Projeto08.R

rodrigolima82

2019-08-01

```r
# 01 - Iniciando o script de Machine Learning
```

```r
# Carregando o Dataset
dfTrain <- read.csv("data/train.csv")
dfTest <- read.csv("data/test.csv")

# Observacoes Iniciais
```

```r
# O dataset de treino contém 14.803 observacoes e 32 atributos.
# O dataset de teste contém 4.932 observacoes e 32 atributos.
# A coluna "Appliances" é o alvo.

# Juntando os dados de treino e teste
# Optei por realizar esse processo pois irei fazer um split dos dados no momento de treinamento do mode
# Assim, incorporo também os padroes do dataset de teste
df <- rbind(dfTrain,dfTest)

# Visualizando os dados
head(df)
```

```
##                  date Appliances lights       T1     RH_1   T2     RH_2
## 1 2016-01-11 17:00:00         60     30 19.89000 47.59667 19.2 44.79000
## 2 2016-01-11 17:10:00         60     30 19.89000 46.69333 19.2 44.72250
## 3 2016-01-11 17:20:00         50     30 19.89000 46.30000 19.2 44.62667
## 4 2016-01-11 17:40:00         60     40 19.89000 46.33333 19.2 44.53000
## 5 2016-01-11 17:50:00         50     40 19.89000 46.02667 19.2 44.50000
## 6 2016-01-11 18:10:00         60     50 19.85667 45.56000 19.2 44.50000
##      T3     RH_3       T4     RH_4       T5  RH_5       T6     RH_6
## 1 19.79 44.73000 19.00000 45.56667 17.16667 55.20 7.026667 84.25667
## 2 19.79 44.79000 19.00000 45.99250 17.16667 55.20 6.833333 84.06333
## 3 19.79 44.93333 18.92667 45.89000 17.16667 55.09 6.560000 83.15667
## 4 19.79 45.00000 18.89000 45.53000 17.20000 55.09 6.366667 84.89333
## 5 19.79 44.93333 18.89000 45.73000 17.13333 55.03 6.300000 85.76667
## 6 19.73 44.90000 18.89000 45.86333 17.10000 54.90 6.190000 86.42333
##         T7     RH_7   T8     RH_8       T9  RH_9    T_out Press_mm_hg
## 1 17.20000 41.62667 18.2 48.90000 17.03333 45.53 6.600000    733.5000
## 2 17.20000 41.56000 18.2 48.86333 17.06667 45.56 6.483333    733.6000
## 3 17.20000 41.43333 18.2 48.73000 17.00000 45.50 6.366667    733.7000
## 4 17.20000 41.23000 18.1 48.59000 17.00000 45.40 6.133333    733.9000
## 5 17.13333 41.26000 18.1 48.59000 17.00000 45.29 6.016667    734.0000
## 6 17.10000 41.20000 18.1 48.59000 17.00000 45.29 5.916667    734.1667
##     RH_out Windspeed Visibility Tdewpoint      rv1      rv2   NSM
## 1 92.00000  7.000000   63.00000  5.300000 13.27543 13.27543 61200
## 2 92.00000  6.666667   59.16667  5.200000 18.60619 18.60619 61800
## 3 92.00000  6.333333   55.33333  5.100000 28.64267 28.64267 62400
## 4 92.00000  5.666667   47.66667  4.900000 10.08410 10.08410 63600
## 5 92.00000  5.333333   43.83333  4.800000 44.91948 44.91948 64200
## 6 91.83333  5.166667   40.00000  4.683333 33.03989 33.03989 65400
```

```
##   WeekStatus Day_of_week
## 1    Weekday      Monday
## 2    Weekday      Monday
## 3    Weekday      Monday
## 4    Weekday      Monday
## 5    Weekday      Monday
## 6    Weekday      Monday
```

```r
# Visualizando os nomes das colunas
names(df)
```

```
##  [1] "date"        "Appliances"  "lights"      "T1"          "RH_1"
##  [6] "T2"          "RH_2"        "T3"          "RH_3"        "T4"
## [11] "RH_4"        "T5"          "RH_5"        "T6"          "RH_6"
## [16] "T7"          "RH_7"        "T8"          "RH_8"        "T9"
## [21] "RH_9"        "T_out"       "Press_mm_hg" "RH_out"      "Windspeed"
## [26] "Visibility"  "Tdewpoint"   "rv1"         "rv2"         "NSM"
## [31] "WeekStatus"  "Day_of_week"
```

```r
# Descricao das variáveis
```

```r
# date: tempo de coleta dos dados pelos sensores (year-month-day hour:minute)
# Appliances: uso de energia (em W)
# lights: potencia de energia de eletrodomesticos na casa (em W)
# TXX: Temperatura em um lugar da casa (em Celsius)
# RH_XX: umidade em um lugar da casa (em %)
# T_out: temperatura externa (em Celsius) in Celsius
# Pressure: pressão externa (em mm Hg)
# RH_out: umidade externa (em %)
# Wind speed: velocidade do vento (em m/s)
# Visibility; visibilidade (em km)
# Tdewpoint: nao descobri o que significa mas acredito que dados de algum sensor
# rv1: variavel randomica adicional
# rv2, variavel randomica adicional
# WeekStatus: indica se é dia de semana ou final de semana (weekend ou weekday)
# Day_of_week: dia da semana
# NSM: medida do tempo em segundos
```

```r
# 02 - Aplicando Engenharia de Atributos (Feature Engineering)
```

```r
# Transformar o objeto de data
df$date <- strptime(as.character(df$date),format="%Y-%m-%d %H:%M")
df$date <- as.POSIXct(df$date , tz="UTC")

# Extraindo ano, mes, dia, hora e minuto do campo data
df$day   <- as.integer(format(df$date, "%d"))
df$month <- as.factor(format(df$date, "%m"))
df$hour <- as.integer(format(df$date, "%H"))

# Transformando variáveis numéricas em variáveis categóricas
df$lights <- as.factor(df$lights)
```

```r
# 03 - Analise Exploratoria de Dados
```

```r
# Carregando os Pacotes
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(ggplot2)
library(PerformanceAnalytics)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```r
library(corrgram)
```

```
## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
##
## Attaching package: 'corrgram'
## The following object is masked from 'package:lattice':
##
##     panel.fill
```

```r
library(zoo)

# Verificar se existem valores ausentes (missing) em cada coluna
# Nenhum valor encontrado
any(is.na(df))
```

```
## [1] FALSE
```

```r
# Verificando os dados estatisticos das variaveis numericas
describe(df)
```

```
## df
##
##  35  Variables      19735  Observations
## ---------------------------------------------------------------------------
## date
##                    n             missing            distinct
##                19735                   0               19735
##                 Info                Mean                 Gmd
##                    1 2016-03-20 05:30:00 1970-02-15 16:26:40
##                  .05                 .10                 .25
## 2016-01-18 13:27:00 2016-01-25 09:54:00 2016-02-14 23:15:00
##                  .50                 .75                 .90
## 2016-03-20 05:30:00 2016-04-23 11:45:00 2016-05-14 01:06:00
##                  .95
## 2016-05-20 21:33:00
##
## lowest : 2016-01-11 17:00:00 2016-01-11 17:10:00 2016-01-11 17:20:00 2016-01-11 17:30:00 2016-01-11 
## highest: 2016-05-27 17:20:00 2016-05-27 17:30:00 2016-05-27 17:40:00 2016-05-27 17:50:00 2016-05-27 
## ---------------------------------------------------------------------------
## Appliances
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0       92    0.982    97.69    80.27       30       40
