



Universitat  
Oberta  
de Catalunya

# PRAC 2

M2.851 - Tipología y Ciclo de vida de los datos · PRAC 2 · 2018-2019

Pablo Panero · Master en Ciencia de Datos

Proyecto de GitHub → [UOC Projects : Tipología](#)

# Índice

<b>Wiki</b>	<b>3</b>
<b>Preguntas de la práctica</b>	<b>4</b>
Descripción del dataset	4
Integración y selección de los datos de interés a analizar	5
Limpieza de los datos.	11
¿Los datos contienen ceros o elementos vacíos?	11
Identificación y tratamiento de valores extremos	16
Análisis de los datos	21
Selección de los grupos de datos que se quieren analizar/comparar	21
Comprobación de la normalidad y homogeneidad de la varianza	22
Aplicación de pruebas estadísticas para comparar los grupos de datos	24
Independencia y predictividad	24
Clasificación	26
Reglas de asociación	27
Representación de los resultados a partir de tablas y gráficas	29
Contraste de hipótesis	35
Resolución del problema	37
Código	38
<b>Referencias</b>	<b>38</b>

# Wiki

Proyecto de GitHub → [UOC Projects : Tipología](#)

Práctica realizada por Pablo Panero. La estructura de los ficheros a entregar es la siguiente:

- Las respuestas a las preguntas de la práctica se encuentran en los apartados siguientes en el archivo "*ppanero\_PRAC2.pdf*".
- El código se encuentra en el archivo "*ppanero\_prac2.r*".
- Los CSV con los datos originales se encuentran en el directorio "*original\_dataset*".
- Los CSV con los datasets generados durante la realización de la práctica se encuentran en "*processed\_dataset*". Estos son dos:
  - *filteredData*: Dataset generado tras la limpieza de los datos para la ejecución de las pruebas estadísticas.
  - *asociationData*: Dataset generado para aplicar un algoritmo de reglas de asociación, es decir, con todas las variables discretizadas.

# Preguntas de la práctica

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

El conjunto de datos, o *dataset*, pertenece a la competición "[Titanic: Machine Learning from Disaster](#)". El objetivo de este estudio es entender las características que permitieron a unos pasajeros sobrevivir y a otros no, teniendo en mente intentar predecir este hecho con la máxima precisión posible en el futuro (Participar en la competición). A pesar de ser el anterior el objetivo principal, indirectamente podría, por ejemplo, ayudar a entender qué características tienen los fallecidos y estudiar por qué estos no pudieron sobrevivir (Ej. la situación de los camarotes en el barco). Obteniendo, así ideas de como mejorar la seguridad o incrementar las posibilidades de supervivencia en el caso del hundimiento de un barco.

El dataset se compone de dos archivos *train.csv* y *test.csv*, ambos con la misma estructura. Sin embargo, se debe utilizar el primero para construir los modelos de clasificación, y el segundo para comprobar la eficacia de estos. Además, se provee un tercer fichero, *gender\_submission.csv*, a modo de ejemplo de cómo se deben entregar los archivos a Kaggle en caso de que quisiéramos participar en la competición.

Los atributos del dataset son:

- PassengerId: Identificador del pasajero en el Titanic.
- Survived: Indica si el pasajero sobrevive (1) o no (0).
- Pclass: Indica la clase de ticket que obtuvo el pasajero. Los valores posibles son alta (1), media (2), y baja (3). Se puede relacionar este atributo con la clase socioeconómica del pasajero.
- Name: Nombre del pasajero.
- Sex: Género del pasajero. Valores posibles hombre (male) o mujer (female).
- Age: Edad del pasajero en años. Si es menor que 1 año se especifica en forma fraccionaria.
- SibSp: Número de hermanos/as (incluye hermanastros/as) y/o cónyuges (novios/as y prometidos/as no se tienen en cuenta) a bordo del Titanic.
- Parch: Número de padres y/o hijos (incluye hijastros/as) a bordo del Titanic.
- Ticket: Número de ticket del pasajero.
- Fare: Tarifa del pasajero.
- Cabin: Camarote del pasajero.
- Embarked: Puerto en el cual embarcó el pasajero. Los valores posibles son Cherbourg (C), Queenstown (Q), y Southampton (S).

## Integración y selección de los datos de interés a analizar

En primer lugar leemos los datos, utilizando el parámetro “header” con valor “TRUE” ya que la primera fila de los archivos contiene la cabecera de los datos (nombre de los atributos o columnas).

```
> datasetDir = "~/dataset/"
> dataTrain <- read.csv(paste(datasetDir, "train.csv", sep="/"), header = TRUE)
> dataTest <- read.csv(paste(datasetDir, "test.csv", sep="/"), header = TRUE)
```

Acto seguido comprobamos las dimensiones, y vemos que el dataset de entrenamiento (*train.csv*) tiene una columna más que el de prueba (*test.csv*), un total de 12 con respecto a 11. Esto se debe a que en el segundo dataset el atributo “Survived” no está presente (ya que es el que se pretende predecir). Podemos ver que el dataset de entrenamiento tiene 891 observaciones y el de prueba 418.

```
> dim(dataTrain)
[1] 891 12
> dim(dataTest)
[1] 418 11
```

Además vemos que hay dos atributos más que no se encontraban en la descripción de Kaggle (Sí se incluyeron en la descripción anterior): “Name”, que contiene el nombre del pasajero, y “PassengerId”, que contiene el identificador del pasajero en el Titanic.

*NOTA: De ahora en adelante trataremos solamente el dataset de entrenamiento, ya que es en base al cual debemos obtener el modelo, pero las operaciones realizadas deberían ser aplicables al dataset de prueba.*

En el siguiente paso obtenemos la clase de las distintas variables, para ver de qué tipo son:

```
> vars <- sapply(dataTrain, class)
> table(data.frame(variables=names(vars), clase=as.vector(vars)))
```

Y se obtienen los siguientes valores:

Variable	Clase <sup>1</sup>
PassengerId	Integer
Survived	Integer
Pclass	Integer
Name	Factor

---

<sup>1</sup> Para el dataset *test.csv* se obtienen los mismo valores, salvo el atributo “Survived” que no está presente.

Sex	Factor
Age	Numeric
SibSp	Integer
Parch	Integer
Ticket	Factor
Fare	Numeric
Cabin	Factor
Embarked	Factor

En primer lugar vemos (utilizando la función *view*) que los atributos “PassengerId” y “Name” no aportan ninguna información útil para conseguir nuestro objetivo. El primero es un contador entero, y el segundo una cadena de caracteres con el nombre del pasajero. Por lo que los eliminamos.

```
> view(dataTrain)
> library(dplyr)
> filteredData <- select(dataTrain, -PassengerId)
> filteredData <- select(filteredData, -Name)
```

Exploramos nuevamente los datos y vemos que el atributo “Ticket” corresponde con una cadena de números y/o caracteres que no aportan tampoco ningún valor de cara a conseguir nuestro objetivo. Por lo tanto lo eliminamos también:

```
> filteredData <- select(filteredData, -Ticket)
```

Por último vemos que valores podrían necesitar ser convertidos (discretizados). Mostramos los atributos y su tipo:

```
> str(filteredData)
'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 : chr "male" "female" "female" "female" ...
 $ 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 ...
 $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin : chr "" "C85" "" "C123" ...
 $ Embarked: chr "S" "C" "S" "S" ...
```

De la descripción del apartado anterior sospechamos que “Survived”, “Pclass”, “Sex”, y “Embarked” toman valores dentro de un conjunto de etiquetas/valores determinado. Lo comprobamos:

```
> colsToFactor<-c("Survived","Pclass","Sex","Embarked")
> for (i in colsToFactor){
  print(i)
  print(summary(as.factor(filteredData[,i])))
}
```

[1] "Survived"

0	1
549	342

[1] "Pclass"

1	2	3
216	184	491

[1] "Sex"

Female	Male
314	577

[1] "Embarked"

NA	C	Q	S
2	168	77	644

Por lo tanto, estos atributos deberían ser discretizados:

```
> for (i in colsToFactor){
  filteredData[,i] <- as.factor(filteredData[,i])
}
> str(filteredData)
'data.frame': 891 obs. of 9 variables:
 $ 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 ...
 $ 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 1 2 0 ...
$ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
$ Cabin : chr "" "C85" "" "C123" ...
$ Embarked: Factor w/ 4 levels "", "C", "Q", "S": 4 2 4 4 4 3 4 4 4 2 ...
```

Una vez discretizados, es interesante realizar un análisis descriptivo de los atributos discretizados para tener una noción general de los datos que tenemos. En las siguientes tablas se muestran los porcentajes de cada una de las clases de los atributos.

Se puede apreciar que el dataset está formado en gran parte, casi el doble, por hombres, con un 64.76% frente a un 35.24% de mujeres. Se aprecia también que la gran mayoría de los pasajeros se subieron al Titanic en la primera parada, Southampton, con un 72.28% de los pasajeros. Además, vemos que el número de pasajeros en primera y segunda clase suponen casi la mitad de los pasajeros con un total de 44.89% (24.24% en primera y 20.65% en segunda clase), siendo la otra mitad (55.11%) procedente de tercera clase. Finalmente, vemos que el porcentaje de supervivencia es del 38.38%.

```
> for (i in colsToFactor){
  print(i)
  print(table(filteredData[,i])*100/NROW(filteredData[,i]))
}
```

[1] "Sex"

Female	Male
35.2413	64.7587

[1] "Embarked"

NA	C	Q	S
0.2244669	18.8552189	8.6419753	72.2783389

[1] "Pclass"

1	2	3
24.24242	20.65095	55.10662

[1] "Survived"

0	1
61.61616	38.38384



Los atributos “SibSp” y “Parch” también son candidatos a ser discretizados ya que tienen pocas clases.

```
> summary(as.factor(filteredData$SibSp))
```

0	1	2	3	4	5	8
608	209	28	16	18	5	7

```
> summary(as.factor(filteredData$Parch))
```

0	1	2	3	4	5	6
678	118	80	5	4	5	1

Al igual que antes, realizamos un análisis descriptivo:

```
> colsToFactor<-c("SibSp","Parch")
> for (i in colsToFactor){
  print(i)
  print(table(filteredData[,i])*100/NROW(filteredData[,i]))
}
```

[1] "SibSp"

