Práctica 2: Limpieza y validación de los datos

Irati Boda Ezeiza - iboda001

11 de junio de 2018

Detalles de la actividad

Presentación

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

Competencias

En esta práctica se desarrollan las siguientes competencias del *Máster de Data Science*:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar

RESOLUCIÓN

El dataset, el código y el informe generado están disponibles este enlace de github.

Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

1. Descripción del dataset

El dataset escogido ha sido "Breast Cancer Wisconsin (Diagnostic) Data Set" disponible en kaggle en este enlace. Dicho dataset es un conjunto de datos que incluye características de masas mamarias malignas como benignas obtenidos mediante imágenes digitalizadas. Está compuesto por un total de 569 muestras y sus respectivas 32 características (columnas) descritas a continuación:

- **id**: Número identificativo. *Tipo de dato: Numérico.*
- **diagnosis**: Diagnóstico (M = maligno, B = benigno). *Tipo de dato:Booleano*

Las siguientes características describen las características de los núcleos celulares presentes en la imagen:

- **radius_mean**: Media de las distancias entre en centro y el perímetro. *Tipo de dato: Numérico.*
- **radius_se**: Error estándar de las distancias entre en centro y el perímetro. *Tipo de dato: Numérico.*
- **radius_worst**: "Peor" o mayor valor medio de las distancias entre en centro y el perímetro. *Tipo de dato: Numérico.*
- **texture_mean**: Media de valores de la escala de grises. *Tipo de dato: Numérico.*
- **texture_se**: Error estándar de valores de la escala de grises. *Tipo de dato: Numérico.*
- **texture_worst**: "Peor" o mayor valor medio de valores de la escala de grises. *Tipo de dato: Numérico.*
- **perimeter_mean**: Media del perímetro. *Tipo de dato: Numérico.*
- **perimeter_se**: Error estándar del perímetro. *Tipo de dato: Numérico.*
- **perimeter_worst**: "Peor" o mayor valor medio del perímetro. *Tipo de dato: Numérico.*
- **area_mean**: Media del área *Tipo de dato: Numérico.*
- **area_se**: Error estándar del área *Tipo de dato: Numérico.*
- **area_worst**: "Peor" o mayor valor medio del área *Tipo de dato: Numérico.*
- **smoothness_mean**: Media de la variación local de longitudes de radio. *Tipo de dato: Numérico.*
- **smoothness_se**: Error estándar de la variación local de longitudes de radio. *Tipo de dato: Numérico.*
- **smoothness_worst**: "Peor" o mayor valor medio de la variación local de longitudes de radio. *Tipo de dato: Numérico.*
- **compactness_mean**: Media del perímetro al cuadrado partido entre el área - 1. *Tipo de dato: Numérico.*
- **compactness_se**: Error estándar del perímetro al cuadrado partido entre el área - 1. *Tipo de dato: Numérico.*
- **compactness_worst**: "Peor" o mayor valor medio del perímetro al cuadrado partido entre el área - 1. *Tipo de dato: Numérico.*
- **concavity_mean**: Media de severidad de porciones cóncavas del contorno *Tipo de dato: Numérico.*
- **concavity_se**: Error estándar de severidad de porciones cóncavas del contorno *Tipo de dato: Numérico.*

- **concavity_worst**: "Peor" o mayor valor medio de severidad de porciones cóncavas del contorno *Tipo de dato: Numérico.*
- **concave points_mean**: Media del número de proporciones cóncavas del contorno. *Tipo de dato: Numérico.*
- **concave points_se**: Error estándar del número de proporciones cóncavas del contorno. *Tipo de dato: Numérico.*
- **concave points_worst**: "Peor" o mayor valor medio del número de proporciones cóncavas del contorno. *Tipo de dato: Numérico.*
- **symmetry_mean**: Media de simetría. *Tipo de dato: Numérico.*
- **symmetry_se**: Error estándar de simetría. *Tipo de dato: Numérico.*
- **symmetry_worst**: "Peor" o mayor valor medio de simetría. *Tipo de dato: Numérico.*
- **fractal_dimension_mean**: Media de la "aproximación costera" - 1. *Tipo de dato: Numérico*
- **fractal_dimension_se**: Error estándar de la "aproximación costera" - 1. *Tipo de dato: Numérico*
- **fractal_dimension_worst**: "Peor" o mayor valor medio de la "aproximación costera" - 1. *Tipo de dato: Numérico*

Este dataset es de gran interés tanto para comunidad científica como para los pacientes y futuros pacientes. Para los primeros, hablamos de una base de datos real y de alta, de la que además se disponen de otras dos bases de datos complementarias: Breast Cancer Wisconsin (Origiginal) y Breast Cancer Wisconsin (Prognostic). Son muchos los trabajos científicos realizados con este dataset, tanto de machine learning (árboles de decisión, redes neuronales) como de imaging (visualización y localización de la evolución del tumor).

En el caso que nos ocupa, el **objetivo** es construir un clasificador de que sea capaz de reconocer si la masa mamaria es maligna o benigna basándose en las características.

2. Integración y selección de los datos de interés a analizar.

El dataset está compuesto por un solo archivo delimitado por comas (csv). Por lo tanto, comenzaremos con la lectura de dicho fichero mediante la función *read.csv*() y comprobaremos que la carga se ha realizado correctamente imprimiendo las primeras filas y columnas.

```
#Carga de datos
setwd(path)
data=read.csv(file="data.csv")
old.data<-data[,-ncol(data)]
head(data[,1:6])
```

```
##         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
```

Además, es de nuestro interés analizar si los tipos de datos asignados automáticamente coinciden con los descritos en el apartado anterior:

```
#Tipo de dato
tipo.variable <- sapply(data,class)
knitr::kable(data.frame(variables=names(tipo.variable),clase=as.vector(tipo.variable)))
```

variables	clase
id	integer
diagnosis	factor
radius_mean	numeric
texture_mean	numeric
perimeter_mean	numeric
area_mean	numeric
smoothness_mean	numeric
compactness_mean	numeric
concavity_mean	numeric
concave.points_mean	numeric
symmetry_mean	numeric
fractal_dimension_mean	numeric
radius_se	numeric
texture_se	numeric
perimeter_se	numeric
area_se	numeric
smoothness_se	numeric
compactness_se	numeric
concavity_se	numeric
concave.points_se	numeric
symmetry_se	numeric
fractal_dimension_se	numeric
radius_worst	numeric
texture_worst	numeric
perimeter_worst	numeric
area_worst	numeric
smoothness_worst	numeric
compactness_worst	numeric
concavity_worst	numeric
concave.points_worst	numeric
symmetry_worst	numeric
fractal_dimension_worst	numeric
X	logical

Observamos que los tipos de datos asignados automáticamente por R se corresponden con la realidad, aunque aparece una última variable X que deberá ser eliminada en el próximo apartado.

