Universitat Oberta de Catalunya

Estudios de Informática, Multimedia y Telecomunicaciones

Práctica 2: Limpieza y validación de los datos

Autores: Mejía Quintero Dayana, Peterson Christopher

Junio 2022

CONTENIDO

1. COMPETENCIAS DE LA PRÁCTICA
2. OBJETIVOS DE LA PRÁCTICA
3. DESCRIPCIÓN DEL DEL DATASET
4. IMPORTANCIA Y OBJETIVOS DEL ANÁLISIS DEL DATASET.
5. INTEGRACIÓN Y SELECCIÓN DE LOS DATOS A ANALIZAR.
6. PROCESO DE LIMPIEZA DE LOS DATOS.

    6.1 Eliminación de valores nulos y vacíos

    6.2 Identificación y gestión de valores extremos u outliers

7. ANÁLISIS DE LOS DATOS.

    7.1 Selección de grupos de datos a analizar.

    7.2 Comprobación de la normalidad y homogeneidad de la varianza

8. APLICACIÓN DE PRUEBAS ESTADISTICAS
9. REPRESENTACIÓN GRÁFICA DE LOS RESULTADOS A PARTIR DE TABLAS Y GRÁFICAS
10. RESOLUCIÓN DE PROBLEMAS Y CONCLUSIONES
11. EXPORTACIÓN DEL CÓDIGO
12. BIBLIOGRAFÍA

1. COMPETENCIAS DE LA PRACTICA

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.

2. OBJETIVOS DE LA PRÁCTICA

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.

3. DESCRIPCIÓN DEL DATASET

El dataset escogido representa a las personas que embarcaron en el titanic, el cual naufragó en el Océano del Atlántico Norte el 15 de abril de 1912 al ser impactado por un iceberg. Dado de que vamos a usar dos datasets: Train y test, vamos a unificarlos en un solo dataset para proceder a la limpieza de datos y posterior análisis. El fichero contiene: 1309 observaciones de 12 variables que anteriormente se mencionaron en la descripción del dataset.

El dataset final contiene las siguientes variables:

• PassengerID: ID del pasajero.

• Survival: Sobreviviente. Está compuesto con 0=No, 1= Yes.

• Name: Nombre de los pasajeros

• Pclass: Clase del tiquete. Está compuesto con 1 = 1era, 2= 2da, 3 = 3ra. En donde funciona como

proxy del estatus socioeconómico. 1era = Clase alta, 2da= Clase Media, 3ra= Clase Baja.

• Sex: Sexo.

• Age: Edad en años. Donde Age es fraccional si es menor a 1. Si la edad es estimada, entonces tiene forma de xx.5.

• Sibsp: # de hermanos / pareja dentro del titanic. El dataset define a las relaciones familiares como Sibling= hermano, hermana, hermanastro, hermanastra. Pareja = esposa, esposo (amantes y prometidos fueron ignorados).

• Parch: # de padres/hijos dentro del titanic. El dataset define las relaciones familiares como: Parent = madre, padre. Child = Hija, hijo, hijastra, hijastro. Algunos niños viajaron solo con su niñera, por lo tanto, parch= 0 para ellos.

• Ticket: Número del tiquete.

• Fare: Tarifa del pasajero.

• Cabin: Número de la cabina.

• Embarked: Puerto de embarcación. Está compuesto por C= Cherbourg, Q = Queenstown, S = Southampton.

4. IMPORTANCIA Y OBJETIVOS DEL ANÁLISIS DEL DATASET.

Hemos escogido el dataset relacionado con las personas que embarcaron el titanic que se encuentra en la página kaggle: https://www.kaggle.com/competitions/titanic/data en la cual separa los datos en dos datasets: test y train. Donde train.csv contiene los detalles de un subset de los pasajeros a bordo del titanic los cuales son 891 en total y en donde se revelará si sobreviven o no. El test.csv contiene información similar, pero con ella debemos predecir cuál de estas condiciones sucede.

El objetivo de esta práctica es predecir si los pasajeros a bordo sobreviven o no y también encontrar si ciertas variables como la “pclass” que identifica la clase del tiquete influenciaron en la sobrevivencia de los pasajeros y si otras variables entraron como a influenciar de forma más impredecible.

5. INTEGRACIÓN Y SELECCIÓN DE LOS DATOS A ANALIZAR.

Es importante escoger las variables que consideramos importantes que nos ayudarán en el proceso del análisis del dataset para posteriormente llegar a los objetivos planteados en esta práctica. Dichas variables deben contener la información más relevante que nos ayude llegar a dicho paso y resolver el problema planteado. Al observar el dataset y ver cómo se comporta las variables podemos reducir la dimensionalidad y también reducir el dataset, eliminando las variables que consideramos que no ayudan a la resolución.
En nuestro caso, se eliminará las siguientes variables:

• PassengerID: ID del pasajero.
• Name: Nombre de los pasajeros
• Ticket: Número del tiquete.
• Cabin: Número de la cabina. Instalamos los paquetes Instalamos y cargamos las librarías requeridas.

Dado de que vamos a usar dos datasets: Train y test, vamos a unificarlos en un solo dataset para proceder luego a la limpieza de datos y posterior análisis. El fichero unificado contiene: 1309 observaciones de 12 variables que anteriormente se mencionaron en la descripción del dataset. Por ahora mantendremos todas las variables para su observación total.

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

# Creamos un nuevo dataset con ambos archivos como se había mencionado anteriormente.
df <- bind_rows(train,test)
len_train=dim(train)[1]
```

Hacemos una rápida observación del dataset donde vemos el número de variables y el número de observaciones que ya se ha mencionado. También se puede ver las características de las variables del dataset.

```
str(df)
```

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

Observamos las estadísticas principales de las variables:

```
summary(df)
```

```
##   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
```

Eliminamos variables que no vamos a utilizar:

```
#Eliminamos PassengerId, Name, Ticket, Cabin,
dfaux<-df[,c("Survived","Pclass","Sex","Age","Fare","SibSp","Parch", "Embarked")]
```

Quedaría así:

```
summary(dfaux);
```

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

---

6. PROCESO DE LIMPIEZA DE LOS DATOS.

Ahora procederemos a la limpieza de los datos analizando los valores vacíos, nulos y los valores extremos u outliers.

6.1 Eliminación de valores nulos y vacíos.

Para comenzar en la limpieza de datos, observamos las variables que contienen valores vacíos la cual la razón suele ser porque no se llegó a registrar la información.

```
# Valores vacios
colSums(is.na(dfaux))
```

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

```
colSums(dfaux=="")
```

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

Como podemos ver, las variables “Age” y la variable “Embarked” contiene valores vacíos. Existen diferentes métodos para poder solucionar este problema. En nuestro caso, utilizaremos el método de reemplazo con la media para dichos valores. En el caso de “Embarked” utilizaremos que vamos a reemplazar los valores vacíos con la primera opción que es “S”.

6.2 Identificación y gestión de valores extremos u outliers.

Los valores extremos u outliers son aquellos valores que se encuentran alejados del resto de observaciones y pueden llegar a ser valores tanto muy pequeños o grandes. Para su análisis es necesario también comprender las razones del porque se pueden generar este tipo de valores para no eliminarlos y sesgar el análisis afectando el modelo. Utilizaremos la herramienta de boxplot.stats para identificar dichos valores. Primero vamos a convertir las variables de "Survived", "Pclass", "Sex", "Embarked" a factores dado de que estos valores toman valores finitos:

```
# Convertimos los datos de Survived, Pclass, Sex, Embarked a factores

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

Veamos las variables con los valores extremos usando boxplot.stats.

