PRAC 2: Limpieza y análisis de datos

Alejandro Medina, Federico Clavijo

Mayo 20

---

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El data set seleccionado corresponde a training set (train.csv) del respositorio Titanic - Machine Learning from Disaster encontrado en la página kaggle (link: https://www.kaggle.com/competitions/titanic/data (https://www.kaggle.com/competitions/titanic/data)). Este data set corresponde a un fichero con 891 registros de pasajeros que abordaron el Titanic. Las (12) varibables que lo compone son:

- PassengerId: (integer) Número entero de identificación de cada pasajero
- Survived: (integer) Binario indicando: 0 - No sobreviviente / 1 - Sobreviviente
- Pclass: (integer) Número entero de la clase 1, 2 o 3
- Name: (character) Nombre del pasajero de la siguiente manera “Apellido, Mr./Mrs/Miss. Nombres (Otro nombre)”
- Sex: (character) Sexo female / male
- Age: (integer) Número entero de la edad
- SibSp: (integer) Número de hermanos o conyugue abordo
- Parch: (integer) Número de padres o hijos abordo
- Ticket: (character) Número del tiquete algunos con letra al comienzo y espacio
- Fare: (numeric) Tarifa
- Cabin: (character) Cabina
- Embarked: (character) Lugar donde el pasajero abordó (S) Southampton, (C) Cherbourg, and (Q) Queenstown

Los objetivos que nos hemos trazado en esta actividad corresponden a:

- Determinar si hay correlación entre las variables género y el hecho de supervivivencia
- Calcular el modelo que permita predecir la superviviencia de pasajeros en función de las variables: género, edad, Pclass, y Embarked.

2. Integración y selección de los datos de interés a analizar. Puede ser el resultado de adicionar diferentes datasets o una subselección útil de los datos originales, en base al objetivo que se quiera conseguir.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Carga del data set orieginal a la variable dt_original
dt_original <- read.csv("train.csv", sep=",")
#head(dt_original)

# Número de filas del data set
n <- nrow(dt_original)
#n

# Data set resultante filtrando quitando algunas columnas no requeridas.
dt_1 <- select(dt_original, - SibSp, - Ticket, - Fare )
attach(dt_1)
head(dt_1)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                  Name    Sex Age Parch Cabin
## 1                             Braund, Mr. Owen Harris   male  22     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     0   C85
## 3                              Heikkinen, Miss. Laina female  26     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     0  C123
## 5                            Allen, Mr. William Henry   male  35     0
## 6                                    Moran, Mr. James   male  NA     0
##   Embarked
## 1        S
## 2        C
## 3        S
## 4        S
## 5        S
## 6        Q
```

```
# Resumen descriptivo de las variables del data set seleccionado
summary(dt_1)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            Parch           Cabin
##  Length:891         Min.   : 0.42   Min.   :0.0000   Length:891
##  Class :character   1st Qu.:20.12   1st Qu.:0.0000   Class :character
##  Mode  :character   Median :28.00   Median :0.0000   Mode  :character
##                     Mean   :29.70   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :6.0000
##                     NA's   :177
##    Embarked
##  Length:891
##  Class :character
##  Mode  :character
##
##
##
##
```

---

3. Limpieza de los datos

3.1. ¿Los datos contienen ceros o elementos vacíos? Gestiona cada uno de estos casos.

```
# Se identifican los siguientes elementos a ser ajustados:
# La varible Age tiene 177 NA's, para lo cual se decide pasar a 0 para luego manejarlo co
mo valor extremo.

# Por estética se quita los decimales en la edad
dt_1[, 6] <-trunc(dt_1$Age)
# dt_1[,6]

# Pasar NA's a 0
dt_1$Age[is.na(dt_1$Age)] <- 0

# Comprobación de que han sido tratados los NA's identificados
# summary(dt_1$Age)

# La variable Cabin tiene espacios vacíos (null) serán llenados con 0
i=1
for (i in 1:n) {
  if ((dt_1[i, 8]) == "") {
    dt_1[i, 8] <- 0
  }
i+1
}

# Comprobación de que han sido tratados los espacios vacíos identificados
# head(dt_1$Cabin)

# valores faltantes de "Embarked" se cambian por X
dt_1["Embarked"][dt_1["Embarked"] == ""] <- "X"
```

3.2. Identifica y gestiona los valores extremos.

