

Zillow EDA

Yue Wen

This RMD file is the initial exploratory data analysis of zillow data set.

```
# install necessary library
library(tabplot)
library(lattice)
setwd("D:/data_camp/zillow_project")
train <- read.csv("train_property.csv", stringsAsFactors = FALSE)
```

1. Data cleaning

We noticed that there are some missing values, let us examine the missing value first.

```
#check the number of missing values in each column
num.NA <- sort(colSums(sapply(train,is.na)))
num.NA.perc <- num.NA/dim(train)[1]
```

We can see a lot features have a large portion of missing value. However, before we decided to delete some of these features, let us take a look at whether the missing proportion of each row might influence the respond variable. The purpose of doing so is to make sure that the percentage of missing value in each row do not have influence on logerror.

```
row.NA.perc <- rowSums(sapply(train,is.na))/dim(train)[2]
log.error <- train$logerror
cor(log.error,row.NA.perc)

## [1] -0.005944324
```

We can see the correlation is all most 0. So for now, let us delete these features that has more than 80% missing value

```

# if more than 80% of feature is missing, then we delete the feature.
delete.thredshold <- 0.2
remain.rows <- names(num.NA)[num.NA.perc < delete.thredshold]
train <- train[, remain.rows]
print(paste(ncol(train)/ncol(train), "of the features are not deleted"))

## [1] "1 of the features are not deleted"

```

Done! We cleaned up the NA part for now.

The other thing we have to do is to set som

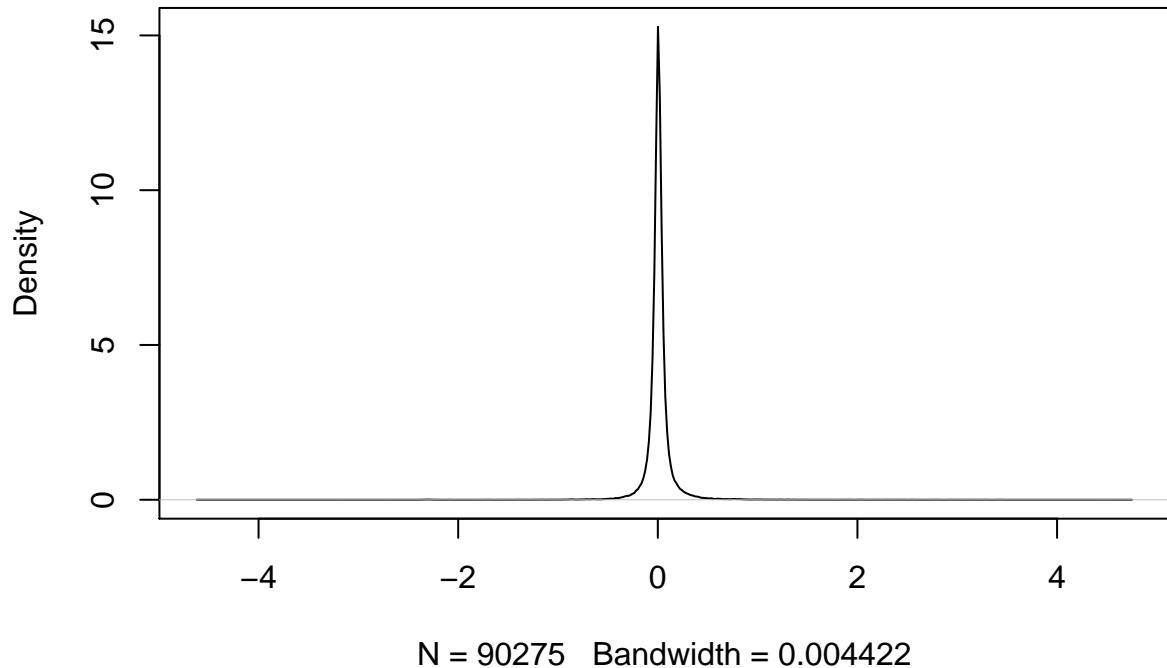
```
train[, c('fips', 'propertylandusetypeid', 'rawcensussubtractandblock', 'regionidcc')]
```

2. EDK

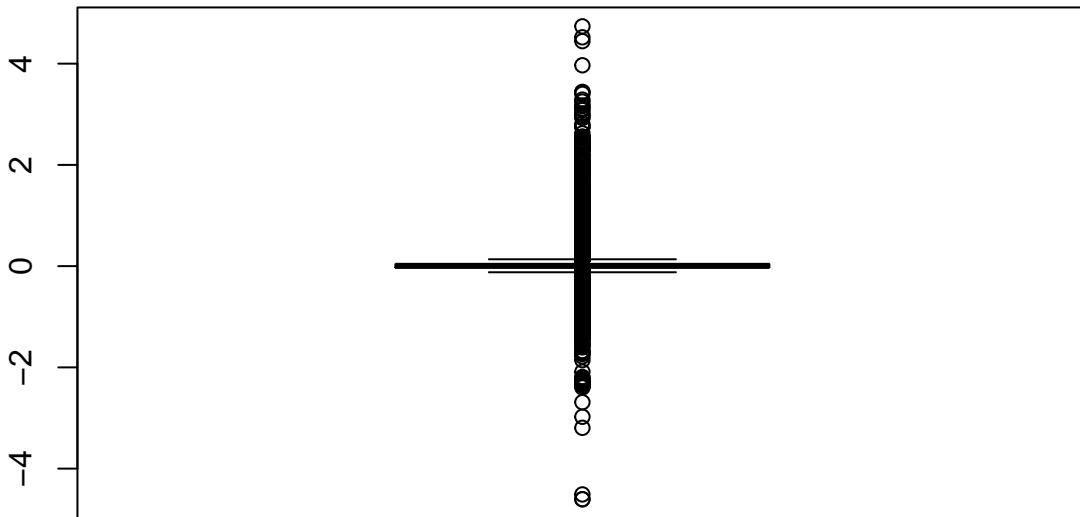
THESE TWO ARE TURNS OF TOWER AND THIS IS THE DISTRIBUTION OF TOWER

```
plot(density(train$logerror))
```

