**Final Project Title:**

**Hotel Room Pricing In the Indian Market**

State City

MH- Mumbai, Pune

Delhi

Goa

Kerala - Thrissur , Kochi, Munnar

UP- Lucknow, Kanpur, Agra, Varanasi

Tamilnadu- Madhurai, Thiruvananthpuram, Chennai, Ooty

MP- Indore

Punjab- Amritsar, Chandigarh

Karnataka- Banglore, Mysore, Manglore,

Gujarat- Rajkot, Ahemdabad, Surat

Rajasthan- Jodhpur, Udaipur, Jaisalmer,Surat

Jammu and Kashmir- Shrinagar

Assam- Guwahati

Odissa- Bhubaneshwar, Puri

Uttarakhand- Haridwar, Rishikesh, Nainital

Himachal Pradesh- Shimla, Manali

Haryana- Panchkula

West Bengal- Darjeeling, Kolkata

Sikkim- Gangtok

Telangana- Hyderabad

**Task 1-9**

**Task-1 Reading Data**

#Read data

cities42.data.raw.df <- read.csv(paste("Cities42.csv", sep=""))

cities42Temp.df <- read.csv(paste("Cities42.csv", sep="")) #a copy

#Output the number of missing values for each column (8 missing)

sapply(cities42.data.raw.df,function(x) sum(is.na(x)))

#Cleaning data

#omitting rows which have value NA in column HotelDescription

cities42.df <- na.omit(cities42.data.raw.df)

cities42Temp.df <- na.omit(cities42.data.raw.df)

#Checking new dimensions (now 13224 rows, before 13332 rows)

dim(cities42.df)

# Cleaning Data: Merging duplicate levels of Date,i.e. same dates in different format

ha <- list(

"04-Jan-17" = c("04-Jan-17", "Jan 04 2017", "Jan 4 2017"),

"08-Jan-17" = c("08-Jan-17", "Jan 08 2017", "Jan 8 2017"),

"18-Dec-16" = c("18-Dec-16", "Dec 18 2016"),

"21-Dec-16" = c("21-Dec-16", "Dec 21 2016"),

"24-Dec-16" = c("24-Dec-16", "Dec 24 2016"),

"25-Dec-16" = c("25-Dec-16", "Dec 25 2016"),

"28-Dec-16" = c("28-Dec-16", "Dec 28 2016"),

"31-Dec-16" = c("31-Dec-16", "Dec 31 2016")

)

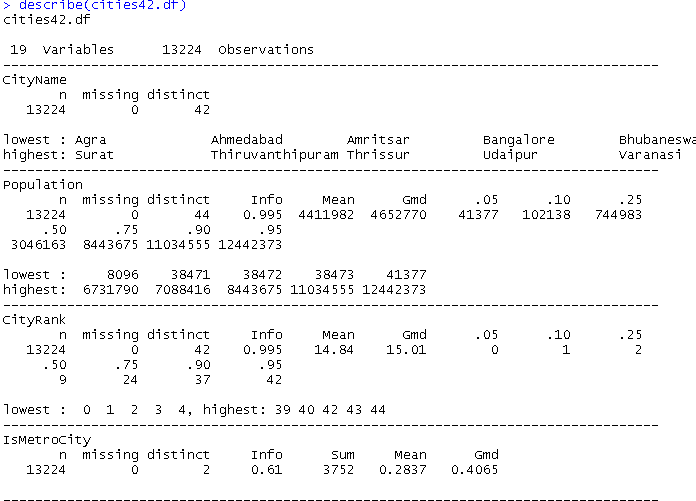
for (i in 1:length(ha)) levels(cities42.df$Date)[levels(cities42.df$Date)%in%ha[[i]]] <- names(ha)[i]

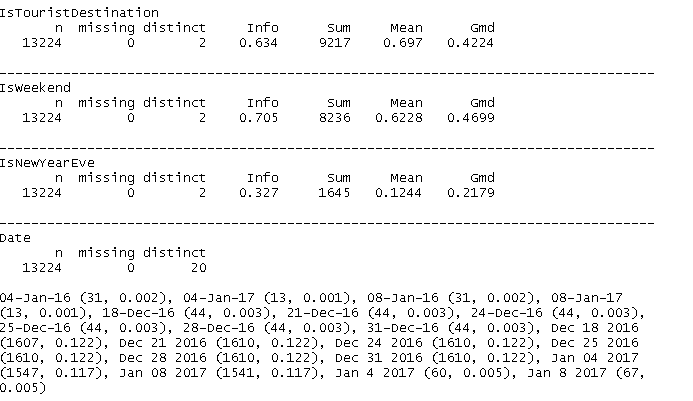
levels(cities42.df$Date)

