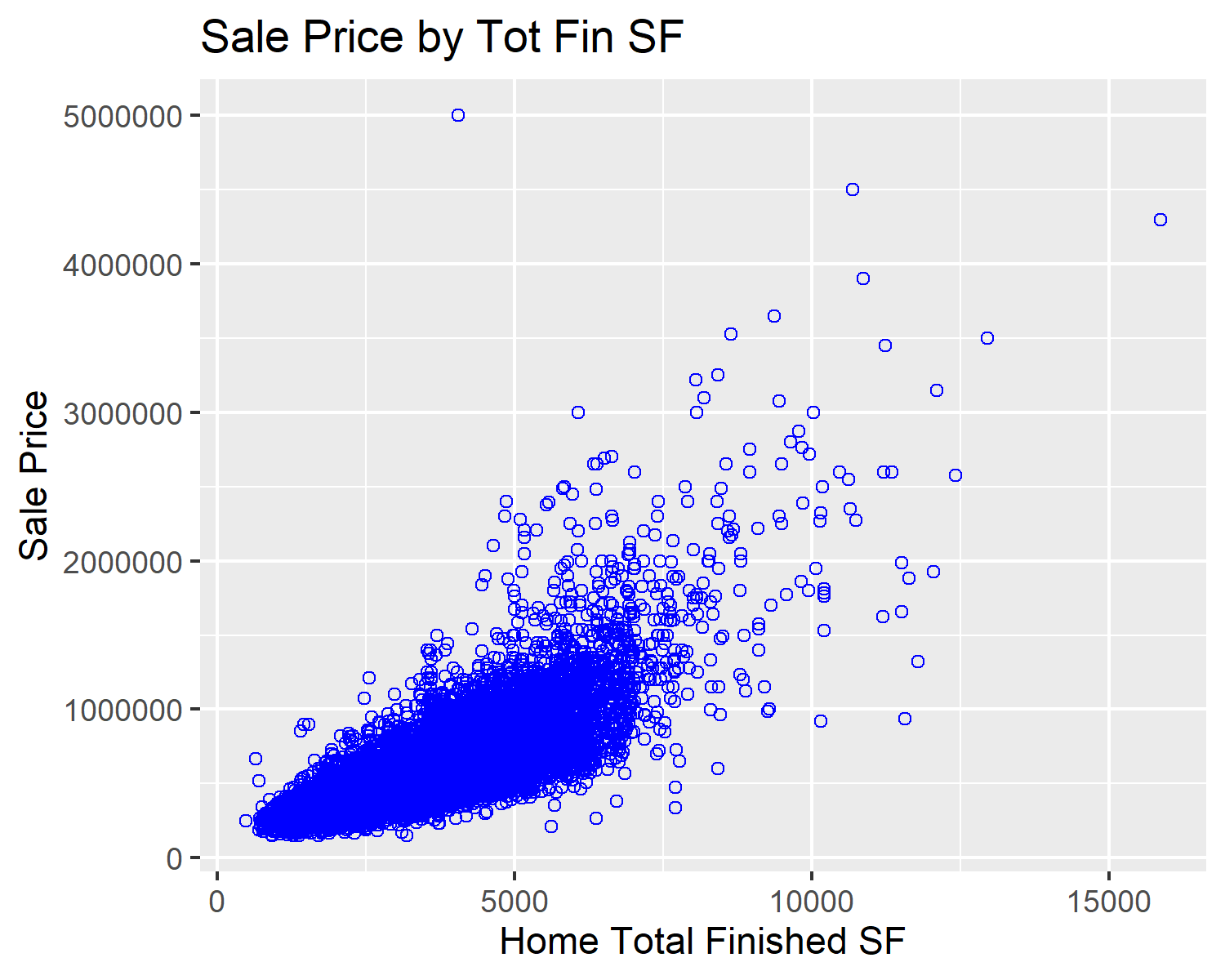
ggplot(dcSalesFinal, aes(x=`Sale Year`, y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Sale Year")



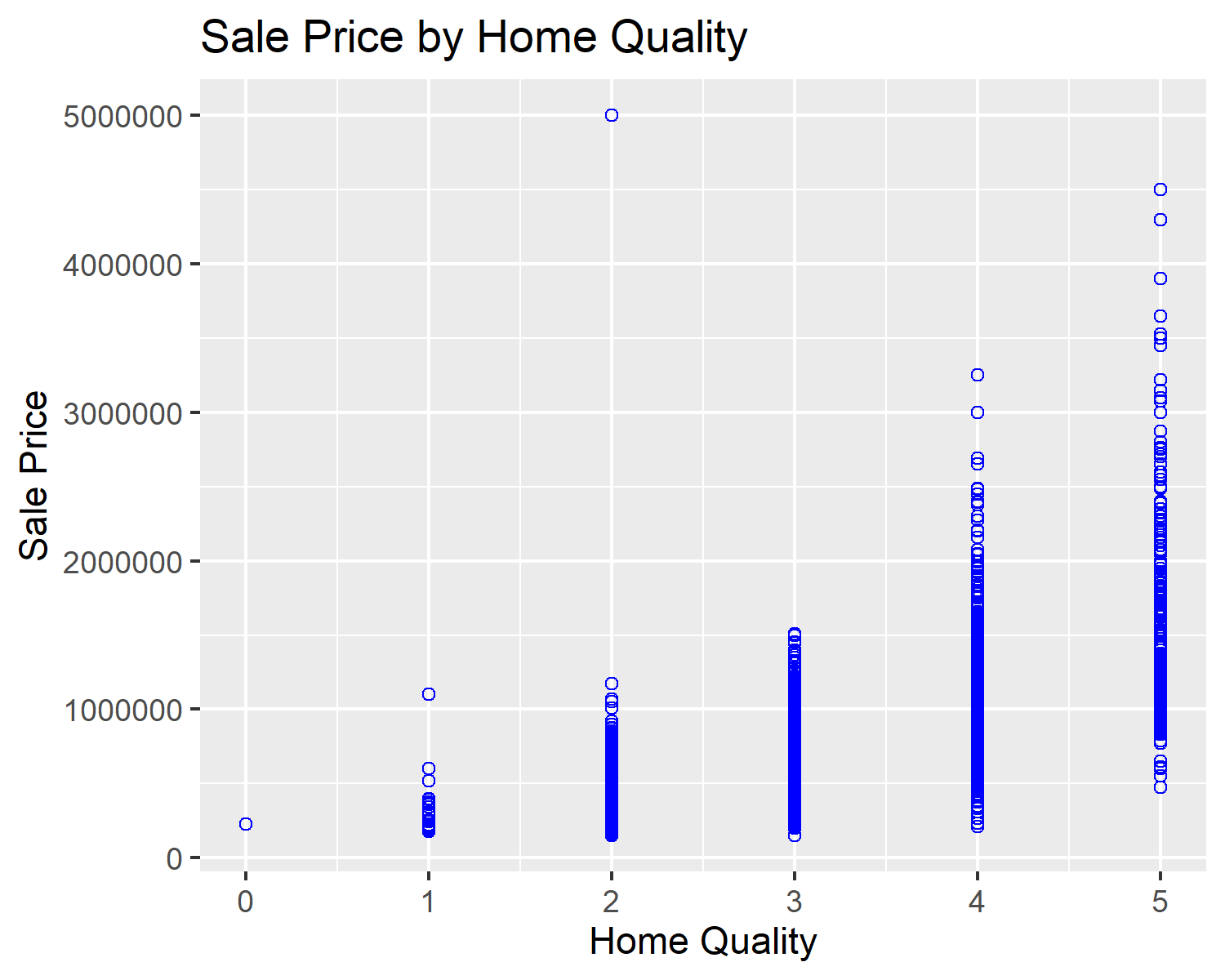
ggplot(dcSalesFinal, aes(x=`Improvmnt SF`,y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Home SF") + xlab("Home SF") + ylab("Sale Price")



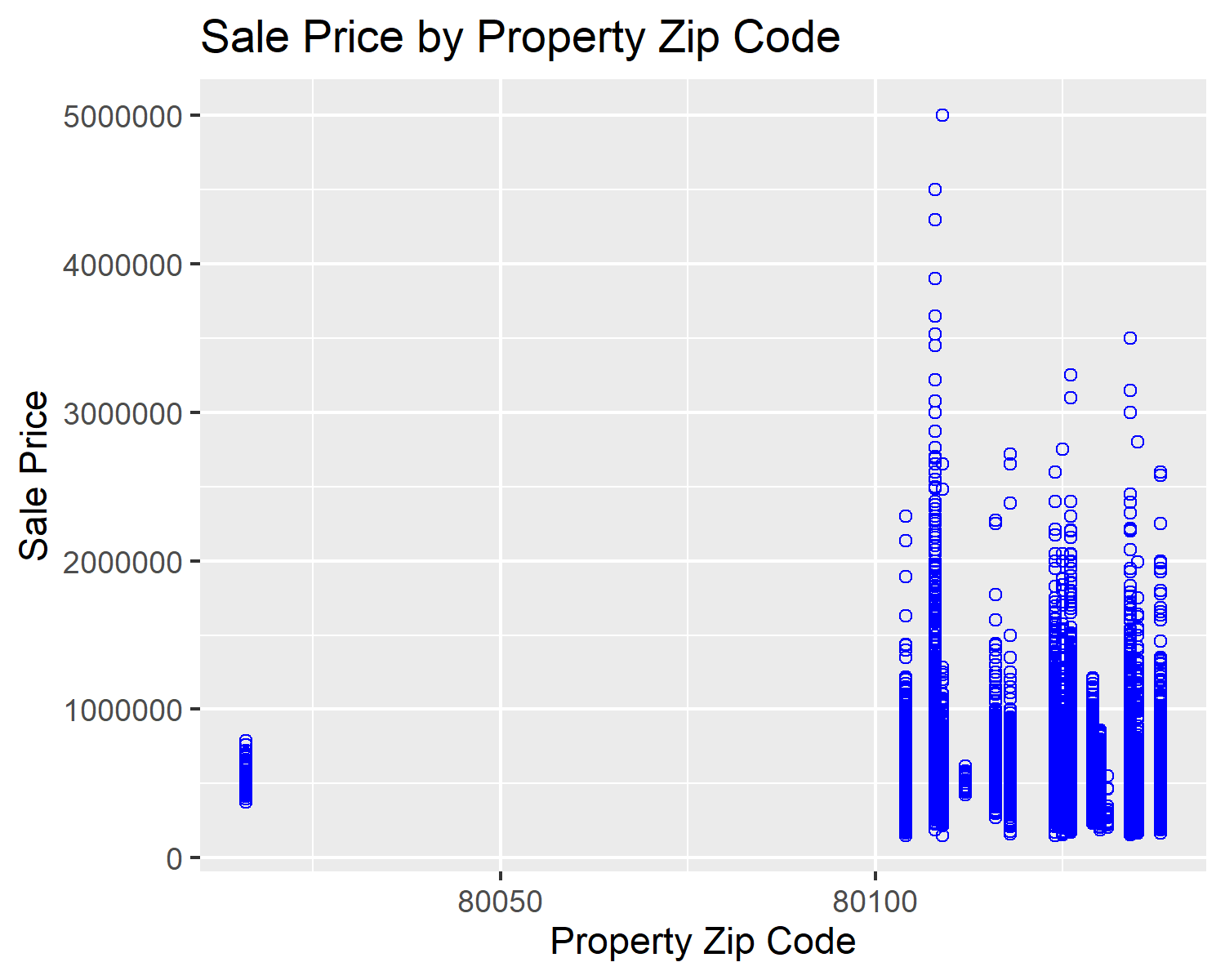
dcSalesFinal$`Total Finished SF` <- dcSalesFinal$`Improvmnt SF` + dcSalesFinal$`Finished Basement SF`  
ggplot(dcSalesFinal, aes(x=dcSalesFinal$`Total Finished SF`,y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Tot Fin SF") + xlab("Home Total Finished SF") + ylab("Sale Price")



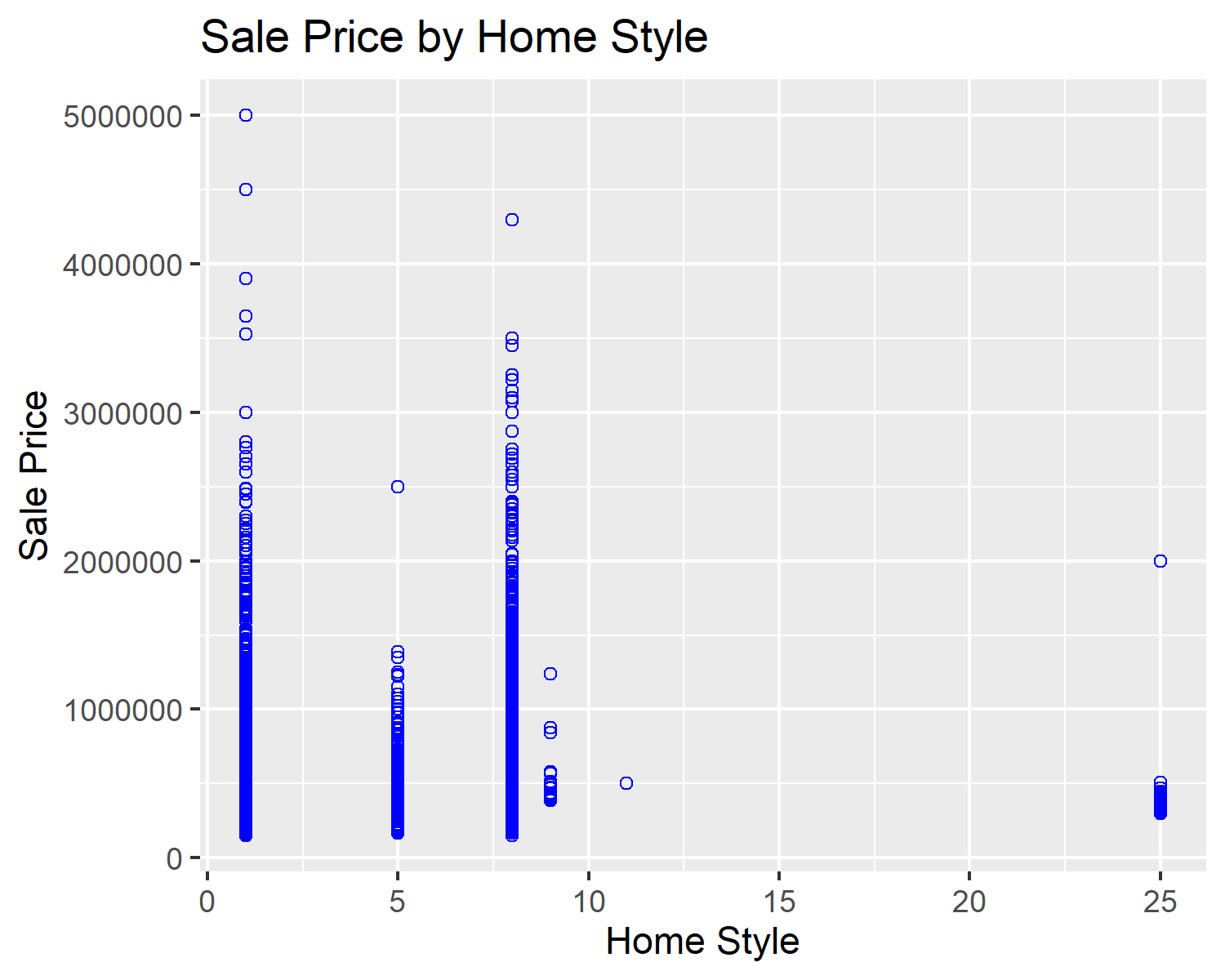
ggplot(dcSalesFinal, aes(x=Quality,y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Home Quality") + xlab("Home Quality") + ylab("Sale Price")



