Práctica 2: Limpieza y validación de los datos

Tania Piñeiro y Jordi Sánchez

09/05/2021

Contents

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

El dataset seleccionado es el dataset "Titanic" de Kaggle. Ese conjunto de datos contiene datos sobre los pasajeros de este transantlántico como su edad, género, la clase en la que viajaba y también incluye información sobre si sobrevivió al fatal accidente.

Como es bien conocido, el 15 de abril de 1912, el transatlántico de pasajeros más grande jamás construido chocó con un iceberg durante su viaje inaugural. Cuando el Titanic se hundió, mató a 1502 de los 2224 pasajeros y tripulación. Una de las razones por las que el naufragio resultó en tal pérdida de vidas fue que no había suficientes botes salvavidas para los pasajeros y la tripulación. Aunque hubo algún elemento de

suerte involucrado en sobrevivir al hundimiento, algunos grupos de personas tenían más probabilidades de sobrevivir que otros.

El análisis de este dataset es importante porque puede ofrecer información sobre si hubo diferencias en la supervivencia de los pasajeros en función de sus características como por ejemplo ¿Hubo una mayor probabilidad de fallecidos entre los pasajeros de las clases más bajas? Los resultados nos permitirán obtener conclusiones valiosas sobre el incidente.

Precisamente éste es el problema que pretende resolver el análisis de este dataset, obtener respuestas sobre las características de los pasajeros con mayores posibilidades de sobrevivir.

En primer lugar, vamos a cargar el dataset y analizar de cuantas variables y registros disponemos para abordar este análisis. Se va a cargar el conjunto "train" de entrenamiento disponible en Kaggle.

```
# Cargamos el juego de datos
ds<- read.csv(file='train.csv',header=T,dec='.', sep=",")
ds_test<- read.csv(file='test.csv',header=T,dec='.', sep=",")
# Verificamos la estructura del conjunto de datos
str(ds)
```

```
## '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)"
##  $ 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  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Se observa que se trata de una base de datos con 891 observaciones y 12 variables en las que se recogen algunas características de los pasajeros. Entre las variables hay 7 variables numéricas y 5 categóricas. A continuación se describen las variables:

- PassengerId (tipo int): código de identificación del pasajero
- Survived (tipo int): informa si el pasajero murió o sobrevivió en el accidente(0 = No, 1 = Si)
- Pclass (tipo int): hace referencia a la clase en la que viajaban los pasajeros (1 = primera clase, 2 = segunda clase...)
- Name (tipo char): nombre del pasajero
- Sex (tipo char): sexo del pasajero
- Age (tipo int): edad del pasajero
- SibSp (tipo int): número de familiares (hermanos o esposa) a bordo del Titanic
- Parch (tipo int): número de familiares (padres o hijos) a bordo del Titanic
- Ticket (tipo char): código del ticket del pasajero
- Fare (tipo num): precio del ticket para viajar en el Titanic
- Cabin (tipo char): número del camarote
- Embarked (tipo char): puerto de embarque (C = Cherbourg, Q = Queenstown, S = Southampton)

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

En el apartado inicial se ha realizado un análisis preliminar de los datos de los que disponemos en el dataset, obteniendo que tiene un total de 12 columnas y 891 registros. Sin embargo, llegado este punto debemos determinar si realmente necesitamos para nuestro análisis todas esas variables.

Se ha considerado que hay algunas variables que no van a aportar demasiada información al modelo como son: el código del ticket ("Ticket) y el puerto de embarque ("Embarked"). Sin embargo la variable"Ticket" nos permite conocer cuantas personas han comprado el billete con el mismo ticket y de esta forma, calcular el precio por persona de la variable "Fare", que estará agrupada. La variable "Embarked" se va a excluir del análisis. Además, se ha observado que el número de camarote ("Cabin") está ausente en un gran número de registros, sin embargo, por el momento se va a conservar para analizar si se pueden obtener algunas relaciones entre la ubicación del camarote y el desenlace de los pasajeros. El nombre de los pasajeros ("Name"), a priori puede parecer poco relevante pero como incluye el título del pasajero ("Mr., Miss...") se van a conservar por si pudiese ser de utilidad.

De esta forma el dataset que vamos a pre-procesar y analizar contará con 11 columnas y 891 registros.

```
# Selección de las variables de interés
ds<-select(ds, "PassengerId","Name","Survived","Pclass", "Sex","Age","SibSp", "Parch", "Fare", "Cabin",
```

Además, para facilitar el trabajo con estos datos vamos a convertir a factores las variables "Pclass" y "Sex" empleando la función factor. Esto nos facilitará encontrar valores erróneos en futuros pasos y el tratamiento de estas variables para las representaciones gráficas.

```
#La función factor convierte a factor las variables seleccionadas
ds$Pclass<-factor(ds$Pclass)
ds$Sex<-factor(ds$Sex)
ds$Parch<-factor(ds$Parch)
ds$SibSp<-factor(ds$SibSp)
```

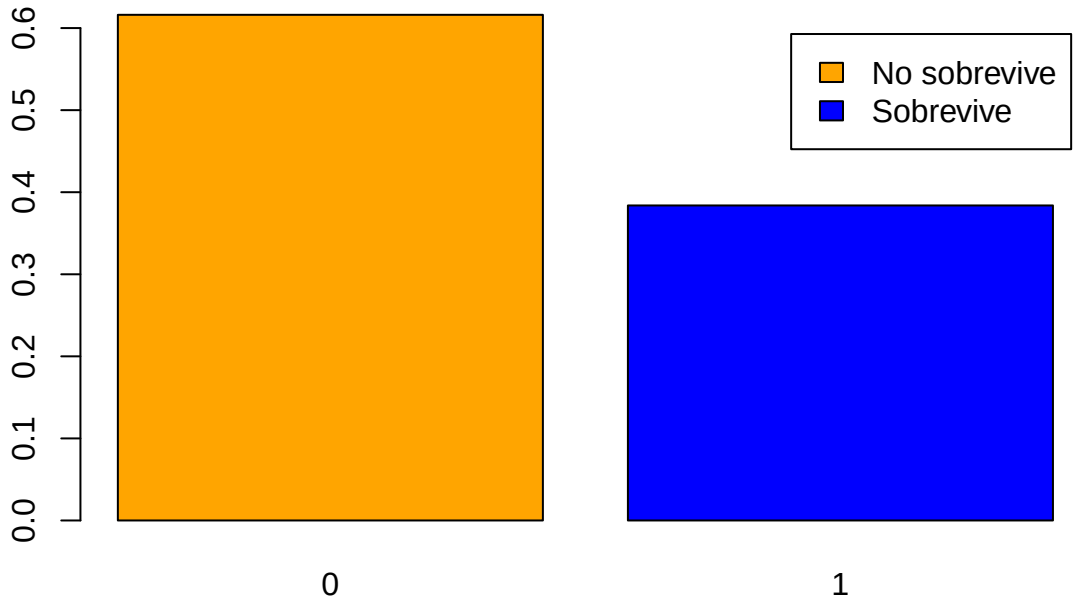
Finalmente, se van a obtener las primeras estadísticas descriptivas del conjunto de datos para empezar a conocer más en profundidad como era la distribución de los pasajeros.

```
summary(ds)
```

```
##   PassengerId        Name              Survived      Pclass      Sex
##  Min.   :  1.0   Length:891         Min.   :0.0000   1:216   female:314
##  1st Qu.:223.5   Class :character   1st Qu.:0.0000   2:184   male  :577
##  Median :446.0   Mode  :character   Median :0.0000   3:491
##  Mean   :446.0                      Mean   :0.3838
##  3rd Qu.:668.5                      3rd Qu.:1.0000
##  Max.   :891.0                      Max.   :1.0000
##
##       Age        SibSp   Parch        Fare           Cabin
##  Min.   : 0.42   0:608   0:678   Min.   :  0.00   Length:891
##  1st Qu.:20.12   1:209   1:118   1st Qu.:  7.91   Class :character
##  Median :28.00   2: 28   2: 80   Median : 14.45   Mode  :character
##  Mean   :29.70   3: 16   3:  5   Mean   : 32.20
##  3rd Qu.:38.00   4: 18   4:  4   3rd Qu.: 31.00
##  Max.   :80.00   5:  5   5:  5   Max.   :512.33
##  NA's   :177     8:  7   6:  1
##     Ticket
```

```
##  Length:891
##  Class :character
##  Mode  :character
##
##
##
##
```

```
barplot(prop.table(table(ds$Survived)),col=c("orange","blue"),
        legend.text=c("No sobrevive","Sobrevive"))
```



De los resultados obtenidos se obtienen las siguientes conclusiones:

- De los 891 pasajeros, 549 no sobrevivieron al accidente y 342 sí lo hicieron.
- De los 891 pasajeros, 216 iban en primera clase, 184 en segunda y 491 en tercera.
- De los 891 pasajeros, 314 eran mujeres y 577 hombres.
- La edad de los pasajeros se encuentra entre los 0.42 (posible error) y los 80 años, siendo la edad media 29.7
- Los pasajeros tenían de 0-8 hermanos/esposas a bordo y entre 0-6 hijos/padres.

3. Limpieza de los datos

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

El siguiente paso consistirá en determinar si hay valores nulos y si es así eliminarlos o sustituirlos por otros.

```
# Estadísticas de valores vacíos
colSums(is.na(ds)) %>% kable(caption="Número de NAs por columna") %>% kable_styling(latex_options = "ho
```

Table 1: Número de NAs por columna

	x
PassengerId	0
Name	0
Survived	0
Pclass	0
Sex	0
Age	177
SibSp	0
Parch	0
Fare	0
Cabin	0
Ticket	0

```
colSums(ds=="", na.rm = TRUE) %>% kable(caption="Número de valores vacíos por columna") %>% kable_stylin
```

Table 2: Número de valores vacíos por columna

	x
PassengerId	0
Name	0
Survived	0
Pclass	0
Sex	0
Age	0
SibSp	0
Parch	0
Fare	0
Cabin	687
Ticket	0

En las tablas superiores se han obtenido el número de registros con valor "NA" para cada columna y también el número de valores vacíos. En la primera tabla se observa que la única columna se muestra que la única variable que presenta valores ausentes es "Age" con 177 registros NA.

En la segunda tabla se observa que la única variable que tiene valores vacíos es "Cabin" con 687 registros vacíos. Este es un número muy elevado teniendo en cuenta que nuestro dataset cuenta con 897 registros, significa que el 77% de los registros de esta columna no están disponibles.

En cuanto a la gestión que se va a hacer de estos valores faltantes hemos determinado conservar la variable "Age" ya que consideramos que puede ser muy relevante para el análisis. Se va a realizar una imputación de los valores faltantes, obteniendo la edad media para cada uno de los grupos de títulos de los pasajeros

("Mr, Miss"). Es decir, para una edad faltante de una pasajera tipo "Miss" se le inputará la edad promedio de todas las pasajeras tipo "Miss" del dataset.

Por otra parte, se decide finalmente eliminar la variable "Cabin" ya que se considera que un 77% de datos faltantes es demasiado elevado como para intentar algún tipo de imputación y se podría introducir error.

```
# Se elimina la variable "Cabin"
ds <- ds[,-(10)]
```

Como se ha comentado la edad se va a estimar en función del título del pasajero. Para ello es necesario, previamente separar el título de la variable "Name" y recogerlo en una variable independiente "Title". Para realizar esta separación se va a emplear una expresión regular, aprovechando que el título aparece tras una coma y un espacio y antes de un punto (Ej: "Braund, Mr. Owen Harris"). A continuación se va a identificar cuántos tipos de títulos hay y su recuento.

```
# Separación del título del pasajero en una nueva columna
ds$Title <- gsub('(.*, )|(\\..*)', '', ds$Name)

table(ds$Title)
```

```
##
##         Capt          Col          Don           Dr     Jonkheer         Lady
##            1            2            1            7            1            1
##        Major       Master         Miss         Mlle          Mme           Mr
##            2           40          182            2            1          517
##          Mrs           Ms          Rev          Sir the Countess
##          125            1            6            1            1
```

Se obtiene que hay un total de 17 títulos diferente, aunque la mayoría de ellos son poco frecuentes y solo se encuentran en un par de registros como "Capt", "Major". Estos títulos poco frecuentes se van a agrupar en un único factor, de forma que finalmente habrá 5 niveles: "Miss", "Master", "Mrs", "Mr" y "Other".

