

M2.851 - Tipología y ciclo de vida de los datos Practica 2

Francisco Javier Melchor y Enrique Otero

04/01/2021

Contents

Paquetes	2
Presentación	3
Competencias	3
Objetivos	3
Descripción de la Práctica a realizar	4
Preguntas y desarrollo de respuestas	4
1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?	4
2. Integración y selección de los datos de interés a utilizar	5
3. Limpieza de datos	5
3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos? .	7
3.2 Identificación y tratamiento de valores extremos	9
4. Análisis de los datos.	14
4.2. Comprobación de la normalidad y homogeneidad de la varianza.	31
4.2.1 Comprobación de la normalidad y la homogeneidad de la varianza en muestras de supervivencia por la clase del pasajero	32
4.2.2 Comprobación de la normalidad y la homogeneidad de la varianza en muestras de supervivencia por el sexo del pasajero	35
4.1 Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).	39
4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos.	40
4.3.1 Tests de hipótesis	40
4.3.2 Modelos	41
5. Representación de los resultados a partir de tablas y gráficas	52
5.2 Tabla resumen de los resultados obtenidos en los modelos predictivos	52
6. Conclusiones obtenidas	53
Contribuciones	53

Paquetes

Los paquetes que se van a utilizar para el desarrollo de esta actividad, son los siguientes:

```
if(!require(ggplot2)){
  install.packages("ggplot2")
  library(ggplot2)
}

if(!require(arc)){
  install.packages("arc")
  library(arc)
}

if(!require(ggcorrplot)){
  install.packages("ggcorrplot")
  library(ggcorrplot)
}

if(!require(ggpubr)){
  install.packages("ggpubr")
  library(ggpubr)
}

if(!require(BSDA)){
  install.packages("BSDA")
  library(BSDA)
}

if(!require(tidyverse)){
  install.packages("tidyverse")
  library(tidyverse)
}

if(!require(lattice)){
  install.packages("lattice")
  library(lattice)
}

if(!require(caret)){
  install.packages("caret")
  library(caret)
}

if(!require(dplyr)){
  install.packages("dplyr")
  library(dplyr)
}

if(!require(lattice)){
  install.packages("lattice")
  library(lattice)
}

if(!require(plyr)){
  install.packages("plyr")
}
```

```
library(plyr)
}

if(!require(rsample)){
  install.packages("rsample")
  library(rsample)
}

if(!require(yardstick)){
  install.packages("yardstick")
  library(yardstick)
}
```

Presentación

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas. Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace Github (<https://github.com>) donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo. Podéis utilizar la Wiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github. Aunque no se trata del mismo enunciado, los siguientes ejemplos de ediciones anteriores os pueden servir como guía:

- Ejemplo: <https://github.com/Bengis/nba-gap-cleaning>
- Ejemplo complejo (archivo adjunto).

Competencias

En esta práctica se desarrollan las siguientes competencias del Máster de Data Science:

A. Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.

B.Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Objetivos

Los objetivos concretos de esta práctica son:

1. **Aprender** a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
2. **Saber** identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
3. **Aprender** a analizar los datos adecuadamente para abordar la información contenida en los datos.
4. **Identificar** la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
5. **Actuar** con los principios éticos y legales relacionados con la manipulación de datos en Tipología y ciclo de vida de los datos Práctica 2 pág 2 función del ámbito de aplicación.
6. **Desarrollar** las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.

7. **Desarrollar** la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

Descripción de la Práctica a realizar

El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle (<https://www.kaggle.com>). Algunos ejemplos de dataset con los que podéis trabajar son:

Red Wine Quality (<https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009>)

Titanic: Machine Learning from Disaster (<https://www.kaggle.com/c/titanic>)

El último ejemplo corresponde a una competición activa de Kaggle de manera que, opcionalmente, podéis aprovechar el trabajo realizado durante la práctica para entrar en esta competición.

Preguntas y desarrollo de respuestas

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

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

Para nuestra practica especifica hemos elegido el dataset asociado al ejemplo de Kaggle:

Titanic: Machine Learnin from Disaster

El hundimiento del Titanic es uno de los naufragios más trágicos de la historia.

El 15 de abril de 1912, durante su viaje inaugural, el RMS Titanic, ampliamente considerado “insumergible”, se hundió tras chocar con un iceberg. Desafortunadamente, no había suficientes botes salvavidas para todos a bordo, lo que resultó en la muerte de 1502 de los 2224 pasajeros y la tripulación.

Si bien hubo algún elemento de suerte involucrado en sobrevivir, parece que algunos grupos de personas tenían más probabilidades de sobrevivir que otros.

En este desafío, se pide crear un modelo predictivo que responda a la pregunta: “¿Qué tipo de personas tenían más probabilidades de sobrevivir?” utilizando datos de pasajeros (es decir, nombre, edad, sexo, clase socioeconómica, etc.). En términos de analítica se trata de un problema de **clasificación**, esto es, usar esas variables independientes para predecir la categoría a la que pertenece cada registro, o, dicho de otra manera, predecir si un pasajero dado va a sobrevivir o no.

El enlace de descarga de este ejemplo contiene tres ficheros:

train.csv. Se trata del dataset *test* sobre el que entrenamos a nuestros modelos de analítica.

test.csv. Es el dataset donde probamos, con nuevos datos, nuestros modelos de analítica.

Para este caso no se incluye el resultado (variable dependiente *Survived*) ya que es el objetivo del concurso.

gender_submission.csv. Contiene un ejemplo de como debe presentarse el formato de salida con el resultado de nuestros modelos. Se trata de un conjunto de predicciones que asumen que todas y solo mujeres sobreviven.

Según los datos proporcionados en la web de Kaggle, las variables de los datasets son:

Variable	Definition	Key
Survived	If passenger survived	0 = No, 1 = Yes
Pclass	Ticket Class	1 = 1st, 2 = 2nd, 3 = 3rd

Variable	Definition	Key
Sex	Passenger Sex	
Age	Passenger Age in years	
SibSp	Number of sibings/spouses of the passenger aboard the Titanic	
Parch	Number of parents/children of the passenger aboard the Titanic	
Ticket	Ticket number	
Fare	Passenger fare	
Cabin	Cabin Number	
Embarked	Port of Embarkation	

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

En este caso al estar realizando el análisis de un único dataset, no es necesario realizar ninguna integración de distintas fuentes, pues sólo existe una.

Por otro lado, con respecto a la selección, en este caso al no ser un dataset excesivamente grande y al no tener fijado un objetivo diferente que analizar todo el conjunto de sus datos y no una parte de ellos, no se realizará ninguna selección del dataset de origen ni se acotará el mismo.

3. Limpieza de datos

En esta sección realizaremos una limpieza del dataset incluido en el fichero **train.csv**.

Para ello, lo primero que realizaremos es la lectura del fichero **train.csv** y comprobar como han sido interpretadas por R las variables que forman el mismo.

```
ttc <- read.csv("./Data/train.csv",na.strings=c("", " ", "NA"))
head(ttc)
```

```
##   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 Parch
## 1                                Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                                Heikkinen, Miss. Laina female  26     0     0
## 4 Futrelle, Mrs. Jacques Heath (Lily May Peel) female    35     1     0
## 5                                Allen, Mr. William Henry   male  35     0     0
## 6                                Moran, Mr. James         male  NA     0     0
##
##   Ticket     Fare Cabin Embarked
## 1  A/5 21171  7.2500 <NA>      S
## 2    PC 17599 71.2833   C85      C
## 3 STON/O2. 3101282  7.9250 <NA>      S
## 4    113803 53.1000  C123      S
## 5    373450  8.0500 <NA>      S
## 6    330877  8.4583 <NA>      Q
```

```
str(ttc)
```

```
## 'data.frame':   891 obs. of  12 variables:
```

```
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Survived : int 0 1 1 1 0 0 0 0 1 1 ...
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex : chr "male" "female" "female" "female" ...
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr NA "C85" NA "C123" ...
## $ Embarked : chr "S" "C" "S" "S" ...
```

Contamos con 891 observaciones de las 12 variables descritas al inicio de esta seccion.

Como se puede observar, la mayoría de las variables han sido interpretadas correctamente por R, pero tanto la variable **Sex** como la variable **Embarked**, han sido formateadas como variables de tipo carácter (chr) y realmente son variables categóricas. Por otro lado, la variable **Survived** y **Pclass** han sido interpretadas como variables numéricas cuando realmente son variables categóricas. Procedemos a continuación a convertir las variables nombradas.

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

```
str(ttc)
```

```
## 'data.frame': 891 obs. of 12 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 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr NA "C85" NA "C123" ...
## $ Embarked : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

Por otro lado, la variable **Age**, ha sido interpretada por R como una variable numérica y no como una variable entera, lo que indica que esta posee valores decimales. Se procede a continuación a verificar cuantos casos hay de valores decimales en la variable Age.

```
decimalAges<-c()
```

```
for (i in 1:(nrow(ttc))){
  if(!is.na(ttc$Age[i])){
    if(as.integer(ttc$Age[i]) != ttc$Age[i])
      decimalAges<-c(decimalAges,ttc$Age[i])
  }
}
decimalAges
```

```
## [1] 28.50 0.83 14.50 70.50 32.50 32.50 36.50 55.50 40.50 45.50 20.50 23.50
```

```
## [13] 0.92 45.50 0.75 40.50 0.75 24.50 28.50 0.67 30.50 0.42 30.50 0.83
## [25] 34.50
```

```
length(decimalAges)
```

```
## [1] 25
```

Como se puede observar, existen 25 casos de edades decimales.

Para solventar esta problemática sin perder datos, procederemos a realizar un redondeo al entero más cercano y en caso de ser un valor menor que 1, lo redondearemos a uno directamente. Procedemos a continuación a realizar dicha conversión:

```
roundValues <- function(x){
  if (!is.na(x['Age'])){
    if(x['Age'] < 1)
      x['Age'] = 1
    else
      x['Age'] = round(as.numeric(x['Age']))
  }
  return(x['Age'])
}
```

```
ttc$Age <- apply(ttc,1,roundValues)
ttc$Age <- as.integer(ttc$Age)
str(ttc)
```

```
## 'data.frame': 891 obs. of 12 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 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 1 1 ...
## $ Age : int 22 38 26 35 35 NA 54 1 27 14 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr NA "C85" NA "C123" ...
## $ Embarked : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

Una vez que todas las variables han sido interpretadas correctamente, podemos proceder a realizar la limpieza y el procesamiento de los datos que contiene este dataframe.

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

A continuación se procede a comprobar si existen valores nulos o elementos vacíos en el dataframe a analizar:

```
colSums(is.na(ttc))
```

```
## PassengerId Survived Pclass Name Sex Age
##          0         0         0         0         0      177
## SibSp      Parch Ticket Fare Cabin Embarked
##          0         0         0         0        687         2
```

Como se puede observar, existen valores faltantes en las columnas **Age**, **Cabin** y **Embarked**.

Para responder a la pregunta de *¿Cómo gestionarías cada uno de estos casos?*, primero hay que analizar el número total de filas del dataset que se está analizando:

```
nrow = nrow(ttc)
```

El número total de filas con las que cuenta el dataset son: 891

Teniendo en cuenta la dimensión del dataframe y el número de valores faltantes, procedemos a realizar las siguientes consideraciones:

- En el caso de la variable **Age**, al contar con una proporción de 19% de valores faltantes, al ser una proporción baja, se realizará una imputación de dichos valores. Dicha imputación será a través del estimador de la mediana, para evitar sesgos causados por valores atípicos, y dicha imputación se dividirá por clases, es decir, se calcularán la mediana de edad resultante en cada una de las clases y dependiendo de si el valor faltante pertenece a una clase u otra se le imputará la mediana resultante de la edad en dicha clase.
- En el caso de la variable **Cabin**, al tratarse de más de un 70% de valores nulos o no válidos, dicha variable será eliminada del conjunto de datos a tratar, ya que no tenemos suficiente información en la que basarnos (un 30% de los casos) para realizar una imputación.
- Por último, para la variable **Embarked**, al tratarse únicamente de 2 casos con respecto al total que son 'r nrow, se eliminarán aquellas filas con dicho valor a nulo, pues al ser una cantidad tan pequeña, no merece la pena realizar una imputación.

A continuación procedemos a realizar los cambios comentados:

Primero comenzaremos con la eliminación de la variable **Cabin**

```
ttc$Cabin <- NULL  
head(ttc)
```

```
## 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 Parch  
## 1              Braund, Mr. Owen Harris   male  22      1      0  
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38      1      0  
## 3              Heikkinen, Miss. Laina female  26      0      0  
## 4 Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35      1      0  
## 5              Allen, Mr. William Henry   male  35      0      0  
## 6              Moran, Mr. James         male  NA      0      0  
##  
##      Ticket      Fare Embarked  
## 1      A/5 21171   7.2500      S  
## 2      PC 17599  71.2833      C  
## 3 STON/O2. 3101282   7.9250      S  
## 4      113803  53.1000      S  
## 5      373450   8.0500      S  
## 6      330877   8.4583      Q
```

Ahora procederemos a eliminar aquellas filas donde la variable **Embarked** tiene un valor no válido:

```
ttc <- ttc[!is.na(ttc$Embarked),]  
colSums(is.na(ttc))
```

```
## PassengerId      Survived      Pclass      Name      Sex      Age
```



```
##          0          0          0          0          0          177
##      SibSp      Parch      Ticket      Fare      Embarked
##          0          0          0          0          0
```

Por último, procedemos a realizar la imputación de la variable **Age**:

```
imputationFunct <- function(x){
  if (is.na(x["Age"])){
    x["Age"]<- median(ttc$Age[ttc$Pclass==x["Pclass"] & !is.na (ttc$Age)])
  } else{
    x<-x
  }
  return (x["Age"])
}

ttc$Age <- apply(ttc,1,imputationFunct)
ttc$Age <- as.numeric(ttc$Age)
sapply(ttc,function(x) sum(is.na(x)))
```

```
## PassengerId      Survived      Pclass      Name      Sex      Age
##          0          0          0          0          0          0
##      SibSp      Parch      Ticket      Fare      Embarked
##          0          0          0          0          0
```

3.2 Identificación y tratamiento de valores extremos

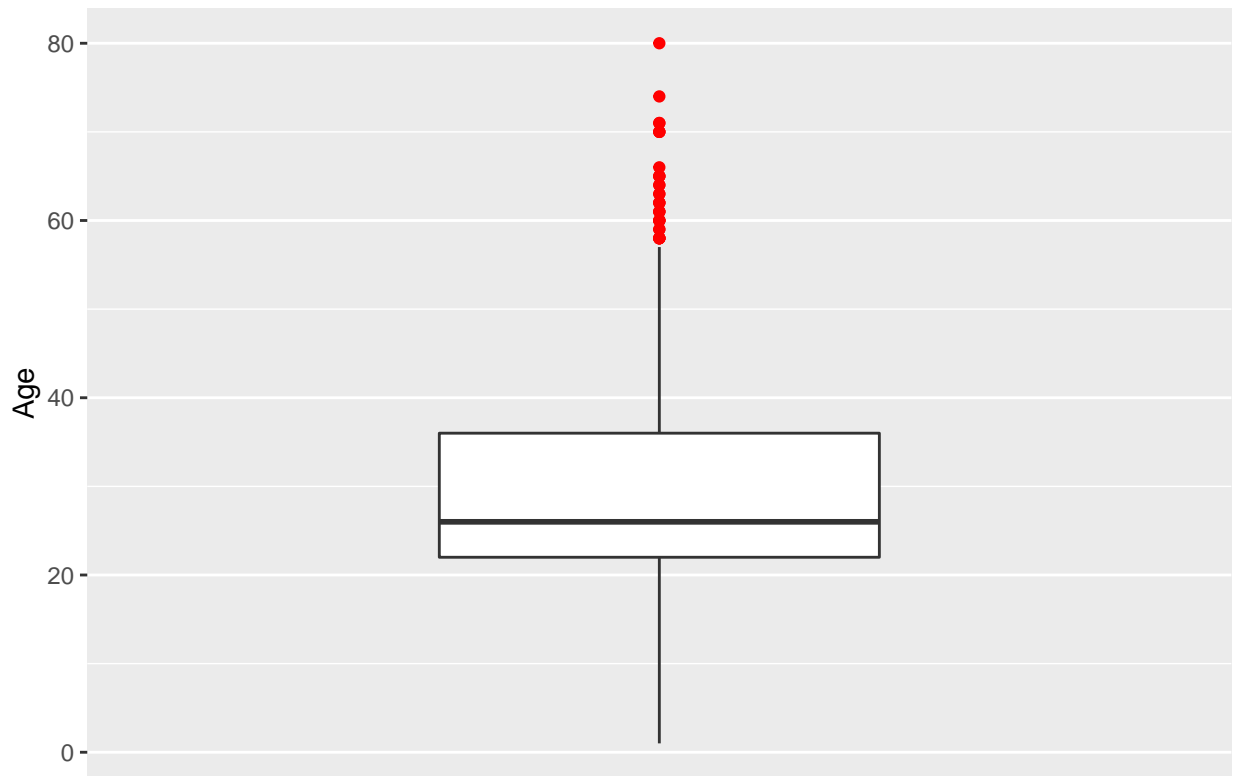
Una vez analizados y resueltos los valores faltantes del dataset a analizar, se procede a comprobar si existen valores atípicos en el mismo. Para ello, primero representaremos las variables numéricas con un diagrama de cajas y bigotes, lo cual nos permitirá visualizar gráficamente a simple vista si existen valores atípicos.

```
# Boxplot para la variable Age
summary(ttc$Age)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.0   22.0   26.0   28.8   36.0   80.0
```

```
ggplot(data = ttc, aes(y = Age)) +
  geom_boxplot(outlier.colour = "red") +
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
  ggtitle("Passenger Age Boxplot") +
  theme(plot.title = element_text(size = 20, hjust = 0.5))
```

