Estudios de Informática, Multimedia y Telecomunicaciones

Tipología y ciclo de vida de los datos

https://github.com/MMontalvoN/LimpiezaAnalisis (https://github.com/MMontalvoN/LimpiezaAnalisis)

Miguel Ángel Montalvo Navidad

Enero 2022

---

1 Resolución

1.1 Descripción del dataset

A continuación, utilizaremos el juego de datos “Titanic.csv” que recoge datos sobre el famoso transatlántico de pasajeros británico.

1.2 Importancia y objetivos de los análisis

- Las actividades que llevaremos a cabo en el desarrollo de la siguiente práctica hace referenica a la limpieza y análisis de los datos para un proyecto de datos. Tiene como objetivo obtener un dominio de los datos para su posterior análisis. 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.

1.3 Limpieza de los datos

- Como paso previo procedemos a instalamos y cargar 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')
```

- Ahora cargaremos el fichero de datos.

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

- Procedemos a guardar los datos filtrados por tripulación “engineering crew” para hacer análisis posteriores.

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

- Con el siguiente comando verificamos la estructura del data set principal.

```
str(totalData)
```

```
## 'data.frame':    2207 obs. of  11 variables:
##  $ name    : chr  "Abbing, Mr. Anthony" "Abbott, Mr. Eugene Joseph" "Abbott, Mr. Ross
more Edward" "Abbott, Mrs. Rhoda Mary 'Rosa'" ...
##  $ 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" ...
```

Podemos observar que tenemos 2207 registros que se corresponden a los viajeros y tripulación del crucero 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.

1.4 Análisis de los datos

- Ahora procedemos a sacar algunas estadísticas básicas y después analizaremos 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
```

- Por ejemplo 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
```

- Para estos casos (81) de país, asignamos valor “Desconocido” para los valores vacíos de la variable “country”.

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

- Para el caso de edad (2), 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.

1.5 Pruebas estadísticas

- 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
```

- Procedemos a discretizar con intervalos.

```
totalData["segmento_edad"] <- cut(totalData$age, breaks = c(0,10,20,30,40,50,60,70,100),
labels = c("0-9", "10-19", "20-29", "30-39","40-49","50-59","60-69","70-79"))
```

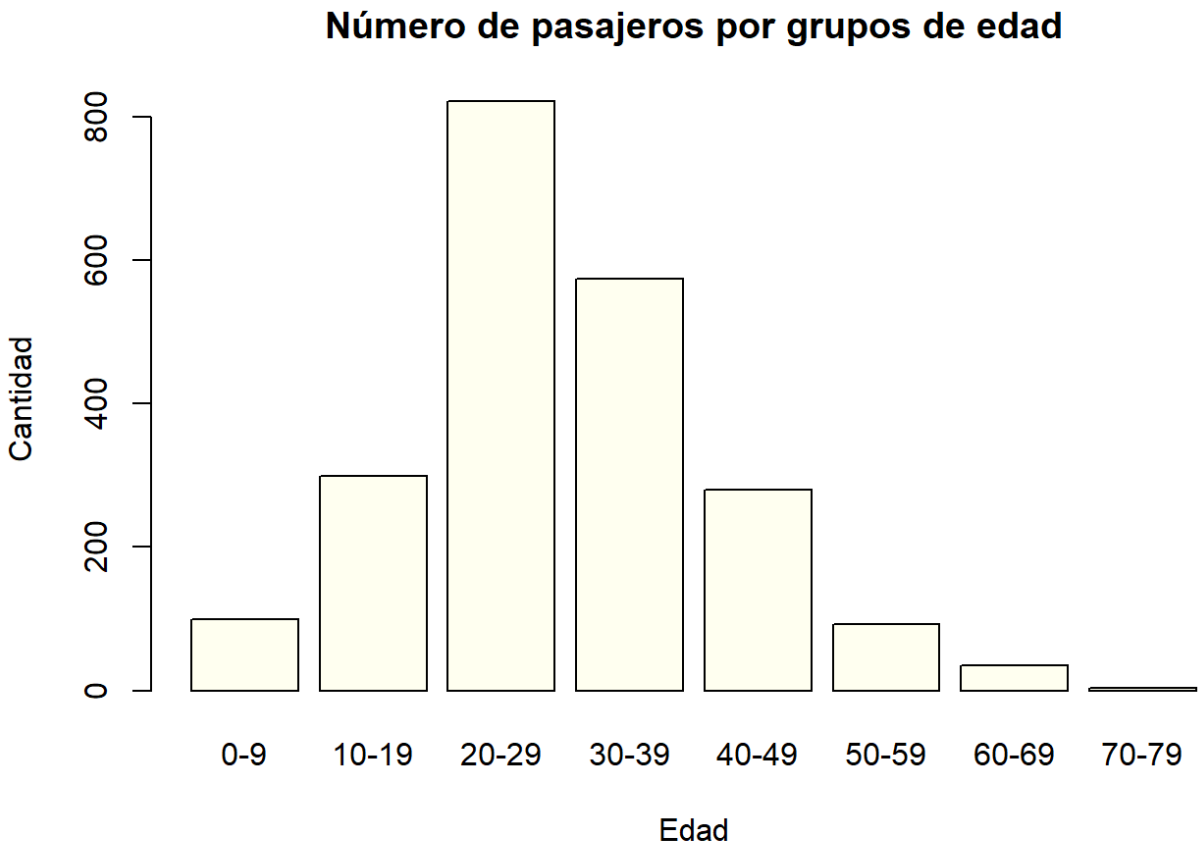
- Y Observamos los datos discretizados.

```
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
```

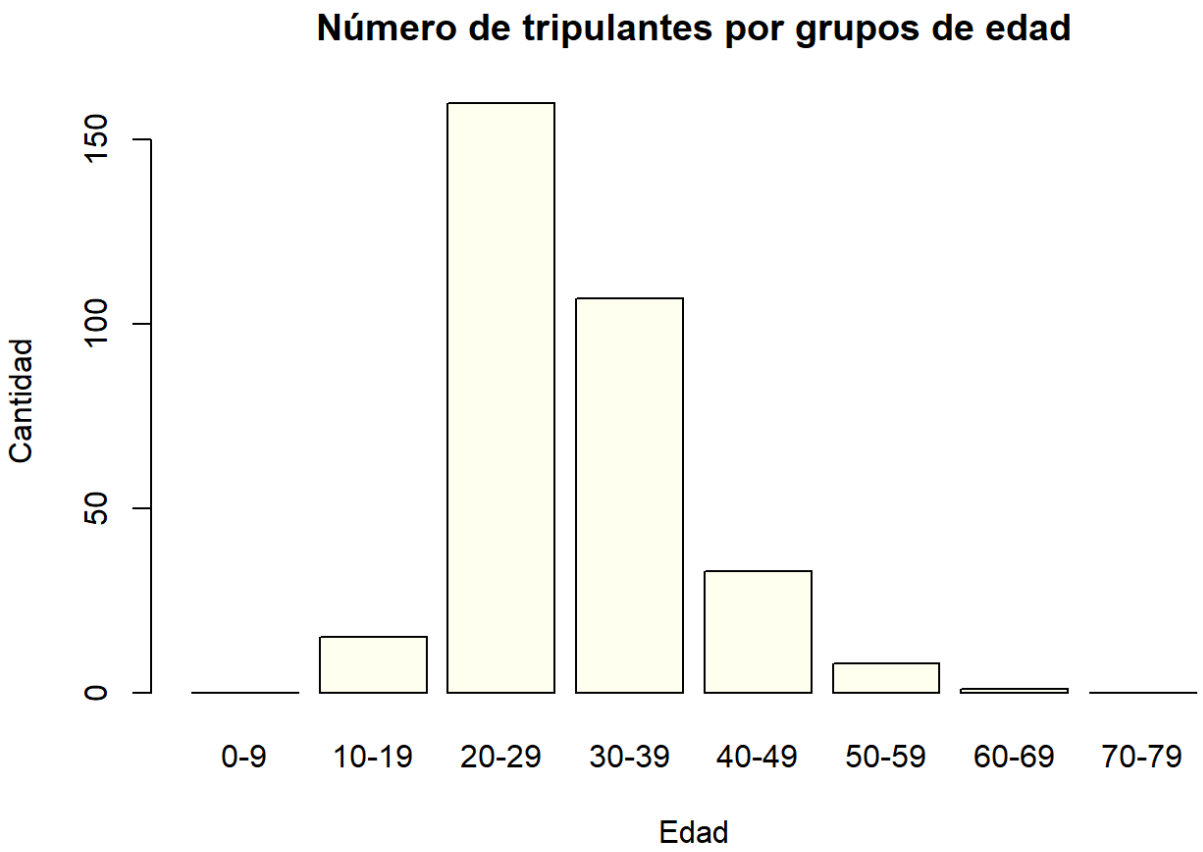
- Ahora podemos ver como se agrupaban por grupos de edad.

