Apartment-Rent-Regression

Paula Ramirez - Student id 8963215

12/11/2024

Apartment-Rent-Regression

1. Preliminary and Exploratory

1. Rename all variables

```
#Reading the file
lease_data_PR <- read.table(here("Apartment-Rent-Regression", "Rent.txt"),
                                    header = TRUE, sep = ",")
#Converting it to dataframe.
lease_data_PR <- as.data.frame(lease_data_PR)
#Append PR initials to all variables in the dataframe
colnames(lease_data_PR) <- paste(colnames(lease_data_PR), "PR", sep = "_")
#Changing to factor
lease_data_PR <-as.data.frame(unclass(lease_data_PR), stringsAsFactors = TRUE)
str(lease_data_PR)
```

```
## 'data.frame':    1051 obs. of  9 variables:
##  $ Prc_PR     : num  596 3130 2300 2840 2790 3230 2240 2030 1800 1600 ...
##  $ Bed_PR     : int  3 2 1 1 4 3 1 3 4 1 ...
##  $ floor_PR   : int  3 5 4 2 1 5 1 2 1 1 ...
##  $ TotFloor_PR: int  4 7 6 4 2 5 4 4 5 4 ...
##  $ Bath_PR    : int  4 2 2 1 3 2 1 1 1 1 ...
##  $ Sqft_PR    : int  501 1275 982 2418 1655 1024 864 671 609 834 ...
##  $ City_PR    : Factor w/ 3 levels "Blossomville",..: 1 3 2 3 3 2 3 1 2 3 ...
##  $ Comp_PR    : Factor w/ 2 levels "Leaseflow","Rentopia": 1 2 1 1 2 1 2 2 2 2 ...
##  $ Dist_PR    : num  10.6 1.4 6.8 0.4 5.3 0.5 3.5 3.6 1.4 4 ...
```

```
#Showing first results
head(lease_data_PR)
```

```
##   Prc_PR Bed_PR floor_PR TotFloor_PR Bath_PR Sqft_PR      City_PR   Comp_PR
## 1    596      3        3           4       4     501 Blossomville Leaseflow
## 2   3130      2        5           7       2    1275    Terranova  Rentopia
## 3   2300      1        4           6       2     982    Riverport Leaseflow
## 4   2840      1        2           4       1    2418    Terranova Leaseflow
## 5   2790      4        1           2       3    1655    Terranova  Rentopia
## 6   3230      3        5           5       2    1024    Riverport Leaseflow
```

```
##   Dist_PR
## 1    10.6
## 2     1.4
## 3     6.8
## 4     0.4
## 5     5.3
## 6     0.5
```

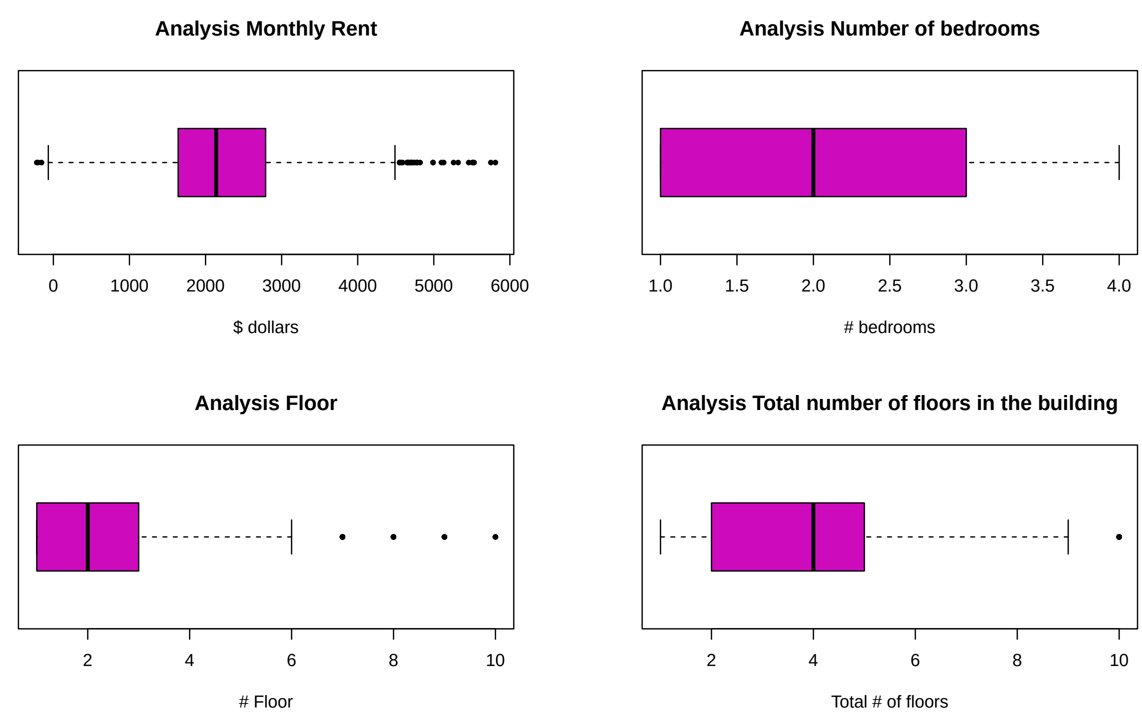
2. Graphical and Exploratory Data Summaries

```r
# Evaluating atypical errors
round(stat.desc(lease_data_PR),2)
```

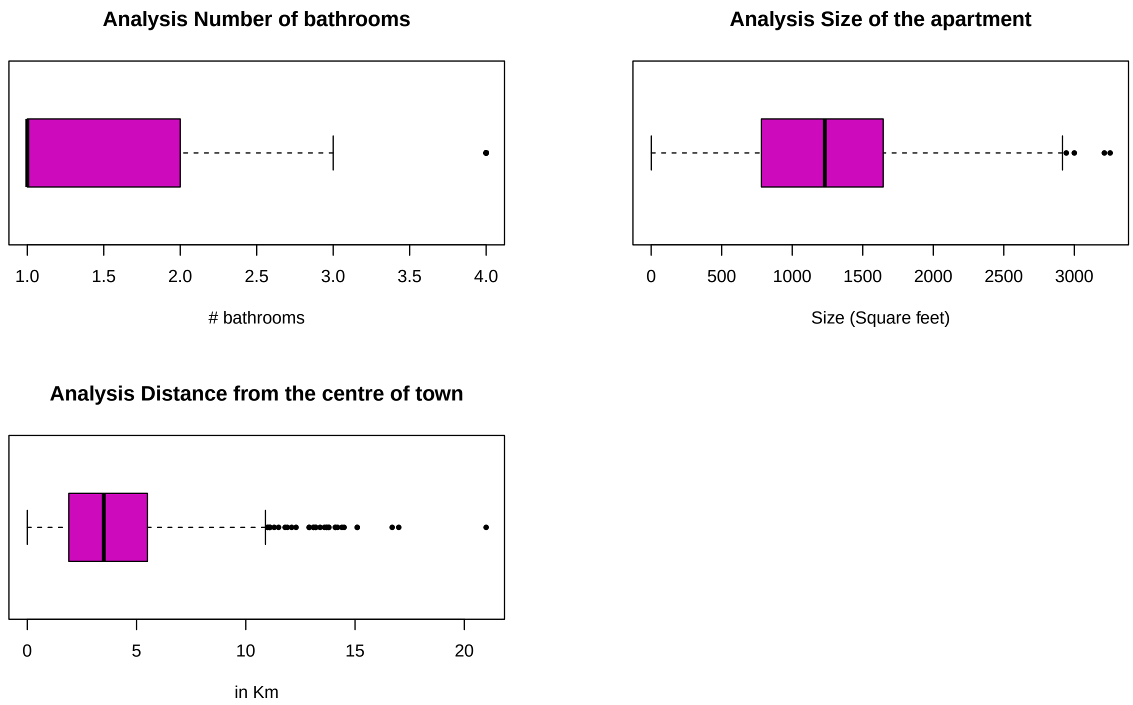
```
##                 Prc_PR  Bed_PR floor_PR TotFloor_PR Bath_PR    Sqft_PR City_PR
## nbr.val        1051.00 1051.00  1051.00     1051.00 1051.00    1051.00      NA
## nbr.null          0.00    0.00     0.00        0.00    0.00       0.00      NA
## nbr.na            0.00    0.00     0.00        0.00    0.00       0.00      NA
## min            -218.00    1.00     1.00        1.00    1.00       1.00      NA
## max            5810.00    4.00    10.00       10.00    4.00    3254.00      NA
## range          6028.00    3.00     9.00        9.00    3.00    3253.00      NA
## sum         2375179.70 2236.00  2388.00     4097.00 1701.00 1354839.00      NA
## median         2140.00    2.00     2.00        4.00    1.00    1231.00      NA
## mean           2259.92    2.13     2.27        3.90    1.62    1289.10      NA
## SE.mean          29.49    0.04     0.05        0.06    0.03      17.01      NA
## CI.mean.0.95     57.87    0.07     0.10        0.13    0.06      33.39      NA
## var          914096.29    1.37     2.77        4.33    0.96  304237.69      NA
## std.dev         956.08    1.17     1.66        2.08    0.98     551.58      NA
## coef.var          0.42    0.55     0.73        0.53    0.60       0.43      NA
##             Comp_PR Dist_PR
## nbr.val          NA 1051.00
## nbr.null         NA    1.00
## nbr.na           NA    0.00
## min              NA    0.00
## max              NA   21.00
## range            NA   21.00
## sum              NA 4319.60
## median           NA    3.50
## mean             NA    4.11
## SE.mean          NA    0.09
## CI.mean.0.95     NA    0.18
## var              NA    8.60
## std.dev          NA    2.93
## coef.var         NA    0.71
```

```r
# Creating vector to names for each chart
names_variables_PR <- c("Monthly Rent",
                        "Number of bedrooms",
                        "Floor",
                        "Total number of floors in the building",
                        "Number of bathrooms",
                        "Size of the apartment",
                        "City",
```

