Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

Previsão de Consumo de Energia Residencial por Eletrodomésticos com Machine Learning

Este projeto de IoT tem como objetivo a criação de modelos preditivos para a previsão de consumo de energia de eletrodoméstico. Os dados utilizados incluem medições de sensores de temperatura e umidade de uma rede sem fio, previsão do tempo de uma estação de um aeroporto e uso de energia utilizada por luminárias.

Nesse projeto de aprendizado de máquina precisamos realizar a filtragem de dados para remover parâmetros não preditivos e selecionar os melhores recursos para a previsão. O conjunto de dados foi coletado por um período de 10 minutos por cerca de 5 meses. As condições de temperatura e umidade da casa foram monitoradas com uma rede de sensores sem fio da ZigBee.

Cada nó sem fio transmitia as condições de temperatura e umidade em torno de 3 min. Em seguida, a média dos dados foi calculada para períodos de 10 minutos. Os dados de energia foram registrados a cada 10 minutos com medidores de energia de barramento "m". O tempo da estação meteorológica mais próxima do aeroporto (Aeroporto de Chievres, Bélgica) foi baixado de um conjunto de dados públicos do Reliable Prognosis (rp5.ru) e mesclado com os conjuntos de dados experimentais usando a coluna de data e hora. Duas variáveis aleatórias foram incluídas no conjunto de dados para testar os modelos de regressão e filtrar os atributos não preditivos (parâmetros).

Nosso trabalho agora é construir um modelo preditivo que possa prever o consumo de energia com base nos dados de sensores IoT coletados. Usaremos a linguagem R para a realização deste projeto.

Os dados podem ser baixados no link abaixo:

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

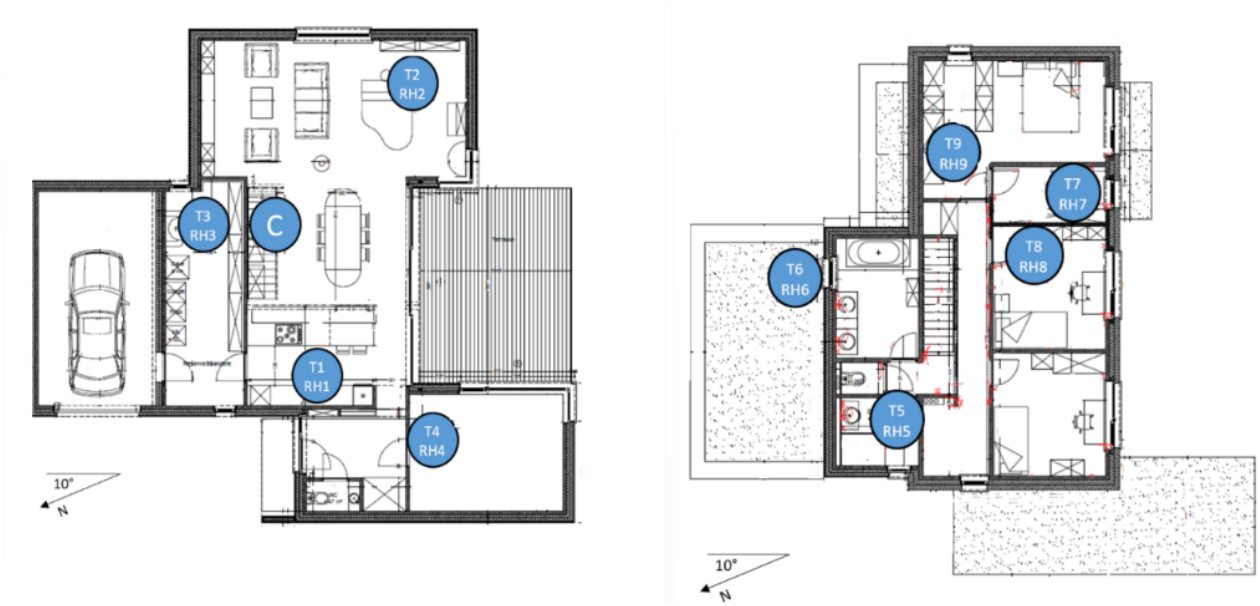
Dicionário de Dados

- **Appliances**, energia utilizada por eletrodomésticos em Wh
- **lights**, energia utilizada pelas lâmpadas da casa em Wh
- **T1**, Temperatura na área da Cozinha, em Celsius
- **RH_1**, Umidade na área da Cozinha, em %
- **T2**, Temperatura na área da Sala de Estar, em Celsius
- **RH_2**, Umidade na área da Sala de Estar, em %
- **T3**, Temperatura na área da Lavanderia, em Celsius
- **RH_3**, Umidade na área da Lavanderia, em %
- **T4**, Temperatura na área do Escritório, em Celsius

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

- **RH_4**, Umidade na área do Escritório, em %
- **T5**, Temperatura na área do Banheiro, em Celsius
- **RH_5**, Umidade na área do Banheiro, em %
- **T6**, Temperatura na área Externa (Lado Norte), em Celsius
- **RH_6**, Umidade na área Externa (Lado Norte), em %
- **T7**, Temperatura na área da Sala de Engomar, em Celsius
- **RH_7**, Umidade na área da Sala de Engomar, em %
- **T8**, Temperatura na área do Quarto Filhos, em Celsius
- **RH_8**, Umidade na área do Quarto Filhos, em %
- **T9**, Temperatura na área do Quarto Casal, em Celsius
- **RH_9**, Umidade na área do Quarto Casal, em %
- **To**, Temperatura Local (Medido da Estação Meteorológica de Chievres), em Celsius
- **Pressure**, Pressão Atmosférica (Medido da Estação Meteorológica de Chievres), em mmHg
- **RH_out**, Umidade Local (Medido da Estação Meteorológica de Chievres), em %
- **Windspeed**, Velocidade do Vento (Medido da Estação Meteorológica de Chievres), em m/s
- **Visibility**, Visibilidade (Medido da Estação Meteorológica de Chievres), em km
- **Tdewpoint**, Temperatura de Condensação da Água ou Ponto de Orvalho (Medido da Estação Meteorológica de Chievres), em Celsius
- **rv1**, Variável Aleatória Não Dimensional
- **rv2**, Variável Aleatória Não Dimensional



Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

Carregando Pacotes

```
# 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)
```

Carregando os Dados

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

Limpeza e Organização dos Dados

```
# 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" ...
```

Podemos perceber que temos algumas variáveis do tipo "CHR". A variável "date" deve ser do tipo data para que possamos, se necessário mais a frente, quebrar em dia, mês, ano, portanto vamos alterar seu tipo neste momento.

As Variáveis "WeekStatus" e "Day_of_Week" vamos analisar seus valores únicos e aplicar o tipo fator.

```
# 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)
```

Feita as adequações, vamos verificar se temos dados ausentes.

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

```
[1] 0
```

Por fim não temos valores ausentes em nosso dataset. Para finalizar, vamos remover a coluna "MSM" pois não temos descrição do que se trata a variável, portanto vamos considerá-la desnecessária.

```
# 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 ...
```

Agora estamos prontos para iniciar a análise exploratória dos dados!

Análise Exploratória dos Dados

Vamos separar as Variáveis Numéricas das Variáveis Categóricas.

```
# 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"
```

Elaborado por Thiago Bulgarelli

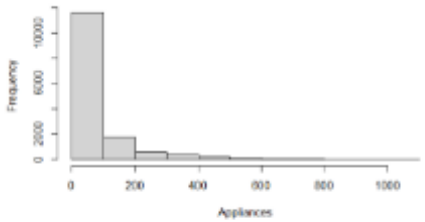
Contato: bugath36@gmail.com

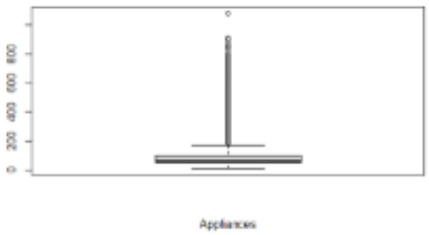
```
# Variáveis Categóricas
C <- df[colnames(df)[30:31]]
```

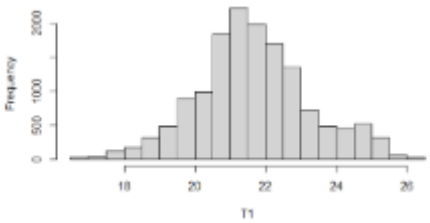
Separada as Variáveis, vamos analisar o comportamento de cada uma e identificar possíveis insights.