density.default(x = train\$logerror)



```
boxplot(train$logerror)
```

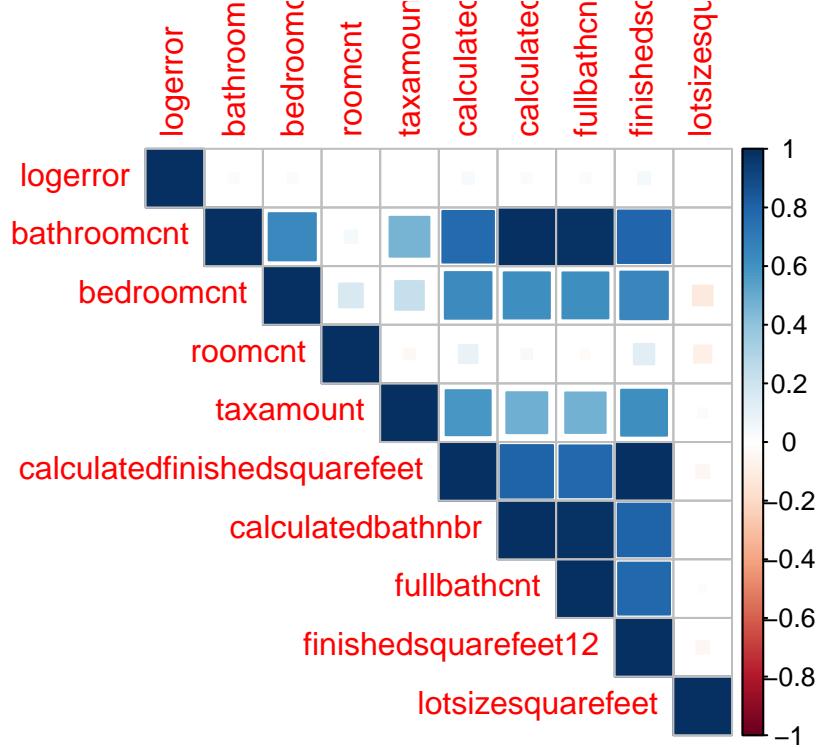


Just get a basic idea of the whole data set, we can see it is pretty centered at zero.

Then, one natural thing to do is to look at the correlation of different numeric features. #correlation -> use more in continuous variable

```
library(corrplot)
correlations <- cor(train[, c('logerror', 'bathroomcnt', 'bedroomcnt', 'roomcnt',
                           'taxamount', 'calculatedfinishedsquarefeet',
                           'calculatedbathnbr', 'fullbathcnt', 'finishedsquarefeet12', 'lotsizesquarefeet',
                           use = "pairwise.complete.obs")]

corrplot(correlations, method = "square", tl.cex = 1, type = 'upper')
```



We can see the value in coraltion matrix are very small, which only means there are not linear relationship, but there might be other relationships. We still cannot draw any conclusion too easily.

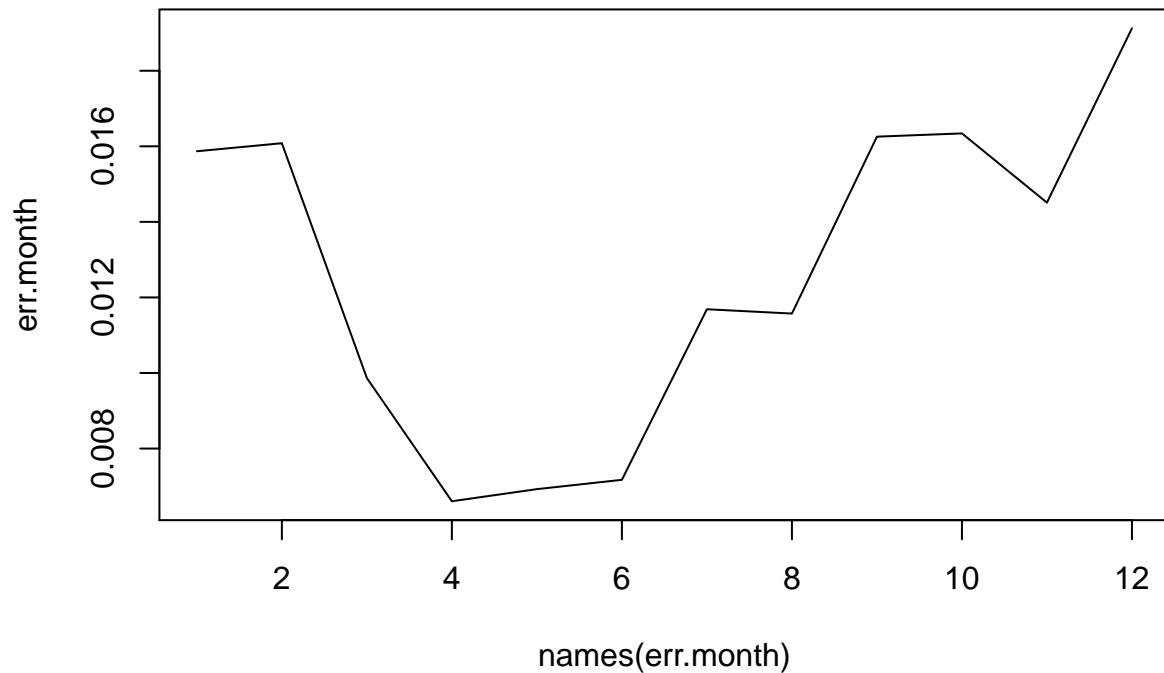
Then let us take a look at the relationships between different features and response variable.

1. transaction month

```
#extract transaction month from date data
train$txnmouth <- sapply(strsplit(train$transactiondate, '-'),
                         function(x) x[2])

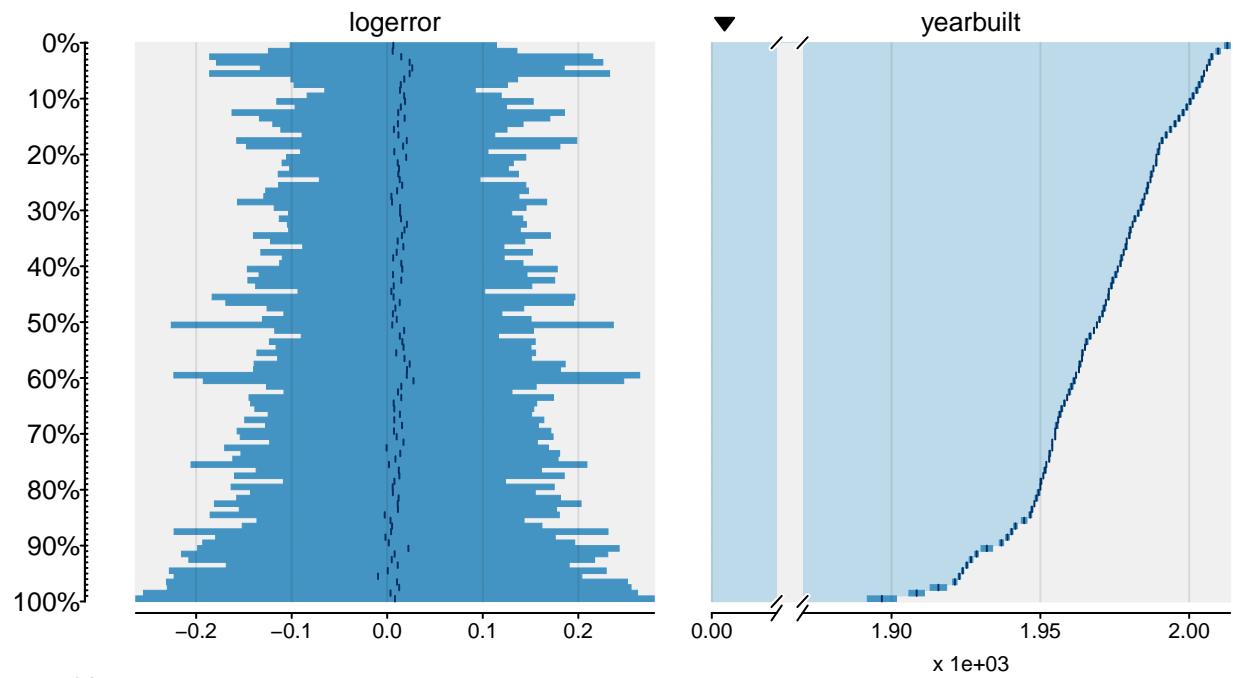
err.month <- by(train,train$txnmouth,function(x) mean(x$logerror))

plot(names(err.month),err.month,type = "l")
```