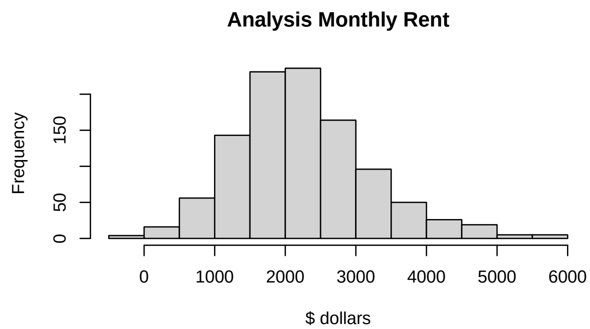
```
                          "Leasing company",
                          "Distance from the centre of town"
                          )

# Creating x label for each chart
x_labels_PR <- c("$ dollars",
                  "# bedrooms",
                  "# Floor",
                  "Total # of floors",
                  "# bathrooms",
                  "Size (Square feet)",
                  "City",
                  "Leasing company count",
                  "in Km"
                )

# Creating plots
par(mfrow=c(2,2))

for (i in 1:ncol(lease_data_PR)) {
    if (is.numeric(lease_data_PR[,i])) {
      boxplot(lease_data_PR[,i],
              main=paste("Analysis", names_variables_PR[i]),
              xlab=x_labels_PR[i],
              horizontal=TRUE,
              pch=20,
              col=6)
  }
}
```

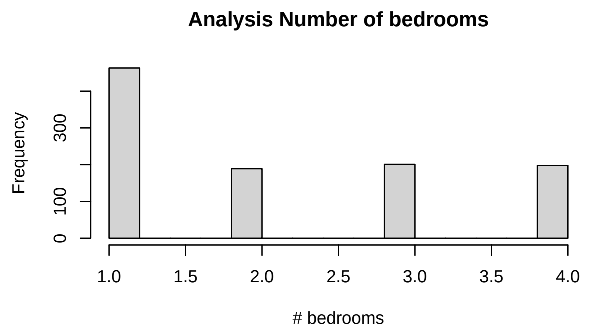


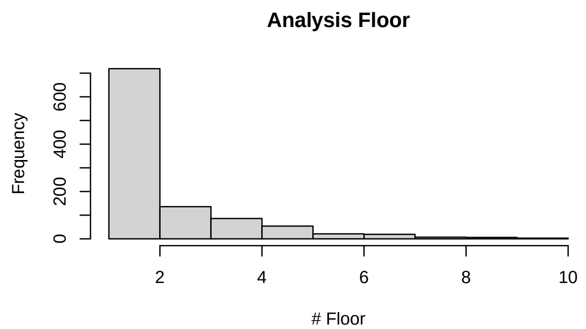
```r
par(mfrow=c(2,2))
```

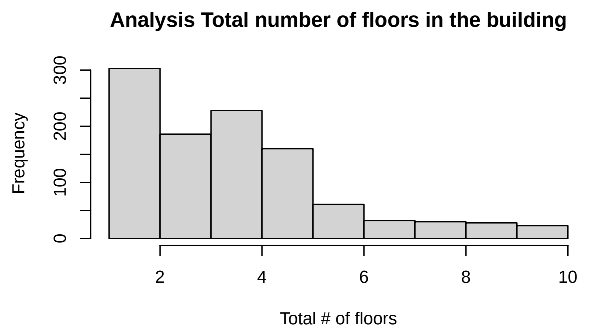


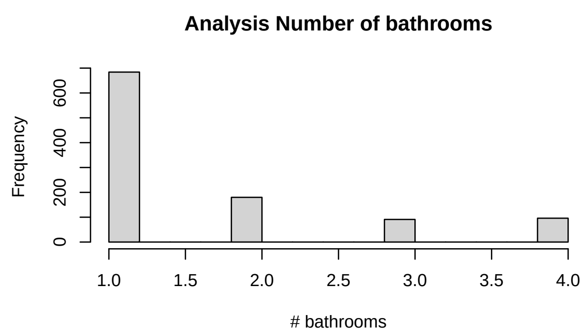

```r
# Creating histograms
par(mfrow=c(2,2))
for (i in 1:ncol(lease_data_PR)) {
  if (is.numeric(lease_data_PR[,i])) {
    hist(lease_data_PR[,i], main=paste("Analysis", names_variables_PR[i]),xlab=x_labels_PR[i])
  }
}
```

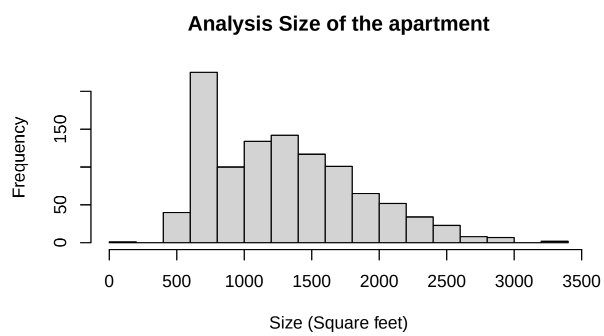


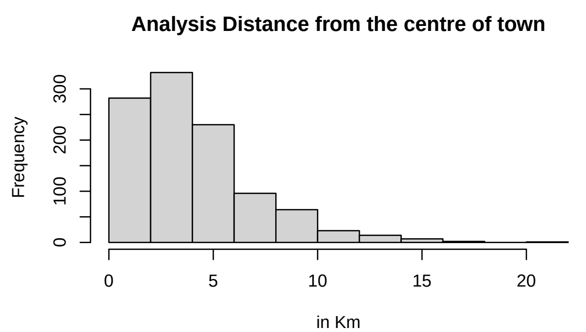












Observations and findings:

- *Total # Floor and Bathrooms*: Most buildings in the dataset have between 2 and 5 floors, this is the IQR or 50% of the data set. Regarding to bathrooms, some properties have a higher number of bathrooms (4) but is not unusual.

- *Distance of center of town*: 50% of the properties are located between ~2.5 and 5 km from the center of town.

Outliers: I found some relevant outliers in the data.

- *Monthly Rent (Prc_PR)*: Some observations have rental prices that are below to 0.
- *Size (Sqft_PR)*: Some properties have a larger size compared to the median, which is not unusual. However, some apartments have a listed size of 0 square feet.

