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

analysis <- function(HousePrice, i, labels){  
 plot(HousePrice$price~HousePrice[,i],main = labels[1],xlab = labels[2],ylab = labels[3],col.main='Blue',col.axis='Gray',col.lab = 'Gray',col = 'Gray')  
   
 hist(HousePrice[,i],main = labels[1],xlab = labels[2],ylab = labels[3],col.main='Blue',col.axis='Gray',col.lab = 'Gray',col = 'Gray')  
  
 boxplot(HousePrice[,i],main = labels[1],col.main='Blue',col.axis='Gray',col.lab = 'Gray',col = 'Gray')  
   
 cor(HousePrice[,i],HousePrice$price)  
}

# Data Importing And Cleaning

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

## 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

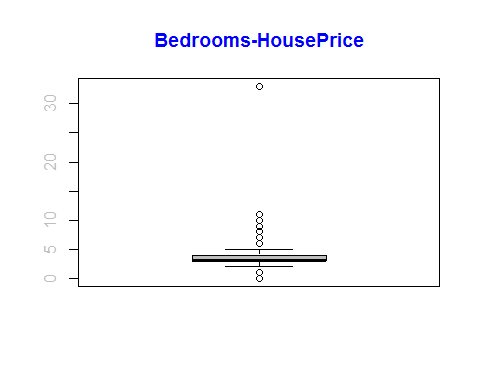
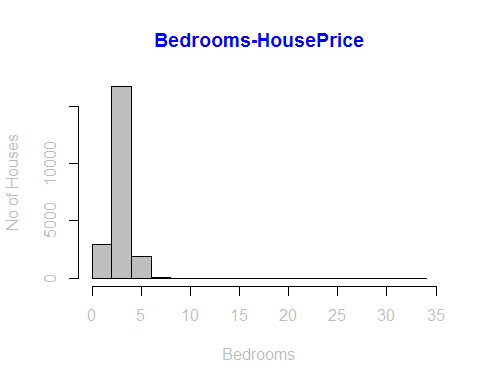
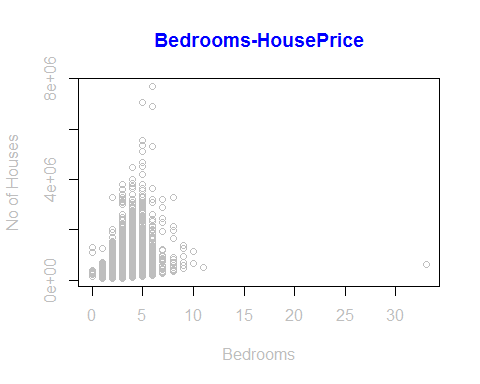
colnames(HousePrice)

## [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(HousePrice)  
if(length(naColumns)>0){  
 cat("Found NA Colums:")  
 for(i in 1:length(naColumns)) {  
 cat(sprintf("%s,", colnames(HousePrice)[i]))  
 }  
}  
# Here we conclude that this data does not hold any column with NA.

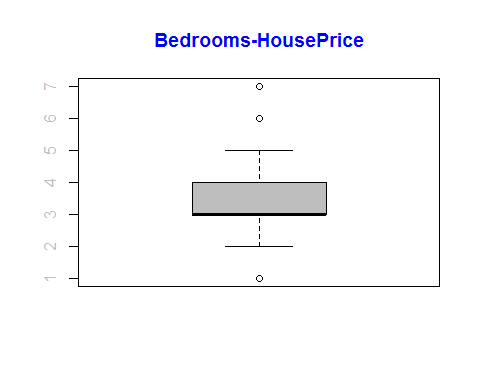
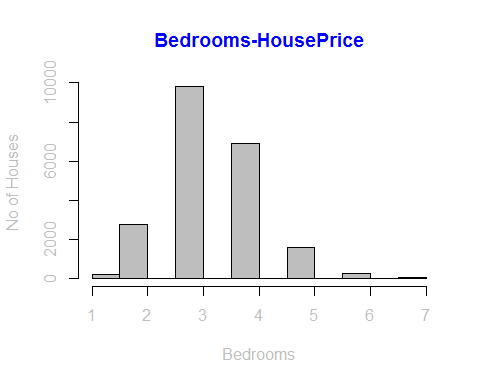
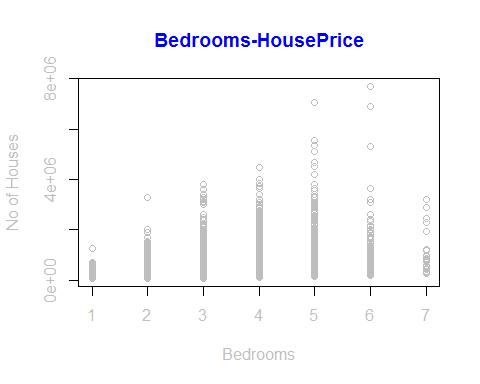
# BEDROOM ANALYSIS

analysis(HousePrice,4,c('Bedrooms-HousePrice','Bedrooms', 'No of Houses'))



## [1] 0.3083496

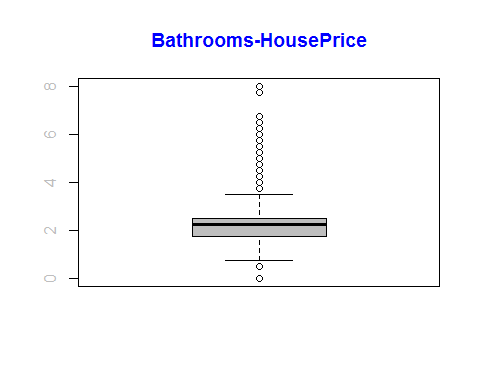
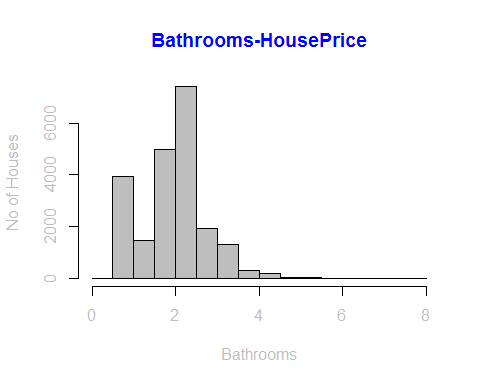
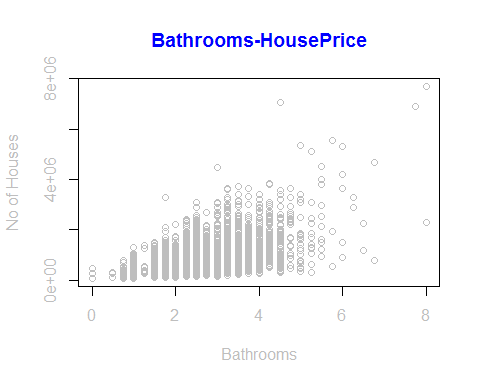
#\*\*\*\*\*\*\*Removing the outliers  
#Since more than 7 bedrooms are very rare.Also it's the outlier for my model.  
#I have removed the outlier data.  
HousePrice<-subset(HousePrice,bedrooms>=1 & bedrooms<=7)  
#\*\*\*\*\*\*\*Once we removed the outliers, again get the analysis  
analysis(HousePrice,4,c('Bedrooms-HousePrice','Bedrooms', 'No of Houses'))



## [1] 0.3156734

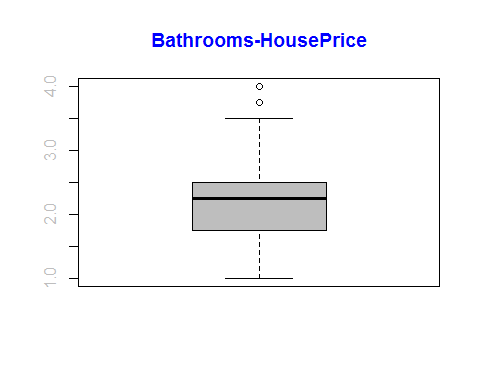
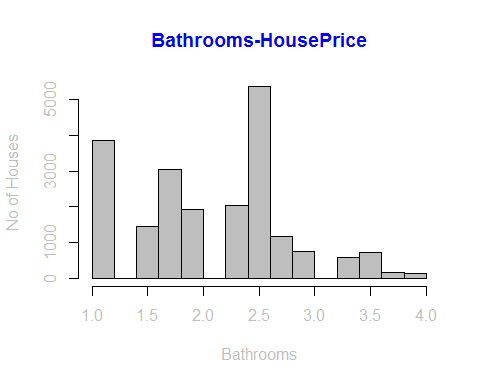
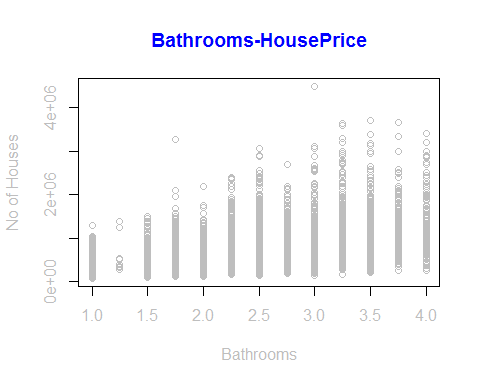
# BATHROOM ANALYSIS

analysis(HousePrice,5,c('Bathrooms-HousePrice','Bathrooms', 'No of Houses'))



