Tipología y ciclo de vida de los datos: Prática 2

Güise Lorenzo Rodríguez Aguiar

Detalles de la práctica

Descripción

En el presente documento se seleccionará un dataset existente de Kaggle para aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

Resolución de la práctica

```
# Importación de librerías
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
library(nortest)
library(randomForest)
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:dplyr':
##
##     combine
library(data.table)
##
## Attaching package: 'data.table'
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
library(ggplot2)
##
## Attaching package: 'ggplot2'
## The following object is masked from 'package:randomForest':
##
##     margin
```

Descripción del Dataset

```
# Guardamos el conjunto de test y train en un único dataframe

test <- read.csv('titanic/test.csv', stringsAsFactors = FALSE)
train <- read.csv('titanic/train.csv', stringsAsFactors = FALSE)

data <- bind_rows(train,test)

# Verificamos la estructura del dataset
str(data)
## 'data.frame':    1309 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" ...
```

El conjunto de datos con el que se realizará la práctica es el correspondiente al barco Titanic y sus datos son relativos a las características de los pasajeros utilizando el enlace de Kaggle proporcionado en el enunciado de la práctica. Este dataset está constituido, como podemos apreciar, por 1309 observaciones con 12 atributos cada una. Estos atributos son los siguientes:

- *PassengerId:* Identificador numérico del pasajero.
- *Survived:* Valor booleano que indica si sobrevivió (1: Sí, 0: No).
- *Pclass:* Clase del billete (1 = primera clase, 2 = segunda clase, 3 = tercera clase).
- *Name:* Nombre del pasajero.
- *Sex:* Sexo del pasajero (male o female) .
- *Age:* Edad del pasajero.
- *SibSp:* Nº de hermanos/cónyuges en el Titanic.
- *Parch:* Nº de padres/hijos en el Titanic.
- *Ticket:* Nº de ticket.
- *Fare:* Tarifa abonada por el pasajero.
- *Cabin:* Nº de cabina.
- *Embarked:* Puerto donde embarcaron los pasajeros (C = Cherbourg, Q = Queenstown, S = Southampton).

Importancia y objetivos del análisis

En esta práctica se pretende observar qué variables tuvieron más importancia para sobrevivir en el hundimiento del Titanic. Estas variables permitirán crear sistemas inteligentes que clasifiquen las instancias según el valor de Survived teniendo en cuenta como entrada el resto de sus características y realizar, además, contrastes de hipótesis que nos permitan identificar características interesantes de las muestras y que estas puedan ser inferidas con respecto a la población.

Este análisis puede ser de gran importancia para observar cómo se comportaba la sociedad de principios del siglo XX cuando una tragedia ocurría, pudiendo servir de complemento a investigaciones históricas realizadas por otras ramas de conocimiento como la psicología, la historia o la propia antropología.

Integración y selección de los datos de interés a analizar.

En la tarea anterior realizamos la integración de los datos procedentes del conjunto de train y test creado por Kaggle.

En este apartado observamos cómo hay variables como *PassengerId*, *Name*, *Ticket* y *Cabin* que no nos aportan información de interés para nuestra tarea de análisis. Por ello borramos dichos atributos del dataframe con el que trabajamos:

```
data$PassengerId <- NULL
data$Name <- NULL
data$Ticket <- NULL
data$Cabin <- NULL

str(data)
```

```
## 'data.frame':    1309 obs. of  8 variables:
##  $ Survived: int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass  : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ 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 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: chr  "S" "C" "S" "S" ...
```

De esta manera, usaremos sólo las 7 variables restantes, que pueden observarse en el código superior.

Esperamos al proceso de limpieza de los datos para realizar los procesos de factorización y discretización de las variables, para que sean aplicadas una vez hemos sustituido los valores missing.

Limpieza de los datos

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

Observamos los atributos con valores vacíos de la siguiente manera:

```
colSums(is.na(data))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##      418        0        0      263        0        0        1        0
```

```
colSums(data=="")
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##       NA        0        0       NA        0        0       NA        2
```

Observamos así cómo 418 instancias de nuestro dataset tienen la variable Survived a Null, 283 en el caso de Age y 1 en el caso de Fare. Por otro lado, la variable Embarked tiene dos instancias con una string vacía.

En el caso de Survived, observamos cómo coincide el nº de instancias con valor null con el nº de filas de test que teníamos, es por ello por lo que, en aquellas filas vacías pondremos una nueva clase llamada "NA" (Not available).

En el caso de Age y de Fare reemplazamos los valores perdidos por la media de cada variable en nuestro conjunto de datos. Por último, en el caso de la variable Embarked, al tratarse de solo dos valores nulos lo sustituímos por el valor más frecuente (el puerto del que embarcaron más personas).

```
data$Survived[is.na(data$Survived)] <- "NA"

data$Age[is.na(data$Age)] <- mean(data$Age, na.rm =T)
data$Fare[is.na(data$Fare)] <- mean(data$Fare, na.rm =T)
```

Como podemos observar, el valor más frecuente corresponde al puerto de Southampton. Por lo tanto, utilizamos dicho valor para reemplazarlo.

```r
as.data.frame(table(data$Embarked))
```

```
##   Var1 Freq
## 1         2
## 2    C  270
## 3    Q  123
## 4    S  914
```

```r
data$Embarked[data$Embarked==""]="S"
```

Identificación y tratamiento de valores extremos.

Los outliers o valores extremos son quellos que se diferencian en gran medida del resto de valores del mismo atributo, tanto que si lo viéramos por separado dudaríamos que perteneciera al conjunto original.

Para identificarlos, en la presente práctica utilizaremos la función *boxplots.stats()* de R con cada variable, de forma que se observarán los valores de cada variable que se quedan fuera del rango intercuartílico.

