Redes Neuronales

Maria Jose Bustamante - Nicolas Jadan

Importamos las librerias a usar

```
library(caret)
```

```
Loading required package: ggplot2
```

```
Loading required package: lattice
```

```
library(neuralnet)
library(ggplot2)
library(lattice)
```

Se lee un conjunto de datos con la función **`read.table()`**.

```
data <- read.table(file = "wdbc.data", header = FALSE, sep = ",")
head(data)
```

```
        V1 V2    V3    V4     V5     V6      V7      V8     V9     V10     V11
1   842302  M 17.99 10.38 122.80 1001.0 0.11840 0.27760 0.3001 0.14710 0.2419
2   842517  M 20.57 17.77 132.90 1326.0 0.08474 0.07864 0.0869 0.07017 0.1812
3 84300903  M 19.69 21.25 130.00 1203.0 0.10960 0.15990 0.1974 0.12790 0.2069
4 84348301  M 11.42 20.38  77.58  386.1 0.14250 0.28390 0.2414 0.10520 0.2597
5 84358402  M 20.29 14.34 135.10 1297.0 0.10030 0.13280 0.1980 0.10430 0.1809
6   843786  M 12.45 15.70  82.57  477.1 0.12780 0.17000 0.1578 0.08089 0.2087
      V12    V13    V14   V15    V16      V17     V18     V19     V20     V21
1 0.07871 1.0950 0.9053 8.589 153.40 0.006399 0.04904 0.05373 0.01587 0.03003
2 0.05667 0.5435 0.7339 3.398  74.08 0.005225 0.01308 0.01860 0.01340 0.01389
3 0.05999 0.7456 0.7869 4.585  94.03 0.006150 0.04006 0.03832 0.02058 0.02250
```

```
4 0.09744 0.4956 1.1560 3.445  27.23 0.009110 0.07458 0.05661 0.01867 0.05963
5 0.05883 0.7572 0.7813 5.438  94.44 0.011490 0.02461 0.05688 0.01885 0.01756
6 0.07613 0.3345 0.8902 2.217  27.19 0.007510 0.03345 0.03672 0.01137 0.02165
       V22   V23   V24    V25    V26    V27    V28    V29    V30    V31     V32
1 0.006193 25.38 17.33 184.60 2019.0 0.1622 0.6656 0.7119 0.2654 0.4601 0.11890
2 0.003532 24.99 23.41 158.80 1956.0 0.1238 0.1866 0.2416 0.1860 0.2750 0.08902
3 0.004571 23.57 25.53 152.50 1709.0 0.1444 0.4245 0.4504 0.2430 0.3613 0.08758
4 0.009208 14.91 26.50  98.87  567.7 0.2098 0.8663 0.6869 0.2575 0.6638 0.17300
5 0.005115 22.54 16.67 152.20 1575.0 0.1374 0.2050 0.4000 0.1625 0.2364 0.07678
6 0.005082 15.47 23.75 103.40  741.6 0.1791 0.5249 0.5355 0.1741 0.3985 0.12440
```

"V2" se convierte en un factor. Se usa **`complete.cases()`** para verificar la cantidad de casos completos en ese conjunto de datos.

```r
data$V2 <- as.factor(data$V2)
```

1. Descripción de los mismos numérica y gráficamente

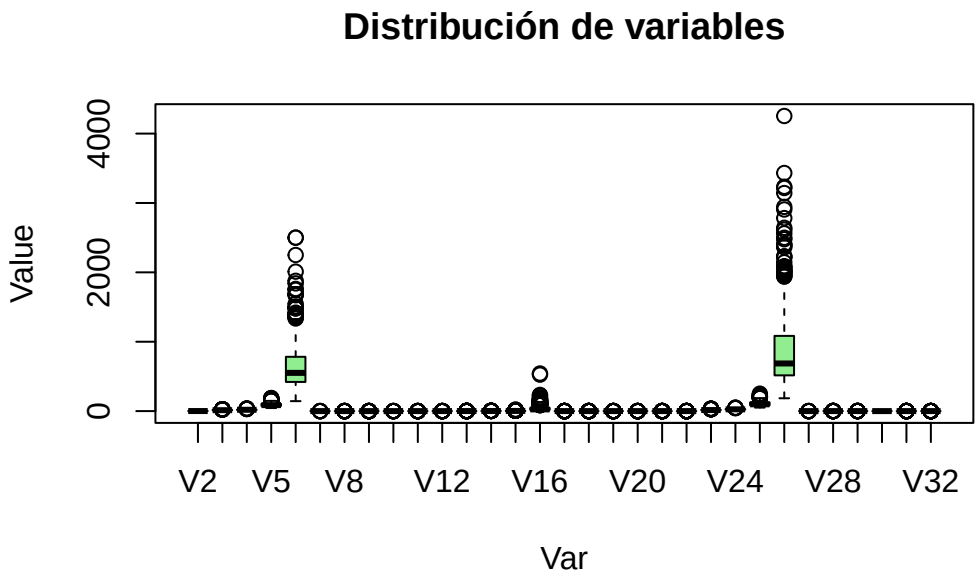
El objetivo es generar un resumen para cada columna o variable en el conjunto de datos. Este resumen incluirá estadísticas descriptivas importantes, como el valor mínimo, el primer cuartil, la mediana, el tercer cuartil y el valor máximo para las variables numéricas presentes en los datos. De esta manera, se podrá obtener una visión general de la distribución y el rango de valores de cada variable en el conjunto de datos.

```r
summary(data)
```

```
      V1            V2            V3               V4               V5        
 Min.   :     8670   B:357   Min.   : 6.981   Min.   : 9.71   Min.   : 43.79  
 1st Qu.:   869218   M:212   1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17  
 Median :   906024           Median :13.370   Median :18.84   Median : 86.24  
 Mean   : 30371831           Mean   :14.127   Mean   :19.29   Mean   : 91.97  
 3rd Qu.:  8813129           3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10  
 Max.   :911320502           Max.   :28.110   Max.   :39.28   Max.   :188.50  
      V6               V7                V8                V9          
 Min.   : 143.5   Min.   :0.05263   Min.   :0.01938   Min.   :0.00000  
 1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956  
 Median : 551.1   Median :0.09587   Median :0.09263   Median :0.06154  
 Mean   : 654.9   Mean   :0.09636   Mean   :0.10434   Mean   :0.08880  
 3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070  
```

