Práctica 2

Oscar Ramirez González y Alejandro Prieto Velasco

26/5/2021

Contents

Presentación

En esta práctica se elaborará un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

Para hacer esta práctica trabajamos en un grupo de 2 personas. Tenemos que entregar un solo archivo con el enlace Github (https://github.com)) donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo.

Se puede utilizar la Wiki de Github para describir nuestro equipo y los diferentes archivos que corresponden a nuestra entrega. Cada miembro del equipo tiene que contribuir con su usuario Github.

Competencias

En esta práctica se desarrollan las siguientes competencias del Master 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 dato 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.

Descripción del dataset.

Con el análisis del dataset del Titánic se quiere mostrar cómo fue la distribución de la gente que murió y sobrevivió al hundimiento del barco según diversos parámetros, realizando un análisis detallado de los datos disponibles en el dataset como son el sexo, la clase contratada para realizar el viaje, si viajaban con familia o no, etc.

El dataset original está compuesto por un total de 1309 pasajeros, y tiene 14 variables.

Para nuestro estudio hemos suprimido dos variables que consideramos irrelevantes para el trabajo que queremos hacer sobre el dataset, que son:

- Body: número de identificación del cuerpo
- Home.dest: lugar de origen y de destino del pasajero

Se ha dividido el dataset en dos conjuntos, uno de train con 891 pasajeros, y otro de test con 418 pasajeros, para poder hacer una predicción con el modelo creado sobre si cada uno de los 418 pasajeros habría sobrevivido o no siguiendo el modelo creado.

A continuación, se da una descripción detallada de cada una de las 12 variables utilizadas en el dataset:

1. PassengerId: número único de orden para identificar a cada uno de los pasajeros.
2. Survived: 0 = no sobrevivio, 1 = si sobrevivio

3. Pclass: clase en la que realizaba el viaje cada pasajero.

· 1: primera clase

· 2: segunda clase

· 3: tercera clase

4. Name: nombre completo de cada pasajero

5. Sex: sexo de cada pasajero

6. Age: edad de cada pasajero

7. SibSp: número de familiares a bordo de cada pasajero de estos tipos: hermano, hermana, cuñado, cuñada, pareja. Prometidos y amantes no incluidos

8. Parch: número de familiares a bordo de cada pasajero de estos tipos: madre, padre, hijo, hija, hijastro, hijastra.

9. Ticket: número de billete

10. Fare: tárifa del billete en libras inglesas

11. Cabin: número de camarote

12. Embarked: lugar de embarque

· C=Cherbourg;

· Q = Queenstown

· S = Southampton

Este dataset ha sido creado por varios investigadores, entre ellos:

· Eaton & Haas (1994) Titanic: Triumph and Tragedy. Fueron uno de los creadores originales.

· Patrick Stephens Ltd, incluye una lista de pasajeros creado por muchos investigadores y editada por Michael A. Findlay.

Con el análisis detallado de este dataset se persigue saber de una manera lo más objetiva posible si las circunstancias de cada pasajero (sexo, edad, si iba con familiares, la clase en la que viajaba, etc) influyó de alguna manera en la supervivencia o no de los pasajeros. Además, se incluye un modelo de predicción que nos permite estimar si un pasajero habría sobrevivido o no al hundimiento del Titanic en función del análisis realizado y de los datos de dicho pasajero.

Librería

```
if (!require(ggplot2)) {
  install.packages('ggplot2')
  library(ggplot2)
}

if (!require(plyr)) {
  install.packages("plyr")
  library(plyr)
}


if (!require(dplyr)) {
```

```
  install.packages("dplyr")
  library(dplyr)
}

if (!require(gridExtra)) {
  install.packages("gridExtra")
  library(gridExtra)
}
```

