Tipologia y Ciclo de vida de los Datos. Practica 2

Autores: Alexis Arroyo y Gabriel Pulido de Torres

Enero 2020

Contents

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

Hemos optado por analizar el dataset de datos del titanic.

En este conjunto de datos tenemos información acerca de los pasajeros que iban en el Titanic, naufragado en 1912 y en el que murieron 1500 personas. Además de información descriptiva del tipo de pasajero tenemos también el indicador de si sobrevivió al hundimiento o no. El objetivo es decidir si las variables aportadas son suficientes para crear un modelo predictivo que prediga la supervivencia o no de un pasajero.

Realizamos un primer contacto con el conjunto de datos, visualizando su estructura.

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

# Cargamos el fichero de datos
data <- read.csv('train.csv',stringsAsFactors = FALSE, na.strings = "")
dim.data.frame(data)
```

```
## [1] 891  12
```

```
# Verificamos la estructura del conjunto de datos
str(data)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. L
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

El conjunto de datos incluye 12 variables y 891 observaciones.

Descripción del conjunto de datos:

- Passengerld: Contador de pasajeros del 1 al 891.
- Survived: esta variable toma dos valores e indica si el pasajero sobrevivió. (0= No, 1= Si).
- pClass: clase del ticket. 1=1st (clase alta), 2=2nd (clase media) y 3=3rd (clase baja).
- Name: nombre completo del pasajero.
- Sex: sexo del pasajero (Female o Male).
- Age: edad del pasajero
- SibSp: número de hermanos/hermanas, hermanastros/hermanastros y marido o esposa del pasajero que también iban a bordo.

- Parch: número de hijas, hijos, padre y madre del pasajero a bordo del Titanic.
- Ticket: El número del ticket del pasajero.
- Fare: Es la tarifa del pasajero en dólares.
- Cabin: Código identificativo de la cabina.
- Embarked: el puerto en el que embarcó el pasajero (C = Cherbourg, Q =Queenstown, S = Southampton).

1.1 Objetivo

El objetivo es decidir si las variables aportadas son suficientes para crear un modelo predictivo que prediga la supervivencia o no de un pasajero.

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

Hemos leido los datos en un dataframe (al que llamamos "data"). Vemos su dimensionalidad, los nombres y tipos de las variables:

```
dim.data.frame(data)
```

```
## [1] 891  12
```

```
sapply(data, function(x) class(x))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##   "integer"   "integer"   "integer" "character" "character"   "numeric"
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##   "integer"   "integer" "character"   "numeric" "character" "character"
```

Aprovechamos para comprobar si hay registros duplicados usando el comando unique()

```
data_unique <- unique(data)
dim.data.frame(data_unique)
```

```
## [1] 891  12
```

```
remove(data_unique)
```

Se comprueba que la dimensionalidad ha quedado exactamente igual que cuando cargamos el dataset, por lo tanto, no hay registros duplicados (nos referimos a completos duplicados).

Mostramos una muestra de los primeros registros de nuestro dataframe: TODO#########

Hay algunas variables que carecen de interés por ser identificativas de cada registro. Se tratan de "PassengerId", "Ticket" y "Name". No nos interesa tener unívocamente identificado cada caso para realizar ningún tipo de análisis, no aportan nada.

2.1 PersonasTicket

Antes de eliminar la variable Ticket, nos va a servir para conocer el precio unitario que han pagado los pasajeros, ya que la variable Fare es la tarifa pagada en el ticket, pero dentro del mismo ticket pueden estar incluidas varias personas. Inicialmente vamos a obtener el número de personas que están incluidas en un mismo ticket.

```
nrow(table(data$Ticket, data$Fare))
```

```
## [1] 681
```

```
length(unique(data$Ticket))
```

```
## [1] 681
```

```
data %>% group_by(Ticket, Fare) %>% filter(row_number() == 1)
```

PassengerId	Survived	Pclass	Name
1	0	3	Braund, Mr. Owen Harris
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)
3	1	3	Heikkinen, Miss. Laina
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)
5	0	3	Allen, Mr. William Henry
6	0	3	Moran, Mr. James
7	0	1	McCarthy, Mr. Timothy J
8	0	3	Palsson, Master. Gosta Leonard
9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)
10	1	2	Nasser, Mrs. Nicholas (Adele Achem)
11	1	3	Sandstrom, Miss. Marguerite Rut
12	1	1	Bonnell, Miss. Elizabeth
13	0	3	Saundercock, Mr. William Henry
14	0	3	Andersson, Mr. Anders Johan
15	0	3	Vestrom, Miss. Hulda Amanda Adolfina
16	1	2	Hewlett, Mrs. (Mary D Kingcome)
17	0	3	Rice, Master. Eugene
18	1	2	Williams, Mr. Charles Eugene
19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)
20	1	3	Masselmani, Mrs. Fatima
21	0	2	Fynney, Mr. Joseph J
22	1	2	Beesley, Mr. Lawrence
23	1	3	McGowan, Miss. Anna "Annie"
24	1	1	Sloper, Mr. William Thompson
26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)
27	0	3	Emir, Mr. Farred Chehab
28	0	1	Fortune, Mr. Charles Alexander
29	1	3	O'Dwyer, Miss. Ellen "Nellie"
30	0	3	Todoroff, Mr. Lalio
31	0	1	Uruchurtu, Don. Manuel E
32	1	1	Spencer, Mrs. William Augustus (Marie Eugenie)
33	1	3	Glynn, Miss. Mary Agatha
34	0	2	Wheadon, Mr. Edward H
35	0	1	Meyer, Mr. Edgar Joseph
36	0	1	Holverson, Mr. Alexander Oskar
37	1	3	Mamee, Mr. Hanna
38	0	3	Cann, Mr. Ernest Charles
39	0	3	Vander Planke, Miss. Augusta Maria
40	1	3	Nicola-Yarred, Miss. Jamila
41	0	3	Ahlin, Mrs. Johan (Johanna Persdotter Larsson)
42	0	2	Turpin, Mrs. William John Robert (Dorothy Ann Wonnacott)
43	0	3	Kraeff, Mr. Theodor
44	1	2	Laroche, Miss. Simonne Marie Anne Andree
45	1	3	Devaney, Miss. Margaret Delia
46	0	3	Rogers, Mr. William John
47	0	3	Lennon, Mr. Denis
48	1	3	O'Driscoll, Miss. Bridget
49	0	3	Samaan, Mr. Youssef
50	0	3	Arnold-Franchi, Mrs. Josef (Josefine Franchi)
51	0	3	Panula, Master. Juha Niilo
52	0	3	Nosworthy, Mr. Richard Cater
53	1	1	Harper, Mrs. Henry Sleeper (Myna Haxtun)
54	1	2	Faunthorpe, Mrs. Lizzie (Elizabeth Anne Wilkinson)
55	0	1	Ostby, Mr. Engelhart Cornelius
56	1	1	Woolner, Mr. Hugh
57	1	2	Rugg, Miss. Emily
58	0	3	Novel, Mr. Mansouer

Podemos obtener el precio por persona, dividiendo Fare / integrantes del ticket:

```
ticket_personas <- as.data.frame(data %>%
                                 group_by(Ticket) %>%
                                 dplyr::summarize(PersonasTicket=n()))
```

Creamos un nuevo dataframe (llamado "ticket_personas") con las variables Ticket y PersonasTicket

```
df_status(ticket_personas)
```

```
##          variable q_zeros p_zeros q_na p_na q_inf p_inf      type unique
## 1          Ticket       0       0    0    0     0     0 character    681
## 2 PersonasTicket       0       0    0    0     0     0   integer      7
```

Este nuevo dataframe lo combinamos con nuestro dataframe original (un join) a través del número de ticket (variable Ticket) para poder incorporar el número de personas al dataframe original. Después de juntarlos eliminamos el df ticket_personas pues no lo necesitaremos.

```
data <- merge(data, ticket_personas, by = "Ticket")
remove(ticket_personas)
```

Ahora tenemos en nuestro dataframe original "data" una nueva variable "PersonasTicket". Esa variable será utilizada para dividir la tarifa del ticket "Fare" entre esta nueva columna incorporada. Con ello obtenemos una nueva variable que le llamamos "Price" y es el precio por persona que se paga en el billete (obteniendo un precio por persona lineal por billete).

```
data$Price <- data$Fare / data$PersonasTicket
```

Ahora ya podemos eliminar todas las variables que no vamos a necesitar: PassengerId, Name, Ticket, PersonasTicket:

```
data <- select(data, -PassengerId, -Name, -Ticket, -PersonasTicket)
```

2.2 FamilySize

Revisando Parch (número de Padres / hijos a bordo) y SibSp (sibling: número de hermanos, hermanas, hermanastros y hermanastras del pasajero, spouse: Marido o mujer del pasajero en el titanic) vemos que las podemos agregar en una nueva variable FamilySize que será la suma de estas dos variables mas el propio viajero en cuestion. Por ello esta variable como mínimo valdra uno.

```
data$FamilySize = data$SibSp + data$Parch + 1
```

2.3 Revisión y conversion de tipos

Tenemos algunas variables que, aunque a priori aparecen como "numeric" o "character" deberíamos convertir a "factor". Estas variables son: "Survived", "Pclass", "Embarked" y "Sex". Tienen un número finito de valores y aunque puedan ser numéricas, ese número no nos aporta información con lo que los discretizamos convirtiendolos en factores:

```
data$Sex <- as.factor(data$Sex)
data$Survived <- as.factor(data$Survived)
data$Pclass <- as.factor(data$Pclass)
data$Embarked <- as.factor(data$Embarked)
```

La variable Cabin es de tipo character, pero como vamos a eliminarla no haremos ninguna conversión.

Finalmente nos quedarían como variables numéricas: Age, SubSp, Parch, FamilySize, Fare y Price

```
summary(data)
```

```
##  Survived Pclass      Sex           Age             SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  1:342    2:184   male  :577   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##           3:491                 Median :28.00   Median :0.000   Median :0.0000
##                                 Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                                 3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                                 Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                                 NA's   :177
##       Fare           Cabin           Embarked       Price
##  Min.   :  0.00   Length:891         C   :168   Min.   :  0.000
##  1st Qu.:  7.91   Class :character   Q   : 77   1st Qu.:  7.763
##  Median : 14.45   Mode  :character   S   :644   Median :  8.850
##  Mean   : 32.20                      NA's:  2   Mean   : 17.789
##  3rd Qu.: 31.00                                 3rd Qu.: 24.288
##  Max.   :512.33                                 Max.   :221.779
##
##    FamilySize
##  Min.   : 1.000
##  1st Qu.: 1.000
##  Median : 1.000
##  Mean   : 1.905
##  3rd Qu.: 2.000
##  Max.   :11.000
##
```

3 Limpieza de los datos.

Analizamos los valores nulos y vacíos. Para ello nos valemos de la salida de la función df_status, que no s muestra un resumen del estado de nuestras 8 variables actuales:

```
df_status(data)
```

```
##       variable q_zeros p_zeros q_na  p_na q_inf p_inf      type unique
## 1     Survived     549   61.62    0  0.00     0     0    factor      2
## 2       Pclass       0    0.00    0  0.00     0     0    factor      3
## 3          Sex       0    0.00    0  0.00     0     0    factor      2
## 4          Age       0    0.00  177 19.87     0     0   numeric     88
## 5        SibSp     608   68.24    0  0.00     0     0   integer      7
## 6        Parch     678   76.09    0  0.00     0     0   integer      7
## 7         Fare      15    1.68    0  0.00     0     0   numeric    248
## 8        Cabin       0    0.00  687 77.10     0     0 character    147
## 9     Embarked       0    0.00    2  0.22     0     0    factor      3
## 10       Price      15    1.68    0  0.00     0     0   numeric    248
## 11 FamilySize       0    0.00    0  0.00     0     0   numeric      9
```

3.1 Elementos nulos y ceros.

3.1.1 Elementos nulos

```
data %>% group_by(Embarked) %>% count(Embarked)
```

Embarked	n
C	168
Q	77
S	644
NA	2

En la tabla se observa que el campo Age tiene un 19.87% de nulos, Cabin más de un 77% y dos nulos en Embarked

3.1.1.1 Embarked Solo hay 2 valores nulos, al ser muy pocos casos y además la variable solo toma 3 posibles valores, imputamos el valor que mas se repite, que es Embarked=S (Southampton)

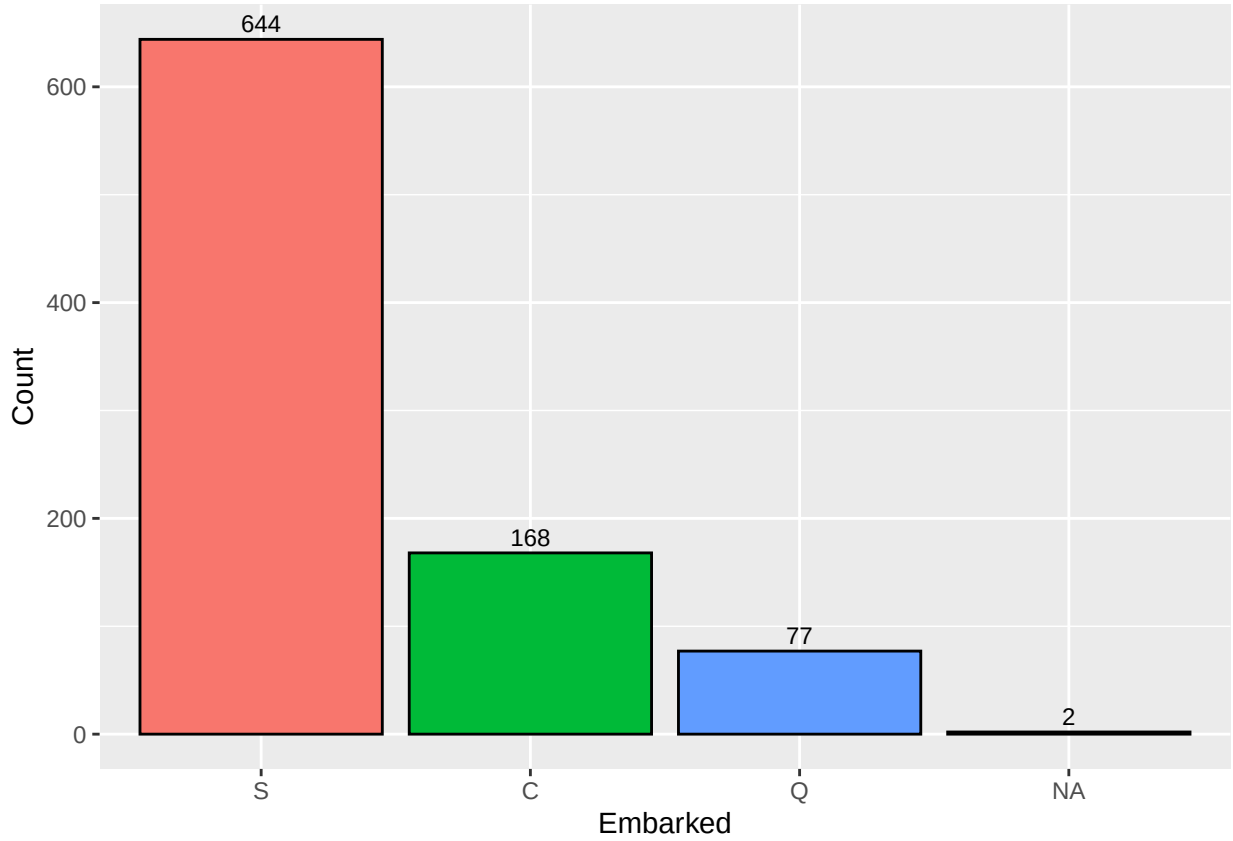
```
data %>% group_by(Embarked) %>% count(Embarked)
```

Embarked	n
C	168
Q	77
S	644
NA	2

```
data_Embarked <- sort(table(data$Embarked, useNA = "ifany"), decreasing = TRUE)
```

Gráficamente:

```
dat_plot <- as.data.frame(data_Embarked)
ggplot(dat_plot, aes(x=Var1, y=Freq, fill=Var1)) +
  geom_bar(stat = "identity", color = "black") +
  geom_text(aes(label=Freq), vjust = -0.4, color="black", size=3) +
  labs(x='Embarked', y='Count') +
  theme(legend.position = "none")
```



Asignamos el valor mas frecuente a los casos nulos:

```
data$Embarked[is.na(data$Embarked)] <- names(data_Embarked[1])
```

3.1.1.2 Cabin Dado que el número de nulos es muy elevado, un 77%, se opta por eliminar esta variable ya que tendríamos que imputar valores a una parte muy importante del dataset (con su consiguiente error). Además no parece que pueda ser una variable determinante para predecir la supervivencia.

```
data <- select(data, -Cabin)
```

3.1.1.3 Age Tiene mas del un 19% de calores nulos (concretamente 177 registros)

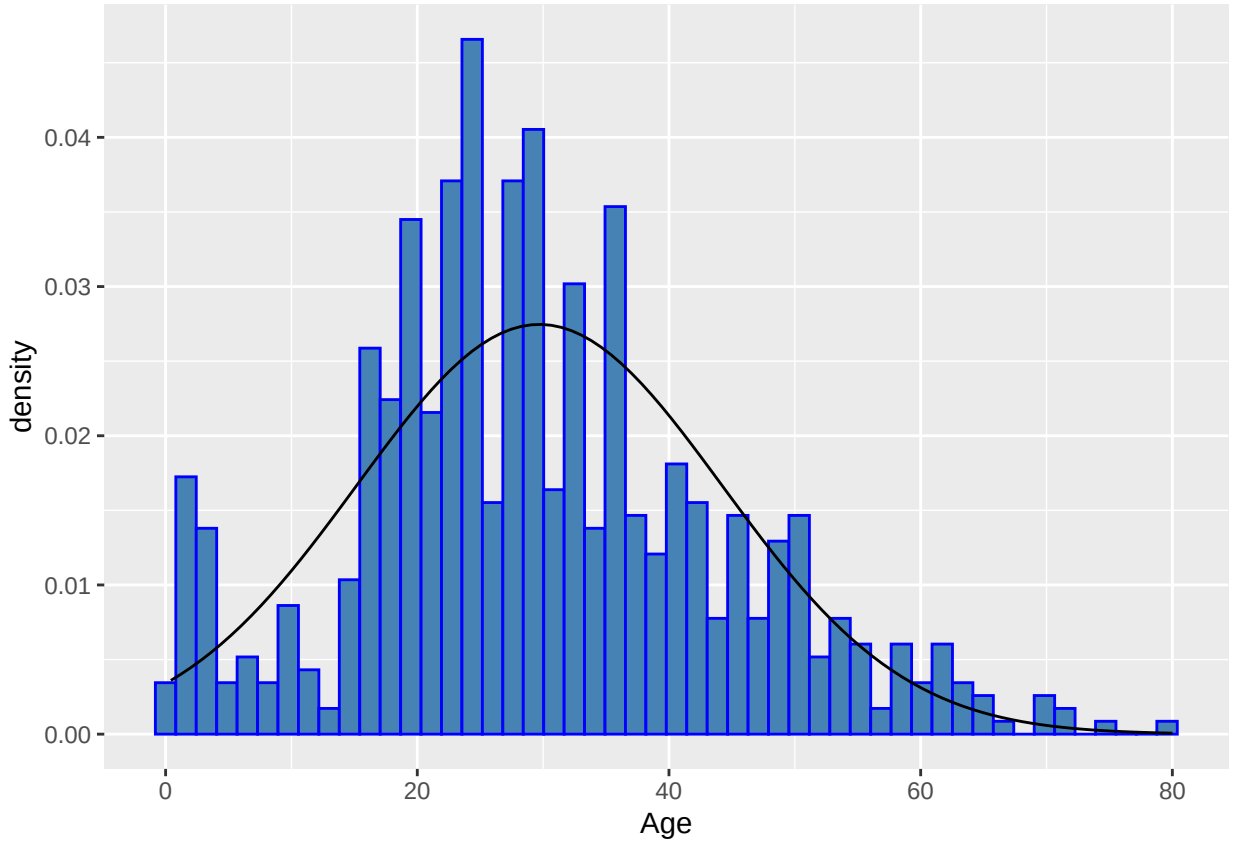
Una opción para solucionarlos sería calcular la media del resto de registros e imputarla, pero vamos a estudiar si podemos delimitar la media a aplicar para obtener un resultado más depurado.