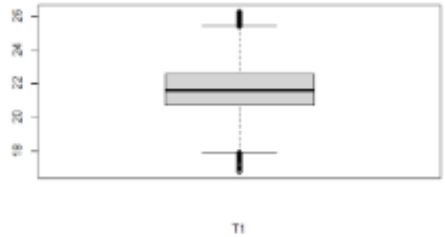
Variáveis Numéricas

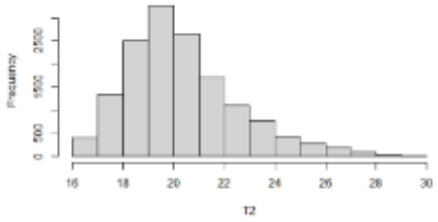
```
# Histogramas das Variáveis Numéricas
for (i in 2:29){
  hist(N[, i],
       xlab = colnames(N[i]),
       main = '')
  boxplot(N[, i],
          xlab = colnames(N[i]),
          main = '')
}
```

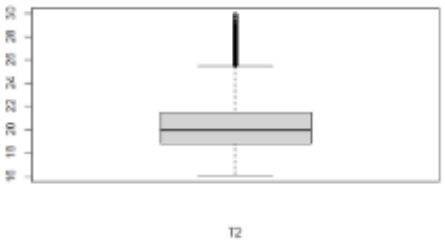


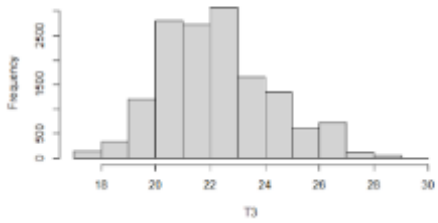


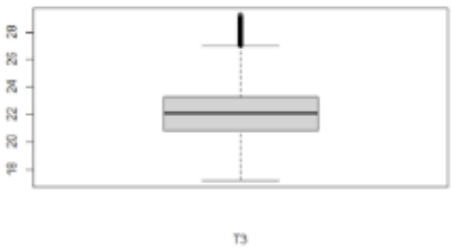






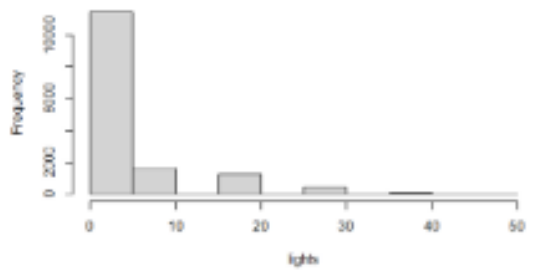


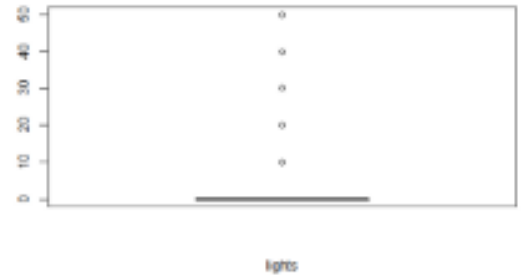


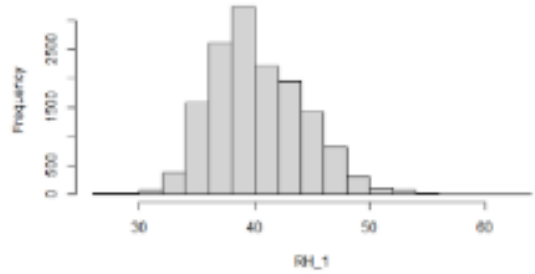


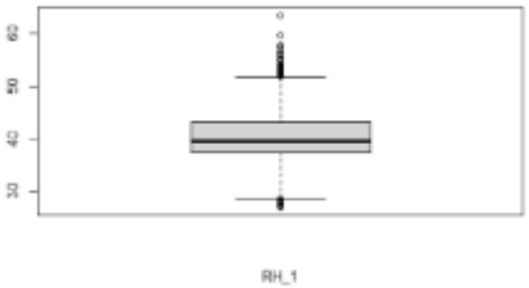
Elaborado por Thiago Bulgarelli

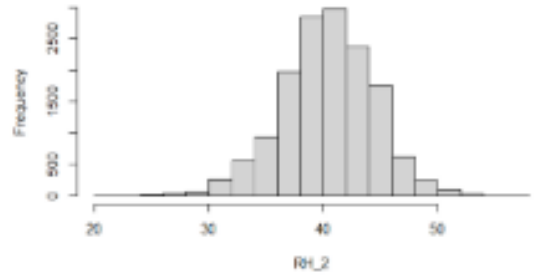
Contato: bugath36@gmail.com

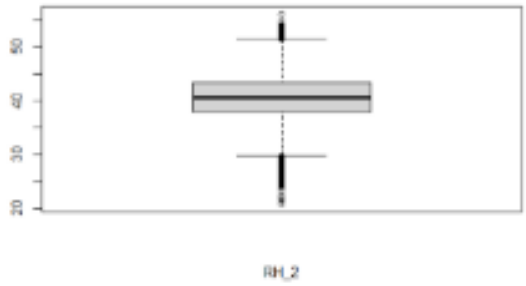


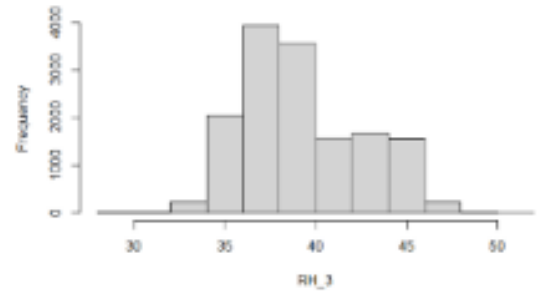


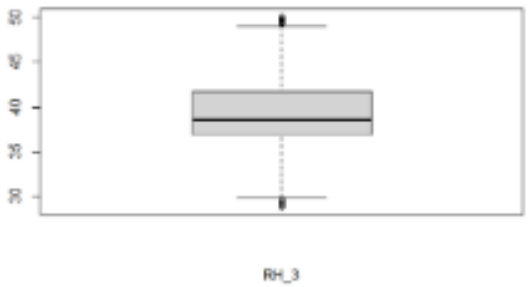


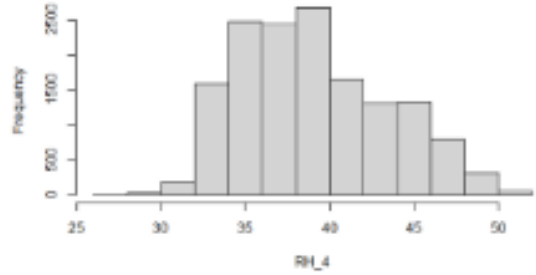


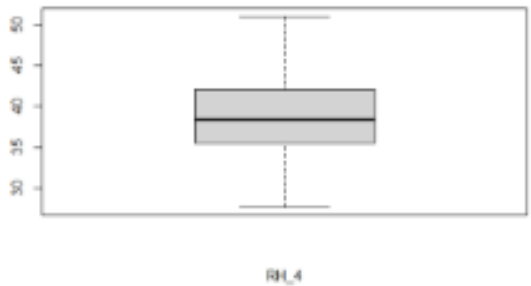






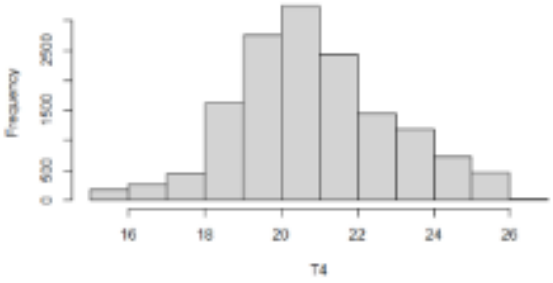


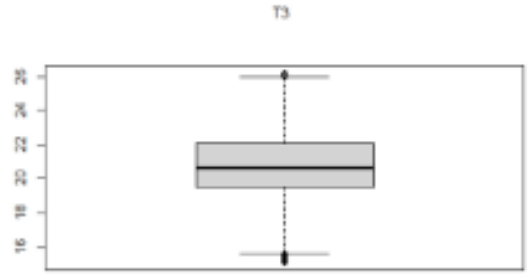


