DESCRIPCIÓN DE LA PRÁCTICA

Presentación

En aquesta pràctica s'elabora un cas pràctic orientat a aprendre a identificar les dades rellevants per un projecte analític i usar les eines d'integració, neteja, validació i anàlisi de les mateixes. Per fer aquesta pràctica haureu de treballar en grups de fins a 3 persones, o si preferiu, també podeu fer-ho de manera individual. Haureu de lliurar un sol fitxer amb l'enllaç Github (https://github.com) on es trobin les solucions incloent els noms dels components de l'equip. Podeu utilitzar la Wiki de Github per descriure el vostre equip i els diferents arxius que corresponen a la vostra entrega. Cada membre de l'equip haurà de contribuir amb el seu usuari Github

Objetivos

Els objectius concrets d'aquesta pràctica són:

- Aprendre a aplicar els coneixements adquirits i la seva capacitat de resolució de problemes en entorns nous o poc coneguts dintre de contextos més amplis o multidisciplinaris.
- Saber identificar les dades rellevants i els tractaments necessaris (integració, neteja i
- validació) per dur a terme un projecte analític.
- Aprendre a analitzar les dades adequadament per abordar la informació continguda en les dades.
- Identificar la millor representació dels resultats per tal d'aportar conclusions sobre el problema plantejat en el procés analític.
- Actuar amb els principis ètics i legals relacionats amb la manipulació de dades en funció de l'àmbit d'aplicació.
- Desenvolupar les habilitats d'aprenentatge que els permetin continuar estudiant d'una manera que haurà de ser en gran manera autodirigida o autònoma.
- Desenvolupar la capacitat de cerca, gestió i ús d'informació i recursos en l'àmbit de la ciència de dades.

Descripción de la práctica

L'objectiu d'aquesta activitat serà el tractament d'un dataset, que pot ser el creat a la pràctica 1o bé qualsevol dataset lliure disponible a Kaggle (https://www.kaggle.com). Alguns exemples de dataset amb els que podeu treballar són:

● Red Wine Quality (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009).

● Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic).

L'últim exemple correspon a una competició activa a Kaggle de manera que, opcionalment, podeu aprofitar el treball realitzat durant la pràctica per entrar en aquesta competició.

Resolución Práctica

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

Para la realización de la práctica se ha utilizado el dataset Titanic obtenido del repositorio https://www.kaggle.com/c/titanic

Con el análisis de estos datos podríamos estudiar si se podría predecir qué pasajeros se iban a salvar a priori analizando los atributos disponibles. Y también qué variables tienen más peso a la hora de que una persona se salve o no.

El dataset "de entrenamiento" contiene la información de 891 pasajeros y está constituido por los siguientes 12 atributos:

Variable	Definición	tipo	Valores
PassengerId	identificador del pasajero	int	autonumérico que empieza con el valor 1
Survived	si el pasajero sobrevivió finalmente o no. Esta es la variable que queremos predecir en base a los otros atributos	int	0 = No (Muere), 1 = SÍ (Vive)
Pclass	clase a la que pertenece el billete	int	1 = 1ra, 2 = 2da, 3 = 3ra
Name	nombre del pasajero	string	

Sex	Sexo de los pasajeros	string	“male” “female”
Age	Años que tiene el pasajero 0.x si tiene menos de un año, en caso contrario es un entero	float	1,2,3,...99
sibsp	número de hermanos y maridos/esposas que tiene el pasajero a bordo. Los prometidos no cuentan	int	1,2,3,...
parch	número de padres e hijos que tiene el pasajero a bordo. Algunos niños viajan solos o con su niñera (no cuenta)	int	1,2,3..
ticket	número de ticket	int	
fare	precio del billete	float	
cabin	número de la cabina en la que se queda el pasajero	string	C85,...
embarked	puerto en el que embarcó el pasajero	string	C = Cherbourg, Q = Queenstown, S = Southampton

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

Los atributos del dataset corresponden a los datos de los pasajeros y a priori todos son potenciales atributos interesantes para realizar el análisis. Sin embargo, podemos prescindir de algunos campos que pensamos que aportan poca información para la realización del análisis que hemos planteado:
Los atributos que podemos descartar son:

- PassengerId: identificador del pasajero
- Name: este campo a priori no es relevante, porque el nombre no aporta mucha información, pero en este caso lo vamos a mantener hasta que obtengamos un nuevo campo “title” que hace referencia al título de las personas (Mr, Mrs,..) y que obtendremos del nombre del pasajero, ya que estos contienen el título, Por ejemplo: Mrs. Nicholas (Adele Achem)
- Ticket: en número del ticket tampoco es relevante para el análisis ya que no aporta información. Además los rangos de los números billetes suelen corresponder con las clases de los mismo. Los de primera clase empiezan por 1, los de 2da por 2 y los de 3ra por 3.

El resto de campos no tenemos claros si serán de utilidad o no, después del análisis podremos tener la certeza y podemos proceder a eliminarlos en una segunda fase.

En este caso aparece el campo Title que es el que almacena el título de las personas y que comentamos en el punto 3.3

```
#eliminar columnas no relevantes para el estudio
data$PassengerId<-NULL
data$Name<-NULL
data$Ticket<-NULL
data$Fare<-NULL
str(data)
```

```
> str(data)
'data.frame':           891 obs. of  9 variables:
 $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age     : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp   : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch   : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Cabin   : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
 $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
 $ Title   : chr  "Mr" "Mrs" "Miss" "Mrs" ...
```

3. Limpieza de los datos

Para la realización de la práctica usaremos el lenguaje R. Antes de realizar cualquier acción debemos cargar el dataset en la variable “data” desde el fichero csv usando la función read.csv, una vez cargados los datos usamos la funciones str (obtener la estructura del dataset y los tipos de los datos)y summary (obtener un resumen de los datos, con los valores nulos, distribuciones,...), head (para listar los 10 primeros elementos) para ver si se ha cargado bien el fichero y ver un resumen de los datos.
El dataset contiene 891 observaciones y 12 variables.