```
# Se identifican los siguientes elementos a ser ajustados:
# La variable Age tiene datos extremos incluyendo el 0, razón por la cual se decide que p
ara edades inferiores a 5 años, se realizará imputación por la media aritmética en Age. P
ara ello se imputará con la respectiva media aritmética según si el nombre indica si es M
aster/Miss o Mr/Mrs, para identificar si es adulto o niño/adolecente

# Cálculo del promedio de Age según sea Master/Miss, Mr/Mrs, Rev o Dr
i=0
n_master_miss=0
n_mr_mrs=0
n_rev=0
n_dr=0
suma_age_master_miss=0
suma_age_mr_mrs=0
suma_age_rev=0
suma_age_dr=0
for (i in 1:n) {
  if(grepl(pattern = '(Master|Miss)', dt_1[i,4])) {
    n_master_miss=n_master_miss+1
    suma_age_master_miss = suma_age_master_miss+(dt_1[i, 6])
  } else if (grepl(pattern = '(Mr|Mrs)', dt_1[i,4])) {
    n_mr_mrs=n_mr_mrs+1
    suma_age_mr_mrs = suma_age_mr_mrs+(dt_1[i, 6])
  } else if (grepl(pattern = '(Rev)', dt_1[i,4])) {
    n_rev=n_rev+1
    suma_age_rev = suma_age_rev+(dt_1[i, 6])
  } else {
    n_dr=n_dr+1
    suma_age_dr = suma_age_dr+(dt_1[i, 6])
    }
i+1
}

# El promedio de la edad de las personas Master / Miss que correspondería niñ@s y jóvenes
promedio_age_master_miss=trunc(suma_age_master_miss/n_master_miss)
#promedio_age_master_miss

# El promedio de la edad de las personas Mr / Mrs que correspondería hombres y mujeres
promedio_age_mr_mrs=trunc(suma_age_mr_mrs/n_mr_mrs)
#promedio_age_mr_mrs

# El promedio de la edad de las personas Rev que correspondería Curas
promedio_age_rev=trunc(suma_age_rev/n_rev)
#promedio_age_rev

# El promedio de la edad de las personas Dr que correspondería Doctor@s
promedio_age_dr=trunc(suma_age_dr/n_dr)
#promedio_age_dr
```

```
# Asignación de la variable Age a valores que sean inferiores a 5 años, dependiendo si la persona es Master, Miss, Mr, Mrs, Rev o Dr.
i=1
for (i in 1:n) {
  if ((dt_1[i, 6]) <= 0 & (grepl(pattern = '(Master|Miss)', dt_1[i,4]))) {
    dt_1[i, 6] <- promedio_age_master_miss
  } else if ((dt_1[i, 6]) <= 0 & (grepl(pattern = '(Mr|Mrs)', dt_1[i,4]))) {
    dt_1[i, 6] <- promedio_age_mr_mrs
  } else if ((dt_1[i, 6]) <= 0 & (grepl(pattern = '(Rev)', dt_1[i,4]))) {
    dt_1[i, 6] <- promedio_age_rev
  } else if ((dt_1[i, 6]) <= 0 & (grepl(pattern = '(Dr)', dt_1[i,4]))) {
    dt_1[i, 6] <- promedio_age_dr
  }
i+1
}

# Comprobar que no hay edades inferiores a 5, no se muestra por tema de límite de página s.
# summary(dt_1$Age)

# Se comprueba por cada variable la existencia de valores no coherentes. Y se identifica  que no se requiere más ajustes
head(dt_1 %>% distinct(Sex, .keep_all = FALSE))
```

```
##      Sex
## 1   male
## 2 female
```

```
head(dt_1 %>% distinct(Survived, .keep_all = FALSE))
```

```
##   Survived
## 1        0
## 2        1
```

```
head(dt_1 %>% distinct(Pclass, .keep_all = FALSE))
```

```
##   Pclass
## 1      3
## 2      1
## 3      2
```

```
head(dt_1 %>% distinct(Parch, .keep_all = FALSE))
```

```
##   Parch
## 1     0
## 2     1
## 3     2
## 4     5
## 5     3
## 6     4
```

```
head(dt_1 %>% distinct(Cabin, .keep_all = FALSE))
```

```
##   Cabin
## 1     0
## 2   C85
## 3  C123
## 4   E46
## 5    G6
## 6  C103
```

