M2.851 - Tipología y ciclo de vida de los datos: Practico 2

Autor: Héctor Alejandro Castillo Jeria

Junio 2022

Contents

Introducción **1**

Competencias **2**

Objetivos **2**

Importancia de los análisis **2**

Inicio de Actividad. **3**

Comprensión de los datos. 3

Carga de librerias y fichero de datos. 3

Exploración de la base de datos de **test** 3

Exploración de la base de datos de **train** 6

Preparación de los datos. 8

Conclusiones previas. 13

Fase de Modelado. 13

Evaluación 13

Implantación o despliegue. 13

Introducción

Este documento contiene el desarrollo del Práctico número 2, en el cual, se elabora 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.

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.

Importancia de los análisis

A partir de este conjunto de datos se plantea la problemática de determinar qué variables son las que más influyen en la sobrevivencia o no de los pasajeros del RMS Titanic.

Además, se pretende crear modelos predictivos para emplearlos en el set de datos de test y determinar las probabilidades de que un pasajero pueda o no sobrevivir a la tragedia.

Inicio de Actividad.

Comprensión de los datos.

La muestra con la que trabajaremos corresponde a los datos de pasajeros del **RMS Titanic**, famoso transatlantico que sufre un accidente y se hunde en us viaje inaugural, el día 15 de Abril de 1912, donde fallecen 1502 de sus 2224 pasajeros y tripulantes.

Este juego de datos se encuentra en los archivo **train.csv** y **test.csv** ambos archivos, obtenido desde Kaggle **"Titanic - Machine Learning from Disaster"** https://www.kaggle.com/competitions/titanic

El archivo **train.csv** posee la información de los pasajeros, lo que permitirá entrenar nuestro modelo, para emplear luego el archivo **test.csv** que contiene información de pasajeros, para predecir si los pasajeros de esta muestra sobreviven al hundimiento del Titanic, de acuerdo a ciertos factores que obtendremos en el desarrollo de práctico.

Carga de librerias y fichero de datos.

Instalamos y cargamos las librerías ggplot2 y dplry.

```
# https://cran.r-project.org/web/packages/ggplot2/index.html
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
# https://cran.r-project.org/web/packages/dplyr/index.html
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')

if(!require(grid)){
    install.packages('grid', repos='http://cran.us.r-project.org')
    library(grid)
}
if(!require(gridExtra)){
    install.packages('gridExtra', repos='http://cran.us.r-project.org')
    library(gridExtra)
}
```

Cargamos los ficheros de datos *train.csv* y *test.csv*.

```
trainData <- read.csv('train.csv',stringsAsFactors = FALSE)
testData  <- read.csv('test.csv' ,stringsAsFactors = FALSE)
```

Exploración de la base de datos de test

Calcularemos las dimensiones de nuestra base de datos y analizaremos qué tipos de atributos tenemos. Mediante la función dim().

```
dim(testData)
```

```
## [1] 418  11
```

Como parte de la preparación de los datos, verificaremos si hay valores missing.

```
missing <- testData[is.na(testData),]
dim(missing)
```

```
## [1] 87 11
```

Verificamos la estructura del juego de datos principal.

```
str(testData)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis
##  $ Sex        : chr  "male" "female" "male" "male" ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

En nuestra carga de datos, vemos que tenemos 418 registros (filas) que se corresponden a las reseñas y 11 variables (columnas) que los caracterizan.

Revisamos la descripción de las variables contenidas al fichero y si los tipos de variable se corresponde al que hemos cargado:

Atributo	Descripción
PassengerId	Número Identificador del pasajero
Pclass	Clase Pasajero (1=primera Clase; 2=Segunda Clase; 3=Tercera Clase)
Name	Nombre
Sex	Sexo
Age	Edad
SibSp	Número de hermanos/cónyuges a bordo
Parch	Número de padres/hijos a bordo
Ticket	Número de Ticket
Fare	Tarifa de pasajero
Cabin	Cabina
Embarked	Puerto de embarque (C=Cherbourg; Q=Queenstown; S=Southampton)

Validación de la data test.

Obtendremos estadísticas básicas, para luego trabajar con los atributos que no poseen valores o se encuentran vacíos.

Estadísticas básicas

```
summary(testData)
```

```
##   PassengerId         Pclass          Name               Sex
##  Min.   : 892.0   Min.   :1.000   Length:418         Length:418
##  1st Qu.: 996.2   1st Qu.:1.000   Class :character   Class :character
##  Median :1100.5   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##       Age            SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##  NA's   :86
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:418         Length:418
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 35.627
##  3rd Qu.: 31.500
##  Max.   :512.329
##  NA's   :1
```

Verificación de valores vacíos.

