Practica 2 - Limpieza y validación de los datos. M2851 - UOC

Ernesto Peralta Macías

Enero 2019

Table of Contents

1.-Descripción del dataset.

Los datos que analiaremos contienen información del naufragio del transatlántico británico RMS Titanic, después de colisionar con un iceberg el 14 de Abril de 1912 y donde murieron 1.502 de los 2.224 viajeros y tripulantes que iban en el navío.

Los datos objeto del análisis se han obtenido de la competición "Titanic: Machine Learning from Disaster" de kaggle.com. Tenemos dos conjuntos de datos: train y test.

El fichero de datos *train.csv* se utiliza para hacer un estudio analítico de los datos y para la construcción de un modelo que predice si el pasajero sobrevivirá o no al naufragio. Tiene 891 registros y 12 variables. Estas variables son: PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, Embarked

Variable	Description
Survived	Sobrevive (1). No sobrevive (0)
Pclass	Clase del camarote del pasajero/a
Name	Nombre del pasajero/a
Sex	Sexo del pasajero/a
Age	Edad del pasajero/a
SibSp	Número de hermanos/as, esposos/as
Parch	Número de padres / niños
Ticket	Número de Ticket
Fare	Tarifa
Cabin	Camarote
Embarked	Puerto de embarque

El fichero de datos *test.csv* se utiliza para aplicar el modelo recién construido con la finalidad de obtener una predicción de si el pasajero sobrevivirá al naufrafio o no y tiene 418 registros.

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

La finalidad del estudio será obtener un modelo que nos prediga quien sobrevivirá y quien no. Aunque hubo algún elemento de suerte en el hecho de sobrevivir al hundimiento, a priori, ya sabemos que algunos grupos de personas tenían más probabilidades de sobrevivir que otros, como las mujeres, los niños y la clase alta.

En un principio contamos con todas las variables del dataset. En función del estudio que vayamos haciendo veremos si hay que descartar alguna variable y su motivo.

2.1.-Indicar el tipo de variable estadística de cada una de las variables

---

```r
# factor
var.factor <- c(2,3,5,12)
var.integer <- c(1,6,7,8)
var.numeric <- c(10)
var.char <- c(4,9,11)
var.tipus <- vector(mode="character",length=ncol(Titanic))
var.tipus[var.factor] <- "factor"
var.tipus[var.integer] <- "integer"
var.tipus[var.numeric] <- "numeric"
var.tipus[var.char] <-"character"
print(var.tipus)
```

```
##  [1] "integer"   "factor"    "factor"    "character" "factor"
##  [6] "integer"   "integer"   "integer"   "character" "numeric"
## [11] "character" "factor"
```

Son variables cualitativas nominales: Survived, Sex, Embarked

Son variables cualitativas ordinales: Pclass

Son variables cuantitativas discretas: PassengerId, Age, SibSp, Parch

Son variables cuantitativas continuas: Fare

Son variables de texto: Name, Ticket, Cabin

2.2.-Asignar a cada variable el tipo de variable R adecuada

La lectura del fichero con la función `read.csv()` ha realizado la seguiente asignación a cada variable

```r
res <- sapply(Titanic,class)
kable(data.frame(variables=names(res),clase=as.vector(res)))
```

variables	clase
PassengerId	integer
Survived	integer
Pclass	integer
Name	character
Sex	character
Age	numeric
SibSp	integer
Parch	integer
Ticket	character

Fare	numeric
Cabin	character
Embarked	character

```
var_wrong <- n.var[res != var.tipus]
```

Por tanto, las variables con asignación equivocada y que es necesario corregir son: Survived, Pclass, Sex, Age, Embarked

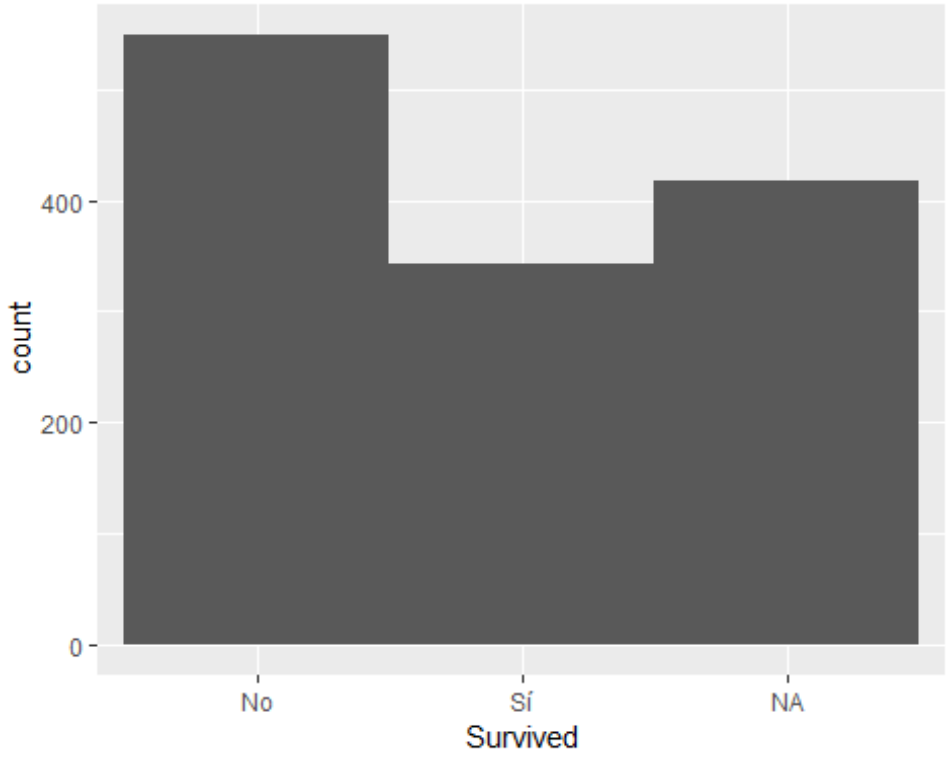
La asignación a realizar es:

```
kable(data.frame(variables= var_wrong, clase= c("factor","factor",
"factor", "integer", "factor")))
```

variables	clase
Survived	factor
Pclass	factor
Sex	factor
Age	integer
Embarked	factor