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



- Procedemos a repetir los pasos anteriores pero solo sobre el subconjunto de tripulación filtrado antes "engineering crew".

```
totalData_crew["segmento_edad"] <- cut(totalData_crew$age, breaks = c(0,10,20,30,40,50,6
0,70,100), labels = c("0-9", "10-19", "20-29", "30-39","40-49","50-59","60-69","70-79"))
plot(totalData_crew$segmento_edad,main="Número de tripulantes por grupos de edad",xlab=
"Edad", ylab="Cantidad",col = "ivory")
```



De la discretización de la edad observamos que realmente la gente que viajaba era muy joven. El segmento más grande era de 20 a 29 años. También podemos observar la juventud de la tripulación del crucero.

- 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'
```

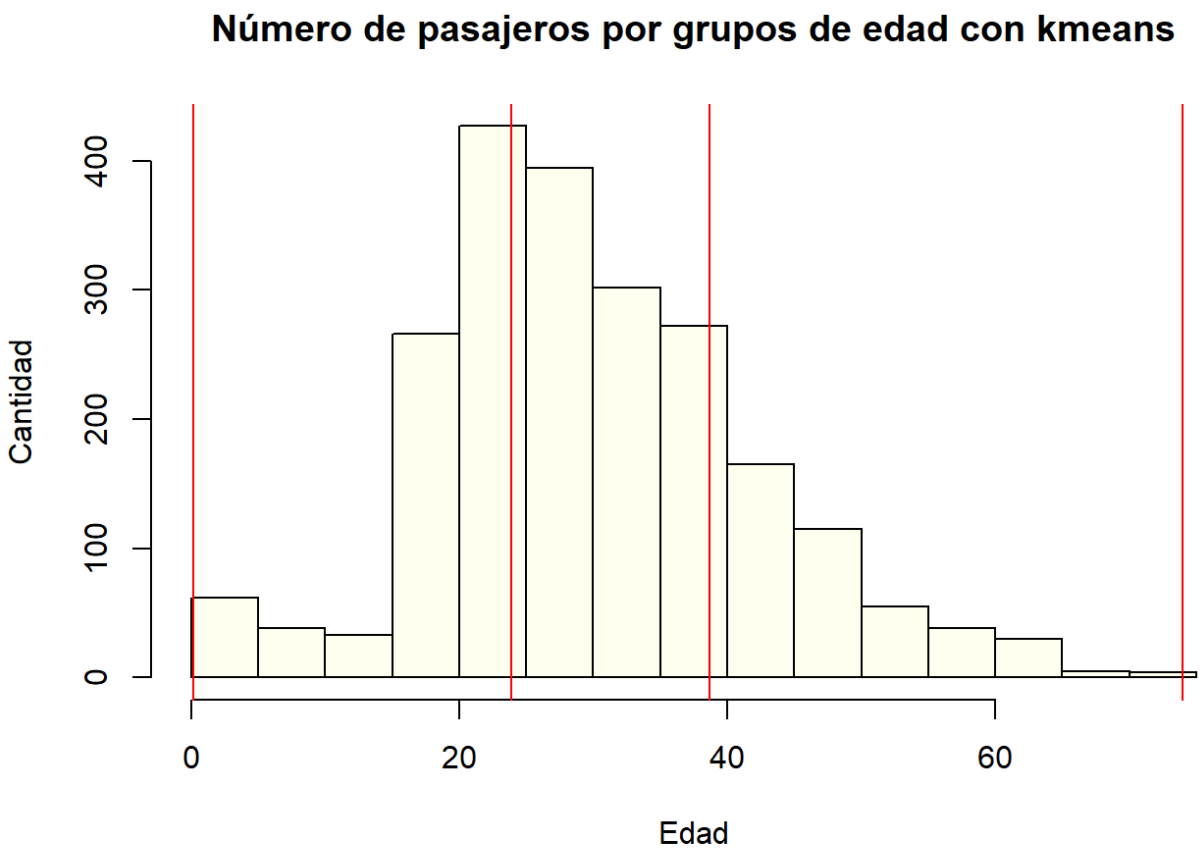
```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## 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$age, main="Número de pasajeros por grupos de edad con kmeans",xlab="Edad"
, ylab="Cantidad",col = "ivory")
abline(v=discretize(totalData$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

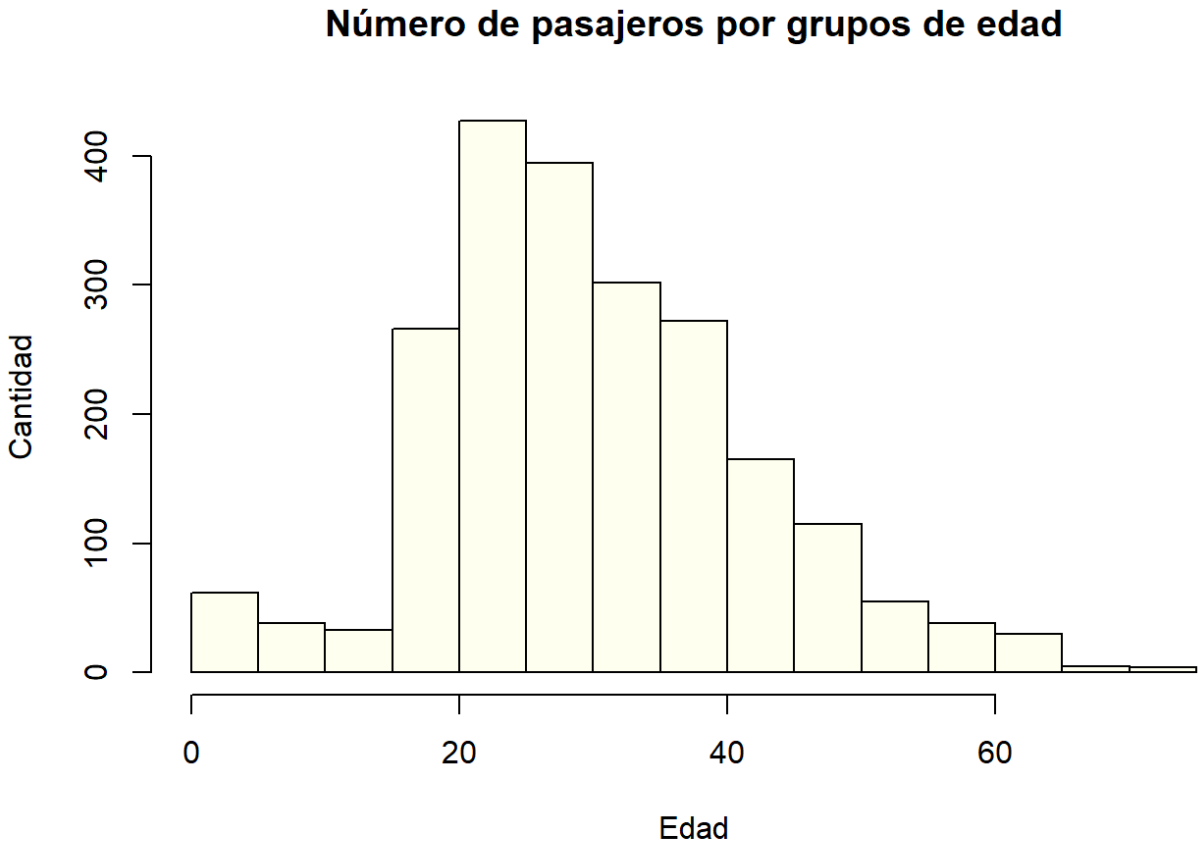
```
totalData$age_ND = (totalData$age-min(totalData$age))/(max(totalData$age)-min(totalData
$age))

max(totalData$age)
```