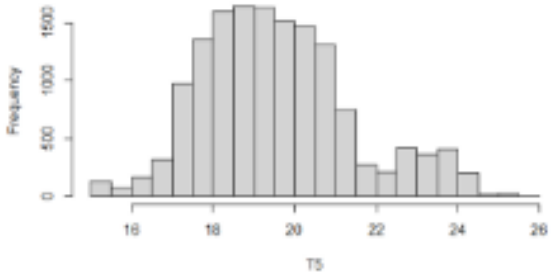


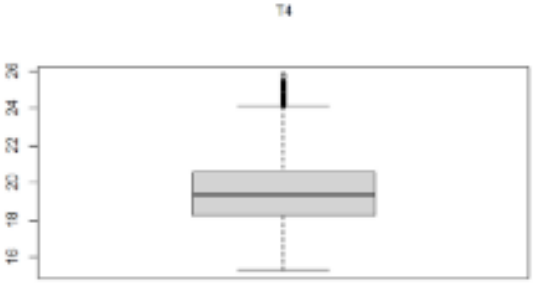
Elaborado por Thiago Bulgarelli

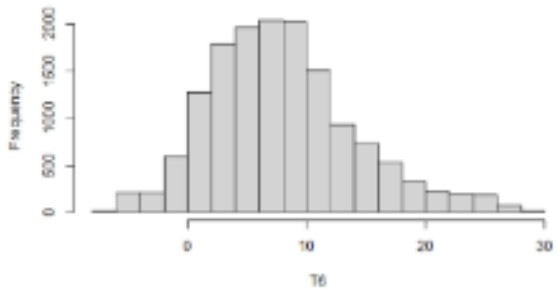
Contato: bugath36@gmail.com

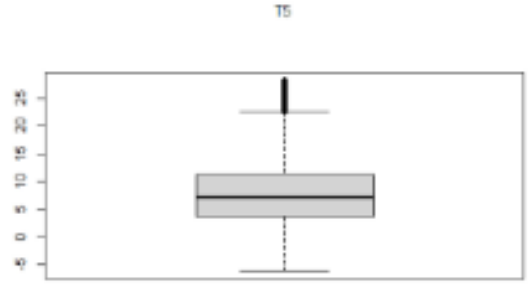


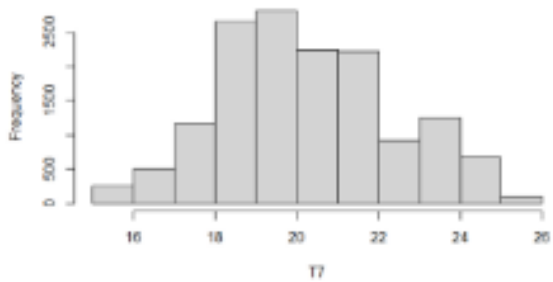


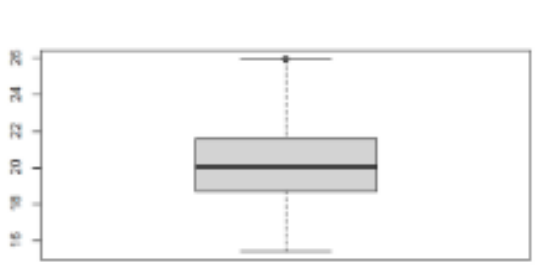


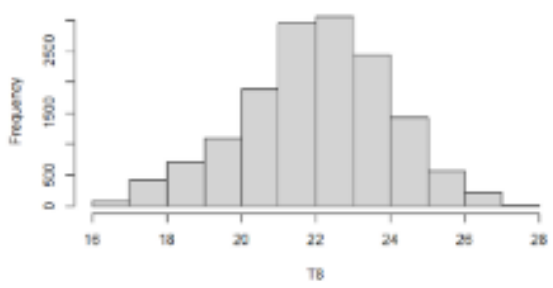


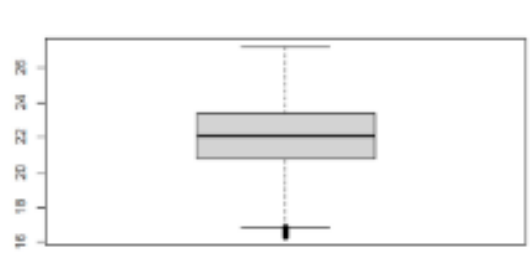






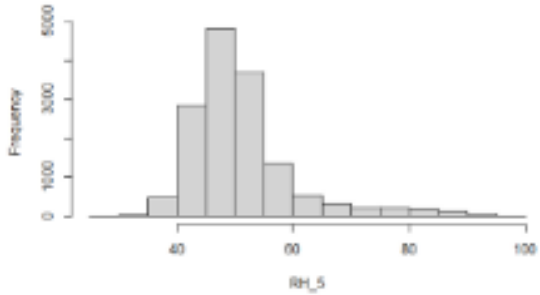


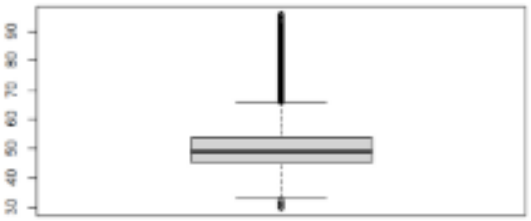


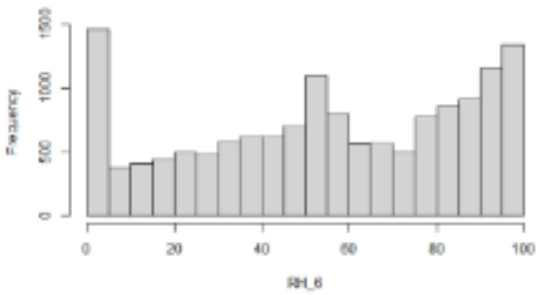


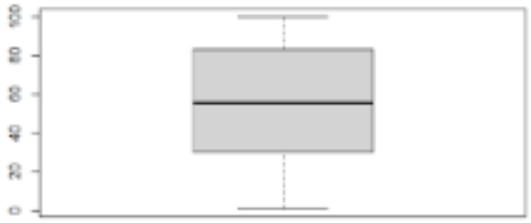
Elaborado por Thiago Bulgarelli

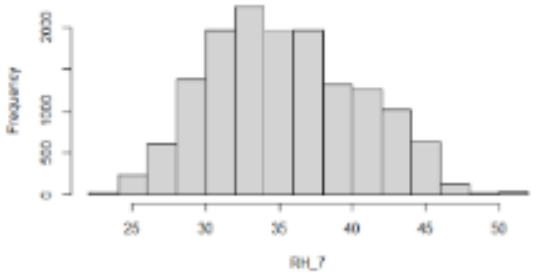
Contato: bugath36@gmail.com

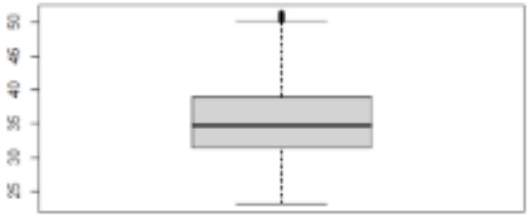


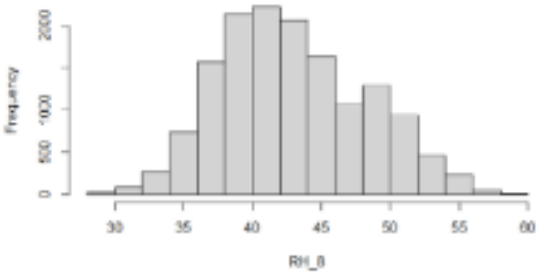


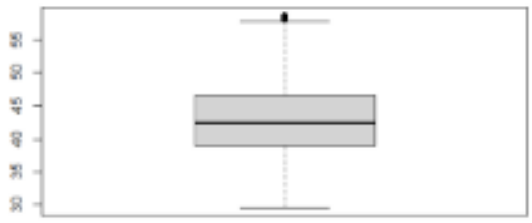


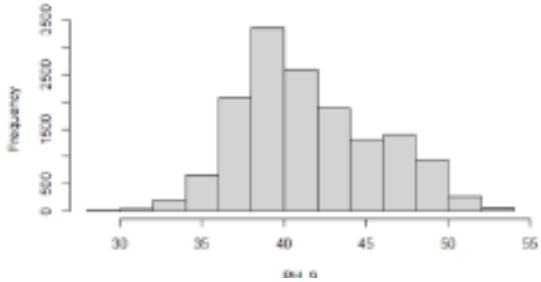


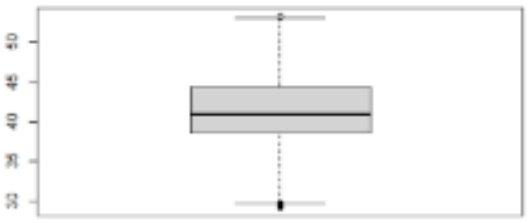






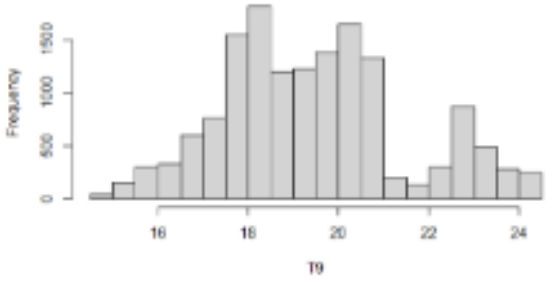


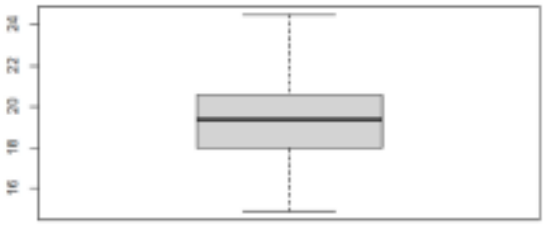


