MSDS 660 Week 2 Assignment

Jeremy Beard

2022-07-13

#### Introduction

In this week’s assignment we will be exploring a dataset which came from Zillow, a popular website used for finding housing, apartments, etc. It has data on price, number of rooms, tax information, and the year each house was built. Additionally, it contains some data which is not numeric and for this week’s assignment, we will only pay attention to the numeric data. The purpose of this assignment is to perform a simple linear regression analysis. It will involve finding correlations between variables, removing outliers, removing null values, plotting, and creating models. Let’s begin!

# first we will import packages, read in the data, create a dataframe, and view some summary information  
  
# load the data.table, ggolot2, and dplyr libraries and the zillow\_price.csv file  
library('ggplot2')  
library('Rmisc')

## Loading required package: lattice

## Loading required package: plyr

library('stargazer')

##   
## Please cite as:

## Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

library('dplyr')

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

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

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

library('purrr')

##   
## Attaching package: 'purrr'

## The following object is masked from 'package:plyr':  
##   
## compact

dt <- read.csv("C:\\Users\\jerem\\OneDrive\\Documents\\School\\\_REGIS\\2022-05\_Summer\\MSDS660\\Week2\\zillow\_price.csv")  
  
# Convert the file to a data table  
dt <- as.data.frame(dt)  
  
head(dt)

## parcelid airconditioningtypeid architecturalstyletypeid basementsqft  
## 1 10711738 1 NA NA  
## 2 10711755 1 NA NA  
## 3 10711805 1 NA NA  
## 4 10711816 1 NA NA  
## 5 10711858 1 NA NA  
## 6 10711910 NA NA NA  
## bathroomcnt bedroomcnt buildingclasstypeid buildingqualitytypeid  
## 1 3 4 NA 4  
## 2 3 3 NA 4  
## 3 2 3 NA 4  
## 4 2 4 NA 4  
## 5 2 4 NA 4  
## 6 2 3 NA 4  
## calculatedbathnbr decktypeid finishedfloor1squarefeet  
## 1 3 NA NA  
## 2 3 NA NA  
## 3 2 NA NA  
## 4 2 NA NA  
## 5 2 NA NA  
## 6 2 NA NA  
## calculatedfinishedsquarefeet finishedsquarefeet12 finishedsquarefeet13  
## 1 2538 2538 NA  
## 2 1589 1589 NA  
## 3 2411 2411 NA  
## 4 2232 2232 NA  
## 5 1882 1882 NA  
## 6 1477 1477 NA  
## finishedsquarefeet15 finishedsquarefeet50 finishedsquarefeet6 fips  
## 1 NA NA NA 6037  
## 2 NA NA NA 6037  
## 3 NA NA NA 6037  
## 4 NA NA NA 6037  
## 5 NA NA NA 6037  
## 6 NA NA NA 6037  
## fireplacecnt fullbathcnt garagecarcnt garagetotalsqft hashottuborspa  
## 1 NA 3 NA NA   
## 2 NA 3 NA NA   
## 3 NA 2 NA NA   
## 4 NA 2 NA NA   
## 5 NA 2 NA NA   
## 6 NA 2 NA NA   
## heatingorsystemtypeid latitude longitude lotsizesquarefeet poolcnt  
## 1 2 34220381 -118620802 11012 1  
## 2 2 34222040 -118622240 11010 1  
## 3 2 34220427 -118618549 11723 1  
## 4 2 34222390 -118618631 9002 NA  
## 5 2 34222544 -118617961 9002 1  
## 6 2 34221864 -118615739 11285 1  
## poolsizesum pooltypeid10 pooltypeid2 pooltypeid7 propertycountylandusecode  
## 1 NA NA NA 1 0101  
## 2 NA NA NA 1 0101  
## 3 NA NA NA 1 0101  
## 4 NA NA NA NA 0100  
## 5 NA NA NA 1 0101  
## 6 NA NA NA 1 0101  
## propertylandusetypeid propertyzoningdesc rawcensustractandblock regionidcity  
## 1 261 LARE11 60371132 12447  
## 2 261 LARE11 60371132 12447  
## 3 261 LARE9 60371132 12447  
## 4 261 LARE9 60371132 12447  
## 5 261 LARE9 60371132 12447  
## 6 261 LARE11 60371132 12447  
## regionidcounty regionidneighborhood regionidzip roomcnt storytypeid  
## 1 3101 268588 96339 0 NA  
## 2 3101 268588 96339 0 NA  
## 3 3101 268588 96339 0 NA  
## 4 3101 268588 96339 0 NA  
## 5 3101 268588 96339 0 NA  
## 6 3101 268588 96339 0 NA  
## threequarterbathnbr typeconstructiontypeid unitcnt yardbuildingsqft17  
## 1 NA NA 1 NA  
## 2 NA NA 1 NA  
## 3 NA NA 1 NA  
## 4 NA NA 1 NA  
## 5 NA NA 1 NA  
## 6 NA NA 1 NA  
## yardbuildingsqft26 yearbuilt numberofstories fireplaceflag  
## 1 NA 1978 NA   
## 2 NA 1959 NA   
## 3 NA 1973 NA   
## 4 NA 1973 NA   
## 5 NA 1973 NA   
## 6 NA 1960 NA   
## structuretaxvaluedollarcnt taxvaluedollarcnt assessmentyear  
## 1 245180 567112 2015  
## 2 254691 459844 2015  
## 3 235114 384787 2015  
## 4 262309 437176 2015  
## 5 232037 382055 2015  
## 6 57098 76860 2015  
## landtaxvaluedollarcnt taxdelinquencyflag taxdelinquencyyear  
## 1 321932 NA  
## 2 205153 NA  
## 3 149673 NA  
## 4 174867 NA  
## 5 150018 NA  
## 6 19762 NA  
## censustractandblock price logerror transactiondate  
## 1 6.037113e+13 622343.10 0.0276 2016-08-02  
## 2 6.037113e+13 594921.55 -0.0182 2016-08-02  
## 3 6.037113e+13 420397.41 -0.1009 2016-05-03  
## 4 6.037113e+13 479316.38 -0.0121 2016-04-05  
## 5 6.037113e+13 420538.79 -0.0481 2016-07-15  
## 6 6.037113e+13 96246.55 0.2897 2016-08-30

