

Proyecto 02

Jorge Porras & Alex Cruz

12/4/2020

¿Es posible predecir desbalances de tension?

¿Por que no usar series de tiempo?

huecos en las mediciones (faltantes)

Carga de librerias

```
knitr::opts_chunk$set(echo = TRUE)
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
```

```
#library(RODBC)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##   date
```

```
library(feather)
library(ggplot2)
library(tidyr)
library(caTools)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

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

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(e1071)
library(mltest)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

Rango de fechas (lubridate)

```
final_dateCR <- floor_date(now(), "week") + days(1) ## primer lunes hacia atras
initial_dateCR <- final_dateCR - years(3) ## 3 años hacia atras
```

Fecha-hora a UTC (lubridate) (DB está en UTC)

```
initial_date <- with_tz(initial_dateCR, tzone = "UTC")
final_date <- with_tz(final_dateCR, tzone = "UTC")
```

Valores Nominales (vectores)

```
nom_voltage <- 35000
cut_voltage <- 34000
clasif <- c("Low" = 0.3,
            "Low_Mid" = 0.5,
            "Mid" = 0.7,
            "Mid_High" = 0.9,
            "High" = 1.1)
```

Conexion a SQL Server y carga de tablas (RODBC)

```

channel <- odbcConnect("SQL_ION", uid="sa", pwd="Con3$adm.")
#sources <- sqlQuery(channel , "select top 100 ID, Name, DisplayName from Source where Name like 'Coope'
sources <- sqlQuery(channel , "select top 100 ID, Name, DisplayName from Source where Name like 'Coopea'
#sources$Name <- gsub("Coopeguanacaste.", '', sources$Name)
sources$Name <- gsub("Coopealfaroruiz.", '', sources$Name)l
sources <- sources %>% filter(ID %in% c(9))

quantity <- sqlQuery(channel , "select top 1500000 ID, Name from Quantity where Name like 'Voltage%'"
quantity <- quantity %>% filter(grepl("^Voltage Phases [ABC][ABC] Mean$", Name))
quantity$Name <- c('Vab', 'Vbc', 'Vca')

sources_ids <- paste0(sources$ID, collapse = ",")
quantity_ids <- paste0(quantity$ID, collapse = ",")
dataLog <- sqlQuery(channel , paste0("select top 500000 * from dataLog2 where ",
                                     "SourceID in (", sources_ids, ")",
                                     " and QuantityID in (", quantity_ids, ")",
                                     " and TimestampUTC >= '", initial_date, "'",
                                     " and TimestampUTC < '", final_date, "'"))

odbcCloseAll()

```

Guardar archivos con las tablas

```

#write_feather(dataLog, "featherFiles/dataLog.feather")
#write_feather(quantity, "featherFiles/quantity.feather")
#write_feather(sources, "featherFiles/sources.feather")

```

Leer Archivos de las tablas

```

#rm(dataLog, quantity, sources)
dataLog <- read_feather("featherFiles/dataLog.feather")
quantity <- read_feather("featherFiles/quantity.feather")
sources <- read_feather("featherFiles/sources.feather")

```

Contenido de las tablas

Datalog

```
glimpse(dataLog)
```

```

## Observations: 315,360
## Variables: 5
## $ ID          <int> 31652855, 31653877, 31654633, 31655697, 31656715, 316577...
## $ Value       <dbl> 35683.51, 35718.07, 35744.92, 35743.99, 35745.72, 35756....
## $ SourceID    <int> 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,...
## $ QuantityID  <int> 167, 167, 167, 167, 167, 167, 167, 167, 167, 167, 167, 1...
## $ TimestampUTC <fct> 2017-04-20 06:00:00.0000000, 2017-04-20 06:15:00.0000000...

```

Quantity

```
glimpse(quantity)
```

```
## Observations: 3
## Variables: 2
## $ ID      <int> 167, 173, 176
## $ Name    <chr> "Vab", "Vbc", "Vca"
```

Sources

```
glimpse(sources)
```

```
## Observations: 1
## Variables: 3
## $ ID          <int> 9
## $ Name        <chr> "Sub_Laguna"
## $ DisplayName <fct> Coopealfaror Ruiz.Sub_Laguna
```

Transformacion de datos para analisis previo

Union de tablas, borrado de columnas no importantes y Categorizacion de valores

```
## Transformacion de columnas
dataLog$TimestampUTC <- as_datetime(dataLog$TimestampUTC)
dataLog$TimestampCR <- with_tz(dataLog$TimestampUTC, tzone = "America/Costa_Rica")
dataLog$TimestampUTC <- NULL
dataLog$ID <- NULL
dataLog$year <- year(dataLog$TimestampCR)
dataLog$month <- month(dataLog$TimestampCR)
dataLog$day <- day(dataLog$TimestampCR)
dataLog$hour <- hour(dataLog$TimestampCR)
dataLog$minute <- minute(dataLog$TimestampCR)
dataLog$yday <- wday(dataLog$TimestampCR)
dataLog$ydayName <- weekdays(dataLog$TimestampCR)
dataLog$monthName <- month.abb[dataLog$month]

dataLog$hour2 <- hour(dataLog$TimestampCR) + (minute(dataLog$TimestampCR)/60)

dataLog <- dataLog %>% left_join(quantity, by = c('QuantityID' = "ID")) %>%
  left_join(sources, by = c('SourceID' = "ID"))

names(dataLog)[names(dataLog) == "Name.x"] <- "Quantity"
names(dataLog)[names(dataLog) == "Name.y"] <- "Meter"

dataLog$SourceID <- NULL
dataLog$QuantityID <- NULL
dataLog$DisplayName <- NULL
#rm(quantity, sources)
```

Contenido de la tabla Datalog

