Tipología y ciclo de vida de los datos: Práctica 2

Autor: Nombre estudiante

5 de junio 2021

Contents

Descripción del dataset

El conjunto de datos del titanic es ampliamente concido en la comunidad del ML. Es más, forma parte de los retos de iniciación en la plataforma kaggle.

Este conjunto de datos, es la representación de las personas que embarcaron en el titanic. En el, se recogen multitud de datos sobre cada persona, relativos a su edad, pais y clase en la que embarcaron, además de si sobrevivieron o no.

¿Por qué es importante?

Este cojunto de datos tiene la posibilidad de explicar algunos datos de la catastrofe. Puede aclarar si hubo algún condicionante para la muerte o supervivencia de las personas más allá del puro azar.

¿Qué pregunta pretende responder?

Este cojunto de datos pretende elaborar un modelo predictivo y con el responder a la pregunta: ¿Qué tipo de personas tenían más probabilidades de sobrevivir?

Integración de los datos de interés a analizar

Antes de integrar, vamos a describir las variables que caracterízan a estos datos:

- **PassengerId** Id del pasajero
- **Name** Nombre del pasajero en formato cadena.
- **Sex** Factor con los niveles male/female.
- **Age** Valor numérico con la edad.
- **Pclass** Factor con la clase del ticket(1 = 1st, 2 = 2nd, 3 = 3rd)
- **Embarked** Factor con el embarque de la persona(C = Cherbourg, Q = Queenstown, S = Southampton).
- **Cabin** camarote
- **Fare** Valor numérico con el precio del ticket (NA para miembros de la tripulación musicos y empleados de la compañia).
- **SibSp** Número de conyuges a bordo.
- **Parch** Número de padres, hijos a bordo.
- **Survived** Factor con dos niveles indicando si sobrevivió o no(0 = No, 1 = Yes).

Escogemos las columnas que serán factores:

```
factors = c("Sex"="factor","Pclass"="factor","Cabin"="factor", "Embarked"="factor","Survived"="factor")
```

Cargamos los datos:

```
titanicData <- read.csv('titanic.csv', stringsAsFactors = FALSE, colClasses = factors );
```

```
summary(titanicData);
```

```
##   PassengerId    Survived Pclass      Name              Sex
##  Min.   :  1.0   0:549    1:216   Length:891         female:314
##  1st Qu.:223.5   1:342    2:184   Class :character   male  :577
##  Median :446.0            3:491   Mode  :character
##  Mean   :446.0
##  3rd Qu.:668.5
##  Max.   :891.0
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare                Cabin     Embarked
##  Min.   :  0.00             :687    :  2
##  1st Qu.:  7.91   B96 B98   :  4   C:168
##  Median : 14.45   C23 C25 C27:  4   Q: 77
##  Mean   : 32.20   G6        :  4   S:644
##  3rd Qu.: 31.00   C22 C26   :  3
##  Max.   :512.33   D         :  3
##                   (Other)   :186
```

Los datos de interés de ese cojunto de datos serán los que nos aporten algo de información sobre las personas pero a nivel de cojunto, es decir datos como nombre, identificador de pasajero o número de ticket no nos resultan de utilidad. Por lo que podemos crear un conjunto de datos solamente con los datos adecuados:

```
cols_remove <- c("PassengerId", "Name", "Ticket")
titanicData <- titanicData[, !(colnames(titanicData) %in% cols_remove)]
summary(titanicData);
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  1:342    2:184   male  :577   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##           3:491                Median :28.00   Median :0.000   Median :0.0000
##                                Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                                3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                                NA's   :177
##       Fare                Cabin     Embarked
##  Min.   :  0.00             :687    :  2
##  1st Qu.:  7.91   B96 B98    :  4   C:168
##  Median : 14.45   C23 C25 C27:  4   Q: 77
##  Mean   : 32.20   G6         :  4   S:644
##  3rd Qu.: 31.00   C22 C26    :  3
##  Max.   :512.33   D          :  3
##                   (Other)    :186
```

Limpieza de los datos.

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

Para comprobar si los datos contienen elementos vacíos se ejecuta la siguiente sentencia.

```
colSums(is.na(titanicData))
```

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

Se puede observar que la columna Age contiene 177 valores nulos. Existen diferentes políticas para el tratamiento de los valores nulos:

- **Eliminarlos**: En Ocasiones, compensa eliminar estas filas, ya que pueden generar distorsiones a la hora de hacer cálculos con las columnas que contienen los valores nulos.
- **Reemplazo**: Se podrían reemplazar los valores por la media, la mediana o la moda. Estas medidas se pueden intentar particular en función de otras columnas para que no siempre sean los mismos para todas las entradas nulas.
- **Asignación de una categoría**: Si se discretizan los datos en, por ejemplo, rangos de edad, se puede particularizar todos los valores nulos en una categoría especial llamada "edad desconocida".
- **Predicción de los valores nulos**: Por último, se pueden inferir los valores mediante predicciones.

En este caso, se van a inferir los valores en función de otros parámentros. Para hacer esto, partimos de que es muy probable que la edad media de las personas que viajan en Pclass3 es diferente a la edad media de las personas que viajan en Pclass1. Además, esa edad será diferente en función de si estamos ante un

hombre o una mujer. Por tanto, para inferir los valores de edad perdidos, se agrupa por Sex y Pclass, para posteriormente calcular las medianas de cada serie agrupada.

Primero, se calcula la media y la mediana de edad en función de la clase y el género del pasajero.

```
by_sex_class <- titanicData %>% group_by(Sex, Pclass)  %>% summarise(mean = mean(Age, na.rm = TRUE), me

by_sex_class
```

```
## # A tibble: 6 x 4
## # Groups:   Sex [2]
##   Sex    Pclass  mean median
##   <fct>  <fct>  <dbl>  <dbl>
## 1 female 1       34.6   35
## 2 female 2       28.7   28
## 3 female 3       21.8   21.5
## 4 male   1       41.3   40
## 5 male   2       30.7   30
## 6 male   3       26.5   25
```

Se observa que la media y la mediana de edad varía en función del género y la clase en la que viajaban. Se procede a rellenar los valores nulos en la columna de edad por los valores de mediana en función de Sex y Pclass.

```
titanicData$Age[titanicData$Sex == "female" & titanicData$Pclass == "1" & is.na(titanicData$Age)] <- by

titanicData$Age[titanicData$Sex == "female" & titanicData$Pclass == "2" & is.na(titanicData$Age)] <- by

titanicData$Age[titanicData$Sex == "female" & titanicData$Pclass == "3" & is.na(titanicData$Age)] <- by

titanicData$Age[titanicData$Sex == "male" & titanicData$Pclass == "1" & is.na(titanicData$Age)] <- by_s

titanicData$Age[titanicData$Sex == "male" & titanicData$Pclass == "2" & is.na(titanicData$Age)] <- by_s

titanicData$Age[titanicData$Sex == "male" & titanicData$Pclass == "3" & is.na(titanicData$Age)] <- by_s
```

