Entrega final

Bryan Henao, J. Steven Delgado V.

11/29/2017

```
## corrplot 0.84 loaded
```

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

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

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

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

```
## Loading required package: grid
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2017c.
## 1.0/zoneinfo/America/Bogota'
```

Descripción

El dataset que presentaremos a continuación fue tomado por los estudiantes de proyecto de grado en el semestre 17-1, este dataset representa 1779 registros de tráfico web (filas), junto con 27 variables más significativas (columnas). La última variable llamada ***tipo***, nos muestra la clasificación de si cada registro de tráfico es maligno o benigno. A método de ilustrar el dataset hemos tomado las 28 variables y cada una con sus medidas de tendencia central:

```
summary(dataset_proyecto)
```

```
##       URL          URL_LENGTH     NUMBER_SPECIAL_CHARACTERS
##  B0_1   :   1   Min.   : 16.00   Min.   : 5.00
##  B0_10  :   1   1st Qu.: 39.00   1st Qu.: 8.00
##  B0_100 :   1   Median : 49.00   Median :10.00
##  B0_1000:   1   Mean   : 56.94   Mean   :11.11
##  B0_1001:   1   3rd Qu.: 68.00   3rd Qu.:13.00
##  B0_1002:   1   Max.   :249.00   Max.   :43.00
##  (Other):1773
```

```
##          CHARSET                     SERVER
##  ISO-8859    :   1   APACHE           :622
##  ISO-8859-1  : 561   NGINX            :337
##  US-ASCII    : 154   MICROSOFT-HTTPAPI:113
##  UTF-8       :1054   CLOUDFLARE-NGINX : 94
##  WINDOWS-1251:   1   MICROSOFT-IIS    : 85
##  WINDOWS-1252:   1   (Other)          :352
##  NA's        :   7   NA's             :176
##                                                    CACHE_CONTROL
##  NO-CACHE                                                      :194
##  NO-STORE, NO-CACHE, MUST-REVALIDATE, POST-CHECK=0, PRE-CHECK=0:166
##  PRIVATE, S-MAXAGE=0, MAX-AGE=0, MUST-REVALIDATE               : 71
##  PRIVATE                                                       : 58
##  PRIVATE, NO-CACHE, NO-STORE, MUST-REVALIDATE                  : 52
##  (Other)                                                       :469
##  NA's                                                          :769
##  CONTENT_LENGTH   WHOIS_COUNTRY   WHOIS_STATEPROV         WHOIS_REGDATE
##  Min.   :     0   US     :1106   CA       :376    17/09/2008 0:00:  62
##  1st Qu.:   324   CA     :  84   NY       : 76    13/01/2001 0:12:  59
##  Median :  1853   ES     :  63   WA       : 65    31/07/2000 0:00:  47
##  Mean   : 11732   AU     :  35   BARCELONA: 62    15/02/2005 0:00:  41
##  3rd Qu.: 11283   PA     :  21   FL       : 61    29/03/1997 0:00:  33
##  Max.   :649263   (Other): 165   (Other)  :777    (Other)        :1410
##  NA's   :812      NA's   : 305   NA's     :362    NA's           : 127
##            UPDATE_DATE            WHITIN_DOMAIN  TCP_CONVERSATION_EXCHANGE
##  2/09/2016 0:00  :  64   COLEYGLESIAS.COM:  62   Min.   :   0.00
##  12/12/2015 10:16:  59   WIKIPEDIA.ORG   :  59   1st Qu.:   0.00
##  29/06/2016 0:00 :  47   BLOGSPOT.COM    :  47   Median :   7.00
##  14/01/2017 0:00 :  42   YOUTUBE.COM     :  42   Mean   :  16.23
##  29/11/2016 0:00 :  36   FACEBOOK.COM    :  33   3rd Qu.:  22.00
##  (Other)         :1392   AMAZON.COM      :  29   Max.   :1194.00
##  NA's            : 139   (Other)         :1507
##  DIST_REMOTE_TCP_PORT   REMOTE_IPS       APP_BYTES         UDP_PACKETS
##  Min.   :  0.000      Min.   : 0.000   Min.   :      0   Min.   :0.00000
##  1st Qu.:  0.000      1st Qu.: 0.000   1st Qu.:      0   1st Qu.:0.00000
##  Median :  0.000      Median : 2.000   Median :    672   Median :0.00000
##  Mean   :  5.479      Mean   : 3.063   Mean   :   2982   Mean   :0.01012
##  3rd Qu.:  5.000      3rd Qu.: 5.000   3rd Qu.:   2327   3rd Qu.:0.00000
##  Max.   :708.000      Max.   :17.000   Max.   :2362906   Max.   :1.00000
##
##  TCP_URG_PACKETS SOURCE_APP_PACKETS REMOTE_APP_PACKETS SOURCE_APP_BYTES
##  Min.   :0       Min.   :   0.00    Min.   :   0.00    Min.   :      0
##  1st Qu.:0       1st Qu.:   0.00    1st Qu.:   0.00    1st Qu.:      0
##  Median :0       Median :   8.00    Median :   9.00    Median :    579
##  Mean   :0       Mean   :  18.51    Mean   :  18.71    Mean   :  15840
##  3rd Qu.:0       3rd Qu.:  26.00    3rd Qu.:  25.00    3rd Qu.:   9762
##  Max.   :0       Max.   :1198.00    Max.   :1284.00    Max.   :2060012
##
##  REMOTE_APP_BYTES     DURATION         AVG_LOCAL_PKT_RATE
##  Min.   :      0   Min.   :1.263e+03   Min.   :0.0001
```

```
##  1st Qu.:      0   1st Qu.:3.119e+06   1st Qu.:0.0001
##  Median :    735   Median :8.025e+06   Median :0.0001
##  Mean   :   3155   Mean   :1.036e+08   Mean   :0.0046
##  3rd Qu.:   2696   3rd Qu.:1.404e+07   3rd Qu.:0.0065
##  Max.   :2362906   Max.   :4.295e+09   Max.   :0.0238
##                    NA's   :1147        NA's   :1770
##  AVG_REMOTE_PKT_RATE  APP_PACKETS      DNS_QUERY_TIMES       TIPO
##  Min.   :0.0001      Min.   :   0.00   Min.   : 0.000   BENIGNA:1563
##  1st Qu.:0.0001      1st Qu.:   0.00   1st Qu.: 0.000   MALIGNA: 216
##  Median :0.0001      Median :   8.00   Median : 0.000
##  Mean   :0.0037      Mean   :  18.51   Mean   : 2.264
##  3rd Qu.:0.0045      3rd Qu.:  26.00   3rd Qu.: 4.000
##  Max.   :0.0206      Max.   :1198.00   Max.   :20.000
##  NA's   :1769                          NA's   :1
```