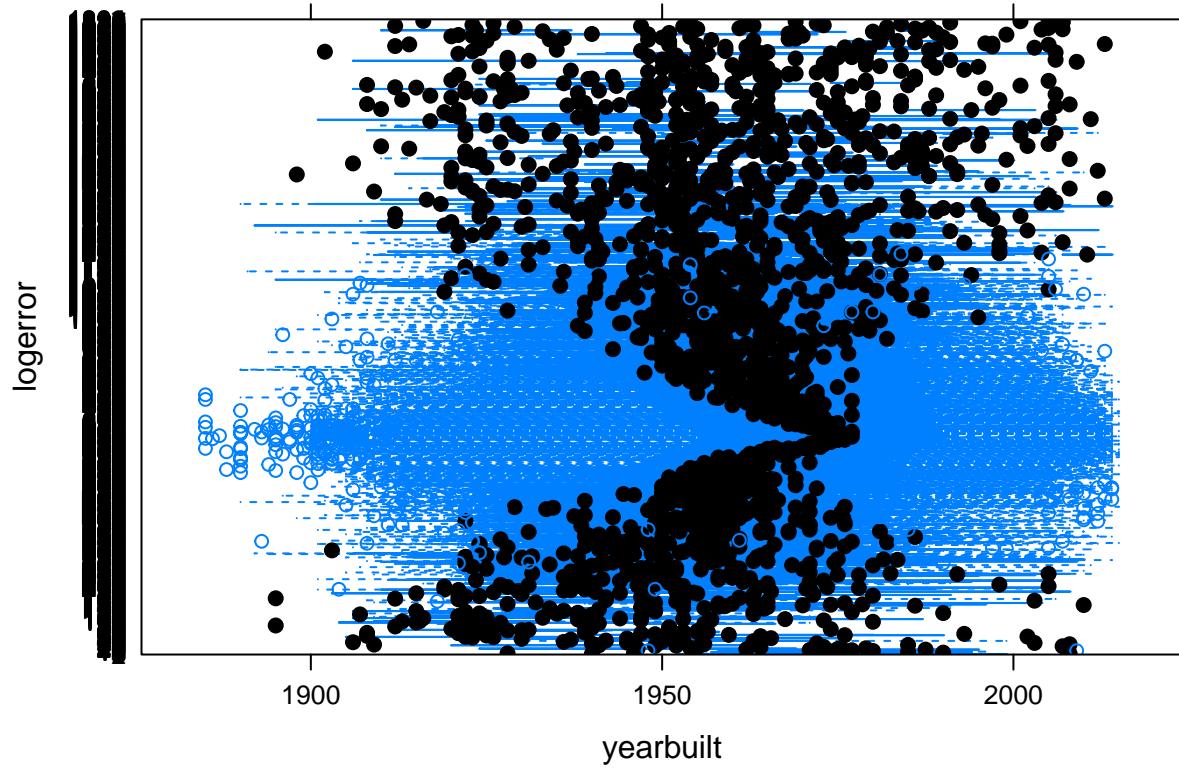


The mean value is relatively low from March to August, there are definitely information here which might help us predict logerror in the future. ##2. built year

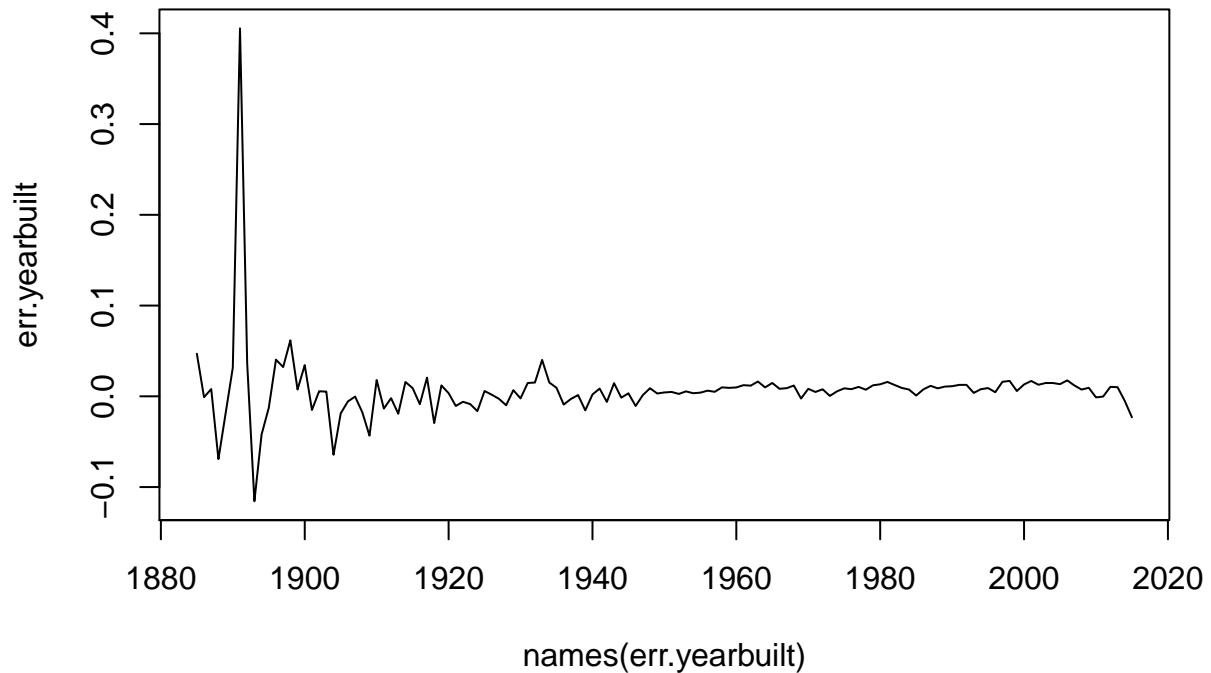
```
tableplot(train, select = c('logerror', 'yearbuilt'), sortCol = "yearbuilt")
```



```
bwplot(logerror ~ yearbuilt, data = train)
```



