Trabajo Final

Mauricio Lucero

2023-06-20

```
library(readr)
library(DataExplorer)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v stringr   1.5.0
## v forcats   1.0.0     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)
library(rsample)
library(parsnip)
library(recipes)
```

```
##
## Attaching package: 'recipes'
##
## The following object is masked from 'package:stringr':
##
##     fixed
##
## The following object is masked from 'package:stats':
##
##     step
```

```
library(workflows)
library(yardstick)
```

```
##
## Attaching package: 'yardstick'
##
## The following object is masked from 'package:readr':
##
##     spec
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following objects are masked from 'package:yardstick':
##
##     precision, recall, sensitivity, specificity
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(tensorflow)
```

```
##
## Attaching package: 'tensorflow'
##
## The following object is masked from 'package:caret':
##
##     train
```

```
library(keras)
```

```
##
## Attaching package: 'keras'
##
## The following object is masked from 'package:yardstick':
##
##     get_weights
```

```
library(reticulate)
library(nnet)
library(neuralnet)
```

```
##
## Attaching package: 'neuralnet'
##
## The following object is masked from 'package:dplyr':
##
##     compute
```

```
library(stargazer)
```

```
##
## Please cite as:
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
library(broom)
library(ggplot2)
library(modelr)
```

```
##
## Attaching package: 'modelr'
```

```
##
## The following object is masked from 'package:broom':
##
##     bootstrap
##
## The following objects are masked from 'package:yardstick':
##
##     mae, mape, rmse
```

```
library(MASS)
```

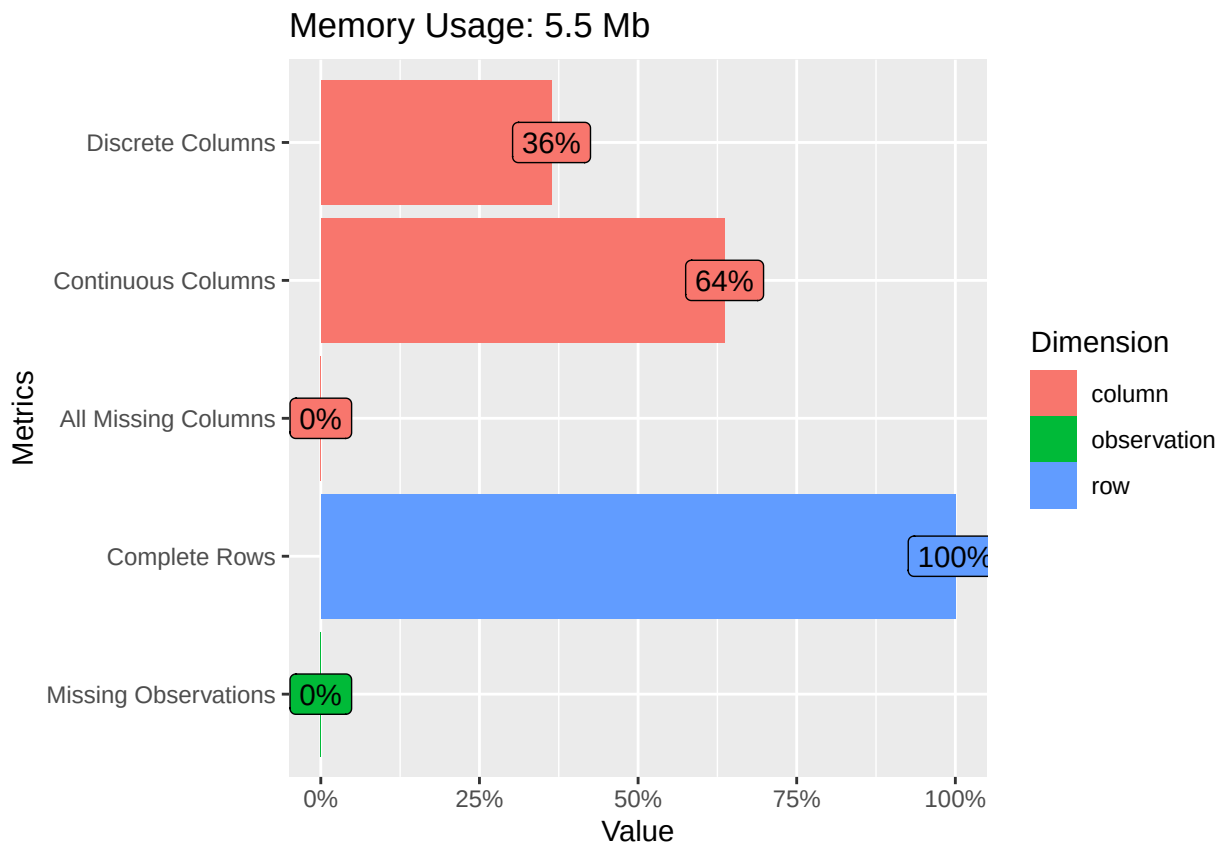
```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
set.seed(163)
```

```
df <-read.csv("Steel_industry_data.csv")
glimpse(df)
```

```
## Rows: 35,040
## Columns: 11
## $ date                                <chr> "01/01/2018 00:15", "01/01/2018 0~
## $ Usage_kWh                           <dbl> 3.17, 4.00, 3.24, 3.31, 3.82, 3.2~
## $ Lagging_Current_Reactive.Power_kVarh <dbl> 2.95, 4.46, 3.28, 3.56, 4.50, 3.5~
## $ Leading_Current_Reactive_Power_kVarh <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ CO2.tCO2.                           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ Lagging_Current_Power_Factor        <dbl> 73.21, 66.77, 70.28, 68.09, 64.72~
## $ Leading_Current_Power_Factor        <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ NSM                                 <int> 900, 1800, 2700, 3600, 4500, 5400~
## $ WeekStatus                          <chr> "Weekday", "Weekday", "Weekday", ~
## $ Day_of_week                         <chr> "Monday", "Monday", "Monday", "Mo~
## $ Load_Type                           <chr> "Light_Load", "Light_Load", "Ligh~
```

```
plot_intro(df)
```

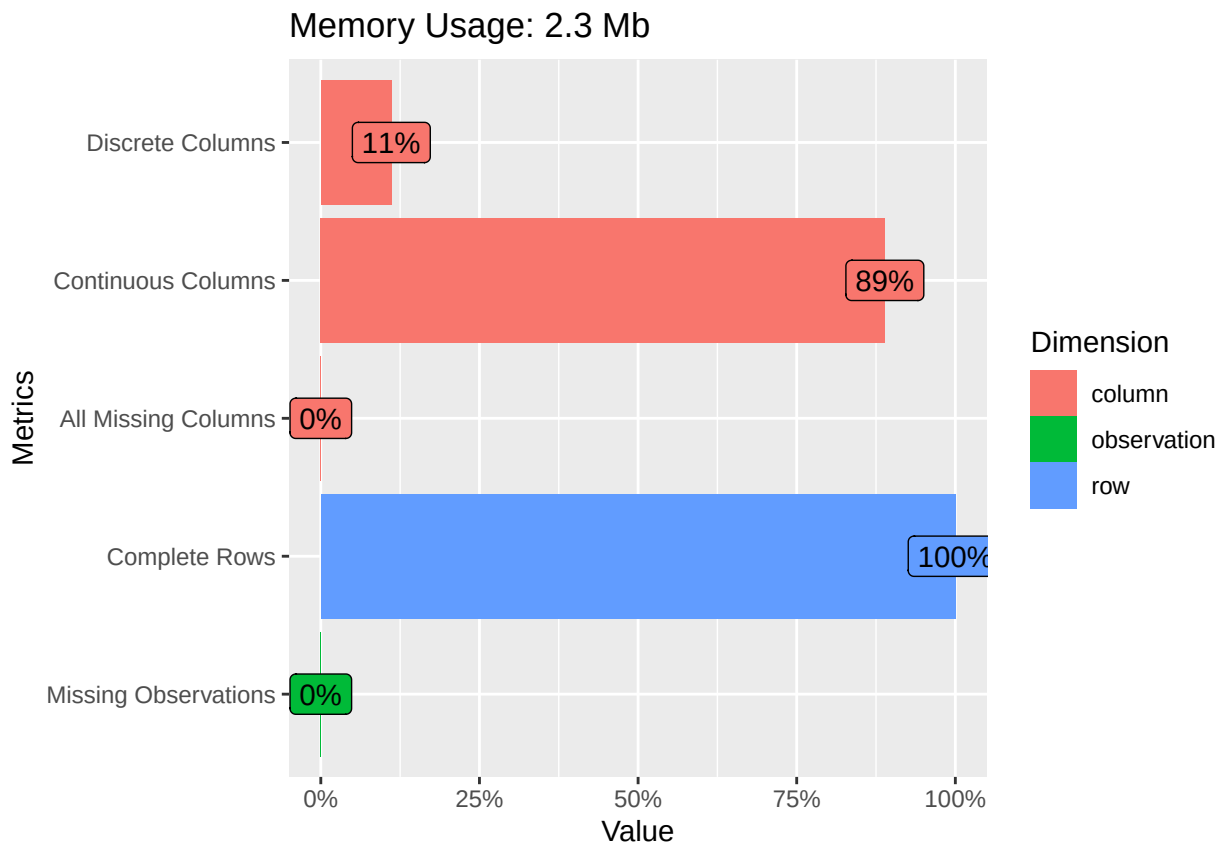



```
df=df %>% dplyr::select(-date)
df$WeekStatus[df$WeekStatus == "Weekend"]<-0
df$WeekStatus[df$WeekStatus == "Weekday"]<-1
df$WeekStatus<-as.double(df$WeekStatus)
glimpse(df)
```

