Práctica
II

Tipología y ciclo de vida de los datos

Arturo González Díez
71454421-E

.

Contenido

0. Introducción. 3

1. Descripción del dataset. ¿Por qué es importante y qué 3

pregunta/problema pretende responder? 3

2. Integración y selección de los datos de interés a analizar. 4

3. Limpieza de los datos. 5

3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos? 5

3.2. Identificación y tratamiento de valores extremos. 6

4. Análisis de los datos. 7

4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar). 7

4.2. Comprobación de la normalidad y homogeneidad de la varianza. 8

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. 10

5. Representación de los resultados a partir de tablas y gráficas. 10

6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema? 17

7. Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python. 17

0. Introducción.

Debido a que el dataset empleado en la práctica anterior tiene pocos datos y ya se encuentran limpios, he escogido el dataset 'train.csv' del Titanic obtenido de la página https://www.kaggle.com/c/titanic/data .

En esta práctica trataremos de analizar los datos del dataset para eliminar aquellos que sean redundantes o prescindibles y adaptar aquellos que no se encuentren en el formato correcto o contengan valores no válidos. Además, una vez limpiado el código, trataremos de validarlos y obtener resultados que nos ayuden a realizar predicciones y/o entrenamientos sobre otros posibles registros.

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

El dataset de esta práctica es un conjunto de datos correspondientes a los datos de los tripulantes que viajaban en el Titanic el día 15 de abril de 1912, día en el que chocó contra un iceberg y produjo la muerte de gran parte de esta tripulación.

El objetivo es poder predecir qué pasajeros (uno por cada registro) sobreviviría al hundimiento y cuáles morirían. Para ello realizaremos una selección de los datos útiles, que nos sean relevantes para el estudio, a continuación, limpiaremos estos datos seleccionados, eliminando lo que no se encuentre en su formato correcto o adaptándolo a las necesidades tratando de no transformar esta información (lo que nos conduciría a predicciones incorrectas). Una vez tratados, podremos analizar y comprobar en profundidad estos datos y seleccionar cuáles vamos a emplear para realizar las predicciones. Por último, representaremos gráficamente mediante distintas tablas los datos obtenidos y trataremos de generar reglas/conclusiones a partir de estas.

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

El dataset consta de doce campos que se describen a continuación:

PassengerId: Campo de tipo numérico generado de manera incremental y con un valor único para cada pasajero.

Survived: Campo de valor booleano, output, que indica si el pasajero ha sobrevivido (1) o no (0).

Pclass: Campo categórico que indica a qué clase pertenecía el pasajero, el tipo del ticket. (1 clase, 2 clase ó 3 clase).

Name: Cadena de texto con el formato ‘ Apellido, TituloPersonal Nombre’

Sex: Campo categórico (male, female) que contiene el género del pasajero.

Age: Edad que tenía el pasajero en el momento del viaje del Titanic (15-04-1912).

SibSp: Hermanos o cónyuges que tenía el pasajero a bordo del Titanic (amantes y novios fueron ignorados).

Parch: Indica si viajaban con padres o hijos. En el caso de los niños que viajaban solo con una niñera, el valor será 0.

Ticket: Cadena de texto con el nombre del ticket, los hay en varios formatos, íntegramente numéricos o con caracteres incluídos.

Fare: Valor numérico que indica la tarifa del ticket.

Cabin: Número correspondiente a la cabina en la que se hospedaba el pasajero.

Embarked: Dato categórico correspondiente a la ciudad en la que embarcó (C = Cherbourg, Q = Queenstown, S = Southampton).

¿Qué campos quedan excluídos de nuestro estudio?

PassengerId: Los ids para este caso de estudios son prescindibles ya que son datoas generados para el almacenaje en la base de datos y para la asignación de un valor único a cada registro. En este caso, necesitamos saber ‘si sobrevivió’ en base a una serie de parámetros y no ‘quién sobrevivió’.

Name: Se aplica la misma lógica que a PassengerId y responde a la misma pregunta, ¿Sobrevivió? Por lo tanto lo eliminamos de nuestro dataset.

Ticket, **Fare**: Quedan excluidos por falta de información, los tickets no aportan información relevante al estudio mientras que las tarifas carecen de una unidad de medida por lo que no sabe a qué hace referencia, se podría intuir que son dólares, no obstante, la coma flotante separa por la derecha cuatro valores en vez

de tres y de igual manera, existen valores mucho mayores para algunas tarifas de tercera clase que de primera lo que hace pensar que no se refiere a cuantías económicas.