## [1] 0.5259342

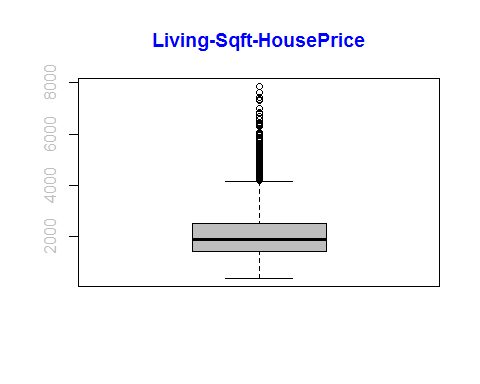
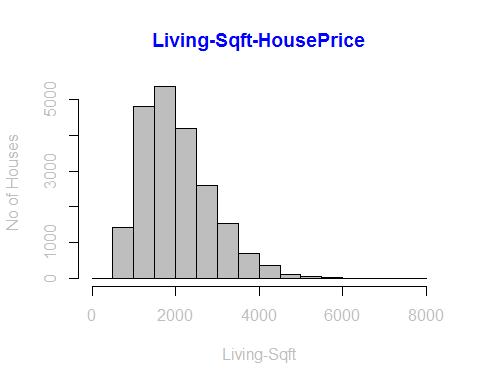
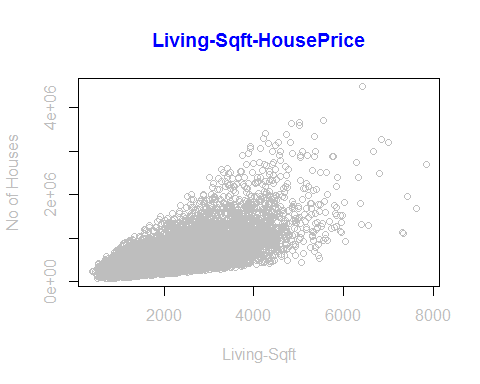
#\*\*\*\*\*\*\*Removing the outliers  
#More than 4 bathrooms are very rare in this data.So I am removing it.  
HousePrice<-subset(HousePrice,bathrooms>=1 & bathrooms<=4)  
analysis(HousePrice,5,c('Bathrooms-HousePrice','Bathrooms', 'No of Houses'))



## [1] 0.475159

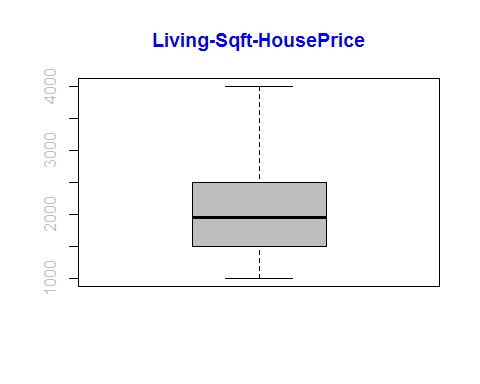
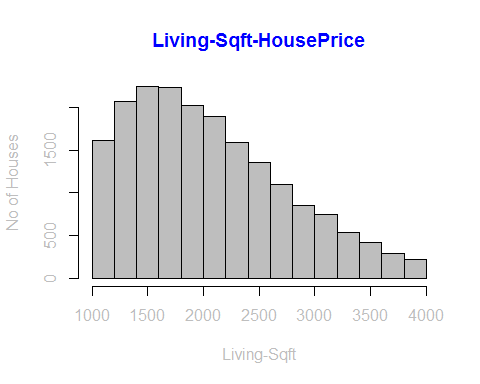
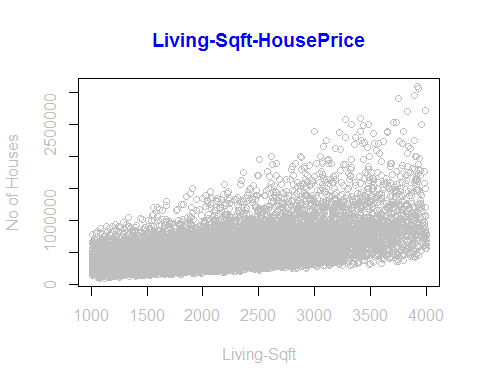
# SQFT LIVING ANALYSIS

analysis(HousePrice,6,c('Living-Sqft-HousePrice','Living-Sqft', 'No of Houses'))



## [1] 0.6701029

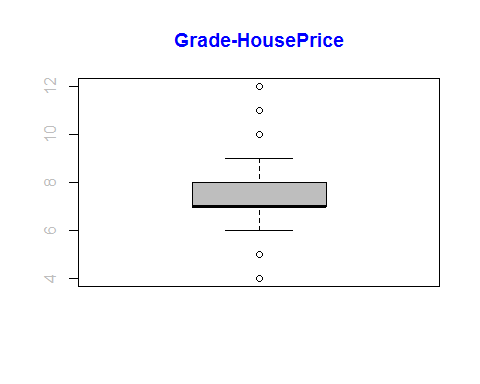
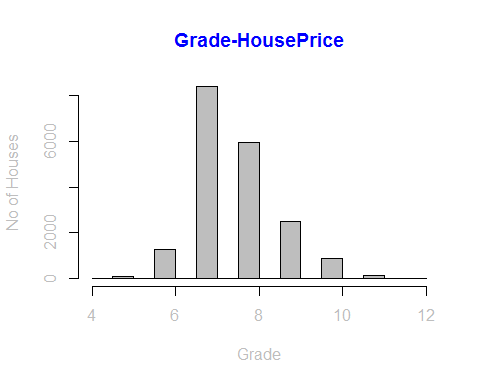
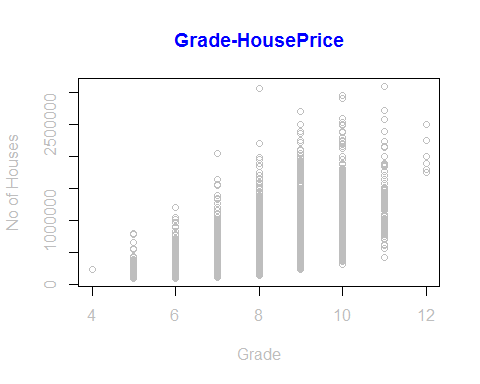
#\*\*\*\*\*\*\*Removing the outliers  
HousePrice<-subset(HousePrice,sqft\_living >1000 & sqft\_living<=4000)  
analysis(HousePrice,6,c('Living-Sqft-HousePrice','Living-Sqft', 'No of Houses'))



## [1] 0.5938015

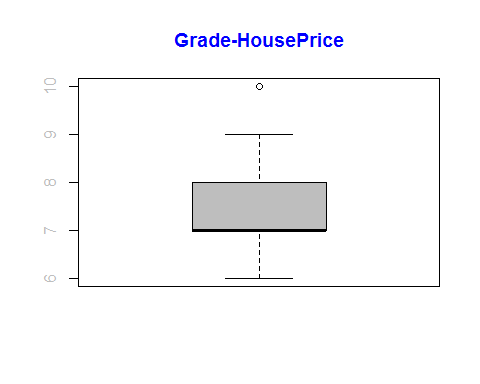
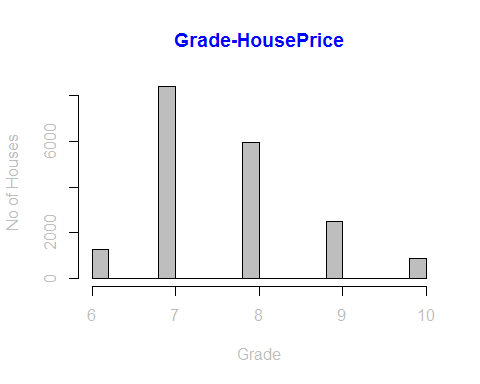
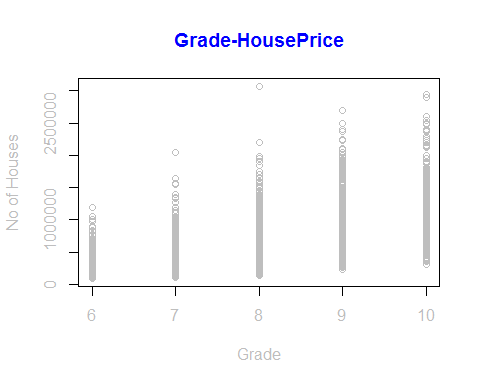
## GRADE ANALYSIS

analysis(HousePrice,12,c('Grade-HousePrice','Grade', 'No of Houses'))



