Práctica 2: Limpieza y análisis de datos

Aleix Cortina, Kilian Cañizares

09/06/2020

Contents

1. Descripción del Dataset

En esta práctica se va a analizar el juego de datos del Titanic que se puede encontrar en:

https://www.kaggle.com/c/titanic

y cuyos datos se encuentran en el fichero *"train.csv"*.

Este es un conjunto de datos referente a los pasajeros del Titanic que se compone de un total de 12 variables que a continuación se describen:

- **PassagerId**: Identifica a cada pasejero por un identificador. Formato entero.
- **Survived**: Especifica si un pasajero sobrevivió (1) o no (0). Formato entero.
- **Pclass**: Indica la clase en que un pasajero viajaba. Formato entero.
- **Name**: Nombre del pasajero. Formato factor.
- **Sex**: Sexo del pasajero. Formato factor.
- **Age**: Edad del pasajero en años. Formato numérico.

- **SibSp**: Número de familiares o cónyugues a bordo. Formato entero.
- **Parch**: Número padres e hijos a bordo. Formato entero.
- **Ticket**. Número de ticket del pasajero. Formato factor.
- **Fare**: Precio del ticket en dolares. Formato numérico.
- **Cabin**: Número de cabina del pasajero. Formato factor.
- **Embarked**: Lugar de embarque del pasajero. Formato factor.

Con un número total de 891 observaciones.

El objetivo de este conjunto de datos es analizar que determina la probabilidad de supervivencia de un pasajero. Si bien el hecho de predecir la supervivencia en el naufragio del Titanic no es relevante en la actuaidad, las mejoras derivadas del entrenamiento de este tipo de algoritmos pueden tener aplicaciones importantes en áreas como la salud (i.e. predecir la mortalidad en función de hábitos y variables físicas del paciente), seguros (i.e. predecir el valor de un seguro en función de las características de un cliente), entre otras.

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

```
# Carga del juego de datos
datos<-read.csv("../data/train.csv", sep = ",", header = TRUE)

# Estructura de los datos
str(datos)
```

```
## '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       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 581
##  $ 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     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

```
# Descipción de los datos
summary(datos)
```

```
##   PassengerId       Survived          Pclass     
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000  
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000  
##  Median :446.0   Median :0.0000   Median :3.000  
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309  
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000  
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000  
##                                                  
##                                     Name         Sex           Age       
##  Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.42  
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.12  
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.00  
##  Abelson, Mr. Samuel                   :  1                Mean   :29.70  
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00  
```

```
##  Adahl, Mr. Mauritz Nils Martin       :  1                Max.   :80.00
##  (Other)                              :885                NA's   :177
##      SibSp           Parch             Ticket         Fare
##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33
##                                   (Other) :852
##          Cabin     Embarked
##             :687    :  2
##  B96 B98    :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6         :  4   S:644
##  C22 C26    :  3
##  D          :  3
##  (Other)    :186
```

Como se puede observar, existen dos variables que tienen un valor único para cada observación: **PassengerId** y **Name**. Estas variables no se tendrán en cuenta en el análisis ya que al ser variables con valor único, no dan información diferencial para calcular la probabilidad de que un pasejero sobreviva. Por el momento se mantienen estas variables ya que pueden ser de ayuda para algún tratamiento de datos, pero para no sobre-especializar el algoritmo a nivel de la identidad de los pasajeros (nos interesa entender el comportamiento de las variables más genéricas) no se tendrán en cuenta para el análisis y se procederá con su eliminación antes de comenzar.

El resto de variables serán usadas para comprender de que depende la probabilidad de supervivencia. En primer lugar, se van a discretizar las variables **Pclass** referente a la clase donde viajó el pasajero, y **Survived** que se refiere a si sobrevivió o no, ya que ambas a pesar de estar en formato *int* se consideran categóricas.

```
# Discretización de variables.
datos$Survived<-as.factor(datos$Survived)
datos$Pclass<-as.factor(datos$Pclass)
```

3. Limpieza de datos

Es este apartado, además de limpiar los datos, se van a efectuar transformaciones en las variables para optimizar su uso en el análisis.

3.1. Ceros, elementos vacios y NA

Para evaluar la presencia de NA, elementos vacíos o ceros, se ejecutan las siguientes rutinas

```
# Presencia de valores NA
colSums(is.na(datos))
```

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

```
# Presencia de valores vacíos
colSums(datos=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0          NA
```

```
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

```
# Presencia de valores 0
colSums(datos==0)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         549           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##         608         678           0          15           0           0
```

La variable **Age** contiene un total de 177 valores NA. Además, las variables **Cabin** y **Embarked** tienen un total de 687 y 2 valores vacios respectivamente. Por otro lado, los ceros pueden ser un valor aceptable en variables como **Survived** indicando que el pasajero no sobrevivió, en **SibSp** indicando el número de familiares o cónyugues, o en **Parch** indicando el número de padres e hijos. Sin embargo, los 15 valores 0 de la variable **Fare** carecen de sentido, así que serán tratados como valores perdidos.

3.1.1. Valores vacíos y 0

Las variables a tratar en este apartado son **Fare**, **Cabin** y **Embarked**. Respecto a la variable **Cabin**, proporciona información de la cabina donde se alojó un pasajero. Los valores vacíos corresponden a pasajeros que no se han alojado en cabina. El hecho que esté alojado en cabina debería de estar relacionado con la clase en al que se aloja el pasajero, y por lo tanto seguramente con la probabilidad de supervivencia. Primero se comprobará mediante un test *Chi-cuadrado* si existe o no dependencia entre tener asignado un número de cabina y la supervivencia mediante el siguiente test de hipótesis:

$$H_0\text{: Las variables son independientes.}$$

$$H_1\text{: Las variables no son independientes.}$$

donde, H_0 representa la hipótesis nula y H_1 la hipótesis alternativa.

```
library(plyr)
# Discretización variable Cabin

cabin_cat <- as.factor(
  sapply(datos$Cabin, function (cabin) if (cabin=="") "without_cabin" else "with_cabin")
  )

chisq.test(datos$Survived, cabin_cat)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  datos$Survived and cabin_cat
## X-squared = 87.941, df = 1, p-value < 2.2e-16
```

Como se puede observar la $p < \alpha = 0.05$ y se puede concluir que las variables no son independientes, luego existe una relación entre la supervivencia y el hecho de tener asignada una cabina. Así que se categorizará esta variable con dos niveles: *with_cabin* y *without_cabin* y se usará para el análisis.

```
# Se crea la nueva varibale Cabin
datos$Cabin<-cabin_cat

# eliminamos la variable utilizada del entorno de trabajo
remove(cabin_cat)
```

Respecto a la variable **Fare**, existen 15 valores cuyo valor es 0, debido al bajo número de valores faltantes, se procederá a la imputación de datos a partir del método de los vecinos más próximos mediante la función `kNN()`. Se considera que las variables que pueden tener relación con el valor del billete son: Clase del billete (**Pclass**), si es o no de cabina (**Cabin**) y el lugar de embarque (**Embarked**). La variable **Embarked** tiene dos valores vacíos, pero ninguno de ellos coincide con los valores que queremos imputar.

```
library(VIM)
# Se crea el datFrame para imputar los valores
datos_a_usar <- subset(datos,
                       select = c("Pclass", "Embarked", "Cabin" ,"Fare"))

# Se pasan a NA los valores que queremos imputar
index_imputados_fare <- which(datos$Fare==0)
datos_a_usar$Fare[index_imputados_fare] <- NA

# Se escojen 3 vecinos más proximos
datos_imputados<-kNN(datos_a_usar,k=3)

# Se muestran los datos imputados
datos_imputados$Fare[index_imputados_fare]
```

```
##  [1]  7.9250 51.8625  7.9250 16.0000  7.9250 16.0000 16.0000 16.0000  7.9250
## [10] 47.1000 16.0000 16.0000 51.8625 51.8625 47.1000
```

```
# Se pasan al dataFrame original
datos$Fare <- datos_imputados$Fare

# eliminamos los conjuntos de datos que no vamos a usar más
remove(datos_a_usar, datos_imputados, index_imputados_fare)
```

La variable **Embarked** tiene 3 niveles diferentes, e informa del puerto en que embarcó el pasajero. Contiene dos valores vacios que al igual que en el caso de **Fare** serán imputados. Para esta imputación se usarán también las variables **Fare**, **Pclass** y **Cabin**, ya que se considera que hay una relación entre precio, si es cabina o no, la clase y el puerto de embarque.

```
library(gdata)
# Se crea el datFrame para imputar los valores
datos_a_usar <- subset(datos,
                       select = c("Pclass", "Embarked", "Cabin" ,"Fare"))

# Se pasan a NA los valores que queremos imputar
index_imputados_embarked<-which(datos$Embarked=="")
datos_a_usar$Embarked[index_imputados_embarked]<-NA

# Se escojen 3 vecinos más proximos
datos_imputados<-kNN(datos_a_usar,k=3)

# Se muestran los datos imputados
datos_imputados$Embarked[index_imputados_embarked]
```

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

```
# Se pasan al dataFrame original y se eliminan los niveles que ya no se usan (i.e. "")
datos$Embarked<-datos_imputados$Embarked
datos$Embarked<-drop.levels(datos$Embarked)
```

```
# eliminamos las variables no usadas posteriormente
remove(datos_a_usar, datos_imputados, index_imputados_embarked)
```

