Minería de datos: PEC1

Autor: Jorge Alonso Hernández

octubre 2021

Contents

Introducción **1**

Presentación 1

Objetivos 2

Descripción de la PEC a realizar 2

Recursos 2

Formato y fecha de entrega 2

Nota: Propiedad intelectual 2

Ejemplo de solución mínimo del ejercicio 2 **3**

Objetivos 3

Procesos iniciales con los datos 3

Procesos de análisis visuales del juego de datos 13

Conclusiones finales 23

Ejercicios **23**

Ejercicio 1: 23

Ejercicio 2: 23

Criterios de evaluación **25**

Introducción

Presentación

Esta prueba de evaluación continuada cubre los módulos "El proceso de minería de datos" y "Preprocesado de los datos y gestión de características" del programa de la asignatura.

Objetivos

- Asimilar correctamente los módulos citados.
- Qué es y que no es MD.
- Ciclo de vida de los proyectos de MD.
- Diferentes tipologías de MD.
- Conocer las técnicas propias de una fase de conocimiento, preparación de datos y objetivos a lograr.

Descripción de la PEC a realizar

La prueba está estructurada en 1 ejercicio teórico/práctico y 1 ejercicio práctico que pide que se desarrolle la fase de conocimiento y preparación con un juego de datos. Se tienen que responderse todos los ejercicios para poder superar la PEC. La PEC está pensada para resolverla en el entorno Markdown con RStudio con R como lenguaje preferido. Se recomienda hacerlo así. Si tenéis las competencias para hacerlo en Python no hay ningún problema. Podéis hacerlo. Simplemente sustituis los chunks de R por chunks en Python.

Recursos

Para realizar esta práctica recomendamos como punto de partida la lectura de los siguientes documentos:

- Los módulos "El proceso de minería de datos" y "Preprocesado de los datos y gestión de características" del programa de la asignatura.
- Ciclo de vida de un proyecto de minería de datos: https://es.wikipedia.org/wiki/cross_industry_standard_process_for_data_mining#Fases_principales
- Al apartado del enunciado de la actividad disponéis de unos materiales de ggplot2
- El aula laboratorio de R para resolver dudas o problemas.
- RStudio Cheat Sheet: Disponible en el aula Laboratorio de Minería de datos.
- R Base Cheat Sheet: Disponible en el aula Laboratorio de Minería de datos.

Formato y fecha de entrega

El formato de entrega es: **usernameestudiante-PEC1.html (pdf o word) y rmd**. Fecha de Entrega: 27/10/2021. Se tiene que librar la PEC en el buzón de entregas del aula.

Nota: Propiedad intelectual

A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por lo tanto comprensible hacerlo en el marco de una práctica de los estudios de Informática, Multimedia y Telecomunicación de la UOC, siempre que esto se documente claramente y no suponga plagio en la práctica.

Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se tiene que presentar junto con ella un documento en que se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar donde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL ...). El estudiante tendrá que asegurarse que la licencia no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por copyright.

Habréis, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.

Ejemplo de solución mínimo del ejercicio 2

Objetivos

Como muestra, trabajaremos con el juego de datos "Titanic.csv" que recoge datos sobre el famoso crucero.

Las actividades que llevaremos a cabo en esta práctica se hacen en las fases iniciales de un proyecto de minería de datos. Tienen como objetivo obtener un dominio de los datos con las que construiremos el modelo de minería. Tenemos que conocer profundamente los datos tanto en su formato como contenido. Tareas típicas pueden ser la selección de características o variables, la preparación del juego de datos para posteriormente ser consumido por un algoritmo e intentar extraer el máximo conocimiento posible de los datos. Desarrollaremos un subconjunto de tareas mínimas y de ejemplo. Podemos incluir muchas más y mucho más profundas, como hemos visto en el material docente.

Procesos iniciales con los datos

Primer contacto con el juego de datos.

Instalamos y cargamos las librerías ggplot2 y dplry.

```
# https://cran.r-project.org/web/packages/ggplot2/index.html
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
# https://cran.r-project.org/web/packages/dplyr/index.html
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
```

Cargamos el fichero de datos.

```
totalData <- read.csv('titanic.csv',stringsAsFactors = FALSE)
filas=dim(totalData)[1]
```

Guardamos los datos filtrados por tripulación para hacer estudios posteriores.

```
totalData_crew=subset(totalData, totalData$class=="engineering crew")
```

Verificamos la estructura del juego de datos principal.

```
str(totalData)
```

```
## 'data.frame':    2207 obs. of  11 variables:
##  $ name    : chr  "Abbing, Mr. Anthony" "Abbott, Mr. Eugene Joseph" "Abbott, Mr. Rossmore Edward" "Al
##  $ gender  : chr  "male" "male" "male" "female" ...
##  $ age     : num  42 13 16 39 16 25 30 28 27 20 ...
##  $ class   : chr  "3rd" "3rd" "3rd" "3rd" ...
##  $ embarked: chr  "S" "S" "S" "S" ...
##  $ country : chr  "United States" "United States" "United States" "England" ...
```

```
##  $ ticketno: int  5547 2673 2673 2673 348125 348122 3381 3381 2699 3101284 ...
##  $ fare    : num  7.11 20.05 20.05 20.05 7.13 ...
##  $ sibsp   : int  0 0 1 1 0 0 1 1 0 0 ...
##  $ parch   : int  0 2 1 1 0 0 0 0 0 0 ...
##  $ survived: chr  "no" "no" "no" "yes" ...
```

Vemos que tenemos 2207 registros que se corresponden a los viajeros y tripulación del Titánic y 11 variables que los caracterizan.

Revisamos la descripción de las variables contenidas al fichero y si los tipos de variable se corresponde al que hemos cargado:

name string with the name of the passenger.

gender factor with levels male and female.

age numeric value with the persons age on the day of the sinking. The age of babies (under 12 months) is given as a fraction of one year (1/month).

class factor specifying the class for passengers or the type of service aboard for crew members.

embarked factor with the persons place of of embarkment.

country factor with the persons home country.

ticketno numeric value specifying the persons ticket number (NA for crew members).

fare numeric value with the ticket price (NA for crew members, musicians and employees of the shipyard company).

sibsp ordered factor specifying the number if siblings/spouses aboard; adopted from Vanderbild data set.

parch an ordered factor specifying the number of parents/children aboard; adopted from Vanderbild data set.

survived a factor with two levels (no and yes) specifying whether the person has survived the sinking.

Vamos ahora a sacar estadísticas básicas y después trabajamos los atributos con valores vacíos.

```
summary(totalData)
```

```
##      name              gender              age             class
##  Length:2207        Length:2207        Min.   : 0.1667   Length:2207
##  Class :character   Class :character   1st Qu.:22.0000   Class :character
##  Mode  :character   Mode  :character   Median :29.0000   Mode  :character
##                                        Mean   :30.4367
##                                        3rd Qu.:38.0000
##                                        Max.   :74.0000
##                                        NA's   :2
##    embarked           country             ticketno            fare
##  Length:2207        Length:2207        Min.   :      2   Min.   :  3.030
##  Class :character   Class :character   1st Qu.:  14262   1st Qu.:  7.181
##  Mode  :character   Mode  :character   Median : 111427   Median : 14.090
##                                        Mean   : 284216   Mean   : 33.405
##                                        3rd Qu.: 347077   3rd Qu.: 31.061
##                                        Max.   :3101317   Max.   :512.061
##                                        NA's   :891       NA's   :916
##      sibsp            parch          survived
##  Min.   :0.0000   Min.   :0.0000   Length:2207
##  1st Qu.:0.0000   1st Qu.:0.0000   Class :character
```

```
##  Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :0.4996   Mean   :0.3856
##  3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :8.0000   Max.   :9.0000
##  NA's   :900      NA's   :900
```

Estadísticas de valores vacíos.

```
colSums(is.na(totalData))
```

```
##     name   gender      age    class embarked  country ticketno     fare
##        0        0        2        0        0       81      891      916
##    sibsp    parch survived
##      900      900        0
```

```
colSums(totalData=="")
```

```
##     name   gender      age    class embarked  country ticketno     fare
##        0        0       NA        0        0       NA       NA       NA
##    sibsp    parch survived
##       NA       NA        0
```

Asignamos valor "Desconocido" para los valores vacíos de la variable "country".

```
totalData$country[is.na(totalData$country)] <- "Desconocido"
```

Asignamos la media para valores vacíos de la variable "age".

```
totalData$age[is.na(totalData$age)] <- mean(totalData$age,na.rm=T)
```

De la información mostrada destacamos que el pasajero más joven tenía 6 meses y el más grande 74 años. La media de edad la tenían en 30 años. También podemos ver 891 sin billete. Revisaremos si se corresponde a la tripulación. También podemos observar el que se pagó por el billete. En este caso se entienden las discrepancias en la fiabilidad de este dato. Parece que los pasajeros que embarcaron a Southampton hacían transbordo de un barco que tenía la tripulación en huelga y por eso no tuvieron que pagar lo que explicaría la diferencia. Recordemos que la tripulación no pagaba. Sibsp y parch también muestran datos interesantes el viajero con quien más familiar viajaba eran 8 hermanos o mujer y 9 hijos o paro/madre.

Si observamos los NA (valores nulos) vemos que los datos están bastante bien. Decidimos sustituir el valor NA de country por Desconocido por una mayor legibilidad. También proponemos sustituir los NA de age por la media a pesar de que realmente no hace falta.

