**Analysis on Room Rent in hotels of various cities in India**

Name: Abir Pattnaik

College: Maharaja Agrasen Institute of Technology

Email ID:abir.pattanaik@gmail.com

# Analysis on Room Rent in hotels of various cities in India  
# NAME: Abir Pattnaik  
# EMAIL: abir.pattanaik@gmail.com  
# COLLEGE: Maharaja Agrasen Institute of Technology  
  
# Reading the data onto the HotelData.df  
# Please change the destination on where the dataset is stored  
# Displaying the variables and formats that are present  
  
HotelData.df<-read.csv("C:/Users/DRDO HQ/Desktop/DATA ANALYTICS INTERNSHIP/CAPSTONE PROJECT/Cities42.csv")  
str(HotelData.df)

## 'data.frame': 13232 obs. of 20 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ CityName : Factor w/ 42 levels "Agra","Ahmedabad",..: 26 26 26 26 26 26 26 26 26 26 ...  
## $ Population : int 12442373 12442373 12442373 12442373 12442373 12442373 12442373 12442373 12442373 12442373 ...  
## $ CityRank : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ IsMetroCity : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ IsTouristDestination: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ IsWeekend : int 1 0 1 1 0 1 0 1 1 0 ...  
## $ IsNewYearEve : int 0 0 0 0 0 1 0 0 0 0 ...  
## $ Date : Factor w/ 20 levels "18-Dec-16","21-Dec-16",..: 11 12 13 14 15 16 17 18 11 12 ...  
## $ HotelName : Factor w/ 1670 levels "14 Square Amanora",..: 1635 1635 1635 1635 1635 1635 1635 1635 1409 1409 ...  
## $ RoomRent : int 12375 10250 9900 10350 12000 11475 11220 9225 6800 9350 ...  
## $ StarRating : num 5 5 5 5 5 5 5 5 4 4 ...  
## $ Airport : num 21 21 21 21 21 21 21 21 20 20 ...  
## $ HotelAddress : Factor w/ 2108 levels " H.P. High Court Mall Road, Shimla",..: 925 928 930 933 935 937 940 941 699 746 ...  
## $ HotelPincode : int 400005 400006 400007 400008 400009 400010 400011 400012 400039 400040 ...  
## $ HotelDescription : Factor w/ 1226 levels "#NAME?","10 star hotel near Queensroad, Amritsar",..: 1030 1030 1030 1030 1030 1030 1030 1030 1006 1006 ...  
## $ FreeWifi : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ FreeBreakfast : int 0 0 0 0 0 0 0 0 1 1 ...  
## $ HotelCapacity : int 287 287 287 287 287 287 287 287 28 28 ...  
## $ HasSwimmingPool : int 1 1 1 1 1 1 1 1 0 0 ...

#Converting Date to one format  
  
table(HotelData.df$Date)

##   
## 18-Dec-16 21-Dec-16 24-Dec-16 25-Dec-16 28-Dec-16 31-Dec-16   
## 44 44 44 44 44 44   
## 4-Jan-16 4-Jan-17 8-Jan-16 8-Jan-17 Dec 18 2016 Dec 21 2016   
## 31 13 31 13 1608 1611   
## Dec 24 2016 Dec 25 2016 Dec 28 2016 Dec 31 2016 Jan 04 2017 Jan 08 2017   
## 1611 1611 1611 1611 1548 1542   
## Jan 4 2017 Jan 8 2017   
## 60 67

HotelData.df$Date<-gsub("18-Dec-16","Dec 18 2016",HotelData.df$Date)  
HotelData.df$Date<-gsub("21-Dec-16","Dec 21 2016",HotelData.df$Date)  
HotelData.df$Date<-gsub("24-Dec-16","Dec 24 2016",HotelData.df$Date)  
HotelData.df$Date<-gsub("25-Dec-16","Dec 25 2016",HotelData.df$Date)  
HotelData.df$Date<-gsub("28-Dec-16","Dec 28 2016",HotelData.df$Date)  
HotelData.df$Date<-gsub("31-Dec-16","Dec 31 2016",HotelData.df$Date)  
HotelData.df$Date<-gsub("4-Jan-16","Jan 04 2017",HotelData.df$Date)  
HotelData.df$Date<-gsub("4-Jan-17","Jan 04 2017",HotelData.df$Date)  
HotelData.df$Date<-gsub("8-Jan-16","Jan 08 2017",HotelData.df$Date)  
HotelData.df$Date<-gsub("8-Jan-17","Jan 08 2017",HotelData.df$Date)  
HotelData.df$Date<-gsub("Jan 4 2017","Jan 04 2017",HotelData.df$Date)  
HotelData.df$Date<-gsub("Jan 8 2017","Jan 08 2017",HotelData.df$Date)  
  
table(HotelData.df$Date)

##   
## Dec 18 2016 Dec 21 2016 Dec 24 2016 Dec 25 2016 Dec 28 2016 Dec 31 2016   
## 1652 1655 1655 1655 1655 1655   
## Jan 04 2017 Jan 08 2017   
## 1652 1653

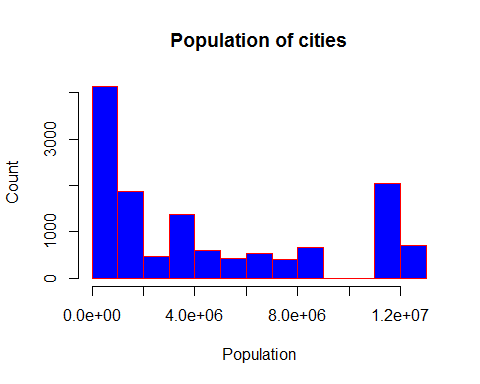
#Converting Data into different formats  
  
str(HotelData.df)

## 'data.frame': 13232 obs. of 20 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ CityName : Factor w/ 42 levels "Agra","Ahmedabad",..: 26 26 26 26 26 26 26 26 26 26 ...  
## $ Population : int 12442373 12442373 12442373 12442373 12442373 12442373 12442373 12442373 12442373 12442373 ...  
## $ CityRank : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ IsMetroCity : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ IsTouristDestination: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ IsWeekend : int 1 0 1 1 0 1 0 1 1 0 ...  
## $ IsNewYearEve : int 0 0 0 0 0 1 0 0 0 0 ...  
## $ Date : chr "Dec 18 2016" "Dec 21 2016" "Dec 24 2016" "Dec 25 2016" ...  
## $ HotelName : Factor w/ 1670 levels "14 Square Amanora",..: 1635 1635 1635 1635 1635 1635 1635 1635 1409 1409 ...  
## $ RoomRent : int 12375 10250 9900 10350 12000 11475 11220 9225 6800 9350 ...  
## $ StarRating : num 5 5 5 5 5 5 5 5 4 4 ...  
## $ Airport : num 21 21 21 21 21 21 21 21 20 20 ...  
## $ HotelAddress : Factor w/ 2108 levels " H.P. High Court Mall Road, Shimla",..: 925 928 930 933 935 937 940 941 699 746 ...  
## $ HotelPincode : int 400005 400006 400007 400008 400009 400010 400011 400012 400039 400040 ...  
## $ HotelDescription : Factor w/ 1226 levels "#NAME?","10 star hotel near Queensroad, Amritsar",..: 1030 1030 1030 1030 1030 1030 1030 1030 1006 1006 ...  
## $ FreeWifi : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ FreeBreakfast : int 0 0 0 0 0 0 0 0 1 1 ...  
## $ HotelCapacity : int 287 287 287 287 287 287 287 287 28 28 ...  
## $ HasSwimmingPool : int 1 1 1 1 1 1 1 1 0 0 ...

