Tipología y ciclo de vida de los datos: Práctica 2 - Limpieza y análisis de datos

Autoras: Eva Garía Ocaña y Carmen nieves Ojeda Guerra
Enero 2022

Descripción del dataset

El *dataset* escogido para el trabajo es el **Breast Cancer Wisconsin (Diagnostic) Data Set** de UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)) que analiza la incidencia del cáncer de mama en una población de 569 mujeres.

Motivación: Según la OMS, el cáncer de mama es el tipo de cáncer más común, con más de 2,2 millones de casos en 2020 y cerca de una de cada 12 mujeres enfermarán de cáncer de mama a lo largo de su vida. El cáncer de mama es la principal causa de mortalidad en las mujeres. En 2020, alrededor de 685000 mujeres fallecieron como consecuencia de esa enfermedad. Sin embargo, desde 1980 se han realizado importantes avances en el tratamiento del cáncer de mama debido a la combinación de la **detección precoz** y las terapias eficaces, basadas en cirugía, radioterapia y farmacoterapia (https://www.who.int/es/news-room/fact-sheets/detail/breast-cancer, marzo 2021). En España, según los últimos datos recogidos por el Sistema Europeo de Información del Cáncer (ECIS) de 2020, aproximadamente el 30% de los cánceres diagnosticados en mujeres se originan en la mama (https://www.geicam.org/sala-de-prensa/el-cancer-de-mama-en-espana, 2021). Una detección temprana de la presencia de células cancerosas malignas aumenta la posibilidad de vida de las pacientes, sobre todo cuando se localiza un tumor pequeño y aún no ramificado. Para la realización de este estudio, el análisis de los datos obtenidos a partir de la información de estas células puede detectar la malignidad o benignidad de las mismas.

El objetivo de este trabajo es analizar el conjunto de datos indicado anteriormente para ver qué atributos o características influyen mayormente en que un tumor sea considera benigno o maligno y, posteriormente, realizar un clasificador que ayude a indicar si una muestra de tumor es benigna o no.

Potencial analítico del conjunto de datos: Debido a las características del conjunto de datos (atributos de los que dispone) se pueden plantear preguntas que ayuden a comprender mejor cómo afectan los factores clave en la población estudiada y analizar qué variables pueden ser decisivas a la hora de conocer si un tumor es benigno o maligno. Una medida del potencial del conjunto de datos se puede ver en la cantidad de artículos de reconocido prestigio que lo usan en sus investigaciones, como por ejemplo:

1- Chien-Hsing Chen, *A hybrid intelligent model of analyzing clinical breast cancer data using clustering techniques with feature selection*, Applied Soft Computing, Volume 20, 2014, Pages 4-14, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2013.10.024.

2- Lingxi Peng, Wenbin Chen, Wubai Zhou, Fufang Li, Jin Yang, Jiandong Zhang, *An immune-inspired semi-supervised algorithm for breast cancer diagnosis*, Computer Methods and

Programs in Biomedicine, Volume 134, 2016, Pages 259-265, ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2016.07.020.

3- Bichen Zheng, Sang Won Yoon, Sarah S. Lam, *Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms*, Expert Systems with Applications, Volume 41, Issue 4, Part 1, 2014, Pages 1476-1482, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2013.08.044.

4- S. Punitha, Thompson Stephan, Amir H. Gandomi, *A Novel Breast Cancer Diagnosis Scheme With Intelligent Feature and Parameter Selections*, Computer Methods and Programs in Biomedicine, 2021, 106432, ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2021.106432.

Para comenzar el análisis, se carga el *dataset* desde la fuente origen:

```
breast_data <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data", sep = ",", col.names = c("ID", "Diagnosis", "Radius_mean", "Texture_mean", "Perimeter_mean", "Area_mean", "Smoothness_mean", "Compactness_mean", "Concavity_mean", "Concave.points_mean", "Symmetry_mean", "Fractal_dimension_mean", "Radius_se", "Texture_se", "Perimeter_se", "Area_se", "Smoothness_se", "Compactness_se", "Concavity_se", "Concave.points_se", "Symmetry_se", "Fractal_dimension_se", "Radius_worst", "Texture_worst","Perimeter_worst", "Area_worst", "Smoothness_worst", "Compactness_worst", "Concavity_worst", "Concave.points_worst", "Symmetry_worst", "Fractal_dimension_worst"), stringsAsFactors = T)
str(breast_data)
```

```
'data.frame':   569 obs. of  32 variables:
 $ ID                     : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 ...
 $ Diagnosis              : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
 $ Radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
 $ Texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
 $ Perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
 $ Area_mean              : num  1001 1326 1203 386 1297 ...
 $ Smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ Compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ Concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ Concave.points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ Symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
 $ Fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
 $ Radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
 $ Texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
 $ Perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
 $ Area_se                : num  153.4 74.1 94 27.2 94.4 ...
 $ Smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
 $ Compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
 $ Concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
 $ Concave.points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
 $ Symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
 $ Fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
 $ Radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
 $ Texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
 $ Perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
 $ Area_worst             : num  2019 1956 1709 568 1575 ...
 $ Smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
 $ Compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
 $ Concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
 $ Concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
 $ Symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
 $ Fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

Los atributos se calculan a partir de una imagen digitalizada de una masa mamaria. Describen características de los núcleos celulares presentes en la imagen. Como se puede observar existen 569 registros con 32 variables, de las cuales el atributo **Diagnosis** es la variable objetivo. De

cada célula se obtienen 10 características, existiendo 3 datos para cada una (con el sufijo **mean**, **se** y **worst**). Estas son:

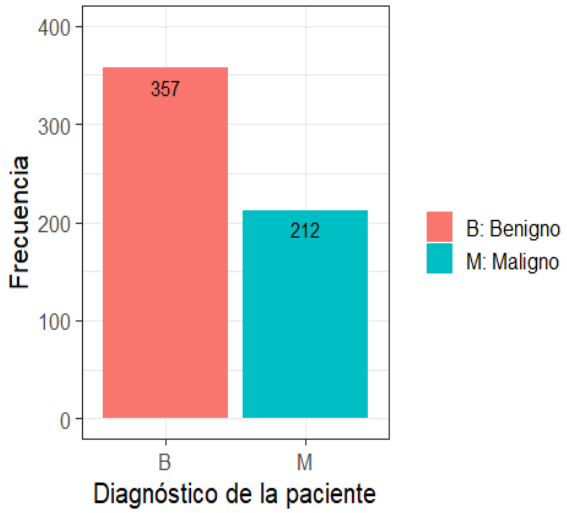
Radius: media de las distancias del centro a los puntos del perímetro de la célula. Compuesta por los atributos **Radius_mean**, **Radius_se** y **Radius_worst**.
Texture: desviación estándar de los valores de la escala de grises. Compuesta por los atributos **Texture_mean**, **Texture_se** y **Texture_worst**.
Perimeter: perímetro de la célula. Compuesta por los atributos **Perimeter_mean**, **Perimeter_se** y **Perimeter_worst**.
Area: área de la célula. Compuesta por los atributos **Area_mean**, **Area_se** y **Area_worst**.
Smothness: variación local de las longitudes de los radios. Compuesta por los atributos **Smothness_mean**, **Smothness_se** y **Smothness_worst**.
Compactness: perimeter^2 / area - 1.0. Compuesta por los atributos **Compactness_mean**, **Compactness_se** y **Compactness_worst**.
Concavity: gravedad de las partes cóncavas del contorno. Compuesta por los atributos **Concavity_mean**, **Concavity_se** y **Concavity_worst**.
Concave points: número de porciones cóncavas del contorno. Compuesta por los atributos **Concave.points_mean**, **Concave.points_se** y **Concave.points_worst**.
Symmetry: simetría de la célula. Compuesta por los atributos **Symmetry_mean**, **Symmetry_se** y **Symmetry_worst**.
Fractal dimension: "aproximación al borde" - 1. Compuesta por los atributos **Fractal_dimension_mean**, **Fractal_dimension_se** y **Fractal_dimension_worst**.

Además de las características anteriores de la células analizadas en una muestra, también se conoce:

ID (int): identificador de la muestra.
Diagnosis (Factor): variable objetivo en la que "B" indica benignidad y "M", malignidad de la muestra.

El total de mujeres de la muestra con un diagnóstico benigno o maligno se muestra en la siguiente gráfica:

```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')

datos <- breast_data %>% group_by(Diagnosis) %>% summarise(Total=n())
ggplot(datos, aes(x = Diagnosis, y=Total, fill=Diagnosis) ) +
  geom_bar(width = 0.9, stat="identity", position = position_dodge()) +
ylim(c(0,400))+ labs(x="Diagnóstico de la paciente", y= "Frecuencia") +
  labs(fill = "")+

  geom_text(aes(label=Total), vjust=1.6, color="black",
              position = position_dodge(0.9),  size=4.0
            ) +

  theme_bw(base_size = 15) + scale_fill_hue(labels = c("B: Benigno", "M:
Maligno"))
```



Los datos no están bien distribuidos ya que existe un mayor número de muestras con un valor 'B' de la variable objetivo.

Integración y selección de los datos de interés a analizar

Los datos proceden de una única fuente de datos, por lo que no hay necesidad de intergrar datos de fuentes diferentes.

Asimismo, después de analizar los datos se considera que hay que eliminar el atributo **ID**, ya que no aporta información para poder clasificar una muestra de datos como posible benigna o maligna. Así, el nuevo *dataset* sería el siguiente:

```
breast_data_def <- breast_data[c(2:32)]
```

El conjunto de datos tendrá 31 variables, aunque posteriormente se hará un análisis más profundo para disminuir la dimiensión.

Limpieza de los datos

Identificación y tratamiento de elementos vacíos, ceros o nulos

Analizando los datos:

```
summary(breast_data_def)
```

```
 Diagnosis  Radius_mean      Texture_mean   Perimeter_mean     Area_mean      Smoothness_mean   Compactness_mean  Concavity_mean
 B:357      Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5   Min.   :0.05263   Min.   :0.01938   Min.   :0.00000
 M:212      1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956
            Median :13.370   Median :18.84   Median : 86.24   Median : 551.1   Median :0.09587   Median :0.09263   Median :0.06154
            Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9   Mean   :0.09636   Mean   :0.10434   Mean   :0.08880
            3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070
            Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.   :0.34540   Max.   :0.42680
 Concave.points_mean Symmetry_mean    Fractal_dimension_mean   Radius_se       Texture_se      Perimeter_se       Area_se
 Min.   :0.00000     Min.   :0.1060   Min.   :0.04996        Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802
 1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850
 Median :0.03350     Median :0.1792   Median :0.06154        Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530
 Mean   :0.04892     Mean   :0.1812   Mean   :0.06280        Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337
 3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190
 Max.   :0.20120     Max.   :0.3040   Max.   :0.09744        Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200
 Smoothness_se      Compactness_se     Concavity_se     Concave.points_se   Symmetry_se       Fractal_dimension_se  Radius_worst   Texture_worst
 Min.   :0.001713   Min.   :0.002252   Min.   :0.00000   Min.   :0.000000   Min.   :0.007882   Min.   :0.0008948   Min.   : 7.93   Min.   :12.02
 1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509   1st Qu.:0.007638   1st Qu.:0.015160   1st Qu.:0.0022480   1st Qu.:13.01   1st Qu.:21.08
 Median :0.006380   Median :0.020450   Median :0.02589   Median :0.010930   Median :0.018730   Median :0.0031870   Median :14.97   Median :25.41
 Mean   :0.007041   Mean   :0.025478   Mean   :0.03189   Mean   :0.011796   Mean   :0.020542   Mean   :0.0037949   Mean   :16.27   Mean   :25.68
 3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205   3rd Qu.:0.014710   3rd Qu.:0.023480   3rd Qu.:0.0045580   3rd Qu.:18.79   3rd Qu.:29.72
 Max.   :0.031130   Max.   :0.135400   Max.   :0.39600   Max.   :0.052790   Max.   :0.078950   Max.   :0.0298400   Max.   :36.04   Max.   :49.54
 Perimeter_worst    Area_worst      Smoothness_worst  Compactness_worst Concavity_worst  Concave.points_worst Symmetry_worst
 Min.   : 50.41   Min.   : 185.2   Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000      Min.   :0.1565
 1st Qu.: 84.11   1st Qu.: 515.3   1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493      1st Qu.:0.2504
 Median : 97.66   Median : 686.5   Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993      Median :0.2822
 Mean   :107.26   Mean   : 880.6   Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461      Mean   :0.2901
 3rd Qu.:125.40   3rd Qu.:1084.0   3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140      3rd Qu.:0.3179
 Max.   :251.20   Max.   :4254.0   Max.   :0.22260   Max.   :1.05800   Max.   :1.2520   Max.   :0.29100      Max.   :0.6638
 Fractal_dimension_worst
 Min.   :0.05504
 1st Qu.:0.07146
 Median :0.08004
 Mean   :0.08395
 3rd Qu.:0.09208
 Max.   :0.20750
```

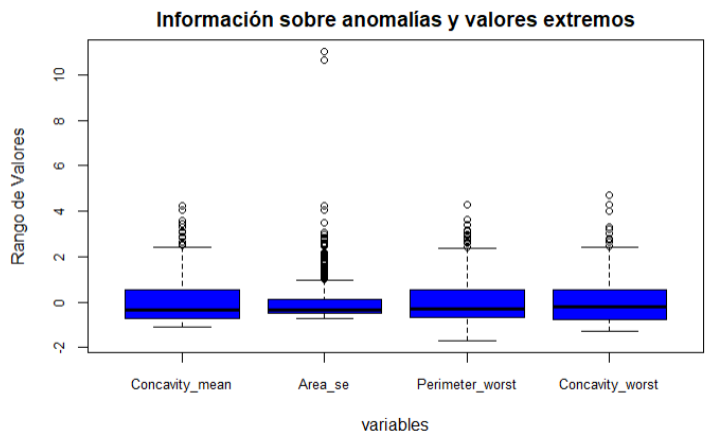
Se puede observar que no hay datos perdidos (no hay valores NA) y que hay una serie de atributos que tienen valor 0 en alguna muestra, pero todos tienen que ver con la concavidad de la célula, por lo que se asume que son valores correctos. En caso de que hubiera elementos nulos se puede, entre otros:

- Eliminar los registros donde existan elementos nulos: solo debe utilizarse cuando el proceso de recogida de datos es aleatorio (en otro caso, produce un sesgo) y si hay datos suficientes (si no, puede afectar a la representación de la muestra).
- Cambiar los valores nulos por la media, mediana o moda de valores de ese atributo: distorsiona la verdadera distribución de la variable y la correlación entre variables, así como dificulta la estimación de la varianza.

Identificación y tratamiento de valores extremos

En la siguiente imagen se pueden observar los valores extremos (**outliers**) de una serie de atributos del conjunto de datos:

```
boxplot(scale(breast_data_def[,c(8, 15, 24, 28)]), xlab ="variables", cex.axis=0
.8, ylab = "Rango de Valores", col="blue", main="Información sobre anomalías y v
alores extremos")
```



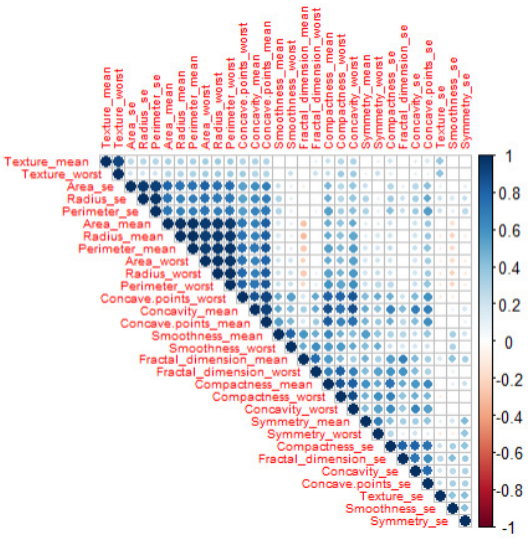
Se puede observar que hay datos atípicos en los atributos analizados. Si los valores atípicos corresponden con tumores malignos, pueden ser valores correctos y no errores en las medidas, sin embargo, cuando los valores atípicos están en en muestras benignas, se podrían corregir. En un apartado posterior se analizará si corresponden con muestras malignas o no y si no lo son, se tratarán estos valores.

Análisis de los datos

Planificación de los análisis a aplicar

Inicialmente se analizará la correlación total (sin contar con la variable objetivo), para comprobar si se pueden eliminar variables debido a la alta correlación entre ellas. Así:

```
if (!require('corrplot')) install.packages('corrplot'); library('corrplot
')
if (!require('ggcorrplot')) install.packages('ggcorrplot'); library('ggco
rrplot')
cor.mtest <- function(mat, ...) {
    mat <- as.matrix(mat)
    n <- ncol(mat)
    p.mat<- matrix(NA, n, n)
    diag(p.mat) <- 0
    for (i in 1:(n - 1))
        for (j in (i + 1):n) {
            tmp <- cor.test(mat[, i], mat[, j], ...)
            p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
        }
    colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
    p.mat
}
# matriz de los p-values de la correlación
p.mat_total <- cor.mtest(breast_data_def[c(2:31)])
M_total <- cor(breast_data_def[c(2:31)])
corrplot(M_total, type="upper", order="hclust", method="circle",
  p.mat = p.mat_total, sig.level = 0.05, insig = "blank", tl.cex  = 0.65)
```



Revisando la información de correlación, se puede ver que **Perimeter_worst**, **Area_mean**, **Radius_mean**, **Perimeter_mean**, **Area_worst** y **Radius_worst** tienen una correlación de 1. Asimismo, **Area_se**, **Radius_se** y **Permiter_se** también tienen una correlación de 1. Lo mismo ocurre entre **Concavity_mean**, **Concave.points_worst** y **Concave.points_mean**. En todos los casos se puede eliminar todas ellas, menos una de cada grupo. Así:

```
d_num <- as.numeric(breast_data_def[,1])
col_eliminar <- c("Area_mean", "Radius_mean", "Perimeter_mean", "Area_wor
st", "Radius_worst", "Radius_se", "Perimeter_se", "Concave.points_worst",
"Concave.points_mean")

breast_data_def_corr <- breast_data_def[, !(names(breast_data_def) %in% c
ol_eliminar)]
```

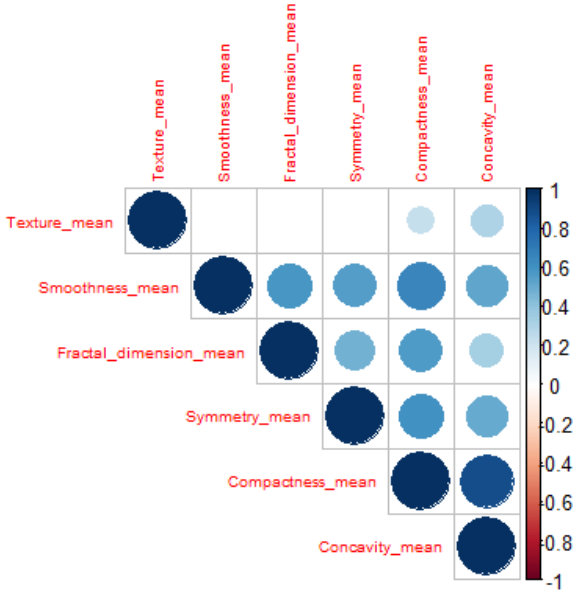
La nueva dimensión del conjunto de datos es de 22 variables.

A continuación, se seleccionan los grupos dentro del conjunto de datos que pueden resultar de interés para analizar y/o comparar. Sin embargo, en un análisis posterior se verá que no todos los atributos se van a utilizar. Los datos se van a dividir en tres grandes grupos: media (**mean**), error estándar (**se**) y peor (**worst**):