Integración y análisis inicial

El dataset viene ya dividido en train y test, en los datos set no está la variable objetivo "Survived" la cual es la que hay predecir para la competición. Vamos a unir ambos para poder trabajarlos de forma más sencilla, una vez los datos hayan sido limpiados y tratados volveremos a separarlos.

```
# Loading datest
train <- read.csv("train.csv")
test <- read.csv("test.csv")

# Combine dataset
full_original <- bind_rows(train, test)

# Create a copy to work with

full <- full_original
```

Ahora vamos a estudiar un poco el contenido del dataset, primero vamos a ver las 10 primeras filas y un primer summary.

```
head(full, 10)
```

```
##    PassengerId Survived Pclass
## 1            1        0      3
## 2            2        1      1
## 3            3        1      3
## 4            4        1      1
## 5            5        0      3
## 6            6        0      3
## 7            7        0      1
## 8            8        0      3
## 9            9        1      3
## 10          10        1      2
##                                                  Name    Sex Age SibSp Parch
## 1                             Braund, Mr. Owen Harris   male  22     1     0
## 2  Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                              Heikkinen, Miss. Laina female  26     0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                            Allen, Mr. William Henry   male  35     0     0
## 6                                    Moran, Mr. James   male  NA     0     0
## 7                             McCarthy, Mr. Timothy J   male  54     0     0
## 8                      Palsson, Master. Gosta Leonard   male   2     3     1
```

```
## 9    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) female  27     0     2
## 10                 Nasser, Mrs. Nicholas (Adele Achem) female  14     1     0
##              Ticket    Fare Cabin Embarked
## 1         A/5 21171  7.2500              S
## 2          PC 17599 71.2833   C85        C
## 3  STON/O2. 3101282  7.9250              S
## 4            113803 53.1000  C123        S
## 5            373450  8.0500              S
## 6            330877  8.4583              Q
## 7             17463 51.8625   E46        S
## 8            349909 21.0750              S
## 9            347742 11.1333              S
## 10           237736 30.0708              C
```

```
# First summary of the info
summary(full)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age            SibSp            Parch
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :28.00   Median :0.0000   Median :0.000
##                     Mean   :29.88   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   :80.00   Max.   :8.0000   Max.   :9.000
##                     NA's   :263
##     Ticket               Fare            Cabin             Embarked
##  Length:1309        Min.   :  0.000   Length:1309        Length:1309
##  Class :character   1st Qu.:  7.896   Class :character   Class :character
##  Mode  :character   Median : 14.454   Mode  :character   Mode  :character
##                     Mean   : 33.295
##                     3rd Qu.: 31.275
##                     Max.   :512.329
##                     NA's   :1
```

Vemos como el set contiene un total de 1309 registros de los cuales 418 pertenecen al set test (son los NA's de la columna "Survived") y el resto a al set de entrenamiento.

Observamos como la variable "Survived" toma valores de 1 (sobrevive) y 0 (muere), vamos a cambiar la nomenclatura para que se más fácil de entender.. Vemos como la variable Pclass está como integer, quizá interese cambiarla a factor ya que en realidad esta es una variable categórica, el igual que Sex, Ticket y Embark, aunque estas las ha leído como caracteres.

Ahora vamos cambiar el formato de las columnas.

```
# Renaming 0 and 1 to "Diddn't survived" and "Survived"
full$Survived[full$Survived == 0] <- "Didn't survived"
```

```r
full$Survived[full$Survived == 1] <- "Survived"

# Changing to factor
full$Survived <- as.factor(full$Survived)
full$Pclass <- as.factor(full$Pclass)
full$Sex <- as.factor(full$Sex)
full$Embarked <- as.factor(full$Embarked)
```

Limpieza de los datos.

En esta etapa del proyecto vamos primero a identificar y tratar aquellos datos que estén vacíos, y después vamos a pasar a tratar los outliers.

