Assignment 2 - Data Transformation, Statistical Inference and Comparisons

Shubham Maheshwari

2024-10-05

#if(!is.null(dev.list())) dev.off()  
#cat("\014")   
#rm(list=ls())  
options(scipen=9)

if(!require(tinytex)){install.packages("tinytex")}

## Loading required package: tinytex

library("tinytex")  
  
if(!require(pastecs)){install.packages("pastecs")}

## Loading required package: pastecs

library("pastecs")  
  
if(!require(lattice)){install.packages("lattice")}

## Loading required package: lattice

library("lattice")

# 1. Data Transformation and Preperation

## 1. Initial Transformation

### a. Rename all variables with your initials appended

getwd()

## [1] "C:/PROG8435 Data Analytics/Assignment 2"

df <- read.table("PROG8435-24F-Assign02.txt",   
 sep = ",", header = TRUE)  
head(df)

## Index Room Ren DT TM S1\_L S2\_L S3\_L S1\_T S2\_T S3\_T FN  
## 1 1 111 New 47 21 1 1 1 24.479 21.651 22.227 1106  
## 2 2 111 New 3 2 1 1 1 23.771 24.167 22.104 1146  
## 3 3 311 Old 27 10 1 1 1 24.760 21.143 22.881 1109  
## 4 4 211 New 5 22 1 1 1 22.130 23.925 19.353 1111  
## 5 5 311 Old 28 18 1 1 1 24.425 24.832 24.287 1120  
## 6 6 211 New 9 18 1 1 1 22.158 20.393 26.612 1078

colnames(df) <- paste(colnames(df),"SM",sep = "\_")  
head(df)

## Index\_SM Room\_SM Ren\_SM DT\_SM TM\_SM S1\_L\_SM S2\_L\_SM S3\_L\_SM S1\_T\_SM S2\_T\_SM  
## 1 1 111 New 47 21 1 1 1 24.479 21.651  
## 2 2 111 New 3 2 1 1 1 23.771 24.167  
## 3 3 311 Old 27 10 1 1 1 24.760 21.143  
## 4 4 211 New 5 22 1 1 1 22.130 23.925  
## 5 5 311 Old 28 18 1 1 1 24.425 24.832  
## 6 6 211 New 9 18 1 1 1 22.158 20.393  
## S3\_T\_SM FN\_SM  
## 1 22.227 1106  
## 2 22.104 1146  
## 3 22.881 1109  
## 4 19.353 1111  
## 5 24.287 1120  
## 6 26.612 1078

str(df)

## 'data.frame': 4323 obs. of 12 variables:  
## $ Index\_SM: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Room\_SM : int 111 111 311 211 311 211 211 211 211 211 ...  
## $ Ren\_SM : chr "New" "New" "Old" "New" ...  
## $ DT\_SM : int 47 3 27 5 28 9 58 28 18 57 ...  
## $ TM\_SM : int 21 2 10 22 18 18 14 13 13 16 ...  
## $ S1\_L\_SM : int 1 1 1 1 1 1 0 0 1 1 ...  
## $ S2\_L\_SM : int 1 1 1 1 1 1 0 0 1 1 ...  
## $ S3\_L\_SM : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ S1\_T\_SM : num 24.5 23.8 24.8 22.1 24.4 ...  
## $ S2\_T\_SM : num 21.7 24.2 21.1 23.9 24.8 ...  
## $ S3\_T\_SM : num 22.2 22.1 22.9 19.4 24.3 ...  
## $ FN\_SM : int 1106 1146 1109 1111 1120 1078 1079 1112 1116 1172 ...