```
# Se agrupan los títulos poco frecuentes
other <- c('Dona', 'Lady', 'the Countess','Capt', 'Col', 'Don',
                'Dr', 'Major', 'Rev', 'Sir', 'Jonkheer', 'Mlle', 'Ms', 'Mme', 'Lady')
ds$Title[ds$Title %in% other]  <- 'Other'
# Se eliminan los niveles no empleados
ds$Title<-factor(ds$Title)
table(ds$Title)
```

```
##
## Master   Miss     Mr    Mrs  Other
##     40    182    517    125     27
```

```
ds_test$Title <- gsub('(.*, )|(\\..*)', '', ds_test$Name)
# Se agrupan los títulos poco frecuentes
ds_test$Title[ds_test$Title %in% other]  <- 'Other'
# Se eliminan los niveles no empleados
ds_test$Title<-factor(ds_test$Title)
```

Ahora observamos que hemos obtenido los cinco grupos de títulos de los pasajeros. Vamos a aprovechar esta información para realizar la inputación de los valores faltantes como el valor promedio de estos grupos.

```r
# Se localizan los na de las variables Weight y Height
i_na<-is.na(ds$Age)
dcomplete<-ds[!i_na,]
# Se estima el promedio de edad para cada uno de los grupos
av_Age_title<-aggregate(x = dcomplete$Age, by = list(dcomplete$Title), FUN = mean)
# Imputación
ds[which(is.na(ds$Age)&ds$Title=="Master"),"Age"] <- round(av_Age_title[1,2])
ds[which(is.na(ds$Age)&ds$Title=="Miss"),"Age"] <- av_Age_title[2,2]
ds[which(is.na(ds$Age)&ds$Title=="Mr"),"Age"] <- av_Age_title[3,2]
ds[which(is.na(ds$Age)&ds$Title=="Mrs"),"Age"] <- av_Age_title[4,2]
ds[which(is.na(ds$Age)&ds$Title=="Other"),"Age"] <- av_Age_title[5,2]
colSums(is.na(ds))
```

```
## PassengerId        Name    Survived      Pclass         Sex         Age
##           0           0           0           0           0           0
##       SibSp       Parch        Fare      Ticket       Title
##           0           0           0           0           0
```

```r
# Se localizan los na de las variables Weight y Height
i_na<-is.na(ds_test$Age)
dcomplete<-ds_test[!i_na,]
# Se estima el promedio de edad para cada uno de los grupos
av_Age_title<-aggregate(x = dcomplete$Age, by = list(dcomplete$Title), FUN = mean)
# Imputación
ds_test[which(is.na(ds_test$Age)&ds_test$Title=="Master"),"Age"] <- round(av_Age_title[1,2])
ds_test[which(is.na(ds_test$Age)&ds_test$Title=="Miss"),"Age"] <- av_Age_title[2,2]
ds_test[which(is.na(ds_test$Age)&ds_test$Title=="Mr"),"Age"] <- av_Age_title[3,2]
ds_test[which(is.na(ds_test$Age)&ds_test$Title=="Mrs"),"Age"] <- av_Age_title[4,2]
ds_test[which(is.na(ds_test$Age)&ds_test$Title=="Other"),"Age"] <- av_Age_title[5,2]
colSums(is.na(ds_test))
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0           0           0
##       Parch      Ticket        Fare       Cabin    Embarked       Title
##           0           0           1           0           0           0
```

Comprobamos que tras realizar la imputación de valores a la variable "Age" ya no hay ningun registro con valores NA. De esta forma confirmamos que se ha realizado adecuadamente la imputación de valores. Se podrían haber realizado otros procedimientos para el tratamiento de estos datos faltantes como la eliminación de toda la fila donde se encuentre un registro faltante o la imputación directamente de la edad promedio global. Sin embargo, el número de valores faltantes (177) se consideró demasiado elevado como para eliminar los registros, ya que se perdería gran cantidad de información. Además, la variable age tiene un rango bastante amplio (0.4 - 80 años) de forma que si imputásemos el promedio global de todos los pasajeros probablemente el error que estaríamos introduciendo sería mayor.

3.2. Identificación y tratamiento de valores extremos.

Para el tratamiento de valores extremos trabajaremos con las variables "Age", "Fare", "SibSp" y "Parch", ya que son las únicas variables númericas de las que disponemos en el dataset.

Sin embargo, para empezar, verificaremos que entre las variables "Survived", "Pclass", "Sex", "SibSp" y "Parch" que son tipo factor no hay ningún nivel que pueda ser anómalo.

```r
# Comprobación de las variables tipo factor
summary(ds[c("Survived", "Pclass", "Sex", "SibSp","Parch")])
```

```
##     Survived      Pclass      Sex      SibSp   Parch
##  Min.   :0.0000   1:216   female:314   0:608   0:678
##  1st Qu.:0.0000   2:184   male  :577   1:209   1:118
##  Median :0.0000   3:491                2: 28   2: 80
##  Mean   :0.3838                        3: 16   3:  5
##  3rd Qu.:1.0000                        4: 18   4:  4
##  Max.   :1.0000                        5:  5   5:  5
##                                        8:  7   6:  1
```

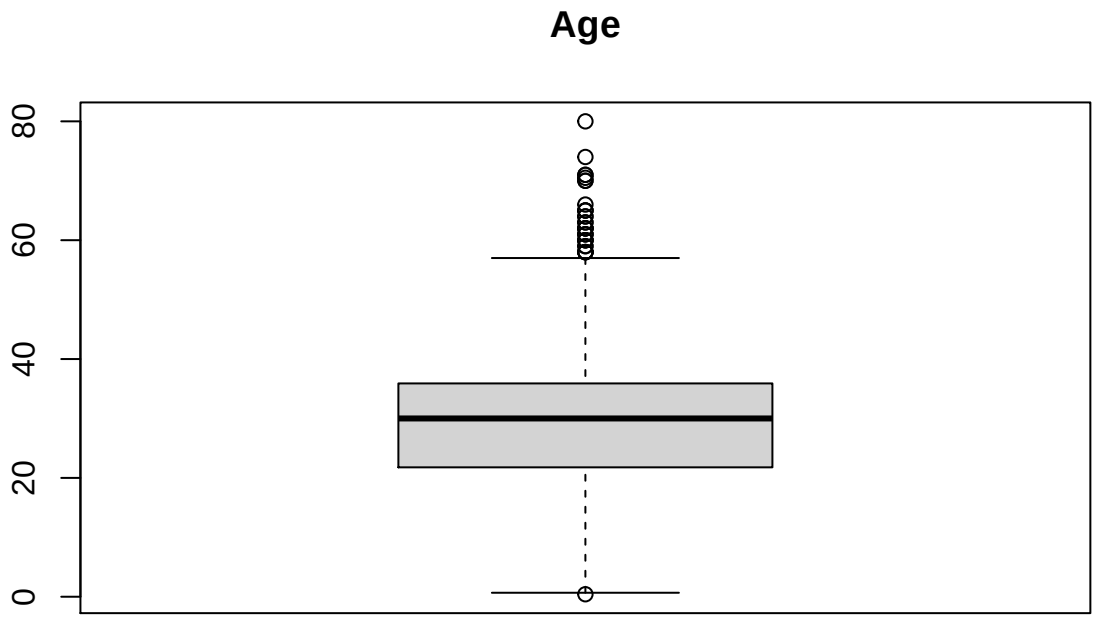
En los resultados obtenidos se comprueba que no hay ningún valor extremo en estas variables:

Survived: se comprueba que toma valores 0 o 1. **Pclass:** toma valores entre 1 y 3. **Sex:** se comprueba que hay dos niveles "female" y "male" **SibSp:** toma valores entre 0 - 8, siendo los más frecuentes 0 (608 pasajeros) y 1 (209 pasajeros) **Parch:** toma valores entre 0 - 6, siendo los más frecuentes 0 (678 pasajeros) y 1 (118 pasajeros)

Vamos ahora a verificar si en las variables "Age" y "Fare" hay valores extremos. Si es así, y se trata de un valor anormalmente alto o bajo, se sustituirá el valor por "NA", para realizar posteriormente una posible imputación. Para localizar estos valores extremos se va a emplear la representación boxplot que permite obtener los outliers (valores atípicos) de una determinada variable. Empleando "$out" obtenemos estos valores extremos que son posteriormente sustituidos por NA en las variables originales.

Age

```r
# Se representan el boxplot de la variable Age
b1<-boxplot(ds$Age, main="Age")
```




```
# Se obtienen las estadísticas
b1$stats
```

```
##          [,1]
## [1,]  0.67000
## [2,] 21.77397
## [3,] 30.00000
## [4,] 35.89815
## [5,] 57.00000
```

```
# Se contabilizan los outliers
length(b1$out)
```

```
## [1] 34
```

```
summary(ds[,"Age"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   21.77   30.00   29.76   35.90   80.00
```

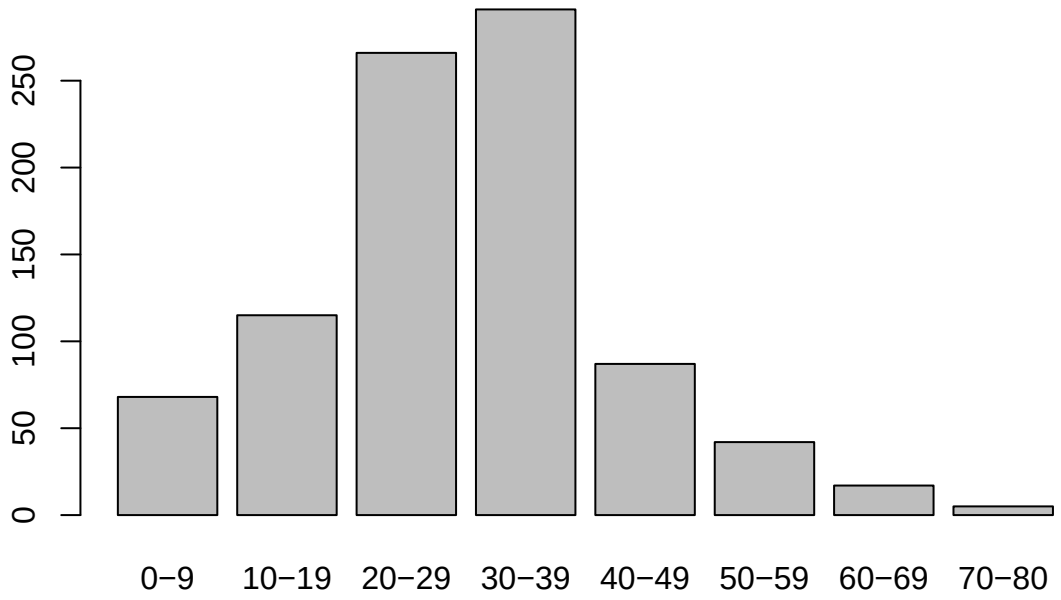
Para la variable Age se obtiene que el valor del bigote inferior es 0.67 y el del bigote superior 57.0. Los extremos de la caja son 21.77 y 35.89, inferior y superior respectivamente, siendo la mediana de la edad de los pasajeros 30.0 años. La interpretación de estos resultados indica que la mitad de los pasajeros a bordo del titanic tenía entre 21.77 y 35.89 años. Además se ha obtenido que hay 34 puntos considerados outliers por

situarse alejados del resto de datos, 1.5 veces menor o mayor que los extremos de los bigotes. Sin embargo si observamos el rango de la edad de los pasajeros observamos que los valores se encuentran acotados entre 0.42 y 80 años, que son edades razonables. De forma que no se van a eliminar estos outliers, porque forman parte de la diversidad de la muestra. Podrían aplicarse otros procedimientos como la eliminación de las filas con outliers o realizar imputación de estos valores como hicimos en el punto anterior. En este caso se va a realizar la discretización de la variable en grupos de edad, para reducir el ruido de la misma.