```
breast_data_groups <- breast_data_def_corr
# Agrupación mean
breast_data_groups.mean <- breast_data_groups[c('Texture_mean', 'Smoothne
ss_mean', 'Compactness_mean', 'Concavity_mean', 'Symmetry_mean', 'Fractal
_dimension_mean')]
# Agrupación se
breast_data_groups.se <- breast_data_groups[c('Texture_se', 'Area_se', 'S
moothness_se', 'Compactness_se', 'Concavity_se', 'Concave.points_se', 'Sy
mmetry_se', 'Fractal_dimension_se')]
# Agrupación worst
breast_data_groups.worst <- breast_data_groups[c('Texture_worst', 'Perime
ter_worst', 'Smoothness_worst', 'Compactness_worst', 'Concavity_worst', '
Symmetry_worst', 'Fractal_dimension_worst')]
```

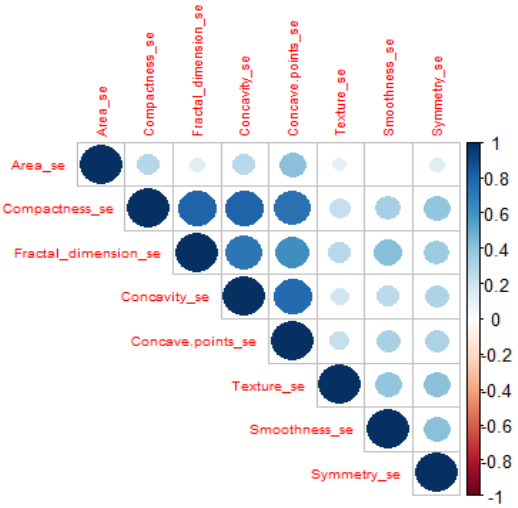
Se va a analizar la correlación entre las variables de cada grupo. Para el grupo **mean** se tiene la siguiente matriz de correlación:

```
if (!require('ggcorrplot')) install.packages('ggcorrplot'); library('ggco
rrplot')
# matriz de los p-values de la correlación
p.mat_mean <- cor.mtest(breast_data_groups.mean)
M_mean <- cor(breast_data_groups.mean)
corrplot(M_mean, type="upper", order="hclust", method="circle",
   p.mat = p.mat_mean, sig.level = 0.05, insig = "blank", tl.cex  = 0.65)
```



A partir del análisis de correlación, se puede eliminar **Compactness_mean**, ya que tiene una correlación con **Concavity_mean** cercana a 1. S**e** se tiene la siguiente correlación:

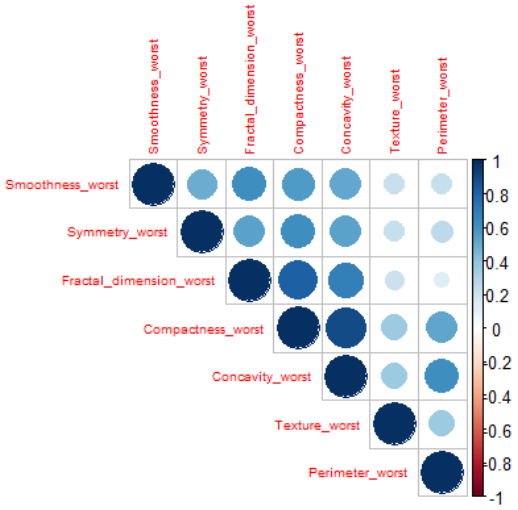
```
if (!require('ggcorrplot')) install.packages('ggcorrplot'); library('ggco
rrplot')
# matriz de los p-values de la correlación
p.mat_se <- cor.mtest(breast_data_groups.se)
M_se <- cor(breast_data_groups.se)
corrplot(M_se, type="upper", order="hclust", method="circle",
     p.mat = p.mat_se, sig.level = 0.05, insig = "blank", tl.cex  = 0.65)
```



En este caso no se elimina ninguna variable. Para el grupo **worst** se tiene la siguiente correlación:

```
if (!require('ggcorrplot')) install.packages('ggcorrplot'); library('ggco
rrplot')
# matriz de los p-values de la correlación
p.mat_worst <- cor.mtest(breast_data_groups.worst)
M_worst <- cor(breast_data_groups.worst)
```

```
corrplot(M_worst, type="upper", order="hclust", method="circle",
         p.mat = p.mat_worst, sig.level = 0.05, insig = "blank", tl.cex
= 0.65)
```



A partir del análisis de correlación, se puede eliminar **Compactness_worst**, ya que tiene una correlación con **Concavity_worst** cercana a 1.

Por tanto, el nuevo conjunto de datos será:

```
breast_data_def_grouping <- breast_data_groups[c('Diagnosis', 'Texture_me
an', 'Smoothness_mean', 'Concavity_mean', 'Symmetry_mean', 'Fractal_dimen
sion_mean', 'Texture_se', 'Area_se', 'Smoothness_se', 'Compactness_se', '
Concavity_se', 'Concave.points_se', 'Symmetry_se', 'Fractal_dimension_se'
, 'Texture_worst', 'Perimeter_worst', 'Smoothness_worst', 'Concavity_wors
t', 'Symmetry_worst', 'Fractal_dimension_worst')]

dim(breast_data_def_grouping)

## [1] 569  20
```

Finalmente, el conjunto de datos tendrá 20 atributos sobre los que se hará el análisis posterior, que podrá modificar la dimensión.

En apartados anteriores se comprobó que los valores de la mediana de los atributos están muy cerca de la mitad de la caja, indicando que los valores de los datos son más o menos simétricos. En el análisis descriptivo que se realizó en el apartado "Identificación y tratamiento de elementos vacíos, ceros o nulos" se puede comprobar que los atributos no están todos en el mismo rango, existiendo en algunos de ellos, una gran diferencia entre los valores mínimo y máximo.

En los siguientes apartados se hará un estudio de la normalidad y homogeneidad de la varianza, así como se planterán algunas pruebas estadísticas.

Normalidad y homogeneidad de la varianza

Para comprobar cuales de las variables cuantitativas de las que disponemos sigue una distribución normal se empleará la prueba de normalidad de **Shapiro-Wilk**. Si la variable tiene un p-valor superior a 0.05 se considera que sigue una distribucion normal:

```
alpha = 0.05
col.names = colnames(breast_data_def_grouping)
for (i in 1:ncol(breast_data_def_grouping)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n\n")
  if (is.integer(breast_data_def_grouping[,i]) | is.numeric(breast_data_d
ef_grouping[,i])) {
    p_val = shapiro.test(breast_data_def_grouping[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      if (i < ncol(breast_data_def_grouping)) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
}
```

```
## Variables que no siguen una distribución normal:
##
## Texture_mean, Smoothness_mean,
## Concavity_mean, Symmetry_mean, Fractal_dimension_mean,
## Texture_se, Area_se, Smoothness_se,
## Compactness_se, Concavity_se, Concave.points_se,
## Symmetry_se, Fractal_dimension_se, Texture_worst,
## Perimeter_worst, Smoothness_worst, Concavity_worst,
## Symmetry_worst, Fractal_dimension_worst
```

Se puede observar que todos los atributos no siguen una distribución normal. En cuanto a la homogeneidad de la varianza, se va a utilizar el test de **Fligner-Killeen**, debido a ese mismo hecho. Para ello, se van a comparar las varianzas para un diagnóstico benigno y maligno de los distintos atributos. Así:

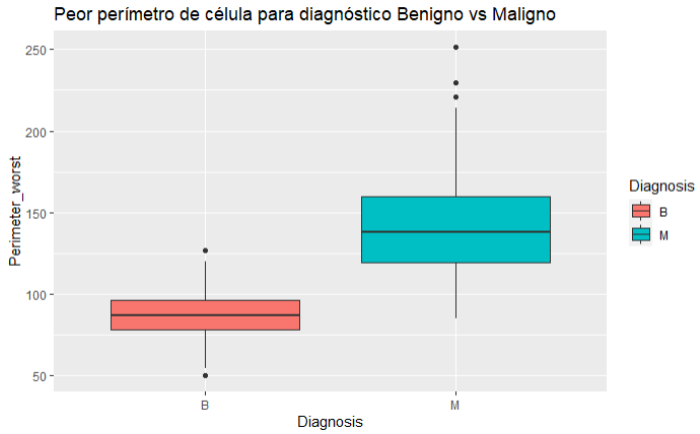
```
alpha = 0.05
col.names = colnames(breast_data_def_grouping)
datosB <- filter(breast_data_def_grouping, breast_data_def_grouping$Diagn
osis=="B")
datosM <- filter(breast_data_def_grouping, breast_data_def_grouping$Diagn
osis=="M")
for (i in 2:ncol(breast_data_def_grouping)) {
  if (i == 2) cat("Los siguientes atributos presenta varianzas estadístic
amente diferentes para los dos grupos de la variable objetivo:\n\n")
  a <- datosB[,i]
  b <- datosM[,i]
  p_val <- fligner.test(x=list(a,b))$p.value
  if (p_val < alpha)
        cat(col.names[i], "\n")
}
```

```
## Los siguientes atributos presenta varianzas estadísticamente diferente
s para los dos grupos de la variable objetivo:
##
## Concavity_mean
## Fractal_dimension_mean
## Texture_se
## Area_se
```