# how many observations and columns are there?  
# number of observations = number of rows = 90275  
nrow(dt)

## [1] 90275

# number of columns = 60   
ncol(dt)

## [1] 60

# use str and summary to see how many missing values we have,  
# and what the data looks like  
str(dt)

## 'data.frame': 90275 obs. of 60 variables:  
## $ parcelid : int 10711738 10711755 10711805 10711816 10711858 10711910 10712086 10712162 10712163 10712195 ...  
## $ airconditioningtypeid : int 1 1 1 1 1 NA 1 1 1 1 ...  
## $ architecturalstyletypeid : int NA NA NA NA NA NA NA NA NA NA ...  
## $ basementsqft : int NA NA NA NA NA NA NA NA NA NA ...  
## $ bathroomcnt : num 3 3 2 2 2 2 2 3 3 3 ...  
## $ bedroomcnt : int 4 3 3 4 4 3 4 3 4 3 ...  
## $ buildingclasstypeid : int NA NA NA NA NA NA NA NA NA NA ...  
## $ buildingqualitytypeid : int 4 4 4 4 4 4 4 4 4 4 ...  
## $ calculatedbathnbr : num 3 3 2 2 2 2 2 3 3 3 ...  
## $ decktypeid : int NA NA NA NA NA NA NA NA NA NA ...  
## $ finishedfloor1squarefeet : int NA NA NA NA NA NA NA NA NA NA ...  
## $ calculatedfinishedsquarefeet: int 2538 1589 2411 2232 1882 1477 1850 3193 2421 1678 ...  
## $ finishedsquarefeet12 : int 2538 1589 2411 2232 1882 1477 1850 3193 2421 1678 ...  
## $ finishedsquarefeet13 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ finishedsquarefeet15 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ finishedsquarefeet50 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ finishedsquarefeet6 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ fips : int 6037 6037 6037 6037 6037 6037 6037 6037 6037 6037 ...  
## $ fireplacecnt : int NA NA NA NA NA NA NA NA NA NA ...  
## $ fullbathcnt : int 3 3 2 2 2 2 2 3 3 3 ...  
## $ garagecarcnt : int NA NA NA NA NA NA NA NA NA NA ...  
## $ garagetotalsqft : int NA NA NA NA NA NA NA NA NA NA ...  
## $ hashottuborspa : chr "" "" "" "" ...  
## $ heatingorsystemtypeid : int 2 2 2 2 2 2 2 2 2 2 ...  
## $ latitude : int 34220381 34222040 34220427 34222390 34222544 34221864 34226039 34226833 34226843 34223689 ...  
## $ longitude : int -118620802 -118622240 -118618549 -118618631 -118617961 -118615739 -118618527 -118612917 -118612422 -118612746 ...  
## $ lotsizesquarefeet : num 11012 11010 11723 9002 9002 ...  
## $ poolcnt : int 1 1 1 NA 1 1 1 1 1 NA ...  
## $ poolsizesum : int NA NA NA NA NA NA NA NA NA NA ...  
## $ pooltypeid10 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ pooltypeid2 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ pooltypeid7 : int 1 1 1 NA 1 1 1 1 1 NA ...  
## $ propertycountylandusecode : chr "0101" "0101" "0101" "0100" ...  
## $ propertylandusetypeid : int 261 261 261 261 261 261 261 261 261 261 ...  
## $ propertyzoningdesc : chr "LARE11" "LARE11" "LARE9" "LARE9" ...  
## $ rawcensustractandblock : num 60371132 60371132 60371132 60371132 60371132 ...  
## $ regionidcity : int 12447 12447 12447 12447 12447 12447 12447 12447 12447 12447 ...  
## $ regionidcounty : int 3101 3101 3101 3101 3101 3101 3101 3101 3101 3101 ...  
## $ regionidneighborhood : int 268588 268588 268588 268588 268588 268588 268588 268588 268588 268588 ...  
## $ regionidzip : int 96339 96339 96339 96339 96339 96339 96339 96339 96339 96339 ...  
## $ roomcnt : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ storytypeid : int NA NA NA NA NA NA NA NA NA NA ...  
## $ threequarterbathnbr : int NA NA NA NA NA NA NA NA NA NA ...  
## $ typeconstructiontypeid : int NA NA NA NA NA NA NA NA NA NA ...  
## $ unitcnt : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ yardbuildingsqft17 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ yardbuildingsqft26 : int NA NA NA NA NA NA NA NA NA NA ...  
## $ yearbuilt : int 1978 1959 1973 1973 1973 1960 1974 1964 1962 1961 ...  
## $ numberofstories : int NA NA NA NA NA NA NA NA NA NA ...  
## $ fireplaceflag : chr "" "" "" "" ...  
## $ structuretaxvaluedollarcnt : num 245180 254691 235114 262309 232037 ...  
## $ taxvaluedollarcnt : num 567112 459844 384787 437176 382055 ...  
## $ assessmentyear : int 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...  
## $ landtaxvaluedollarcnt : num 321932 205153 149673 174867 150018 ...  
## $ taxdelinquencyflag : chr "" "" "" "" ...  
## $ taxdelinquencyyear : int NA NA NA NA NA NA NA NA NA NA ...  
## $ censustractandblock : num 6.04e+13 6.04e+13 6.04e+13 6.04e+13 6.04e+13 ...  
## $ price : num 622343 594922 420397 479316 420539 ...  
## $ logerror : num 0.0276 -0.0182 -0.1009 -0.0121 -0.0481 ...  
## $ transactiondate : chr "2016-08-02" "2016-08-02" "2016-05-03" "2016-04-05" ...