## [1] 0.6106929

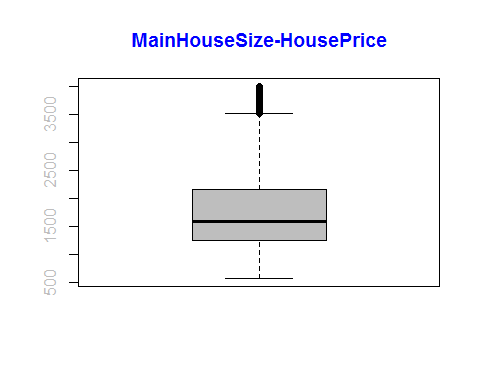
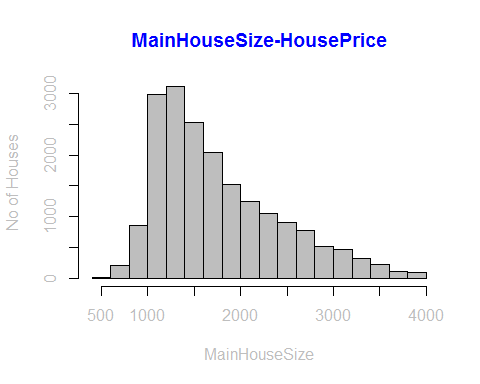
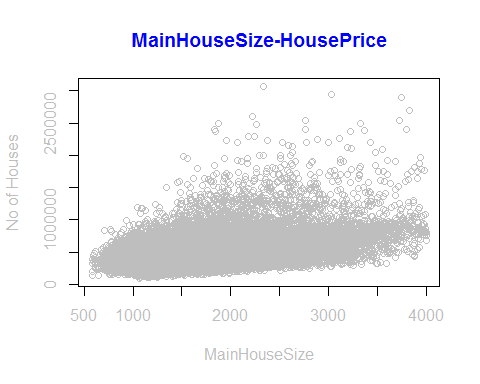
#\*\*\*\*\*\*\*Removing the outliers  
#Most of the houses grades are between 6-10   
HousePrice<-subset(HousePrice,grade >= 6 & grade<=10)  
analysis(HousePrice,12,c('Grade-HousePrice','Grade', 'No of Houses'))



## [1] 0.592697

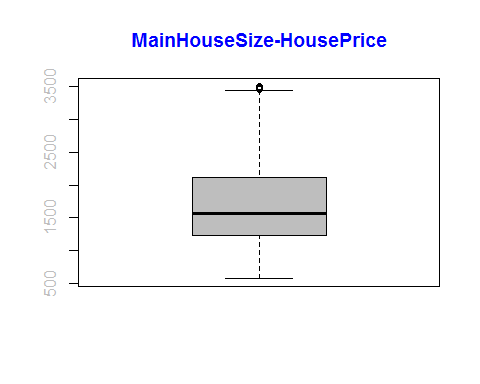
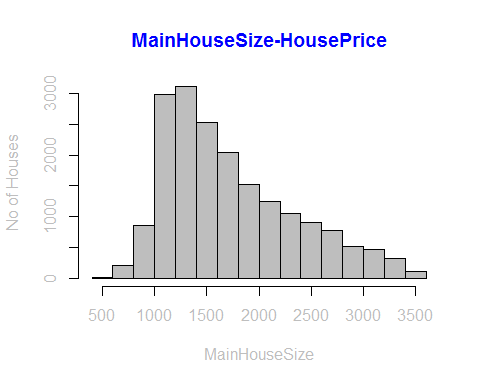
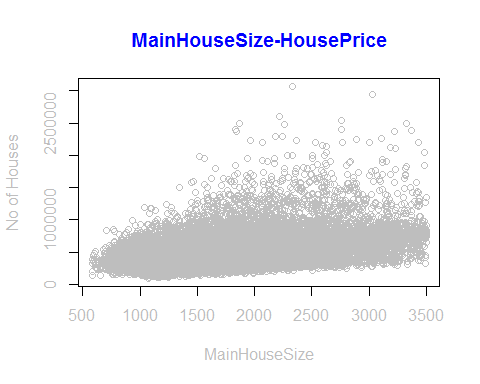
# SQFT\_ABOVE ANALYSIS

analysis(HousePrice,13,c('MainHouseSize-HousePrice','MainHouseSize', 'No of Houses'))



## [1] 0.4553313

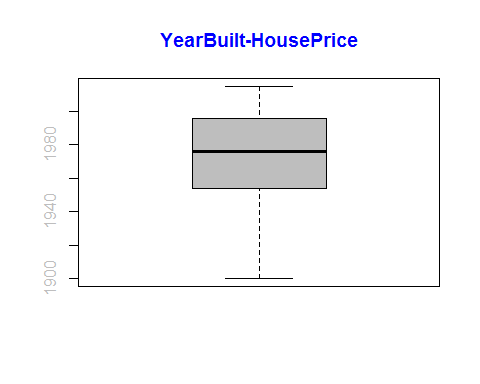
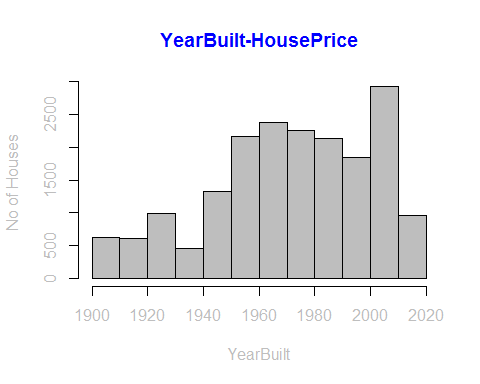
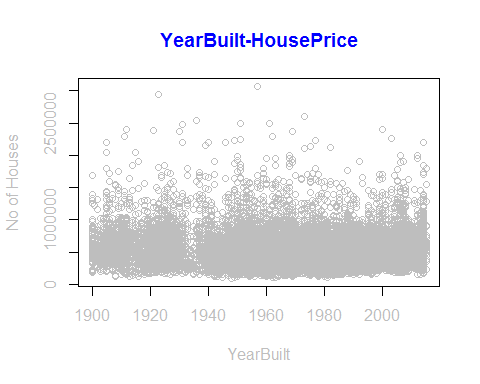
#\*\*\*\*\*\*\*Removing the outliers  
HousePrice<-subset(HousePrice,sqft\_above >=500 & sqft\_above<=3500)  
analysis(HousePrice,13,c('MainHouseSize-HousePrice','MainHouseSize', 'No of Houses'))



## [1] 0.4254812

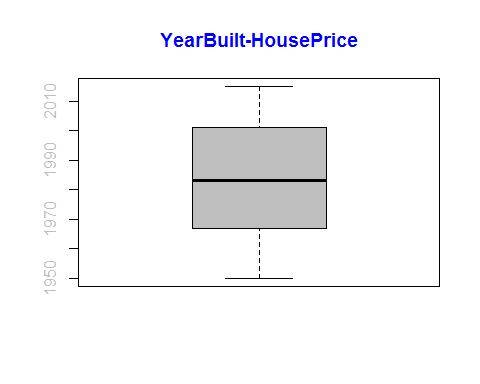
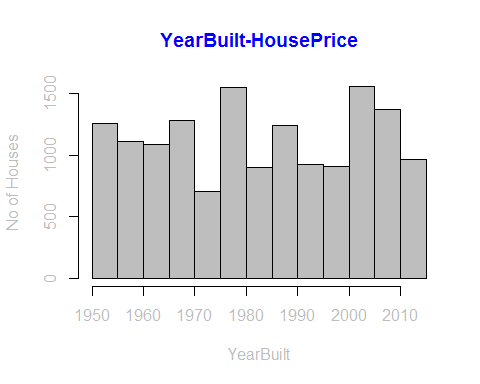
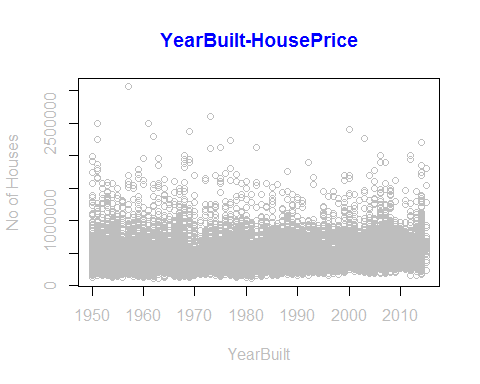
# YR\_BUILT ANALYSIS

analysis(HousePrice,15,c('YearBuilt-HousePrice','YearBuilt', 'No of Houses'))



