

# Estudios de Informática, Multimedia y Telecomunicaciones

## Minería de datos: PEC1

Autor: Eduardo Mora González

octubre 2021

- 1 Introducción
  - 1.1 Presentación
  - 1.2 Objetivos
  - 1.3 Descripción de la PEC a realizar
  - 1.4 Recursos
  - 1.5 Formato y fecha de entrega
  - 1.6 Nota: Propiedad intelectual
- 2 Ejemplo de solución mínimo del ejercicio 2
  - 2.1 Objetivos
  - 2.2 Procesos iniciales con los datos
  - 2.3 Procesos de análisis visuales del juego de datos
  - 2.4 Conclusiones finales
- 3 Ejercicios
  - 3.1 Ejercicio 1:
  - 3.2 Ejercicio 2:
  - 3.3 CARGA Y VERIFICACIÓN DE LOS DATOS
  - 3.4 LIMPIEZA DEL CONJUNTO DE DATOS
  - 3.5 NORMALIZACIÓN Y DISCRETIZACIÓN DE LOS DATOS
  - 3.6 Procesos de análisis del conjunto de datos
  - 3.7 Conclusión
- 4 Criterios de evaluación

---

## 1 Introducción

---

### 1.1 Presentación

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

## 1.2 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.

## 1.3 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 resolvela 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 sustituís los chunks de R por chunks en Python.

## 1.4 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](https://es.wikipedia.org/wiki/cross_industry_standard_process_for_data_mining#Fases_principales)  
([https://es.wikipedia.org/wiki/cross\\_industry\\_standard\\_process\\_for\\_data\\_mining#Fases\\_principales](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.

## 1.5 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.

## 1.6 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.

# 2 Ejemplo de solución mínimo del ejercicio 2

## 2.1 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.

## 2.2 Procesos iniciales con los datos

Primer contacto con el juego de datos.

Instalamos y cargamos las librerías ggplot2 y dplyr.

```
# 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. 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" ...

```

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 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 curioso 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 campo contendrá el valor de la edad discretizada 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", "10-19", "20-29", "30-39","40-49","50-59","60-69","70-79"))
```

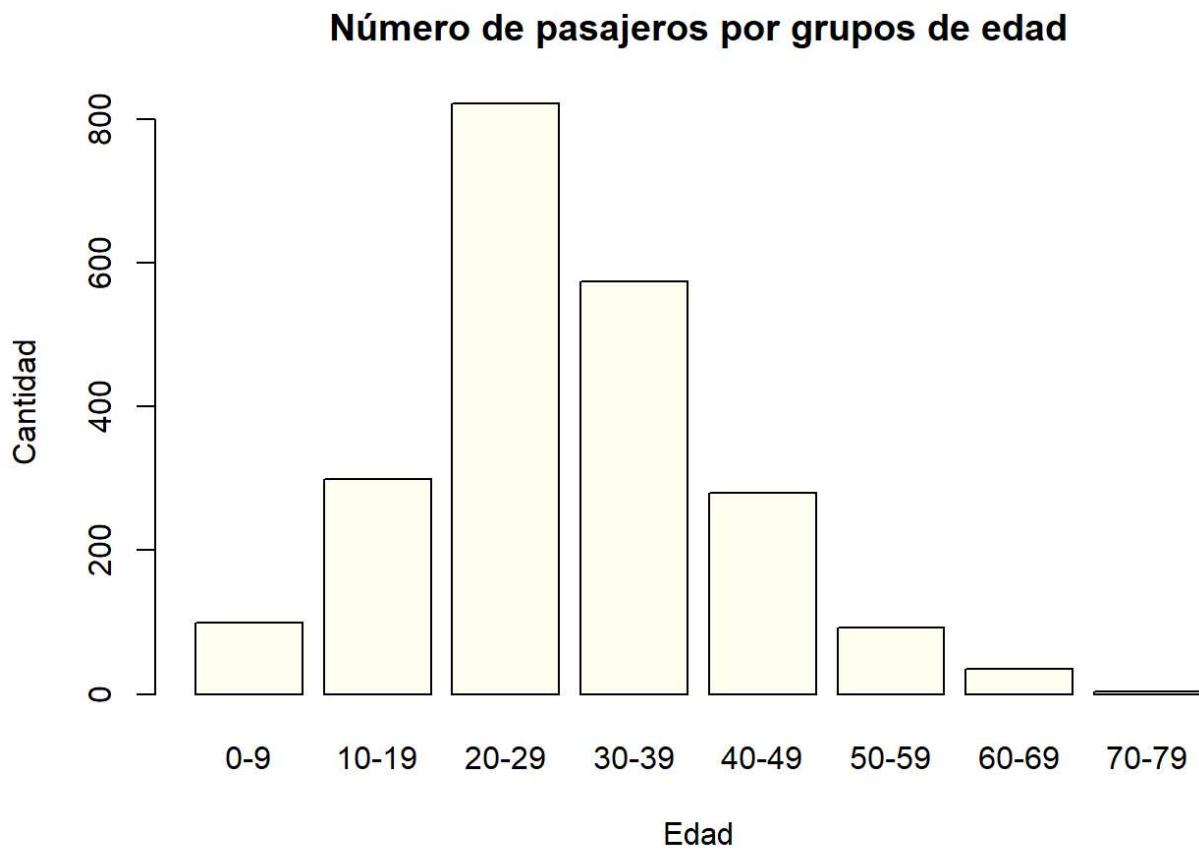
Observamos los datos discretizados.

```
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 JÃrgensen	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.

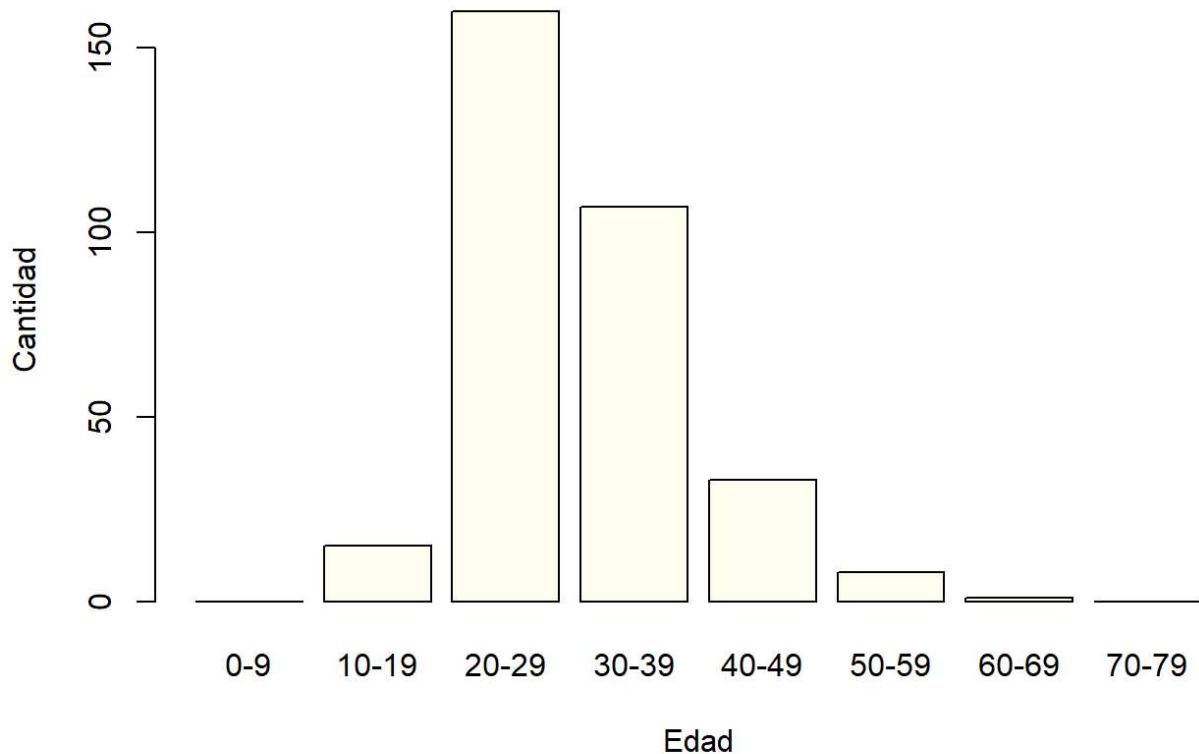
```
plot(totalData$segmento_edad,main="Número de pasajeros por grupos de edad",xlab="Edad",  
ylab="Cantidad",col = "ivory")
```



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

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

## Número de tripulantes por grupos de edad



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 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'

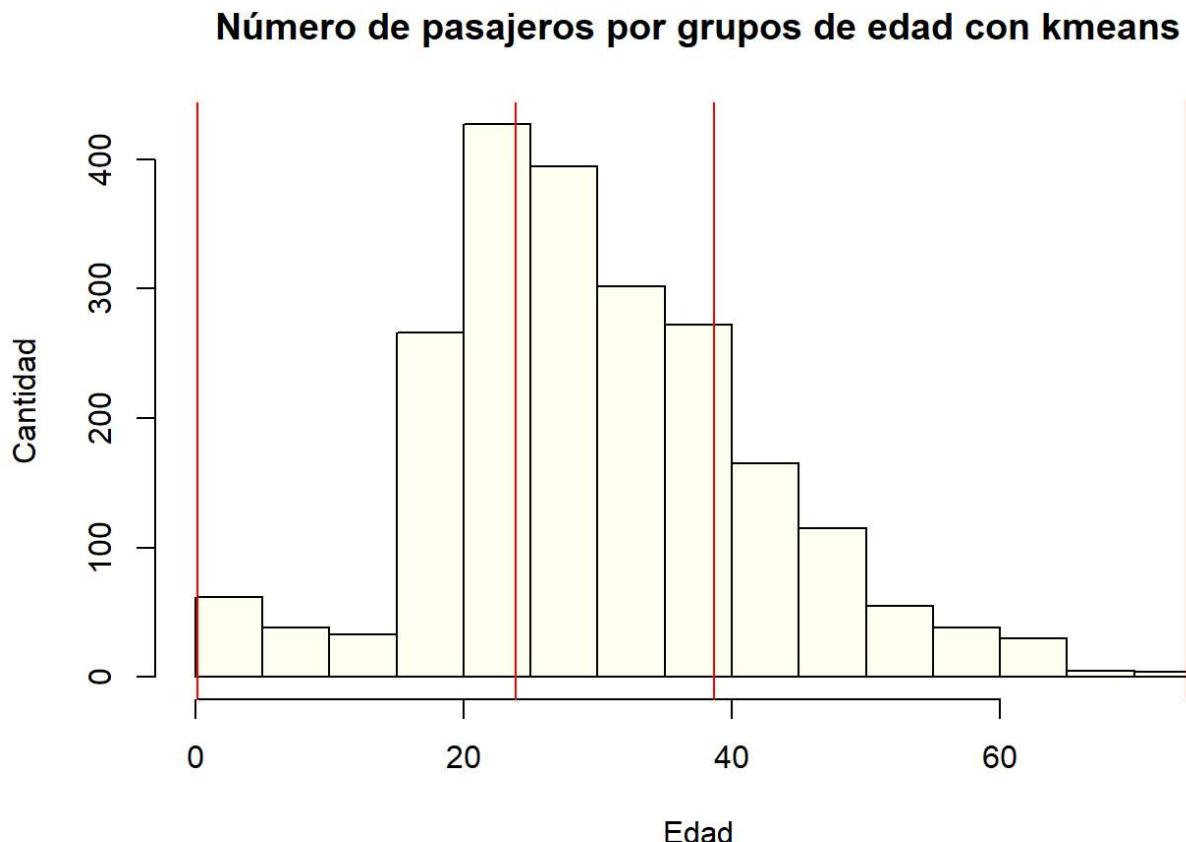
## The following object is masked from 'package:dplyr':
## 
##     recode

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

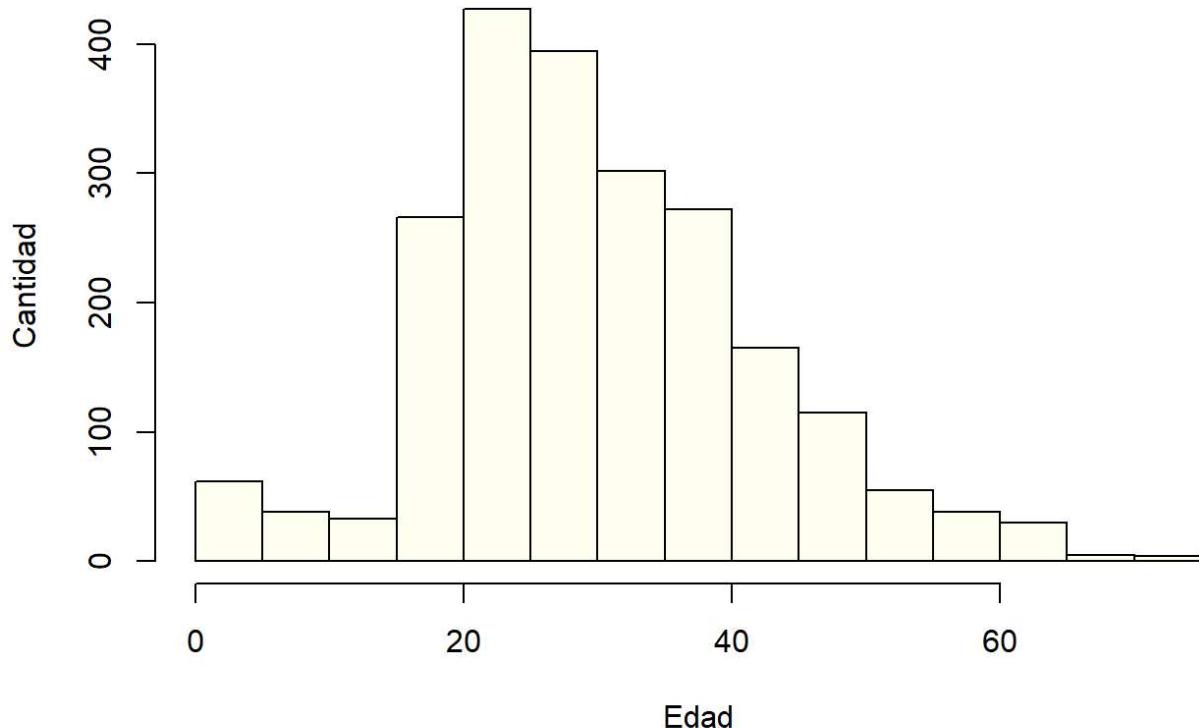
```

```

hist(totalData$age,xlab="Edad", col="ivory",ylab="Cantidad", main="NÃºmero de pasajeros por grupos de edad")