To get more details I will create a density plot for these variables.

```
#RENTAL MONTHLY PRICE
#Density Plot - looking for more details in rental prices
densityplot( ~ lease_data_PR$Prc_PR, pch=3,
main='Details in Monthly rent Data',
xlab="Price in dollars",
col=4)
```


```
head(lease_data_PR[order(lease_data_PR$Prc_PR),c("Prc_PR", "Bed_PR", "floor_PR", "TotFloor_PR", "Bath_Pl
```

```
##     Prc_PR Bed_PR floor_PR TotFloor_PR Bath_PR Sqft_PR Dist_PR
## 520 -218.0      4        1           1       1     637     9.5
## 181 -198.0      3        3           5       1    1198    10.9
## 132 -156.0      4        5           9       1    1304    17.0
## 157  -66.3      3        1           2       1     690    10.0
## 736  255.0      4        2           5       1     662     2.0
## 144  322.0      1        2           3       1     550    13.2
```

```
## 359  352.0      1        1           3       1     646     6.1
## 173  373.0      1        1           3       1     509     7.3
```

```
unusual_price_pr <- which(lease_data_PR$Prc_PR <= 0)
unusual_price_pr
```

```
## [1] 132 157 181 520
```

```
# SIZE OF APARTMENTS
#Density Plot - looking for more details in size
densityplot( ~ lease_data_PR$Sqft_PR, pch=3,
main='Details in Size of the apartment',
xlab="in Square feet",
col=2)
```


```
head(lease_data_PR[order(lease_data_PR$Sqft_PR),c("Prc_PR", "Bed_PR", "floor_PR", "TotFloor_PR", "Bath_
```

```
##     Prc_PR Bed_PR floor_PR TotFloor_PR Bath_PR Sqft_PR Dist_PR
## 145   3080      1        3           4       2       1     6.7
## 1      596      3        3           4       4     501    10.6
## 719   2110      2        2           2       2     501     0.7
## 828   1950      4        2           2       1     505     4.2
## 420   2420      1        3           4       1     506     5.6
```

```
head(lease_data_PR[rev(order(lease_data_PR$Sqft_PR)),c("Prc_PR", "Bed_PR", "floor_PR", "TotFloor_PR", "
```

```
##     Prc_PR Bed_PR floor_PR TotFloor_PR Bath_PR Sqft_PR Dist_PR
```

```
## 225    1700       2          1             1        1     3254      3.5
## 216    2960       2          1             3        1     3213      3.0
## 401    3080       1          2             5        1     3000      2.4
## 869    2670       1          4             5        1     2943      3.5
## 821    3400       1          1             3        4     2916      5.5
```

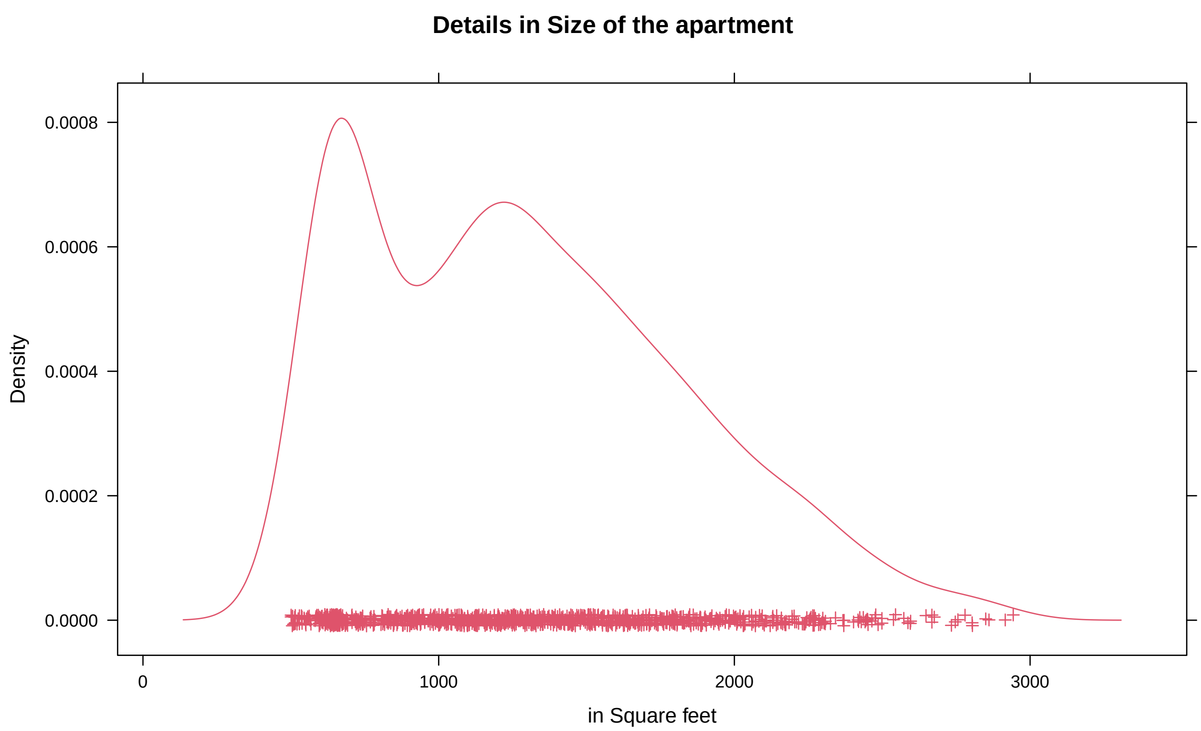
```
unusual_sqft_pr <- which(lease_data_PR$Sqft_PR <= 250 | lease_data_PR$Sqft_PR >=3000)
unusual_sqft_pr
```

```
## [1] 145 216 225 401
```

Decisions In those density plots, I found data points that were significantly distant from the other observations. Due to this, I decided to remove the data set points with rental prices bellow to 0 and property sizes bellow 250 or above 3000 square feet.

```
#DELETING ODD VALUES
lease_data_PR <- lease_data_PR[-c(unusual_sqft_pr,unusual_price_pr),]

#AFTER DELETION
#RENTAL MONTHLY PRICE
#Density Plot - looking for more details in rental prices
densityplot( ~ lease_data_PR$Prc_PR, pch=3,
main='Details in Monthly rent Data',
xlab="Price in dollars",
col=4)
```

```r
# SIZE OF APARTMENTS
#Density Plot - looking for more details in size
densityplot( ~ lease_data_PR$Sqft_PR, pch=3,
main='Details in Size of the apartment',
xlab="in Square feet",
col=2)
```