3. Limpieza de los datos.

3.1. Ceros y elementos vacíos

Los valores nulos o ceros, entendiendo estos últimos como indicador de ausencia de valores, pueden resultar ser un problema de gran importancia en las bases de datos. Las razones de su existencia suelen ser de distinta índole: errores manuales, errores técnicos, mediciones incorrectas o incluso de manipulación intencionada. Para garantizar la calidad de nuestro análisis, es imprescindible examinar la existencia de estos, y si la hubiera, tomar la decisión bien de eliminarlos o de imputarlos mediante alguno de los métodos recomendados. En cualesquiera de los casos citados, la manipulación de los datos originales deberá ser citado y justificado.

```
#Ceros y elementos vacíos
na.count <-sapply(data, function(x) sum((is.na(x))))
na.count
```

```
##                      id               diagnosis             radius_mean
##                       0                       0                       0
##            texture_mean          perimeter_mean               area_mean
##                       0                       0                       0
##         smoothness_mean        compactness_mean          concavity_mean
##                       0                       0                       0
##     concave.points_mean           symmetry_mean  fractal_dimension_mean
##                       0                       0                       0
##               radius_se              texture_se            perimeter_se
##                       0                       0                       0
##                 area_se           smoothness_se          compactness_se
##                       0                       0                       0
##            concavity_se       concave.points_se             symmetry_se
##                       0                       0                       0
##    fractal_dimension_se            radius_worst           texture_worst
##                       0                       0                       0
##         perimeter_worst              area_worst        smoothness_worst
##                       0                       0                       0
##       compactness_worst         concavity_worst    concave.points_worst
##                       0                       0                       0
##          symmetry_worst fractal_dimension_worst                       X
##                       0                       0                     569
```

Observamos que ninguna de las variables contiene valores vacíos excepto la variable X que está totalmente vacía. Por lo tanto, procederemos a eliminar la característica X.

```
#Eliminamos la columna vacía "X"
data=data[,!colnames(data)=="X"]
attach(data)
```

3.2. Identificación y tratamiento de valores extremos.

Los valores extremos o *outliers*, son aquellas observaciones que son numéricamente distantes del resto de los datos. Por lo tanto, son sospechosos de no pertenecer al conjunto de datos de donde proceden. Identificaremos los valores extremos de nuestro conjunto de datos con la función $boxplot.stats(x)\$out$ de R, y las graficaremos mediante diagramas de cajas.

```
#Valores extremos (outliers)
numeric.data <- data[,c(-1,-2)]
```

```r
for(i in 1:length(numeric.data) ) {
  cat('Outliers de la variable "',names(numeric.data)[i],'" :\n')
  print(boxplot.stats(numeric.data[,i])$out)
}
```

```
## Outliers de la variable " radius_mean " :
##  [1] 25.22 22.27 24.25 23.27 27.22 23.29 28.11 23.21 23.51 25.73 22.01
## [12] 27.42 23.09 24.63
## Outliers de la variable " texture_mean " :
## [1] 32.47 33.81 39.28 33.56 31.12 30.72 30.62
## Outliers de la variable " perimeter_mean " :
##  [1] 171.5 152.8 166.2 152.1 182.1 158.9 188.5 153.5 155.1 174.2 186.9
## [12] 152.1 165.5
## Outliers de la variable " area_mean " :
##  [1] 1404 1878 1509 1761 1686 2250 1685 2499 1670 1364 1419 1491 1747 2010
## [15] 1546 1482 1386 1335 1407 1384 2501 1682 1841 1347 1479
## Outliers de la variable " smoothness_mean " :
## [1] 0.14250 0.13980 0.14470 0.16340 0.13710 0.05263
## Outliers de la variable " compactness_mean " :
##  [1] 0.2776 0.2839 0.2396 0.2458 0.2293 0.3454 0.2665 0.2768 0.2867 0.2832
## [11] 0.2413 0.3114 0.2364 0.2363 0.2576 0.2770
## Outliers de la variable " concavity_mean " :
##  [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 0.3174 0.3514
## Outliers de la variable " concave.points_mean " :
##  [1] 0.1604 0.1845 0.1823 0.2012 0.1878 0.1620 0.1595 0.1913 0.1562 0.1689
## Outliers de la variable " symmetry_mean " :
##  [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 0.2538 0.1060
## Outliers de la variable " fractal_dimension_mean " :
##  [1] 0.09744 0.08243 0.08046 0.08980 0.08142 0.08261 0.09296 0.08116
##  [9] 0.08104 0.08743 0.08450 0.07950 0.09502 0.09575 0.07976
## Outliers de la variable " radius_se " :
##  [1] 1.0950 0.9555 1.0460 0.8529 1.2140 0.9811 0.9806 0.9317 0.8973 1.2150
## [11] 1.5090 1.2960 1.0000 1.0880 0.8601 2.8730 0.9553 1.0580 1.0040 1.2920
## [21] 1.1720 1.1670 0.8811 1.1110 1.0720 1.0090 0.9948 0.9761 1.2070 1.0080
## [31] 1.3700 0.9291 2.5470 0.9289 1.2910 0.9915 0.9622 1.1760
## Outliers de la variable " texture_se " :
##  [1] 3.568 2.910 3.120 2.508 2.664 4.885 2.612 2.454 2.777 2.509 2.836
## [12] 2.878 2.542 2.643 3.647 2.635 2.927 2.904 3.896 2.463
## Outliers de la variable " perimeter_se " :
##  [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  6.051 18.650  9.635  7.050  8.758  7.673
## Outliers de la variable " area_se " :
##  [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  87.87 170.00
## [21]  90.47 233.00 101.90  93.91 119.30  97.07  97.85 122.30 128.70 111.70
## [31] 525.60 124.40 109.90 155.80 137.90  92.81 106.40 138.50  90.94 199.70
## [41] 156.80 133.00 130.80  87.17  88.25 164.10 153.10 103.60 224.10 130.20
## [51] 176.50 103.90 115.20 542.20 104.90  89.74  95.77 180.20 139.90 100.40
## [61]  87.78 118.80 158.70  99.04  86.22
## Outliers de la variable " smoothness_se " :
```

```
##  [1] 0.01721 0.01340 0.01385 0.01291 0.01835 0.02333 0.01496 0.01286
##  [9] 0.01439 0.01380 0.01345 0.03113 0.01604 0.01380 0.01418 0.01574
## [17] 0.02075 0.01289 0.01736 0.01582 0.01474 0.01307 0.01459 0.02177
## [25] 0.01262 0.01546 0.01288 0.01266 0.01547 0.01291
## Outliers de la variable " compactness_se " :
##  [1] 0.07458 0.07217 0.08297 0.10060 0.07056 0.08606 0.09368 0.06835
##  [9] 0.08668 0.07446 0.06760 0.09806 0.09586 0.08808 0.13540 0.08555
## [17] 0.08262 0.10640 0.06590 0.06559 0.07643 0.06669 0.06213 0.06657
## [25] 0.07025 0.07471 0.06457 0.06158
## Outliers de la variable " concavity_se " :
##  [1] 0.08890 0.09723 0.30380 0.10910 0.10400 0.14350 0.09263 0.12780
##  [9] 0.39600 0.11970 0.11660 0.08958 0.14380 0.08880 0.09518 0.09960
## [17] 0.10270 0.09953 0.15350 0.09472 0.11140 0.09252
## Outliers de la variable " concave.points_se " :
##  [1] 0.04090 0.02638 0.03322 0.02593 0.02801 0.05279 0.02794 0.02765
##  [9] 0.03927 0.03024 0.03487 0.02771 0.02536 0.02919 0.03441 0.02598
## [17] 0.02721 0.02853 0.02624
## Outliers de la variable " symmetry_se " :
##  [1] 0.05963 0.04484 0.03672 0.05333 0.04183 0.04192 0.04197 0.07895
##  [9] 0.05014 0.04547 0.05168 0.05628 0.03880 0.05113 0.03799 0.04783
## [17] 0.04499 0.04077 0.06146 0.04022 0.04243 0.03756 0.03675 0.05543
## [25] 0.03710 0.03997 0.03759
## Outliers de la variable " fractal_dimension_se " :
##  [1] 0.009208 0.010080 0.012840 0.008093 0.009559 0.021930 0.010390
##  [8] 0.012980 0.009875 0.009423 0.009368 0.011780 0.029840 0.017920
## [15] 0.011720 0.012560 0.008675 0.008660 0.022860 0.012200 0.012330
## [22] 0.008925 0.008133 0.011300 0.009627 0.010450 0.011480 0.008313
## Outliers de la variable " radius_worst " :
##  [1] 29.17 30.00 28.40 28.01 33.12 28.11 27.90 31.01 32.49 28.19 30.67
## [12] 33.13 30.75 27.66 36.04 30.79 29.92
## Outliers de la variable " texture_worst " :
## [1] 45.41 44.87 49.54 47.16 42.79
## Outliers de la variable " perimeter_worst " :
##  [1] 188.0 211.7 206.8 220.8 188.5 206.0 214.0 195.9 202.4 229.3 199.5
## [12] 195.0 251.2 211.5 205.7
## Outliers de la variable " area_worst " :
##  [1] 2019 1956 2398 2615 2215 2145 2562 2360 2073 2232 2403 3216 2089 1986
## [15] 2499 2009 2477 2944 2010 1972 3432 2384 2053 1938 2906 3234 3143 2227
## [29] 1946 2081 2022 4254 2782 2642 2027
## Outliers de la variable " smoothness_worst " :
## [1] 0.20980 0.19090 0.07117 0.22260 0.21840 0.19020 0.20060
## Outliers de la variable " compactness_worst " :
##  [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 0.7917 0.8681
## Outliers de la variable " concavity_worst " :
##  [1] 1.1050 1.2520 0.9608 0.8216 0.8488 0.7892 0.8489 0.8402 0.9034 0.9019
## [11] 1.1700 0.9387
## Outliers de la variable " concave.points_worst " :
## numeric(0)
## Outliers de la variable " symmetry_worst " :
##  [1] 0.4601 0.6638 0.4378 0.4366 0.4218 0.4667 0.4264 0.4761 0.4270 0.4863
## [11] 0.4670 0.4228 0.5440 0.4882 0.5774 0.5166 0.4753 0.4432 0.4724 0.5558
## [21] 0.4245 0.4824 0.4677
## Outliers de la variable " fractal_dimension_worst " :
```

```
##  [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 0.1297 0.1297 0.1403 0.1249
## [21] 0.1252 0.1364 0.1409 0.1240
```