##      .25      .50      .75      .90      .95
##       50       60      100      196      330
##
## lowest :   10   20   30   40   50, highest:  890  900  910 1070 1080
## ---------------------------------------------------------------------------
## lights
##        n  missing distinct
##    19735        0        8
##
## Value          0     10     20     30     40     50     60     70
```

```
## Frequency  15252  2212  1624   559    77     9     1     1
## Proportion 0.773 0.112 0.082 0.028 0.004 0.000 0.000 0.000
## ---------------------------------------------------------------------------
## T1
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0      722        1    21.69    1.793    19.10    19.70
##      .25      .50      .75      .90      .95
##    20.76    21.60    22.60    23.96    24.73
##
## lowest : 16.79000 16.82333 16.85667 16.89000 16.96333
## highest: 26.06667 26.10000 26.16667 26.20000 26.26000
## ---------------------------------------------------------------------------
## RH_1
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     2547        1    40.26    4.474    34.70    35.47
##      .25      .50      .75      .90      .95
##    37.33    39.66    43.07    45.70    47.33
##
## lowest : 27.02333 27.23333 27.36000 27.43000 27.54500
## highest: 57.42333 57.49667 57.66333 59.63333 63.36000
## ---------------------------------------------------------------------------
## T2
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     1650        1    20.34    2.404    17.32    17.89
##      .25      .50      .75      .90      .95
##    18.79    20.00    21.50    23.33    24.56
##
## lowest : 16.10000 16.13333 16.20000 16.23000 16.26000
## highest: 29.53333 29.66333 29.66667 29.79000 29.85667
## ---------------------------------------------------------------------------
## RH_2
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     3376        1    40.42    4.533    33.43    35.32
##      .25      .50      .75      .90      .95
##    37.90    40.50    43.26    45.23    46.66
##
## lowest : 20.46333 20.59667 20.83333 20.89333 21.04000
## highest: 53.98498 54.09000 54.65667 54.76667 56.02667
## ---------------------------------------------------------------------------
## T3
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     1426        1    22.27    2.248    19.50    19.92
##      .25      .50      .75      .90      .95
##    20.79    22.10    23.29    25.10    26.20
##
## lowest : 17.20000 17.23000 17.26000 17.29000 17.31500
## highest: 29.10000 29.16000 29.19857 29.20000 29.23600
## ---------------------------------------------------------------------------
## RH_3
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     2618        1    39.24    3.673    34.76    35.40
##      .25      .50      .75      .90      .95
##    36.90    38.53    41.76    44.36    45.09
##
```

```
## lowest : 28.76667 28.86000 29.00000 29.29333 29.49333
## highest: 49.65667 49.80000 49.93000 50.09000 50.16333
## ---------------------------------------------------------------------------
## T4
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     1390        1    20.86    2.292    17.79    18.50
##      .25      .50      .75      .90      .95
##    19.53    20.67    22.10    23.79    24.50
##
## lowest : 15.10000 15.13000 15.16000 15.19000 15.22667
## highest: 26.12857 26.14000 26.14286 26.18000 26.20000
## ---------------------------------------------------------------------------
## RH_4
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     2987        1    39.03     4.93    33.00    33.76
##      .25      .50      .75      .90      .95
##    35.53    38.40    42.16    45.50    46.79
##
## lowest : 27.66000 28.13571 28.42429 28.71600 28.77800
## highest: 50.93000 50.96333 51.00000 51.06333 51.09000
## ---------------------------------------------------------------------------
## T5
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     2263        1    19.59    2.053    17.10    17.50
##      .25      .50      .75      .90      .95
##    18.28    19.39    20.62    22.48    23.39
##
## lowest : 15.33000 15.33500 15.34000 15.34500 15.35000
## highest: 25.46667 25.63333 25.70000 25.74500 25.79500
## ---------------------------------------------------------------------------
## RH_5
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     7571        1    50.95    8.986    40.79    42.59
##      .25      .50      .75      .90      .95
##    45.40    49.09    53.66    60.88    70.39
##
## lowest : 29.81500 29.85667 30.03000 30.03333 30.16667
## highest: 95.38833 95.60889 95.80222 95.95389 96.32167
## ---------------------------------------------------------------------------
## T6
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     4446        1    7.911    6.761  -0.6667   0.8333
##      .25      .50      .75      .90      .95
##   3.6267   7.3000  11.2560  16.0580  19.5512
##
## lowest : -6.065000 -6.030000 -6.028333 -6.010000 -6.000000
## highest: 28.150000 28.200000 28.218000 28.236000 28.290000
## ---------------------------------------------------------------------------
## RH_6
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     9709        1    54.61    35.77    1.000    5.291
##      .25      .50      .75      .90      .95
##   30.025   55.290   83.227   94.467   98.590
##
```

```
## lowest :  1.000000  1.020000  1.033333  1.040000  1.042857
## highest: 99.800000 99.830000 99.833333 99.866667 99.900000
## ---------------------------------------------------------------------------
## T7
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     1955        1    20.27    2.394    17.01    17.79
##      .25      .50      .75      .90      .95
##    18.70    20.03    21.60    23.39    24.08
##
## lowest : 15.39000 15.39611 15.40833 15.41444 15.42056
## highest: 25.82333 25.89000 25.92667 25.96333 26.00000
## ---------------------------------------------------------------------------
## RH_7
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     5891        1    35.39    5.833    27.70    29.03
##      .25      .50      .75      .90      .95
##    31.50    34.86    39.00    42.59    44.20
##
## lowest : 23.20000 23.23000 23.26000 23.29000 23.32333
## highest: 51.15111 51.17556 51.19778 51.32778 51.40000
## ---------------------------------------------------------------------------
## T8
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     2228        1    22.03    2.208    18.36    19.39
##      .25      .50      .75      .90      .95
##    20.79    22.10    23.39    24.46    25.10
##
## lowest : 16.30667 16.36222 16.36778 16.37333 16.38444
## highest: 27.06667 27.10000 27.13333 27.20000 27.23000
## ---------------------------------------------------------------------------
## RH_8
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     6649        1    42.94    5.936    35.26    36.63