```
head(dt_1 %>% distinct(Embarked, .keep_all = FALSE))
```

```
##   Embarked
## 1        S
## 2        C
## 3        Q
## 4        X
```

---

4. Análisis de los datos.

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

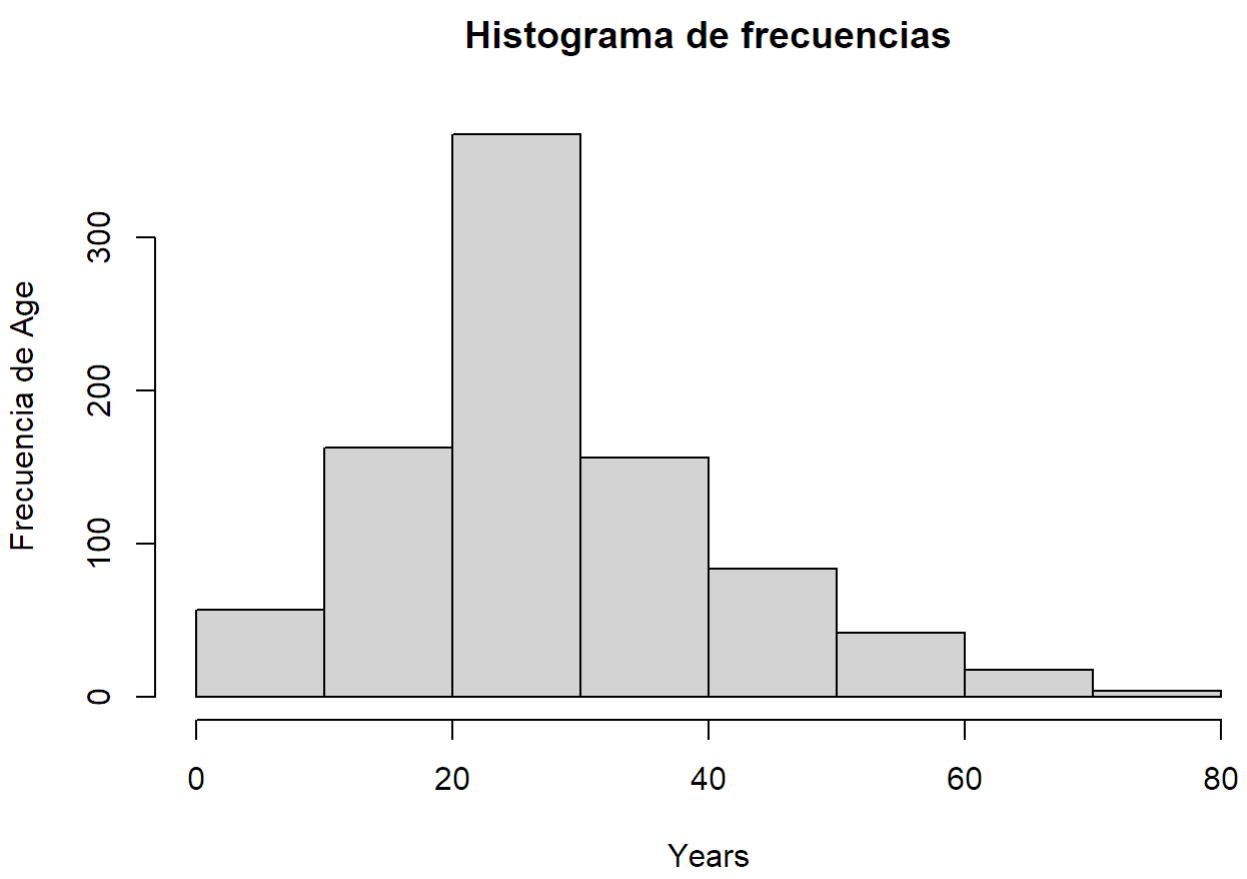
```
# Según los objetivos iniciales se requieren los siguientes subgrupos de datos:
# - Objetivo 1: Determinar si hay correlación entre las variables género y el hecho de su pervivivencia
dt_obj1 <- select(dt_1, Survived, Sex)

# Análisis requeridos para el objetivo 1:
# - Analizar gráfica de las variables Survived y Sex
# - Estudio de correlación lineal entre las variables

# - Objetivo 2: Calcular el modelo que permita predecir la superviviencia de pasajeros en función de las variables: género, edad, Pclass, y Embarked.
dt_obj2 <- select(dt_1, - Cabin, - PassengerId, - Name, - Parch )

dt_obj2_SST <- table(Survived,Sex)
# prop.table(dt_obj2_SST, margin = 1)

dt_obj2_SAT <- table(Survived,Age)
# prop.table(dt_obj2_SAT, margin = 1)

dt_obj2_SPcT <- table(Survived,Pclass)
# prop.table(dt_obj2_SPcT, margin = 1)

dt_obj2_SET <- table(Survived,Embarked)
# prop.table(dt_obj2_SET, margin = 1)

