zillow by Xiaodan

Xiaodan Chen

2017年12月28日

1.Define Problem

Given new data to predict its logerror, which is the log of estimated house price minus the log of real sales price.

2.Clean Data

```
#load data
setwd("F:/tiger/zillow")
train<-read.csv('train_property.csv',stringsAsFactors = F)

#remove variables with missing values more than 20%
NA_rate<-colMeans(sapply(train,is.na))
remain_col<-names(train)[which(NA_rate<.2)]
train2<-train[,remain_col]
train2<-train2[,-1]
#remained variables
names(train2)
```

```
##  [1] "parcelid"                     "logerror"
##  [3] "transactiondate"              "bathroomcnt"
##  [5] "bedroomcnt"                   "calculatedbathnbr"
##  [7] "calculatedfinishedsquarefeet" "finishedsquarefeet12"
##  [9] "fips"                         "fullbathcnt"
## [11] "hashottuborspa"               "latitude"
## [13] "longitude"                    "lotsizesquarefeet"
## [15] "propertycountylandusecode"    "propertylandusetypeid"
## [17] "propertyzoningdesc"           "rawcensustractandblock"
## [19] "regionidcity"                 "regionidcounty"
## [21] "regionidzip"                  "roomcnt"
## [23] "yearbuilt"                    "fireplaceflag"
## [25] "structuretaxvaluedollarcnt"   "taxvaluedollarcnt"
## [27] "assessmentyear"               "landtaxvaluedollarcnt"
## [29] "taxamount"                    "taxdelinquencyflag"
## [31] "censustractandblock"
```

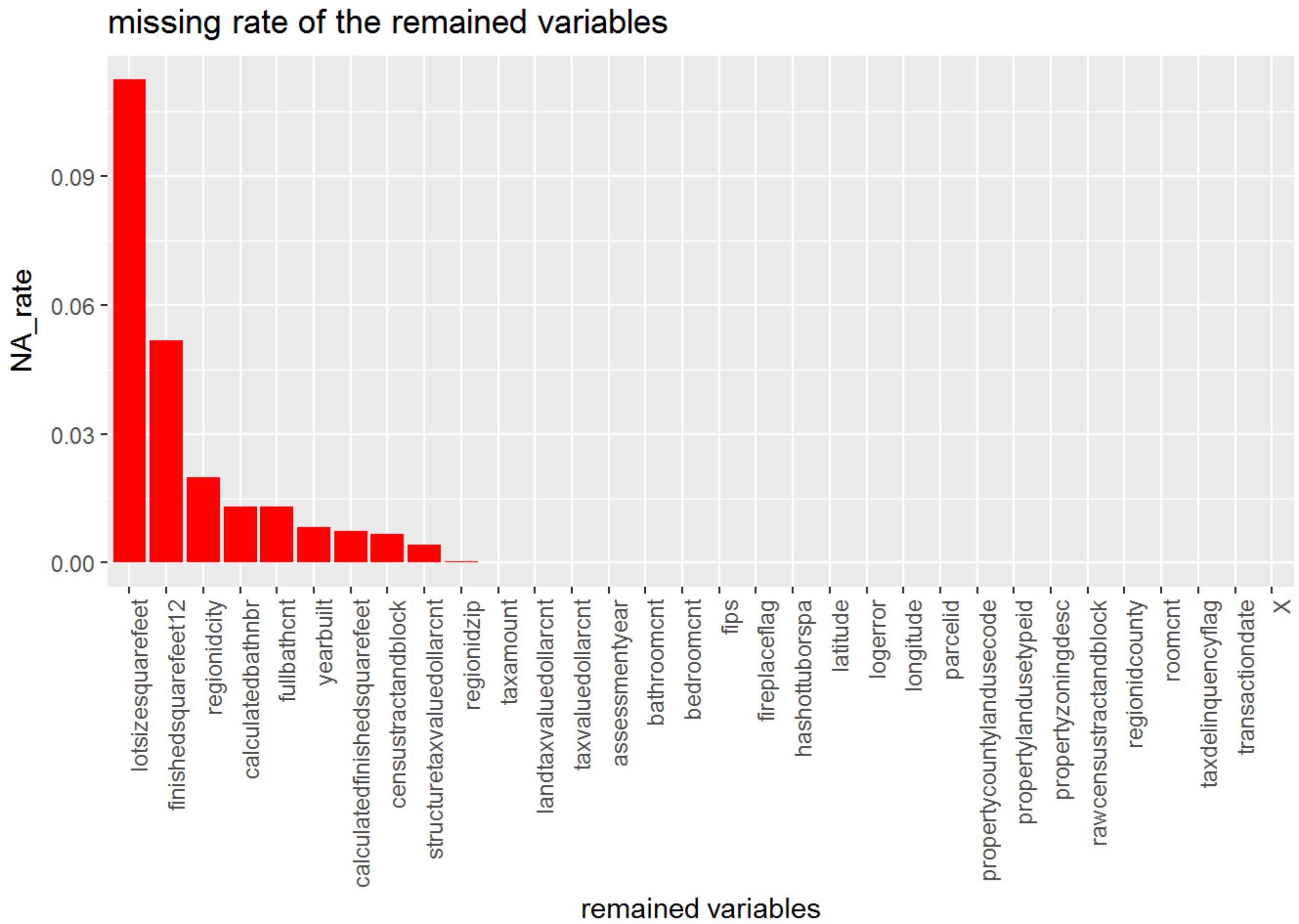
```
head(train2)
```

```
##   parcelid logerror transactiondate bathroomcnt bedroomcnt
## 1 10711738   0.0276      2016-08-02           3          4
## 2 10711755  -0.0182      2016-08-02           3          3
## 3 10711805  -0.1009      2016-05-03           2          3
## 4 10711816  -0.0121      2016-04-05           2          4
## 5 10711858  -0.0481      2016-07-15           2          4
## 6 10711910   0.2897      2016-08-30           2          3
##   calculatedbathnbr calculatedfinishedsquarefeet finishedsquarefeet12 fips
## 1                 3                         2538                 2538 6037
## 2                 3                         1589                 1589 6037
## 3                 2                         2411                 2411 6037
## 4                 2                         2232                 2232 6037
## 5                 2                         1882                 1882 6037
## 6                 2                         1477                 1477 6037
##   fullbathcnt hashottuborspa latitude  longitude lotsizesquarefeet
## 1           3                34220381 -118620802             11012
## 2           3                34222040 -118622240             11010
## 3           2                34220427 -118618549             11723
## 4           2                34222390 -118618631              9002
## 5           2                34222544 -118617961              9002
## 6           2                34221864 -118615739             11285
##   propertycountylandusecode propertylandusetypeid propertyzoningdesc
## 1                      0101                   261             LARE11
## 2                      0101                   261             LARE11
## 3                      0101                   261              LARE9
## 4                      0100                   261              LARE9
## 5                      0101                   261              LARE9
## 6                      0101                   261             LARE11
##   rawcensustractandblock regionidcity regionidcounty regionidzip roomcnt
## 1               60371132        12447           3101       96339       0
## 2               60371132        12447           3101       96339       0
## 3               60371132        12447           3101       96339       0
## 4               60371132        12447           3101       96339       0
## 5               60371132        12447           3101       96339       0
## 6               60371132        12447           3101       96339       0
##   yearbuilt fireplaceflag structuretaxvaluedollarcnt taxvaluedollarcnt
## 1      1978                                   245180            567112
## 2      1959                                   254691            459844
## 3      1973                                   235114            384787
## 4      1973                                   262309            437176
## 5      1973                                   232037            382055
## 6      1960                                    57098             76860
##   assessmentyear landtaxvaluedollarcnt taxamount taxdelinquencyflag
## 1           2015                321932   7219.18
## 2           2015                205153   6901.09
## 3           2015                149673   4876.61
## 4           2015                174867   5560.07
## 5           2015                150018   4878.25
## 6           2015                 19762   1116.46
##   censustractandblock
## 1        6.037113e+13
## 2        6.037113e+13
## 3        6.037113e+13
## 4        6.037113e+13
## 5        6.037113e+13
## 6        6.037113e+13
```

```
#missing rate of the remained variables from the highest to the lowest
miss<-data.frame(var=remain_col,NA_rate=NA_rate[remain_col],row.names = NULL,stringsAsFactors =
 F)
miss<-miss[order(miss$NA_rate,decreasing = T),]
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
ggplot(miss,aes(x=reorder(var,-NA_rate),y=NA_rate))+geom_bar(stat='identity',fill='red')+
  labs(x='remained variables',title='missing rate of the remained variables')+
  theme(axis.text.x = element_text(angle=90,hjust=1))
```




```
str(train2)
```

```
## 'data.frame':    90275 obs. of  31 variables:
##  $ parcelid                    : int  10711738 10711755 10711805 10711816 10711858 10711910
 10712086 10712162 10712163 10712195 ...
##  $ 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"
 ...
##  $ bathroomcnt                 : num  3 3 2 2 2 2 2 3 3 3 ...
##  $ bedroomcnt                  : int  4 3 3 4 4 3 4 3 4 3 ...
##  $ calculatedbathnbr           : num  3 3 2 2 2 2 2 3 3 3 ...
##  $ calculatedfinishedsquarefeet: int  2538 1589 2411 2232 1882 1477 1850 3193 2421 1678 ...
##  $ finishedsquarefeet12        : int  2538 1589 2411 2232 1882 1477 1850 3193 2421 1678 ...
##  $ fips                        : int  6037 6037 6037 6037 6037 6037 6037 6037 6037 6037 ...
##  $ fullbathcnt                 : int  3 3 2 2 2 2 2 3 3 3 ...
##  $ hashottuborspa              : chr  "" "" "" "" ...
##  $ 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 ...
##  $ 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 ...
##  $ regionidzip                 : int  96339 96339 96339 96339 96339 96339 96339 96339 96339
 96339 ...
##  $ roomcnt                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ yearbuilt                   : int  1978 1959 1973 1973 1973 1960 1974 1964 1962 1961 ...
##  $ 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 ...
##  $ taxamount                   : num  7219 6901 4877 5560 4878 ...
##  $ taxdelinquencyflag          : chr  "" "" "" "" ...
##  $ censustractandblock         : num  6.04e+13 6.04e+13 6.04e+13 6.04e+13 6.04e+13 ...
```

3.Explore data

The original train dataset has 90275 observations and 61 variables. After removing data with missing rate more than 20%, 31 variables are remained.

Among these 31 variables, 18 are continuous variables. Except the outcome variablE logerror, the other 17 variables can be grouped into 5 categories.

1. room count related variables: bathroomcnt, bedroomcnt, caculatedbathnbr, fullbathcnt, roomcnt
2. house size related variables: calculatedfinishedsquarefeet, finishedsquarefeet12, lotsizesquarefeet
3. house location related: longitute, latitute

4)house value related: taxvaluedollarcnt, landtaxvaluedollarcnt, taxamount,structuretaxvaluedollarcnt

5. date related: yearbuilt, transactiondate , assessmentyear

There are 13 categorical variables: 1) parcelid

2. house adress related: fips, regionidcity, regionidcounty, regionidzip
3. house feature related: hashottuborspa, fireplaceflag
4. property use variable: propertycountylandusecode, propertyzoningdesc, propertylandusetypeid
5. tax : taxdelinquencyflag
6. census track and block variables: censustractandblock and rawcensustractandblock