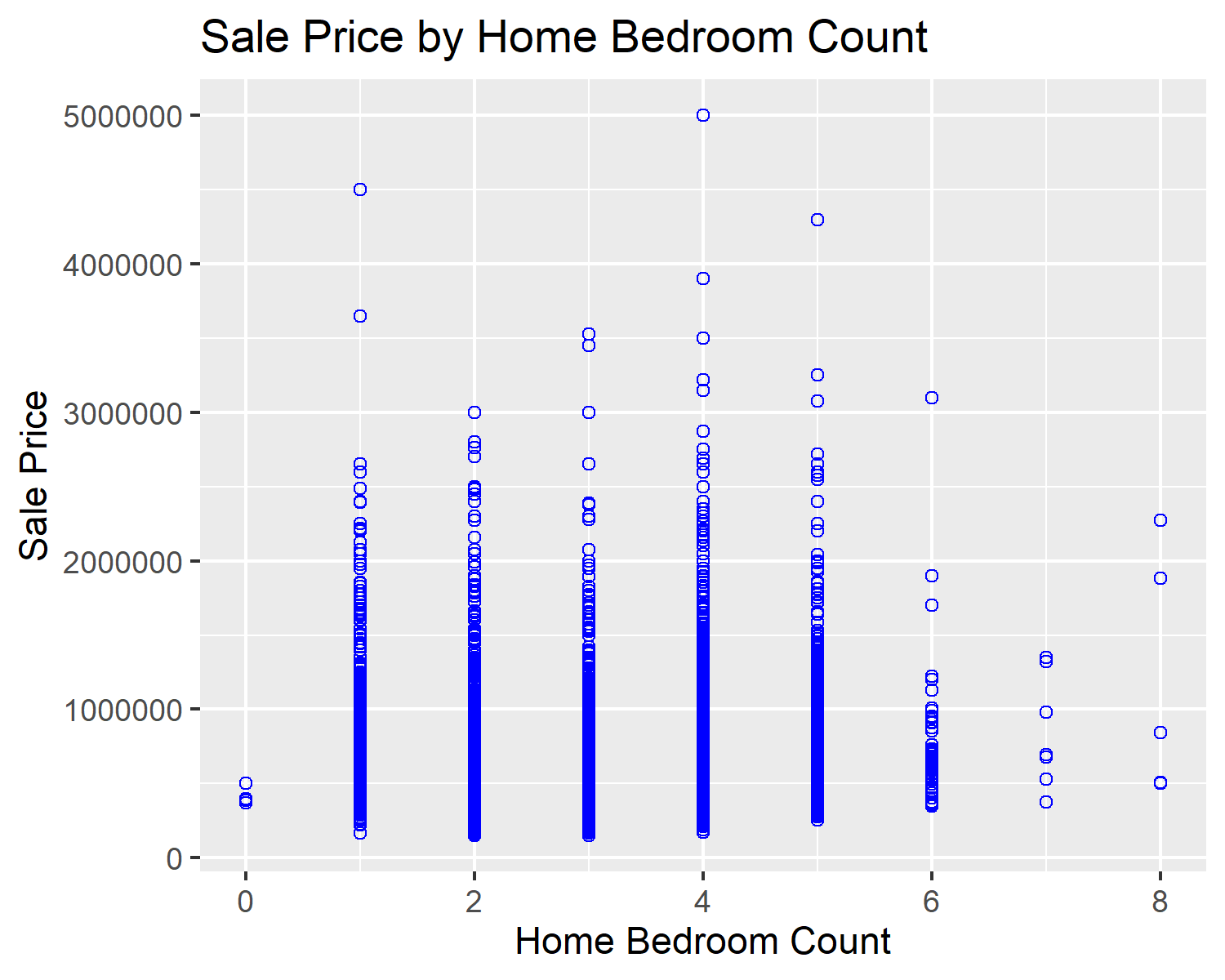
ggplot(dcSalesFinal, aes(x=`Situs Zip Code`,y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Property Zip Code") + xlab("Property Zip Code") + ylab("Sale Price")



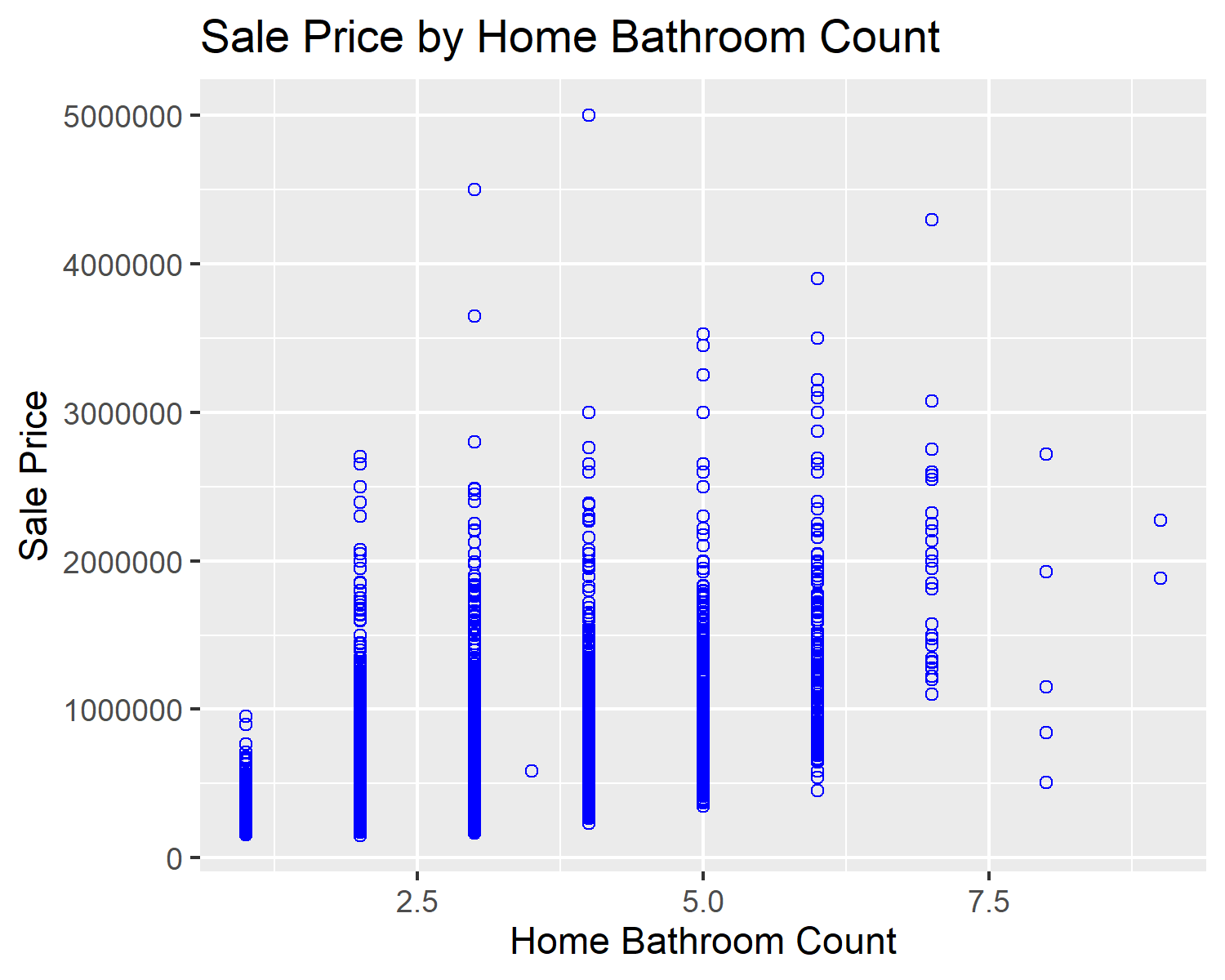
ggplot(dcSalesFinal, aes(x=Style,y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Home Style") + xlab("Home Style") + ylab("Sale Price")



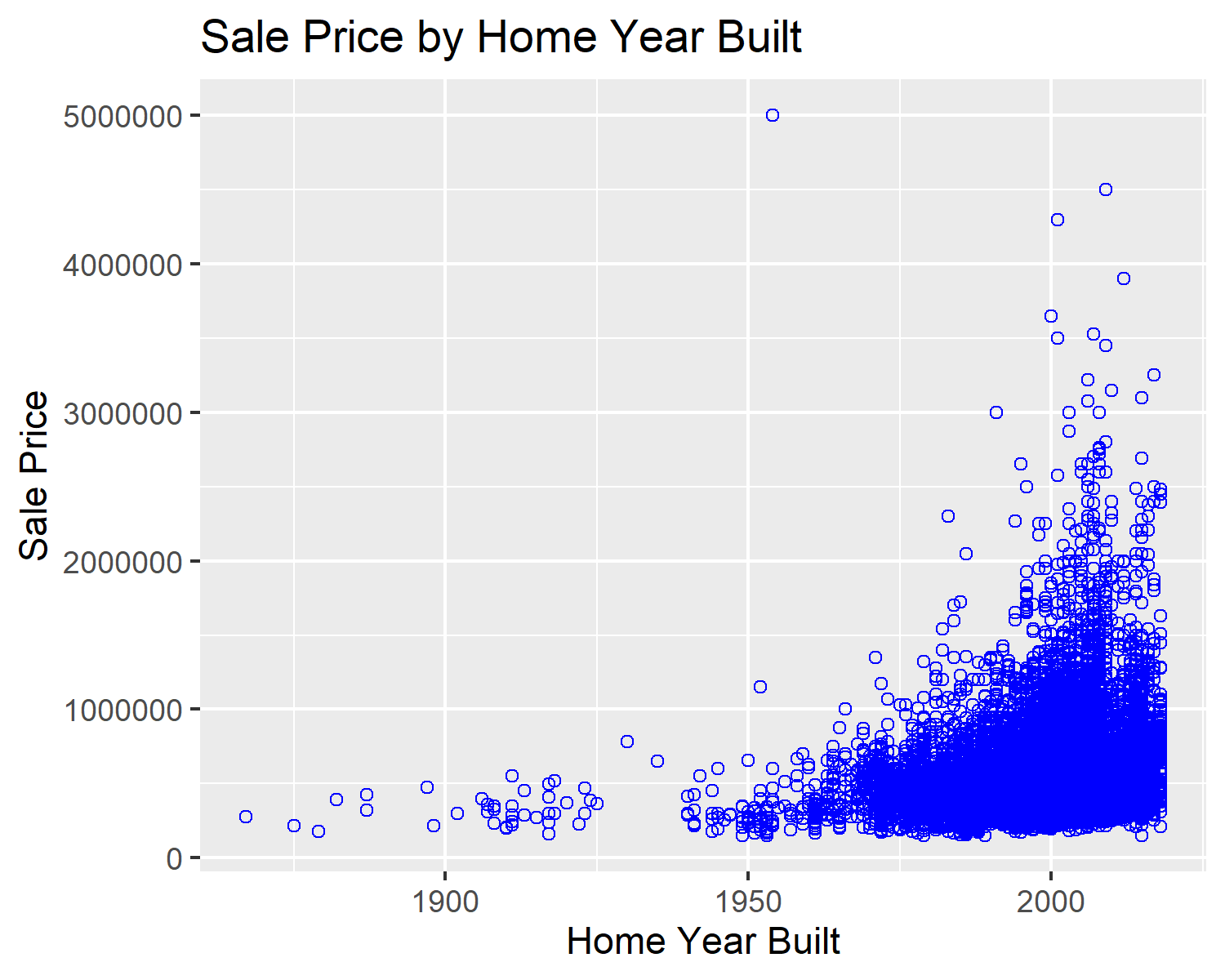
ggplot(dcSalesFinal, aes(x=`Bedroom Count`,y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Home Bedroom Count") + xlab("Home Bedroom Count") + ylab("Sale Price")



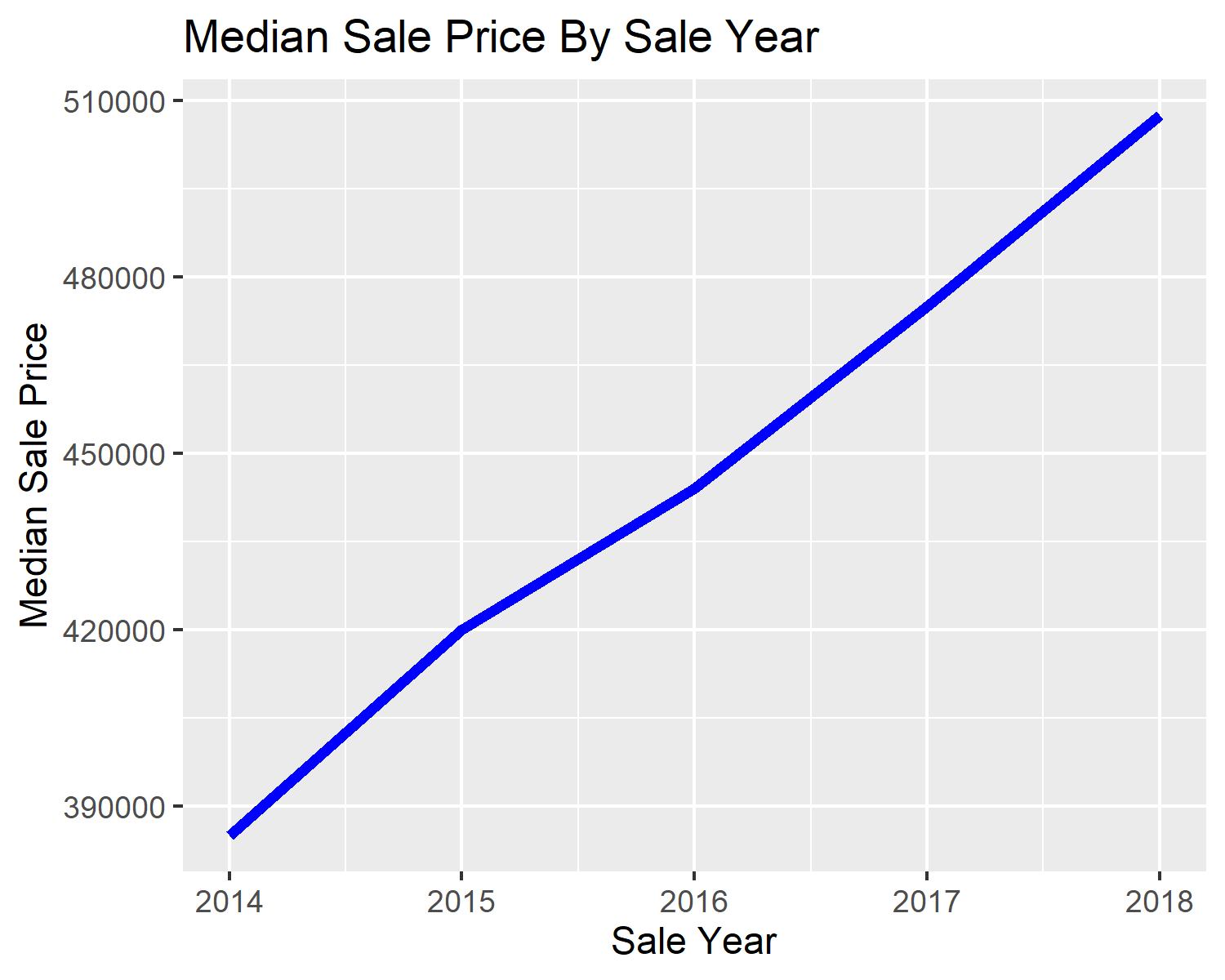
ggplot(dcSalesFinal, aes(x=`Bathroom Count`,y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Home Bathroom Count") + xlab("Home Bathroom Count") + ylab("Sale Price")



ggplot(dcSalesFinal, aes(x=`Year Built`,y=`Sale Price`)) + geom\_point(shape=1, color='blue') + ggtitle("Sale Price by Home Year Built") + xlab("Home Year Built") + ylab("Sale Price")