Passenger Age Boxplot

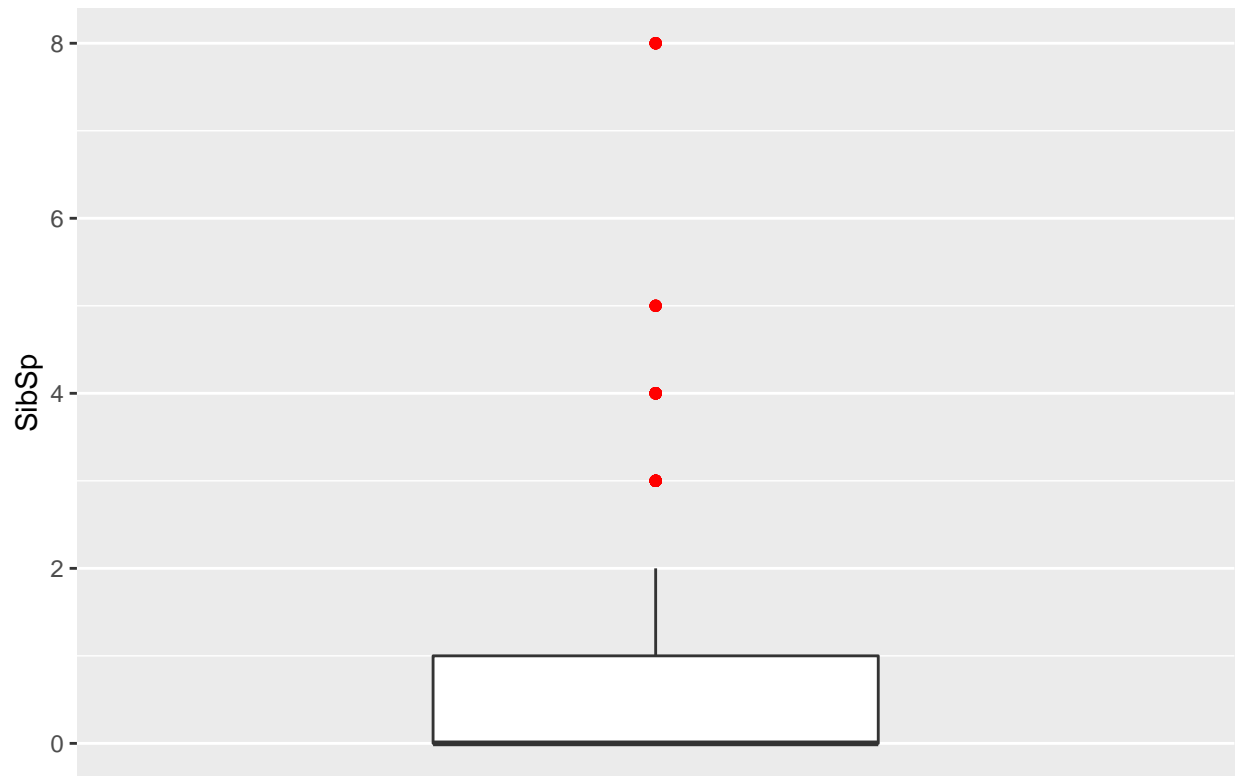


```
# Boxplot para la variable SibSp  
summary(ttc$SibSp)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## 0.0000  0.0000  0.0000  0.5242  1.0000  8.0000
```

```
ggplot(data = ttc, aes(y = SibSp)) +  
  geom_boxplot(outlier.colour = "red") +  
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +  
  ggtitle("Passenger sibings/spouses Boxplot") +  
  theme(plot.title = element_text(size = 20, hjust = 0.5))
```

Passenger siblings/spouses Boxplot



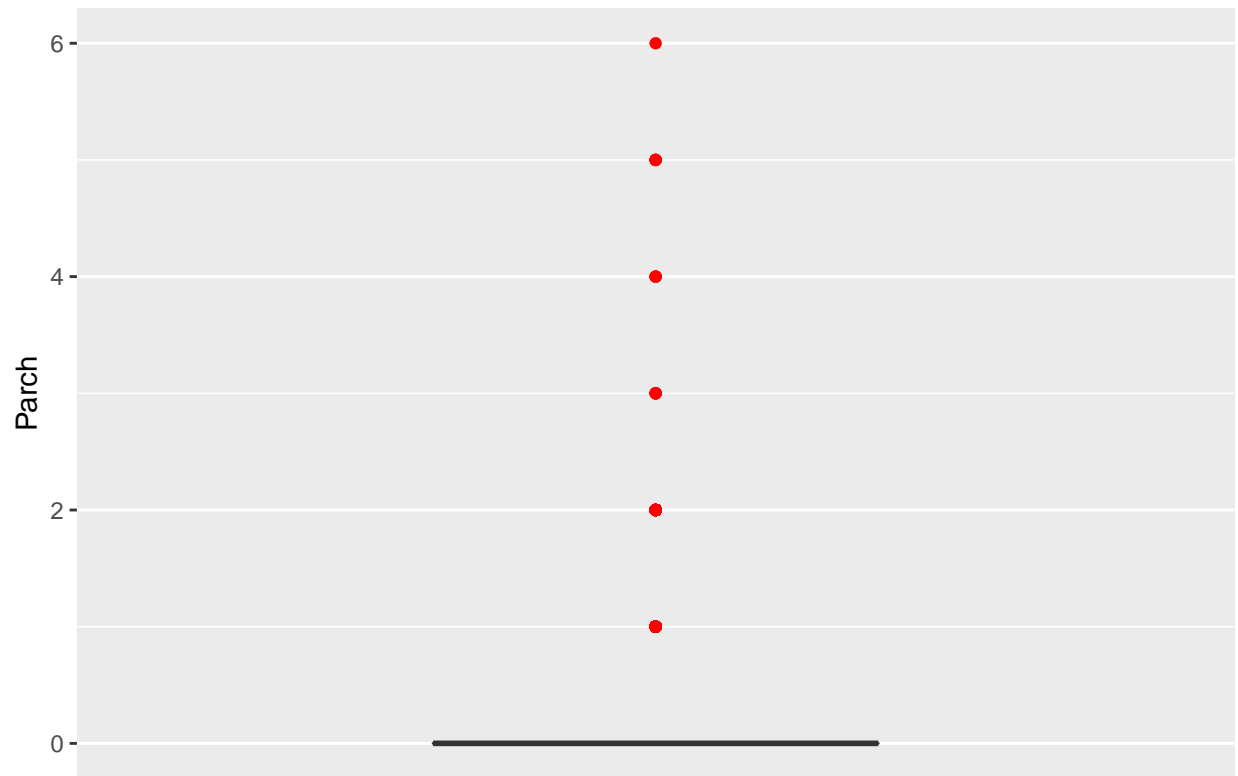
```
# Boxplot para la variable Parch
```

```
summary(ttc$Parch)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000  0.0000  0.0000  0.3825  0.0000  6.0000
```

```
ggplot(data = ttc, aes(y = Parch)) +
  geom_boxplot(outlier.colour = "red") +
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
  ggtitle("Passenger parents/children Boxplot") +
  theme(plot.title = element_text(size = 20, hjust = 0.5))
```

Passenger parents/children Boxplot



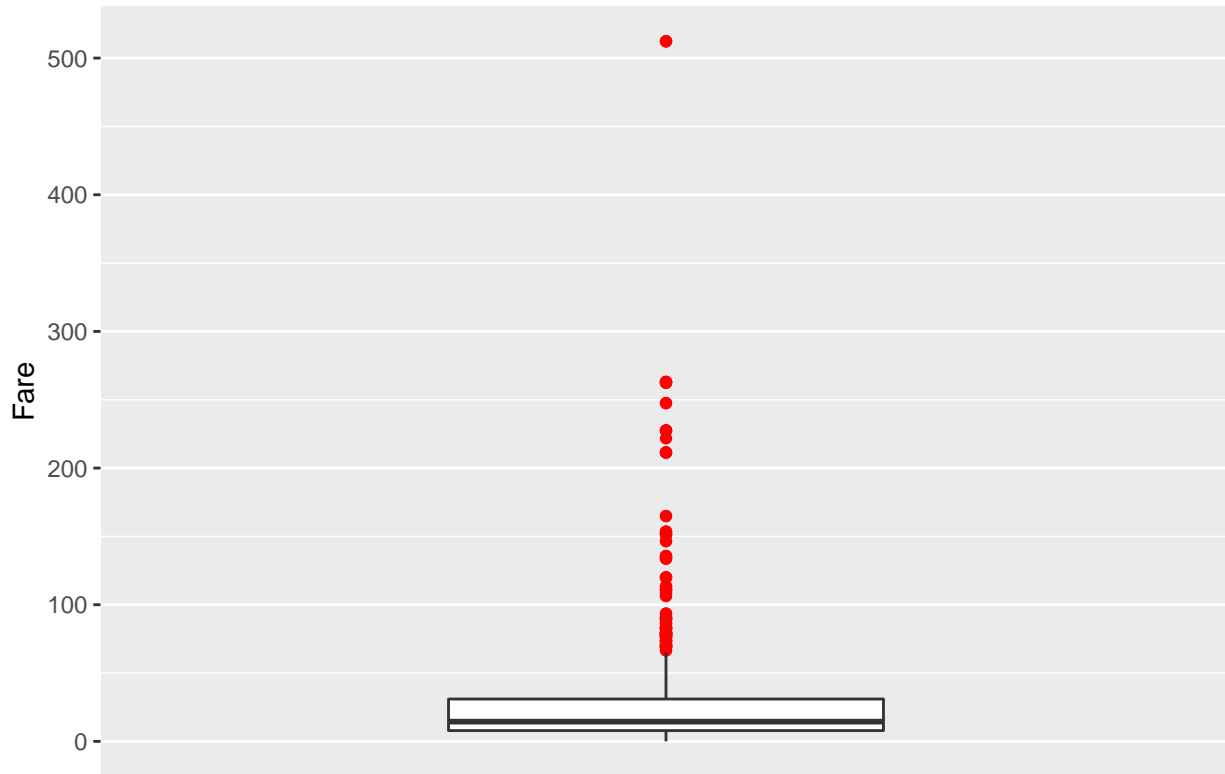
```
# Boxplot para la variable Fare
```

```
summary(ttc$Fare)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.000   7.896   14.454   32.097   31.000  512.329
```

```
ggplot(data = ttc, aes(y = Parch)) +
  geom_boxplot(outlier.colour = "red") +
  scale_x_discrete(guide = guide_axis(n.dodge = 2)) +
  ggtitle("Passenger paid Fare Boxplot") +
  theme(plot.title = element_text(size = 20, hjust = 0.5))
```

Passenger paid Fare Boxplot



Una vez representadas las diferentes variables numéricas, procedemos a extraer las conclusiones oportunas de cada una de ellas. Cuando existen valores atípicos en los datos, pueden darse debido a varias opciones, puede ser porque son valores tomados con una unidad diferente que haga que algunos casos difieran de manera atípica de otros, puede que esos valores representen un valor faltante o nulo o puede que dichos valores formen parte de la muestra y por lo tanto sean valores reales y que hay que tener en cuenta a la hora de realizar los diferentes análisis. También puede darse el caso que sean valores que por el contexto se denote que no han sido tomados correctamente (como por ejemplo una edad de 150 años).

En este caso, nos encontramos con que tenemos valores atípicos en todas las variables numéricas, aunque sí que es cierto que en algunas de ellas dichos valores se encuentran más aislados, como es el caso de la variable **Fare**, que indica la tarifa del pasajero/a.

- En el caso de la variable **Age**, vemos que la mediana se encuentra aproximadamente entre los 25 años, pero que existen valores atípicos a partir de 60 y que hay casos de pasajeros hasta con 80 años. Sí que es verdad que se trata de un valor atípico con respecto a la mayoría de personas que se encuentran en el barco, pero no es un valor imposible de encontrar, por lo que se considera que forma parte del conjunto de datos y que se tiene que tener en cuenta a la hora de realizar el análisis.
- En el caso de la variable **Sibing/spouses** el valor más destacado es el caso de 8. Sí que es verdad que se trata de un valor poco casual, pero puede darse el caso de que una persona tenga 7 hermanos/as y una esposa, o múltiples combinaciones, es decir, a simple vista, no parece ser un valor irreal, por lo que se considera que también se debe tener en cuenta para el análisis.
- En el caso de la variable **Parents/children** pasa un poco como con la variable anterior, el valor máximo es 6 pero no se trata de un valor imposible o improbable, y más por la época en la que se basan los datos, en la que tener 5 o 6 hijos era algo común, por lo que no se considera oportuno realizar ningún cambio en dichos valores.
- Por último, la variable **Fare** resulta ser la variable que contiene los valores atípicos que más se alejan de

la desviación estándar de la misma, pues ya de por sí un valor por encima de los 250 resulta ser bastante atípico (según los datos), con lo que en el caso de estar por encima de 500, sitúa dicho valor demasiado alejado de los demás. Dado el significado de la variable y el contexto, puede tratarse perfectamente de un valor real, ya que en los cruceros existen pasajes muy lujosos que tienen un precio muy por encima de un pasaje estándar. No obstante, si que es cierto, que aunque pueda tratarse de valores reales, al estar tan extremadamente alejado de la desviación estándar de la población, puede hacer que los diferentes análisis que se apliquen estén sesgados por dichos valores.

A continuación se procede a estudiar cuantos casos de la variable **Fare** se encuentran por encima de 500 y a realizar una comparación del resultado obtenido por una medida de dispersión robusta a la presencia de valores atípicos, la mediana, con una no robusta a ellos, la media.

```
outlier_cases = nrow(ttc[ttc$Fare > 500,])
mean_Fare = mean(ttc$Fare)
median_Fare = median(ttc$Fare)
```

```
outlier_cases
```

```
## [1] 3
```

```
mean_Fare
```

```
## [1] 32.09668
```

```
median_Fare
```

```
## [1] 14.4542
```

Como se puede observar, el número total de casos atípicos son 3 en todo el dataset.

Por otro lado, el valor obtenido por la media es de 32.0966809, mientras que por la mediana es de 14.4542. Si comparamos ambos resultados, podemos ver que el valor obtenido por la media es aproximadamente el doble que el obtenido por la mediana, lo que indica que los valores atípicos están sesgando dicha medida de dispersión, pero dicho sesgo no se debe a los 3 casos que se encuentran por encima de 500, si que es cierto que influirán, pero el grueso del sesgo se debe que existen muchos casos por encima de la mediana.

Al tratarse de tan pocos casos de los que se encuentran exageradamente desviados (3) y al parecer por el contexto que pueden tratarse de valores reales, se considera que deben ser utilizados para los distintos análisis o métodos estadísticos que se realicen, pero que se deberá tener en cuenta su presencia para aplicar análisis que sean robustos a la presencia de valores atípicos. Pues realizar una imputación de todos los valores que son realmente atípicos o eliminarlos, conllevaría una gran pérdida de información que no es necesaria.

4. Análisis de los datos.

A continuación, procederemos a realizar una visualización de las diferentes columnas o variables que forman el dataset, para ver como se distribuyen las mismas.

Comenzaremos por aquellas variables categóricas o cualitativas, estas son:

```
factors = unlist(lapply(ttc, is.factor))
which(factors, arr.ind = TRUE)
```

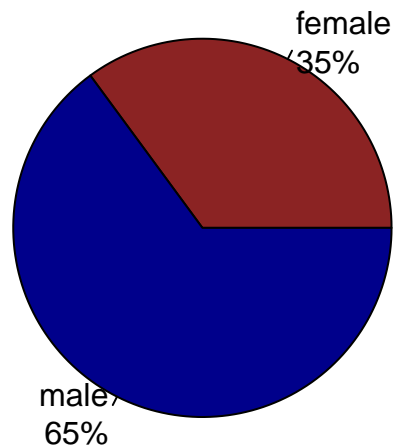
```
## Survived  Pclass      Sex Embarked
##          2         3         5       11
```

Procedemos a continuación a representar cada una de ellas:

```
mytableSex <- table(ttc$Sex)
pctSex <- round(mytableSex/sum(mytableSex)*100)
lblsSex <- paste(names(mytableSex), "\n", pctSex, sep="")
```

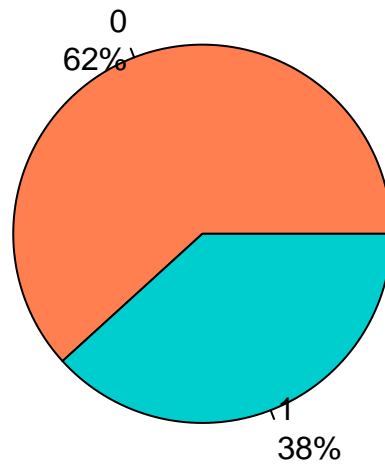
```
lblsSex <- paste (lblsSex, '%', sep="")
pie(mytableSex, labels = lblsSex,
    main="Distribución de la variable Sex\n", col=c("brown4","darkblue"))
```

Distribución de la variable Sex



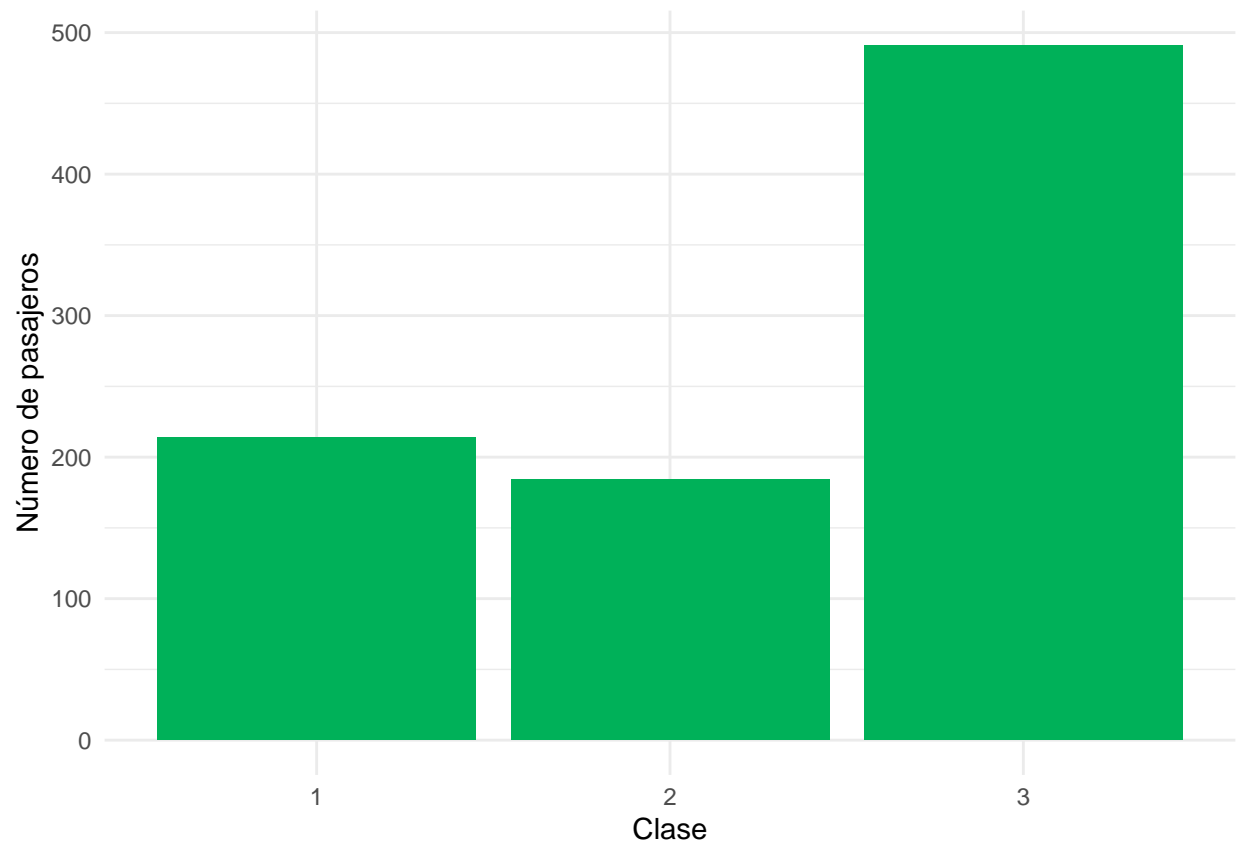
```
mytableSurvived <- table(ttc$Survived)
pctSurvived <- round(mytableSurvived/sum(mytableSurvived)*100)
lblsSurvived<- paste(names(mytableSurvived), "\n", pctSurvived, sep="")
lblsSurvived <- paste (lblsSurvived, '%', sep="")
pie(mytableSurvived, labels = lblsSurvived,
    main="Distribución de la variable Survived\n", col = c("coral","cyan3"))
```