```
colSums(is.na(testData))
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0          86           0
##       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           1           0           0
```

```
colSums(testData=="")
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##           0           0           0           0          NA           0
##       Parch      Ticket        Fare       Cabin    Embarked
##           0           0          NA         327           0
```

Las estadísticas obtenidas indican lo siguiente sobre la data:

El atributo **Age**, posee 86 datos vacíos, como este dato es relevante y puede influir en el resultado del análisis, el registro será eliminado.

El atributo **Fare**, posee 1 dato vacío, este dato no es relevante y se cargará el valor 0 por defecto.

El atributo **Cabin**, posee 327 datos vacíos, este dato no es relevante y se eliminará el atributo.

El atributo **Name**, este dato no es relevante y se eliminará el atributo.

El atributo **PassengerId**, este dato no es relevante y se eliminará el atributo.

Exploración de la base de datos de train

Calcularemos las dimensiones de nuestra base de datos y analizaremos qué tipos de atributos tenemos. Mediante la función dim(). Obtenemos que disponemos de 891 registros (filas) y 12 variables (columnas).

```
dim(trainData)
```

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

Como parte de la preparación de los datos, verificaremos si hay valores missing.

```
missing <- trainData[is.na(trainData),]
dim(missing)
```

```
## [1] 177  12
```

Verificamos la estructura del juego de datos principal.

```
str(trainData)
```

```
## '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" ...
```

En nuestra carga de datos, vemos que tenemos 891 registros (filas) que se corresponden a las reseñas y 12 variables (columnas) que los caracterizan.

Revisamos la descripción de las variables contenidas al fichero y si los tipos de variable se corresponde al que hemos cargado:

Atributo	Descripción
PassengerId	Número Identificador del pasajero
Survived	Sobreviviente (0=No; 1=Si)
Pclass	Clase Pasajero (1=primera Clase; 2=Segunda Clase; 3=Tercera Clase)
Name	Nombre
Sex	Sexo
Age	Edad
SibSp	Número de hermanos/cónyuges a bordo
Parch	Número de padres/hijos a bordo
Ticket	Número de Ticket
Fare	Tarifa de pasajero

Atributo	Descripción
Cabin	Cabina
Embarked	Puerto de embarque (C=Cherbourg; Q=Queenstown; S=Southampton)

Validación de la data train.

Obtendremos estadísticas básicas, para luego trabajar con los atributos que no poseen valores o se encuentran vacíos.

Estadísticas básicas

```
summary(trainData)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

Estadísticas de valores vacíos.

```
colSums(is.na(trainData))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

```
colSums(trainData=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

Las estadísticas obtenidas indican lo siguiente sobre la data:

El atributo **Age**, posee 177 datos vacíos, como este dato es relevante y puede influir en el resultado del análisis, el registro será eliminado.

El atributo **Cabin**, posee 687 datos vacíos, este dato no es relevante y se eliminará el atributo.

El atributo **Embarked**, posee 2 datos vacíos, este dato no es relevante y se cargará un valor por defecto.

El atributo **Name**, este dato no es relevante y se eliminará el atributo.

El atributo **PassengerId**, este dato no es relevante y se eliminará el atributo.

Preparación de los datos.

En este paso realizaremos la normalización de algunos atributos, la clusterización de otros, la eliminación de datos no relevantes, la creación de nuevos atributos, basicamente en este punto tomamos el set de datos, para luego aplicar tecnicas de normalización, completar atributos inexistentes, agregar valores por defecto, incorporar un atributo empleando formulas estadisticas como la media. Como ejemplo. En este punto, a los atributos de tipo texto, le cargaremos el valor por defecto "desconocido", cuando el atributo es nulo o vacío.