Comenzamos para Edad "Age".

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

```
##   [1]  2.00 58.00 55.00  2.00 66.00 65.00  0.83 59.00 71.00 70.50  2.00 55.50
##  [13]  1.00 61.00  1.00 56.00  1.00 58.00  2.00 59.00 62.00 58.00 63.00 65.00
##  [25]  2.00  0.92 61.00  2.00 60.00  1.00  1.00 64.00 65.00 56.00  0.75  2.00
##  [37] 63.00 58.00 55.00 71.00  2.00 64.00 62.00 62.00 60.00 61.00 57.00 80.00
##  [49]  2.00  0.75 56.00 58.00 70.00 60.00 60.00 70.00  0.67 57.00  1.00  0.42
##  [61]  2.00  1.00 62.00  0.83 74.00 56.00 62.00 63.00 55.00 60.00 60.00 55.00
##  [73] 67.00  2.00 76.00 63.00  1.00 61.00 60.50 64.00 61.00  0.33 60.00 57.00
##  [85] 64.00 55.00  0.92  1.00  0.75  2.00  1.00 64.00  0.83 55.00 55.00 57.00
##  [97] 58.00  0.17 59.00 55.00 57.00
```

La variable "Pclass":

```
boxplot.stats(dfaux$Pclass)$out
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors

## factor(0)
## Levels: 1 2 3
```

La variable "sex":

```
boxplot.stats(dfaux$Sex)$out
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors

## factor(0)
## Levels: female male
```

La variable “Embarked”:

```
boxplot.stats(dfaux$Embarked)$out
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors

## factor(0)
## Levels: C Q S
```

La variable “Fare”:

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

```
##   [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000
##   [9] 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000  66.6000  69.5500
##  [17]  69.5500 146.5208  69.5500 113.2750  76.2917  90.0000  83.4750  90.0000
##  [25]  79.2000  86.5000 512.3292  79.6500 153.4625 135.6333  77.9583  78.8500
##  [33]  91.0792 151.5500 247.5208 151.5500 110.8833 108.9000  83.1583 262.3750
##  [41] 164.8667 134.5000  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000
##  [49] 263.0000  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  91.0792
##  [65]  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000 221.7792 106.4250
##  [73]  71.0000 106.4250 110.8833 227.5250  79.6500 110.8833  79.6500  79.2000
##  [81]  78.2667 153.4625  77.9583  69.3000  76.7292  73.5000 113.2750 133.6500
##  [89]  73.5000 512.3292  76.7292 211.3375 110.8833 227.5250 151.5500 227.5250
##  [97] 211.3375 512.3292  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583
## [105] 211.3375  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 263.0000
## [121] 262.3750 262.3750 263.0000 211.5000 211.5000 221.7792  78.8500 221.7792
## [129]  75.2417 151.5500 262.3750  83.1583 221.7792  83.1583  83.1583 247.5208
## [137]  69.5500 134.5000 227.5250  73.5000 164.8667 211.5000  71.2833  75.2500
## [145] 106.4250 134.5000 136.7792  75.2417 136.7792  82.2667  81.8583 151.5500
## [153]  93.5000 135.6333 146.5208 211.3375  79.2000  69.5500 512.3292  73.5000
## [161]  69.5500  69.5500 134.5000  81.8583 262.3750  93.5000  79.2000 164.8667
## [169] 211.5000  90.0000 108.9000
```

La variable “Survived”:

```
boxplot.stats(dfaux$Survived)$out
```

```
## Warning in Ops.factor(x[floor(d)], x[ceiling(d)]): '+' not meaningful for
## factors

## factor(0)
## Levels: 0 1
```

La variable “SibSp”:

```
boxplot.stats(dfaux$SibSp)$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 5 4 3
## [39] 4 8 4 3 4 8 4 8 3 4 5 3 4 8 4 8 4 3 3
```

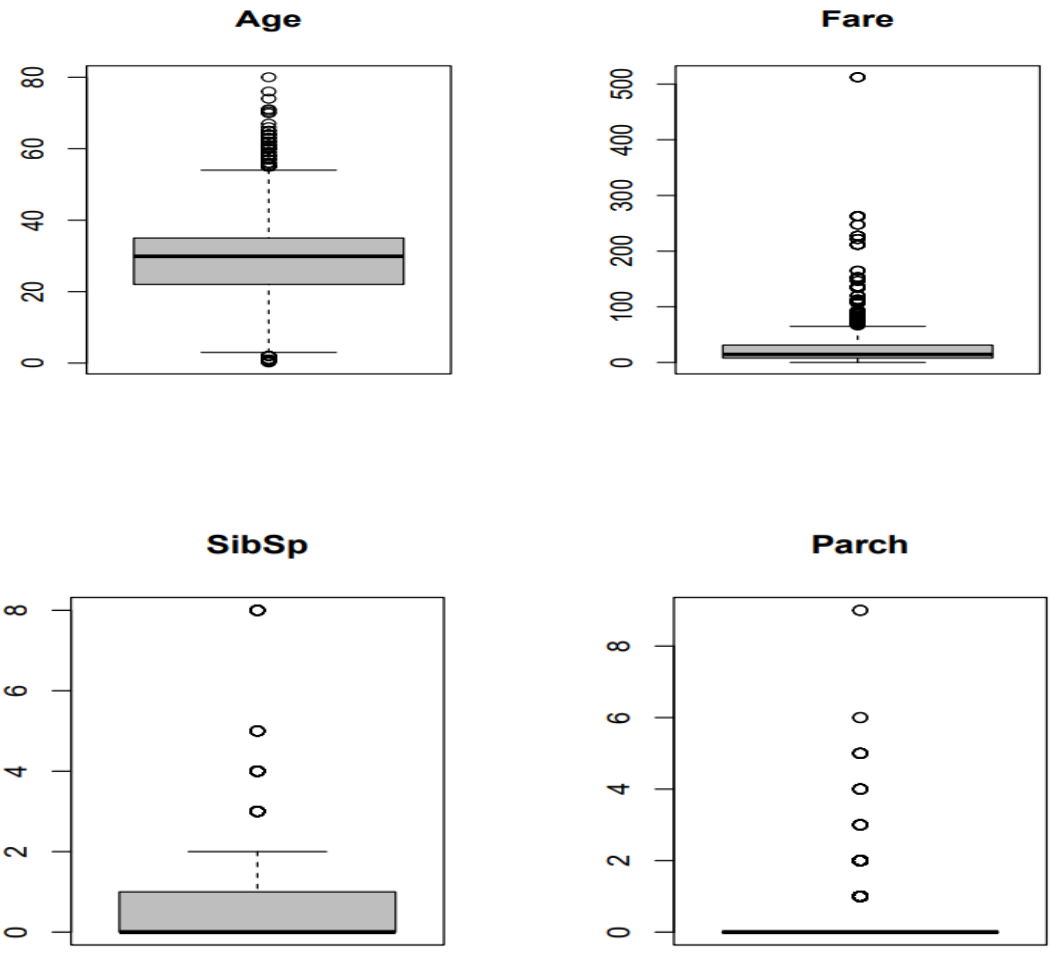
La variable “Parch”:

```
boxplot.stats(dfaux$Parch)$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 2 1
##  [38] 1 2 1 4 1 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 1 2 1 2
##  [75] 2 1 1 2 1 1 2 1 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2 2 3 4 1 2 1
## [112] 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 2 1 1 2 5 2 1 1
## [149] 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 1 2 1 2 3 1 2 1 2 2
## [186] 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 1 5 2 1 1 1 1 3 1 2 2 1
## [223] 2 1 2 1 2 4 1 1 2 1 1 1 4 6 2 3 1 1 2 2 2 1 1 2 5 2 3 2 1 1 1 2 1 2 2 2 1
## [260] 2 1 1 2 1 2 1 2 1 2 2 1 1 1 1 1 2 1 1 2 1 1 1 2 1 2 9 1 1 1 2 2 2 1 9 1 1
## [297] 2 2 1 1 2 1 1 1 1 1 1
```