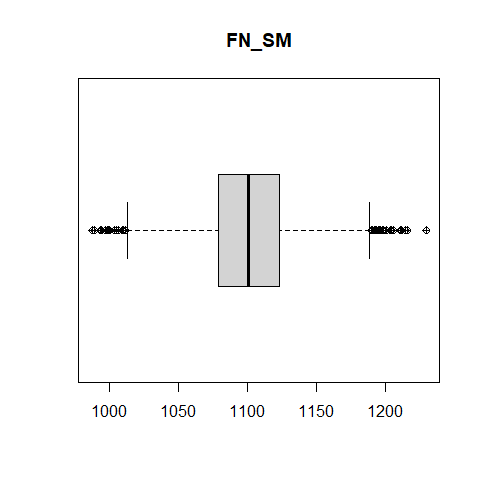
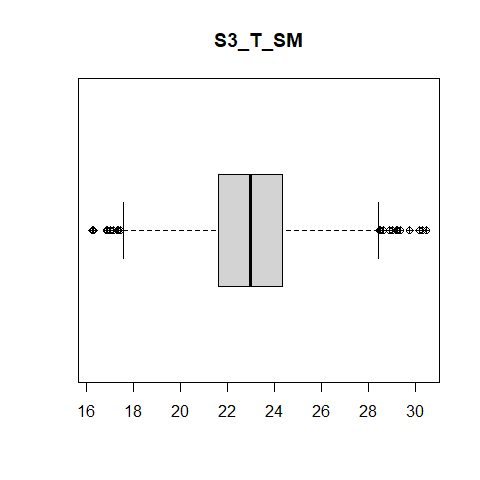
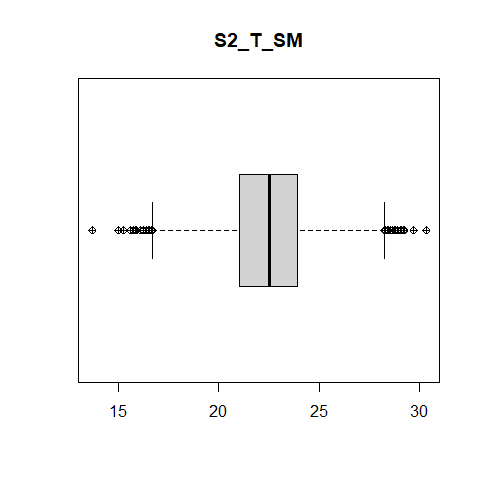
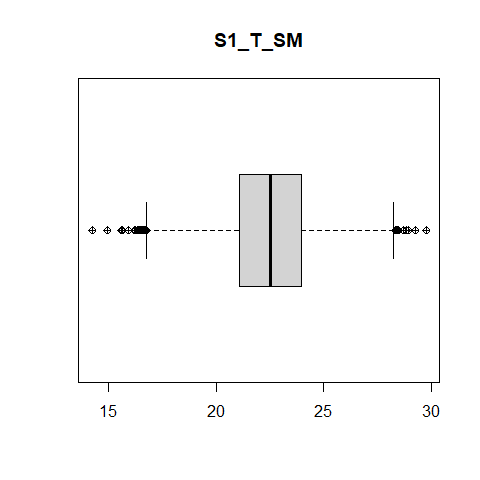
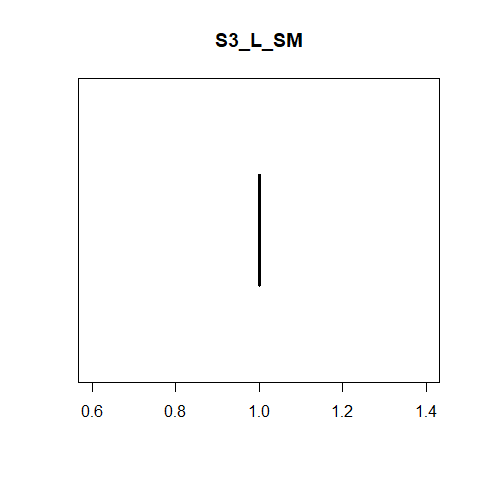
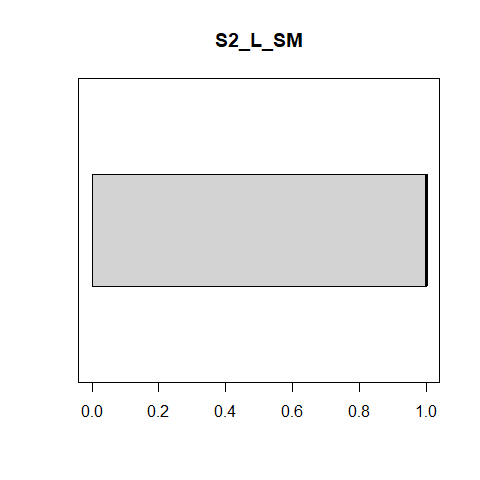
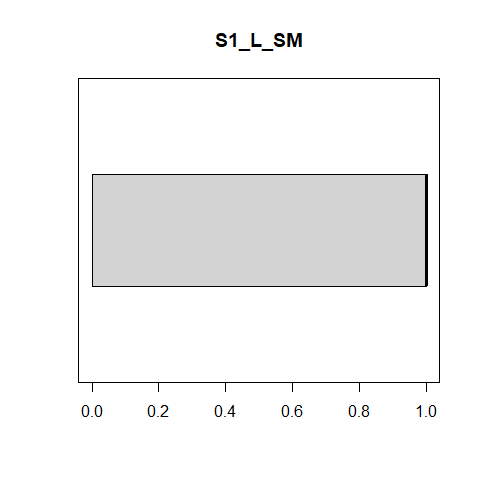
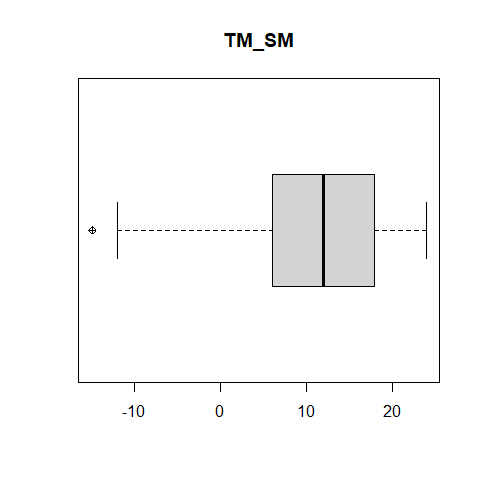
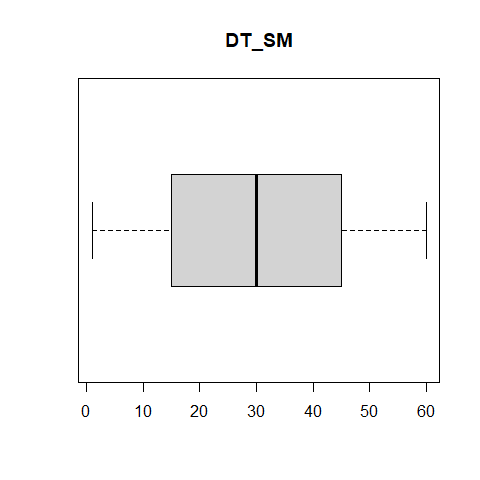
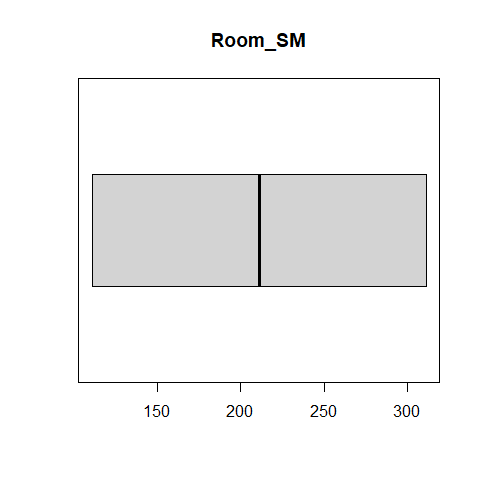
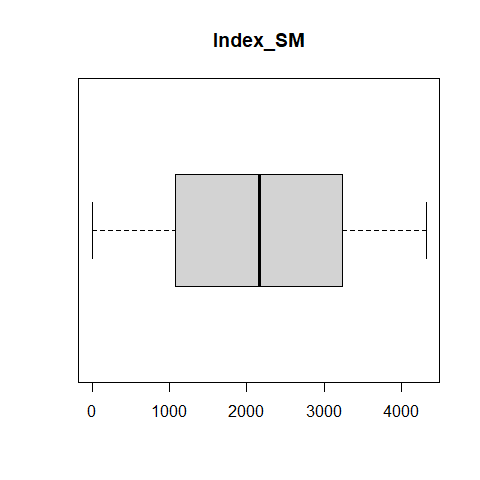
### b. Transform character variables to factor variables.