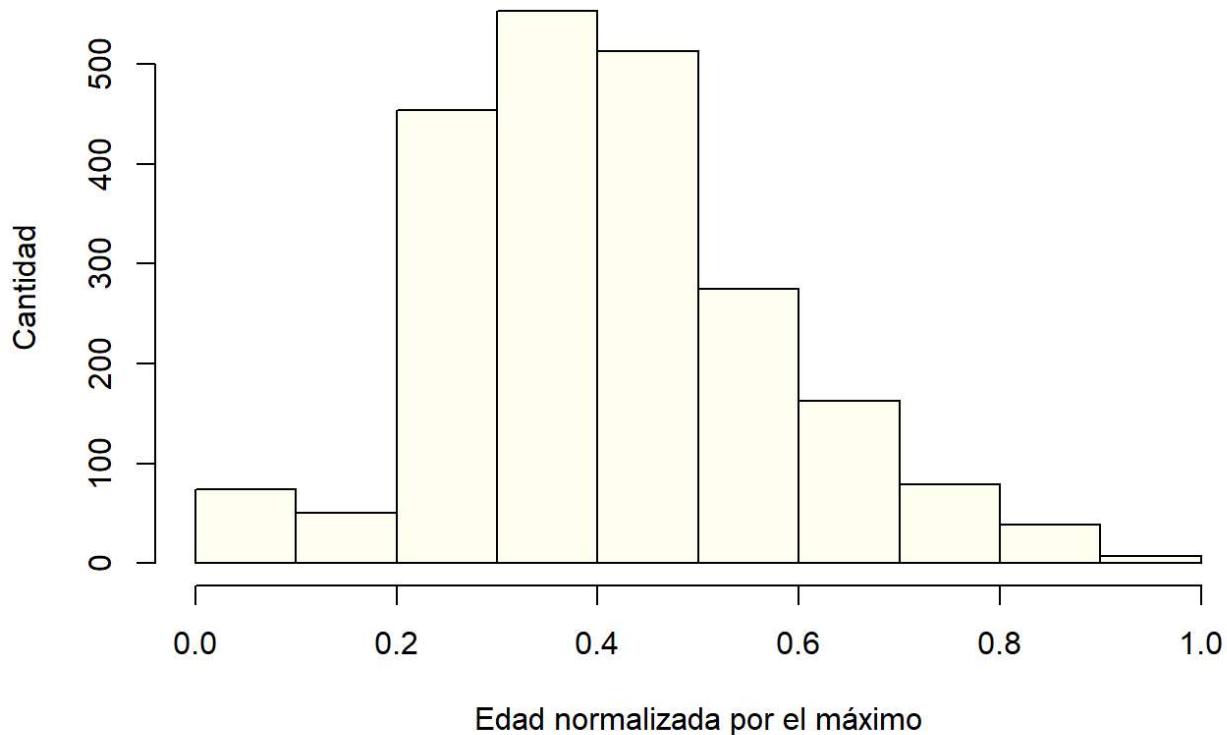
```

### Número de pasajeros por grupos de edad

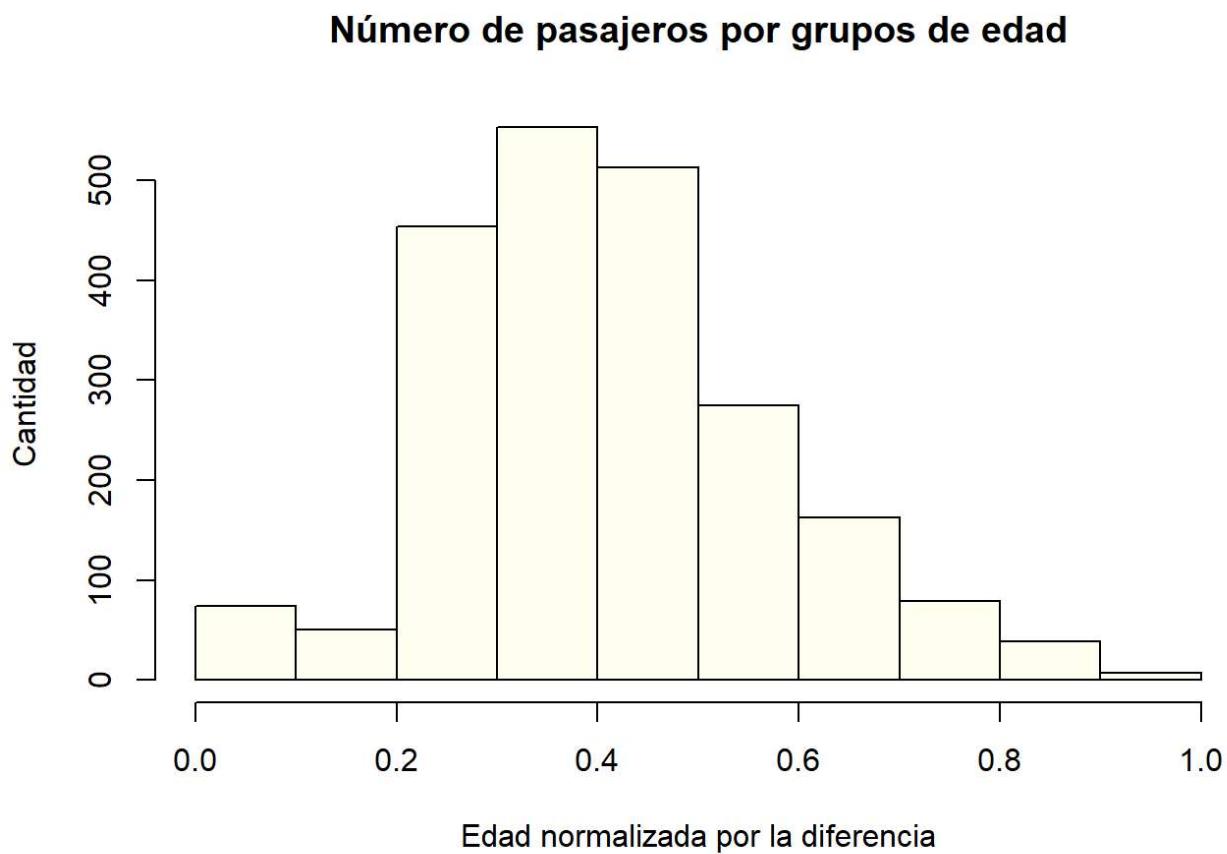


```
hist(totalData$age_NM,xlab="Edad normalizada por el máximo", ylab="Cantidad",col="ivory", main="Número de pasajeros por grupos de edad")
```

### 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")
```



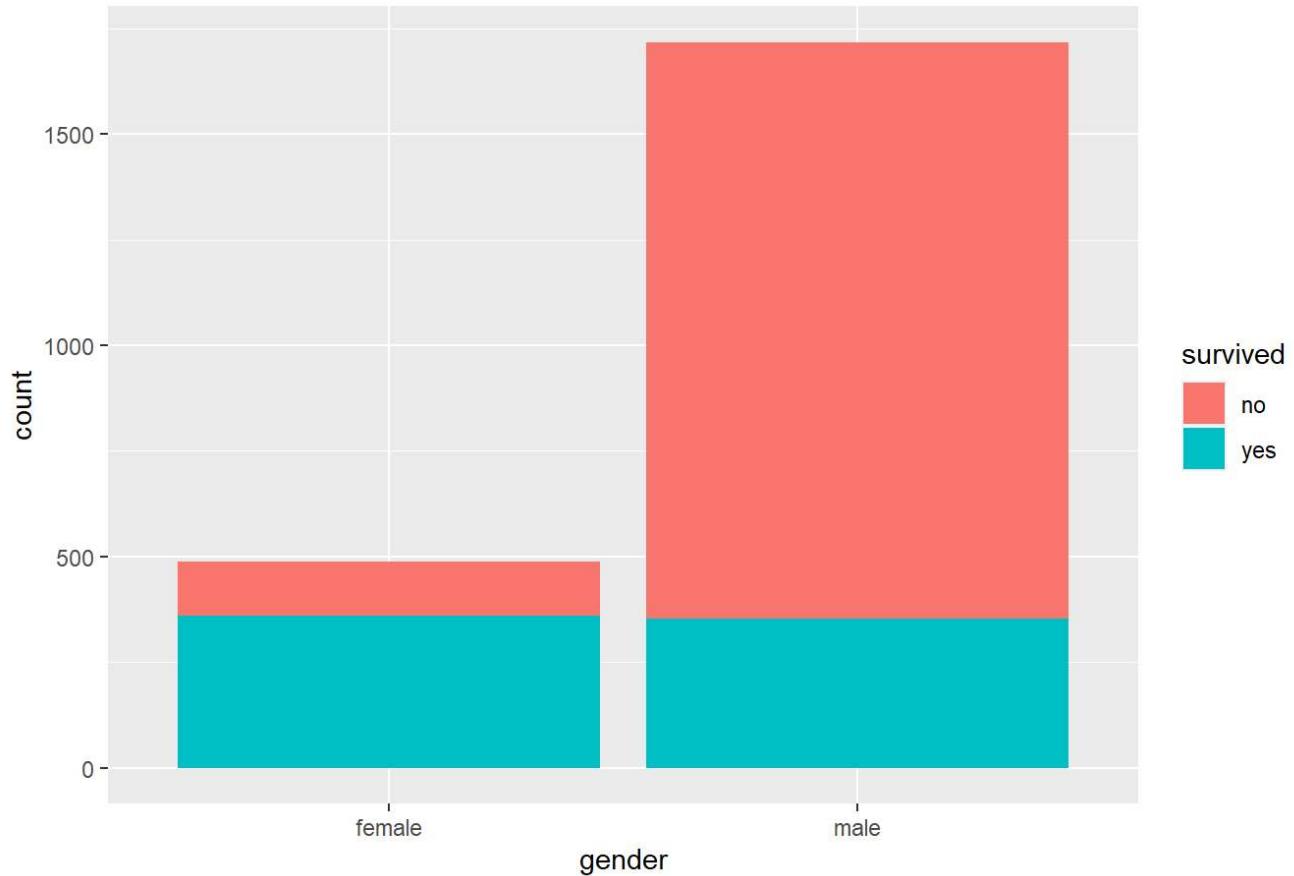
## 2.3 Procesos de análisis visuales del juego de datos

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("Relació n entre las variables gender y survived")
```

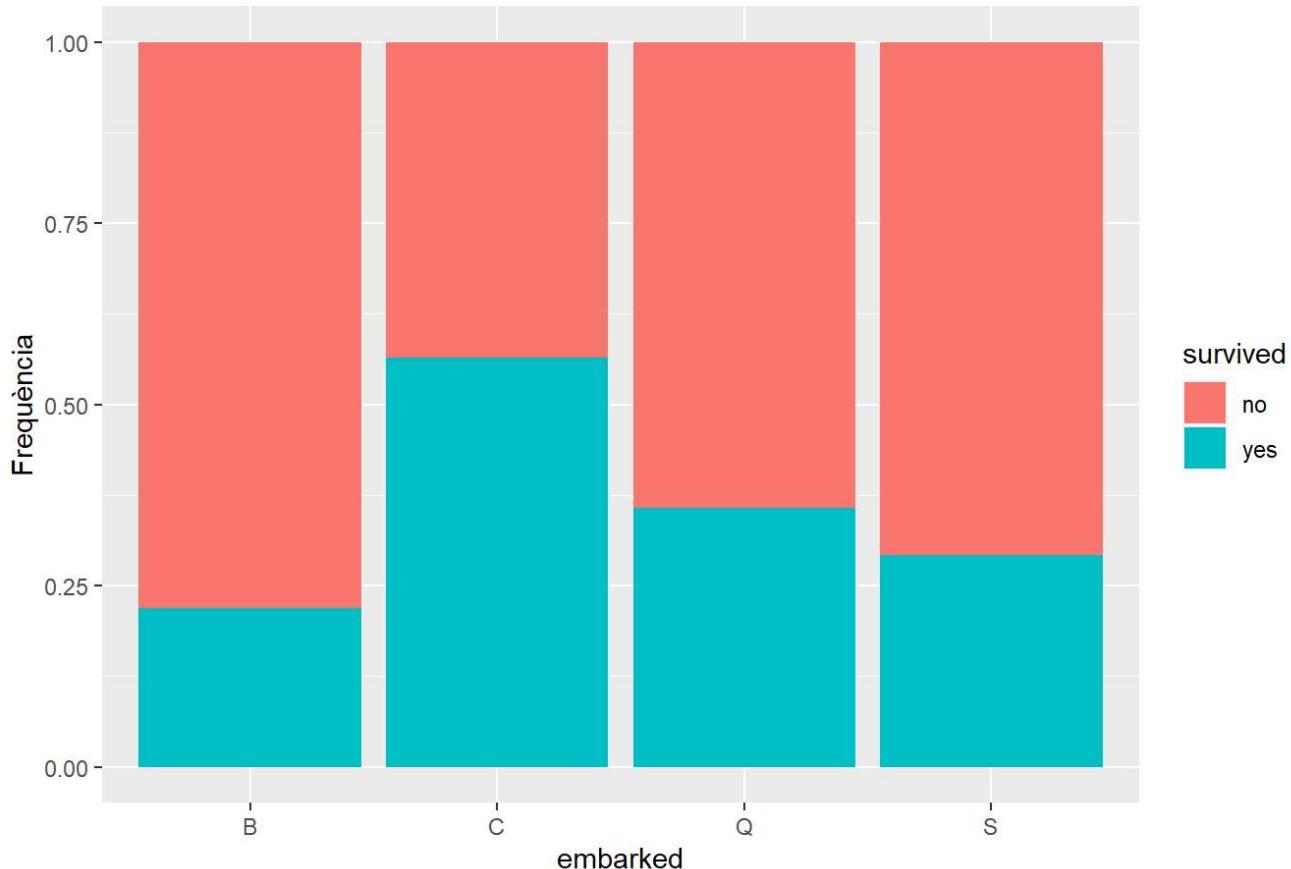
## Relación entre las variables gender y survived



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")
```