Utilizando un diagrama de caja nos dará lo siguiente en forma gráfica:

```
outliers <- function(dfaux) {
    par(mfrow=c(1,2))
    for(i in 1:ncol(dfaux)) {
        if (is.numeric(dfaux[,i])){
            boxplot(dfaux[,i], main = colnames(dfaux)[i], width = 100, col="gray")
        }
    }

    max(dfaux$Age, na.rm = TRUE)
    min(dfaux$Age, na.rm = TRUE)
    fivenum(dfaux$Age)

    max(dfaux$Fare, na.rm = TRUE)
    min(dfaux$Fare, na.rm = TRUE)
    fivenum(dfaux$Fare)
}
outliers(dfaux)
```




```
## [1]   0.0000   7.8958  14.4542  31.2750 512.3292
```

De lo anterior podemos observar que la variable “age” tiene valores extremos, pero hay que tener en cuenta que las edades comprendidas entre 60 años y 80 años son

normales, también que una persona tenga 0.92 años ya que representa a un bebe. También en el caso de “Fare”, existen también valores extremos, pero de acuerdo con la cabina comprada puede ser normal dicho precio gastado por los pasajeros. Esto en resumen nos lleva a que no quitaremos los valores extremos porque podemos asegurarnos que son válidos dado las condiciones de las variables.

7. ANÁLISIS DE LOS DATOS.

Analizaremos el dataset ya limpio para observar cómo se comportan las variables.

7.1 Selección de grupos de datos a analizar.

Agrupamos los datos en grupos:
Agrupamos los datos que se quieren comparar.

```
dfaux.third_class <- dfaux[dfaux$Pclass == 3,]

print(paste("First_class: ", nrow(dfaux.first_class)))
```

```
## [1] "First_class:  323"
```

```
print(paste("Second_class: ", nrow(dfaux.second_class)))
```

```
## [1] "Second_class:  277"
```

```
print(paste("Third_class: ", nrow(dfaux.third_class)))
```

```
## [1] "Third_class:  709"
```

```
# Por genero
dfaux.male <- dfaux[dfaux$Sex == "male",]
dfaux.female <- dfaux[dfaux$Sex == "female",]

print(paste("Male: ", nrow(dfaux.male)))
```

```
## [1] "Male:  843"
```

```
print(paste("Female: ", nrow(dfaux.female)))
```

```
## [1] "Female:  466"
```

```
# Por Embarque
dfaux.C <- dfaux[dfaux$Embarked == "C",]
dfaux.Q <- dfaux[dfaux$Embarked == "Q",]
dfaux.S <- dfaux[dfaux$Embarked == "S",]

print(paste("Cherbourg: ", nrow(dfaux.C)))
```

```
print(paste("Queenstown: ", nrow(dfaux.Q)))
```

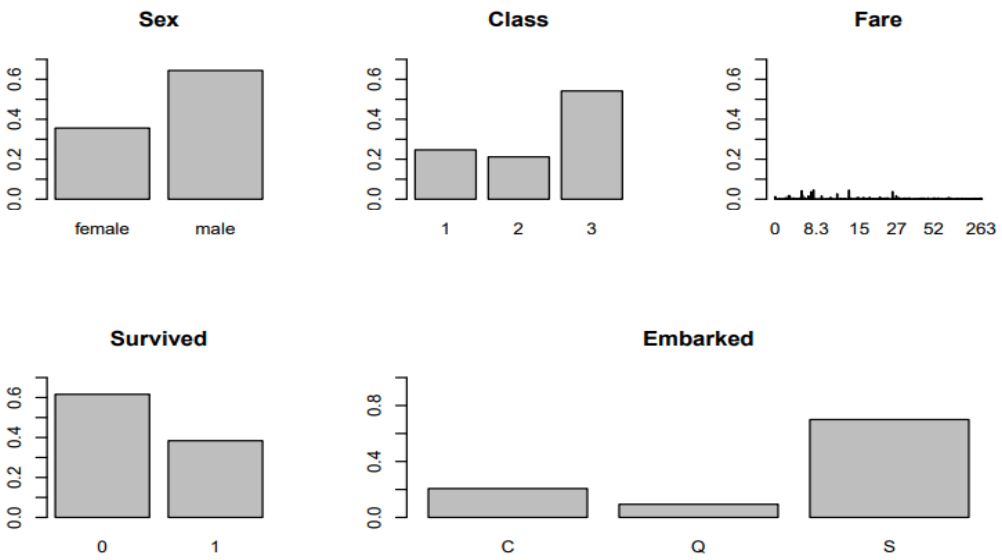
```
## [1] "Queenstown:  123"
```

```
print(paste("Southampton: ", nrow(dfaux.S)))
```

