MÁSTER U. DE CIENCIA DE DATOS

Tipología y Ciclo de Vida de los Datos. Aula 1

(M2.851)

PRÁCTICA 2

Alumnos:

Olast Arrizibita Iriarte - oarrizibita@uoc.edu

Enrique Pérez Balbuena - eperezbal@uoc.edu

Índice

1. Dataset. 3
1.1. Descripción de dataset. 3
1.2. ¿Por qué es importante? ¿Qué pretende responder? 5

2. Integración y selección de datos. 6

3. Limpieza de datos. 8
3.1. Gestión de datos con ceros o elementos vacíos. 8
3.2. Identificación y tratamiento de valores extremos. 8

4. Análisis de datos. 9
4.1. Selección de los grupos de datos a analizar/comparar. 9
4.2. Comprobación de la normalidad y homogeneidad varianza. 11
4.3. Aplicación de pruebas estadísticas. 15

5. Representación de los resultados. 23

6. Resolución del problema. 27

7. Código en R de la práctica y tabla de contribuciones. 27

8. Bibliografía. 28

1. Dataset

1.1 Descripción de dataset

El conjunto de datos que queremos analizar se ha obtenido a través de la plataforma kaggle [1] y pertenece a un dataset libre sobre datos de los pasajeros del Titanic.

La plataforma web ha dividido el dataset principal en tres conjuntos de datos.

Un dataset denominado 'train.csv' que es el conjunto de datos con el que entrenaremos nuestro modelo. Este conjunto de datos está formado por 891 registros y 12 variables. Podemos observar una serie de resumen estadístico en la siguiente imagen.

```
##   PassengerId       Survived          Pclass
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000
##  Median :446.0   Median :0.0000   Median :3.000
##  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
##
##                                     Name         Sex           Age
##  Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.42
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.12
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.00
##  Abelson, Mr. Samuel                   :  1                Mean   :29.70
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00
##  Adahl, Mr. Mauritz Nils Martin        :  1                Max.   :80.00
##  (Other)                               :885                NA's   :177
##      SibSp           Parch             Ticket         Fare
##  Min.   :0.000   Min.   :0.0000   1601     :  7   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082   :  7   1st Qu.:  7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45
##  Mean   :0.523   Mean   :0.3816   3101295  :  6   Mean   : 32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088   :  6   3rd Qu.: 31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144  :  6   Max.   :512.33
##                                   (Other)  :852
##          Cabin     Embarked
##              :687    :  2
##  B96 B98     :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6          :  4   S:644
##  C22 C26     :  3
##  D           :  3
##  (Other)     :186
```

Resumen estadístico del dataset 'train.csv'

También tenemos un dataset denominado 'test.csv', con el que podremos validar el modelo. Está formado por 418 registros y 11 variables. Podemos observar un resumen estadístico de los mismos en la siguiente imagen.

```
##   PassengerId         Pclass
##  Min.   : 892.0   Min.   :1.000
##  1st Qu.: 996.2   1st Qu.:1.000
##  Median :1100.5   Median :3.000
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##                                          Name         Sex
##  Abbott, Master. Eugene Joseph             :  1   female:152
##  Abelseth, Miss. Karen Marie               :  1   male  :266
##  Abelseth, Mr. Olaus Jorgensen             :  1
##  Abrahamsson, Mr. Abraham August Johannes :  1
##  Abrahim, Mrs. Joseph (Sophie Halaut Easu):  1
##  Aks, Master. Philip Frank                 :  1
##  (Other)                                   :412
##       Age            SibSp            Parch             Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   PC 17608:  5
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   113503  :  4
##  Median :27.00   Median :0.0000   Median :0.0000   CA. 2343:  4
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923   16966   :  3
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000   220845  :  3
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000   347077  :  3
##  NA's   :86                                        (Other) :396
##       Fare                     Cabin     Embarked
##  Min.   :  0.000                 :327   C:102
##  1st Qu.:  7.896   B57 B59 B63 B66:  3   Q: 46
##  Median : 14.454   A34           :  2   S:270
##  Mean   : 35.627   B45           :  2
##  3rd Qu.: 31.500   C101          :  2
##  Max.   :512.329   C116          :  2
##  NA's   :1         (Other)       : 80
```

Resumen estadístico del dataset 'test.csv'

Finalmente un dataset denominado 'gender_subbmision.csv' donde tenemos la variable objetivo, es decir, las etiquetas respuesta de los datos del 'test.csv'. Está formado por los 418 registros y 2 variables.

Sumando ambos ficheros en total tendríamos 1309 registros, lo que supone una división del 68 - 32% para los datos de entrenamiento y test respectivamente.

Vamos a describir a continuación las diferentes variables del dataset 'train.csv' y 'test.csv' de forma conjunta, ya que tienen mismas, pero con distintos datos:

PassengerId: es el ID, el identificador de cada pasajero.

Survived: indica si el pasajero sobrevivió o no. Esta variable toma dos valores posibles. Valor '0' significa que la persona no sobrevivió. Valor '1' que sí sobrevivió.

Pclass: es la clase del billete. Puede tomar tres tipos de valores. Valor '1' significa que el pasajero pertenece a 1ª clase, el valor '2' que pertenece a 2ª clase y el valor '3' que pertenece a 3ª clase.

Name: es el nombre del pasajero.

Sex: es el género del pasajero. Puede tomar dos valores categóricos. Valor 'male' que significa que era un hombre o el valor 'female' que era mujer.

Age: es la edad del pasajero. Toma valores decimales. El valor mínimo es '0.17' año y el máximo es de '80' años. Tenemos 263 registros con valores 'NA'.

SibSp: es el número de hermanos o cónyuges del pasajero. Los valores oscilan entre el '0' y el '8'

Parch: es el número de parientes del pasajero. Los valores oscilan entre '0' y el '6'. Si el valor es '0' significa que el niño estaba con una nanny.

Ticket: es número del billete.

Fare: es el precio del billete en dólares. Oscila entre un valor mínimo de '0' hasta un máximo de '512.3292'. Toma valores decimales.

Cabin: es el número de la cabina que el pasajero ocupaba. Tenemos 1014 registros con valores vacíos.

Embarked: es el nombre del puerto desde donde el pasajero accedía al barco. Toma tres valores posibles. El valor de 'C' era el puerto de Cherbourng, en Francia, el valor 'Q' era el puerto Queenstown, en Irlanda y el valor de 'S' pertenecía al puerto de Southampton, en Inglaterra. Tenemos dos registros con valores vacíos.

1.2 ¿Por qué es importante? ¿Qué pretende responder?

Para entender un poco mejor el contexto de los datos, hemos acudido a la Wikipedia [2] en donde queda documentado por ejemplo el número de pasajeros de cada clase, la relación de los camarotes o cómo estaban dispuestos los botes salvavidas, los puertos de embarque, etc.

Así podemos observar cómo el número total de pasajeros de tercera clase (unas 706 personas) era igual a la suma de los de primera (unas 325 personas) más los de segunda (unas 285 personas). También sabemos que dependiendo en qué lado del barco (estribor o babor) accedieses a los botes salvavidas, si eras hombre, podías acceder o no a los mismos. En el lado de estribor se permitían plaza a los hombres, mientras que en lado de babor lo tenían prohibido.

Sabemos que el barco salió de Southampton (Inglaterra). Atravesó el Canal de la Mancha para detenerse en Cherburgo (Francia), realizando una última escala en Queenstown (Irlanda) antes de partir para New York. La mayoría del pasaje de tercera clase y el correo embarcó en el puerto irlandés de Queenstown.

Una de las cuestiones que se pretende responder es saber qué tipo de pasajeros tenían más probabilidad de poder sobrevivir al hundimiento del barco, así que trataremos de predecir qué personas sobrevivieron, en base a ciertas características (o 'features').

