Developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

Prediction of Residential Energy Consumption by Household Appliances with Machine Learning

This IoT project aims to create predictive models for forecasting appliance energy consumption. The data used includes temperature and humidity sensor measurements from a wireless network, weather forecasts from an airport station, and energy usage from luminaries.

In this machine learning project, data filtering is required to remove non-predictive parameters and select the best features for prediction. The dataset was collected over a period of 10 minutes for approximately 5 months. The house's temperature and humidity conditions were monitored using a ZigBee wireless sensor network.

Each wireless node transmitted temperature and humidity conditions every 3 minutes. Then, the average of the data was calculated for 10-minute intervals. Energy data was recorded every 10 minutes using "m-bus" energy meters. The weather time from the nearest weather station (Chievres Airport, Belgium) was downloaded from a public dataset from Reliable Prognosis (rp5.ru) and merged with the experimental datasets using the date and time column. Two random variables were included in the dataset to test regression models and filter out non-predictive attributes (parameters).

Our task now is to build a predictive model that can forecast energy consumption based on the collected IoT sensor data. We will be using the R language for this project.

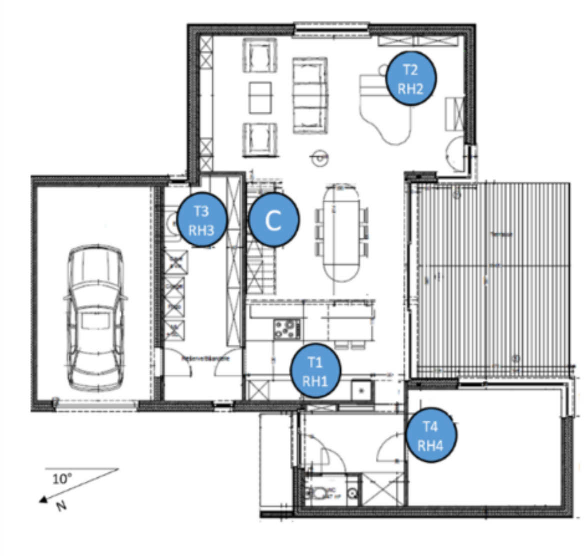
The data can be downloaded from the following link:

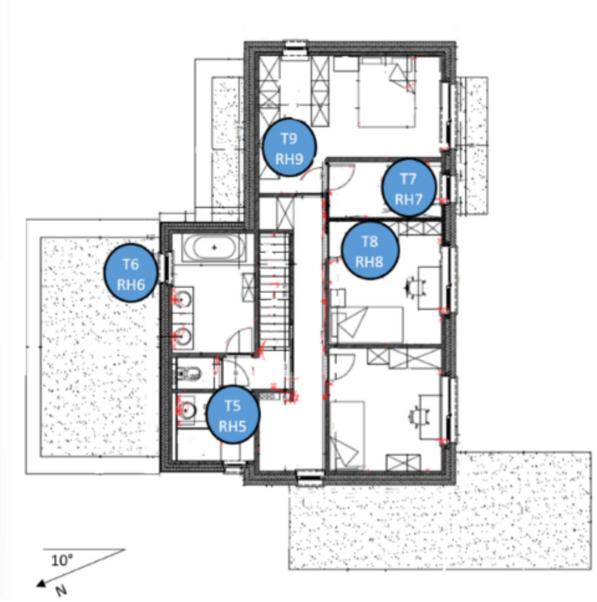
https://www.kaggle.com/competitions/appliances-energy-prediction

Data Dictionary

- • Appliances: energy used by household appliances in Wh (watt-hours)
- • Lights: energy used by house lights in Wh
- • T1: temperature in the Kitchen area in Celsius
- • RH_1: humidity in the Kitchen area in %
- • T2: temperature in the Living Room area in Celsius
- • RH_2: humidity in the Living Room area in %
- • T3: temperature in the Laundry area in Celsius
- • RH_3: humidity in the Laundry area in %
- • T4: temperature in the Office area in Celsius
- • RH_4: humidity in the Office area in %
- • T5: temperature in the Bathroom area in Celsius
- • RH_5: humidity in the Bathroom area in %

- • T6: temperature in the Outside area (North side) in Celsius
- • RH_6: humidity in the Outside area (North side) in %
- • T7: temperature in the Ironing Room area in Celsius
- • RH_7: humidity in the Ironing Room area in %
- • T8: temperature in the Children's Room area in Celsius
- • RH_8: humidity in the Children's Room area in %
- • T9: temperature in the Master Bedroom area in Celsius
- • RH_9: humidity in the Master Bedroom area in %
- • To: Local temperature (measured from Chievres Weather Station) in Celsius
- • Pressure: Atmospheric pressure (measured from Chievres Weather Station) in mmHg
- • RH_out: Local humidity (measured from Chievres Weather Station) in %
- • Windspeed: Wind speed (measured from Chievres Weather Station) in m/s
- • Visibility: Visibility (measured from Chievres Weather Station) in km
- • Tdewpoint: Dew Point Temperature or Dew Point (measured from Chievres Weather Station) in Celsius
- • rv1: Non-dimensional Random Variable
- • rv2: Non-dimensional Random Variable

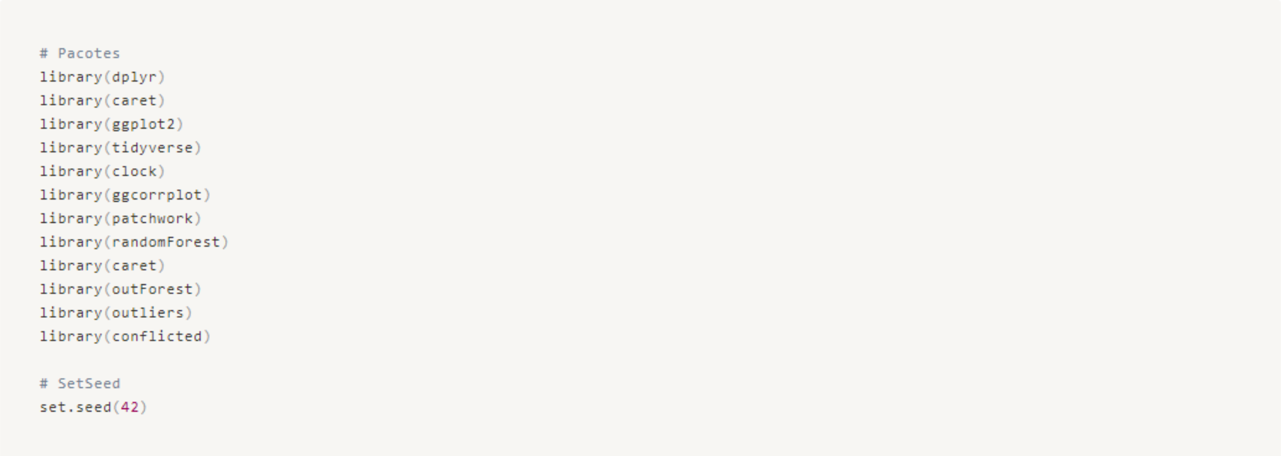




Developed by Thiago Bulgarelli

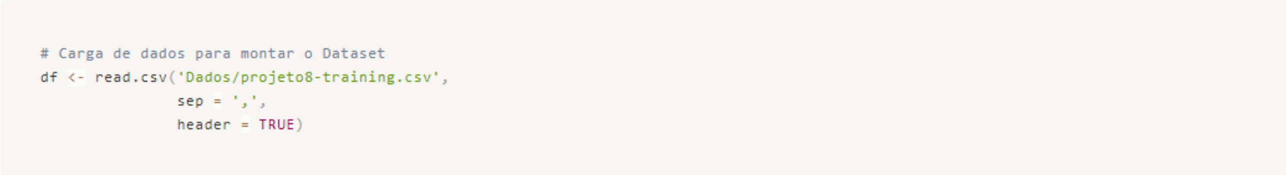
Contact: bugath36@gmail.com

Loading Packages



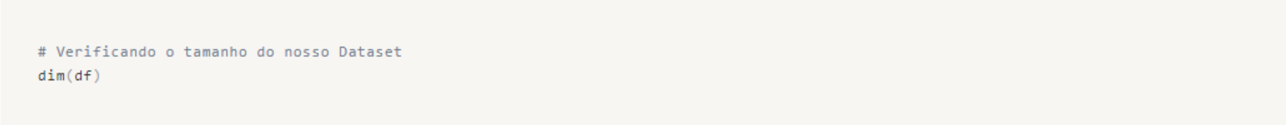
```
# Pacotes
library(dplyr)
library(caret)
library(ggplot2)
library(tidyverse)
library(clock)
library(ggcorrplot)
library(patchwork)
library(randomForest)
library(caret)
library(outForest)
library(outliers)
library(conflicted)

# SetSeed
set.seed(42)
```

Loading the Data

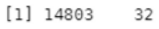


```
# Carga de dados para montar o Dataset
df <- read.csv('Dados/projeto8-training.csv',
              sep = ',',
              header = TRUE)
```

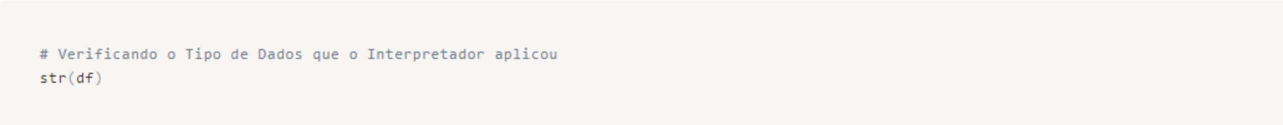
Data Cleaning and Organization



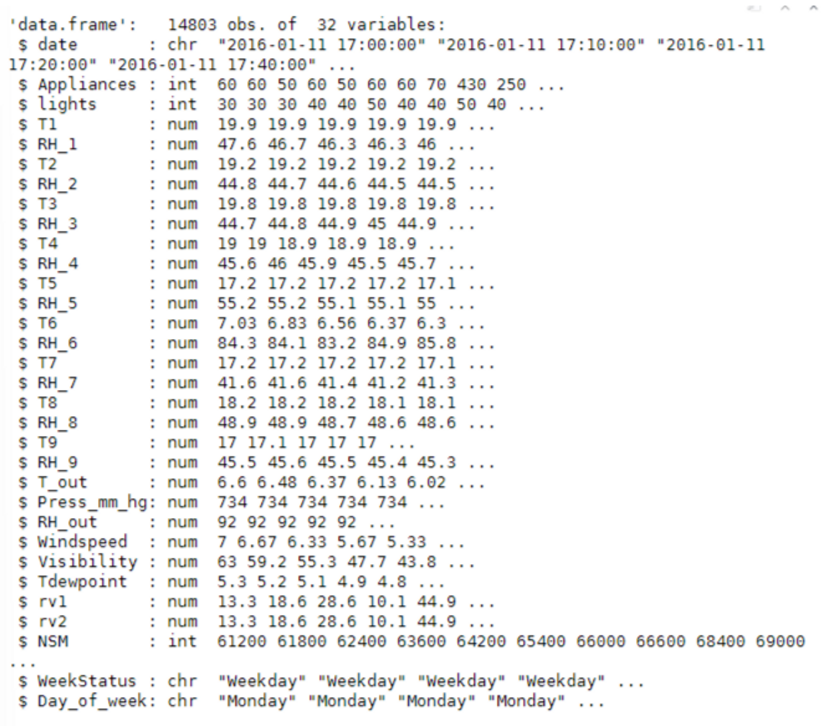
```
# Verificando o tamanho do nosso Dataset
dim(df)
```



```
[1] 14803    32
```



```
# Verificando o Tipo de Dados que o Interpretador aplicou
str(df)
```



```
'data.frame':	14803 obs. of  32 variables:
 $ date       : chr  "2016-01-11 17:00:00" "2016-01-11 17:10:00" "2016-01-11
17:20:00" "2016-01-11 17:40:00" ...
 $ Appliances : int  60 60 50 60 50 60 60 70 430 250 ...
 $ lights     : int  30 30 30 40 40 50 40 40 50 40 ...
 $ T1         : num  19.9 19.9 19.9 19.9 19.9 ...
 $ RH_1       : num  47.6 46.7 46.3 46.3 46 ...
 $ T2         : num  19.2 19.2 19.2 19.2 19.2 ...
 $ RH_2       : num  44.8 44.7 44.6 44.5 44.5 ...
 $ T3         : num  19.8 19.8 19.8 19.8 19.8 ...
 $ RH_3       : num  44.7 44.8 44.9 45 44.9 ...
 $ T4         : num  19 19 18.9 18.9 18.9 ...
 $ RH_4       : num  45.6 46 45.9 45.5 45.7 ...
 $ T5         : num  17.2 17.2 17.2 17.2 17.1 ...
 $ RH_5       : num  55.2 55.2 55.1 55.1 55 ...
 $ T6         : num  7.03 6.83 6.56 6.37 6.3 ...
 $ RH_6       : num  84.3 84.1 83.2 84.9 85.8 ...
 $ T7         : num  17.2 17.2 17.2 17.2 17.1 ...
 $ RH_7       : num  41.6 41.6 41.4 41.2 41.3 ...
 $ T8         : num  18.2 18.2 18.2 18.1 18.1 ...
 $ RH_8       : num  48.9 48.9 48.7 48.6 48.6 ...
 $ T9         : num  17 17.1 17 17 17 ...
 $ RH_9       : num  45.5 45.6 45.5 45.4 45.3 ...
 $ T_out      : num  6.6 6.48 6.37 6.13 6.02 ...
 $ Press_mm_hg: num  734 734 734 734 734 ...
 $ RH_out     : num  92 92 92 92 92 ...
 $ Windspeed  : num  7 6.67 6.33 5.67 5.33 ...
 $ Visibility : num  63 59.2 55.3 47.7 43.8 ...
 $ Tdewpoint  : num  5.3 5.2 5.1 4.9 4.8 ...
 $ rv1        : num  13.3 18.6 28.6 10.1 44.9 ...
 $ rv2        : num  13.3 18.6 28.6 10.1 44.9 ...
 $ NSM        : int  61200 61800 62400 63600 64200 65400 66000 66600 68400 69000
...
 $ WeekStatus : chr  "Weekday" "Weekday" "Weekday" "Weekday" ...
 $ Day_of_week: chr  "Monday" "Monday" "Monday" "Monday" ...
```

We can observe that we have some variables of type "CHR". The "date" variable should be of the date type so that we can, if necessary later on, break it down into day, month, and year. Therefore, let's change its type at this moment.

For the variables "WeekStatus" and "Day_of_Week", we will analyze their unique values and apply the factor type.

```
# Transformando Datas do Formato CHR para Date
df$date <- strptime(df$date,
                    format = "%Y-%m-%d %H:%M:%S")
```

```
# Transformando Variáveis CHR para tipo Fator
df$WeekStatus <- as.factor(df$WeekStatus)
df$Day_of_week <- as.factor(df$Day_of_week)
```

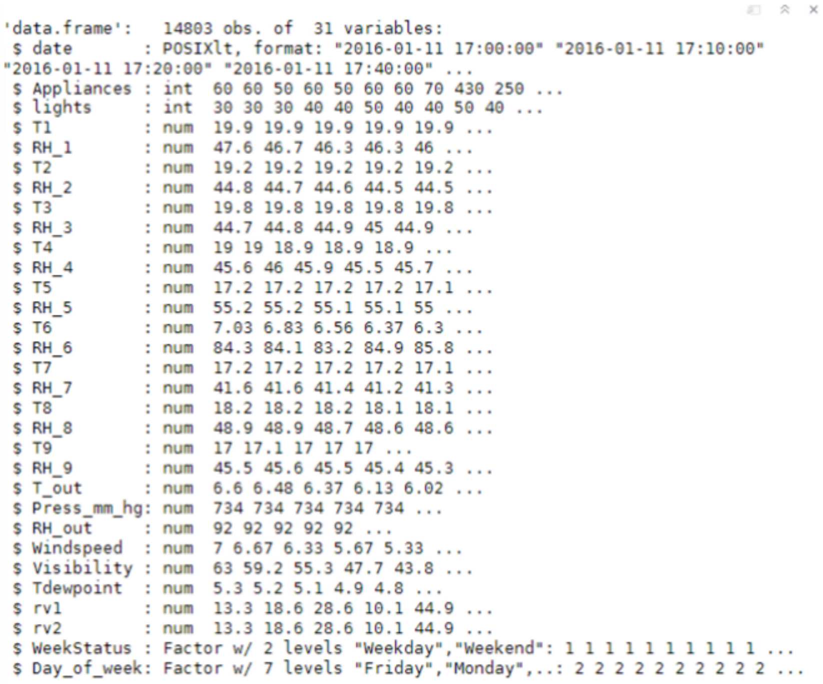
After making the necessary adjustments, let's check for missing data.

```
# Somando as observações sem informação (NaN)
sum(is.na(df))
```

```
[1] 0
```

Finally, we do not have any missing values in our dataset. To conclude, let's remove the "MSM" column as we do not have a description of what this variable represents, so we will consider it unnecessary.

```
# Removendo MSM
df$NSM <- NULL
str(df)
```



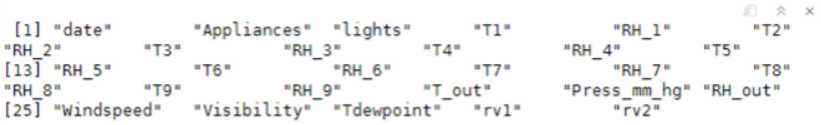
```
'data.frame':   14803 obs. of  31 variables:
 $ date       : POSIXlt, format: "2016-01-11 17:00:00" "2016-01-11 17:10:00"
"2016-01-11 17:20:00" "2016-01-11 17:40:00" ...
 $ Appliances : int  60 60 50 60 50 60 60 70 430 250 ...
 $ lights     : int  30 30 30 40 40 50 40 40 50 40 ...
 $ T1         : num  19.9 19.9 19.9 19.9 19.9 ...
 $ RH_1       : num  47.6 46.7 46.3 46.3 46 ...
 $ T2         : num  19.2 19.2 19.2 19.2 19.2 ...
 $ RH_2       : num  44.8 44.7 44.6 44.5 44.5 ...
 $ T3         : num  19.8 19.8 19.8 19.8 19.8 ...
 $ RH_3       : num  44.7 44.8 44.9 45 44.9 ...
 $ T4         : num  19 19 18.9 18.9 18.9 ...
 $ RH_4       : num  45.6 46 45.9 45.5 45.7 ...
 $ T5         : num  17.2 17.2 17.2 17.2 17.1 ...
 $ RH_5       : num  55.2 55.2 55.1 55.1 55 ...
 $ T6         : num  7.03 6.83 6.56 6.37 6.3 ...
 $ RH_6       : num  84.3 84.1 83.2 84.9 85.8 ...
 $ T7         : num  17.2 17.2 17.2 17.2 17.1 ...
 $ RH_7       : num  41.6 41.6 41.4 41.2 41.3 ...
 $ T8         : num  18.2 18.2 18.2 18.1 18.1 ...
 $ RH_8       : num  48.9 48.9 48.7 48.6 48.6 ...
 $ T9         : num  17 17.1 17 17 17 ...
 $ RH_9       : num  45.5 45.6 45.5 45.4 45.3 ...
 $ T_out      : num  6.6 6.48 6.37 6.13 6.02 ...
 $ Press_mm_hg: num  734 734 734 734 734 ...
 $ RH_out     : num  92 92 92 92 92 ...
 $ Windspeed  : num  7 6.67 6.33 5.67 5.33 ...
 $ Visibility : num  63 59.2 55.3 47.7 43.8 ...
 $ Tdewpoint  : num  5.3 5.2 5.1 4.9 4.8 ...
 $ rv1        : num  13.3 18.6 28.6 10.1 44.9 ...
 $ rv2        : num  13.3 18.6 28.6 10.1 44.9 ...
 $ WeekStatus : Factor w/ 2 levels "Weekday","Weekend": 1 1 1 1 1 1 1 1 1 1 ...
 $ Day_of_week: Factor w/ 7 levels "Friday","Monday",..: 2 2 2 2 2 2 2 2 2 2 ...
```

Now we are ready to start the exploratory data analysis!