# Analisis requeridos para el objetivo 2:
# - Análizar gráfica de las variables de dt_1
# - revisar que variables son predictoras de la variable sobrevive
# - revision de cada variable y hacer una mapa de correlación
# - Discretizar variables, analizar y finalmente generar gráfico de arbol
# - Estudio de correlación logística entre las variables. Aplicando modelos de regresión
 logística binaria.
```

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

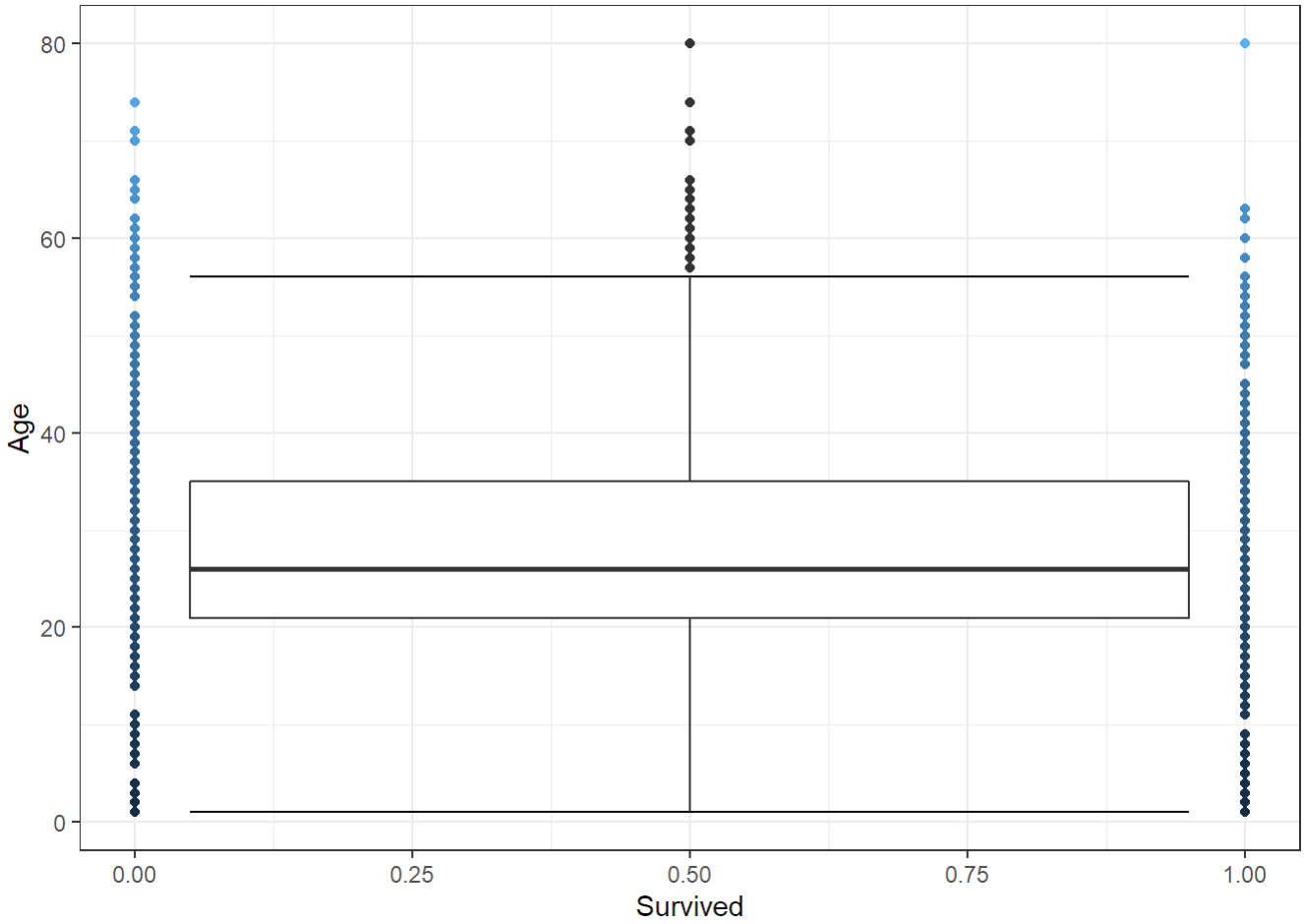
```
library(dplyr)

# Se comprueba se tiene un comportamiento normal las variables de interés: Ages
hist(dt_1$Age, main = "Histograma de frecuencias", ylab = "Frecuencia de Age", xlab = "Ye
ars")
```