Claro, si descubrimos que el grupo de personas que más sobrevivió eran las que pertenecían a la primera clase, con un mayor poder económico y social y las que fallecieron en el hundimiento del barco, correspondían en su gran mayoría a la tercera clase, que eran pasajeros con un nivel económico mucho más bajo, podemos tener la tentación de pensar que hubo una discriminación a la hora de salvar vidas y entender como que la vida de los más ricos era más valiosa que la del resto.

También sabemos que para salvaguardar las distancias entre clases, hubo zonas de acceso a la tercera clase, que estaban cerradas en el momento del impacto contra el iceberg. Esto unido a que los camarotes de esa clase se encontraban los más alejados a la cubierta en donde se encontraban los botes salvavidas y al hecho de que muchos de los pasajeros de esta clase eran inmigrantes, y muchos de ellos no sabían hablar inglés, hizo que tuviesen mayores impedimentos y dificultades para orientarse por el barco y acceder cuanto antes a la cubierta del barco.

Así que es importante conocer los detalles del porqué de la tragedia. No sólo sus cifras, sino tratar también de averiguar el relato de los sucesos. Esto nos ayudará a comprender mejor las causas y las responsabilidades. Hará que como consecuencia del análisis del hundimiento, podamos mejorar en nuevas medidas de seguridad, por ejemplo, con un mayor número de botes salvavidas por pasajero, nuevo diseño de los barcos (con nuevos materiales, nueva disposición de los compartimentos estancos, accesos a las diferentes clases) o en nuevos protocolos de actuación que garanticen mayor seguridad a los pasajeros y su tripulación. Todo para evitar en lo posible que se repitan tragedias como las del Titanic.

2. Integración y selección de datos

Aunque el dataset original viene dividido en tres conjuntos separados, nosotros vamos a integrar en un primero momento todos los datos en un solo dataset. De esta forma podremos realizar la limpieza y gestionar cada caso particular (falta de valores, registros con ceros, etc.) de forma conjunta a todas las variables por igual (tanto los datos del fichero 'train' como los del 'test').

De momento no vamos a eliminar ni seleccionar los datos. Una vez podamos analizarlos, estudiar los valores vacíos o la falta de datos, podremos decidir con qué variables nos quedamos.

```
df_test_gender <- merge(df_gender, df_test,  by='PassengerId')
```

Finalmente creamos el dataset final con los dos dataframes.

```
df <- rbind(df_train, df_test_gender)
```

En la siguiente imagen podemos observar un resumen estadísticos de los 1309 registros

```
##   passengerId       survived          pclass
##  Min.   :   1   Min.   :0.0000   Min.   :1.000
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000
##  Median : 655   Median :0.0000   Median :3.000
##  Mean   : 655   Mean   :0.3774   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##
##                                  name          sex           age
##  Connolly, Miss. Kate               :   2   female:466   Min.   : 0.17
##  Kelly, Mr. James                   :   2   male  :843   1st Qu.:21.00
##  Abbing, Mr. Anthony                :   1                Median :28.00
##  Abbott, Mr. Rossmore Edward        :   1                Mean   :29.88
##  Abbott, Mrs. Stanton (Rosa Hunt):   1                3rd Qu.:39.00
##  Abelson, Mr. Samuel                :   1                Max.   :80.00
##  (Other)                            :1301                NA's   :263
##      sibSp            parch            ticket          fare
##  Min.   :0.0000   Min.   :0.000   CA. 2343:  11   Min.   :  0.000
##  1st Qu.:0.0000   1st Qu.:0.000   1601    :   8   1st Qu.:  7.896
##  Median :0.0000   Median :0.000   CA 2144 :   8   Median : 14.454
##  Mean   :0.4989   Mean   :0.385   3101295 :   7   Mean   : 33.295
##  3rd Qu.:1.0000   3rd Qu.:0.000   347077  :   7   3rd Qu.: 31.275
##  Max.   :8.0000   Max.   :9.000   347082  :   7   Max.   :512.329
##                                   (Other) :1261   NA's   :1
##              cabin      embarked
##                 :1014    :  2
##  C23 C25 C27    :   6   C:270
##  B57 B59 B63 B66:   5   Q:123
##  G6             :   5   S:914
##  B96 B98        :   4
##  C22 C26        :   4
##  (Other)        : 271
```

Resumen estadístico del dataset 'df.csv'

Ya podemos comprobar algunas cosas en esta imagen, que en el siguiente apartado tendremos que valorar y darle una respuesta.

Por ejemplo, que la variable 'age' tiene 263 registros con valor 'NA'. También la variable 'fare' tiene 1 registro con valor 'NA'. Que la variable 'cabin' tiene 1014 registros vacíos o que la variable 'fare' tiene algún registro con valor 0. Recordamos que la variable 'fare' era el precio del pasaje.

3. Limpieza de los datos

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

Los datos sí que contienen elementos vacíos y valores 0 que no podemos tomar como ciertos. Después de realizar este cambio (0→NA). Tenemos, las siguientes cantidades de NAs por variable:

passengerId	survived	pclass	name	sex	age	sibSp	parch
0	0	0	0	0	263	0	0
ticket	fare	cabin	embarked				
0	18	1014	2				

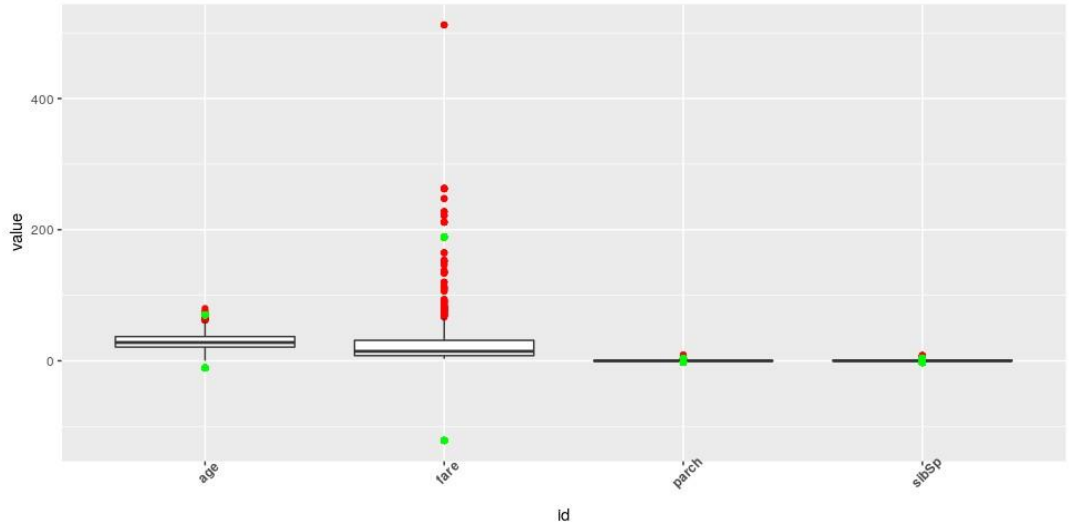
Por lo tanto, podemos observar que tenemos NAs en las variables *age*, *fare*, *cabin* y *embarked*. Hay una variable que nos llama mucho la atención, hablamos de la variable *cabin* que tiene 1014 NAs de los 1309 registros que tiene nuestra base. Por lo tanto, al ser una variable tan pobre en la información que nos da, decidimos suprimirla.

Para imputar el resto de valores faltantes utilizaremos el algoritmo de K-vecinos más próximos siendo K=2. Esto lo implementaremos con el comando knnImputation de la librería DmwR de R. Esta función nos escala las variables para que el resultado sea el correcto. Este método consiste en clasificar los valores más cercanos (vecinos) al valor perdido y luego imputar ese valor o si tenemos más de uno haciendo la media de los valores o mediana. Podemos comprobar que no tenemos valores perdidos.

```
> anyNA(df)
[1] FALSE
```

3.2 Identificación y tratamiento de valores extremos

En segunda instancia observaremos si tenemos valores extremos o no. Para ello, utilizaremos las cajas de diagrama. Los puntos rojos nos indicaran los valores extremos que tiene cada variable, por otro lado los puntos verdes nos indicaran el valor que toma el punto de la media +- 3 veces la desviación estándar.