Se vuelve a comprobar si existen valores nulos.

```
colSums(is.na(titanicData))
```

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

Se han eliminado los valores nulos. Ahora, se comprueba que el summary no difiere mucho del original.

```
summary(titanicData)
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  1:342    2:184   male  :577   1st Qu.:21.50   1st Qu.:0.000   1st Qu.:0.0000
##           3:491                Median :26.00   Median :0.000   Median :0.0000
##                                Mean   :29.11   Mean   :0.523   Mean   :0.3816
##                                3rd Qu.:36.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0000
##
##       Fare                Cabin     Embarked
```

```
##  Min.   :  0.00              :687    :  2
##  1st Qu.:  7.91   B96 B98    :  4   C:168
##  Median : 14.45   C23 C25 C27:  4   Q: 77
##  Mean   : 32.20   G6         :  4   S:644
##  3rd Qu.: 31.00   C22 C26    :  3
##  Max.   :512.33   D          :  3
##                   (Other)    :186
```

La media de edad ha bajado ligeramente, pero en general los datos se mantienen estables aún habiendo inferido 177 entradas.

En el summary se pueden observar diferentes columnas con datos anómalos o vacíos. Se procede a analizar uno a uno cada caso.

La columna SibSp contiene muchos valores a 0. En esta columna este valor es perfectamente normal, ya que indica el número de hermanos o esposas a bordo del barco para cada persona.

La columna Parch también contienen valores a 0, pero también cuadra, ya que este campo indica el núero de padres o hijos a bordo.

La columna Fare indica el precio que el pasajero pagó para estar en el barco. El mínimno de esta columna es 0, lo cual no tendría demasiado sentido. Se procede a mirar cuántas entradas tienen 0 en el precio.

```
filter(titanicData, Fare == 0)
```

```
##    Survived Pclass  Sex Age SibSp Parch Fare Cabin Embarked
## 1         0      3 male  36     0     0    0              S
## 2         0      1 male  40     0     0    0   B94        S
## 3         1      3 male  25     0     0    0              S
## 4         0      2 male  30     0     0    0              S
## 5         0      3 male  19     0     0    0              S
## 6         0      2 male  30     0     0    0              S
## 7         0      2 male  30     0     0    0              S
## 8         0      2 male  30     0     0    0              S
## 9         0      3 male  49     0     0    0              S
## 10        0      1 male  40     0     0    0              S
## 11        0      2 male  30     0     0    0              S
## 12        0      2 male  30     0     0    0              S
## 13        0      1 male  39     0     0    0   A36        S
## 14        0      1 male  40     0     0    0  B102        S
## 15        0      1 male  38     0     0    0              S
```

Es destacable que todas aquellas entradas que tienen Fare = 0 son de hombres. Dado que no han pagado nada, estas personas podrían ser tripulación del barco, por lo que se mantienen estas entradas.

La columna Cabin contiene muchísimos valores nulos, y no se puede inferir de ninguna manera. Tampoco parece que aporte demasiada información útil, por lo que se desecha.

```
titanicData <- subset(titanicData, select = -c(Cabin))
```

Por último, la columna Embarked contiene dos valores nulos que sí serían interesantes de completar. En este caso, para evitar tener 4 categorías y que se distorsionen un poco esos datos, se imputan estos dos valores con la moda, es decir, con el valor "S".

```
titanicData$Embarked[titanicData$Embarked == ""] <- "S"

# Se elimina la clase sobrante.

titanicData$Embarked <- as.factor(as.character(titanicData$Embarked))
```

Se hace un último summary para comprobar que todo ha quedado correctamente.

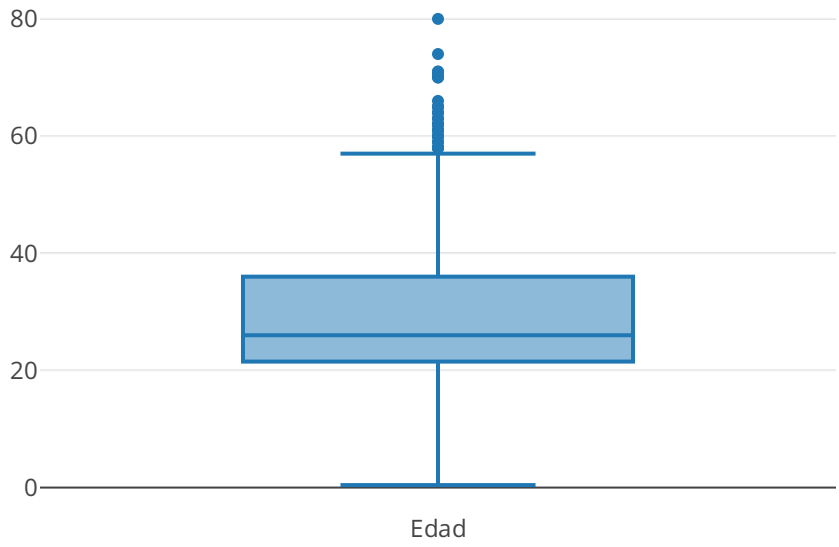
```
summary(titanicData)
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  1:342    2:184   male  :577   1st Qu.:21.50   1st Qu.:0.000   1st Qu.:0.0000
##           3:491                Median :26.00   Median :0.000   Median :0.0000
##                                Mean   :29.11   Mean   :0.523   Mean   :0.3816
##                                3rd Qu.:36.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0000
##       Fare        Embarked
##  Min.   :  0.00   C:168
##  1st Qu.:  7.91   Q: 77
##  Median : 14.45   S:646
##  Mean   : 32.20
##  3rd Qu.: 31.00
##  Max.   :512.33
```

Identificación y tratamiento de valores extremos.

La mejor manera de tratar los valores extremos es ir mostrando los diferentes valores de columnas numéricas en un diagrama de cajas y bigotes o *boxplot*. Para la realización de estos diagramas se ha utilizado la librería Plotly.