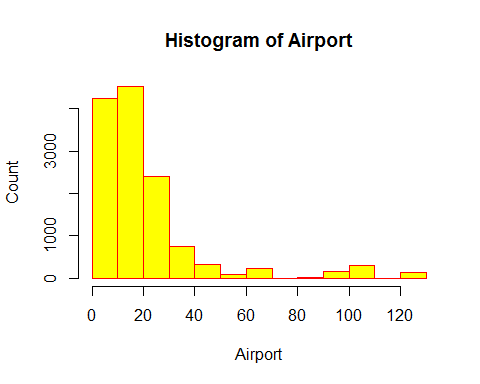
HotelData.df$Date<-as.Date(c("Dec 18 2016","Dec 21 2016","Dec 24 2016","Dec 25 2016","Dec 28 2016","Dec 31 2016","Jan 04 2017","Jan 08 2017"),"%b %d %Y")  
HotelData.df$HotelAddress<-as.character(HotelData.df$HotelAddress)  
HotelData.df$HotelDescription<-as.character(HotelData.df$HotelDescription)  
HotelData.df$HotelName<-as.character(HotelData.df$HotelName)  
  
# Summary of the data   
# (a)Population inference:Min.-8096 Mean-4416837 Median-3046163 Max.-12442373  
# (b)RoomRent:Min.-299 Median-4000 Mean-5474 Max.-322500  
# RoomRent outliers- 3rd quantile+IQR\*1.5= 6299+5794.5=12093.5  
# RoomRent outliers- 1st quantile-IQR\*1.5= 2436-5794.5=-3358.5  
# (c)Star Rating=3.46  
# (d)Airport Distance Min.-0.20 Median-15 Mean-21.16 Max.-124  
# Airport Distance outliers- 3rd quantile+IQR\*1.5= 24+23.4=47.4  
# Airport Distance outliers- 1st quantile-IQR\*1.5= 8.4-23.4=-15  
# (e)Hotel Capacity Min.0 Median-34 Mean-62.5 Max.-600  
  
summary(HotelData.df)

## X CityName Population CityRank   
## Min. : 1 Delhi :2048 Min. : 8096 Min. : 0.00   
## 1st Qu.: 3309 Jaipur : 768 1st Qu.: 744983 1st Qu.: 2.00   
## Median : 6616 Mumbai : 712 Median : 3046163 Median : 9.00   
## Mean : 6616 Bangalore: 656 Mean : 4416837 Mean :14.83   
## 3rd Qu.: 9924 Goa : 624 3rd Qu.: 8443675 3rd Qu.:24.00   
## Max. :13232 Kochi : 608 Max. :12442373 Max. :44.00   
## (Other) :7816   
## IsMetroCity IsTouristDestination IsWeekend IsNewYearEve   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :1.0000 Median :1.0000 Median :0.0000   
## Mean :0.2842 Mean :0.6972 Mean :0.6228 Mean :0.1244   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
##   
## Date HotelName RoomRent StarRating   
## Min. :2016-12-18 Length:13232 Min. : 299 Min. :0.000   
## 1st Qu.:2016-12-23 Class :character 1st Qu.: 2436 1st Qu.:3.000   
## Median :2016-12-26 Mode :character Median : 4000 Median :3.000   
## Mean :2016-12-27 Mean : 5474 Mean :3.459   
## 3rd Qu.:2017-01-01 3rd Qu.: 6299 3rd Qu.:4.000   
## Max. :2017-01-08 Max. :322500 Max. :5.000   
##   
## Airport HotelAddress HotelPincode HotelDescription   
## Min. : 0.20 Length:13232 Min. : 100025 Length:13232   
## 1st Qu.: 8.40 Class :character 1st Qu.: 221001 Class :character   
## Median : 15.00 Mode :character Median : 395003 Mode :character   
## Mean : 21.16 Mean : 397430   
## 3rd Qu.: 24.00 3rd Qu.: 570001   
## Max. :124.00 Max. :7000157   
##   
## FreeWifi FreeBreakfast HotelCapacity HasSwimmingPool   
## Min. :0.0000 Min. :0.0000 Min. : 0.00 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.: 16.00 1st Qu.:0.0000   
## Median :1.0000 Median :1.0000 Median : 34.00 Median :0.0000   
## Mean :0.9259 Mean :0.6491 Mean : 62.51 Mean :0.3558   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.: 75.00 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :600.00 Max. :1.0000   
##

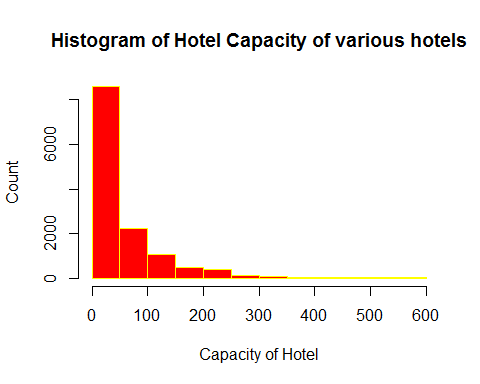
# Individual Variables  
# (a)Histogram of Population   
# Most cities have population in the range of 0-4000 and only few cities have population over 12000---These could be metropolitan cities   
# (b)Histogram of Airport Distance  
# Positively skewed graph with most of the hotels in the range of 0-40 km  
# (c)Histogram of HotelCapacity  
# Positively skewed graph with capacity of hotels in the range of 0-300  
# (d)Histogram of StarRating of hotels  
# Most of hotel ratings are in the range of 2.5-3.5   
  
attach(HotelData.df)  
hist(Population,col="blue",border="red",main="Population of cities",ylab="Count")