Por lo que vemos, tenemos algunos posibles valores extremos en la variable 'fare' por la parte de arriba. Podríamos pensar que esos valores son valores falsos. Pero revisando un poco la historia del Titanic (https://www.lne.es/internacional/2015/04/15/10-curiosidades-hundimiento-titanic/1741698.html) nos damos cuenta que estos valores sí que podían ser. Por lo tanto, no los consideraremos como valores extremos.

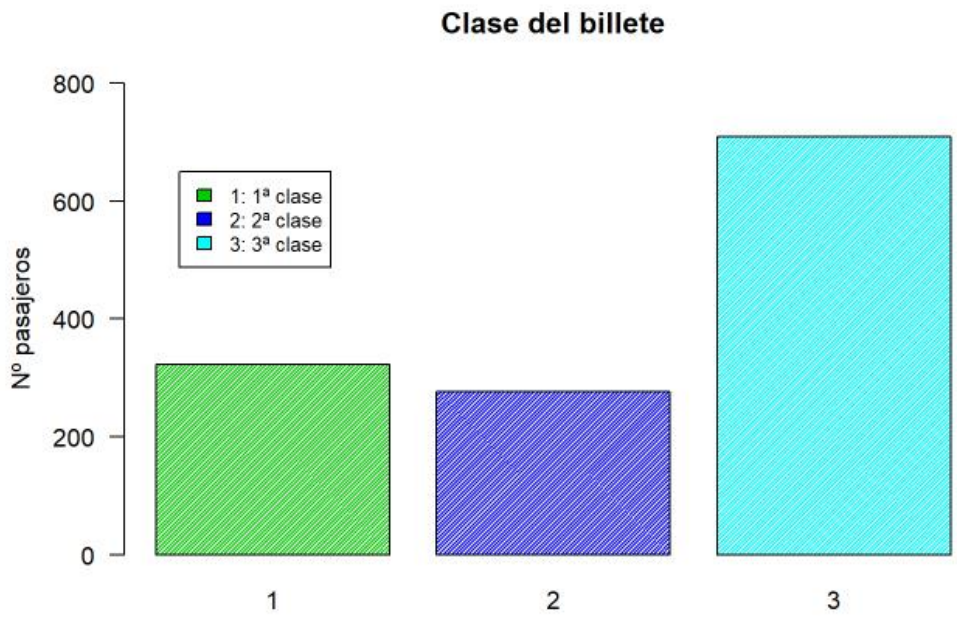
4. Análisis de los datos

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

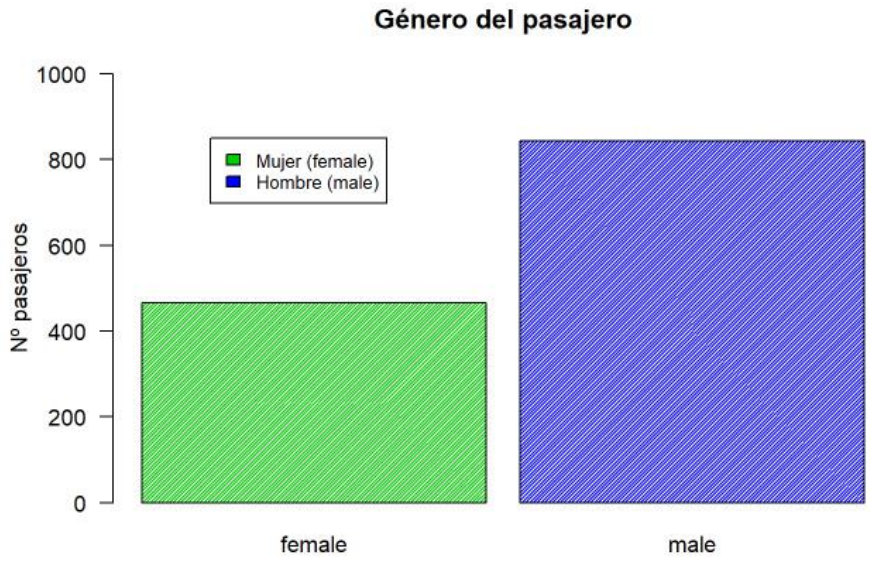
Una vez limpiado los datos, para realizar las pruebas estadísticas, vamos a seleccionar aquellas variables que tiene cierta significación y pueden resultar buenas variables explicativas.

'*El análisis o exploración de los datos tiene como objetivo explicar las principales características de los mismos, para así tratar de responder a las preguntas planteadas en el marco de un proyecto de datos*'. [1]

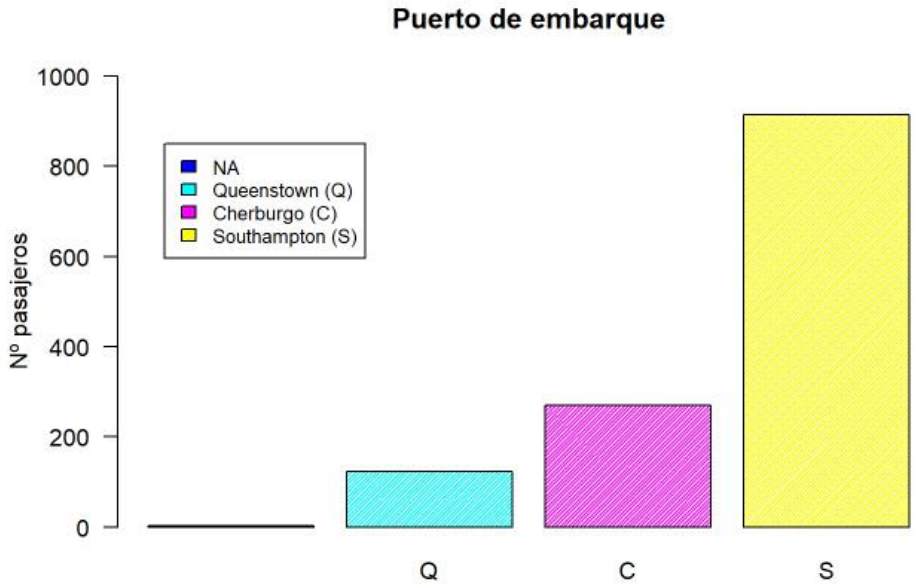
Veamos primero gráficamente cómo se comportan algunas variables.



Distribución de la variable 'pclass'



Distribución de la variable 'sex'



Distribución de la variable 'embarked'

Tal como podemos comprobar en los gráficos, la clase mayoritaria es la tercera. Prácticamente es igual a la suma de las otras dos clases juntas. El género masculino casi duplica al femenino. El puerto de embarque desde donde más pasajeros subieron al barco fue el puerto inglés de Southampton. Finalmente en cuanto a la distribución de la edad, el rango de edades comprendido entre los 20 y los 30 años es el mayoritario.

Las variables que hemos seleccionado están divididas en dos grupos:

Variables cuantitativas: 'age', 'sibSp', 'parch', y 'fare'.

Variables cualitativas: 'pclass', 'sex', 'embarked', y 'survived'.

4.2 Comprobación de la normalidad y homogeneidad de la varianza

En estadística para comprobar la normalidad de una variable podemos usar diferentes herramientas como son el histograma, los gráficos cuantil cuantil (QQplot) o realizar pruebas de hipótesis.

En este caso, hemos usado una de las pruebas más utilizadas y eficientes como es el test de Shapiro-Wilk.