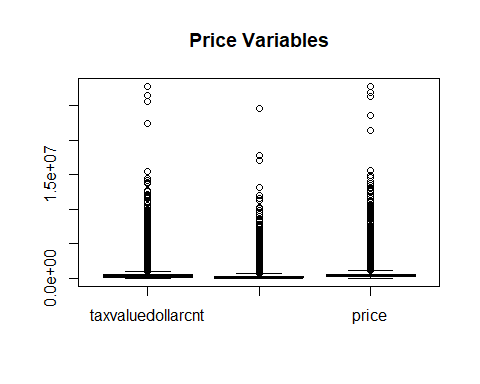
summary(dt)

## parcelid airconditioningtypeid architecturalstyletypeid  
## Min. : 10711738 Min. : 1.00 Min. : 2.00   
## 1st Qu.: 11559500 1st Qu.: 1.00 1st Qu.: 7.00   
## Median : 12547337 Median : 1.00 Median : 7.00   
## Mean : 12984656 Mean : 1.82 Mean : 7.23   
## 3rd Qu.: 14227552 3rd Qu.: 1.00 3rd Qu.: 7.00   
## Max. :162960842 Max. :13.00 Max. :21.00   
## NA's :61494 NA's :90014   
## basementsqft bathroomcnt bedroomcnt buildingclasstypeid  
## Min. : 100.0 Min. : 0.000 Min. : 0.000 Min. :4   
## 1st Qu.: 407.5 1st Qu.: 2.000 1st Qu.: 2.000 1st Qu.:4   
## Median : 616.0 Median : 2.000 Median : 3.000 Median :4   
## Mean : 713.6 Mean : 2.279 Mean : 3.032 Mean :4   
## 3rd Qu.: 872.0 3rd Qu.: 3.000 3rd Qu.: 4.000 3rd Qu.:4   
## Max. :1555.0 Max. :20.000 Max. :16.000 Max. :4   
## NA's :90232 NA's :90259   
## buildingqualitytypeid calculatedbathnbr decktypeid   
## Min. : 1.00 Min. : 1.000 Min. :66   
## 1st Qu.: 4.00 1st Qu.: 2.000 1st Qu.:66   
## Median : 7.00 Median : 2.000 Median :66   
## Mean : 5.57 Mean : 2.309 Mean :66   
## 3rd Qu.: 7.00 3rd Qu.: 3.000 3rd Qu.:66   
## Max. :12.00 Max. :20.000 Max. :66   
## NA's :32911 NA's :1182 NA's :89617   
## finishedfloor1squarefeet calculatedfinishedsquarefeet finishedsquarefeet12  
## Min. : 44 Min. : 2 Min. : 2   
## 1st Qu.: 938 1st Qu.: 1184 1st Qu.: 1172   
## Median :1244 Median : 1540 Median : 1518   
## Mean :1348 Mean : 1773 Mean : 1745   
## 3rd Qu.:1614 3rd Qu.: 2095 3rd Qu.: 2056   
## Max. :7625 Max. :22741 Max. :20013   
## NA's :83419 NA's :661 NA's :4679   
## finishedsquarefeet13 finishedsquarefeet15 finishedsquarefeet50  
## Min. :1056 Min. : 560 Min. : 44   
## 1st Qu.:1392 1st Qu.: 1648 1st Qu.: 938   
## Median :1440 Median : 2104 Median :1248   
## Mean :1405 Mean : 2380 Mean :1356   
## 3rd Qu.:1440 3rd Qu.: 2862 3rd Qu.:1619   
## Max. :1584 Max. :22741 Max. :8352   
## NA's :90242 NA's :86711 NA's :83419   
## finishedsquarefeet6 fips fireplacecnt fullbathcnt   
## Min. : 257 Min. :6037 Min. :1.00 Min. : 1.000   
## 1st Qu.:1112 1st Qu.:6037 1st Qu.:1.00 1st Qu.: 2.000   
## Median :2028 Median :6037 Median :1.00 Median : 2.000   
## Mean :2303 Mean :6049 Mean :1.19 Mean : 2.241   
## 3rd Qu.:3431 3rd Qu.:6059 3rd Qu.:1.00 3rd Qu.: 3.000   
## Max. :7224 Max. :6111 Max. :5.00 Max. :20.000   
## NA's :89854 NA's :80668 NA's :1182   
## garagecarcnt garagetotalsqft hashottuborspa heatingorsystemtypeid  
## Min. : 0.00 Min. : 0.0 Length:90275 Min. : 1.00   
## 1st Qu.: 2.00 1st Qu.: 0.0 Class :character 1st Qu.: 2.00   
## Median : 2.00 Median : 433.0 Mode :character Median : 2.00   
## Mean : 1.81 Mean : 345.5 Mean : 3.93   
## 3rd Qu.: 2.00 3rd Qu.: 484.0 3rd Qu.: 7.00   
## Max. :24.00 Max. :7339.0 Max. :24.00   
## NA's :60338 NA's :60338 NA's :34195   
## latitude longitude lotsizesquarefeet poolcnt   
## Min. :33339295 Min. :-119447865 Min. : 167 Min. :1   
## 1st Qu.:33811538 1st Qu.:-118411692 1st Qu.: 5703 1st Qu.:1   
## Median :34021500 Median :-118173431 Median : 7200 Median :1   
## Mean :34005411 Mean :-118198868 Mean : 29110 Mean :1   
## 3rd Qu.:34172742 3rd Qu.:-117921588 3rd Qu.: 11686 3rd Qu.:1   
## Max. :34816009 Max. :-117554924 Max. :6971010 Max. :1   
## NA's :10150 NA's :72374   
## poolsizesum pooltypeid10 pooltypeid2 pooltypeid7   
## Min. : 28.0 Min. :1 Min. :1 Min. :1   
## 1st Qu.: 