3.1.2. Valores NA

Existen un total de 177 valores NA que se han de estudiar para tomar una decisión a cerca de su tratamiento. Debido al gran numero con respecto al total (177 sobre 891) no se considera una buena opción eliminar los casos. Por lo tanto, primero se evaluará si dichos valores (nulos) tienen una relación con el hecho que el indidividuo sobreviva o no. Anterioremente, se ha visto que la clase determina en cierto modo las proabibilidades de sobrevivir o no, y posiblemente el hecho de disponer o no de la edad puede depender de la clase. Para ver si hay o no dependencia entre **Survived** y **Age** primero se categorizará la variable **Age** en función de si se especifica o no la edad, para posteriomente hacer un test *Chi-cuadrado* de dependencia de variables mediante la función `chisq.test()` donde se contrastan las siguiente hipótesis:

$$H_0\text{: Las variables son independientes.}$$

$$H_1\text{: Las variables no son independientes.}$$

donde, H_0 representa la hipótesis nula y H_1 la hipótesis alternativa.

```
edad_presente <- sapply(datos$Age, function (age) if (is.na(age))
                        "edad_no_especificada" else "edad_especificada")

# Creación de tabla de contingencia
table_prop<-table(edad_presente,datos$Survived)
# Test Chi cuadrado
chisq.test(table_prop)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table_prop
## X-squared = 7.106, df = 1, p-value = 0.007683
```

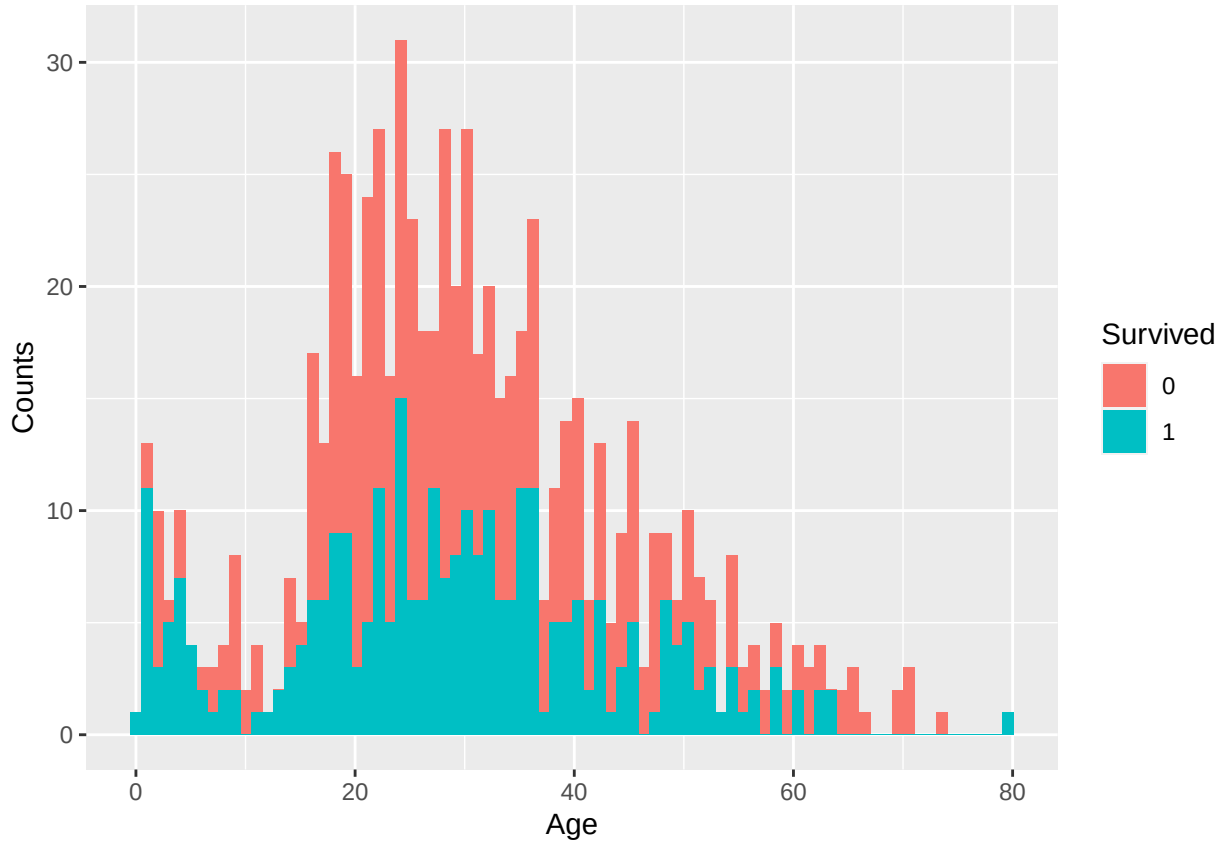
```
# eliminamos variables no usadas posteriormente
remove(edad_presente, table_prop)
```

Como se puede observar el p-valor $< \alpha = 0.05$, donde $\alpha = 0.05$ es el nivel de significación para un nivel de confianza del 95%, por lo tanto, rechazamos H_0 concluyendo que las variables son dependientes, luego existe una relación entre haber proporcionado la edad y la supervivencia.

Por ello, no se recomienda hacer ningún cálculo como puede ser la media para completar esta variable ya que le estariamos dando un valor promedio no beneficiandonos de su posible poder de predicción para el análisis e introducir error. Una posible solución es imputar los valores por medios probabilísticos a partir de la función `knn()`, y que los casos más cercanos en cuanto a otros atributos determinen el valor. Sin embargo, el gran número de casos de esta variable puede hacer que su gran numero de predicciones (177 sobre 891) la haga poco realista. Se optará por discretizar la variable en función de grupos de edad, y crear un grupo con estas valores con la etiqueta *no_age*. Para ello se ve si existe primero alguna relación entre edad y supervivencia, para posteriomente crear los grupos.

```
library(ggplot2)

age_survived_with_specified_age <- subset(
  datos[!is.na(datos$Age), ],
  select = c(Age, Survived)
)
```

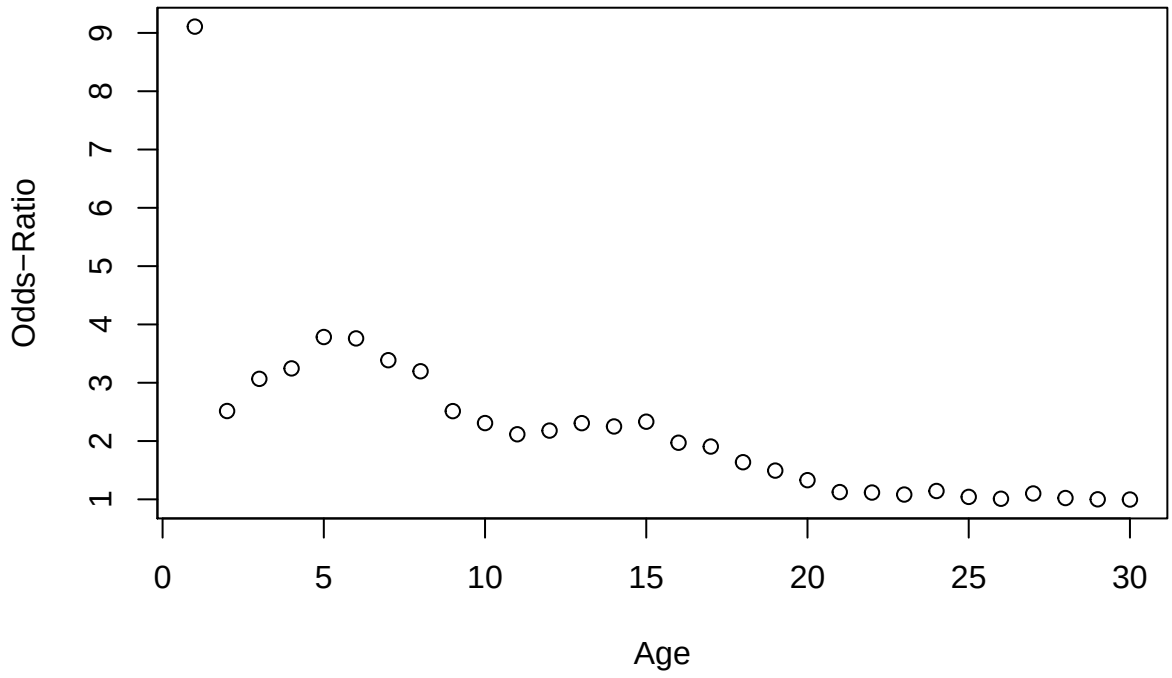
```r
# Grafico de conteos totales en
ggplot(data = age_survived_with_specified_age,
       aes(x=Age,  fill=Survived)
       )+geom_histogram(bins = 80)+ylab("Counts")+xlab("Age")
```



En el histograma de la variable **age** junto con la probabilidad de supervivencia es posible observar que a menos edad mayor tasa de supervivencia. Por lo tanto, nos encontramos ante un escenario dónde sabemos que la edad no especificada es dependiente con la variable **Survived**. Además, pertenecer al rango de menos edad significa tener más probabilidad de sobrevivir. Por lo tanto, para discretizar la variable se ha de escoger un valor óptimo que discretice la edad en grupos maximizando la información sobre la supervivencia o no del pasajero. Para ello se usará los *Odds-Ratio*, que nos medirá si es más probable sobrevivir a una edad con respecto al resto. Se van a escoger edades entre 1 y 30 como humbral de discrimanción y se va a hacer un gráfico con el resultado.

```r
survived_age<-relevel(age_survived_with_specified_age$Survived,ref="1")
result<-double(30)
for (i in 1:30){

  age_th <- as.factor(sapply(age_survived_with_specified_age$Age,
                  function (age) if (age>i) 0 else 1))

  age_th<-as.factor(age_th)
  age_th<-relevel(age_th,ref="1")
  tabla<-table(age_th,survived_age)

  or<-(tabla[1]*tabla[4])/(tabla[2]*tabla[3])
```

```
  result[i]<-or
}
plot(result, xlab ="Age", ylab="Odds-Ratio", yaxp  = c(0, 10, 10))
```



Como se puede observar, a medida que disminuye la edad aumenta la probabilidad de supervivencia, la cual se mantiene al doble para edades menores de 16 años. A partir de 21, se puede considerar que la edad no es significativa siendo la *Odds-ratio* similiar a 1. Escoger una edad muy baja, mejora la predicción , sin embargo, como se ve en el histograma, la mayoria de la poblacion esta concentrada en edades alrededor de 30 años. Por lo tanto, esta variable dejará de ser util para un segmento amplio de nuestro juego de datos. Escoger un valor muy alto de edad, puede hacer la variable mas útil para una mayor parte del juego de datos pero con menor capacidad predictiva. Por ello se escogera un valor de compromiso, considerando como que el doble de probabilidad de supervivencia es un buen indicador y además incluye a un amplio segmento de la población joven. Por lo tanto, para discretizar la variable se usará 16 años como limite entre niño y adulto. Teniendo en cuenta esto, la variable **Age** se discretizará de la siguiente manera para maximizar la información entre los grupos.

