Tipología y ciclo de vida de los datos: Práctica 2: Limpieza y validación de los datos

Autores: Youness El Guennouni y Mario Gutiérrez Calvo de Mora

Mayo 2019

Contents

Introducción

---

trabajaremos con el juego de datos "Titanic" que recoge datos sobre el famoso crucero y sobre el que es fácil realizar tareas de clasificación predictiva sobre la variable "Survived".

Las actividades que llevaremos a cabo en esta práctica suelen enmarcarse en las fases iniciales de un proyecto de minería de datos y consisten en la selección de caraterásticas o variables y la preparación del juego de datos para posteriormente ser consumido por un algoritmo.

Primeramente realizaremos el estudio de las variables de un juego de datos, es decir, haremos un trabajo descriptivo del mismo. Y de forma posterior, realizaremos el estudio de algoritmos predictivos y las conclusiones que se pueden extraer del estudio.

Siguiendo las principales etapas de un proyecto analático, las diferentes tareas a realizar (y justificar) son las siguientes:

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?
2. Integración y selección de los datos de interés a analizar.
3. Limpieza de los datos. 3.1. ¿Los datos contienen ceros o elementos vacáos? ¿Cómo gestionarías cada uno de estos casos? 3.2. Identificación y tratamiento de 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). 4.2. Comprobación de la normalidad y homogeneidad de la varianza. 4.3. Aplicación de pruebas estadásticas 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. Aplicar al menos tres métodos de análisis diferentes.
5. Representación de los resultados a partir de tablas y gráficas.
6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?
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 preferás, también podéis trabajar en Python.

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

El conjunto de datos objeto de análisis se ha obtenido a partir de este enlace en Kaggle y está constituido por 12 caracterásticas (columnas) que presentan a los 891 pasajeros (filas o registros). Entre los campos de este conjunto de datos, encontramos los siguientes:

- **PassengerId**: identificador unico del pasajero.
- **Survived**: Si el pasajero ha sobrevivido (1) o no (0)
- **Pclass**: En que clase viajaba
- **Name**: Nombre de pasajero

- **Sex**: genéro de pasajero
- **SibSp**: Numero de hermanos / cónyuges a bordo
- **Parch**: Numero de padres / hijos a bordo
- **Ticket**: Numero de ticket
- **Fare**: tárifa del viaje
- **Cabin**: Cabina
- **Embarked**: El puerto desde el cual ha embarcado el pasajero (C- Cherbourg, S - Southampton, Q - Queenstown)

Es importante saber a que preguntas debemos de responder para definir un objetivo claro y no desviarnos de ello. En nuestro caso el problema que debemos de solventar será ¿Que factores influyen directamente o indirectamente en la superviviencia o no de un pasajero? Además, se podrá proceder a crear modelos de regresión que permitan predecir la supervivencia o no de un pasajero en función de sus caráctiristas y contrastes de hipótesis que ayuden a identificar propiedades interesantes en las muestras que puedan ser inferidas con respecto a la población.

Integración y selección de los datos de interés a analizar

Desde el análisis de los datos podemos descartar algunos factores desde el inicio como el numero de ticket, tarifa o nombre de pasajero. Otro dato que decidimos no tenerle en cuenta es la cabina ya que más de 70% vienen vacios.

Lectura de ficheros

Cargar a los archivos `train.csv` y`test.csv`. Una vez cargado el archivo, valida que los tipos de datos son los correctos. Si no es asá, conviertelos al tipo oportuno.

- Archivo de datos (`train.csv`)

```
train <- read.csv( "./data/train.csv")
head(train)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                 Name    Sex Age SibSp
## 1                            Braund, Mr. Owen Harris   male  22     1
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1
## 3                             Heikkinen, Miss. Laina female  26     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1
```

```
## 5                            Allen, Mr. William Henry   male  35     0
## 6                                    Moran, Mr. James   male  NA     0
##   Parch           Ticket    Fare Cabin Embarked
## 1     0        A/5 21171  7.2500              S
## 2     0         PC 17599 71.2833   C85        C
## 3     0 STON/O2. 3101282  7.9250              S
## 4     0           113803 53.1000  C123        S
## 5     0           373450  8.0500              S
## 6     0           330877  8.4583              Q
```

```
sapply( train, class)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##   "integer"   "integer"   "integer"    "factor"    "factor"   "numeric"
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##   "integer"   "integer"    "factor"   "numeric"    "factor"    "factor"
```

- Archivo de los tests (`test.csv`)

```
test <- read.csv( "./data/test.csv")
head(test)
```

```
##   PassengerId Pclass                                          Name    Sex
## 1         892      3                              Kelly, Mr. James   male
## 2         893      3              Wilkes, Mrs. James (Ellen Needs) female
## 3         894      2                     Myles, Mr. Thomas Francis   male
## 4         895      3                              Wirz, Mr. Albert   male
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female
## 6         897      3                    Svensson, Mr. Johan Cervin   male
##    Age SibSp Parch  Ticket    Fare Cabin Embarked
## 1 34.5     0     0  330911  7.8292              Q
## 2 47.0     1     0  363272  7.0000              S
## 3 62.0     0     0  240276  9.6875              Q
## 4 27.0     0     0  315154  8.6625              S
## 5 22.0     1     1 3101298 12.2875              S
## 6 14.0     0     0    7538  9.2250              S
```

```
sapply( test, class)
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##   "integer"   "integer"    "factor"    "factor"   "numeric"   "integer"
##       Parch      Ticket        Fare       Cabin    Embarked
##   "integer"    "factor"   "numeric"    "factor"    "factor"
```

Borrar a las columnas innecesarias

```
train <- select(train, -Name, -Ticket )
test  <- select(test, -Name, -Ticket )
head (train)
```

```
##   PassengerId Survived Pclass    Sex Age SibSp Parch    Fare Cabin
## 1           1        0      3   male  22     1     0  7.2500
## 2           2        1      1 female  38     1     0 71.2833   C85
## 3           3        1      3 female  26     0     0  7.9250
## 4           4        1      1 female  35     1     0 53.1000  C123
## 5           5        0      3   male  35     0     0  8.0500
## 6           6        0      3   male  NA     0     0  8.4583
##   Embarked
## 1        S
## 2        C
## 3        S
## 4        S
## 5        S
## 6        Q
```

Limpieza de datos

En esta sección vamos a llevar a cabo el proceso de limpieza de datos que consiste en:

Los datos contienen ceros o elementos vacíos

A continuación vamos a detectar a los valores vacios y nulos.

```
# Estadísticas de valores vacíos
colSums(is.na(train))
```

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

```
colSums(train=="")
```

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

Llegados a este punto debemos decidir cómo manejar estos registros que contienen valores desconocidos para algun campo. Una opción podrá ser eliminar esos registros que incluyen este tipo de valores, pero ello supondría desaprovechar información.
Como alternativa, se empleará un método de imputación de valores basado en la similitud o diferencia entre los registros: la imputación basada en k vecinos más próximos. La elección de esta alternativa se realiza bajo la hipótesis de que nuestros registros guardan cierta relación. No obstante, es mejor trabajar con datos aproximados que con los propios elementos vacíos, ya que obtendremos análisis con menor margen de error.

```
#Para los valores perdidos procedemos con aplicar la distancia de Gower
train <- kNN(train)

#Rempazar la edad por la media
```

```
train$Age[is.na(train$Age)] <- winsor.mean(train$Age,trim=0.05)

# Tomamos valor "C" para los valores vacáos de la variable "Embarked"
train$Embarked[train$Embarked==""]="C"

sapply(train, function(x) sum(is.na(x)))
```

```
##     PassengerId        Survived          Pclass             Sex
##               0               0               0               0
##             Age           SibSp           Parch            Fare
##               0               0               0               0
##           Cabin        Embarked PassengerId_imp    Survived_imp
##               0               0               0               0
##      Pclass_imp         Sex_imp         Age_imp       SibSp_imp
##               0               0               0               0
##       Parch_imp        Fare_imp       Cabin_imp    Embarked_imp
##               0               0               0               0
```

Identificación y tratamiento de valores extremos

- Cuadro de las estimaciónes no robustas y robustas.

En el siguiente cuadro se van a mostrar a las estimaciones no robustas y robustas por un posible uso a la hora de remplazar a los valores perdidos o a los extremos.

```
age_s<-summary(train$Age)
pclass_s<-summary(train$Pclass)
sibSp_s<-summary(train$SibSp)
parch_s<-summary(train$Parch)

table_s <- suppressWarnings(rbind(age_s,pclass_s,sibSp_s,parch_s))
age_r <- c(sd(train$Age), winsor.mean(train$Age,trim=0.05), IQR(train$Age))
pclass_r <- c(sd(train$Pclass), "NA", IQR(train$Pclass))
sibSp_r <- c(sd(train$SibSp), winsor.mean(train$SibSp,trim=0.05), IQR(train$SibSp))
parch_r <- c(sd(train$Parch), winsor.mean(train$Parch,trim=0.05), IQR(train$Parch))

table_r <- rbind(age_r,pclass_r,sibSp_r, parch_r)
table_res <- cbind(table_s, table_r)
colnames( table_res) <- c("Min", "1st Qu", "Median", "Mean", "3rd Qu", "Max", "SD", "WINSOR", "IQR")
kable(table_res)
```

	Min	1st Qu	Median	Mean	3rd Qu	Max	SD	WINSOR	IQR
age_s	0.42	21	28	29.3907631874299	36.25	80	13.8561859986118	29.1683501683502	15.2
pclass_s	1	2	3	2.30864197530864	3	3	0.836071240977049	NA	1
sibSp_s	0	0	0	0.52300785634119	1	8	1.10274343229343	0.452300785634119	1
parch_s	0	0	0	0.381593714927048	0	6	0.806057221129948	0.345679012345679	0