420.0 1st Qu.:1 1st Qu.:1 1st Qu.:1   
## Median : 500.0 Median :1 Median :1 Median :1   
## Mean : 519.8 Mean :1 Mean :1 Mean :1   
## 3rd Qu.: 600.0 3rd Qu.:1 3rd Qu.:1 3rd Qu.:1   
## Max. :1750.0 Max. :1 Max. :1 Max. :1   
## NA's :89306 NA's :89114 NA's :89071 NA's :73578   
## propertycountylandusecode propertylandusetypeid propertyzoningdesc  
## Length:90275 Min. : 31.0 Length:90275   
## Class :character 1st Qu.:261.0 Class :character   
## Mode :character Median :261.0 Mode :character   
## Mean :261.8   
## 3rd Qu.:266.0   
## Max. :275.0   
##   
## rawcensustractandblock regionidcity regionidcounty regionidneighborhood  
## Min. :60371011 Min. : 3491 Min. :1286 Min. : 6952   
## 1st Qu.:60373203 1st Qu.: 12447 1st Qu.:1286 1st Qu.: 46736   
## Median :60376200 Median : 25218 Median :3101 Median :118887   
## Mean :60491795 Mean : 33761 Mean :2525 Mean :190647   
## 3rd Qu.:60590423 3rd Qu.: 45457 3rd Qu.:3101 3rd Qu.:274800   
## Max. :61110091 Max. :396556 Max. :3101 Max. :764167   
## NA's :1803 NA's :54263   
## regionidzip roomcnt storytypeid threequarterbathnbr  
## Min. : 95982 Min. : 0.000 Min. :7 Min. :1.00   
## 1st Qu.: 96193 1st Qu.: 0.000 1st Qu.:7 1st Qu.:1.00   
## Median : 96393 Median : 0.000 Median :7 Median :1.00   
## Mean : 96586 Mean : 1.479 Mean :7 Mean :1.01   
## 3rd Qu.: 96987 3rd Qu.: 0.000 3rd Qu.:7 3rd Qu.:1.00   
## Max. :399675 Max. :18.000 Max. :7 Max. :4.00   
## NA's :35 NA's :90232 NA's :78266   
## typeconstructiontypeid unitcnt yardbuildingsqft17 yardbuildingsqft26  
## Min. : 4.00 Min. : 1.00 Min. : 25.0 Min. : 18.0   
## 1st Qu.: 6.00 1st Qu.: 1.00 1st Qu.: 180.0 1st Qu.: 100.0   
## Median : 6.00 Median : 1.00 Median : 259.5 Median : 159.0   
## Mean : 6.01 Mean : 1.11 Mean : 310.1 Mean : 311.7   
## 3rd Qu.: 6.00 3rd Qu.: 1.00 3rd Qu.: 384.0 3rd Qu.: 361.0   
## Max. :13.00 Max. :143.00 Max. :2678.0 Max. :1366.0   
## NA's :89976 NA's :31922 NA's :87629 NA's :90180   
## yearbuilt numberofstories fireplaceflag structuretaxvaluedollarcnt  
## Min. :1885 Min. :1.00 Length:90275 Min. : 100   
## 1st Qu.:1953 1st Qu.:1.00 Class :character 1st Qu.: 81245   
## Median :1970 Median :1.00 Mode :character Median : 132000   
## Mean :1969 Mean :1.44 Mean : 180093   
## 3rd Qu.:1987 3rd Qu.:2.00 3rd Qu.: 210534   
## Max. :2015 Max. :4.00 Max. :9948100   
## NA's :756 NA's :69705 NA's :380   
## taxvaluedollarcnt assessmentyear landtaxvaluedollarcnt taxdelinquencyflag  
## Min. : 22 Min. :2015 Min. : 22 Length:90275   
## 1st Qu.: 199023 1st Qu.:2015 1st Qu.: 82228 Class :character   
## Median : 342872 Median :2015 Median : 192970 Mode :character   
## Mean : 457673 Mean :2015 Mean : 278335   
## 3rd Qu.: 540589 3rd Qu.:2015 3rd Qu.: 345420   
## Max. :27750000 Max. :2015 Max. :24500000   
## NA's :1 NA's :1   
## taxdelinquencyyear censustractandblock price logerror   
## Min. : 6.0 Min. :6.037e+13 Min. : 4231 Min. :-4.60500   
## 1st Qu.:13.0 1st Qu.:6.037e+13 1st Qu.: 247658 1st Qu.:-0.02530   
## Median :14.0 Median :6.038e+13 Median : 391616 Median : 0.00600   
## Mean :13.4 Mean :6.049e+13 Mean : 515860 Mean : 0.01146   
## 3rd Qu.:15.0 3rd Qu.:6.059e+13 3rd Qu.: 594922 3rd Qu.: 0.03920   
## Max. :99.0 Max. :6.111e+13 Max. :27753111 Max. : 4.73700   
## NA's :88492 NA's :605 NA's :6   
## transactiondate   
## Length:90275   
## Class :character   
## Mode :character   
##   
##   
##   
##