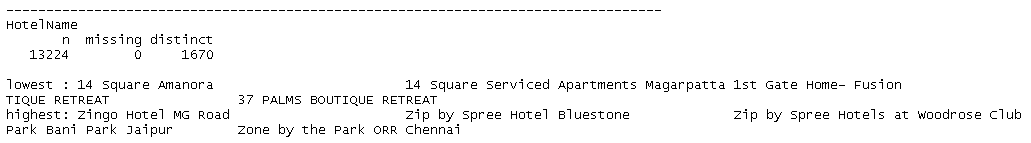
**Task-2**

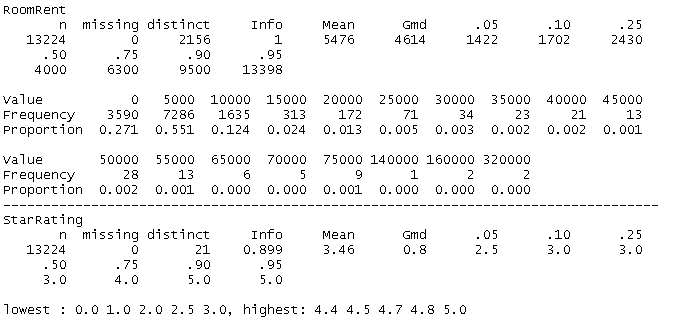
**Summarize the Data**

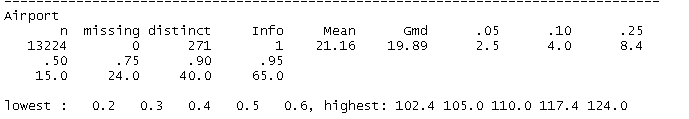
describe(cities42.df)

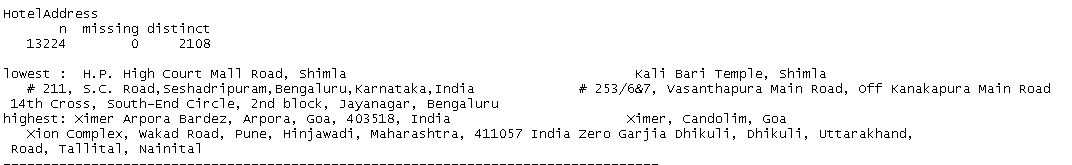


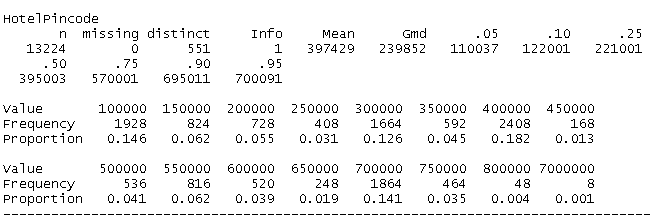


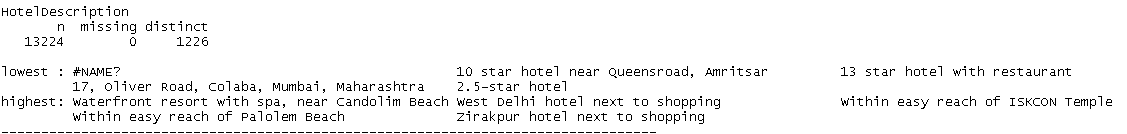


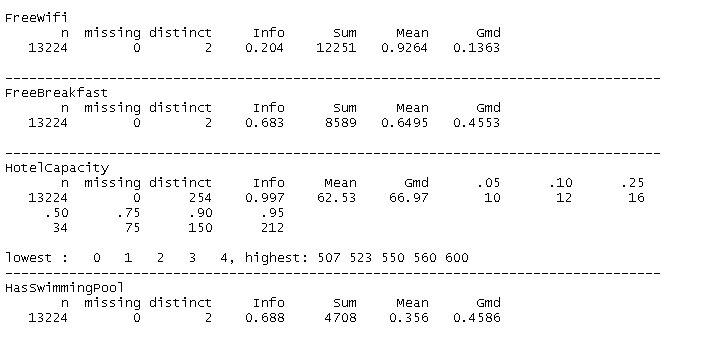












# Considering selected fields only

cities42Temp.df <- cities42.df[,c(2, 4:7,10:12, 16:19)]

describe(cities42Temp.df)

# Data Types

str(cities42.df)

## 'data.frame': 13224 obs. of 19 variables:

## $ 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 "04-Jan-16","04-Jan-17",..: 1## 1 12 13 14 15 16 17 18 11 12 ...

## $ HotelName : Factor w/ 1670 levels "14 Square Amanora",..: 163## 5 1635 1635 1635 1635 1635 1635 1635 1409 1409 ...

## $ RoomRent : int 12375 10250 9900 10350 12000 11475 11220 922## 5 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 40## 0011 400012 400039 400040 ...

## $ HotelDescription : Factor w/ 1226 levels "#NAME?","10 star hotel nea## r Queensroad, Amritsar",..: 1030 1030 1030 1030 1030 1030 1030 1030 1006 1## 006 ...

## $ 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 ...

**Task-3**

Y = F(x1, x2, x3, ....)