- Detactamos a los valores extremos

Los valores extremos o outliers son aquellos que parecen no ser congruentes sin los comparamos con el resto

de los datos. Para identificarlos, podemos hacer uso de dos vías: (1) representar un diagrama de caja por cada variable y ver qué valores distan mucho del rango intercuartílico(la caja) o (2) utilizar la función boxplots.stats() de R, la cual se emplea a continuación. Así, se mostrarán sólo los valores atípicos para aquellas variables que los contienen:

```
#Los valores atápicos Age
boxplot.stats(train$Age)$out
```

```
##  [1] 66.0 65.0 71.0 70.5 61.0 61.0 62.0 63.0 65.0 61.0 60.0 64.0 65.0 63.0
## [15] 71.0 64.0 62.0 62.0 60.0 61.0 61.0 80.0 60.0 70.0 60.0 60.0 70.0 62.0
## [29] 74.0
```

```
#Los valores atápicos SibSp.
boxplot.stats(train$SibSp)$out
```

```
##  [1] 3 4 3 3 4 5 3 4 5 3 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 3 5 3 5 3 4 4 3 3
## [36] 5 4 3 4 8 4 3 4 8 4 8
```

```
#Los valores atápicos Parch.
boxplot.stats(train$Parch)$out
```

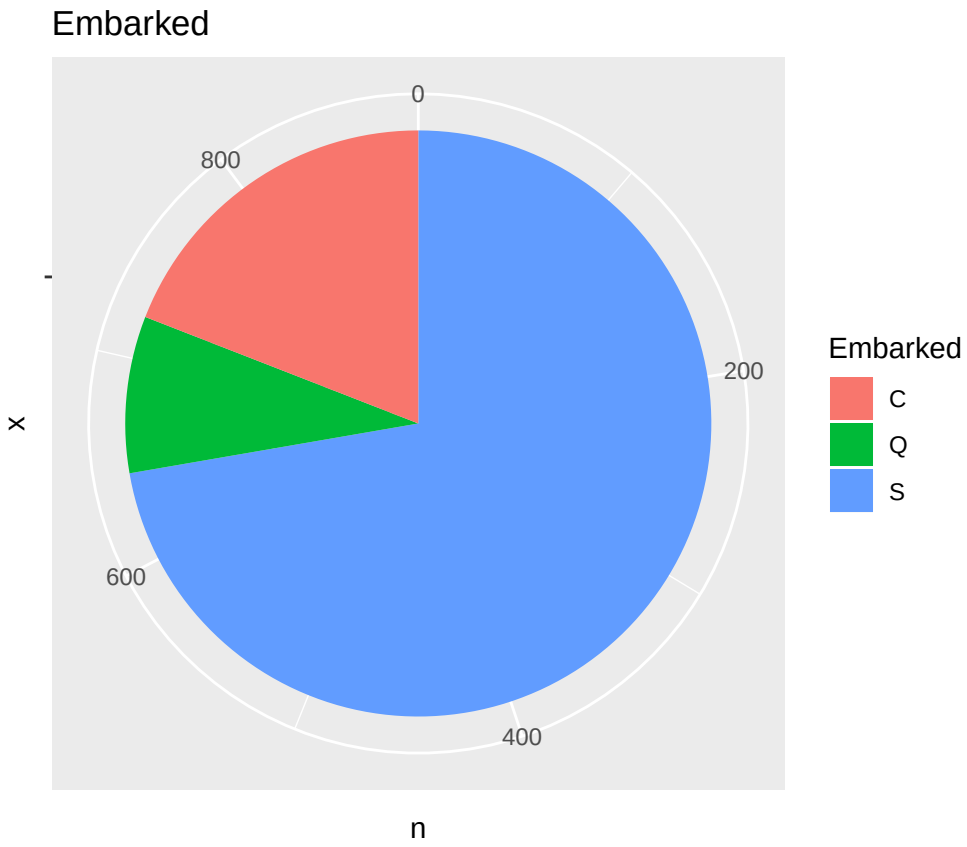
```
##   [1] 1 2 1 5 1 1 5 2 2 1 1 2 2 2 1 2 2 2 3 2 2 1 1 1 1 2 1 1 2 2 1 2 2 2 1
##  [36] 2 1 1 2 1 4 1 1 1 1 2 2 1 2 1 1 1 2 1 1 2 2 2 1 1 2 2 1 2 1 1 1 1 1 1
##  [71] 1 2 1 2 2 1 1 2 1 1 2 1 1 1 1 2 1 1 1 4 1 1 2 2 2 2 2 1 1 1 2 2 1 1 2
## [106] 2 3 4 1 2 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 2 2 2 2 2 1 1 2 1 4 1 1 2 1
## [141] 2 1 1 2 5 2 1 1 1 2 1 5 2 1 1 1 2 1 6 1 2 1 2 1 1 1 1 1 1 1 3 2 1 1 1
## [176] 1 2 1 2 3 1 2 1 2 2 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 3 2 1 1 1
## [211] 1 5 2
```

No obstante, si revisamos los anteriores datos para varios pasajeros escogido aleatoriamente de esta web, comprobamos que son valores que perfectamente pueden darse. Es por ello que el manejo de estos valores extremos consistirá en simplemente dejarlos como actualmente están recogidos.

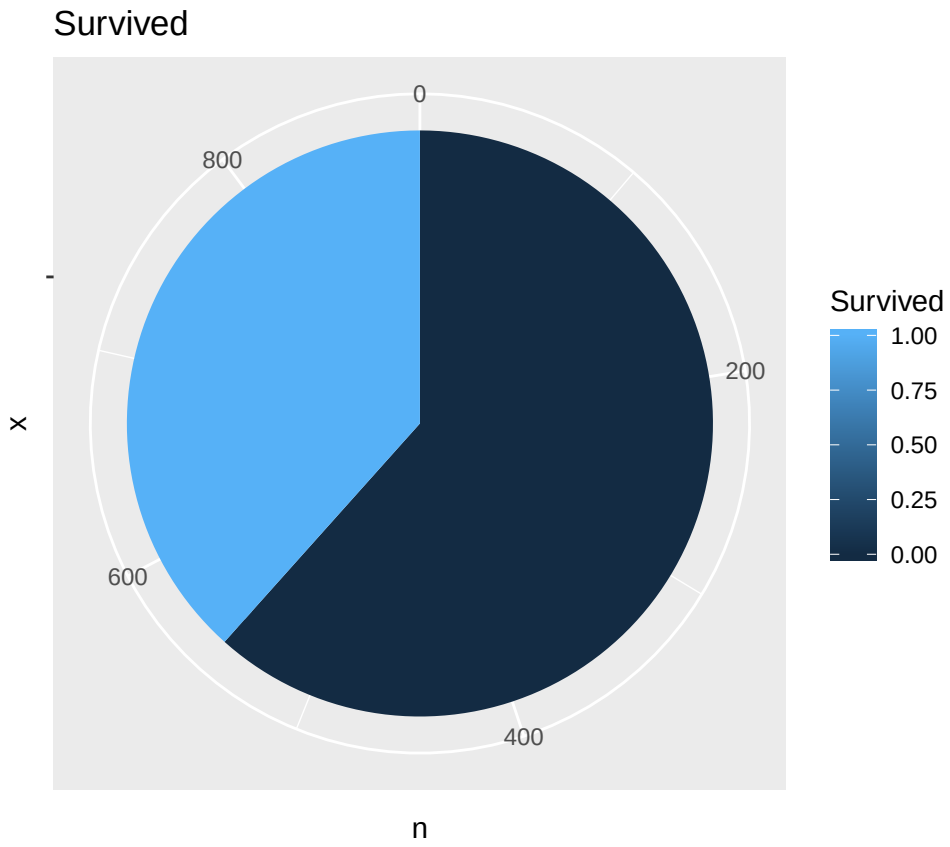
Análisis de los datos

Se representan gráficamente las variables del conjunto de datos para poder visualizar la distribución de valores de las variables.

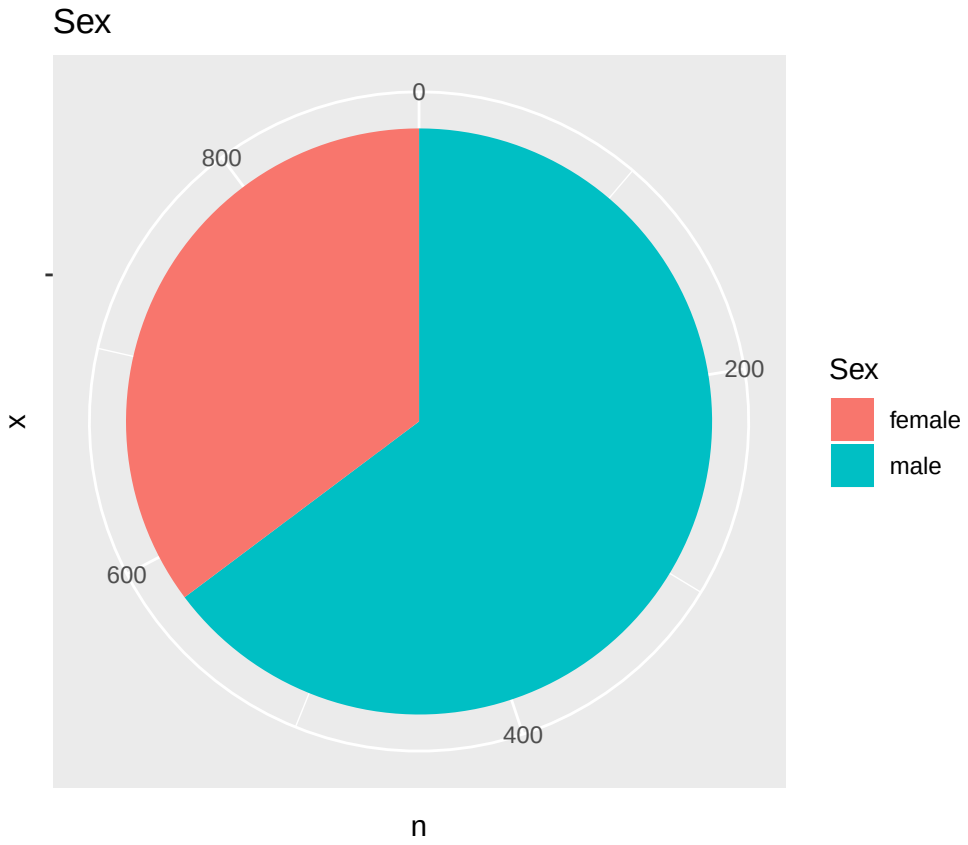
```
Embarkedsum <- summarize( group_by(train, Embarked), n=length(Embarked))
ggplot( Embarkedsum, aes(x="", y=n, fill=Embarked)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Embarked")
```