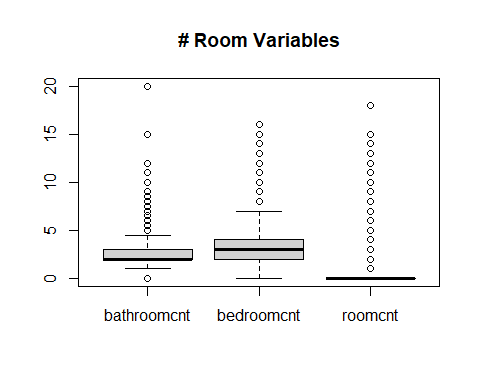
#### Methods

So, as we can see from the information above, there are many non-numeric columns in the dataset and there are many null/na values. For this week’s assignment, it will make sense to remove these columns and these na values. For the simple linear regression, we will be creating boxplots of the data, finding which variables are most correlated to price, creating a model and scatterplot of this variable+price, and plotting the model. From this, we hope to gain an understanding of the highest correlation to price in the Zillow dataset, and also if removing outliers changes our conclusion at all.

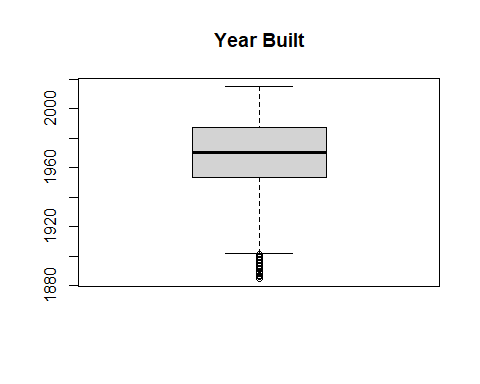
# first I'm going to just create some boxplots to satisfy my curiosity  
small\_nums <- c('bathroomcnt',  
 'bedroomcnt',  
 'roomcnt')  
  
big\_nums <- c('taxvaluedollarcnt',  
 'landtaxvaluedollarcnt',  
 'price')  
dt\_smallnums <- dt[ , small\_nums]  
dt\_bignums <- dt[ , big\_nums]  
boxplot(dt\_bignums)  
title("Price Variables")



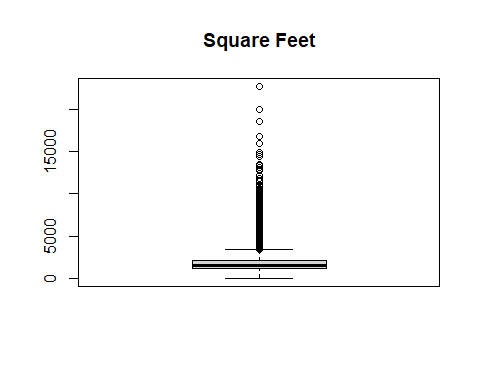
boxplot(dt\_smallnums)  
title("# Room Variables")



boxplot(dt$yearbuilt)  
title("Year Built")



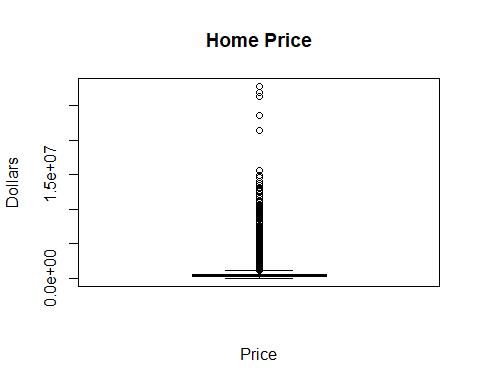
boxplot(dt$calculatedfinishedsquarefeet)  
title("Square Feet")



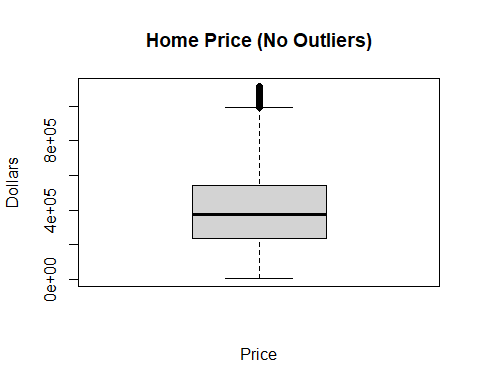
# columns that are numeric and don't have lots of missing values  
# you can add others if you like  
numeric\_cols <- c('bathroomcnt',  
 'bedroomcnt',  
 'calculatedfinishedsquarefeet',  
 'roomcnt',  
 'yearbuilt',  
 'taxvaluedollarcnt',  
 'landtaxvaluedollarcnt',  
 'price')  
  
# Simplify your dataset by only selecting the columns of your choosing dt[, numeric\_cols, with = FALSE]  
dt\_num <- dt[ , numeric\_cols]  
  
summary(dt\_num)