## [1] -0.07805272

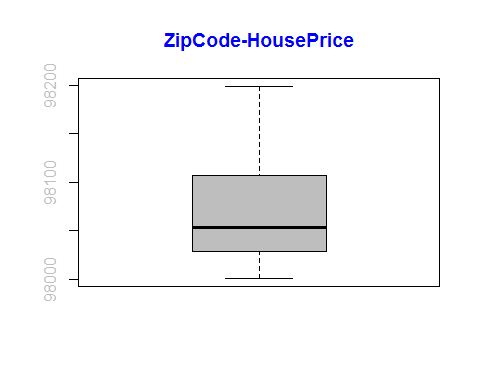
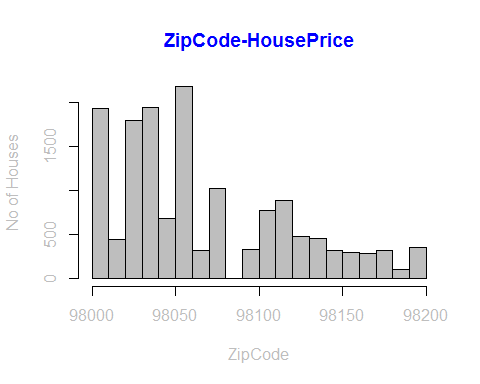
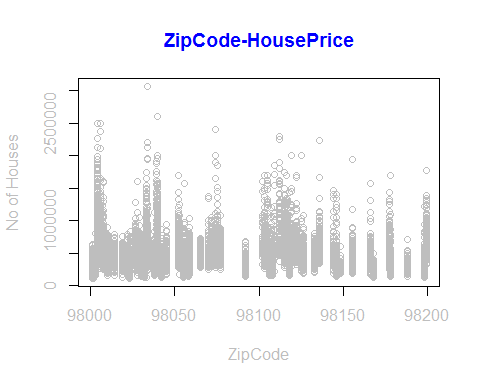
#\*\*\*\*\*\*\*Removing the outliers  
#In our data some records are too old..I just removed that data from my model.  
#Because It doesn't make any sense to keep more than 100 years house in our model  
HousePrice<-subset(HousePrice,yr\_built>=1950& yr\_built<=2015)  
analysis(HousePrice,15,c('YearBuilt-HousePrice','YearBuilt', 'No of Houses'))



## [1] 0.08555947

# ZIPCODE ANALYSIS

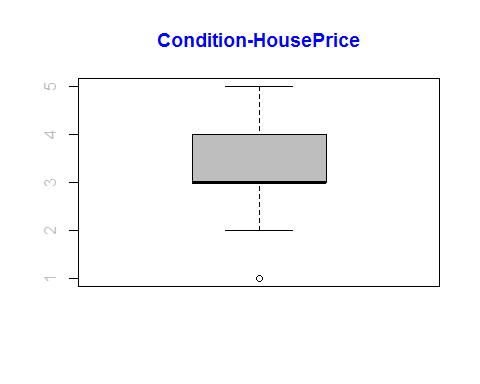
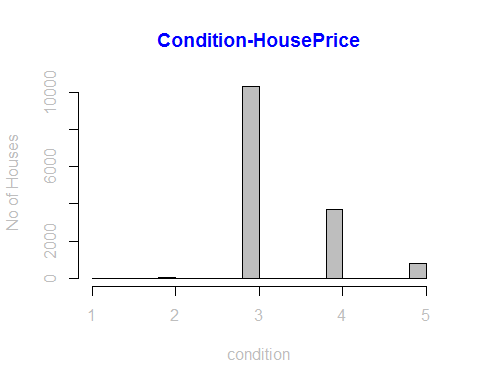
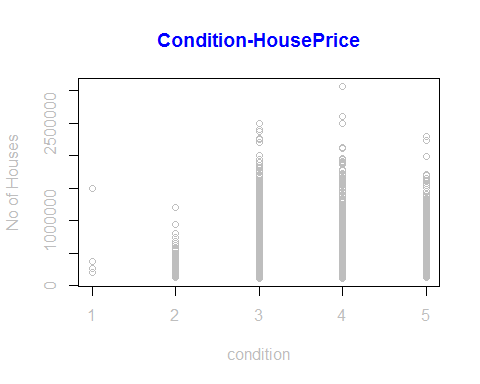
analysis(HousePrice,17,c('ZipCode-HousePrice','ZipCode', 'No of Houses'))



## [1] -0.02647096

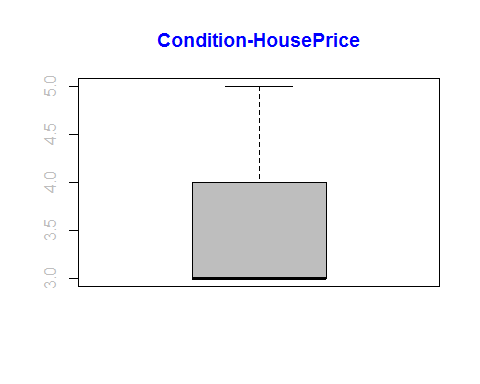
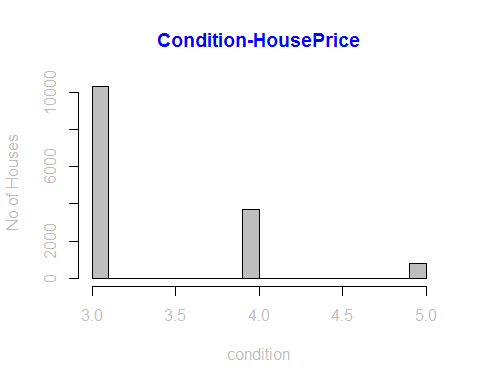
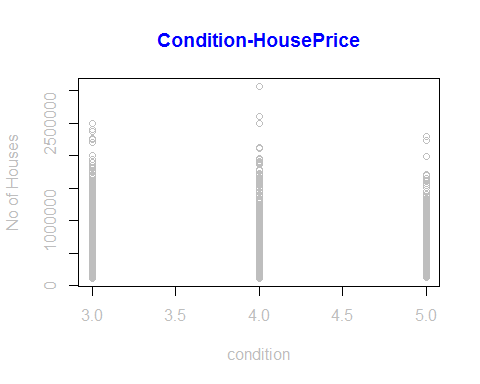
## CONDITION aNALYSIS

analysis(HousePrice,11,c('Condition-HousePrice','condition', 'No of Houses'))



## [1] 0.0300221

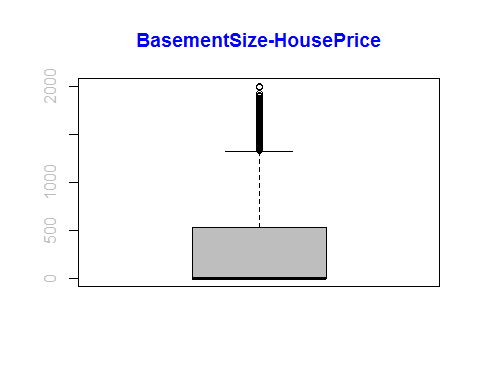
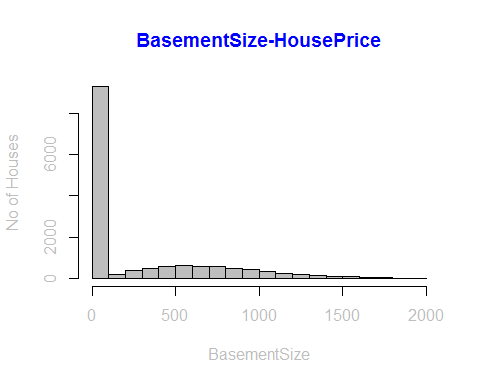
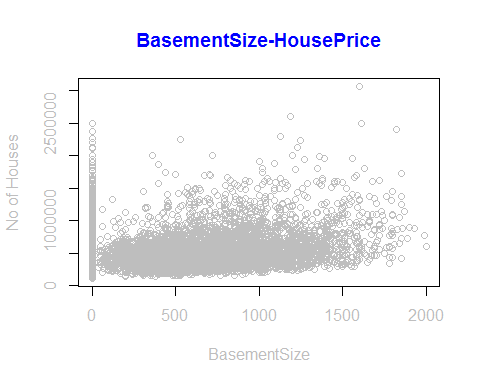
#\*\*\*\*\*\*\*Removing the outliers  
HousePrice<-subset(HousePrice,condition>=3& condition<=5)  
analysis(HousePrice,11,c('Condition-HousePrice','condition', 'No of Houses'))