3. Limpieza de los datos.

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

En este dataset el valor 0 únicamente se encuentra como dato medible bien en campos numéricos o bien en campos booleanos donde representa el valor 'false'.

Por otra parte, campos vacíos únicamente se encuentran en la columna 'Cabin'. En este caso, saber el número de la habitación no nos aporta información relevante ya que no sabemos la distribución de los dormitorios en el barco no obstante, esta información podemos transformarla convirtiendo este dato a boleano e indicándonos si se sabe su cabina o no, si tiene un valor en el dataset se marcará a 1 y si es un campo vacío, a 0.

Usando el lenguaje R podemos realizar un contado de cuántos campos tienen valores null, mediante la función 'sapply':

```
> sapply(titanic, function(x) sum(is.na(x)))
Survived   Pclass      Sex      Age    SibSp    Parch     Fare    Cabin Embarked
       0        0        0      177        0        0        0        0        0
```

Ahí podemos observar que en 177 casos no se dispone de información acerca de la edad.

Para el problema de los valores nulos en la edad, emplearemos el método de los k-Nearest Neighbour, tomando un valor en función de los valores más próximos. Esta elección se toma ya que es preferible trabajar con edades aproximadas antes que eliminar esos registros o trabajar con campos vacíos o ceros.

Desde el punto de vista personal, recomiendo barajar la gestión de todos los campos ya que así generaríamos un código de limpieza optimizado para cualquier dataset evitando los problemas que traería consigo el no contemplar estos tratamientos.

3.2. Identificación y tratamiento de valores extremos.

Los valores extremos se dan en valores numéricos, en nuestro caso estamos trabajando con la mayoría de los campos categóricos para todos los campos excepto la edad, SibSp y Parch

Una vez aplicado el método k-Nearest Neighbour, llamaremos a las funciones min y max para comprobar que todos los campos se encuentran dentro de una lógica y que no van a distorsionar los resultados:

```
> summary(titanic$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.42   21.00   28.00   29.59   37.00   80.00
```

Se puede observar que el rango de valores se encuentra [0.42, 80] por lo que no se han localizados valores extremos que tratar, la media para este conjunto de datos es de 29.59.

Para el caso de SibSp y Parch:

```
> summary(titanic$SibSp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   0.000   0.523   1.000   8.000
> summary(titanic$Parch)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.3816  0.0000  6.0000
```

El máximo de hermanos / conyuges que viajaban es de 8 mientras que en el caso de padres/hijos, la máxima relación fue de 6 personas. Aunque las medias de ambos datos sean bastante bajas, 0.523 y 0.3816 respectivamente, no podemos decir que los máximos sean valores extremos pues la diferencia numérica entre el mínimo y el máximo es también muy baja.

4. Análisis de los datos.

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

A continuación, vamos a seleccionar grupos dentro de nuestro dataset que puedan resultarnos de utilidad para analizar y realizar pruebas estadísticas, lo haremos introduciendo en un array las variables categóricas de los datos, algunos datos no los emplearemos para el análisis final no obstante, los estudiamos:

```
titanic.Status.Survivor <- titanic[titanic$Survived == 1,]
titanic.Status.Dead <- titanic[titanic$Survived == 0,]

titanic.Class.First <- titanic[titanic$Pclass == "1",]
titanic.Class.Second <- titanic[titanic$Pclass == "2",]
titanic.Class.Third <- titanic[titanic$Pclass == "3",]

titanic.Embarked.Southampton <- titanic[titanic$Embarked == "S",]
titanic.Embarked.Cherbourg <- titanic[titanic$Embarked == "C",]
titanic.Embarked.Queenstown <- titanic[titanic$Embarked == "Q",]

titanic.Sex.Male <- titanic[titanic$Sex == 0,]
titanic.Sex.Female <- titanic[titanic$Sex == 1,]

titanic.Cabin.Assigned <- titanic[titanic$Cabin == 1,]
titanic.Cabin.NonAssigned <- titanic[titanic$Cabin == 0,]
```