## bathroomcnt bedroomcnt calculatedfinishedsquarefeet  
## Min. : 0.000 Min. : 0.000 Min. : 2   
## 1st Qu.: 2.000 1st Qu.: 2.000 1st Qu.: 1184   
## Median : 2.000 Median : 3.000 Median : 1540   
## Mean : 2.279 Mean : 3.032 Mean : 1773   
## 3rd Qu.: 3.000 3rd Qu.: 4.000 3rd Qu.: 2095   
## Max. :20.000 Max. :16.000 Max. :22741   
## NA's :661   
## roomcnt yearbuilt taxvaluedollarcnt landtaxvaluedollarcnt  
## Min. : 0.000 Min. :1885 Min. : 22 Min. : 22   
## 1st Qu.: 0.000 1st Qu.:1953 1st Qu.: 199023 1st Qu.: 82228   
## Median : 0.000 Median :1970 Median : 342872 Median : 192970   
## Mean : 1.479 Mean :1969 Mean : 457673 Mean : 278335   
## 3rd Qu.: 0.000 3rd Qu.:1987 3rd Qu.: 540589 3rd Qu.: 345420   
## Max. :18.000 Max. :2015 Max. :27750000 Max. :24500000   
## NA's :756 NA's :1 NA's :1   
## price   
## Min. : 4231   
## 1st Qu.: 247658   
## Median : 391616   
## Mean : 515860   
## 3rd Qu.: 594922   
## Max. :27753111   
## NA's :6

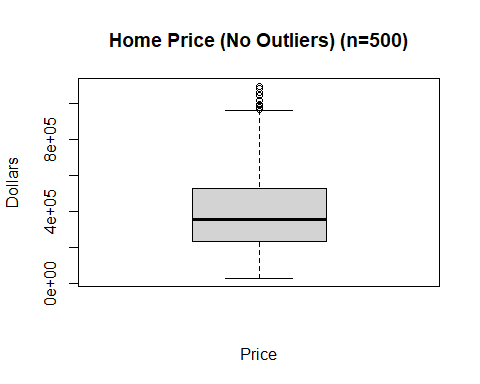
# We want to try to correlate home price with another variable.  
# Let's look to see if there are any outliers in the price column we need to remove  
# Create a boxplot of the price data  
price\_data <- c('price')  
dt\_price <- dt[ , price\_data]  
boxplot(dt\_price, xlab="Price", ylab="Dollars")  
title("Home Price")



# Wow there are expensive homes!  
  
# Remove the outliers. dt[!which(dt$price %in% boxplot(dt$price)$out)]  
dt\_price\_wo\_outliers = dt\_price[!dt\_price %in% boxplot.stats(dt\_price)$out]  
  
# How many outliers did we drop? And lets plot a new box plot to see the column  
# 90275 - 84180 = 6095 entries were dropped!!! that's a lot  
boxplot(dt\_price\_wo\_outliers, xlab="Price", ylab="Dollars")  
title("Home Price (No Outliers)")



# In our case, we have too many observations.   
# Use sample() to only sample a few hundred (maybe 500) points to plot.  
# plot a few of the more interesting pairs together  
sample\_dt\_price\_wo\_outliers = sample(dt\_price\_wo\_outliers, 500)  
boxplot(sample\_dt\_price\_wo\_outliers, xlab="Price", ylab="Dollars")  
title("Home Price (No Outliers) (n=500)")



# create a new data.table by dropping any missing values  
# look up 'complete.cases()'  
# use dim() to see how many cases we dropped  
dt\_no\_na <- dt\_num[complete.cases(dt\_num),]  
#dt\_no\_na # this created waaaaaay too much output!  
dim(dt\_num)

## [1] 90275 8

dim(dt\_no\_na)

## [1] 89499 8

# get the pearson correlation between price and another variable using cor()  
#...there are other types of correlations  
# try ?cor to see options, and try another correlation   
#?cor  
cor(dt\_no\_na$yearbuilt, dt\_no\_na$price, method="pearson")

## [1] 0.1156169

cor.test(dt\_no\_na$yearbuilt, dt\_no\_na$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: dt\_no\_na$yearbuilt and dt\_no\_na$price  
## t = 34.822, df = 89497, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.1091480 0.1220759  
## sample estimates:  
## cor   
## 0.1156169

cor.test(dt\_no\_na$bathroomcnt, dt\_no\_na$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: dt\_no\_na$bathroomcnt and dt\_no\_na$price  
## t = 163.24, df = 89497, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.4739231 0.4840200  
## sample estimates:  
## cor   
## 0.4789874

cor.test(dt\_no\_na$bedroomcnt, dt\_no\_na$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: dt\_no\_na$bedroomcnt and dt\_no\_na$price  
## t = 77.857, df = 89497, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.2457155 0.2579874  
## sample estimates:  
## cor   
## 0.2518616

cor.test(dt\_no\_na$calculatedfinishedsquarefeet, dt\_no\_na$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: dt\_no\_na$calculatedfinishedsquarefeet and dt\_no\_na$price  
## t = 218.27, df = 89497, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.5851119 0.5936631  
## sample estimates:  
## cor   
## 0.589404

cor.test(dt\_no\_na$roomcnt, dt\_no\_na$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: dt\_no\_na$roomcnt and dt\_no\_na$price  
## t = -9.4569, df = 89497, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.03813935 -0.02504945  
## sample estimates:  
## cor   
## -0.03159575

cor.test(dt\_no\_na$taxvaluedollarcnt, dt\_no\_na$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: dt\_no\_na$taxvaluedollarcnt and dt\_no\_na$price  
## t = 929.05, df = 89497, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.951249 0.952480  
## sample estimates:  
## cor   
## 0.9518683

cor.test(dt\_no\_na$landtaxvaluedollarcnt, dt\_no\_na$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: dt\_no\_na$landtaxvaluedollarcnt and dt\_no\_na$price  
## t = 639.43, df = 89497, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9045873 0.9069404  
## sample estimates:  
## cor   
## 0.9057708