```
glimpse(dataLog)
```

```
## Observations: 315,360
## Variables: 13
## $ Value      <dbl> 35683.51, 35718.07, 35744.92, 35743.99, 35745.72, 35756.9...
## $ TimestampCR <dtm> 2017-04-20 00:00:00, 2017-04-20 00:15:00, 2017-04-20 00:...
## $ year       <dbl> 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017, 201...
## $ month      <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ...
## $ day        <int> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 2...
## $ hour       <int> 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, ...
## $ minute     <int> 0, 15, 30, 45, 0, 15, 30, 45, 0, 15, 30, 45, 0, 15, 30, 4...
## $ wday       <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ...
## $ wdayName    <chr> "Thursday", "Thursday", "Thursday", "Thursday", "Thursday...
## $ monthName   <chr> "Apr", "Apr", "Apr", "Apr", "Apr", "Apr", "Apr", "Apr", "...
## $ hour2      <dbl> 0.00, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00, 2.2...
## $ Quantity    <chr> "Vab", "Vab", "Vab", "Vab", "Vab", "Vab", "Vab", "Vab", "...
## $ Meter       <chr> "Sub_Laguna", "Sub_Laguna", "Sub_Laguna", "Sub_Laguna", "...
```

Análisis de datos

Histogramas y boxplots

Cantidad de filas inicial

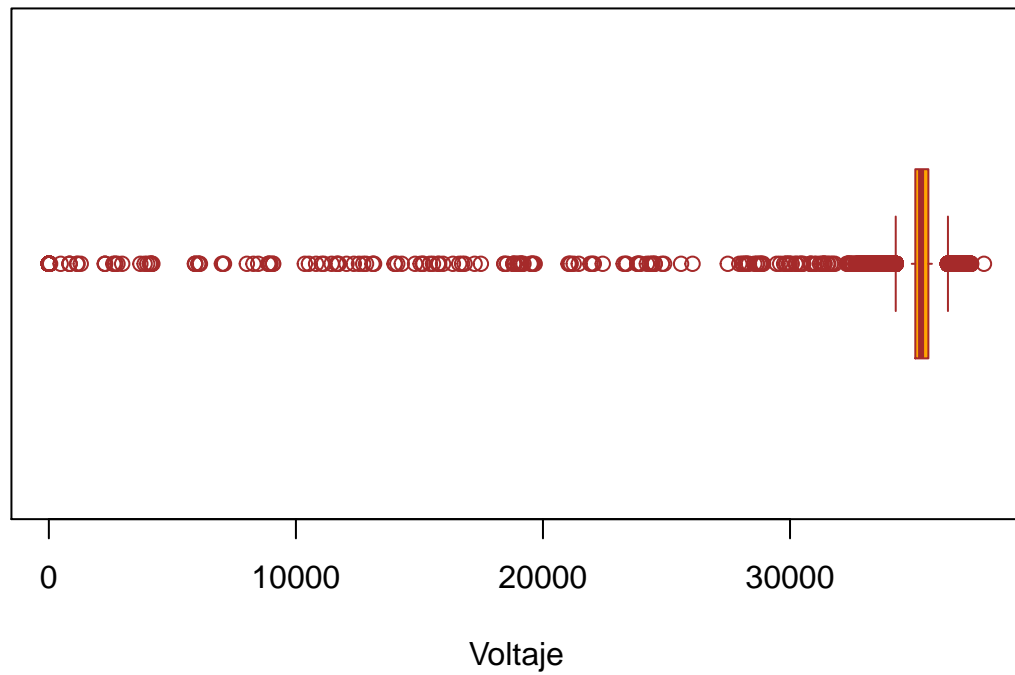
```
initial_rows <- nrow(dataLog)
initial_rows
```

```
## [1] 315360
```

Boxplot del comportamiento de la tension (horizontal)

```
boxplot(dataLog$Value, xlab= "Voltaje", col="Orange", border = "brown", horizontal = T, main = "Voltaje")
```

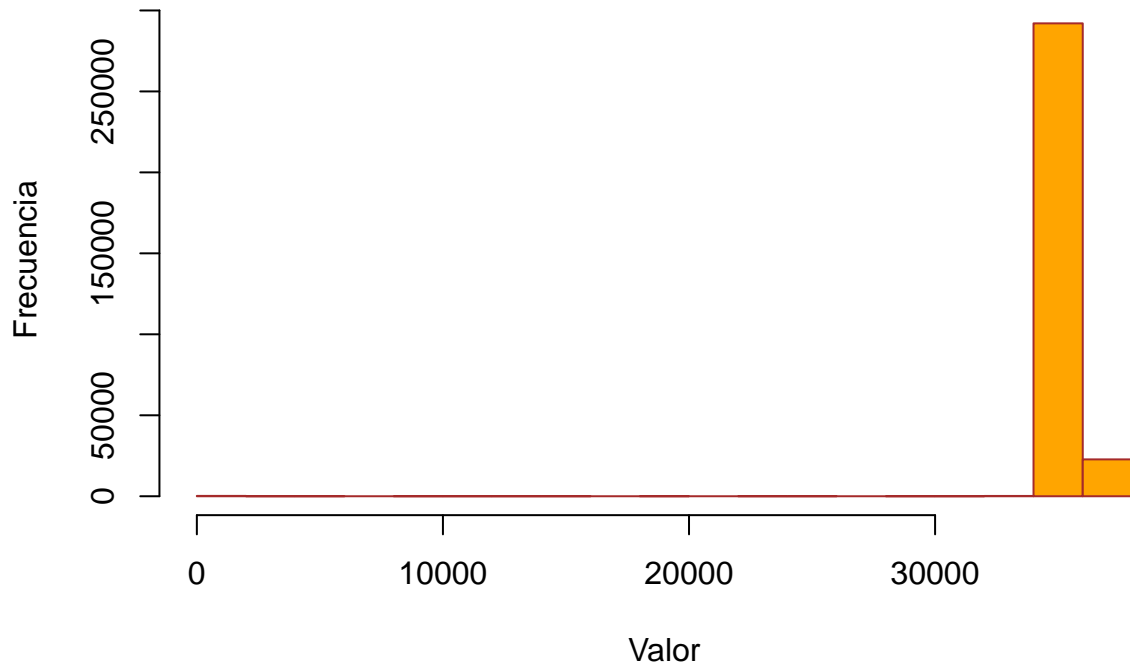
Voltaje Promedio



Histograma inicial

```
hist(dataLog$Value, col="Orange", border = "brown", xlab = "Valor", ylab = "Frecuencia", main = "Histograma inicial")
```

Histograma de Tension



Variable temporal solo para análisis (originalmente se tenían 15825 rows)

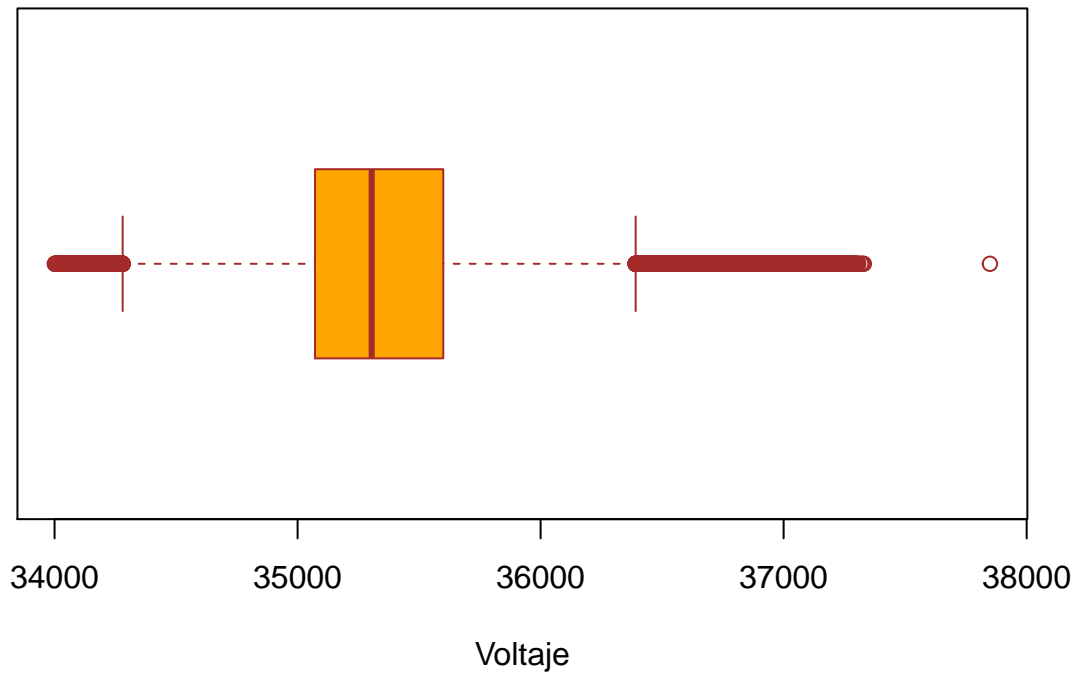
```
dl_temp <- dataLog %>% filter(Value > cut_voltage)
### Cantidad de filas
final_rows <- nrow(dl_temp)
print(paste0 ("Se eliminaron el ", round(100*((initial_rows - final_rows)/initial_rows), 2), "% de las :"))