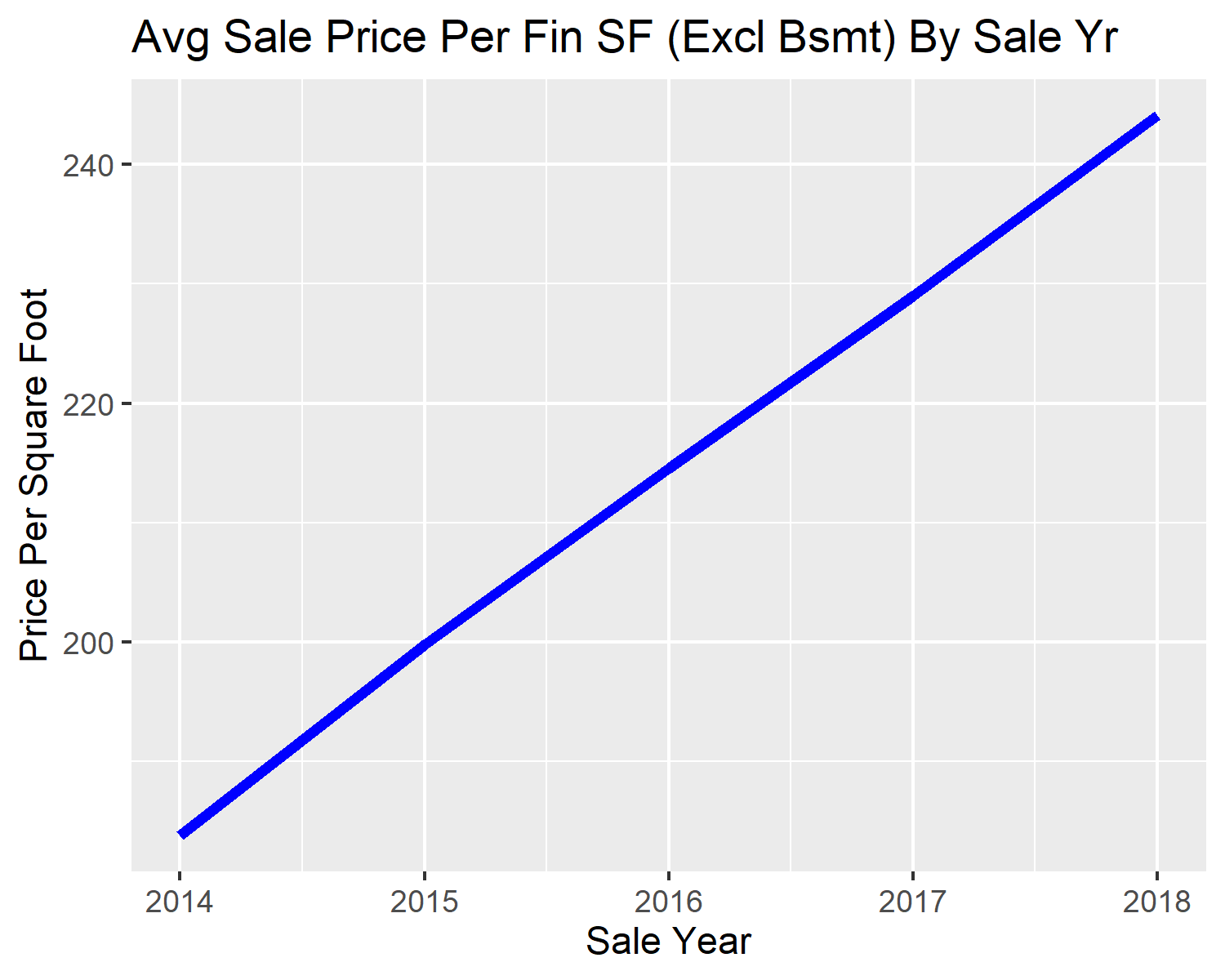


median\_price\_by\_year = dcSalesFinal %>%  
 group\_by(`Sale Year`) %>%  
 summarise(median\_price\_year = median(`Sale Price`)  
 )  
   
ggplot(median\_price\_by\_year, aes(x=`Sale Year`,y=`median\_price\_year`)) + xlab("Sale Year") + ylab("Median Sale Price") +  
 geom\_line(size=1.5, color="blue") +  
 labs(title="Median Sale Price By Sale Year")



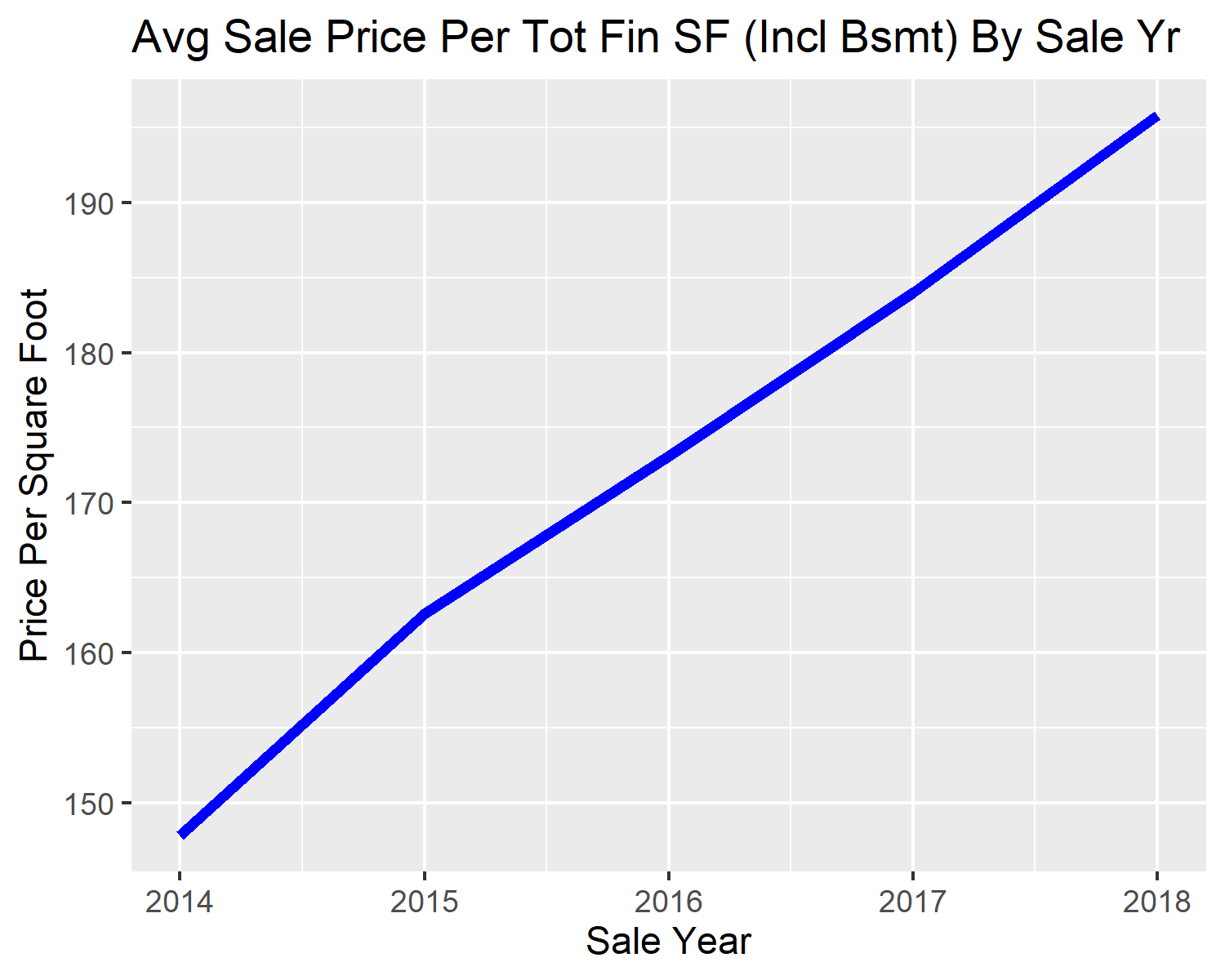
### Create columns for price for finished square foot (excludes basement) and create the average price for square foot and plot it.

dcSalesFinal$price\_home\_sqft <- dcSalesFinal$`Sale Price`/dcSalesFinal$`Improvmnt SF`  
   
ave\_price\_per\_sqft = dcSalesFinal %>%  
 group\_by(`Sale Year`) %>%  
 summarise(ave\_price\_per\_sqft = mean(price\_home\_sqft)  
 )  
   
ggplot(ave\_price\_per\_sqft, aes(x=`Sale Year`,y=ave\_price\_per\_sqft)) + xlab("Sale Year") + ylab("Price Per Square Foot") +  
 geom\_line(size=1.5, color="blue") +  
 labs(title="Avg Sale Price Per Fin SF (Excl Bsmt) By Sale Yr")



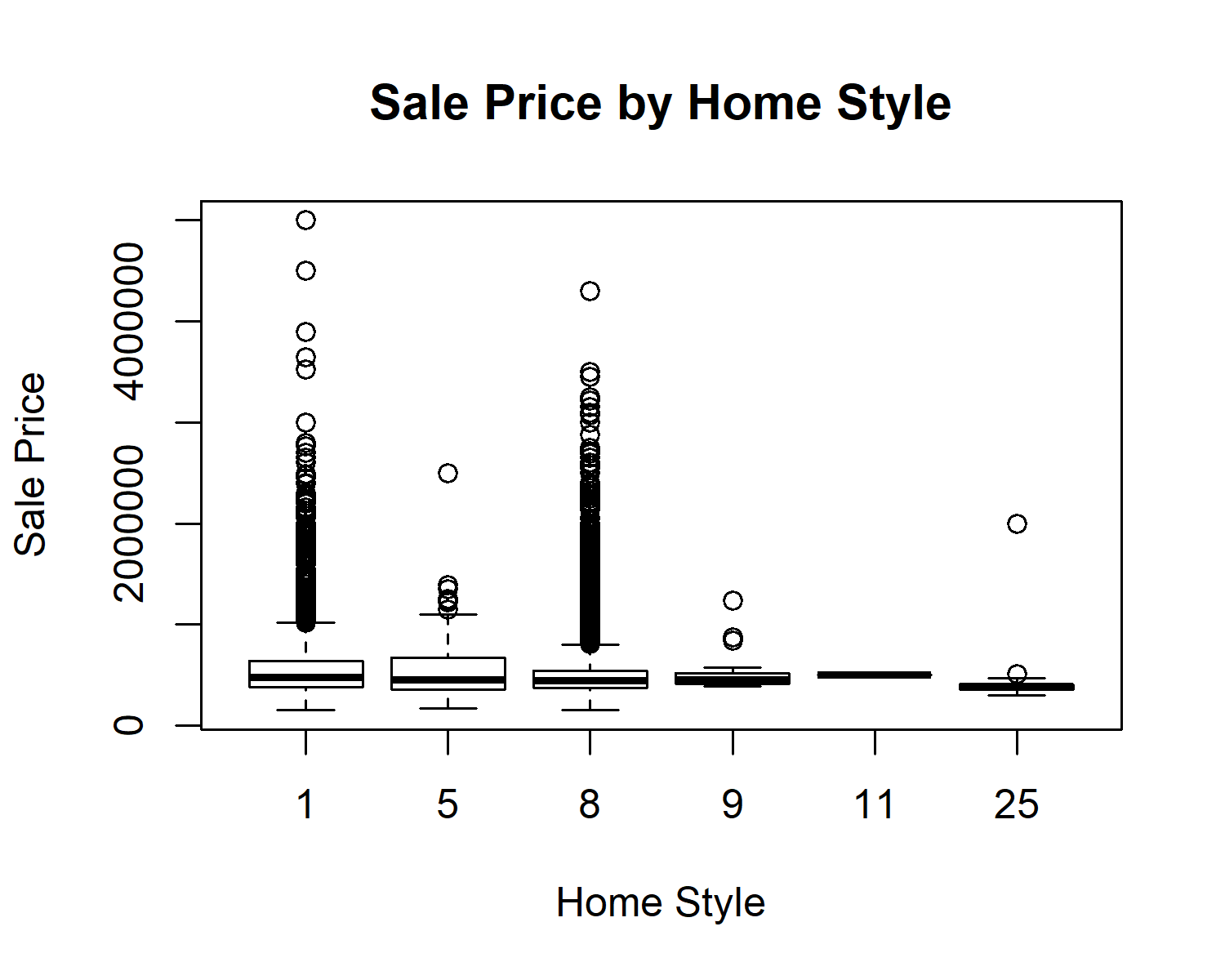
### Create columns for price for total finished square foot (includes basement) and create the average price for square foot and plot it.

dcSalesFinal$price\_tot\_home\_sqft <- dcSalesFinal$`Sale Price`/(dcSalesFinal$`Improvmnt SF` + dcSalesFinal$`Finished Basement SF`)  
   
# computes the average price per sq ft   
ave\_price\_tot\_sqft = dcSalesFinal %>%  
 group\_by(`Sale Year`) %>%  
 summarise(ave\_price\_tot\_sqft = mean(price\_tot\_home\_sqft)  
 )  
   
ggplot(ave\_price\_tot\_sqft, aes(x=`Sale Year`,y=ave\_price\_tot\_sqft)) + xlab("Sale Year") + ylab("Price Per Square Foot") +  
 geom\_line(size=1.5, color="blue") +  
 labs(title="Avg Sale Price Per Tot Fin SF (Incl Bsmt) By Sale Yr")