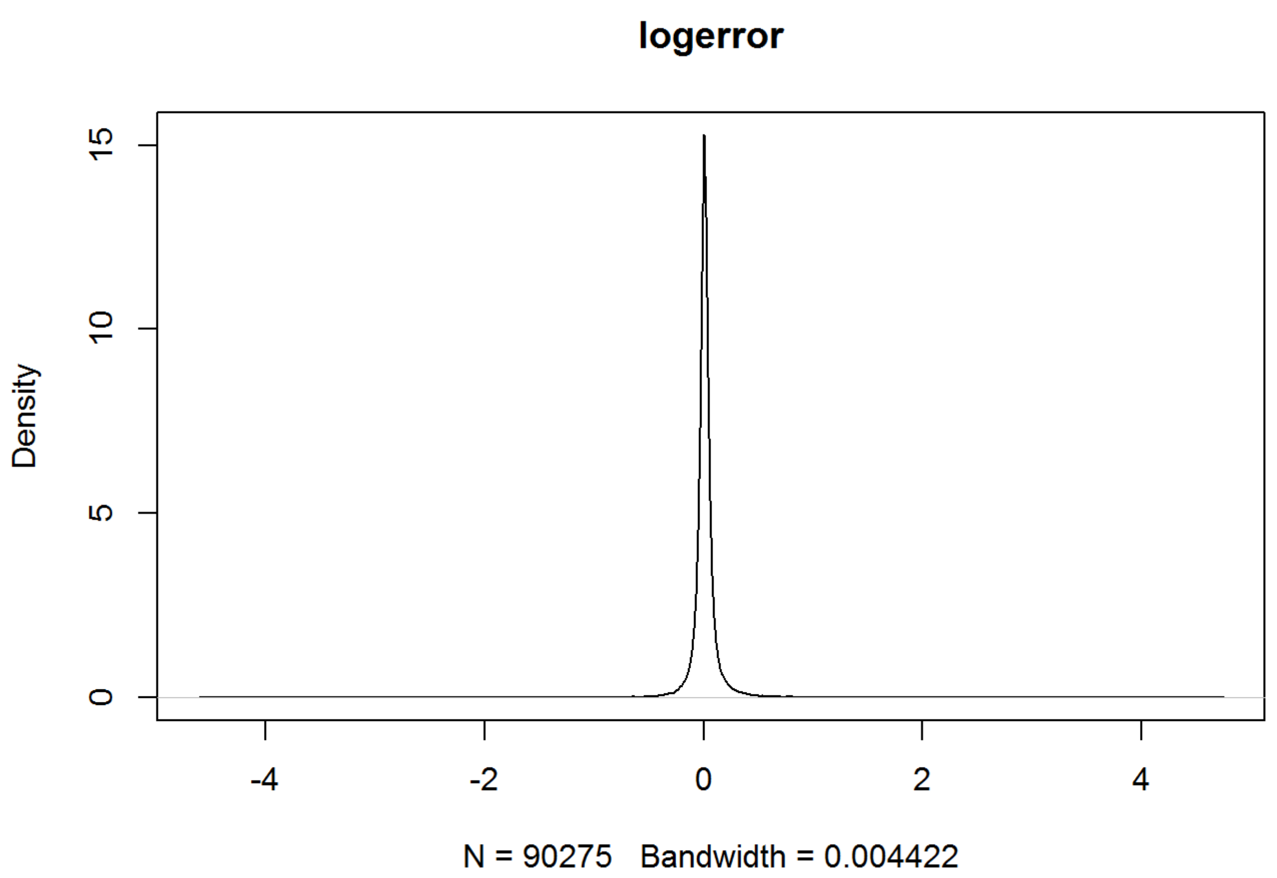
3.1 Univariate analysis

3.11 the outcome variable: logerror

```
summary(train2$logerror)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## -4.60500 -0.02530  0.00600  0.01146  0.03920  4.73700
```

```
plot(density(train2$logerror),main='logerror')
```



Findings: in general, the logerror between estimated house price and the actual sale price is very small

3.12 Continuous variable

Roomcount related

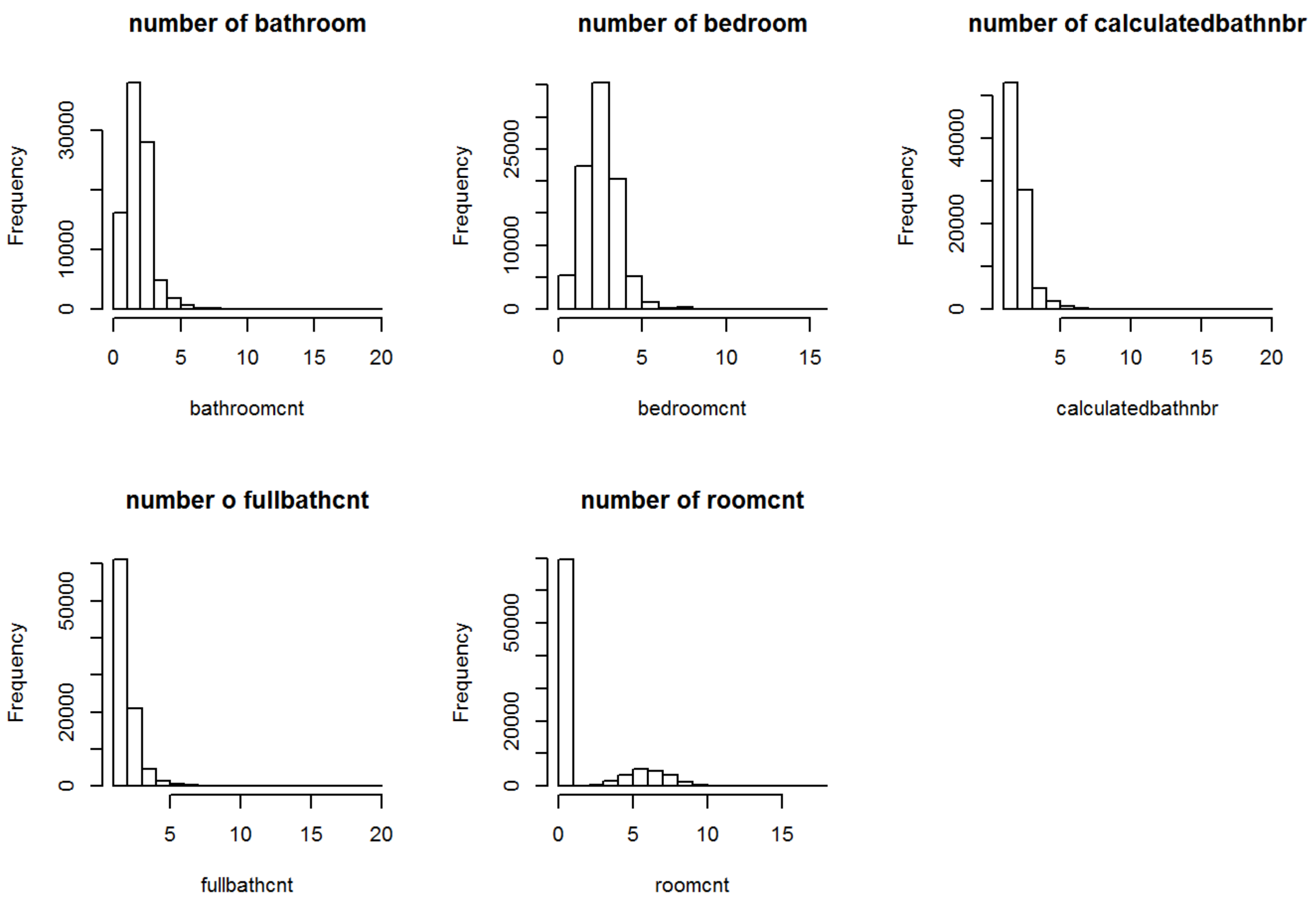
```
names(train2)
```

```
##  [1] "parcelid"                     "logerror"
##  [3] "transactiondate"              "bathroomcnt"
##  [5] "bedroomcnt"                   "calculatedbathnbr"
##  [7] "calculatedfinishedsquarefeet" "finishedsquarefeet12"
##  [9] "fips"                         "fullbathcnt"
## [11] "hashottuborspa"               "latitude"
## [13] "longitude"                    "lotsizesquarefeet"
## [15] "propertycountylandusecode"    "propertylandusetypeid"
## [17] "propertyzoningdesc"           "rawcensustractandblock"
## [19] "regionidcity"                 "regionidcounty"
## [21] "regionidzip"                  "roomcnt"
## [23] "yearbuilt"                    "fireplaceflag"
## [25] "structuretaxvaluedollarcnt"   "taxvaluedollarcnt"
## [27] "assessmentyear"               "landtaxvaluedollarcnt"
## [29] "taxamount"                    "taxdelinquencyflag"
## [31] "censustractandblock"
```

```
summary(train2[,c(4,5,6,10,22)])
```

```
##   bathroomcnt       bedroomcnt     calculatedbathnbr  fullbathcnt
##  Min.   : 0.000   Min.   : 0.000   Min.   : 1.000    Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000    1st Qu.: 2.000
##  Median : 2.000   Median : 3.000   Median : 2.000    Median : 2.000
##  Mean   : 2.279   Mean   : 3.032   Mean   : 2.309    Mean   : 2.241
##  3rd Qu.: 3.000   3rd Qu.: 4.000   3rd Qu.: 3.000    3rd Qu.: 3.000
##  Max.   :20.000   Max.   :16.000   Max.   :20.000    Max.   :20.000
##                                    NA's   :1182      NA's   :1182
##     roomcnt
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 0.000
##  Mean   : 1.479
##  3rd Qu.: 0.000
##  Max.   :18.000
##
```

```
par(mfrow=c(2,3))
hist(train2[,4],xlab='bathroomcnt',main='number of bathroom')
hist(train2[,5],xlab='bedroomcnt',main='number of bedroom')
hist(train2[,6],xlab='calculatedbathnbr',main='number of calculatedbathnbr')
hist(train2[,10],xlab='fullbathcnt',main='number o fullbathcnt')
hist(train2[,22],xlab='roomcnt',main='number of roomcnt')
```



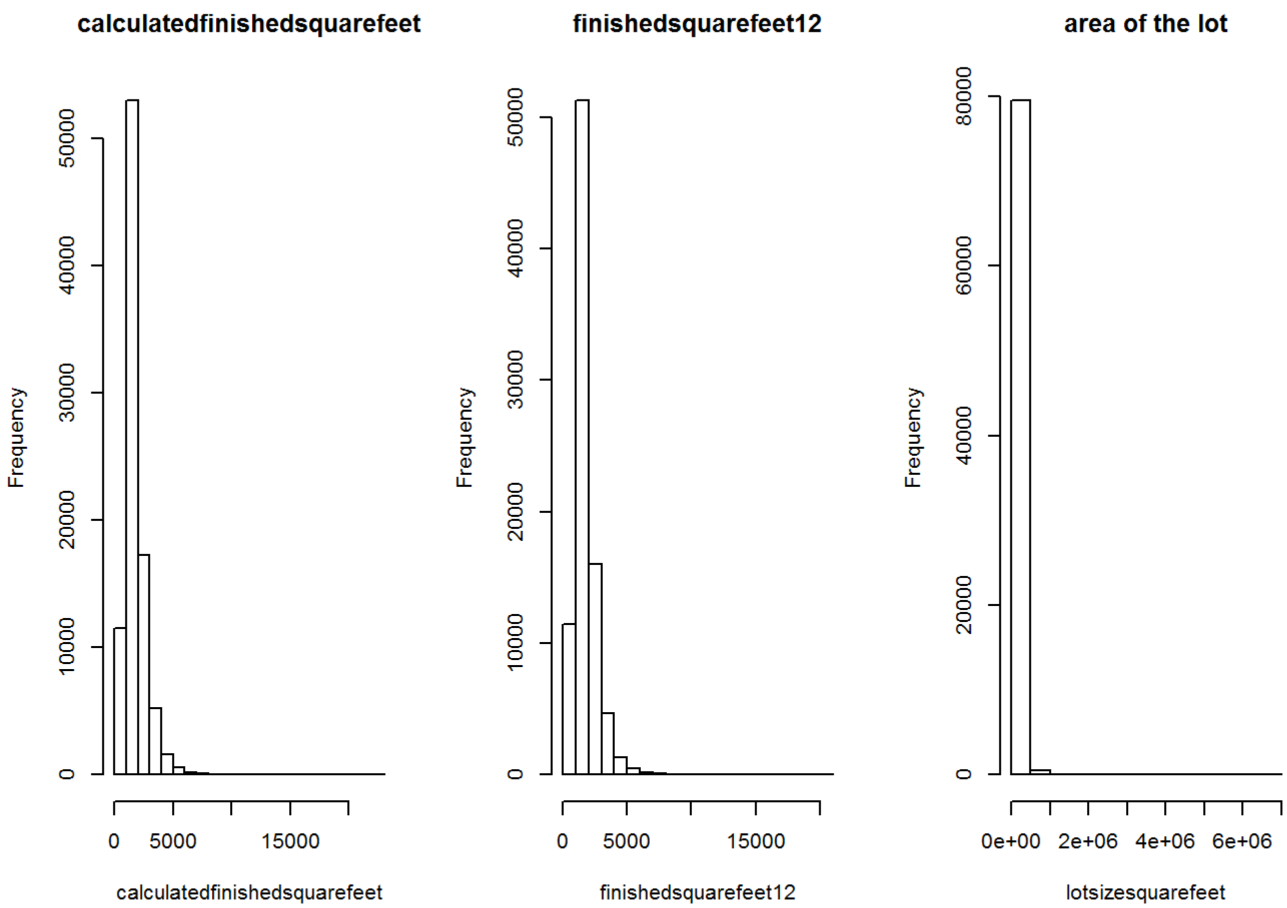
Findings: 1) the distribution of bathroom and bedroom are positively skewed 2) most houses have less than 5 bedroom and bathrooms 3) apart from some houses with only one room, a lot of houses have 5~8 rooms in total 4) bathroomcnt, bedroomcnt and roomcnt have better data quality cuz they do not have NAs

