Tipología y ciclo de vida de los datos

Práctica 2: Limpieza y validación de los datos

Daniel Mato Regueira e Iago Veiras Lens

Junio de 2019

Índice

Detalles de la actividad

1. Descripción del dataset

1.1. Variables del dataset

1.2. Importancia y objetivo del análisis

2. Integración y selección de los datos

Cargamos los dos conjuntos de datos

```
data <- read.csv2(
  file = "https://archive.ics.uci.edu/ml/machine-learning-databases/horse-colic/horse-colic.data",
  header = F, sep = "")
data <- rbind(
  data,
  read.csv(
    file = "https://archive.ics.uci.edu/ml/machine-learning-databases/horse-colic/horse-colic.test",
    header = F, sep = ""))
```

Damos nombre a las columnas según el repositorio (enlace al documento con la descripción de variables)

```
colnames(data) <- c("surgery", "age", "Hospital_Number", "rectal_temperature", "pulse",
                    "respiratory_rate", "temperature_of_extremities", "peripheral_pulse",
                    "mucous_membranes", "capillary_refill_time", "pain", "peristalsis",
                    "abdominal_distension", "nasogastric_tube", "nasogastric_reflux",
                    "nasogastric_reflux_PH", "rectal_examination", "abdomen",
                    "packed_cell_volume","total_protein", "abdominocentesis_appearance",
                    "abdomcentesis_total_protein", "outcome", "surgical_lesion",
                    "type_lesion_1", "type_lesion_2", "type_lesion_3", "cp_data")
```

Convertimos los ? a NaN

```
data[data == '?'] <- NaN
```

Descartamos las variables inútiles

```
data <- data[-c(3, 6, 16, 28)]
```

Refactorizamos las variables categóricas

```
data$surgery <- factor(data$surgery, labels = c("yes", "no"), levels = c(1, 2))
data$age <- factor(data$age, labels = c("adult", "young"), levels = c(1, 9))
data$temperature_of_extremities <- factor(data$temperature_of_extremities,
                                          labels = c("normal", "warm", "cool", "cold"),
                                          levels = c(1, 2, 3, 4))
data$peripheral_pulse <- factor(data$peripheral_pulse,
                                labels = c("normal", "increased", "reduced", "absent"),
                                levels = c(1, 2, 3, 4))
data$mucous_membranes <- factor(data$mucous_membranes,
                                labels = c("normal pink", "bright pink", "pale pink",
                                           "pale cyanotic", "bright red",
                                           "dark cyanotic"),
                                levels = c(1, 2, 3, 4, 5, 6))
data$capillary_refill_time <- factor(data$capillary_refill_time,
```

```
                                    labels = c("< 3s", "> 3s"), levels = c(1, 2))
data$pain <- factor(data$pain,
                    labels = c("alert", "depressed", "intermittent mild pain",
                               "intermittent severe pain", "continuous severe pain"),
                    levels = c(1, 2, 3, 4, 5))
data$peristalsis <- factor(data$peristalsis,
                           labels = c("hypermotile", "normal", "hypomotile", "absent"),
                           levels = c(1, 2, 3, 4))
data$abdominal_distension <- factor(data$abdominal_distension,
                                    labels = c("none", "slight", "moderate", "severe"),
                                    levels = c(1, 2, 3, 4))
data$nasogastric_tube <- factor(data$nasogastric_tube,
                                labels = c("none", "slight", "significant"),
                                levels = c(1, 2, 3))
data$nasogastric_reflux <- factor(data$nasogastric_reflux,
                                  labels = c("none", "> 1l", "< 1l"),
                                  levels = c(1, 2, 3))
data$rectal_examination <- factor(data$rectal_examination,
                                  labels = c("normal", "increased", "decreased",
                                             "absent"),
                                  levels = c(1, 2, 3, 4))
data$abdomen <- factor(data$abdomen,
                       labels = c("normal", "other", "firm feces li", "distended si",
                                  "distended li"),
                       levels = c(1, 2, 3, 4, 5))
data$abdominocentesis_appearance <- factor(data$abdominocentesis_appearance,
                                           labels = c("clear", "cloudy",
                                                      "serosanguineous"),
                                           levels = c(1, 2, 3))
data$outcome <- factor(data$outcome,
                       labels = c("lived", "died", "euthanized"), levels = c(1, 2, 3))
data$surgical_lesion <- factor(data$surgical_lesion,
                               labels = c("yes", "no"), levels = c(1, 2))
```