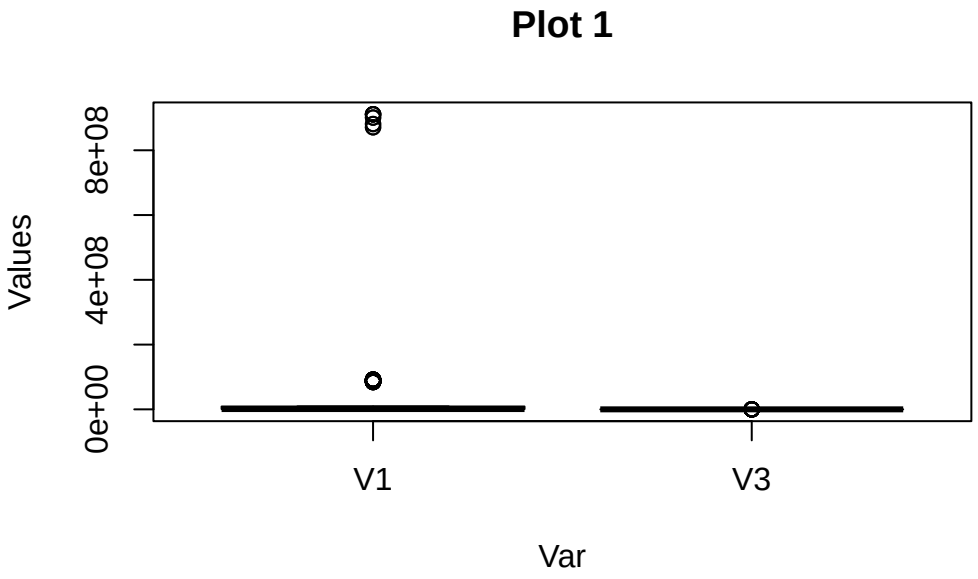
```
Max.   :2501.0   Max.   :0.16340   Max.   :0.34540   Max.   :0.42680
     V10               V11               V12               V13
Min.   :0.00000   Min.   :0.1060   Min.   :0.04996   Min.   :0.1115
1st Qu.:0.02031   1st Qu.:0.1619   1st Qu.:0.05770   1st Qu.:0.2324
Median :0.03350   Median :0.1792   Median :0.06154   Median :0.3242
Mean   :0.04892   Mean   :0.1812   Mean   :0.06280   Mean   :0.4052
3rd Qu.:0.07400   3rd Qu.:0.1957   3rd Qu.:0.06612   3rd Qu.:0.4789
Max.   :0.20120   Max.   :0.3040   Max.   :0.09744   Max.   :2.8730
     V14               V15               V16               V17
Min.   :0.3602   Min.   : 0.757   Min.   :  6.802   Min.   :0.001713
1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169
Median :1.1080   Median : 2.287   Median : 24.530   Median :0.006380
Mean   :1.2169   Mean   : 2.866   Mean   : 40.337   Mean   :0.007041
3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146
Max.   :4.8850   Max.   :21.980   Max.   :542.200   Max.   :0.031130
     V18               V19               V20               V21
Min.   :0.002252   Min.   :0.00000   Min.   :0.000000   Min.   :0.007882
1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160
Median :0.020450   Median :0.02589   Median :0.010930   Median :0.018730
Mean   :0.025478   Mean   :0.03189   Mean   :0.011796   Mean   :0.020542
3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480
Max.   :0.135400   Max.   :0.39600   Max.   :0.052790   Max.   :0.078950
     V22               V23               V24               V25
Min.   :0.0008948   Min.   : 7.93   Min.   :12.02   Min.   : 50.41
1st Qu.:0.0022480   1st Qu.:13.01   1st Qu.:21.08   1st Qu.: 84.11
Median :0.0031870   Median :14.97   Median :25.41   Median : 97.66
Mean   :0.0037949   Mean   :16.27   Mean   :25.68   Mean   :107.26
3rd Qu.:0.0045580   3rd Qu.:18.79   3rd Qu.:29.72   3rd Qu.:125.40
Max.   :0.0298400   Max.   :36.04   Max.   :49.54   Max.   :251.20
     V26               V27               V28               V29
Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000
1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145
Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267
Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722
3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829
Max.   :4254.0   Max.   :0.22260   Max.   :1.05800   Max.   :1.2520
     V30               V31               V32
Min.   :0.00000   Min.   :0.1565   Min.   :0.05504
1st Qu.:0.06493   1st Qu.:0.2504   1st Qu.:0.07146
Median :0.09993   Median :0.2822   Median :0.08004
Mean   :0.11461   Mean   :0.2901   Mean   :0.08395
3rd Qu.:0.16140   3rd Qu.:0.3179   3rd Qu.:0.09208
Max.   :0.29100   Max.   :0.6638   Max.   :0.20750
```

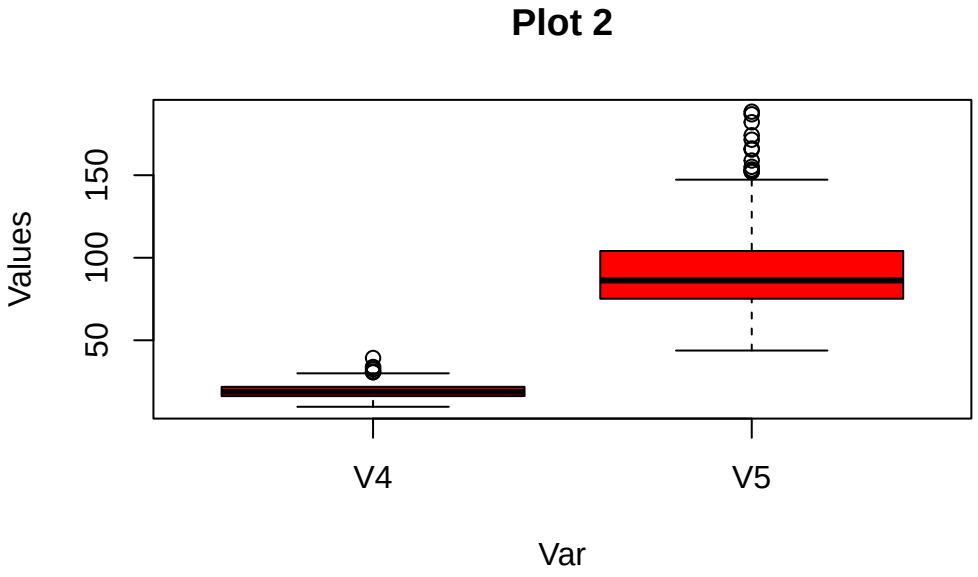
```r
boxplot(data[, -1], col = "lightgreen", main = "Distribución de variables", xlab = "Var",
```