##      .25      .50      .75      .90      .95
##    39.07    42.38    46.54    50.37    52.07
##
## lowest : 29.60000 29.67500 29.70000 29.72667 29.79000
## highest: 58.67556 58.70278 58.73000 58.74000 58.78000
## ---------------------------------------------------------------------------
## T9
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0      924        1    19.49    2.269    16.39    17.10
##      .25      .50      .75      .90      .95
##    18.00    19.39    20.60    22.70    23.20
##
## lowest : 14.89000 14.96333 15.00000 15.02500 15.03333
## highest: 24.43400 24.43714 24.45286 24.45600 24.50000
## ---------------------------------------------------------------------------
## RH_9
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     3388        1    41.55    4.686    35.70    36.85
##      .25      .50      .75      .90      .95
##    38.50    40.90    44.34    47.74    49.05
##
```

```
## lowest : 29.16667 29.20000 29.23000 29.29000 29.35667
## highest: 53.00000 53.09000 53.16333 53.22333 53.32667
## ---------------------------------------------------------------------------
## T_out
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     1730        1    7.412    5.926  -0.3000   0.9733
##      .25      .50      .75      .90      .95
##   3.6667   6.9167  10.4083  14.5500  17.1000
##
## lowest : -5.000000 -4.988889 -4.977778 -4.966667 -4.955556
## highest: 25.833333 25.900000 25.966667 26.033333 26.100000
## ---------------------------------------------------------------------------
## Press_mm_hg
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     2189        1    755.5    8.327    742.2    745.8
##      .25      .50      .75      .90      .95
##    750.9    756.1    760.9    764.8    766.6
##
## lowest : 729.3000 729.3333 729.3667 729.4000 729.4333
## highest: 772.2333 772.2500 772.2667 772.2833 772.3000
## ---------------------------------------------------------------------------
## RH_out
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0      566        1    79.75    16.39    51.00    57.33
##      .25      .50      .75      .90      .95
##    70.33    83.67    91.67    95.67    97.00
##
## lowest :  24.00000  24.50000  25.00000  25.16667  25.33333
## highest:  99.66667  99.75000  99.83333  99.91667 100.00000
## ---------------------------------------------------------------------------
## Windspeed
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0      189    0.996     4.04    2.708    1.000    1.000
##      .25      .50      .75      .90      .95
##    2.000    3.667    5.500    7.667    9.000
##
## lowest :  0.0000000  0.1666667  0.3333333  0.5000000  0.6666667
## highest: 12.6666667 12.8333333 13.0000000 13.5000000 14.0000000
## ---------------------------------------------------------------------------
## Visibility
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0      413    0.951    38.33     12.7    21.67    23.67
##      .25      .50      .75      .90      .95
##    29.00    40.00    40.00    59.27    62.67
##
## lowest :  1.000000  1.166667  1.333333  1.500000  1.666667
## highest: 64.500000 64.666667 64.833333 65.000000 66.000000
## ---------------------------------------------------------------------------
## Tdewpoint
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0     1409        1    3.761    4.729   -3.100   -1.350
##      .25      .50      .75      .90      .95
##    0.900    3.433    6.567    9.200   11.233
##
```

```
## lowest : -6.600000 -6.550000 -6.516667 -6.500000 -6.483333
## highest: 15.200000 15.300000 15.316667 15.400000 15.500000
## ---------------------------------------------------------------------------
## rv1
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0    19735        1    24.99    16.74    2.434    4.901
##      .25      .50      .75      .90      .95
##   12.498   24.898   37.584   45.156   47.534
##
## lowest :  0.005321682  0.006032793  0.010019040  0.013538753  0.013988744
## highest: 49.981673900 49.991010746 49.992758071 49.993172963 49.996529683
## ---------------------------------------------------------------------------
## rv2
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0    19735        1    24.99    16.74    2.434    4.901
##      .25      .50      .75      .90      .95
##   12.498   24.898   37.584   45.156   47.534
##
## lowest :  0.005321682  0.006032793  0.010019040  0.013538753  0.013988744
## highest: 49.981673900 49.991010746 49.992758071 49.993172963 49.996529683
## ---------------------------------------------------------------------------
## NSM
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0      144        1    42907    28798     4200     8400
##      .25      .50      .75      .90      .95
##    21600    43200    64200    77400    81600
##
## lowest :     0   600  1200  1800  2400, highest: 83400 84000 84600 85200 85800
## ---------------------------------------------------------------------------
## WeekStatus
##        n  missing distinct
##    19735        0        2
##
## Value      Weekday Weekend
## Frequency    14263    5472
## Proportion   0.723   0.277
## ---------------------------------------------------------------------------
## Day_of_week
##        n  missing distinct
##    19735        0        7
##
## Value         Friday    Monday  Saturday    Sunday  Thursday   Tuesday
## Frequency       2845      2778      2736      2736      2880      2880
## Proportion     0.144     0.141     0.139     0.139     0.146     0.146
##
## Value      Wednesday
## Frequency       2880
## Proportion     0.146
## ---------------------------------------------------------------------------
## day
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0       31    0.999    16.06    9.744        2        4
##      .25      .50      .75      .90      .95
##        9       16       23       27       29
```

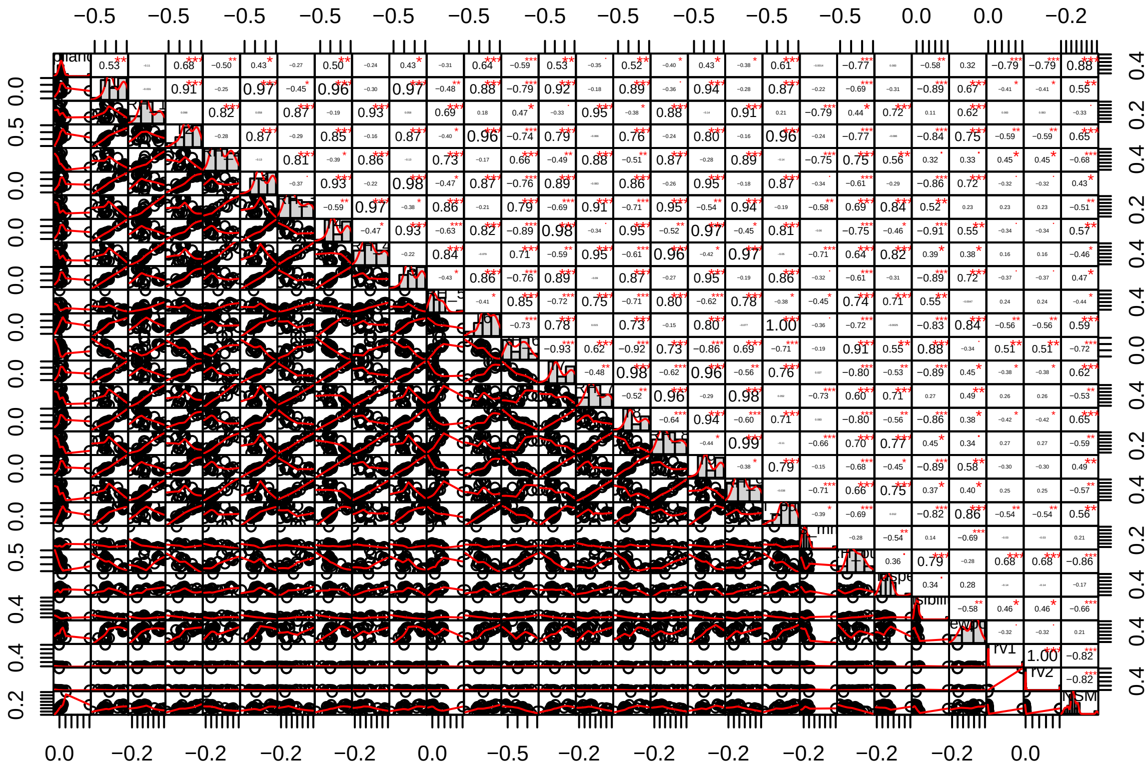
```
##
## lowest :  1  2  3  4  5, highest: 27 28 29 30 31
## ---------------------------------------------------------------------------
## month
##        n  missing distinct
##    19735        0        5
##
## Value          1     2     3     4     5
## Frequency   2922  4176  4464  4320  3853
## Proportion 0.148 0.212 0.226 0.219 0.195
## ---------------------------------------------------------------------------
## hour
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    19735        0       24    0.998     11.5    7.986        1        2
##      .25      .50      .75      .90      .95
##        6       12       17       21       22
##
## lowest :  0  1  2  3  4, highest: 19 20 21 22 23
## ---------------------------------------------------------------------------
```

```r
# Observacoes Estatisticas
```

