# Titanic Data Clean

*Ricardo Garcia Ruiz*

*10 de junio, 2018*

## Contents

# 1 Práctica 2: Limpieza y validación de los datos

## 1.1 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) * Predict Future Sales (https://www.kaggle.com/c/competitive-data-sciencepredict-future-sales/).

Los últimos dos ejemplos corresponden a competiciones activas de Kaggle de manera que, opcionalmente, podríais aprovechar el trabajo realizado durante la práctica para entrar en alguna de estas competiciones.

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?
2. Integración y selección de los datos de interés a analizar.
3. Limpieza de los datos:
   3.1. ¿Los datos contienen ceros o elementos vacíos? ¿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 estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.
5. Representación de los resultados a partir de tablas y gráficas.
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 preferís, también podéis trabajar en Python.

## 1.2 Objetivos

Los objetivos concretos de esta práctica son:

- 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.

- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.

- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.

- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuar estudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

### 1.3 Competencias

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

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

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

El proceso de integración y selección de los datos a analizar se irá realizando a lo largo del proceso de limpieza y analisis de las distintas variables del conjunto de datos de entrenamiento de Titanic.

Realizarlo de una manera aprioristica no mejora el proceso de verificación de comportamiento de las variables del conjunto ni da en ningún caso un planteamiento del problema mejorado.

En este proceso se pretende ir analizando las distintas variables en el proceso de limpieza y análisis, y en función de las características observables de las diversas variables se tomará la decisión de utilizar un conjunto seleccionado de las mismas que pueda ser útil para la predicción del modelo y la comprobación con el conjunto de test o validación.

## 3 Proceso de limpieza del dataset titanic

### 3.1 Descripción del dataset

El conjunto de datos de análisis escogido ha sido finalmente el Titanic de Kaggle [https://www.kaggle.com/c/titanic/data].

El conjunto de datos se encuentra descrito en la tabla siguiente, contando con 891 filas o registros de datos para cada variable:

| Variable | Definición | Clave |
|---|---|---|
| survived | Supervivencia | 0 = No, 1 = Sí |
| pclass | Clase de ticket | 1 = Primera, 2 = Segunda, 3 = Tercera |
| name | nombre | |
| sex | Sexo | |
| Age | Edad | en años |
| sibsp | # hermanos / cónyuges a bordo del Titanic | |

| Variable | Definición | Clave |
|---|---|---|
| parch | # padres / niños a bordo del Titanic | |
| ticket | Numero de ticket | |
| fare | Tarifa del pasajero | |
| cabin | Número de cabina | |
| embarked | Puerto de embarque | C = Cherbourg, Q = Queenstown, S = Southampton |

Adicionalmente, para comprender los elementos del dataset es preciso tomar en consideración las siguientes notas adicionales:

**pclass**: Esta variable es indicadora indirecta del estado socioeconómico al que pertenecería cada pasajero:

- 1 = Clase alta
- 2 = Clase media
- 3 = Clase baja

**age**: La edad es fraccional si es menor que 1. Si la edad es estimada, ¿está en la forma de xx.5

**sibsp**: En el dataset se definen las relaciones familiares de esta manera:

- Hermano = hermano, hermana, hermanastro, hermanastra
- Cónyuge = esposo, esposa (las amantes y los novios fueron ignorados)

**parch**: En el dataset se definen las relaciones familiares de esta manera:

- Padre = madre, padre
- Hijo = hija, hijo, hijastra, hijastro
- Algunos niños viajaron solo con una niñera, por lo tanto parch = 0 para ellos.

Este conjunto de datos puede responder a las causas de las muertes en el naufragio del Titanic, permitiendo establecer modelos de inferencia sobre las causas últimas relativas a la mortandad entre diversos tipos de pasajeros

También puede facilitar un modelo de interés sobre qué variables han influido en las muertes, causas circunstanciales que influyeron finalmente a pesar de las medidas de seguridad del barco y de la dotación.

## 3.2 Integración y selección de los datos de interés a analizar

Antes de comenzar con la limpieza de los datos, procedemos a realizar la lectura del fichero en formato CSV en el que se encuentran. El resultado devuelto por la llamada a la función read.csv() será un objeto data.frame:

Table 2: train.csv

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

Table 3: test.csv

| PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticke |
|---|---|---|---|---|---|---|---|---|
| 892 | 0 | 3 | Kelly, Mr. James | male | 34,5 | 0 | 0 | 33091 |
| 893 | 1 | 3 | Wilkes, Mrs. James (Ellen Needs) | female | 47,0 | 1 | 0 | 36327 |
| 894 | 0 | 2 | Myles, Mr. Thomas Francis | male | 62,0 | 0 | 0 | 24027 |
| 895 | 0 | 3 | Wirz, Mr. Albert | male | 27,0 | 0 | 0 | 31515 |
| 896 | 1 | 3 | Hirvonen, Mrs. Alexander (Helga E Lindqvist) | female | 22,0 | 1 | 1 | 310129 |
| 897 | 0 | 3 | Svensson, Mr. Johan Cervin | male | 14,0 | 0 | 0 | 753 |

El tipo de datos asignado automáticamente a cada campo es el siguiente:

Table 4: Tipo de dato asignado a cada campo: train data

| | x |
|---|---|
| PassengerId | integer |
| Survived | integer |
| Pclass | integer |
| Name | factor |
| Sex | factor |
| Age | numeric |
| SibSp | integer |
| Parch | integer |
| Ticket | factor |
| Fare | numeric |
| Cabin | factor |
| Embarked | factor |

Table 5: Tipo de dato asignado a cada campo: test data

| | x |
|---|---|
| PassengerId | integer |
| Survived | integer |
| Pclass | integer |
| Name | factor |
| Sex | factor |
| Age | numeric |
| SibSp | integer |
| Parch | integer |
| Ticket | factor |
| Fare | numeric |
| Cabin | factor |
| Embarked | factor |

En primer lugar debemos observar que hay una variable 'PassengerId' que es una variable de tipo Identificador, pero que no aporta nada al estudio desde el conjunto de datos de entrenamiento. Por ello procedemos a eliminarla, directamente:

Table 6: Dataset Titanic sin PassengerId: train data

| Survived | Pclass | Name | Sex | Age | SibSp | Parch | |
|---|---|---|---|---|---|---|---|
| 0 | 3 | Braund, Mr. Owen Harris | male | 22 | 1 | 0 | A/5 |
| 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Thayer) | female | 38 | 1 | 0 | PC |
| 1 | 3 | Heikkinen, Miss. Laina | female | 26 | 0 | 0 | STON/O2. 3 |
| 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35 | 1 | 0 | |
| 0 | 3 | Allen, Mr. William Henry | male | 35 | 0 | 0 | |
| 0 | 3 | Moran, Mr. James | male | NA | 0 | 0 | |

Table 7: Dataset Titanic sin PassengerId: test data

| PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticke |
|---|---|---|---|---|---|---|---|---|
| 892 | 0 | 3 | Kelly, Mr. James | male | 34,5 | 0 | 0 | 33091 |
| 893 | 1 | 3 | Wilkes, Mrs. James (Ellen Needs) | female | 47,0 | 1 | 0 | 36327 |
| 894 | 0 | 2 | Myles, Mr. Thomas Francis | male | 62,0 | 0 | 0 | 24027 |
| 895 | 0 | 3 | Wirz, Mr. Albert | male | 27,0 | 0 | 0 | 31515 |
| 896 | 1 | 3 | Hirvonen, Mrs. Alexander (Helga E Lindqvist) | female | 22,0 | 1 | 1 | 310129 |
| 897 | 0 | 3 | Svensson, Mr. Johan Cervin | male | 14,0 | 0 | 0 | 753 |

En la tabla 'Tipo de dato asignado a cada campo' podemos ver que hay algunas asignaciones de clase que no son correctas. Procedemos a ajustarlas segun el conjunto de datos de cada variable.

Podemos observar que el conjunto de trenes se compone de 891 observaciones con 11 características. Podemos ver primero que algunas características necesitan ser transformado en factores, tales como **Survived** o **Pclass**, por ejemplo, para los que ya KNO que son representantes de los niveles. En la siguiente parte del estudio, vamos a ver cada característica incluida en este conjunto de datos, para mejorar algunas características interesantes, y así obtener un conjunto de entrenamiento final limpiado y mejorado para ser equipado con un modelo.
Veremos qué modificación vale la pena conservar en el análisis y, por lo tanto, la aplicaremos al conjunto de trenes destinados a ser equipados con un modelo de regresión.

### 3.2.1 Variables 'integer' que son de tipo 'factor'

Tenemos basicamente 2 variables, 'Survived' y 'Pclass' que realmente son de tipo factor, ya que los números indican categorias.

```
# ajuste en dataset train
titanic_train$Survived <- as.factor(titanic_train$Survived)
class(titanic_train$Survived)
```

```
## [1] "factor"
```

```
titanic_train$Pclass <- as.factor(titanic_train$Pclass)
class(titanic_train$Pclass)
```

```
## [1] "factor"
```

```
# ajuste en dataset test
titanic_test$Pclass <- as.factor(titanic_test$Pclass)
class(titanic_test$Pclass)
```

```
## [1] "factor"
```

### 3.2.2 Variables 'factor' que son realmente 'character'

En este grupo tenemos a 2 variables: 'Name' y 'Ticket'. Claramente no tiene utilidad su gestión como variables tipo 'factor'.

```
titanic_train$Name <- as.character(titanic_train$Name)
class(titanic_train$Name)
```

```
## [1] "character"
```

```
titanic_train$Ticket <- as.character(titanic_train$Ticket)
class(titanic_train$Ticket)
```

```
## [1] "character"
```

Table 8: Tipo de dato asignado finalmente a las variables seleccionadas

| | x |
|---|---|
| Survived | factor |
| Pclass | factor |
| Name | character |
| Sex | factor |
| Age | numeric |
| SibSp | integer |
| Parch | integer |
| Ticket | character |
| Fare | numeric |
| Cabin | factor |
| Embarked | factor |

## 3.3 Análisis de las variables y limpieza de los datos

En este punto vamos a ir realizando un análisis detenido de cada una de las variables y tomando decisiones en función del resultado de la evaluación de cada variable.

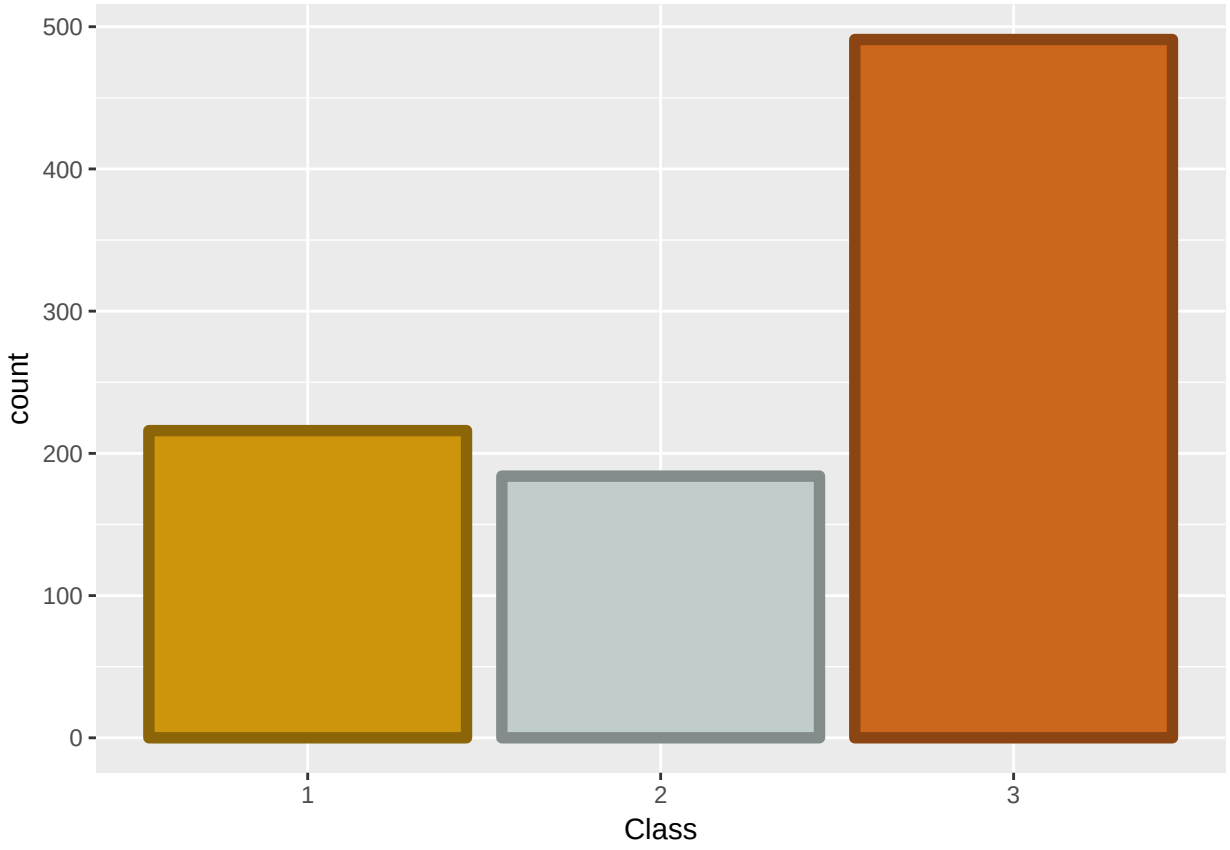
### 3.3.1 Class

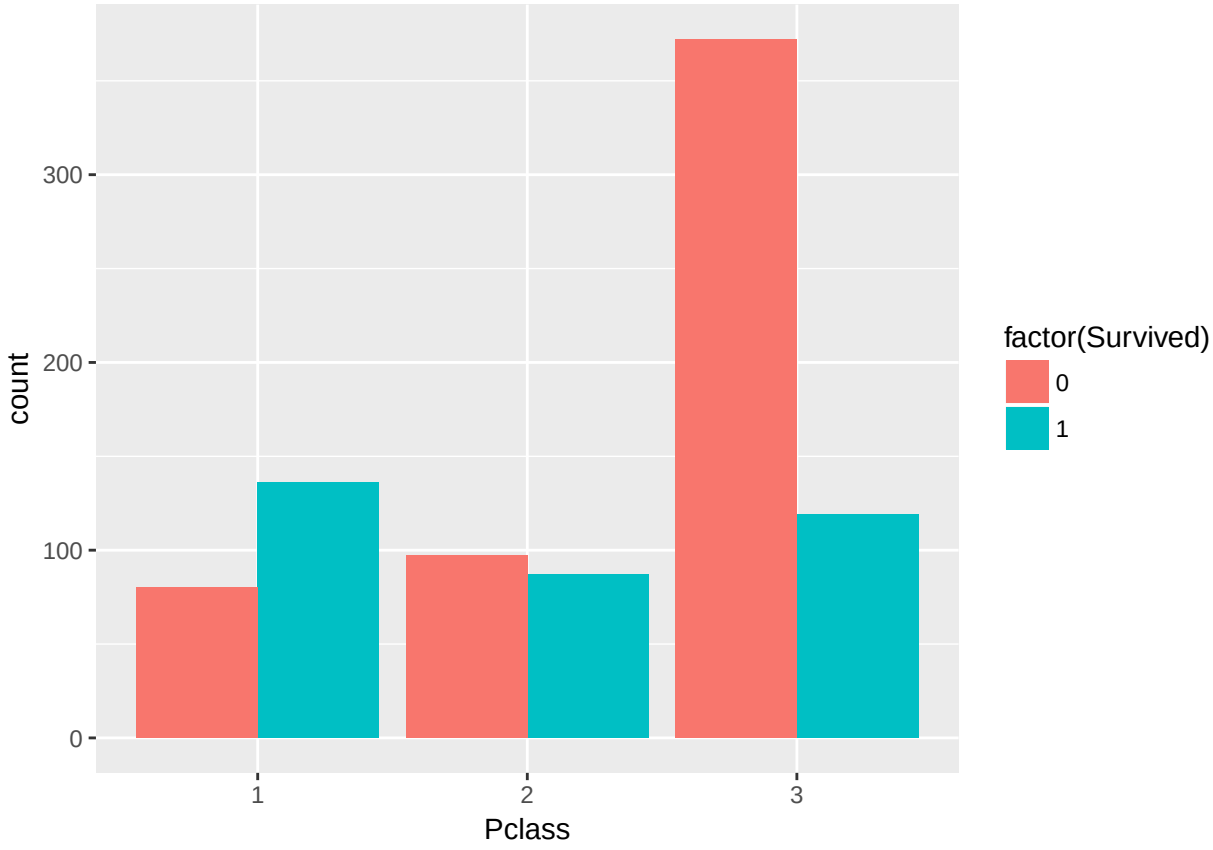
La variable Class tiene 3 niveles distintos. El grupo más grande está en la clase 3 con una división cercana entre las clases 1 y 2, un poco más en la clase 1.

```
titanic_train$Pclass <- as.factor(titanic_train$Pclass)
summary(titanic_train$Pclass)
```

```
##   1   2   3
## 216 184 491
```

La mayoría de la gente está en 3ra clase (entrenador), y aunque puede ser barato, no le va bien. Solo de este gráfico, parece que los pasajeros de tercera clase tienen casi 3 veces más probabilidades de no hacerlo.


500
400
300
200
100
0
count
1
2
3
Class




### 3.3.2 Name

La longitud promedio de un nombre es de 27 caracteres con un máximo de 82.

```
# Resumen de las longitudes del nombre
summary(sapply(as.character(unique(titanic_train$Name)),nchar))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12,00   20,00   25,00   26,97   30,00   82,00
```

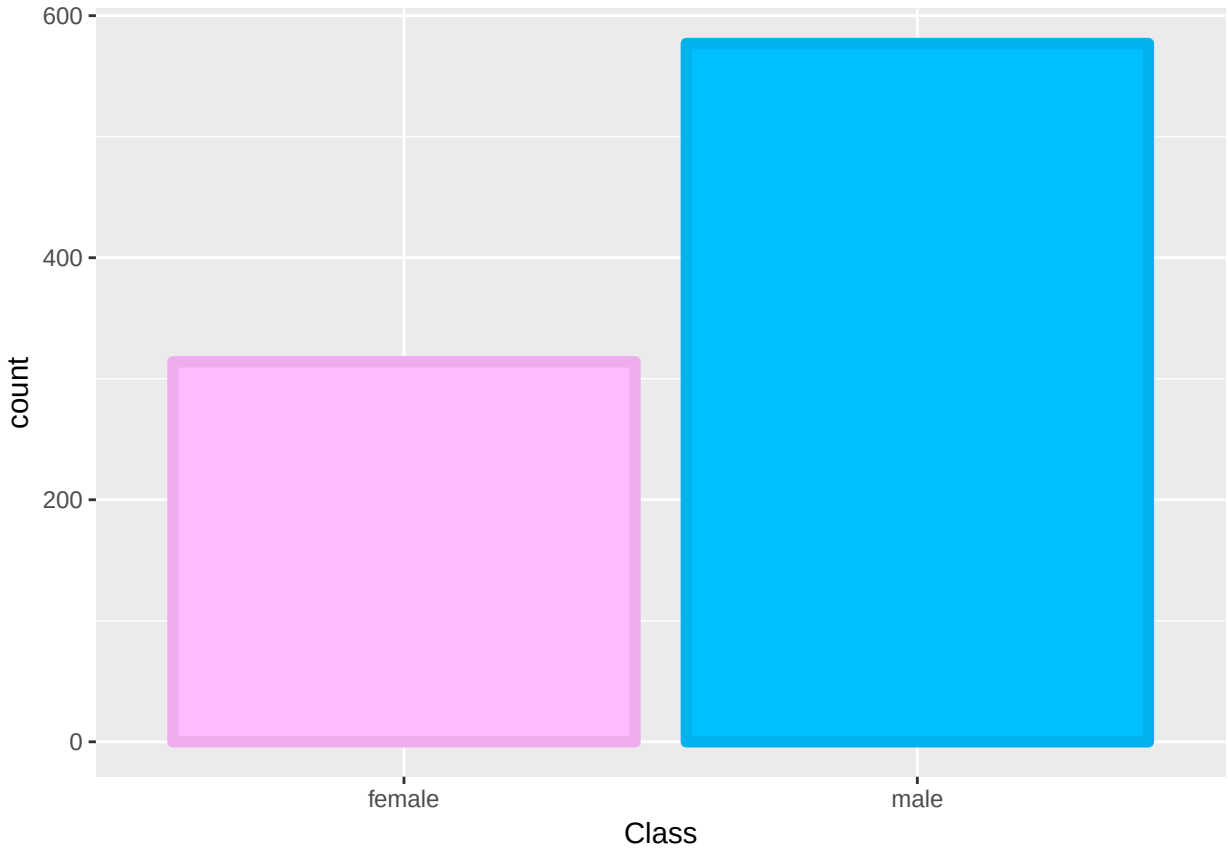
### 3.3.3 Sex

Casi el doble de hombres:

```
#Casi el doble de hombres
summary(titanic_train$Sex)
```

```
## female   male
##    314    577
```

```
sb <- ggplot(titanic_train, aes(x=Sex)) +
  geom_bar(fill=c(colors()[542], colors()[121]),
           col=c(colors()[543], colors()[123]), lwd = 2) +
  labs(x="Class")
sb
```




### 3.3.4 Age

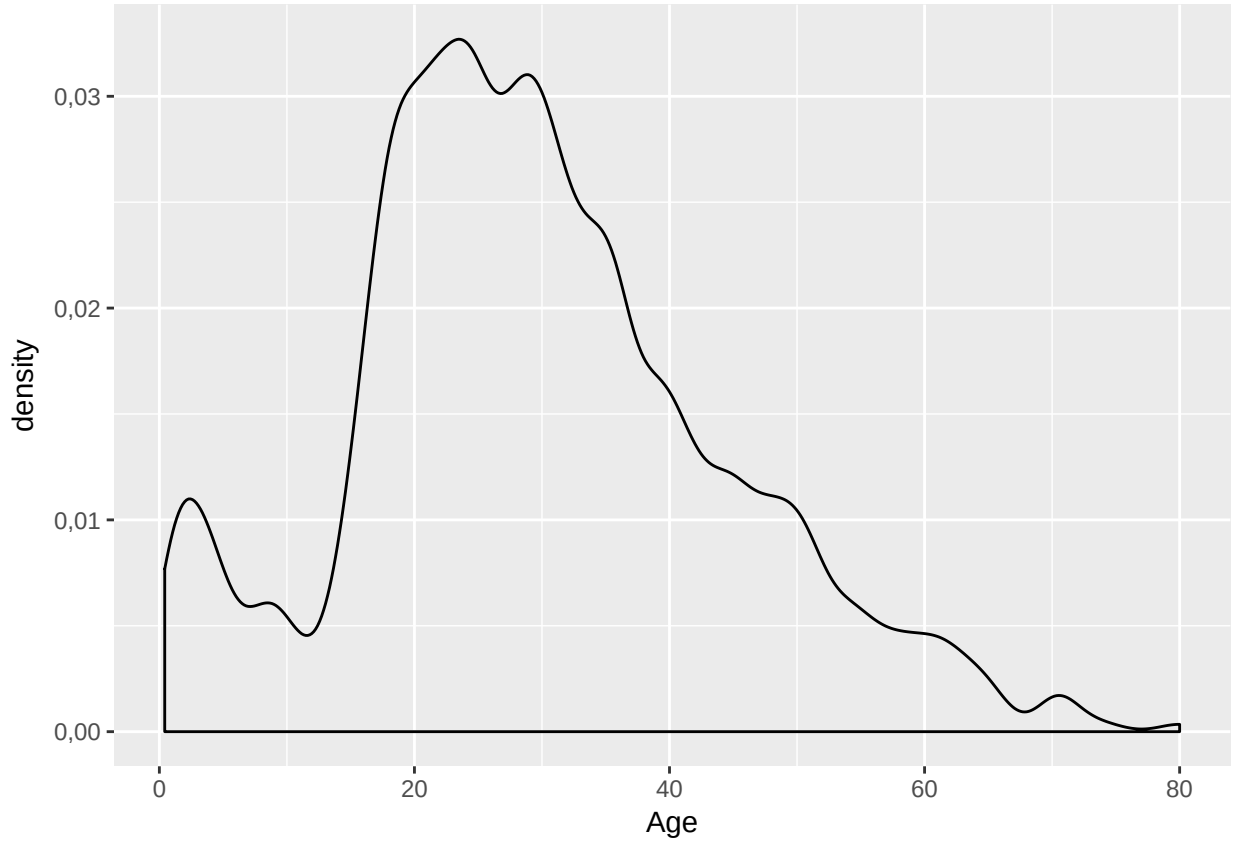
Promedio de edad alrededor de los 30 años con 177 NA. Existe una necesidad de encontrar una forma de imputar estos valores.

```
# # 177 NA's, la edad media es 28 años aka nacido en 1884 (inicio de la entrada)
summary(titanic_train$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0,42   20,12   28,00   29,70   38,00   80,00     177
```

```
ap <- ggplot(titanic_train, aes(x=Age))+geom_density(adjust=.5)
ap
```

```
## Warning: Removed 177 rows containing non-finite values (stat_density).
```




### 3.3.5 Hermanos y cónyuges

La mayoría de las personas viajan solas. La mediana es 0, y solo aproximadamente 1/4 de las personas están con hermanos o cónyuges.

```
unique(titanic_train$SibSp)
```

```
## [1] 1 0 3 4 2 5 8
```

```
# Max Spouse es 1, crea datos de hermanos
# Max parents is 2, create children data
# Recopilar datos de riqueza de los puntos de embarque
titanic_train$SibSp <- as.integer(titanic_train$SibSp) #posiblemente solo frente a variable familiar
summary(titanic_train$SibSp) #mediana es 0
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0,000   0,000   0,000   0,523   1,000   8,000
```

```
dim(titanic_train[titanic_train$SibSp > 0,])
```

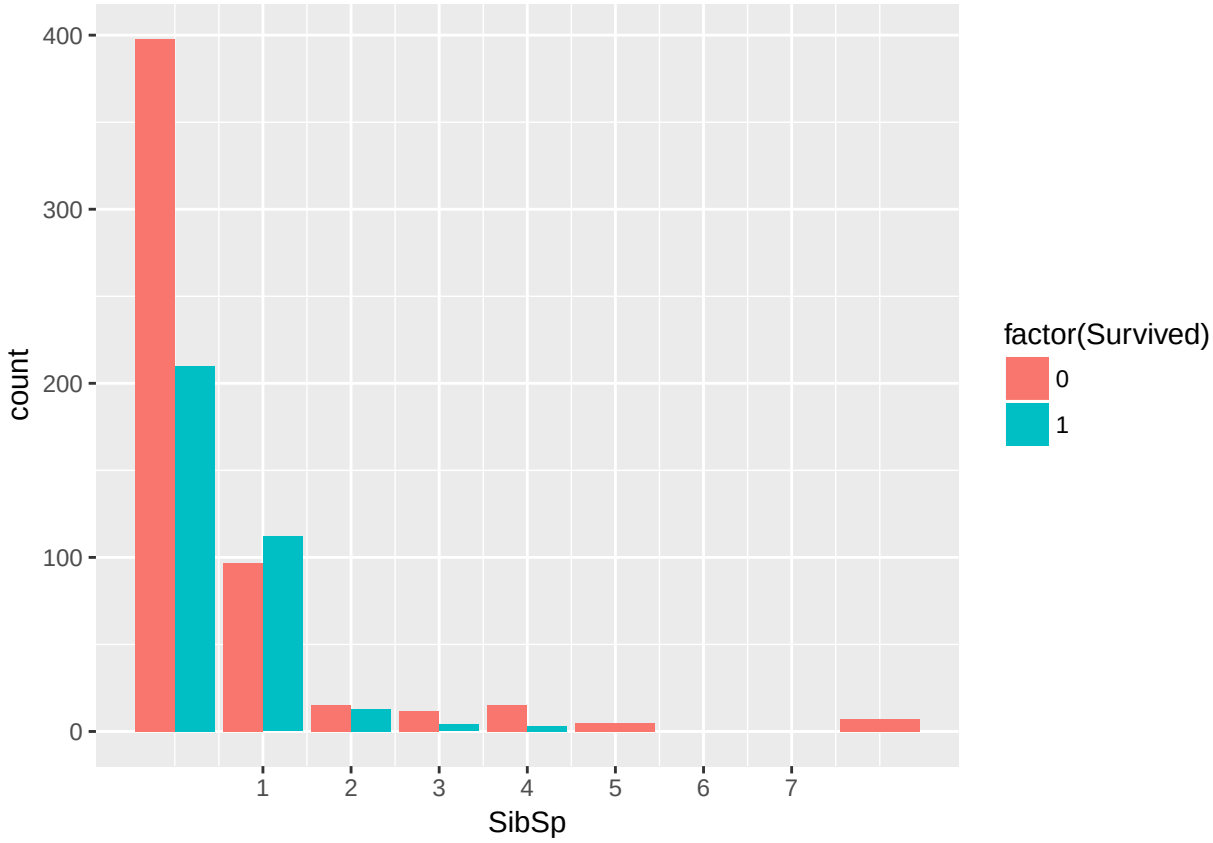
```
## [1] 283  11
```

```
dim(titanic_train[titanic_train$SibSp == 0,])
```

```
## [1] 608  11
```

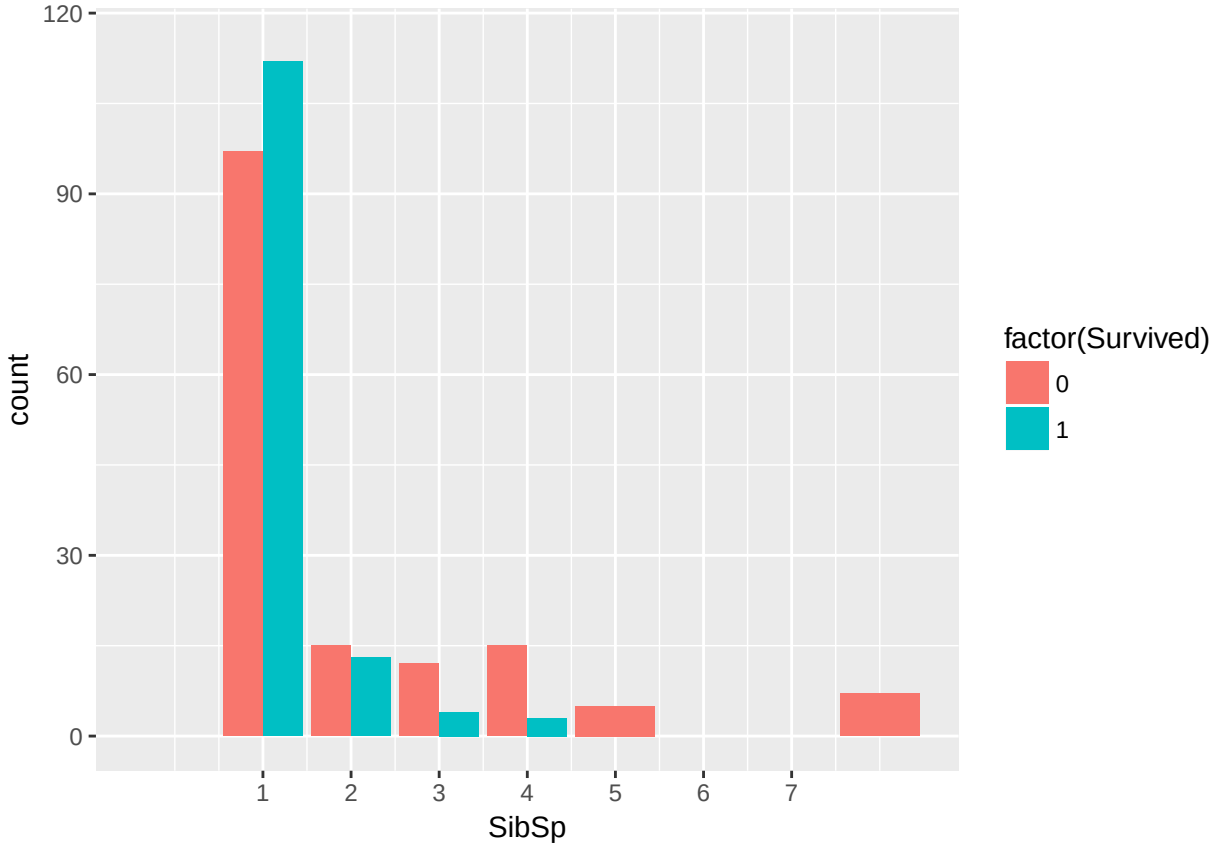
```
ggplot(titanic_train, aes(x = SibSp, fill = factor(Survived))) +
  geom_bar(stat='count', position='dodge') +
```

```
  scale_x_continuous(breaks=c(1:7)) +
  labs(x = 'SibSp')
```




Estas tablas sugieren que la única configuración beneficiosa es tener 1 hermano o cónyuge. Ellos son el único grupo que tiene más probabilidades de sobrevivir. Esto podría ser indicativo de madres solteras, o tal vez de parejas, aunque Jack y Rose son un contador formidable de esa proposición.