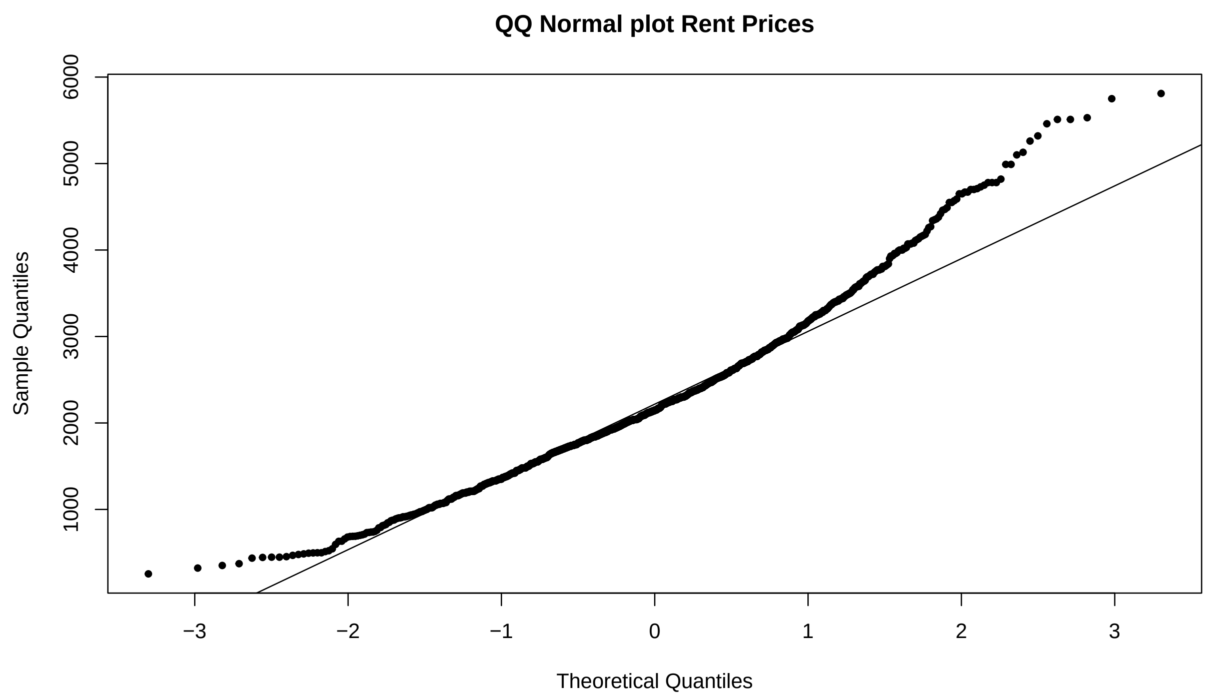
3. Analysis main companies

Trying to execute T-Test…

```r
# Identify rent prices between the two companies

#Shapiro test
shapiro.test(lease_data_PR$Prc_PR)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lease_data_PR$Prc_PR
## W = 0.97063, p-value = 0.0000000000001099
```

```r
#Checking normal distribution
qqnorm(lease_data_PR$Prc_PR, main="QQ Normal plot Rent Prices", pch=20)
qqline(lease_data_PR$Prc_PR)
```




```r
#Comparing Variance F-Test
var.test(Prc_PR ~ Comp_PR, data=lease_data_PR)
```

```
##
##  F test to compare two variances
##
## data:  Prc_PR by Comp_PR
## F = 1.1642, num df = 360, denom df = 681, p-value = 0.09483
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9740584 1.3985304
## sample estimates:
## ratio of variances
##           1.164177
```

Explanations I found the following results for each assumption fro T-test:

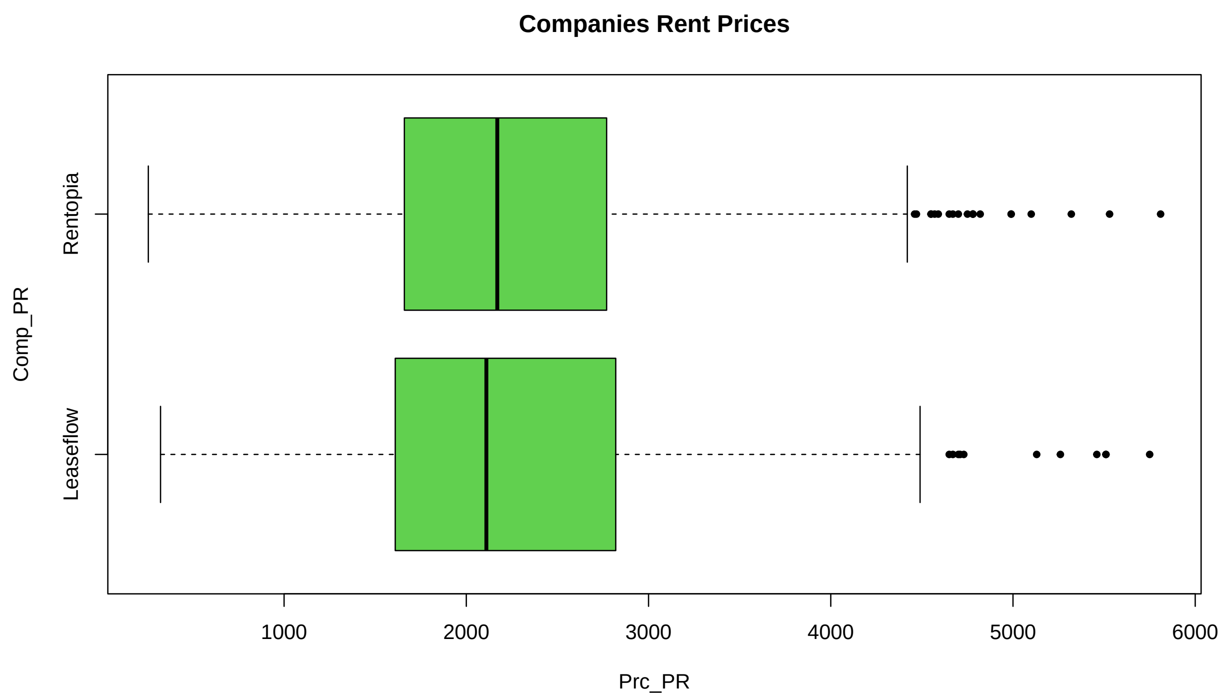
1. Data are independent –> PASS
2. Data is NOT normal distributed. The *S2_T_PR* did not passed the Shapiro Test because p-value is < 0.05 (I rejected the hipothesis) and QQ Normal plot shows a deviation from the diagonal line. FAIL
3. F-Test –> PASS p-value $= 0.1227 > 0.05$. The variances of the prices in both companies are equal (96% confident)

```r
#Wilcoxon test
wilcox.test(Prc_PR ~ Comp_PR, data=lease_data_PR)
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  Prc_PR by Comp_PR
## W = 120436, p-value = 0.5648
## alternative hypothesis: true location shift is not equal to 0
```

```r
#showing box plot
boxplot(Prc_PR ~ Comp_PR ,
data=lease_data_PR,
main="Companies Rent Prices",
horizontal=TRUE, col=3,pch=20)
```



I could not use T-Test because this metric violates 2/3 normality assumptions. For that reason I used Wilcoxon test. The Wilcoxon test result was p-value > 0.05, indicating that there is not significant evidence to reject the hypothesis that rental prices are the same btw the two companies.

4. Training and Test Set

Spliting the dataframe into a training and a test the rate of data for my train and test set is 65/35 My speed is –> 3215

```r
# Number of rows of data
n.row <- nrow(lease_data_PR)
# Choose sampling rate
set.seed(3215)
sr_pr <- 0.65
#Choose the rows for the training sample with my student id
training.rows <- sample(1:n.row, sr_pr*n.row, replace=FALSE)
#Assign to the training sample
```

```r
train_pr <- subset(lease_data_PR[training.rows,])
# Assign the balance to the Test Sample (rest of data)
test_pr <- subset(lease_data_PR[-c(training.rows),])
```

Comparisson train and test dataset Some sumarizations

```r
#summaries
summary(train_pr)
```

```
##      Prc_PR         Bed_PR         floor_PR       TotFloor_PR      Bath_PR
##  Min.   : 255   Min.   :1.000   Min.   : 1.000   Min.   : 1.0   Min.   :1.000
##  1st Qu.:1690   1st Qu.:1.000   1st Qu.: 1.000   1st Qu.: 2.0   1st Qu.:1.000
##  Median :2180   Median :2.000   Median : 2.000   Median : 4.0   Median :1.000
##  Mean   :2311   Mean   :2.089   Mean   : 2.258   Mean   : 3.9   Mean   :1.653
##  3rd Qu.:2820   3rd Qu.:3.000   3rd Qu.: 3.000   3rd Qu.: 5.0   3rd Qu.:2.000
##  Max.   :5810   Max.   :4.000   Max.   :10.000   Max.   :10.0   Max.   :4.000
##     Sqft_PR             City_PR         Comp_PR       Dist_PR
##  Min.   : 505   Blossomville:220   Leaseflow:219   Min.   : 0.000
##  1st Qu.: 779   Riverport   :240   Rentopia :458   1st Qu.: 1.900
##  Median :1233   Terranova   :217                   Median : 3.600
##  Mean   :1275                                      Mean   : 4.106
##  3rd Qu.:1628                                      3rd Qu.: 5.300
##  Max.   :2943                                      Max.   :16.700
```

```r
summary(test_pr)
```

```
##      Prc_PR         Bed_PR         floor_PR       TotFloor_PR
##  Min.   : 373   Min.   :1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:1572   1st Qu.:1.000   1st Qu.: 1.000   1st Qu.: 2.000
##  Median :2070   Median :2.000   Median : 2.000   Median : 4.000
##  Mean   :2188   Mean   :2.191   Mean   : 2.301   Mean   : 3.899
##  3rd Qu.:2710   3rd Qu.:3.000   3rd Qu.: 3.000   3rd Qu.: 5.000
##  Max.   :5750   Max.   :4.000   Max.   :10.000   Max.   :10.000
##     Bath_PR         Sqft_PR               City_PR         Comp_PR
##  Min.   :1.000   Min.   : 501.0   Blossomville:142   Leaseflow:142
##  1st Qu.:1.000   1st Qu.: 826.8   Riverport   :113   Rentopia :224
##  Median :1.000   Median :1224.0   Terranova   :111
##  Mean   :1.566   Mean   :1307.3
##  3rd Qu.:2.000   3rd Qu.:1684.5
##  Max.   :4.000   Max.   :2916.0
##     Dist_PR
##  Min.   : 0.200
##  1st Qu.: 1.900
##  Median : 3.200
##  Mean   : 4.034
##  3rd Qu.: 5.500
##  Max.   :21.000
```

```r
#mean each set
round(mean(train_pr$Prc_PR),6)
```

```
## [1] 2310.582
```

```r
round(mean(test_pr$Prc_PR),6)
```

```
## [1] 2187.798
```

```r
#commparing median with wilcox test
wilcox.test(train_pr$Prc_PR, test_pr$Prc_PR)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  train_pr$Prc_PR and test_pr$Prc_PR
## W = 132228, p-value = 0.07259
## alternative hypothesis: true location shift is not equal to 0
```