Transformar la variable Survived a tipo factor

```
Titanic$Survived <- factor(Titanic$Survived, levels= c(0,1), labels=
c("No","Sí"))
#Comprobamos la conversión del variable Survived
str(Titanic$Survived)

##  Factor w/ 2 levels "No","Sí": 1 2 2 2 1 1 1 1 2 2 ...

#Vemos su valores y tenemos en cuenta que los valores NA son los
correspondentes al dataset de test y que son los que tendremos que
rellenar nosotros en la predicción.
summary(Titanic$Survived)

##   No   Sí NA's
##  549  342  418

p <- ggplot(Titanic, aes(x=Survived))  + geom_bar(width=1)
p
```



Transformar la variable Pclass a tipo ordered

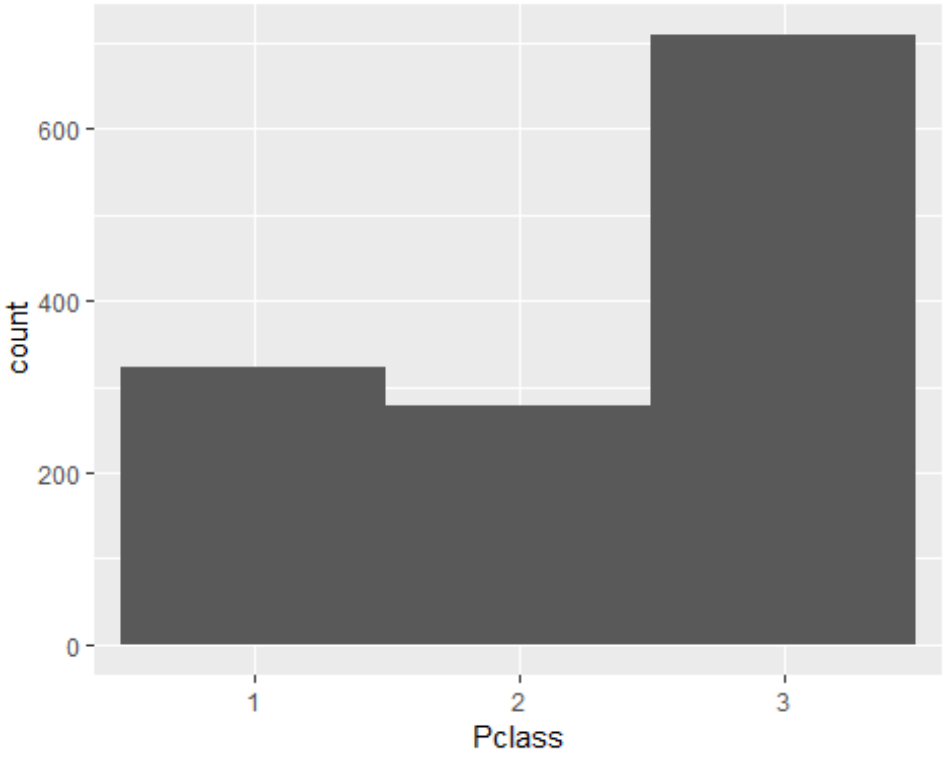
```
Titanic$Pclass <- ordered(Titanic$Pclass)
#Comprobamos la conversión del variable Pclass
str(Titanic$Pclass)
```

```
##  Ord.factor w/ 3 levels "1"<"2"<"3": 3 1 3 1 3 3 1 3 3 2 ...
```

```
#Vemos su valores
summary(Titanic$Pclass)
```

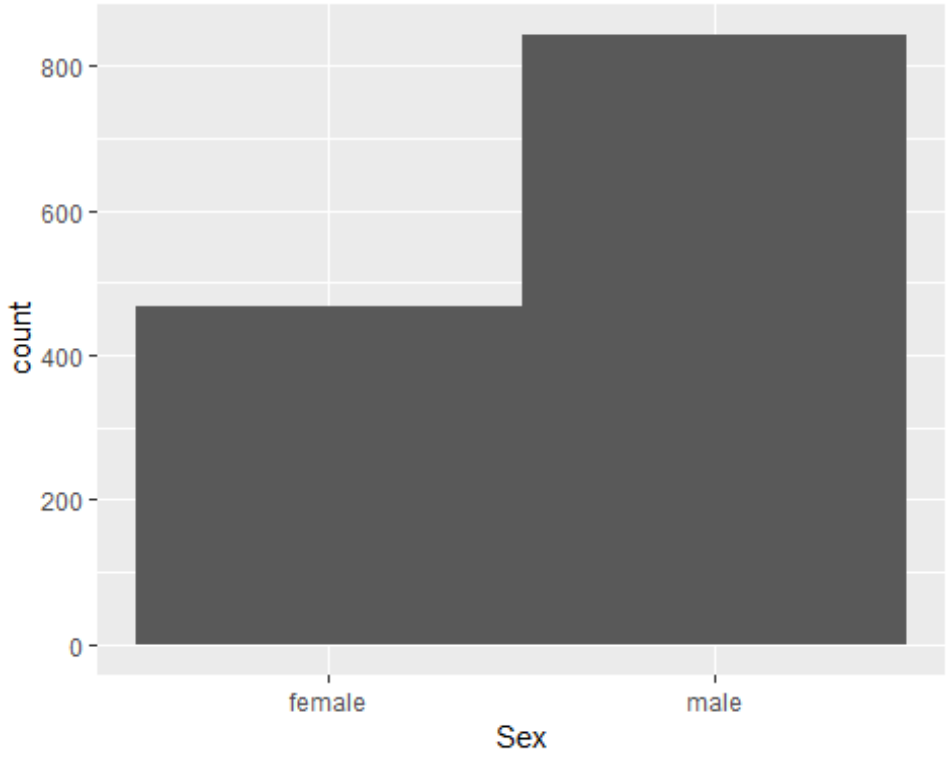
```
##   1   2   3
## 323 277 709
```

```
p <- ggplot(Titanic, aes(x=Pclass))  + geom_bar(width=1)
p
```