```
# Discretizamos
ds["segmento_edad"] <- cut(ds$Age, breaks = c(0,10,20,30,40,50,60,70,100), labels = c("0-9", "10-19", "
# Observamos los datos discretizados.
head(ds)
```

```
##   PassengerId                                                 Name Survived
## 1           1                              Braund, Mr. Owen Harris        0
## 2           2 Cumings, Mrs. John Bradley (Florence Briggs Thayer)        1
## 3           3                               Heikkinen, Miss. Laina        1
## 4           4       Futrelle, Mrs. Jacques Heath (Lily May Peel)        1
## 5           5                             Allen, Mr. William Henry        0
## 6           6                                     Moran, Mr. James        0
##   Pclass    Sex      Age SibSp Parch    Fare           Ticket Title
## 1      3   male 22.00000     1     0  7.2500        A/5 21171    Mr
## 2      1 female 38.00000     1     0 71.2833         PC 17599   Mrs
## 3      3 female 26.00000     0     0  7.9250 STON/O2. 3101282  Miss
## 4      1 female 35.00000     1     0 53.1000           113803   Mrs
## 5      3   male 35.00000     0     0  8.0500           373450    Mr
## 6      3   male 32.36809     0     0  8.4583           330877    Mr
##   segmento_edad
## 1         20-29
## 2         30-39
## 3         20-29
## 4         30-39
## 5         30-39
## 6         30-39
```

```
# Vemos como se agrupan los datos.
plot(ds["segmento_edad"])
```



Se ha discretizado la variable "Age" en ocho intervalos de 10 años cada uno. Esta segmentación será de utilidad en apartados posteriores para analizar las relaciones de los datos. En el gráfico de barras representado se observa que, como ya habíamos obtenido la mayoría de los pasajeros se concentran en los segmentos "20-29" y "30-39".

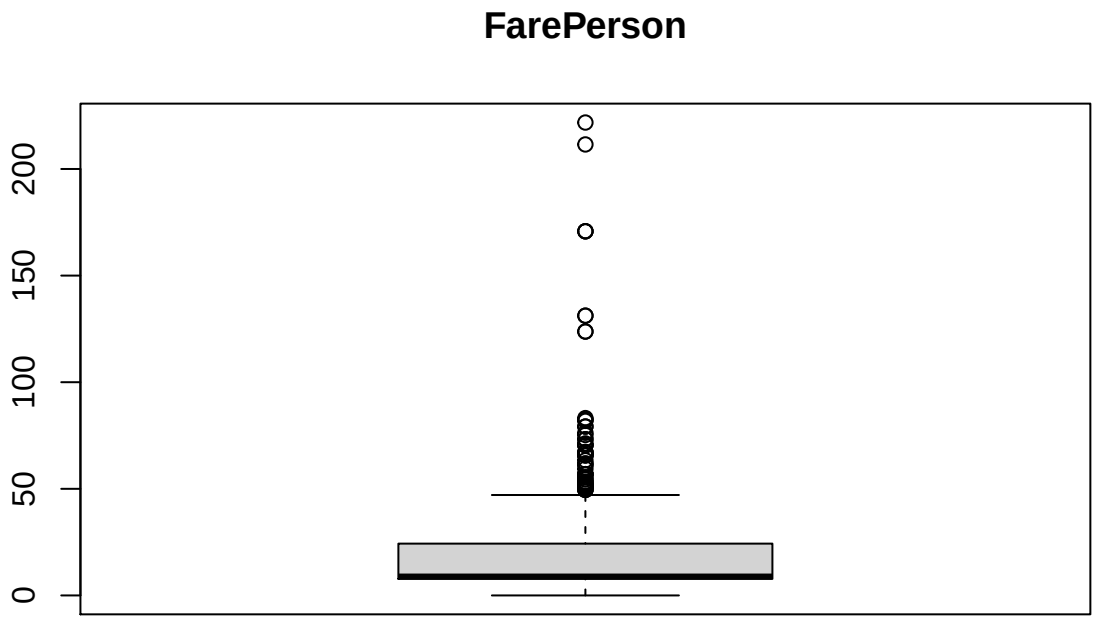
Fare

Se va a realizar el mismo análisis para la variable Fare para tratar de determinar si hay valores extremos en esta variable. En primer lugar, aunque el atributo Ticket es algo inútil en cuanto a extraer información del propio número de ticket, proporciona información sobre cuántos tickets se compraron con una tarifa determinada, de modo que podamos calcular la tarifa por persona, que es lo que necesitamos ya que nuestra unidad de observación (una fila) es una persona. Creamos un atributo FarePerson y lo comparamos con el atributo Fare:

```
# Cuenta de tickets
counts <- aggregate(ds$Ticket, by=list(ds$Ticket),
                    FUN=function(ticket) sum(!is.na(ticket)))
# Función para el cálculo de ratio de tickets
compute_fare_person <- function(ds) {
  fare <- as.numeric(ds["Fare"])
  # Cuenta
  count_ticket_i <- counts[which(counts[,1] == ds["Ticket"]), 2]
  result <- round(fare/count_ticket_i,2)
  return(result)
}
# Aplicamos la función para obtener FarePerson
ds$FarePerson <- apply(X=ds, MARGIN=1, FUN=compute_fare_person)
```

A continuación, se representará el diagrama de cajas y bigotes y finalmente se analizarán los quartiles obtenidos y los posibles outliers.

```
# Se representan los boxplot de la variable Fare
b2<-boxplot(ds$FarePerson, main="FarePerson")
```




```
b2$stats
```

```
##        [,1]
## [1,]  0.000
## [2,]  7.765
## [3,]  8.850
## [4,] 24.290
## [5,] 47.100
```

```
length(b2$out)
```

```
## [1] 59
```

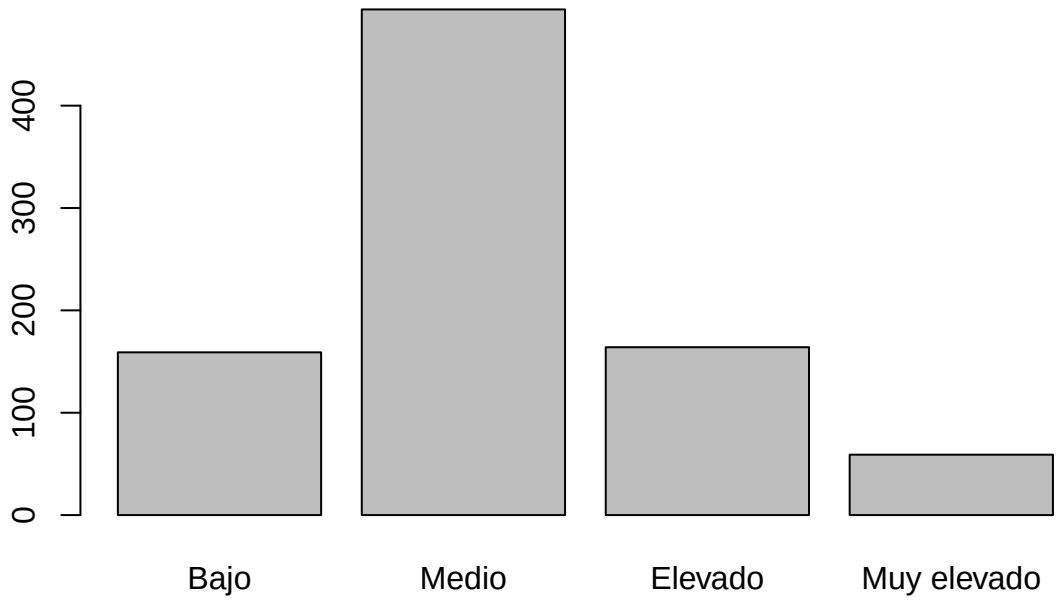
Para la variable Age se obtiene que el valor del bigote inferior es 0 y el del bigote superior 47.10 dólares. Los extremos de la caja son 7.765 y 24.29 dólares, inferior y superior respectivamente, siendo la mediana del precio del ticket por pasajero 8.85 dólares. La interpretación de estos resultados indica que la mitad de los pasajeros a bordo del titanic pagó entre 7.765 y 24.29 dólares por el billete. Además se ha obtenido que hay 59 puntos considerados outliers por situarse alejados del resto de datos, 1.5 veces mayor que el extremo de

los bigotes. Para tratar estos valores extremos podrían aplicarse procedimientos como la eliminación de las filas con outliers o realizar imputación de estos valores como hicimos en el punto anterior. En este caso se va a realizar la discretización de la variable en grupos de precio, para reducir el ruido de la misma.

```
# Discretizamos
ds["segmento_fare"] <- cut(ds$FarePerson, breaks = c(0,7.7,24.3,47.10,300), labels = c("Bajo", "Medio",

# Observamos los datos discretizados.
head(ds)
```

```
##   PassengerId                                                Name Survived
## 1           1                             Braund, Mr. Owen Harris        0
## 2           2 Cumings, Mrs. John Bradley (Florence Briggs Thayer)        1
## 3           3                              Heikkinen, Miss. Laina        1
## 4           4        Futrelle, Mrs. Jacques Heath (Lily May Peel)        1
## 5           5                            Allen, Mr. William Henry        0
## 6           6                                    Moran, Mr. James        0
##   Pclass    Sex      Age SibSp Parch    Fare           Ticket Title
## 1      3   male 22.00000     1     0  7.2500        A/5 21171    Mr
## 2      1 female 38.00000     1     0 71.2833         PC 17599   Mrs
## 3      3 female 26.00000     0     0  7.9250 STON/O2. 3101282  Miss
## 4      1 female 35.00000     1     0 53.1000           113803   Mrs
## 5      3   male 35.00000     0     0  8.0500           373450    Mr
## 6      3   male 32.36809     0     0  8.4583           330877    Mr
##   segmento_edad FarePerson segmento_fare
## 1         20-29       7.25          Bajo
## 2         30-39      71.28   Muy elevado
## 3         20-29       7.92         Medio
## 4         30-39      26.55       Elevado
## 5         30-39       8.05         Medio
## 6         30-39       8.46         Medio
```

```
# Vemos como se agrupan los datos.
plot(ds["segmento_fare"])
```



Se ha discretizado la variable "FarePerson" en cuatro intervalos (Bajo, Medio, Elevado y Muy elevado) en función de los rangos intercuartílicos obtenidos en el gráfico de caja y bigotes. . Esta segmentación será de utilidad en apartados posteriores para analizar las relaciones de los datos. En el gráfico de barras representado se observa la distribución del número de billetes de cada categoría.

4. Análisis de los datos.

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

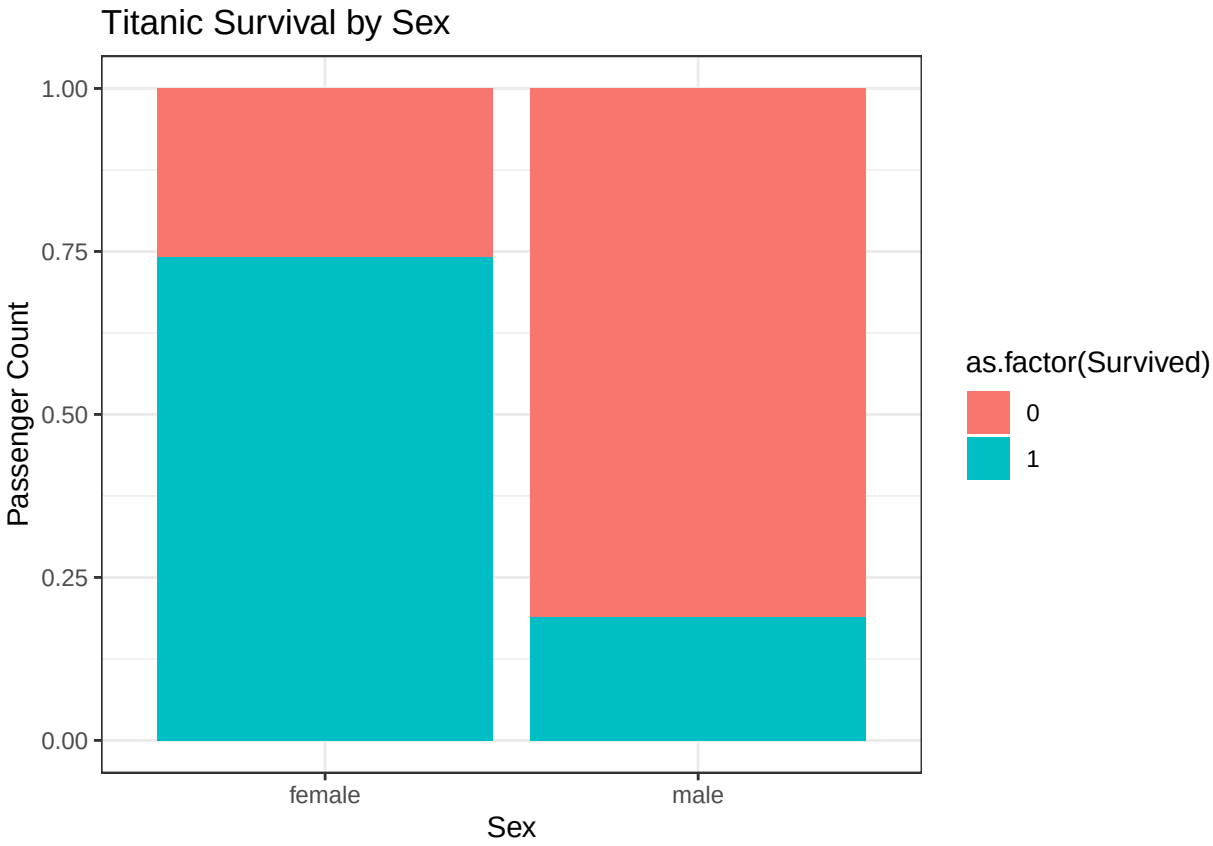
En este apartado se va a realizar un diseño del estudio estadístico que se quiere desarrollar. El objetivo último es definir si hay diferencias en la variable de clasificación "Survived", es decir en el resultado de supervivencia del pasajero, entre los diferentes grupos de pasajeros.

De las variables disponibles en el dataset se van a incluir en el estudio: La variable **"Sex"**, categórica, para determinar si hay diferencias entre hombres y mujeres en la supervivencia. En el conjunto de entrenamiento hay un total de 314 mujeres y 577 hombres.