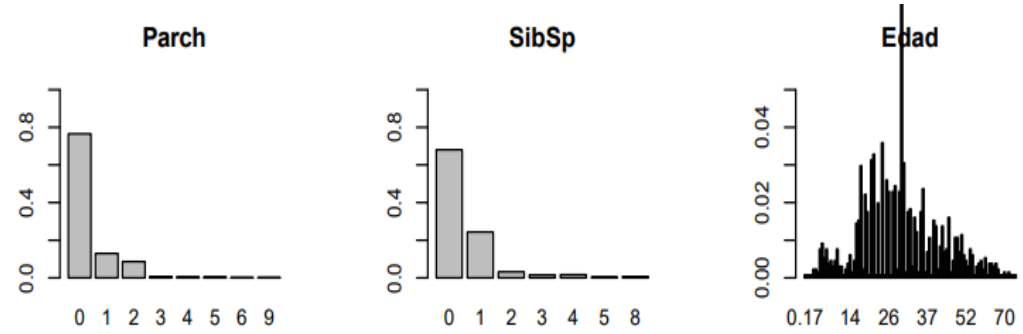
```
## [1] "Southampton:  916"
```

Primero vamos a graficar las frecuencias usando un barplot con todas las variables y observar cómo se comportan:

```
# Gráfica de las Frecuencias de cada una de las variables del dataset
dataaux<-layout(matrix(c(1,2,3,4,5,5), 2, 3, byrow=TRUE),respect=TRUE);
barplot(prop.table(table(dfaux$Sex)),ylim=c(0,0.7), main="Sex");
barplot(prop.table(table(dfaux$Pclass)),ylim=c(0,0.7), main="Class");
barplot(prop.table(table(dfaux$Fare)),ylim=c(0,0.7), main="Fare");
barplot(prop.table(table(dfaux$Survived)),ylim=c(0,0.7), main="Survived");
barplot(prop.table(table(dfaux$Embarked)),ylim=c(0,1), main="Embarked");
```

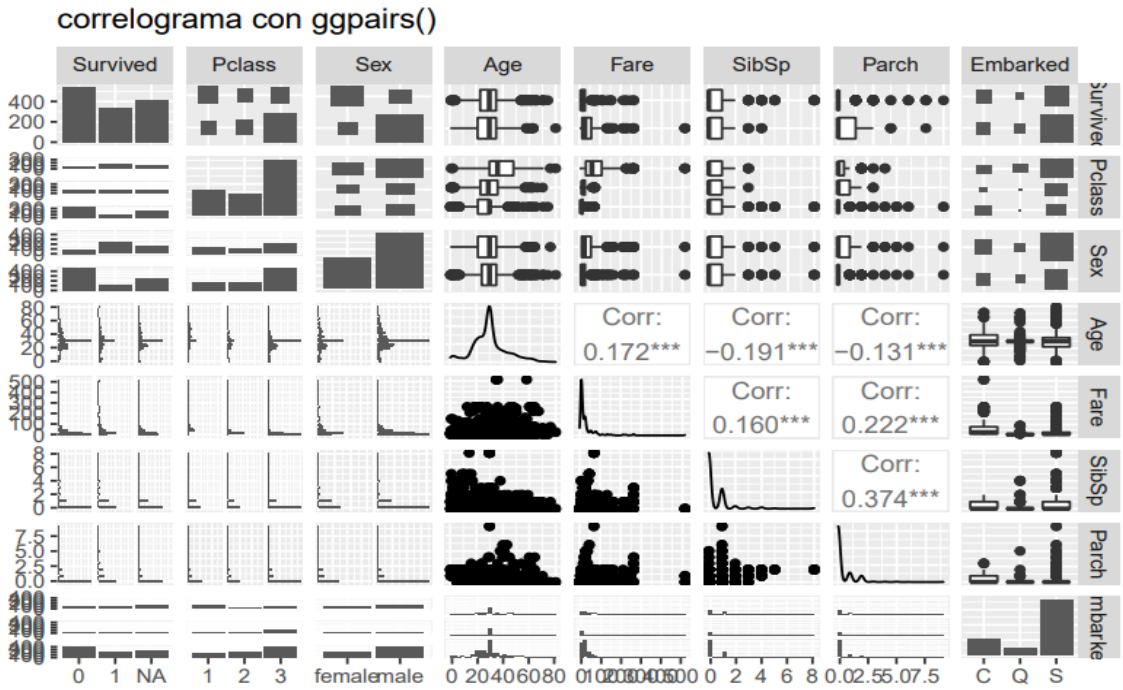



```
barplot(prop.table(table(dfaux$Parch)),ylim=c(0,1), main="Parch");
barplot(prop.table(table(dfaux$SibSp)),ylim=c(0,1), main="SibSp");
barplot(prop.table(table(dfaux$Age)),ylim=c(0,0.05), main="Edad");
```



Ahora vamos a observamos las correlaciones de ellas usando varias gráficas como el scatterplot, las distribuciones y el coeficiente de correlación.

```
# Vemos las correlaciones (usando scatterplots), distribuciones e imprimimos el coeficiente de correlac
ggpairs(dfaux, title="correlograma con ggpairs()")
```



7.2 Comprobación de la normalidad y homogeneidad de la varianza.

Ya que hemos anteriormente agrupado las variables, podemos realizar la comprobación de la homogeneidad con la función de Fligner- Killeen la cual es un test no paramétrico para la homogeneidad de un grupo de varianzas basada en rangos.

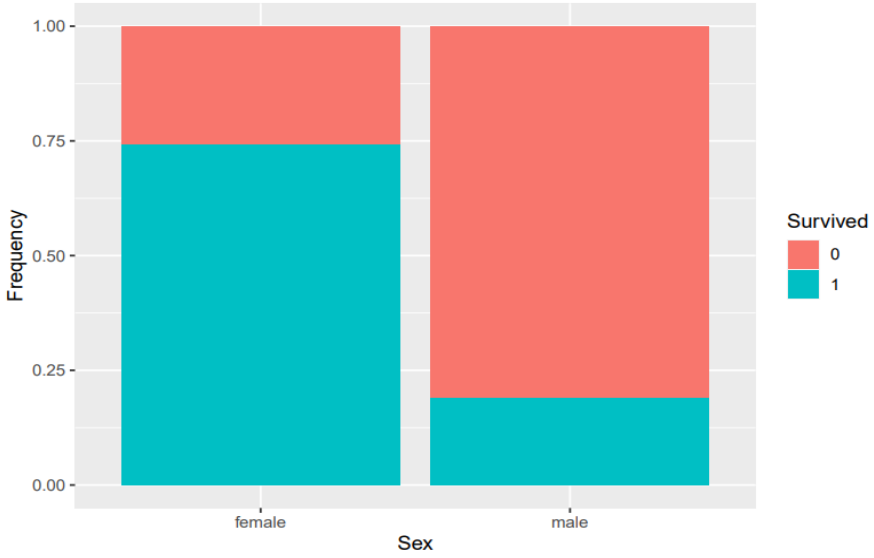
```
varianza <- fligner.test(dfaux);
varianza;
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  dfaux
## Fligner-Killeen:med chi-squared = 5407.1, df = 7, p-value < 2.2e-16
```

Esto nos trae como conclusión que p-value < 2.2e-16 lo que las varianzas de las variables son diferentes.
Ahora, en el caso de la normalidad, vamos a hacer el análisis con dos variables, es decir, survived, y otra variable.

Relación entre Sex y survival: Podemos observar que alrededor del 75% de las mujeres fueron sobrevivientes del naufragio mientras que hay un porcentaje menor de sobrevivientes hombres con menos del 23%.