Exploratory Data Analysis

Let's separate the numerical variables from the categorical variables.

```
# Variáveis Numéricas
colnames(df)[1:29]
N <- df[colnames(df)[1:29]]
```



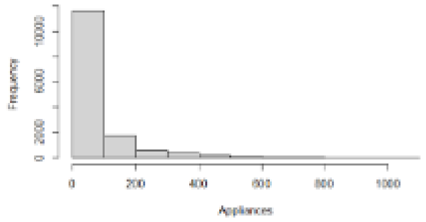
```
 [1] "date"        "Appliances"  "lights"      "T1"          "RH_1"        "T2"
"RH_2"        "T3"          "RH_3"        "T4"          "RH_4"        "T5"
[13] "RH_5"        "T6"          "RH_6"        "T7"          "RH_7"        "T8"
"RH_8"        "T9"          "RH_9"        "T_out"       "Press_mm_hg" "RH_out"
[25] "Windspeed"   "Visibility"  "Tdewpoint"   "rv1"         "rv2"
```

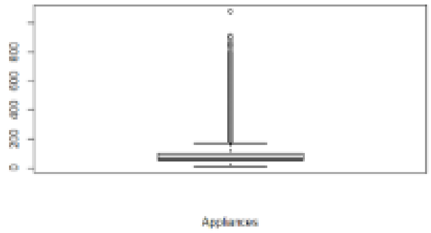
```
# Variáveis Categóricas
C <- df[colnames(df)[30:31]]
```

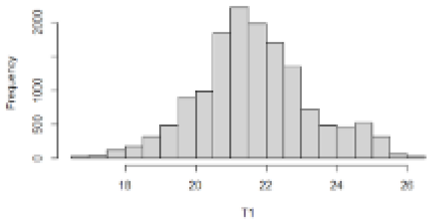
After separating the variables, let's analyze the behavior of each one and identify possible insights.

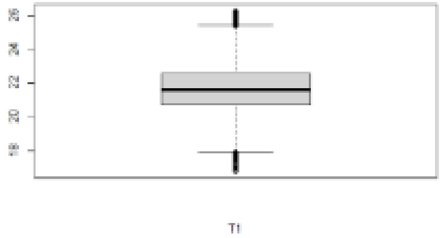
Numerical Variables

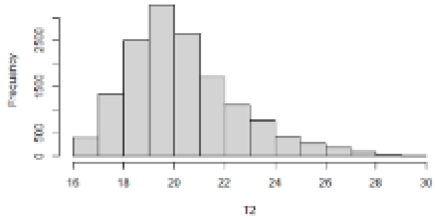
```
# Histogramas das Variáveis Numéricas
for (i in 2:29){
  hist(N[, i],
       xlab = colnames(N[i]),
       main = '')
  boxplot(N[, i],
          xlab = colnames(N[i]),
          main = '')
}
```

