#library(lubridate)  
#library(lazyeval)  
install.packages('plotly', repos = 'http://cran.us.r-project.org')

## Installing package into 'C:/Users/Geeta/Documents/R/win-library/3.3'  
## (as 'lib' is unspecified)

## package 'plotly' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\Geeta\AppData\Local\Temp\RtmpumKjJk\downloaded\_packages

library(ggplot2)  
library(GGally)

checkNaFunction <- function(houseData){  
naColumns <- c()  
#checking NA for each columns  
for(i in 1:ncol(houseData)) {  
 #cat(sprintf("Checking NA: %s \n", colnames(houseData)[i]))  
 if(length(which(is.na(houseData[,i]))) > 0){  
 #cat(sprintf("There is NA: %s \n" , colnames(houseData)[i]))  
 naColumns <- c(naColumns, colnames(houseData)[i])  
 }  
}  
return(naColumns)  
}

bucketByColumn <- function(houseData,i){  
minP <- min(as.numeric(houseData[,i]))  
maxP <- max(as.numeric(houseData[,i]))  
rangeP <- range(as.numeric(houseData[,i]))  
rangeP  
cat(sprintf("Min-Max value for: %s , MAX: %d, MIN: %d \n", colnames(houseData)[i], minP, maxP))  
}

analysis <- function(houseData, i, labels, plotLog){  
 plot(houseData$price~houseData[,i],main = labels[1],xlab = labels[2],ylab = labels[3], col=(c("gold","darkgreen")))  
   
 plot(houseData[,i],log(houseData$price), main=cat(labels[1]), xlab=cat(labels[2]), ylab=cat("Log of ",labels[3]), col=(c("gold","darkgreen")))  
   
 if(plotLog=='Y'){  
 plot(log(houseData[,i]),log(houseData$price), main=cat("Log ",labels[1]), xlab=cat("Log of ",labels[2]), ylab=cat("Log of ",labels[3]), col=(c("gold","darkgreen")))  
 }  
   
 hist(houseData[,i],main = labels[1],xlab = labels[2],ylab = labels[3],col=(c("gold","darkgreen")))  
  
 boxplot(houseData[,i],main = labels[1],col=(c("gold","darkgreen")))  
   
 cor(houseData[,i],houseData$price)  
}

# Data Importing And Cleaning

houseData <- read.csv("file:///G:/Ryerson-BigData/capstone-R/CKME-136/data/kc\_house\_data.csv")  
head(houseData)

## id date price bedrooms bathrooms sqft\_living  
## 1 7129300520 20141013T000000 221900 3 1.00 1180  
## 2 6414100192 20141209T000000 538000 3 2.25 2570  
## 3 5631500400 20150225T000000 180000 2 1.00 770  
## 4 2487200875 20141209T000000 604000 4 3.00 1960  
## 5 1954400510 20150218T000000 510000 3 2.00 1680  
## 6 7237550310 20140512T000000 1225000 4 4.50 5420  
## sqft\_lot floors waterfront view condition grade sqft\_above sqft\_basement  
## 1 5650 1 0 0 3 7 1180 0  
## 2 7242 2 0 0 3 7 2170 400  
## 3 10000 1 0 0 3 6 770 0  
## 4 5000 1 0 0 5 7 1050 910  
## 5 8080 1 0 0 3 8 1680 0  
## 6 101930 1 0 0 3 11 3890 1530  
## yr\_built yr\_renovated zipcode lat long sqft\_living15 sqft\_lot15  
## 1 1955 0 98178 47.5112 -122.257 1340 5650  
## 2 1951 1991 98125 47.7210 -122.319 1690 7639  
## 3 1933 0 98028 47.7379 -122.233 2720 8062  
## 4 1965 0 98136 47.5208 -122.393 1360 5000  
## 5 1987 0 98074 47.6168 -122.045 1800 7503  
## 6 2001 0 98053 47.6561 -122.005 4760 101930

colnames(houseData)

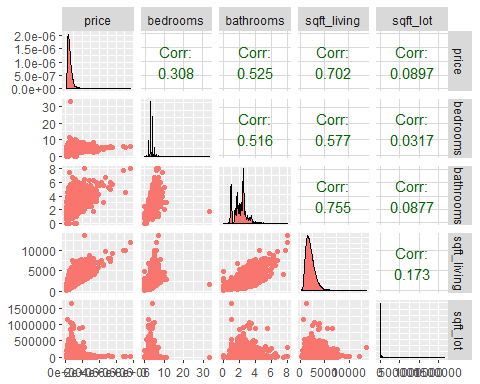
## [1] "id" "date" "price" "bedrooms"   
## [5] "bathrooms" "sqft\_living" "sqft\_lot" "floors"   
## [9] "waterfront" "view" "condition" "grade"   
## [13] "sqft\_above" "sqft\_basement" "yr\_built" "yr\_renovated"   
## [17] "zipcode" "lat" "long" "sqft\_living15"  
## [21] "sqft\_lot15"

naColumns <- checkNaFunction(houseData)  
if(length(naColumns)>0){  
 cat("Found NA Colums:")  
 for(i in 1:length(naColumns)) {  
 cat(sprintf("%s,", colnames(houseData)[i]))  
 }  
}  
  
#houseData$date<-(substr(houseData$date, 1, 8))  
#houseData$date<- ymd(houseData$date)  
#houseData$date<-as.numeric(as.Date(houseData$date, origin = "1900-01-01"))  
  
# Here we conclude that this data does not hold any column with NA.  
bucketByColumn(houseData,3)

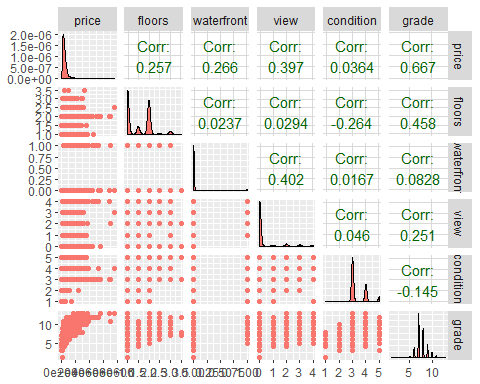
## Min-Max value for: price , MAX: 75000, MIN: 7700000

# Price with other attributes

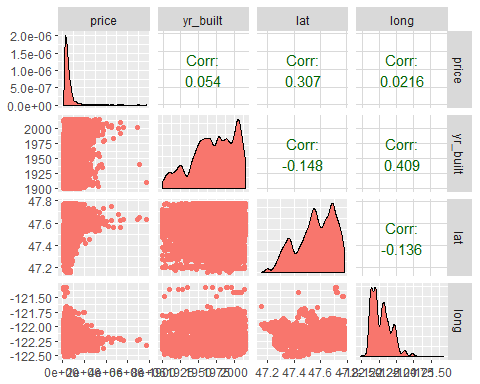
## verify the relationship between price, bedrooms, bathrooms, sqft\_living and sqft lot  
plot1 <- ggpairs(data=houseData, columns=3:7, mapping = aes(color = "dark green"), axisLabels="show")  
plot1



## verify the relationship between price, floors, waterfront, view, condition and grade  
plot2 <- ggpairs(data=houseData, columns=c(3,8:12),  
 mapping = aes(color = "dark green"),  
 axisLabels="show")  
plot2



## verify the relationship between price, yr built, lat and long  
plot3 <- ggpairs(data=houseData, columns=c(3,15,18,19),  
 mapping = aes(color = "dark green"),  
 axisLabels="show")  
plot3



## Correlation among all the variables

#Only sqft\_living & sqft\_above,sqft\_living & grade,sqft\_living & bathrooms have good correlation between them  
#Remove the columns which does not hold any significance in predicing house price  
houseData$date <- NULL  
houseData$id <- NULL  
cor(houseData)

## price bedrooms bathrooms sqft\_living  
## price 1.00000000 0.308349598 0.52513751 0.70203505  
## bedrooms 0.30834960 1.000000000 0.51588364 0.57667069  
## bathrooms 0.52513751 0.515883638 1.00000000 0.75466528  
## sqft\_living 0.70203505 0.576670693 0.75466528 1.00000000  
## sqft\_lot 0.08966086 0.031703243 0.08773966 0.17282566  
## floors 0.25679389 0.175428935 0.50065317 0.35394929  
## waterfront 0.26636943 -0.006582479 0.06374363 0.10381782  
## view 0.39729349 0.079531852 0.18773702 0.28461119  
## condition 0.03636179 0.028472104 -0.12498193 -0.05875259  
## grade 0.66743426 0.356966725 0.66498253 0.76270448  
## sqft\_above 0.60556730 0.477600161 0.68534248 0.87659660  
## sqft\_basement 0.32381602 0.303093375 0.28377003 0.43504297  
## yr\_built 0.05401153 0.154178069 0.50601944 0.31804877  
## yr\_renovated 0.12643379 0.018840823 0.05073898 0.05536293  
## zipcode -0.05320285 -0.152668487 -0.20386627 -0.19943004  
## lat 0.30700348 -0.008931010 0.02457295 0.05252946  
## long 0.02162624 0.129472975 0.22304184 0.24022330  
## sqft\_living15 0.58537890 0.391637524 0.56863429 0.75642026  
## sqft\_lot15 0.08244715 0.029244224 0.08717536 0.18328555  
## sqft\_lot floors waterfront view  
## price 0.089660861 0.256793888 0.266369434 0.397293488  
## bedrooms 0.031703243 0.175428935 -0.006582479 0.079531852  
## bathrooms 0.087739662 0.500653173 0.063743629 0.187737024  
## sqft\_living 0.172825661 0.353949290 0.103817818 0.284611186  
## sqft\_lot 1.000000000 -0.005200991 0.021603683 0.074710106  
## floors -0.005200991 1.000000000 0.023698320 0.029443820  
## waterfront 0.021603683 0.023698320 1.000000000 0.401857351  
## view 0.074710106 0.029443820 0.401857351 1.000000000  
## condition -0.008958250 -0.263767946 0.016653157 0.045989737  
## grade 0.113621124 0.458182514 0.082774914 0.251320585  
## sqft\_above 0.183512281 0.523884710 0.072074592 0.167649344  
## sqft\_basement 0.015286202 -0.245704542 0.080587939 0.276946579  
## yr\_built 0.053080367 0.489319425 -0.026161086 -0.053439851  
## yr\_renovated 0.007643505 0.006338401 0.092884837 0.103917288  
## zipcode -0.129574486 -0.059120642 0.030284728 0.084826917  
## lat -0.085682788 0.049614131 -0.014273776 0.006156732  
## long 0.229520859 0.125419028 -0.041910200 -0.078399712  
## sqft\_living15 0.144608174 0.279885265 0.086463136 0.280439082  
## sqft\_lot15 0.718556752 -0.011269187 0.030703283 0.072574568  
## condition grade sqft\_above sqft\_basement  
## price 0.036361789 0.66743426 0.6055672984 0.32381602  
## bedrooms 0.028472104 0.35696673 0.4776001614 0.30309338  
## bathrooms -0.124981933 0.66498253 0.6853424759 0.28377003  
## sqft\_living -0.058752587 0.76270448 0.8765965987 0.43504297  
## sqft\_lot -0.008958250 0.11362112 0.1835122809 0.01528620  
## floors -0.263767946 0.45818251 0.5238847103 -0.24570454  
## waterfront 0.016653157 0.08277491 0.0720745917 0.08058794  
## view 0.045989737 0.25132058 0.1676493441 0.27694658  
## condition 1.000000000 -0.14467367 -0.1582136164 0.17410491  
## grade -0.144673671 1.00000000 0.7559229376 0.16839182  
## sqft\_above -0.158213616 0.75592294 1.0000000000 -0.05194331  
## sqft\_basement 0.174104914 0.16839182 -0.0519433068 1.00000000  
## yr\_built -0.361416562 0.44696320 0.4238983517 -0.13312410  
## yr\_renovated -0.060617787 0.01441428 0.0232846879 0.07132290  
## zipcode 0.003025524 -0.18486209 -0.2611899765 0.07484461  
## lat -0.014941006 0.11408406 -0.0008164986 0.11053796  
## long -0.106500448 0.19837215 0.3438030175 -0.14476477  
## sqft\_living15 -0.092824268 0.71320209 0.7318702924 0.20035498  
## sqft\_lot15 -0.003405523 0.11924790 0.1940498619 0.01727618  
## yr\_built yr\_renovated zipcode lat  
## price 0.05401153 0.126433793 -0.053202854 0.3070034800  
## bedrooms 0.15417807 0.018840823 -0.152668487 -0.0089310097  
## bathrooms 0.50601944 0.050738978 -0.203866274 0.0245729528  
## sqft\_living 0.31804877 0.055362927 -0.199430043 0.0525294622  
## sqft\_lot 0.05308037 0.007643505 -0.129574486 -0.0856827882  
## floors 0.48931942 0.006338401 -0.059120642 0.0496141310  
## waterfront -0.02616109 0.092884837 0.030284728 -0.0142737756  
## view -0.05343985 0.103917288 0.084826917 0.0061567321  
## condition -0.36141656 -0.060617787 0.003025524 -0.0149410064  
## grade 0.44696320 0.014414281 -0.184862093 0.1140840571  
## sqft\_above 0.42389835 0.023284688 -0.261189977 -0.0008164986  
## sqft\_basement -0.13312410 0.071322902 0.074844608 0.1105379580  
## yr\_built 1.00000000 -0.224873518 -0.346869178 -0.1481224021  
## yr\_renovated -0.22487352 1.000000000 0.064357057 0.0293976092  
## zipcode -0.34686918 0.064357057 1.000000000 0.2670479500  
## lat -0.14812240 0.029397609 0.267047950 1.0000000000  
## long 0.40935620 -0.068372369 -0.564071606 -0.1355117836  
## sqft\_living15 0.32622890 -0.002672555 -0.279032997 0.0488579321  
## sqft\_lot15 0.07095793 0.007853765 -0.147221069 -0.0864188072  
## long sqft\_living15 sqft\_lot15  
## price 0.02162624 0.585378904 0.082447153  
## bedrooms 0.12947298 0.391637524 0.029244224  
## bathrooms 0.22304184 0.568634290 0.087175361  
## sqft\_living 0.24022330 0.756420259 0.183285551  
## sqft\_lot 0.22952086 0.144608174 0.718556752  
## floors 0.12541903 0.279885265 -0.011269187  
## waterfront -0.04191020 0.086463136 0.030703283  
## view -0.07839971 0.280439082 0.072574568  
## condition -0.10650045 -0.092824268 -0.003405523  
## grade 0.19837215 0.713202093 0.119247897  
## sqft\_above 0.34380302 0.731870292 0.194049862  
## sqft\_basement -0.14476477 0.200354983 0.017276181  
## yr\_built 0.40935620 0.326228900 0.070957926  
## yr\_renovated -0.06837237 -0.002672555 0.007853765  
## zipcode -0.56407161 -0.279032997 -0.147221069  
## lat -0.13551178 0.048857932 -0.086418807  
## long 1.00000000 0.334604984 0.254451288  
## sqft\_living15 0.33460498 1.000000000 0.183191749  
## sqft\_lot15 0.25445129 0.183191749 1.000000000

# we can see that zip-code has very weak co-orelation -0.053202854, so let us remove it  
houseData$zipcode <- NULL  
houseData <- read.csv("file:///G:/Ryerson-BigData/capstone-R/CKME-136/data/kc\_house\_data.csv")

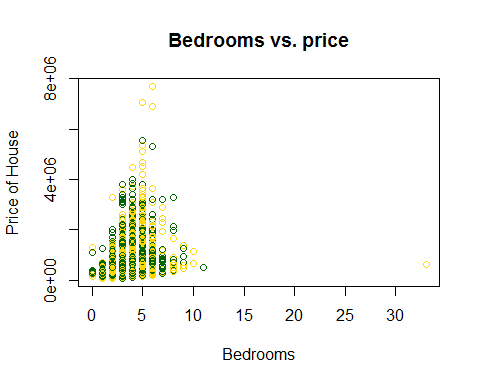
# Now Let us do analysis of price with all other variables

# Bedroom Vs Price analysis

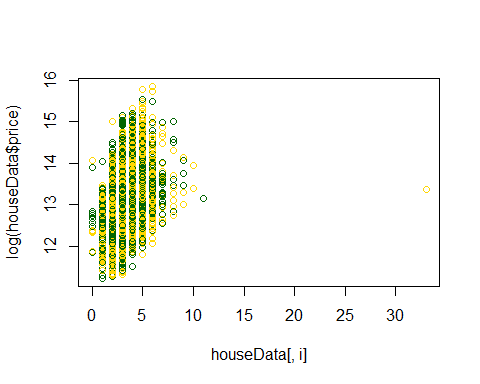
bucketByColumn(houseData,4)

## Min-Max value for: bedrooms , MAX: 0, MIN: 33

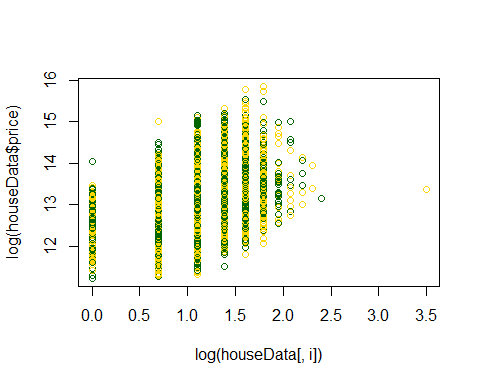
analysis(houseData,4,c('Bedrooms vs. price','Bedrooms', 'Price of House'), 'Y')



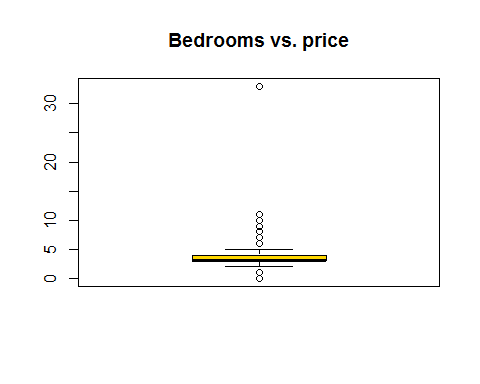
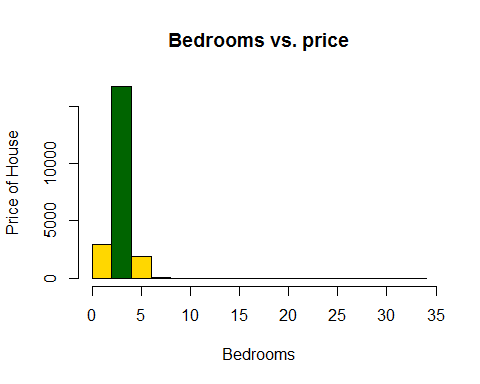
## BedroomsLog of Price of House