Reemplazo de valores nulos o vacíos.

```
testData$Age[is.na(testData$Age)] <- mean(testData$Age[is.na(testData$Age)==FALSE])
testData$Fare[is.na(testData$Fare)] <- 0
```

Eliminación de registros con valores nulos o vacíos.

```
testData <- subset(testData, !is.na(testData$Age))
trainData <- subset(trainData, !is.na(trainData$Age))

trainData <- subset(trainData, !is.na(trainData$Embarked))
trainData <- subset(trainData, trainData$Embarked!="")
```

Eliminación de atributos.

```
testData  <- subset( testData, select = -c(PassengerId, Cabin, Name, Ticket ) )
trainData <- subset( trainData, select = -c(PassengerId, Cabin, Name, Ticket ) )
```

Verificación de corrección de valores vacíos **test**.

```
colSums(is.na(testData))
```

```
##   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        0        0        0        0        0        0        0
```

```
colSums(testData=="")
```

```
##   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        0        0        0        0        0        0        0
```

Verificación de corrección de valores vacíos **train**.

```
colSums(is.na(trainData))
```

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

```
colSums(trainData=="")
```

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

Creación de nuevos atributos.

Agregaremos un nuevo campo a los datos. Este campo contendrá el valor de la edad discretitzada con un método simple de intervalos de igual amplitud.

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.42   20.00   28.00   29.64   38.00   80.00
```

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.17   23.00   30.27   30.27   35.75   76.00
```

Discretizamos con intervalos.

```
trainData["Rango_Age"] <- cut(trainData$Age, breaks = c(0,10,20,30,40,50,60,70,80,110), labels = c("0-9
testData["Rango_Age"]  <- cut(testData$Age , breaks = c(0,10,20,30,40,50,60,70,80,100), labels = c("0-9

trainData["Sobreviviente"] <-""
trainData$Sobreviviente[trainData$Survived=="1"] <- "SI"
trainData$Sobreviviente[trainData$Survived=="0"] <- "NO"

testData  <- subset( testData , select = -c( Age ) )
trainData <- subset( trainData, select = -c( Age, Survived ) )
```

Nuevo Formato de la data

Nuesta data **test**, ahora presenta 418 registros (filas) que se corresponden a las reseñas y 7 variables (columnas) que los caracterizan.

Atributo	Descripción
Pclass	Clase Pasajero (1=primera Clase; 2=Segunda Clase; 3=Tercera Clase)
Sex	Sexo
SibSp	Número de hermanos/cónyuges a bordo
Parch	Número de padres/hijos a bordo
Fare	Tarifa de pasajero
Embarked	Puerto de embarque (C=Cherbourg; Q=Queenstown; S=Southampton)
Rango_Age	Rango de Edad

Nuesta data **train**, ahora presenta 418 registros (filas) que se corresponden a las reseñas y 8 variables (columnas) que los caracterizan.

Atributo	Descripción
Pclass	Clase Pasajero (1=primera Clase; 2=Segunda Clase; 3=Tercera Clase)
Sex	Sexo
SibSp	Número de hermanos/cónyuges a bordo
Parch	Número de padres/hijos a bordo
Fare	Tarifa de pasajero
Embarked	Puerto de embarque (C=Cherbourg; Q=Queenstown; S=Southampton)
Rango_Age	Rango de Edad
Sobreviviente	Indica si el pasajero sobrevició o no al accidente SI-NO