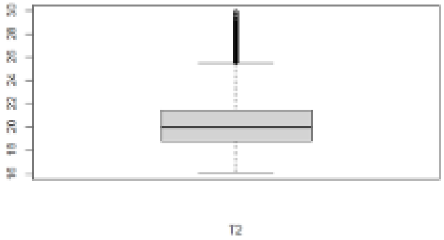


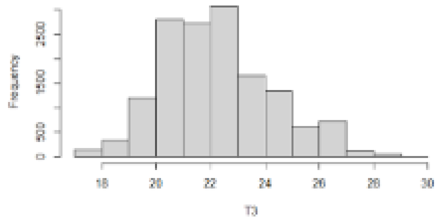


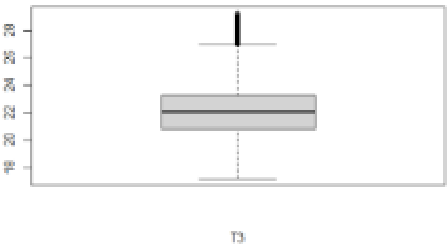


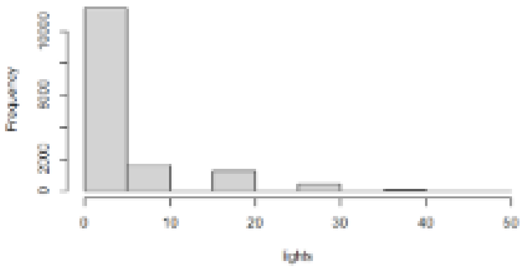


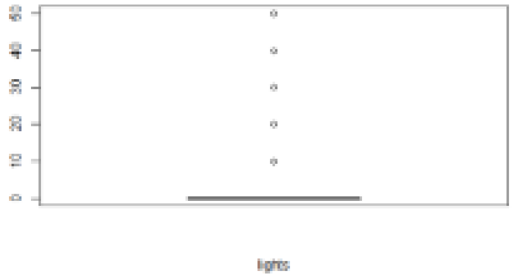


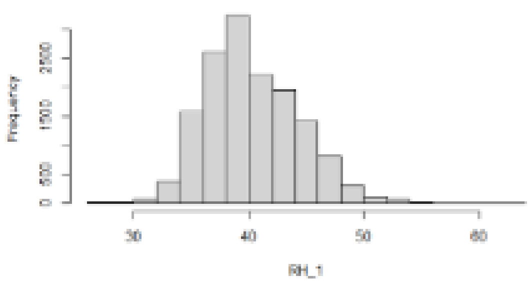


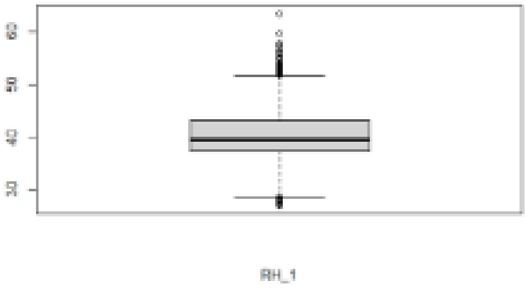


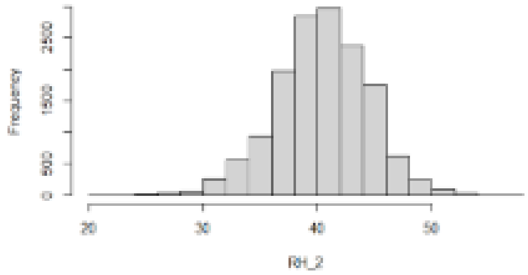


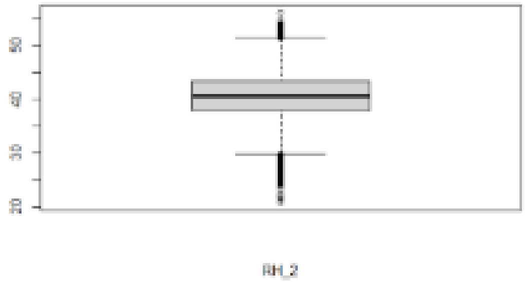

Frequency
0
2000
6000
10000
0
10
20
30
40
50
lights

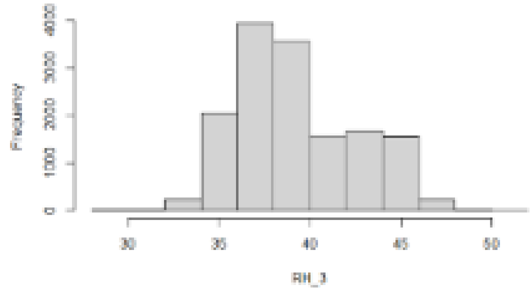

0
10
20
30
40
50
lights

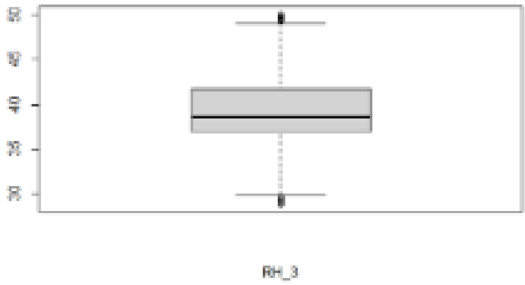

Frequency
0
500
1500
2500
30
40
50
60
RH_1

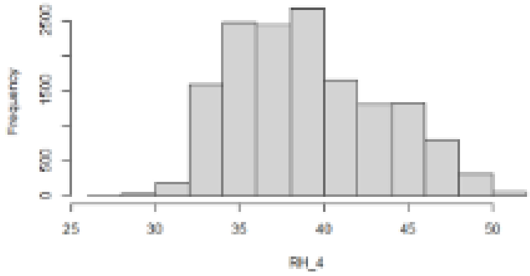

30
40
50
60
RH_1

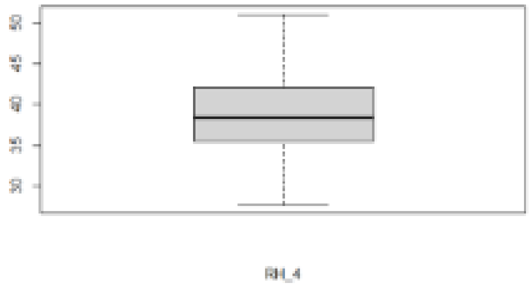

Frequency
0
500
1500
2500
20
30
40
50
RH_2

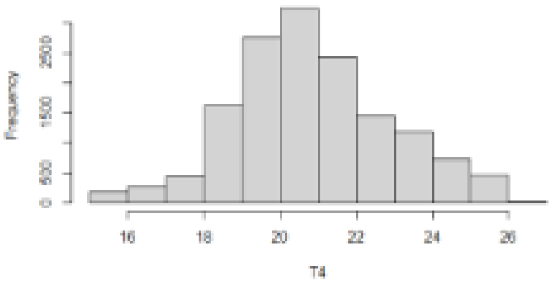

20
30
40
50
RH_2

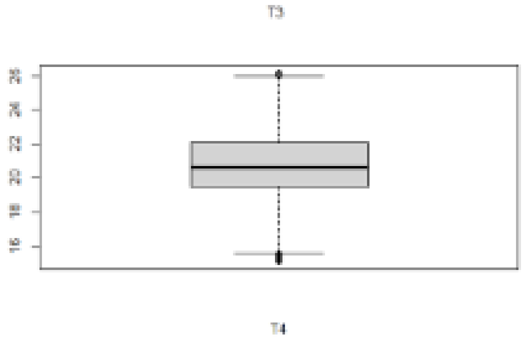

Frequency
0
1000
2000
3000
4000
30
35
40
45
50
RH_3

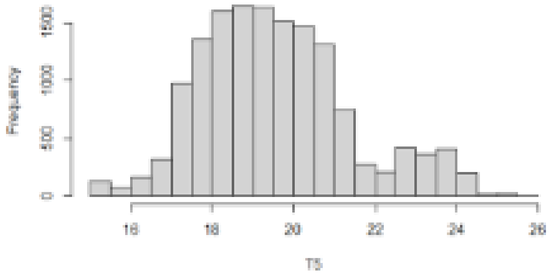

30
35
40
45
50
RH_3

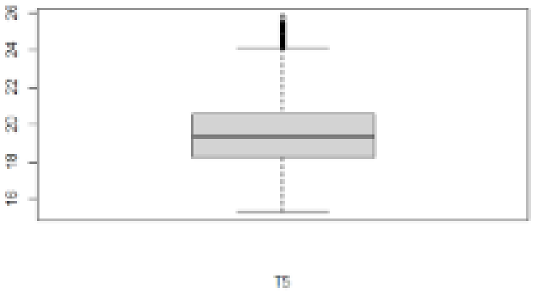

Frequency
0
500
1500
2500
25
30
35
40
45
50
RH_4

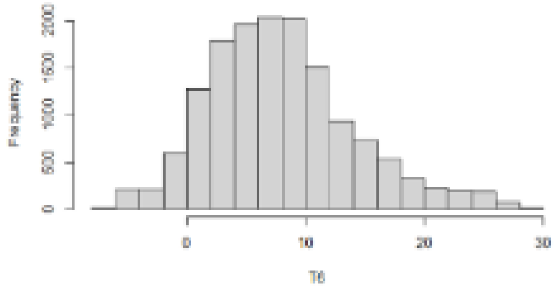

30
35
40
45
50
RH_4

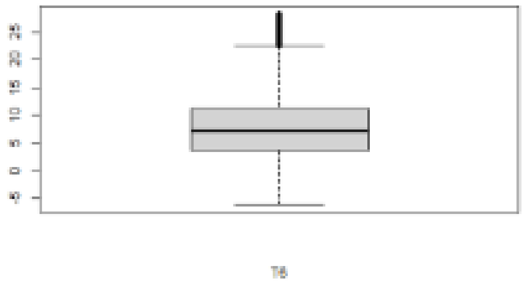

Frequency
0
500
1500
2500
16
18
20
22
24
26
T4

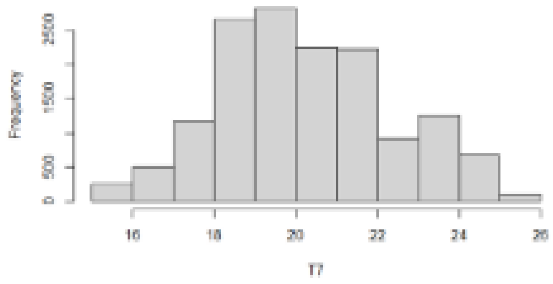

T3
16
18
20
22
24
26
T4

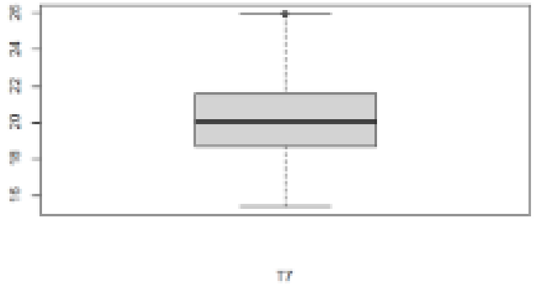

Frequency
0
500
1000
1500
16
18
20
22
24
26
T5

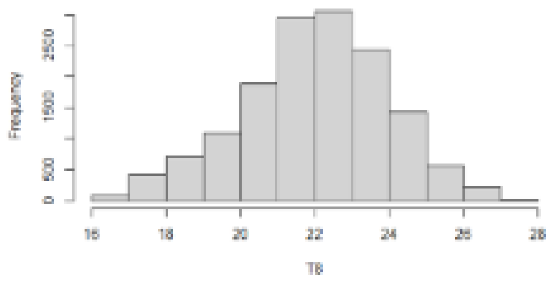

16
18
20
22
24
26
T5

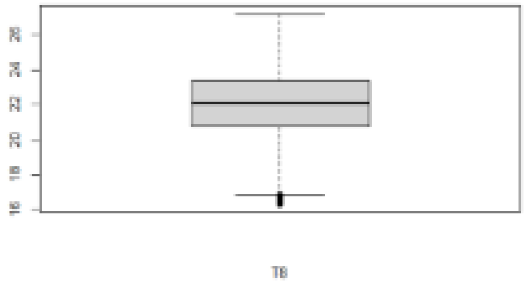

Frequency
0
500
1000
1500
2000
0
10
20
30
T6

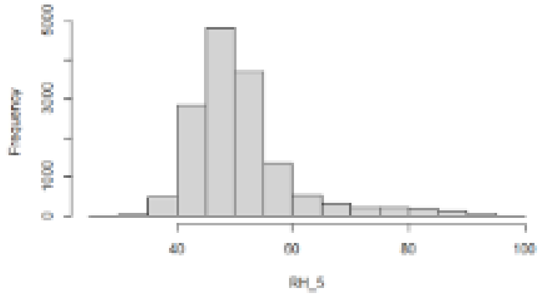

-5
0
5
10
15
20
25
T6

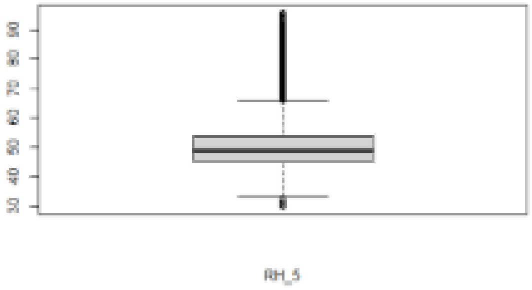

Frequency
0
500
1500
2500
16
18
20
22
24
26
T7

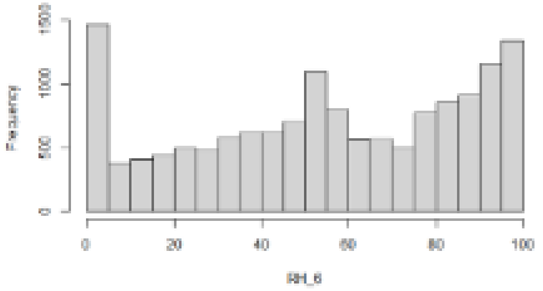

16
18
20
22
24
26
T7

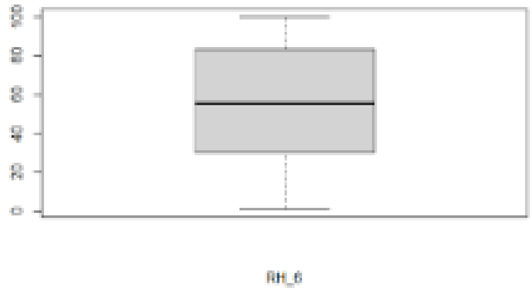

Frequency
0
500
1500
2500
16
18
20
22
24
26
28
T8

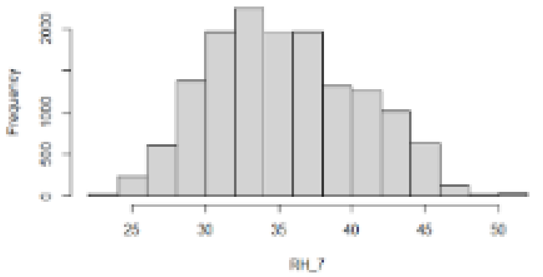

16
18
20
22
24
26
T8

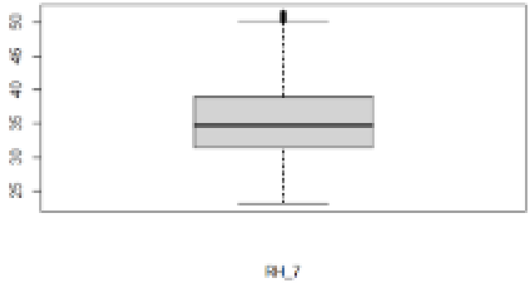

Frequency
0
1000
3000
5000
40
60
80
100
RH_5

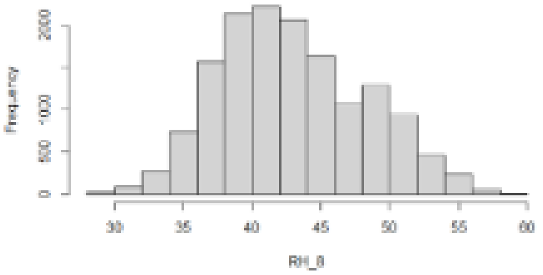

30
40
50
60
70
80
90
RH_5

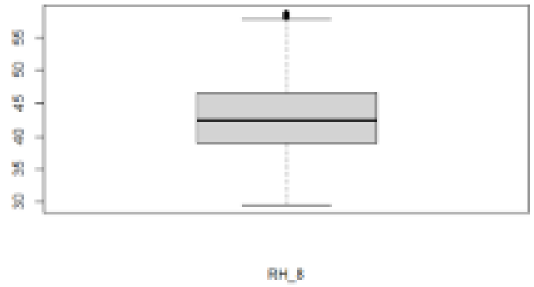

Frequency
0
500
1000
1500
0
20
40
60
80
100
RH_6

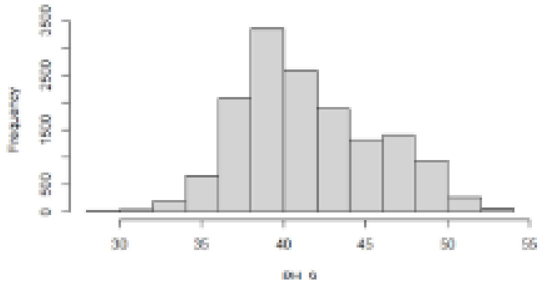

0
20
40
60
80
100
RH_6

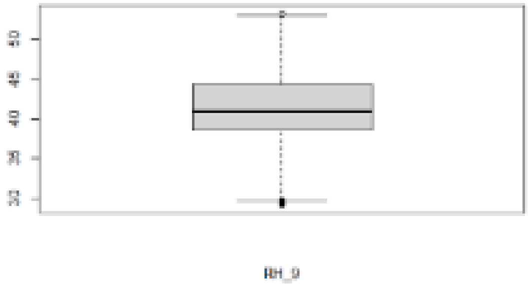

Frequency
0
500
1000
2000
25
30
35
40
45
50
RH_7

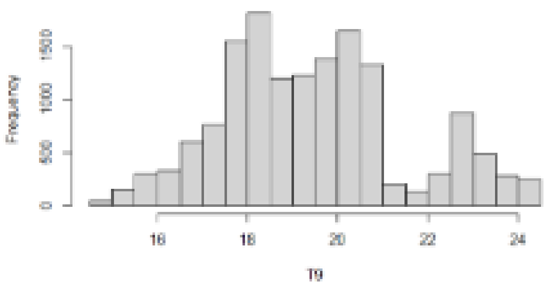

25
30
35
40
45
50
RH_7

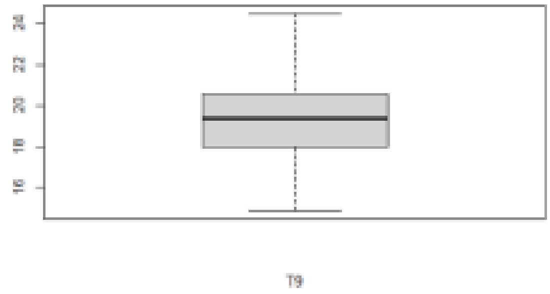

Frequency
0
500
1000
2000
30
35
40
45
50
55
60
RH_8

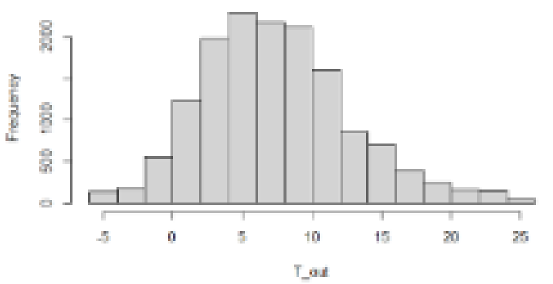

30
35
40
45
50
55
RH_8

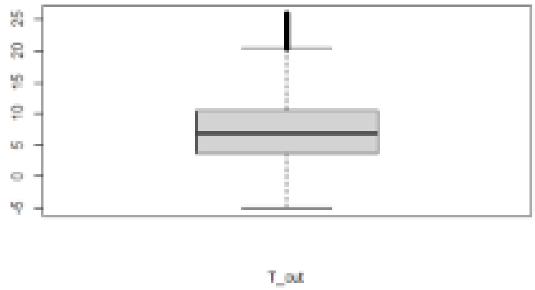

Frequency
0
500
1500
2500
3500
30
35
40
45
50
55
RH_9

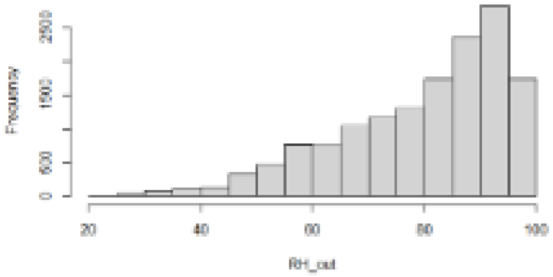

30
35
40
45
50
RH_9

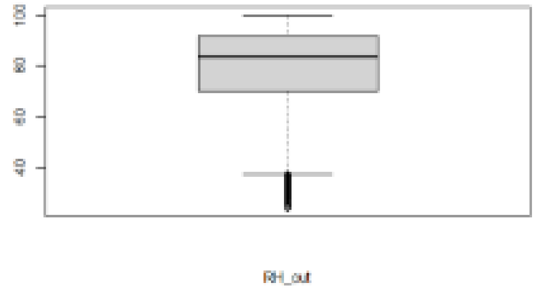

Frequency
0
500
1000
1500
16
18
20
22
24
T9

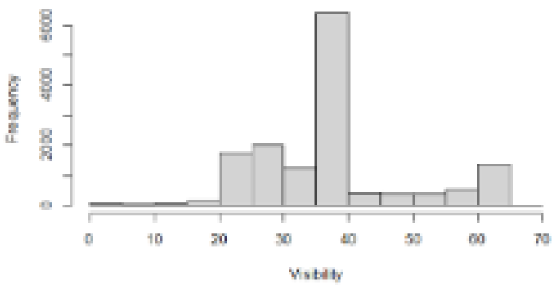

16
18
20
22
24
T9

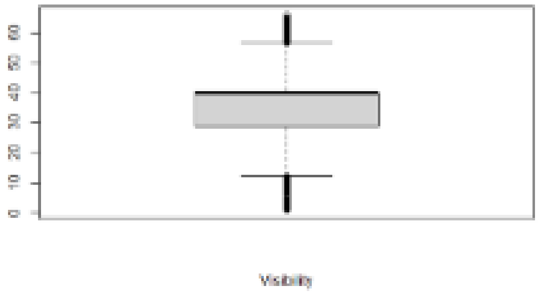

Frequency
0
500
1000
2000
-5
0
5
10
15
20
25
T_out

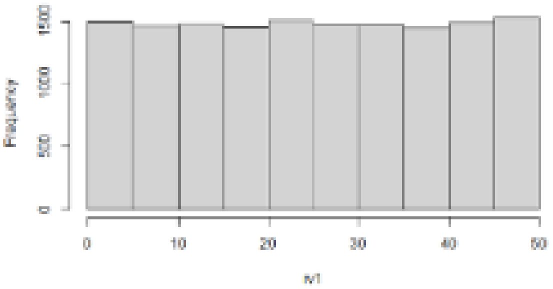

-5
0
5
10
15
20
25
T_out

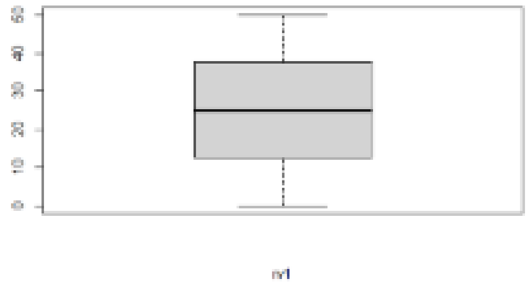

Frequency
0
500
1500
2500
20
40
60
80
100
RH_out

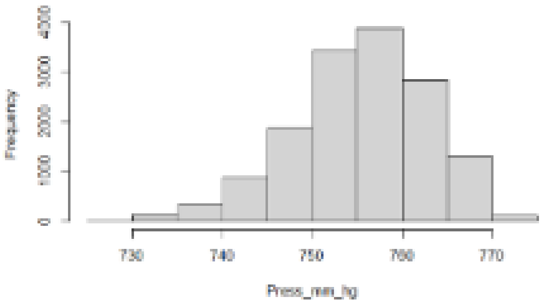

40
60
80
100
RH_out

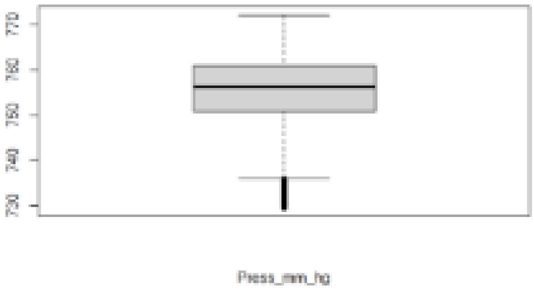

Frequency
0
2000
4000
6000
0
10
20
30
40
50
60
70
Visibility

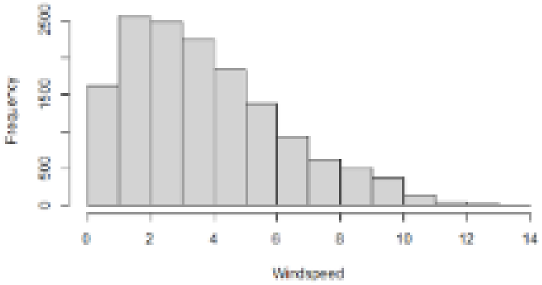

0
10
20
30
40
50
60
Visibility

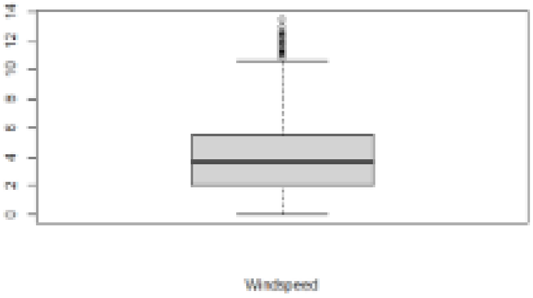

Frequency
0
500
1000
1500
0
10
20
30
40
50
rv1

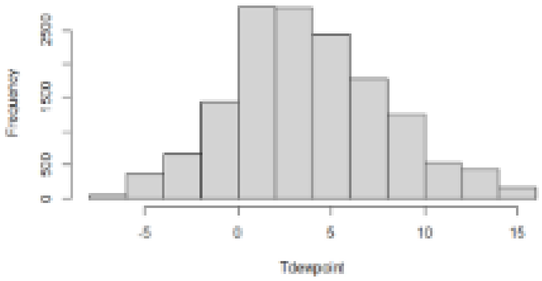

0
10
20
30
40
50
rv1

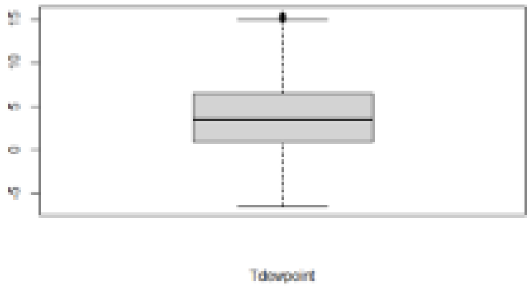


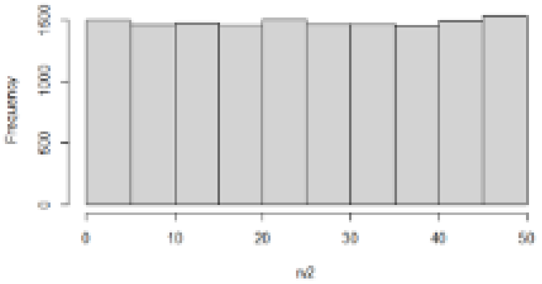


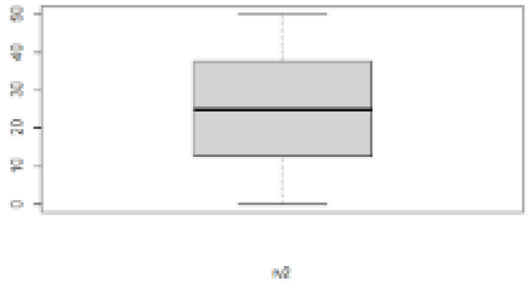










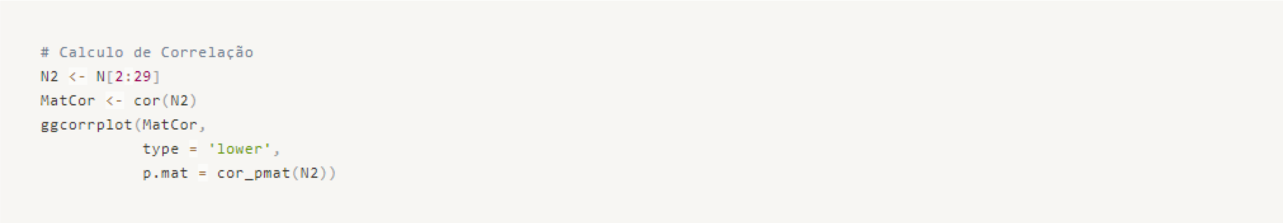


Some interesting observations:

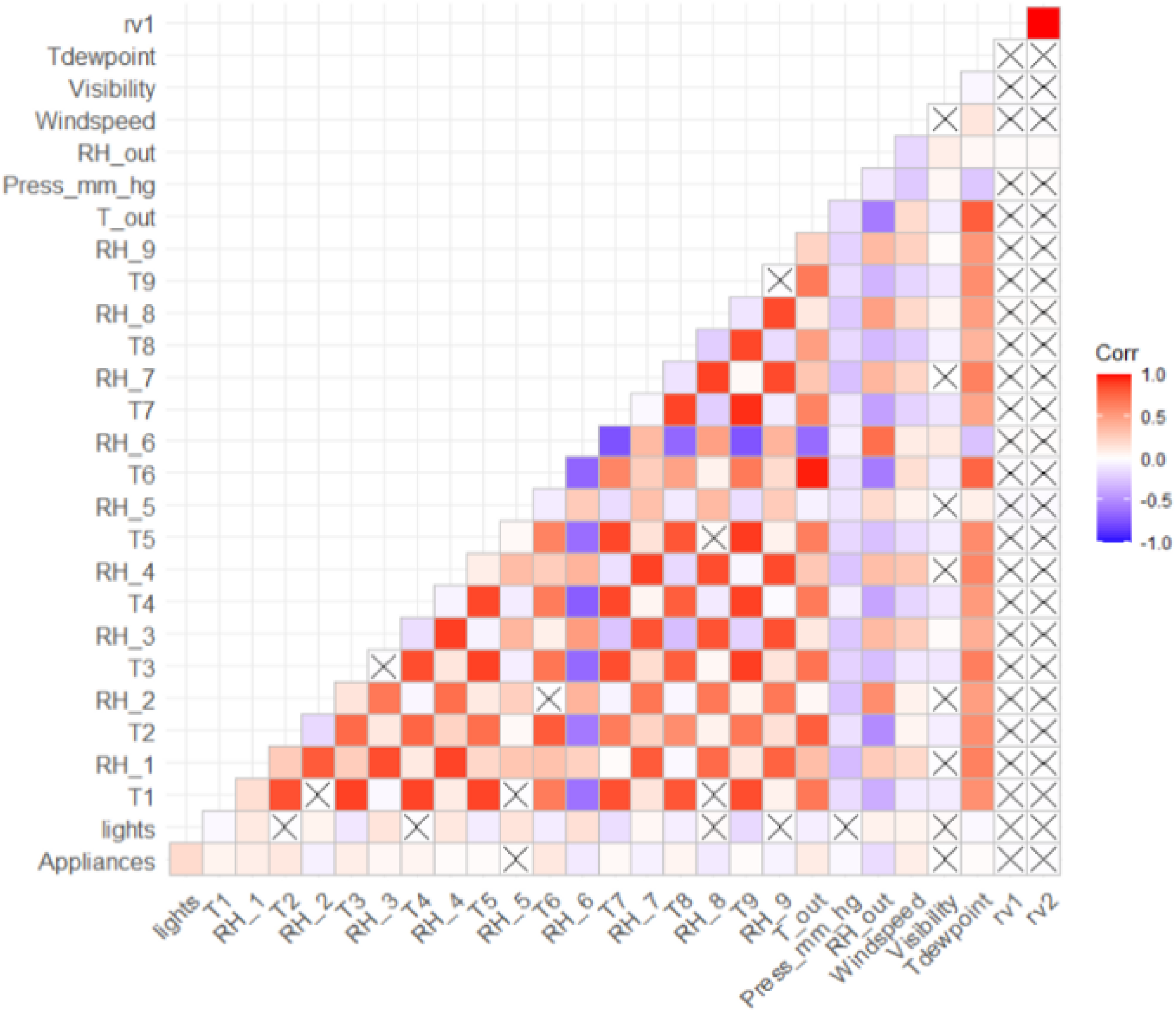
- The temperature and humidity variables have distributions that are very similar to normal, as their mean and median values are very close. We will not apply a Shapiro test to verify the hypothesis of normality as it is not relevant to the initial results of our predictive modeling.