Podemos apreciar que disponemos de bastantes *outliers*. De los cuales destacamos los dos valores de *radius_se* que parecen ser los que más notablemente difieren del resto. Una vez analizados estos datos, todos ellos parecen ser valores que pertenecen a la realidad, es decir, son valores anómalos pero no erróneos. Se decide mantener todos los datos pero en un proyecto real, será recomendable consultar al oncólogo especialista sobre estos detalles.

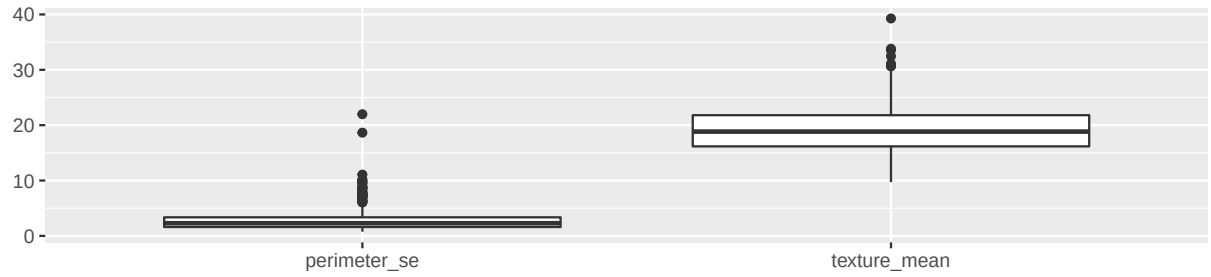
Como se ha mencionado anteriormente, graficaremos las distribuciones de las características así como sus outliers. Para ello, y con el fin de que los diagramas de cajas sean más fácil de interpretar, se han agrupado las variables según su rango de valores.

```
#Cálculo de mediana de cada variable
median.n <- as.vector(sapply(numeric.data,median,na.rm=TRUE))
print(median.n)
```

```
##  [1] 1.337e+01 1.884e+01 8.624e+01 5.511e+02 9.587e-02 9.263e-02 6.154e-02
##  [8] 3.350e-02 1.792e-01 6.154e-02 3.242e-01 1.108e+00 2.287e+00 2.453e+01
## [15] 6.380e-03 2.045e-02 2.589e-02 1.093e-02 1.873e-02 3.187e-03 1.497e+01
## [22] 2.541e+01 9.766e+01 6.865e+02 1.313e-01 2.119e-01 2.267e-01 9.993e-02
## [29] 2.822e-01 8.004e-02
```