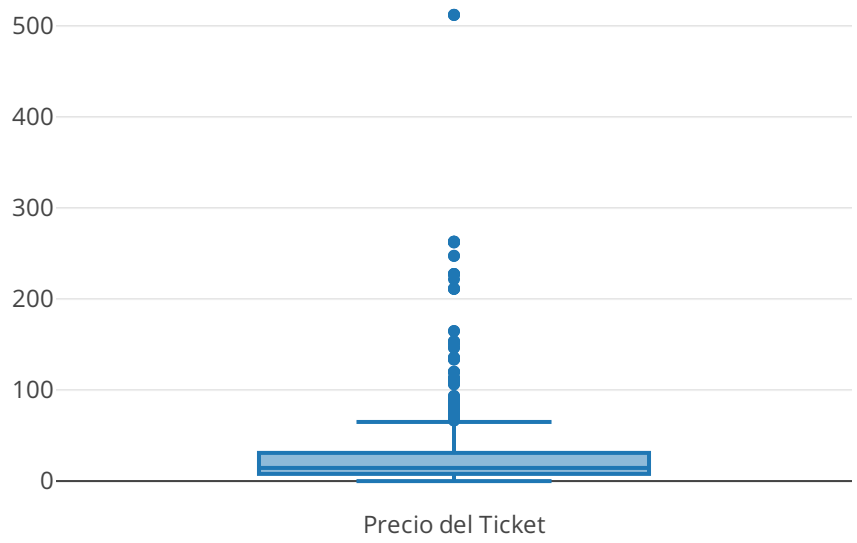
```
fig <- plot_ly(y = titanicData$Age, type = "box", name = "Edad")

fig
```



Aunque vemos unos cuantos outliers en el campo Edad, son perfectamente normales.

Veamos la columna Fare.

```
fig <- plot_ly(y = titanicData$Fare, type = "box", name = "Precio del Ticket")

fig
```



Se observan bastantes outliers, pero hay un precio que destaca más que los demás. Vamos a analizar esas filas.

```
filter(titanicData, Fare > 500)
```

```
##   Survived Pclass    Sex Age SibSp Parch     Fare Embarked
## 1        1      1 female  35     0     0 512.3292        C
## 2        1      1   male  36     0     1 512.3292        C
## 3        1      1   male  35     0     0 512.3292        C
```

Las tres filas pertenecen a personas del mismo rango de edad que embarcaron desde el mismo puerto. Por la exactitud de los datos y su homogeneidad, parecen datos correctos, por lo que se mantienen en el dataset.

Análisis de los datos

En esta sección dividiremos el conjunto de datos en subconjuntos para analizar y compararlos entre ellos.

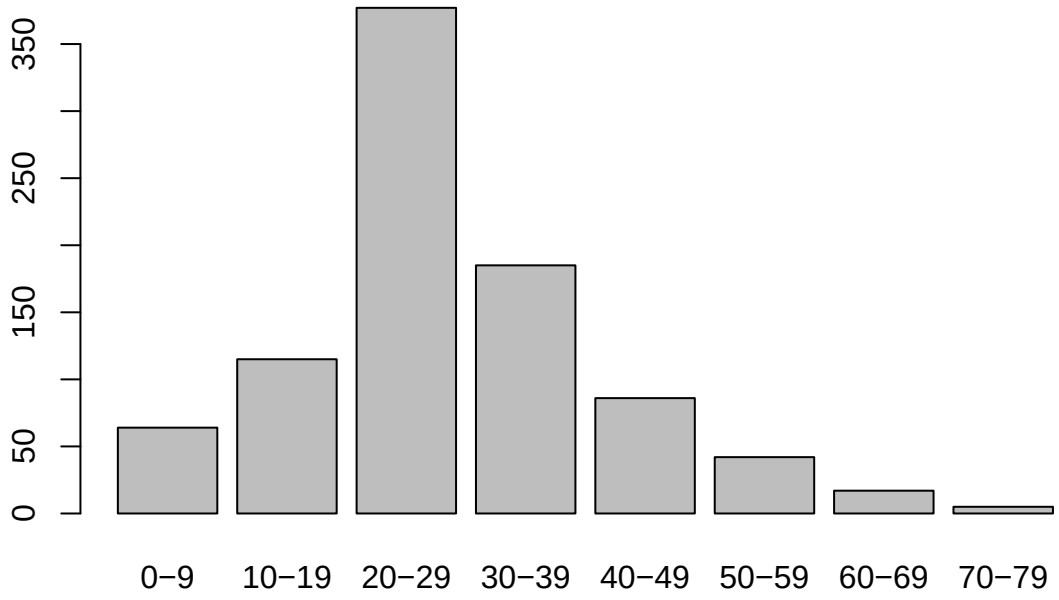
Selección de grupos de datos a analizar/comparar

El conjunto de datos lo analizaremos en fución a los siguientes grupos para determinar si guardan relación con la supervivencia:

- **Rango de edad** Columna AgeGroup
- **Poder adquisitivo** Columna Pclass nos dice en que clase embarcaron, lo que nos ayuda a inferir su nivel adquisitivo.
- **Sexo** Columna Sex.

Crearemos una columna en base a la edad, diviendola en segmentos para así poder realizar un análisis por categorías:

```
titanicData$AgeGroup <- cut(titanicData$Age, breaks = c(0,10,20,30,40,50,60,70,100), labels = c("0-9", "
plot(titanicData$AgeGroup)
```