```
# Esta es la gráfica de la variable Age es de los valores sin imputar

# Análisis de homogeneidad de la variables (Hay que ajustar mejor la gráfica)
ggplot(data = dt_obj2, aes(x = Survived, y = Age, colour = Age)) +
  stat_boxplot(geom = "errorbar", width = 0.2) +
  geom_boxplot() +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none")
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```




```
# Se identifican datos outliers, media, los cuartiles 1(25%), 2(50%) y 3(75%) de las vari
ables Survived y Age

# Prueba con la función de R t.test, donde identificamos que los valores son iguales de m
edias de la población.
t.test(dt_obj2$Age, dt_obj2$Survived, alternative="less", var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  dt_obj2$Age and dt_obj2$Survived
## t = 63.646, df = 1780, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 28.93355
## sample estimates:
##  mean of x  mean of y
## 28.5881033  0.3838384
```

```
# # El test no encuentra diferencias significativas entre las medias de los dos grupos

var.test(x = dt_obj2[dt_obj2$Survived == "1", "Age"],
         y = dt_obj2[dt_obj2$Survived == "0", "Age"] )
```

```
##
##  F test to compare two variances
##
## data:  dt_obj2[dt_obj2$Survived == "1", "Age"] and dt_obj2[dt_obj2$Survived == "0", "A
ge"]
## F = 1.1531, num df = 341, denom df = 548, p-value = 0.1404
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9542208 1.3993989
## sample estimates:
## ratio of variances
##           1.153072
```

```
# El test no encuentra diferencias significativas entre las varianzas de los dos grupos
```

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. Aplicar al menos tres métodos de análisis diferentes.

Para resolución del objetivo 1: Determinar si hay correlación entre las variables género y el hecho de supervivivencia

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

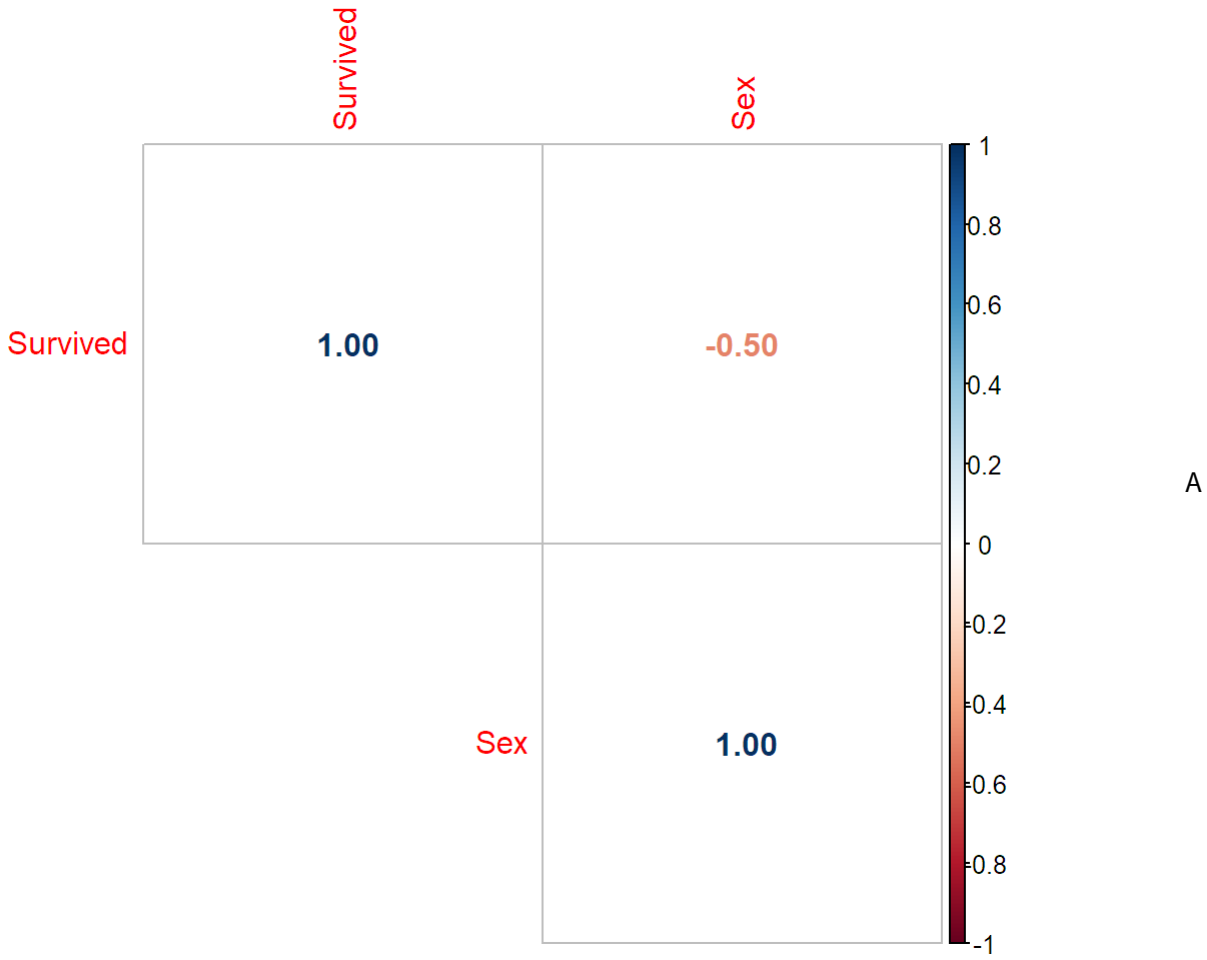
```
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.1.3
```

```r
# renombrar variables male y female por 1 y 0 respectivamente
i=1
for (i in 1:nrow(dt_obj2)) {
  if (dt_obj2[i, 3] == "male") {
      dt_obj2[i, 3] <- 1
  } else {
      dt_obj2[i, 3] <- 0
    }
i+1
}

# Convertir variable Sex a numeric
dt_obj2$Sex <- as.numeric(dt_obj2$Sex)

# Ingreso de datos variables
matrix_cor_variables <- data.frame(
  "Survived" = dt_obj2$Survived,
  "Sex" = dt_obj2$Sex
)

#head(matrix_cor_variables)

# comando calcular matriz de correlación
round(cor(matrix_cor_variables),2)
```

```
##          Survived   Sex
## Survived     1.00 -0.54
## Sex         -0.54  1.00
```

```r
# matriz de correlación de forma gráfica
correlacion_variables<-round(cor(matrix_cor_variables), 1)
corrplot(correlacion_variables, method="number", type="upper")
```