df$Ren\_SM<- as.factor(df$Ren\_SM)  
str(df)

## 'data.frame': 4323 obs. of 12 variables:  
## $ Index\_SM: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Room\_SM : int 111 111 311 211 311 211 211 211 211 211 ...  
## $ Ren\_SM : Factor w/ 2 levels "New","Old": 1 1 2 1 2 1 1 1 1 1 ...  
## $ DT\_SM : int 47 3 27 5 28 9 58 28 18 57 ...  
## $ TM\_SM : int 21 2 10 22 18 18 14 13 13 16 ...  
## $ S1\_L\_SM : int 1 1 1 1 1 1 0 0 1 1 ...  
## $ S2\_L\_SM : int 1 1 1 1 1 1 0 0 1 1 ...  
## $ S3\_L\_SM : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ S1\_T\_SM : num 24.5 23.8 24.8 22.1 24.4 ...  
## $ S2\_T\_SM : num 21.7 24.2 21.1 23.9 24.8 ...  
## $ S3\_T\_SM : num 22.2 22.1 22.9 19.4 24.3 ...  
## $ FN\_SM : int 1106 1146 1109 1111 1120 1078 1079 1112 1116 1172 ...

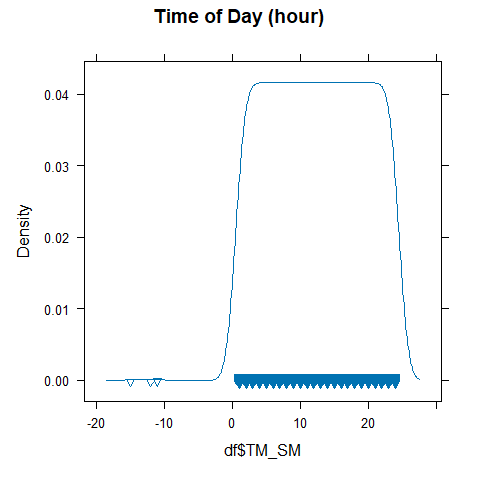
## 2. Outliers

# We will be checking for any outliers using the boxplot technique and further investigate into it accordingly   
  
# Running a for loop to generate box plots for numerical values only.   
for (i in 1:ncol(df))  
{  
 if(is.numeric(df[,i]))  
 {  
 boxplot(df[i], main = names(df)[i],  
 horizontal=TRUE, pch=10)  
 }  
}

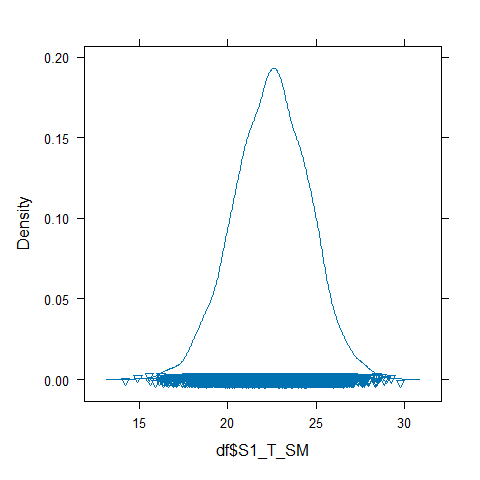
 Observations from the box plots:

1. DT: The date ranges from 0 - 60 and detects no anomolity in it
2. TM ( Time of Day): It shows values that are negative which are anamolous since time cannot be negative, will further look into it using the density plot.
3. Temperatues from sensor 1 looks to be in the normal range which can also be seen by the other sensors as well.
4. No anamolities detected in the fan speeds either.