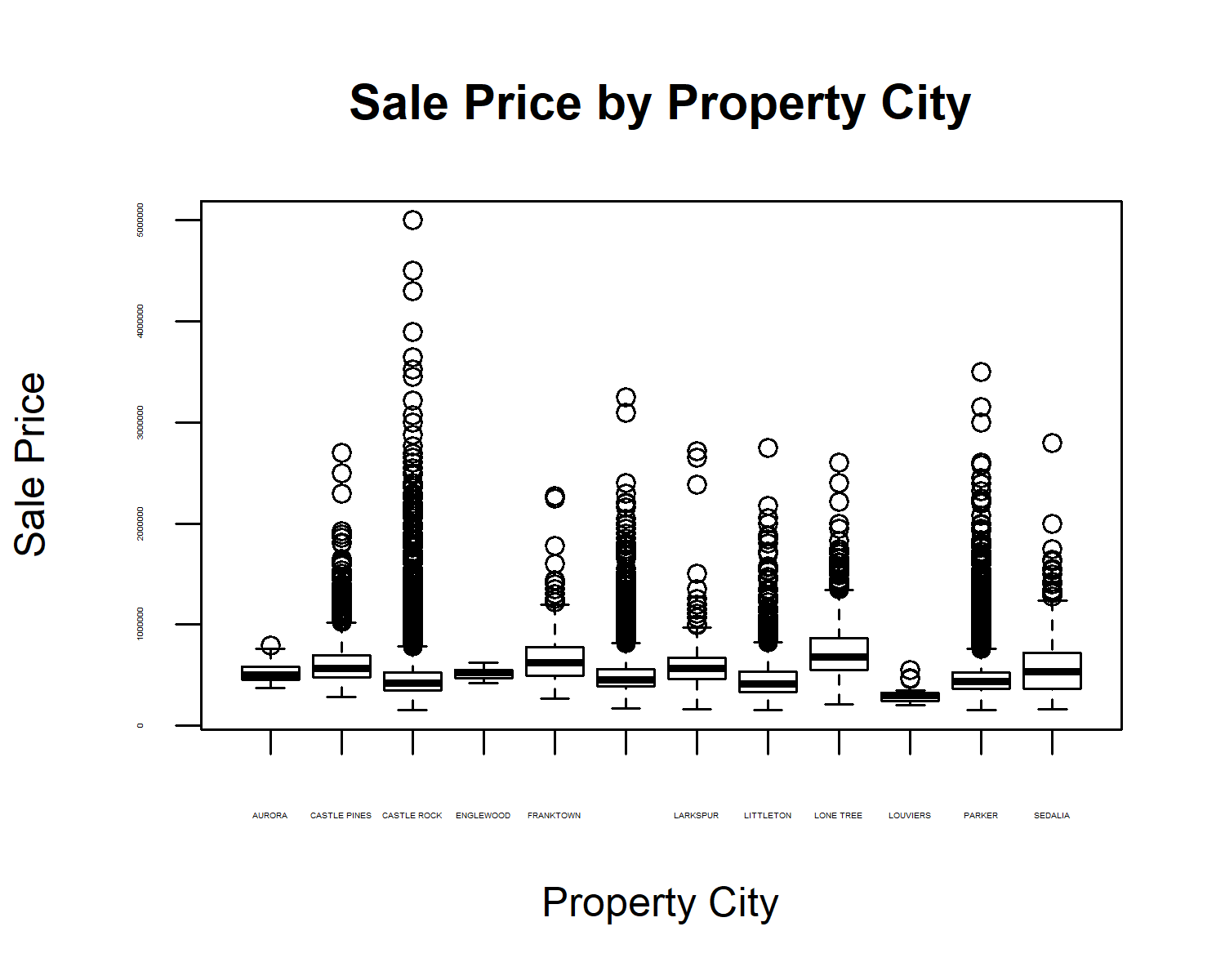


### Create box plot of some additional variables for evaluation.

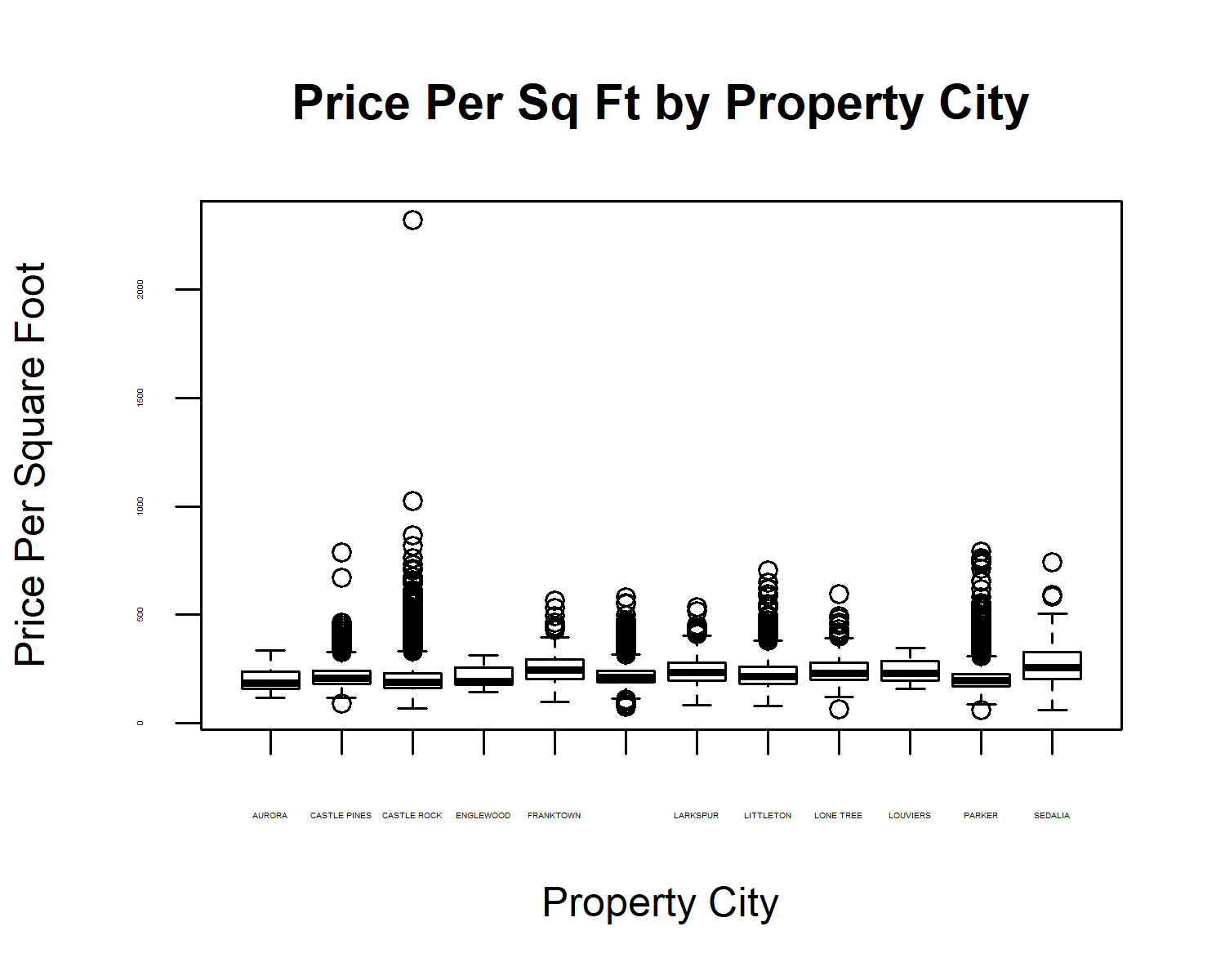
boxplot(`Sale Price` ~ Style, data = dcSalesFinal, xlab="Home Style", ylab="Sale Price",main="Sale Price by Home Style")



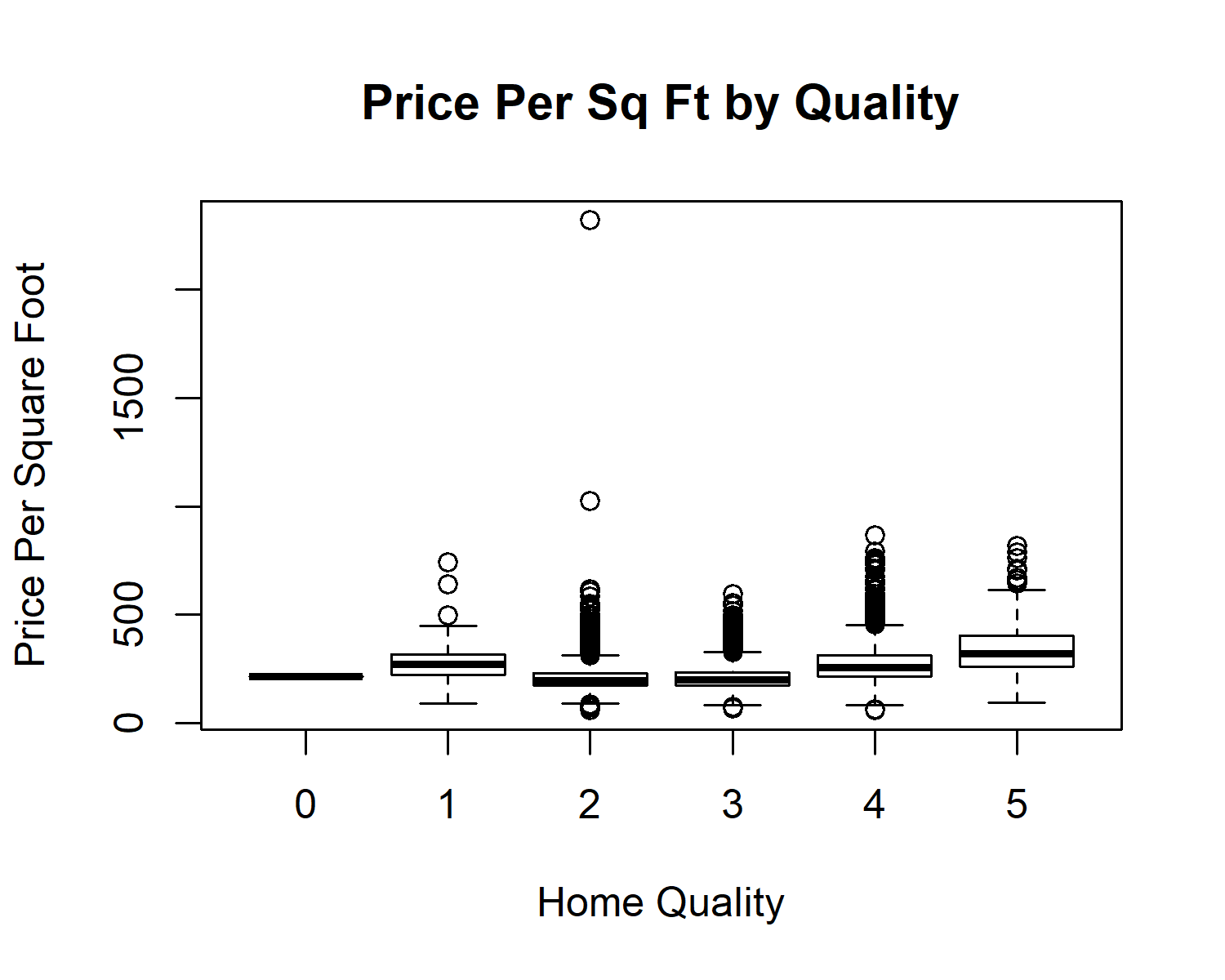
boxplot(`Sale Price` ~ `Situs City`, data = dcSalesFinal, xlab="Property City", ylab="Sale Price",main="Sale Price by Property City", cex.axis=.2)



boxplot(price\_home\_sqft ~ `Situs City`, data = dcSalesFinal, xlab="Property City", ylab="Price Per Square Foot",main="Price Per Sq Ft by Property City", cex.axis=.2)



boxplot(price\_home\_sqft ~ Quality, data = dcSalesFinal, xlab="Home Quality", ylab="Price Per Square Foot",main="Price Per Sq Ft by Quality")



### Create file containing only those independent variables used for predictions and the depending variable (Sales Price).

myvars = c("Acres", "Improvmnt SF", "Garage SF", "Basement SF", "Finished Basement SF", "Total Porch SF", "Year Built", "Situs Zip Code", "Sale Year", "Style", "Stories", "Bedroom Count", "Bathroom Count", "Quality", "Walkout Basement", "Sale Price")  
dcSalesPartial = dcSalesFinal[myvars]  
str(dcSalesPartial)

## 'data.frame': 28523 obs. of 16 variables:  
## $ Acres : num 7.2 1 1 1 1.01 ...  
## $ Improvmnt SF : num 3497 1619 1619 1194 2932 ...  
## $ Garage SF : num 1043 0 0 992 1044 ...  
## $ Basement SF : num 2477 1167 1167 858 0 ...  
## $ Finished Basement SF: num 2353 1108 1108 795 0 ...  
## $ Total Porch SF : num 105 1058 1058 902 12 ...  
## $ Year Built : num 1997 1978 1978 1950 1970 ...  
## $ Situs Zip Code : num 80125 80125 80125 80125 80125 ...  
## $ Sale Year : num 2017 2017 2018 2016 2014 ...  
## $ Style : num 1 1 1 1 1 8 1 1 1 1 ...  
## $ Stories : num 1 1 1 1 1 2 1 1 1 1 ...  
## $ Bedroom Count : num 3 1 1 3 3 3 2 2 3 1 ...  
## $ Bathroom Count : num 3 2 2 1 3 3 2 2 2 3 ...  
## $ Quality : num 4 2 2 2 3 3 2 2 2 2 ...  
## $ Walkout Basement : num 1 0 0 0 0 0 0 0 0 1 ...  
## $ Sale Price : num 1540000 520000 605000 655000 406000 640000 480000 370000 699000 460000 ...

summary(dcSalesPartial)

## Acres Improvmnt SF Garage SF Basement SF   
## Min. : 0.0600 Min. : 465 Min. : 0 Min. : 0   
## 1st Qu.: 0.1350 1st Qu.: 1833 1st Qu.: 440 1st Qu.: 760   
## Median : 0.1740 Median : 2281 Median : 584 Median :1138   
## Mean : 0.6205 Mean : 2406 Mean : 599 Mean :1226   
## 3rd Qu.: 0.2490 3rd Qu.: 2859 3rd Qu.: 685 3rd Qu.:1604   
## Max. :80.0000 Max. :10027 Max. :5455 Max. :7637   
## Finished Basement SF Total Porch SF Year Built Situs Zip Code   
## Min. : 0.0 Min. : 0.0 Min. :1867 Min. :80016   
## 1st Qu.: 0.0 1st Qu.: 239.0 1st Qu.:1995 1st Qu.:80109   
## Median : 523.0 Median : 385.0 Median :2000 Median :80126   
## Mean : 636.4 Mean : 496.6 Mean :2000 Mean :80123   
## 3rd Qu.:1099.0 3rd Qu.: 604.0 3rd Qu.:2005 3rd Qu.:80134   
## Max. :6451.0 Max. :11967.0 Max. :2018 Max. :80138   
## Sale Year Style Stories Bedroom Count   
## Min. :2014 Min. : 1.000 Min. :1.000 Min. :0.000   
## 1st Qu.:2015 1st Qu.: 8.000 1st Qu.:2.000 1st Qu.:3.000   
## Median :2016 Median : 8.000 Median :2.000 Median :3.000   
## Mean :2016 Mean : 6.586 Mean :1.795 Mean :3.331   
## 3rd Qu.:2017 3rd Qu.: 8.000 3rd Qu.:2.000 3rd Qu.:4.000   
## Max. :2018 Max. :25.000 Max. :3.000 Max. :8.000   
## Bathroom Count Quality Walkout Basement Sale Price   
## Min. :1.000 Min. :0.000 Min. :0.0000 Min. : 150000   
## 1st Qu.:3.000 1st Qu.:2.000 1st Qu.:0.0000 1st Qu.: 373500   
## Median :3.000 Median :2.000 Median :0.0000 Median : 449000   
## Mean :3.047 Mean :2.459 Mean :0.2775 Mean : 507284   
## 3rd Qu.:3.000 3rd Qu.:3.000 3rd Qu.:1.0000 3rd Qu.: 560200   
## Max. :9.000 Max. :5.000 Max. :1.0000 Max. :5000000

### Create initial model prior to change number data to factors.

dcSalesPartialLm <- lm(`Sale Price` ~ ., data = dcSalesPartial)  
confint(dcSalesPartialLm)

## 2.5 % 97.5 %  
## (Intercept) -107203619.32942 -89940918.56373  
## Acres 6796.81098 7902.31998  
## `Improvmnt SF` 82.92236 90.71698  
## `Garage SF` 51.75857 68.09423  
## `Basement SF` 22.33363 30.50647  
## `Finished Basement SF` 44.00529 49.43644  
## `Total Porch SF` 77.92541 86.01695  
## `Year Built` 1637.68462 1946.26798  
## `Situs Zip Code` 286.82636 495.39164  
## `Sale Year` 30614.49907 32529.92528  
## Style -422.76478 3210.78322  
## Stories -53763.23933 -25810.76640  
## `Bedroom Count` -31204.03153 -26571.90252  
## `Bathroom Count` 30679.71894 36885.80677  
## Quality 84660.09270 90814.40130  
## `Walkout Basement` -7966.87512 -914.80640

summary(dcSalesPartialLm)