R

```
data<-read.csv("titanic_train.csv",header=T,sep=",")
attach(data)

str(data)
summary(data)
head (data , 10)
```

Consola R

```
> attach(data)
> str(data)
'data.frame':	891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
 $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
> summary(data)
```

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
Min. : 1.0	Min. :0.0000	Min. :1.000	Abbing, Mr. Anthony : 1	female:314	Min. : 0.42	Min. :0.000	Min. :0.0000	1601 : 7
1st Qu.:223.5	1st Qu.:0.0000	1st Qu.:2.000	Abbott, Mr. Rossmore Edward: 1	male :577	1st Qu.:20.12	1st Qu.:0.000	1st Qu.:0.0000	347082 : 7
Median :446.0	Median :0.0000	Median :3.00	Abbott, Mrs. Stanton : 1		Median :28.00	Median :0.000	Median :0.0000	CA. 2343: 7
Mean :446.0	Mean :0.3838	Mean :2.309	Abelson, Mr. Samuel : 1		Mean :29.70	Mean :0.523	Mean :0.3816	3101295 : 6
3rd Qu.:668.5	3rd Qu.:1.0000	3rd Qu.:3.000	Abelson, Mrs. Samuel : 1		3rd Qu.:38.00	3rd Qu.:1.000	3rd Qu.:0.0000	347088 : 6
Max. :891.0	Max. :1.0000	Max. :3.000	Adahl, Mr. Mauritz Nils Martin 1		Max. :80.00	Max. :8.000	Max. :6.0000	CA 2144 : 6
			(Other) :885		NA's :177			(Other) :852

Fare	Cabin	Embarked
Min. : 0.00	:687	: 2
1st Qu.: 7.91	B96 B98	: 4 C:168
Median : 14.45	C23 C25 C27	: 4 Q: 77
Mean : 32.20	G6	: 4 S:644
3rd Qu.: 31.00	C22 C26	: 3
Max. :512.33	D :	3
	(Other) :186	

```
> head (data , 10)
```

	PasId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500		S
2	2	1	1	Cumings, Mrs. John Bradley	female	38	1	0	PC 17599	71.2833	C85	C
3	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7.9250		S
4	4	1	1	Futrelle, Mrs. Jacques Heath	female	35	1	0	113803	53.1000	C123	S
5	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.0500		S
6	6	0	3	Moran, Mr. James	male	NA	0	0	330877	8.4583		Q
7	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
8	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21.0750		S
9	9	1	3	Johnson, Mrs. Oscar W)	female	27	0	2	347742	11.1333		S
10	10	1	2	Nasser, Mrs. Nicholas	female	14	1	0	237736	30.0708		C

3.1. Los datos contienen elementos vacíos? cómo gestionamos estos casos?

Para comprobar si hay datos nulos, usaremos la información ya obtenida mediante las funciones “summaty” y “head”, junto con la funcion “is.na” que sirve para comprobar si existen campos vacios. También comprobamos cuántos campos de tipo string hay vacíos, tal y como se desprende de la función headm en la que se ven varias observaciones con el campo cabin vacío
Podemos verificar que para el atributo edad tenemos 177 valores vacíos (NA), 687 campos “” para el campo Cabin y 2 campos vacíos para el campo Embarked

R

```
# vemos si hay campos sin valor
colSums(is.na(data))
# vemos si hay campos con valor ""
colSums(data=="")
```

Consola R

```
> # vemos si hay campos sin valor
> colSums(is.na(data))
PassengerId    Survived      Pclass        Name         Sex         Age       SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           0           0         177           0           0           0           0           0           0

> colSums(data=="")
PassengerId    Survived      Pclass        Name         Sex         Age       SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           0           0          NA           0           0           0           0         687           2
```

Embarked
En este caso al disponer solo de dos observaciones con el campo Embarked vacío, optamos por asignar a estas dos observaciones el valor “S” que corresponde a Southamptonm por ser el elemento que más se repite con 644 apariciones, frente al valor C que aparece 168 veces y el valor Q que aparece 77 veces

Cabin
En este caso, la mayoría de los pasajeros no disponen de camarote, ya que sólo disponían de ella los viajeros de primera clase, para ello nos inventamos una cabina ficticia Z0 para todos aquellos que no tenían camarote asignado

Edad
Este es el caso más complejo, ya que hay 177 pasajeros sin edad y es complicado obtener la edad.
Para resolver esta situación nos podemos plantear varias alternativas:

- Eliminar el atributo si vemos que no es relevante para el análisis
- Eliminar las observaciones que dispongan de ese campo vacío
- Asignar la edad media de todos los pasajeros de manera genérica
- Asignar la edad media en función del tratamiento (title) de la persona (Mr,Mrs, Miss), la explicación de la creación del campo tratamiento se explica en el punto 3.3.

En nuestro caso optamos por la última opción, ya que de este modo no restringimos los datos a procesar y además se asigna una edad acorde a su título. Filtramos los pasajeros por título y luego obtenemos la media (mean) para cada título.
La edad media asignada para cada título es:

- Master: 4,57
- Miss: 21,94
- Mr: 32,96
- Mrs:35,61

Una vez obtenemos la media, asignamos dicho valor a los campos NA y verificamos que ya no queda ningún campo NA.

```
# calculamos la media para cada título
 Masterfilter <-filter(data,data$Title=="Master"  )
 z <- Masterfilter$Age
 mean(z, na.rm = TRUE)

 Missfilter <-filter(data,data$Title=="Miss"  )
 z <- Missfilter$Age
 mean(z, na.rm = TRUE)

 Mrfilter <-filter(data,data$Title=="Mr"  )
 z <- Mrfilter$Age
 mean(z, na.rm = TRUE)

 Mrsfilter <-filter(data,data$Title=="Mrs"  )
 z <- Mrsfilter$Age
 mean(z, na.rm = TRUE)

for (i in 1:length(data$Age))
 {
  if ( is.na(data$Age[i]) && data$Title[i]=="Master") {
   data$Age[i]<-4.57
  }
  if ( is.na(data$Age[i]) && data$Title[i]=="Miss") {
   data$Age[i]<-21.94
  }
  if ( is.na(data$Age[i]) && data$Title[i]=="Mr") {
   data$Age[i]<-32.96
  }
  if ( is.na(data$Age[i]) && data$Title[i]=="Mrs") {
   data$Age[i]<-35.61
  }

 }

# vemos si hay campos sin valor
colSums(is.na(data))
```

Consola R

```
> # calculamos la media para cada titulo
>   Masterfilter <-filter(data,data$Title=="Master"  )
> z <- Masterfilter$Age
> mean(z, na.rm = TRUE)
[1] 4.574167
> Missfilter <-filter(data,data$Title=="Miss"  )
> z <- Missfilter$Age
> mean(z, na.rm = TRUE)
[1] 21.94521
> Mrfilter <-filter(data,data$Title=="Mr"  )
> z <- Mrfilter$Age
```

```
> mean(z, na.rm = TRUE)
[1] 32.96301
> Mrsfilter <-filter(data,data$Title=="Mrs"  )
> z <- Mrsfilter$Age
> mean(z, na.rm = TRUE)
[1] 35.61947


> colSums(is.na(data))
Survived  Pclass     Sex     Age   SibSp   Parch   Cabin Embarked   Title
       0       0       0       0       0       0       0        0       0
```