### 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%

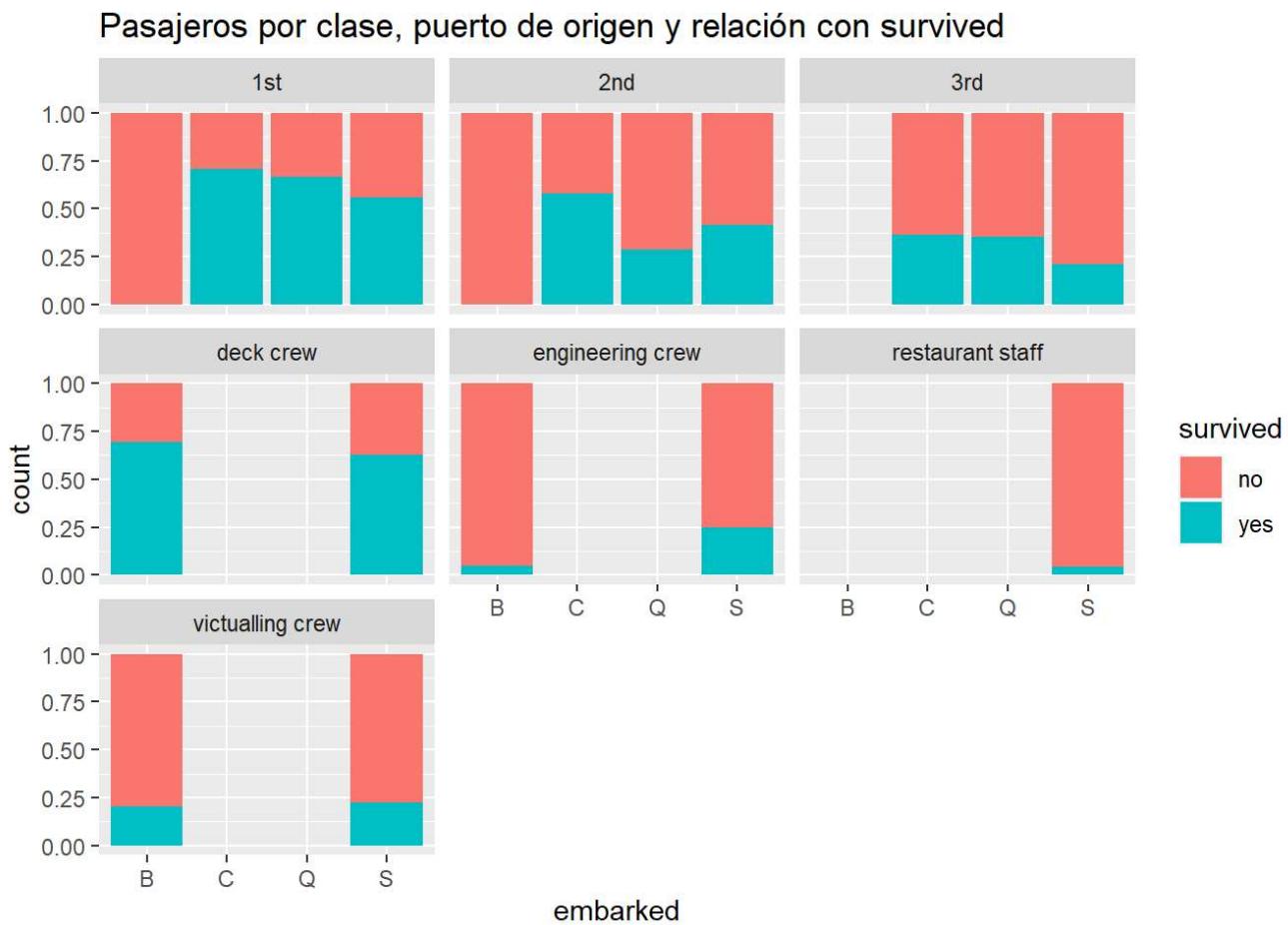
```
t<-table(totalData[1:filas,]$embarked,totalData[1:filas,]$survived)
for (i in 1:dim(t)[1]){
  t[i,]<-t[i,]/sum(t[i,])*100
}
t
```

```
##
##          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 Pclass:

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

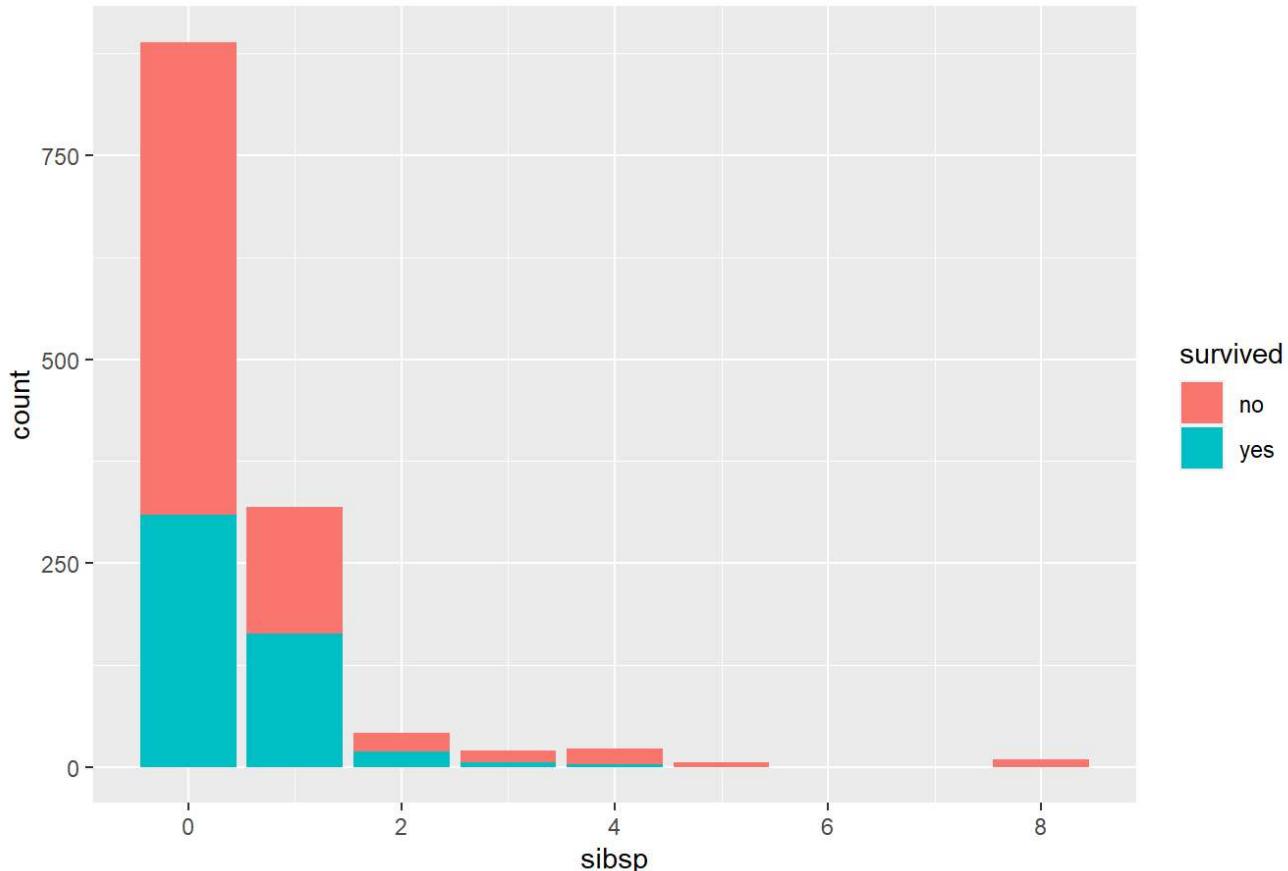


Aquí ya podemos extraer mucha información. Como propuesta de mejora se podría hacer un gráfico similar trabajando solo la clase. Habría que unificar toda la tripulación a una única categoría.

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

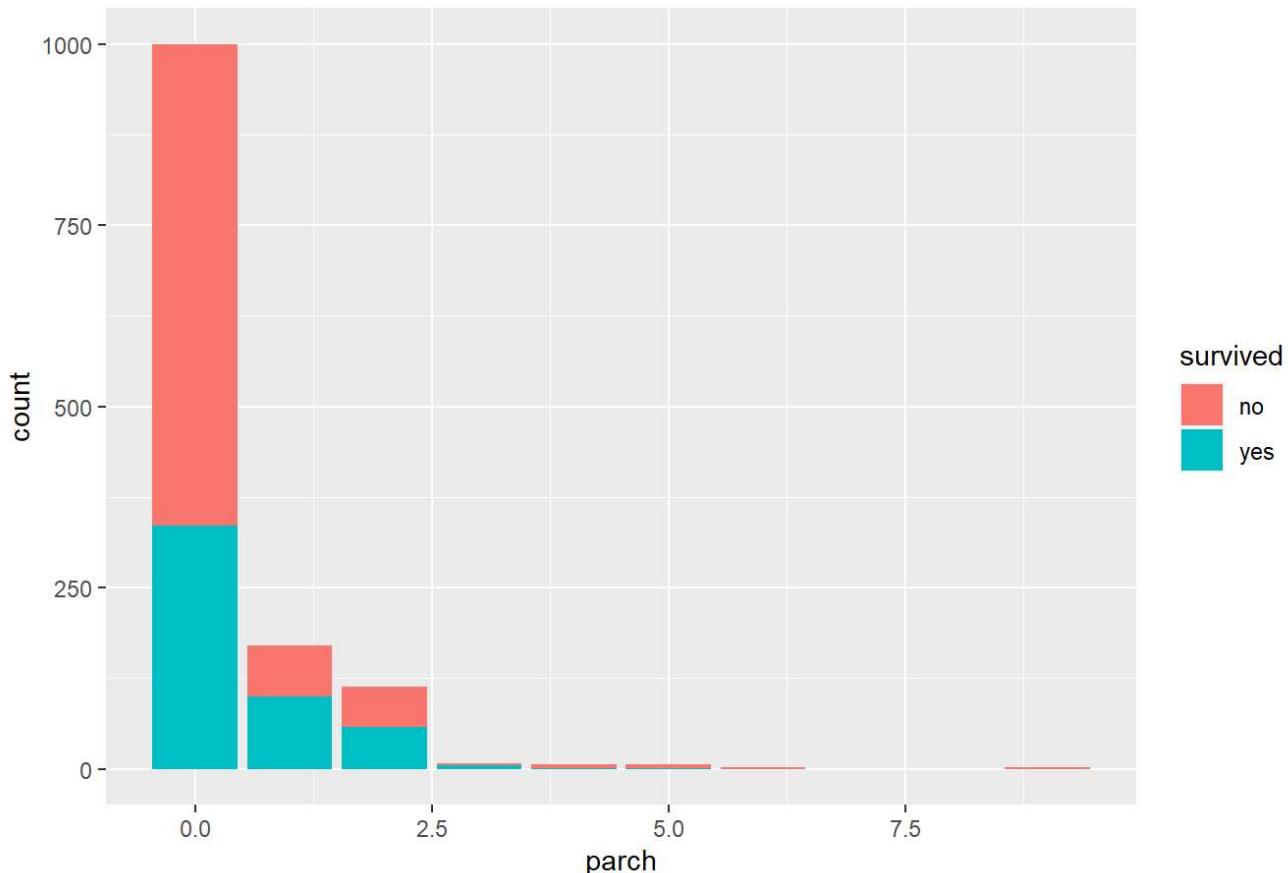
```
ggplot(data = totalData[1:filas,],aes(x=sibsp,fill=survived))+geom_bar() + ggtitle("Sobrevivir en función de tener a bordo cónyuges y/o hermanos")
```

### Sobrevivir en función de tener a bordo cónyuges y/o hermanos



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

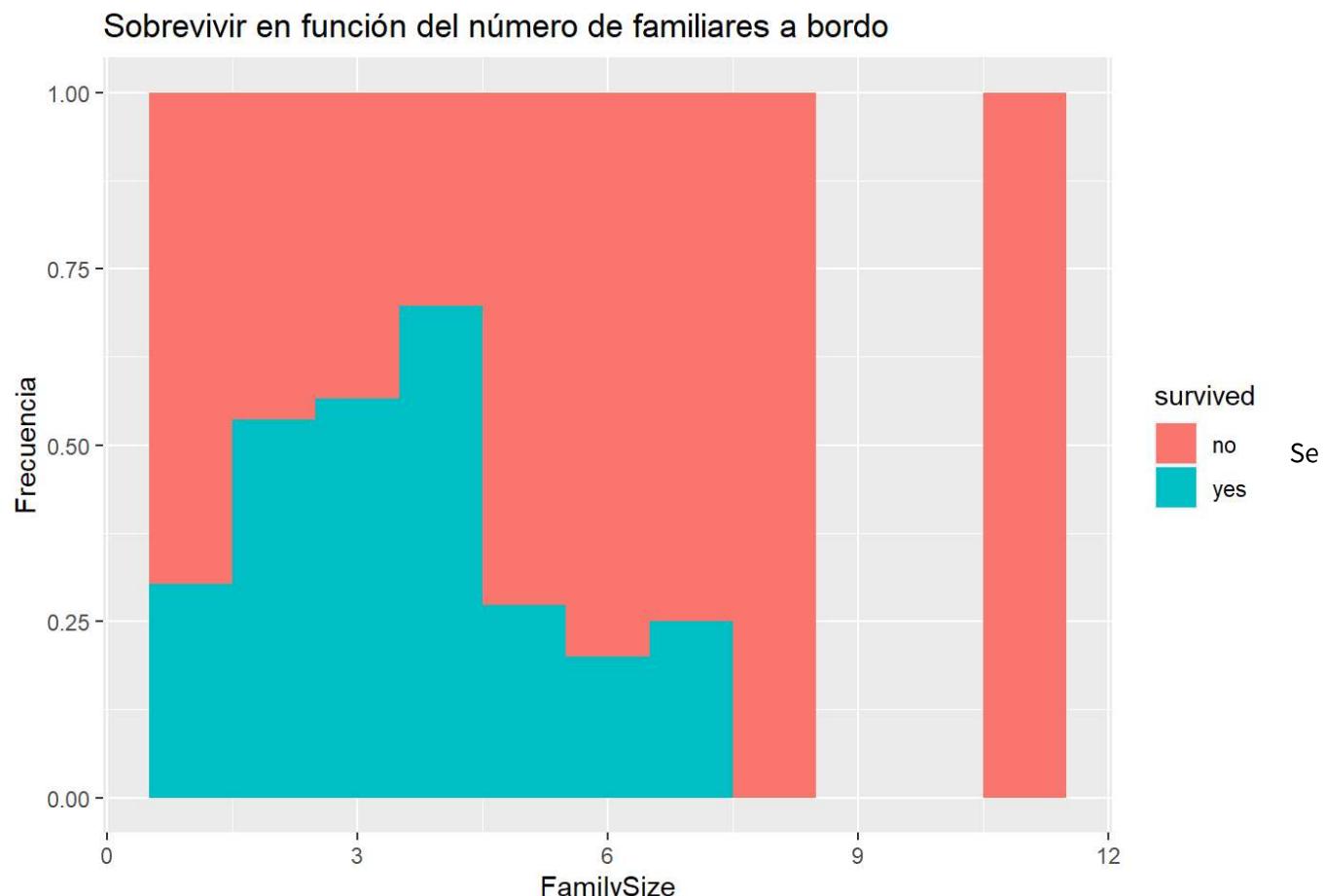
### Sobrevivir en función de tener a bordo padres y/o hijos



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")
```

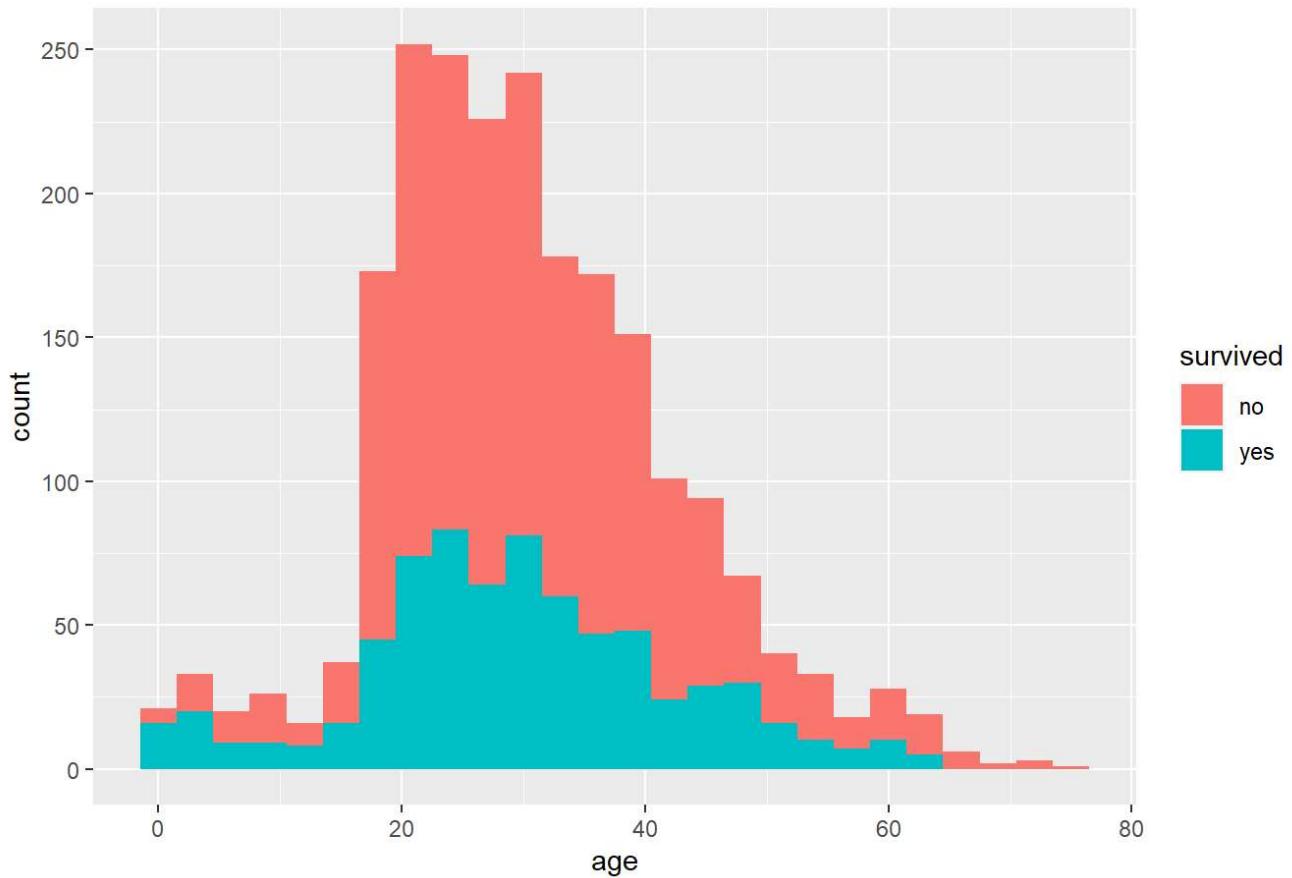


confirma el hecho de que los pasajeros viajaban mayoritariamente en familia. No podemos afirmar que el tamaño de la familia tuviera nada que ver con la posibilidad de sobrevivir pues nos tememos que estadísticamente el hecho de haber más familias de alrededor de cuatro miembros debería de ser habitual.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