Y is dependednt variable; x1, x2, x3.. are independent variables

RoomRent = F {StarRating, CityName, IsMetroCity,………..}

**Task-4**

Dependent variable = Y = RoomRent

**Task-5**

**Picking 3 most important independent variables,**

These predictors seem of high importance: StarRating, IsTouristDestination, HotelName, HasSwimmingPool, IsMetroCity, ISNewYearEve…

Factor HotelName is accounted in StarRating

3 most important: StarRating,IsTouristDestination, HasSwimmingPool

RoomRent = Function (StarRating, IsTouristDestination, HasSwimmingPool)

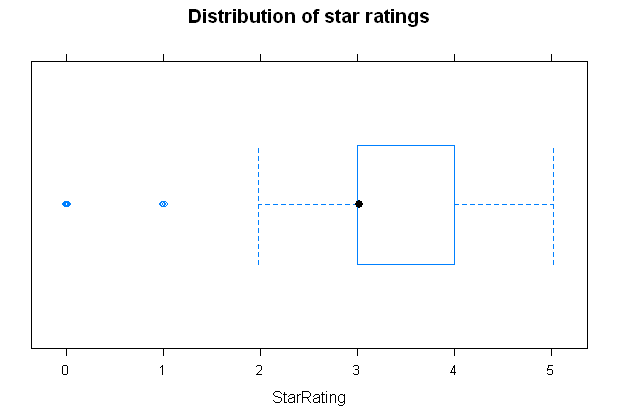
**Task-6**

**Visualizing StarRating, IsTouristDestination and HasSwimmingPool independently**

# Distribution of StarRating

bwplot(jitter(cities42.df$StarRating), horizontal=TRUE,

main = "Distribution of star ratings", xlab="StarRating")



Mean StarRating is around 3

# How many hotels are in location which are at tourist destination?

table(cities42.df$IsTouristDestination)

## 0 1

## 4007 9217

About 70% hotels are located in City which are tourist destination

# How many hotels have Swimming pool?

table(cities42.df$HasSwimmingPool)

## 0 1

## 8516 4708

Majority of hotels don’t have swimming pool.

**TASK-7**

**Scatter Plots to understand how are RoomRent, StarRating, IsTouristDestination and HasSwimmingPool correlated pair-wise**

**# Checking relation between RoomRent and StarRating**

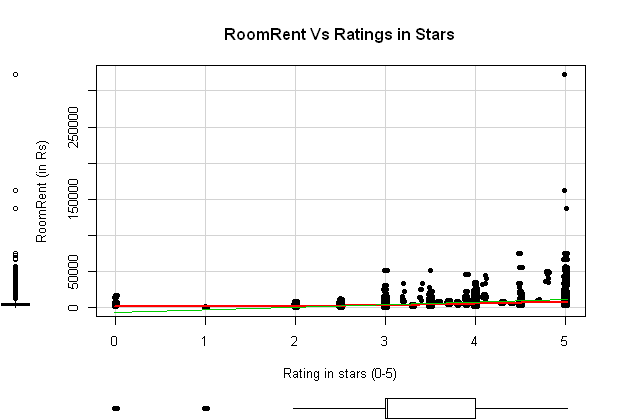
# Using Jitter

scatterplot ( jitter(RoomRent) ~ jitter(StarRating), data = cities42.df,

spread=FALSE, pch=19,

ylab="RoomRent (in Rs)",xlab = "Rating in stars (0-5)",

main = "RoomRent Vs Ratings in Stars")



**# Checking relation between RoomRent and Tourist Destination**

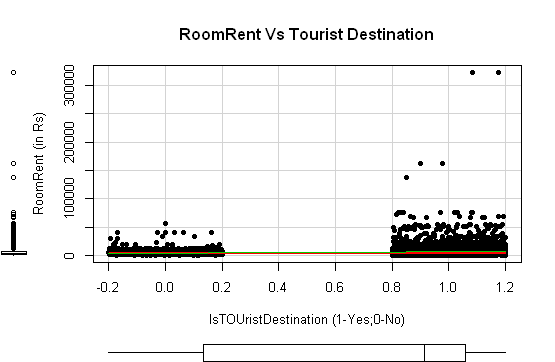
# Using Jitter

scatterplot (jitter(RoomRent)~jitter(IsTouristDestination), data = cities42.df,

spread=FALSE, pch=19,

ylab="RoomRent (in Rs)", xlab = "IsTouristDestination (1-Yes;0-No)",

main = "RoomRent Vs Tourist Destination")



# Mean RoomRent based on whether City is tourist destination or not

aggregate (RoomRent ~ IsTouristDestination, data = cities42.df, mean)