## [1] 0.02442002

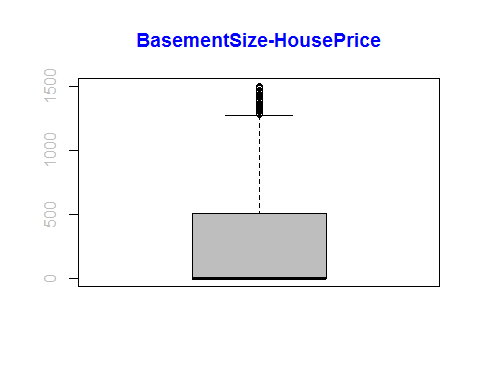
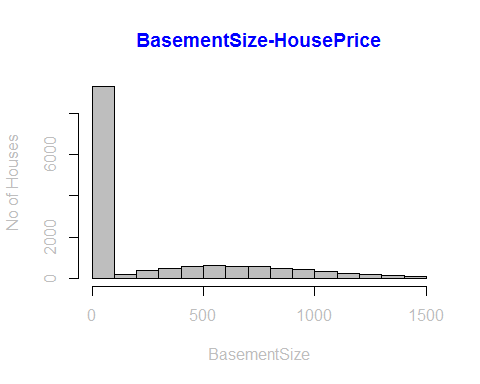
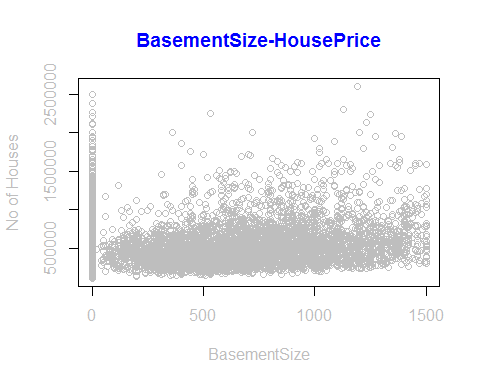
## SQFT\_BASEMENT ANALYSIS

analysis(HousePrice,14,c('BasementSize-HousePrice','BasementSize', 'No of Houses'))



## [1] 0.2254519

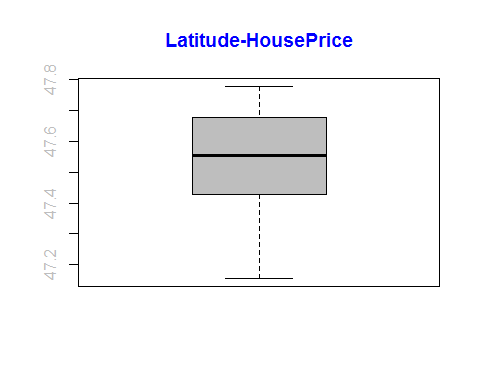
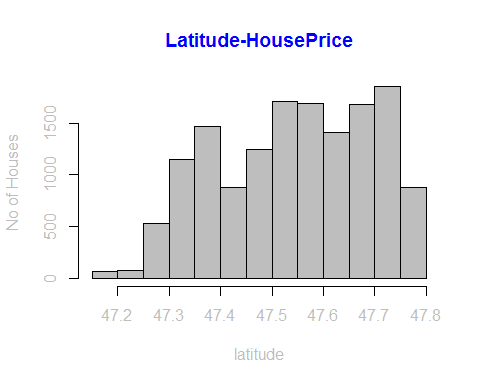
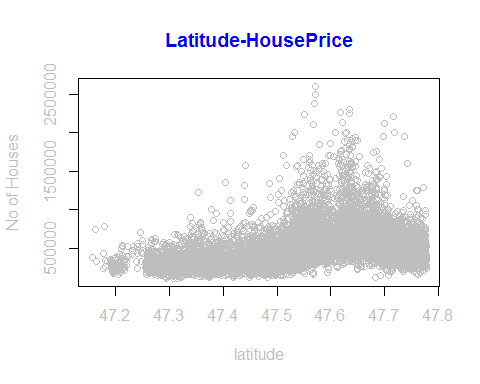
#\*\*\*\*\*\*\*Removing the outliers  
HousePrice<-subset(HousePrice,sqft\_basement >=0 & sqft\_basement<=1500)  
analysis(HousePrice,14,c('BasementSize-HousePrice','BasementSize', 'No of Houses'))



## [1] 0.1879221

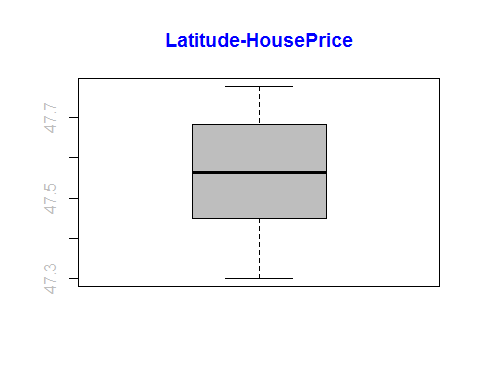
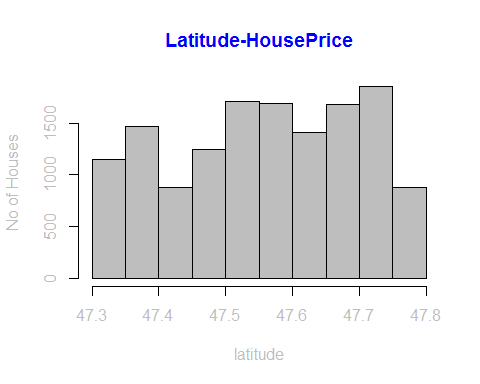
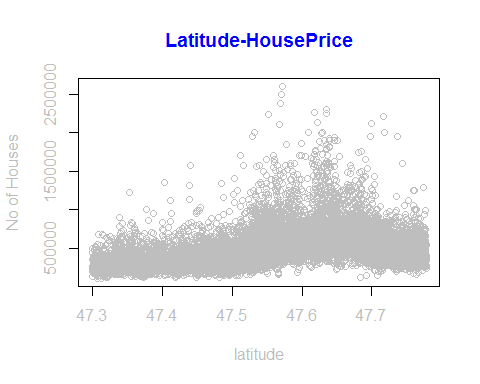
## LAT ANALYSIS

analysis(HousePrice,18,c('Latitude-HousePrice','latitude', 'No of Houses'))



## [1] 0.4161934

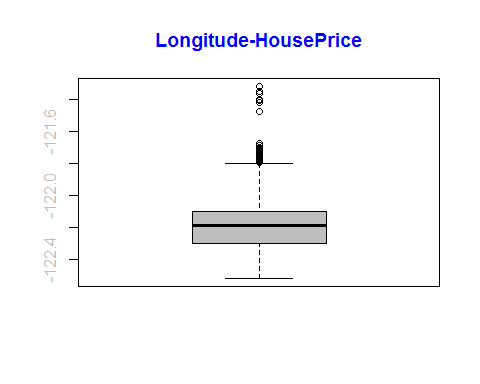
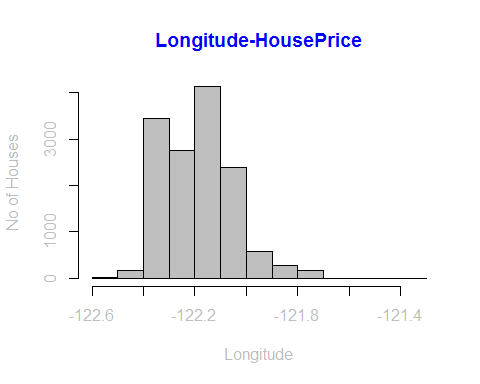
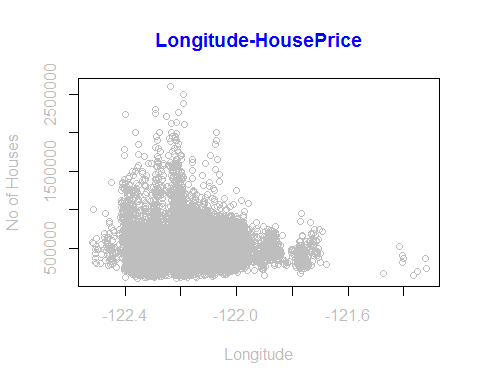
#\*\*\*\*\*\*\*Removing the outliers  
HousePrice<-subset(HousePrice,lat>=47.3)  
analysis(HousePrice,18,c('Latitude-HousePrice','latitude', 'No of Houses'))