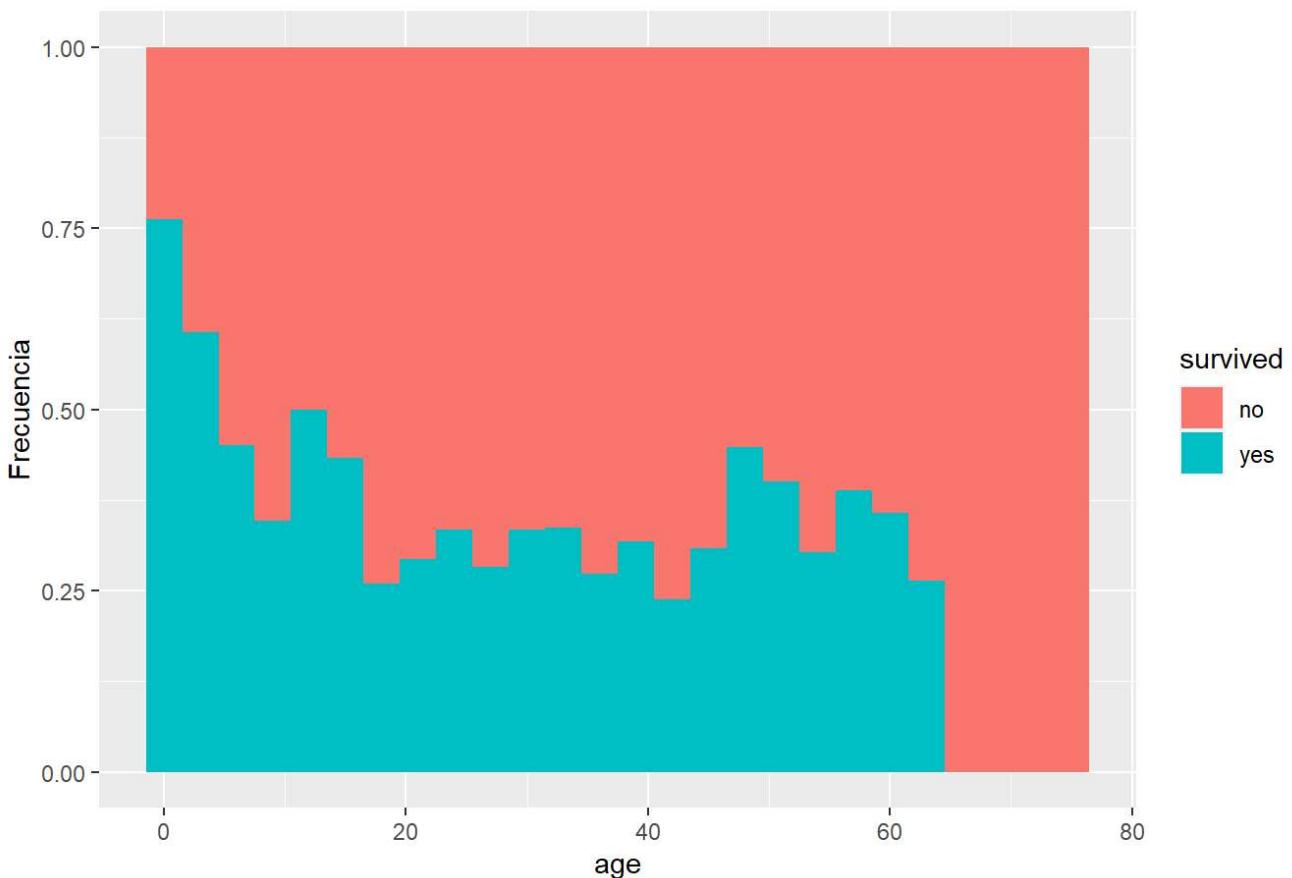
```
ggplot(data = totalData1[!(is.na(totalData[1:filas,]$age)),],aes(x=age,fill= survived))+  
geom_histogram(binwidth =3)+ggtitle("Sobrevivir en función de edad")
```

### Sobrevivir en función de edad



```
ggplot(data = totalData1[!is.na(totalData1[,filas],]$age),],aes(x=age,fill=survived))+geom_histogram(binwidth = 3,position="fill")+ylab("Frecuencia")+ggtitle("Sobrevivir en función de edad")
```

### Sobrevivir en función 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 dplyr    1.1.4     v stringr  1.4.0
## v readr    2.0.2     vforcats  0.5.1

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

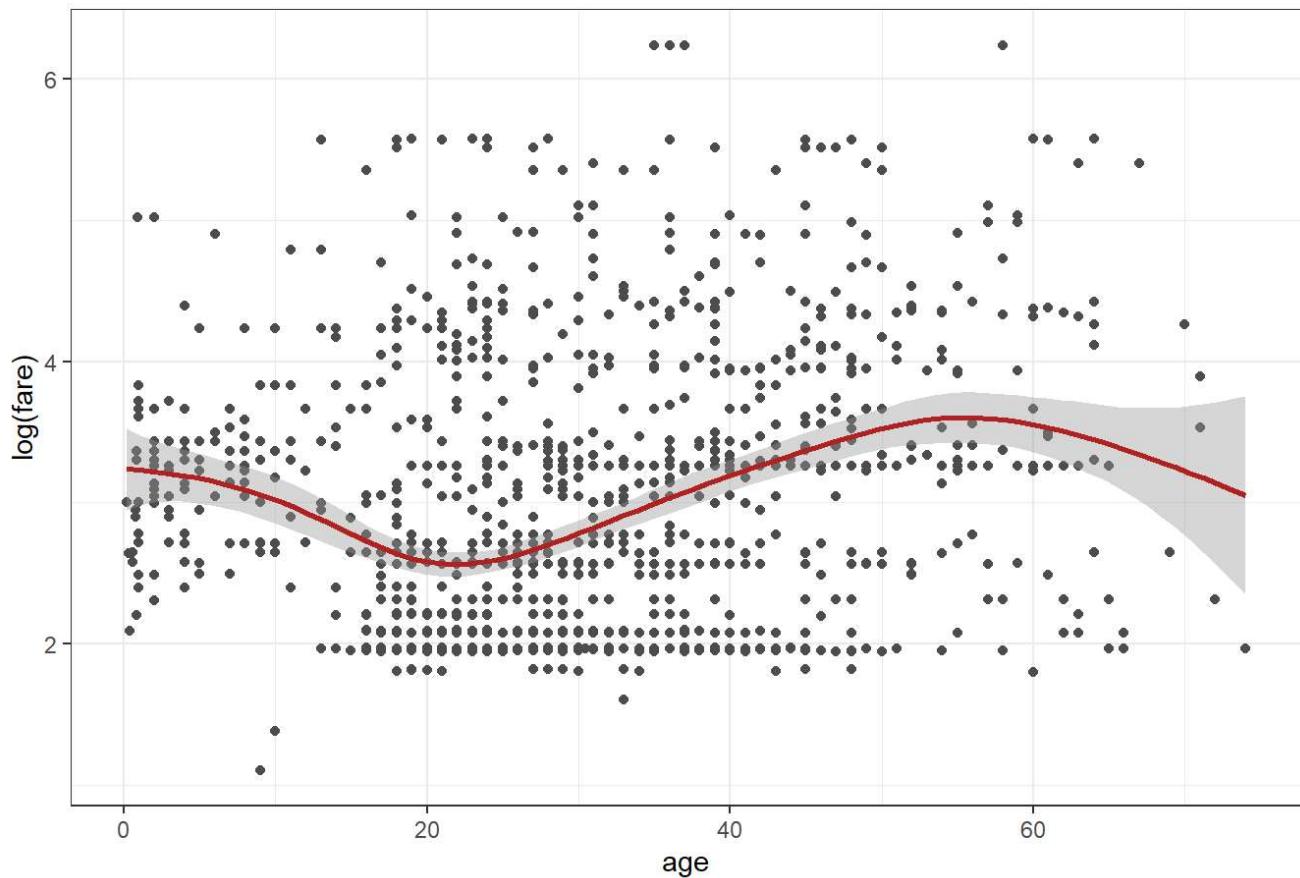
ggplot(data = totalData, aes(x = age, y = log(fare))) + geom_point(color = "gray30") + geom_smooth(color = "firebrick") + theme_bw() + ggtitle("Correlación entre precio billete y edad")

## `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).
```

Correlación entre precio billete y edad



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.

## 2.4 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.

## 3 Ejercicios

### 3.1 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

## **Definición de la tarea de minería de datos**

Actualmente estoy trabajando en una empresa eléctrica, exactamente en la parte de distribución. Una de las preocupaciones principales es mantener el suministro de energía eléctrica de una manera constante. Por eso un posible proyecto de minería de datos puede ser la predicción de los fallos en las líneas eléctricas siguiendo los patrones de consumo. Un ejemplo práctico para entender el fin del proyecto podría ser lo siguiente.

### *Ejemplo*

Tenemos una línea de distribución en la que se encuentra 3 transformadores de media a baja tensión y un contador del flujo de la electricidad en cada uno de los transformadores. El flujo de electricidad que corre por dicha red es de ~100 Vm , y tras pasar por los contadores de los transformadores esta se reduce a ~30 Vm metro en cada uno.

Con el paso del tiempo, el contador de uno de los transformadores empieza a contar una variación intermitente en el flujo de corriente (~25 Vm) que a simple vista puede no significar nada, pero con el paso de las semanas se produce una avería en el transformador que implique un corte en la línea eléctrica.

El modelo de predicción nos puede permitir detectar estos patrones para darnos una señal de alarma de que algo está pasando por ciertas variaciones en el flujo de corriente.

## **Origen de los datos**

Para obtener los datos y que puedan ser analizados se debería obtener de dos conjuntos de datos distintos: el primero que nos proporcione los datos relativos al consumo durante un periodo de tiempo, y el segundo conjunto que nos proporcione todos los datos relativos a las averías.

Los dos conjuntos de datos son necesarios para contrastar la información, y comprobar que variaciones o parámetros se han ido alterando y con qué periodo de tiempo se han producido estas variaciones antes de que se produjeran una avería.

Una aproximación de los atributos que podrían tener estos conjuntos de datos es: - Relativo al consumo de la Red: fecha, zona, flujo de corriente, versión de los equipos, parámetros relativos a la corriente... - Relativo a las averías de la Red: Nombre de la incidencia, fecha, zona, tiempo de resolución, coste...

## **Preparación de los datos**

La preparación de los datos es una etapa fundamental. Esto se debe a que se quiere obtener un modelo predictivo, lo más lógico (a mi entender) sería elegir las zonas con mayor calidad de datos y luego hacer una extrapolación al resto de las zonas.

### *Limpieza de los datos*

Como he comentado antes, solamente nos interesaría tener los datos sobre unas zonas concretas, para ello se tomaría los datos de la o las zonas en donde exista un menor número de datos incompletos, redundantes o inconsistentes.

Otra cosa para tener en cuenta es que debemos coger los datos a partir del último cambio en la infraestructura, es decir si tenemos un registro de 100 datos, y solamente el 30% son obtenidos después de un cambio considerable en la manera de la distribución de la electricidad (a nivel de infraestructura), son solamente esos datos los que se deben escoger y descartar el resto.

Además, se debe procurar que las zonas en las que se van a elegir el conjunto de los datos tengan la misma infraestructura.

Por otra parte, anteriormente se ha dicho que se van a usar dos conjuntos de datos distintas, en este proceso de limpieza de datos, afectaría a los dos conjuntos de datos.

### *Transformación de los datos*

En el proceso de transformación de los datos, debemos preparar los dos conjuntos de datos y fusionarlos.

Lo primero es normalizar los datos, por ejemplo, el tipo de avería se podría clasificar por un ID, así pasamos de una variable categórica a un valor numérico.

Otro ejemplo, podemos agrupar los valores similares dentro de un periodo de tiempo, es decir si durante 5 horas el flujo siempre ha sido el mismo, simplificar esos 5 registros en uno solo. Una vez que tenemos los dos conjuntos de datos preparados debemos fusionarlos para obtener un nuevo conjunto enriquecido.

De los conjuntos de datos, yo usaría las fechas y las zonas para la fusión de los datos, tiendo como resultado un conjunto de datos (aproximado) con los siguientes parámetros: - Fecha -> la fecha de obtención de los datos - Zona -> Indica la zona de donde se ha recogido los datos - ¿avería? -> Nos dice si tiene una avería, si no hay, estaría a Nulo. - Tipo de avería -> nos da el tipo de avería (con su ID) y si no hay estaría a Nulo. - Conjuntos de parámetros relativos al flujo eléctrico. -> un conjunto de datos que nos de las mediciones de los distintos sensores.

**Proceso de construcción de modelos** Una vez que tenemos los datos ya podemos hacer la fase de minería de datos, obteniendo modelos de predicción.

Una visión interesante puede ser, que, para cada tipo de avería, que datos sufren mas variaciones y con que periodo de tiempo es cuando ocurre, con esto se puede sacar unos modelos que dados una serie de circunstancia nos proporcione la probabilidad que ocurra una avería.

Para comprobar el modelo, se puede usar otro conjunto de datos en donde se pruebe si el porcentaje de predicción es válido. Por ejemplo, se puede usar otra zona en la que no se han obtenido mediciones y probar la eficacia del modelo.

**Integración de los resultados** Estos modelos, una vez probados y comprobado su funcionamiento, pueden ser integrados en los paneles de control (estos paneles comprueban y recogen los datos de la distribución de la corriente y se comprueba su correcto funcionamiento de los elementos de la red).

La integración con los paneles puede dar un aviso con las mediciones de los días de alteraciones producidas.

## 3.2 Ejercicio 2:

A partir del juego de datos disponible en el siguiente enlace <https://www.kaggle.com/rdoume/beerreviews> (<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.

## 3.3 CARGA Y VERIFICACIÓN DE LOS DATOS

```
# Cargamos el juego de datos (Los campos con espacio en blanco se pondrán como nulos)
Datos_Beer_Completos <- read.csv('beer_reviews.csv', stringsAsFactors = FALSE, na.strings = '')

#Creamos una nueva variable en donde haremos los cambios
Datos_Beer <- Datos_Beer_Completos

#Obtenemos el numero de filas
filas=dim(Datos_Beer)[1]

#Verificamos la estructura del juego de datos principal.
str(Datos_Beer)
```

```

## 'data.frame': 1586614 obs. of 13 variables:
## $ brewery_id      : int 10325 10325 10325 10325 1075 1075 1075 1075 1075 ...
## $ brewery_name    : chr "Vecchio Birraio" "Vecchio Birraio" "Vecchio Birraio" "Ve
cchio Birraio" ...
## $ review_time     : int 1234817823 1235915097 1235916604 1234725145 1293735206 13
25524659 1318991115 1306276018 1290454503 1285632924 ...
## $ 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 Stou
t" "German Pilsener" ...
## $ 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 521
59 ...

```

Comprobamos que tenemos 1586614 registros relativos a diversos tipos de cervezas y 13 variables que los caracterizan.

A continuación, haré un análisis de las variables contenidas en el fichero:

#### VARIABLES RELATIVAS A LA CERVECERIA

**brewery\_id** -> Esta variable corresponde con el ID que tiene la cervecería

**brewery\_name** -> Esta variable corresponde con el nombre de la cervecería

#### VARIABLES RELATIVAS A LA CERVEZA

**beer\_beerid** -> Esta variable corresponde con el id que se la da a cada cerveza.

**beer\_name** -> Esta variable corresponde con el nombre de la cerveza.

**beer\_style** -> Esta variable corresponde con el estilo de la cerveza.

**beer\_abv** -> Esta variable corresponde con el grado de alcohol de la cerveza.

#### VARIABLES RELATIVAS A LOS USUARIOS Y SUS OPINIONES

**review\_profilename** -> Esta variable corresponde con el usuario que ha hecho la review.

**review\_palate** -> Esta variable corresponde con la sensación en el paladar de la cerveza.

**review\_taste** -> Esta variable corresponde con el sabor de la cerveza.

**review\_aroma** -> Esta variable corresponde con el aroma de la cerveza.

**review\_appearance** -> Esta variable corresponde con la apariencia de la cerveza.

**review\_overall** -> Esta variable corresponde con la valoración general de la cerveza.

**review\_time** -> Esta variable corresponde con la fecha de realización de la review.

La primeras conclusiones en relación a las variables son:

- **brewery\_id** y **brewery\_name** -> se puede concluir que asocia cada ID con el nombre de cada cervecería.

- **beer\_beerid** y **beer\_name** -> se puede concluir que asocia cada ID con el nombre de cada cerveza.
- **review\_time** -> Esta variable debe ser normalizada de formato UNIX a formato de fecha.