House size related

```
summary(train2[,c(7,8,14)])
```

```
##  calculatedfinishedsquarefeet finishedsquarefeet12 lotsizesquarefeet
##  Min.   :    2                Min.   :    2        Min.   :    167
##  1st Qu.: 1184                1st Qu.: 1172        1st Qu.:   5703
##  Median : 1540                Median : 1518        Median :   7200
##  Mean   : 1773                Mean   : 1745        Mean   :  29110
##  3rd Qu.: 2095                3rd Qu.: 2056        3rd Qu.:  11686
##  Max.   :22741                Max.   :20013        Max.   :6971010
##  NA's   :661                  NA's   :4679         NA's   :10150
```

```
par(mfrow=c(1,3))
hist(train2[,7],xlab='calculatedfinishedsquarefeet',main='calculatedfinishedsquarefeet')
hist(train2[,8],xlab='finishedsquarefeet12',main='finishedsquarefeet12')
hist(train2[,14],xlab='lotsizesquarefeet',main='area of the lot')
```



House location related variables

```
summary(train2[,c(12,13)])
```

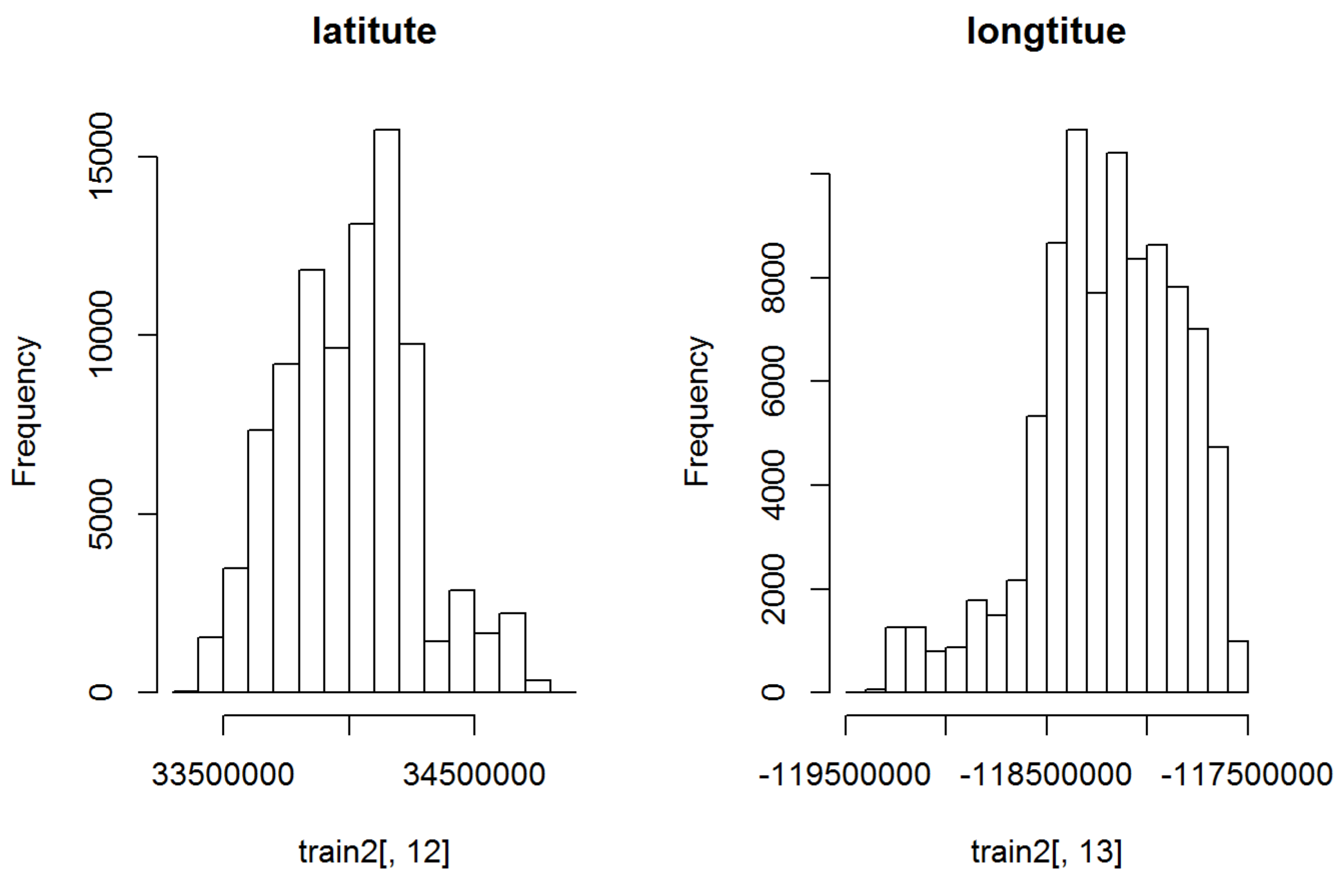
```
##     latitude          longitude
##  Min.   :33339295   Min.   :-119447865
##  1st Qu.:33811538   1st Qu.:-118411692
##  Median :34021500   Median :-118173431
##  Mean   :34005411   Mean   :-118198868
##  3rd Qu.:34172742   3rd Qu.:-117921588
##  Max.   :34816009   Max.   :-117554924
```

```
par(mfrow=c(1,2))
hist(train2[,12],main='latitute')
```

```
## Warning in n * h: 整数上溢产生了NA
```

```
hist(train2[,13],main='longtitue')
```

```
## Warning in n * h: 整数上溢产生了NA
```

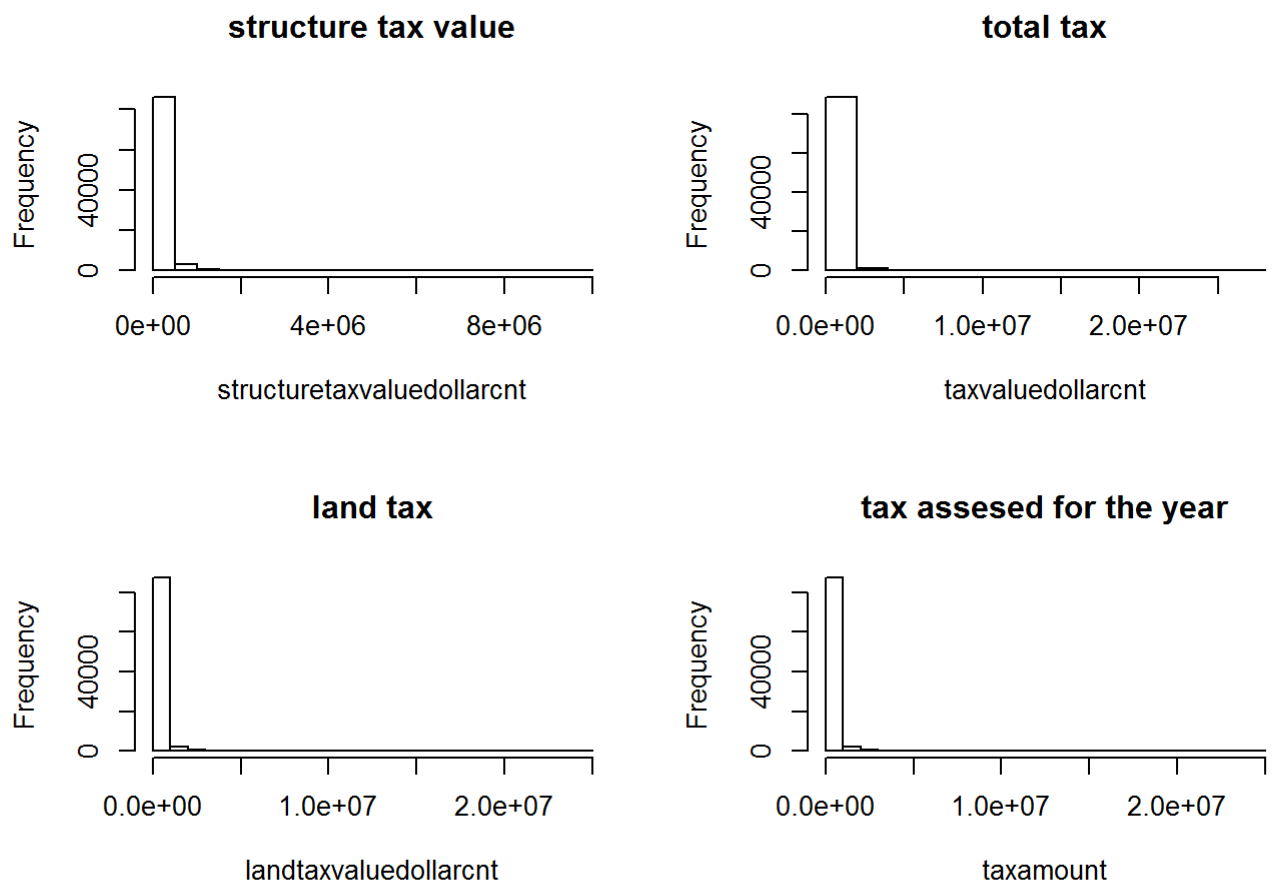


House value related variables

```
summary(train2[,c(25,26,28,29)])
```

```
##  structuretaxvaluedollarcnt taxvaluedollarcnt  landtaxvaluedollarcnt
##  Min.   :    100            Min.   :       22  Min.   :       22
##  1st Qu.:  81245            1st Qu.:   199023  1st Qu.:    82228
##  Median : 132000            Median :   342872  Median :   192970
##  Mean   : 180093            Mean   :   457673  Mean   :   278335
##  3rd Qu.: 210534            3rd Qu.:   540589  3rd Qu.:   345420
##  Max.   :9948100            Max.   :27750000   Max.   :24500000
##  NA's   :380                NA's   :1          NA's   :1
##    taxamount
##  Min.   :    49.1
##  1st Qu.:  2872.8
##  Median :  4542.8
##  Mean   :  5984.0
##  3rd Qu.:  6901.1
##  Max.   :321936.1
##  NA's   :6
```

```
par(mfrow=c(2,2))
hist(train2[,25],xlab='structuretaxvaluedollarcnt',main='structure tax value')
hist(train2[,26],xlab='taxvaluedollarcnt',main='total tax')
hist(train2[,28],xlab='landtaxvaluedollarcnt',main='land tax')
hist(train2[,28],xlab='taxamount',main='tax assesed for the year')
```



####Date variables

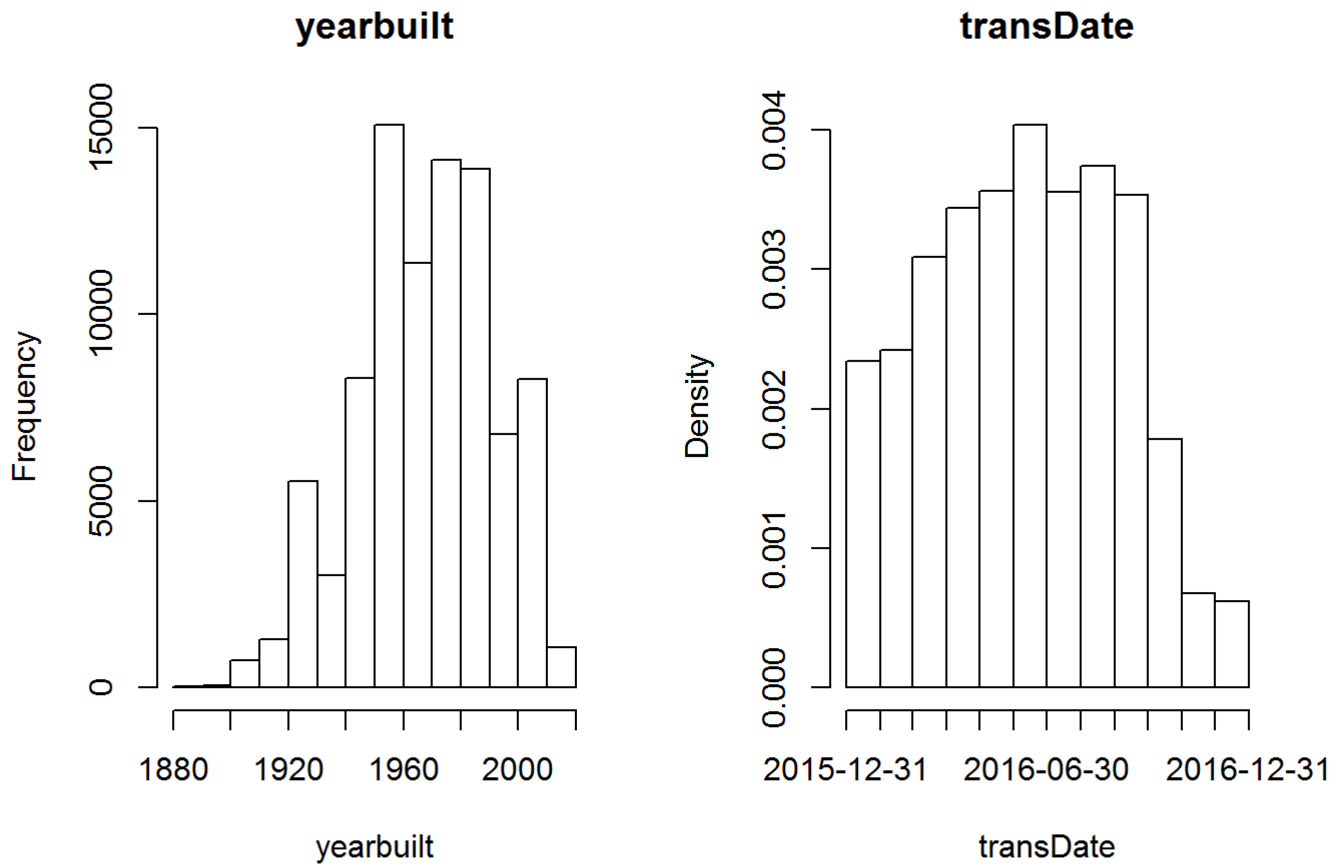
```
#convert transactiondate from character to date format and create a new variable transdate to s
tore it
train2$transDate<-as.Date(train2$transactiondate,'%Y-%m-%d')
summary(train2[,c(23,27,32)])
```

```
##    yearbuilt    assessmentyear   transDate
##  Min.   :1885   Min.   :2015   Min.   :2016-01-01
##  1st Qu.:1953   1st Qu.:2015   1st Qu.:2016-04-05
##  Median :1970   Median :2015   Median :2016-06-14
##  Mean   :1969   Mean   :2015   Mean   :2016-06-11
##  3rd Qu.:1987   3rd Qu.:2015   3rd Qu.:2016-08-19
##  Max.   :2015   Max.   :2015   Max.   :2016-12-30
##  NA's   :756
```

```
str(train2[,c(23,27,32)])
```

```
## 'data.frame':    90275 obs. of  3 variables:
##  $ yearbuilt     : int  1978 1959 1973 1973 1973 1960 1974 1964 1962 1961 ...
##  $ assessmentyear: int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ transDate     : Date, format: "2016-08-02" "2016-08-02" ...
```

```
par(mfrow=c(1,2))
hist(train2[,23],xlab='yearbuilt',main='yearbuilt')
hist(train2[,32],xlab='transDate',main='transDate',breaks='months')
```