Convertimos a continuas las variables numéricas

```
data$rectal_temperature <- as.numeric(levels(data$rectal_temperature))[
  data$rectal_temperature]
data$pulse <- as.numeric(levels(data$pulse))[data$pulse]
data$packed_cell_volume <- as.numeric(levels(data$packed_cell_volume))[
  data$packed_cell_volume]
data$total_protein <- as.numeric(levels(data$total_protein))[data$total_protein]
data$abdomcentesis_total_protein <-
  as.numeric(levels(data$abdomcentesis_total_protein))[data$abdomcentesis_total_protein]
```

Nos interesa el número de lesiones, no el tipo. Convertimos las 3 variables en una única

```
data$type_lesion_1[data$type_lesion_1 > 0] <- 1
data$type_lesion_2[data$type_lesion_2 > 0] <- 1
data$type_lesion_3[data$type_lesion_3 > 0] <- 1
data$num_lesion <- data$type_lesion_1 + data$type_lesion_2 + data$type_lesion_3
data <- data[-c(22, 23, 24)]
```

Resultado de los datos integrados (no limpios)

```
summary(data)
```

```
##  surgery       age      rectal_temperature     pulse
##  yes :214   adult:340   Min.   :35.40      Min.   : 30.00
##  no  :152   young: 28   1st Qu.:37.80      1st Qu.: 48.00
##  NA's:  2               Median :38.10      Median : 60.00
##                         Mean   :38.13      Mean   : 70.76
##                         3rd Qu.:38.50      3rd Qu.: 88.00
##                         Max.   :40.80      Max.   :184.00
##                         NA's   :69         NA's   :26
##  temperature_of_extremities  peripheral_pulse      mucous_membranes
##  normal: 95                  normal   :151     normal pink  :98
##  warm  : 39                  increased:  6     bright pink  :38
##  cool  :135                  reduced  :116     pale pink    :81
##  cold  : 34                  absent   : 12     pale cyanotic:50
##  NA's  : 65                  NA's     : 83     bright red   :28
##                                                dark cyanotic:25
##                                                NA's         :48
##  capillary_refill_time                        pain         peristalsis
##  < 3s:232              alert                    :49   hypermotile: 49
##  > 3s: 96              depressed                :77   normal     : 22
##  NA's: 40              intermittent mild pain   :82   hypomotile :154
##                        intermittent severe pain:47   absent     : 91
##                        continuous severe pain   :50   NA's       : 52
##                        NA's                     :63
##
##  abdominal_distension    nasogastric_tube nasogastric_reflux
##  none    :101         none       : 89     none:141
##  slight  : 75         slight     :121     > 1l: 45
##  moderate: 85         significant: 27     < 1l: 49
##  severe  : 42         NA's       :131     NA's:133
##  NA's    : 65
##
##
##  rectal_examination          abdomen    packed_cell_volume total_protein
##  normal   : 68      normal       : 31   Min.   : 4.00      Min.   : 3.30
##  increased: 14      other        : 24   1st Qu.:37.25      1st Qu.: 6.50
##  decreased: 61      firm feces li: 19   Median :44.00      Median : 7.50
##  absent   : 97      distended si : 55   Mean   :45.66      Mean   :24.77
##  NA's     :128      distended li : 96   3rd Qu.:52.00      3rd Qu.:58.00
##                     NA's         :143   Max.   :75.00      Max.   :89.00
##                                         NA's   :37         NA's   :43
##   abdominocentesis_appearance abdomcentesis_total_protein       outcome
##  clear          : 52          Min.   : 0.100              lived     :225
##  cloudy         : 62          1st Qu.: 2.000              died      : 89
##  serosanguineous: 60          Median : 2.100              euthanized: 52
##  NA's           :194          Mean   : 2.948              NA's      :  2
##                               3rd Qu.: 3.900
##                               Max.   :10.100
##                               NA's   :235
##  surgical_lesion   num_lesion
##  yes:232         Min.   :0.0000
```

```
##  no :136         1st Qu.:1.0000
##                  Median :1.0000
##                  Mean   :0.8478
##                  3rd Qu.:1.0000
##                  Max.   :3.0000
##
```