Nos centraremos solo en las variables numéricas que no correspondan a categorías:

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

```
## $stats
## [1]  3.00000 22.00000 29.88114 35.00000 54.00000
##
## $n
## [1] 1309
##
## $conf
## [1] 29.31342 30.44885
##
## $out
##   [1]  2.00 58.00 55.00  2.00 66.00 65.00  0.83 59.00 71.00 70.50  2.00
##  [12] 55.50  1.00 61.00  1.00 56.00  1.00 58.00  2.00 59.00 62.00 58.00
##  [23] 63.00 65.00  2.00  0.92 61.00  2.00 60.00  1.00  1.00 64.00 65.00
##  [34] 56.00  0.75  2.00 63.00 58.00 55.00 71.00  2.00 64.00 62.00 62.00
##  [45] 60.00 61.00 57.00 80.00  2.00  0.75 56.00 58.00 70.00 60.00 60.00
##  [56] 70.00  0.67 57.00  1.00  0.42  2.00  1.00 62.00  0.83 74.00 56.00
##  [67] 62.00 63.00 55.00 60.00 60.00 55.00 67.00  2.00 76.00 63.00  1.00
##  [78] 61.00 60.50 64.00 61.00  0.33 60.00 57.00 64.00 55.00  0.92  1.00
##  [89]  0.75  2.00  1.00 64.00  0.83 55.00 55.00 57.00 58.00  0.17 59.00
## [100] 55.00 57.00
```

Observamos cómo los valores que se quedan fuera del rango intercuantílico de la variable son aquellos pocos frecuentes en la época pero, no por ello, incorrectos.

```r
boxplot.stats(data$SibSp)
```

```
## $stats
## [1] 0 0 0 1 2
##
## $n
## [1] 1309
##
## $conf
## [1] -0.04367041  0.04367041
```

```
##
## $out
##  [1] 3 4 3 3 4 5 3 4 5 3 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 3 5 3 5 3 4 4 3 3
## [36] 5 4 3 4 8 4 3 4 8 4 8 3 4 5 3 4 8 4 8 4 3 3
```

```
boxplot.stats(data$Parch)
```

```
## $stats
## [1] 0 0 0 0 0
##
## $n
## [1] 1309
##
## $conf
## [1] 0 0
##
## $out
##   [1] 1 2 1 5 1 1 5 2 2 1 1 2 2 2 1 2 2 2 3 2 2 1 1 1 1 2 1 1 2 2 1 2 2 2 1
##  [36] 2 1 1 2 1 4 1 1 1 1 2 2 1 2 1 1 1 2 1 1 2 2 2 1 1 2 2 1 2 1 1 1 1 1 1
##  [71] 1 2 1 2 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2
## [106] 2 3 4 1 2 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 4 1 1 2 1
## [141] 2 1 1 2 5 2 1 1 1 2 1 5 2 1 1 1 2 1 6 1 2 1 2 1 1 1 1 1 1 1 3 2 1 1 1
## [176] 1 2 1 2 3 1 2 1 2 2 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 3 2 1 1 1
## [211] 1 5 2 1 1 1 1 3 1 2 2 1 2 1 2 1 2 4 1 1 2 1 1 1 4 6 2 3 1 1 2 2 2 1 1
## [246] 2 5 2 3 2 1 1 1 2 1 2 2 2 1 2 1 1 2 1 2 1 2 1 2 2 1 1 1 1 1 2 1 1 2 1
## [281] 1 1 2 1 2 9 1 1 1 2 2 2 1 9 1 1 2 2 1 1 2 1 1 1 1 1 1
```

Lo mismo ocurre con el nº de hermanos/cónyuges y el nº de padres/hijos. Que los valores se salgan de los percentiles calculados no implica que sean incorrectos.

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

```
## $stats
## [1]  0.0000  7.8958 14.4542 31.2750 65.0000
##
## $n
## [1] 1309
##
## $conf
## [1] 13.43322 15.47518
##
## $out
##   [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750
##   [8]  73.5000 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000
##  [15]  66.6000  69.5500  69.5500 146.5208  69.5500 113.2750  76.2917
##  [22]  90.0000  83.4750  90.0000  79.2000  86.5000 512.3292  79.6500
##  [29] 153.4625 135.6333  77.9583  78.8500  91.0792 151.5500 247.5208
##  [36] 151.5500 110.8833 108.9000  83.1583 262.3750 164.8667 134.5000
##  [43]  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000 263.0000
##  [50]  75.2500  69.3000 135.6333  82.1708 211.5000 227.5250  73.5000
##  [57] 120.0000 113.2750  90.0000 120.0000 263.0000  81.8583  89.1042
##  [64]  91.0792  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000
##  [71] 221.7792 106.4250  71.0000 106.4250 110.8833 227.5250  79.6500
##  [78] 110.8833  79.6500  79.2000  78.2667 153.4625  77.9583  69.3000
##  [85]  76.7292  73.5000 113.2750 133.6500  73.5000 512.3292  76.7292
##  [92] 211.3375 110.8833 227.5250 151.5500 227.5250 211.3375 512.3292
```

```
##  [99]  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583 211.3375
## [106]  79.2000  69.5500 120.0000  93.5000  80.0000  83.1583  69.5500
## [113]  89.1042 164.8667  69.5500  83.1583  82.2667 262.3750  76.2917
## [120] 263.0000 262.3750 262.3750 263.0000 211.5000 211.5000 221.7792
## [127]  78.8500 221.7792  75.2417 151.5500 262.3750  83.1583 221.7792
## [134]  83.1583  83.1583 247.5208  69.5500 134.5000 227.5250  73.5000
## [141] 164.8667 211.5000  71.2833  75.2500 106.4250 134.5000 136.7792
## [148]  75.2417 136.7792  82.2667  81.8583 151.5500  93.5000 135.6333
## [155] 146.5208 211.3375  79.2000  69.5500 512.3292  73.5000  69.5500
## [162]  69.5500 134.5000  81.8583 262.3750  93.5000  79.2000 164.8667
## [169] 211.5000  90.0000 108.9000
```

Por último, observamos un gran número de instancias con un valor de la tarifa abonada por el pasajero que se salen fuera del cuarto percentil. Esto se debe a que estos pasajeros pertenecerán, sin lugar a dudas, a la primera clase del Titanic.

Factorización y discretización de las variables

En este apartado adicional, observaremos en qué variables tendría sentido discretizar sus valores a unas ocas clases:

```
apply(data,2, function(x) length(unique(x)))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        3        3        2       99        7        8      282        3
```

Por ello decidimos discretizar las variables *Survived*, *Pclass*, *Sex* y *Embarked*, ya que tienen pocos valores únicos.

```
values<-c("Survived","Pclass","Sex","Embarked")
for (i in values){
  data[,i] <- as.factor(data[,i])
}
```

De esta manera, los datos quedarán de la siguiente manera:

```
str(data)
```

```
## 'data.frame':    1309 obs. of  8 variables:
##  $ Survived: Factor w/ 3 levels "0","1","NA": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : num  22 38 26 35 35 ...
##  $ SibSp   : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch   : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

Debido a que para realizar los posteriores análisis teniendo en cuenta la variable *Survived* nos interesa que dicha variable sea numérica, realizaremos una copia del dataframe con las propiedades actuales en la variable *data_cat* y realizaremos la transformación comentada en el actual dataframe.

También eliminamos las instancias pertenecientes al conjunto de test original (debido a que tienen la clase NA en la variable *Survived*)