Con esta sentencia lo que se consigue es obtener nuevos dataset con los datos que cumplan la característica. Por ejemplo, en el primer caso, se creará un conjunto de datos con todos aquellos pasajeros que hayan sobrevivido:

```
> titanic.Status.Survivor
    Survived Pclass Sex   Age SibSp Parch Cabin Embarked
2          1      1   0 38.00     1     0     1        C
3          1      3   0 26.00     0     0     0        S
4          1      1   0 35.00     1     0     1        S
9          1      3   0 27.00     0     2     0        S
10         1      2   0 14.00     1     0     0        C
11         1      3   0  4.00     1     1     1        S
12         1      1   0 58.00     0     0     1        S
16         1      2   0 55.00     0     0     0        S
18         1      2   1 34.00     0     0     0        S
20         1      3   0 18.00     0     0     0        C
22         1      2   1 34.00     0     0     1        S
23         1      3   0 15.00     0     0     0        Q
24         1      1   1 28.00     0     0     1        S
26         1      3   0 38.00     1     5     0        S
29         1      3   0 26.00     0     0     0        Q
32         1      1   0 23.00     1     0     1        C
33         1      3   0 22.00     0     0     0        Q
37         1      3   1 24.00     0     0     0        C
40         1      3   0 14.00     1     0     0        C
44         1      2   0  3.00     1     2     0        C
45         1      3   0 19.00     0     0     0        Q
48         1      3   0 22.00     0     0     0        Q
53         1      1   0 49.00     1     0     1        C
54         1      2   0 29.00     1     0     0        S
56         1      1   1 40.00     0     0     1        S
57         1      2   0 21.00     0     0     0        S
59         1      2   0  5.00     1     2     0        S
```

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

Para la comprobación de la normalidad vamos a usar la prueba de Anderson-Darling, una prueba no paramétrica sore si los datos provienen de una distribución específica o no, siguiendo la fórmula

$$A^2 = -N - S$$

$$S = \sum_{k=1}^{N} \frac{2k-1}{N} \left[\ln F(Y_k) + \ln(1 - F(Y_{N+1-k}))\right]$$

Para la que generaremos el código en R utilizando la librería nortest y aplicando un nivel de significación de 0.05, para las que se obtendrá un p-valor superior a 0.05.

```
library(nortest)

alpha = 0.05
col.names = colnames(titanic)
for (i in 1:ncol(titanic)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
  if (is.integer(titanic[,i]) | is.numeric(titanic[,i])) {
    p_val = ad.test(titanic[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
      if (i < ncol(titanic) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
}
```

```
Variables que no siguen una distribución normal:
Survived, Pclass, Sex,
Age, SibSp, Parch,
Cabin
```

Para comprobar la Varianza podríamos usar la prueba de Levene:

```
y <- c(titanic.Status.Survivor, titanic.Status.Dead)
group <- as.factor(c(rep(1, length(titanic.Status.Survivor)), rep(2, length(titanic.Status.Dead))))
var <- leveneTest(y ~ group)
```

No obstante para este caso utilizaremos el test de Fligner-Killeen, estudiaremos la homogeneidad entre dos de los grupos, y podremos observar que el p-value es superior a 0.05 así que se acpeta como hipótesis de que las varianzas de ambas muestras son homogéneas.

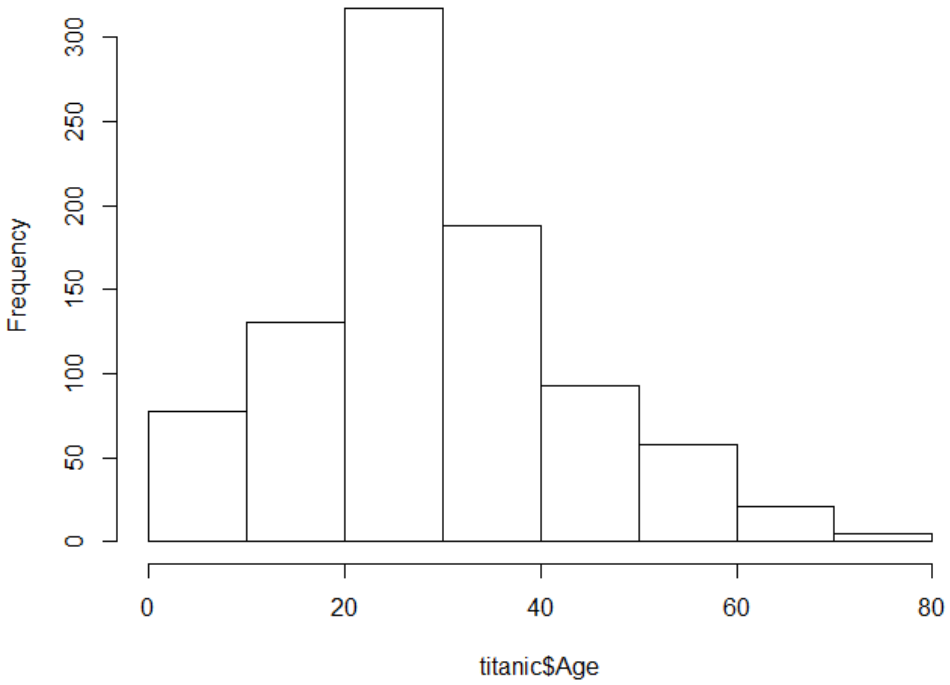
```
> fligner.test(Age ~ Survived, data = titanic)

        Fligner-Killeen test of homogeneity of variances

data:  Age by Survived
Fligner-Killeen:med chi-squared = 0.41206, df = 1, p-value = 0.5209
```

4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

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

(Ambas preguntas respondidas en la misma respuesta)

Mediante la función de representación de un histograma (hist) podemos ver de manera gráfica qué rangos prevalecían sobre otros. De esta manera, realizando un hist(titanic$Age) podemos comprobar que el mayor rango era el comprendido entre 20 y 30 años.