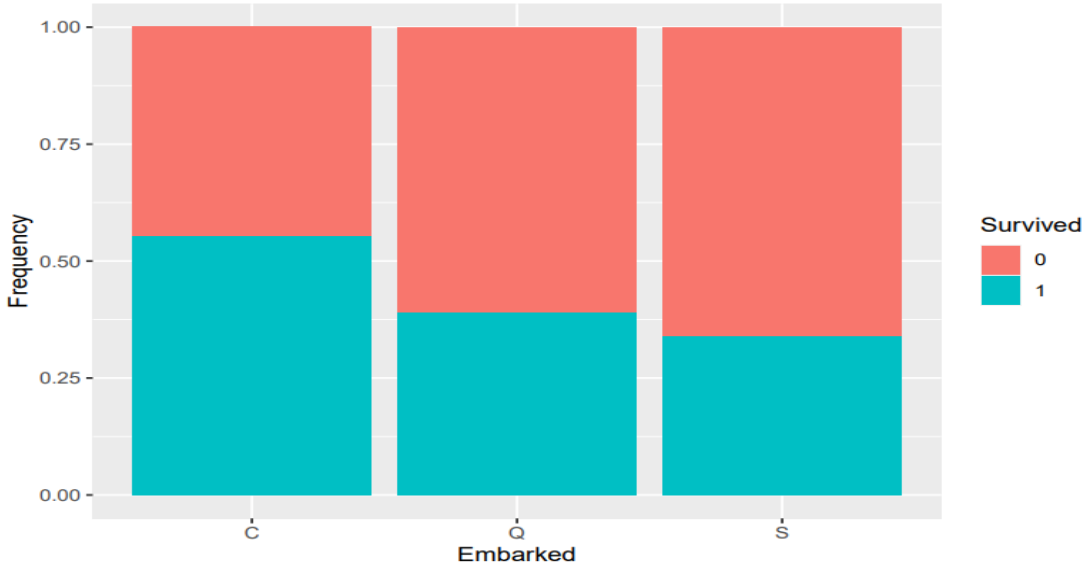
```
# Comprobación de la normalidad.
#Vamos a ver la relación entre sex y survival.
ggplot(data=dfaux[1:len_train,],aes(x=Sex,fill=Survived))+geom_bar(position="fill")+ylab("Frequency")
```



Relación entre Embarked y survival.

Los pasajeros que más sobrevivieron fueron los que los embarcaron desde Cherbourg con alrededor del 56%, alrededor del 38% que embarcaron en Queenstown sobrevivieron, y alrededor del 35% que embarcaron en S = Southampton sobrevivieron.

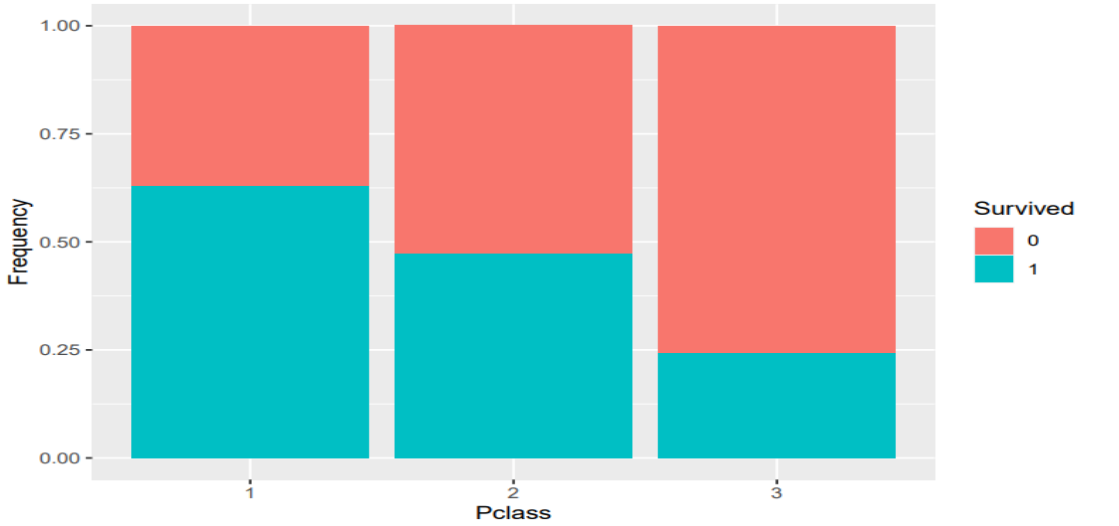
```
# Survival como función de embarked:
ggplot(data = dfaux[1:len_train,],aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+ylab("Freque
```



Relación entre PClass y survival

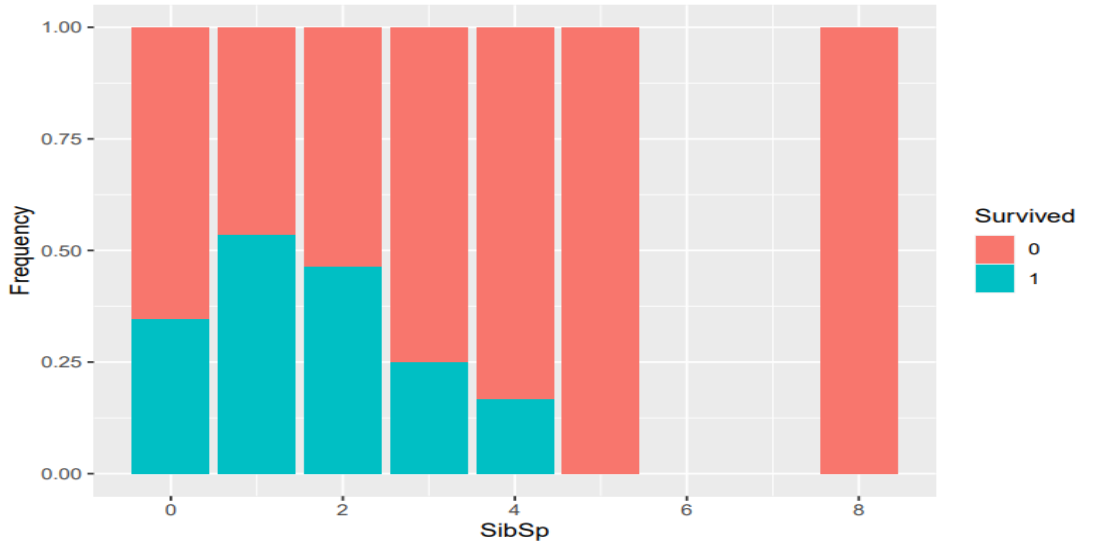
Podemos observar que alrededor de la clase 1 el 63% fueron sobrevivientes, de la clase 2, alrededor del 48% sobrevivieron y de la clase 3 del 25% sobrevivieron.

```
# Survival como función de Pclass:
ggplot(data = dfaux[1:len_train,],aes(x=Pclass,fill=Survived))+geom_bar(position="fill")+ylab("Frequenc
```



