

PRA2 - Tipología y ciclo de los datos

Yésica Fernández Ramos

Carlos Ruiz Salvador

2022-06-06

Antes de empezar con la práctica, vamos a cargar los paquetes necesarios para la realización del ejercicio.

```
# https://cran.r-project.org/web/packages/dplyr/index.html
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
# https://cran.r-project.org/web/packages/ggplot2/index.html
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
```

1. Descripción del dataset

Objetivo

El objetivo de esta práctica es el tratamiento de un dataset. En este caso, vamos a trabajar con un dataset que se encuentra en Kaggle en el siguiente enlace: <https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>

Descripción

El dataset contiene datos de tumores mamarios. Las características del tumor se calculan a partir de una imagen digitalizada de una masa mamaria. Las variables con las que cuenta el dataset son las siguientes:

- **id:** Identificador.
- **diagnosis:** Variable categórica que indica si el tumor es maligno (M) o benigno (B).
- **radius_mean:** Media de las distancias desde el centro a los puntos del perímetro.
- **texture_mean:** Desviación estándar de los valores de la escala de grises.
- **perimeter_mean:** Tamaño medio del tumor central.
- **area_mean:** Media del área del tumor.
- **smoothness_mean:** Media de variación local en longitudes de radio.
- **compactness_mean:** Media del perímetro² / área - 1.0.
- **concavity_mean:** Media de la severidad de las porciones cóncavas del contorno.
- **concave points_mean:** Media del número de porciones cóncavas del contorno.
- **symmetry_mean:** Media de la simetría.
- **fractal_dimension_mean:** Media para “aproximación a la costa” - 1.
- **radius_se:** Error estándar para la media de las distancias del centro a los puntos del perímetro.
- **texture_se:** Error estándar para la desviación estándar de los valores de escala de grises.
- **perimeter_se:** Error estándar para la media del perímetro.
- **area_se:** Error estándar para la media del área.
- **smoothness_se:** Error estándar para la variación local en longitudes de radio.
- **compactness_se:** Error estándar para perímetro² / área - 1.0.
- **concavity_se:** Error estándar para la severidad de las porciones cóncavas del contorno.

- **concave points_se:** Error estándar para el número de porciones cóncavas del contorno.
- **symmetry_se:** Error estándar de la media de la simetría.
- **fractal_dimension_se:** Error estándar para “aproximación de la línea de costa” - 1.
- **radius_worst:** Valor medio “peor” o mayor para la media de las distancias desde el centro hasta los puntos del perímetro.
- **texture_worst:** Valor medio “peor” o mayor para la desviación estándar de los valores de escala de grises.
- **perimeter_worst:** Valor medio “peor” o mayor para el perímetro medio.
- **area_worst:** Valor medio “peor” o mayor para la media del área.
- **smoothness_worst:** Valor medio “peor” o más grande para la variación local en longitudes de radio.
- **compactness_worst:** Valor medio “peor” o mayor para el $\text{perímetro}^2/\text{área} - 1,0$.
- **concavity_worst:** Valor medio “peor” o mayor para la gravedad de las partes cóncavas del contorno.
- **concave points_worst:** Valor medio “peor” o mayor para el número de porciones cóncavas del contorno.
- **symmetry_worst:** Valor medio “peor” o mayor de la media de la simetría.
- **fractal_dimension_worst:** Valor medio “peor” o mayor para “aproximación a la línea de costa” - 1.

El cáncer de mama afecta a millones de mujeres en todo el mundo. Con este dataset se podría desarrollar un algoritmo de predicción de manera que a través de los datos de una imagen digitalizada de la masa mamaria se pueda predecir si el tumor es benigno o maligno. Ayudaría a la diagnosis del cáncer sin pruebas demasiado invasivas.

Realizamos la carga del dataset y vemos un resumen del mismo.

```
cancer <- read.csv("data.csv", sep = ",", stringsAsFactors = TRUE)
summary(cancer)
```

```
##           id           diagnosis radius_mean texture_mean
## Min.      :    8670      B:357      Min.      : 6.981      Min.      : 9.71
## 1st Qu.:   869218      M:212      1st Qu.:11.700      1st Qu.:16.17
## Median :   906024                      Median :13.370      Median :18.84
## Mean      : 30371831                      Mean      :14.127      Mean      :19.29
## 3rd Qu.:   8813129                      3rd Qu.:15.780      3rd Qu.:21.80
## Max.      :911320502                      Max.      :28.110      Max.      :39.28
## perimeter_mean area_mean smoothness_mean compactness_mean
## Min.      : 43.79      Min.      :143.5      Min.      :0.05263      Min.      :0.01938
## 1st Qu.: 75.17      1st Qu.: 420.3      1st Qu.:0.08637      1st Qu.:0.06492
## Median : 86.24      Median : 551.1      Median :0.09587      Median :0.09263
## Mean      : 91.97      Mean      : 654.9      Mean      :0.09636      Mean      :0.10434
## 3rd Qu.:104.10      3rd Qu.: 782.7      3rd Qu.:0.10530      3rd Qu.:0.13040
## Max.      :188.50      Max.      :2501.0      Max.      :0.16340      Max.      :0.34540
## concavity_mean concave.points_mean symmetry_mean fractal_dimension_mean
## Min.      :0.00000      Min.      :0.00000      Min.      :0.1060      Min.      :0.04996
## 1st Qu.:0.02956      1st Qu.:0.02031      1st Qu.:0.1619      1st Qu.:0.05770
## Median :0.06154      Median :0.03350      Median :0.1792      Median :0.06154
## Mean      :0.08880      Mean      :0.04892      Mean      :0.1812      Mean      :0.06280
## 3rd Qu.:0.13070      3rd Qu.:0.07400      3rd Qu.:0.1957      3rd Qu.:0.06612
## Max.      :0.42680      Max.      :0.20120      Max.      :0.3040      Max.      :0.09744
## radius_se texture_se perimeter_se area_se
## Min.      :0.1115      Min.      :0.3602      Min.      : 0.757      Min.      : 6.802
## 1st Qu.:0.2324      1st Qu.:0.8339      1st Qu.: 1.606      1st Qu.:17.850
## Median :0.3242      Median :1.1080      Median : 2.287      Median :24.530
## Mean      :0.4052      Mean      :1.2169      Mean      : 2.866      Mean      :40.337
```

```
## 3rd Qu.:0.4789 3rd Qu.:1.4740 3rd Qu.: 3.357 3rd Qu.: 45.190
## Max. :2.8730 Max. :4.8850 Max. :21.980 Max. :542.200
## smoothness_se compactness_se concavity_se concave.points_se
## Min. :0.001713 Min. :0.002252 Min. :0.00000 Min. :0.000000
## 1st Qu.:0.005169 1st Qu.:0.013080 1st Qu.:0.01509 1st Qu.:0.007638
## Median :0.006380 Median :0.020450 Median :0.02589 Median :0.010930
## Mean :0.007041 Mean :0.025478 Mean :0.03189 Mean :0.011796
## 3rd Qu.:0.008146 3rd Qu.:0.032450 3rd Qu.:0.04205 3rd Qu.:0.014710
## Max. :0.031130 Max. :0.135400 Max. :0.39600 Max. :0.052790
## symmetry_se fractal_dimension_se radius_worst texture_worst
## Min. :0.007882 Min. :0.0008948 Min. : 7.93 Min. :12.02
## 1st Qu.:0.015160 1st Qu.:0.0022480 1st Qu.:13.01 1st Qu.:21.08
## Median :0.018730 Median :0.0031870 Median :14.97 Median :25.41
## Mean :0.020542 Mean :0.0037949 Mean :16.27 Mean :25.68
## 3rd Qu.:0.023480 3rd Qu.:0.0045580 3rd Qu.:18.79 3rd Qu.:29.72
## Max. :0.078950 Max. :0.0298400 Max. :36.04 Max. :49.54
## perimeter_worst area_worst smoothness_worst compactness_worst
## Min. : 50.41 Min. : 185.2 Min. :0.07117 Min. :0.02729
## 1st Qu.: 84.11 1st Qu.: 515.3 1st Qu.:0.11660 1st Qu.:0.14720
## Median : 97.66 Median : 686.5 Median :0.13130 Median :0.21190
## Mean :107.26 Mean : 880.6 Mean :0.13237 Mean :0.25427
## 3rd Qu.:125.40 3rd Qu.:1084.0 3rd Qu.:0.14600 3rd Qu.:0.33910
## Max. :251.20 Max. :4254.0 Max. :0.22260 Max. :1.05800
## concavity_worst concave.points_worst symmetry_worst fractal_dimension_worst
## Min. :0.0000 Min. :0.00000 Min. :0.1565 Min. :0.05504
## 1st Qu.:0.1145 1st Qu.:0.06493 1st Qu.:0.2504 1st Qu.:0.07146
## Median :0.2267 Median :0.09993 Median :0.2822 Median :0.08004
## Mean :0.2722 Mean :0.11461 Mean :0.2901 Mean :0.08395
## 3rd Qu.:0.3829 3rd Qu.:0.16140 3rd Qu.:0.3179 3rd Qu.:0.09208
## Max. :1.2520 Max. :0.29100 Max. :0.6638 Max. :0.20750
## X
## Mode:logical
## NA's:569
##
##
##
##
```

```
head(cancer)
```

```
##          id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1    842302         M      17.99      10.38      122.80     1001.0
## 2    842517         M      20.57      17.77      132.90     1326.0
## 3  84300903         M      19.69      21.25      130.00     1203.0
## 4  84348301         M      11.42      20.38       77.58       386.1
## 5  84358402         M      20.29      14.34      135.10     1297.0
## 6    843786         M      12.45      15.70       82.57       477.1
## smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1          0.11840          0.27760          0.3001          0.14710
## 2          0.08474          0.07864          0.0869          0.07017
## 3          0.10960          0.15990          0.1974          0.12790
## 4          0.14250          0.28390          0.2414          0.10520
## 5          0.10030          0.13280          0.1980          0.10430
## 6          0.12780          0.17000          0.1578          0.08089
```

```

## symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1 0.2419 0.07871 1.0950 0.9053 8.589
## 2 0.1812 0.05667 0.5435 0.7339 3.398
## 3 0.2069 0.05999 0.7456 0.7869 4.585
## 4 0.2597 0.09744 0.4956 1.1560 3.445
## 5 0.1809 0.05883 0.7572 0.7813 5.438
## 6 0.2087 0.07613 0.3345 0.8902 2.217
## area_se smoothness_se compactness_se concavity_se concave.points_se
## 1 153.40 0.006399 0.04904 0.05373 0.01587
## 2 74.08 0.005225 0.01308 0.01860 0.01340
## 3 94.03 0.006150 0.04006 0.03832 0.02058
## 4 27.23 0.009110 0.07458 0.05661 0.01867
## 5 94.44 0.011490 0.02461 0.05688 0.01885
## 6 27.19 0.007510 0.03345 0.03672 0.01137
## symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
## 1 0.03003 0.006193 25.38 17.33 184.60
## 2 0.01389 0.003532 24.99 23.41 158.80
## 3 0.02250 0.004571 23.57 25.53 152.50
## 4 0.05963 0.009208 14.91 26.50 98.87
## 5 0.01756 0.005115 22.54 16.67 152.20
## 6 0.02165 0.005082 15.47 23.75 103.40
## area_worst smoothness_worst compactness_worst concavity_worst
## 1 2019.0 0.1622 0.6656 0.7119
## 2 1956.0 0.1238 0.1866 0.2416
## 3 1709.0 0.1444 0.4245 0.4504
## 4 567.7 0.2098 0.8663 0.6869
## 5 1575.0 0.1374 0.2050 0.4000
## 6 741.6 0.1791 0.5249 0.5355
## concave.points_worst symmetry_worst fractal_dimension_worst X
## 1 0.2654 0.4601 0.11890 NA
## 2 0.1860 0.2750 0.08902 NA
## 3 0.2430 0.3613 0.08758 NA
## 4 0.2575 0.6638 0.17300 NA
## 5 0.1625 0.2364 0.07678 NA
## 6 0.1741 0.3985 0.12440 NA