#from these, it appears that taxvaluedollarcnt is the most correlated with price  
  
  
# use the lm() command to fit a linear model of price to the   
# one variable you think is most correlated or predictive of price  
# lm stands for 'linear model'  
m1 <- lm(price ~ taxvaluedollarcnt, data = dt\_no\_na)

So the code above shows the correlation analysis and model creation of a simple linear regression. BUT this is with outliers still in the dataset. Now we will remove the outliers and perform the same analysis and then compare the models

# The method commented out below didn't work, but I wanted to keep it here for posterity  
#findOutliers <- function(dataframe){  
# dataframe %>%  
# select\_if(is.numeric) %>%   
# map(~ boxplot.stats(.x)$out)  
#}  
  
#outliers <- findOutliers(dt\_num)  
#temp <- list()  
#for (col in names(outliers)) {  
# outlier <- outliers[[col]]  
# if (length(outlier) > 0) {  
# temp[col] <- dt\_num[-which(dt\_num[[col]] %in% outlier),][col]  
# } else {  
# temp[col] <- dt\_num[col]  
# }  
#}  
  
#boxplot(temp)  
#removing the outliers makes all the row numbers different, hmm  
  
#let's try something different  
#find Q1, Q3, and interquartile range for values in column A  
Q1 <- quantile(dt\_no\_na$price, .25)  
Q3 <- quantile(dt\_no\_na$price, .75)  
IQR <- IQR(dt\_no\_na$price)  
  
#only keep rows in dataframe that have values within 1.5\*IQR of Q1 and Q3  
no\_outliers <- subset(dt\_no\_na, dt\_no\_na$price> (Q1 - 1.5\*IQR) & dt\_no\_na$price< (Q3 + 1.5\*IQR))  
  
#view row and column count of new data frame before and after  
dim(dt\_no\_na)

## [1] 89499 8

dim(no\_outliers)

## [1] 83479 8

cor.test(no\_outliers$yearbuilt, no\_outliers$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: no\_outliers$yearbuilt and no\_outliers$price  
## t = 57.754, df = 83477, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.1894829 0.2025288  
## sample estimates:  
## cor   
## 0.1960145

cor.test(no\_outliers$bathroomcnt, no\_outliers$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: no\_outliers$bathroomcnt and no\_outliers$price  
## t = 121.26, df = 83477, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.3812035 0.3927389  
## sample estimates:  
## cor   
## 0.3869863

cor.test(no\_outliers$bedroomcnt, no\_outliers$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: no\_outliers$bedroomcnt and no\_outliers$price  
## t = 70.642, df = 83477, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.231092 0.243894  
## sample estimates:  
## cor   
## 0.2375033

cor.test(no\_outliers$calculatedfinishedsquarefeet, no\_outliers$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: no\_outliers$calculatedfinishedsquarefeet and no\_outliers$price  
## t = 154.9, df = 83477, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.4672179 0.4777562  
## sample estimates:  
## cor   
## 0.472504

cor.test(no\_outliers$roomcnt, no\_outliers$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: no\_outliers$roomcnt and no\_outliers$price  
## t = -1.0573, df = 83477, p-value = 0.2904  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.01044278 0.00312425  
## sample estimates:  
## cor   
## -0.003659433

cor.test(no\_outliers$taxvaluedollarcnt, no\_outliers$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: no\_outliers$taxvaluedollarcnt and no\_outliers$price  
## t = 637.69, df = 83477, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9097067 0.9120175  
## sample estimates:  
## cor   
## 0.9108692

cor.test(no\_outliers$landtaxvaluedollarcnt, no\_outliers$price, method="pearson")

##   
## Pearson's product-moment correlation  
##   
## data: no\_outliers$landtaxvaluedollarcnt and no\_outliers$price  
## t = 414.23, df = 83477, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.8179640 0.8224044  
## sample estimates:  
## cor   
## 0.8201966

#from these, it appears that taxvaluedollarcnt is STILL the most correlated with price  
m2 <- lm(price ~ taxvaluedollarcnt, data = no\_outliers)

Now after creating the models both with and without outliers included, we will view summaries and plots of each and then compare the results.

#### Results

As we can see from the results below, the analysis using outliers was slightly different than the analysis with outliers removed. Looking at just the scatterplots with the regression line overlaid, the scatterplot with outliers removed looks much more widespread and scattered at first glance, but this is only because it is more-or-less a “zoomed-in” view of the scatterplot with outliers included. Removing outliers seemed to enhance the scatterplot a bit.

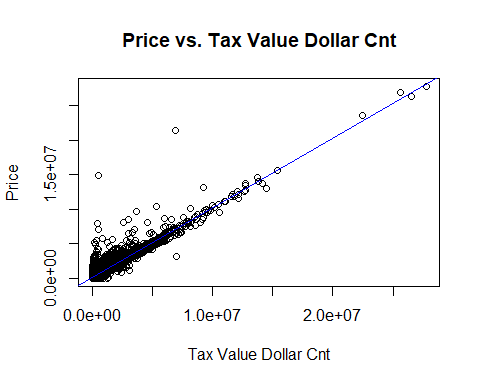
With regard to the model created for each of the two cases, we can see that the error has been reduced overall. The max residual was reduced from 14,391,401 to 971,142. The residual standard error was reduced from 179800 to 94970.