Observamos los datos discretizados.

```
head(trainData)
```

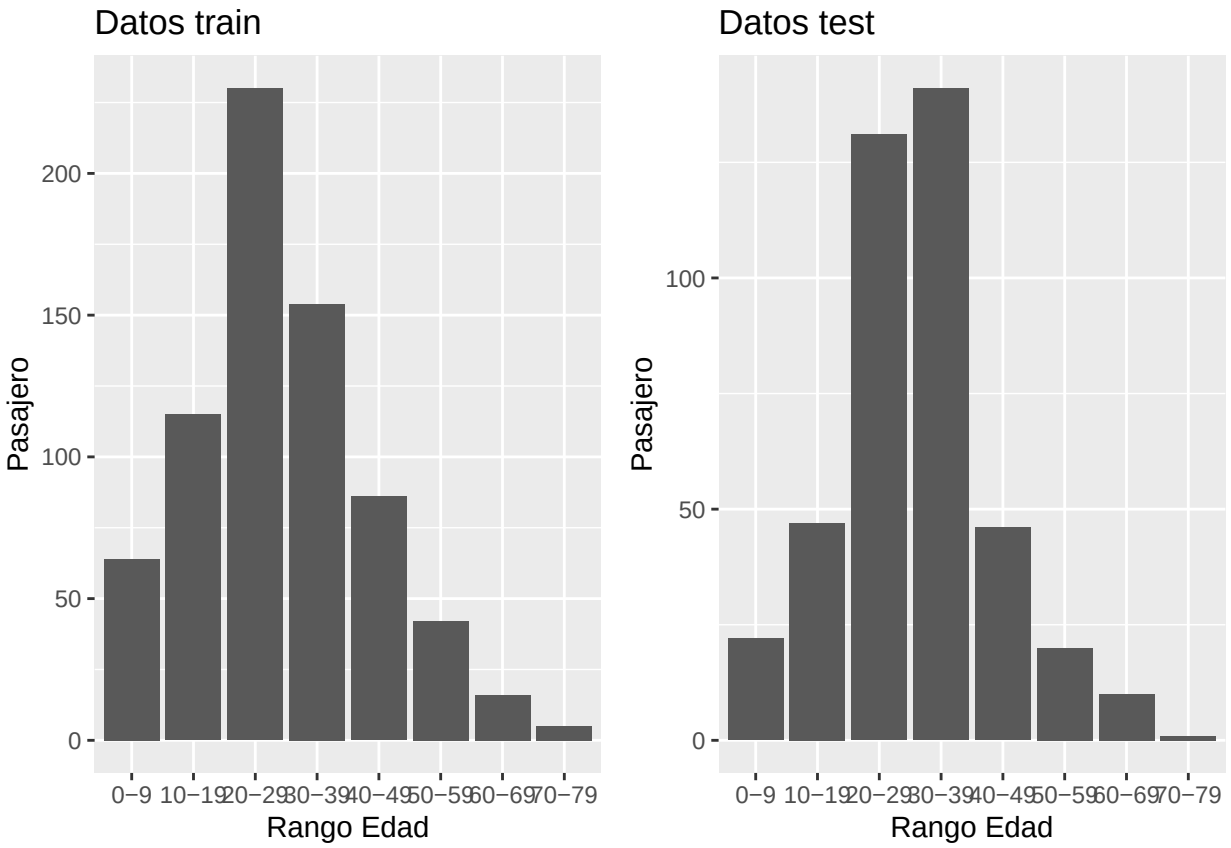
```
##   Pclass    Sex SibSp Parch    Fare Embarked Rango_Age Sobreviviente
## 1      3   male     1     0  7.2500        S     20-29            NO
## 2      1 female     1     0 71.2833        C     30-39            SI
## 3      3 female     0     0  7.9250        S     20-29            SI
## 4      1 female     1     0 53.1000        S     30-39            SI
## 5      3   male     0     0  8.0500        S     30-39            NO
## 7      1   male     0     0 51.8625        S     50-59            NO
```

```
head(testData)
```

```
##   Pclass    Sex SibSp Parch    Fare Embarked Rango_Age
## 1      3   male     0     0  7.8292        Q     30-39
## 2      3 female     1     0  7.0000        S     40-49
## 3      2   male     0     0  9.6875        Q     60-69
## 4      3   male     0     0  8.6625        S     20-29
## 5      3 female     1     1 12.2875        S     20-29
## 6      3   male     0     0  9.2250        S     10-19
```

Vemos como los datos se agrupan por segmento de edad.

```
grid.newpage()

#plotTrbyAge <- ggplot(trainData$Rango_Age,main="train. Número de pasajeros por grupos de #edad",xlab="

#plotTebyAge <-ggplot(testData$Rango_Age,main="test. Número de pasajeros por grupos de #edad",xlab="Edad

plotTrbyAge <- ggplot(trainData,aes(Rango_Age))+geom_bar() +labs(x="Rango Edad", y="Pasajero")+ guides(

plotTebyAge <- ggplot(testData,aes(Rango_Age))+geom_bar() +labs(x="Rango Edad", y="Pasajero")+ guides(fi


grid.arrange(plotTrbyAge,plotTebyAge,ncol=2)
```