Primero analizamos la distribución de la variable Age teniendo en cuenta sólo los registros donde hay valores (es decir algo mas del 80% del dataframe). Para ello creamos un nuevo dataset sin los registros nulos de Age

```
data_NoNA = data[which(!is.na(data$Age)),]
```

Observamos gráficamente como se distribuye la variable Age en este dataset:

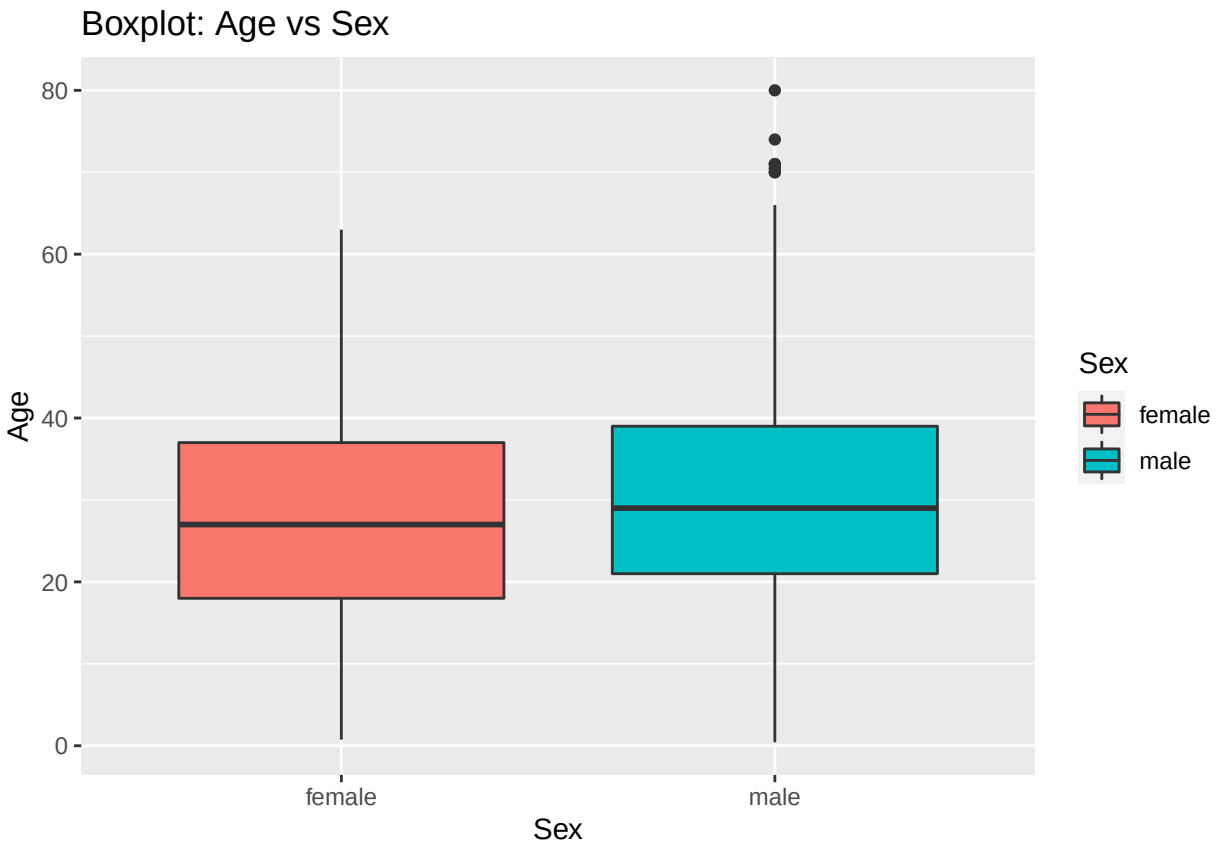
```
ggplot(data_NoNA, aes(Age))+
  geom_histogram(aes(y = ..density..), bins=50, fill="steelblue", color="blue") +
  stat_function(fun = dnorm, args = list(mean = mean(data_NoNA$Age), sd = sd(data_NoNA$Age)))
```



Comprobamos si hay alguna relación entre la edad alguna del resto de variables, para tenerlo en cuenta a la hora de imputar valores

3.1.1.4 Age vs Sex Comprobamos gráficamente la relación entre la edad y el género

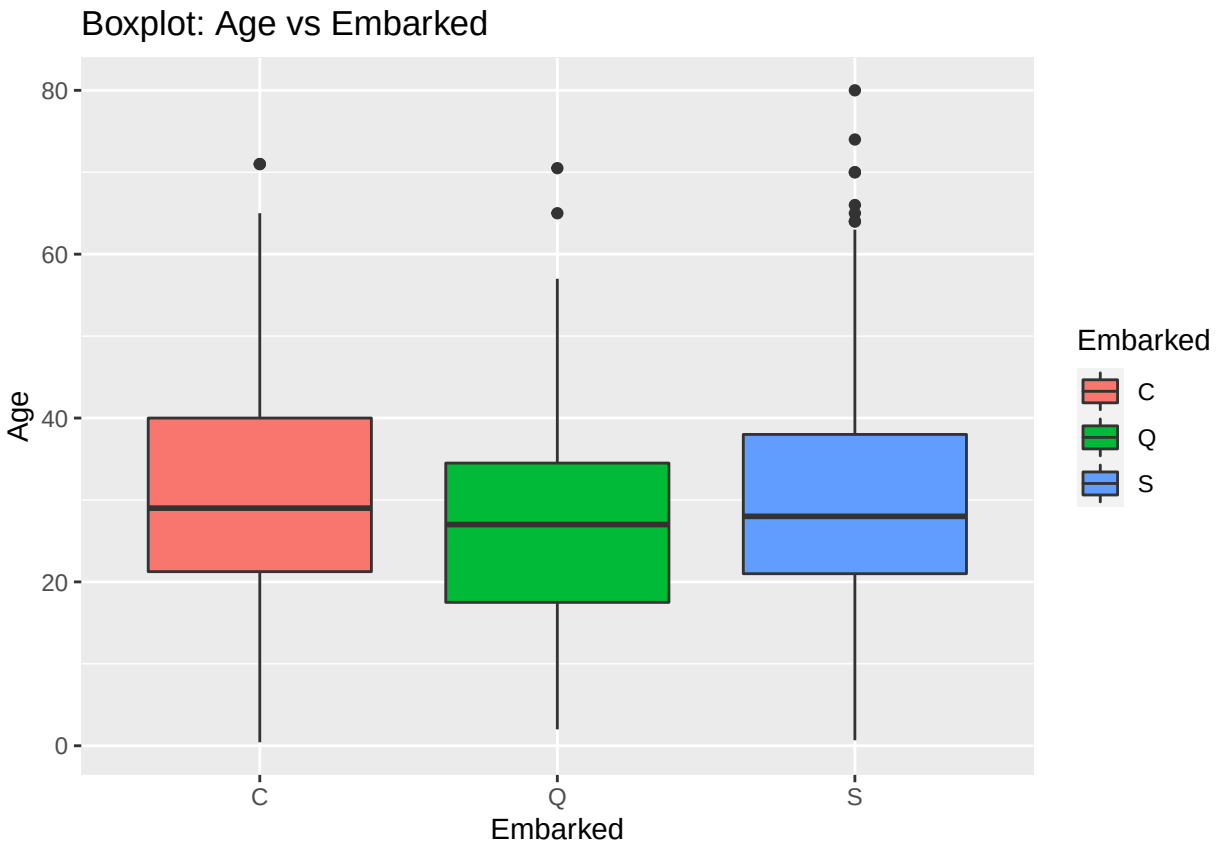
```
titulo <- 'Age vs Sex'
ggplot(data_NoNA, aes(y=Age, x=Sex, fill=Sex)) + geom_boxplot() + labs(title = paste0('Boxplot: ', titulo)) + ylab("Ag
```



No se aprecian casi diferencias de edad en función del género

3.1.1.5 Age vs Embarked Comprobamos gráficamente la relación entre la edad y el puerto de Embarque

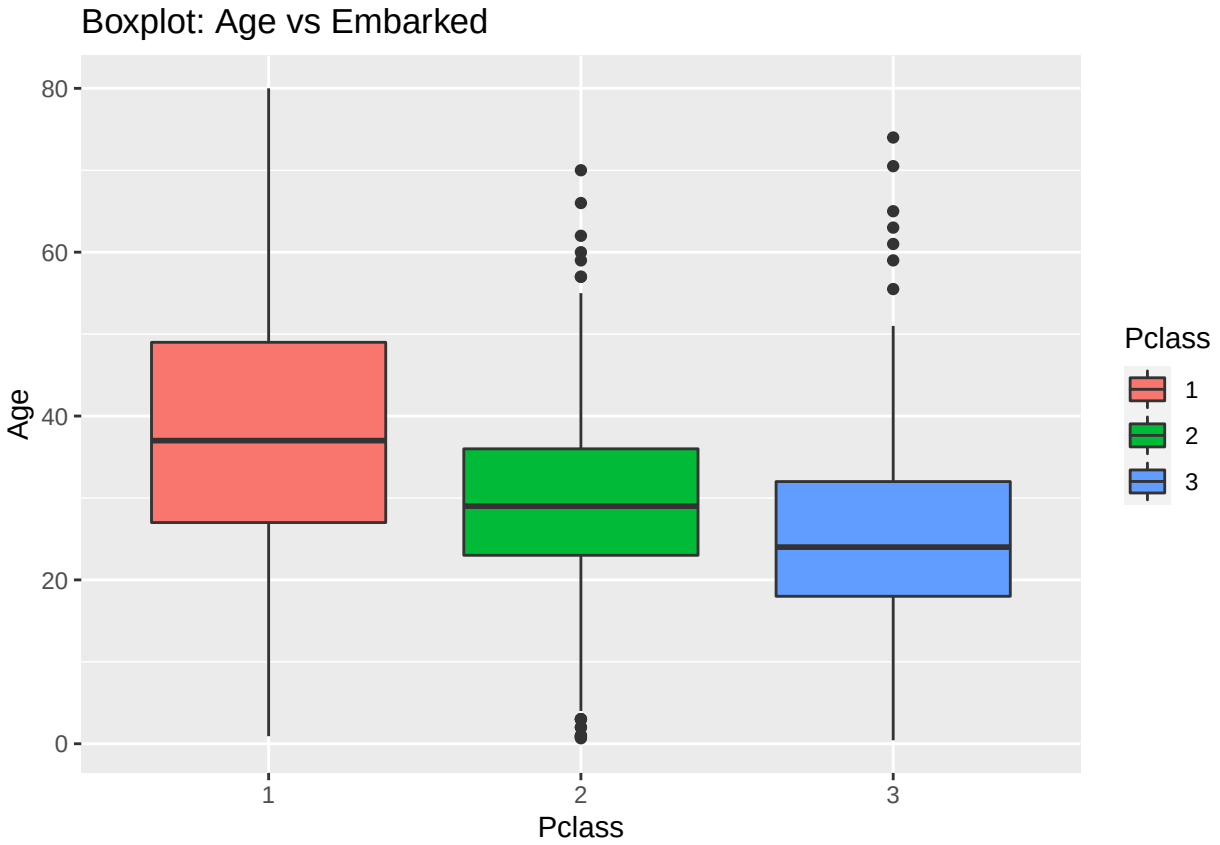
```
titulo <- 'Age vs Embarked'
ggplot(data_NoNA, aes(y=Age, x=Embarked, fill=Embarked)) +
  geom_boxplot() + labs(title = paste0('Boxplot: ', titulo)) + ylab("Age") + xlab("Embarked")
```



Tampoco se aprecian diferencias significativas en este caso con respecto a las medias de edad en cada una de las clases de Embarked.

3.1.1.6 Age vs Pclass Representamos visualmente la relación entre Age y Pclass:

```
titulo <- 'Age vs Embarked'
ggplot(data_NoNA, aes(y=Age, x=Pclass, fill=Pclass)) +
  geom_boxplot() + labs(title = paste0('Boxplot: ', titulo)) + ylab("Age") + xlab("Pclass")
```



En este caso si se observa una relación entre la edad y la clase en que viajaban los pasajeros: Los pasajeros de clase 1 (alta) tenían generalmente mayor edad que los de clase 2 (media) e igualmente sucede con los de clase 3 (baja).

3.1.1.7 Age vs Variables numéricas TODO######### código para la correlación

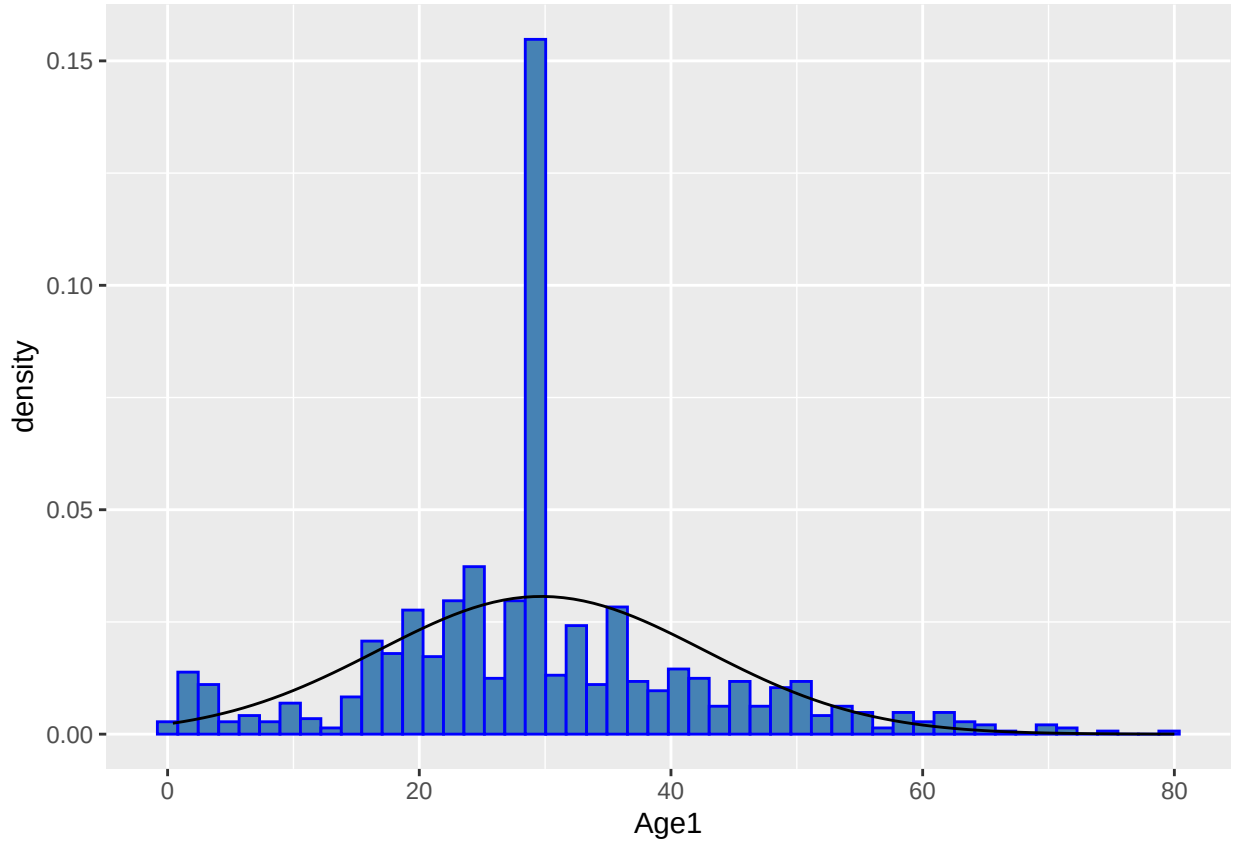
Se observa que existe correlación negativa con SibSp y con Parch (también con FamilySize, pero eso es completamente normal porque FamilySize es una transformación lineal de las otras 2). También hay correlación positiva con Price, pero entendemos que podemos tener en cuenta la parte familiar por el tema de esposa, hijos, hermanos, etc. puede tener efecto en la edad.

3.1.1.8 Imputación de valores a Age

Tras el análisis de las diversas relaciones entre las variables y Age vamos a realizar cuatro simulaciones diferentes de imputación y nos quedaremos con una sola:

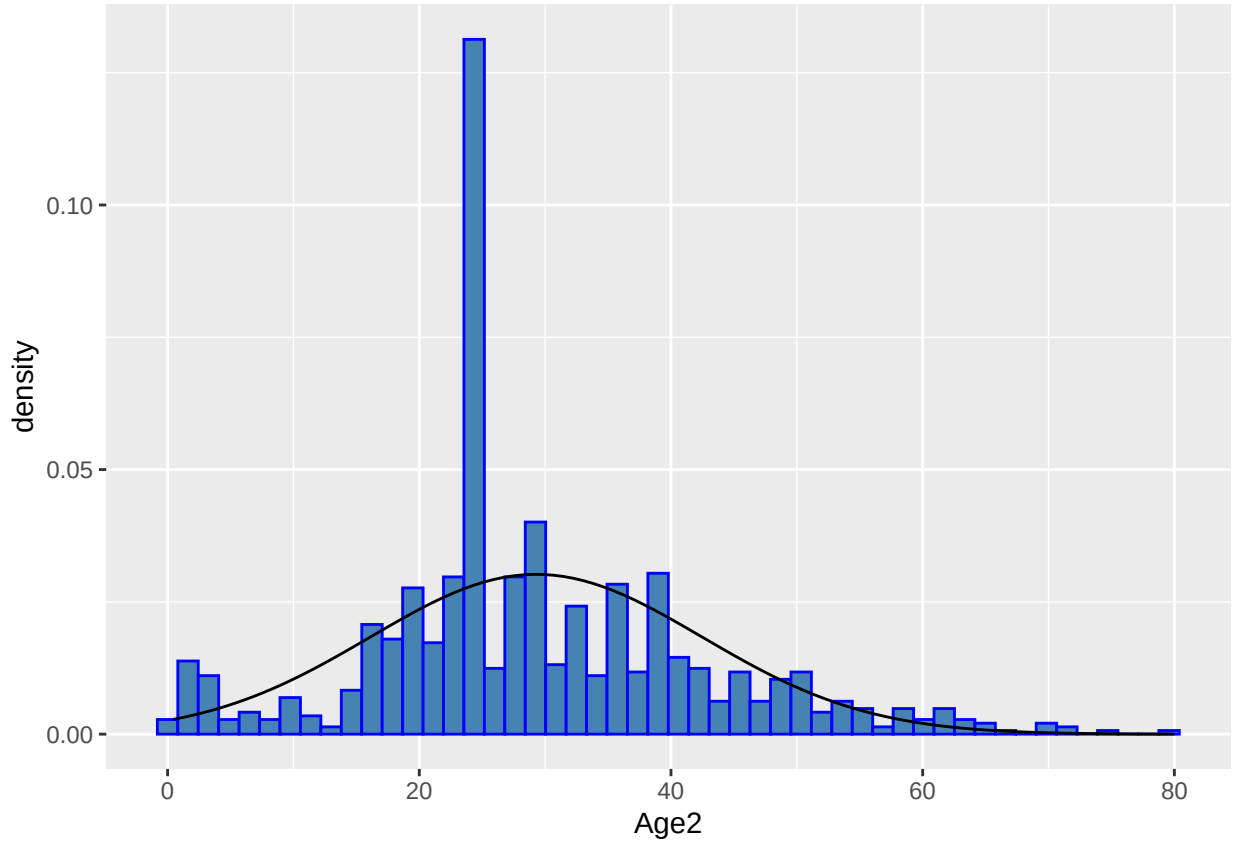
- Caso 1: Imputar la media de Age a todos los elementos faltantes

```
data$Age1 <- data$Age
data$Age1[is.na(data$Age1)] <- mean(data_NoNA$Age)

ggplot(data, aes(Age1)) +
  geom_histogram(aes(y = ..density..), bins=50, fill="steelblue", color="blue") +
  stat_function(fun = dnorm, args = list(mean = mean(data$Age1), sd = sd(data$Age1)))
```



- Caso 2: Imputar la media de Age pero por cada una de las clases(“Pclass”) ya que hemos visto que hay una relación entre ambas variables.

```
data$Age2 <- data$Age
data$Age2[is.na(data$Age2)&data$Pclass == 1] <- mean(data$Age2[!is.na(data$Age2)&data$Pclass == 1])
data$Age2[is.na(data$Age2)&data$Pclass == 2] <- mean(data$Age2[!is.na(data$Age2)&data$Pclass == 2])
data$Age2[is.na(data$Age2)&data$Pclass == 3] <- mean(data$Age2[!is.na(data$Age2)&data$Pclass == 3])

ggplot(data, aes(Age2)) +
  geom_histogram(aes(y = ..density..), bins=50, fill="steelblue", color="blue") +
  stat_function(fun = dnorm, args = list(mean = mean(data$Age2), sd = sd(data$Age2)))
```



- Caso 3: Imputar la datos en Age, pero teniendo en cuenta Pclass, Parch y SibSp, ya que hemos visto correlación entre las variables. Para este caso lo que hacemos es tener en cuenta las 3 variables y generar una agrupación de datos calculando la media para las combinaciones (recordemos que son variables discretas). Una vez que obtenemos esos valores medios, los imputamos. Luego verificamos si quedó algún valor sin imputar (que serán muy pocos) y para esos pocos casos faltantes, imputar los valores por cada clase.

```
medias_clase_fam <- as.data.frame(data_NoNA %>% group_by(Pclass, SibSp, Parch) %>% dplyr::summarize(Media_cla
```