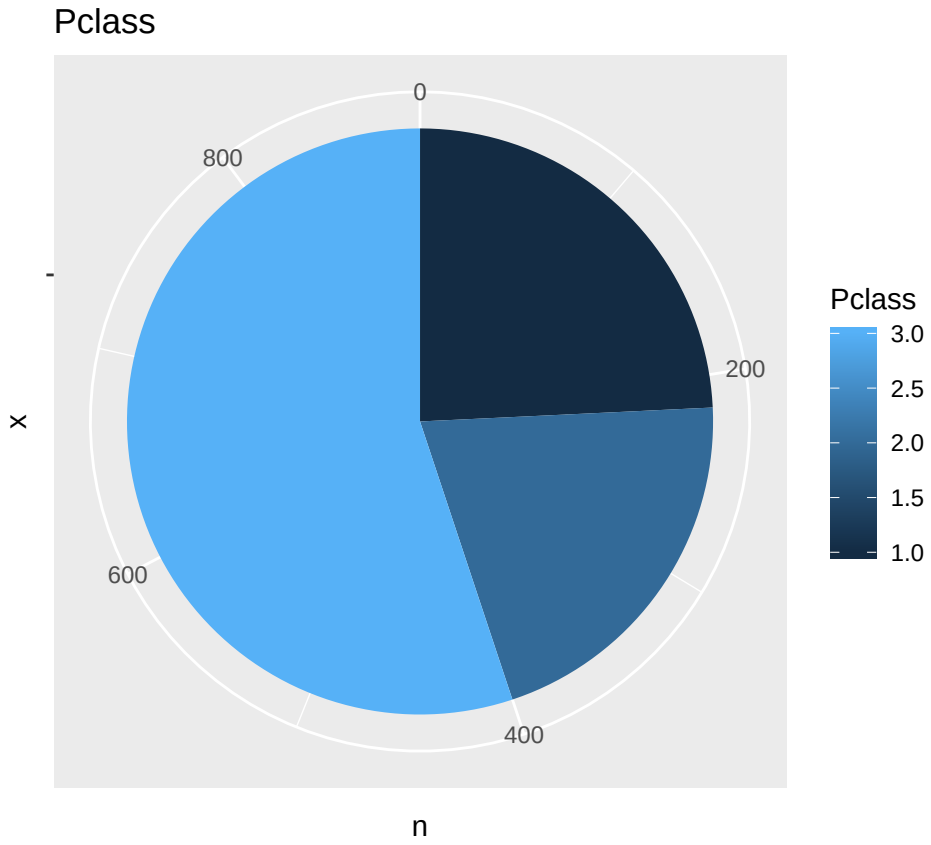

```
Survivedsum <- summarize( group_by(train, Survived), n=length(Survived))
ggplot( Survivedsum, aes(x="", y=n, fill=Survived)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Survived")
```



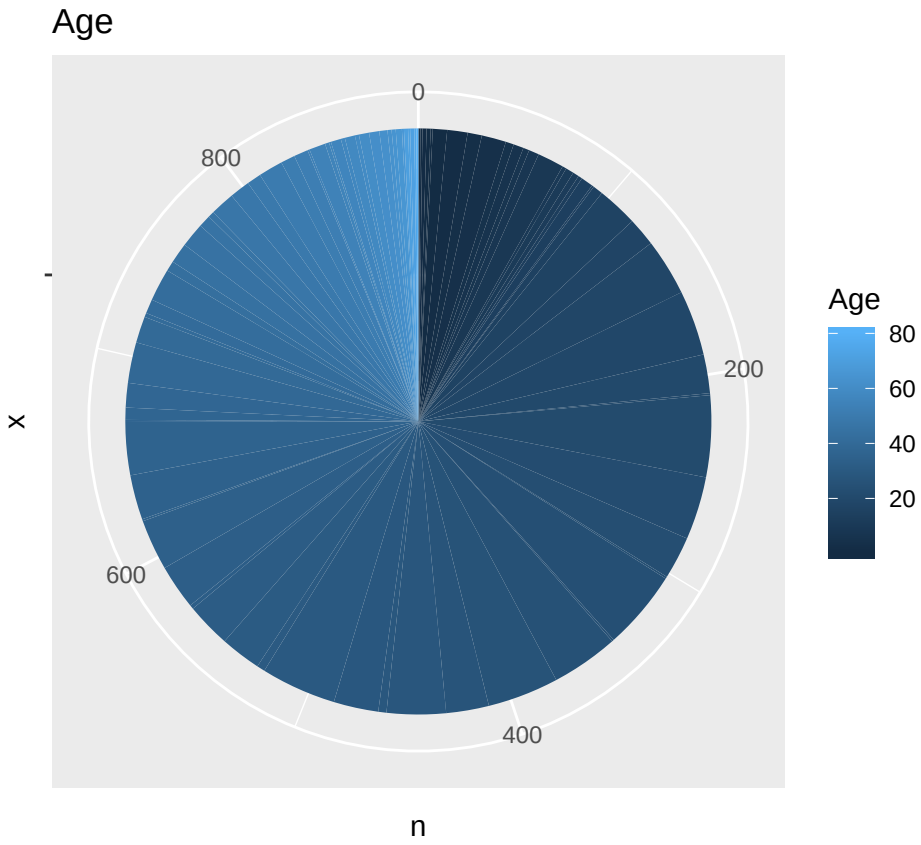

```
Sexsum <- summarize( group_by(train, Sex), n=length(Sex))
ggplot( Sexsum, aes(x="", y=n, fill=Sex)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Sex")
```



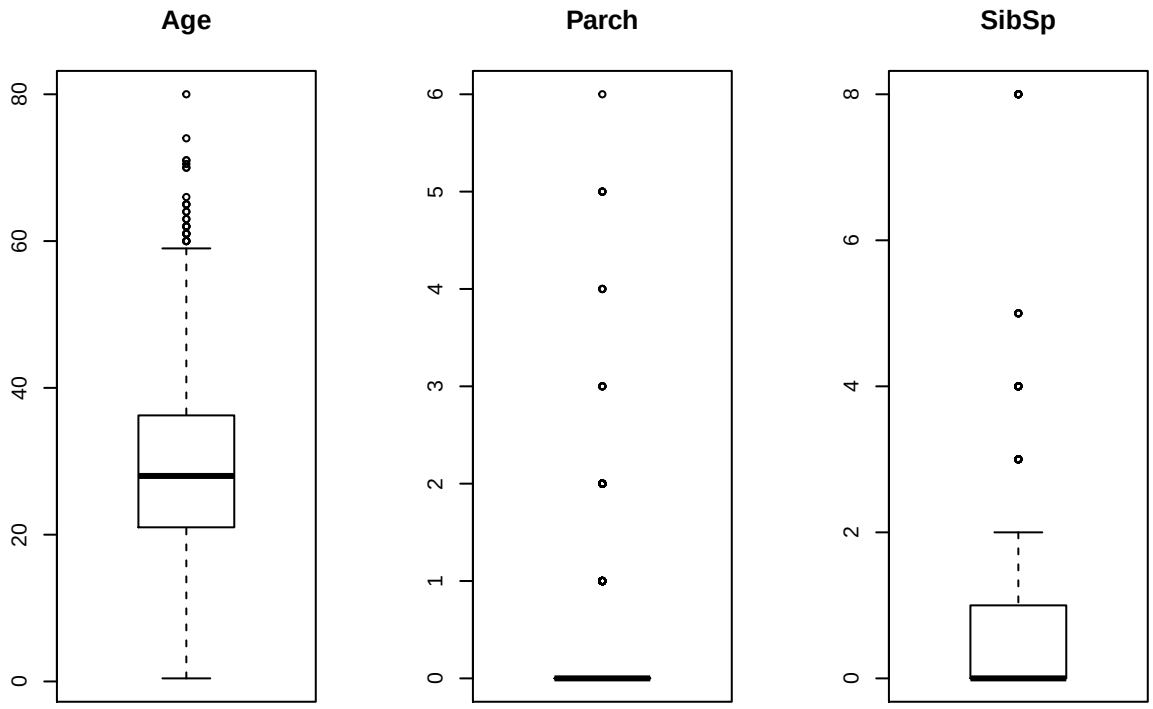

```
Pclasssum <- summarize( group_by(train, Pclass), n=length(Pclass))
ggplot( Pclasssum, aes(x="", y=n, fill=Pclass)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Pclass")
```




```
Agesum <- summarize( group_by(train, Age), n=length(Age))
ggplot(Agesum, aes(x="", y=n, fill=Age))+
geom_bar(width = 1, stat = "identity") + ggtitle("Age")+
coord_polar("y", start=0)
```




```
par(mfrow=c(1,3))
boxplot(train$Age, main="Age")
boxplot(train$Parch, main="Parch")
boxplot(train$SibSp, main="SibSp")
```




```
par(mfrow=c(1,1))
```