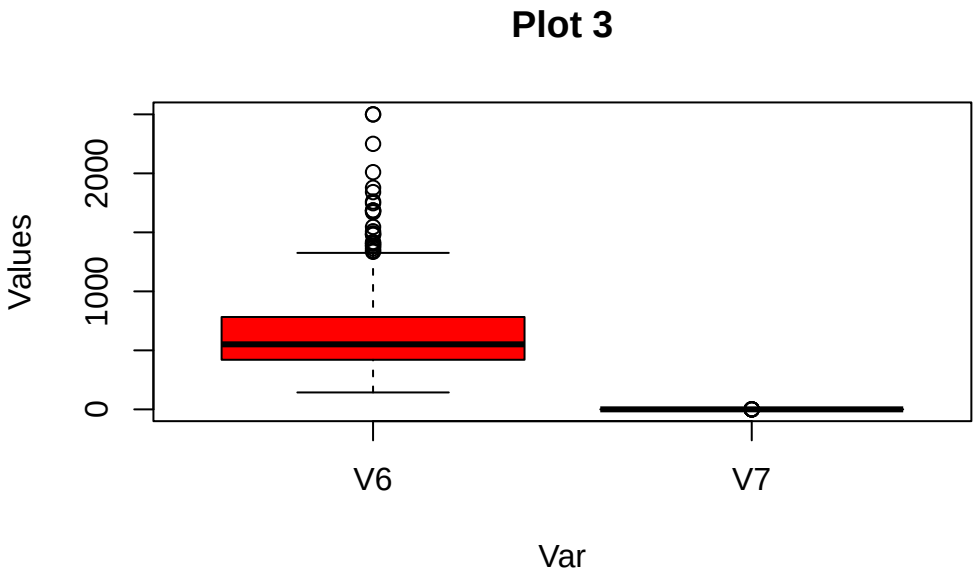


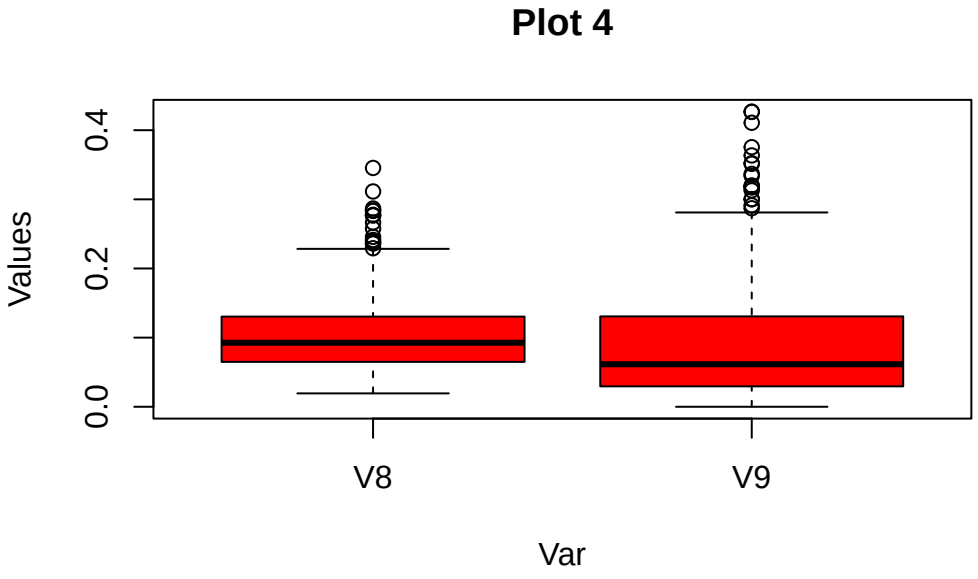
Este código generaría una ventana gráfica , mostrando boxplots para las variables del conjunto de datos Cada boxplot mostraría la distribución de los valores de las variables correspondientes.

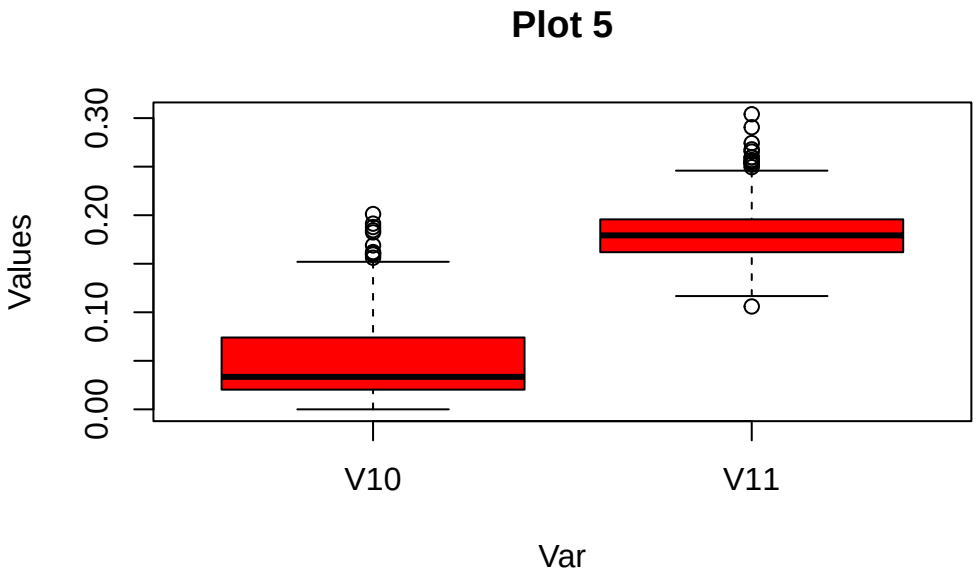
```r
variables <- names(data)[-2]
grupos <- split(variables, ceiling(seq_along(variables) / 2))

for (i in seq_along(grupos)) {
  par(mfrow = c(1, 1))  # Restablece la ventana gráfica a 1 fila y 1 columna
  boxplot(data[, grupos[[i]]], col = "red", main = paste("Plot", i),
          xlab = "Var", ylab = "Values")
}
```