3.1.3categorical variables

House adress related variables

```
#fips distribution
table(train2$fips)/nrow(train2)
```

```
##
##       6037       6059       6111
## 0.64883966 0.27144835 0.07971199
```

```
#6037 for LA county, 6059 for Orange county and 6111 for ventura
```

```
#city distribution
#table(train2$regionidcity)

#county distribution
#table(train2$regionidcounty)

#zip
#table(train2$regionidzip)
```

House feature variable

```
table(train2$hashottuborspa)/nrow(train2)
```

```
##
##                  true
## 0.97380227 0.02619773
```

```
table(train2$fireplaceflag)/nrow(train2)
```

```
##
##                    true
## 0.997540847 0.002459153
```

Findings: very few houses have fireplace or spa tub

propertyuse variable

```
table(train2$propertycountylandusecode)
```

```
##
##           0   010  0100  0101  0102  0103  0104  0108  0109  010C  010D
##     1     1     1 30846  7435     3   100   348    46    27 10264  2209
##  010E  010F  010G  010H  010M  010V  0110  0111  0114  012C  012D  012E
##  2286    28    80    72    59   201     4     1     1   523     4     7
##  0130  0131  01DC  01DD  01HC  0200  0201  020G  020M  0210  0300  0301
##     1     1   251     1   148  2153    43     4     1     1   578     1
##  0303  030G  0400  0401  040A  040V  0700  070D     1  100V  1011  1012
##     1     2   747     4     1     6    54     7  2915     5     2     2
##  1014   105  1110  1111  1112  1116  1117  1128  1129  1200  1210   122
##    32     4  1117  3883     5    11    46   356  1643     1    47 15383
##  1222  1310  1321  1333   135  1410  1420  1421  1432  1720  1722   200
##    28     4     7     1    35    13     1     2     2     7     1     1
##    34    38  6050    73  8800    96
##  5946   106     1    11     1   104
```

```
#table(train2$propertyzoningdesc)
```

tax related

```
table(train2$taxdelinquencyflag)/nrow(train2)
```

```
##
##                     Y
## 0.98024924 0.01975076
```

Findings:2% of the houses have tax delinquency

3.2 Bivariate analysis

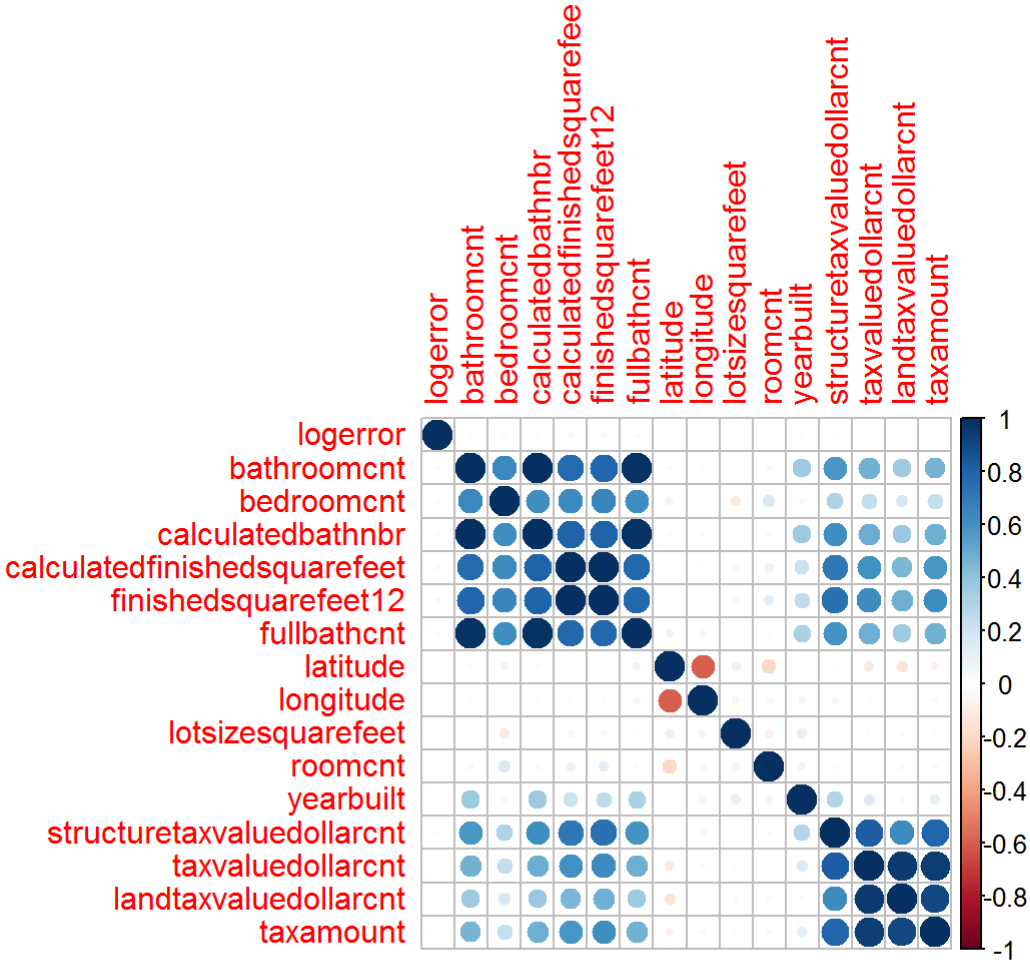
3.2.1 continuous vs outcome

```
corr<-cor(train2[,c(2,4:8,10,12:14,22,23,25,26,28,29)],use='pairwise.complete.obs')
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.3
```

```
## corrplot 0.84 loaded
```

```
corrplot(corr)
```

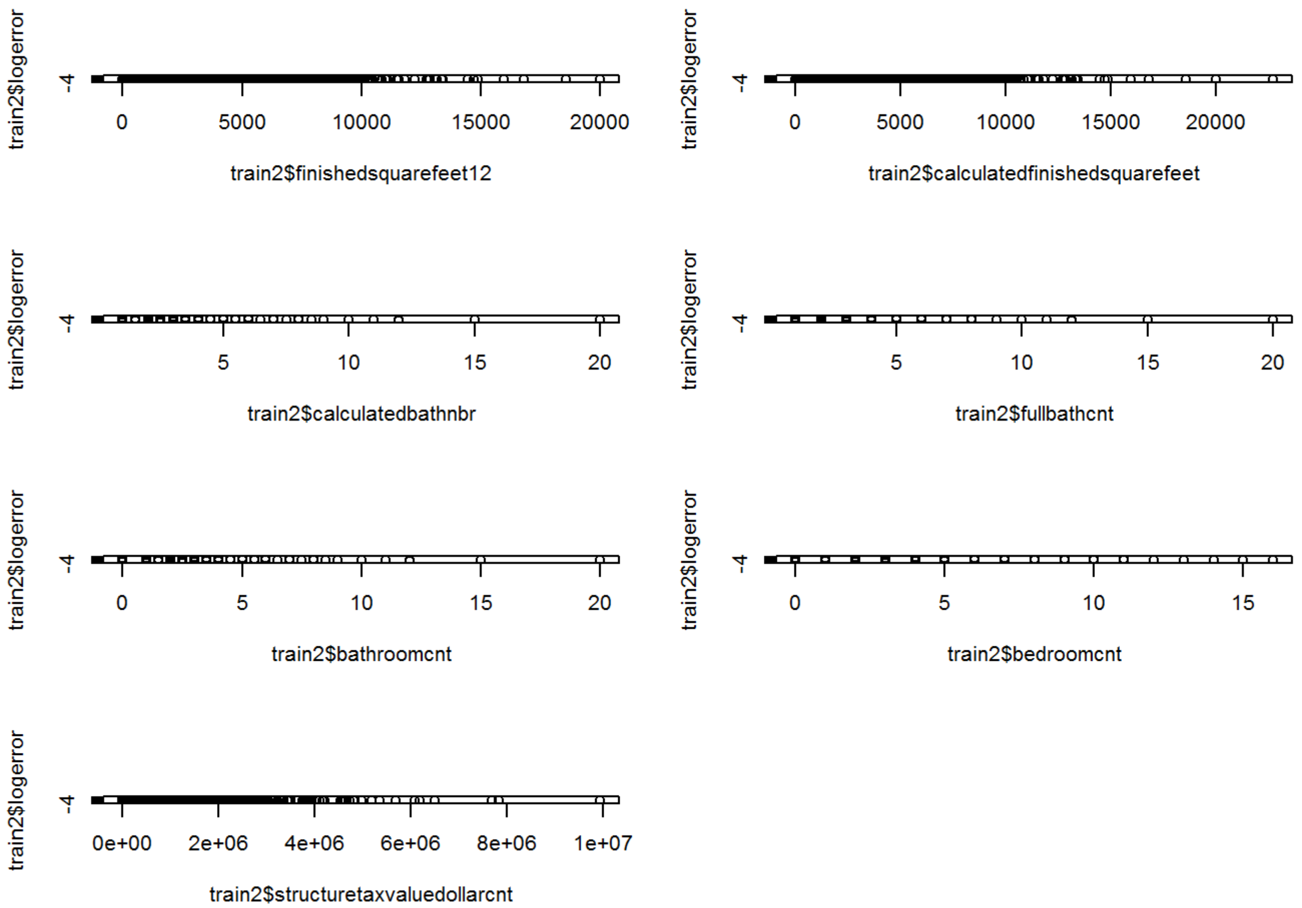



```
cor_logerror<-sort(corr[,1],decreasing = T)
cor_logerror
```

```
##                     logerror          finishedsquarefeet12
##                  1.000000000                   0.041922367
## calculatedfinishedsquarefeet             calculatedbathnbr
##                  0.038784069                   0.029447685
##                  fullbathcnt                   bathroomcnt
##                  0.028845122                   0.027889287
##                   bedroomcnt    structuretaxvaluedollarcnt
##                  0.025467090                   0.022084970
##                    yearbuilt             taxvaluedollarcnt
##                  0.017312211                   0.006507999
##                      roomcnt                      latitude
##                  0.005759796                   0.004915466
##            lotsizesquarefeet         landtaxvaluedollarcnt
##                  0.004835250                  -0.003051035
##                    longitude                     taxamount
##                 -0.003432217                  -0.006671116
```

```
#corelations are all soooo weak
```

```
#show corelation between each variable and the logerror
par(mfrow=c(4,2))
plot(train2$finishedsquarefeet12,train2$logerror)
plot(train2$calculatedfinishedsquarefeet,train2$logerror)
plot(train2$calculatedbathnbr,train2$logerror)
plot(train2$fullbathcnt,train2$logerror)
plot(train2$bathroomcnt,train2$logerror)
plot(train2$bedroomcnt,train2$logerror)
plot(train2$structuretaxvaluedollarcnt,train2$logerror)
```