A

partir de la matriz de correlación se identifica que las dos variables Sex y Survived tienen una relación correlación negativa moderada que signifca que de manera moderada el comportamiento de una de las variables puede explicar el comportamiento de la otra.

Para resolución del objetivo 2: Calcular el modelo que permita predecir la superviviencia de pasajeros en función de las variables

```
# Análisis Valores de la V de Cramér y Phi entre 0.1 y 0.3 nos indican que la asociación
 estadística es baja, y entre 0.3 y 0.5 se puede considerar una asociación media. Finalme
nte, superior a 0.5 la asociación estadística entre las variables sería alta.

Phi(dt_obj2_SST)
```

```
## [1] 0.5433514
```

```
CramerV(dt_obj2_SST)
```

```
## [1] 0.5433514
```

```
Phi(dt_obj2_SAT)
```

```
## [1] 0.3819382
```

```
CramerV(dt_obj2_SAT)
```

```
## [1] 0.3819382
```

```
Phi(dt_obj2_SPcT)
```

```
## [1] 0.3398174
```

```
CramerV(dt_obj2_SPcT)
```

```
## [1] 0.3398174
```

```
Phi(dt_obj2_SET)
```

```
## [1] 0.1824838
```

```
CramerV(dt_obj2_SET)
```

```
## [1] 0.1824838
```

Trabajamos la variable AGE y aleatorizamos el dataframe

```
set.seed(1)
data_random <- dt_obj2[sample(nrow(dt_obj2)),]
```

Para la futura evaluación del árbol de decisión, es necesario dividir el conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento es el subconjunto del conjunto original de datos utilizado para construir un primer modelo.

La variable por la que clasificaremos es el campo de si el pasajero sobrevivió o no, que está en la primera columna. De esta forma, tendremos un conjunto de datos para el entrenamiento y uno para la validación

```
set.seed(666)
y <- data_random[,1]
X <- data_random[,2:5]
```

De forma dinámica podemos definir una forma de separar los datos en función de un parámetro, en este caso del “split_prop”. Definimos un parámetro que controla el split de forma dinámica en el test.

```
split_prop <- 3
max_split<-floor(nrow(X)/split_prop)
tr_limit <- nrow(X)-max_split
ts_limit <- nrow(X)-max_split+1

trainX <- X[1:tr_limit,]
trainy <- y[1:tr_limit]
testX <- X[(ts_limit+1):nrow(X),]
testy <- y[(ts_limit+1):nrow(X)]
```