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

A continuación, se seleccionan los grupos dentro de nuestro conjunto de datos que pueden resultar interesantes para analizar y/o comparar. No obstante, como se verá en el apartado consistente en la realización de pruebas estadásticas, no todos se utilizarán.

```
# Agrupación por genero
train.female <- train[train$Sex == "female",]
train.male <- train[train$Sex == "male",]

# Agrupación por puerta de embarque
train.c <- train[train$Embarked == "C",]
train.s <- train[train$Embarked == "S",]
train.q <- train[train$Embarked == "Q",]

# Agrupación por Parch
train.parch.cero <- train[train$Parch == "0",]
train.parch.mayor <- train[train$Parch > "0",]

# Agrupación por Parch
train.sibSp.cero <- train[train$SibSp == "0",]
```

```
train.sibSp.mayor <- train[train$SibSp > "0",]

#Usar el termino de tamaño de la familia sumando parch y SibSp
train$FamilySize <- train$SibSp + train$Parch +1;
```

Comprobación de la normalidad y homogeneidad de la varianza

Para la comprobación de que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, utilizaremos la prueba de normalidad de Anderson- Darling.

Asi, se comprueba que para que cada prueba se obtiene un p-valor superior al nivel de significación prefijado $\alpha = 0,05$. Si esto se cumple, entonces se considera que variable en cuestión sigue una distribución normal.

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

```
## Variables que no siguen una distribución normal:
## PassengerId, Survived, Pclass,
## Age, SibSp,
## Parch, Fare, FamilySize
```

Aplicación de pruebas estadásticas 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. Aplicar al menos tres métodos de análisis diferentes

Vamos a estudiar la homogeneidad de varianzas mediante la aplicación de un test de Fligner-Killeen. En este caso, estudiaremos esta homogeneidad en cuanto a la cabina del pasajero. En el siguiente test, la hipótesis nula consiste en que ambas varianzas son iguales.

```
fligner.test(Survived ~ Cabin , data = train)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Cabin
## Fligner-Killeen:med chi-squared = 88.127, df = 147, p-value = 1
```

Puesto que obtenemos un p-valor superior a 0,05, aceptamos la hipótesis de que las varianzas de ambas muestras son homogéneas. Detectamos la cabina de pasajero no es un factor que ha influido en la supervivencia de los pasajeros.

```
#fligner.test(Survived ~ Embarked, data = train)
```

Procedemos a realizar un análisis de correlación entre las distintas variables para determinar cuáles de ellas ejercen una mayor influencia sobre el precio final del vehículo. Para ello, se utilizará el coeficiente de correlación de Spearman, puesto que hemos visto que tenemos datos que no siguen una distribución normal.

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
# Calcular el coeficiente de correlación para cada variable cuantitativa
# con respecto al campo "Survived"
for (i in 3:(ncol(train) - 1)) {
  if (is.integer(train[,i]) | is.numeric(train[,i])) {
    spearman_test = cor.test(train[,i], train[,2], method = "spearman")
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    #z Add row to matrix
    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)] <- colnames(train)[i]
  }
}
print(corr_matrix)
```

```
##           estimate      p-value
## Pclass -0.33966794 1.687608e-25
## Age    -0.03327318 3.211623e-01
## SibSp   0.08887948 7.941431e-03
## Parch   0.13826563 3.453591e-05
## Fare    0.32373614 3.471228e-23
```

```
is_number <- sapply(train,is.numeric)
correlacion <-cor(train[,is_number])
correlacion
```

```
##              PassengerId     Survived      Pclass         Age       SibSp
## PassengerId  1.000000000 -0.005006661 -0.03514399  0.03907512 -0.05752683
## Survived    -0.005006661  1.000000000 -0.33848104 -0.06048851 -0.03532250
## Pclass      -0.035143994 -0.338481036  1.00000000 -0.41016308  0.08308136
## Age          0.039075120 -0.060488506 -0.41016308  1.00000000 -0.32367551
## SibSp       -0.057526834 -0.035322499  0.08308136 -0.32367551  1.00000000
## Parch       -0.001652012  0.081629407  0.01844267 -0.20944429  0.41483770
## Fare         0.012658219  0.257306522 -0.54949962  0.10408482  0.15965104
## FamilySize  -0.040142931  0.016638989  0.06599691 -0.32585599  0.89071167
##                    Parch        Fare  FamilySize
## PassengerId -0.001652012  0.01265822 -0.04014293
## Survived     0.081629407  0.25730652  0.01663899
```

```
## Pclass       0.018442671 -0.54949962  0.06599691
## Age         -0.209444286  0.10408482 -0.32585599
## SibSp        0.414837699  0.15965104  0.89071167
## Parch        1.000000000  0.21622494  0.78311078
## Fare         0.216224945  1.00000000  0.21713841
## FamilySize   0.783110775  0.21713841  1.00000000
```

Así, identificamos cuáles son las variables más correlacionadas con el precio en función de su proximidad con los valores -1 y +1. Teniendo esto en cuenta, queda patente cómo la variable más relevante en la supervivencia es la clase donde viajaba el pasajero (Pclass).

Nota. Para cada coeficiente de correlación se muestra también su p-valor asociado, puesto que éste puede dar información acerca del peso estadístico de la correlación obtenida.

¿Ha influido el genero en la supervivencia de los pasajeros?

Nos preguntamos si existen diferencias significativas en la supervivencia de los hombres en relación a las mujeres. Para responder a esta pregunta, siga los pasos que se detallan a continuación.

el cotraste de hipótesis de dos muestras sobre sobre la diferencia de medidas, el cual es unilateral atendiendo a la formulación de la hipótesis alternativa:

$H_0 : \mu_1 - \mu_2 = 0$

$H_1 : \mu_1 - \mu_2 < 0$

donde u1 es la media de la población de la que se extrae la primera muestra y u2 es la media de la población de la que extrae la segunda. Así, tomaremos $\alpha = 0, 05$.

Test de Shapiro

```
#Test d'igualtat de variances
#H0: F (rati de variàncies) = 1
#H0 : F diferent d'1
shapiro.test(train$Survived[train$Sex=="male"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  train$Survived[train$Sex == "male"]
## W = 0.47706, p-value < 2.2e-16
```

Según el test Shapiro Wilk, podemos asumir normalidad. Por lo tanto, aplicamos un test de dos muestras independientes para la diferencia de las medias. Aplicamos el caso de datos normales, con varianza desconocida. El test es bilateral (dos colas).

```
#Test d'igualtat de variances
#H0: F (rati de variàncies) = 1
#H0 : F diferent d'1
var.test(train$Survived[train$Sex=="male"], train$Survived[train$Sex=="female"], alternative = "two.sid
```

```
##
##  F test to compare two variances
##
## data:  train$Survived[train$Sex == "male"] and train$Survived[train$Sex == "female"]
## F = 0.7993, num df = 576, denom df = 313, p-value = 0.02218
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.6558632 0.9685628
## sample estimates:
## ratio of variances
##           0.799295
```

El resultado del test F es que podemos asumir igualdad de varianzas. Por lo tanto, aplicamos test t de dos muestras independientes para la diferencia de medias, varianzas desconocidas e iguales. El test es bilateral.

Calculo manual

```
nF<-nrow( train.male)
meanF<-mean(train.male$Survived)
sdF <- sd( train.male$Survived)
#
nM<-nrow( train.female )
meanM<-mean(train.female$Survived)
sdM <- sd( train.female$Survived)
#
s <-sqrt( ((nF-1)*sdF^2 + (nM-1)*sdM^2 )/(nF+nM-2) )
Sb <- s*sqrt(1/nF + 1/nM)
t <- (meanF-meanM)/ Sb
t
```

```
## [1] -19.29782
```

```
alfa <- (1-0.95)
tcritical <- qt( alfa/2, df=nF+nM-2, lower.tail=FALSE )
#two-sided
pvalue<-pt( abs(t), df=nF+nM-2, lower.tail=FALSE )*2
#Print info
info<-data.frame(nF,meanF,sdF,nM,meanM,sdM,t,tcritical,pvalue)

info
```

```
##    nF     meanF       sdF  nM     meanM       sdM         t tcritical
## 1 577 0.1889081 0.3917753 314 0.7420382 0.4382112 -19.29782  1.962636
##         pvalue
## 1 1.406066e-69
```

como el valor de p es 1.406066110^{-69} < 0.05 podemos rechazar la hipótesis nula. La respuesta seria que si ha influido el genero.

Aplicando el test no parametrico de Mann-Whitney