However, the R-squared value tells us that the model was actually not as well fit to the data without outliers, as it was with the outliers included. The R-squared value with outliers was 0.9061 while the R-squared value without outliers was 0.8297. This was also found when viewing the Pearson correlation with and without outliers, the same effect was found.

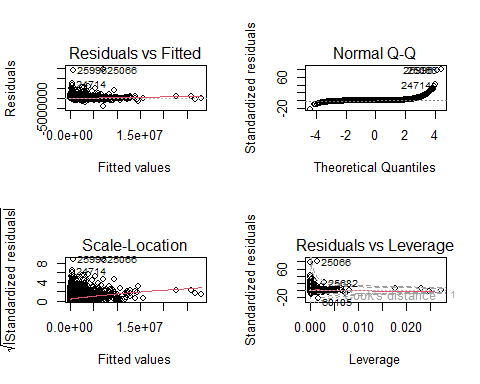
# view the model summary   
summary(m1)

##   
## Call:  
## lm(formula = price ~ taxvaluedollarcnt, data = dt\_no\_na)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3932089 -47320 -28139 2677 14391401   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.325e+04 7.800e+02 68.28 <2e-16 \*\*\*  
## taxvaluedollarcnt 1.009e+00 1.086e-03 929.05 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 179800 on 89497 degrees of freedom  
## Multiple R-squared: 0.9061, Adjusted R-squared: 0.9061   
## F-statistic: 8.631e+05 on 1 and 89497 DF, p-value: < 2.2e-16

# plot a scatter plot of the price and the variable you chose  
plot(dt\_no\_na$taxvaluedollarcnt, dt\_no\_na$price, main = "Price vs. Tax Value Dollar Cnt", xlab = "Tax Value Dollar Cnt", ylab="Price")  
  
# add the regression line to the current plot using abline()  
abline(m1, col = "blue")



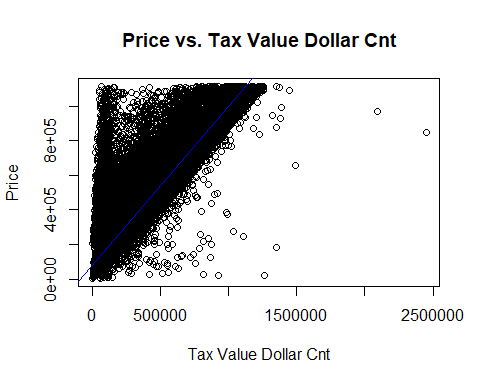
# R makes it very easy to plot the diagnostics of a fit  
# here's a decent resources explaining the plots:   
# http://data.library.virginia.edu/diagnostic-plots/  
# plot the fit diagnostics here  
par(mfrow=c(2,2)) # Change the panel layout to 2 x 2  
plot(m1)



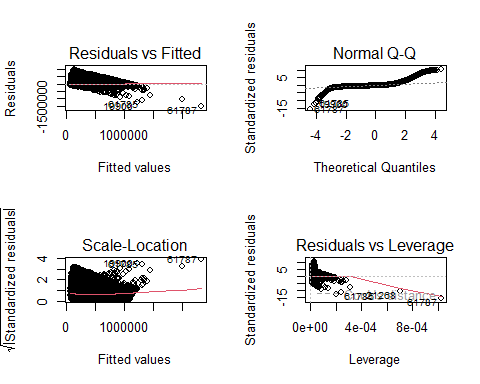
par(mfrow=c(1,1)) # Change back to 1 x 1  
  
# view the model summary   
summary(m2)

##   
## Call:  
## lm(formula = price ~ taxvaluedollarcnt, data = no\_outliers)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1487009 -44399 -22876 13467 971142   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.481e+04 6.188e+02 120.9 <2e-16 \*\*\*  
## taxvaluedollarcnt 9.218e-01 1.446e-03 637.7 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 94970 on 83477 degrees of freedom  
## Multiple R-squared: 0.8297, Adjusted R-squared: 0.8297   
## F-statistic: 4.066e+05 on 1 and 83477 DF, p-value: < 2.2e-16

# plot a scatter plot of the price and the variable you chose  
plot(no\_outliers$taxvaluedollarcnt, no\_outliers$price, main = "Price vs. Tax Value Dollar Cnt", xlab = "Tax Value Dollar Cnt", ylab="Price")  
  
# add the regression line to the current plot using abline()  
abline(m2, col = "blue")



# R makes it very easy to plot the diagnostics of a fit  
# here's a decent resources explaining the plots:   
# http://data.library.virginia.edu/diagnostic-plots/  
# plot the fit diagnostics here  
par(mfrow=c(2,2)) # Change the panel layout to 2 x 2  
plot(m2)



par(mfrow=c(1,1)) # Change back to 1 x 1

#### Conclusion

In conclusion, we found that when outliers were removed from the Price data from the Zillow dataset, the error was reduced overall but the simple linear regression model was also less correlated with the data, less accurately fit. This may be because the outliers helped to provide an indication of the general trend of the data and without them, the data was more scattered and thus it was more difficult to fit a model to the data. Overall, it was found that the Tax Value Dollar Count was the most correlated to home price, in both cases of outliers-included or outliers-removed.

In the future, we can improve this by finding other correlations between other variables in this dataset, or possibly by imputing null values instead of merely removing them. We can also include more columns or features from this dataset in our analysis and convert categorical data to numeric to give us more information to work with.

Thank you! Jeremy Beard