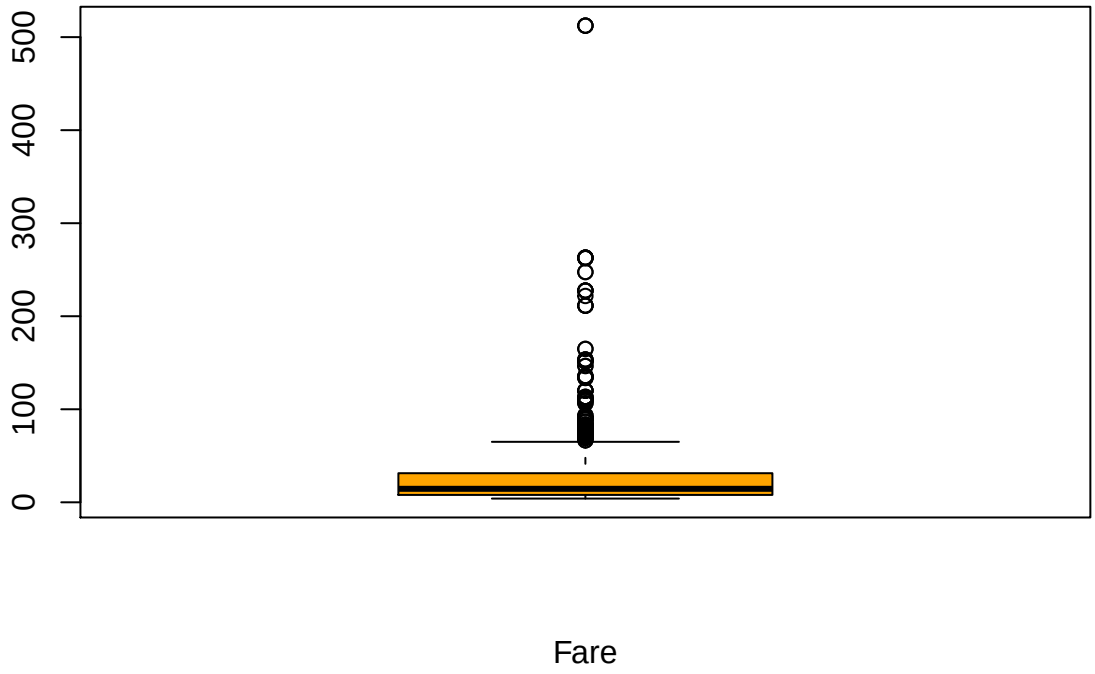
```
## Grupos niño, adulto y sin especificar
categorizar_edad <- function(age) {
  if (is.na(age)) {
    "no_age"
  }
  else if (age <= 16){
    "child"
  }
  else {
    "adult"
```

```
  }
}

datos$Age<-as.factor(mapply(categorizar_edad,
                            age=datos$Age))
```

3.2. Identificación y tratamiento de valores extremos.

El análisis de los valores extremos, se va a efectuar sobre las variables continuas que son: **Fare**, **SibSp** y **Parch**.

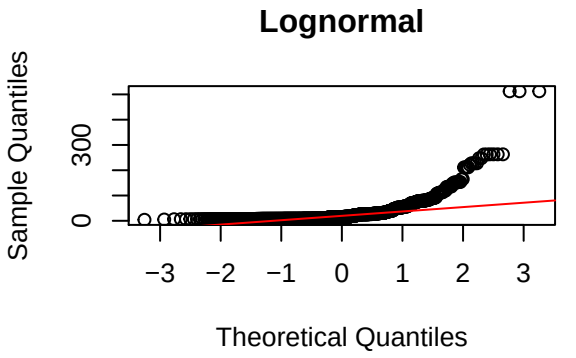
```
library(DescTools)

boxplot(datos$Fare, col="orange", xlab="Fare")
```

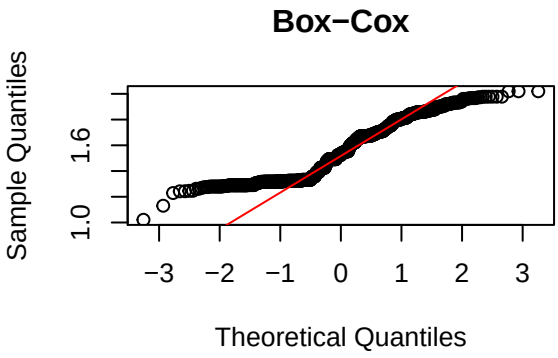


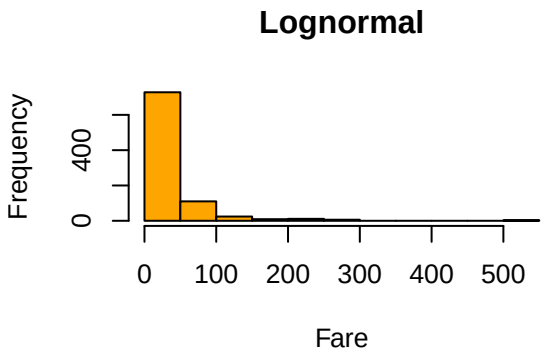
Debido a la naturaleza de la variable **Fare**, predominan los precios bajos y, por lo tanto, sigue una distribución lognormal lo que va a provocar una gran cantidad de valores extremos asociados a tickets más caros debido a las condiciones de compra (i.e. cabina, embarque...). Mediante la transformación de Box Cox se intenta mejorar la normalidad y homocedasticidad, lo que llevará a una menor proporción de valores extremos.

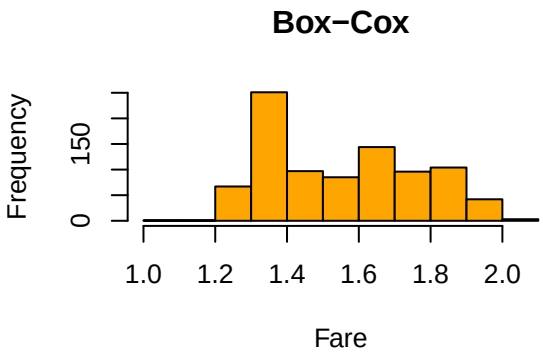
```
x<-datos$Fare
x.norm<- BoxCox(x, lambda = BoxCoxLambda(x))

par(mfrow=c(2,2))

qqnorm(x, main="Lognormal")
qqline(x,col=2)
```

```
qqnorm(x.norm, main="Box-Cox")
qqline(x.norm,col=2)

hist(x,main="Lognormal", xlab = "Fare", col="orange")
hist(x.norm, main="Box-Cox", xlab = "Fare", col="orange")
```





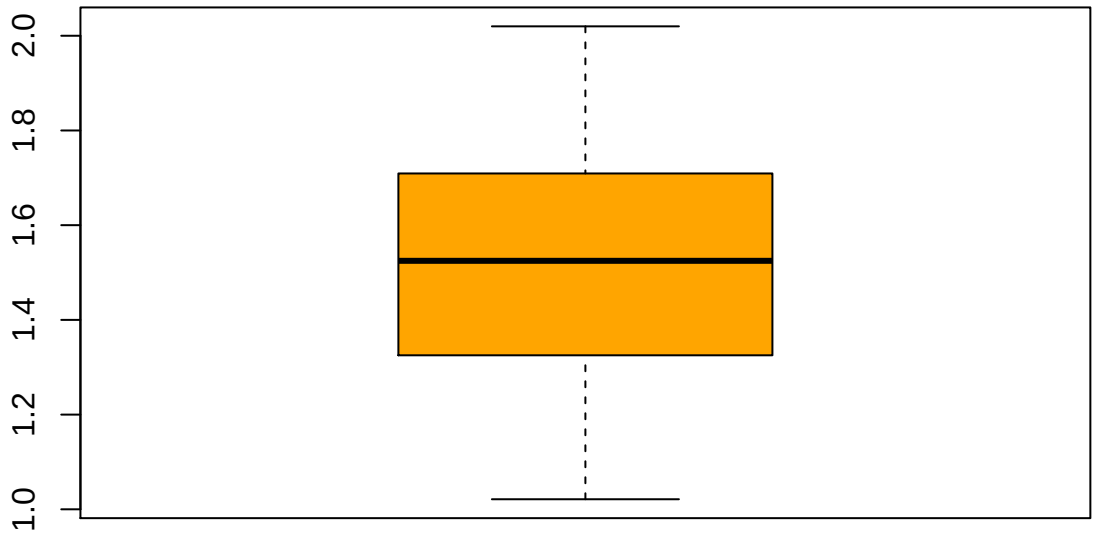





```
# Se introducen los datos en el dataFrame

datos$Fare<-x.norm
```