Distribución de la variable Survived



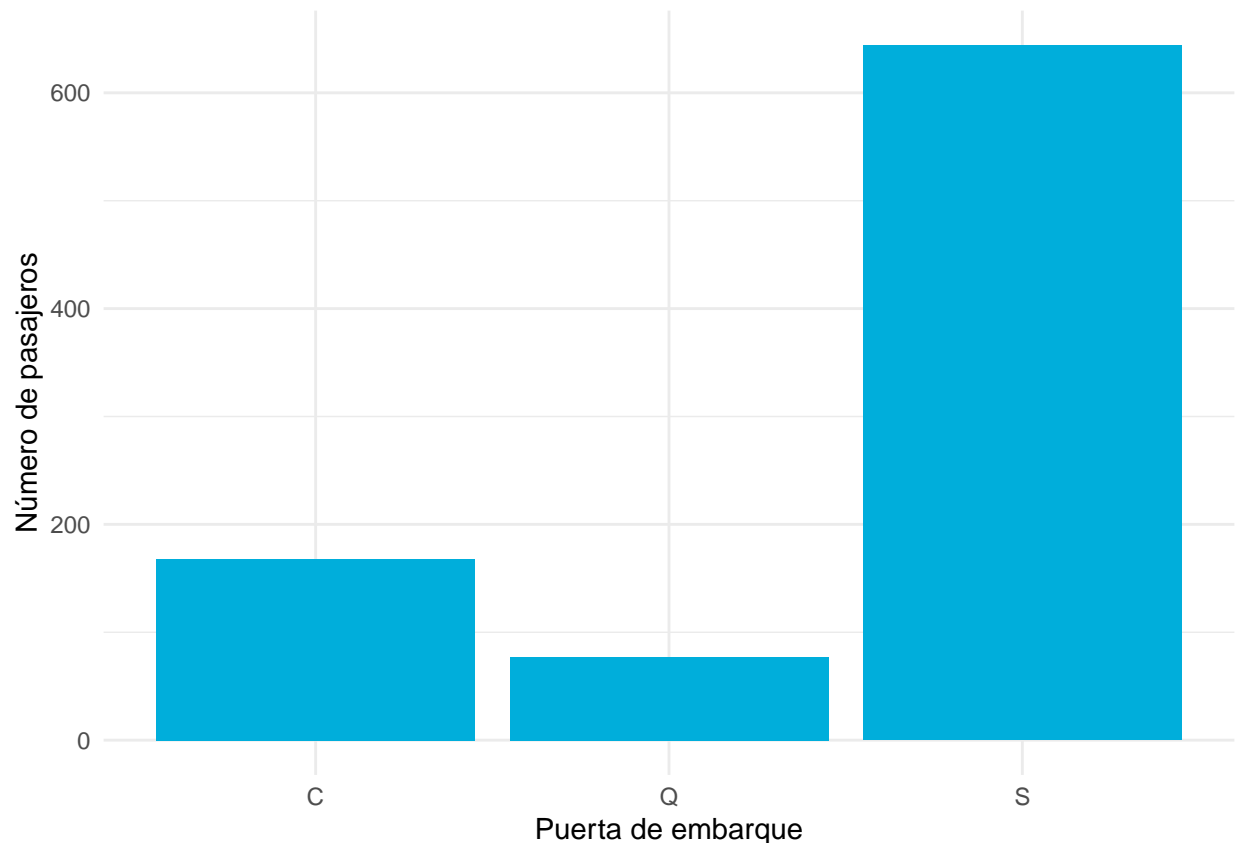
```
tablePclass<-table(ttc$Pclass)
dfPclass<-data.frame(tablePclass)

p<-ggplot(data=dfPclass, aes(x=Var1, y=Freq)) +
  geom_bar(stat="identity", fill="#00b159")+
  theme_minimal()+
  xlab("Clase")+
  ylab("Número de pasajeros")
p
```

```
tableEmb<-table(ttc$Embarked)
dfEmb<-data.frame(tableEmb)

p<-ggplot(data=dfEmb, aes(x=Var1, y=Freq)) +
  geom_bar(stat="identity", fill="#00aedb")+
  theme_minimal()+
  xlab("Puerta de embarque")+
  ylab("Número de pasajeros")
p
```



De las gráficas anteriores podemos obtener las siguientes conclusiones:

- Hay una mayor proporción de hombres que de mujeres a bordo.
- La mayoría de los personas que iban a bordo no sobrevivieron.
- La mayoría de las personas viajaron en tercera clase, y el número de personas que viajaban en segunda y en primera clase era muy similar.
- La gran mayoría de pasajeros entraron por la puerta de embarque “S”

Una vez representadas las variables categóricas, procedemos a representar las variables continuas del dataset, las cuales son:

```
numerics = unlist(lapply(ttc, is.numeric))
which(numerics, arr.ind = TRUE)
```

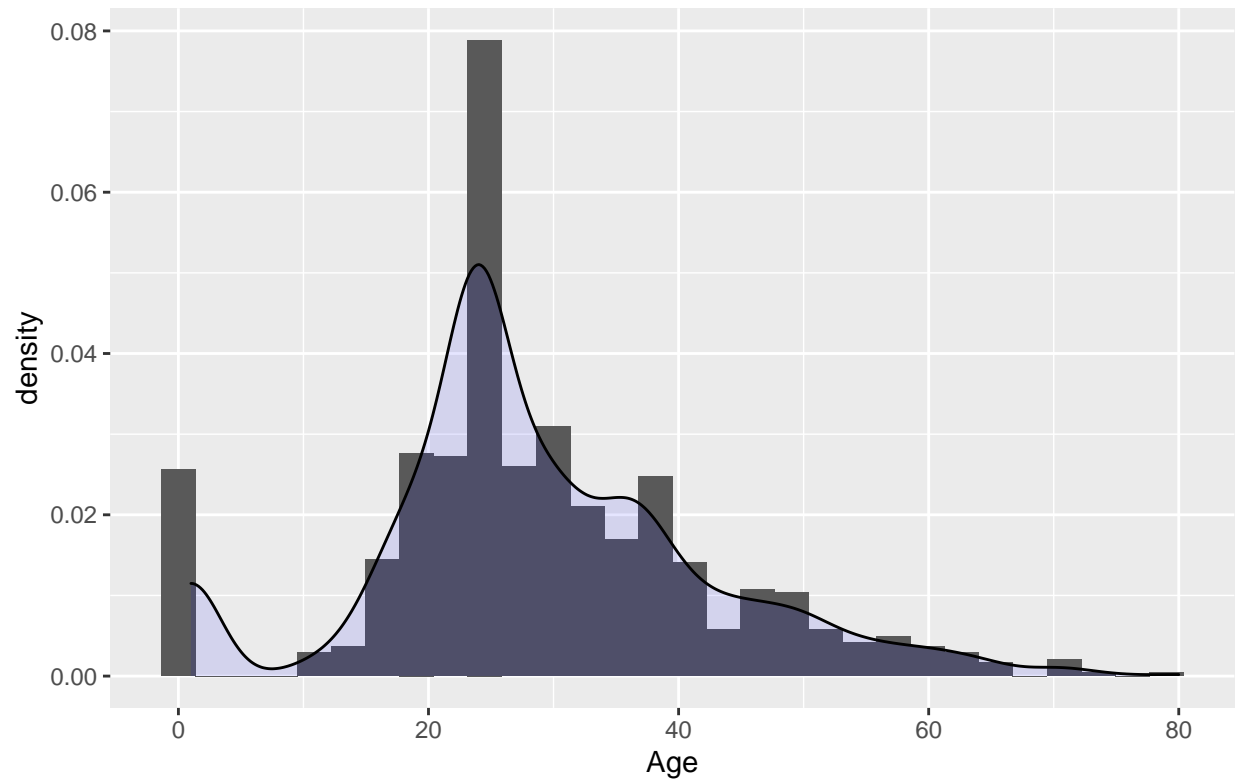
```
## PassengerId      Age      SibSp      Parch      Fare
##           1         6         7         8         10
```

De todas las variables que han resultado ser numéricas, representaremos todas menos la variable **PassengerId** que indica únicamente el identificador de cada pasajero o pasajera.

```
# Histograma para la variable Age
ggplot(ttc, aes(x = Age)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.1, fill = "blue") +
  ggtitle("Passengers Age Density Histogram") +
  theme(plot.title = element_text(size = 20, hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

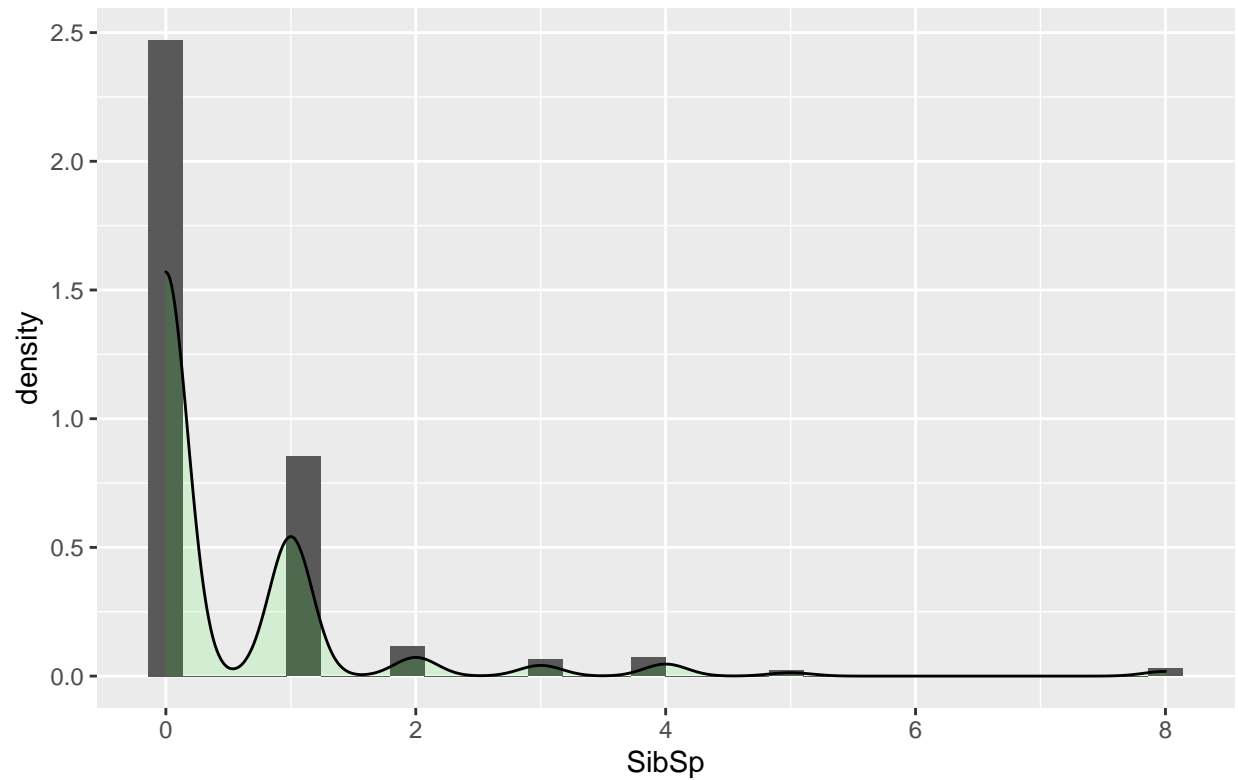
Passengers Age Density Histogram



```
# Histograma para la variable SibSp
ggplot(ttc, aes(x = SibSp)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.1, fill = "green") +
  ggtitle("Passengers sibings/spouses Density Histogram") +
  theme(plot.title = element_text(size = 20, hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

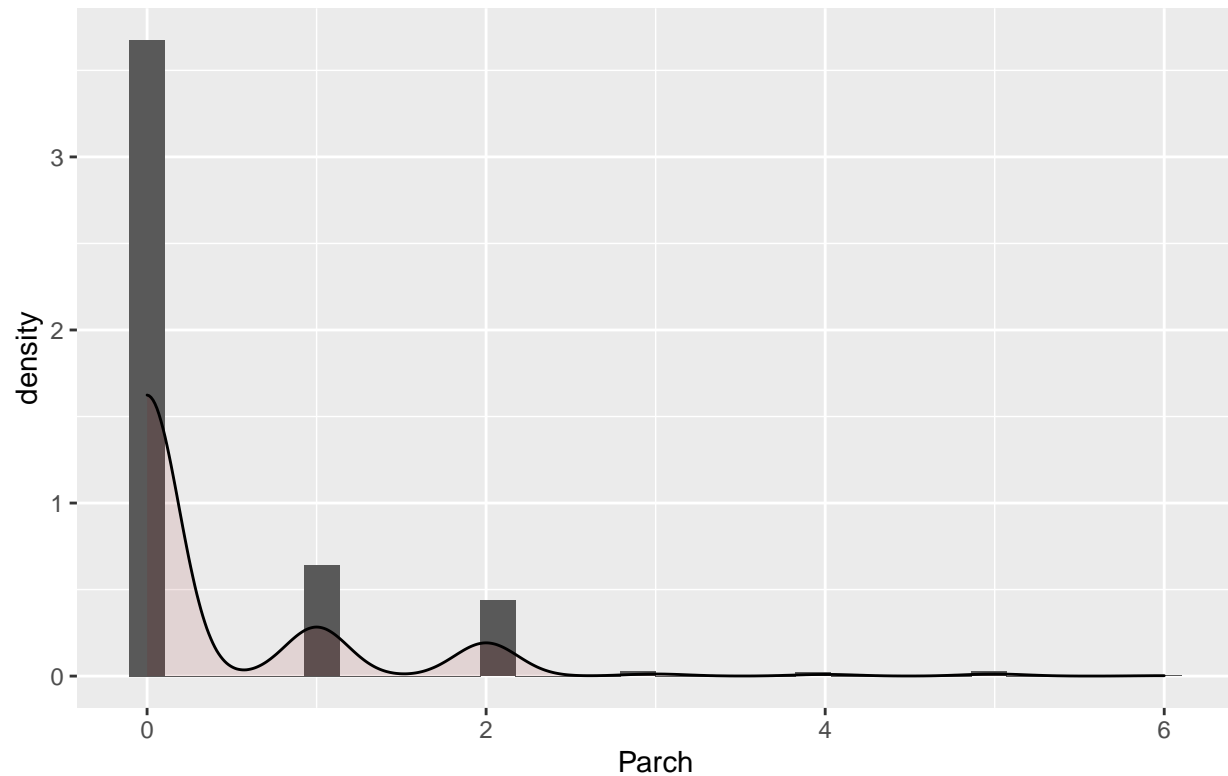
Passengers siblings/spouses Density Histogram



```
# Histograma para la variable Parch
ggplot(ttc, aes(x = Parch)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.1, fill = "darkred") +
  ggtitle("Passengers parents/children Density Histogram") +
  theme(plot.title = element_text(size = 20, hjust = 0.5))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

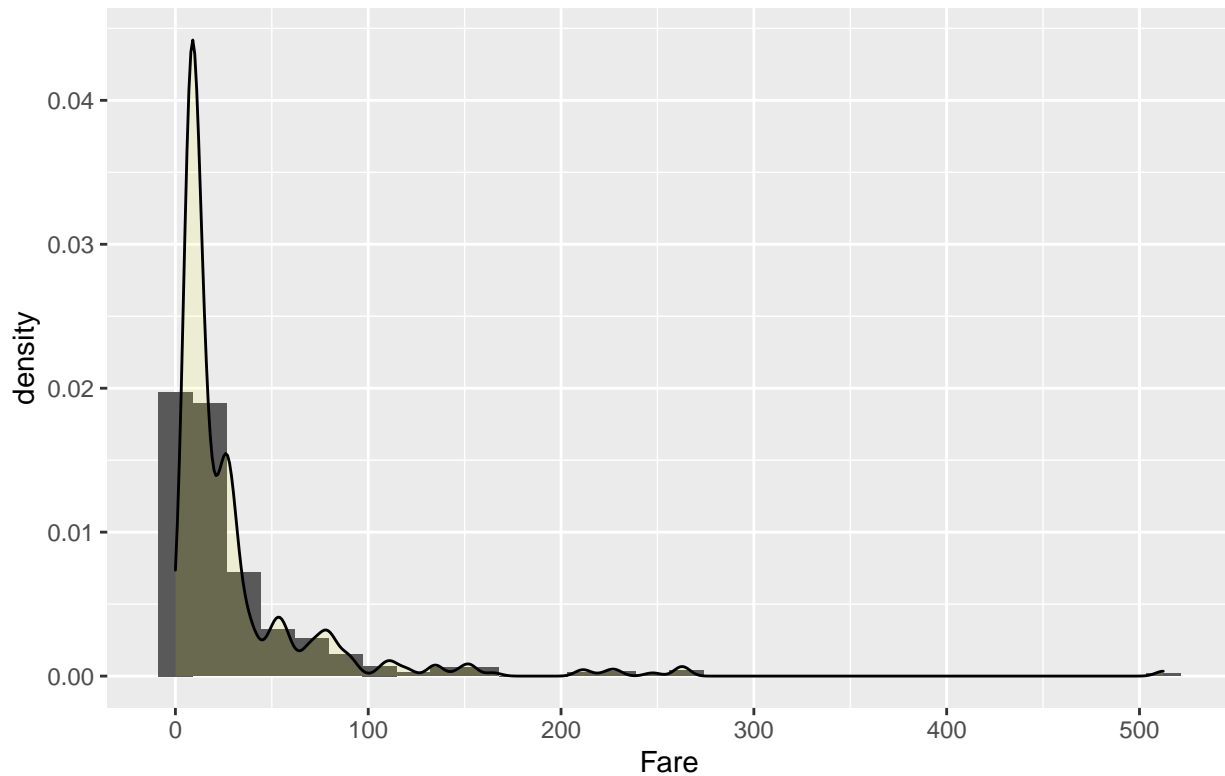
Passengers parents/children Density Histogram