Transformar la variable Sex a tipo factor

```
Titanic$Sex <- factor(Titanic$Sex)
#Comprobamos la conversión del variable Sex
str(Titanic$Sex)

##  Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...

#Vemos su valores
summary(Titanic$Sex)

## female   male
##    466    843

p <- ggplot(Titanic, aes(x=Sex))  + geom_bar(width=1)
p
```



Transformar la variable edad a tipo integer

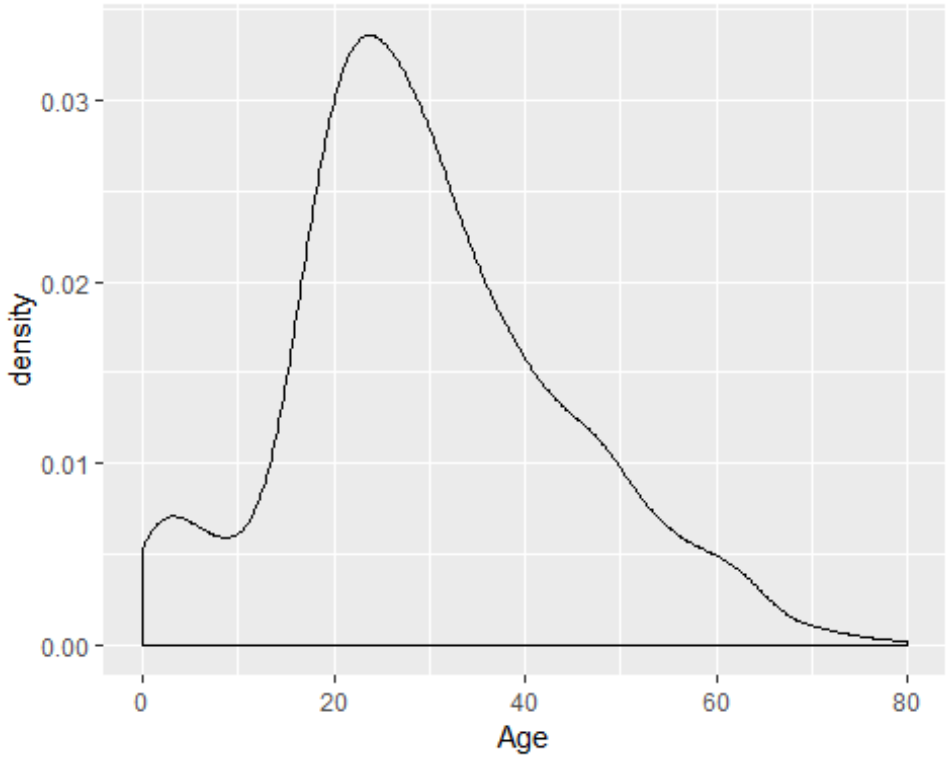
```
Titanic$Age <- as.integer(Titanic$Age)
#Comprobamos la conversión del variable Age
str(Titanic$Age)
```

```
##  int [1:1309] 22 38 26 35 35 NA 54 2 27 14 ...
```

```
#Vemos su valores y tenemos en cuenta que los valores NA son los
correspondentes al dataset de test y que son los que tendremos que
rellenar nosostrs en la predicción.
summary(Titanic$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   21.00   28.00   29.86   39.00   80.00     263
```

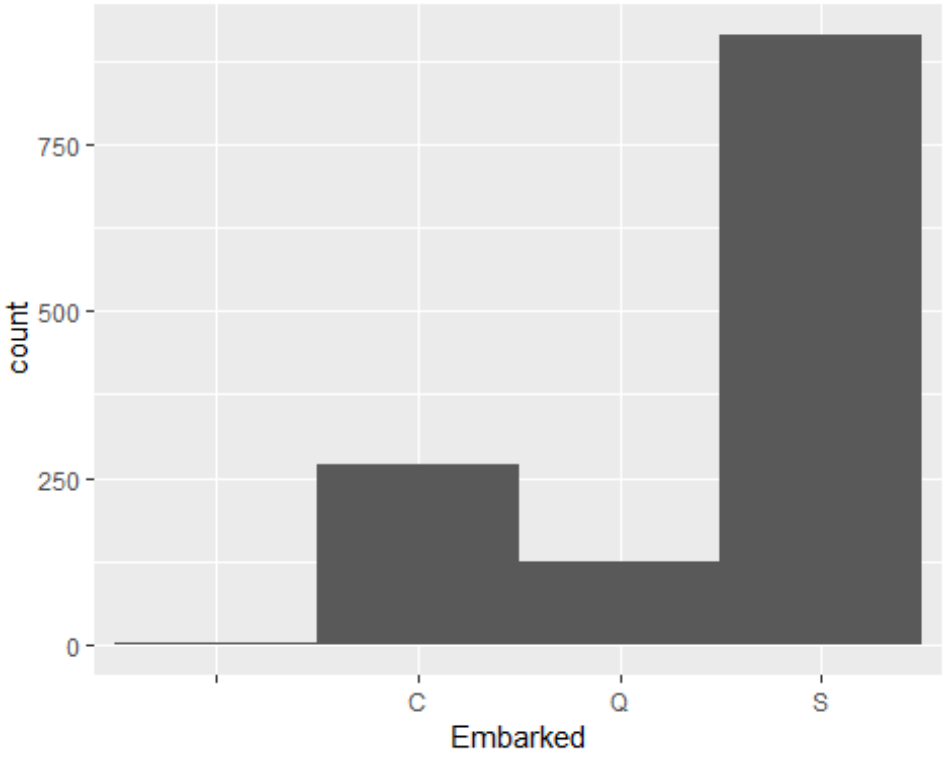
```
#
p <- ggplot(Titanic, aes(x=Age))  + geom_density(na.rm = TRUE)
p
```