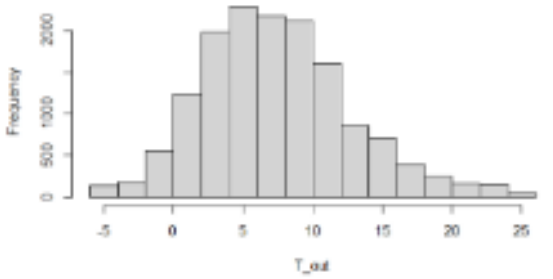


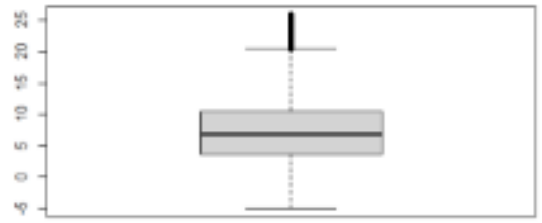
Elaborado por Thiago Bulgarelli

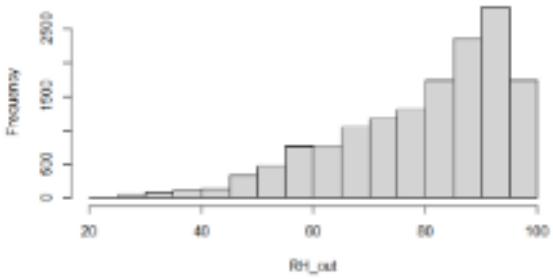
Contato: bugath36@gmail.com

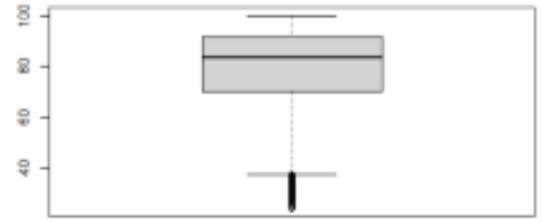


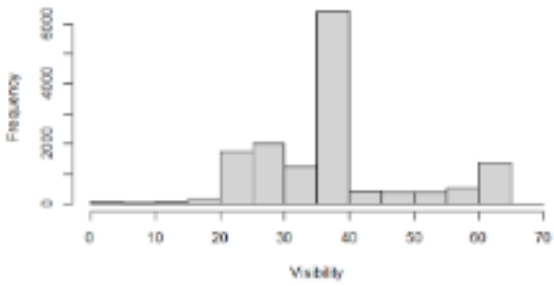


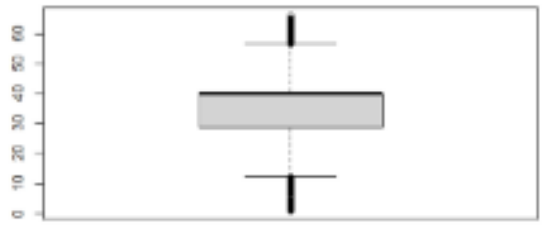


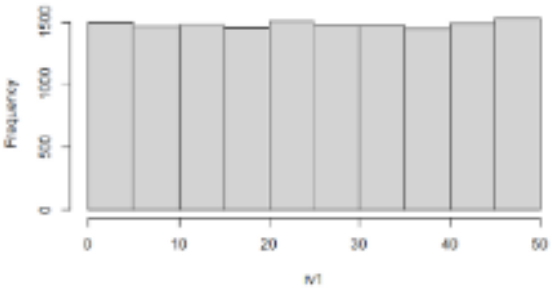


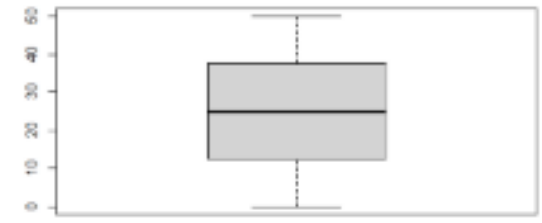


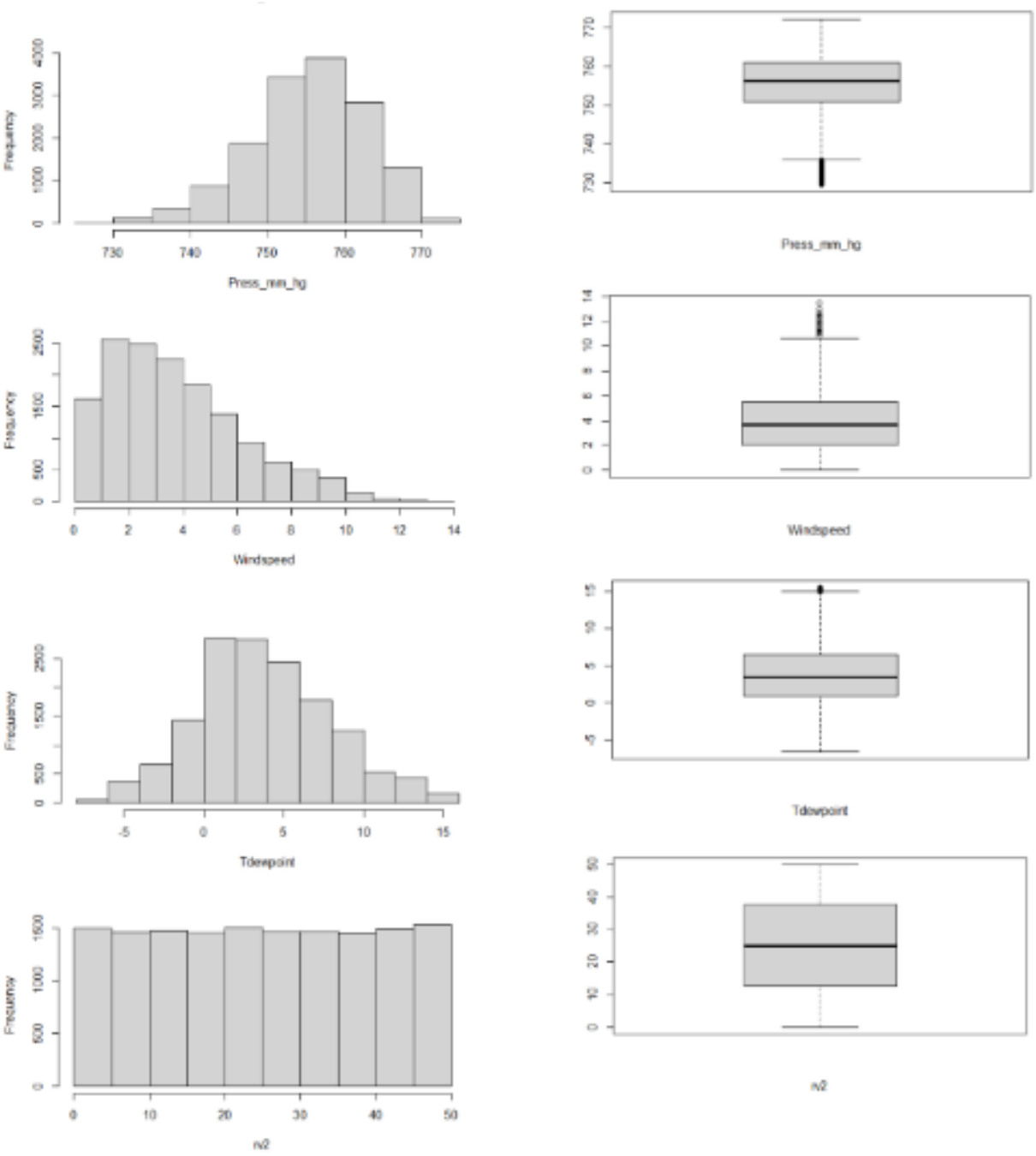












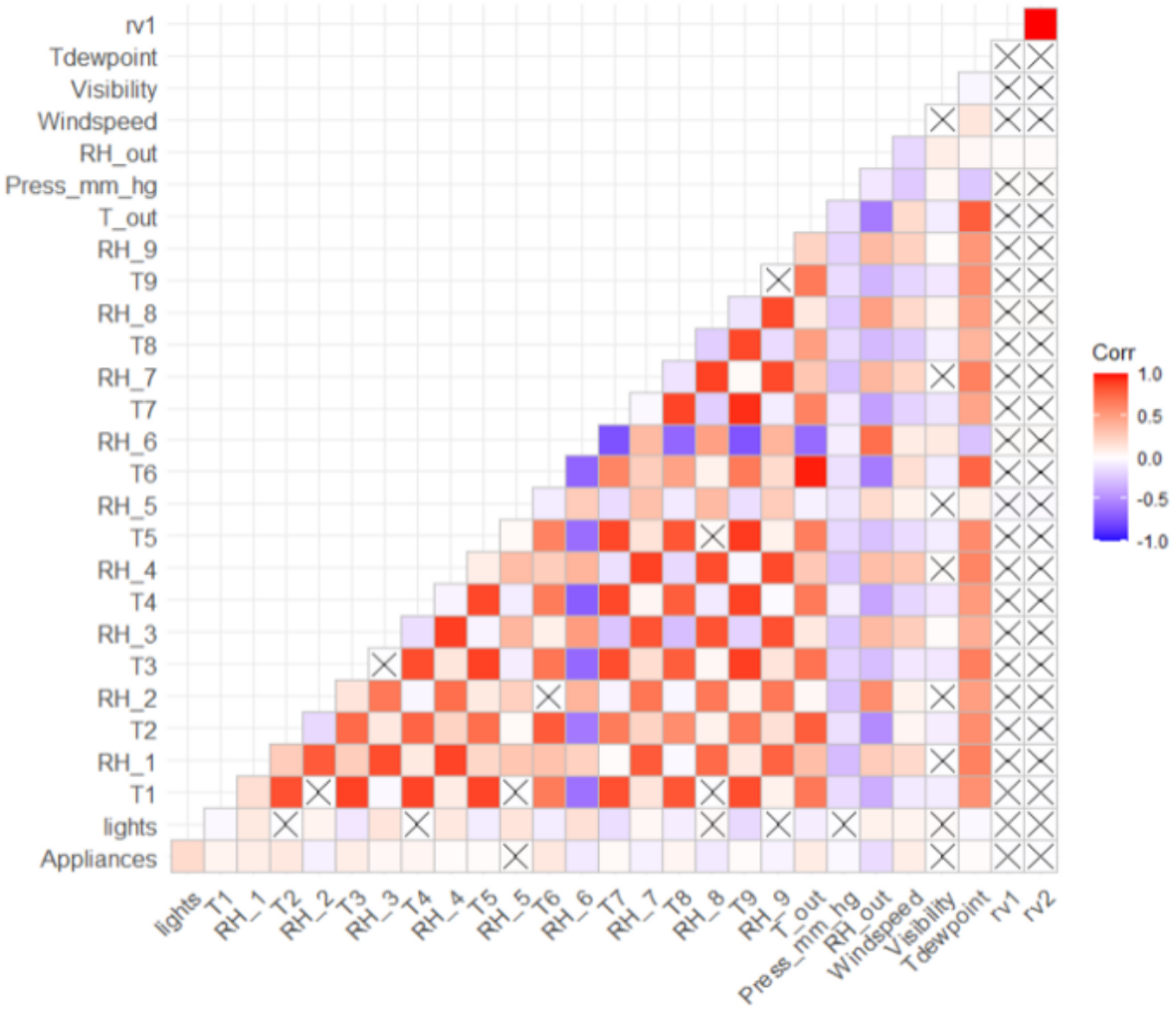
Algumas observações interessantes:

- As variáveis de Temperatura e Umidade possuem distribuições muito semelhantes a normal, pois possuem valores de Média e Mediana muito próximos. Não vamos aplicar um teste de Shapiro para verificar a hipótese de normalidade pois não é relevante ao resultado de nossa modelagem preditiva a princípio.
- As variáveis "r1" e "r2" são praticamente constantes dentro de nossas observações, claramente serão descartadas em nossa seleção de variáveis mais importantes.