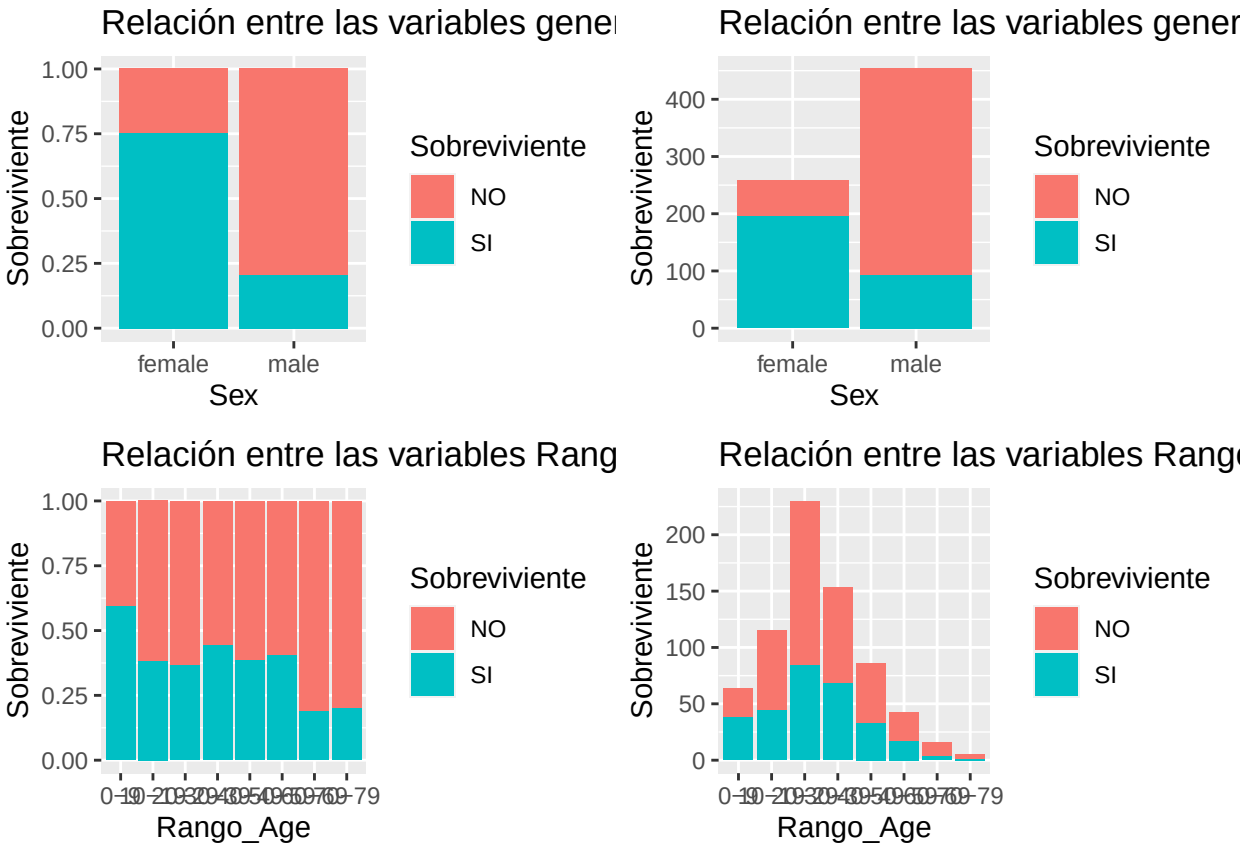
Procesos de análisis visuales del juego de datos

Nos proponemos analizar las relaciones entre las diferentes variables del juego de datos para ver si se relacionan y como.