Para responder al supuesto de normalidad, vamos a establecer una hipótesis nula, en el que afirmamos que la muestra proviene de una distribución normal y una hipótesis alternativa, donde la muestra no proviene de una distribución normal. El nivel de significancia con el que trabajaremos será un alfa igual a 0.05 (es decir, del 5%). El criterio de decisión será, si p-valor es menor que alfa, rechazaremos la hipótesis nula. En caso contrario, no se rechazará.

Comenzaremos estudiando la variable 'age'

```
        Shapiro-Wilk normality test

data:  df$age
W = 0.97718, p-value = 1.551e-13
```

Como p-valor es menor que 0.05 (nuestro nivel de significación – alfa = 5%) rechazamos la hipótesis nula en favor de la hipótesis alternativa.

Otra variable podría ser 'fare'

```
        Shapiro-Wilk normality test

data:  df$fare[1:891]
W = 0.51849, p-value < 2.2e-16
```

También en este caso debemos rechazar la hipótesis nula, en favor de la hipótesis alternativa.

Veremos si tiene una distribución normal la variable 'sibSp'

```
        Shapiro-Wilk normality test

data:  df$sibSp[1:891]
W = 0.51297, p-value < 2.2e-16
```

Observamos que la variable tiene un p-valor menor que el nivel de significación que es 0.05, por tanto, rechazamos la hipótesis nula. Concluimos que no tiene una distribución normal.

Por último para la variable 'parch', tenemos que:

```
        Shapiro-Wilk normality test

data:  df$parch[1:891]
W = 0.53281, p-value < 2.2e-16
```

Vemos que tampoco tiene una distribución normal, por rechazar la hipótesis nula.

De todas formas, aun habiendo visto estos datos, sabemos que la media de una muestra de cualquier conjunto de datos es cada vez más normal a medida que aumenta la cantidad de observaciones [1]. A medida que aumentamos la muestra N, la distribución de la media se parece más a una distribución normal. Esto es lo que nos dice el teorema central del límite y es el que aplicaremos para nuestro caso.

Al tener más de 891 registros la muestra podemos aplicar el teorema central del límite y considerar que los datos siguen una distribución normal.

Podemos usar otros métodos para estudiar la normalidad en la distribución de nuestra muestra. Por ejemplo podemos usar la prueba de Anderson-Darling.

Comenzamos estudiando la normalidad en la variable 'age'.

```
        Anderson-Darling normality test

data:  df$age[1:891]
A = 5.6351, p-value = 6.846e-14
```

Observamos que p-valor es menor que 0.05, por tanto rechazamos la hipótesis nula. Podemos decir que la variable 'age' no sigue una distribución normal.

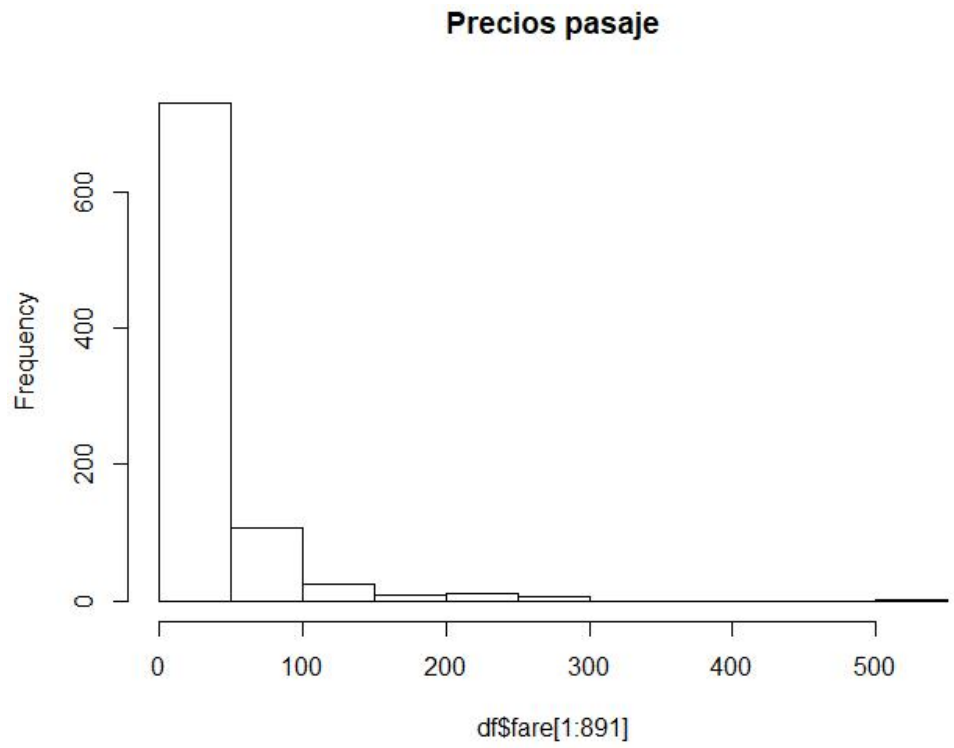
Esta vez estudiaremos la variable 'fare'.

```
        Anderson-Darling normality test

data:  df$fare[1:891]
A = 123.11, p-value < 2.2e-16
```

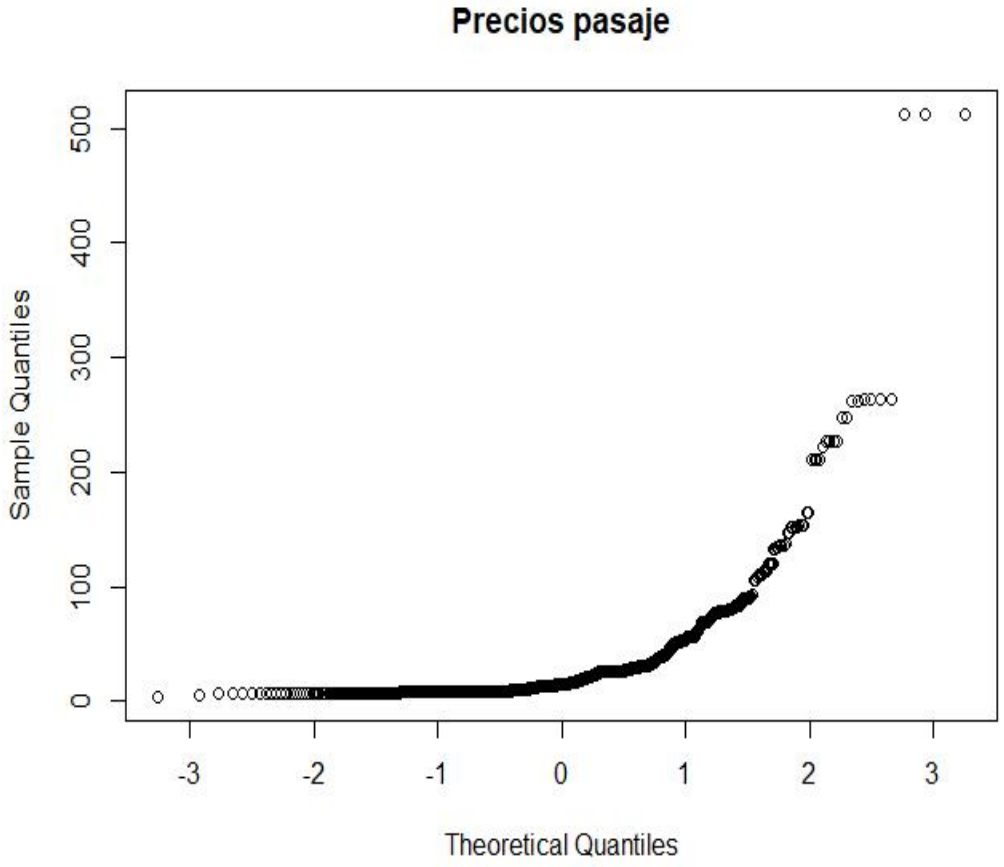
Comprobamos que p-valor es menor que el nivel de significancia, por lo que volvemos a rechazar la hipótesis nula. Esta variable tampoco sigue una distribución normal.