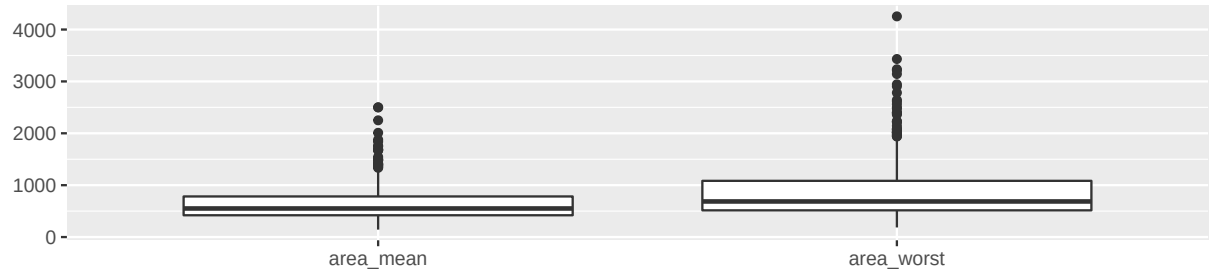
```
#Agrupación de valores según rango de valores
var.e01.1<-numeric.data[,c(13,2)]
var.e01.2<-numeric.data[,c(1,21,22)]
var.e02<-numeric.data[,c(4,24)]
var.min1.1<-numeric.data[,c(9,11,25,20)]
var.min1.2<-numeric.data[,c(26,27,29)]
var.min2.1<-numeric.data[,c(5,6,7,8)]
var.min2.2<-numeric.data[,c(10,15,16,17)]
var.min2.3<-numeric.data[,c(18,28,30)]
var.00<-numeric.data[,12]
var.min3<-numeric.data[,c(14,23,3)]
```

```
#Visualización de valores extremos (outliers)
ggplot(stack(var.e01.1), aes(x = ind, y = values))+geom_boxplot()+xlab("")+ylab("")
```

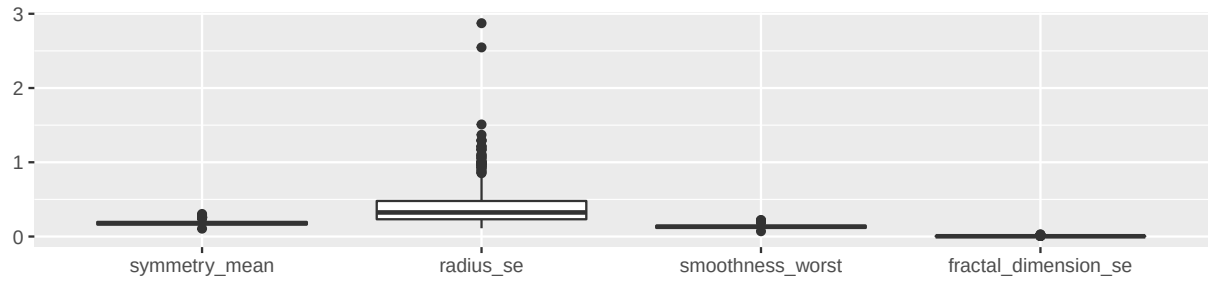



```
ggplot(stack(var.e01.2), aes(x = ind, y = values))+geom_boxplot()+xlab("")+ylab("")
```

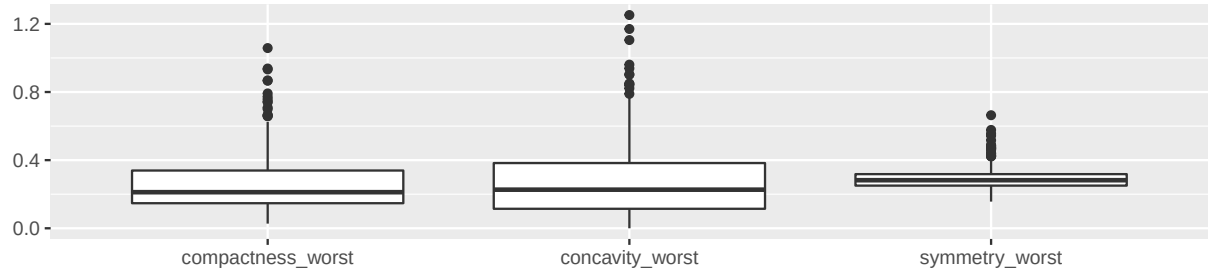

```
ggplot(stack(var.e02), aes(x = ind, y = values))+geom_boxplot()+xlab("")+ylab("")
```



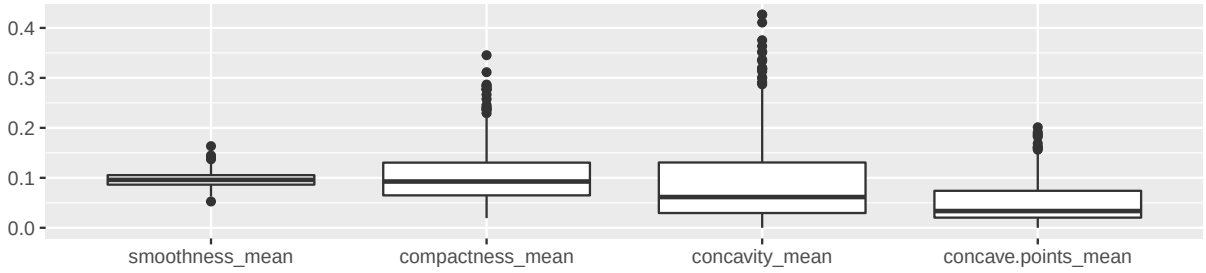

```
ggplot(stack(var.min1.1), aes(x = ind, y = values))+geom_boxplot()+xlab("")+ylab("")
```



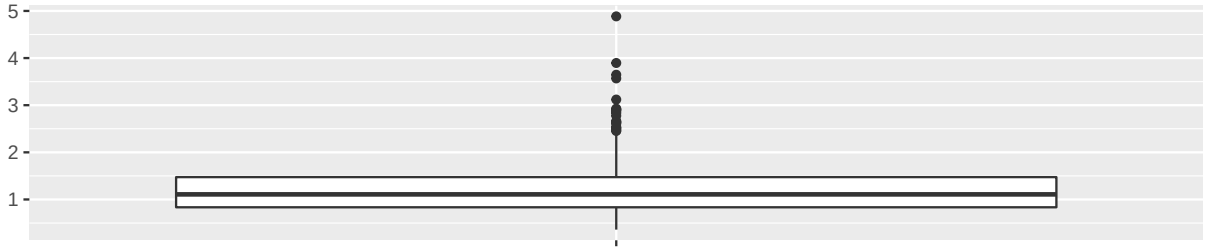

```
ggplot(stack(var.min1.2), aes(x = ind, y = values))+geom_boxplot()+xlab("")+ylab("")
```




```
ggplot(stack(var.min2.1), aes(x = ind, y = values))+geom_boxplot()+xlab("")+ylab("")
```




```
ggplot(numeric.data, aes(x= "", y = numeric.data[,12])) +geom_boxplot()+xlab("")+ylab("")
```




```
ggplot(stack(var.min3), aes(x = ind, y = values))+geom_boxplot()+xlab("")+ylab("")
```

Tras finalizar el proceso de carga, validación y limpieza de datos, guardaremos los datos resultantes en el archivo *data.csv* en la carpeta *data/processed*.

```
#Guardamos los datos preprocesados
write.csv(data,"./data/processed/data.csv")
```

4. Análisis de los datos.

Nuestro principal objetivo es construir un modelo que pueda predecir si un tumor es benigno o maligno según sus características. Para ello, deberemos analizar las características e intentar comprender cuáles tienen valor predictivo.