```

2. Integración y selección de datos.

Para la realización del estudio se van a descartar las variables que van a utilizar todas las variables con las que cuenta el dataset. Después de realizar la limpieza y el análisis, igual se descarta alguna variable para la creación del modelo si se ve que estas no son útiles para el modelo.

3. Limpieza de los datos

3.1. ¿Los datos contienen ceros o elementos vacíos?

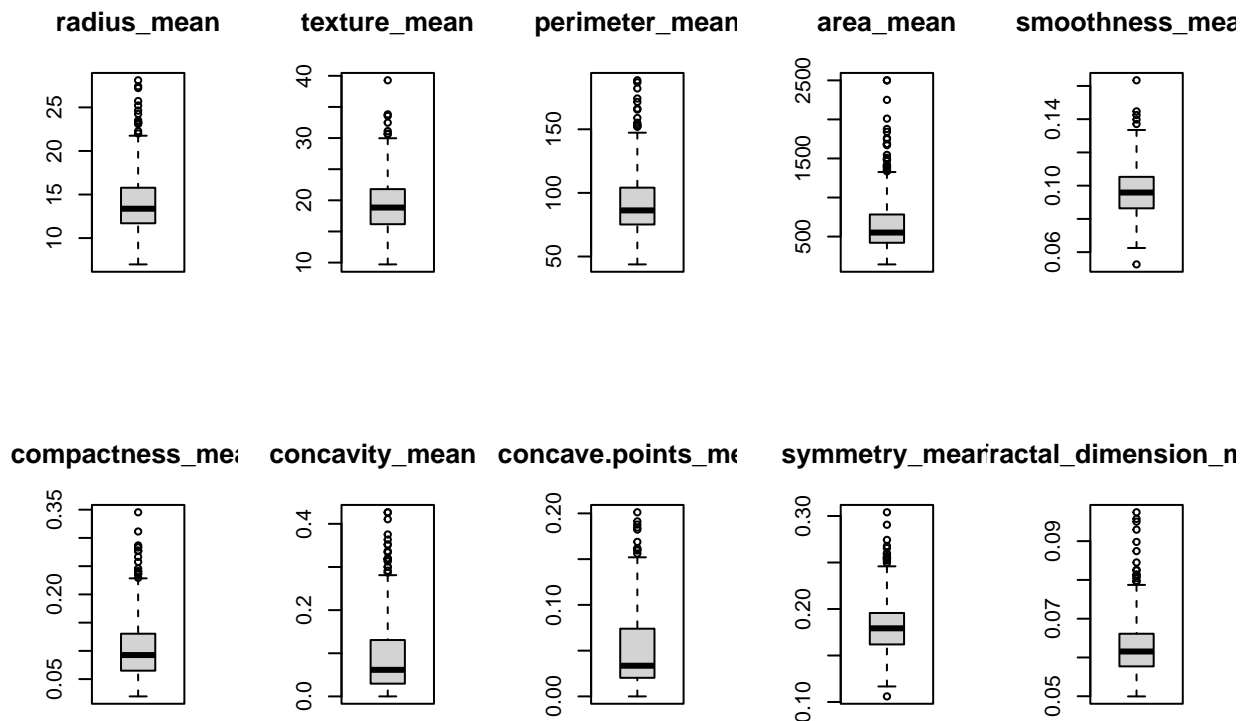
En primer lugar, se va a observar si el dataset contiene NA o valores nulos. En este caso, como se puede observar en el resumen, todas las variables son completas, no contienen NA. Al realizar la carga de datos se ha añadido una columna "X" que contiene NA para todas las observaciones. Por lo tanto, se va a eliminar esta columna.

```
cancer <- cancer[, !names(cancer) %in% c("X")]
```

3.2. Identifica y gestiona los valores extremos.

Se procede a evaluar ahora los outliers de las variables. Para ello, se crearán gráficos de cajas y se estudiarán los valores extremos.

```
par(mfrow = c(2, 5))
g_radius_mean <- boxplot(cancer$radius_mean, main="radius_mean")
g_texture_mean <- boxplot(cancer$texture_mean, main="texture_mean")
g_perimeter_mean <- boxplot(cancer$perimeter_mean, main="perimeter_mean")
g_area_mean <- boxplot(cancer$area_mean, main="area_mean")
g_smoothness_mean <- boxplot(cancer$smoothness_mean, main="smoothness_mean")
g_compactness_mean <- boxplot(cancer$compactness_mean, main="compactness_mean")
g_concavity_mean <- boxplot(cancer$concavity_mean, main="concavity_mean")
g_concave.points_mean <- boxplot(cancer$concave.points_mean, main="concave.points_mean")
g_symmetry_mean <- boxplot(cancer$symmetry_mean, main="symmetry_mean")
g_fractal_dimension_mean <- boxplot(cancer$fractal_dimension_mean, main="fractal_dimension_mean")
```