3. Limpieza de los datos

3.1. Elementos vacíos

Eliminamos la filas que tenga más de un 50 % de variables faltantes

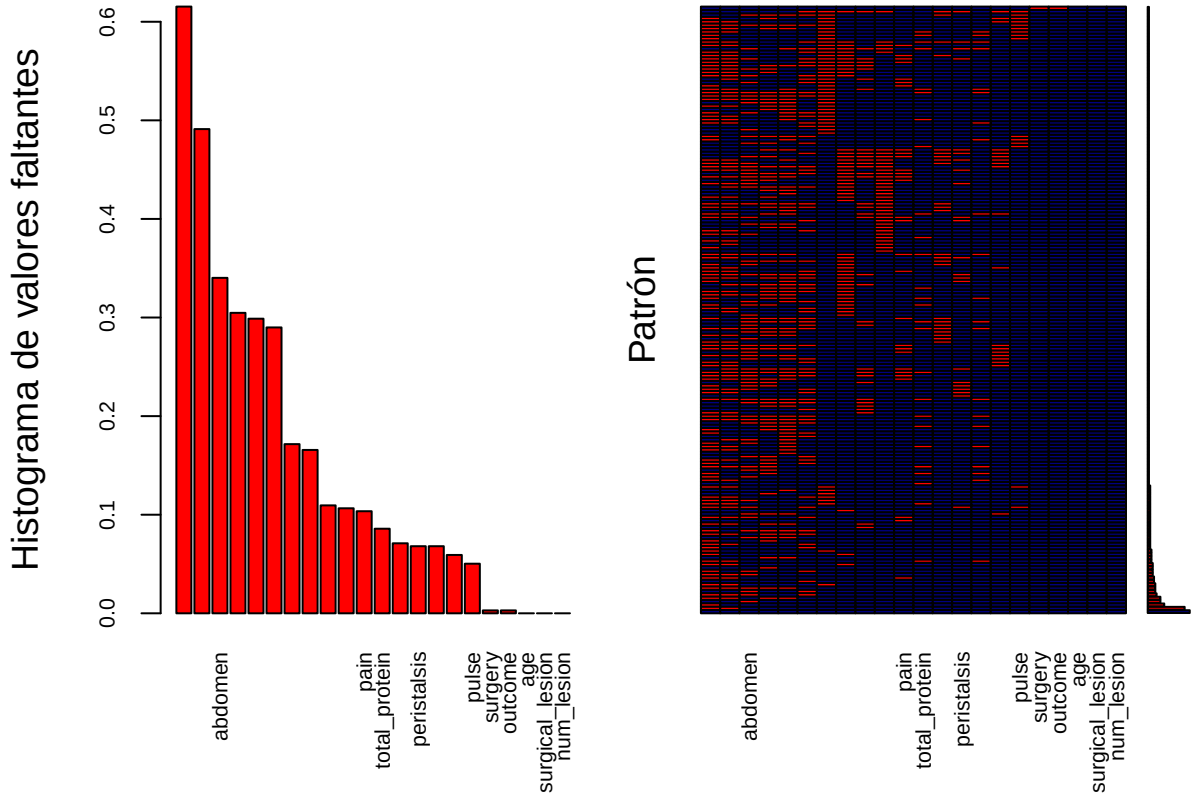
```
filas_nan <- apply(data, 1, function(y) sum(is.na(y))) < dim(data)[2]/2
data <- data[filas_nan,]
```

Cargamos la librería "VIM" para estudiar los patrones de datos faltantes

```
if (!require("VIM")) install.packages("VIM")
library(VIM)
```

Representamos los patrones de datos faltantes y el conteo de variables con mayor número de datos faltantes

```
aggr_plot <- aggr(data, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE,
                  labels=names(data), cex.axis=.7,
                  gap=3, ylab=c("Histograma de valores faltantes", "Patrón"))
```



```
##
##  Variables sorted by number of missings:
##                      Variable      Count
##  abdomcentesis_total_protein 0.61538462
##  abdominocentesis_appearance 0.49112426
##                      abdomen 0.34023669
##           nasogastric_reflux 0.30473373
##             nasogastric_tube 0.29881657
##           rectal_examination 0.28994083
##           rectal_temperature 0.17159763
##             peripheral_pulse 0.16568047
##         abdominal_distension 0.10946746
##   temperature_of_extremities 0.10650888
##                         pain 0.10355030
##                total_protein 0.08579882
##             mucous_membranes 0.07100592
##                   peristalsis 0.06804734
##           packed_cell_volume 0.06804734
##        capillary_refill_time 0.05917160
##                        pulse 0.05029586
##                      surgery 0.00295858
##                      outcome 0.00295858
##                         age 0.00000000
##              surgical_lesion 0.00000000
##                   num_lesion 0.00000000
```