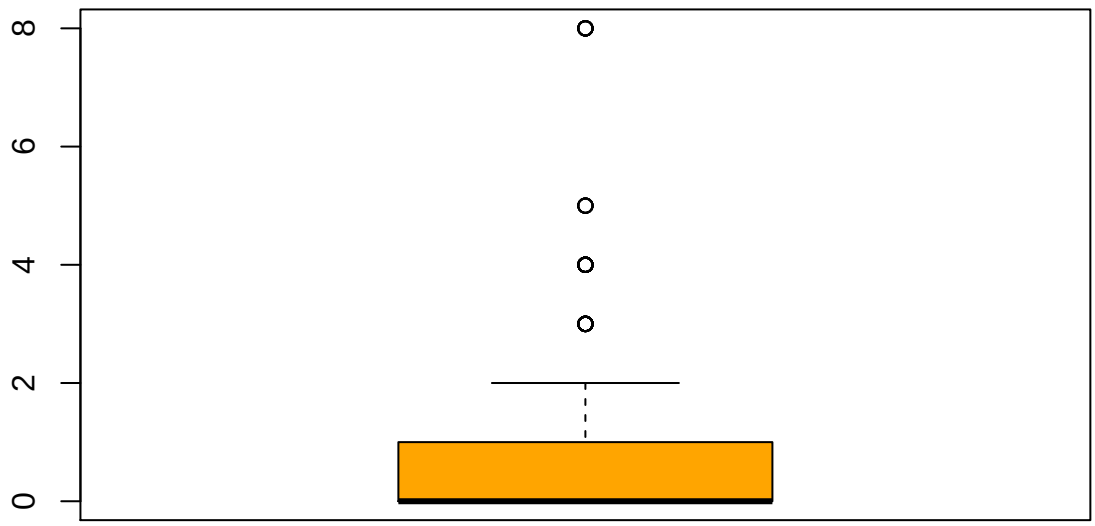
Despues de la transformación, ya no se observa la presencia de *outliers* los cuales habían sido previamente aceptados por la naturalidad de los datos:

```
boxplot(datos$Fare, col="orange", xlab="Fare")
```

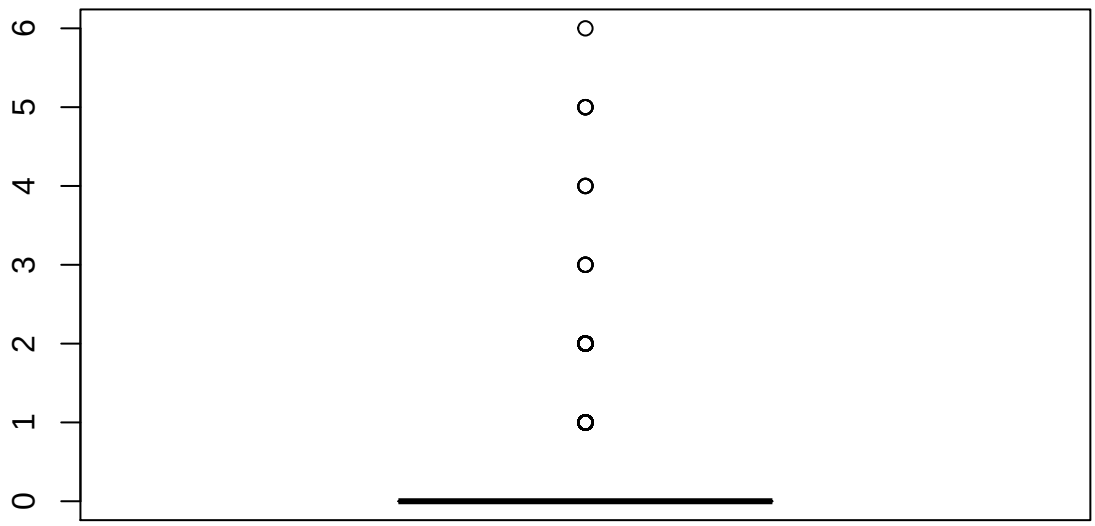


En el caso de las variables **SibSp** y **Parch** si realizamos el *boxplot* se observa que presentan muchos valores extremos. Esto es debido a la naturaleza de las variables que se puede visualizar mediante histrogramas y la curva de los cuartiles teoricos vs observados. Como las variables **SibSp** y **Parch** contienen el número de parentescos es común tener predominancia en los núemeros bajos y menos casos a medida que aumenta el valor de la variable. Aunque son númericas, su distribución es similiar a la de las variable discretas con las observaciones agrupándose en determinados valores (i.e. 0, 1, 2....). Debido a su naturaleza se considera que la discretización de ambas aportará más valor al análisis ya que no es posible normalizar la variable.

```
x<-datos$SibSp
y<-datos$Parch

boxplot(x, xlab="SibSp", col = "orange")
```

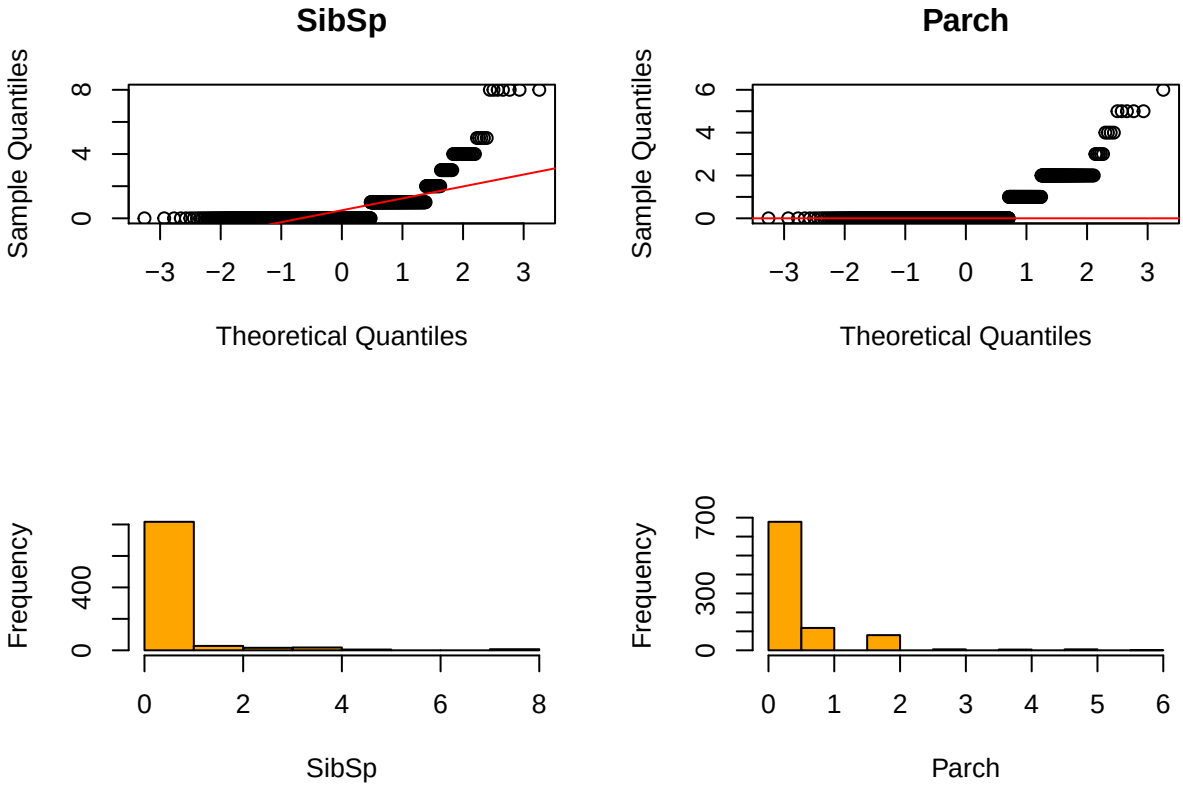



```
boxplot(y, xlab= "Parch", col= "orange")
```



Parch

```
par(mfrow=c(2,2))

qqnorm(x, main="SibSp")
qqline(x,col=2)

qqnorm(y, main="Parch")
qqline(y,col=2)

hist(x, main="", xlab = "SibSp", col="orange")
hist(y, main="", xlab = "Parch", col="orange")
```



Si la variable **SibSp** nos indica el número de conyugales o hermanos y la variable **Parch** nos indica el número de padres e hijos debe de haber una relación entre dichas variables.

```
SibSp_factor<- as.factor(datos$SibSp)
Parch_factor<- as.factor(datos$Parch)

chisq.test(Parch_factor, SibSp_factor)
```

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

```
##
##  Pearson's Chi-squared test
##
## data:  Parch_factor and SibSp_factor
## X-squared = 341.68, df = 36, p-value < 2.2e-16
```

Teniendo en cuenta que p-valor<0.05, ambas variables son dependientes. Para evitar información redundante en los algoritmos de predicción se crea una nueva variable denominada **Parents** (i.e. parientes) que será la suma de **SibSp** y **Parch**.

```
# creamos la nueva variable

datos$Parents <- as.factor(datos$SibSp + datos$Parch)
chisq.test(datos$Parents, datos$Survived)
```

```
## Warning in chisq.test(datos$Parents, datos$Survived): Chi-squared approximation
## may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  datos$Parents and datos$Survived
## X-squared = 80.672, df = 8, p-value = 3.58e-14
```

Teniendo en cuenta que el p-valor<0.05 se puede afirmar que existe una dependencia entre la nueva variable **Parents** y la supervivencia.Finalmente se eliminan las variables **SibSp** y **Parch** antes de iniciar el análisis de los datos.

```
datos$SibSp<-NULL
datos$Parch<-NULL
```

3.3. Comprobación de la variable Ticket

Se cree que la información de la variable **Ticket** está implícita en la variable **Fare** ya que el precio de un mismo número de ticket debería de tener igual valor. Para comprobar esta posibilidad se realiza el siguiente análisis.

```
# comprobamos si existe alguna diferencia de precio para algún ticket cuyo valor es igual.
tickets_check<-c()
for (i in levels(datos$Ticket)){
 a <- length(unique(datos$Fare[which(datos$Ticket==i)]))
 if(a>1) {
   tickets_check <- append(tickets_check,i)
   }
}
tickets_check
```

```
## [1] "7534"
```

```
subset(datos[datos$Ticket==tickets_check[1], ],
       select = c(PassengerId, Ticket, Cabin, Embarked, Fare))
```

```
##     PassengerId Ticket         Cabin Embarked     Fare
## 139         139   7534 without_cabin        S 1.380507
## 877         877   7534 without_cabin        S 1.403477
```

Se observa que solo un número de ticket presenta valores diferentes en la variable **Fare** y que esta diferencia es muy pequeña. Por lo tanto, consideramos que la información de la variable **Ticket** ya está explicada en la variable **Fare** y, por lo tanto, no es necesaria para el análisis. Además, se considera más conveniente utilizar la variable **Fare** ya que no es identificadora y ayudará más a comprender la supervivencia de los pasajeros e función del precio que hayan pagado por el billete.

```
# imprimos le fichero después de las modificaciones realizadas.
write.csv(datos, '../data/titanic_clean_data.csv', row.names = F)
```

```
# eliminamos las variables que no usaremos para el análisis
datos$PassengerId<-NULL
datos$Name<-NULL
datos$Ticket<-NULL
```

Realizamos un `attach` de los datos después de modificar las limpiar las variables para facilitar el uso de ellas en el análisis.

```
attach(datos)
```

4. Análisis de los datos

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

En primer lugar, debido a qué el objetivo del análisis es entender qué variables influyen y en qué medida en la supervivencia, se van a realizar contrastes de hipótesis para medir la dependencia de las variables con respecto a l asupervivencia. En el caso de las variables categóricas se realizarán test *Chi-cuadrado* de la siguiente manera:

- **PClass** vs **Survived**
- **Sex** vs **Survived**
- **Age** vs **Survived**
- **Parents** vs **Survived**
- **PClass** vs **Survived**
- **Cabin** vs **Survived**
- **Embarked** vs **Survived**

Además de ver si estas variables son dependientes o no, se va a calcular el estadístico Gamma de Goodman Kruskal que nos da una idea del grado de dependencia de éstas. Los valores están comprendidos entre -1 y 1, con el signo indicando si la dependencia es positiva o negativa y el valor absoluto la fuerza de esta. Los valores cercanos a 0 indican que las dos variables son independientes.