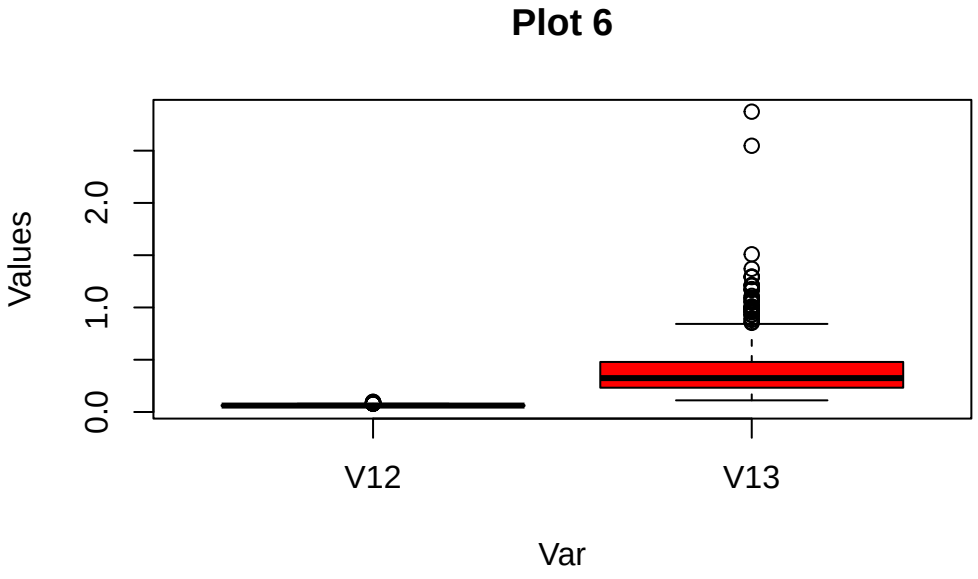

Plot 1
Values
0e+00
4e+08
8e+08
V1
V3
Var

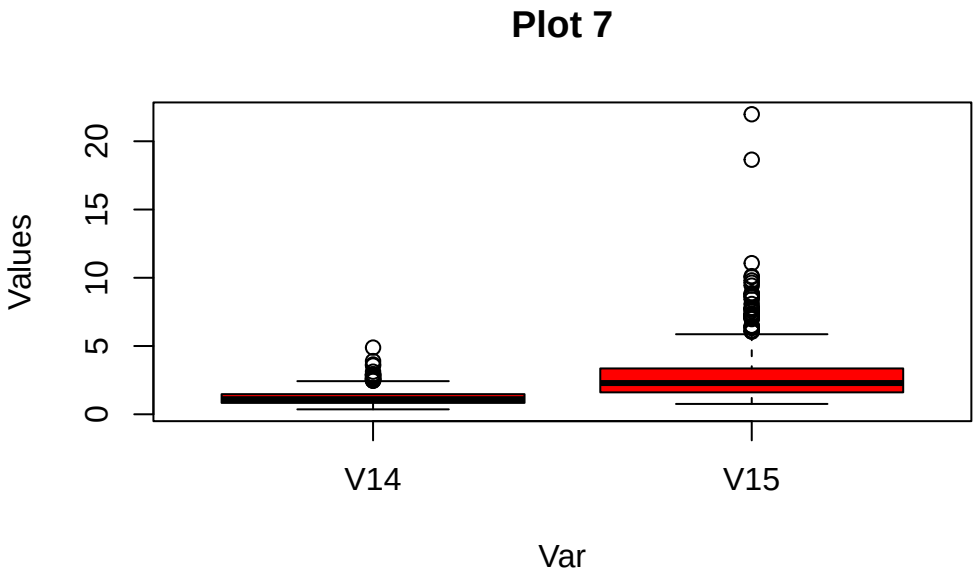

Plot 2
Values
50
100
150
V4
V5
Var

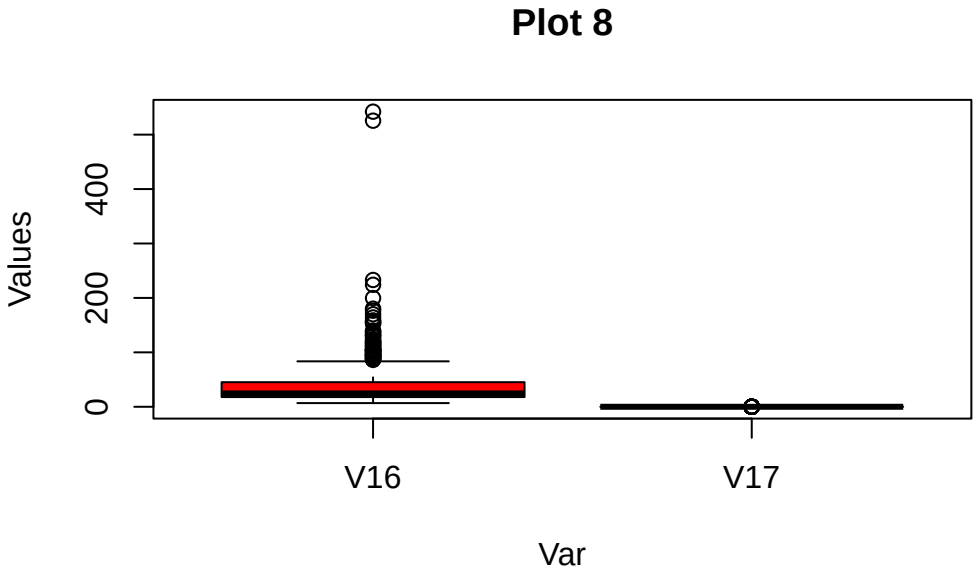

Plot 3
Values
0
1000
2000
V6
V7
Var

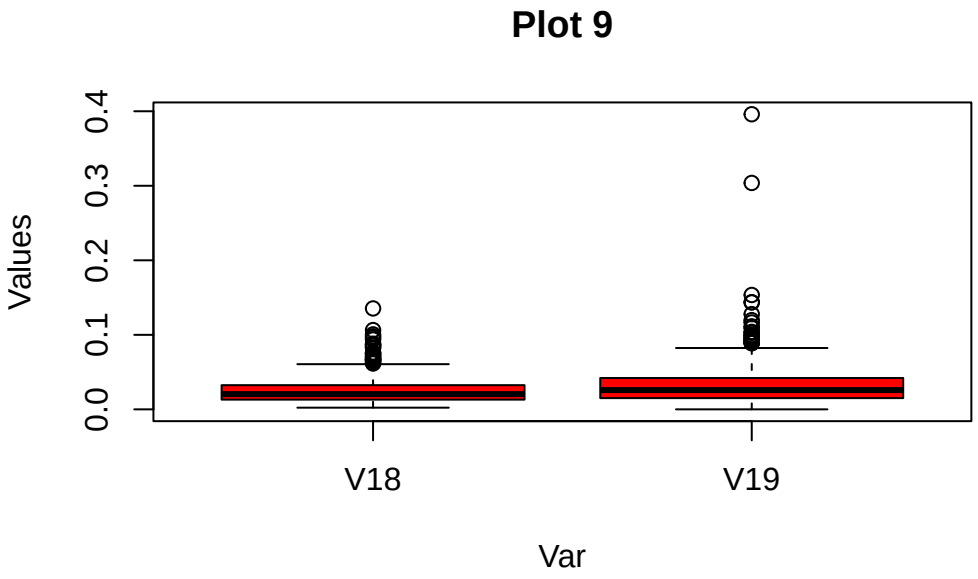

Plot 4
Values
0.0
0.2
0.4
V8
V9
Var

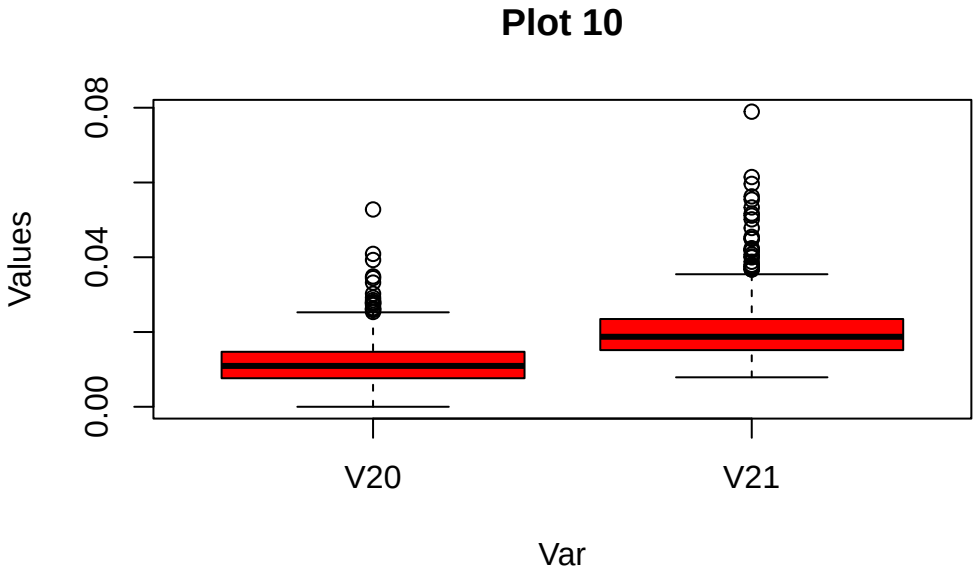

Plot 5
Values
0.00
0.10
0.20
0.30
V10
V11
Var

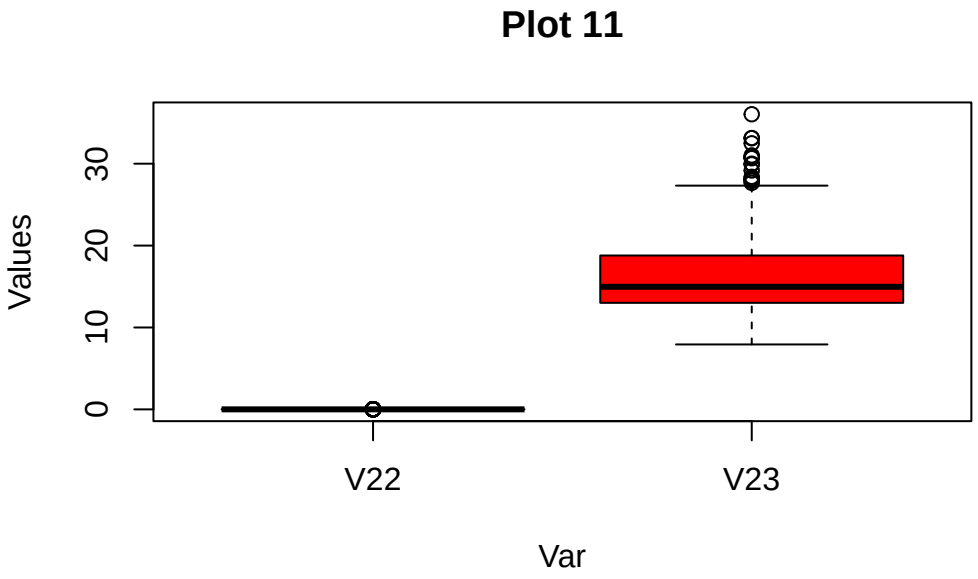

Plot 6
Values
0.0
1.0
2.0
V12
V13
Var

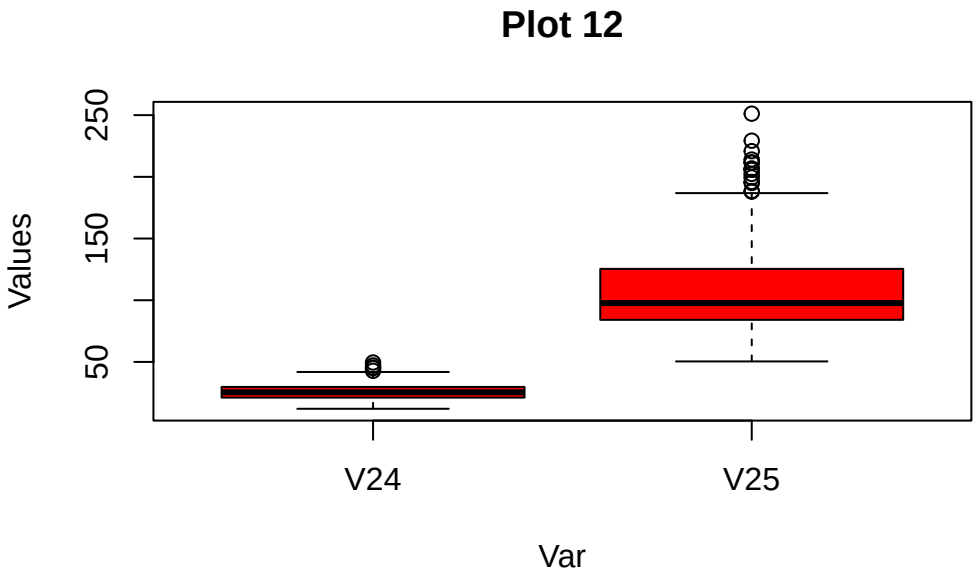

Plot 7
Values
0
5
10
15
20
V14
V15
Var

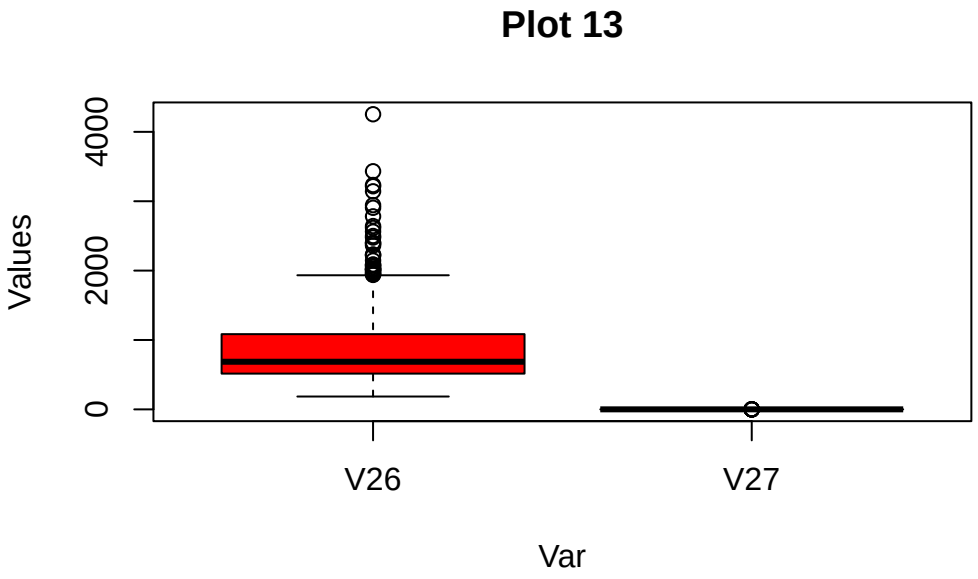

Plot 8
Values
0
200
400
V16
V17
Var

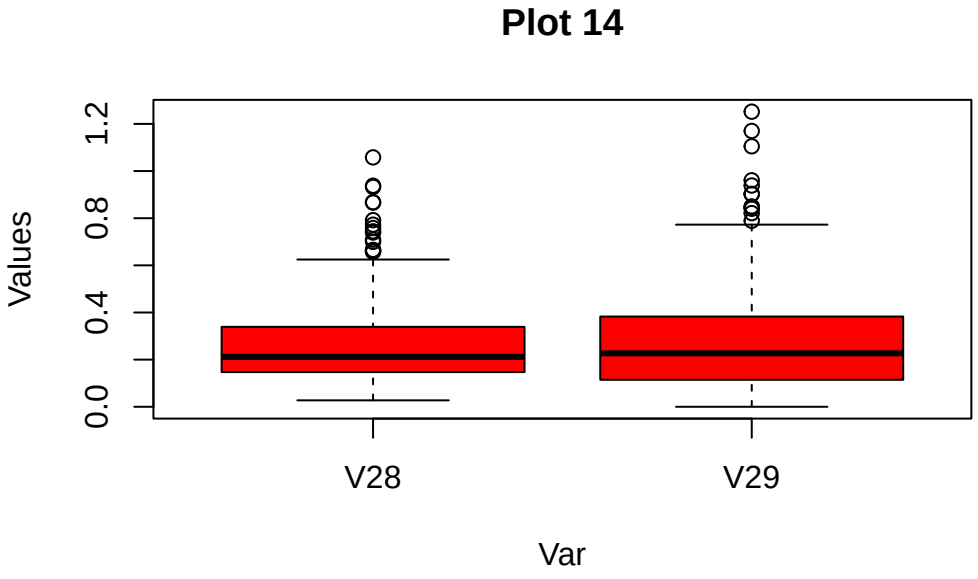

Plot 9
Values
0.0
0.1
0.2
0.3
0.4
V18
V19
Var

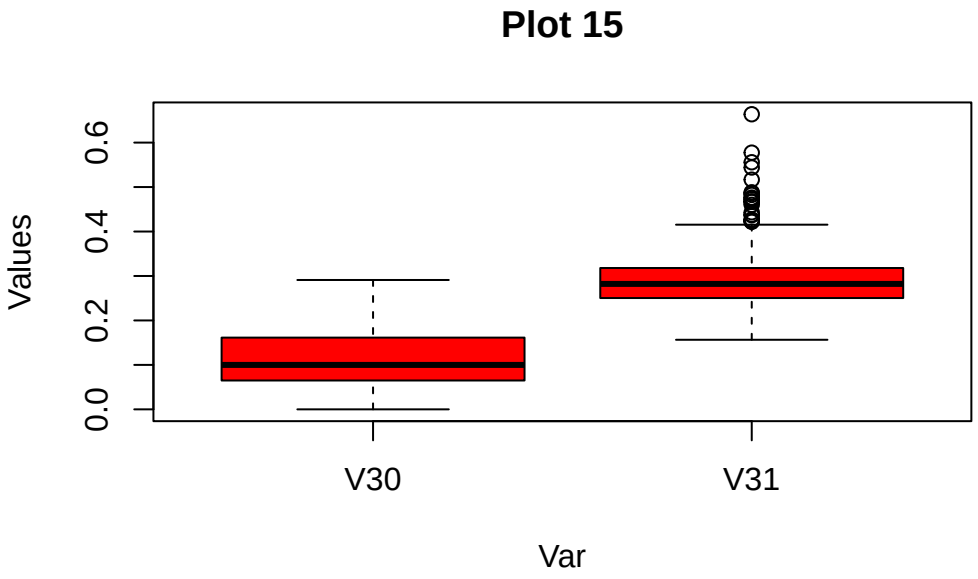