Hacemos un meMerge entre nuestro dataset original y el de Medias (por Pclass, SIbSp y Parch) a través de las columnas usadas en la agrupación. El merge e sun LEFT JOIN ya que puede que no existan todas las combinaciones en medias_clase_fam

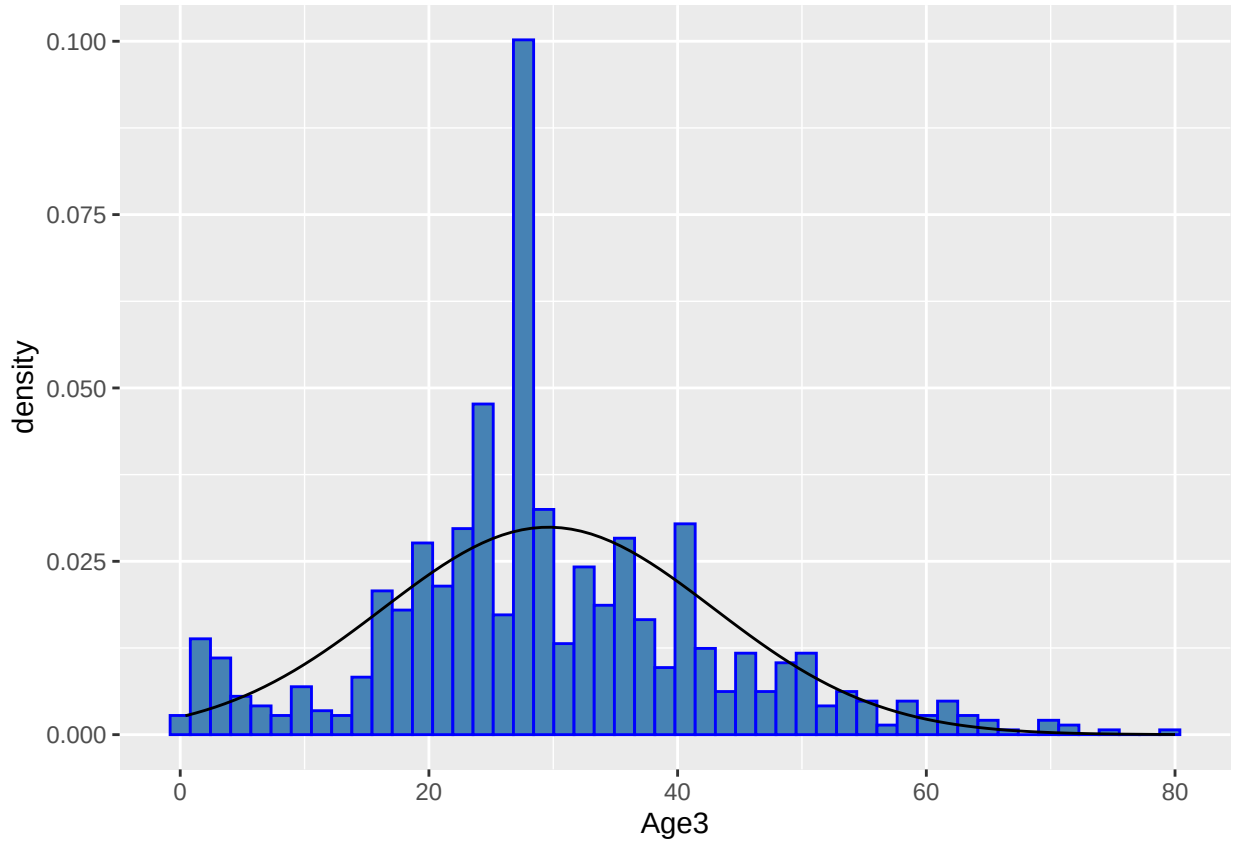
```
data$Age3 <- data$Age
data <- merge(data, medias_clase_fam, by = c("Pclass", "SibSp", "Parch"), all.x = TRUE)
data$Age3[is.na(data$Age3)] <- data$Media_clase_fam[is.na(data$Age3)]
```

Como es posible que nos hayan quedado alguno sin poder asignar (al no existir la combinación Pclass + sibSp + Parch) a los que faltan (que son 7) les asignamos directamente por la Pclass sin tener en cuenta los otros valores

```
data$Age3[is.na(data$Age3)&data$Pclass == 1] <- mean(data$Age3[!is.na(data$Age3)&data$Pclass == 1])
data$Age3[is.na(data$Age3)&data$Pclass == 2] <- mean(data$Age3[!is.na(data$Age3)&data$Pclass == 2])
data$Age3[is.na(data$Age3)&data$Pclass == 3] <- mean(data$Age3[!is.na(data$Age3)&data$Pclass == 3])
```

Vemos ahora la distribución que nos queda

```
ggplot(data, aes(Age3)) +
  geom_histogram(aes(y = ..density..), bins=50, fill="steelblue", color="blue") +
  stat_function(fun = dnorm, args = list(mean = mean(data$Age3), sd = sd(data$Age3)))
```



- Caso 4: Imputando datos de Age, con MICE (Multivariate Imputation via Chained Equations). En este caso se predicen los valores de Age, con el resto de valores observados (usamos para este caso Pclass, SibSp, Parch, Sex y Age).

```
columnas <- c('Pclass', 'SibSp', 'Parch', 'Sex', 'Age')

mice_imputar <- mice(data = data[, columnas], method = "rf")
```

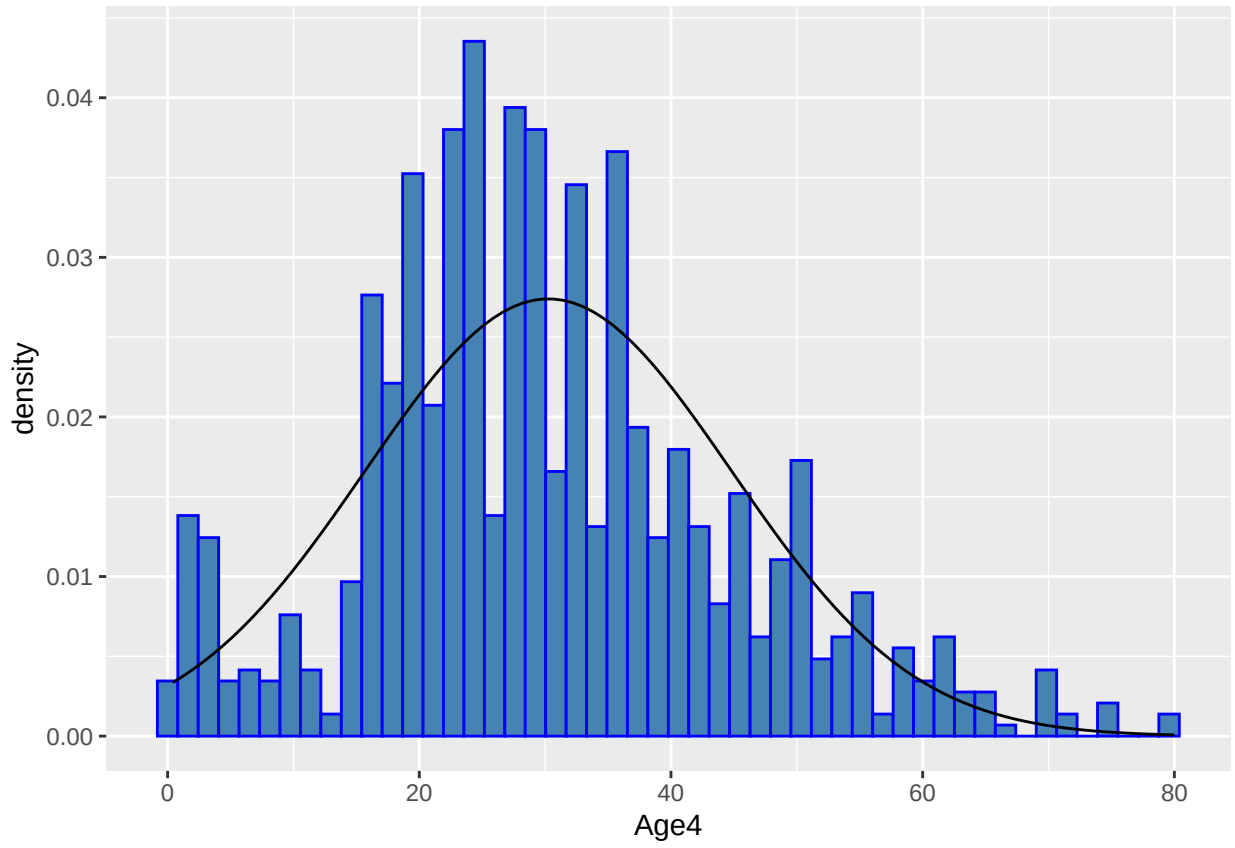
```
##
##  iter imp variable
##   1   1  Age
##   1   2  Age
##   1   3  Age
##   1   4  Age
##   1   5  Age
##   2   1  Age
##   2   2  Age
##   2   3  Age
##   2   4  Age
##   2   5  Age
##   3   1  Age
##   3   2  Age
##   3   3  Age
```

```
##   3   4  Age
##   3   5  Age
##   4   1  Age
##   4   2  Age
##   4   3  Age
##   4   4  Age
##   4   5  Age
##   5   1  Age
##   5   2  Age
##   5   3  Age
##   5   4  Age
##   5   5  Age
```

```
#mice_imputar <-  parlmice(data = data[, columnas], method = "rf", n.core = 8, n.imp.core = 50)
mice_completo <- mice::complete(mice_imputar)
data$Age4 <- data$Age
data$Age4[is.na(data$Age4)]<- mice_completo$Age[is.na(data$Age4)]
```

Tras la imputación observamos la gráfica:

```
ggplot(data, aes(Age4)) +
  geom_histogram(aes(y = ..density..), bins=50, fill="steelblue", color="blue") +
  stat_function(fun = dnorm, args = list(mean = mean(data$Age4), sd = sd(data$Age4)))
```



Resumiendo el resultado de las 4 opciones de imputación mas la original:

```
summary(data$Age)
```

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

```
summary(data$Age1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   22.00   29.70   29.70   35.00   80.00
```

```
summary(data$Age2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   22.00   26.00   29.29   37.00   80.00
```

```
summary(data$Age3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   22.00   28.24   29.60   37.00   80.00
```

```
summary(data$Age4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   21.00   29.00   30.25   38.00   80.00
```

Tras ver los resultados nos decantamos por la cuarta opción (Age4), ya que la distribución se parece mucho más a la original. Asignamos y limpiamos el dataset:

```
data$Age <- data$Age4
data <- select(data, -Age1, -Age2, -Age3, -Age4, -Media_clase_fam)
```

3.1.2 Ceros

Recordamos los valores con ceros del dataset

```
df_status(data)
```

```
##       variable q_zeros p_zeros q_na p_na q_inf p_inf    type unique
## 1       Pclass       0    0.00    0    0     0     0  factor      3
## 2        SibSp     608   68.24    0    0     0     0 integer      7
## 3        Parch     678   76.09    0    0     0     0 integer      7
## 4     Survived     549   61.62    0    0     0     0  factor      2
## 5          Sex       0    0.00    0    0     0     0  factor      2
## 6          Age       0    0.00    0    0     0     0 numeric     88
## 7         Fare      15    1.68    0    0     0     0 numeric    248
## 8     Embarked       0    0.00    0    0     0     0  factor      3
## 9        Price      15    1.68    0    0     0     0 numeric    248
## 10 FamilySize       0    0.00    0    0     0     0 numeric      9
```

Tenemos un alto número de ceros en SibSp y Parch, pero son valores válidos para estas variables, pues cuentan el número de acompañantes (familiares) del pasajero. Hay también un porcentaje pequeño de ceros en Fare (tarifa), esto podría tener algún sentido (por ejemplo, tickets sin coste por ser un premio o un regalo) por lo que, en principio, vamos a dejar presentes estos ceros.

3.1.3 Conclusión limpieza nulos y ceros

Viendo nuevamente los resultados con summary, hemos eliminado los nulos y tenemos un nuevo valor de media y mediana para Age

```
summary(data)
```

```
##  Pclass       SibSp           Parch        Survived     Sex           Age
```

```
##  1:216   Min.   :0.000   Min.   :0.0000   0:549   female:314   Min.   : 0.42
##  2:184   1st Qu.:0.000   1st Qu.:0.0000   1:342   male  :577   1st Qu.:21.00
##  3:491   Median :0.000   Median :0.0000                        Median :29.00
##          Mean   :0.523   Mean   :0.3816                        Mean   :30.25
##          3rd Qu.:1.000   3rd Qu.:0.0000                        3rd Qu.:38.00
##          Max.   :8.000   Max.   :6.0000                        Max.   :80.00
##       Fare        Embarked      Price           FamilySize
##  Min.   :  0.00   C:168    Min.   :  0.000   Min.   : 1.000
##  1st Qu.:  7.91   Q: 77    1st Qu.:  7.763   1st Qu.: 1.000
##  Median : 14.45   S:646    Median :  8.850   Median : 1.000
##  Mean   : 32.20            Mean   : 17.789   Mean   : 1.905
##  3rd Qu.: 31.00            3rd Qu.: 24.288   3rd Qu.: 2.000
##  Max.   :512.33            Max.   :221.779   Max.   :11.000
```

Y con "df_status" vemos también como nos han quedado los datos después de eliminar variables, y de imputar valores en las variables que le faltaban algunos valores. El dataset queda libre de nulos, y con los ceros que hemos aceptado que tiene que mantener y que tienen sentido.

```
df_status(data)
```

```
##       variable q_zeros p_zeros q_na p_na q_inf p_inf    type unique
## 1       Pclass       0    0.00    0    0     0     0  factor      3
## 2        SibSp     608   68.24    0    0     0     0 integer      7
## 3        Parch     678   76.09    0    0     0     0 integer      7
## 4     Survived     549   61.62    0    0     0     0  factor      2
## 5          Sex       0    0.00    0    0     0     0  factor      2
## 6          Age       0    0.00    0    0     0     0 numeric     88
## 7         Fare      15    1.68    0    0     0     0 numeric    248
## 8     Embarked       0    0.00    0    0     0     0  factor      3
## 9        Price      15    1.68    0    0     0     0 numeric    248
## 10 FamilySize       0    0.00    0    0     0     0 numeric      9
```

3.2 Identificación y tratamiento de valores extremos.

Se considera un valor extremo, outlier, a un valor fuera de rango. Son valores que se salen de la escala esperada visualizando el resto de las observaciones. En la actualidad, el criterio más habitual es considerar un valor extremo a aquel que se encuentra alejado de la media unas tres veces la desviación típica. En nuestro caso vamos a comenzar por separar aquellas variables que son numéricas y verlas en un gráfico bloxplot:

TODO##########################Gráfico boxplot variables numéricas

De todas las variables numéricas sólo 3 son contínuas: Age, Price y Fare

Analicemos esas tres variables por separado

3.2.1 Fare

El boxplot parece indicar que Fare tiene valores extremos. Los identificamos usando el criterio de tres veces la desviación típica:

```
data_out <- as.data.frame(data$Fare)
data_out$outlier <- FALSE
for (i in 1:ncol(data_out) - 1){
  columna = data_out[, i]
  if (is.numeric(columna)) {
    media = mean(columna)
    desviacion = sd(columna)
```

```
    data_out$outlier = (columna> (media+3*desviacion) | columna<(media-3*desviacion))
  }
}
table(data_out$outlier)
```

```
##
## FALSE  TRUE
##   871    20
```

Con este criterio tenemos identificados 20 posibles outliers, observando mas detenidamente los valores que nos indica boxplot.stats y teniend en cuenta que el máximo es 512, no parecen exagerados:

```
boxplot.stats(data$Fare)$out
```

```
##   [1]  86.5000  86.5000  86.5000 151.5500 227.5250 227.5250 211.3375  80.0000
##   [9]  80.0000 135.6333  79.2000  79.2000  93.5000 512.3292  69.3000 221.7792
##  [17] 227.5250 134.5000 110.8833 135.6333  83.1583 135.6333 106.4250 153.4625
##  [25]  76.2917  78.8500  77.9583  69.3000  79.2000 146.5208 512.3292  76.7292
##  [33] 153.4625 153.4625  77.2875  77.2875 512.3292  83.1583 211.3375 247.5208
##  [41] 211.3375 247.5208 164.8667  71.0000  79.6500 113.2750 110.8833 211.5000
##  [49]  81.8583  76.7292  82.1708  90.0000  66.6000  89.1042  77.9583  90.0000
##  [57] 113.2750 113.2750  66.6000 106.4250  78.2667  83.4750  83.4750 133.6500
##  [65]  89.1042 227.5250  91.0792 108.9000  77.9583  90.0000  82.1708  75.2500
##  [73]  78.8500  71.2833 146.5208  76.7292  91.0792  78.2667 108.9000  79.6500
##  [81]  71.0000  79.6500 164.8667 110.8833 134.5000 110.8833  79.2000  83.1583
##  [89]  93.5000 120.0000 120.0000 120.0000 151.5500 151.5500 151.5500 120.0000
##  [97] 263.0000 133.6500  90.0000 262.3750 262.3750 263.0000 263.0000 263.0000
## [105]  73.5000  73.5000  73.5000  73.5000  73.5000  69.5500  69.5500  69.5500
## [113]  69.5500  69.5500  69.5500  69.5500
```

Los valores, además, están bien distribuidos, los más altos en las clases altas.

Por todo esto, decidimos no realizar ninguna acción con estos outliers ya que incluso ueden estar aportando información importante:

TODO################## Gráfico outlier Fare

3.2.2 Price

Hacemos un análisis similar al realizado con Fare:

```
data_out <- as.data.frame(data$Price)
data_out$outlier <- FALSE
for (i in 1:ncol(data_out) - 1){
  columna = data_out[, i]
  if (is.numeric(columna)) {
    media = mean(columna)
    desviacion = sd(columna)
    data_out$outlier = (columna> (media+3*desviacion) | columna<(media-3*desviacion))
  }
}
table(data_out$outlier)
```

```
##
## FALSE  TRUE
##   878    13
```

De nuevo con boxplot.stats analizamos los outliers, el valor máximo 221:

```
boxplot.stats(data$Price)$out
```

```
##  [1]  56.88125  56.88125  70.44583  50.00000  50.49580 170.77640  49.50420
##  [8] 221.77920  56.88125  67.25000  53.21250  51.86250  51.15417  76.29170
## [15]  79.20000  73.26040 170.77640  49.50420  51.15417  61.97920  51.15417
## [22] 170.77640  70.44583  61.37920 123.76040  70.44583 123.76040  63.35830
## [29]  82.43335  49.50000 211.50000  81.85830  51.86250  59.40000  53.21250
## [36]  66.82500  61.17500  53.10000  56.88125  54.45000  52.55420  52.00000
## [43]  55.44170  75.25000  71.28330  73.26040  54.45000  82.43335  67.25000
## [50]  79.20000  83.15830  65.75000  66.82500  51.47920 131.18750 131.18750
## [57]  65.75000  65.75000  65.75000
```

Y al igual que con Fare, los valores más altos en las clases altas: TODO############Gráfico boxplot Fare

Al igual que con Fare optamos por considerarlos valores válidos y no realizamos ninguna accion.

3.2.3 Age

De nuevo realizamos el mismo procedimiento que con las otras dos variables

```
data_out <- as.data.frame(data$Age)
data_out$outlier <- FALSE
for (i in 1:ncol(data_out) - 1){
  columna = data_out[, i]
  if (is.numeric(columna)) {
    media = mean(columna)
    desviacion = sd(columna)
    data_out$outlier = (columna> (media+3*desviacion) | columna<(media-3*desviacion))
  }
}
table(data_out$outlier)
```

```
##
## FALSE  TRUE
##   886     5
```

y el resultado de boxplot.stats

```
boxplot.stats(data$Age)$out
```

```
##  [1] 64.0 80.0 65.0 71.0 65.0 71.0 65.0 74.0 70.0 64.0 70.0 66.0 65.0 70.5 70.0
## [16] 70.0 74.0 74.0 70.5 80.0
```

Donde vemos que podríamos tener 2 valores outliers, pero observando los valores devueltos por boxplot.stats que son totalmente normales no los vamos a considerar outliers.