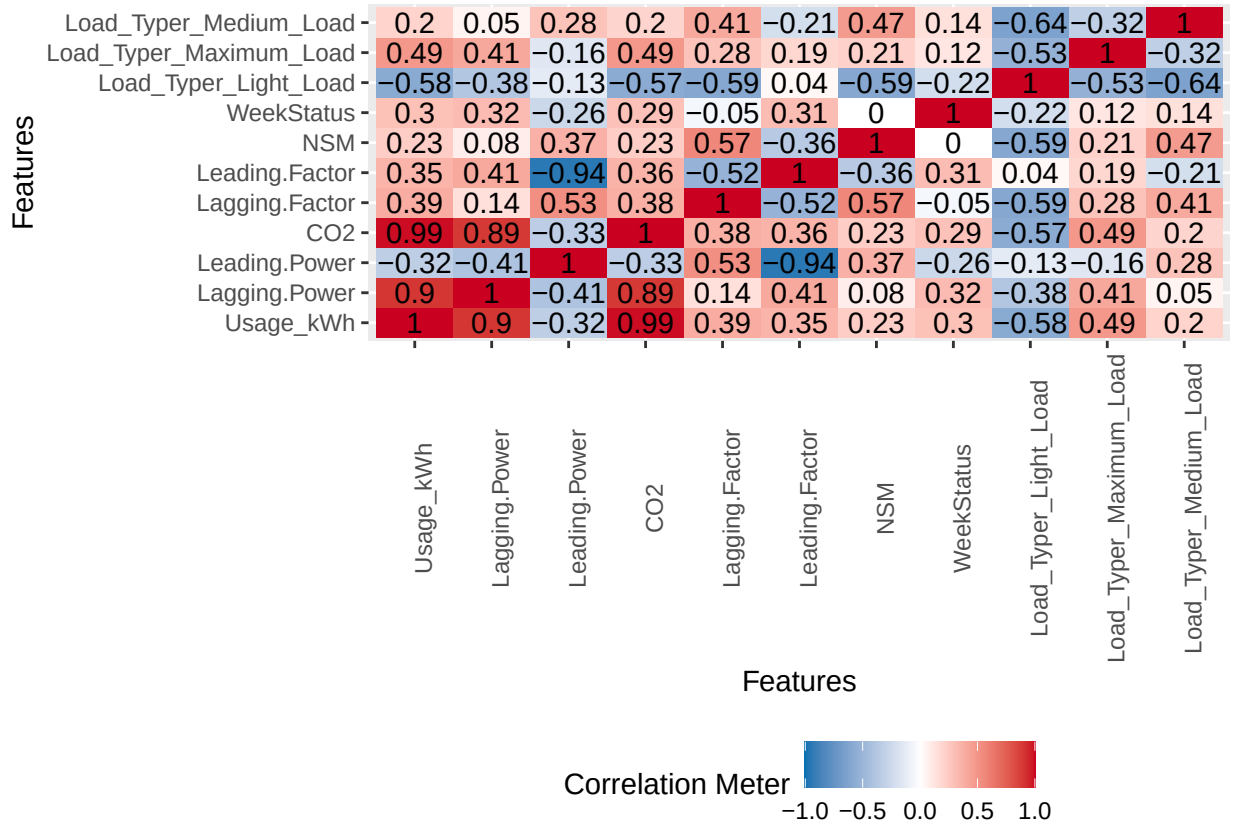
```
## Rows: 35,040
## Columns: 10
## $ Usage_kWh                              <dbl> 3.17, 4.00, 3.24, 3.31, 3.82, 3.2~
## $ Lagging_Current_Reactive.Power_kVarh <dbl> 2.95, 4.46, 3.28, 3.56, 4.50, 3.5~
## $ Leading_Current_Reactive_Power_kVarh <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ CO2.tCO2.                              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ Lagging_Current_Power_Factor           <dbl> 73.21, 66.77, 70.28, 68.09, 64.72~
## $ Leading_Current_Power_Factor           <dbl> 100, 100, 100, 100, 100, 100, 100~
## $ NSM                                    <int> 900, 1800, 2700, 3600, 4500, 5400~
## $ WeekStatus                             <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ Day_of_week                            <chr> "Monday", "Monday", "Monday", "Mo~
## $ Load_Type                              <chr> "Light_Load", "Light_Load", "Ligh~
```

```
df=df %>% dplyr::select(-Day_of_week)
colnames(df)<-c("Usage_kWh","Lagging.Power","Leading.Power","CO2","Lagging.Factor","Leading.Factor","NSM
```

```
plot_intro(df)
```