```
table(ds$Sex)
```

```
##
## female   male
##    314    577
```

```
# Representación
ggplot(ds, aes(x = Sex, fill = as.factor(Survived))) + theme_bw() + geom_bar(position="fill") + labs(y =
```



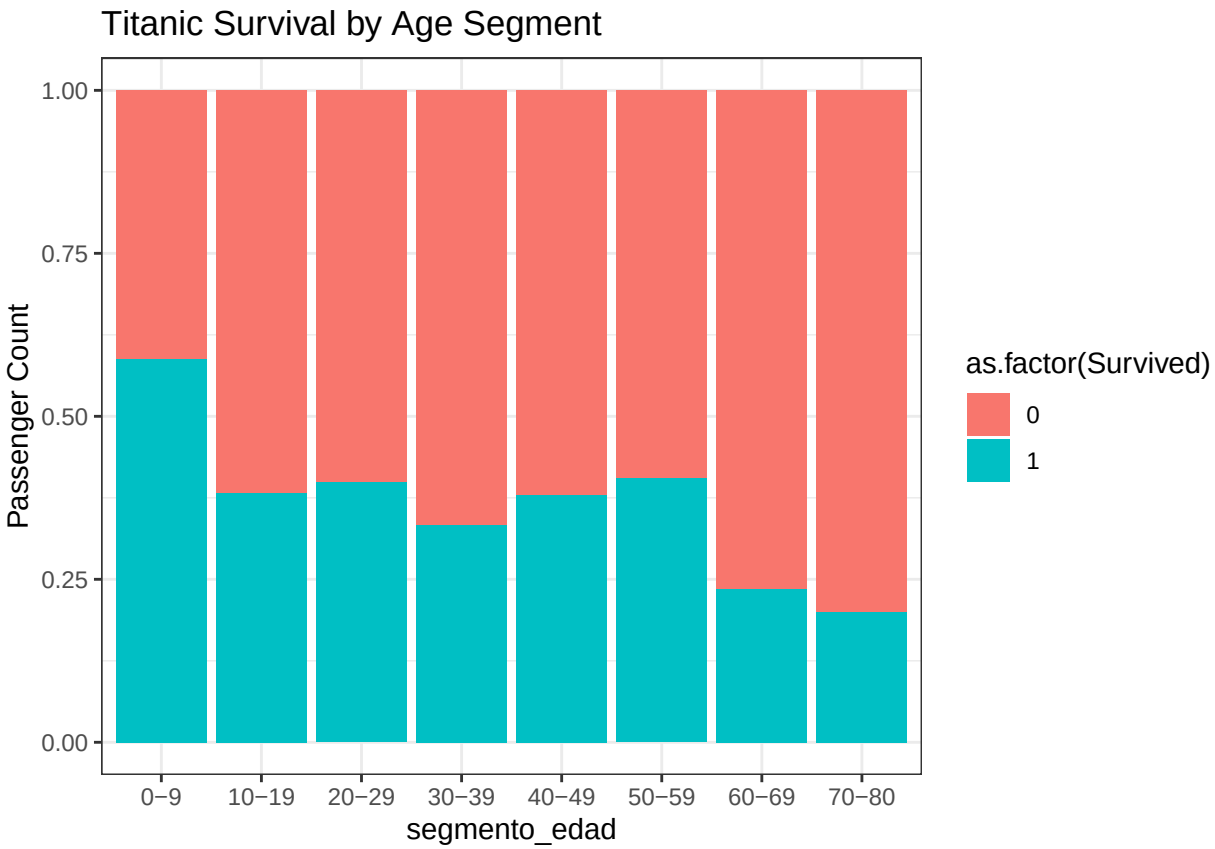
Además, en el gráfico representado se observa que de las mujeres que viajaban en el barco hay un mayor porcentaje (75%) de supervivientes que entre los hombres.

La variable **"Age_segment"**, para determinar si hay diferencias entre los distintos rangos de edad en la supervivencia. En la tabla inferior se muestra el número de pasajeros que hay en cada grupo de edad, se observa que los grupos más numerosos son los correspondientes a los intervales "20-29" y "30-39" años.

```
table(ds["segmento_edad"])
```

```
##
##   0-9 10-19 20-29 30-39 40-49 50-59 60-69 70-80
##    68   115   266   291    87    42    17     5
```

```
ggplot(ds, aes(x = segmento_edad, fill = as.factor(Survived))) + theme_bw() + geom_bar(position="fill")
```



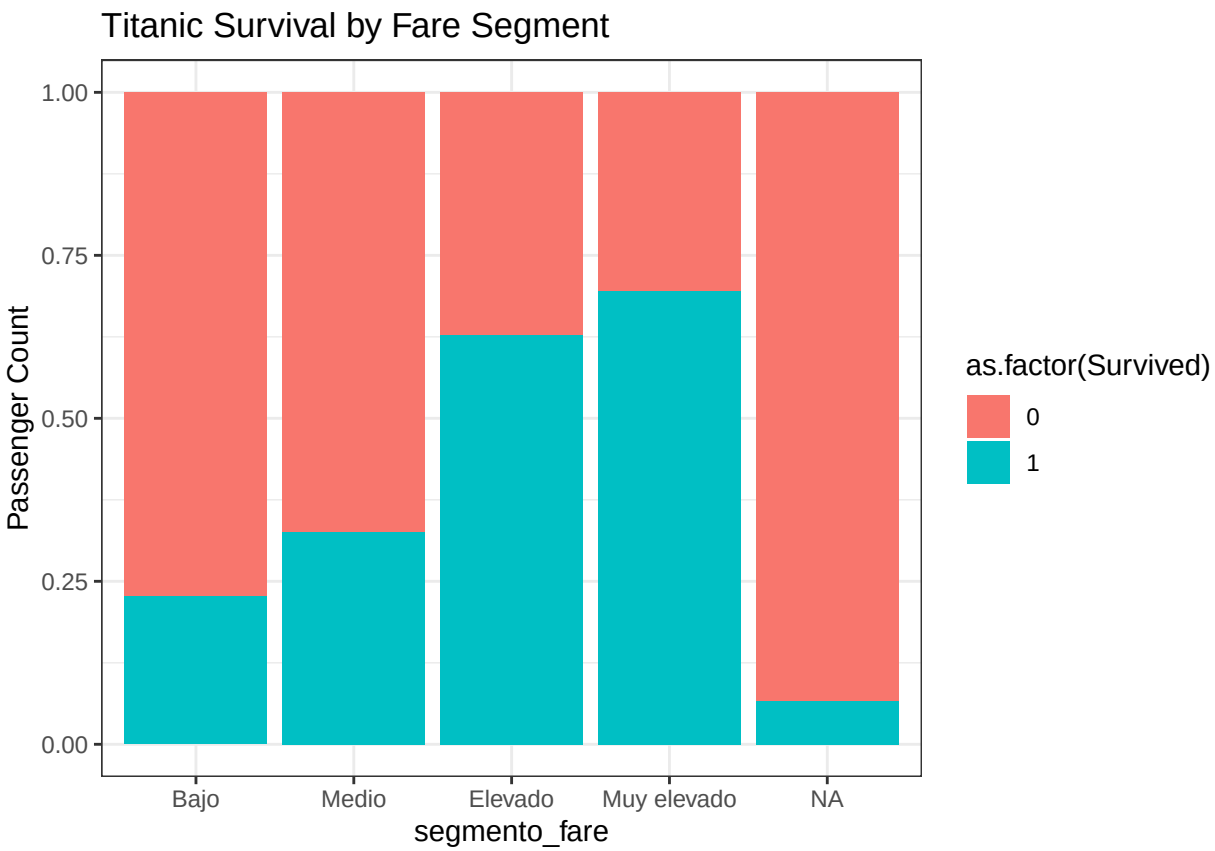
En el gráfico observamos que el segmento de edad con una mayor tasa de supervivencia es el de entre 0-9 años con aproximadamente un 60%. Los grupos con una menor tasa de supervivencia son los de los sujetos con una edad más avanzada.

La variable **FarePerson** para tratar de determinar si aquellos pasajeros que pagaron un ticket más caro tenían más posibilidades de supervivencia o a la inversa.La gran mayoría de los pasajeros compraron un billete con un precio medio.

```
table(ds["segmento_fare"])
```

```
##
##        Bajo       Medio     Elevado Muy elevado
##         159         494         164          59
```

```
ggplot(ds, aes(x = segmento_fare, fill = as.factor(Survived))) + theme_bw() + geom_bar(position="fill")
```



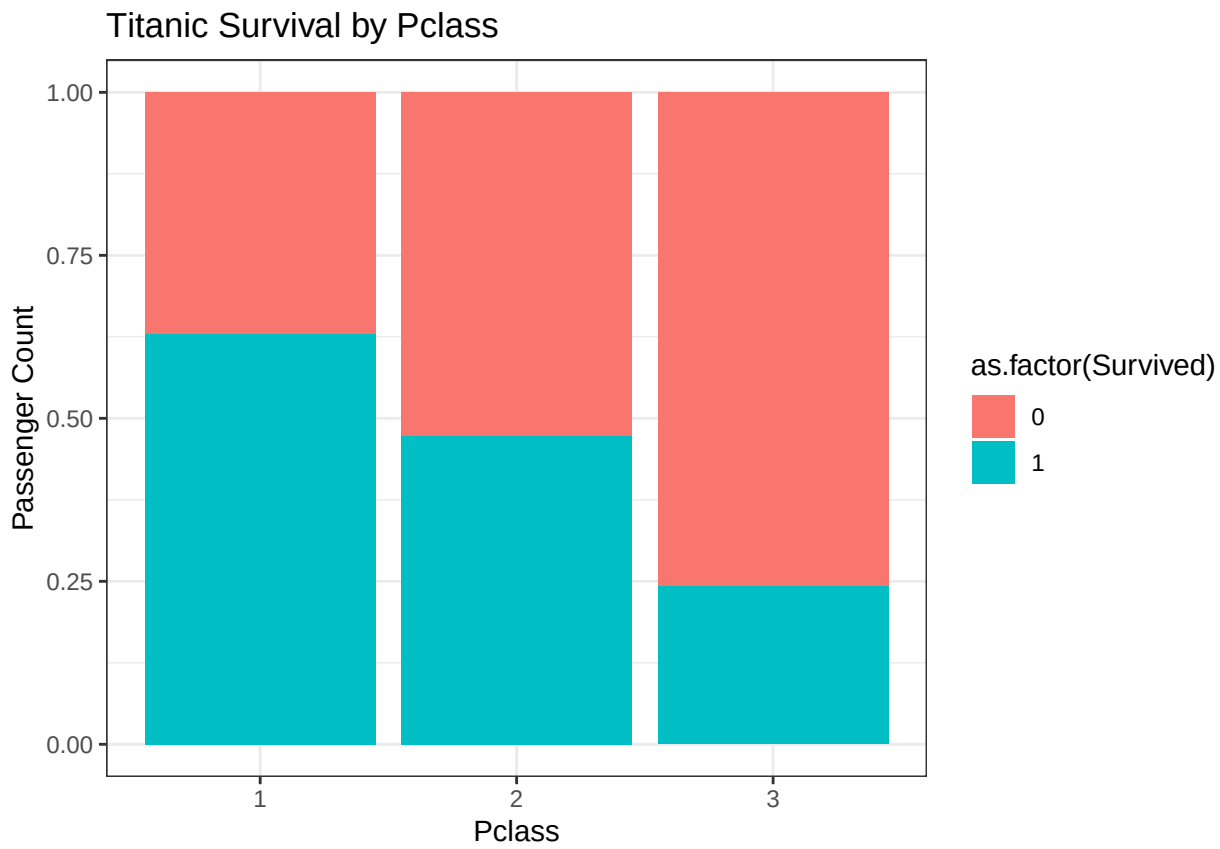
En este nuevo gráfico obtenemos que la tasa de supervivencia aumenta en relación al precio pagado por el billete. De esta forma se observa que aquellos pasajeros que pagaron un precio muy elevado por los billetes tenían más posibilidades de sobrevivir (aprox. 70%) frente a menos del 25% de aquellos pasajeros que pagaron un precio bajo.

La variable **Pclass** para determinar si hay diferentes en la superviviencia entre los pasajeros que viajaban en diferentes clases. Como se observa en la tabla inferior, el grupo más numeroso es el de pasajeros que viajaban en tercera clase.