Hemos realizado una conversión de la edad a números enteros. Los valores con algún decimal se han truncado. Eran mayoritariamente los bebés de 0 años. Adicionalmente hemos detectado 263 filas que no tienen valor.

Transformar la variable Embarked a tipo factor

```
Titanic$Embarked <- factor(Titanic$Embarked)
#Comprobamos la conversión del variable Embarked
str(Titanic$Embarked)

##  Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...

#Vemos su valores y tenemos en cuenta que los valores NA son los
correspondentes al dataset de test y que son los que tendremos que
rellenar nosotros en la predicción.
summary(Titanic$Embarked)

##       C   Q   S
##   2 270 123 914

p <- ggplot(Titanic, aes(x=Embarked))  + geom_bar(width=1)
p
```



Vemos que en la gráfica que existen 2 pasajeros que no tienen valor en el campo Embarked.

Normalizar/Estandarizar variable cuantitativa Fare

```
Titanic$Fare <- round(Titanic$Fare,2)
```

Normalizamos la variable Fare, tarifa, a 2 decimales.

3.-Limpieza de los datos.

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

Valores faltantes NA

```
# Números de valores desconocidos por campo
sapply(Titanic, function(x) sum(is.na(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex
Age
##           0         418           0           0           0
263
##       SibSp       Parch      Ticket        Fare       Cabin
Embarked
##           0           0           0           1           0
0
```

```
suppressWarnings(suppressMessages(library(VIM)))
Titanic$Fare <- kNN(Titanic)$Fare
Titanic$Age <- kNN(Titanic)$Age
#
sapply(Titanic, function(x) sum(is.na(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex
Age
##           0         418           0           0           0
0
##       SibSp       Parch      Ticket        Fare       Cabin
Embarked
##           0           0           0           0           0
0
```

Las variables con valores perdidos NA son Survived (que son los del dataset de Test y está así contemplado), Age que tiene 263 valores que calcularemos y Fare con un registro.

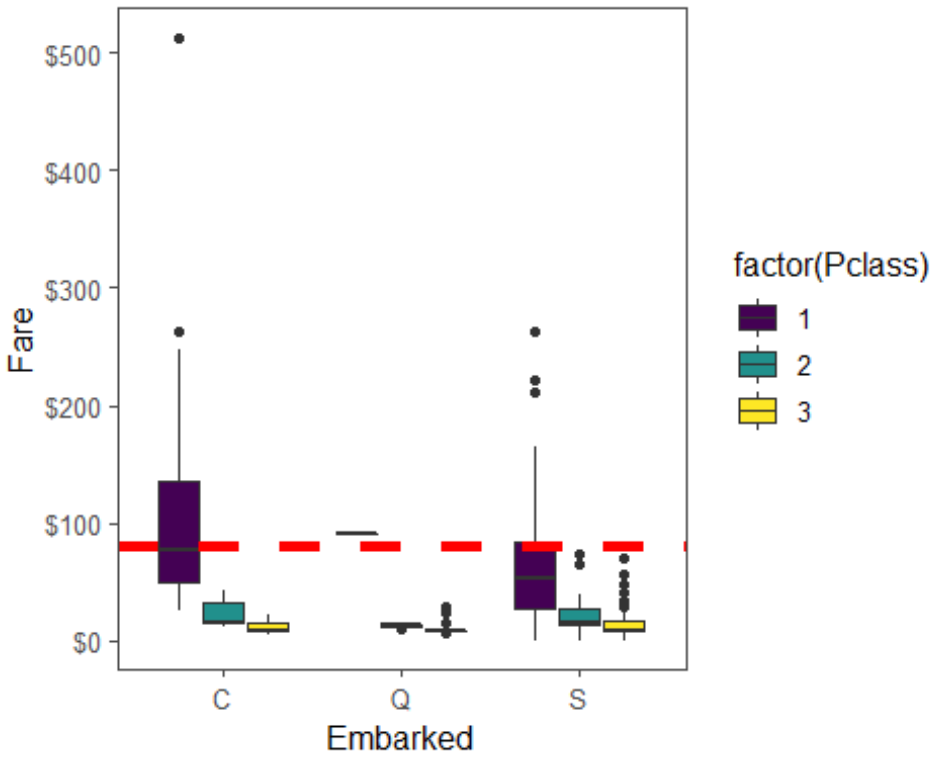
Valores faltantes vacío “”

La variable Embarked tiene dos valores vacíos en los registros 62, 830. Dado que sus valores de Pclass: 1, 1, sus valores de Fare: 80, 80 y sus valores de Cabin: B28, B28 son iguales, les haremos el mismo tratamiento.