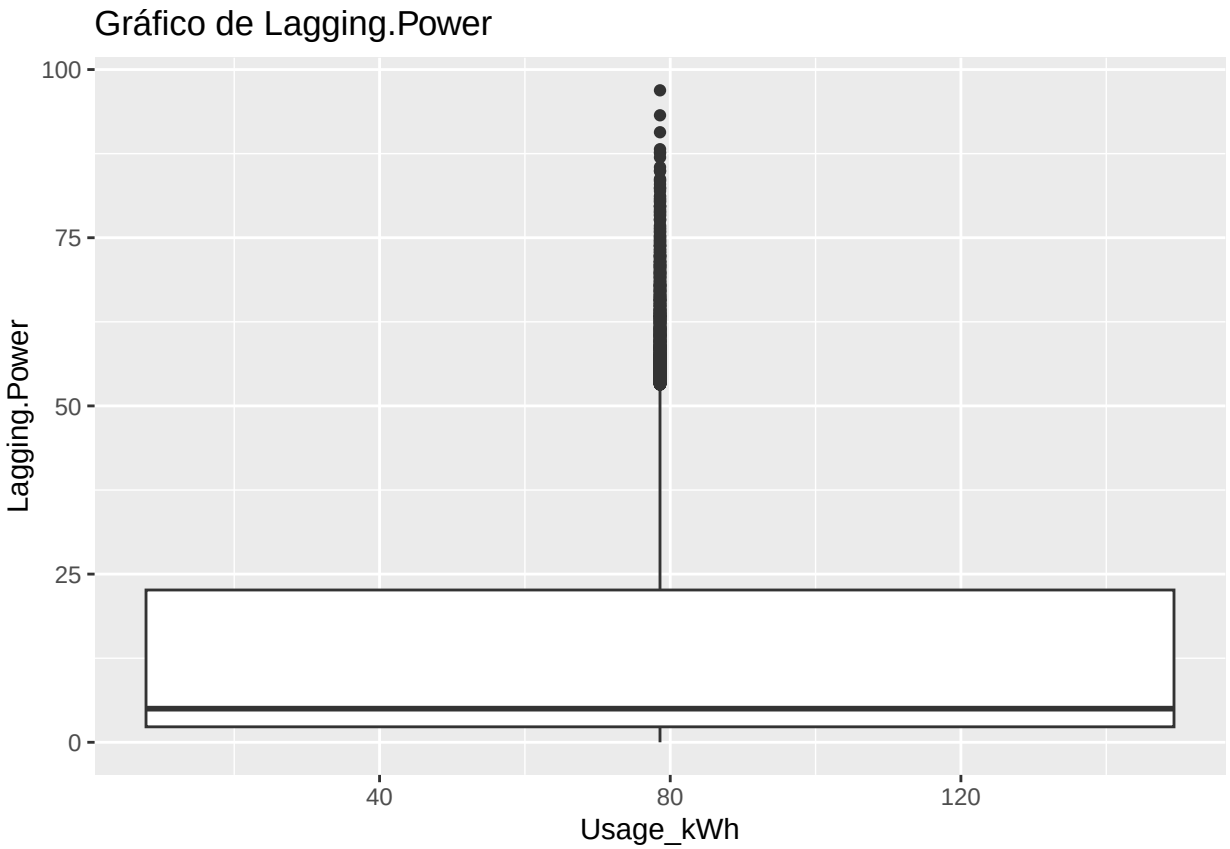

```
plot_correlation(df)
```



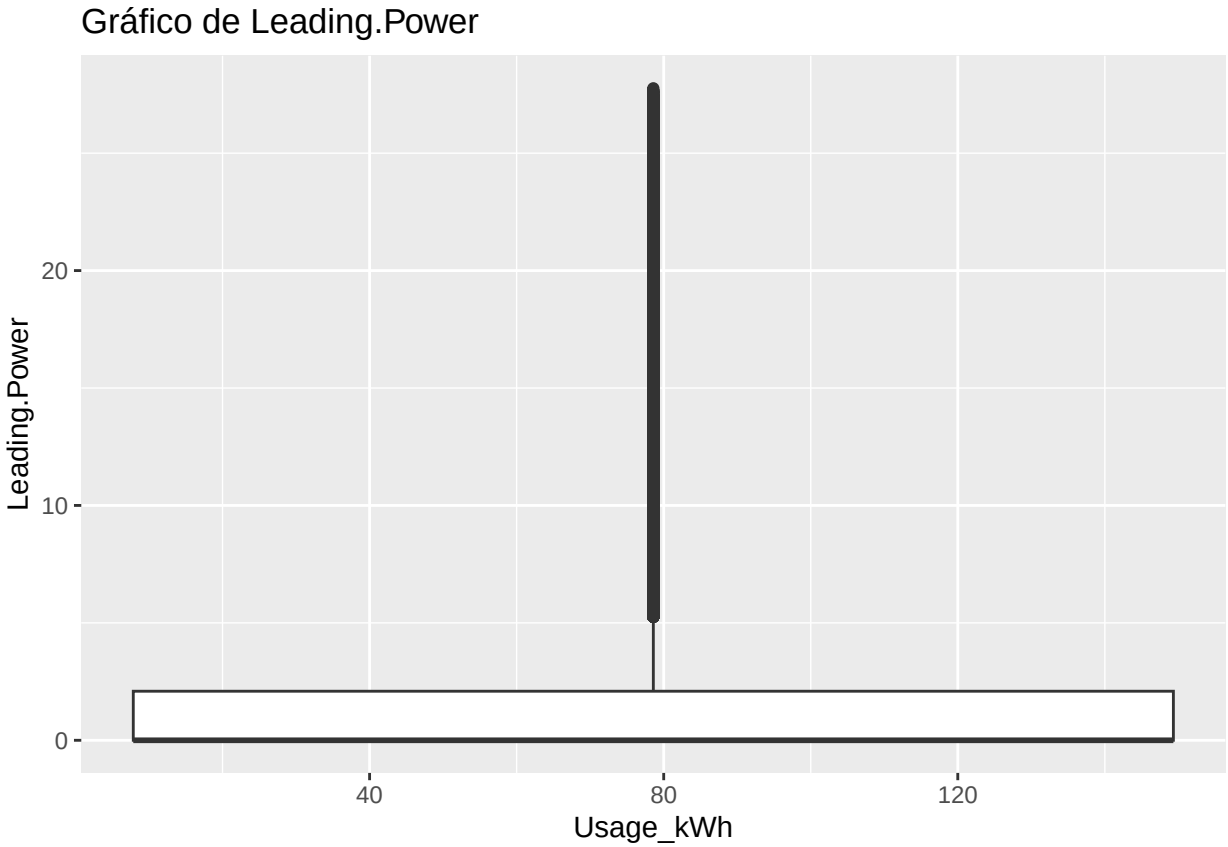

```
variables <- c("Lagging.Power","Leading.Power","CO2","Lagging.Factor","Leading.Factor","NSM")

plots <- list()

for (variable in variables) {
  plot <- ggplot(df) +
    geom_boxplot(aes(x = Usage_kWh, y = .data[[variable]], fill = Usage_kWh), shape = "circle") +
    scale_fill_hue(direction = -1) +
    theme_gray() +
    ggtitle(paste("Gráfico de", variable))
  print(plot)
  plots[[variable]] <- plot
}
```