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

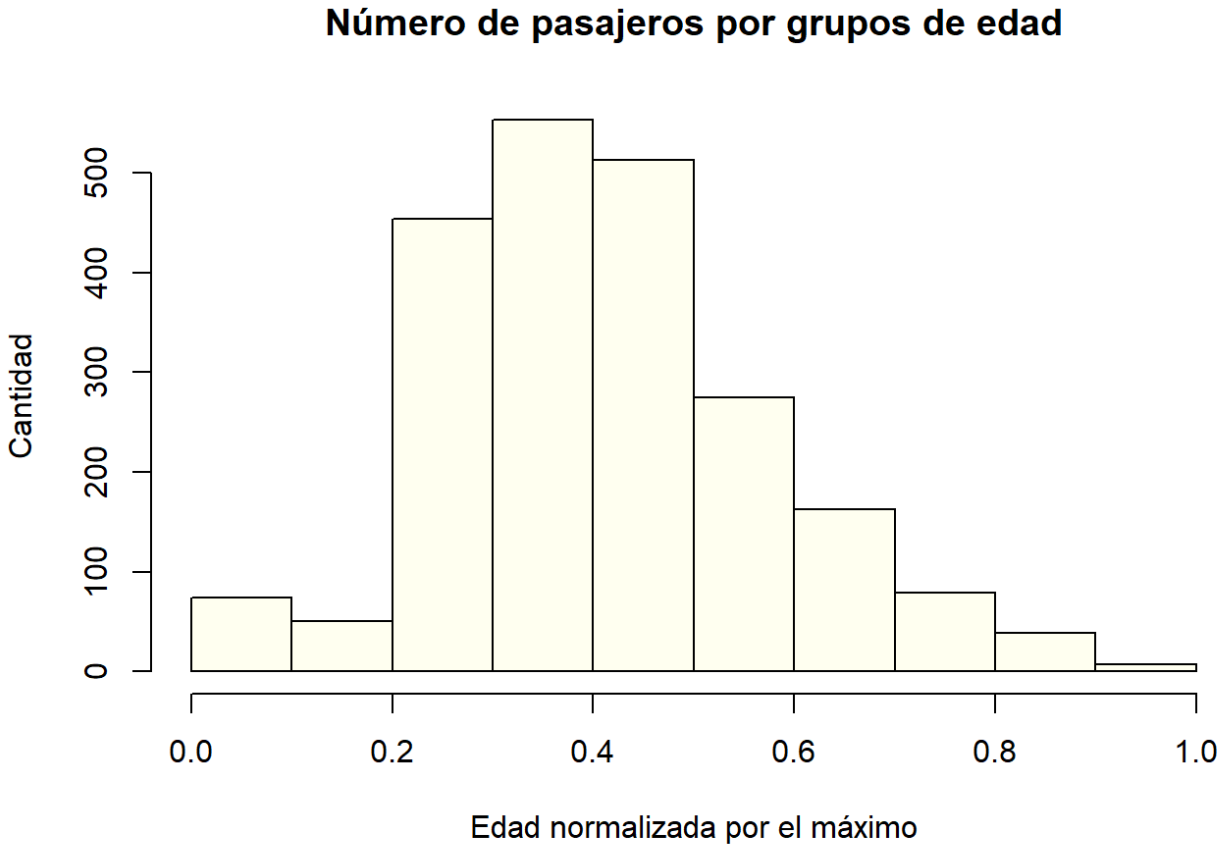
```
min(totalData$age)
```

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

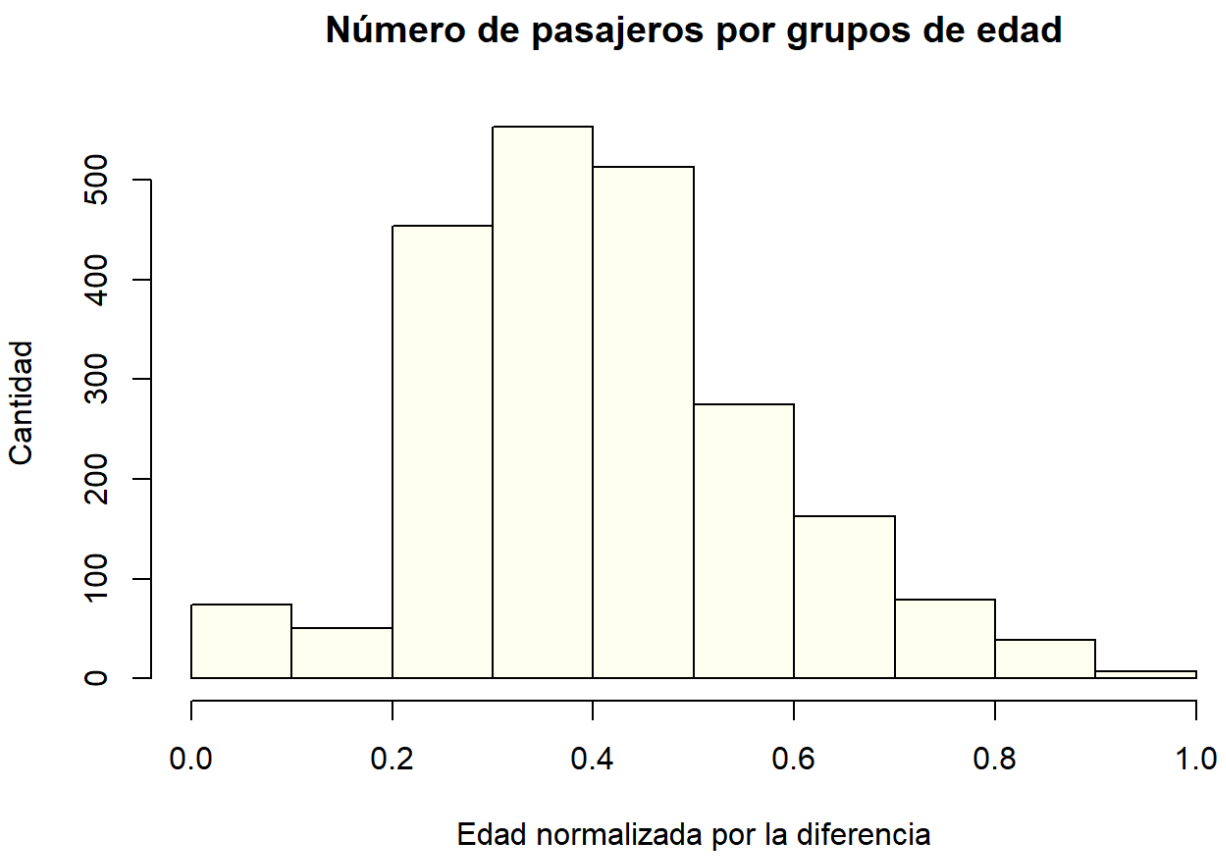
```
hist(totalData$age,xlab="Edad", col="ivory",ylab="Cantidad", main="Número de pasajeros p
or grupos de edad")
```




```
hist(totalData$age_NM,xlab="Edad normalizada por el máximo", ylab="Cantidad",col="ivory"
, main="Número de pasajeros por grupos de edad")
```