Plot 10
Values
0.00
0.04
0.08
V20
V21
Var

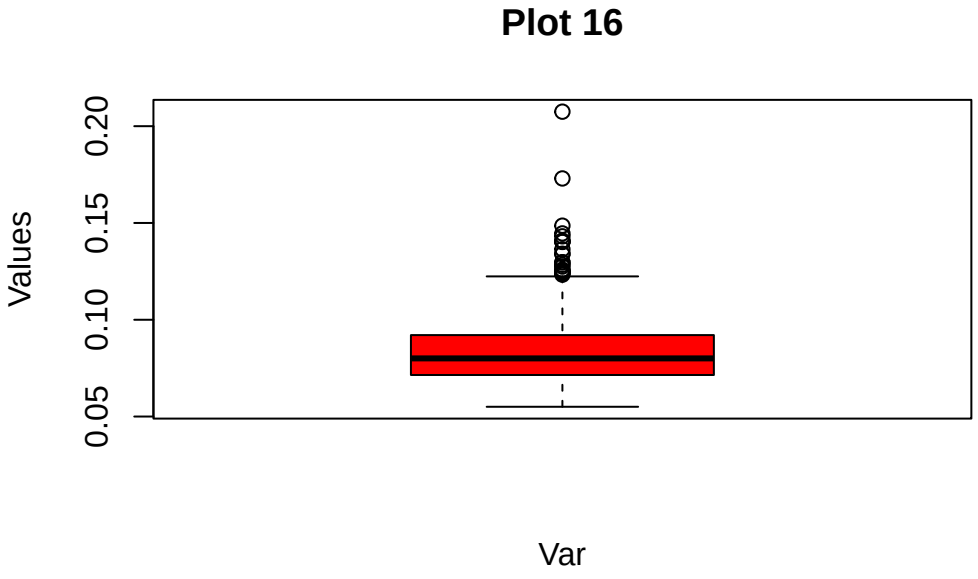

Plot 11
Values
0
10
20
30
V22
V23
Var


Plot 12
Values
50
150
250
V24
V25
Var


Plot 13
Values
0
2000
4000
V26
V27
Var


Plot 14
Values
0.0
0.4
0.8
1.2
V28
V29
Var





Estos gráficos ilustran la distribución de los valores de las variables pertenecientes al primer grupo. La información sobre los cuartiles y la mediana se representa mediante los elementos

de la caja en el gráfico, mientras que los valores mínimo y máximo se indican mediante los "bigotes". Además, es posible identificar la existencia de valores atípicos o extremos a través de los puntos individuales que se encuentran fuera de los "bigotes". Estos gráficos proporcionan una representación visual completa de la distribución de los valores de las variables y permiten observar patrones y posibles anomalías en los datos.