Observamos que los resultados son idénticos a los anteriores.

Pondremos un ejemplo de la variable 'fare' calculada gráficamente mediante un histograma y mediante el cuantil cuantil. Es otra forma, esta vez, mediante gráficas de comprobar la normalidad de la distribución de una variable.



Histograma de la variable ‘fare’



Gráfica cuantil cuantil de la variable ‘fare’

A continuación estudiaremos la homogeneidad de la varianza. Si por ejemplo tendríamos muestras con distribuciones normales, hubiese sido mejor utilizar el F-test o el test de Bartlett.

Pero como no es el caso, emplearemos el test de Fligner-Killeen. Es un test no paramétrico que compara las varianzas basándose en la mediana. Es una alternativa cuando no se cumple la condición de normalidad en las muestras.

Vamos a comparar con el test, las varianzas entre las variables 'age – fare'

```
        Fligner-Killeen test of homogeneity of variances

data:  df$age by df$fare
Fligner-Killeen:med chi-squared = 377.88, df = 295, p-value = 0.0007799
```

Entre las variables 'fare – pclass'

```
        Fligner-Killeen test of homogeneity of variances

data:  df$fare by df$pclass
Fligner-Killeen:med chi-squared = 543.12, df = 2, p-value < 2.2e-16
```

Entre las variables 'fare – sex'

```
        Fligner-Killeen test of homogeneity of variances

data:  df$fare by df$sex
Fligner-Killeen:med chi-squared = 82.988, df = 1, p-value < 2.2e-16
```

Entre las variables 'age – sibSp'

```
        Fligner-Killeen test of homogeneity of variances

data:  df$age by df$sibSp
Fligner-Killeen:med chi-squared = 48.176, df = 6, p-value = 1.09e-08
```

Entre variables 'age –pclass'

```
        Fligner-Killeen test of homogeneity of variances

data:  df$age by df$pclass
Fligner-Killeen:med chi-squared = 45.815, df = 2, p-value = 1.126e-10
```

Y finalmente entre las variables 'age – embarked'

```
        Fligner-Killeen test of homogeneity of variances

data:  df$age by df$embarked
Fligner-Killeen:med chi-squared = 11.74, df = 2, p-value = 0.002823
```

Podemos observar cómo en todos los casos el p-valor es inferior a 0.05, nuestro nivel de significación, por lo que podemos asumir que no hay igualdad de varianzas entre los

diferentes grupos que hemos comparado. Podemos decir que presentan varianzas estadísticamente diferentes al rechazar la hipótesis nula de homocedasticidad.

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.

Cuando la normalidad y la homocedasticidad se cumplen se podrían aplicar pruebas de contraste de hipótesis de tipo paramétrico como la prueba de t de Student.

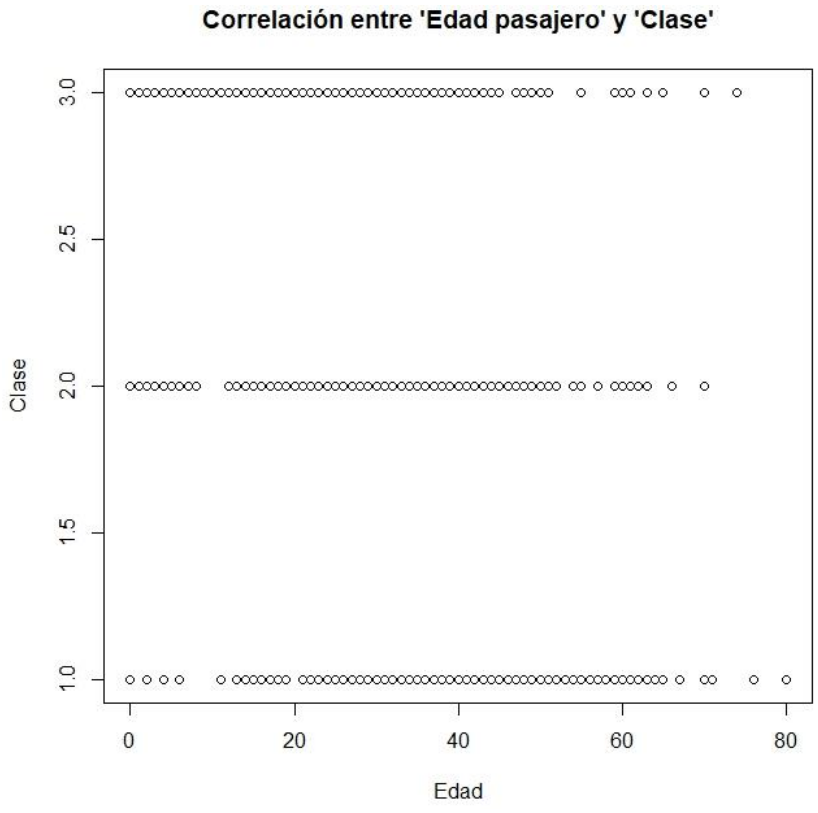
A continuación veremos si encontramos algunos signos de correlación entre las variables y de qué tipo puede ser.

Primero crearemos una matriz de correlación entre las diferentes variables cuantitativas que habíamos seleccionado.

	age	pclass	sibSp	parch	fare
age	1.0000000	-0.42397243	-0.24126789	-0.1610111	0.1823653
pclass	-0.4239724	1.00000000	0.06083201	0.0183222	-0.5626310
sibSp	-0.2412679	0.06083201	1.00000000	0.3735872	0.1571123
parch	-0.1610111	0.01832220	0.37358719	1.0000000	0.2186555
fare	0.1823653	-0.56263104	0.15711231	0.2186555	1.0000000

Podemos observar que las correlaciones más altas (positivas o negativas) es entre los pares de variables ‘age - pclass’, ‘pclass – fare’, ‘parch – sibSp’

Gráficamente podríamos verlo así



En este caso vemos como los pasajeros de menor edad se sitúan en la clase 2 y 3, mientras que los pasajeros con edad más avanzada estarían en la clase 1.

Tiene una relación negativa y tiene su lógica. Si los pasajeros mayores tienen más recursos económicos, es normal pensar que pudiesen comprar los billetes más caros, que deberían de corresponder con los de primera clase. Tampoco es que sea muy significativa la relación entre ambas variables ya que no llega ni al 0.4.

Podemos observar la correlación entre el precio del pasaje y el puerto de embarque

```
        Pearson's product-moment correlation

data:  df$fare[1:891] and as.numeric(df$embarked[1:891])
t = -6.8225, df = 889, p-value = 1.653e-11
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2845644 -0.1597182
sample estimates:
       cor
-0.2230557
```

Hay una relación negativa. A mayor precio del pasaje, parece ser que el pasajero debió de acceder al barco por el puerto francés de Cherbourng. Los pasajeros que accedieron por el puerto inglés de Southampton tenían los billetes más baratos (de hecho la mayoría eran inmigrantes que no contaban con muchos recursos económicos). El valor de la relación entre variables sigue dándonos bajo, no llega al 0.23.

Por último vamos a aplicar una prueba estadística mediante una regresión lineal, que es un modelo matemático que tiene como objetivo aproximar la relación de dependencia lineal entre una variable dependiente y una o varias, variables independientes.