```
## Smoothness_se
## Compactness_se
## Perimeter_worst
## Concavity_worst
## Symmetry_worst
## Fractal_dimension_worst
```

Por ejemplo, si analizamos gráficamente la relación del atributo **Perimeter_worst** con la variable objetivo:

```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')

ggplot(data=breast_data_def_grouping,aes(x=Diagnosis,y=Perimeter_worst,fi
ll=Diagnosis))+geom_boxplot()+ggtitle("Peor perímetro de célula para diag
nóstico Benigno vs Maligno")
```



Se observa cómo la varianza es mayor en los casos de malignidad, aunque la mediana está centrada en ambos diagnósticos. Esto también ocurre con el resto de las variables que posteriormente se seleccionarán para la generación de los clasificadores.

Por tanto, del análisis anterior se puede ver que en el conjunto de datos, los atributos no siguen una distribución normal y 10 de ellos no presentan igualdad de varianza respecto a la variable objetivo.

Pruebas estadísticas

Por tanto:

¿Qué atributos tienen una mayor relación con la posibilidad de tener un tumor benigno o maligno?

Puesto que las variables no se ajustan a una distribucion normal, se empleará el coeficiente de correlación de **Spearman** para comprobar qué variables muestran una mayor relación con el tipo de tumor (benigno o maligno):

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
# Calcular el coeficiente de correlación para cada variable cuantitativa
# con respecto al campo "Diagnosis"
for (i in 2:(ncol(breast_data_def_grouping))) {
  if (is.integer(breast_data_def_grouping[,i]) | is.numeric(breast_data_d
```

```
ef_grouping[,i])) {
    spearman_test = cor.test(breast_data_def_grouping[,i], as.numeric(bre
ast_data_def_grouping[,1]), method = "spearman",exact=FALSE)
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    # Se añade la fila a la matriz
    pair = matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(breast_data_def_
grouping)[i]
  }
}
cat("Se ordenan los valores en base a su correlación, para conocer qué va
riables se correlacionan más con el diagnóstico: \n")
## Se ordenan los valores en base a su correlación, para conocer qué vari
ables se correlacionan más con el diagnóstico:
corr_matrix[sort(abs(corr_matrix[,1]),decreasing=T,index.return=T)[[2]],]
##                            estimate       p-value
## Perimeter_worst          0.79631860 6.742652e-126
## Concavity_mean           0.73330788  4.509903e-97
## Area_se                  0.71418372  6.704014e-90
## Concavity_worst          0.70573401  6.459758e-87
## Concave.points_se        0.48871728  1.702914e-35
## Texture_worst            0.47672008  1.262636e-33
## Concavity_se             0.47033813  1.164972e-32
## Texture_mean             0.46197092  2.000481e-31
## Smoothness_worst         0.42551309  1.992578e-26
```

Se aprecia que las variables que más influyen son: **Perimeter_worst**, **Concavity_mean**, **Area_se** y **Concavity_worst** (todas por encima del 50% de estimación) que, además, están en el listado de atributos que presentan varianzas estadísticamente diferentes para los dos grupos de la variable objetivo (la diferencia de comportamiento respecto a los valores de la variable objetivo se mostró gráficamente para algunos de ellos).

Por otro lado:

¿Los valores outliers son errores o corresponden con tumores malignos?

Para comprobar esta hipótesis, que se planteó anteriormente, se empleará la columna **Perimeter_worst** puesto que es la que mayor correlación presenta con la variable objetivo, para confirmar o descartar esta hipótesis a través del test de **Mann-Whitney** (contrasta si dos muestras proceden de poblaciones equidistribuidas, es decir, comprueba que los valores de una población no tienden a ser mayores que los de otra), puesto que los valores no siguen una distribucion normal y el número de muestras es pequeño.

```
# primer grupo de muestras: conjunto de datos con los registros que tiene
n los outliers de Perimeter_worst
out_perimeter_worst <- boxplot.stats(breast_data_def_grouping$Perimeter_w
orst)$out
out_perimeter_worst_ind <- which(breast_data_def_grouping$Perimeter_worst
%in% out_perimeter_worst)
out_perimeter_diagnosis <- breast_data_def_grouping[out_perimeter_worst_i
```

```
nd, ]
# segundo grupo de muestras: conjunto de datos con los registros que NO t
ienen los outliers de Perimeter_worst
in_perimeter_worst <- subset(breast_data_def_grouping$Perimeter_worst, !(
breast_data_def_grouping$Perimeter_worst %in% out_perimeter_diagnosis$Per
imeter_worst))
in_perimeter_worst_ind <- which(breast_data_def_grouping$Perimeter_worst
%in% in_perimeter_worst)
in_perimeter_diagnosis <- breast_data_def_grouping[in_perimeter_worst_ind
, ]

wilcox.test(as.numeric(out_perimeter_diagnosis$Diagnosis), as.numeric(in_
perimeter_diagnosis$Diagnosis),  alternative = "g", mu = 0)

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  as.numeric(out_perimeter_diagnosis$Diagnosis) and as.numeric(in
_perimeter_diagnosis$Diagnosis)
## W = 6832.5, p-value = 1.809e-07
## alternative hypothesis: true location shift is greater than 0
```

Al ser **p_value** menor que el valor de la significancia (0.05), se rechaza la hipótesis nula, por tanto, la media del primer grupo de muestras debe ser mayor (alternative es 'g' o 'greater') a la media del segundo grupo de muestras y si el primer grupo de muestras pertenece a tumores malignos (valor 2), tiene un valor asignado mayor al que tendrían el otro grupo en el que habría tumores benignos (valor 1) y malignos (valor 2). Esto quiere decir que podemos afirmar que el primer grupo de muestras está formado por muestras que corresponden con tumores malignos, es decir, los valores **outliers** (que forman completamente ese grupo) están en muestras con tumores malignos, luego se considerarán valores válidos. En el apartado final sobre resultados, se presenta de forma gráfica cómo los valores **outliers** pertenecen a muestras de tumores malignos.

En función de los análisis anteriores, se seleccionan las variables del conjunto de datos teniendo en cuenta cuáles de ellas están más correladas con la variable objetivo. Estas son: **Perimeter_worst**, **Concavity_mean**, **Area_se**, **Concavity_worst** y la variable objetivo **Diagnosis**. Así:

```
# se seleccionan las variables mejor correladas con la variable objetivo
del grupo inicial
data_selected <- breast_data_def_grouping[c('Diagnosis', 'Perimeter_worst
', 'Concavity_worst', 'Concavity_mean', 'Area_se')]
```

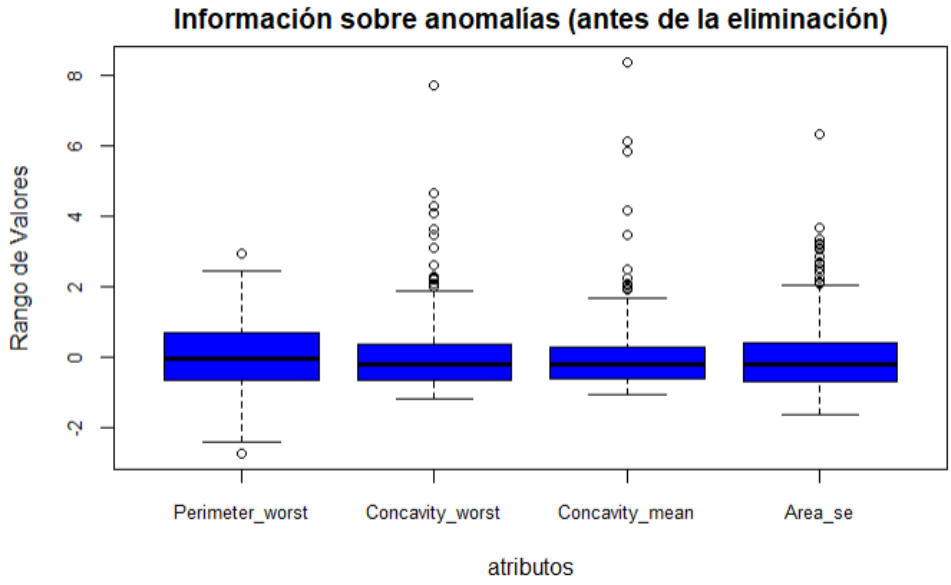
Antes de empezar el análisis y tal y como se indicó anteriormente, se va a eliminar las anomalías en las muestras de tumores benignos sustituyéndolas por la **media** de los valores de las mismas características, sin embargo, previamente hay que descubrir los verdaderos *outliers* de esas muestras. Para encontrar estos valores se van a analizar el primer y tercer cuartil de cada variable seleccionada (ya que entre esos dos valores o rango intercuartílico, está el 50% de todos los valores obtenidos en el estudio). Así:

Variable	Q1	Q3	IRQ=Q3-Q1	US:Q3+1.5*IQR	UI:Q-1.5*IQR
Perimeter_worst	84,11	125,40	41,29	187,335	22,175