In the summaries, I did not evidence any dissimilarities. The means show that there are not significant differences between sets.

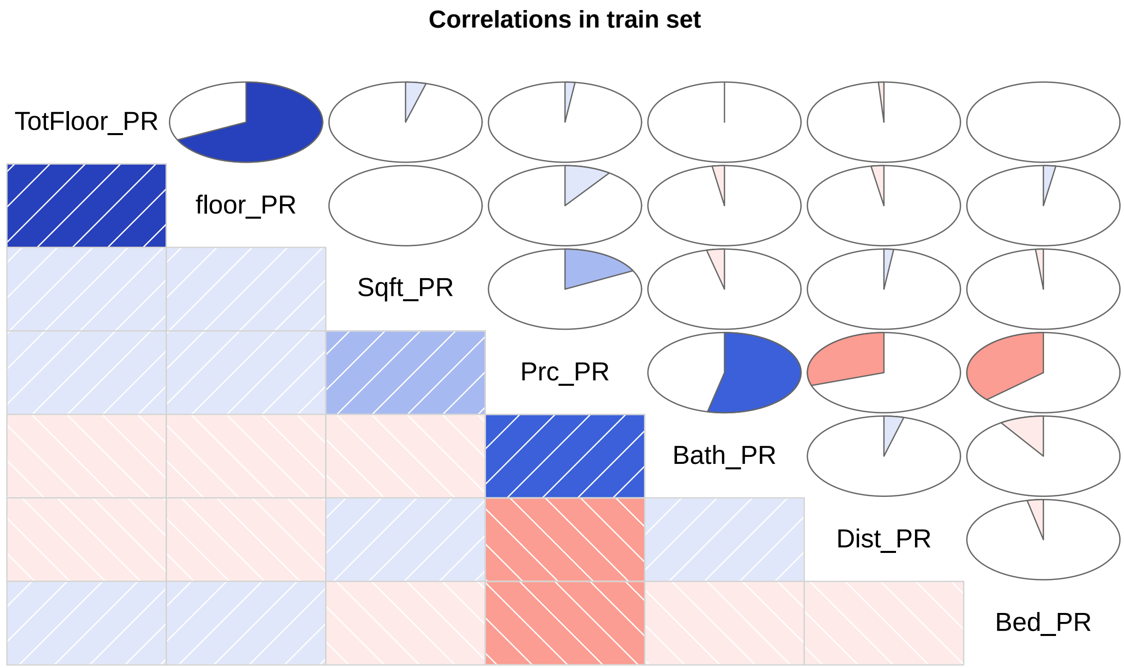
In addition, the result of wilcoxon test (p-value = 0.07) indicates that the medians are the same. Based on these findings, it is appropriate to proceed with model creation.

2. Simple Linear Regression

1. Correlations

Graphical and numerical correlations

```r
# Correlation plot
corrgram(train_pr, order=TRUE, lower.panel=panel.shade,
         upper.panel=panel.pie, text.panel=panel.txt,
         main="Correlations in train set")
```




```r
#Numerical correlations
train_cor_pr <- cor(train_pr[sapply(train_pr, is.numeric)], method="spearman")
round(train_cor_pr, 2)
```

```
##              Prc_PR Bed_PR floor_PR TotFloor_PR Bath_PR Sqft_PR Dist_PR
## Prc_PR         1.00  -0.38     0.15        0.08    0.44    0.17   -0.29
## Bed_PR        -0.38   1.00     0.02       -0.01   -0.07   -0.01   -0.05
## floor_PR       0.15   0.02     1.00        0.65   -0.02   -0.01   -0.03
## TotFloor_PR    0.08  -0.01     0.65        1.00   -0.01    0.05   -0.03
## Bath_PR        0.44  -0.07    -0.02       -0.01    1.00   -0.02    0.04
## Sqft_PR        0.17  -0.01    -0.01        0.05   -0.02    1.00    0.01
## Dist_PR       -0.29  -0.05    -0.03       -0.03    0.04    0.01    1.00
```

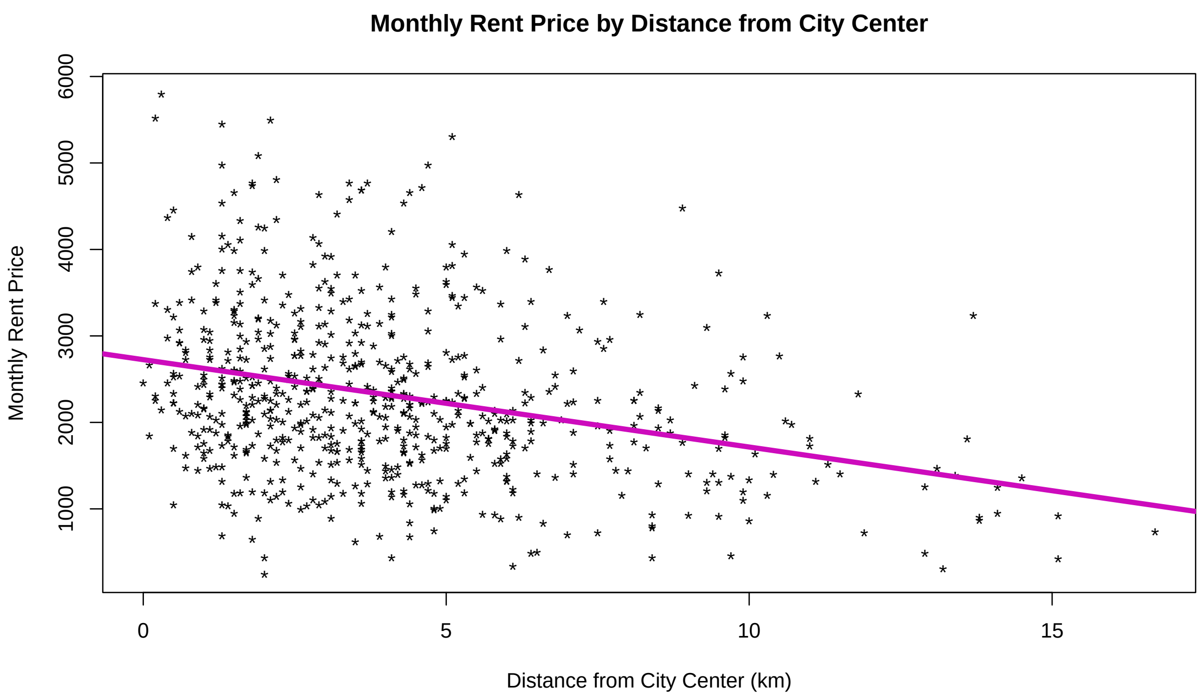
Findings

- *TotFloor_PR and floor_PR* 65% of correlation. There is an obvious positive correlation between both variables. Which indicates that buildings with more floors have apartments located in higher floors.
- *Prc_PR and Bath_PR* 45% of correlation. Indicates that apartments with more bathrooms tend to have a higher monthly rent
- *Bed_PR and Prc_PR* -38% of surprising correlation. There is a negative correlations, which means that apartments with less bedrooms curiously are more expensive than apartments with more bedrooms.
- *Prc_PR and Dist_PR* -29% of correlation. This negative correlations indicates that if there less distance is between apartments and center of town more expensive apartment is.
- *Prc_PR and Sqft_PR* 17% correlation. There is a positive correlation between price and apartment size. This indicates that bigger apartments tend to have higher rent price. Which is expected, however the correlation is weak.