## Bedrooms vs. priceLog of BedroomsLog of Price of House

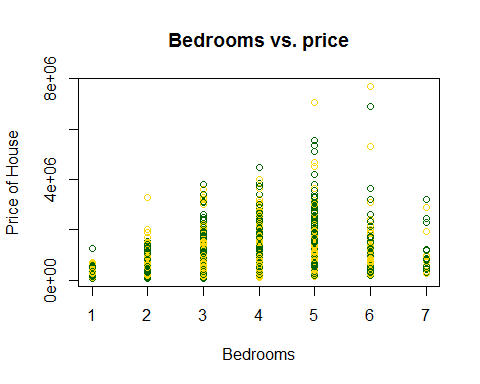


## Log Bedrooms vs. price

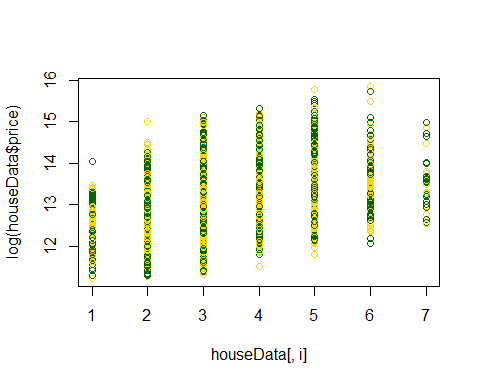


## [1] 0.3083496

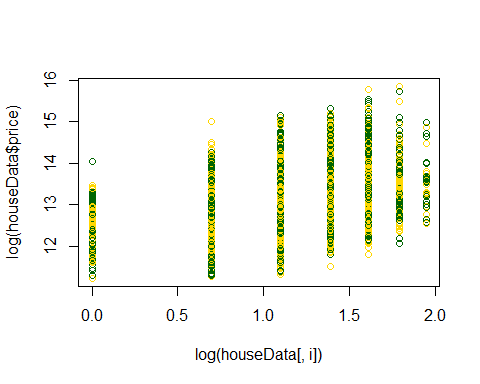
#\*\*\*\*\*\*\*Removing the outliers  
#Since more than 7 bedrooms are very rare.Also it's the outlier for my model.  
#I have removed the outlier data.  
houseData<-subset(houseData,bedrooms>=1 & bedrooms<=7)  
#\*\*\*\*\*\*\*Once we removed the outliers, again get the analysis  
analysis(houseData,4,c('Bedrooms vs. price','Bedrooms', 'Price of House'), 'Y')



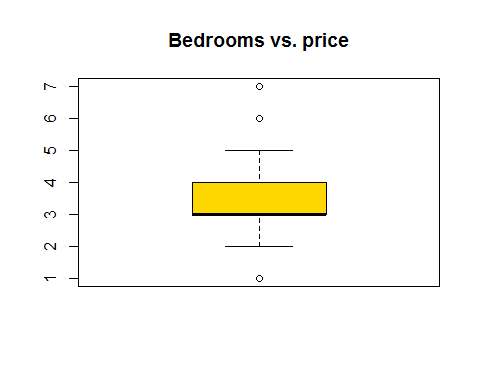
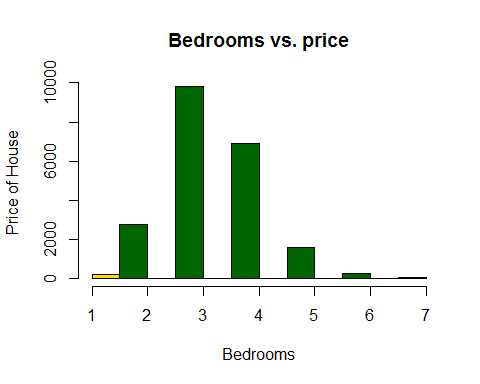
## BedroomsLog of Price of House



## Bedrooms vs. priceLog of BedroomsLog of Price of House



## Log Bedrooms vs. price



## [1] 0.3156734

bucketByColumn(houseData,4)

## Min-Max value for: bedrooms , MAX: 1, MIN: 7

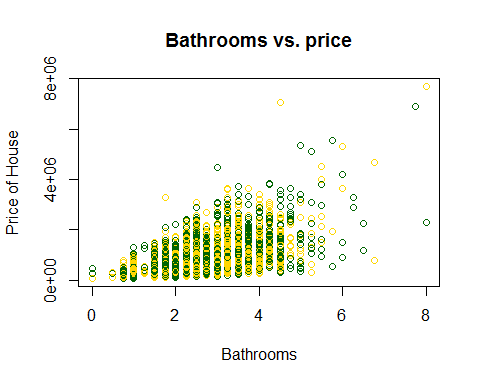
## here we found that log of bedroom give better performance.

# Bathroom Vs Price analysis

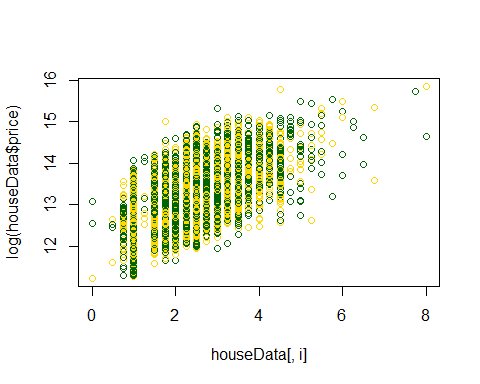
bucketByColumn(houseData,5)

## Min-Max value for: bathrooms , MAX: 0, MIN: 8

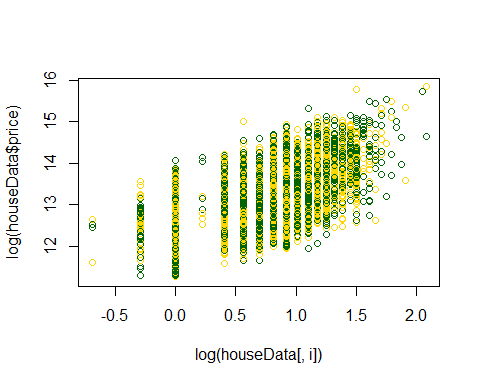
analysis(houseData,5,c('Bathrooms vs. price','Bathrooms', 'Price of House'), 'Y')



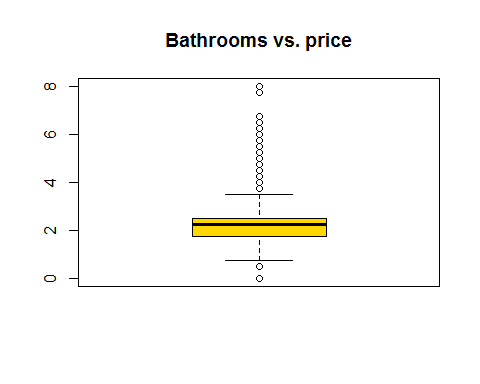
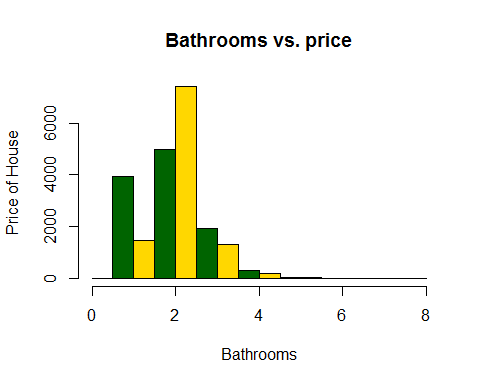
## BathroomsLog of Price of House



## Bathrooms vs. priceLog of BathroomsLog of Price of House

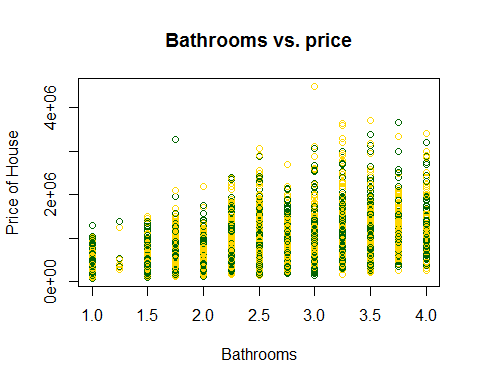


## Log Bathrooms vs. price

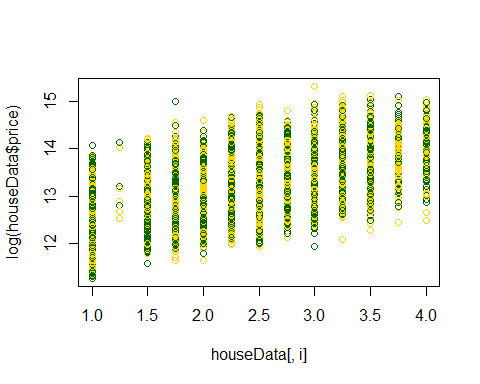


## [1] 0.5259342

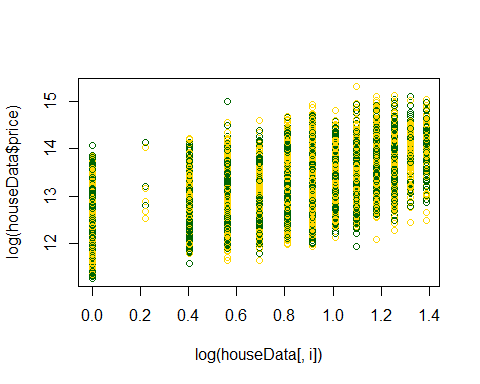
## Price vs. Bathrooms, here we can find good correlation, as number of bahtrooms increases, price increases as well, with one expection in when bathroom=7  
#\*\*\*\*\*\*\*Removing the outliers  
#More than 4 bathrooms are very rare in this data.So I am removing it.  
houseData<-subset(houseData,bathrooms>=1 & bathrooms<=4)  
analysis(houseData,5,c('Bathrooms vs. price','Bathrooms', 'Price of House'), 'Y')



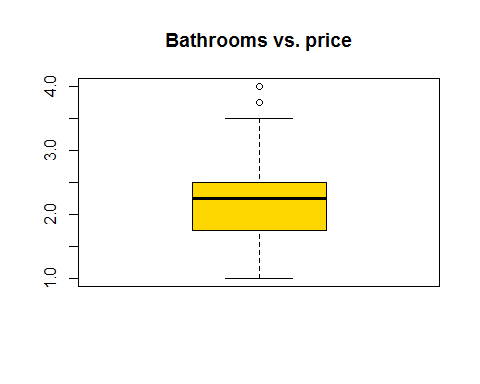
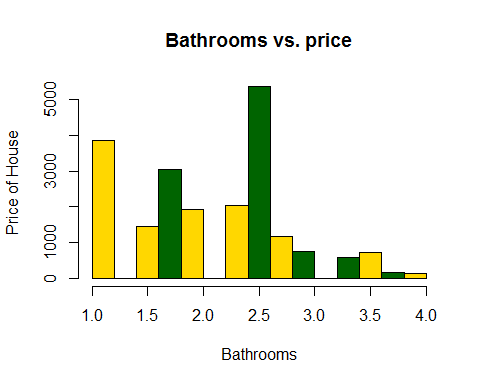
## BathroomsLog of Price of House



## Bathrooms vs. priceLog of BathroomsLog of Price of House



## Log Bathrooms vs. price



## [1] 0.475159

bucketByColumn(houseData,5)

## Min-Max value for: bathrooms , MAX: 1, MIN: 4

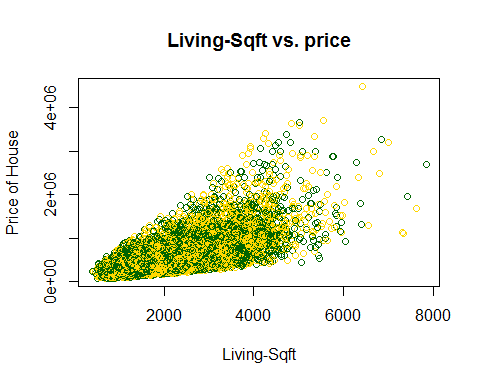
## here we found that log of bathrooms give better performance.

# SQFT Living Vs Price analysis

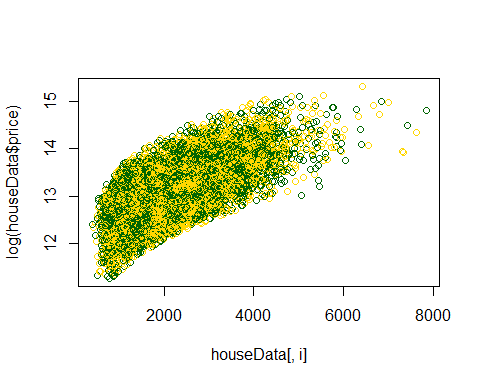
bucketByColumn(houseData,6)

## Min-Max value for: sqft\_living , MAX: 390, MIN: 7850

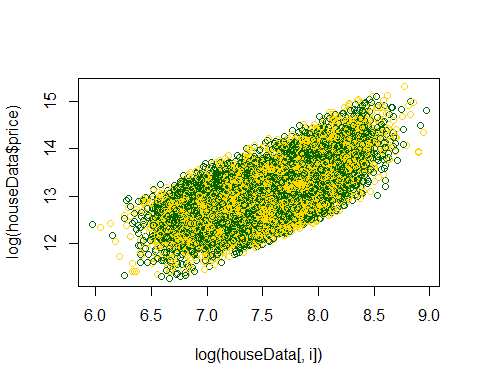
analysis(houseData,6,c('Living-Sqft vs. price','Living-Sqft', 'Price of House'), 'Y')



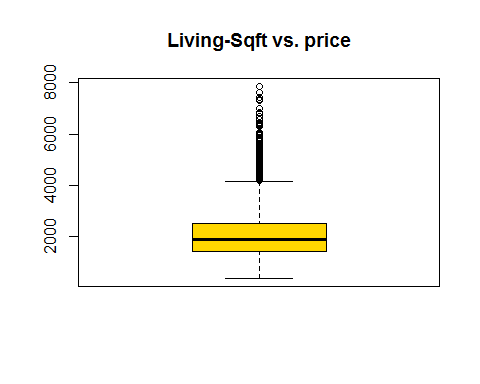
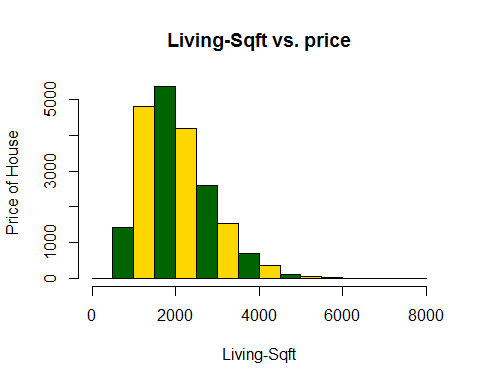
## Living-SqftLog of Price of House



## Living-Sqft vs. priceLog of Living-SqftLog of Price of House

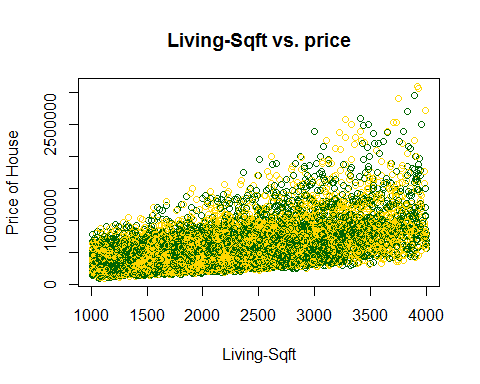


## Log Living-Sqft vs. price

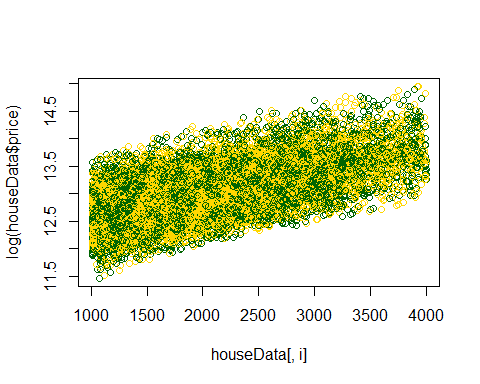


## [1] 0.6701029

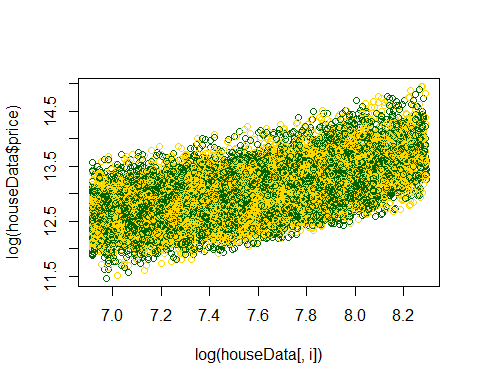
## Price vs. Sqft\_living ->> Nice correlation, as sqft increases, price increases as well.  
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,sqft\_living >1000 & sqft\_living<=4000)  
analysis(houseData,6,c('Living-Sqft vs. price','Living-Sqft', 'Price of House'), 'Y')



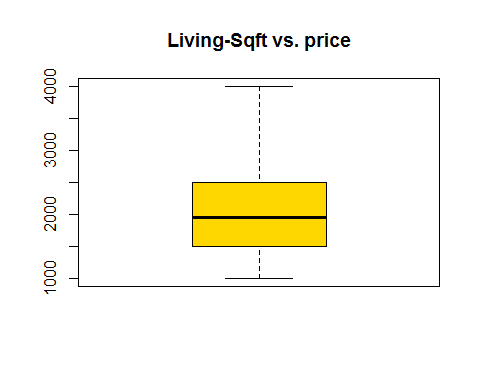
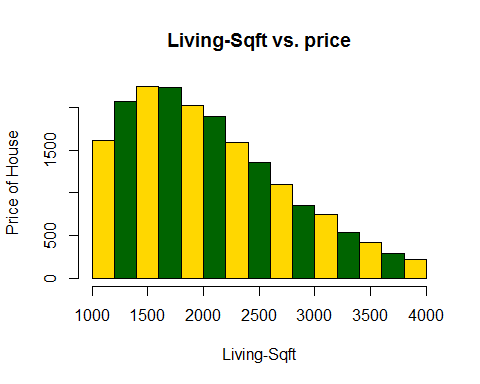
## Living-SqftLog of Price of House



## Living-Sqft vs. priceLog of Living-SqftLog of Price of House



## Log Living-Sqft vs. price



## [1] 0.5938015

bucketByColumn(houseData,6)

## Min-Max value for: sqft\_living , MAX: 1008, MIN: 4000

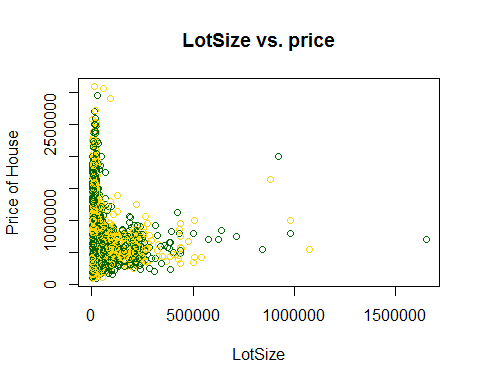
## Price vs. Sqft\_living ->> Nice correlation, as sqft increases, price increases as well.