##   
## Call:  
## lm(formula = `Sale Price` ~ ., data = dcSalesPartial)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1098688 -46082 871 39431 4072822   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -98572268.947 4403644.137 -22.384  
## Acres 7349.565 282.011 26.061  
## `Improvmnt SF` 86.820 1.988 43.664  
## `Garage SF` 59.926 4.167 14.381  
## `Basement SF` 26.420 2.085 12.672  
## `Finished Basement SF` 46.721 1.385 33.722  
## `Total Porch SF` 81.971 2.064 39.712  
## `Year Built` 1791.976 78.718 22.764  
## `Situs Zip Code` 391.109 53.204 7.351  
## `Sale Year` 31572.212 488.617 64.615  
## Style 1394.009 926.903 1.504  
## Stories -39787.003 7130.561 -5.580  
## `Bedroom Count` -28887.967 1181.637 -24.447  
## `Bathroom Count` 33782.763 1583.148 21.339  
## Quality 87737.247 1569.939 55.886  
## `Walkout Basement` -4440.841 1798.954 -2.469  
## Pr(>|t|)   
## (Intercept) < 0.0000000000000002 \*\*\*  
## Acres < 0.0000000000000002 \*\*\*  
## `Improvmnt SF` < 0.0000000000000002 \*\*\*  
## `Garage SF` < 0.0000000000000002 \*\*\*  
## `Basement SF` < 0.0000000000000002 \*\*\*  
## `Finished Basement SF` < 0.0000000000000002 \*\*\*  
## `Total Porch SF` < 0.0000000000000002 \*\*\*  
## `Year Built` < 0.0000000000000002 \*\*\*  
## `Situs Zip Code` 0.000000000000202 \*\*\*  
## `Sale Year` < 0.0000000000000002 \*\*\*  
## Style 0.1326   
## Stories 0.000000024300039 \*\*\*  
## `Bedroom Count` < 0.0000000000000002 \*\*\*  
## `Bathroom Count` < 0.0000000000000002 \*\*\*  
## Quality < 0.0000000000000002 \*\*\*  
## `Walkout Basement` 0.0136 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 115500 on 28507 degrees of freedom  
## Multiple R-squared: 0.7706, Adjusted R-squared: 0.7705   
## F-statistic: 6384 on 15 and 28507 DF, p-value: < 0.00000000000000022

### Change several columns to factor to give a more accurate evaluation then create model and evaluate.

dcSalesPartFix <- dcSalesPartial  
dcSalesPartFix$`Situs Zip Code` <- factor(dcSalesPartFix$`Situs Zip Code`)  
dcSalesPartFix$`Sale Year` <- factor(dcSalesPartFix$`Sale Year`)  
dcSalesPartFix$Style <- factor(dcSalesPartFix$Style)  
dcSalesPartFix$Stories <- factor(dcSalesPartFix$Stories)  
dcSalesPartFix$`Bedroom Count` <- factor(dcSalesPartFix$`Bedroom Count`)  
dcSalesPartFix$`Bathroom Count` <- factor(dcSalesPartFix$`Bathroom Count`)  
dcSalesPartFix$Quality <- factor(dcSalesPartFix$Quality)  
dcSalesPartFix$`Walkout Basement` <- factor(dcSalesPartFix$`Walkout Basement`)  
str(dcSalesPartFix)

## 'data.frame': 28523 obs. of 16 variables:  
## $ Acres : num 7.2 1 1 1 1.01 ...  
## $ Improvmnt SF : num 3497 1619 1619 1194 2932 ...  
## $ Garage SF : num 1043 0 0 992 1044 ...  
## $ Basement SF : num 2477 1167 1167 858 0 ...  
## $ Finished Basement SF: num 2353 1108 1108 795 0 ...  
## $ Total Porch SF : num 105 1058 1058 902 12 ...  
## $ Year Built : num 1997 1978 1978 1950 1970 ...  
## $ Situs Zip Code : Factor w/ 16 levels "80016","80104",..: 9 9 9 9 9 9 9 9 9 9 ...  
## $ Sale Year : Factor w/ 5 levels "2014","2015",..: 4 4 5 3 1 3 1 4 3 1 ...  
## $ Style : Factor w/ 6 levels "1","5","8","9",..: 1 1 1 1 1 3 1 1 1 1 ...  
## $ Stories : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 2 1 1 1 1 ...  
## $ Bedroom Count : Factor w/ 9 levels "0","1","2","3",..: 4 2 2 4 4 4 3 3 4 2 ...  
## $ Bathroom Count : Factor w/ 10 levels "1","2","3","3.5",..: 3 2 2 1 3 3 2 2 2 3 ...  
## $ Quality : Factor w/ 6 levels "0","1","2","3",..: 5 3 3 3 4 4 3 3 3 3 ...  
## $ Walkout Basement : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 2 ...  
## $ Sale Price : num 1540000 520000 605000 655000 406000 640000 480000 370000 699000 460000 ...

dcSalesPartFixLm <- lm(`Sale Price` ~ ., data = dcSalesPartFix)  
summary(dcSalesPartFixLm)

##   
## Call:  
## lm(formula = `Sale Price` ~ ., data = dcSalesPartFix)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1082249 -35066 -287 29444 4065972   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -4344066.246 189295.879 -22.949  
## Acres 8942.988 273.447 32.705  
## `Improvmnt SF` 84.159 1.817 46.317  
## `Garage SF` 78.319 3.746 20.909  
## `Basement SF` 28.044 1.860 15.079  
## `Finished Basement SF` 36.402 1.235 29.480  
## `Total Porch SF` 65.290 1.874 34.831  
## `Year Built` 2205.817 75.909 29.059  
## `Situs Zip Code`80104 2256.213 10615.618 0.213  
## `Situs Zip Code`80108 61688.449 10644.947 5.795  
## `Situs Zip Code`80109 15428.741 10598.015 1.456  
## `Situs Zip Code`80112 64008.512 24060.846 2.660  
## `Situs Zip Code`80116 -19461.485 12286.564 -1.584  
## `Situs Zip Code`80118 -29503.423 11302.849 -2.610  
## `Situs Zip Code`80124 123841.695 10916.966 11.344  
## `Situs Zip Code`80125 38839.913 10867.106 3.574  
## `Situs Zip Code`80126 94843.717 10618.608 8.932  
## `Situs Zip Code`80129 89683.591 10729.198 8.359  
## `Situs Zip Code`80130 73943.927 10701.642 6.910  
## `Situs Zip Code`80131 134306.988 25529.967 5.261  
## `Situs Zip Code`80134 24671.608 10522.790 2.345  
## `Situs Zip Code`80135 16802.468 12391.726 1.356  
## `Situs Zip Code`80138 25967.467 10672.019 2.433  
## `Sale Year`2015 34831.937 1876.948 18.558  
## `Sale Year`2016 61618.395 1910.636 32.250  
## `Sale Year`2017 91716.403 1900.635 48.256  
## `Sale Year`2018 128000.544 1937.810 66.054  
## Style5 -28979.635 28400.258 -1.020  
## Style8 -47048.856 29104.093 -1.617  
## Style9 -125805.356 39236.150 -3.206  
## Style11 59135.159 105841.679 0.559  
## Style25 -36166.633 39105.976 -0.925  
## Stories2 6503.282 29021.982 0.224  
## Stories3 -5075.013 39205.766 -0.129  
## `Bedroom Count`1 38014.223 51100.908 0.744  
## `Bedroom Count`2 -17818.546 50942.700 -0.350  
## `Bedroom Count`3 -45443.169 50932.016 -0.892  
## `Bedroom Count`4 -59406.356 50948.393 -1.166  
## `Bedroom Count`5 -74763.775 51018.673 -1.465  
## `Bedroom Count`6 -116672.590 52722.177 -2.213  
## `Bedroom Count`7 -170554.268 63942.914 -2.667  
## `Bedroom Count`8 -166439.340 85821.339 -1.939  
## `Bathroom Count`2 -21171.520 5607.586 -3.776  
## `Bathroom Count`3 3757.231 5960.880 0.630  
## `Bathroom Count`3.5 17848.671 101871.119 0.175  
## `Bathroom Count`4 14184.635 6579.287 2.156  
## `Bathroom Count`5 59997.477 7655.061 7.838  
## `Bathroom Count`6 210069.472 11191.062 18.771  
## `Bathroom Count`7 381287.388 22083.014 17.266  
## `Bathroom Count`8 126944.569 54019.430 2.350  
## `Bathroom Count`9 325184.382 100888.696 3.223  
## Quality1 93201.762 103833.883 0.898  
## Quality2 34179.804 101982.370 0.335  
## Quality3 64898.146 101998.229 0.636  
## Quality4 202446.952 102045.183 1.984  
## Quality5 536399.082 102230.005 5.247  
## `Walkout Basement`1 5957.858 1598.077 3.728  
## Pr(>|t|)   
## (Intercept) < 0.0000000000000002 \*\*\*  
## Acres < 0.0000000000000002 \*\*\*  
## `Improvmnt SF` < 0.0000000000000002 \*\*\*  
## `Garage SF` < 0.0000000000000002 \*\*\*  
## `Basement SF` < 0.0000000000000002 \*\*\*  
## `Finished Basement SF` < 0.0000000000000002 \*\*\*  
## `Total Porch SF` < 0.0000000000000002 \*\*\*  
## `Year Built` < 0.0000000000000002 \*\*\*  
## `Situs Zip Code`80104 0.831690   
## `Situs Zip Code`80108 0.00000000690030859 \*\*\*  
## `Situs Zip Code`80109 0.145455   
## `Situs Zip Code`80112 0.007812 \*\*   
## `Situs Zip Code`80116 0.113213   
## `Situs Zip Code`80118 0.009052 \*\*   
## `Situs Zip Code`80124 < 0.0000000000000002 \*\*\*  
## `Situs Zip Code`80125 0.000352 \*\*\*  
## `Situs Zip Code`80126 < 0.0000000000000002 \*\*\*  
## `Situs Zip Code`80129 < 0.0000000000000002 \*\*\*  
## `Situs Zip Code`80130 0.00000000000496295 \*\*\*  
## `Situs Zip Code`80131 0.00000014449926530 \*\*\*  
## `Situs Zip Code`80134 0.019055 \*   
## `Situs Zip Code`80135 0.175128   
## `Situs Zip Code`80138 0.014971 \*   
## `Sale Year`2015 < 0.0000000000000002 \*\*\*  
## `Sale Year`2016 < 0.0000000000000002 \*\*\*  
## `Sale Year`2017 < 0.0000000000000002 \*\*\*  
## `Sale Year`2018 < 0.0000000000000002 \*\*\*  
## Style5 0.307547   
## Style8 0.105982   
## Style9 0.001346 \*\*   
## Style11 0.576362   
## Style25 0.355059   
## Stories2 0.822696   
## Stories3 0.897006   
## `Bedroom Count`1 0.456940   
## `Bedroom Count`2 0.726509   
## `Bedroom Count`3 0.372276   
## `Bedroom Count`4 0.243620   
## `Bedroom Count`5 0.142817   
## `Bedroom Count`6 0.026908 \*   
## `Bedroom Count`7 0.007651 \*\*   
## `Bedroom Count`8 0.052466 .   
## `Bathroom Count`2 0.000160 \*\*\*  
## `Bathroom Count`3 0.528494   
## `Bathroom Count`3.5 0.860917   
## `Bathroom Count`4 0.031096 \*   
## `Bathroom Count`5 0.00000000000000475 \*\*\*  
## `Bathroom Count`6 < 0.0000000000000002 \*\*\*  
## `Bathroom Count`7 < 0.0000000000000002 \*\*\*  
## `Bathroom Count`8 0.018781 \*   
## `Bathroom Count`9 0.001269 \*\*   
## Quality1 0.369404   
## Quality2 0.737511   
## Quality3 0.524607   
## Quality4 0.047277 \*   
## Quality5 0.00000015571641858 \*\*\*  
## `Walkout Basement`1 0.000193 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 101700 on 28466 degrees of freedom  
## Multiple R-squared: 0.8226, Adjusted R-squared: 0.8223   
## F-statistic: 2358 on 56 and 28466 DF, p-value: < 0.00000000000000022

### Create a normalization function, test it, and then create a model and evaluate.

normalize <- function(x) {  
 return ((x - min(x)) / (max(x) - min(x)))  
}  
normalize(c(1,2,3,4,5))

## [1] 0.00 0.25 0.50 0.75 1.00

normalize(c(10, 20, 30, 40, 50))

## [1] 0.00 0.25 0.50 0.75 1.00

dcSalesPartFixNorm <- dcSalesPartFix  
dcSalesPartFixNorm[1:7] <- as.data.frame(lapply(dcSalesPartFixNorm[1:7], normalize))  
summary (dcSalesPartFixNorm)

## Acres Improvmnt SF Garage SF Basement SF   
## Min. :0.0000000 Min. :0.0000 Min. :0.00000 Min. :0.00000   
## 1st Qu.:0.0009382 1st Qu.:0.1431 1st Qu.:0.08066 1st Qu.:0.09952   
## Median :0.0014261 Median :0.1899 Median :0.10706 Median :0.14901   
## Mean :0.0070115 Mean :0.2030 Mean :0.10981 Mean :0.16050   
## 3rd Qu.:0.0023643 3rd Qu.:0.2504 3rd Qu.:0.12557 3rd Qu.:0.21003   
## Max. :1.0000000 Max. :1.0000 Max. :1.00000 Max. :1.00000   
##   
## Finished Basement SF Total Porch SF Year Built Situs Zip Code  
## Min. :0.00000 Min. :0.00000 Min. :0.0000 80134 :6254   
## 1st Qu.:0.00000 1st Qu.:0.01997 1st Qu.:0.8477 80126 :3445   
## Median :0.08107 Median :0.03217 Median :0.8808 80109 :3109   
## Mean :0.09865 Mean :0.04150 Mean :0.8784 80104 :2989   
## 3rd Qu.:0.17036 3rd Qu.:0.05047 3rd Qu.:0.9139 80108 :2568   
## Max. :1.00000 Max. :1.00000 Max. :1.0000 80138 :2216   
## (Other):7942   
## Sale Year Style Stories Bedroom Count Bathroom Count   
## 2014:5734 1 : 5944 1: 5957 3 :13260 3 :18324   
## 2015:6041 5 : 242 2:22444 4 :10202 2 : 4468   
## 2016:5642 8 :22197 3: 122 2 : 2734 4 : 4225   
## 2017:5764 9 : 22 5 : 1566 5 : 896   
## 2018:5342 11: 1 1 : 687 1 : 417   
## 25: 117 6 : 58 6 : 159   
## (Other): 16 (Other): 34   
## Quality Walkout Basement Sale Price   
## 0: 1 0:20608 Min. : 150000   
## 1: 25 1: 7915 1st Qu.: 373500   
## 2:18405 Median : 449000   
## 3: 7441 Mean : 507284   
## 4: 2283 3rd Qu.: 560200   
## 5: 368 Max. :5000000   
##

dcSalesPartFixNormLm <- lm(`Sale Price` ~ ., data = dcSalesPartFixNorm)  
confint(dcSalesPartFixNormLm)