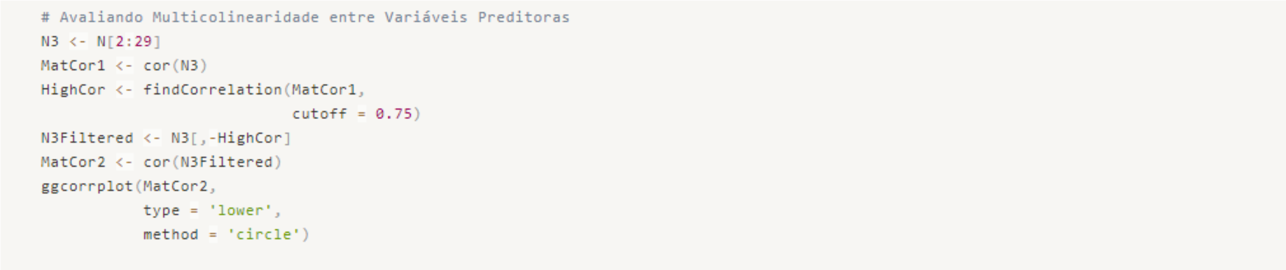
- The variables "r1" and "r2" are almost constant within our observations and will clearly be discarded in our selection of important variables.
- The "lights" variable has a distribution close to normal, but we will further analyze its correlation with the response variable (Appliance) as they are initially competing and independent variables. We will confirm this by examining the correlations.
- The response variable (Appliance) has a distribution that resembles a normal distribution, but it has some outliers, as do most predictor variables. We will address this issue later.

Let's now examine how the variables correlate with each other.

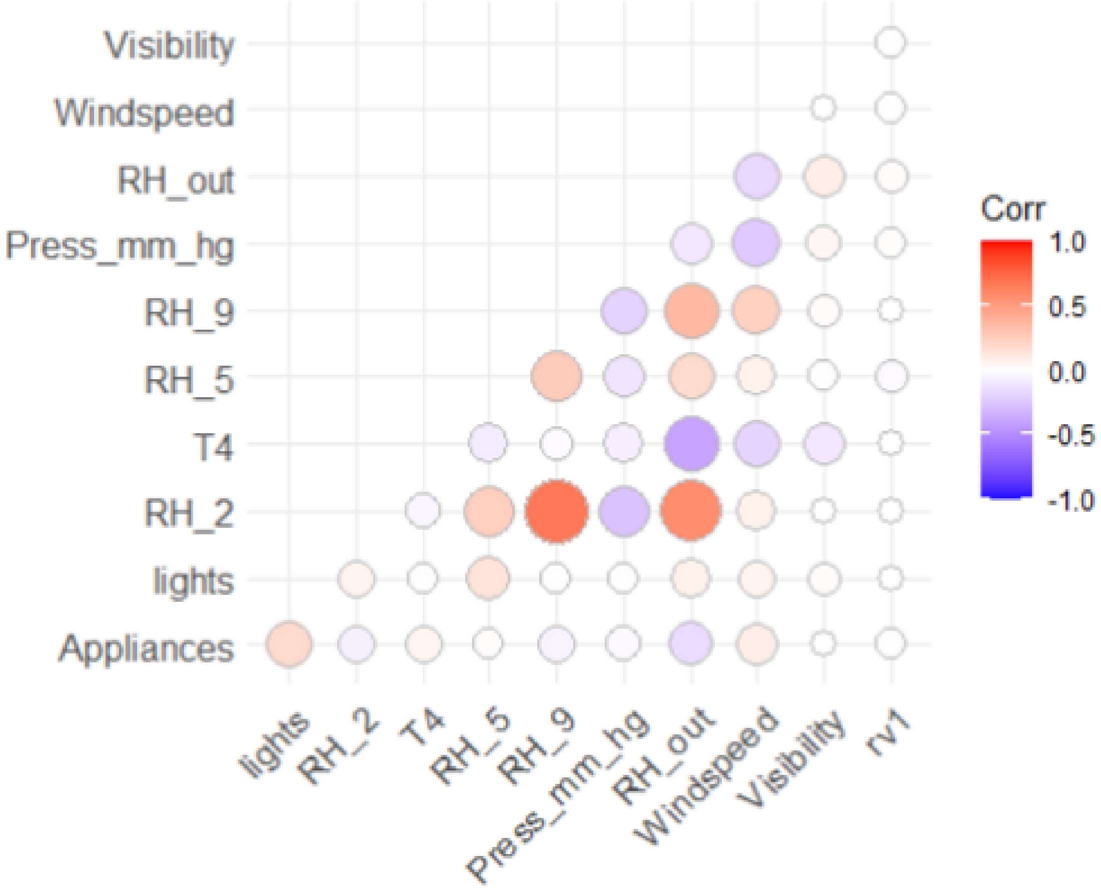


```
# Calculo de Correlação
N2 <- N[2:29]
MatCor <- cor(N2)
ggcorrplot(MatCor,
           type = 'lower',
           p.mat = cor_pmat(N2))
```





```
# Avaliando Multicolinearidade entre Variáveis Preditoras
N3 <- N[2:29]
MatCor1 <- cor(N3)
HighCor <- findCorrelation(MatCor1,
                           cutoff = 0.75)
N3Filtered <- N3[,-HighCor]
MatCor2 <- cor(N3Filtered)
ggcorrplot(MatCor2,
           type = 'lower',
           method = 'circle')
```



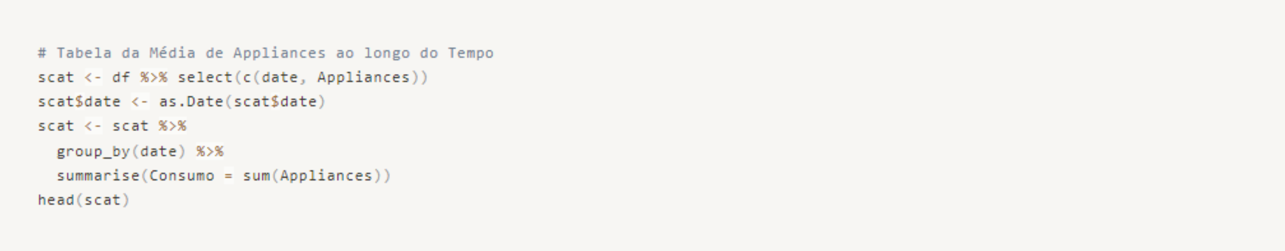
Some points draw our attention:

- The variables rv1 and rv2 have no correlation with any other variable since, as mentioned before, they are almost constant and were artificially inserted into our data by Kaggle.

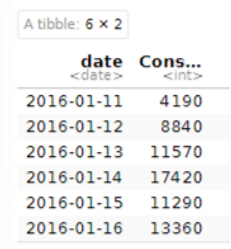
- Our response variable, Appliances, shows some mild positive and negative correlations with some of our variables, indicating that we have the possibility of solving our problem with a linear structured model.
- The Windspeed variable has some low correlations with the Temperature of the Children's Room, Humidity of the Master Bedroom, Local Humidity, and Atmospheric Pressure. It has little to no correlation with other variables.
- The temperature of the Living Room has a strong negative influence on Local Humidity, while the humidity of the Living Room has a strong positive influence from Local Humidity, suggesting that it may be an area with ventilation windows.
- The humidity of the Master Bedroom is positively influenced by Local Humidity.

We notice that the "lights" variable is the only one with a stronger correlation with the response variable.

To conclude our analysis of the numerical variables, let's study how the response variable behaves over time using the "Date" variable.

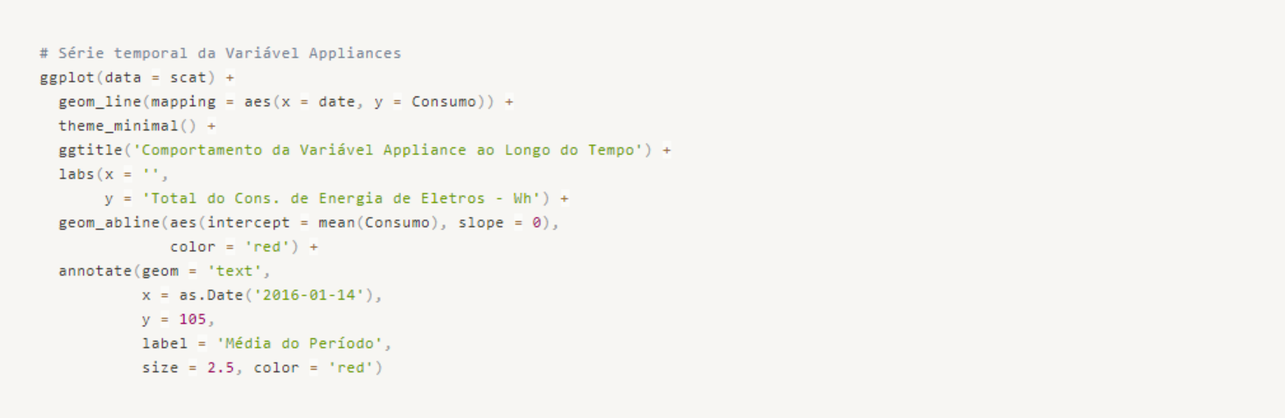


```
# Tabela da Média de Appliances ao longo do Tempo
scat <- df %>% select(c(date, Appliances))
scat$date <- as.Date(scat$date)
scat <- scat %>%
  group_by(date) %>%
  summarise(Consumo = sum(Appliances))
head(scat)
```

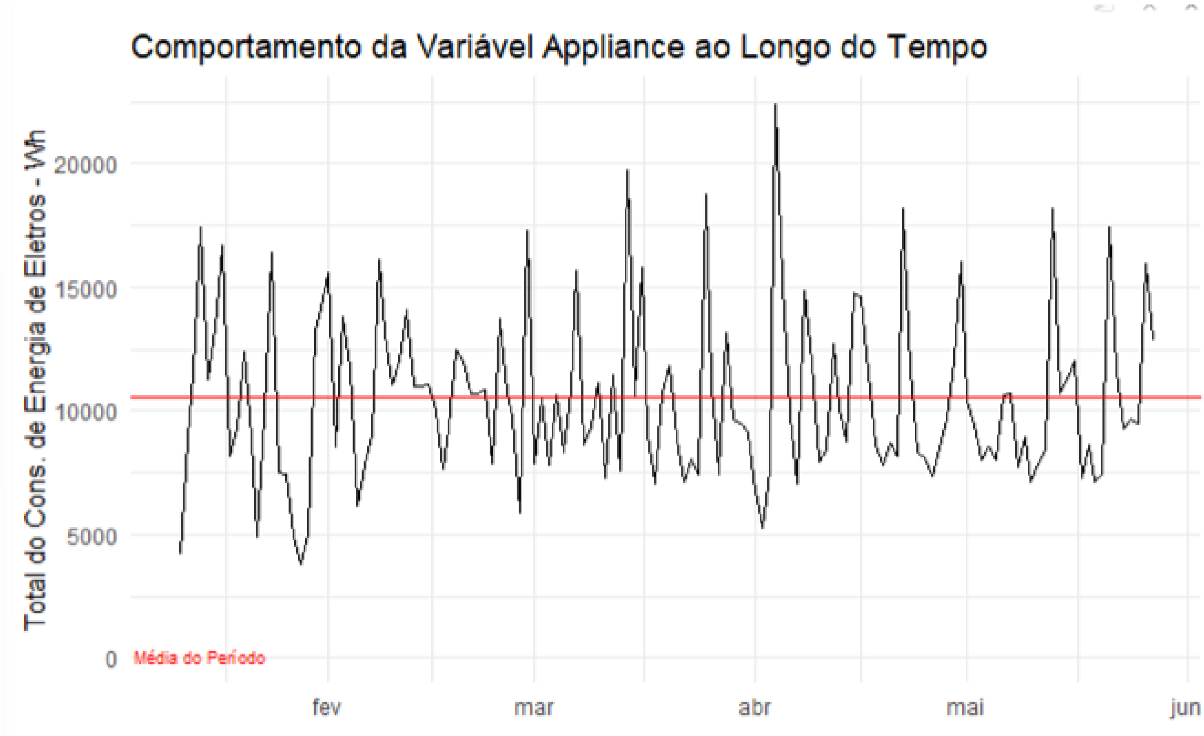


A tibble: 6 × 2

date <date>	Cons... <int>
2016-01-11	4190
2016-01-12	8840
2016-01-13	11570
2016-01-14	17420
2016-01-15	11290
2016-01-16	13360



```
# Série temporal da Variável Appliances
ggplot(data = scat) +
  geom_line(mapping = aes(x = date, y = Consumo)) +
  theme_minimal() +
  ggtitle('Comportamento da Variável Appliance ao Longo do Tempo') +
  labs(x = '',
       y = 'Total do Cons. de Energia de Eletros - Wh') +
  geom_abline(aes(intercept = mean(Consumo), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = 105,
           label = 'Média do Período',
           size = 2.5, color = 'red')
```



We can point out the following:

- The highest peak of consumption occurred in April, reaching twice the average for the period.
- As expected, within the one-month time window, we have significant volatility with minimum values ranging mostly between 75 Wh and 150 Wh. This could be due to some appliances consuming a larger amount of energy but not being used frequently, such as washing machines, dryers, and irons, among others.

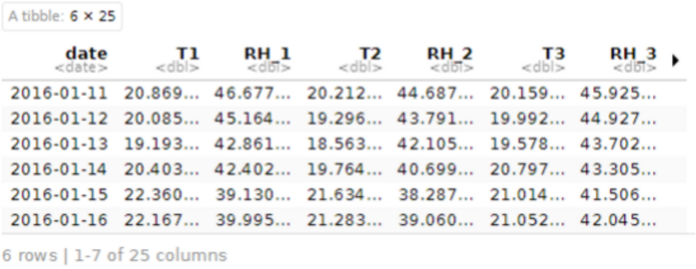
Since the Appliances variable behaved differently in April compared to other months, I believe it is worth conducting a time-based analysis on the other variables to identify similar patterns.

Developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

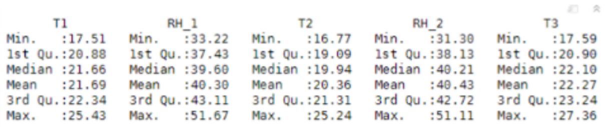
```
# Criando um dataset específico para Séries Temporais
dfst <- df

dfst$date <- as.Date(df$date)

dfst <- dfst %>%
  group_by(date) %>%
  summarise(T1 = mean(T1), RH_1 = mean(RH_1),
            T2 = mean(T2), RH_2 = mean(RH_2),
            T3 = mean(T3), RH_3 = mean(RH_3),
            T4 = mean(T4), RH_4 = mean(RH_4),
            T5 = mean(T5), RH_5 = mean(RH_5),
            T6 = mean(T6), RH_6 = mean(RH_6),
            T7 = mean(T7), RH_7 = mean(RH_7),
            T8 = mean(T8), RH_8 = mean(RH_8),
            T9 = mean(T9), RH_9 = mean(RH_9),
            T_out = mean(T_out), Press_mm_hg = mean(Press_mm_hg),
            RH_out = mean(RH_out), Windspeed = mean(Windspeed),
            Visibility = mean(Visibility), Tdewpoint = mean(Tdewpoint))
```

```
# Visualiza o Novo Dataset de Séries Temporais
head(dfst)
```

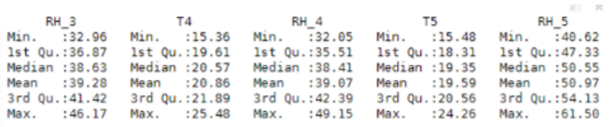


```
A tibble: 6 × 25
      date        T1       RH_1        T2      RH_2        T3      RH_3
    <date>      <dbl>      <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
2016-01-11  20.869...  46.677...  20.212...  44.687...  20.159...  45.925...
2016-01-12  20.085...  45.164...  19.296...  43.791...  19.992...  44.927...
2016-01-13  19.193...  42.861...  18.563...  42.105...  19.578...  43.702...
2016-01-14  20.403...  42.402...  19.764...  40.699...  20.797...  43.305...
2016-01-15  22.360...  39.130...  21.634...  38.287...  21.014...  41.506...
2016-01-16  22.167...  39.995...  21.283...  39.060...  21.052...  42.045...
6 rows | 1-7 of 25 columns
```

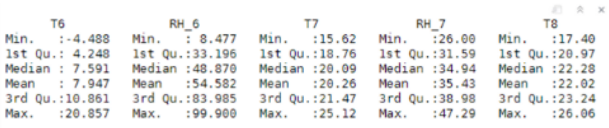
```
# Visualiza as Medidas Centrais
summary(dfst[2:6])
summary(dfst[7:11])
summary(dfst[12:16])
summary(dfst[17:21])
summary(dfst[22:25])
```



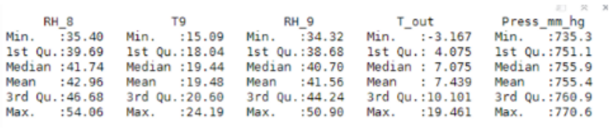
```
       T1              RH_1             T2              RH_2             T3       
 Min.   :17.51   Min.   :33.22   Min.   :16.77   Min.   :31.30   Min.   :17.59  
 1st Qu.:20.88   1st Qu.:37.43   1st Qu.:19.09   1st Qu.:38.13   1st Qu.:20.90  
 Median :21.66   Median :39.60   Median :19.94   Median :40.21   Median :22.10  
 Mean   :21.69   Mean   :40.30   Mean   :20.36   Mean   :40.43   Mean   :22.27  
 3rd Qu.:22.34   3rd Qu.:43.11   3rd Qu.:21.31   3rd Qu.:42.72   3rd Qu.:23.24  
 Max.   :25.43   Max.   :51.67   Max.   :25.24   Max.   :51.11   Max.   :27.36  
```