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




### 3.3.6 Padres e hijos

Esto también sugiere que la mayoría de la gente no tiene familia. Aproximadamente 1/4 tienen padres o niños a bordo.

```
unique(titanic_train$Parch)
```

```
## [1] 0 1 2 5 3 4 6
```

```
summary(titanic_train$Parch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0,0000  0,0000  0,0000  0,3816  0,0000  6,0000
```

Hay 213 personas que viajan con sus hijos o padres.

```
dim(titanic_train[titanic_train$Parch > 0,])
```

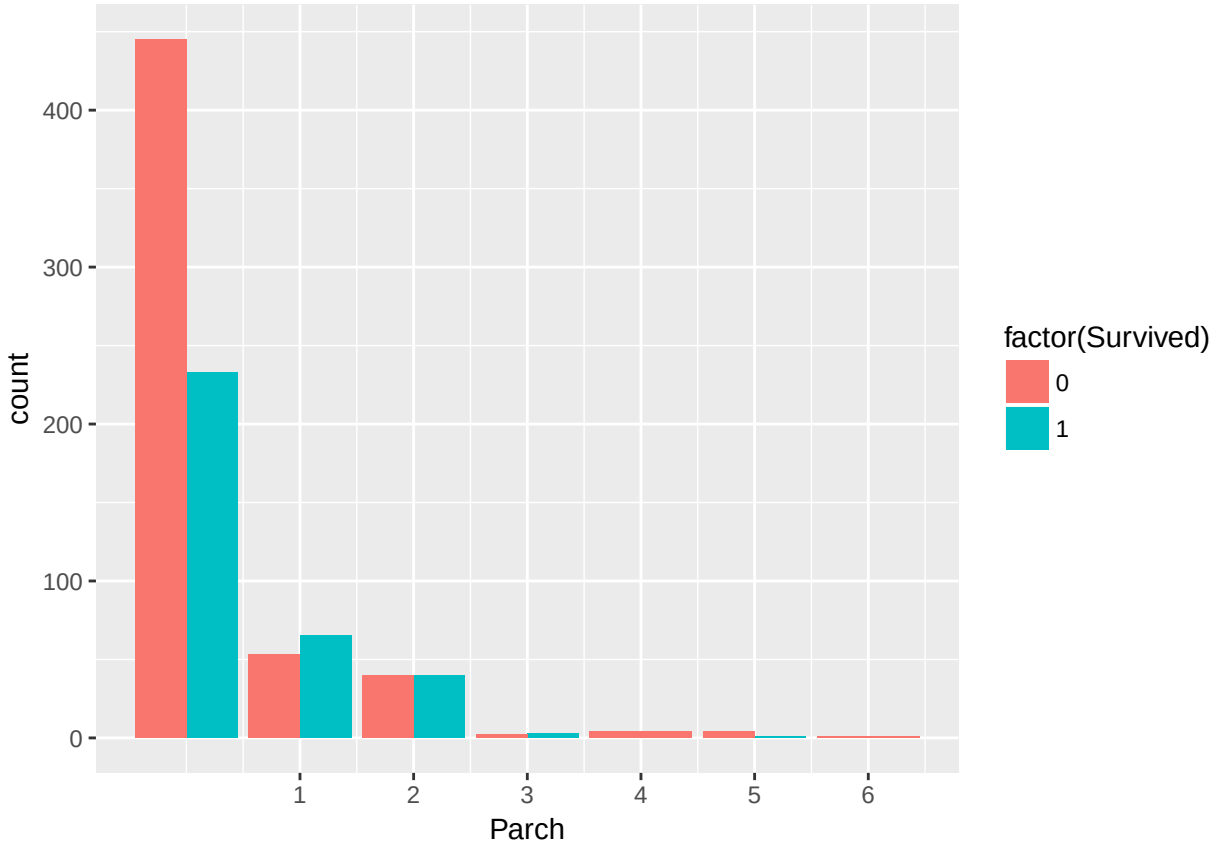
```
## [1] 213  11
```

Los otros 678 están solos.

```
dim(titanic_train[titanic_train$Parch == 0,])
```

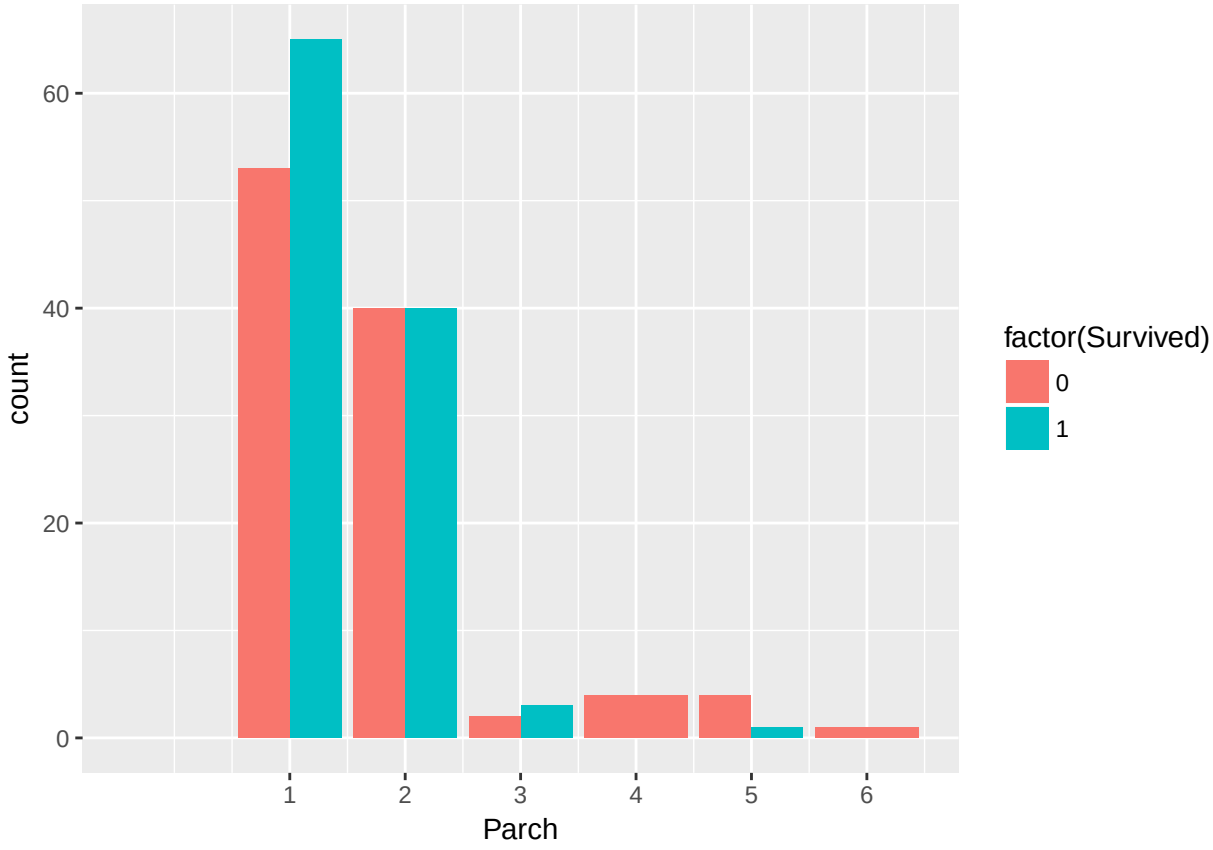
```
## [1] 678  11
```

Este primer cuadro muestra la abrumadora desventaja de viajar solo.




Este muestra que tener una relación 1-3 niños/padres es potencialmente beneficioso.

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

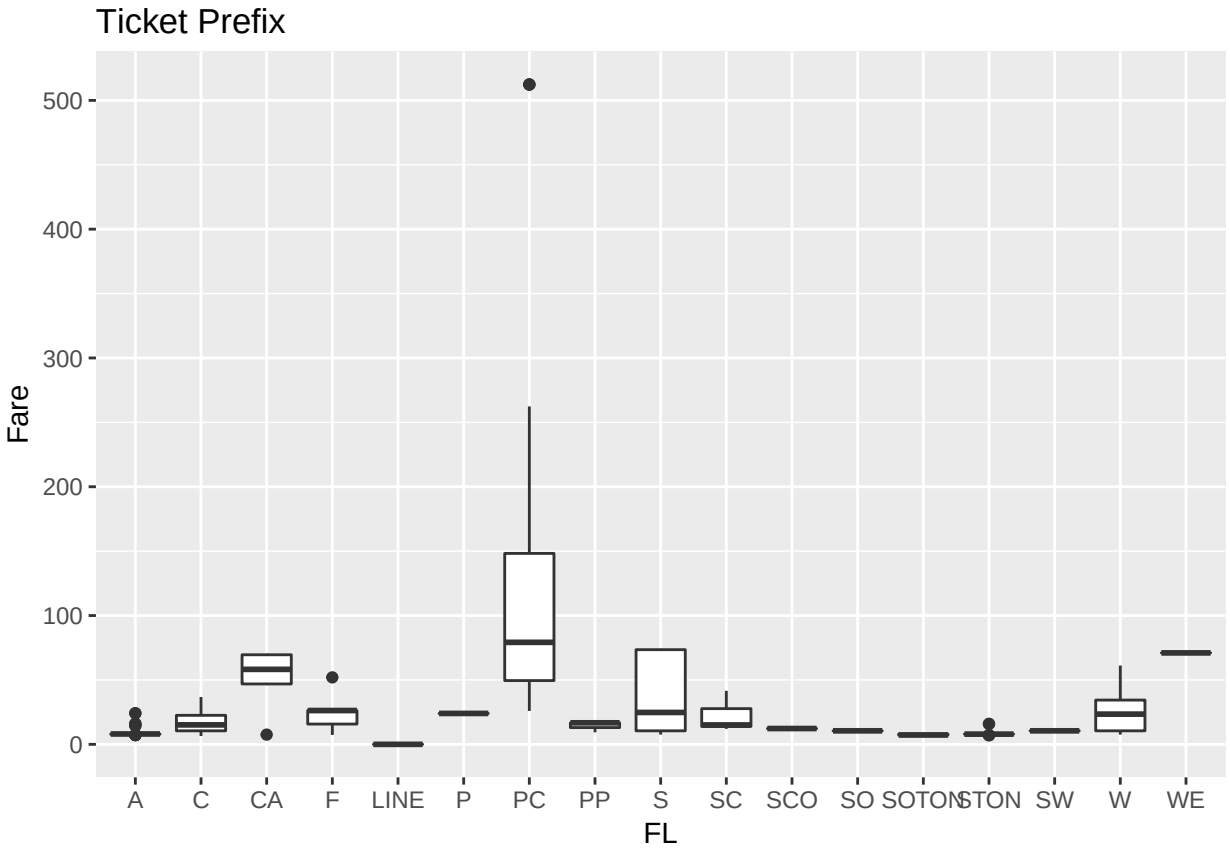



### 3.3.7 Ticket

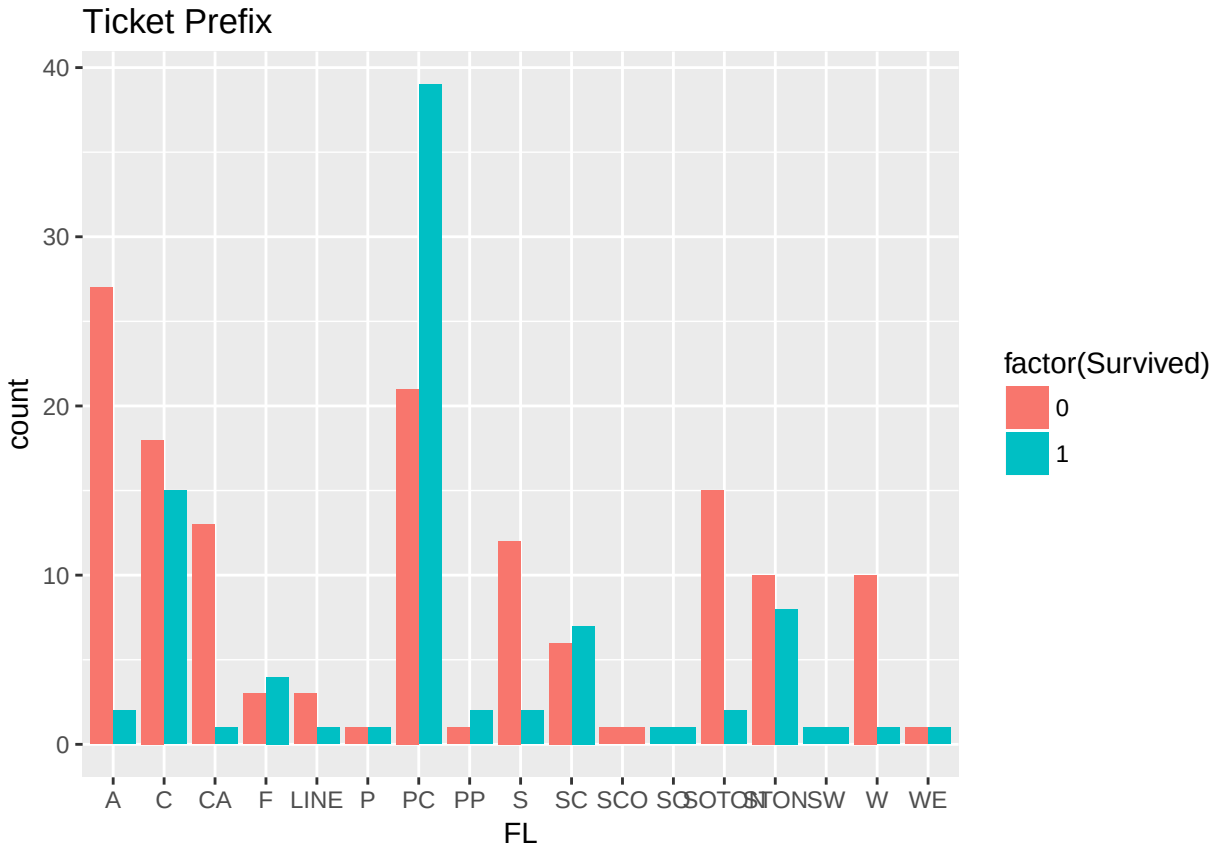
Hay 681 identificadores de tickets únicos. Los boletos tienen identificadores de prefijo y dígitos iniciales limitados que podrían codificar más información sobre el pasajero.

```
## [1] 681
```

Este gráfico muestra que algunos de los prefijos de tickt están asociados con precios más altos. La tarifa ya debería dar cuenta de esta variación, por lo tanto, a menos que podamos encontrar otra relación con el prefijo, entonces no vale la pena incluir esta variable.




Esto confirma que el ticket codifica más información, pero para que sea útil tendrá que cavar más profundo. La gráfica a continuación muestra que aquellos con el prefijo de ticket para PC tienen muchas más probabilidades de sobrevivir. Este gráfico está confundido por el precio del boleto. Algunos de los prefijos son intrigantes. A, CA, SOTON y W tienen tasas de bajas extremadamente altas en relación con todas las demás.




Si profundizamos un poco más, ¿qué tienen en común estos prefijos? Si miramos la longitud de los componentes numéricos de los tickets, encontramos algo interesante: Los sufijos que mencionamos generalmente tienen números de ticket más cortos. El máximo es 2 menos, y la mediana es 2 menos. Esto es particularmente interesante porque hay entradas sin prefijos que también tienen entradas más cortas, así que tal vez tengamos algo bueno.

```
tfix <- tpref[tpref$FL %in% c("A", "CA", "Soton", "W"),]
summary(sapply(as.character(titanic_train$FT),nchar))
```

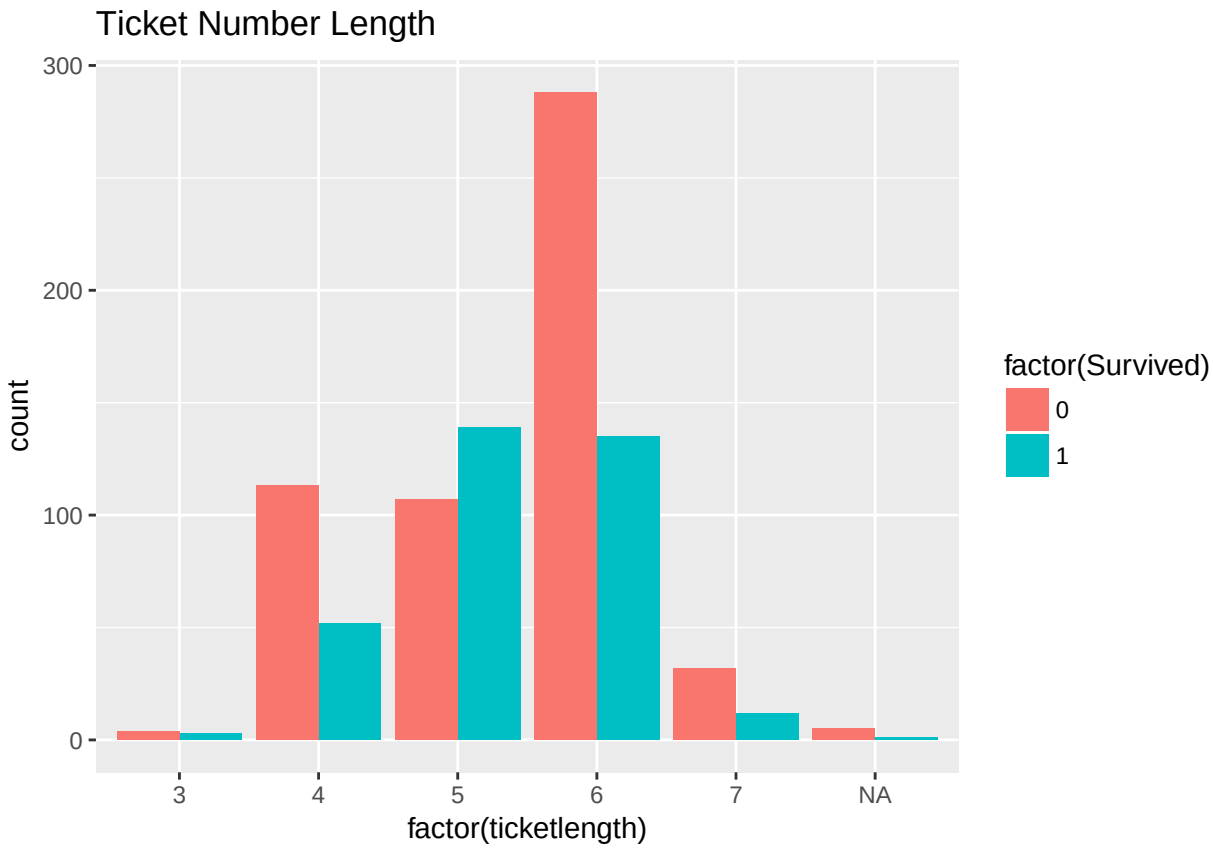
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   3,000   5,000   6,000   5,375   6,000   7,000       6
```

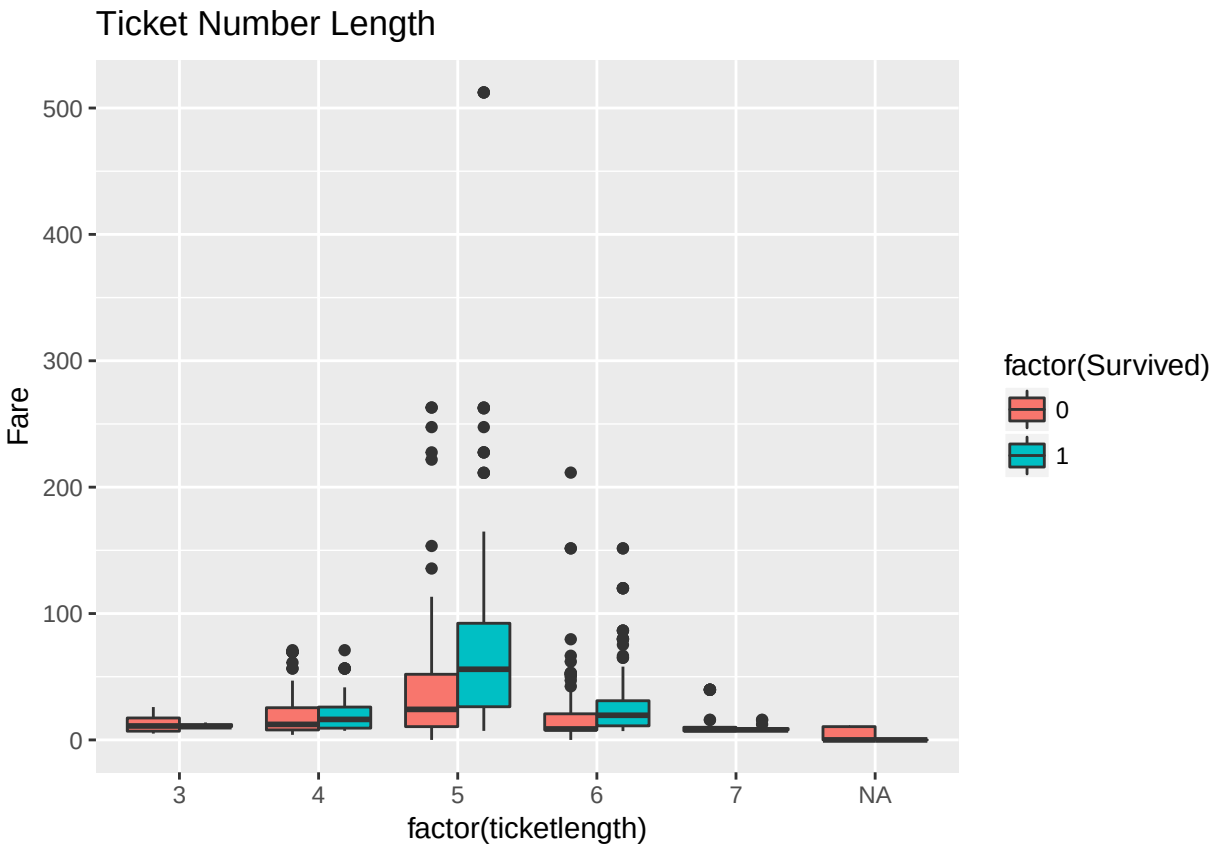
```
summary(sapply(as.character(tfix$FT), nchar))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3,000   4,000   4,000   4,315   5,000   5,000
```

Regresaremos sobre esto más tarde una vez que completemos esta decodificación.

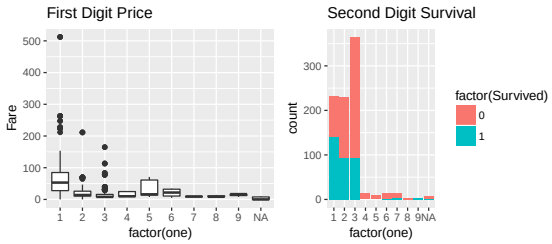
```
titanic_train$ticketlength <- sapply(as.character(titanic_train$FT),nchar)
```


Ticket Number Length
300
200
100
0
count
3
4
5
6
7
NA
factor(ticketlength)
factor(Survived)
0
1

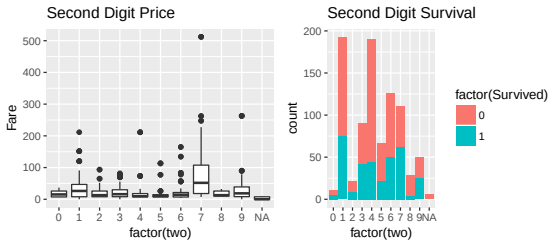



A riesgo de ir un paso demasiado lejos, veamos los números de los boletos individuales. La premisa es que no son aleatorios y potencialmente codifican datos de ubicación de pasajeros que podrían afectar el resultado de la supervivencia. El desglose de los precios se incluye junto para verificar el efecto de confusión.

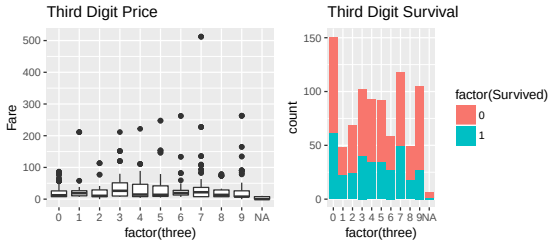
Desde mi punto de vista, parece que hay efectos interesantes sucediendo en los primeros 5 dígitos. La gran noticia es que no tenemos que entenderlo completamente: ahí es donde entra el aprendizaje automático.


First Digit Price
Fare
factor(one)

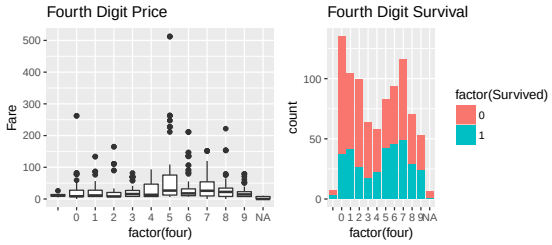

Second Digit Survival
count
factor(one)
factor(Survived)
0
1


Second Digit Price
Fare
factor(two)

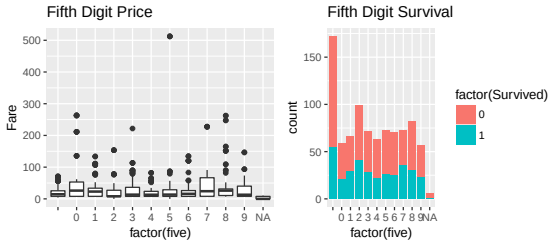

Second Digit Survival
count
factor(two)
factor(Survived)
0
1


Third Digit Price
Fare
factor(three)

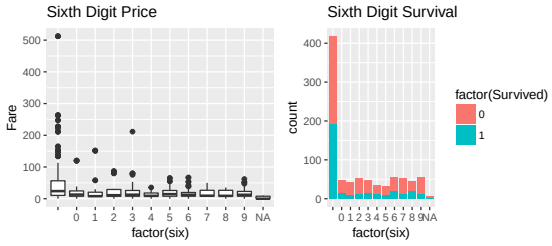

Third Digit Survival
count
factor(three)
factor(Survived)
0
1


Fourth Digit Price
Fare
factor(four)

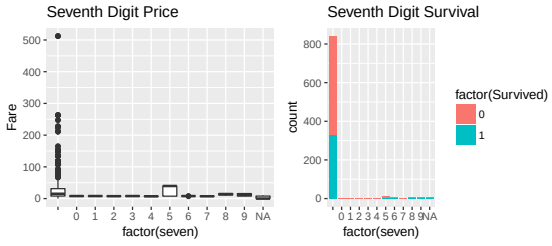

Fourth Digit Survival
count
factor(four)
factor(Survived)
0
1


Fifth Digit Price
Fare
factor(five)


Fifth Digit Survival
count
factor(five)
factor(Survived)
0
1


Sixth Digit Price
Fare
factor(six)


Sixth Digit Survival
count
factor(six)
factor(Survived)
0
1


Seventh Digit Price
Fare
factor(seven)


Seventh Digit Survival
count
factor(seven)
factor(Survived)
0
1

### 3.3.8 Fare

Aparentemente en el Titanic, solo los hombres obtienen viajes gratis. Las mujeres pagan entre 5 y 24 dolares más por boleto, dependiendo de la clase.

```
titanic_train[titanic_train$Fare==0,]$Sex
```

```
##  [1] male male male male male male male male male male male male male male male
## [15] male
## Levels: female male
```

Hay 248 precios únicos de entradas.

```
length(unique(titanic_train$Fare))
```

```
## [1] 248
```

Como se esperaba, la primera clase es mucho más costosa.

```
titanic_train %>% group_by(Pclass) %>% summarise_each(funs(min, max, mean, median),Fare)
```

```
## # A tibble: 3 x 5
##   Pclass Fare_min Fare_max Fare_mean Fare_median
##   <fctr>    <dbl>    <dbl>     <dbl>       <dbl>
## 1 1             0    512        84.2       60.3
## 2 2             0     73.5      20.7       14.2
## 3 3             0     69.6      13.7        8.05
```

Resulta que las mujeres pagan 24 dolares más en promedio por boleto.

```
titanic_train %>% group_by(Sex) %>% summarise_each(funs(min, max, mean, median),Fare)
```

```
## # A tibble: 2 x 5
##   Sex    Fare_min Fare_max Fare_mean Fare_median
##   <fctr>    <dbl>    <dbl>     <dbl>       <dbl>
## 1 female     6.75      512      44.5        23.0
## 2 male       0         512      25.5        10.5
```

### 3.3.9 Cabin

Hay 163 valores de cabina declarados con 148 únicos, faltan muchos. Es posible que la información de la cabina se pueda imputar según el número de boleto, el puerto de embarque y la clase.

```
length(unique(titanic_train$Cabin))
```

```
## [1] 148
```

```
## [1] 204
```

Show 10 entries Search:

| Cabin | Count |
|---|---|
| | 687 |
| B96 B98 | 4 |
| C23 C25 C27 | 4 |
| G6 | 4 |
| C22 C26 | 3 |
| D | 3 |
| E101 | 3 |
| F2 | 3 |
| F33 | 3 |
| B18 | 2 |

Showing 1 to 10 of 100 entries Previous 1 2 3 4 5 … 10 Next

Creamos ahora una variable de la letra de Cabin.