Es curios como los valores NA de sibsp y parch nos permite deducir que viajaban muchas familias. De hecho a simple vista, restante la tripulación la gente que viajaba sola era mínima. Este dato la podríamos contrastar también. Sería interesante relacionar la mortalidad del accidente con el tamaño de las familias que viajaban.

Ahora añadiremos un campo nuevo a los datos. Este campos contendrá el valor de la edad discretitzada con un método simple de intervalos de igual amplitud.

```
summary(totalData[,"age"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1667 22.0000 29.0000 30.4367 38.0000 74.0000
```

Discretizamos con intervalos.

```
totalData["segmento_edad"] <- cut(totalData$age, breaks = c(0,10,20,30,40,50,60,70,100), labels = c("0-9
```

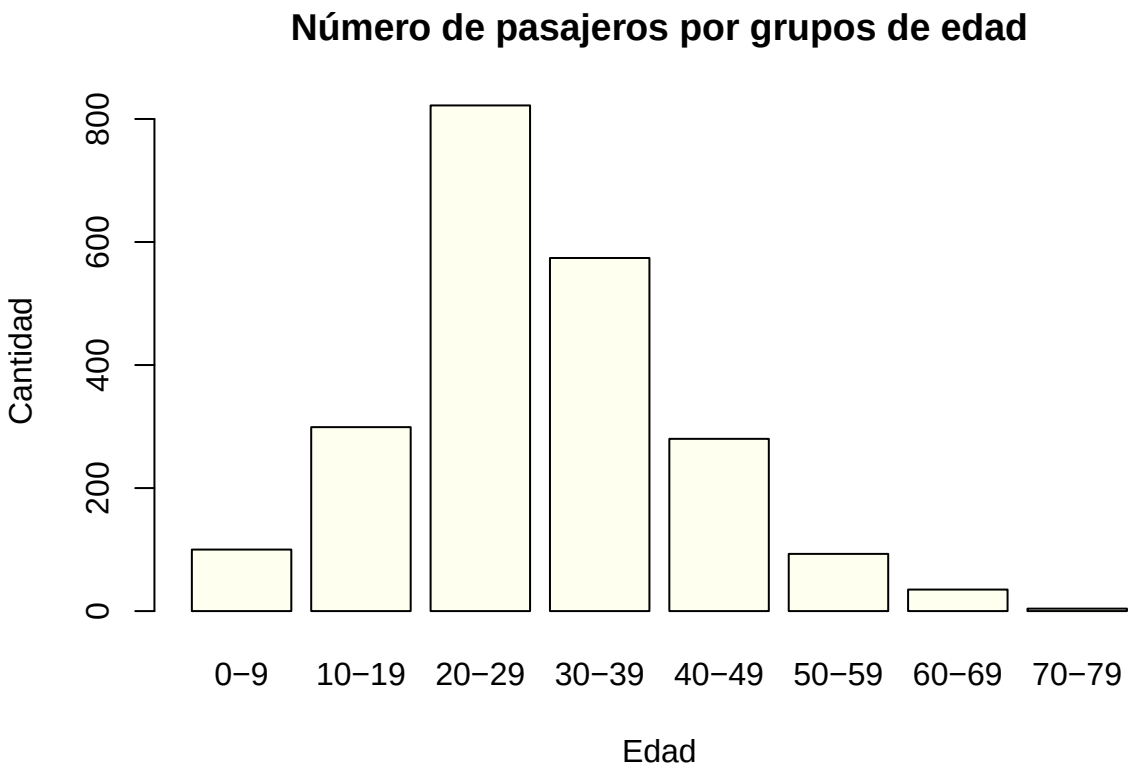
Observamos los datos discretizados.

```
head(totalData)
```

```
##                             name gender age class embarked       country
## 1            Abbing, Mr. Anthony   male  42   3rd        S United States
## 2      Abbott, Mr. Eugene Joseph   male  13   3rd        S United States
## 3    Abbott, Mr. Rossmore Edward   male  16   3rd        S United States
## 4 Abbott, Mrs. Rhoda Mary 'Rosa' female  39   3rd        S       England
## 5    Abelseth, Miss. Karen Marie female  16   3rd        S        Norway
## 6 Abelseth, Mr. Olaus JÃ¸rgensen   male  25   3rd        S United States
##   ticketno  fare sibsp parch survived segmento_edad
## 1     5547  7.11     0     0       no         40-49
## 2     2673 20.05     0     2       no         10-19
## 3     2673 20.05     1     1       no         10-19
## 4     2673 20.05     1     1      yes         30-39
## 5   348125  7.13     0     0      yes         10-19
## 6   348122  7.13     0     0      yes         20-29
```

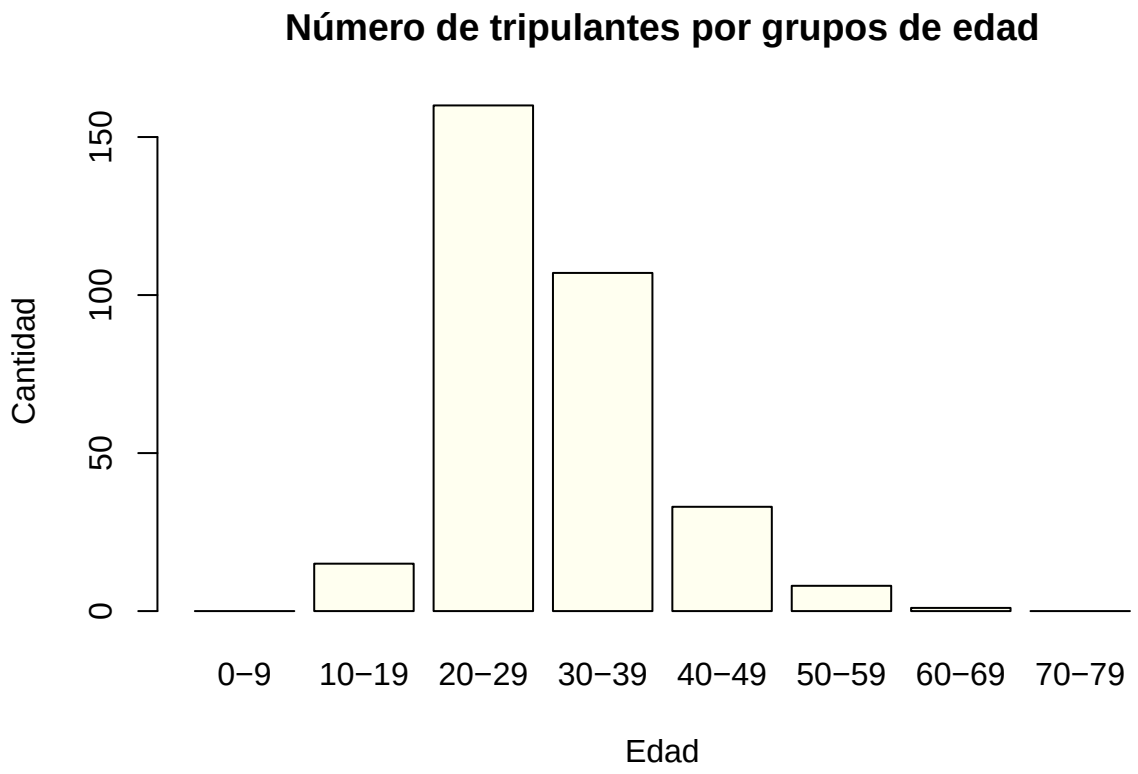
Vemos como se agrupaban por edad.

```
plot(totalData$segmento_edad,main="Número de pasajeros por grupos de edad",xlab="Edad", ylab="Cantidad"
```



Ahora repetimos por el proceso pero solo por el subconjunto de tripulación filtrado antes.

```
totalData_crew["segmento_edad"] <- cut(totalData_crew$age, breaks = c(0,10,20,30,40,50,60,70,100), labe
plot(totalData_crew$segmento_edad,main="Número de tripulantes por grupos de edad",xlab="Edad", ylab="Ca
```



De la discretizaón de la edad observamos que realmente la gente que viajaba era muy joven. El segmento más grande era de 20 a 29 años. También vemos de la juventud de la tripulación.

Como alternativa a la discretización realizada discretizaremos ahora edad con kmeans.