Al observar estas 10 variables se puede ver que existen outliers para todas ellas, al rededor de 10 valores cada variable, más o menos. Como el conjunto de datos ya cuenta con pocas observaciones se van a sustituir estos valores con la media de las variables. Para ello, primero se sustituyen por NA para que estos valores no influyan a la hora del cálculo de la media.

```

cancer$radius_mean <- ifelse(cancer$radius_mean %in% g_radius_mean$out, NA, cancer$radius_mean)
cancer$texture_mean <- ifelse(cancer$texture_mean %in% g_texture_mean$out, NA, cancer$texture_mean)
cancer$perimeter_mean <- ifelse(cancer$perimeter_mean %in% g_perimeter_mean$out, NA, cancer$perimeter_mean)
cancer$area_mean <- ifelse(cancer$area_mean %in% g_area_mean$out, NA, cancer$area_mean)
cancer$smoothness_mean <- ifelse(cancer$smoothness_mean %in% g_smoothness_mean$out, NA, cancer$smoothness_mean)
cancer$compactness_mean <- ifelse(cancer$compactness_mean %in% g_compactness_mean$out, NA, cancer$compactness_mean)
cancer$concavity_mean <- ifelse(cancer$concavity_mean %in% g_concavity_mean$out, NA, cancer$concavity_mean)
cancer$concave.points_mean <- ifelse(cancer$concave.points_mean %in% g_concave.points_mean$out, NA, cancer$concave.points_mean)
cancer$symmetry_mean <- ifelse(cancer$symmetry_mean %in% g_symmetry_mean$out, NA, cancer$symmetry_mean)
cancer$fractal_dimension_mean <- ifelse(cancer$fractal_dimension_mean %in% g_fractal_dimension_mean$out, NA, cancer$fractal_dimension_mean)

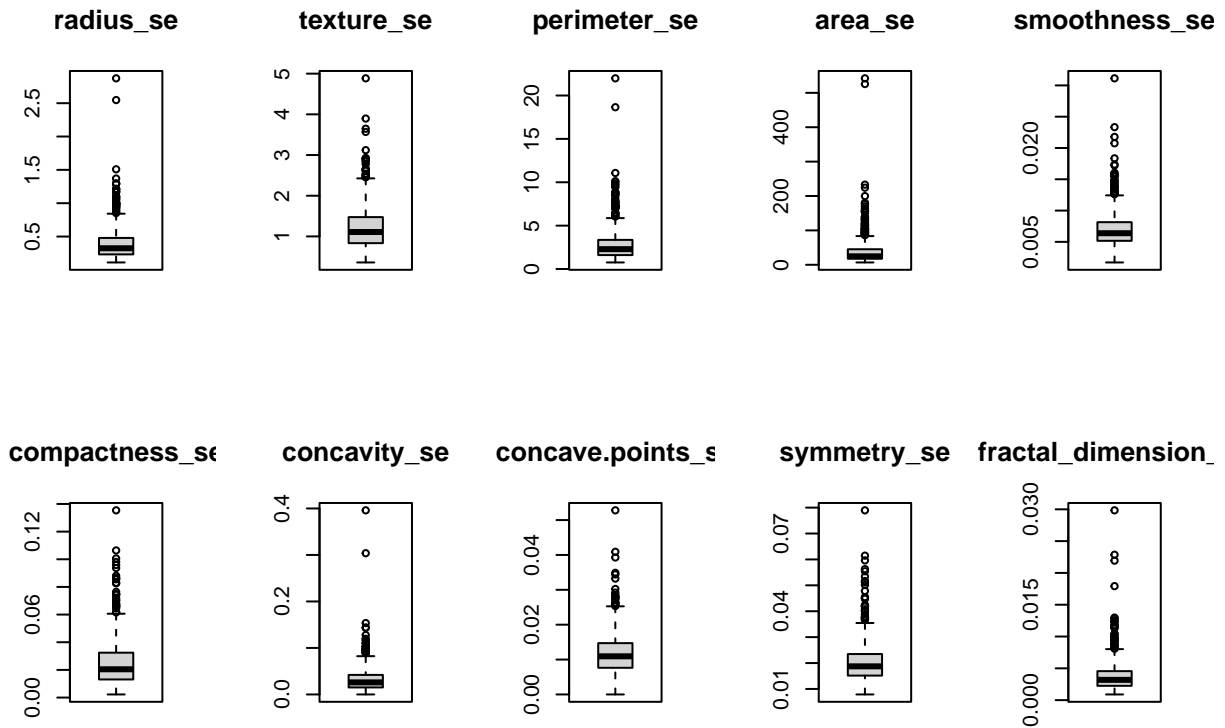
```

Seguimos observando los valores extremos.

```

par(mfrow = c(2, 5))
g_radius_se <- boxplot(cancer$radius_se, main= "radius_se")
g_texture_se <- boxplot(cancer$texture_se, main="texture_se")
g_perimeter_se <- boxplot(cancer$perimeter_se, main="perimeter_se")
g_area_se <- boxplot(cancer$area_se, main = "area_se")
g_smoothness_se <- boxplot(cancer$smoothness_se, main="smoothness_se")
g_compactness_se <- boxplot(cancer$compactness_se, main = "compactness_se")
g_concavity_se <- boxplot(cancer$concavity_se, main = "concavity_se")
g_concave.points_se <- boxplot(cancer$concave.points_se, main="concave.points_se")
g_symmetry_se <- boxplot(cancer$symmetry_se, main="symmetry_se")
g_fractal_dimension_se <- boxplot(cancer$fractal_dimension_se, main="fractal_dimension_se")