Como podemos observar, algunas columnas tienen muchos NA's en sus registros. Tomemos por ejemplo ***AVG_LOCAL_PKT_RATE*** y ***AVG_REMOTE_PKT_RATE*** las cuales tienen un 99.49% de Na's, no obstante las columnas ***UDP_PACKETS***, ***TCP_URG_PACKETS***, están en 0, en el caso de ***CACHE_CONTROL***, ***DURATION***, ***CONTENT_LENGTH*** su número de NA's oscilan entre 40% y 60%, finalmente ***WITHIN_DOMAIN*** tiene un 84.71% de los datos en la categoría ***otros***, debido a que la cantidad de NA's,número de 0's,la variable ***otros***,es muy grande, procedemos a eliminar estas columnas ya que no serán relevantes en nuestro análisis estadístico, esto incluye también a todas las variables cuyos registros son fechas.

Luego de eliminar estas columnas, quedamos con un total de 16 columnas:

```
dataset_proyecto_tipos[0,]
```

```
##  [1] URL                         URL_LENGTH
##  [3] NUMBER_SPECIAL_CHARACTERS CHARSET
##  [5] SERVER                      TCP_CONVERSATION_EXCHANGE
##  [7] DIST_REMOTE_TCP_PORT      REMOTE_IPS
##  [9] APP_BYTES                   SOURCE_APP_PACKETS
## [11] REMOTE_APP_PACKETS        SOURCE_APP_BYTES
## [13] REMOTE_APP_BYTES          APP_PACKETS
## [15] DNS_QUERY_TIMES           TIPO
## <0 rows> (or 0-length row.names)
```

Observando ahora ***DNS_QUERY_TIMES*** respectivamente tenemos:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.000   0.000   2.264   4.000  20.000       1
```

Debido a que hay pocos NA's con respecto a el número total de la muestra, podemos predecir estos reemplazando por la mediana

Ahora tenemos:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   2.263   4.000  20.000
```