3.2. Identificación y tratamiento de valores extremos

Los valores extremos son aquellos que parecen no ser congruentes sin los comparamos con el resto de los datos (suelen estar muy lejos de la media , > 3 varianzas). Para identificarlos, podemos usar diferentes alternativas:

- representar un diagrama de caja por cada variable y ver qué valores distan mucho del rango intercuartílico
- utilizar la función boxplots.stats() para cada atributo. Esta función nos devuelve los valores extremos que encuentra

Aplicamos la función boxplots.stats a los atributos que no son discretos (factores), ya que no tiene sentido para atributos discretos.

En este caso todos los valores que se obtienen son valores posibles, tanto para la edad,fare, como para los parientes que viajan en el barco, por tanto no hay que hacer ningún tratamiento para los datos actuales.

```
#2   Encontrar valores
anómalos#####################################################
  boxplot.stats(data$Pclass)$out
  boxplot.stats(data$Age)$out
  boxplot.stats(data$SibSp)$out
  boxplot.stats(data$Parch)$out
  boxplot.stats(data$Fare)$out
```

Consola R

```
> #2   Encontrar valores anómalos#####################################################
>   boxplot.stats(data$Pclass)$out
integer(0)
> boxplot.stats(data$Age)$out
 [1] 58.00 66.00 65.00  0.83 59.00 71.00 70.50  1.00 61.00  1.00  1.00 58.00 59.00 62.00 58.00 63.00 65.00  0.92 61.00 60.00  1.00  1.00
64.00 65.00  0.75 63.00 58.00 71.00 64.00 62.00 62.00 60.00 61.00 57.00 80.00  0.75 58.00 70.00 60.00 60.00
[41] 70.00  0.67 57.00  1.00  0.42  1.00 62.00  0.83 74.00
> boxplot.stats(data$SibSp)$out
 [1] 3 4 3 3 4 5 3 4 5 3 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 3 5 3 5 3 4 4 3 3 5 4 3 4 8 4 3 4 8 4 8
> boxplot.stats(data$Parch)$out
  [1] 1 2 1 5 1 1 5 2 2 1 1 2 2 2 1 2 2 2 3 2 2 1 1 1 1 2 1 1 2 2 1 2 2 2 1 2 2 2 1 2 1 1 2 1 1 1 2 1 1 2 1 2 2 2 1 1 2 2 1 2 1 2 1 1 1 1 1 1 2 1
2 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2 2 3 4 1 2 1 1 2 1 2 1 2 1 1 2
[121] 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 4 1 1 2 1 2 1 1 2 5 2 1 1 1 2 1 5 2 1 1 1 2 1 6 1 2 1 2 1 1 1 1 1 1 1 3 2 1 1 1 1 2 1 2 3 1 2 1 2 2 1 1 2 1 2 1 2
1 1 1 2 1 1 2 1 2 1 1 1 1 3 2 1 1 1 1 5 2
> boxplot.stats(data$Fare)$out
  [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000 263.0000  77.2875 247.5208  73.5000  77.2875
79.2000  66.6000  69.5500  69.5500 146.5208  69.5500 113.2750  76.2917  90.0000  83.4750  90.0000  79.2000  86.5000
 [27] 512.3292  79.6500 153.4625 135.6333  77.9583  78.8500  91.0792 151.5500 247.5208 151.5500 110.8833 108.9000  83.1583
262.3750 164.8667 134.5000  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000 263.0000  75.2500  69.3000 135.6333
 [53]  82.1708 211.5000 227.5250  73.5000 120.0000 113.2750  90.0000 120.0000 263.0000  81.8583  89.1042  91.0792  90.0000
```

```
78.2667 151.5500  86.5000 108.9000  93.5000 221.7792 106.4250  71.0000 106.4250 110.8833 227.5250  79.6500 110.8833
 [79]  79.6500  79.2000  78.2667 153.4625  77.9583  69.3000  76.7292  73.5000 113.2750 133.6500  73.5000 512.3292  76.7292
211.3375 110.8833 227.5250 151.5500 227.5250 211.3375 512.3292  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583
[105] 211.3375  79.2000  69.5500 120.0000  93.5000  80.0000  83.1583  69.5500  89.1042 164.8667  69.5500  83.1583
>
```

3.3. Creación de nuevos atributos/columnas

Pensamos que para el ejercicio puede ser interesante saber si se trata de un señor, señora, señorita o señorito, ya que supuestamente las señoras y niños en caso de naufragio a priori tienen más posibilidades de sobrevivir porque son los primeros en subir a los botes salvavidas.
Creamos el campo title y le asignamos su título en base al texto que aparece en el nombre de las personas.

Estas son las asignaciones que hemos creado

- Don, Major, Master, Capt, Jonkheer, Rev, Col, Mr, Dr --> Mr
- Mrs, Countless, Mne --> Mrs
- Ms, Mlle,Miss --> Miss
- Master --> Master

verificamos que no se queda ningún pasajero sin clasificar.