```
titanic_train$CL <- substring(titanic_train$Cabin, 1, 1)
titanic_train$CL <- as.factor(titanic_train$CL)
unique(titanic_train$CL)
```

```
## [1]   C E G D A B F T
## Levels:  A B C D E F G T
```

El subconjunto es desproporcionado para la primera clase en Cherbourg y Southampton.

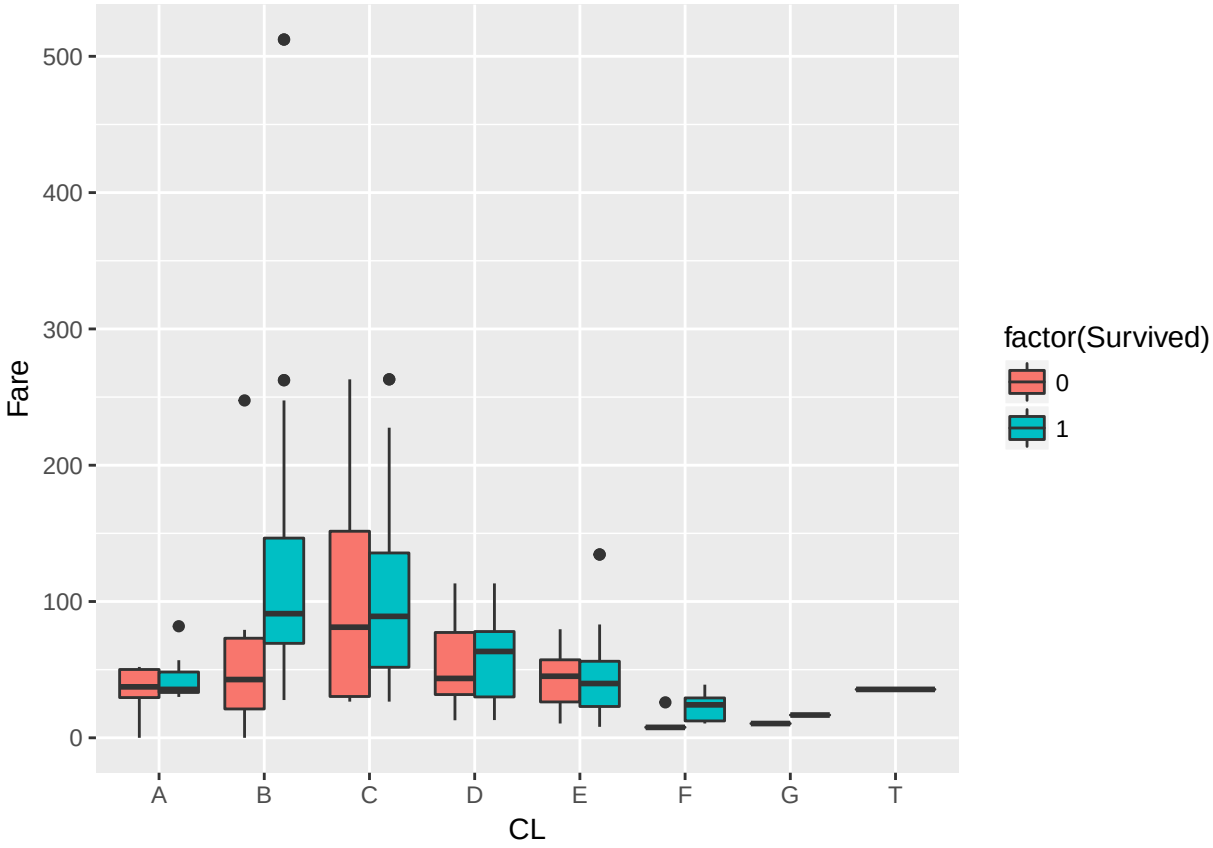
```
summary(titanic_train[!substring(titanic_train$Cabin, 1, 1) == "",]$Embarked)
```

```
##       C   Q   S
##   2  69   4 129
```

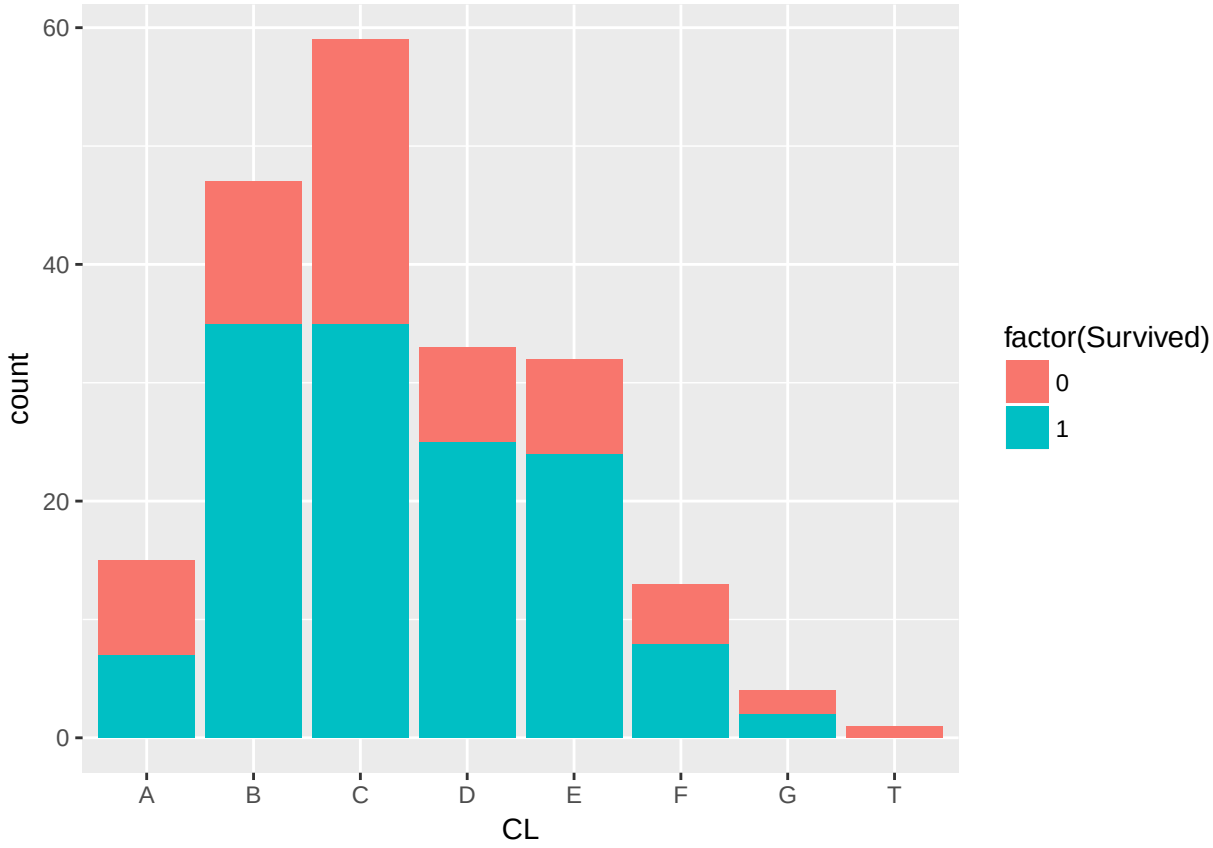
```
summary(titanic_train[!substring(titanic_train$Cabin, 1, 1) == "",]$Pclass)
```

```
##   1   2   3
## 176  16  12
```

El primer gráfico muestra que las cabinas B y C son más caras.




El segundo gráfico muestra que puede haber algunas relaciones interesantes aquí para la supervivencia. Es importante recordar que este subconjunto ya es más probable que sobreviva debido a su estado de primera clase.




Según los datos, parece difícil imputar con precisión las asignaciones de cabina para el resto del conjunto de datos. En cambio, creemos un variable binaria para "habitación asignada" o "no asignada". Esto podría representar una mayor variación en el conjunto de datos inicial. Una de las hipótesis es que las personas que tienen asignada una habitación tienen alguna posición o estatus y tienen prioridad de elección, lo que puede reflejarse en relación con su supervivencia.

```
titanic_train$Assigned <- 0
titanic_train[!substring(titanic_train$Cabin, 1, 1) == "",]$Assigned <- 1
```

### 3.3.10 Embarked

Hay 3 ubicaciones de salida, la mayoría de las personas son de Southampton, Cherburgo y Queenstown.

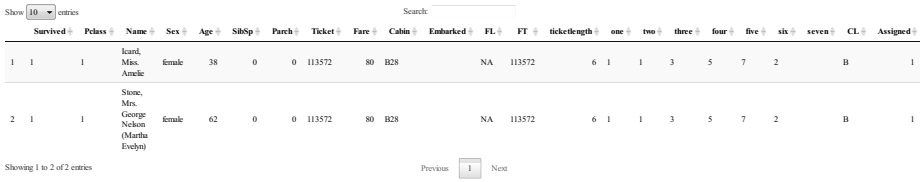
```
unique(titanic_train$Embarked)
```

```
## [1] S C Q
## Levels:  C Q S
```

```
summary(titanic_train$Embarked)
```

```
##       C   Q   S
##   2 168  77 644
```

Hay 2 valores faltantes para el punto de embarque. Lo que nos lleva a la imputación.



Show 10 entries Search:

| | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked | FL | FT | ticketlength | one | two | three | four | five | six | seven | CL | Assigned |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | Icard, Miss. Amelie | female | 38 | 0 | 0 | 113572 | 80 | B28 | | NA | 113572 | 6 | 1 | 1 | 3 | 5 | 7 | 2 | | B | 1 |
| 2 | 1 | 1 | Stone, Mrs. George Nelson (Martha Evelyn) | female | 62 | 0 | 0 | 113572 | 80 | B28 | | NA | 113572 | 6 | 1 | 1 | 3 | 5 | 7 | 2 | | B | 1 |

Showing 1 to 2 of 2 entries Previous 1 Next

### 3.3.11 Ajuste de los datos con ceros o valores nulos

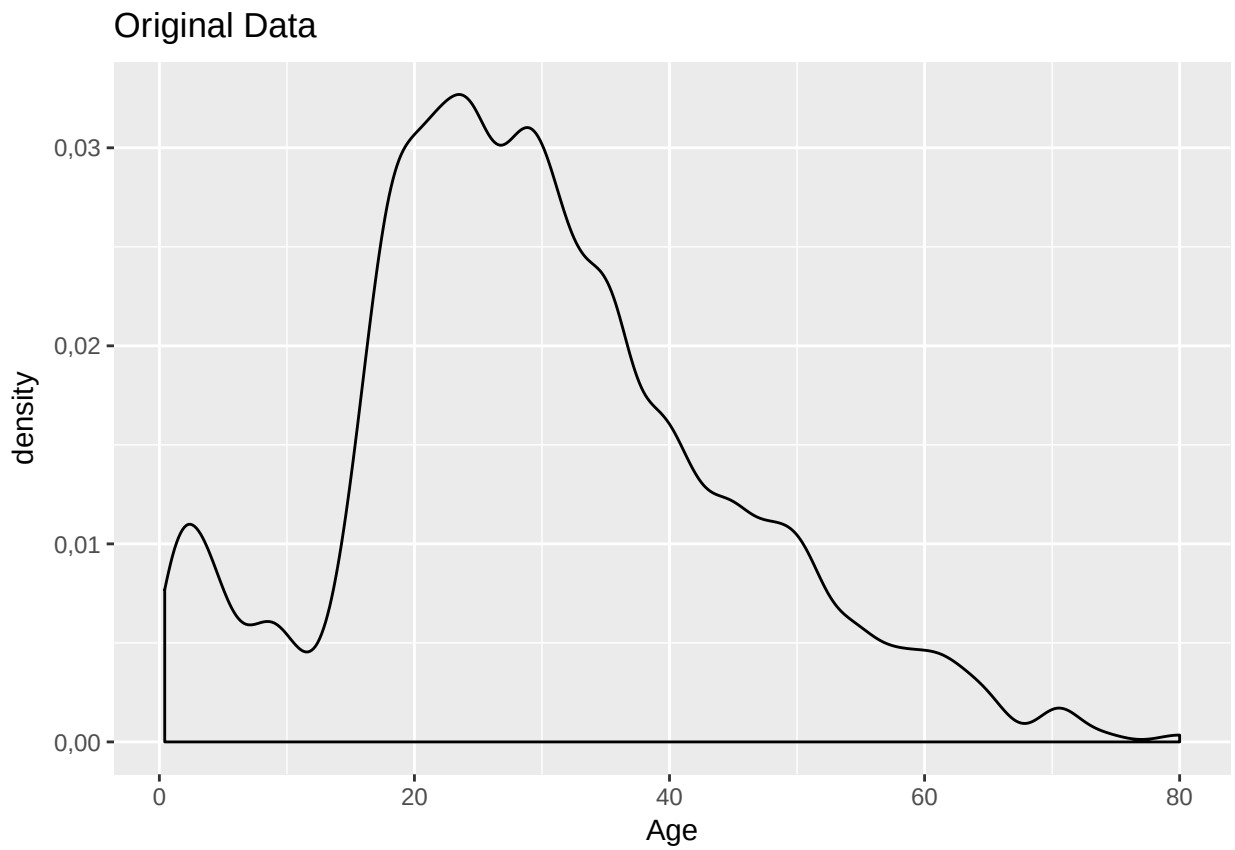
Los únicos valores faltantes se encuentran en las columnas Cabin, Age y Embarked.

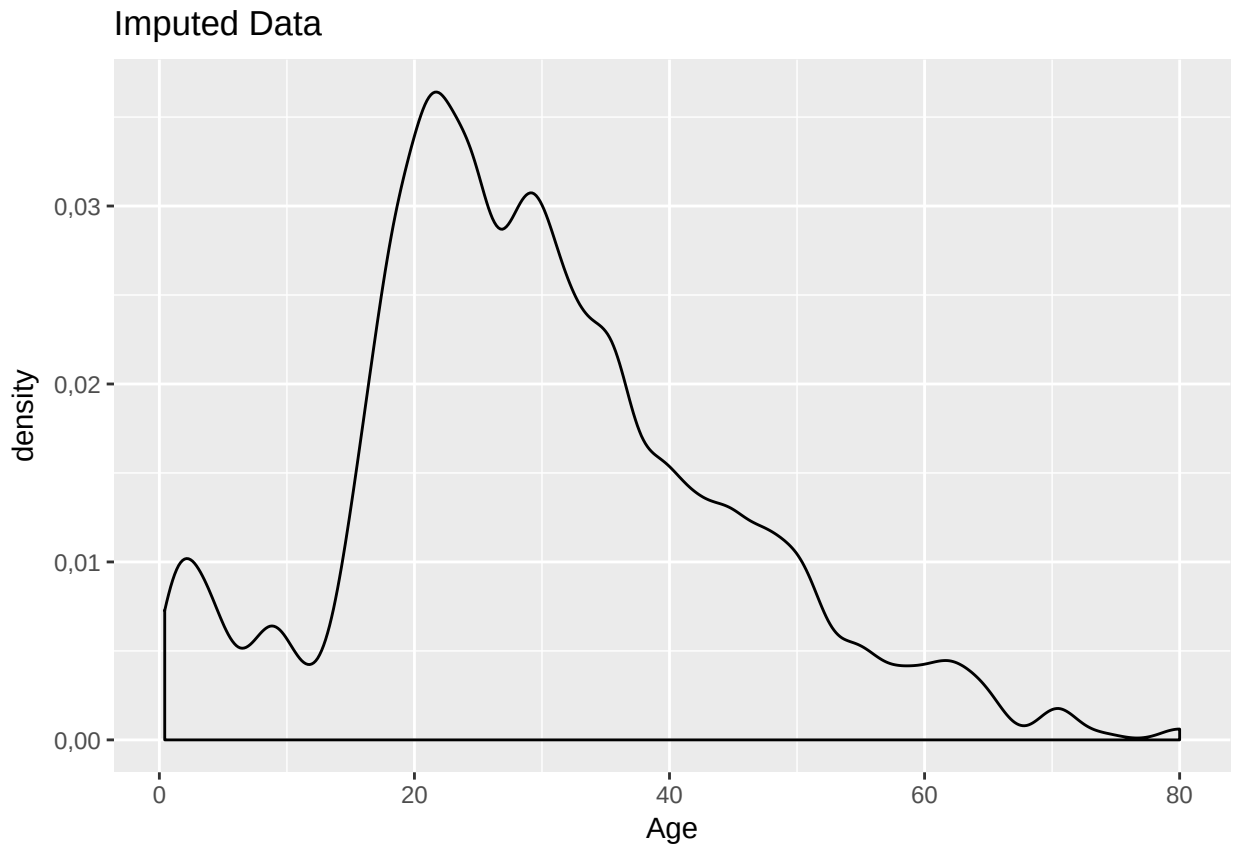
```
##      Survived       Pclass         Name          Sex          Age
##             0            0            0            0           NA
##        SibSp        Parch       Ticket         Fare        Cabin
##             0            0            0            0          687
##     Embarked           FL           FT ticketlength          one
##             2            0           NA           NA           NA
##          two        three         four         five          six
##           NA           NA           NA           NA           NA
##        seven           CL     Assigned
##           NA          687            0
```

```
##      Survived       Pclass         Name          Sex          Age
##             0            0            0            0          177
##        SibSp        Parch       Ticket         Fare        Cabin
##             0            0            0            0            0
##     Embarked           FL           FT ticketlength          one
##             0            0            6            6            6
##          two        three         four         five          six
##            6            6            6            6            6
##        seven           CL     Assigned
##            6            0            0
```

Ajustamos los datos faltantes en la variable Age:

```
titanic_train <- as.data.frame(titanic_train)
#imputar los valores de edad faltantes con el paquete MICE
impute <- mice(titanic_train[, !names(titanic_train) %in% c('PassengerId','Name','Ticket','Cabin','Surv

trained_mouse <- complete(impute)
```


Original Data
0,03
0,02
0,01
0,00
density
0
20
40
60
80
Age




Dado que los resultados están razonablemente bien emparejados, podemos reemplazar la columna original con los valores imputados.

```
titanic_train$Age <- trained_mouse$Age
```

**Ahora vamos a analizar y ajustar los valores en la variable Enbarked.**

Ahora estamos en esos 2 valores de la variable Embarked que faltan. Lo primero que viene a la mente es verificar con el valor de la variable Cabin. A partir de los datos, todas las cabinas que comienzan en B se embarcaron desde Southampton o Charbourg.

```
unique(titanic_train[grep("*^B", titanic_train$Cabin),]$Embarked)
```

```
## [1] C   S
## Levels:  C Q S
```

Los billetes de viaje cuestan 80 USD, que es muy similar a la tarifa promedio de los pasajeros S en cabinas tipo B.

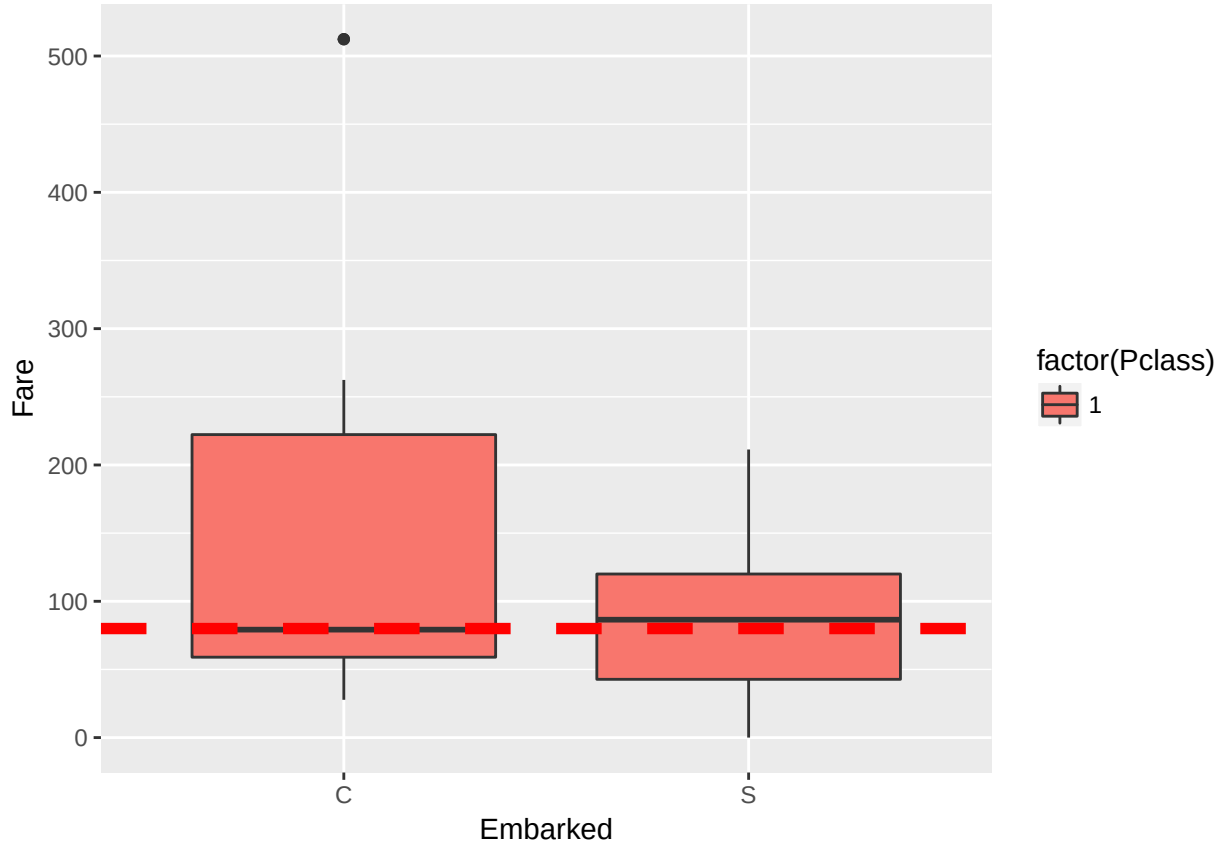
```
## # A tibble: 3 x 2
##   Embarked  Fare
##   <fctr>   <dbl>
## 1 ""        80.0
## 2 C        146
## 3 S         85.4
```

Sin embargo, la tarifa es más cercana a la mediana de la variable Fare de pasajeros tipo C.

```
## # A tibble: 3 x 2
##   Embarked  Fare
```

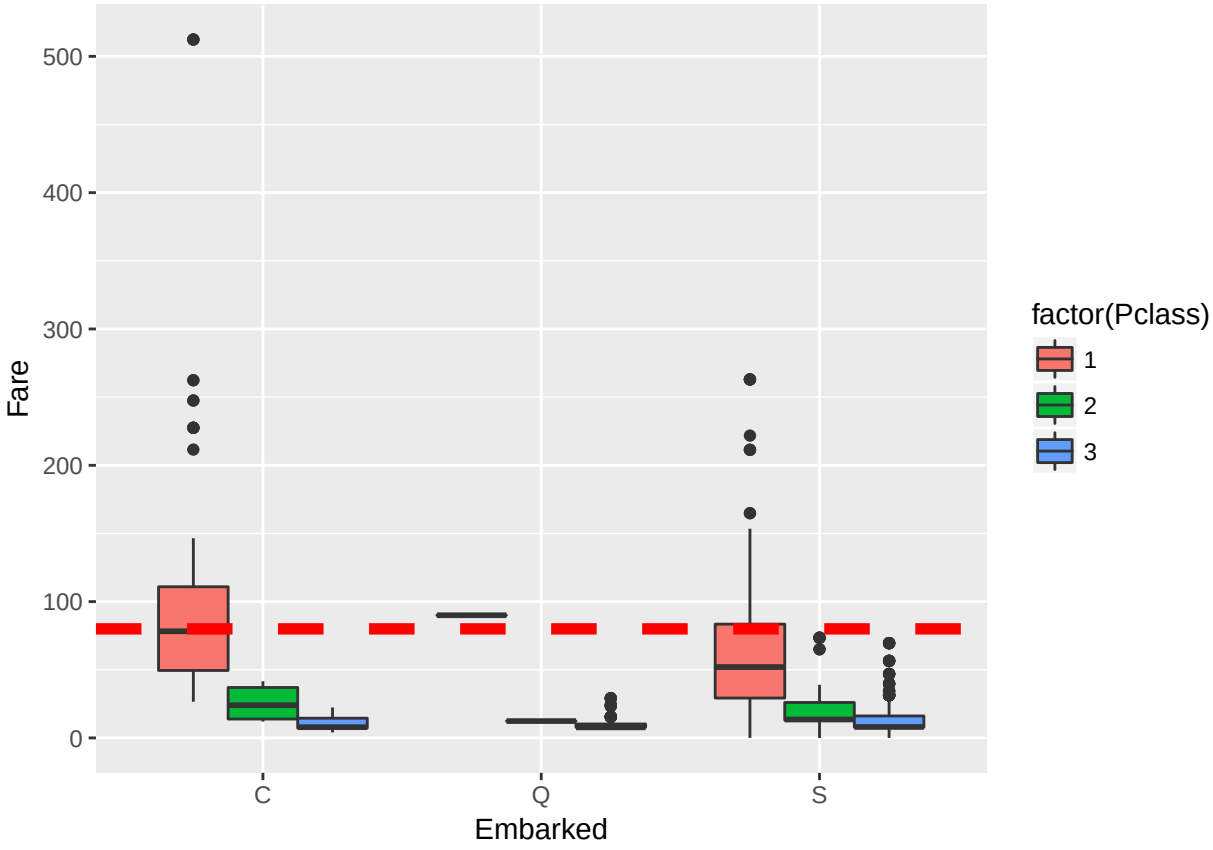
```
##   <fctr>   <dbl>
## 1 ""        80.0
## 2 C         79.2
## 3 S         86.5
```

Entonces, lo que parece ser un inocente valor perdido aislado es en realidad una pregunta interesante sobre la imputación basada en la media o la mediana.




Puede parecer razonable imputar por la mediana de las tarifas de C porque ese es el valor más cercano. Sin embargo, si miramos hacia atrás en la sección de tarifas, vemos que hay un 72,4% de posibilidades de que un pasajero determinado se embarque desde S.

```
##       C   Q   S
##   2 168  77 644
```




```
##     C  Q  S
##  2 22  0 23
```

Al final, decidí ampliar 'S' a la variable y ver dónde eso nos conduce en el análisis posterior.

```
titanic_train$Embarked[c(62, 830)] <- 'S'
```

# 4 Análisis de las variables del dataset titanic

Para los fines de este estudio, trabajamos con solo cuatro variables de entrada y una variable de respuesta. Como se mencionó anteriormente, las cuatro variables de entrada son Class, Sex, Age y Puerto de Embarque. La variable de respuesta es si sobrevivieron o no.

Podemos recortar los datos según nuestras necesidades usando los siguientes comandos:

```
df = titanic_train[,c(1,2,4,5,11)]
```

También hacemos que Age sea una variable categórica de la siguiente manera:

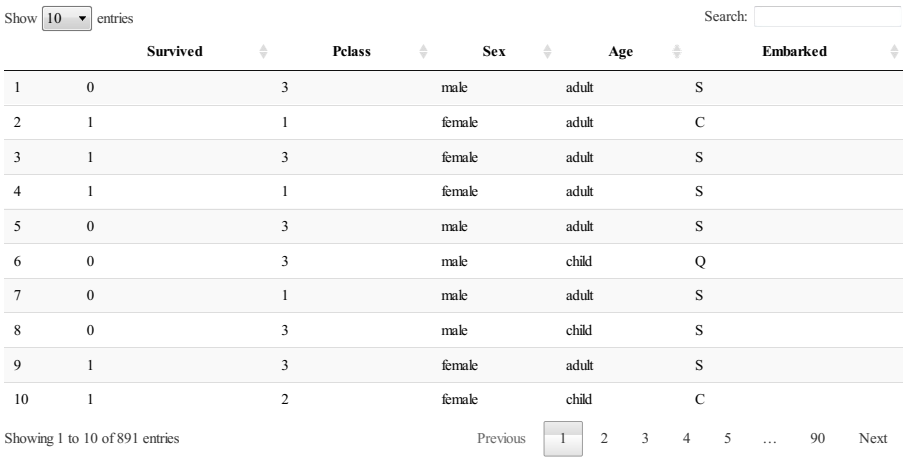
- Si age <= 18, entonces age = child
- Si 18 < age <= 60, entonces age = adult
- Si age > 60, entonces age = senior

```
df$Age[df$Age <= 18] = "child"
df$Age[(df$Age > 18) & (df$Age <= 60) & (df$Age != "child")] = "adult"
df$Age[(df$Age != "child") & (df$Age != "adult")] = "senior"
df$Age = as.factor(df$Age)
```

Después de realizar esta operación, nuestros datos se ven así:

Table 9: Dataset seleccionado

| Survived | Pclass | Sex | Age | Embarked |
|---|---|---|---|---|
| 0 | 3 | male | adult | S |
| 1 | 1 | female | adult | C |
| 1 | 3 | female | adult | S |
| 1 | 1 | female | adult | S |
| 0 | 3 | male | adult | S |
| 0 | 3 | male | child | Q |


Show 10 entries Search:

| | Survived | Pclass | Sex | Age | Embarked |
|---|---|---|---|---|---|
| 1 | 0 | 3 | male | adult | S |
| 2 | 1 | 1 | female | adult | C |
| 3 | 1 | 3 | female | adult | S |
| 4 | 1 | 1 | female | adult | S |
| 5 | 0 | 3 | male | adult | S |
| 6 | 0 | 3 | male | child | Q |
| 7 | 0 | 1 | male | adult | S |
| 8 | 0 | 3 | male | child | S |
| 9 | 1 | 3 | female | adult | S |
| 10 | 1 | 2 | female | child | C |

Showing 1 to 10 of 891 entries Previous 1 2 3 4 5 … 90 Next