Cargamos la librería "MICE" para imputar datos faltantes

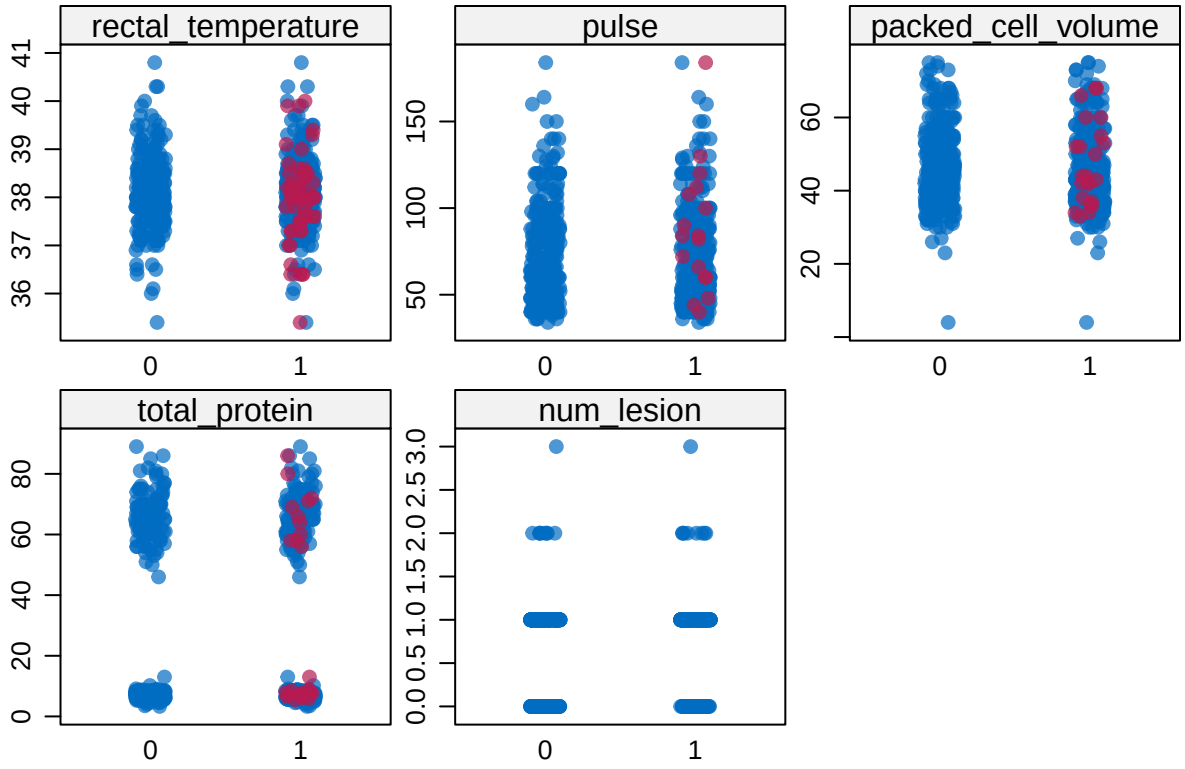
```
if (!require("mice")) install.packages("mice")
library(mice)
```

Eliminamos las variables con más de un 50 % de valores faltantes

```
data <- data[-c(18, 19)]
```

Generamos varias capas de imputación para los datos faltantes y los representamos

```
#temp_data <- mice(data, printFlag = F)
temp_data <- mice(data, printFlag = F, m= 1, maxiter=1)
stripplot(temp_data, pch=20, cex = 1.2)
```