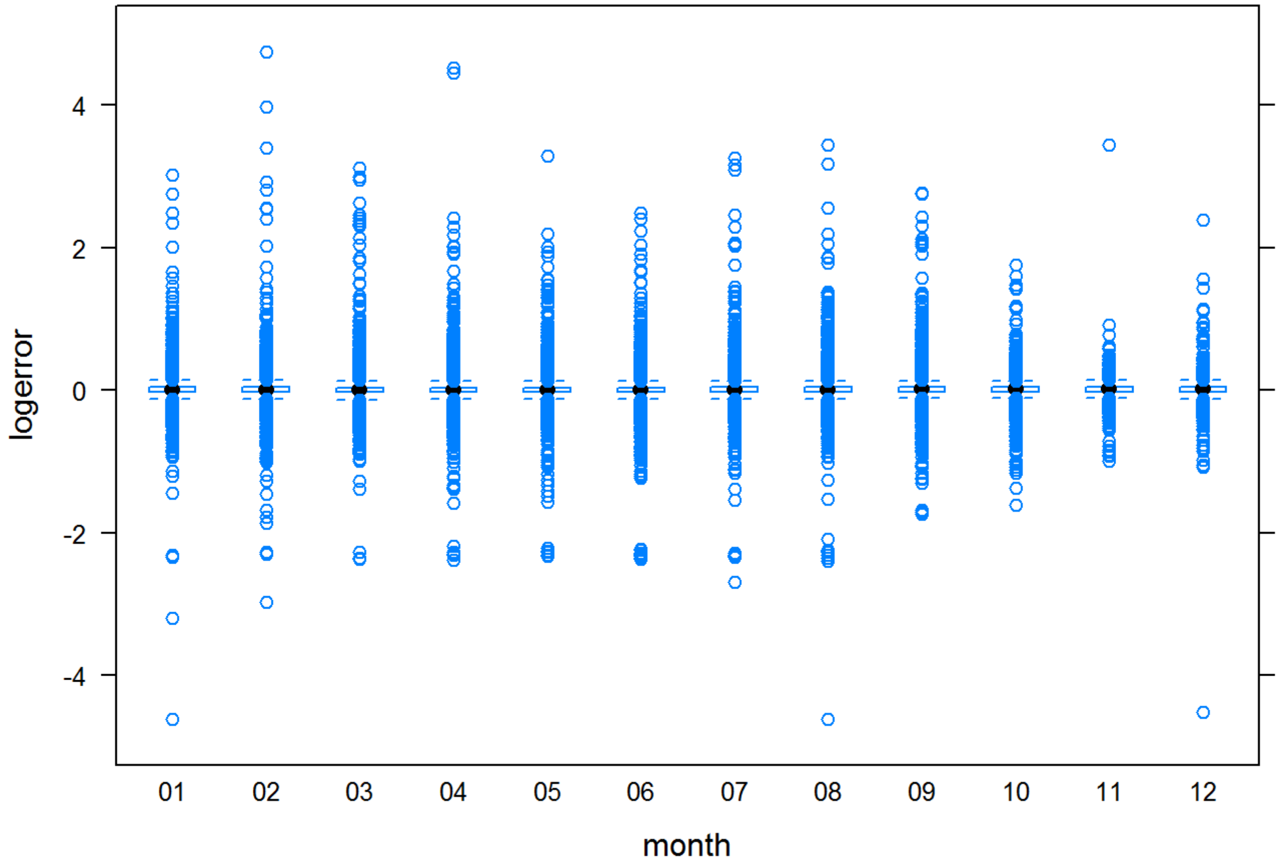
the points are randomly scattered and indicates the corelation is weak

correlation between the transDate and the logerror

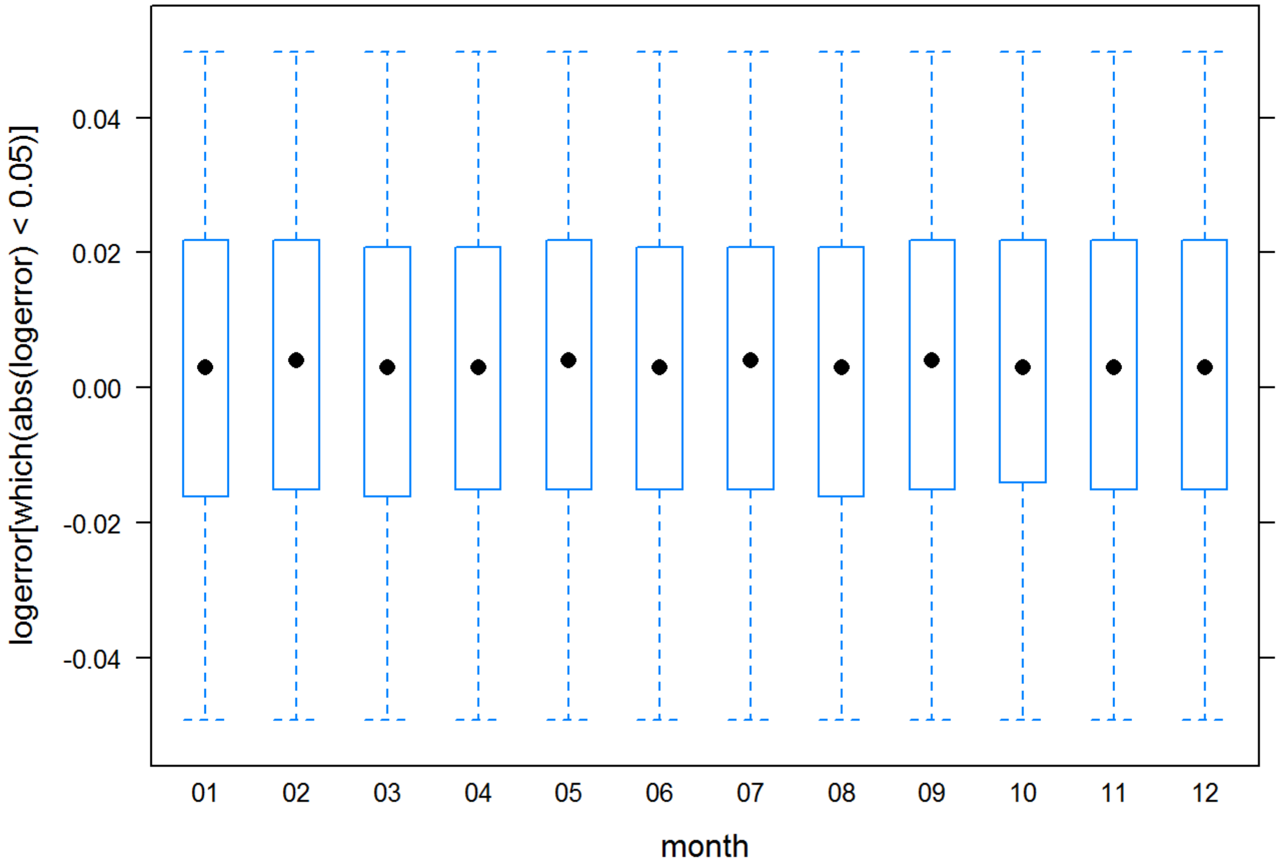
```
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 3.4.3
```

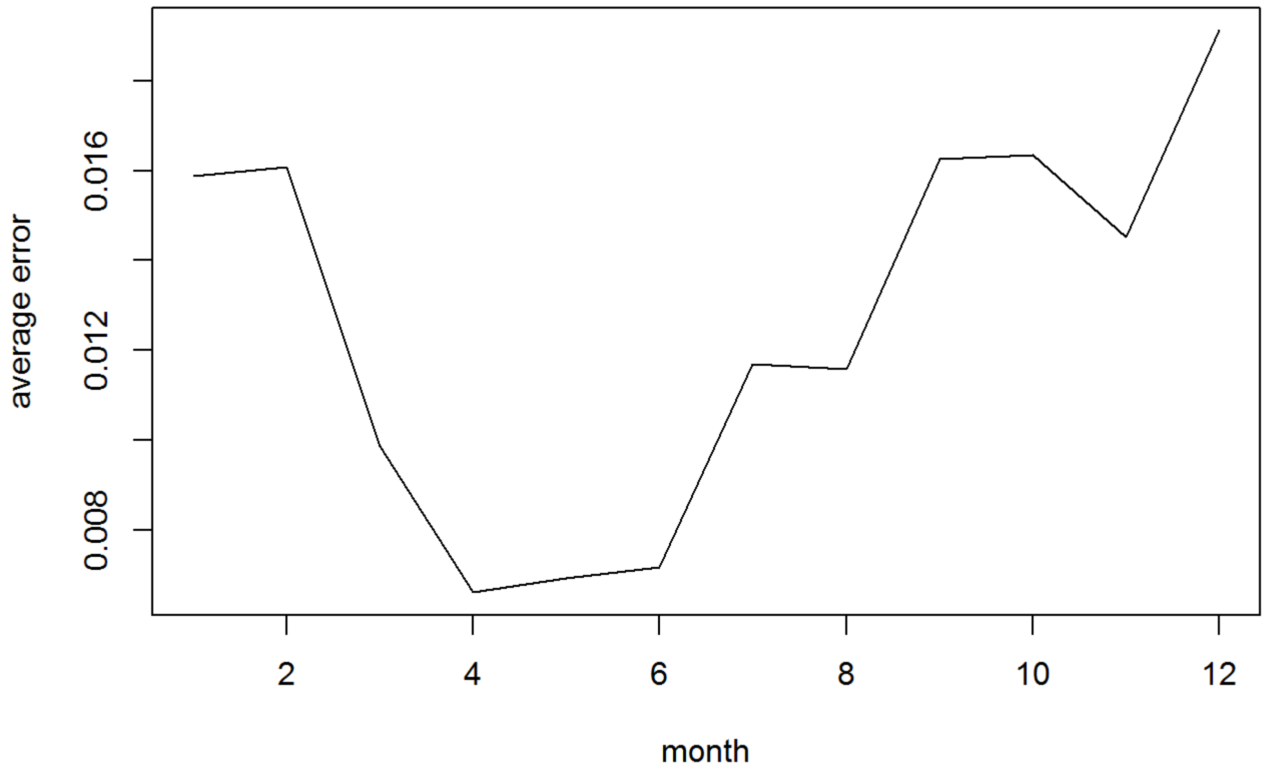
```
train2$transMonth<-sapply(strsplit(train2$transactiondate,'-'),function(x) x[2])
bwplot(logerror~transMonth,data=train2,xlab='month')
```




```
bwplot(logerror[which(abs(logerror)<0.05)]~transMonth,data=train2,xlab='month')
```



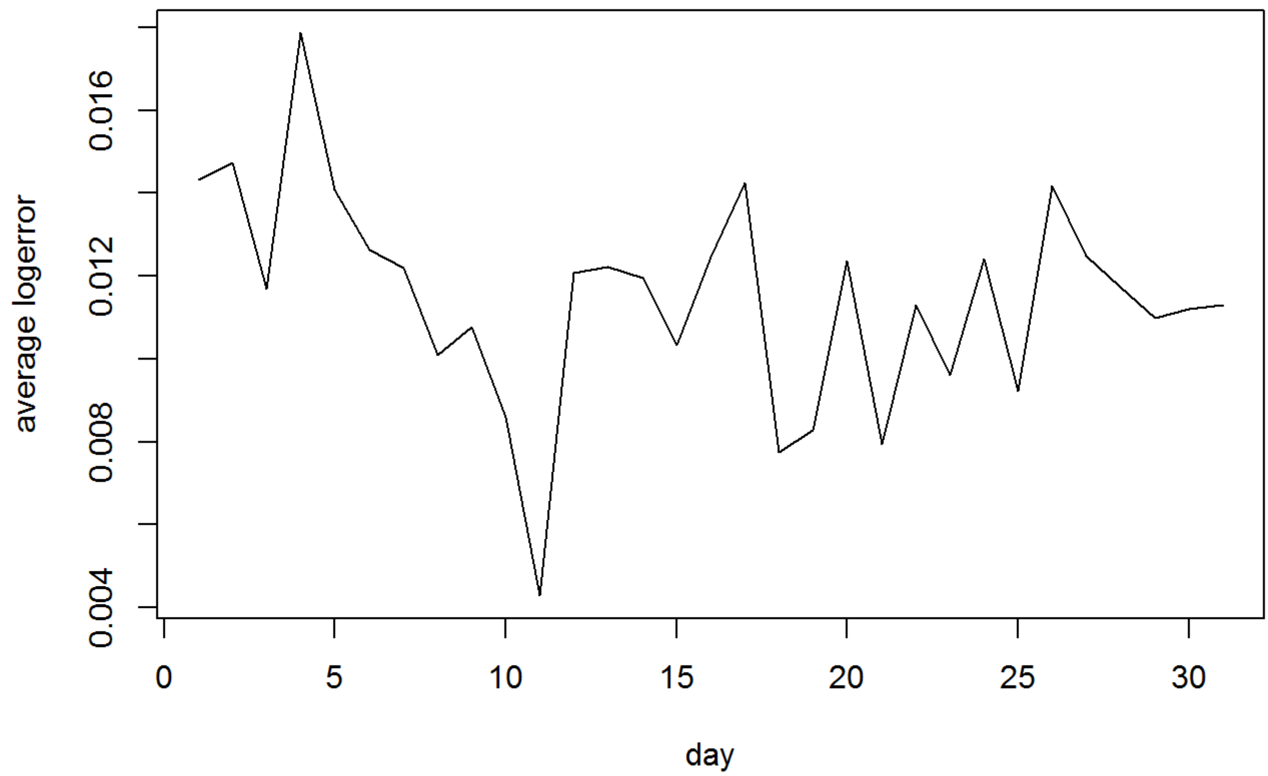
```
err.month<-by(train2,train2$transMonth,function(x){return(mean(x$logerror))})
plot(names(err.month),err.month,type='l',xlab='month',ylab='average error')
```