```
summary(df)
```

```
##  Survived Pclass      Sex          Age      Embarked
##  0:549    1:216   female:314   adult :698    :  0
##  1:342    2:184   male  :577   child :164   C:168
##           3:491                senior: 29   Q: 77
##                                             S:646
```

## 4.1 Diseño experimental

El análisis de datos está organizado de la siguiente manera:

- Computing efectos principales para los cuatro factores
- Computing efectos de interacción para los seis pares de factores Análisis computarizado de la varianza (ANOVA) para los cuatro efectos principales y los seis efectos de interacción.

### 4.1.1 Justificación del diseño

La justificación racional detrás de este diseño es la siguiente. Claramente, tenemos un conjunto de datos a mano que se compone de cuatro variables de entrada, tres de las cuales son categóricas y la cuarta es numérica discreta. La variable de respuesta también es categórica.

Lo primero que se me ocurre es analizar individualmente el efecto de cada una de las variables de entrada en la variable de respuesta. Esto no es más que calcular los efectos principales.

El segundo nivel de investigación que se me ocurre es comprobar si un par de dos factores tiene un efecto sinérgico en la variable de respuesta que parece ser más que el efecto combinado de los dos factores. Esto no es más que computar los efectos de interacción.

En tercer lugar, desde un punto de vista puramente estadístico, estamos tratando con muestras de cuatro variables aleatorias como entradas. Si solo tuviéramos muestras de dos variables aleatorias, habríamos optado por una prueba z o una prueba t. Sin embargo, dado que hay más de dos variables aleatorias, se prefiere ANOVA, ya que hace exactamente eso.

Las siguientes subsecciones arrojan luz sobre los temas de aleatorización, Replicación y Medidas repetidas, y Bloqueo desde un punto de vista puramente teórico, así como desde el punto de vista de este estudio.

### 4.1.2 Aleatorización

La aleatorización se realiza para permitir la mayor fiabilidad y validez de las estimaciones estadísticas de los efectos del tratamiento. Más precisamente, se refiere a asignar aleatoriamente las unidades experimentales a través de los grupos de tratamiento. La aleatorización reduce el sesgo al minimizar el efecto de los factores de molestia o el ruido estocástico en los datos.

En este estudio, los pasajeros del Titanic se dividieron en tres clases separadas, determinadas por su precio de boleto y su riqueza y clase social. Los pasajeros en primera clase eran los más ricos del lote e incluían personas prominentes de clase alta, hombres de negocios, políticos, personal militar de alto rango, industriales, banqueros, etc. Los pasajeros de segunda clase incluían profesores, autores, clérigos, turistas, etc. Pasajeros en tercera clase fueron emigrantes que se mudaron a los Estados Unidos y Canadá. Además, los pasajeros fueron variados en etnia también. Hubo pasajeros del Imperio Otomano, otros tenían orígenes árabes, etc. Por lo tanto, para los fines de este estudio, hubo una buena cantidad de aleatorización de los pasajeros en función de su nacionalidad, etnia, condición económica, condición social, riqueza, etc.

### 4.1.3 Replicación y / o medidas repetidas

La replicación se refiere a la repetición de un experimento para reducir la variabilidad asociada con el fenómeno que se estudia. En cualquier proceso de muestreo, la variación que es inherente no se puede eliminar, pero lo mejor que se puede hacer es eliminar la variación causada por causas especiales. Esto es lo que se logra mediante la Replicación.

Hay una línea delgada que separa la replicación de medidas repetidas. La medición repetida se refiere al uso de los mismos sujetos dentro de cada rama del estudio, incluido el grupo de control.

En este estudio, los datos disponibles están escritos en piedra. Realizar la replicación significaría hacer que el barco del Titanic navegue varias veces y recopilar los datos en todos esos viajes: algunos de los cuales pueden hundirse y otros no. Realizar medidas repetidas significaría hacer que la nave del Titanic navegue en un número de universos paralelos con el mismo grupo de pasajeros. Claramente, ambos no son prácticos y, por lo tanto, en nuestro estudio, no hay evidencia de replicación o medidas repetidas.

### 4.1.4 Bloqueo

El bloqueo se refiere a organizar unidades experimentales en grupos (bloques) que son similares entre sí de alguna manera, forma o forma. Estos bloques se analizan juntos para reducir la variabilidad conocida. La idea principal detrás del bloqueo es que una variabilidad que ocurre en cualquiera de las variables de entrada que no se pueden superar se confunde con una interacción para eliminar su influencia en la variable de respuesta.

La base teórica para el bloqueo se puede entender a partir de la siguiente ecuación. Dadas dos variables aleatorias X e Y,

Var (X-Y) = Var (X) + Var (Y) -2Cov (X, Y)

La varianza de la diferencia se puede minimizar maximizando la covarianza (o la correlación) entre X e Y.

En este estudio, hemos analizado los datos como un todo porque no había ninguna razón para sospechar la existencia de variabilidad conocida para ninguna de las variables de entrada consideradas.

## 4.2 Análisis estadístico

### 4.2.1 Análisis exploratorio de datos

Resumen de los datos limpios:

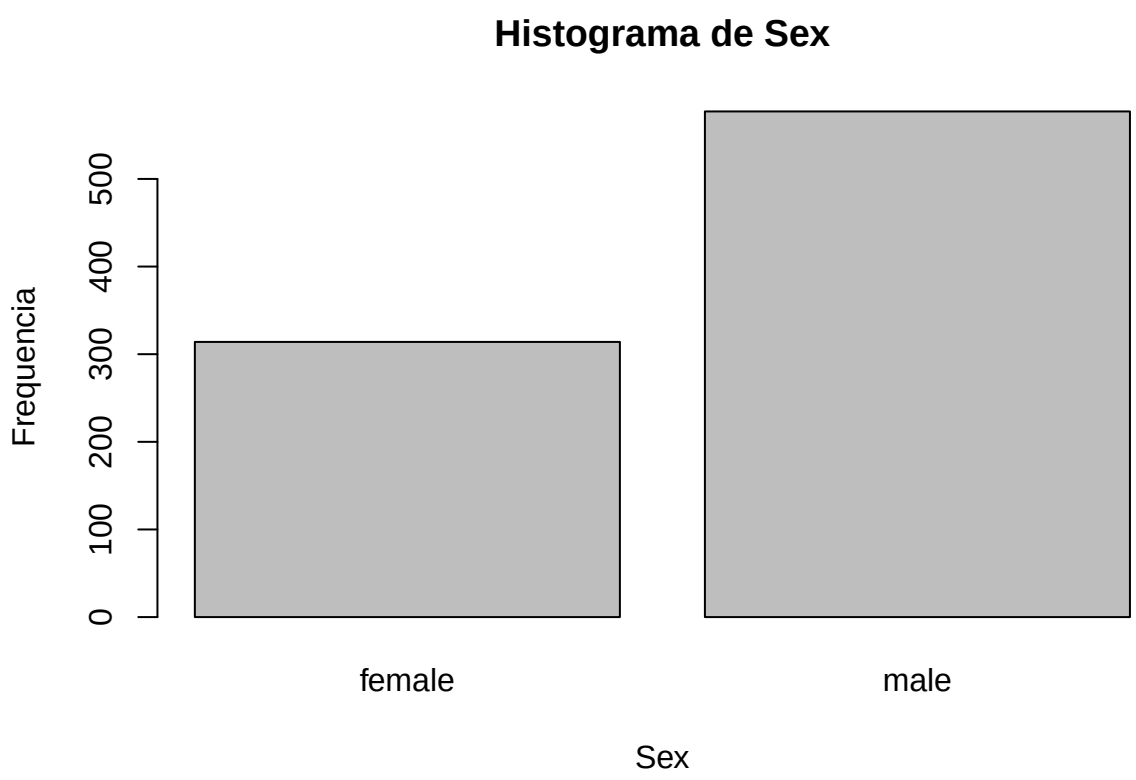
```
summary(df)
```

```
##  Survived Pclass      Sex          Age      Embarked
##  0:549    1:216   female:314   adult :698    :  0
##  1:342    2:184   male  :577   child :164   C:168
##           3:491                senior: 29   Q: 77
##                                             S:646
```

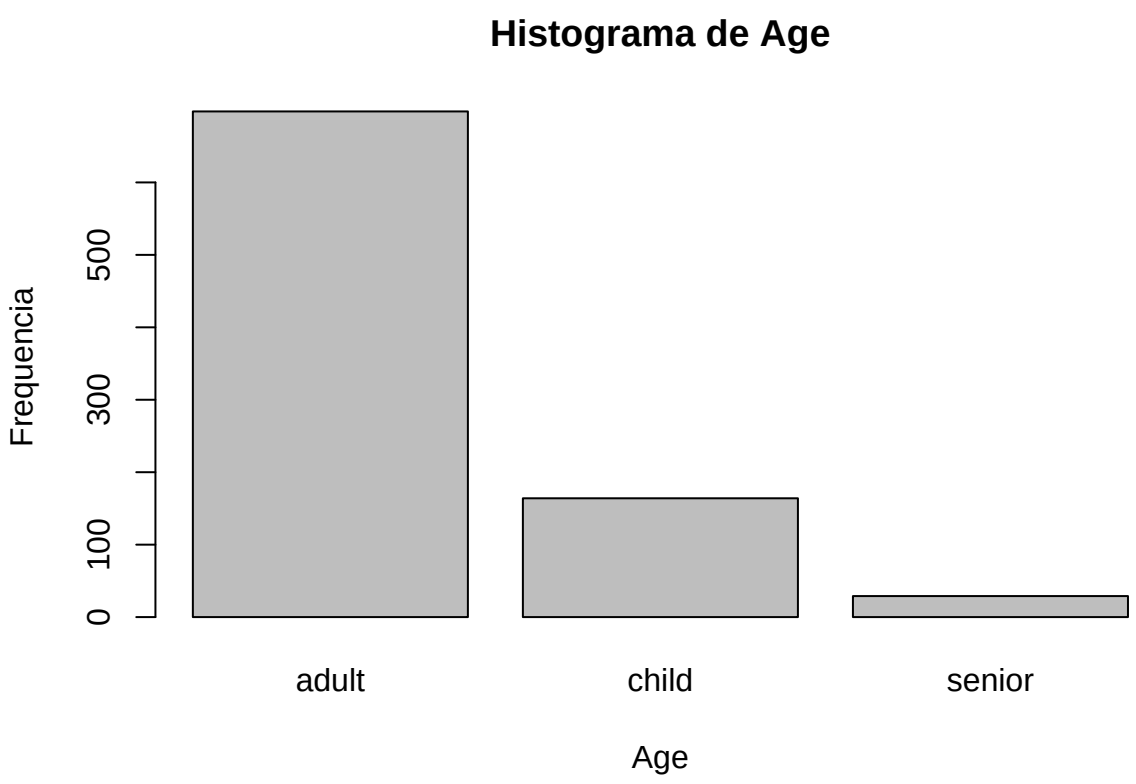
Visualizamos los histogramas de las cuatro variables de entrada:

```
barplot(table(df$Pclass), xlab="Class", ylab="Frequency", main="Histograma de Passenger Class")
```

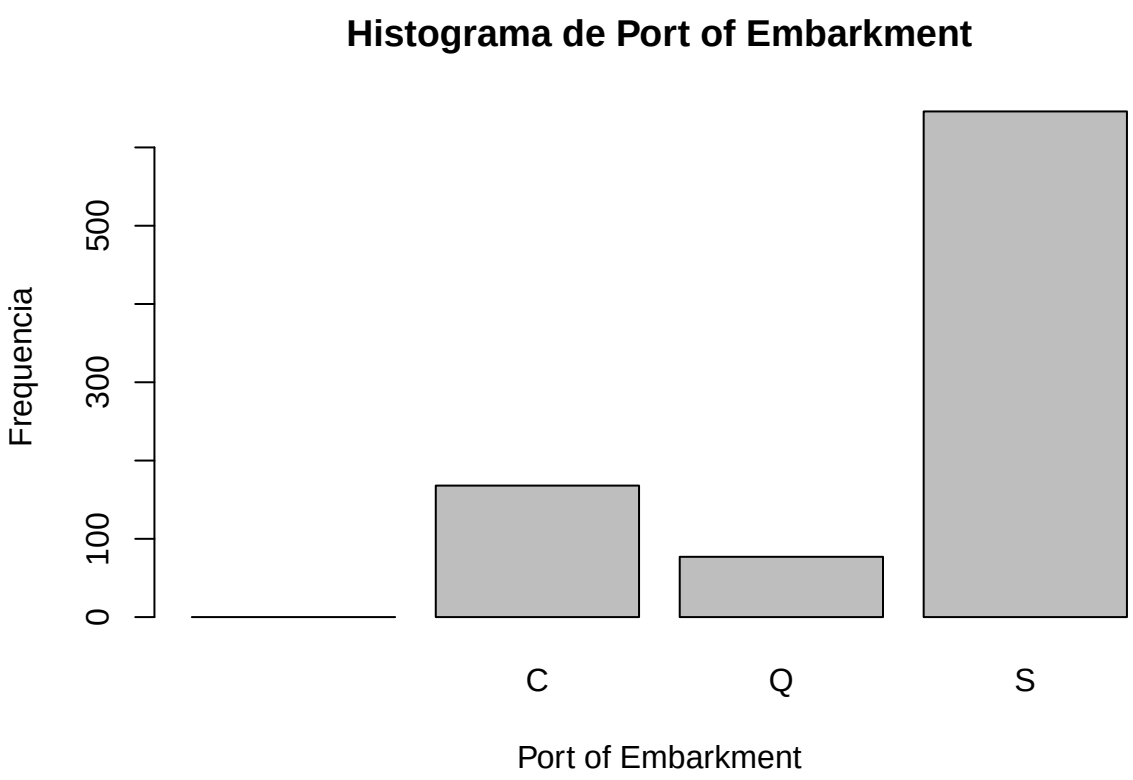

```r
barplot(table(df$Sex), xlab="Sex", ylab="Frequencia", main="Histograma de Sex")
```




```r
barplot(table(df$Age), xlab="Age", ylab="Frequencia", main="Histograma de Age")
```




```
barplot(table(df$Embarked), xlab="Port of Embarkment", ylab="Frequencia", main="Histograma de Port of Em
```




#### 4.2.2 Test

Antes de comenzar con las pruebas, convertimos el dataframe categórico en un dataframe numérico.

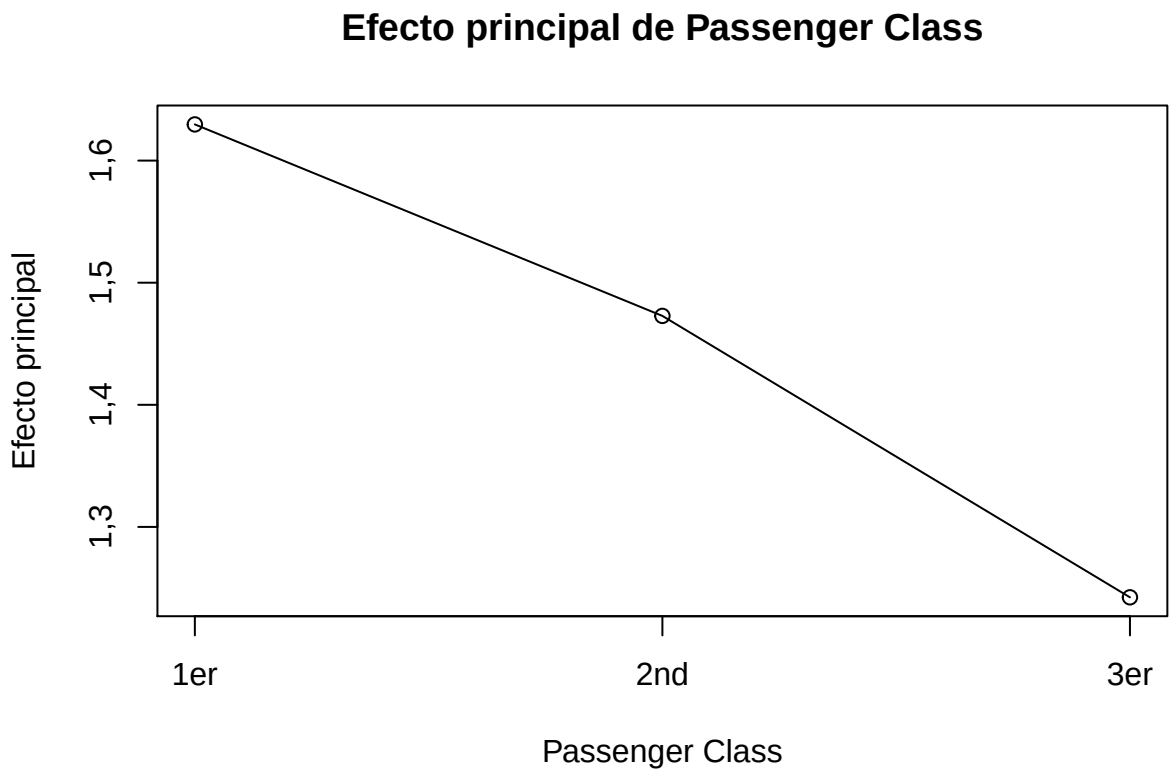
```
old_df = df
df$Pclass = as.integer(df$Pclass)
df$Sex = as.integer(df$Sex)
df$Age = as.integer(df$Age)
df$Embarked = as.integer(df$Embarked)
df$Survived = as.integer(df$Survived)
head(df)
```

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

Parece haber más sobrevivientes en promedio de la 1ra clase en comparación con los otros dos. El número más bajo de sobrevivientes en promedio fue de la 3ra clase.

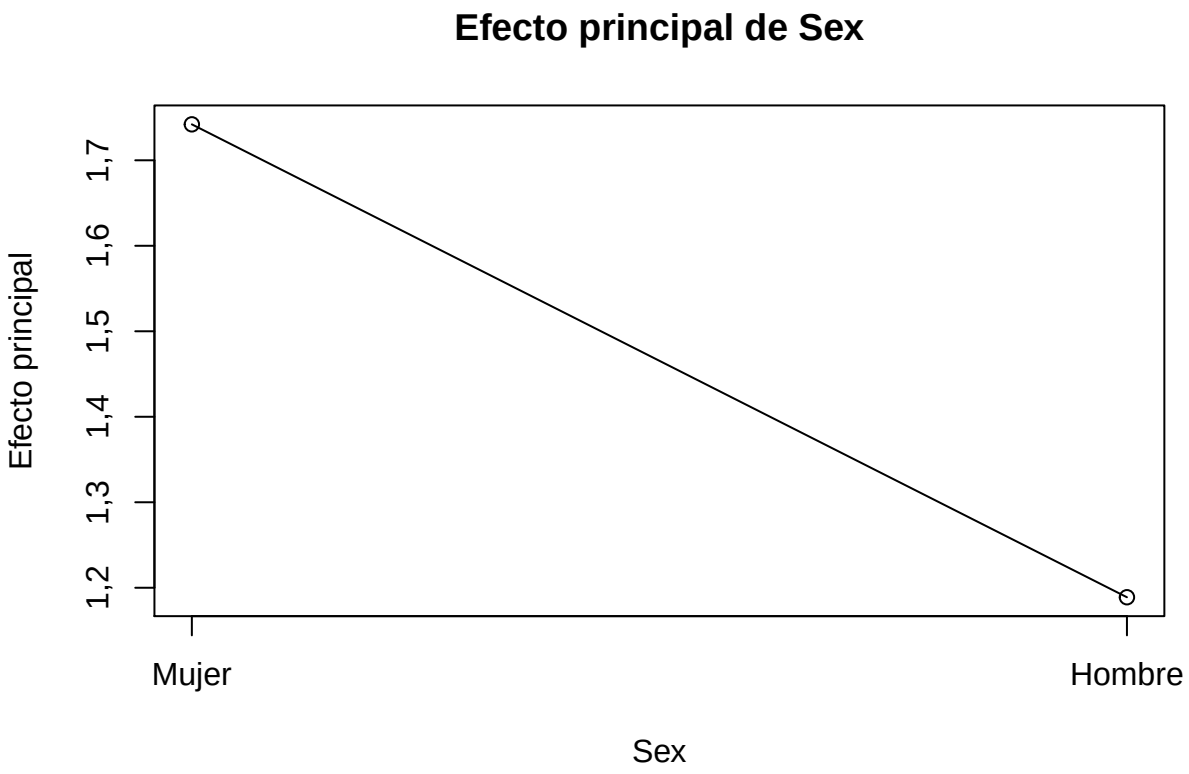
```
me_pclass = c(0,0,0)
me_pclass[1] = mean(df$Survived[df$Pclass==1])
me_pclass[2] = mean(df$Survived[df$Pclass==2])
me_pclass[3] = mean(df$Survived[df$Pclass==3])
```

```
plot(me_pclass, type="o", main="Efecto principal de Passenger Class", xlab="Passenger Class", ylab="Efe
     xaxt="n")
axis(1, at=c(1,2,3), labels=c("1er", "2nd", "3er"))
```



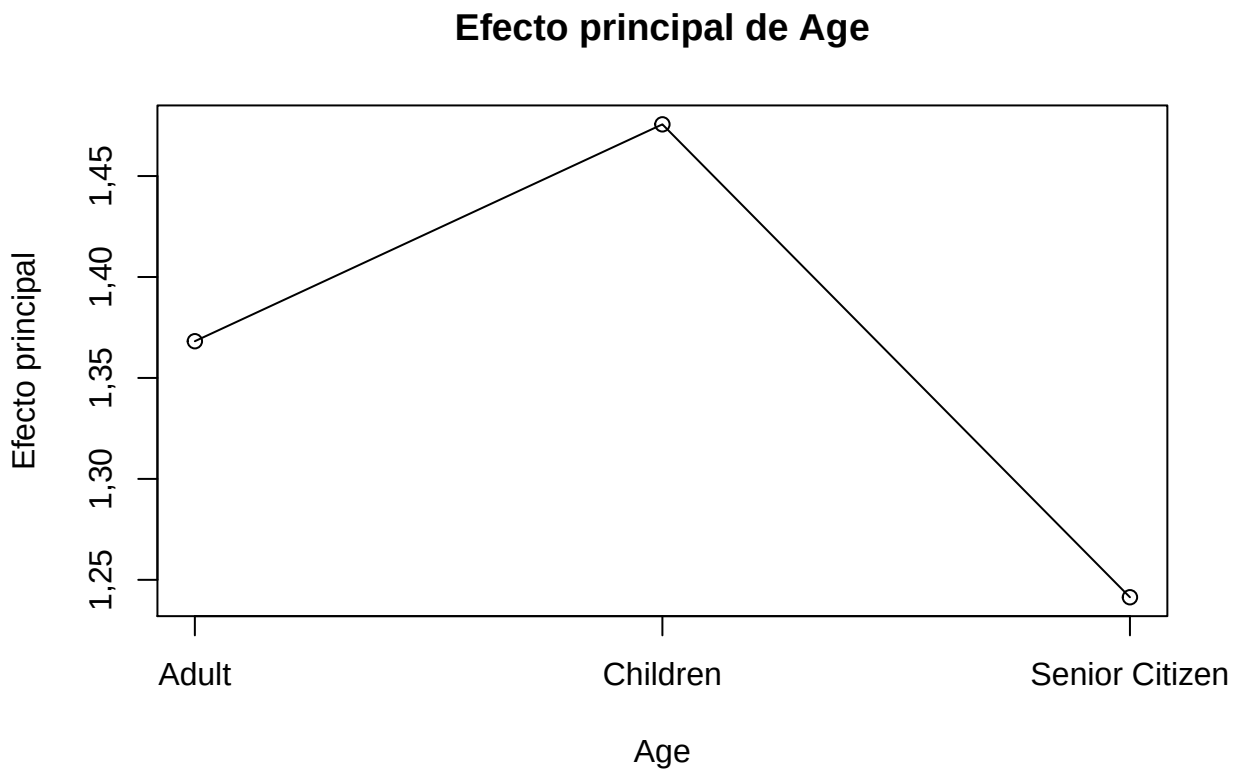

Las sobrevivientes femeninas eran mucho más numerosas que los hombres.

```
me_sex = c(0,0)
me_sex[1] = mean(df$Survived[df$Sex==1])
me_sex[2] = mean(df$Survived[df$Sex==2])
plot(me_sex, type="o", main="Efecto principal de Sex", xlab="Sex", ylab="Efecto principal", xaxt="n")
axis(1, at=c(1,2), labels=c("Mujer", "Hombre"))
```



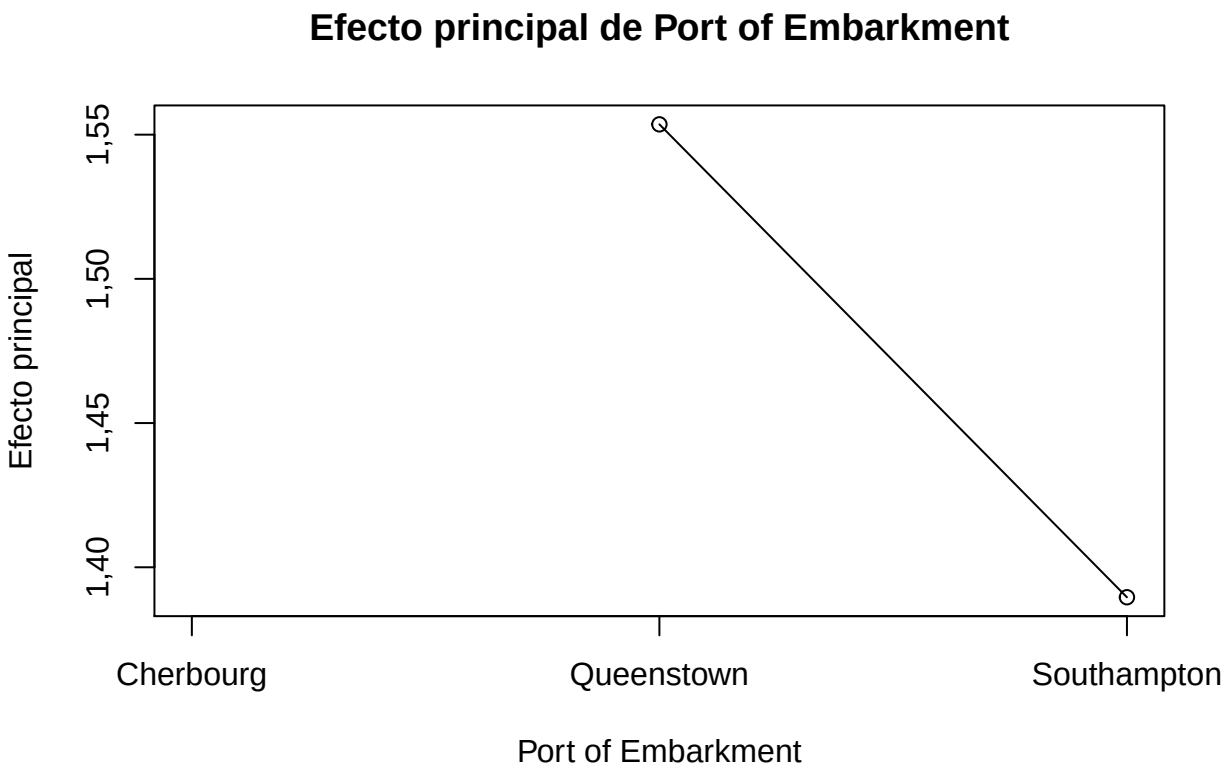

Los supervivientes máximos en promedio de la categoría de edad eran niños, seguidos de adultos y personas mayores.

```
me_age = c(0,0,0)
me_age[1] = mean(df$Survived[df$Age==1])
me_age[2] = mean(df$Survived[df$Age==2])
me_age[3] = mean(df$Survived[df$Age==3])
plot(me_age, type="o", main="Efecto principal de Age", xlab="Age", ylab="Efecto principal", xaxt="n")
axis(1, at=c(1,2,3), labels=c("Adult", "Children", "Senior Citizen"))
```



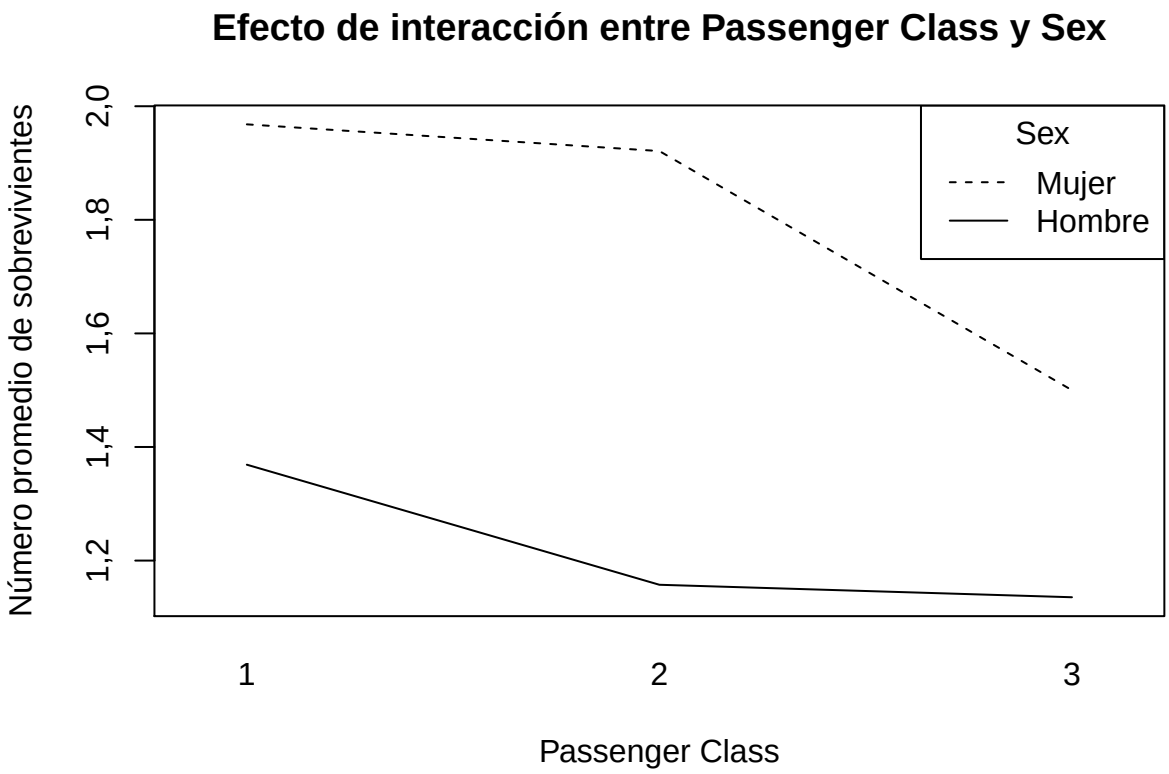

La gente que abordó el Titanic en Cherbourg tenía el número máximo de supervivientes en promedio, seguidos de los que abordaron el barco en Southampton y finalmente en Queenstown.