En nuestro caso será un modelo de regresión lineal múltiple, con el que poder analizar cómo se relacionan las variables ‘age’, ‘pclass’, ‘sex’, ‘embarked’, ‘fare’, ‘sibsp y ‘parch’ que son las variables independientes o explicativas y la variable dependiente ‘survived’ que es la variable objetivo. Usamos la función ‘lm’ (linear model) de R.

```
lm(formula = survived ~ age + pclass + sex + fare + embarked +
    sibSp + parch, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.08272 -0.14760 -0.05735  0.14588  1.01134

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.2545526  0.0545378  23.003  < 2e-16 ***
age         -0.0046048  0.0007889  -5.837 6.72e-09 ***
pclass      -0.1245633  0.0152217  -8.183 6.52e-16 ***
sexmale     -0.6652000  0.0204190 -32.578  < 2e-16 ***
fare         0.0003041  0.0002327   1.306 0.191621
embarkedQ    0.0262801  0.0389256   0.675 0.499709
embarkedS   -0.0222309  0.0244480  -0.909 0.363352
sibSp       -0.0383777  0.0099307  -3.865 0.000117 ***
parch       -0.0117674  0.0121223  -0.971 0.331867
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3355 on 1300 degrees of freedom
Multiple R-squared:  0.5243,    Adjusted R-squared:  0.5214
F-statistic: 179.1 on 8 and 1300 DF,  p-value: < 2.2e-16
```

Para determinar qué variables son las más significativas a la hora de tratar de explicar la variable dependiente, deberemos de atender al valor de p-valor, que en el modelo son: 'age', 'pclass', 'sex' – male, 'sibSp' y que son las que tienen las tres estrellas.

El valor de R^2 (R-squared), que es una medida de calidad del modelo y que toma valores entre 0 y 1, nos da 0.524. Es decir el modelo es capaz de explicar el 52.4% de la variabilidad observada. No es un valor excesivamente alto para el coeficiente de determinación.

Seguramente para obtener unos mejores resultados deberíamos de quitar aquellas variables que el modelo nos da como poco significativas y tienen los p-valor altos (mayor que el nivel de significación, alfa = 0.05).

Ahora plantearemos una regresión logística múltiple. Esta, es una extensión de la regresión logística simple. Explicaremos brevemente en que consiste este método. Este método fue desarrollado por David Cox en 1958, es un método de regresión que permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa. Una de las principales aplicaciones de la regresión logística es la de clasificación binaria, en el que las observaciones se clasifican en un grupo u otro dependiendo del valor que tome la variable empleada como predictor.

Para poder hacer una regresión logística múltiple primero deberemos observar las variables que queremos meter en el modelo múltiple cómo se comportan en simple con la variable predictora. Para poder meterlos el p-valor deberá ser <0.2 (para que sea significativa debe ser <0.05).

Por lo tanto, lo primero que haremos será seleccionar las variables que nos interesan: PassengerId, pclass, sex, age, sibSp, parch, fare, embarked. Esto es, las variables name y ticket no nos interesan para este modelo.

Ahora tal como hemos dicho analizaremos cómo se comportan están variables en la regresión logística simple. Para esto, separaremos las variables cualitativas y cuantitativas.

Empezaremos por las cualitativas.

- Sex

```
> survi_sex<-glm(survived~sex, family=binomial,data=df)
> summary(survi_sex) # Null deviance: 1735.1, Residual deviance: 1079.7

Call:
glm(formula = survived ~ sex, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8707  -0.5262  -0.5262   0.6180   2.0227

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.5588     0.1222   12.75   <2e-16 ***
sexmale      -3.4660     0.1596  -21.71   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.1  on 1308  degrees of freedom
Residual deviance: 1079.7  on 1307  degrees of freedom
AIC: 1083.7

Number of Fisher Scoring iterations: 4

> pchisq(1735.1-1079.7,df=1,lower.tail=FALSE)
[1] 1.495288e-144
```

Es significativa ya que p-value tiene un valor de 1.495288e-144, menor que 0.05

- embarked

```
> survi_embarked<-glm(survived~embarked, family=binomial,data=df)
> summary(survi_embarked) # Null deviance: 1735.1, Residual deviance: 1710.0

Call:
glm(formula = survived ~ embarked, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1680  -0.9011  -0.9011   1.2751   1.4816

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.02214    0.12150  -0.182    0.855
embarkedQ   -0.20463    0.21780  -0.940    0.347
embarkedS   -0.66937    0.14029  -4.771 1.83e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.1  on 1308  degrees of freedom
Residual deviance: 1710.0  on 1306  degrees of freedom
AIC: 1716

Number of Fisher Scoring iterations: 4

> pchisq(1735.1-1710.0,df=2,lower.tail=FALSE)
[1] 3.544902e-06
>
```

Es significativa ya que p-value tiene un valor de 3.544902e-06

- pclass (esta variable la tomaremos como factor).

```
> survi_pclass<-glm(survived~pclass, family=binomial,data=df)
> summary(survi_pclass) # Null deviance: 1735.1, Residual deviance: 1643.8

Call:
glm(formula = survived ~ pclass, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3097  -0.7923  -0.7923   1.0506   1.6196

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.3058     0.1126   2.716 0.006611 **
pclass2      -0.6188     0.1657  -3.733 0.000189 ***
pclass3      -1.3035     0.1409  -9.254  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.1  on 1308  degrees of freedom
Residual deviance: 1643.8  on 1306  degrees of freedom
AIC: 1649.8

Number of Fisher Scoring iterations: 4

> pchisq(1735.1-1643.8,df=2,lower.tail=FALSE)
[1] 1.494366e-20
```

Es significativa ya que p-value tiene un valor de 1.494366e-20.

- sibSp

```
> survi_sibSp<-glm(survived~sibSp, family=binomial,data=df)
> summary(survi_sibSp) # Null deviance: 1735.1, Residual deviance: 1690.2

Call:
glm(formula = survived ~ sibSp, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2122  -0.9024  -0.9024   1.2595   2.0963

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.68810    0.07101  -9.691  < 2e-16 ***
sibSp1       0.76965    0.13267   5.801 6.59e-09 ***
sibSp2       0.49705    0.31804   1.563    0.118
sibSp3      -0.41051    0.52126  -0.788    0.431
sibSp4      -0.81598    0.55731  -1.464    0.143
sibSp5      -0.92134    1.09774  -0.839    0.401
sibSp8      -1.39134    1.06301  -1.309    0.191
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.1  on 1308  degrees of freedom
Residual deviance: 1690.2  on 1302  degrees of freedom
AIC: 1704.2

Number of Fisher Scoring iterations: 4

> pchisq(1735.1-1690.2,df=6,lower.tail=FALSE)
[1] 4.899295e-08
```

Es significativa ya que p-value tiene un valor de 4.899295e-08.

- Parch.

```
> survi_parch<-glm(survived~parch, family=binomial,data=df)
> summary(survi_parch) # Null deviance: 1735.1, Residual deviance: 1689.6

Call:
glm(formula = survived ~ parch, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.4006  -0.8972  -0.8972   1.4864   1.8930

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.702143   0.067116 -10.462  < 2e-16 ***
parch1        0.890937   0.168060   5.301 1.15e-07 ***
parch2        0.826195   0.200098   4.129 3.64e-05 ***
parch3        1.212968   0.733374   1.654   0.0981 .
parch4        0.008996   0.868622   0.010   0.9917
parch5       -0.907295   1.097499  -0.827   0.4084
parch6      -12.863924 378.592874  -0.034   0.9729
parch9        0.702143   1.415805   0.496   0.6199
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.1  on 1308  degrees of freedom
Residual deviance: 1689.6  on 1301  degrees of freedom
AIC: 1705.6

Number of Fisher Scoring iterations: 12

> pchisq(1735.1-1689.6,df=7,lower.tail=FALSE)
[1] 1.093564e-07
```

Es significativa ya que p-value tiene un valor de 1.093564e-07.

Por lo tanto, todas nuestras variables cualitativas las podremos introducir en el modelo múltiple.

Ahora empezaremos con las variables cuantitativas.