## [1] 0.3848454

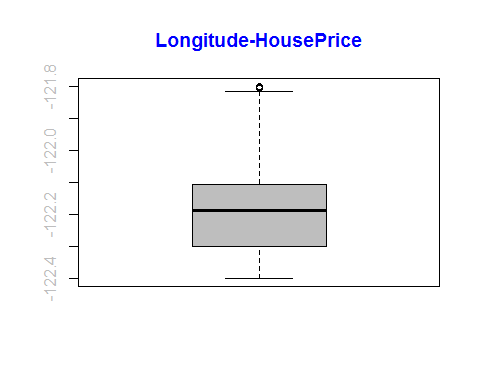
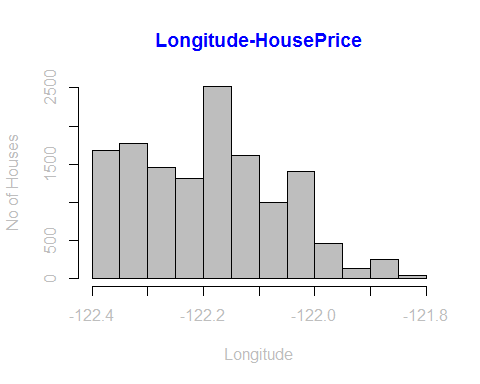
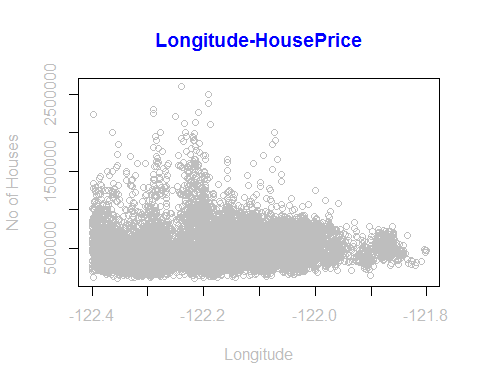
## LONG ANALYSIS

analysis(HousePrice,19,c('Longitude-HousePrice','Longitude', 'No of Houses'))



## [1] 0.03740839

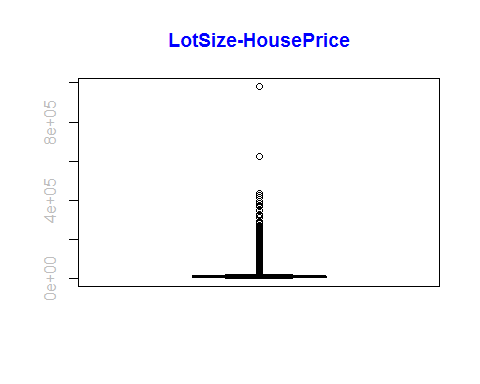
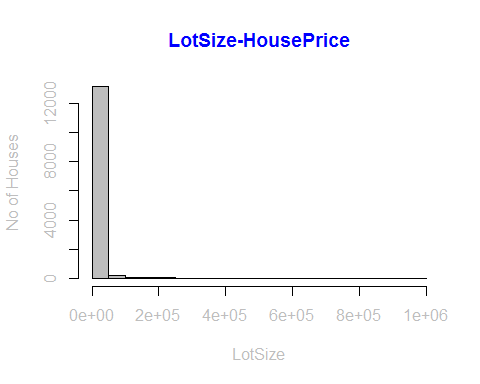
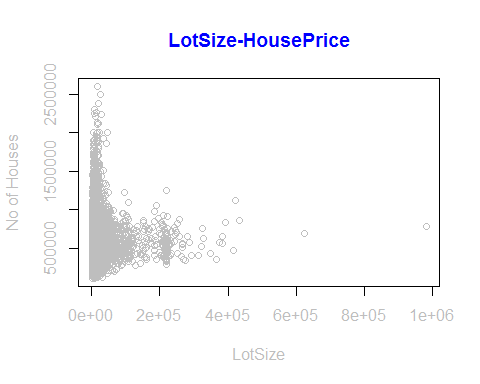
#\*\*\*\*\*\*\*Removing the outliers  
HousePrice<-subset(HousePrice,long>=-122.4 & long < -121.8)  
analysis(HousePrice,19,c('Longitude-HousePrice','Longitude', 'No of Houses'))



## [1] 0.0746599

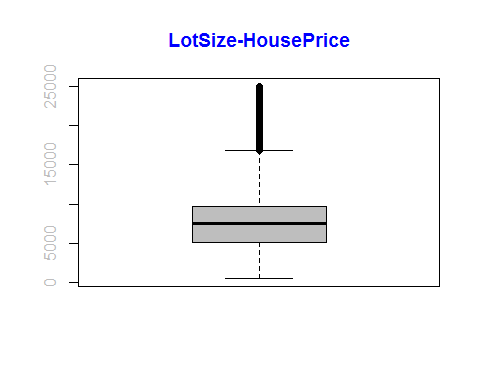
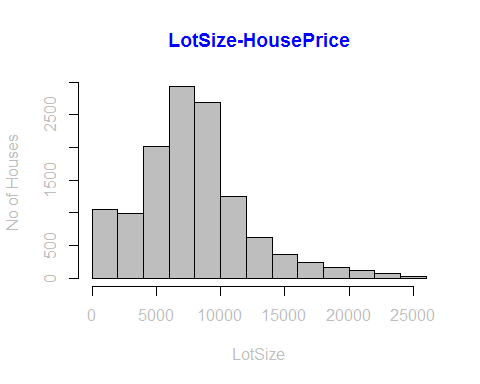
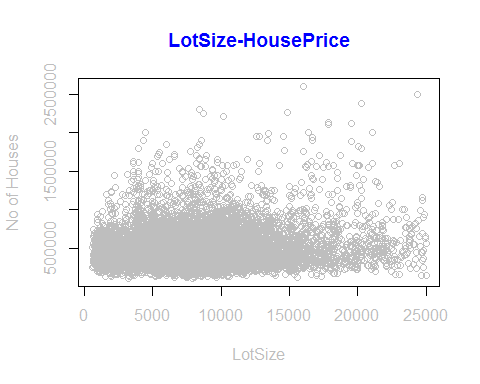
## SQFT\_LOT ANALYSIS

analysis(HousePrice,7,c('LotSize-HousePrice','LotSize', 'No of Houses'))



## [1] 0.09390182

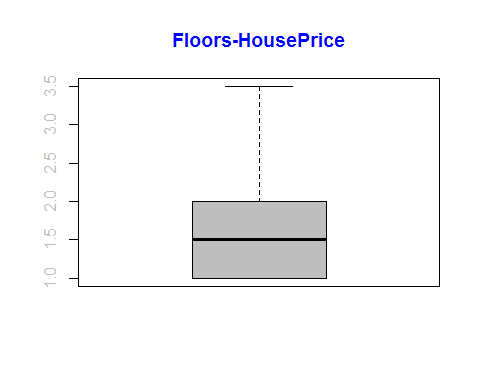
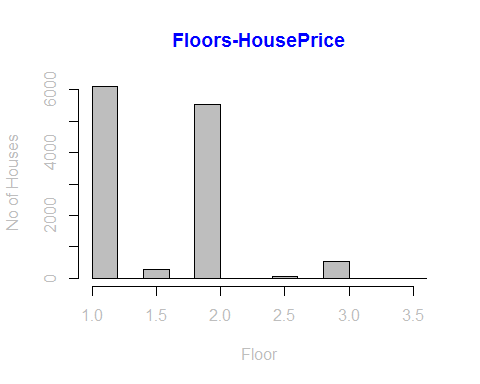
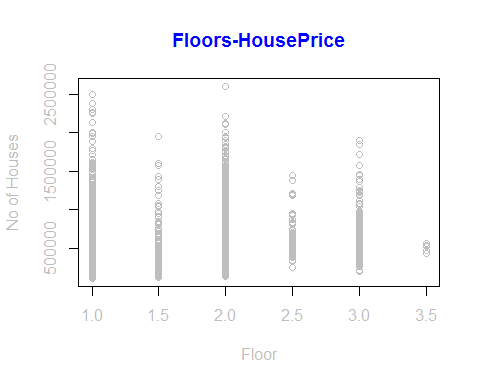
#\*\*\*\*\*\*\*Removing the outliers  
HousePrice<-subset(HousePrice,sqft\_lot>=0 & sqft\_lot<=25000)  
analysis(HousePrice,7,c('LotSize-HousePrice','LotSize', 'No of Houses'))