Después de una extracción aleatoria de casos es altamente recomendable efectuar un análisis de datos mínimo para asegurarnos de no obtener clasificadores sesgados por los valores que contiene cada muestra. En este caso, verificaremos que la proporción del supervivientes es más o menos constante en los dos conjuntos:

```
summary(trainX);
```

```
##      Pclass          Sex             Age          Embarked
##  Min.   :1.000   Min.   :0.0000   Min.   : 1.00   Length:594
##  1st Qu.:2.000   1st Qu.:0.0000   1st Qu.:20.00   Class :character
##  Median :3.000   Median :1.0000   Median :26.00   Mode  :character
##  Mean   :2.325   Mean   :0.6246   Mean   :28.22
##  3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:35.00
##  Max.   :3.000   Max.   :1.0000   Max.   :74.00
```

```
summary(trainy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3855  1.0000  1.0000
```

```
summary(testX)
```

```
##      Pclass          Sex             Age          Embarked
##  Min.   :1.000   Min.   :0.0000   Min.   : 1.00   Length:296
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:22.00   Class :character
##  Median :3.000   Median :1.0000   Median :26.00   Mode  :character
##  Mean   :2.274   Mean   :0.6926   Mean   :29.34
##  3rd Qu.:3.000   3rd Qu.:1.0000   3rd Qu.:35.00
##  Max.   :3.000   Max.   :1.0000   Max.   :80.00
```

```
summary(testy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3818  1.0000  1.0000
```

Verificamos fácilmente que no hay diferencias graves que puedan sesgar las conclusiones.

Adiconalmente, se crea el árbol de decisión usando los datos de entrenamiento (no hay que olvidar que la variable outcome es de tipo factor)

```
if(!require(C50)){
    install.packages('C50', repos='http://cran.us.r-project.org')
    library(C50)
}
```

```
## Loading required package: C50
```

```
## Warning: package 'C50' was built under R version 4.1.3
```

```
summary(trainy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3855  1.0000  1.0000
```

```
trainy = as.factor(trainy)
model <- C50::C5.0(trainX, trainy,rules=TRUE )
summary(model)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Jun 07 01:01:43 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 594 cases (5 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (371/68, lift 1.3)
##  Sex > 0
##  ->  class 0  [0.815]
##
## Rule 2: (329/84, lift 1.2)
##  Pclass > 2
##  ->  class 0  [0.743]
##
## Rule 3: (17, lift 2.5)
##  Pclass <= 2
##  Age <= 15
##  ->  class 1  [0.947]
##
## Rule 4: (71/12, lift 2.1)
##  Sex <= 0
##  Embarked in {C, Q}
##  ->  class 1  [0.822]
##
## Rule 5: (223/62, lift 1.9)
##  Sex <= 0
##  ->  class 1  [0.720]
##
## Default class: 0
##
##
## Evaluation on training data (594 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       5  103(17.3%)   <<
##
##
```

```
##     (a)   (b)    <-classified as
##    ----  ----
##     347    18    (a): class 0
##      85   144    (b): class 1
##
##
##  Attribute usage:
##
##  100.00% Sex
##   58.25% Pclass
##   11.95% Embarked
##    2.86% Age
##
##
## Time: 0.0 secs
```