## SQFT\_LOT Vs Price analysis

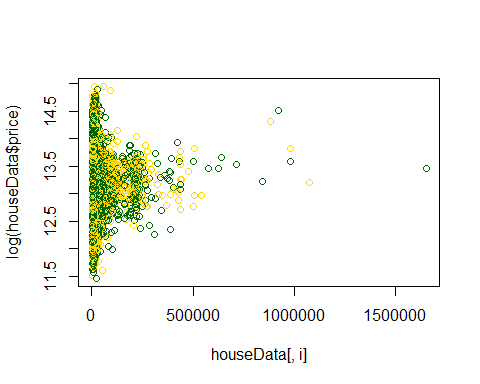
bucketByColumn(houseData,7)

## Min-Max value for: sqft\_lot , MAX: 520, MIN: 1651359

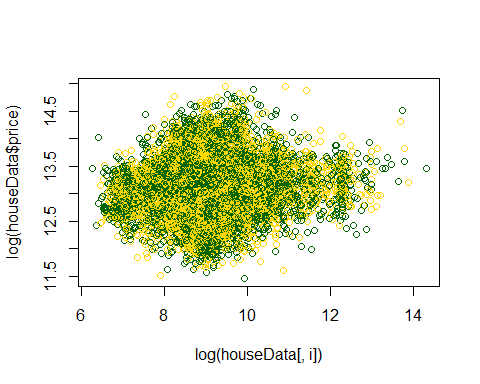
analysis(houseData,7,c('LotSize vs. price','LotSize', 'Price of House'), 'Y')



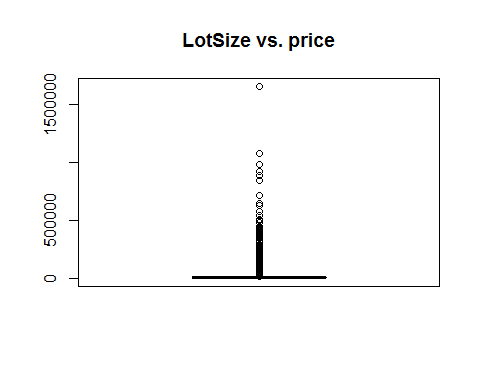
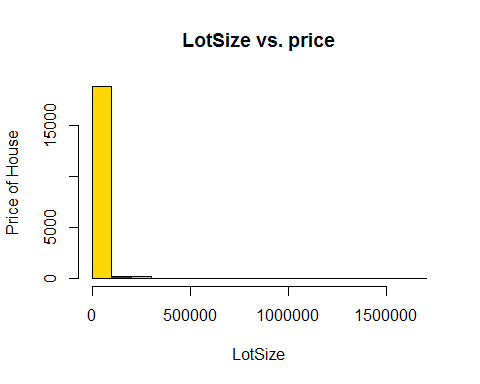
## LotSizeLog of Price of House



## LotSize vs. priceLog of LotSizeLog of Price of House

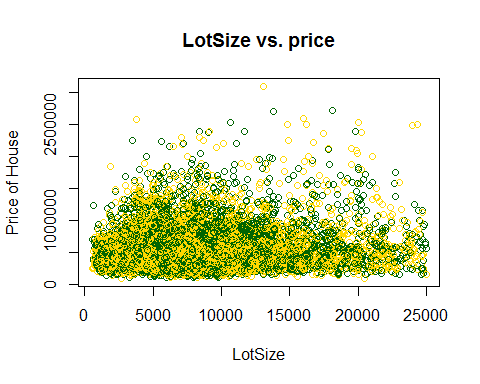


## Log LotSize vs. price

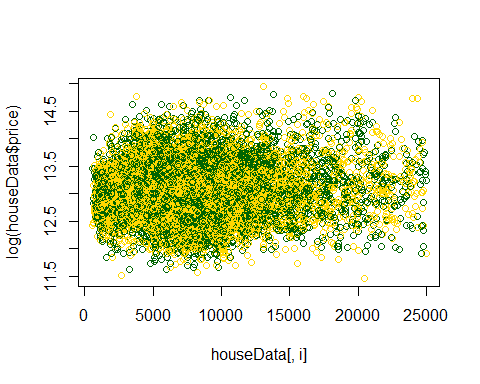


## [1] 0.06105332

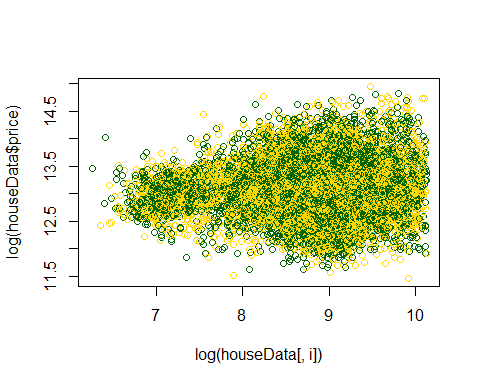
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,sqft\_lot>=0 & sqft\_lot<=25000)  
analysis(houseData,7,c('LotSize vs. price','LotSize', 'Price of House'), 'Y')



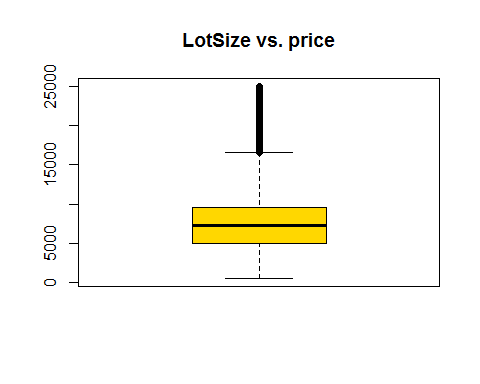
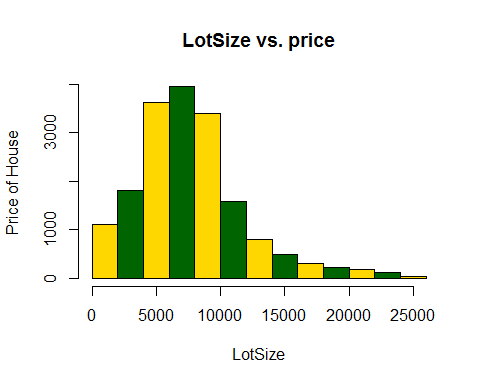
## LotSizeLog of Price of House



## LotSize vs. priceLog of LotSizeLog of Price of House



## Log LotSize vs. price



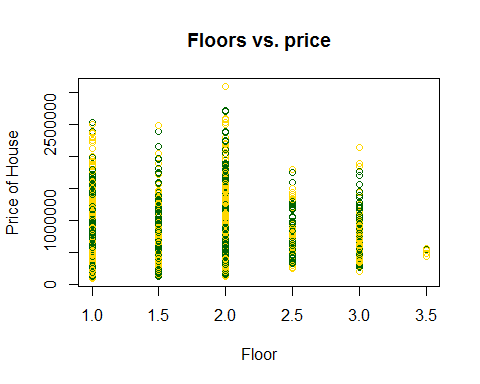
## [1] 0.06714415

bucketByColumn(houseData,7)

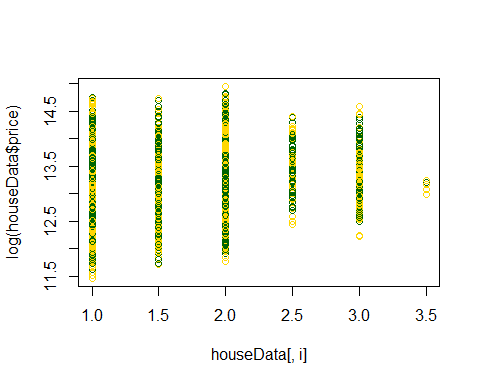
## Min-Max value for: sqft\_lot , MAX: 520, MIN: 25000

## FLOOR Vs Price analysis

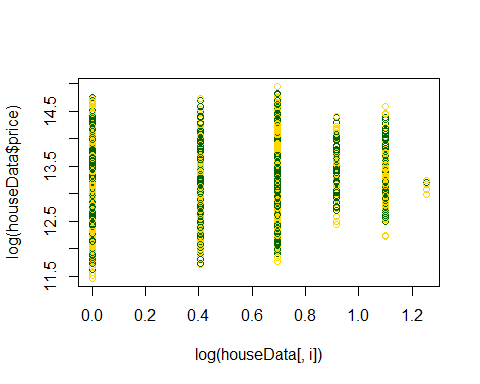
analysis(houseData,8,c('Floors vs. price','Floor', 'Price of House'), 'Y')



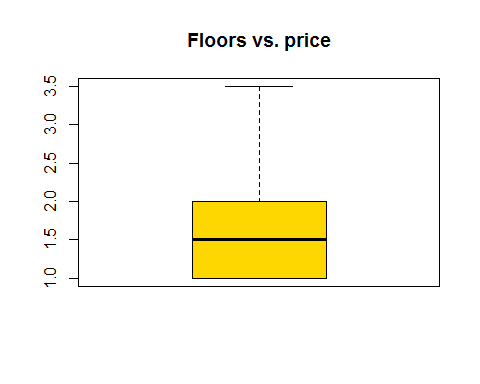
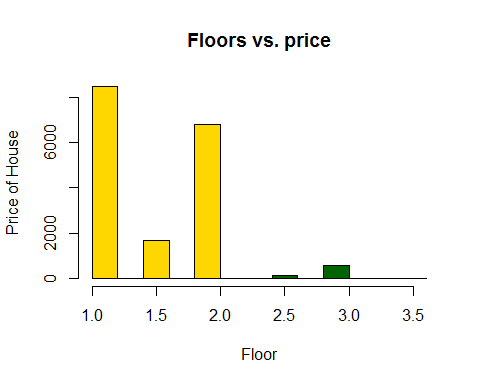
## FloorLog of Price of House



## Floors vs. priceLog of FloorLog of Price of House



## Log Floors vs. price



## [1] 0.2072373

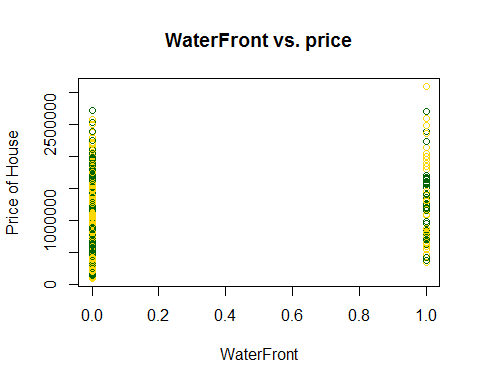
#bucketByColumn(houseData,8)

## SQFT\_LOT Vs Price analysis

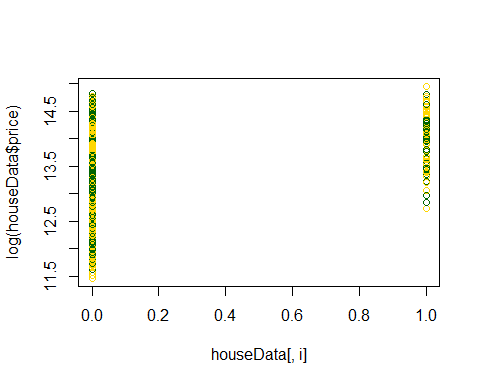
bucketByColumn(houseData,9)

## Min-Max value for: waterfront , MAX: 0, MIN: 1

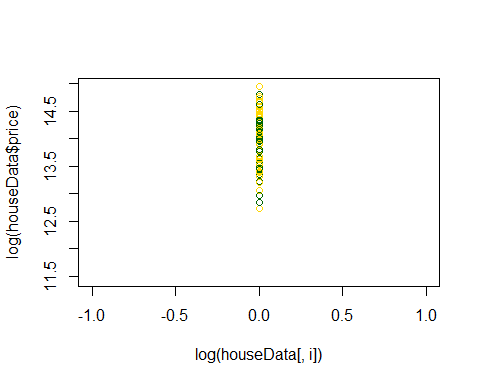
analysis(houseData,9,c('WaterFront vs. price','WaterFront', 'Price of House'), 'Y')



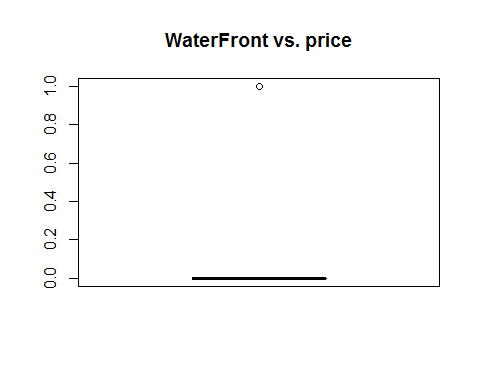
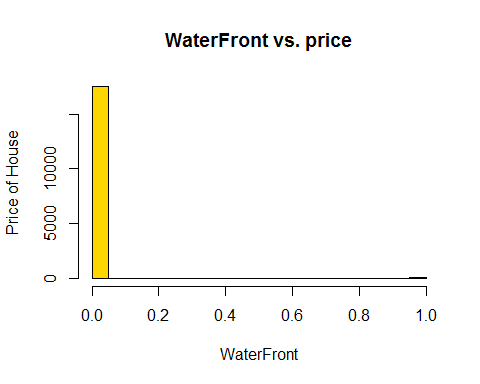
## WaterFrontLog of Price of House



## WaterFront vs. priceLog of WaterFrontLog of Price of House

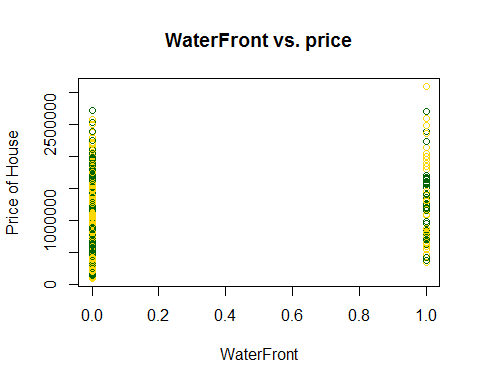


## Log WaterFront vs. price

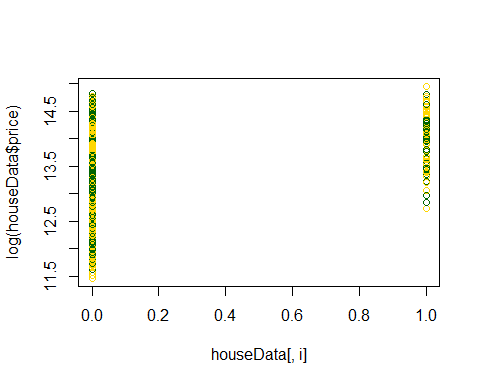


## [1] 0.209102

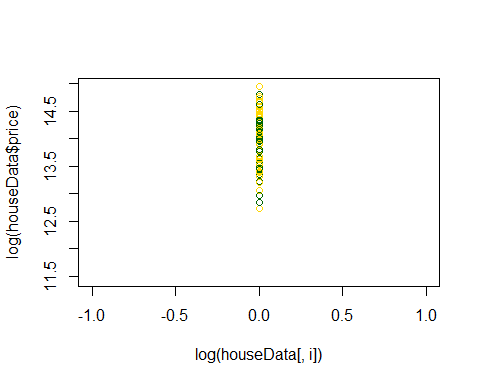
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,sqft\_lot>=0 & sqft\_lot<=25000)  
analysis(houseData,9,c('WaterFront vs. price','WaterFront', 'Price of House'), 'Y')



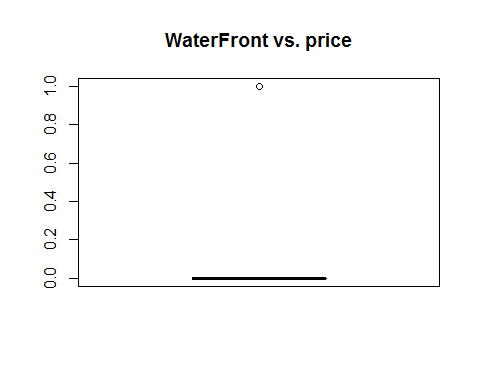
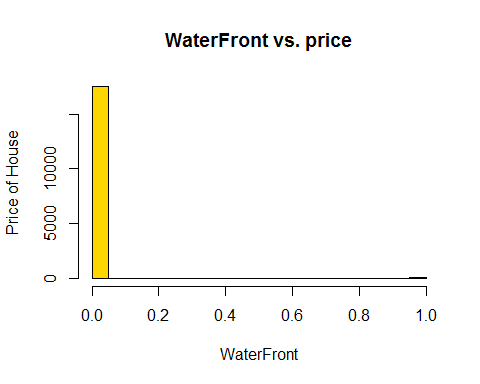
## WaterFrontLog of Price of House



## WaterFront vs. priceLog of WaterFrontLog of Price of House



## Log WaterFront vs. price



## [1] 0.209102

bucketByColumn(houseData,9)

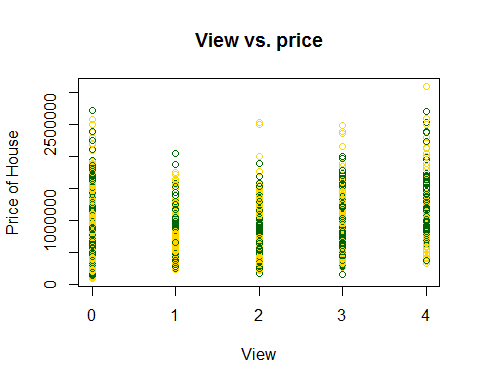
## Min-Max value for: waterfront , MAX: 0, MIN: 1