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

Solamente eliminaremos la identificación *id* y la última caracteristica X que contenía solo valores nulos. La variable *diagnostic* la utilizaremos como factor de grupo.

Antes de proseguir con la comprobación de la normalidad y homogeneidad de la varianza, conviene analizar brevemente la variable que utilizaremos como factor de grupo:

```
#Número de casos benignos vs malignos
data=data[,!colnames(data)=="id"]
attach(data)
```

```
## The following objects are masked from data (pos = 3):
##
##     area_mean, area_se, area_worst, compactness_mean,
##     compactness_se, compactness_worst, concave.points_mean,
##     concave.points_se, concave.points_worst, concavity_mean,
##     concavity_se, concavity_worst, diagnosis,
##     fractal_dimension_mean, fractal_dimension_se,
##     fractal_dimension_worst, perimeter_mean, perimeter_se,
##     perimeter_worst, radius_mean, radius_se, radius_worst,
##     smoothness_mean, smoothness_se, smoothness_worst,
##     symmetry_mean, symmetry_se, symmetry_worst, texture_mean,
##     texture_se, texture_worst
```

```
diagnostic <- plyr::count(data$diagnosis)
print(sprintf("Maligno: %d | Benigno: %d",diagnostic$freq[2],diagnostic$freq[1]))
```

```
## [1] "Maligno: 212 | Benigno: 357"
```

```
#Porcentaje de casos malignos
print(sprintf(
  "Porcentaje de tumores malignos: %.2f%%",round(diagnostic$freq[2]/nrow(data)*100,2)
  ))
```

```
## [1] "Porcentaje de tumores malignos: 37.26%"
```

Tratándose de una base de datos cuyo principal interés es identificar aquellos tumores malignos, el porcentaje de tumores malignos es sorprendentemente bajo. El conjunto de datos no representa en este caso una distribución típica de análisis médico,dispondría de un número elevado de tumores malignos frente a un número algo menor de tumores benignos.

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

Normalidad

Para contrastar que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, aplicaremos el test de Shapiro Wilk en cada variables numérica. Para un nivel de significación prefijado a $\alpha = 0.05$, en el caso de que el p-valor obtenido sea mayor a dicho valor, rechazaremos la hipótesis nula de que la población sigue una distribución normal.

```
for(i in 1:length(numeric.data) ) {
  if (shapiro.test(numeric.data[,i])$p.value<0.05){
  cat('\nRechazamos la hipótesis nula para la variable numérica ',names(numeric.data)[i])
  }
  else{
  cat('\nNo rechazamos la hipótesis nula para la variable numérica', names(numeric.data)[i])
  }
}
```

```
##
## Rechazamos la hipótesis nula para la variable numérica  radius_mean
## Rechazamos la hipótesis nula para la variable numérica  texture_mean
## Rechazamos la hipótesis nula para la variable numérica  perimeter_mean
## Rechazamos la hipótesis nula para la variable numérica  area_mean
## Rechazamos la hipótesis nula para la variable numérica  smoothness_mean
## Rechazamos la hipótesis nula para la variable numérica  compactness_mean
## Rechazamos la hipótesis nula para la variable numérica  concavity_mean
## Rechazamos la hipótesis nula para la variable numérica  concave.points_mean
## Rechazamos la hipótesis nula para la variable numérica  symmetry_mean
## Rechazamos la hipótesis nula para la variable numérica  fractal_dimension_mean
## Rechazamos la hipótesis nula para la variable numérica  radius_se
## Rechazamos la hipótesis nula para la variable numérica  texture_se
## Rechazamos la hipótesis nula para la variable numérica  perimeter_se
## Rechazamos la hipótesis nula para la variable numérica  area_se
## Rechazamos la hipótesis nula para la variable numérica  smoothness_se
## Rechazamos la hipótesis nula para la variable numérica  compactness_se
## Rechazamos la hipótesis nula para la variable numérica  concavity_se
## Rechazamos la hipótesis nula para la variable numérica  concave.points_se
## Rechazamos la hipótesis nula para la variable numérica  symmetry_se
## Rechazamos la hipótesis nula para la variable numérica  fractal_dimension_se
## Rechazamos la hipótesis nula para la variable numérica  radius_worst
## Rechazamos la hipótesis nula para la variable numérica  texture_worst
## Rechazamos la hipótesis nula para la variable numérica  perimeter_worst
## Rechazamos la hipótesis nula para la variable numérica  area_worst
## Rechazamos la hipótesis nula para la variable numérica  smoothness_worst
## Rechazamos la hipótesis nula para la variable numérica  compactness_worst
## Rechazamos la hipótesis nula para la variable numérica  concavity_worst
## Rechazamos la hipótesis nula para la variable numérica  concave.points_worst
## Rechazamos la hipótesis nula para la variable numérica  symmetry_worst
## Rechazamos la hipótesis nula para la variable numérica  fractal_dimension_worst
```

Como era de esperar con presencia de tantos outliers, rechazamos la hipótesis nula para todas las variables.

Homogeneidad de la varianza

Para analizar la homogeneidad de las varianzas respecto al tipo de tumor (maligno/benigno) y teniendo en cuenta que hemos rechazado la hipótesis de que las variables siguen una distribución normal, las dos opciones más adecuadas son el test de Levene o el test Fligner-Killeen. Trabajaremos con este último en el que la hipótesis nula es que las dos varianzas son iguales. Así, para los p-valores superiores a un nivel de significación prefijado a $\alpha = 0.05$, en el caso de que el p-valor obtenido sea mayor a dicho valor, rechazaremos la hipótesis nula de que las varianzas sean iguales.

```
for(i in 3:ncol(data)-1 ) {
  if (fligner.test(data[,i]~diagnosis,data=data)$p.value<0.05){
  cat('\nRechazamos la hipótesis de que las varianzas de ambas
```

```
      muestras sean homogéneas ',names(data)[i])
  }
  else{
  cat('\nNo rechazamos la hipótesis de que las varianzas de ambas
      muestras sean homogéneas', names(data)[i])
  }
}
```

```
##
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  radius_mean
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas texture_mean
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  perimeter_mean
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  area_mean
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas smoothness_mean
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  compactness_mean
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  concavity_mean
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  concave.points_mean
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas symmetry_mean
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  fractal_dimension_mean
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  radius_se
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  texture_se
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  perimeter_se
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  area_se
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  smoothness_se
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  compactness_se
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas concavity_se
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas concave.points_se
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas symmetry_se
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas fractal_dimension_se
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  radius_worst
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas texture_worst
## Rechazamos la hipótesis de que las varianzas de ambas
```

```
##       muestras sean homogéneas  perimeter_worst
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  area_worst
## No rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas smoothness_worst
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  compactness_worst
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  concavity_worst
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  concave.points_worst
## Rechazamos la hipótesis de que las varianzas de ambas
##       muestras sean homogéneas  symmetry_worst
```