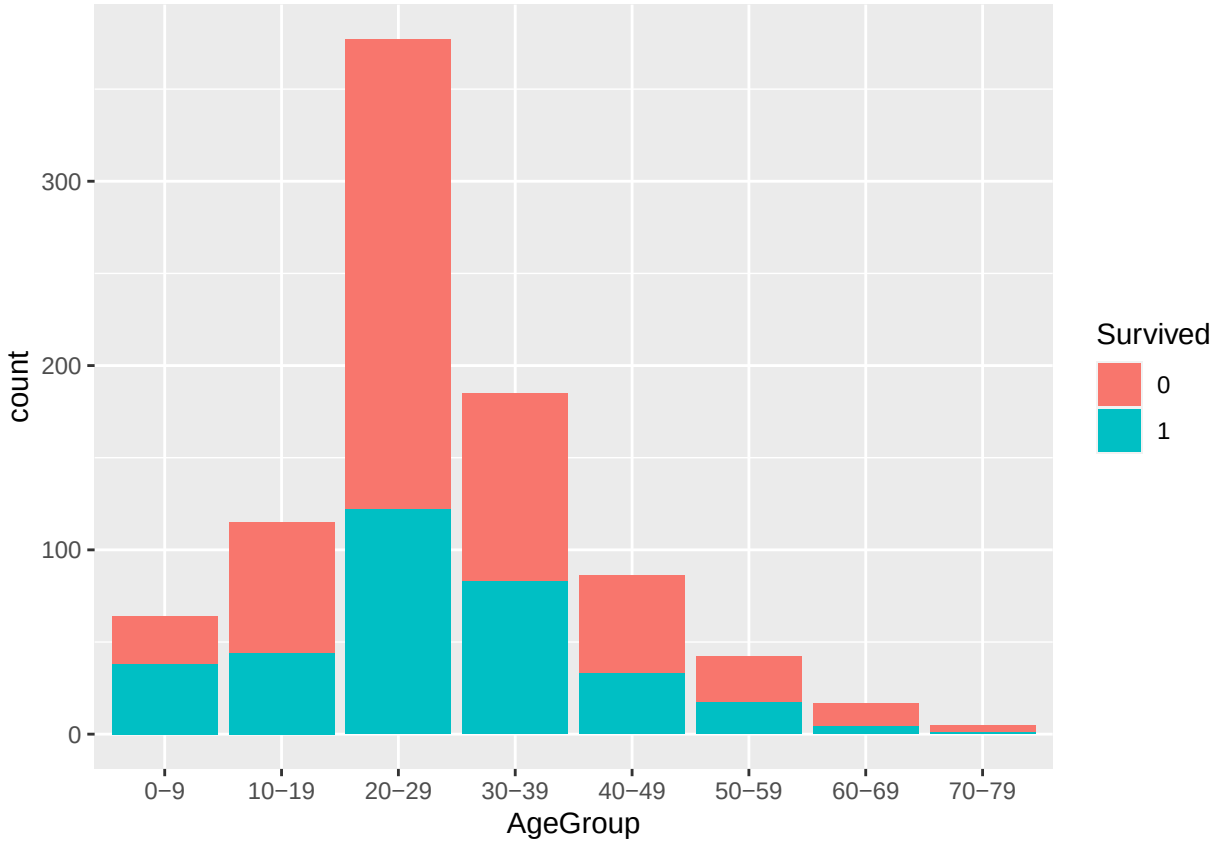



```
# Por poder adquisitivo
titanicData.firstClass <- titanicData[titanicData$Pclass == "1",]
titanicData.firstClass <- titanicData[titanicData$Pclass == "2",]
titanicData.firstClass <- titanicData[titanicData$Pclass == "3",]

# Por sexo
titanicData.female <- titanicData[titanicData$Sex == "female",]
titanicData.male <- titanicData[titanicData$Sex == "male",]
```

Vamos a mostrar con gráficas cada uno de los grupos nombrados junto con la supervivencia.

Rango de edad

```
ggplot(data=titanicData[1:nrow(titanicData),],aes(x=AgeGroup,fill=Survived))+geom_bar()
```

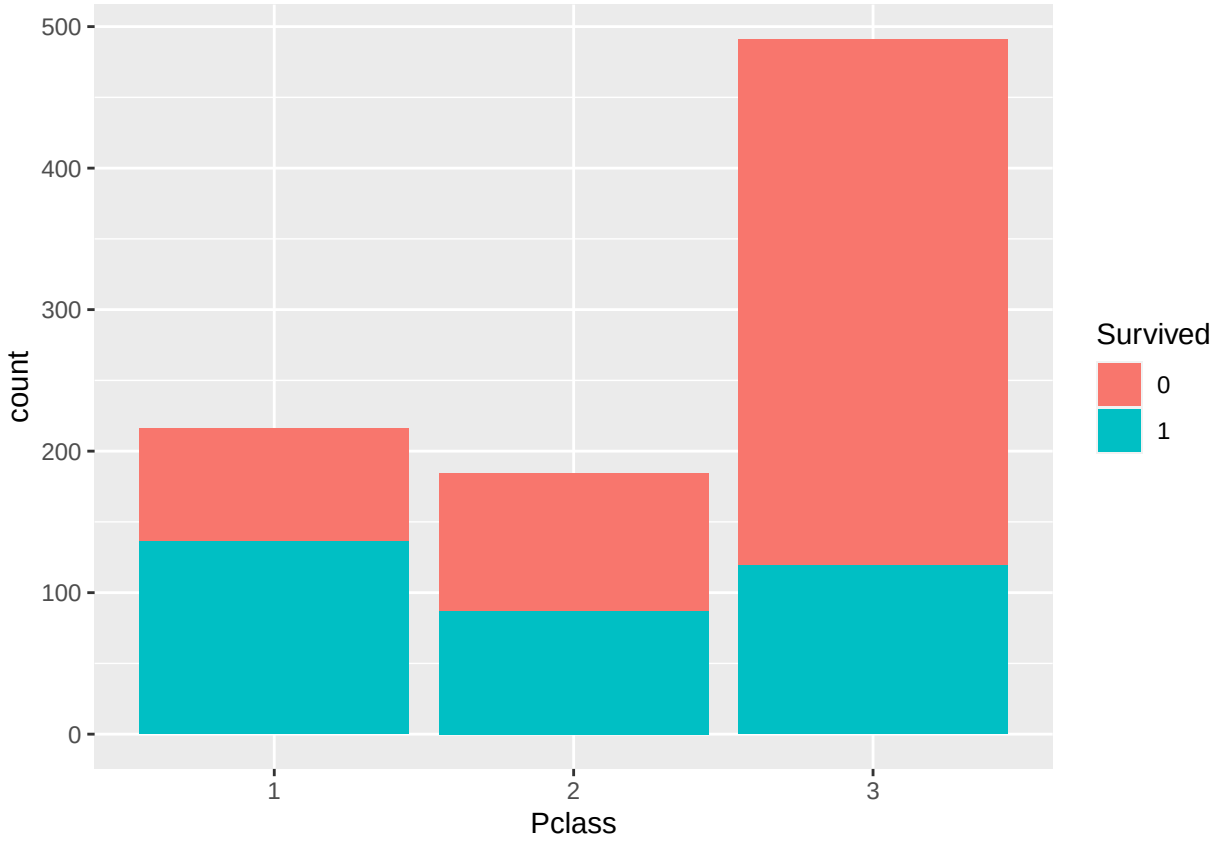



```
ggplot(data = titanicData,aes(x=Age,fill=Survived))+geom_histogram(binwidth = 3,position="fill")+ylab("
```

Con este último gráfico, podemos deducir que los niños tuvieron más posibilidades de salvarse y los mayores de 60 muchas menos.