```
# Histograma para la variable Fare
ggplot(ttc, aes(x = Fare)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.1, fill = "yellow") +
  ggtitle("Passengers paid Fare Density Histogram") +
  theme(plot.title = element_text(size = 20, hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Passengers paid Fare Density Histogram



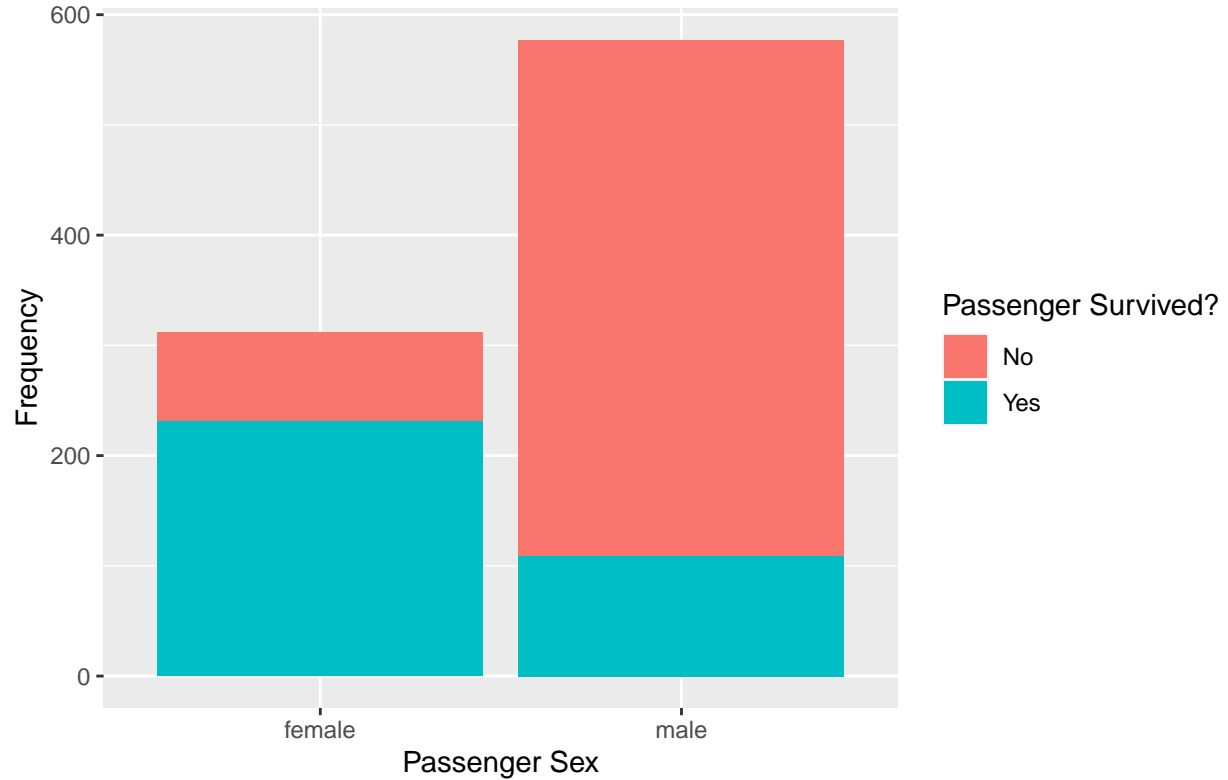
De las gráficas anteriores podemos obtener las siguientes conclusiones:

- La variable **Age** se distribuye aproximadamente de una forma normal.
- Las demás variables numéricas presentan una distribución unimodal sesgada hacia la izquierda.

Para una visualización general de los datos, podemos representar gráficamente los supervivientes agrupados por diversas variables.

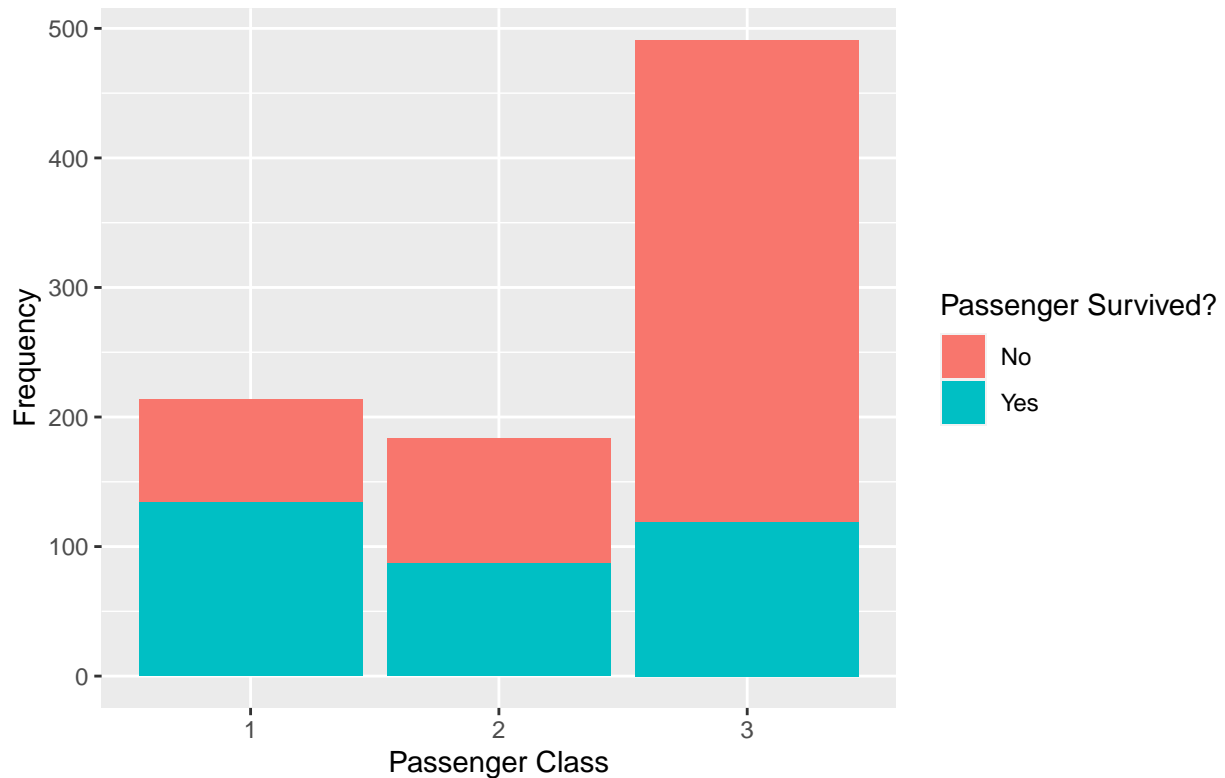
```
# Por ejemplo, la frecuencia de supervivientes por Sexo
ggplot(as.data.frame(table(ttc$Survived, ttc$Sex)), aes(Var2, Freq, fill=Var1)) +
  geom_bar(stat="identity") +
  scale_fill_discrete(name = "Passenger Survived?", labels = c("No", "Yes")) +
  ggtitle("Passenger Survival Frequency by Sex") +
  theme(plot.title = element_text(size = 20, hjust = 0.5)) +
  xlab("Passenger Sex") + ylab("Frequency")
```

Passenger Survival Frequency by Sex



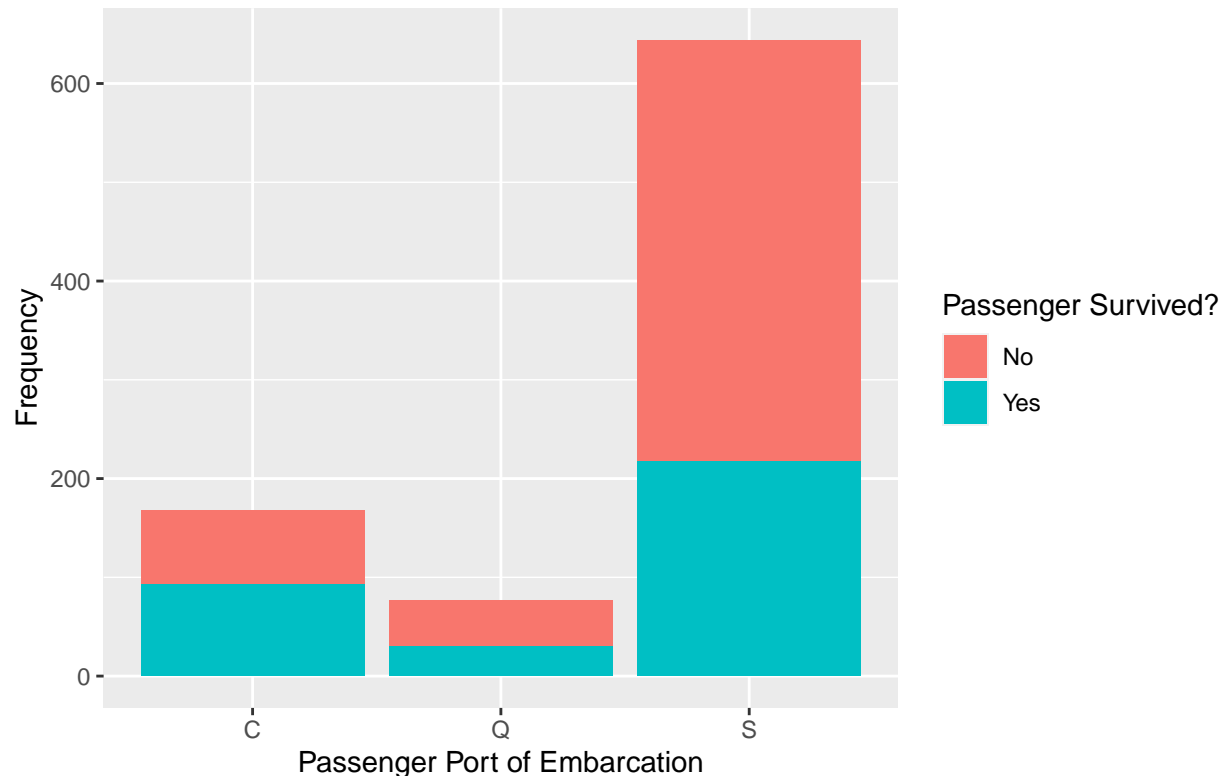
```
# O la frecuencia de supervivientes por Clase
ggplot(as.data.frame(table(ttc$Survived, ttc$Pclass)), aes(Var2, Freq, fill=Var1)) +
  geom_bar(stat="identity") +
  scale_fill_discrete(name = "Passenger Survived?", labels = c("No", "Yes")) +
  ggtitle("Passenger Survival Frequency by Class") +
  theme(plot.title = element_text(size = 20, hjust = 0.5)) +
  xlab("Passenger Class") + ylab("Frequency")
```

Passenger Survival Frequency by Class



```
# O incluso la frecuencia de supervivientes por puerto de embarque
ggplot(as.data.frame(table(ttc$Survived, ttc$Embarked)), aes(Var2, Freq, fill=Var1)) +
  geom_bar(stat="identity") +
  scale_fill_discrete(name = "Passenger Survived?", labels = c("No", "Yes")) +
  ggtitle("Passenger Survival by Port of Embarcation") +
  theme(plot.title = element_text(size = 20, hjust = 0.5)) +
  xlab("Passenger Port of Embarcation") + ylab("Frequency")
```

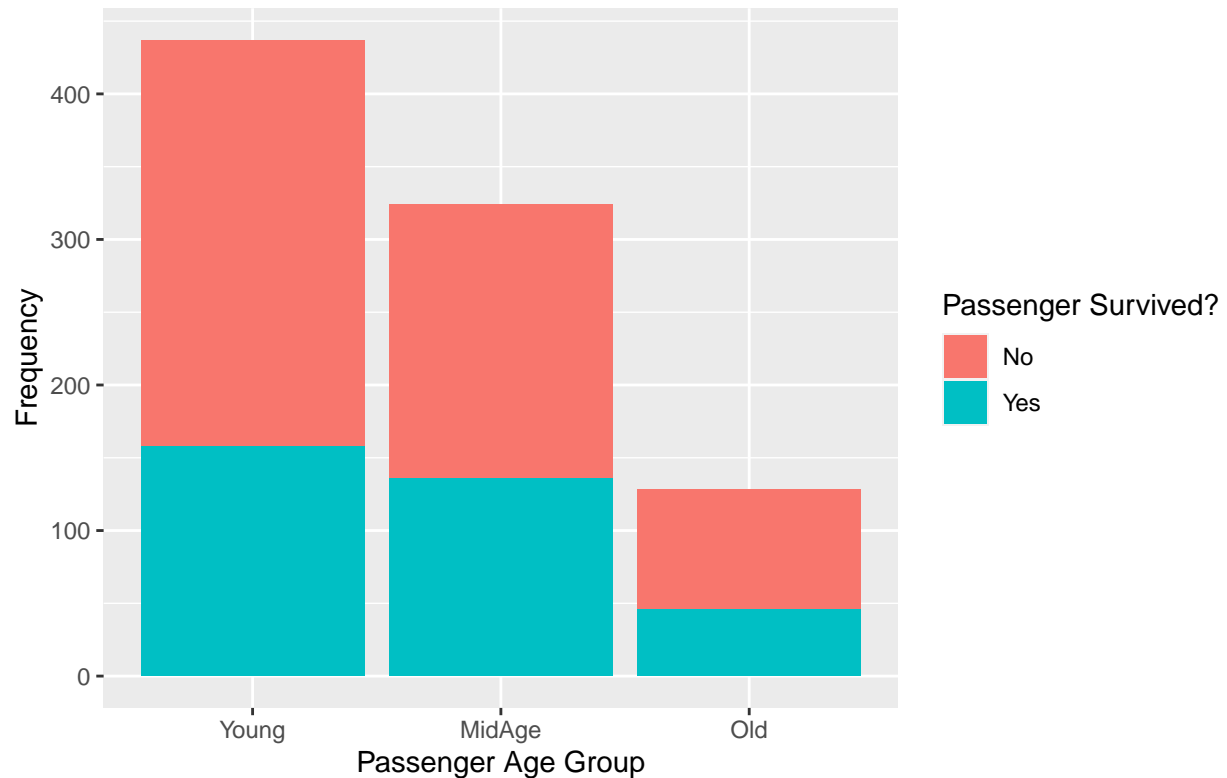

Passenger Survival by Port of Embarcation



*# Tambien por el grupo de Edad, aunque previamente debemos discretizar la variable Age
ya que es numerica continua.*

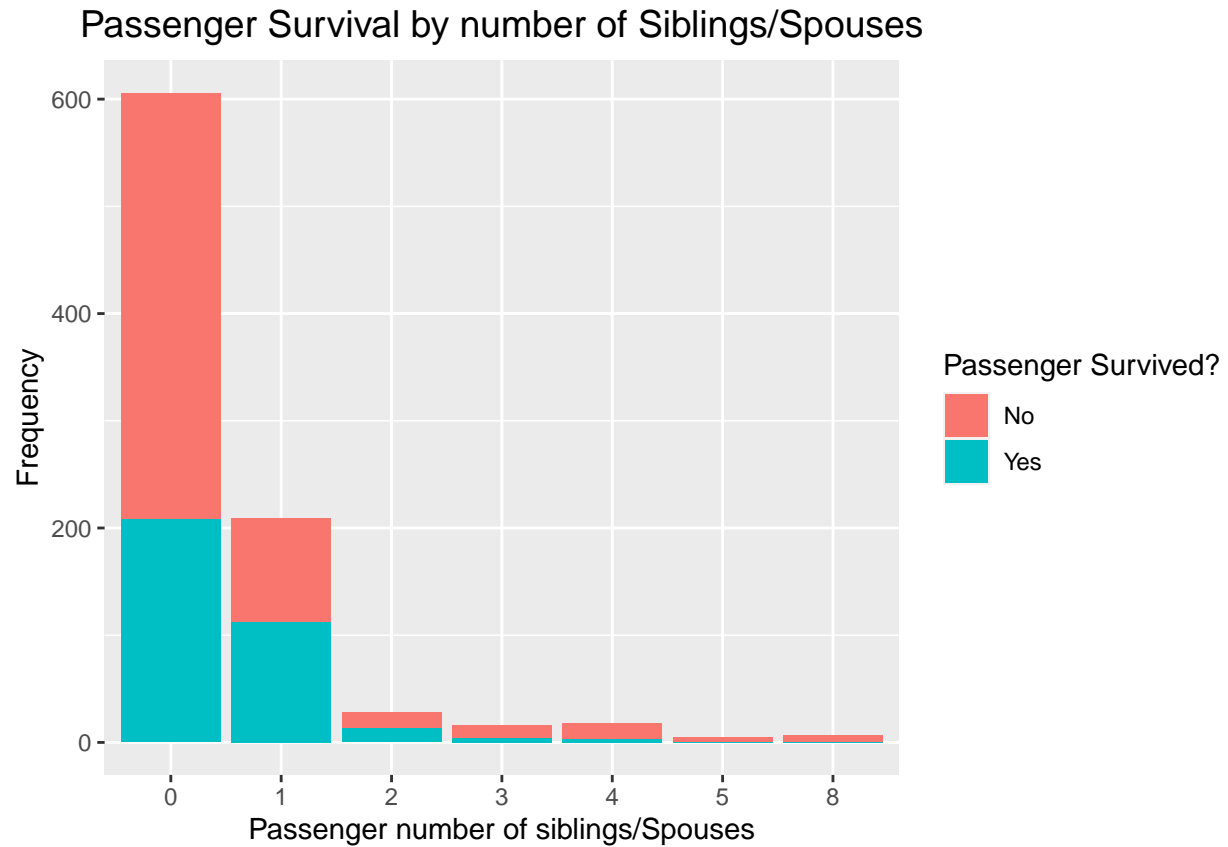
```
ttc$AgeD <- discretize(ttc$Age,  
                        method = "cluster", breaks = 3, labels=c("Young", "MidAge", "Old"))  
  
ggplot(as.data.frame(table(ttc$Survived, ttc$AgeD)), aes(Var2, Freq, fill=Var1)) +  
  geom_bar(stat="identity") +  
  scale_fill_discrete(name = "Passenger Survived?", labels = c("No", "Yes")) +  
  ggtitle("Passenger Survival by Age Group") +  
  theme(plot.title = element_text(size = 20, hjust = 0.5)) +  
  xlab("Passenger Age Group") + ylab("Frequency")
```