¿ Qué valor les ponemos?

```
# Visualmente vemos las medias de las tarifas por ciudad de embarque y
marcamos las 80 libras en el gráfico.
Tarifa_Func_embarque <- Titanic %>%  filter(PassengerId != 62 &
PassengerId != 830)
ggplot(Tarifa_Func_embarque, aes(x = Embarked, y = Fare, fill =
factor(Pclass))) +
```

```
  geom_boxplot() +
  geom_hline(aes(yintercept=80),
    colour='red', linetype='dashed', lwd=2) +
  scale_y_continuous(labels=dollar_format()) +
  theme_few()
```



Vemos en el gráfico que la media de los embarcados en Cherburgo de 1ª clase pagaron 80 libras como nuestras pasajeras. Así que ese valor es el que le asignamos.

```
Titanic$Embarked[c(62, 830)] <- 'C'
```

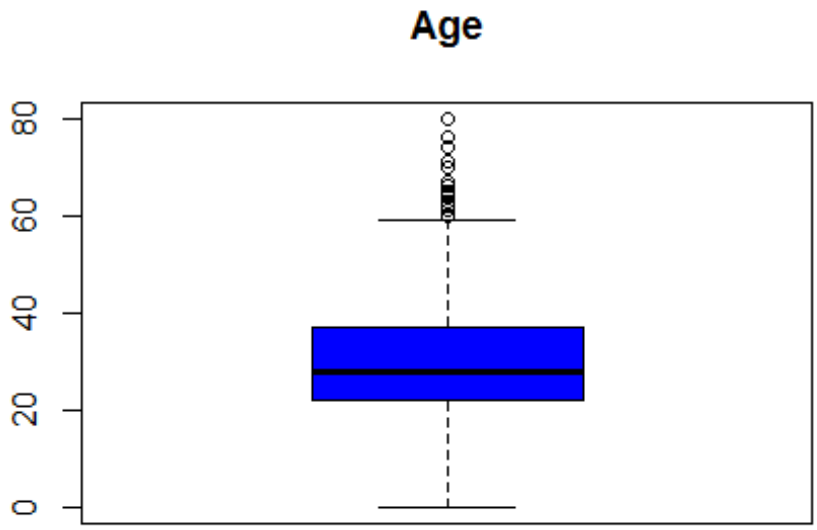
3.2.-Identificación y tratamiento de valores extremos.

Edad

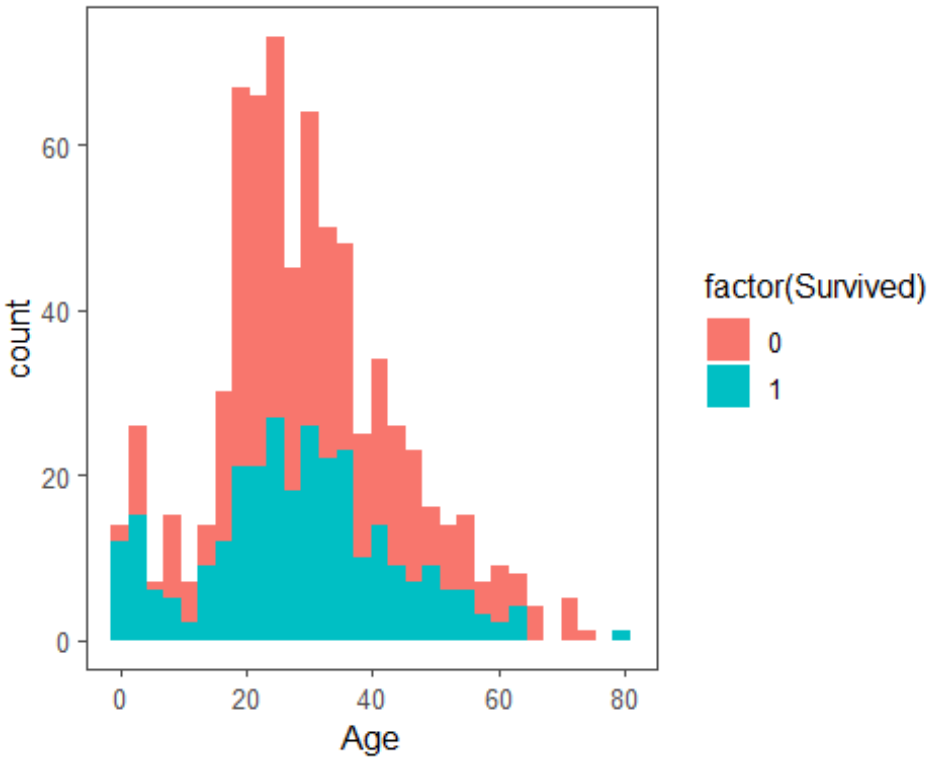
```
boxplot.stats(Titanic$Age)$out
```

```
##  [1] 66 65 71 70 61 62 63 65 61 60 64 65 63 71 64 62 62 60 61 80 70 60
60
## [24] 60 70 62 74 62 63 60 60 67 76 63 61 60 64 61 60 64 64
```

```r
# Boxplot
boxplot(Titanic$Age,main="Age", col="blue")
```




```r
# Histograma de Age y Survived
ggplot(train, aes(x = Age, fill = factor(Survived))) +
  geom_histogram(bins =30) +
  theme_few()
```