```
#Sacamos Las estadísticas básicas
summary(Datos_Beer)
```

```
##      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
##  3rd Qu.:4.000  3rd Qu.:4.500
##  Max.   :5.000  Max.   :5.000
##                               NA's   :67785
##
##      beer_beerid
##  Min.   :    3
##  1st Qu.: 1717
##  Median :13906
##  Mean   :21713
##  3rd Qu.:39441
##  Max.   :77317
##
```

## 3.4 LIMPIEZA DEL CONJUNTO DE DATOS

Como se puede comprobar, y en el análisis inicial que he realizado, las variables ID`s y los nombres de las cervecerías y cervezas están relacionadas.

Aunque a simple vista de las 4 variables se pueden descartar la mitad, creo que seria correcto (antes de hacer en análisis) gestionar si tienen algunos valores nulos para luego a la hora de obtener las conclusiones no se ponga simplemente un ID refiriendo a la cerveza o cervecería, si no que estas tengan un nombre.

```

# Cargamos los paquetes R que vamos a usar
library(ggplot2)
library(dplyr)
library(plyr)
library(scales)

# Estadísticas de valores Nulos (como antes he asignado el valor "" como nulo, solo se filtra por los valores nulos)

colSums(is.na(Datos_Beer))

```

```

##      brewery_id      brewery_name      review_time      review_overall
##            0                  15                  0                  0
##      review_aroma  review_appearance  review_profilename      beer_style
##            0                      0                  348                  0
##      review_palate      review_taste      beer_name      beer_abv
##            0                      0                  0                67785
##      beer_beerid
##            0

```

Como se puede comprobar, tenemos 15 valores nulos en los nombres de la cervecerías (brewery\_name), 348 en los nombre de usuarios (review\_profilename) y 67785 en la graduación de alcohol de las cervezas (beer\_abv).

Para el primer y último caso se va a intentar asignar los valores buscando sus ID's y viendo si en otra entrada con el mismo ID estos valores están completos y para el caso de los nombres de usuarios (review\_profilename) se asignará un valor por defecto (“Anonimo”).

### Valores nulos en el nombre de usuario (review\_profilename)

```

#Asignamos el valor "Anonimo" en los valores nulos
Datos_Beer$review_profilename[is.na(Datos_Beer$review_profilename)] <- "Anonimo"

#Comprobamos que para ese tipo de datos ya no hay valores nulos
colSums(is.na(Datos_Beer))

```

```

##      brewery_id      brewery_name      review_time      review_overall
##            0                  15                  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

```

### Valores nulos en el nombre de la cervecería (brewery\_name)

```

#Obtenemos los ID's de los valores cuyo nombre de cervecería sea nulo
valor = Datos_Beer$brewery_id[is.na(Datos_Beer$brewery_name)]

#Comprobamos que el tamaño coincide con los valores obtenidos antes
length(valor)

```

```
## [1] 15
```

```
#Quitamos los duplicados  
valorSinDuplicados = valor[!duplicated(valor)]
```

```
#Comprobamos el tamaño después de quitar duplicados  
length(valorSinDuplicados)
```

```
## [1] 2
```

```
#Vemos los ID's (Al ser 2 se pueden ver facilmente)  
valorSinDuplicados
```

```
## [1] 1193    27
```

```
#Comprobamos si para cada Id existe algún registro completo
```

```
for(i in valorSinDuplicados){  
  
  aux = is.na(Datos_Beer$brewery_name[valorSinDuplicados[i]])  
  
  print(aux)  
}
```

```
## [1] TRUE  
## [1] TRUE
```

Como se puede comprobar los ID's 1193 y 27 no tienen ningún campo cuyo nombre de la cervecería (brewery\_name) tenga asignado un valor, por lo que asignamos el valor “Desconocido” para los valores vacíos de la variable brewery\_name.

```
#Asignamos el valor "Desconocido"  
Datos_Beer$brewery_name[is.na(Datos_Beer$brewery_name)] <- "Desconocido"
```

```
#Comprobamos que para ese tipo de datos ya no hay valores nulos  
colSums(is.na(Datos_Beer))
```

```
##      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
```

**Valores nulos en la graduación (beer\_abv) de la cerveza**

```
#Obtenemos Los ID`s de La cerveza de Los valores cuya graduación sea nula  
valorId = Datos_Beer$beer_beerid[is.na(Datos_Beer$beer_abv)]
```

```
#Comprobamos que el tamaño coincide con Los valores obtenidos antes  
length(valorId)
```

```
## [1] 67785
```

```
#Quitamos Los duplicados  
valorIdSinDuplicados= valorId[!duplicated(valorId)]
```

```
#Comprobamos el tamaño después de quitar duplicados  
length(valorIdSinDuplicados)
```

```
## [1] 17043
```

```
#Comprobamos si para cada Id existe algún registro completo
```

```
for(i in valorIdSinDuplicados){
```

```
#Obtenemos Los valores que no sean nulos, excluyendo Los nulos.
```

```
listOfValues <- na.omit(Datos_Beer$beer_abv[Datos_Beer$beer_beerid == i])
```

```
#Si existe algun caso, asignamos el valor de La graduación a todos Las cervezas con el mismo ID.
```

```
if(length(listOfValues)>0 ){  
    Datos_Beer$beer_abv[Datos_Beer$beer_beerid == i] <- listOfValues[1]  
}  
}
```

```
#Comprobamos que para esa tipo de datos si ya no hay valores nulos  
colSums(is.na(Datos_Beer))
```

```
##      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
```

Para los valores cuyo ID`s que no tienen ningún campo beer\_abv completo, se le asignará un valor por defecto. El valor por defecto será el más común.

```

#Función para calcular el valor más común
common_value <- function(x) {
  uniqx <- unique(na.omit(x))
  uniqx[which.max(tabulate(match(x, uniqx)))]
}

#Calculamos el valor más comun
abv_comun <- common_value(Datos_Beer$beer_abv)

#Asignamos el valor
Datos_Beer$beer_abv[is.na(Datos_Beer$beer_abv)] <- abv_comun

#Comprobamos que para esa tipo de datos ya no hay valores nulos
colSums(is.na(Datos_Beer))

```

##	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	0
##	beer_beerid			
##	0			

Ahora que hemos tratado los valores nulos, para simplificar el análisis voy a excluir de manera temporal el nombre de la cervecería (brewery\_name) y el nombre de la cerveza (beer\_name), ya que estos tienen su propio ID. Dichos nombres serán recuperados para las conclusiones.

Por otro lado el nombre de quien ha hecho la review (review\_profilename) no lo veo necesario, ya que las conclusiones las queremos sacar de las cervezas en general.

```

#Eliminamos Los campos que no usaremos en el análisis.
Datos_Beer$beer_name <- NULL
Datos_Beer$brewery_name <- NULL
Datos_Beer$review_profilename <- NULL

```

## 3.5 NORMALIZACIÓN Y DISCRETIZACIÓN DE LOS DATOS

### Normalizar review\_time

En el conjunto de datos, el campo review\_time estaba en formato UNIX, para el análisis debemos tenerlo en forma de fecha.

```

#Función para cambiar de UNIX a fecha normal
unix_Date_Format <- function(x) {
  as.POSIXct(x, origin="1970-01-01")
}

#Cambiamos el formato en todos Los campos
Datos_Beer$review_time <- unix_Date_Format(Datos_Beer$review_time)

#Nos quedamos solamente con La parte de Los años
Datos_Beer$year <- as.Date(Datos_Beer$review_time)
Datos_Beer$year <- as.numeric(format(Datos_Beer$year, '%Y'))

#Eliminamos el campo review_time que ya ha sido normalizado y preparado
Datos_Beer$review_time <- NULL

```

## Discretizacion de atributos

```

#Sacamos información sobre las distintas posibilidades que existen para cada atributo
apply(Datos_Beer, 2, function(x) length(unique(x)))

```

	brewery_id	review_overall	review_aroma	review_appearance
##	5840	10	9	10
##	beer_style	review_palate	review_taste	beer_abv
##	104	9	9	530
##	beer_beerid	year		
##	66055	16		

## Discretizamos las variables con pocas clases

```

#Preparamos la variable beer_style

#Discretizamos la variable beer_style
style_ID<-as.factor(Datos_Beer$beer_style)

#Añadimos el campo a la tabla
Datos_Beer$beer_style_ID <- unclass(style_ID)

# Discretizamos las variables con pocas clases
cols<-c("review_overall","review_aroma","review_appearance","review_palate", "review_taste", "year")

for (i in cols){
  Datos_Beer[,i] <- as.factor(Datos_Beer[,i])
}

#Eliminamos el campo beer_style que ya ha sido normalizado y preparado
Datos_Beer$beer_style <- NULL

#Después de los cambios, analizamos la nueva estructura del conjunto de datos
str(Datos_Beer)

```

```

## 'data.frame': 1586614 obs. of 10 variables:
## $ brewery_id      : int 10325 10325 10325 10325 1075 1075 1075 1075 1075 1075 ...
## $ review_overall   : Factor w/ 10 levels "0","1","1.5",...: 3 6 6 6 8 6 7 6 8 9 ...
## $ review_aroma     : Factor w/ 9 levels "1","1.5","2",...: 3 4 4 5 8 6 6 4 5 6 ...
## $ review_appearance: Factor w/ 10 levels "0","1","1.5",...: 5 6 6 7 8 7 7 7 7 10 ...
## $ review_palate    : Factor w/ 9 levels "1","1.5","2",...: 2 5 5 4 7 5 7 3 6 7 ...
## $ review_taste     : Factor w/ 9 levels "1","1.5","2",...: 2 5 5 5 8 6 7 6 7 7 ...
## $ 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 ...
## $ year              : Factor w/ 16 levels "1996","1998",...: 13 13 13 13 13 14 16 15 15 1 ...
## $ beer_style_ID     : int 66 52 60 62 10 67 67 67 67 ...
## ... - attr(*, "levels")= chr [1:104] "Altbier" "American Adjunct Lager" "American Amber / Red Ale" "American Amber / Red Lager" ...

```

Como la graduación del alcohol tiene demasiados valores para el análisis, con lo que vamos a agrupar los 530 niveles de estudio en 4 niveles.

```

# 1-4: Sin alcohol(1), Tradicional(2), Especial(3), Gran Reserva(4)
Datos_Beer$beer_abv[between(Datos_Beer$beer_abv,0.0,1.0)] <- 1
Datos_Beer$beer_abv[between(Datos_Beer$beer_abv,1.01,5.0)] <- 2
Datos_Beer$beer_abv[between(Datos_Beer$beer_abv,5.01,15.0)] <- 3
Datos_Beer$beer_abv[between(Datos_Beer$beer_abv,15.01,60)] <- 4

str(Datos_Beer)

```

```

## 'data.frame': 1586614 obs. of 10 variables:
## $ brewery_id      : int 10325 10325 10325 10325 1075 1075 1075 1075 1075 ...
## $ review_overall   : Factor w/ 10 levels "0","1","1.5",...: 3 6 6 6 8 6 7 6 8 9 ...
## $ review_aroma     : Factor w/ 9 levels "1","1.5","2",...: 3 4 4 5 8 6 6 4 5 6 ...
## $ review_appearance: Factor w/ 10 levels "0","1","1.5",...: 5 6 6 7 8 7 7 7 7 10 ...
## $ review_palate    : Factor w/ 9 levels "1","1.5","2",...: 2 5 5 4 7 5 7 3 6 7 ...
## $ review_taste     : Factor w/ 9 levels "1","1.5","2",...: 2 5 5 5 8 6 7 6 7 7 ...
## $ beer_abv         : num 2 3 3 2 3 2 2 2 2 ...
## $ beer_beerid      : int 47986 48213 48215 47969 64883 52159 52159 52159 52159 5215 ...
9 ...
## $ year              : Factor w/ 16 levels "1996","1998",...: 13 13 13 13 13 14 16 15 15 1 ...
4 14 ...
## $ beer_style_ID     : int 66 52 60 62 10 67 67 67 67 ...
## ... - attr(*, "levels")= chr [1:104] "Altbier" "American Adjunct Lager" "American Amb ...
er / Red Ale" "American Amber / Red Lager" ...

```

## 3.6 Procesos de análisis del conjunto de datos

```

#La Librería zoom se usa para ver Los gráficos de una manera más grande y poder ampliar para ver los datos
#install.packages("zoom")
library(zoom)
#Para usar el zoom, poner el comando de abajo después del código del grafico
#zm()

```

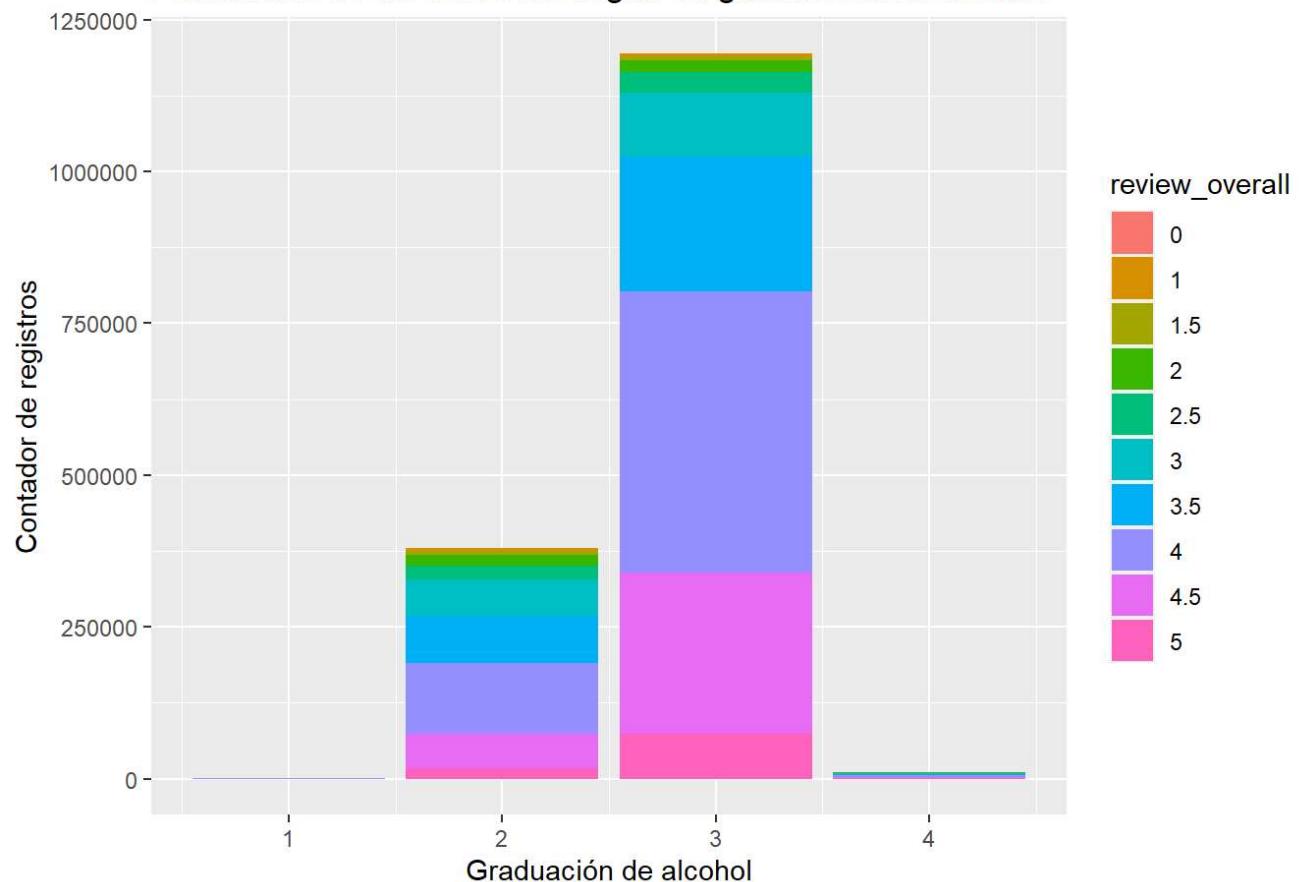
Nos proponemos analizar las relaciones entre las diferentes variables del conjunto de datos.