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```

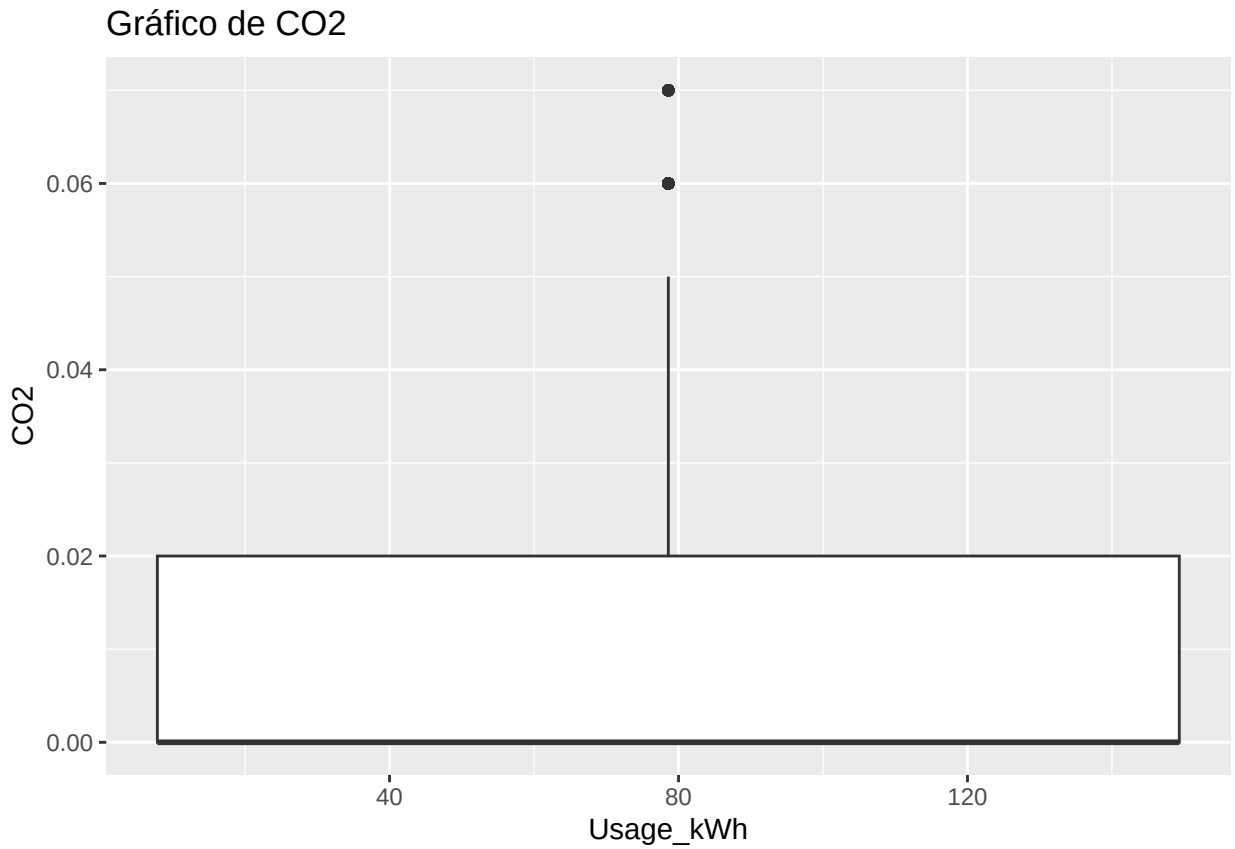
```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



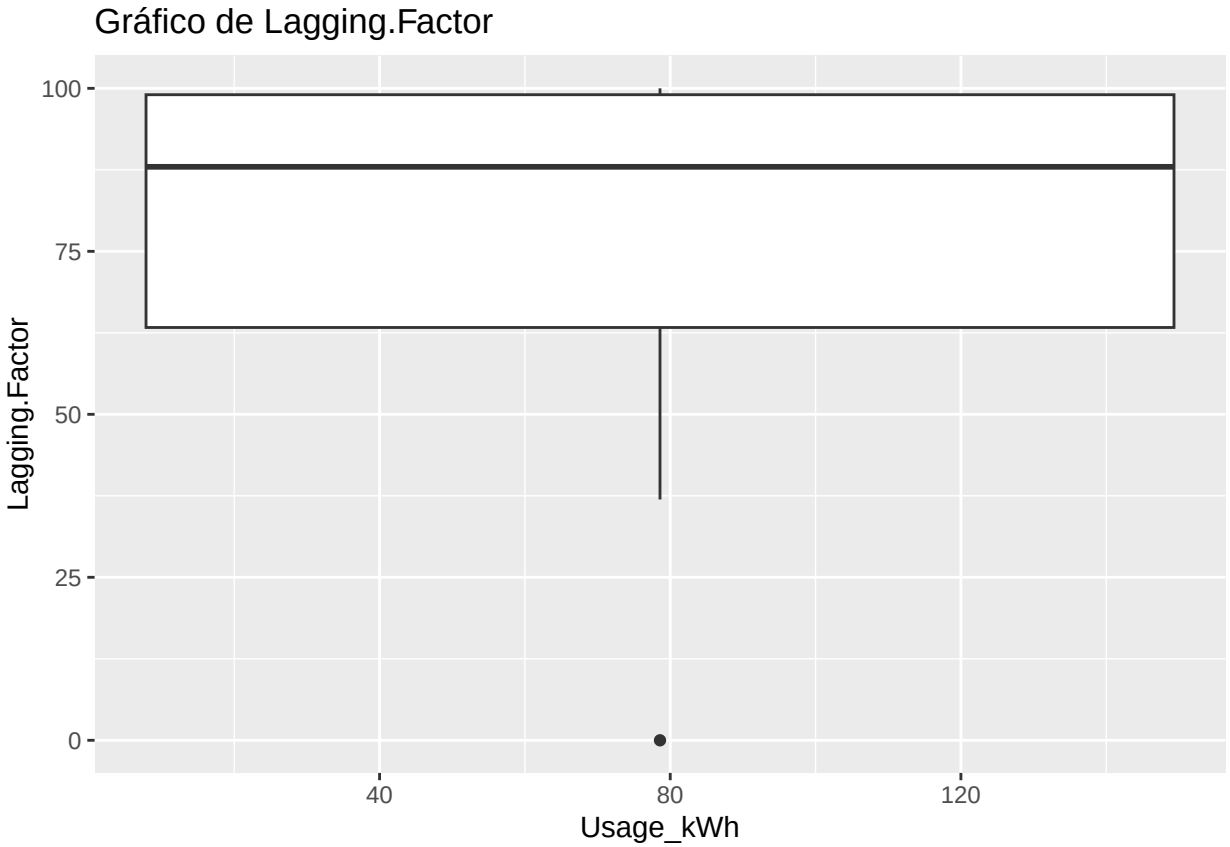

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



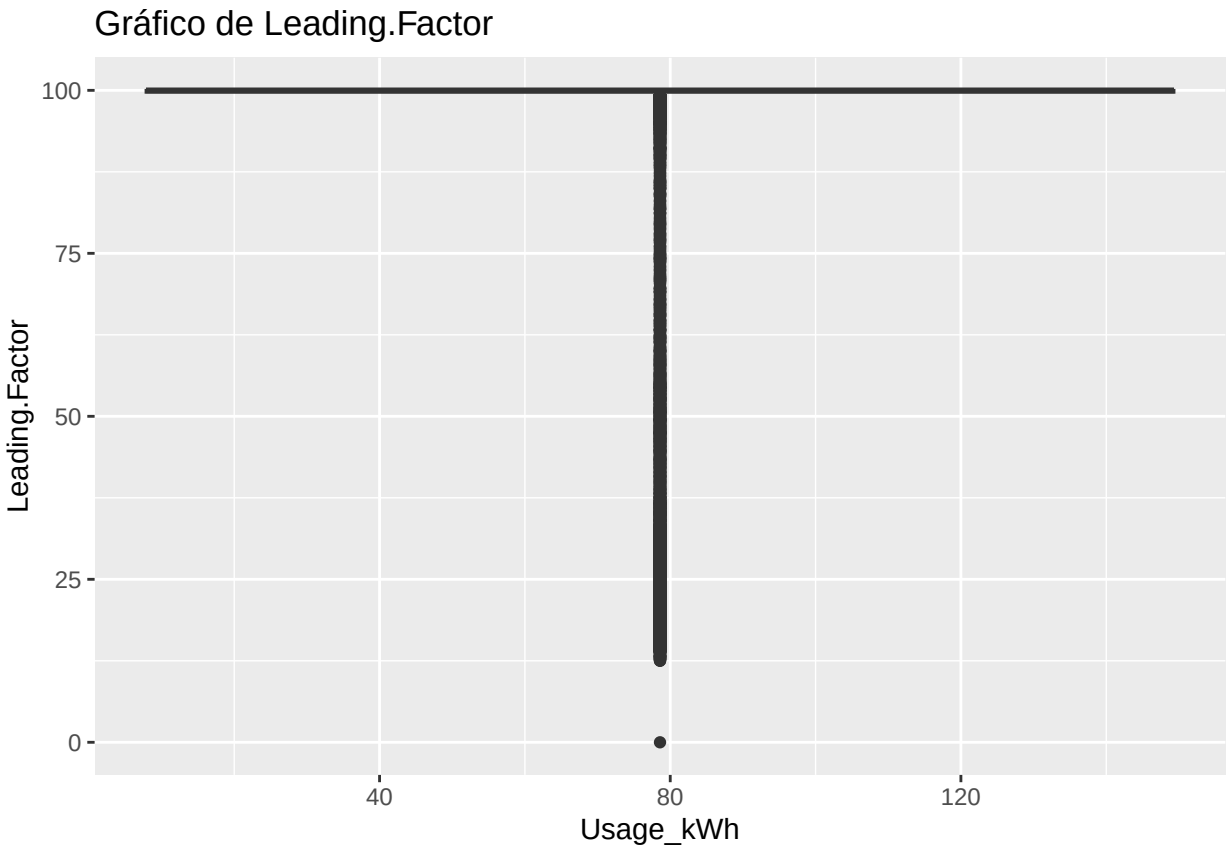

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



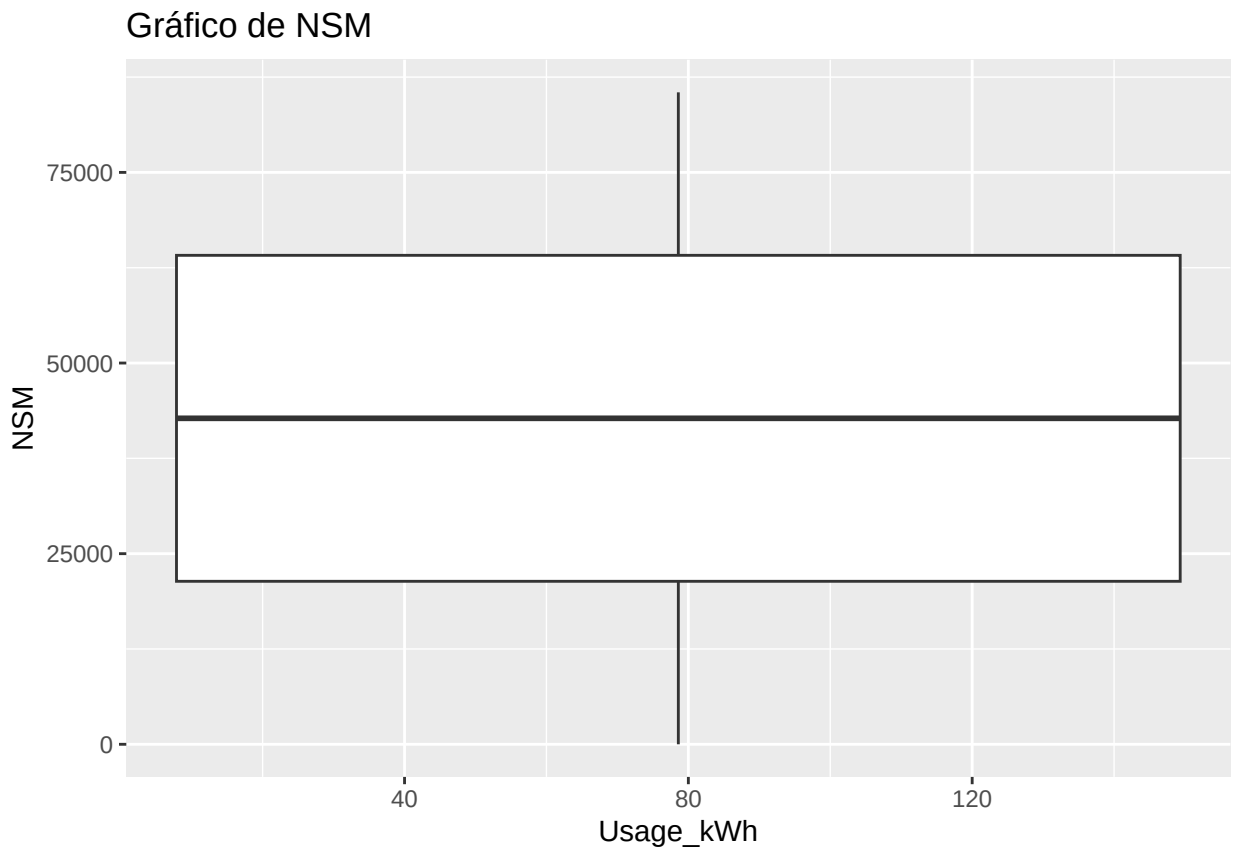

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



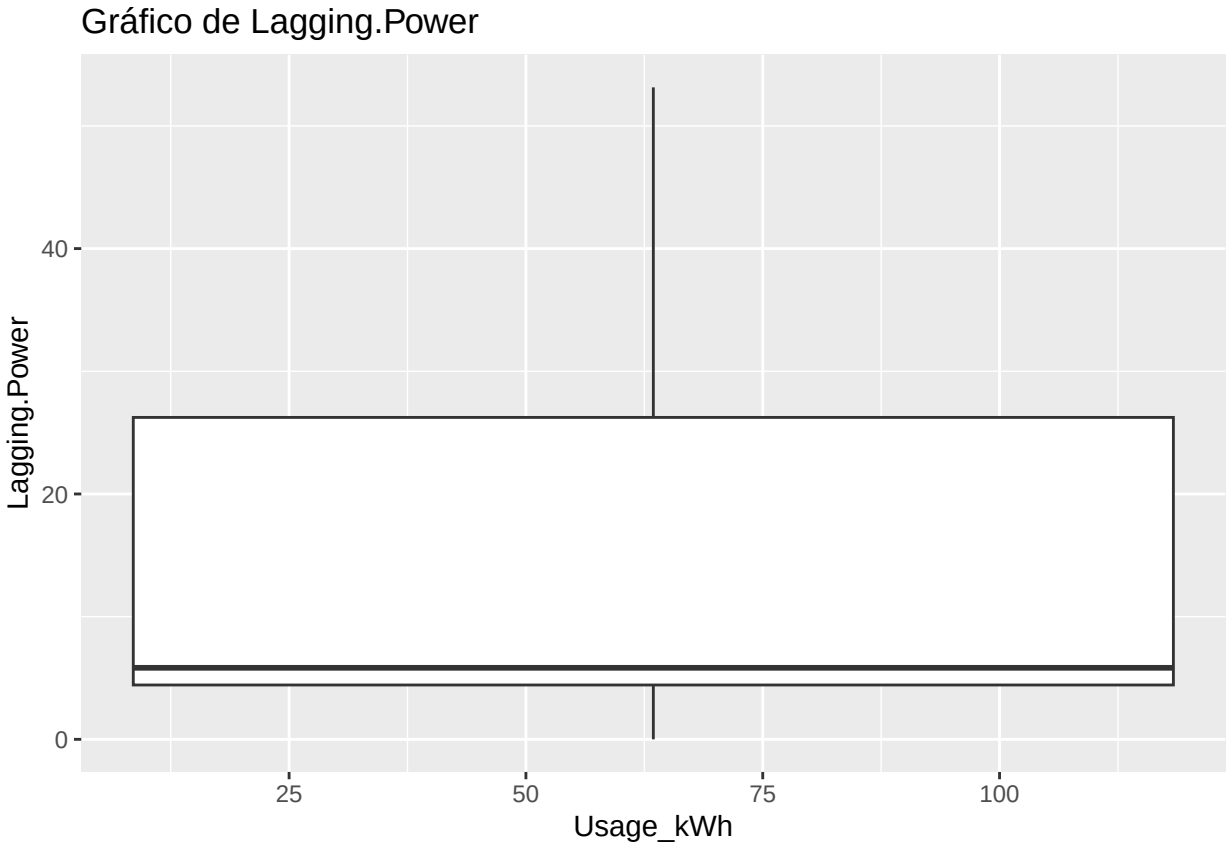