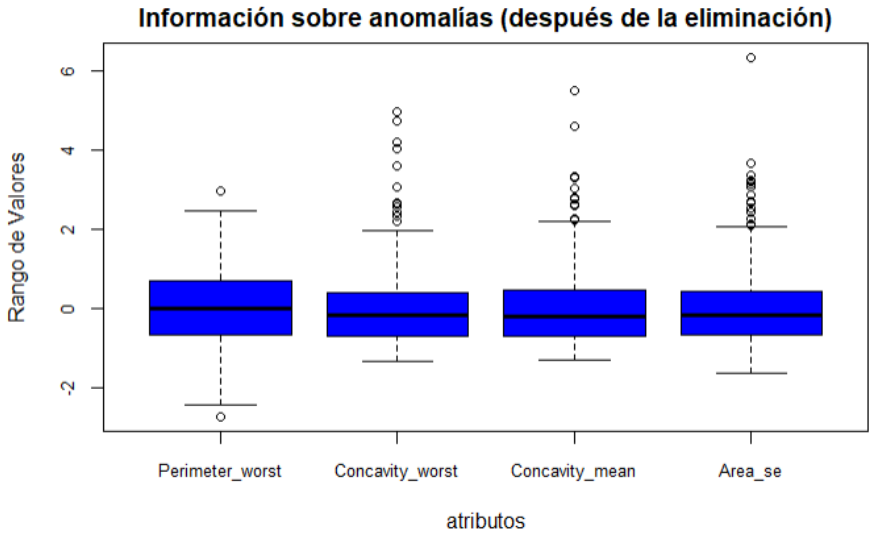
Concavity_mean	0,02956	0,1307	0,10114	0,24281	-0,12215
Area_se	17,85	45,19	27,34	86,2	-23,16
Concavity_worst	0,1145	0,3829	0,2684	0,7855	-0,2881

Por tanto, los valores que estén fuera de los rangos anteriores (UI..US) se modificarán para ser sustituidos por la **media** de los datos de las mismas características. Los valores atípicos que estén dentro del rango, se considerarán valores válidos. Así:

```
data_selected_B <- filter(data_selected, data_selected$Diagnosis == 'B')
boxplot(scale(data_selected_B[2:5]), xlab ="atributos", cex.axis=0.8, yla
b = "Rango de Valores", col="blue", main="Información sobre anomalías (an
tes de la eliminación)")
```




```
for (i in 1:nrow(data_selected)) {
  if (data_selected[i,1] == 'B') {
    if (data_selected[i,2] < 22.175 | data_selected[i,2] > 187.335)
      data_selected[i,2] <- mean(data_selected_B$Perimeter_worst)
    if (data_selected[i,4] < -0.12215 | data_selected[i,4] > 0.24281)
      data_selected[i,4] <- mean(data_selected_B$Concavity_mean)
    if (data_selected[i,5] < -23.16 | data_selected[i,5] > 86.2)
      data_selected[i,5] <- mean(data_selected_B$Area_se)
    if (data_selected[i,3] < -0.2881 | data_selected[i,3] > 0.7855)
      data_selected[i,3] <- mean(data_selected_B$Concavity_worst)
  }
}
data_selected_B <- filter(data_selected, data_selected$Diagnosis == 'B')
boxplot(scale(data_selected_B[2:5]), xlab ="atributos", cex.axis=0.8, yla
b = "Rango de Valores", col="blue", main="Información sobre anomalías (de
spués de la eliminación)")
```



Una vez eliminados los datos **outliers** de las muestras de tumores benignos, se genera el archivo de salida con los datos a analizar:

```
data_to_analyze <- data_selected
# se genera el archivo de salida
write.csv(data_to_analyze, "TCVD_CancerWisconsi.csv")
```

A continuación se van a utilizar diferentes métodos de clasificación usando los atributos indicados anteriormente del conjunto de datos y se evaluará la calidad de los método usados mediante Validación Cruzada o K-fold Cross Validation.

Inicialmente se van a crear los grupos de entrenamiento y test, el primero con 2/3 de los datos y el segundo con 1/3 de los datos originales:

```
if (!require('caTools')) install.packages('caTools'); library('caTools')
# se dividen en grupo de entrenamiento y grupo de test
set.seed(123)
split <- sample.split(data_to_analyze$Diagnosis, SplitRatio = 0.75)
training_set <- subset(data_to_analyze, split == TRUE)
test_set <- subset(data_to_analyze, split == FALSE)
dim(training_set)

## [1] 427   5

dim(test_set)

## [1] 142   5
```

La distribución de muestras malignas y benignas debe ser simétrica en ambos conjuntos:

```
cat("Porcentaje de muestras benignas y malignas en el conjunto de entrena
miento: ")

## Porcentaje de muestras benignas y malignas en el training set:

prop.table(table(training_set$Diagnosis))
```

```
##         B         M
## 0.6276347 0.3723653

cat("\nPorcentaje de muestras benignas y malignas en el conjunto de test:
")
## Porcentaje de muestras benignas y malignas en el test set:

prop.table(table(test_set$Diagnosis))

##
##         B         M
## 0.6267606 0.3732394
```

Modelo de Regresión Logística

La **Regresión Logística** calcula las probabilidades de ocurrencia de alguna de las clases del modelo a partir del uso de la función logística. No requiere de ciertas condiciones como linealidad, normalidad y homocedasticidad. Inicialmente se entrena el modelo y se observan los datos estadísticos:

```
if (!require('stats')) install.packages('stats'); library('stats')

clasificadorRL <- glm(as.factor(Diagnosis) ~ ., family = binomial, data =
training_set)
```

Una vez que el clasificador está entrenado, se puede usar para predecir el resultado. Ya que la regresión logística ofrece como resultado las probabilidades de ocurrencia de cada clase, se va a tomar como umbral el valor de 0.5, de modo que cualquier valor por encima de esa probabilidad se tome como 1 "tumor benigno", y cualquier valor por debajo como 0 como "tumor maligno":

```
if (!require('caret')) install.packages('caret'); library('caret')
pred_train_RL <- predict(clasificadorRL, type = 'response', ndata = train
ing_set)
pred_train_RL <- ifelse(pred_train_RL > 0.5, 1, 0)
pred_train_RL <- factor(pred_train_RL, levels = c("0", "1"), labels = c("
B", "M"))
```

Analizando la matriz de confusión para evaluar la calidad de la predicción:

```
matrizConfusion_train_RL <- confusionMatrix(data=pred_train_RL, reference
=training_set$Diagnosis, positive = "B", dnn = c('Predicción', 'Realidad'
))
matrizConfusion_train_RL

## Confusion Matrix and Statistics
##           Realidad
## Predicción   B   M
##          B 259  10
##          M   9 149
##                Accuracy : 0.9555
##                  95% CI : (0.9314, 0.973)
##     No Information Rate : 0.6276
##     P-Value [Acc > NIR] : <2e-16
```

```
##                  Kappa : 0.9047
##  Mcnemar's Test P-Value : 1
##             Sensitivity : 0.9664
##             Specificity : 0.9371
##          Pos Pred Value : 0.9628
##          Neg Pred Value : 0.9430
##              Prevalence : 0.6276
##          Detection Rate : 0.6066
##    Detection Prevalence : 0.6300
##       Balanced Accuracy : 0.9518
##        'Positive' Class : B
```

Se observa que de 427 mujeres del grupo de entrenamiento, con el modelo usado se han clasificado correctamente 408 (259 verdaderos positivos y 149 verdaderos negativos). Asimismo hay 9 falsos negativos (es decir, 9 mujeres que no tenían cáncer de mama que se predijeron que lo tenían) y 10 falsos positivos (es decir, 10 mujeres que tenían cáncer de mama y que se predijeron que no lo tenían). El clasificador tiene una precisión de 95% (mayor al 62% que es el número de mujeres sin cáncer de mama). Por otro lado, el clasificador tiene una sensibilidad del 97% (verdaderos positivos identificados).

Si hacemos la misma prueba sobre el grupo de test:

```
if (!require('caret')) install.packages('caret'); library('caret')
pred_test_RL <- predict(clasificadorRL, type = 'response', newdata = test
_set)
pred_test_RL <- ifelse(pred_test_RL > 0.5, 1, 0)
pred_test_RL <- factor(pred_test_RL, levels = c("0", "1"), labels = c("B"
, "M"))
matrizConfusion_test_RL <- confusionMatrix(data=pred_test_RL, reference=t
est_set$Diagnosis, positive = "B", dnn = c('Predicción', 'Realidad'))

matrizConfusion_test_RL
```

```
## Confusion Matrix and Statistics
##           Realidad
## Predicción  B  M
##          B 86  5
##          M  3 48
##                Accuracy : 0.9437
##                  95% CI : (0.892, 0.9754)
##     No Information Rate : 0.6268
##     P-Value [Acc > NIR] : <2e-16
##                   Kappa : 0.8787
##  Mcnemar's Test P-Value : 0.7237
##             Sensitivity : 0.9663
##             Specificity : 0.9057
##          Pos Pred Value : 0.9451
##          Neg Pred Value : 0.9412
##              Prevalence : 0.6268
##          Detection Rate : 0.6056
##    Detection Prevalence : 0.6408
##       Balanced Accuracy : 0.9360
##        'Positive' Class : B
```

Se observa que de 142 mujeres del grupo de test, con el modelo usado se han clasificado correctamente 134 (86 verdaderos positivos y 48 verdaderos negativos). Asimismo hay 3 falsos negativos (es decir, 3 mujeres que no tenían cáncer de mama que se predijeron que lo tenían) y 5 falsos positivos (es decir, 5 mujeres que tenían cáncer de mama y que se predijeron que no lo tenían). El clasificador tienen una precisión de 94% (mayor al 62% que es el número de mujeres sin cáncer de mama). Por otro lado, el clasificador tiene una sensibilidad del 97% (verdaderos positivos identificados).