```
t.test(train.male$Survived, train.female$Survived, alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  train.male$Survived and train.female$Survived
## t = -18.672, df = 584.43, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.5043259
## sample estimates:
## mean of x mean of y
## 0.1889081 0.7420382
```

Puesto que obtenemos un p-valor menor que el valor de significación fijado, rechazamos la hipótesis nula. Por tanto, podemos concluir que, efectivamente, el genero de un pasajero ha influido en la supervivencia.

Modelo de regresión lineal

Tal y como se planteó en los objetivos de la actividad, resultará de mucho interés poder realizar predicciones sobre la supervivencia de un pasajero dadas sus características. Así, se calculará un modelo de regresión lineal utilizando regresores tanto cuantitativos como cualitativos con el que poder realizar las predicciones de la supervivencia o no.

Para obtener un modelo de regresión lineal considerablemente eficiente, lo que haremos será obtener varios modelos de regresión utilizando las variables que estén más correladas con respecto al precio, según la tabla obtenido en el apartado 5.3. Así, de entre todos los modelos que tengamos, escogeremos el mejor

```
# Regresores cuantitativos con mayor coeficiente
# de correlación con respecto a la supervivencia
legth = train$length
width = train$width
train.Age = train$Age
train.sibSp = train$SibSp
train.Parch = train$Parch
train.FamilySize = train$FamilySize
# Regresores cualitativos
train.Sex = train$Sex
train.Pclass = train$Pclass
train.Embarked = train$Embarked
train.Fare = train$Fare
# Variable a predecir
Survived = train$Survived
# Generación de varios modelos
modelo1 <- lm(Survived ~ train.Age +train.Fare + train.Sex + train.Embarked + train.sibSp + train.Parch
#modelo1 <- lm(Survived ~ train.Age + train.Pclass, data = train)
modelo2 <- lm(Survived ~ train.sibSp + train.Fare + train.Sex + train.Embarked + train.Parch + train.Pcl
modelo3 <- lm(Survived ~ train.Age + train.Fare + train.Sex + train.Embarked + train.Parch + train.Pclas
modelo4 <- lm(Survived ~ train.Age + train.Fare + train.Sex + train.Embarked + train.sibSp + train.Pclas
modelo5 <- lm(Survived ~ train.Age + train.Fare + train.Sex + train.Embarked + train.sibSp + train.Parch
```

```
modelo6 <- lm(Survived ~ train.Age + train.Fare + train.Sex + train.Embarked + train.FamilySize + train
modelo7 <- lm(Survived ~ train.Sex + train.Fare + train.Embarked + train.FamilySize + train.Pclass, data
```

Para los anteriores modelos de regresión lineal múltiple obtenidos, podemos utilizar el coeficiente de determinación para medir la bondad de los ajustes y quedarnos con aquel modelo que mejor coeficiente presente.

```
tabla.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
                               2, summary(modelo2)$r.squared,
                               3, summary(modelo3)$r.squared,
                               4, summary(modelo4)$r.squared,
                               5, summary(modelo5)$r.squared,
                               6, summary(modelo6)$r.squared,
                               7, summary(modelo7)$r.squared),
                               ncol = 2, byrow = TRUE)
colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes
```

```
##      Modelo       R^2
## [1,]      1 0.4038482
## [2,]      2 0.3780334
## [3,]      3 0.3939728
## [4,]      4 0.4033187
## [5,]      5 0.3485828
## [6,]      6 0.4026206
## [7,]      7 0.3775618
```

Modelo de regresión logística

Trabajaremos con la variable binaria (Survived) que indica la condición de sobrevivir o no. Estimar el modelo de regresión logística donde la variable dependiente es "Survived" y las explicativas son la capacidad pulmonar Fare, Age y Sex.

Evaluar si alguno de los regresores tiene influencia significativa (p-valor del contraste individual inferior al 5%).

```
Model.2.1=glm(Survived~Fare + Age + Sex, family=binomial, data=train)
summary(Model.2.1)
```

```
##
## Call:
## glm(formula = Survived ~ Fare + Age + Sex, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2889  -0.6127  -0.5794   0.8048   2.0031
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.805434   0.215966   3.729 0.000192 ***
## Fare         0.011554   0.002334   4.951 7.39e-07 ***
## Age         -0.006239   0.006121  -1.019 0.308108
```

```
## Sexmale     -2.401733   0.171454 -14.008  < 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: 1186.66  on 890  degrees of freedom
## Residual deviance:  883.27  on 887  degrees of freedom
## AIC: 891.27
##
## Number of Fisher Scoring iterations: 5
```

```
sel <- which(summary(Model.2.1)$coefficients[-1,4] < 0.05)
sel <- sel + 1
```

Ha sido algo significativo el test parcial sobre la variable Sex(Male). siendo la estimación de su coeficiente -2.4017.

Se podría decir que un individuo con más posibilidades de ser mujer, tiene mayor probabilidad de sobrevivir, ya que el signo negativo en Sex(Male) es un factor de "protección" ante el riesgo de sobrevivir. entonces cuantas menos posibilidades de ser hombre, mayor probabilidad de sobrevivir.

Creación del modelo, calidad del modelo y extracción de reglas en clasificación arboles de decisión

Nuestro objetivo es crear un árbol de decisión que permita analizar qué tipo de pasajero del Titanic tenía probabilidades de sobrevivir o no. Por lo tanto, la variable por la que clasificaremos es el campo de si el pasajero sobrevivió o no. De todas maneras, al imprimir las primeras y últimas 10 filas nos damos cuenta de que los datos están ordenados, por lo tanto, nos interesará "desordenarlos". Guardaremos los datos con el nuevo nombre como "train_random". Vamos a usar solo el juego de datos "train" dado que es el único que contiene la variable Survived.

```
#Mediante head() obtenemos las primeras filas de nuestro dataframe
head(train,10)
```

```
##    PassengerId Survived Pclass    Sex Age SibSp Parch    Fare Cabin
## 1            1        0      3   male  22     1     0  7.2500
## 2            2        1      1 female  38     1     0 71.2833   C85
## 3            3        1      3 female  26     0     0  7.9250
## 4            4        1      1 female  35     1     0 53.1000  C123
## 5            5        0      3   male  35     0     0  8.0500
## 6            6        0      3   male  21     0     0  8.4583
## 7            7        0      1   male  54     0     0 51.8625   E46
## 8            8        0      3   male   2     3     1 21.0750
## 9            9        1      3 female  27     0     2 11.1333
## 10          10        1      2 female  14     1     0 30.0708
##    Embarked PassengerId_imp Survived_imp Pclass_imp Sex_imp Age_imp
## 1         S           FALSE        FALSE      FALSE   FALSE   FALSE
## 2         C           FALSE        FALSE      FALSE   FALSE   FALSE
## 3         S           FALSE        FALSE      FALSE   FALSE   FALSE
## 4         S           FALSE        FALSE      FALSE   FALSE   FALSE
## 5         S           FALSE        FALSE      FALSE   FALSE   FALSE
## 6         Q           FALSE        FALSE      FALSE   FALSE    TRUE
```

```
## 7         S           FALSE        FALSE      FALSE   FALSE   FALSE
## 8         S           FALSE        FALSE      FALSE   FALSE   FALSE
## 9         S           FALSE        FALSE      FALSE   FALSE   FALSE
## 10        C           FALSE        FALSE      FALSE   FALSE   FALSE
##    SibSp_imp Parch_imp Fare_imp Cabin_imp Embarked_imp FamilySize
## 1      FALSE     FALSE    FALSE     FALSE        FALSE          2
## 2      FALSE     FALSE    FALSE     FALSE        FALSE          2
## 3      FALSE     FALSE    FALSE     FALSE        FALSE          1
## 4      FALSE     FALSE    FALSE     FALSE        FALSE          2
## 5      FALSE     FALSE    FALSE     FALSE        FALSE          1
## 6      FALSE     FALSE    FALSE     FALSE        FALSE          1
## 7      FALSE     FALSE    FALSE     FALSE        FALSE          1
## 8      FALSE     FALSE    FALSE     FALSE        FALSE          5
## 9      FALSE     FALSE    FALSE     FALSE        FALSE          3
## 10     FALSE     FALSE    FALSE     FALSE        FALSE          2
```

```
#Mediante head() obtenemos las últimas filas de nuestro dataframe
tail(train,10)
```

```
##     PassengerId Survived Pclass    Sex Age SibSp Parch    Fare Cabin
## 882         882        0      3   male  33     0     0  7.8958
## 883         883        0      3 female  22     0     0 10.5167
## 884         884        0      2   male  28     0     0 10.5000
## 885         885        0      3   male  25     0     0  7.0500
## 886         886        0      3 female  39     0     5 29.1250
## 887         887        0      2   male  27     0     0 13.0000
## 888         888        1      1 female  19     0     0 30.0000   B42
## 889         889        0      3 female  16     1     2 23.4500
## 890         890        1      1   male  26     0     0 30.0000  C148
## 891         891        0      3   male  32     0     0  7.7500
##     Embarked PassengerId_imp Survived_imp Pclass_imp Sex_imp Age_imp
## 882        S           FALSE        FALSE      FALSE   FALSE   FALSE
## 883        S           FALSE        FALSE      FALSE   FALSE   FALSE
## 884        S           FALSE        FALSE      FALSE   FALSE   FALSE
## 885        S           FALSE        FALSE      FALSE   FALSE   FALSE
## 886        Q           FALSE        FALSE      FALSE   FALSE   FALSE
## 887        S           FALSE        FALSE      FALSE   FALSE   FALSE
## 888        S           FALSE        FALSE      FALSE   FALSE   FALSE
## 889        S           FALSE        FALSE      FALSE   FALSE    TRUE
## 890        C           FALSE        FALSE      FALSE   FALSE   FALSE
## 891        Q           FALSE        FALSE      FALSE   FALSE   FALSE
##     SibSp_imp Parch_imp Fare_imp Cabin_imp Embarked_imp FamilySize
## 882     FALSE     FALSE    FALSE     FALSE        FALSE          1
## 883     FALSE     FALSE    FALSE     FALSE        FALSE          1
## 884     FALSE     FALSE    FALSE     FALSE        FALSE          1
## 885     FALSE     FALSE    FALSE     FALSE        FALSE          1
## 886     FALSE     FALSE    FALSE     FALSE        FALSE          6
## 887     FALSE     FALSE    FALSE     FALSE        FALSE          1
## 888     FALSE     FALSE    FALSE     FALSE        FALSE          1
## 889     FALSE     FALSE    FALSE     FALSE        FALSE          4
## 890     FALSE     FALSE    FALSE     FALSE        FALSE          1
## 891     FALSE     FALSE    FALSE     FALSE        FALSE          1
```

```r
set.seed(666)
#Queremos "desordenar" los datos
train_random <- train[sample(nrow(train)),]
```

Vamos a separar las columnas que consideramos más representaticas y la variable por la que clasificaremos que será si ha sobrevivido o no.