## View Vs Price analysis

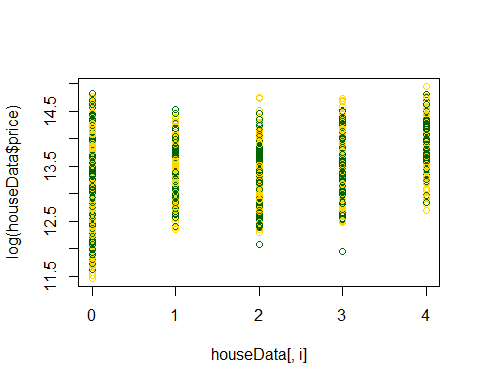
bucketByColumn(houseData,10)

## Min-Max value for: view , MAX: 0, MIN: 4

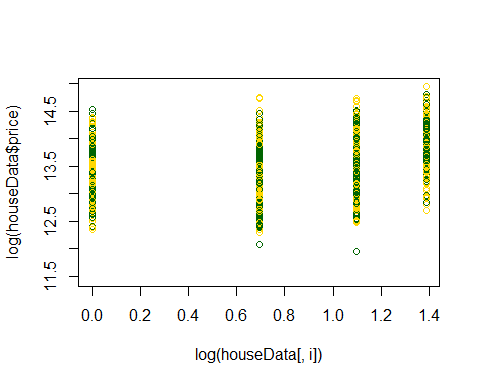
analysis(houseData,10,c('View vs. price','View', 'Price of House'), 'Y')



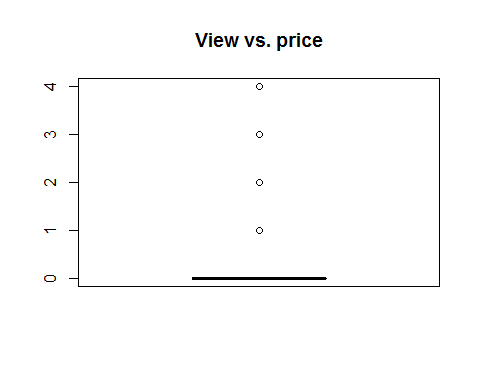
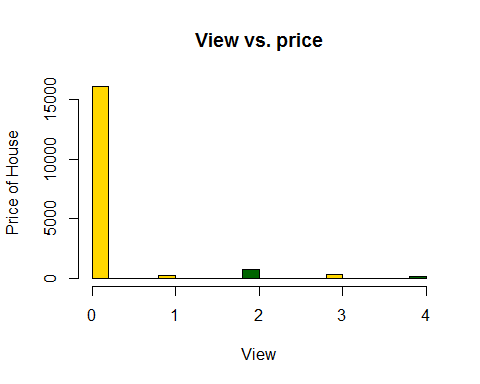
## ViewLog of Price of House



## View vs. priceLog of ViewLog of Price of House

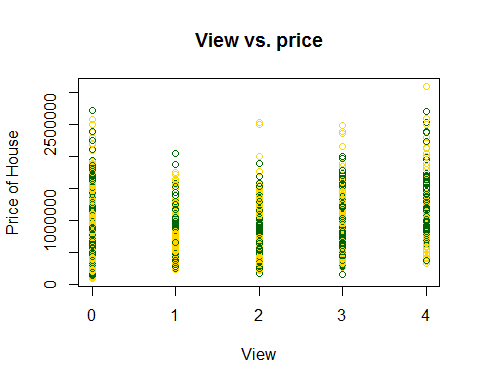


## Log View vs. price

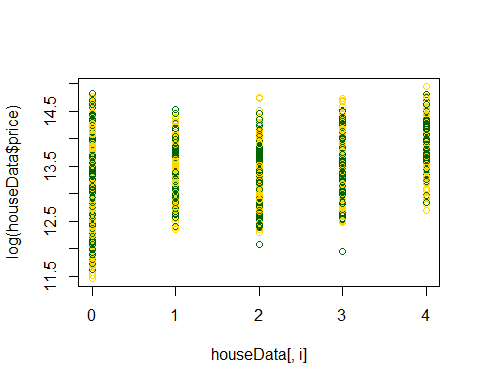


## [1] 0.3578146

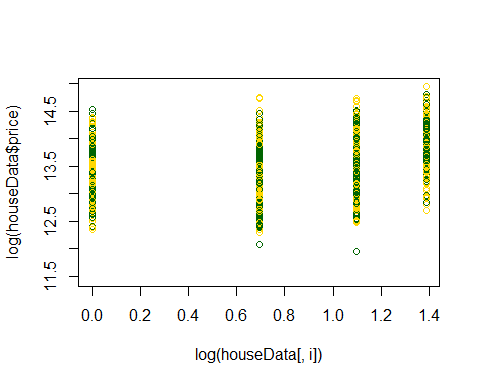
## Price vs. View ->> Nice correlation, view increases [median of bar plot], price increases as well  
#\*\*\*\*\*\*\*Removing the outliers  
analysis(houseData,10,c('View vs. price','View', 'Price of House'), 'Y')



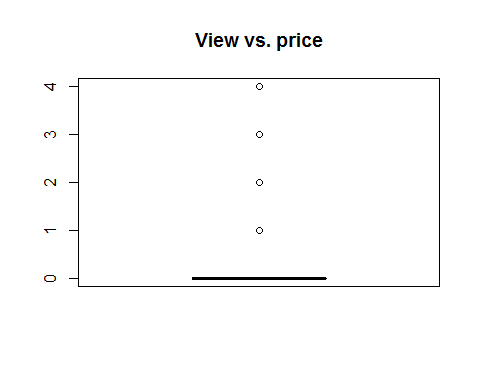
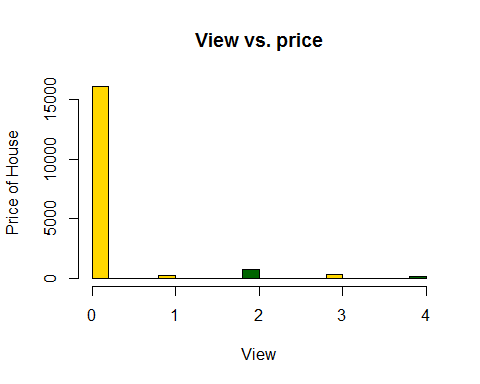
## ViewLog of Price of House



## View vs. priceLog of ViewLog of Price of House



## Log View vs. price



## [1] 0.3578146

bucketByColumn(houseData,10)

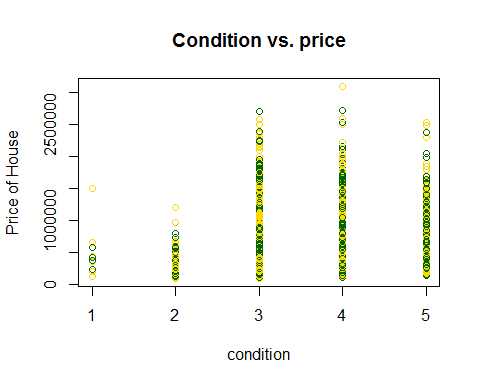
## Min-Max value for: view , MAX: 0, MIN: 4

## CONDITION Vs Price analysis

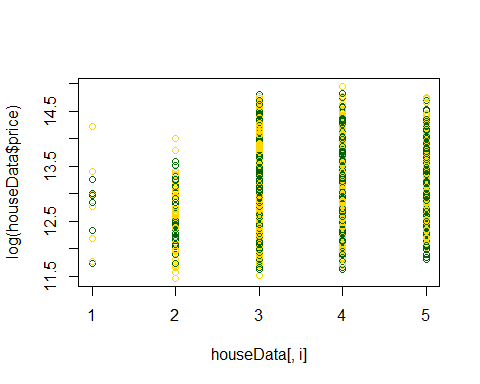
bucketByColumn(houseData,11)

## Min-Max value for: condition , MAX: 1, MIN: 5

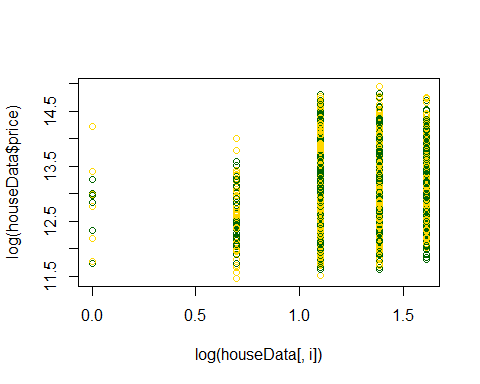
analysis(houseData,11,c('Condition vs. price','condition', 'Price of House'), 'Y')



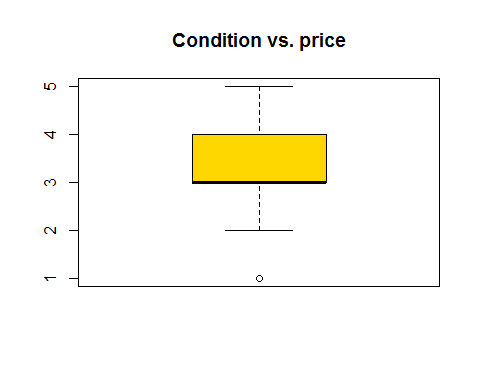
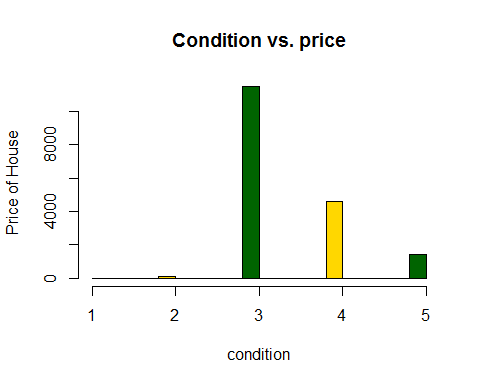
## conditionLog of Price of House



## Condition vs. priceLog of conditionLog of Price of House

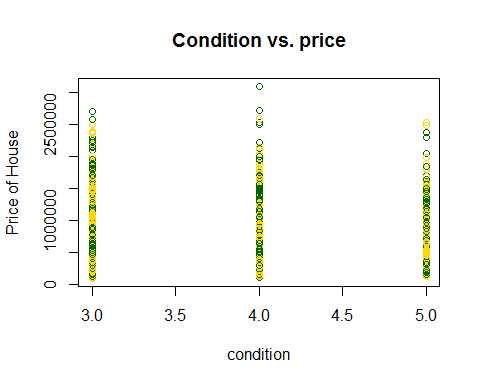


## Log Condition vs. price

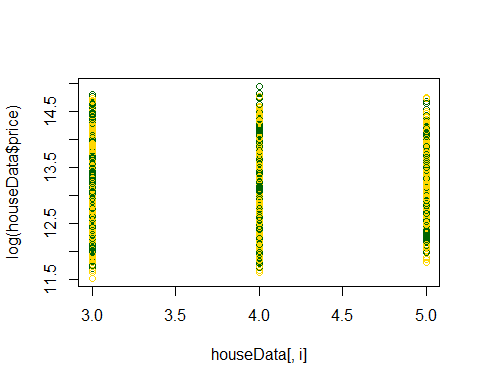


## [1] 0.07301379

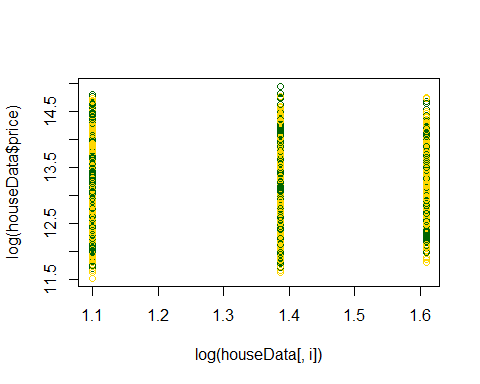
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,condition>=3& condition<=5)  
analysis(houseData,11,c('Condition vs. price','condition', 'Price of House'), 'Y')



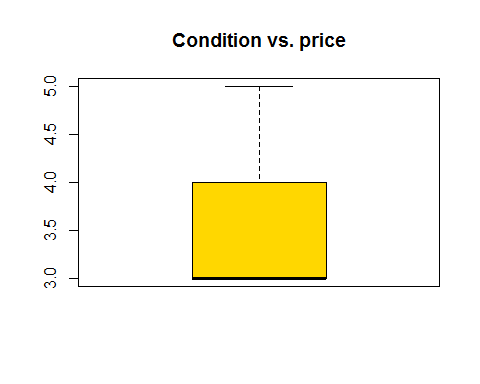
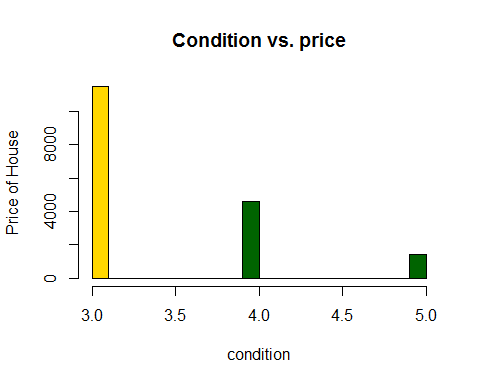
## conditionLog of Price of House



## Condition vs. priceLog of conditionLog of Price of House



## Log Condition vs. price



## [1] 0.06582398

bucketByColumn(houseData,11)

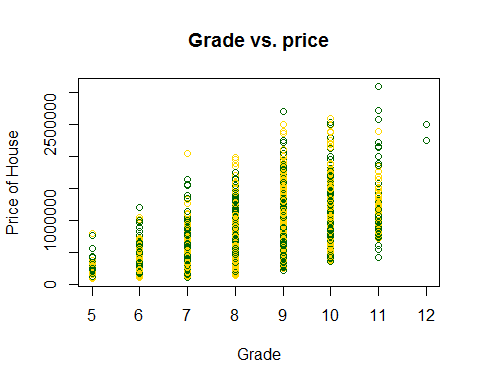
## Min-Max value for: condition , MAX: 3, MIN: 5

## Grade Vs Price analysis

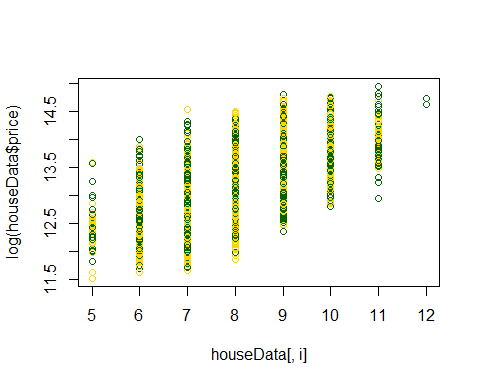
bucketByColumn(houseData,12)

## Min-Max value for: grade , MAX: 5, MIN: 12

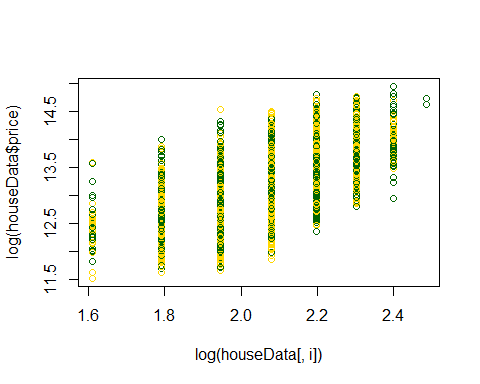
analysis(houseData,12,c('Grade vs. price','Grade', 'Price of House'), 'Y')



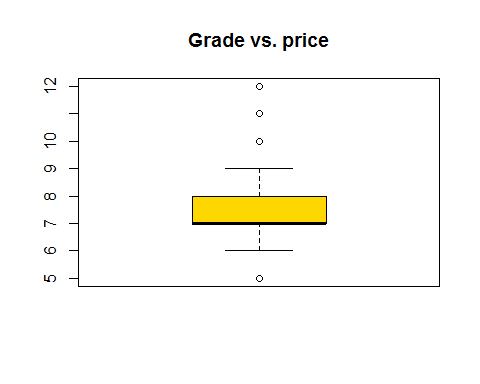
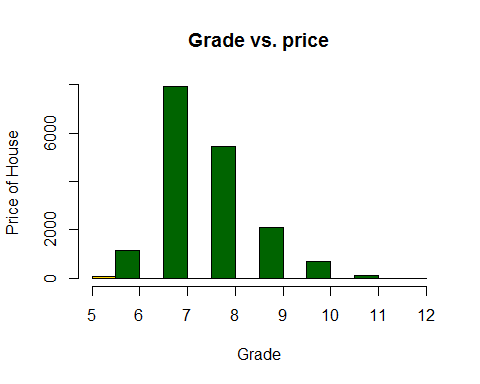
## GradeLog of Price of House



## Grade vs. priceLog of GradeLog of Price of House

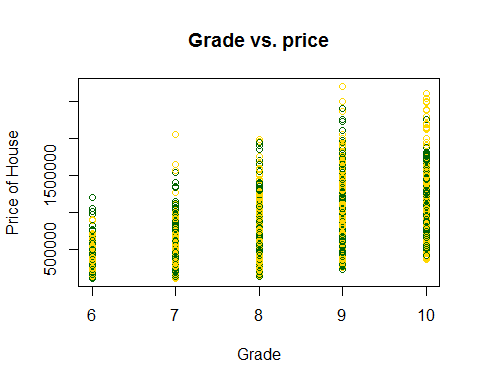


## Log Grade vs. price

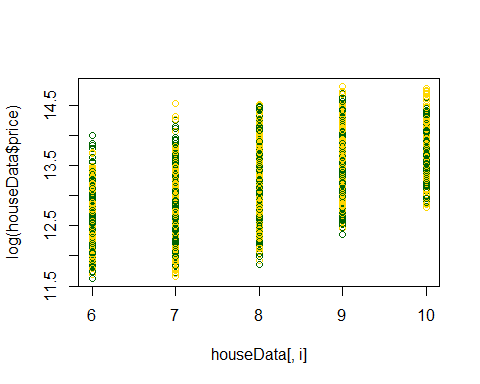


## [1] 0.6086724

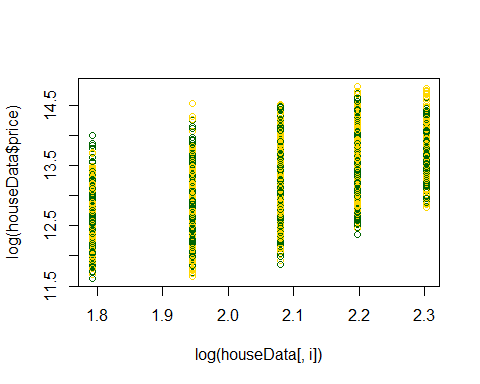
## Price vs. Grade ->> Nice correlation, grade increases [median of bar plot], price increases as well  
#\*\*\*\*\*\*\*Removing the outliers  
#Most of the houses grades are between 6-10   
houseData<-subset(houseData,grade >= 6 & grade<=10)  
analysis(houseData,12,c('Grade vs. price','Grade', 'Price of House'), 'Y')