Modelo Naive Bayes

La clasificacion **naive bayes** se basa en el teorema de **bayes** para clasificar. Inicialmente se entrena el modelo y se observan los datos estadísticos:

```
if (!require('e1071')) install.packages('e1071'); library('e1071')
clasificadorBayes <- naiveBayes(Diagnosis ~ ., data = training_set)
```

Analizando el modelo sobre el conjunto de entrenamiento:

```
pred_train_bayes <- predict(clasificadorBayes, newdata = training_set)
matrizConfusion_bayes <- confusionMatrix(data=pred_train_bayes, reference
=training_set$Diagnosis, positive = "B", dnn = c('Predicción', 'Realidad'
))
matrizConfusion_bayes
```

```
## Confusion Matrix and Statistics
##           Realidad
## Predicción   B   M
##          B 258  16
##          M  10 143
##                Accuracy : 0.9391
##                  95% CI : (0.9121, 0.9598)
##     No Information Rate : 0.6276
##     P-Value [Acc > NIR] : <2e-16
##                   Kappa : 0.8687
##  Mcnemar's Test P-Value : 0.3268
##             Sensitivity : 0.9627
##             Specificity : 0.8994
##          Pos Pred Value : 0.9416
##          Neg Pred Value : 0.9346
##              Prevalence : 0.6276
##          Detection Rate : 0.6042
##    Detection Prevalence : 0.6417
##       Balanced Accuracy : 0.9310
##        'Positive' Class : B
```

Se observa que de 427 mujeres del grupo de entrenamiento, con el modelo usado se han clasificado correctamente a 401. El clasificador tiene una precisión del 94% (mayor al 62% que es el número de mujeres sin cáncer de mama). Por otro lado, el clasificador tiene una sensibilidad del 96% (verdaderos positivos identificados).

Si hacemos la misma prueba sobre el grupo de test:

```
pred_test_bayes <- predict(clasificadorBayes, newdata = test_set)
matrizConfusion_bayes <- confusionMatrix(data=pred_test_bayes, reference=
```

```
test_set$Diagnosis, positive = "B", dnn = c('Predicción', 'Realidad'))
matrizConfusion_bayes
```

```
## Confusion Matrix and Statistics
##           Realidad
## Predicción  B  M
##          B 82  6
##          M  7 47
##                Accuracy : 0.9085
##                  95% CI : (0.8485, 0.9503)
##     No Information Rate : 0.6268
##     P-Value [Acc > NIR] : 1.905e-14
##                   Kappa : 0.8051
##  Mcnemar's Test P-Value : 1
##             Sensitivity : 0.9213
##             Specificity : 0.8868
##          Pos Pred Value : 0.9318
##          Neg Pred Value : 0.8704
##              Prevalence : 0.6268
##          Detection Rate : 0.5775
##    Detection Prevalence : 0.6197
##       Balanced Accuracy : 0.9041
##        'Positive' Class : B
```

Se observa que de 142 mujeres del grupo de entrenamiento, con el modelo usado se han clasificado correctamente a 129. Ha habido 6 falsos positivos y 7 falsos negativos. El clasificador tienen una precisión de 91% (mayor al 62% que es el número de mujeres sin cáncer de mama). Por otro lado, el clasificador tiene una sensibilidad del 92% (verdaderos positivos identificados).

Modelo de Árbol de Decisión

La clasificacion **Decision Tree** se basa en reglas lógicas a partir de los datos de entrada. Inicialmente se entrena el modelo y se observan los datos estadísticos:

```
if (!require('rpart')) install.packages('rpart'); library('rpart')
clasificadorDT  <- rpart(Diagnosis ~ ., data = training_set)
```

Analizando el modelo sobre el conjunto de entrenamiento:

```
pred_train_tree <- predict(clasificadorDT, newdata = training_set)
pred_df <- data.frame(pred_train_tree)
fact = cut(pred_df$B,2,labels=c("M","B"))
matrizConfusion_tree <- confusionMatrix(data=fact, reference=training_set
$Diagnosis, positive = "B", dnn = c('Predicción', 'Realidad'))
matrizConfusion_tree
```

```
## Confusion Matrix and Statistics
##           Realidad
## Predicción   B   M
##          B 258   9
##          M  10 150
##                Accuracy : 0.9555
##                  95% CI : (0.9314, 0.973)
```

```
##     No Information Rate : 0.6276
##     P-Value [Acc > NIR] : <2e-16
##                   Kappa : 0.9049
##  Mcnemar's Test P-Value : 1
##             Sensitivity : 0.9627
##             Specificity : 0.9434
##          Pos Pred Value : 0.9663
##          Neg Pred Value : 0.9375
##              Prevalence : 0.6276
##          Detection Rate : 0.6042
##    Detection Prevalence : 0.6253
##       Balanced Accuracy : 0.9530
##        'Positive' Class : B
```

Se observa que de 427 mujeres del grupo de entrenamiento, con el modelo usado se han clasificado correctamente a 408. El clasificador tiene una precisión de 95% (mayor al 62% que es el número de mujeres sin cáncer de mama). Por otro lado, el clasificador tiene una sensibilidad del 96% (verdaderos positivos identificados).

Si se hace la misma prueba sobre el grupo de test:

```
pred_test_tree <- predict(clasificadorDT, newdata = test_set)
pred_df <- data.frame(pred_test_tree)
fact = cut(pred_df$B,2,labels=c("M","B"))
matrizConfusion_tree <- confusionMatrix(data=fact, reference=test_set$Dia
gnosis, positive = "B", dnn = c('Predicción', 'Realidad'))
matrizConfusion_tree
```

```
## Confusion Matrix and Statistics
##           Realidad
## Predicción  B  M
##          B 78  4
##          M 11 49
##                Accuracy : 0.8944
##                  95% CI : (0.8318, 0.9397)
##     No Information Rate : 0.6268
##     P-Value [Acc > NIR] : 5.498e-13
##                   Kappa : 0.7801
##  Mcnemar's Test P-Value : 0.1213
##             Sensitivity : 0.8764
##             Specificity : 0.9245
##          Pos Pred Value : 0.9512
##          Neg Pred Value : 0.8167
##              Prevalence : 0.6268
##          Detection Rate : 0.5493
##    Detection Prevalence : 0.5775
##       Balanced Accuracy : 0.9005
##        'Positive' Class : B
```

Se observa que de 142 mujeres del grupo de entrenamiento, con el modelo usado se han clasificado correctamente a 127. Ha habido 4 falsos positivos y 11 falsos negativos. El clasificador tienen una precisión de 89% (mayor al 62% que es el número de mujeres sin cáncer

de mama). Por otro lado, el clasificador tiene una sensibilidad del 88% (verdaderos positivos identificados).

Modelo Random Forests

La clasificación **Random Forests** construye al azar una gran cantidad de árboles de decisión (**ntree** en la función usada) sobre un mismo conjunto de datos, y la decisión final de la clasificación es tomada a partir de calcular el voto de la mayoría de las predicciones ofrecidas por cada uno de los árboles que conforman el bosque.

```
if (!require('randomForest')) install.packages('randomForest'); library('
randomForest')
clasificadorRF <- randomForest(Diagnosis ~ ., data = training_set, ntree
= 250)
```

Analizando el modelo sobre el conjunto de entrenamiento:

```
pred_train_RF <- predict(clasificadorRF, newdata = training_set)
matrizConfusion_train_RF <- confusionMatrix(data=pred_train_RF, reference
=training_set$Diagnosis, positive = "B", dnn = c('Predicción', 'Realidad'
))
matrizConfusion_train_RF
```

```
## Confusion Matrix and Statistics
##           Realidad
## Predicción   B   M
##          B 268   0
##          M   0 159
##                Accuracy : 1
##                  95% CI : (0.9914, 1)
##     No Information Rate : 0.6276
##     P-Value [Acc > NIR] : < 2.2e-16
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6276
##          Detection Rate : 0.6276
##    Detection Prevalence : 0.6276
##       Balanced Accuracy : 1.0000
##        'Positive' Class : B
```

Se observa que de 427 mujeres del grupo de entrenamiento, con el modelo usado se han clasificado correctamente todas ellas. El clasificador tienen una precisión de 100% (mayor al 62% que es el número de mujeres sin cáncer de mama). Por otro lado, el clasificador tiene una sensibilidad del 100% (verdaderos positivos identificados).

Si hacemos la misma prueba sobre el grupo de test:

```
pred_test_RF <- predict(clasificadorRF, newdata = test_set)
matrizConfusion_test_RF <- confusionMatrix(data=pred_test_RF, reference=t
```

```
est_set$Diagnosis, positive = "B", dnn = c('Predicción', 'Realidad'))
matrizConfusion_test_RF
```

```
## Confusion Matrix and Statistics
##           Realidad
## Predicción  B  M
##          B 81  5
##          M  8 48
##                Accuracy : 0.9085
##                  95% CI : (0.8485, 0.9503)
##     No Information Rate : 0.6268
##     P-Value [Acc > NIR] : 1.905e-14
##                   Kappa : 0.8065
##  Mcnemar's Test P-Value : 0.5791
##             Sensitivity : 0.9101
##             Specificity : 0.9057
##          Pos Pred Value : 0.9419
##          Neg Pred Value : 0.8571
##              Prevalence : 0.6268
##          Detection Rate : 0.5704
##    Detection Prevalence : 0.6056
##       Balanced Accuracy : 0.9079
##        'Positive' Class : B
```

Se observa que de 142 mujeres del grupo de test, con el modelo usado se han clasificado correctamente 129 (81 verdaderos positivos y 48 verdaderos negativos). Asimismo hay 8 falsos negativos (es decir, 8 mujeres que no tenían cáncer de mama que se predijeron que lo tenían) y 5 falsos positivos (es decir, 5 mujeres que tenían cáncer de mama y que se predijeron que no lo tenían). El clasificador tienen una precisión de 91% (mayor al 62% que es el número de mujeres sin cáncer de mama). Por otro lado, el clasificador tiene una sensibilidad del 91% (verdaderos positivos identificados).