4 Análisis de los datos.

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

Del dataset completo nos interesa poder realizar diferentes análisis en función de diferentes subconjuntos de datos, como puede ser el género, la clase en la que viajan los pasajeros, el puerto en el que han embarcado, incluso se pueden definir grupos por edad, y ver realmente si es cierto y se cumplió aquello que dicen en las películas "las mujeres y los niños primero" y poder comprobar si realmente los niños tienen mejor índice de supervivencia que los adultos. Podemos definir diferentes agrupaciones que podemos usar más adelante para estudiar los casos por grupos

4.1.1 Niños

Vamos a definir una variable Child para aquellso registros en los que la edad sea menor que 8

```
edad_corte = 8
data$Child[data$Age <= edad_corte] <- 1
data$Child[data$Age > edad_corte] <- 0
data$Child <- as.factor((data$Child))
```

4.1.2 Género

Agrupamos por el género, creando una variable para cada uno de los géneros

```
Mujeres <- data[which(data$Sex == 'female'),]
Hombres <- data[which(data$Sex == 'male'),]
```

4.1.3 Lugar de embarque

Agrupamos por el lugar de embarque creando una variable por cada uno de los lugares

```
EmbarqueC <- data[which(data$Embarked == 'C'),]
EmbarqueQ <- data[which(data$Embarked == 'Q'),]
EmbarqueS <- data[which(data$Embarked == 'S'),]
```

4.1.4 Clase

Agrupamos por clase

```
FirstClass <- data[which(data$Pclass == 1),]
SecondClass <- data[which(data$Pclass == 2),]
ThirdClass <- data[which(data$Pclass == 3),]
```

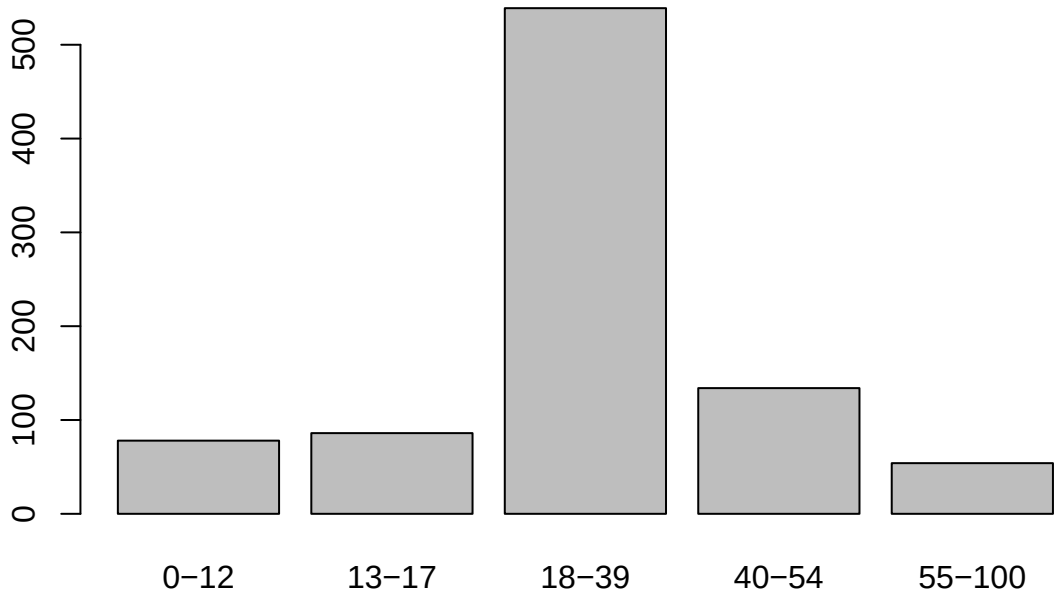
4.1.5 Edades

Vamos a discretizar los valores por grupos de edades, añadiendo una columan al dataset (AgeInterval)

```
data["AgeInterval"] <- cut(data$Age, breaks = c(0,13,18,40,55, 100), labels = c("0-12", "13-17", "18-39","40-54", "55-100"))
#intervalos_edad <- c(0,13,18,40,55)
#data$AgeInterval <- findInterval(data$Age, intervalos_edad)
data$AgeInterval <- as.factor(data$AgeInterval)
table(data$AgeInterval)
```

```
##
##   0-12  13-17  18-39  40-54 55-100
##     78     86    539    134     54
```

```
plot(data$AgeInterval)
```



4.2 Comprobación de la normalidad y homogeneidad de la varianza.

4.2.1 Normalidad

Vamos a verificar si las variables cuantitativas continuas siguen una distribución normal. Algunos test estadísticos requieren que las variables que van a ser analizadas sigan una distribución normal, por tanto tenemos que conocer cuáles son las distribuciones de nuestras variables continuas. Las únicas variables cuantitativas continuas que tenemos en el dataset original son las variables Age y Fare. Además, hemos generado una nueva variable a partir de Fare, que hemos llamado Price, y que, por tanto también es una variable cualitativa continua.

En general, la prueba de Shapiro-Wilk se considera una prueba muy potente para contrastar la normalidad de distribuciones. Se asume como hipótesis nula que la población sigue una distribución normal. Si el p-valor obtenido es inferior al nivel de significancia (normalmente = 0,05) entonces se rechaza la hipótesis nula (y por tanto se concluye que los datos no vienen de una distribución normal). En cambio, si el p-valor es superior al nivel de significancia, entonces no se puede rechazar la hipótesis nula y se asume que los datos siguen una distribución normal. Para poder tener más seguridad, vamos a aplicar otros dos métodos, la prueba de Anderson-Darling y la prueba de Kolmogorov-Smirnov (conocida también como K-S)

Creamos un data frame para resumir los tests:

```
tabla.normalidad <- data.frame('variable' = character(),
                               'Test de Normalidad' = character(),
                               'Valor Estadístico' = numeric(),
                               'p-value' = numeric(),
```

```
                        stringsAsFactors = FALSE)
str(tabla.normalidad)
```

```
## 'data.frame':    0 obs. of  4 variables:
##  $ variable          : chr
##  $ Test.de.Normalidad: chr
##  $ Valor.Estadístico : num
##  $ p.value           : num
```

Ahora recorremos todas las variables continuas aplicando los tres tests y añadiéndolos al dataframe:

```
var.continuas <-c("Age", "Fare", "Price")
library(nortest)
for (i in 1:length(var.continuas)){
  variable = var.continuas[i]
  #Test Shapiro-wil
  test = shapiro.test(data[,variable])
  tabla.normalidad[nrow(tabla.normalidad)+1,] = c(variable, test$method, test$statistic, test$p.value)

  #Test Anderson-Darling
  test = ad.test(data[,variable])
  tabla.normalidad[nrow(tabla.normalidad)+1,] = c(variable, test$method, test$statistic, test$p.value)

  #Test Kolmogorov-Smirnov
  test = ks.test(data[,variable], "pnorm", mean=mean(data[,variable]), sd=sd(data[,variable]))
  tabla.normalidad[nrow(tabla.normalidad)+1,] = c(variable, test$method, test$statistic, test$p.value)
}

knitr::kable(tabla.normalidad)
```

variable	Test.de.Normalidad	Valor.Estadístico	p.value
Age	Shapiro-Wilk normality test	0.978725525509169	4.10673866775266e-10
Age	Anderson-Darling normality test	5.48588379408523	1.55683609082601e-13
Age	One-sample Kolmogorov-Smirnov test	0.0661030933298594	0.000830474125728897
Fare	Shapiro-Wilk normality test	0.521891302117355	1.08404452322613e-43
Fare	Anderson-Darling normality test	122.169627214592	3.7e-24
Fare	One-sample Kolmogorov-Smirnov test	0.281848040985975	0
Price	Shapiro-Wilk normality test	0.566484900244852	3.00259150054125e-42
Price	Anderson-Darling normality test	110.325199878766	3.7e-24
Price	One-sample Kolmogorov-Smirnov test	0.26829589819932	0

Con estos resultados se puede decir que ninguna de las 3 variables sigue una distribución normal, en todos los casos el p-value ha sido inferior a 0.05 y por tanto se han rechazado la hipótesis nula (que la variable sigue una distribución normal).

No obstante, vamos a revisar gráficamente la distribución de cada una de las variables usando su histograma, curva de densidad y gráficas Q-Q

TODO##########################Graficas histogŕamas y q-q

4.2.2 Homocedasticidad (comprobación de varianzas)

Cuando comparamos varianzas lo que estamos comprobando es que las varianzas entre los grupos a comparar son iguales. Si los datos siguen una distribución normal, podemos usar el test de Levene, en caso contrario podemos usar por ejemplo el test de Fligner-Killeen, que es la alternativa no paramétrica que se utiliza cuando los datos no siguen una distribución normal (o cuando hay problemas con outliers no resueltos)

En ambos test (Levene y Fligner-Killeen), la hipótesis nula asume la igualdad de varianzas en los diferentes grupos de datos, con lo que si el p-valor obtenido es inferior al nivel de significancia (generalmente alpha = 0,05) se rechaza la hipótesis nula y se concluye que hay heterocedasticidad.

4.2.2.1 Age La variable Age está próxima a una distribución normal por lo cual podemos utilizar el test de Levene para la comprobación de varianzas.

4.2.2.1.1 Age con Survived Primero analizamos si las varianzas son iguales cuando comprobamos Age y el grupo es Survived, es decir estamos comprobando la homogeneidad de varianzas de la edad en los grupos de supervivientes y no supervivientes.

```
leveneTest(data = data, Age ~ Survived, center = mean)
```

	Df	F value	Pr(>F)
group	1	0.8355953	0.3609074
	889	NA	NA

En este caso el p-value es superior a 0.05 y por tanto asumimos que hay homogeneidad de varianzas entre los grupos

TODO############ Representación muestras forma gráfica

4.2.2.1.2 Age con Embarked Ahora Comprobamos también cómo se comportan las varianzas cuando se trata de la variable edad "Age" con "Embarked".

```
leveneTest(data = data, Age ~ Embarked, center = mean)
```

	Df	F value	Pr(>F)
group	2	1.32557	0.2661766
	888	NA	NA

En este caso hay omogeneidad de varianzas de Age en los grupos que define la variable Embarked

4.2.2.1.3 Age con Pclass Ahora Comprobamos también cómo se comportan las varianzas cuando se trata de la variable edad "Age" con "Pclass". En este caso vamos a aplicar tanto el test de Levene como el de Fligner-Killeen:

```
leveneTest(data = data, Age ~ Pclass, center = mean)
```

	Df	F value	Pr(>F)
group	2	3.512445	0.0302389
	888	NA	NA

```
fligner.test(Age ~ Pclass, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Pclass
## Fligner-Killeen:med chi-squared = 7.8028, df = 2, p-value = 0.02021
```

En los dos tests (parmétrico y no paramétrico), se rechaza la hipótesis nula, con lo que hay heterogeneidad d varianzas entre las muestras de Age cuando se agrupan por Pclass

```
leveneTest(data = data, Age ~ Sex, center = mean)
```

4.2.2.1.4 Age con Sex

	Df	F value	Pr(>F)
group	1	0.2422369	0.6227166
	889	NA	NA

```
fligner.test(Age ~ Sex, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Sex
## Fligner-Killeen:med chi-squared = 0.50496, df = 1, p-value = 0.4773
```

En los dos tests el p-value es claramente superior a 0.05 por lo que podemos afirmar que sí hay homogeneidad de varianzas para Age cuando está agrupado por Sex

4.2.2.2 Fare

Hacemos el mismo estudio con la variable Fare pero usando Fligner

```
fligner.test(Fare ~ Sex, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by Sex
## Fligner-Killeen:med chi-squared = 55.949, df = 1, p-value = 7.436e-14
```

```
fligner.test(Fare ~ Survived, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by Survived
## Fligner-Killeen:med chi-squared = 96.253, df = 1, p-value < 2.2e-16
```

```
fligner.test(Fare ~ Embarked, data = data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by Embarked
## Fligner-Killeen:med chi-squared = 133.23, df = 2, p-value < 2.2e-16
```

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by Pclass
## Fligner-Killeen:med chi-squared = 365.8, df = 2, p-value < 2.2e-16
```

Comprobamos la variable “Fare”, los resultados indican que hay heterogeneidad de varianza de esas variables respecto a los grupos con los que se ha aplicado el test no paramétrico.

¿? #### Price

4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos.

4.3.1 Análisis de relaciones entre variables

Primero vamos a realizar diversos análisis de relaciones variables (principalmente pares de variables), para aclarar la dependencia entre ellas y como pueden afectar a las posibilidades de supervivencia

4.3.1.1 Age y Sex

Primero vamos a analizar si hay diferencias significativas entre la media de edad de mujeres y de hombres. Para hacerlo aplicamos el test paramétrico t-test de Student, que requiere que las muestras a comparar sigan una distribución normal. Ya vimos que Age no sigue exactamente una distribución normal, pero si consideramos el teorema central del límite con una muestra suficientemente grande (mayor de 30) se puede asumir que la variable sigue una distribución normal. Si la variable no siguiese una distribución normal habríamos aplicado una prueba no paramétrica como por ejemplo el test de Mann-Whitney.

Para ejecutar la prueba t de Student, disponemos en R del comando "t.test". Este mismo comando nos permite realizar el test de Welch cuando las varianzas entre las muestras son diferentes. Comprobemos que la variable Age sigue una distribución normal primero aplicando el F-test mediante el comando var.test

```
var.test(Mujeres$Age, Hombres$Age)
```

```
##
##  F test to compare two variances
##
## data:  Mujeres$Age and Hombres$Age
## F = 0.98531, num df = 313, denom df = 576, p-value = 0.8895
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8131183 1.2007933
## sample estimates:
## ratio of variances
##          0.9853135
```

El resultado nos indica que las varianzas de edad en los grupos de mujeres y hombres son iguales.

Aplicamos ahora un test para comparar las medias de los 2 grupos, la hipótesis nula será que no hay diferencias significativas entre la media de edades para hombres y mujeres. Podemos usar el t-Test indicando que las varianzas de los grupos son iguales (por el resultado anterior).

```
t.test(Mujeres$Age, Hombres$Age, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  Mujeres$Age and Hombres$Age
## t = -2.9702, df = 889, p-value = 0.003056
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.014686 -1.024255
## sample estimates:
## mean of x mean of y
##  28.29538  31.31485
```

El valor de p-values es inferior a 0.05, por lo que rechazamos la hipótesis nula. Es decir, la media de edad por género tiene una diferencia significativa.

Esstablecemos una nueva hipótesis alternativa: que la media de edad de las mujeres es menor que la de los hombres. Por lo tanto, la hipótesis nula corresponde a que la edad de las mujeres es mayor o igual a la de los hombres:

```
t.test(Mujeres$Age, Hombres$Age, alternative = "less", var.equal = TRUE)
```

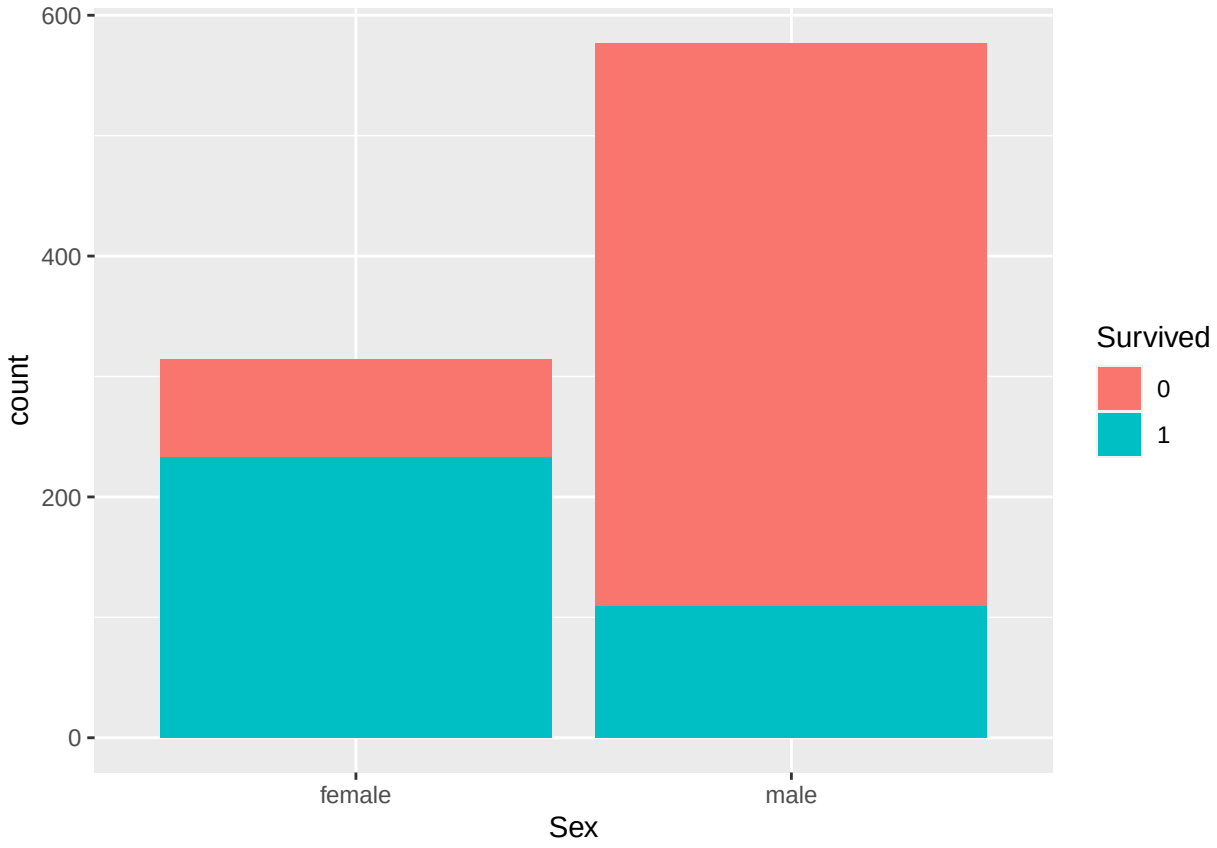
```
##
##  Two Sample t-test
##
## data:  Mujeres$Age and Hombres$Age
## t = -2.9702, df = 889, p-value = 0.001528
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -1.345568
## sample estimates:
## mean of x mean of y
##  28.29538  31.31485
```

Con un p-value menor de 0.5 volvemos a rechazar la hipótesis nula, es decir, podemos afirmar que la media de edad de las mujeres del Titanic es inferior a la media de edad de los hombres a bordo.

4.3.1.2 Survival y Sex

Si hacemos una inspección visual de la relación entre las dos variables:

```
ggplot(data,aes(x=Sex,fill=Survived))+geom_bar()
```



Parece indicar que ser mujer tenía mas posibilidades de supervivencia que siendo hombre. Vamos a confirmar esto realizando algunos contrastes de hipótesis.

Primero, vamos a comprobar la relación entre la supervivencia y el género, analizando si el género fue un factor importante a la hora de la supervivencia, o dicho de otro modo, si dependiendo de si se era mujer u hombre las posibilidades de supervivencia cambiaban significativamente. Para hacerlo aplicamos el test

exacto de Fisher que analiza tablas de contigencia. En este caso la hipótesis nula es que la proporción de mujeres que mueren coincide con la proporción de hombre que mueren en el accidente del Titanic.

```
fisher.test(table(data$Sex, data$Survived))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table(data$Sex, data$Survived)
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.0575310 0.1138011
## sample estimates:
## odds ratio
## 0.08128333
```

El resultado no deja lugar a dudas con un p-valor < 0.05 rechazamos la hipótesis nula, es decir: hay diferente proporción de supervivencia entre hombres y mujeres.

Una vez confirmado que el género si que es importante a ala hora de las probabilidades de supervivencia, vamos a ver quien tiene mas probabilidades de supervivencia, si las mujeres o los hombres. Para ello añadimos una condición a la hipótesis alternativa, usamos "less" para indicar (en la hipótesis nula) que el primer grupo (en este caso las mujeres ya que en orden alfabético es female y male) tiene menor proporción (probabilidad) si se rechaza la hipótesis nula. Es decir, la hipótesis nula dice que "la proporción de mujeres que mueren es mayor que la de los hombres".