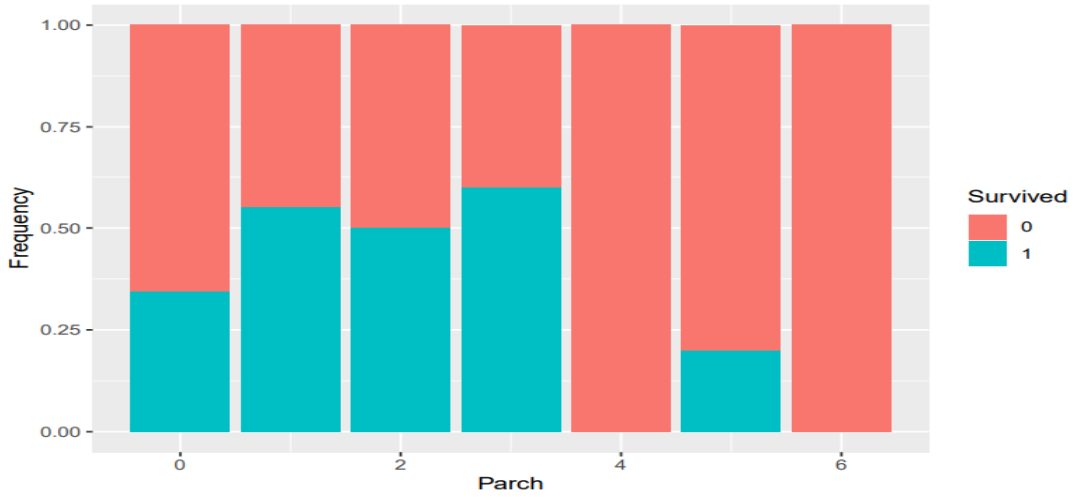
Relación entre SibSp y survival

```
# Survivial as a function of SibSp
ggplot(data = dfaux[1:len_train,],aes(x=SibSp,fill=Survived))+geom_bar(position="fill")+ylab("Frequency
```



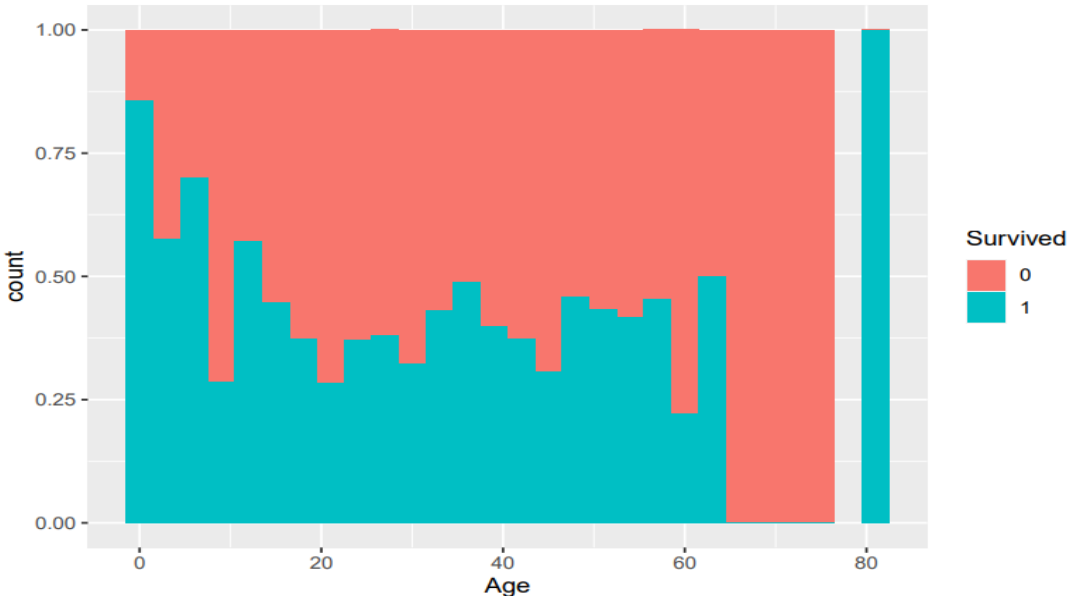
Relación entre Parch y survival

```
# Survivial como función de  Parch
ggplot(data = dfaux[1:len_train,],aes(x=Parch,fill=Survived))+geom_bar(position="fill")+ylab("Frequency
```



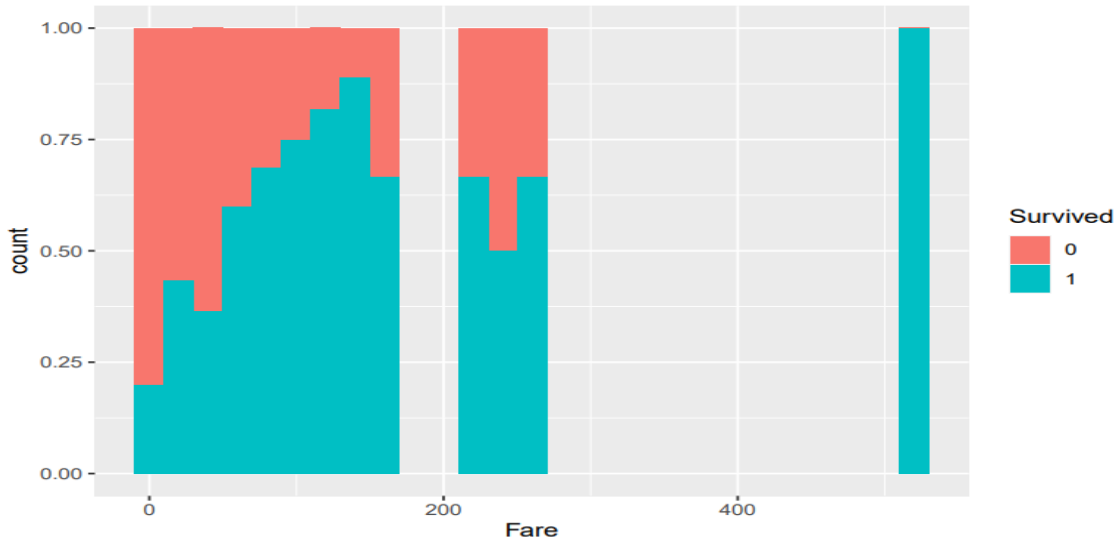
Relación entre Age y survival

```
# Survival as a function of age:

ggplot(data = dfaux[1:len_train,],aes(x=Age,fill=Survived))+geom_histogram(binwidth =3, position="fill'
```



Relación entre Fare y survival

```
# Correlación entre Fare y Survival
ggplot(data = dfaux[1:len_train,],aes(x=Fare,fill=Survived))+geom_histogram(binwidth =20, position="fi
```