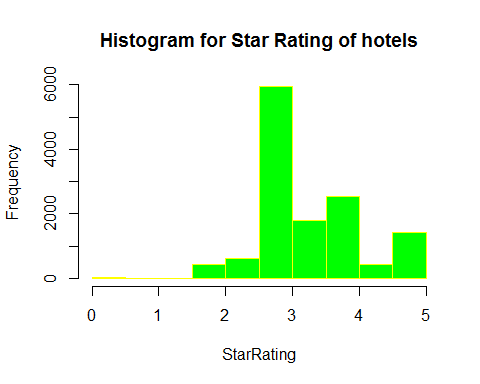
hist(Airport,col="yellow",border="red",ylab="Count")



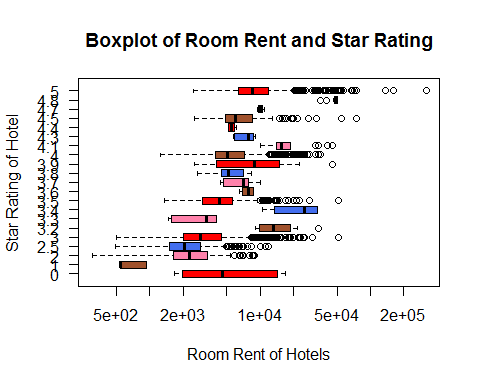
hist(HotelCapacity,col="red",border="yellow",main="Histogram of Hotel Capacity of various hotels",xlab="Capacity of Hotel",ylab="Count")



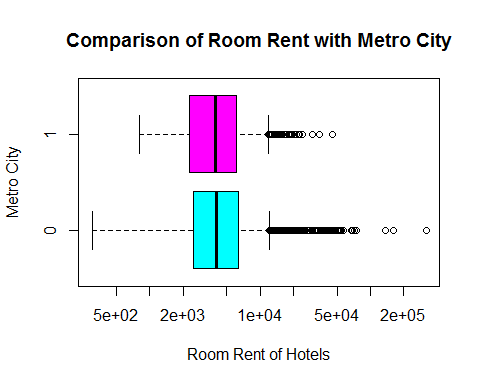
hist(StarRating,col="green",border="yellow",main="Histogram for Star Rating of hotels")



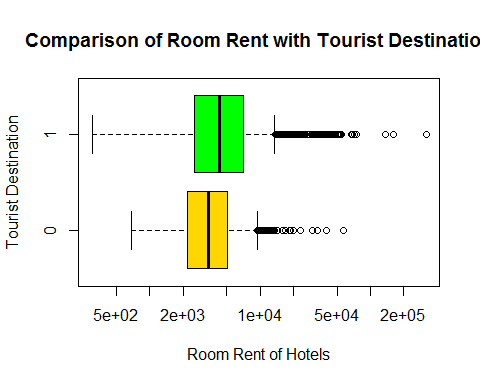
# Comparison of Function Variable to other variables  
# (a)Boxplot of Room Rent and Star Rating of Hotel  
# Room Rent Increases with Increase in Star Rating  
# (b)Box Plot of Room Rent and whether city is a Metro City  
# Non Metro cities have a higher room rent than Metro Cities  
# (c)Boxplot of Room Rent and whether the city is a tourist destination   
# Those with tourist destinations have higher room rents than those which are not tourist destination  
# (d)Boxplot of Room Rent and whether it was New Year's Eve  
# No considerable difference in room rent and New Year's Eve  
# (e)Boxplot of Room Rent and whether hotel has a free Wi-fi  
# Room Rent are higher when Free Wi-fi is available  
# (f)Boxplot of Room Rent and Whether hotel has swimming pool  
# Room Rents are considerably high when Swimming pools are available  
# (g)Boxplot of Room Rent and Whether hotel provides breakfast  
# There is no considerable difference in hotel rent and breakfast barring one or two outliers  
  
boxplot(RoomRent~StarRating,log="x",las=1,col=c("red","sienna","palevioletred1","royalblue2"),horizontal=TRUE,xlab="Room Rent of Hotels",ylab="Star Rating of Hotel",main="Boxplot of Room Rent and Star Rating")



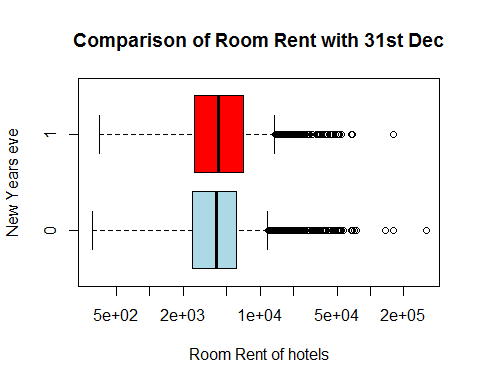
boxplot(RoomRent~IsMetroCity,log="x",col=c("cyan","magenta"),horizontal=TRUE,xlab="Room Rent of Hotels",ylab="Metro City",main="Comparison of Room Rent with Metro City")



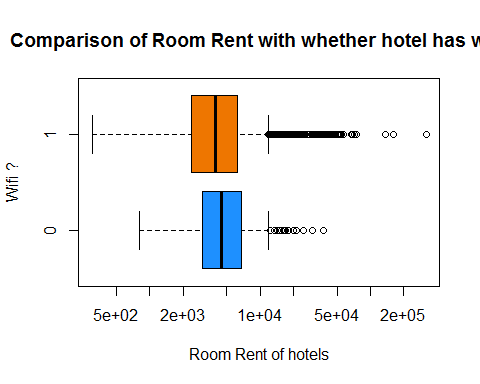
boxplot(RoomRent~IsTouristDestination,log="x",horizontal=TRUE,col=c("gold","green"),xlab="Room Rent of Hotels",ylab="Tourist Destination",main="Comparison of Room Rent with Tourist Destination")



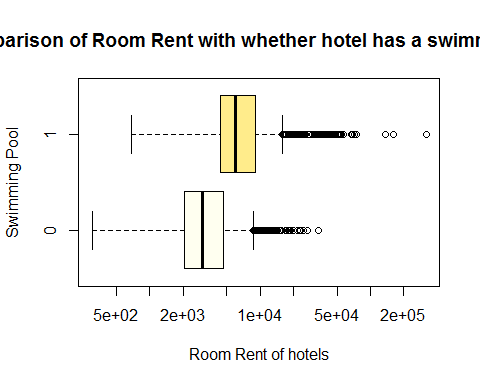
boxplot(RoomRent~IsNewYearEve,log="x",horizontal=TRUE,col=c("lightblue","red"),main="Comparison of Room Rent with 31st Dec",xlab="Room Rent of hotels",ylab="New Years eve")



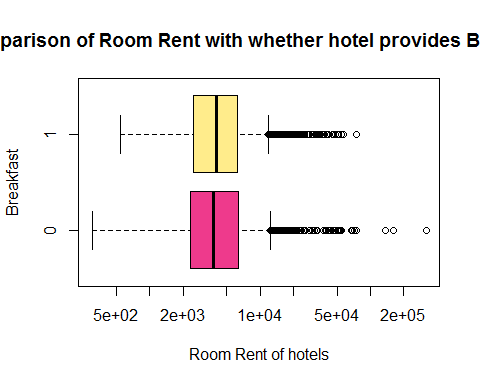
boxplot(RoomRent~FreeWifi,log="x",horizontal=TRUE,col=c("dodgerblue1","darkorange2"),main="Comparison of Room Rent with whether hotel has wi-fi",xlab="Room Rent of hotels",ylab="Wifi ?")