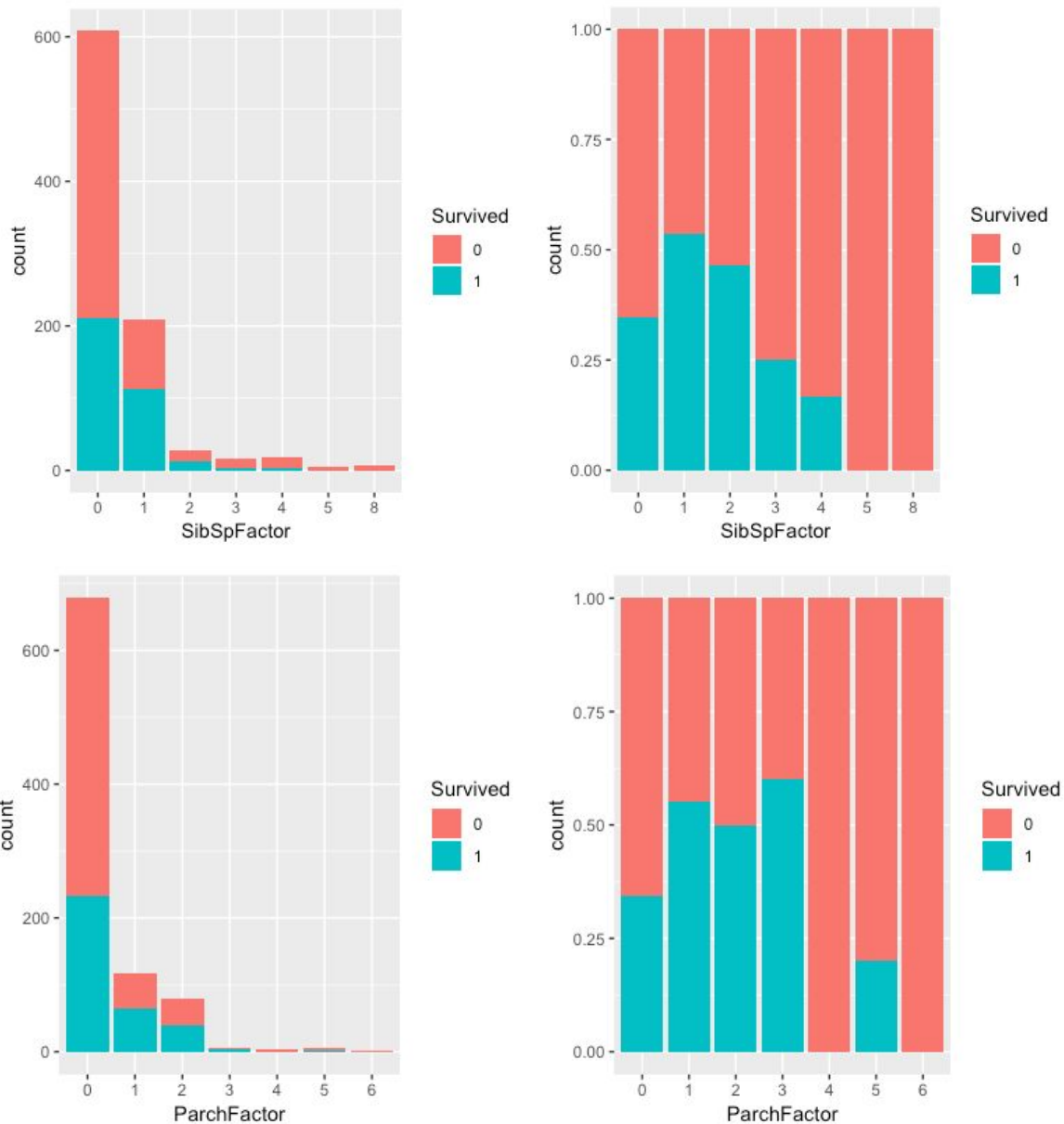
0	1	2	3	4	5	8
68.237935	23.45679	3.1425365	1.7957351	2.0202020	0.5611672	0.7856341

[1] "Parch"

0	1	2	3	4	5	6
76.09427 6	13.24354 7	8.978675 6	0.561167 2	0.448933 8	0.561167 2	0.112233 4

Del análisis anterior podemos apreciar que para los casos con valores mayores o iguales que 2 los porcentajes de aparición son significativamente bajos, por lo que estos dejan de ser informativos. En la siguiente fase agruparemos ambas variables en tres grupos según su valor: 0, 1 y  $\geq 2$ .

Vemos que ambas variables siguen una distribución muy similar. También con respecto a la clase a predecir “Survived” (lo corroboramos con las siguientes gráficas):



Puede ser interesante crear una nueva variable que represente el tamaño de la familia (“SibSp” + “Parch” + 1, el individuo de la observación). Tanto para discretizar las variables, para generar la agrupación mencionada anteriormente y para crear la nueva vamos a esperar a terminar la fase siguiente (búsqueda de valores nulos) para evitar contaminarla con valores erróneos o incompletos.

Para concluir esta fase, realizamos el análisis descriptivo de las variables restantes: “Fare” y “Age”. Ignoramos “Cabin” en este paso, ya que se trata de una cadena de caracteres.

```
> colsToDescribe<-c("Age","Fare")
```

```
> for (i in colsToDescribe){
  print(i)
  print(summary(filteredData[,i]))
  print('SD')
  print(sd(filteredData[,i], na.rm=TRUE))
}
```

[1] "Age"

Min	1st Qu.	Median	Mean	3rd Qu.	Max	SD	NA's
0.42	20.12	28.00	29.70	38.00	80.00	14.5265	177

[1] "Fare"

Min	1st Qu.	Median	Mean	3rd Qu.	Max	SD	NA's
0.00	7.91	14.45	32.20	31.00	512.33	49.69343	0

No vemos nada que destaque en la variable "Age", con un mínimo de 0.42 (bebé) y máximo de 80 (anciano) y la mayoría de los casos entre 20 y 40 años (deducción a partir de los cuartiles). Sin embargo, en "Fare" vemos que el máximo se separa en más de tres desviaciones estándar de la media, por lo que es probable que tenga que ser tratado en la fase siguiente.

Terminamos la selección de los datos relevantes para el análisis con 9 variables, 4 categóricas, 2 decimales, 2 enteras y 1 cadena de caracteres. Pasamos a comprobar la existencia de valores nulos, vacíos, extremos, etc. y seguir limpiando los datos en función de ello.

## Limpieza de los datos.

¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Dados los tipos de datos que tenemos deberíamos buscar por *NA (Not Assigned)* en todos ellos. Por la cadena vacía "" o " " en las cadenas de caracteres, y por valores menores que 0 en los numéricos.

```
> colSums(is.na(filteredData))
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	0	177	0	0	0	0	0

> colSums(filteredData=="", na.rm=T)

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	0	0	0	0	0	687	2

> colSums(filteredData==" ", na.rm=T)

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	0	0	0	0	0	0	0

> colSums(filteredData<0, na.rm=T)

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	0	0	0	0	0	0	0

Antes de realizar ninguna transformación para tratar los atributos nulos o vacíos comprobamos las variables categóricas, es decir, que no existan casos como por ejemplo la existencia de dos etiquetas para el mismo valor pero escritas diferente (Ej. "Male" y "male"):

```
> levels(filteredData$Survived)
[1] "0" "1"
> levels(filteredData$Pclass)
[1] "1" "2" "3"
> levels(filteredData$Sex)
[1] "female" "male"
> levels(filteredData$Embarked)
[1] "" "C" "Q" "S"
```

Vemos que los posibles valores de los atributos categóricos son correctos, excepto la etiqueta vacía ("") en "Embarked". Sin embargo, podemos ver que esta corresponde a las dos observaciones encontradas en el segundo filtro de las operaciones anteriores:

```
> summary(filteredData$Embarked)
```

Survived	C	Q	S
2	168	77	644

Explorados los datos en este aspecto, pasamos a tratar los valores nulos o vacíos encontrados.

En primer lugar vemos que el atributo “Cabin” tiene 687 valores vacíos. Teniendo en cuenta que el dataset tiene 891 observaciones, estos suponen un 77.1%. Llegamos a la conclusión de que no disponemos de datos suficientes para inferir los datos faltantes sin asumir un porcentaje de error muy alto (e incluso el sesgo del resultado). Por lo tanto, borramos dicha columna de los datos a analizar.

Sin embargo, hubiera sido interesante analizar la relación entre la letra/posición del camarote en el barco y la supervivencia (Aunque tampoco sabemos si los pasajeros estaban en el camarote en el momento de la evacuación).

```
> filteredData <- select(filteredData, -Cabin)
```

En cuanto a los atributos “Age” y “Embarked” podríamos asignar el valor medio y la moda respectivamente. A pesar de ello, utilizaremos el algoritmo de imputación de valores mediante *K-Nearest Neighbours* [5] ya que será más preciso (sobre todo para el atributo “Age”). Para aplicar este algoritmo al atributo “Embarked” necesitamos convertir los vacíos en NAs y normalizar los atributos “Age” y “Fare” para que no sesguen el algoritmo.

Para conservar los valores de las variables numéricas originales (para los análisis posteriores), realizamos la normalización y la imputación *kNN* en una matriz auxiliar y después copiamos los valores asignados a la matrix *filteredData*.

Apuntamos algunos identificadores que fueran nulos para los atributos. Para “Embarked” 62 y 830, para “Age” 18, 30 y 236. Y aplicamos el algoritmo *kNN* [6]. Este tiene en cuenta todos los datos del primer parámetro, en nuestro caso el dataset completo, para asignar valores a las apariciones de NA en los atributos del parámetro ‘variable’. Este algoritmo calcula la distancia de Gower, una medida de distancia utilizada cuando los atributos son tanto numéricos como categóricos.

```
# Embarked
> filteredData$Embarked[filteredData$Embarked==""]=NA
> normalizedData <- filteredData
> normalizedData$Age <- scale(normalizedData$Age, center=FALSE)
> normalizedData$Fare <- scale(normalizedData$Fare, center=FALSE)
> normalizedData <- kNN(normalizedData, variable = c("Embarked"))
> filteredData$Embarked <- normalizedData$Embarked
# Age
> normalizedData <- filteredData
```

```

> normalizedData$Fare <- scale(normalizedData$Fare, center=FALSE)
> normalizedData <- kNN(normalizedData, variable = c("Age"))
> filteredData$Age <- normalizedData$Age

```

Y Comprobamos los valores:

```

> filteredData$Embarked[62]
[1] S
Levels: C Q S
> filteredData$Embarked[830]
[1] S
Levels: C Q S
> filteredData$Age[18]
[1] 34
> filteredData$Age[30]
[1] 33
> filteredData$Age[236]
[1] 24

```

Podemos observar que en el caso de “Embarked” el valor aplicado coincide con la moda. Sin embargo, en el caso de “Age” obtenemos mejores valores que la *media* que podríamos haber aplicado, ya que el algoritmo tiene en cuenta a los vecinos. Es decir, el resto de atributos, no solo los valores del atributo en cuestión, lo cual sería agnóstico a la observación.

Como medida de seguridad, comprobamos que no existen más valores nulos o vacíos:

```
> colSums(is.na(filteredData))
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	0	0	0	0	0	0

```
> colSums(filteredData=="", na.rm=T)
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	0	0	0	0	0	0

```
> colSums(filteredData==" ", na.rm=T)
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	0	0	0	0	0	0

```
> colSums(filteredData<0, na.rm=T)
```

Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	0	0	0	0	0	0

Antes de pasar a la fase siguiente procedemos a tratar las variables “SibSp” y “Parch” y a crear la nueva “FSize” (*Family Size*, o tamaño de familia). En primer lugar creamos esta última y comprobamos su distribución.

```
> filteredData$FSize <- filteredData$SibSp + filteredData$Parch + 1
```

Vemos que la nueva variable se encuentra en un caso similar que “SibSp” y “Parch” y sería útil categorizarla en tres grupos:

```
> summary(as.factor(filteredData$FSize))
```

1	2	3	4	5	6	7	8	11
537	161	102	29	15	22	12	6	7

Asignamos grupo 2 para “SibSp” y “Parch” y 3 para “FSize” a todos aquellos valores  $\geq 2$  y  $\geq 3$  respectivamente; y comprobamos los porcentajes de cada uno de los grupos:

```
> filteredData$SibSp[filteredData$SibSp>2]=2
> summary(as.factor(filteredData$SibSp))
```

0	1	2
608	209	74
68.237935	23.456790	8.305275

```
> filteredData$Parch[filteredData$Parch>2]=2
> summary(as.factor(filteredData$Parch))
```

0	1	2
678	118	95
76.09428	13.24355	10.66218

```
> filteredData$FSize[filteredData$FSize>3]=3
> summary(as.factor(filteredData$FSize))
```

1	2	3
537	161	193
60.26936	18.06958	21.66105

Podemos ver (tercera fila de cada tabla) que la mayor parte de los pasajeros (aprox. 60.27%) fueron solos, y la cantidad de pasajeros con un familiar, o dos o más es similar (18.07% y 21.66% respectivamente).