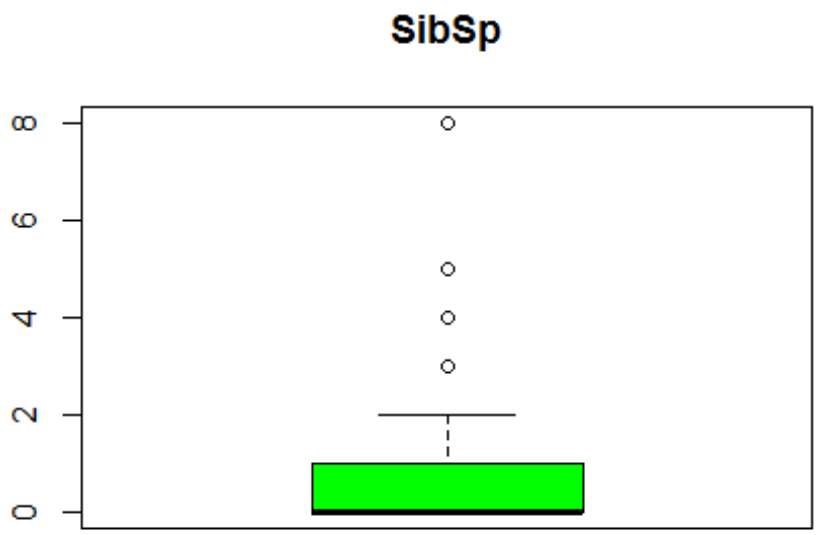
Los valores de edad mayores de 60 años son marcados como valores extremos por la distribución de los datos, pero no son los suficientemente extremos para ser fallos en la inscripción o erratas. Son datos válidos, lógicos y no creo que sea conveniente quitarlos porque podría influir negativamente en la predicción de nuestro modelo.

SibSp

```
boxplot.stats(Titanic$SibSp)$out
```

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

```
boxplot(Titanic$SibSp,main="SibSp", col="green")
```




```
summary(Titanic$SibSp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4989  1.0000  8.0000
```

Marca como valores extemos de los datos el hecho de tener como número de hermanos más esposo/a los valores mayores de 3 hasta 8. Es normal que los muestre como atípico pero son perfectamente normales en función de cada familia y los dejamos en nuestro modelo.

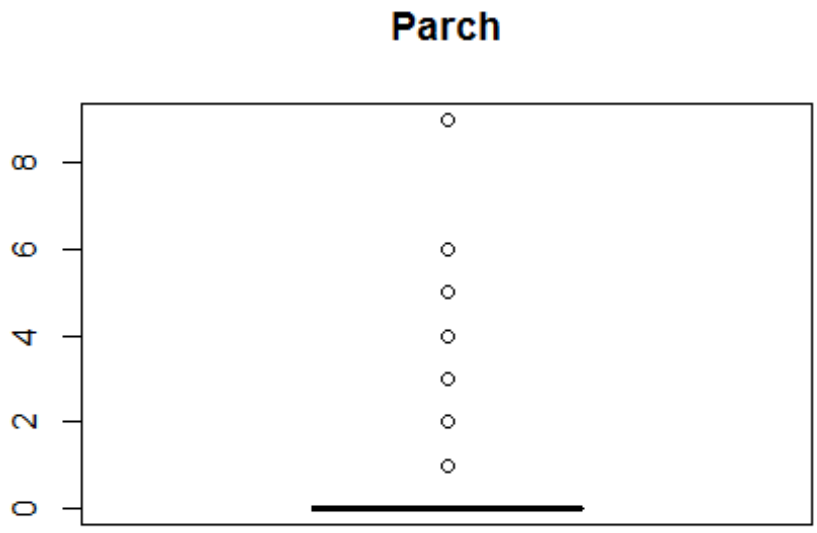
Parch

```
boxplot.stats(Titanic$Parch)$out
```

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

```
## [176] 1 2 1 2 3 1 2 1 2 2 1 1 2 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1 3 2
1 1 1
## [211] 1 5 2 1 1 1 1 3 1 2 2 1 2 1 2 1 2 4 1 1 2 1 1 1 4 6 2 3 1 1 2 2
2 1 1
## [246] 2 5 2 3 2 1 1 1 2 1 2 2 2 1 2 1 1 2 1 2 1 2 1 2 2 1 1 1 1 1 2 1
1 2 1
## [281] 1 1 2 1 2 9 1 1 1 2 2 2 1 9 1 1 2 2 1 1 2 1 1 1 1 1 1
```

```
boxplot(Titanic$Parch,main="Parch", col="red")
```



Marca como valores extemos de los datos el hecho de tener como número de padres más niños los valores mayores de 1 hasta 9. Es normal que los muestre como atípico pero son perfectamente normales en función de cada familia y los dejamos en nuestro modelo.