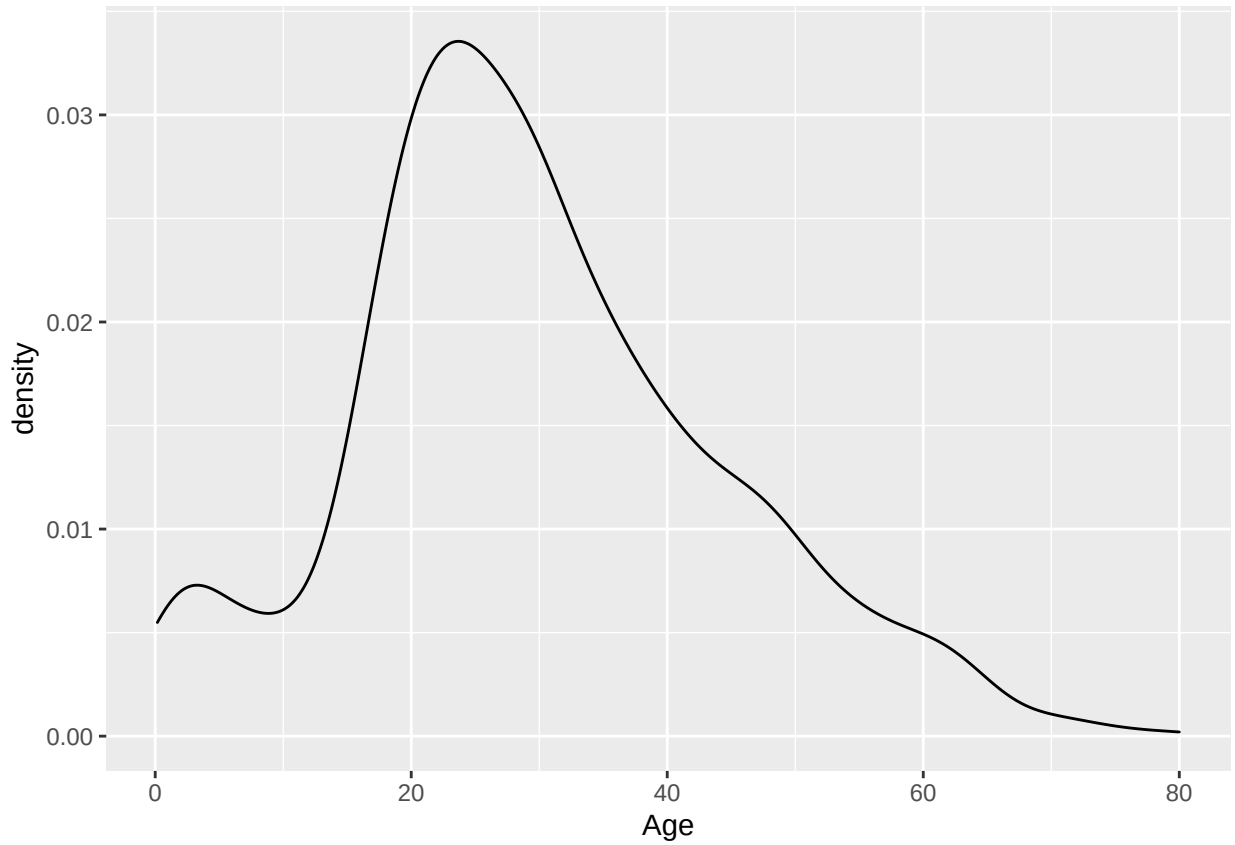
Datos vacíos

```r
# Let's see the NA by columns
colSums(is.na(full))
```