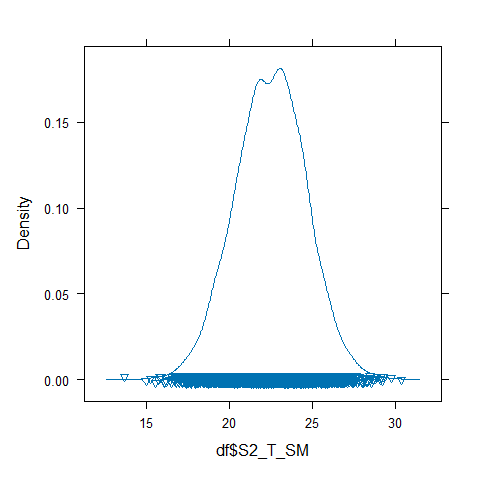
# Using density plot to check for values that are below 0   
  
densityplot(df$TM\_SM, main = "Time of Day (hour)", pch=6)



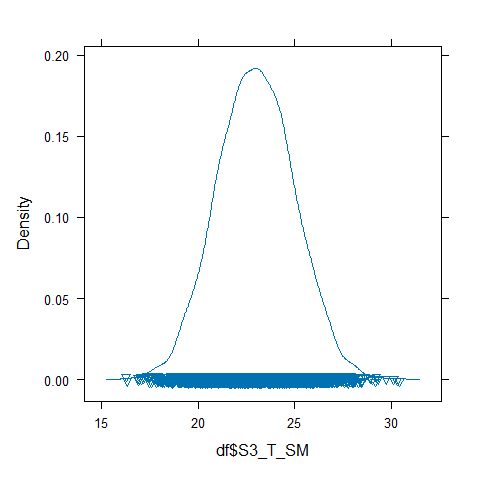
# Checking other variables for fun   
densityplot(df$S1\_T\_SM, pch=6)



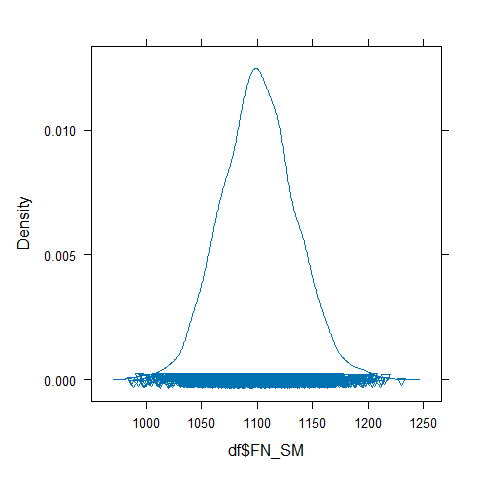
densityplot(df$S2\_T\_SM, pch=6)



densityplot(df$S3\_T\_SM, pch=6)

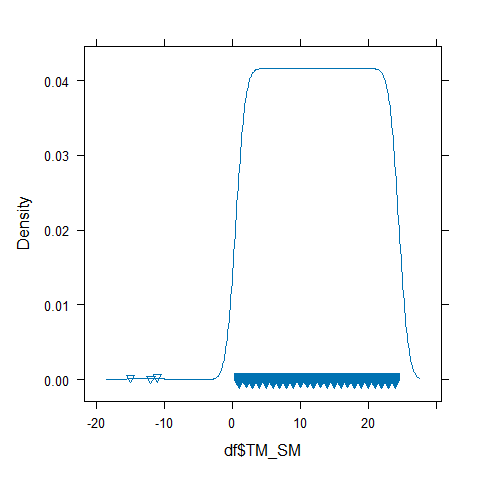


densityplot(df$FN\_SM, pch=6)

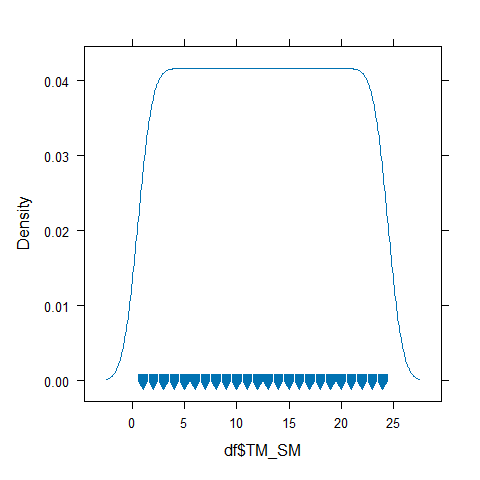
 Observation:

It confirms that there are 3 negative values, I will be proceeding to delete them as there can’t be no negative values for hours.

densityplot( ~ df$TM\_SM, pch=6)



nr <- which(df$TM\_SM < 0) #Finds values less than 0   
df <- df[-c(nr),]  
densityplot( ~ df$TM\_SM, pch=6)