```

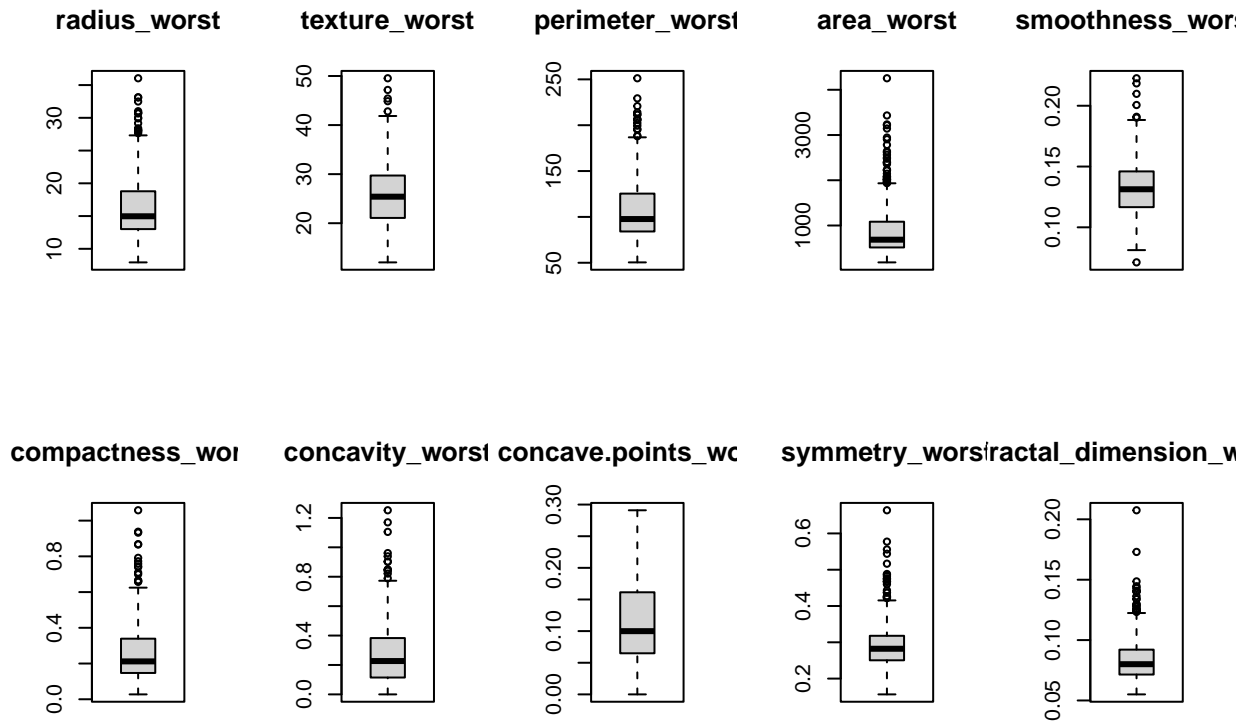


Para estas variables se realiza lo mismo que en el caso anterior, se sustituyen por NA para posteriormente sustituírlos por la media.

```
cancer$radius_se <- ifelse(cancer$radius_se %in% g_radius_se$out, NA, cancer$radius_se)
cancer$texture_se <- ifelse(cancer$texture_se %in% g_texture_se$out, NA, cancer$texture_se)
cancer$perimeter_se <- ifelse(cancer$perimeter_se %in% g_perimeter_se$out, NA, cancer$perimeter_se)
cancer$area_se <- ifelse(cancer$area_se %in% g_area_se$out, NA, cancer$area_se)
cancer$smoothness_se <- ifelse(cancer$smoothness_se %in% g_smoothness_se$out, NA, cancer$smoothness_se)
cancer$compactness_se <- ifelse(cancer$compactness_se %in% g_compactness_se$out, NA, cancer$compactness_se)
cancer$concavity_se <- ifelse(cancer$concavity_se %in% g_concavity_se$out, NA, cancer$concavity_se)
cancer$concave.points_se <- ifelse(cancer$concave.points_se %in% g_concave.points_se$out, NA, cancer$concave.points_se)
cancer$symmetry_se <- ifelse(cancer$symmetry_se %in% g_symmetry_se$out, NA, cancer$symmetry_se)
cancer$fractal_dimension_se <- ifelse(cancer$fractal_dimension_se %in% g_fractal_dimension_se$out, NA, cancer$fractal_dimension_se)
```

Se ven ahora los valores extremos de las últimas variables.

```
par(mfrow = c(2, 5))
g_radius_worst <- boxplot(cancer$radius_worst, main = "radius_worst" )
g_texture_worst <- boxplot(cancer$texture_worst, main="texture_worst")
g_perimeter_worst <- boxplot(cancer$perimeter_worst, main="perimeter_worst")
g_area_worst <- boxplot(cancer$area_worst, main="area_worst")
g_smoothness_worst <- boxplot(cancer$smoothness_worst, main="smoothness_worst")
g_compactness_worst <- boxplot(cancer$compactness_worst, main="compactness_worst")
g_concavity_worst <- boxplot(cancer$concavity_worst, main="concavity_worst")
g_concave.points_worst <- boxplot(cancer$concave.points_worst, main="concave.points_worst")
g_symmetry_worst <- boxplot(cancer$symmetry_worst, main="symmetry_worst")
g_fractal_dimension_worst <- boxplot(cancer$fractal_dimension_worst, main="fractal_dimension_worst")
```



```
cancer$radius_worst <- ifelse(cancer$radius_worst %in% g_radius_worst$out, NA, cancer$radius_worst)
cancer$texture_worst <- ifelse(cancer$texture_worst %in% g_texture_worst$out, NA, cancer$texture_worst)
cancer$perimeter_worst <- ifelse(cancer$perimeter_worst %in% g_perimeter_worst$out, NA, cancer$perimeter_worst)
cancer$area_worst <- ifelse(cancer$area_worst %in% g_area_worst$out, NA, cancer$area_worst)
cancer$smoothness_worst <- ifelse(cancer$smoothness_worst %in% g_smoothness_worst$out, NA, cancer$smoothness_worst)
cancer$compactness_worst <- ifelse(cancer$compactness_worst %in% g_compactness_worst$out, NA, cancer$compactness_worst)
cancer$concavity_worst <- ifelse(cancer$concavity_worst %in% g_concavity_worst$out, NA, cancer$concavity_worst)
cancer$concave.points_worst <- ifelse(cancer$concave.points_worst %in% g_concave.points_worst$out, NA, cancer$concave.points_worst)
cancer$symmetry_worst <- ifelse(cancer$symmetry_worst %in% g_symmetry_worst$out, NA, cancer$symmetry_worst)
cancer$fractal_dimension_worst <- ifelse(cancer$fractal_dimension_worst %in% g_fractal_dimension_worst$out, NA, cancer$fractal_dimension_worst)
```

Ahora podemos sustituir los valores NA por las medias de las variables.

```
cancer$radius_mean <- ifelse(is.na(cancer$radius_mean), mean(na.omit(cancer$radius_mean)), cancer$radius_mean)
cancer$texture_mean <- ifelse(is.na(cancer$texture_mean), mean(na.omit(cancer$texture_mean)), cancer$texture_mean)
cancer$perimeter_mean <- ifelse(is.na(cancer$perimeter_mean), mean(na.omit(cancer$perimeter_mean)), cancer$perimeter_mean)
cancer$area_mean <- ifelse(is.na(cancer$area_mean), mean(na.omit(cancer$area_mean)), cancer$area_mean)
cancer$smoothness_mean <- ifelse(is.na(cancer$smoothness_mean), mean(na.omit(cancer$smoothness_mean)), cancer$smoothness_mean)
cancer$compactness_mean <- ifelse(is.na(cancer$compactness_mean), mean(na.omit(cancer$compactness_mean)), cancer$compactness_mean)
cancer$concavity_mean <- ifelse(is.na(cancer$concavity_mean), mean(na.omit(cancer$concavity_mean)), cancer$concavity_mean)
cancer$concave.points_mean <- ifelse(is.na(cancer$concave.points_mean), mean(na.omit(cancer$concave.points_mean)), cancer$concave.points_mean)
cancer$symmetry_mean <- ifelse(is.na(cancer$symmetry_mean), mean(na.omit(cancer$symmetry_mean)), cancer$symmetry_mean)
cancer$fractal_dimension_mean <- ifelse(is.na(cancer$fractal_dimension_mean), mean(na.omit(cancer$fractal_dimension_mean)), cancer$fractal_dimension_mean)

cancer$radius_se <- ifelse(is.na(cancer$radius_se), mean(na.omit(cancer$radius_se)), cancer$radius_se)
```



```

cancer$texture_se <- ifelse(is.na(cancer$texture_se), mean(na.omit(cancer$texture_se)), cancer$texture_se)
cancer$perimeter_se <- ifelse(is.na(cancer$perimeter_se), mean(na.omit(cancer$perimeter_se)), cancer$perimeter_se)
cancer$area_se <- ifelse(is.na(cancer$area_se), mean(na.omit(cancer$area_se)), cancer$area_se)
cancer$smoothness_se <- ifelse(is.na(cancer$smoothness_se), mean(na.omit(cancer$smoothness_se)), cancer$smoothness_se)
cancer$compactness_se <- ifelse(is.na(cancer$compactness_se), mean(na.omit(cancer$compactness_se)), cancer$compactness_se)
cancer$concavity_se <- ifelse(is.na(cancer$concavity_se), mean(na.omit(cancer$concavity_se)), cancer$concavity_se)
cancer$concave.points_se <- ifelse(is.na(cancer$concave.points_se), mean(na.omit(cancer$concave.points_se)), cancer$concave.points_se)
cancer$symmetry_se <- ifelse(is.na(cancer$symmetry_se), mean(na.omit(cancer$symmetry_se)), cancer$symmetry_se)
cancer$fractal_dimension_se <- ifelse(is.na(cancer$fractal_dimension_se), mean(na.omit(cancer$fractal_dimension_se)), cancer$fractal_dimension_se)

cancer$radius_worst <- ifelse(is.na(cancer$radius_worst), mean(na.omit(cancer$radius_worst)), cancer$radius_worst)
cancer$texture_worst <- ifelse(is.na(cancer$texture_worst), mean(na.omit(cancer$texture_worst)), cancer$texture_worst)
cancer$perimeter_worst <- ifelse(is.na(cancer$perimeter_worst), mean(na.omit(cancer$perimeter_worst)), cancer$perimeter_worst)
cancer$area_worst <- ifelse(is.na(cancer$area_worst), mean(na.omit(cancer$area_worst)), cancer$area_worst)
cancer$smoothness_worst <- ifelse(is.na(cancer$smoothness_worst), mean(na.omit(cancer$smoothness_worst)), cancer$smoothness_worst)
cancer$compactness_worst <- ifelse(is.na(cancer$compactness_worst), mean(na.omit(cancer$compactness_worst)), cancer$compactness_worst)
cancer$concavity_worst <- ifelse(is.na(cancer$concavity_worst), mean(na.omit(cancer$concavity_worst)), cancer$concavity_worst)
cancer$concave.points_worst <- ifelse(is.na(cancer$concave.points_worst), mean(na.omit(cancer$concave.points_worst)), cancer$concave.points_worst)
cancer$symmetry_worst <- ifelse(is.na(cancer$symmetry_worst), mean(na.omit(cancer$symmetry_worst)), cancer$symmetry_worst)
cancer$fractal_dimension_worst <- ifelse(is.na(cancer$fractal_dimension_worst), mean(na.omit(cancer$fractal_dimension_worst)), cancer$fractal_dimension_worst)

```

De esta manera ya está realizado el tratamiento de los valores extremos.

