Parcial 2

Moreno - Ochoa - Casarubio

29/10/2021

2 Examen Parcial Ciencias de Datos

NOMBRES:

- Miguel Moreno.
- Esteban Ochoa.
- David Casarrubia.

Instrucciones:

- El parcial será desarrollado en los grupos (3 estudiantes máx.) asignados.
- No olvide escribir los nombres completos de TODOS los integrantes
- Cada grupo debe responder a las preguntas que están en el documento, con su respectivo código y solución.
- Adicional al documento (preferiblemente escrito en markdown, extension .Rmd), debe entregar el Script con el
  código debidamente comentado.

Introdución

El cáncer de mama (CM) es uno de los cánceres más comunes entre las mujeres en todo el mundo, y representa la mayoría de los casos nuevos de cáncer y las muertes relacionadas con el cáncer según las estadísticas mundiales, lo que lo convierte en un problema de salud pública importante en la sociedad actual. El diagnóstico precoz de CM puede mejorar el pronóstico y mejorar tasa de supervivencia significativamente, ya que puede promover el tratamiento oportuno a los pacientes que lo padecen. Una clasificación más precisa de los tumores benignos puede evitar que los pacientes se sometan a tratamientos innecesarios. Por tanto, el diagnóstico correcto de CM y la clasificación de los pacientes en grupos malignos o benignos es objeto de mucha investigación. Teniendo en cuenta, el desarrollo de la inteligencia artificial y la aplicación de algoritmos de aprendizaje automático (ML), la clasificación de patrones de CM y el modelado pronóstico ha permitdo la deteccion temprana cancer en sus etapas iniciales.

Los métodos de clasificación y extracción de datos son una forma eficaz de clasificar eventos adversos y son técnicas ampliamente en el diagnóstico y análisis para tomar decisiones en el sector clínico.

Pregunta de Investigación:

¿Cuáles son las variables más relevantes del data set que permiten predecir si una mujer tiene o no cancer de mama?

Siga los pasos de esta guía y responda a las siguientes siguientes preguntas:

Paso 1: Carge las libraries que utilizará y el dataset (cancer_Breast)

Importamos las librerias que necesitamos:

```
library(caret)
library(dplyr)
library(ggplot2)
library(tidyverse)
library(mice)
library(pROC)
library("gridExtra")
library(PerformanceAnalytics)
```

Importamos el data set:

```
cancer <- read_delim("breastCancer_parcial.csv",
                     delim = ";", escape_double = FALSE, trim_ws = TRUE)
  cancer <- data.frame(cancer)
```

Paso 2: Exploración de los datos. **(+15 puntos)**

```
 cat("La dimension del data set es:",dim(cancer))
```

```
## La dimension del data set es: 500 32
```

```
 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            0.0
## 2     0.01389             0.003532        24.99         23.41          158.8
## 3     0.02250             0.004571        23.57         25.53          152.5
## 4     0.05963             0.009208        14.91         26.50            0.0
## 5     0.01756             0.005115        22.54         16.67          152.2
## 6     0.02165             0.005082        15.47         23.75          103.4
##   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
## 1               0.2654         0.4601                 0.11890
## 2               0.1860         0.2750                 0.08902
## 3               0.2430         0.3613                 0.08758
## 4               0.2575         0.6638                 0.17300
## 5               0.1625         0.2364                 0.07678
## 6               0.1741         0.3985                 0.12440
```

```
glimpse(cancer)
```

```
## Rows: 500
## Columns: 32
## $ id                      <dbl> 842302, 842517, 84300903, 84348301, 84358402, ~
## $ diagnosis               <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "~
## $ radius_mean             <dbl> 17.990, 20.570, 19.690, 11.420, 20.290, 12.450~
## $ texture_mean            <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15.70, 19.9~
## $ perimeter_mean          <dbl> 122.80, 132.90, 130.00, 77.58, 135.10, 82.57, ~
## $ area_mean               <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0, 477.1, ~
## $ smoothness_mean         <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0.10030, 0~
## $ compactness_mean        <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0.13280, 0~
## $ concavity_mean          <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0.19800, 0~
## $ concave.points_mean     <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0.10430, 0~
## $ symmetry_mean           <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809, 0.2087~
## $ fractal_dimension_mean  <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0.05883, 0~
## $ radius_se               <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572, 0.3345~
## $ texture_se              <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813, 0.8902~
## $ perimeter_se            <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.217, 3.18~
## $ area_se                 <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27.19, 53.~
## $ smoothness_se           <dbl> 0.006399, 0.005225, 0.006150, 0.009110, 0.0114~
## $ compactness_se          <dbl> 0.049040, 0.013080, 0.040060, 0.074580, 0.0246~
## $ concavity_se            <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0.05688, 0~
## $ concave.points_se       <dbl> 0.015870, 0.013400, 0.020580, 0.018670, 0.0188~
## $ symmetry_se             <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0.01756, 0~
## $ fractal_dimension_se    <dbl> 0.006193, 0.003532, 0.004571, 0.009208, 0.0051~
## $ radius_worst            <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15.47, 22.8~
## $ texture_worst           <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23.75, 0.00~
## $ perimeter_worst         <dbl> 0.00, 158.80, 152.50, 0.00, 152.20, 103.40, 15~
```

```
## $ area_worst              <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0, 741.6, ~
## $ smoothness_worst        <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374, 0.1791~
## $ compactness_worst       <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050, 0.5249~
## $ concavity_worst         <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0.40000, 0~
## $ concave.points_worst    <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0.16250, 0~
## $ symmetry_worst          <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364, 0.3985~
## $ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0.07678, 0~
```

Responder las siguientes preguntas:

1. Convierte la variable *diagnosis* a una variable categorica (factor), para esto debes usar la función mutute y %>% (pipe) del package **DPLYR**. Esta variable será el outcome.

```
cancer$diagnosis <- as.factor(cancer$diagnosis) # Se convierte en una variable factor a diagnosis
glimpse(cancer)
```

```
## Rows: 500
## Columns: 32
## $ id                      <dbl> 842302, 842517, 84300903, 84348301, 84358402, ~
## $ diagnosis               <fct> M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M~
## $ radius_mean             <dbl> 17.990, 20.570, 19.690, 11.420, 20.290, 12.450~
## $ texture_mean            <dbl> 10.38, 17.77, 21.25, 20.38, 14.34, 15.70, 19.9~
## $ perimeter_mean          <dbl> 122.80, 132.90, 130.00, 77.58, 135.10, 82.57, ~
## $ area_mean               <dbl> 1001.0, 1326.0, 1203.0, 386.1, 1297.0, 477.1, ~
## $ smoothness_mean         <dbl> 0.11840, 0.08474, 0.10960, 0.14250, 0.10030, 0~
## $ compactness_mean        <dbl> 0.27760, 0.07864, 0.15990, 0.28390, 0.13280, 0~
## $ concavity_mean          <dbl> 0.30010, 0.08690, 0.19740, 0.24140, 0.19800, 0~
## $ concave.points_mean     <dbl> 0.14710, 0.07017, 0.12790, 0.10520, 0.10430, 0~
## $ symmetry_mean           <dbl> 0.2419, 0.1812, 0.2069, 0.2597, 0.1809, 0.2087~
## $ fractal_dimension_mean  <dbl> 0.07871, 0.05667, 0.05999, 0.09744, 0.05883, 0~
## $ radius_se               <dbl> 1.0950, 0.5435, 0.7456, 0.4956, 0.7572, 0.3345~
## $ texture_se              <dbl> 0.9053, 0.7339, 0.7869, 1.1560, 0.7813, 0.8902~
## $ perimeter_se            <dbl> 8.589, 3.398, 4.585, 3.445, 5.438, 2.217, 3.18~
## $ area_se                 <dbl> 153.40, 74.08, 94.03, 27.23, 94.44, 27.19, 53.~
## $ smoothness_se           <dbl> 0.006399, 0.005225, 0.006150, 0.009110, 0.0114~
## $ compactness_se          <dbl> 0.049040, 0.013080, 0.040060, 0.074580, 0.0246~
## $ concavity_se            <dbl> 0.05373, 0.01860, 0.03832, 0.05661, 0.05688, 0~
## $ concave.points_se       <dbl> 0.015870, 0.013400, 0.020580, 0.018670, 0.0188~
## $ symmetry_se             <dbl> 0.03003, 0.01389, 0.02250, 0.05963, 0.01756, 0~
## $ fractal_dimension_se    <dbl> 0.006193, 0.003532, 0.004571, 0.009208, 0.0051~
## $ radius_worst            <dbl> 25.38, 24.99, 23.57, 14.91, 22.54, 15.47, 22.8~
## $ texture_worst           <dbl> 17.33, 23.41, 25.53, 26.50, 16.67, 23.75, 0.00~
## $ perimeter_worst         <dbl> 0.00, 158.80, 152.50, 0.00, 152.20, 103.40, 15~
## $ area_worst              <dbl> 2019.0, 1956.0, 1709.0, 567.7, 1575.0, 741.6, ~
## $ smoothness_worst        <dbl> 0.1622, 0.1238, 0.1444, 0.2098, 0.1374, 0.1791~
## $ compactness_worst       <dbl> 0.6656, 0.1866, 0.4245, 0.8663, 0.2050, 0.5249~
## $ concavity_worst         <dbl> 0.71190, 0.24160, 0.45040, 0.68690, 0.40000, 0~
## $ concave.points_worst    <dbl> 0.26540, 0.18600, 0.24300, 0.25750, 0.16250, 0~
## $ symmetry_worst          <dbl> 0.4601, 0.2750, 0.3613, 0.6638, 0.2364, 0.3985~
## $ fractal_dimension_worst <dbl> 0.11890, 0.08902, 0.08758, 0.17300, 0.07678, 0~
```

R// se comvirtio en una vaiable factor a diagnosis

2. Cuántas pacientes y predictores tiene el data set ?

```
dim(cancer)
```

```
## [1] 500  32
```

R// Numero de paciente 500 y predictores 30 sin contar id, diagnosis

3. Realice un resumen de sus datos:
    - El data set cuenta con datos faltantes? Cree una tabla que indique el Número de NA por cada variable, si aplica.
    - El data set cuenta con datos Nulos, (ceros). Cree una tabla si aplica con el conteo de estos dato nulos.

R//

```
Na <- sum(is.na(cancer))
cat("Numero de datos NA:",Na)
```

```
## Numero de datos NA: 133
```

```
  # Crear una tabla con valores faltantes
  missing.values <- cancer %>%
    gather(key = "key", value = "val") %>%
    mutate(is.missing = is.na(val)) %>%
    group_by(key, is.missing) %>%
    summarise(num.missing = n()) %>%
    filter(is.missing==T)  %>%
    arrange(desc(num.missing))
  missing.values  %>% knitr::kable()
```

key	is.missing	num.missing
texture_worst	TRUE	11
symmetry_mean	TRUE	10
fractal_dimension_worst	TRUE	8
compactness_mean	TRUE	7
compactness_worst	TRUE	7
radius_worst	TRUE	7
concave.points_mean	TRUE	6
perimeter_se	TRUE	6
smoothness_mean	TRUE	6
texture_se	TRUE	6
compactness_se	TRUE	5
concave.points_se	TRUE	5
fractal_dimension_mean	TRUE	5
smoothness_se	TRUE	5
area_mean	TRUE	4
concavity_mean	TRUE	4
concavity_se	TRUE	4

key	is.missing	num.missing
fractal_dimension_se	TRUE	4
perimeter_mean	TRUE	4
perimeter_worst	TRUE	4
radius_se	TRUE	4
symmetry_se	TRUE	4
smoothness_worst	TRUE	2
area_se	TRUE	1
area_worst	TRUE	1
concave.points_worst	TRUE	1
radius_mean	TRUE	1
texture_mean	TRUE	1

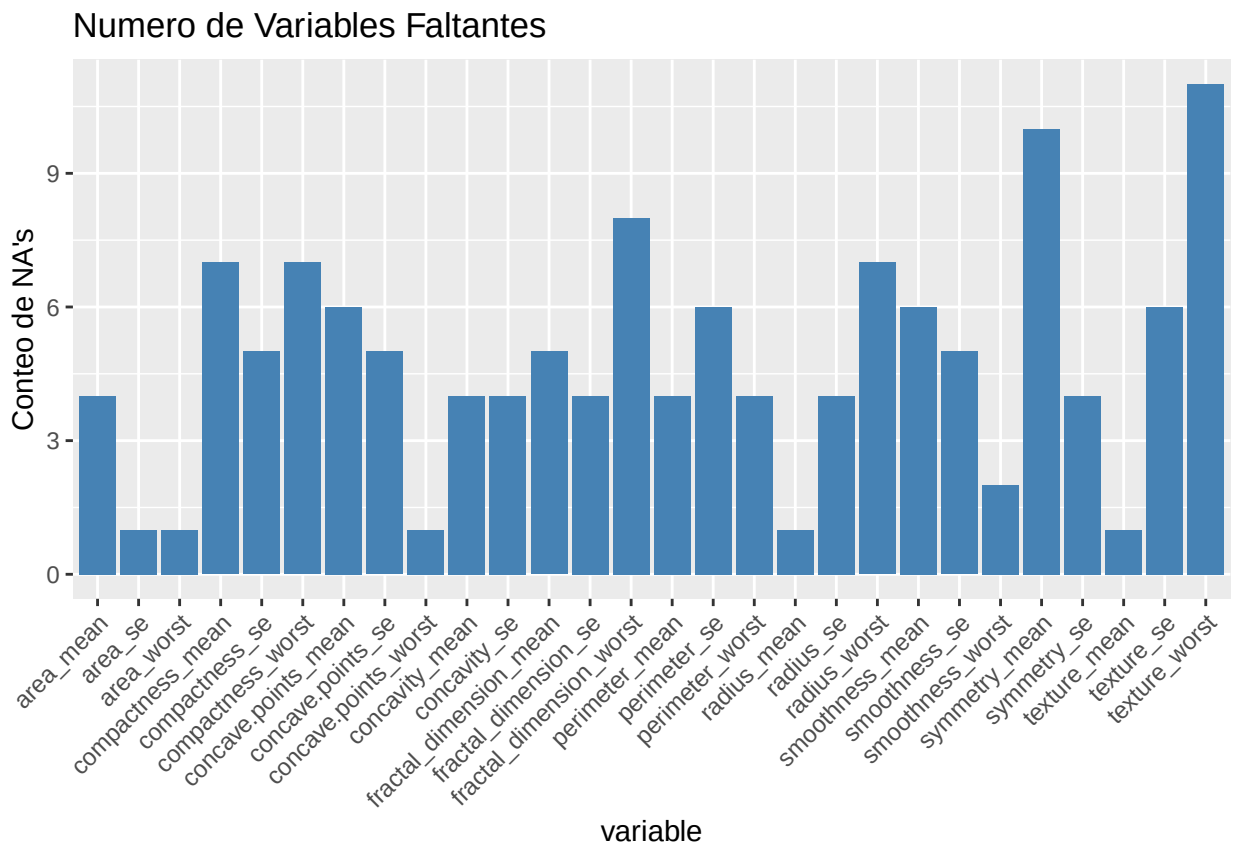
```
Null <- sum(is.null(cancer))
cat("Numero de datos Null:",Null)
```

```
## Numero de datos Null: 0
```

4. En una gráfica puede resumir(contar) estos valores (NA y Null). Recomendación: use ggplot2.

R// Tabla de datos faltante

```
missing.values %>%
   ggplot() +
   geom_bar(aes(x=key, y = num.missing), stat = 'identity', fill="steelblue") +
   labs(x='variable', y="Conteo de NA's", title='Numero de Variables Faltantes') +
   theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



5. En la variable "diagnosis", asigne Maligno a valores con M y Benigno a valores con B; y conviertala nuevamente a factor.

```
cancer$diagnosis <- as.numeric(cancer$diagnosis) # Se convierte en variable numerica
 cancer$diagnosis[cancer$diagnosis == 2] <- "Maligno" # Se cambia el valor
 cancer$diagnosis[cancer$diagnosis == 1] <- "Benigno" # Se cambia el valor
 cancer$diagnosis <- as.factor(cancer$diagnosis) # Se convierte nueva mente en factor
 str(cancer$diagnosis)
```

```
##  Factor w/ 2 levels "Benigno","Maligno": 2 2 2 2 2 2 2 2 2 2 ...
```

6. ¿Cuál es el porcentaje de pacientes con diagnóstico de cancer Beningno y cuáles son Malignos?. Use el package ggplot2 para graficar un barplot con estos datos.