## 3. Reduce Dimensionality

### a. Drop any variables that do not contribute any useful analytical information at all.

# Index column doesn't provide us with any usefule analytical information at all so we will be removing it.   
  
df <- df[-c(1)]

### b. Apply the Missing Value Filter to remove appropriate columns of data.

summary(df)

## Room\_SM Ren\_SM DT\_SM TM\_SM S1\_L\_SM   
## Min. :111 New:2880 Min. : 1.00 Min. : 1.00 Min. :0.0000   
## 1st Qu.:111 Old:1440 1st Qu.:15.75 1st Qu.: 6.75 1st Qu.:0.0000   
## Median :211 Median :30.50 Median :12.50 Median :1.0000   
## Mean :211 Mean :30.50 Mean :12.50 Mean :0.5035   
## 3rd Qu.:311 3rd Qu.:45.25 3rd Qu.:18.25 3rd Qu.:1.0000   
## Max. :311 Max. :60.00 Max. :24.00 Max. :1.0000   
## S2\_L\_SM S3\_L\_SM S1\_T\_SM S2\_T\_SM S3\_T\_SM   
## Min. :0.0000 Min. :1 Min. :14.22 Min. :13.67 Min. :16.26   
## 1st Qu.:0.0000 1st Qu.:1 1st Qu.:21.07 1st Qu.:21.03 1st Qu.:21.62   
## Median :1.0000 Median :1 Median :22.51 Median :22.53 Median :22.98   
## Mean :0.5035 Mean :1 Mean :22.48 Mean :22.47 Mean :22.99   
## 3rd Qu.:1.0000 3rd Qu.:1 3rd Qu.:23.95 3rd Qu.:23.92 3rd Qu.:24.35   
## Max. :1.0000 Max. :1 Max. :29.78 Max. :30.34 Max. :30.45   
## FN\_SM   
## Min. : 987   
## 1st Qu.:1079   
## Median :1101   
## Mean :1101   
## 3rd Qu.:1123   
## Max. :1230

There are no missing values present in the data.

### c. Apply the Low Variance Filter to remove appropriate columns of data

stat.desc(df)

## Room\_SM Ren\_SM DT\_SM TM\_SM S1\_L\_SM  
## nbr.val 4320.000000 NA 4320.0000000 4320.0000000 4320.000000000  
## nbr.null 0.000000 NA 0.0000000 0.0000000 2145.000000000  
## nbr.na 0.000000 NA 0.0000000 0.0000000 0.000000000  
## min 111.000000 NA 1.0000000 1.0000000 0.000000000  
## max 311.000000 NA 60.0000000 24.0000000 1.000000000  
## range 200.000000 NA 59.0000000 23.0000000 1.000000000  
## sum 911520.000000 NA 131760.0000000 54000.0000000 2175.000000000  
## median 211.000000 NA 30.5000000 12.5000000 1.000000000  
## mean 211.000000 NA 30.5000000 12.5000000 0.503472222  
## SE.mean 1.242404 NA 0.2635170 0.1053299 0.007607955  
## CI.mean.0.95 2.435749 NA 0.5166287 0.2065007 0.014915498  
## var 6668.210234 NA 299.9861079 47.9277611 0.250045825  
## std.dev 81.659110 NA 17.3201070 6.9229879 0.500045823  
## coef.var 0.387010 NA 0.5678724 0.5538390 0.993194461  
## S2\_L\_SM S3\_L\_SM S1\_T\_SM S2\_T\_SM  
## nbr.val 4320.000000000 4320 4320.00000000 4320.00000000  
## nbr.null 2145.000000000 0 0.00000000 0.00000000  
## nbr.na 0.000000000 0 0.00000000 0.00000000  
## min 0.000000000 1 14.21900000 13.67300000  
## max 1.000000000 1 29.77900000 30.34500000  
## range 1.000000000 0 15.56000000 16.67200000  
## sum 2175.000000000 4320 97104.41000000 97062.69100000  
## median 1.000000000 1 22.51450000 22.52700000  
## mean 0.503472222 1 22.47787269 22.46821551  
## SE.mean 0.007607955 0 0.03202347 0.03243521  
## CI.mean.0.95 0.014915498 0 0.06278243 0.06358967  
## var 0.250045825 0 4.43017031 4.54482580  
## std.dev 0.500045823 0 2.10479698 2.13185970  
## coef.var 0.993194461 0 0.09363862 0.09488336  
## S3\_T\_SM FN\_SM  
## nbr.val 4320.00000000 4320.00000000  
## nbr.null 0.00000000 0.00000000  
## nbr.na 0.00000000 0.00000000  
## min 16.25800000 987.00000000  
## max 30.45000000 1230.00000000  
## range 14.19200000 243.00000000  
## sum 99328.97100000 4757839.00000000  
## median 22.98250000 1101.00000000  
## mean 22.99281736 1101.35162037  
## SE.mean 0.03033041 0.50795770  
## CI.mean.0.95 0.05946318 0.99585787  
## var 3.97411451 1114.65081932  
## std.dev 1.99351812 33.38638674  
## coef.var 0.08670178 0.03031401

summary(df)