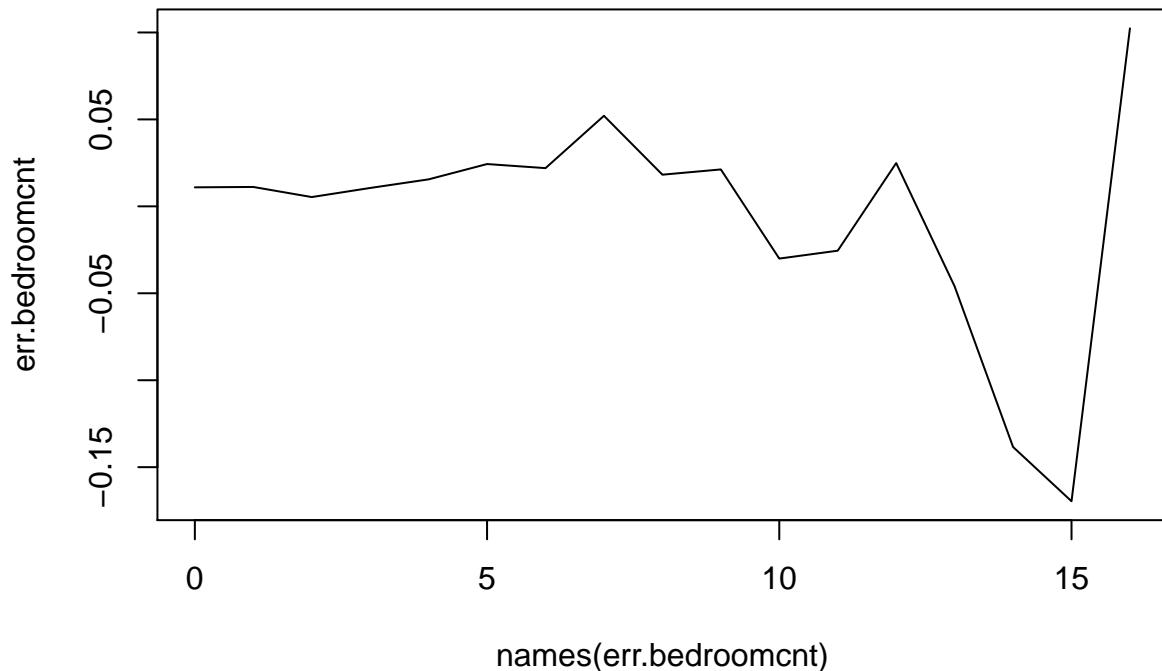
```
err.yearbuilt <- by(subset(train,train$logerror<0.9), subset(train,train$logerror<0.9)$yearbuil, function(x){  
  plot(names(x), x, type = "l")  
})
```



Built year does not seem to influence the response variable.

3. bedroom

```
err.bedroomcnt <- by(train, train$bedroomcnt, function(x) mean(x$logerror))
plot(names(err.bedroomcnt), err.bedroomcnt, type = "l")
```



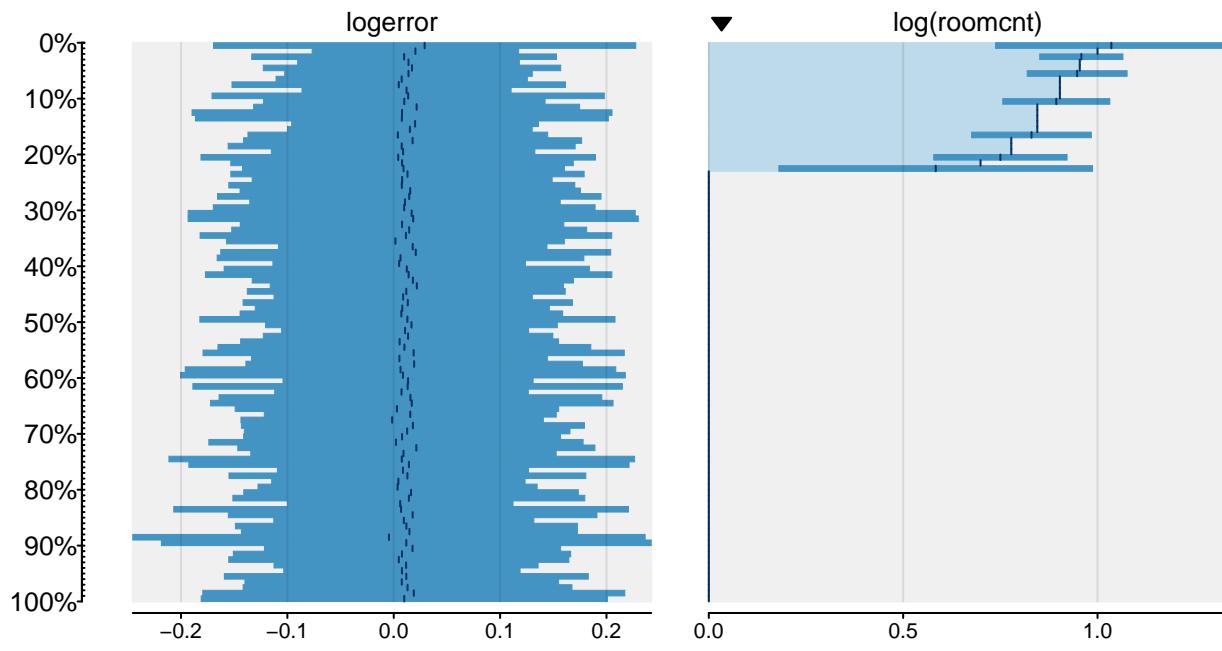
It seems like it declines a little bit when the bedroom number gets larger. ##4. bathroom

```
err.bathroomcnt <- by(train, train$bathroomcnt, function(x) mean(x$logerror))
plot(names(err.bathroomcnt), err.bathroomcnt, type = "l")
```



The trend is pretty similar with that of the bathroom. ##5. roomcnt

```
tableplot(train, select = c('logerror', 'roomcnt'), sortCol = "roomcnt")
```