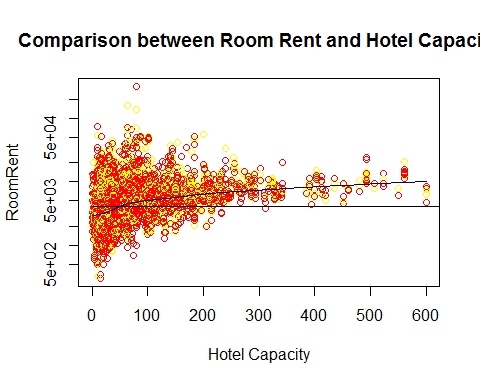
boxplot(RoomRent~HasSwimmingPool,log="x",horizontal=TRUE,col=c("ivory","lightgoldenrod1"),main="Comparison of Room Rent with whether hotel has a swimming pool",xlab="Room Rent of hotels",ylab="Swimming Pool")



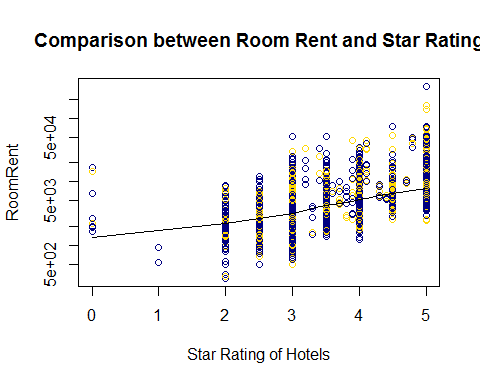
boxplot(RoomRent~FreeBreakfast,log="x",horizontal=TRUE,col=c("violetred2","lightgoldenrod1"),main="Comparison of Room Rent with whether hotel provides BreakFast",xlab="Room Rent of hotels",ylab="Breakfast")



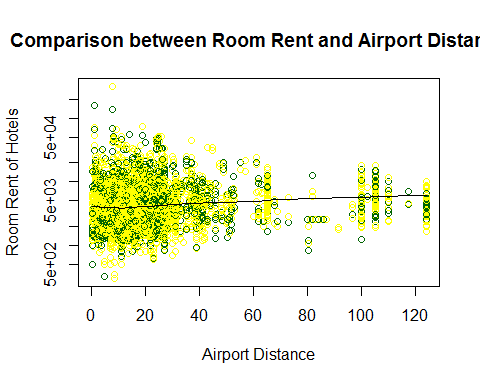
# Plots for continuos variables  
# (a)Plot for Room Rent and Hotel Capacity  
# Most of the hotel room rents are higher when hotel capacity is less than 300.Slow Increase in room rent   
# (b)Plot for Room Rent and Star Rating  
# Considerable Increase in the rates of Hotel rent as the ratings are increased  
# (c)Plot for Room Rent and Airport Distance  
# Slight Increase in the difference of Room rents and distance of hotel from airport  
  
plot(~HotelCapacity+RoomRent,log="y",type="p",col=c("yellow","red"),main="Comparison between Room Rent and Hotel Capacity",xlab="Hotel Capacity",ylab="Room Rent of Hotels")  
lines(lowess(RoomRent))  
lines(lowess(RoomRent~HotelCapacity),col="navy")



plot(~StarRating+RoomRent,log="y",col=c("gold","navy"),type="p",main="Comparison between Room Rent and Star Rating",xlab="Star Rating of Hotels",ylab="Room Rent of Hotels")  
lines(lowess(RoomRent~StarRating))

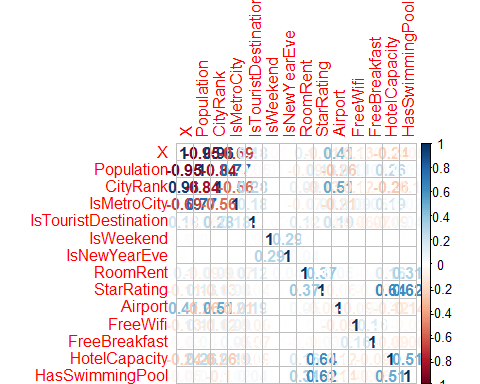


plot(RoomRent~Airport,log="y",col=c("darkgreen","yellow"),type="p",main="Comparison between Room Rent and Airport Distance",xlab="Airport Distance",ylab="Room Rent of Hotels")  
lines(lowess(RoomRent~Airport))

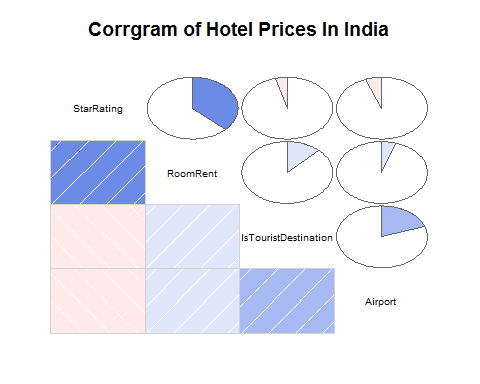


# Drawing the correlation matrix that specifies which objects are highly correlated.  
# A value that reaches near correlation value 1 has positive correlation while that has value going  
# towards -1 has negative value.  
# (a)Star Rating has a correlation value of 0.60 with Hotel Rating   
# (b)Star Rating has a correlation value of 0.62 with HasSwimmingPool  
# (c)Hotel Capacity has a correlation value of 0.51 with HasSwimmingPool  
# (d)Rest have low correlation values  
  
library(corrplot)  
corrplot(cor(HotelData.df[,-c(2,9,10,14,15,16)]),method="number")  
  
# Variables taken  
# StarRating,IsTouristDestination,Airport  
# Experimenting the variables with the RoomRent  
# Inferences: (a)All the correlations were not too high as they reduced homoscedasticity  
# (b)The pairs function showed interesting plots with respect to the RoomRent as only IsTouristDestination  
# contained the categorical variable rest both of them were continuos  
library(corrgram)