## Room\_SM Ren\_SM DT\_SM TM\_SM S1\_L\_SM   
## Min. :111 New:2880 Min. : 1.00 Min. : 1.00 Min. :0.0000   
## 1st Qu.:111 Old:1440 1st Qu.:15.75 1st Qu.: 6.75 1st Qu.:0.0000   
## Median :211 Median :30.50 Median :12.50 Median :1.0000   
## Mean :211 Mean :30.50 Mean :12.50 Mean :0.5035   
## 3rd Qu.:311 3rd Qu.:45.25 3rd Qu.:18.25 3rd Qu.:1.0000   
## Max. :311 Max. :60.00 Max. :24.00 Max. :1.0000   
## S2\_L\_SM S3\_L\_SM S1\_T\_SM S2\_T\_SM S3\_T\_SM   
## Min. :0.0000 Min. :1 Min. :14.22 Min. :13.67 Min. :16.26   
## 1st Qu.:0.0000 1st Qu.:1 1st Qu.:21.07 1st Qu.:21.03 1st Qu.:21.62   
## Median :1.0000 Median :1 Median :22.51 Median :22.53 Median :22.98   
## Mean :0.5035 Mean :1 Mean :22.48 Mean :22.47 Mean :22.99   
## 3rd Qu.:1.0000 3rd Qu.:1 3rd Qu.:23.95 3rd Qu.:23.92 3rd Qu.:24.35   
## Max. :1.0000 Max. :1 Max. :29.78 Max. :30.34 Max. :30.45   
## FN\_SM   
## Min. : 987   
## 1st Qu.:1079   
## Median :1101   
## Mean :1101   
## 3rd Qu.:1123   
## Max. :1230

All the columns consists of valuable information so there is no need to drop any columns.

### d. Apply the High Correlation Filter to remove appropriate columns of data.

numeric\_cols <- df[,3:11]  
  
cor(numeric\_cols,method="spearman")

## Warning in cor(numeric\_cols, method = "spearman"): the standard deviation is  
## zero

## DT\_SM TM\_SM S1\_L\_SM S2\_L\_SM S3\_L\_SM  
## DT\_SM 1.000000000 0.000000000 -0.012257327 -0.012257327 NA  
## TM\_SM 0.000000000 1.000000000 0.016754102 0.016754102 NA  
## S1\_L\_SM -0.012257327 0.016754102 1.000000000 1.000000000 NA  
## S2\_L\_SM -0.012257327 0.016754102 1.000000000 1.000000000 NA  
## S3\_L\_SM NA NA NA NA 1  
## S1\_T\_SM 0.011243713 -0.005405957 0.004716144 0.004716144 NA  
## S2\_T\_SM 0.005996509 0.000727548 -0.007598881 -0.007598881 NA  
## S3\_T\_SM 0.006933454 -0.002157974 0.020696867 0.020696867 NA  
## FN\_SM -0.011117654 0.012908367 -0.023355684 -0.023355684 NA  
## S1\_T\_SM S2\_T\_SM S3\_T\_SM FN\_SM  
## DT\_SM 0.011243713 0.0059965092 0.0069334536 -0.011117654  
## TM\_SM -0.005405957 0.0007275480 -0.0021579742 0.012908367  
## S1\_L\_SM 0.004716144 -0.0075988811 0.0206968671 -0.023355684  
## S2\_L\_SM 0.004716144 -0.0075988811 0.0206968671 -0.023355684  
## S3\_L\_SM NA NA NA NA  
## S1\_T\_SM 1.000000000 0.1033000621 -0.0130664380 0.012990484  
## S2\_T\_SM 0.103300062 1.0000000000 0.0002544604 0.003486114  
## S3\_T\_SM -0.013066438 0.0002544604 1.0000000000 0.001813274  
## FN\_SM 0.012990484 0.0034861137 0.0018132745 1.000000000

From the data above it’s clear that the values for light in Sensor 1 and Sensor 2 are highly correlated, so we will be removing one of them.

df <- df[-c(6)]  
head(df)

## Room\_SM Ren\_SM DT\_SM TM\_SM S1\_L\_SM S3\_L\_SM S1\_T\_SM S2\_T\_SM S3\_T\_SM FN\_SM  
## 1 111 New 47 21 1 1 24.479 21.651 22.227 1106  
## 2 111 New 3 2 1 1 23.771 24.167 22.104 1146  
## 3 311 Old 27 10 1 1 24.760 21.143 22.881 1109  
## 4 211 New 5 22 1 1 22.130 23.925 19.353 1111  
## 5 311 Old 28 18 1 1 24.425 24.832 24.287 1120  
## 6 211 New 9 18 1 1 22.158 20.393 26.612 1078

### e. Based on our discussions in class, what are some specific benefits of reducing the dimensionality of this particular dataset?

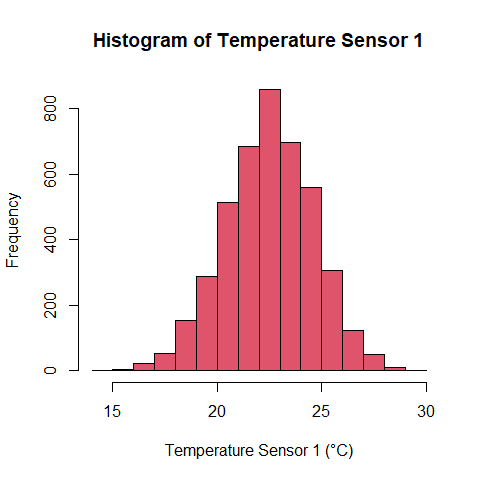
In the current data set I had removed 2 variables out of 12 making the entire memory usgage 16.66% less and hence reducing the response time by the 16.66% and bringing improvement in the computational efficiency.