```r
#la variable por la que clasificaremos es el campo de si el pasajero sobrevivió o no,
#que está en la cuarta columna.

train_random$Pclass<-as.factor(train_random$Pclass)
train_random$Survived<-as.factor(train_random$Survived)

set.seed(666)
trainy <-train_random[,c(2)] #SURVIVED
trainX <- train_random[,c(3,4,9,10)] #Resto de variables

levels(trainX$Cabin)[1] = "missing"
levels(trainX$Embarked)[1] = "missing"
```

Ejecutamos el modelo.

```r
#Se crea el arbol de decision usando los datos de Entrenamiento.
model_class <- C50::C5.0(trainX, trainy, rules=TRUE )
summary(model_class)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy, rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Jun 11 20:20:04 2019
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 891 cases (5 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (577/109, lift 1.3)
##  Sex = male
##  ->  class 0  [0.810]
##
## Rule 2: (491/119, lift 1.2)
##  Pclass = 3
##  ->  class 0  [0.757]
##
## Rule 3: (170/9, lift 2.5)
##  Pclass in {1, 2}
##  Sex = female
##  ->  class 1  [0.942]
```

```
##
## Rule 4: (111/18, lift 2.2)
##  Sex = female
##  Embarked in {C, Q}
##  ->  class 1  [0.832]
##
## Default class: 0
##
##
## Evaluation on training data (891 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       4  168(18.9%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##     523    26    (a): class 0
##     142   200    (b): class 1
##
##
##  Attribute usage:
##
##   90.12% Sex
##   74.19% Pclass
##   12.46% Embarked
##
##
## Time: 0.0 secs
```

Errors muestra el número y porcentaje de casos mal clasificados, erróneamente 168 de los 891 casos dados, una tasa de error del 18.9%.

A partir del árbol de decisión que hemos modelado, se pueden extraer las siguientes reglas de decisión (gracias a rules=TRUE podemos imprimir las reglas directamente):

SEX = "male" → Muere Validez: 81%

CLASS = "3a" → Muere Validez: 75,7%

CLASS = (1,2) y SEX = "female" → Sobrevive Validez: 94,2%

CLASS (C,Q) y SEX = "female" → Sobrevive Validez: 83,2%

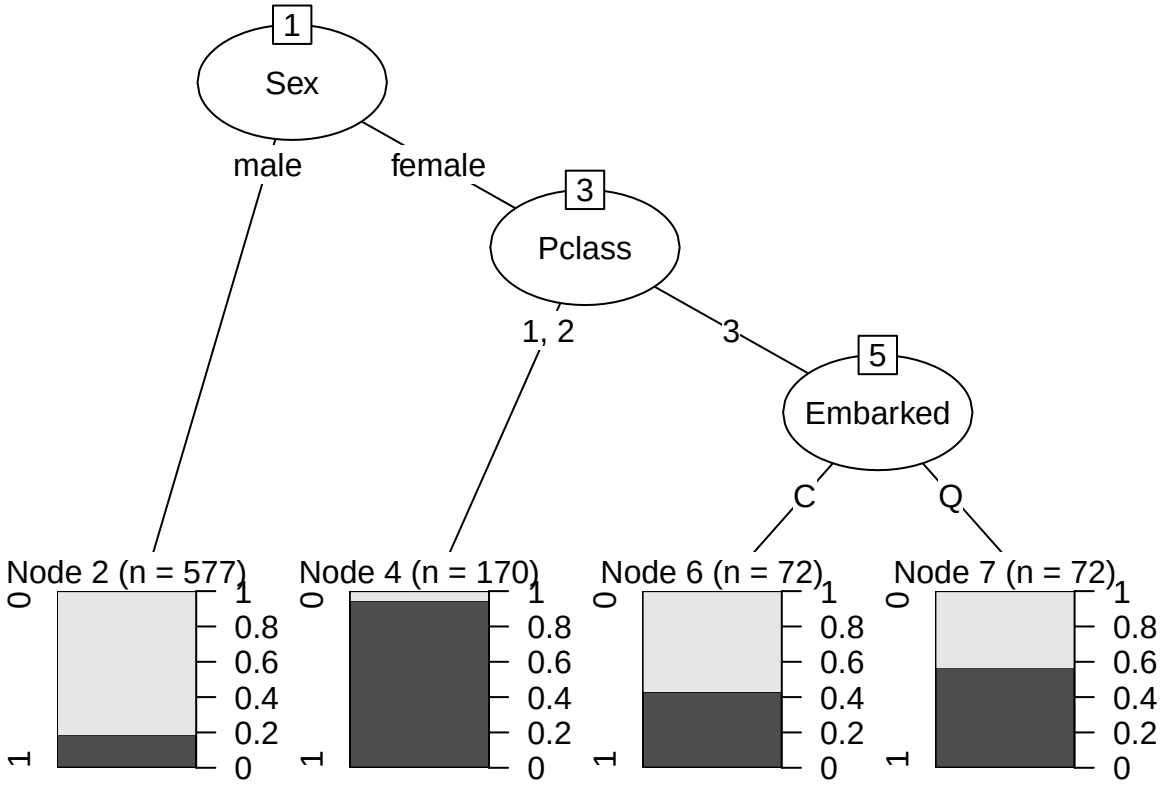
Por tanto podemos concluir que el conocimiento extraido se resume en "las mujeres que viajaban en 1a y 2a clase sobreviven". También que "Las mujeres que embarcaron en Cherbourg o en Queenstown sobreviven", esta última con un porcentaje de predicción menor.

Visualizaremos el árbol en el siguiente apartado.

Representación de los resultados a partir de tablas y gráficas

Generación del árbol.

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



Discretizamos cuando tiene sentido y en función de cada variable.

```
# ¿Para qué variables tendría sentido un proceso de discretización?
apply(train,2, function(x) length(unique(x)))
```

```
##     PassengerId        Survived          Pclass             Sex
##             891               2               3               2
##             Age           SibSp           Parch            Fare
##              88               7               7             248
##           Cabin        Embarked PassengerId_imp    Survived_imp
##             148               3               1               1
##      Pclass_imp        Sex_imp        Age_imp      SibSp_imp
##               1               1               2               1
##       Parch_imp        Fare_imp       Cabin_imp    Embarked_imp
##               1               1               1               1
##      FamilySize
##               9
```

```
# Discretizamos las variables con pocas clases
cols<-c("Survived","Pclass","Sex","Embarked")
for (i in cols){
  train[,i] <- as.factor(train[,i])
```

```
}

# Después de los cambios, analizamos la nueva estructura del juego de datos
str(train)
```

```
## 'data.frame':    891 obs. of  21 variables:
##  $ PassengerId    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived       : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass         : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex            : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age            : num  22 38 26 35 35 21 54 2 27 14 ...
##  $ SibSp          : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch          : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare           : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin          : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked       : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
##  $ PassengerId_imp: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Survived_imp   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Pclass_imp     : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Sex_imp        : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Age_imp        : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
##  $ SibSp_imp      : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Parch_imp      : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Fare_imp       : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Cabin_imp      : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Embarked_imp   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ FamilySize     : num  2 2 1 2 1 1 1 5 3 2 ...
```

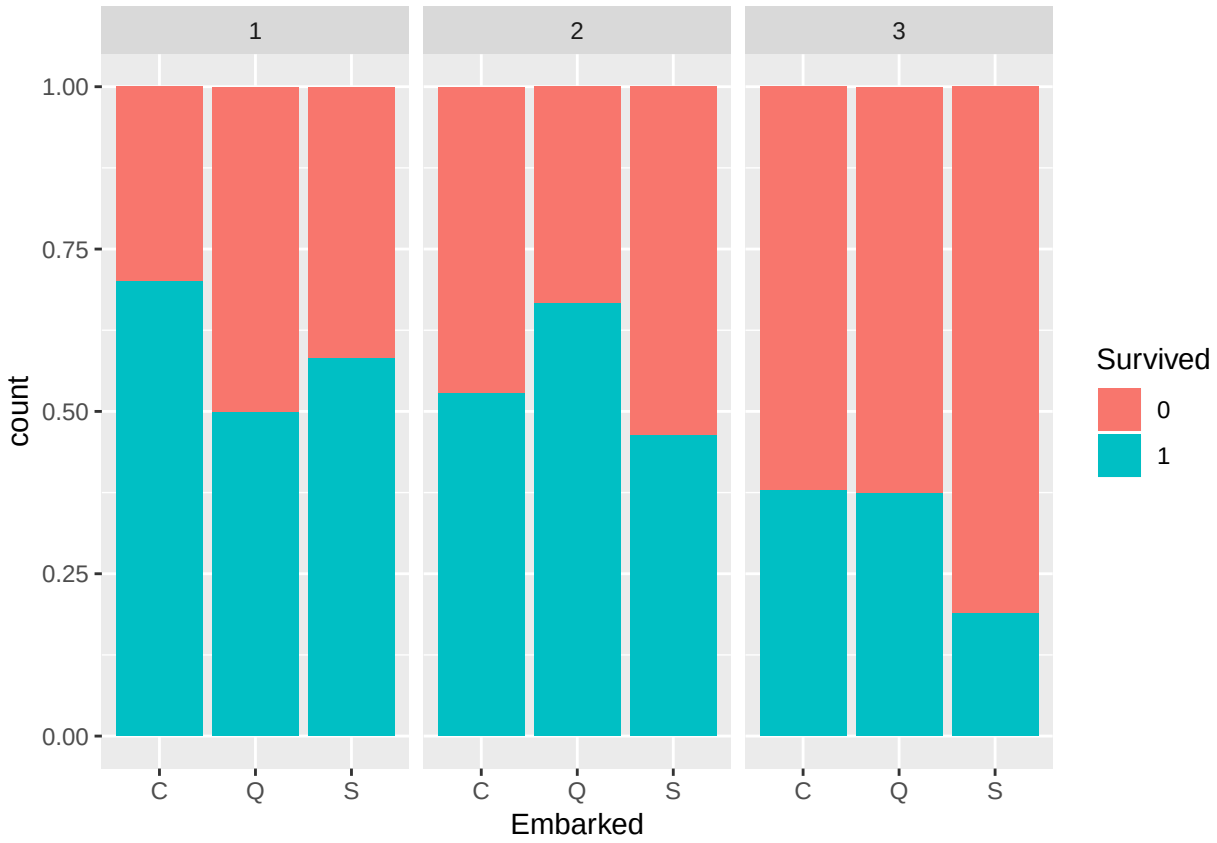
Obtenemos una matriz de porcentages de frecuencia.
Vemos, por ejemplo que la probabilidad de sobrebivir si se embarcó en “C” es de un 55,88%