```
fisher.test(table(data$Sex, data$Survived), alternative = 'less')
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table(data$Sex, data$Survived)
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is less than 1
## 95 percent confidence interval:
##  0.0000000 0.1081391
## sample estimates:
## odds ratio
## 0.08128333
```

De nuevo rechazamos la hiótesis nula (p-value inferior a 0.05) con lo que se cumple la hipótesis alternativa: la proporción de mujeres que mueren es inferior a la de los hombres.

Si tenemos en cuenta que la tabla de contingencia que estamos evaluando es la siguiente:

```
table(data$Sex, data$Survived)
```

```
##
##            0   1
##   female  81 233
##   male   468 109
```

El odd ratio de la tabla de contigengica lo calculamos así:

```
tab.contigencia = table(data$Sex, data$Survived)
odd_ratio = (tab.contigencia[1,1] / tab.contigencia[1,2]) /
  (tab.contigencia[2,1] / tab.contigencia[2,2])
odd_ratio
```

[1] 0.08096732

Como vimos antes en el test de Fisher, la hipótesis alternativa se hace verdadera cuando el odd_ratio se vuelve 1. Luego cuando a la hipótesis alternativa le ponemos “less”, esta se hace verdadera cuando odd_ratio es menor que 1. Y si ponemos “greater” la hipótesis alternativa se hace verdadera si el odd_ratio es mayor que 1.

Podemos obtener intervalos de confianza, por ejemplo, al 99% de que la media de las edades de diferentes muestras, van a estar entre los valores 28.8 y 31.26 de media de edad.

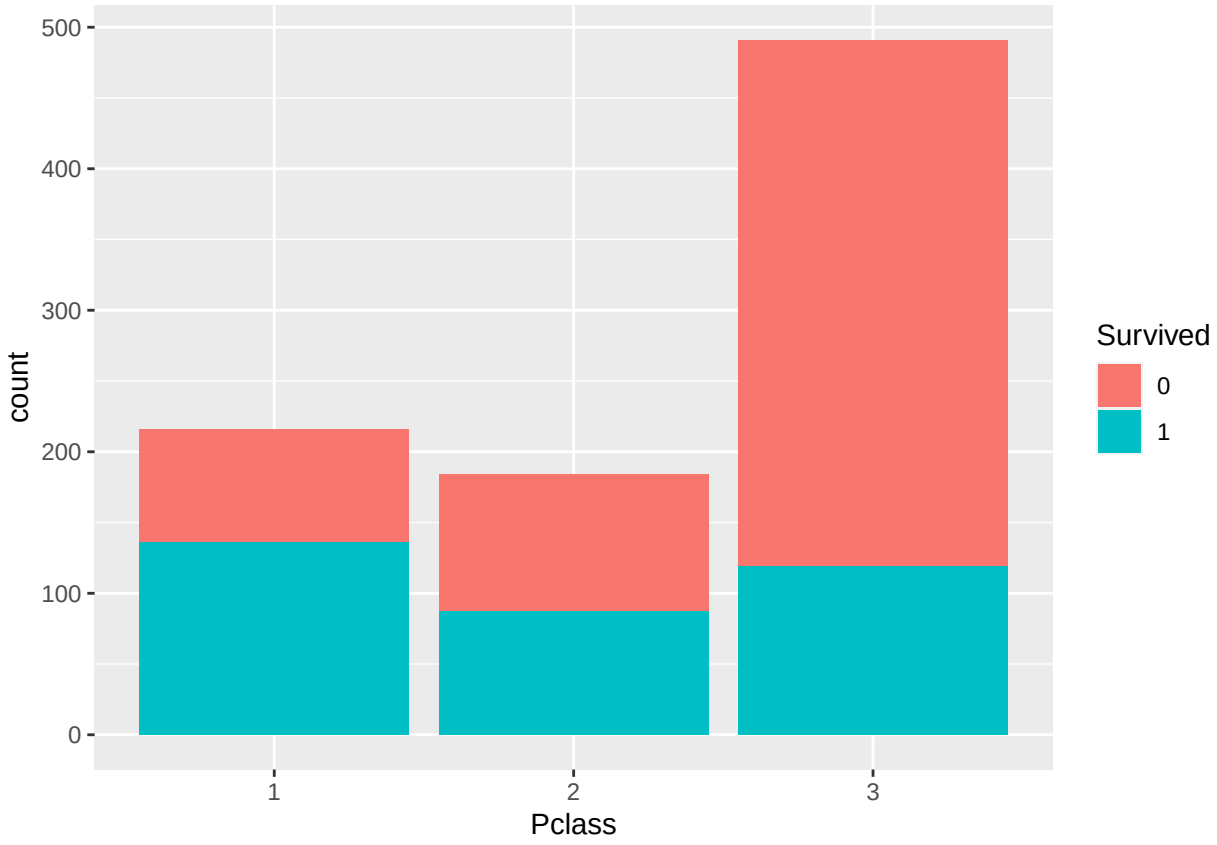
```
test = t.test(data$Age, conf.level = 0.99)
test$conf.int
```

[1] 28.99161 31.50989
attr(,”conf.level”)
[1] 0.99

4.3.1.3 Survival y Pclass

De nuevo hacemos una primera inspección visual

```
ggplot(data,aes(x=Pclass,fill=Survived))+geom_bar()
```



En el gráfico se aprecia claramente que el porcentaje de supervivencia por clases disminuye desde la primera hasta la tercera clase.

Para realizar el análisis numéricamente, al ser dos variables categóricas vamos a aplicar por ejemplo el test Chi-Cuadrado. Aquí la hipótesis nula nos dice que las variables son independientes. Vamos a comprobarlo:

```
test_chisq <- chisq.test(data$Pclass, data$Survived)
test_chisq
```

```
##
##  Pearson's Chi-squared test
##
## data:  data$Pclass and data$Survived
## X-squared = 102.89, df = 2, p-value < 2.2e-16
```

El p-value obtenido es inferior a 0.05, es decir, rechazamos la hipótesis nula, y por tanto afirmamos que las variables no son independientes. Esto quiere decir que el grado de supervivencia era distinto dependiendo de la clase en la que estuvieses embarcado. Confirma el análisis visual, que nos aporta también que las clases con mas superviencia eran las mejores.

También se puede observar el valor de df que son los grados de libertad, que lo podemos obtener a través de la tabla de contingencias y el estadístico de contraste. Si el estadístico de contraste obtenido supera el valor crítico entonces se rechaza la hipótesis nula.

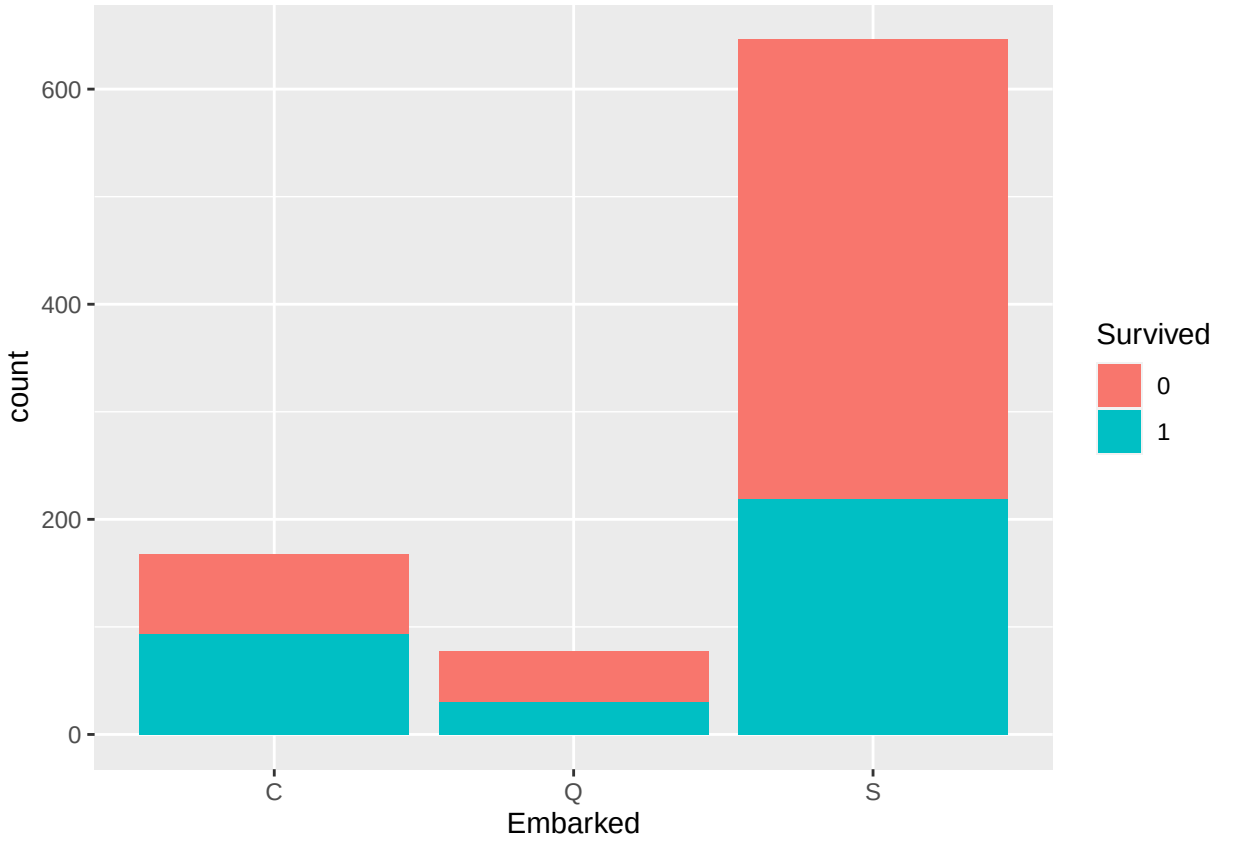
```
df_chisq <- (length(levels(data$Pclass)) -1 ) * (length(levels(data$Survived)) - 1)
valorcritico_chisq <- qchisq(p=0.05, df = df_chisq, lower.tail = FALSE)
cat(sprintf("El estadístico obtenido es: %f y el valor crítico es: %f", test_chisq$statistic, valorcritico_chisq))
```

```
## El estadístico obtenido es: 102.888989 y el valor crítico es: 5.991465
```

4.3.1.4 Survived y Embarked

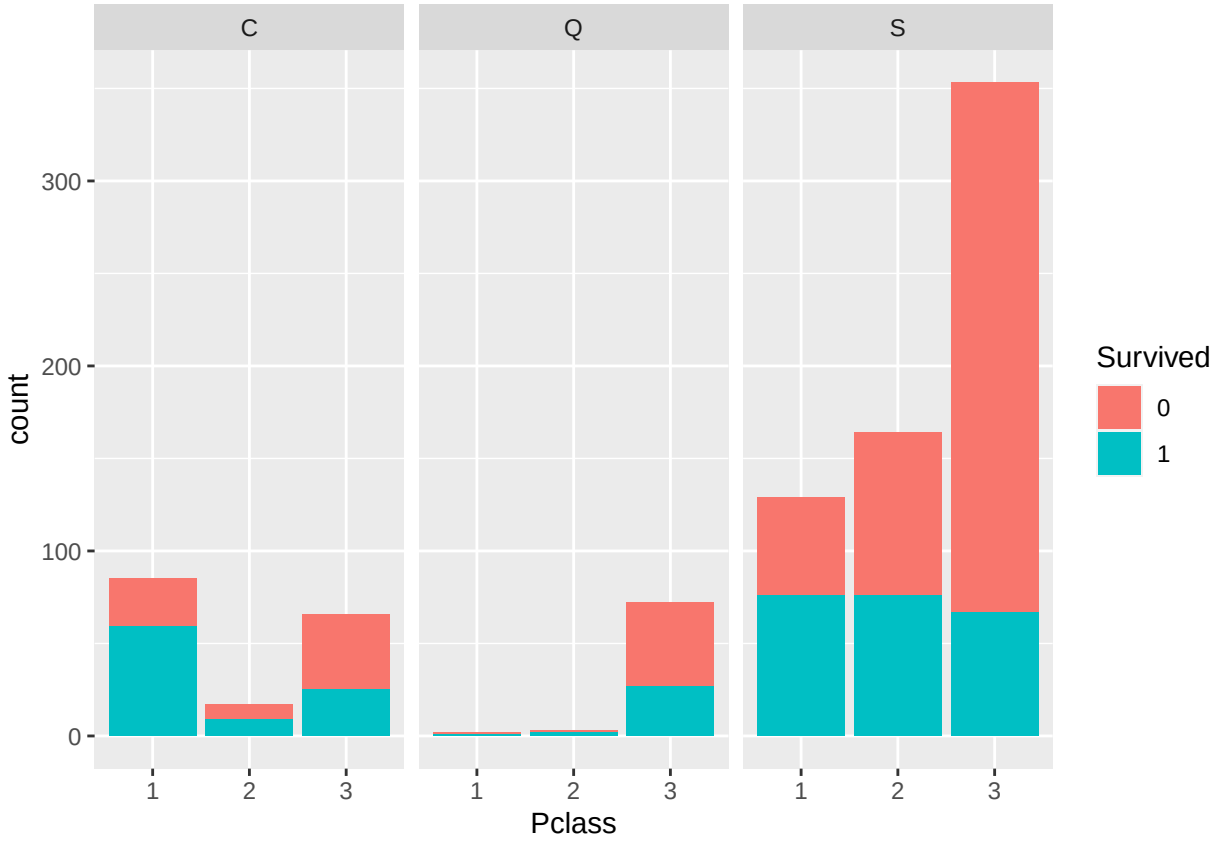
Realizamos el mismo análisis

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



El gráfico parece indicar que los embarcados en Southampton tenían menos posibilidades de sobrevivir. Puede ser que esté relacionado con la Pclass de cada lugar de embarque:

```
ggplot(data,aes(x=Pclass,fill=Survived))+geom_bar()+facet_wrap(~Embarked)
```



Este nos confirma los embarcados en S eran mayoritariamente de clase 3 por lo que tiene sentido que su ratio de supervivencia sea menor.

Vemos ahora el análisis con contrastes de hipótesis.

Primero analizamos si hay relación entre las variables Embarked y Survived usando de nuevo la Chi-Squared

```
test_chisq <- chisq.test(data$Embarked, data$Survived)
test_chisq
```

```
##
##  Pearson's Chi-squared test
##
## data:  data$Embarked and data$Survived
## X-squared = 25.964, df = 2, p-value = 2.301e-06
```

```
df_chisq <- (length(levels(data$Embarked)) -1 ) * (length(levels(data$Survived)) - 1)
valorcritico_chisq <- qchisq(p=0.05, df = df_chisq, lower.tail = FALSE)
cat(sprintf("El estadístico obtenido es: %f y el valor crítico es: %f", test_chisq$statistic, valorcritico_chisq))
```

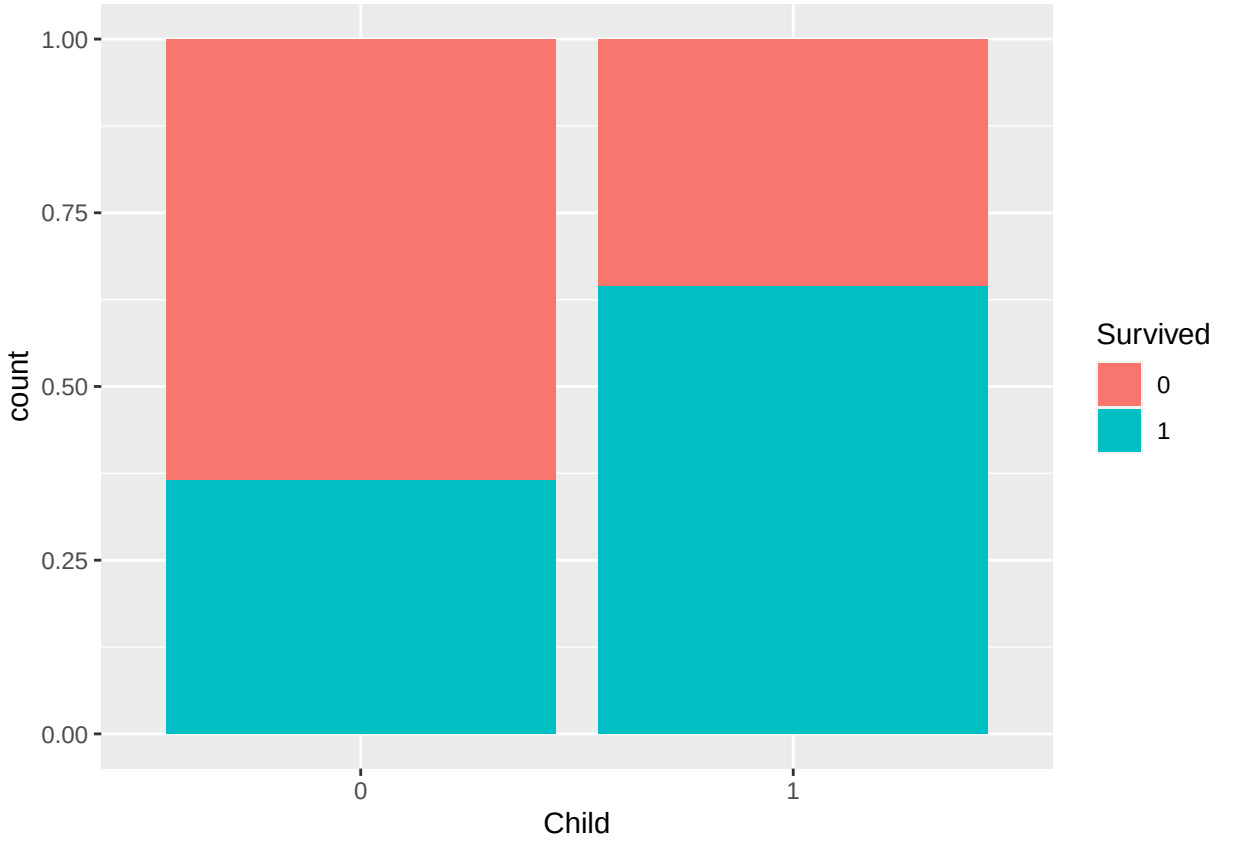
```
## El estadístico obtenido es: 25.964453 y el valor crítico es: 5.991465
```

De nuevo se rechaza la hipótesis nula, por lo tanto las variables son dependientes.

4.3.1.5 Survived y Child Analizamos ahora la nueva variable creada Child que corresponde a los niños de edad menor o igual a 8 años (que es un 1 en el valor de la clase)

De nuevo un análisis visual primero:

```
ggplot(data,aes(x=Child,fill=Survived))+geom_bar(position ='fill')
```



Parece que si que hay mas esperanza de supervivencia en ese caso.

Aplicamos el test de Fisher para comprobar si tienen las mismas probabilidades de supervivencia que el resto de pasajeros.

```
fisher.test(table(data$Child, data$Survived))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table(data$Child, data$Survived)
## p-value = 4.066e-05
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.758776 5.740933
## sample estimates:
## odds ratio
##   3.138619
```

Se rechaza la hipótesis nula, es decir que las propociones no coinciden. Analizamos ahora si las probabilidades de supervivencia son mayores o menores. Para ello analizamos si la proporcion de no-niños que no sobreviven es menor a la proporoción de niños que no sobreviven:

```
fisher.test(table(data$Child, data$Survived), alternative = 'greater')
```

```
##
##  Fisher's Exact Test for Count Data
```

```
##
## data:  table(data$Child, data$Survived)
## p-value = 2.562e-05
## alternative hypothesis: true odds ratio is greater than 1
## 95 percent confidence interval:
##  1.915509      Inf
## sample estimates:
## odds ratio
##   3.138619
```

Se rechaza la hipótesis nula, y se acepta la alternativa, es decir que la probabilidad de muerte de los no-niños es mayor que la de los niños.