## 2.5 % 97.5 %  
## (Intercept) -411054.918 38785.110  
## Acres 672057.135 757747.858  
## `Improvmnt SF` 770675.910 838785.618  
## `Garage SF` 387179.763 467280.081  
## `Basement SF` 186335.272 242015.962  
## `Finished Basement SF` 219215.554 250442.171  
## `Total Porch SF` 737356.182 825289.868  
## `Year Built` 310611.691 355545.103  
## `Situs Zip Code`80104 -18550.900 23063.327  
## `Situs Zip Code`80108 40823.849 82553.050  
## `Situs Zip Code`80109 -5343.870 36201.352  
## `Situs Zip Code`80112 16848.115 111168.909  
## `Situs Zip Code`80116 -43543.732 4620.762  
## `Situs Zip Code`80118 -51657.542 -7349.305  
## `Situs Zip Code`80124 102443.926 145239.464  
## `Situs Zip Code`80125 17539.871 60139.955  
## `Situs Zip Code`80126 74030.744 115656.691  
## `Situs Zip Code`80129 68653.856 110713.327  
## `Situs Zip Code`80130 52968.202 94919.651  
## `Situs Zip Code`80131 84267.045 184346.931  
## `Situs Zip Code`80134 4046.442 45296.774  
## `Situs Zip Code`80135 -7485.901 41090.837  
## `Situs Zip Code`80138 5049.805 46885.128  
## `Sale Year`2015 31153.031 38510.844  
## `Sale Year`2016 57873.458 65363.332  
## `Sale Year`2017 87991.069 95441.737  
## `Sale Year`2018 124202.345 131798.742  
## Style5 -84645.485 26686.216  
## Style8 -104094.255 9996.542  
## Style9 -202710.067 -48900.645  
## Style11 -148319.541 266589.860  
## Style25 -112816.196 40482.930  
## Stories2 -50381.176 63387.740  
## Stories3 -81920.170 71770.144  
## `Bedroom Count`1 -62145.976 138174.421  
## `Bedroom Count`2 -117668.649 82031.556  
## `Bedroom Count`3 -145272.330 54385.992  
## `Bedroom Count`4 -159267.617 40454.905  
## `Bedroom Count`5 -174762.789 25235.240  
## `Bedroom Count`6 -220010.551 -13334.629  
## `Bedroom Count`7 -295885.405 -45223.132  
## `Bedroom Count`8 -334653.226 1774.546  
## `Bathroom Count`2 -32162.655 -10180.385  
## `Bathroom Count`3 -7926.375 15440.837  
## `Bathroom Count`3.5 -181823.544 217520.886  
## `Bathroom Count`4 1288.922 27080.349  
## `Bathroom Count`5 44993.195 75001.758  
## `Bathroom Count`6 188134.461 232004.483  
## `Bathroom Count`7 338003.635 424571.141  
## `Bathroom Count`8 21063.930 232825.209  
## `Bathroom Count`9 127437.762 522931.001  
## Quality1 -110317.563 296721.088  
## Quality2 -165710.467 234070.075  
## Quality3 -135023.210 264819.502  
## Quality4 2433.564 402460.340  
## Quality5 336023.435 736774.730  
## `Walkout Basement`1 2825.552 9090.164

summary(dcSalesPartFixNormLm)

##   
## Call:  
## lm(formula = `Sale Price` ~ ., data = dcSalesPartFixNorm)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1082249 -35066 -287 29444 4065972   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -186135 114752 -1.622 0.104802  
## Acres 714903 21859 32.705 < 0.0000000000000002  
## `Improvmnt SF` 804731 17375 46.317 < 0.0000000000000002  
## `Garage SF` 427230 20433 20.909 < 0.0000000000000002  
## `Basement SF` 214176 14204 15.079 < 0.0000000000000002  
## `Finished Basement SF` 234829 7966 29.480 < 0.0000000000000002  
## `Total Porch SF` 781323 22432 34.831 < 0.0000000000000002  
## `Year Built` 333078 11462 29.059 < 0.0000000000000002  
## `Situs Zip Code`80104 2256 10616 0.213 0.831690  
## `Situs Zip Code`80108 61688 10645 5.795 0.00000000690030859  
## `Situs Zip Code`80109 15429 10598 1.456 0.145455  
## `Situs Zip Code`80112 64009 24061 2.660 0.007812  
## `Situs Zip Code`80116 -19462 12287 -1.584 0.113213  
## `Situs Zip Code`80118 -29503 11303 -2.610 0.009052  
## `Situs Zip Code`80124 123842 10917 11.344 < 0.0000000000000002  
## `Situs Zip Code`80125 38840 10867 3.574 0.000352  
## `Situs Zip Code`80126 94844 10619 8.932 < 0.0000000000000002  
## `Situs Zip Code`80129 89684 10729 8.359 < 0.0000000000000002  
## `Situs Zip Code`80130 73944 10702 6.910 0.00000000000496295  
## `Situs Zip Code`80131 134307 25530 5.261 0.00000014449926531  
## `Situs Zip Code`80134 24672 10523 2.345 0.019055  
## `Situs Zip Code`80135 16803 12392 1.356 0.175128  
## `Situs Zip Code`80138 25968 10672 2.433 0.014971  
## `Sale Year`2015 34832 1877 18.558 < 0.0000000000000002  
## `Sale Year`2016 61618 1911 32.250 < 0.0000000000000002  
## `Sale Year`2017 91716 1901 48.256 < 0.0000000000000002  
## `Sale Year`2018 128001 1938 66.054 < 0.0000000000000002  
## Style5 -28980 28400 -1.020 0.307547  
## Style8 -47049 29104 -1.617 0.105982  
## Style9 -125805 39236 -3.206 0.001346  
## Style11 59135 105842 0.559 0.576362  
## Style25 -36167 39106 -0.925 0.355059  
## Stories2 6503 29022 0.224 0.822696  
## Stories3 -5075 39206 -0.129 0.897006  
## `Bedroom Count`1 38014 51101 0.744 0.456940  
## `Bedroom Count`2 -17819 50943 -0.350 0.726509  
## `Bedroom Count`3 -45443 50932 -0.892 0.372276  
## `Bedroom Count`4 -59406 50948 -1.166 0.243620  
## `Bedroom Count`5 -74764 51019 -1.465 0.142817  
## `Bedroom Count`6 -116673 52722 -2.213 0.026908  
## `Bedroom Count`7 -170554 63943 -2.667 0.007651  
## `Bedroom Count`8 -166439 85821 -1.939 0.052466  
## `Bathroom Count`2 -21172 5608 -3.776 0.000160  
## `Bathroom Count`3 3757 5961 0.630 0.528494  
## `Bathroom Count`3.5 17849 101871 0.175 0.860917  
## `Bathroom Count`4 14185 6579 2.156 0.031096  
## `Bathroom Count`5 59998 7655 7.838 0.00000000000000475  
## `Bathroom Count`6 210070 11191 18.771 < 0.0000000000000002  
## `Bathroom Count`7 381287 22083 17.266 < 0.0000000000000002  
## `Bathroom Count`8 126945 54019 2.350 0.018781  
## `Bathroom Count`9 325184 100889 3.223 0.001269  
## Quality1 93202 103834 0.898 0.369404  
## Quality2 34180 101982 0.335 0.737511  
## Quality3 64898 101998 0.636 0.524607  
## Quality4 202447 102045 1.984 0.047277  
## Quality5 536399 102230 5.247 0.00000015571641858  
## `Walkout Basement`1 5958 1598 3.728 0.000193  
##   
## (Intercept)   
## Acres \*\*\*  
## `Improvmnt SF` \*\*\*  
## `Garage SF` \*\*\*  
## `Basement SF` \*\*\*  
## `Finished Basement SF` \*\*\*  
## `Total Porch SF` \*\*\*  
## `Year Built` \*\*\*  
## `Situs Zip Code`80104   
## `Situs Zip Code`80108 \*\*\*  
## `Situs Zip Code`80109   
## `Situs Zip Code`80112 \*\*   
## `Situs Zip Code`80116   
## `Situs Zip Code`80118 \*\*   
## `Situs Zip Code`80124 \*\*\*  
## `Situs Zip Code`80125 \*\*\*  
## `Situs Zip Code`80126 \*\*\*  
## `Situs Zip Code`80129 \*\*\*  
## `Situs Zip Code`80130 \*\*\*  
## `Situs Zip Code`80131 \*\*\*  
## `Situs Zip Code`80134 \*   
## `Situs Zip Code`80135   
## `Situs Zip Code`80138 \*   
## `Sale Year`2015 \*\*\*  
## `Sale Year`2016 \*\*\*  
## `Sale Year`2017 \*\*\*  
## `Sale Year`2018 \*\*\*  
## Style5   
## Style8   
## Style9 \*\*   
## Style11   
## Style25   
## Stories2   
## Stories3   
## `Bedroom Count`1   
## `Bedroom Count`2   
## `Bedroom Count`3   
## `Bedroom Count`4   
## `Bedroom Count`5   
## `Bedroom Count`6 \*   
## `Bedroom Count`7 \*\*   
## `Bedroom Count`8 .   
## `Bathroom Count`2 \*\*\*  
## `Bathroom Count`3   
## `Bathroom Count`3.5   
## `Bathroom Count`4 \*   
## `Bathroom Count`5 \*\*\*  
## `Bathroom Count`6 \*\*\*  
## `Bathroom Count`7 \*\*\*  
## `Bathroom Count`8 \*   
## `Bathroom Count`9 \*\*   
## Quality1   
## Quality2   
## Quality3   
## Quality4 \*   
## Quality5 \*\*\*  
## `Walkout Basement`1 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 101700 on 28466 degrees of freedom  
## Multiple R-squared: 0.8226, Adjusted R-squared: 0.8223   
## F-statistic: 2358 on 56 and 28466 DF, p-value: < 0.00000000000000022

### Create training and testing.

samp <- sample(nrow(dcSalesPartFix), 0.8 \* nrow(dcSalesPartial))  
dcSalesTrain <- dcSalesPartFix[samp, ]  
dcSalesTest <- dcSalesPartFix[-samp, ]

### Create Linear Regression model for evaluation.

modelLm <- lm(`Sale Price` ~ ., data = dcSalesTrain)  
summary (modelLm)

##   
## Call:  
## lm(formula = `Sale Price` ~ ., data = dcSalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1056496 -34702 -329 29498 2034350   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) -4453737.813 199057.715 -22.374  
## Acres 8083.868 297.382 27.183  
## `Improvmnt SF` 85.273 1.984 42.986  
## `Garage SF` 77.285 4.124 18.739  
## `Basement SF` 26.766 2.013 13.295  
## `Finished Basement SF` 37.718 1.342 28.105  
## `Total Porch SF` 64.797 2.019 32.092  
## `Year Built` 2262.708 83.006 27.260  
## `Situs Zip Code`80104 -3300.660 11690.705 -0.282  
## `Situs Zip Code`80108 55130.676 11716.604 4.705  
## `Situs Zip Code`80109 9593.118 11665.029 0.822  
## `Situs Zip Code`80112 59076.692 26000.666 2.272  
## `Situs Zip Code`80116 -21430.776 13529.614 -1.584  
## `Situs Zip Code`80118 -26583.825 12423.565 -2.140  
## `Situs Zip Code`80124 117494.791 12007.457 9.785  
## `Situs Zip Code`80125 34010.649 11956.238 2.845  
## `Situs Zip Code`80126 89613.261 11688.448 7.667  
## `Situs Zip Code`80129 84311.834 11811.813 7.138  
## `Situs Zip Code`80130 67969.067 11780.787 5.769  
## `Situs Zip Code`80131 139640.272 26706.413 5.229  
## `Situs Zip Code`80134 20052.991 11585.050 1.731  
## `Situs Zip Code`80135 14317.853 13639.482 1.050  
## `Situs Zip Code`80138 21788.258 11749.266 1.854  
## `Sale Year`2015 34315.963 2040.434 16.818  
## `Sale Year`2016 61401.050 2076.227 29.573  
## `Sale Year`2017 91565.590 2072.619 44.179  
## `Sale Year`2018 126701.464 2111.810 59.997  
## Style5 -46118.664 30111.461 -1.532  
## Style8 -61715.254 30847.305 -2.001  
## Style9 -132807.385 41263.057 -3.219  
## Style11 41595.211 103749.595 0.401  
## Style25 -42841.960 41719.780 -1.027  
## Stories2 20157.747 30760.789 0.655  
## Stories3 605.865 41754.359 0.015  
## `Bedroom Count`1 40508.862 49803.700 0.813  
## `Bedroom Count`2 -14391.218 49608.832 -0.290  
## `Bedroom Count`3 -43241.538 49595.422 -0.872  
## `Bedroom Count`4 -57818.233 49615.282 -1.165  
## `Bedroom Count`5 -74878.829 49700.438 -1.507  
## `Bedroom Count`6 -133786.480 51647.631 -2.590  
## `Bedroom Count`7 -165109.299 62286.350 -2.651  
## `Bedroom Count`8 -164505.850 83626.338 -1.967  
## `Bathroom Count`2 -22530.171 6264.414 -3.597  
## `Bathroom Count`3 3138.687 6637.380 0.473  
## `Bathroom Count`3.5 17251.386 99200.440 0.174  
## `Bathroom Count`4 13121.467 7300.754 1.797  
## `Bathroom Count`5 55617.893 8429.554 6.598  
## `Bathroom Count`6 213242.613 12331.074 17.293  
## `Bathroom Count`7 425394.013 23550.078 18.063  
## `Bathroom Count`8 125041.609 52801.449 2.368  
## `Bathroom Count`9 319362.447 98536.458 3.241  
## Quality1 88885.545 102058.876 0.871  
## Quality2 34760.103 99340.369 0.350  
## Quality3 65659.237 99359.004 0.661  
## Quality4 201325.120 99414.625 2.025  
## Quality5 537760.791 99639.323 5.397  
## `Walkout Basement`1 5568.472 1737.625 3.205  
## Pr(>|t|)   
## (Intercept) < 0.0000000000000002 \*\*\*  
## Acres < 0.0000000000000002 \*\*\*  
## `Improvmnt SF` < 0.0000000000000002 \*\*\*  
## `Garage SF` < 0.0000000000000002 \*\*\*  
## `Basement SF` < 0.0000000000000002 \*\*\*  
## `Finished Basement SF` < 0.0000000000000002 \*\*\*  
## `Total Porch SF` < 0.0000000000000002 \*\*\*  
## `Year Built` < 0.0000000000000002 \*\*\*  
## `Situs Zip Code`80104 0.777692   
## `Situs Zip Code`80108 0.0000025492486977 \*\*\*  
## `Situs Zip Code`80109 0.410868   
## `Situs Zip Code`80112 0.023088 \*   
## `Situs Zip Code`80116 0.113210   
## `Situs Zip Code`80118 0.032382 \*   
## `Situs Zip Code`80124 < 0.0000000000000002 \*\*\*  
## `Situs Zip Code`80125 0.004451 \*\*   
## `Situs Zip Code`80126 0.0000000000000183 \*\*\*  
## `Situs Zip Code`80129 0.0000000000009759 \*\*\*  
## `Situs Zip Code`80130 0.0000000080545184 \*\*\*  
## `Situs Zip Code`80131 0.0000001721983928 \*\*\*  
## `Situs Zip Code`80134 0.083477 .   
## `Situs Zip Code`80135 0.293851   
## `Situs Zip Code`80138 0.063690 .   
## `Sale Year`2015 < 0.0000000000000002 \*\*\*  
## `Sale Year`2016 < 0.0000000000000002 \*\*\*  
## `Sale Year`2017 < 0.0000000000000002 \*\*\*  
## `Sale Year`2018 < 0.0000000000000002 \*\*\*  
## Style5 0.125635   
## Style8 0.045440 \*   
## Style9 0.001290 \*\*   
## Style11 0.688483   
## Style25 0.304479   
## Stories2 0.512277   
## Stories3 0.988423   
## `Bedroom Count`1 0.416014   
## `Bedroom Count`2 0.771747   
## `Bedroom Count`3 0.383280   
## `Bedroom Count`4 0.243897   
## `Bedroom Count`5 0.131926   
## `Bedroom Count`6 0.009593 \*\*   
## `Bedroom Count`7 0.008035 \*\*   
## `Bedroom Count`8 0.049178 \*   
## `Bathroom Count`2 0.000323 \*\*\*  
## `Bathroom Count`3 0.636303   
## `Bathroom Count`3.5 0.861942   
## `Bathroom Count`4 0.072305 .   
## `Bathroom Count`5 0.0000000000426003 \*\*\*  
## `Bathroom Count`6 < 0.0000000000000002 \*\*\*  
## `Bathroom Count`7 < 0.0000000000000002 \*\*\*  
## `Bathroom Count`8 0.017886 \*   
## `Bathroom Count`9 0.001193 \*\*   
## Quality1 0.383805   
## Quality2 0.726410   
## Quality3 0.508729   
## Quality4 0.042868 \*   
## Quality5 0.0000000684132912 \*\*\*  
## `Walkout Basement`1 0.001354 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 98920 on 22761 degrees of freedom  
## Multiple R-squared: 0.8322, Adjusted R-squared: 0.8318   
## F-statistic: 2016 on 56 and 22761 DF, p-value: < 0.00000000000000022

modelLmPred <- predict(modelLm, newdata = dcSalesTest)  
modelLmPredActuals <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelLmPred)) # create modelLmPredActuals dataframe.  
corrAccLm <- cor(modelLmPredActuals) # 88.5% Accuracy  
corrAccLm