2. Simple linear regression model Prc_PR ~ Dist_PR

Using rental price the dependent variable and distance from town centre as the independent variable

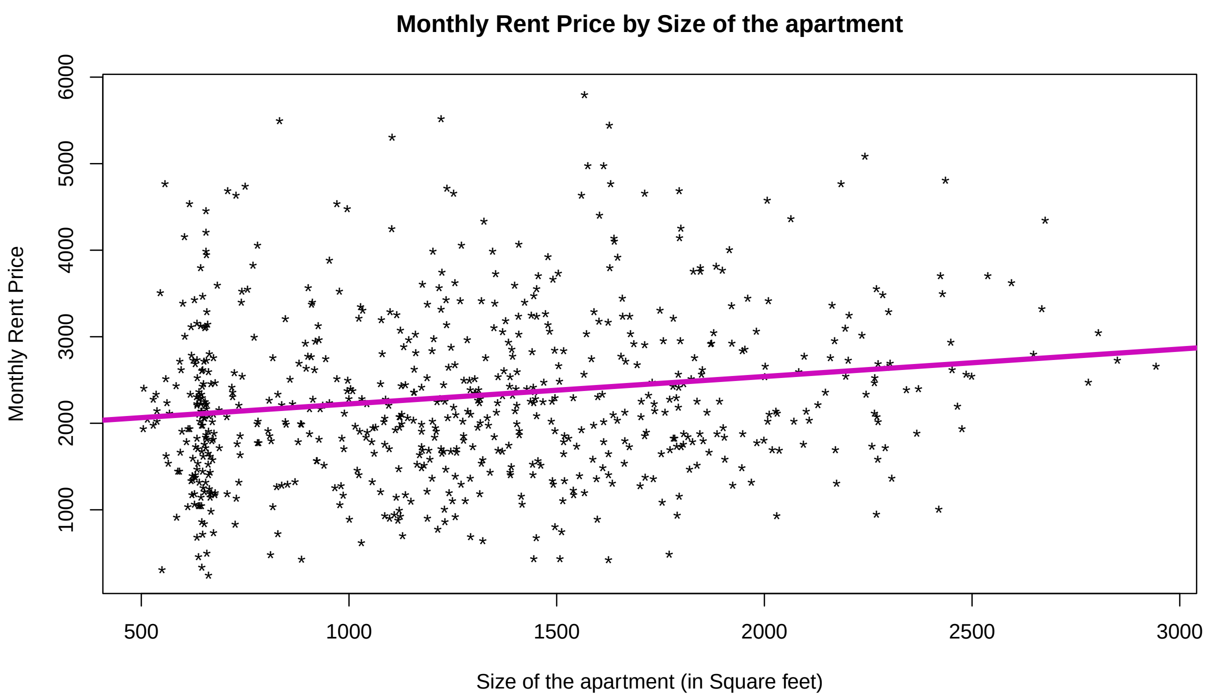
```
# Creating linear model
mod.Dist_PR <- lm(Prc_PR ~ Dist_PR, data = train_pr)

# Creating plot with regression line
plot(Prc_PR ~ Dist_PR, data = train_pr, pch = 42
     , main = "Monthly Rent Price by Distance from City Center"
     ,xlab = "Distance from City Center (km)",
     ylab = "Monthly Rent Price")
abline(mod.Dist_PR, col = 14, lwd = 4)
```



3. Simple linear regression model Prc_PR ~ Sqft_PR

Using rental price the dependent variable and size of the apartment as the independent variable

```
# Creating linear model
mod.Sqft_PR <- lm(Prc_PR ~ Sqft_PR, data = train_pr)

# Creating plot with regression line
plot(Prc_PR ~ Sqft_PR, data = train_pr, pch = 42, main = "Monthly Rent Price by Size of the apartment"
     ,xlab = "Size of the apartment (in Square feet)",
     ylab = "Monthly Rent Price")
abline(mod.Sqft_PR, col = 14, lwd = 4)
```



4. Comparing the models mod.Dist_PR and mod.Sqft_PR

To select the best model, it is necessary to compare the following summaries:

```
# Comparing summaries Dist_PR
summary(mod.Dist_PR)
```

```
##
## Call:
## lm(formula = Prc_PR ~ Dist_PR, data = train_pr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2268.4  -593.8  -114.4   450.2  3114.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2725.42      61.53  44.292  < 2e-16 ***
## Dist_PR      -101.02      12.28  -8.228 9.81e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 917.8 on 675 degrees of freedom
## Multiple R-squared:  0.09115,    Adjusted R-squared:  0.08981
## F-statistic:  67.7 on 1 and 675 DF,  p-value: 9.808e-16
```

```r
pred.Dist_PR <- predict(mod.Dist_PR, newdata=train_pr)
RMSE_trn_Dist_PR <- sqrt(mean((train_pr$Prc_PR - pred.Dist_PR)^2))
RMSE_trn_Dist_PR
```

```
## [1] 916.456
```

```r
# Model in test set Dist_PR
pred.Dist_tst_PR <- predict(mod.Dist_PR, newdata = test_pr)
RMSE_tst_Dist_PR <- sqrt(mean((test_pr$Prc_PR - pred.Dist_tst_PR)^2))
RMSE_tst_Dist_PR
```

```
## [1] 873.1584
```

```r
# Comparing summaries Sqft_PR
summary(mod.Sqft_PR)
```

```
##
## Call:
## lm(formula = Prc_PR ~ Sqft_PR, data = train_pr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1984.6  -637.3  -112.1   508.8  3406.8
##
## Coefficients:
##               Estimate Std. Error t value   Pr(>|t|)
## (Intercept) 1906.29376   95.28154  20.007    < 2e-16 ***
## Sqft_PR        0.31713    0.06906   4.592 0.00000523 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 948 on 675 degrees of freedom
## Multiple R-squared:  0.03029,    Adjusted R-squared:  0.02886
## F-statistic: 21.09 on 1 and 675 DF,  p-value: 0.000005234
```

```r
pred.Sqft_PR <- predict(mod.Sqft_PR, newdata=train_pr)
RMSE_trn_Sqft_PR <- sqrt(mean((train_pr$Prc_PR - pred.Sqft_PR)^2))
RMSE_trn_Sqft_PR
```

```
## [1] 946.6432
```

```r
# Model in test set Sqft_PR
pred.Sqft_tst_PR <- predict(mod.Sqft_PR, newdata = test_pr)
RMSE_tst_Dist_PR <- sqrt(mean((test_pr$Prc_PR - pred.Sqft_tst_PR)^2))
RMSE_tst_Dist_PR
```

```
## [1] 902.6996
```

Findings

- **Analysis————-mod.Dist_PR—————-mod.Sqft_PR**

F-Stat——— p-value(9.808e-16) - PASS—— p-value (0.000005234) - PASS

R2 Adj———- 8.9 (could be better)——— 2.8 (could be better)

Residual————No symmetric—————-No symmetric

t-test———- p-value < 0.05 - PASS———-p-value < 0.05 - PASS

Coefficients—- consistency (negative)———consistency (positive)

RMSE train————916.456——————-946.6432

RMSE test————-873.1584——————902.6996

Conclusions

- Both models have p-values for the f-stat significantly low. Which means that those variables are useful to predict rental price. Even though R2 Adj. is relatively low, the variability explained by Dist_PR is better with 8.9.
- The residuals in both models are no symmetrical, the minimum and maximum are so separated and the median is not close to 0.
- Related to coefficients, the t-test suggest that both variables got a p-value less than 0.05 and the coefficients are consistent with the correlations previously shown.
- The RMSE results of mod.Dist_PR (916.456) in comparison with the mod.Sqft_PR (946.6432) is better. In both models, the training set and test set are relatively closed (916.456 vs. 873.1584 for mod.Dist_PR, and 946.6432 vs. 902.6996 for mod.Sqft_PR). This suggest that models are not over-fitting neither under-fitting.