```
 # Porcentaje de personas con diagnostico maligno
cat("Procentaje de personas con diagnostico maligno :",
    round(((sum(cancer$diagnosis=="Maligno"))/(sum(cancer$diagnosis=="Maligno")+
                                                 sum(cancer$diagnosis=="Benigno")))*100 , 2))
```

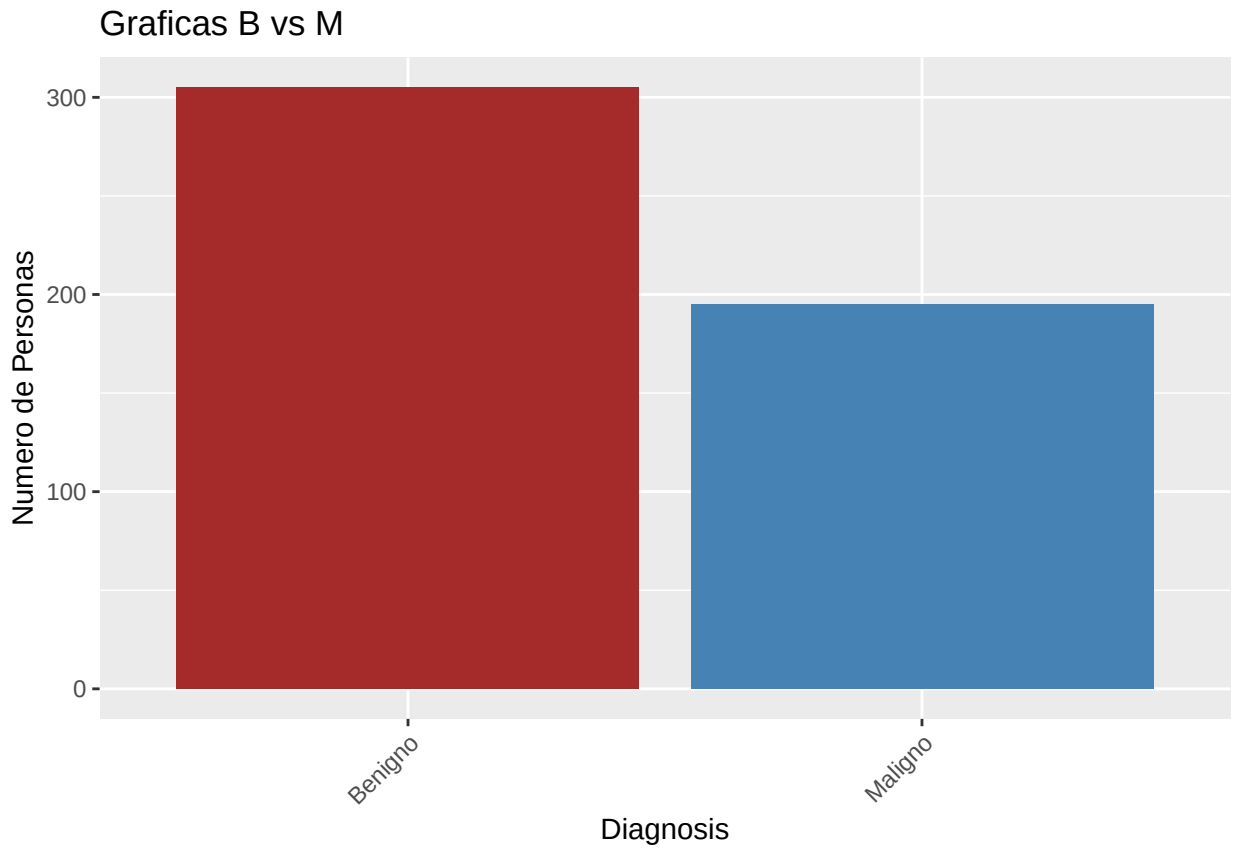
```
## Procentaje de personas con diagnostico maligno : 39
```

```
 # Porcentaje de personas con diagnostico benigno
cat("Procentaje de personas con diagnostico maligno :",
    round(((sum(cancer$diagnosis=="Benigno"))/(sum(cancer$diagnosis=="Maligno")+
                                                 sum(cancer$diagnosis=="Benigno")))*100 , 2) )
```

```
## Procentaje de personas con diagnostico maligno : 61
```

```
  #Grafica de B vs M
  cancer %>%
    ggplot() +
    geom_bar(aes(x= cancer$diagnosis), fill= c("brown","steelblue")) +
    labs(x='Diagnosis', y="Numero de Personas", title='Graficas B vs M') +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Use of ‘cancer$diagnosis‘ is discouraged. Use ‘diagnosis‘ instead.
```



Paso 3: Preparación de los datos **(+ 15 puntos)**

7. Teniendo en cuenta el número de valores nulos y NA del dataset, seleccione un método para que realizar la imputación de valores: i) Omitir los datos faltantes, ii) calcular la media de cada variable y asignar el el valor a los datos faltantes, iii) Usar la funcion MICE para hacer la imputación de valores Nota: la selección del método dependerá la exactitud de modelo.

```
#Remplazo de los datos Na por la media
  cancer$radius_mean[is.na(cancer$radius_mean)] <- mean(cancer$radius_mean, na.rm = TRUE)
  cancer$texture_mean[is.na(cancer$texture_mean)] <- mean(cancer$texture_mean, na.rm = TRUE)
  cancer$perimeter_mean[is.na(cancer$perimeter_mean)] <- mean(cancer$perimeter_mean, na.rm = TRUE)
  cancer$area_mean[is.na(cancer$area_mean)] <- mean(cancer$area_mean, na.rm = TRUE)
  cancer$smoothness_mean[is.na(cancer$smoothness_mean)] <- mean(cancer$smoothness_mean, na.rm = TRUE)
  cancer$compactness_mean[is.na(cancer$compactness_mean)] <- mean(cancer$compactness_mean, na.rm = TRUE
```

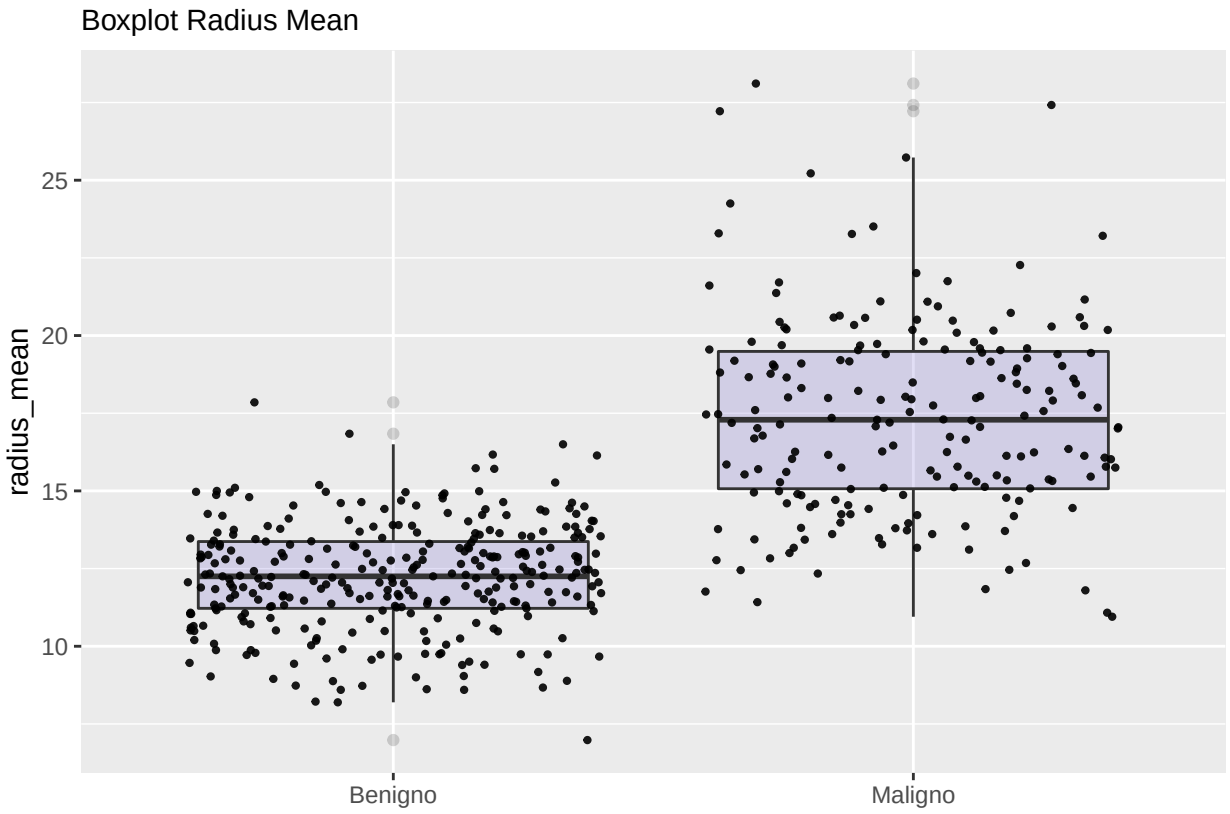
```r
cancer$concavity_mean[is.na(cancer$concavity_mean)] <- mean(cancer$concavity_mean, na.rm = TRUE)
cancer$concave.points_mean[is.na(cancer$concave.points_mean)] <- mean(cancer$concave.points_mean, na.
cancer$symmetry_mean[is.na(cancer$symmetry_mean)] <- mean(cancer$symmetry_mean, na.rm = TRUE)
cancer$fractal_dimension_mean[is.na(cancer$fractal_dimension_mean)] <- mean(cancer$fractal_dimension_
cancer$radius_se[is.na(cancer$radius_se)] <- mean(cancer$radius_se, na.rm = TRUE)
cancer$texture_se[is.na(cancer$texture_se)] <- mean(cancer$texture_se, na.rm = TRUE)
cancer$perimeter_se[is.na(cancer$perimeter_se)] <- mean(cancer$perimeter_se, na.rm = TRUE)
cancer$area_se[is.na(cancer$area_se)] <- mean(cancer$area_se, na.rm = TRUE)
cancer$smoothness_se[is.na(cancer$smoothness_se)] <- mean(cancer$smoothness_se, na.rm = TRUE)
cancer$compactness_se[is.na(cancer$compactness_se)] <- mean(cancer$compactness_se, na.rm = TRUE)
cancer$concavity_se[is.na(cancer$concavity_se)] <- mean(cancer$concavity_se, na.rm = TRUE)
cancer$concave.points_se[is.na(cancer$concave.points_se)] <- mean(cancer$concave.points_se, na.rm = TR
cancer$symmetry_se[is.na(cancer$symmetry_se)] <- mean(cancer$symmetry_se, na.rm = TRUE)
cancer$fractal_dimension_se[is.na(cancer$fractal_dimension_se)] <- mean(cancer$fractal_dimension_se, n
cancer$radius_worst[is.na(cancer$radius_worst)] <- mean(cancer$radius_worst, na.rm = TRUE)
cancer$texture_worst[is.na(cancer$texture_worst)] <- mean(cancer$texture_worst, na.rm = TRUE)
cancer$perimeter_worst[is.na(cancer$perimeter_worst)] <- mean(cancer$perimeter_worst, na.rm = TRUE)
cancer$area_worst[is.na(cancer$area_worst)] <- mean(cancer$area_worst, na.rm = TRUE)
cancer$smoothness_worst[is.na(cancer$smoothness_worst)] <- mean(cancer$smoothness_worst, na.rm = TRUE)
cancer$compactness_worst[is.na(cancer$compactness_worst)] <- mean(cancer$compactness_worst, na.rm = TR
cancer$concave.points_worst[is.na(cancer$concave.points_worst)] <- mean(cancer$concave.points_worst, n
cancer$fractal_dimension_worst[is.na(cancer$fractal_dimension_worst)] <- mean(cancer$fractal_dimensio
summary(cancer)
```

```
##        id              diagnosis    radius_mean      texture_mean
##  Min.   :     8670   Benigno:305   Min.   : 6.981   Min.   : 9.71
##  1st Qu.:   866704   Maligno:195   1st Qu.:11.807   1st Qu.:16.07
##  Median :   901432                 Median :13.445   Median :18.70
##  Mean   : 32630491                 Mean   :14.226   Mean   :19.09
##  3rd Qu.:  8910808                 3rd Qu.:16.115   3rd Qu.:21.56
##  Max.   :911320502                 Max.   :28.110   Max.   :39.28
##  perimeter_mean     area_mean      smoothness_mean   compactness_mean
##  Min.   : 43.79   Min.   : 143.5   Min.   :0.06251   Min.   :0.01938
##  1st Qu.: 76.13   1st Qu.: 432.0   1st Qu.:0.08629   1st Qu.:0.06374
##  Median : 87.03   Median : 558.6   Median :0.09595   Median :0.09235
##  Mean   : 92.73   Mean   : 664.4   Mean   :0.09603   Mean   :0.10401
##  3rd Qu.:106.22   3rd Qu.: 800.8   3rd Qu.:0.10495   3rd Qu.:0.13040
##  Max.   :188.50   Max.   :2501.0   Max.   :0.14470   Max.   :0.34540
##  concavity_mean    concave.points_mean symmetry_mean    fractal_dimension_mean
##  Min.   :0.00000   Min.   :0.00000     Min.   :0.1167   Min.   :0.04996
##  1st Qu.:0.02940   1st Qu.:0.02035     1st Qu.:0.1621   1st Qu.:0.05761
##  Median :0.06583   Median :0.03484     Median :0.1799   Median :0.06128
##  Mean   :0.09036   Mean   :0.04946     Mean   :0.1814   Mean   :0.06245
##  3rd Qu.:0.13215   3rd Qu.:0.07432     3rd Qu.:0.1953   3rd Qu.:0.06567
##  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.:0.2343   1st Qu.:0.8261   1st Qu.: 1.647   1st Qu.: 18.035
##  Median :0.3281   Median :1.0880   Median : 2.328   Median : 24.970
##  Mean   :0.4106   Mean   :1.1984   Mean   : 2.903   Mean   : 41.130
##  3rd Qu.:0.4954   3rd Qu.:1.4578   3rd Qu.: 3.407   3rd Qu.: 46.742
##  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.   :0.003012   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.005152   1st Qu.:0.013048   1st Qu.:0.01510   1st Qu.:0.007634
##  Median :0.006254   Median :0.020350   Median :0.02599   Median :0.010905
##  Mean   :0.006910   Mean   :0.025713   Mean   :0.03219   Mean   :0.011810
##  3rd Qu.:0.007975   3rd Qu.:0.032468   3rd Qu.:0.04237   3rd Qu.:0.014615
##  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.   : 0.00   Min.   : 0.00
##  1st Qu.:0.015027   1st Qu.:0.0022050    1st Qu.:12.84   1st Qu.:21.05
##  Median :0.018795   Median :0.0031195    Median :14.85   Median :25.39
##  Mean   :0.020689   Mean   :0.0037765    Mean   :15.63   Mean   :25.39
##  3rd Qu.:0.023533   3rd Qu.:0.0044662    3rd Qu.:18.50   3rd Qu.:29.25
##  Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54
##  perimeter_worst    area_worst     smoothness_worst  compactness_worst
##  Min.   :  0.00   Min.   :   0.0   Min.   :0.07117   Min.   :0.02729
##  1st Qu.: 83.66   1st Qu.: 521.6   1st Qu.:0.11650   1st Qu.:0.14592
##  Median : 97.18   Median : 691.8   Median :0.13130   Median :0.21685
##  Mean   :104.22   Mean   : 893.6   Mean   :0.13200   Mean   :0.25651
##  3rd Qu.:125.17   3rd Qu.:1150.8   3rd Qu.:0.14600   3rd Qu.:0.34223
##  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.   :0.0000       Min.   :0.1565   Min.   :0.00000
##  1st Qu.:0.1145   1st Qu.:0.0633       1st Qu.:0.2517   1st Qu.:0.07109
##  Median :0.2314   Median :0.1007       Median :0.2831   Median :0.07996
##  Mean   :0.2764   Mean   :0.1160       Mean   :0.2922   Mean   :0.08311
##  3rd Qu.:0.3895   3rd Qu.:0.1668       3rd Qu.:0.3201   3rd Qu.:0.09186
##  Max.   :1.2520   Max.   :0.2910       Max.   :0.6638   Max.   :0.20750
```

8. Identifique y elimine los outliers(datos anormales, si aplica) Recomendación: use la boxplot del package ggplot2 .