## IsTouristDestination RoomRent

## 1 0 4111.003

## 2 1 6068.946

**# Checking relation between RoomRent and HasSwimmingPool**

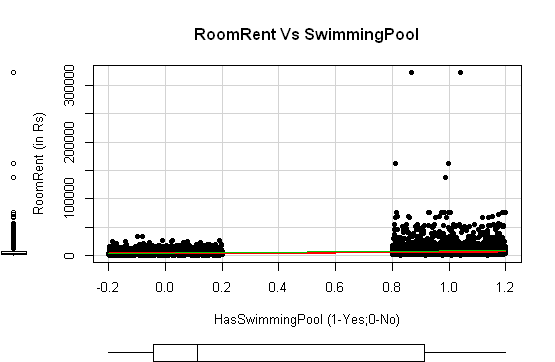
# Using Jitter

scatterplot (jitter(RoomRent)~jitter(HasSwimmingPool), data = cities42.df,

spread=FALSE, pch=19,

ylab="RoomRent (in Rs)", xlab = "HasSwimmingPool (1-Yes;0-No)",

main = "RoomRent Vs SwimmingPool")



# Mean RoomRent based on whether Hotel has Swimming pool or not

aggregate (RoomRent ~ HasSwimmingPool, data = cities42.df, mean)

## HasSwimmingPool RoomRent

## 1 0 3776.576

## 2 1 8549.052

**# Checking relation between StarRating and HasSwimmingPool**

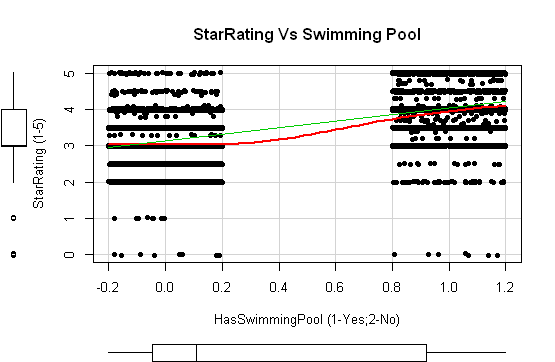
**#Using Jitter**

scatterplot ( jitter(StarRating) ~ jitter(HasSwimmingPool), data = cities42.df,

spread=FALSE, pch=19,

ylab="StarRating (1-5)", xlab = "HasSwimmingPool (1-Yes;2-No)",

main = "StarRating Vs Swimming Pool")



aggregate (StarRating ~ HasSwimmingPool, data = cities42.df, mean)

## HasSwimmingPool StarRating

## 1 0 3.112541

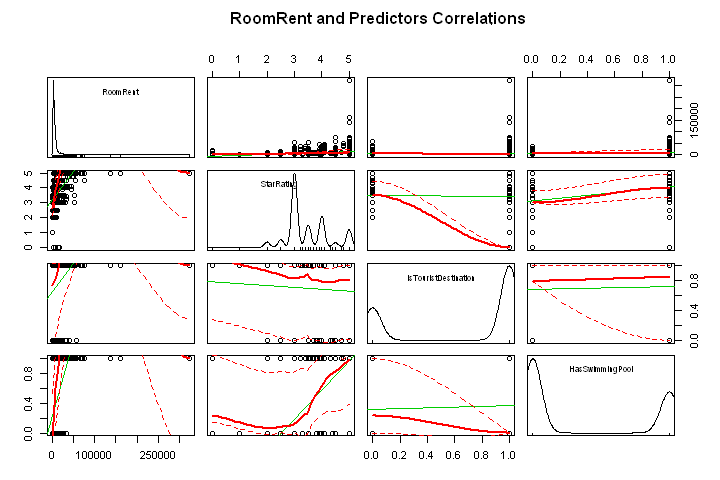
## 2 1 4.087978

**Scatterplot Matrix of RoomRent, RoomRent + StarRating + IsTouristDestination + HasSwimmingPool**

scatterplotMatrix(~RoomRent + StarRating + IsTouristDestination +

HasSwimmingPool, data=cities42.df,

main="RoomRent and predictors correlation")



**Task-8**

Corrgram of RoomRent, StarRating, IsTouristDestination and HasSwimmingPool

# Taking only dependent and independent variable

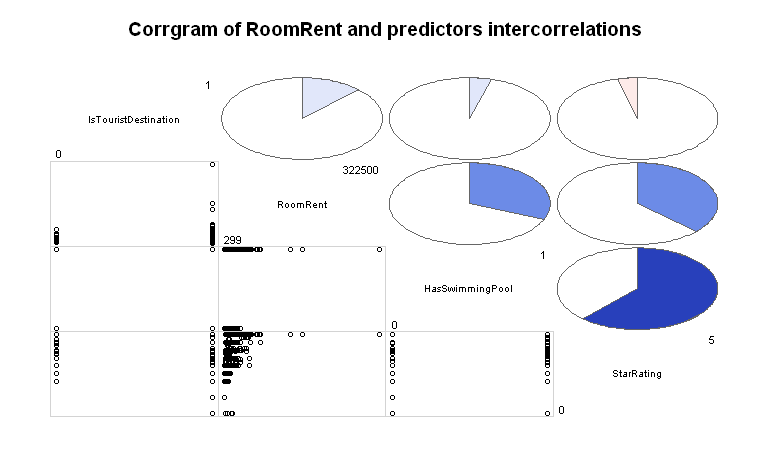
cities42Subset <- cities42.df[,c(5,10:11, 19)]

corrgram(cities42Subset, order=TRUE, lower.panel=panel.pts,

upper.panel=panel.pie, text.panel=panel.txt,

diag.panel=panel.minmax,

main="Corrgram of RoomRent and predictors intercorrelations")



Result:

Significant positive correlation of RoomRent with HasSwimmingPool and StarRating.