## [1] "Se eliminaron el 0.18% de las filas"
```

Boxplot con un filtro temporal de los datos (para análisis unicamente)

```
boxplot(dl_temp$Value, xlab= "Voltaje", col="Orange", border = "brown", horizontal = T, main = "Voltaje")
```

Voltaje Promedio



Histograma eliminando outliers de la tabla temporal

```
hist(dl_temp$Value, col="Orange", border = "brown", xlab = "Tension", ylab = "Frecuencia", breaks = 60,
```


Histograma de Tension

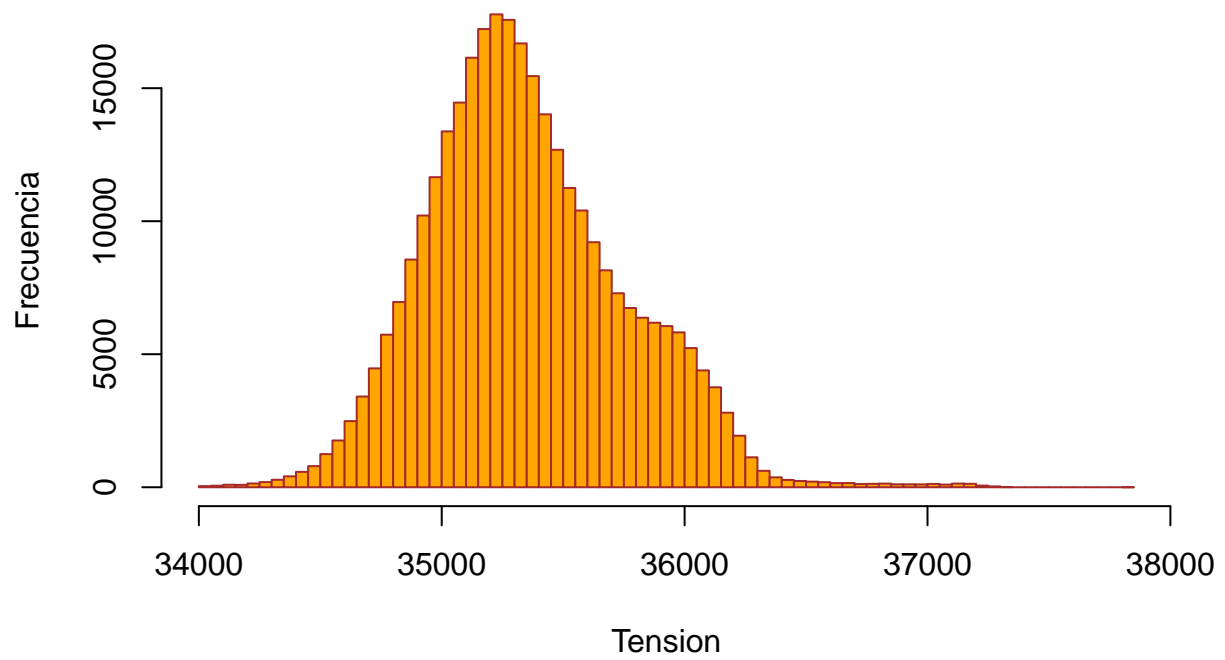
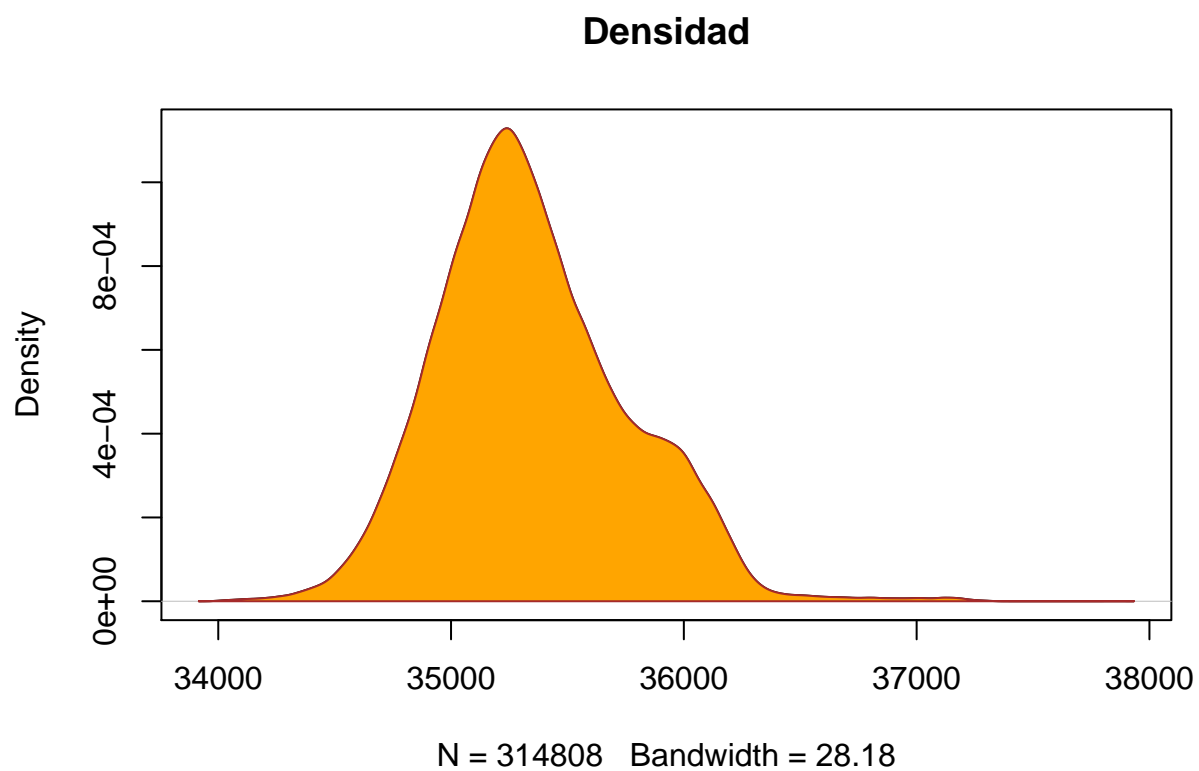


Grafico de densidad

```
d <- density(dl_temp$Value)
plot(d, main = "Densidad")
polygon(d, col = "Orange", border = "brown")
```



Boxplot separado por variable