Por último discretizar las tres variables:

```
> filteredData$SibSp <- as.factor(filteredData$SibSp)
> filteredData$Parch <- as.factor(filteredData$Parch)
> filteredData$FSize <- as.factor(filteredData$FSize)
```

Terminamos esta etapa con 11 variables, 9 categóricas y 2 numéricas.

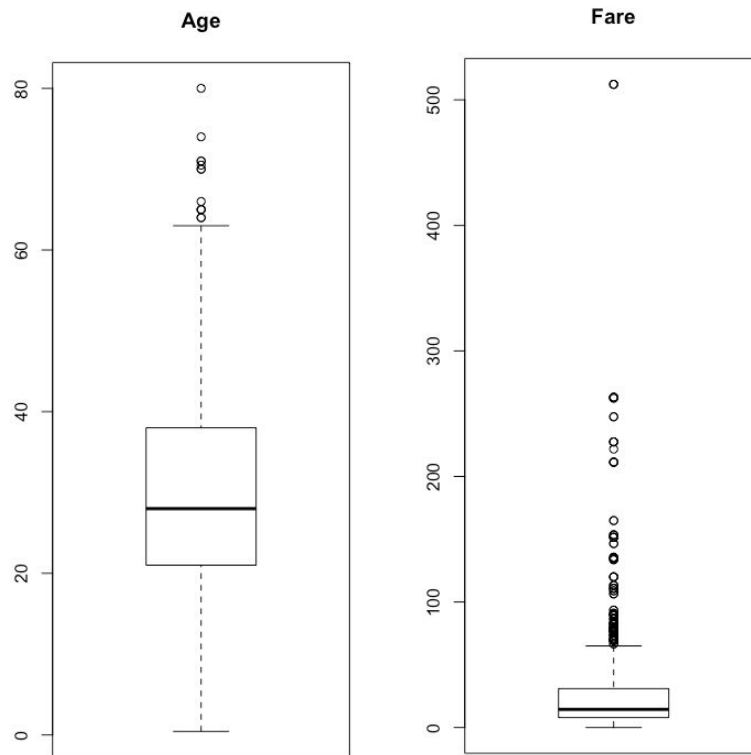
### Identificación y tratamiento de valores extremos

En las variables categóricas no tiene sentido realizar un análisis de valores extremos ya que solo toman valores dentro de un conjunto concreto. Una posible mejora podría ser realizar un análisis multivariable [7].

Realizamos el análisis con el paquete *boxplot* de R [8] sobre "Age" y "Fare". En primer lugar obtenemos los diagramas de cajas:

```
> boxplot(filteredData$Age)
> boxplot(filteredData$Fare)
```





Comprobamos los valores:

```
> boxplot(filteredData$Age)$out
[1] 66.0 65.0 71.0 70.5 65.0 64.0 65.0 71.0 64.0 80.0 70.0 70.0 74.0
> boxplot(filteredData$Fare)$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
[13] 77.2875 79.2000 66.6000 69.5500 69.5500 146.5208 69.5500 113.2750 76.2917
90.0000 83.4750 90.0000
[25] 79.2000 86.5000 512.3292 79.6500 153.4625 135.6333 77.9583 78.8500
91.0792 151.5500 247.5208 151.5500
[37] 110.8833 108.9000 83.1583 262.3750 164.8667 134.5000 69.5500 135.6333
153.4625 133.6500 66.6000 134.5000
[49] 263.0000 75.2500 69.3000 135.6333 82.1708 211.5000 227.5250 73.5000
120.0000 113.2750 90.0000 120.0000
[61] 263.0000 81.8583 89.1042 91.0792 90.0000 78.2667 151.5500 86.5000
108.9000 93.5000 221.7792 106.4250
[73] 71.0000 106.4250 110.8833 227.5250 79.6500 110.8833 79.6500 79.2000
78.2667 153.4625 77.9583 69.3000
[85] 76.7292 73.5000 113.2750 133.6500 73.5000 512.3292 76.7292 211.3375
110.8833 227.5250 151.5500 227.5250
[97] 211.3375 512.3292 78.8500 262.3750 71.0000 86.5000 120.0000 77.9583
211.3375 79.2000 69.5500 120.0000
[109] 93.5000 80.0000 83.1583 69.5500 89.1042 164.8667 69.5500 83.1583
```

Observamos que para “Age” los valores no son extremos. Incluso el valor máximo es un valor posible y razonable para la edad de una persona:

```
> max(boxplot(filteredData$Age)$out)
[1] 80
```

*Nota: no se calculan los mínimos porque ningún valor es menor que 0 (o 64 en "Age"), siendo 0 (o 64) un valor aceptado.*

En el caso de "Fare" debemos investigar un poco más el porqué de los precios (tarifas) tan altas para los billetes.

Obtenemos el mínimo y la media de los outliers y lo comparamos con la clase (socioeconómica) a la que pertenecen:

```
> min(boxplot(filteredData$Fare)$out)
[1] 66.6
> pclass_min <- filteredData$Pclass[filteredData$Fare >= 66.6]
> summary(pclass_min)
```

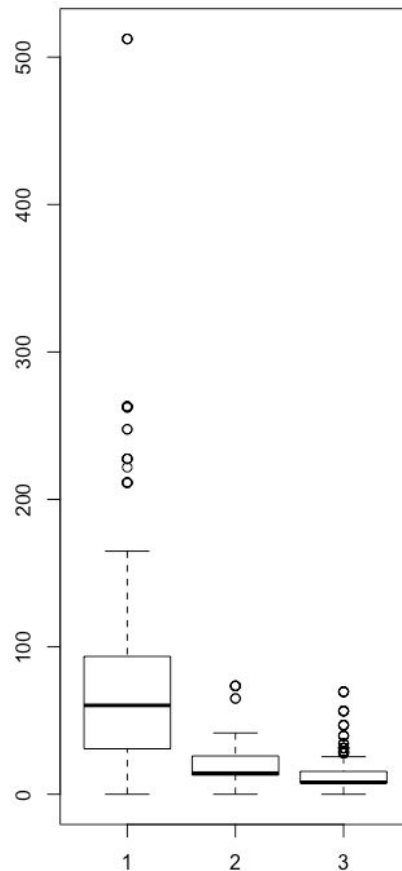
1st	2nd	3rd
104	5	7

```
> mean(boxplot(filteredData$Fare)$out)
[1] 128.2916
> pclass_mean <- filteredData$Pclass[filteredData$Fare >= 128.2916]
> summary(pclass_mean)
```

1st	2nd	3rd
38	0	0

Todo lleva a indicar que los billetes "caros" que se identificaron como *outliers* son de primera clase, por lo que es normal que sean más (incluso excesivamente) caros. Para comprobar esta teoría, realizamos el diagrama de cajas separado por clases ("Pclass").

Cada una de las cajas representa una clase (1 para primera, 2 para segunda y 3 para tercera), siendo el eje Y el precio ("Fare").



Si bien es cierto que algunos billetes de segunda y tercera clase son más caros de lo normal, no son lo suficientemente caros como para hacernos sospechar que son datos erróneos (Podría tratarse por ejemplo de billetes de última hora, que son más caros de lo normal). Esto mismo podría ocurrir con los precios de los billetes de primera clase que cuestan más de 200, sin embargo, hay tres apariciones por encima de 500.

```
> highPrice <- filteredData$Fare[filteredData$Fare > 500]
> highPrice
[1] 512.3292 512.3292 512.3292
```

También llama la atención la existencia de tickets con precio 0 (15 casos):

```
> lowPrice <- filteredData$Fare[filteredData$Fare == 0]
> lowPrice
[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Aunque la aparición de más de una observación con el mismo precio pudiera suponer un dato legítimo (Ej. camarote presidencial, o ganados en un sorteo para los de coste 0), estos cuestan casi el doble que el siguiente ticket más caro (511 vs 263), sin hablar de su distancia en términos de desviaciones estándar (casi 5 veces).

```
> sd(filteredData$Fare)
[1] 49.69343
```

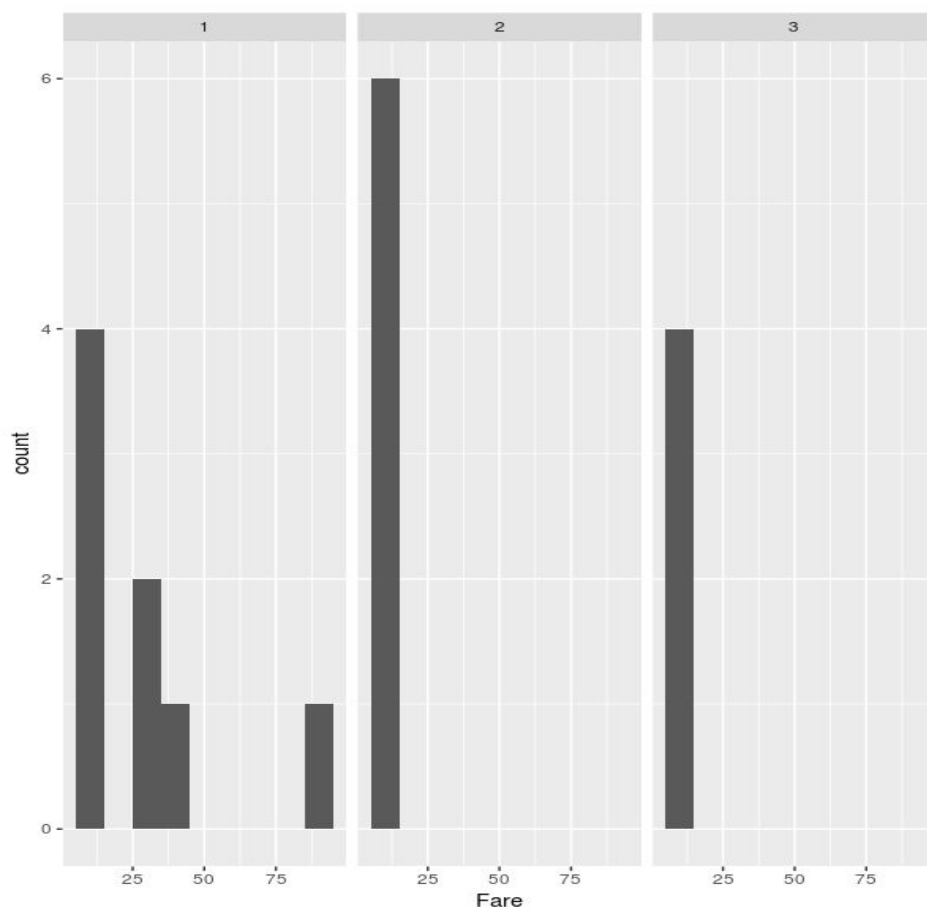
También es poco creíble que los billetes tengan coste 0, porque aunque se hubieran ganado en un sorteo, alguien los tendría que comprar para sortearlos. Por lo tanto también los trataremos como valores extremos.

Para tratarlos, le asignaremos el valor *NA* y utilizaremos el método de imputación *kNN*, al igual que realizamos anteriormente con un dataset auxiliar con los valores normalizados para el resto de atributos.