```
 ## eliminacion de out por variables

# Distribucion de los datos antes de eliminar los outlider

 ## distribucion de los datos
 cancer %>%
   ggplot( aes(x=diagnosis, y=radius_mean)) +
   geom_boxplot(fill="slateblue", alpha=0.2) +
   geom_jitter(color="black", size=0.8, alpha=0.9) +
   theme(
     legend.position="none",
     plot.title = element_text(size=11)
   ) +
   ggtitle("Boxplot Radius Mean") +
   xlab("")
```




```
## Variable 1

  boxplot.stats(cancer$radius_mean) # out encima de 22.271
```

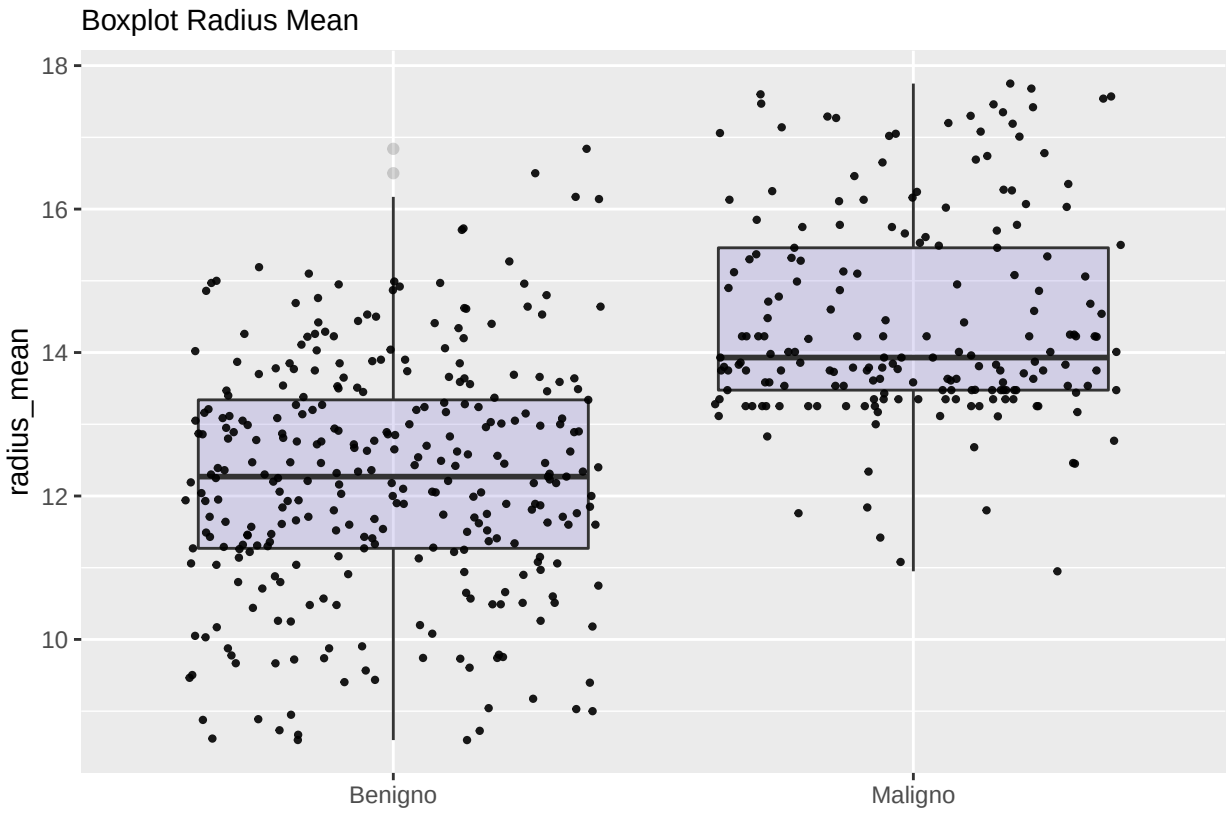
```
## $stats
## [1]  6.981 11.805 13.445 16.120 22.270
##
## $n
## [1] 500
##
## $conf
## [1] 13.1401 13.7499
##
## $out
##  [1] 25.22 24.25 23.27 27.22 23.29 28.11 23.21 23.51 25.73 27.42
```

```
  cancer$radius_mean[cancer$radius_mean > 22.270]<- mean(cancer$radius_mean)
  cancer$radius_mean[cancer$radius_mean > 21.370]<- mean(cancer$radius_mean)
  cancer$radius_mean[cancer$radius_mean > 20.940]<- mean(cancer$radius_mean)
  cancer$radius_mean[cancer$radius_mean > 20.730]<- mean(cancer$radius_mean)
  cancer$radius_mean[cancer$radius_mean > 20.640]<- mean(cancer$radius_mean)
  cancer$radius_mean[cancer$radius_mean > 20.590]<- mean(cancer$radius_mean)
  cancer$radius_mean[cancer$radius_mean > 20.510]<- mean(cancer$radius_mean)
  cancer$radius_mean[cancer$radius_mean > 20.340]<- mean(cancer$radius_mean)
  cancer$radius_mean[cancer$radius_mean > 19.810]<- mean(cancer$radius_mean)
```

```r
cancer$radius_mean[cancer$radius_mean > 19.690]<- mean(cancer$radius_mean)
cancer$radius_mean[cancer$radius_mean > 19.550]<- mean(cancer$radius_mean)
cancer$radius_mean[cancer$radius_mean > 19.440]<- mean(cancer$radius_mean)
cancer$radius_mean[cancer$radius_mean > 19.020]<- mean(cancer$radius_mean)
cancer$radius_mean[cancer$radius_mean > 18.490]<- mean(cancer$radius_mean)
cancer$radius_mean[cancer$radius_mean < 8.1960]<- mean(cancer$radius_mean)
cancer$radius_mean[cancer$radius_mean > 17.930]<- mean(cancer$radius_mean)
cancer$radius_mean[cancer$radius_mean > 17.750]<- mean(cancer$radius_mean)
cancer$radius_mean[cancer$radius_mean < 8.5970]<- mean(cancer$radius_mean)
boxplot.stats(cancer$radius_mean)
```

```
## $stats
## [1]  8.59700 11.84500 13.25276 14.21000 17.75000
##
## $n
## [1] 500
##
## $conf
## [1] 13.08565 13.41987
##
## $out
## numeric(0)
```

```r
# Distribucion de los datos despues de eliminar los outlider

## distribucion de los datos
cancer %>%
  ggplot( aes(x=diagnosis, y=radius_mean)) +
  geom_boxplot(fill="slateblue", alpha=0.2) +
  geom_jitter(color="black", size=0.8, alpha=0.9) +
  theme(
    legend.position="none",
    plot.title = element_text(size=11)
  ) +
  ggtitle("Boxplot Radius Mean") +
  xlab("")
```




```
  ## Variable 2

  boxplot.stats(cancer$texture_mean) # out encima de 29.290
```

```
## $stats
## [1]  9.710 16.070 18.700 21.565 29.290
##
## $n
## [1] 500
##
## $conf
## [1] 18.31172 19.08828
##
## $out
## [1] 32.47 33.81 39.28 33.56 31.12 29.81 30.72 29.97
```

```
  cancer$texture_mean[cancer$texture_mean > 29.290]<- mean(cancer$texture_mean)
  boxplot.stats(cancer$texture_mean)
```

```
## $stats
## [1]  9.710 16.070 18.700 21.445 29.290
##
## $n
## [1] 500
##
```

```
## $conf
## [1] 18.3202 19.0798
##
## $out
## numeric(0)
```

```
  ## Variable 3

  boxplot.stats(cancer$perimeter_mean) # out encima de 147.300
```

```
## $stats
## [1]  43.790  76.120  87.035 106.250 147.300
##
## $n
## [1] 500
##
## $conf
## [1] 84.90602 89.16398
##
## $out
##  [1] 171.5 152.8 166.2 152.1 182.1 158.9 188.5 153.5 155.1 174.2 186.9
```

```
  cancer$perimeter_mean[cancer$perimeter_mean > 147.300]<- mean(cancer$perimeter_mean)
  cancer$perimeter_mean[cancer$perimeter_mean > 142.700]<- mean(cancer$perimeter_mean)
  cancer$perimeter_mean[cancer$perimeter_mean > 141.300]<- mean(cancer$perimeter_mean)
  boxplot.stats(cancer$perimeter_mean)
```

```
## $stats
## [1]  43.790  76.120  87.035 102.450 141.300
##
## $n
## [1] 500
##
## $conf
## [1] 85.17453 88.89547
##
## $out
## numeric(0)
```

```
  ## Variable 4

  boxplot.stats(cancer$area_mean) # out encima de 1335
```

```
## $stats
## [1]  143.50  431.95  558.65  801.55 1335.00
##
## $n
## [1] 500
##
## $conf
## [1] 532.5342 584.7658
##
```

```
## $out
##  [1] 1404 1878 1509 1761 1686 2250 1685 2499 1670 1364 1419 1491 1747 2010 1546
## [16] 1482 1386 1407 1384 2501
```

```
  cancer$area_mean[cancer$area_mean > 1335]<- mean(cancer$area_mean)
  cancer$area_mean[cancer$area_mean > 1174]<- mean(cancer$area_mean)
  cancer$area_mean[cancer$area_mean > 1007]<- mean(cancer$area_mean)
  cancer$area_mean[cancer$area_mean > 994]<- mean(cancer$area_mean)
  cancer$area_mean[cancer$area_mean > 963.700]<- mean(cancer$area_mean)
  cancer$area_mean[cancer$area_mean > 951.6]<- mean(cancer$area_mean)
  boxplot.stats(cancer$area_mean)
```

```
## $stats
## [1] 143.5000 431.9500 551.5204 640.4000 951.6000
##
## $n
## [1] 500
##
## $conf
## [1] 536.7914 566.2494
##
## $out
## numeric(0)
```

```
  ## Variable 5

  boxplot.stats(cancer$smoothness_mean) # out encima de 0.140
```

```
## $stats
## [1] 0.062510 0.086210 0.095955 0.105000 0.132600
##
## $n
## [1] 500
##
## $conf
## [1] 0.0946273 0.0972827
##
## $out
## [1] 0.1425 0.1398 0.1447 0.1335
```

```
  cancer$smoothness_mean[cancer$smoothness_mean > 0.13600]<- mean(cancer$smoothness_mean)
  cancer$smoothness_mean[cancer$smoothness_mean > 0.129100]<- mean(cancer$smoothness_mean)
  boxplot.stats(cancer$smoothness_mean)
```

```
## $stats
## [1] 0.062510 0.086210 0.095865 0.104400 0.129100
##
## $n
## [1] 500
##
## $conf
## [1] 0.0945797 0.0971503
```

```
##
## $out
## numeric(0)
```

```
  ## Variable 6

  boxplot.stats(cancer$compactness_mean) #out encima de 0.229301
```

```
## $stats
## [1] 0.019380 0.063735 0.092350 0.130400 0.229300
##
## $n
## [1] 500
##
## $conf
## [1] 0.08763947 0.09706053
##
## $out
##  [1] 0.2776 0.2839 0.2396 0.2458 0.3454 0.2665 0.2768 0.2867 0.2832 0.2413
## [11] 0.3114 0.2364 0.2363 0.2576
```

```
  cancer$compactness_mean[cancer$compactness_mean > 0.229300]<- mean(cancer$compactness_mean)
  cancer$compactness_mean[cancer$compactness_mean > 0.223300]<- mean(cancer$compactness_mean)
  cancer$compactness_mean[cancer$compactness_mean > 0.219000]<- mean(cancer$compactness_mean)
  boxplot.stats(cancer$compactness_mean)
```

```
## $stats
## [1] 0.019380 0.063735 0.092350 0.126450 0.219000
##
## $n
## [1] 500
##
## $conf
## [1] 0.08791857 0.09678143
##
## $out
## numeric(0)
```

```
  ## Variable 7

  boxplot.stats(cancer$concavity_mean) # out encima de 0.28100
```

```
## $stats
## [1] 0.00000 0.02932 0.06583 0.13220 0.28100
##
## $n
## [1] 500
##
## $conf
## [1] 0.05856053 0.07309947
##
## $out
```

```
##  [1] 0.3001 0.3130 0.3754 0.3339 0.4264 0.3003 0.4268 0.4108 0.2871 0.3523
## [11] 0.3201 0.3176 0.2914 0.3368 0.3189 0.3635
```

```
  cancer$concavity_mean[cancer$concavity_mean > 0.28100]<- mean(cancer$concavity_mean)
  cancer$concavity_mean[cancer$concavity_mean > 0.25080]<- mean(cancer$concavity_mean)
  cancer$concavity_mean[cancer$concavity_mean > 0.24170]<- mean(cancer$concavity_mean)
  cancer$concavity_mean[cancer$concavity_mean > 0.231900]<- mean(cancer$concavity_mean)
  boxplot.stats(cancer$concavity_mean)
```

```
## $stats
## [1] 0.00000 0.02932 0.06583 0.11265 0.23190
##
## $n
## [1] 500
##
## $conf
## [1] 0.05994192 0.07171808
##
## $out
## numeric(0)
```

```
  ## Variable 8

  boxplot.stats(cancer$concave.points_mean) # out encima de 0.150400
```

```
## $stats
## [1] 0.000000 0.020335 0.034840 0.074490 0.150400
##
## $n
## [1] 500
##
## $conf
## [1] 0.03101342 0.03866658
##
## $out
##  [1] 0.1604 0.1845 0.1823 0.2012 0.1878 0.1620 0.1595 0.1913 0.1562 0.1689
```

```
  cancer$concave.points_mean[cancer$concave.points_mean > 0.150400]<- mean(cancer$concave.points_mean)
  cancer$concave.points_mean[cancer$concave.points_mean > 0.141000]<- mean(cancer$concave.points_mean)
  cancer$concave.points_mean[cancer$concave.points_mean > 0.131000]<- mean(cancer$concave.points_mean)
  boxplot.stats(cancer$concave.points_mean)
```

```
## $stats
## [1] 0.000000 0.020335 0.034840 0.065995 0.131000
##
## $n
## [1] 500
##
## $conf
## [1] 0.03161368 0.03806632
##
## $out
## numeric(0)
```

```r
  ## Variable 9

  boxplot.stats(cancer$symmetry_mean) # out encima 0.24190
```

```
## $stats
## [1] 0.11670 0.16205 0.17995 0.19535 0.24190
##
## $n
## [1] 500
##
## $conf
## [1] 0.177597 0.182303
##
## $out
##  [1] 0.2597 0.2521 0.3040 0.2743 0.2906 0.2556 0.2655 0.2678 0.2540 0.2548
## [11] 0.2495 0.2595 0.2569 0.2459 0.2538
```

```r
  cancer$symmetry_mean[cancer$symmetry_mean > 0.24190]<- mean(cancer$symmetry_mean)
  cancer$symmetry_mean[cancer$symmetry_mean > 0.24030]<- mean(cancer$symmetry_mean)
  cancer$symmetry_mean[cancer$symmetry_mean > 0.23980]<- mean(cancer$symmetry_mean)
  cancer$symmetry_mean[cancer$symmetry_mean > 0.23950]<- mean(cancer$symmetry_mean)
  cancer$symmetry_mean[cancer$symmetry_mean > 0.23840]<- mean(cancer$symmetry_mean)
  boxplot.stats(cancer$symmetry_mean)
```

```
## $stats
## [1] 0.11670 0.16205 0.17955 0.19300 0.23840
##
## $n
## [1] 500
##
## $conf
## [1] 0.1773631 0.1817369
##
## $out
## numeric(0)
```

```r
  ## Variable 10

  boxplot.stats(cancer$fractal_dimension_mean) #out encima de 0.077690
```

```
## $stats
## [1] 0.049960 0.057585 0.061285 0.065675 0.077690
##
## $n
## [1] 500
##
## $conf
## [1] 0.06071336 0.06185664
##
## $out
##  [1] 0.07871 0.09744 0.08243 0.07800 0.07799 0.08046 0.08980 0.08142 0.07818
## [10] 0.07782 0.07839 0.08261 0.09296 0.08116 0.08104 0.08743 0.08450 0.07950
```

```
  cancer$fractal_dimension_mean[cancer$fractal_dimension_mean > 0.077690]<- mean(cancer$fractal_dimensi
  cancer$fractal_dimension_mean[cancer$fractal_dimension_mean > 0.075420]<- mean(cancer$fractal_dimensi
  cancer$fractal_dimension_mean[cancer$fractal_dimension_mean > 0.074690]<- mean(cancer$fractal_dimensi
  cancer$fractal_dimension_mean[cancer$fractal_dimension_mean > 0.074510]<- mean(cancer$fractal_dimensi
  boxplot.stats(cancer$fractal_dimension_mean)
```

```
## $stats
## [1] 0.049960 0.057585 0.061285 0.064360 0.074510
##
## $n
## [1] 500
##
## $conf
## [1] 0.06080628 0.06176372
##
## $out
## numeric(0)
```

```
  ## Variable 11

  boxplot.stats(cancer$radius_se) # out encima de 0.88110
```

```
## $stats
## [1] 0.11150 0.23410 0.32805 0.49545 0.88110
##
## $n
## [1] 500
##
## $conf
## [1] 0.3095831 0.3465169
##
## $out
##  [1] 1.0950 0.9555 1.0460 1.2140 0.9811 0.9806 0.9317 0.8973 1.2150 1.5090
## [11] 1.2960 1.0000 1.0880 2.8730 0.9553 1.0580 1.0040 1.2920 1.1720 1.1670
## [21] 1.1110 1.0720 1.0090 0.9948 0.9761 1.2070 1.0080 1.3700 0.9291 2.5470
## [31] 0.9289
```

```
  cancer$radius_se[cancer$radius_se > 0.88110]<- mean(cancer$radius_se)
  cancer$radius_se[cancer$radius_se > 0.71280]<- mean(cancer$radius_se)
  cancer$radius_se[cancer$radius_se > 0.6643000]<- mean(cancer$radius_se)
  boxplot.stats(cancer$radius_se)
```

```
## $stats
## [1] 0.1115000 0.2341000 0.3280500 0.4106343 0.6643000
##
## $n
## [1] 500
##
## $conf
## [1] 0.3155761 0.3405239
##
## $out
## numeric(0)
```

```
  ## Variable 12

  boxplot.stats(cancer$texture_se) #out encima de 2.3420
```

```
## $stats
## [1] 0.3602 0.8257 1.0880 1.4585 2.3420
##
## $n
## [1] 500
##
## $conf
## [1] 1.043287 1.132713
##
## $out
##  [1] 3.568 2.910 3.120 2.508 2.664 4.885 2.612 2.454 2.777 2.509 2.836 2.878
## [13] 2.542 2.426 2.643 3.647
```

```
  cancer$texture_se[cancer$texture_se > 2.3420]<- mean(cancer$texture_se)
  cancer$texture_se[cancer$texture_se > 2.2200]<- mean(cancer$texture_se)
  cancer$texture_se[cancer$texture_se > 2.2000]<- mean(cancer$texture_se)
  cancer$texture_se[cancer$texture_se > 2.1880]<- mean(cancer$texture_se)
  cancer$texture_se[cancer$texture_se > 2.1740]<- mean(cancer$texture_se)
  cancer$texture_se[cancer$texture_se > 2.1290]<- mean(cancer$texture_se)
  boxplot.stats(cancer$texture_se)
```

```
## $stats
## [1] 0.3602 0.8257 1.0880 1.3575 2.1290
##
## $n
## [1] 500
##
## $conf
## [1] 1.050423 1.125577
##
## $out
## numeric(0)
```

```
  ## Variable 13

  boxplot.stats(cancer$perimeter_se) #out encima de 6.0510
```

```
## $stats
## [1] 0.7570 1.6450 2.3275 3.4145 6.0510
##
## $n
## [1] 500
##
## $conf
## [1] 2.202468 2.452532
##
## $out
##  [1]  8.589 11.070  7.276  8.077  8.830  6.311  8.649  7.382 10.050  9.807
```