```
filas=dim(train)[1]
t<-table(train[1:filas,]$Embarked,train[1:filas,]$Survived)
for (i in 1:dim(t)[1]){
    t[i,]<-t[i,]/sum(t[i,])*100
}
t
```

```
##
##            0        1
##
##   C 44.11765 55.88235
##   Q 61.03896 38.96104
##   S 66.30435 33.69565
```

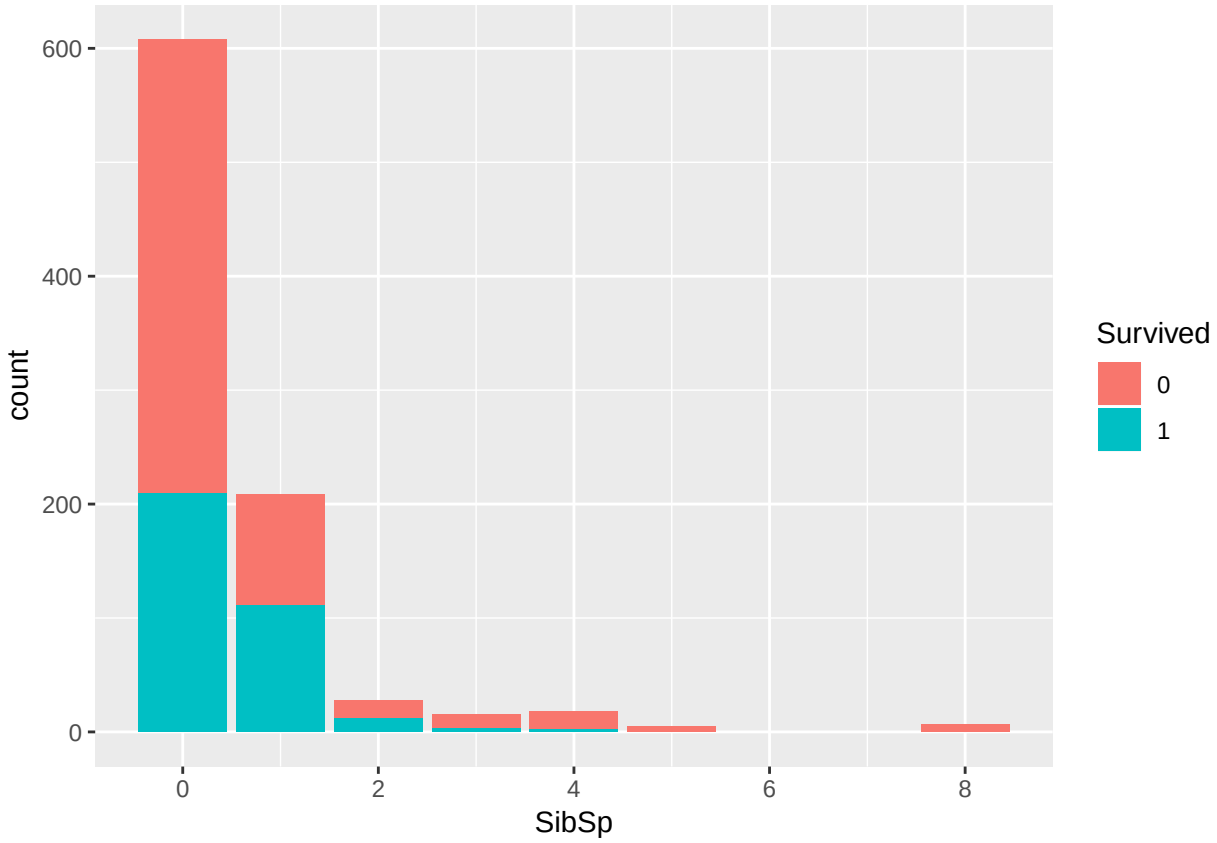
Veamos ahora como en un mismo gráfico de frecuencias podemos trabajar con 3 variables: Embarked, Survived y Pclass.

```
# Now, let's devide the graph of Embarked by Pclass:
ggplot(data = train[1:filas,],aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+facet_wrap(~Pclas
```

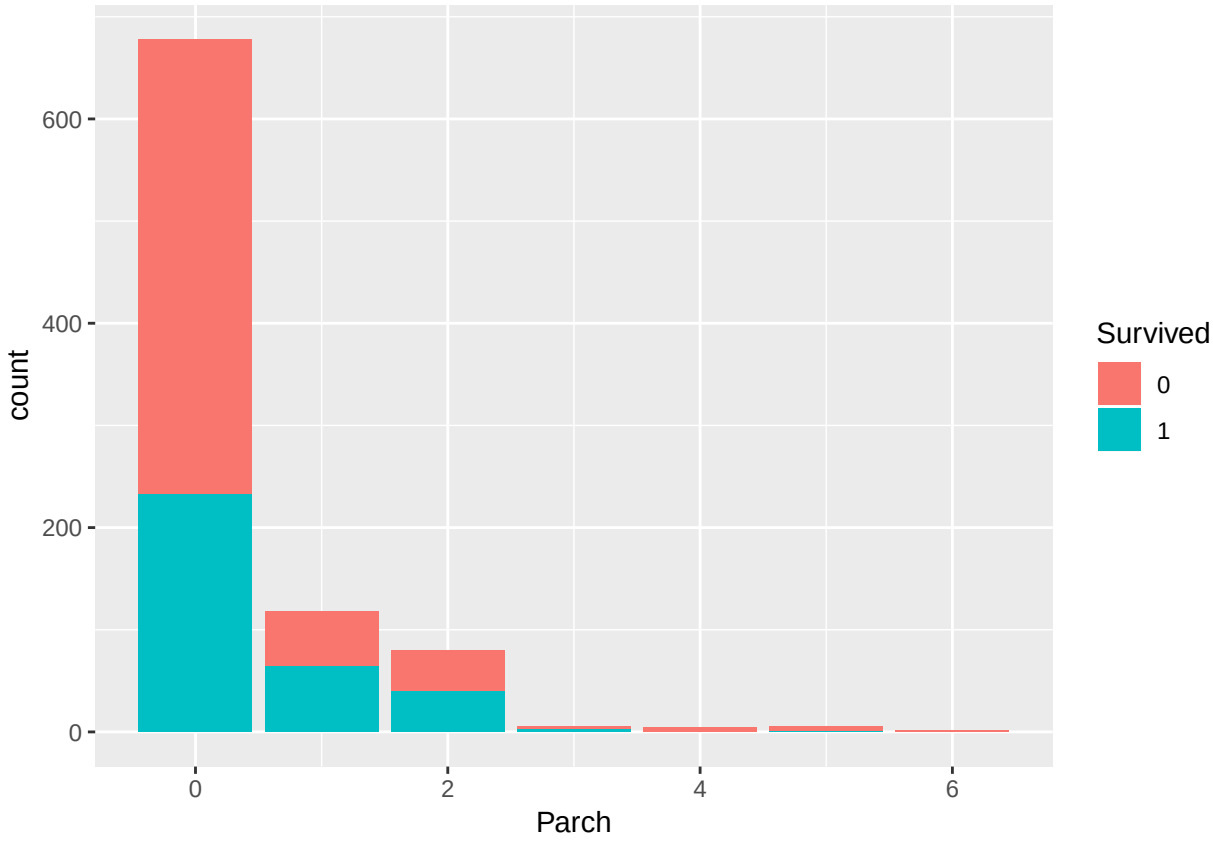


Comparemos ahora dos gráficos de frecuencias: Survived-SibSp y Survived-Parch