```
bp <- ggplot(dl_temp, aes(Quantity, Value))  
bp <- bp + geom_boxplot(aes(colour = Quantity))  
bp <- bp + scale_color_brewer(palette="Dark2")  
bp
```

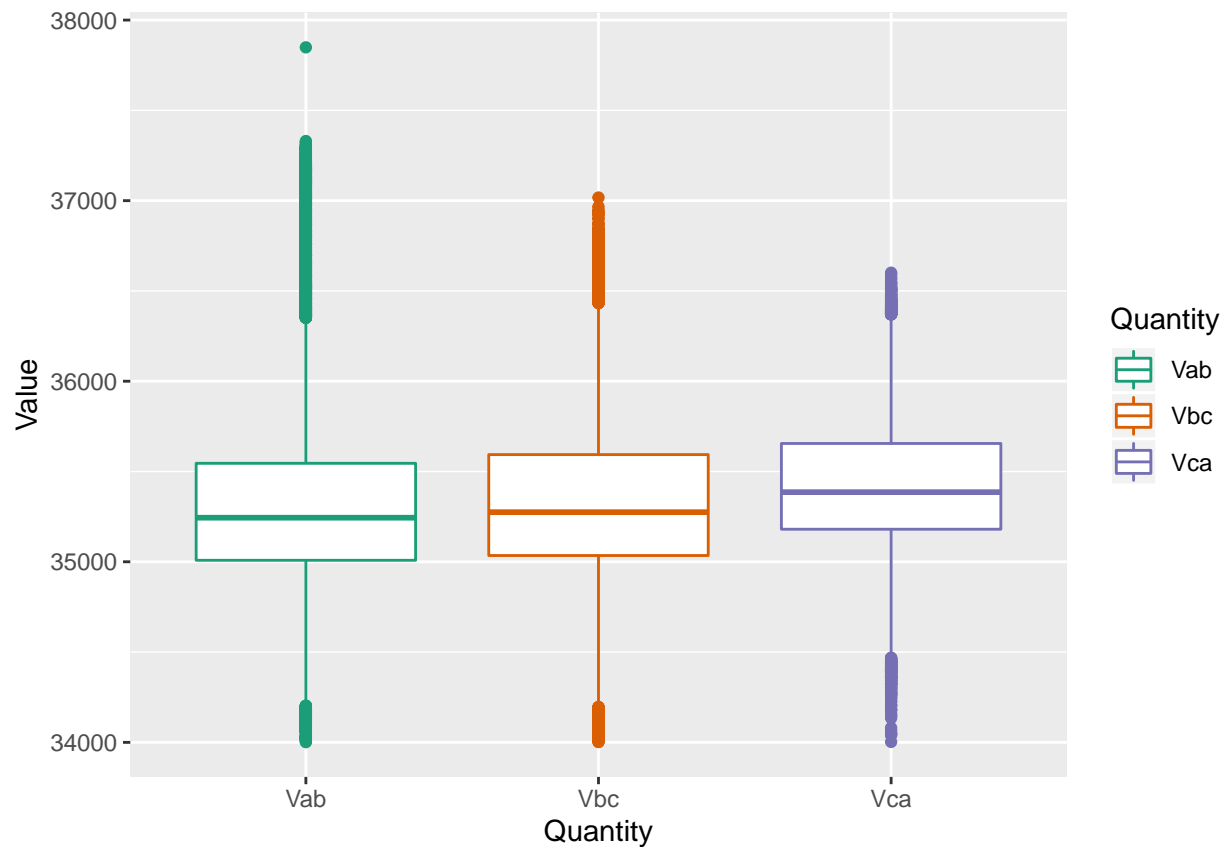
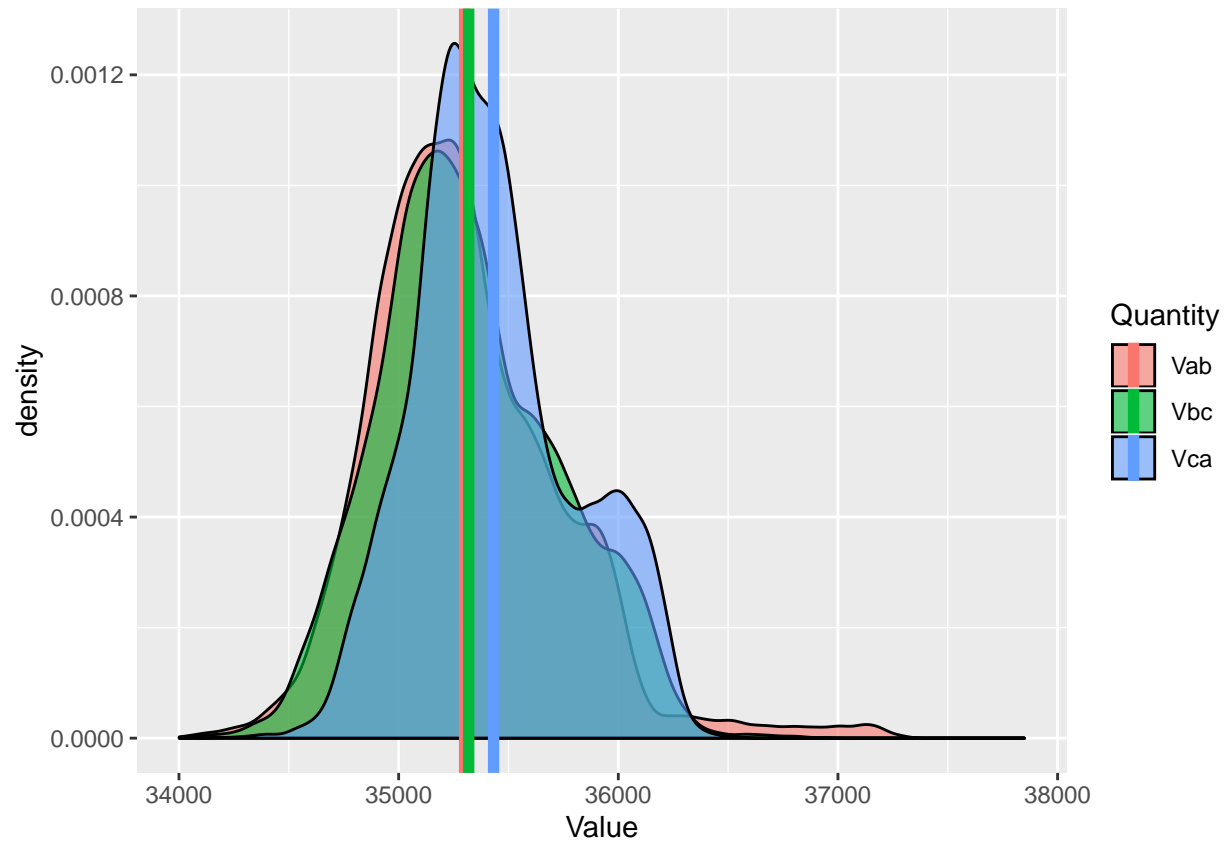


Grafico de densidad para cada variable

```
lineas <- dl_temp %>% group_by(Quantity) %>% summarise(v = mean(Value))

p <- ggplot(dl_temp, aes(x=Value, fill = Quantity)) +
  geom_density(alpha = 0.6) +
  geom_vline(data=lineas, aes(xintercept=v, color=Quantity), size = 2)
p
```



Aqui inician la modificaciones reales de los datos

```
## pivot respecto de Quantity
dataLog2 <- tidyr::spread(dataLog, Quantity, Value)