Por otro lado, en la variable continua **Fare** se hará un contraste de medias respecto a **Survived** con el objetivo de entender si un precio diferente de ticket influye en la probabilidad de supervivencia.

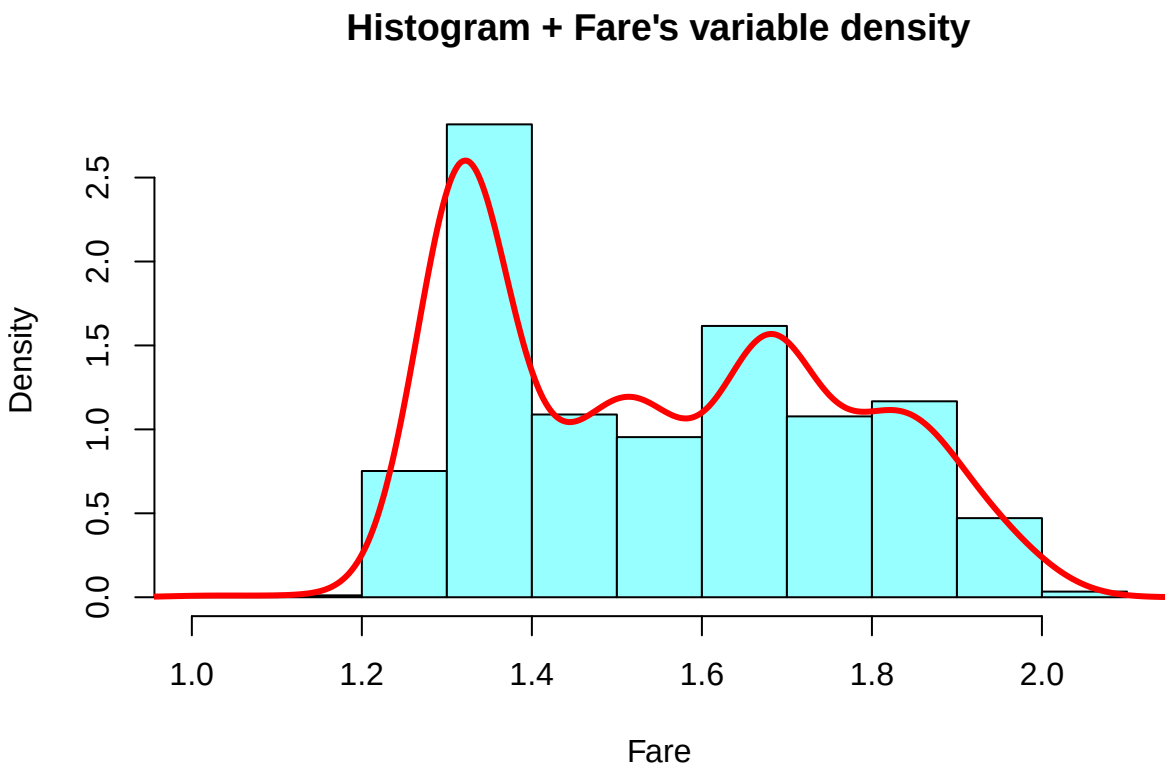
Una vez comprendida la relación de cada variable con la supervivencia, se va a utilizar esta información para saber que variables van a ser usadas en los algoritmos de predicción. En concreto se van a aplicar dos algoritmos supervisados de clasificación para entrenar un modelo que nos permita predecir la probabilidad de supervivencia de un pasajero en función de los atributos que disponemos. Estos algoritmos son:

- Árbol de decisión: Es un algoritmo que va a intentar subdividir el espacio de entrada de datos para generar regiones disyuntas. Cada una de estas regiones estará asociada a los atributos de entrada y tendrá asociada una probabilidad de supervivencia(Roig et al. 2018).
- randomforest: Es un algoritmo que usa varios clasificadores basados en árboles de decisión para mejorar la predicción. Se construye cada árbol con m predictores escogidos de forma aleatoria, y se promedia el conjunto de modelos(Roig et al. 2018).

4.2. Comprobación de la normalidad y homogeneidad de la varianza

En el conjunto de datos analizado tan sólo existe una variable numérica, la variable **Fare**. En el apartado 3 se ha transformado para intentar normalizarla ya que ésta seguía una distribución lognormal. En primer lugar, se realiza un test de normalidad para entender si se ha conseguido o no normalizar dicha variable.

```
hist(Fare, col = 'darkslategray1',
     main = "Histogram + Fare's variable density",
     freq = FALSE,
     xlab = 'Fare',
     ylab= "Density")
lines(density(Fare), col = 'red', lwd='3')
```



En el gráfico anterior vemos que pese a qué la variable ha sido transformada no sigue una distribución normal. Lo podemos reforzar con un test de normalidad de *saphiro* donde las hipótesis a contrastar son las siguientes:

H_0: La variable se distribuye como una normal.

H_1: La varaible no se distribuye como una normal.

donde, H_0 representa la hipótesis nula y H_1 la hipótesis alternativa.

```
shapiro.test(Fare)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Fare
## W = 0.92116, p-value < 2.2e-16
```

Debido a qué $p - valor < \alpha = 0.05$ rechazamos la hipótesis nula y aceptamos que los datos no siguen una distribución normal. Por lo tanto, pese a haber intentado normalizar los datos con una transformación, estos no siguen dicha distribución y se tendrá en cuenta para los posteriores análisis.

Para comprobar la homogeneidad de la variable **Fare** se van a utilizar las variables que se consideran más críticas para definir el precio. Estas son las variables **Embarked**, **Pclass** y **Cabin**. Debido a qué la variable **Fare** no sigue una distribución normal, nos vemos obligados a realizar un test basado en la mediana. Se va a aplicar el test de Levene ya que este nos permite medir la homocedasticidad o homogeneidad a partir de la mediana sin importar que los datos no sigan una distribución normal. Las hipótesis a contrastar en el test es la siguiente:

H_0: No hay diferencias significativas entre las varianzas.

H_1: Hay diferencias significativas entre las varianzas

donde, H_0 representa la hipótesis nula y H_1 la hipótesis alternativa.

```
# Fare - Embarked
LeveneTest(y = Fare, group = Embarked, center = 'median')
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value    Pr(>F)
## group   2  23.928 7.561e-11 ***
##       888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Fare - Pclass
LeveneTest(y = Fare, group = Pclass, center = 'median')
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value   Pr(>F)
## group   2  5.6586 0.003614 **
##       888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Fare - Cabin
LeveneTest(y = Fare, group = Cabin, center = 'median')
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##        Df F value    Pr(>F)
## group   1  26.959 2.576e-07 ***
##       889
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En los tres resultados del test el $p - valor < \alpha = 0.05$, por lo tanto, rechazamos la hipótesis nula y aceptamos que para todas la variable **Fare** tiene diferencias significativas en términos de varianza para los diferentes factores de las variables **Embarked**, **Pclass** y **Cabin**.

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

A continuación se van a relizar las pruebas estadísticas planificadas en el apartado 4.1.

4.3.1. Dependencia de las variables con respecto a la probabilidad de sobrevivir.

Primero se analizará la dependencia de cada una de las variables del conjunto de datos con la variable clase objetivo(i.e. **Survived**)

4.3.1.1. Variables categóricas

Mediante el test *Chi-cuadrado* se van a contrastar las siguientes hipótesis:

H_0: Las variables son independientes.

H_1: Las variables no son independientes.

donde, H_0 representa la hipótesis nula y H_1 la hipótesis alternativa.

Los pares de variables a analizar son:
- **PClass** vs **Survived**
- **Sex** vs **Survived**
- **Age** vs **Survived**
- **Parents** vs **Survived**
- **Cabin** vs **Survived**
- **Embarked** vs **Survived**

```
# Test chi-cuadrado
chis_test_matrix<-matrix(
                    c(
                      chisq.test(Pclass, Survived)$p.value,
                      chisq.test(Sex, Survived)$p.value,
                      chisq.test(Age, Survived)$p.value,
                      chisq.test(Parents, Survived)$p.value,
                      chisq.test(Cabin, Survived)$p.value,
                      chisq.test(Embarked, Survived)$p.value)
                      ,
                    dimnames = list(c("PClass vs Survived",
                                      "Sex vs Survived",
                                      "Age vs Survived",
                                      "Parents vs Survived",
                                      "Cabin vs Survived",
                                      "Embarked vs Survived")))


chis_test_matrix
```

```
##                              [,1]
## PClass vs Survived   4.549252e-23
## Sex vs Survived      1.197357e-58
## Age vs Survived      1.401033e-04
## Parents vs Survived  3.579669e-14
## Cabin vs Survived    6.741970e-21
## Embarked vs Survived 2.300863e-06
```