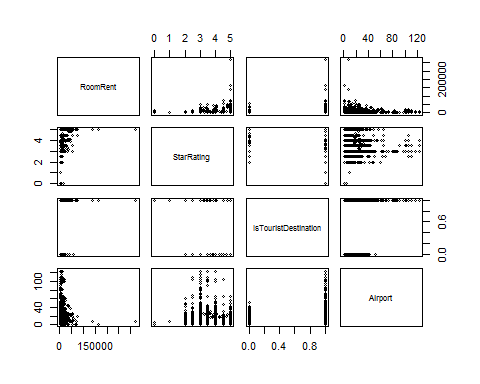
## Warning: package 'corrgram' was built under R version 3.4.1



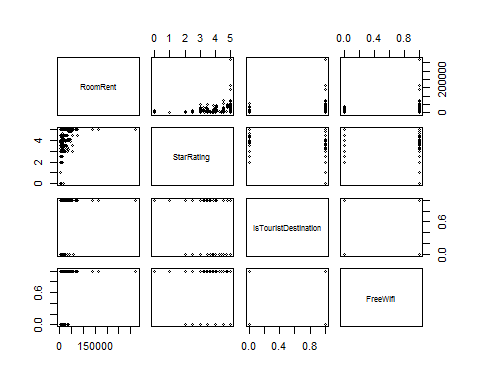
corrgram.data=data.frame(RoomRent,StarRating,IsTouristDestination,Airport)  
corrgram(corrgram.data, order=TRUE, lower.panel=panel.shade,upper.panel=panel.pie, text.panel=panel.txt,main="Corrgram of Hotel Prices In India")



library(car)  
pairs(RoomRent~StarRating+IsTouristDestination+Airport,cex=0.6,data=HotelData.df)



library(car)  
pairs(RoomRent~StarRating+IsTouristDestination+FreeWifi,cex=0.6,data=HotelData.df)



# t.tests  
# T test for when the average of Room Rent is higher when It is a tourist Destination  
# T test for when the average of Room Rent is higher when Hotel has a swimming pool  
  
t.test(RoomRent~IsTouristDestination,alternative="greater")

##   
## Welch Two Sample t-test  
##   
## data: RoomRent by IsTouristDestination  
## t = -19.449, df = 12888, p-value = 1  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## -2120.376 Inf  
## sample estimates:  
## mean in group 0 mean in group 1   
## 4111.003 6066.024

t.test(RoomRent~HasSwimmingPool,alternative="greater")

##   
## Welch Two Sample t-test  
##   
## data: RoomRent by HasSwimmingPool  
## t = -29.013, df = 5011.3, p-value = 1  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## -5044.158 Inf  
## sample estimates:  
## mean in group 0 mean in group 1   
## 3775.566 8549.052

# Aggregate command for Room Rent and variables-Airport,Star Rating and IsTouristDestination  
# (a)The Room Rent doesn't quite depend on the StarRating linearly but a 4.8 rating starred hotel has average of 46000 Room Rent  
# (b)Rates are quite high when it is closer ranging from 3000-24000  
# (c)Room Rents are higher when it is a tourist destination  
# (d)On New Years Eve the rates of Room Rent is higher as compared to a regular day  
  
aggregate(RoomRent,by=list(StarRating),mean)

## Group.1 x  
## 1 0.0 7237.125  
## 2 1.0 686.625  
## 3 2.0 2783.166  
## 4 2.5 2520.816  
## 5 3.0 3694.811  
## 6 3.2 15937.500  
## 7 3.3 2841.062  
## 8 3.4 23437.500  
## 9 3.5 4843.346  
## 10 3.6 7769.500  
## 11 3.7 6701.958  
## 12 3.8 5400.062  
## 13 3.9 13062.750  
## 14 4.0 6393.105  
## 15 4.1 19075.000  
## 16 4.3 7423.125  
## 17 4.4 5563.500  
## 18 4.5 8699.920  
## 19 4.7 10125.000  
## 20 4.8 46752.812  
## 21 5.0 12398.221

aggregate(RoomRent,by=list(Airport),mean)