```
table(ds$Pclass)
```

```
##
##   1   2   3
## 216 184 491
```

```
# Representación
ggplot(ds, aes(x = Pclass, fill = as.factor(Survived))) + theme_bw() + geom_bar(position="fill") + labs
```



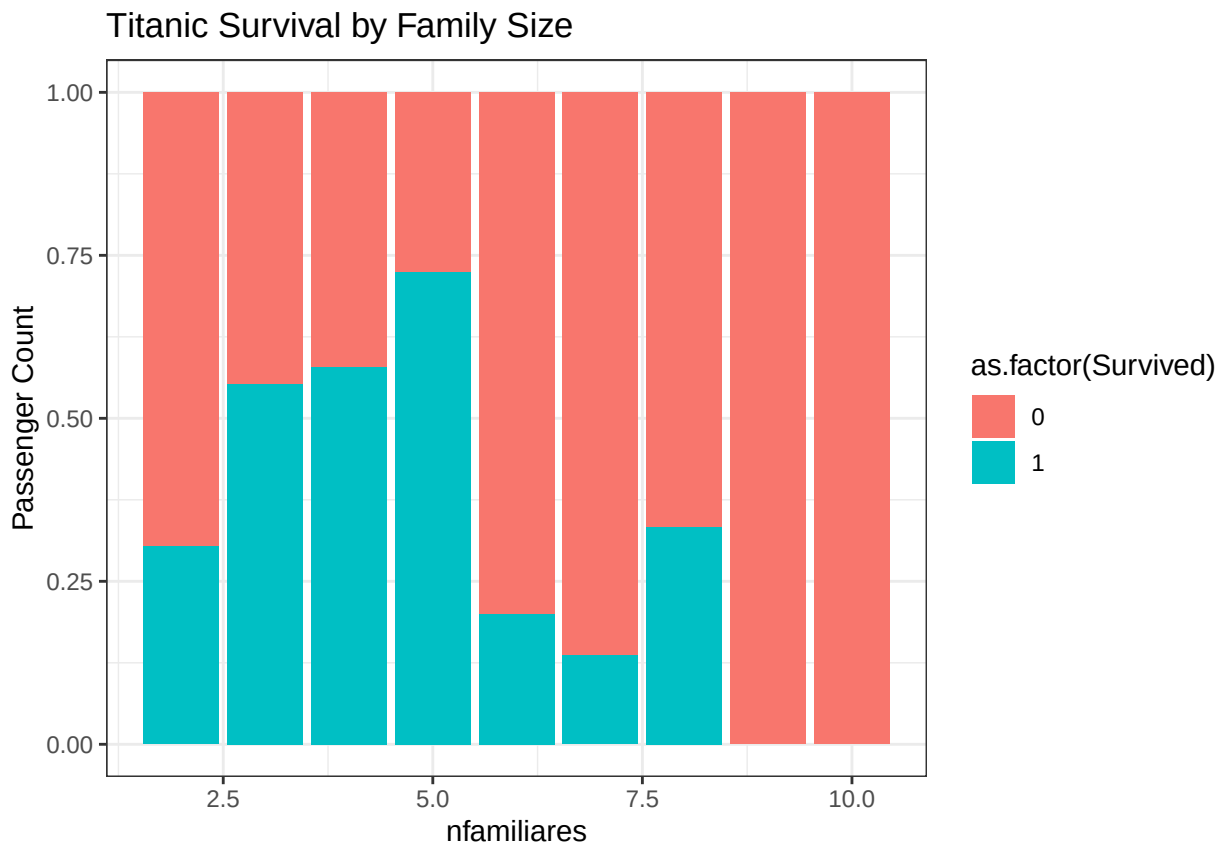
Del gráfico se obtiene que hay un mayor porcentaje de supervivencia entre los pasajeros de primera clase (63%), así que parece que esta variable también tiene influencia en el resultado.

Finalmente se trabajará con la variable **nfamilares**, que representará el número de familiares que tenía cada pasajero a bordo del Titanic. Esta variable se obtendrá como suma de la variable "SibSp" y "Parch", de forma que no se diferenciará por el tipo de familiar si no por el número total de familiares. De esta forma se tratará de determinar si hay una mayor supervivencia en los pasajeros que tenían más (o menos) familiares a bordo.Como se observa en la tabla inferior, la mayoría de los pasajeros tenían 2 familiares a bordo, seguido de 3 y 4.

```
ds$nfamiliares=as.numeric(ds$SibSp)+as.numeric(ds$Parch)
table(ds$nfamiliares)
```

```
##
##   2   3   4   5   6   7   8   9  10
## 537 161 102  29  15  22  12   6   7
```

```
# Representación
ggplot(ds, aes(x = nfamiliares, fill = as.factor(Survived))) + theme_bw() + geom_bar(position="fill") +
```



En el gráfico superior se observa que la tasa de supervivencia aumenta con el número de familiares que tenía cada pasajero hasta llegar a 5 donde alcanza su máximo (70%). A partir de 5 familiares decrece esta tasa.

De esta forma, se concluye que las variables que se van a analizar en los siguientes apartados para estudiar su correlación con la supervivencia de los pasajeros son: "Sex", "Age_segment", "Fare_person","Pclass" y "nfamiliares".

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

En primer lugar, se van a agrupar las variables que vamos a analizar en un nuevo dataset, en este caso en las variables continuas de las que disponemos en el dataset: Age y FarePerson. Vamos además a representar estas dos variables para analizar preliminarmente la distribución de estas variables a partir de un histograma.

```
library(mltools)
```

```
## Warning: package 'mltools' was built under R version 4.0.5
```

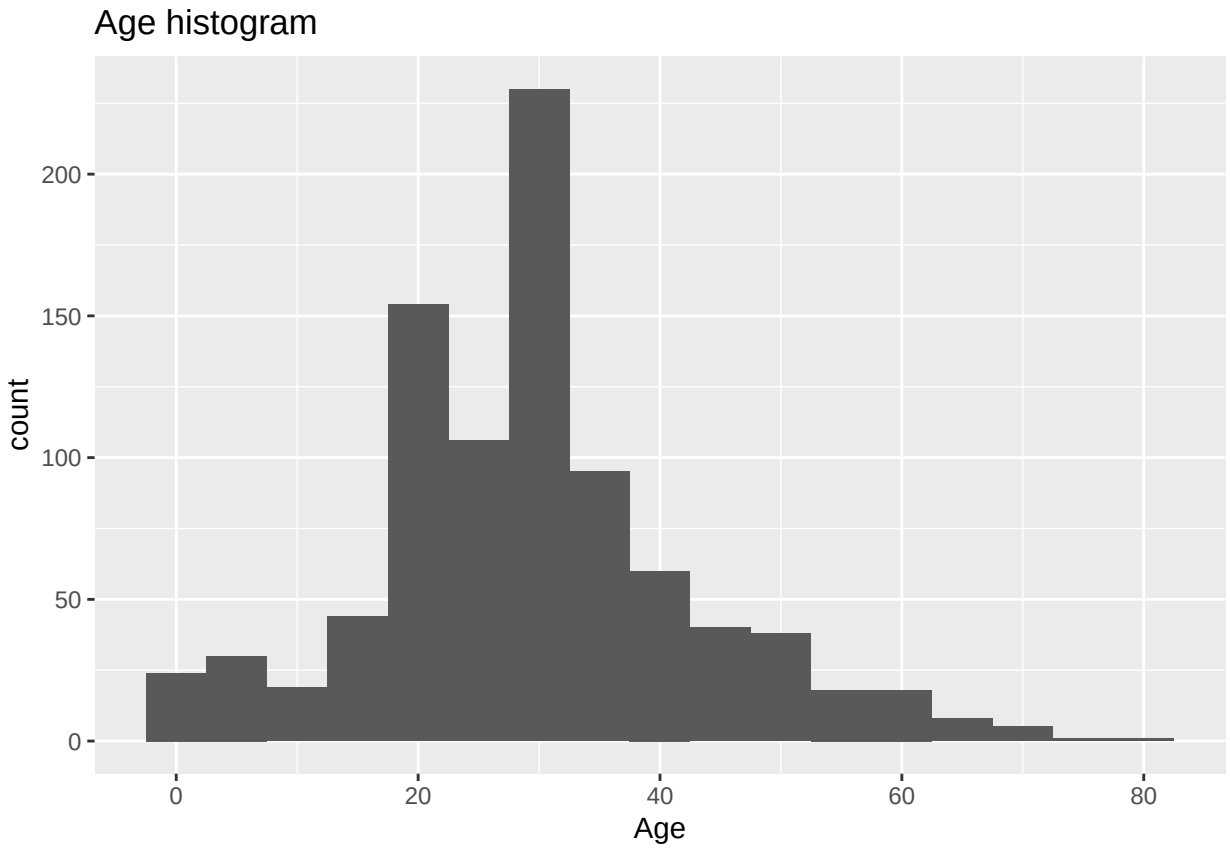
```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.3
```

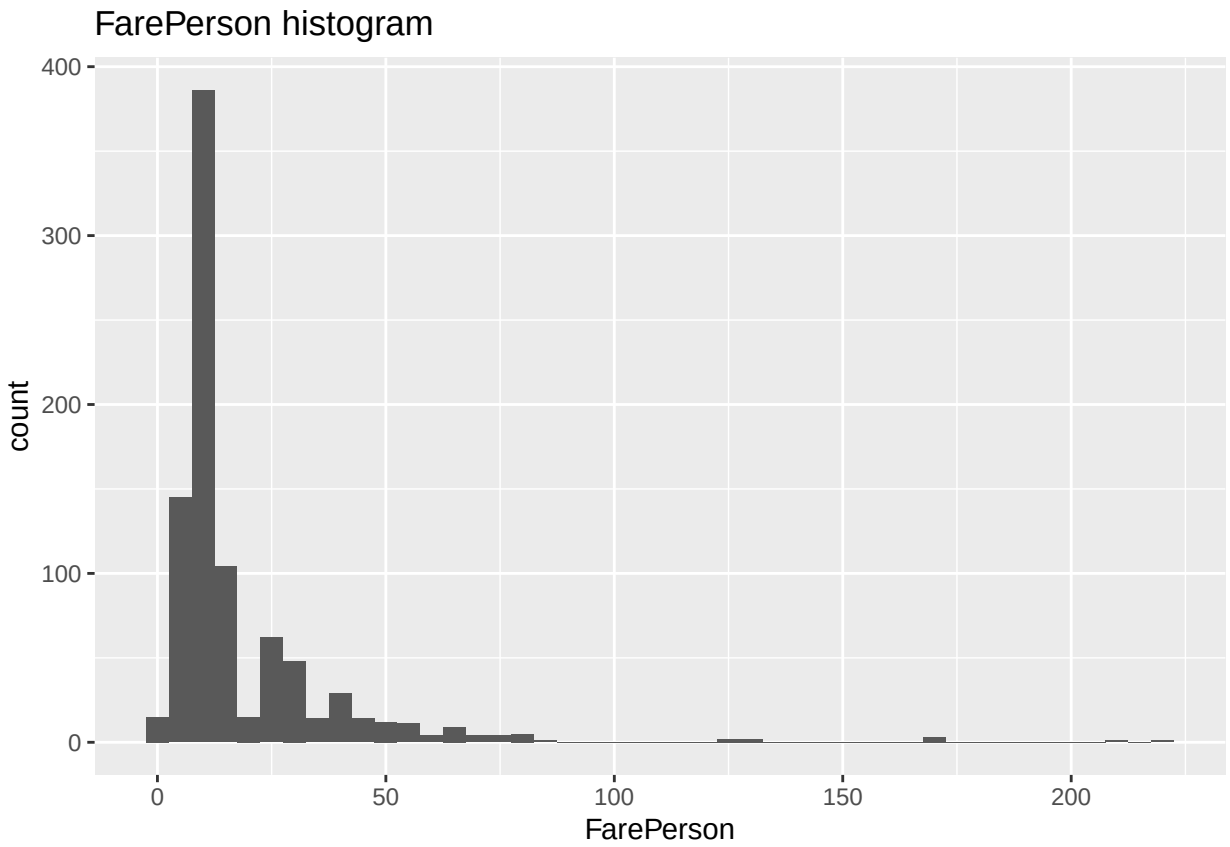
```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
incluir <- c("Age","FarePerson")
newds <- ds[ , (names(ds) %in% incluir)]
ggplot(data=ds, aes(Age)) + geom_histogram(binwidth=5) + labs(title = "Age histogram")
```




```
ggplot(data=ds, aes(FarePerson)) + geom_histogram(binwidth=5) + labs(title = "FarePerson histogram")
```



En los gráficos obtenidos observamos que la variable Age sí que se aproxima más a una distribución normal, aunque no sucede lo mismo con la variable FarePerson. Vamos a realizar ahora las pruebas de normalidad aplicando el test Anderson-Darling. En este test la hipótesis nula H0 es que la muestra proviene de una distribución normal. Aplicaremos el test a las dos variables que hemos analizado previamente.