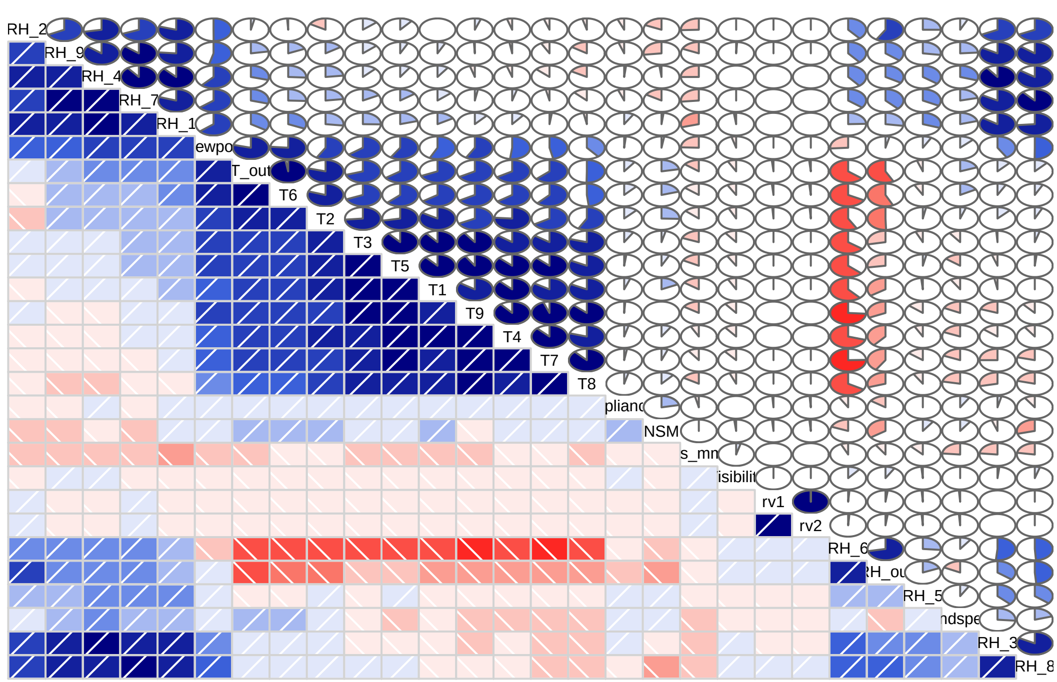
```r
# Temperaturas internas: variacao entre 14.89 a 29.95 graus Celsius
# Temperaturas externas (T6 e T_out): variacao entre -6.06 a 28.29 graus Celsius.

# Umidade interna: variacao entre 20.60% a 63.36% com excecao do RH_5
# Umidade externa (RH_6 e  RH_out): variacao entre 1% to 100%

# Energia: 75% do consumo de energia é menor que 100W
# O maior consumo é de 1080W o que representa um outlier no dataset

# Lights: 15.252 valores 0 (zero) em 19.735 observacoes
# Verificar se tem significancia para performance do modelo

# WeekStatus: 72,3% das observacoes foram durante a semana e 27,7% nos finais de semana


# Analise de Correlacao
```

```r
# Separando as colunas numericas para correlacao
numeric.vars <- c('Appliances','T1','RH_1','T2','RH_2','T3','RH_3','T4','RH_4','T5','RH_5','T6','RH_6',
                  'T_out','Press_mm_hg','RH_out','Windspeed','Visibility','Tdewpoint','rv1','rv2','NSM'
data_cor <- cor(df[,numeric.vars])

# Visualizando a correlacao usando metodo 'spearman'
# É uma visualizacao completa entre as variaveis
chart.Correlation(data_cor,
                  method="spearman",
                  histogram=TRUE,
                  pch=16)
```

```
## Warning in cor.test.default(as.numeric(x), as.numeric(y), method = method):
## Cannot compute exact p-value with ties

## Warning in cor.test.default(as.numeric(x), as.numeric(y), method = method):
## Cannot compute exact p-value with ties
```



```r
# Visualizando um corrgram
# É uma visualizacao mais confortavel, porem com interpretacao dos dados numericos
corrgram(data_cor, order=TRUE, lower.panel = panel.shade,
         upper.panel = panel.pie, text.panel = panel.txt)
```

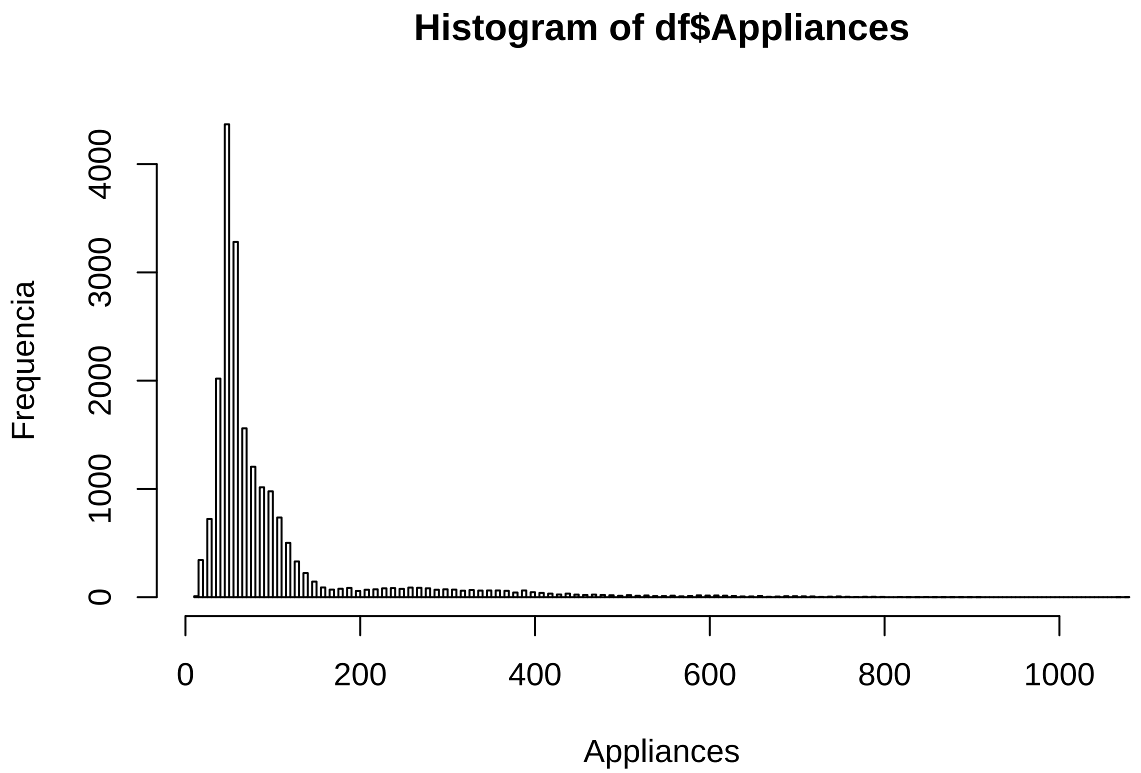


```r
# Observacoes da correlacao
```

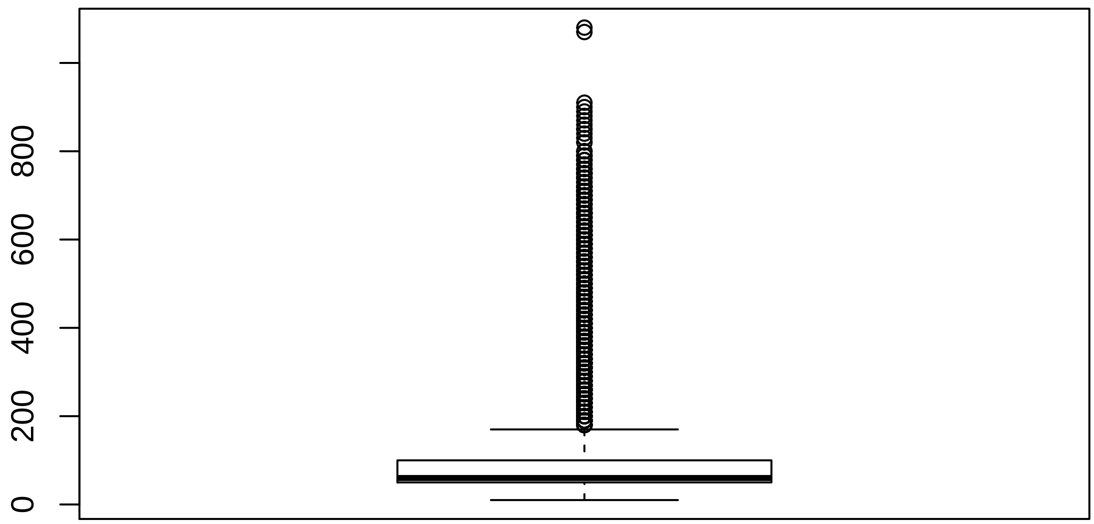
```r
# Temperaturas: todas essas features tem correlacao positiva com o target "Appliances"
# Atributos do Tempo: Visibility, Tdewpoint, Press_mm_hg tem correlacao baixa
```

```r
# Umidade: nao tem correlacao significante (> 0.9 por exemplo)
# Variaveis Randomicas: sem influencia


# Avaliando a variavel target "Appliances"
```

```r
hist(df$Appliances, "FD", xlab="Appliances", ylab="Frequencia")
```




```r
# Verificar a quantidade outliers para o consumo de energia
boxplot(df$Appliances)
```



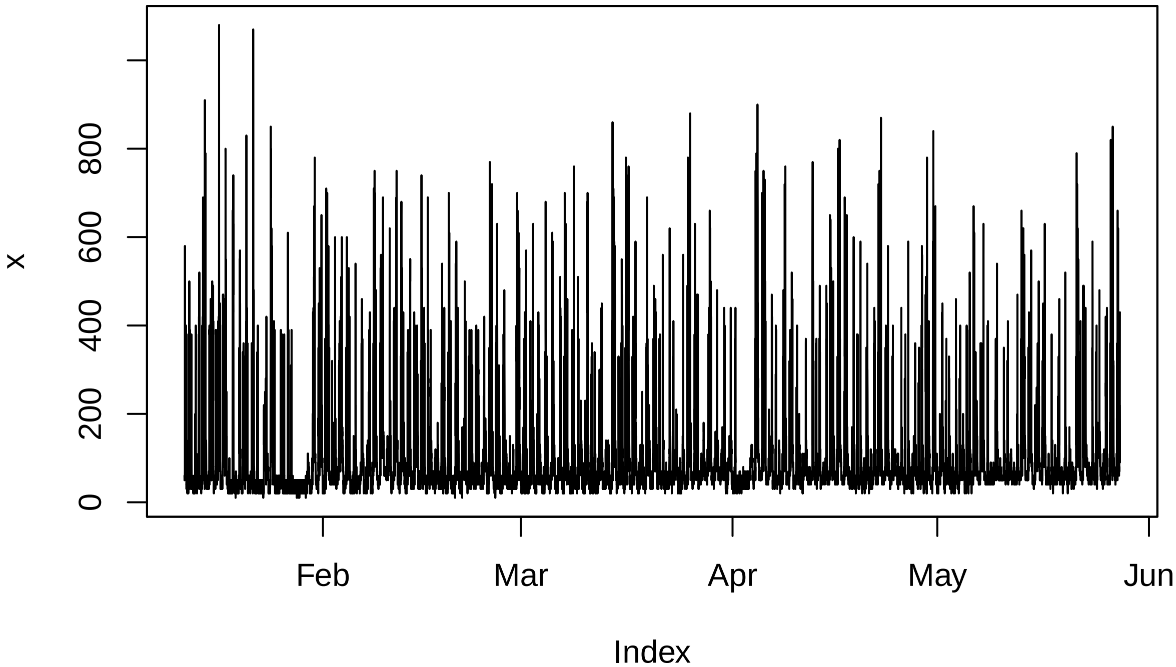

```r
outliers <- boxplot(df$Appliances, plot=FALSE)$out
head(outliers)
```

```
## [1] 430 250 400 400 390 240
```

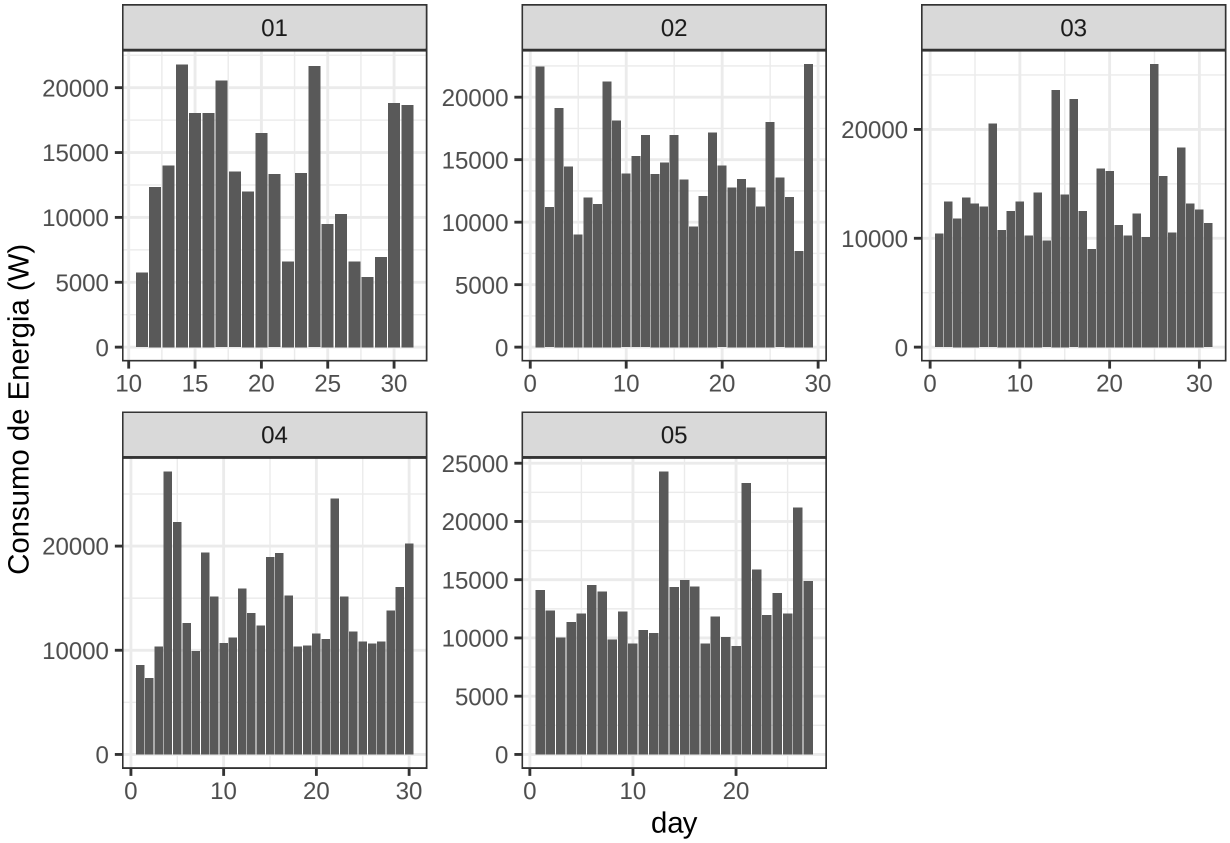
```r
# Observacoes da variavel target
```

```r
# 1. esta coluna esta com skewed positivo
# 2. muitos valores estao com media de 100W (75% dos dados)
# 3. existem outliers (neste caso 2.138 registros sao outliers)


# Visualizando alguns plots para analise
```