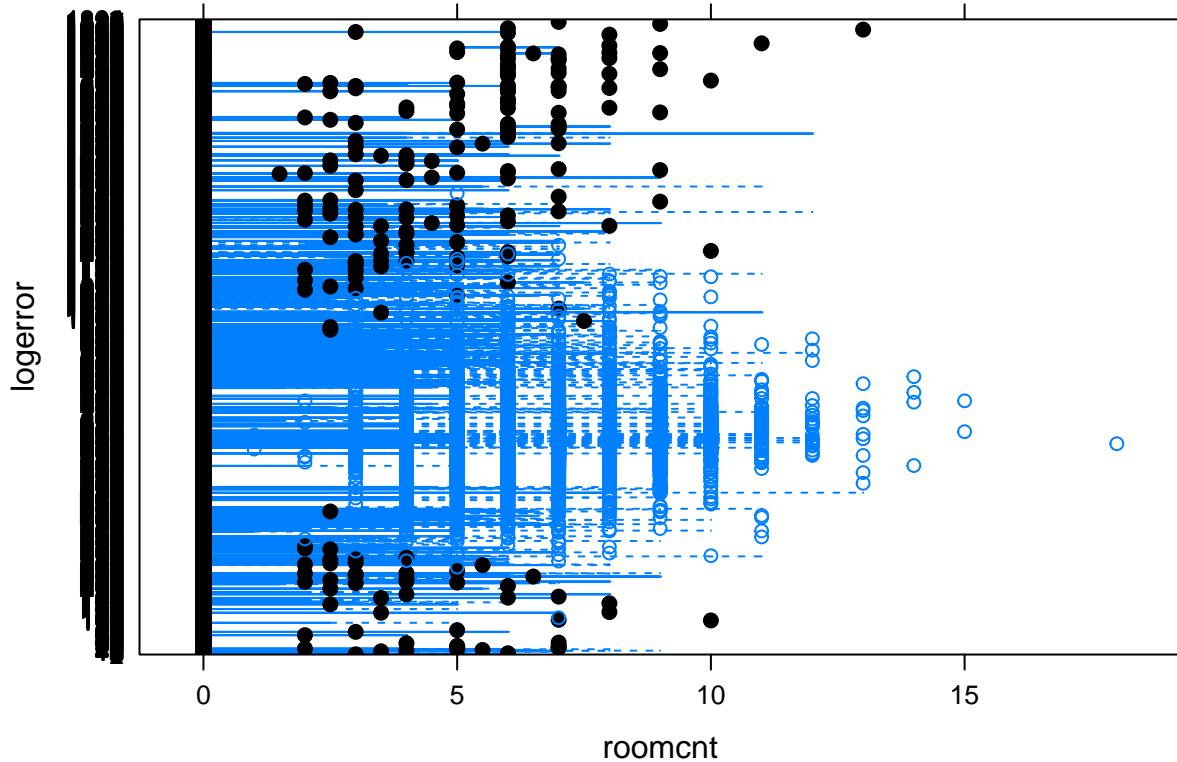
row bins: 100

objects:

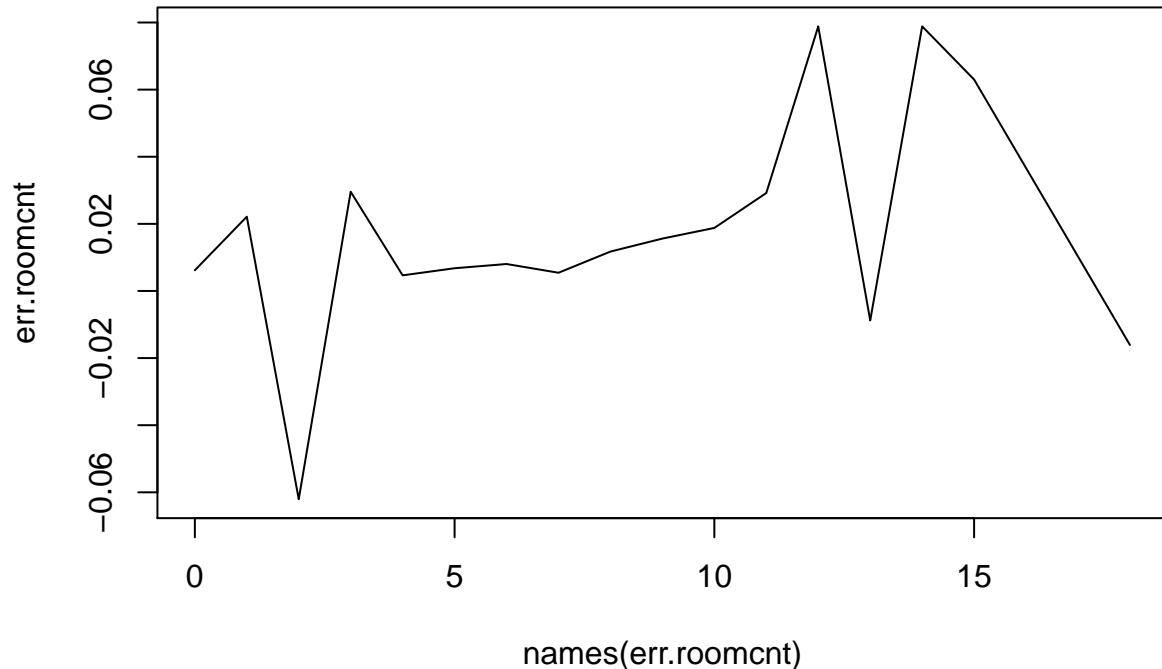
90,275

903 (per bin)

```
bwplot(logerror ~ roomcnt, data = train)
```



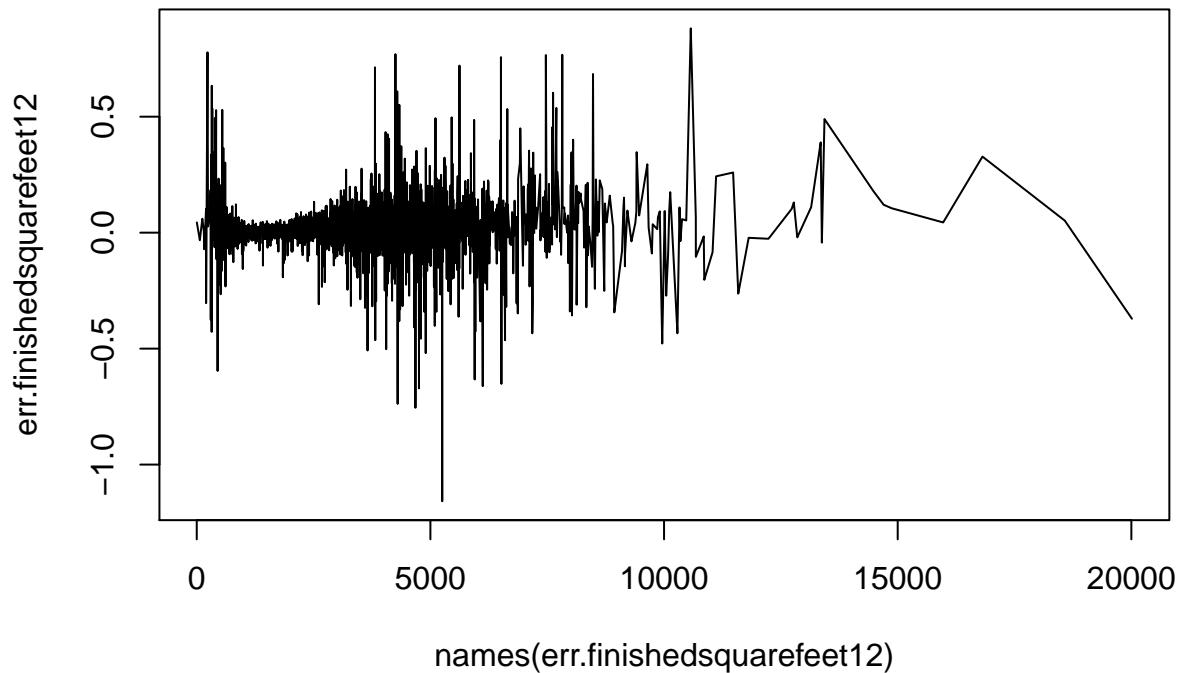
```
err.roomcnt <- by(subset(train,train$logerror<0.9),
                   subset(train,train$logerror<0.9)$roomcnt,
                   function(x) mean(x$logerror))
plot(names(err.roomcnt), err.roomcnt,type = "l")
```