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

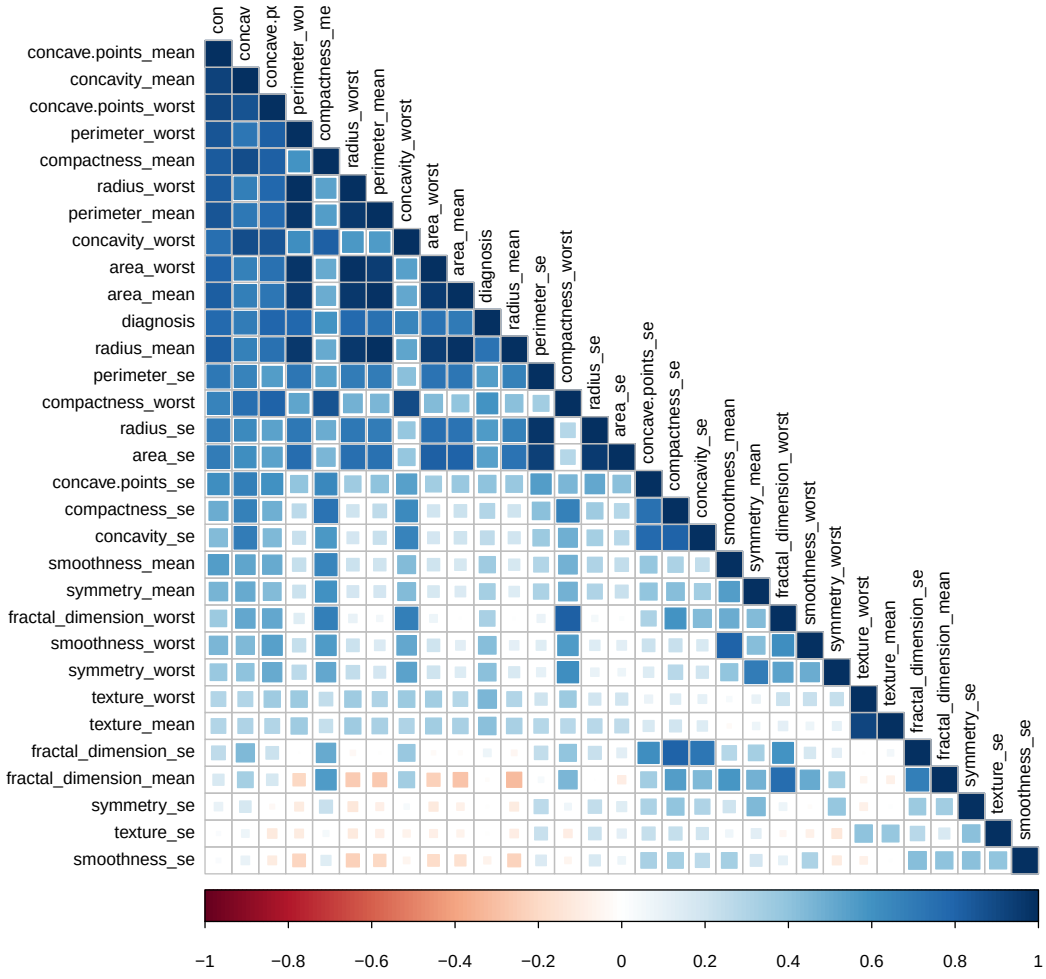
En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

¿Existe correlación entre las variables?

```
numeric.data$diagnosis <- as.integer(factor(data$diagnosis))-1

correlations <- cor(numeric.data,method="pearson")

corrplot(correlations, number.cex = .9, method = "square",
         hclust.method = "ward", order = "FPC",
         type = "lower", tl.cex=0.8,tl.col = "black")
```

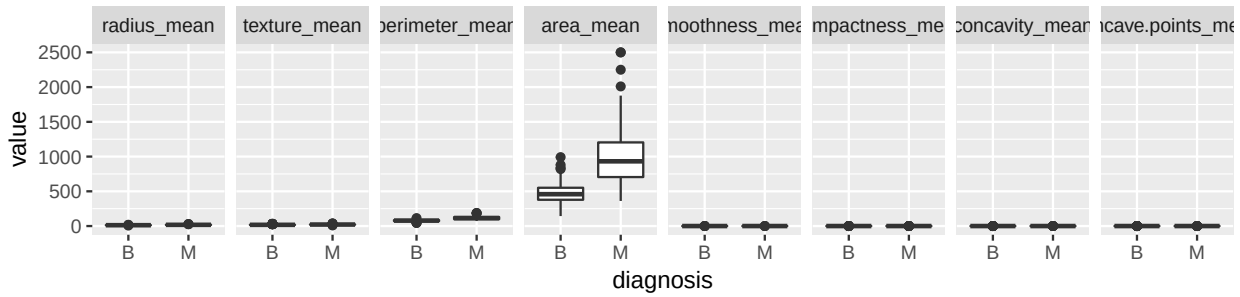


Las variables más correladas (aquellas con tonalidad más oscura) son: *area_worst* y *radius_worst*; *perimeter_mean* y *radius_worst*; *perimeter_worst* y *radius_worst*; *perimeter_mean*, *area_worst*, *area_mean* y *radius_mean* y, por último, *texture_mean* y *texture_worst*.

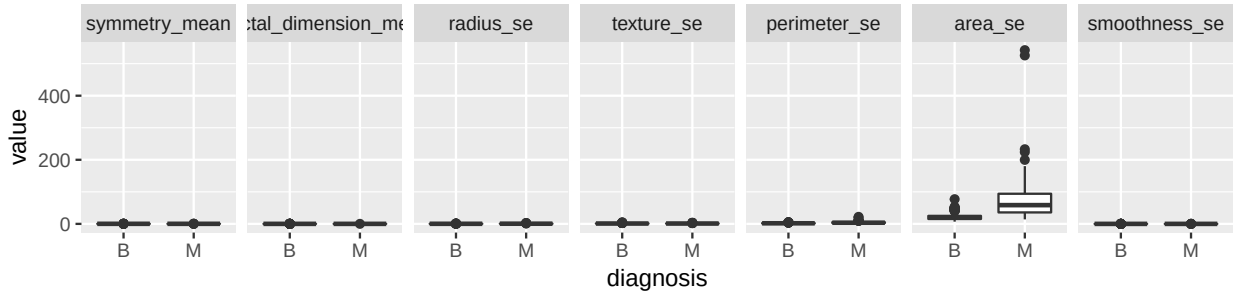
¿Qué variables son las que influyen en mayor medida en la malignidad/benignidad de los datos?