```
me_emb = c(0,0,0)
me_emb[1] = mean(df$Survived[df$Embarked==1])
me_emb[2] = mean(df$Survived[df$Embarked==2])
me_emb[3] = mean(df$Survived[df$Embarked==3])
plot(me_emb, type="o", main="Efecto principal de Port of Embarkment", xlab="Port of Embarkment", ylab="
     xaxt="n")
axis(1, at=c(1,2,3), labels=c("Cherbourg", "Queenstown", "Southampton"))
```



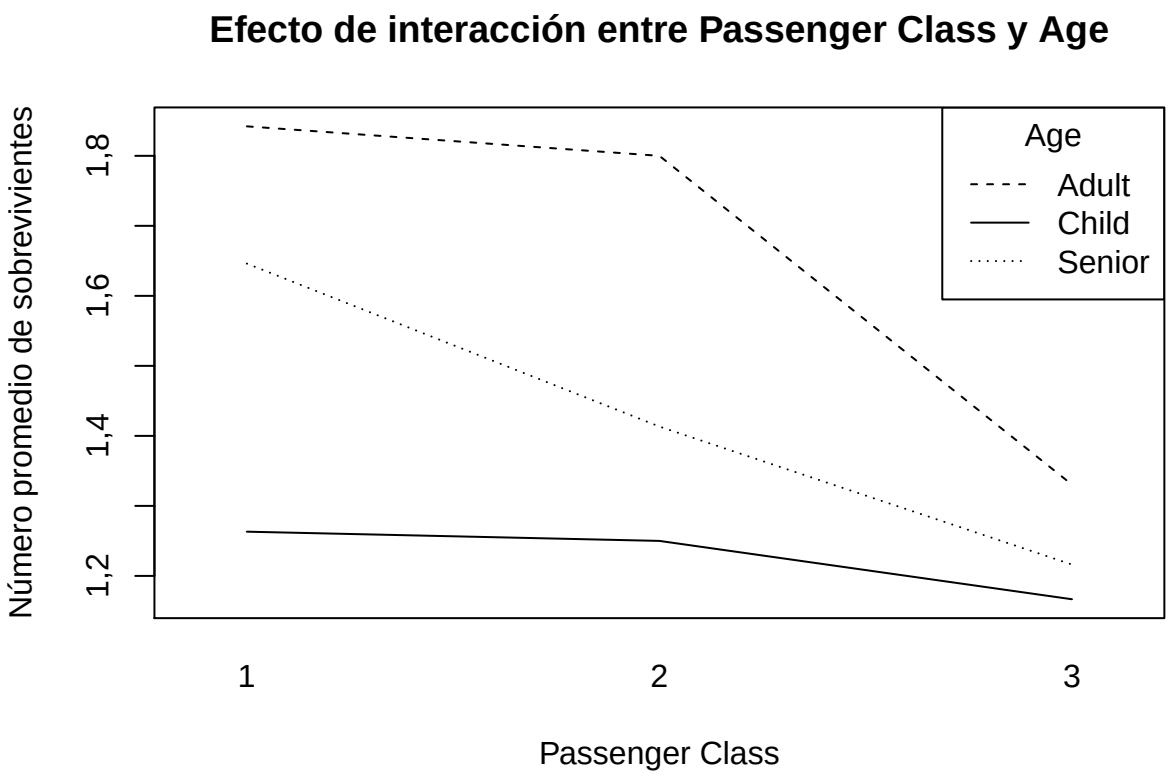

Existe un claro efecto de interacción entre la clase de pasajeros y el sexo. Los pasajeros femeninos de primera y segunda clase tuvieron un mayor número de sobrevivientes que las pasajeras de tercera clase. Los pasajeros varones de primera clase tenían un número promedio mayor de sobrevivientes que los pasajeros masculinos de segunda y tercera clase.

```
interaction.plot(df$Pclass, df$Sex, df$Survived, xlab="Passenger Class", ylab="Número promedio de sobre
                 main="Efecto de interacción entre Passenger Class y Sex", legend=FALSE)
legend("topright", c("Mujer","Hombre"), lty=c("dashed", "solid"), title="Sex")
```



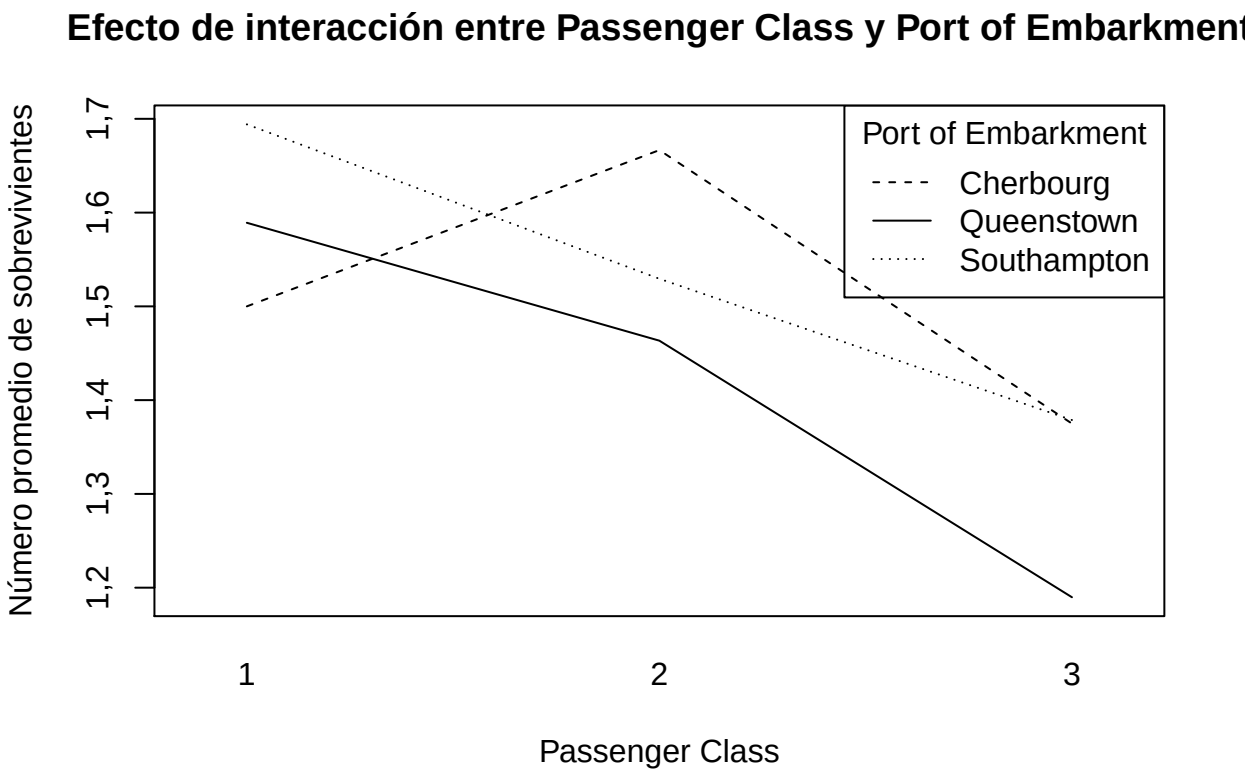

Hay un efecto de interacción entre la clase de pasajero y la edad. En general, los adultos tenían una media de sobrevivientes mejor que las personas mayores, que eran mejores que los niños. Más adultos de 1ra clase sobrevivieron que la 2da clase, que tuvo más sobrevivientes que la 3ra clase. Lo mismo es la tendencia para las personas mayores también. Sin embargo, el número promedio de sobrevivientes fue más alto para los niños de 2a clase y el más bajo para los niños de 1ra clase.

```
interaction.plot(df$Pclass, df$Age, df$Survived, xlab="Passenger Class", ylab="Número promedio de sobrev
                 main="Efecto de interacción entre Passenger Class y Age", legend=FALSE)
legend("topright", c("Adult","Child", "Senior"), lty=c("dashed", "solid", "dotted"), title="Age")
```




También hay un efecto de interacción entre la Clase de Pasajero y el Puerto de Embarque. Para los pasajeros que abordaron el barco desde Queenstown y Southampton, los pasajeros de primera clase sobrevivieron más y los pasajeros de tercera clase sobrevivieron lo mínimo. Para los pasajeros que abordaron el barco desde Cherbourg, el número medio de supervivientes era casi el mismo para los pasajeros de 1ra y 2da clase, que eran mayores que los pasajeros de 3ra clase. Para los pasajeros de segunda y tercera clase que abordaron desde Cherbourg y Queenstown, no parece haber ningún efecto de interacción.

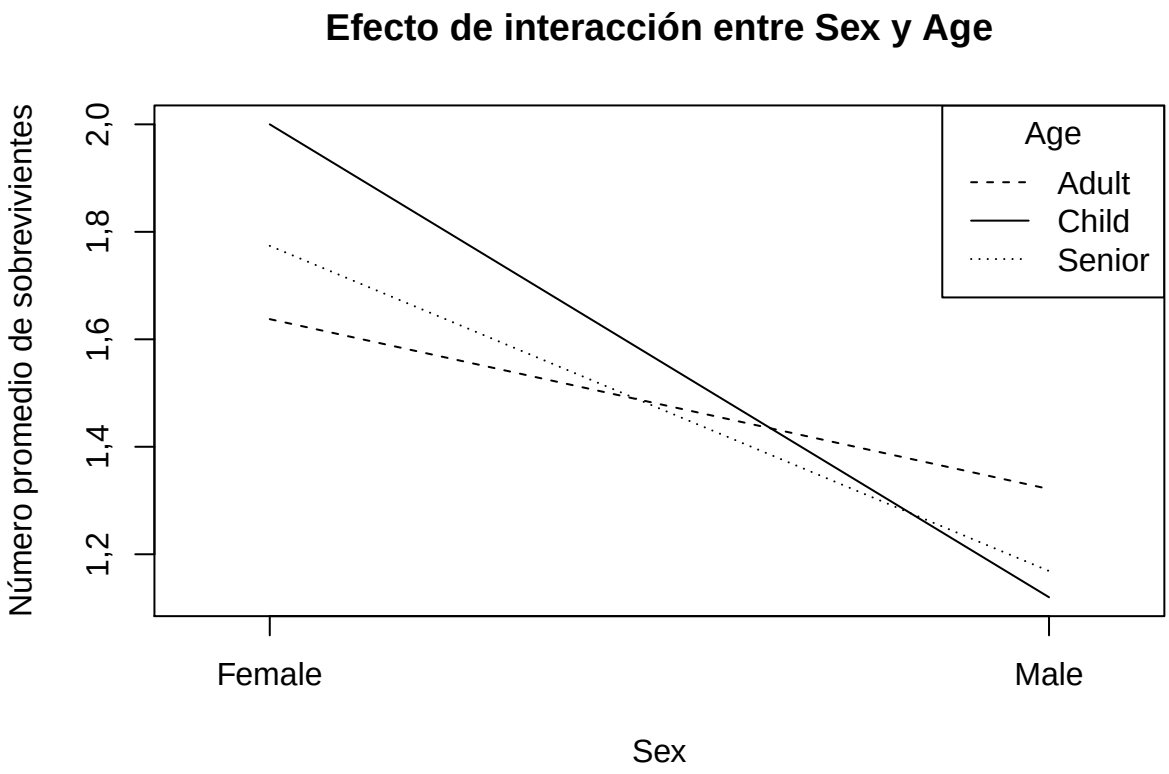
```
interaction.plot(df$Pclass, df$Embarked, df$Survived, xlab="Passenger Class", ylab="Número promedio de s
                 main="Efecto de interacción entre Passenger Class y Port of Embarkment", legend=FALSE)
legend("topright", c("Cherbourg","Queenstown","Southampton"), lty=c("dashed", "solid", "dotted"),
       title="Port of Embarkment")
```




Es interesante observar que en un momento tan crítico (cuando el barco se hundía), el número medio de supervivientes para la 1ra clase era mucho menor que las otras dos clases. Esto esencialmente refleja y refuerza su poder y estatura en la sociedad: se les dio la prioridad de alcanzar la seguridad, incluso en un momento en que todo el barco se hundía y casi todos tenían una oportunidad sustancial de encontrarse con su muerte ese día.

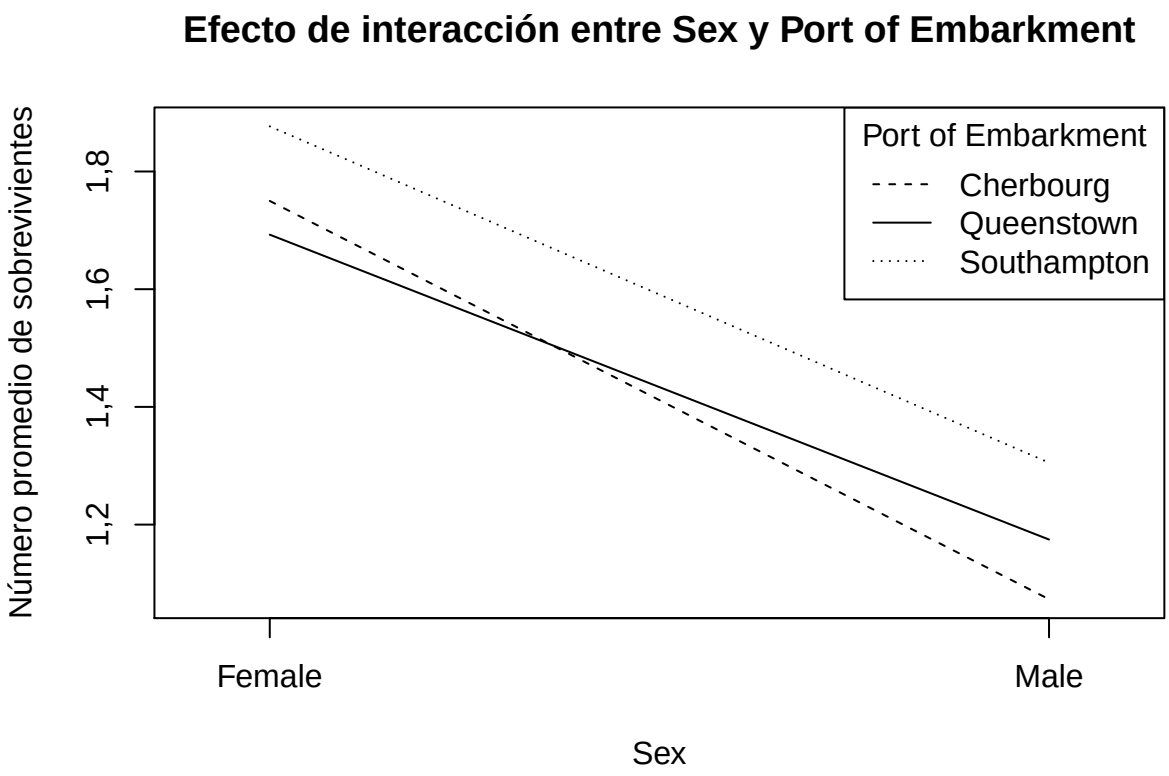
Parece haber un efecto de interacción entre la edad y el sexo. Hubo más sobrevivientes de hombres que adultos varones, que fueron mayores que los varones. Por el contrario, hubo más sobrevivientes de niñas que mujeres de la tercera edad, que fueron mayores que las mujeres adultas. En general, hubo más mujeres supervivientes en comparación con los hombres en todos los grupos de edad.

```
interaction.plot(df$Sex, df$Age, df$Survived, xlab="Sex", ylab="Número promedio de sobrevivientes",
                 main="Efecto de interacción entre Sex y Age", legend=FALSE, xtick=FALSE, xaxt="n")
axis(1, c(1,2), labels=c("Female", "Male"))
legend("topright", c("Adult","Child", "Senior"), lty=c("dashed", "solid", "dotted"), title="Age")
```



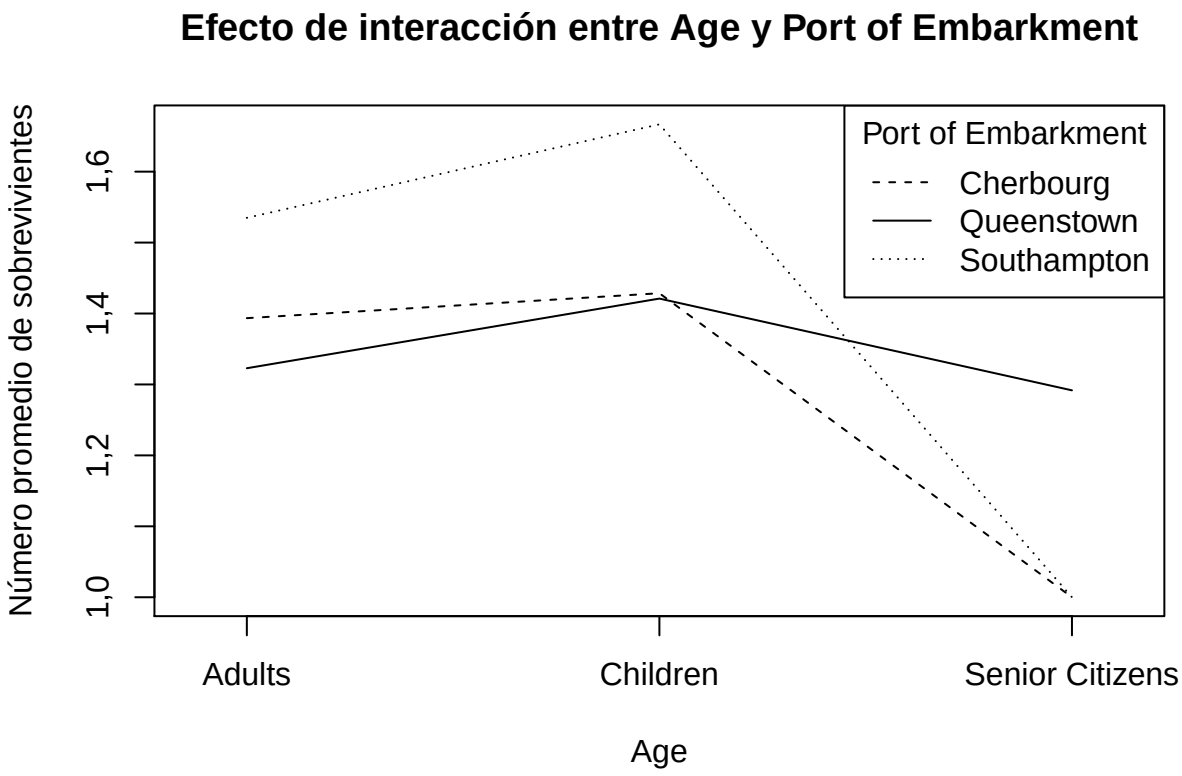

No parece haber ningún efecto de interacción entre Sexo y Puerto de Embarque.

```
interaction.plot(df$Sex, df$Embarked, df$Survived, xlab="Sex", ylab="Número promedio de sobrevivientes"
                 main="Efecto de interacción entre Sex y Port of Embarkment", legend=FALSE, xtick = FALS
axis(1, c(1,2), labels=c("Female", "Male"))
legend("topright", c("Cherbourg","Queenstown","Southampton"), lty=c("dashed", "solid", "dotted"),
       title="Port of Embarkment")
```




Hay un efecto de interacción entre la edad y el puerto de embarque. Los jubilados de Southampton y Cherbourg sobrevivieron en menor número que los que abordaron el barco desde Queenstown. Para adultos y niños, las personas que abordaron en Southampton sobrevivieron más que las de Queenstown, que eran más que las de Cherbourg. No parece haber un efecto de interacción en la parte adultos-niños del gráfico.

```
interaction.plot(df$Age, df$Embarked, df$Survived, xlab="Age", ylab="Número promedio de sobrevivientes"
                 main="Efecto de interacción entre Age y Port of Embarkment", legend=FALSE, xtick = FALS
axis(1, c(1,2,3), labels=c("Adults", "Children", "Senior Citizens"))
legend("topright", c("Cherbourg","Queenstown","Southampton"), lty=c("dashed", "solid", "dotted"),
       title="Port of Embarkment")
```




## 4.3 ANOVA

Los principales efectos de Clase de Pasajero, Sexo y Puerto de Embarque son significativos. Hubo más sobrevivientes en la primera clase en comparación con la segunda y tercera clase. Hubo más mujeres sobrevivientes. No hay un efecto significativo de la edad, lo que significa que el número medio de supervivientes de todos los grupos de edad fue casi el mismo.

```
me1 = aov(df$Survived ~ df$Pclass)
anova(me1)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## df$Pclass   1  24,143 24,1429  115,03 < 2,2e-16 ***
## Residuals 889 186,584  0,2099
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

```
me2 = aov(df$Survived ~ df$Sex)
anova(me2)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##            Df  Sum Sq Mean Sq F value    Pr(>F)
```

```
## df$Sex      1  62,213  62,213  372,41 < 2,2e-16 ***
## Residuals 889 148,514   0,167
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

```
me3 = aov(df$Survived ~ df$Age)
anova(me3)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##            Df  Sum Sq Mean Sq F value Pr(>F)
## df$Age      1   0,205 0,20506   0,866 0,3523
## Residuals 889 210,522 0,23681
```

```
me4 = aov(df$Survived ~ df$Embarked)
anova(me4)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## df$Embarked   1   5,925  5,9246  25,717 4,811e-07 ***
## Residuals   889 204,803  0,2304
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

Existen importantes efectos principales de Clase de Pasajero, Sexo y Puerto de Embarque. Esto se ve reforzado una vez más por los ANOVA de interacción bidireccional. Los efectos de interacción de la Clase de Pasajeros son significativos con Sexo y Edad, pero no con Puerto de Embarque. Los efectos de interacción del sexo no son significativos con la edad o el puerto de embarque. El efecto de interacción de Age with Port of Embarkment no es significativo.

```
ie12 = aov(df$Survived ~ df$Pclass * df$Sex)
anova(ie12)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## df$Pclass          1  24,143  24,143 164,061 < 2,2e-16 ***
## df$Sex             1  53,337  53,337 362,450 < 2,2e-16 ***
## df$Pclass:df$Sex   1   2,718   2,718  18,471 1,916e-05 ***
## Residuals        887 130,529   0,147
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

```
ie13 = aov(df$Survived ~ df$Pclass * df$Age)
anova(ie13)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##                   Df  Sum Sq Mean Sq  F value    Pr(>F)
## df$Pclass          1  24,143 24,1429 116,0488 < 2,2e-16 ***
## df$Age             1   0,223  0,2231   1,0723  0,300711
## df$Pclass:df$Age   1   1,829  1,8289   8,7912  0,003108 **
## Residuals        887 184,532  0,2080
```

```
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

```
ie14 = aov(df$Survived ~ df$Pclass * df$Embarked)
anova(ie14)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##                          Df  Sum Sq Mean Sq F value    Pr(>F)
## df$Pclass                 1  24,143 24,1429 116,817 < 2,2e-16 ***
## df$Embarked               1   2,754  2,7540  13,325 0,0002771 ***
## df$Pclass:df$Embarked     1   0,512  0,5115   2,475 0,1160273
## Residuals               887 183,319  0,2067
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

```
ie23 = aov(df$Survived ~ df$Sex * df$Age)
anova(ie23)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##                 Df  Sum Sq Mean Sq  F value  Pr(>F)
## df$Sex           1  62,213  62,213 373,9415 < 2e-16 ***
## df$Age           1   0,009   0,009   0,0516 0,82041
## df$Sex:df$Age    1   0,934   0,934   5,6147 0,01802 *
## Residuals      887 147,571   0,166
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

```
ie24 = aov(df$Survived ~ df$Sex * df$Embarked)
anova(ie24)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##                       Df  Sum Sq Mean Sq  F value    Pr(>F)
## df$Sex                 1  62,213  62,213 378,4640 < 2,2e-16 ***
## df$Embarked            1   2,526   2,526  15,3691 9,524e-05 ***
## df$Sex:df$Embarked     1   0,180   0,180   1,0932    0,2961
## Residuals            887 145,808   0,164
## ---
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

```
ie34 = aov(df$Survived ~ df$Age * df$Embarked)
anova(ie34)
```

```
## Analysis of Variance Table
##
## Response: df$Survived
##                       Df  Sum Sq Mean Sq F value    Pr(>F)
## df$Age                 1   0,205  0,2051  0,8892    0,3459
## df$Embarked            1   5,930  5,9303 25,7155 4,818e-07 ***
## df$Age:df$Embarked     1   0,041  0,0406  0,1762    0,6748
## Residuals            887 204,551  0,2306
## ---
```

```
## Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1
```

### 4.3.1 Estimacion

En base al análisis de datos exploratorios, efectos principales, efectos de interacción bidireccional y ANOVA para todos los efectos principales y de interacción, podemos estimar que si el buque del Titanic volviera a zarpar (para el caso, cualquier barco) y si fuera para terminar con el mismo destino, entonces, en promedio, habría más sobrevivientes de la clase alta, más sobrevivientes que son mujeres y niños y más sobrevivientes que abordaron el barco en Cherbourg.

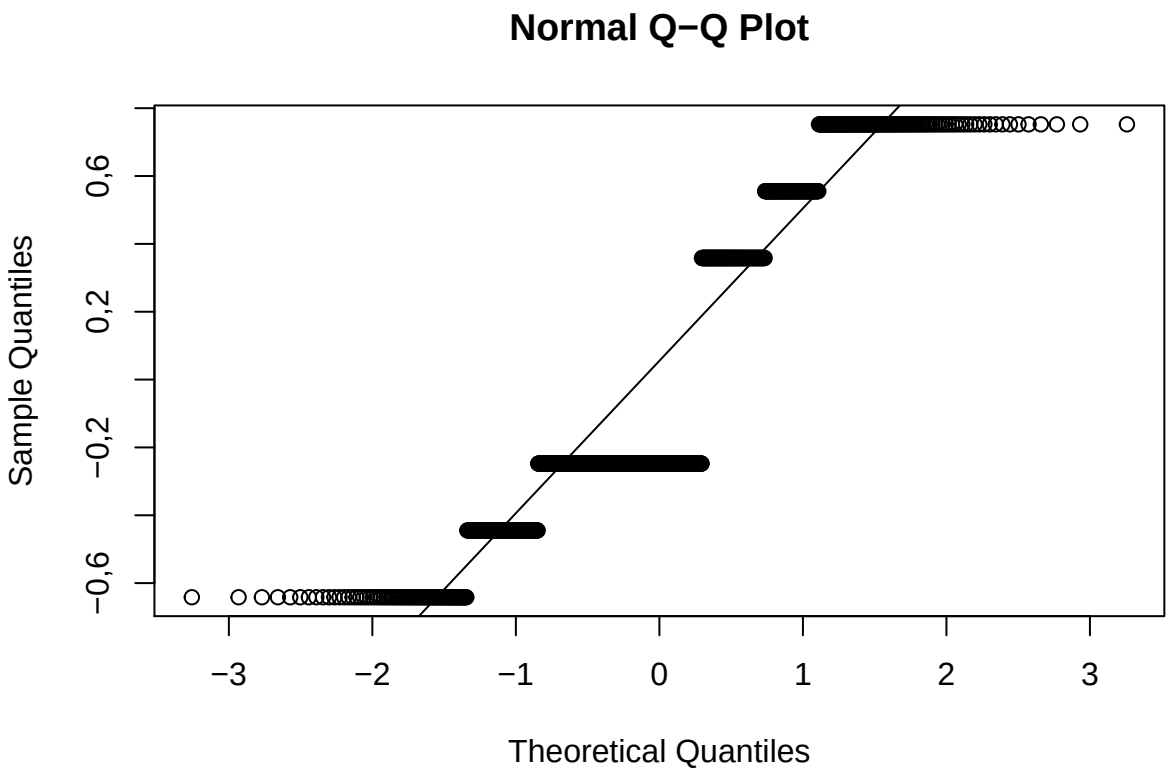
### 4.3.2 Diagnóstico y verificación de adecuación del modelo

El diagrama Q-Q es una herramienta para comparar dos distribuciones entre sí en base a la comparación de sus cuantiles. Ninguno de los gráficos de Q-Q a continuación se adhieren a la línea, lo que significa que los datos son de naturaleza altamente no lineal. Además, la no linealidad de los puntos sugiere que los datos no se distribuyen normalmente.

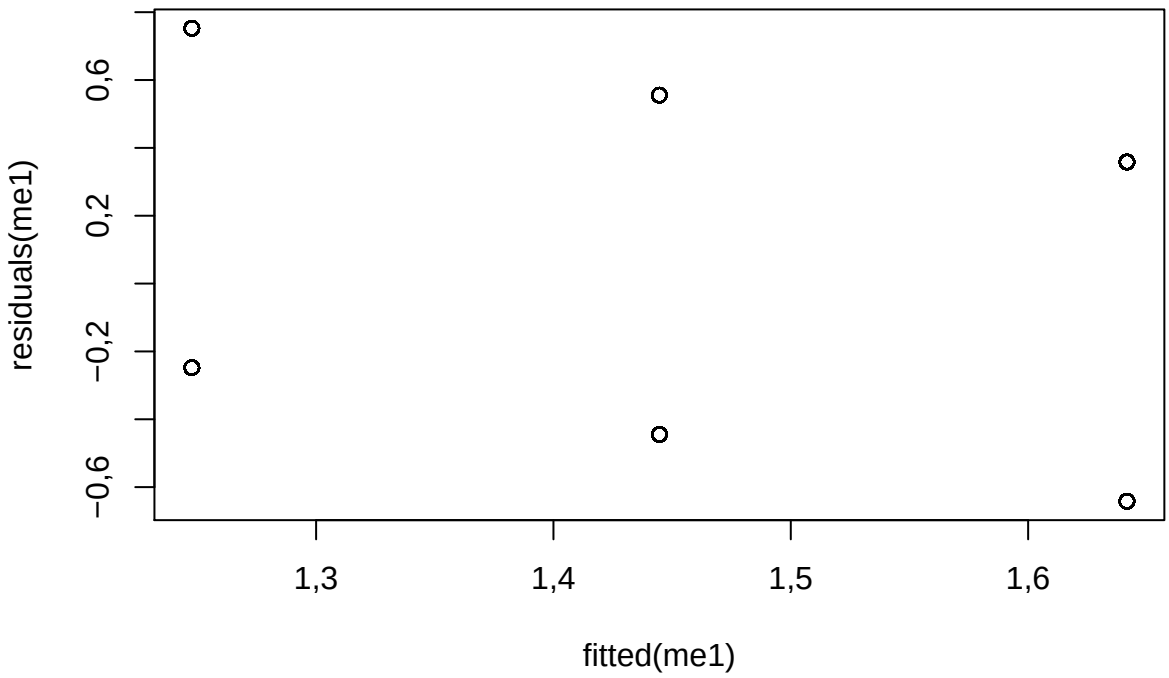
El gráfico residual se utiliza en el análisis de datos estadísticos para detectar la no linealidad de los datos, varianzas de errores desiguales y valores atípicos. En la mayoría de las siguientes parcelas residuales vs fit, los residuos no rebotan aleatoriamente alrededor de la línea cero, pero parecen ajustarse a una línea que tiene una pendiente negativa. Esto sugiere que los datos no son de naturaleza lineal. Los residuos forman aproximadamente una banda horizontal alrededor de la línea cero, lo que sugiere que las varianzas de los términos de error son iguales. Por último, no hay residuos que se destaquen del patrón básico de los residuos, lo que sugiere que no hay valores atípicos.

Estas observaciones sugieren que el conjunto de datos no ha cumplido todas las suposiciones para implementar ANOVA. Y esto es cierto porque, como se explicó anteriormente, no hubo alcance de replicación y medidas repetidas. Sin embargo, aún obtenemos resultados razonables porque hubo algunas suposiciones para ANOVA que eran verdaderas en el conjunto de datos.

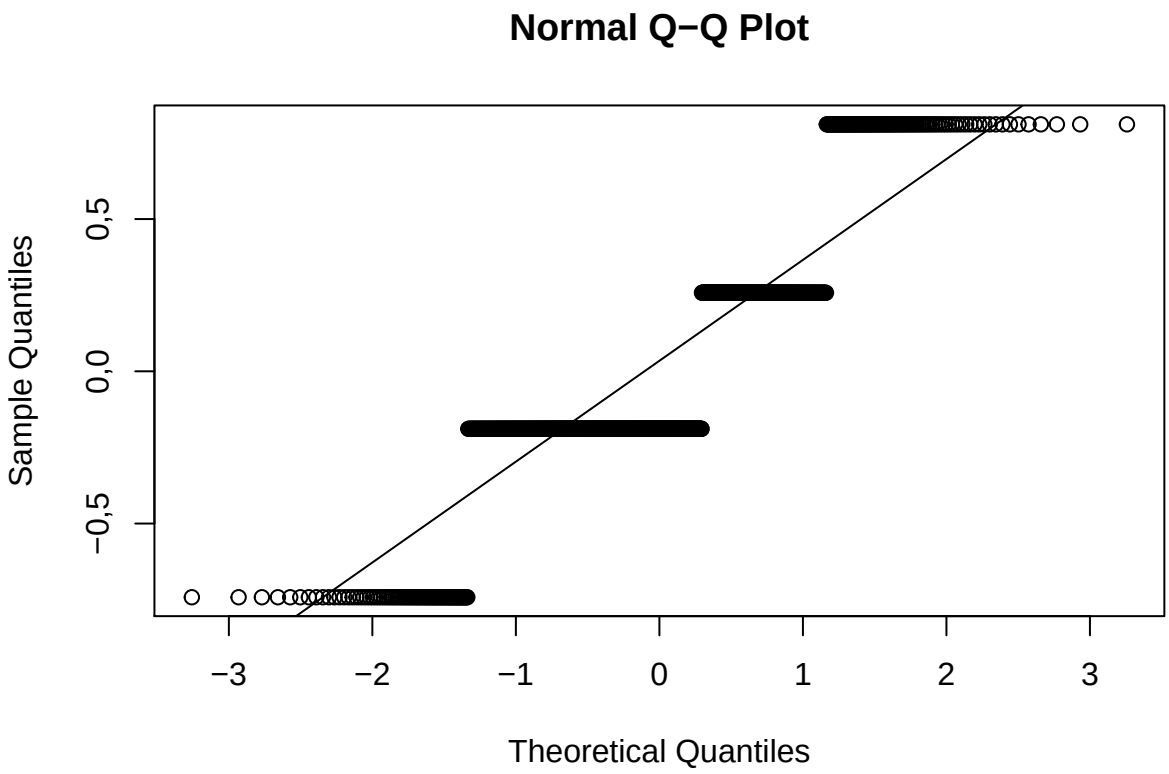
```
qqnorm(residuals(me1))
qqline(residuals(me1))
```



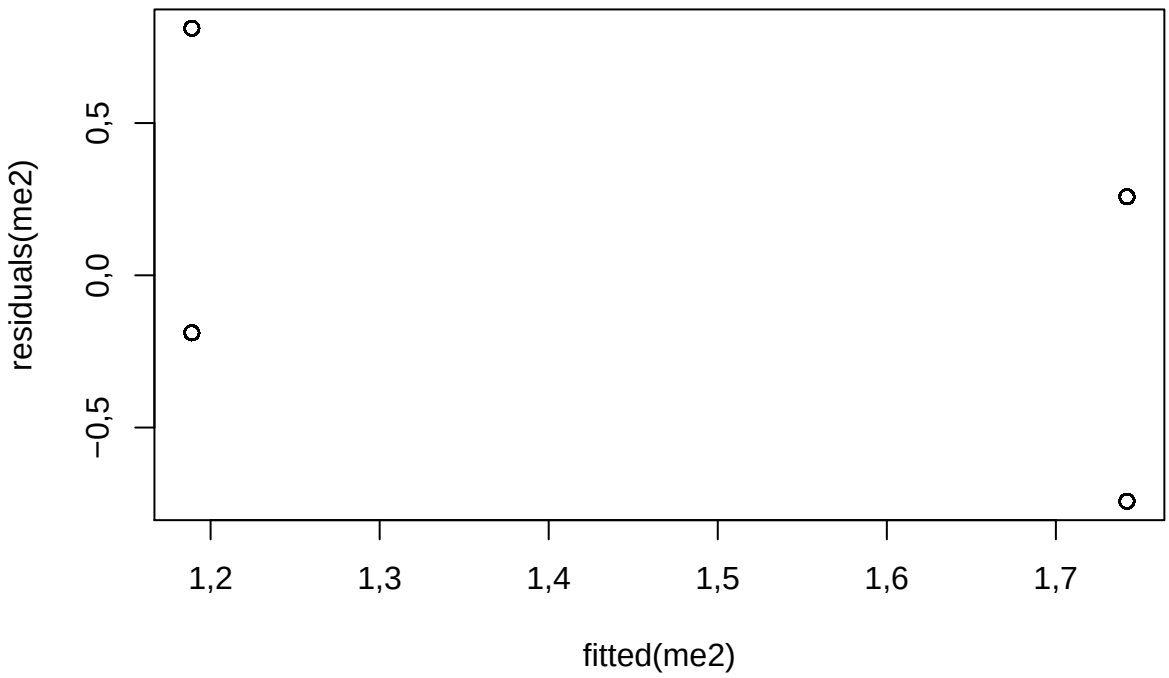