## filtro de valores menores al corte
dataLog2 <- dataLog2 %>% filter(Vab > cut_voltage, Vbc > cut_voltage, Vca > cut_voltage)

## definicion de metodos
## Calculo de desbalance maximo
unbal_calc <- function(va, vb, vc){
  maximo = pmax(abs(va-vb), abs(vb-vc), abs(vc-va))
  promedio = (va + vb + vc)/3
  unb <- 100*maximo/promedio
  return(unb)
}

## categoria del desbalance
unbal_categ <- function(unb){
  unb_class <- case_when(unb < clasif["Low"] ~ "Low",
    unb < clasif["Low_Mid"] ~ "Low_Mid",
    unb < clasif["Mid"] ~ "Mid",
```

```

        unb <- clasif["Mid_High"] ~ "Mid_High",
        unb <- clasif["High"] ~ "High",
        TRUE ~ "Very_High"
      )

    return(unb_class)
  }

  ## nuevas columnas
  dataLog2 <- dataLog2 %>%
    mutate (unbalance = unbal_calc(Vab, Vbc, Vca)) %>%
    mutate (unbal_cat = factor(unbal_categ(unbalance), levels = c("Low","Low_Mid","Mid","Mid_High","High"))
    select (year, monthName, day, hour, minute, wdayName, unbalance, unbal_cat)

  ## conversion de variables
  dataLog2$year <- as.integer(dataLog2$year)
  dataLog2$monthName <- factor(dataLog2$monthName, levels = month.abb[1:12])
  dataLog2$day <- factor(dataLog2$day, levels = c(1:31))
  dataLog2$hour <- factor(dataLog2$hour, levels = c(0:23))
  dataLog2$minute <- factor(dataLog2$minute, levels = c(0, 15, 30, 45))
  dataLog2$wdayName <- factor(dataLog2$wdayName)

  ## tabla resultado
  glimpse(dataLog2)

```

```

## Observations: 104,917
## Variables: 8
## $ year      <int> 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017, 2017,...
## $ monthName <fct> Apr, Apr, Apr, Apr, Apr, Apr, Apr, Apr, Apr, Apr, Apr, Apr, Apr,...
## $ day       <fct> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20,...
## $ hour      <fct> 0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4,...
## $ minute    <fct> 0, 15, 30, 45, 0, 15, 30, 45, 0, 15, 30, 45, 0, 15, 30, 45,...
## $ wdayName  <fct> Thursday, Thursday, Thursday, Thursday, Thursday, Thursday, Thursday,...
## $ unbalance <dbl> 0.7842734, 0.7162294, 0.7333590, 0.7271818, 0.7305882, 0.74...
## $ unbal_cat <fct> Mid_High, Mid_High, Mid_High, Mid_High, Mid_High, Mid_High,...

```

tabla de conteo de elementos en cada categoria

```
table(dataLog2$unbal_cat)
```

```
##
##      Low  Low_Mid      Mid  Mid_High      High Very_High
##      9825   27384   32487   19206   7412      8603

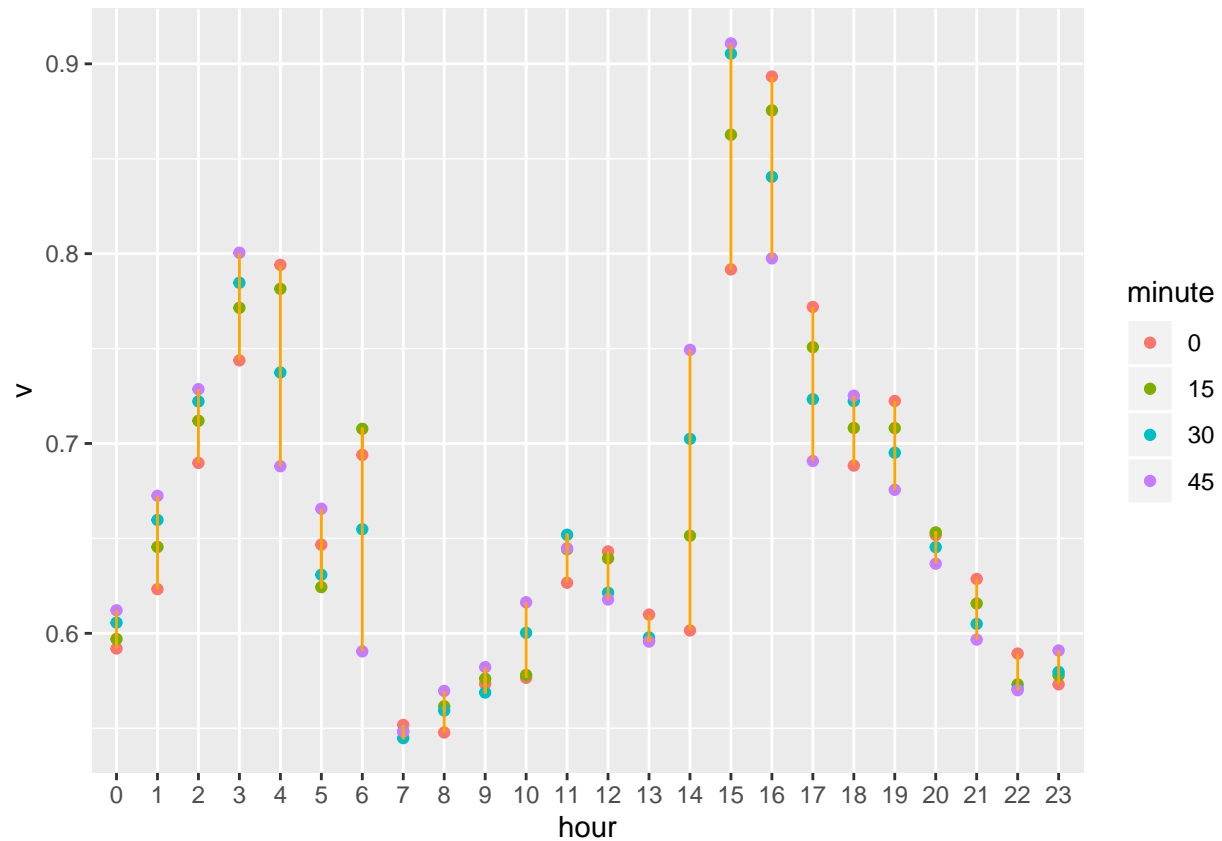
```

Desbalance promedio

```

a <- dataLog2 %>% group_by(hour, minute) %>% summarise(v = mean(unbalance))
ggplot(a, aes(hour, v)) + geom_point(aes(colour = minute)) + geom_line(col = "orange")

```



Particionamiento de los datos

```
dL <- dataLog2 %>% filter (year %in% c(2019)) %>%
  select(-unbalance)

## select(-unbalance, -day, -minute)

set.seed(4)
mascara <- sample.split(dL$unbal_cat, SplitRatio = 7/10)
training_data <- dL[mascara,]
test_data <- dL[!mascara,]

table(dL$unbal_cat)
```

```
##
##      Low  Low_Mid      Mid  Mid_High      High  Very_High
##      3265   11587   12565    4911    1158     1492
```

para obtener la importancia de la variables

Modelo 1