Passenger Survival by Age Group



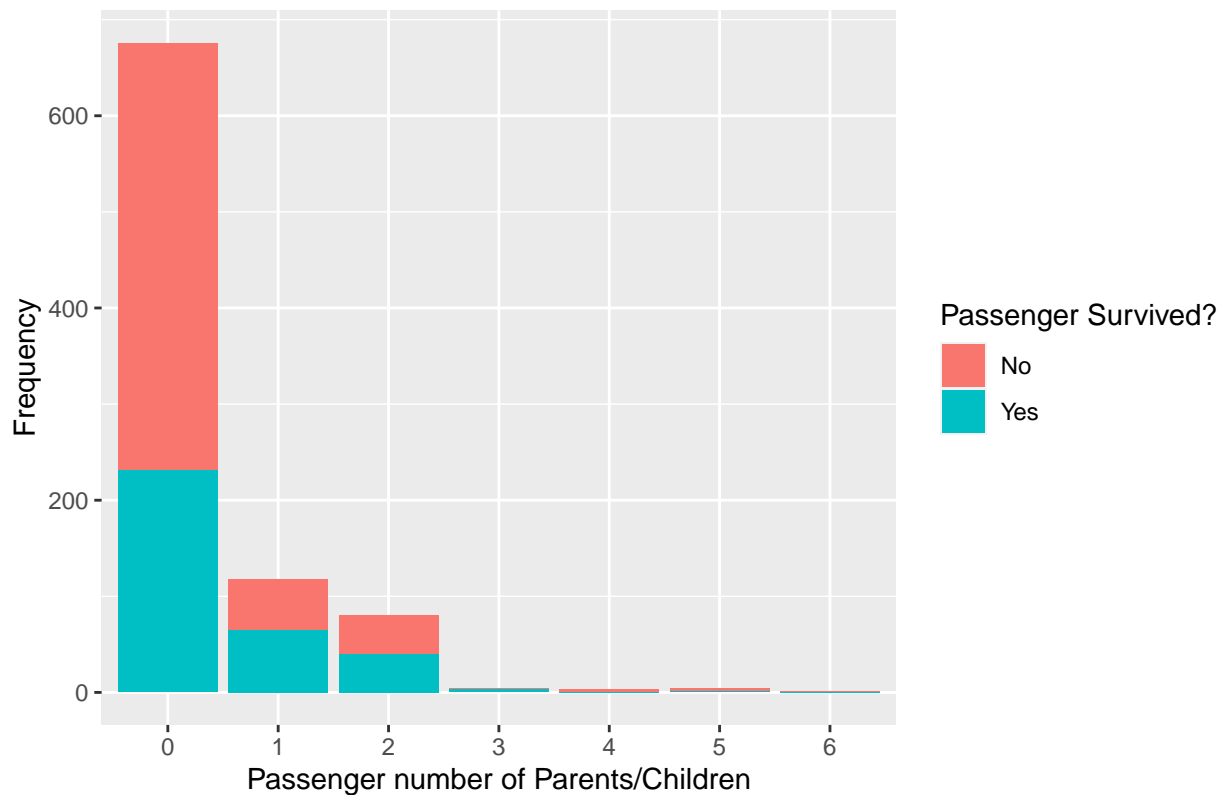
*# Otra grafica interesante puede ser aquella que muestre la frecuencia de supervivencia
dependiendo de si el pasajero tenia familiares con el en el barco o viajaban solos*

```
ggplot(as.data.frame(table(ttc$Survived, ttc$SibSp)), aes(Var2, Freq, fill=Var1)) +  
  geom_bar(stat="identity") +  
  scale_fill_discrete(name = "Passenger Survived?", labels = c("No", "Yes")) +  
  ggtitle("Passenger Survival by number of Siblings/Spouses") +  
  theme(plot.title = element_text(size = 14, hjust = 0.5)) +  
  xlab("Passenger number of siblings/Spouses") + ylab("Frequency")
```



```
ggplot(as.data.frame(table(ttc$Survived, ttc$Parch)), aes(Var2, Freq, fill=Var1)) +  
  geom_bar(stat="identity") +  
  scale_fill_discrete(name = "Passenger Survived?", labels = c("No", "Yes")) +  
  ggtitle("Passenger Survival by number of Parents/Children") +  
  theme(plot.title = element_text(size = 14, hjust = 0.5)) +  
  xlab("Passenger number of Parents/Children") + ylab("Frequency")
```

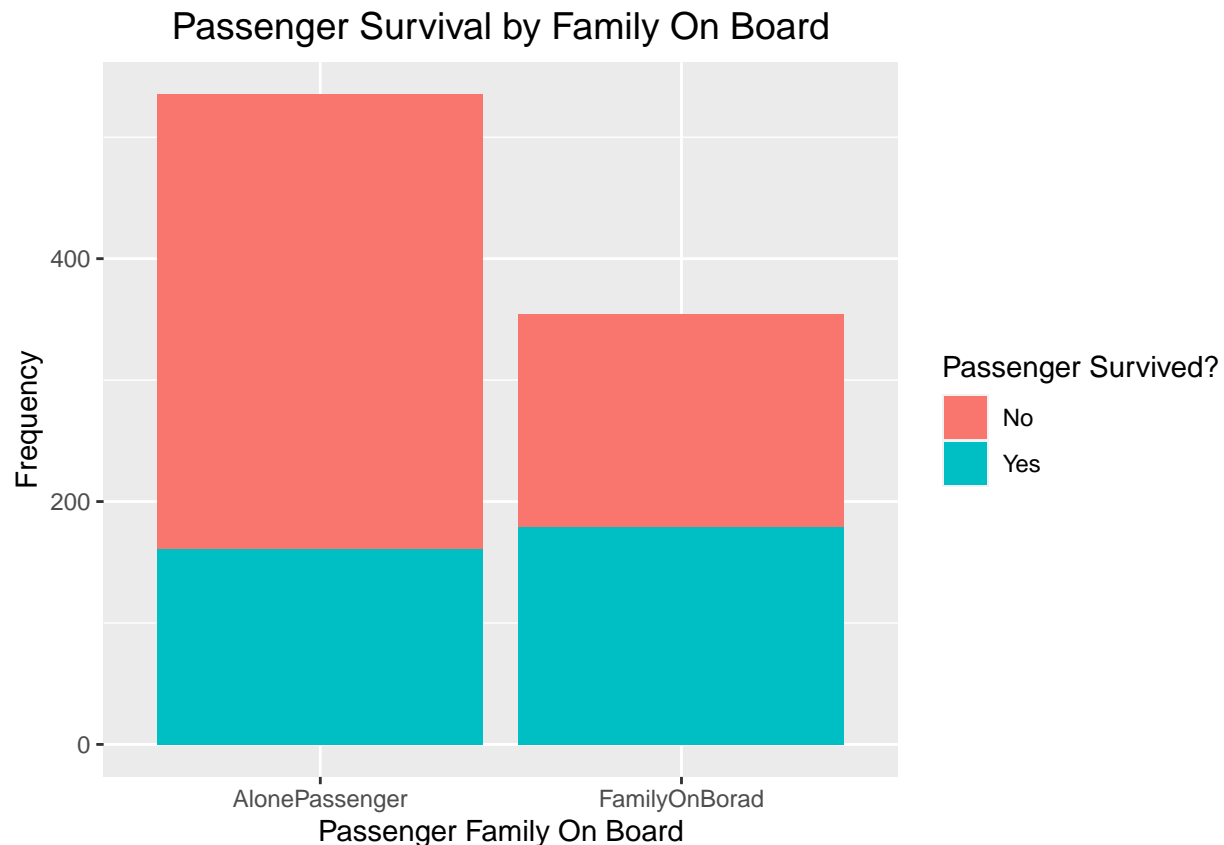
Passenger Survival by number of Parents/Children



```
# 0 en general, si el pasajero tenia familia a bordo
ttc$PassengerFamily <- ifelse(ttc$SibSp != 0 | ttc$Parch != 0, 'FamilyOnBorad', "AlonePassenger")
table(ttc$Survived, ttc$PassengerFamily)
```

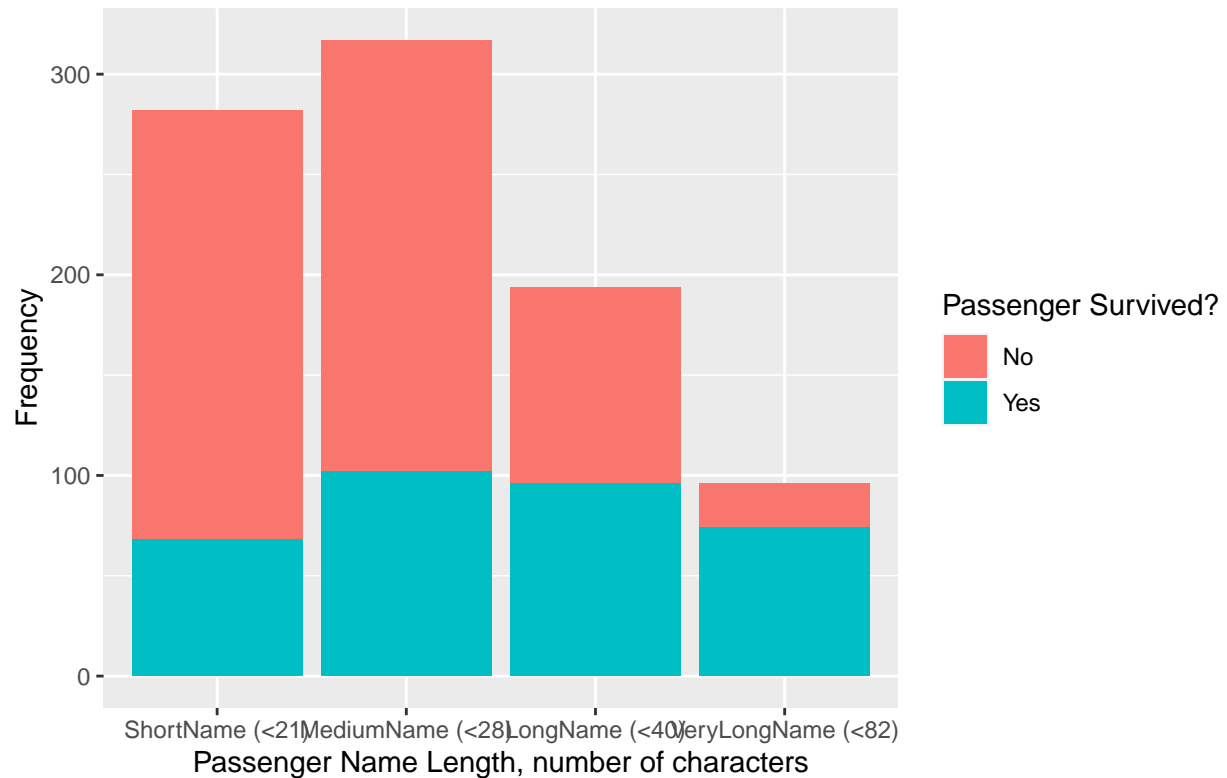
```
##
##      AlonePassenger FamilyOnBorad
## 0              374             175
## 1              161             179
```

```
ggplot(as.data.frame(table(ttc$Survived, ttc$PassengerFamily)), aes(Var2, Freq, fill=Var1)) +
  geom_bar(stat="identity") +
  scale_fill_discrete(name = "Passenger Survived?", labels = c("No", "Yes")) +
  ggtitle("Passenger Survival by Family On Board") +
  theme(plot.title = element_text(size = 14, hjust = 0.5)) +
  xlab("Passenger Family On Board") + ylab("Frequency")
```



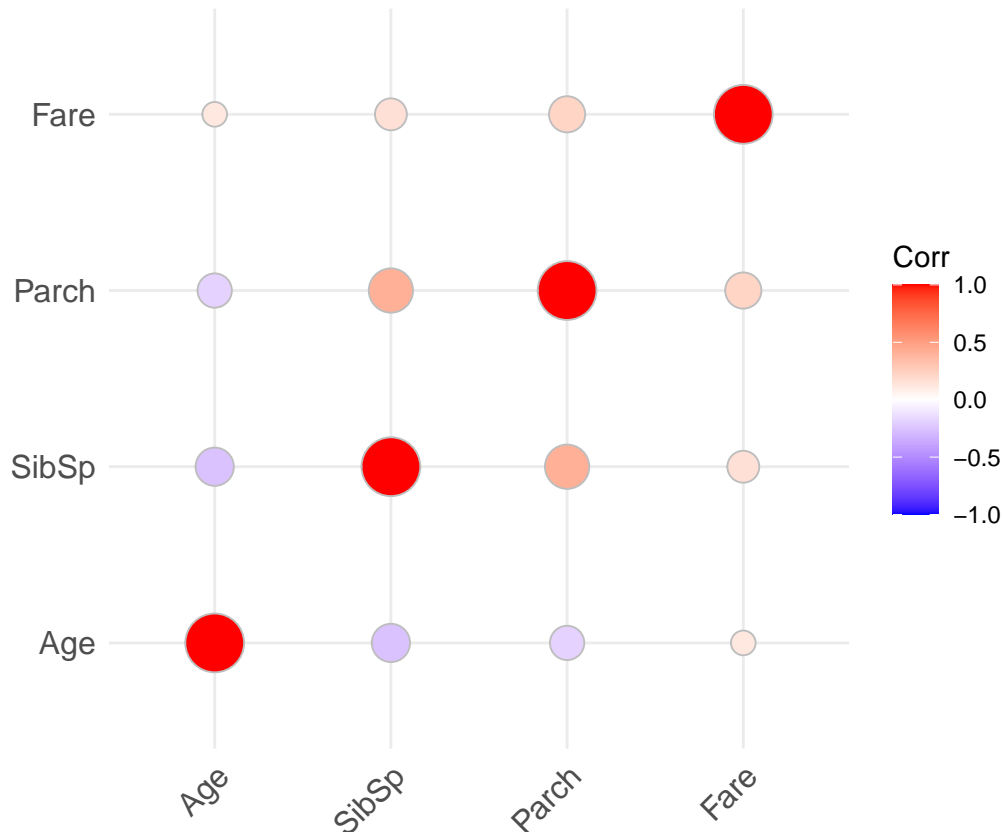
```
# Por ultimo una relacion interesante, es la frecuencia de supervivencia asociada
# a la longitud del nombre del pasajero, bajo una premisa inicial de que, cuanto
# mas largo fuera el nombre, el pasajero podria tener una clase social mas elevada
ttc$NameLength <- vector("numeric", nrow(ttc))
for (i in 1:nrow(ttc)) {
  ttc$NameLength[i] <- nchar(as.character(ttc$Name)[i])
}
ttc$NameLengthD <- discretize(ttc$NameLength,
  method = "cluster", breaks = 4, labels=c("ShortName (<21)",
                                             "MediumName (<28)",
                                             "LongName (<40)",
                                             "VeryLongName (<82)"))
ggplot(as.data.frame(table(ttc$Survived, ttc$NameLengthD)), aes(Var2, Freq, fill=Var1)) +
  geom_bar(stat="identity") +
  scale_fill_discrete(name = "Passenger Survived?", labels = c("No", "Yes")) +
  ggtitle("Passenger Survival by Name Length") +
  theme(plot.title = element_text(size = 20, hjust = 0.5)) +
  xlab("Passenger Name Length, number of characters") + ylab("Frequency")
```

Passenger Survival by Name Length



Por ultimo en el proceso de exploracion de los datos, se puede obtener una matriz de correlacion sobre las variables numericas del dataset:

```
ttc_num <- subset(ttc, select=c(Age, SibSp, Parch, Fare))
ttccorr <- cor(ttc_num)
ggcorrplot(ttccorr, method = "circle")
```



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

Se plantean 2 tipos de estrategias de analisis de los datos:

1. Analisis estadistico inferencial centrado en 2 contrastes de Hipotesis.
2. Modelizacion predictiva aplicando 3 modelos de clasificacion (regresion logistica, Random Forest y SupportVector Machine - SVM).

A continuacion se desarrollan cada uno de ellos:

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

Los chequeos de normalidad y homogeneidad de varianza son importantes para aplicar las diferentes herramientas y Tests de Hipótesis. En general usaremos pruebas Z-Test que requieren normalidad de las variables y el cálculo del estadístico usa una fórmula u otra dependiendo de si las varianzas son homogéneas o no.

Para el análisis de contrastes de hipótesis sobre la media usamos una métrica específica: el ratio de supervivencia. Esto es, si tomamos una muestra aleatoria de pasajeros bajo una condición fija (mismo sexo, o misma clase) y obtenemos la media de la variable Survived, dicha media nos dará el ratio de supervivencia sobre esa sub muestra.

Por ejemplo, si tomamos una muestra de 100 pasajeros de primera clase y obtenemos la media de la variable Survived:

```
mean(as.numeric(as.character(ttc[sample(which(ttc$Pclass == 1),100),]$Survived)))
```

```
## [1] 0.66
```

Este valor nos dice la media del ratio de supervivencia de esa muestra en particular.

El teorema del límite central establece que, tomando un número suficiente de estas submuestras, esta media del ratio de supervivencia se distribuye siguiendo una normal. Vamos a aplicar contrastes de hipótesis atendiendo a grupos por Clase (Primera y Tercera) y Sexo.