```
plot(fitted(me1), residuals(me1))
```



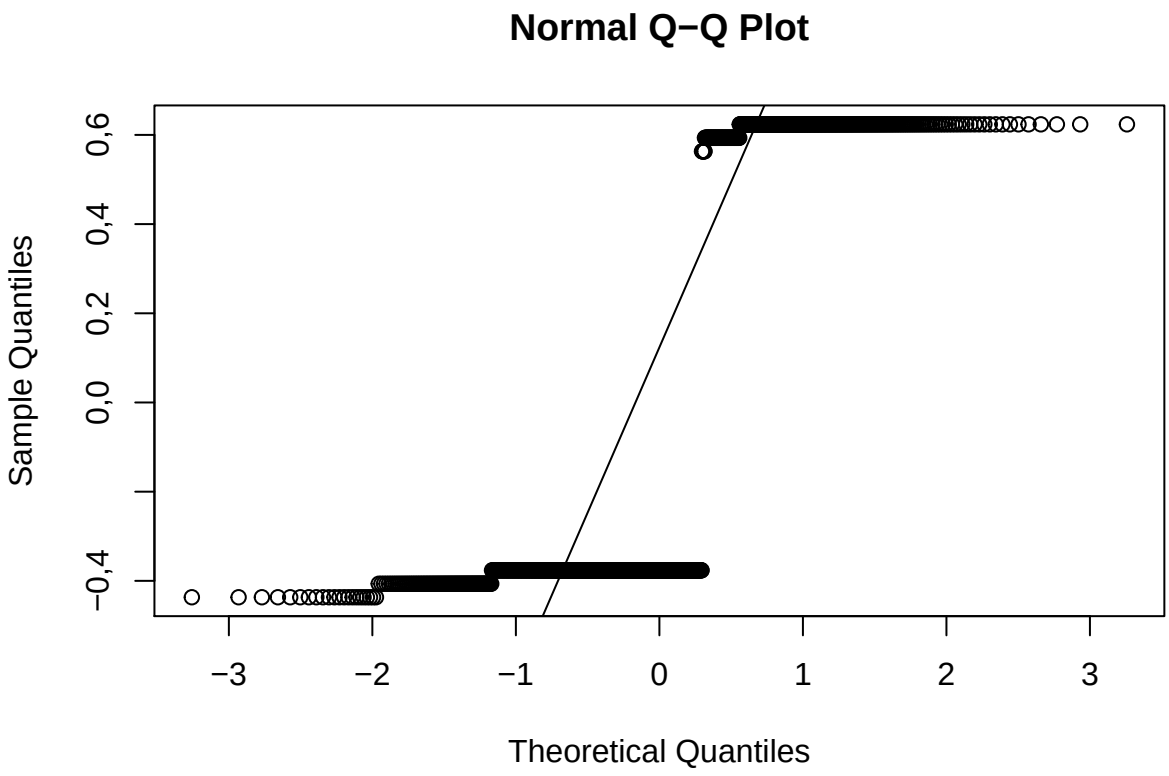

```
qqnorm(residuals(me2))
qqline(residuals(me2))
```



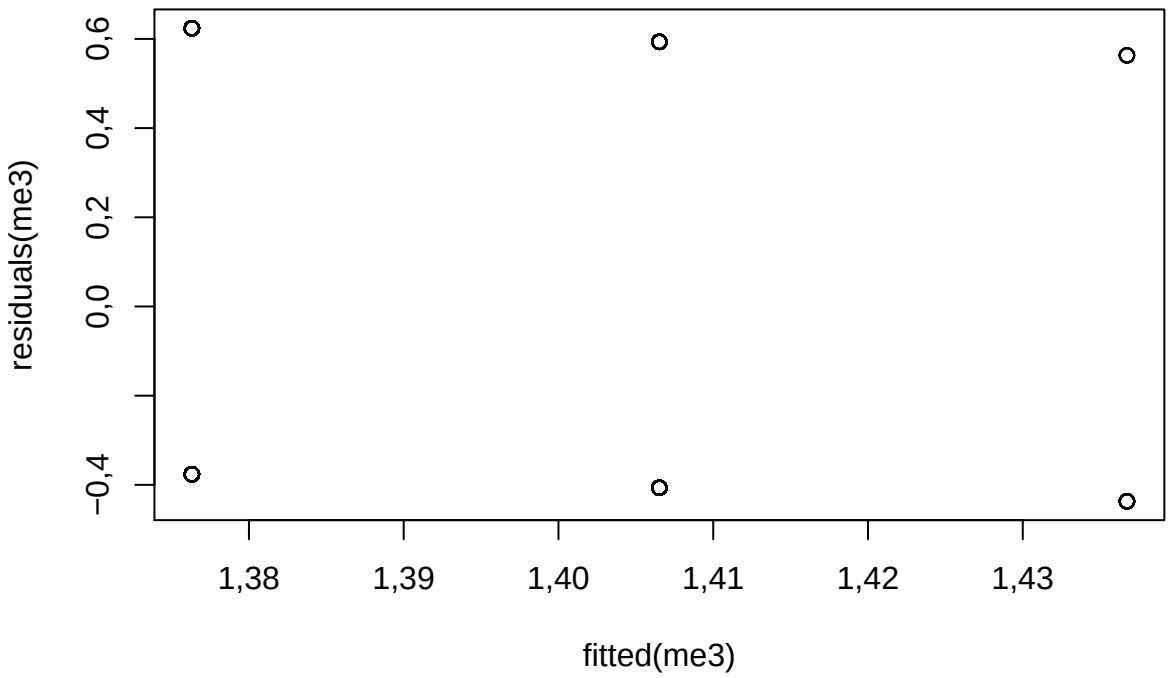

```
plot(fitted(me2), residuals(me2))
```



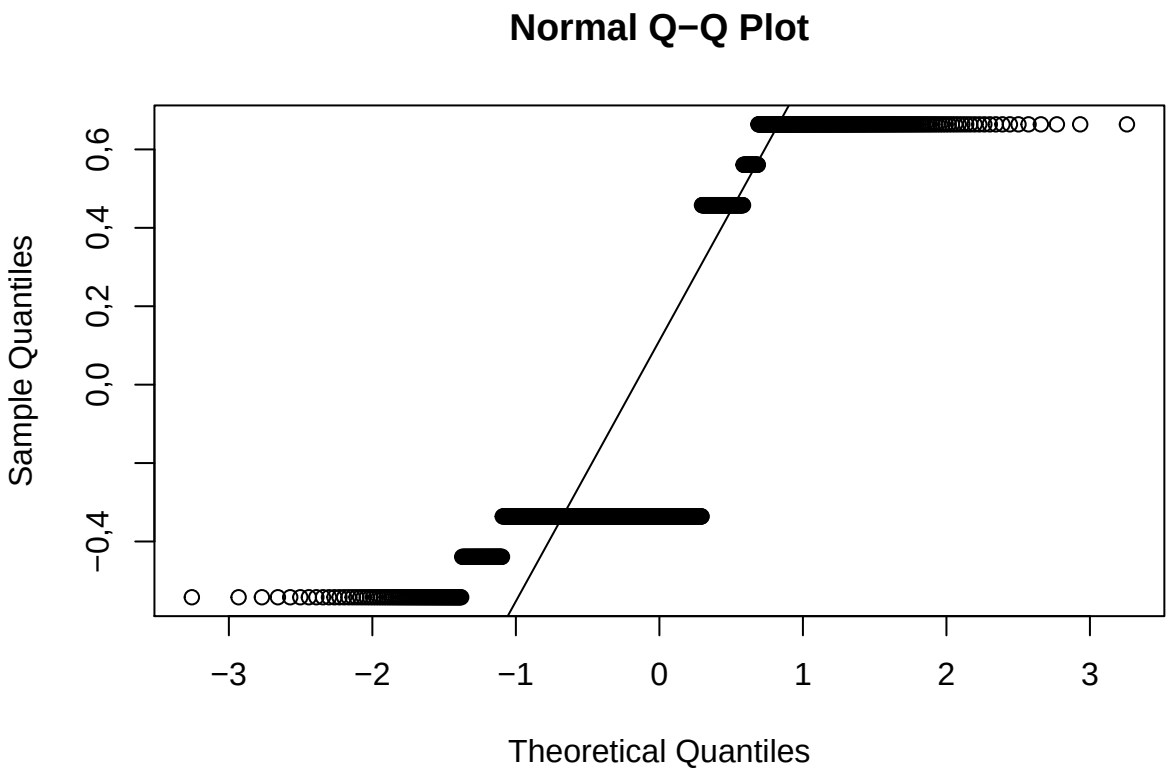

```
qqnorm(residuals(me3))
qqline(residuals(me3))
```



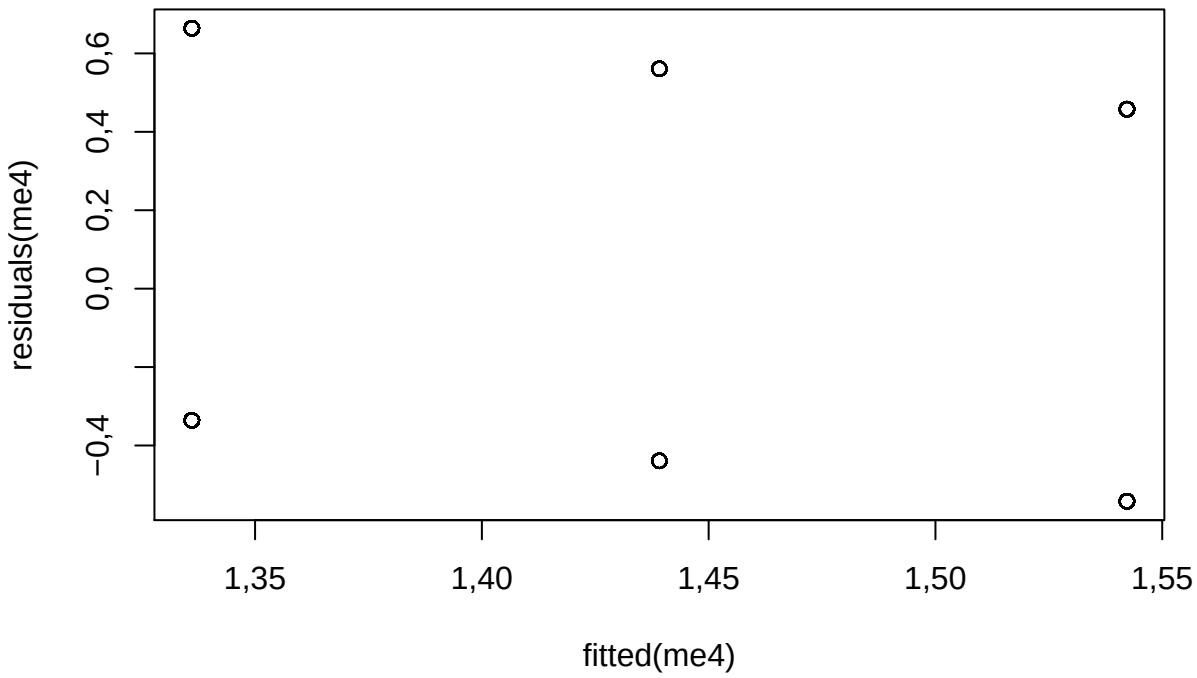

```
plot(fitted(me3), residuals(me3))
```



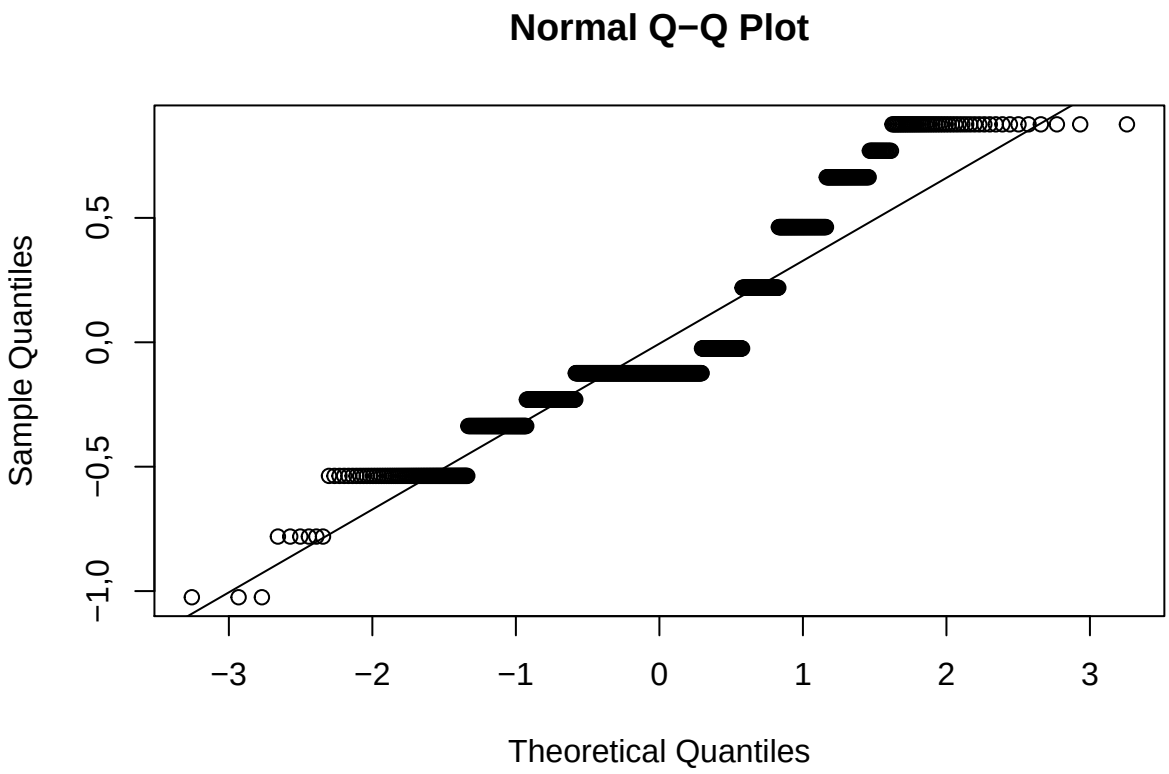

```
qqnorm(residuals(me4))
qqline(residuals(me4))
```



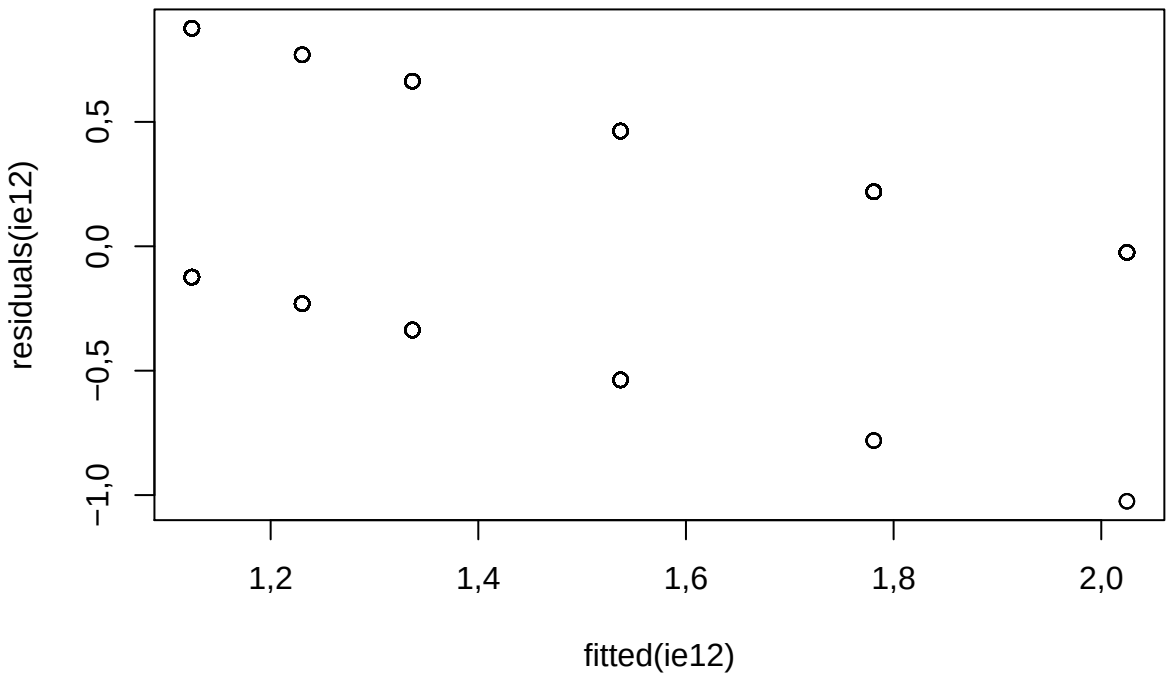

```
plot(fitted(me4), residuals(me4))
```



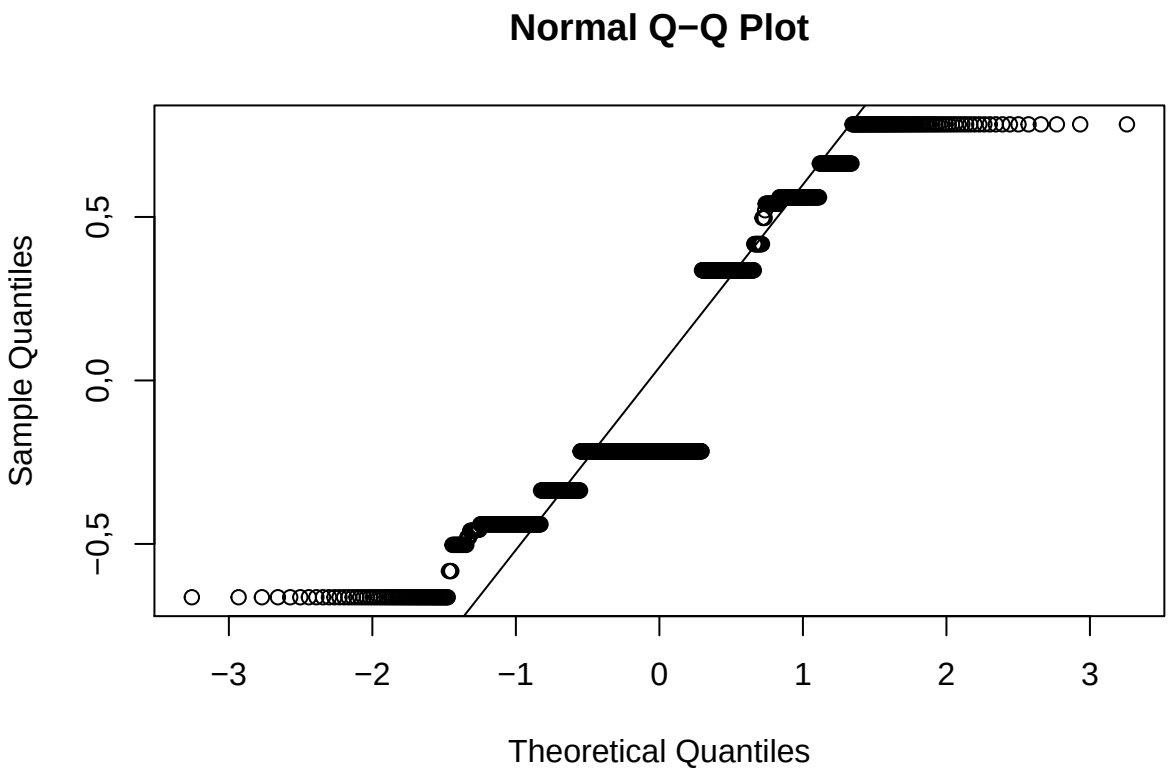

```
qqnorm(residuals(ie12))
qqline(residuals(ie12))
```



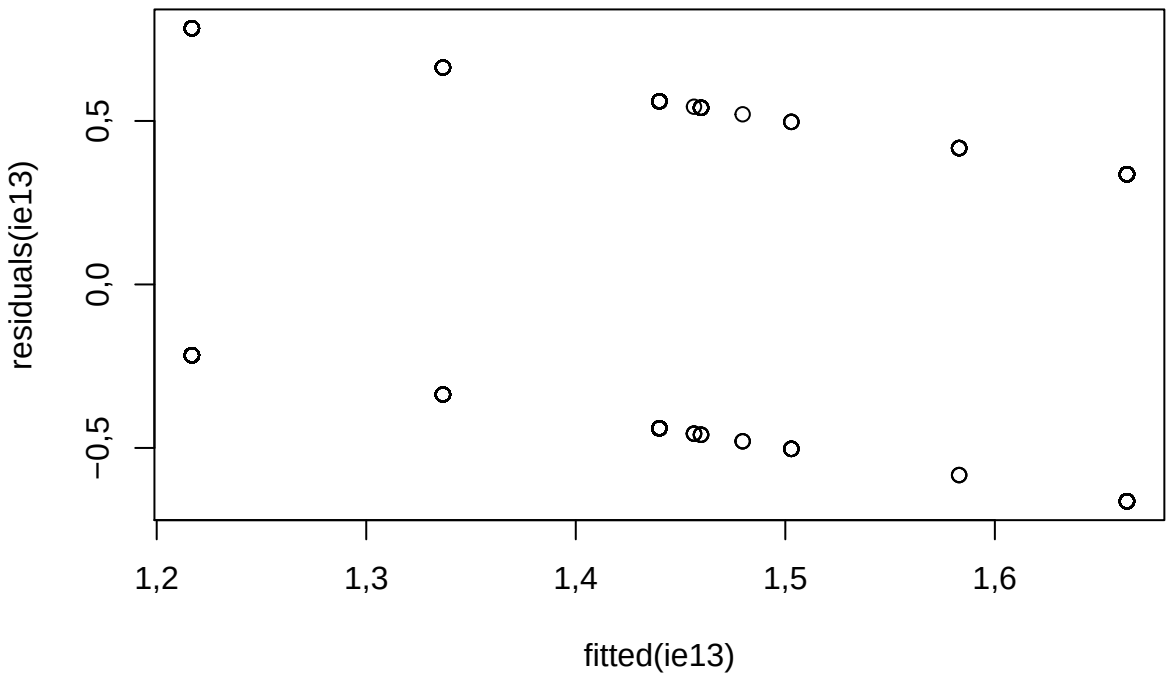

```
plot(fitted(ie12), residuals(ie12))
```



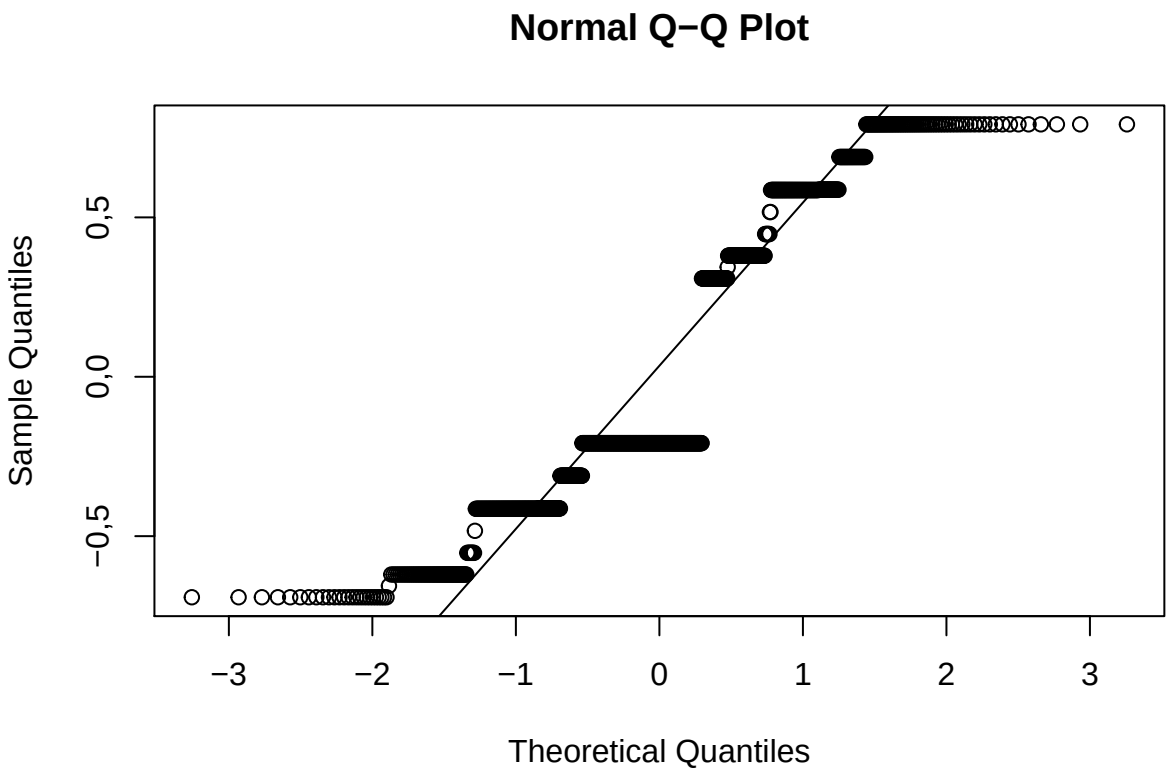

```
qqnorm(residuals(ie13))
qqline(residuals(ie13))
```



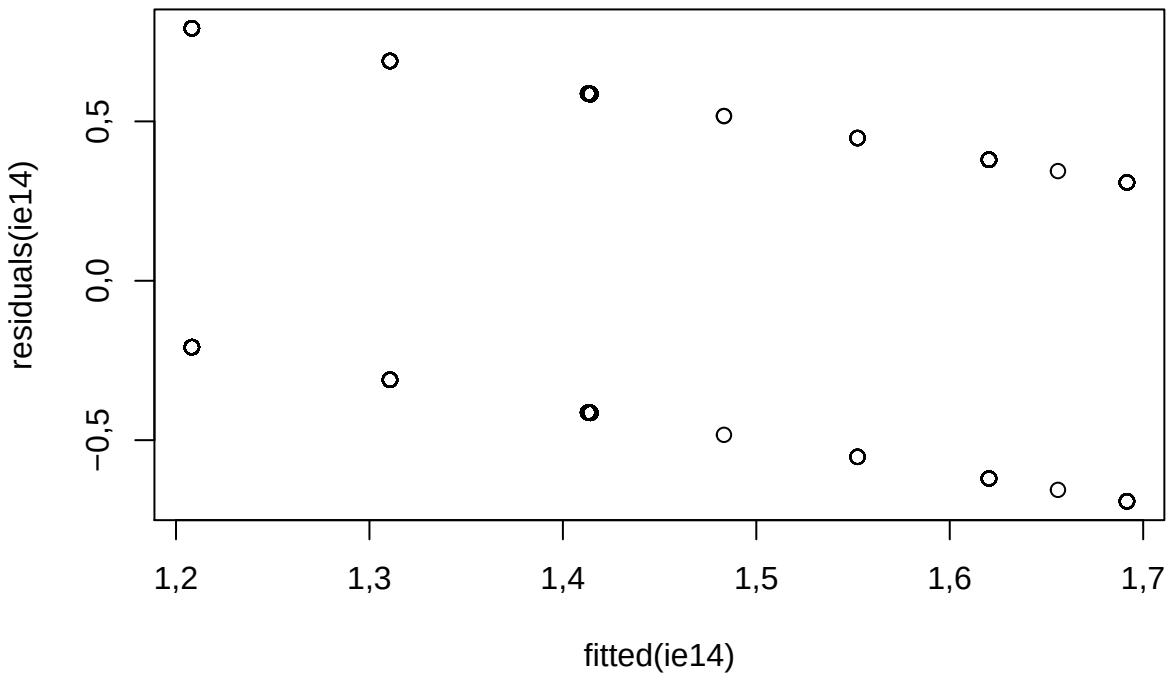