R

```
#añadido atributo Title  con la denominación de las personas

# Don, Major, Master, Capt, Jonkheer, Rev, Col, Mr, Dr --> Mr
# Mrs, Countless, Mne --> Mrs
# Ms, Mlle,Miss --> Miss
# Master --> Master

Don =grep("Don", data$Name)
for (i in 1:length(Don)) {  data$Title[Don[i]]="Mr"}
Major =grep("Major", data$Name)
for (i in 1:length(Major)) {  data$Title[Major[i]]="Mr"}
capt =grep("Capt", data$Name)
for (i in 1:length(capt)) {  data$Title[capt[i]]="Mr"}
Jonkheer =grep("Jonkheer", data$Name)
for (i in 1:length(Jonkheer)) {  data$Title[Jonkheer[i]]="Mr"}
Rev =grep("Rev", data$Name)
for (i in 1:length(Rev)) {  data$Title[Rev[i]]="Mr"}
Col =grep("Col", data$Name)
for (i in 1:length(Col)) {  data$Title[Col[i]]="Mr"}
Mr =grep("Mr.", data$Name)
for (i in 1:length(Mr)) {  data$Title[Mr[i]]="Mr"}
Dr=grep("Dr.", data$Name)
for (i in 1:length(Dr)) {  data$Title[Dr[i]]="Mr"}

mrs= grep("Mrs.", data$Name)
for (i in 1:length(mrs)) {  data$Title[mrs[i]]="Mrs"}
Countess= grep("Countess", data$Name)
for (i in 1:length(Countess)) {  data$Title[Countess[i]]="Mrs"}
Mme= grep("Mme", data$Name)
for (i in 1:length(Mme)) {  data$Title[Mme[i]]="Mrs"}

Mlle=grep("Mlle.", data$Name)
for (i in 1:length(Mlle)) {  data$Title[Mlle[i]]="Miss"}
Ms=grep("Ms.", data$Name)
for (i in 1:length(Ms)) {  data$Title[Ms[i]]="Miss"}
Miss=grep("Miss.", data$Name)
```

```
for (i in 1:length(Miss)) {  data$Title[Miss[i]]="Miss"}

Master =grep("Master.", data$Name)
for (i in 1:length(Master)) {  data$Title[Master[i]]="Master"}

# vemos si hay campos sin valor
colSums(is.na(data))
```

Consola R

```
PassengerId   Survived     Pclass       Name        Sex        Age      SibSp      Parch     Ticket       Fare      Cabin   Embarked      Title
          0          0          0          0          0        177          0          0          0          0          0          0          0
```

3.4 Discretización de los datos

El objetivo de este procesamiento es discretizar la variables que tienen unos valores acotados para que no sean tratadas como variables numéricas. En nuestro caso las variables que son consideradas enteras pero en realidad tienen valores discretos son Survived, plcass, y title

Aplicamos la función factor para discretizar las variables y luego usamos la función str para comprobar que las variables ahora son de tipo factor

R

```
#realizar alguna discreción de los datos
#se seleccionan las variables que tendría sentido aplicar una proceso de discretización
 apply(data,2, function(x) length(unique(x)))
# Discretizar las variables con pocas clases
cols<-c("Survived","Pclass",”Title”)
for (i in cols){
  data[,i] <- as.factor(data[,i])
}
str(data)
```

Consola R

```
'data.frame':	891 obs. of  13 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
 $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581 ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : Factor w/ 149 levels "","A10","A14",..: NA 83 NA 57 NA NA 131 NA NA NA ...
 $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
 $ Title      : Factor w/ 3 levels "Miss","Mr","Mrs": 2 3 1 3 2 2 2 NA 3 3 ...
```

4. Análisis de los datos.

4.1. Selección del grupo de datos a analizar/comparar (planificación de los análisis de aplicar

De todas los campos a analizar elegimos los atributos que nos parece más interesante analizar.

campo clase (Pclass)
Pclass = 1
Pclass = 2
Pclass = 3

campo sexo (Sex)
Sex = male
Sex = female

Campo embarcado (Embarked)
Embarked=S
Embarked=C
Embarked=Q

Campo título (Title)
Title= Mr
Title= Mrs
Title= Miss
Title= Master

R

```
# 4 ######### Análisis de los datos ###########################################################################

# Agrupación por clase
Pclass.1 <- data[data$Pclass == "1",]
Pclass.2 <- data[data$Pclass == "2",]
Pclass.3 <- data[data$Pclass == "3",]

#agrupación por sexo
Sex.male <-data[data$Sex=="male",]
Sex.female <-data[data$Sex=="female",]

#agrupación por embarked
Embarked.S <-data[data$Embarked=="S",]
Embarked.C <-data[data$Embarked=="C",]
Embarked.Q <-data[data$Embarked=="Q",]

#agrupacion por título
Title.Mr <- data[data$Title=="Mr",]
Title.Mrs <- data[data$Title=="Mrs",]
Title.Miss <- data[data$Title=="Miss",]
Title.Master <- data[data$Title=="Master",]
```

Consola R

```
> # Agrupación por clase
> Pclass.1 <- data[data$Pclass == "1",]
> Pclass.2 <- data[data$Pclass == "2",]
> Pclass.3 <- data[data$Pclass == "3",]
> #agrupación por sexo
> Sex.male <-data[data$Sex=="male",]
> Sex.female <-data[data$Sex=="female",]
> #agrupación por embarked
> Embarked.S <-data[data$Embarked=="S",]
> Embarked.C <-data[data$Embarked=="C",]
> Embarked.Q <-data[data$Embarked=="Q",]
> #agrupacion por título
> Title.Mr <- data[data$Title=="Mr",]
> Title.Mrs <- data[data$Title=="Mrs",]
> Title.Miss <- data[data$Title=="Miss",]
> Title.Master <- data[data$Title=="Master",]
>
```

4.2. Comprobación de la normalidad y homogeneidad de la varianza

Normalidad

El objetivo de esta comprobación es asegurarnos que las variables que toman valores cuantitativos siguen una distribución normal (Anderson-Darling).

Comprobamos que para cada pasajero se obtiene un valor > al nivel prefijado de 0,05.

Si se cumple podemos decir que el atributo sigue una distribución normal, y no la sigue en caso contrario.

En nuestro caso, ninguna de las 4 variables numéricas siguen una distribución normal:

R

```
#4.2. Comprobación de la normalidad y homogeneidad de la varianza ########################################
library(nortest)
alpha = 0.05
col.names = colnames(data)
for (i in 1:ncol(data)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
  if (is.integer(data[,i]) | is.numeric(data[,i])) {
    p_val = ad.test(data[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
      if (i < ncol(data) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
}
```

Consola R

```
Variables que no siguen una distribución normal:
Age, SibSp, Parch, Fare,
```

Homogeneidad

Como en nuestro caso lo que queremos calcular es si los pasajeros sobreviven o no y esta variable al no ser numérica, no se puede aplicar la homogeneidad de las varianzas aplicando el test de Fligner-killeen.

Pero para ver su aplicación comparamos la homogeneidad de las variables Fare respecto a pclass. Al obtener un valor de p-valor menor a 0,05, la varianza de ambas variables no es homogénea.

```
R
#homoeneidad
fligner.test(Fare ~ Pclass, data = data)
```

```
Consola R
> fligner.test(Fare ~ Pclass, data = data)

        Fligner-Killeen test of homogeneity of variances

data:  Fare by Pclass
Fligner-Killeen:med chi-squared = 365.8, df = 2, p-value < 2.2e-16

>
```

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y del objetivo de estudio, aplicamos pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

¿Qué variables cuantitativas influyen más en la supervivencia?