```
library(nortest)
```

```
## Warning: package 'nortest' was built under R version 4.0.3
```

```
ad.test(newds$Age)
```

```
##
##  Anderson-Darling normality test
##
## data:  newds$Age
## A = 7.7909, p-value < 2.2e-16
```

```
ad.test(newds$FarePerson)
```

```
##
##  Anderson-Darling normality test
##
## data:  newds$FarePerson
## A = 110.32, p-value < 2.2e-16
```

Tanto para la variable Age como para la variable FarePerson se obtiene un pvalor menor que 2.2e-16. De esta forma ambos pvalores son menores que el nivel de significancia alfa (alfa=0.05). De esta forma no se puede rechazar la hipótesis nula para niguna de las variables y no podemos asumir que ninguna de ellas siga una distribución normal.

Vamos ahora a verificar si para estas variables podemos asumir la homogeneidad de la varianza para las clases que vamos a intentar predecir con el modelo (Survived). Para ello, se va a emplear la prueba de Fligner-Killeen que es una de las muchas pruebas de homogeneidad de varianzas y que es robusta frente a desviaciones de la normalidad. La hipótesis nula para este test es que la varianza de las poblaciones son iguales.

```
fligner.test(Age ~ Survived, data = ds)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Survived
## Fligner-Killeen:med chi-squared = 7.0008, df = 1, p-value = 0.008147
```

```
fligner.test(FarePerson ~ Survived, data = ds)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  FarePerson by Survived
## Fligner-Killeen:med chi-squared = 126.23, df = 1, p-value < 2.2e-16
```

Puesto que obtenemos un p-valor inferior a 0,05, rechazamos la hipótesis de que las varianzas de ambas muestras son homogéneas.

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

Como el objetivo del estudio es predecir qué pasajeros sobrevivieron y cuáles no, el análisis de cada variable se hace en relación a la variable respuesta Survived. Analizando los datos de esta forma, se pueden empezar a extraer ideas sobre qué variables están más relacionadas con la supervivencia.

La distribución de la edad de los pasajeros parece ser muy similar entre el grupo de supervivientes y fallecidos, con dos excepciones: en el rango de edad aproximado de 0 a 10 años, el porcentaje de supervivencia es mucho mayor, mientras que, en el extremo opuesto, a partir de los 60 años, la tendencia se invierte. Dos hipótesis que podrían explicar estos patrones son: que, según los registros, en el protocolo de evacuación del Titanic se priorizó que mujeres y niños subiesen a los botes salvavidas, y que los ancianos tuviesen menor movilidad para alcanzar las zonas de evacuación.

```
# Estadísticos de la edad de los supervivientes y fallecidos
ds %>% filter(!is.na(Age)) %>% group_by(Survived) %>%
  summarise(media = mean(Age),
            mediana = median(Age),
            min = min(Age),
            max = max(Age))
```

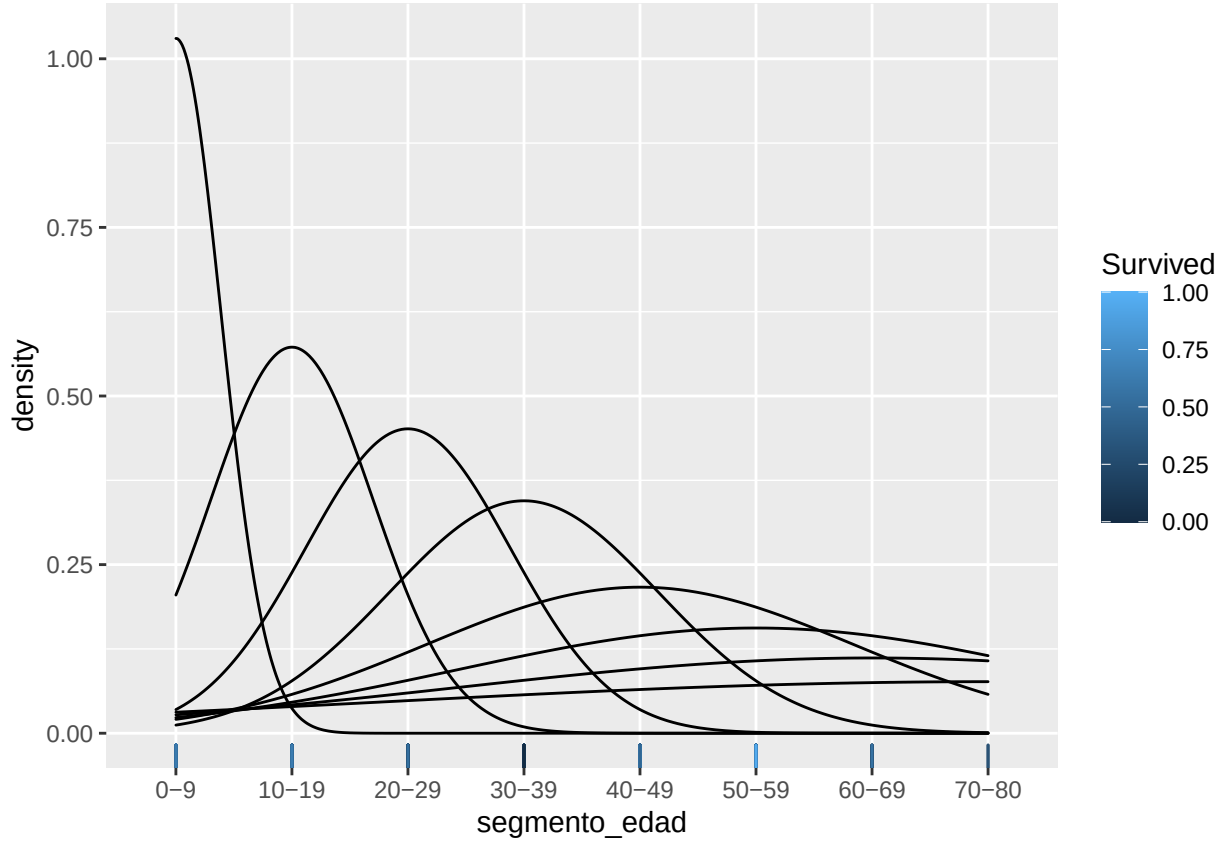
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 5
##   Survived media mediana   min   max
##      <int> <dbl>   <dbl> <dbl> <dbl>
## 1        0  30.7      32  1       74
## 2        1  28.3      28  0.42    80
```

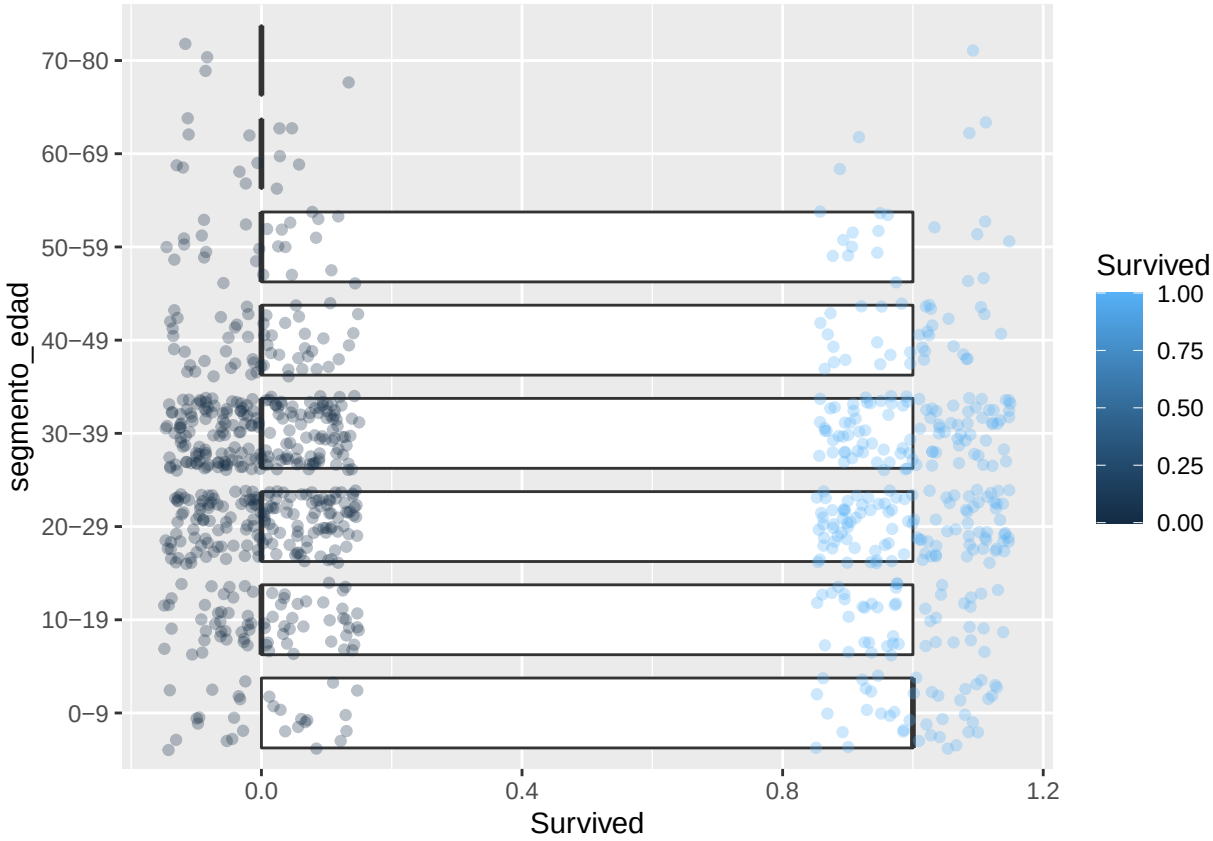
```
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.0.3
```

```
ggplot(data = ds, aes(x = segmento_edad, fill = Survived)) +
  geom_density(alpha = 0.5) +
  geom_rug(aes(color = Survived), alpha = 0.5)
```




```
ggplot(data = ds, aes(x = Survived, y = segmento_edad, color = Survived)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(alpha = 0.3, width = 0.15)
```



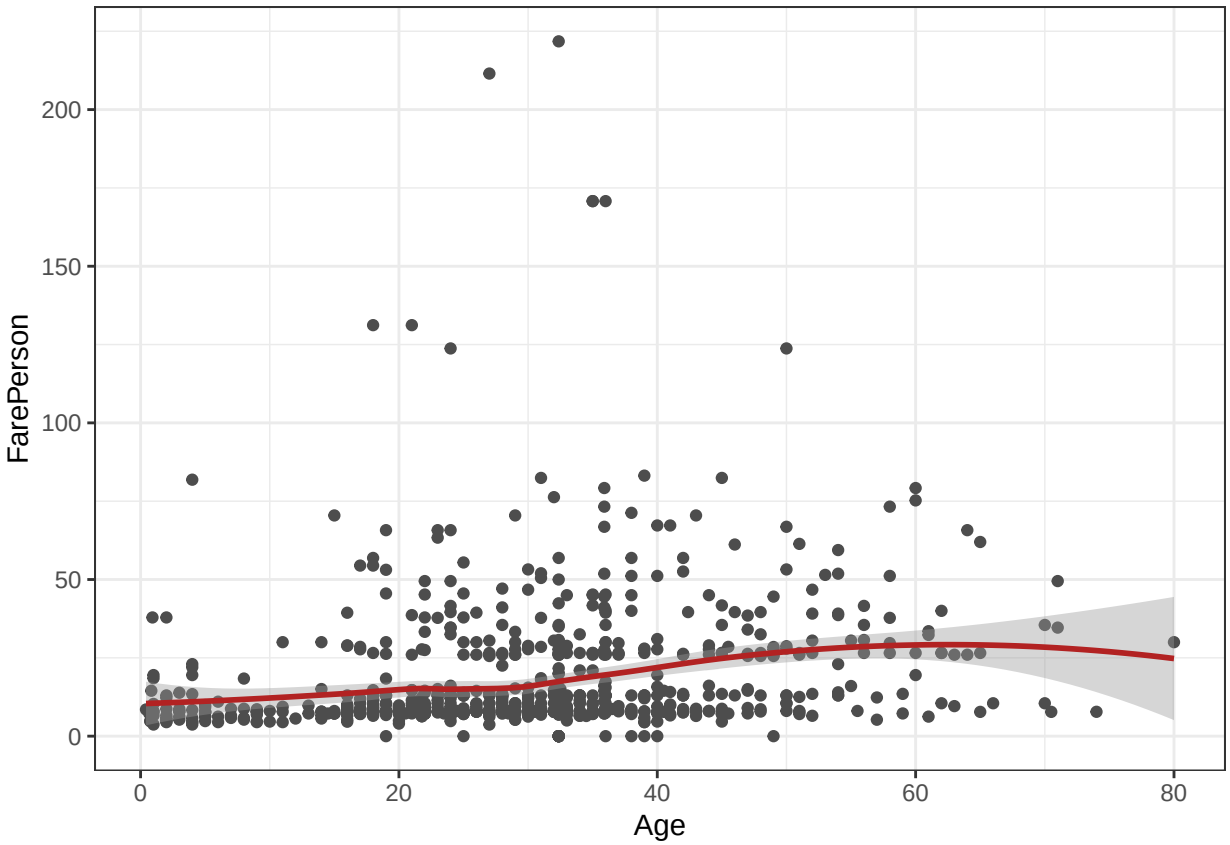
Podemos analizar la correlación existente entre variables contínuas.

```
cor.test(x = ds$Age, y = ds$FarePerson, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  ds$Age and ds$FarePerson
## t = 6.578, df = 889, p-value = 8.131e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1519094 0.2771927
## sample estimates:
##       cor
## 0.2154374
```

```
ggplot(data = ds, aes(x = Age, y = FarePerson)) +
  geom_point(color = "gray30") +
  geom_smooth(color = "firebrick") +
  theme_bw()
```