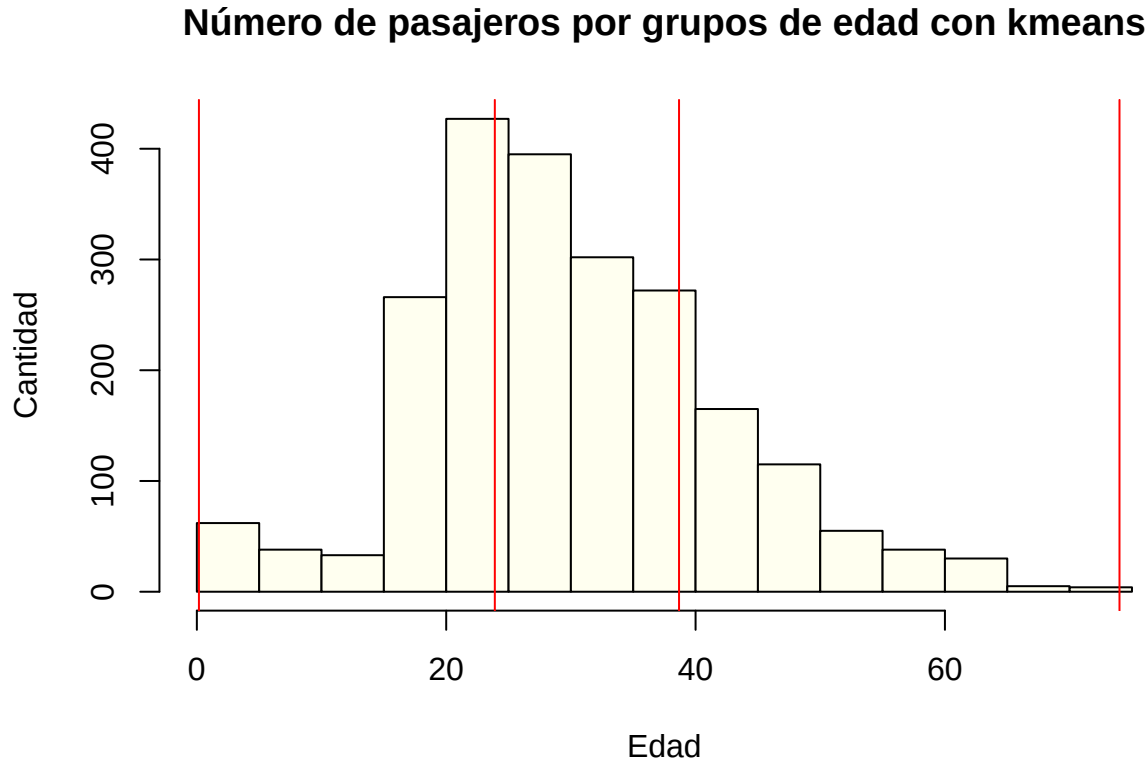
```
# https://cran.r-project.org/web/packages/arules/index.html
if (!require('arules')) install.packages('arules'); library('arules')
```

```
## Loading required package: arules

## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
set.seed(2)
table(discretize(totalData$age, "cluster" ))
```

```
##
## [0.167,25.4)     [25.4,40)       [40,74]
##          826           916           465
```

```
hist(totalData$age, main="Número de pasajeros por grupos de edad con kmeans",xlab="Edad", ylab="Cantidad")
abline(v=discretize(totalData$age, method="cluster", onlycuts=TRUE),col="red")
```



Podemos observar que sin pasar ningún argumento y que el algoritmo escoja el conjunto de particiones se muestran tres clústeres que agrupan las edades en las franjas mencionadas. Podemos asignar el propio clúster como una variable más al dataset para trabajar después.

```
totalData$edad_KM <- (discretize(totalData$age, "cluster" ))
head(totalData)
```

```
##                             name gender age class embarked       country
## 1              Abbing, Mr. Anthony   male  42   3rd        S United States
## 2          Abbott, Mr. Eugene Joseph   male  13   3rd        S United States
## 3        Abbott, Mr. Rossmore Edward   male  16   3rd        S United States
## 4 Abbott, Mrs. Rhoda Mary 'Rosa' female  39   3rd        S       England
## 5        Abelseth, Miss. Karen Marie female  16   3rd        S        Norway
## 6 Abelseth, Mr. Olaus JÃ¸rgensen   male  25   3rd        S United States
```

```
##   ticketno  fare sibsp parch survived segmento_edad      edad_KM
## 1     5547  7.11     0     0       no         40-49    [38.7,74]
## 2     2673 20.05     0     2       no         10-19 [0.167,23.9)
## 3     2673 20.05     1     1       no         10-19 [0.167,23.9)
## 4     2673 20.05     1     1      yes         30-39    [38.7,74]
## 5   348125  7.13     0     0      yes         10-19 [0.167,23.9)
## 6   348122  7.13     0     0      yes         20-29  [23.9,38.7)
```

Ahora normalizaremos la edad de los pasajeros por el máximo añadiendo un nuevo valor a los datos que contendrá el valor.

```
totalData$age_NM <- (totalData$age/max(totalData[,"age"]))
head(totalData$age_NM)
```

```
## [1] 0.5675676 0.1756757 0.2162162 0.5270270 0.2162162 0.3378378
```

Supongamos que queremos normalizar por la diferencia para ubicar entre 0 y 1 la variable edad del pasajero dado que el algoritmo de minería que utilizaremos así lo requiere. observamos la distribución de la variable original y las tres generadas

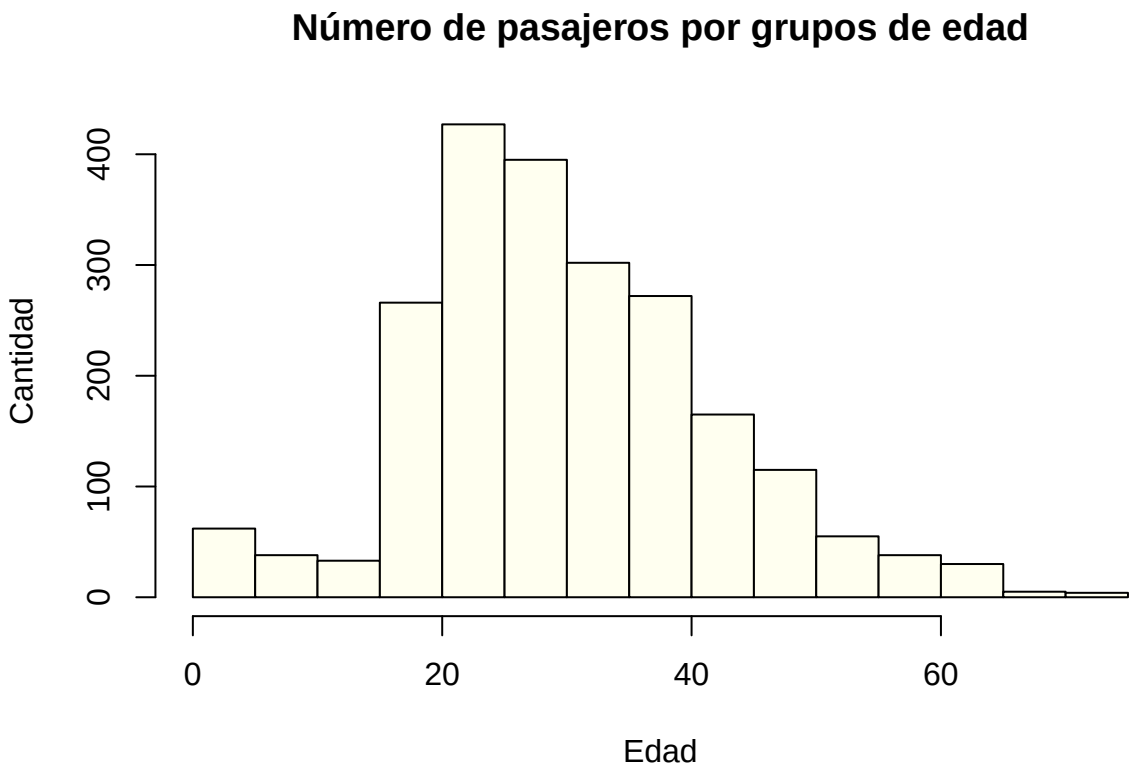
```
totalData$age_ND = (totalData$age-min(totalData$age))/(max(totalData$age)-min(totalData$age))

max(totalData$age)
```

```
## [1] 74
```

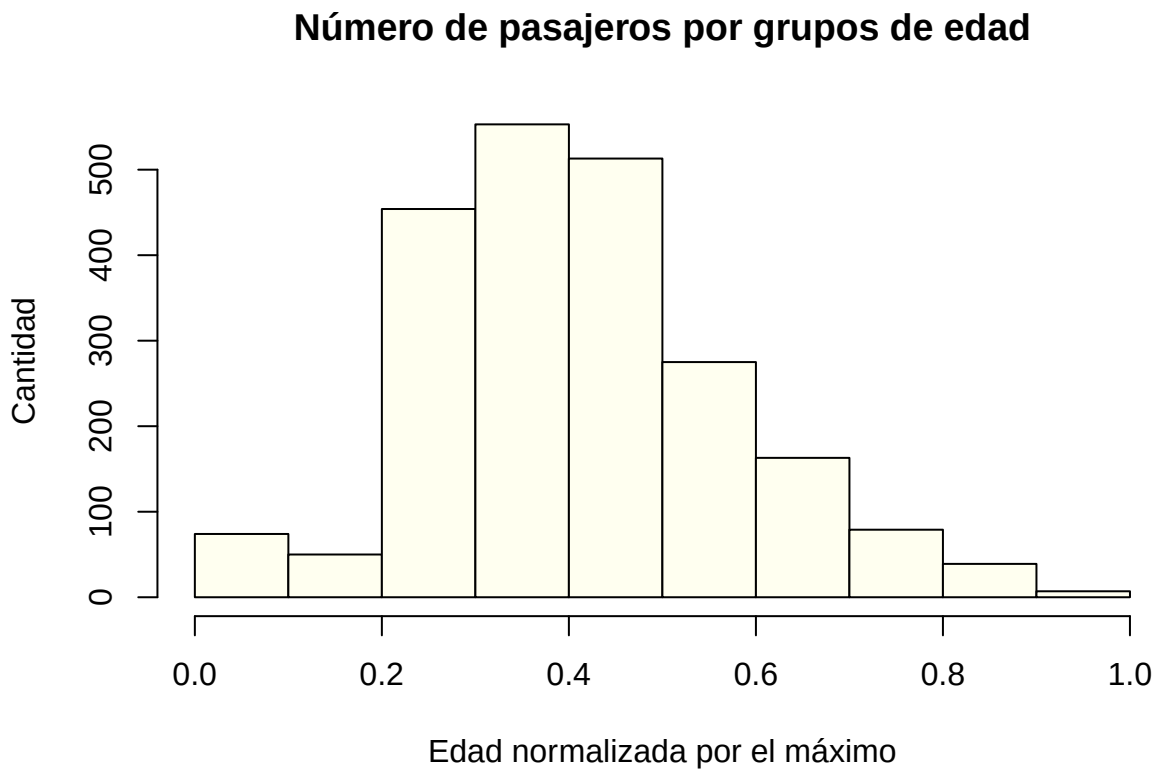
```
min(totalData$age)
```

```
## [1] 0.1666667
```