Realizamos un análisis de correlación entre las distintas variables para determinar cuáles de ellas tienen una mayor influencia sobre la supervivencia o no de los pasajeros. Para ello usamos el coeficiente de correlación de Spearman, puesto que hemos visto que nuestros datos no siguen una distribución normal. Las variables fare es la que tiene una mayor correlación con la supervivencia

```

```

```
#4.3  pruebas estadísticas

corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
# Calcular el coeficiente de correlación para cada variable cuantitativa
# con respecto al campo "precio"
for (i in 1:(ncol(data) - 1)) {
```

```
  if (is.integer(data[,i]) | is.numeric(data[,i])) {
    spearman_test = cor.test(data[,i],
                             data[,length(data)],
                             method = "spearman")
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    # Add row to matrix
    pair = matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(data)[i]
  }
}

print(corr_matrix)
```

```
Consola R
> print(corr_matrix)
           estimate      p-value
Survived  1.00000000 0.000000e+00
Pclass   -0.33966794 1.687608e-25
Age      -0.07271755 2.997518e-02
SibSp     0.08887948 7.941431e-03
Parch     0.13826563 3.453591e-05
Fare      0.32373614 3.471228e-23
```

5. Representació dels resultats a partir de taules i gràfiques.

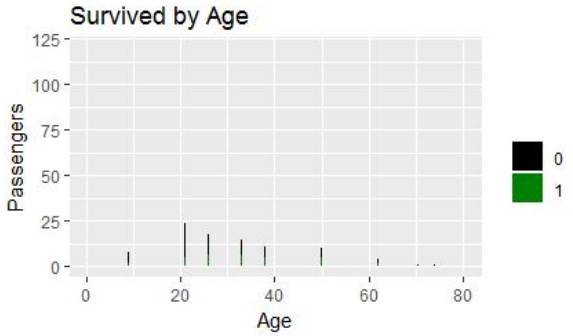
Si nos planteamos qué variables influyen más a la hora de sobrevivir, a priori parece claro que la edad y el sexo son factores diferenciales, ya que las mujeres y niños suelen tener prioridad a la hora de coger un bote. La otra variable que puede influir es la categoría en la que viajan los pasajeros.

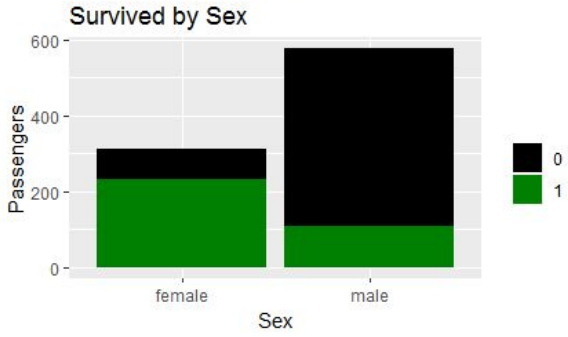
Hipótesis: ¿Influye la edad, sexo y clase a la hora de sobrevivir en caso de accidente?

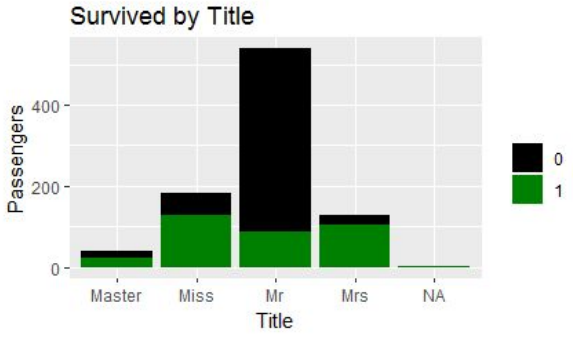
Como en nuestro caso muchas variables no son cuantitativas, vamos a usar algunas herramientas de visualización, para ello utilizaremos algunos paquetes de R que nos proporcionan dicha funcionalidad (ggplot2, grid, gridExtra)

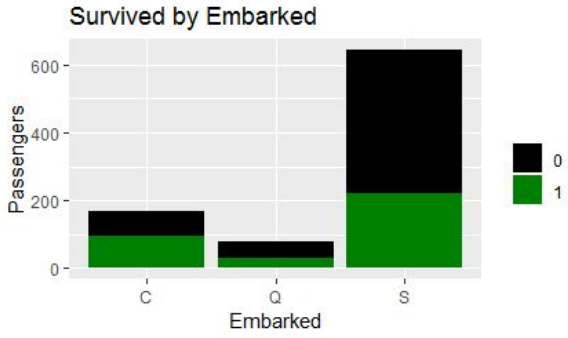
Nos interesa describir la relación entre la supervivencia y cadauna de la variables disponibles. Por ese motivos dibujaremos mediante gráficas de barras, la cantidad de muertos y vivos segun la clase, edad, sexe, titulo. Por otro lado mostraremos también los datos mediante una tabla de contingencia

```
str(data)
grid.newpage()
plotbyClass<-ggplot(data,aes(Pclass,fill=Survived))+geom_bar() +labs(x="Pclass", y="Passengers")+
guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("black","#008000"))+ggtitle("Survived by Class")
plotbyAge<-ggplot(data,aes(Age,fill=Survived))+geom_bar() +labs(x="Age", y="Passengers")+
guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("black","#008000"))+ggtitle("Survived by Age")
plotbySex<-ggplot(data,aes(Sex,fill=Survived))+geom_bar() +labs(x="Sex", y="Passengers")+
guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("black","#008000"))+ggtitle("Survived by Sex")
plotbyTitle<-ggplot(data,aes(Title,fill=Survived))+geom_bar() +labs(x="Title", y="Passengers")+
guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("black","#008000"))+ggtitle("Survived by Title")
grid.arrange(plotbyClass,plotbyAge,plotbySex,plotbyTitle,ncol=2)
```









R

Tablas para la variables Class

```
tabla_SPC <- table(Pclass,Survived)
tabla_SPC
prop.table(tabla_SPC)
prop.table(tabla_SPC,1)
```

Consola R

```
> # Tablas para la variables Class
> tabla_SPC <- table(Pclass,Survived)
> tabla_SPC
      Survived
Pclass   0   1
     1  80 136
     2  97  87
     3 372 119
> prop.table(tabla_SPC)
      Survived
Pclass          0          1
     1 0.08978676 0.15263749
     2 0.10886644 0.09764310
     3 0.41750842 0.13355780
> prop.table(tabla_SPC,1)
      Survived
```

```
Pclass        0         1
    1 0.3703704 0.6296296
    2 0.5271739 0.4728261
    3 0.7576375 0.2423625
>
```