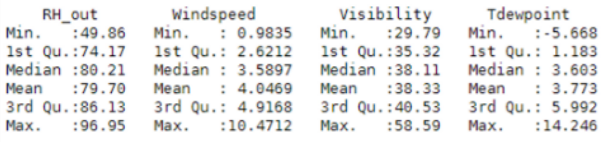
```
      RH_3             T4              RH_4             T5              RH_5      
 Min.   :32.96   Min.   :15.36   Min.   :32.05   Min.   :15.48   Min.   :40.62  
 1st Qu.:36.87   1st Qu.:19.61   1st Qu.:35.51   1st Qu.:18.31   1st Qu.:47.33  
 Median :38.63   Median :20.57   Median :38.41   Median :19.35   Median :50.55  
 Mean   :39.28   Mean   :20.86   Mean   :39.07   Mean   :19.59   Mean   :50.97  
 3rd Qu.:41.42   3rd Qu.:21.89   3rd Qu.:42.39   3rd Qu.:20.56   3rd Qu.:54.13  
 Max.   :46.17   Max.   :25.48   Max.   :49.15   Max.   :24.26   Max.   :61.50  
```



```
       T6              RH_6              T7             RH_7             T8       
 Min.   :-4.488   Min.   : 8.477   Min.   :15.62   Min.   :26.00   Min.   :17.40  
 1st Qu.: 4.248   1st Qu.:33.196   1st Qu.:18.76   1st Qu.:31.59   1st Qu.:20.97  
 Median : 7.591   Median :48.870   Median :20.09   Median :34.94   Median :22.28  
 Mean   : 7.947   Mean   :54.582   Mean   :20.26   Mean   :35.43   Mean   :22.02  
 3rd Qu.:10.861   3rd Qu.:83.985   3rd Qu.:21.47   3rd Qu.:38.98   3rd Qu.:23.24  
 Max.   :20.857   Max.   :99.900   Max.   :25.12   Max.   :47.29   Max.   :26.06  
```



```
      RH_8             T9             RH_9            T_out          Press_mm_hg   
 Min.   :35.40   Min.   :15.09   Min.   :34.32   Min.   :-3.167   Min.   :735.3  
 1st Qu.:39.69   1st Qu.:18.04   1st Qu.:38.68   1st Qu.: 4.075   1st Qu.:751.1  
 Median :41.74   Median :19.44   Median :40.70   Median : 7.075   Median :755.9  
 Mean   :42.96   Mean   :19.48   Mean   :41.56   Mean   : 7.439   Mean   :755.4  
 3rd Qu.:46.68   3rd Qu.:20.60   3rd Qu.:44.24   3rd Qu.:10.101   3rd Qu.:760.9  
 Max.   :54.06   Max.   :24.19   Max.   :50.90   Max.   :19.461   Max.   :770.6  
```



```
     RH_out         Windspeed          Visibility       Tdewpoint     
 Min.   :49.86   Min.   : 0.9835   Min.   :29.79   Min.   :-5.668  
 1st Qu.:74.17   1st Qu.: 2.6212   1st Qu.:35.32   1st Qu.: 1.183  
 Median :80.21   Median : 3.5897   Median :38.11   Median : 3.603  
 Mean   :79.70   Mean   : 4.0469   Mean   :38.33   Mean   : 3.773  
 3rd Qu.:86.13   3rd Qu.: 4.9168   3rd Qu.:40.53   3rd Qu.: 5.992  
 Max.   :96.95   Max.   :10.4712   Max.   :58.59   Max.   :14.246  
```

```
# Criando os Elementos Gráficos das Séries Temporais
T1 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T1)) +
  theme_minimal() +
  labs(y = '', x = '', subtitle = 'Variável T1') +
  geom_abline(aes(intercept = mean(T1), slope = 0), color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T1) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')
```

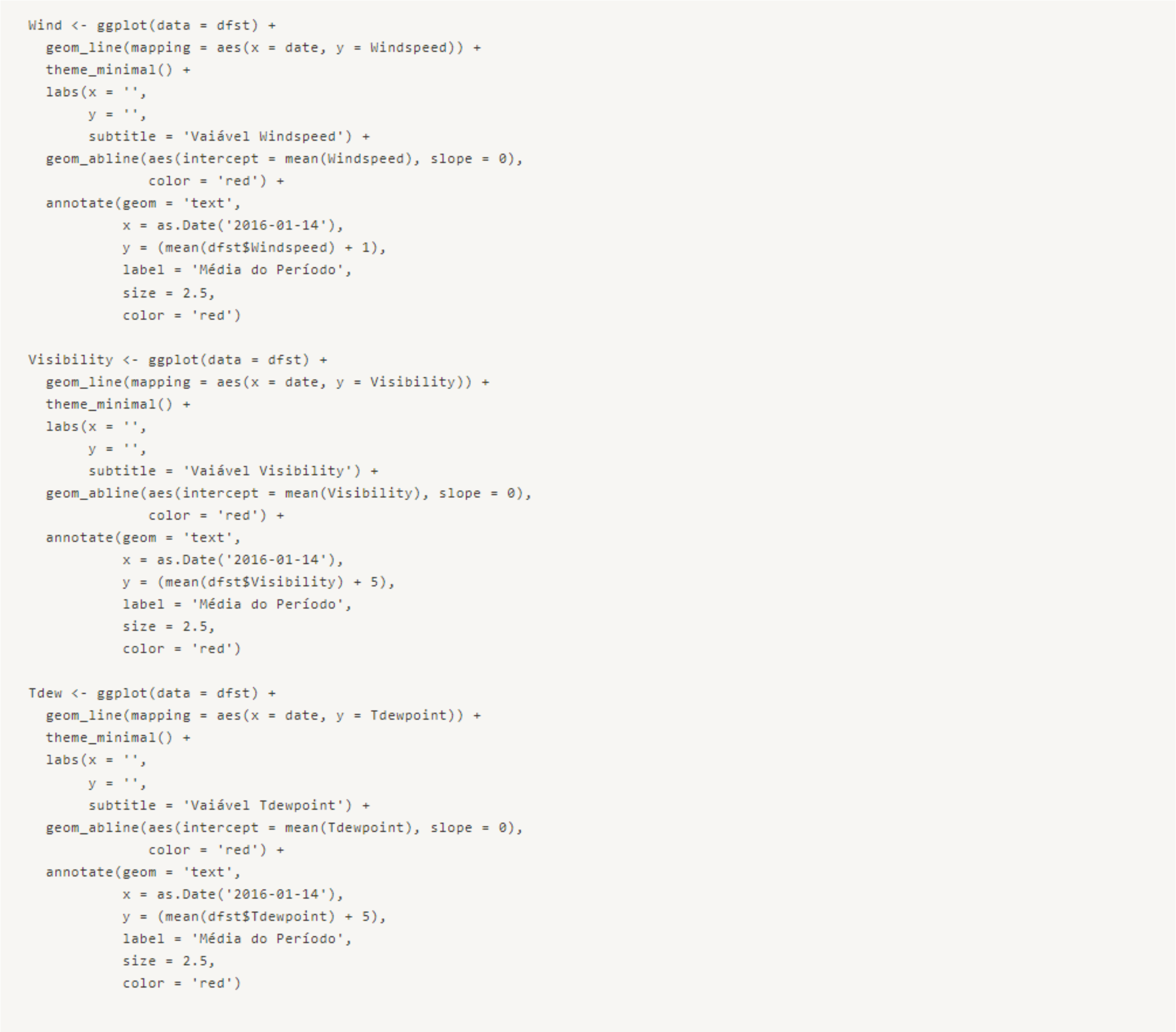
```
RH1 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_1)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_1') +
  geom_abline(aes(intercept = mean(RH_1), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_1) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

T2 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T2)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T2') +
  geom_abline(aes(intercept = mean(T2), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T2) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

RH2 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_2)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_2') +
  geom_abline(aes(intercept = mean(RH_2), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_2) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

T3 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T3)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T3') +
  geom_abline(aes(intercept = mean(T3), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T3) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')
```

```
RH3 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_3)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_3') +
  geom_abline(aes(intercept = mean(RH_3), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_3) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

T4 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T4)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T4') +
  geom_abline(aes(intercept = mean(T4), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T4) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

RH4 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_4)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_4') +
  geom_abline(aes(intercept = mean(RH_4), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_4) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

T5 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T5)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T5') +
  geom_abline(aes(intercept = mean(T5), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T5) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')
```

```
RH5 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_5)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_5') +
  geom_abline(aes(intercept = mean(RH_5), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_5) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

T6 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T6)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T6') +
  geom_abline(aes(intercept = mean(T6), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T6) + 3),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

RH6 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_6)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_6') +
  geom_abline(aes(intercept = mean(RH_6), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_6) + 6),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

T7 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T7)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T7') +
  geom_abline(aes(intercept = mean(T7), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T7) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')
```

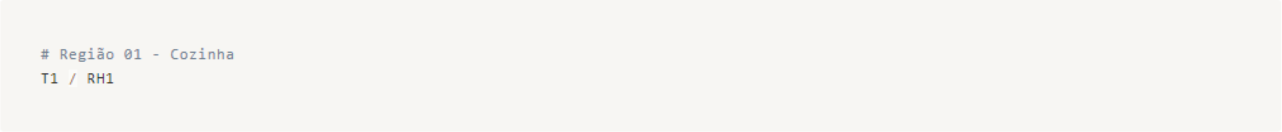
```r
RH7 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_7)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_7') +
  geom_abline(aes(intercept = mean(RH_7), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_7) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

T8 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T8)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T8') +
  geom_abline(aes(intercept = mean(T8), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T8) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

RH8 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_8)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_8') +
  geom_abline(aes(intercept = mean(RH_8), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_8) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

T9 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T9)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T9') +
  geom_abline(aes(intercept = mean(T9), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T9) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')
```

```
RH9 <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_9)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_9') +
  geom_abline(aes(intercept = mean(RH_9), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_9) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

Tout <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = T_out)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável T_out') +
  geom_abline(aes(intercept = mean(T_out), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$T_out) + 3),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

RHout <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = RH_out)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável RH_out') +
  geom_abline(aes(intercept = mean(RH_out), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$RH_out) + 3),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

Press <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = Press_mm_hg)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável Press_mm_hg') +
  geom_abline(aes(intercept = mean(Press_mm_hg), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$Press_mm_hg) + 10),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')
```

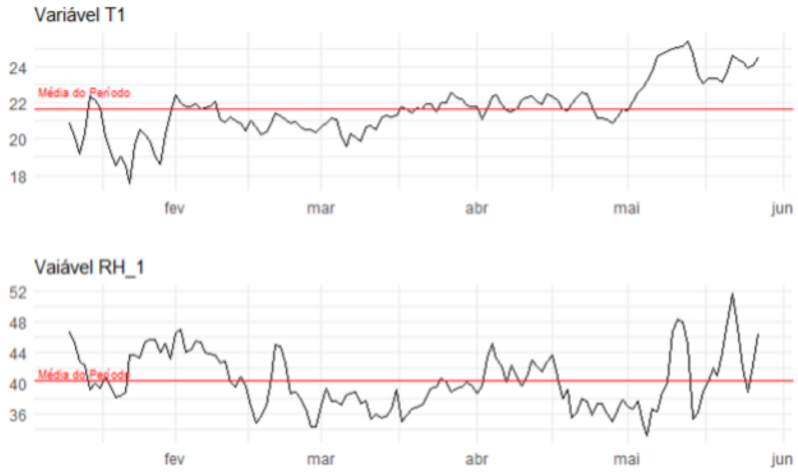


```
Wind <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = Windspeed)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável Windspeed') +
  geom_abline(aes(intercept = mean(Windspeed), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$Windspeed) + 1),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

Visibility <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = Visibility)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável Visibility') +
  geom_abline(aes(intercept = mean(Visibility), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$Visibility) + 5),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')

Tdew <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = Tdewpoint)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável Tdewpoint') +
  geom_abline(aes(intercept = mean(Tdewpoint), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           x = as.Date('2016-01-14'),
           y = (mean(dfst$Tdewpoint) + 5),
           label = 'Média do Período',
           size = 2.5,
           color = 'red')
```

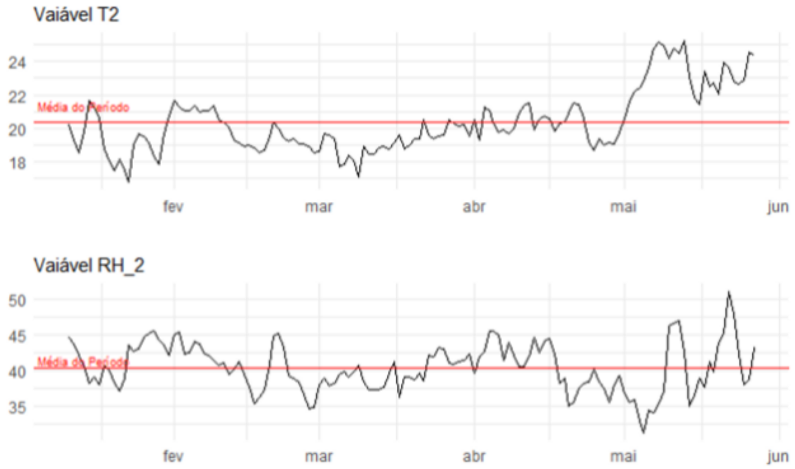
Let's visualize the time series for each region of the house and assess if there is any relationship with time in each case.



```
# Região 01 - Cozinha
T1 / RH1
```



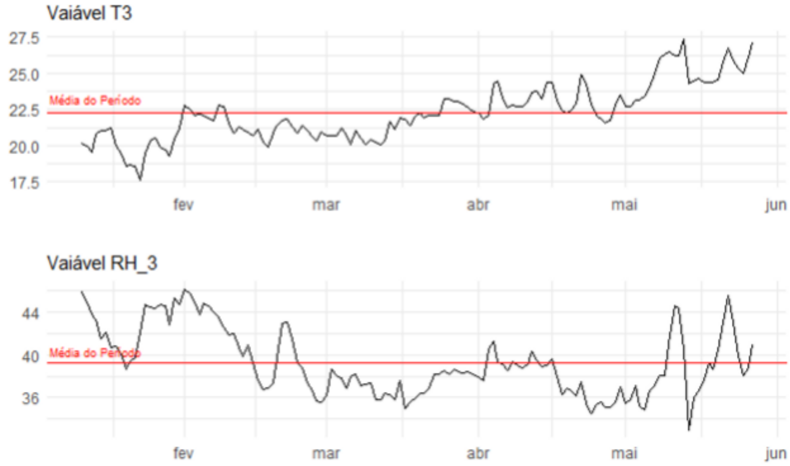
Humidity exhibits higher volatility compared to Temperature, which is plausible considering the rainy season. We can observe a period from May to June with higher and more stable temperatures, but with more volatile humidity ranging between 36% and 52%. Both variables show an upward trend from May to June.



```
# Região 02 - Sala de Estar
T2 / RH2
```

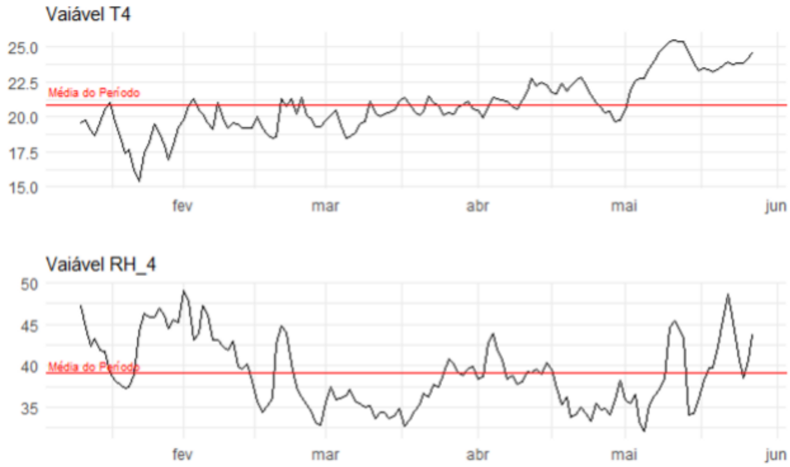
We have a very similar behavior in region 01, with May and June showing an upward trend as well.

```
# Região 03 - Lavanderia
T3 / RH3
```



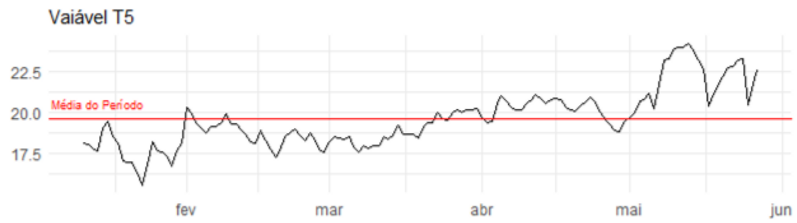
In the Laundry room, we have a slightly different scenario. The temperature shows an upward trend throughout the period, while humidity shows a downward trend. Several factors can explain this behavior, such as exposure to sunlight, morning or afternoon sun, frequent use of appliances that generate heat like washing machines, dryers, and other common equipment. Among the regions analyzed so far, it has the smallest range of humidity values, with lower minimums and maximums.

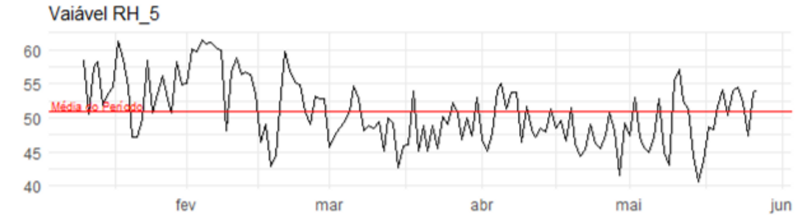
```
# Região 04 - Escritório
T4 / RH4
```



For the Office, we observe a similar behavior to the other regions, with a general downward trend in humidity and an increase in temperature. The May-June period also exhibits a similar pattern to the other regions.