```
plot(fitted(ie13), residuals(ie13))
```



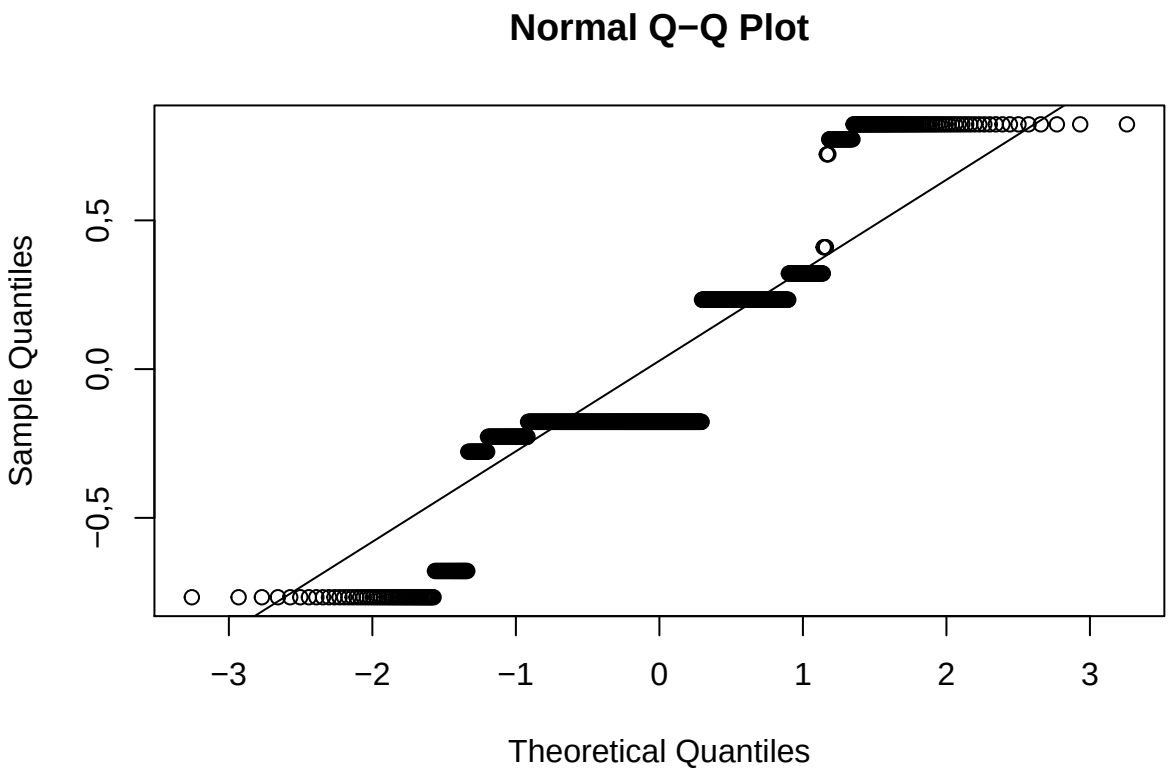

```
qqnorm(residuals(ie14))
qqline(residuals(ie14))
```



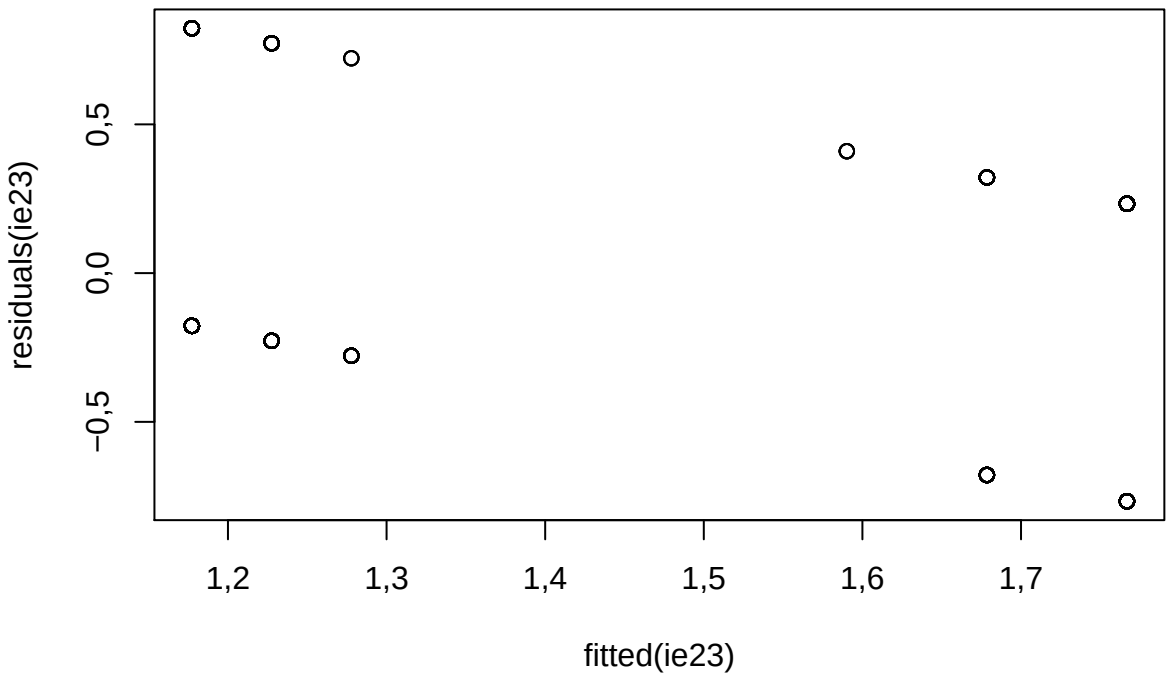

```
plot(fitted(ie14), residuals(ie14))
```



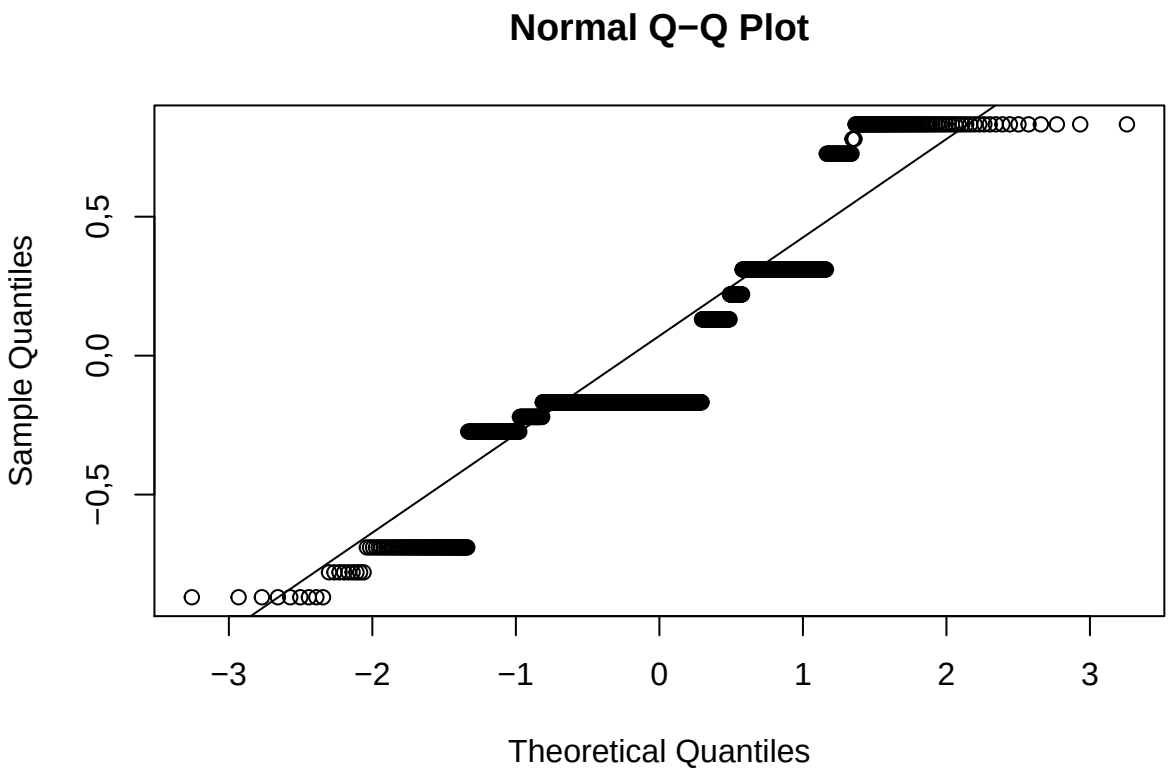

```
qqnorm(residuals(ie23))
qqline(residuals(ie23))
```



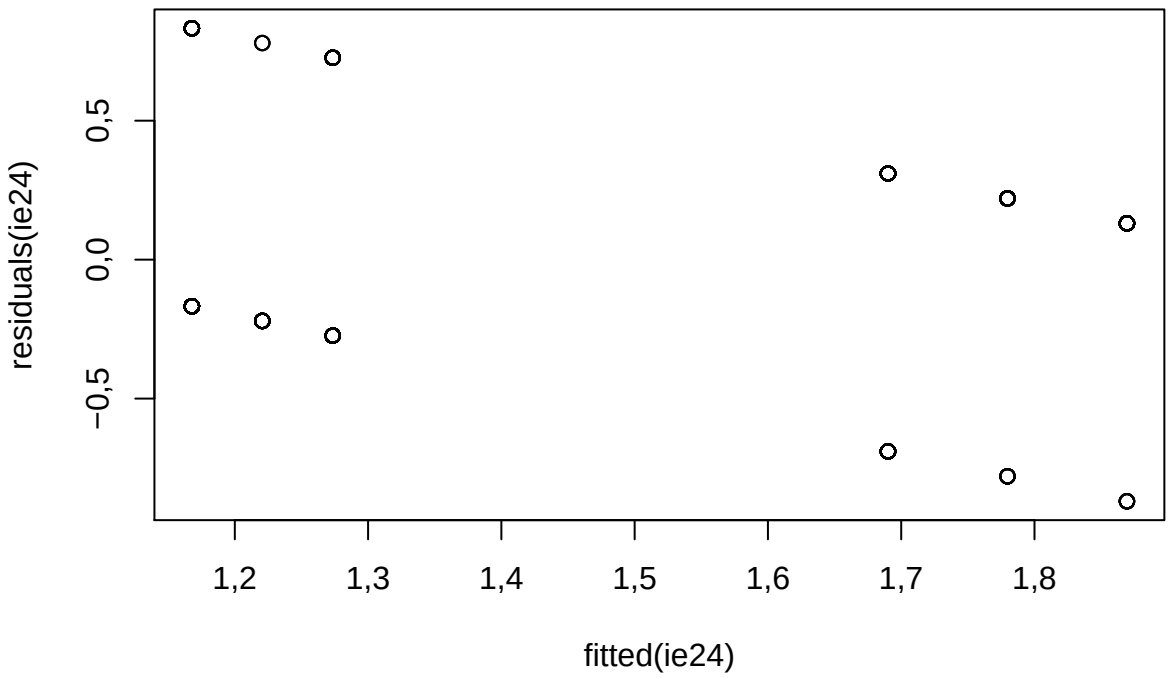

```
plot(fitted(ie23), residuals(ie23))
```



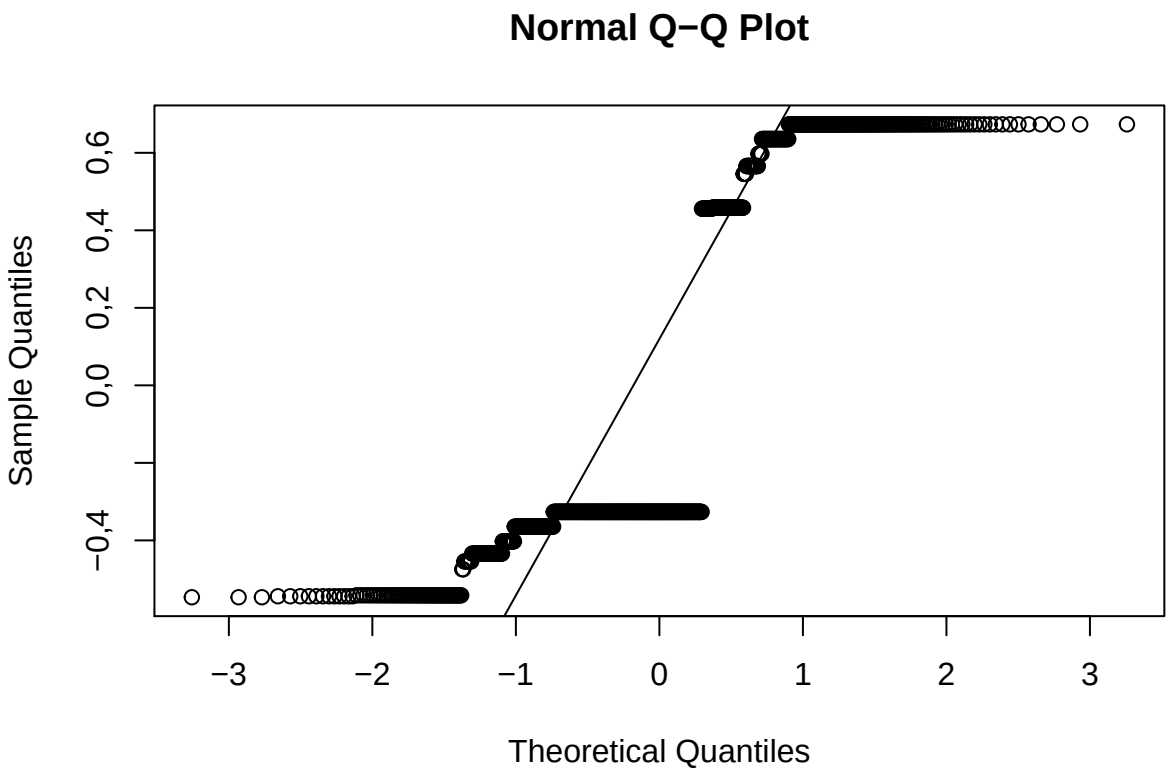

```
qqnorm(residuals(ie24))
qqline(residuals(ie24))
```



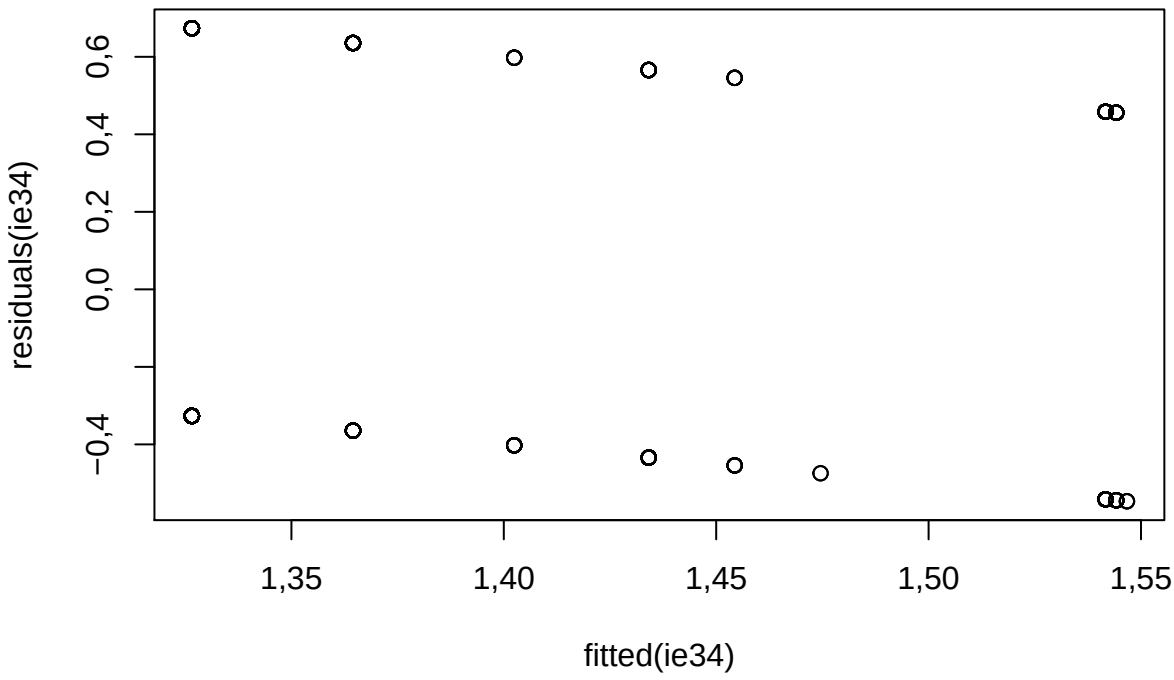

```
plot(fitted(ie24), residuals(ie24))
```




```
qqnorm(residuals(ie34))
qqline(residuals(ie34))
```




```
plot(fitted(ie34), residuals(ie34))
```




## 4.4 Evaluación de algoritmos

Intentaremos algunos algoritmos lineales y no lineales: * Algoritmos Lineales: Regresión Logística (LG), Análisis Lineal Discriminado (LDA) y Regresión Logística Regularizada (GLMNET). * Algoritmos no lineales: k-Nearest Neighbors (KNN), árboles de clasificación y regresión (CART), Naive Bayes (NB) y máquinas de vectores de soporte con funciones de base radial (SVM).

### 4.4.1 Selección y verificación de variables

A partir de los valores de muestra que se muestran en el conjunto de datos, podemos ver que PassengerId, Name, Ticket, Cabin y Embarked probablemente no serán útiles en nuestro análisis, por lo que decidimos que podemos eliminar estas variables.

```
# se eliminan variables redundantes
datasetTrain <- titanic_train[,c(-3, -8, -10, -11, -c(12:23))]

datasetTrain$Pclass = as.integer(datasetTrain$Pclass)
datasetTrain$Age = as.integer(datasetTrain$Age)
#datasetTrain$Survived = as.integer(datasetTrain$Survived)
# sumario
summary(datasetTrain)
```

```
##  Survived     Pclass          Sex           Age            SibSp
##  0:549    Min.   :1,000   female:314   Min.   : 0,00   Min.   :0,000
##  1:342    1st Qu.:2,000   male  :577   1st Qu.:21,00   1st Qu.:0,000
```

```
##            Median :3,000                Median :28,00   Median :0,000
##            Mean   :2,309                Mean   :29,72   Mean   :0,523
##            3rd Qu.:3,000                3rd Qu.:38,00   3rd Qu.:1,000
##            Max.   :3,000                Max.   :80,00   Max.   :8,000
##      Parch             Fare
##  Min.   :0,0000   Min.   :  0,00
##  1st Qu.:0,0000   1st Qu.:  7,91
##  Median :0,0000   Median : 14,45
##  Mean   :0,3816   Mean   : 32,20
##  3rd Qu.:0,0000   3rd Qu.: 31,00
##  Max.   :6,0000   Max.   :512,33
```

```
sapply(datasetTrain, class)
```

```
##  Survived    Pclass       Sex       Age     SibSp     Parch      Fare
##  "factor" "integer"  "factor" "integer" "integer" "integer" "numeric"
```

Echemos un vistazo más de cerca a las distribuciones de clase.

```
# distribución de clases
cbind(freq=table(datasetTrain$Survived), percentage=prop.table(table(datasetTrain$Survived))*100)
```

```
##   freq percentage
## 0  549   61,61616
## 1  342   38,38384
```

Hay algún desequilibrio en los valores de la clase. Observamos una división aproximada de 60% a 40% para Died (= 0) y Survived (= 1) en los valores de clase.

```
datasetTrain[,3] <- as.numeric((datasetTrain[,3]))
complete_cases <- complete.cases(datasetTrain)

kable(cor(datasetTrain[complete_cases,2:5]), caption = "Correlación del conjunto de datos",digits = 3, p
```

Table 10: Correlación del conjunto de datos

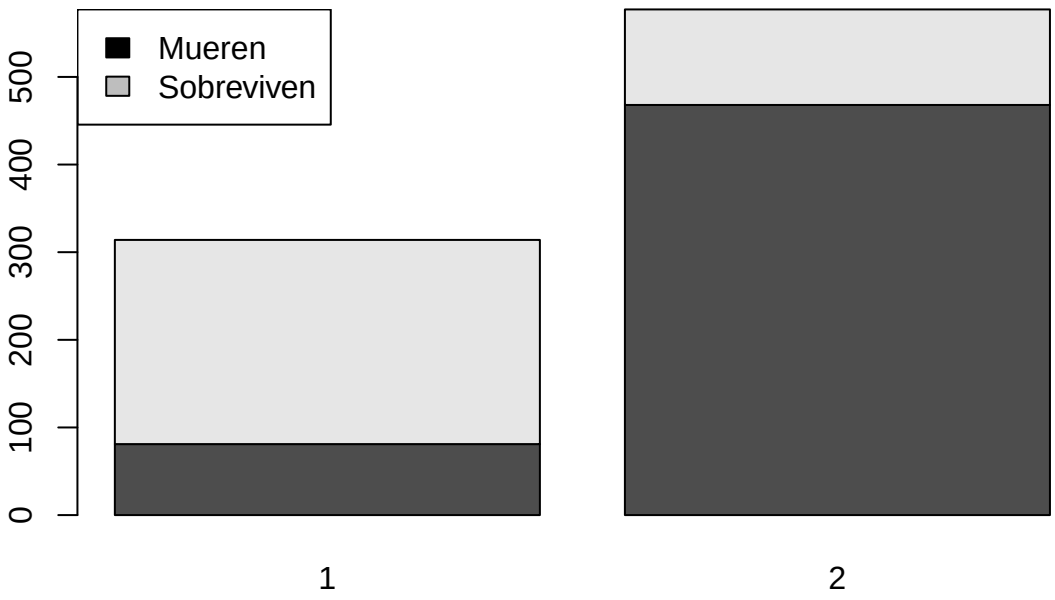
| | Pclass | Sex | Age | SibSp |
|---|---|---|---|---|
| Pclass | 1,000 | 0,132 | -0,351 | 0,083 |
| Sex | 0,132 | 1,000 | 0,104 | -0,115 |
| Age | -0,351 | 0,104 | 1,000 | -0,241 |
| SibSp | 0,083 | -0,115 | -0,241 | 1,000 |

```
#cor(datasetTrain[complete_cases,2:5])
```

Los valores por encima de 0,75 o por debajo de -0,75 son indicativos de una correlación alta positiva o alta negativa. A partir de los resultados anteriores, las variables no están altamente correlacionadas en este conjunto de datos.

### 4.4.2 Visualizacion del conjunto de datos

```
# barra de hombres y mujeres que sobrevivieron
barplot(table(datasetTrain$Survived, datasetTrain[,3]))
legend("topleft", legend = c("Mueren", "Sobreviven"), fill=c("black","grey"))
```




### 4.4.3 Opciones de prueba y métrica de evaluación.

Vamos a definir un test de control. Usaremos una validación cruzada de 10 veces con 3 repeticiones. Esta es una buena configuración de test de control estándar. Es un problema de clasificación binario. Para simplificar, utilizaremos métricas de precisión y Kappa.

```
FALSE corrplot 0.84 loaded
```

```
# 10-fold validación cruzada con 3 repeticiones
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "Accuracy"
```

### 4.4.4 Algoritmos de muestreo (Spot-Check Algorithms)

```
# LG
set.seed(7)
fit.glm <- train(Survived~., data=datasetTrain, method="glm", metric=metric,
                 trControl=trainControl)

# LDA
set.seed(7)
fit.lda <- train(Survived~., data=datasetTrain, method="lda", metric=metric,
                 trControl=trainControl)

# GLMNET
```

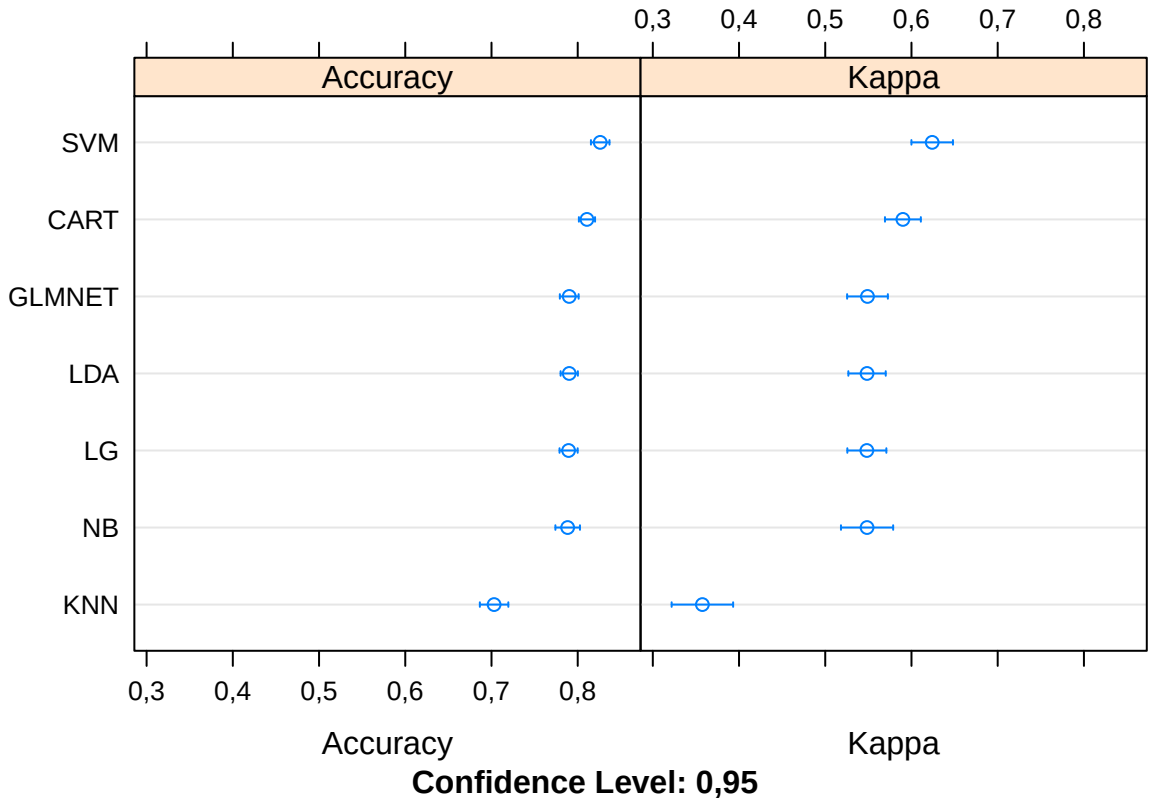
```r
set.seed(7)
fit.glmnet <- train(Survived~., data=datasetTrain, method="glmnet", metric=metric,
                    trControl=trainControl)

# KNN
set.seed(7)
fit.knn <- train(Survived~., data=datasetTrain, method="knn", metric=metric,
                 trControl=trainControl)

# CART
set.seed(7)
fit.cart <- train(Survived~., data=datasetTrain, method="rpart", metric=metric,
                  trControl=trainControl)

# Naive Bayes
set.seed(7)
fit.nb <- train(Survived~., data=datasetTrain, method="nb", metric=metric,
                trControl=trainControl)

# SVM
set.seed(7)
fit.svm <- train(Survived~., data=datasetTrain, method="svmRadial", metric=metric,
                 trControl=trainControl)
```

```r
# Comparar algorithms
results <- resamples(list(LG=fit.glm, LDA=fit.lda, GLMNET=fit.glmnet, KNN=fit.knn,
    CART=fit.cart, NB=fit.nb, SVM=fit.svm))
summary(results)
```

```
FALSE
FALSE Call:
FALSE summary.resamples(object = results)
FALSE
FALSE Models: LG, LDA, GLMNET, KNN, CART, NB, SVM
FALSE Number of resamples: 30
FALSE
FALSE Accuracy
FALSE             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
FALSE LG     0,7303371 0,7752809 0,7933208 0,7894128 0,8089888 0,8444444    0
FALSE LDA    0,7415730 0,7752809 0,7877029 0,7901492 0,8105805 0,8333333    0
FALSE GLMNET 0,7191011 0,7752809 0,7877029 0,7901661 0,8089888 0,8444444    0
FALSE KNN    0,6067416 0,6750624 0,7078652 0,7030181 0,7365615 0,7865169    0
FALSE CART   0,7752809 0,7888889 0,8089888 0,8107241 0,8222222 0,8666667    0
FALSE NB     0,7111111 0,7647004 0,7877029 0,7883480 0,8158836 0,8636364    0
FALSE SVM    0,7752809 0,8089888 0,8212235 0,8260763 0,8426966 0,8863636    0
FALSE
FALSE Kappa
FALSE             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
FALSE LG     0,4157549 0,5088855 0,5520819 0,5482583 0,5899019 0,6657825    0
FALSE LDA    0,4368638 0,5145115 0,5470603 0,5484747 0,5920085 0,6489613    0
FALSE GLMNET 0,3878955 0,5088855 0,5470603 0,5489880 0,5885233 0,6657825    0
FALSE KNN    0,1624092 0,2995518 0,3521837 0,3575056 0,4364379 0,5401142    0
FALSE CART   0,5074709 0,5461841 0,5861461 0,5900530 0,6210950 0,7194805    0
FALSE NB     0,4045802 0,4883741 0,5552968 0,5484930 0,6083261 0,7060134    0
```

```
FALSE SVM    0,5131291 0,5818325 0,6142702 0,6240252 0,6713877 0,7550111    0
```

```
dotplot(results)
```




SVM tiene la mayor precisión con un 82%.

### 4.4.5 Evaluación de los Algoritmos

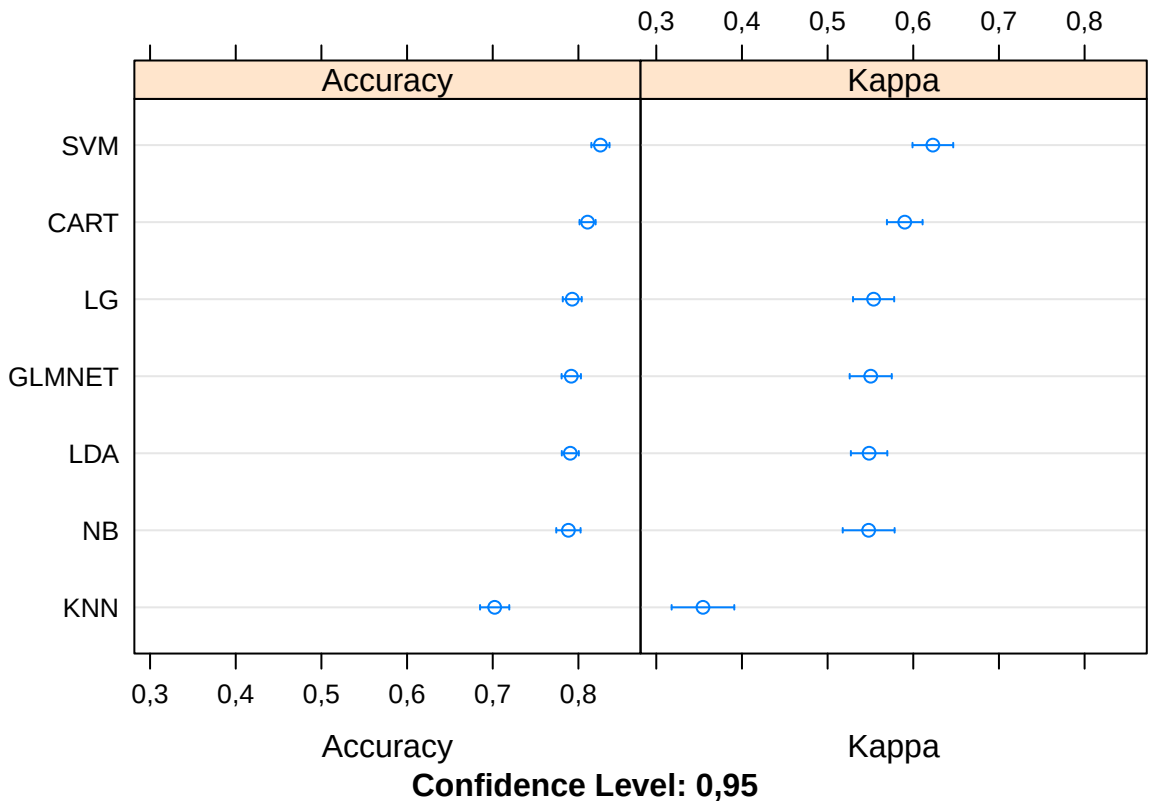
Aplicaríamos una transformación Box-Cox para aplanar la distribución.