Gráficas

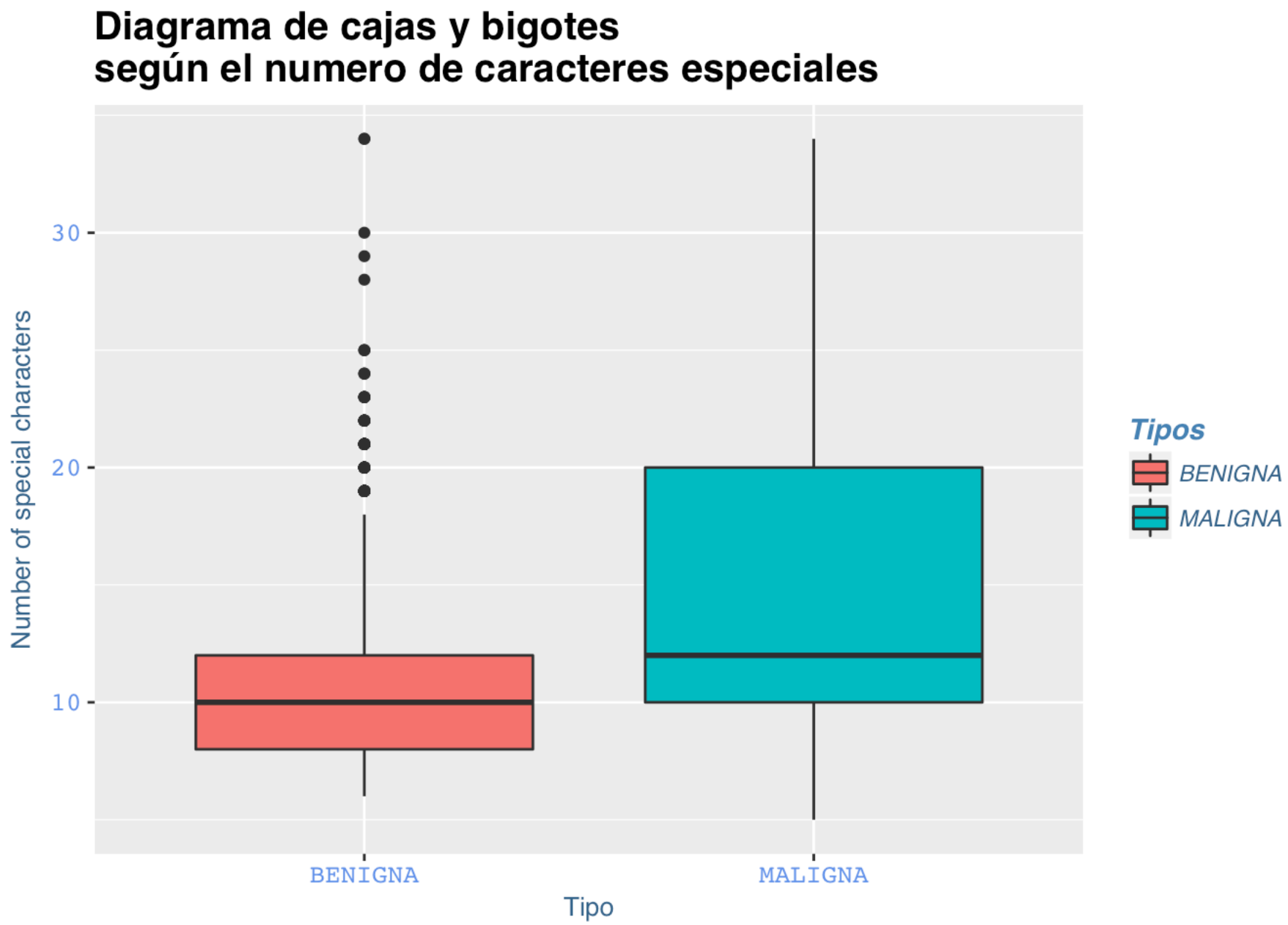
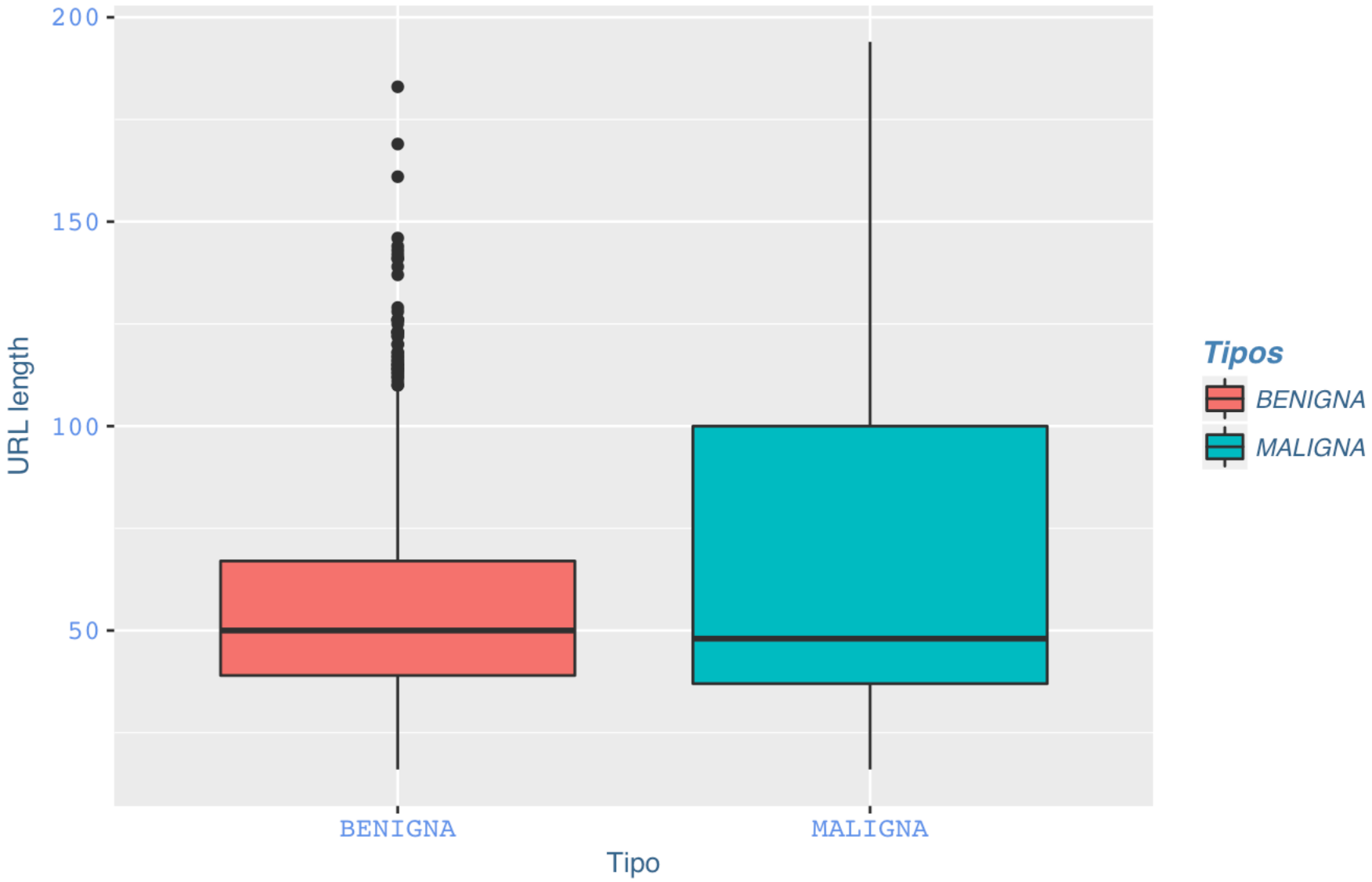