- A variável "lights" possui uma distribuição próxima a normal, porém vamos analisar melhor sua correlação com a variável resposta (Applience) visto que inicialmente são variáveis concorrentes e independentes entre si. Vamos confirmar analisando as correlações.
- A variável Resposta (Applience) tem uma distribuição que se assemelha a Normal, porém possuímos alguns dados outliers, assim como a maioria das variáveis preditoras. Vamos tratar esse tema mais adiante.

Vamos verificar como as variáveis se correlacionam.

```
# 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')
```

Alguns pontos nos chamam a atenção:

- As variáveis rv1 e rv2, não possuem correlação com absolutamente nenhuma outra variável, pois como mencionamos, são praticamente constantes e foram inseridas artificialmente em nossos dados pelo Kaggle.
- Nossa Variável Resposta Appliances demonstra ter algumas correlações positivas e negativas leves com algumas de nossas variáveis. O que nos mostra que temos a possibilidade de resolver nosso problema com um modelo de estrutura linear.

- A Variável Windspeed possuem alguma correlação, apesar de baixa, com a Temperatura do Quarto do Filho, Umidade do Quarto do Casal e Umidade Local, e ainda com a Pressão Atmof. do Local. tem pouca correlação ou nenhuma com nossos dados.
- A Temperatura da Sala sofre forte influência negativa com a Umidade Local. A Umidade da Sala sofre forte influência positiva da Umidade Local. Provavelmente é uma área com janelas de ventilação.
- A Umidade do Quarto do Casal possui influência positiva da Umidade Local.

Percebemos que a Variável lights é a única variável com correlação mais forte com a variável resposta.

Para finalizarmos nossa análise das variáveis numéricas, vamos estudar como a variável resposta se comporta ao longo do tempo, utilizando a variável Date.

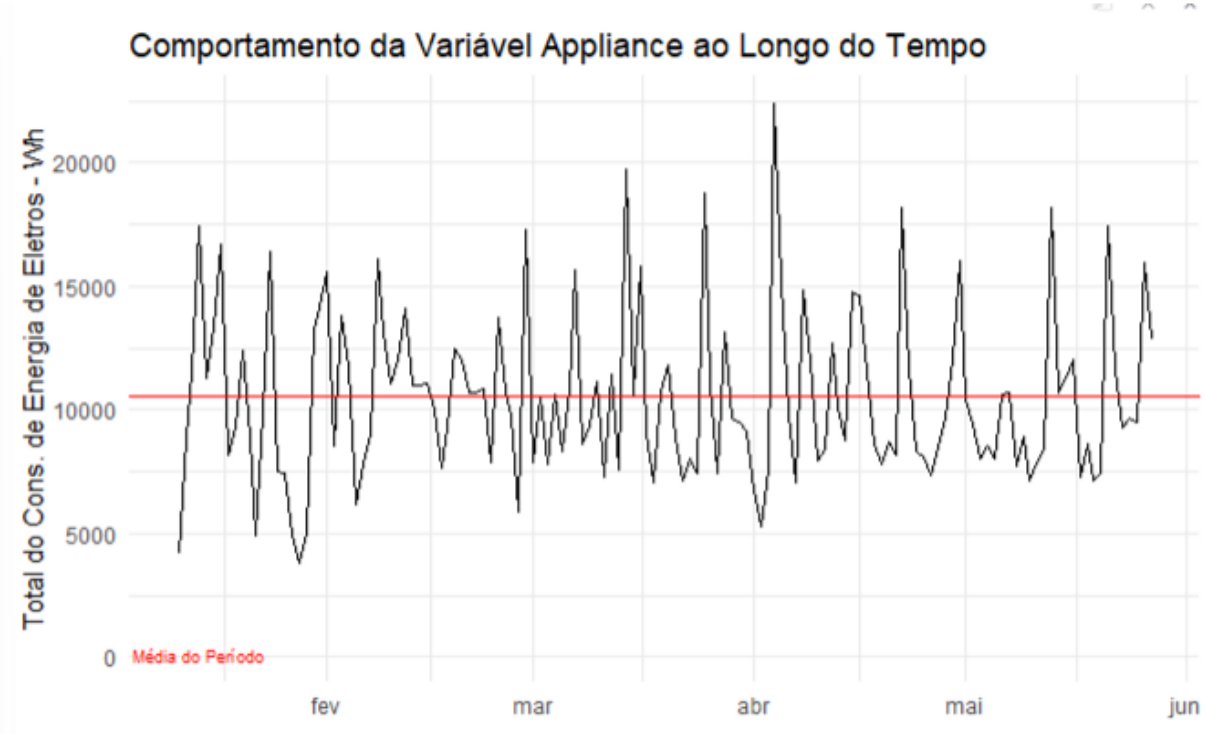
```
# Tabela da Média de Appliances ao longo do Tempo
scat <- df %>% select(c(date, Appliances))
scat$date <- as.Date(scat$date)
scat <- scat %>%
  group_by(date) %>%
  summarise(Consumo = sum(Appliances))
head(scat)
```

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

6 rows

```
# 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')
```



Podemos pontuar a respeito:

- O maior pico de consumo ocorreu em Abril, o dobro da média do período.
- Como era de se esperar, dentro da janela de tempo de 1 mês temos uma volatilidade grande com valores mínimos entre 75 Wh e 150 Wh em sua grande maioria. Isso pode ocorrer porque alguns eletrodomésticos utilizam maior quantidade de energia mas não são usados com muita frequência, como Máquinas de Lavar e Secar, Ferros de Passar entre outros.

Como a Variável Appliances teve um comportamento diferente em Abril em relação aos outros meses, acredito que caiba uma análise temporal as demais variáveis para identificar comportamento semelhante.

```
# 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))
```

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

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

A tibble: 6 × 25

date <date>	T1 <dbl>	RH_1 <dbl>	T2 <dbl>	RH_2 <dbl>	T3 <dbl>	RH_3 <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')
```

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

```
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')
```

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

```
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')
```

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

```
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')
```

```
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')
```

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