```

#Comparamos el grado de alcohol de la cerveza (beer_abv) con la puntuacion que esta tiene (review_overall)
ggplot(data=Datos_Beer[1:filas,],aes(x=beer_abv,fill=review_overall))+geom_bar()+ylab("Contador de registros")+xlab("Graduación de alcohol")+ggtitle("Puntuación de las cervezas según su graduación de alcohol")

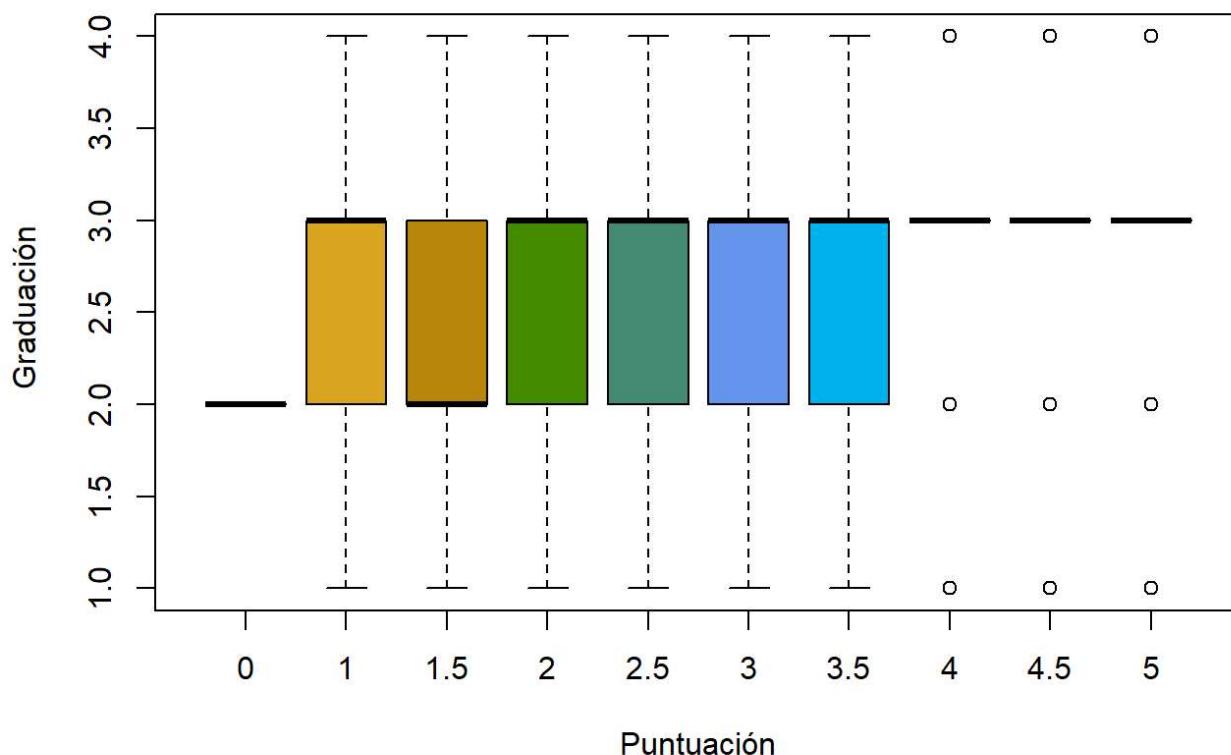
```

### Puntuación de las cervezas según su graduación de alcohol



```
#http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf Link de Los colores
boxplot (beer_abv ~ review_overall, Datos_Beer,
main = "Distribución de puntuación por graduación de alcohol", xlab = "Puntuación", ylab
= " Graduación",
col = c("coral1", "goldenrod", "darkgoldenrod", "chartreuse4","aquamarine4","cornflowerblue","deepskyblue2","darkorchid1","hotpink3","deeppink"))
```

## Distribución de puntuación por graduación de alcohol



```
#Para un análisis más efectivo se calcularan los porcentajes
t<-table(Datos_Beer[1:filas,]$beer_abv,Datos_Beer[1:filas,]$review_overall)
for (i in 1:dim(t)[1]){
t[i,]<-t[i,]/sum(t[i,])*100
}
t
```

```
##
##          0           1           1.5          2           2.5
## 1  0.000000000 12.84755129 10.450623202 16.203259827 15.436241611
## 2  0.001842289  1.379085277  1.708328728  4.460709229  6.310367879
## 3  0.000000000  0.437603805  0.514498241  1.722719837  2.824092474
## 4  0.000000000  3.344801223  2.159785933  4.950305810  6.049311927
##
##          3           3.5          4           4.5          5
## 1 17.545541707 11.697027804 11.601150527  2.492809204  1.725790988
## 2 15.412067523 20.722861760 30.755970334 14.686468647  4.562298335
## 3  8.845286555 18.491396441 38.720489982 22.316120638  6.127792025
## 4 11.343654434 18.702217125 28.822629969 17.641437309  6.985856269
```

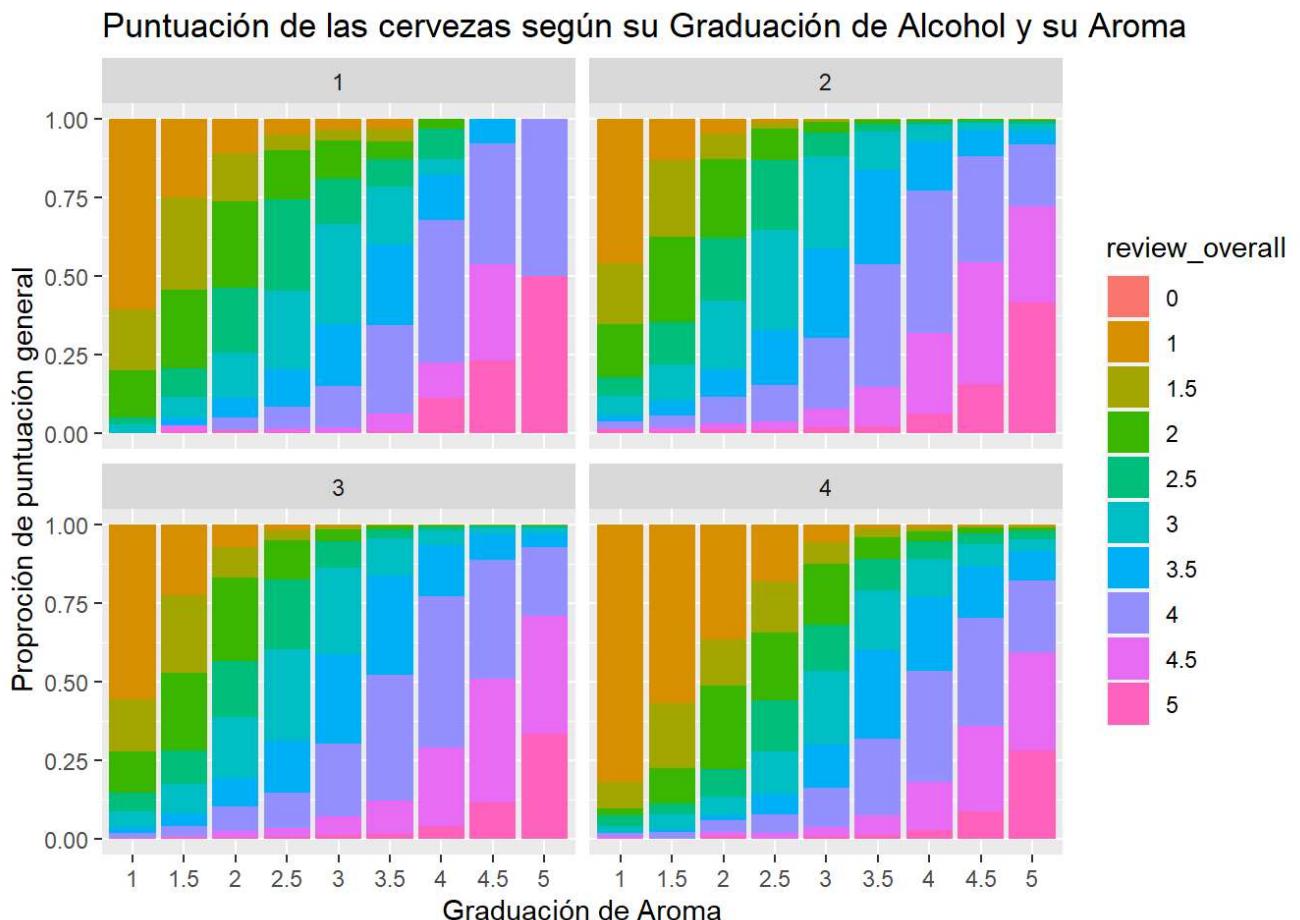
Como se puede observar, las cervezas con una graduación media (Tradicional y Especial) tienen un mayor numero de valoraciones respecto a las de una graduación mas baja o mucho más alta.

Si nos fijamos en los porcentajes, el grueso de las valoraciones respecto a la graduación de la cerveza se encuentra en la puntuación 4 y la mejor puntuación (un 5) se las lleva las cervezas con una graduación muy alta.

Sabiendo esto, compararemos las diferentes reviews realizadas respecto al alcohol de la cerveza.

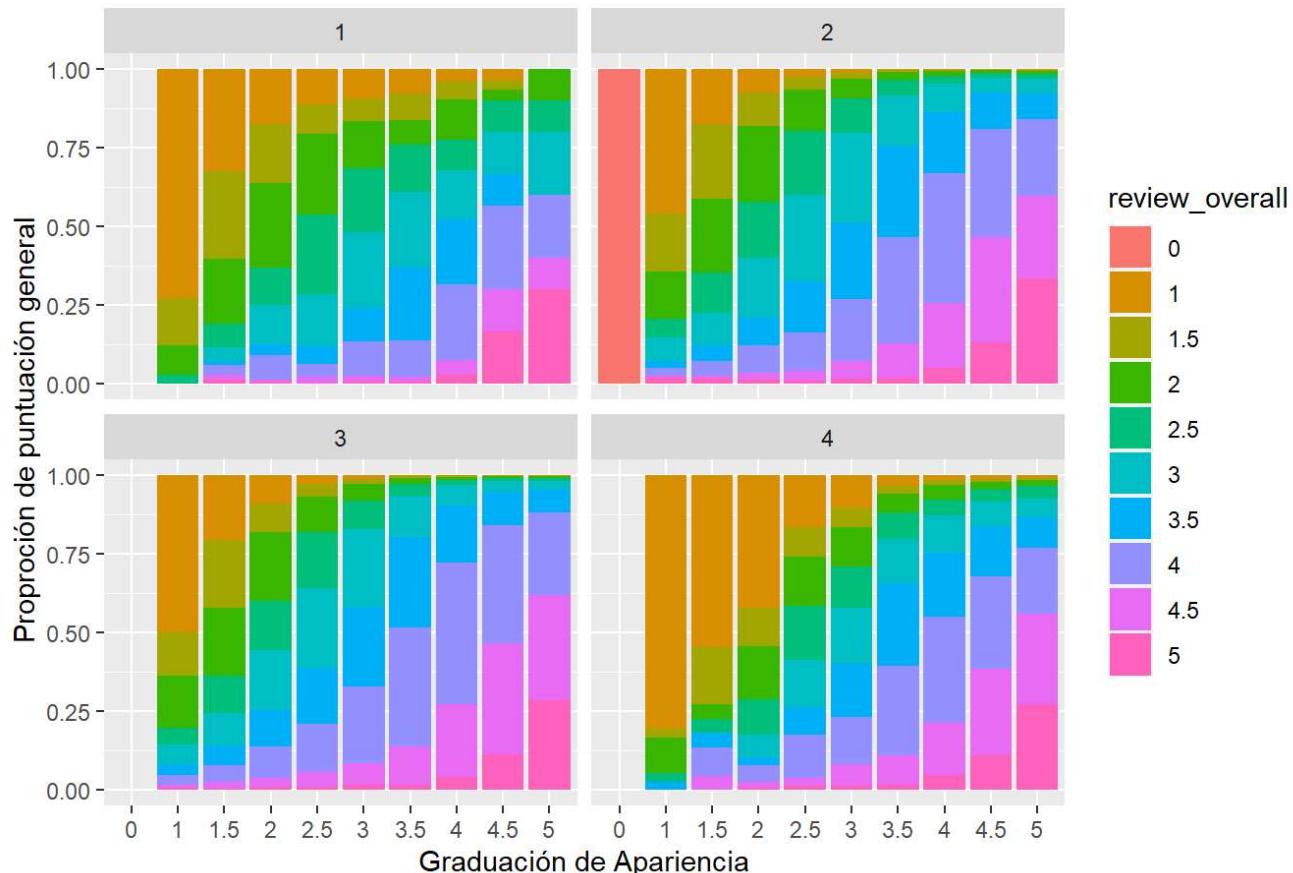
#Comparamos el alcohol de las cervezas, la valoración de estas con las distintas review: aroma, apariencia, sabor paladar, sabor.