R

```
# Tablas para la variable Sex
tabla_SST <- table(Sex,Survived)
tabla_SST
prop.table(tabla_SST)
prop.table(tabla_SST, margin = 1)
```

Consola R

```
> # Tablas para la variable Sex
> tabla_SST
        Survived
Sex        0   1
  female  81 233
  male   468 109
> prop.table(tabla_SST)
        Survived
Sex              0         1
  female 0.09090909 0.26150393
  male   0.52525253 0.12233446
> prop.table(tabla_SST, margin = 1)
        Survived
Sex              0         1
  female 0.2579618 0.7420382
  male   0.8110919 0.1889081
```

R

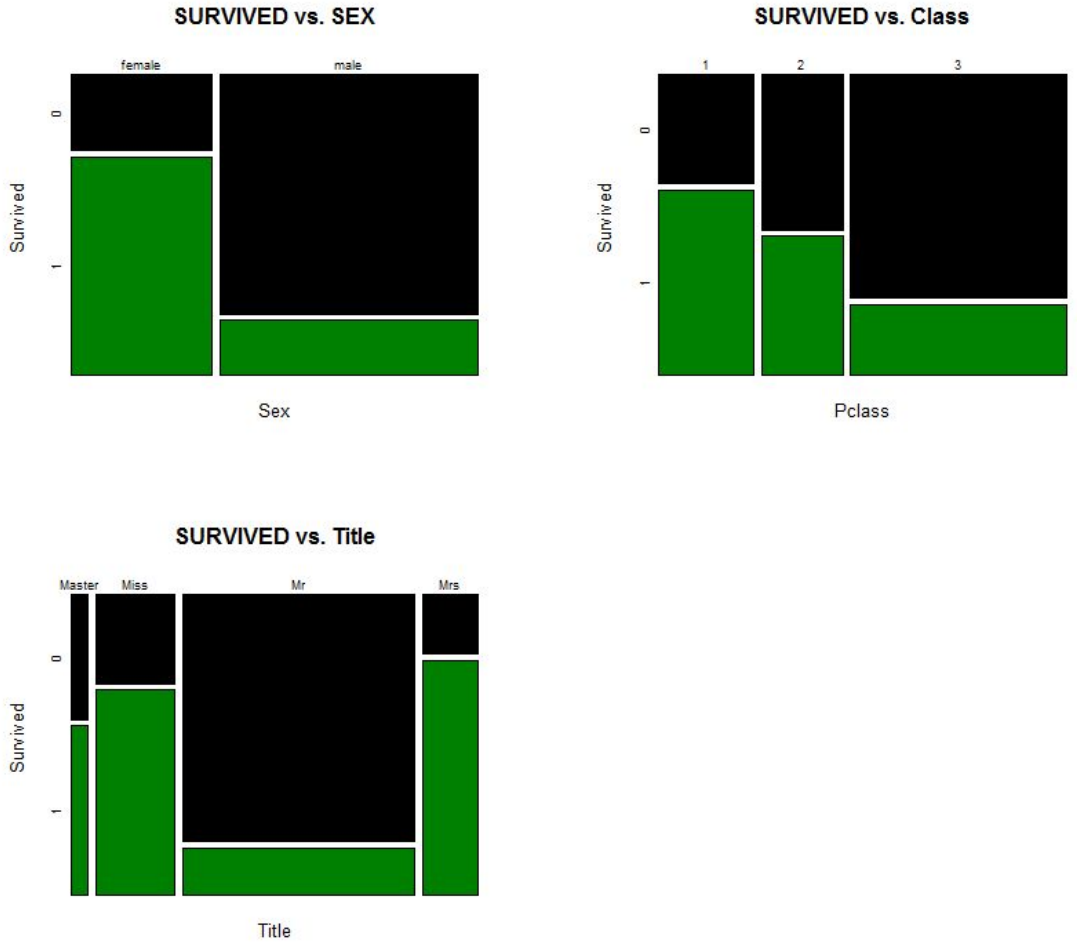
```
# Tablas para la variables Title
tabla_ST <- table(Title,Survived)
tabla_ST
prop.table(tabla_ST)
prop.table(tabla_ST,1)
```

Consola R

```
> # Tablas para la variables Class
> tabla_ST <- table(Title,Survived)
> tabla_ST
        Survived
Title      0   1
  Master  17  23
  Miss    55 128
  Mr     451  87
  Mrs     26 103
> prop.table(tabla_ST)
        Survived
Title            0         1
  Master 0.01910112 0.02584270
  Miss   0.06179775 0.14382022
  Mr     0.50674157 0.09775281
  Mrs    0.02921348 0.11573034
> prop.table(tabla_ST,1)
        Survived
Title          0         1
  Master 0.4250000 0.5750000
  Miss   0.3005464 0.6994536
  Mr     0.8382900 0.1617100
  Mrs    0.2015504 0.7984496
```

R
```
# mostramos graficas en porcentaje

par(mfrow=c(2,2))
plot(tabla_SST, col = c("black","#008000"), main = "SURVIVED vs. SEX")
plot(tabla_SPC, col = c("black","#008000"), main = "SURVIVED vs. Class")
plot(tabla_ST, col = c("black","#008000"), main = "SURVIVED vs. Title")
```



De las gráficas se puede obtener información muy valiosa que completamos con las tablas de contingencia.
En relación al Sexo, pese a que hay muchos más hombres (577) que mujeres (314), sobrevivieron más mujeres (233) que hombres (109), siendo el porcentaje de mujeres sobrevivientes del 74,2% frente al 18,9% de hombres que sobrevivieron

En relación a la clase a la que pertenecen los pasajeros, más de la mitad pertenecían a la 3ra clase (491), siendo el número de pasajeros de la 2da clase de 184 y el número de

pasajeros de 1ra fue de 216. Pese a que los pasajeros de tercera son muchos más, en número sobrevivieron casi los mismos en todas las clases, siendo el porcentaje de supervivientes en 1ra de 62%, frente al 47% de 2da y 24,2% de 3ra Clase. Por tanto el pertenecer a una clase u otra si influye a la hora de sobrevivir

En relación a la clase Title, el valor con más observaciones es para los señores (Mr) con 538 pasajeros, frente a las Señoras (Mrs) con 126, señoritas (Miss) con 183 ocurrencias y los niños con 40. No obstante el porcentaje de Mr que se salvan es del 16,1% frente al 57% de master, 69,9 Miss y 79,84% Mrs.