```
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')
```

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

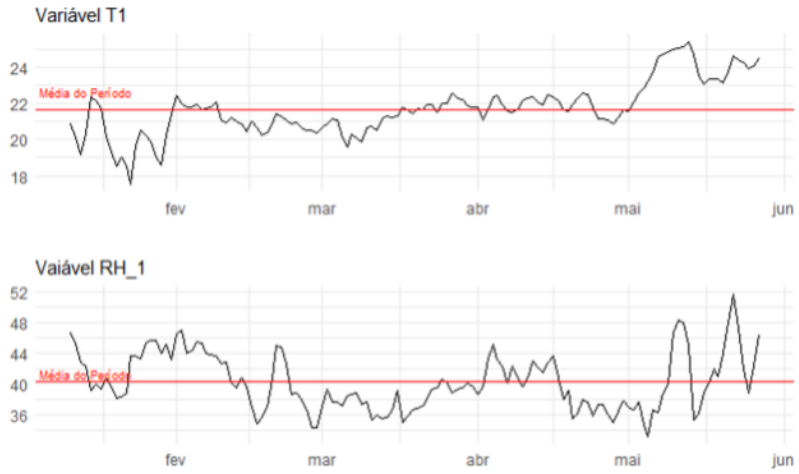
```
Wind <- ggplot(data = dfst) +
  geom_line(mapping = aes(x = date, y = Windspeed)) +
  theme_minimal() +
  labs(x = '',
       y = '',
       subtitle = 'Vaiável Windspeed') +
  geom_abline(aes(intercept = mean(Windspeed), slope = 0),
              color = 'red') +
  annotate(geom = 'text',
           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')
```

Vamos visualizar as Séries Temporais para cada Região da Casa e avaliar se temos em cada caso, alguma relação como tempo.

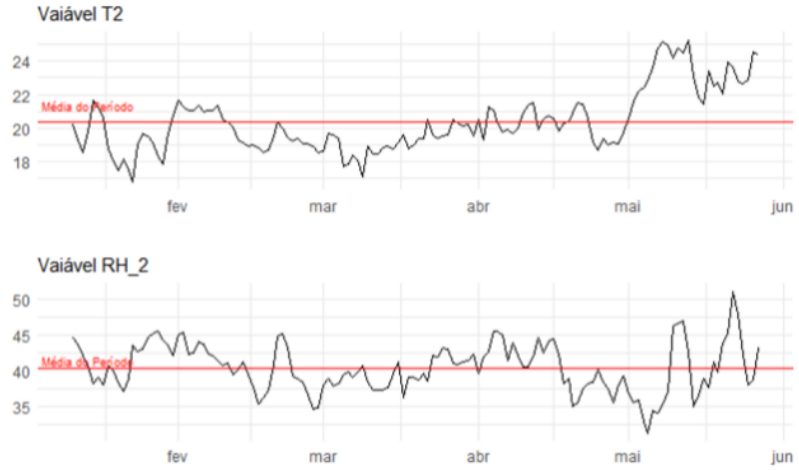
```
# Região 01 - Cozinha
T1 / RH1
```

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com



A umidade tem uma volatilidade maior em relação a Temperatura, o que é plausível, visto a época de chuvas. Temos um período de Maio a Junho com temperaturas mais altas e mais estáveis, porém com umidade mais volátil variando entre 36% e 52%. Ambas possuem tendência de alta de Maio para Junho.




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

Temos um comportamento bastante semelhante a região 01, com maio e junho trazendo uma tendência de alta.

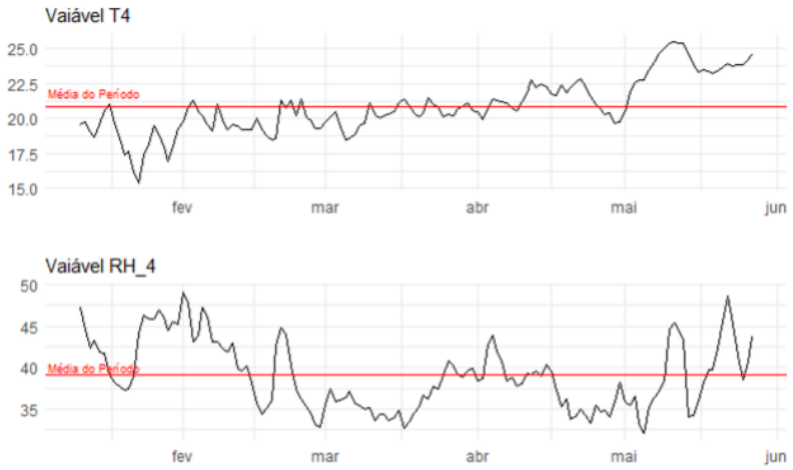
Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

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

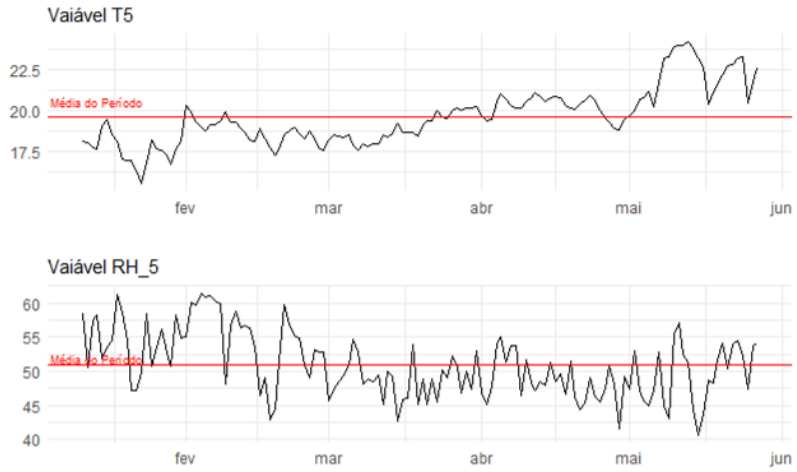
Na Lavanderia temos um cenário um pouco diferente. A temperatura em todo o Período tem uma tendência de crescimento, enquanto a umidade tem uma tendência de queda. Muitos fatores podem explicar esse comportamento, como exposição ao Sol, Sol matutino ou Vespertino, uso frequente dos equipamentos que podem gerar calor como máquina de lavar, secadora entre outros equipamentos comuns. Das regiões analisadas até o momento, é a que possuem mínimos e máximos menores de umidade.

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



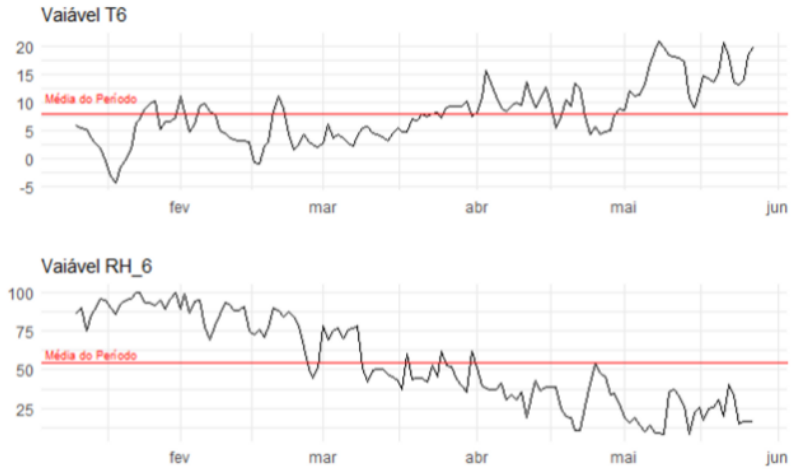
Temos para o Escritório comportamento semelhante as demais regiões, no geral uma tendência de queda da umidade e aumento da temperatura. Caracteriza também a janela de maio-junho como tendo comportamento semelhante as demais regiões.

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



No banheiro temos oscilações de umidade de amplitudes maiores, o que não parece incomum, visto o uso para banho por exemplo. No mais temos um comportamento semelhante ao restante dos ambientes que estudamos.

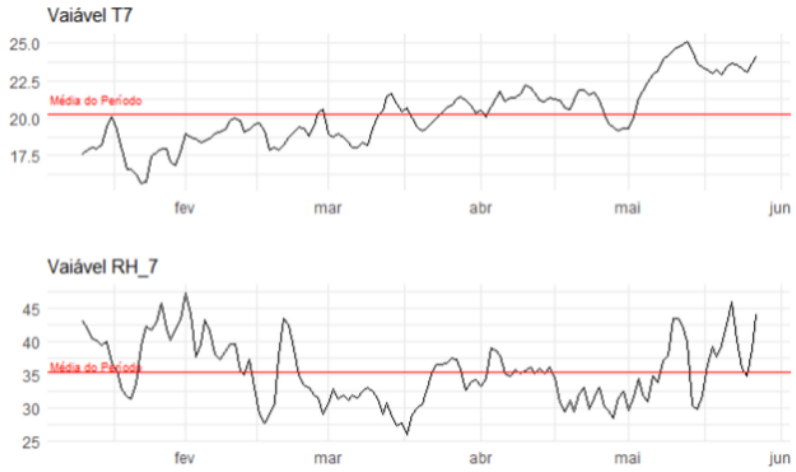
```
# Região 06 - Área Externa do Segundo Andar
T6 / RH6
```



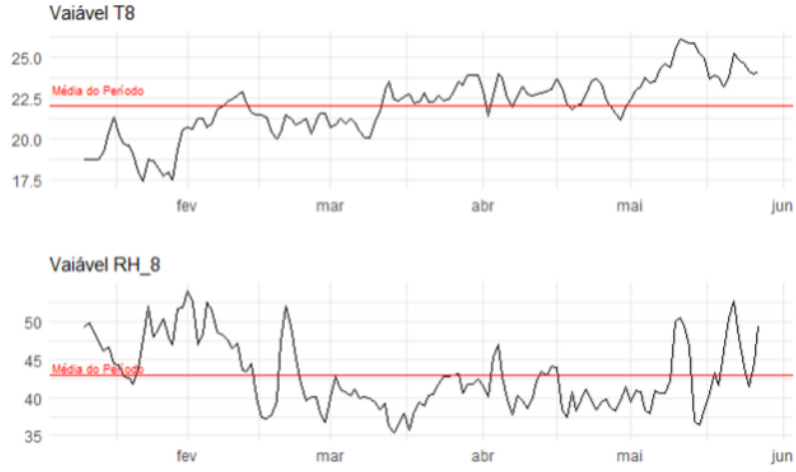
Para a área externa, temos um comportamento diferente quanto a umidade do período de maio e junho. Temos uma tendência de queda bastante incisiva, enquanto a temperatura em tendência de alta como nos demais ambientes.

Os valores de Umidade também são bem maiores que no restante da casa, assim como as temperaturas são bem mais baixas.

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



Podemos ver durante o período de maio-junho uma tendência de alta tanto para umidade quanto para temperatura. Nos períodos anteriores, o gráfico mostra uma tendência de queda para a umidade e alta para temperatura.

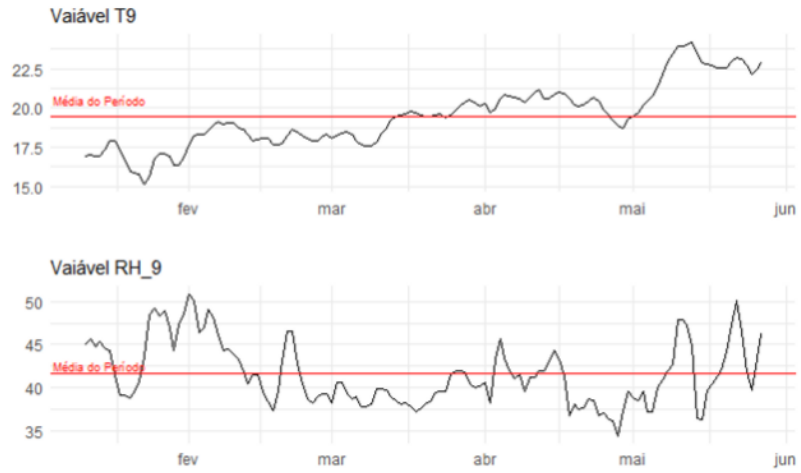


Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

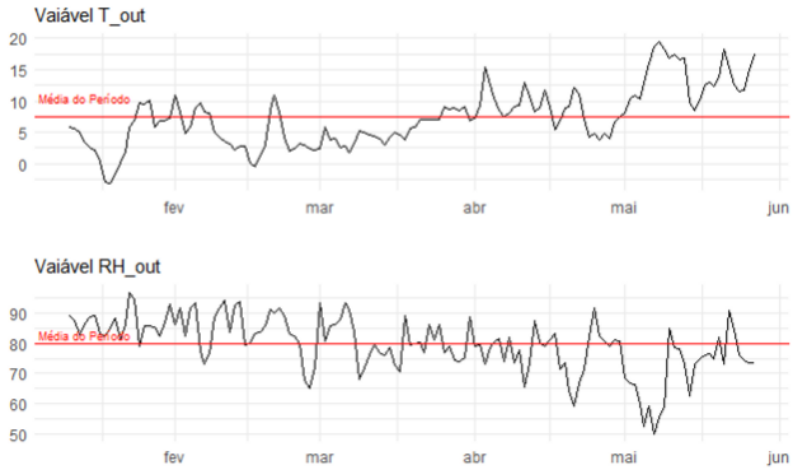
Comportamento semelhante aos outros ambientes da casa, com exceção da área externa (região 6).

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



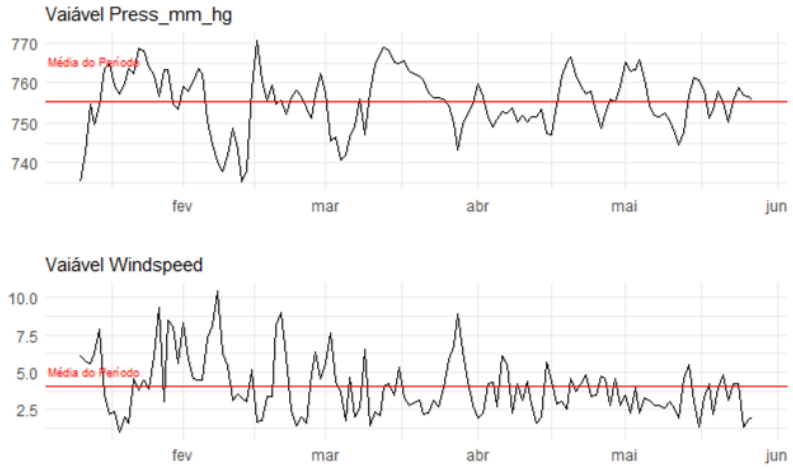
Novamente temos o mesmo padrão de outros ambientes. porém apesar da temperatura no geral ter uma tendência de alta, as oscilações no quarto do Casal são menos agressivas que nos demais ambientes.

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



Podemos perceber que as temperaturas na estação são muito menores que na casa, provavelmente pela calefação ou mesmo por ser uma área mais fechada que a externa. Já a umidade atinge valores muito mais altos com média de 80%.

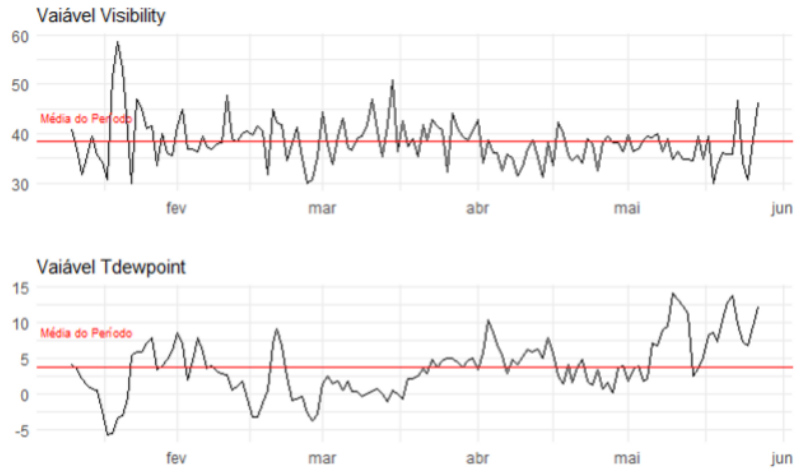
```
# Pressão Atmof e Velocidade do Vento na Estação
Press / Wind
```



Apesar da volatilidade da Pressão, não temos tendência de alta ou baixa. A variação é de +- 15 mm hg o que podemos dizer que é praticamente constante no período.

Quanto a Velocidade do Vento, temos uma tendência de queda que talvez possa corroborar com algumas hipóteses do aumento do consumo de energia pelos eletrodomésticos.

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



A visibilidade oscila sobre a média em +- 10 km ao longo de todo o período, sem tendência de aumento ou queda.

O ponto de orvalho sobe com o aumento da temperatura ambiente.

Por fim, uma teoria que podemos desenvolver, porém seria necessário mais informações para corroborar, seria que de abril em diante, os gráficos de temperatura passam a ser mais positivos em relação as suas médias. Eletrodomésticos como Ar condicionado, ventiladores, Churrasqueiras elétricas, são mais utilizados. As pessoas tendem a sair com mais frequência de suas casas, utilizam mais roupas, necessitando de maior frequência de uso de máquinas de lavar, secar e ferros de passar. Também se arrumam mais, aumento o uso de secadores de cabelo, chapinhas e etc.

Por fim, analisando comportamento das pessoas e pesquisando sobre a região de Chievre, entendemos que o período de Abril a Junho é o início das temperaturas mais agradáveis, momento esperado pelos habitantes, consequentemente corrobora com as suposições que criamos acima. Portando a mudança do Clima e estação em Chievre pode ajudar a explicar porque temos um pico de consumo de energia em eletrodomésticos em Abril e então uma tendência de alta a partir deste período.

Variáveis Categóricas

Para as variáveis Categóricas, temos janelas de tempo menores, como Workdays e Weekends, assim como os 7 dias da semana. Poderíamos analisar o comportamento de cada variável para estas janelas de tempo, mas para o nosso modelo e compreensão dos dados não se faz necessário. Porém vamos estudar como as variáveis se comportam nestes dias em média.