```
data <- data[data$Survived != "NA",]
data_cat <- copy(data)
```

```
data_cat$Survived <- factor(data_cat$Survived)
data$Survived <- as.numeric(as.character(data$Survived))
```

```
str(data)
```

```
## 'data.frame':    891 obs. of  8 variables:
##  $ Survived: num  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : num  22 38 26 35 35 ...
##  $ SibSp   : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch   : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

Análisis de los datos

Selección de los grupos de datos que se quieren analizar/comparar

Dentro de nuestro conjunto de datos hay diferentes grupos que resulta interesante analizar/comparar:

```
# Agrupación por clase en el barco

data.class1 <- data[data$Pclass == "1",]
data.class2 <- data[data$Pclass == "2",]
data.class3 <- data[data$Pclass == "3",]

# Agrupación por sexo

data.male <- data[data$Sex == "male",]
data.female <- data[data$Sex == "female",]

# Agrupación por el puerto de embarque

data.c <- data[data$Embarked == "C",]
data.q <- data[data$Embarked == "Q",]
data.s <- data[data$Embarked == "S",]
```

En la representación gráfica de los datos mostraremos cómo se comportan estas agrupaciones con respecto a la clase Survival, aunque para ello no se utilizarán los dataframes declarados aquí.

Comprobación de la normalidad y homogeneidad de la varianza

Para esta tarea se utilizará la librería de R *nortest* (test Anderson-Darling) para comprobar que los valores de las variables numéricas pertenecen a una población que sigue una distribución normal.

Para realizarlo, se verifica que en las pruebas el p-valor es superior al valor prefijado de alpha 0.05. En caso de que esto se cumpla, se considerá que la variable en cuestión sigue una distribución normal.

```
alpha = 0.05
columnas = colnames(data)

print("Variables que no siguen una distribución normal:")
```

```
## [1] "Variables que no siguen una distribución normal:"
```

```
for (i in 1:ncol(data)){
  if (is.numeric(data[,i])){
    if (ad.test(data[,i])$p.value < alpha){
      print(columnas[i])
    }
  }
}
```

```
## [1] "Survived"
## [1] "Age"
## [1] "SibSp"
## [1] "Parch"
## [1] "Fare"
```

También se analiza en este apartado la homogeneidad de varianzas respecto del grupo de hombres del titanic frente a las mujeres. Para ello utilizamos el test de Fligner-Killeen, donde partimos de la hipótesis nula que consiste en que las varianzas son iguales:

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Sex
## Fligner-Killeen:med chi-squared = 5.7729, df = 1, p-value =
## 0.01627
```

Debido a que obtenermos un valor inferior a 0.05, no aceptamos la hipótesis de que ambas muestras son homogéneas.

Aplicación de pruebas estadísticas para comparar los grupos de datos

¿Podemos predecir la muerte de los pasajeros en base al resto de variables?

En esta tarea observaremos, utilizando para ello un random forest (modelo de clasificación), la importancia que tienen las variables utilizadas para clasificar a los pasajeros según si han sobrevivido o no.

Para ello necesitamos quedarnos solo con las instancias que tengan en el valor de Survived los valores de 0 (No) y 1 (Sí), siendo estos factores. Es por ello, por lo que utilizamos el dataframe data_cat.

Debido a que sólo nos interesa conocer cómo realiza la división de las instancias en base de sus atributos, realizaremos el entrenamiento del modelo con todo el dataset (aunque podremos conocer la capacidad de predicción en base al Out-of-Bag error rate).

```
summary(data_cat)
```

```
##  Survived Pclass      Sex           Age            SibSp
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000
##  1:342    2:184   male  :577   1st Qu.:22.00   1st Qu.:0.000
##           3:491                Median :29.88   Median :0.000
##                                Mean   :29.74   Mean   :0.523
##                                3rd Qu.:35.00   3rd Qu.:1.000
##                                Max.   :80.00   Max.   :8.000
##      Parch             Fare        Embarked
##  Min.   :0.0000   Min.   :  0.00   C:168
##  1st Qu.:0.0000   1st Qu.:  7.91   Q: 77
##  Median :0.0000   Median : 14.45   S:646
```

```
##  Mean   :0.3816   Mean   : 32.20
##  3rd Qu.:0.0000   3rd Qu.: 31.00
##  Max.   :6.0000   Max.   :512.33
```

De esta manera, realizamos la clasificación de la siguiente forma:

```
set.seed(2019)
model <- randomForest(Survived ~ ., data= data_cat, importance = TRUE)
model
```

```
##
## Call:
##  randomForest(formula = Survived ~ ., data = data_cat, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 16.95%
## Confusion matrix:
##     0   1 class.error
## 0 502  47   0.0856102
## 1 104 238   0.3040936
```

```
importance(model)
```

```
##                  0         1 MeanDecreaseAccuracy MeanDecreaseGini
## Pclass   19.603036 32.438629             39.55259         33.62682
## Sex      63.242007 83.627064             87.49389        104.04643
## Age      20.387446 24.369352             30.27903         53.18031
## SibSp    19.933992  5.075115             21.72743         16.49647
## Parch    13.602611  9.208306             16.54425         12.84268
## Fare     20.526356 24.723515             35.88984         64.32797
## Embarked  8.167794 16.281965             17.97208         11.80456
```

La columna de MeanDecreaseAccuracy se basa en cuánto accuracy decrece si no se utiliza dicha variable mientras que MeanDecreaseGini ese basa en el descenso de la impuridad de Gini cuando la variable se utiliza en un nodo para realizar la división del dataset.

Según la primera métrica, las variables más importantes son: *Sex*, *Pclass*, *Fare*, *Age* y *SibSp*.

Por otro lado, este orden cambia si nos basamos en los valores de la media de decrecimiento de Gini: *Sex*, *Fare*, *Age*, *Pclass* y *SibSp*.

Damos por hecho que las variables *Pclass* y *Fare* están correlacionadas, ya que a mayor clase en el barco, por lógica, mayor gasto en el ticket. Esta correlación la investigaremos en el apartado de representación gráfica del dataset.

De esta manera, observamos cómo el sexo, el gasto realizado en el billete (o la categoría del mismo), la edad y el nº de hermanos o cónyuges influía en determinar quién se salvaba del hundimiento del barco.

¿Qué variables numéticas influyen más en la supervivencia del pasajero?

En este punto, observamos si existen correlaciones entre las variables numéricas de nuestro conjunto de datos.

Para ello seleccionamos las variables numéricas utilizando la función *is.numeric()* junto a *sapply()* y utilizamos dicha selección para filtrar las columnas de nuestro dataset.

Debido a que ninguna de las variables numéricas siguen una distribución normal, hecho indispensable para utilizar el método de correlación de *Pearson*, se ha optado por el método de Spearman.