This feature influences the mean logerror as well, we can see point 2 minimizes the mean logerror.
##6.finishedsquarefeet12

```
err.finishedsquarefeet12 <- by(subset(train,train$logerror<0.9),
                                subset(train,train$logerror<0.9)$finishedsquarefeet12,
                                function(x) mean(x$logerror))

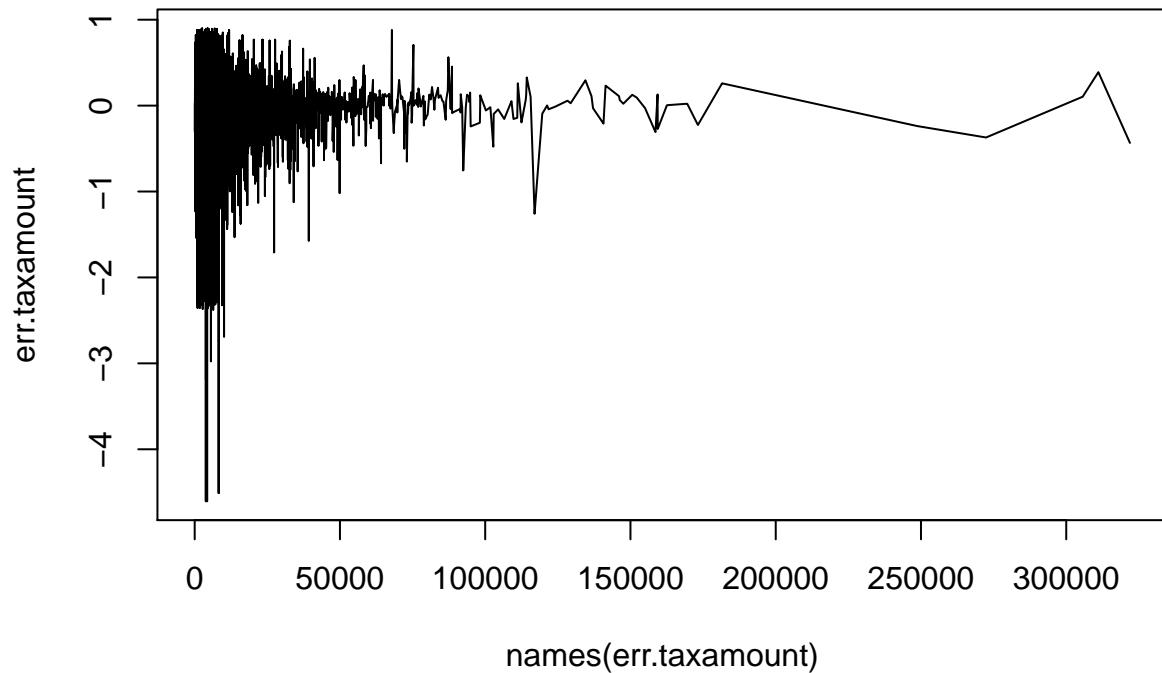
plot(names(err.finishedsquarefeet12), err.finishedsquarefeet12,type = "l")
```



Look like a signal, however, it can just be noise. I don't think this can be a good predictor.

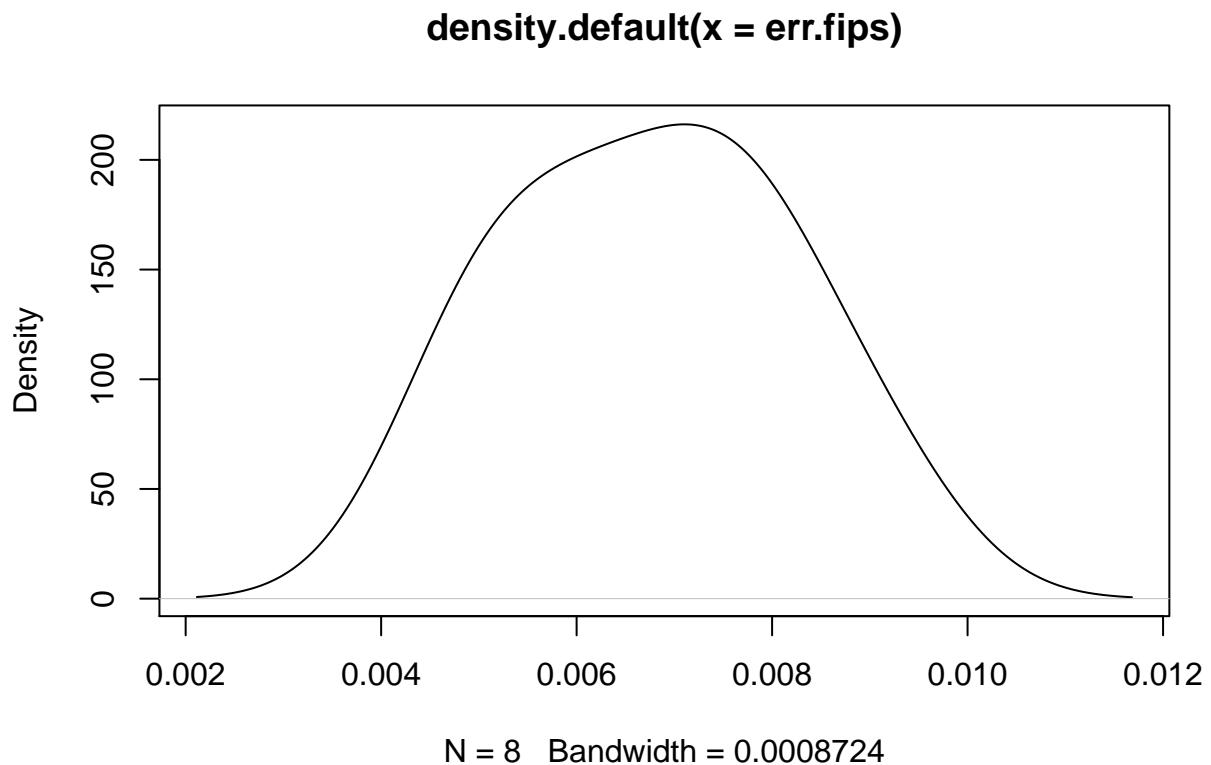
7. taxamont

```
err.taxamount <- by(subset(train,train$logerror<0.9),
                     subset(train,train$logerror<0.9)$taxamount,
                     function(x) mean(x$logerror))
plot(names(err.taxamount), err.taxamount, type = "l")
```