# Organizing Data

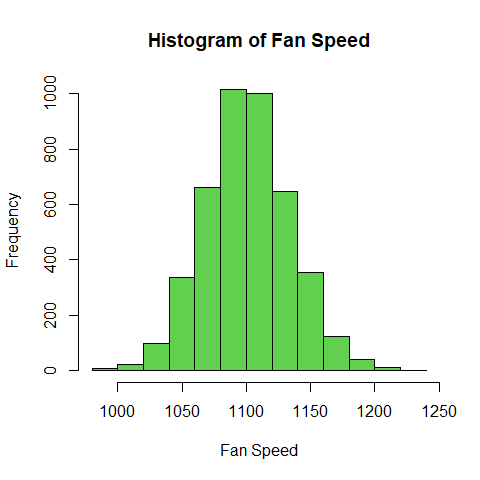
## 1. Scatter Plots

### a. Create a histogram for Temperature from Sensor 1.

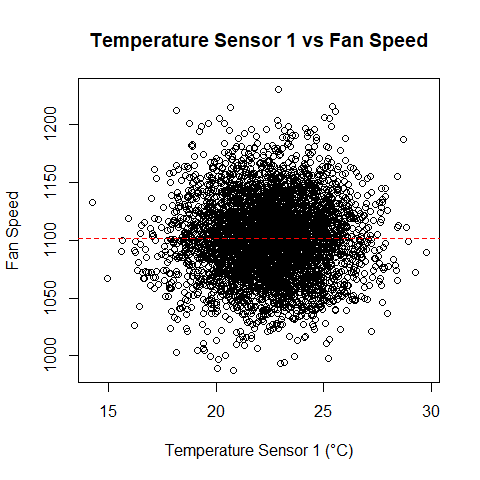
hist(df$S1\_T\_SM,  
 col=2,  
 xlab="Temperature Sensor 1 (°C)",  
 main="Histogram of Temperature Sensor 1")



hist(df$FN\_SM,  
 col=3,   
 xlab="Fan Speed",  
 main="Histogram of Fan Speed")



plot(FN\_SM ~ S1\_T\_SM,  
 data = df,  
 main = "Temperature Sensor 1 vs Fan Speed",  
 xlab = "Temperature Sensor 1 (°C)",  
 ylab = "Fan Speed")  
  
# Add a horizontal line at the mean fan speed  
abline(h = mean(df$FN\_SM), col = "red", lty = 2)

 ### d. What conclusions, if any, can you draw from the chart? The data from the scatter plot shows to be densely populated in the center between 20 - 25 C and fan speed of 1050 - 1150

It shows No Linear Relationship between the Temperature and the Fan Speeds as there are many points which shows the opposite. Low fan speed and yet a high temperature or vice verca

print("Pearson Correlation")

## [1] "Pearson Correlation"

round(cor(df$FN\_SM, df$S1\_T\_SM),3)

## [1] 0.017

I went with Pearson Method as it measures the linear relationship between two continuous variables

Since r is close to 0, then there is no linear relationship between the variables.

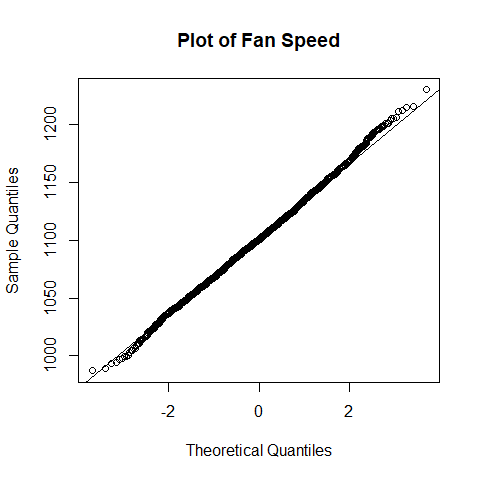
REFERENCE: <https://datascientest.com/en/pearson-and-spearman-correlations-a-guide-to-understanding-and-applying-correlation-methods>

# 3. INFERENCE

## 1. Normality

### a. Create a QQ Normal plot of for Fan Speed.

# QQ plot for Fan Speed  
qqnorm(df$FN, main = "Plot of Fan Speed")  
qqline(df$FN)

 ### b.Conduct a statistical test for normality on Fan Speed.

shapiro.test(df$FN\_SM)

##   
## Shapiro-Wilk normality test  
##   
## data: df$FN\_SM  
## W = 0.99897, p-value = 0.009554

### c. Is Fan Speed normally distributed? What led you to this conclusion?

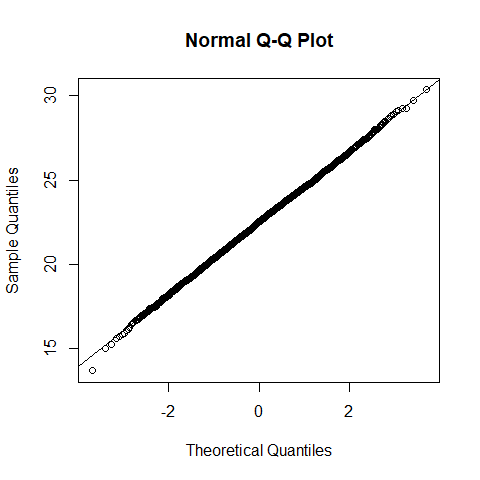
QQ Normal shows that the majority of the points lie on the lie but still can not confirm yet however we can say that approximately the Fan Speed in Normally Distributed