```
cor(data[, sapply(data, is.numeric)], method = "spearman")
```

```
##             Survived         Age       SibSp      Parch      Fare
## Survived  1.00000000 -0.03910946  0.08887948  0.1382656 0.3237361
## Age      -0.03910946  1.00000000 -0.14703452 -0.2172899 0.1188471
## SibSp     0.08887948 -0.14703452  1.00000000  0.4500140 0.4471130
## Parch     0.13826563 -0.21728987  0.45001397  1.0000000 0.4100738
## Fare      0.32373614  0.11884708  0.44711299  0.4100738 1.0000000
```

Así es cómo comprobamos que, aparentemente, ninguna de las variables tienen un nivel de correlación significativa para que sean descartadas o se considere que una influye en otra, tal y cómo planteábamos en el inicio de este apartado.

¿Se puede estimar la edad de los pasajeros del titanic utilizando un modelo de regresión?

Para realizar esta tarea, utilizamos la función *lm* de R.

```
regresor = lm(Age~., data=data)
print(regresor)
```

```
##
## Call:
## lm(formula = Age ~ ., data = data)
##
## Coefficients:
## (Intercept)     Survived      Pclass2      Pclass3      Sexmale
##    42.40247     -5.46936     -9.37395    -13.91415     -0.53144
##       SibSp        Parch         Fare    EmbarkedQ    EmbarkedS
##    -2.01750     -1.18127     -0.01412      3.86833      1.38753
```

```
summary(regresor)
```

```
##
## Call:
## lm(formula = Age ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.271  -8.170  -0.211   5.800  44.765
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.40247    1.76323  24.048  < 2e-16 ***
## Survived     -5.46936    1.01769  -5.374 9.85e-08 ***
## Pclass2      -9.37395    1.36237  -6.881 1.13e-11 ***
## Pclass3     -13.91415    1.26436 -11.005  < 2e-16 ***
## Sexmale      -0.53144    1.01263  -0.525   0.5998
## SibSp        -2.01750    0.39744  -5.076 4.70e-07 ***
## Parch        -1.18127    0.55825  -2.116   0.0346 *
## Fare         -0.01412    0.01043  -1.353   0.1764
## EmbarkedQ     3.86833    1.69301   2.285   0.0226 *
## EmbarkedS     1.38753    1.07115   1.295   0.1955
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 11.64 on 881 degrees of freedom
## Multiple R-squared:  0.2062, Adjusted R-squared:  0.1981
## F-statistic: 25.42 on 9 and 881 DF,  p-value: < 2.2e-16
```

En base a los valores de R cuadrado y el R cuadrado ajustado, podemos verificar que si utilizamos las variables en cuestión, esta regresión lineal no se ajusta a la edad del pasajero. Es por ello por lo que se debería realizar un estudio en mayor profundidad de las variables y probar diferentes modelos.

A continuación, intentaremos realizar la regresión de la variable Age utilizando un Random Forest Regresor. Para ello, como en el modelo del clasificador, utilizamos todo el conjunto de datos para el entrenamiento del modelo, ya que lo que más nos interesa de este modelo es conocer la importancia de las diferentes variables para predecir la edad del pasajero.

```
set.seed(2019)
model_regresor <- randomForest(Age ~ ., data= data_cat, importance = TRUE)
model_regresor
```

```
##
## Call:
##  randomForest(formula = Age ~ ., data = data_cat, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 119.1134
##                     % Var explained: 29.46
```

```
importance(model_regresor)
```

```
##           %IncMSE IncNodePurity
## Survived 28.63076      5016.829
## Pclass   36.39069     13524.340
## Sex      18.63756      3352.848
## SibSp    35.71470     12149.467
## Parch    38.29093     14616.473
## Fare     30.55258     22397.705
## Embarked 10.65357      3041.405
```

En este modelo observamos cómo la media de cuadrados residuales y la varianza explicada es bastante elevada para considerar este modelo aceptable. Por ello, consideramos que las variables utilizadas no permiten al modelo estimar correctamente la edad de los pasajeros.

Respecto a la importancia de las variables, observamos cómo las variables que más incrementan el error cuadrático medio de nuestro modelo son *Parch*, *Pclass* y *SibSp*. Sin embargo, al no representar este modelo la realidad debido a su incapacidad de explicar la varianza de la variable edad, estos resultados de importancia no son fiables.

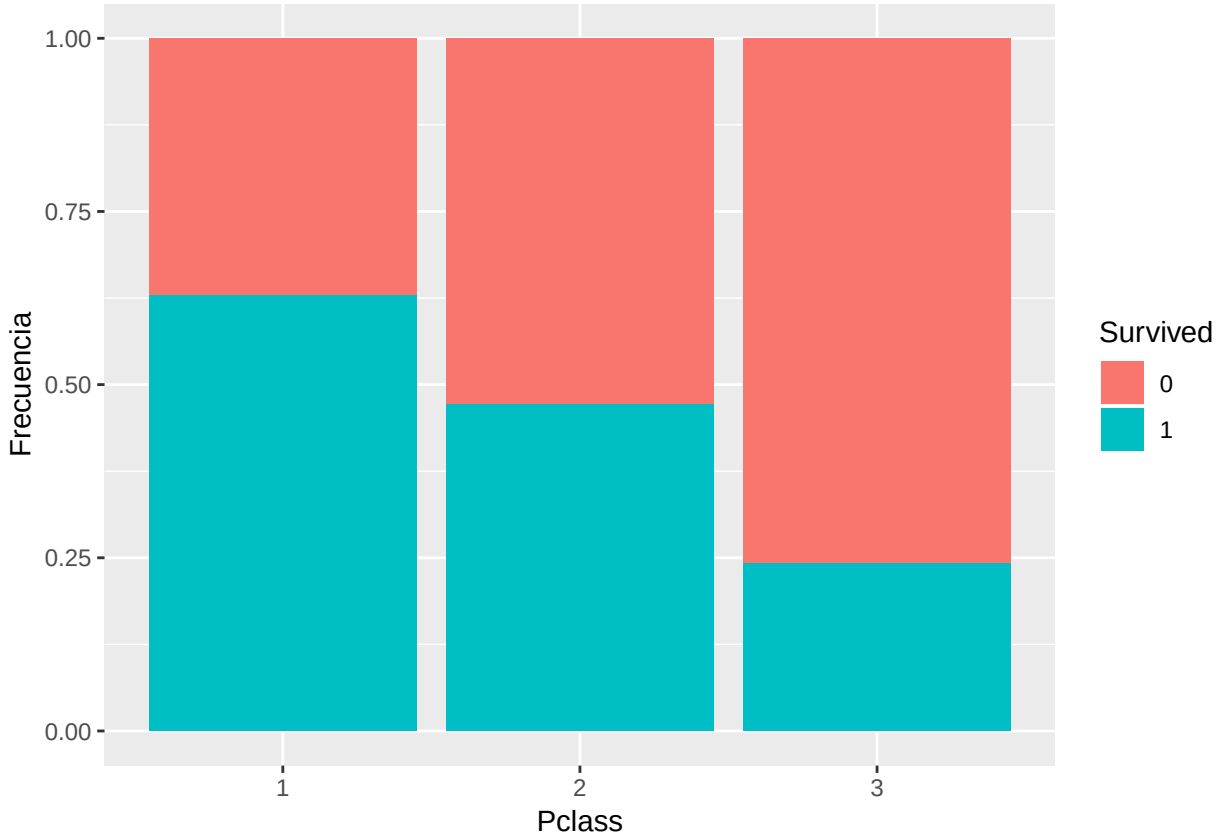
```
str(data_cat)
```