```
ggplot(data = Datos_Beer[1:filas,],aes(x=review_aroma,fill=review_overall))+geom_bar(position="fill")+facet_wrap(~beer_abv)+ylab("Proporción de puntuación general") +xlab("Graduación de Aroma") +ggtitle("Puntuación de las cervezas según su Graduación de Alcohol y su Aroma")
```



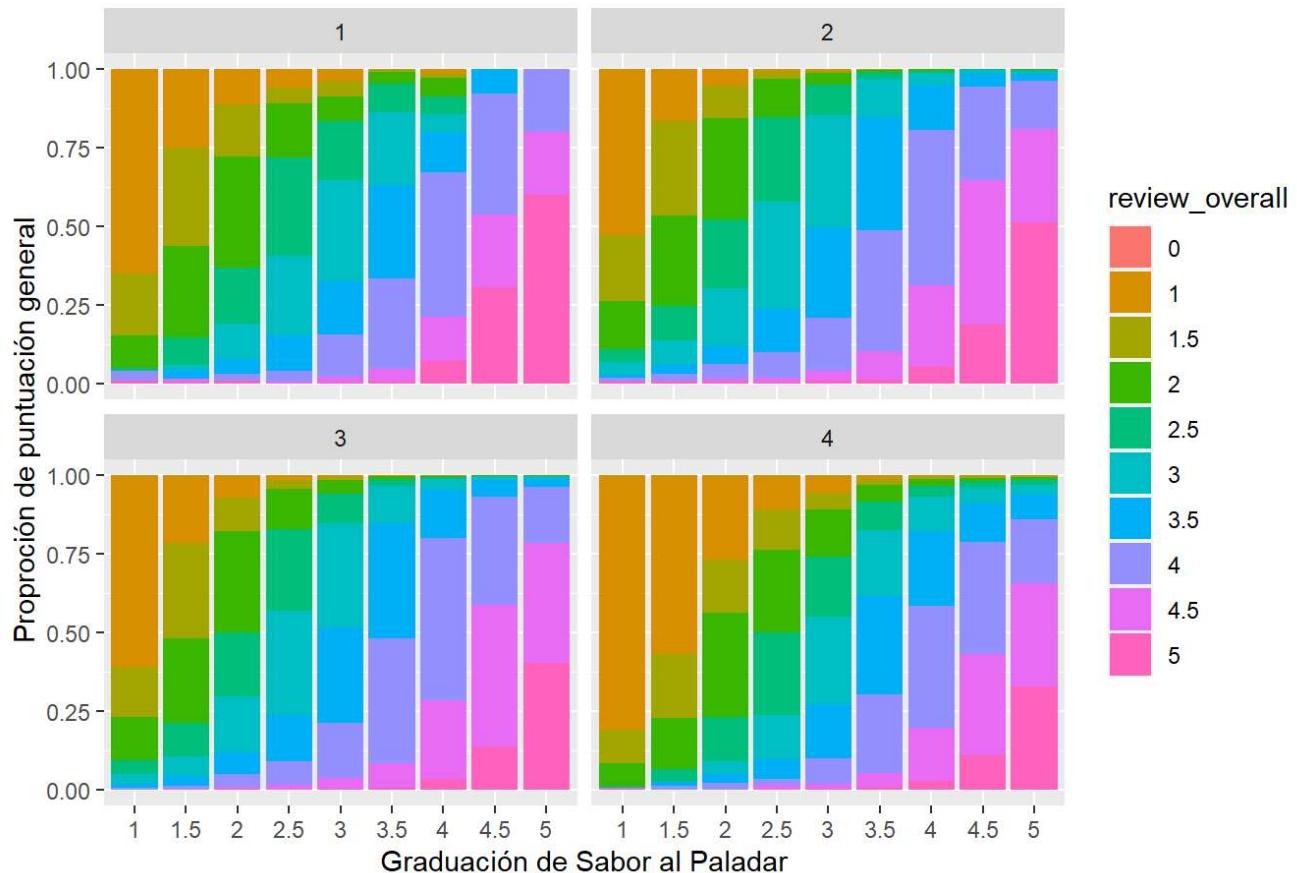
```
ggplot(data = Datos_Beer[1:filas,],aes(x=review_appearance,fill=review_overall))+geom_bar(position="fill")+facet_wrap(~beer_abv)+facet_wrap(~beer_abv)+ylab("Proporción de puntuación general") +xlab("Graduación de Apariencia") +ggtitle("Puntuación de las cervezas según su Graduación de Alcohol y su Apariencia")
```

## Puntuación de las cervezas según su Graduación de Alcohol y su Apariencia



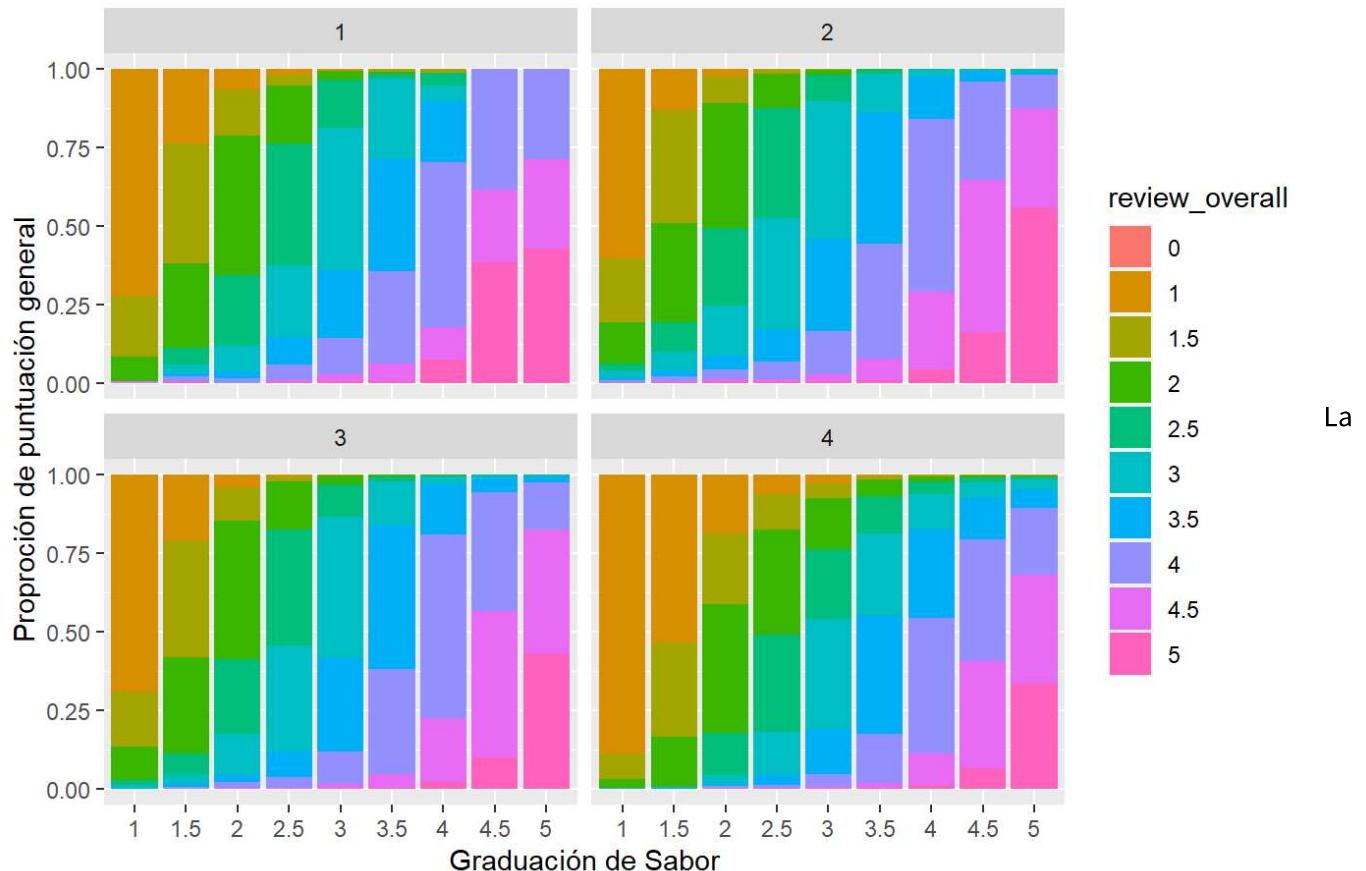
```
ggplot(data = Datos_Beer[1:filas,],aes(x=review_palate,fill=review_overall))+geom_bar(po
sition="fill")+facet_wrap(~beer_abv)+facet_wrap(~beer_abv)+ylab("Proporción de puntuació
n general") +xlab("Graduación de Sabor al Paladar") +ggtitle("Puntuación de las cervezas s
egún su Graduación de Alcohol y su Sabor Paladar")
```

## Puntuación de las cervezas según su Graduación de Alcohol y su Sabor Paladar



```
ggplot(data = Datos_Beer[1:filas,],aes(x=review_taste,fill=review_overall))+geom_bar(position="fill")+facet_wrap(~beer_abv)+facet_wrap(~beer_abv)+ylab("Proporción de puntuación general") +xlab("Graduación de Sabor")+ggtitle("Puntuación de las cervezas según su Graduación de Alcohol y su Sabor")
```

## Puntuación de las cervezas según su Graduación de Alcohol y su Sabor



primera conclusión que se puede obtener a simple vista es que dejando a un lado la graduación de alcohol, las intensidades mas altas (de aroma, apariencia, sabor del paladar y sabor) suelen tener mejor puntuación.

Analizando cada valoración por separado, vemos que el aroma tiene una mejor puntuación en cervezas sin alcohol y el grueso de valoraciones bajas las tiene las cervezas con una graduación de nivel 4 y en el aroma mas bajo.

Respecto a la apariencia, a nivel de mejores puntuaciones (un 5) todas están mas o menos similares, no obstante, si seleccionamos las 3 puntuaciones mas altas (4, 4.5 y 5) las cervezas con una graduación de alcohol de nivel 1 son las peores valoradas y las de los niveles 2 y 3 mejores. La peor valoración como se ve de una manera clara la tiene las cervezas de nivel 2 y una apariencia con puntuación más baja.

Al igual que en el aroma, las cervezas sin alcohol (tipo 1) son las mejores valoradas (con un 5) con las intensidades respecto al sabor del paladar y el grueso de valoraciones bajas las tiene las cervezas con una graduación de nivel 4 y en el aroma más bajo.

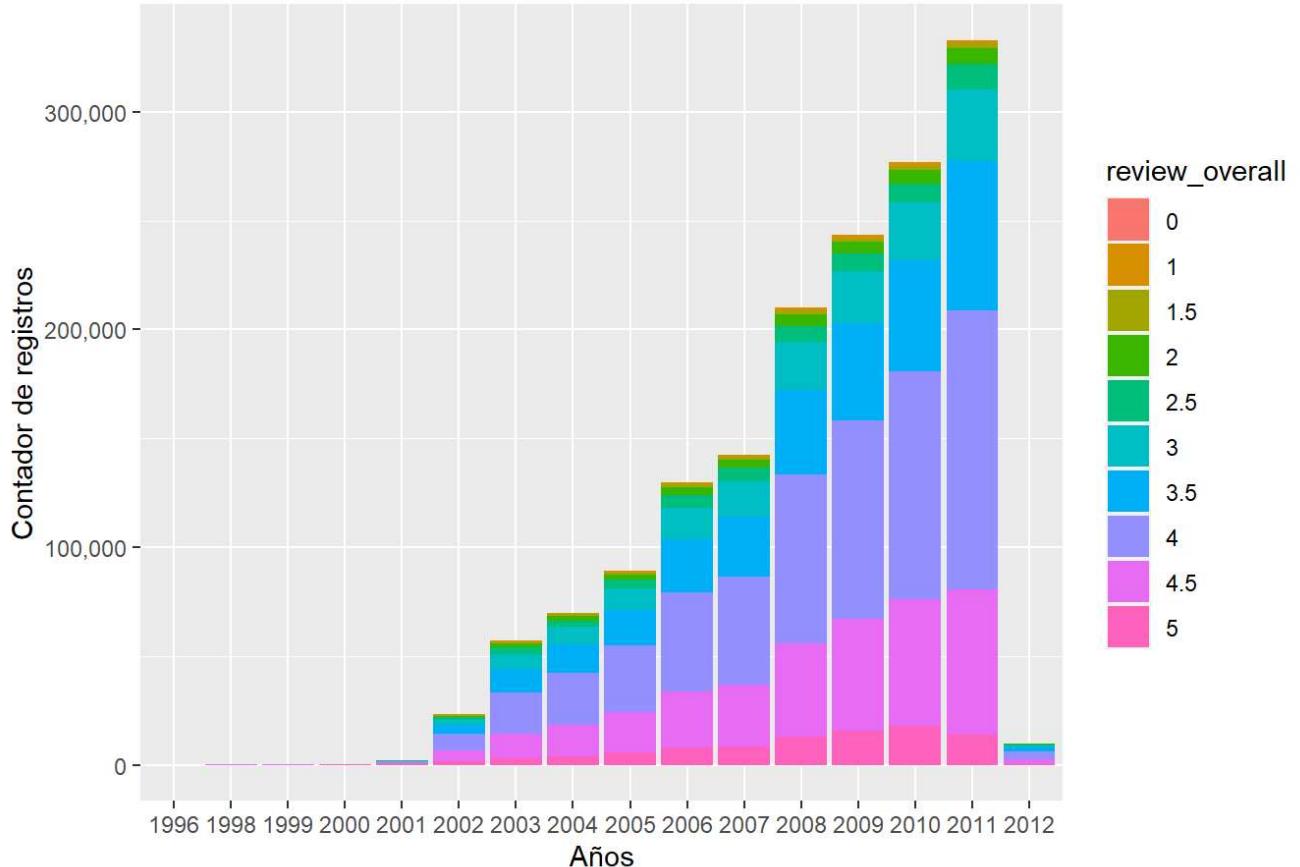
Finalmente, en lo relativo al sabor, nos damos cuenta de que las cervezas con un sabor mas fuerte (4, 4.5 y 5) son las mejores valoradas, y se nota un crecimiento en la puntuación considerable respecto a un sabor con puntuaciones más bajas. El grueso de las peores valoraciones son las que tienen una graduación de 4.

Una vez terminado de comparar la puntuaciones de las review con la graduación de la cerveza, se va a proceder a analizar las fechas en que se han realizado las review con las puntuaciones que se han dado.

#Comparamos La valoracion con Las fechas en que se ha creado cada review.

```
ggplot(data=Datos_Beer[1:filas,],aes(x=year,fill=review_overall))+geom_bar()+ylab("Contador de registros")+xlab("Años") +ggttitle("Puntuación de las cervezas según el año") +scale_y_continuous(labels = scales::comma)
```

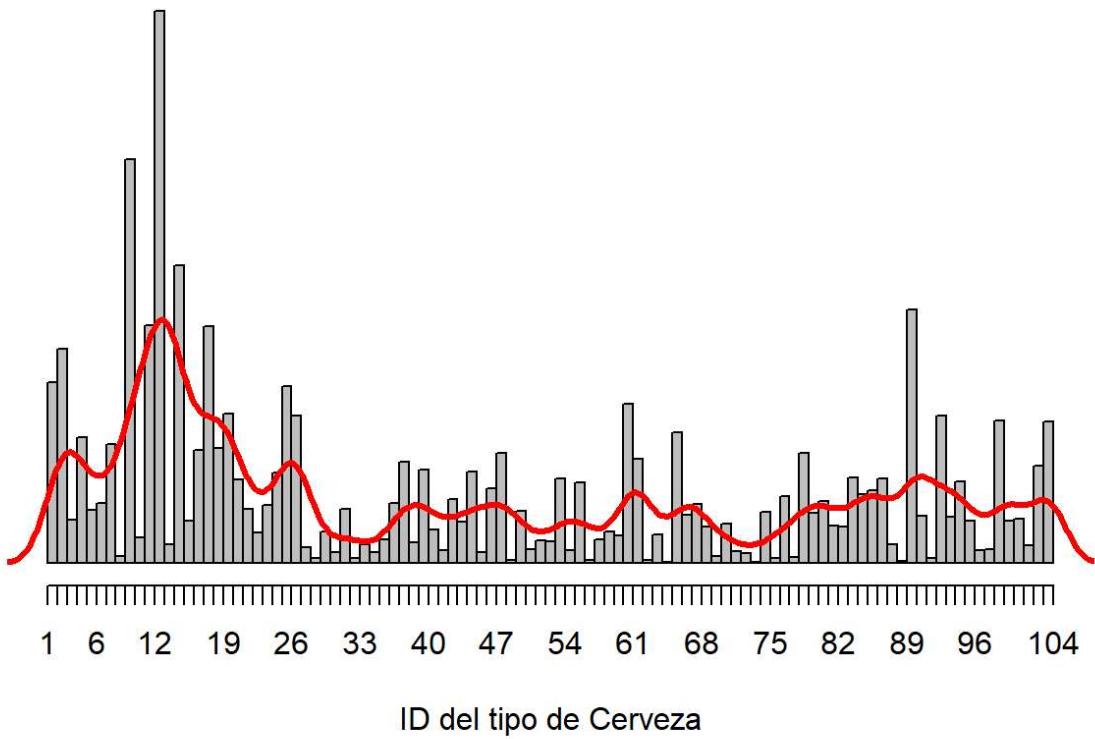
## Puntuación de las cervezas según el año



Aunque la salida del grafico se podía intuir, al verlo se demuestra que las puntuaciones de valoraciones son proporcionales (en mayor o menor medida) al número de encuestas realizadas. Además, el grafico se nos muestra que el año 2011 fue el año donde mas encuestas se hicieron y el 2001 el año en que menos.