## actuals preds  
## actuals 1.0000000 0.8851628  
## preds 0.8851628 1.0000000

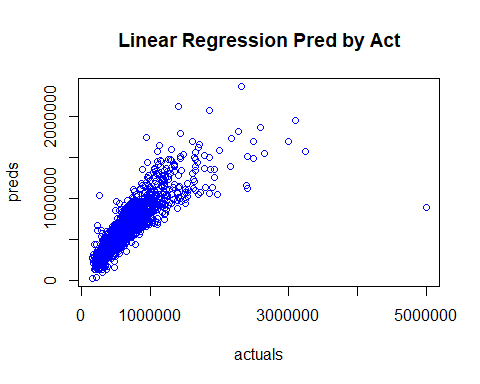
regr.eval(modelLmPredActuals$actuals, modelLmPredActuals$preds)

## mae mse rmse   
## 54144.45406297 12570422533.16545868 112117.89568648   
## mape   
## 0.09903332

head(modelLmPredActuals)

## actuals preds  
## 7 480000 379866.9  
## 8 370000 471432.5  
## 11 1336000 988496.3  
## 17 760000 648235.7  
## 22 1475000 1535521.0  
## 26 1700000 1622049.7

plot(modelLmPredActuals,main="Linear Regression Pred by Act",col="blue")



### Create Cross Validation Random Forest model for evaluation.

myControl = trainControl(method = "cv", number = 5, verboseIter = FALSE)  
modelRf = train(`Sale Price` ~ ., data = dcSalesTrain, tuneLength = 1, method = "ranger", importance = 'impurity', trControl = myControl)  
summary(modelRf)

## Length Class Mode   
## predictions 22818 -none- numeric   
## num.trees 1 -none- numeric   
## num.independent.variables 1 -none- numeric   
## mtry 1 -none- numeric   
## min.node.size 1 -none- numeric   
## variable.importance 56 -none- numeric   
## prediction.error 1 -none- numeric   
## forest 8 ranger.forest list   
## splitrule 1 -none- character  
## treetype 1 -none- character  
## r.squared 1 -none- numeric   
## call 9 -none- call   
## importance.mode 1 -none- character  
## num.samples 1 -none- numeric   
## replace 1 -none- logical   
## xNames 56 -none- character  
## problemType 1 -none- character  
## tuneValue 3 data.frame list   
## obsLevels 1 -none- logical   
## param 1 -none- list

modelRf

## Random Forest   
##   
## 22818 samples  
## 15 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 18252, 18255, 18255, 18255, 18255   
## Resampling results across tuning parameters:  
##   
## splitrule RMSE Rsquared MAE   
## variance 93117.70 0.8541681 48642.02  
## extratrees 96418.81 0.8449061 50796.23  
##   
## Tuning parameter 'mtry' was held constant at a value of 7  
## Tuning  
## parameter 'min.node.size' was held constant at a value of 5  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were mtry = 7, splitrule =  
## variance and min.node.size = 5.

t.test(dcSalesTrain$"Sale Price", modelRf$predictions)

##   
## One Sample t-test  
##   
## data: dcSalesTrain$"Sale Price"  
## t = 317.74, df = 22817, p-value < 0.00000000000000022  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 504235.0 510494.6  
## sample estimates:  
## mean of x   
## 507364.8

modelRfPred <- predict(modelRf, newdata = dcSalesTest)  
modelRfPredAct <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelRfPred)) # create modelRfPredAct dataframe.  
corrAccRf <- cor(modelRfPredAct) # 91.0% Accuracy  
corrAccRf

## actuals preds  
## actuals 1.0000000 0.9100423  
## preds 0.9100423 1.0000000

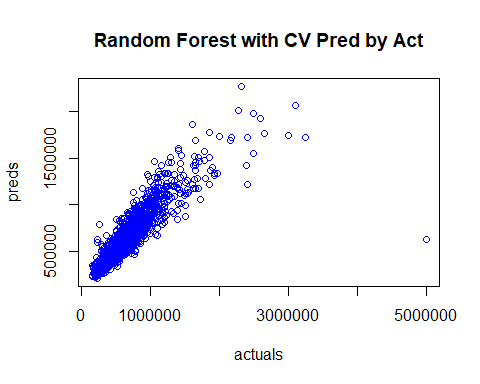
regr.eval(modelRfPredAct$actuals, modelRfPredAct$preds)

## mae mse rmse   
## 47770.26465398 10487215704.97301865 102407.10768776   
## mape   
## 0.08905617

head(modelRfPredAct)

## actuals preds  
## 1 480000 446391.6  
## 2 370000 466839.5  
## 3 1336000 911728.8  
## 4 760000 476873.1  
## 5 1475000 1312478.2  
## 6 1700000 1509312.1

plot(modelRfPredAct,main="Random Forest with CV Pred by Act",col="blue")



### Create Cross Validation Gradient Boosting Machine model for evaluation.

modelSgb = train(`Sale Price` ~ ., data = dcSalesTrain, tuneLength = 2, method = "gbm", trControl = myControl)

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 50: `Bathroom Count`9 has no variation.

## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 54102230929.6836 nan 0.1000 3664315833.6030  
## 2 50880156711.5388 nan 0.1000 3295052756.0048  
## 3 47950217635.8049 nan 0.1000 2944972323.5862  
## 4 45265177959.7983 nan 0.1000 2711932308.7724  
## 5 42877174972.8594 nan 0.1000 2443908171.0497  
## 6 40790675115.7024 nan 0.1000 2129774942.7309  
## 7 38856030004.9121 nan 0.1000 1882144992.3226  
## 8 37080901957.5111 nan 0.1000 1641938513.1495  
## 9 35395591348.0802 nan 0.1000 1684821931.8865  
## 10 33835726838.4507 nan 0.1000 1465012659.8940  
## 20 23917101737.5438 nan 0.1000 644950941.7185  
## 40 16281082616.4628 nan 0.1000 214285435.2792  
## 60 13664653792.0240 nan 0.1000 18538729.7782  
## 80 12394222631.2443 nan 0.1000 52444102.9206  
## 100 11573920359.0653 nan 0.1000 33820429.6749

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 50: `Bathroom Count`9 has no variation.

## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 52643559820.8618 nan 0.1000 5122962881.1450  
## 2 47522205916.1131 nan 0.1000 5269312844.9087  
## 3 43851203672.4251 nan 0.1000 3676214244.5976  
## 4 40544499654.9881 nan 0.1000 3298332450.0572  
## 5 37660497054.8413 nan 0.1000 2847698568.1956  
## 6 35041641485.7767 nan 0.1000 2506246847.0890  
## 7 32597915460.6496 nan 0.1000 2209764925.8951  
## 8 30776687644.5104 nan 0.1000 1486864188.9969  
## 9 29040872569.6946 nan 0.1000 1703649452.5752  
## 10 27080981779.6948 nan 0.1000 1800636682.7739  
## 20 17750679673.9148 nan 0.1000 575389063.7565  
## 40 12495211452.7226 nan 0.1000 90669559.4049  
## 60 10722512543.7146 nan 0.1000 6373428.5992  
## 80 9659448027.9752 nan 0.1000 44518418.2286  
## 100 9015368029.0625 nan 0.1000 33105936.1779  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 54114341703.2367 nan 0.1000 3593551856.0574  
## 2 50881853270.0362 nan 0.1000 3180890316.7567  
## 3 47967007211.8439 nan 0.1000 2964736119.6959  
## 4 45333722531.3343 nan 0.1000 2659251858.6218  
## 5 42947651578.1616 nan 0.1000 2278206542.5464  
## 6 40863843500.1806 nan 0.1000 2068432029.7254  
## 7 38941519492.7806 nan 0.1000 1861322807.9085  
## 8 37181832608.8569 nan 0.1000 1797894794.5558  
## 9 35630482537.0807 nan 0.1000 1581149894.2313  
## 10 34181497647.8095 nan 0.1000 1483636797.6293  
## 20 23924670728.7856 nan 0.1000 682998769.2922  
## 40 16135196650.7349 nan 0.1000 214965393.5438  
## 60 13453490200.8684 nan 0.1000 72343856.9239  
## 80 12228489056.4992 nan 0.1000 5071778.3632  
## 100 11397328803.6470 nan 0.1000 36554980.9980  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 52924480856.0411 nan 0.1000 4894592165.6486  
## 2 48605672659.7917 nan 0.1000 4056363957.3677  
## 3 44070195473.5739 nan 0.1000 4644856509.1923  
## 4 40336779172.0873 nan 0.1000 3841653224.1579  
## 5 37329226365.4706 nan 0.1000 2820984144.5413  
## 6 34767273193.1734 nan 0.1000 2698833527.0429  
## 7 32445615729.2668 nan 0.1000 2337427195.6889  
## 8 30234868033.2953 nan 0.1000 2010140180.5997  
## 9 28402821290.5828 nan 0.1000 1856355053.0305  
## 10 26819532686.8833 nan 0.1000 1525397841.3087  
## 20 17703980408.6188 nan 0.1000 502396076.5957  
## 40 12517391936.2282 nan 0.1000 55600723.2910  
## 60 10719602790.2756 nan 0.1000 29590437.3205  
## 80 9654851728.4779 nan 0.1000 -5219280.3398  
## 100 9125377887.7510 nan 0.1000 31959163.5165  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 54906924415.4299 nan 0.1000 3676390399.1981  
## 2 51713260139.4738 nan 0.1000 3275650583.6458  
## 3 48614488155.8739 nan 0.1000 3060257239.5190  
## 4 45884276094.1340 nan 0.1000 2725911802.2071  
## 5 43421971532.6795 nan 0.1000 2244842566.8751  
## 6 41273062473.0939 nan 0.1000 2096860004.8057  
## 7 39185391429.0690 nan 0.1000 1900967393.8785  
## 8 37357477423.0996 nan 0.1000 1806604631.8412  
## 9 35704685450.4681 nan 0.1000 1671230593.5201  
## 10 34177121551.8374 nan 0.1000 1358494159.8815  
## 20 23965137067.0802 nan 0.1000 684028227.8830  
## 40 16219827296.2317 nan 0.1000 210921620.6399  
## 60 13658261630.2920 nan 0.1000 18886567.1146  
## 80 12481139618.3556 nan 0.1000 43439313.7464  
## 100 11660658387.1685 nan 0.1000 15726565.8718  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 53699674787.7434 nan 0.1000 4811338086.6309  
## 2 49099631335.3536 nan 0.1000 4442336232.5244  
## 3 45088591090.6971 nan 0.1000 3921363144.7369  
## 4 41780040577.2913 nan 0.1000 3349830977.0749  
## 5 37936151615.1752 nan 0.1000 3643244383.5404  
## 6 35278158473.5774 nan 0.1000 2585867781.9087  
## 7 33009252638.1050 nan 0.1000 2037535410.0368  
## 8 30886007443.1868 nan 0.1000 2030428876.4141  
## 9 29057586249.1633 nan 0.1000 1780657015.7843  
## 10 27211854557.6411 nan 0.1000 1854437020.0964  
## 20 17831374308.8487 nan 0.1000 597657530.8384  
## 40 12733342825.5683 nan 0.1000 138518365.4285  
## 60 10884197145.4854 nan 0.1000 45276288.8865  
## 80 10026847438.1565 nan 0.1000 -39253945.3449  
## 100 9433791098.6969 nan 0.1000 -551722.7660

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 30: Style11 has no variation.

## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 55116735419.7814 nan 0.1000 3528853029.6890  
## 2 51791005143.3790 nan 0.1000 3288863065.9227  
## 3 48908241585.1663 nan 0.1000 2970770669.8579  
## 4 46259338918.1955 nan 0.1000 2462086667.4072  
## 5 43684410003.5812 nan 0.1000 2476109889.0862  
## 6 41560778170.3123 nan 0.1000 2180726880.0380  
## 7 39581166067.3286 nan 0.1000 1999432926.7516  
## 8 37808169381.1979 nan 0.1000 1852567843.7604  
## 9 36164004347.9191 nan 0.1000 1635898638.1795  
## 10 34599451092.0233 nan 0.1000 1585827422.6709  
## 20 24372466117.0038 nan 0.1000 714211388.4110  
## 40 16593893446.6809 nan 0.1000 183497343.5865  
## 60 14005430081.9227 nan 0.1000 98770043.8412  
## 80 12763441404.6000 nan 0.1000 11426349.6744  
## 100 11929308466.4596 nan 0.1000 39005778.9130

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 30: Style11 has no variation.

## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 53503913154.6180 nan 0.1000 5133771602.3973  
## 2 48247559669.1735 nan 0.1000 5374898042.2208  
## 3 44631775159.4483 nan 0.1000 3661576829.1111  
## 4 41135698164.9781 nan 0.1000 3548459465.2100  
## 5 38371890668.2273 nan 0.1000 2522016884.3648  
## 6 35321825940.4766 nan 0.1000 3034300889.3117  
## 7 32864964126.3081 nan 0.1000 2313326806.9338  
## 8 30796467676.8725 nan 0.1000 2005584594.6138  
## 9 28813535054.3827 nan 0.1000 2013377027.6967  
## 10 27203728622.3798 nan 0.1000 1536708586.2313  
## 20 18053617980.0157 nan 0.1000 461994313.1588  
## 40 12989307571.2632 nan 0.1000 158014453.9473  
## 60 11075634275.7327 nan 0.1000 66228437.9758  
## 80 9981903712.4823 nan 0.1000 72493739.6626  
## 100 9317657360.2814 nan 0.1000 -34977552.4585

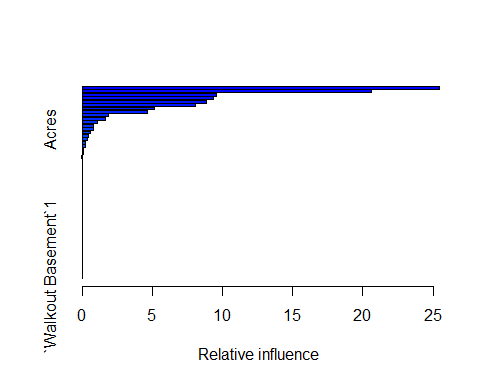
## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 44: `Bathroom Count`3.5 has no variation.

## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 54828422052.8725 nan 0.1000 3481798287.3059  
## 2 51582831356.4587 nan 0.1000 3248766179.1691  
## 3 48516664007.4153 nan 0.1000 2990424803.4736  
## 4 45877854484.4543 nan 0.1000 2697233280.2752  
## 5 43544640896.7760 nan 0.1000 2138396182.8199  
## 6 41550944521.5449 nan 0.1000 1949769773.9396  
## 7 39607519728.0402 nan 0.1000 1953127584.7588  
## 8 37885696600.0067 nan 0.1000 1756017916.9323  
## 9 36032650489.6807 nan 0.1000 1757028829.2680  
## 10 34529112695.7407 nan 0.1000 1552666149.2626  
## 20 24231930299.2349 nan 0.1000 729244498.8706  
## 40 16421731379.9666 nan 0.1000 214837272.8957  
## 60 13676286082.3117 nan 0.1000 80072031.5254  
## 80 12452981689.5945 nan 0.1000 10325230.7455  
## 100 11713838535.1853 nan 0.1000 22840802.9461

## Warning in (function (x, y, offset = NULL, misc = NULL, distribution =  
## "bernoulli", : variable 44: `Bathroom Count`3.5 has no variation.

## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 53322135875.4201 nan 0.1000 4890526996.6919  
## 2 48975090715.8535 nan 0.1000 4301226377.6846  
## 3 44424110989.2046 nan 0.1000 4451206148.8210  
## 4 41096294535.8717 nan 0.1000 3459077022.8660  
## 5 37676946916.2388 nan 0.1000 3400984575.9795  
## 6 35256989085.8137 nan 0.1000 2284418077.6923  
## 7 32822879943.6194 nan 0.1000 2225966757.8058  
## 8 30650558897.6374 nan 0.1000 2073753176.4833  
## 9 28869194012.7474 nan 0.1000 1772739252.4681  
## 10 27381590352.4274 nan 0.1000 1472857700.2459  
## 20 17795438023.1786 nan 0.1000 536932695.5340  
## 40 12626363701.7245 nan 0.1000 113874486.2656  
## 60 11027336141.9800 nan 0.1000 -26729242.0188  
## 80 9933569243.9552 nan 0.1000 7961898.8442  
## 100 9323946682.4726 nan 0.1000 -1588890.2126  
##   
## Iter TrainDeviance ValidDeviance StepSize Improve  
## 1 52919675014.3993 nan 0.1000 5112915075.8483  
## 2 48699370081.5876 nan 0.1000 4195123717.6396  
## 3 44793993150.4670 nan 0.1000 3928938202.2912  
## 4 41526426776.9200 nan 0.1000 3070754119.7672  
## 5 37967535851.2891 nan 0.1000 3609251619.1467  
## 6 35390299534.7209 nan 0.1000 2300621214.9092  
## 7 32974376787.0363 nan 0.1000 2312943900.8076  
## 8 31081392802.4992 nan 0.1000 1792524586.5081  
## 9 29308904404.3632 nan 0.1000 1622812659.9655  
## 10 27347706278.7348 nan 0.1000 1975921090.1082  
## 20 18128880780.9838 nan 0.1000 411579740.6292  
## 40 12869246142.0605 nan 0.1000 71996840.9519  
## 60 11035324456.3535 nan 0.1000 54516922.7759  
## 80 10103893108.5714 nan 0.1000 -22446057.8782  
## 100 9473225717.2818 nan 0.1000 52612277.8469

summary(modelSgb)



## var rel.inf  
## `Improvmnt SF` `Improvmnt SF` 25.46812098  
## `Basement SF` `Basement SF` 20.61238658  
## Quality2 Quality2 9.53150653  
## `Garage SF` `Garage SF` 9.36338559  
## `Finished Basement SF` `Finished Basement SF` 8.86576950  
## Quality5 Quality5 8.03220572  
## `Total Porch SF` `Total Porch SF` 5.17971736  
## Quality3 Quality3 4.62163341  
## `Sale Year`2018 `Sale Year`2018 1.88365025  
## `Year Built` `Year Built` 1.63153464  
## Quality4 Quality4 1.10517703  
## `Sale Year`2017 `Sale Year`2017 0.78614264  
## Acres Acres 0.76438106  
## `Situs Zip Code`80126 `Situs Zip Code`80126 0.57586183  
## `Situs Zip Code`80108 `Situs Zip Code`80108 0.46434005  
## `Situs Zip Code`80124 `Situs Zip Code`80124 0.39987673  
## `Situs Zip Code`80104 `Situs Zip Code`80104 0.26080093  
## `Bedroom Count`1 `Bedroom Count`1 0.22192431  
## `Situs Zip Code`80129 `Situs Zip Code`80129 0.11261579  
## `Bathroom Count`7 `Bathroom Count`7 0.05147843  
## `Situs Zip Code`80134 `Situs Zip Code`80134 0.03565500  
## `Sale Year`2016 `Sale Year`2016 0.03183562  
## `Situs Zip Code`80109 `Situs Zip Code`80109 0.00000000  
## `Situs Zip Code`80112 `Situs Zip Code`80112 0.00000000  
## `Situs Zip Code`80116 `Situs Zip Code`80116 0.00000000  
## `Situs Zip Code`80118 `Situs Zip Code`80118 0.00000000  
## `Situs Zip Code`80125 `Situs Zip Code`80125 0.00000000  
## `Situs Zip Code`80130 `Situs Zip Code`80130 0.00000000  
## `Situs Zip Code`80131 `Situs Zip Code`80131 0.00000000  
## `Situs Zip Code`80135 `Situs Zip Code`80135 0.00000000  
## `Situs Zip Code`80138 `Situs Zip Code`80138 0.00000000  
## `Sale Year`2015 `Sale Year`2015 0.00000000  
## Style5 Style5 0.00000000  
## Style8 Style8 0.00000000  
## Style9 Style9 0.00000000  
## Style11 Style11 0.00000000  
## Style25 Style25 0.00000000  
## Stories2 Stories2 0.00000000  
## Stories3 Stories3 0.00000000  
## `Bedroom Count`2 `Bedroom Count`2 0.00000000  
## `Bedroom Count`3 `Bedroom Count`3 0.00000000  
## `Bedroom Count`4 `Bedroom Count`4 0.00000000  
## `Bedroom Count`5 `Bedroom Count`5 0.00000000  
## `Bedroom Count`6 `Bedroom Count`6 0.00000000  
## `Bedroom Count`7 `Bedroom Count`7 0.00000000  
## `Bedroom Count`8 `Bedroom Count`8 0.00000000  
## `Bathroom Count`2 `Bathroom Count`2 0.00000000  
## `Bathroom Count`3 `Bathroom Count`3 0.00000000  
## `Bathroom Count`3.5 `Bathroom Count`3.5 0.00000000  
## `Bathroom Count`4 `Bathroom Count`4 0.00000000  
## `Bathroom Count`5 `Bathroom Count`5 0.00000000  
## `Bathroom Count`6 `Bathroom Count`6 0.00000000  
## `Bathroom Count`8 `Bathroom Count`8 0.00000000  
## `Bathroom Count`9 `Bathroom Count`9 0.00000000  
## Quality1 Quality1 0.00000000  
## `Walkout Basement`1 `Walkout Basement`1 0.00000000

modelSgb

## Stochastic Gradient Boosting   
##   
## 22818 samples  
## 15 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 18254, 18254, 18255, 18253, 18256   
## Resampling results across tuning parameters:  
##   
## interaction.depth n.trees RMSE Rsquared MAE   
## 1 50 123152.1 0.7646627 74080.60  
## 1 100 110739.2 0.7944598 64836.00  
## 2 50 110569.2 0.7986273 65279.95  
## 2 100 100772.7 0.8277398 57876.06  
##   
## Tuning parameter 'shrinkage' was held constant at a value of 0.1  
##   
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were n.trees = 100,  
## interaction.depth = 2, shrinkage = 0.1 and n.minobsinnode = 10.

t.test(dcSalesTrain$"Sale Price", modelSgb$predictions)

##   
## One Sample t-test  
##   
## data: dcSalesTrain$"Sale Price"  
## t = 317.74, df = 22817, p-value < 0.00000000000000022  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 504235.0 510494.6  
## sample estimates:  
## mean of x   
## 507364.8

modelSgbPred <- predict(modelSgb, newdata = dcSalesTest)  
modelSgbPredAct <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelSgbPred)) # create modelSgbPredAct dataframe.  
corrAccSgb <- cor(modelSgbPredAct) # 88.9% Accuracy  
corrAccSgb

## actuals preds  
## actuals 1.0000000 0.8890776  
## preds 0.8890776 1.0000000

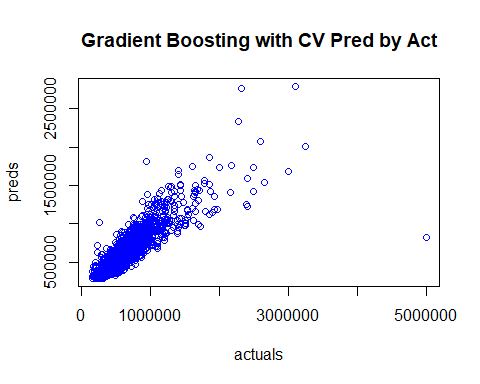
regr.eval(modelSgbPredAct$actuals, modelSgbPredAct$preds)

## mae mse rmse   
## 58868.3399881 12468530427.9425468 111662.5739805   
## mape   
## 0.1129083

head(modelSgbPredAct)

## actuals preds  
## 1 480000 396846.3  
## 2 370000 431927.2  
## 3 1336000 950630.3  
## 4 760000 615290.4  
## 5 1475000 1273283.7  
## 6 1700000 1304400.0

plot(modelSgbPredAct,main="Gradient Boosting with CV Pred by Act",col="blue")



### Create Extreme Gradient Boosting model for evaluation.

xgbTuneGrid = expand.grid(nrounds = c(50, 100), lambda = seq(0.1, 0.5, 0.1), alpha = seq(0.1, 0.5, 0.1), eta = c(0.3, 0.4))  
modelXgb = train(`Sale Price` ~ ., data = dcSalesTrain, tuneLength = 3, method = "xgbLinear", trControl = myControl, tunegrid = xgbTuneGrid)  
summary(modelXgb)

## Length Class Mode   
## handle 1 xgb.Booster.handle externalptr  
## raw 392119 -none- raw   
## niter 1 -none- numeric   
## call 6 -none- call   
## params 5 -none- list   
## callbacks 1 -none- list   
## feature\_names 56 -none- character   
## nfeatures 1 -none- numeric   
## xNames 56 -none- character   
## problemType 1 -none- character   
## tuneValue 4 data.frame list   
## obsLevels 1 -none- logical   
## param 1 -none- list

modelXgb

## eXtreme Gradient Boosting   
##   
## 22818 samples  
## 15 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 18254, 18254, 18255, 18256, 18253   
## Resampling results across tuning parameters:  
##   
## lambda alpha nrounds RMSE Rsquared MAE   
## 0.0000 0.0000 50 90516.73 0.8594551 43221.11  
## 0.0000 0.0000 100 90537.32 0.8594794 42584.82  
## 0.0000 0.0000 150 90768.27 0.8587936 42582.35  
## 0.0000 0.0001 50 90516.73 0.8594551 43221.11  
## 0.0000 0.0001 100 90537.32 0.8594794 42584.82  
## 0.0000 0.0001 150 90768.27 0.8587936 42582.35  
## 0.0000 0.1000 50 90516.73 0.8594551 43221.11  
## 0.0000 0.1000 100 90537.31 0.8594794 42584.82  
## 0.0000 0.1000 150 90768.26 0.8587936 42582.35  
## 0.0001 0.0000 50 90464.49 0.8596799 43218.09  
## 0.0001 0.0000 100 90410.68 0.8600219 42589.64  
## 0.0001 0.0000 150 90582.50 0.8595512 42555.19  
## 0.0001 0.0001 50 90464.49 0.8596799 43218.09  
## 0.0001 0.0001 100 90410.68 0.8600219 42589.64  
## 0.0001 0.0001 150 90582.50 0.8595512 42555.19  
## 0.0001 0.1000 50 90464.48 0.8596799 43218.09  
## 0.0001 0.1000 100 90410.68 0.8600219 42589.64  
## 0.0001 0.1000 150 90582.50 0.8595512 42555.19  
## 0.1000 0.0000 50 90499.12 0.8604517 43523.37  
## 0.1000 0.0000 100 90036.64 0.8618539 42774.62  
## 0.1000 0.0000 150 90090.21 0.8617003 42587.11  
## 0.1000 0.0001 50 90499.12 0.8604517 43523.37  
## 0.1000 0.0001 100 90036.64 0.8618539 42774.62  
## 0.1000 0.0001 150 90090.21 0.8617003 42587.11  
## 0.1000 0.1000 50 90499.12 0.8604517 43523.37  
## 0.1000 0.1000 100 90036.64 0.8618539 42774.62  
## 0.1000 0.1000 150 90090.20 0.8617003 42587.11  
##   
## Tuning parameter 'eta' was held constant at a value of 0.3  
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were nrounds = 100, lambda =  
## 0.1, alpha = 0.1 and eta = 0.3.

t.test(dcSalesTrain$"Sale Price", modelXgb$predictions)

##   
## One Sample t-test  
##   
## data: dcSalesTrain$"Sale Price"  
## t = 317.74, df = 22817, p-value < 0.00000000000000022  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 504235.0 510494.6  
## sample estimates:  
## mean of x   
## 507364.8

modelXgbPred <- predict(modelXgb, newdata = dcSalesTest)  
modelXgbPredAct <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelXgbPred)) # create modelXgbPredAct dataframe.  
corrAccXgb <- cor(modelXgbPredAct) # 92.2% Accuracy  
corrAccXgb

## actuals preds  
## actuals 1.0000000 0.9219998  
## preds 0.9219998 1.0000000

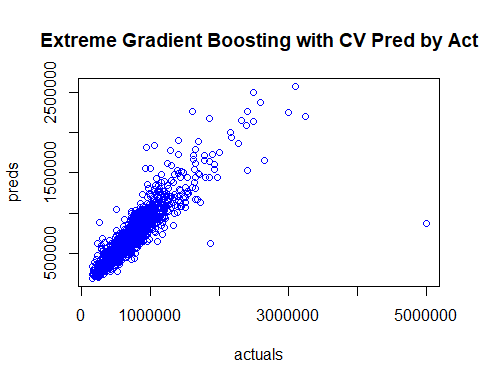
regr.eval(modelXgbPredAct$actuals, modelXgbPredAct$preds)

## mae mse rmse   
## 41414.58451194 8728859479.82512474 93428.36549906   
## mape   
## 0.07448972

head(modelXgbPredAct)

## actuals preds  
## 1 480000 467544.0  
## 2 370000 526126.9  
## 3 1336000 852502.0  
## 4 760000 439452.1  
## 5 1475000 1509719.1  
## 6 1700000 1885874.8

plot(modelXgbPredAct,main="Extreme Gradient Boosting with CV Pred by Act",col="blue")



### Create Support Vector Machine (SVM) model for evaluation.

modelSvm = svm(`Sale Price`~., data = dcSalesTrain, kernel="linear", scale = FALSE)  
summary(modelSvm)

##   
## Call:  
## svm(formula = `Sale Price` ~ ., data = dcSalesTrain, kernel = "linear",   
## scale = FALSE)  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.01754386   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 22818

modelSvm

##   
## Call:  
## svm(formula = `Sale Price` ~ ., data = dcSalesTrain, kernel = "linear",   
## scale = FALSE)  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.01754386   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 22818

modelSvmPred <- predict(modelSvm, newdata = dcSalesTest)  
modelSvmPredAct <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelSvmPred)) # create modelSvmPredAct dataframe.  
corrAccSvm <- cor(modelSvmPredAct) # 80.8% Accuracy  
corrAccSvm

## actuals preds  
## actuals 1.0000000 0.8078086  
## preds 0.8078086 1.0000000

regr.eval(modelSvmPredAct$actuals, modelSvmPredAct$preds)

## mae mse rmse   
## 79584.7799882 21908711776.3620033 148015.9173074   
## mape   
## 0.1464853

head(modelSvmPredAct)

## actuals preds  
## 7 480000 433218.4  
## 8 370000 434940.0  
## 11 1336000 804697.2  
## 17 760000 698319.1  
## 22 1475000 837349.7  
## 26 1700000 884373.5

plot(modelSvmPredAct,main="Support Vector Machine Pred by Act",col="blue")