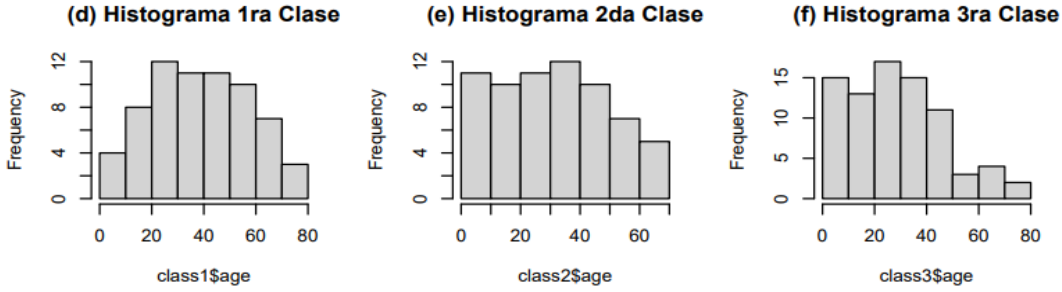


Si vemos la influencia de la edad, el sexo y la clase en los sobrevivientes tendríamos: Anteriormente habíamos hecho subset de las variables como agrupación de datos, pero ahora lo haremos con relación al “Pclass”. Primero agrupamos por “Pclass”:

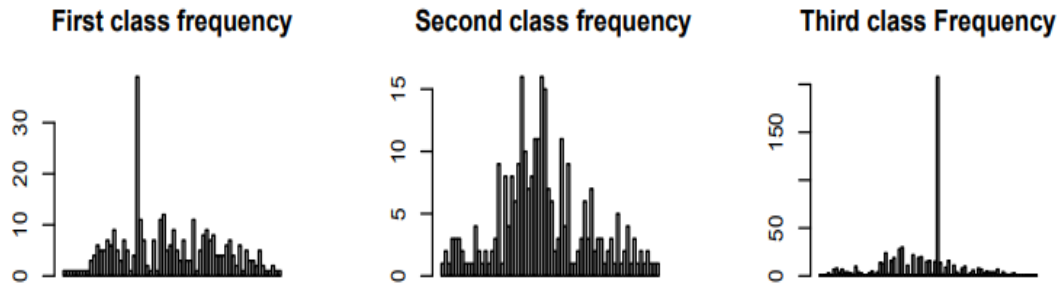
```
# Agregamos las variables que queremos estudiar que influyen la sobrevivencia como sex y age, con Pclas
sex_tot=aggregate(dfaux$Pclass, by=list(sex=dfaux$Sex, pclass=dfaux$Pclass), FUN=function(x){NROW(x)});
Pclass_tot=aggregate(dfaux$Pclass, by=list(pclass=dfaux$Pclass), FUN=function(x){NROW(x)});
age_tot=aggregate(dfaux$Pclass, by=list(age=dfaux$Age, pclass=dfaux$Pclass), FUN=function(x){NROW(x)});

# Hacemos un subset basado en los valores de sex

men<-subset(sex_tot, sex=='male');
women<-subset(sex_tot, sex=='female');

# Ahora vemos el porcentaje de hombres y mujeres.
men$percentage <- round(prop.table(men$x),4)*100;
women$percentage <- round(prop.table(women$x),4)*100;

#Sacamos el subset de edad basado en Pclass.
class1<-subset(age_tot, pclass=='1');
class2<-subset(age_tot, pclass=='2');
class3<-subset(age_tot, pclass=='3');
```

```
# Vamos a graficar lo anterior en relación a la pclass. Utilizaremos el histrograma, el barplot y el qq
data1<-layout(matrix(c(1,2,3,4,5,6), 2, 3, byrow=TRUE),respect=TRUE);
```

```
hist(class1$age, main="(d) Histograma 1ra Clase");
hist(class2$age, main="(e) Histograma 2da Clase");
hist(class3$age, main="(f) Histograma 3ra Clase");
```

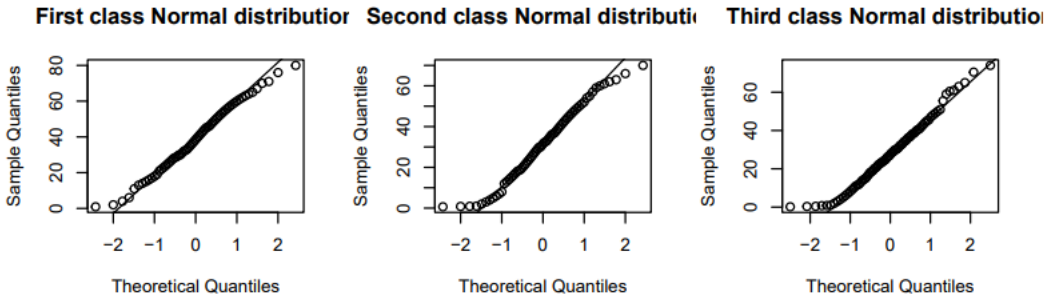


```
# Usando el barplot
data1<-layout(matrix(c(1,2,3,4,5,6), 2, 3, byrow=TRUE),respect=TRUE);

barplot(class1$x, main="First class frequency");
barplot(class2$x, main=" Second class frequency");
barplot(class3$x, main="Third class Frequency");
```



Usamos un qq-plot o "quantile-quantile" plot para determinar la distribución que maneja los datos y ver si maneja la distribución normal, la cual es así como podemos ver en la gráfica.

```
# Usando el qqnorm con qqline
data1<-layout(matrix(c(1,2,3,4,5,6), 2, 3, byrow=TRUE),respect=TRUE);

qqnorm(class1$age, main="First class Normal distribution");
qqline(class1$age);
qqnorm(class2$age, main="Second class Normal distribution");
qqline(class2$age);
qqnorm(class3$age, main= "Third class Normal distribution");
qqline(class3$age);
```



8. APLICACIÓN DE PRUEBAS ESTADISTICAS.

Vamos a aplicar ahora pruebas estadísticas para observar cuales son las variables que influyen más en la sobrevivencia de los pasajeros y cual sería la sobrevivencia de los pasajeros aplicando modelos. En nuestro caso, utilizaremos de nuevo nuestros dataset del train y del test. Aquí aplicaremos el método del árbol de decisión, pero primero realizaremos la regresión logística.

```r
# Seleccionamos de nuevo los datos de train y test y escogemos las variables que vamos a utilizar
train<-dfaux[1:len_train,c("Survived","Pclass","Sex","Age","Fare","SibSp","Parch", "Embarked")]

len_test<-dim(test)[1]
test<-tail(dfaux,len_test)
test<-test[,c("Survived","Pclass","Sex","Age","Fare","SibSp","Parch", "Embarked")]


# Hacemos un regresion logistica

model <- glm(Survived ~.,family=binomial(link='logit'),data=train)
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),

##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6271  -0.6093  -0.4218   0.6173   2.4497
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.108317   0.476722   8.618  < 2e-16 ***
## Pclass2     -0.932800   0.297867  -3.132  0.00174 **
## Pclass3     -2.156069   0.297799  -7.240 4.49e-13 ***
## Sexmale     -2.718678   0.201099 -13.519  < 2e-16 ***
## Age         -0.039136   0.007872  -4.972 6.64e-07 ***
## Fare         0.002292   0.002469   0.928  0.35325
## SibSp       -0.323596   0.109731  -2.949  0.00319 **
## Parch       -0.097449   0.119052  -0.819  0.41305
## EmbarkedQ   -0.025521   0.382000  -0.067  0.94673
## EmbarkedS   -0.440410   0.239742  -1.837  0.06621 .
## ---
## 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:  784.29  on 881  degrees of freedom
## AIC: 804.29
##
## Number of Fisher Scoring iterations: 5
```