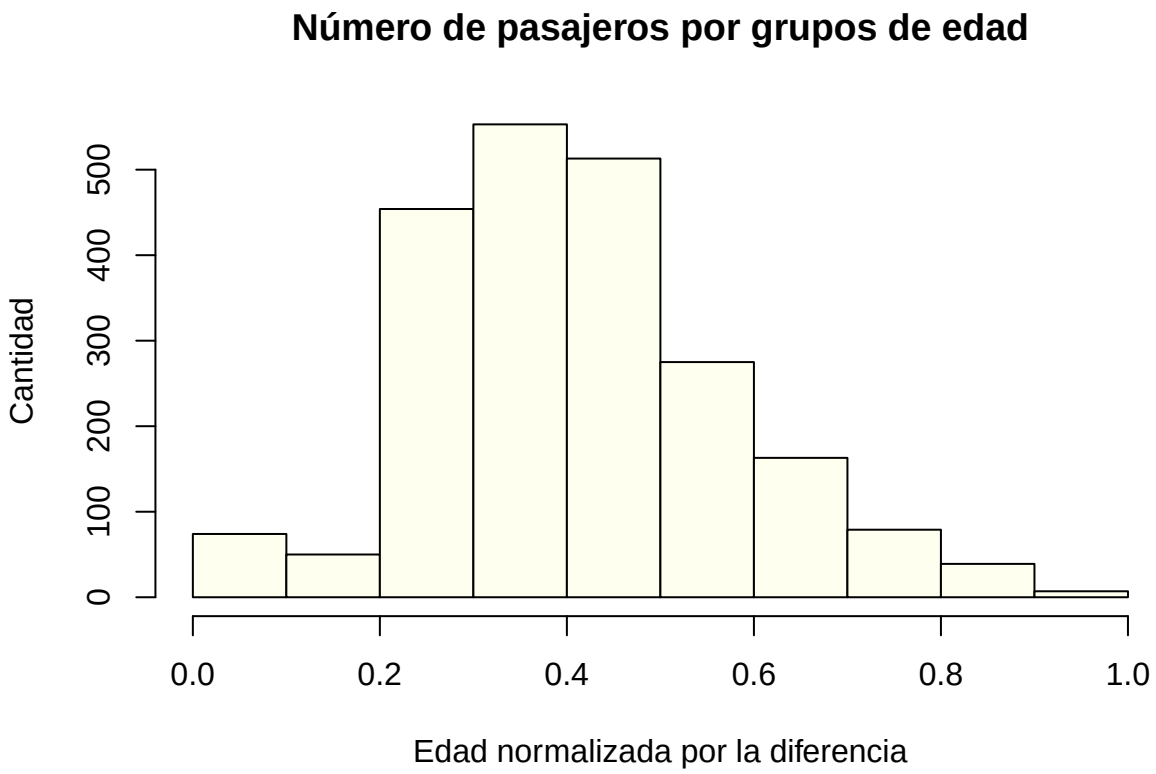
```
hist(totalData$age,xlab="Edad", col="ivory",ylab="Cantidad", main="Número de pasajeros por grupos de eda
```




```
hist(totalData$age_NM,xlab="Edad normalizada por el máximo", ylab="Cantidad",col="ivory", main="Número d
```




```
hist(totalData$age_ND,xlab="Edad normalizada por la diferencia",ylab="Cantidad", col="ivory", main="Núm
```

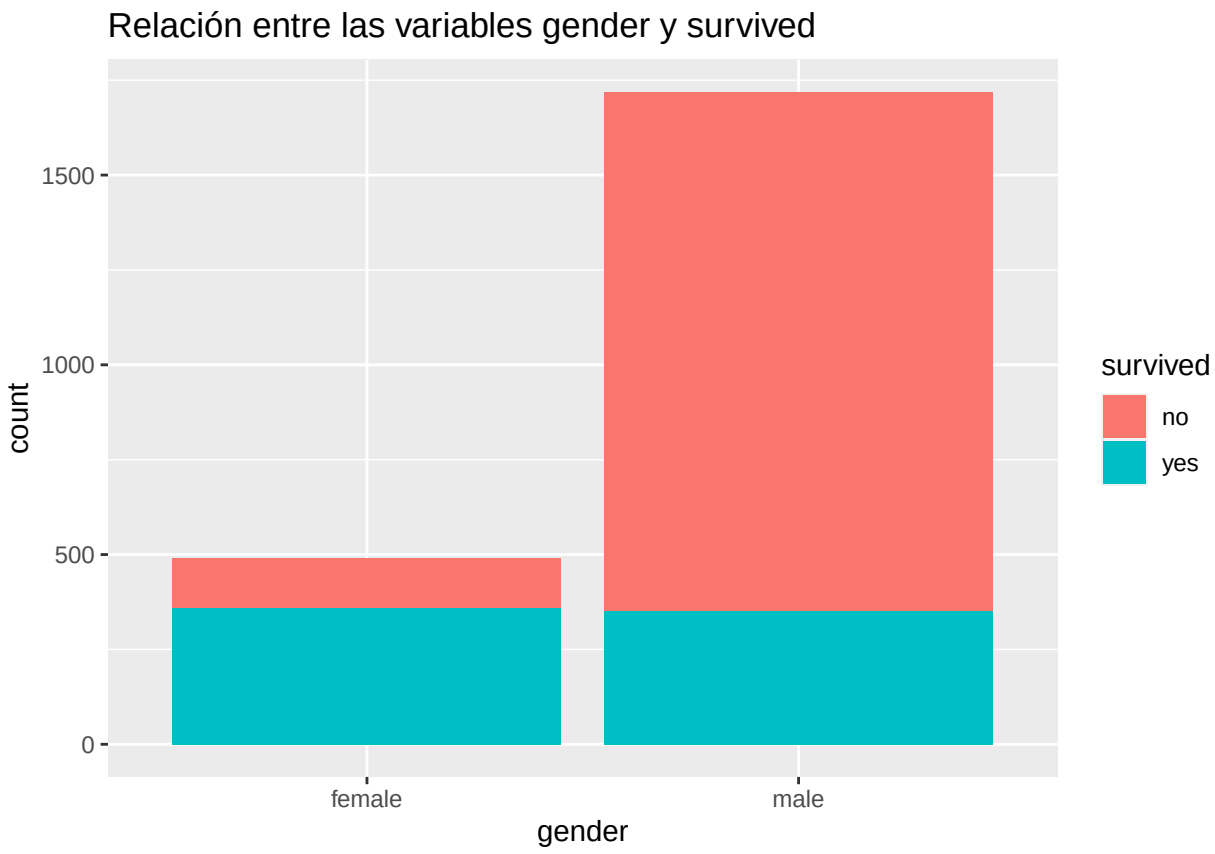


Procesos de análisis visuales del juego de datos

Nos proponemos analizar las relaciones entre las diferentes variables del juego de datos para ver si se relacionan y como.