A continuación analizamos qué grupos de pasajeros sobrevivieron en un mayor porcentaje:

Supervivientes por clase

```
> firstSurvived <- sum(titanic.Class.First$Survived)
> porcentaje <- (firstSurvived * 100) / totalFirst
> porcentaje
[1] 62.96296
> totalSecond <- nrow(titanic.Class.Second)
> totalSecond <- nrow(titanic.Class.Second)
> secondSurvived <- sum(titanic.Class.Second$Survived)
> porcentaje2 <- (secondSurvived * 100) / totalSecond
> porcentaje2
[1] 47.28261
> totalThird <- nrow(titanic.Class.Third)
> thirdSurvived <- sum(titanic.Class.Third$Survived)
> porcentaje3 <- (thirdSurvived * 100) / totalThird
> porcentaje3
[1] 24.23625
```

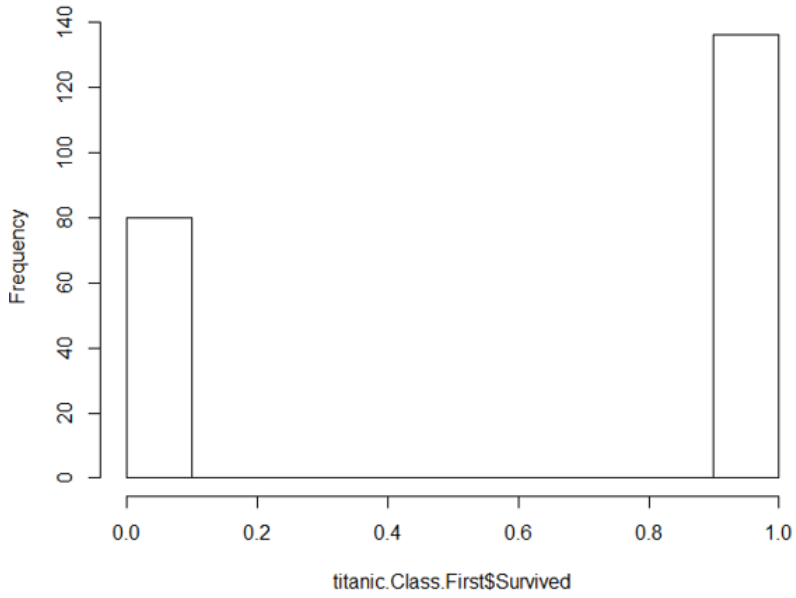
Los resultados obtenidos:

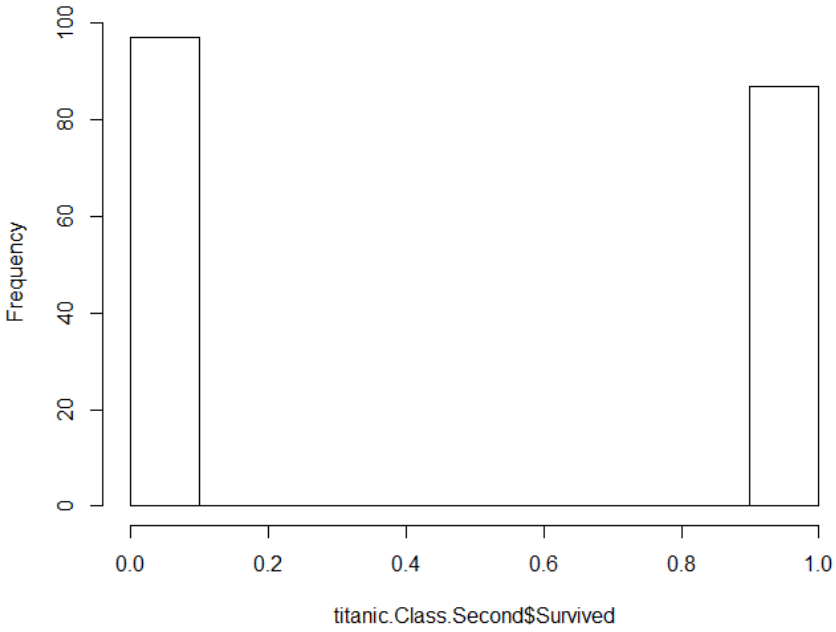
1ª clase : 63% de supervivencia

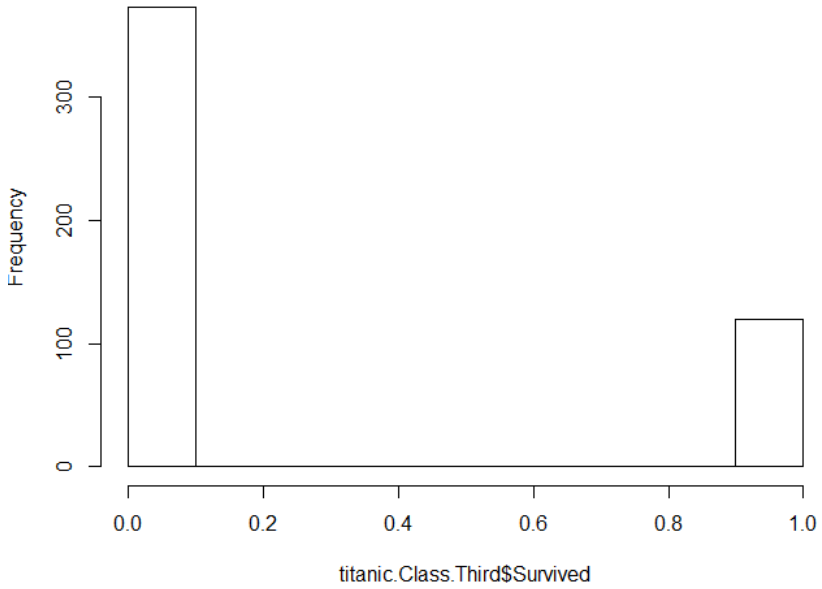
2ª clase: 47% de supervivencia

3ª clase: 24% de supervivencia

Gráficamente se puede representar utilizando la función hist:







Supervivientes por sexo

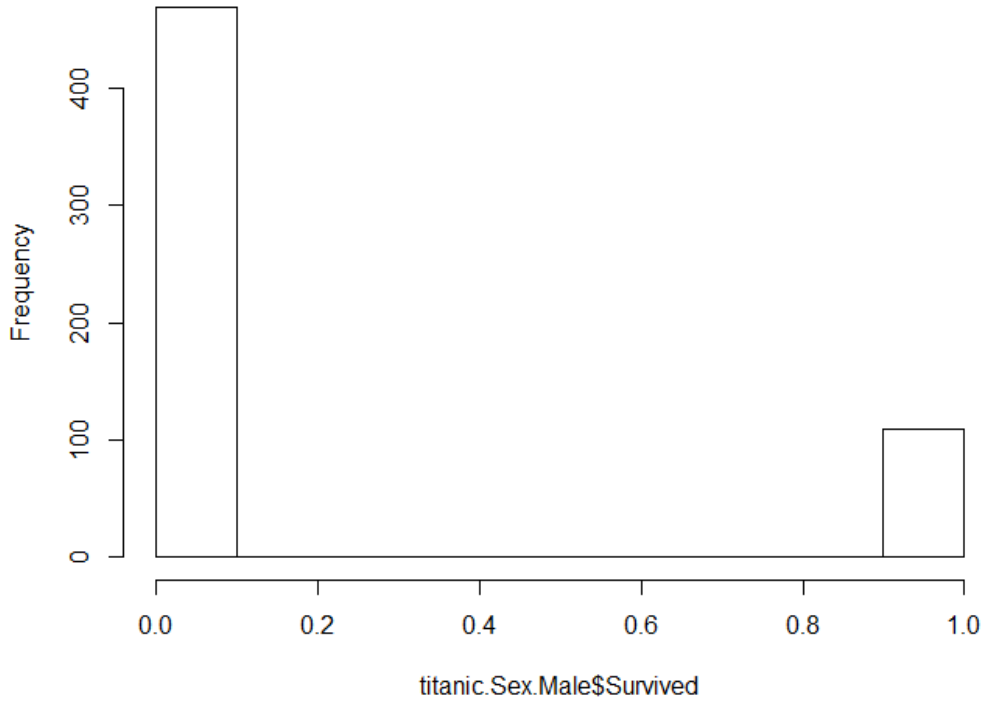
```
> totalMale <- nrow(titanic.Sex.Male)
> maleSurvived <- sum(titanic.Sex.Male$Survived)
> porcentaje4 <- (maleSurvived * 100) / totalMale
>
> totalFemale <- nrow(titanic.Sex.Female)
> femaleSurvived <- sum(titanic.Sex.Female$Survived)
```