```
## [11]  8.419  6.971  7.337  7.029 21.980  6.487  7.247  6.372  7.158 10.120
## [21]  6.146  7.749  8.867  7.237  7.804  6.076  6.462  7.222  7.128  7.733
## [31]  7.561  9.424 18.650
```

```
  cancer$perimeter_se[cancer$perimeter_se > 6.0510]<- mean(cancer$perimeter_se)
  cancer$perimeter_se[cancer$perimeter_se > 5.0290]<- mean(cancer$perimeter_se)
  cancer$perimeter_se[cancer$perimeter_se > 4.78200]<- mean(cancer$perimeter_se)
  boxplot.stats(cancer$perimeter_se)
```

```
## $stats
## [1] 0.757000 1.645000 2.327500 2.903064 4.782000
##
## $n
## [1] 500
##
## $conf
## [1] 2.238606 2.416394
##
## $out
## numeric(0)
```

```
  ## Variable 14

  boxplot.stats(cancer$area_se) #out encima de 89.740
```

```
## $stats
## [1]  6.802 18.030 24.970 46.875 89.740
##
## $n
## [1] 500
##
## $conf
## [1] 22.93182 27.00818
##
## $out
##  [1] 153.40  94.03  94.44 116.20 112.40  93.99 102.60 111.40  93.54 105.00
## [11] 106.00 104.90  98.81 102.50  96.05 134.80 116.40 120.00 170.00  90.47
## [21] 233.00 101.90  93.91 119.30  97.07  97.85 122.30 128.70 111.70 525.60
## [31] 124.40 109.90 155.80 137.90  92.81 106.40 138.50  90.94 199.70 156.80
## [41] 133.00 130.80 164.10 153.10 103.60 224.10 130.20 176.50 103.90 115.20
## [51] 542.20 104.90  95.77
```

```
  cancer$area_se[cancer$area_se > 89.740]<- mean(cancer$area_se)
  cancer$area_se[cancer$area_se > 75.09000]<- mean(cancer$area_se)
  boxplot.stats(cancer$area_se)
```

```
## $stats
## [1]  6.80200 18.03000 24.97000 41.13012 75.09000
##
## $n
## [1] 500
##
```

```
## $conf
## [1] 23.33775 26.60225
##
## $out
## numeric(0)
```

```
  ## Variable 15

  boxplot.stats(cancer$smoothness_se) #out encima de 0.0119300
```

```
## $stats
## [1] 0.0017130 0.0051455 0.0062545 0.0079750 0.0119300
##
## $n
## [1] 500
##
## $conf
## [1] 0.006054568 0.006454432
##
## $out
##  [1] 0.01721 0.01340 0.01385 0.01291 0.01835 0.02333 0.01243 0.01496 0.01286
## [10] 0.01439 0.01380 0.01345 0.03113 0.01604 0.01236 0.01380 0.01418 0.01574
## [19] 0.02075 0.01289 0.01736 0.01582 0.01474 0.01307 0.01459
```

```
  cancer$smoothness_se[cancer$smoothness_se > 0.0119300]<- mean(cancer$smoothness_se)
  cancer$smoothness_se[cancer$smoothness_se > 0.0109800]<- mean(cancer$smoothness_se)
  cancer$smoothness_se[cancer$smoothness_se > 0.0107500]<- mean(cancer$smoothness_se)
  cancer$smoothness_se[cancer$smoothness_se > 0.0106100]<- mean(cancer$smoothness_se)
  cancer$smoothness_se[cancer$smoothness_se < 0.001714]<- mean(cancer$smoothness_se)
  cancer$smoothness_se[cancer$smoothness_se > 0.0105600]<- mean(cancer$smoothness_se)
  boxplot.stats(cancer$smoothness_se)
```

```
## $stats
## [1] 0.0026670 0.0051635 0.0062765 0.0073355 0.0105600
##
## $n
## [1] 500
##
## $conf
## [1] 0.006123027 0.006429973
##
## $out
## numeric(0)
```

```
   ## Variable 16

  boxplot.stats(cancer$compactness_se) #out encima de 0.06630
```

```
## $stats
## [1] 0.003012 0.013015 0.020350 0.032485 0.060630
##
## $n
```

```
## [1] 500
##
## $conf
## [1] 0.01897425 0.02172575
##
## $out
##  [1] 0.07458 0.07217 0.08297 0.10060 0.07056 0.08606 0.09368 0.06835 0.08668
## [10] 0.07446 0.06760 0.09806 0.09586 0.08808 0.13540 0.08555 0.08262 0.10640
## [19] 0.06590 0.06559 0.07643 0.06669 0.06213 0.06657 0.07025 0.07471
```

```
  cancer$compactness_se[cancer$compactness_se > 0.060630]<- mean(cancer$compactness_se)
  cancer$compactness_se[cancer$compactness_se > 0.054700]<- mean(cancer$compactness_se)
  cancer$compactness_se[cancer$compactness_se > 0.051560]<- mean(cancer$compactness_se)
  cancer$compactness_se[cancer$compactness_se > 0.050570]<- mean(cancer$compactness_se)
  cancer$compactness_se[cancer$compactness_se > 0.049600]<- mean(cancer$compactness_se)
  boxplot.stats(cancer$compactness_se)
```

```
## $stats
## [1] 0.003012 0.013015 0.020350 0.027720 0.049600
##
## $n
## [1] 500
##
## $conf
## [1] 0.01931095 0.02138905
##
## $out
## numeric(0)
```

```
   ## Variable 17

  boxplot.stats(cancer$concavity_se) #out encima de 0.082320
```

```
## $stats
## [1] 0.000000 0.015100 0.025985 0.042420 0.082320
##
## $n
## [1] 500
##
## $conf
## [1] 0.02405458 0.02791542
##
## $out
##  [1] 0.08890 0.09723 0.30380 0.10910 0.10400 0.14350 0.09263 0.12780 0.39600
## [10] 0.11970 0.11660 0.08958 0.14380 0.08880 0.09518 0.09960 0.10270 0.09953
## [19] 0.15350 0.09472 0.11140
```

```
  cancer$concavity_se[cancer$concavity_se > 0.08230]<- mean(cancer$concavity_se)
  cancer$concavity_se[cancer$concavity_se > 0.068990]<- mean(cancer$concavity_se)
  cancer$concavity_se[cancer$concavity_se > 0.065770]<- mean(cancer$concavity_se)
  cancer$concavity_se[cancer$concavity_se > 0.063890]<- mean(cancer$concavity_se)
  cancer$concavity_se[cancer$concavity_se > 0.063290]<- mean(cancer$concavity_se)
```

```
  cancer$concavity_se[cancer$concavity_se > 0.062710]<- mean(cancer$concavity_se)
  cancer$concavity_se[cancer$concavity_se > 0.061650]<- mean(cancer$concavity_se)
  boxplot.stats(cancer$concavity_se)
```

```
## $stats
## [1] 0.000000 0.015100 0.025985 0.033900 0.061650
##
## $n
## [1] 500
##
## $conf
## [1] 0.0246566 0.0273134
##
## $out
## numeric(0)
```

```
   ## Variable 18

  boxplot.stats(cancer$concave.points_se) #out encima de 0.024800
```

```
## $stats
## [1] 0.000000 0.007631 0.010905 0.014620 0.024800
##
## $n
## [1] 500
##
## $conf
## [1] 0.01041116 0.01139884
##
## $out
##  [1] 0.04090 0.02638 0.03322 0.02593 0.02801 0.05279 0.02794 0.02765 0.03927
## [10] 0.03024 0.03487 0.02771 0.02527 0.02536 0.02919 0.03441 0.02598 0.02721
```

```
  cancer$concave.points_se[cancer$concave.points_se > 0.024800]<- mean(cancer$concave.points_se)
  cancer$concave.points_se[cancer$concave.points_se > 0.023110]<- mean(cancer$concave.points_se)
  cancer$concave.points_se[cancer$concave.points_se > 0.022580]<- mean(cancer$concave.points_se)
  cancer$concave.points_se[cancer$concave.points_se > 0.022340]<- mean(cancer$concave.points_se)
  cancer$concave.points_se[cancer$concave.points_se > 0.02215000]<- mean(cancer$concave.points_se)
  boxplot.stats(cancer$concave.points_se)
```

```
## $stats
## [1] 0.00000000 0.00763100 0.01087704 0.01347500 0.02215000
##
## $n
## [1] 500
##
## $conf
## [1] 0.01046410 0.01128997
##
## $out
## numeric(0)
```

```
  ## Variable 19

boxplot.stats(cancer$symmetry_se) #out encima de 0.035460
```

```
## $stats
## [1] 0.007882 0.015025 0.018795 0.023535 0.035460
##
## $n
## [1] 500
##
## $conf
## [1] 0.01819369 0.01939631
##
## $out
##  [1] 0.05963 0.04484 0.03672 0.05333 0.04183 0.04192 0.04197 0.07895 0.05014
## [10] 0.04547 0.05168 0.05628 0.03880 0.05113 0.03799 0.04783 0.04499 0.04077
## [19] 0.06146 0.04022 0.04243 0.03756 0.03675 0.05543 0.03710
```

```
cancer$symmetry_se[cancer$symmetry_se > 0.035460]<- mean(cancer$symmetry_se)
cancer$symmetry_se[cancer$symmetry_se > 0.032320]<- mean(cancer$symmetry_se)
cancer$symmetry_se[cancer$symmetry_se > 0.031270]<- mean(cancer$symmetry_se)
cancer$symmetry_se[cancer$symmetry_se > 0.030030]<- mean(cancer$symmetry_se)
cancer$symmetry_se[cancer$symmetry_se > 0.02951]<- mean(cancer$symmetry_se)
cancer$symmetry_se[cancer$symmetry_se > 0.02897]<- mean(cancer$symmetry_se)
boxplot.stats(cancer$symmetry_se)
```

```
## $stats
## [1] 0.00788200 0.01502500 0.01871000 0.02068905 0.02897000
##
## $n
## [1] 500
##
## $conf
## [1] 0.01830978 0.01911022
##
## $out
## numeric(0)
```

```
  ## Variable 20

boxplot.stats(cancer$fractal_dimension_se) #out encima de 0.0077310
```

```
## $stats
## [1] 0.0008948 0.0022050 0.0031195 0.0044695 0.0077310
##
## $n
## [1] 500
##
## $conf
## [1] 0.002959491 0.003279509
##
## $out
```

```
##  [1] 0.009208 0.010080 0.012840 0.008093 0.009559 0.021930 0.010390 0.012980
##  [9] 0.009875 0.009423 0.009368 0.011780 0.029840 0.017920 0.011720 0.012560
## [17] 0.008675 0.008660 0.008015 0.022860 0.007877 0.012200 0.012330 0.008925
## [25] 0.008133 0.011300 0.009627
```

```
  cancer$fractal_dimension_se[cancer$fractal_dimension_se > 0.0077310]<- mean(cancer$fractal_dimension_s
  boxplot.stats(cancer$fractal_dimension_se)
```

```
## $stats
## [1] 0.0008948 0.0022050 0.0031195 0.0041040 0.0068840
##
## $n
## [1] 500
##
## $conf
## [1] 0.002985317 0.003253683
##
## $out
##  [1] 0.007444 0.007646 0.007054 0.007098 0.007555 0.007259 0.007610 0.007596
##  [9] 0.007330 0.007358 0.007731
```

```
  cancer$fractal_dimension_se[cancer$fractal_dimension_se > 0.0068840]<- mean(cancer$fractal_dimension_s
  boxplot.stats(cancer$fractal_dimension_se)
```

```
## $stats
## [1] 0.0008948 0.0022050 0.0031195 0.0040005 0.0065170
##
## $n
## [1] 500
##
## $conf
## [1] 0.00299263 0.00324637
##
## $out
## [1] 0.006792 0.006820 0.006758 0.006884 0.006736
```

```
  cancer$fractal_dimension_se[cancer$fractal_dimension_se > 0.0065170]<- mean(cancer$fractal_dimension_s
  boxplot.stats(cancer$fractal_dimension_se)
```

```
## $stats
## [1] 0.0008948 0.0022050 0.0031195 0.0039230 0.0063550
##
## $n
## [1] 500
##
## $conf
## [1] 0.002998107 0.003240893
##
## $out
## [1] 0.006517
```

```
  cancer$fractal_dimension_se[cancer$fractal_dimension_se > 0.0063550]<- mean(cancer$fractal_dimension_s
  boxplot.stats(cancer$fractal_dimension_se)
```

```
## $stats
## [1] 0.0008948 0.0022050 0.0031195 0.0039045 0.0063550
##
## $n
## [1] 500
##
## $conf
## [1] 0.002999414 0.003239586
##
## $out
## numeric(0)
```

```
   ## Variable 21

  boxplot.stats(cancer$radius_worst) #out encima de 26.730 y por debajo de 7.930
```

```
## $stats
## [1]  7.930 12.840 14.845 18.500 26.730
##
## $n
## [1] 500
##
## $conf
## [1] 14.44507 15.24493
##
## $out
##  [1]  0.00  0.00 29.17  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
## [13]  0.00 28.40  0.00  0.00  0.00  0.00  0.00 28.01  0.00 28.11 27.90 31.01
## [25] 32.49 28.19 30.67 33.13 30.75 27.66 36.04  0.00  0.00
```

```
  cancer$radius_worst[cancer$radius_worst > 26.730] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst < 7.930] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst > 24.560] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst > 22.750] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst > 20.880] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst > 19.96000] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst < 9.41400] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst > 19.2800] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst < 9.56500] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst < 9.62800] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst < 9.69900] <- mean(cancer$radius_worst)
  cancer$radius_worst[cancer$radius_worst > 19.200] <- mean(cancer$radius_worst)
  boxplot.stats(cancer$radius_worst)
```

```
## $stats
## [1]  9.69900 13.24500 14.71110 15.63203 19.20000
##
## $n
## [1] 500
```

```
##
## $conf
## [1] 14.54244 14.87977
##
## $out
## numeric(0)
```

```
   ## Variable 22

  boxplot.stats(cancer$texture_worst) #out encima de 40.68 y de bajo de 12.0200
```

```
## $stats
## [1] 12.02000 21.04500 25.39472 29.25500 40.68000
##
## $n
## [1] 500
##
## $conf
## [1] 24.81461 25.97484
##
## $out
##  [1]  0.00  0.00 41.85 45.41 41.78 44.87 49.54 47.16  0.00 41.61
```

```
  cancer$texture_worst[cancer$texture_worst > 40.6800]<- mean(cancer$texture_worst)
  cancer$texture_worst[cancer$texture_worst < 12.0200]<- mean(cancer$texture_worst)
  boxplot.stats(cancer$texture_worst)
```

```
## $stats
## [1] 12.02000 21.14000 25.39472 28.98000 40.68000
##
## $n
## [1] 500
##
## $conf
## [1] 24.84075 25.94870
##
## $out
## numeric(0)
```

```
   ## Variable 23

   boxplot.stats(cancer$perimeter_worst) #out encima de 186.80 y de bajo 50.41
```

```
## $stats
## [1]  50.41  83.64  97.18 125.25 186.80
##
## $n
## [1] 500
##
## $conf
## [1]  94.23985 100.12015
##
```

```
## $out
##  [1]   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0 211.7
## [13]   0.0   0.0 206.8   0.0   0.0   0.0   0.0   0.0 220.8 188.5 206.0 214.0
## [25] 195.9 202.4 229.3 199.5 195.0 251.2
```

```
  cancer$perimeter_worst[cancer$perimeter_worst > 186.80]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst < 50.41]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst > 174.90]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst > 166.80]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst > 160.50]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst > 153.90]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst > 146.60]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst > 136.10]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst > 132.900]<- mean(cancer$perimeter_worst)
  cancer$perimeter_worst[cancer$perimeter_worst < 56.65]<- mean(cancer$perimeter_worst)
  boxplot.stats(cancer$perimeter_worst)
```

```
## $stats
## [1]  56.65000  85.07000  98.37904 104.22315 132.90000
##
## $n
## [1] 500
##
## $conf
## [1] 97.02569 99.73240
##
## $out
## numeric(0)
```

```
   ## Variable 24

  boxplot.stats(cancer$area_worst) #out encima de 2089.00
```

```
## $stats
## [1]    0.00  521.60  691.75 1151.50 2089.00
##
## $n
## [1] 500
##
## $conf
## [1] 647.2414 736.2586
##
## $out
##  [1] 2398 2615 2215 2145 2562 2360 2232 2403 3216 2499 2477 2944 3432 2384 2906
## [16] 3234 3143 2227 4254
```

```
  cancer$area_worst[cancer$area_worst > 2089]<- mean(cancer$area_worst)
  cancer$area_worst[cancer$area_worst > 1688]<- mean(cancer$area_worst)
  cancer$area_worst[cancer$area_worst > 1437]<- mean(cancer$area_worst)
  cancer$area_worst[cancer$area_worst > 1292]<- mean(cancer$area_worst)
  cancer$area_worst[cancer$area_worst < 185.2]<- mean(cancer$area_worst)
  cancer$area_worst[cancer$area_worst > 1272]<- mean(cancer$area_worst)
  boxplot.stats(cancer$area_worst)
```

```
## $stats
## [1]  185.2000  523.5500  688.5287  824.2569 1272.0000
##
## $n
## [1] 500
##
## $conf
## [1] 667.2808 709.7765
##
## $out
## numeric(0)
```

```
  ## Variable 25

boxplot.stats(cancer$smoothness_worst) #out encima de 0.18830
```

```
## $stats
## [1] 0.08125 0.11640 0.13130 0.14600 0.18830
##
## $n
## [1] 500
##
## $conf
## [1] 0.1292085 0.1333915
##
## $out
## [1] 0.20980 0.19090 0.07117 0.22260 0.21840
```