Una vez que hemos analizado qué variables tienen mayor correlación entre ellas, estudiaremos cuáles de ellas ejercen una mayor influencia sobre la malignidad/benignidad del tumor. Comenzaremos graficando boxplots y funciones de densidad de las caracteristicas diferenciando tumores malignos vs benignos con el fin de detectar cuáles presentan una diferencia mayor:

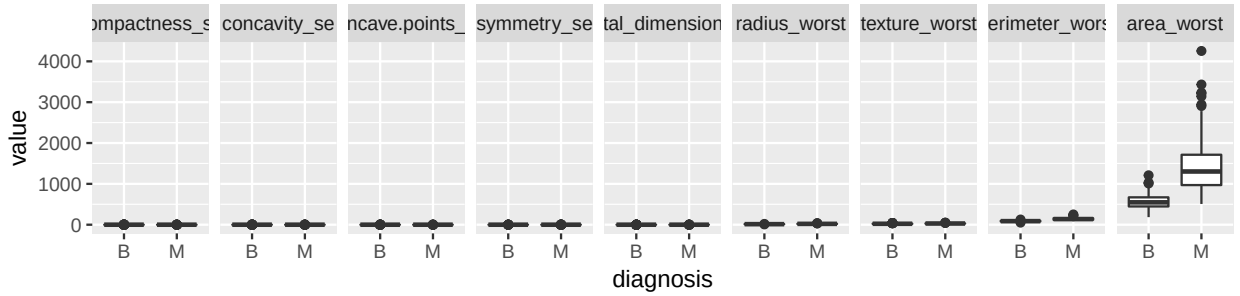
```
#Visualización de valores extremos (outliers) por grupos
mm<-melt(old.data, id=c('id','diagnosis'))
ggplot(mm[c(1:4552),])+geom_boxplot(aes(x=diagnosis, y=value))+facet_grid(.~variable)
```



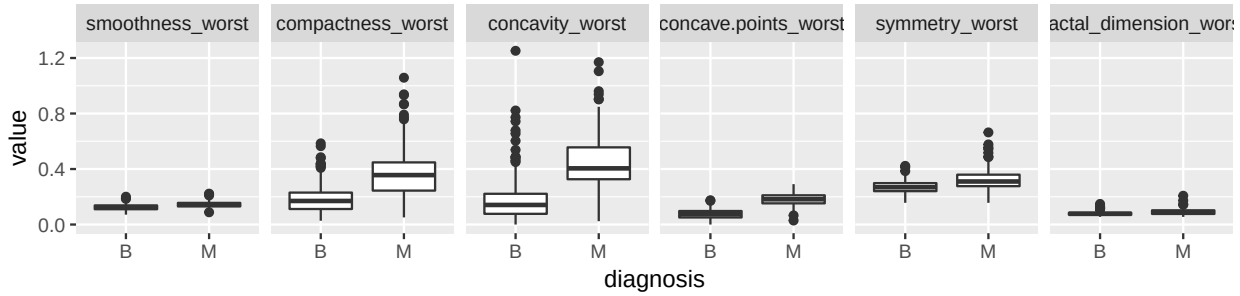

```
ggplot(mm[c(4553:8535),])+geom_boxplot(aes(x=diagnosis, y=value))+facet_grid(.~variable)
```



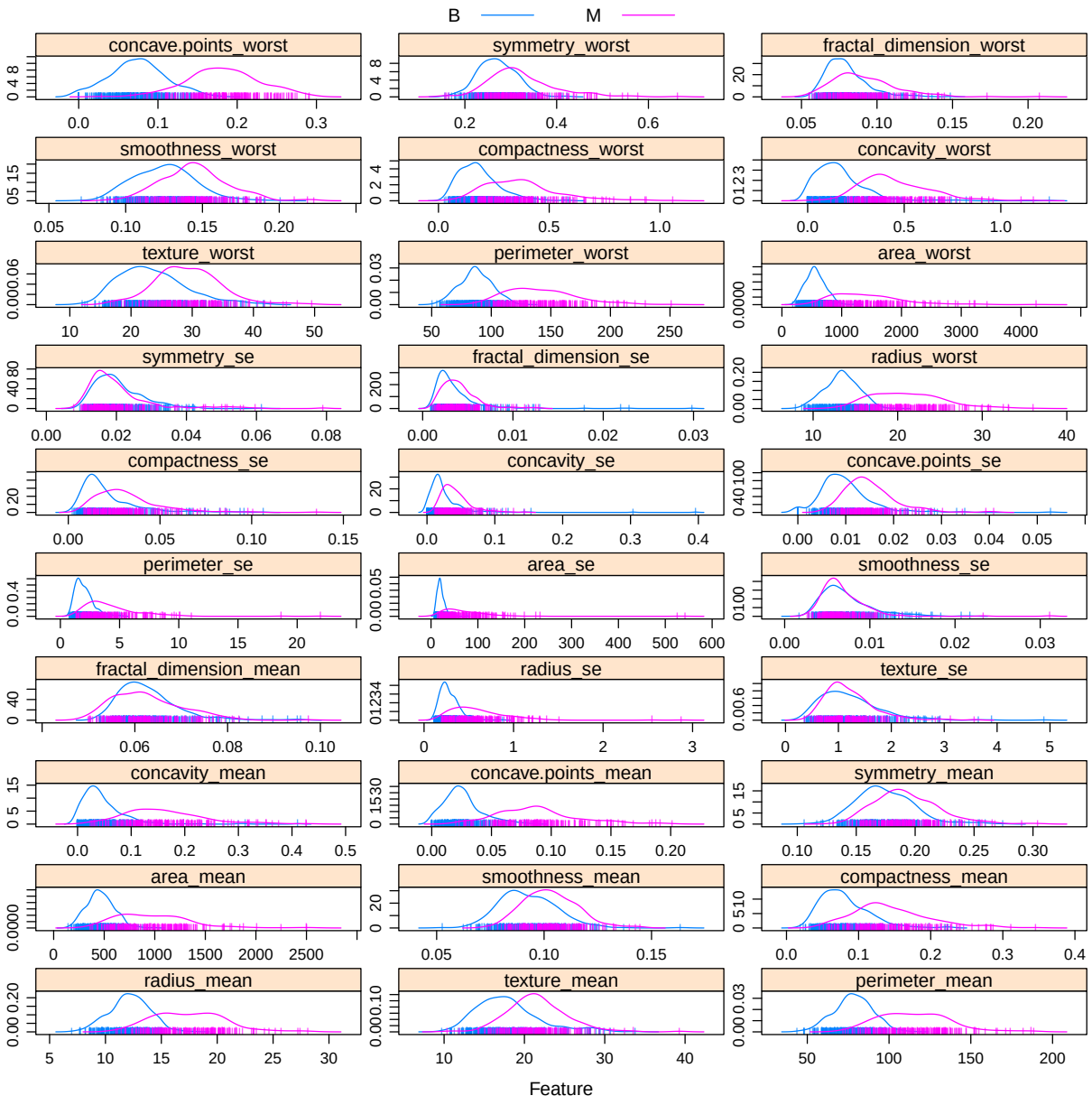

```
ggplot(mm[c(8536:13656),])+geom_boxplot(aes(x=diagnosis, y=value))+facet_grid(.~variable)
```




```
ggplot(mm[c(13657:17070),])+geom_boxplot(aes(x=diagnosis, y=value))+facet_grid(.~variable)
```




```
featurePlot(x=old.data[,c(3:32)], y=old.data[,2], plot="density",
            scales = list(x = list(relation="free"), y = list(relation="free"),cex=0.8),
            layout = c(3,10), auto.key = list(columns = 2), pch = "|")
```



No observamos ninguna variable perfectamente separada pero destacamos que tenemos separaciones bastante buenas para las siguientes características: *concavo.points_worst*, *concavity_worst*, *perimeter_worst*, *area_mean* y $perimeter_mean$. También tenemos superposición ajustada para algunos de los valores, como *simetry_se*, $smoothness_se$.