4.2.1 Comprobación de la normalidad y la homogeneidad de la varianza en muestras de supervivencia por la clase del pasajero

Para comprobar la normalidad y la homogeneidad de la varianza, se genera un array de medias sobre 100 submuestras aleatorias de 100 pasajeros de cada una de las clases, y se comprobará la normalidad y la homogeneidad sobre dicho array, de manera que si sobre ese conjunto aleatorio obtenido se cumple, se cumplirá para todo el conjunto, según el teorema del límite central.

```
# Generamos un array de medias sobre 100 submuestras aleatorias de 100
# pasajeros de primera clase cada submuestra.
iter <- 100
vars <- 1
First_class_SampleMeans <- matrix(ncol=vars, nrow=iter)
for(i in 1:iter){
  set.seed(i*16)
  First_class_SampleMeans[i,] <- mean(as.numeric(as.character
                                         (ttc[sample(which
                                                    (ttc$Pclass == 1)
                                                    ,100),]$Survived)))
}
mean(First_class_SampleMeans)
```

```
## [1] 0.6331
```

```
First_class_SampleMeans <- data.frame(First_class_SampleMeans)

# Hacemos lo mismo pero para 100 submuestras aleatorias de 100 pasajeros de tercera clase
# cada submuestra.
Third_class_SampleMeans <- matrix(ncol=vars, nrow=iter)
for(i in 1:iter){
  set.seed(i*16)
  Third_class_SampleMeans[i,] <- mean(as.numeric(as.character
                                         (ttc[sample(which
                                                    (ttc$Pclass == 3)
                                                    ,100),]$Survived)))
}
mean(Third_class_SampleMeans)
```

```
## [1] 0.2411
```

```
Third_class_SampleMeans <- data.frame(Third_class_SampleMeans)

# Revisamos la distribucion de estas variables a nivel grafico de densidad:
SF_FirstClass_Density <- ggplot(First_class_SampleMeans, aes(x = First_class_SampleMeans)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.5, fill = "green") +
  ggtitle("Survival rate Frequency in First Class") +
  theme(plot.title = element_text(size = 10, hjust = 0.5))

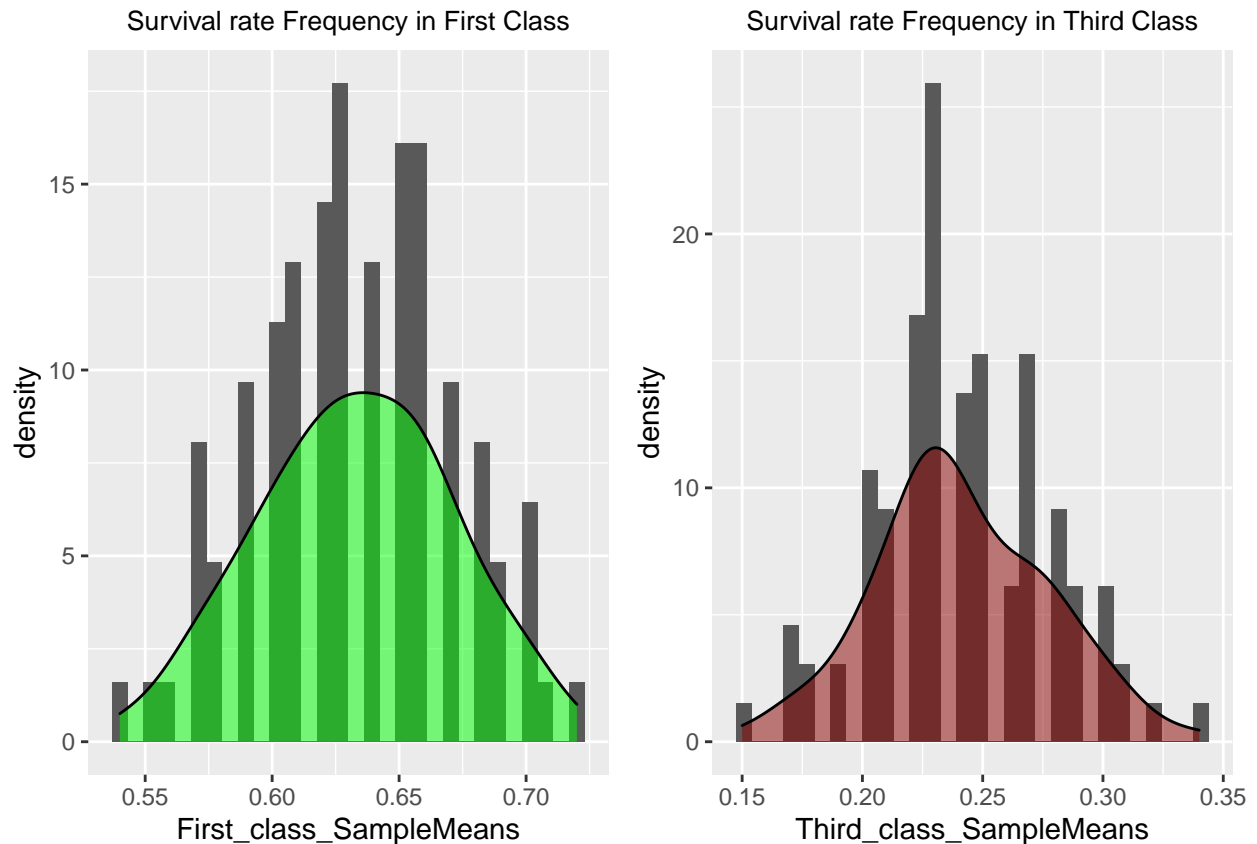
SF_ThirdClass_Density <- ggplot(Third_class_SampleMeans, aes(x = Third_class_SampleMeans)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.5, fill = "darkred") +
```



```
ggtitle("Survival rate Frequency in Third Class") +
theme(plot.title = element_text(size = 10, hjust = 0.5))

ggarrange(SF_FirstClass_Density , SF_ThirdClass_Density)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Para confirmar si ambas distribuciones pueden aproximarse a una normal, aplicamos el test de Shapiro sobre ambas:

Este test toma como: *Hipotesis Nula*: La Variable se distribuye segun una Normal.
Hipotesis Alternativa: La variable NO se distribuye segun una normal.

```
shapiro.test(First_class_SampleMeans$First_class_SampleMeans)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  First_class_SampleMeans$First_class_SampleMeans
## W = 0.98953, p-value = 0.627
```

```
shapiro.test(Third_class_SampleMeans$Third_class_SampleMeans)
```

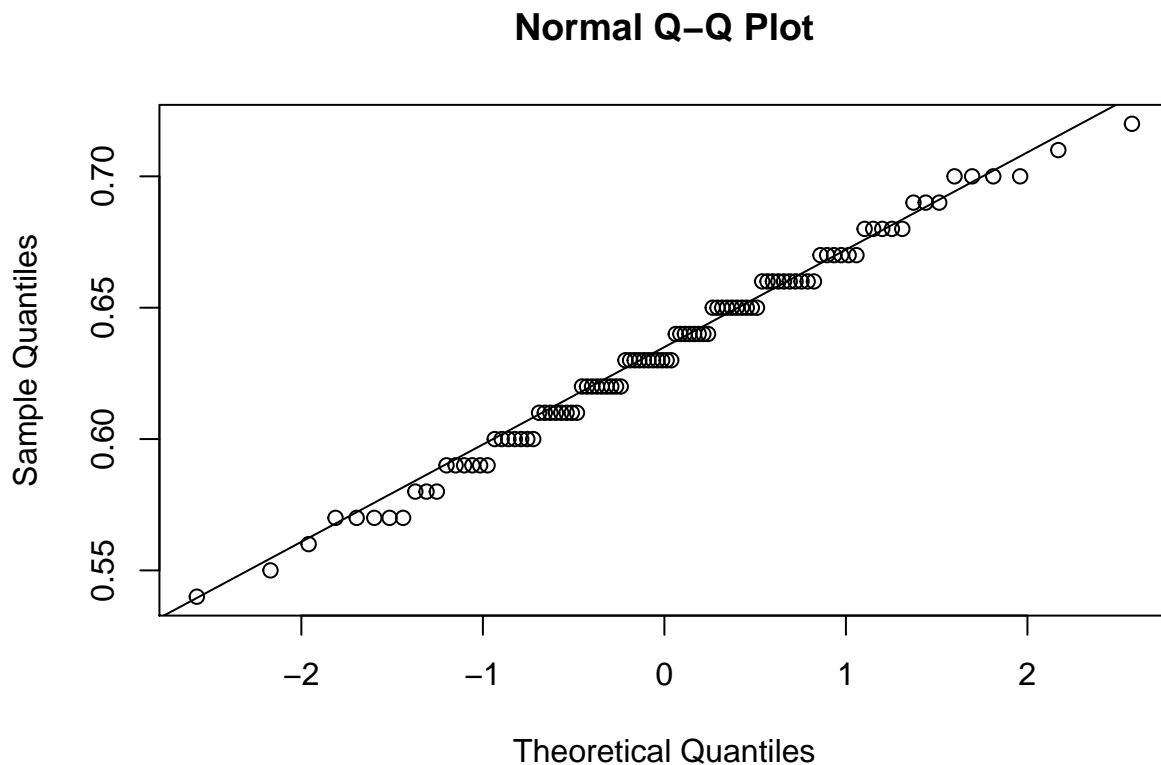
```
##
##  Shapiro-Wilk normality test
##
## data:  Third_class_SampleMeans$Third_class_SampleMeans
```

```
## W = 0.98669, p-value = 0.4173
```

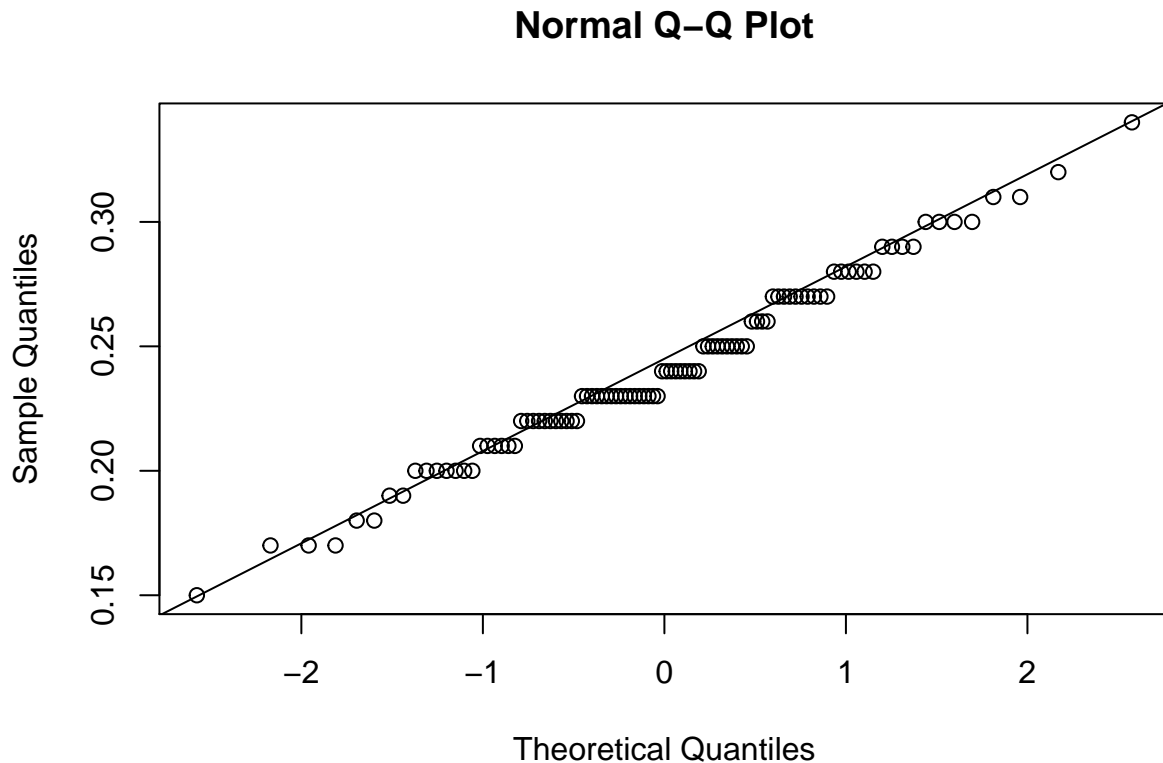
A un nivel de confianza del 95%, los p-values obtenidos son superiores al nivel de significación ($p\text{-value} > 0.05$), por lo que no podemos descartar la Hipotesis Nula (es decir, no podemos descartar que estas distribuciones son normales).

Alternativamente podemos usar los Quantile-Quantile Plots o Q-Q Plots para ver la correlación entre cada variable y una normal, observando que se ajustan a la recta en 45 grados (son variables distribuidas normalmente).

```
qqnorm(First_class_SampleMeans$First_class_SampleMeans)
qqline(First_class_SampleMeans$First_class_SampleMeans)
```



```
qqnorm(Third_class_SampleMeans$Third_class_SampleMeans)
qqline(Third_class_SampleMeans$Third_class_SampleMeans)
```



De manera análoga podemos verificar la homogeneidad de varianzas en ambas muestras utilizando el test de Bartlett:

```
bartlett.test(list(First_class_SampleMeans$First_class_SampleMeans
                  ,Third_class_SampleMeans$Third_class_SampleMeans))

##
## Bartlett test of homogeneity of variances
##
## data: list(First_class_SampleMeans$First_class_SampleMeans, Third_class_SampleMeans$Third_class_SampleMeans)
## Bartlett's K-squared = 0.30977, df = 1, p-value = 0.5778
```

En este caso, el p-value es superior a nuestro nivel de significación (0.05) por lo que no podemos descartar que la diferencia entre las varianzas de ambas muestras sea nula

4.2.2 Comprobación de la normalidad y la homogeneidad de la varianza en muestras de supervivencia por el sexo del pasajero

Al igual que en el apartado anterior, para comprobar la normalidad y la homogeneidad de la varianza, se genera un array de medias sobre 100 submuestras aleatorias de 100 pasajeros de cada uno de los sexos y posteriormente se comprueba la normalidad y la homogeneidad de estos arrays obtenidos.

```
# Generamos un array de medias sobre 100 submuestras aleatorias de 100
# pasajeros de sexo masculino.
iter <- 100
vars <- 1
Male_sex_SampleMeans <- matrix(ncol=vars, nrow=iter)
for(i in 1:iter){
```

```

set.seed(i*16)
Male_sex_SampleMeans[i,] <- mean(as.numeric(as.character
                                     (ttc[sample(which
                                                (ttc$Sex == 'male')
                                                ,100),]$Survived)))
}
mean(Male_sex_SampleMeans)

## [1] 0.189

Male_sex_SampleMeans <- data.frame(Male_sex_SampleMeans)

# Hacemos lo mismo pero para 100 submuestras aleatorias de 100 pasajeros de sexo
# femenino.
Female_sex_SampleMeans <- matrix(ncol=vars, nrow=iter)
for(i in 1:iter){
  set.seed(i*16)
  Female_sex_SampleMeans[i,] <- mean(as.numeric(as.character
                                                  (ttc[sample(which
                                                             (ttc$Sex == 'female')
                                                             ,100),]$Survived)))
}
mean(Female_sex_SampleMeans)

## [1] 0.7425

Female_sex_SampleMeans <- data.frame(Female_sex_SampleMeans)

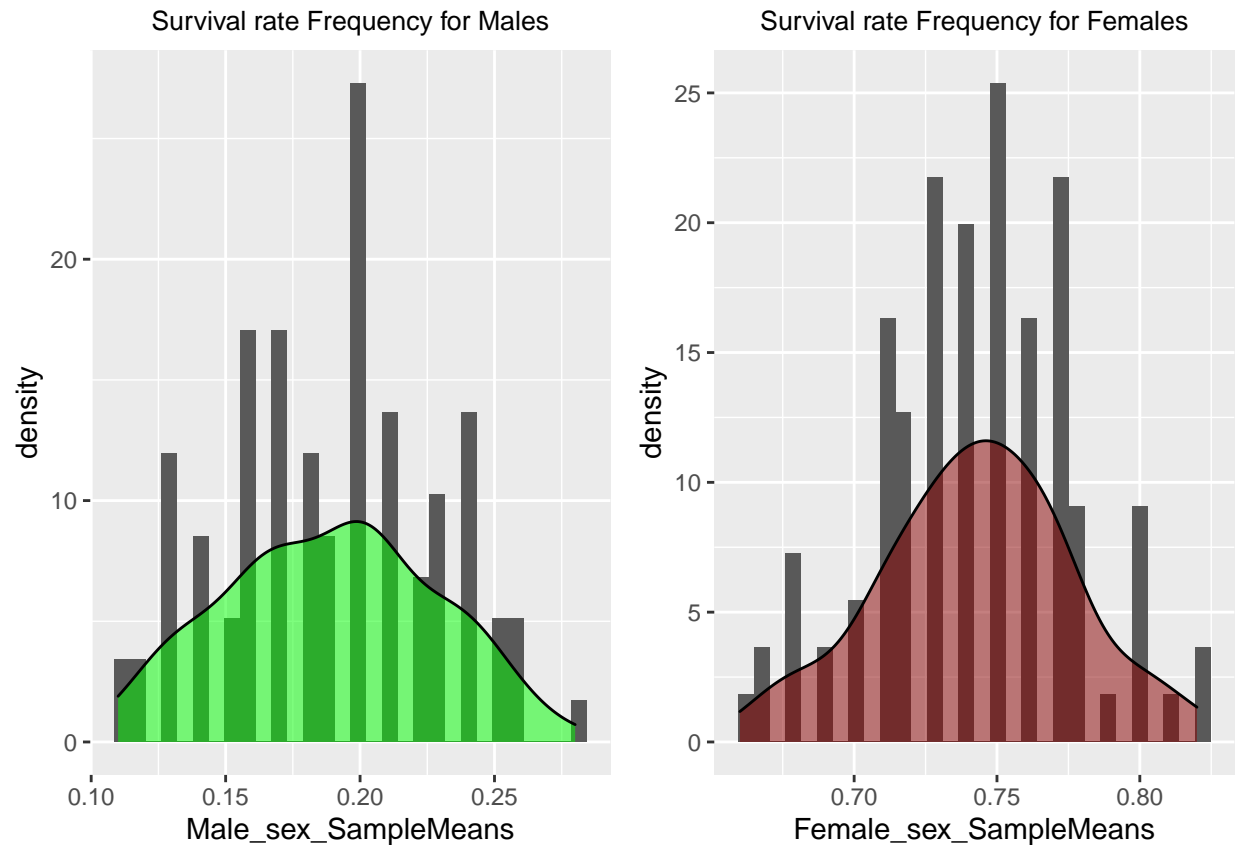
# Revisamos la distribucion de estas variables a nivel grafico de densidad:
SF_MaleSex_Density <- ggplot(Male_sex_SampleMeans, aes(x = Male_sex_SampleMeans)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.5, fill = "green") +
  ggtitle("Survival rate Frequency for Males") +
  theme(plot.title = element_text(size = 10, hjust = 0.5))

SF_FemaleSex_Density <- ggplot(Female_sex_SampleMeans, aes(x = Female_sex_SampleMeans)) +
  geom_histogram(aes(y = ..density..)) +
  geom_density(alpha = 0.5, fill = "darkred") +
  ggtitle("Survival rate Frequency for Females") +
  theme(plot.title = element_text(size = 10, hjust = 0.5))

ggarrange(SF_MaleSex_Density , SF_FemaleSex_Density)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

```



Para comprobar la normalidad, procedemos a aplicar el test de Shapiro.