Performing the Shapiro Wilks Test, confirms further, since P value is less than 5% it confirms our suspicion that the Fan speed is not normally distributed

## 2. Statistically Significant Differences

### a. Compare Temperature from Sensor 2 between ‘New’ and ‘Old’ rooms in your dataset using a suitable hypothesis test.

# Performing Normality Tests   
  
qqnorm(df$S2\_T\_SM)  
qqline(df$S2\_T\_SM)



shapiro.test(df$S2\_T\_SM)

##   
## Shapiro-Wilk normality test  
##   
## data: df$S2\_T\_SM  
## W = 0.99964, p-value = 0.6667

# Performing the T-Test since it matches the assumptions   
  
t.test(S2\_T\_SM ~ Ren\_SM, data = df, var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: S2\_T\_SM by Ren\_SM  
## t = -9.3776, df = 4318, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group New and group Old is not equal to 0  
## 95 percent confidence interval:  
## -0.7723875 -0.5052743  
## sample estimates:  
## mean in group New mean in group Old   
## 22.25527 22.89410

### b. Explain why you chose the test you did.

I chose to go with t-test for couple of reasons: 1. We have a continuous data so had to compare the means of it .

1. Since two groups are being compared and the data is normally distributed which meets all the assumptions for T-Test to be performed.

### c. Do you have strong evidence that Temperature from Sensor 2 is different between new and old rooms?

Yes, the p value is significantly less than 5% that clearly indicates the difference of temperature between the two rooms.

## 3. Multiple Statistical Differences

### a. Determine if Temperature from Sensor 1 varies by Room Number using ANOVA (statistical) and a sequence of boxplots (graphical).

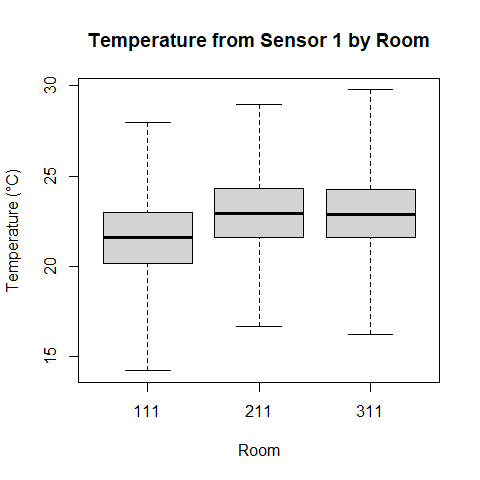
summary(aov(S1\_T\_SM ~ Room\_SM, data=df))

## Df Sum Sq Mean Sq F value Pr(>F)   
## Room\_SM 1 1238 1238.4 298.8 <2e-16 \*\*\*  
## Residuals 4318 17896 4.1   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

print(" ")

## [1] " "

boxplot(S1\_T\_SM ~ Room\_SM, data=df,  
 main="Temperature from Sensor 1 by Room",  
 xlab = "Room",   
 ylab = "Temperature (°C)",  
 range=0)

 Conclusion: A Significant p-value less than 0.05 suggests that temperatures varies across the different rooms. Boxplot further confirms visually the difference in temperatures across the rooms.

### b. Determine if Temperature from Sensor 3 varies by Room Number using ANOVA (statistical) and a sequence of boxplots (graphical).

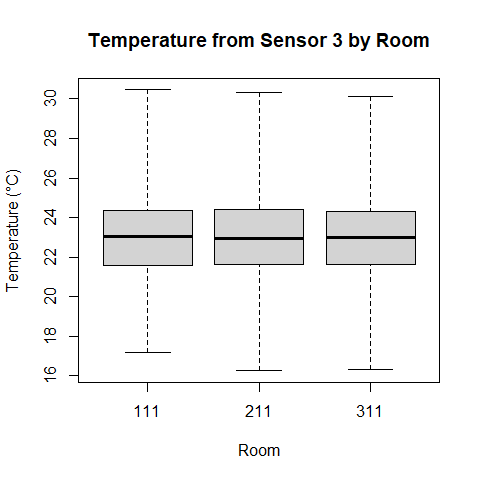
summary(aov(S3\_T\_SM ~ Room\_SM, data=df))

## Df Sum Sq Mean Sq F value Pr(>F)  
## Room\_SM 1 0 0.003 0.001 0.977  
## Residuals 4318 17164 3.975

print(" ")

## [1] " "

boxplot(S3\_T\_SM ~ Room\_SM, data=df,  
 main="Temperature from Sensor 3 by Room",  
 xlab = "Room",   
 ylab = "Temperature (°C)",  
 range=0)

 Conclusion: A Significant p-value= 0.977 is greater than 0.05 suggests that temperatures are similar across the different rooms. Box plot further confirms visually the similarity in temperatures across the rooms.