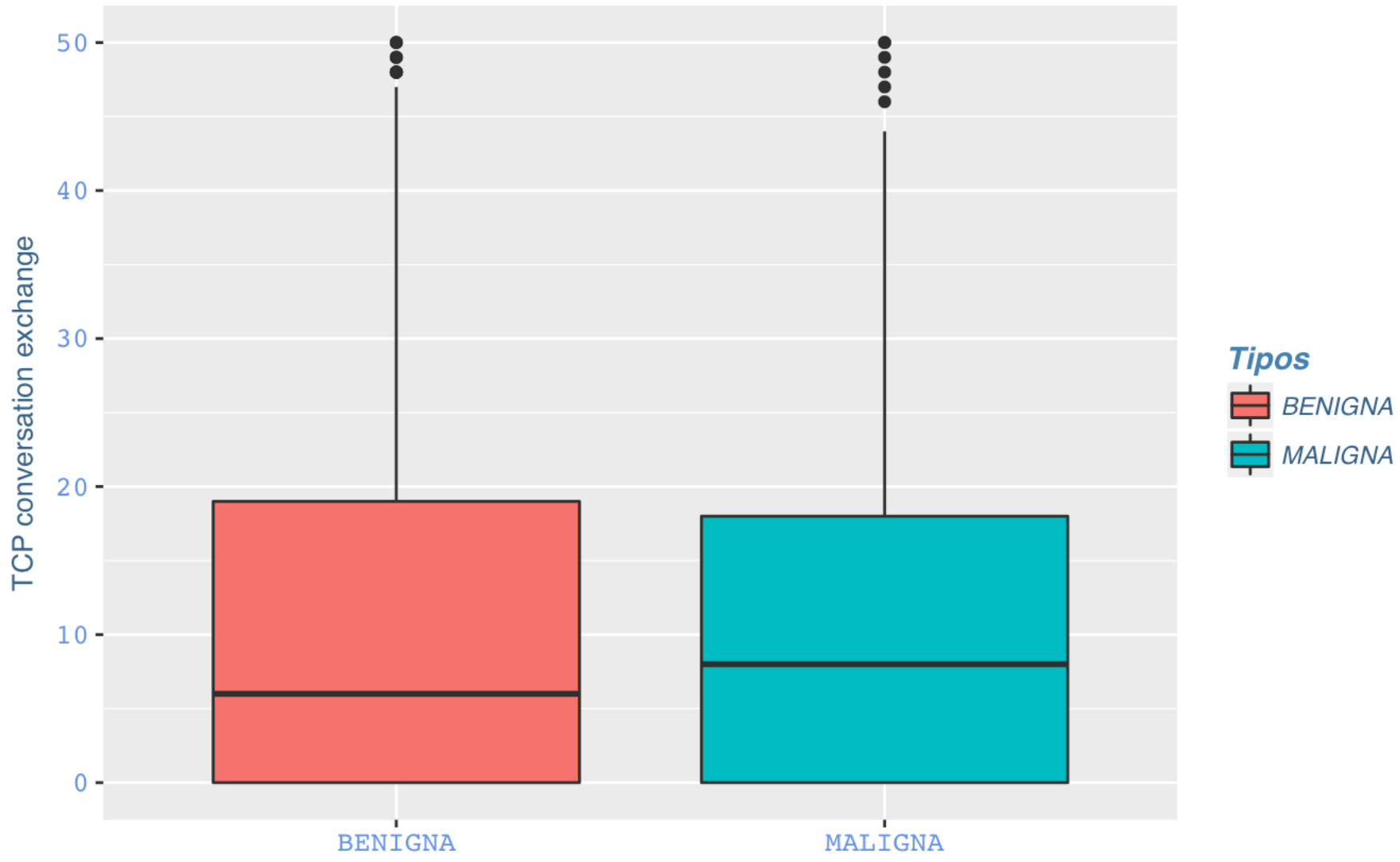
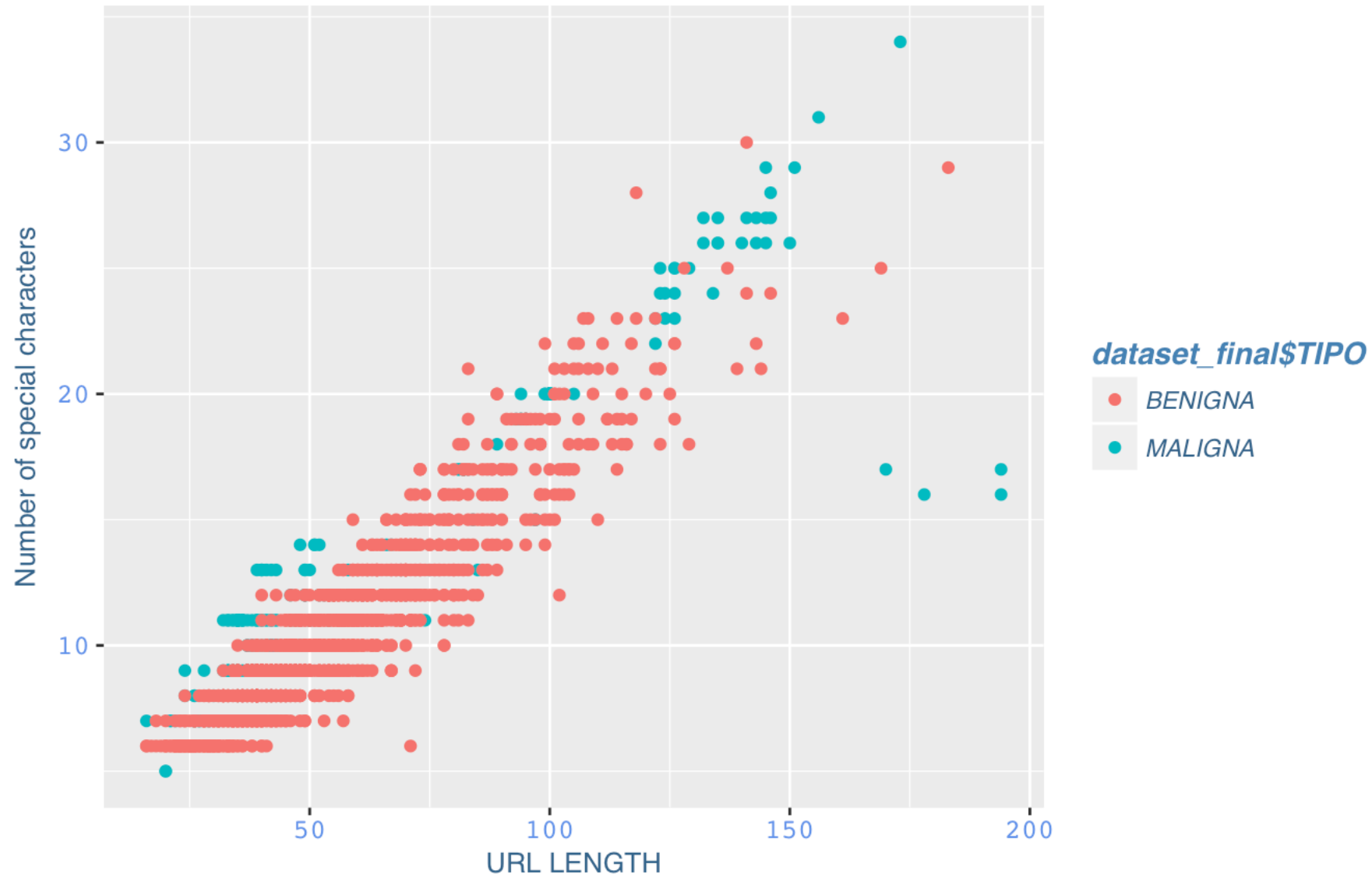