4. Análisis de los datos

4.1. Selección de los grupos de datos que se quieren analizar/comparar

Para empezar el análisis, vamos a ver unos descriptivos básicos de los datos.

```
summary(cancer)
```

```

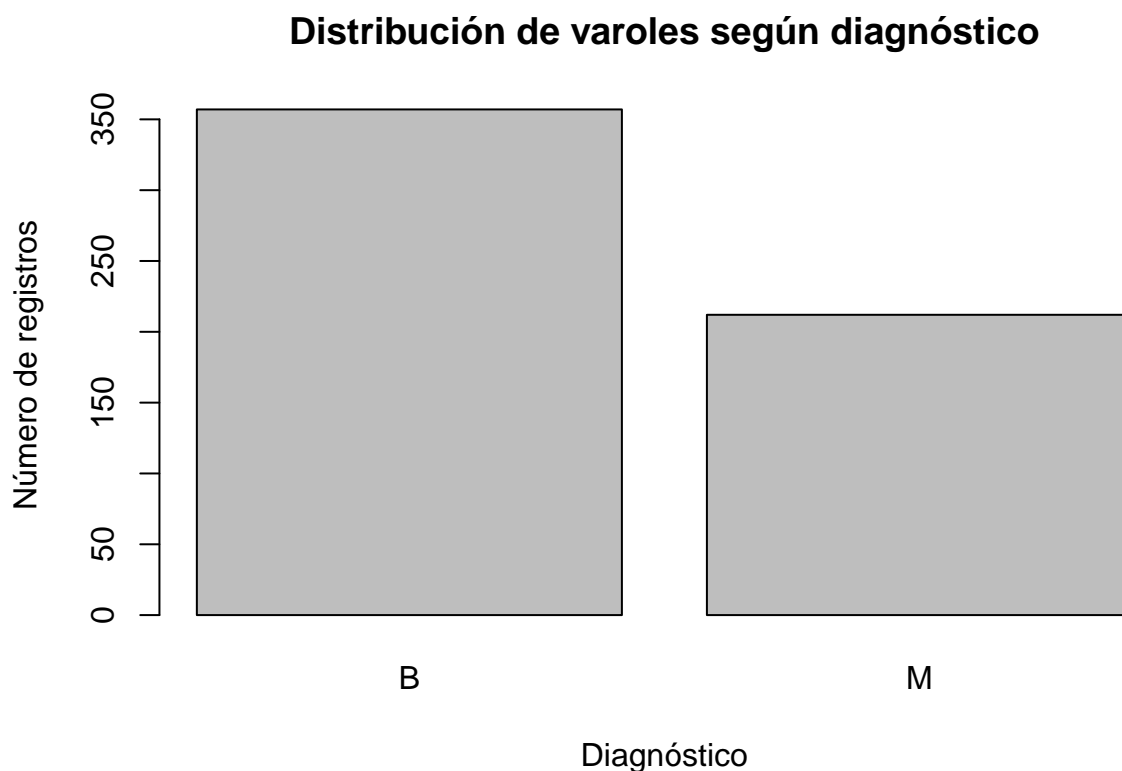
##           id           diagnosis radius_mean texture_mean
## Min.      :    8670      B:357      Min.      : 6.981   Min.      : 9.71
## 1st Qu.:  869218      M:212      1st Qu.:11.700   1st Qu.:16.17
## Median   :  906024                      Median :13.370   Median :18.84
## Mean     : 30371831                      Mean   :13.865   Mean    :19.12
## 3rd Qu.:  8813129                      3rd Qu.:15.460   3rd Qu.:21.60
## Max.     :911320502                      Max.    :21.750   Max.    :29.97
## perimeter_mean area_mean smoothness_mean compactness_mean
## Min.      : 43.79   Min.      :143.5   Min.      :0.06251   Min.      :0.01938
## 1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08641   1st Qu.:0.06492
## Median   : 86.24   Median   :551.1   Median :0.09592   Median :0.09263
## Mean     : 90.24   Mean     :608.2   Mean    :0.09600   Mean    :0.09959
## 3rd Qu.:102.40   3rd Qu.: 710.6   3rd Qu.:0.10490   3rd Qu.:0.12750
## Max.     :147.30   Max.     :1326.0   Max.     :0.13350   Max.     :0.22840
## concavity_mean concave.points_mean symmetry_mean fractal_dimension_mean
## Min.      :0.00000   Min.      :0.00000   Min.      :0.1167   Min.      :0.04996
## 1st Qu.:0.02956   1st Qu.:0.02031   1st Qu.:0.1620   1st Qu.:0.05770
## Median   :0.06154   Median   :0.03350   Median :0.1792   Median :0.06154
## Mean     :0.08055   Mean     :0.04666   Mean    :0.1792   Mean    :0.06217
## 3rd Qu.:0.11680   3rd Qu.:0.06847   3rd Qu.:0.1943   3rd Qu.:0.06569

```

```
## Max. :0.28100 Max. :0.15200 Max. :0.2459 Max. :0.07871
## radius_se texture_se perimeter_se area_se
## Min. :0.1115 Min. :0.3602 Min. :0.757 Min. : 6.802
## 1st Qu.:0.2324 1st Qu.:0.8339 1st Qu.:1.606 1st Qu.:17.850
## Median :0.3242 Median :1.1080 Median :2.287 Median :24.530
## Mean :0.3515 Mean :1.1530 Mean :2.464 Mean :28.351
## 3rd Qu.:0.4212 3rd Qu.:1.4100 3rd Qu.:2.974 3rd Qu.:31.980
## Max. :0.8426 Max. :2.4260 Max. :5.865 Max. :83.500
## smoothness_se compactness_se concavity_se concave.points_se
## Min. :0.001713 Min. :0.002252 Min. :0.00000 Min. :0.000000
## 1st Qu.:0.005169 1st Qu.:0.013080 1st Qu.:0.01509 1st Qu.:0.007638
## Median :0.006380 Median :0.020450 Median :0.02589 Median :0.010930
## Mean :0.006556 Mean :0.022649 Mean :0.02793 Mean :0.011128
## 3rd Qu.:0.007702 3rd Qu.:0.030260 3rd Qu.:0.03863 3rd Qu.:0.014030
## Max. :0.012430 Max. :0.060630 Max. :0.08232 Max. :0.025270
## symmetry_se fractal_dimension_se radius_worst texture_worst
## Min. :0.007882 Min. :0.0008948 Min. : 7.93 Min. :12.02
## 1st Qu.:0.015160 1st Qu.:0.0022480 1st Qu.:13.01 1st Qu.:21.08
## Median :0.018730 Median :0.0031870 Median :14.97 Median :25.41
## Mean :0.019254 Mean :0.0033619 Mean :15.84 Mean :25.50
## 3rd Qu.:0.022030 3rd Qu.:0.0041740 3rd Qu.:17.79 3rd Qu.:29.41
## Max. :0.035460 Max. :0.0080150 Max. :27.32 Max. :41.85
## perimeter_worst area_worst smoothness_worst compactness_worst
## Min. : 50.41 Min. : 185.2 Min. :0.08125 Min. :0.02729
## 1st Qu.: 84.11 1st Qu.: 515.3 1st Qu.:0.11660 1st Qu.:0.14720
## Median : 97.66 Median : 686.5 Median :0.13140 Median :0.21190
## Mean :104.52 Mean : 778.7 Mean :0.13170 Mean :0.23897
## 3rd Qu.:120.30 3rd Qu.: 906.6 3rd Qu.:0.14510 3rd Qu.:0.31500
## Max. :186.80 Max. :1933.0 Max. :0.18830 Max. :0.62470
## concavity_worst concave.points_worst symmetry_worst fractal_dimension_worst
## Min. :0.0000 Min. :0.00000 Min. :0.1565 Min. :0.05504
## 1st Qu.:0.1145 1st Qu.:0.06493 1st Qu.:0.2504 1st Qu.:0.07146
## Median :0.2267 Median :0.09993 Median :0.2821 Median :0.08004
## Mean :0.2576 Mean :0.11461 Mean :0.2821 Mean :0.08160
## 3rd Qu.:0.3728 3rd Qu.:0.16140 3rd Qu.:0.3109 3rd Qu.:0.08960
## Max. :0.7727 Max. :0.29100 Max. :0.4154 Max. :0.12240
```