## Group.1 x  
## 1 0.2 5247.000  
## 2 0.3 2217.500  
## 3 0.4 4987.025  
## 4 0.5 2437.458  
## 5 0.6 4805.781  
## 6 0.7 2894.750  
## 7 0.8 4687.500  
## 8 0.9 4356.128  
## 9 1.0 4452.812  
## 10 1.1 43487.500  
## 11 1.2 5154.050  
## 12 1.4 15562.500  
## 13 1.5 7273.250  
## 14 1.6 10460.469  
## 15 1.7 4968.091  
## 16 1.8 3298.681  
## 17 1.9 6440.400  
## 18 2.0 5804.821  
## 19 2.1 3400.975  
## 20 2.2 4448.000  
## 21 2.3 8030.938  
## 22 2.4 5419.656  
## 23 2.5 5863.000  
## 24 2.6 3027.396  
## 25 2.7 4659.054  
## 26 2.8 4563.417  
## 27 2.9 3667.232  
## 28 3.0 4636.929  
## 29 3.1 5859.875  
## 30 3.2 7848.667  
## 31 3.3 8960.125  
## 32 3.4 6088.854  
## 33 3.5 4267.839  
## 34 3.6 6699.828  
## 35 3.7 2603.125  
## 36 3.8 3145.825  
## 37 3.9 4075.250  
## 38 4.0 4013.944  
## 39 4.1 4218.688  
## 40 4.2 3857.075  
## 41 4.3 4388.656  
## 42 4.4 5525.062  
## 43 4.5 6135.208  
## 44 4.6 5061.025  
## 45 4.7 2378.125  
## 46 4.8 3889.250  
## 47 4.9 3421.219  
## 48 5.0 6144.041  
## 49 5.1 6547.000  
## 50 5.2 4390.306  
## 51 5.3 4119.250  
## 52 5.4 4705.900  
## 53 5.5 4371.625  
## 54 5.6 5430.900  
## 55 5.7 6046.594  
## 56 5.8 5625.607  
## 57 5.9 5016.650  
## 58 6.0 3875.970  
## 59 6.1 5853.375  
## 60 6.2 2701.828  
## 61 6.3 1705.125  
## 62 6.4 3212.479  
## 63 6.5 4871.208  
## 64 6.6 3623.625  
## 65 6.7 2994.292  
## 66 6.8 4132.357  
## 67 6.9 2749.950  
## 68 7.0 4028.469  
## 69 7.1 2366.806  
## 70 7.2 3611.604  
## 71 7.3 2522.042  
## 72 7.4 3571.975  
## 73 7.5 4816.667  
## 74 7.6 4420.141  
## 75 7.7 27828.708  
## 76 7.8 5656.594  
## 77 7.9 2730.958  
## 78 8.0 9879.685  
## 79 8.1 5120.639  
## 80 8.2 3684.607  
## 81 8.3 5117.825  
## 82 8.4 4164.208  
## 83 8.5 2991.688  
## 84 8.6 1378.938  
## 85 8.7 4457.446  
## 86 8.8 3498.562  
## 87 8.9 2627.250  
## 88 9.0 3618.694  
## 89 9.1 5122.458  
## 90 9.2 9520.790  
## 91 9.3 4147.375  
## 92 9.4 5233.500  
## 93 9.5 8224.909  
## 94 9.6 3226.050  
## 95 9.7 4376.083  
## 96 9.8 4045.625  
## 97 9.9 7947.732  
## 98 10.0 5184.302  
## 99 10.2 2780.000  
## 100 10.3 2587.000  
## 101 10.4 2346.000  
## 102 10.6 1574.375  
## 103 10.7 7025.000  
## 104 10.8 12157.875  
## 105 10.9 1949.812  
## 106 11.0 4665.262  
## 107 11.1 2867.250  
## 108 11.3 1948.812  
## 109 11.7 4069.000  
## 110 11.9 7264.938  
## 111 12.0 5014.164  
## 112 12.2 3113.458  
## 113 12.3 1746.750  
## 114 12.6 4241.000  
## 115 12.7 4566.750  
## 116 13.0 6872.332  
## 117 13.1 2525.000  
## 118 13.3 4881.250  
## 119 13.5 1831.250  
## 120 13.6 4371.333  
## 121 13.7 5908.812  
## 122 13.8 2507.500  
## 123 14.0 3632.243  
## 124 14.2 1801.000  
## 125 14.4 4002.167  
## 126 14.5 3847.500  
## 127 14.6 6711.438  
## 128 14.7 5431.167  
## 129 14.8 7086.625  
## 130 14.9 4631.250  
## 131 15.0 4804.245  
## 132 15.3 2983.875  
## 133 15.4 5179.125  
## 134 15.6 4233.375  
## 135 15.7 3385.250  
## 136 15.8 5960.500  
## 137 15.9 9961.875  
## 138 16.0 5052.724  
## 139 16.1 10451.000  
## 140 16.2 4637.250  
## 141 16.4 2404.250  
## 142 16.5 4639.250  
## 143 16.7 6648.281  
## 144 17.0 5245.613  
## 145 17.1 3251.000  
## 146 17.2 4874.500  
## 147 17.4 1911.750  
## 148 17.5 16538.125  
## 149 17.6 6273.000  
## 150 17.8 4139.438  
## 151 18.0 5023.542  
## 152 18.3 6125.000  
## 153 18.5 3543.250  
## 154 18.6 6693.750  
## 155 18.7 2782.625  
## 156 19.0 10216.920  
## 157 19.5 2262.500  
## 158 19.9 7232.500  
## 159 20.0 5474.096  
## 160 20.2 8412.500  
## 161 20.3 3930.812  
## 162 20.5 2169.625  
## 163 20.9 6281.750  
## 164 21.0 4546.419  
## 165 21.4 6944.500  
## 166 21.5 3882.750  
## 167 22.0 4453.590  
## 168 22.1 5305.000  
## 169 22.2 3235.000  
## 170 22.4 3887.500  
## 171 22.5 6103.250  
## 172 23.0 5019.740  
## 173 23.2 10887.500  
## 174 23.3 5088.000  
## 175 23.4 4942.375  
## 176 24.0 3863.335  
## 177 24.2 38115.625  
## 178 24.3 16894.500  
## 179 24.5 5305.750  
## 180 24.6 45274.375  
## 181 24.7 2078.000  
## 182 24.9 20867.438  
## 183 25.0 5229.457  
## 184 25.6 7140.625  
## 185 25.7 6137.500  
## 186 25.9 15937.500  
## 187 26.0 6258.703  
## 188 26.1 26156.250  
## 189 26.3 2369.250  
## 190 26.4 7483.000  
## 191 26.5 6112.500  
## 192 26.7 7992.500  
## 193 27.0 5835.206  
## 194 27.1 23437.500  
## 195 27.2 4832.000  
## 196 28.0 3282.277  
## 197 28.1 7140.625  
## 198 28.6 7518.750  
## 199 28.7 3781.625  
## 200 29.0 3602.364  
## 201 30.0 5784.393  
## 202 30.5 20500.000  
## 203 31.0 4943.406  
## 204 31.2 6193.750  
## 205 31.3 9125.000  
## 206 31.9 4204.750  
## 207 32.0 5803.528  
## 208 32.9 7936.875  
## 209 33.0 3026.100  
## 210 33.4 6292.000  
## 211 34.0 5784.875  
## 212 35.0 8111.898  
## 213 36.0 7528.882  
## 214 36.2 6871.500  
## 215 37.0 8712.878  
## 216 38.0 6006.755  
## 217 38.3 8117.875  
## 218 39.0 4524.650  
## 219 39.9 2206.500  
## 220 40.0 5576.768  
## 221 41.0 5355.676  
## 222 42.0 3292.293  
## 223 42.7 4118.750  
## 224 43.0 7559.758  
## 225 43.9 9247.500  
## 226 44.0 5925.000  
## 227 44.5 4233.125  
## 228 44.6 7147.000  
## 229 44.8 33033.500  
## 230 46.0 4236.850  
## 231 47.0 7256.000  
## 232 47.5 19108.125  
## 233 48.0 4268.750  
## 234 48.4 3000.000  
## 235 49.0 18237.500  
## 236 50.0 5681.875  
## 237 50.1 2360.875  
## 238 50.5 3417.750  
## 239 51.0 3178.250  
## 240 52.0 4198.375  
## 241 52.7 7820.000  
## 242 53.0 4062.500  
## 243 55.0 18950.000  
## 244 57.2 15375.000  
## 245 60.0 2846.000  
## 246 61.0 14319.062  
## 247 62.0 5412.719  
## 248 63.0 8687.500  
## 249 63.5 3900.000  
## 250 63.6 2625.000  
## 251 65.0 6257.888  
## 252 67.6 4149.750  
## 253 69.0 2682.125  
## 254 73.1 3172.500  
## 255 80.0 2554.000  
## 256 80.3 1117.750  
## 257 81.0 2554.000  
## 258 82.0 6717.111  
## 259 83.0 2554.000  
## 260 84.0 2554.000  
## 261 85.0 2554.000  
## 262 86.0 2554.000  
## 263 87.0 2554.000  
## 264 91.3 1758.875  
## 265 96.5 3821.375  
## 266 100.0 6144.257  
## 267 102.4 6444.750  
## 268 105.0 8162.371  
## 269 110.0 5976.109  
## 270 117.4 6337.375  
## 271 124.0 4629.648

aggregate(RoomRent,by=list(IsTouristDestination),mean)

## Group.1 x  
## 1 0 4111.003  
## 2 1 6066.024

aggregate(RoomRent,by=list(Date),mean)