```
> filteredData$Fare[filteredData$Fare == 0 | filteredData$Fare > 500]=NA
> normalizedData <- filteredData
> normalizedData$Age <- scale(normalizedData$Age, center=FALSE)
> normalizedData <- kNN(normalizedData, variable = c("Fare"))
> filteredData$Fare <- normalizedData$Fare
```

Comprobamos los valores que asigna:

```
> ggplot(data=normalizedData[normalizedData$Fare_imp==TRUE,],
aes(x=Fare))+geom_histogram(binwidth = 10)+facet_wrap(~Pclass)
```



Estos valores tienen sentido ya que asigna valores más altos a los tickets de 1ra clase, y menor a los de 2da y 3ra.

Damos por terminada la fase de tratamiento de valores extremos.

## Análisis de los datos

Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar)

Los datos que se desean analizar han sido seleccionados en el apartado anterior, y son: "Sex", "Age", "Pclass", "Fare", "Sibsp", "Parch", "FSize" y "Embarked". El resto de datos han sido descartados por no ser relevantes para nuestra tarea o bien por no tener la suficiente calidad (Ej. "Cabin").

En cuanto a los análisis que se quieren realizar, sería interesante obtener la relación entre la edad y el género con respecto a la supervivencia. Lo mismo con la clase socio-económica ("Pclass" y "Fare"), y por último de con respecto al tamaño de la familia ("Sibsp", "Parch" y "FSize"). Respondiendo a preguntas como:

- ¿Tienen más posibilidades de salvarse los hombres o las mujeres? ¿De qué edad?
- ¿Tienen más posibilidades de salvarse los pasajeros de clase socioeconómica elevada?
- ¿Tienen más posibilidades de salvarse las familias? ¿De qué tamaño?
- ¿Tienen más posibilidades de salvarse los hombres o las mujeres de una familia? ¿De qué tamaño? ¿De qué edad? ¿De qué clase social?
- ¿Existe alguna relación entre el puerto de embarque y la supervivencia?

Además como dato curioso, se podría estudiar a las relaciones entre las clase socio-económica y el tamaño de la familia, y la edad/género.

Finalmente otro punto interesante sería estudiar la supervivencia de los pasajeros frente a la tripulación. Sin embargo, ha sido imposible identificar qué usuarios son tripulación, se ha llegado a la conclusión de que esta no está incluida en el conjunto de datos proporcionado. El análisis se han realizado mediante las funciones *head*, *tail*, *view* y *summary* en RStudio. Las conclusiones son:

- El porcentaje de datos faltantes en el atributo Cabin es demasiado alto (77.1%) como para reflejar la tripulación.
- No se encontró ninguna relación entre el tipo de ticket y la cabina. Por ejemplo, algún tipo de código, letra o rango en los tickets para los usuarios que no tienen asignado un camarote.
- No todos los pasajeros que no tienen camarote se subieron al Titanic en el mismo (primer) puerto. Lo cual debería ser cierto para la tripulación.
- No se encontró ninguna relación entre el precio del billete y la falta de cabina o tipo de ticket. Por ejemplo que aquellos sin camarote tengan precio 0 o algún código en el ticket por ser parte de la tripulación.

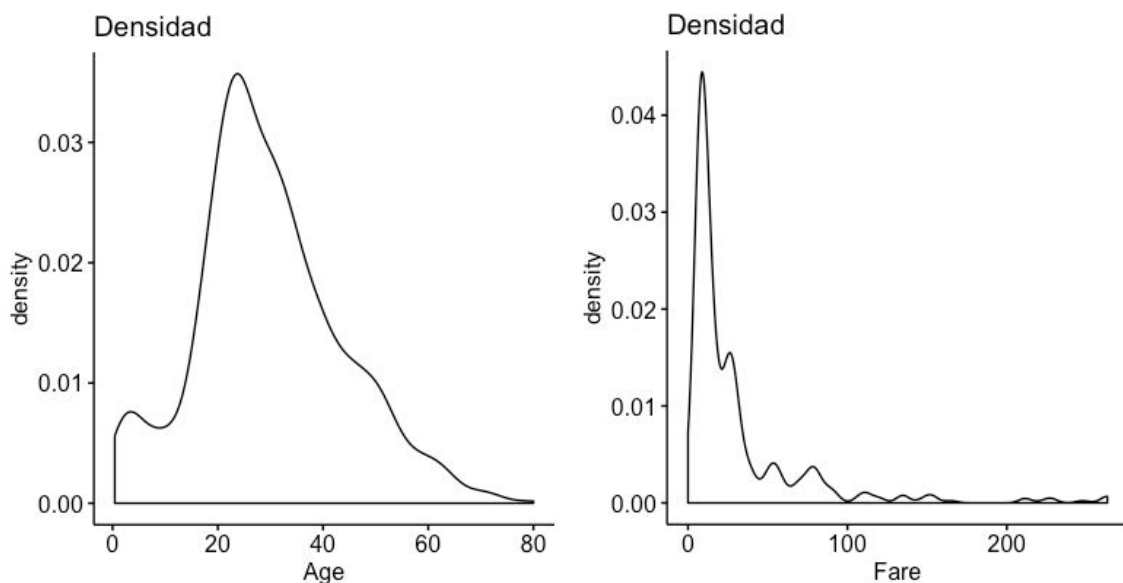
## Comprobación de la normalidad y homogeneidad de la varianza.

En el dataset generado en el apartado anterior tenemos 11 variables, 9 categóricas y 2 numéricas. Comprobar la normalidad solamente tiene sentido para las numéricas, por lo tanto estudiaremos la normalidad en “Age”, y “Fare”.

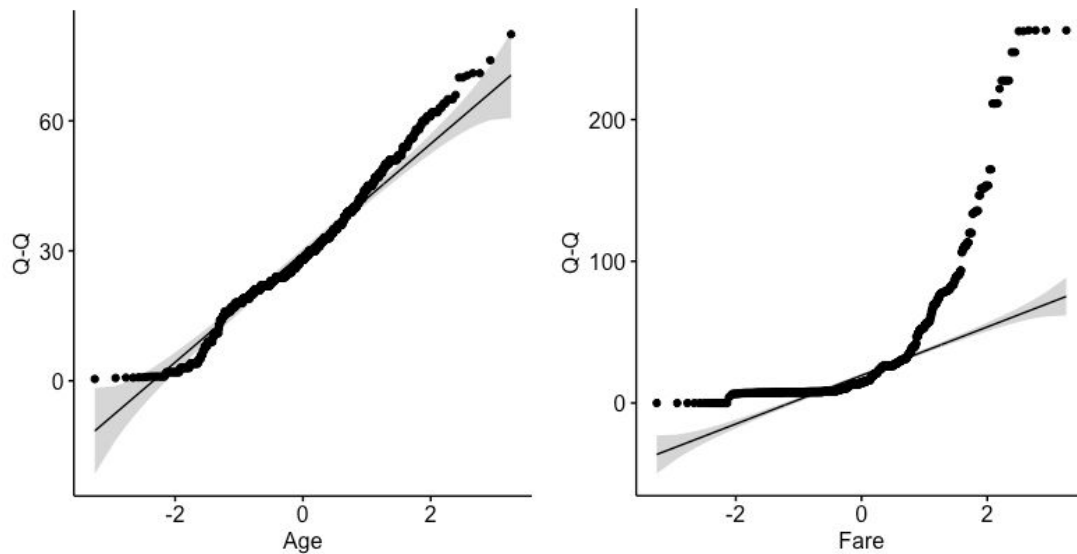
En primer lugar realizaremos un estudio visual de la normalidad mediante los gráficos de densidad (similar a un histograma) y de quantile-quantile [9].

```
> library("ggpubr")
> ggdensity(filteredData$Age, main="Densidad", xlab = "Age")
> ggdensity(filteredData$Fare, main="Densidad", xlab = "Fare")
> ggqqplot(filteredData$Age, ylab="Q-Q", xlab="Age")
> ggqqplot(filteredData$Fare, ylab="Q-Q", xlab="Fare")
```

Observamos las gráficas de densidad y vemos, a priori, que “Age” se aproxima bastante a una distribución normal (forma de campana) mientras que “Fare” parece no seguir la misma distribución.



Comprobamos con las gráficas de Q-Q (quantile-quantile). Podemos ver de nuevo que “Age” se aproxima a una distribución normal (línea recta en 45 grados [10]), mientras que “Fare” no.



Finalmente comprobamos mediante el test de Shapiro-Wilk [11] ya que la interpretación gráfica puede dar lugar a errores [12].

```
> shapiro.test(filteredData$Age)
  Shapiro-Wilk normality test
data: filteredData$Age
W = 0.98059, p-value = 1.665e-09
> shapiro.test(filteredData$Fare)
  Shapiro-Wilk normality test
data: filteredData$Fare
W = 0.59839, p-value < 2.2e-16
```

En ambos casos el p-value es menor que 0.05 por lo que se rechaza la hipótesis nula. Es decir, las variables no siguen una distribución normal. Por lo tanto debemos utilizar métodos de análisis que no supongan que las variables siguen una distribución normal.

No obstante, tal y como afirma Mordkoff, J. [13] cuando el número de observaciones (N) es mayor o igual a 30, debido al teorema central del límite, se pueden utilizar pruebas paramétricas ya que a medida que se incluyen más datos, su distribución se va volviendo más normal. En otras palabras, las medias de muestras (con  $N \geq 30$ ) aleatorias e independientes del conjunto de observaciones seguirán una distribución en forma de campana (normal). Esto nos permite utilizar técnicas paramétricas, aunque es posible que se obtengan mejores resultados con pruebas no paramétricas [14].

Pasamos a comprobar la homogeneidad de las varianzas (HOV) [15] [16]. Utilizaremos el test de Flinge-Killneen ya que es el más apropiado para variables no paramétricas que no siguen una distribución normal [17] [18].