Weak positive correlation of Roomrent with IsTouristDestination.

Positive weak correlation of IsTouristDestination with SwimmingPool.

Negative weak correlation of IsTouristDestination with StarRating.

Strong positive correlation between HasSwimmingPool and StarRating

**Task-9**

Variance-Covariance Matrix for RoomRent, StarRating, IsTouristDestination and HasSwimmingPool

corr.test(cities42Subset, use="complete")

## Call:corr.test(x = cities42Subset, use = "complete")

## Correlation matrix

## IsTouristDestination RoomRent StarRating HasSwimmingPool

## IsTouristDestination 1.00 0.12 -0.04 0.04

## RoomRent 0.12 1.00 0.37 0.31

## StarRating -0.04 0.37 1.00 0.62

## HasSwimmingPool 0.04 0.31 0.62 1.00

## Sample Size

## [1] 13224

## Probability values (Entries above the diagonal are adjusted for multiple t## ests.)

## IsTouristDestination RoomRent StarRating HasSwimmingPool

## IsTouristDestination 0 0 0 0

## RoomRent 0 0 0 0

## StarRating 0 0 0 0

## HasSwimmingPool 0 0 0 0

**Task-10**

**Hypothesis (or two) that we could test using a Regression Model**

H1: Hotels in metro-cities charge higher rent for room than those which are not in metro cities.

H2: Hotels with swimming pool charge higher rent for room than those without one

**Task-11**

**T-tests appropriate to check hypothesis**

**H1: Hotels in metro-cities charge higher rent for room than those which are not in metro cities.**

The following code compares Metro-city Hotel (group 1) and non-metro-city Hotels (group 0) on rent of rooms, using a two-tailed test without the assumption of equal variances:

Testing with Original cleaned Data-set Cities42.df (Cities42B.df is used for Regression Model)

t.test(RoomRent ~ IsMetroCity, cities42.df)

##

## Welch Two Sample t-test

##

## data: RoomRent by IsMetroCity

## t = 10.674, df = 13218, p-value < 2.2e-16

## alternative hypothesis: true difference in means is not equal to 0

## 95 percent confidence interval:

## 883.6805 1281.2485

## sample estimates:

## mean in group 0 mean in group 1

## 5782.794 4700.329

Result: P < 0.01, so we reject NULL hypothesis that both groups are equal or both groups have equal rents. Mean difference is of statistical significance. Surprisingly, mean of group 0 (not in metrocity) is higher. It could be because of large number of hotel

Now, using Data-set (cities42B.df), i.e. after removing very high RoomRent values (quantiles above .95). RegressionModel is based on this data-set.

t.test(RoomRent ~ IsMetroCity, cities42B.df)

##

## Welch Two Sample t-test

##

## data: RoomRent by IsMetroCity

## t = -0.62006, df = 6774.3, p-value = 0.5352

## alternative hypothesis: true difference in means is not equal to 0

## 95 percent confidence interval:

## -132.31114 68.72257

## sample estimates:

## mean in group 0 mean in group 1

## 4408.155 4439.949

Result: p-value > 0.05. We fail to reject hypothesis that these two groups have equal room-rents. Mean difference is not of statistical significance.

**H2: Hotels with swimming pool charge higher rent for room than those without one**

The following code compares Hotel with swimming-pool (group 1) and Hotels without it (group 0) on rent of rooms, using a two-tailed test without the assumption of equal variances:

Testing with Original cleaned Data-set Cities42.df (Cities42B.df is used for Regression Model)

t.test(RoomRent ~ HasSwimmingPool, cities42.df)

##

## Welch Two Sample t-test

##

## data: RoomRent by HasSwimmingPool

## t = -29.007, df = 5011.8, p-value < 2.2e-16

## alternative hypothesis: true difference in means is not equal to 0

## 95 percent confidence interval:

## -5095.028 -4449.923

## sample estimates:

## mean in group 0 mean in group 1

## 3776.576 8549.052

Now, using Data-set (cities42B.df), i.e. after removing very high RoomRent values (quantiles above .95). RegressionModel is based on this data-set.

t.test(RoomRent ~ HasSwimmingPool, cities42B.df)

##

## Welch Two Sample t-test

##

## data: RoomRent by HasSwimmingPool

## t = -51.624, df = 7275.1, p-value < 2.2e-16

## alternative hypothesis: true difference in means is not equal to 0

## 95 percent confidence interval:

## -2499.143 -2316.289

## sample estimates:

## mean in group 0 mean in group 1

## 3623.442 6031.159

Result: p<0.01, so we reject NULL hypoyhesis that two groups are equal. Mean difference is statistical significance.