---

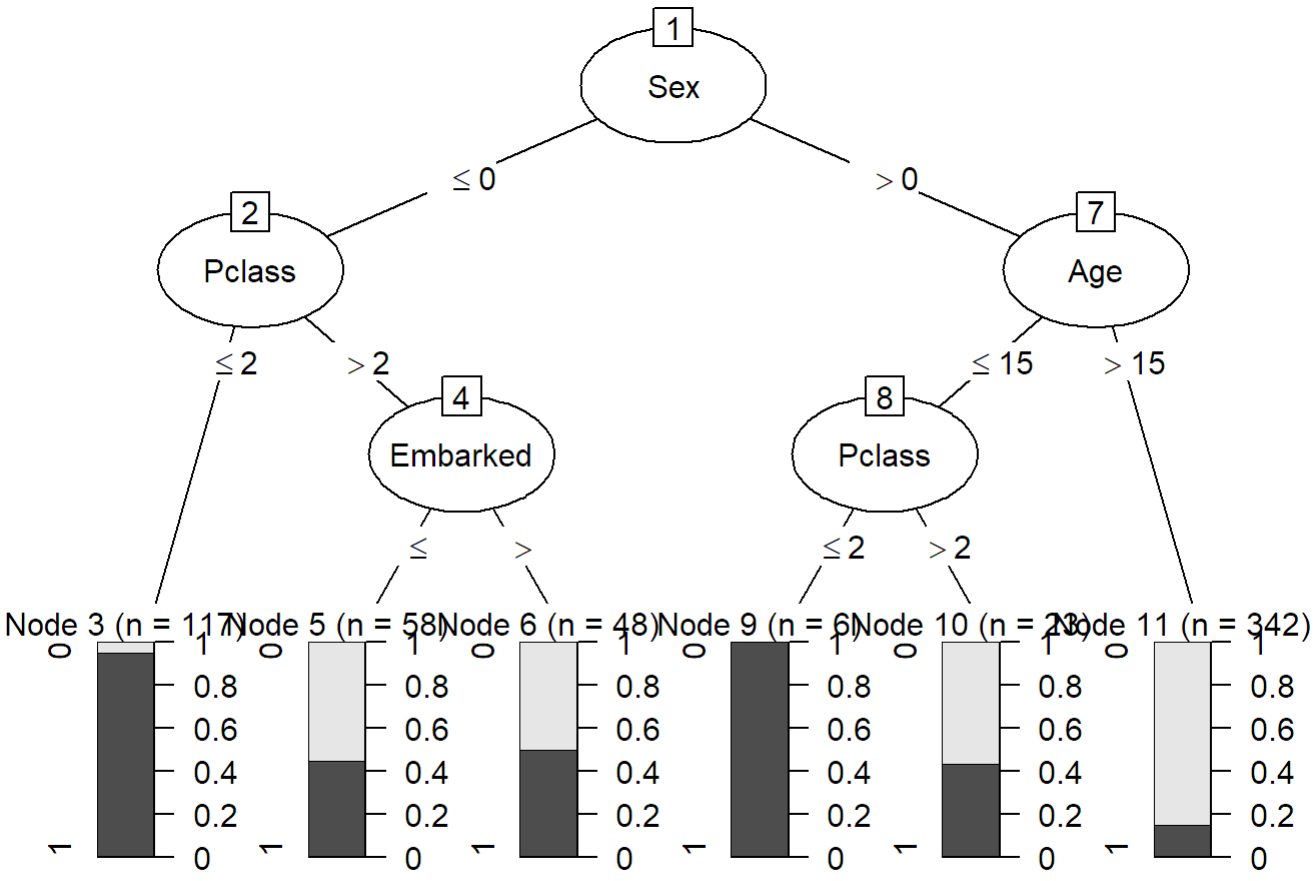
5. Representación de los resultados a partir de tablas y gráficas. Este apartado se puede responder a lo largo de la práctica, sin necesidad de concentrar todas las representaciones en este punto de la práctica.

A continuación, mostramos el árbol obtenido.

```
model <- C50::C5.0(trainX, trainy)
plot(model)
```

```
## Warning in partysplit(varid = as.integer(i), breaks = as.numeric(j[1]), : NAs
## introducidos por coerción
```

```
## Warning in .bincode(as.numeric(x), breaks = unique(c(-Inf,
## breaks_split(split), : NAs introducidos por coerción
```




```
# Exportar archivo final
write.csv(x = dt_obj2, file = "PRAC2_fclavijo_amedinau_dt_out.csv", row.names = TRUE)
```

---

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

Dentro del modelo creado, podemos ver a partir de Errors, el número y porcentaje de casos mal clasificados en el subconjunto de entrenamiento. El árbol obtenido clasifica erróneamente 103 de los 594 casos dados, una tasa de error del 17.3%.

A partir del árbol de decisión de dos hojas que hemos modelado, se pueden extraer las siguientes reglas de decisión (gracias a rules=TRUE podemos imprimir las reglas directamente):

- Sex = “Hombre” → Muere. Validez: 81,5%
- Pclass “3ª” → Muere. Validez: 74,3%
- Pclass “1ª”, “2ª” y AGE “menos e iguales a 15 años” → Sobrevive. Validez: 94,7%
- Sex = “Mujer” y Embarked “C y Q” → Sobrevive. Validez: 82,2%
- Sex = “Mujer” → Sobrevive. Validez: 72%

Por tanto, podemos concluir que el conocimiento extraído y cruzado con el análisis visual se resume en que la sobrevivencia esta sujeta a que sea mujer del embarque C y Q. Y tambien los menores de 15 años de la clase 1ra y 2da.

```
predicted_model <- predict( model, testX, type="class" )
print(sprintf("La precisión del árbol es: %.4f %%",100*sum(predicted_model == testy) / le
ngth(predicted_model)))
```

```
## [1] "La precisión del árbol es: 82.0946 %"
```

---

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 se encuentra presente en el documento denominado: Desarrollo_codigo_VF.Rmd

---

Tabla de contribuciones

Contribuciones Investigación previa: - AM Alejandro Medina Uicab, FC Federico Clavijo López

Redacción de las respuestas: - AM Alejandro Medina Uicab, FC Federico Clavijo López

Desarrollo del código: - AM Alejandro Medina Uicab, FC Federico Clavijo López

```
```