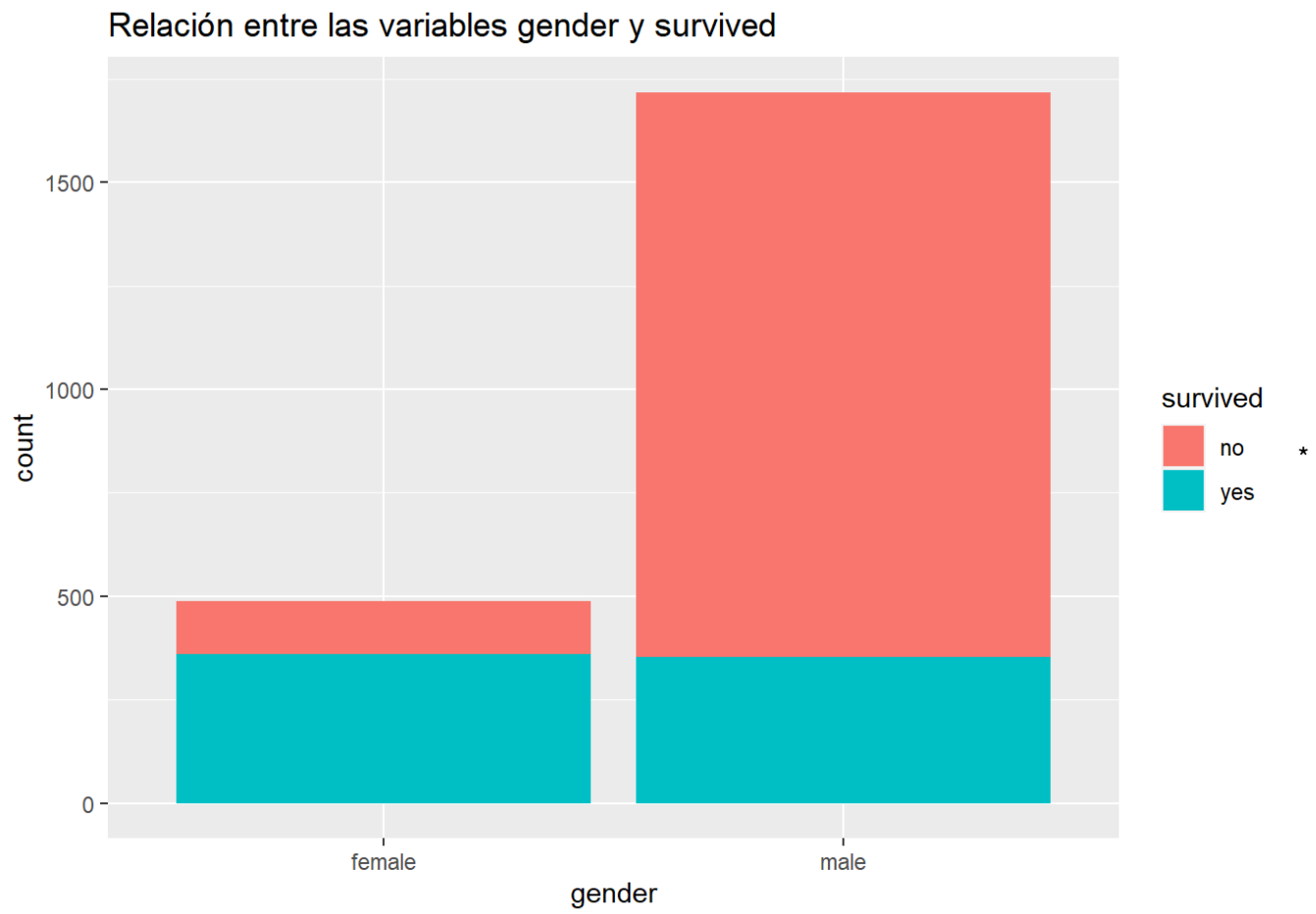
```
hist(totalData$age_ND,xlab="Edad normalizada por la diferencia",ylab="Cantidad", col="ivory", main="Número de pasajeros por grupos de edad")
```



1.6 Procesos de análisis visuales del juego de datos

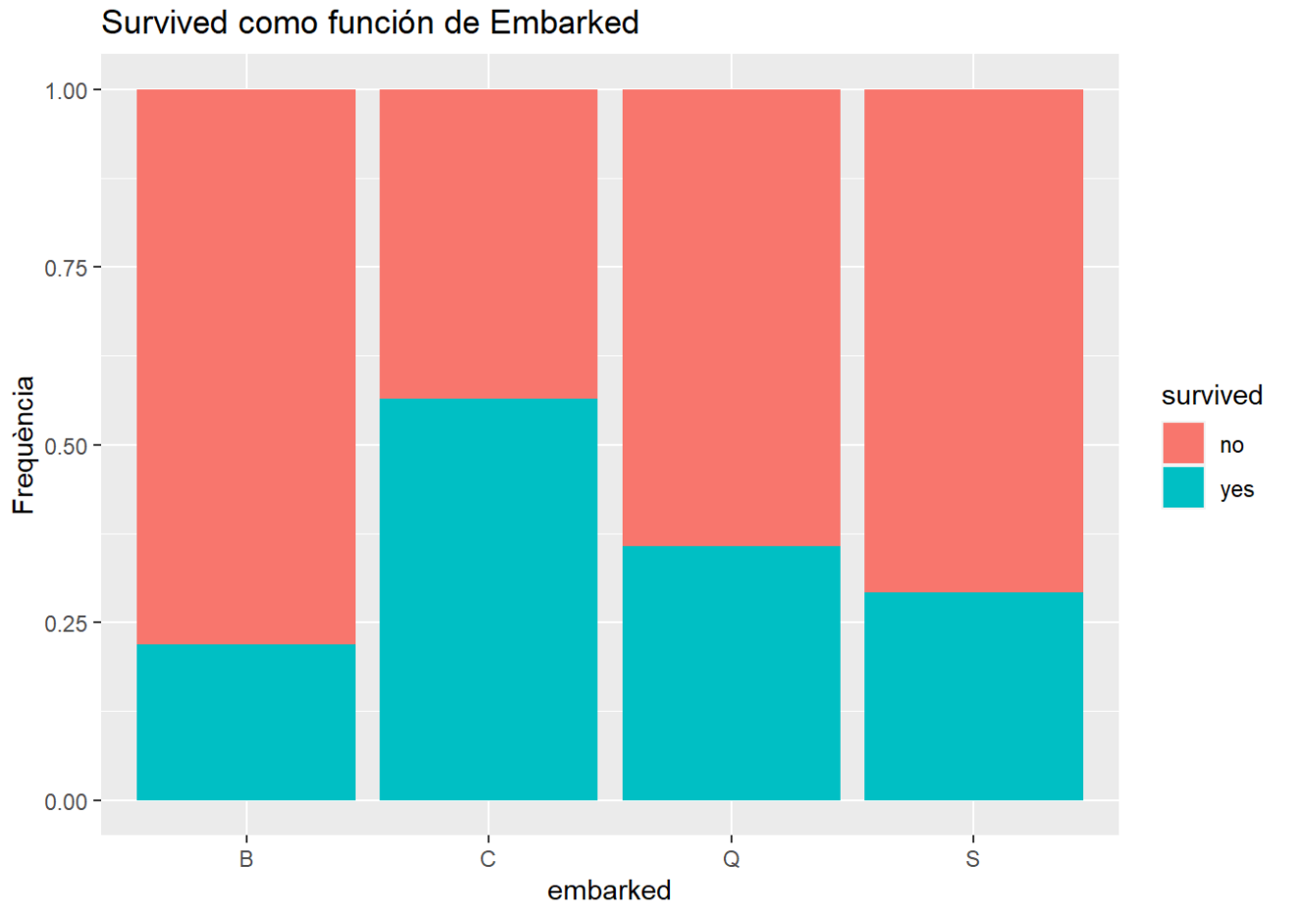
- Nos proponemos analizar las relaciones entre las diferentes variables del juego de datos para ver si se relacionan y como. Visualizamos la relación entre las variables “gender” y “survived”:

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



Otro punto de vista. Survived como función de Embarked:

```
ggplot(data=totalData[1:filas,],aes(x=embarked,fill=survived))+geom_bar(position="fill")
+ylab("Frequència")+ggtitle("Survived como función de Embarked")
```



- 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?