```
for (i in c("Lagging.Power","Leading.Power","CO2","Lagging.Factor","Leading.Factor","NSM"))
{
outliers <- boxplot.stats(df[[i]])$out
df[[i]][df[[i]] %in% outliers] <- NA
}
df <- filter_if(df, is.numeric , all_vars(!is.na(.)))
```

```
variables <- c("Lagging.Power","Leading.Power","CO2","Lagging.Factor","Leading.Factor","NSM")

plots <- list()

for (variable in variables) {
  plot <- ggplot(df) +
    geom_boxplot(aes(x = Usage_kWh, y = .data[[variable]], fill = Usage_kWh), shape = "circle") +
    scale_fill_hue(direction = -1) +
    theme_gray() +
    ggtitle(paste("Gráfico de", variable))
  print(plot)
  plots[[variable]] <- plot
}
```

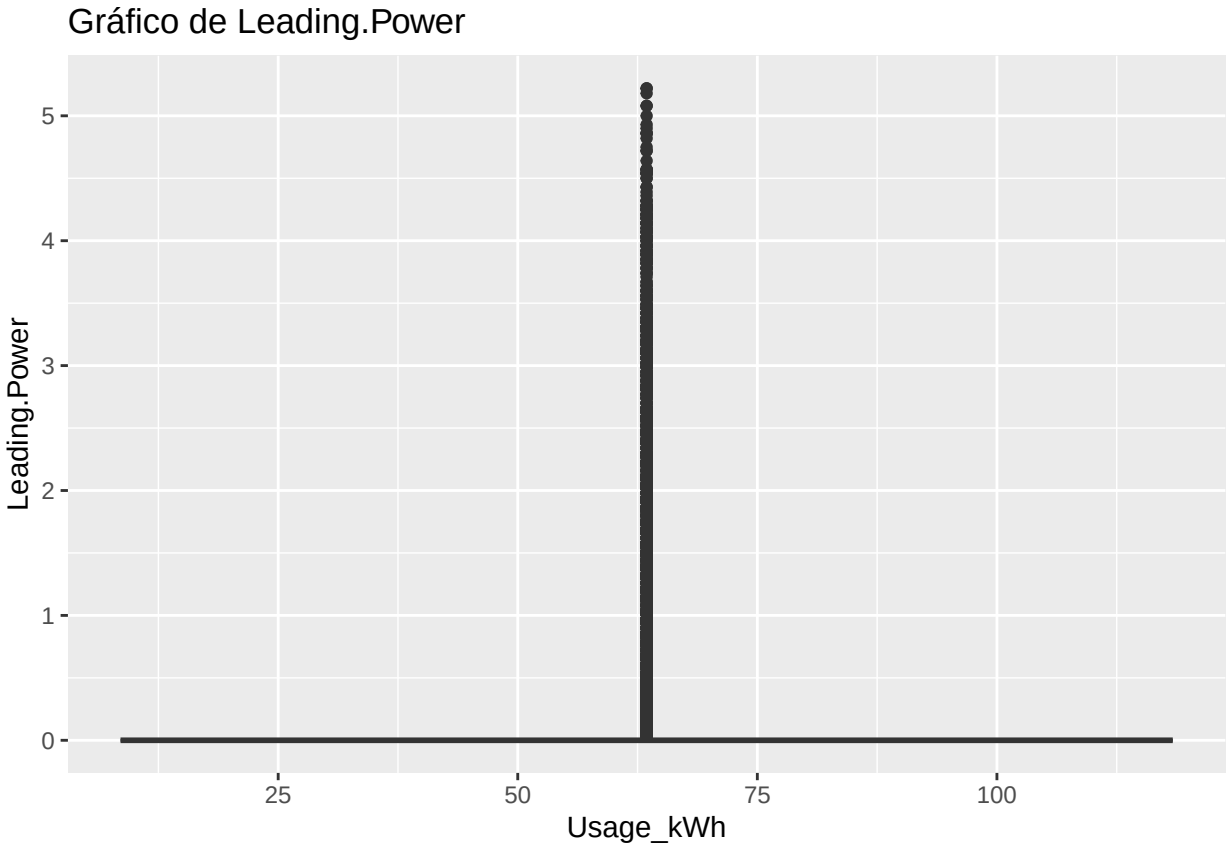
```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
```

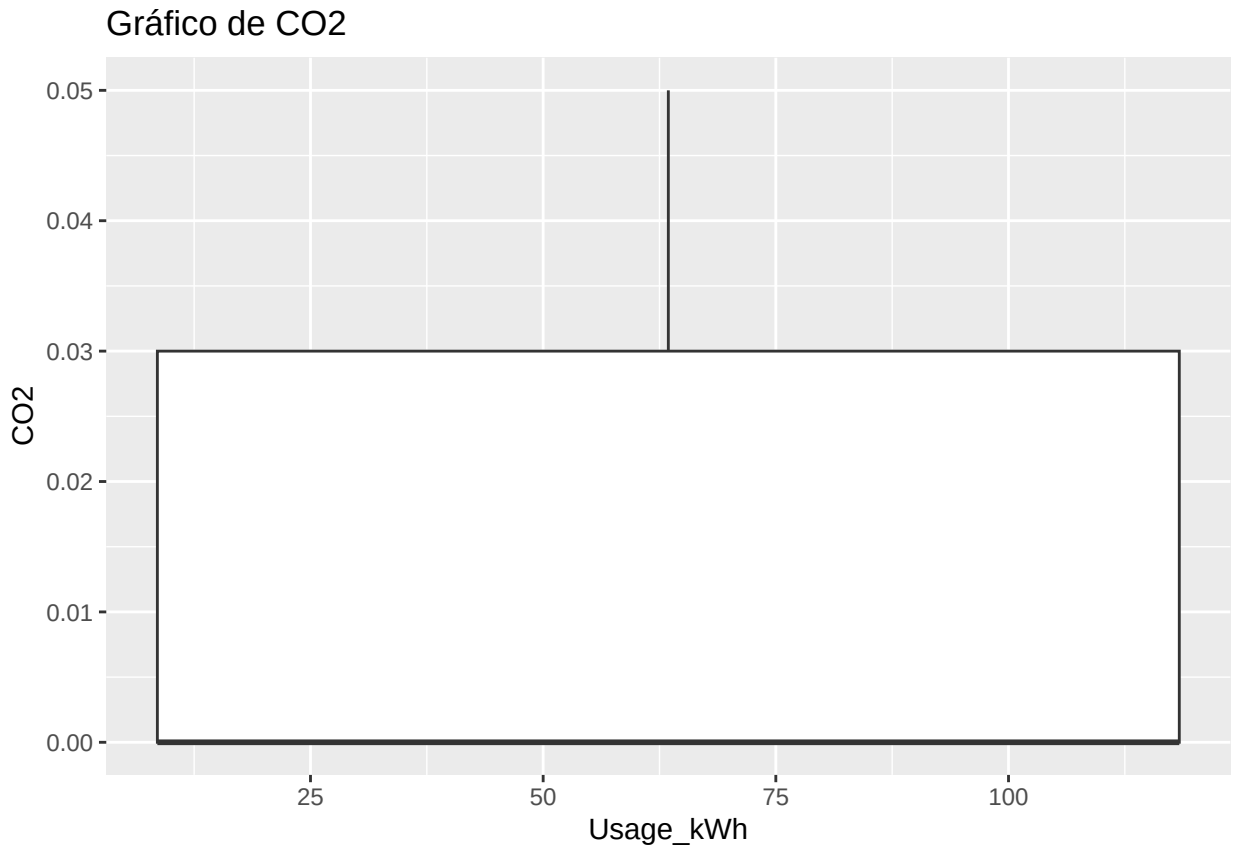
```
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