The same conclusion as the previous one. ##8.fips

```
err.fips <- by(subset(train,train$logerror<0.9),
                 subset(train,train$logerror<0.9)$fips,
                 function(x) mean(x$logerror))
plot(density(err.fips))
```

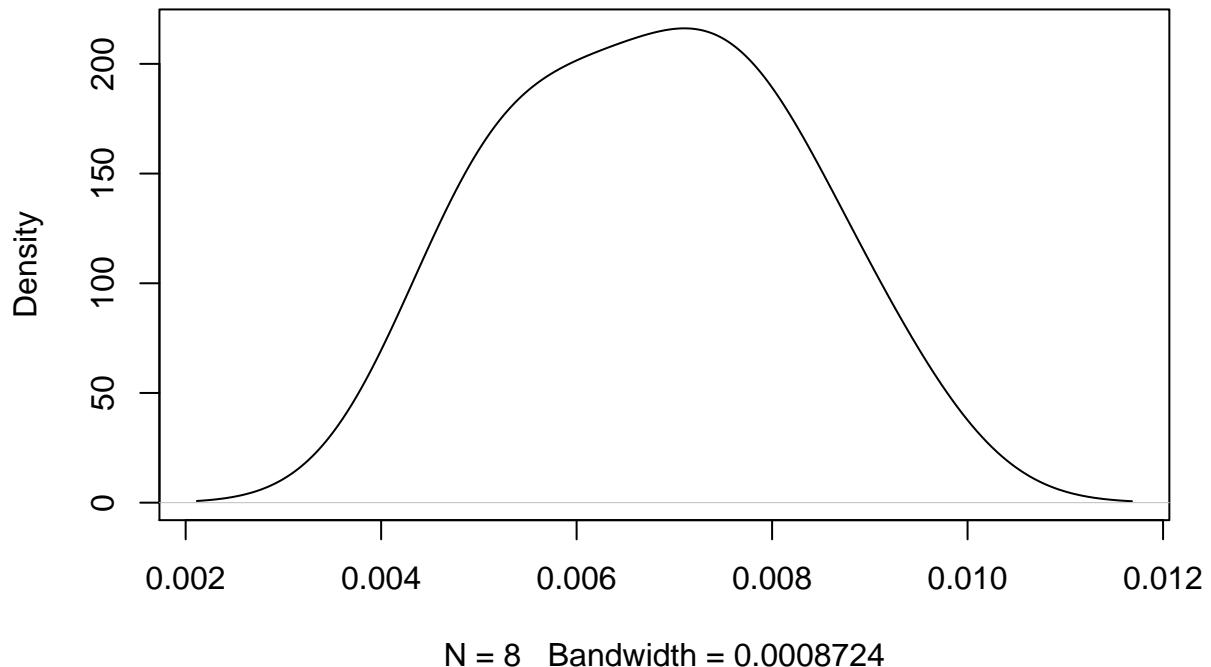


We can see the density is like a normal distribution. So fips does influence logerror.

9.regionidzip

```
err.regionidzip <- by(subset(train,train$logerror<0.9),
                      subset(train,train$logerror<0.9)$regionidzip,
                      function(x) mean(x$logerror))
plot(density(err.regionidzip))
```

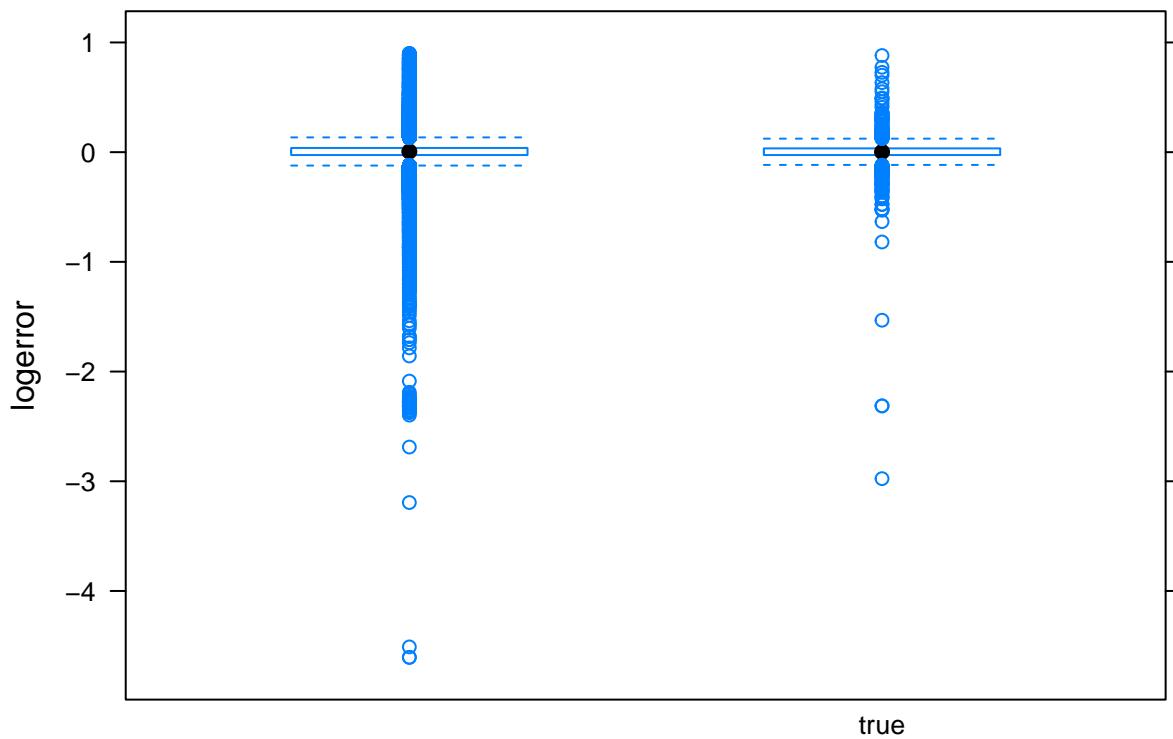
density.default(x = err.regionidzip)



The same as the previous one, they are all considered as region factors.