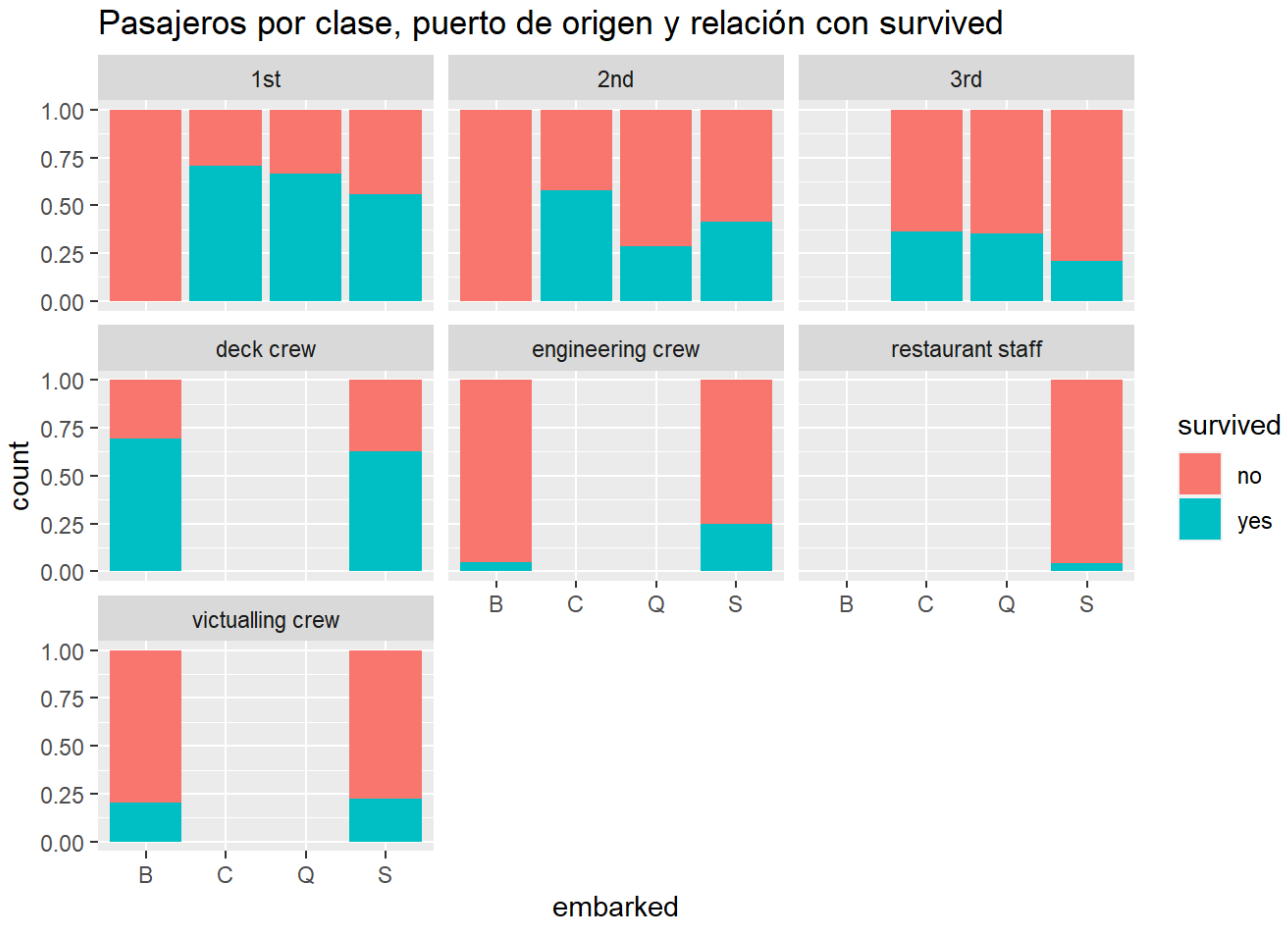
*Obtenemos ahora una matriz de porcentajes de frecuencia. Vemos, por ejemplo que la probabilidad de sobrevivir si se embarcó en “C” es de un 56.45%

```
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
```

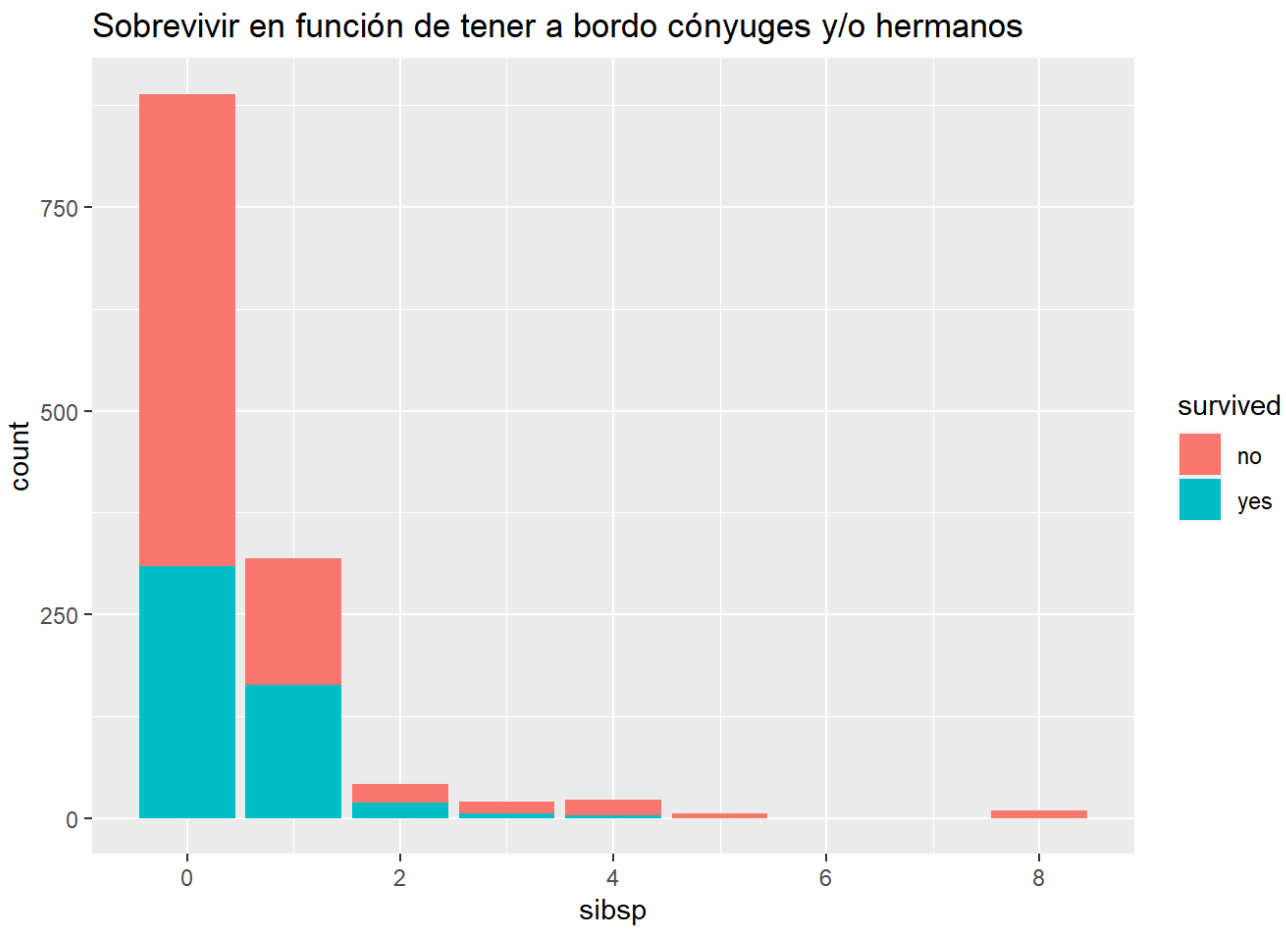
- Veamos ahora como en un mismo gráfico de frecuencias podemos trabajar con 3 variables: Embarked, Survived y class. Mostramos el gráfico de embarcados por class:

```
ggplot(data = totalData[1:filas,],aes(x=embarked,fill=survived))+geom_bar(position="fil
l")+facet_wrap(~class)+ggtitle("Pasajeros por clase, puerto de origen y relación con sur
vived")
```