```
# 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... <fctr>	App <dbl>	Lig... <dbl>
Weekday	207....	8.634
Weekend	207....	6.560

2 rows

Podemos perceber claramente que não existe um consumo de energia nos eletrodoméstico significativamente maior entre Dias da Semana e Finais de Semana. Para as luzes o consumo ao longo da semana é maior que aos fins de semana. Provavelmente pela cultura de deixar luzes acesas ao sair de casa.

```
# 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))
```

```
# 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

Podemos ver aqui um comportamento no uso dos eletrodomésticos, visto que de sexta-feira a segunda-feira, temos os maiores consumos. Em compensação o consumo de energia elétrica pelas luzes é claramente superior na maioria dos dias da semana, chegando a dobrar em alguns casos.

O fato curioso é o consumo de energia das luzes ser muito baixo nas sextas-feiras e Sábados, metade em relação aos demais dias. Um pico na Segunda-feira também se destaca.

Pré-processamento

Para preparar nossos dados vamos desconsiderar a janela de tempo da variável 'date', pois nosso objetivo não é realizar modelagem preditiva de série temporal. Porém vamos manter as variáveis 'WeekStatus' e 'Day_of_Week' pois retratam uma janela estática, com possível influência na variável resposta. Sendo as variáveis do tipo categóricas, é importante fatorizá-las.

Também vamos dropar a variável 'lights' pois se trata de uma variável de consumo de energia e não possui relação com a variável resposta 'Appliances'.

Ainda precisamos reduzir a dimensionalidade e aplicar uma Padronização nos Dados para igualar as escalas e evitar que nosso modelo fique tendencioso e não generalizável.

```
# 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 ...
```

Como podemos verificar, as variáveis categóricas já são do tipo Fator, o que facilita nosso trabalho para Label Encoder.

```
# 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
```

Lembrando que a semana começa por Sunday (Domingo) sendo este o número 1, seguido pelos demais dias da semana em sequência (Seg - 2, Ter-3, Qua - 4, Qui - 5, Sex - 6, Sab - 7).

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

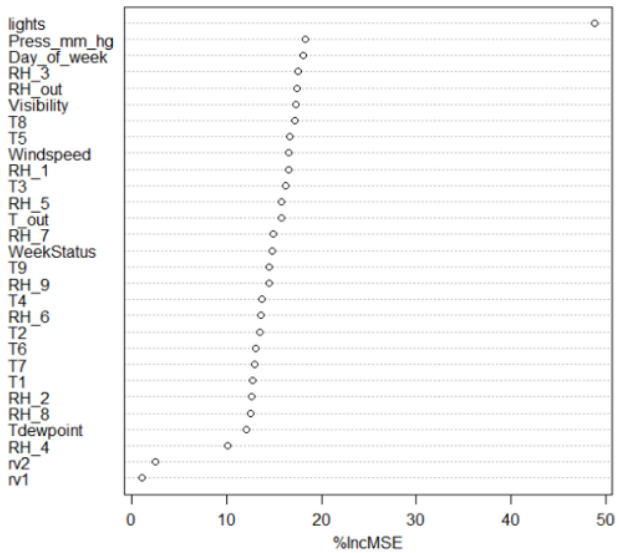
```
# Dividindo os dados em Treino e Teste
part <- createDataPartition(dffinal$Appliances,
                            p = 0.75,
                            list = FALSE)
training <- dffinal[part, ]
test <- dffinal[-part, ]
```

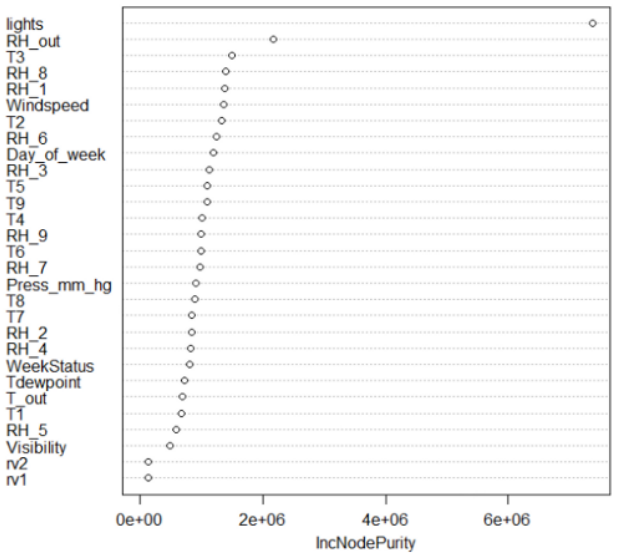
Dados divididos!

Agora vamos trabalhar na Seleção de Variáveis, utilizando o modelo Random Forest. Assim reduzimos a dimensionalidade e complexidade do Modelo Final.

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

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





No intuito de reduzir a complexidade do nosso modelo, vamos trabalhar com as 10 variáveis mais importantes apenas. Dessa forma nosso modelo fica menos complexo e mais generalizável.