- age.

```
> survi_age<-glm(survived~age, family=binomial,data=df)
> summary(survi_age)

Call:
glm(formula = survived ~ age, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0627  -0.9867  -0.9426   1.3741   1.5693

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.276034   0.137082  -2.014   0.0440 *
age         -0.007626   0.004257  -1.791   0.0732 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.1  on 1308  degrees of freedom
Residual deviance: 1731.9  on 1307  degrees of freedom
AIC: 1735.9

Number of Fisher Scoring iterations: 4
```

Como podemos observar el p-valor es 0.0732, por lo tanto, no sería significativa (p>0.05). Pero para el modelo múltiple sí que lo meteremos ya que el p-valor es menor de 0.2 tal como hemos afirmado antes.

- fare:

```
> survi_fare<-glm(survived~fare, family=binomial,data=df)
> summary(survi_fare)

Call:
glm(formula = survived ~ fare, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1155  -0.8936  -0.8656   1.3656   1.5296

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.877477   0.075103 -11.684  < 2e-16 ***
fare         0.011415   0.001533   7.447 9.54e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1735.1  on 1308  degrees of freedom
Residual deviance: 1658.7  on 1307  degrees of freedom
AIC: 1662.7

Number of Fisher Scoring iterations: 4
```

Como podemos observar el p-valor es 9.54e-14, por lo tanto, es significativo.

Por lo tanto, para el modelo logístico múltiple utilizaremos todas las variables que tenemos excepto las variables name y ticket. Para luego optimizar el modelo que tengamos.

Haremos un primer modelo con todas las variables que hemos seleccionado, para observar cómo reaccionan las variables.

```
> modelo1<-glm(survived~pclass+sex+age+sibSp+parch+fare+embarked,family=binomial,data=df)
> # summary(modelo1)
> drop1(modelo1,test="Chi")
Single term deletions

Model:
survived ~ pclass + sex + age + sibSp + parch + fare + embarked
         Df Deviance     AIC    LRT  Pr(>Chi)
<none>        936.69  978.69
pclass    2   986.37 1024.37  49.68 1.628e-11 ***
sex       1  1506.76 1546.76 570.07 < 2.2e-16 ***
age       1   959.72  999.72  23.04 1.590e-06 ***
sibSp     6   962.96  992.96  26.27 0.0001982 ***
parch     7   945.65  973.65   8.97 0.2551725
fare      1   937.79  977.79   1.11 0.2924732
embarked  2   938.06  976.06   1.38 0.5027861
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Por lo tanto, como podemos observar tenemos algunas variables que no nos dicen mucho, exactamente las variables ‘parch’, ‘fare’ y ‘embarked’. Por lo tanto, hacemos un segundo modelo sin estas variables y comprobamos si el segundo modelo es mejor. Para esta comprobación hacemos una ANOVA.

```
> modelo2<-glm(survived~pclass+sex+age+sibSp,family=binomial,data=df)
> # summary(modelo2)
> drop1(modelo2,test="Chi")
Single term deletions

Model:
survived ~ pclass + sex + age + sibSp
       Df Deviance     AIC    LRT  Pr(>Chi)
<none>      948.34  970.34
pclass  2  1045.58 1063.58  97.24 < 2.2e-16 ***
sex     1  1569.69 1589.69 621.34 < 2.2e-16 ***
age     1   980.26 1000.26  31.92 1.607e-08 ***
sibSp   6   976.63  986.63  28.29 8.285e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> anova(modelo1,modelo2,test="Chi")
Analysis of Deviance Table

Model 1: survived ~ pclass + sex + age + sibSp + parch + fare + embarked
Model 2: survived ~ pclass + sex + age + sibSp
  Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
1      1288     936.69
2      1298     948.34 -10  -11.658   0.3086
```

Como el p-valor es igual a 0.3086, nos quedamos con este segundo modelo

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

Testearemos el modelo que hemos creado ahora. Para ello, crearemos un conjunto de datos de entrenamiento y otro de test.

Lo primero que haremos será desordenar la base. Como podemos observar estamos utilizando el comando set.seed(). Pero, ¿Para qué sirve ese comando? Pues su función será la de que la aleatorización de los elementos sea reproducible. Así, podremos reproducir los resultados que consigamos.

Para la evaluación del árbol de decisión que queremos crear deberemos de crear dos conjuntos de datos. Uno de entrenamiento para generar un modelo predictivo y el otro de prueba, para comprobar la eficacia de este modelo para hacer predicciones correctas. Normalmente para el del entrenamiento se le dan 2/3 de la base y al segundo el resto.

```
set.seed(4)
df_random <- df_tes[sample(nrow(df_tes)),]

y <- df_random[,5]
X <- df_random[,1:4]

indexes = sample(1:nrow(df_tes), size=floor((2/3)*nrow(df_tes)))
trainX<-X[indexes,]
trainy<-y[indexes]
testX<-X[-indexes,]
testy<-y[-indexes]
```

Después de esto crearemos el modelo de predicción.

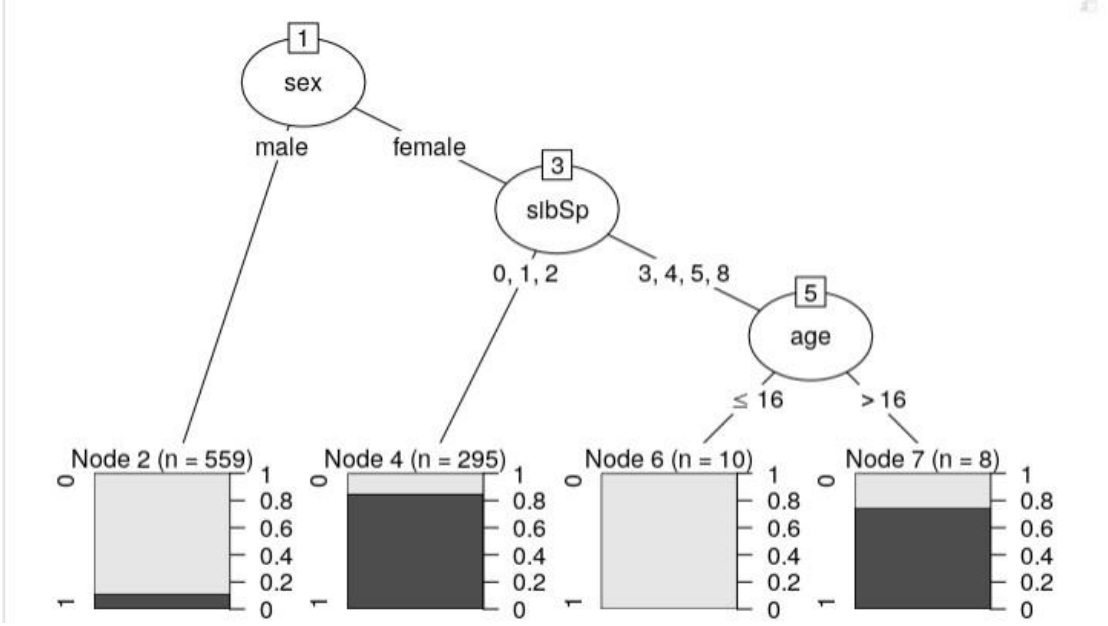
```
> arbol_1 <- C50::C5.0(trainX, trainy,rules=TRUE )
> summary(arbol_1)

Call:
C5.0.default(x = trainX, y = trainy, rules = TRUE)


C5.0 [Release 2.07 GPL Edition]         Sun May 31 17:30:46 2020
-------------------------------

Class specified by attribute `outcome'

Read 872 cases (5 attributes) from undefined.data
```

```
Rules:

Rule 1: (29/1, lift 1.5)
        age <= 16
        sibSp in {3, 4, 5, 8}
        ->  class 0  [0.935]

Rule 2: (559/64, lift 1.4)
        sex = male
        ->  class 0  [0.884]

Rule 3: (295/46, lift 2.3)
        sex = female
        sibSp in {0, 1, 2}
        ->  class 1  [0.842]

Rule 4: (266/42, lift 2.3)
        sex = female
        age > 16
        ->  class 1  [0.840]
```

- Rule 1 nos indica que si age <= 16 o sibSp dentro de {3, 4, 5, 8}. Entonces, no sobrevive. Validez: 94%.
- Rule 2 nos indica que si sex = male. Entonces, no sobrevive. Validez: 88%.
- Rule 3 nos indica que si sex = female o sibSp dentro de {0, 1, 2}. Entonces, sobrevive. Validez: 84%.
- Rule 4 nos indica que si sex = female o age > 16. Entonces, sobrevive. Validez: 84%.

Lo dicho, ahora lo podemos representar gráficamente.