```
tab.contigencia = table(data$Child, data$Survived)
tab.contigencia
```

```
##
##       0   1
##   0 528 304
##   1  21  38
```

```
odd_ratio = (tab.contigencia[1,1] / tab.contigencia[1,2]) /
  (tab.contigencia[2,1] / tab.contigencia[2,2])
odd_ratio
```

```
## [1] 3.142857
```

4.3.1.6 Survived y Age Vamos a analizar ahora si la media de edad de las personas que murieron es coincidente con la media de edad que sobrevivieron. Aplicamos el test t-Test

```
t.test(data$Age[which(data$Survived==0)], data$Age[which(data$Survived == 1)])
```

```
##
##  Welch Two Sample t-test
##
## data:  data$Age[which(data$Survived == 0)] and data$Age[which(data$Survived == 1)]
## t = 2.7627, df = 701.83, p-value = 0.005883
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.8060883 4.7659211
## sample estimates:
## mean of x mean of y
##  31.32013  28.53412
```

Se observa que el p-valor es inferior a 0.05, con lo que rechazamos la hipótesis nula (que las medias de edad son coincidentes para fallecidos y supervivientes), pero como ya sabíamos, la distribución de Age no sigue una normal, por lo que vamos a aplicar el test no paramétrico U de Mann-Whitney

```
wilcox.test(x = data$Age[which(data$Survived==0)],
            y = data$Age[which(data$Survived==1)],
            paired =FALSE
            )
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data$Age[which(data$Survived == 0)] and data$Age[which(data$Survived == 1)]
## W = 102236, p-value = 0.02524
```

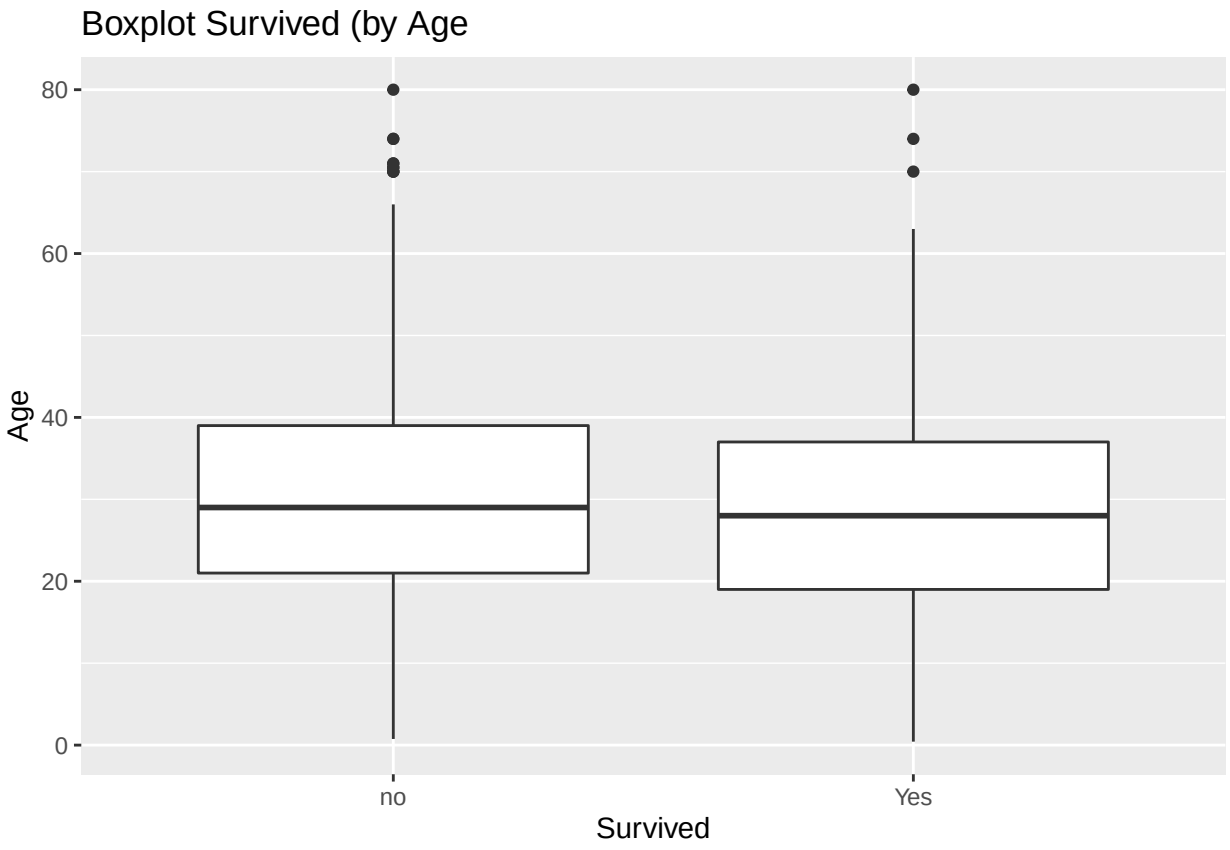
alternative hypothesis: true location shift is not equal to 0

La prueba no-paramétrica nos devuelve un p-value por encima de 0.05 y por tanto no rechazamos la hipótesis nula, con lo que podemos decir que la media de edad es coincidente entre los muertos y los supervivientes del accidente del Titanic.

Gráficamente podemos comprobarlo mediante un Boxplot

TODO###################### colores boxplot

```
ggplot(data, aes(x=Survived, y=Age)) +
  scale_x_discrete(name = "Survived", labels = c("0"="no", "1"="Yes")) +
  geom_boxplot() +
  #geom_boxplot(color="black", fill = c(colores_defecto_ggplot[1], colores_defecto_ggplot[2])) +
  ggtitle("Boxplot Survived (by Age")
```



Se ve claramente que las distribuciones son muy parecidas.

4.3.1.7 Age y Embarked Para analizar si Age es independiente del puerto de embarque Embarked , podemos realizar un test Anova, donde la hipótesis nula nos dice que la media de la edad es independiente del puerto de embarque, es decir que la media de edad de la gente que subió en "C" (Cherbourg), coincide con la media de los que embarcaron en "Q" (Queenstown) y también con los que embarcaron en "S" (Southampton). Y por el contrario la hipótesis nula es que alguna de las medias es diferente.

```
modelo_anova <- aov(formula = Age ~ Embarked, data =data )
resumen_anova <- summary(modelo_anova)
resumen_anova
```

Df Sum Sq Mean Sq F value Pr(>F)

```
## Embarked      2    283   141.6   0.668  0.513
## Residuals   888 188393   212.2
```

Observamos dos filas, una que pone “Embarked” y otra que pone “Residuals”. La primera fila corresponde a todo lo relativo a la varianza explicada (la de lavariable independiente “Embarked”) y la segunda fila relativo a la varianza no explicada o residual.

Explicación de los resultados del test Anova:

Df son los grados de libertad, para el caso de la varianza explicada (la de la variable independiente) es k-1. En nuestro caso como “Embarked” tiene 3 valores posibles, entonces Df=2.

Para la parte de la varianza residual, es n - k, por tanto el valor Df=888 (891 son los registros del dataframe, si le restamos los 2 de antes quedan los 888).

Sum Sq es la suma de la diferencia de los cuadrados, conocidos como SCDe (variación entre-grupos) y SCDi (variación intra-grupos). La función nos ha devuelto 427 y 175873 respectivamente.

Mean Sq es la media cuadrática referente a entre-grupos e intra-grupos. Lo que también llamamos como Varianza explicada (línea superior, es decir 213.6) y varianza no explicada (línea inferior, es decir 198.1). Corresponde al cociente entre la suma de diferencias de cuadrados y los grados de libertad, es decir realmente Mean Sq = Sum Sq / Df lo que se puede comprobar realizando los cálculos sobre los datos anteriores.

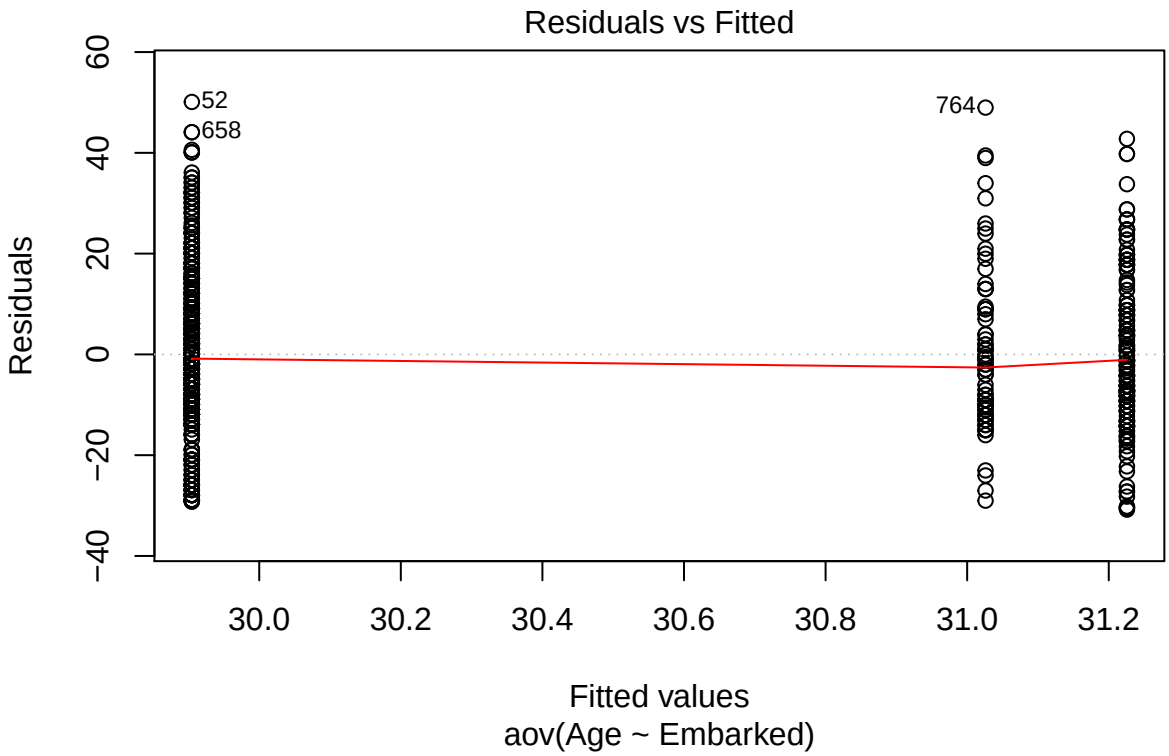
F Es el estadístico (Fisher-Snedecor), que es el cociente entre la varianza explicada entre la varianza no explicada. Por tanto es el cociente entre los valores Mean Sq, siendo el numerador la fila superior (213.6) y el denominador la fila inferior (213.6). El valor del cociente es 0.956, es decir el estadístico F.

Pr(>F) Es la probabilidad asociada, el p-valor que nos indicará si se rechaza o no la hipotesis nula. En este caso el valor es: 0.442 que es mayor que 0.05 y nos permite aceptar (no rechazar) la hipótesis nula que decía que las medias de la variable “Age” en función del valor de “Embarked” eran iguales.

Revisamos los distintos gráficos que nos proporciona anova

- Gráfico Residuals vs Fitted

```
plot(modelo_anova, 1)
```

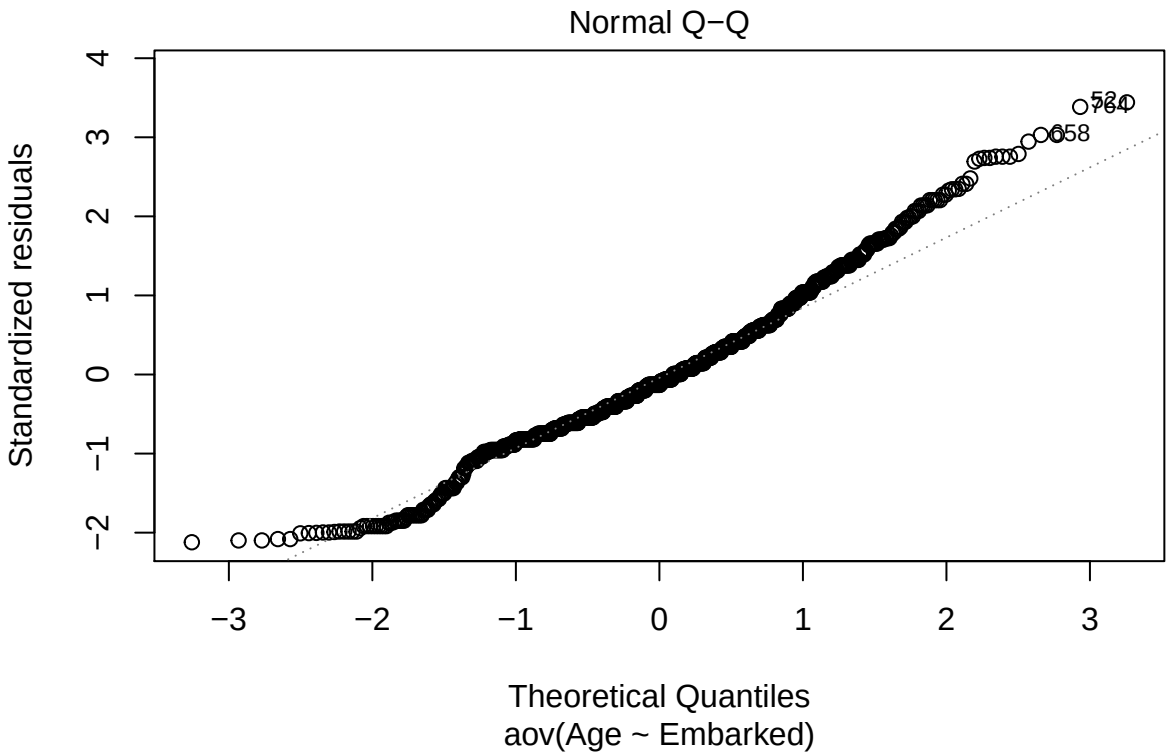


Esta gráfica nos muestra si los residuos tienen patrones no lineales. Puede existir una relación no lineal entre las variables explicativas y la variable explicada, y ese patrón podría verse en esta gráfica si el modelo no captura esa relación no lineal. Si los residuos están igualmente distribuidos alrededor de una línea horizontal sin patrones diferentes, es una buena señal de que no hay relaciones no lineales.

En nuestro caso concreto, se ve una línea casi horizontal (hay una mini curva) y los residuos se distribuyen alrededor de dicha línea. Podemos decir que no hay indicios de relaciones no lineales.

- Gráfico Normal Q-Q

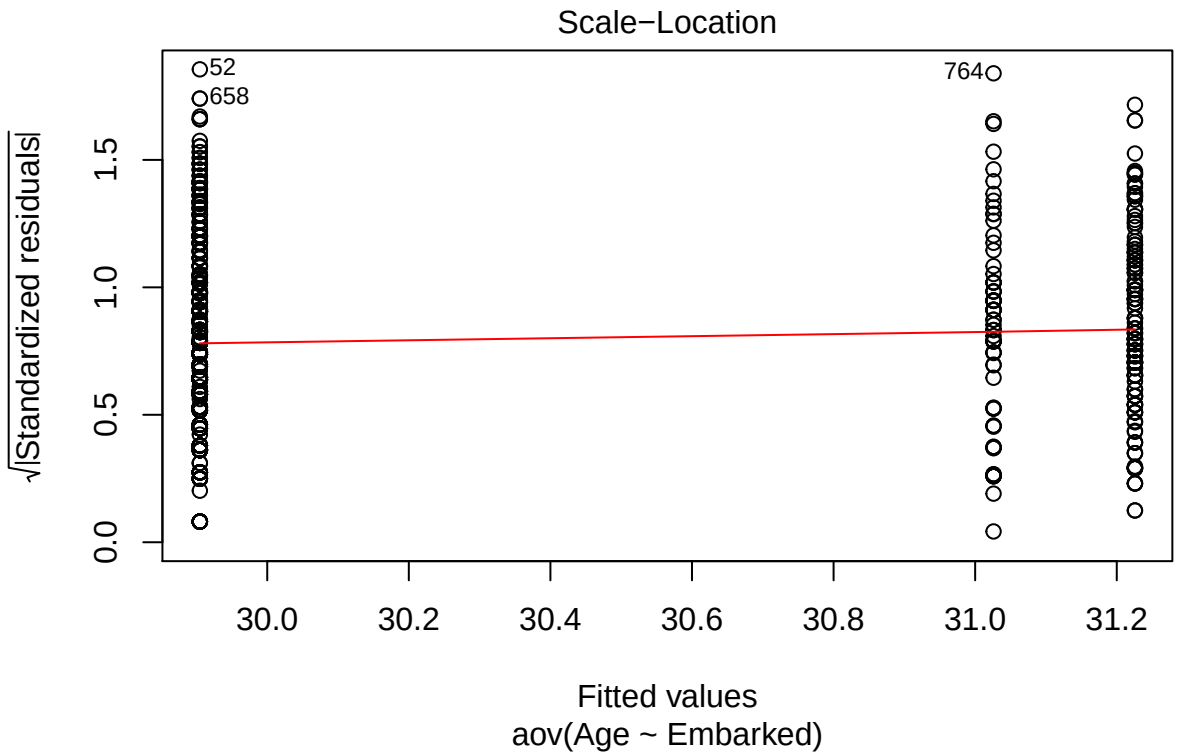
```
plot(modelo_anova, 2)
```



Si no nos ponemos muy estrictos en la exigencia, con el gráfico Normal Q-Q podemos ver que la variable dependiente sigue una distribución aproximadamente normal. La mayoría de los puntos se encuentran alineados en la línea de puntos y solamente es en los extremos donde esa situación no se cumple.

- Gráfica Scale-Location:

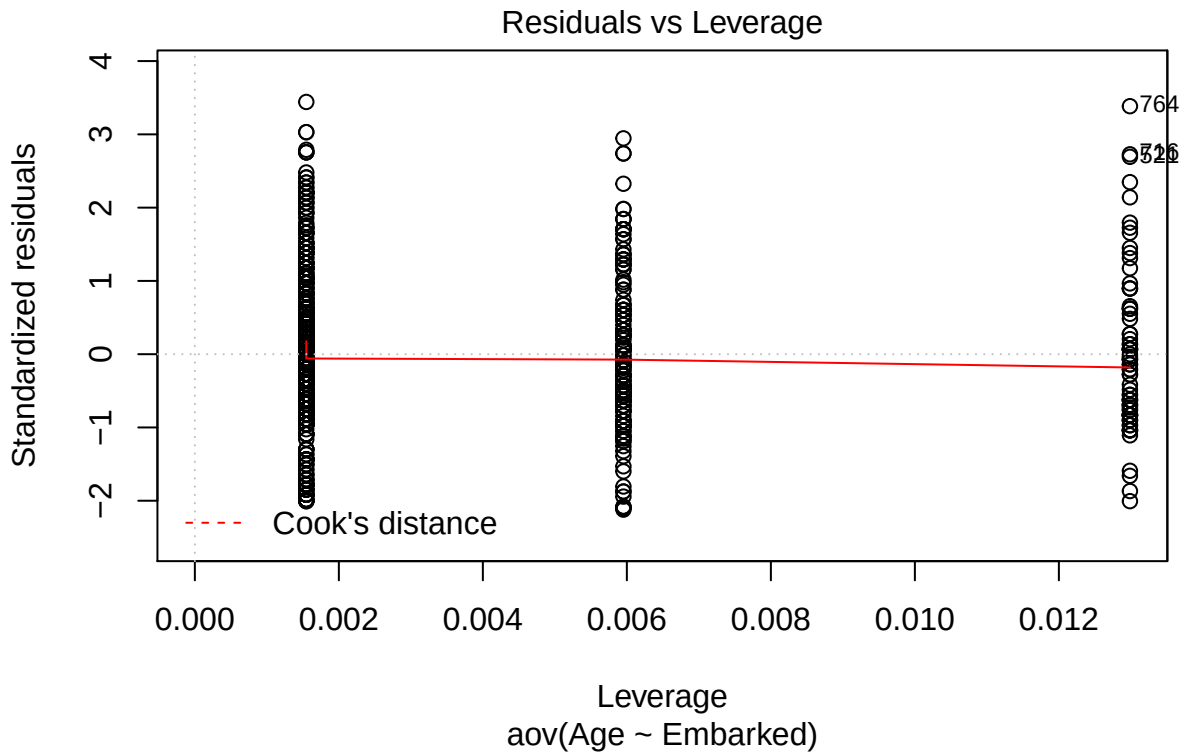
```
plot(modelo_anova, 3)
```



La gráfica Scale-Location también es conocida con el nombre de gráfica Spread-Location. Aquí vemos si los residuos se reparten de forma equitativa.En nuestro ejemplo, vemos casi una recta horizontal y los puntos se distribuyen de forma muy similar a los lados de la línea y en los extremos también parece un gráfico simétrico (tanto en horizontal como en vertical). Es decir que el gráfico nos está indicando que se cumple la homocedasticidad en nuestro modelo.

- Gráfica Residuals vs Leverage