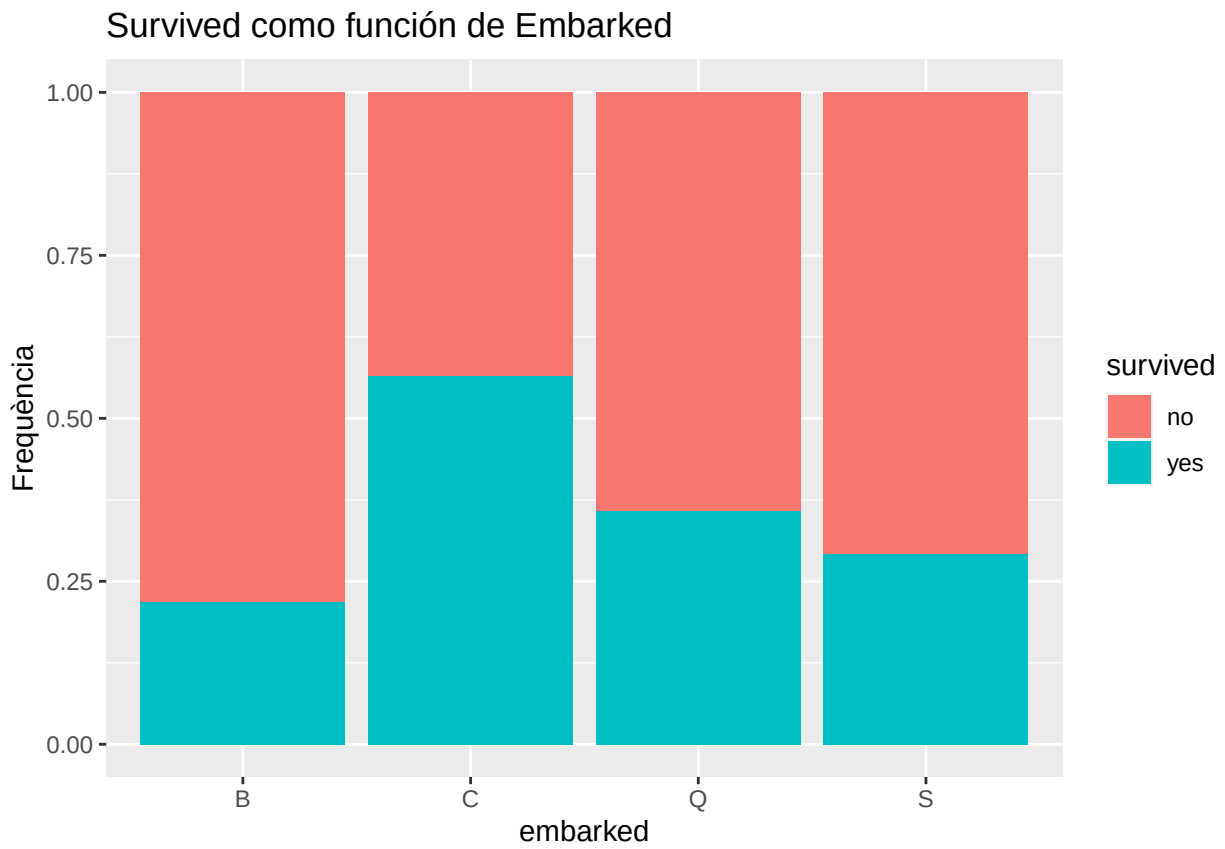
Visualizamos la relación entre las variables “gender” y “survived”:

```
ggplot(data=totalData[1:filas,],aes(x=gender,fill=survived))+geom_bar()+ggtitle("Relación entre las vari
```



Otro punto de vista. Survived como función de Embarked:

```
ggplot(data=totalData[1:filas,],aes(x=embarked,fill=survived))+geom_bar(position="fill")+ylab("Frequènc
```



En la primera gráfica podemos observar fácilmente la cantidad de mujeres que viajaban respecto hombres y observar los que no sobrevivieron. Numéricamente el número de hombres y mujeres supervivientes es similar.

En la segunda gráfica de forma porcentual observamos los puertos de embarque y los porcentajes de supervivencia en función del puerto. Se podría trabajar el puerto C (Cherburgo) para ver de explicar la diferencia en los datos. Quizás porcentualmente embarcaron más mujeres o niños... ¿O gente de primera clase?

Obtenemos ahora una matriz de porcentajes de frecuencia. Vemos, por ejemplo que la probabilidad de sobrevivir si se embarcó en "C" es de un 56.45%

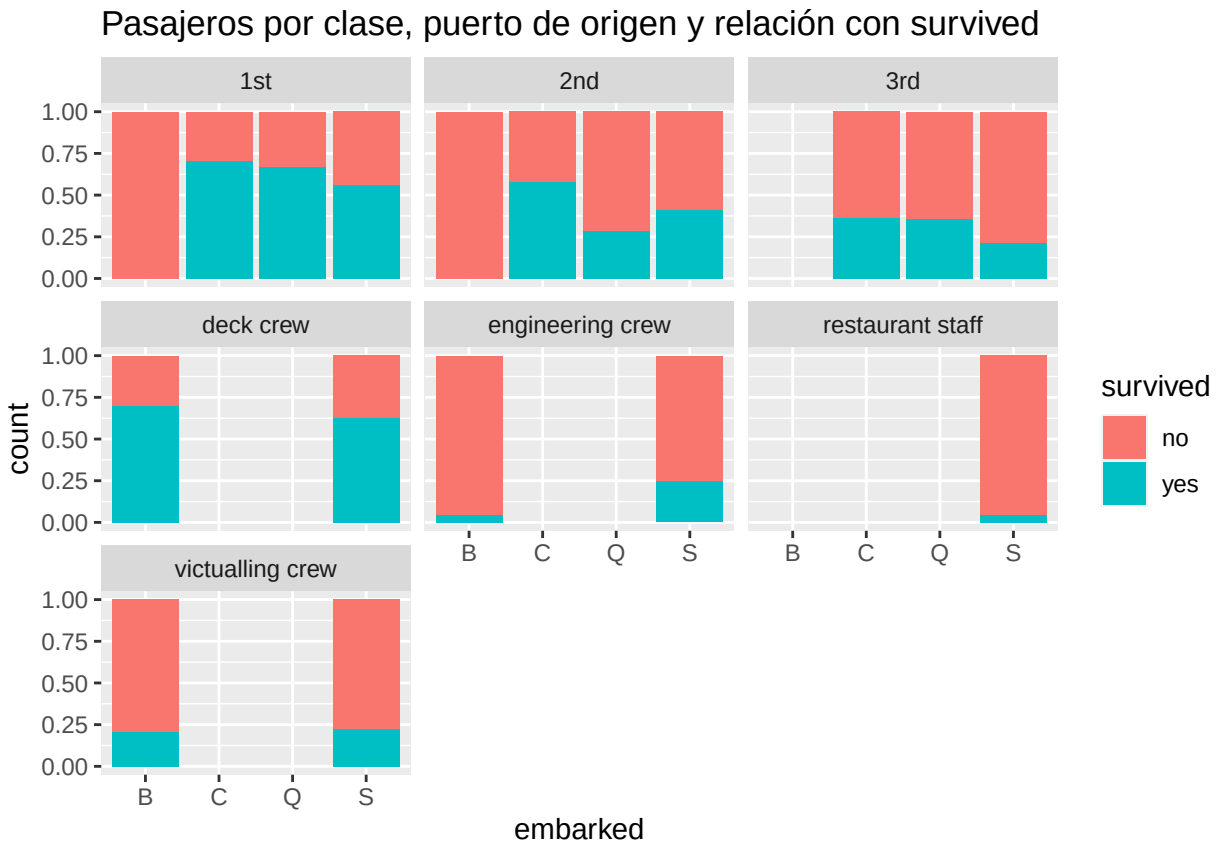
```
t<-table(totalData[1:filas,]$embarked,totalData[1:filas,]$survived)
for (i in 1:dim(t)[1]){
    t[i,]<-t[i,]/sum(t[i,])*100
}
t
```

```
##
##           no      yes
##   B 78.17259 21.82741
##   C 43.54244 56.45756
##   Q 64.22764 35.77236
##   S 70.85396 29.14604
```

Veamos ahora como en un mismo gráfico de frecuencias podemos trabajar con 3 variables: Embarked, Survived y class.

Mostramos el gráfico de embarcados por Pclass:

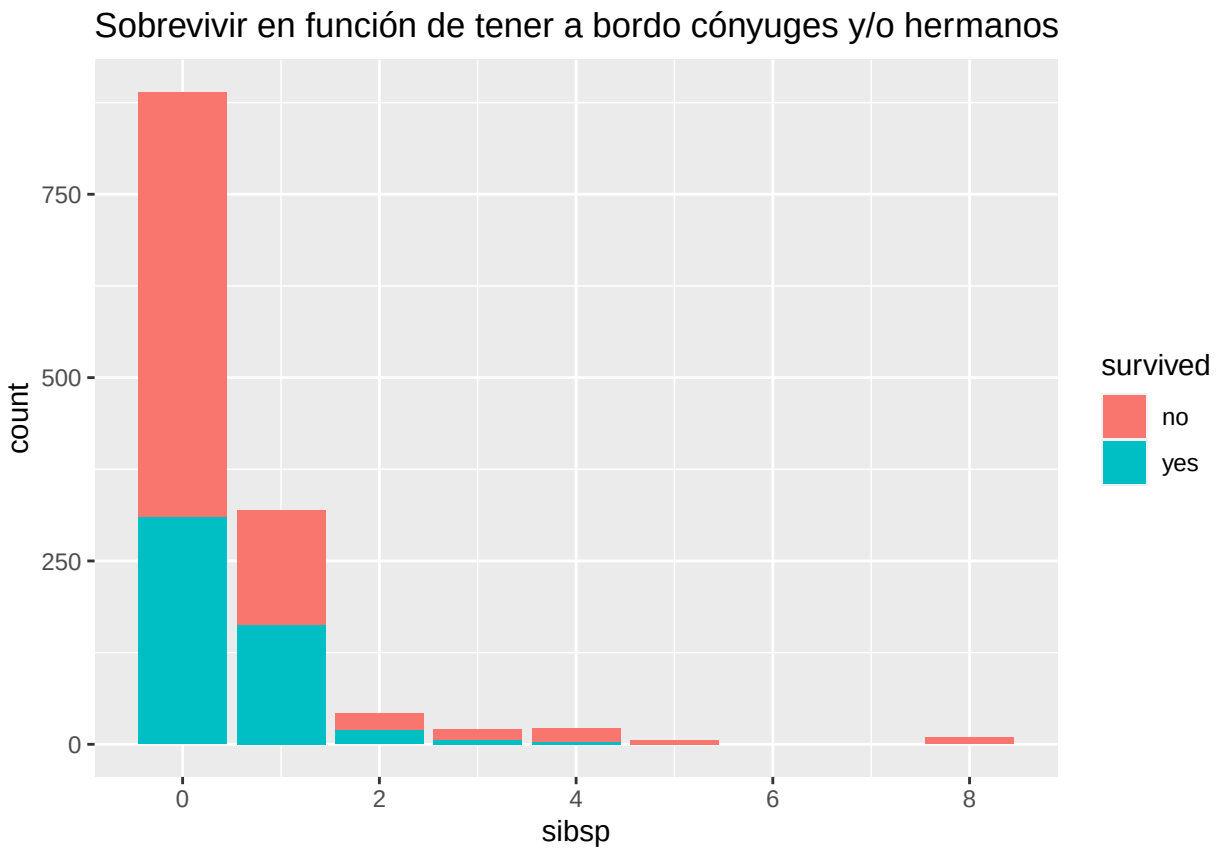
```
ggplot(data = totalData[1:filas,],aes(x=embarked,fill=survived))+geom_bar(position="fill")+facet_wrap(~
```