En cuanto a las edades, tal y como está en el dataset, sin agruparlas en varios rangos, es difícil hacer uso de este atributo, ya que hay pocas muestras para muchas de las edades. Para hacer un mejor uso se podrían crear dos categorías: niños, adultos, entendiendo por niños los menores de 12 años

Hemos añadido la gráfica del atributo Embarked y comprobamos que la probabilidad de sobrevivir no depende del puerto de Embarked, ya que la todas son parecidas.

Para obtener la relación de supervivencia, combinando las variables más influyentes, eliminamos las columnas que no nos interesan y pintamos la relación de los 4 atributos restantes (Sex, PcClass, Survived,Title)

R

```
#eliminamos las columnas que no queremos usar para el análisis y obtenemos tablas cruzando las variables restantes
data$Age<-NULL
data$SibSp<-NULL
data$Parch<-NULL
data$Fare<-NULL
data$Cabin<-NULL
data$Embarked<-NULL
data$Survived1<-NULL

#mostramos la relación de las variables que quedan, todas con todas
table(data)
```

Consola R

```
> table(data)

, , Sex = female, Title = Master
            Pclass
Survived  1  2  3
         0  0  0  0
         1  0  0  0

, , Sex = male, Title = Master
            Pclass
Survived  1  2  3
         0  0  0 17
         1  3  9 11

, , Sex = female, Title = Miss
            Pclass
Survived  1  2  3
         0  2  2 51
         1 44 33 51
```

```
, , Sex = male, Title = Miss
              Pclass
Survived  1  2  3
         0  0  0  0
         1  0  0  0

, , Sex = female, Title = Mr
            Pclass
Survived  1  2  3
        0  0  0  0
        1  1  0  0

, , Sex = male, Title = Mr
            Pclass
Survived  1  2  3
     0  77  91 283
     1  42  8  36

, , Sex = female, Title = Mrs
             Pclass
 Survived  1  2  3
         0  1  4  21
         1  45  37  21

, , Sex = male, Title = Mrs
            Pclass
Survived  1  2  3
        0  0  0  0
        1  0  0  0
```

De estas nuevas tablas combinando las variables que queremos analizar, obtenemos la siguiente información:

- Los 17 niños (Master) que murieron, eran de 3ra clase, en cambio sobrevivieron 3 niños de 1ra, 9 de 2da y 11 de 3ra
- De las señoritas que murieron, casi todas eran de 3ra, en dicha clase solo se salvaron el 50% de las niñas, en cambio en 1ra y 2da se salvaron casi todas con sólo 2 muertes para cada una de dichas clases.
- De los señores que murieron, se salvaron en mayor proporción los de 1ra llegando a salvarse 42 pasajeros de dicha clases,
- En cuanto a las señoras que se salvaron, otra vez se salvaron en mayor proporción las de 1ra y 2da clase. Siendo las de 3ra clase las que se murieron en mayor proporción (50%)

Creación Árbol de decisión

Tal y con los datos que nos proporciona el data set, y teniendo en cuenta que disponemos del atributo “Survived”, el primera análisis que nos viene a la cabeza es crear un árbol de decisión para analizar qué tipo de pasajero del Titanic tiene más probabilidades de sobrevivir.

Aunque tengamos 2 conjuntos de datos, “Titanic_test” y “Titanic_train”, no podemos usar ambos conjuntos ya que el dataset Titanic_test no dispone del atributo survived y por tanto no podemos validar cómo de bien clasifica nuestro algoritmo.

Para la realización del ejercicio contamos con el dataset titanica_train, que dividiremos en dos conjuntos, el primer conjunto lo usaremos para entrenar el sistema, en cambio el segundo conjunto lo usaremos para testear el algoritmo.

Almacenamos en la variable “y” los valores del atributo superviviente y en la variable “x” los valores del resto de atributos. Después aplicamos el algoritmo C5.0[1] que es una mejora del famoso algoritmo de clasificación ID3.

```
R
######### CREACIÓN DEL ÁRBOL DE DECISIÓN
str(data)

set.seed(666)
#cogemos 2/3 de los datos para entrenamiento y 1/3 para validacion

y<-data[,1] #SURVIVED
X <- data[,2:4] #PCLASS, TITLE, SEX

indexes = sample(1:nrow(data), size=floor((2/3)*nrow(data)))
trainx<-X[indexes,]
trainy<-y[indexes]
testx<-X[-indexes,]
testy<-y[-indexes]

str(trainy)

#Creamos el arbol de decision con los datos entrenamiento
model <- C50::C5.0(trainx, trainy,rules=TRUE )
#model <- C50::C5.0(trainx, trainy, control = C5.0Control(noGlobalPruning = TRUE,minCases=1))
summary(model)
model <- C50::C5.0(trainX, trainy)
plot(model)
```

Consola R

```
> summary(model)
Call:
C5.0.default(x = trainx, y = trainy, rules = TRUE)

C5.0 [Release 2.07 GPL Edition]      Thu Jan 03 22:13:29 2019
-------------------------------
Class specified by attribute `outcome'
Read 594 cases (4 attributes) from undefined.data
Rules:
Rule 1: (349/62, lift 1.4)
        Title = Mr
        ->  class 0  [0.821]

Rule 2: (245/71, lift 1.8)
        Title in {Master, Miss, Mrs}
        ->  class 1  [0.709]
Default class: 0

Evaluation on training data (594 cases):

        Rules
    ----------------
      No      Errors

       2  133(22.4%)   <<
```