Nivel adquisitivo

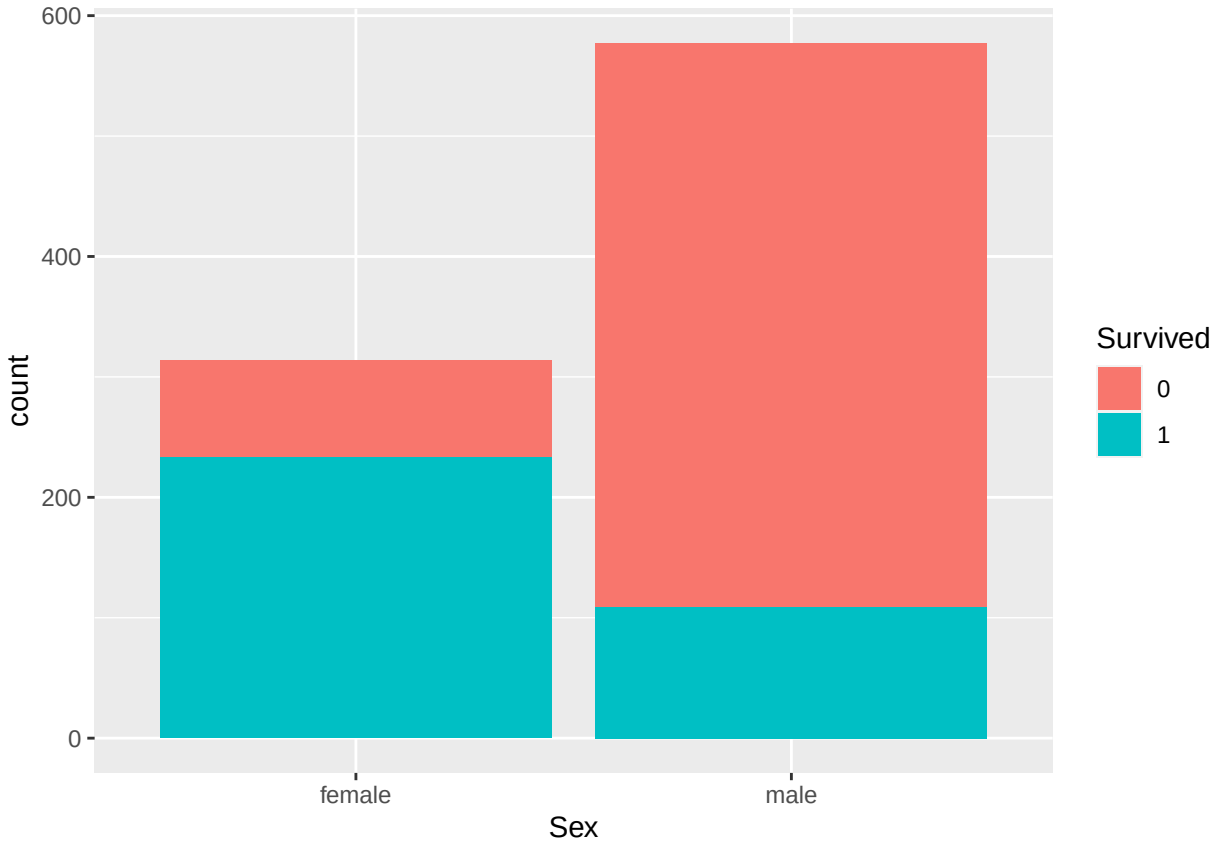
```
ggplot(data=titanicData[1:nrow(titanicData),],aes(x=Pclass,fill=Survived))+geom_bar()
```



También se aprecia un aumento de la supervivencia en las clases 1 y 2 con respecto a la 3.

Sexo

```
ggplot(data=titanicData[1:nrow(titanicData),],aes(x=Sex,fill=Survived))+geom_bar()
```



Comprobación de normalidad y homogeneidad de la varianza.

Comprobación de normalidad y homogeneidad de la varianza.

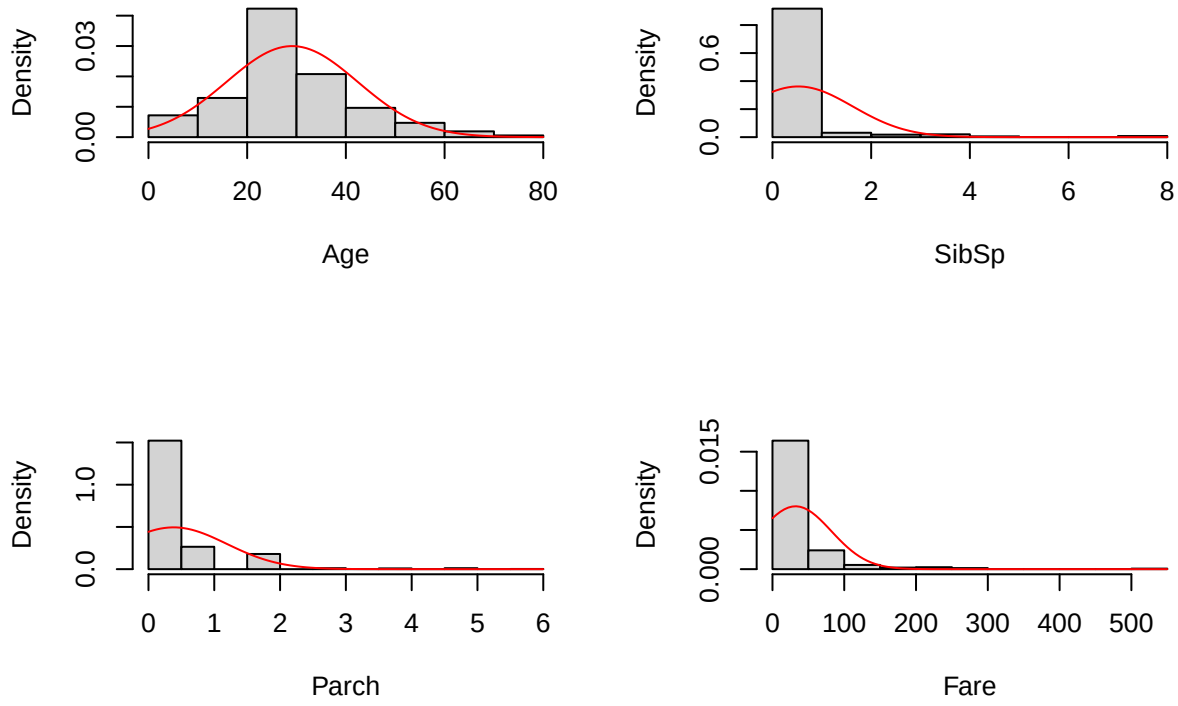
```
summary(titanicData)
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  1:342    2:184   male  :577   1st Qu.:21.50   1st Qu.:0.000   1st Qu.:0.0000
##           3:491                Median :26.00   Median :0.000   Median :0.0000
##                                Mean   :29.11   Mean   :0.523   Mean   :0.3816
##                                3rd Qu.:36.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0000
##
##       Fare        Embarked    AgeGroup
##  Min.   :  0.00   C:168    20-29  :377
##  1st Qu.:  7.91   Q: 77    30-39  :185
##  Median : 14.45   S:646    10-19  :115
##  Mean   : 32.20            40-49  : 86
##  3rd Qu.: 31.00            0-9    : 64
##  Max.   :512.33            50-59  : 42
##                            (Other): 22
```