```
arbol_2 <- C50::C5.0(trainX, trainy)
plot(arbol_2)
```



Hemos conseguido un modelo en base al subconjunto que habíamos creado para entrenamiento. A la vez que este habíamos creado otro subconjunto que lo íbamos a utilizar para comprobar la calidad del modelo. Esto lo haremos prediciendo la severidad del subconjunto de prueba.

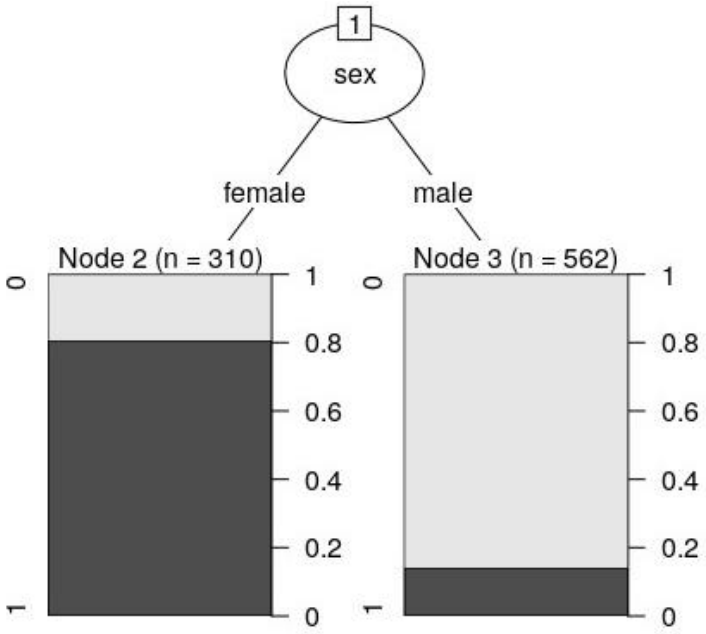
```
> confusionMatrix(predicted_model, testy)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 241  48
         1  21 127

               Accuracy : 0.8421
                 95% CI : (0.8045, 0.875)
    No Information Rate : 0.5995
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6625

 Mcnemar's Test P-Value : 0.001748

            Sensitivity : 0.9198
            Specificity : 0.7257
         Pos Pred Value : 0.8339
         Neg Pred Value : 0.8581
             Prevalence : 0.5995
         Detection Rate : 0.5515
   Detection Prevalence : 0.6613
      Balanced Accuracy : 0.8228

       'Positive' Class : 0
```

Obtenemos una exactitud (Accuracy) del 84%. Respecto de la severidad, supervivencia nuestro modelo tiene una sensibilidad del 91%. Esto es, somos capaces de predecir el 91% de las personas que no van a sobrevivir.

Ahora lo que haremos será hacer un nuevo modelo con sets de entrenamiento y de test distintos, para comprobar si se mantiene lo conseguido ahora

```
set.seed(5)
df_random1 <- df_tes[sample(nrow(df_tes)),]

y1 <- df_random1[,5]
X1 <- df_random1[,1:4]

indexes = sample(1:nrow(df_tes), size=floor((2/3)*nrow(df_tes)))
trainX1<-X1[indexes,]
trainy1<-y1[indexes]
testX1<-X1[-indexes,]
testy1<-y1[-indexes]


arbol_11 <- C50::C5.0(trainX1, trainy1,rules=TRUE )
arbol_21 <- C50::C5.0(trainX1, trainy1)
plot(arbol_21)
```

Consiguiendo esta gráfica,



Como antes, vamos a comprobar la calidad,

```
> predicted_model1 <- predict( arbol_11, testX1, type="class" )
> confusionMatrix(predicted_model1, testy1)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 251  30
         1  21 135

               Accuracy : 0.8833
                 95% CI : (0.8494, 0.9119)
    No Information Rate : 0.6224
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.749

 Mcnemar's Test P-Value : 0.2626

            Sensitivity : 0.9228
            Specificity : 0.8182
         Pos Pred Value : 0.8932
         Neg Pred Value : 0.8654
             Prevalence : 0.6224
         Detection Rate : 0.5744
   Detection Prevalence : 0.6430
      Balanced Accuracy : 0.8705

       'Positive' Class : 0
```

En este nuevo caso la predicción es prácticamente la misma que en la anterior y lo mismo ocurre con la calidad. Pero debemos resaltar que el este modelo es bastante diferente respecto al anterior modelo en lo que se refiere a las particiones.

6. Resolución del problema

6.1 A partir de los resultados obtenidos. ¿Cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Una de las cuestiones que se planteaba al inicio de este trabajo, era responder a la pregunta de qué tipo de pasajeros tenía más probabilidad de poder sobrevivir al hundimiento del barco, es decir, queríamos predecir qué personas en base a ciertas características podrían sobrevivir.

Para responder a esto, lo primero que realizamos fue una labor de limpieza de los datos, tomando decisiones sobre qué hacer con variables que tenían elementos vacíos o identificando valores extremos. Después analizamos los datos, estudiando su significancia, el peso y su importancia, en cuanto a si eran o no variables explicativas que contribuyesen a mejorar el modelo. Para ello realizamos diferentes pruebas estadísticas, estudios de análisis de correlación entre variables, contrastes de hipótesis, regresiones, etc.

Finalmente elegimos aquellas variables candidatas ('pclass', 'age', 'sex' y 'sibsp') que entendíamos que eran las mejores para crear nuestro modelo.

Hemos planteado un par de modelos para poder comparar los resultados. Los modelos que hemos desarrollado son capaces de predecir con una exactitud (Accuracy) de entre el 84% y 88%. Así que podemos afirmar como conclusión, que sí somos capaces de responder a la cuestión inicial.

¿Y qué tipo de persona es?, ¿qué características tiene la persona que sobrevive? La persona con más probabilidad de sobrevivir al hundimiento del Titanic era una mujer, mayor de 16 años y con un número de hermanos o cónyuges entre 0 y 2.

7. Código en R de la práctica y tabla de contribuciones

Adjuntamos el enlace Github donde se puede acceder al código y al documento PDF

https://github.com/eperezbal/Practica2_limpieza_analisis_datos

Tabla de contribuciones al trabajo por parte del equipo.

Contribuciones	Firma
Investigación previa	OA, EP
Redacción de las respuestas	OA, EP
Desarrollo código	OA, EP

8. Bibliografía

[1] Calvo M., Subirats L., Pérez D. (2019). *Introducción a la limpieza y análisis de los datos*. Editorial UOC.

[2] Megan Squire (2015). *Clean Data*. Packt Publishing Ltd.

[3] Peter Dalgaard (2008). *Introductory Statistics with R*. Springer Science & Business Media.

[4] Joaquín Amat. *Regresión logística simple y múltiple*. [En línea]. Última actualización: Agosto de 2016. Disponible en: https://www.cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple#regresi%C3%B3n_log%C3%ADstica_simple [Consultado en: mayo 2020]