```
cancer$smoothness_worst[cancer$smoothness_worst > 0.18830]<- mean(cancer$smoothness_worst)
cancer$smoothness_worst[cancer$smoothness_worst < 0.08125]<- mean(cancer$smoothness_worst)
cancer$smoothness_worst[cancer$smoothness_worst > 0.18730]<- mean(cancer$smoothness_worst)
cancer$smoothness_worst[cancer$smoothness_worst > 0.18620]<- mean(cancer$smoothness_worst)
boxplot.stats(cancer$smoothness_worst)
```

```
## $stats
## [1] 0.08125 0.11660 0.13130 0.14460 0.18620
##
## $n
## [1] 500
##
## $conf
## [1] 0.1293215 0.1332785
##
## $out
## numeric(0)
```

```
  ## Variable 26

boxplot.stats(cancer$compactness_worst) #out encima de 0.62470
```

```
## $stats
## [1] 0.02729 0.14585 0.21685 0.34245 0.62470
```

```
##
## $n
## [1] 500
##
## $conf
## [1] 0.2029583 0.2307417
##
## $out
##  [1] 0.6656 0.8663 1.0580 0.7725 0.6577 0.6643 0.6590 0.7444 0.7394 0.6997
## [11] 0.7584 0.9327 0.9379 0.7090
```

```
  cancer$compactness_worst[cancer$compactness_worst > 0.62470]<- mean(cancer$compactness_worst)
  cancer$smoothness_worst[cancer$smoothness_worst > 0.57750]<- mean(cancer$smoothness_worst)
  boxplot.stats(cancer$smoothness_worst)
```

```
## $stats
## [1] 0.08125 0.11660 0.13130 0.14460 0.18620
##
## $n
## [1] 500
##
## $conf
## [1] 0.1293215 0.1332785
##
## $out
## numeric(0)
```

```
   ## Variable 27

  boxplot.stats(cancer$concavity_worst) #out encima de 0.78920
```

```
## $stats
## [1] 0.00000 0.11445 0.23140 0.39000 0.78920
##
## $n
## [1] 500
##
## $conf
## [1] 0.2119297 0.2508703
##
## $out
## [1] 1.1050 1.2520 0.9608 0.8216 0.8488 0.8489 0.8402 0.9034 0.9019
```

```
  cancer$concavity_worst[cancer$concavity_worst > 0.78920]<- mean(cancer$concavity_worst)
  cancer$concavity_worst[cancer$concavity_worst > 0.7727]<- mean(cancer$concavity_worst)
  boxplot.stats(cancer$concavity_worst)
```

```
## $stats
## [1] 0.00000 0.11445 0.23140 0.37895 0.77270
##
## $n
## [1] 500
```

```
##
## $conf
## [1] 0.2127105 0.2500895
##
## $out
## numeric(0)
```

```
   ## Variable 29

  boxplot.stats(cancer$concave.points_worst) #no tiene out
```

```
## $stats
## [1] 0.000000 0.063255 0.100650 0.167000 0.291000
##
## $n
## [1] 500
##
## $conf
## [1] 0.0933194 0.1079806
##
## $out
## numeric(0)
```

```
   ## Variable 30

  boxplot.stats(cancer$symmetry_worst) #out encima de 0.4228
```

```
## $stats
## [1] 0.1565 0.2516 0.2831 0.3201 0.4228
##
## $n
## [1] 500
##
## $conf
## [1] 0.2782598 0.2879402
##
## $out
##  [1] 0.4601 0.6638 0.4378 0.4366 0.4667 0.4264 0.4761 0.4270 0.4863 0.4670
## [11] 0.5440 0.4882 0.5774 0.5166 0.4753 0.4432 0.4724 0.5558 0.4245 0.4824
## [21] 0.4677
```

```
  cancer$symmetry_worst[cancer$symmetry_worst > 0.4228]<- mean(cancer$symmetry_worst)
  cancer$symmetry_worst[cancer$symmetry_worst > 0.40450]<- mean(cancer$symmetry_worst)
  cancer$symmetry_worst[cancer$symmetry_worst > 0.39930]<- mean(cancer$symmetry_worst)
  cancer$symmetry_worst[cancer$symmetry_worst < 0.1648]<- mean(cancer$symmetry_worst)
  cancer$symmetry_worst[cancer$symmetry_worst > 0.39560]<- mean(cancer$symmetry_worst)
  cancer$symmetry_worst[cancer$symmetry_worst < 0.16520]<- mean(cancer$symmetry_worst)
  cancer$symmetry_worst[cancer$symmetry_worst < 0.17120]<- mean(cancer$symmetry_worst)
  boxplot.stats(cancer$symmetry_worst)
```

```
## $stats
## [1] 0.17120 0.25260 0.28265 0.31030 0.39560
```

```
##
## $n
## [1] 500
##
## $conf
## [1] 0.2785729 0.2867271
##
## $out
## numeric(0)
```

```
  ## Variable 31

  boxplot.stats(cancer$fractal_dimension_worst) #out encima de 0.12240
```

```
## $stats
## [1] 0.05504 0.07105 0.07996 0.09186 0.12240
##
## $n
## [1] 500
##
## $conf
## [1] 0.07848957 0.08143043
##
## $out
##  [1] 0.1730 0.1244 0.2075 0.1431 0.1341 0.1275 0.1402 0.1233 0.1339 0.1405
## [11] 0.1252 0.1486 0.1259 0.1284 0.1446 0.1243 0.0000 0.1297 0.0000 0.1297
## [21] 0.0000 0.0000 0.1403 0.1249
```

```
  cancer$fractal_dimension_worst[cancer$fractal_dimension_worst > 0.12240]<- mean(cancer$fractal_dimens
  cancer$fractal_dimension_worst[cancer$fractal_dimension_worst > 0.1155]<- mean(cancer$fractal_dimensi
  cancer$fractal_dimension_worst[cancer$fractal_dimension_worst > 0.1123]<- mean(cancer$fractal_dimensi
  cancer$fractal_dimension_worst[cancer$fractal_dimension_worst > 0.11080]<- mean(cancer$fractal_dimens
  cancer$fractal_dimension_worst[cancer$fractal_dimension_worst < 0.05504]<- mean(cancer$fractal_dimens
  cancer$fractal_dimension_worst[cancer$fractal_dimension_worst > 0.1094]<- mean(cancer$fractal_dimensi
  boxplot.stats(cancer$fractal_dimension_worst)
```

```
## $stats
## [1] 0.055040 0.071365 0.079900 0.087010 0.109400
##
## $n
## [1] 500
##
## $conf
## [1] 0.07879453 0.08100547
##
## $out
## numeric(0)
```

9. Una vez realice los puntos anteriores, realice una resumen de sus datos y compruebe que su dataset está lista para ser manipulada (use la función summary u otra para este caso).

```r
## Analizamo el data set con la funcion summary  para ver si ya se puede tratar que no  tenga outlide

summary(cancer)
```

```
##        id              diagnosis    radius_mean      texture_mean
##  Min.   :     8670   Benigno:305   Min.   : 8.597   Min.   : 9.71
##  1st Qu.:   866704   Maligno:195   1st Qu.:11.848   1st Qu.:16.07
##  Median :   901432                 Median :13.253   Median :18.70
##  Mean   : 32630491                 Mean   :13.106   Mean   :18.87
##  3rd Qu.:  8910808                 3rd Qu.:14.205   3rd Qu.:21.44
##  Max.   :911320502                 Max.   :17.750   Max.   :29.29
##  perimeter_mean     area_mean     smoothness_mean   compactness_mean
##  Min.   : 43.79   Min.   :143.5   Min.   :0.06251   Min.   :0.01938
##  1st Qu.: 76.13   1st Qu.:432.0   1st Qu.:0.08629   1st Qu.:0.06374
##  Median : 87.03   Median :551.5   Median :0.09587   Median :0.09235
##  Mean   : 90.54   Mean   :542.9   Mean   :0.09553   Mean   :0.09807
##  3rd Qu.:102.42   3rd Qu.:640.2   3rd Qu.:0.10440   3rd Qu.:0.12632
##  Max.   :141.30   Max.   :951.6   Max.   :0.12910   Max.   :0.21900
##  concavity_mean    concave.points_mean symmetry_mean    fractal_dimension_mean
##  Min.   :0.00000   Min.   :0.00000     Min.   :0.1167   Min.   :0.04996
##  1st Qu.:0.02940   1st Qu.:0.02035     1st Qu.:0.1621   1st Qu.:0.05761
##  Median :0.06583   Median :0.03484     Median :0.1795   Median :0.06128
##  Mean   :0.07815   Mean   :0.04536     Mean   :0.1784   Mean   :0.06143
##  3rd Qu.:0.11263   3rd Qu.:0.06598     3rd Qu.:0.1930   3rd Qu.:0.06434
##  Max.   :0.23190   Max.   :0.13100     Max.   :0.2384   Max.   :0.07451
##    radius_se        texture_se      perimeter_se      area_se
##  Min.   :0.1115   Min.   :0.3602   Min.   :0.757   Min.   : 6.802
##  1st Qu.:0.2343   1st Qu.:0.8261   1st Qu.:1.647   1st Qu.:18.035
##  Median :0.3281   Median :1.0880   Median :2.328   Median :24.970
##  Mean   :0.3341   Mean   :1.1204   Mean   :2.376   Mean   :29.266
##  3rd Qu.:0.4106   3rd Qu.:1.3558   3rd Qu.:2.903   3rd Qu.:41.130
##  Max.   :0.6643   Max.   :2.1290   Max.   :4.782   Max.   :75.090
##  smoothness_se      compactness_se      concavity_se     concave.points_se
##  Min.   :0.002667   Min.   :0.003012   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.005166   1st Qu.:0.013048   1st Qu.:0.01510   1st Qu.:0.007634
##  Median :0.006276   Median :0.020350   Median :0.02599   Median :0.010877
##  Mean   :0.006324   Mean   :0.021484   Mean   :0.02597   Mean   :0.010814
##  3rd Qu.:0.007335   3rd Qu.:0.027720   3rd Qu.:0.03390   3rd Qu.:0.013463
##  Max.   :0.010560   Max.   :0.049600   Max.   :0.06165   Max.   :0.022150
##   symmetry_se       fractal_dimension_se  radius_worst    texture_worst
##  Min.   :0.007882   Min.   :0.0008948    Min.   : 9.699   Min.   :12.02
##  1st Qu.:0.015027   1st Qu.:0.0022050    1st Qu.:13.248   1st Qu.:21.16
##  Median :0.018710   Median :0.0031195    Median :14.711   Median :25.39
##  Mean   :0.018426   Mean   :0.0031957    Mean   :14.484   Mean   :25.28
##  3rd Qu.:0.020689   3rd Qu.:0.0039003    3rd Qu.:15.632   3rd Qu.:28.96
##  Max.   :0.028970   Max.   :0.0063550    Max.   :19.200   Max.   :40.68
##  perimeter_worst    area_worst     smoothness_worst  compactness_worst
##  Min.   : 56.65   Min.   : 185.2   Min.   :0.08125   Min.   :0.02729
##  1st Qu.: 85.07   1st Qu.: 523.6   1st Qu.:0.11660   1st Qu.:0.14592
##  Median : 98.38   Median : 688.5   Median :0.13130   Median :0.21685
##  Mean   : 95.91   Mean   : 685.8   Mean   :0.13115   Mean   :0.24196
##  3rd Qu.:104.22   3rd Qu.: 824.3   3rd Qu.:0.14460   3rd Qu.:0.31805
##  Max.   :132.90   Max.   :1272.0   Max.   :0.18620   Max.   :0.62470
```

```
##  concavity_worst  concave.points_worst symmetry_worst   fractal_dimension_worst
##  Min.   :0.0000   Min.   :0.0000       Min.   :0.1712   Min.   :0.05504
##  1st Qu.:0.1145   1st Qu.:0.0633       1st Qu.:0.2526   1st Qu.:0.07141
##  Median :0.2314   Median :0.1007       Median :0.2827   Median :0.07990
##  Mean   :0.2634   Mean   :0.1160       Mean   :0.2823   Mean   :0.08029
##  3rd Qu.:0.3789   3rd Qu.:0.1668       3rd Qu.:0.3103   3rd Qu.:0.08701
##  Max.   :0.7727   Max.   :0.2910       Max.   :0.3956   Max.   :0.10940
```

```
  # ya se han eliminados todos los outlider y no tiene datos NA en data set
```

10.La imputación de los valores faltantes o nulos afecto las distribución de los datos ? Justifique su repuesta.

R// La imputacion de los datos faltantes o nulo en el data set, si afecta la distribucion de los datos, ya que van variando las medias de los valores en cada variable respecto al data set original

Paso 4: Transformación de los datos **(+10 puntos)**

11. Elimine la variable (ID), debido a que solo es el número de identificación del paciente

```
  # se elimina la variable id
   cancer <- cancer %>%
    dplyr::select(-id)
```

12. Como puede observar, Las variables del data set tienen rangos numéricos muy diferentes, esto es, por ejemplo: la variable area_mean tiene valores entre (143.5 a 2501) y la variable symmetry_mean está entre (0.1167 - 0.30). En estos casos se debe realizar la estandarización de datos, es decir: i-Normalización y ii) escalamiento. Para el dataset realice la estandarizacion de los datos del dataset. Puede usar la función standardize o revisar esta información

```
## Datos de las variable area mean y symetry mean
   summary(cancer$area_mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   143.5   432.0   551.5   542.9   640.2   951.6
```

```
   summary(cancer$symmetry_mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1167  0.1621  0.1795  0.1784  0.1930  0.2384
```

```
     # normalizacion de las variables con media de 0 y deviacion de 1
  cancer <- cancer %>% mutate_at (c ('radius_mean','texture_mean','perimeter_mean',
                                     'area_mean','smoothness_mean','compactness_mean',
                                     'symmetry_mean','concavity_mean','concave.points_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','compactness_worst',
                                     'concave.points_worst','fractal_dimension_worst'),
                                     ~ ( scale (.)%>% as.vector ))
  summary(cancer) # analisis de los datos despues de la estandarizacion
```

```
##    diagnosis    radius_mean        texture_mean     perimeter_mean
##  Benigno:305   Min.   :-2.37384   Min.   :-2.4289   Min.   :-2.2998
##  Maligno:195   1st Qu.:-0.66243   1st Qu.:-0.7432   1st Qu.:-0.7090
##                Median : 0.07745   Median :-0.0461   Median :-0.1726
##                Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##                3rd Qu.: 0.57881   3rd Qu.: 0.6795   3rd Qu.: 0.5845
##                Max.   : 2.44528   Max.   : 2.7608   Max.   : 2.4967
##    area_mean        smoothness_mean    compactness_mean  concavity_mean
##  Min.   :-2.56238   Min.   :-2.57669   Min.   :-1.8544   Min.   :-1.3169
##  1st Qu.:-0.71161   1st Qu.:-0.72115   1st Qu.:-0.8091   1st Qu.:-0.8215
##  Median : 0.05536   Median : 0.02599   Median :-0.1349   Median :-0.2076
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.62462   3rd Qu.: 0.69197   3rd Qu.: 0.6657   3rd Qu.: 0.5810
##  Max.   : 2.62215   Max.   : 2.61929   Max.   : 2.8496   Max.   : 2.5910
##  concave.points_mean symmetry_mean      fractal_dimension_mean
##  Min.   :-1.4309     Min.   :-2.73656   Min.   :0.04996
##  1st Qu.:-0.7891     1st Qu.:-0.72497   1st Qu.:0.05761
##  Median :-0.3319     Median : 0.04974   Median :0.06128
##  Mean   : 0.0000     Mean   : 0.00000   Mean   :0.06143
##  3rd Qu.: 0.6504     3rd Qu.: 0.64601   3rd Qu.:0.06434
##  Max.   : 2.7013     Max.   : 2.65870   Max.   :0.07451
##    radius_se         texture_se        perimeter_se         area_se
##  Min.   :-1.7797   Min.   :-1.96198   Min.   :-1.81443   Min.   :-1.5037
##  1st Qu.:-0.7985   1st Qu.:-0.75949   1st Qu.:-0.81724   1st Qu.:-0.7518
##  Median :-0.0486   Median :-0.08352   Median :-0.05479   Median :-0.2875
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.6116   3rd Qu.: 0.60754   3rd Qu.: 0.59009   3rd Qu.: 0.7942
##  Max.   : 2.6395   Max.   : 2.60331   Max.   : 2.69532   Max.   : 3.0675
##  smoothness_se      compactness_se     concavity_se        concave.points_se
##  Min.   :-2.16050   Min.   :-1.7717   Min.   :-1.7997440   Min.   :-2.5034
##  1st Qu.:-0.68393   1st Qu.:-0.8092   1st Qu.:-0.7534255   1st Qu.:-0.7361
##  Median :-0.02799   Median :-0.1088   Median : 0.0008247   Median : 0.0145
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000000   Mean   : 0.0000
##  3rd Qu.: 0.59723   3rd Qu.: 0.5981   3rd Qu.: 0.5489293   3rd Qu.: 0.6130
##  Max.   : 2.50272   Max.   : 2.6965   Max.   : 2.4721458   Max.   : 2.6241
##   symmetry_se       fractal_dimension_se  radius_worst     texture_worst
##  Min.   :-2.45362   Min.   :-1.88696     Min.   :-2.5505   Min.   :-2.3892
##  1st Qu.:-0.79082   1st Qu.:-0.81249     1st Qu.:-0.6592   1st Qu.:-0.7419
##  Median : 0.06612   Median :-0.06253     Median : 0.1209   Median : 0.0213
##  Mean   : 0.00000   Mean   : 0.00000     Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.52665   3rd Qu.: 0.57775     3rd Qu.: 0.6117   3rd Qu.: 0.6639
##  Max.   : 2.45368   Max.   : 2.59085     Max.   : 2.5134   Max.   : 2.7762
##  perimeter_worst     area_worst       smoothness_worst  compactness_worst
##  Min.   :-2.5081   Min.   :-2.27120   Min.   :0.08125   Min.   :-1.6453
##  1st Qu.:-0.6924   1st Qu.:-0.73565   1st Qu.:0.11660   1st Qu.:-0.7360
##  Median : 0.1579   Median : 0.01257   Median :0.13130   Median :-0.1925
##  Mean   : 0.0000   Mean   : 0.00000   Mean   :0.13115   Mean   : 0.0000
##  3rd Qu.: 0.5312   3rd Qu.: 0.62842   3rd Qu.:0.14460   3rd Qu.: 0.5832
##  Max.   : 2.3633   Max.   : 2.65997   Max.   :0.18620   Max.   : 2.9333
##  concavity_worst  concave.points_worst symmetry_worst   fractal_dimension_worst
##  Min.   :0.0000   Min.   :-1.7602      Min.   :0.1712   Min.   :-2.14468
##  1st Qu.:0.1145   1st Qu.:-0.7996      1st Qu.:0.2526   1st Qu.:-0.75380
##  Median :0.2314   Median :-0.2328      Median :0.2827   Median :-0.03277
##  Mean   :0.2634   Mean   : 0.0000      Mean   :0.2823   Mean   : 0.00000
```