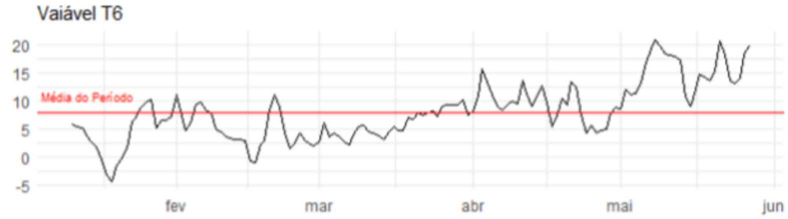
```
# Região 05 - Banheiro
T5 / RH5
```

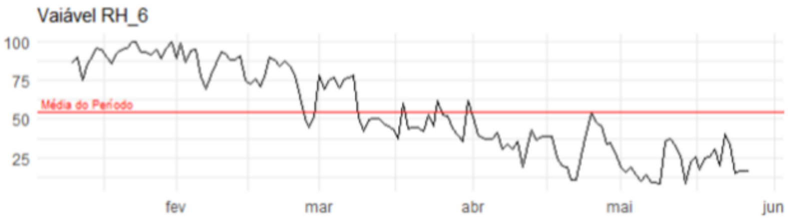




In the Bathroom, we observe larger fluctuations in humidity, which is not uncommon considering its use for activities like bathing. Other than that, we see a behavior similar to the other areas we have studied.

```
# Região 06 - Área Externa do Segundo Andar
T6 / RH6
```

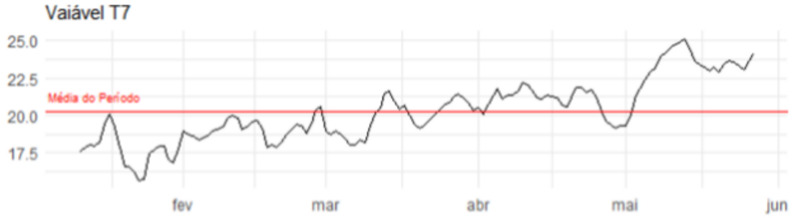


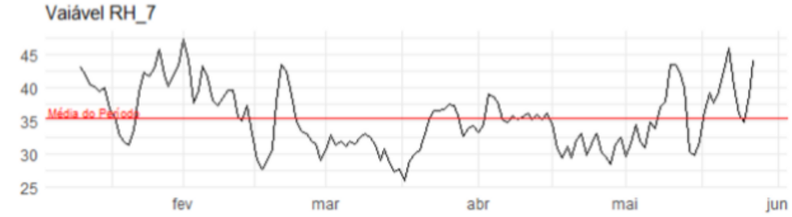


For the Outside area, we observe a different behavior regarding humidity during the May and June period. There is a significant downward trend, while the temperature shows an upward trend similar to the other areas.

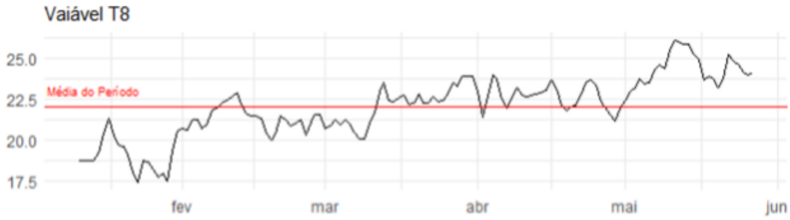
The humidity values in this area are generally higher compared to the rest of the house, while the temperatures are much lower.

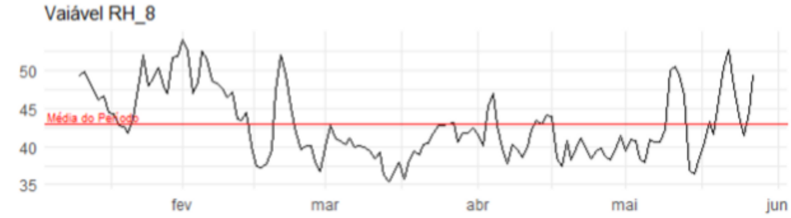
```
# Região 07 - Sala de Passar Roupa / Engomar
T7 / RH7
```





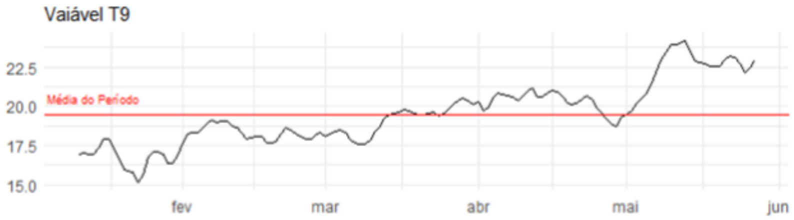
We can observe a rising trend in both humidity and temperature during the May-June period. In the previous periods, the graph shows a downward trend for humidity and an upward trend for temperature.

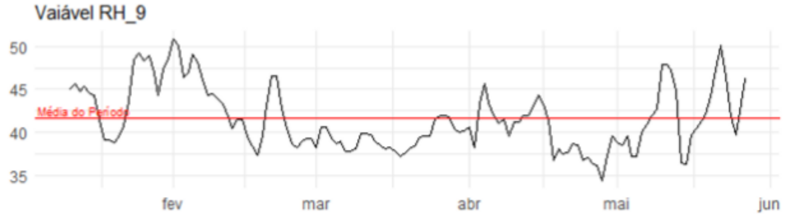




Similar behavior can be observed in the other areas of the house, except for the Outside area (Region 6).

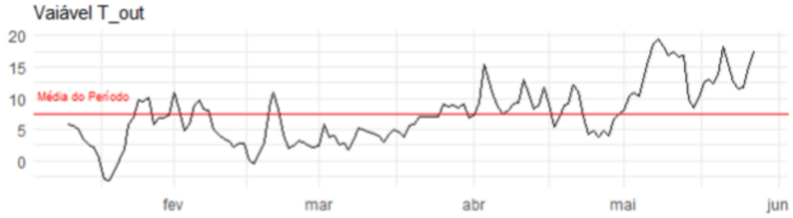
```
# Região 09 - Quarto Casal
T9 / RH9
```

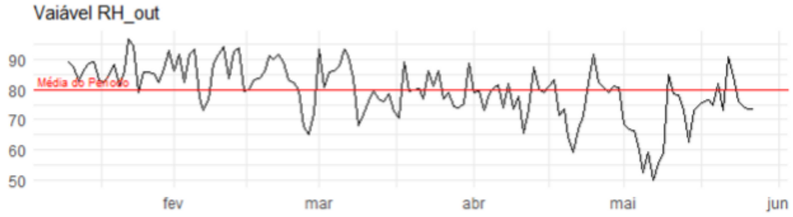




Once again, we see a similar pattern as in the other areas. However, despite the overall upward trend in temperature, the fluctuations in the Master Bedroom are less pronounced compared to the other areas.

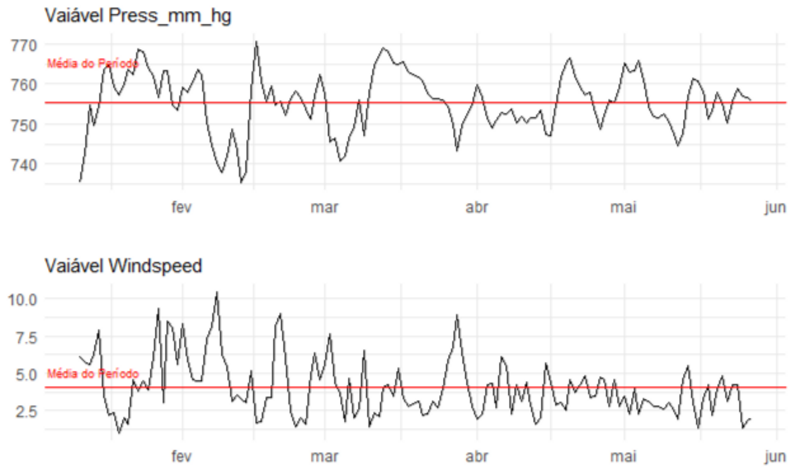
```
# Temperatura e Umidade no Local na Estação de Medição
Tout / RHout
```





We can observe that the temperatures in the station are much lower than in the house, probably due to heating or the fact that it is a more enclosed area than the outside. On the other hand, the humidity reaches much higher values, with an average of 80%.

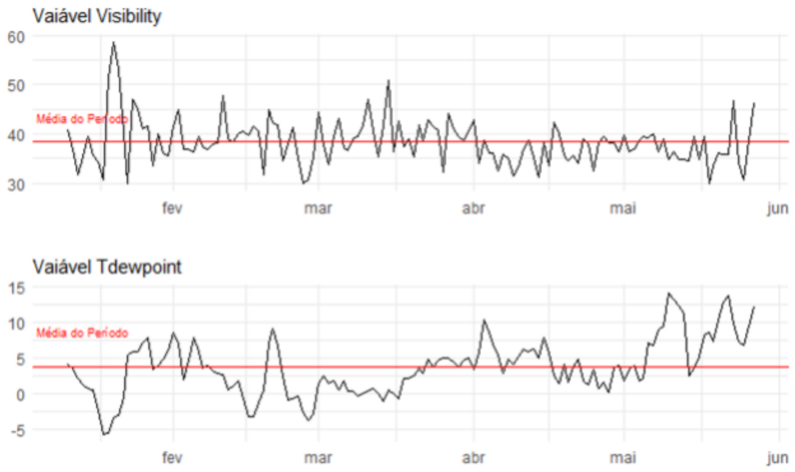
```
# Pressão Atmof e Velocidade do Vento na Estação
Press / Wind
```



Despite the volatility of the Pressure, we don't have a clear upward or downward trend. The variation is around ± 15 mm Hg, which can be considered relatively constant during the period.

Regarding the Windspeed, we observe a downward trend, which may support some hypotheses regarding the increase in energy consumption by household appliances.

```
# Visibilidade e Ponto de Orvalho na Estação
Visibility / Tdew
```



Visibility fluctuates around the average by approximately ± 10 km throughout the period, without any clear upward or downward trend.

The dew point rises as the ambient temperature increases.

Finally, a theory that we can develop, although it would require more information to corroborate, is that from April onwards, the temperature graphs become more positive in relation to their averages. Appliances such as air conditioners, fans, and electric grills are more frequently used. People tend to go out more often, wear more clothes, requiring a higher frequency of using washing machines, dryers, and irons. They also spend more time on personal grooming, increasing the use of hairdryers, straighteners, and so on.

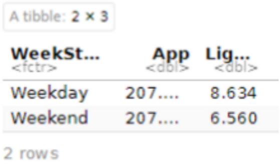
Furthermore, by analyzing people's behavior and researching the Chievre region, we understand that the period from April to June marks the beginning of more pleasant temperatures, which is eagerly awaited by the residents. Consequently, this supports the assumptions we made above. Therefore, the change in climate and season in Chievre can help explain why there is a peak in energy consumption by household appliances in April and a subsequent upward trend.

Categorical Variables

For the categorical variables, we have smaller time windows, such as workdays and weekends, as well as the seven days of the week. We could analyze the behavior of each variable within these time windows, but for our model and understanding of the data, it is not necessary. However, let's study how the variables behave on average during these days.

```
# Calculo de Médias das Variáveis por Tipo de WorkStatus
dfws <- df[2:31] %>%
  group_by(WeekStatus) %>%
  summarise(App = sum(Appliances)/1000, Lights = sum(lights)/1000)

dfws[1,2:3] = (dfws[1,2:3]) / 5
dfws[2, 2:3] = (dfws[2, 2:3]) / 2
```

```
# Visualizando a Tabela
dfws
```

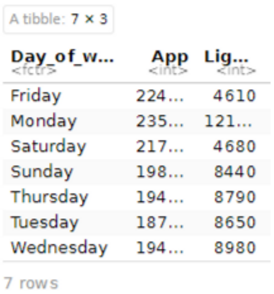


```
A tibble: 2 × 3
WeekSt...     App    Lig...
<fctr>      <dbl>    <dbl>
Weekday    207....   8.634
Weekend    207....   6.560
2 rows
```

We can clearly see that there is no significantly higher energy consumption in household appliances between weekdays and weekends. However, for the lights, the consumption during the week is higher than on weekends. This is likely due to the cultural habit of leaving lights on when people are away from home.

```r
# Calculo de Médias das Variáveis por Tipo de WorkStatus
dfdow <- df[2:31] %>%
  group_by(Day_of_week) %>%
  summarise(App = sum(Appliances), Lights = sum(lights))
```

```r
# Visualizando a Tabela
dfdow
```



A tibble: 7 × 3

Day_of_w... <fctr>	App <int>	Lig... <int>
Friday	224...	4610
Monday	235...	121...
Saturday	217...	4680
Sunday	198...	8440
Thursday	194...	8790
Tuesday	187...	8650
Wednesday	194...	8980

7 rows

We can observe a pattern in the usage of household appliances, as the highest energy consumption occurs from Friday to Monday. On the other hand, the electricity consumption for lighting is clearly higher on most weekdays, even doubling in some cases.

Interestingly, the energy consumption for lighting is significantly lower on Fridays and Saturdays, about half compared to other days. There is also a peak in consumption on Mondays.

Preprocessing

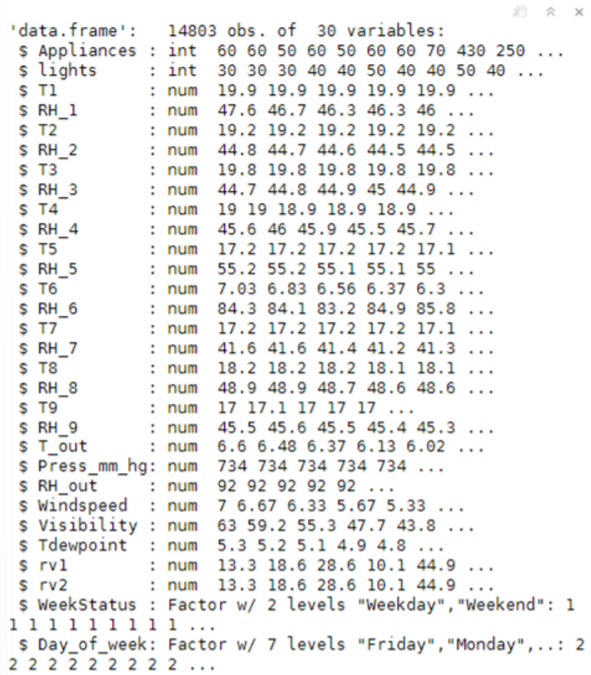
To prepare our data, we will exclude the time window from the 'date' variable since our objective is not to perform time series predictive modeling. However, we will keep the variables 'WeekStatus' and 'Day_of_Week' as they represent a static window that may have an influence on the target variable. As these variables are categorical, it is important to encode them as factors.

We will also drop the 'lights' variable as it represents energy consumption and is not related to the target variable 'Appliances'.

Furthermore, we need to reduce the dimensionality and apply data standardization to equalize the scales and prevent our model from becoming biased and non-generalizable.

```
# Criando um novo Dataset
dfo <- df[2:31]
```

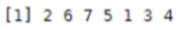
```
# Conferindo os Tipos de Variáveis
str(dfo)
```



```
'data.frame':	14803 obs. of  30 variables:
 $ Appliances : int  60 60 50 60 50 60 60 70 430 250 ...
 $ lights     : int  30 30 30 40 40 50 40 40 50 40 ...
 $ T1         : num  19.9 19.9 19.9 19.9 19.9 ...
 $ RH_1       : num  47.6 46.7 46.3 46.3 46 ...
 $ T2         : num  19.2 19.2 19.2 19.2 19.2 ...
 $ RH_2       : num  44.8 44.7 44.6 44.5 44.5 ...
 $ T3         : num  19.8 19.8 19.8 19.8 19.8 ...
 $ RH_3       : num  44.7 44.8 44.9 45 44.9 ...
 $ T4         : num  19 19 18.9 18.9 18.9 ...
 $ RH_4       : num  45.6 46 45.9 45.5 45.7 ...
 $ T5         : num  17.2 17.2 17.2 17.2 17.1 ...
 $ RH_5       : num  55.2 55.2 55.1 55.1 55 ...
 $ T6         : num  7.03 6.83 6.56 6.37 6.3 ...
 $ RH_6       : num  84.3 84.1 83.2 84.9 85.8 ...
 $ T7         : num  17.2 17.2 17.2 17.2 17.1 ...
 $ RH_7       : num  41.6 41.6 41.4 41.2 41.3 ...
 $ T8         : num  18.2 18.2 18.2 18.1 18.1 ...
 $ RH_8       : num  48.9 48.9 48.7 48.6 48.6 ...
 $ T9         : num  17 17.1 17 17 17 ...
 $ RH_9       : num  45.5 45.6 45.5 45.4 45.3 ...
 $ T_out      : num  6.6 6.48 6.37 6.13 6.02 ...
 $ Press_mm_hg: num  734 734 734 734 734 ...
 $ RH_out     : num  92 92 92 92 92 ...
 $ Windspeed  : num  7 6.67 6.33 5.67 5.33 ...
 $ Visibility : num  63 59.2 55.3 47.7 43.8 ...
 $ Tdewpoint  : num  5.3 5.2 5.1 4.9 4.8 ...
 $ rv1        : num  13.3 18.6 28.6 10.1 44.9 ...
 $ rv2        : num  13.3 18.6 28.6 10.1 44.9 ...
 $ WeekStatus : Factor w/ 2 levels "Weekday","Weekend": 1
1 1 1 1 1 1 1 1 1 ...
 $ Day_of_week: Factor w/ 7 levels "Friday","Monday",..: 2
2 2 2 2 2 2 2 2 2 ...
```

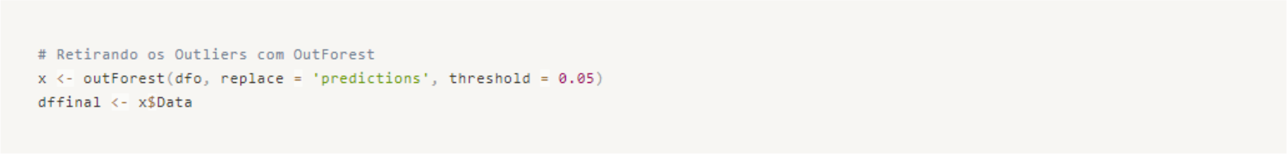
As we can see, the categorical variables are already in factor format, which makes our work easier for label encoding.

```
# Label Encoding das Variáveis Categóricas
dfo$WeekStatus <- as.numeric(dfo$WeekStatus)
dfo$Day_of_week <- as.numeric(dfo$Day_of_week)
unique(dfo$Day_of_week)
```

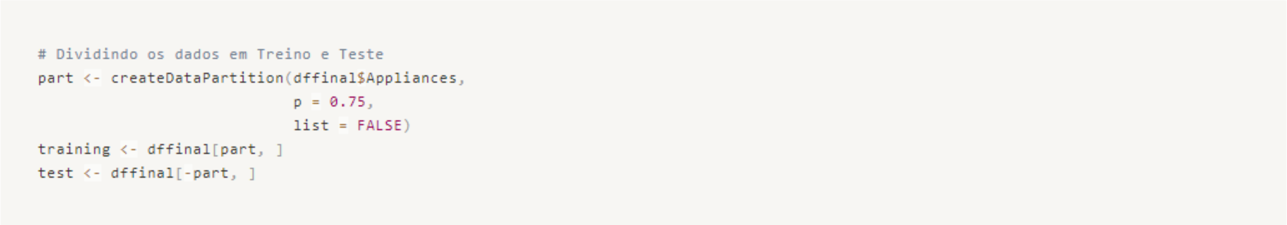


```
[1] 2 6 7 5 1 3 4
```

Please note that the week starts with Sunday (Sun), represented by the number 1, followed by the other days of the week in sequence (Mon - 2, Tue - 3, Wed - 4, Thu - 5, Fri - 6, Sat - 7).