```
## 'data.frame':    891 obs. of  8 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age     : num  22 38 26 35 35 ...
##  $ SibSp   : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch   : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare    : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked: Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

Representación de los resultados a partir de tablas y gráficas.

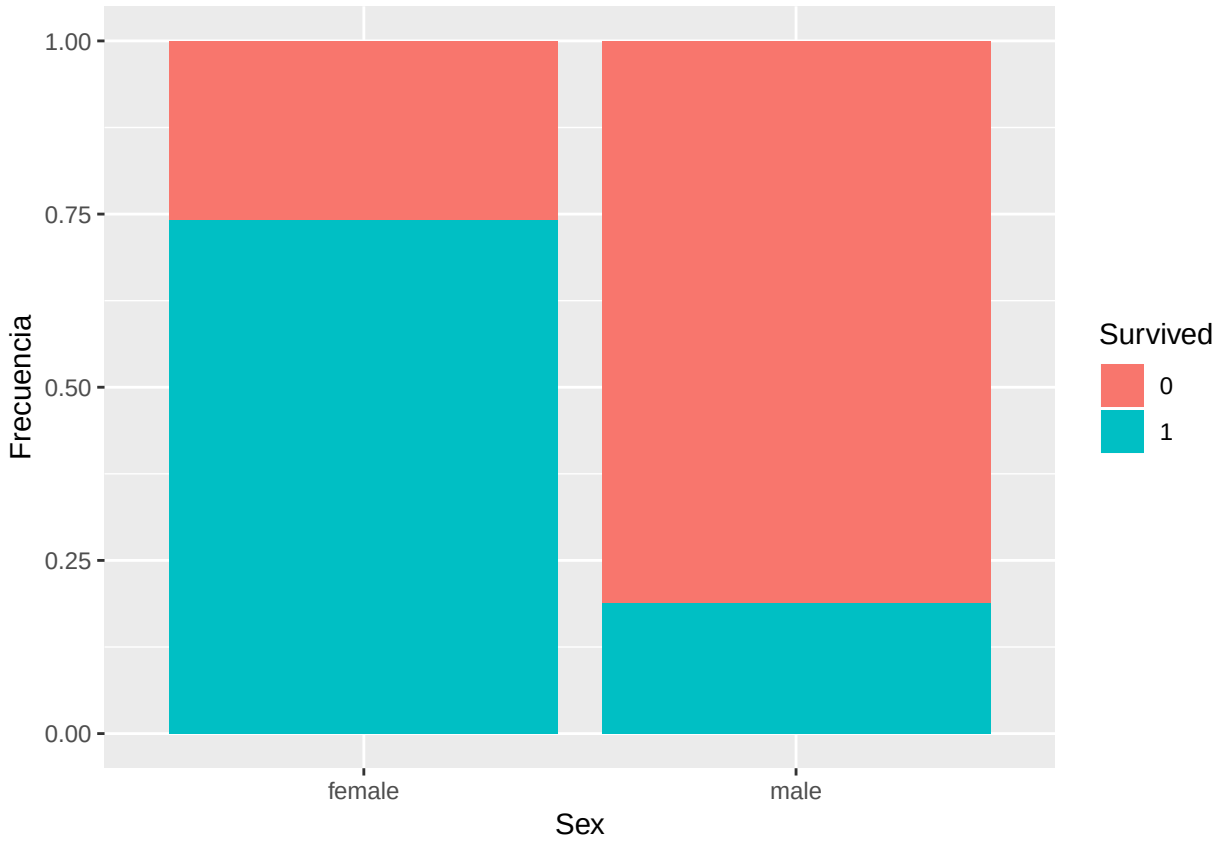
En esta table enfrentaremos, en primer lugar las variables Pclass, Sex y Embarked frente a Survived:

```
ggplot(data = data_cat[1:dim(data_cat)[1],],aes(x=Pclass,fill=Survived))+geom_bar(position="fill")+ylab
```