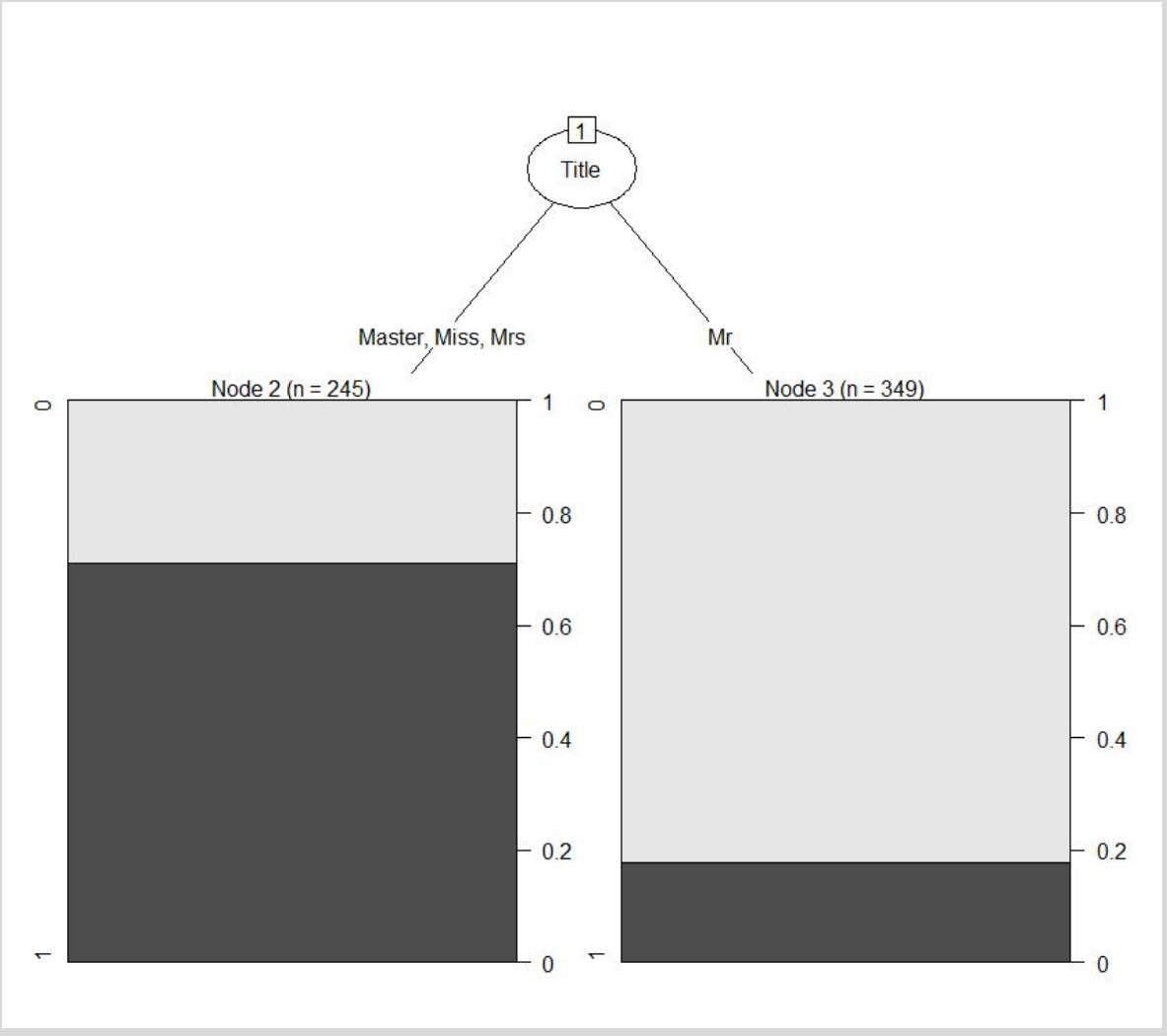
[1] https://es.wikipedia.org/wiki/C4.5

```
        (a)   (b)    <-classified as
       ----  ----
        287    71    (a): class 0
         62   174    (b): class 1

       Attribute usage:

       100.00%  Title

Time: 0.0 secs
```



Después de aplicar el algoritmo, obtenemos dos reglas que nos permiten clasificar a los pasajeros con un 22,4% de error. De 594 casos evaluados, ha clasificado incorrectamente a 133 pasajeros:

- Regla 1. Si el Title = Mr → la probabilidad de morir es del 82%
- Regla 2. Si el Title = {Master, Miss, Mrs} → la probabilidad de vivir es 70,1%

En este caso se usa el algoritmo C5.0 con poda que hace que se obtenga un árbol reducido y lo hace podando las ramas que aportan poca mejora a los resultados obtenidos.

Para validar el buen funcionamiento del modelo podemos usar los datos de validación que hemos reservado (⅓ del total) y obtenemos una precisión de 82,15% que es un poco mejor que la tasa de error obtenida con los datos de entrenamiento. Si usamos la matriz de

confusión podemos observar que se ha equivocado en 27 observaciones al determinar que una persona vivió y en realidad murió y 26 veces en el caso contrario, habiendo acertado en 240 observaciones.

R

```
#verificamos el modelo con los datos de verificación (test)
predicted_model <- predict( model, testx, type="class" )
print(sprintf("La precisión del árbol es : %.4f %%",100*sum(predicted_model == testy) / length(predicted_model)))

#obtenemos la matriz de confusión para obtener más detalle sobre los errores
if(!require(gmodels)){
  install.packages('gmodels', repos='http://cran.us.r-project.org')
  library(gmodels)
}

CrossTable(testy, predicted_model,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = c('Reality', 'Prediction'))
```

Consola R

```
[1] "La precisión del árbol és: 82.1549 %"

> CrossTable(testy, predicted_model,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = c('Reality', 'Prediction'))


   Cell Contents
|-------------------------|
|                       N |
|     N / Table Total |
|-------------------------|

Total Observations in Table:  297

             | Prediction
     Reality |         0 |         1 | Row Total |
-------------|-----------|-----------|-------------|
           0 |       164 |        27 |         191 |
             |     0.552 |     0.091 |             |
-------------|-----------|-----------|-------------|
           1 |        26 |        80 |         106 |
             |     0.088 |     0.269 |             |
-------------|-----------|-----------|-------------|
Col.  Total |       190 |       107 |         297 |
-------------|-----------|-----------|-------------|
```

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

Teniendo en cuenta los datos de los que disponíamos, el objetivos que nos planteamos era saber si se podía predecir si un pasajero iba a sobrevivir o no. Para ello primero hemos limpiado las variables, y hemos creado una variable “Title” que nos permite saber si un pasajero es “MR, Mrs, Miss y Master” y a la postre ha sido la más decisiva.
La mayoría de variables que tenían más relevancia eran cualitativas por tanto el uso de un algoritmo de clasificación era lo más apropiado.

Podemos concluir, que el conocimiento extraído usando el análisis visual y el algoritmo de clasificación es: se cumplió el tópico de que las mujeres y niños primero a la hora de subir a los botes salvavidas. Aunque no se refleje en el árbol, si eres de primera o segunda clase, tienes más probabilidades de vivir que si eres de 3ra.

7. Adjuntar el código de la práctica

El código y los datasets de la práctica se pueden encontrar en:
https://github.com/istorboi/AnalisisDatosTitanic