Comprobación de normalidad

Se procede a comprobar la normalidad de las variables numéricas, es decir, Age, SibSp, Parch, y Fare. Para ello, se crearán histogramas univariados y QQ-plots, que permiten a los analistas de datos identificar datos sesgados hacia los lados y ver la normalidad de las variables. Se utiliza la librería MVN. El test aplicado es el Royston,

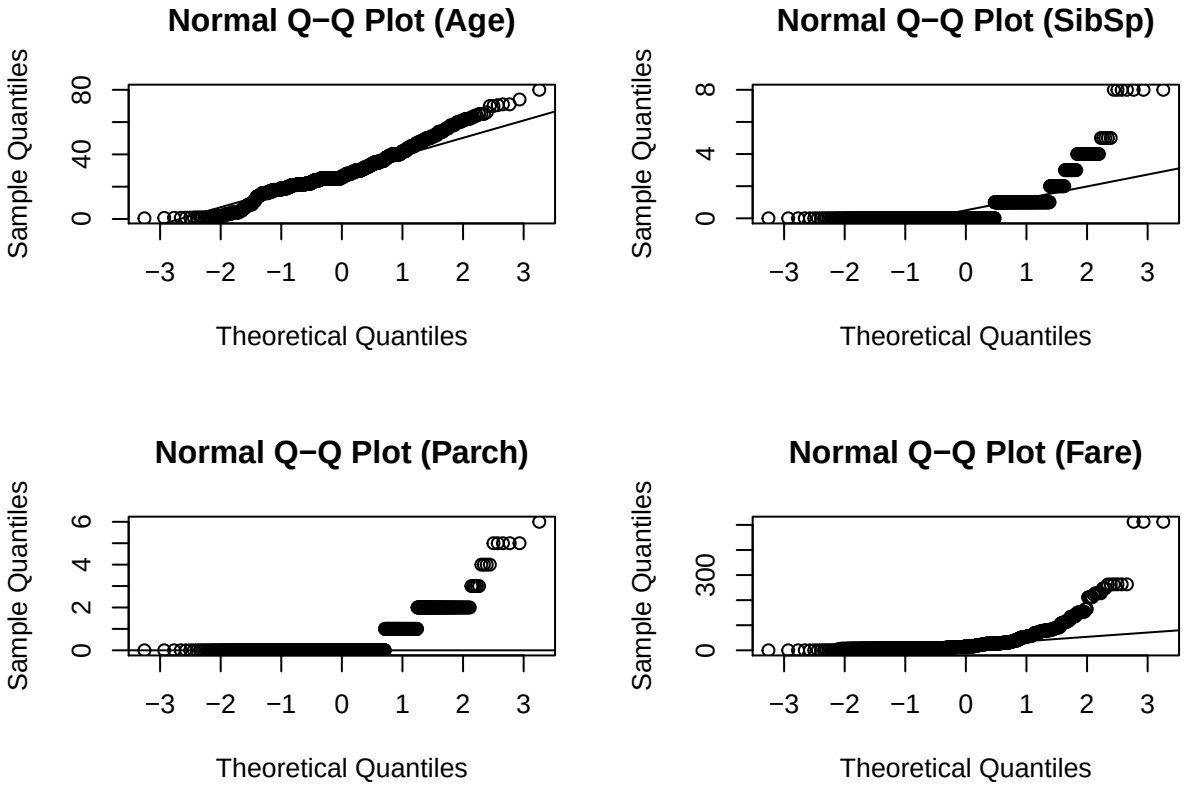
```
titanicData_numeric<- titanicData %>% select_if(is.numeric)

result<- mvn(data=titanicData_numeric, mvnTest="royston", univariatePlot="histogram")
```



En los histogramas se observa claramente que las variables SibSp, Parch y Fare están sesgadas hacia la izquierda y no siguen una distribución normal.

Sin embargo, la variable Age sí sigue aproximadamente una distribución Normal. Se procede a representar los QQ-Plots para corroborar esta suposición.

```
result<-mvn(data=titanicData_numeric, mvnTest = "royston", univariatePlot = "qqplot")
```



De nuevo, se observa que Age sí sigue aproximadamente una distribución normal.

Para contrastar la normalidad de esta variable, se aplica también el Test Shapiro-Wilk. Este test toma las siguientes hipótesis:

- **Hipótesis nula**: Los datos de la muestra no son significativamente diferentes de una población normal.
- **Hipótesis alternativa**: Los datos de la muestra son significativamente diferentes de una población normal.

Para todo valor de probabilidad mayor que el nivel de significación, en este caso 0.05, se acepta la hipótesis nula y se rechaza la alternativa.

```
shapiro.test(titanicData$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanicData$Age
## W = 0.96548, p-value = 1.118e-13
```

Dado que el p-valor es menor que 0.05, se rechaza la hipótesis nula, por lo que la varaible Age no sigue una distribución normal.

Se realiza el mismo test para las otras 3 variables, con idéntico resultado.

```
shapiro.test(titanicData$SibSp)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  titanicData$SibSp
## W = 0.51297, p-value < 2.2e-16
```

```
shapiro.test(titanicData$Parch)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanicData$Parch
## W = 0.53281, p-value < 2.2e-16
```

```
shapiro.test(titanicData$Fare)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  titanicData$Fare
## W = 0.52189, p-value < 2.2e-16
```

Homogeneidad de la varianza.