En primer lugar, vamos a crear un gráfico para ver la distribución de la variable objetivo, diagnosis.

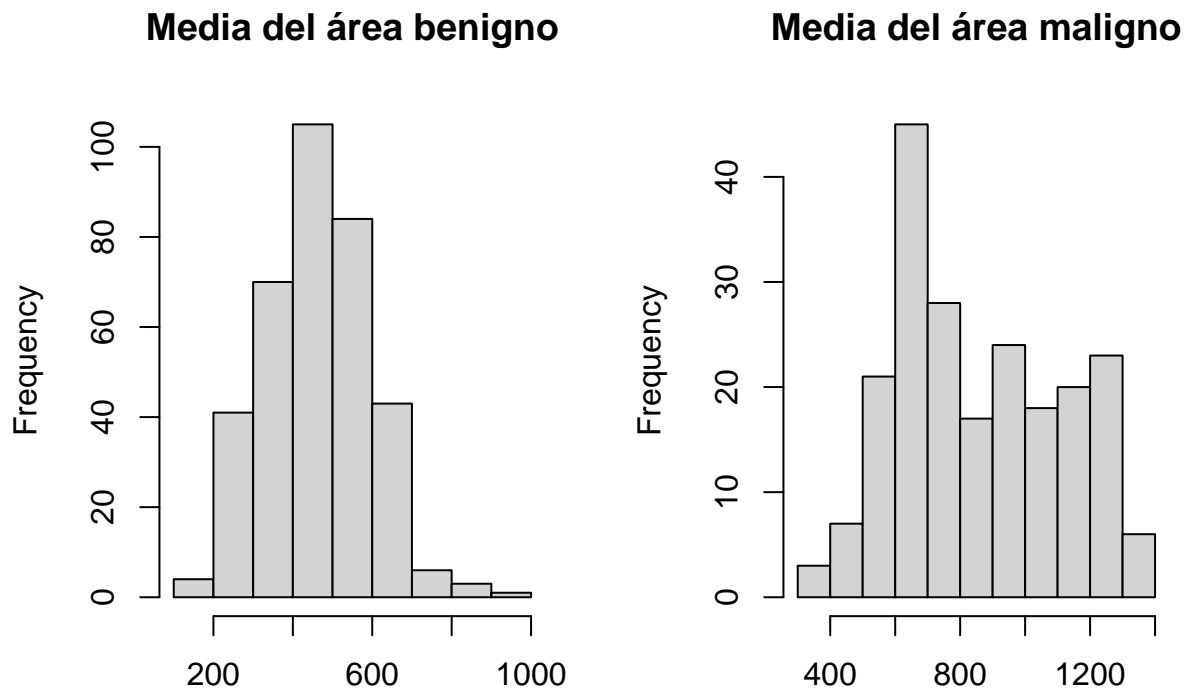
```
plot(x=cancer$diagnosis, main="Distribución de varoles según diagnóstico", xlab="Diagnóstico", ylab="")
```



Se observa que existen más observaciones de muestras benignas que malignas.

Vamos a realizar un par de gráficas para ver la distribución de las variables en base a la variable objetivo.

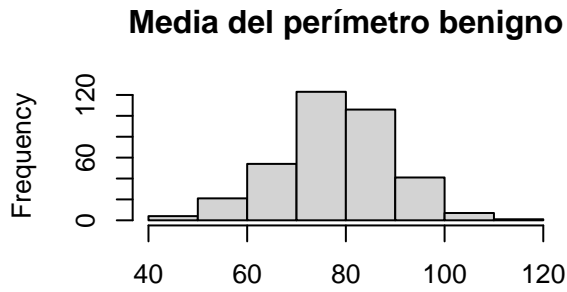
```
par(mfrow = c(1, 2))  
hist(filter(cancer, diagnosis == "B")$area_mean, main = "Media del área benigno")  
hist(filter(cancer, diagnosis == "M")$area_mean, main = "Media del área maligno")
```



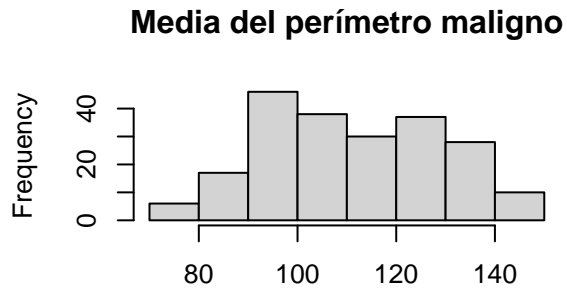
```
filter(cancer, diagnosis == "B")$area_me: filter(cancer, diagnosis == "M")$area_me
```

Se observa que la distribución de la media del área del tumor toma valores mayores en el caso de los tumores malignos. Se realizan los mismos gráficos con un par de variables más.

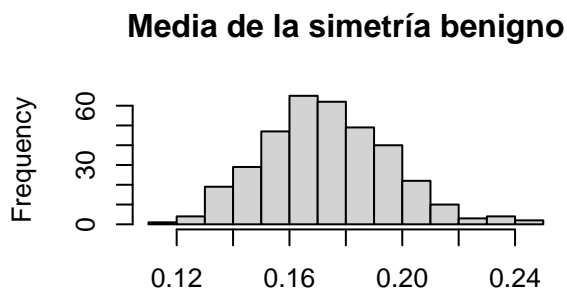
```
par(mfrow = c(2, 2))
hist(filter(cancer, diagnosis == "B")$perimeter_mean, main = "Media del perímetro benigno")
hist(filter(cancer, diagnosis == "M")$perimeter_mean, main = "Media del perímetro maligno")
hist(filter(cancer, diagnosis == "B")$symmetry_mean, main = "Media de la simetría benigno")
hist(filter(cancer, diagnosis == "M")$symmetry_mean, main = "Media de la simetría maligno")
```



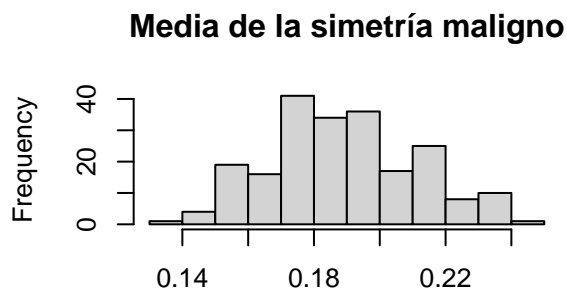
`filter(cancer, diagnosis == "B")$perimeter_mear`



`filter(cancer, diagnosis == "M")$perimeter_mear`



`filter(cancer, diagnosis == "B")$symmetry_mear`



`filter(cancer, diagnosis == "M")$symmetry_mear`

También se nota la diferencia en la distribución de valores para el perímetro, mientras que en la simetría la diferencia es menos notable.

4.2. Comprobación de la normalidad y homogeneidad de la varianza

Para la comprobación de la normalidad utilizaremos el test de Shapiro-Wilk.