5. Realizar un modelo preliminar de una capa sobre la clasificacion begnigno o maligno

La función **normalize** toma un vector y lo normaliza en el rango de 0 a 1, mientras que el código proporcionado aplica esta función a todas las columnas de un data frame, excepto a una columna específica, y guarda el resultado en un nuevo data frame llamado **data_norm**.

```
normalize <- function(x) {
return((x - min(x)) / (max(x) - min(x)))
}
data_norm <- as.data.frame(lapply(data[,-2],normalize))
```

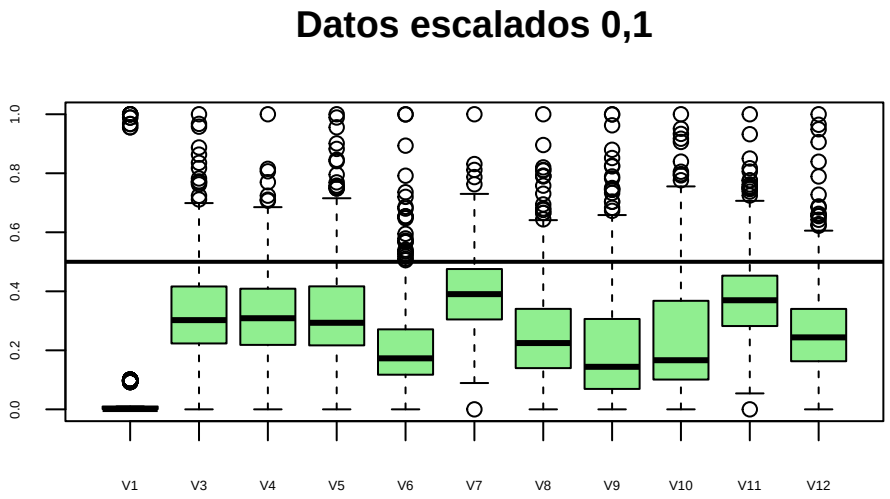
Creación de variables binarias.

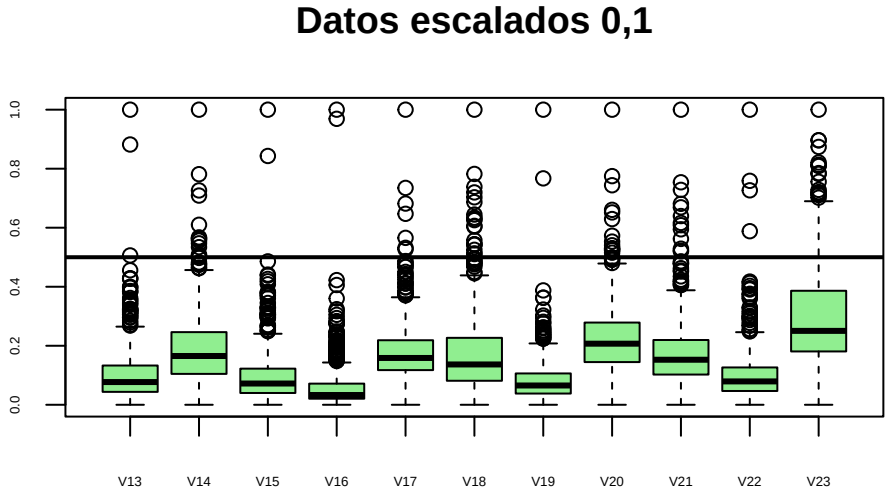
Se asigna valores booleanos a dos columnas nuevas, "M" y "B".