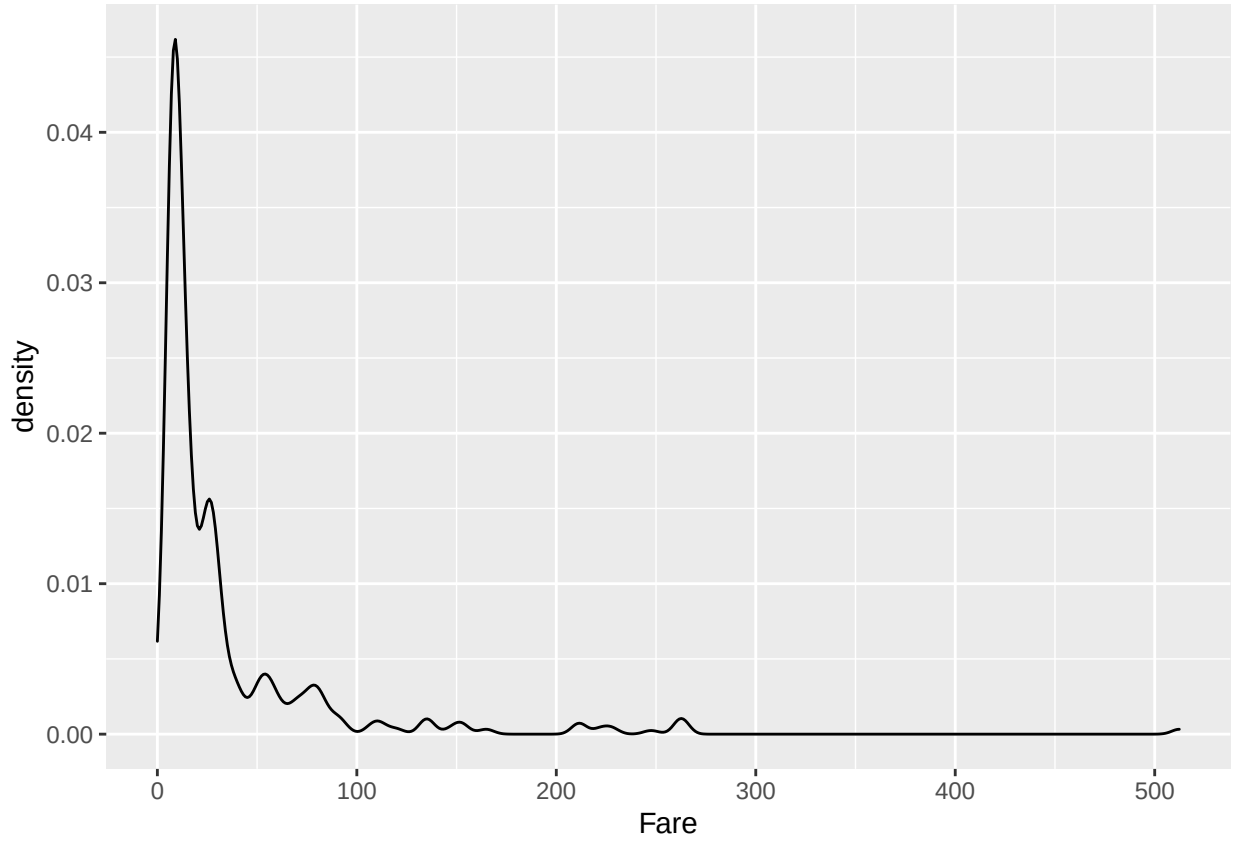
```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1           0           0
```

Vemos que hay 263 NA's en Age y 1 en Fare. Vamos a ver las distribuciones de estas variables para así poder entender mejor como se distribuyen estas dos variables.

```r
# Plotting histogram of Age
ggplot(full, aes(x = Age)) + geom_density()
```




```
# Plotting histogram of Fare
ggplot(full, aes(x = Fare)) + geom_density()
```



Vemos que tanto Age como Fare tienen una asimetría positiva esto junto al hecho de que el dataset es pequeño y no vamos a considerar borrar aquellas lineas con NA, nos hace tomar la decisión de imputar estos NA por la mediana

```
# Filling NA in Age with miss forest
full$Age[is.na(full$Age)] <- median(full$Age, na.rm = TRUE)

# Filling Fare NA with median
full$Fare[is.na(full$Fare)] <- median(full$Fare, na.rm = TRUE)
```

También hay un serie de NA que a pesar de no haber aparecido inicialmente están ahí. Vamos a verlos.

```
colSums(full == "")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0          NA           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0        1014           2
```