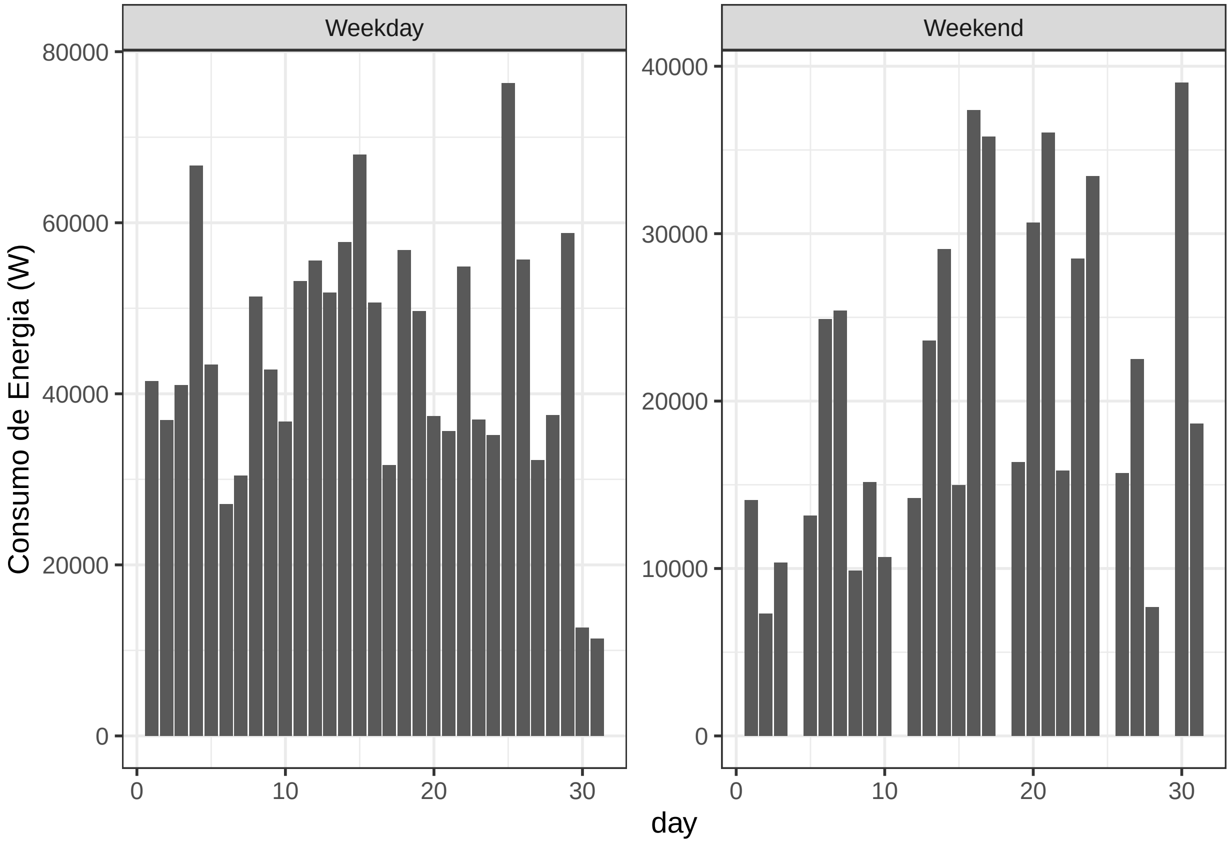
```r
# Analise de Serie Temporal
x <- zoo(df$Appliances, df$date)
plot(x)
```




```r
# Visualizando o consumo de energia por dia x mes
ggplot(df)+
  geom_bar(aes(x=day, y=Appliances), stat="identity")+
  scale_y_continuous(name="Consumo de Energia (W)")+
  facet_wrap(~month, scale="free")+
  theme_bw()
```




```r
# Visualizando o consumo de energia por dia x semana e final de semana
ggplot(df)+
  geom_bar(aes(x=day, y=Appliances), stat="identity")+
  scale_y_continuous(name="Consumo de Energia (W)")+
  facet_wrap(~WeekStatus, scale="free")+
  theme_bw()
```




```
# Observacoes dos graficos
```

```
# 1. pelos graficos apresentados, conseguimos observar um pico de consumo no mes de janeiro
#    e alguns periodos com baixa frequencia de consumo (final de janeiro e inicio de abril principalmen
# 2. o consumo de energia do mes de marco, abril e maio é menor que os meses de janeiro e fevereiro
#    pode ser um periodo de ferias ou devido ao verao
```

```
# 04 - Feature Selection (Selecao de Variaveis)
```

```
# Carregando os Pacotes
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(scales)

# Normalizando as variáveis numericas
scale.features <- function(df, variables){
  for (variable in variables){
    df[[variable]] <- scale(df[[variable]], center=T, scale=T)
  }
  return(df)
}
```

```r
# Definindo as variaveis que serao normalizadas
numeric.vars <- c('T1','RH_1','T2','RH_2','T3','RH_3','T4','RH_4','T5','RH_5','T6','RH_6','T7','RH_7','
                  'T_out','Press_mm_hg','RH_out','Windspeed','Visibility','Tdewpoint','rv1','rv2','NSM')
df <- scale.features(df, numeric.vars)

# Transformando o campo data para index
rownames(df) <- df$date
df$date <- NULL

# Gerando dados de treino e de teste
```

```r
splits <- createDataPartition(df$Appliances, p=0.7, list=FALSE)

# Separando os dados de treino e teste
dados_treino <- df[ splits,]
dados_teste <- df[-splits,]

# Verificando o numero de linhas
nrow(dados_treino)
```

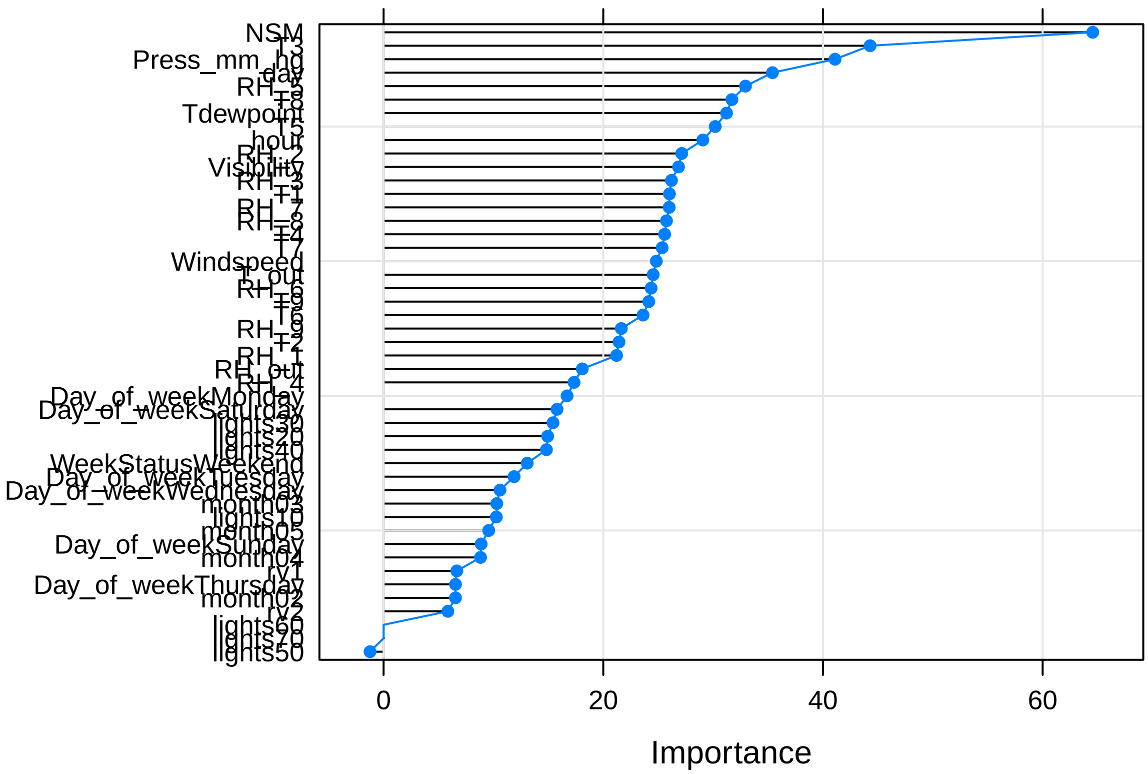
```
## [1] 13817
```

```r
nrow(dados_teste)
```

```
## [1] 5918
```

```r
# Verificando as features mais importantes usando RandomForest
```

```r
formula <- "Appliances ~ ."
formula <- as.formula(formula)
control <- trainControl(method = "repeatedcv", number = 3, repeats = 2)
result <- train(formula, data = dados_treino, method = "rf", trControl = control, importance=T)
importance <- varImp(result, scale = FALSE)

# Plot do resultado
plot(importance, type=c("g", "o"))
```




```
# Observacoes da Feature Importance
```

```