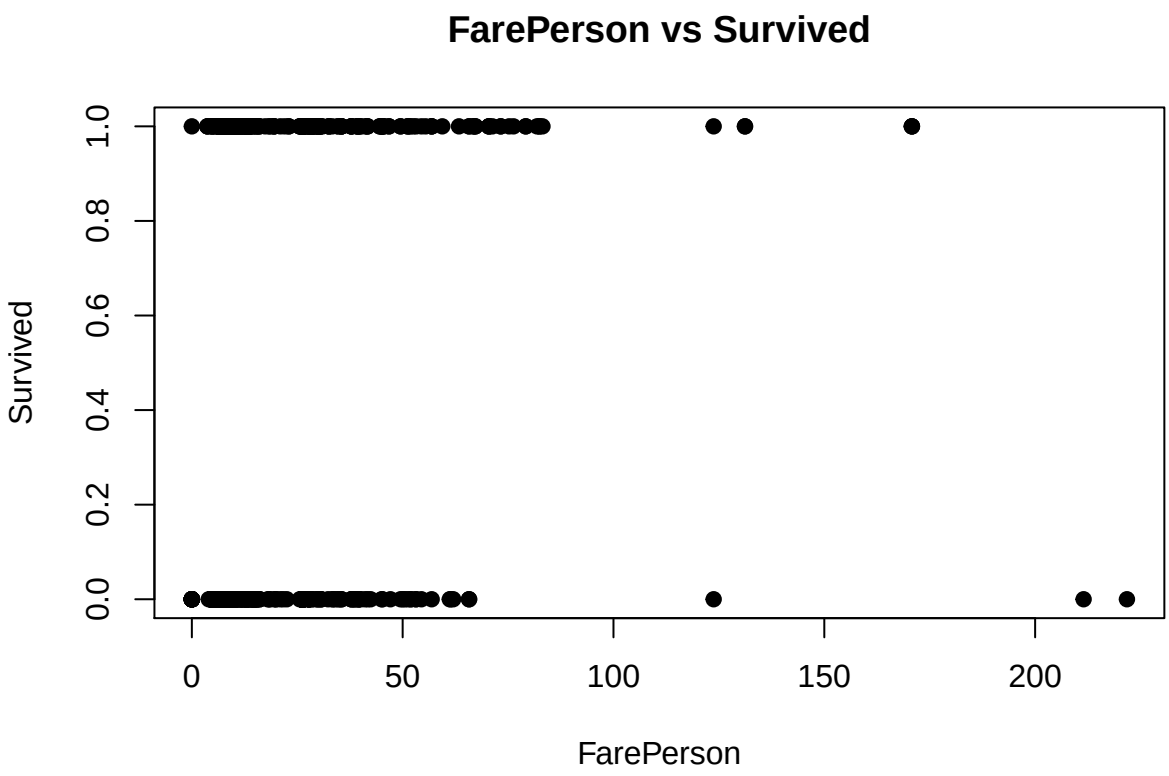
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Como vemos, esta correlación no es significativa y por lo tanto podemos asumir que las variables no contienen información redundante.

Una de las preguntas que nos hemos planteado a lo largo de la práctica es si el coste del ticket tenía relación con la supervivencia del pasajero. Para ello vamos a realizar un scatterplot para dar respuesta a esta pregunta.

```
plot(ds$FarePerson, ds$Survived, main="FarePerson vs Survived",
     xlab="FarePerson", ylab="Survived", pch=19)
```



Como podemos observar, no existe relación entre el coste y la supervivencia, ya que existen probabilidades similares de ambos resultados.

Regresión logística

Veamos ahora que tal se adapta un modelo de regresión logística a estos datos.

```
incluir <- c("Survived","Sex","segmento_edad","FarePerson","Pclass","nfamiliares")
newds <- ds[ , (names(ds) %in% incluir)]

ds_test$Survived <- sample(0:1, 418, replace=TRUE)

ds_test$Pclass<-factor(ds_test$Pclass)
ds_test$Sex<-factor(ds_test$Sex)
ds_test$Parch<-factor(ds_test$Parch)
ds_test$SibSp<-factor(ds_test$SibSp)
ds_test["segmento_edad"] <- cut(ds_test$Age, breaks = c(0,10,20,30,40,50,60,70,100), labels = c("0-9", "

# Cuenta de tickets
counts <- aggregate(ds_test$Ticket, by=list(ds_test$Ticket),
                    FUN=function(ticket) sum(!is.na(ticket)))
# Aplicamos la función para obtener FarePerson
ds_test$FarePerson <- apply(X=ds_test, MARGIN=1, FUN=compute_fare_person)


ds_test$nfamiliares=as.numeric(ds_test$SibSp)+as.numeric(ds_test$Parch)
newds_test <- ds_test[ , (names(ds_test) %in% incluir)]
```

```r
train <- newds
test <- newds_test

## Model Creation
model_glm <- glm(Survived ~.,family=binomial(link='logit'),data=train)

## Model Summary
summary(model_glm)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9030  -0.5780  -0.4455   0.5617   2.4501
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          5.789029   0.666050   8.692  < 2e-16 ***
## Pclass2             -1.039473   0.306472  -3.392 0.000695 ***
## Pclass3             -2.132105   0.307408  -6.936 4.04e-12 ***
## Sexmale             -2.901882   0.208174 -13.940  < 2e-16 ***
## segmento_edad10-19 -2.118152   0.466749  -4.538 5.68e-06 ***
## segmento_edad20-29 -2.323197   0.425882  -5.455 4.90e-08 ***
## segmento_edad30-39 -2.339296   0.430120  -5.439 5.37e-08 ***
## segmento_edad40-49 -2.769420   0.491423  -5.636 1.75e-08 ***
## segmento_edad50-59 -3.041724   0.587878  -5.174 2.29e-07 ***
## segmento_edad60-69 -3.574478   0.805100  -4.440 9.00e-06 ***
## segmento_edad70-80 -3.190833   1.263957  -2.524 0.011587 *
## FarePerson           0.005027   0.005350   0.940 0.347398
## nfamiliares         -0.356140   0.078722  -4.524 6.07e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  772.73  on 878  degrees of freedom
## AIC: 798.73
##
## Number of Fisher Scoring iterations: 5
```

De los resultados del modelo observamos que todas las variables explicativas que hemos empleado son significativas a excepción de FarePerson y el segmento de edad entre 70-80 años. Por ello, vamos a probar a excluir esta variable y analizar el AIC del nuevo modelo para verificar si mejora el ajuste.

```r
model_glm2 <- glm(Survived ~Pclass+Sex+segmento_edad+nfamiliares,family=binomial(link='logit'),data=tra

## Model Summary
summary(model_glm2)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + Sex + segmento_edad + nfamiliares,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9110  -0.5819  -0.4439   0.5634   2.4554
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         5.96319    0.63835   9.342  < 2e-16 ***
## Pclass2            -1.18790    0.26387  -4.502 6.74e-06 ***
## Pclass3            -2.31034    0.24412  -9.464  < 2e-16 ***
## Sexmale            -2.90577    0.20793 -13.975  < 2e-16 ***
## segmento_edad10-19 -2.09453    0.46460  -4.508 6.54e-06 ***
## segmento_edad20-29 -2.29186    0.42298  -5.418 6.01e-08 ***
## segmento_edad30-39 -2.31869    0.42823  -5.415 6.14e-08 ***
## segmento_edad40-49 -2.75863    0.49002  -5.630 1.81e-08 ***
## segmento_edad50-59 -3.02652    0.58629  -5.162 2.44e-07 ***
## segmento_edad60-69 -3.58418    0.80608  -4.446 8.73e-06 ***
## segmento_edad70-80 -3.18251    1.26390  -2.518   0.0118 *
## nfamiliares        -0.34818    0.07767  -4.483 7.37e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  773.65  on 879  degrees of freedom
## AIC: 797.65
##
## Number of Fisher Scoring iterations: 5
```

Observamos que con este nuevo modelo el AIC se ha reducido levemente por lo que hemos mejorado el ajuste del mismo y ahora todas las variables son significativas para el modelo.

```
## Using anova() to analyze the table of devaiance
anova(model_glm2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           890    1186.66
## Pclass        2  103.547       888    1083.11 < 2.2e-16 ***
## Sex           1  256.220       887     826.89 < 2.2e-16 ***
```

```
## segmento_edad  7   29.249       880     797.64 0.0001303 ***
## nfamiliares    1   23.989       879     773.65  9.69e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Predicting Test Data
result <- predict(model_glm2,newdata=test,type='response')
result <- ifelse(result > 0.5,1,0)

result<-factor(result)
test$Survived<-factor(test$Survived)


## Confusion matrix and statistics
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

```
## Loading required package: lattice
```

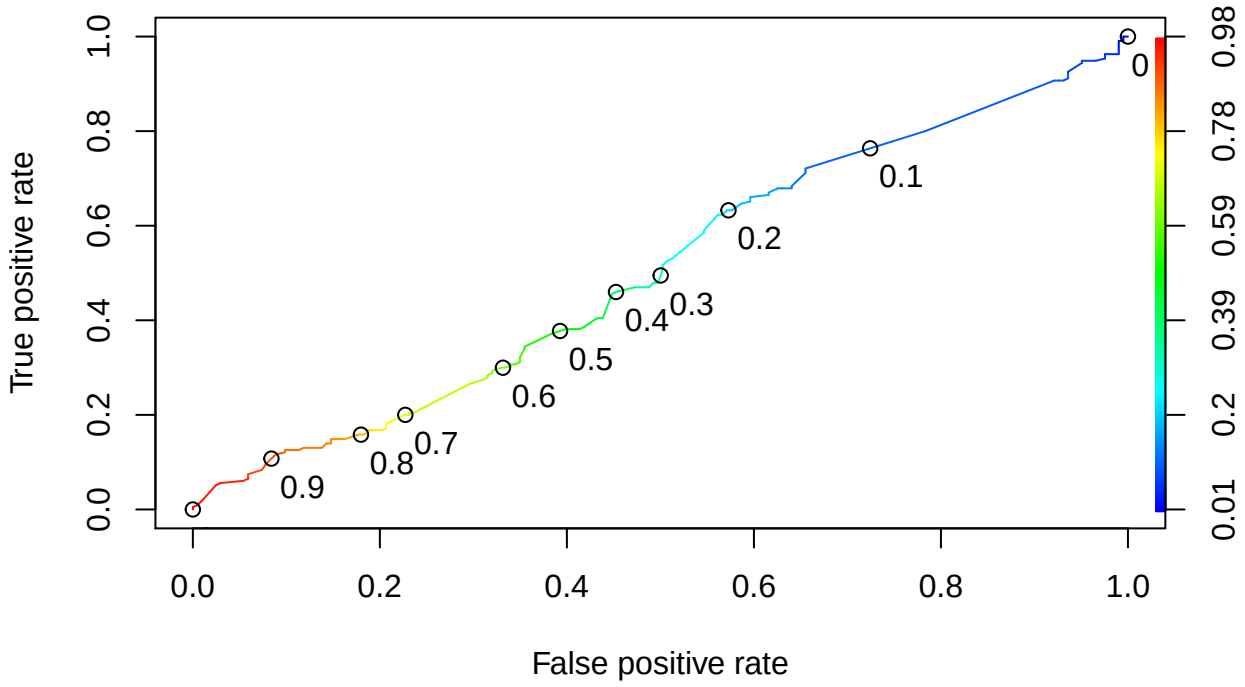
```
confusionMatrix(data=result, reference=test$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 125 135
##          1  78  80
##
##                Accuracy : 0.4904
##                  95% CI : (0.4415, 0.5395)
##     No Information Rate : 0.5144
##     P-Value [Acc > NIR] : 0.8479198
##
##                   Kappa : -0.012
##
##  Mcnemar's Test P-Value : 0.0001245
##
##             Sensitivity : 0.6158
##             Specificity : 0.3721
##          Pos Pred Value : 0.4808
##          Neg Pred Value : 0.5063
##              Prevalence : 0.4856
##          Detection Rate : 0.2990
##    Detection Prevalence : 0.6220
##       Balanced Accuracy : 0.4939
##
##        'Positive' Class : 0
##
```

```
## ROC Curve and calculating the area under the curve(AUC)
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.0.3
```

```
predictions <- predict(model_glm2, newdata=test, type="response")
ROCRpred <- prediction(predictions, test$Survived)
ROCRperf <- performance(ROCRpred, measure = "tpr", x.measure = "fpr")

plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7), print.cutoffs.at = seq(0,1,0.1))
```



Como podemos observar, si usamos una regresión logística obtenemos un accuracy del 82% y observamos una buena evolución de la curva ROC.

Ahora vamos a comparar diferentes modelos contra la regresión logística que ya tenemos hecha utilizando cross validation. Para ello utilizaremos algoritmos lineales y no lineales:

- Linear: Logistic Regression (LG) and Regularized Logistic Regression (GLMNET).
- Non-Linear: k-Nearest Neighbors (KNN), Classification and Regression Trees (CART), and Support Vector Machines with Radial Basis Functions (SVM).