```
# 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))
```

Modelagem Preditiva

Faremos uso de diferentes algoritmos para criar algumas versões de Modelos e analisar suas métricas de performance. Repare que estamos aplicando uma padronização nas variáveis preditoras, usando o argumento preProcess dentro da função train.

Modelo 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.
```

```r
# 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
```

Modelo 01 – Support Vector Machine (SVM)

```r
# 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
```

Modelo 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
```

Podemos verificar que o modelo eXtreme Gradient Boosting teve uma performance melhor, visto que o RSME e o MAE demonstram valores menores, alem de termos um R^2 bem mais eficiente com 82% de explicabilidade das variáveis preditoras em relação a variável resposta.

Sendo assim, vamos analisar um comparativo gráfico dos valores previstos e valores reais de teste.

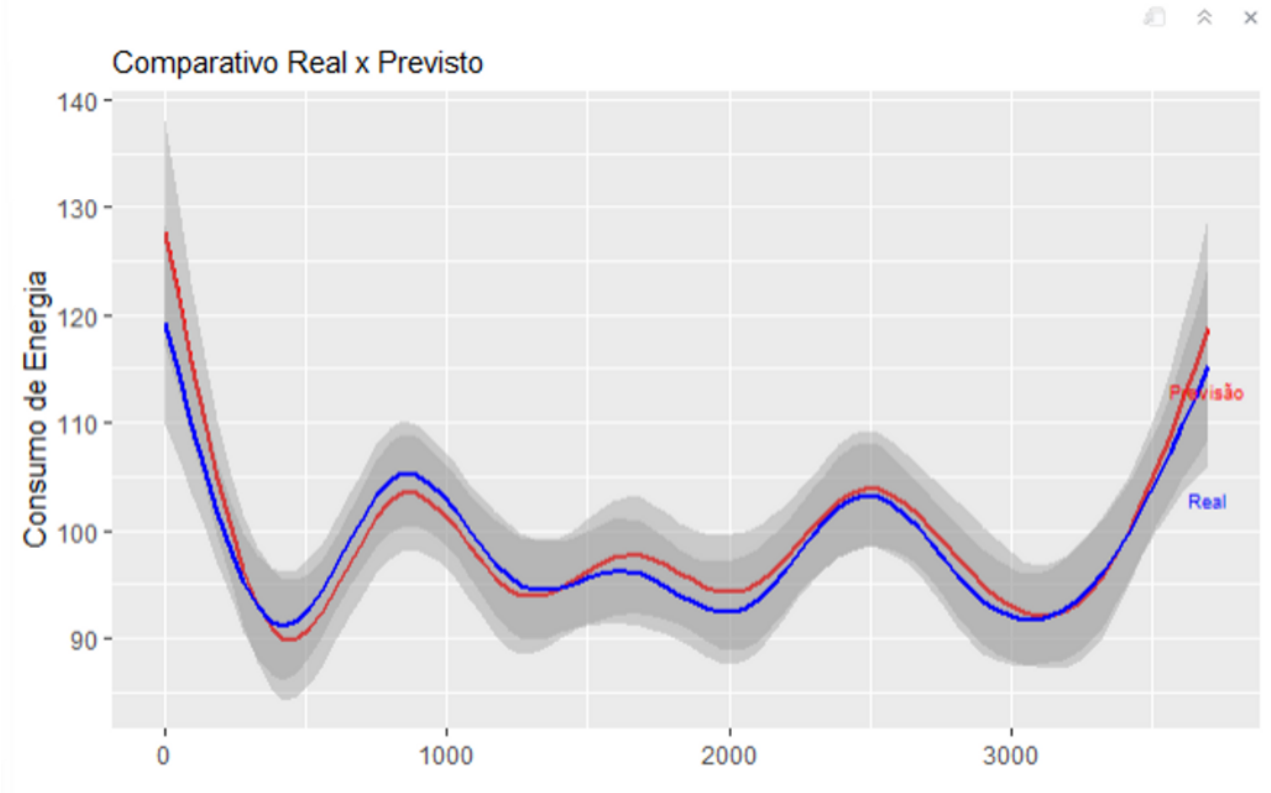
```
# Comparando os Valores Previstos x Valores Reais
nomescol <- c('ID', 'Valores_Reais', 'Valores_Previstos')
idcol <- c(1:length(predGradientB))
comp <- cbind(idcol,
              (as.data.frame(test1$Appliances)),
              (as.data.frame(predGradientB)))
colnames(comp) <- nomescol

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

```
# 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)
```



Deploy do Melhor Modelo

Finalizamos nosso projeto aplicando novos dados ao nosso modelo e entregando as previsões. Normalmente os dados aplicados no Deploy não possuem os valores reais que queremos prever, porém neste caso a Variável resposta se encontra presente, então vamos utilizar para avaliar como o modelo se comportou, comparando as Métricas RMSE, R^2 e MAE entre Previsto e Real.

```
# 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

Elaborado por Thiago Bulgarelli

Contato: 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 <dbl>	Press_m... <dbl>	Day_of_w... <dbl>	RH_3 <dbl>	RH_out <dbl>	Visibility <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)
```

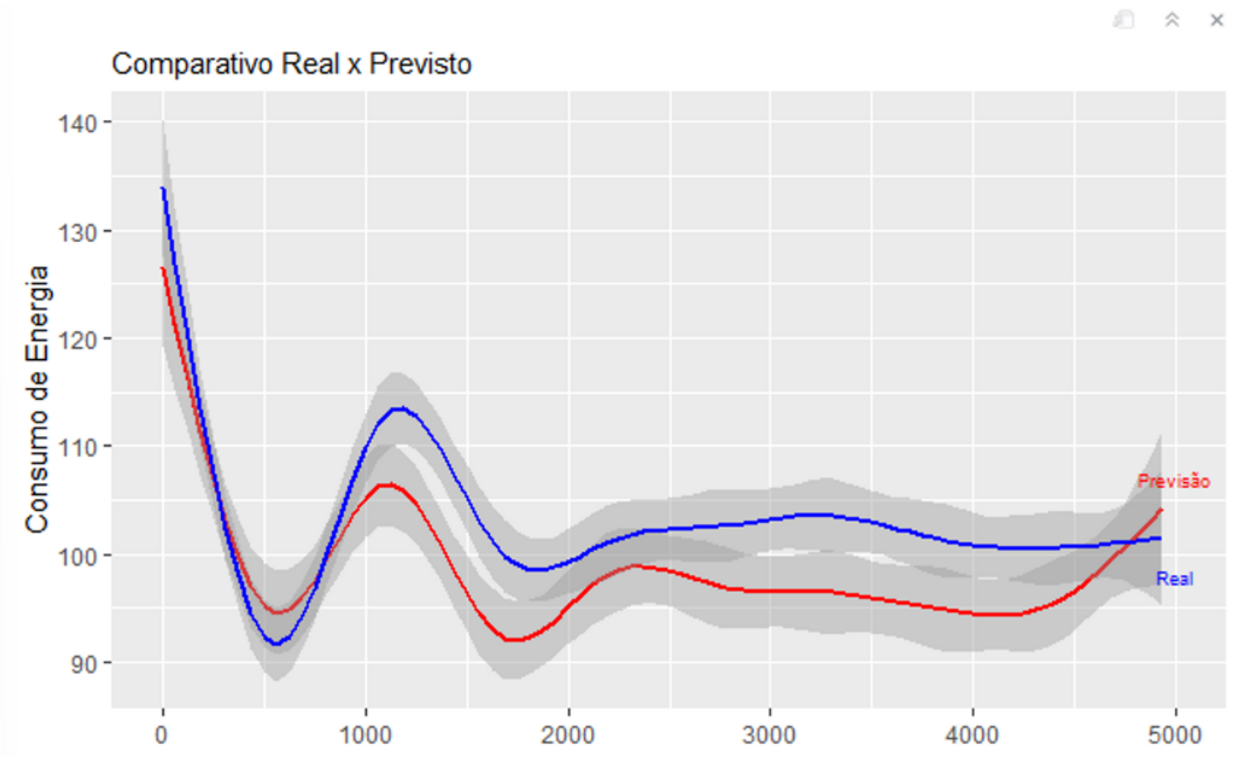
Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com

Description: df [6 × 3]

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



Projeto Entregue!

Elaborado por Thiago Bulgarelli

Contato: bugath36@gmail.com