```

start_time <- Sys.time()

modelo <- randomForest(formula = unbal_cat ~ .,
                        data = training_data, ntree=100,
                        importance=TRUE,
                        test=test_data)

end_time <- Sys.time()
end_time - start_time

## Guardar el modelo en disco
saveRDS(modelo, file = "models/modelo2019.modelo")

```

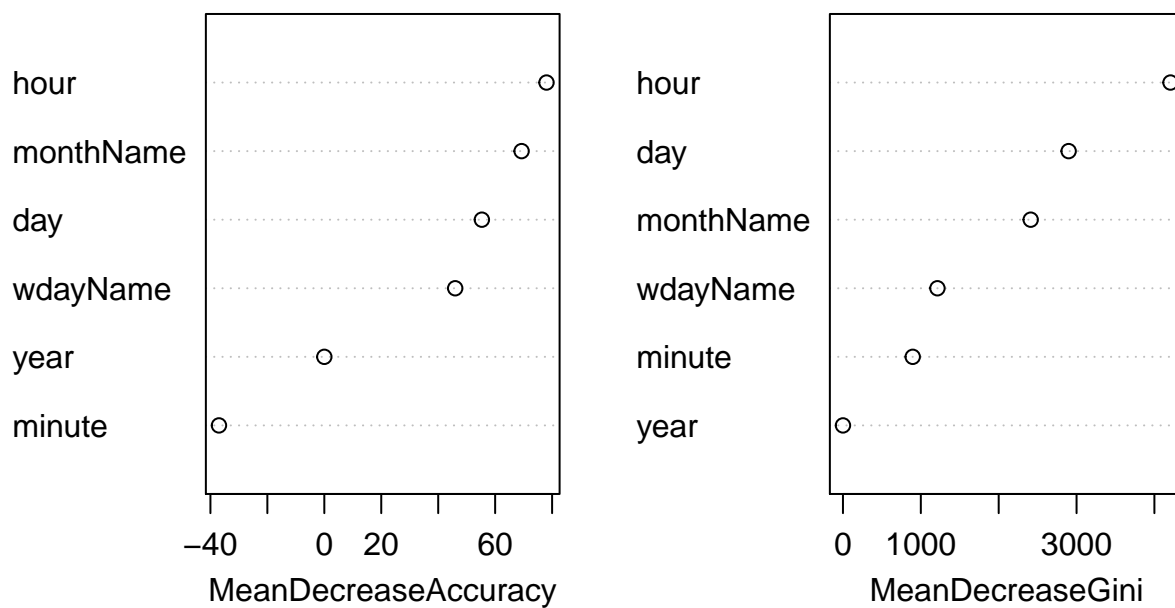
Grafica de importancia

```

modelo <- readRDS(file = "models/modelo2019.modelo")
varImpPlot(modelo)

```

modelo



Calculo del area bajo la curva

```

pred <- predict(modelo, newdata = test_data, type = "prob")
pred_f <- predict(modelo, newdata = test_data)
roc.multi <- multiclass.roc(test_data$unbal_cat, pred)

```

```
modelo1_auc <- auc(roc.multi)
modelo1_auc
```

```
## Multi-class area under the curve: 0.9189
```

```
cm1 <- confusionMatrix(test_data$unbal_cat, pred_f)
cm1
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction  Low Low_Mid  Mid Mid_High High Very_High
##   Low       389     538   51         0    0         1
##   Low_Mid   115    2607  744         9    0         1
##   Mid        4     691 2851        219    2         2
##   Mid_High    1      58  588        804   17         5
##   High        0      13   37        238   44        15
##   Very_High  0      12    6         15    7        408
```

```
##
```

```
## Overall Statistics
```

```
##
```

```
##           Accuracy : 0.677
```

```
##           95% CI : (0.6679, 0.6859)
```

```
##   No Information Rate : 0.4076
```

```
##   P-Value [Acc > NIR] : < 2.2e-16
```

```
##
```

```
##           Kappa : 0.5426
```

```
##
```

```
##   McNemar's Test P-Value : NA
```

```
##
```

```
## Statistics by Class:
```

```
##
```

```
##           Class: Low Class: Low_Mid Class: Mid Class: Mid_High
## Sensitivity           0.76424           0.6652           0.6666           0.62568
## Specificity           0.94090           0.8678           0.8523           0.92734
## Pos Pred Value        0.39734           0.7500           0.7564           0.54582
## Neg Pred Value        0.98739           0.8130           0.7879           0.94667
## Prevalence            0.04851           0.3735           0.4076           0.12247
## Detection Rate        0.03708           0.2485           0.2717           0.07663
## Detection Prevalence  0.09331           0.3313           0.3592           0.14039
## Balanced Accuracy      0.85257           0.7665           0.7594           0.77651
```

```
##           Class: High Class: Very_High
```

```
## Sensitivity           0.628571           0.94444
## Specificity           0.970927           0.99602
## Pos Pred Value        0.126801           0.91071
## Neg Pred Value        0.997437           0.99761
## Prevalence            0.006672           0.04117
## Detection Rate        0.004194           0.03889
## Detection Prevalence  0.033073           0.04270
## Balanced Accuracy      0.799749           0.97023
```


Modelo 2

Quitando la variable del modelo

```
start_time <- Sys.time()

modelo <- randomForest(formula = unbal_cat ~ . - year - minute,
                        data = training_data, ntree=100,
                        importance=TRUE,
                        test=test_data)

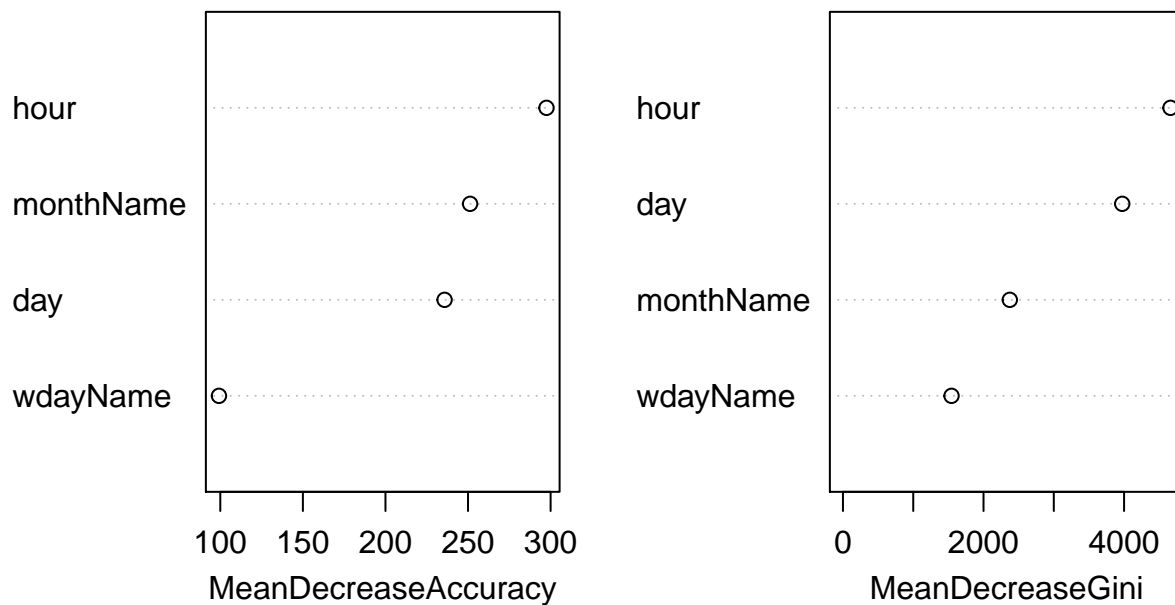
end_time <- Sys.time()
end_time - start_time