```
##  3rd Qu.:0.3789   3rd Qu.: 0.7718      3rd Qu.:0.3103   3rd Qu.: 0.57124
##  Max.   :0.7727   Max.   : 2.6558      Max.   :0.3956   Max.   : 2.47331
```

```
  sd(cancer$radius_mean) # observacion de la desviciancion
```

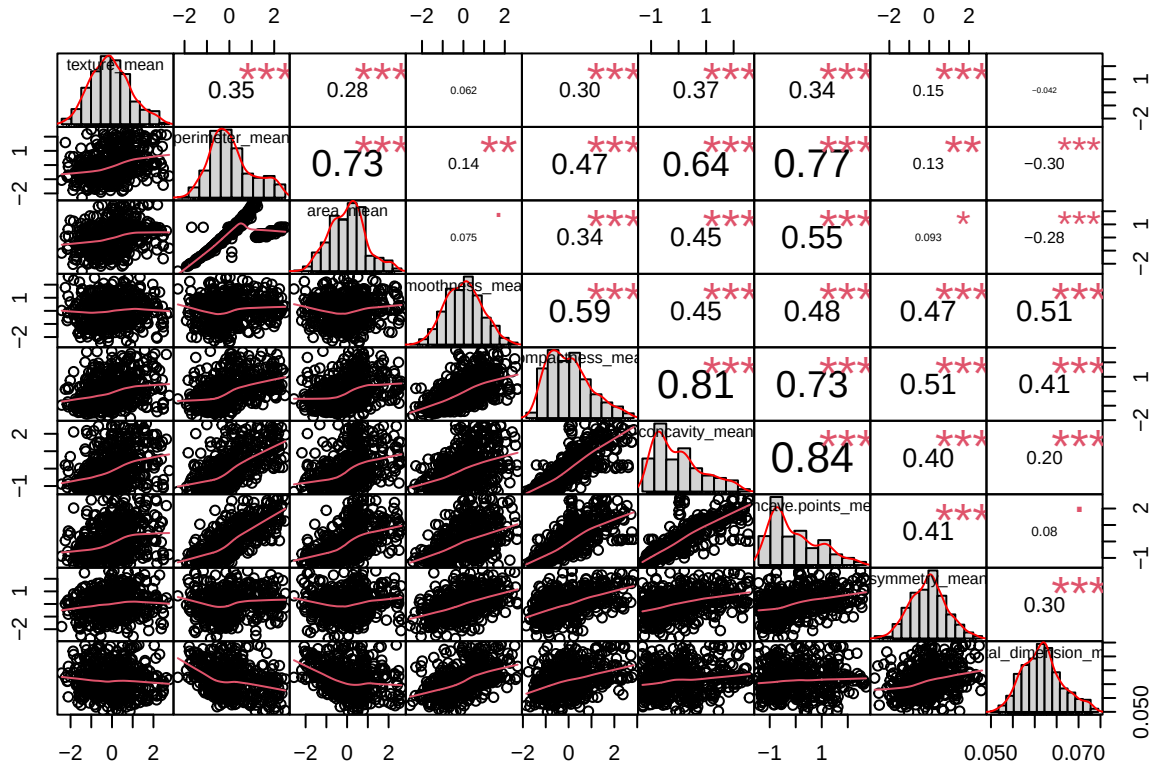
```
## [1] 1
```

```
  sd(cancer$texture_mean)  #observacion de la desviciancion
```

```
## [1] 1
```

Paso 5: Correlación entre las variables **(+25 puntos)** En esta sección encontraremos las varaibles que estan asociadas entre si.

```
  # correlacion de los datos
  Correl <- cancer
  Correl <- Correl %>%
    dplyr::select(-diagnosis)

chart.Correlation(cancer[,c(3:11)],histogram=TRUE, col="grey10", pch=1, main="Cancer Mean")
```




```
# Nos aseguramos que nuestro resultados sean repetibles
set.seed(5464)
names(cancer)
```

```
##  [1] "diagnosis"               "radius_mean"
##  [3] "texture_mean"            "perimeter_mean"
##  [5] "area_mean"               "smoothness_mean"
##  [7] "compactness_mean"        "concavity_mean"
##  [9] "concave.points_mean"     "symmetry_mean"
## [11] "fractal_dimension_mean"  "radius_se"
## [13] "texture_se"              "perimeter_se"
## [15] "area_se"                 "smoothness_se"
## [17] "compactness_se"          "concavity_se"
## [19] "concave.points_se"       "symmetry_se"
## [21] "fractal_dimension_se"    "radius_worst"
## [23] "texture_worst"           "perimeter_worst"
## [25] "area_worst"              "smoothness_worst"
## [27] "compactness_worst"       "concavity_worst"
## [29] "concave.points_worst"    "symmetry_worst"
## [31] "fractal_dimension_worst"
```

```
str(cancer)
```

```
## 'data.frame':    500 obs. of  31 variables:
##  $ diagnosis              : Factor w/ 2 levels "Benigno","Maligno": 2 2 2 2 2 2 2 2 2 2 ...
##  $ radius_mean            : num  0.0774 0.3816 0.2519 -0.8875 0.3391 ...
##  $ texture_mean           : num  -2.251 -0.293 0.63 0.399 -1.202 ...
##  $ perimeter_mean         : num  1.587 2.084 1.941 -0.638 2.192 ...
##  $ area_mean              : num  0.0784 0.5127 0.5127 -1.0059 0.5127 ...
##  $ smoothness_mean        : num  1.784 -0.842 1.098 0.039 0.372 ...
##  $ compactness_mean       : num  0.14 -0.458 1.457 0.14 0.818 ...
##  $ concavity_mean         : num  0.2059 0.1475 2.0096 0.0164 2.0197 ...
##  $ concave.points_mean    : num  0.0499 0.7825 2.6035 1.8875 1.8591 ...
##  $ symmetry_mean          : num  0.0219 0.1229 1.2622 0.1303 0.1096 ...
##  $ fractal_dimension_mean : num  0.0625 0.0567 0.06 0.0625 0.0588 ...
##  $ radius_se              : num  0.612 1.674 0.225 1.291 0.225 ...
##  $ texture_se             : num  -0.555 -0.997 -0.861 0.092 -0.875 ...
##  $ perimeter_se           : num  0.59 1.14 2.47 1.2 0.17 ...
##  $ area_se                : num  0.794 3 0.794 -0.136 0.794 ...
##  $ smoothness_se          : num  0.0444 -0.6492 -0.1027 1.6461 0.0854 ...
##  $ compactness_se         : num  2.643 -0.806 1.782 0.406 0.3 ...
##  $ concavity_se           : num  1.923 -0.511 0.856 2.123 2.142 ...
##  $ concave.points_se      : num  1.17 0.599 2.261 1.819 1.86 ...
##  $ symmetry_se            : num  0.031 -1.056 0.948 0.527 -0.201 ...
##  $ fractal_dimension_se   : num  2.458 0.276 1.128 0.476 1.574 ...
##  $ radius_worst           : num  0.736 0.736 0.558 0.227 0.323 ...
##  $ texture_worst          : num  -1.4322 -0.3364 0.0457 0.2205 -1.5512 ...
##  $ perimeter_worst        : num  0.369 0.356 0.276 0.369 0.276 ...
##  $ area_worst             : num  0.628 0.628 0.628 -0.536 0.347 ...
##  $ smoothness_worst       : num  0.162 0.124 0.144 0.132 0.137 ...
##  $ compactness_worst      : num  0.111 -0.424 1.399 0.111 -0.283 ...
##  $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ concave.points_worst   : num  2.267 1.062 1.927 2.147 0.706 ...
##  $ symmetry_worst         : num  0.292 0.275 0.361 0.292 0.236 ...
##  $ fractal_dimension_worst: num  0.0514 0.742 0.6197 0.2395 -0.2978 ...
```

```r
# Medir la correlación entre dos variables
cor(cancer$concavity_mean, cancer$concave.points_mean, use = 'complete.obs')
```

```
## [1] 0.8448512
```

```r
# Teniendo en cuenta la prueba de significancia y el valor de confianza
cor.test(cancer$concavity_mean, cancer$concave.points_mean, use = 'complete.obs')
```

```
##
##  Pearson's product-moment correlation
##
## data:  cancer$concavity_mean and cancer$concave.points_mean
## t = 35.24, df = 498, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8177435 0.8682193
## sample estimates:
##       cor
## 0.8448512
```

```r
# Calculamos la matrix de correlación para todo el dataframe
correlationMatrix <- cor(Correl[,1:30])  # Remover el outcome

# Resumir la matrix de correlaciòn
print(correlationMatrix)
```

```
##                          radius_mean texture_mean perimeter_mean   area_mean
## radius_mean              1.000000000   0.26997023     0.69839065  0.93351033
## texture_mean             0.269970230   1.00000000     0.34516006  0.28162618
## perimeter_mean           0.698390653   0.34516006     1.00000000  0.72818496
## area_mean                0.933510330   0.28162618     0.72818496  1.00000000
## smoothness_mean          0.084714626   0.06154174     0.14109246  0.07493978
## compactness_mean         0.350355518   0.29702015     0.47009494  0.34378274
## concavity_mean           0.453822103   0.36789878     0.63856165  0.45265364
## concave.points_mean      0.533634447   0.33559873     0.76778854  0.55285936
## symmetry_mean            0.067867920   0.15241645     0.13477580  0.09273803
## fractal_dimension_mean  -0.256714417  -0.04241470    -0.29651936 -0.27716347
## radius_se                0.290774021   0.33825977     0.46463117  0.32966309
## texture_se              -0.121210901   0.38857671    -0.07134318 -0.11865648
## perimeter_se             0.288336569   0.34622906     0.48228161  0.32985202
## area_se                  0.498098392   0.38638116     0.69838672  0.53639848
## smoothness_se           -0.215858735   0.06412053    -0.18089536 -0.22540302
## compactness_se           0.180328477   0.24815673     0.27814220  0.17023327
## concavity_se             0.268876301   0.30763682     0.39626746  0.27756133
## concave.points_se        0.298316762   0.24915130     0.42888921  0.32051867
## symmetry_se             -0.223827708  -0.02124864    -0.17526276 -0.21721799
## fractal_dimension_se     0.008985427   0.16286831     0.08807278  0.01283128
## radius_worst             0.771403183   0.23613370     0.62333612  0.73896242
## texture_worst            0.270286949   0.87857485     0.32898264  0.27816146
## perimeter_worst          0.825023984   0.25827961     0.66398592  0.81200109
## area_worst               0.827326574   0.28873509     0.70908186  0.84572467
## smoothness_worst         0.109079343   0.15037659     0.16084612  0.10218166
```

```
## compactness_worst        0.388526127   0.28915460     0.43551143  0.37490093
## concavity_worst          0.443529944   0.36273534     0.54547536  0.44750683
## concave.points_worst     0.535546905   0.36251685     0.68122414  0.56123560
## symmetry_worst           0.132911721   0.12285900     0.17999833  0.15757644
## fractal_dimension_worst  0.051353189   0.11450396     0.07994444  0.04440078
##                         smoothness_mean compactness_mean concavity_mean
## radius_mean                  0.08471463       0.35035552     0.45382210
## texture_mean                 0.06154174       0.29702015     0.36789878
## perimeter_mean               0.14109246       0.47009494     0.63856165
## area_mean                    0.07493978       0.34378274     0.45265364
## smoothness_mean              1.00000000       0.59046651     0.44818965
## compactness_mean             0.59046651       1.00000000     0.80755875
## concavity_mean               0.44818965       0.80755875     1.00000000
## concave.points_mean          0.47866063       0.73318295     0.84485120
## symmetry_mean                0.47381369       0.50727246     0.39933382
## fractal_dimension_mean       0.51284062       0.40746914     0.19780609
## radius_se                    0.29522306       0.40128451     0.45018138
## texture_se                   0.10070526       0.05989609     0.08989815
## perimeter_se                 0.28287124       0.48270021     0.50281984
## area_se                      0.28284076       0.46979216     0.57519766
## smoothness_se                0.33686967       0.15912624     0.13385922
## compactness_se               0.32701686       0.65086359     0.57647438
## concavity_se                 0.30636748       0.63855609     0.68555870
## concave.points_se            0.36835529       0.59589588     0.63144223
## symmetry_se                  0.15658948       0.08956506     0.01337033
## fractal_dimension_se         0.42320388       0.53980885     0.44350038
## radius_worst                 0.11454094       0.36157357     0.42841871
## texture_worst                0.09453609       0.26587311     0.36154300
## perimeter_worst              0.13893160       0.41501288     0.49290135
## area_worst                   0.11134241       0.37263482     0.47689693
## smoothness_worst             0.74821814       0.49565516     0.45018966
## compactness_worst            0.42391542       0.79802889     0.69972906
## concavity_worst              0.39584394       0.76023958     0.80663036
## concave.points_worst         0.48323643       0.75685239     0.78504280
## symmetry_worst               0.33894064       0.36940025     0.30098820
## fractal_dimension_worst      0.41596682       0.54168513     0.43195864
##                         concave.points_mean symmetry_mean
## radius_mean                      0.53363445    0.06786792
## texture_mean                     0.33559873    0.15241645
## perimeter_mean                   0.76778854    0.13477580
## area_mean                        0.55285936    0.09273803
## smoothness_mean                  0.47866063    0.47381369
## compactness_mean                 0.73318295    0.50727246
## concavity_mean                   0.84485120    0.39933382
## concave.points_mean              1.00000000    0.40781977
## symmetry_mean                    0.40781977    1.00000000
## fractal_dimension_mean           0.08025944    0.30033912
## radius_se                        0.51935667    0.26442409
## texture_se                       0.03706126    0.13176232
## perimeter_se                     0.55951932    0.28062114
## area_se                          0.65948122    0.22906018
## smoothness_se                    0.07936715    0.19928209
## compactness_se                   0.49752166    0.33453642
## concavity_se                     0.59390519    0.31713738
```

```
## concave.points_se               0.63123733    0.28372499
## symmetry_se                    -0.02981417    0.29778857
## fractal_dimension_se            0.36196590    0.35411018
## radius_worst                    0.50076415    0.09182393
## texture_worst                   0.32098695    0.16535902
## perimeter_worst                 0.55811139    0.13085058
## area_worst                      0.56340563    0.11762313
## smoothness_worst                0.44186644    0.36192016
## compactness_worst               0.63710936    0.38937623
## concavity_worst                 0.72493567    0.37826714
## concave.points_worst            0.82860079    0.38262282
## symmetry_worst                  0.32188103    0.60932899
## fractal_dimension_worst         0.33197774    0.32740453
##                         fractal_dimension_mean   radius_se   texture_se
## radius_mean                      -0.256714417 0.290774021 -0.121210901
## texture_mean                     -0.042414696 0.338259767  0.388576707
## perimeter_mean                   -0.296519364 0.464631170 -0.071343176
## area_mean                        -0.277163469 0.329663091 -0.118656477
## smoothness_mean                   0.512840623 0.295223056  0.100705258
## compactness_mean                  0.407469144 0.401284513  0.059896086
## concavity_mean                    0.197806095 0.450181383  0.089898154
## concave.points_mean               0.080259440 0.519356674  0.037061259
## symmetry_mean                     0.300339117 0.264424087  0.131762323
## fractal_dimension_mean            1.000000000 0.002036295  0.122326259
## radius_se                         0.002036295 1.000000000  0.230066882
## texture_se                        0.122326259 0.230066882  1.000000000
## perimeter_se                      0.049947347 0.859578354  0.241408363
## area_se                          -0.102195920 0.853057610  0.135866547
## smoothness_se                     0.343686490 0.197774936  0.328457152
## compactness_se                    0.352387028 0.352493900  0.191549216
## concavity_se                      0.225949407 0.403874845  0.148492253
## concave.points_se                 0.168194723 0.481335055  0.193206659
## symmetry_se                       0.243402760 0.148349735  0.272404513
## fractal_dimension_se              0.570232581 0.303234677  0.250492149
## radius_worst                     -0.171696115 0.282118119 -0.162418761
## texture_worst                    -0.044081844 0.261845835  0.439653089
## perimeter_worst                  -0.183625853 0.330393345 -0.124324746
## area_worst                       -0.254287992 0.391722636 -0.135184771
## smoothness_worst                  0.428423645 0.224230098  0.006618475
## compactness_worst                 0.345877044 0.278311328 -0.044393269
## concavity_worst                   0.205942259 0.362044827 -0.035466496
## concave.points_worst              0.129114889 0.446847895 -0.047975717
## symmetry_worst                    0.190136234 0.126130709 -0.087946802
## fractal_dimension_worst           0.621998425 0.093707157 -0.010184088
##                         perimeter_se      area_se smoothness_se compactness_se
## radius_mean               0.28833657  0.498098392  -0.215858735      0.1803285
## texture_mean              0.34622906  0.386381158   0.064120528      0.2481567
## perimeter_mean            0.48228161  0.698386717  -0.180895356      0.2781422
## area_mean                 0.32985202  0.536398482  -0.225403019      0.1702333
## smoothness_mean           0.28287124  0.282840755   0.336869673      0.3270169
## compactness_mean          0.48270021  0.469792160   0.159126238      0.6508636
## concavity_mean            0.50281984  0.575197657   0.133859221      0.5764744
## concave.points_mean       0.55951932  0.659481218   0.079367155      0.4975217
## symmetry_mean             0.28062114  0.229060182   0.199282090      0.3345364
```