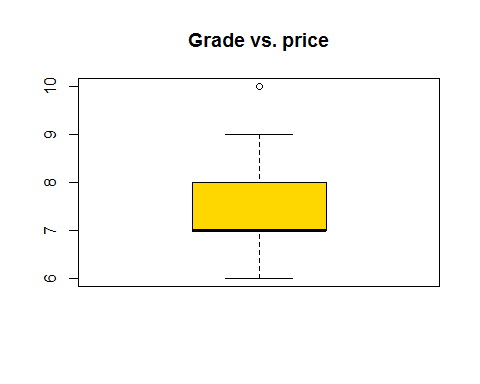
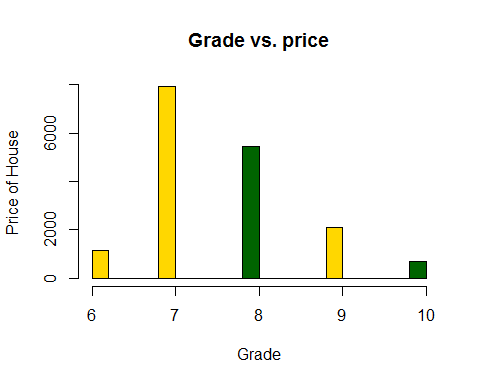
## GradeLog of Price of House



## Grade vs. priceLog of GradeLog of Price of House



## Log Grade vs. price



## [1] 0.5926375

bucketByColumn(houseData,12)

## Min-Max value for: grade , MAX: 6, MIN: 10

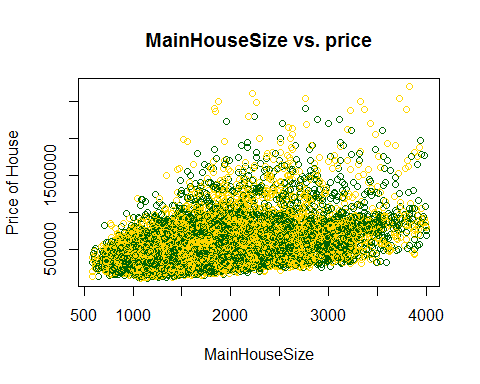
#grade is good without log

# SQFT\_ABOVE Vs Price analysis

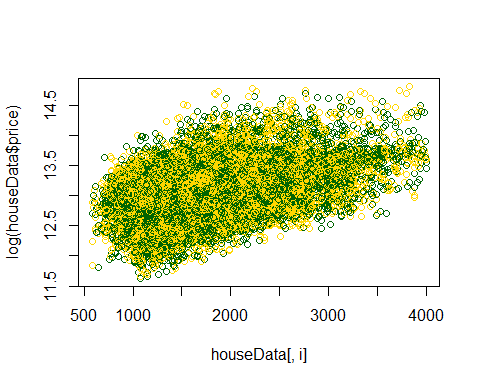
bucketByColumn(houseData,13)

## Min-Max value for: sqft\_above , MAX: 580, MIN: 4000

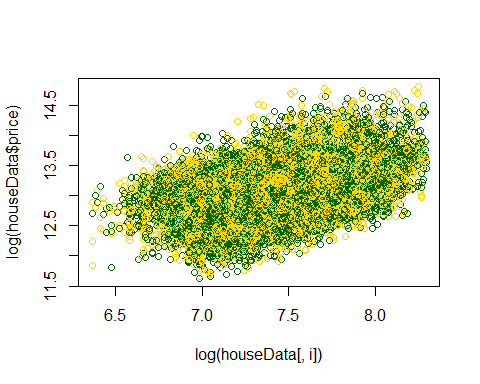
analysis(houseData,13,c('MainHouseSize vs. price','MainHouseSize', 'Price of House'), 'Y')



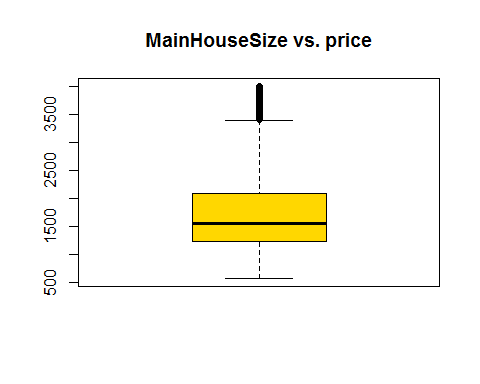
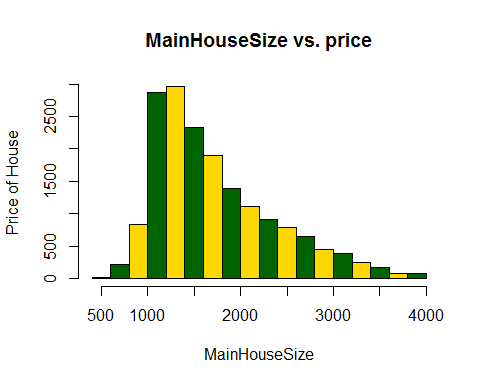
## MainHouseSizeLog of Price of House



## MainHouseSize vs. priceLog of MainHouseSizeLog of Price of House

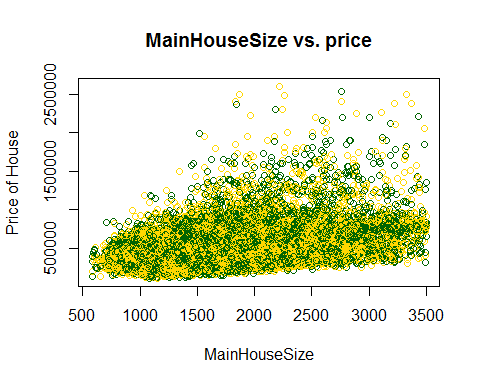


## Log MainHouseSize vs. price

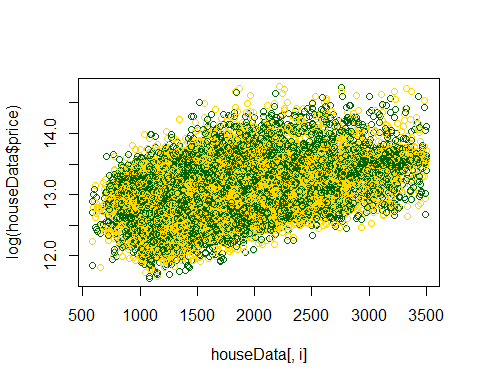


## [1] 0.4454655

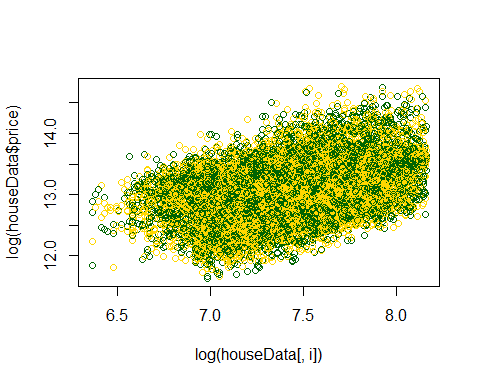
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,sqft\_above >=500 & sqft\_above<=3500)  
analysis(houseData,13,c('MainHouseSize vs. price','MainHouseSize', 'Price of House'), 'Y')



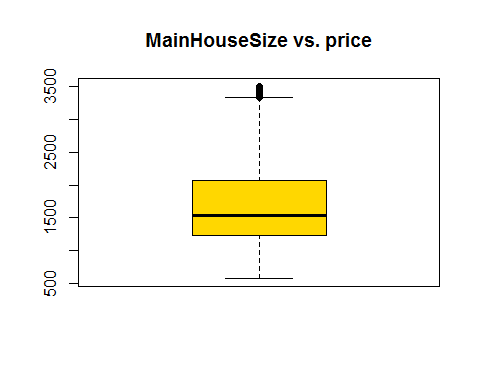
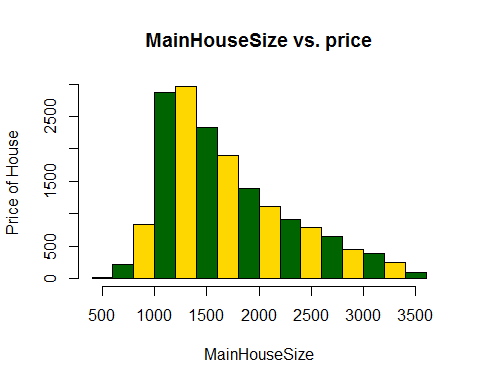
## MainHouseSizeLog of Price of House



## MainHouseSize vs. priceLog of MainHouseSizeLog of Price of House



## Log MainHouseSize vs. price



## [1] 0.4176138

bucketByColumn(houseData,13)

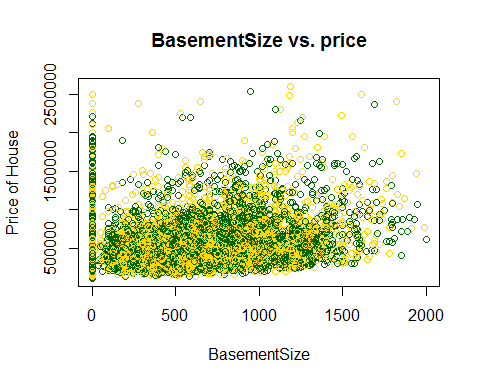
## Min-Max value for: sqft\_above , MAX: 580, MIN: 3500

## SQFT\_BASEMENT Vs Price analysis

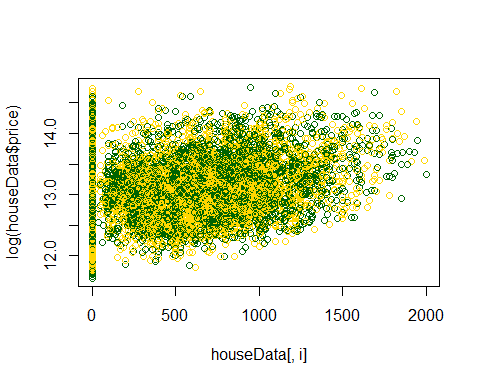
bucketByColumn(houseData,14)

## Min-Max value for: sqft\_basement , MAX: 0, MIN: 2000

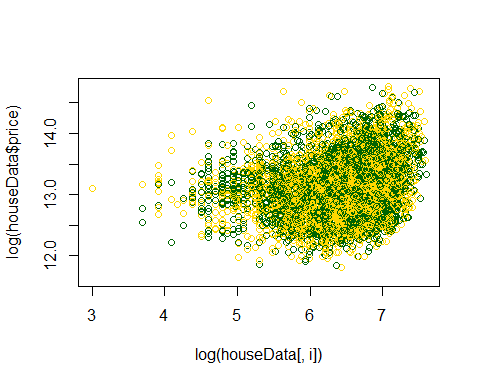
analysis(houseData,14,c('BasementSize vs. price','BasementSize', 'Price of House'), 'Y')



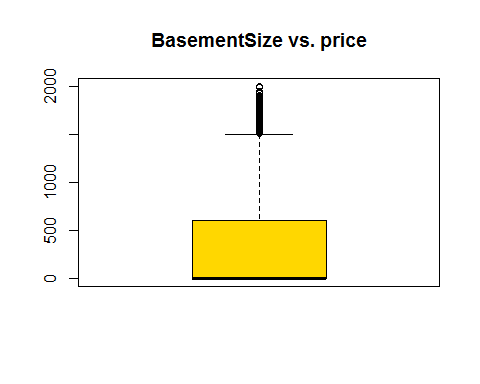
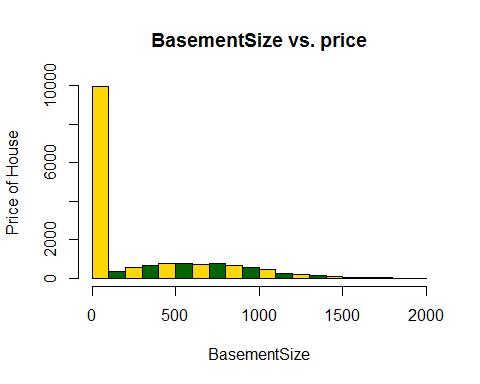
## BasementSizeLog of Price of House



## BasementSize vs. priceLog of BasementSizeLog of Price of House

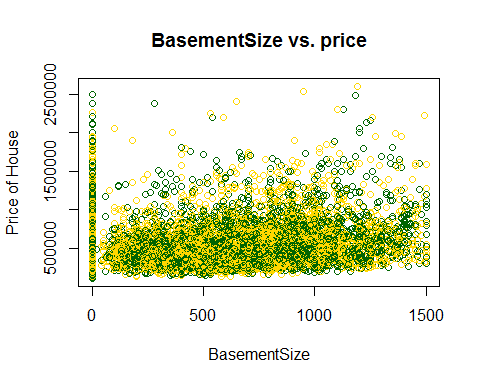


## Log BasementSize vs. price

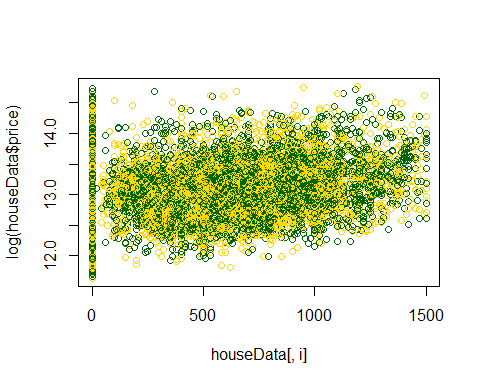


## [1] 0.2686956

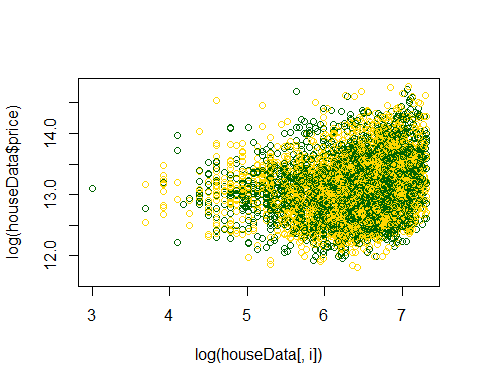
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,sqft\_basement >=0 & sqft\_basement<=1500)  
analysis(houseData,14,c('BasementSize vs. price','BasementSize', 'Price of House'), 'Y')



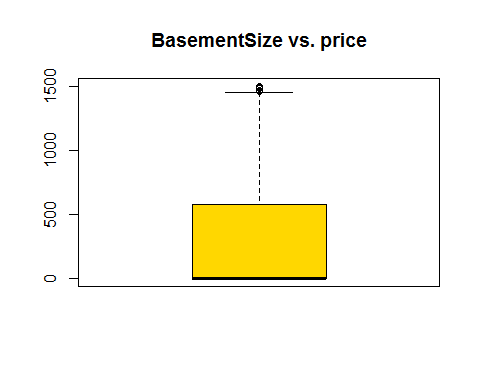
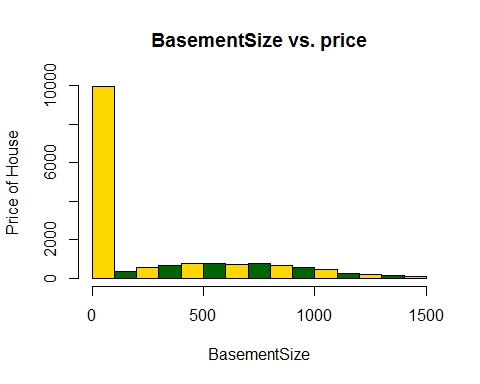
## BasementSizeLog of Price of House



## BasementSize vs. priceLog of BasementSizeLog of Price of House



## Log BasementSize vs. price



## [1] 0.2404668

bucketByColumn(houseData,14)

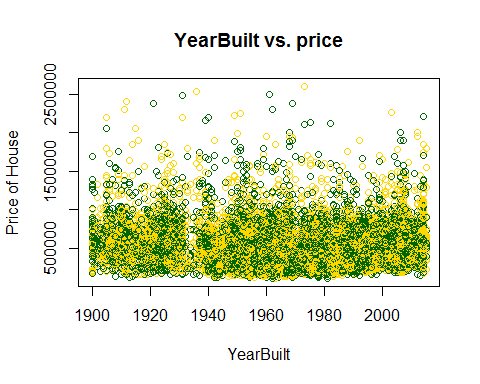
## Min-Max value for: sqft\_basement , MAX: 0, MIN: 1500

# YR\_BUILT Vs Price analysis

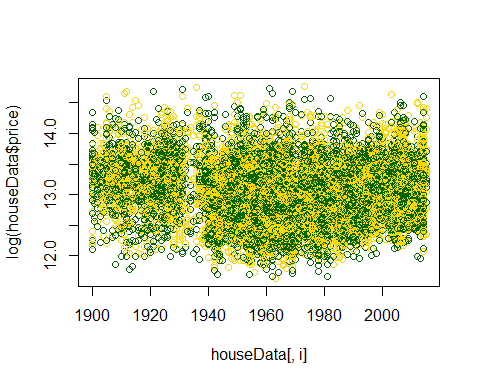
bucketByColumn(houseData,15)

## Min-Max value for: yr\_built , MAX: 1900, MIN: 2015

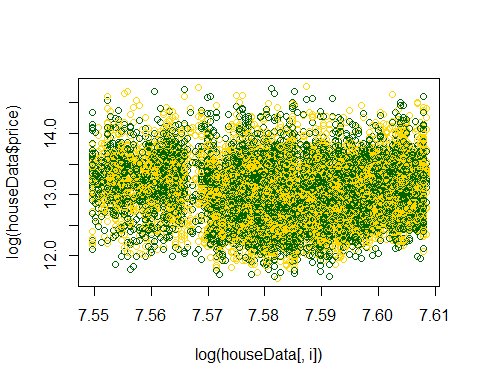
analysis(houseData,15,c('YearBuilt vs. price','YearBuilt', 'Price of House'), 'Y')