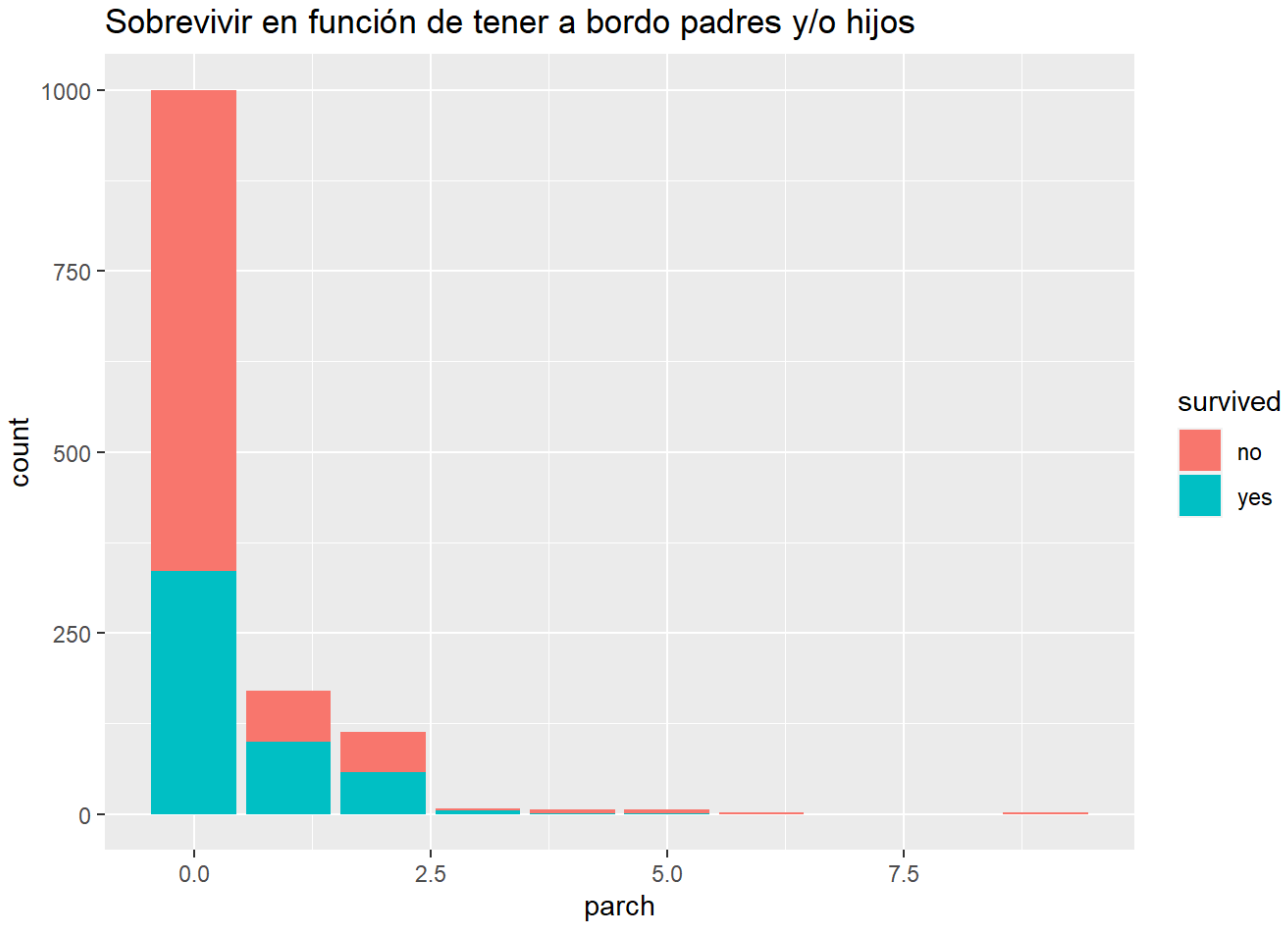


- Aquí ya podemos extraer mucha información. Como propuesta de mejora se podría hacer un gráfico similar trabajando solo la clase. Habría que unificar toda la tripulación a una única categoría.
- Comparamos ahora dos gráficos de frecuencias: Survived-SibSp y Survived-Parch

```
ggplot(data = totalData[1:filas,],aes(x=sibsp,fill=survived))+geom_bar()+ggtitle("Sobrev
ivir en función de tener a bordo cónyuges y/o hermanos")
```

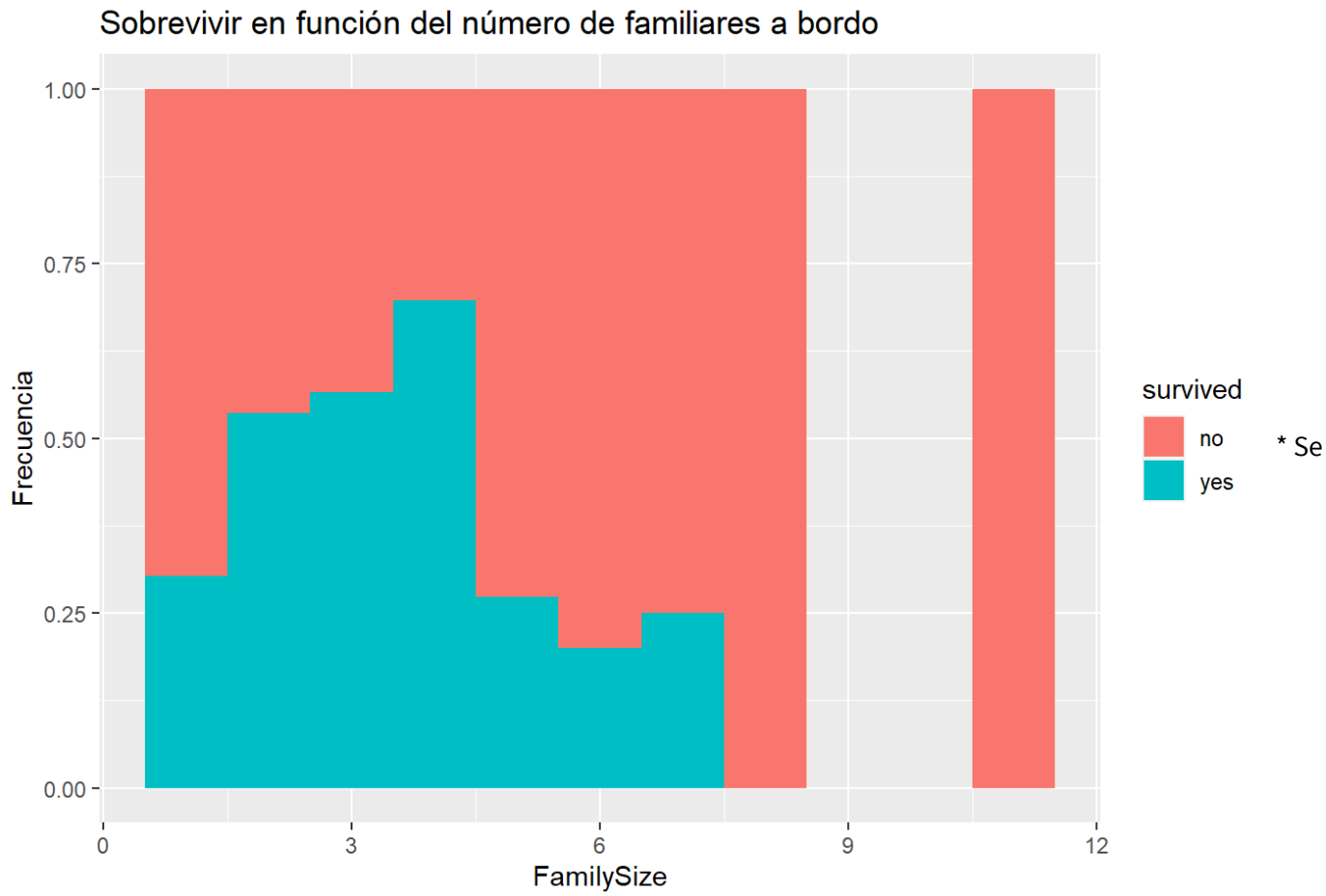



```
ggplot(data = totalData[1:filas,],aes(x=parch,fill=survived))+geom_bar()+ggtitle("Sobrev
ivir en función de tener a bordo padres y/o hijos")
```