I think that the best model based on the coefficients and R2 adjusted is mod.Dist_PR. However, this model could explain what happened but I would not use those models to predict new observations based on the R2 adj. I could add more variables to find a best model.

3. Model Development – Multivariate

Model Using All the variables

```r
# Creating full model
mod.Full_PR <- lm(Prc_PR ~ . , data = train_pr, na.action=na.omit)

# Summaries Model
summary(mod.Full_PR)
```

```
##
## Call:
## lm(formula = Prc_PR ~ ., data = train_pr, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1908.75  -417.17    22.27   434.88  2021.73
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1950.21465  113.14563  17.236  < 2e-16 ***
## Bed_PR            -287.82178   20.45320 -14.072  < 2e-16 ***
## floor_PR            95.67592   19.96165   4.793 2.03e-06 ***
## TotFloor_PR        -44.00865   15.44225  -2.850  0.00451 **
## Bath_PR            505.27175   23.99709  21.056  < 2e-16 ***
## Sqft_PR              0.36438    0.04502   8.093 2.76e-15 ***
## City_PRRiverport   337.78900   58.06449   5.817 9.27e-09 ***
## City_PRTerranova  -104.34949   59.30342  -1.760  0.07894 .
## Comp_PRRentopia     -1.74721   50.69561  -0.034  0.97252
## Dist_PR           -113.90402    8.25436 -13.799  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 615 on 667 degrees of freedom
## Multiple R-squared:  0.5968, Adjusted R-squared:  0.5914
## F-statistic: 109.7 on 9 and 667 DF,  p-value: < 2.2e-16
```

```r
#Calculing RMSE in train set
pred.Full_PR <- predict(mod.Full_PR, newdata=train_pr)
RMSE_trn_Full_PR <- sqrt(mean((train_pr$Prc_PR - pred.Full_PR)^2))
RMSE_trn_Full_PR
```

```
## [1] 610.4129
```

```r
#Calculing RMSE in train set
pred.Full_tst_PR <- predict(mod.Full_PR, newdata = test_pr)
RMSE_tst_Full_PR <- sqrt(mean((test_pr$Prc_PR - pred.Full_tst_PR)^2))
RMSE_tst_Full_PR
```

```
## [1] 636.2402
```

Findings Full Model

- **Analysis**————————— **mod.Full_PR**
  F-Stat———————— p-value(2.2e-16) - PASS
  R2 Adj————————— 59.1 (works)
  Residual———— It's not perfect but it's better
  t-test———————— p-value $<$ 0.05 - 8/10 PASS
  Coefficients————— Match with correlations
  RMSE train——————— 610.4129
  RMSE test———————— 636.2631

Conclusions

- Both models have p-values for the f-stat significantly low.
- The R2 Adj. result is much better because explain about 59% of variability in data. Which means that these variables are useful for predicting rental price.

- The residuals in this model are a little more symmetrical, the minimum and maximum are quite far apart but it looks more symmetrical.
- Regarding the coefficients, mostly all of variables (8/10) have p-value below 0.05, and the coefficients are align with the correlation matrix.
- The training RMSE and the test RMSE are similar Suggesting the model generalizes well and is neither overfitting nor underfitting. Additionally is much better that previous models.

Based on the coefficients, the f-stat, the R2 adjusted, and the RMSE, this model seems to be more effective in predicting rental price that previos models.

Model Using Backward

```r
# Creating backward model
mod.Back_PR <- step(mod.Full_PR, direction="backward", details=TRUE)
```

```
## Start:  AIC=8704.74
## Prc_PR ~ Bed_PR + floor_PR + TotFloor_PR + Bath_PR + Sqft_PR +
##     City_PR + Comp_PR + Dist_PR
##
##                Df Sum of Sq       RSS    AIC
## - Comp_PR       1       449 252253291 8702.7
## <none>                      252252842 8704.7
## - TotFloor_PR   1   3071604 255324446 8710.9
## - floor_PR      1   8688053 260940895 8725.7
## - City_PR       2  24150793 276403635 8762.6
## - Sqft_PR       1  24769890 277022732 8766.2
## - Dist_PR       1  72014845 324267687 8872.8
## - Bed_PR        1  74891921 327144764 8878.7
## - Bath_PR       1 167665225 419918067 9047.8
##
## Step:  AIC=8702.74
## Prc_PR ~ Bed_PR + floor_PR + TotFloor_PR + Bath_PR + Sqft_PR +
##     City_PR + Dist_PR
##
##                Df Sum of Sq       RSS    AIC
## <none>                      252253291 8702.7
## - TotFloor_PR   1   3085025 255338316 8709.0
## - floor_PR      1   8734842 260988133 8723.8
## - City_PR       2  24171451 276424742 8760.7
## - Sqft_PR       1  24770711 277024002 8764.2
## - Dist_PR       1  72057081 324310373 8870.8
## - Bed_PR        1  74893077 327146369 8876.7
## - Bath_PR       1 167702925 419956217 9045.8
```

```r
# Summaries Model
summary(mod.Back_PR)
```

```
##
## Call:
## lm(formula = Prc_PR ~ Bed_PR + floor_PR + TotFloor_PR + Bath_PR +
```

```
##     Sqft_PR + City_PR + Dist_PR, data = train_pr, na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1909.3  -417.7    21.7   436.0  2023.0
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1949.09804  108.32677  17.993  < 2e-16 ***
## Bed_PR           -287.82332   20.43786 -14.083  < 2e-16 ***
## floor_PR           95.72159   19.90273   4.809 1.87e-06 ***
## TotFloor_PR       -44.03792   15.40734  -2.858  0.00439 **
## Bath_PR           505.25792   23.97579  21.074  < 2e-16 ***
## Sqft_PR             0.36436    0.04499   8.099 2.63e-15 ***
## City_PRRiverport  337.70983   57.97563   5.825 8.88e-09 ***
## City_PRTerranova -104.37765   59.25345  -1.762  0.07860 .
## Dist_PR          -113.89639    8.24521 -13.814  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 614.5 on 668 degrees of freedom
## Multiple R-squared:  0.5968, Adjusted R-squared:  0.592
## F-statistic: 123.6 on 8 and 668 DF,  p-value: < 2.2e-16
```

```
# RMSE in train
pred.Back_PR <- predict(mod.Back_PR, newdata=train_pr)
RMSE_trn_Back_PR <- sqrt(mean((train_pr$Prc_PR - pred.Back_PR)^2))
RMSE_trn_Back_PR
```

```
## [1] 610.4134
```

```
# RMSE in test
pred.Back_tst_PR <- predict(mod.Back_PR, newdata = test_pr)
RMSE_tst_Back_PR <- sqrt(mean((test_pr$Prc_PR - pred.Back_tst_PR)^2))
RMSE_tst_Back_PR
```

```
## [1] 636.2359
```

Findings Bck

- **Analysis**————————— **mod.Back_PR**
  F-Stat————————— p-value(2.2e-16) - PASS
  R2 Adj————————— 59.2 (the best at this point)
  Residual—————————- It's not perfect but it's better
  t-test————————— p-value $<$ 0.05 - 8/10 PASS
  Coefficients————————— Match with correlations
  RMSE train—————————- 610.4134
  RMSE test————————— 636.2359

Conclusions

- The process started with all variables included, but in the second step the variable Comp was removed.