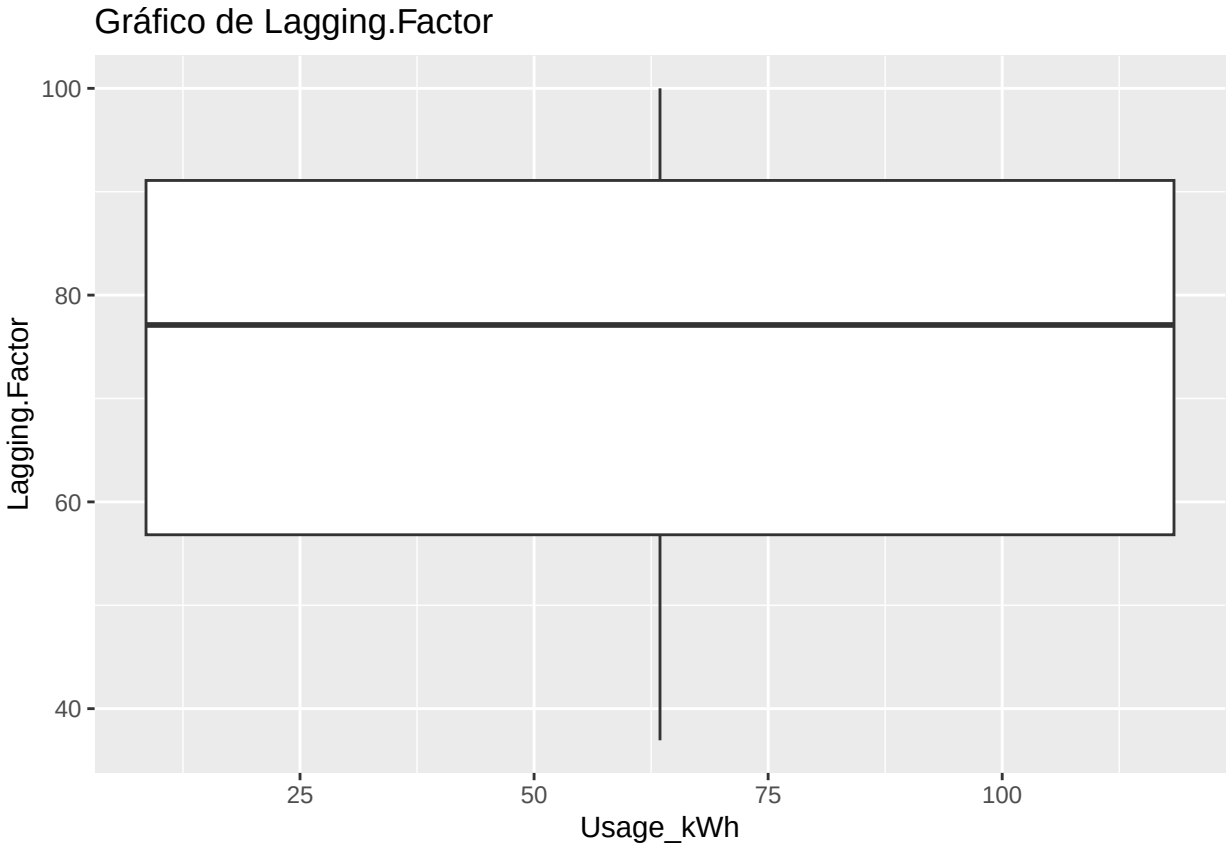

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



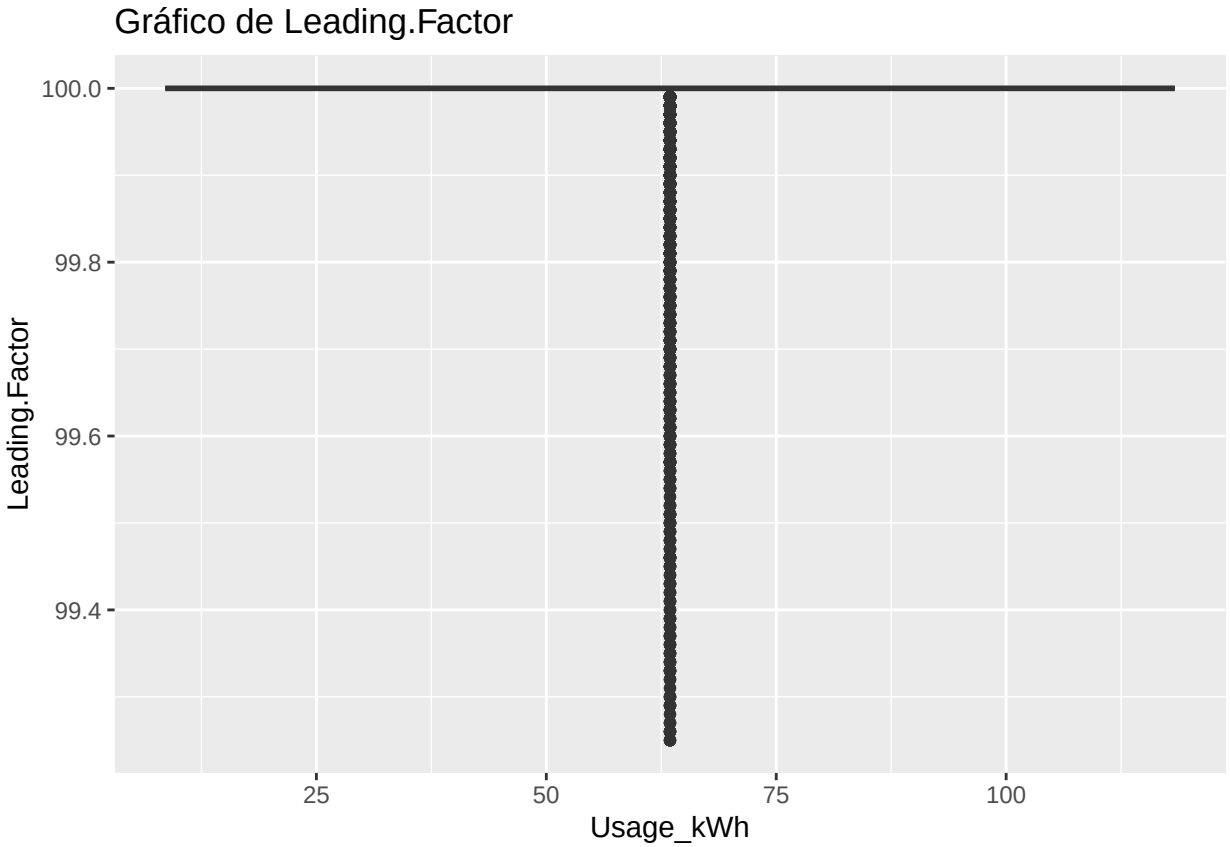

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



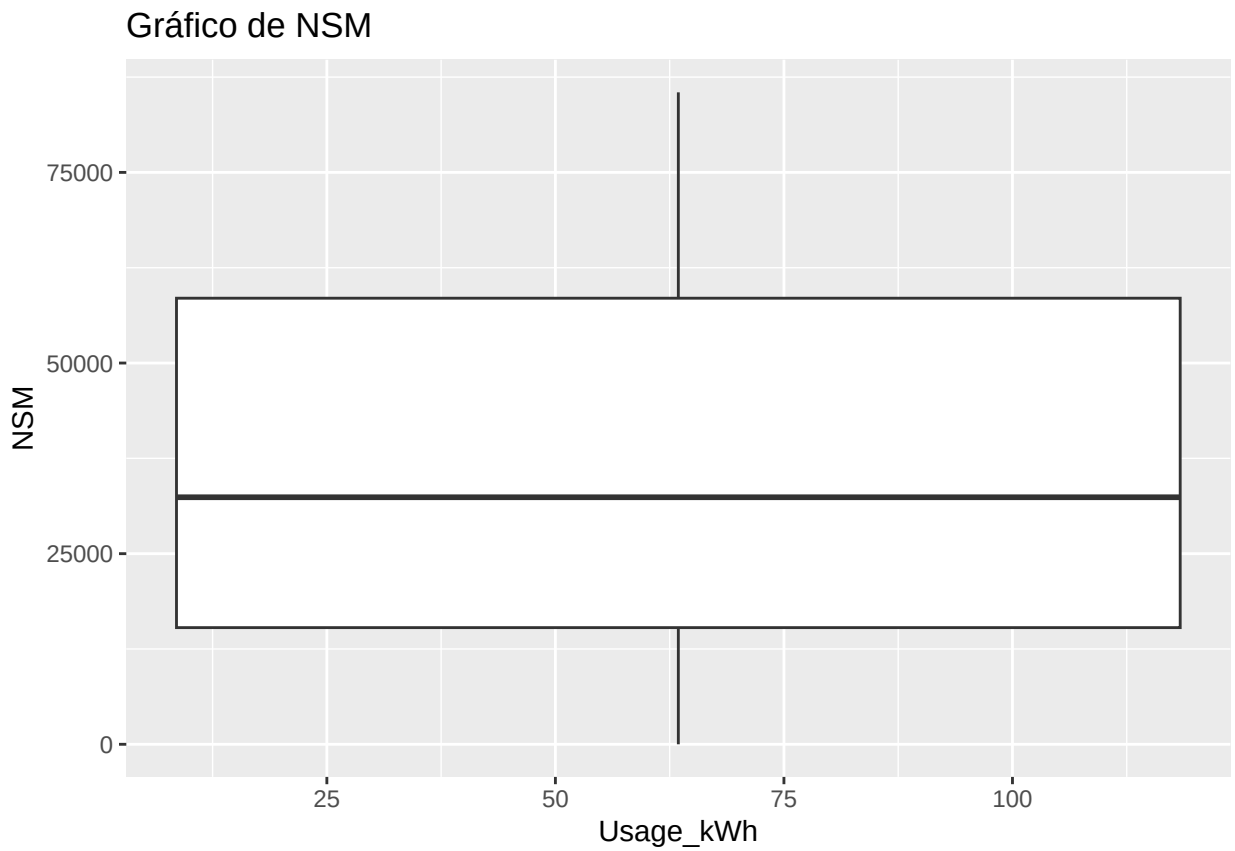

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



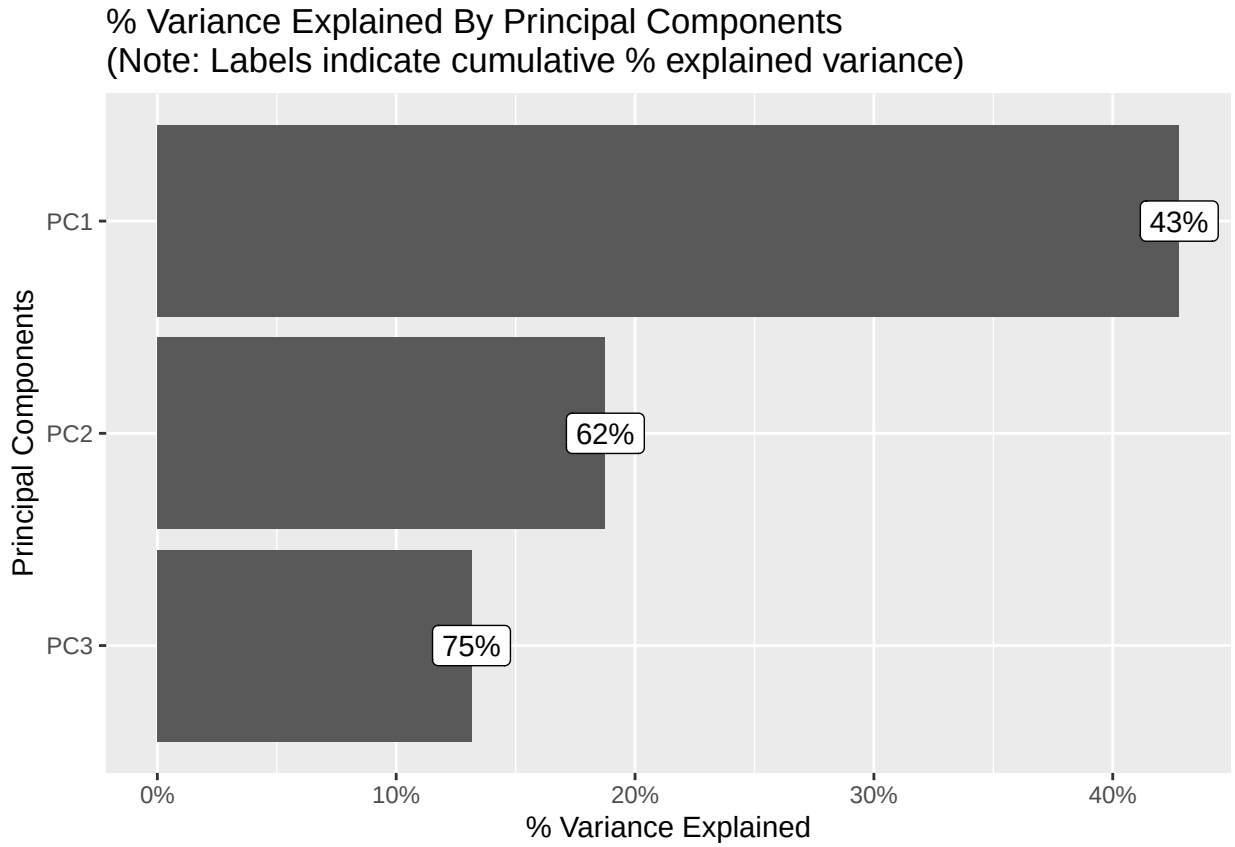

```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

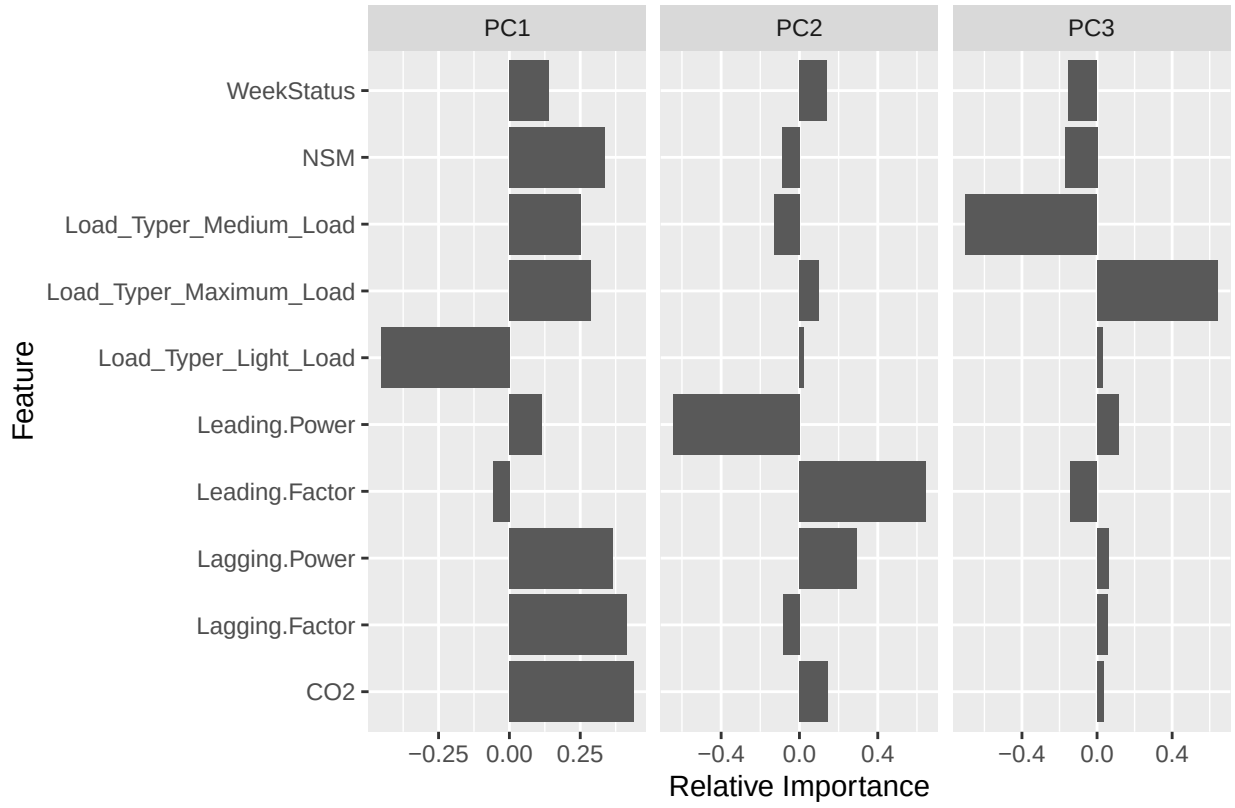



```
## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## The following aesthetics were dropped during statistical transformation: fill
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
y <-df$Usage_kWh
x=df %>% dplyr::select(-Usage_kWh)
plot_prcomp(x)
```


% Variance Explained By Principal Components
(Note: Labels indicate cumulative % explained variance)
Principal Components
PC1
PC2
PC3
43%
62%
75%
0%
10%
20%
30%
40%
% Variance Explained




```
df=df %>% dplyr::select(-Load_Typer)
df <- as.data.frame(scale(df))
```

```
split <- initial_split(df, prop = 0.8)
train_data <- training(split)
test_data <- testing(split)
```

Metodologías Supervisadas

```
fit1 <- glm(formula=Usage_kWh ~ .,data=train_data)
summary(fit1)
```

Regresión Logística