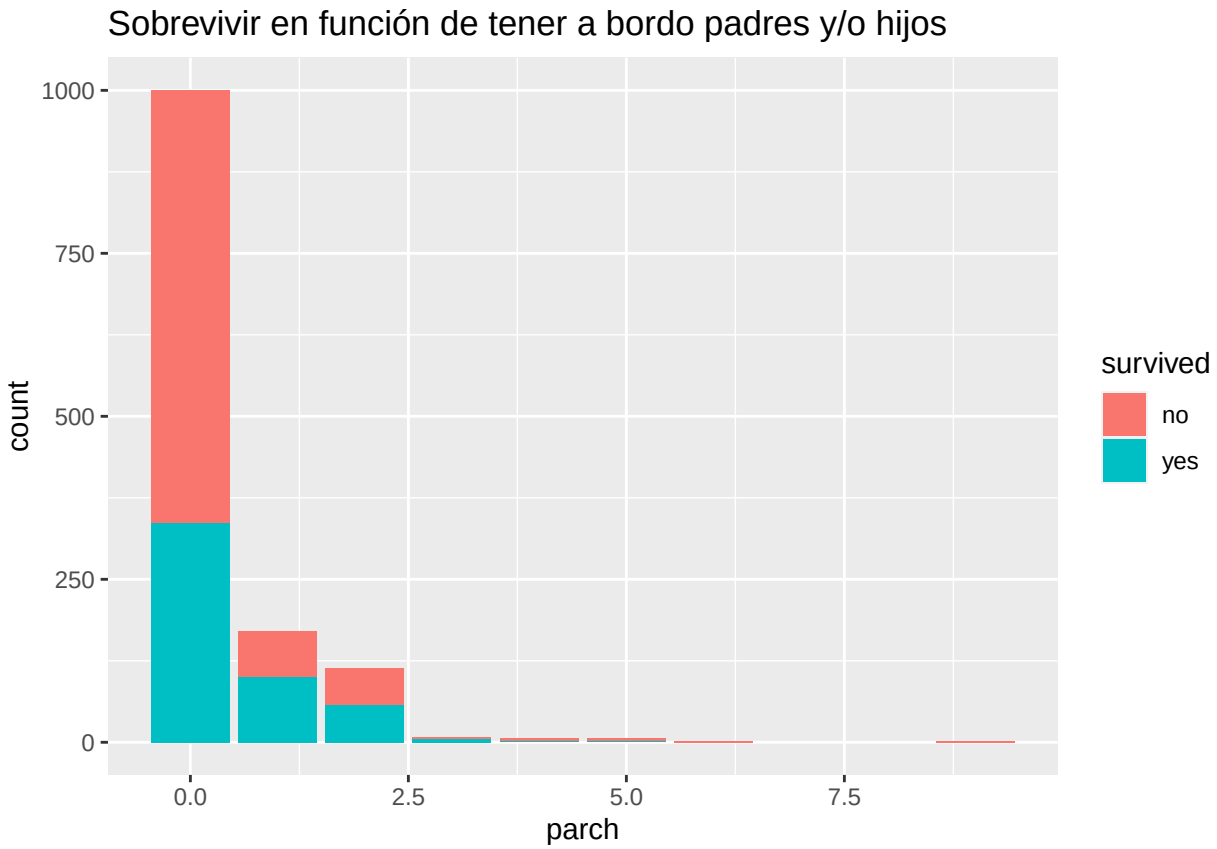
Aquí ya podemos extraer mucha información. Como propuesta de mejora se podría hacer un gráfico similar trabajando solo la clase. Habría que unificar toda la tripulación a una única categoría.

Comparamos ahora dos gráficos de frecuencias: Survived-SibSp y Survived-Parch

```
ggplot(data = totalData[1:filas,],aes(x=sibsp,fill=survived))+geom_bar()+ggtitle("Sobrevivir en función
```



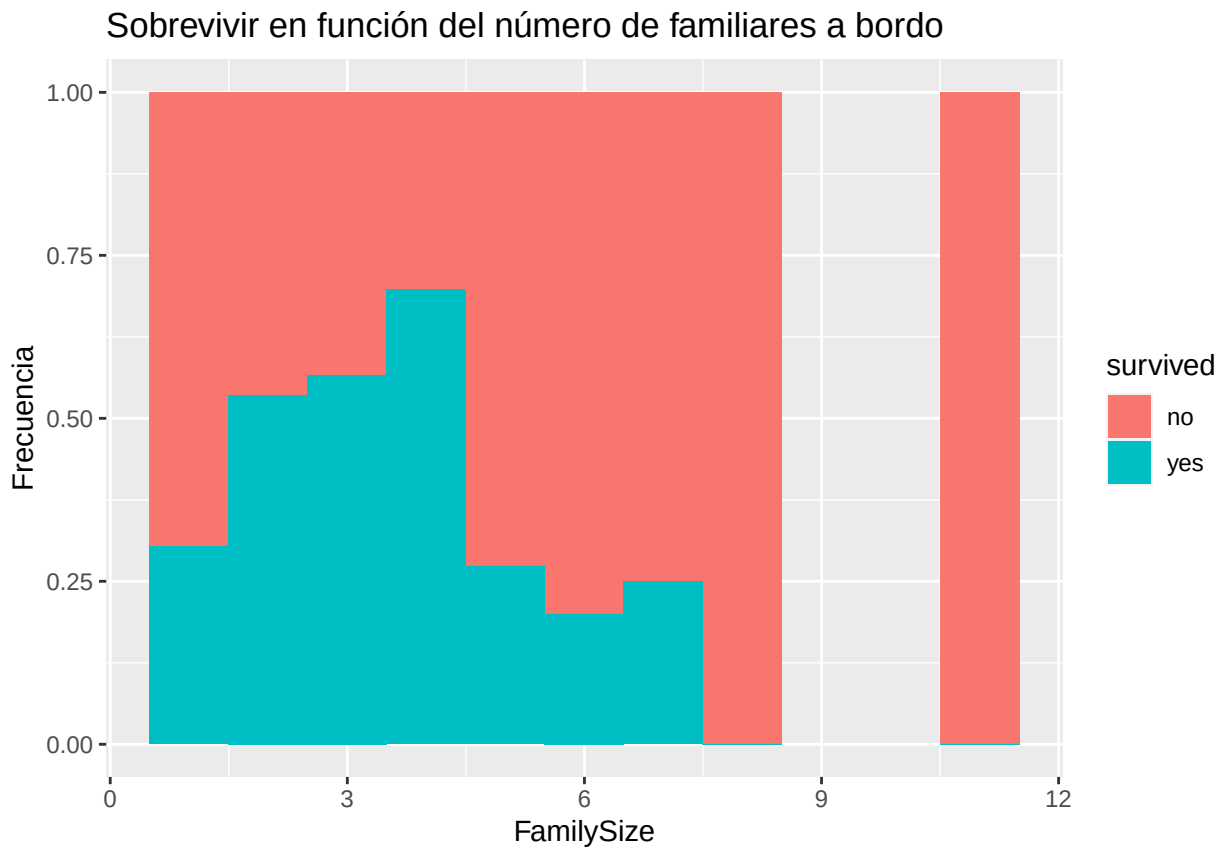

```
ggplot(data = totalData[1:filas,],aes(x=parch,fill=survived))+geom_bar()+ggtitle("Sobrevivir en función
```



Vemos como la forma de estos dos gráficos es similar. Este hecho nos puede indicar presencia de correlaciones altas. Hecho previsible en función de la descripción de las variables.