- 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.

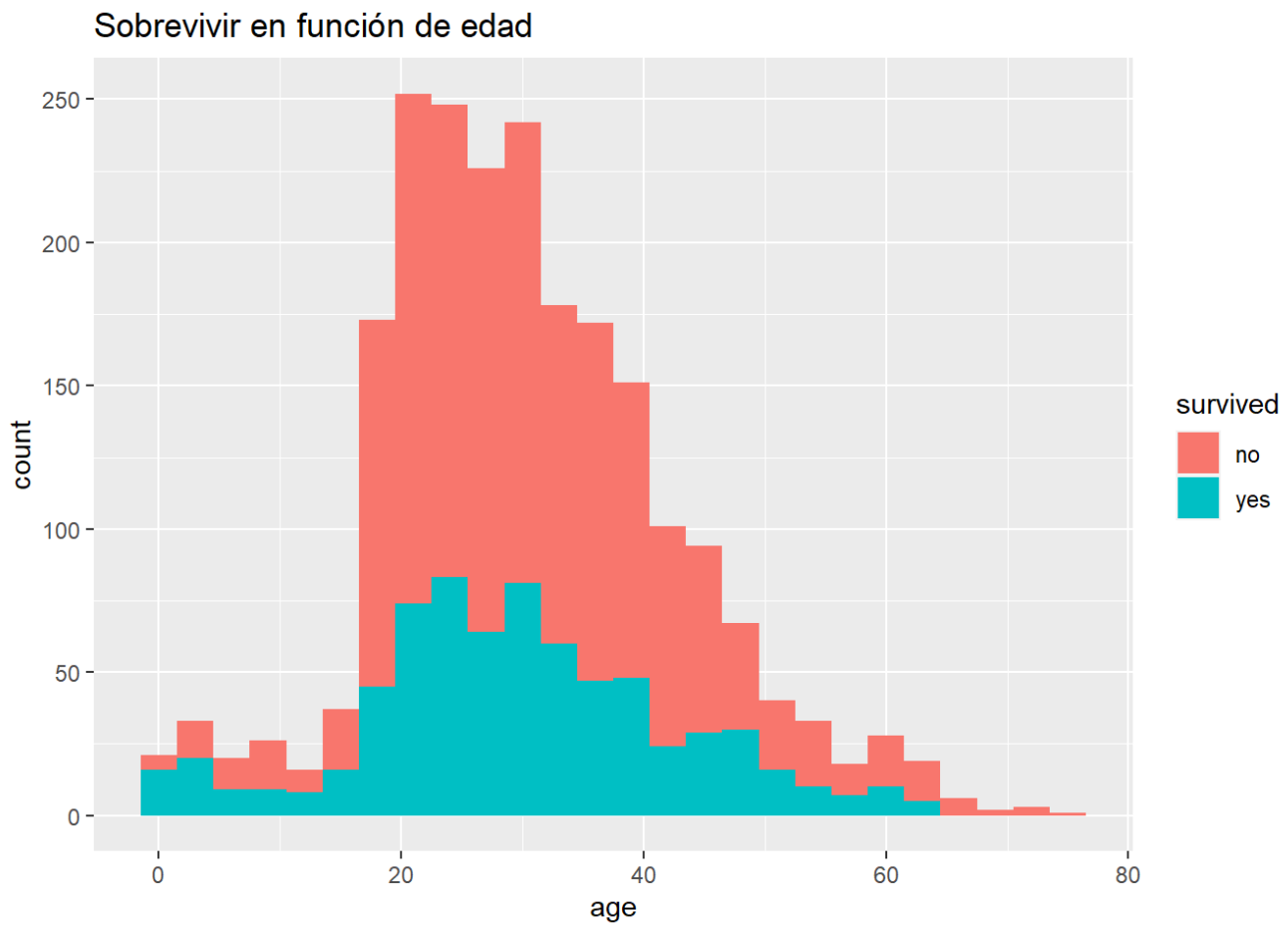
```
totalData$FamilySize <- totalData$sibsp + totalData$parch +1;
totalData1<-totalData[1:filas,]
ggplot(data = totalData1[!is.na(totalData[1:filas,]$FamilySize),],aes(x=FamilySize,fill=
survived))+geom_histogram(binwidth =1,position="fill")+ylab("Frecuencia")+ggtitle("Sobre
vivir en función del número de familiares a bordo")
```