Pasamos a estudiar la homogenidad de la varianza mediante la aplicación del test de *Fligner-Killeen*. Ya que es una alternativa cuando no se cumple la normalidad de las muestras

```
fligner.test(as.integer(titanicData$Sex), as.integer(titanicData$Survived))
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  as.integer(titanicData$Sex) and as.integer(titanicData$Survived)
## Fligner-Killeen:med chi-squared = 36.761, df = 1, p-value = 1.336e-09
```

```
fligner.test(as.integer(titanicData$Pclass), as.integer(titanicData$Survived))
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  as.integer(titanicData$Pclass) and as.integer(titanicData$Survived)
## Fligner-Killeen:med chi-squared = 23.648, df = 1, p-value = 1.157e-06
```

Debido a que 3 de las 4 variables no siguen una distribución normal, se debe utilizar un test que permita obtener la homogeneidad en variables no normales. Por eso, hemos decidido utilizar en test Fligner-Killeen, un test no paramétrico que compara las varianzas basándose en la mediana.

Este test toma como premisas las siguientes hipótesis:

- **Hipótesis nula**: Todas las varianzas de las poblaciones son iguales.
- **Hipótesis alternativa**: Al menos dos de ellas difieren.

Primero, se aplica el test a todas las variables a la vez.

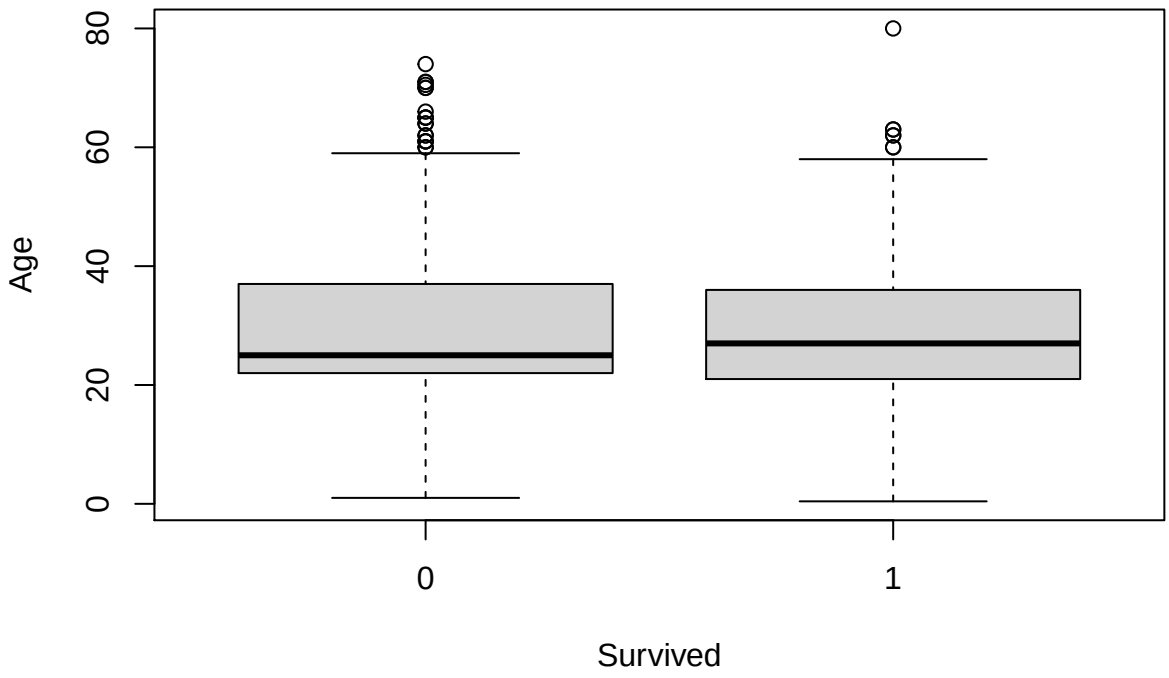
```
fligner.test(x = titanicData_numeric)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  titanicData_numeric
## Fligner-Killeen:med chi-squared = 2026.3, df = 3, p-value < 2.2e-16
```

Dado que el p-valor es inferior a 0.05, concluimos que tomando todas las variables al mismo tiempo no todas las varianzas son homogéneas. Se procede ahora a comprobar las variables 2 a 2.

Homogeneidad de varianza entre Age y Survived. Se mirará la homogeneidad de la varianza de la variable Age en función de la Supervivencia. Primero, se representa ambas varianzas en un boxplot para hacernos una idea de si podría existir homogeneidad.

```
boxplot(Age ~ Survived, data = titanicData)
```



Parece que podría cumplirse la homogeneidad. Se procede a ejecutar el test.

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Survived
## Fligner-Killeen:med chi-squared = 8.5407, df = 1, p-value = 0.003473
```

Aplicación de pruebas estadísticas

Correlaciones

¿ Qué variables influyen en la supervivencia? En este apartado comprobaremos las correlaciones entre distintas variables y su influencia en la supervivencia.

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
```

```
# con respecto al campo "survive"
for (i in c("Age", "Sex","Pclass")) {
    spearman_test = cor.test(
        as.double(titanicData[,i]),
        as.integer(titanicData$Survived),
        method = "spearman", exact = FALSE)

  corr_coef = spearman_test$estimate
  p_val = spearman_test$p.value

    pair = matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
  rownames(corr_matrix)[nrow(corr_matrix)] <- i
}
print(corr_matrix)
```

```
##           estimate      p-value
## Age    -0.03856812 2.501210e-01
## Sex    -0.54335138 1.406066e-69
## Pclass -0.33966794 1.687608e-25
```

La menos correlacionada quizá sería la edad y la más fuertemente correlacionada la edad.

Contraste de hipótesis:

Dependencia entre Pclass y Survived En primer lugar, se comprobará si las diferentes clases en las que viajaban los pasajeros influyó en si sobrevivieron al accidente o no. Por tanto, se identifican las siguientes hipótesis:

- Hipótesis nula: El factor PClass y el el factor Survived son independientes.
- Hipótesis alternativa: Los dos factores son dependientes.

Para comprobar la dependencia entre dos variables categóricas, se aplica el test chi-cuadrado. Primero, se genera una tabla de contingencia entre ambos factores.

```
contingece <- table(titanicData$Survived, titanicData$Pclass)
contingece
```

```
##
##       1   2   3
##   0  80  97 372
##   1 136  87 119
```

Una vez creada la tabla, se aplica el test.

```
chisq.test(contingece)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingece
## X-squared = 102.89, df = 2, p-value < 2.2e-16
```

Dado que el p-valor es inferior a 0.05, se rechaza la hipótesis nula, y aceptamos la hipótesis alternativa.