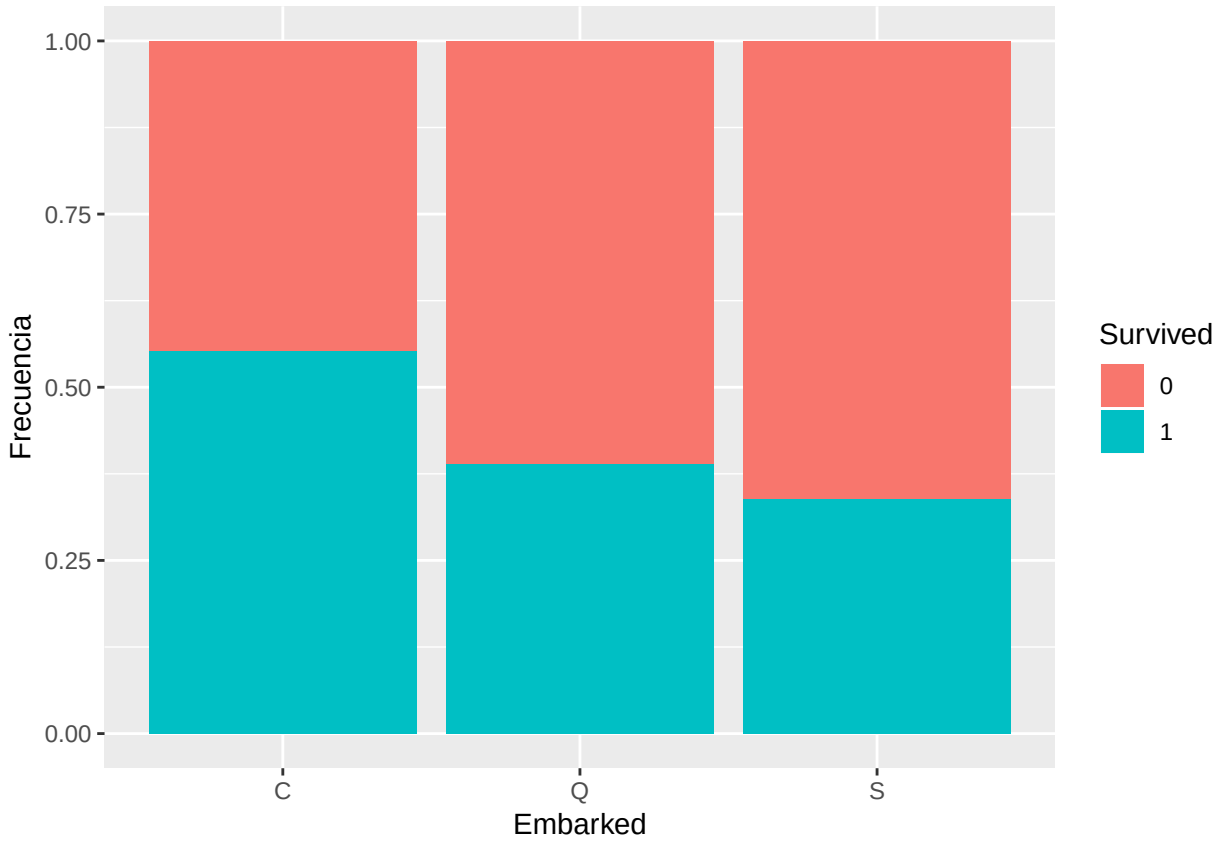
En esta gráfica podemos observar la correlación entre estar en una clase de mayor categoría y la mayor probabilidad de salvarse del hundimiento del Titanic.

```
ggplot(data = data_cat[1:dim(data_cat)[1],],aes(x=Sex,fill=Survived))+geom_bar(position="fill")+ylab("F
```



Por otro lado, observamos cómo casi el 75% de las mujeres se salvaron del accidente mientras que menos del 25% de los hombres lograron sobrevivir.

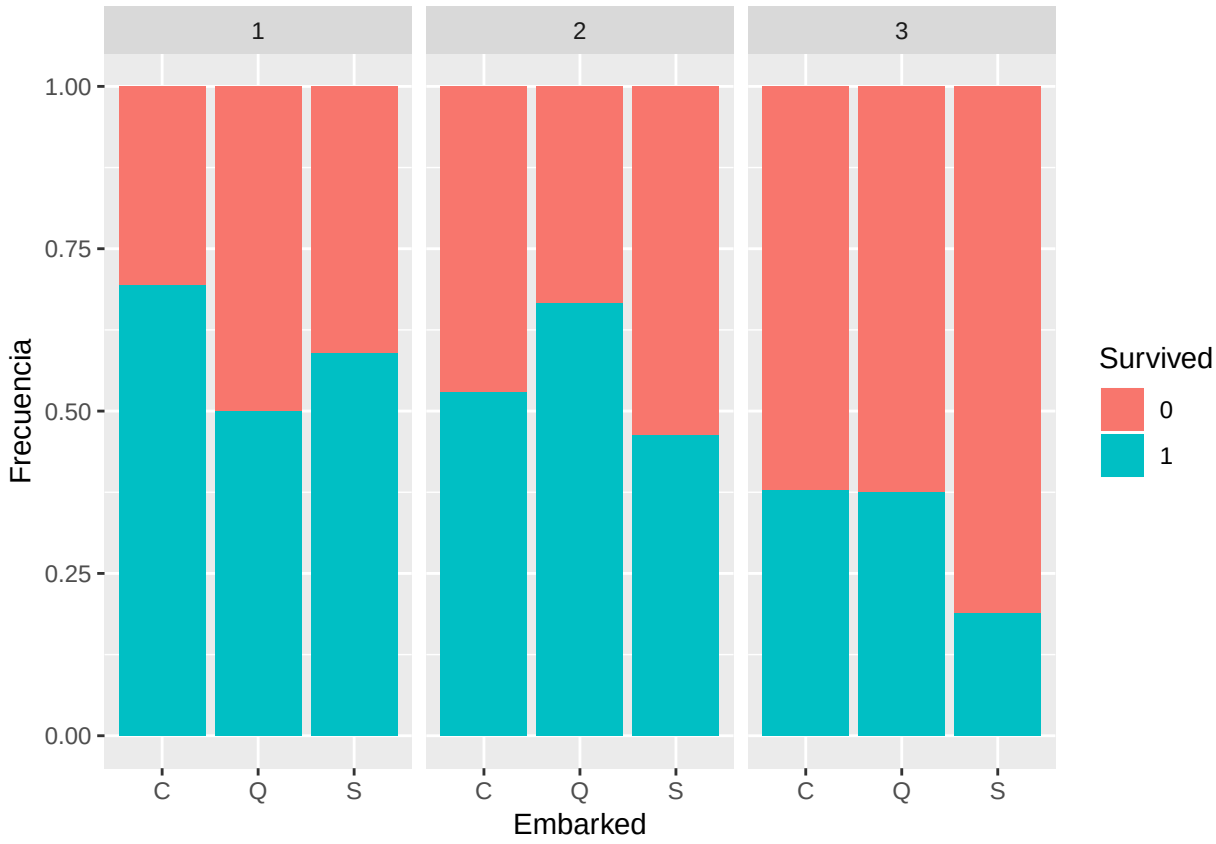
```
ggplot(data = data_cat[1:dim(data_cat)[1],],aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+yla
```