# NSM: tem uma grande importancia dentro do dataset
# Features de Temperaturas: todas as features de temperatura tem importancia entre 20 a 40%
# Features de Umidade: todas as features de umidade tem importancia entre 20 e 40% (com excecao da RH_o
# Features de Tempo: tambem estao dentro do quadrante de 20-40%
# Essas sera as variaveis selecionadas para a criaçao dos modelos:
#   NSM, T1...T9, T_out, RH_1...RH_9, RH_out, Press_mm_hg, Tdewpoint, Visibility, Windspeed, day, hour
```

```
# 05 - Criando alguns modelos de ML para comparacoes
```

```
# Definindo a formula com as features selecionadas
formula <- "Appliances ~ NSM+
                         Press_mm_hg+
                         T1+T2+T3+T4+T5+T6+T7+T8+T9+
                         RH_1+RH_2+RH_3+RH_4+RH_5+RH_6+RH_7+RH_8+RH_9+
                         T_out+RH_out+
                         day+hour"

formula <- as.formula(formula)


# Construindo um modelo Multiple Logistic Regression (GLM)
```

```
controlGLM <- trainControl(method="cv", number=5)
modeloGLM <- train(formula, data = dados_treino, method = "glm", metric="Rsquared", trControl=controlGL

# Resumo do Modelo Linear Model
print(modeloGLM)
```

```
## Generalized Linear Model
```

```
##
## 13817 samples
##    24 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 11053, 11055, 11053, 11053, 11054
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   96.00066  0.1438915  54.68165
```

```r
# Construindo um modelo Generalized Boosted Regression Modeling (GBM)
```

```r
controlGBM <- trainControl(method="cv", number=5)
modeloGBM <- train(formula, data=dados_treino, method="gbm", verbose=FALSE, metric="Rsquared", trContro

# Resumo do Modelo GBM
print(modeloGBM)
```

```
## Stochastic Gradient Boosting
##
## 13817 samples
##    24 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 11054, 11054, 11054, 11054, 11052
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  RMSE      Rsquared   MAE
##   1                   50      96.68703  0.1432909  53.14604
##   1                  100      95.35733  0.1636502  52.37173
##   1                  150      94.51843  0.1771723  51.72911
##   2                   50      94.12331  0.1906691  51.35790
##   2                  100      92.12148  0.2194044  50.19483
##   2                  150      90.96210  0.2367726  49.43090
##   3                   50      92.45626  0.2197256  50.17672
##   3                  100      89.95823  0.2567462  48.53453
##   3                  150      88.61865  0.2758728  47.66602
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Rsquared was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```r
# Construindo um modelo eXtreme Gradient Boosting (XGBoost)
```

```r
controlXGB <- trainControl(method = "cv", number = 5)
modeloXGB <- train(formula, data=dados_treino, method="xgbLinear", trControl=controlXGB)

# Resumo do Modelo GBM
print(modeloXGB)
```