```
shapiro.test(Male_sex_SampleMeans$Male_sex_SampleMeans)

##
##  Shapiro-Wilk normality test
##
## data:  Male_sex_SampleMeans$Male_sex_SampleMeans
## W = 0.97913, p-value = 0.1137

shapiro.test(Female_sex_SampleMeans$Female_sex_SampleMeans)

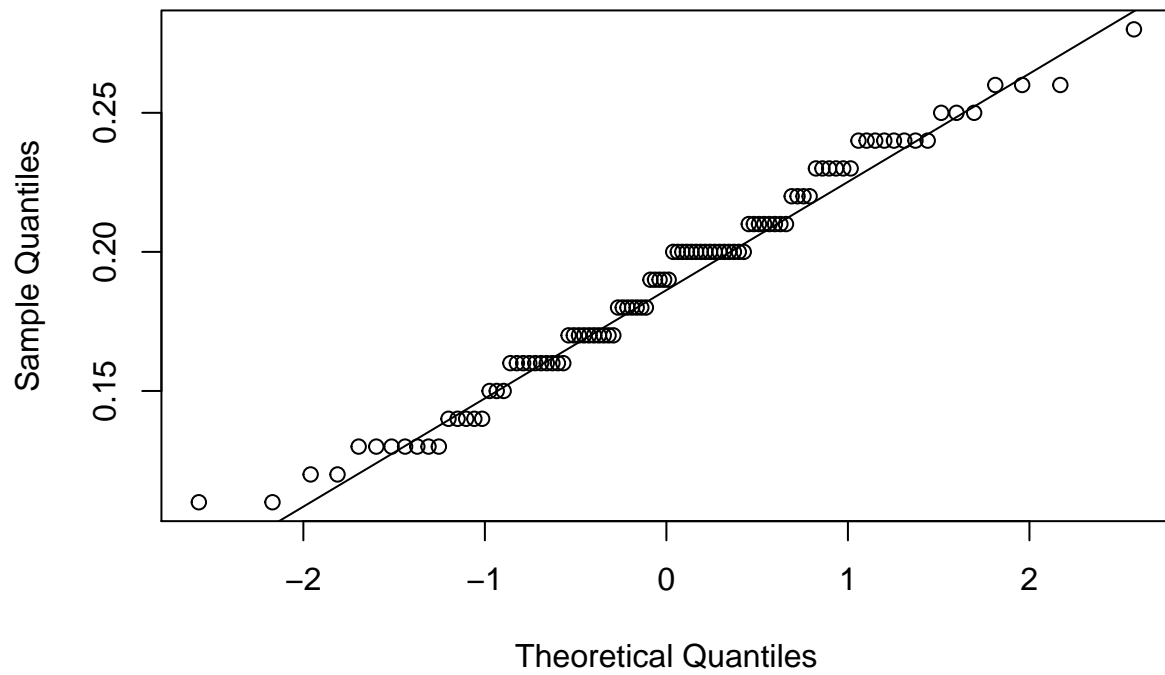
##
##  Shapiro-Wilk normality test
##
## data:  Female_sex_SampleMeans$Female_sex_SampleMeans
## W = 0.98454, p-value = 0.2942
```

Según los resultados obtenidos, no podemos descartar la Hipótesis Nula de Normalidad.

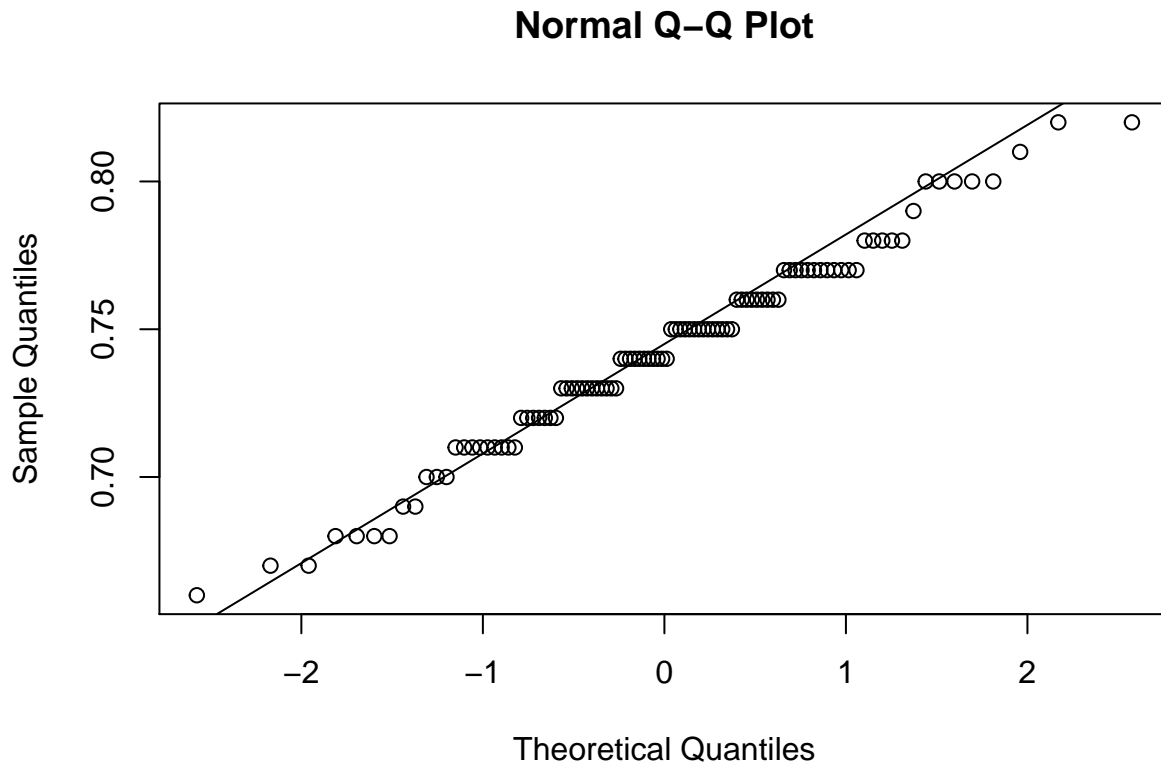
Además, como en el apartado anterior, podemos ver que estos datos se comportan de una forma normal aplicando los Quantile-Quantile Plots o Q-Q Plots.

```
qqnorm(Male_sex_SampleMeans$Male_sex_SampleMeans)
qqline(Male_sex_SampleMeans$Male_sex_SampleMeans)
```

Normal Q-Q Plot



```
qqnorm(Female_sex_SampleMeans$Female_sex_SampleMeans)  
qqline(Female_sex_SampleMeans$Female_sex_SampleMeans)
```



De manera análoga podemos verificar la homogeneidad de varianzas en ambas muestras:

```
# Test de Homogeneidad de Varianzas
```

```
bartlett.test(list(Male_sex_SampleMeans$Male_sex_SampleMeans
                  ,Female_sex_SampleMeans$Female_sex_SampleMeans))
```

```
##
```

```
## Bartlett test of homogeneity of variances
```

```
##
```

```
## data: list(Male_sex_SampleMeans$Male_sex_SampleMeans, Female_sex_SampleMeans$Female_sex_SampleMeans)
```

```
## Bartlett's K-squared = 1.8907, df = 1, p-value = 0.1691
```

En este caso, el p-value es superior a nuestro nivel de significación (0.05) por lo que no podemos descartar que la diferencia entre las varianzas de ambas muestras sea nula.

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

Tras analizar y explorar este dataframe, teniendo en cuenta las variables con las que se cuentan tras realizar la limpieza, hemos considerado interesante realizar los siguientes análisis:

- **Test de hipótesis sobre la proporción de hombres que sobrevivieron frente a la proporción de mujeres** que lo hicieron, planteando como hipótesis alternativa que la proporción de hombres que sobrevivieron es mayor que la proporción de mujeres.
- **Test de hipótesis sobre la proporción de personas que pertenecen a las distintas clases**, planteando como hipótesis alternativa que la proporción de personas que sobrevivieron en primera clase es mayor que la proporción de personas que sobrevivieron de las dos clases restantes.

- **Modelo de Regresión Logística** tratando de predecir si un pasajero sobrevivirá o no.
- **Comparación de modelo de Bosque Aleatorio contra un modelo de Máquinas de Vectores de soporte** para ver cual ofrece mejores resultados.

4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos.

4.3.1 Tests de hipótesis

4.3.1.1 Tests de hipótesis sobre la media de supervivencia según las clases

Planteamos como Hipótesis:

Hipótesis Nula: La media del ratio de supervivencia de los pasajeros en primera clase es la misma que la media del ratio de supervivencia para los pasajeros de tercera clase.

Hipótesis Alternativa: La media del ratio de supervivencia de los pasajeros en primera clase es superior a la media del ratio de supervivencia para los pasajeros de tercera clase.

A continuación, aplicamos un Z test de una cola sobre dos medias con varianzas muestrales distintas, a pesar de que el test de Barlett no descartaba la posibilidad de que estas fuesen iguales.

```
z.test(First_class_SampleMeans$First_class_SampleMeans,
       Third_class_SampleMeans$Third_class_SampleMeans,
       sigma.x=sqrt(var(First_class_SampleMeans$First_class_SampleMeans)),
       sigma.y=sqrt(var(Third_class_SampleMeans$Third_class_SampleMeans)),
       alternative="greater",
       conf.level = 0.95)

##
## Two-sample z-Test
##
## data: First_class_SampleMeans$First_class_SampleMeans and Third_class_SampleMeans$Third_class_SampleMeans
## z = 74.496, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3833448      NA
## sample estimates:
## mean of x mean of y
##    0.6331    0.2411
```

El resultado obtenido por el test es que el p-value es inferior a nuestro nivel de significación (0.05). Lo que quiere decir, que con un nivel de confianza del 95%, podemos descartar la hipótesis nula y aceptar la alternativa:

- **La media del ratio de supervivencia de los pasajeros en primera clase es superior a la media del ratio de supervivencia para los pasajeros de tercera clase**

4.3.1.2 Tests de hipótesis sobre la media de supervivencia según el sexo

Para este contraste fijamos las siguientes hipótesis:

Hipótesis Nula: La media del ratio de supervivencia de los hombres es la misma que la media del ratio de supervivencia para las mujeres.

Hipótesis Alternativa: La media del ratio de supervivencia de los hombres es inferior a la media del ratio de supervivencia para las mujeres.

A continuación, aplicamos un Z test de una cola sobre dos medias con varianzas muestrales distintas, a pesar de que el test de Barlett no descartaba la posibilidad de que estas fuesen iguales.

```
z.test(Male_sex_SampleMeans$Male_sex_SampleMeans,
       Female_sex_SampleMeans$Female_sex_SampleMeans,
       sigma.x=sqrt(var(Male_sex_SampleMeans$Male_sex_SampleMeans)),
```



```

sigma.y=sqrt(var(Female_sex_SampleMeans$Female_sex_SampleMeans)),
alternative="less",
conf.level = 0.95)

##
## Two-sample z-Test
##
## data: Male_sex_SampleMeans$Male_sex_SampleMeans and Female_sex_SampleMeans$Female_sex_SampleMeans
## z = -107.15, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      NA -0.5450032
## sample estimates:
## mean of x mean of y
##      0.1890      0.7425

# El resultado es que el p-value es inferior a nuestro nivel de confianza. Esto es, con un
# nivel de confianza del 95%, podemos descartar la hipótesis nula y aceptar la
# alternativa:
# La media del ratio de supervivencia de los hombres es inferior a la media del ratio de
# supervivencia para las mujeres

```

El resultado obtenido por el test es que el p-value es inferior a nuestro nivel de significación (0.05). Lo que quiere decir, que con un nivel de confianza del 95%, podemos descartar la hipótesis nula y aceptar la alternativa:

- **La media del ratio de supervivencia de los hombres es inferior a la media del ratio de supervivencia para las mujeres**

4.3.2 Modelos

4.3.2.1 Acotación principal de variables a utilizar De todas las variables con las que cuenta el dataframe que se está analizando en esta práctica, se descartan de primeras las siguientes variables para la construcción del modelo predictivo:

- **PassengerId**
- **Name**
- **Ticket**

Estas variables no tiene sentido utilizarlas para construir un modelo predictivo porque son variables que identifican a cada uno de los pasajeros y pasajeras.

Por lo tanto, las únicas variables que tiene sentido poder utilizar para la construcción del modelo predictivo son las siguientes:

- **Pclass**
- **Sex**
- **Age**
- **SibSp**
- **Parch**
- **Fare**
- **Embarked**

Y la variable **Survived**, que nos servirá como variable objetivo del modelo.

De todas las posibles variables a utilizar que se han citado, se considera que la variable **Fare** y la variable **Pclass** vienen a indicar prácticamente la misma información o muy similar, pues la variable **Pclass** indica la clase en la que se encuentra el pasajero/a y la variable **Fare** el precio del billete o ticket de embarque.

Normalmente a mayor precio del billete, más alta será la clase, aunque puede que existan casos que sean precios más altos pero pertenezcan a una clase baja y que dicho precio se haya visto incrementado porque se vendió durante los últimos días disponibles de venta del mismo. Pero de normal, a un mayor precio, más alta será la clase, lo que quiere decir que ambas variables están muy relacionadas entre sí. En los modelos predictivos, utilizar variables que de alguna manera indiquen la misma información puede hacer que empeore el modelo o en el mejor de los casos, no aportará mucha información al mismo, por lo que conviene elegir una de estas dos variables. En este caso, como el modelo predictivo es de clasificación, tiene más sentido utilizar la variable **Pclass** y eliminar por tanto la variable **Fare**.

De esta manera, las únicas variables que pueden tener sentido utilizar para la construcción del modelo predictivo son las siguientes:

- **Survived** (como variable objetivo)
- **Pclass**
- **Sex**
- **Age**
- **SibSp**
- **Parch**
- **Embarked**

De todas estas posibles variables a utilizar, determinaremos cual utilizar o no realizando una selección de características. Para ello, se estudiará la relación de cada una de las variables con la variable objetivo mediante el test ***Chi-squared***, de manera que sólo se utilizarán para el modelo predictivo aquellas variables que podamos afirmar con un grado de confianza determinado que presentan cierta relación con la variable objetivo.

Antes de realizar la selección de características, será necesario convertir aquellas variables numéricas a variables categóricas dividiendo las mismas en intervalos, ya que las variables numéricas no proporcionan mucha información en los modelos de clasificación.

Se procede a continuación a aplicar dicha conversión para todas las variables numéricas.

4.3.2.2 Data Binning A la conversión de variables numéricas a variables categóricas agrupando las mismas por intervalos, se le conoce como ***Data Binning*** o **Agrupación de Datos**. Esta técnica, presenta diferentes variantes de aplicación, pero en este caso dividiremos las variables numéricas por intervalos del mismo tamaño.

Comenzaremos con la variable **Age**, dividiendo la misma en rangos de 5 años.

```
ttc$Age <- cut(ttc$Age,breaks = 5*(0:16))
head(ttc$Age)

## [1] (20,25] (35,40] (25,30] (30,35] (30,35] (20,25]
## 16 Levels: (0,5] (5,10] (10,15] (15,20] (20,25] (25,30] (30,35] ... (75,80]
```

Por otro lado, las variables **Parch** y **SibSp** se dividirán en grupos de 2.

```
ttc$SibSp <- cut(ttc$SibSp,breaks = 2*(0:4), include.lowest = TRUE)
ttc$Parch <- cut(ttc$Parch,breaks = 2*(0:3), include.lowest = TRUE)

head(ttc$SibSp)

## [1] [0,2] [0,2] [0,2] [0,2] [0,2] [0,2]
## Levels: [0,2] (2,4] (4,6] (6,8]

head(ttc$Parch)
```

```
## [1] [0,2] [0,2] [0,2] [0,2] [0,2] [0,2]
## Levels: [0,2] (2,4] (4,6]
```

4.3.2.3 Selección de características Una vez realizadas las conversiones necesarias, como se ha explicado anteriormente, en el apartado 4.3.1, se procede a aplicar el test **Chi-squared** para cada una de las posibles variables a utilizar para construir el modelo predictivo, con la variable objetivo.

El test **Chi-squared** es un test de hipótesis que consta de las siguientes hipótesis:

- **Hipótesis nula:** Las variables sobre las que se está aplicando el test son independientes una de la otra.
- **Hipótesis alternativa:** Las variables sobre las que se está aplicando el test están relacionadas una con la otra.

El resultado de este test indicará la probabilidad de que la Hipótesis nula se cumpla, y se deberá fijar un nivel de confianza que nos permita rechazar esta o no. Al aplicar este test se suele utilizar un nivel de confianza de un 95%, que será el que utilizaremos nosotros también. De esta manera, rechazaremos la Hipótesis nula cuando la probabilidad de que esta se cumpla sea menor a un 5%. Hablamos de probabilidades porque se está midiendo la relación de dos variables aleatorias, por lo que no se puede asegurar sin un margen de error que estas están relacionadas o no.

A continuación, se procede a aplicar dicho test para cada una de las variables candidatas a formar el modelo predictivo.

El orden que seguiremos para aplicar el test chi-cuadrado, es el siguiente:

- 1. Pclass
- 2. Sex
- 3. Age
- 4. SibSp
- 5. Parch
- 6. Embarked

```
tab_Pclass <- table(ttc$Pclass,ttc$Survived)
chisq_Pclass <- chisq.test(tab_Pclass)
chisq_Pclass
```

```
##
## Pearson's Chi-squared test
##
## data:  tab_Pclass
## X-squared = 100.98, df = 2, p-value < 2.2e-16
```

Dado que el p-valor obtenido es menor que 0.05, se puede afirmar con un 95% de probabilidad que la variable **Pclass** se encuentra relacionada con la variable **Survived**, por lo que esta será utilizada para formar el modelo predictivo.

```
related_features <- c("Pclass")
```

```
tab_Sex <- table(ttc$Sex, ttc$Survived)
chisq_sex <- chisq.test(tab_Sex)
chisq_sex
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab_Sex
```

```
## X-squared = 258.43, df = 1, p-value < 2.2e-16
```

Dado que el p-valor obtenido es menor que 0.05, se puede afirmar con un 95% de probabilidad que la variable **Sex** se encuentra relacionada con la variable **Survived**, por lo que esta será utilizada para formar el modelo predictivo.

```
related_features <- c(related_features,"Sex")
```

```
tab_Age <- table(ttc$Age, ttc$Survived)
chisq_Age <- chisq.test(tab_Age)
```

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

```
chisq_Age
```

```
##
## Pearson's Chi-squared test
##
## data:  tab_Age
## X-squared = 38.45, df = 15, p-value = 0.0007739
```

Dado que el p-valor obtenido es menor que 0.05, se puede afirmar con un 95% de probabilidad que la variable **Age** se encuentra relacionada con la variable **Survived**, por lo que esta será utilizada para formar el modelo predictivo.