De esa forma comparamos los p-valor obtenidos con $\alpha = 0,05$. Si el valor obtenido para cada variable es mayor, esta sigue una distribución normal y si no es el caso, lo mostramos en la salida.

```
alpha = 0.05
col.names = colnames(cancer)

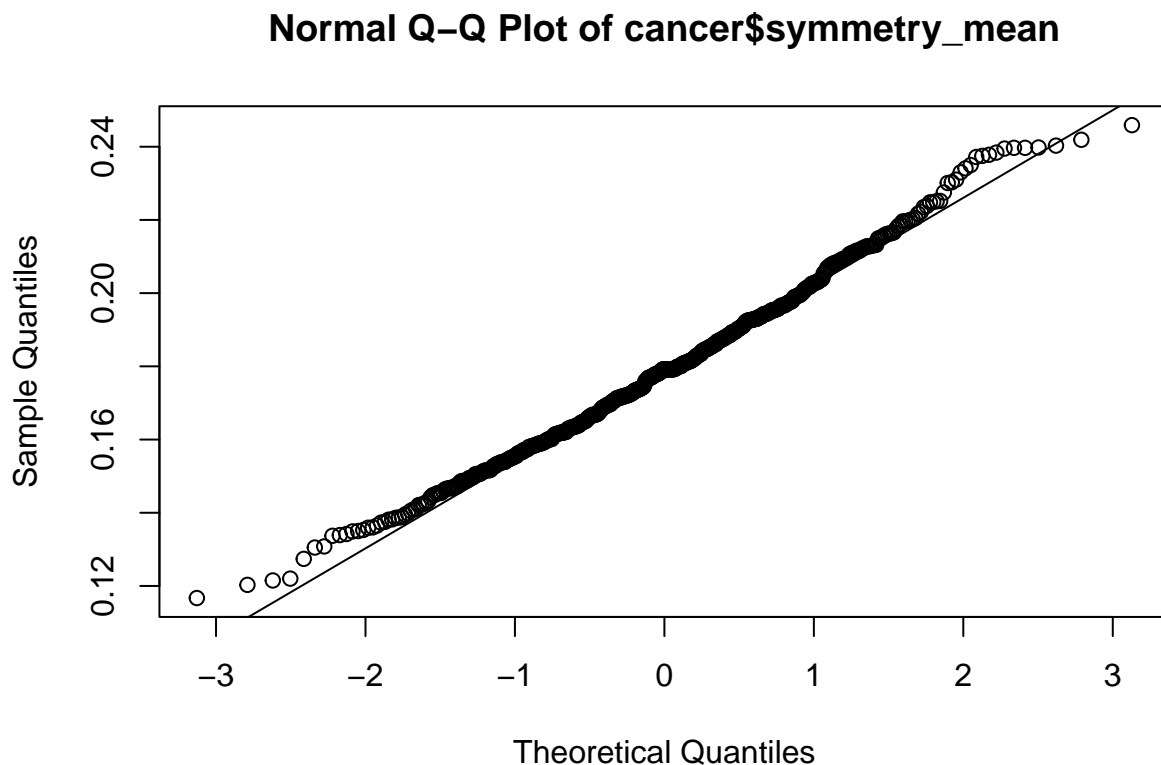
for (i in 1:ncol(cancer)) {
  if (i == 1) cat("Variables que no siguen una distribución normal y su p-value:\n")
  if (is.integer(cancer[,i]) | is.numeric(cancer[,i])) {
    p_val = shapiro.test(cancer[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      cat("(")
      cat(p_val)
      cat(")")
      # Format output
      if (i < ncol(cancer) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
}
```

```
}
}
```

```
## Variables que no siguen una distribución normal y su p-value:
## id(3.07445e-43), radius_mean(7.460958e-12),
## texture_mean(5.195521e-05), perimeter_mean(3.02881e-12), area_mean(1.206725e-17),
## compactness_mean(9.871618e-13), concavity_mean(3.043752e-18),
## concave.points_mean(8.317034e-18), symmetry_mean(0.0131504), fractal_dimension_mean(1.132071e-08),
## radius_se(6.060747e-18), texture_se(9.830276e-09), perimeter_se(7.199905e-17),
## area_se(7.553385e-23), smoothness_se(2.431925e-09), compactness_se(3.783127e-15),
## concavity_se(1.437786e-12), concave.points_se(0.000152261), symmetry_se(1.84265e-12),
## fractal_dimension_se(5.191323e-14), radius_worst(4.11553e-15), texture_worst(0.0001471121),
## perimeter_worst(4.780865e-15), area_worst(5.140758e-20), smoothness_worst(0.01133407),
## compactness_worst(9.694684e-14), concavity_worst(6.459841e-14), concave.points_worst(1.984878e-10),
## symmetry_worst(0.003773075)fractal_dimension_worst(9.326099e-11)
```

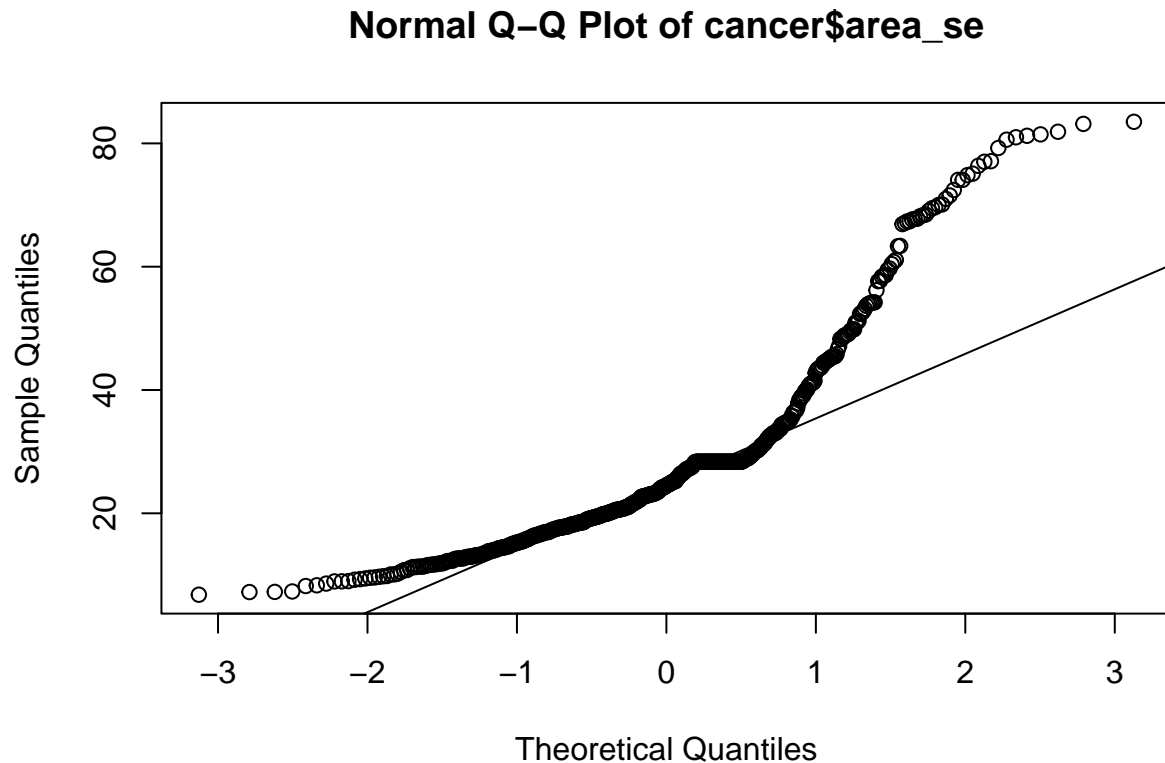
Veamos un ejemplo gráfico para la variable `cancer$symmetry_mean` (p-valor más elevado)

```
qqnorm(y = cancer$symmetry_mean, main = "Normal Q-Q Plot of cancer$symmetry_mean")
qqline(y = cancer$symmetry_mean)
```



Veamos un ejemplo gráfico para la variable `cancer$area_se` (p-valor más bajo)

```
qqnorm(y = cancer$area_se, main = "Normal Q-Q Plot of cancer$area_se")
qqline(y = cancer$area_se)
```



Seguidamente, pasamos a estudiar la homogeneidad de varianzas mediante la aplicación de un test de Fligner-Killeen.