Imputamos los datos en el conjunto original

```
data <- complete(temp_data)
```

3.2. Valores extremos

Estudiamos los outliers de "rectal_temperature". Los valores son extremos pero plausibles

```
boxplot.stats(data$rectal_temperature)$out
```

```
##  [1] 39.9 36.4 35.4 36.4 40.3 39.9 39.7 36.4 36.4 40.3 36.5 39.9 35.4 36.1
## [15] 40.8 36.4 40.0 36.5 39.7 36.0 40.0
```

Estudiamos los outliers de "pulse". Eliminamos los valores extremos, ya que no son valores alcanzables por caballos

```
boxplot.stats(data$pulse)$out
```

```
## [1] 164 160 184 184 150 150
```

```
out_pulse <- boxplot.stats(data$pulse)$out
data <- data[-which(data$pulse %in% out_pulse), ]
```

Estudiamos los outliers de “packed_cell_volume”. Eliminamos los valores extremos, ya que no son valores plausibles

```
boxplot.stats(data$packed_cell_volume)$out
```

```
## [1] 75 75  4 74
```

```
out_cells <- boxplot.stats(data$packed_cell_volume)$out
data <- data[-which(data$packed_cell_volume %in% out_cells), ]
```

Estudiamos los outliers de “total_protein”. No hay ninguno

```
boxplot.stats(data$total_protein)$out
```

```
## numeric(0)
```

Estudiamos los outliers de “num_lesion”. No los eliminamos, ya que son valores calculados anteriormente y estos eran plausibles

```
boxplot.stats(data$num_lesion)$out
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 2 0 2 2 0 0 0 0 0 0 0 0 0 0 0
## [36] 0 0 2 0 0 0 0 0 0 2 0 0 3 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 2
```

Generamos un fichero con los datos filtrados y tratados.

```
write.csv(data, file = "clean_data_horse_colic.csv", row.names = F)
```

4. Análisis de los datos

4.1. Selección de grupos a analizar

4.2. Comprobación de normalidad y homocedasticidad

Aplicamos el test de Shapiro-Wilk a las cuatro variables numéricas

```
shapiro.test(data$rectal_temperature)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$rectal_temperature
## W = 0.96797, p-value = 1.213e-06
```

```
shapiro.test(data$pulse)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$pulse
## W = 0.91651, p-value = 1.557e-12
```

```
shapiro.test(data$packed_cell_volume)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$packed_cell_volume
## W = 0.95931, p-value = 6.517e-08
```

```
shapiro.test(data$total_protein)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$total_protein
## W = 0.67577, p-value < 2.2e-16
```

Cargamos la librería “car” para estudiar la homocedasticidad de los dos grupos de datos generados

```
if (!require("car")) install.packages("car")
library(car)
```

Realizamos el test de Levene para comprobar la homocedasticidad

```
leveneTest(total_protein ~ surgical_lesion, data)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.8167 0.3668
##       326
```

4.3. Aplicación de pruebas estadísticas

4.3.1. Estudio de correlación

Cargamos la librería "dummies" para convertir en numéricas las variables categóricas

```
if (!require("dummies")) install.packages("dummies")
library(dummies)
```

Extendemos la variable "abdominal_distension"

```
data <- dummy.data.frame(data, names = "abdominal_distension",
                         dummy.classes="ALL", sep = "_")
```

Asumimos que los caballos a los que se les practica la eutanasias mueren.

```
levels(data$outcome) <- c(0, 1, 1)
data$outcome <- as.numeric(data$outcome) - 1
```