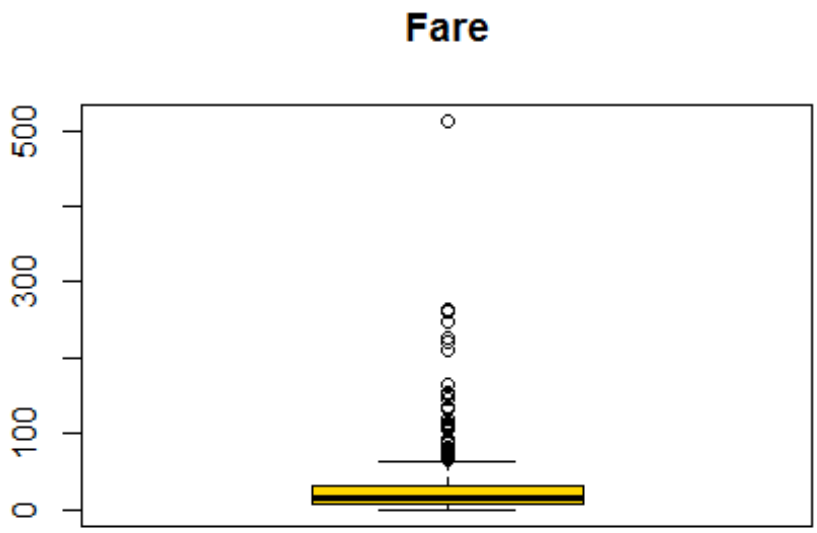
Fare

```
boxplot.stats(Titanic$Fare)$out
```

```
##   [1]  71.28 263.00 146.52  82.17  76.73  80.00  83.47  73.50 263.00
77.29
##  [11] 247.52  73.50  77.29  79.20  66.60  69.55  69.55 146.52  69.55
113.28
##  [21]  76.29  90.00  83.47  90.00  79.20  86.50 512.33  79.65 153.46
```

```
135.63
##  [31]  77.96  78.85  91.08 151.55 247.52 151.55 110.88 108.90  83.16
262.38
##  [41] 164.87 134.50  69.55 135.63 153.46 133.65  66.60 134.50 263.00
75.25
##  [51]  69.30 135.63  82.17 211.50 227.53  73.50 120.00 113.28  90.00
120.00
##  [61] 263.00  81.86  89.10  91.08  90.00  78.27 151.55  86.50 108.90
93.50
##  [71] 221.78 106.42  71.00 106.42 110.88 227.53  79.65 110.88  79.65
79.20
##  [81]  78.27 153.46  77.96  69.30  76.73  73.50 113.28 133.65  73.50
512.33
##  [91]  76.73 211.34 110.88 227.53 151.55 227.53 211.34 512.33  78.85
262.38
## [101]  71.00  86.50 120.00  77.96 211.34  79.20  69.55 120.00  93.50
80.00
## [111]  83.16  69.55  89.10 164.87  69.55  83.16  82.27 262.38  76.29
263.00
## [121] 262.38 262.38 263.00 211.50 211.50 221.78  78.85 221.78  75.24
151.55
## [131] 262.38  83.16 221.78  83.16  83.16 247.52  69.55 134.50 227.53
73.50
## [141] 164.87 211.50  71.28  75.25 106.42 134.50 136.78  75.24 136.78
82.27
## [151]  81.86 151.55  93.50 135.63 146.52 211.34  79.20  69.55 512.33
73.50
## [161]  69.55  69.55 134.50  81.86 262.38  93.50  79.20 164.87 211.50
90.00
## [171] 108.90
```

```r
boxplot(Titanic$Fare,main="Fare", col="gold")
```



Marca como valores extremos tarifas muy alta. Hay que tener en cuenta que, en su época era el transalántico más lujoso del planeta y que había camarotes y servicios que disparaban las tarifas media de forma desorbitada. Tambíen hay que tener en cuenta que los pasajeros subidos en el puerto de Cherburgo (Fracia) pagaban más, según vimos en el gráfico del punto 3.1.2 anterior.

```
##   PassengerId   Survived   Pclass      Name              Sex
##  Min.   :   1   No  :549   1:323   Length:1309        female:466
##  1st Qu.: 328   Sí  :342   2:277   Class :character   male  :843
##  Median : 655   NA's:418   3:709   Mode  :character
##  Mean   : 655
##  3rd Qu.: 982
##  Max.   :1309
##       Age            SibSp            Parch          Ticket
##  Min.   : 0.00   Min.   :0.0000   Min.   :0.000   Length:1309
##  1st Qu.:22.00   1st Qu.:0.0000   1st Qu.:0.000   Class :character
##  Median :28.00   Median :0.0000   Median :0.000   Mode  :character
##  Mean   :29.53   Mean   :0.4989   Mean   :0.385
##  3rd Qu.:37.00   3rd Qu.:1.0000   3rd Qu.:0.000
##  Max.   :80.00   Max.   :8.0000   Max.   :9.000
##       Fare           Cabin           Embarked
##  Min.   :  0.00   Length:1309         :  0
##  1st Qu.:  7.90   Class :character   C:272
##  Median : 14.45   Mode  :character   Q:123
##  Mean   : 33.29                      S:914
##  3rd Qu.: 31.27
##  Max.   :512.33
```

4.-Análisis de los datos

4.1.-Selección de los grupos de datos que se quieren analizar/comparar

Grupo 1

En el Modelo deñ grupo 1 utilizaremos las variables Pclass + Sex + Age + SibSp + Parch + Fare + Embarked

Grupo 2

En este grupo introduciremos una variable nueva. Obtendremos información del nombre, el título, y la incorporaremos al estudio y vemos cómo influye en el modelo.