```
> fligner.test(Age ~ Fare, filteredData)
```

```
      Fligner-Killeen test of homogeneity of variances  
data:  Age by Pclass  
fligner-Killeen:med chi-squared = 334.24, df = 245, p-value = 0.0001288
```

Obtenemos como resultado que las variables numéricas no son homogéneas en varianza ( $p\text{-value} < 0.05$ ), tenemos que tener esto en cuenta cuando aplicamos los métodos analíticos en caso de que asuman homogeneidad de las varianzas (ej. ANOVA).

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.

### Independencia y predictividad

En primer lugar vamos a comprobar qué variables (predictores) son más influyentes en la variable que nos interesa "Survived".

Para las variables categóricas utilizaremos el test de Chi-Square [19] [20] que nos permite comprobar la influencia (si son dependientes o no) de la variable predictora en la respuesta ("Survived") suponiendo que ambas variables son categóricas [21].

Para las variables numéricas utilizaremos el test ANOVA [22].

### Variables categóricas

```
> corr_matrix <- matrix(nc = 2, nr = 0)  
> colnames(corr_matrix) <- c("variable", "p-value")  
> categoricas <- select(filteredData, -Survived, -Age, -Fare)  
> for(i in 1:(ncol(categoricas))) {  
  pair = matrix(ncol = 2, nrow = 1)  
  pair[1][1] = colnames(categoricas)[i]  
  pair[2][1] = chisq.test(filteredData$Survived, categoricas[,i])$p.value  
  corr_matrix <- rbind(corr_matrix, pair)  
}  
> corr_matrix
```

Variable	X-Square	Df	P-Value
Sex	263.050	1	3.71174777011346e-59
PClass	102.888	2	4.54925171129874e-23
FSize	39.625	2	2.48572281574168e-09
SibSp	28.264	2	7.28560209980573e-07
Parch	21.071	2	2.6567727450344e-05



Embarked	25.964	2	2.30086264814496e-06
----------	--------	---	----------------------

Podemos ver que el test de Chi-Square muestra valores muy pequeños para p-value, menores que 0.05, por lo que rechazamos la hipótesis nula y concluimos que las variables son dependientes.

De la tabla de distribución obtenemos que para alpha 0.05 los valores son, según el grado de libertad (1, 2 o 4 en nuestro caso):

Df	1	2
	3.841	5.991

Comparando esta última tabla con los valores obtenidos en el paso anterior podemos ver que todos los indicadores de  $\chi^2$  son bastante elevados. Por lo tanto podemos ver en la tabla resultante de aplicar el test Chi-Square cuan predictivas son estas variables (están ordenadas según esta cualidad).

### Variables numéricas

Debido al teorema central del límite [12], no hace falta normalizar los datos ya que disponemos de más de 30 observaciones.

```
> anova_fare <- aov(Fare ~ Survived, filteredData)
> summary(anova_fare)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Survived	1	101442	101442	<b>64.55</b>	<b>2.98e-15 ***</b>
Residuals	889	1397082	1572		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> anova_age <- aov(Age ~ Survived, filteredData)
> summary(anova_age)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Survived	1	1424	1424.1	<b>7.333</b>	<b>0.0069 **</b>
Residuals	889	172647	194.2		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En ambos casos, podemos ver que el valor de F [23] es elevado por lo que podemos decir que ambas son relevantes.

## Clasificación

En el apartado anterior hemos visto que todas las variables son relevantes para la supervivencia, con valores más elevados para: “Sex”, “Pclass” y “Fare”, “Age” y “Fsize”.

Vamos a utilizar el algoritmo C50 [24] para obtener un modelo de clasificación:

Antes de ejecutar el algoritmo cabe destacar que la variable “Embarked” tiene una clase vacía (“”), que aunque no tenga ninguna observación en ella, hará fallar el algoritmo C50. Por ello, le asignamos el valor “missing”.

```
> levels(filteredData$Embarked)[1] = "missing"
```

Aplicamos el algoritmo con el parámetro “rules” a “TRUE” para poder visualizar las reglas.

```
> library(C50)
> dim(filteredData)
[1] 891  9
> Y <- filteredData[,1]
> X <- filteredData[,2:9]
> model <- C50::C5.0(X, Y, rules=TRUE )
> summary(model)
```

En contraste con los resultados del punto anterior, vemos que “FSize” no ha sido utilizada por el modelo y sin embargo “SibSp” si, además de tener “PClass” menos importancia que la anterior.

Attribute usage:

100%	Sex
63.08%	Age
6.62%	SibSp
4.49%	Pclass
4.49%	Fare

Utilizando la función summary sobre el objeto modelo podemos ver las reglas obtenidas:

Rule 1	Sex = male & SibSp = 2 -> class 0 [0.881]
Rule 2	Pclass = 3 & Sex = female & Fare > 23.25-> class 0 [0.862]
Rule 3	Sex = male & Age > 9 -> class 0 [0.836]

Rule 4	Sex = male & Age <= 9 & SibSp in {0, 1} -> class 1 [0.952]
Rule 5	Pclass = 1 & Age <= 53 & 26 < Fare <= 27 -> class 1 [0.867]
Rule 6	Sex = female -> class 1 [0.741]

Analizamos las reglas:

1. Con un 88% si el pasajero es un hombre con 2 o más hermanos o cónyuges fallecerá.
2. Si el pasajero es una mujer en 3ra clase que pagó más de 23.25 por su billete fallecerá con una probabilidad del 86%.
3. Un hombre joven o adulto (mayor de 9 años) fallecerá con una probabilidad del 83%.
4. Si un niño (masculino) menor de 9 años tiene entre 0 y 1 hermanos sobrevivirá con un 95% de probabilidad.
5. Las personas mayores (mayores o igual a 53 años) de primera clase que pagaron entre 26 y 27 por su billete sobrevivirán con una probabilidad del 86%.
6. Las mujeres sobreviven con un 74% de probabilidades.

Si resumimos las reglas vemos un punto común: Los niños y las mujeres sobreviven en la mayoría de los casos (Ver regla 1, 3, 4 y 6), seguidos por los pasajeros de alta categoría socioeconómica (Ver regla 2 y 5).

### Reglas de asociación

Por último vamos a comprobar los resultados obtenidos del modelo de clasificación anterior mediante el uso de reglas de asociación.

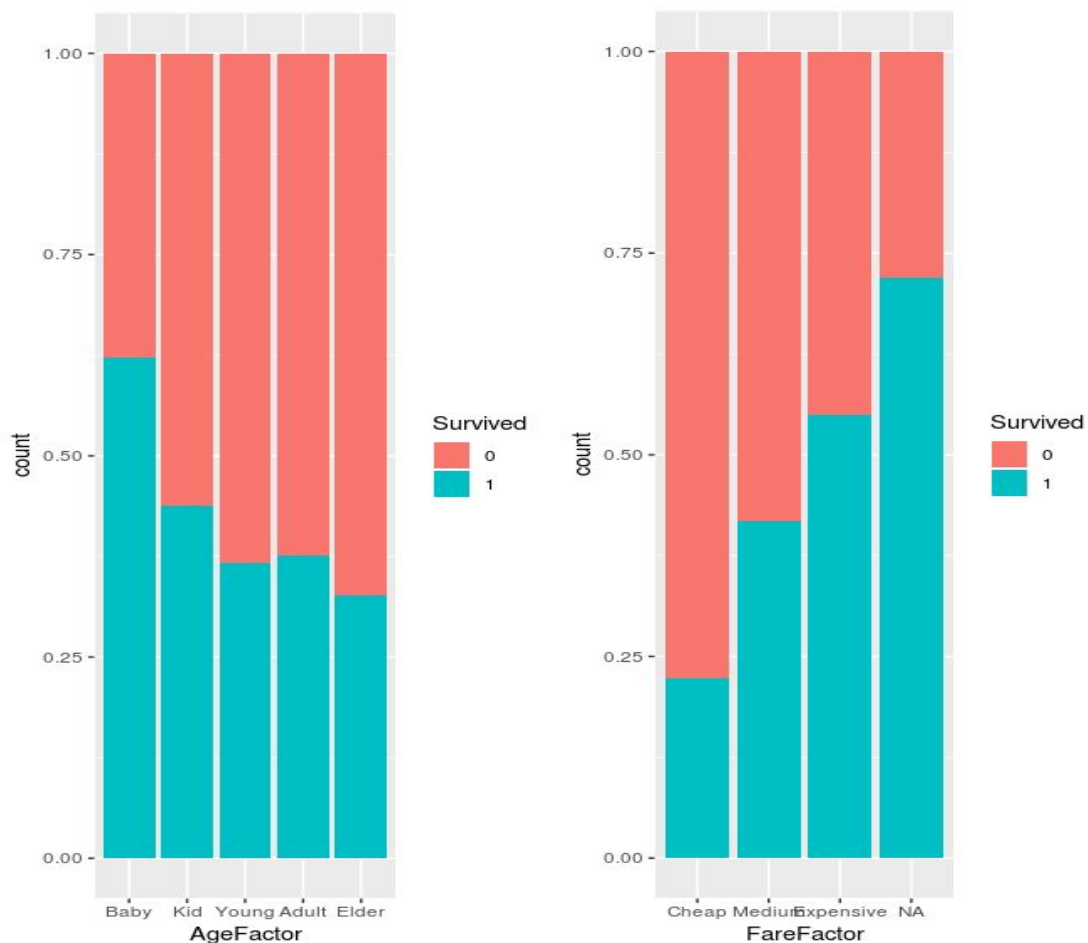
Para ello debemos binarizar los datos, por lo que tenemos que modificar las variables "Age" y "Fare":

```
# Edad
> filteredData$AgeFactor <- NA
> filteredData$AgeFactor[filteredData$Age < 5] = "Baby"
> filteredData$AgeFactor[filteredData$Age > 5 & filteredData$Age < 15] = "Kid"
> filteredData$AgeFactor[filteredData$Age >= 5 & filteredData$Age < 15] = "Kid"
> filteredData$AgeFactor[filteredData$Age >= 15 & filteredData$Age < 25] = "Young"
> filteredData$AgeFactor[filteredData$Age >= 25 & filteredData$Age < 50] = "Adult"
> filteredData$AgeFactor[filteredData$Age >= 50] = "Elder"
> filteredData$AgeFactor <- as.factor(filteredData$AgeFactor)
# Tarifa
> filteredData$FareFactor <- NA
> filteredData$FareFactor[filteredData$Fare < 12] = "Cheap"
> filteredData$FareFactor[filteredData$Fare >= 12 & filteredData$Fare < 30] = "Medium"
> filteredData$FareFactor[filteredData$Fare >= 30 & filteredData$Fare < 100] =
"Expensive"
> filteredData$FareFactor[filteredData$Fare >= 100] = "Ridiculous"
> filteredData$FareFactor <- as.factor(filteredData$FareFactor)
```

Comprobamos los resultados en los gráficos:

```
# Reorder factors
> filteredData$AgeFactor <- factor(filteredData$AgeFactor, levels = c("Baby", "Kid", "Young", "Adult", "Elder"))
> ggplot(data=filteredData, aes(x=AgeFactor, fill=Survived))+geom_bar(position="fill")
> filteredData$FareFactor <- factor(filteredData$FareFactor, levels = c("Cheap", "Medium", "Expensive", "Ridiculous"))
> ggplot(data=filteredData, aes(x=FareFactor, fill=Survived))+geom_bar(position="fill")
```

Vemos que las premisas se mantienen: Hay mayor porcentaje de niños y bebés, y pasajeros de alto estrato económico que sobreviven.