Construcción del modelo predictivo

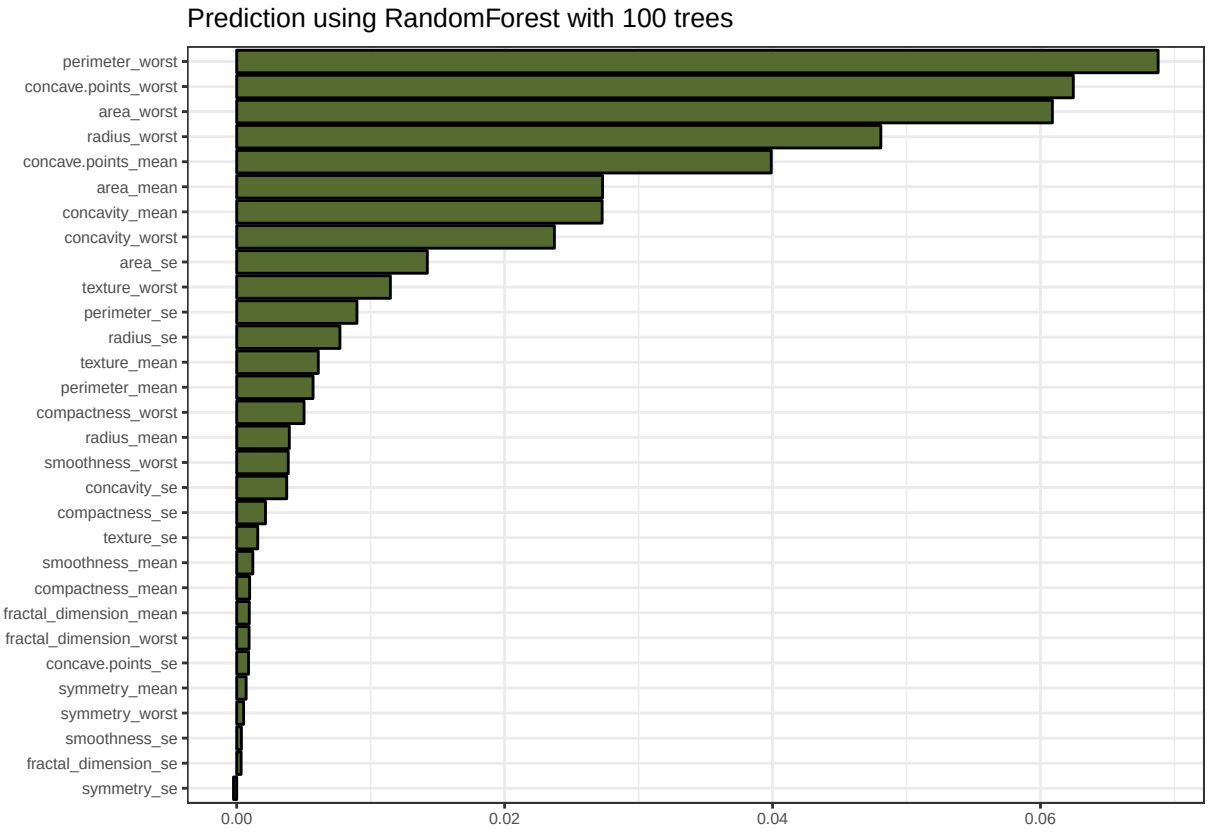
Dadas las características de nuestro dataset, el modelo predictivo escogido ha sido Random Forest (bosque de árboles). Esta, es una técnica de agregación que mejora la precisión en la clasificación mediante la incorporación de aleatoriedad en la construcción de cada clasificador individual. Construiremos un bosque de 100 árboles y entrenaremos el modelo con una partición del dataset original para así poder testearlo con el dataset restante (80%-20%).

```
numeric.data <- old.data[,2:32]
numeric.data$diagnosis = as.integer(factor(numeric.data $diagnosis))-1
set.seed(314)
#Separamos el conjunto de datos en un conjunto de entrenamiento y test
training.index <- sample(1:nrow(numeric.data), 0.8 * nrow(numeric.data))
training.data = numeric.data[training.index,]
test.data = numeric.data[-training.index,]
random.forest<-randomForest::randomForest(diagnosis ~ ., data=training.data,ntree=100,
                                          keep.forest=TRUE, importance=TRUE)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
variable.importance <- data.frame(random.forest$importance)

ggplot(variable.importance,
       aes(x=reorder(rownames(variable.importance),X.IncMSE), y=X.IncMSE)) + geom_bar(stat="identity", f
coord_flip() + theme_bw(base_size = 8) +
  labs(title="Prediction using RandomForest with 100 trees")+ylab("")+xlab("")
```



Las características más importantes son *perimeter_worst*,*concave.points_worst*, *area_worst*, *radius_worst*,*concave.points_mean*, *area_mean*, *concavity_mean* y $concavity_worst$ respectivamente.

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.4.4
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
breast.pred <- predict(random.forest,test.data)

print(sprintf("Area under curve (AUC) : %.3f",auc(test.data$diagnosis, breast.pred)))
```

```
## [1] "Area under curve (AUC) : 0.985"
```

Obtenemos un AUC muy alto, lo cual es realmente satisfactorio ya que el AUC es la probabilidad de que, tomados un caso positivo y uno negativo al azar, el scoring del modelo para el primero sea superior al segundo.

5. Resolución del problema. A partir de los resultados obtenidos.

Tras el exhaustivo análisis del dataset, la limpieza, la validación y la construcción del árbol, concluímos que las características más importantes son *perimeter_worst*,*concave.points_worst*, *area_worst*, *radius_worst*,*concave.points_mean*, *area_mean*, *concavity_mean* y $concavity_worst$ respectivamente. Además, afirmamos que hemos podido construir un modelo capaz de predecir la malignidad/benignidad del tumor con una exactitud muy satisfactoria.

7. Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

El código en R esta incluido en este fichero con extensión rmd y tambien se puede descargar en GitHub desde la siguiente dirección:

https://github.com/iboda001/PRA2_limpiezayvalidacion/blob/master/Codigo/LimpiezaValidacion.R