Findings: distributon of logerror is similar, but average logerror differs in the month of april, may, june

transaction day with logerror

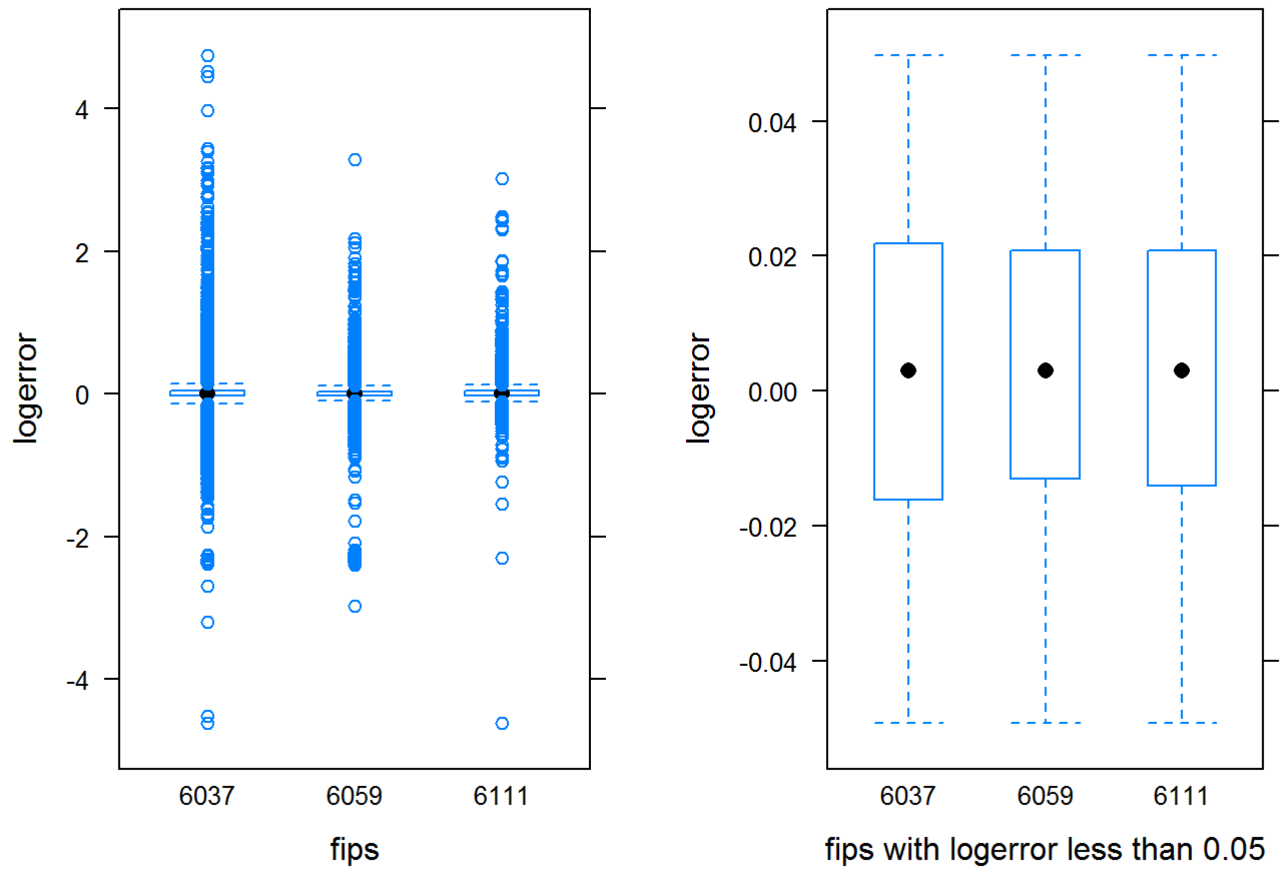
```
train2$transDay<-sapply(strsplit(train2$transactiondate,'-'),function(x) x[3])
err.day<-by(train2,train2$transDay,function(x){return(mean(x$logerror))})
plot(names(err.day),err.day,type='l',xlab='day',ylab='average logerror')
```



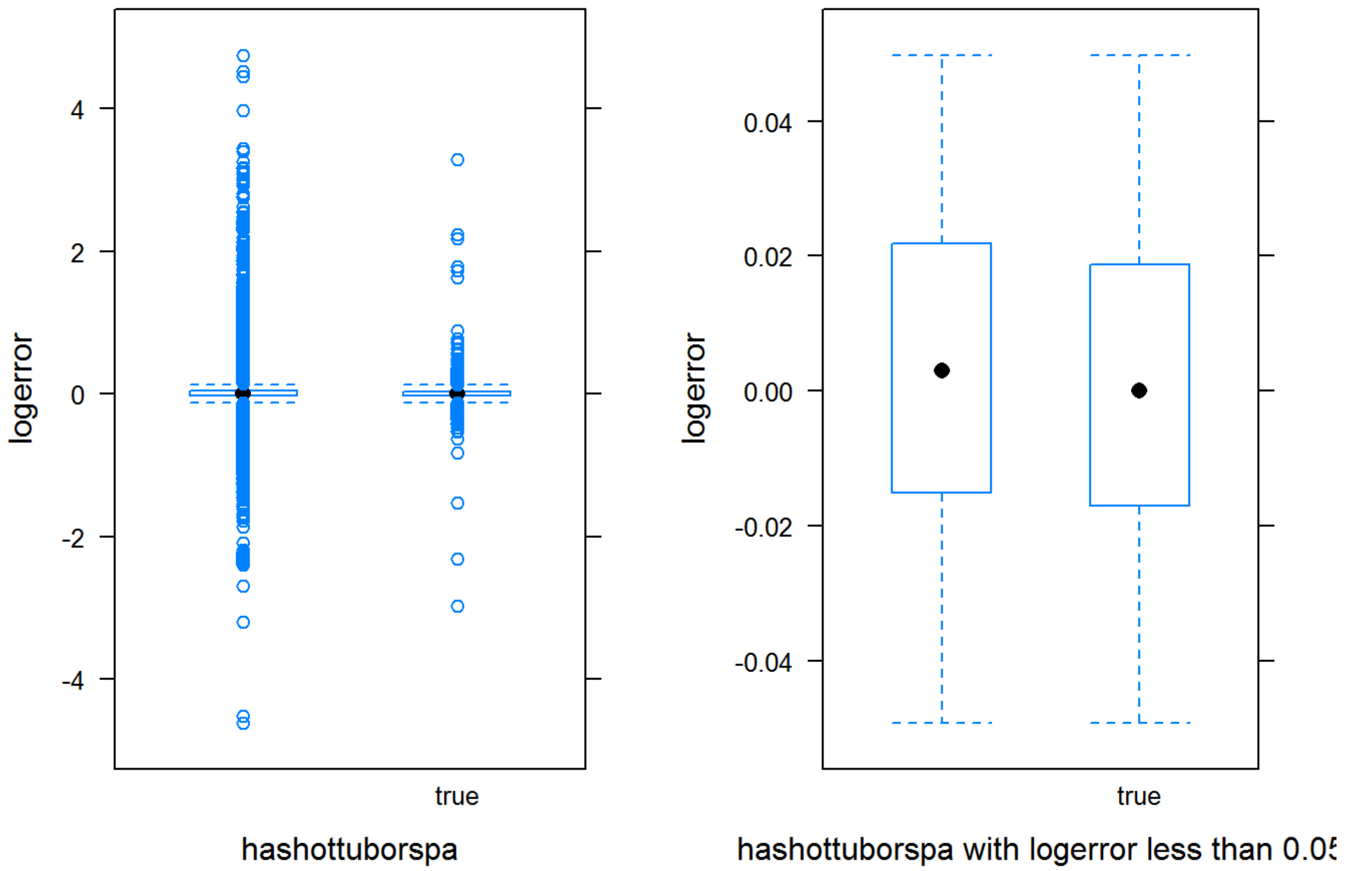
3.2.2categorical variables vs outcome

For categorical variables , I would explore the relationship between the logerror and the fips, hashottuborspa,fireplaceflag,propertylanusetypeid, taxdelinquency

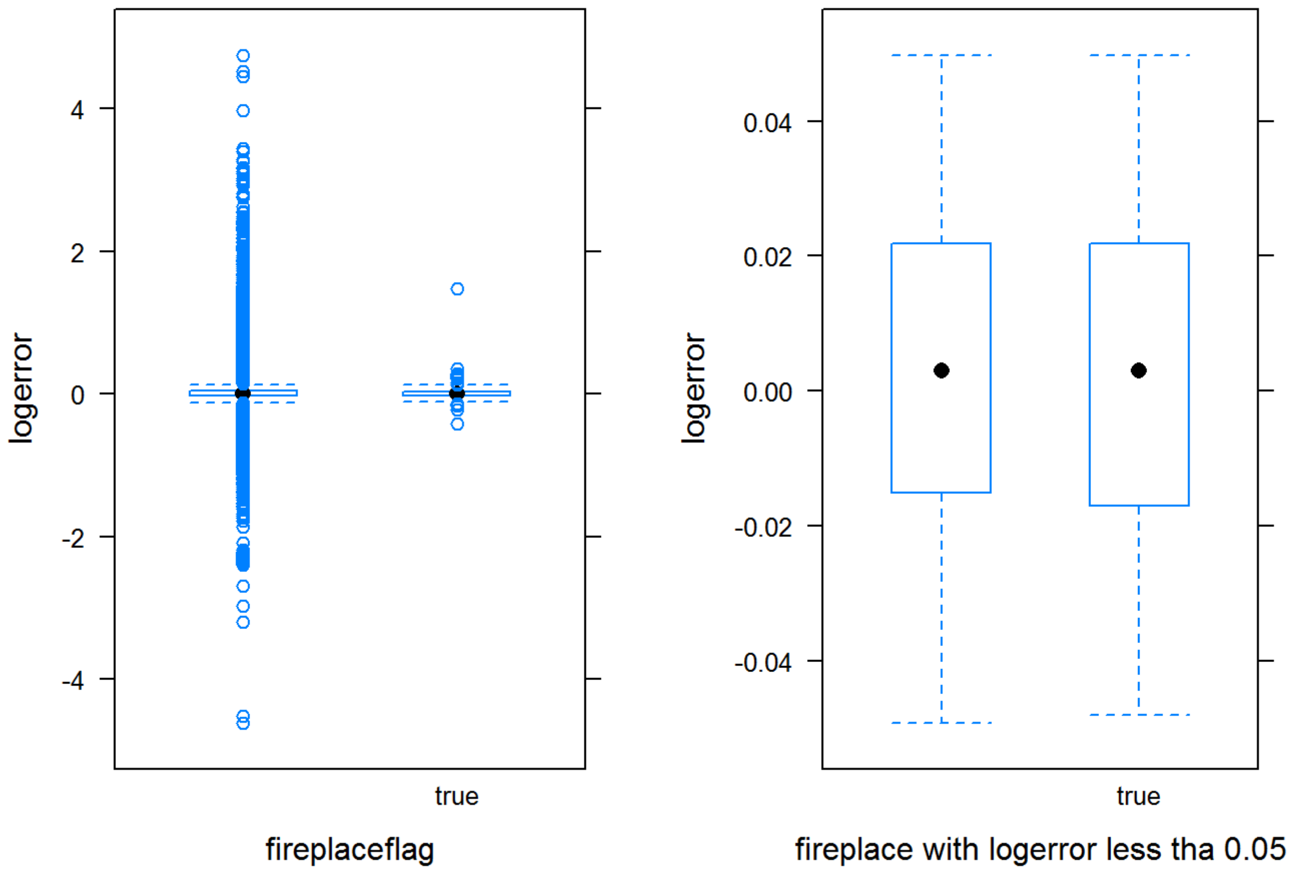
```
library(lattice)
library(gridExtra)
fipsPlot1<-bwplot(logerror~as.character(fips),data=train2,xlab='fips')
fipsPlot2<-bwplot(logerror~as.character(fips),data=subset(train2,abs(logerror)<.05),
                  xlab='fips with logerror less than 0.05')
grid.arrange(fipsPlot1,fipsPlot2,ncol=2)
```