Vamos a realizar la predicción de los sobrevivientes con nuestro modelo con el dataset del train:

```r
# Ahora, vemos la prediccion de los sobrevivientes

pred.train <- predict(model,train)
pred.train <- ifelse(pred.train > 0.5,1,0)

# Media de la prediccion verdadera
mean(pred.train==train$Survived)
```

```
## [1] 0.8136925
```

```r
t1<-table(pred.train,train$Survived)
# precisión y recall del modelo
precision<- t1[1,1]/(sum(t1[1,]))
recall<- t1[1,1]/(sum(t1[,1]))
precision
```

```
## [1] 0.799687
```

```r
recall
```

```
## [1] 0.9307832
```

```r
# F1 score
F1<- 2*precision*recall/(precision+recall)
F1
```

```
## [1] 0.8602694
```

Vamos a ver el accuracy de nuestro modelo.

```r
table(train$Survived, pred.train >= 0.5)
```

```
##
##     FALSE TRUE
##   0   511   38
##   1   128  214
```

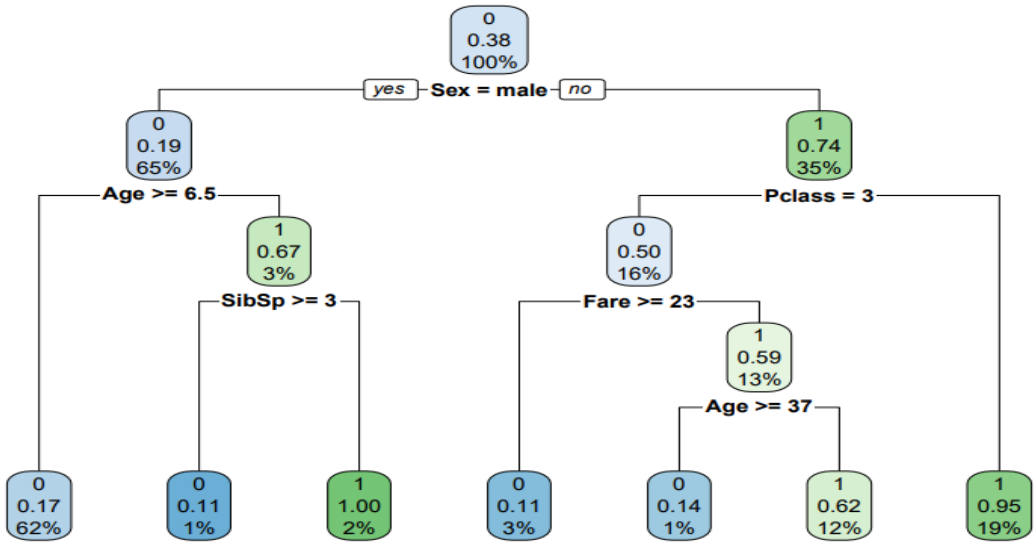
```r
accuracy = (244 + 458) / nrow(train)
sensitivity = 244 / (244 + 98)
specificity = 458 / (458 + 91)

cat("accuracy: ", accuracy)
```

```
## accuracy:  0.7878788
```

9. REPRESENTACIÓN GRÁFICA DE LOS RESULTADOS A PARTIR DE TABLAS Y GRÁFICAS

Existen diferentes gráficas para usar como modelos de predicción. Uno de ellos es el árbol de decisión el cual se busca predecir la probabilidad de que se llegue el objetivo en base a ciertas condiciones. Usamos en nuestro caso para representar los resultados, un árbol de decisión el cual al comienzo transforma la variable “survived” para que a partir de sexo puede ir ramificándose la resolución al problema.

Graficamos con rpart.plot el árbol de decisión donde se puede varios niveles.



Podemos también exportar los resultados de la predicción en un archivo csv.

```
pred.test <- predict(model,test)
pred.test <- ifelse(pred.test > 0.5,1,0)

pred.test
```

```
test$Survived<- pred.test

write.csv(pred.test,file="prediction.csv",row.names = F)
```

Tambien se exporta el archivo ya limpio.

```
write.csv(dfaux,file="archivolimpio.csv",row.names = F)
```

10. RESOLUCIÓN DE PROBLEMAS Y CONCLUSIONES

Con el uso de árboles de decisión para la resolución de nuestro problema en cuanto a la sobrevivencia de los pasajeros y el impacto de ciertas variables en ella, hemos encontrado que nos trae los siguientes resultados comprendidos en varios niveles:

- si es hombre (male) y tiene una edad Age>=6.5 entonces muere
- Si es hombre (male) y tiene una edad Age< 6.5 y SibSp>=2.5 entonces muere
- Si es hombre (male) y tiene una edad Age< 6.5 y SibSp<2.5 entonces sobrevive
- si es mujer (female) y pclass< 3 entonces sobrevive
- si es mujer (female), Pclass=3, Fare>=23.35, entonces muere
- si es mujer (female), Pclass=3, Fare< 23.35, Age>=36.5 entonces muere
- si es mujer (female), Pclass=3, Fare< 23.35, Age< 36.5 entonces sobrevive

Se puede observar que con estos resultados llegamos a la conclusión de que no hay influencia de las clases en los hombres en donde hay prioridad a los niños menores de 6.5 años y que tienen una cantidad menor de hermanos. En el caso de los hombres, la supervivencia es muy pequeña a diferencia de las mujeres. En el caso de las mujeres, vemos que las clases 1 y 2 que son consideradas alta y media, sobreviven, los que nos demuestra que juega un papel importante el aspecto socioeconómico en la supervivencia. En el caso de las mujeres con la “pclass” 3, que es la baja, entra los factores en donde cuando son menores de 36.5 años sobreviven. De aquí podría comprobarse la teoría de mujeres y niños/as primero son los que más probabilidades tuvieron de sobrevivir en el titanic. Se observa también que la variable “pclass” tiene una influencia mucho mayor que las otras variables como anteriormente se había mencionado.

11. EXPORTACIÓN DEL CÓDIGO

En el github se puede observar el archivo subido con el código realizado en el programa r. También se encuentra en el github los archivos csv de los dataset, el dataset limpio y el de las predicciones.

• Mejia Quintero Dayana: https://github.com/danamejia1810/Practica-2-Titanic-dataset.git

• Peterson Christopher: https://github.com/christopherapeterson/Practica-2-Titanic-dataset.git

12. BIBLIOGRAFÍA

• Calvo M, Subirats L, Pérez D (2019). Introducción a la limpieza y análisis de los datos. Editorial

UOC.

• Dalgaard, Peter (2008). Introductory statistics with R. Springer Science & Business Media.

• Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan

Kaufmann.

• Osborne, Jaso W. (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores.

Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.

• Squire, Megan (2015). Clean Data. Packt Publishing Ltd.

• Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.

Contribuciones Firma

Investigación previa	DM, CP
Redacción de las respuestas	DM, CP
Desarrollo código	DM, CP