Eliminamos los atributos no categóricos y binarizamos los datos:

```
> asociaData <- select(filteredData, -Age, -Fare)
> library(arules)
> mba <- as(asociaData, "transactions")
> summary(mba)
transactions as itemMatrix in sparse format with
891 rows (elements/itemsets/transactions) and
29 columns (items) and a density of 0.3084098
```

```
most frequent items:
Parch=0 Embarked=S SibSp=0 Sex=male Survived=0 (Other)
678 646 608 577 549 4961
```

Vemos que se han generado 29 columnas, y debajo lo atributos más comunes.

Procedemos a aplicar las reglas de asociación con un nivel de soporte del 5% y una confianza del 95%. También forzamos a que el resultado sea “Survived=1” para obtener reglas relevantes. Utilizaremos el la función *apriori* del paquete *arules* [25].

```
> rules <- apriori(mba, parameter = list(supp = 0.05, conf = 0.95),
appearance=list(rhs='Survived=1',default='lhs'))
```

Vemos que se han generado 6 reglas:

```
> rules <- sort(rules, by="confidence", decreasing=TRUE)
> inspect(rules[1:6], ruleSep = "---->", itemSep = " + ", setStart = "", setEnd = "", linebreak =
FALSE)
```

Estas no mencionan nada con respecto a la edad. Sin embargo, corroboran que las mujeres, y más aún las de clase social alta, sobreviven. Sin ir más lejos, la primera regla lo corrobora con un soporte del 6% ¡Y una confianza del 100%!

Regla	Soporte	Confianza
Pclass=1 + Sex=female + FareFactor=Expensive	0.06172840	1.0000000
Pclass=1 + Sex=female + Parch=0	0.07070707	0.9843750
Pclass=1 + Sex=female + SibSp=0	0.05387205	0.9795918
Pclass=1 + Sex=female + AgeFactor=Adult	0.05274972	0.9791667
Pclass=1 + Sex=female	0.10213244	0.9680851
Pclass=1 + Sex=female + Embarked=S	0.05387205	0.9600000

## Representación de los resultados a partir de tablas y gráficas

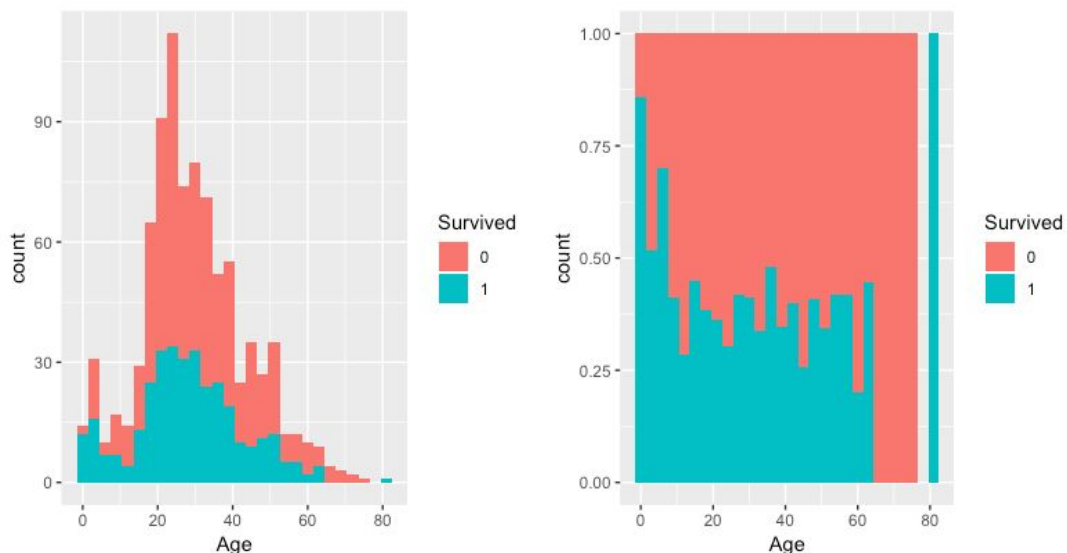
En los gráficos los mostraremos por total y en forma de porcentaje (proporción acumulada). Los primeros estarán a la izquierda y los segundos a la derecha.

```
# Género
> ggplot(data=filteredData,aes(x=Sex,fill=Survived))+geom_bar()
> ggplot(data=filteredData,aes(x=Sex,fill=Survived))+geom_bar(position="fill")
```

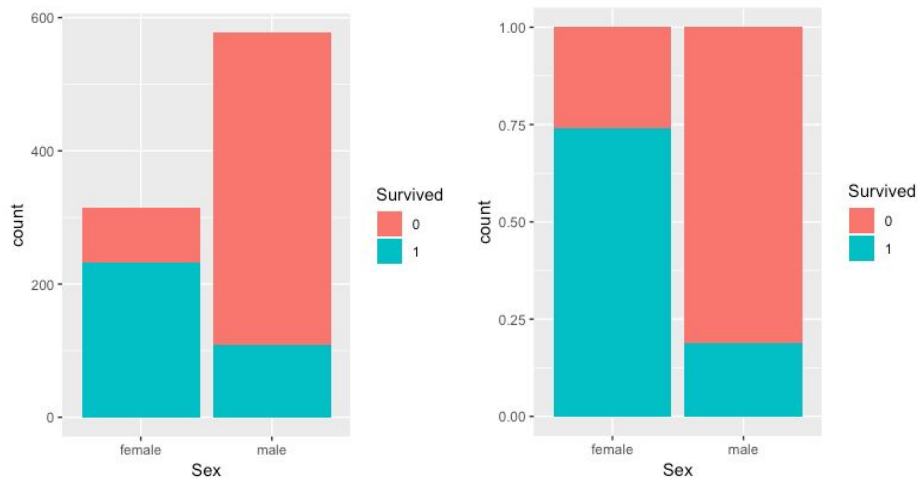
```
# Edad
> ggplot(data=filteredData,aes(x=Age,fill=Survived))+geom_histogram(binwidth=3)
> ggplot(data=filteredData,aes(x=Age,fill=Survived))+geom_histogram(binwidth=3,
position="fill")
# Género y Edad combinadas
> ggplot(data=filteredData, aes(x=Age, fill=Survived))+geom_histogram(binwidth =
10)+facet_wrap(~Sex)
> ggplot(data=filteredData, aes(x=Age, fill=Survived))+geom_histogram(binwidth = 10,
position="fill")+facet_wrap(~Sex)
```

### ¿Tienen más posibilidades de salvarse los hombres o las mujeres? ¿De qué edad?

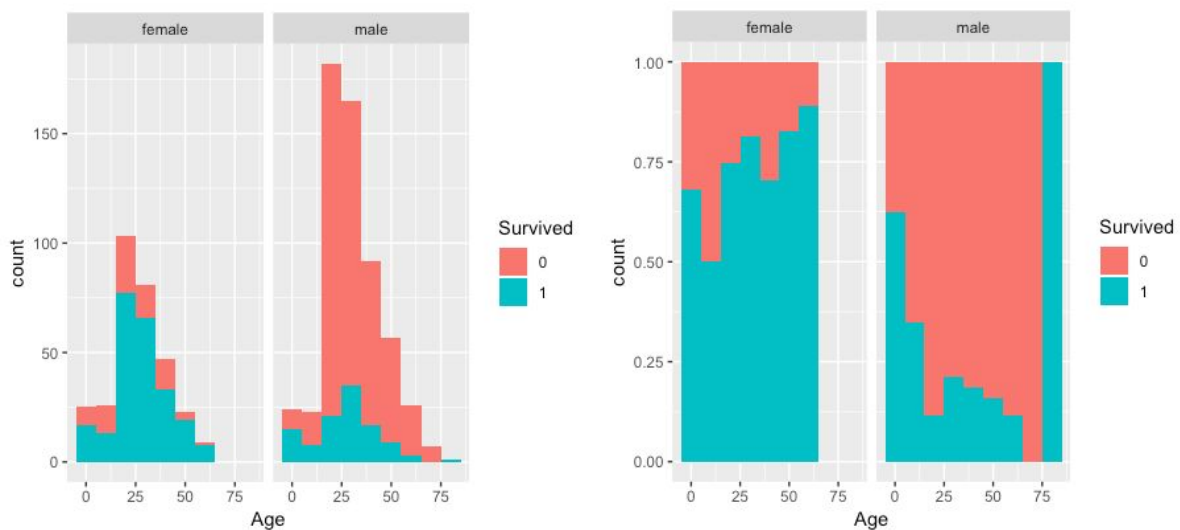
En las dos siguientes gráficas podemos ver la cantidad y el porcentaje de supervivientes según su edad. Llegamos a la conclusión de que aunque la mayor parte de los pasajeros estuvieran entre 18 y 50 años, los grupos con más probabilidad de salvarse son los bebés/niños (entre 0 y 15 años) y los ancianos (80 años).



En cuanto al género, podemos ver que el número de hombres presentes en el barco representa casi dos tercios del total. Sin embargo, vemos que el total de mujeres supervivientes es casi el doble que el de varones. Podemos observar que aproximadamente un 75% de mujeres se salvaron frente a un 25% de hombres.