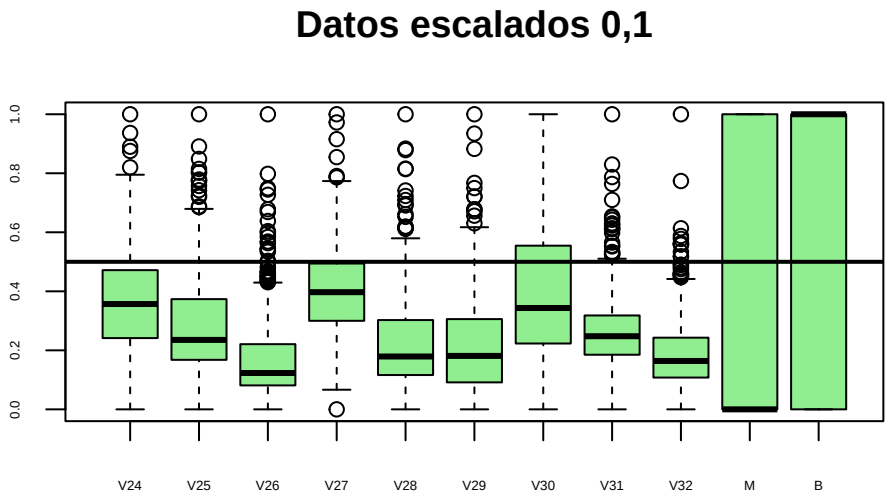
```
data_norm$M <- ifelse(data$V2 == "M", TRUE, FALSE)
data_norm$B <- ifelse(data$V2 == "B", TRUE, FALSE)
```

Interpretacion

```
par(mfrow = c(1, 1))
for (i in 1:3) {
  col_start <- (i - 1) * 11 + 1
  col_end <- i * 11

  boxplot(data_norm[, col_start:col_end], main = 'Datos escalados 0,1', col = 'lightgreen'
  abline(h = 0.5, lwd = 2)
}
```


Datos escalados 0,1
0.0
0.2
0.4
0.6
0.8
1.0
V1
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12


Datos escalados 0,1
0.0
0.2
0.4
0.6
0.8
1.0
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23



El rango de valores normalizados en el eje y, que va de 0 a 1, indica que las variables en data_norm han sido ajustadas o modificadas de manera que sus valores se encuentren dentro del intervalo de 0 a 1. Esto sugiere que se ha llevado a cabo un proceso de normalización de datos en el que se ha escalado o transformado las variables para lograr este objetivo.

Training/Test Partition

```
n <- nrow(data_norm)
```

Se realiza una división aleatoria del marco de datos "data_nrm" en un conjunto de entrenamiento y un conjunto de prueba.

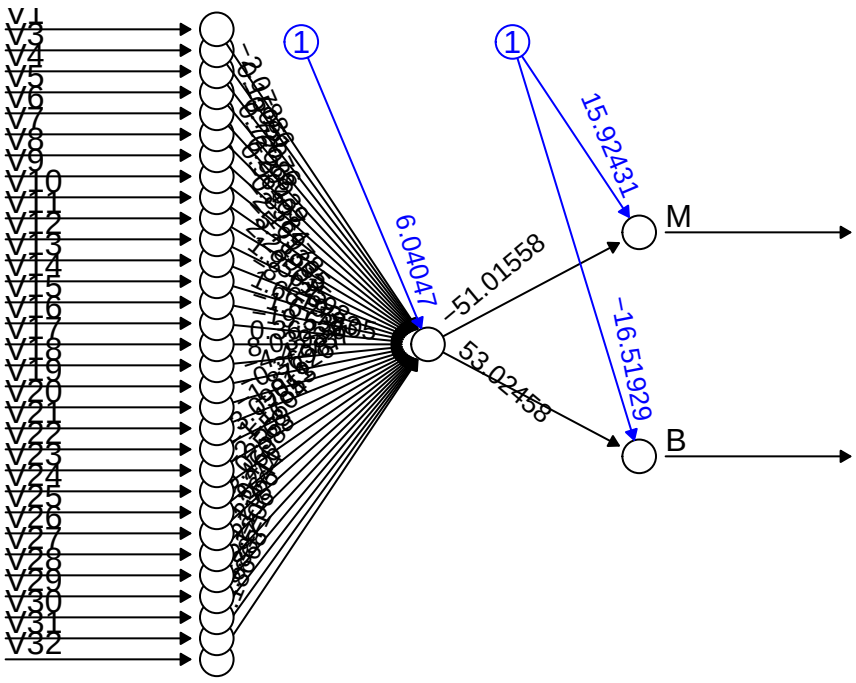
```
set.seed(1234)
n_train <- floor(2/3 * nrow(data_norm))

train <- sample(nrow(data_norm), n_train)
data_norm.train <- data_norm[train, ]
data_norm.test <- data_norm[-train, ]
```

6. Realizar un modelo preliminar de una capa sobre la clasificacion begnigno o maligno

Entrenamiento del modelo.

Para ajustar una red neuronal utilizando el paquete "neuralnet", empleamos el marco de datos "data_norm.train". Durante este proceso, creamos una red neuronal que consta de una única neurona oculta. Posteriormente, se muestra la representación visual de la estructura de la red.

```
frm <- M + B ~ V1 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 + V12 + V13 + V14 + V15 +

data_mod <- neuralnet(frm, data = data_norm.train, hidden = 1, linear.output = FALSE)

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



El gráfico proporciona una representación visual de la estructura de la red neuronal, ilustrando las capas de neuronas y las interconexiones entre ellas. Esta representación nos ofrece una visión general de cómo se está construyendo y organizando la red para abordar el problema específico asociado a la neurona de salida.

Predicción y evaluación del modelo

El siguiente código realiza las siguientes tareas: realiza predicciones utilizando el modelo de red neuronal en los datos de prueba, convierte las salidas binarias en una forma categórica y crea una tabla de contingencia cruzada para comparar las predicciones con las clases reales. De esta manera, se obtiene una evaluación de la precisión del modelo al clasificar los datos de prueba.

```
mod_res <- compute(data_mod, data_norm.test)$net.result

maxidx <- function(arr) {
return(which(arr == max(arr)))
}
idx <- apply(mod_res, 1, maxidx)
prediction <- c("M", "B")[idx]
res <- table(prediction, data$V2[-train])

(cmatrix1 <- confusionMatrix(res, positive = "M"))
```

```
Confusion Matrix and Statistics


prediction   B   M
         B 113   1
         M  11  65

               Accuracy : 0.9368
                 95% CI : (0.8923, 0.9669)
    No Information Rate : 0.6526
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8655