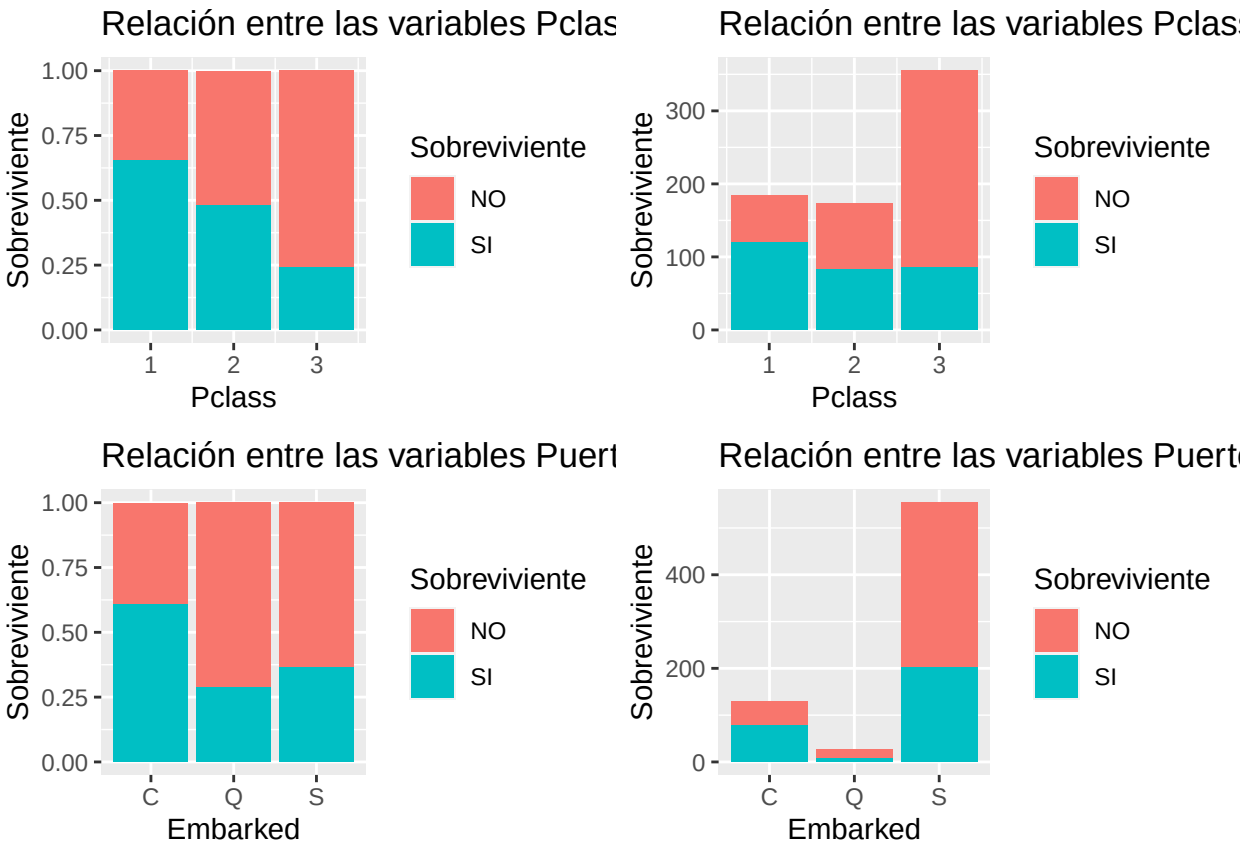
Visualizamos la relaciones entre variable

```
plotTrbySex1 <- ggplot(data=trainData[1:filasTrain,],aes(x=Sex,fill=Sobreviviente))+geom_bar(position="


plotTrbySex2 <- ggplot(data=trainData[1:filasTrain,],aes(x=Sex,fill=Sobreviviente))+geom_bar()+ylab("Sob
```

```
plotTrbyAge1 <- ggplot(data=trainData[1:filasTrain,],aes(x=Rango_Age,fill=Sobreviviente))+geom_bar(posi
plotTrbyAge2 <- ggplot(data=trainData[1:filasTrain,],aes(x=Rango_Age,fill=Sobreviviente))+geom_bar()+yl
plotTrbyCla <- ggplot(data=trainData[1:filasTrain,],aes(x=Pclass,fill=Sobreviviente))+geom_bar(position
plotTrbyEmb <- ggplot(data=trainData[1:filasTrain,],aes(x=Embarked,fill=Sobreviviente))+geom_bar()+ylab
grid.arrange(plotTrbySex1,plotTrbySex2,plotTrbyAge1,plotTrbyAge2,ncol=2)
```




```
#grid.newpage()
plotTrbyCla1 <- ggplot(data=trainData[1:filasTrain,],aes(x=Pclass,fill=Sobreviviente))+geom_bar(positio
plotTrbyCla2 <- ggplot(data=trainData[1:filasTrain,],aes(x=Pclass,fill=Sobreviviente))+geom_bar()+ylab(
plotTrbyEmb1 <- ggplot(data=trainData[1:filasTrain,],aes(x=Embarked,fill=Sobreviviente))+geom_bar(posit
plotTrbyEmb2 <- ggplot(data=trainData[1:filasTrain,],aes(x=Embarked,fill=Sobreviviente))+geom_bar()+ylab

grid.arrange(plotTrbyCla1,plotTrbyCla2,plotTrbyEmb1,plotTrbyEmb2,ncol=2)
```



Conclusiones previas.

De acuerdo a los graficos obtenidos, Existe una grán posibilidad de sobrevivir, si el pasajeto, es mujer, es menor de 60 años, tiene un ticket de primera clase y embarca en el puerto de Cherbourg.

Fase de Modelado.

En esta fase se construirán y evaluarán varios modelos, se probarán algoritmos y técnicas hasta encontrar un modelo adecuado.

Evaluación

En esta fase se evaluará si el modelo, se adapta a lo esperado.

Implantación o despliegue.

Esta fase es donde se entrega un objeto, con el cual, el cliente ya puede obtener resultados.