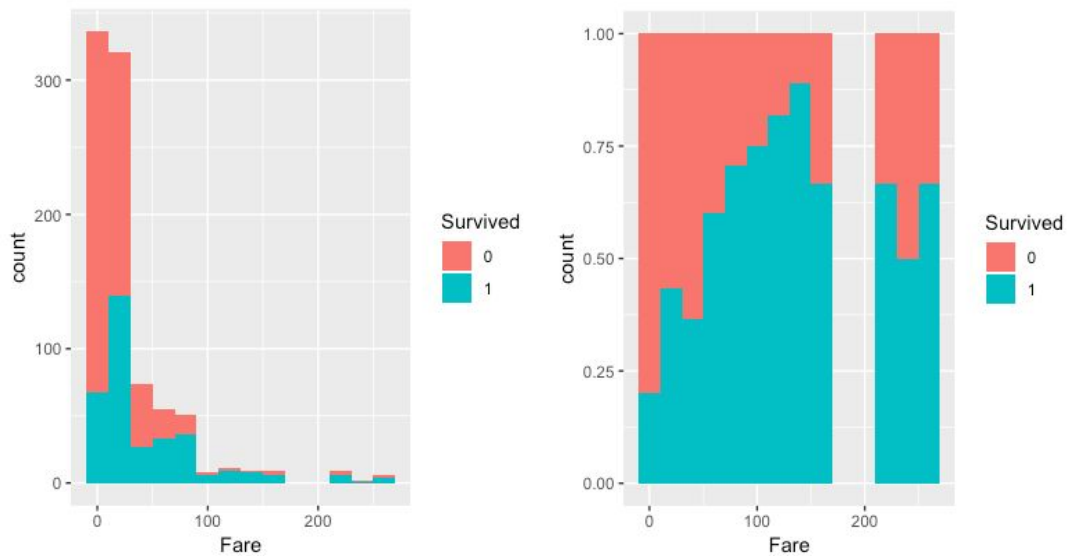
Si agrupamos por sexo en función de la edad, obtenemos que efectivamente se cumple la premisa: Las mujeres y los niños primero. En las siguientes gráficas podemos ver que el porcentaje de mujeres supervivientes es elevado independientemente de la edad, pero que en el caso de los hombres el grueso de supervivientes se encuentra en los niños.



### ¿Tienen más posibilidades de salvarse los pasajeros de clase socioeconómica elevada?

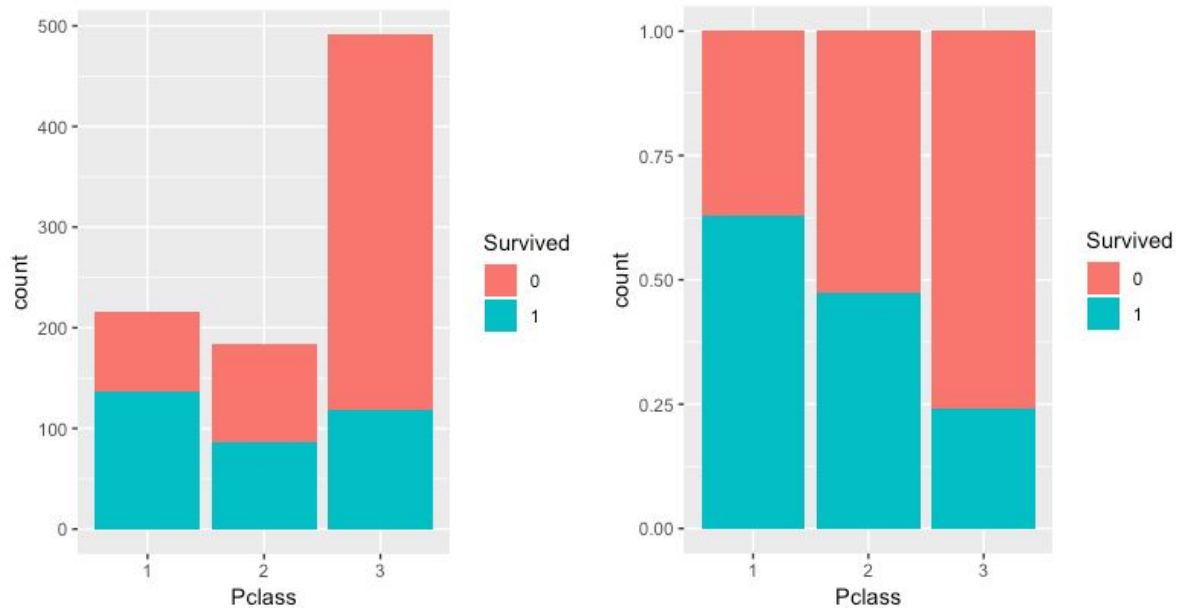
La respuesta parece evidente: Sí. Podemos ver en los gráficos del precio que cuanto más alto era el precio del billete, mayor porcentaje de supervivientes hubo. Sin embargo, también se puede apreciar que el número de pasajeros decrece cuanto más alto es el coste del billete.

```
# Tarifa
> ggplot(data=filteredData,aes(x=Fare,fill=Survived))+geom_histogram(binwidth=20)
> ggplot(data=filteredData,aes(x=Fare,fill=Survived))+geom_histogram(binwidth=20,
position="fill")
```



Los gráficos correspondientes a “Pclass” corroboran este hecho. A pesar de haber un mayor porcentaje de pasajeros en 3ra clase (más del doble que de 1ra y 2da), el porcentaje de supervivientes es de aproximadamente un 25%, menos que la mitad que los de 1ra clase y aproximadamente la mitad que los de 2da clase.

```
# Clase
> ggplot(data=filteredData,aes(x=Pclass,fill=Survived))+geom_bar()
> ggplot(data=filteredData,aes(x=Pclass,fill=Survived))+geom_bar(position="fill")
```



**¿Tienen más posibilidades de salvarse las familias? ¿De qué tamaño?**

En este caso estudiaremos las variables “SibSp”, “Parch” y “Fsize”.

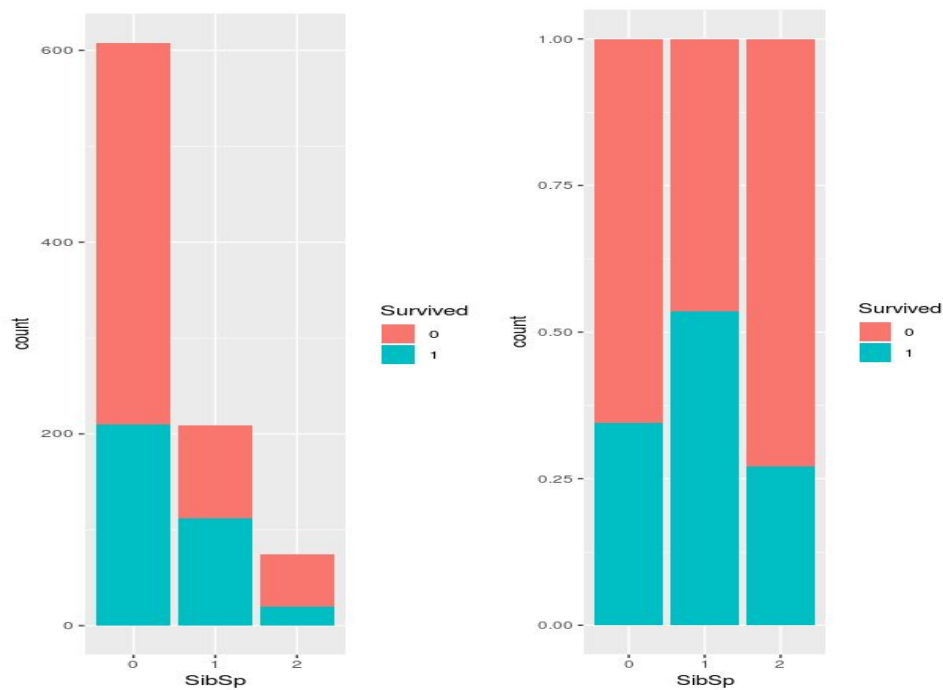


Comenzamos por “SibSp”, que si recordamos representa el número de hermanos y cónyuges que se encontraban a bordo del Titanic. En las gráficas podemos observar, que el mayor porcentaje de supervivientes se encuentra en aquellos con 1 parientes a bordo.

```
# Hermanos y conyuges
```

```
> ggplot(data=filteredData,aes(x=SibSpFactor,fill=Survived))+geom_bar()
```

```
> ggplot(data=filteredData,aes(x=SibSpFactor,fill=Survived))+geom_bar(position="fill")
```

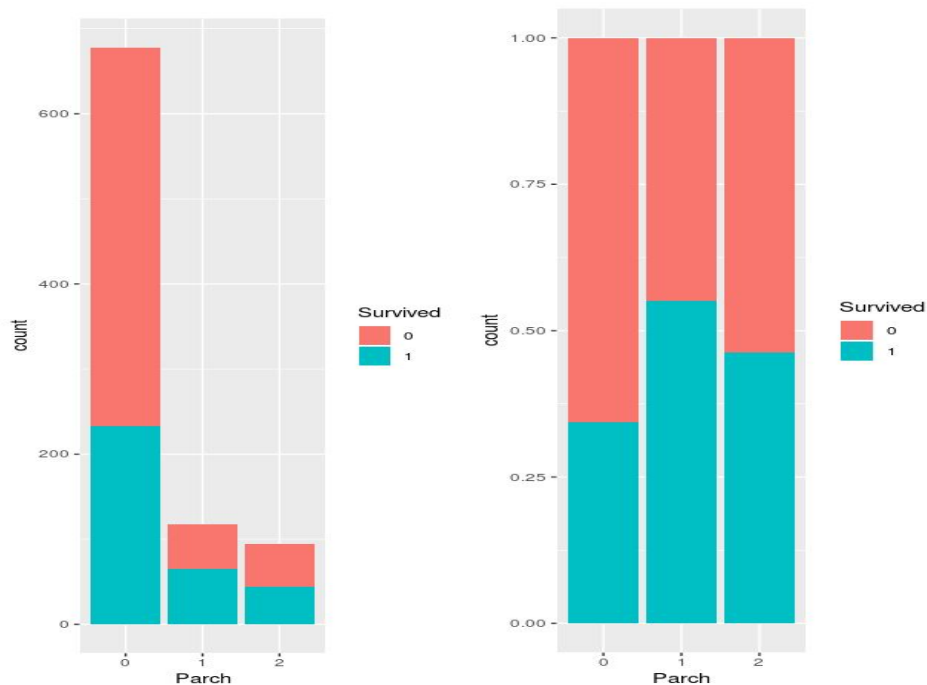


Vemos que sucede con “Parch”, que representa el número de padres y/o hijos a bordo. Al igual que antes los supervivientes tenían entre 1 padre o hijo en el Titanic. Seguidos por aquellos con 2 o más.

```
# Padres e hijos
```

```
> ggplot(data=filteredData,aes(x=ParchFactor,fill=Survived))+geom_bar()
```

```
> ggplot(data=filteredData,aes(x=ParchFactor,fill=Survived))+geom_bar(position="fill")
```