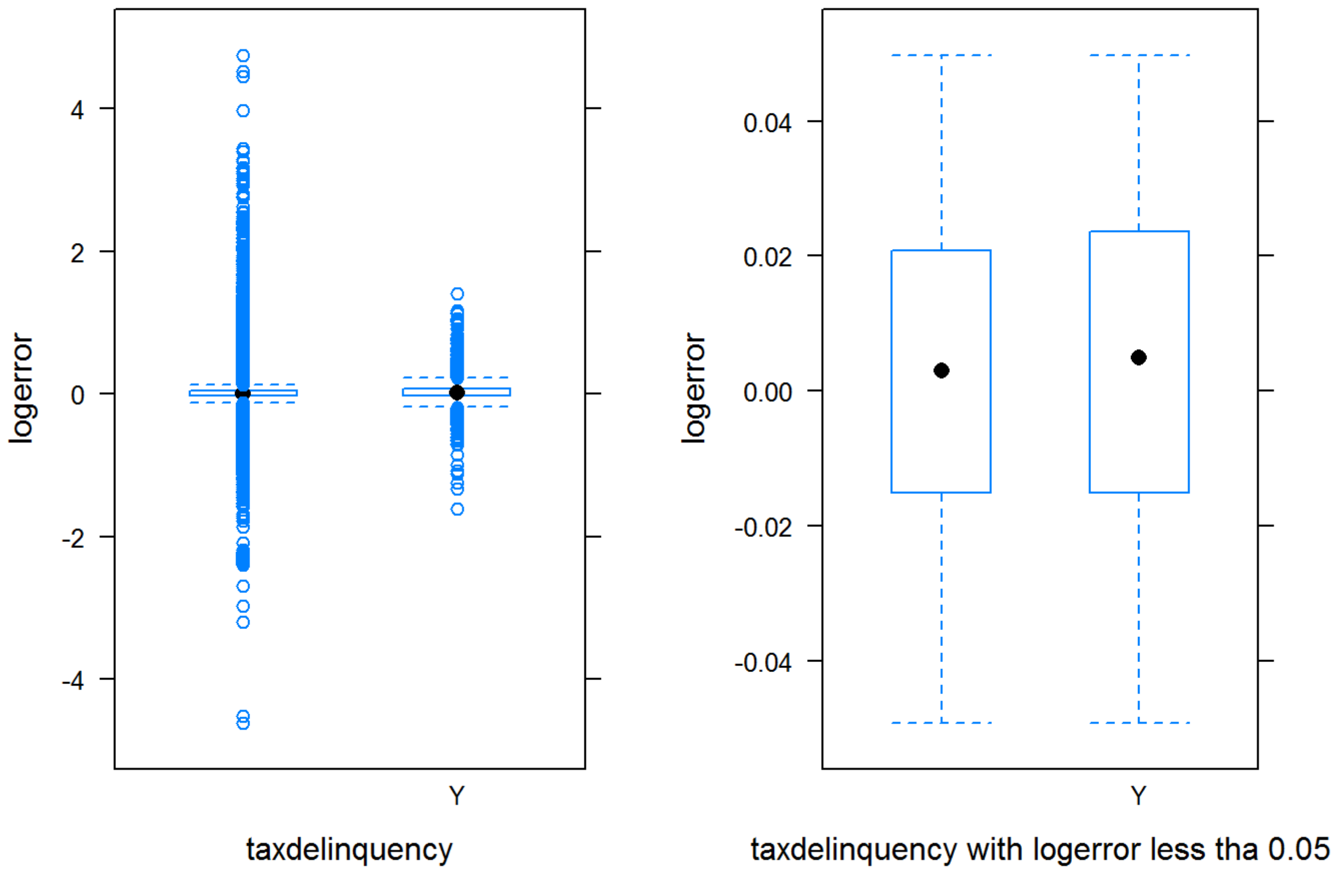

```
spaPlot1<-bwplot(logerror~hashottuborspa,data=train2,xlab='hashottuborspa')
spaPlot2<-bwplot(logerror~hashottuborspa,data=subset(train2,abs(logerror)<.05),
                  xlab='hashottuborspa with logerror less than 0.05')
grid.arrange(spaPlot1,spaPlot2,ncol=2)
```



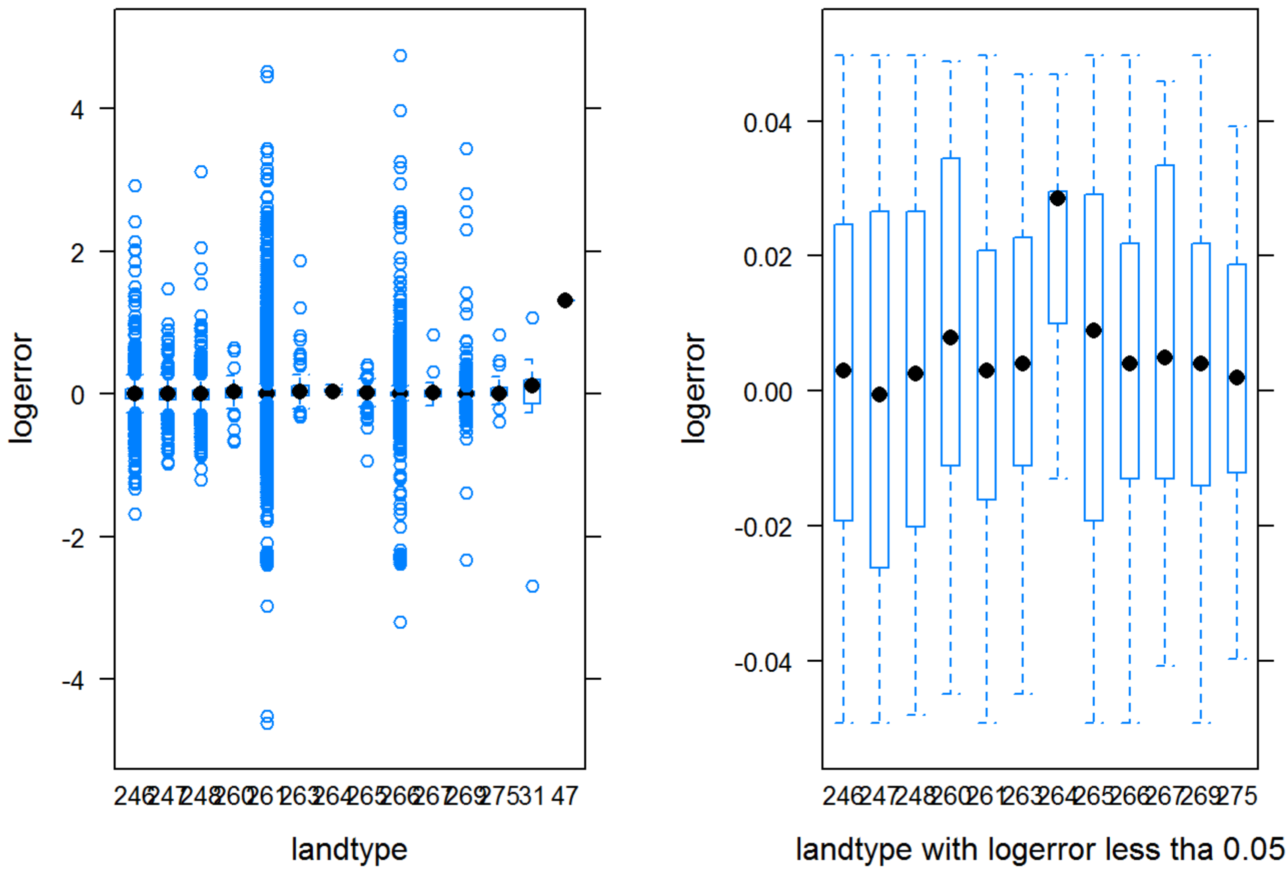

```
firePlot1<-bwplot(logerror~fireplaceflag,data=train2,xlab='fireplaceflag')
firePlot2<-bwplot(logerror~fireplaceflag,data=subset(train2,abs(logerror)<.05),
                                                     xlab='fireplace with logerror less tha 0.0
5')
grid.arrange(firePlot1,firePlot2,ncol=2)
```




```
taxPlot1<-bwplot(logerror~taxdelinquencyflag,data=train2,xlab='taxdelinquency')
taxPlot2<-bwplot(logerror~taxdelinquencyflag,data=subset(train2,abs(logerror)<.05),
                              xlab='taxdelinquency with logerror less tha 0.05')
grid.arrange(taxPlot1,taxPlot2,ncol=2)
```




```
landPlot1<-bwplot(logerror~as.character(propertylandusetypeid),data=train2,xlab='landtype')
landPlot2<-bwplot(logerror~as.character(propertylandusetypeid),data=subset(train2,abs(logerror)
<.05),
                              xlab='landtype with logerror less tha 0.05')
grid.arrange(landPlot1,landPlot2,ncol=2)
```




```
anova(with(train2,lm(logerror~as.character(fips))))
```

```
## Analysis of Variance Table
##
## Response: logerror
##                       Df  Sum Sq  Mean Sq F value   Pr(>F)
## as.character(fips)     2    0.27 0.136935  5.2781 0.005104 **
## Residuals          90272 2342.01 0.025944
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(with(train2,lm(logerror~as.character(propertylandusetypeid))))
```

```
## Analysis of Variance Table
##
## Response: logerror
##                                         Df Sum Sq  Mean Sq F value
## as.character(propertylandusetypeid)     13    2.8 0.215373  8.3094
## Residuals                            90261 2339.5 0.025919
##                                         Pr(>F)
## as.character(propertylandusetypeid) < 2.2e-16 ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(with(train2,lm(logerror~hashottuborspa)))
```

```
## Analysis of Variance Table
##
## Response: logerror
##                   Df  Sum Sq  Mean Sq F value Pr(>F)
## hashottuborspa     1    0.06 0.062835  2.4218 0.1197
## Residuals      90273 2342.22 0.025946
```

```
anova(with(train2,lm(logerror~fireplaceflag)))
```

```
## Analysis of Variance Table
##
## Response: logerror
##                  Df Sum Sq   Mean Sq F value Pr(>F)
## fireplaceflag     1    0.0 0.0032303  0.1245 0.7242
## Residuals     90273 2342.3 0.0259466
```

```
anova(with(train2,lm(logerror~taxdelinquencyflag)))
```

```
## Analysis of Variance Table
##
## Response: logerror
##                       Df  Sum Sq Mean Sq F value    Pr(>F)
## taxdelinquencyflag     1    0.84 0.83984   32.38 1.272e-08 ***
## Residuals          90273 2341.44 0.02594
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Findings: taxdelinquence, fips and propertylandusetypeid really matters

3.2.3 create new variables

number of houses sold by city

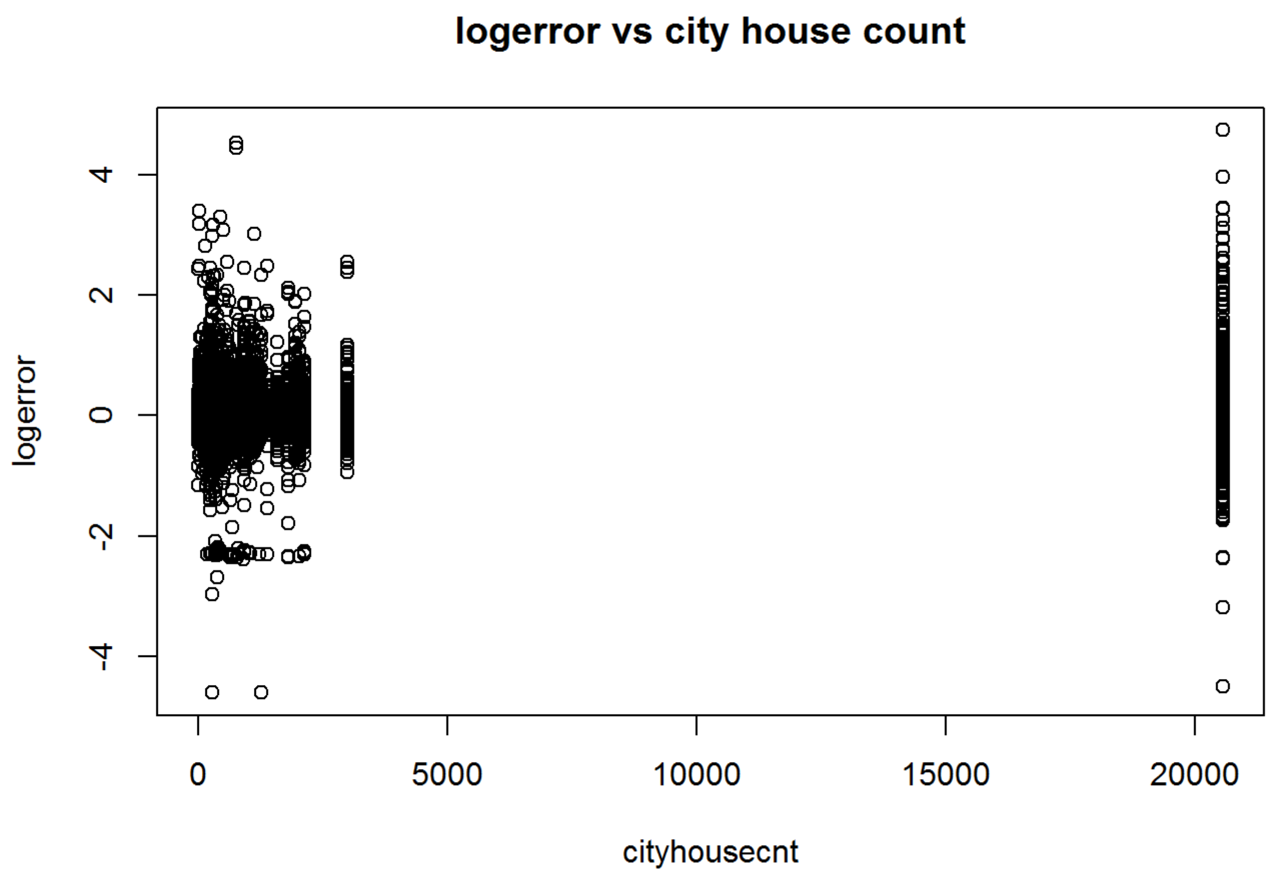
```
#calculate the number of houses sold by city
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.4.3
```

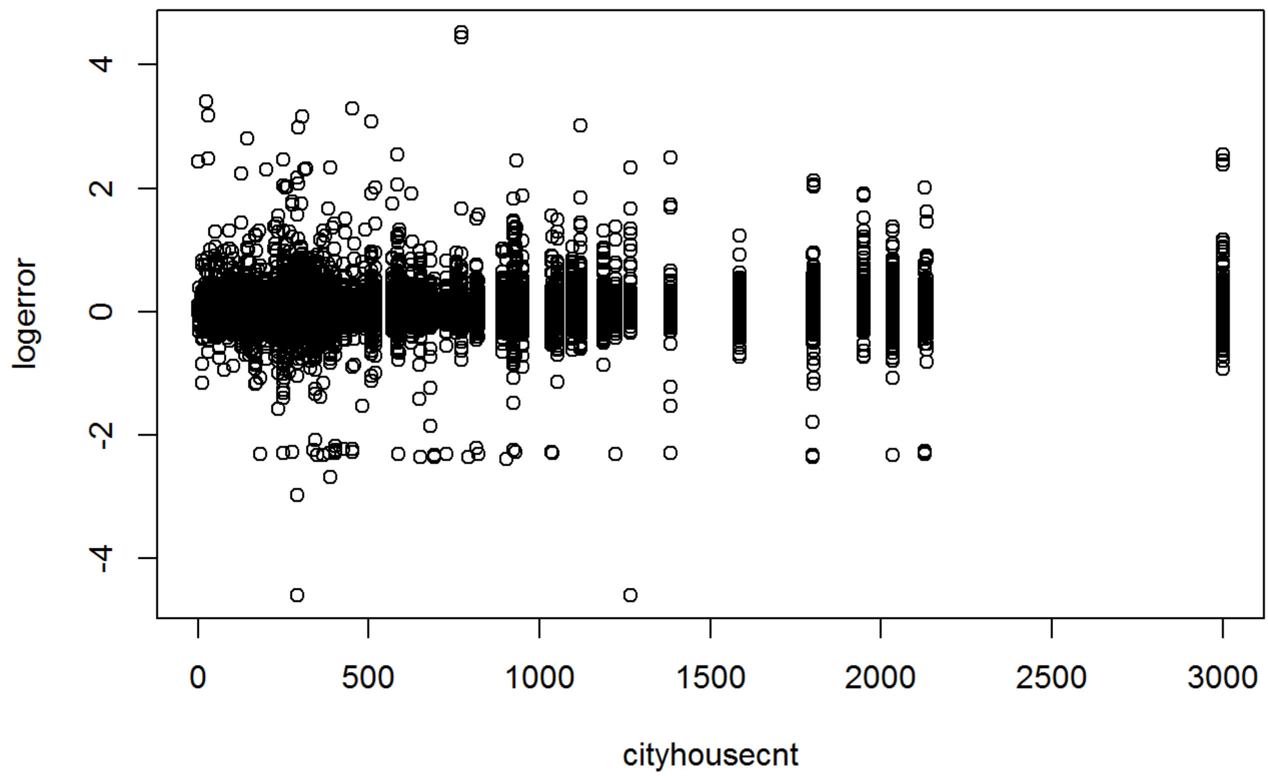
```
house.city<-ddply(train2,.(regionidcity),summarise,cityhousecnt=length(regionidcity))