Seleccionamos únicamente las variables numéricas

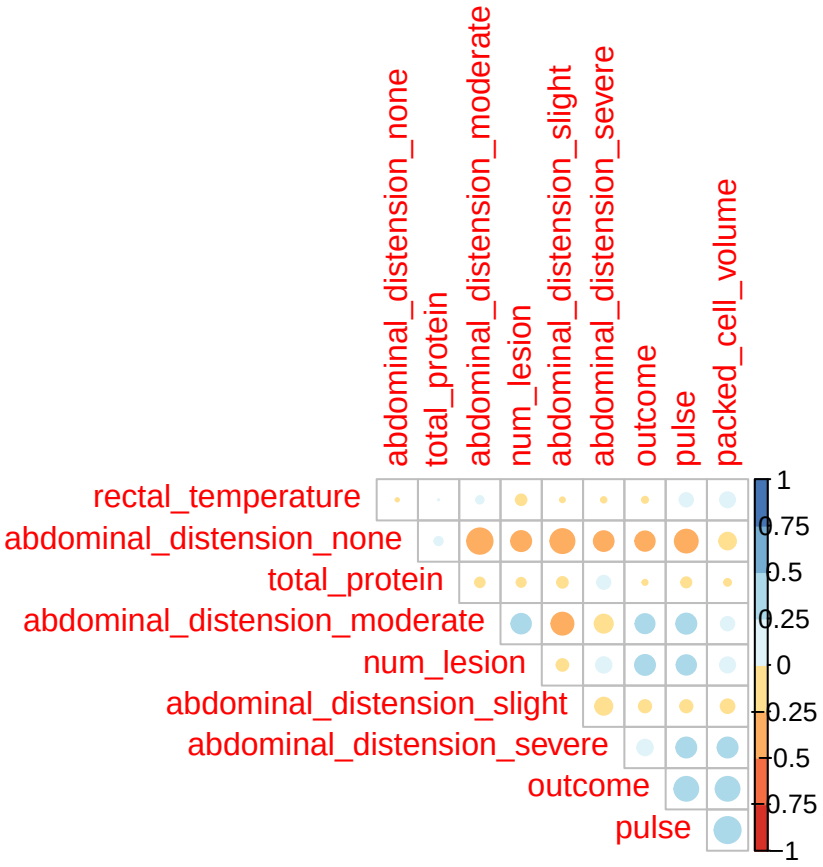
```
nums <- unlist(lapply(data, is.numeric))
data_nums <- data[ , nums]
```

Cargamos las librerías "corrplot" y "RColorBrewer" para representar la correlación entre las variables continuas

```
if (!require("corrplot")) install.packages("corrplot")
library(corrplot)
if (!require("RColorBrewer")) install.packages("RColorBrewer")
library(RColorBrewer)
```

Representamos la correlación entre las diferentes variables continuas, haciendo especial hincapié en los resultados para la variable "outcome"

```
corrplot(cor(data_nums), type = "upper", order = "hclust",
         col = brewer.pal(n = 8, name = "RdYlBu"), diag = F)
```



4.3.2. Prueba de contraste de hipótesis

Creamos dos subconjuntos de datos de "total_protein" en función de si fueron operados o no

```
data_oper <- data[data$surgery == "yes", ]$total_protein
data_noper <- data[data$surgery == "no", ]$total_protein
```

Realizamos un contraste de hipótesis con hipótesis nula que las medias de los dos subconjuntos sn iguales. No se puede rechazar

```
t.test(data_noper, data_oper)
```

```
##
##  Welch Two Sample t-test
##
## data:  data_noper and data_oper
## t = -2.0594, df = 308.31, p-value = 0.04029
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -12.4643801  -0.2839655
## sample estimates:
## mean of x mean of y
##  22.41176  28.78594
```

4.3.3. Modelo de regresión logística

Creamos dos conjuntos a partir de los datos originales, uno para estimar el modelo y otro para testearlo

```
train_ind <- sample(seq_len(nrow(data)), size = 0.8*dim(data)[1])
train <- data[train_ind, ]
test <- data[-train_ind, ]
```

Estimamos el modelo de regresión logística y representamos sus características más reseñables

```
out_model <- glm(outcome ~ pulse + packed_cell_volume + num_lesion +
                   abdominal_distension_moderate,
                 data = train, family = "binomial")
summary(out_model)
```

```
##
## Call:
## glm(formula = outcome ~ pulse + packed_cell_volume + num_lesion +
##     abdominal_distension_moderate, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1493  -0.8218  -0.3259   0.8653   2.2506
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -6.333254   0.918160  -6.898 5.28e-12 ***
## pulse                          0.020803   0.007131   2.917 0.003531 **
## packed_cell_volume             0.053235   0.017251   3.086 0.002030 **
## num_lesion                     1.887981   0.550541   3.429 0.000605 ***
## abdominal_distension_moderate  0.693821   0.330624   2.099 0.035859 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 349.35  on 261  degrees of freedom
## Residual deviance: 265.59  on 257  degrees of freedom
## AIC: 275.59
##
## Number of Fisher Scoring iterations: 5
```

Cargamos la librería "caret" para calcular la matriz de confusión de los resultados

```
if (!require("caret")) install.packages("caret")
library(caret)
```