Parece ser que en cabin la gran mayoría de valores están vacíos, vamos a sustituir estos por Unknown.

```
# Replacing epmpty values for Unknown
full$Cabin[full$Cabin == ""] <- "Unknown"
```

Al ser solo 2 vamos a sustituir los valores vacíos de embarked por la clase mayoritaria

```r
# Group by and count by embarked
full %>%
  group_by %>%
  count(Embarked)
```

```
## # A tibble: 4 x 2
##   Embarked     n
##   <fct>    <int>
## 1 ""           2
## 2 "C"        270
## 3 "Q"        123
## 4 "S"        914
```

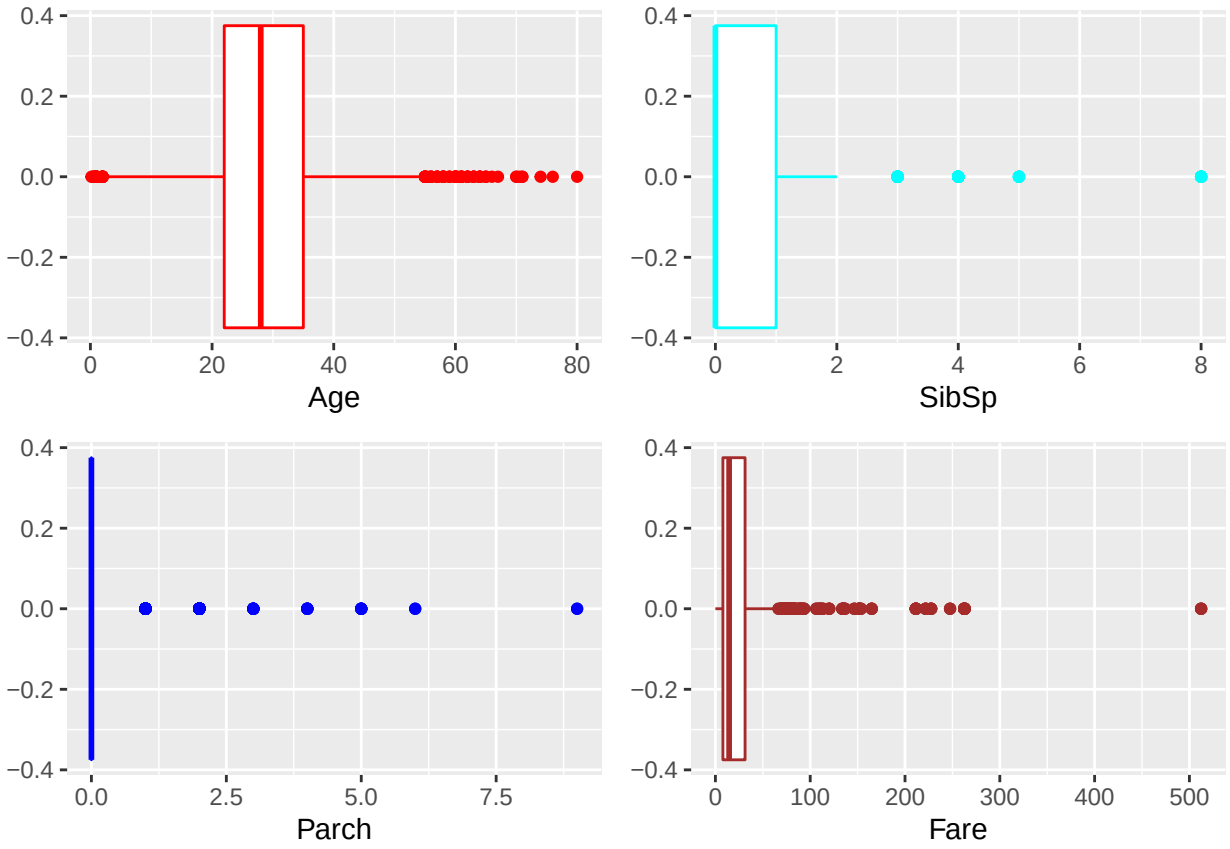
```r
# Replace empties for S
full$Embarked[full$Embarked == ""] <- "S"
```