10.has hot tub or spa

```
bwplot(logerror~hashottuborspa, data = subset(train,train$logerror < 0.9))
```



I thought this one might influence the result, however, it seems like it does not.

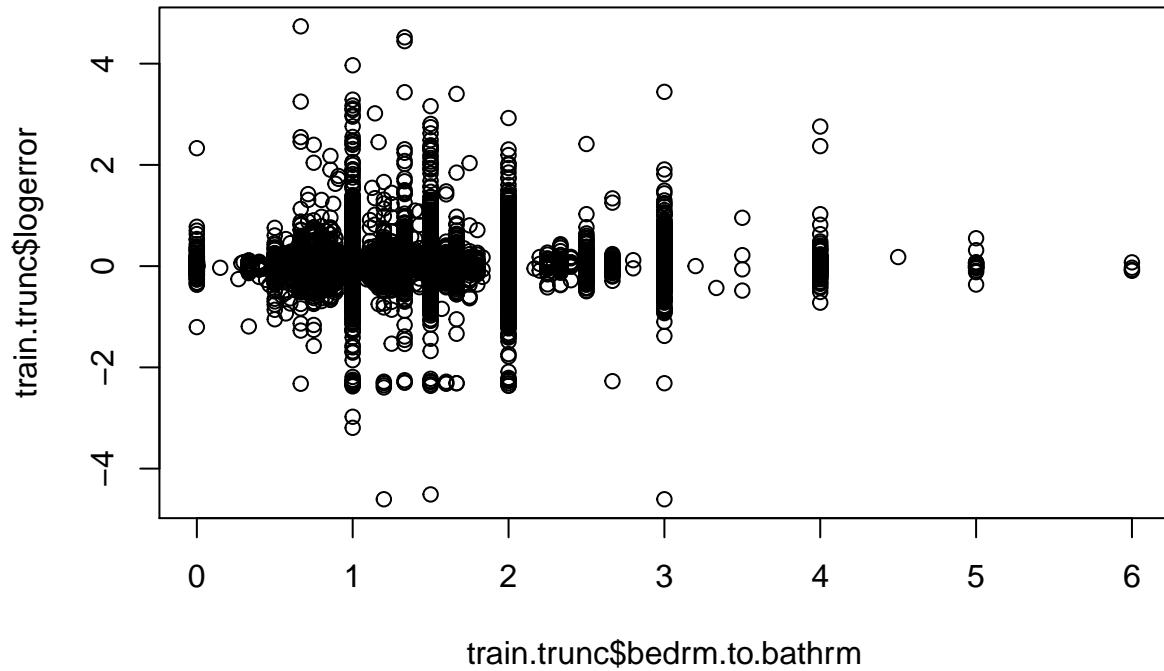
Except those feature, we create some new features.

new features

1. bedroom to bathroom count

```
train$bedrm.to.bathrm <- train$bedroomcnt/train$bathroomcnt

train.trunc <- subset(train,!is.na(train$bedrm.to.bathrm))
plot(train.trunc$bedrm.to.bathrm, train.trunc$logerror)
```

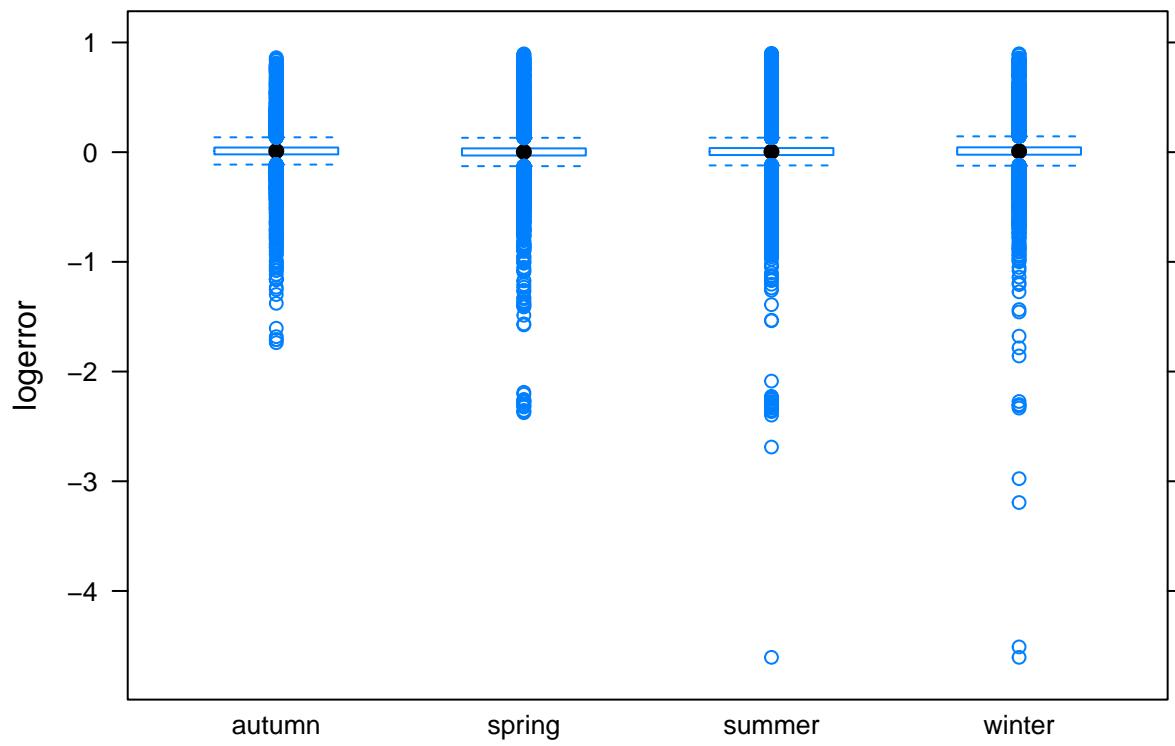


We can see that When the ratio is low, it seems logerror is bigger.

2. seasonality

```
season <- function(num){
  num <- as.integer(num)
  ifelse(num %in% c(1,2,12),"winter",
        ifelse(num %in% c(3,4,5),"spring",
              ifelse(num %in% c(6,7,8),"summer","autumn")))
}

train$season <- sapply(train$txnmonth, season)
bwplot(logerror~season,data=subset(train,train$logerror < 0.9))
```



It seems like it does not differ too much from season to season

Next steps:

keep exploring what other features can be and take a deeper look at current features.