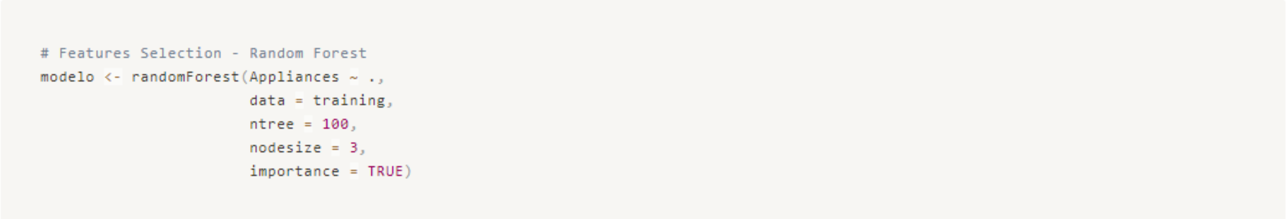
```
# Retirando os Outliers com OutForest
x <- outForest(dfo, replace = 'predictions', threshold = 0.05)
dffinal <- x$Data
```



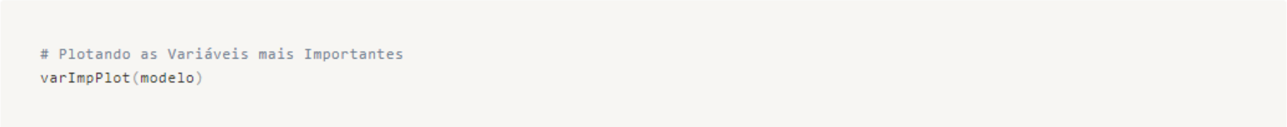
```
# Dividindo os dados em Treino e Teste
part <- createDataPartition(dffinal$Appliances,
                            p = 0.75,
                            list = FALSE)
training <- dffinal[part, ]
test <- dffinal[-part, ]
```

Data divided!

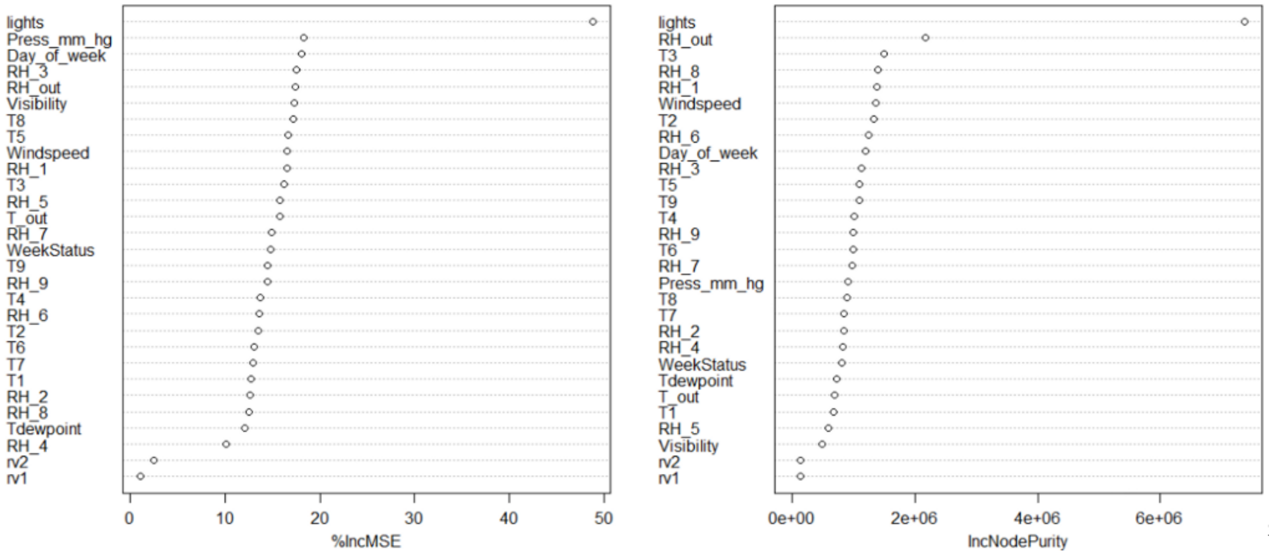
Now we will work on Variable Selection using the Random Forest model. This will help us reduce the dimensionality and complexity of the final model.



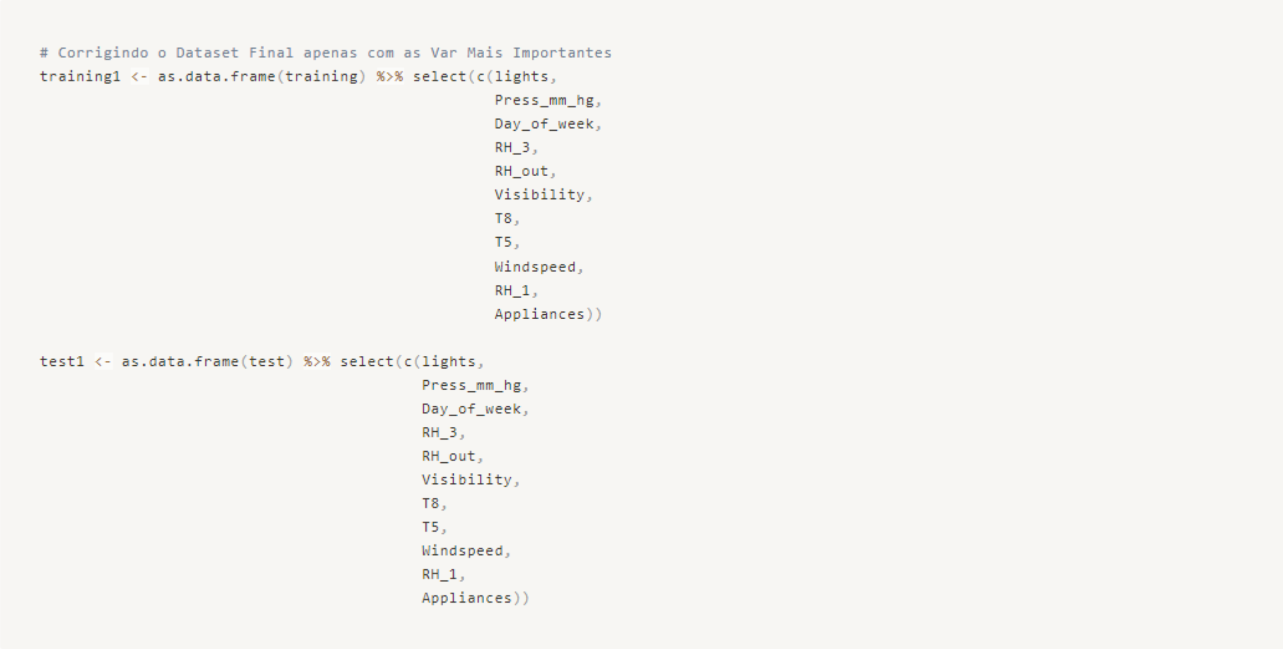
```
# Features Selection - Random Forest
modelo <- randomForest(Appliances ~ .,
                       data = training,
                       ntree = 100,
                       nodesize = 3,
                       importance = TRUE)
```



```
# Plotando as Variáveis mais Importantes
varImpPlot(modelo)
```



In order to reduce the complexity of our model, we will work with only the top 10 most important variables. This way, our model becomes less complex and more generalizable.

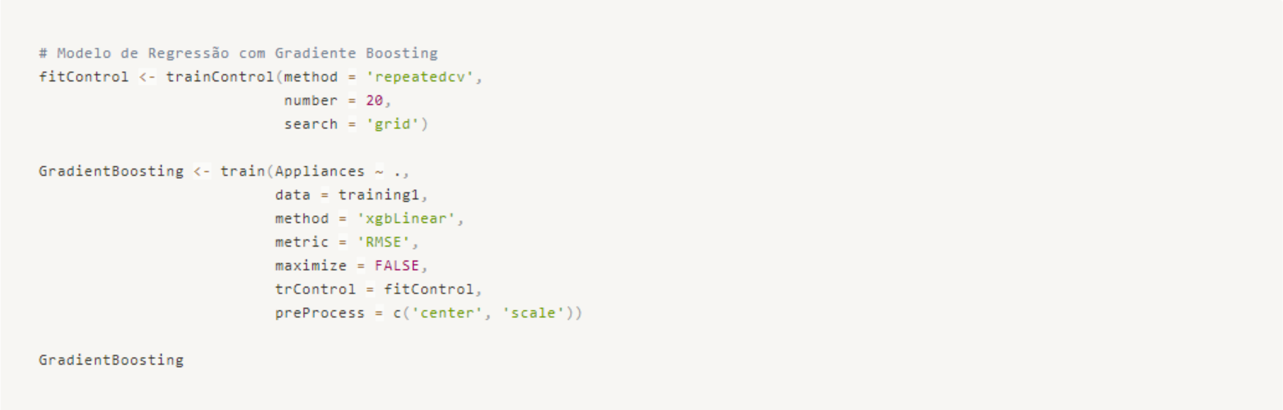


```
# Corrigindo o Dataset Final apenas com as Var Mais Importantes
training1 <- as.data.frame(training) %>% select(c(lights,
                                                  Press_mm_hg,
                                                  Day_of_week,
                                                  RH_3,
                                                  RH_out,
                                                  Visibility,
                                                  T8,
                                                  T5,
                                                  Windspeed,
                                                  RH_1,
                                                  Appliances))

test1 <- as.data.frame(test) %>% select(c(lights,
                                          Press_mm_hg,
                                          Day_of_week,
                                          RH_3,
                                          RH_out,
                                          Visibility,
                                          T8,
                                          T5,
                                          Windspeed,
                                          RH_1,
                                          Appliances))
```

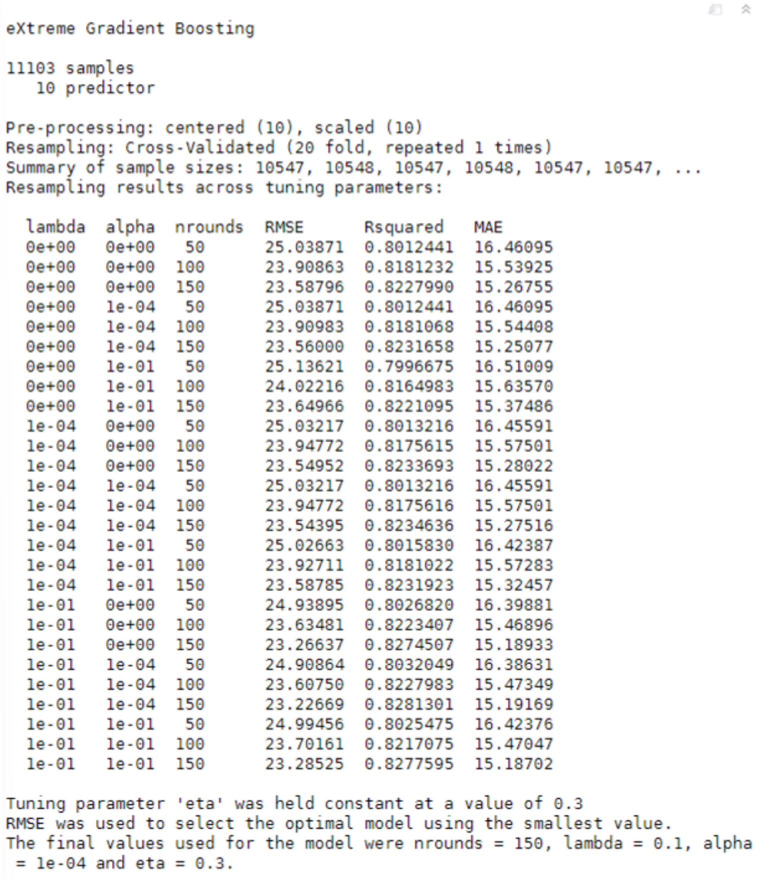
Predictive Modeling

We will use different algorithms to create several versions of models and analyze their performance metrics. Note that we are applying standardization to the predictor variables using the "preProcess" argument within the train function.

Model 00 – Gradient Boosting

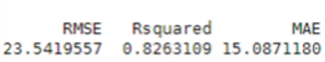


```
# Modelo de Regressão com Gradiente Boosting
fitControl <- trainControl(method = 'repeatedcv',
                           number = 20,
                           search = 'grid')

GradientBoosting <- train(Appliances ~ .,
                          data = training1,
                          method = 'xgbLinear',
                          metric = 'RMSE',
                          maximize = FALSE,
                          trControl = fitControl,
                          preProcess = c('center', 'scale'))

GradientBoosting
```



```
eXtreme Gradient Boosting

11103 samples
   10 predictor

Pre-processing: centered (10), scaled (10)
Resampling: Cross-Validated (20 fold, repeated 1 times)
Summary of sample sizes: 10547, 10548, 10547, 10548, 10547, 10547, ...
Resampling results across tuning parameters:

  lambda  alpha  nrounds  RMSE      Rsquared   MAE
  0e+00   0e+00   50      25.03871  0.8012441  16.46095
  0e+00   0e+00  100      23.90863  0.8181232  15.53925
  0e+00   0e+00  150      23.58796  0.8227990  15.26755
  0e+00   1e-04   50      25.03871  0.8012441  16.46095
  0e+00   1e-04  100      23.90983  0.8181068  15.54408
  0e+00   1e-04  150      23.56000  0.8231658  15.25077
  0e+00   1e-01   50      25.13621  0.7996675  16.51009
  0e+00   1e-01  100      24.02216  0.8164983  15.63570
  0e+00   1e-01  150      23.64966  0.8221095  15.37486
  1e-04   0e+00   50      25.03217  0.8013216  16.45591
  1e-04   0e+00  100      23.94772  0.8175615  15.57501
  1e-04   0e+00  150      23.54952  0.8233693  15.28022
  1e-04   1e-04   50      25.03217  0.8013216  16.45591
  1e-04   1e-04  100      23.94772  0.8175616  15.57501
  1e-04   1e-04  150      23.54395  0.8234636  15.27516
  1e-04   1e-01   50      25.02663  0.8015830  16.42387
  1e-04   1e-01  100      23.92711  0.8181022  15.57283
  1e-04   1e-01  150      23.58785  0.8231923  15.32457
  1e-01   0e+00   50      24.93895  0.8026820  16.39881
  1e-01   0e+00  100      23.63481  0.8223407  15.46896
  1e-01   0e+00  150      23.26637  0.8274507  15.18933
  1e-01   1e-04   50      24.90864  0.8032049  16.38631
  1e-01   1e-04  100      23.60750  0.8227983  15.47349
  1e-01   1e-04  150      23.22669  0.8281301  15.19169
  1e-01   1e-01   50      24.99456  0.8025475  16.42376
  1e-01   1e-01  100      23.70161  0.8217075  15.47047
  1e-01   1e-01  150      23.28525  0.8277595  15.18702

Tuning parameter 'eta' was held constant at a value of 0.3
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were nrounds = 150, lambda = 0.1, alpha
 = 1e-04 and eta = 0.3.
```

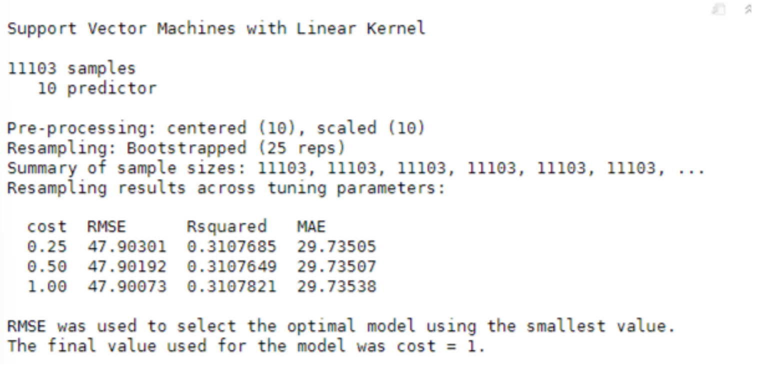
```
# Realizando as previsões e Calculando as Métricas
predGradientB <- predict(GradientBoosting, test1[,-11])
postResample(pred = predGradientB, obs = test1$Appliances)
```



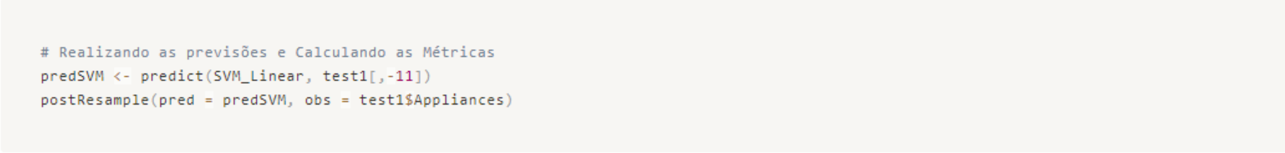
```
      RMSE   Rsquared        MAE 
23.5419557  0.8263109 15.0871180 
```

Model 01 – Support Vector Machine (SVM)

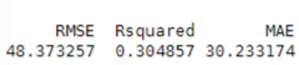
```
# Modelo de Regressão com Support Vector Machine
fitControl <- trainControl(method = 'repeatedcv',
                           number = 20,
                           search = 'grid')

SVM_Linear <- train(Appliances ~ .,
                    data = training1,
                    method = 'svmLinear2',
                    metric = 'RMSE',
                    maximize = FALSE,
                    preProcess = c('center', 'scale'))

SVM_Linear
```



```
Support Vector Machines with Linear Kernel

11103 samples
   10 predictor

Pre-processing: centered (10), scaled (10)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 11103, 11103, 11103, 11103, 11103, 11103, ...
Resampling results across tuning parameters:

  cost  RMSE      Rsquared   MAE
  0.25  47.90301  0.3107685  29.73505
  0.50  47.90192  0.3107649  29.73507
  1.00  47.90073  0.3107821  29.73538

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was cost = 1.
```