```
related_features <- c(related_features,"Age")
```

```
tab_SibSp <- table(ttc$SibSp, ttc$Survived)
chisq_SibSp <- chisq.test(tab_SibSp)
```

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

```
chisq_SibSp
```

```
##
## Pearson's Chi-squared test
##
## data:  tab_SibSp
## X-squared = 12.483, df = 3, p-value = 0.005898
```

Dado que el p-valor obtenido es menor que 0.05, se puede afirmar con un 95% de probabilidad que la variable **SibSp** se encuentra relacionada con la variable **Survived**, por lo que esta será utilizada para formar el modelo predictivo.

```
related_features <- c(related_features,"SibSp")
```

```
tab_Parch <- table(ttc$Parch, ttc$Survived)
chisq_Parch <- chisq.test(tab_Parch)
```

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

```
chisq_Parch
```

```
##
## Pearson's Chi-squared test
##
## data:  tab_Parch
## X-squared = 1.2895, df = 2, p-value = 0.5248
```

Dado que el p-valor obtenido es mayor que 0.05, no se puede afirmar que la variable **Parch** se encuentra relacionada con la variable **Survived**, por lo que esta no será utilizada para formar el modelo predictivo.

```
tab_Embarked <- table(ttc$Embarked, ttc$Survived)
chisq_Embarked <- chisq.test(tab_Embarked)
chisq_Embarked
```

```
##
## Pearson's Chi-squared test
##
## data:  tab_Embarked
## X-squared = 26.489, df = 2, p-value = 1.77e-06
```

Dado que el p-valor obtenido es menor que 0.05, se puede afirmar con un 95% de probabilidad que la variable **Embarked** se encuentra relacionada con la variable **Survived**, por lo que esta será utilizada para formar el modelo predictivo.

```
related_features <- c(related_features, "Embarked")
```

Tras realizar el test a todas las variables que había disponibles para formar el modelo, quedan únicamente para formar este las siguientes:

```
related_features
```

```
## [1] "Pclass" "Sex" "Age" "SibSp" "Embarked"
```

A continuación, procedemos a añadir al vector **related_features** la variable objetivo, para poder obtener del dataframe original todas estas variables que son las que se utilizarán para construir el modelo predictivo.

```
related_features <- c(related_features, "Survived")
ttc_model <- ttc %>% select(related_features)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(related_features)` instead of `related_features` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
head(ttc_model)
```

```
##   Pclass   Sex   Age SibSp Embarked Survived
## 1     3  male (20,25] [0,2]         S         0
## 2     1 female (35,40] [0,2]         C         1
## 3     3 female (25,30] [0,2]         S         1
## 4     1 female (30,35] [0,2]         S         1
## 5     3  male (30,35] [0,2]         S         0
## 6     3  male (20,25] [0,2]         Q         0
```

Antes de la aplicación de los modelos, procedemos a preparar el dataset para poder ser utilizado para crear los diferentes modelos predictivos.

```
ttc_vectors <- ttc_model %>%
  mutate_if(is.factor, as.numeric)
head(ttc_vectors)
```

```
##   Pclass Sex Age SibSp Embarked Survived
## 1     3   2  5     1         3         1
## 2     1   1  8     1         1         2
## 3     3   1  6     1         3         2
## 4     1   1  7     1         3         2
## 5     3   2  7     1         3         1
## 6     3   2  5     1         2         1
```

```
ttc_vectors$Survived <- as.factor(ttc_vectors$Survived)

ttc_vectors$Survived <- revalue(ttc_vectors$Survived, c("1"="NO", "2"="YES"))
head(ttc_vectors)
```

```
##   Pclass Sex Age SibSp Embarked Survived
## 1      3  2  5      1        3        NO
## 2      1  1  8      1        1        YES
## 3      3  1  6      1        3        YES
## 4      1  1  7      1        3        YES
## 5      3  2  7      1        3        NO
## 6      3  2  5      1        2        NO
```

```
train_test_split <- initial_split(ttc_vectors, prop=3/4)
```

```
TRAIN <- training(train_test_split)
TEST <- testing(train_test_split)
```

```
head(TRAIN)
```

```
##   Pclass Sex Age SibSp Embarked Survived
## 1      3  2  5      1        3        NO
## 2      1  1  8      1        1        YES
## 3      3  1  6      1        3        YES
## 5      3  2  7      1        3        NO
## 6      3  2  5      1        2        NO
## 10     2  1  3      1        1        YES
```

```
head(TEST)
```

```
##   Pclass Sex Age SibSp Embarked Survived
## 4      1  1  7      1        3        YES
## 7      1  2 11      1        3        NO
## 8      3  2  1      2        3        NO
## 9      3  1  6      1        3        YES
## 11     3  1  1      1        3        YES
## 16     2  1 11      1        3        YES
```

4.3.2.4 Regresión Logística A continuación, se procede a la creación del modelo predictivo regresión logística.

```
control <- trainControl(method="repeatedcv", number=10, repeats=3,
                        summaryFunction = twoClassSummary, classProbs = TRUE)
glm <- train(Survived~., data = TRAIN, method="glm", trControl=control)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

Este modelo de regresión logística, se entrena aplicando para ello una validación cruzada anidada. De esta forma, esta búsqueda de mejores hiperparámetros se realiza dividiendo el conjunto de datos de entrenamiento en 10 bloques, de esos 10 bloques se utilizará en cada iteración uno de ellos para probar el modelo y el resto para entrenar el mismo. Este proceso se repetirá 3 veces y eso nos permitirá obtener de manera realista como se comporta el modelo creado, pues probándolo únicamente con la parte dedicada a los test, dicha parte puede estar sesgada. Por lo que de esta manera, nos dará una visión más realista de como funciona el modelo.

Este proceso, además debería buscar los mejores hiperparámetros para el modelo basándose en la métrica AUC, pero la regresión logística en el paquete **Caret**, no permite especificar ningún parámetro de ajuste, por lo que no se podrá aprovechar al 100% el potencial de esta función.

Los resultados obtenidos por el modelo tras entrenar el mismo son los siguientes:

```
glm

## Generalized Linear Model
##
## 667 samples
## 5 predictor
## 2 classes: 'NO', 'YES'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 601, 599, 600, 600, 600, 600, ...
## Resampling results:
##
## ROC          Sens          Spec
## 0.8616353    0.8897764    0.7266857
```

Estos resultados son la media de los resultados obtenidos al aplicar la validación cruzada anidada.

Como se puede observar, se tratan de resultados bastante buenos.

A continuación, se procede a probar el modelo creado y obtener las métricas resultantes con los datos destinados para testear el modelo.

```
glm.TEST <- predict(glm, TEST)

class_glm <- predict(glm, TEST)

confusionMatrix(data=class_glm, TEST$Survived)

## Confusion Matrix and Statistics
##
##              Reference
## Prediction  NO YES
##          NO 118  20
##          YES  25  59
##
##              Accuracy : 0.7973
##              95% CI : (0.7383, 0.8481)
##      No Information Rate : 0.6441
##      P-Value [Acc > NIR] : 4.813e-07
##
##              Kappa : 0.564
##
##  Mcnemar's Test P-Value : 0.551
##
##              Sensitivity : 0.8252
##              Specificity : 0.7468
##              Pos Pred Value : 0.8551
##              Neg Pred Value : 0.7024
##              Prevalence : 0.6441
##              Detection Rate : 0.5315
```

```
## Detection Prevalence : 0.6216
## Balanced Accuracy : 0.7860
##
## 'Positive' Class : NO
##
```

4.3.2.4 Bosque Aleatorio A continuación, se procede a la creación del modelo predictivo utilizando el algoritmo de bosque aleatorio.

```
rf <- train(Survived~., data = TRAIN, method="rf", trControl=control,
            tuneLength=4)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

```
rf
```

```
## Random Forest
##
## 667 samples
## 5 predictor
## 2 classes: 'NO', 'YES'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 600, 600, 600, 600, 600, 600, ...
## Resampling results across tuning parameters:
##
## mtry ROC Sens Spec
## 2 0.8697170 0.9418496 0.6591168
## 3 0.8733973 0.9287398 0.6707502
## 4 0.8697917 0.9122358 0.6784425
## 5 0.8648454 0.8966667 0.6861823
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 3.
```

En este caso, como se puede observar, además de aplicar el método de validación cruzada anidada para validar el modelo, se trata de buscar los mejores valores de los hiperparámetros que permitan al modelo obtener las mejores métricas. En este caso, el hiperparámetro es *mtry* y el valor que ha resultado ser mejor, como aparece en la visualización anterior, es el **2**.

Se procede a continuación además a obtener los resultados para los datos destinados para testear el modelo.

```
rf.TEST <- predict(rf, TEST)
class_rf <- predict(rf, TEST)

confusionMatrix(data=class_rf, TEST$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction NO YES
## NO 129 31
## YES 14 48
##
## Accuracy : 0.7973
```



```
##          95% CI : (0.7383, 0.8481)
##    No Information Rate : 0.6441
##    P-Value [Acc > NIR] : 4.813e-07
##
##          Kappa : 0.5355
##
## Mcnemar's Test P-Value : 0.01707
##
##          Sensitivity : 0.9021
##          Specificity : 0.6076
##    Pos Pred Value : 0.8063
##    Neg Pred Value : 0.7742
##          Prevalence : 0.6441
##    Detection Rate : 0.5811
##    Detection Prevalence : 0.7207
##    Balanced Accuracy : 0.7548
##
##    'Positive' Class : NO
##
```

Los resultados obtenidos son mejores aún que en la regresión logística, lo que es normal ya que el algoritmo de bosque aleatorio es un algoritmo más potente que la regresión logística.

4.3.2.5 Máquinas de Vectores de Soporte A continuación, se procede a la creación del modelo predictivo utilizando el algoritmo de máquinas de vectores de soporte de clasificación.

```
svm_poly <- train(Survived~., data = TRAIN, method="svmPoly",
                  trControl=control,preProcess = c("center","scale"),
                  tuneLength = 4)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was not
## in the result set. ROC will be used instead.
```

```
svm_poly
```

```
## Support Vector Machines with Polynomial Kernel
##
## 667 samples
## 5 predictor
## 2 classes: 'NO', 'YES'
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 601, 601, 600, 600, 601, 600, ...
## Resampling results across tuning parameters:
##
##  degree  scale  C      ROC      Sens      Spec
##  1       0.001  0.25  0.8528727  0.8466870  0.7097816
##  1       0.001  0.50  0.8528727  0.8458740  0.7072175
##  1       0.001  1.00  0.8528727  0.8475203  0.7084995
##  1       0.001  2.00  0.8528415  0.8475000  0.7084995
##  1       0.010  0.25  0.8528430  0.8483333  0.7072175
##  1       0.010  0.50  0.8570069  0.8786789  0.6663343
##  1       0.010  1.00  0.8377985  0.8794919  0.6663343
##  1       0.010  2.00  0.8183970  0.8942683  0.6547958
##  1       0.100  0.25  0.8110421  0.8991870  0.6547958
```

```
## 1      0.100 0.50 0.8380649 0.9040650 0.6547958
## 1      0.100 1.00 0.8256508 0.9040650 0.6547958
## 1      0.100 2.00 0.8376969 0.9040650 0.6547958
## 1      1.000 0.25 0.8351951 0.9040650 0.6547958
## 1      1.000 0.50 0.8447163 0.9040650 0.6547958
## 1      1.000 1.00 0.8341676 0.9040650 0.6547958
## 1      1.000 2.00 0.8399246 0.9040650 0.6547958
## 2      0.001 0.25 0.8526863 0.8466870 0.7110636
## 2      0.001 0.50 0.8526863 0.8475000 0.7084995
## 2      0.001 1.00 0.8526550 0.8475000 0.7084995
## 2      0.001 2.00 0.8540066 0.8786789 0.6663343
## 2      0.010 0.25 0.8580079 0.8794919 0.6663343
## 2      0.010 0.50 0.8548731 0.8794919 0.6663343
## 2      0.010 1.00 0.8475314 0.8934553 0.6560779
## 2      0.010 2.00 0.8481314 0.9040650 0.6547958
## 2      0.100 0.25 0.8521917 0.9040650 0.6547958
## 2      0.100 0.50 0.8521466 0.9040650 0.6560779
## 2      0.100 1.00 0.8506474 0.9040650 0.6624406
## 2      0.100 2.00 0.8509920 0.8958740 0.6841406
## 2      1.000 0.25 0.8477101 0.9237602 0.6690883
## 2      1.000 0.50 0.8490539 0.9278252 0.6716524
## 2      1.000 1.00 0.8492660 0.9369106 0.6626781
## 2      1.000 2.00 0.8507547 0.9319512 0.6716524
## 3      0.001 0.25 0.8527488 0.8466870 0.7084995
## 3      0.001 0.50 0.8527488 0.8466870 0.7084995
## 3      0.001 1.00 0.8534426 0.8745732 0.6701804
## 3      0.001 2.00 0.8579024 0.8794919 0.6663343
## 3      0.010 0.25 0.8605824 0.8794919 0.6663343
## 3      0.010 0.50 0.8479870 0.8819512 0.6624881
## 3      0.010 1.00 0.8491597 0.9016260 0.6547958
## 3      0.010 2.00 0.8522722 0.9040650 0.6547958
## 3      0.100 0.25 0.8485804 0.9040650 0.6637227
## 3      0.100 0.50 0.8472187 0.9024390 0.7008072
## 3      0.100 1.00 0.8513809 0.9106911 0.6830959
## 3      0.100 2.00 0.8478186 0.9377236 0.6664767
## 3      1.000 0.25 0.8493558 0.9499593 0.6435897
## 3      1.000 0.50 0.8535171 0.9556911 0.6423077
## 3      1.000 1.00 0.8501076 0.9565041 0.6447768
## 3      1.000 2.00 0.8514822 0.9516463 0.6434948
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were degree = 3, scale = 0.01 and C = 0.25.
```

En este modelo, al igual que en el anterior, se trata de buscar los mejores hiperparámetros para el mismo, aplicando para ello un algoritmo de validación cruzada anidada. En este caso el hiperparámetro es **C** y el valor que ha resultado tener mejores resultados es cuando este tiene un valor de **0.25**. En este caso, además se ha aplicado un preprocesamiento de centrado y escalado de los datos. Este preprocesado aplica el escalado y centrado sobre los datos de entrenamiento que mejores resultados permita dar.

Se procede a continuación además a obtener los resultados para los datos destinados para testear el modelo.

```
class_svm_poly <- predict(svm_poly, TEST)
confusionMatrix(data=class_svm_poly, TEST$Survived)
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  NO YES
##           NO 115 22
##           YES 28 57
##
##           Accuracy : 0.7748
##           95% CI : (0.7141, 0.828)
##           No Information Rate : 0.6441
##           P-Value [Acc > NIR] : 1.801e-05
##
##           Kappa : 0.5169
##
## Mcnemar's Test P-Value : 0.4795
##
##           Sensitivity : 0.8042
##           Specificity : 0.7215
##           Pos Pred Value : 0.8394
##           Neg Pred Value : 0.6706
##           Prevalence : 0.6441
##           Detection Rate : 0.5180
##           Detection Prevalence : 0.6171
##           Balanced Accuracy : 0.7629
##
##           'Positive' Class : NO
##
```

4.3.2.6 Prueba con el dataset test Una vez contruidos y comparados los modelos, se procede a obtener los resultados para el archivo `test.csv`, aplicando para ello el modelo que mejores resultados ha obtenido, que es el modelo de **bosque aleatorio**.

Previamente a la aplicación del modelo predictivo, este dataset será procesado con los cambios que se aplicaron al dataset **train**, es decir, se realizará un redondeo de las edades que tengan decimales y una imputación de aquellas edades que no se tengan información.

```
ttc_test_origin <- read.csv("./Data/test.csv",na.strings=c("", " ", "NA"),
                           stringsAsFactors = TRUE)
head(ttc_test_origin)
```

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

```
features<-head(related_features,-1)
ttc_test <- ttc_test_origin %>% select(features)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(features)` instead of `features` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
head(ttc_test)
```

```
##   Pclass   Sex Age SibSp Embarked
## 1     3  male 34.5    0        Q
## 2     3 female 47.0    1        S
## 3     2  male 62.0    0        Q
## 4     3  male 27.0    0        S
## 5     3 female 22.0    1        S
## 6     3  male 14.0    0        S
```

```
ttc_test$Age <- apply(ttc_test,1,roundValues)
ttc_test$Age <- as.numeric(ttc_test$Age)
```

```
imputationFunct_test <- function(x){
  if (is.na(x["Age"])){
    x["Age"]<- median(ttc_test$Age[ttc_test$Pclass==x["Pclass"] &
                      !is.na (ttc_test$Age)])
  } else{
    x<-x
  }
  return (x["Age"])
}
```

```
ttc_test$Age <- apply(ttc_test,1,imputationFunct_test)
ttc_test$Age <- as.numeric(ttc_test$Age)
```

```
ttc_test_vectors <- ttc_test %>%
  mutate_if(is.factor, as.numeric)
head(ttc_test_vectors)
```

```
##   Pclass Sex Age SibSp Embarked
## 1     3   2  34     0         2
## 2     3   1  47     1         3
## 3     2   2  62     0         2
## 4     3   2  27     0         3
## 5     3   1  22     1         3
## 6     3   2  14     0         3
```

```
ttc_test_vectors.class <- predict(rf, ttc_test_vectors)
ttc_test_origin$Survived <- ttc_test_vectors.class
ttc_test_df = ttc_test_origin %>% select("PassengerId", "Survived")
ttc_test_df$Survived = revalue(ttc_test_df$Survived, c("YES"=1, "NO"=0))
ttc_test_df$Survived = as.numeric(ttc_test_df$Survived)
```

```
write.csv(ttc_test_df,"./Data/test_results.csv", row.names = FALSE)
```

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





5.2 Tabla resumen de los resultados obtenidos en los modelos predictivos

Algoritmos	ROC	Sens	Spec
Regresión Logística	0.844145	0.8627236	0.7026667
Bosque Aleatorio	0.8465613	0.9404268	0.6420513
Máquina de Vectores de Soporte	0.8372377	0.8024593	0.7386154

6. Conclusiones obtenidas

- Con respecto a los modelos predictivos, tras la aplicación de los modelos de: **regresión logística, bosque aleatorio y máquinas de vectores de soporte de clasificación**, el modelo que **mejores resultados** ha obtenido es el modelo que utiliza el algoritmo de **bosque aleatorio**. El modelo de regresión logística fue aplicado para probar los resultados obtenidos con el modelo de clasificación más simple por excelencia, pero los resultados de calidad obtenidos por el mismo han sido bastante buenos, de hecho, mejores que los obtenidos por el modelo de máquinas de vectores de soporte de clasificación, a pesar de que este último es un modelo más potente.

Contribuciones

Contribuciones	Firma	
Investigación Previa		
Redacción de las respuestas		
Desarrollo código	