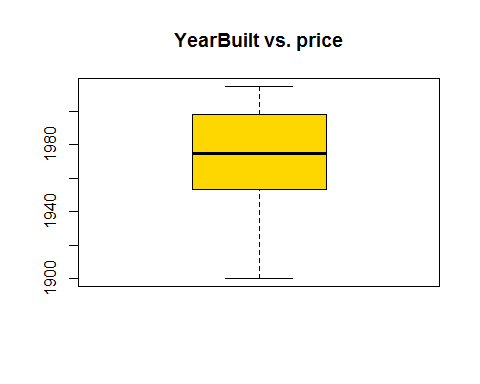
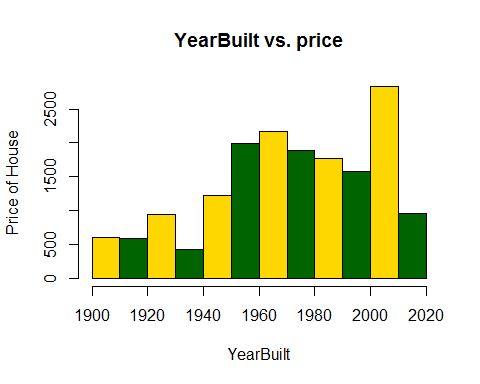
## YearBuiltLog of Price of House



## YearBuilt vs. priceLog of YearBuiltLog of Price of House

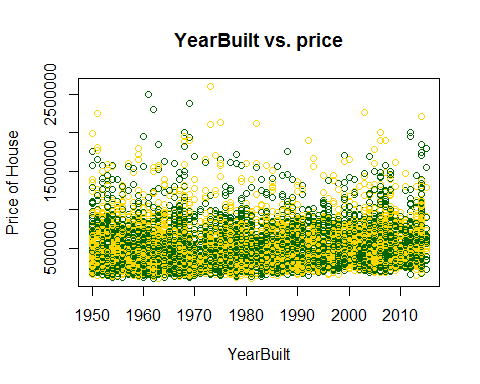


## Log YearBuilt vs. price

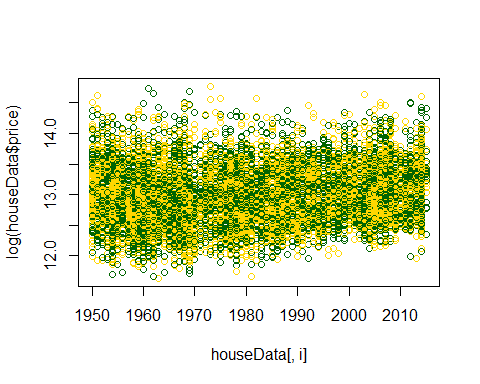


## [1] -0.09523834

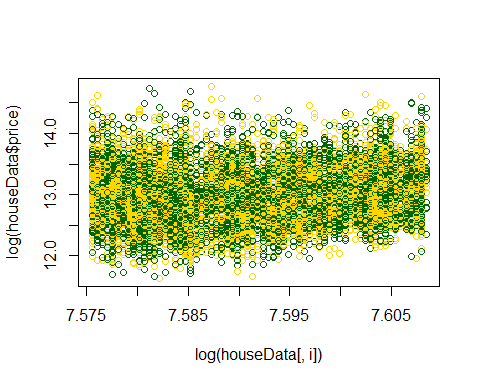
#\*\*\*\*\*\*\*Removing the outliers  
#In our data some records are too old..I just removed that data from my model.  
#Because It doesn't make any sense to keep more than 100 years house in our model  
houseData<-subset(houseData,yr\_built>=1950& yr\_built<=2015)  
analysis(houseData,15,c('YearBuilt vs. price','YearBuilt', 'Price of House'), 'Y')



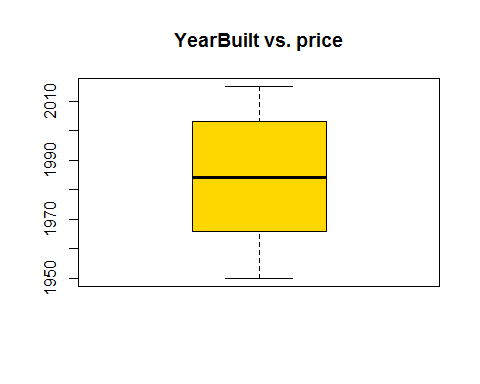
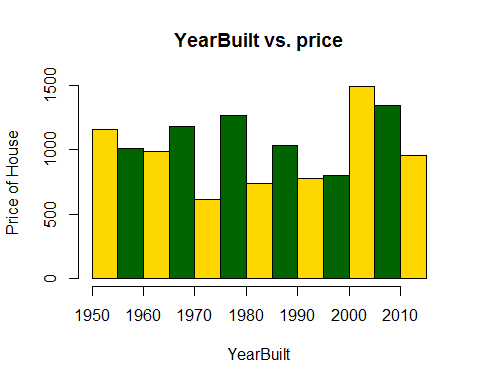
## YearBuiltLog of Price of House



## YearBuilt vs. priceLog of YearBuiltLog of Price of House



## Log YearBuilt vs. price



## [1] 0.100292

bucketByColumn(houseData,15)

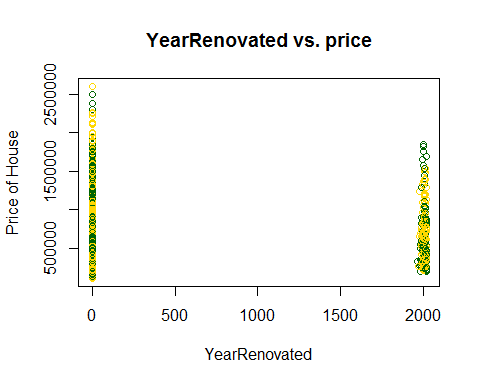
## Min-Max value for: yr\_built , MAX: 1950, MIN: 2015

# YR\_BUILT Vs Price analysis

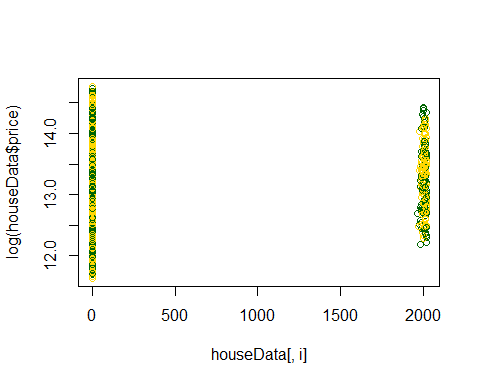
bucketByColumn(houseData,16)

## Min-Max value for: yr\_renovated , MAX: 0, MIN: 2015

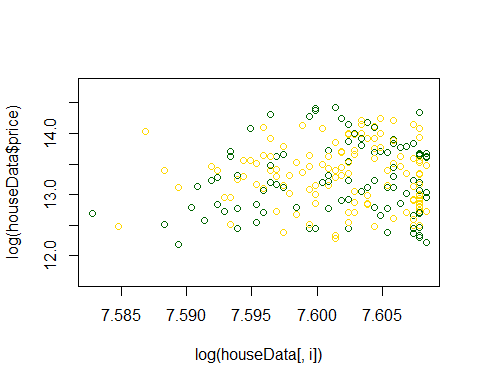
analysis(houseData,16,c('YearRenovated vs. price','YearRenovated', 'Price of House'), 'Y')



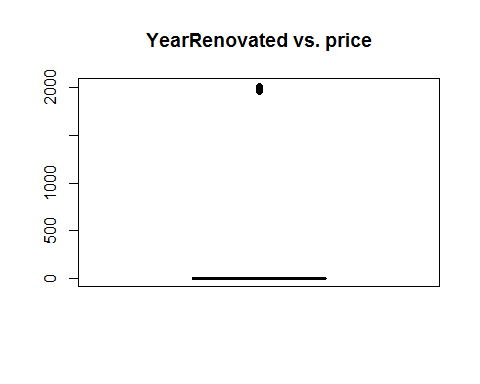
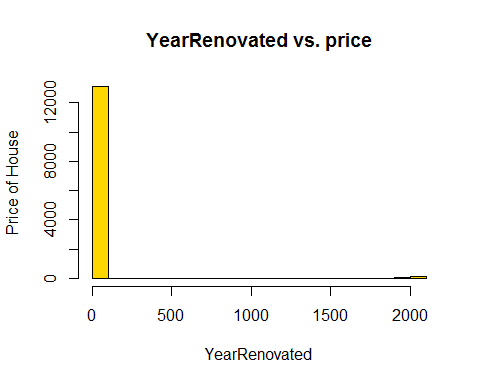
## YearRenovatedLog of Price of House



## YearRenovated vs. priceLog of YearRenovatedLog of Price of House

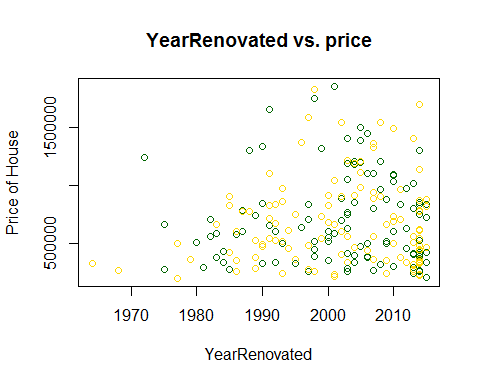


## Log YearRenovated vs. price

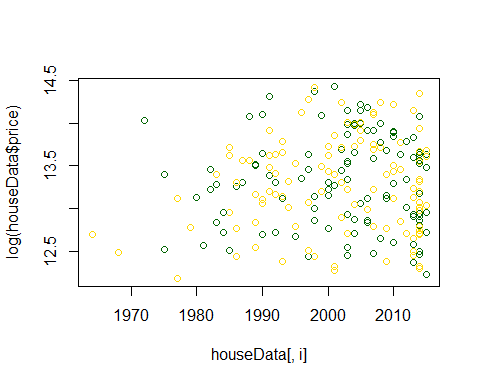


## [1] 0.1115179

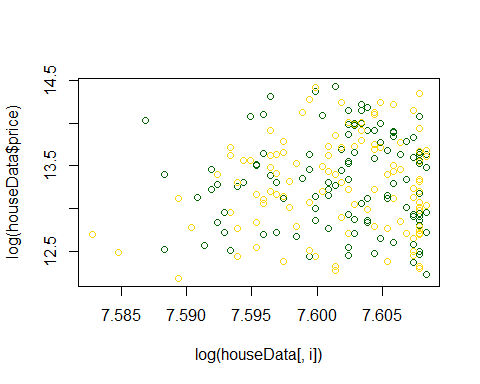
#\*\*\*\*\*\*\*Removing the outliers  
#In our data some records are too old..I just removed that data from my model.  
#Because It doesn't make any sense to keep more than 100 years house in our model  
houseData<-subset(houseData,yr\_renovated>=1950& yr\_renovated<=2015)  
analysis(houseData,16,c('YearRenovated vs. price','YearRenovated', 'Price of House'), 'Y')



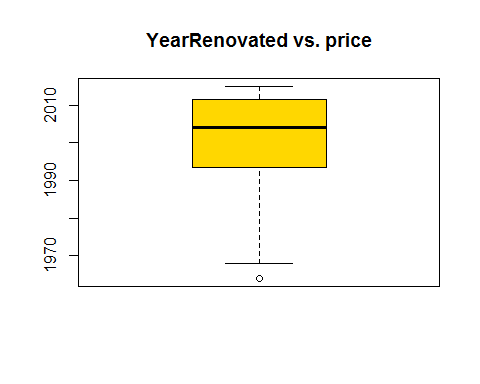
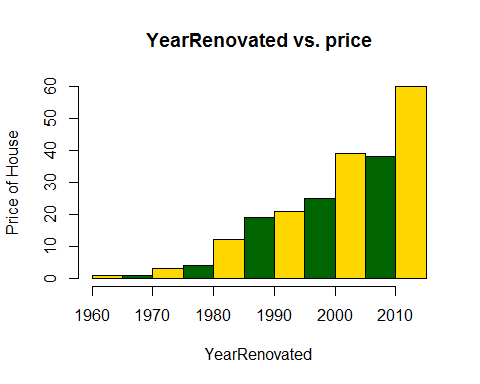
## YearRenovatedLog of Price of House



## YearRenovated vs. priceLog of YearRenovatedLog of Price of House



## Log YearRenovated vs. price



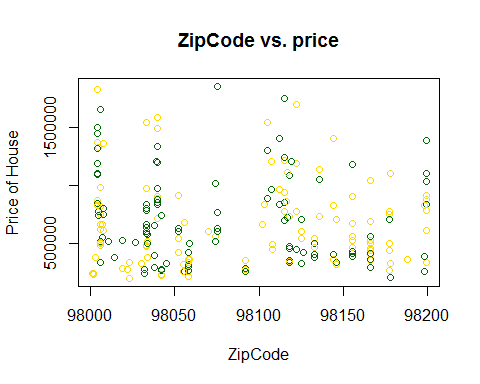
## [1] 0.05704174

bucketByColumn(houseData,16)

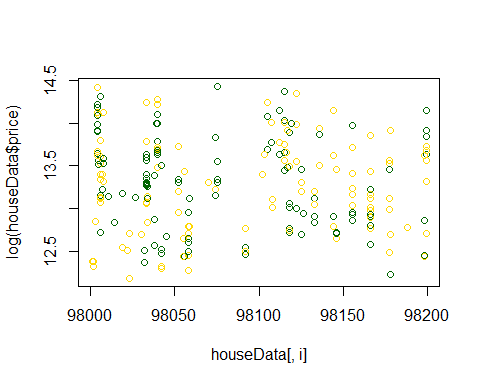
## Min-Max value for: yr\_renovated , MAX: 1964, MIN: 2015

# ZIPCODE Vs Price analysis

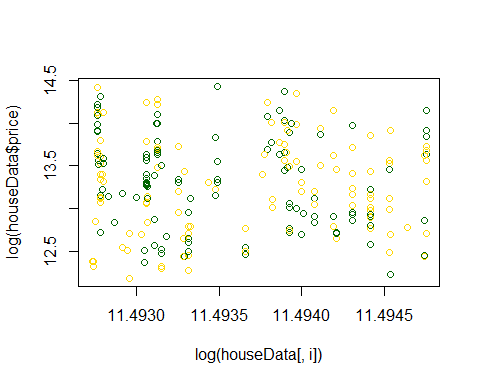
analysis(houseData,17,c('ZipCode vs. price','ZipCode', 'Price of House'), 'Y')



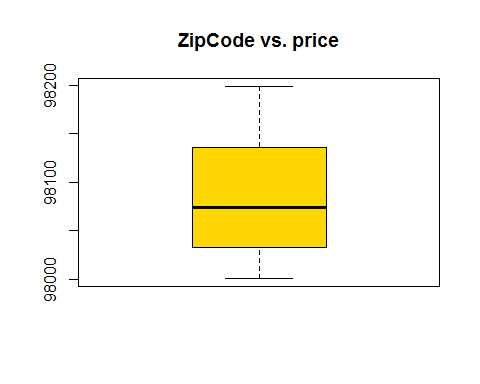
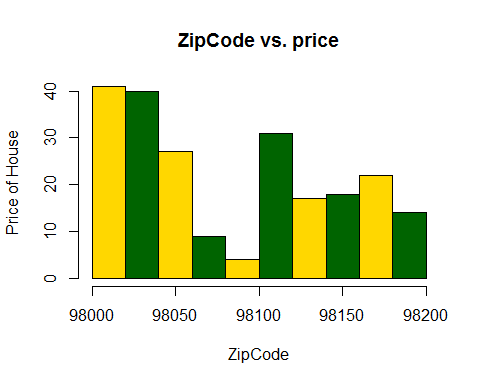
## ZipCodeLog of Price of House



## ZipCode vs. priceLog of ZipCodeLog of Price of House



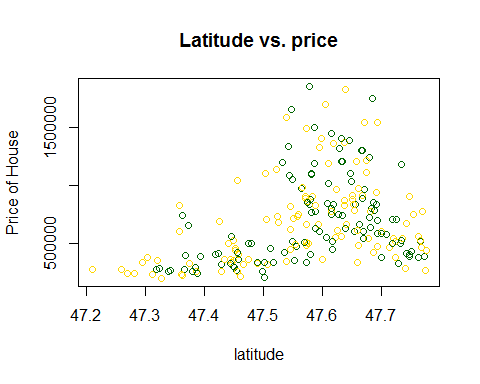
## Log ZipCode vs. price



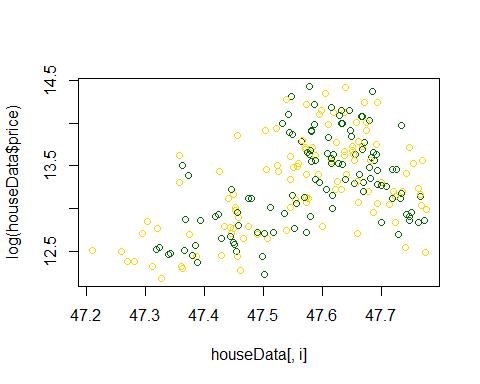
## [1] -0.07696891

## LAT Vs Price analysis

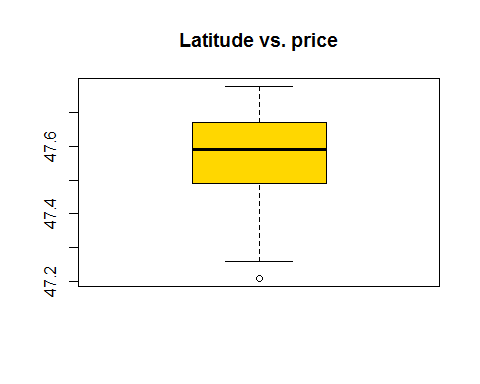
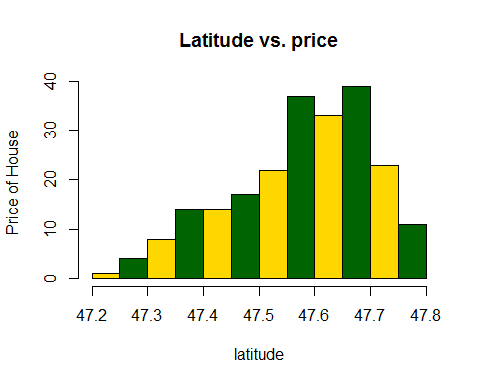
analysis(houseData,18,c('Latitude vs. price','latitude', 'Price of House'), 'N')