```
## fractal_dimension_mean    0.04994735 -0.102195920   0.343686490      0.3523870
## radius_se                 0.85957835  0.853057610   0.197774936      0.3524939
## texture_se                0.24140836  0.135866547   0.328457152      0.1915492
## perimeter_se              1.00000000  0.776224730   0.205360103      0.4196617
## area_se                   0.77622473  1.000000000   0.076792830      0.3166896
## smoothness_se             0.20536010  0.076792830   1.000000000      0.2298685
## compactness_se            0.41966171  0.316689645   0.229868467      1.0000000
## concavity_se              0.45118520  0.404979556   0.216521466      0.7663742
## concave.points_se         0.53570589  0.480266017   0.298725851      0.6251033
## symmetry_se               0.14234558  0.006162693   0.377940035      0.2423769
## fractal_dimension_se      0.36508652  0.249156084   0.369817260      0.6594873
## radius_worst              0.28613397  0.435110347  -0.210090552      0.1712490
## texture_worst             0.26483978  0.325583573  -0.011825078      0.1960955
## perimeter_worst           0.35852747  0.501687698  -0.206075356      0.2459764
## area_worst                0.39543362  0.570046199  -0.238475442      0.1672158
## smoothness_worst          0.21000272  0.246454380   0.359423651      0.2605362
## compactness_worst         0.38447543  0.365417895   0.039768105      0.6222341
## concavity_worst           0.42798218  0.472266609   0.016457040      0.5660613
## concave.points_worst      0.50603702  0.566515693   0.023023597      0.5058466
## symmetry_worst            0.17708383  0.168591183  -0.005059716      0.1968134
## fractal_dimension_worst   0.15731740  0.091265234   0.112769908      0.4550825
##                         concavity_se concave.points_se  symmetry_se
## radius_mean                0.2688763         0.2983168 -0.223827708
## texture_mean               0.3076368         0.2491513 -0.021248639
## perimeter_mean             0.3962675         0.4288892 -0.175262755
## area_mean                  0.2775613         0.3205187 -0.217217986
## smoothness_mean            0.3063675         0.3683553  0.156589481
## compactness_mean           0.6385561         0.5958959  0.089565065
## concavity_mean             0.6855587         0.6314422  0.013370326
## concave.points_mean        0.5939052         0.6312373 -0.029814170
## symmetry_mean              0.3171374         0.2837250  0.297788570
## fractal_dimension_mean     0.2259494         0.1681947  0.243402760
## radius_se                  0.4038748         0.4813351  0.148349735
## texture_se                 0.1484923         0.1932067  0.272404513
## perimeter_se               0.4511852         0.5357059  0.142345584
## area_se                    0.4049796         0.4802660  0.006162693
## smoothness_se              0.2165215         0.2987259  0.377940035
## compactness_se             0.7663742         0.6251033  0.242376920
## concavity_se               1.0000000         0.7073164  0.167282362
## concave.points_se          0.7073164         1.0000000  0.181136145
## symmetry_se                0.1672824         0.1811361  1.000000000
## fractal_dimension_se       0.5664793         0.4931310  0.332221400
## radius_worst               0.2482183         0.2457089 -0.246305745
## texture_worst              0.2664067         0.1672616 -0.126958013
## perimeter_worst            0.3329414         0.3330301 -0.244963561
## area_worst                 0.2817886         0.2993301 -0.247001098
## smoothness_worst           0.2961792         0.2837144 -0.014184766
## compactness_worst          0.5950384         0.4885794 -0.040773075
## concavity_worst            0.6847629         0.5361382 -0.077788913
## concave.points_worst       0.5922330         0.6261868 -0.112280083
## symmetry_worst             0.2145734         0.1136625  0.234720495
## fractal_dimension_worst    0.3950003         0.2577184  0.049226872
##                         fractal_dimension_se radius_worst texture_worst
## radius_mean                      0.008985427   0.77140318    0.27028695
```

```
## texture_mean                    0.162868310   0.23613370    0.87857485
## perimeter_mean                  0.088072784   0.62333612    0.32898264
## area_mean                       0.012831284   0.73896242    0.27816146
## smoothness_mean                 0.423203879   0.11454094    0.09453609
## compactness_mean                0.539808849   0.36157357    0.26587311
## concavity_mean                  0.443500383   0.42841871    0.36154300
## concave.points_mean             0.361965903   0.50076415    0.32098695
## symmetry_mean                   0.354110181   0.09182393    0.16535902
## fractal_dimension_mean          0.570232581  -0.17169612   -0.04408184
## radius_se                       0.303234677   0.28211812    0.26184584
## texture_se                      0.250492149  -0.16241876    0.43965309
## perimeter_se                    0.365086521   0.28613397    0.26483978
## area_se                         0.249156084   0.43511035    0.32558357
## smoothness_se                   0.369817260  -0.21009055   -0.01182508
## compactness_se                  0.659487278   0.17124903    0.19609549
## concavity_se                    0.566479285   0.24821826    0.26640669
## concave.points_se               0.493130957   0.24570891    0.16726163
## symmetry_se                     0.332221400  -0.24630574   -0.12695801
## fractal_dimension_se            1.000000000   0.02473198    0.09661734
## radius_worst                    0.024731979   1.00000000    0.26339082
## texture_worst                   0.096617343   0.26339082    1.00000000
## perimeter_worst                 0.084776486   0.80590372    0.29023429
## area_worst                      0.008978150   0.77535244    0.31575404
## smoothness_worst                0.309043463   0.18461469    0.26180285
## compactness_worst               0.454996059   0.41483532    0.33093797
## concavity_worst                 0.414846008   0.46800754    0.39736140
## concave.points_worst            0.353199757   0.54444431    0.39528552
## symmetry_worst                  0.142893278   0.22125933    0.19950695
## fractal_dimension_worst         0.537363369   0.13471788    0.18154027
##                         perimeter_worst  area_worst smoothness_worst
## radius_mean                  0.82502398  0.82732657      0.109079343
## texture_mean                 0.25827961  0.28873509      0.150376592
## perimeter_mean               0.66398592  0.70908186      0.160846121
## area_mean                    0.81200109  0.84572467      0.102181657
## smoothness_mean              0.13893160  0.11134241      0.748218141
## compactness_mean             0.41501288  0.37263482      0.495655164
## concavity_mean               0.49290135  0.47689693      0.450189661
## concave.points_mean          0.55811139  0.56340563      0.441866439
## symmetry_mean                0.13085058  0.11762313      0.361920161
## fractal_dimension_mean      -0.18362585 -0.25428799      0.428423645
## radius_se                    0.33039335  0.39172264      0.224230098
## texture_se                  -0.12432475 -0.13518477      0.006618475
## perimeter_se                 0.35852747  0.39543362      0.210002723
## area_se                      0.50168770  0.57004620      0.246454380
## smoothness_se               -0.20607536 -0.23847544      0.359423651
## compactness_se               0.24597638  0.16721584      0.260536153
## concavity_se                 0.33294136  0.28178864      0.296179173
## concave.points_se            0.33303011  0.29933008      0.283714397
## symmetry_se                 -0.24496356 -0.24700110     -0.014184766
## fractal_dimension_se         0.08477649  0.00897815      0.309043463
## radius_worst                 0.80590372  0.77535244      0.184614687
## texture_worst                0.29023429  0.31575404      0.261802848
## perimeter_worst              1.00000000  0.89643339      0.181799929
## area_worst                   0.89643339  1.00000000      0.191321806
```

```
## smoothness_worst             0.18179993  0.19132181      1.000000000
## compactness_worst            0.48235638  0.44168998      0.511660448
## concavity_worst              0.52944147  0.50554328      0.505106469
## concave.points_worst         0.59058824  0.60415112      0.552000896
## symmetry_worst               0.19628936  0.19700759      0.438791952
## fractal_dimension_worst      0.13817635  0.09304957      0.486359237
##                         compactness_worst concavity_worst concave.points_worst
## radius_mean                    0.38852613      0.44352994           0.53554691
## texture_mean                   0.28915460      0.36273534           0.36251685
## perimeter_mean                 0.43551143      0.54547536           0.68122414
## area_mean                      0.37490093      0.44750683           0.56123560
## smoothness_mean                0.42391542      0.39584394           0.48323643
## compactness_mean               0.79802889      0.76023958           0.75685239
## concavity_mean                 0.69972906      0.80663036           0.78504280
## concave.points_mean            0.63710936      0.72493567           0.82860079
## symmetry_mean                  0.38937623      0.37826714           0.38262282
## fractal_dimension_mean         0.34587704      0.20594226           0.12911489
## radius_se                      0.27831133      0.36204483           0.44684790
## texture_se                    -0.04439327     -0.03546650          -0.04797572
## perimeter_se                   0.38447543      0.42798218           0.50603702
## area_se                        0.36541790      0.47226661           0.56651569
## smoothness_se                  0.03976811      0.01645704           0.02302360
## compactness_se                 0.62223406      0.56606135           0.50584664
## concavity_se                   0.59503842      0.68476292           0.59223303
## concave.points_se              0.48857937      0.53613823           0.62618677
## symmetry_se                   -0.04077307     -0.07778891          -0.11228008
## fractal_dimension_se           0.45499606      0.41484601           0.35319976
## radius_worst                   0.41483532      0.46800754           0.54444431
## texture_worst                  0.33093797      0.39736140           0.39528552
## perimeter_worst                0.48235638      0.52944147           0.59058824
## area_worst                     0.44168998      0.50554328           0.60415112
## smoothness_worst               0.51166045      0.50510647           0.55200090
## compactness_worst              1.00000000      0.84745897           0.77435727
## concavity_worst                0.84745897      1.00000000           0.84991962
## concave.points_worst           0.77435727      0.84991962           1.00000000
## symmetry_worst                 0.43985013      0.40471844           0.39374606
## fractal_dimension_worst        0.65177071      0.52743516           0.45338382
##                         symmetry_worst fractal_dimension_worst
## radius_mean                 0.132911721              0.05135319
## texture_mean                0.122858997              0.11450396
## perimeter_mean              0.179998325              0.07994444
## area_mean                   0.157576439              0.04440078
## smoothness_mean             0.338940641              0.41596682
## compactness_mean            0.369400251              0.54168513
## concavity_mean              0.300988198              0.43195864
## concave.points_mean         0.321881028              0.33197774
## symmetry_mean               0.609328986              0.32740453
## fractal_dimension_mean      0.190136234              0.62199843
## radius_se                   0.126130709              0.09370716
## texture_se                 -0.087946802             -0.01018409
## perimeter_se                0.177083834              0.15731740
## area_se                     0.168591183              0.09126523
## smoothness_se              -0.005059716              0.11276991
## compactness_se              0.196813412              0.45508254
```

```
## concavity_se               0.214573381             0.39500029
## concave.points_se          0.113662490             0.25771836
## symmetry_se                0.234720495             0.04922687
## fractal_dimension_se       0.142893278             0.53736337
## radius_worst               0.221259330             0.13471788
## texture_worst              0.199506952             0.18154027
## perimeter_worst            0.196289360             0.13817635
## area_worst                 0.197007590             0.09304957
## smoothness_worst           0.438791952             0.48635924
## compactness_worst          0.439850134             0.65177071
## concavity_worst            0.404718445             0.52743516
## concave.points_worst       0.393746059             0.45338382
## symmetry_worst             1.000000000             0.37607627
## fractal_dimension_worst    0.376076270             1.00000000
```

```
# encontrar variables que están altamente correlacionas (idealmente> 0,75)
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.75)

# Nombre de la variable(s) correlacionadas
cat("Variable(s) correlacionada(s):")
```

```
## Variable(s) correlacionada(s):
```

```
colnames(Correl[highlyCorrelated])
```

```
##  [1] "concave.points_worst" "concavity_mean"       "concave.points_mean"
##  [4] "compactness_mean"     "concavity_worst"      "concavity_se"
##  [7] "area_se"              "perimeter_worst"      "area_worst"
## [10] "perimeter_se"         "area_mean"            "radius_worst"
## [13] "texture_mean"
```

```
# Eliminar las variables que no tienen fuerte asociacion con el outcome ............
# Correlation Matrix
correlationMatrix <- cor(cancer[,3:31])            # Select only numerical Variables
hcorrelated       <- findCorrelation(correlationMatrix, cutoff=0.6)# Threshold >0.6, Find Features that
print(hcorrelated)                                    # print indexes of highly correlated attribute
```

```
##  [1] 27  6  7  5 26 25 16  2 17 13 15 22 23 12 20  4  8 29 21
```

```
highly_cor_var    <- colnames(cancer[hcorrelated])        # displaying highly correlated variables
data.frame(highly_cor_var)
```

```
##          highly_cor_var
## 1     compactness_worst
## 2       smoothness_mean
## 3      compactness_mean
## 4             area_mean
## 5      smoothness_worst
## 6           area_worst
## 7         smoothness_se
## 8           radius_mean
```

```
## 9        compactness_se
## 10           texture_se
## 11              area_se
## 12         radius_worst
## 13        texture_worst
## 14            radius_se
## 15          symmetry_se
## 16       perimeter_mean
## 17       concavity_mean
## 18 concave.points_worst
## 19 fractal_dimension_se
```

Paso 6: Implementación de los modelos **(+25 puntos)**

Para esto puede usar el paquete Caret:

13. Con el data set completo y limpio, realice la division del dataset de la siguiente forma:
    i) 80% de los datos para el entrenamiento del modelo (seleccionando las primeras 400 observaciones)
    ii) 20% de los datos para la evaluación del modelo (seleccionando las ultimas 200 observaciones)

```
#Data det con los datos filtrado
cancer1 <- cancer %>%
  dplyr::select(diagnosis, concave.points_worst, concavity_mean, concave.points_mean,
                compactness_mean, concavity_worst, concavity_se,
                area_se, perimeter_worst, area_worst, perimeter_se,
                area_mean, radius_worst, texture_mean)

 # Split 80% y 20% datos completos

set.seed(899)   # Valores aleatorios
inTrain <- createDataPartition(y=cancer$diagnosis, p = 0.80, list =FALSE)  #
train   <- cancer[inTrain,]
test    <- cancer[-inTrain,]

dim(train)
```

```
## [1] 400  31
```

```
dim(test)
```

```
## [1] 100  31
```

```
# Split 80% y 20% datos filtrado

set.seed(899)   # Valores aleatorios
inTrain2 <- createDataPartition(y=cancer1$diagnosis, p = 0.80, list =FALSE)
train2   <- cancer1[inTrain2,]
test2    <- cancer1[-inTrain2,]

dim(train2)
```

```
## [1] 400  14
```

```r
dim(test2)
```

```
## [1] 100  14
```

14. Implemente los modelos de regresión logistica del paquete caret.

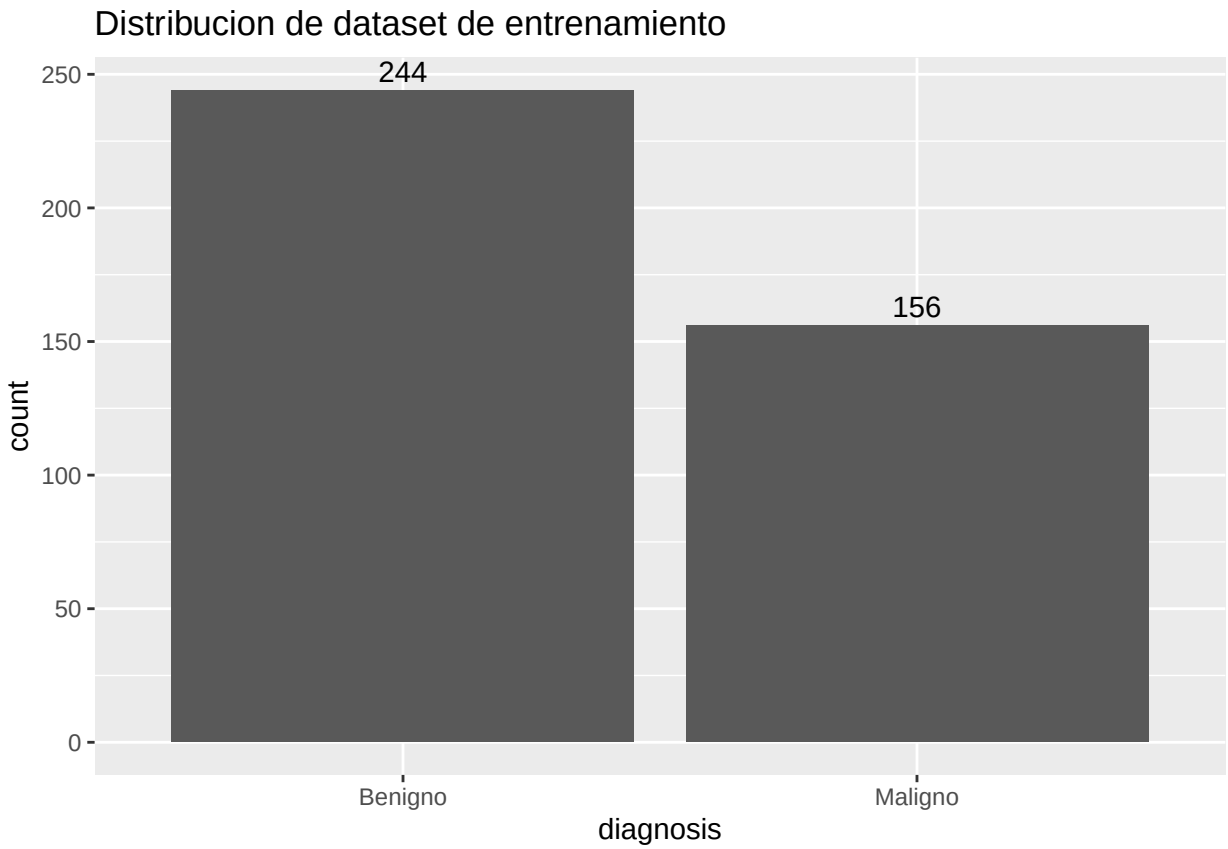
```r
#MODELOS LOGISTICA Y KNN