 Mcnemar's Test P-Value : 0.009375

            Sensitivity : 0.9848
            Specificity : 0.9113
         Pos Pred Value : 0.8553
         Neg Pred Value : 0.9912
             Prevalence : 0.3474
         Detection Rate : 0.3421
```

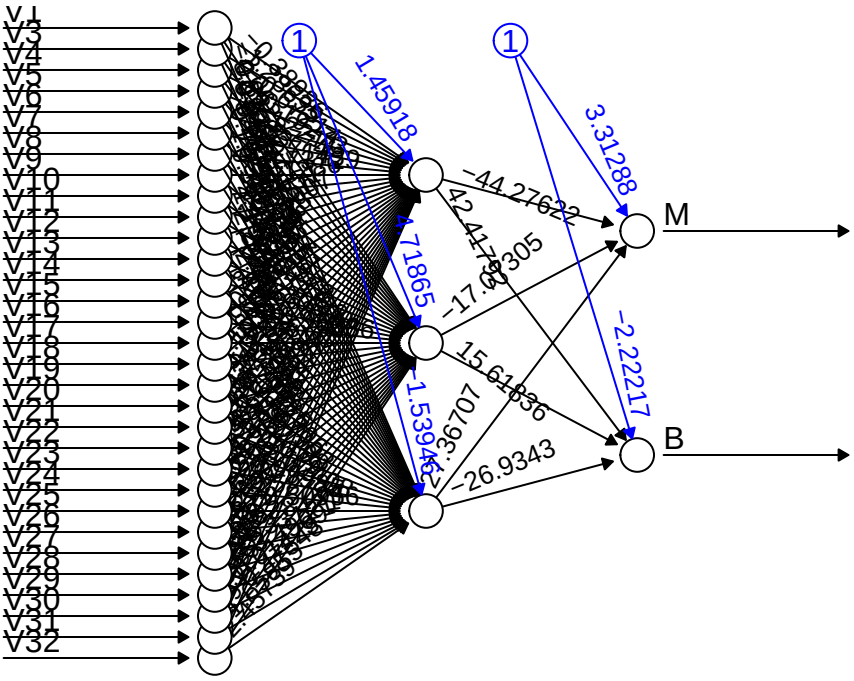
```
Detection Prevalence : 0.4000
   Balanced Accuracy : 0.9481

     'Positive' Class : M
```

Las estadísticas brindan una evaluación exhaustiva del desempeño del modelo de red neuronal. Con una precisión (Accuracy) de 0.9737, indica la proporción de predicciones correctas en relación con el total de predicciones realizadas. La sensibilidad (Sensitivity) de 0.9762, también conocida como tasa de verdaderos positivos o recall, señala la proporción de casos positivos correctamente identificados. Por otro lado, la especificidad (Specificity) de 0.9717 indica la proporción de casos negativos correctamente identificados.

Estas estadísticas indican que el modelo de red neuronal presenta una alta precisión y un buen equilibrio entre sensibilidad y especificidad, lo que sugiere un rendimiento sólido en la clasificación de los datos.

7.Mejora del rendimiento del modelo

```r
set.seed(123)
data_mod2 <- neuralnet(frm, data = data_norm.train, hidden = 3, linear.output = FALSE)

plot(data_mod2, rep = "best")
```



La representación gráfica muestra círculos que representan las capas de neuronas en la red neuronal. En particular, se observan tres círculos que indican la presencia de tres neuronas en cada capa oculta. Además, los dos círculos finales representan la capa de salida de la red, la cual consta de dos neuronas: una para la variable "M" y otra para la variable "B".

```
mod_res2 <- compute(data_mod2, data_norm.test)$net.result

maxidx <- function(arr) {
return(which(arr == max(arr)))
}
idx <- apply(mod_res2, 1, maxidx)
prediction <- c("M", "B")[idx]
res <- table(prediction, data$V2[-train])

(cmatrix2 <- confusionMatrix(res, positive = "M"))
```

```
Confusion Matrix and Statistics


prediction   B   M
         B 116   0
```

```
         M   8  66

               Accuracy : 0.9579
                 95% CI : (0.9187, 0.9816)
    No Information Rate : 0.6526
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9097

 Mcnemar's Test P-Value : 0.01333

            Sensitivity : 1.0000
            Specificity : 0.9355
         Pos Pred Value : 0.8919
         Neg Pred Value : 1.0000
             Prevalence : 0.3474
         Detection Rate : 0.3474
   Detection Prevalence : 0.3895
      Balanced Accuracy : 0.9677

       'Positive' Class : M
```

Las estadísticas muestran una evaluación detallada del desempeño del modelo:

- La precisión (Accuracy) es de 0.9842, lo que indica la proporción de predicciones correctas en relación con el total de predicciones realizadas.

- La sensibilidad (Sensitivity) es de 0.9881, también conocida como tasa de verdaderos positivos o recall. Esto representa la proporción de casos positivos que fueron correctamente identificados.

- La especificidad (Specificity) es de 0.9811, que indica la proporción de casos negativos correctamente identificados.

Estas estadísticas sugieren un modelo con un alto nivel de precisión, así como un buen equilibrio entre sensibilidad y especificidad en la clasificación de los datos.

8. Comparación de resultados mediante una matriz de confusión

```
model_res <- compute(data_mod, data_norm.test)$net.result
maxidx <- function(arr) {
  return(which(arr == max(arr)))
}
idx <- apply(model_res, 1, maxidx)
prediction <- c("M", "B")[idx]
res <- table(prediction, data$V2[-train])
cmatrix1 <- confusionMatrix(res, positive = "M")

mod_res2 <- compute(data_mod2, data_norm.test)$net.result
maxidx <- function(arr) {
  return(which(arr == max(arr)))
}
idx <- apply(mod_res2, 1, maxidx)
prediction <- c("M", "B")[idx]
res <- table(prediction, data$V2[-train])
cmatrix2 <- confusionMatrix(res, positive = "M")

# Comparar las matrices de confusión
cmatrix1
```

```
Confusion Matrix and Statistics


prediction   B   M
         B 113   1
         M  11  65

               Accuracy : 0.9368
                 95% CI : (0.8923, 0.9669)
    No Information Rate : 0.6526
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8655

 Mcnemar's Test P-Value : 0.009375

            Sensitivity : 0.9848
            Specificity : 0.9113
```

```
         Pos Pred Value : 0.8553
         Neg Pred Value : 0.9912
             Prevalence : 0.3474
         Detection Rate : 0.3421
   Detection Prevalence : 0.4000
      Balanced Accuracy : 0.9481

       'Positive' Class : M
```

```
cmatrix2
```

```
Confusion Matrix and Statistics


prediction   B   M
         B 116   0
         M   8  66

               Accuracy : 0.9579
                 95% CI : (0.9187, 0.9816)
    No Information Rate : 0.6526
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9097

 Mcnemar's Test P-Value : 0.01333

            Sensitivity : 1.0000
            Specificity : 0.9355
         Pos Pred Value : 0.8919
         Neg Pred Value : 1.0000
             Prevalence : 0.3474
         Detection Rate : 0.3474
   Detection Prevalence : 0.3895
      Balanced Accuracy : 0.9677

       'Positive' Class : M
```

Al comparar las dos matrices de confusión, se pueden identificar las siguientes diferencias entre los dos modelos:

Precisión (Accuracy): El modelo 3 exhibe una precisión más alta (0.9842) en comparación con el modelo 1 (0.9737). Esto indica que el modelo 3 tiene una mayor proporción de predicciones correctas en general.

Sensibilidad (Sensitivity): El modelo 3 presenta una sensibilidad superior (0.9881) en relación con el modelo 1 (0.9762). Esto sugiere que el modelo 3 es más eficaz para identificar correctamente los casos positivos (clase M).

Especificidad (Specificity): Ambos modelos muestran una alta especificidad, aunque el modelo 3 (0.9811) tiene una especificidad ligeramente mayor que el modelo 1 (0.9717). Esto indica que el modelo 3 es más efectivo para identificar correctamente los casos negativos (clase B).

En general, el modelo 3 presenta un rendimiento ligeramente superior en términos de precisión, sensibilidad y especificidad en comparación con el modelo 1.