Para poder utilizar esta función, deberemos convertir la variable **diagnosis** de factor a numérica, por lo que crearemos una nueva columna en la cual el valor “1” representa “B” y el valor “2” representa “M”.

```
library(stats)
cancer$diagnosis_num <- as.numeric(cancer$diagnosis)
fligner.test(diagnosis_num ~ radius_mean, data = cancer)
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data:  diagnosis_num by radius_mean
## Fligner-Killeen:med chi-squared = 467.9, df = 442, p-value = 0.1902
```

```
fligner.test(diagnosis_num ~ interaction(radius_mean+texture_mean+perimeter_mean+area_mean+smoothness_m
```

```
##
## Fligner-Killeen test of homogeneity of variances
##
## data:  diagnosis_num by interaction(radius_mean + texture_mean + perimeter_mean + area_mean + smoothn
## Fligner-Killeen:med chi-squared = NaN, df = 568, p-value = NA
```

Puesto que obtenemos un p-valor superior a 0.05 en el primer caso, aceptamos la hipótesis de que las varianzas de ambas muestras son homogéneas.

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos

4.3.1. Análisis de correlación entre variables

```
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_num"
for (i in 1:(ncol(cancer) - 1)) {
  if (is.integer(cancer[,i]) | is.numeric(cancer[,i])) {
    spearman_test = cor.test(cancer[,i],
                             cancer$diagnosis_num,
                             method = "spearman", exact=FALSE)
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    # Add row to matrix
    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(cancer)[i]
  }
}

corr_matrix[order(corr_matrix[, "estimate"]), ]
```

##	estimate	p-value
## symmetry_se	-0.13351626	1.412136e-03
## id	-0.07986636	5.691505e-02
## fractal_dimension_mean	-0.02220746	5.970616e-01
## smoothness_se	-0.01861251	6.577371e-01
## texture_se	0.02858125	4.962456e-01
## fractal_dimension_se	0.21967330	1.201329e-07
## fractal_dimension_worst	0.27959820	1.116014e-11
## symmetry_mean	0.31955798	5.645390e-15
## symmetry_worst	0.33473538	2.305978e-16
## smoothness_mean	0.36789763	1.119210e-19
## compactness_se	0.39221849	2.295020e-22
## smoothness_worst	0.41980396	1.068080e-25
## texture_mean	0.45736834	9.246165e-31
## texture_worst	0.46399560	1.012646e-31
## concave.points_se	0.48485355	6.943224e-35
## concavity_se	0.48863154	1.757207e-35
## radius_se	0.55479887	3.051137e-47
## perimeter_se	0.55985637	2.976668e-48
## compactness_worst	0.57778203	5.593289e-52
## compactness_mean	0.57849012	3.942091e-52
## area_se	0.62834594	7.765199e-64
## concavity_worst	0.70022692	4.992761e-85


```
## area_mean          0.70704341  2.263673e-87
## radius_mean        0.71482087  3.953044e-90
## concavity_mean     0.72925855  1.674819e-95
## perimeter_mean     0.73469136  1.291877e-97
## area_worst         0.75910665  8.545924e-108
## concave.points_mean 0.76947017  1.717687e-112
## radius_worst       0.77289494  4.248707e-114
## concave.points_worst 0.78167359  2.387021e-118
## perimeter_worst    0.78470761  7.289042e-120
```

Como se puede observar, la variable que más influye es **perimeter_worst**. Al mismo tiempo se puede ver como los p-valores son muy bajos.

4.3.2. Contraste de hipótesis

En este segundo modelo se quiere determinar si el ser diagnosticado depende del `perimeter_mean` (tamaño medio del tumor central), si éste tiene un valor inferior de 92.

```
cancer.low.perimeter_mean <- cancer[cancer$perimeter_mean <= 92,]$diagnosis_num
cancer.high.perimeter_mean <- cancer[cancer$perimeter_mean > 92,]$diagnosis_num

#Escalamos para normalizar
cancer.low.perimeter_mean <- scale(cancer.low.perimeter_mean, center=T, scale=T)
cancer.high.perimeter_mean <- scale(cancer.high.perimeter_mean, center=T, scale=T)

t.test(cancer.low.perimeter_mean, cancer.high.perimeter_mean, alternative = "less")

##
## Welch Two Sample t-test
##
## data: cancer.low.perimeter_mean and cancer.high.perimeter_mean
## t = -3.0128e-15, df = 446.19, p-value = 0.5
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 0.1427813
## sample estimates:
##      mean of x      mean of y
## -1.048615e-17  2.504940e-16
```

Puesto que obtenemos un p-valor superior a 0.05, damos por buena la hipótesis.

4.3.3. Modelo de regresión lineal

Un tercer modelo es la regresión lineal. Entre los diferentes modelos, calcularemos su coeficiente de determinación (R^2) para posteriormente realizar una predicción.

```
# Regresores cuantitativos con mayor coeficiente
# de correlación con respecto al precio
corr1 = cancer$perimeter_worst
corr2 = cancer$concave.points_worst
corr3 = cancer$radius_worst
```

```

corr4 = cancer$concave.points_mean
corr5 = cancer$area_worst
corr6 = cancer$perimeter_mean
corr7 = cancer$concavity_mean
corr8 = cancer$radius_mean
# Variable a predecir
diagnostico = cancer$diagnosis_num
# Generación de varios modelos
modelo1 <- lm(diagnostico ~ corr1 + corr2 + corr3 + corr4 + corr5 + corr6 + corr7 + corr8, data = cancer)
modelo2 <- lm(diagnostico ~ corr1 + corr2 + corr4 + corr6, data = cancer)
modelo3 <- lm(diagnostico ~ corr1 + corr2 + corr3 + corr5, data = cancer)
modelo4 <- lm(diagnostico ~ corr4 + corr6 + corr7 + corr8, data = cancer)

```

Para los anteriores modelos de regresión lineal múltiple obtenidos, podemos utilizar el coeficiente de determinación para medir la bondad de los ajustes y quedarnos con aquel modelo que mejor coeficiente presente.

```

# Tabla con los coeficientes de determinación de cada modelo
tabla.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
2, summary(modelo2)$r.squared,
3, summary(modelo3)$r.squared,
4, summary(modelo4)$r.squared),
ncol = 2, byrow = TRUE)
colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes

```

```

##      Modelo      R^2
## [1,]      1 0.7334748
## [2,]      2 0.7089986
## [3,]      3 0.7219513
## [4,]      4 0.6659149

```

En este caso, el primer modelo tiene un mayor coeficiente de determinación. Ahora, empleando este modelo, realizaremos predicciones:

```

newdata1 <- data.frame(
  corr1 = 184.60,
  corr2 = 0.26540,
  corr3 = 25.38,
  corr4 = 0.14710,
  corr5 = 2019,
  corr6 = 122.80,
  corr7 = 0.30010,
  corr8 = 17.990
)

newdata2 <- data.frame(
  corr1 = 65.13,
  corr2 = 0.06227,
  corr3 = 10.23,
  corr4 = 0.02076,
  corr5 = 314.9,
  corr6 = 60.34,

```

```
corr7 = 0.02956,
corr8 = 9.504
)

# Predecir el diagnostico
predict(modelo1, newdata1)
```

```
##          1
## 2.712086
```

```
predict(modelo1, newdata2)
```

```
##          1
## 0.9233966
```

Como vemos, en estos ejemplos con valores del propio dataset, las predicciones son cercanas a “2” y a “1” respectivamente, tal y como esos mismos registros tienen informado en la variable **diagnosis_num**. Los registros son el 1 y el 22.

5. Representación de los resultados a partir de tablas y gráficas.

Este apartado se ha respondido a lo largo de la práctica.

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

A la luz de los resultados mostrados en cada uno de los apartados anteriores, mediante el uso del dataset indicado, podemos concluir que mediante un modelo de regresión lineal podemos realizar predicciones para la detección del cáncer de mama.

Contribución

```
Contribuciones <- c("Investigación previa", "Redacción de las respuestas", "Desarrollo código")
Firmas <- c("CRS, YFR", "CRS, YFR", "CRS, YFR")
tabla <- data.frame(cbind(Contribuciones, Firmas))
knitr::kable(tabla)
```

Contribuciones	Firmas
Investigación previa	CRS, YFR
Redacción de las respuestas	CRS, YFR
Desarrollo código	CRS, YFR