## Guardar el modelo en disco
saveRDS(modelo, file = "models/modelo2019_2.modelo")
```

Grafica de importancia

```
modelo <- readRDS(file = "models/modelo2019_2.modelo")
varImpPlot(modelo)
```

modelo



Calculo del area bajo la curva

```

pred <- predict(modelo, newdata = test_data, type = "prob")
pred_f <- predict(modelo, newdata = test_data)
roc.multi <- multiclass.roc(test_data$unbal_cat, pred)

modelo2_auc <- auc(roc.multi)
modelo2_auc

```

```
## Multi-class area under the curve: 0.9227
```

```

cm2 <- confusionMatrix(test_data$unbal_cat, pred_f)
cm2

```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

## Prediction	Low	Low_Mid	Mid	Mid_High	High	Very_High
## Low	605	347	26	1	0	0
## Low_Mid	256	2559	632	24	4	1
## Mid	18	649	2778	305	17	2
## Mid_High	3	41	432	900	94	3
## High	0	7	15	177	131	17
## Very_High	0	3	6	8	15	416

```
##
```

```
## Overall Statistics
```

```
##
```

```
##           Accuracy : 0.7043
```

```
##           95% CI : (0.6954, 0.713)
```

```
## No Information Rate : 0.3707
```

```
## P-Value [Acc > NIR] : < 2.2e-16
```

```
##
```

```
##           Kappa : 0.5913
```

```
##
```

```
## McNemar's Test P-Value : NA
```

```
##
```

```
## Statistics by Class:
```

```
##
```

##	Class: Low	Class: Low_Mid	Class: Mid	Class: Mid_High
## Sensitivity	0.68594	0.7097	0.7143	0.63604
## Specificity	0.96108	0.8668	0.8499	0.93687
## Pos Pred Value	0.61798	0.7362	0.7371	0.61100
## Neg Pred Value	0.97088	0.8508	0.8347	0.94290
## Prevalence	0.08406	0.3437	0.3707	0.13486
## Detection Rate	0.05766	0.2439	0.2648	0.08578
## Detection Prevalence	0.09331	0.3313	0.3592	0.14039
## Balanced Accuracy	0.82351	0.7882	0.7821	0.78646

```
##           Class: High Class: Very_High
```

## Sensitivity	0.50192	0.94761
## Specificity	0.97889	0.99682
## Pos Pred Value	0.37752	0.92857
## Neg Pred Value	0.98719	0.99771
## Prevalence	0.02488	0.04184
## Detection Rate	0.01249	0.03965
## Detection Prevalence	0.03307	0.04270

```
## Balanced Accuracy      0.74040      0.97221
```

Modelo 3

```
start_time <- Sys.time()

modelo <- randomForest(formula = unbal_cat ~ . - year - minute,
                      data = training_data,
                      ntree=500,
                      keep.forest=TRUE,
                      importance=TRUE,
                      test=test_data)

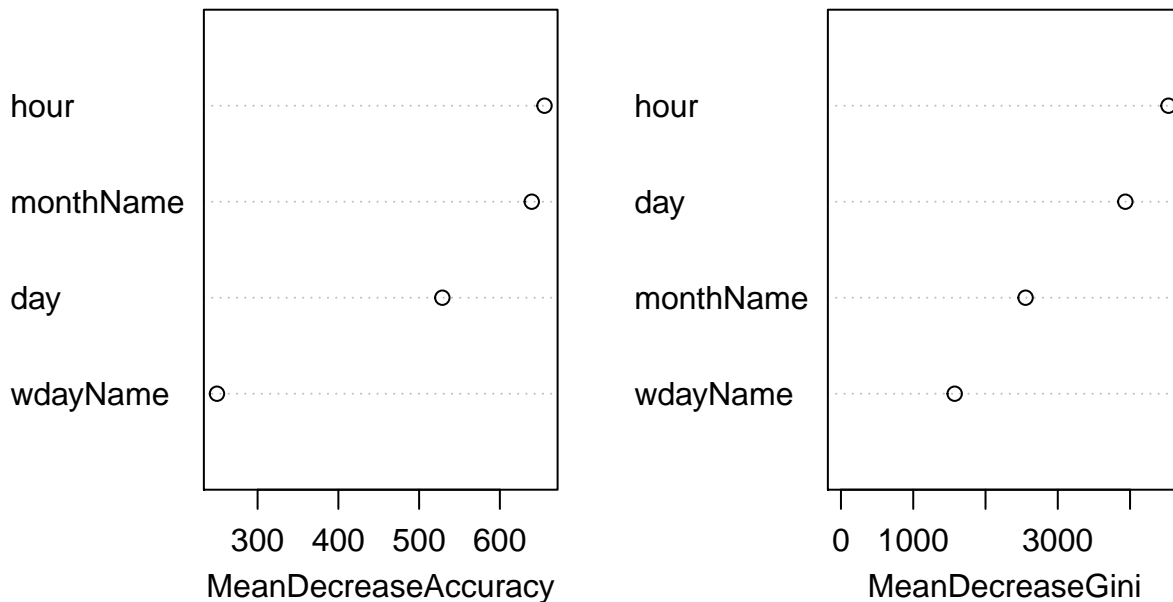
end_time <- Sys.time()
end_time - start_time

## Guardar el modelo en disco
saveRDS(modelo, file = "models/modelo2019_3.modelo")
```

Grafica de importancia

```
modelo <- readRDS(file = "models/modelo2019_3.modelo")
varImpPlot(modelo)
```

modelo



Calculo del area bajo la curva

```
pred <- predict(modelo, newdata = test_data, type = "prob")
pred_f <- predict(modelo, newdata = test_data)
roc.multi <- multiclass.roc(test_data$unbal_cat, pred)

modelo3_auc <- auc(roc.multi)
modelo3_auc
```

```
## Multi-class area under the curve: 0.9299
```

```
cm3 <- confusionMatrix(test_data$unbal_cat, pred_f)
cm3
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

## Prediction	Low	Low_Mid	Mid	Mid_High	High	Very_High
## Low	605	347	26	1	0	0
## Low_Mid	252	2556	641	22	4	1
## Mid	16	627	2792	316	16	2
## Mid_High	3	40	436	899	92	3
## High	0	5	12	179	132	19
## Very_High	0	2	5	8	15	418

```
##
```

```
## Overall Statistics
```

```
##
```

```
##           Accuracy : 0.7055
```

```
##           95% CI : (0.6967, 0.7142)
```

```
## No Information Rate : 0.3729
```

```
## P-Value [Acc > NIR] : < 2.2e-16
```

```
##
```

```
##           Kappa : 0.593
```

```
##
```

```
## McNemar's Test P-Value : NA
```

```
##
```

```
## Statistics by Class:
```

```
##
```