## Group.1 x  
## 1 2016-12-18 4753.288  
## 2 2016-12-21 4916.978  
## 3 2016-12-24 5469.770  
## 4 2016-12-25 5472.848  
## 5 2016-12-28 5720.298  
## 6 2016-12-31 6256.907  
## 7 2017-01-04 5764.073  
## 8 2017-01-08 5437.774

# Creating correlation and covariance  
# These factors although not correlated but show high variances between them  
  
x<-HotelData.df[,c("IsTouristDestination","StarRating", "Airport")]  
y<-HotelData.df[,c("RoomRent")]  
cor(x,y)

## [,1]  
## IsTouristDestination 0.12250296  
## StarRating 0.36937343  
## Airport 0.04965324

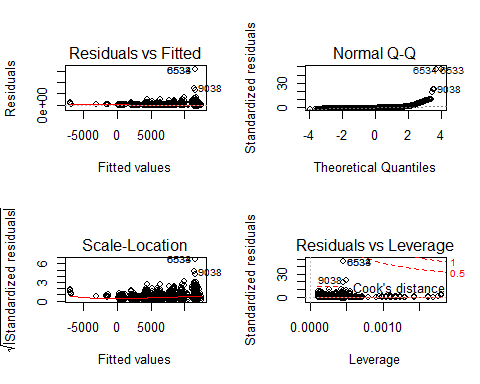
cov(x,y)

## [,1]  
## IsTouristDestination 412.7803  
## StarRating 2048.3755  
## Airport 8287.1786

# Creating a linear model\_1  
# INTERPRETATIONS-The model that was created had high significance factors, this meant that   
# each model created has its own significance and cannot be removed as they help in getting a R^2   
# value although it was low  
# R^2 and adjusted R^2 value are respectively 0.1575 and 0.1574  
# However due to huge RSE value this model is not a correct adaptation of the RoomRent   
# but these factors shouldn't be eliminated also.  
# The outlier test for the model provided the outliers that may affect the model and  
# its overall value but it wasn't much so.  
# VIF stands for Variation Inflation Factor.This shows the variation.  
# Any Variation that is greater than 10 creates a problem but as displayed most   
# of them were 1  
# gvlma model stands for Global Validation of Linear Models Assumptions  
# This model has been created for checking the validation of the model  
# with respect to GlobalStat,Skewness,Kurtosis,Link Function,Heteroscedasticity  
# Plotting a linear model with respect to the values 4 graphs were created.  
# In Residuals Vs Fitted shows a linear relationship that means this is not a correct model  
# The Normal Q-Q model shows that a linear model upto a value but huge outliers create a problem  
# The fourth plot has the cook's distance where the outliers that surpass the 0.5 range are rejected.  
# However it is shown that these outliers didn't matter.  
  
   
lm.fit1=lm(RoomRent~StarRating+Airport+IsTouristDestination,data=HotelData.df)  
summary(lm.fit1)

##   
## Call:  
## lm(formula = RoomRent ~ StarRating + Airport + IsTouristDestination,   
## data = HotelData.df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9076 -2427 -863 904 310970   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8942.499 296.290 -30.182 < 2e-16 \*\*\*  
## StarRating 3660.485 77.563 47.194 < 2e-16 \*\*\*  
## Airport 15.355 2.625 5.849 5.05e-09 \*\*\*  
## IsTouristDestination 2051.468 129.893 15.794 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6731 on 13228 degrees of freedom  
## Multiple R-squared: 0.1575, Adjusted R-squared: 0.1574   
## F-statistic: 824.6 on 3 and 13228 DF, p-value: < 2.2e-16

par(mfrow=c(2,2))  
plot(lm.fit1)



par(mfrow=c(1,1))  
  
library(car)  
outlierTest(lm.fit1)

## rstudent unadjusted p-value Bonferonni p  
## 6533 50.456454 0.0000e+00 0.0000e+00  
## 6534 50.456454 0.0000e+00 0.0000e+00  
## 9038 22.796140 7.3842e-113 9.7708e-109  
## 9039 22.796140 7.3842e-113 9.7708e-109  
## 6535 18.970145 3.3778e-79 4.4695e-75  
## 8769 9.710326 3.2321e-22 4.2767e-18  
## 8770 9.710326 3.2321e-22 4.2767e-18  
## 8772 9.710326 3.2321e-22 4.2767e-18  
## 8776 9.710326 3.2321e-22 4.2767e-18  
## 8657 9.436398 4.4939e-21 5.9463e-17

vif(lm.fit1)

## StarRating Airport IsTouristDestination   
## 1.004589 1.042337 1.040179

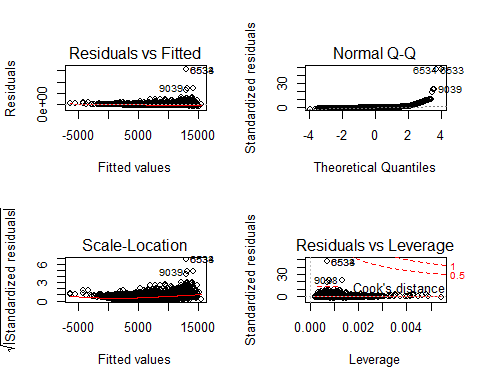
library(gvlma)  
gvmodel <- gvlma(lm.fit1)   
summary(gvmodel)

##   
## Call:  
## lm(formula = RoomRent ~ StarRating + Airport + IsTouristDestination,   
## data = HotelData.df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9076 -2427 -863 904 310970   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8942.499 296.290 -30.182 < 2e-16 \*\*\*  
## StarRating 3660.485 77.563 47.194 < 2e-16 \*\*\*  
## Airport 15.355 2.625 5.849 5.05e-09 \*\*\*  
## IsTouristDestination 2051.468 129.893 15.794 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6731 on 13228 degrees of freedom  
## Multiple R-squared: 0.1575, Adjusted R-squared: 0.1574   
## F-statistic: 824.6 on 3 and 13228 DF, p-value: < 2.2e-16  
##   
##   
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS  
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:  
## Level of Significance = 0.05   
##   
## Call:  
## gvlma(x = lm.fit1)   
##   
## Value p-value Decision  
## Global Stat 3.105e+08 0 Assumptions NOT satisfied!  
## Skewness 8.367e+05 0 Assumptions NOT satisfied!  
## Kurtosis 3.097e+08 0 Assumptions NOT satisfied!  
## Link Function 4.149e+02 0 Assumptions NOT satisfied!  
## Heteroscedasticity 5.662e+02 0 Assumptions NOT satisfied!