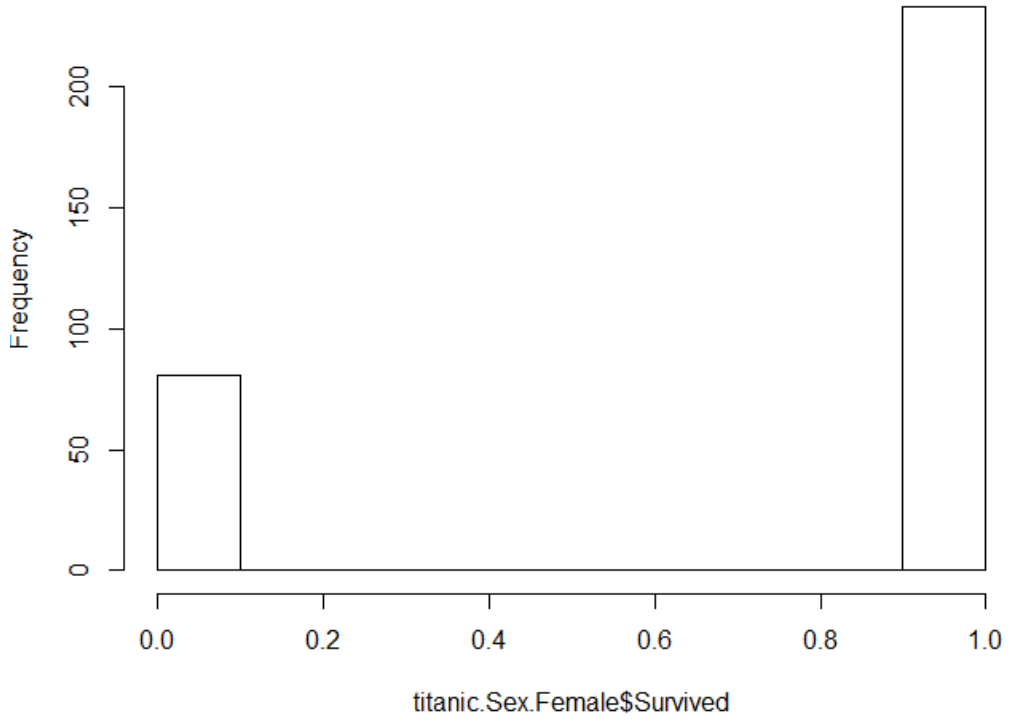
```
> porcentaje5 <- (femaleSurvived * 100) / totalFemale
> porcentaje4
[1] 18.89081
> porcentaje5
[1] 74.20382
```

Los resultados obtenidos son:

- Hombre supervivientes: 19%
- Mujeres supervivientes: 74%

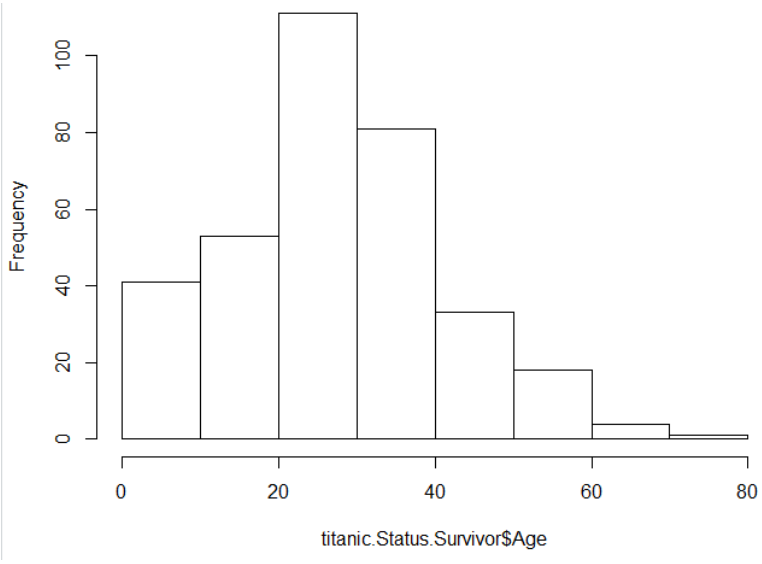
Los gráficos obtenidos:

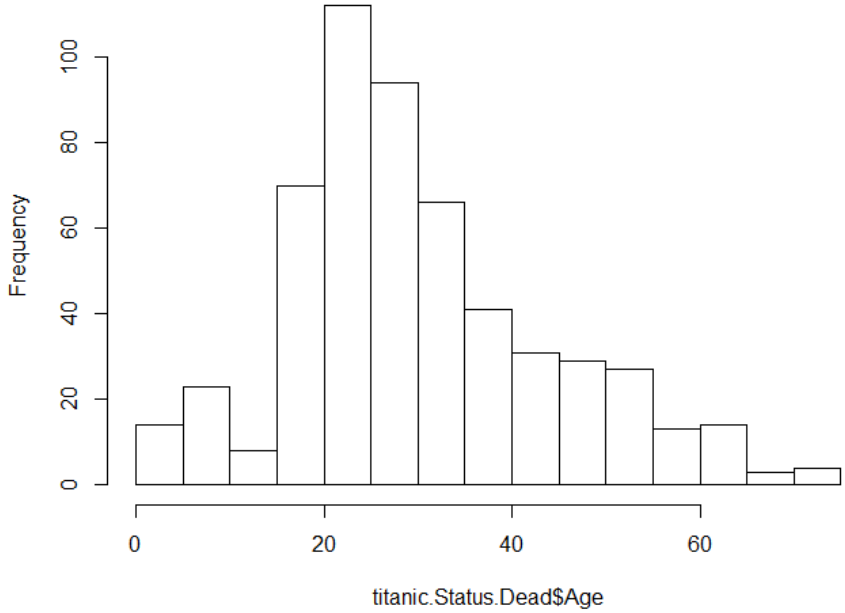




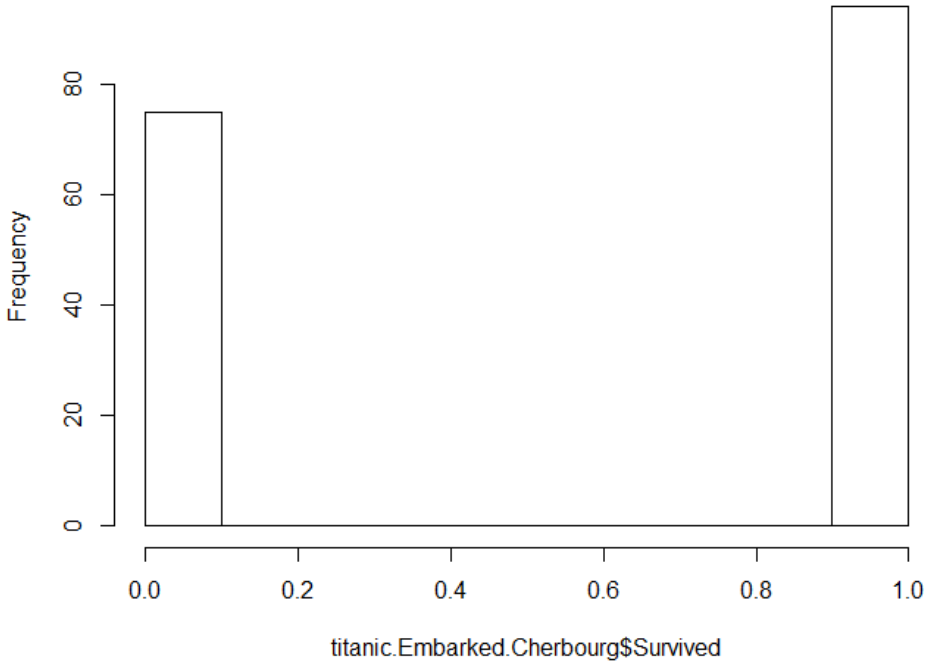
Supervivientes/Muertes por edad

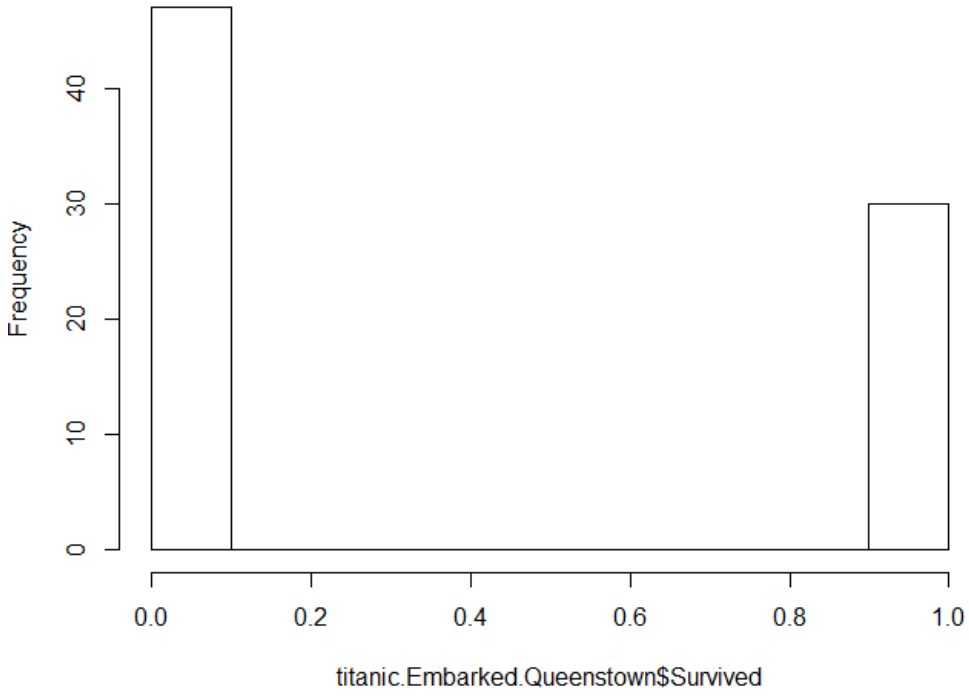
El gráfico de supervivientes / muertes por edad es el siguiente:

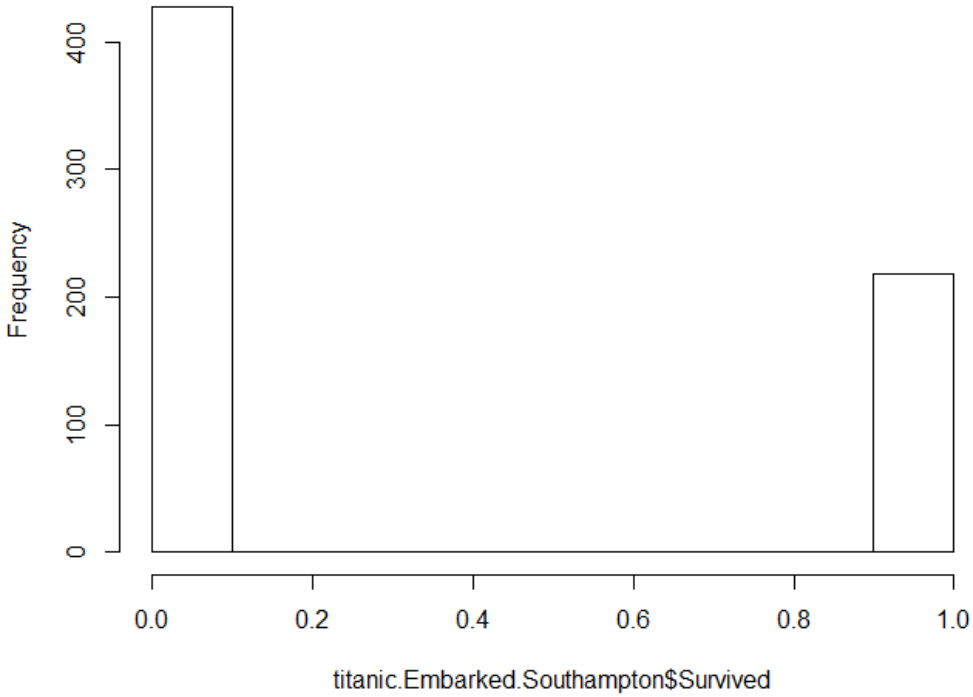




Supervivientes por embarcación







Como se puede observar en las gráficas, todo parece indicar que el lugar de embarque no fue decisivo a la hora de decidir qué pasajeros deberían salvarse.

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

Las conclusiones obtenidas son las siguientes:

- Se dio prioridad a salvar a los pasajeros de primera clase frente a los de segunda y a los de esta frente a los de tercera.

- Se dio prioridad a la salvación de mujeres (sobrevive el 74%) frente a hombres (sobrevive el 18%).

- Comparando los gráficos por edad se puede ver que el mayor número de muertos fue en la franja de 20 a 25 años.

- Los datos de cabina y familiares no aportan información relevante

- Los datos de embarcación, como se puede observar en las gráficas indican que no se dio prioridad a los de un punto de embarcación u otro.

Para poder obtener una información más precisa necesitaríamos aplicar algún algoritmo como puede ser un árbol de decisión o crear reglas de asociación para poder clasificar con más exactitud a cada pasajero que recibiéramos en un fichero de test.

7. Código: Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.