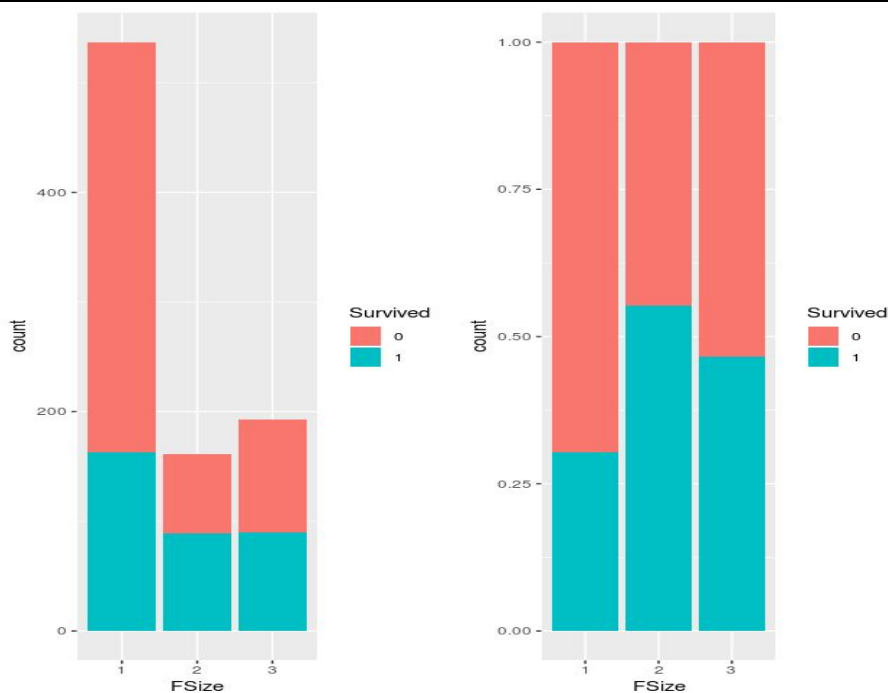


Corroboramos estos hechos mediante las gráficas de “FSize” o tamaño de la familia. Vemos que la mayoría de los supervivientes tienen tamaño de familia de 2 o más, lo cual concuerda con las conclusiones obtenidas en “SibSp” y “Parch” (Ej. “SibSp” = 1, es decir un cónyuge, y “Parch” = 1 ó 2, es decir 1 o 2 hijos. Harían un “FSize” de 2, 3 o más miembros).

```
# Tamaño de la familia
```

```
> ggplot(data=filteredData,aes(x=FSize,fill=Survived))+geom_bar()
```

```
> ggplot(data=filteredData,aes(x=FSize,fill=Survived))+geom_bar(position="fill")
```



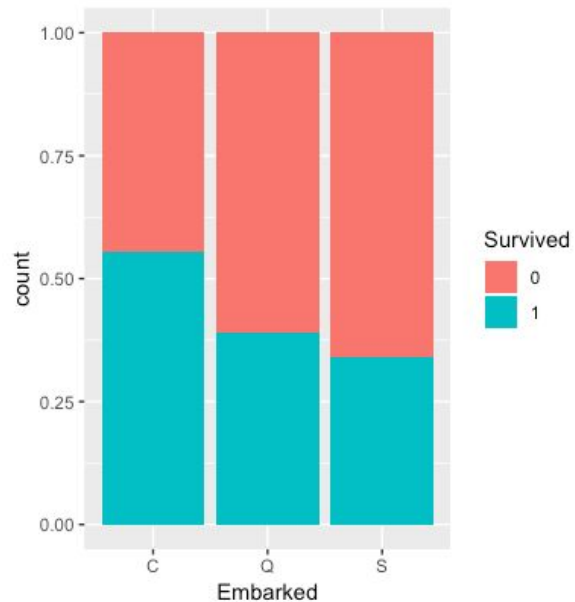
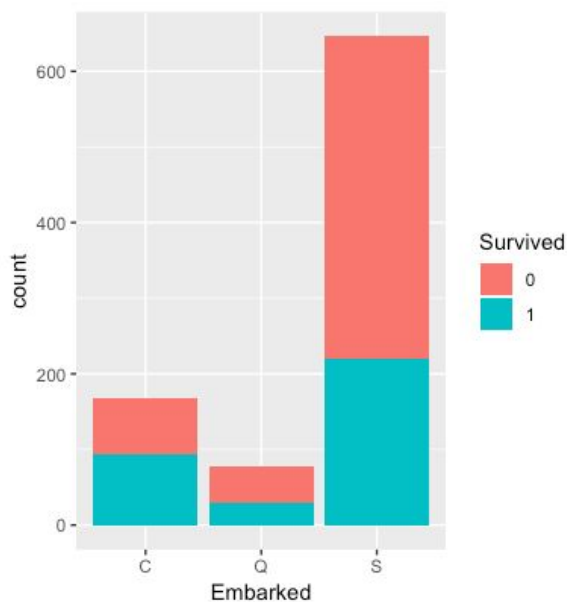
## ¿Existe alguna relación entre el puerto de embarque y la supervivencia?

En las gráficas siguientes podemos ver que el número de pasajeros que subieron al Titanic en Southampton (S) es casi el triple que en Cherbourg (C) y aproximadamente 6 veces que en Queenstown (Q). Sin embargo, no destaca ningún puerto con respecto al porcentaje de supervivientes.

```
# Puerto de embarcación
```

```
> ggplot(data=filteredData,aes(x=Embarked,fill=Survived))+geom_bar()
```

```
> ggplot(data=filteredData,aes(x=Embarked,fill=Survived))+geom_bar(position="fill")
```



## Contraste de hipótesis

En los apartados anteriores formulamos ciertas hipótesis:

- Los grupos con más probabilidad de salvarse son los bebés/niños (entre 0 y 15 años) y los ancianos (80 años).
- Podemos observar que aproximadamente un 75% de mujeres se salvaron frente a un 25% de hombres.
- Los pasajeros con de clase socioeconómica superior (1ra clase) tienen más probabilidad de sobrevivir.
- Los pasajeros con familia de 2 o más miembros a bordo sobrevivieron en mayor porcentaje.

A continuación realizamos pruebas de contraste de hipótesis [26] para comprobar que los conclusiones obtenidas son acertadas. Para ello utilizaremos el test de Mann-Whitney<sup>2</sup> [27] sobre las variables "Age", "Sex", "Pclass" y "FSize".

---

<sup>2</sup> Test no paramétrico del cual según [26] nuestros datos cumplen las premisas.

En primer lugar deberemos convertir algunas de ellas a numéricas:

```
> filteredData$PclassNum <- as.numeric(levels(filteredData$Pclass))[filteredData$Pclass]
> filteredData$FSizeNum <- as.numeric(levels(filteredData$FSize))[filteredData$FSize]
> filteredData$SexNum <- NA
> filteredData$SexNum[filteredData$Sex == 'male']=1
> filteredData$SexNum[filteredData$Sex == 'female']=0
```

Los resultados del test de Mann-Whitney son:

```
> wilcox.test(Age~Survived, filteredData)

    Wilcoxon rank sum test with continuity correction

data:  Age by Survived
W = 101790, p-value = 0.03424
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(SexNum~Survived, filteredData)

    Wilcoxon rank sum test with continuity correction

data:  SexNum by Survived
W = 143990, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(PclassNum~Survived, filteredData)

    Wilcoxon rank sum test with continuity correction

data:  PclassNum by Survived
W = 127940, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(FSizeNum~Survived, filteredData)

    Wilcoxon rank sum test with continuity correction

data:  FSizeNum by Survived
W = 76012, p-value = 4.556e-08
alternative hypothesis: true location shift is not equal to 0
```

Observamos que los p-value son menores que 0.05 y por lo tanto rechazamos la hipótesis nula (las medias son iguales o siguen la misma distribución, dependiendo de la “forma” del histograma de cada uno de los grupos). En ambos casos, y para todos los tests, podemos concluir que las hipótesis se contrastan (los grupos son significativamente diferentes).

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

El problema planteado en el primer apartado:

*Entender las características que permitieron a unos pasajeros sobrevivir y a otros no.*

Tiene como respuesta:

***Las mujeres y los niños primero, aunque en este caso incluimos también a los pasajeros de alto poder adquisitivo (1ra clase).***

Para llegar a estas conclusiones, tras limpiar el dataset y seleccionado los datos relevantes para el problema, se comprobaron la normalidad y homogeneidad de las variables numéricas (que resultaron no serlo). En el siguiente paso se realizaron análisis de significancia e independencia de las variables utilizando los test de *Chi-Square* y *ANOVA*, y se comprobó que las variables son relevantes para predecir la supervivencia de un pasajero.

Después, se generó un modelo de clasificación utilizando árboles de decisión, y se comprueban sus resultados mediante las reglas de asociación. Sin embargo, para este último se realizó una discretización de las variables numéricas ("Fare" y "Age") que puede no ser la más apropiada.

Por último, se realizaron pruebas de contraste de hipótesis sobre las conclusiones obtenidas para comprobar que estas son significantes.

A continuación pasamos a comentar las posibles mejoras:

- El análisis de correlación se hizo sólo univariable. Podría ser interesante realizar análisis multivariable, lo cual quizá permita realizar una ingeniería de los atributos mejor.
- Comprobar la predictibilidad del modelo obtenido y probar con diferentes parámetros del algoritmo C50 (Ej. el número de *trials*).
- Mejorar la discretización de las variables "Fare" y "Age", ya que no se utilizó ningún método estadístico, sino que se realizó mediante la visualización de su histograma.

Finalmente mencionar que en el futuro, se realizará el mismo proceso de limpieza con el dataset de prueba (test.csv), lo cual permitirá comprobar la predictibilidad del modelo y la participación en Kaggle.

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 completo se encuentra en el fichero "ppanero\_prac2.r" presente en el repositorio [GitHub "UOC\\_Projects"](#), en la [rama "tipología"](#). Solo habría que cambiar el valor de la

variable “datasetDir” si los datasets se encontraran en una carpeta diferente que el directorio de trabajo en el cual se ejecuta el código.

## Referencias

1. Squire, Megan (2015). Clean Data. Packt Publishing Ltd.
2. Jiawei Han, Micheline Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
3. Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
4. Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
5. The use of kNN for missing values. [Towards Data Science](#).
6. k-Nearest Neighbour Imputation. [R Documentation](#).
7. Outlier treatment with R. [R Statistics](#).
8. R Boxplots. [Statmethods](#).
9. Normality test with R. [Sthda](#).
10. Understanding Q-Q plots. [University of Virginia Library](#).
11. Shapiro Test. [ETH University](#).
12. How to test normality in a formal way in R. [For Dummies](#).
13. Toby Mordkoff, J (2016). The assumption(s) of normality.
14. Central limit theorem and the normality assumption. [Analyse-it](#).
15. The assumption of homogeneity of variance. [Statistics Solutions](#).
16. Why is homogeneity of variance importante. [Stackexchange](#).
17. Compare multiple sample variances in R. [Sthda](#).
18. Homogeneity of variance. [R Cookbook](#).
19. Chi-Square test. [R-Bloggers](#).
20. Choosing the Correct Statistical Test in SAS, Stata, SPSS and R. [UCLA Institute for Digital Research and Education Search this website](#).
21. Chi-Square Test. [Wikipedia](#).
22. ANOVA Test. [ETH University](#).
23. F Statistic / F Value: Simple Definition and Interpretation. [Statistics How To](#).
24. C50 Decision Trees and Rules based models. [R Documentation](#).
25. A Rules, Apriori Association Mining. [R Documentation](#).
26. Introduction to hypothesis testing in R. [Data Flair](#).
27. Mann-Whitney U test in SPSS. [Statistics Laerd](#).