```
plot(modelo_anova, 5)
```



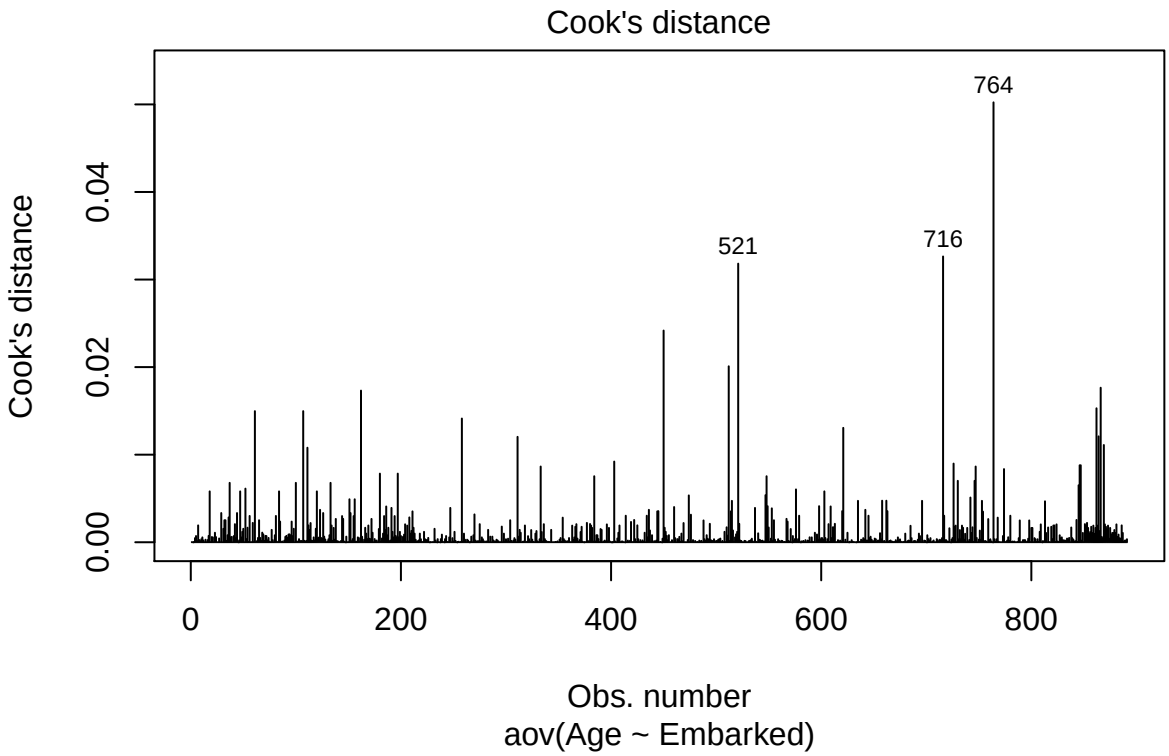
Esta gráfica sirve para ayudarnos a encontrar casos influyentes (es decir, sujetos) si es que los hay. No todos los valores atípicos son influyentes en el análisis de regresión lineal. En el caso de existir valores extremos (outliers), puede que no sean influyentes para determinar la línea de regresión. Eso significa que los resultados no serían muy diferentes si los incluimos o excluimos del análisis, es decir no son influyentes. Por otra parte, algunos casos podrían ser muy influyentes, incluso aunque parezca que están dentro de un rango razonable de los valores. Pueden ser casos extremos contra una línea de regresión y pueden alterar los resultados si los excluimos del análisis. Es decir, estos no están alineados (tendencia) con la mayoría de los casos.

Tenemos que buscar casos que estén fuera de una línea discontinua (que es la distancia de Cook). Cuando encontremos casos fuera de la distancia de Cook (es decir que tienen puntuación alta de distancia de Cook) se consideran casos influyentes en los resultados de la regresión. Y esa regresión se verá afectada si excluimos esos casos.

- Grafica Distancia de Cook

Otra forma de mostrar la distancia de Cook es usando el gráfico que proporciona Anova

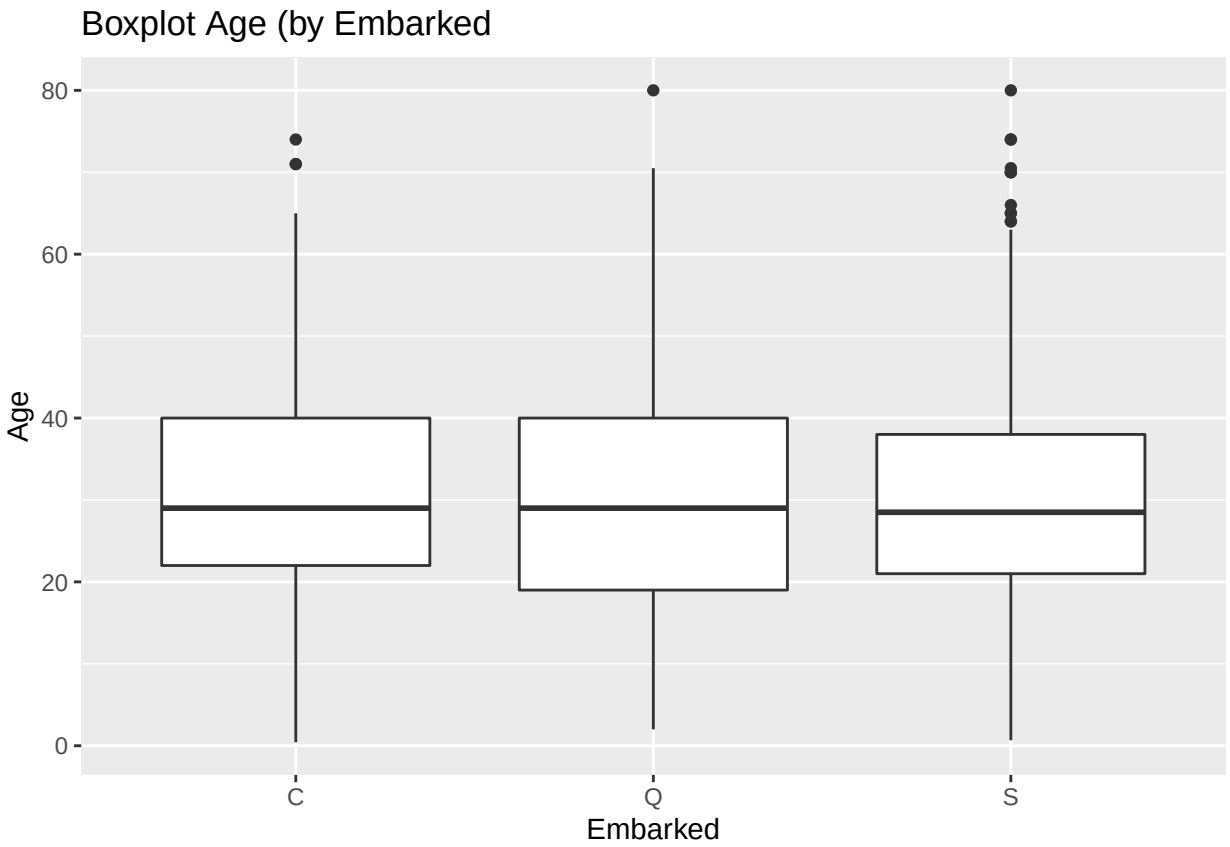
```
plot(modelo_anova, 4)
```



Después de este análisis y dando por aceptadas las condiciones del cumplimiento de Anova, podemos afirmar que las medias de edad en los diferentes puertos de Embarke es coincidente. Y de hecho, si vemos el Boxplot de la variable "Age" en función de la variable "Embarked":

TODO####################Hue_pal

```
ggplot(data, aes(x=Embarked, y=Age)) +
  geom_boxplot() +
  #geom_boxplot(fill = (hue_pal()(3))
  ggtitle("Boxplot Age (by Embarked")
```



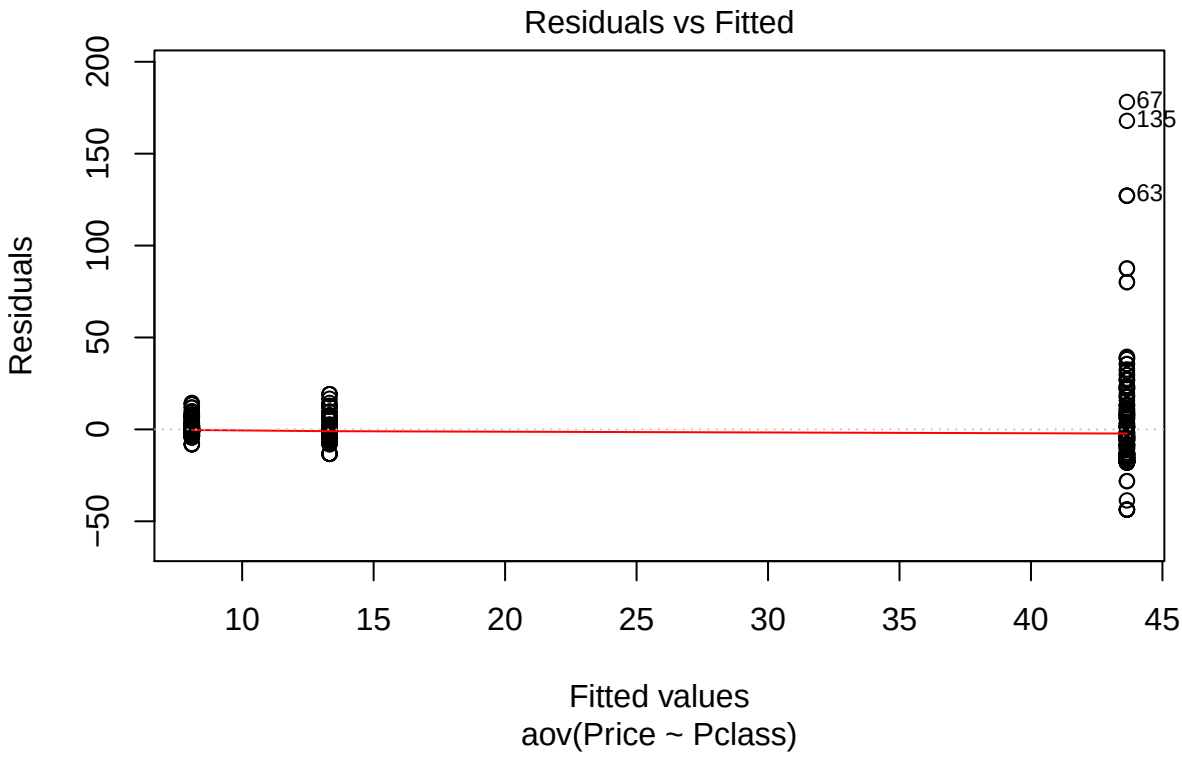
4.3.1.8 Price y Pclass Intentamos realizar Anova con etas dos variables, usando como hipótesis nula que el precio medio pagado es igual para cada una de las 3 clases

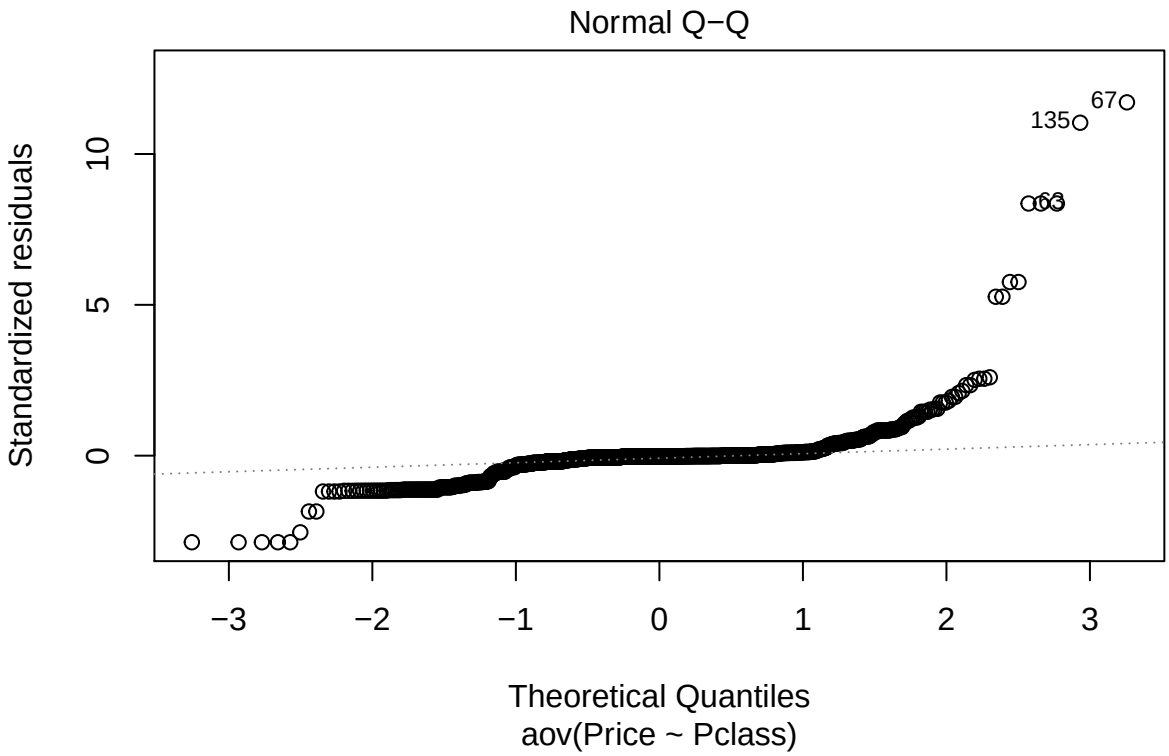
```
modelo_anova <- aov(formula = Price ~ Pclass, data =data )
resumen_anova <- summary(modelo_anova)
resumen_anova
```

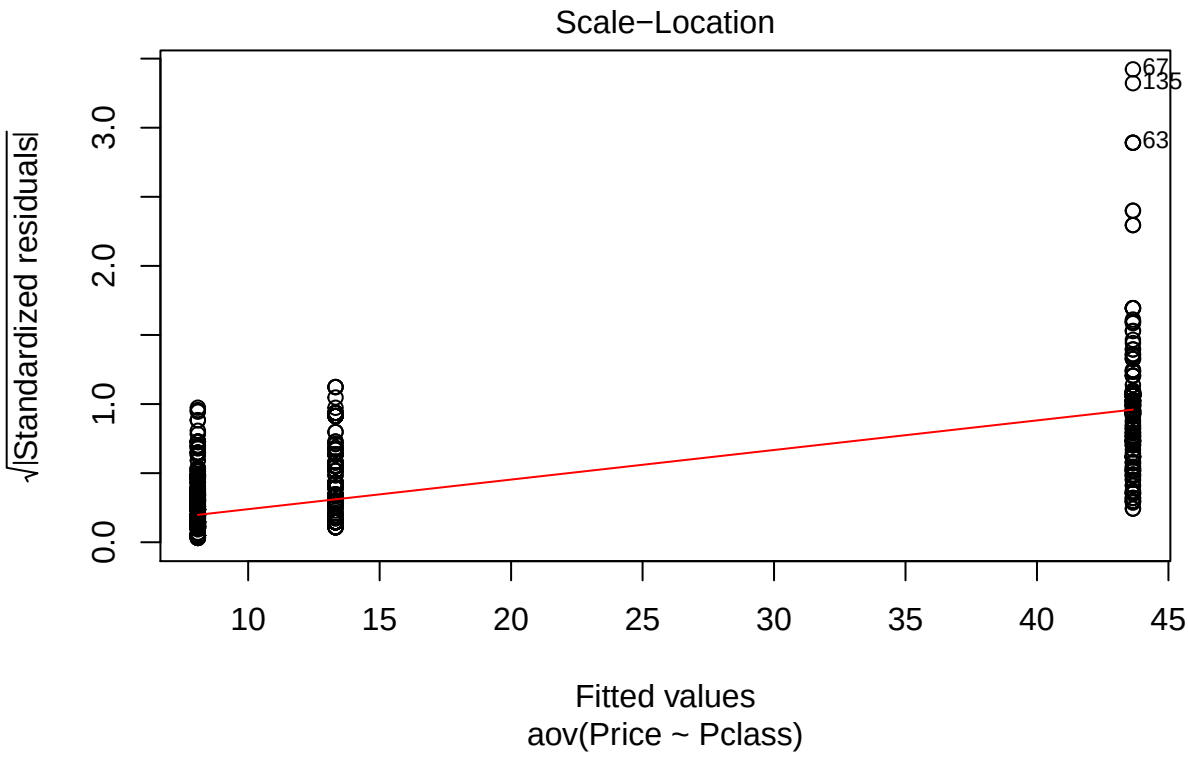
```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Pclass        2 194362   97181   418.3 <2e-16 ***
## Residuals   888 206326     232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

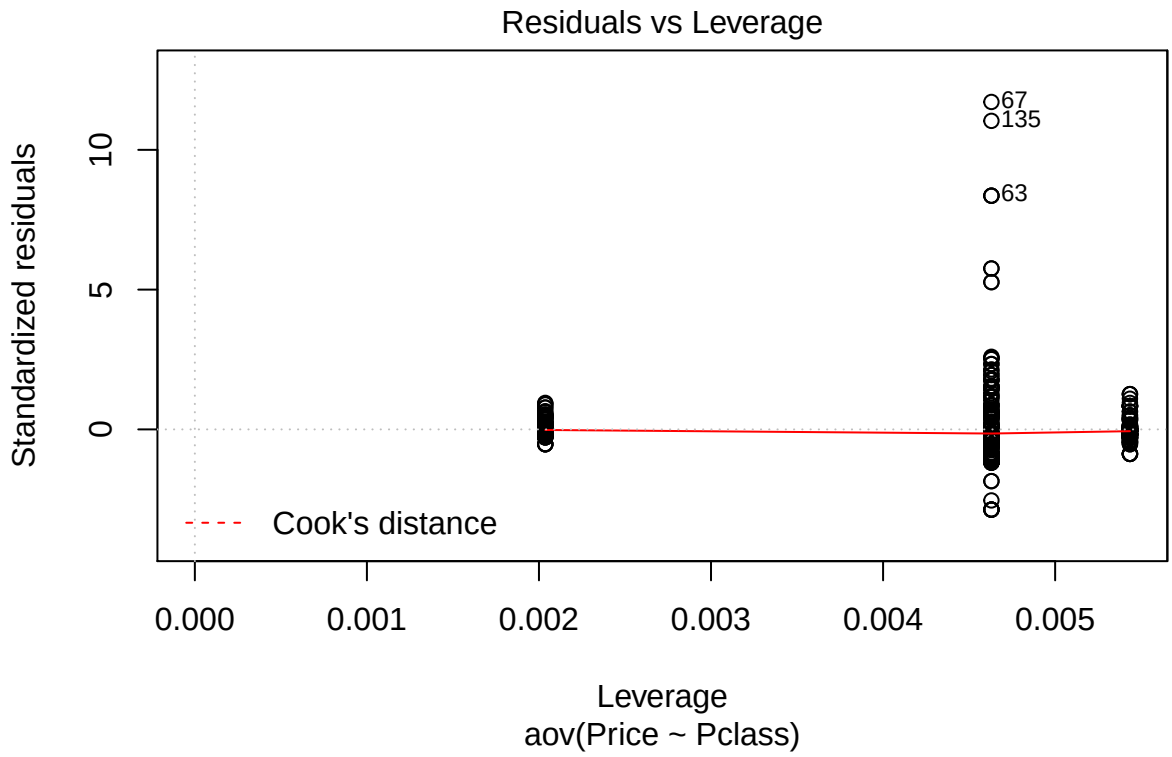
Si mostramos los gráficos:

```
plot(modelo_anova)
```


Residuals vs Fitted
Residuals
200
150
100
50
0
−50
67
135
63
10
15
20
25
30
35
40
45
Fitted values
aov(Price ~ Pclass)


Normal Q−Q
Standardized residuals
0
5
10
−3
−2
−1
0
1
2
3
67
135
Theoretical Quantiles
aov(Price ~ Pclass)


Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
67
135
63
10
15
20
25
30
35
40
45
Fitted values
aov(Price ~ Pclass)



Observamos que no se cumplen las condicies para aplicar Anova y el test no tienve valor. Vamos a intentar usar un test no paramétrico, Kruskal Wallis

```
modelo_kruskal <- kruskal.test(formula = Price ~ Pclass, data =data )
modelo_kruskal
```

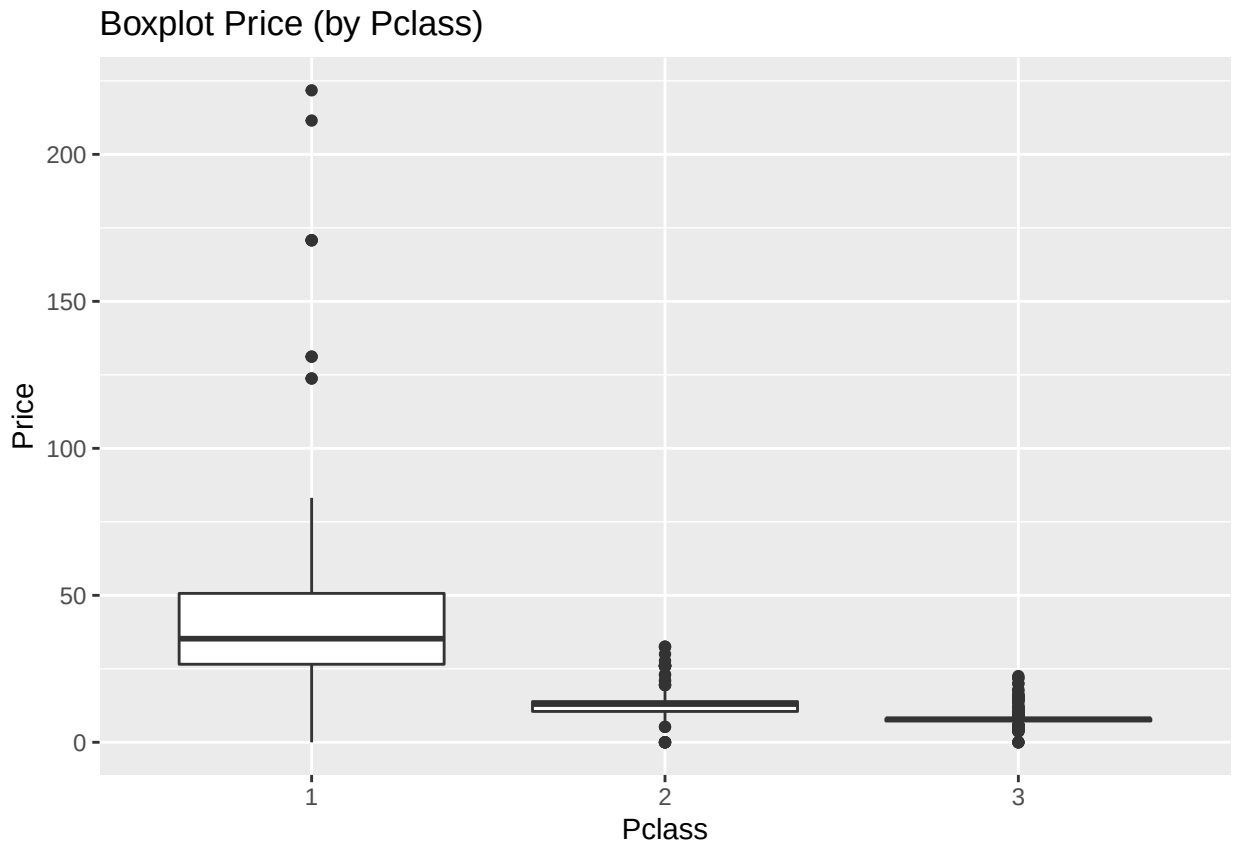
```
##
##  Kruskal-Wallis rank sum test
##
## data:  Price by Pclass
## Kruskal-Wallis chi-squared = 566.65, df = 2, p-value < 2.2e-16
```

El resultado de este test arroja un p-value inferior a 0.05 y por tanto rechazamos la hipótesis nula.