# ProporciÃ³n de outcome en entrenamiento
prop.table(table(train$diagnosis))
```

```
##
## Benigno Maligno
##    0.61    0.39
```

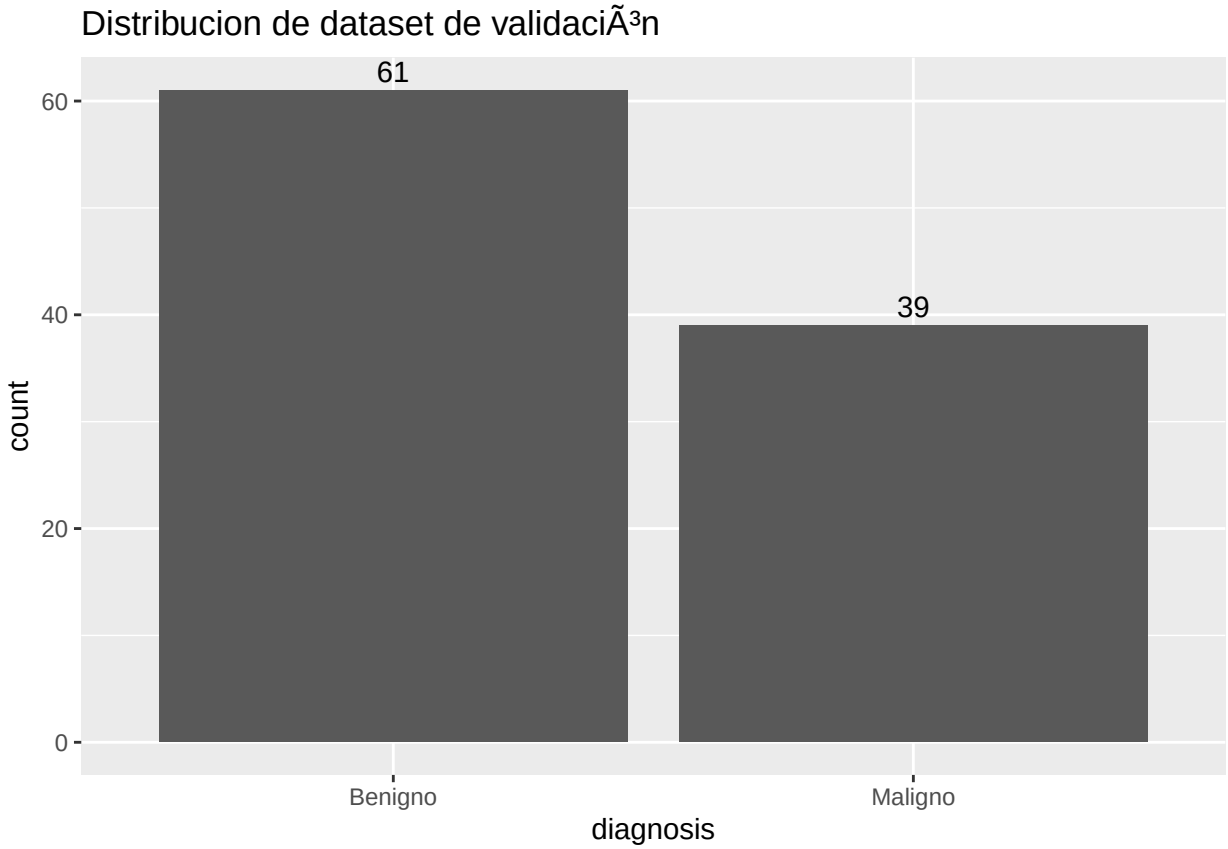
```r
# ProporciÃ³n de Outcome en test
prop.table(table(test$diagnosis))
```

```
##
## Benigno Maligno
##    0.61    0.39
```

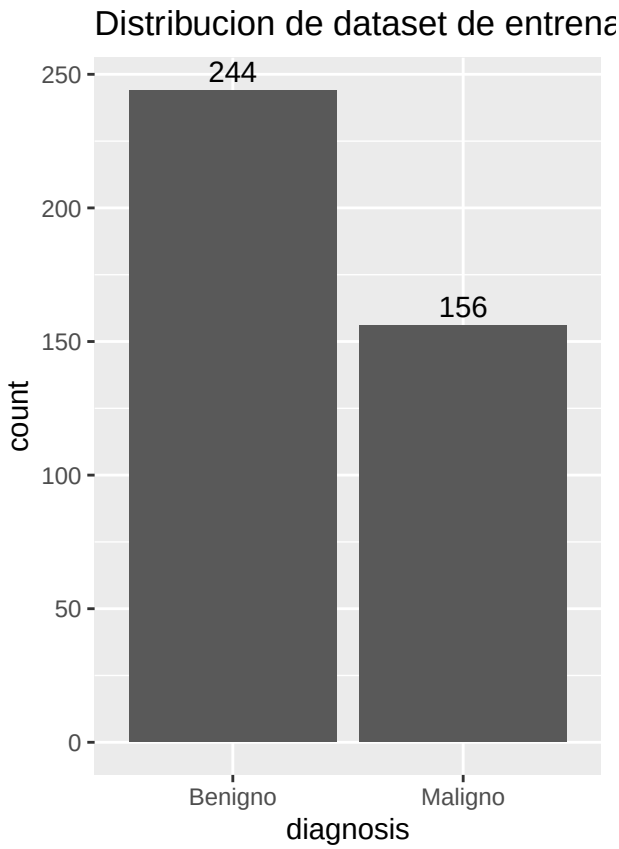
```r
# Visualizar la distribucion de training y test
a1 <-ggplot(data=train,aes(x=diagnosis))+
  geom_bar() +
  ggtitle('Distribucion de dataset de entrenamiento')+
  geom_text(stat='Count',aes(label=..count..),vjust=-0.4)
a1
```

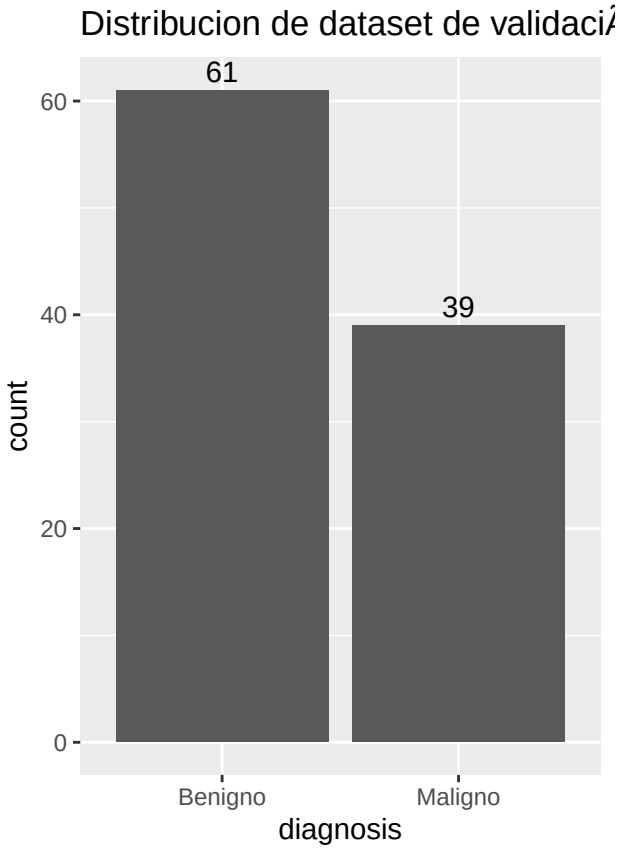



```
a2 <- ggplot(data=test,aes(x=diagnosis))+
  geom_bar() +
  ggtitle('Distribucion de dataset de validaciÃ³n')+
  geom_text(stat='Count',aes(label=..count..),vjust=-0.4)
a2
```




```
grid.arrange(a1, a2, nrow=1)
```






```
#---------- Datos Completos ----------------
#Para entrenar los modelos usamos el mÃ©todo Cross validation PUNTO15----
control <- trainControl(method = 'repeatedcv',number = 10,repeats = 3)

# Logistic Model ----------------------------------
set.seed(1156)
logFit <- train(diagnosis ~.,
                data = train,
                method = 'glm',
                preProc = c("center", "scale"),
                trControl = control)
logFit$results
```

```
##   parameter  Accuracy     Kappa AccuracySD KappaSD
## 1      none 0.9276094 0.8485248 0.04073722 0.08437
```

```
#MODELO KNN-----
control1 <- trainControl(method="repeatedcv", number=10, repeats=3)

set.seed(445)
knnFit <- train(diagnosis ~.,
                data = train,
                method = "knn",
                preProc = c("center", "scale"),
                tuneGrid = data.frame(.k = 1:10),
```

```
                trControl = control1)
knnFit$results
```

```
##     k  Accuracy     Kappa AccuracySD    KappaSD
## 1   1 0.9592203 0.9140142 0.03033691 0.06419236
## 2   2 0.9509876 0.8958855 0.03213494 0.06828003
## 3   3 0.9674927 0.9310533 0.02556518 0.05438490
## 4   4 0.9616359 0.9184762 0.03209744 0.06898872
## 5   5 0.9666166 0.9288355 0.02596205 0.05636991
## 6   6 0.9640953 0.9232952 0.02949164 0.06464954
## 7   7 0.9657630 0.9269614 0.02701230 0.05878064
## 8   8 0.9607823 0.9161977 0.02795696 0.06070918
## 9   9 0.9549479 0.9034335 0.03272299 0.07150108
## 10 10 0.9516338 0.8959628 0.03442734 0.07627937
```

```
#--------------  Datos filtrado----------------------
#Para entrenar los modelos usamos el mÃ©todo Cross validation PUNTO15----
control2 <- trainControl(method = 'repeatedcv',number = 10,repeats = 3)

# Logistic Model ----------------------------------
set.seed(1156)
logFit1 <- train(diagnosis ~.,
                 data = train2,
                 method = 'glm',
                 preProc = c("center", "scale"),
                 trControl = control2)
logFit1$results
```

```
##   parameter  Accuracy     Kappa AccuracySD    KappaSD
## 1      none 0.9492183 0.8929931 0.03742734 0.07897521
```

```
#MODELO KNN-----
control3 <- trainControl(method="repeatedcv", number=10, repeats=3)

set.seed(445)
knnFit1 <- train(diagnosis ~.,
                 data = train2,
                 method = "knn",
                 preProc = c("center", "scale"),
                 tuneGrid = data.frame(.k = 1:10),
                 trControl = control3)
knnFit1$results
```

```
##     k  Accuracy     Kappa AccuracySD    KappaSD
## 1   1 0.9424833 0.8787168 0.03582112 0.07726792
## 2   2 0.9284183 0.8498226 0.04003817 0.08365529
## 3   3 0.9434438 0.8817317 0.04125789 0.08583460
## 4   4 0.9367751 0.8678770 0.03910017 0.08149077
## 5   5 0.9417558 0.8784958 0.04268315 0.08913413
## 6   6 0.9409021 0.8758427 0.03750166 0.07923730
## 7   7 0.9391948 0.8721057 0.03601979 0.07608542
## 8   8 0.9442568 0.8829341 0.03634149 0.07595006
```

```
## 9   9 0.9466959 0.8875010 0.03140477 0.06673250
## 10 10 0.9458828 0.8858776 0.03597184 0.07619943
```

Paso 6: Evaluación del performance del modelo **(+20 puntos)**

15. Una vez apliquen las métricas para medir el performance (desempeño), esto es la matriz de confusión sobre los datos de evaluación (test).

```
## Validacion de los datos

## validacion datos completos

# Modelo logico

predictionglm <- predict(logFit, newdata = test)  # Probabilidad de que sea 1(no cumple)
confusionMatrix(predictionglm,test$diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Benigno Maligno
##    Benigno      58       2
##    Maligno       3      37
##
##                Accuracy : 0.95
##                  95% CI : (0.8872, 0.9836)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : 2.983e-15
##
##                   Kappa : 0.8954
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9508
##             Specificity : 0.9487
##          Pos Pred Value : 0.9667
##          Neg Pred Value : 0.9250
##              Prevalence : 0.6100
##          Detection Rate : 0.5800
##    Detection Prevalence : 0.6000
##       Balanced Accuracy : 0.9498
##
##        'Positive' Class : Benigno
##
```

```
#Modelo KNN
predictionglm1 <- predict(knnFit, newdata = test)  # Probabilidad de que sea 1(no cumple)
confusionMatrix(predictionglm1,test$diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction Benigno Maligno
##    Benigno      61       2
##    Maligno       0      37
##
##                Accuracy : 0.98
##                  95% CI : (0.9296, 0.9976)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9576
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 1.0000
##             Specificity : 0.9487
##          Pos Pred Value : 0.9683
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6100
##          Detection Rate : 0.6100
##    Detection Prevalence : 0.6300
##       Balanced Accuracy : 0.9744
##
##        'Positive' Class : Benigno
##
```

```
## validacion datos completos  filtrado

# Modelo logico

predictionglm2 <- predict(logFit1, newdata = test2)  # Probabilidad de que sea 1(no cumple)
confusionMatrix(predictionglm2,test2$diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Benigno Maligno
##    Benigno      60       3
##    Maligno       1      36
##
##                Accuracy : 0.96
##                  95% CI : (0.9007, 0.989)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : 2.387e-16
##
##                   Kappa : 0.9151
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9836
##             Specificity : 0.9231
##          Pos Pred Value : 0.9524
##          Neg Pred Value : 0.9730
##              Prevalence : 0.6100
##          Detection Rate : 0.6000
```

```
##    Detection Prevalence : 0.6300
##       Balanced Accuracy : 0.9533
##
##        'Positive' Class : Benigno
##
```

```
#Modelo KNN
predictionglm3 <- predict(knnFit1, newdata = test2)  # Probabilidad de que sea 1(no cumple)
confusionMatrix(predictionglm3,test2$diagnosis)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Benigno Maligno
##    Benigno      59       4
##    Maligno       2      35
##
##                Accuracy : 0.94
##                  95% CI : (0.874, 0.9777)
##     No Information Rate : 0.61
##     P-Value [Acc > NIR] : 3.076e-14
##
##                   Kappa : 0.8727
##
##  Mcnemar's Test P-Value : 0.6831
##
##             Sensitivity : 0.9672
##             Specificity : 0.8974
##          Pos Pred Value : 0.9365
##          Neg Pred Value : 0.9459
##              Prevalence : 0.6100
##          Detection Rate : 0.5900
##    Detection Prevalence : 0.6300
##       Balanced Accuracy : 0.9323
##
##        'Positive' Class : Benigno
##
```

```
+ Indique cuál de los dos modelos es más exacto en la detección de tipos de tumores? El modelo donde se
R//
Según los datos obtenidos el modelo KNN es mas exacto debido a que la precisión en ambos casos es mayor
Tambien el modelo logico es mejor en las variables filtrada del data set y el KNN es mejor en los datos
+ Cúal modelo tiene menos errores en cuanto la deteccion de tumores benignos o malinos.
R// El modelo KNN
+ Qué indica la sensibilidad y especificidad en la evaluación de desempeño de los modelos de machine lea
R//
La sensibilidad en machine learning nos indica el número de elementos identificados correctamente como p
+ Que los valores predictivos positivos y negativos en la metricas de evaluación?
+ En un gráfico (barplot) muestre el porcentaje de pacientes detectados por el modelo de regresion  Vs l
  pacientes detectados por el KNN.
```

16. ¿Qué conclusiones puede sacar sobre el analisis de estos datos? ¿Cómo responde a la pregunta de investigación?

Las variables mas relevantes son la que estan mas relacionadas

Variables correlacionada

compactness_worst - smoothness_mean - compactness_mean - area_mean - smoothness_worst - area_worst - smoothness_se - radius_mean - compactness_se - texture_se - area_se - adius_worst - texture_worst - radius_se - symmetry_se - perimeter_mean - concavity_mean - concave.points_worst- fractal_dimension_se-