```
# Survivial como función de SibSp y Parch
ggplot(data = train[1:filas,],aes(x=SibSp,fill=Survived))+geom_bar()
```



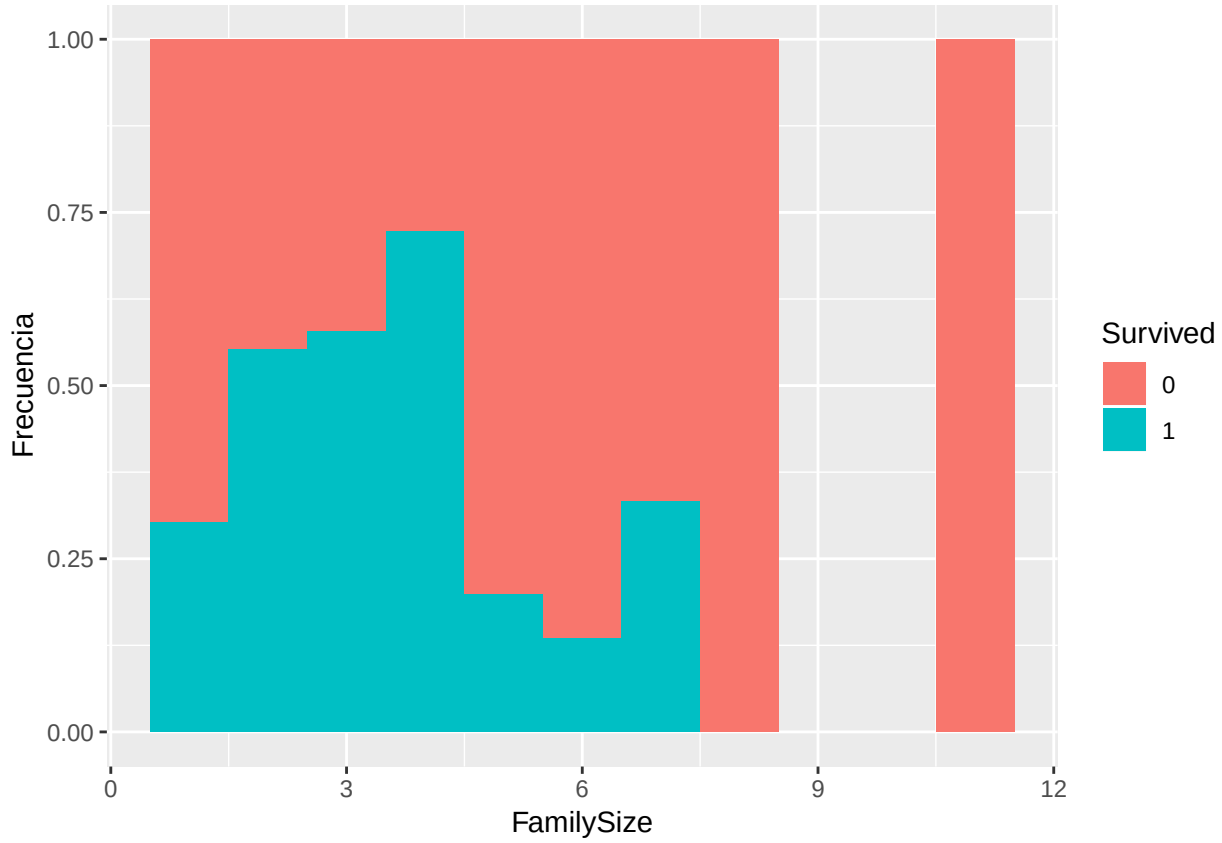

```
ggplot(data = train[1:filas,],aes(x=Parch,fill=Survived))+geom_bar()
```




```
# Vemos como las forma de estos dos gráficos es similar. Este hecho nos puede indicar presencia de corre
```

Veamos un ejemplo de construcción de una variable nueva: Tamaño de familia

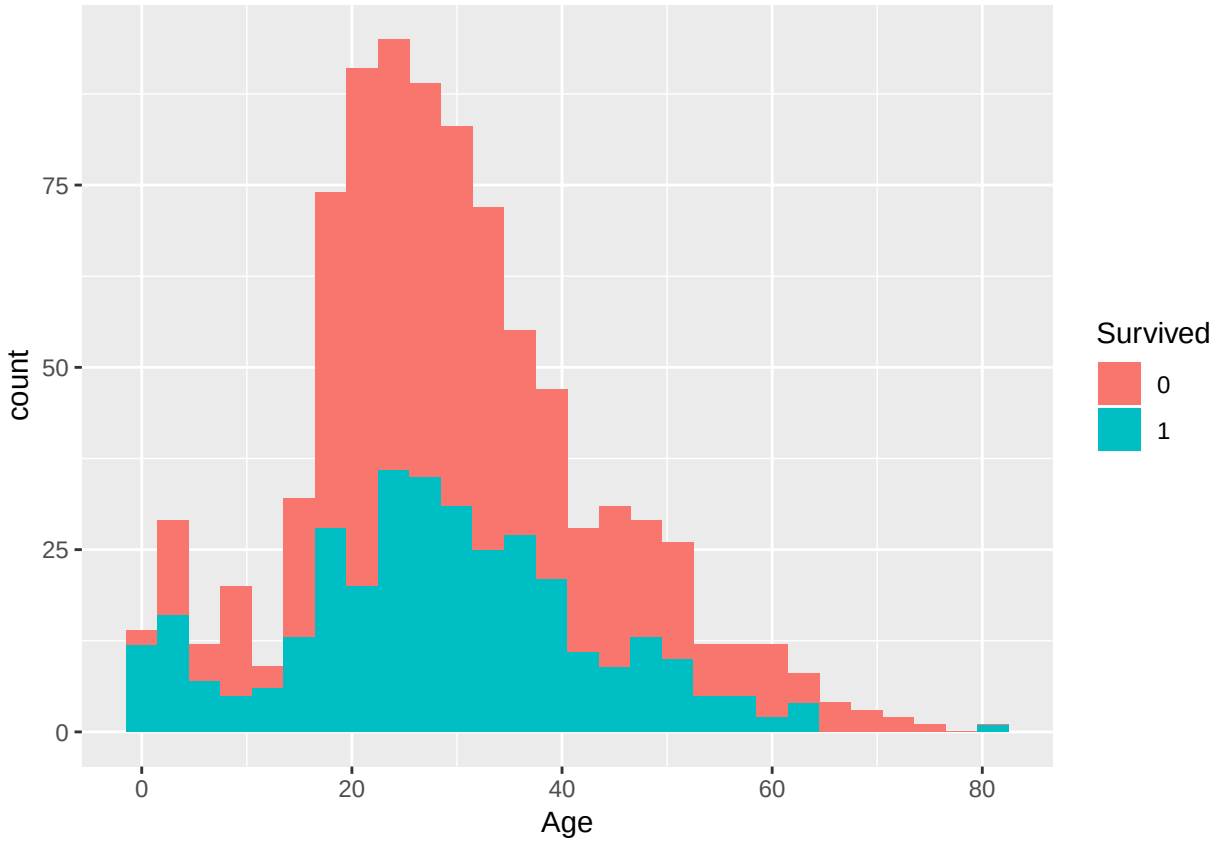
```
train1<-train[1:filas,]
ggplot(data = train1[!is.na(train[1:filas,]$FamilySize),],aes(x=FamilySize,fill=Survived))+geom_histogra
```




```
# Observamos como familias de entre 2 y 6 miembros tienen más del 50% de posibilidades de supervivencia
```

Veamos ahora dos gráficos que nos compara los atributos Age y Survived.
Observamos como el parámetro position="fill" nos da la proporción acumulada de un atributo dentro de otro

```
# Survival como función de age:
ggplot(data = train1[!(is.na(train[1:filas,]$Age)),],aes(x=Age,fill=Survived))+geom_histogram(binwidth =
```




```
ggplot(data = train1[!is.na(train[1:filas,]$Age),],aes(x=Age,fill=Survived))+geom_histogram(binwidth = 
```

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

Aunque consideramos que los datos no eran suficientes en volumen para extraer ideas más contundentes, se han podido extraer ideas como:

1. Las mujeres que viajaban en primera y segunda clase prácticamente sobrevivieron.
2. Que murieron más hombres que mujeres.
3. Que murió mucha gente que viajaba en 3a clase.

Calidad del ajuste

Calcular la matriz de confusión del mejor modelo del apartado 2.3 suponiendo un umbral de discriminación del 75 %. Observad cuantos falsos negativos hay e interpretar qué es un falso negativo en este contexto. Hacer lo mismo con los falsos positivos.

```
train$prob_Survived= predict(modelo6, train, type="response")
train$pred_Survived <- ifelse(train$prob_Survived > 0.75,1,0)
table(train$Survived, train$pred_Survived)
```

```
##
```

```
##       0   1
##   0 544   5
##   1 211 131
```

Un falso negativo en este concepto corresponde a los viajeros que se han predecido como no sobrevivientes cuando realmente sí han sobrevivido. Tenemos 211 falsos negativos.

Un falso positivo en este concepto corresponde a los viajeros que se han predicho como sobrevivientes y no lo han sido realmente. Tenemos 5 falsos positivos.

Conclusiones

A pesar de la limitación de los datos, el análisis demuestra que hay diferencias significativas entre la clase, el sexo y las tarifas del pasaje mientras que no hay diferencias bastante significativas en relación a la edad o numero de miembros de familia. La clase es un buen predictor de la supervivencia. En general, además del genero, se ha observado que la combinación de genero con la clase ha podido influir en la supervivencia en el caso de ser mujer que viaja en la clase 1 o 2. Se podría ampliar la muestra de estudio para comprobar si los resultados apuntados en estas conclusiones observan también con una muestra mayor.