Calculamos la matriz de confusión sobre los datos de test

```
test_pred <- predict.glm(out_model, test, type="response")
confusionMatrix(factor(round(test_pred)), factor(test$outcome))
```

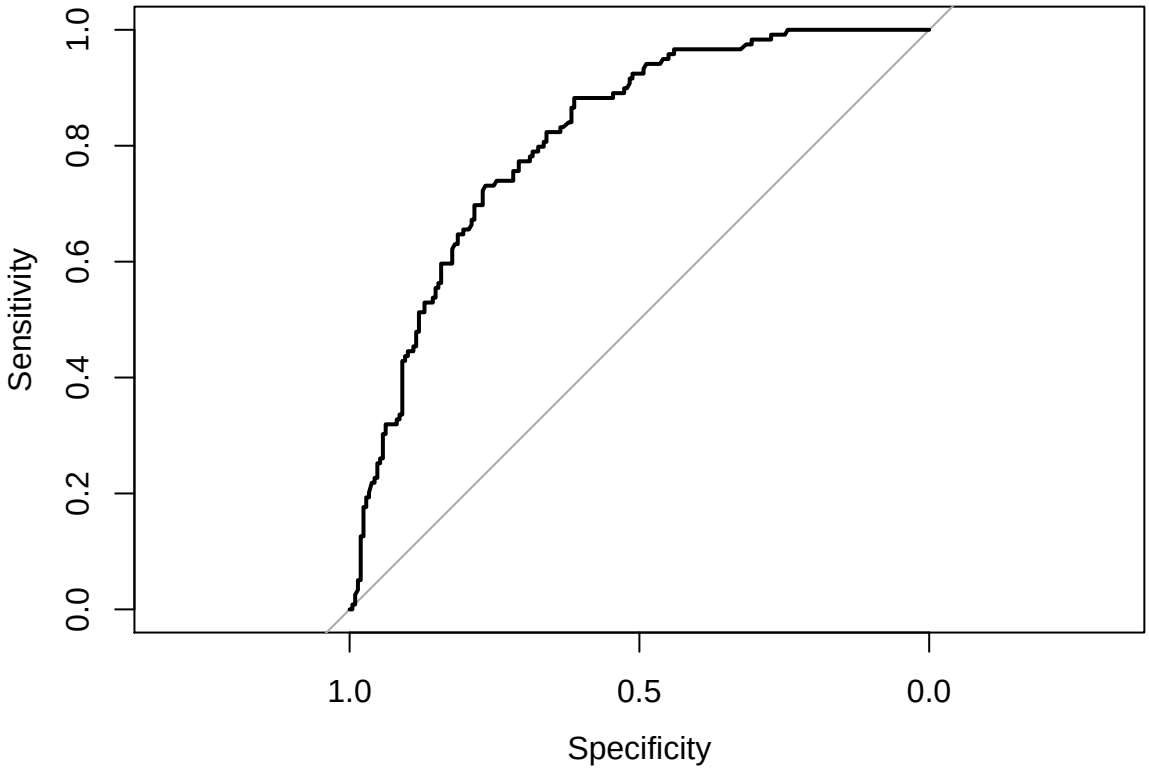
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 36  3
##          1 12 15
##
##                Accuracy : 0.7727
##                  95% CI : (0.653, 0.8669)
##     No Information Rate : 0.7273
##     P-Value [Acc > NIR] : 0.24844
##
##                   Kappa : 0.5045
##  Mcnemar's Test P-Value : 0.03887
##
##             Sensitivity : 0.7500
##             Specificity : 0.8333
##          Pos Pred Value : 0.9231
##          Neg Pred Value : 0.5556
##              Prevalence : 0.7273
##          Detection Rate : 0.5455
##    Detection Prevalence : 0.5909
##       Balanced Accuracy : 0.7917
##
##        'Positive' Class : 0
##
```

Cargamos la librería "pROC" para dibjuar la curva ROC del modelo estimado

```
if (!require("pROC")) install.packages("pROC")
library(pROC)
```

Representamos la curva ROC del modelo estimado

```
curva_roc <- roc(outcome ~ predict.glm(out_model, data, type = "response"), data)
plot(curva_roc)
```


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity

5. Conclusiones