```
#Comparamos La probabilidad de Los diferentes tipos de cervezas y obtenemos su densidad
hist(Datos_Beer$beer_style_ID, breaks = 104, probability = TRUE, col = "grey", axes = FALSE,
      main = "", xlab = "ID del tipo de Cerveza", ylab = "", xaxp= c(1, 104, 104))
axis(1,at = seq(1, 104, by = 1))

# Densidad
lines(density(Datos_Beer$beer_style_ID), lwd = 3, col = "red")
```



Como se puede observar, los tipos de cerveza 12, 9, 14, 89, 17 y 2 (se ha usado la función zm() para ver los numero de una manera fácil) son los que más encuestas tienen.

```
#Creamos un DataFrame con los 6 tipos de cerveza que tienen más valoraciones.
Datos_Beer_Mejor_Tipo <- Datos_Beer[(Datos_Beer$beer_style_ID == 12 | Datos_Beer$beer_style_ID == 9 | Datos_Beer$beer_style_ID == 14 | Datos_Beer$beer_style_ID == 89 | Datos_Beer$beer_style_ID == 17 | Datos_Beer$beer_style_ID == 2 ),]

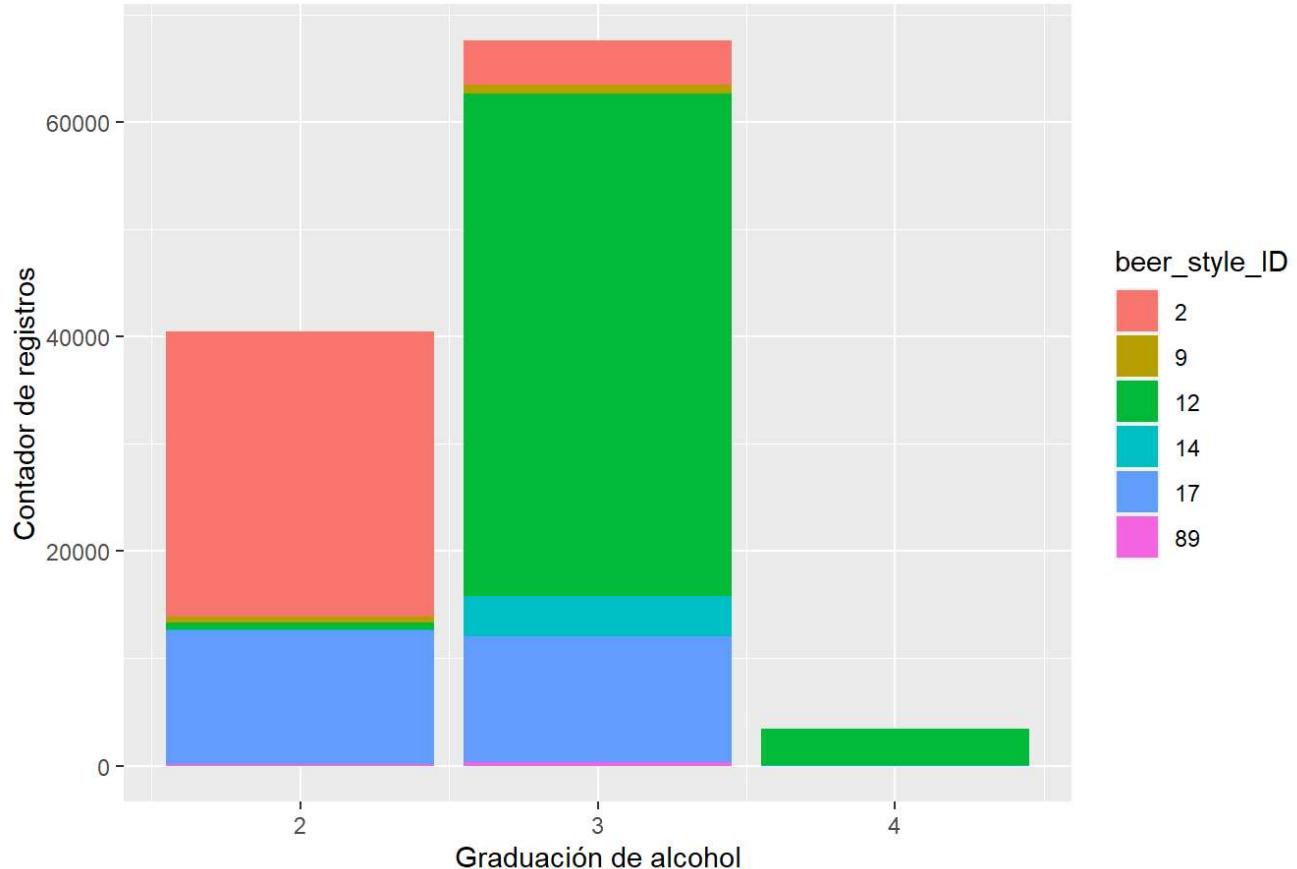
# Discretizamos beer_style_ID
cols<-c("beer_style_ID")

for (i in cols){
  Datos_Beer_Mejor_Tipo[,i] <- as.factor(Datos_Beer_Mejor_Tipo[,i])
}

##Comparamos Los parametros de La cerveza con Los 6 tipos de cerveza que tienen más valoraciones.##

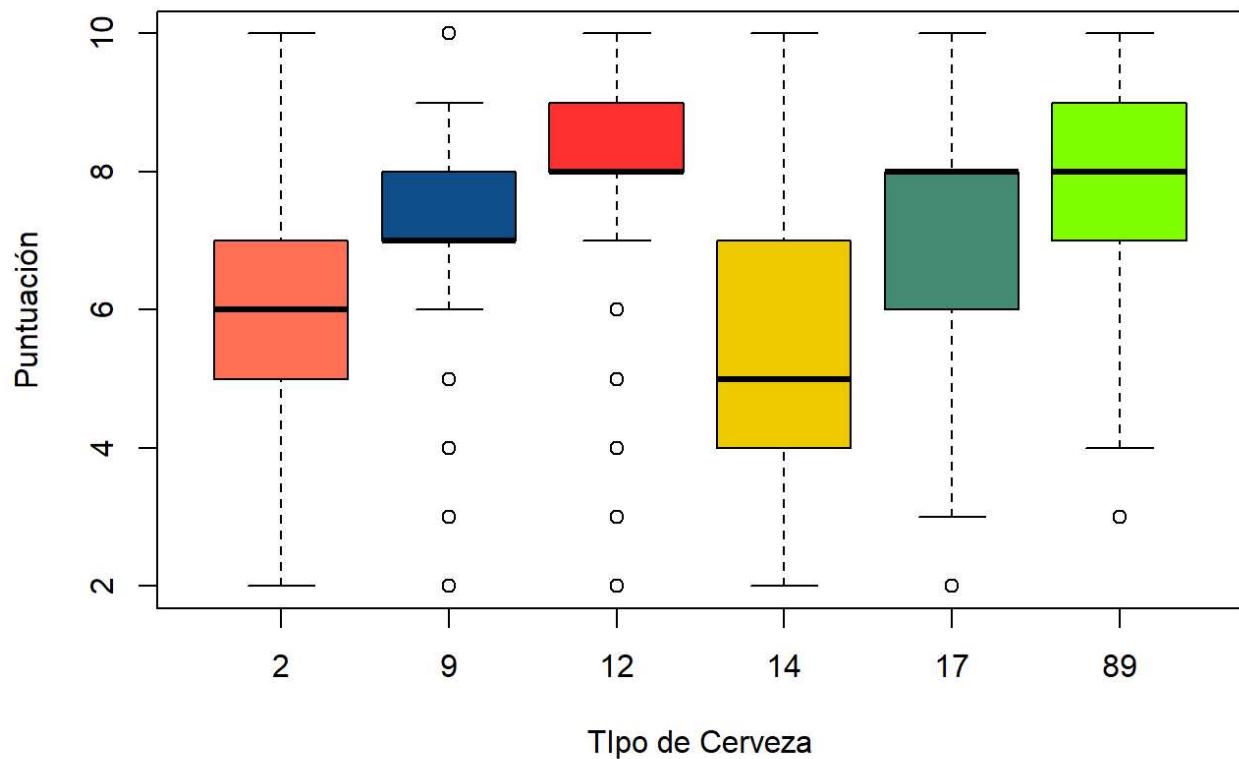
#Comparamos el grado de alcohol de la cerveza (beer_abv) con el tipo de cerveza
ggplot(data=Datos_Beer_Mejor_Tipo[1:filas,],aes(x=beer_abv,fill=beer_style_ID))+geom_bar() +ylab("Contador de registros") +xlab("Graduación de alcohol") +ggtitle("Tipo cervezas según su graduación de alcohol")
```

### Tipo cervezas según su graduación de alcohol



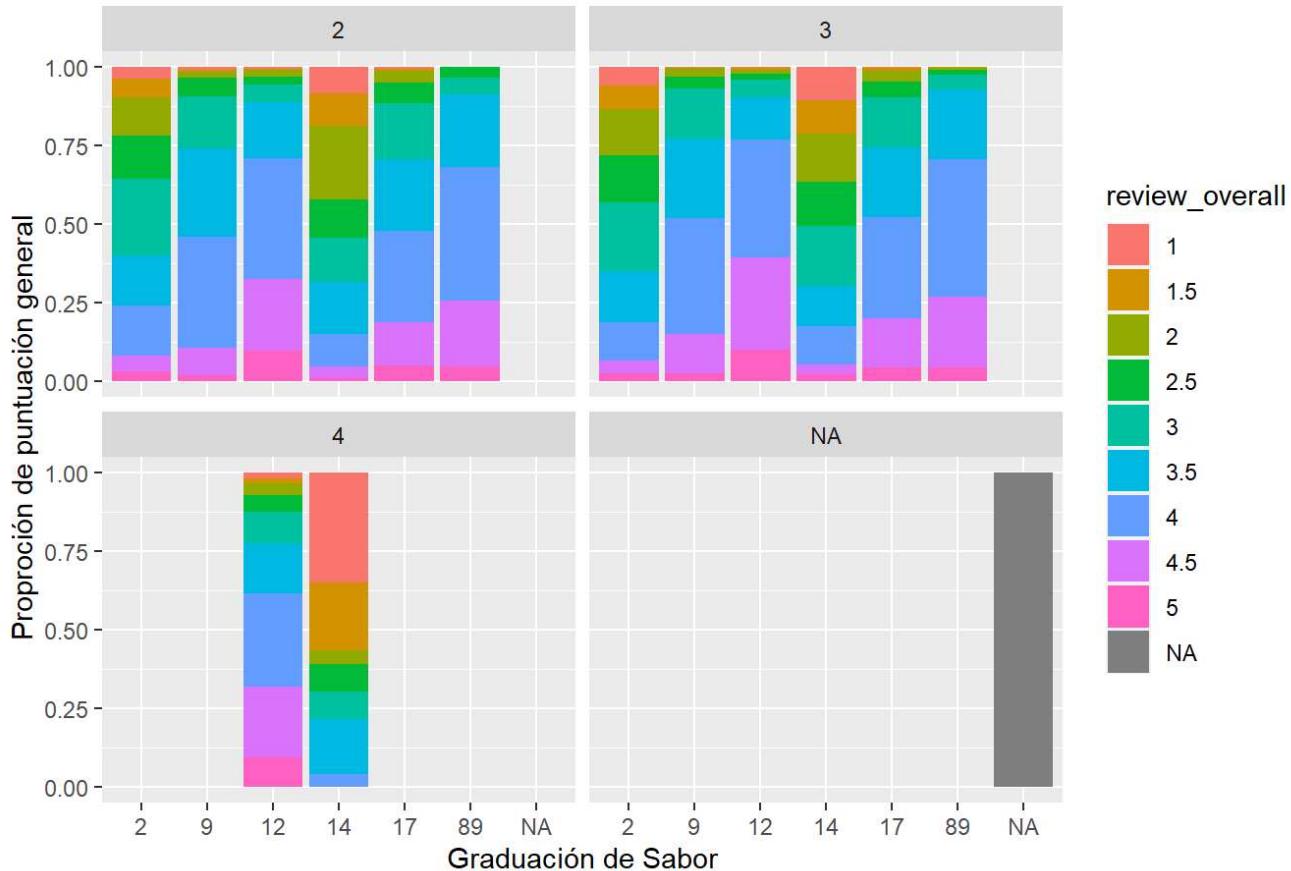
```
#Comparamos La puntuación con el tipo de cerveza
boxplot (review_overall ~ beer_style_ID, Datos_Beer_Mejor_Tipo,
main = "Distribución de puntuación por tipo de cerveza", xlab = "TIpo de Cerveza", ylab
= "Puntuación",
col = c("corall1", "dodgerblue4", "firebrick1", "gold2","aquamarine4","chartreuse"))
```

## Distribución de puntuación por tipo de cerveza



```
#Comparamos La puntuación de Las cervezas según su Graduación de Alcohol y su Tipo  
ggplot(data = Datos_Beer_Mejor_Tipo[1:filas,],aes(x=beer_style_ID,fill=review_overall))+  
geom_bar(position="fill")+facet_wrap(~beer_abv)+ylab("Proporción de puntuación general")  
+xlab("Graduación de Sabor")+ggtitle("Puntuación de las cervezas según su Graduación de  
Alcohol y su Tipo")
```

## Puntuación de las cervezas según su Graduación de Alcohol y su Tipo



Como se puede comprobar en el primer gráfico, el grueso de la graduación de alcohol principal de los tipos de cervezas elegidos es el 3. Comparando con el grafico en que se incluyen todos los tipos, es un resultado bastante similar. La única cosa diferente, es que no tenemos en esta simplificación de tipos alcohol de tipo 0.

El segundo gráfico, nos muestra que los tipos de cerveza 12 y 89 suelen tener una puntuación mas alta que los otros cuatro. Esta vez comparando con el grafico con la densidad, el numero 12 es el tipo de cerveza con mas valoraciones y con esta grafica nos damos cuenta de que también es el tipo con mejores valoraciones.

En el ultimo gráfico, observamos la puntuación de las cervezas, con su graduación y su tipo. Podemos observar que solamente las del tipo 12 y 14 tienen una graduación mas alta. Y se confirma que el tipo 12 tiene una puntuación mas alta en todas sus graduaciones de alcohol.

## 3.7 Conclusión

En este estudio el objetivo ha sido predecir el tipo de gustos que tiene la gente en relación con la cerveza, las encuesta se ha realizado desde el año 1996 y 2012, y el año con mayores valoraciones fue el 2011, no obstante, el número de proporción de puntuaciones respecto al año ha sido de una manera más o menos proporcional.

La primera conclusión que sacamos es que a las personas le gustan las cervezas con un grado de alcohol entre 5 y 15 ° mayoritariamente. Aunque también hay un gran porcentaje que prefiere una graduación un poco mas baja (entre 1 y 5°).

A nivel de valoraciones de las cervezas, mientras tengan un sabor, un gusto al paladar, una apariencia y un aroma más fuertes, son mejores recibidas y mejores puntuadas, respecto a sus homólogos más bajos.

A nivel de tipos de cerveza, la mas valorada a nivel de encuestas ha sido la “American Double / Imperial Stout” (Tipo 12) como ganadora en todos sus rangos de graduación de alcohol. En segunda posición la “Roggenbier” (Tipo 89) en sus dos tipos de graduación (2 y 3).

# 4 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%