Diagrama de cajas y bigotes según la longitud URL



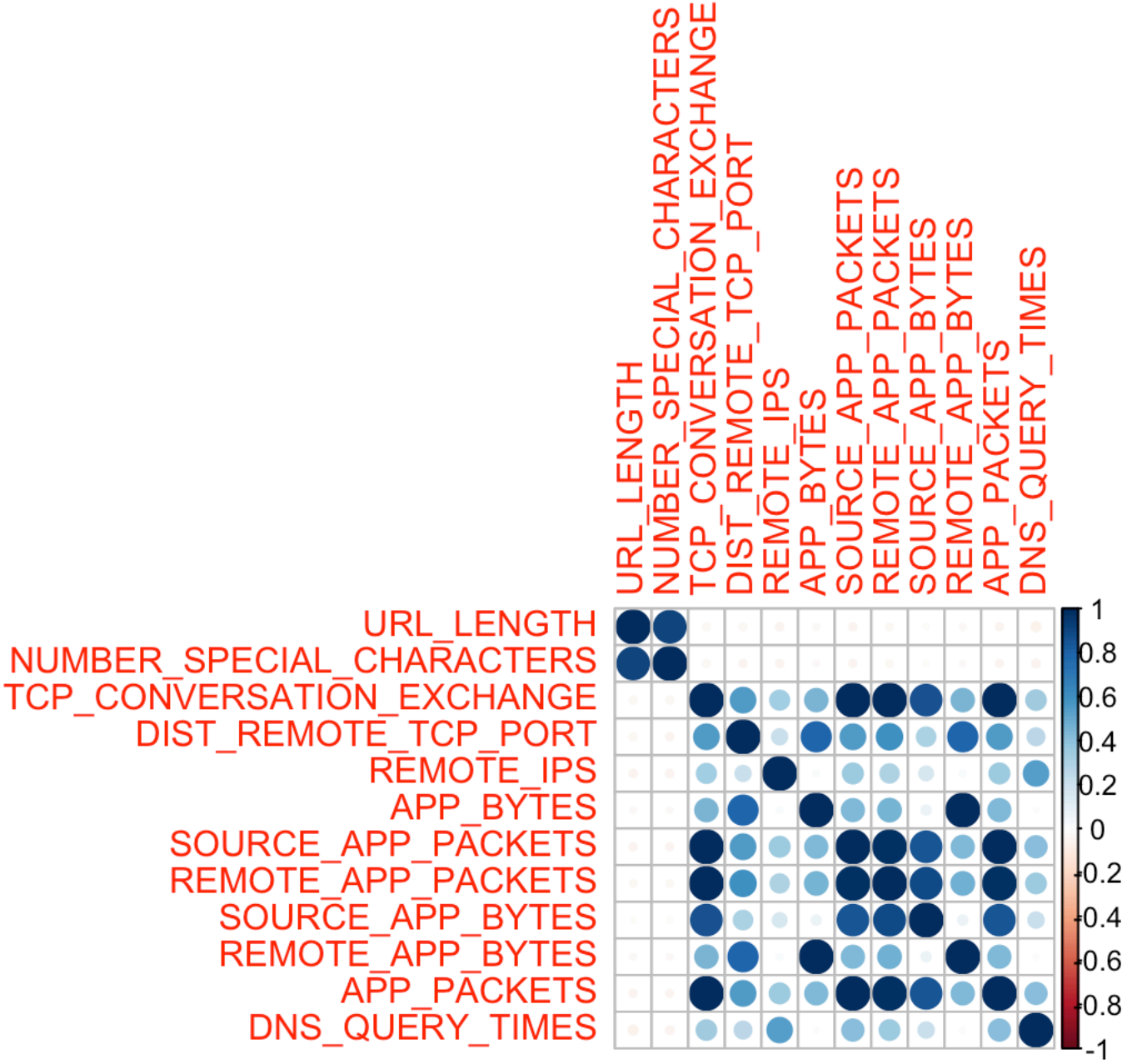


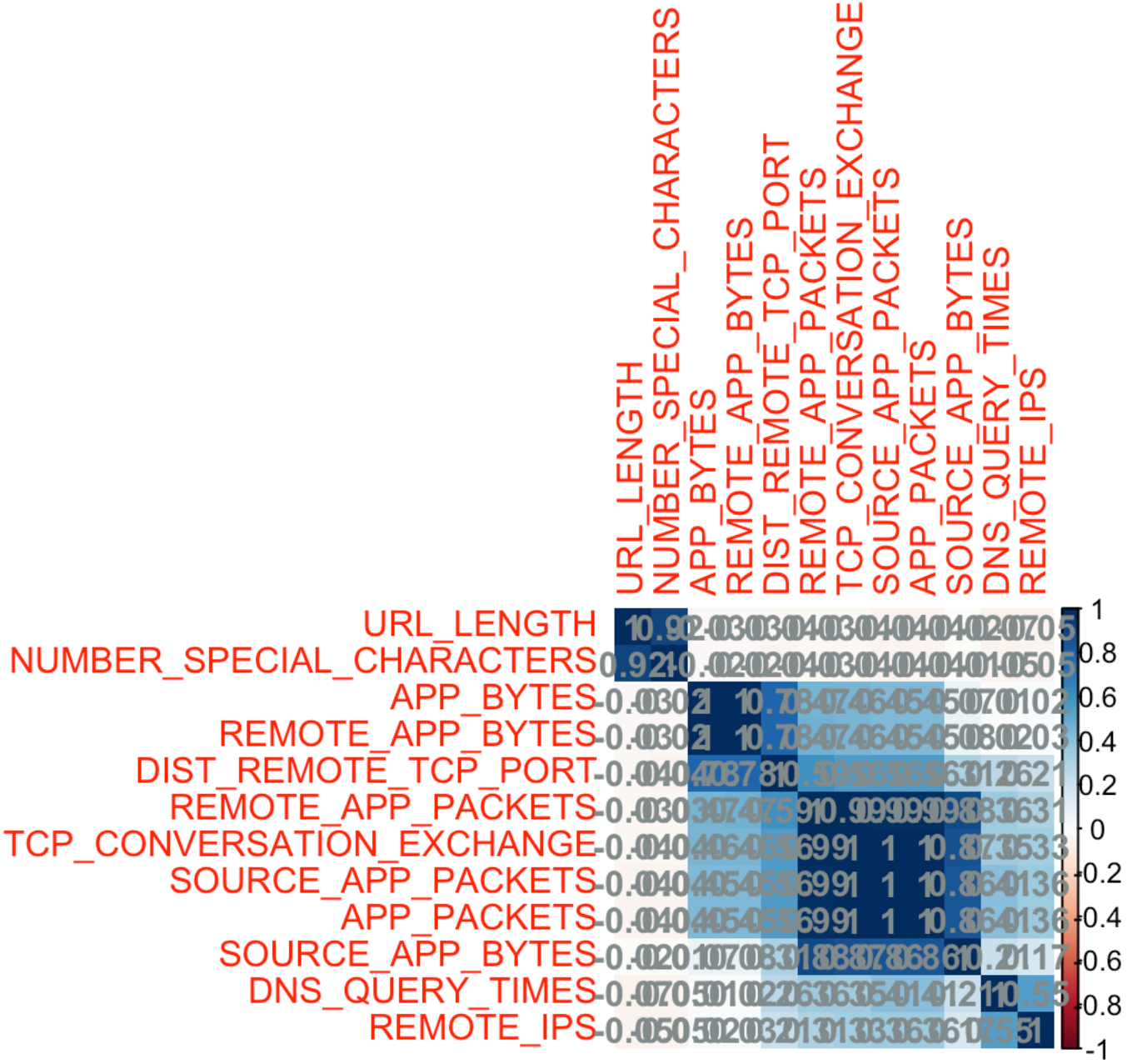
procedemos a hacer scatter plots para tratar de encontrar una correlación dadas dos variables:

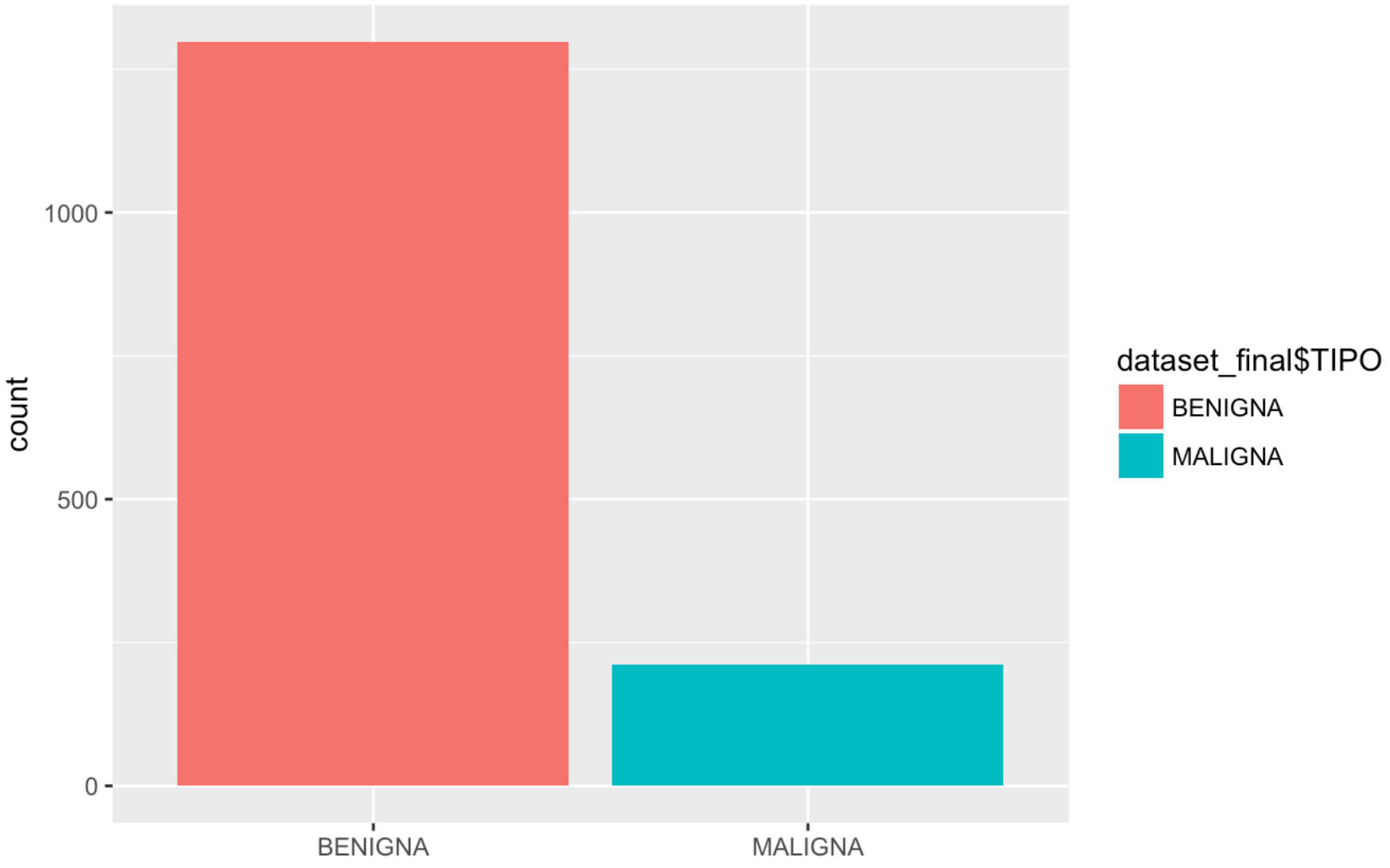


Aplicamos también un correlograma:

```
##                             URL_LENGTH NUMBER_SPECIAL_CHARACTERS
## URL_LENGTH                   1.0000000                 0.9179933
## NUMBER_SPECIAL_CHARACTERS    0.9179933                 1.0000000
##                             TCP_CONVERSATION_EXCHANGE DIST_REMOTE_TCP_PORT
## URL_LENGTH                                -0.0387715          -0.03967407
## NUMBER_SPECIAL_CHARACTERS                 -0.0375213          -0.04250856
##                              REMOTE_IPS   APP_BYTES SOURCE_APP_PACKETS
## URL_LENGTH                -0.04594146 -0.02645086        -0.04261470
## NUMBER_SPECIAL_CHARACTERS -0.04676423 -0.02390452        -0.04014473
##                             REMOTE_APP_PACKETS SOURCE_APP_BYTES
## URL_LENGTH                         -0.03421252      -0.01526726
## NUMBER_SPECIAL_CHARACTERS          -0.03068334      -0.01448022
##                             REMOTE_APP_BYTES APP_PACKETS DNS_QUERY_TIMES
## URL_LENGTH                       -0.02668971 -0.04261470     -0.06913071
## NUMBER_SPECIAL_CHARACTERS        -0.02408815 -0.04014473     -0.05020928
```


URL_LENGTH
NUMBER_SPECIAL_CHARACTERS
TCP_CONVERSATION_EXCHANGE
DIST_REMOTE_TCP_PORT
REMOTE_IPS
APP_BYTES
SOURCE_APP_PACKETS
REMOTE_APP_PACKETS
SOURCE_APP_BYTES
REMOTE_APP_BYTES
APP_PACKETS
DNS_QUERY_TIMES
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


URL_LENGTH
NUMBER_SPECIAL_CHARACTERS
APP_BYTES
REMOTE_APP_BYTES
DIST_REMOTE_TCP_PORT
REMOTE_APP_PACKETS
TCP_CONVERSATION_EXCHANGE
SOURCE_APP_PACKETS
APP_PACKETS
SOURCE_APP_BYTES
DNS_QUERY_TIMES
REMOTE_IPS
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1




```
##
## BENIGNA MALIGNA
##    1297     211
```

```
##
##   BENIGNA   MALIGNA
## 0.8600796 0.1399204
```