Tratamiento de outliers

Vamos a identificar los posibles outliers en la variables numéricas mediante boxplots.

```r
# Making boxplots
box_age <-
  ggplot(full, aes(x = Age)) + geom_boxplot(colour = "red")
box_SibSp <-
  ggplot(full, aes(x = SibSp)) + geom_boxplot(colour = "cyan")
box_Parch <-
  ggplot(full, aes(x = Parch)) + geom_boxplot(colour = "blue")
box_Fare <-
  ggplot(full, aes(x = Fare)) + geom_boxplot(colour = "brown")


# Grouping them in a grid
grid.arrange(box_age, box_SibSp, box_Parch, box_Fare)
```



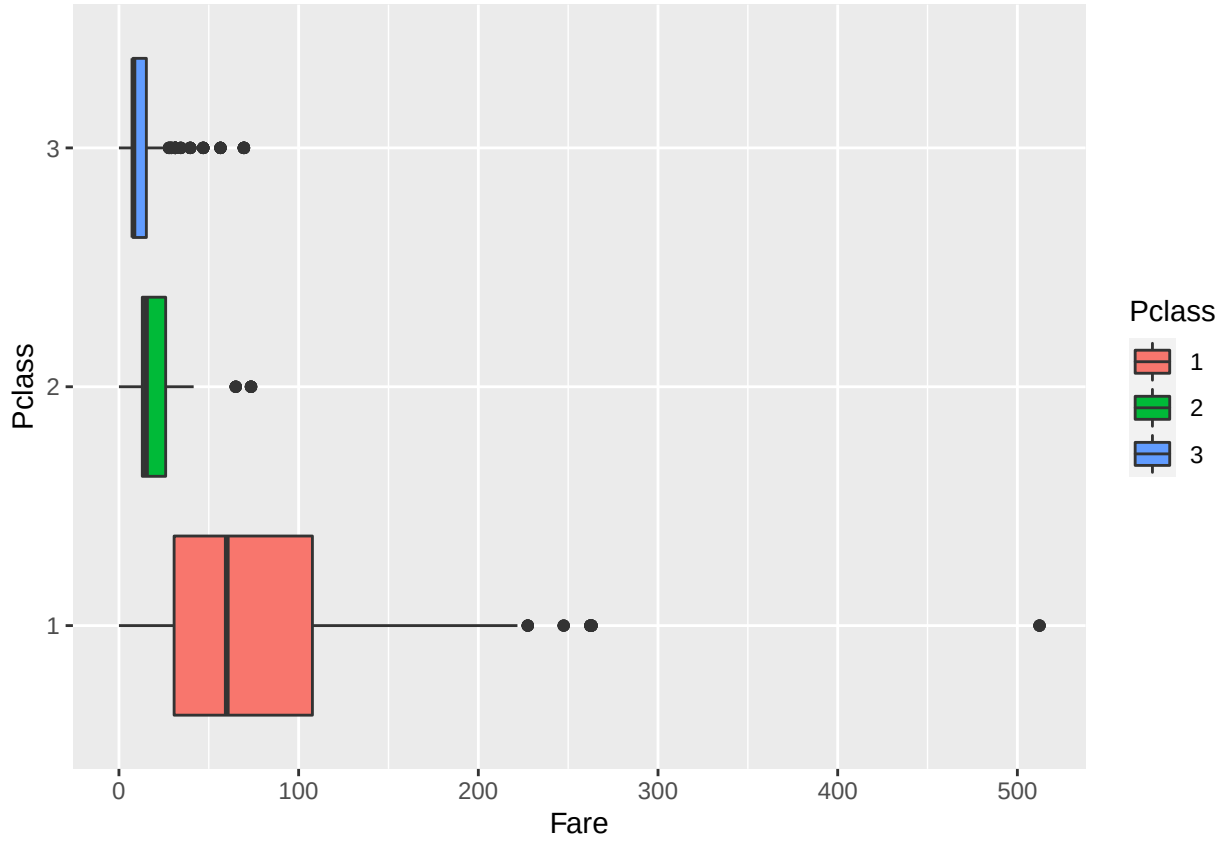
Vemos como hay posibles outliers en todas las variables, vamos a ver los histogramas, para identificar posibles grupos.