Lo podemos comprobar también visualmente con el boxplot relativo a ello:

TODO####################Hue_pal

```
ggplot(data, aes(x=Pclass, y=Price)) +
  geom_boxplot() +
  #geom_boxplot(fill = (hue_pal()(3))
  ggtitle("Boxplot Price (by Pclass)")
```



4.3.2 Modelos

TODO########### explicación mejor de los modelos Vamos a aplicar diversos modelos

Vamos a utilizar como variables explicativas las variables que traía por defecto el conjunto de datos original (menos las que hemos eliminado en el proceso de transformación y limpieza de datos ya comentado).

Creamos un conjunto de entrenamiento y un conjunto de test. Es decir,vamos a realizar una partición, un 75% de datos para el entrenamiento, y el 25% restante para validar los modelos.

```
set.seed(123)
library(caTools)
split = sample.split(data$Survived, SplitRatio = 0.75)
data_train = subset(data, split == TRUE)
data_test = subset(data, split == FALSE)
```

4.3.2.1 Modelo de regresión logística

Un modelo de regresión logística es un tipo de análisis de regresión que se utiliza para predecir el resultado de una variable categórica, en función de las variables independientes. En nuestro caso, la variable objetivo "Survived" es una variable categórica binaria, es decir que tomar valor de 0 o 1 (verdadero o falso). La variable toma el valor 1 cuando el pasajero ha sobrevivido al accidente, y 0 en caso contrario.

- Primer modelo glm

Creamos un primer modelo glm1 con las variables "original": Pclass, SibSp, Parch, Sex, Age, Fare y Embarked

```
modelo_glm1 <- glm(formula=Survived~ Pclass+SibSp+Parch+Sex+Age+Fare+Embarked, data = data_train, family =
summary(modelo_glm1)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + SibSp + Parch + Sex + Age +
##     Fare + Embarked, family = binomial(link = "logit"), data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4814  -0.6618  -0.4308   0.6760   2.3469
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.610350   0.503872   7.165 7.77e-13 ***
## Pclass2     -0.814859   0.334335  -2.437   0.0148 *
## Pclass3     -1.925117   0.332006  -5.798 6.69e-09 ***
## SibSp       -0.267357   0.128522  -2.080   0.0375 *
## Parch       -0.171750   0.140492  -1.222   0.2215
## Sexmale     -2.554317   0.222473 -11.481  < 2e-16 ***
## Age         -0.034588   0.008142  -4.248 2.16e-05 ***
## Fare         0.002462   0.002599   0.947   0.3435
## EmbarkedQ    0.041847   0.433691   0.096   0.9231
## EmbarkedS   -0.271965   0.272774  -0.997   0.3187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 889.27  on 667  degrees of freedom
## Residual deviance: 620.02  on 658  degrees of freedom
## AIC: 640.02
##
## Number of Fisher Scoring iterations: 4
```

Se puede observar en este modelo que tanto las variables "Fare" como "Parch" no son significativas, es decir, no están aportando nada a la hora de predecir la variable "Survived". Además, podemos comprobar el "sentido" de la significación. Por ejemplo, vemos como significativa la variable dummy. Con las variables categóricas se crean automáticamente tantas variables dummy como niveles – 1. En el caso de la variable Sex, al tener 2 valores ha creado la variable "Sexmale", es decir la parte correspondiente a hombres, mientras que la parte de mujeres forma parte del Intercept del modelo. Concretamente vemos que "Sexmale" es una variable significativa, pero con valor negativo (de las más negativas junto con "Pclass3"). Eso significa que esas variables son una influencia "negativa" de cara a la supervivencia. Es decir, esas variables contribuyen negativamente a la supervivencia, se puede ver por ejemplo que Pclass3 afecta más negativamente que Pclass2.

Ahora usamos el conjunto de datos de test para validar nuestro modelo

TODO################NO consigo que funcione el predict y el crosttable

```
predict_glm1 <- predict.glm(modelo_glm1, data_test, type = 'terms')
```

```
#csstab_glm1 <- CrossTable(data_test$Survived, predict_glm1,
#           prop.chisq = FALSE,
#           proc.c = FALSE,
#           prop.r = FALSE,
```

```
#          dnn = c('Reality', 'Prediction'))
```

- Segundo modelo glm

Construimos otro modelo (glm2), similar al anterior pero esta vez partiendo de los resultados del modelo anterior, le vamos a quitar aquellas variables que vimos que no eran significativas para el modelo de predicción. Por lo tanto, tendremos un modelo2 cuyas variables a utilizar serán Pclass, SibSp, Sex y Age.

```
modelo_glm2 <- glm(formula=Survived~ Pclass+SibSp+Sex+Age, data = data_train, family = binomial(link = "logit"))
summary(modelo_glm2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + SibSp + Sex + Age, family = binomial(link = "logit"),
##     data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5489  -0.6544  -0.4349   0.6748   2.5000
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.594507   0.425240   8.453  < 2e-16 ***
## Pclass2     -1.039546   0.292339  -3.556 0.000377 ***
## Pclass3     -2.121255   0.272594  -7.782 7.15e-15 ***
## SibSp       -0.316196   0.121907  -2.594 0.009494 **
## Sexmale     -2.533211   0.213842 -11.846  < 2e-16 ***
## Age         -0.034695   0.008056  -4.307 1.66e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 889.27  on 667  degrees of freedom
## Residual deviance: 624.38  on 662  degrees of freedom
## AIC: 636.38
##
## Number of Fisher Scoring iterations: 4
```

Se comprueba que todas las variables utilizadas son variables significativas para el modelo. Además, el modelo aprece haber mejorado un poco en cuanto a la predicción:

TODO######Código para predict y crosstable

- Tercer modelo glm

Hasta ahora se han utilizado las variables del conjunto de datos "original". Vamos a construir otro modelo (glm3), pero esta vez vamos a intentar utilizar alguna de las variables que hemos construido para ayudar a predecir la supervivencia. Para este caso vamos a utilizar las variables nuevas Child y también FamilySize. Además de esas dos variables, vamos a utilizar otras tres variables originales: Pclass, Sex y Age.

```
modelo_glm3 <- glm(formula=Survived~ Sex+Pclass+Age+Child+FamilySize,, data = data_train, family = binomial(link
summary(modelo_glm3)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + Age + Child + FamilySize,
```

```
##     family = binomial(link = "logit"), data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8182  -0.6406  -0.4422   0.6299   2.4586
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.614517   0.479156   7.544 4.57e-14 ***
## Sexmale     -2.677915   0.224435 -11.932  < 2e-16 ***
## Pclass2     -1.029310   0.294091  -3.500 0.000465 ***
## Pclass3     -2.056783   0.273247  -7.527 5.18e-14 ***
## Age         -0.021819   0.009044  -2.412 0.015848 *
## Child1       1.687212   0.534992   3.154 0.001612 **
## FamilySize  -0.326523   0.088286  -3.698 0.000217 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 889.27  on 667  degrees of freedom
## Residual deviance: 612.49  on 661  degrees of freedom
## AIC: 626.49
##
## Number of Fisher Scoring iterations: 5
```

Al igual que pasaba en el modelo2, todas las variables utilizadas han resultado significativas, y además el AIC del modelo ha mejorado ligeramente. Además la capacidad predictiva del modelo ha aumentado un poco:

TODO###Código matriz confusion predict y crosstable

- Cuarto modelo glm

Antes de construir un cuarto modelo, vamos a analizar si existen variables independientes que afecten a la variable dependiente. Para ello primero definimos una variable 'objetivo' pero numérica

```
data$SurvivedNum <- as.integer(as.character(data$Survived))
```

Agrupamos el dataset por las variables Sex y Pclass

```
data.agrup <- data %>% group_by(Sex, Pclass)
```

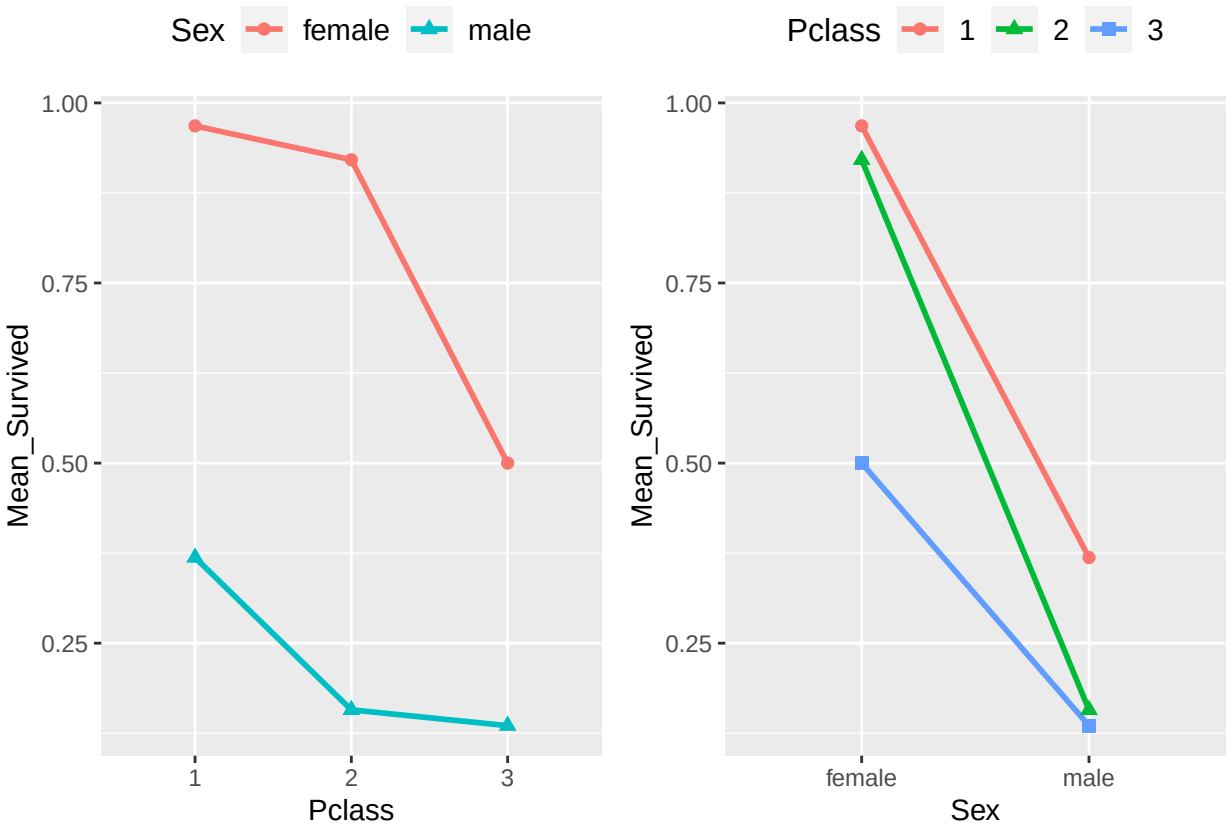
Calculamos la media creando un campo 'Mean_survived' de la media

```
data.mediasurvived <- summarize(.data = data.agrup, Mean_Survived = mean(SurvivedNum))
kable(data.mediasurvived)
```

Sex	Pclass	Mean_Survived
female	1	0.9680851
female	2	0.9210526
female	3	0.5000000
male	1	0.3688525
male	2	0.1574074
male	3	0.1354467

```
plot1 <- ggplot(data = data.mediasurvived, aes(x=Pclass, y = Mean_Survived, group = Sex)) +
  geom_line(aes(color=Sex), size=1) +
```

```
  geom_point(aes(color = Sex, shape=Sex), size=2)+
  theme(legend.position="top") +
  theme(legend.title = element_text(size=12), legend.text = element_text(size=11))
plot2 <- ggplot(data = data.mediasurvived, aes(x=Sex, y = Mean_Survived, group = Pclass)) +
  geom_line(aes(color=Pclass), size=1) +
  geom_point(aes(color = Pclass, shape=Pclass), size=2)+
  theme(legend.position="top") +
  theme(legend.title = element_text(size=12), legend.text = element_text(size=11))
grid.arrange(plot1, plot2, ncol = 2)
```



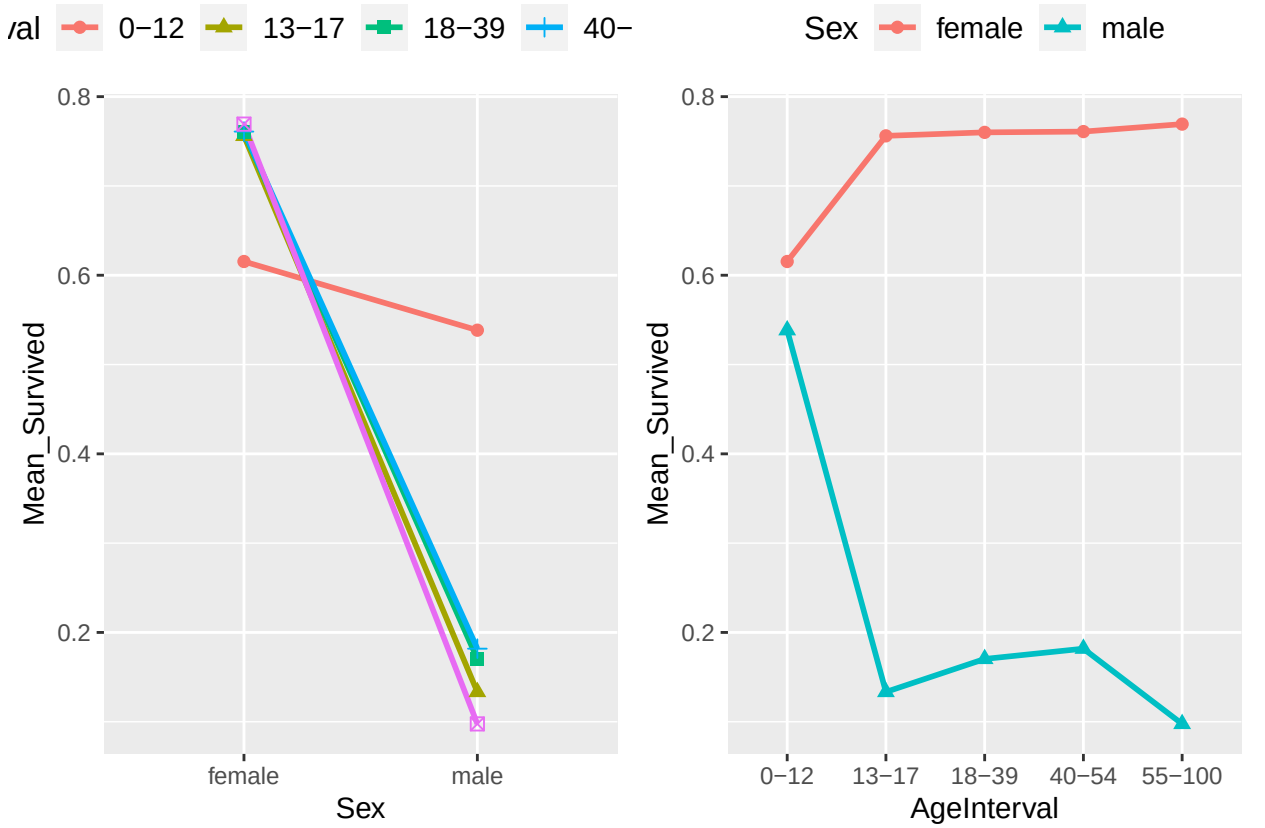
Tanto la variable Sex como la variable Pclass producen efecto en la variable Mean_Survived. También podemos comprobar cómo se produce interacción entre las variables Sex y Pclass respecto a Mean_Survived. Podemos ver cómo hay un descenso muy pronunciado cuando pasamos de mujeres a hombres y estamos tratando la clase 2ª. Observamos además cómo pasar a tercera clase afecta muy negativamente en el caso de las mujeres, ya que aunque no es una línea paralela, el hecho de estar en 1ª o 2ª clase no parece demasiado importante en el caso de las mujeres, pero al pasar a 3ª clase la caída de la media de supervivencia es muy importante.

Agrupamos el dataset por las variables AgeInterval y Sex y realizamos el mismo proceso

```
data.agrup <- data %>% group_by(AgeInterval, Sex)

data.mediasurvived <- summarize(.data = data.agrup, Mean_Survived = mean(SurvivedNum))
kable(data.mediasurvived)
```

AgeInterval	Sex	Mean_Survived
0-12	female	0.6153846
0-12	male	0.5384615
13-17	female	0.7560976
13-17	male	0.1333333
18-39	female	0.7600000
18-39	male	0.1703297
40-54	female	0.7608696
40-54	male	0.1818182
55-100	female	0.7692308
55-100	male	0.0975610

```
plot1 <- ggplot(data = data.mediasurvived, aes(x=Sex, y = Mean_Survived, group = AgeInterval)) +
  geom_line(aes(color=AgeInterval), size=1) +
  geom_point(aes(color = AgeInterval, shape=AgeInterval), size=2)+
  theme(legend.position="top") +
  theme(legend.title = element_text(size=12), legend.text = element_text(size=11))
plot2 <- ggplot(data = data.mediasurvived, aes(x=AgeInterval, y = Mean_Survived, group = Sex)) +
  geom_line(aes(color=Sex), size=1) +
  geom_point(aes(color = Sex, shape=Sex), size=2)+
  theme(legend.position="top") +
  theme(legend.title = element_text(size=12), legend.text = element_text(size=11))
grid.arrange(plot1, plot2, ncol = 2)
```



En este caso llama la atención el impacto de cuando pasamos el primer intervalo (menores o iguales a 12 años) al siguiente intervalo (de 13 a 17 años). La proporción de las mujeres aumenta mientras que en el caso de los hombres disminuye drásticamente. Además, no parece haber apenas diferencia en el primer tramo (los

niños, se salvan con independencia del género)

Ahora construimos nuestro cuarto modelo de regresión logística. Utilizaremos algunas variables ya usadas en otros modelos más AgeInterval:Sex. También hemos añadido la interacción que generan Sex:Pclass. A estas variables le sumamos también las variables independientes Sex, Parch y SibSp.

```
modelo_glm4 <- glm(formula=Survived~Sex+Parch+SibSp+AgeInterval:Sex+Sex:Pclass, data = data_train, family = bin
summary(modelo_glm4)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Parch + SibSp + AgeInterval:Sex +
##     Sex:Pclass, family = binomial(link = "logit"), data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4625  -0.5615  -0.4443   0.4256   2.3716
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      3.8805     0.8341   4.652 3.28e-06 ***
## Sexmale                         -0.7252     0.9532  -0.761  0.44679
## Parch                           -0.2661     0.1641  -1.622  0.10490
## SibSp                           -0.4531     0.1434  -3.159  0.00158 **
## Sexfemale:AgeInterval13-17       0.2816     0.6811   0.413  0.67931
## Sexmale:AgeInterval13-17        -3.0880     0.7054  -4.377 1.20e-05 ***
## Sexfemale:AgeInterval18-39      -0.3339     0.5486  -0.609  0.54272
## Sexmale:AgeInterval18-39        -3.3368     0.5618  -5.940 2.86e-09 ***
## Sexfemale:AgeInterval40-54      -0.8981     0.7300  -1.230  0.21861
## Sexmale:AgeInterval40-54        -3.7725     0.6519  -5.787 7.16e-09 ***
## Sexfemale:AgeInterval55-100     -1.1544     1.2504  -0.923  0.35591
## Sexmale:AgeInterval55-100       -4.3193     0.8671  -4.981 6.31e-07 ***
## Sexfemale:Pclass2               -0.6262     0.7739  -0.809  0.41847
## Sexmale:Pclass2                 -1.5862     0.4124  -3.846  0.00012 ***
## Sexfemale:Pclass3               -3.1721     0.6689  -4.742 2.11e-06 ***
## Sexmale:Pclass3                 -1.6486     0.3160  -5.217 1.82e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 889.27  on 667  degrees of freedom
## Residual deviance: 569.94  on 652  degrees of freedom
## AIC: 601.94
##
## Number of Fisher Scoring iterations: 5
```

TODO##############Matriz de confusión

4.3.2.1.1 Resumen modelos regresion TODO##### Curvas ROC

En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

4.3.2.2 Árboles de decisión

5 Representación de los resultados a partir de tablas y gráficas.

---

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