## [1] 0.1228599

## FLOOR ANALYSIS

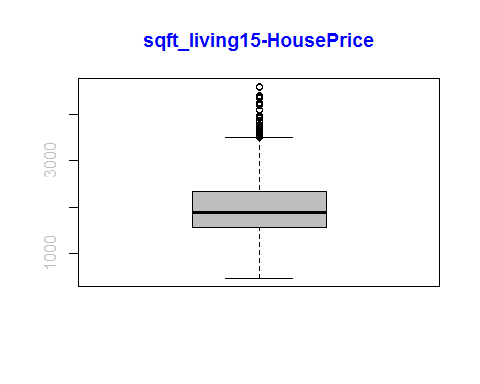
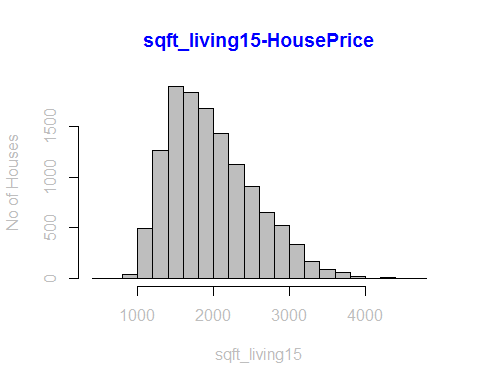
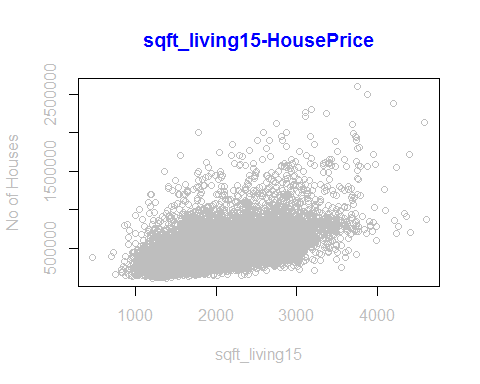
analysis(HousePrice,8,c('Floors-HousePrice','Floor', 'No of Houses'))



## [1] 0.189869

## SQFT\_LIVING15 ANALYSIS

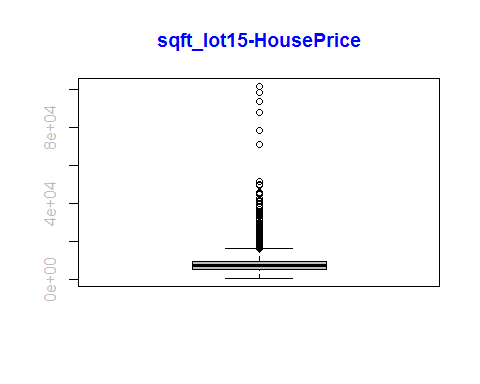
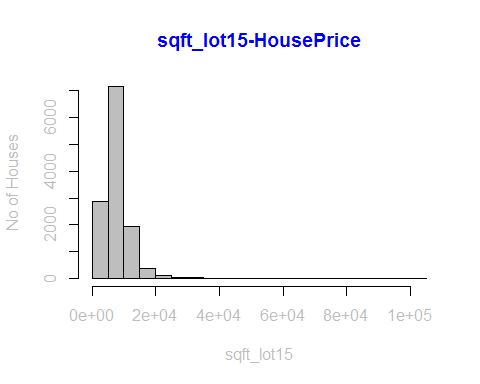
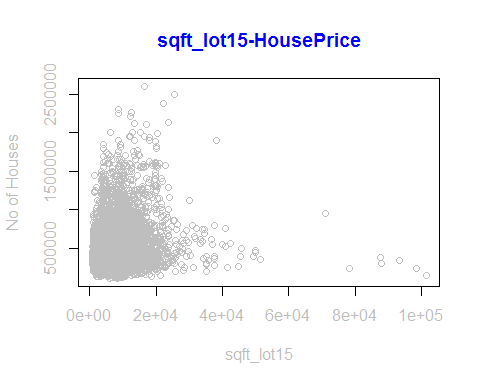
analysis(HousePrice,20,c('sqft\_living15-HousePrice','sqft\_living15', 'No of Houses'))



## [1] 0.5377631

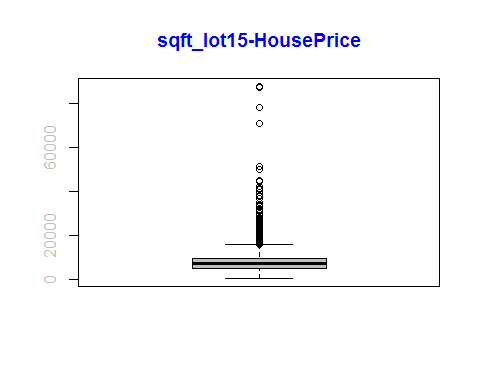
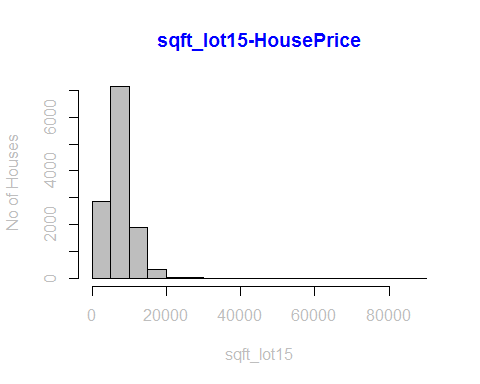
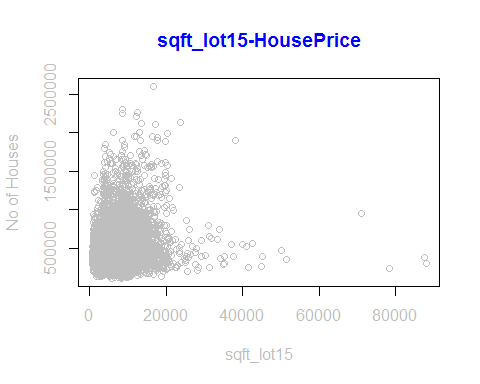
## SQFT\_LOT15 ANALYSIS

analysis(HousePrice,21,c('sqft\_lot15-HousePrice','sqft\_lot15', 'No of Houses'))



## [1] 0.1155467

#\*\*\*\*\*\*\*Removing the outliers  
HousePrice<-subset(HousePrice,sqft\_lot15>=0 & sqft\_lot<=20000)  
analysis(HousePrice,21,c('sqft\_lot15-HousePrice','sqft\_lot15', 'No of Houses'))