```
# Test Gamma

datos$Sex<-relevel(datos$Sex, ref="female")
datos$Age<-relevel(datos$Age, ref="child")
datos$Parents <- ordered(datos$Parents, levels = c("3", "2", "1", "6",
                                                   "0", "4", "5", "7", "10"))

gamma_matrix<-matrix(
                c(
                  GoodmanKruskalGamma(datos$Pclass, y = Survived),
                  GoodmanKruskalGamma(datos$Sex, y = Survived),
                  GoodmanKruskalGamma(datos$Age, y = Survived),
                  GoodmanKruskalGamma(datos$Parents, y = Survived),
                  GoodmanKruskalGamma(datos$Cabin, y = Survived),
                  GoodmanKruskalGamma(datos$Embarked, y = Survived))
                  ,
                dimnames = list(c("PClass vs Survived",
                                  "Sex vs Survived",
                                  "Age vs Survived",
```

```
                                  "Parents vs Survived",
                                  "Cabin vs Survived",
                                  "Embarked vs Survived")))

gamma_matrix
```

```
##                             [,1]
## PClass vs Survived   -0.5486386
## Sex vs Survived      -0.8501947
## Age vs Survived      -0.2727658
## Parents vs Survived  -0.4909406
## Cabin vs Survived    -0.6472603
## Embarked vs Survived -0.3252271
```

Para todas las variables el $p - valor < \alpha = 0.05$, luego se rechaza la hipótesis nula a favor de la alternativa. Por lo tanto, todas estas variables son dependientes por lo que todas tienen un peso en la probabilidad de sobrevivir. Además se observa mediante el test Gamma que dicha fuerza es mayor en las variables **Sex**, **Cabin** y **Pclass** y menor en **Parents**, **Embarked** y **Age**. El hecho que el signo sea negativo nos indica que la probabilidad de sobrevivir disminuye a medida que avanzamos en los diferentes niveles del factor y esto depende de como están o han sido ordenados con las funciones *relevel()* o *ordered*. Dicha relación se verá gráficamente en el apartado 5 del presente informe, sin embargo, y a moddo explicativo de de las variables con una mayor fuerza en la dependencia, decir que el hecho de ser mujer, viajar en cabina y en primera clase aumenta las probabilidades de sobrevivir.

4.3.1.2. Variables continuas.

El objetivo de este análisis es identificar si existe una relación entre la media del precio del ticket y la probabilidad de que una persona sobreviva o no. La variable **Fare** no sigue una distribución normal y la varianza es distinta en las poblaciones. Sin embargo, debido a que el número de observaciones es mayor a 30 y en base al Teorema del Límite Central suponemos normalidad.

Se define el siguiente contraste de hipótesis:

$$H_0 : \mu_1 = \mu_2$$
$$H_1 : \mu_1 \neq \mu_2$$

donde, μ representa la media poblacional, 1 corresponde al conjunto que sobrevivió y 2 al que no sobrevivió.

En base al Teorema del límite central tenemos la variable $\frac{\overline{X}_1 - \overline{X}_2}{S_{\overline{X}_1 - \overline{X}_2}}$ que sigue una distribución normal $N(0, 1)$.

```
# generamos una función para calcular el estadístico z por el teorema del límite central
calculate_z <- function(x, y) {
  # calculamos el numerador
  z_num <- mean(x) - mean(y)
  z_den <- sqrt(
    ((sd(x)^2)/length(x))
    +
    ((sd(y)^2)/length(y))
  )
  z <- z_num / z_den
  z
}

z <- calculate_z(Fare[Survived==1], Fare[Survived==0])
z;
```

```
## [1] 9.882642
```

El p-valor asociado a z es:

$$p = 2P(Z > |z|)$$

```
2*pnorm(z, lower.tail = F) < 0.05
```

```
## [1] TRUE
```

Debido a que el p-valor es inferior al nivel de significancia fijado como $\alpha = 0.05$ rechazamos la hipótesis nula a favor de la hipótesis alternativa y podemos considerar que los precios de los tickets de los pasajeros que sobrevivieron son diferentes a los de los pasejeros que no sobrevivieron. Por lo tanto, es una variable que aporta información sobre la probabilidad de supervivencia del pasajero.

4.3.2. Algoritmos de clasificación

Teniendo en cuenta los resultados del aparado anterior se van a usar la siguientes variables para los algortimos de clasificación: **Pclass**, **Sex**, **Age**, **Parents**, **Cabin**, **Embarked** y **Fare**. Siendo la variable objetivo **Survived**

4.3.2.1 Árbol de decisión

Es interesante aplicar un árbol de decisión para entender que variables son más significativas a la hora de predecir si un pasajero sobrevive.

```
library(rpart)
library(rpart.plot)
library(caret)
```

En primer lugar, preparamos los datos. Se decide utilizar como entrenamiento un 80 % de los registros y testear con un 20 %. Esta proporción es ampliamente aceptada en entrenamiento de algoritmos y está basada en el principio de Pareto, que establece que el 80% de los efectos está provocado por el 20% de las causas.

```
# Selección variable clase y variables predictoras
y_variable<-subset(datos, select = Survived)
x_variables <-subset(datos, select = c(Pclass, Sex, Age, Fare, Cabin, Embarked, Parents))

# Seleccion de los indices para los datos de entrenamiento y test
set.seed(333, sample.kind = "Rounding")
train_index <- sample(1:nrow(x_variables), 0.8 * nrow(x_variables))
test_index <- setdiff(1:nrow(x_variables), train_index)

# Construyendo conjunto de entrenamiento y test
X_train <- x_variables[train_index,]
y_train <- y_variable[train_index,]

X_test <- x_variables[test_index, ]
y_test <- y_variable[test_index, ]
```

Generamos el árbol de decisión utilizando todas las variables .

```
# generamos el árbol de decisión
arbol_1 <- rpart(formula = y_train ~ ., data = X_train)
```

A continuación, se realiza una predicción de los datos de test para analizar la precisión del árbol de decisión.

```r
# predicción
predict_tree <- predict(arbol_1, newdata = X_test, type = "class")
# matriz de confusión
confusionMatrix(predict_tree, y_test)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 115  22
##          1   4  38
##
##                Accuracy : 0.8547
##                  95% CI : (0.7945, 0.9029)
##     No Information Rate : 0.6648
##     P-Value [Acc > NIR] : 6.952e-09
##
##                   Kappa : 0.6479
##
##  Mcnemar's Test P-Value : 0.0008561
##
##             Sensitivity : 0.9664
##             Specificity : 0.6333
##          Pos Pred Value : 0.8394
##          Neg Pred Value : 0.9048
##              Prevalence : 0.6648
##          Detection Rate : 0.6425
##    Detection Prevalence : 0.7654
##       Balanced Accuracy : 0.7999
##
##        'Positive' Class : 0
##
```

El árbol de decisión ha sido capaz de colocar correctamente un 85.47 % de los registros de test, los cuales corresponden a un 20 % de nuestra población.

4.3.2.2 randomforest

A continuación, se aplica el algoritmo de Random Forest. Se aplica este tipo de algoritmo por dos motivos: (1) Se considera una buena alternativa cuando se tienen muchas variables categóricas de entrada, y (2), nos ayuda a comprender el poder del árbol de decisión generado anteriormente ya que el algoritmo genera muchos árboles con diferentes condiciones.

```r
library(randomForest)
# devtools::install_github("MI2DataLab/randomForestExplainer")
library(randomForestExplainer)
```

A continuación, se genera el random forest para predecir la variable **Survived** en relación a todas las demás variables de entrada.

```r
rf <- randomForest(
  y_train ~ .,
  data=X_train
)
```

Se realiza una predicción con la parte restante de los datos no utilizada.