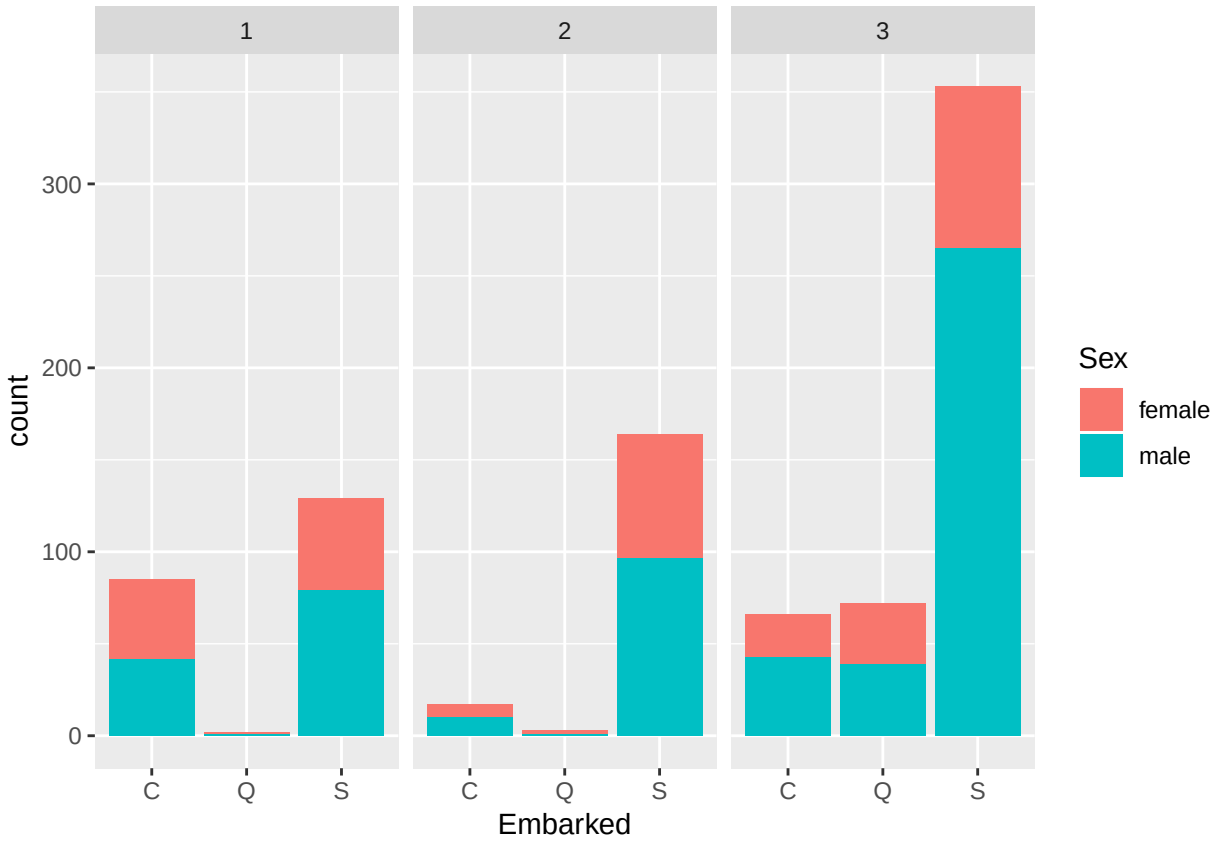
En esta gráfica, podemos observar cómo los pasajeros que embarcaron del puerto de Cherbourg tenían más probabilidades de salvarse que aquellos que embarcaron en los otros puertos. Seguramente se deba al nº de familias con niños o de mujeres que embarcaron desde dicho puerto.

Para comprobar la última hipótesis, realizamos las siguientes gráficas:

```
ggplot(data = data_cat[1:dim(data_cat)[1],],aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+fa
```



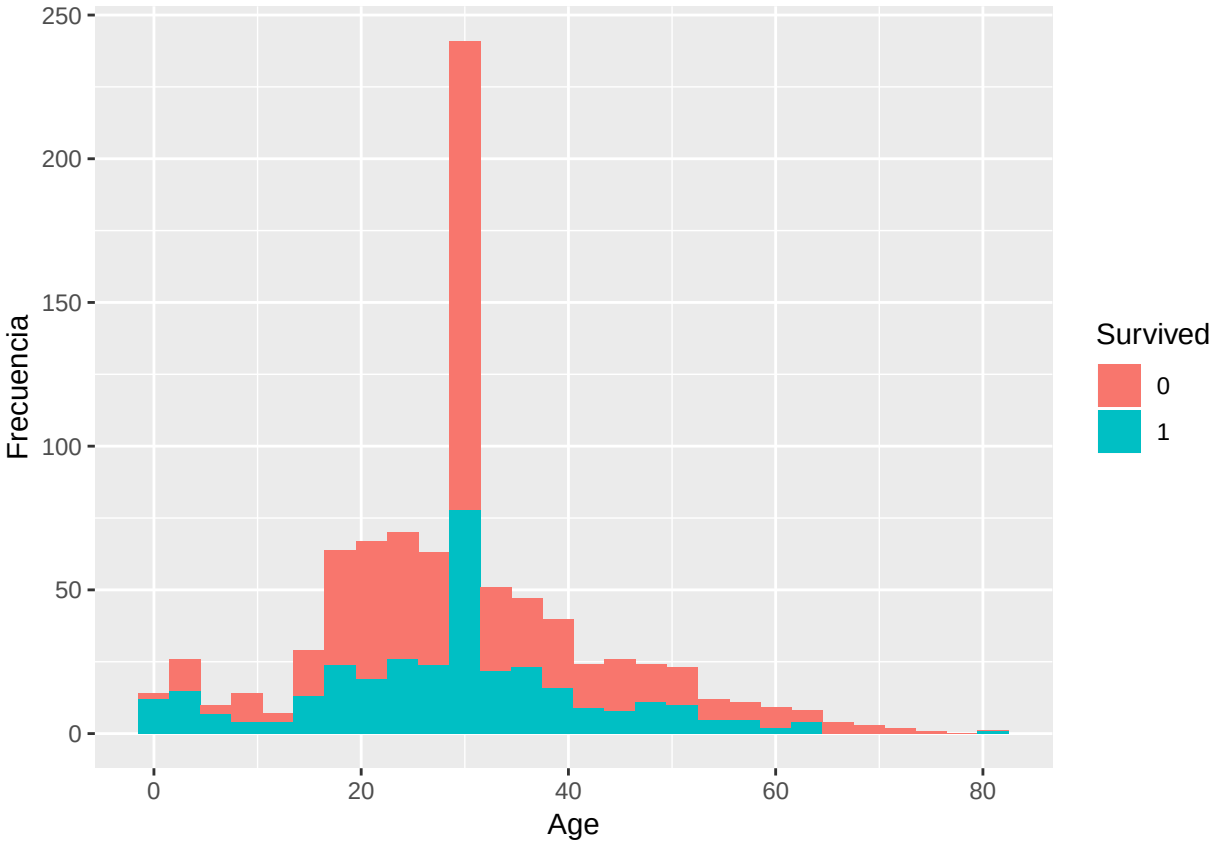

```
ggplot(data = data_cat[1:nrow(data_cat),],aes(x=Embarked,fill=Sex))+geom_bar()+facet_wrap(~Pclass)
```



A pesar de que creíamos que se debía al nº de mujeres o de niños que embarcaron en el puerto de Cherbourg, estas últimas gráficas nos hacen creer en la hipotesis de que los hombres de C fueron alojados en camarotes mejor posicionados en el proceso de desembarco del Titanic, pudiendo salvarse en gran medida junto a las mujeres.