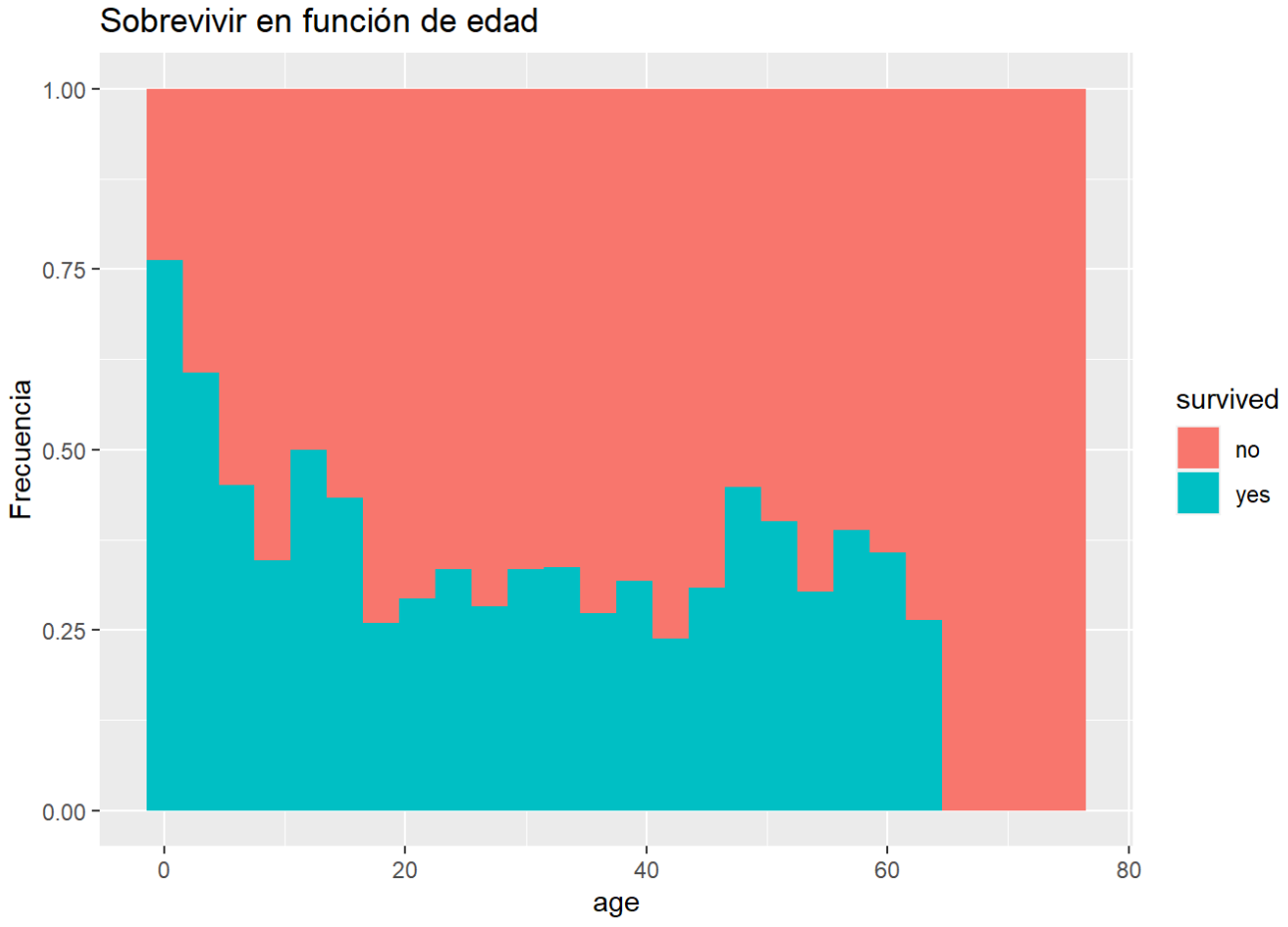
confirma el hecho de que los pasajeros viajaban mayoritariamente en familia. No podemos afirmar que el tamaño de la familia tuviera nada que ver con la posibilidad de sobrevivir pues nos tememos que estadísticamente el hecho de haber más familias de alrededor de cuatro miembros debería de ser habitual. 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))+g
eom_histogram(binwidth =3)+ggtitle("Sobrevivir en función de edad")
```




```
ggplot(data = totalData1[!is.na(totalData[1:filas,]$age),],aes(x=age,fill=survived))+geo
m_histogram(binwidth = 3,position="fill")+ylab("Frecuencia")+ggtitle("Sobrevivir en func
ión de edad")
```



- 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  3.1.5     v purrr   0.3.4
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     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()
```

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

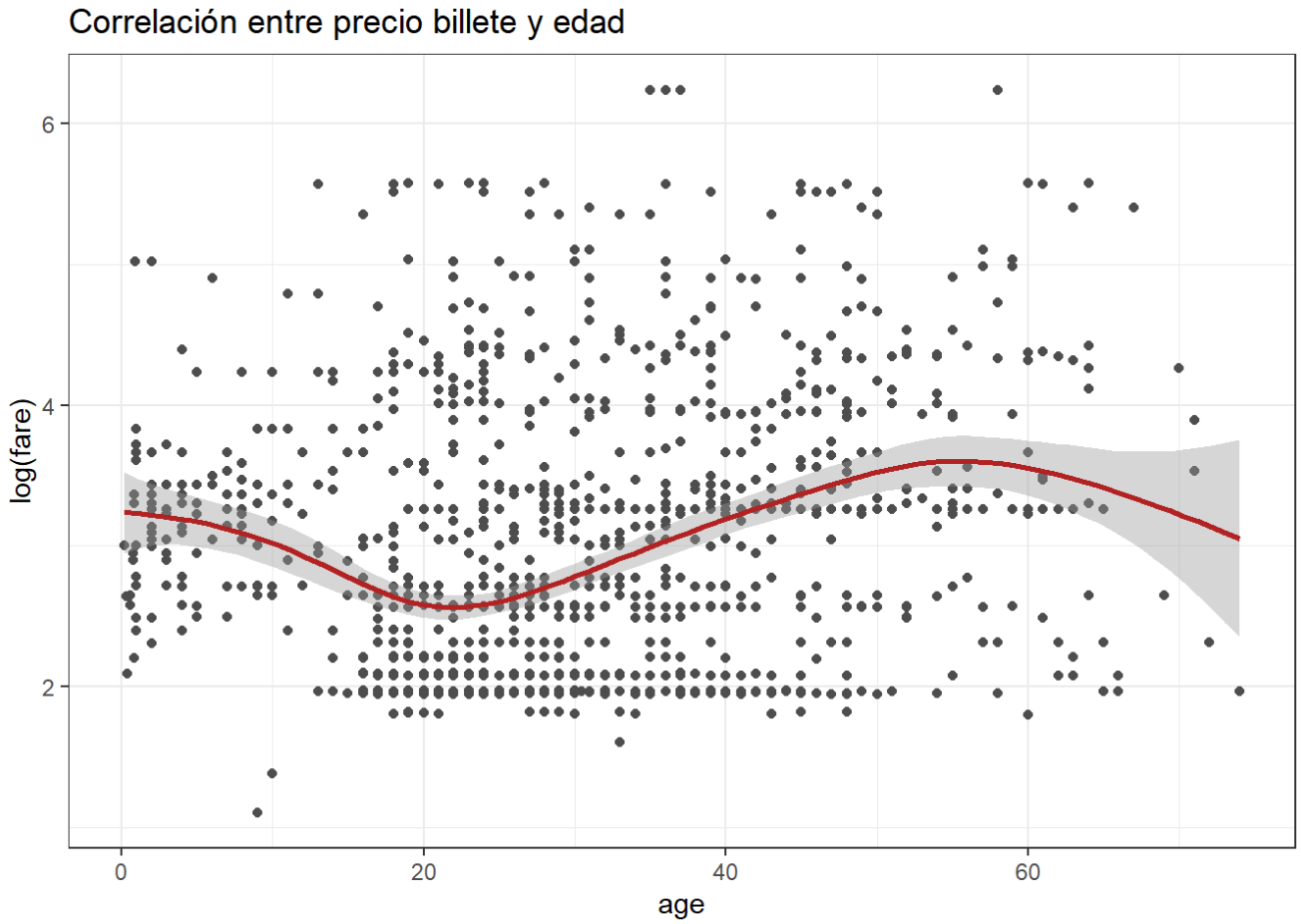
```
##
##  Pearson's product-moment correlation
##
## data:  totalData$age and totalData$fare
## t = 6.7199, df = 1289, p-value = 2.722e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1307297 0.2361631
## sample estimates:
##       cor
## 0.1839756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

```
ggplot(data = totalData, aes(x = age, y = log(fare))) + geom_point(color = "gray30") + g
eom_smooth(color = "firebrick") + theme_bw() +ggtitle("Correlación entre precio billete
 y edad")
```

```
## `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.

1.7 Conclusiones

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.

---

2 Recursos

Los siguientes recursos son de utilidad para la realización de la práctica: * Calvo M., Subirats L., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC. * Megan Squire (2015). Clean Data. Packt Publishing Ltd. * Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann. * Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369. *Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media. Wes McKinney (2012). Python for Data Analysis. O’Reilley Media, Inc.* Tutorial de Github https://guides.github.com/activities/hello-world (https://guides.github.com/activities/hello-world).