Veamos un ejemplo de construcción de una variable nueva: Tamaño de familia

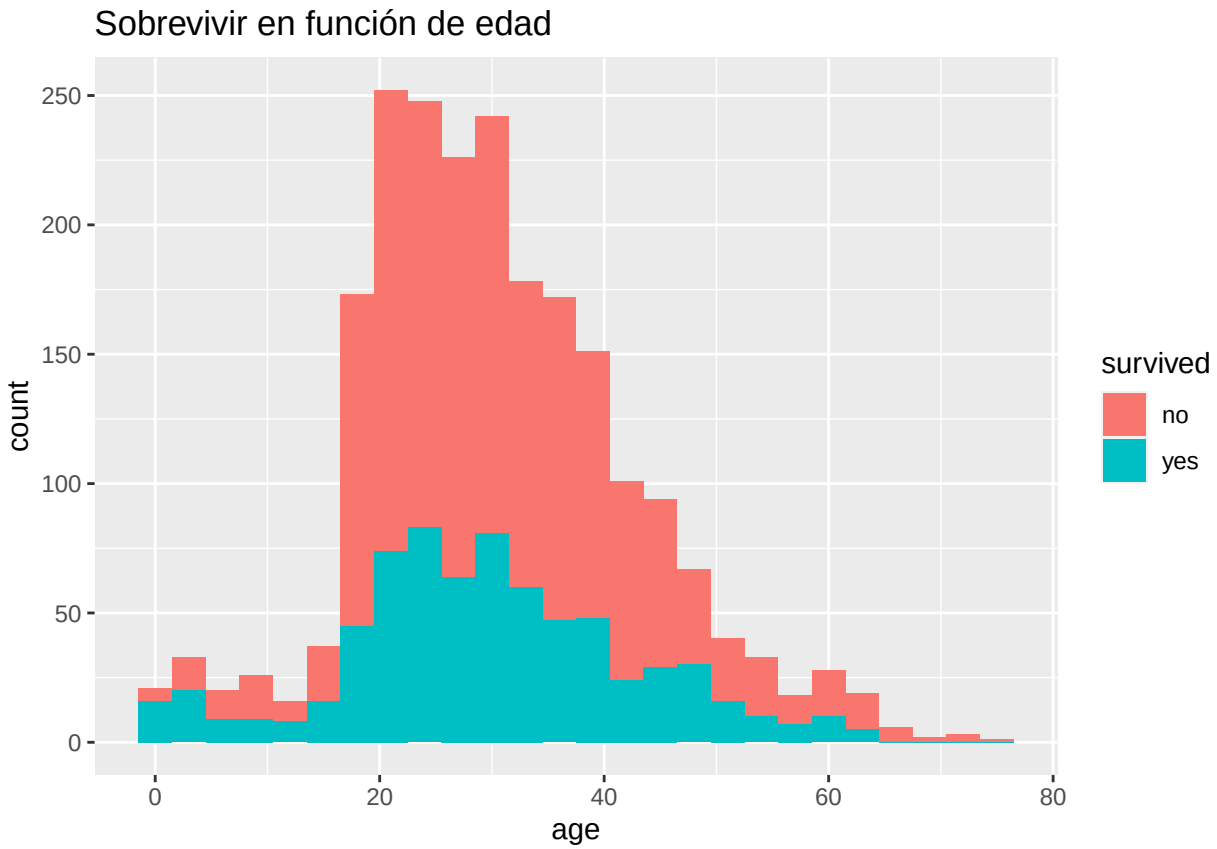
```
totalData$FamilySize <- totalData$sibsp + totalData$parch +1;
totalData1<-totalData[1:filas,]
ggplot(data = totalData1[!is.na(totalData[1:filas,]$FamilySize),],aes(x=FamilySize,fill=survived))+geom
```



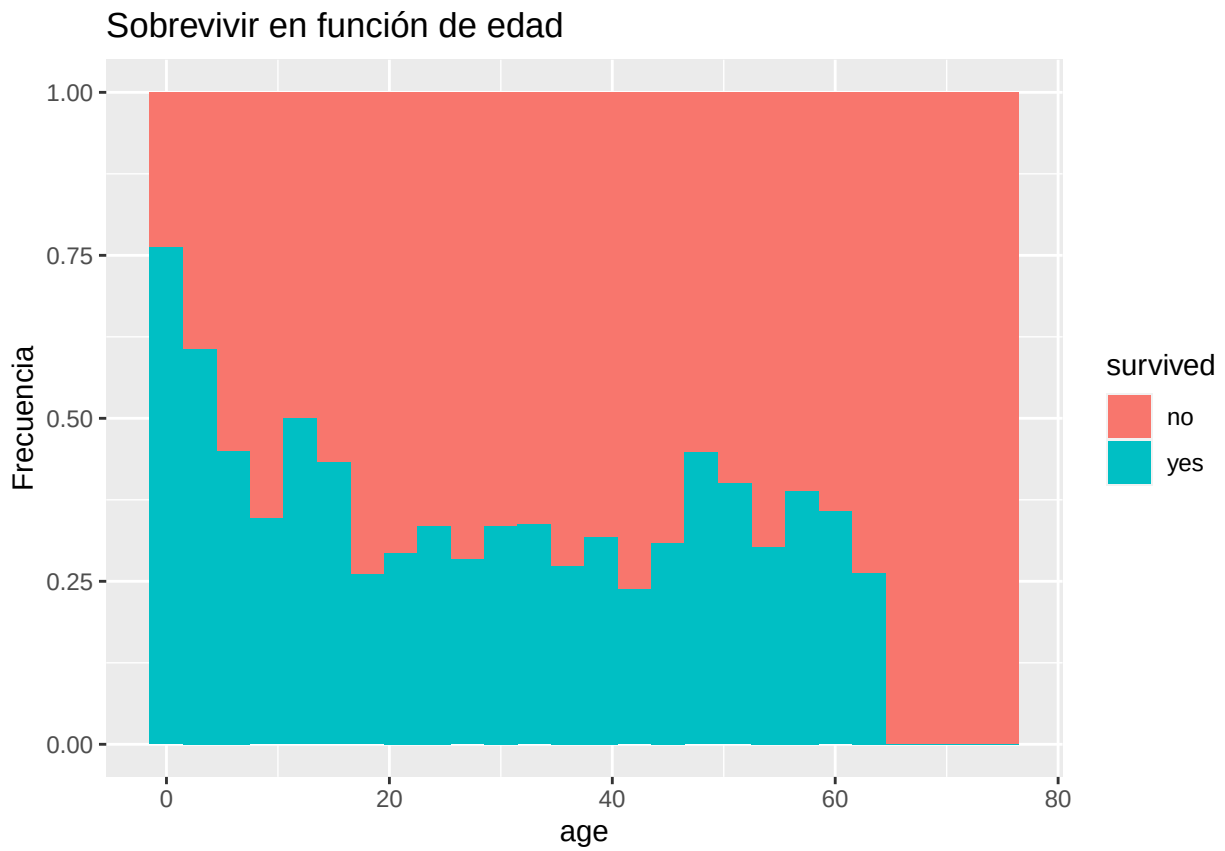
Se confirma el hecho de que los pasajeros viajaban mayoritariamente en familia. No podemos afirmar que el tamaño de la familia tuviera nada que ver con la posibilidad de sobrevivir pues nos tememos que estadísticamente el hecho de haber más familias de alrededor de cuatro miembros debería de ser habitual. Es un punto de partida para investigar más.

Veamos ahora dos gráficos que nos comparan los atributos Age y Survived. Observamos como el parámetro position="fill" nos da la proporción acumulada de un atributo dentro de otro

```
ggplot(data = totalData1[!(is.na(totalData[1:filas,]$age)),],aes(x=age,fill=survived))+geom_histogram(b
```




```
ggplot(data = totalData1[!is.na(totalData[1:filas,]$age),],aes(x=age,fill=survived))+geom_histogram(bin
```



Observamos como el parámetro position="hijo" nos da la proporción acumulada de un atributo dentro de otro. Parece que los niños tuvieron más posibilidad de salvarse.

Vamos a probar si hay una correlación entre la edad del pasajero y el que pagó por el viaje

```
# https://cran.r-project.org/web/packages/tidyverse/index.html
if (!require('tidyverse')) install.packages('tidyverse'); library('tidyverse')
```

```
## Loading required package: tidyverse

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.4     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand()  masks Matrix::expand()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x tidyr::pack()    masks Matrix::pack()
## x arules::recode() masks dplyr::recode()
## x tidyr::unpack()  masks Matrix::unpack()
```

```r
cor.test(x = totalData$age, y = totalData$fare, method = "pearson")
```