## latitudeLog of Price of House

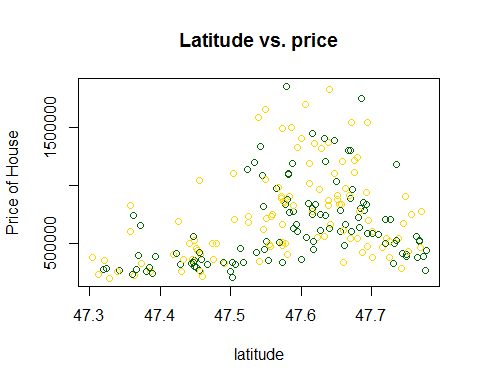


## Latitude vs. price

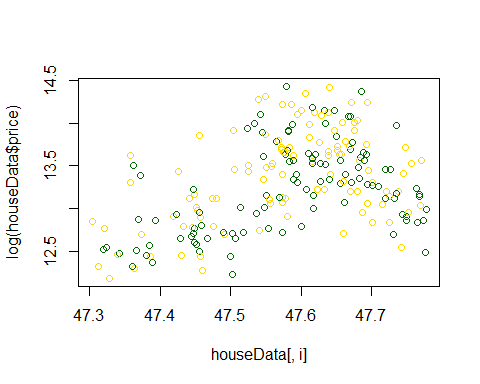


## [1] 0.3502248

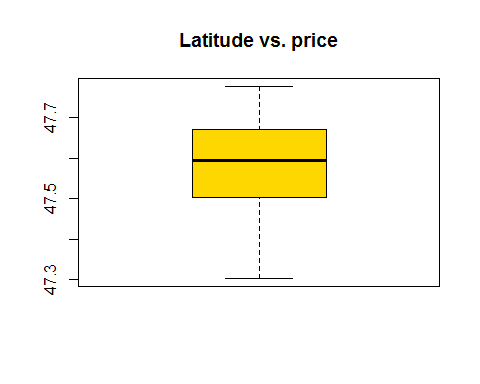
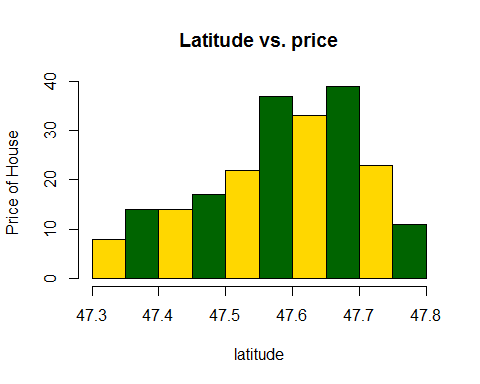
## Price vs. Lat ->> This is more like a normal dist relationship, price peaks around when lat= 47.64 and declines afterwards, but this can be modeled easily. I would say Lat explains the price as well.  
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,lat>=47.3)  
analysis(houseData,18,c('Latitude vs. price','latitude', 'Price of House'), 'N')



## latitudeLog of Price of House



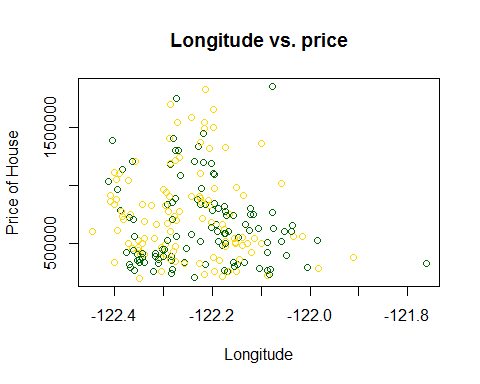
## Latitude vs. price



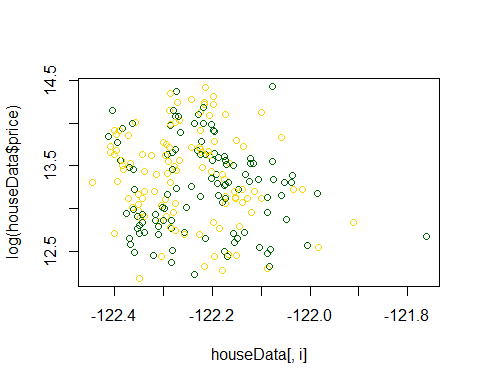
## [1] 0.31377

## LONG Vs Price analysis

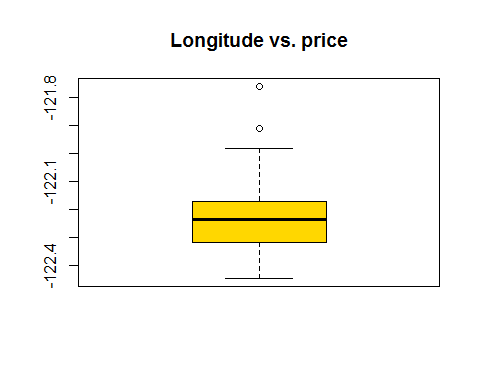
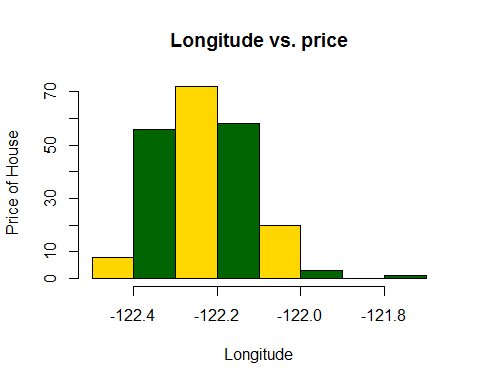
analysis(houseData,19,c('Longitude vs. price','Longitude', 'Price of House'), 'N')



## LongitudeLog of Price of House

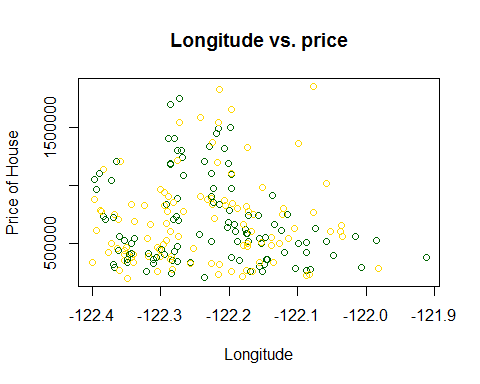


## Longitude vs. price

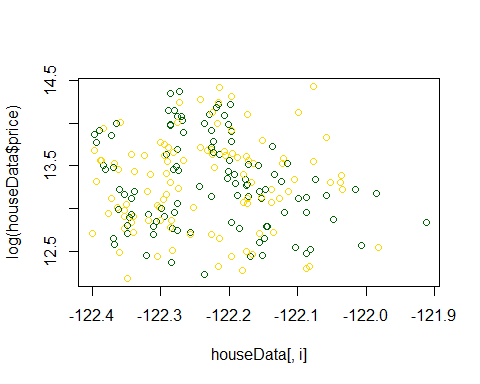


## [1] -0.1101141

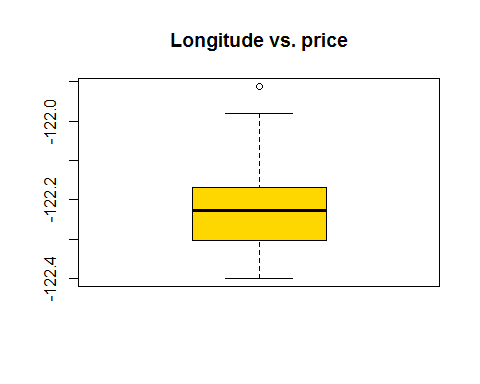
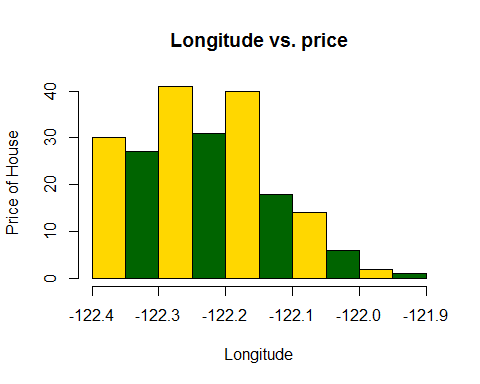
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,long>=-122.4 & long < -121.8)  
analysis(houseData,19,c('Longitude vs. price','Longitude', 'Price of House'), 'N')



## LongitudeLog of Price of House



## Longitude vs. price



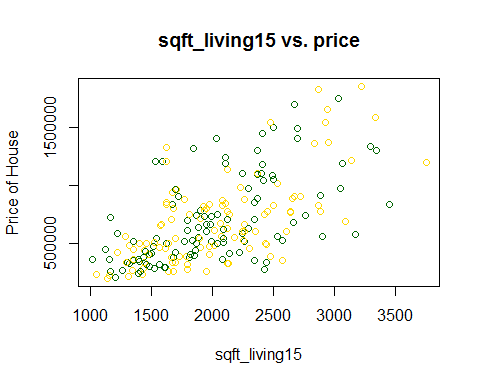
## [1] -0.05926148

## SQFT\_LIVING15 Vs Price analysis

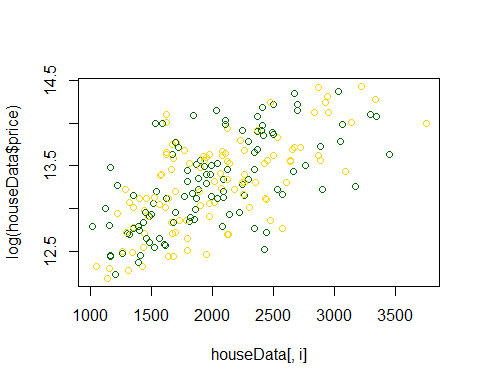
bucketByColumn(houseData,20)

## Min-Max value for: sqft\_living15 , MAX: 1010, MIN: 3750

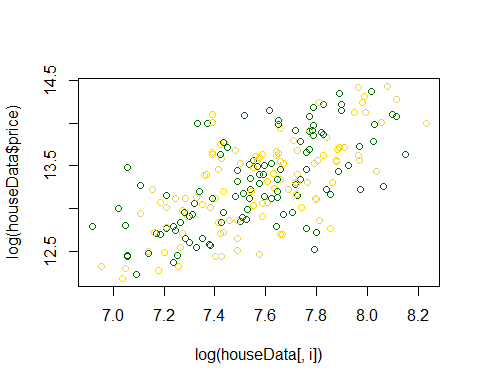
analysis(houseData,20,c('sqft\_living15 vs. price','sqft\_living15', 'Price of House'), 'Y')



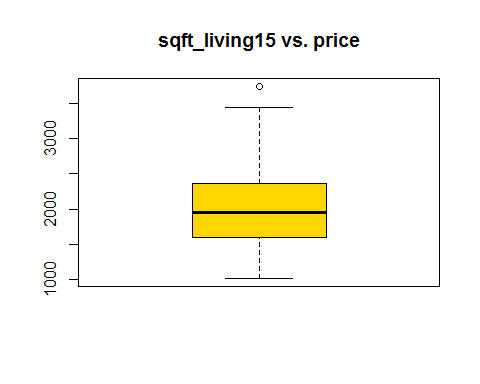
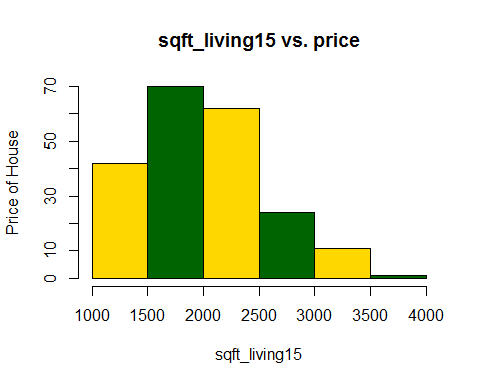
## sqft\_living15Log of Price of House



## sqft\_living15 vs. priceLog of sqft\_living15Log of Price of House



## Log sqft\_living15 vs. price



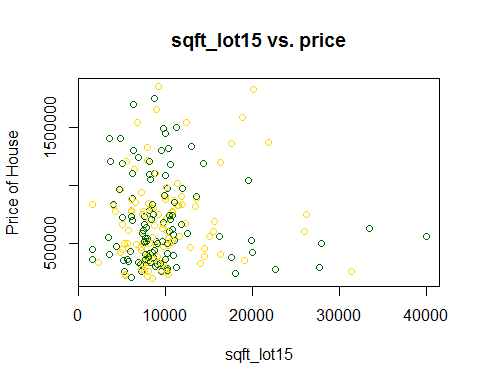
## [1] 0.6262834

## SQFT\_LOT15 Vs Price analysis

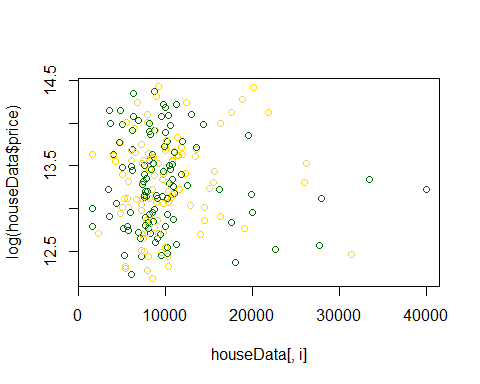
bucketByColumn(houseData,21)

## Min-Max value for: sqft\_lot15 , MAX: 1517, MIN: 39921

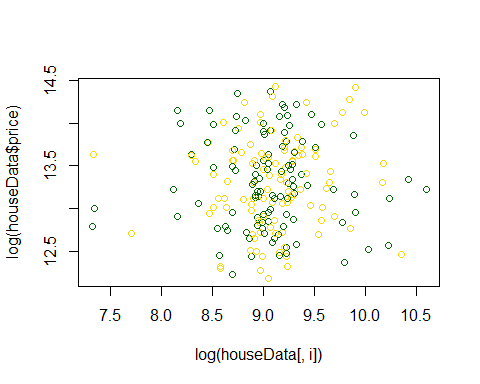
analysis(houseData,21,c('sqft\_lot15 vs. price','sqft\_lot15', 'Price of House'), 'Y')



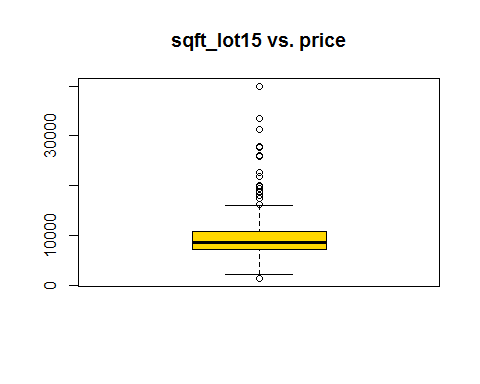
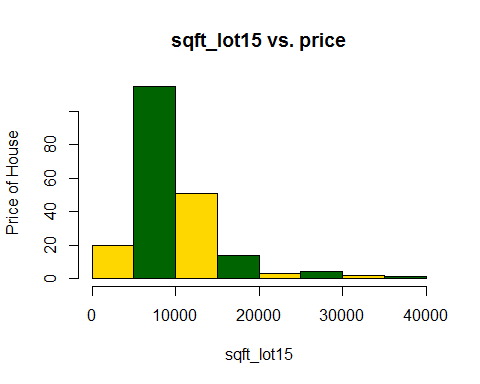
## sqft\_lot15Log of Price of House



## sqft\_lot15 vs. priceLog of sqft\_lot15Log of Price of House

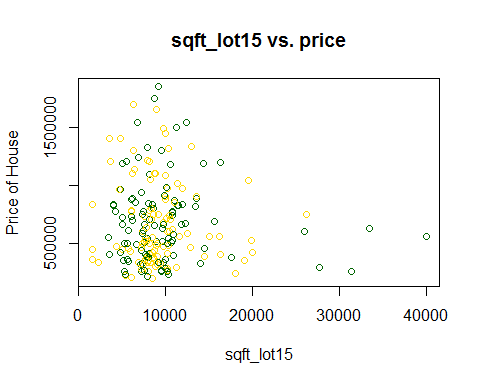


## Log sqft\_lot15 vs. price

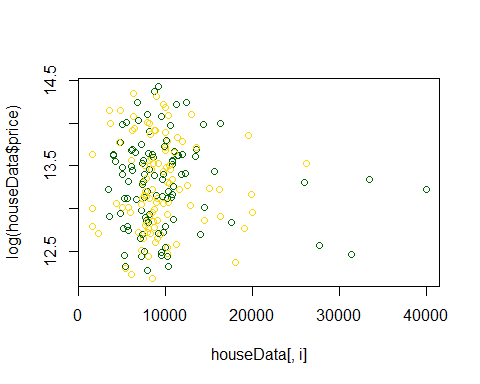


## [1] 0.007371907

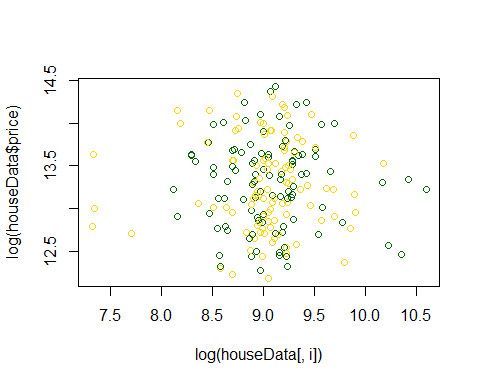
#\*\*\*\*\*\*\*Removing the outliers  
houseData<-subset(houseData,sqft\_lot15>=0 & sqft\_lot<=20000)  
analysis(houseData,21,c('sqft\_lot15 vs. price','sqft\_lot15', 'Price of House'), 'Y')



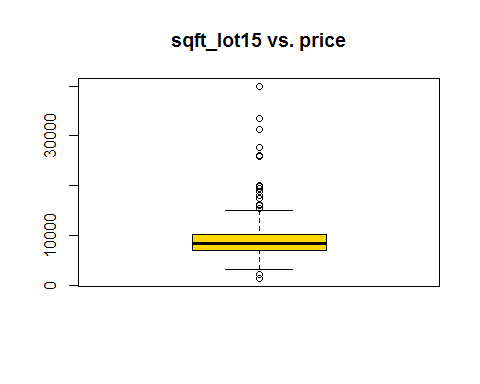
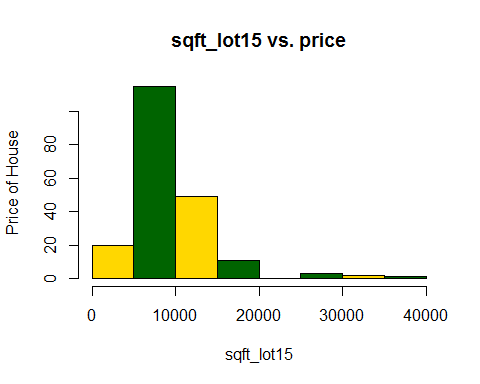
## sqft\_lot15Log of Price of House



## sqft\_lot15 vs. priceLog of sqft\_lot15Log of Price of House



## Log sqft\_lot15 vs. price



## [1] -0.06148357

bucketByColumn(houseData,21)