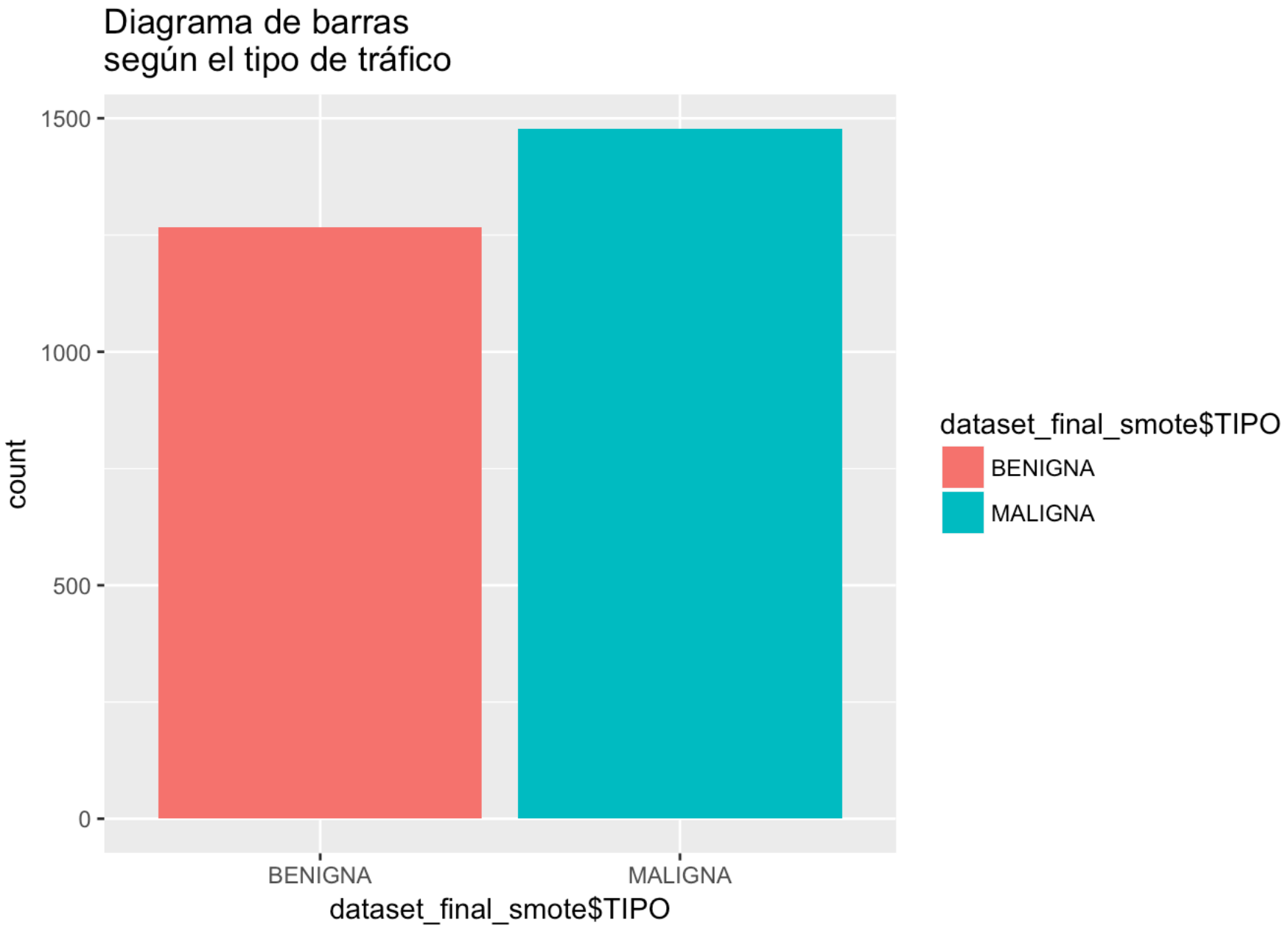
Como podemos observar, tenemos un dataset bastante desbalanceado, para remediar esto procedemos a utilizar SMOTE (*synthetic minority oversampling technique*)

```
#Aplicando SMOTE
dataset_final_smote<- SMOTE(TIPO ~ ., dataset_final, perc.over = 600, perc.under=100)
table(dataset_final_smote$TIPO)
```

```
##
## BENIGNA MALIGNA
##    1266    1477
```

```
prop.table(table(dataset_final_smote$TIPO))
```

```
##
##   BENIGNA   MALIGNA
## 0.4615385 0.5384615
```



Observamos pues, que hemos balanceado de una forma notoria el dataset. Procedemos a particionarlo, y luego a entrenar los algoritmo de clasificación.

```
set.seed(1234)
splitIndex <- createDataPartition(dataset_final$TIPO, p = .70,
                                  list = FALSE,
                                  times = 1)
trainSplit <- dataset_final[ splitIndex,]
testSplit <- dataset_final[-splitIndex,]

trainSplitSmote <- dataset_final_smote[ splitIndex,]
testSplitSmote <- dataset_final_smote[-splitIndex,]
```

Primero, Árbol con poda:

```
## Arbol con poda
library(rpart)
set.seed(9999)

#Modelo
arbolPoda <- train(TIPO~., data = trainSplit, method = "rpart")
predicciones <- predict(arbolPoda, newdata=testSplit)
confusionMatrix(predicciones,testSplit$TIPO)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA     383      48
##    MALIGNA       6      15
##
##                Accuracy : 0.8805
##                  95% CI : (0.847, 0.909)
##     No Information Rate : 0.8606
##     P-Value [Acc > NIR] : 0.1229
##
##                   Kappa : 0.309
##  Mcnemar's Test P-Value : 2.414e-08
##
##             Sensitivity : 0.9846
##             Specificity : 0.2381
##          Pos Pred Value : 0.8886
##          Neg Pred Value : 0.7143
##              Prevalence : 0.8606
##          Detection Rate : 0.8473
##    Detection Prevalence : 0.9535
##       Balanced Accuracy : 0.6113
##
##        'Positive' Class : BENIGNA
##
```

```
modelos[2,1]<-as.numeric(data.frame(as.list(confusionMatrix(predicciones,testSplit$TI
PO)$overall))[1])

#Modelo con SMOTE
arbolPodaSmote <- train(TIPO~., data = trainSplitSmote, method = "rpart")

prediccionesSmote <- predict(arbolPodaSmote, newdata=testSplitSmote)
confusionMatrix(prediccionesSmote,testSplitSmote$TIPO)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA     361     239
##    MALIGNA      20    1067
##
##                Accuracy : 0.8465
##                  95% CI : (0.8284, 0.8634)
##     No Information Rate : 0.7742
##     P-Value [Acc > NIR] : 7.53e-14
##
##                   Kappa : 0.6352
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9475
##             Specificity : 0.8170
##          Pos Pred Value : 0.6017
##          Neg Pred Value : 0.9816
##              Prevalence : 0.2258
##          Detection Rate : 0.2140
##    Detection Prevalence : 0.3557
##       Balanced Accuracy : 0.8823
##
##        'Positive' Class : BENIGNA
##
```

```
modelos[2,2]<-as.numeric(data.frame(as.list(confusionMatrix(prediccionesSmote,testSpl
itSmote$TIPO)$overall))[1])
```