##	Class: Low	Class: Low_Mid	Class: Mid	Class: Mid_High
## Sensitivity	0.69064	0.7146	0.7137	0.63088
## Specificity	0.96111	0.8670	0.8515	0.93669
## Pos Pred Value	0.61798	0.7353	0.7408	0.61032
## Neg Pred Value	0.97151	0.8545	0.8334	0.94168
## Prevalence	0.08349	0.3409	0.3729	0.13582
## Detection Rate	0.05766	0.2436	0.2661	0.08568
## Detection Prevalence	0.09331	0.3313	0.3592	0.14039
## Balanced Accuracy	0.82587	0.7908	0.7826	0.78379

```
##           Class: High Class: Very_High
```

```
## Sensitivity 0.50965 | 0.94357 |
```

```
## Specificity 0.97899 | 0.99701 |
```

```
## Pos Pred Value 0.38040 | 0.93304 |
```

```
## Neg Pred Value 0.98748 | 0.99751 |
```

```
## Prevalence 0.02469 | 0.04222 |
```

```
## Detection Rate          0.01258          0.03984
## Detection Prevalence    0.03307          0.04270
## Balanced Accuracy       0.74432          0.97029
```

Comparacion de las AUC

```
print(paste("Area bajo la curva modelo 1: ", modelo1_auc))
```

```
## [1] "Area bajo la curva modelo 1: 0.918866574356504"
```

```
print(paste("Area bajo la curva modelo 2: ", modelo2_auc))
```

```
## [1] "Area bajo la curva modelo 2: 0.922737168107493"
```

```
print(paste("Area bajo la curva modelo 3: ", modelo3_auc))
```

```
## [1] "Area bajo la curva modelo 3: 0.929945486482963"
```

Comparacion de las matrices de confusion

```
cm1
```

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction  Low Low_Mid  Mid Mid_High High Very_High
```

```
##   Low       389    538   51      0      0      1
```

```
##   Low_Mid   115   2607  744      9      0      1
```

```
##   Mid        4    691 2851     219     2      2
```

```
##   Mid_High    1     58  588     804    17      5
```

```
##   High        0     13   37     238    44     15
```

```
##   Very_High   0     12    6      15     7    408
```

```
##
```

```
## Overall Statistics
```

```
##
```

```
##           Accuracy : 0.677
```

```
##           95% CI : (0.6679, 0.6859)
```

```
##   No Information Rate : 0.4076
```

```
##   P-Value [Acc > NIR] : < 2.2e-16
```

```
##
```

```
##           Kappa : 0.5426
```

```
##
```

```
##   McNemar's Test P-Value : NA
```

```
##
```

```
## Statistics by Class:
```

```
##
```

```
##           Class: Low Class: Low_Mid Class: Mid Class: Mid_High
```

```
## Sensitivity          0.76424          0.6652          0.6666          0.62568
```

```
## Specificity          0.94090          0.8678          0.8523          0.92734
```

```

## Pos Pred Value      0.39734      0.7500      0.7564      0.54582
## Neg Pred Value      0.98739      0.8130      0.7879      0.94667
## Prevalence          0.04851      0.3735      0.4076      0.12247
## Detection Rate      0.03708      0.2485      0.2717      0.07663
## Detection Prevalence 0.09331      0.3313      0.3592      0.14039
## Balanced Accuracy    0.85257      0.7665      0.7594      0.77651
##
## Class: High Class: Very_High
## Sensitivity          0.628571      0.94444
## Specificity          0.970927      0.99602
## Pos Pred Value      0.126801      0.91071
## Neg Pred Value      0.997437      0.99761
## Prevalence          0.006672      0.04117
## Detection Rate      0.004194      0.03889
## Detection Prevalence 0.033073      0.04270
## Balanced Accuracy    0.799749      0.97023

```

cm2

Confusion Matrix and Statistics

```

##
## Reference
## Prediction Low Low_Mid Mid Mid_High High Very_High
## Low        605      347   26         1      0
## Low_Mid    256     2559  632        24      4
## Mid        18       649 2778       305     17
## Mid_High   3        41  432       900     94
## High       0         7   15       177    131
## Very_High  0         3    6         8     15
##
##

```

Overall Statistics

```

##
## Accuracy : 0.7043
## 95% CI : (0.6954, 0.713)
## No Information Rate : 0.3707
## P-Value [Acc > NIR] : < 2.2e-16
##

```

```

## Kappa : 0.5913
##

```

```

## McNemar's Test P-Value : NA
##

```

Statistics by Class:

```

##
## Class: Low Class: Low_Mid Class: Mid Class: Mid_High
## Sensitivity      0.68594      0.7097      0.7143      0.63604
## Specificity      0.96108      0.8668      0.8499      0.93687
## Pos Pred Value   0.61798      0.7362      0.7371      0.61100
## Neg Pred Value   0.97088      0.8508      0.8347      0.94290
## Prevalence       0.08406      0.3437      0.3707      0.13486
## Detection Rate   0.05766      0.2439      0.2648      0.08578
## Detection Prevalence 0.09331      0.3313      0.3592      0.14039
## Balanced Accuracy 0.82351      0.7882      0.7821      0.78646
##
## Class: High Class: Very_High
## Sensitivity      0.50192      0.94761
## Specificity      0.97889      0.99682

```

```
## Pos Pred Value      0.37752      0.92857
## Neg Pred Value      0.98719      0.99771
## Prevalence          0.02488      0.04184
## Detection Rate      0.01249      0.03965
## Detection Prevalence 0.03307      0.04270
## Balanced Accuracy    0.74040      0.97221
```

cm3

Confusion Matrix and Statistics

```
##
##               Reference
## Prediction   Low Low_Mid Mid Mid_High High Very_High
## Low          605    347   26         1    0         0
## Low_Mid      252   2556  641        22    4         1
## Mid           16    627 2792       316   16         2
## Mid_High      3     40  436       899   92         3
## High          0      5   12       179  132        19
## Very_High     0      2    5         8   15       418
```

Overall Statistics

```
##
##               Accuracy : 0.7055
##               95% CI : (0.6967, 0.7142)
##       No Information Rate : 0.3729
##       P-Value [Acc > NIR] : < 2.2e-16
##
##               Kappa : 0.593
##
## Mcnemar's Test P-Value : NA
```

Statistics by Class:

```
##
##               Class: Low Class: Low_Mid Class: Mid Class: Mid_High
## Sensitivity      0.69064      0.7146      0.7137      0.63088
## Specificity      0.96111      0.8670      0.8515      0.93669
## Pos Pred Value   0.61798      0.7353      0.7408      0.61032
## Neg Pred Value   0.97151      0.8545      0.8334      0.94168
## Prevalence       0.08349      0.3409      0.3729      0.13582
## Detection Rate   0.05766      0.2436      0.2661      0.08568
## Detection Prevalence 0.09331      0.3313      0.3592      0.14039
## Balanced Accuracy 0.82587      0.7908      0.7826      0.78379
##
##               Class: High Class: Very_High
## Sensitivity      0.50965      0.94357
## Specificity      0.97899      0.99701
## Pos Pred Value   0.38040      0.93304
## Neg Pred Value   0.98748      0.99751
## Prevalence       0.02469      0.04222
## Detection Rate   0.01258      0.03984
## Detection Prevalence 0.03307      0.04270
## Balanced Accuracy 0.74432      0.97029
```