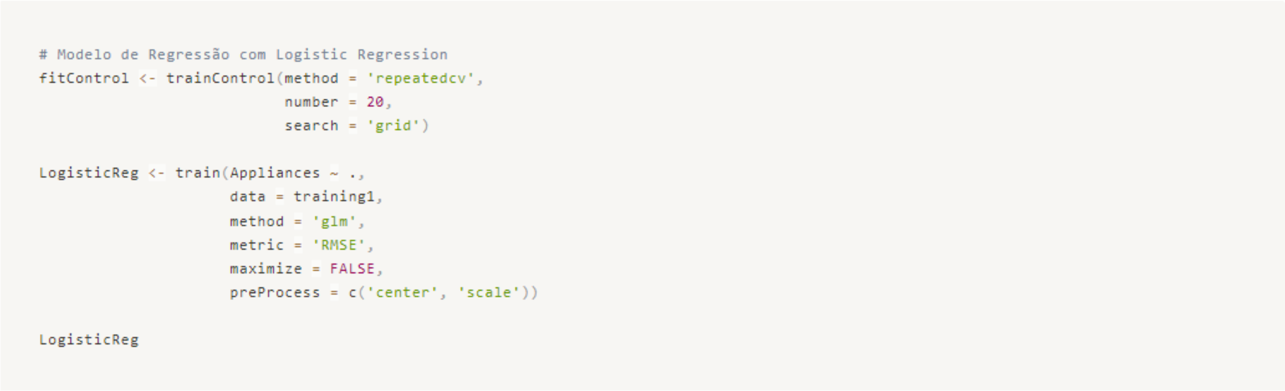


```
# Realizando as previsões e Calculando as Métricas
predSVM <- predict(SVM_Linear, test1[,-11])
postResample(pred = predSVM, obs = test1$Appliances)
```

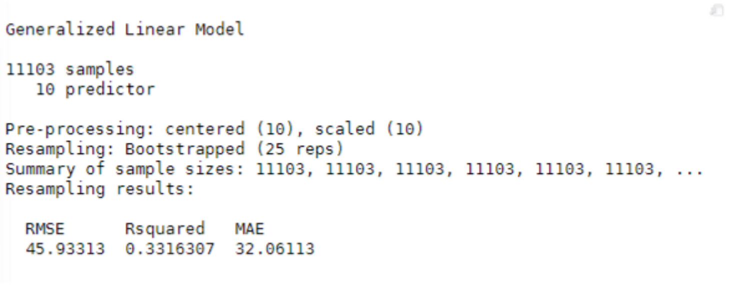


```
     RMSE  Rsquared       MAE
48.373257  0.304857 30.233174
```

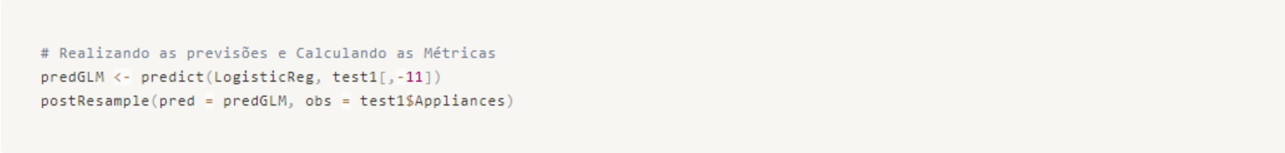
Model 02 – Logistic Regression



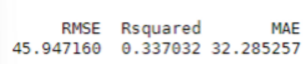
```
# Modelo de Regressão com Logistic Regression
fitControl <- trainControl(method = 'repeatedcv',
                           number = 20,
                           search = 'grid')

LogisticReg <- train(Appliances ~ .,
                     data = training1,
                     method = 'glm',
                     metric = 'RMSE',
                     maximize = FALSE,
                     preProcess = c('center', 'scale'))

LogisticReg
```



```
Generalized Linear Model

11103 samples
   10 predictor

Pre-processing: centered (10), scaled (10)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 11103, 11103, 11103, 11103, 11103, 11103, ...
Resampling results:

  RMSE      Rsquared   MAE
  45.93313  0.3316307  32.06113
```



```
# Realizando as previsões e Calculando as Métricas
predGLM <- predict(LogisticReg, test1[,-11])
postResample(pred = predGLM, obs = test1$Appliances)
```

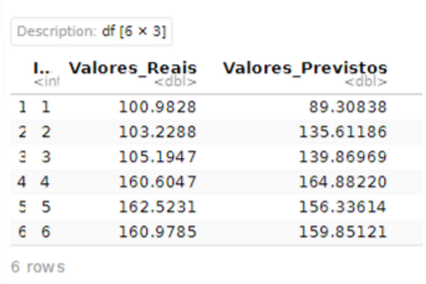


```
     RMSE  Rsquared       MAE
45.947160  0.337032 32.285257
```

We can see that the eXtreme Gradient Boosting model performed better, as indicated by the lower RMSE and MAE values, and a higher R² value, indicating 82% explainability of the predictor variables with respect to the response variable.

Therefore, let's analyze a graphical comparison between the predicted and actual values for the test data.

```r
# Comparando os Valores Previstos x Valores Reais
nomescol <- c('ID', 'Valores_Reais', 'Valores_Previstos')
idcol <- c(1:length(predGradientB))
comp <- cbind(idcol,
              (as.data.frame(test1$Appliances)),
              (as.data.frame(predGradientB)))
colnames(comp) <- nomescol

head(comp)
```

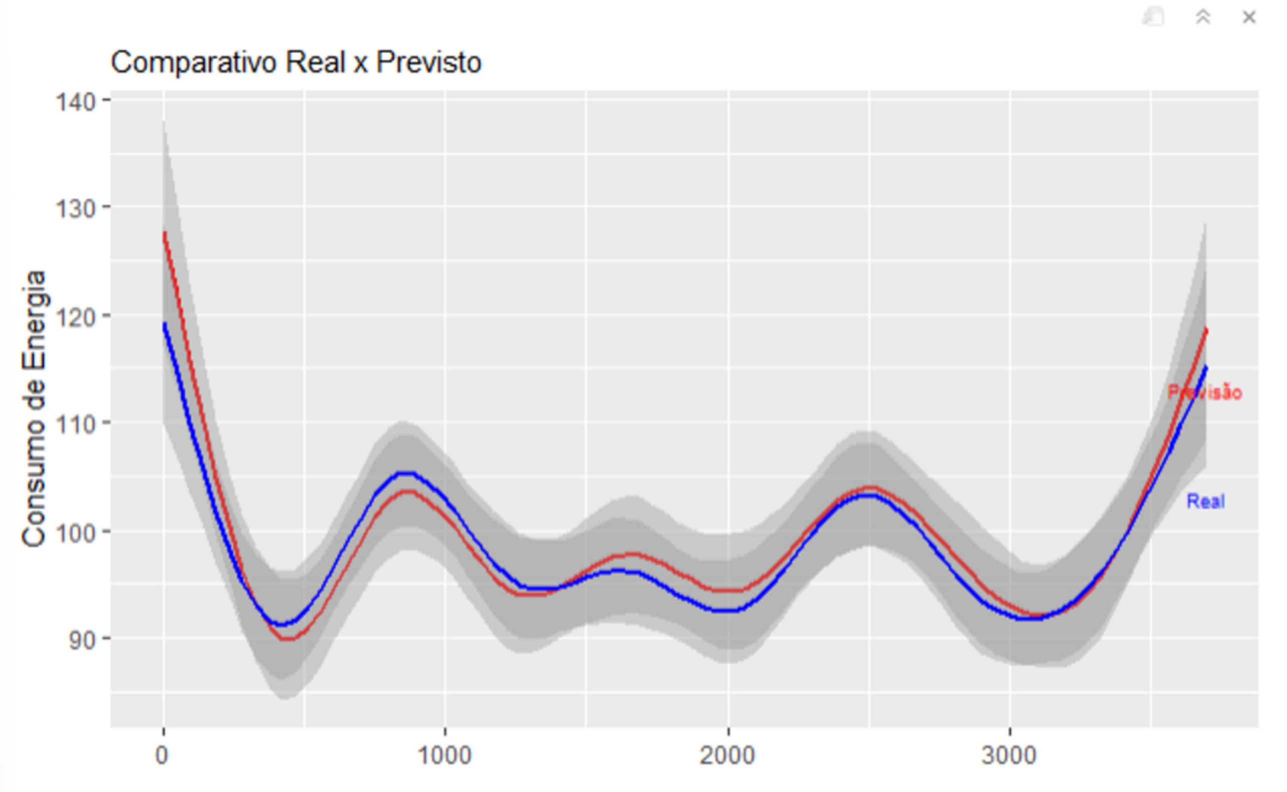


Description: df [6 × 3]

	I.. <int>	Valores_Reais <dbl>	Valores_Previstos <dbl>
1	1	100.9828	89.30838
2	2	103.2288	135.61186
3	3	105.1947	139.86969
4	4	160.6047	164.88220
5	5	162.5231	156.33614
6	6	160.9785	159.85121

6 rows

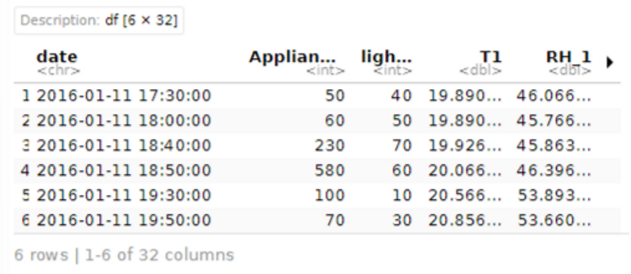
```r
# Criando Gráfico de Comparação
ggplot(comp) +
  geom_smooth(aes(x = ID, y = Valores_Reais),
              color = 'red',
              show.legend = 'TRUE') +
  geom_smooth(aes(x = ID, y = Valores_Previstos),
              color = 'blue',
              show.legend = TRUE) +
  labs(y = 'Consumo de Energia',
       x = '',
       subtitle = 'Comparativo Real x Previsto') +
  annotate(geom = 'text',
           x = 3700,
           y = 113,
           label = 'Previsão',
           color = 'red',
           size = 2.5) +
  annotate(geom = 'text',
           x = 3700,
           y = 103,
           label = 'Real',
           color = 'blue',
           size = 2.5)
```



Model Deploy

We conclude our project by deploying the best model and generating predictions using new data. Typically, the data used for deployment does not include the actual values we want to predict. However, in this case, the response variable is present, so we can use it to evaluate how the model performed by comparing the RMSE, R², and MAE metrics between the predicted and actual values.

```
# Carregando novos dados
dfd <- read.csv('Dados/projeto8-testing.csv', sep = ',')

# Visualizando os dados
head(dfd)
```



Description: df [6 × 32]

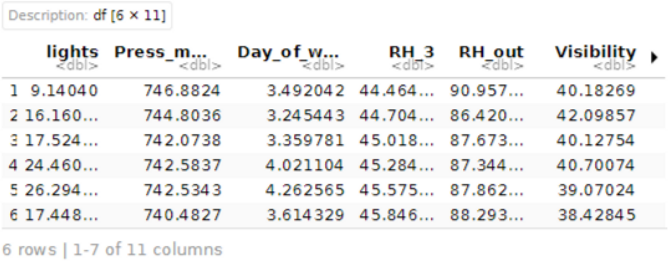
	date <chr>	Applian... <int>	ligh... <int>	T1 <dbl>	RH_1 <dbl>
1	2016-01-11 17:30:00	50	40	19.890...	46.066...
2	2016-01-11 18:00:00	60	50	19.890...	45.766...
3	2016-01-11 18:40:00	230	70	19.926...	45.863...
4	2016-01-11 18:50:00	580	60	20.066...	46.396...
5	2016-01-11 19:30:00	100	10	20.566...	53.893...
6	2016-01-11 19:50:00	70	30	20.856...	53.660...

6 rows | 1-6 of 32 columns

Developed by Thiago Bulgarelli

Contact: bugath36@gmail.com

```
# Preparando os Dados para o Modelo
dfd$date <- NULL
dfd$NSM <- NULL
dfd$Day_of_week <- as.numeric(as.factor(dfd$Day_of_week))
dfd$WeekStatus <- as.numeric(as.factor(dfd$WeekStatus))

# Tratando os Outliers
dfd <- outForest(dfd,
                 replace = 'predictions',
                 threshold = 0.01)$Data

# Alocando somente as Variáveis Importantes
dfdf <- as.data.frame(dfd) %>% select(c(lights,
                          Press_mm_hg,
                          Day_of_week,
                          RH_3,
                          RH_out,
                          Visibility,
                          T8,
                          T5,
                          Windspeed,
                          RH_1,
                          Appliances))

# Visualizando o Dataset Final
head(dfdf)
```

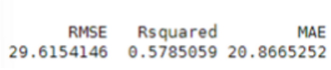


Description: df [6 × 11]

	lights	Press_m...	Day_of_w...	RH_3	RH_out	Visibility
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	9.14040	746.8824	3.492042	44.464...	90.957...	40.18269
2	16.160...	744.8036	3.245443	44.704...	86.420...	42.09857
3	17.524...	742.0738	3.359781	45.018...	87.673...	40.12754
4	24.460...	742.5837	4.021104	45.284...	87.344...	40.70074
5	26.294...	742.5343	4.262565	45.575...	87.862...	39.07024
6	17.448...	740.4827	3.614329	45.846...	88.293...	38.42845

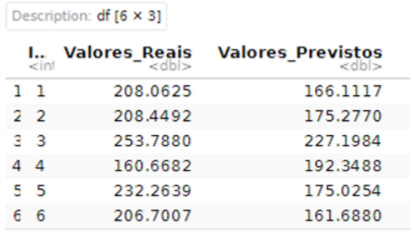
6 rows | 1-7 of 11 columns

```
# Aplicando Modelo eXtreme Gradient Boosting
deploy <- predict(GradientBoosting, dfdf[,-11])
postResample(pred = deploy, obs = dfdf$Appliances)
```



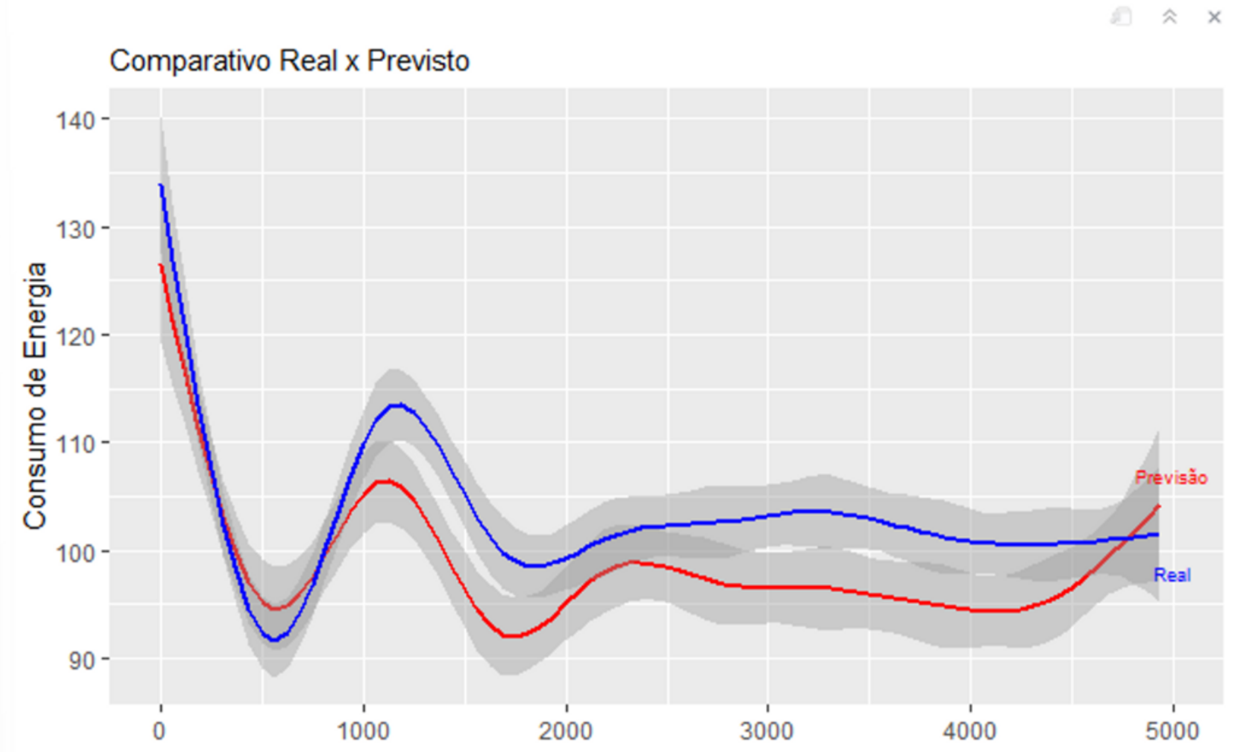
```
      RMSE   Rsquared        MAE
29.6154146  0.5785059 20.8665252
```

```
# Tabela Comparativa da Performance
ColNomes <- c('ID', 'Valores_Reais', 'Valores_Previstos')
ID <- c(1:length(deploy))
Resultado <- cbind(ID,
                   (as.data.frame(dfdf$Appliances)),
                   (as.data.frame(deploy)))
colnames(Resultado) <- ColNomes

head(Resultado)
```



```
Description: df [6 × 3]

    I..   Valores_Reais   Valores_Previstos
   <int>          <dbl>               <dbl>
1   1          208.0625            166.1117
2   2          208.4492            175.2770
3   3          253.7880            227.1984
4   4          160.6682            192.3488
5   5          232.2639            175.0254
6   6          206.7007            161.6880
```

```
# Gráfico Comparativo da Performance
ggplot(Resultado) +
  geom_smooth(aes(x = ID, y = Valores_Reais),
              color = 'red',
              show.legend = 'TRUE') +
  geom_smooth(aes(x = ID, y = Valores_Previstos),
              color = 'blue',
              show.legend = TRUE) +
  labs(y = 'Consumo de Energia',
       x = '',
       subtitle = 'Comparativo Real x Previsto') +
  annotate(geom = 'text',
           x = 5000,
           y = 107,
           label = 'Previsão',
           color = 'red',
           size = 2.5) +
  annotate(geom = 'text',
           x = 5000,
           y = 98,
           label = 'Real',
           color = 'blue',
           size = 2.5)
```



Project Delivered!