A continuaciñon observaremos cómo se distribuye la tasa de supervivientes en las diferentes edades de los pasajeros:

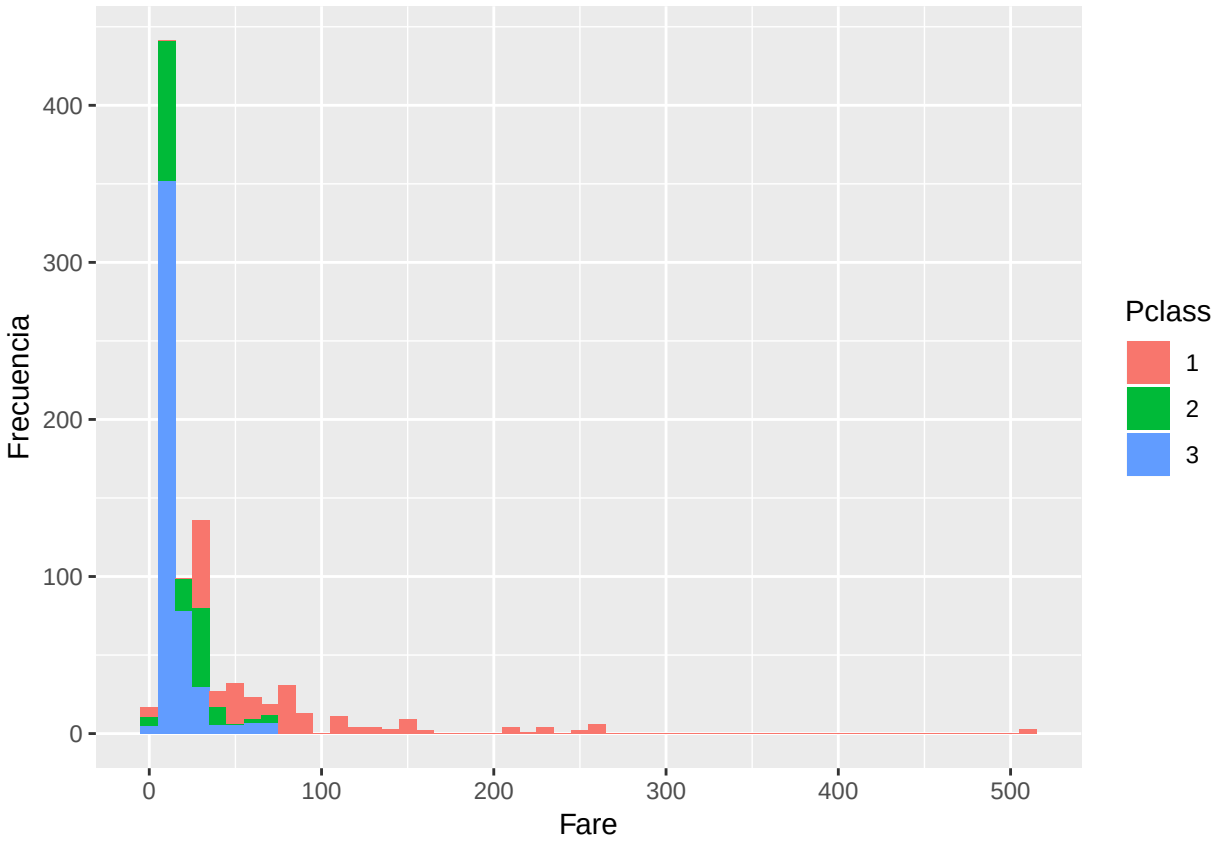
```
ggplot(data = data_cat[!(is.na(data_cat[1:dim(data_cat)[1],]$Age)),],aes(x=Age,fill=Survived))+geom_his
```



Es en esta gráfica donde se observa que, los menores de 15 años tenían mayor probabilidad de salvarse que el resto de las personas del barco.

Por último, comprobamos visualmente que la variable Pclass y Fare están correlacionadas, como nos indica la lógica:

```
ggplot(data = data_cat[!(is.na(data_cat[1:dim(data_cat)[1],]$Fare)),],aes(x=Fare,fill=Pclass))+geom_his
```



Por mencionar alguna de las tablas con las que hemos trabajado en los anteriores apartados, en en modelo de clasificación descubrimos que las variables *Sex*, *Pclass*, *Fare*, *Age* y *SibSp* eran a las que el modelo otorgaba una mayor importancia.

Por otro lado, aunque el modelo de regresión con Random Forest no era aceptable, indicaba que Parch, Pclass y SibSp eran las variables que mayor información aportaban para realizar la regresión correctamente.

Resolución del problema.

Aunque la resolución de las diferentes cuestiones planteadas al principio de esta práctica se han explicado en los diferentes apartados desarrollados en la misma, se aprovechará este apartado para resumirlas:

- Las variables han permitido al modelo de clasificación determinar con un valor bajo de error (para ser un clasificador inicial) qué pasajeros sobrevivieron.
- Las variables numéricas seleccionadas no siguen una distribución normal.
- No se han encontrado variables numéricas correlacionadas entre sí, aunque la que mayor valor de correlación tiene con Survived es Fare, es decir, el coste del billete. Esto también lo hemos deducido de las gráficas que enfrentan Pclass y Survived.
- También hemos comprobado que, para ninguno de nuestros dos modelos regresores, el resto de variables permiten predecir de forma aceptable la edad de los pasajeros.
- En la visualización gráfica, hemos comprobado cómo están correlacionadas las variables *Pclass*, *Sex* y *Age*, tal y cómo mostraba la función *importance* al pasarle por parámetro el modelo de Random Forest de clasificación utilizado.

Código

El código se encuentra disponible en el presente fichero, que está disponible tanto en formato pdf como Rmd. Este último formato se ofrece para la ejecución y la comprobación de los resultados.

También está disponible en el respositorio los datos originales y los datos finales con los que se ha trabajado.

```
write.csv(data, file = "titanic_modified.csv")
```

Componentes del proyecto

Debido a que trabajo y a que mis horarios me complican coordinarme con otras personas, la profesora colaboradora de la asignatura me autorizó por email realizar la práctica de forma individual sin penalización.

Contribuciones	Firma
Investigación Previa	GLRA
Redacción de las respuestas	GLRA
Desarrollo código	GLRA