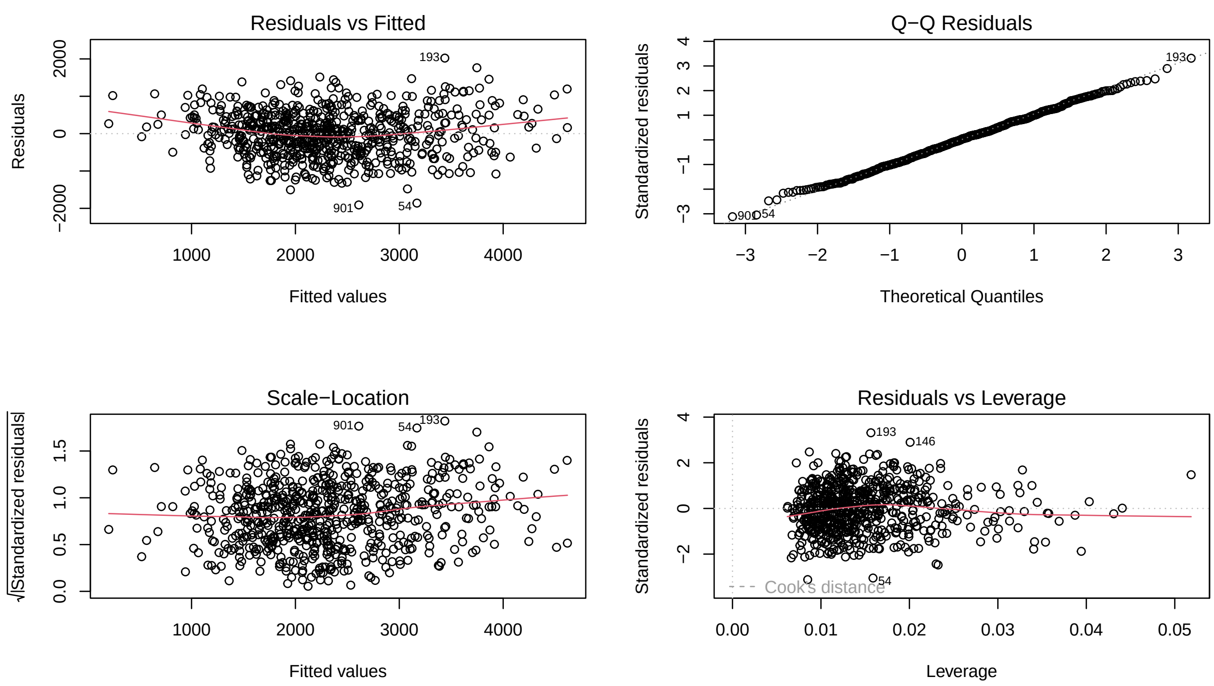
- Only one variable was removed, compared with the full model. For this reason, the final model is similar to the full model, with minimal differences.

It can possible notice a slight improvement in the R2 adj. which increased from 59.1 in the full model to 59.2 in the final model.

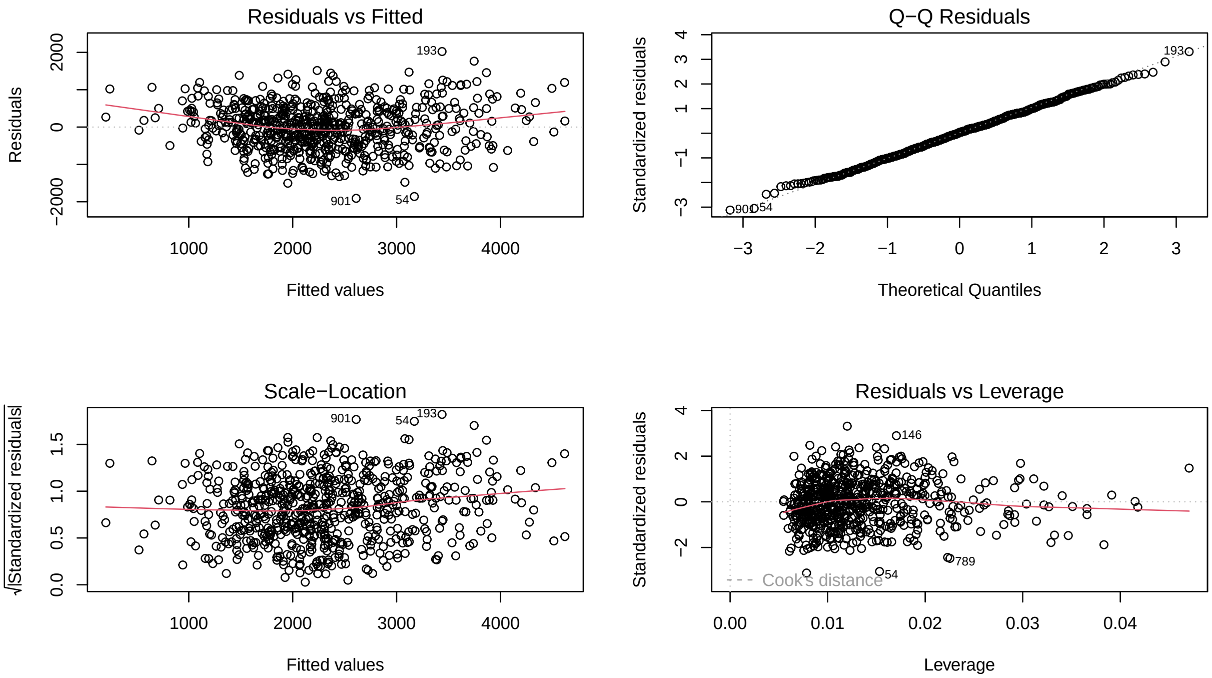
4. Model Evaluation – Verifying Assumptions – Multivariate

Plot Residuals

```r
# Evaluating the Models  - residuals
# Model 1
par(mfrow = c(2, 2))
plot(mod.Full_PR)
```




```r
# Model 2
par(mfrow = c(2, 2))
plot(mod.Back_PR)
```



Shapirto test

```r
#creating vectors the residual for each model,

full.res_pr <- residuals(mod.Full_PR)
back.res_pr <- residuals(mod.Back_PR)


# Validating if residuals are normal in full model
shapiro.test(full.res_pr)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  full.res_pr
## W = 0.99856, p-value = 0.8629
```

```r
# Validating if residuals are normal in back model
shapiro.test(back.res_pr)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  back.res_pr
## W = 0.99856, p-value = 0.8624
```

Analyzing the errors

- **Linearity** - Both models meets this assumption. The relationship between response variables and predictor variables are linear. There are not patterns in there
- **Independence of predictors** - Both models meets this assumption. Observations are independent of each other, no linear relationship.
- **Distribution of Error Terms** - Both models meets this assumption. The QQ plot are very similar, indicating that errors are normaly distributed, The Shapiro test got results above to 0.8 indicating that both model are normal.
- **The residuals are homoscedastic** - Both models meet assumption. In the models the variance of the errors is constant, which means both models are stables.

In the residuals vs Leverage charts, we can see an observation which has a high leverage and an influential point. However, points did not fall in the Cook's distance, meaning that we don't have significant influences.

5. Final Recommendation – Multivariate

Compare all RMSE

```r
#RMSE FULL
RMSE_full_PR <- c(RMSE_trn_Full_PR,RMSE_tst_Full_PR)
round(RMSE_full_PR,2)
```

```
## [1] 610.41 636.24
```

```r
#RMSE BCK
RMSE_back_PR <- c(RMSE_trn_Back_PR,RMSE_tst_Back_PR)
round(RMSE_back_PR,2)
```

```
## [1] 610.41 636.24
```

```r
#Mean residuals
mean(full.res_pr)
```

```
## [1] -4.112919e-15
```

```r
mean(back.res_pr)
```

```
## [1] -3.34164e-14
```

- Based on the results, the RMSE in both models (full and backward) is almost the same, providing more precision in predicting rental prices. Considering rental price range (min: 255, median: 2180, and max: 5810), an RMSE of ~610 in the training, is not bad in relation to the median rental price.
- The RMSE results for both training and test, are similar, indicating that there is neither overfitting nor underfitting.
- The R2 in both model is also similar, which means the models can explain 59% of the variability.
- Both models meet the residuals' assumptions.
- The mean of the residuals are close to 0 (full: -4.112919e-15; back: -3.34164e-14)

The biggest difference between the full model and the backward model is the number of variables, as the backward model eliminated Comp_PR. Due to its simplicity, I recommend the backward model for predicting rental prices

References

Ngo, L. (2023, January 10). The Ultimate Guide to Logical Operators in R. Built In. https://builtin.com/data-science/and-in-r Conestoga College. (2024). PROG8435 – Data Analysis, Modeling and Algorithms - LECTURE 8 – REGRESSION ANALYSIS [PowerPoint slides]. eConestoga.