train3<-merge(train2,house.city,by='regionidcity',all.x=T)

with(train3,plot(cityhousecnt,logerror,main='logerror vs city house count' ))
```




```
train3_sub<-train3[train3$cityhousecnt != max(train3$cityhousecnt),]
with(train3_sub,plot(cityhousecnt,logerror))
```

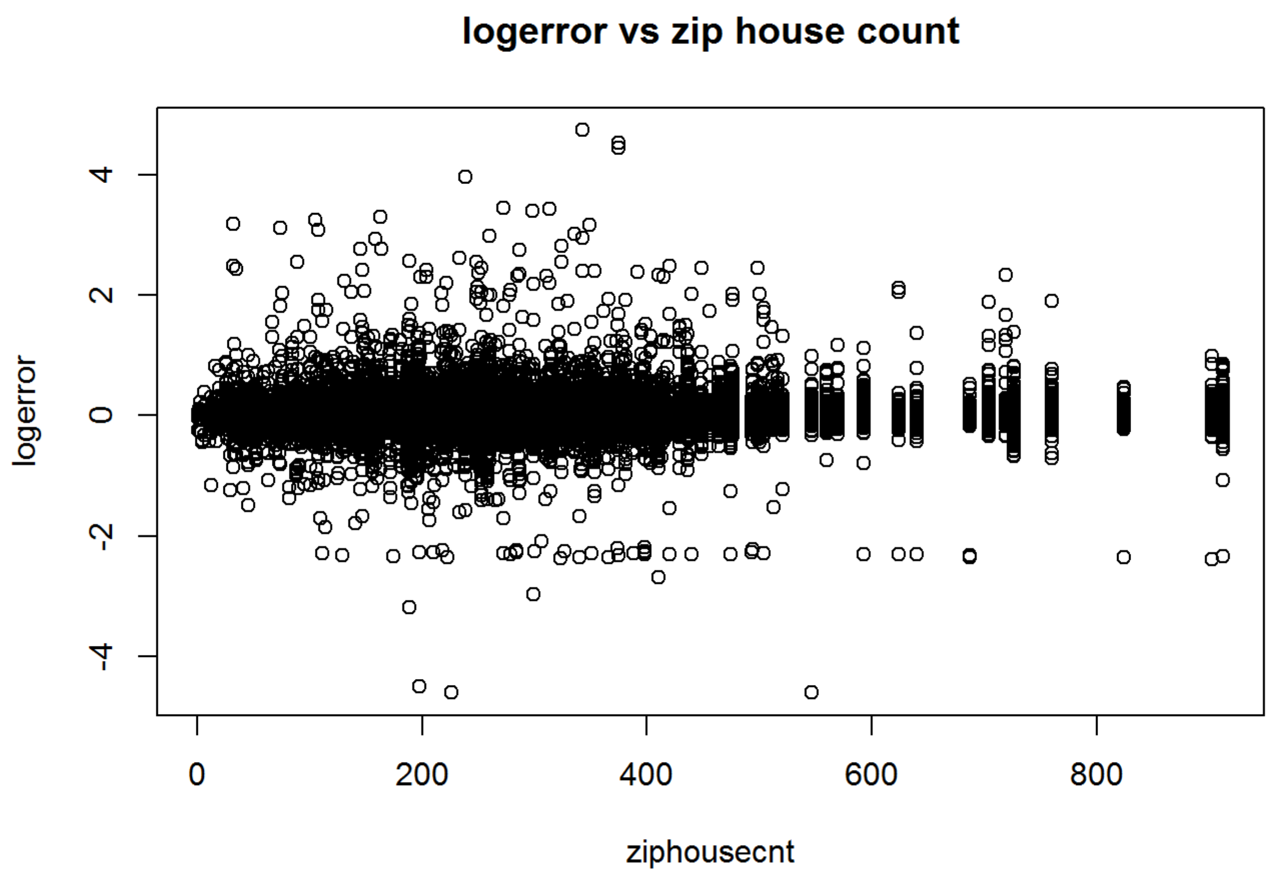


####number of house sold by zip

```
house.zip<-ddply(train2,.(regionidzip),summarise,ziphousecnt=length(regionidzip))

train4<-merge(train2,house.zip,by='regionidzip',all.x=T)

with(train4,plot(ziphousecnt,logerror,main='logerror vs zip house count' ))
```



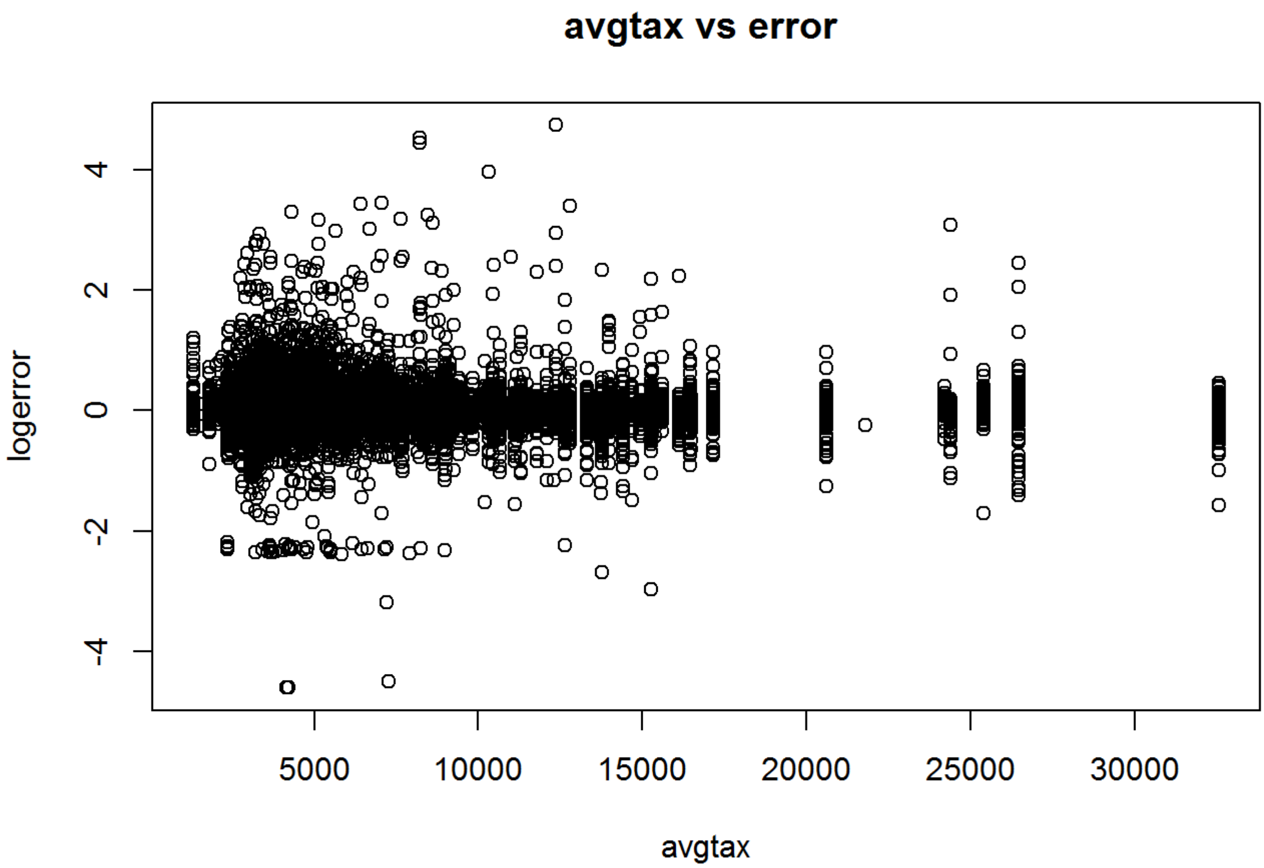

```
cor(train4$ziphousecnt,train$logerror)
```

```
## [1] 0.0002476373
```

community quality

```
price.zip<-ddply(train4,.(regionidzip),summarize,avgtax=mean(taxamount,na.rm = T))
train5<-merge(train4,price.zip,by='regionidzip',all.x=T)

with(train5,plot(avgtax,logerror,main='avgtax vs error ' ))
```


avgtax vs error
logerror
avgtax
-4
-2
0
2
4
5000
10000
15000
20000
25000
30000