Comparación de los modelos usando K-Fold Cross Validation

A continuación se comparan los resultados de los modelos anteriores usando K-Fold Cross Validation. Se van a usar 10 subconjuntos:

```
if (!require('caret')) install.packages('caret'); library('caret')
folds <- createFolds(training_set$Diagnosis, k = 10)

if (!require('class')) install.packages('class'); library('class')
if (!require('rpart')) install.packages('rpart'); library('rpart')
if (!require('randomForest')) install.packages('randomForest'); library('
randomForest')
# Regresion Logistica
cvRegresionLogistica <- lapply(folds, function(x) {
    training_fold <- training_set[-x, ]
    test_fold <- training_set[x, ]
    clasificadorRL <- glm(Diagnosis ~ ., family = binomial, data = traini
ng_fold)
    y_pred <- predict(clasificadorRL, type = 'response', newdata = test_f
old)
    y_pred <- ifelse(y_pred > 0.5, 1, 0)
    y_pred <- factor(y_pred, levels = c("0", "1"), labels = c("B", "M"))
```

```
    cm <- table(test_fold$Diagnosis, y_pred)
    precision <- (cm[1,1] + cm[2,2]) / (cm[1,1] + cm[2,2] +cm[1,2] + cm[2
,1])
    return(precision)
})
precisionRegresionLogistica <- mean(as.numeric(cvRegresionLogistica))
# Naive-Bayes
cvNaiveBayes <- lapply(folds, function(x){
  training_fold <- training_set[-x, ]
  test_fold <- training_set[x, ]
  clasificadorNB <- naiveBayes(Diagnosis ~ ., data = training_fold)
  y_pred <- predict(clasificadorNB, newdata = test_fold)
  cm <- table(test_fold$Diagnosis, y_pred)
  precision <- (cm[1,1] + cm[2,2]) / (cm[1,1] + cm[2,2] +cm[1,2] + cm[2,1
])
  return(precision)
})
precisionNaiveBayes <- mean(as.numeric(cvNaiveBayes))
# Árbol de decisión
cvDecisionTree <- lapply(folds, function(x){
  training_fold <- training_set[-x, ]
  test_fold <- training_set[x, ]
  clasificadorDT <- rpart(Diagnosis ~ ., data = training_fold)
  y_pred <- predict(clasificadorDT, newdata = test_fold, type = 'class')
  cm <- table(test_fold$Diagnosis, y_pred)
  precision <- (cm[1,1] + cm[2,2]) / (cm[1,1] + cm[2,2] +cm[1,2] + cm[2,1
])
  return(precision)
})
precisionDecisionTree <- mean(as.numeric(cvDecisionTree))
# Random Forest
cvRandomForest <- lapply(folds, function(x){
  training_fold <- training_set[-x, ]
  test_fold <- training_set[x, ]
  clasificadorRF <- randomForest(Diagnosis ~ ., data = training_fold, ntr
ee = 250)
  y_pred <- predict(clasificadorRF, newdata = test_fold)
  cm <- table(test_fold$Diagnosis, y_pred)
  precision <- (cm[1,1] + cm[2,2]) / (cm[1,1] + cm[2,2] +cm[1,2] + cm[2,1
])
  return(precision)
})
precisionRandomForest <- mean(as.numeric(cvRandomForest))

precisionRegresionLogistica
## [1] 0.9506091
precisionNaiveBayes
## [1] 0.9414175
precisionDecisionTree
## [1] 0.9274086
precisionRandomForest
## [1] 0.9460133
```

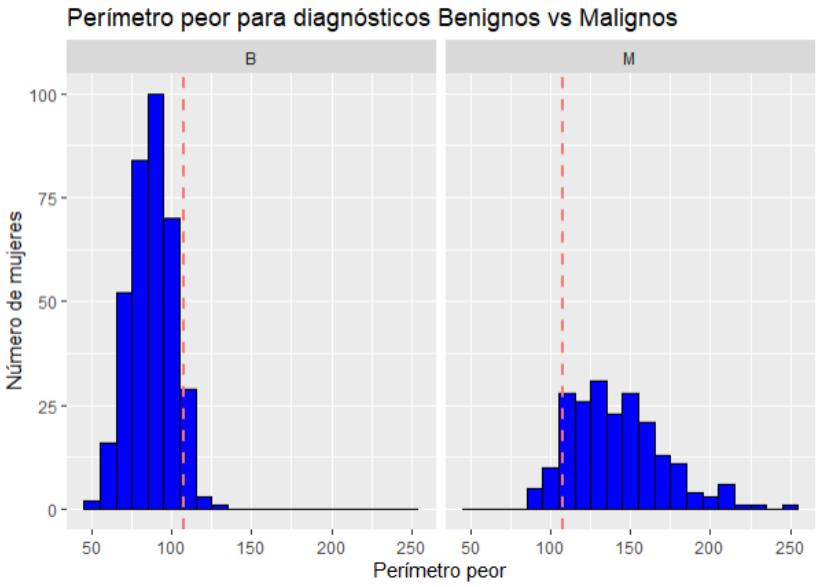
Observándose que el mejor clasificador es el modelo de **Regresión Logística** con un 95% de precisión (mayor que en la prueba previa), frente a un 93% del modelo **Árbol de decisión**.

Representación de los resultados

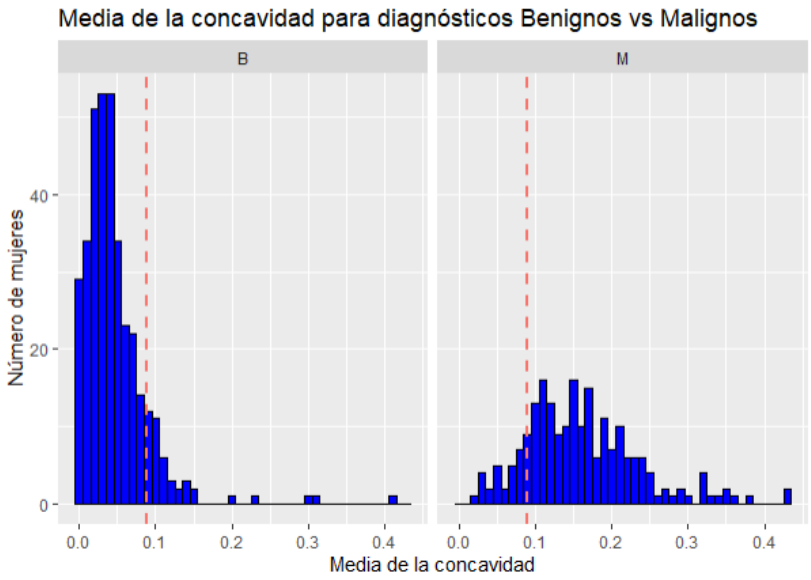
A continuación se van a presentar distintos gráficos estadísticos que demuestran las pruebas estadísticas realizadas. Por un lado se comprobará que los datos seleccionados y usados en los clasificadores tienen calidad suficiente para detectar, a partir de una muestra, si un tumor en la mama es benigno o maligno y por otro se comprobará si los **outliers** de las variables seleccionadas coinciden con muestras de tumores malignos.

Así, para comprobar la calidad de los atributos seleccionados se va a presentar el histograma de los mismos junto con la media de cada uno. En las gráficas se puede comprobar que las mujeres con tumores benignos tienen resultados mayoritariamente por debajo de la media mientras que las que tienen tumores malignos están mayoritariamente por encima de la media:

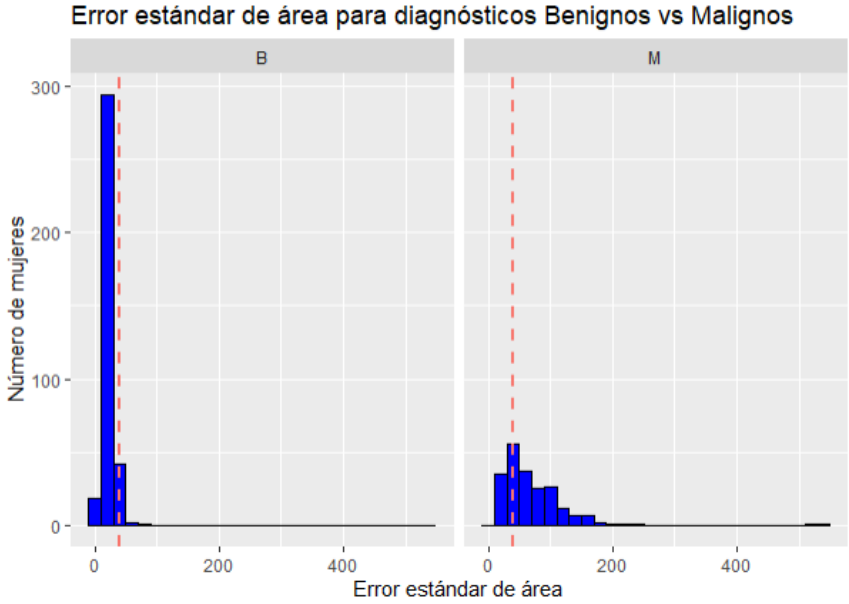
```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')

ggplot(breast_data, aes(x = Perimeter_worst)) + geom_histogram(binwidth =
10, col='black', fill='blue') + facet_wrap(~ Diagnosis)+ggtitle("Perímetr
o peor para diagnósticos Benignos vs Malignos") +labs(x="Perímetro peor",
y="Número de mujeres") +
  geom_vline(aes(xintercept = mean(Perimeter_worst), colour="media"),
             linetype = "dashed", size = 1)
```