```
## eXtreme Gradient Boosting
##
## 13817 samples
##    24 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 11054, 11053, 11053, 11054, 11054
## Resampling results across tuning parameters:
##
##   lambda  alpha  nrounds  RMSE      Rsquared   MAE
##   0e+00   0e+00   50      79.46318  0.4150564  40.19958
##   0e+00   0e+00  100      77.36230  0.4487753  38.59569
##   0e+00   0e+00  150      76.76012  0.4596759  38.01187
##   0e+00   1e-04   50      79.46290  0.4150605  40.19912
##   0e+00   1e-04  100      77.36200  0.4487794  38.59536
##   0e+00   1e-04  150      76.75993  0.4596782  38.01166
##   0e+00   1e-01   50      79.45968  0.4153290  40.18864
##   0e+00   1e-01  100      77.70760  0.4438743  38.70710
##   0e+00   1e-01  150      76.87201  0.4585540  38.04744
##   1e-04   0e+00   50      79.36383  0.4166988  40.12530
##   1e-04   0e+00  100      77.38057  0.4484493  38.53766
##   1e-04   0e+00  150      76.83131  0.4591003  37.97009
##   1e-04   1e-04   50      79.36383  0.4166988  40.12530
##   1e-04   1e-04  100      77.38057  0.4484493  38.53766
##   1e-04   1e-04  150      76.83129  0.4591005  37.97009
##   1e-04   1e-01   50      79.46037  0.4153154  40.18928
##   1e-04   1e-01  100      77.74161  0.4439178  38.75822
##   1e-04   1e-01  150      76.90297  0.4581150  38.11395
##   1e-01   0e+00   50      79.63768  0.4130439  40.42224
##   1e-01   0e+00  100      77.90458  0.4425535  38.92804
##   1e-01   0e+00  150      77.16231  0.4553340  38.26466
##   1e-01   1e-04   50      79.63768  0.4130439  40.42224
##   1e-01   1e-04  100      77.90458  0.4425535  38.92804
##   1e-01   1e-04  150      77.16230  0.4553340  38.26466
##   1e-01   1e-01   50      79.63708  0.4131893  40.49439
##   1e-01   1e-01  100      77.81556  0.4432951  38.92154
##   1e-01   1e-01  150      76.94751  0.4574995  38.22581
##
## 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,
##  alpha = 1e-04 and eta = 0.3.
```

```r
# Observacoes da Performance dos modelos (dados de treino)
```

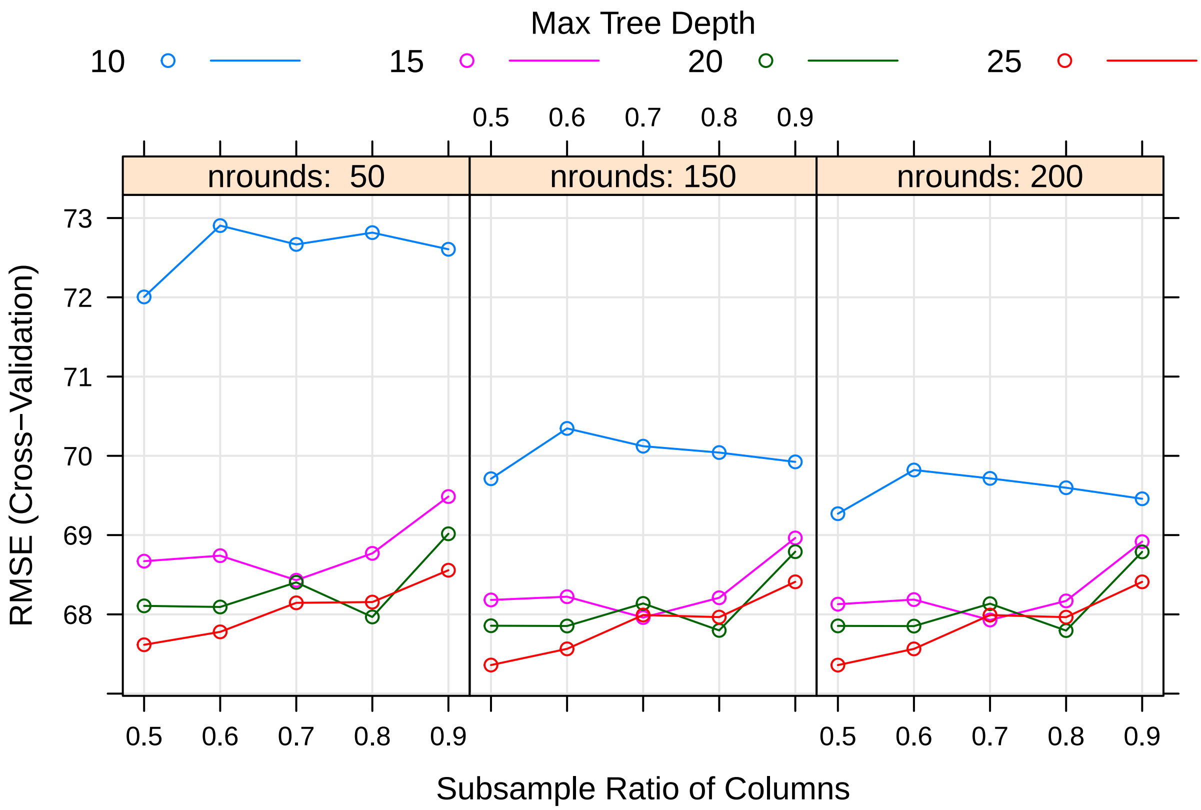
```r
# Multiple Logistic Regression (GLM): RMSE=93.63 e R_squared=0.14
# Generalized Boosted Regression Modeling (GBM): RMSE=85.79 e R_squared=0.28
# eXtreme Gradient Boosting (XGBoost): RMSE=73.70 e R_squared=0.47
```

```r
# 06 - Otimizando o modelo eXtreme Gradient Boosting
```

```r
# Carregando os Pacotes
library(xgboost)
```

```
##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
# Definindo os parametros de controle do modelo
xgb_trcontrol = trainControl(
  method = "cv",
  number = 5,
  allowParallel = TRUE,
  verboseIter = FALSE,
  returnData = FALSE
)

# Modificando os hyperparametros usando o gridSearch
xgbGrid <- expand.grid(nrounds = c(50, 150, 200),
                       max_depth = c(10, 15, 20, 25),
                       colsample_bytree = seq(0.5, 0.9, length.out = 5),
                       eta = 0.1,
                       gamma=0,
                       min_child_weight = 1,
                       subsample = 1
)

# Treinando o modelo otimizado
xgb_model <- train(formula, data=dados_treino, method="xgbTree", trControl=xgb_trcontrol, tuneGrid=xgbGr

# Visualizando a comparacao dos hyperparametros
plot(xgb_model)
```




```r
# Verificando os melhores parametros do modelo
xgb_model$bestTune
```

```
##    nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 48     200        25 0.1     0              0.5                1         1
```

```r
# Treinando o modelo com os melhores parametros
xgb_model <- train(formula, data=dados_treino, method="xgbTree", trControl=xgb_trcontrol, tuneGrid=xgb_m

# Resultado do modelo
print(xgb_model)
```

```
## eXtreme Gradient Boosting
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 11054, 11054, 11053, 11054, 11053
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   67.32344  0.5789    30.46626
##
## Tuning parameter 'nrounds' was held constant at a value of 200
##  0.5
## Tuning parameter 'min_child_weight' was held constant at a value of
##  1
## Tuning parameter 'subsample' was held constant at a value of 1
```

```r
# Observacoes da Performance no modelo XGBoost Otimizado (dados de treino)
```

```r