**Task-12**

**Formulating Regression Model**

y = b0 + b1\*x1 + b2\*x2 + ..

y = RoomRent

x = {x1, x2, ..} = {CityName + StarRating + ISMetroCity + Date + ……. }

**Task-13**

**Fitting Linear Regression Model**

**# Model containing all predictors, except HotelDescription, HotelAddress,**

# **HotelPincode**

Model1 <- RoomRent ~ CityName + Population + CityRank + IsMetroCity + IsTouristDestination + IsWeekend + IsNewYearEve + Date +

HotelName + StarRating + Airport + HotelPincode + FreeWifi +

FreeBreakfast + HotelCapacity + HasSwimmingPool

fit1 <-lm(Model1, data = cities42.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.765, Adjusted R-squared: 0.731

**# Removing outlier in RoomRent, which are six digit values**

**# cities42.df <- na.omit(cities42.data.raw.df)**

cityOmitHighRent.df <- cities42.df[which(cities42.df$RoomRent <99999),]

Model1 <- RoomRent ~ CityName + Population + CityRank + IsMetroCity + IsTouristDestination + IsWeekend + IsNewYearEve + Date +

HotelName + StarRating + Airport + HotelPincode + FreeWifi +

FreeBreakfast + HotelCapacity + HasSwimmingPool

fit1 <-lm(Model1, data = cityOmitHighRent.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.881, Adjusted R-squared: 0.864

**# Final model after filtering it**

Model1 <- RoomRent ~ StarRating + HotelName + Date

fit1 <-lm(Model1, data = cityOmitHighRent.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.88, Adjusted R-squared: 0.862

Result: But I can't interpret HotelName in Final Regression Model (y = b0 + b1\*x1 + b2\*x2...), so I am rejecting this Model.

**# Testing model based on 3 most important metrics chosen in Task-5**

Model1 <- RoomRent ~ StarRating + IsTouristDestination + HasSwimmingPool

fit1 <-lm(Model1, data = cityOmitHighRent.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.234

Result: It is poor model. I conclude that it is not sensitive to variations.

**# Again starting from zero by involving all predictors, except those**

**# which can't be**

**# interpreted.**

**# Removing Columns (HotelDescription, HotelAddress, HotelName,**

**# Pincode)** **whose**

**# impact can't be interpreted in model**

Model1 <- RoomRent ~ CityName + Population + CityRank + IsMetroCity + IsTouristDestination + IsWeekend + IsNewYearEve + Date +

StarRating + Airport + FreeWifi + FreeBreakfast + HotelCapacity + HasSwimmingPool

fit1 <-lm(Model1, data = cityOmitHighRent.df)

summary(fit1)

##

## Multiple R-squared: 0.326

**# Excluding outliers which aren't important exceptions; like frauds in**

**# bank data**

**# analysis, which we must keep.**

**# Excluding RoomRent quantile over 0.95 (at 0.95 value is 13398)**

describe(cities42.df$RoomRent)

cities42B.df <- cities42.df[which(cities42.df$RoomRent < 13500), ]

fit1 <-lm(Model1, data = cities42B.df)

summary(fit1)

##

## Multiple R-squared: 0.4911

**Final filtered Model (after removing not important predictors)**

Model1 <- RoomRent ~ CityName + Date + StarRating + HasSwimmingPool

fit1 <-lm(Model1, data = cities42B.df)

summary(fit1) # Multiple R-squared: 0.4818  
##

## Call:

## lm(formula = Model1, data = cities42B.df)

##

## Residuals:

## Min 1Q Median 3Q Max

## -6013.8 -1202.7 -316.8 840.0 9576.0

##

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) -1940.651 368.523 -5.266 1.42e-07 \*\*\*