```
# Comparar algorithms
# 10-fold validación cruzada con 3 repeticiones
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "Accuracy"
# LG
set.seed(7)
fit.glm <- train(Survived~., data=datasetTrain, method="glm", metric=metric, preProc=c("BoxCox"),
    trControl=trainControl)
# LDA
set.seed(7)
fit.lda <- train(Survived~., data=datasetTrain, method="lda", metric=metric, preProc=c("BoxCox"),
    trControl=trainControl)
# GLMNET
set.seed(7)
fit.glmnet <- train(Survived~., data=datasetTrain, method="glmnet", metric=metric,
    preProc=c("BoxCox"), trControl=trainControl)
# KNN
```

```r
set.seed(7)
fit.knn <- train(Survived~., data=datasetTrain, method="knn", metric=metric, preProc=c("BoxCox"),
    trControl=trainControl)
# CART
set.seed(7)
fit.cart <- train(Survived~., data=datasetTrain, method="rpart", metric=metric,
    preProc=c("BoxCox"), trControl=trainControl)
# Naive Bayes
set.seed(7)
fit.nb <- train(Survived~., data=datasetTrain, method="nb", metric=metric, preProc=c("BoxCox"),
    trControl=trainControl)
# SVM
set.seed(7)
fit.svm <- train(Survived~., data=datasetTrain, method="svmRadial", metric=metric,
    preProc=c("BoxCox"), trControl=trainControl)
# Compare algorithms
transformResults <- resamples(list(LG=fit.glm, LDA=fit.lda, GLMNET=fit.glmnet, KNN=fit.knn,
    CART=fit.cart, NB=fit.nb, SVM=fit.svm))
summary(transformResults)
```

```
FALSE
FALSE Call:
FALSE summary.resamples(object = transformResults)
FALSE
FALSE Models: LG, LDA, GLMNET, KNN, CART, NB, SVM
FALSE Number of resamples: 30
FALSE
FALSE Accuracy
FALSE             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
FALSE LG     0,7191011 0,7752809 0,7977528 0,7928005 0,8158836 0,8444444    0
FALSE LDA    0,7415730 0,7752809 0,7865169 0,7905278 0,8084461 0,8333333    0
FALSE GLMNET 0,7191011 0,7752809 0,7933208 0,7916726 0,8158836 0,8444444    0
FALSE KNN    0,5842697 0,6827120 0,6944444 0,7022608 0,7365615 0,7752809    0
FALSE CART   0,7752809 0,7888889 0,8089888 0,8107241 0,8222222 0,8666667    0
FALSE NB     0,7191011 0,7647004 0,7865169 0,7883481 0,8158836 0,8636364    0
FALSE SVM    0,7865169 0,8089888 0,8156679 0,8257059 0,8426966 0,8863636    0
FALSE
FALSE Kappa
FALSE             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
FALSE LG     0,3878955 0,5131291 0,5567239 0,5536862 0,5945812 0,6657825    0
FALSE LDA    0,4368638 0,5145115 0,5470603 0,5484341 0,5853110 0,6489613    0
FALSE GLMNET 0,3878955 0,5131291 0,5527683 0,5503366 0,6031442 0,6705447    0
FALSE KNN    0,1244350 0,3043785 0,3393517 0,3544966 0,4267551 0,5186587    0
FALSE CART   0,5074709 0,5461841 0,5861461 0,5900530 0,6210950 0,7194805    0
FALSE NB     0,4084020 0,4883741 0,5495991 0,5479227 0,6083261 0,7060134    0
FALSE SVM    0,5237961 0,5818325 0,6055127 0,6228628 0,6713877 0,7550111    0
```

```r
dotplot(transformResults)
```




La precisión de SVM aumentó ligeramente al 83%. Aún no es lo suficientemente bueno.

## 4.5 Mejorar la precisión

Vamos ahora a probar un poco de ajuste de los mejores algoritmos, específicamente SVM y veamos si podemos aumentar la precisión.

### 4.5.1 Afinación del algoritmo

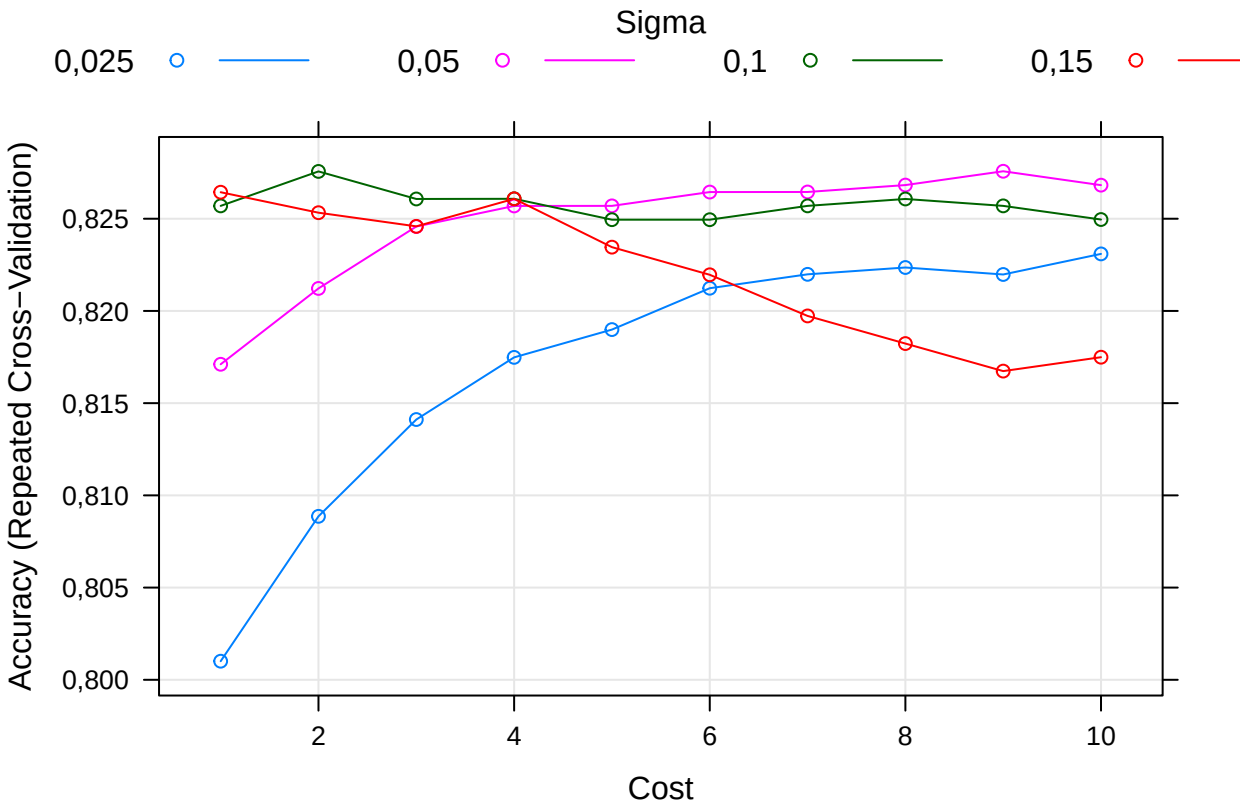
La implementación SVM tiene dos parámetros que podemos sintonizar con el paquete caret. El sigma, que es un término de suavizado, y C, que es una restricción de costos.

```
# 10-fold validación cruzada con 3 repeticiones
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "Accuracy"
set.seed(7)
grid <- expand.grid(.sigma=c(0.025, 0.05, 0.1, 0.15), .C=seq(1, 10, by=1))
fit.svm <- train(Survived~., data=datasetTrain, method="svmRadial", metric=metric, tuneGrid=grid,
    preProc=c("BoxCox"), trControl=trainControl)
print(fit.svm)
```

```
FALSE Support Vector Machines with Radial Basis Function Kernel
FALSE
FALSE 891 samples
FALSE   6 predictor
```

```
FALSE   2 classes: '0', '1'
FALSE
FALSE Pre-processing: Box-Cox transformation (1)
FALSE Resampling: Cross-Validated (10 fold, repeated 3 times)
FALSE Summary of sample sizes: 802, 801, 802, 802, 802, 801, ...
FALSE Resampling results across tuning parameters:
FALSE
FALSE   sigma  C   Accuracy   Kappa
FALSE   0,025   1  0,8010067  0,5686501
FALSE   0,025   2  0,8088639  0,5853700
FALSE   0,025   3  0,8141118  0,5962642
FALSE   0,025   4  0,8174828  0,6041957
FALSE   0,025   5  0,8189894  0,6075715
FALSE   0,025   6  0,8212325  0,6126001
FALSE   0,025   7  0,8219857  0,6143266
FALSE   0,025   8  0,8223561  0,6152172
FALSE   0,025   9  0,8219774  0,6141708
FALSE   0,025  10  0,8230926  0,6166517
FALSE   0,050   1  0,8171083  0,6034461
FALSE   0,050   2  0,8212199  0,6126616
FALSE   0,050   3  0,8245782  0,6202044
FALSE   0,050   4  0,8256935  0,6224361
FALSE   0,050   5  0,8256977  0,6224621
FALSE   0,050   6  0,8264468  0,6242690
FALSE   0,050   7  0,8264509  0,6244232
FALSE   0,050   8  0,8268255  0,6253247
FALSE   0,050   9  0,8275745  0,6270559
FALSE   0,050  10  0,8268212  0,6252690
FALSE   0,100   1  0,8256976  0,6229760
FALSE   0,100   2  0,8275661  0,6270675
FALSE   0,100   3  0,8260721  0,6235029
FALSE   0,100   4  0,8260804  0,6238684
FALSE   0,100   5  0,8249484  0,6213821
FALSE   0,100   6  0,8249484  0,6215145
FALSE   0,100   7  0,8256974  0,6229742
FALSE   0,100   8  0,8260720  0,6237222
FALSE   0,100   9  0,8256974  0,6228040
FALSE   0,100  10  0,8249567  0,6208324
FALSE   0,150   1  0,8264383  0,6244221
FALSE   0,150   2  0,8253271  0,6219698
FALSE   0,150   3  0,8245864  0,6203671
FALSE   0,150   4  0,8260804  0,6235865
FALSE   0,150   5  0,8234585  0,6173214
FALSE   0,150   6  0,8219602  0,6144546
FALSE   0,150   7  0,8197297  0,6095955
FALSE   0,150   8  0,8182316  0,6065184
FALSE   0,150   9  0,8167376  0,6032692
FALSE   0,150  10  0,8174908  0,6047890
FALSE
FALSE Accuracy was used to select the optimal model using the largest value.
FALSE The final values used for the model were sigma = 0,05 and C = 9.
```

```
plot(fit.svm)
```




### 4.5.2 Conjuntos

Veamos 4 métodos de conjunto:

- **Bagging**: contenedor CART (BAG) y random forest (RF).
- **Boosting**: aumento de gradiente estocástico (GBM) y C5.0 (C50).

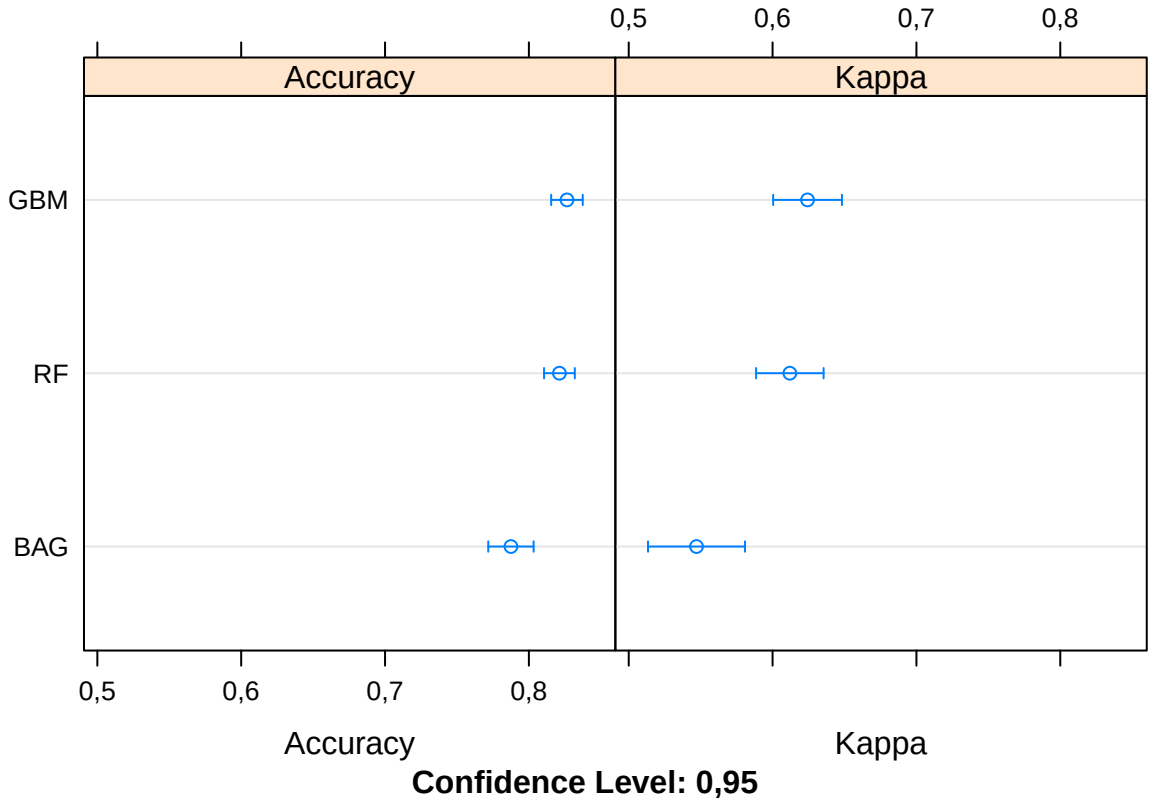
Utilizaremos la misma prueba de control que anteriormente, incluida la transformación de Box-Cox que aplana las distribuciones.

```
# 10-fold validación cruzada con 3 repeticiones
trainControl <- trainControl(method="repeatedcv", number=10, repeats=3)
metric <- "Accuracy"

# Bagged CART
set.seed(7)
fit.treebag <- train(Survived~., data=datasetTrain, method="treebag", metric=metric,
    trControl=trainControl)

# Random Forest
set.seed(7)
fit.rf <- train(Survived~., data=datasetTrain, method="rf", metric=metric, preProc=c("BoxCox"),
    trControl=trainControl)

# Stochastic Gradient Boosting
set.seed(7)
```

```
fit.gbm <- train(Survived~., data=datasetTrain, method="gbm", metric=metric, preProc=c("BoxCox"),
    trControl=trainControl, verbose=FALSE)

# C5.0
#set.seed(7)
#fit.c50 <- train(Survived~., data=datasetTrain, method="C5.0", metric=metric, preProc=c("BoxCox"),
#    trControl=trainControl)

# Compare results
#ensembleResults <- resamples(list(BAG=fit.treebag, RF=fit.rf, GBM=fit.gbm, C50=fit.c50))
ensembleResults <- resamples(list(BAG=fit.treebag, RF=fit.rf, GBM=fit.gbm))
summary(ensembleResults)
```

```
FALSE
FALSE Call:
FALSE summary.resamples(object = ensembleResults)
FALSE
FALSE Models: BAG, RF, GBM
FALSE Number of resamples: 30
FALSE
FALSE Accuracy
FALSE          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
FALSE BAG 0,7191011 0,7556180 0,7865169 0,7875232 0,8105805 0,8777778    0
FALSE RF  0,7444444 0,8089888 0,8202247 0,8212153 0,8314607 0,9000000    0
FALSE GBM 0,7666667 0,8000000 0,8212235 0,8264590 0,8535176 0,8988764    0
FALSE
FALSE Kappa
FALSE          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
FALSE BAG 0,3732591 0,4823603 0,5436216 0,5471214 0,6102393 0,7387863    0
FALSE RF  0,4595300 0,5763290 0,6059768 0,6120015 0,6410325 0,7862797    0
FALSE GBM 0,4960000 0,5709815 0,6149270 0,6243005 0,6765872 0,7821594    0
```

```
dotplot(ensembleResults)
```




### 4.5.3 Finalizar el modelo

Vamos a finalizar el modelo de SVM para usar en todo nuestro conjunto de entrenamiento.

Tendremos que eliminar las filas con valores perdidos del conjunto de datos de entrenamiento, así como el conjunto de datos de validación. El siguiente código muestra la preparación de los parámetros de preprocesamiento utilizando el conjunto de datos de entrenamiento.

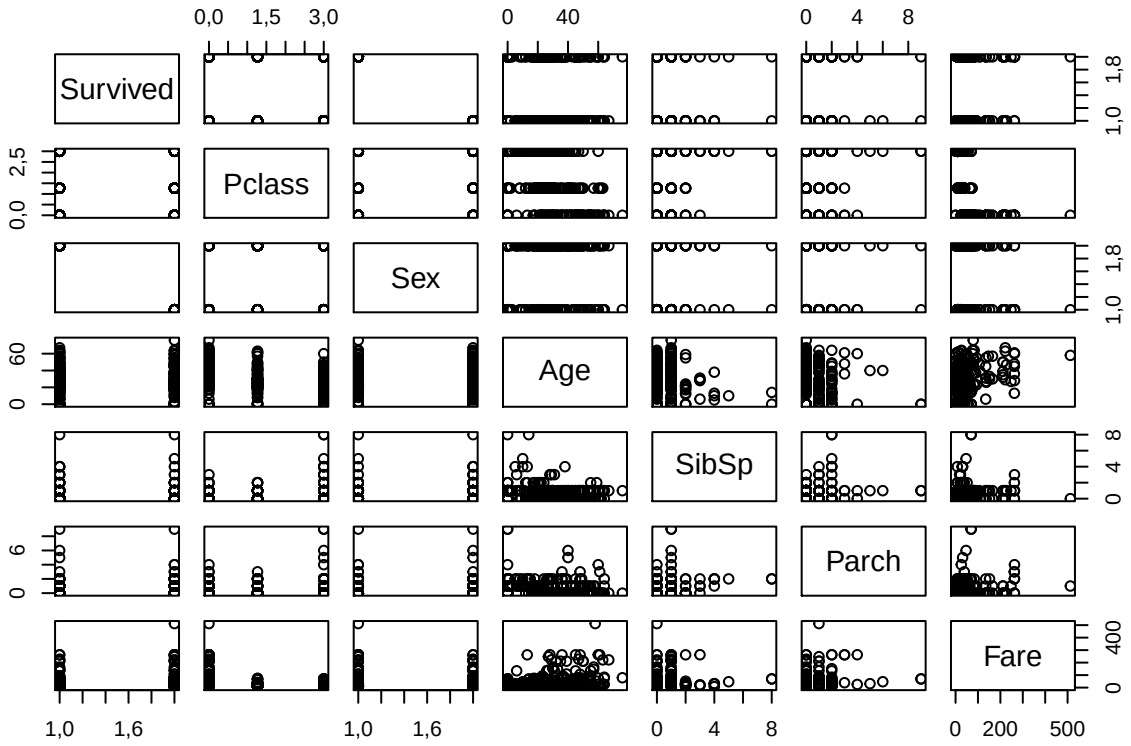
```
# preparar parámetros para la transformación de datos
# set.seed(7)
model <- svm(Survived ~ ., data = datasetTrain)

# se eliminan variables redundantes
datasetTrain <- titanic_test[,c(-3, -8, -10, -11, -c(12:23))]

# Se ajustan los datos como en el conjunto de entrenamiennto
datasetTest <- titanic_test
testData <- datasetTest[,c(-1, -4, -9, -11, -12)]

testData$Pclass = as.integer(testData$Pclass)
testData$Age = as.integer(testData$Age)
testData$Sex <- as.numeric(testData$Sex)
testData$Survived <- as.factor(testData$Survived)

preprocessParams <- preProcess(testData, method=c("BoxCox"))
testData$Age[is.na(testData$Age)] <- 0
```

```
testData$Fare[is.na(testData$Fare)] <- 0
testData <- predict(preprocessParams, testData)

predictions <- predict(model, testData, type="class")
submit <- data.frame(PassengerId = datasetTest$PassengerId, Survived = predictions)

# ajustamos la salidad de los resultados
csv_dir = paste(baseDirectory, "/", "resultados", sep="")
setwd(csv_dir)
write.csv(submit, file = "firstSVM.csv", row.names = FALSE)
# retornamos al directorio para trabajar con el shp
setwd(baseDirectory)
```

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

Datos del test:

```
plot(testData)
```




```
#plot(predictions)
```

```
predictedTest_mg = merge(x = submit, y = datasetTest,by.x="PassengerId", by.y =  "PassengerId")
predicted_names = names(predictedTest_mg)
predicted_names[2]="Surv. Predicted"
predicted_names[3]="Surv. Original"
```

```
names(predictedTest_mg) = predicted_names

wrongSurvivedPred = predictedTest_mg[predictedTest_mg$`Surv. Predicted`!=predictedTest_mg$`Surv. Origina
successSurvivedPred = predictedTest_mg[predictedTest_mg$`Surv. Predicted`==predictedTest_mg$`Surv. Orig

datatable(wrongSurvivedPred)
```

Show 10 entries Search:

| | PassengerId | Surv. Predicted | Surv. Original | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 22 | 913 | 1 | 0 | 3 | Olsen, Master. Artur Karl | male | 9 | 0 | 1 | C 17368 | 3.1708 | | S |
| 33 | 924 | 0 | 1 | 3 | Dean, Mrs. Bertram (Eva Georgetta Light) | female | 33 | 1 | 2 | C.A. 2315 | 20.575 | | S |
| 70 | 961 | 0 | 1 | 1 | Fortune, Mrs. Mark (Mary McDougald) | female | 60 | 1 | 4 | 19950 | 263 | C23 C25 C27 | S |
| 81 | 972 | 1 | 0 | 3 | Boulos, Master. Akar | male | 6 | 1 | 1 | 2678 | 15.2458 | | C |
| 90 | 981 | 1 | 0 | 2 | Wells, Master. Ralph Lester | male | 2 | 1 | 1 | 29103 | 23 | | S |
| 133 | 1024 | 0 | 1 | 3 | Lefebre, Mrs. Frank (Frances) | female | | 0 | 4 | 4133 | 25.4667 | | S |
| 141 | 1032 | 0 | 1 | 3 | Goodwin, Miss. Jessie Allis | female | 10 | 5 | 2 | CA 2144 | 46.9 | | S |
| 162 | 1053 | 1 | 0 | 3 | Touma, Master. Georges Youssef | male | 7 | 1 | 1 | 2650 | 15.2458 | | C |
| 189 | 1080 | 0 | 1 | 3 | Sage, Miss. Ada | female | | 8 | 2 | CA. 2343 | 69.55 | | S |
| 195 | 1086 | 1 | 0 | 2 | Drew, Master. Marshall Brines | male | 8 | 0 | 2 | 28220 | 32.5 | | S |

Showing 1 to 10 of 21 entries Previous 1 2 3 Next

La anterior es una tabla con los resultados que no se aciertan con los esperados del conjunto de test.

```
length(wrongSurvivedPred$`Surv. Predicted`)/length(predictedTest_mg$`Surv. Predicted`)
```

```
## [1] 0,05023923
```

```
success = length(successSurvivedPred$`Surv. Predicted`)/length(predictedTest_mg$`Surv. Predicted`)
fail = length(wrongSurvivedPred$`Surv. Predicted`)/length(predictedTest_mg$`Surv. Predicted`)

df.pred.results = data.frame(success = success, fail = fail)
```

Se muestran los porcentajes de acierto sobre los resultados esperados:

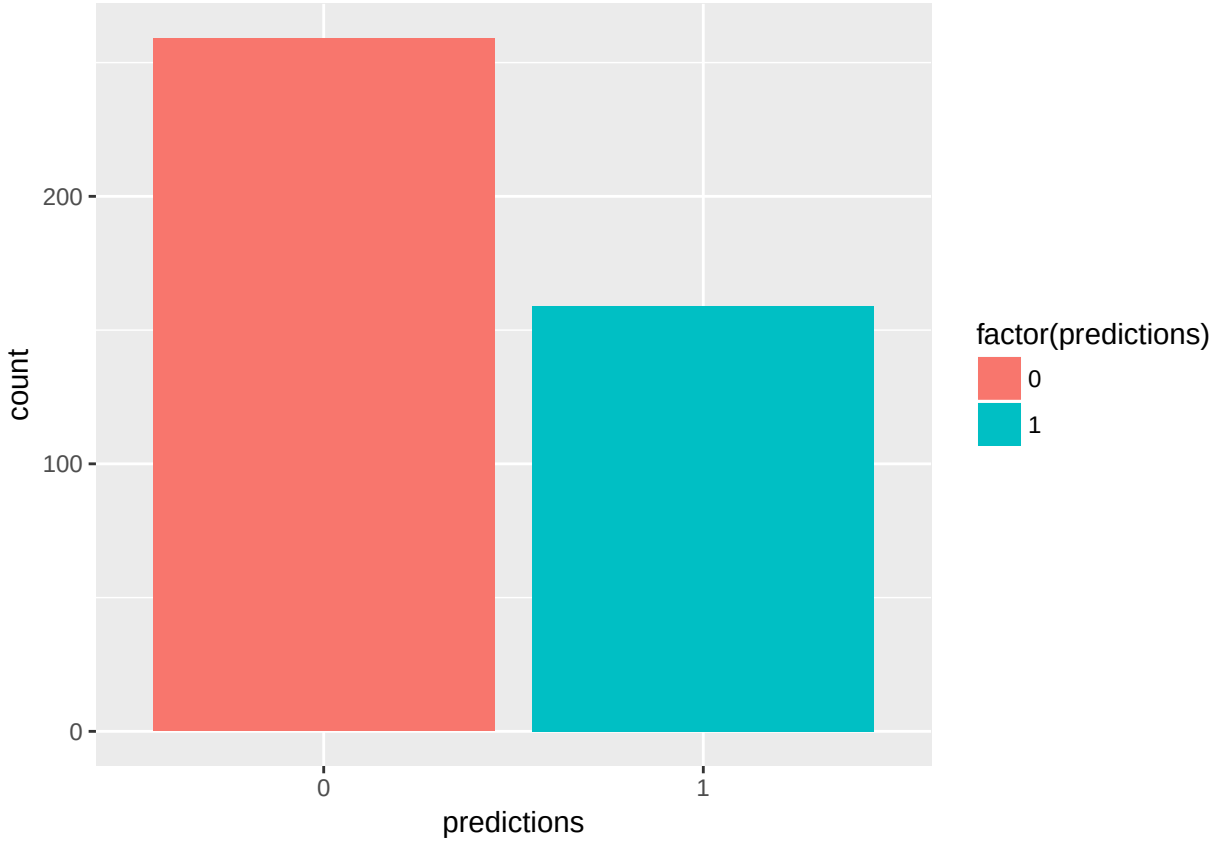
Table 11: Porcentajes de resultados de acierto en la predicción con SVM

| success | fail |
|---|---|
| 0,95 | 0,05 |

Como se puede ver, el acierto esta por encima del 94%, aunque siempre es posible mejorarlo.

Ahora mostramos una distribución de las prediciones finales:

```
ggplot(as.data.frame(as.integer(predictions)) , aes(x = predictions, fill = factor(predictions))) +
  geom_bar(stat='count', position='dodge')
```




# 6 Resolución del problema

A partir de los resultado obtenidos podemos indicar que de todos los modelos analizados, el modelo utilizado de SVM permite acercarse a un porcentaje de resolución del 95% en el acierto de los resultados de supervivencia de los pasajeros del Titanic. Según todas la pruebas estadísticas realziadas, el SVM era el que mejor respondia sobre los diversos modelos propuestos para la resolución del problema.

Los resultados permiten responder los planteamientos inciales del problema, a saber, que con los datos aportados por el dataset se pueden extraer conclusiones sobre la posibilidad de supervivencia de los pasajeros en función de las variables aportadas.

Como se ha podido comprobar, no se ha utilizado una selección a priori de las variables a utilizar para abordar la resolución del problema, ya que se ha considerado más prudente observar el comportamiento de las variables, su ajuste en el proceso de limpieza y análisis, y el posterior comportamiento en las métricas de resultados segun se han ido realizando pruebas con los distintos modelos.

# 7 Código en R

Este modelo de resolución de supervivencia del titanic se encuentra en el repositorio GitHub, en la localización: https://github.com/rgarciarui/titanicDataClean