## [1] 0.1046768

## Correlation among all the variables

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

## id price bedrooms bathrooms sqft\_living  
## id 1.00000000 0.03088511 -0.00248828 0.05248387 0.05591911  
## price 0.03088511 1.00000000 0.21316110 0.40694539 0.58753189  
## bedrooms -0.00248828 0.21316110 1.00000000 0.35967366 0.54595425  
## bathrooms 0.05248387 0.40694539 0.35967366 1.00000000 0.61084492  
## sqft\_living 0.05591911 0.58753189 0.54595425 0.61084492 1.00000000  
## sqft\_lot -0.04881811 0.09879146 0.21379756 -0.15003858 0.21349737  
## floors 0.03142587 0.19599487 -0.01231716 0.48824267 0.21834143  
## waterfront -0.00601867 0.18723046 -0.02388155 0.02402709 0.04512995  
## view 0.04845104 0.32126248 0.04370921 0.10743868 0.20290469  
## condition -0.05442527 0.01080196 0.05075847 -0.20275433 -0.08704616  
## grade 0.06542157 0.62532384 0.17613009 0.51361808 0.62530895  
## sqft\_above 0.06310017 0.47583099 0.37950422 0.52925673 0.80769249  
## sqft\_basement -0.01105019 0.18482790 0.27249273 0.13661508 0.31810283  
## yr\_built 0.07163873 0.11886923 -0.02014115 0.56553240 0.25215473  
## yr\_renovated -0.01141468 0.10260750 0.01761968 0.01698352 0.02780418  
## zipcode -0.03344919 -0.06661544 -0.10383298 -0.08873701 -0.13655332  
## lat -0.01028660 0.39782088 -0.04362736 0.03452169 0.02941721  
## long 0.09490813 0.06021992 0.09157411 0.15541240 0.21179591  
## sqft\_living15 0.07487106 0.53192136 0.34578440 0.44219575 0.73234542  
## sqft\_lot15 -0.05544805 0.10467684 0.16855680 -0.13671006 0.17733937  
## sqft\_lot floors waterfront view condition  
## id -0.04881811 0.03142587 -0.006018670 0.04845104 -0.05442527  
## price 0.09879146 0.19599487 0.187230461 0.32126248 0.01080196  
## bedrooms 0.21379756 -0.01231716 -0.023881549 0.04370921 0.05075847  
## bathrooms -0.15003858 0.48824267 0.024027088 0.10743868 -0.20275433  
## sqft\_living 0.21349737 0.21834143 0.045129953 0.20290469 -0.08704616  
## sqft\_lot 1.00000000 -0.46764775 0.061281460 0.10171693 0.27240654  
## floors -0.46764775 1.00000000 0.020419644 -0.01755120 -0.36800514  
## waterfront 0.06128146 0.02041964 1.000000000 0.34042155 0.01009095  
## view 0.10171693 -0.01755120 0.340421549 1.00000000 0.03451890  
## condition 0.27240654 -0.36800514 0.010090951 0.03451890 1.00000000  
## grade 0.01799467 0.40495046 0.058898054 0.18464079 -0.19021868  
## sqft\_above 0.10812840 0.46766878 0.019014619 0.06247316 -0.19571109  
## sqft\_basement 0.17132458 -0.39896785 0.042393068 0.22760950 0.17395516  
## yr\_built -0.47268086 0.72279773 -0.025271092 -0.08730803 -0.45609638  
## yr\_renovated 0.04836387 -0.03403060 0.050423835 0.06473431 -0.04580786  
## zipcode -0.20648742 0.02202050 0.049153202 0.11697692 -0.09996766  
## lat -0.05925894 0.05026221 -0.007680641 0.01912452 -0.04700562  
## long 0.15246300 0.07677540 -0.010330764 -0.10532559 -0.04109049  
## sqft\_living15 0.26717897 0.16001569 0.063053624 0.22038305 -0.07723497  
## sqft\_lot15 0.80071360 -0.40438723 0.101403098 0.10986952 0.24966404  
## grade sqft\_above sqft\_basement yr\_built  
## id 0.0654215660 0.063100165 -0.0110501872 0.07163873  
## price 0.6253238358 0.475830987 0.1848279021 0.11886923  
## bedrooms 0.1761300924 0.379504224 0.2724927258 -0.02014115  
## bathrooms 0.5136180759 0.529256730 0.1366150794 0.56553240  
## sqft\_living 0.6253089478 0.807692487 0.3181028274 0.25215473  
## sqft\_lot 0.0179946663 0.108128401 0.1713245799 -0.47268086  
## floors 0.4049504567 0.467668775 -0.3989678500 0.72279773  
## waterfront 0.0588980541 0.019014619 0.0423930680 -0.02527109  
## view 0.1846407902 0.062473162 0.2276095048 -0.08730803  
## condition -0.1902186768 -0.195711088 0.1739551648 -0.45609638  
## grade 1.0000000000 0.629292059 -0.0008512015 0.41391909  
## sqft\_above 0.6292920588 1.000000000 -0.3020484503 0.43178138  
## sqft\_basement -0.0008512015 -0.302048450 1.0000000000 -0.28659200  
## yr\_built 0.4139190905 0.431781375 -0.2865920034 1.00000000  
## yr\_renovated 0.0137545251 0.005527641 0.0360665829 -0.14488568  
## zipcode -0.0963141760 -0.207857824 0.1134416369 -0.10071641  
## lat 0.1244168337 -0.030479955 0.0965731536 -0.05718985  
## long 0.1212933902 0.346947946 -0.2154373008 0.25864663  
## sqft\_living15 0.6095840261 0.677956453 0.0939590371 0.21331396  
## sqft\_lot15 0.0242929122 0.088065155 0.1451238319 -0.41247570  
## yr\_renovated zipcode lat long  
## id -0.011414678 -0.03344919 -0.010286601 0.09490813  
## price 0.102607501 -0.06661544 0.397820880 0.06021992  
## bedrooms 0.017619683 -0.10383298 -0.043627362 0.09157411  
## bathrooms 0.016983524 -0.08873701 0.034521688 0.15541240  
## sqft\_living 0.027804177 -0.13655332 0.029417207 0.21179591  
## sqft\_lot 0.048363871 -0.20648742 -0.059258942 0.15246300  
## floors -0.034030600 0.02202050 0.050262205 0.07677540  
## waterfront 0.050423835 0.04915320 -0.007680641 -0.01033076  
## view 0.064734312 0.11697692 0.019124515 -0.10532559  
## condition -0.045807864 -0.09996766 -0.047005621 -0.04109049  
## grade 0.013754525 -0.09631418 0.124416834 0.12129339  
## sqft\_above 0.005527641 -0.20785782 -0.030479955 0.34694795  
## sqft\_basement 0.036066583 0.11344164 0.096573154 -0.21543730  
## yr\_built -0.144885684 -0.10071641 -0.057189850 0.25864663  
## yr\_renovated 1.000000000 0.04008558 0.021904703 -0.03691168  
## zipcode 0.040085583 1.00000000 0.185270334 -0.54354459  
## lat 0.021904703 0.18527033 1.000000000 -0.08224426  
## long -0.036911680 -0.54354459 -0.082244257 1.00000000  
## sqft\_living15 0.002850830 -0.25878003 0.012087716 0.33806073  
## sqft\_lot15 0.060629185 -0.16726839 -0.043272796 0.13205797  
## sqft\_living15 sqft\_lot15  
## id 0.07487106 -0.05544805  
## price 0.53192136 0.10467684  
## bedrooms 0.34578440 0.16855680  
## bathrooms 0.44219575 -0.13671006  
## sqft\_living 0.73234542 0.17733937  
## sqft\_lot 0.26717897 0.80071360  
## floors 0.16001569 -0.40438723  
## waterfront 0.06305362 0.10140310  
## view 0.22038305 0.10986952  
## condition -0.07723497 0.24966404  
## grade 0.60958403 0.02429291  
## sqft\_above 0.67795645 0.08806515  
## sqft\_basement 0.09395904 0.14512383  
## yr\_built 0.21331396 -0.41247570  
## yr\_renovated 0.00285083 0.06062919  
## zipcode -0.25878003 -0.16726839  
## lat 0.01208772 -0.04327280  
## long 0.33806073 0.13205797  
## sqft\_living15 1.00000000 0.25196255  
## sqft\_lot15 0.25196255 1.00000000

# HousePrice regression Model

# I have selected some of the attributes that are important for my model based on the above analysis

# Then I made linear regression model which could best predict the house Prices .

# I took 60% for the training and 40 % for the testing dataset.

set.seed(1)  
newHousePrice <- subset(HousePrice, select = c(price,bathrooms,sqft\_living,grade,sqft\_above,grade,sqft\_living15))  
i=0.6  
for(i in seq(from=0.60, to=0.95, by=0.05)){  
 rn\_train <- sample(nrow(newHousePrice),floor(nrow(newHousePrice)\*i))  
 train <- newHousePrice[rn\_train,colnames(newHousePrice)]  
 test <- newHousePrice[-rn\_train,colnames(newHousePrice)]  
 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: 170757.2  
## testing\_rmse: 172159.7  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bathrooms sqft\_living grade sqft\_above   
## -656067.19 3041.40 137.72 114557.59 -55.03   
## grade.1 sqft\_living15   
## NA 34.71

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

##   
## training\_rmse: 172884.4  
## testing\_rmse: 168126.9  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bathrooms sqft\_living grade sqft\_above   
## -672870.30 386.13 148.32 115671.51 -66.73   
## grade.1 sqft\_living15   
## NA 41.87

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

##   
## training\_rmse: 170497.4  
## testing\_rmse: 172870.2  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bathrooms sqft\_living grade sqft\_above   
## -677054.89 979.50 143.08 116617.07 -66.48   
## grade.1 sqft\_living15   
## NA 44.39

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

##   
## training\_rmse: 169695.3  
## testing\_rmse: 175740.4  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bathrooms sqft\_living grade sqft\_above   
## -666883.90 -1736.47 145.42 115683.44 -72.88   
## grade.1 sqft\_living15   
## NA 48.86

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

##   
## training\_rmse: 170642.8  
## testing\_rmse: 173498.2  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bathrooms sqft\_living grade sqft\_above   
## -676994.52 -3433.47 138.80 117598.50 -63.50   
## grade.1 sqft\_living15   
## NA 46.86

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

##   
## training\_rmse: 172319  
## testing\_rmse: 164853.6  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bathrooms sqft\_living grade sqft\_above   
## -685514.37 -3248.66 141.66 118380.12 -66.79   
## grade.1 sqft\_living15   
## NA 48.69

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

##   
## training\_rmse: 170902.4  
## testing\_rmse: 174052.9  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bathrooms sqft\_living grade sqft\_above   
## -674823.28 -2109.95 148.24 117091.23 -69.37   
## grade.1 sqft\_living15   
## NA 42.02

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

##   
## training\_rmse: 171054.5  
## testing\_rmse: 174090.8  
## Call:  
## lm(formula = price ~ ., data = train)  
##   
## Coefficients:  
## (Intercept) bathrooms sqft\_living grade sqft\_above   
## -675178.26 -1394.08 144.00 116775.17 -67.13   
## grade.1 sqft\_living15   
## NA 45.10