## CityNameAhmedabad 719.369 129.705 5.546 2.98e-08 \*\*\*

## CityNameAmritsar 403.864 185.136 2.181 0.029169 \*

## CityNameBangalore 295.614 120.153 2.460 0.013895 \*

## CityNameBhubaneswar 901.869 194.335 4.641 3.51e-06 \*\*\*

## CityNameChandigarh 685.261 137.882 4.970 6.79e-07 \*\*\*

## CityNameChennai 355.303 129.942 2.734 0.006260 \*\*

## CityNameDarjeeling 2597.390 187.254 13.871 < 2e-16 \*\*\*

## CityNameDelhi 573.970 100.811 5.694 1.27e-08 \*\*\*

## CityNameGangtok 1937.896 189.755 10.213 < 2e-16 \*\*\*

## CityNameGoa 3128.471 123.293 25.374 < 2e-16 \*\*\*

## CityNameGuwahati 1652.040 286.081 5.775 7.89e-09 \*\*\*

## CityNameHaridwar 624.959 286.370 2.182 0.029103 \*

## CityNameHyderabad 156.159 122.889 1.271 0.203849

## CityNameIndore 884.659 174.872 5.059 4.28e-07 \*\*\*

## CityNameJaipur 810.554 115.834 6.998 2.74e-12 \*\*\*

## CityNameJaisalmer 2174.105 152.578 14.249 < 2e-16 \*\*\*

## CityNameJodhpur 2146.078 166.161 12.916 < 2e-16 \*\*\*

## CityNameKanpur 866.503 478.274 1.812 0.070052 .

## CityNameKochi 919.890 121.854 7.549 4.69e-14 \*\*\*

## CityNameKolkata 1273.092 123.603 10.300 < 2e-16 \*\*\*

## CityNameLucknow 2593.378 190.789 13.593 < 2e-16 \*\*\*

## CityNameMadurai 1945.784 200.341 9.712 < 2e-16 \*\*\*

## CityNameManali 2098.588 144.393 14.534 < 2e-16 \*\*\*

## CityNameMangalore 1211.917 205.610 5.894 3.86e-09 \*\*\*

## CityNameMumbai 2613.145 116.645 22.403 < 2e-16 \*\*\*

## CityNameMunnar 4452.624 141.392 31.491 < 2e-16 \*\*\*

## CityNameMysore 586.391 175.188 3.347 0.000819 \*\*\*

## CityNameNainital 3008.302 185.544 16.213 < 2e-16 \*\*\*

## CityNameOoty 3040.136 188.739 16.108 < 2e-16 \*\*\*

## CityNamePanchkula -7.951 251.843 -0.032 0.974815

## CityNamePune 630.882 119.266 5.290 1.25e-07 \*\*\*

## CityNamePuri 2733.466 267.273 10.227 < 2e-16 \*\*\*

## CityNameRajkot 81.147 193.138 0.420 0.674384

## CityNameRishikesh 2052.224 221.059 9.284 < 2e-16 \*\*\*

## CityNameShimla 2420.207 147.469 16.412 < 2e-16 \*\*\*

## CityNameSrinagar 3740.211 387.202 9.660 < 2e-16 \*\*\*

## CityNameSurat 639.282 229.114 2.790 0.005275 \*\*

## CityNameThiruvanthipuram 165.907 137.128 1.210 0.226353

## CityNameThrissur 405.794 344.868 1.177 0.239352

## CityNameUdaipur 2500.206 134.731 18.557 < 2e-16 \*\*\*

## CityNameVaranasi 1081.903 155.960 6.937 4.20e-12 \*\*\*

## Date04-Jan-17 -653.635 348.956 -1.873 0.061076 .

## Date08-Jan-16 168.871 476.717 0.354 0.723167

## Date08-Jan-17 -756.822 348.933 -2.169 0.030104 \*

## Date18-Dec-16 -1410.222 348.550 -4.046 5.24e-05 \*\*\*

## Date21-Dec-16 -1283.381 348.546 -3.682 0.000232 \*\*\*

## Date24-Dec-16 -973.514 348.601 -2.793 0.005236 \*\*

## Date25-Dec-16 -937.364 348.582 -2.689 0.007174 \*\*

## Date28-Dec-16 -936.876 348.588 -2.688 0.007206 \*\*

## Date31-Dec-16 -706.008 348.647 -2.025 0.042889 \*

## StarRating 1679.979 30.665 54.785 < 2e-16 \*\*\*

## HasSwimmingPool 1037.584 48.667 21.320 < 2e-16 \*\*\*

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1877 on 12513 degrees of freedom

Multiple R-squared: 0.4811, Adjusted R-squared: 0.479

F-statistic: 223.1 on 52 and 12513 DF, p-value: < 2.2e-16

Interpreting b2 \*CityName:

ci is regression coefficient for particular City. Term CityNamei is 1 for corresponding city, for other cities it will be zero.

For example: If city is Varanasi, then = 1, and for rest city = 0. Only coefficient of Varanasi remains.

Similarly b1\*Date is Interpreted

di is regression coefficient for particular date. Term datei is 1 for corresponding date, for other dates it will be zero.

#Comparing RoomRent with Fitted Values

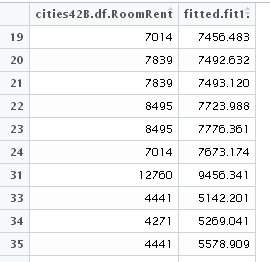
#Comparing RoomRent predicted by the model with the actual RoomRent given in the data

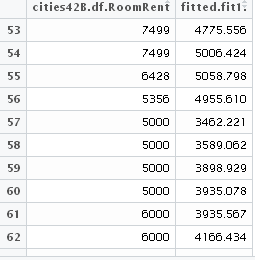
predictedRoomRent = data.frame(fitted(fit1))

actualRoomRent = data.frame(cities42B.df$RoomRent)

RoomRentComparison = cbind(actualRoomRent, predictedRoomRent)