```
##
## Call:
## glm(formula = Usage_kWh ~ ., data = train_data)
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0001165  0.0009989   0.117 0.907198
## Lagging.Power  0.1085539  0.0023209  46.772  < 2e-16 ***
## Leading.Power  0.0061804  0.0015967   3.871 0.000109 ***
## CO2            0.8387915  0.0029864 280.872  < 2e-16 ***
## Lagging.Factor 0.0651408  0.0018015  36.158  < 2e-16 ***
## Leading.Factor 0.0124608  0.0015571   8.002 1.28e-15 ***
```

```
## NSM            0.0071989  0.0011700   6.153 7.74e-10 ***
## WeekStatus     0.0021973  0.0010366   2.120 0.034036 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.02035177)
##
##     Null deviance: 20397.43  on 20395  degrees of freedom
## Residual deviance:   414.93  on 20388  degrees of freedom
## AIC: -21543
##
## Number of Fisher Scoring iterations: 2
```

```
pred_logistic<-predict(fit1,test_data,type="response")
```

```
head(pred_logistic)
```

```
##          2          8         11         14         16         20
## -0.8135542 -0.8220164 -0.8203856 -0.8177655 -0.8251620 -0.8122296
```

```
cor(pred_logistic, test_data$Usage_kWh)
```

```
## [1] 0.9899757
```

```
diferencias_cuadradas <- (pred_logistic - test_data$Usage_kWh)^2
mse <- mean(diferencias_cuadradas)
mse
```

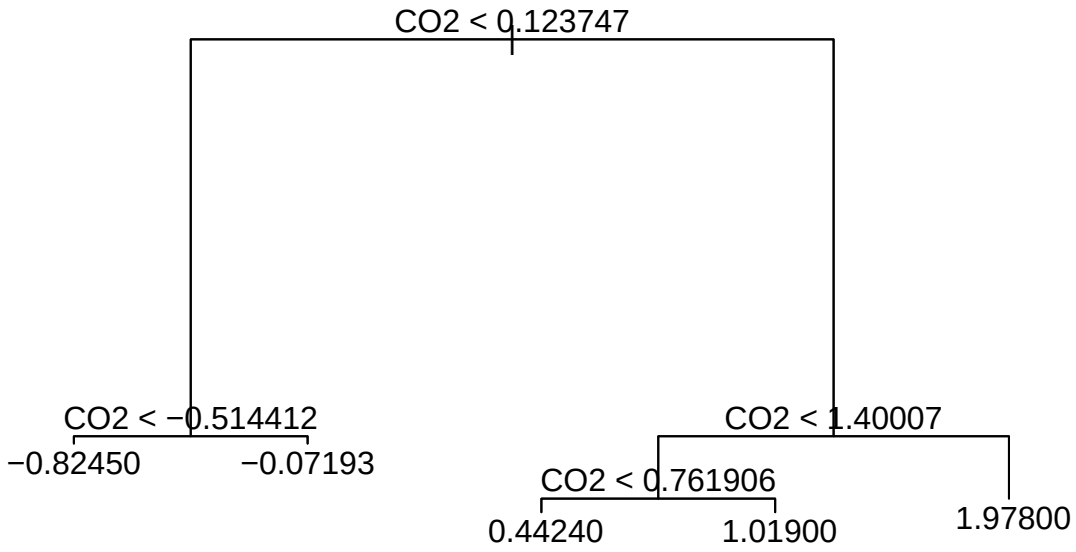
```
## [1] 0.01993748
```

```
library(tree)
tree.fit <- tree(Usage_kWh ~ ., data = train_data)
summary(tree.fit)
```

Árbol de decisiones

```
##
## Regression tree:
## tree(formula = Usage_kWh ~ ., data = train_data)
## Variables actually used in tree construction:
## [1] "CO2"
## Number of terminal nodes:  5
## Residual mean deviance:  0.03152 = 642.8 / 20390
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.55690 -0.03061 -0.01270  0.00000  0.03312  3.73100
```

```
plot(tree.fit)
text(tree.fit, pretty = 0)
```




```
tree_pred <- predict(tree.fit, test_data)
```

```
cor(tree_pred, test_data$Usage_kWh)
```

```
## [1] 0.984236
```

```
diferencias_cuadradas <- (tree_pred - test_data$Usage_kWh)^2
mse <- mean(diferencias_cuadradas)
mse
```

```
## [1] 0.03126526
```

```
library(e1071)
```

Máquina de Vectores de Soporte

```
##
## Attaching package: 'e1071'
```

```
## The following object is masked from 'package:parsnip':
##
##     tune
```

```
## The following object is masked from 'package:rsample':
##
##     permutations
```

```
svm1<-svm(formula = Usage_kWh~., data=train_data, kernel="radial")
svm2<-svm(formula = Usage_kWh~., data=train_data, kernel="linear")
```

```
pred1<-predict(svm1, newdata=test_data)
pred2<-predict(svm2, newdata=test_data)
```

```
cor(pred1, test_data$Usage_kWh)
```

```
## [1] 0.998248
```

```
cor(pred2, test_data$Usage_kWh)
```

```
## [1] 0.9898099
```

```
diferencias_cuadradas <- (pred1 - test_data$Usage_kWh)^2
mse <- mean(diferencias_cuadradas)
mse
```

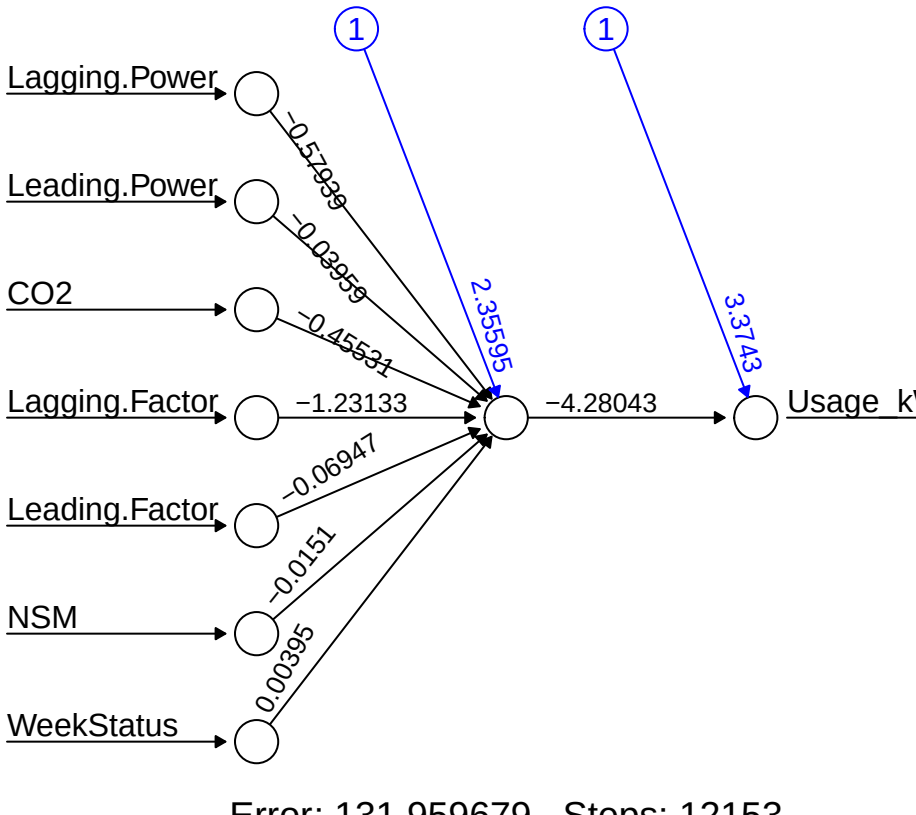
```
## [1] 0.00373797
```

```
diferencias_cuadradas <- (pred2 - test_data$Usage_kWh)^2
mse <- mean(diferencias_cuadradas)
mse
```

```
## [1] 0.0208494
```

```
concrete_model <- neuralnet(Usage_kWh ~ .,data = train_data)
```

```
plot(concrete_model, rep="best")
```



```r
predictions <- compute(concrete_model, test_data)
```

```r
predicted_values <- predictions$net.result
```

```r
cor(predicted_values, test_data$Usage_kWh)
```

```
##           [,1]
## [1,] 0.9933906
```

```r
diferencias_cuadradas <- (predicted_values - test_data$Usage_kWh)^2
mse <- mean(diferencias_cuadradas)
mse
```

```
## [1] 0.01317425
```

```r
train_predictions <- compute(concrete_model, train_data)
train_predicted_values <- train_predictions$net.result
train_error <- sum((train_predicted_values - train_data$Usage_kWh)^2) / nrow(train_data)

# Error en el conjunto de prueba
test_predictions <- compute(concrete_model, test_data)
test_predicted_values <- test_predictions$net.result
test_error <- sum((test_predicted_values - test_data$Usage_kWh)^2) / nrow(test_data)

# Imprimir los errores
cat("Error en el conjunto de entrenamiento:", train_error, "\n")
```

```
## Error en el conjunto de entrenamiento: 0.01293976
```

```r
cat("Error en el conjunto de prueba:", test_error, "\n")
```

```
## Error en el conjunto de prueba: 0.01317425
```