# Creating a linear model\_2  
# This model was created containing variables that are not descriptive but are both continuos and categorical variable  
# INTERPRETATIONS-The model that was created had high significance factors, this meant that   
# each model created has its own significance and cannot be removed as they help in getting a R^2   
# value although it was low  
# R^2 and adjusted R^2 value are respectively 0.1904 and 0.1899  
# However due to huge RSE value this model is not a correct adaptation of the RoomRent   
# but these factors shouldn't be eliminated also.  
# The outlier test for the model provided the outliers that may affect the model and  
# its overall value but it wasn't much so.  
# VIF stands for Variation Inflation Factor.This shows the variation.  
# Any Variation that is greater than 10 creates a problem but as displayed most   
# of them were 1  
# gvlma model stands for Global Validation of Linear Models Assumptions  
# This model has been created for checking the validation of the model  
# with respect to GlobalStat,Skewness,Kurtosis,Link Function,Heteroscedasticity  
# Plotting a linear model with respect to the values 4 graphs were created.  
# In Residuals Vs Fitted shows a linear relationship that means this is not a correct model  
# The Normal Q-Q model shows that a linear model upto a value but huge outliers create a problem  
# The fourth plot has the cook's distance where the outliers that surpass the 0.5 range are rejected.  
# However it is shown that these outliers didn't matter.  
  
lm.fit2=lm(RoomRent~Population+IsMetroCity+IsTouristDestination+IsNewYearEve+StarRating+Airport+FreeWifi+HotelCapacity+HasSwimmingPool,data=HotelData.df)  
summary(lm.fit2)

##   
## Call:  
## lm(formula = RoomRent ~ Population + IsMetroCity + IsTouristDestination +   
## IsNewYearEve + StarRating + Airport + FreeWifi + HotelCapacity +   
## HasSwimmingPool, data = HotelData.df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11839 -2385 -691 1045 309532   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8.560e+03 4.055e+02 -21.109 < 2e-16 \*\*\*  
## Population -1.244e-04 2.263e-05 -5.499 3.88e-08 \*\*\*  
## IsMetroCity -6.369e+02 2.132e+02 -2.988 0.00282 \*\*   
## IsTouristDestination 1.918e+03 1.374e+02 13.958 < 2e-16 \*\*\*  
## IsNewYearEve 8.430e+02 1.739e+02 4.849 1.26e-06 \*\*\*  
## StarRating 3.598e+03 1.104e+02 32.582 < 2e-16 \*\*\*  
## Airport 1.001e+01 2.716e+00 3.684 0.00023 \*\*\*  
## FreeWifi 5.952e+02 2.217e+02 2.685 0.00726 \*\*   
## HotelCapacity -1.040e+01 1.029e+00 -10.115 < 2e-16 \*\*\*  
## HasSwimmingPool 2.147e+03 1.598e+02 13.434 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6600 on 13222 degrees of freedom  
## Multiple R-squared: 0.1904, Adjusted R-squared: 0.1899   
## F-statistic: 345.5 on 9 and 13222 DF, p-value: < 2.2e-16

par(mfrow=c(2,2))  
plot(lm.fit2)



par(mfrow=c(1,1))  
  
library(car)  
outlierTest(lm.fit2)

## rstudent unadjusted p-value Bonferonni p  
## 6533 51.381281 0.0000e+00 0.0000e+00  
## 6534 51.381281 0.0000e+00 0.0000e+00  
## 9039 22.980791 1.2600e-114 1.6673e-110  
## 9038 22.851528 2.1880e-113 2.8951e-109  
## 6535 19.132790 1.6434e-80 2.1746e-76  
## 8769 9.545747 1.5853e-21 2.0977e-17  
## 8770 9.545747 1.5853e-21 2.0977e-17  
## 8772 9.545747 1.5853e-21 2.0977e-17  
## 8776 9.545747 1.5853e-21 2.0977e-17  
## 8657 9.271790 2.1105e-20 2.7926e-16

vif(lm.fit2)

## Population IsMetroCity IsTouristDestination   
## 2.820418 2.807261 1.210458   
## IsNewYearEve StarRating Airport   
## 1.000013 2.118451 1.160325   
## FreeWifi HotelCapacity HasSwimmingPool   
## 1.024652 1.888342 1.777718

library(gvlma)  
gvmodel <- gvlma(lm.fit2)   
summary(gvmodel)

##   
## Call:  
## lm(formula = RoomRent ~ Population + IsMetroCity + IsTouristDestination +   
## IsNewYearEve + StarRating + Airport + FreeWifi + HotelCapacity +   
## HasSwimmingPool, data = HotelData.df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11839 -2385 -691 1045 309532   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -8.560e+03 4.055e+02 -21.109 < 2e-16 \*\*\*  
## Population -1.244e-04 2.263e-05 -5.499 3.88e-08 \*\*\*  
## IsMetroCity -6.369e+02 2.132e+02 -2.988 0.00282 \*\*   
## IsTouristDestination 1.918e+03 1.374e+02 13.958 < 2e-16 \*\*\*  
## IsNewYearEve 8.430e+02 1.739e+02 4.849 1.26e-06 \*\*\*  
## StarRating 3.598e+03 1.104e+02 32.582 < 2e-16 \*\*\*  
## Airport 1.001e+01 2.716e+00 3.684 0.00023 \*\*\*  
## FreeWifi 5.952e+02 2.217e+02 2.685 0.00726 \*\*   
## HotelCapacity -1.040e+01 1.029e+00 -10.115 < 2e-16 \*\*\*  
## HasSwimmingPool 2.147e+03 1.598e+02 13.434 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6600 on 13222 degrees of freedom  
## Multiple R-squared: 0.1904, Adjusted R-squared: 0.1899   
## F-statistic: 345.5 on 9 and 13222 DF, p-value: < 2.2e-16  
##   
##   
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS  
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:  
## Level of Significance = 0.05   
##   
## Call:  
## gvlma(x = lm.fit2)   
##   
## Value p-value Decision  
## Global Stat 3.472e+08 0 Assumptions NOT satisfied!  
## Skewness 8.821e+05 0 Assumptions NOT satisfied!  
## Kurtosis 3.463e+08 0 Assumptions NOT satisfied!  
## Link Function 8.256e+02 0 Assumptions NOT satisfied!  
## Heteroscedasticity 5.482e+02 0 Assumptions NOT satisfied!

# Anova stands for Analysis of Variation Model  
# It takes in a full model and the model that has been created by the user  
# and then comparison is made  
# For this model,A high RSS is achieved which clearly shows that this is not a correct predictive model  
  
  
anova(lm.fit2,lm.fit1)

## Analysis of Variance Table  
##   
## Model 1: RoomRent ~ Population + IsMetroCity + IsTouristDestination +   
## IsNewYearEve + StarRating + Airport + FreeWifi + HotelCapacity +   
## HasSwimmingPool  
## Model 2: RoomRent ~ StarRating + Airport + IsTouristDestination  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 13222 5.7602e+11   
## 2 13228 5.9940e+11 -6 -2.338e+10 89.443 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

detach(HotelData.df)  
  
#Detaching the model