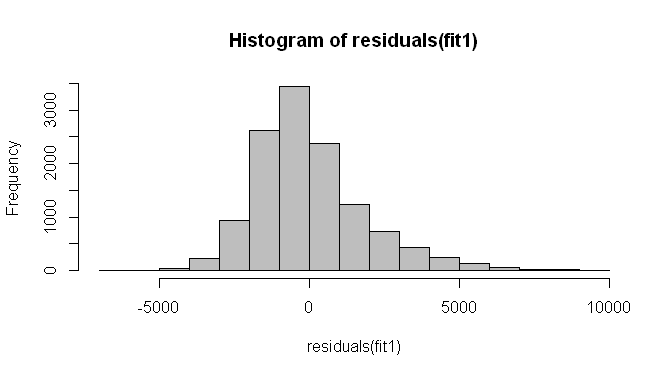
View(RoomRentComparison)





**#Histogram of residuals**

hist(residuals(fit1), col = "gray")



# majority of residuals are between -2500 to +2500

**Framework to test cities individually**

#Framework to test cities individually

#Reading Data

cityMumbai.df <- read.csv(paste("0 Mumbai.csv", sep=""))

cityIndore.df <- read.csv(paste("11 Lucknow.csv", sep=""))

cityLucknow.df <- read.csv(paste("14 Indore.csv", sep=""))

cityDelhi.df <- read.csv(paste("1 Delhi.csv", sep=""))

cityJaipur.df <- read.csv(paste("9 Jaipur.csv", sep=""))

cityBangalore.df <- read.csv(paste("2 Bangalore.csv", sep=""))

#Cleaning of NA value

cityMumbai.df <- na.omit(cityMumbai.df)

cityIndore.df <- na.omit(cityIndore.df)

cityLucknow.df <- na.omit(cityLucknow.df)

cityDelhi.df <- na.omit(cityDelhi.df)

cityJaipur.df <- na.omit(cityJaipur.df)

cityBangalore.df <- na.omit(cityBangalore.df)

**# Testing model city-wise**

# Now predictors are {Date, StarRating, SwimmingPool}

cityMumbaiB.df <- cityMumbai.df[which(cityMumbai.df$RoomRent < 13500),]

Model1 <- RoomRent ~ Date + StarRating + FreeBreakfast+ + Airport + HotelCapacity

fit1 <-lm(Model1, data = cityMumbaiB.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.42

cityIndoreB.df <- cityIndore.df[which(cityIndore.df$RoomRent < 13500),]

Model1 <- RoomRent ~ Date + StarRating + FreeBreakfast+ + Airport + HotelCapacity

fit1 <-lm(Model1, data = cityIndoreB.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.55

cityLucknowB.df <- cityLucknow.df[which(cityLucknow.df$RoomRent < 13500),]

Model1 <- RoomRent ~ Date + StarRating + FreeBreakfast+ + Airport + HotelCapacity

fit1 <-lm(Model1, data = cityLucknowB.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.48

cityDelhiB.df <- cityDelhi.df[which(cityDelhi.df$RoomRent < 13500),]

Model1 <- RoomRent ~ Date + StarRating + FreeBreakfast+ + Airport + HotelCapacity

fit1 <-lm(Model1, data = cityDelhiB.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.6

cityJaipurB.df <- cityJaipur.df[which(cityJaipur.df$RoomRent < 13500),]

Model1 <- RoomRent ~ Date + StarRating + FreeBreakfast+ + Airport + HotelCapacity

fit1 <-lm(Model1, data = cityJaipurB.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.46

cityBangaloreB.df <- cityBangalore.df[which(cityBangalore.df$RoomRent < 13500),]

Model1 <- RoomRent ~ Date + StarRating + FreeBreakfast+ + Airport + HotelCapacity

fit1 <-lm(Model1, data = cityBangaloreB.df)

summary(fit1)

## <omitting rest>

## Multiple R-squared: 0.56

**Task-14**

**Using Linear Regression Model outputs to test Hypothesis and draw references**

Linear Regression Model:

H1: Hotels in metro-cities charge higher rent for room than those which are not in metro cities.

IsMetroCity doesn’t infuences RoomRent. So, I reject this hypothesis. T-test gave same result.

H2: Hotels with swimming pool charge higher rent for room than those without one

HasSwimmingPool influences RoomRent, and increases rent by 1037. So, this hypothesis is true. T-test gave same result.

**Task 15**

**List of insights based on Regression Analysis**

Linear Regression Model:

Date influences hotel room pricing strategy

City influences hotel room pricing strategy.

Under similar conditions, Increase in StarRating causes rent to go upwards

Under similar conditions, availability of swimming pool causes rent to go upwards

There is customer base which seeks luxurious amentites. So, if more amenities are added in hotel these customers would be willing to pay.  
  
Then there are customer base who are more interested in economic option. If more economic amenties (more value for less money) are added then these customers might be willing to pay more.