```r
predict_random_forest = predict(rf, newdata=X_test)
# matriz de confusión
confusionMatrix(predict_random_forest, y_test)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 113  21
##          1   6  39
##
##                Accuracy : 0.8492
##                  95% CI : (0.7882, 0.8982)
##     No Information Rate : 0.6648
##     P-Value [Acc > NIR] : 2.03e-08
##
##                   Kappa : 0.6392
##
##  Mcnemar's Test P-Value : 0.007054
##
##             Sensitivity : 0.9496
##             Specificity : 0.6500
##          Pos Pred Value : 0.8433
##          Neg Pred Value : 0.8667
##              Prevalence : 0.6648
##          Detection Rate : 0.6313
##    Detection Prevalence : 0.7486
##       Balanced Accuracy : 0.7998
##
##        'Positive' Class : 0
##
```

Se observa cómo la precisión ha disminuido levemente con respecto al árbol de decisión (i.e. de 85.5% a 84.9%) . Por lo tanto, en este caso randomforest tiene un poder predictivo ligeramente inferior al del árbol de decisión aplicado anteriormente.

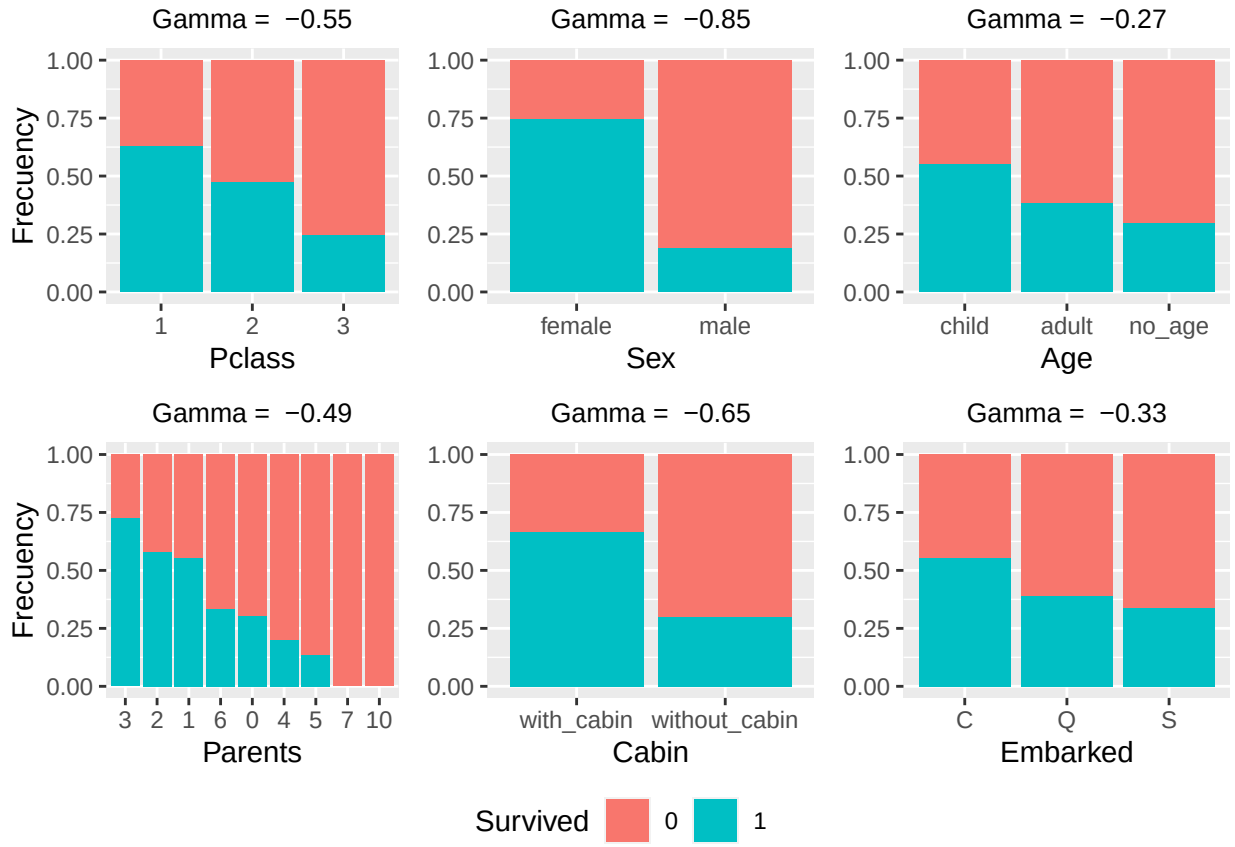
5. Representación de los resultados a partir de tablas y gráficas

5.1.Dependencia de las variables con respecto a la probabilidad de sobrevivir

En este apartado se va a representar gráficamente la dependencia de las variables categóricas con respecto a la probabilidad de sobrevivir.

```r
library(ggpubr)

p1<- ggplot(data=datos, aes(x=Pclass,fill=Survived)) + geom_bar(position="fill")+
     ylab("Frecuency") + xlab("Pclass")+
     ggtitle(paste("Gamma = ",toString(round(gamma_matrix[1], digits = 2)))) +
     theme(plot.title = element_text(size=10,hjust = 0.5))

p2<-ggplot(data=datos, aes(x=Sex,fill=Survived))+geom_bar(position="fill") +
    xlab("Sex")+theme(axis.title.y=element_blank())+
    ggtitle(paste("Gamma = ",toString(round(gamma_matrix[2], digits = 2))))+
    theme(plot.title = element_text(size=10,hjust = 0.5))
```

```
p3<-ggplot(data=datos, aes(x=Age,fill=Survived))+geom_bar(position="fill")+
    ylab("Frecuency")+xlab("Age")+theme(axis.title.y=element_blank())+
    ggtitle(paste("Gamma = ",toString(round(gamma_matrix[3], digits = 2))))+
    theme(plot.title = element_text(size=10,hjust = 0.5))

p4<-ggplot(data=datos, aes(x=Parents,fill=Survived))+geom_bar(position="fill")+
    ylab("Frecuency")+xlab("Parents")+
    ggtitle(paste("Gamma = ",toString(round(gamma_matrix[4], digits = 2))))+
    theme(plot.title = element_text(size=10,hjust = 0.5))

p5<-ggplot(data=datos, aes(x=Cabin,fill=Survived))+geom_bar(position="fill")+
    ylab("Frecuency")+xlab("Cabin")+theme(axis.title.y=element_blank())+
    ggtitle(paste("Gamma = ",toString(round(gamma_matrix[5], digits = 2))))+
    theme(plot.title = element_text(size=10,hjust = 0.5))

p6<-ggplot(data=datos, aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+
    ylab("Frecuency")+xlab("Embarked")+theme(axis.title.y=element_blank())+
    ggtitle(paste("Gamma = ",toString(round(gamma_matrix[6], digits = 2))))+
    theme(plot.title = element_text(size=10,hjust = 0.5))


ggarrange(p1, p2,p3,p4,p5,p6, ncol = 3 , nrow = 2,
          common.legend = TRUE, legend="bottom")
```



Como ya se avanzó en el apartado anterior el sexo, el hecho de tener cabino o no y la clase donde se compró el

billete determinan en mayor medida la probabilidad de sobrevivir. Ser mujer, y alojarse en cabina en primera clase aumentan estas posibilidades. Por contra, ser hombre, estar en tercera clase y no tener cabina asignada la disminuyen. El número de familiares, el sitio de embarque y la edad también presentan una dependencia con la probabilidad de sobrevivir aunque más débil. Un número de familiares entre 1 y 3 aumentan la posibilidad de supervivencia, disminuyendo para familias mas grandes y en el caso que se viaje solo. El hecho de ser niño también aumentan la probabilidad de supervivencia, así como el sitio donde se embarcó, siendo Cherbourg ("C") el lugar de embarque que las aumenta frente a Queenstown ("Q") y Southampton ("S").

Estas variables pueden tener interacción entre ellas, influyendo alguno de los niveles de unas en las otras. Por ejemplo, es asumible pensar que el hecho de tener cabina o no, está asociado al tipo de clase en que se aloja el pasajero. Por ello ejecutar algoritmos de clasificación ayudará a mejorar la interpretación de los resultados ya que teniendo en cuenta la interacción entre variables darán información de cuales de estas variables son más importantes a la hora de determinar la probabilidad de supervivencia.

5.2. Representación del árbol de decisión

Debido a como funciona el algoritmo de una arbol de decisión, su aplicación aporta una buena descripción de los datos. Al analizar un árbol de decisión se puede comprender qué peso y cómo se utilizan las variables para determinar la supervivencia o no de los pasajeros.

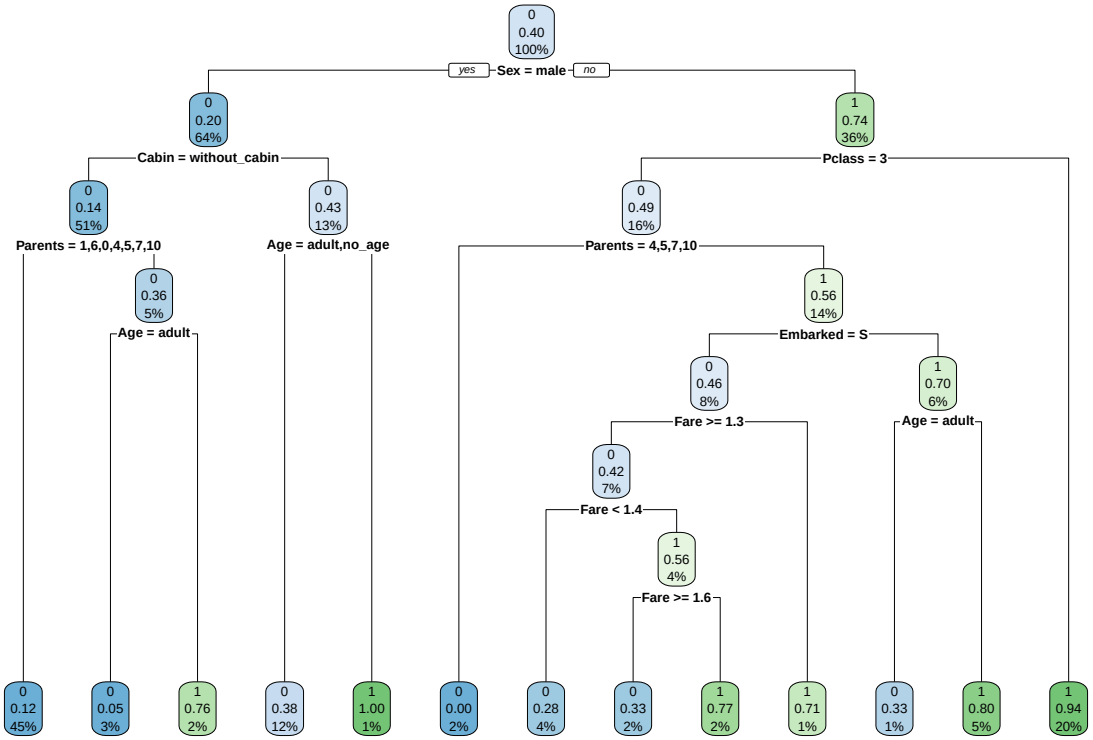
A continuación, se muestra el árbol de decisión.