En la variable Age todos parecen están en un rango lógico de 0 a 80 años.

Con respecto a SipSp (pariente o esposas a bordo) y Parch (Parents/ children), parece lógico que alguien pueda sumar hasta 8 ente parientes y la esposa o entre hijos y padres (si tenemos en cuenta la época no es d extrañar familias con 8 hijos).

Vamos a ver en más detalle la variable Fare, parece que hay un valor muy alto en torno a 500. Vamos a dividir el boxplot por clase y ver como se distribuye.

```
ggplot(full, aes(x = Fare, y = Pclass, fill = Pclass)) + geom_boxplot( )
```



Vemos como ahora aunque haya algunos puntos fuera de los bigotes en 2 y 3 clase, no están tan poco muy alejados. El punto de 500 en primera clase podría ser el precio de la suite más lujosa del barco, así que por ahora lo vamos a dejar.

Creación de nuevas variables

Este data set presenta varias variables interesantes que, o bien combinándolas con otras, o bien mediante la extracción de información de ellas.

La primera nueva variable que vamos a crear va a ser a partir de Name. Esta variable representa el nombre apelli y título de cada pasajero (Miss, Mr., etc.), vamos a extraerlos primero y despues consolidarlos para tener solo los títulos "Other", "Mr", "Mr" y "Miss".

```
# Extracting the title form the Name column
full$Title <- as.factor(gsub('(.*, )|(\\..*)', '', full$Name))

# Showing different levels of the factor
levels(full$Title)
```

```
##  [1] "Capt"         "Col"          "Don"          "Dona"         "Dr"
##  [6] "Jonkheer"     "Lady"         "Major"        "Master"       "Miss"
## [11] "Mlle"         "Mme"          "Mr"           "Mrs"          "Ms"
## [16] "Rev"          "Sir"          "the Countess"
```

```r
# Consolidating names
full$Title <-
  revalue(
    full$Title,
    c(
      "Capt" = "Other",
      "Col" = "Other",
      "Don" = "Mr",
      "Dona" = "Mrs",
      "Dr" = "Other",
      "Jonkheer" = "Other",
      "Lady" = "Miss",
      "Major" = "Other",
      "Master" = "Mr",
      "Mlle" = "Miss",
      "Mme" = "Mrs",
      "Ms" = "Miss",
      "Rev" = "Other",
      "Sir" = "Other",
      "the Countess" = "Other"
    )
  )

# Cheking they make sense with the sex
table(full$Sex, full$Title)
```

```
##
##          Other  Mr Mrs Miss
##   female     2   0 199  265
##   male      24 819   0    0
```

La siguiente variable que vamos a crear es el tamaño de la familia con las que viaja cada pasajero a partir de las variable SibSp y Parch, además de sumarle 1 del propio pasajero

```r
# Family size, from SibSp and Parch + the passenger itself
full$FamilyS <- full$SibSp + full$Parch + 1
```

!!!!!!!!!!!corregir unknown!!!!!!!!!!!!!!!!!!!!!!!!!

También podemos extraer la planta en la que se hospeda cada pasajero sabiendo el número de habitación que tienen. Esto podría influir en un modelo que prediga la supervivencia.

```r
#Let's extract the deck from the cabin column
full$Deck <- as.factor(substr(full$Cabin, 0, 1))
```