# RMSE=65.38 e R_squared=0.58
```

```r
# 07 - Avaliando o modelo XGBoost Otimizado nos dados de teste
```

```r
# Calculando o RMSE
predicted = predict(xgb_model, dados_teste)
residuals = dados_teste$Appliances - predicted
RMSE = sqrt(mean(residuals^2))
cat('O RMSE nos dados de teste é: ', round(RMSE,3),'\n')
```

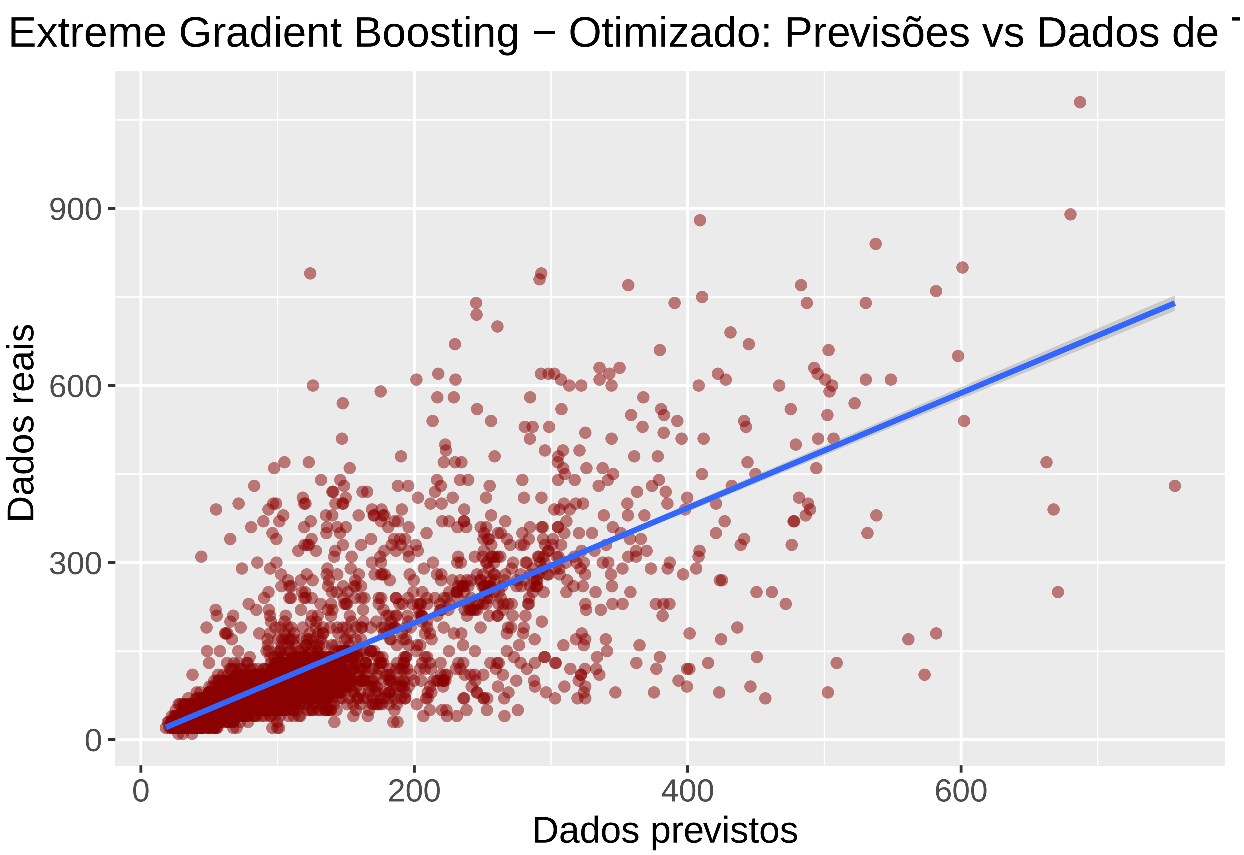
```
## O RMSE nos dados de teste é:  61.984
```

```r
# Calculando o R-square
y_test_mean = mean(dados_teste$Appliances)
tss =  sum((dados_teste$Appliances - y_test_mean)^2 )
rss =  sum(residuals^2)
rsq  =  1 - (rss/tss)
cat('O R-square nos dados de teste é: ', round(rsq,3), '\n')
```

```
## O R-square nos dados de teste é:  0.612
```

```r
# Visualizando as previsoes do modelo
```

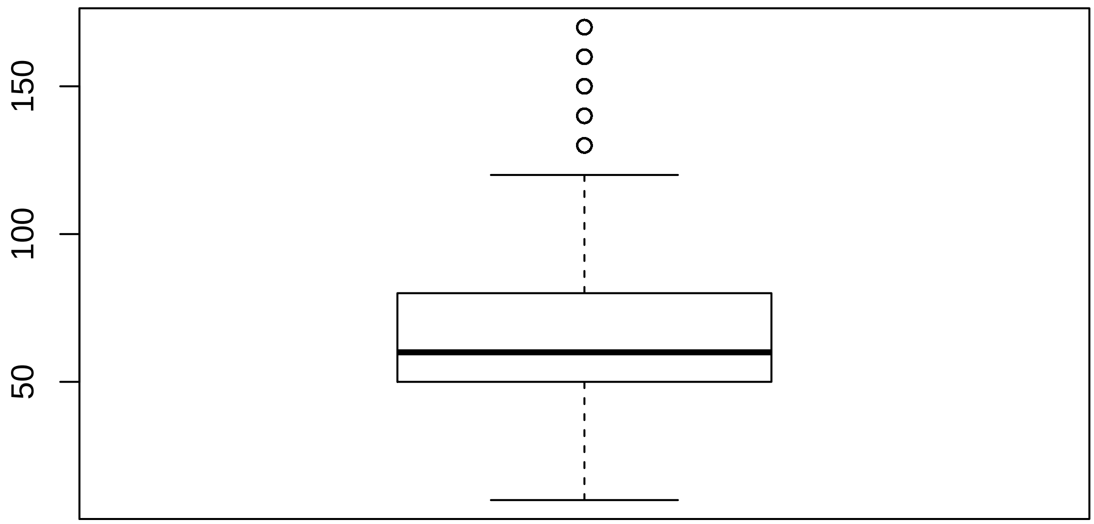
```r
options(repr.plot.width=8, repr.plot.height=4)

dfPrevisoes = as.data.frame(cbind(predicted = predicted,
                                  observed = dados_teste$Appliances))

# Plot previsoes vs dados de teste
ggplot(dfPrevisoes,aes(predicted, observed)) +
  geom_point(color = "darkred", alpha = 0.5) +
  geom_smooth(method=lm) +
  ggtitle('Linear Regression ') +
  ggtitle("Extreme Gradient Boosting - Otimizado: Previsões vs Dados de Teste") +
  xlab("Dados previstos ") +
  ylab("Dados reais ") +
  theme(plot.title = element_text(color="black",size=16,hjust = 0.5),
        axis.text.y = element_text(size=12), axis.text.x = element_text(size=12,hjust=.5),
        axis.title.x = element_text(size=14), axis.title.y = element_text(size=14))
```




```
# Observacoes Finais da primeira versao
```

```
# Na primeira versao selecionei todas as variaveis do dataset
# mas com alguns testes usando variaveis visualizadas pelo Feature Importance usando RandomForest
# obtive obtive a melhor acuracia no modelo XGBoost
# Usei ele para otimizar e obter um resultado de:
# Dados de Treino: RMSE 65.38 e R_square 0.58
# Dados de Teste : RMSE 65.81 e R_square 0.61


# Realizando o treinamento removendo os outliers da variavel target 'Appliances'
```

```
# Conforme identificado na analise exploratoria, 2.138 registros sao outliers
# Vamos remover esses 10% e analisar a performance do modelo
df2 <- rbind(dfTrain,dfTest)
df2 <- df2[-which(df2$Appliances %in% outliers),]
boxplot(df2$Appliances)
```




```r
# Realizar as transformacoes na feature date
df2$date <- strptime(as.character(df2$date),format="%Y-%m-%d %H:%M")
df2$date <- as.POSIXct(df2$date , tz="UTC")
df2$day   <- as.integer(format(df2$date, "%d"))
df2$month <- as.factor(format(df2$date, "%m"))
df2$hour <- as.integer(format(df2$date, "%H"))
# Transformando variáveis numéricas em variáveis categóricas
df2$lights <- as.factor(df2$lights)
# Aplicando a mesma escala nos dados numericos
df2 <- scale.features(df2, numeric.vars)
# Transformando o campo data para index
rownames(df2) <- df2$date
df2$date <- NULL

# Gerando dados de treino e de teste
splits2 <- createDataPartition(df2$Appliances, p=0.7, list=FALSE)
dados_treino2 <- df[ splits2,]
dados_teste2 <- df[-splits2,]

# Treinando o modelo com os melhores parametros
xgb_model2 <- train(formula, data=dados_treino2, method="xgbTree", trControl=xgb_trcontrol, tuneGrid=xgb

# Resultado do modelo
print(xgb_model2)
```

```
## eXtreme Gradient Boosting
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 9856, 9855, 9856, 9857, 9856
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   68.91773  0.5548355  31.5179
##
## Tuning parameter 'nrounds' was held constant at a value of 200
##  0.5
```

```
## Tuning parameter 'min_child_weight' was held constant at a value of
##  1
## Tuning parameter 'subsample' was held constant at a value of 1
```

```
# Observacoes da Performance no modelo XGBoost Otimizado (dados de treino s/ outliers)
```

```
# RMSE=69.01 e R_squared=0.55


# Conclusao Final
```

```
# O melhor algoritmo para esse dataset é o eXtreme Gradient Boosting
# O modelo otimizado e sem tratamento de outliers foi capaz de explicar 61% da variancia nos dados de t
# Realizando a remocao de outliers no dataset, nao houve melhora significativa na performance do modelo
# O ideal agora seria obter mais dados para aumentar a performance do modelo
```