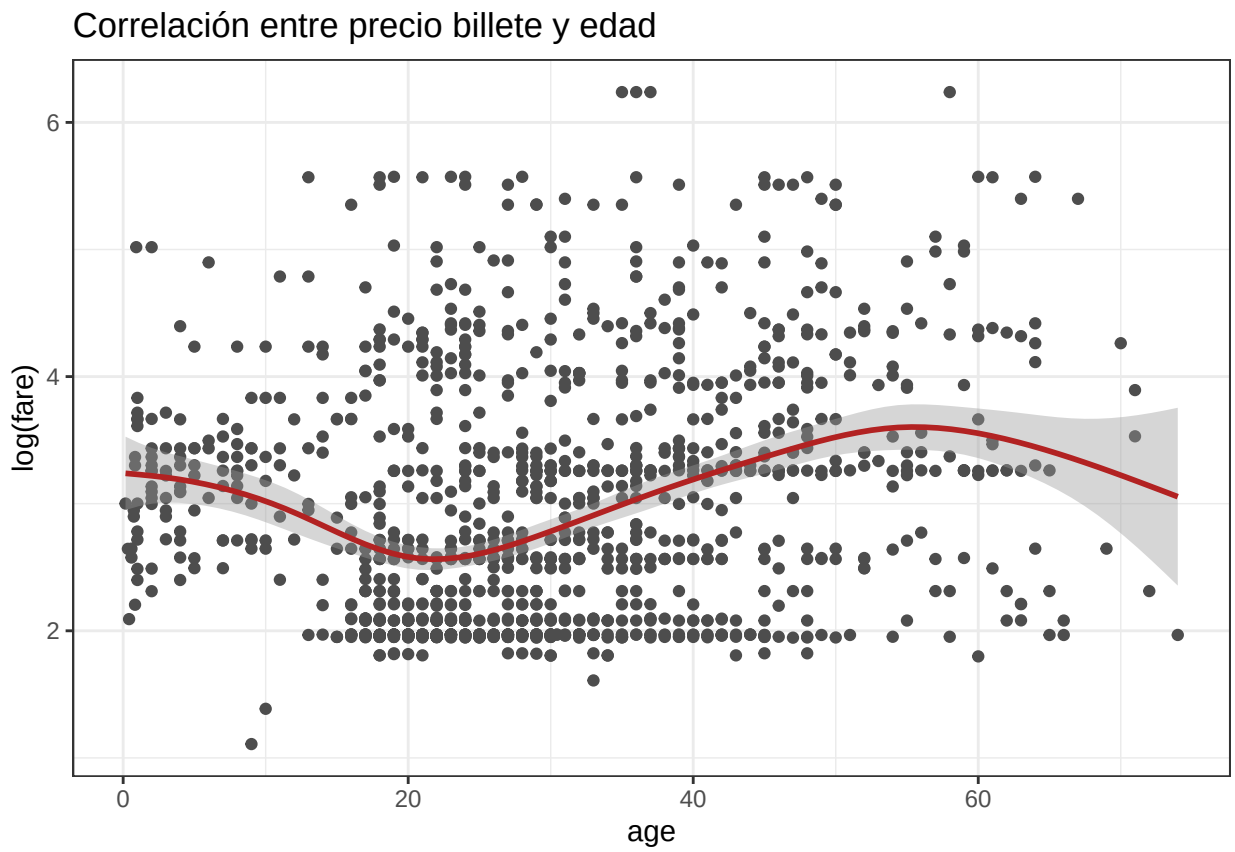
```
##
##  Pearson's product-moment correlation
##
## data:  totalData$age and totalData$fare
## t = 6.7199, df = 1289, p-value = 2.722e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1307297 0.2361631
## sample estimates:
##       cor
## 0.1839756
```

```r
ggplot(data = totalData, aes(x = age, y = log(fare))) + geom_point(color = "gray30") + geom_smooth(colo
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 916 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 916 rows containing missing values (geom_point).
```



Cómo podemos observar no parece haber correlación lineal entre la edad del pasajero y el precio del billete. El diagrama de dispersión tampoco apunta a ningún tipo de relación no lineal evidente.

Conclusiones finales

Los datos tienen una calidad correcta y están mayoritariamente bien informados. Disponen de una variable de clase "survived" que los hace aptos para un clasificador. A parte de la mayor supervivencia de mujeres y niños y de pasajeros de primera clase podemos observar la juventud de los pasajeros y la tripulación. Se observa también una gran cantidad de personas que viajaban en familia.

Ejercicios

Ejercicio 1:

Propon un proyecto completo de minería de datos. La organización de la respuesta tiene que coincidir con las fases típicas del ciclo de vida de un proyecto de minería de datos. *No hay que hacer las tareas de cada fase*. Para cada fase indica cuál es el objetivo de la fase y el producto que se obtendrá. Utiliza ejemplos de qué y como podrían ser las tareas. Si hay alguna característica que hace diferente el ciclo de vida de un proyecto de minería respecto a otros proyectos indícalo.

> Escribe aquí la respuesta a la pregunta

Ejercicio 2:

A partir del juego de datos disponible en el siguiente enlace https://www.kaggle.com/rdoume/beerreviews , realiza las tareas previas a la generación de un modelo de minería de datos explicadas en los módulos "El proceso de minería de datos" y "Preprocesado de los datos y gestión de características". Puedes utilizar de referencia el ejemplo del Titánic.

```
# Cargamos el juego de datos
beerData <- read.csv('beer_reviews.csv',stringsAsFactors = FALSE)

#Verificamos la estructura de datos
str(beerData)
```

```
## 'data.frame':    1586614 obs. of  13 variables:
##  $ brewery_id        : int  10325 10325 10325 10325 1075 1075 1075 1075 1075 1075 ...
##  $ brewery_name      : chr  "Vecchio Birraio" "Vecchio Birraio" "Vecchio Birraio" "Vecchio Birraio"
##  $ review_time       : int  1234817823 1235915097 1235916604 1234725145 1293735206 1325524659 131899
##  $ review_overall    : num  1.5 3 3 3 4 3 3.5 3 4 4.5 ...
##  $ review_aroma      : num  2 2.5 2.5 3 4.5 3.5 3.5 2.5 3 3.5 ...
##  $ review_appearance : num  2.5 3 3 3.5 4 3.5 3.5 3.5 3.5 5 ...
##  $ review_profilename: chr  "stcules" "stcules" "stcules" "stcules" ...
##  $ beer_style        : chr  "Hefeweizen" "English Strong Ale" "Foreign / Export Stout" "German Pilser
##  $ review_palate     : num  1.5 3 3 2.5 4 3 4 2 3.5 4 ...
##  $ review_taste      : num  1.5 3 3 3 4.5 3.5 4 3.5 4 4 ...
##  $ beer_name         : chr  "Sausa Weizen" "Red Moon" "Black Horse Black Beer" "Sausa Pils" ...
##  $ beer_abv          : num  5 6.2 6.5 5 7.7 4.7 4.7 4.7 4.7 4.7 ...
##  $ beer_beerid       : int  47986 48213 48215 47969 64883 52159 52159 52159 52159 52159 ...
```

```r
summary(beerData)
```

```
##    brewery_id    brewery_name        review_time        review_overall
##  Min.   :    1   Length:1586614     Min.   :8.407e+08   Min.   :0.000
##  1st Qu.:  143   Class :character   1st Qu.:1.173e+09   1st Qu.:3.500
##  Median :  429   Mode  :character   Median :1.239e+09   Median :4.000
##  Mean   : 3130                      Mean   :1.224e+09   Mean   :3.816
##  3rd Qu.: 2372                      3rd Qu.:1.289e+09   3rd Qu.:4.500
##  Max.   :28003                      Max.   :1.326e+09   Max.   :5.000
##
##   review_aroma   review_appearance review_profilename  beer_style
##  Min.   :1.000   Min.   :0.000     Length:1586614     Length:1586614
##  1st Qu.:3.500   1st Qu.:3.500     Class :character   Class :character
##  Median :4.000   Median :4.000     Mode  :character   Mode  :character
##  Mean   :3.736   Mean   :3.842
##  3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :5.000   Max.   :5.000
##
##  review_palate    review_taste    beer_name            beer_abv
##  Min.   :1.000   Min.   :1.000   Length:1586614     Min.   : 0.01
##  1st Qu.:3.500   1st Qu.:3.500   Class :character   1st Qu.: 5.20
##  Median :4.000   Median :4.000   Mode  :character   Median : 6.50
##  Mean   :3.744   Mean   :3.793                      Mean   : 7.04
##  3rd Qu.:4.000   3rd Qu.:4.500                      3rd Qu.: 8.50
##  Max.   :5.000   Max.   :5.000                      Max.   :57.70
##                                                     NA's   :67785
##   beer_beerid
##  Min.   :    3
##  1st Qu.: 1717
##  Median :13906
##  Mean   :21713
##  3rd Qu.:39441
##  Max.   :77317
##
```

```r
colSums(is.na(beerData))
```

```
##         brewery_id       brewery_name        review_time     review_overall
##                  0                  0                  0                  0
##       review_aroma  review_appearance review_profilename         beer_style
##                  0                  0                  0                  0
##      review_palate       review_taste          beer_name           beer_abv
##                  0                  0                  0              67785
##        beer_beerid
##                  0
```

```r
beerData$beer_abv[is.na(beerData$beer_abv)] <- mean(beerData$beer_abv)
colSums(beerData=="S")
```

```
##         brewery_id       brewery_name        review_time     review_overall
##                  0                  0                  0                  0
##       review_aroma  review_appearance review_profilename         beer_style
```

```
##                0                0                0                0
##     review_palate    review_taste       beer_name        beer_abv
##                0                0                7               NA
##       beer_beerid
##                0
```

```
beerData$beer_name[beerData$beer_name==""] <- "Unknown"
# Redacta aquí el código R para el estudio del juego de datos

levels(factor(beerData$review_overall))
```

```
##  [1] "0"   "1"   "1.5" "2"   "2.5" "3"   "3.5" "4"   "4.5" "5"
```

Escribe aquí la respuesta a la pregunta

Criterios de evaluación

Ejercicio 1

Concepto y peso en la nota final

El objetivo del proyecto está correctamente definido con suficiente concreción y se puede resolver con técnicas de minería de datos. 15%

Las fases del ciclo de vida están bien expresadas. Los ejemplos son clarificadores. Se justifica y argumenta de las decisiones que se han tomado. 20%

Ejercicio 2

Se carga la base de datos, se visualiza su estructura y se explican los hechos básicos de los datos. 5%

Se estudia si existen atributos vacíos o en diferentes escalas que haya que normalizar. Si es el caso se adoptan medidas para tratar estos atributos. Se construye un nueva variable útil a partir de las existentes. Se discretiza algún atributo. 20%

Se analizan los datos de forma visual y extraen conclusiones tangibles. Hay que elaborar un discurso coherente y con conclusiones claras. 30%

Se trata en profundidad alguno otro aspecto respecto a los datos presentado en los módulos “Preprocesado de los datos y gestión de características” 10%