Es decir, el test determina que la clase de los pasajeros es determinante a la hora de predecir si sobrevivió o no.

Dependencia entre AgeGroup y Survived A continuación, se comprobará si la edad de los pasajeros influyó en si sobrevivieron al accidente o no. Por tanto, se identifican las siguientes hipótesis:

- Hipótesis nula: El factor Age y el el factor Survived son independientes.
- Hipótesis alternativa: Los dos factores son dependientes.

Para comprobar la dependencia entre dos variables categóricas, se aplica el test chi-cuadrado. Primero, se genera una tabla de contingencia entre ambos factores. Se utilizará los grupos de edad.

```
contingece <- table(titanicData$Survived, titanicData$AgeGroup)
contingece
```

```
##
##     0-9 10-19 20-29 30-39 40-49 50-59 60-69 70-79
##   0  26    71   255   102    53    25    13     4
##   1  38    44   122    83    33    17     4     1
```

Una vez creada la tabla, se aplica el test.

```
chisq.test(contingece)
```

```
## Warning in chisq.test(contingece): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingece
## X-squared = 23.371, df = 7, p-value = 0.001469
```

Dado que el p-valor es inferior a 0.05, se rechaza la hipótesis nula, y aceptamos la hipótesis alternativa.

Es decir, el test determina que la edad de los pasajeros es determinante a la hora de predecir si sobrevivió o no.

Dependencia entre Sex y Survived A continuación, se comprobará si el sexo de los pasajeros influyó en si sobrevivieron al accidente o no. Por tanto, se identifican las siguientes hipótesis:

- Hipótesis nula: El factor Sex y el el factor Survived son independientes.
- Hipótesis alternativa: Los dos factores son dependientes.

Para comprobar la dependencia entre dos variables categóricas, se aplica el test chi-cuadrado. Primero, se genera una tabla de contingencia entre ambos factores. Se utilizará los grupos de edad.

```
contingece <- table(titanicData$Survived, titanicData$Sex)
contingece
```

```
##
##     female male
##   0     81  468
##   1    233  109
```

Una vez creada la tabla, se aplica el test.

```
chisq.test(contingece)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  contingece
## X-squared = 260.72, df = 1, p-value < 2.2e-16
```

Dado que el p-valor es inferior a 0.05, se rechaza la hipótesis nula, y aceptamos la hipótesis alternativa.

Es decir, el test determina que el sexo de los pasajeros es determinante a la hora de predecir si sobrevivió o no.

Regresión logística:

```
model1 <- glm(formula = as.factor(Survived) ~ Sex , family = "binomial", data= titanicData);
summary(model1)
```

```
##
## Call:
## glm(formula = as.factor(Survived) ~ Sex, family = "binomial",
##     data = titanicData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6462  -0.6471  -0.6471   0.7725   1.8256
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.0566     0.1290   8.191 2.58e-16 ***
## Sexmale      -2.5137     0.1672 -15.036  < 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: 1186.7  on 890  degrees of freedom
## Residual deviance:  917.8  on 889  degrees of freedom
## AIC: 921.8
##
## Number of Fisher Scoring iterations: 4
```

```
model2 <- glm(formula = as.factor(Survived) ~ Sex + Age , family = "binomial", data= titanicData);
summary(model2)
```

```
##
## Call:
## glm(formula = as.factor(Survived) ~ Sex + Age, family = "binomial",
##     data = titanicData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6557  -0.6483  -0.6447   0.7724   1.8442
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.0794932  0.2132888   5.061 4.17e-07 ***
## Sexmale     -2.5113789  0.1680182 -14.947  < 2e-16 ***
## Age         -0.0008391  0.0062152  -0.135    0.893
## ---
## 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:  917.79  on 888  degrees of freedom
## AIC: 923.79
##
## Number of Fisher Scoring iterations: 4
```

```
model3 <- glm(formula = as.factor(Survived) ~ Sex + Age + Pclass, family = "binomial", data= titanicData
summary(model3)
```

```
##
## Call:
## glm(formula = as.factor(Survived) ~ Sex + Age + Pclass, family = "binomial",
##     data = titanicData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7134  -0.6631  -0.4328   0.6390   2.4803
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.730341   0.382564   9.751  < 2e-16 ***
## Sexmale     -2.579528   0.186948 -13.798  < 2e-16 ***
## Age         -0.037255   0.007669  -4.858 1.19e-06 ***
## Pclass2     -1.212406   0.262645  -4.616 3.91e-06 ***
## Pclass3     -2.502962   0.256630  -9.753  < 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: 1186.7  on 890  degrees of freedom
## Residual deviance:  801.6  on 886  degrees of freedom
## AIC: 811.6
##
## Number of Fisher Scoring iterations: 5
```

Obtenemos una tabla para comparar los modelos generados:

```
tabla.coeficientes <- matrix(c(1, summary(model1)$aic,
    2, summary(model2)$aic,
    3, summary(model3)$aic),
    ncol = 2, byrow = TRUE)
colnames(tabla.coeficientes) <- c("Modelo", "AIC")
```

Representación de los resultados

A lo largo de los apartados hemos explicado y aportado gráficas de los resultados obtenidos. Ahora vamos a hacer un pequeño resumen. Como se puede ver en la matriz de correlaciones, Age y Sex están correlacionados con la supervivencia.

```
print(corr_matrix)
```

```
##           estimate      p-value
## Age    -0.03856812 2.501210e-01
## Sex    -0.54335138 1.406066e-69
```

```
## Pclass -0.33966794 1.687608e-25
```

Mediante el contraste de hipótesis hemos obtenido que Survived tiene dependencias con las tres variables estudiadas (ver apartado Contraste de hipótesis 4.3.X).

Mediante las regresiones logísticas hemos utilizado estas variables para elborar modelos más complejos(el último con las 3 variables) sus resultados son los siguientes:

```
tabla.coeficientes
```

```
##      Modelo      AIC
## [1,]      1 921.8039
## [2,]      2 923.7857
## [3,]      3 811.5976
```

Resolución del problema y conclusiones

Con los modelos elaborados (regresión, contraste de hipótesis y correlaciones) llegamos a la conclusión de que todas las variables influyen en la supervivencia de los individuos. El orden de influencia sería el siguiente:

- Sex
- Age
- PClass

Contribuciones

Contribuciones	Firma
Investigación previa	David Herrero Pascual, Andrés Baamonde Lozano
Redacción de las respuestas	David Herrero Pascual, Andrés Baamonde Lozano
Desarrollo y código	David Herrero Pascual, Andrés Baamonde Lozano