```
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)

# LG
set.seed(7)
```

```
fit.glm <- train(as.factor(Survived)~., data=train, method="glm", trControl=trainControl)
# GLMNET
set.seed(7)
fit.glmnet <- train(as.factor(Survived)~., data=train, method="glmnet", trControl=trainControl)
# KNN
set.seed(7)
fit.knn <- train(as.factor(Survived)~., data=train, method="knn",trControl=trainControl)
# CART
set.seed(7)
fit.cart <- train(as.factor(Survived)~., data=train, method="rpart", trControl=trainControl)
# SVM
set.seed(7)
fit.svm <- train(as.factor(Survived)~., data=train, method="svmRadial", trControl=trainControl)

# Compare algorithms
results <- resamples(list(LG=fit.glm, GLMNET=fit.glmnet, KNN=fit.knn,
                          CART=fit.cart, SVM=fit.svm))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: LG, GLMNET, KNN, CART, SVM
## Number of resamples: 30
##
## Accuracy
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LG     0.7191011 0.7752809 0.7988764 0.7986753 0.8202247 0.8764045    0
## GLMNET 0.7303371 0.7799625 0.8089888 0.8005523 0.8202247 0.8764045    0
## KNN    0.6404494 0.7303371 0.7555556 0.7553247 0.7859104 0.8333333    0
## CART   0.6966292 0.7865169 0.7977528 0.7983175 0.8202247 0.8777778    0
## SVM    0.7528090 0.7977528 0.8212235 0.8230249 0.8440075 0.9000000    0
##
## Kappa
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LG     0.3807403 0.5096917 0.5724668 0.5655923 0.6138211 0.7396969    0
## GLMNET 0.4020157 0.5143709 0.5884696 0.5692626 0.6138211 0.7396969    0
## KNN    0.2298540 0.4123215 0.4827681 0.4783126 0.5483313 0.6437995    0
## CART   0.3151895 0.5195213 0.5677344 0.5559146 0.5999075 0.7415144    0
## SVM    0.4453258 0.5618162 0.6137363 0.6150933 0.6715713 0.7885117    0
```

Como podemos observar, SVM es el que nos da los mejores resultados con un Accuracy máximo del 92.5%, por lo que seguiremos trabajando con el en los próximos pasos, pero antes podemos probar de hacer un tuning y ver los resultados.

```
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
set.seed(7)
grid <- expand.grid(.sigma=c(0.025, 0.05, 0.1, 0.15), .C=seq(1, 10, by=1))
fit.svm <- train(as.factor(Survived)~., data=train, method="svmRadial",  tuneGrid=grid,
                 preProc=c("BoxCox"), trControl=trainControl)
print(fit.svm)
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 891 samples
##   5 predictor
##   2 classes: '0', '1'
##
## Pre-processing: Box-Cox transformation (1)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 802, 802, 802, 802, 801, 802, ...
## Resampling results across tuning parameters:
##
##   sigma  C   Accuracy   Kappa
##   0.025   1  0.7908470  0.5474120
##   0.025   2  0.8061912  0.5766845
##   0.025   3  0.8050341  0.5711822
##   0.025   4  0.8042892  0.5691609
##   0.025   5  0.8039188  0.5679448
##   0.025   6  0.8050299  0.5698942
##   0.025   7  0.8054044  0.5705379
##   0.025   8  0.8050340  0.5692477
##   0.025   9  0.8053961  0.5698270
##   0.025  10  0.8050257  0.5698375
##   0.050   1  0.8035652  0.5686342
##   0.050   2  0.8061452  0.5722591
##   0.050   3  0.8050300  0.5701609
##   0.050   4  0.8072857  0.5764230
##   0.050   5  0.8065450  0.5760027
##   0.050   6  0.8046682  0.5723452
##   0.050   7  0.8039191  0.5705758
##   0.050   8  0.8050427  0.5730916
##   0.050   9  0.8039232  0.5709112
##   0.050  10  0.8020589  0.5668745
##   0.100   1  0.7964574  0.5501228
##   0.100   2  0.8005649  0.5629921
##   0.100   3  0.7986922  0.5590647
##   0.100   4  0.8009520  0.5652652
##   0.100   5  0.8035738  0.5716418
##   0.100   6  0.8035738  0.5716132
##   0.100   7  0.8035696  0.5718309
##   0.100   8  0.8058127  0.5770441
##   0.100   9  0.8050594  0.5758274
##   0.100  10  0.8039358  0.5737738
##   0.150   1  0.7949551  0.5489682
##   0.150   2  0.7960746  0.5530743
##   0.150   3  0.8028205  0.5690365
##   0.150   4  0.8020756  0.5683638
##   0.150   5  0.8020673  0.5688474
##   0.150   6  0.8009395  0.5667235
##   0.150   7  0.7994245  0.5644875
##   0.150   8  0.7971940  0.5602260
##   0.150   9  0.7971982  0.5603173
##   0.150  10  0.7979431  0.5620271
##
## Accuracy was used to select the optimal model using the largest value.
```

```
## The final values used for the model were sigma = 0.05 and C = 4.
```

Esto nos devuelve que los valores óptimos son sigma = 0.025 y C = 5.

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

Repartimos los datos en train y test y vemos la distribución de supervivencia.

```
library(caret)
set.seed(123)

datos_train <- newds
datos_test  <- newds_test

prop.table(table(datos_train$Survived))
```

```
##
##         0         1
## 0.6161616 0.3838384
```

Vemos que algo más del 61% de los datos de train, pertenece a Survived = 0, algo completamente normal sabiendo que la distribución del total de los datos ronda unos porcentajes similares.

```
modelo_svm <- train(as.factor(Survived)~., method = "svmRadial", data = datos_train, tuneGrid=grid,preP
modelo_svm$finalModel
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.05
##
## Number of Support Vectors : 387
##
## Objective Function Value : -1643.115
## Training error : 0.17284
```

```
summary(modelo_svm$resample$Accuracy)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7500  0.7753  0.8146  0.8114  0.8427  0.8989
```

Después de aplicar SVM obtenemos una accuracy del 89,9% sobre los datos de train, un porcentaje bastante correcto que nos anima a utilizar este modelo para valorar los resultados.

```
## Predicting Test Data
result <- predict(modelo_svm,newdata=datos_test,type='raw')

result<-factor(result)
result
```

```
##   [1] 0 0 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1
##  [38] 1 0 0 0 0 0 1 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0 1 0
##  [75] 1 0 0 1 0 1 0 0 0 0 0 0 1 1 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
## [112] 1 1 1 1 0 0 1 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0
## [149] 0 0 1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 1 0 0 0
## [186] 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 1
## [223] 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 1 0
## [260] 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 1
## [297] 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0
## [334] 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 0 1 0 1 1 0 1 0 0 1 1 0 0
## [371] 1 0 0 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0
## [408] 1 0 1 1 1 0 1 0 0 0
## Levels: 0 1
```

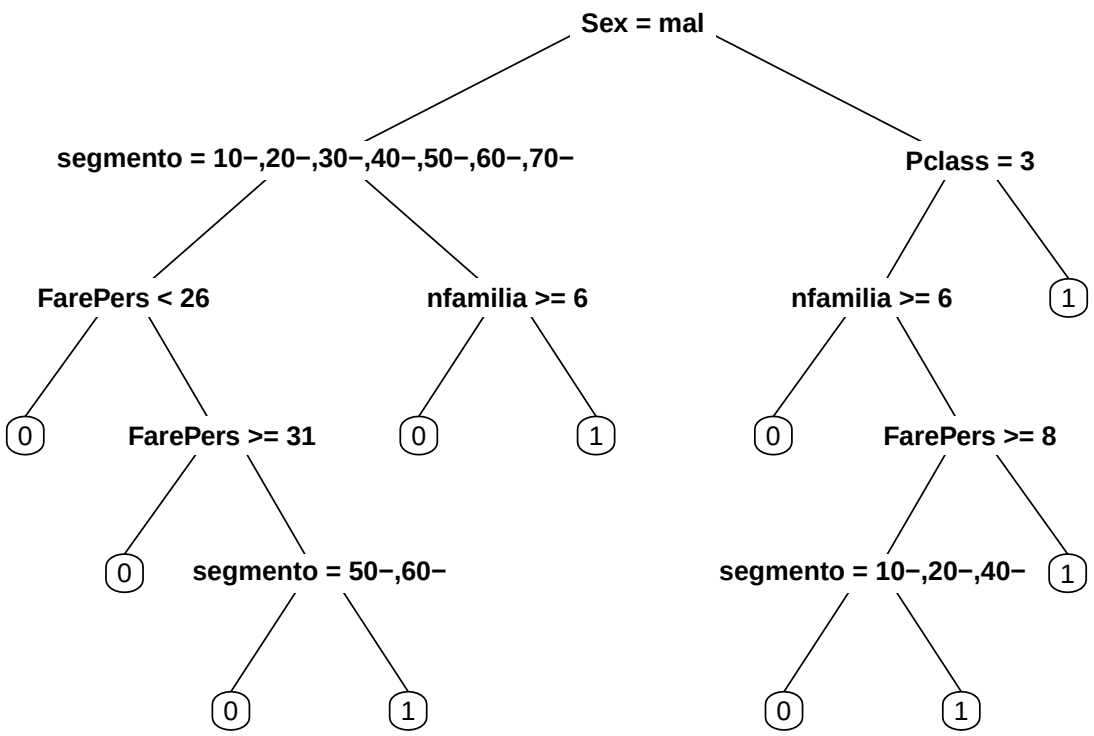
Otra estrategía que podríamos considerar es la de usar un árbol de clasificación con tal de encontrar los patrones que nos lleven a saber si un pasajero sobrevivió o no al accidente.

```
library(rpart)
my_tree<-rpart(Survived ~ Sex + segmento_edad + FarePerson + Pclass + nfamiliares, data = datos_train, m
#plot(my_tree)
#text(my_tree)


#library(rattle) I had lot of trouble installing this... I get this working in the console but when i d

library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.3
```

```
library(RColorBrewer)
new.fit <- prp(my_tree,snip=TRUE)$obj
```

```
## Warning: ignoring snip=TRUE for pdf device
```



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

La exploración de los datos, el estudio de su distribución, y su posible relación con la variable respuesta parecen indicar que los factores que más influyeron en la supervivencia de los pasajeros fueron: el sexo, la clase a la que pertenecían y, en menor medida, si tenían o no al menos un hijo a bordo. También se ha detectado que las variables continuas no están correlacionadas y que las variables Age, Cabin y Embarked tienen valores ausentes. Aún así podemos observar que hay muchos predictores que afectan a que un pasajero sobreviva o no al accidente.

Podemos decir que es mas probable que las mujeres y los niños tenían más probabilidad de sobrevivir que los hombres, algo que refuerza la teoría de mujeres y niños primero y que las personas mayores tienen un índice de supervivencia más bajo, como hemos comentado antes, que podría ser debido a que por movilidad reducida fueran incapaces de llegar a una salida.

También podemos observar que las personas con una clase más baja tienen menos probabilidades de sobrevivir, mientras que con el resto de variables resultan más dificil sacar conclusiones.

Como propuesta de mejora, se podría llegar a analizar los parentescos reforzando el análisis de la familia a partir de los apellidos, algo que podría darnos más información sobre familias que sobrevivieron.

Finalmente, a partir de los resultados obtenidos, podríamos llegar a predecir con un buen margen de acierto si un pasajero sobrevivió o no al accidente basandonos en el modelo entrenado en esta práctica.