Análisis estadístico

Visualizaciones

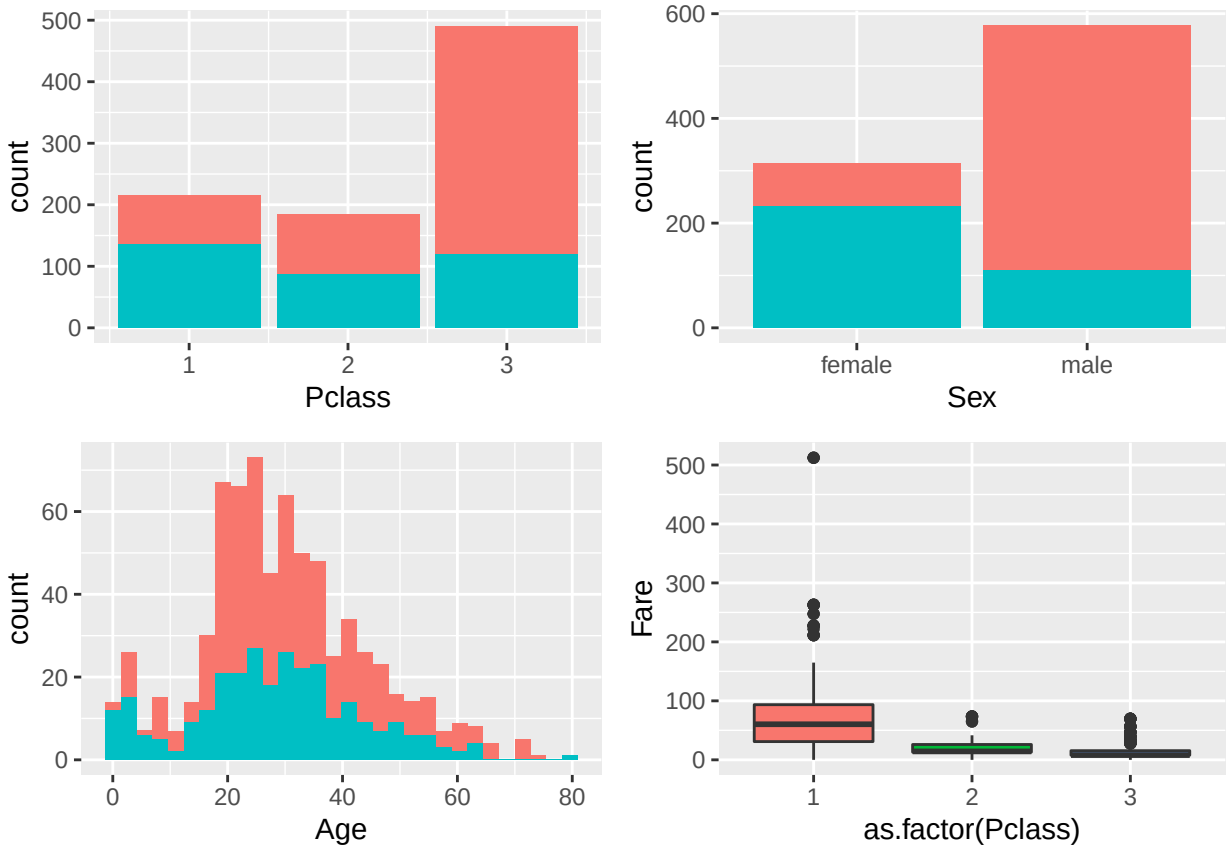
```r
#Plots

Class_surv_plot <-
  ggplot(train, aes(x = Pclass, fill = as.factor(Survived))) + geom_bar(show.legend = FALSE)

Sex_surv_plot <-
  ggplot(train, aes(x = Sex, fill = as.factor(Survived))) + geom_bar(show.legend = FALSE)

Age_surv_plot <-
  ggplot(train, aes(x = Age, fill = as.factor(Survived))) + geom_histogram(show.legend = FALSE)
Fare_plot <-
  ggplot(train, aes(
    x = as.factor(Pclass),
    y = Fare,
    fill = as.factor(Pclass)
  )) + geom_boxplot(show.legend = FALSE)


grid.arrange(Class_surv_plot, Sex_surv_plot, Age_surv_plot, Fare_plot)
```



Conclusiones