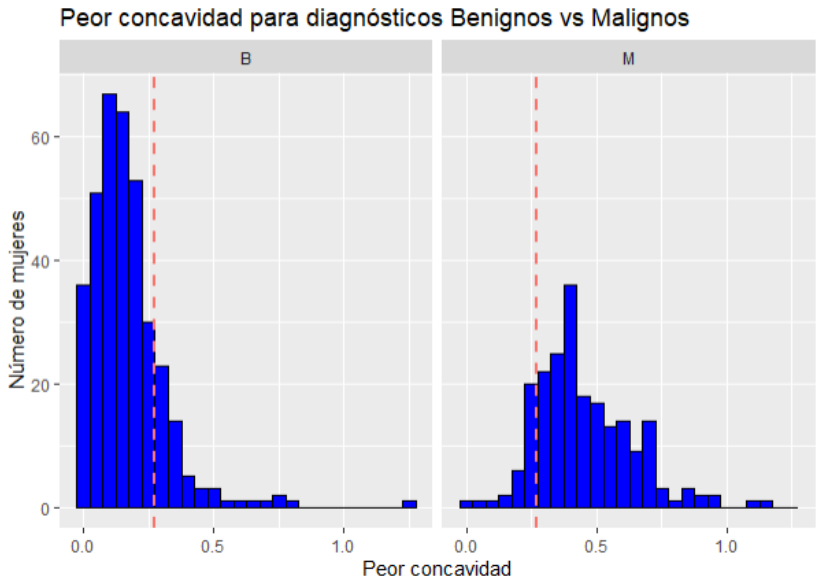

```
ggplot(breast_data, aes(x = Concavity_mean)) + geom_histogram(binwidth =
0.01, col='black', fill='blue') + facet_wrap(~ Diagnosis)+ggtitle("Media
de la concavidad para diagnósticos Benignos vs Malignos") +labs(x="Media
de la concavidad", y="Número de mujeres") +
  geom_vline(aes(xintercept = mean(Concavity_mean), colour="media"),
             linetype = "dashed", size = 1)
```




```
ggplot(breast_data, aes(x = Area_se)) + geom_histogram(binwidth = 20, col
='black', fill='blue') + facet_wrap(~ Diagnosis)+ggtitle("Error estándar
de área para diagnósticos Benignos vs Malignos") +labs(x="Error estándar
de área", y="Número de mujeres") +
  geom_vline(aes(xintercept = mean(Area_se), colour="media"),
             linetype = "dashed", size = 1)
```



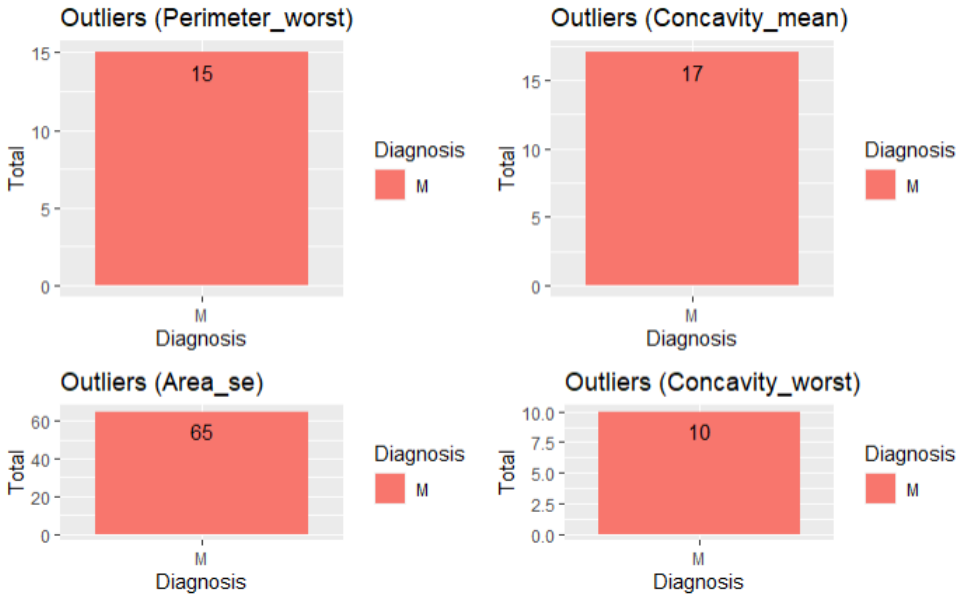

```
ggplot(breast_data, aes(x = Concavity_worst)) + geom_histogram(binwidth =
0.05, col='black', fill='blue') + facet_wrap(~ Diagnosis)+ggtitle("Peor c
oncavidad para diagnósticos Benignos vs Malignos") +labs(x="Peor concavid
ad", y="Número de mujeres") +
  geom_vline(aes(xintercept = mean(Concavity_worst), colour="media"),
             linetype = "dashed",size = 1)
```



Por otro lado, se va a comprobar que los **outliers** detectados en las variables analizadas corresponden con muestras de tumores malignos:

```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
if (!require('grid')) install.packages('grid'); library('grid')
if (!require('gridExtra')) install.packages('gridExtra'); library('gridEx
tra')
grid.newpage()
out_perimeter_worst <- boxplot.stats(data_to_analyze$Perimeter_worst)$out
out_perimeter_worst_ind <- which(data_to_analyze$Perimeter_worst %in% out
_perimeter_worst)
out_perimeter_diagnosis <- data_to_analyze[out_perimeter_worst_ind,]
datos_pw <- out_perimeter_diagnosis %>% group_by(Diagnosis) %>% summarise
(Total=n())
plot1 <- ggplot(datos_pw, aes(x=Diagnosis,  y=Total, fill=Diagnosis))+ gg
title("Outliers (Perimeter_worst)") +
  geom_bar(stat='identity') +
  geom_text(aes(label=Total), vjust=1.6, color="black", position = positi
on_dodge(0.9),  size=4.0)
out_Concavity_mean <- boxplot.stats(data_to_analyze$Concavity_mean)$out
out_Concavity_mean_ind <- which(data_to_analyze$Concavity_mean %in% out_C
oncavity_mean)
out_Concavity_mean_diagnosis <- data_to_analyze[out_Concavity_mean_ind,]
datos_cm <- out_Concavity_mean_diagnosis %>% group_by(Diagnosis) %>% summ
arise(Total=n())
plot2 <- ggplot(datos_cm, aes(x=Diagnosis,  y=Total, fill=Diagnosis))+ gg
title("Outliers (Concavity_mean)") +
  geom_bar(stat='identity') +
  geom_text(aes(label=Total), vjust=1.6, color="black", position = positi
on_dodge(0.9),  size=4.0)
out_Area_se <- boxplot.stats(data_to_analyze$Area_se)$out
out_Area_se_ind <- which(data_to_analyze$Area_se %in% out_Area_se)
out_Area_se_diagnosis <- data_to_analyze[out_Area_se_ind,]
```

```
datos_ase <- out_Area_se_diagnosis %>% group_by(Diagnosis) %>% summarise(
Total=n())
plot3 <- ggplot(datos_ase, aes(x=Diagnosis,  y=Total, fill=Diagnosis))+ g
gtitle("Outliers (Area_se)") +
  geom_bar(stat='identity') +
  geom_text(aes(label=Total), vjust=1.6, color="black", position = positi
on_dodge(0.9),  size=4.0)
out_Concavity_worst <- boxplot.stats(data_to_analyze$Concavity_worst)$out
out_Concavity_worst_ind <- which(data_to_analyze$Concavity_worst %in% out
_Concavity_worst)
out_Concavity_worst_diagnosis <- data_to_analyze[out_Concavity_worst_ind,
]
datos_cw <- out_Concavity_worst_diagnosis %>% group_by(Diagnosis) %>% sum
marise(Total=n())
plot4 <- ggplot(datos_cw, aes(x=Diagnosis,  y=Total, fill=Diagnosis))+ gg
title("Outliers (Concavity_worst)") +
  geom_bar(stat='identity') +
  geom_text(aes(label=Total), vjust=1.6, color="black", position = positi
on_dodge(0.9),  size=4.0)
grid.arrange(plot1, plot2, plot3, plot4, ncol=2, heights=3:2)
```



En todos los casos, los **outliers** corresponden a "tumores malignos", una vez que se han modificado aquellos que correspondían con tumores benignos. No es aconsejable eliminar las muestras con los valores **outliers** de tumores malignos (ni sustituirlos por otros valores), ya que pueden ser valores correctos.

Conclusiones

En el trabajo realizado se ha utilizado el **dataset Breast Cancer Wisconsin (Diagnostic) Data Set** de UCI Machine Learning Repository que analiza la incidencia del cáncer de mama en una población de 569 mujeres. Se ha realizado un estudio del conjunto de datos para describir las distintas variables y se ha comprobado que el conjunto está limpio aunque tiene algunos

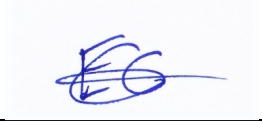
outliers, la mayoría de los cuales se demuestra que pertenecen a muestras de tumores malignos y que, por tanto, se consideran valores correctos.

Por otro lado, se analizó la correlación de los datos del conjunto completo y agrupándolo por tipo de medidas con lo que se eliminaron una gran cantidad de variables que no aportan la suficiente información. Asimismo, se comprobó la normalidad y homocedasticidad, observando que los atributos seleccionados no siguen una distribución normal y 10 de ellos no presentan igualdad de varianza respecto a la variable objetivo.

Asimismo, se hacen dos tipos de pruebas estadísticas: por un lado se analizaron los datos para ver qué atributos tenía más del 50% de correlación con la variable objetivo, ya que estos son los que más influyen en la benignidad o malignidad de un tumor y por otro lado, se analizó si los **outliers** realmente correspondían a medidas correctas. El resultado de estas pruebas se demostró gráficamente en las sección "Representación de los resultados".

Con los atributos seleccionados se realizaron cuatros modelos de clasificación: **Regresión logística**, **Naive-Bayes**, **Árbol de decisión** y **Random Forests**, comprobando su calidad mediante una validación cruzada con 10 **folds**. Finalmente, se deduce que el clasificador del modelo de **Regresión logística** es el que tiene más precisión con un 95% y el modelo **Árbol de decisión** es el que tiene menos con un 93% de precisión.

Contribución

Contribuciones	Firma	
Investigación previa	Eva García y Carmen N. Ojeda	
		
Redacción de las respuestas	Eva García y Carmen N. Ojeda	
		
Desarrollo del código	Eva García y Carmen N. Ojeda