## Min-Max value for: sqft\_lot15 , MAX: 1517, MIN: 39921

## Correlation among all the variables

#\*\*\*\*\*\*\*Only sqft\_living & sqft\_above,sqft\_living & grade,sqft\_living & bathrooms have good correlation between them  
houseData$date <- NULL  
houseData$id <- NULL  
cor(houseData)

## price bedrooms bathrooms sqft\_living sqft\_lot  
## price 1.00000000 0.24534387 0.55037096 0.630896571 -0.04670741  
## bedrooms 0.24534387 1.00000000 0.50198434 0.498213277 0.11646077  
## bathrooms 0.55037096 0.50198434 1.00000000 0.713341225 -0.01608470  
## sqft\_living 0.63089657 0.49821328 0.71334123 1.000000000 0.16887575  
## sqft\_lot -0.04670741 0.11646077 -0.01608470 0.168875749 1.00000000  
## floors 0.33819490 0.20696628 0.38432741 0.369428509 -0.23587572  
## waterfront 0.29194531 -0.02827068 0.08956278 0.114398983 0.01537475  
## view 0.49320149 0.04168977 0.23208185 0.340677542 -0.02748737  
## condition 0.05810468 0.12937252 0.03295516 0.056884391 0.05586661  
## grade 0.74330005 0.20651645 0.51382941 0.577581300 -0.06279296  
## sqft\_above 0.53017146 0.31581056 0.52273253 0.773448581 0.13936737  
## sqft\_basement 0.28970273 0.36456943 0.42884524 0.547508057 0.08090849  
## yr\_built 0.01546749 -0.01232044 0.11623613 0.005833666 -0.01968473  
## yr\_renovated 0.06349496 0.03150590 0.14997618 -0.119493001 -0.06769564  
## zipcode -0.11253315 -0.10462421 0.05526834 0.017271193 -0.24592828  
## lat 0.30569239 -0.05294372 0.15530825 0.079501204 -0.14808742  
## long -0.06190430 -0.01621980 -0.11182024 -0.074018050 0.34082873  
## sqft\_living15 0.60537395 0.17429231 0.40790720 0.601391901 0.22492989  
## sqft\_lot15 -0.06148357 0.03364511 -0.01785772 0.122075769 0.75097078  
## floors waterfront view condition  
## price 0.33819490 0.291945312 0.493201491 0.058104683  
## bedrooms 0.20696628 -0.028270676 0.041689772 0.129372520  
## bathrooms 0.38432741 0.089562775 0.232081850 0.032955157  
## sqft\_living 0.36942851 0.114398983 0.340677542 0.056884391  
## sqft\_lot -0.23587572 0.015374745 -0.027487369 0.055866610  
## floors 1.00000000 0.125439165 0.215350942 0.074238532  
## waterfront 0.12543916 1.000000000 0.536069443 0.025928026  
## view 0.21535094 0.536069443 1.000000000 0.103181693  
## condition 0.07423853 0.025928026 0.103181693 1.000000000  
## grade 0.34789857 0.173479177 0.375717055 -0.006189817  
## sqft\_above 0.57350542 0.128004203 0.252287169 0.109358719  
## sqft\_basement -0.17764159 0.010458064 0.201321980 -0.055143586  
## yr\_built 0.11305927 0.065529884 0.002676771 -0.071904471  
## yr\_renovated -0.12781611 -0.091057992 -0.143337199 -0.342100953  
## zipcode 0.18439419 0.092813853 0.172323323 -0.129499001  
## lat 0.13670734 -0.051904095 -0.002601130 0.021821434  
## long -0.17607971 0.025188194 -0.151927300 0.083260973  
## sqft\_living15 0.24849214 0.221775843 0.429121930 -0.019889073  
## sqft\_lot15 -0.04180592 0.008134168 -0.028301938 0.095022706  
## grade sqft\_above sqft\_basement yr\_built  
## price 0.743300047 0.530171463 0.28970273 0.015467488  
## bedrooms 0.206516447 0.315810558 0.36456943 -0.012320435  
## bathrooms 0.513829408 0.522732530 0.42884524 0.116236131  
## sqft\_living 0.577581300 0.773448581 0.54750806 0.005833666  
## sqft\_lot -0.062792960 0.139367370 0.08090849 -0.019684726  
## floors 0.347898566 0.573505423 -0.17764159 0.113059270  
## waterfront 0.173479177 0.128004203 0.01045806 0.065529884  
## view 0.375717055 0.252287169 0.20132198 0.002676771  
## condition -0.006189817 0.109358719 -0.05514359 -0.071904471  
## grade 1.000000000 0.453490533 0.30730463 0.164667036  
## sqft\_above 0.453490533 1.000000000 -0.10694418 0.046095939  
## sqft\_basement 0.307304632 -0.106944183 1.00000000 -0.051703751  
## yr\_built 0.164667036 0.046095939 -0.05170375 1.000000000  
## yr\_renovated 0.089117618 -0.187806347 0.06050039 0.203625141  
## zipcode 0.074497977 -0.035614122 0.07410805 -0.158744753  
## lat 0.132381928 0.103958090 -0.01253734 -0.114618735  
## long -0.162650045 -0.007593856 -0.10607882 0.302660377  
## sqft\_living15 0.515048399 0.478200586 0.31203228 -0.013978974  
## sqft\_lot15 -0.025336255 0.164193835 -0.02527660 0.035314614  
## yr\_renovated zipcode lat long  
## price 0.063494960 -0.112533150 0.30569239 -0.061904301  
## bedrooms 0.031505902 -0.104624214 -0.05294372 -0.016219803  
## bathrooms 0.149976181 0.055268341 0.15530825 -0.111820238  
## sqft\_living -0.119493001 0.017271193 0.07950120 -0.074018050  
## sqft\_lot -0.067695641 -0.245928280 -0.14808742 0.340828735  
## floors -0.127816115 0.184394189 0.13670734 -0.176079712  
## waterfront -0.091057992 0.092813853 -0.05190410 0.025188194  
## view -0.143337199 0.172323323 -0.00260113 -0.151927300  
## condition -0.342100953 -0.129499001 0.02182143 0.083260973  
## grade 0.089117618 0.074497977 0.13238193 -0.162650045  
## sqft\_above -0.187806347 -0.035614122 0.10395809 -0.007593856  
## sqft\_basement 0.060500385 0.074108045 -0.01253734 -0.106078815  
## yr\_built 0.203625141 -0.158744753 -0.11461874 0.302660377  
## yr\_renovated 1.000000000 0.008297188 -0.12265140 0.017208475  
## zipcode 0.008297188 1.000000000 0.06202073 -0.732507241  
## lat -0.122651397 0.062020729 1.00000000 -0.117978027  
## long 0.017208475 -0.732507241 -0.11797803 1.000000000  
## sqft\_living15 -0.127497172 -0.071576283 0.14868862 0.019900460  
## sqft\_lot15 -0.067311184 -0.123764285 -0.13204537 0.302505709  
## sqft\_living15 sqft\_lot15  
## price 0.60537395 -0.061483575  
## bedrooms 0.17429231 0.033645111  
## bathrooms 0.40790720 -0.017857722  
## sqft\_living 0.60139190 0.122075769  
## sqft\_lot 0.22492989 0.750970784  
## floors 0.24849214 -0.041805923  
## waterfront 0.22177584 0.008134168  
## view 0.42912193 -0.028301938  
## condition -0.01988907 0.095022706  
## grade 0.51504840 -0.025336255  
## sqft\_above 0.47820059 0.164193835  
## sqft\_basement 0.31203228 -0.025276599  
## yr\_built -0.01397897 0.035314614  
## yr\_renovated -0.12749717 -0.067311184  
## zipcode -0.07157628 -0.123764285  
## lat 0.14868862 -0.132045371  
## long 0.01990046 0.302505709  
## sqft\_living15 1.00000000 0.270061898  
## sqft\_lot15 0.27006190 1.000000000

# USING BORUTA for features selection and finally making model with the selected attributes

set.seed(1)  
install.packages("Boruta", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Geeta/Documents/R/win-library/3.3'  
## (as 'lib' is unspecified)

## package 'Boruta' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\Geeta\AppData\Local\Temp\RtmpumKjJk\downloaded\_packages

library(Boruta)

## Loading required package: ranger

boruta.train <- Boruta(price ~., data = houseData , doTrace = 2,ntree = 500)

## 1. run of importance source...

## 2. run of importance source...

## 3. run of importance source...

## 4. run of importance source...

## 5. run of importance source...

## 6. run of importance source...

## 7. run of importance source...

## 8. run of importance source...

## 9. run of importance source...

## 10. run of importance source...

## 11. run of importance source...

## After 11 iterations, +1.4 secs:

## confirmed 11 attributes: bathrooms, grade, lat, long, sqft\_above and 6 more;

## rejected 3 attributes: bedrooms, condition, yr\_built;

## still have 4 attributes left.

## 12. run of importance source...

## 13. run of importance source...

## 14. run of importance source...

## 15. run of importance source...

## 16. run of importance source...

## 17. run of importance source...

## 18. run of importance source...

## 19. run of importance source...

## 20. run of importance source...

## 21. run of importance source...

## 22. run of importance source...

## 23. run of importance source...

## 24. run of importance source...

## 25. run of importance source...

## 26. run of importance source...

## 27. run of importance source...

## 28. run of importance source...

## 29. run of importance source...

## 30. run of importance source...

## 31. run of importance source...

## 32. run of importance source...

## 33. run of importance source...

## 34. run of importance source...

## 35. run of importance source...

## 36. run of importance source...

## 37. run of importance source...

## 38. run of importance source...

## 39. run of importance source...

## 40. run of importance source...

## 41. run of importance source...

## 42. run of importance source...

## 43. run of importance source...

## 44. run of importance source...

## 45. run of importance source...

## 46. run of importance source...

## 47. run of importance source...

## 48. run of importance source...

## 49. run of importance source...

## 50. run of importance source...

## 51. run of importance source...

## 52. run of importance source...

## 53. run of importance source...

## 54. run of importance source...

## 55. run of importance source...

## 56. run of importance source...

## 57. run of importance source...

## 58. run of importance source...

## 59. run of importance source...

## 60. run of importance source...

## 61. run of importance source...

## 62. run of importance source...

## 63. run of importance source...

## 64. run of importance source...

## 65. run of importance source...

## 66. run of importance source...

## 67. run of importance source...

## 68. run of importance source...

## 69. run of importance source...

## 70. run of importance source...

## 71. run of importance source...

## 72. run of importance source...

## 73. run of importance source...

## 74. run of importance source...

## 75. run of importance source...

## 76. run of importance source...

## 77. run of importance source...

## 78. run of importance source...

## 79. run of importance source...

## 80. run of importance source...

## 81. run of importance source...

## 82. run of importance source...

## 83. run of importance source...

## 84. run of importance source...

## 85. run of importance source...

## 86. run of importance source...

## 87. run of importance source...

## 88. run of importance source...

## 89. run of importance source...

## 90. run of importance source...

## 91. run of importance source...

## 92. run of importance source...

## 93. run of importance source...

## 94. run of importance source...

## 95. run of importance source...

## 96. run of importance source...

## 97. run of importance source...

## After 97 iterations, +11 secs:

## rejected 1 attribute: sqft\_lot15;

## still have 3 attributes left.

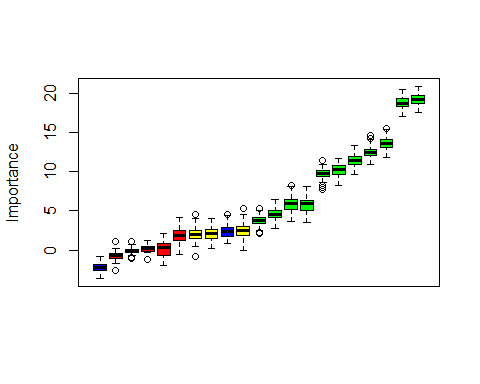
## 98. run of importance source...

## 99. run of importance source...

print(boruta.train)

## Boruta performed 99 iterations in 10.86362 secs.  
## 11 attributes confirmed important: bathrooms, grade, lat, long,  
## sqft\_above and 6 more;  
## 4 attributes confirmed unimportant: bedrooms, condition,  
## sqft\_lot15, yr\_built;  
## 3 tentative attributes left: floors, sqft\_lot, yr\_renovated;

plot(boruta.train, xlab = "", xaxt = "n")



Boruta.Short <- Boruta(price ~ ., data = houseData, maxRuns = 12)

# Start calculating RMSE

houseData$date <- NULL  
for(i in seq(from=0.6, to=0.9, by=0.05)){  
 rn\_train <- sample(nrow(houseData),floor(nrow(houseData)\*i))  
 train <- houseData[rn\_train,colnames(houseData)]  
 test <- houseData[-rn\_train,colnames(houseData)]  
 lm <- lm(formula = price~.,data=train)  
 prediction <- predict(lm,newdata = test)  
 training\_data\_prediction = fitted(lm)  
 training\_rmse = sqrt(sum((training\_data\_prediction-train$price)^2)/nrow(train))  
 testing\_rmse = sqrt(sum((prediction - test$price)^2)/nrow(test))  
 cat("\ntraining\_rmse:",training\_rmse)  
 cat("\ntesting\_rmse:",testing\_rmse)  
 print(lm)  
}

## Warning in predict.lm(lm, newdata = test): prediction from a rank-deficient  
## fit may be misleading

##   
## training\_rmse: 146525.5  
## testing\_rmse: 190020.8  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bedrooms bathrooms sqft\_living sqft\_lot   
## 8.986e+07 -6.178e+02 3.245e+04 1.006e+02 -1.271e+01   
## floors waterfront view condition grade   
## -3.961e+04 2.028e+05 7.107e+04 4.382e+03 1.519e+05   
## sqft\_above sqft\_basement yr\_built yr\_renovated zipcode   
## 3.913e+01 NA -3.538e+03 5.427e+03 -1.562e+03   
## lat long sqft\_living15 sqft\_lot15   
## 6.622e+05 -2.206e+05 1.185e+02 -4.570e+00

## Warning in predict.lm(lm, newdata = test): prediction from a rank-deficient  
## fit may be misleading

##   
## training\_rmse: 167349.9  
## testing\_rmse: 159190.7  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bedrooms bathrooms sqft\_living sqft\_lot   
## 9.455e+07 1.975e+04 -4.718e+04 1.296e+02 -5.035e+00   
## floors waterfront view condition grade   
## 3.350e+04 2.823e+05 7.689e+04 3.809e+04 1.454e+05   
## sqft\_above sqft\_basement yr\_built yr\_renovated zipcode   
## 4.756e+01 NA -3.621e+03 7.011e+03 -2.039e+03   
## lat long sqft\_living15 sqft\_lot15   
## 7.089e+05 -5.215e+05 4.877e+01 1.001e-01

## Warning in predict.lm(lm, newdata = test): prediction from a rank-deficient  
## fit may be misleading

##   
## training\_rmse: 147874.3  
## testing\_rmse: 193766.5  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bedrooms bathrooms sqft\_living sqft\_lot   
## 6.373e+07 6.309e+03 9.792e+03 6.114e+01 2.031e+00   
## floors waterfront view condition grade   
## 3.026e+04 2.502e+05 6.081e+04 1.809e+04 1.558e+05   
## sqft\_above sqft\_basement yr\_built yr\_renovated zipcode   
## 4.732e+01 NA -2.211e+03 5.103e+03 -1.172e+03   
## lat long sqft\_living15 sqft\_lot15   
## 6.710e+05 -1.013e+05 1.094e+02 -1.126e+01

## Warning in predict.lm(lm, newdata = test): prediction from a rank-deficient  
## fit may be misleading

##   
## training\_rmse: 152773.2  
## testing\_rmse: 196597.1  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bedrooms bathrooms sqft\_living sqft\_lot   
## 8.144e+07 -1.593e+04 7.107e+03 8.052e+01 2.260e+00   
## floors waterfront view condition grade   
## 6.538e+04 3.165e+05 4.495e+04 3.652e+04 1.929e+05   
## sqft\_above sqft\_basement yr\_built yr\_renovated zipcode   
## 3.723e+01 NA -5.482e+03 5.881e+03 -1.275e+03   
## lat long sqft\_living15 sqft\_lot15   
## 7.489e+05 -4.613e+04 5.911e+01 -6.497e+00

## Warning in predict.lm(lm, newdata = test): prediction from a rank-deficient  
## fit may be misleading

##   
## training\_rmse: 151298.8  
## testing\_rmse: 194196.8  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bedrooms bathrooms sqft\_living sqft\_lot   
## 8.422e+07 -1.534e+04 -1.807e+03 8.983e+01 -4.048e+00   
## floors waterfront view condition grade   
## 3.588e+03 3.626e+05 6.751e+04 5.878e+04 1.566e+05   
## sqft\_above sqft\_basement yr\_built yr\_renovated zipcode   
## 6.487e+01 NA -2.666e+03 6.348e+03 -1.646e+03   
## lat long sqft\_living15 sqft\_lot15   
## 6.093e+05 -3.242e+05 7.865e+01 -2.809e+00

## Warning in predict.lm(lm, newdata = test): prediction from a rank-deficient  
## fit may be misleading

##   
## training\_rmse: 166470.5  
## testing\_rmse: 119726.2  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bedrooms bathrooms sqft\_living sqft\_lot   
## 7.691e+07 3.896e+03 6.577e+03 8.012e+01 -5.036e+00   
## floors waterfront view condition grade   
## 7.737e+02 2.757e+05 6.598e+04 5.177e+04 1.582e+05   
## sqft\_above sqft\_basement yr\_built yr\_renovated zipcode   
## 4.429e+01 NA -3.276e+03 5.811e+03 -1.556e+03   
## lat long sqft\_living15 sqft\_lot15   
## 7.461e+05 -2.771e+05 9.252e+01 -3.197e+00

## Warning in predict.lm(lm, newdata = test): prediction from a rank-deficient  
## fit may be misleading

##   
## training\_rmse: 159046.5  
## testing\_rmse: 178191.4  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bedrooms bathrooms sqft\_living sqft\_lot   
## 9.778e+07 -3.031e+04 1.254e+04 1.133e+02 -3.342e+00   
## floors waterfront view condition grade   
## 3.063e+04 1.703e+05 6.363e+04 5.237e+04 1.541e+05   
## sqft\_above sqft\_basement yr\_built yr\_renovated zipcode   
## 4.158e+01 NA -3.217e+03 5.954e+03 -1.927e+03   
## lat long sqft\_living15 sqft\_lot15   
## 5.667e+05 -4.711e+05 6.421e+01 -3.907e+00