```
arbol_1
```

```
## n= 712
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 712 282 0 (0.60393258 0.39606742)
##     2) Sex=male 454  91 0 (0.79955947 0.20044053)
##       4) Cabin=without_cabin 360  51 0 (0.85833333 0.14166667)
##         8) Parents=1,6,0,4,5,7,10 321  37 0 (0.88473520 0.11526480) *
##         9) Parents=3,2 39  14 0 (0.64102564 0.35897436)
##          18) Age=adult 22   1 0 (0.95454545 0.04545455) *
##          19) Age=child,no_age 17   4 1 (0.23529412 0.76470588) *
##       5) Cabin=with_cabin 94  40 0 (0.57446809 0.42553191)
##        10) Age=adult,no_age 87  33 0 (0.62068966 0.37931034) *
##        11) Age=child 7   0 1 (0.00000000 1.00000000) *
##     3) Sex=female 258  67 1 (0.25968992 0.74031008)
##       6) Pclass=3 116  57 0 (0.50862069 0.49137931)
##        12) Parents=4,5,7,10 15   0 0 (1.00000000 0.00000000) *
##        13) Parents=3,2,1,6,0 101  44 1 (0.43564356 0.56435644)
##          26) Embarked=S 57  26 0 (0.54385965 0.45614035)
##            52) Fare>=1.317333 50  21 0 (0.58000000 0.42000000)
##             104) Fare< 1.435049 25   7 0 (0.72000000 0.28000000) *
##             105) Fare>=1.435049 25  11 1 (0.44000000 0.56000000)
##               210) Fare>=1.577645 12   4 0 (0.66666667 0.33333333) *
##               211) Fare< 1.577645 13   3 1 (0.23076923 0.76923077) *
##            53) Fare< 1.317333 7   2 1 (0.28571429 0.71428571) *
##          27) Embarked=C,Q 44  13 1 (0.29545455 0.70454545)
##            54) Age=adult 9   3 0 (0.66666667 0.33333333) *
##            55) Age=child,no_age 35   7 1 (0.20000000 0.80000000) *
##       7) Pclass=1,2 142   8 1 (0.05633803 0.94366197) *
```

Se realiza un gráfico del árbol de decisión para visualizar como las variables explican la supervivencia o no.

```
rpart.plot(arbol_1, type = 2)
```



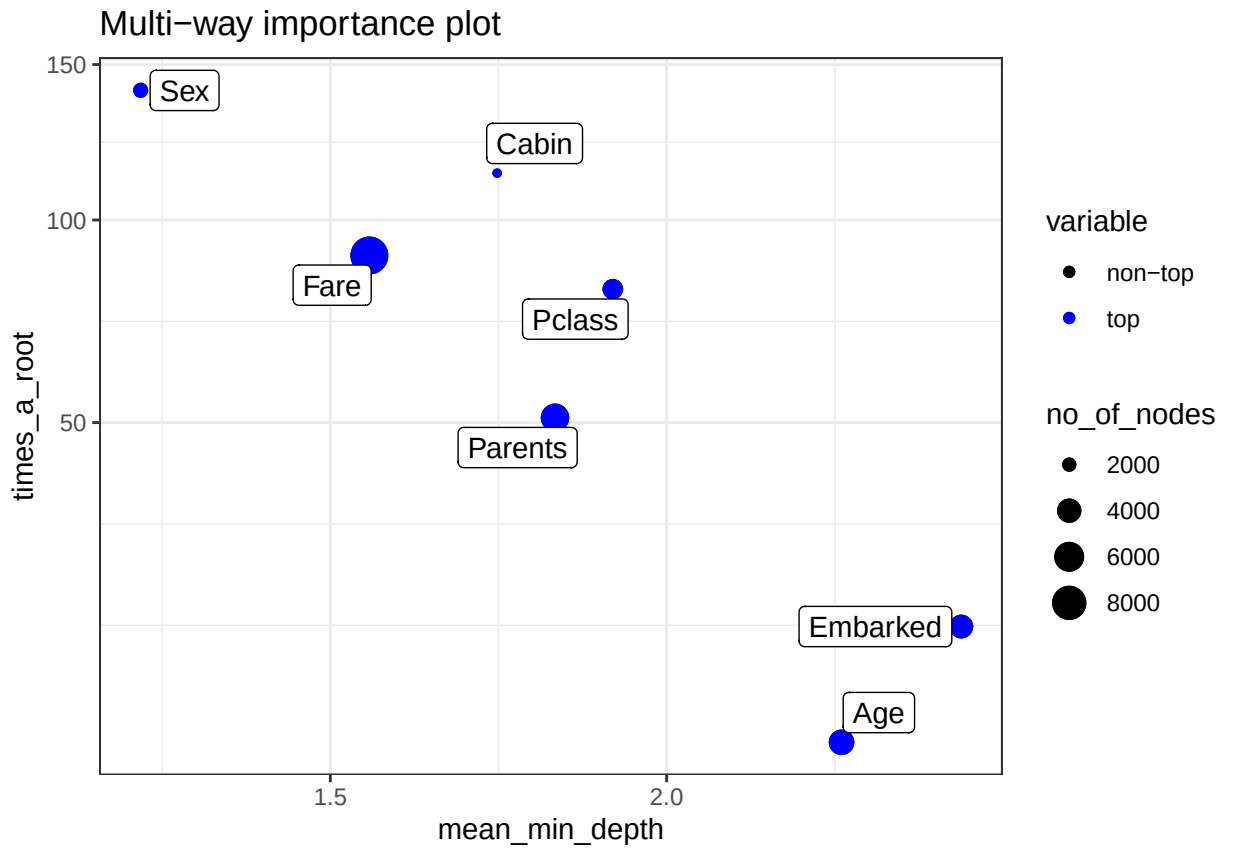
Como se puede observar en el árbol generado, la primera variable a partir de la cual se realiza una buena discriminación es **Sex**. Dependiendo de cada sexo, se utilizan las variables **Cabin** y **Pclass** para determinar si una persona sobrevive o no. Después de estas tres variables, el resto de variables han sido utilizadas para tomar la decisión de la supervivencia o no de los pasajeros. La mayor probabilidad de supervivencia se da con un 94% cuando se es mujer y se aloja en clase 1 o 2.

5.3. Representación del randomforest

El algoritmo de randomforest genera muchos árboles de decisión de forma aleatoria. Su interpretación es más complicada que el árbol de decisión, pero nos aporta información adicional ya que se generan múltiples árboles con entrada de variables aleatorias seleccionando los que mejoran la precisión. Por ello, nos da información de la importancia de las variables en determinar la probabilidad de supervivencia. Sin embargo, debido a su naturaleza puede provocar un sobreajuste de los datos, lo que reduce la precisión del conjunto test. A continuación se va a representar el número de veces que una variable ha sido raiz del árbol, frente al promedio de la profundidad de la primera vez que se separa.

```
plot_multi_way_importance(rf, size_measure = "no_of_nodes")
```

```
## [1] "Warning: your forest does not contain information on local importance so 'accuracy_decrease' mea
```



Las variables más utilizadas cómo raíz y con una menor promedio de la profundidad de la primera vez que se separa son aquellas más explicativas o discriminatorias. Como se puede observar, la variable **Sex** es la más utilizada y coincide con el árbol de decisión. Las variables **Fare** y **Cabin** también determinan fuertemente la supervivencia o no de los pasajeros, y en menor medida, **Pclas** y **Parents**. En cambio, las variable **Age** y **Embarked** son poco determinantes de la supervivencia del pasajero. El caso de la variable **Age** refuerza el análisis realizado anteriormente dónde se ha contemplado que la mayoría de los pasajeros son adultos y para estos casos no era muy relevante.

6. Conclusiones

Se han procesado las variables del juego de datos del Titanic con el objetivo de estudiar que determina la probabilidad de sobrevivir y mejorar la predicción de los algoritmos de clasificación. No se han utilizado las variables **PassangerId** y **Name** por ser variables únicas. Además la variable **Ticket** se descartó por estar implícita su información en la variable **Fare**.

Se usaron para el análisis final un total de 6 variables discretas (i.e. **Pclass**, **Sex**, **Parents**, **Embarked**, **Cabin** y **Age**) y 1 continua (i.e. **Fare**). No hubo ningún preprocesamiento, a parte de la discretización mediante la funcón *as.factor*, en las variables **Pclass** y **Sex**. Por su parte la variable **Parents** es el resultado de la suma de las variables **SibSp** y **Parch** las cuales tampoco fueron objeto de preprocesamiento.

Las variables **Age**,**Cabin**, **Embarked** y **Fare** presentaron valores **NA**, vacíos o 0. Debido al gran número de valores *NA*, la variable **Age** fue discretizada primero en función de la existencia o no de valores *NA*, y posteriormente en función de un umbral óptimo que diferencia entre la probabilidad de supervivencia de los individuos más jóvenes y el número de casos que comprendia dicha población. La variable **Cabin** fue discretizada en función de si tenía (i.e. *with_cabin*) o si no tenía (i.e. *whitout_cabin*) valor asociado. Por su parte, se usaron el conjunto de variables **Fare**, **Cabin**, **Embarked** y **Pclass** para imputar mediante *kNN()* los valores perdidos de la variable **Fare** (de valor 0) y **Embarked** que eran faltantes, ya que se considera

que este grupo de variables tienen una relación que las hace adecuadas para llevar a cabo dicha imputación. Por último, la única variable continua del juego de datos **Fare**, no pasó los test estadísticos de normalidad y homocestacidad. Sim embargo, como contiene más de 30 observaciones y en base al Teorema del Limite Central, se considera que tiene una distribución que tiende a la normalidad y por eso motivo fue utilizada.

La precisión de los algoritmos de clasificación se considera buena al rondar en ambos casos, árbol de decisión y randomForest, el 85% en el conjunto de test. Este hecho, conjuntamente con la confrontación de cada variable individualmente con la supervivencia nos permite poder evaluar que características de cada pasajero determinan su probabilidad de supervivencia. Se observó que el sexo, el hecho de viajar en cabina y la clase determinan mayormente la probabilidad de supervivencia, aumentando considerablemente al ser mujer y viajando en primera o segunda clase en cabina. EL hecho de viajar en cabina y en clases superiores aumenta considerablemente el precio del ticket (i.e. **Fare**) estando, por lo tanto, esta variable relacionada a la probabilidad de supervivencia. Por su parte, el numero de parientes, el sitio de embarque y la edad, aunque influyen en la probabilidad de sobrevivir, lo hacen en una manera menor.

7. Contribuciones

Contribucuiones	Firma
Investigación previa	K.C., A.C.
Redacción de las respuestas	K.C., A.C.
Desarrollo código	K.C., A.C.

8. Referencias

Roig, Jordi Gironès, Jordi Casas Roma, Julià Minguillón Alfonso, and Ramón Caihuelas Quiles. 2018. *Minerı'a de Datos: Modelos Y Algoritmos*. UOC.