Segundo, Árbol C 4.5:

```
##C 4.5

library(RWeka)
set.seed(9876)

c4_5<-J48(TIPO~.,data=trainSplit)
summary(c4_5)
```

```
##
## === Summary ===
##
## Correctly Classified Instances        1022               96.7803 %
## Incorrectly Classified Instances        34                3.2197 %
## Kappa statistic                          0.8558
## Mean absolute error                      0.045
## Root mean squared error                  0.1525
## Relative absolute error                 18.6163 %
## Root relative squared error             43.9264 %
## Total Number of Instances             1056
##
## === Confusion Matrix ===
##
##    a   b   <-- classified as
##  904   4 |   a = BENIGNA
##   30 118 |   b = MALIGNA
```

```
predicciones<- predict(c4_5, testSplit[,-16])
confusionMatrix(predicciones,testSplit$TIPO)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA     381      21
##    MALIGNA       8      42
##
##                Accuracy : 0.9358
##                  95% CI : (0.9092, 0.9566)
##     No Information Rate : 0.8606
##     P-Value [Acc > NIR] : 3.101e-07
##
##                   Kappa : 0.7073
##  Mcnemar's Test P-Value : 0.02586
##
##             Sensitivity : 0.9794
##             Specificity : 0.6667
##          Pos Pred Value : 0.9478
##          Neg Pred Value : 0.8400
##              Prevalence : 0.8606
##          Detection Rate : 0.8429
##    Detection Prevalence : 0.8894
##       Balanced Accuracy : 0.8231
##
##        'Positive' Class : BENIGNA
##
```

```
modelos[3,1]<-as.numeric(data.frame(as.list(confusionMatrix(predicciones,testSplit$TI
PO)$overall))[1])


c4_5smote<-J48(TIPO~.,data=trainSplitSmote)
summary(c4_5smote)
```

```
##
## === Summary ===
##
## Correctly Classified Instances         927               87.7841 %
## Incorrectly Classified Instances       129               12.2159 %
## Kappa statistic                          0.6447
## Mean absolute error                      0.1464
## Root mean squared error                  0.2705
## Relative absolute error                 53.8475 %
## Root relative squared error             73.4391 %
## Total Number of Instances             1056
##
## === Confusion Matrix ===
##
##    a   b   <-- classified as
##  764 121 |   a = BENIGNA
##    8 163 |   b = MALIGNA
```

```
prediccionesSmote<- predict(c4_5smote, testSplitSmote[,-16])
confusionMatrix(prediccionesSmote,testSplitSmote$TIPO)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA     325      59
##    MALIGNA      56    1247
##
##               Accuracy : 0.9318
##                 95% CI : (0.9187, 0.9434)
##     No Information Rate : 0.7742
##     P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.8056
##  Mcnemar's Test P-Value : 0.8521
##
##             Sensitivity : 0.8530
##             Specificity : 0.9548
##          Pos Pred Value : 0.8464
##          Neg Pred Value : 0.9570
##              Prevalence : 0.2258
##          Detection Rate : 0.1926
##    Detection Prevalence : 0.2276
##       Balanced Accuracy : 0.9039
##
##        'Positive' Class : BENIGNA
##
```

```
modelos[3,2]<-as.numeric(data.frame(as.list(confusionMatrix(prediccionesSmote,testSpl
itSmote$TIPO)$overall))[1])
```

Tercero, Naïve Bayes:

```
##Naïve Bayes

set.seed(4444)
#NB
modelo_nb <- train(trainSplit[,-16], trainSplit$TIPO, method = "nb")
modelo_nb
```

```
## Naive Bayes
##
## 1056 samples
##   15 predictor
##    2 classes: 'BENIGNA', 'MALIGNA'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1056, 1056, 1056, 1056, 1056, 1056, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.8031400  0.3747184
##    TRUE      0.8799866  0.5350688
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE
##  and adjust = 1.
```

```
class(modelo_nb$finalModel)
```

```
## [1] "NaiveBayes"
```

```
predicciones <- predict(modelo_nb, testSplit[,-16])
confusionMatrix(predicciones,testSplit$TIPO)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA     353      22
##    MALIGNA      36      41
##
##               Accuracy : 0.8717
##                 95% CI : (0.8373, 0.9011)
##     No Information Rate : 0.8606
##     P-Value [Acc > NIR] : 0.27399
##
##                  Kappa : 0.5107
##  Mcnemar's Test P-Value : 0.08783
##
##             Sensitivity : 0.9075
##             Specificity : 0.6508
##          Pos Pred Value : 0.9413
##          Neg Pred Value : 0.5325
##              Prevalence : 0.8606
##          Detection Rate : 0.7810
##    Detection Prevalence : 0.8296
##       Balanced Accuracy : 0.7791
##
##        'Positive' Class : BENIGNA
##
```

```
modelos[4,1]<-as.numeric(data.frame(as.list(confusionMatrix(predicciones,testSplit$TI
PO)$overall))[1])

#NB con SMOTE
modelo_nbSmote <- train(trainSplitSmote[,-16], trainSplitSmote$TIPO, method = "nb")
modelo_nbSmote
```

```
## Naive Bayes
##
## 1056 samples
##   15 predictor
##    2 classes: 'BENIGNA', 'MALIGNA'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1056, 1056, 1056, 1056, 1056, 1056, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.8495051  0.4853814
##    TRUE      0.8589143  0.5039154
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE
##  and adjust = 1.
```

```
class(modelo_nbSmote$finalModel)
```

```
## [1] "NaiveBayes"
```

```
prediccionesSmote <- predict(modelo_nbSmote, testSplitSmote[,-16])
confusionMatrix(prediccionesSmote,testSplitSmote$TIPO)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction BENIGNA MALIGNA
##    BENIGNA     338     354
##    MALIGNA      43     952
##
##               Accuracy : 0.7647
##                 95% CI : (0.7437, 0.7847)
##     No Information Rate : 0.7742
##     P-Value [Acc > NIR] : 0.8318
##
##                  Kappa : 0.4779
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8871
##             Specificity : 0.7289
##          Pos Pred Value : 0.4884
##          Neg Pred Value : 0.9568
##              Prevalence : 0.2258
##          Detection Rate : 0.2004
##    Detection Prevalence : 0.4102
##       Balanced Accuracy : 0.8080
##
##        'Positive' Class : BENIGNA
##
```

```
modelos[4,2]<-as.numeric(data.frame(as.list(confusionMatrix(prediccionesSmote,testSpl
itSmote$TIPO)$overall))[1])
```

Finalmente, comparamos los modelos utilizados para seleccionar el mejor:

```
comparacionModelos<-as.data.frame.matrix(modelos)
comparacionModelos
```

```
##             Sin SMOTE Con SMOTE
## Baseline    0.8600796 0.5384615
## ?rbol poda  0.8805310 0.8464730
## C 4.5       0.9358407 0.9318317
## Na?ve Bayes 0.8716814 0.7646710
```