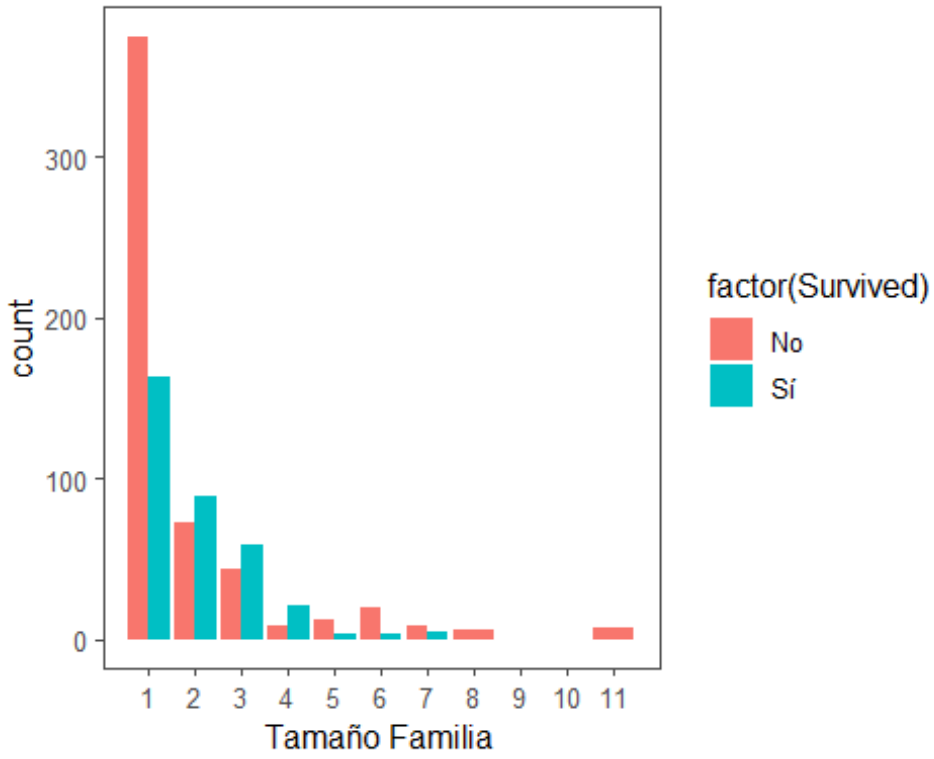
```
# Grab title from passenger names
Titanic$Title <- gsub('(.*, )|(\\..*)', '', Titanic$Name)
Titanic$Title <- as.factor(Titanic$Title)

# Show title counts by sex
table(Titanic$Sex, Titanic$Title)
```

```
##
##          Capt Col Don Dona  Dr Jonkheer Lady Major Master Miss Mlle
Mme
##   female    0   0   0    1   1        0    1     0      0  260    2
1
##   male      1   4   1    0   7        1    0     2     61    0    0
0
##
##           Mr Mrs  Ms Rev Sir the Countess
##   female   0 197   2   0   0            1
##   male   757   0   0   8   1            0
```

Grupo 3

En este grupo introduciremos una variable nueva. Obtendremos información de SibSp y Parch, las sumaremos, y la incorporaremos al estudio y vemos cómo influye en el modelo.

```
Titanic$TamanyoFamilia <- Titanic$SibSp + Titanic$Parch + 1
# Use ggplot2 to visualize the relationship between family size & survival
ggplot(Titanic[1:891,], aes(x = TamanyoFamilia, fill = factor(Survived))) +
  geom_bar(stat='count', position='dodge') +
  scale_x_continuous(breaks=c(1:11)) +
  labs(x = 'Tamaño Familia') +
  theme_few()
```



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

```
alpha = 0.05
col.names = colnames(Titanic)
for (i in 1:ncol(Titanic)) {
```

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

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

Seguidamente, pasamos a estudiar la homogeneidad de varianzas mediante la aplicación de un test de Fligner-Killeen. En este caso, estudiaremos esta homogeneidad en cuanto a los grupos conformados por los supervivientes y la tarifa pagada en el barco. En el siguiente test, la hipótesis nula consiste en que ambas varianzas son iguales

```
fligner.test( as.numeric(Survived) ~ Fare, data = Titanic)
```

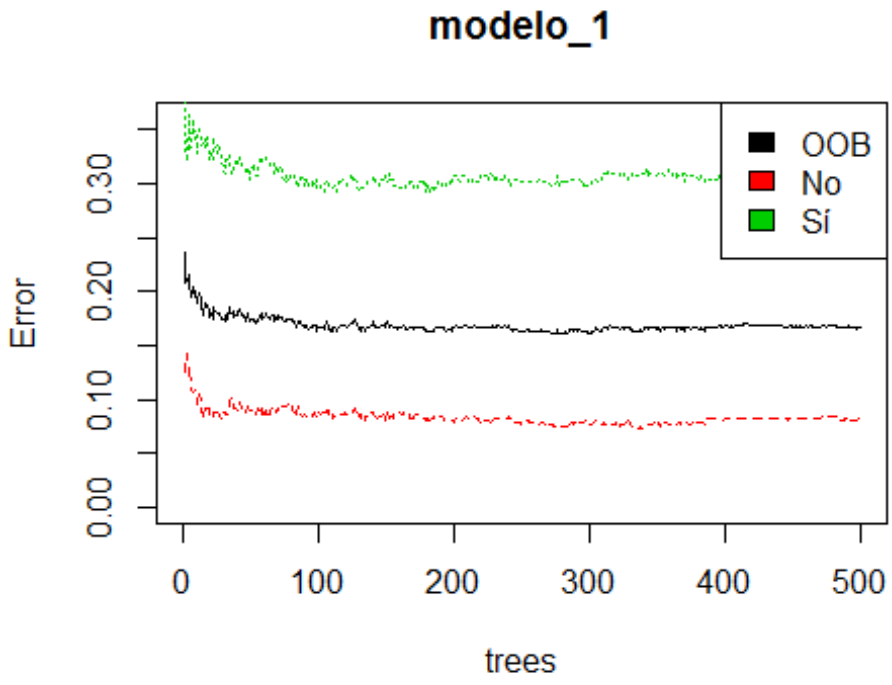
```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  as.numeric(Survived) by Fare
## Fligner-Killeen:med chi-squared = 244.13, df = 235, p-value =
## 0.3276
```

Puesto que obtenemos un p-valor superior a 0,05, aceptamos la hipótesis de que las varianzas de ambas muestras son homogéneas.

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

Modelo del grupo 1.

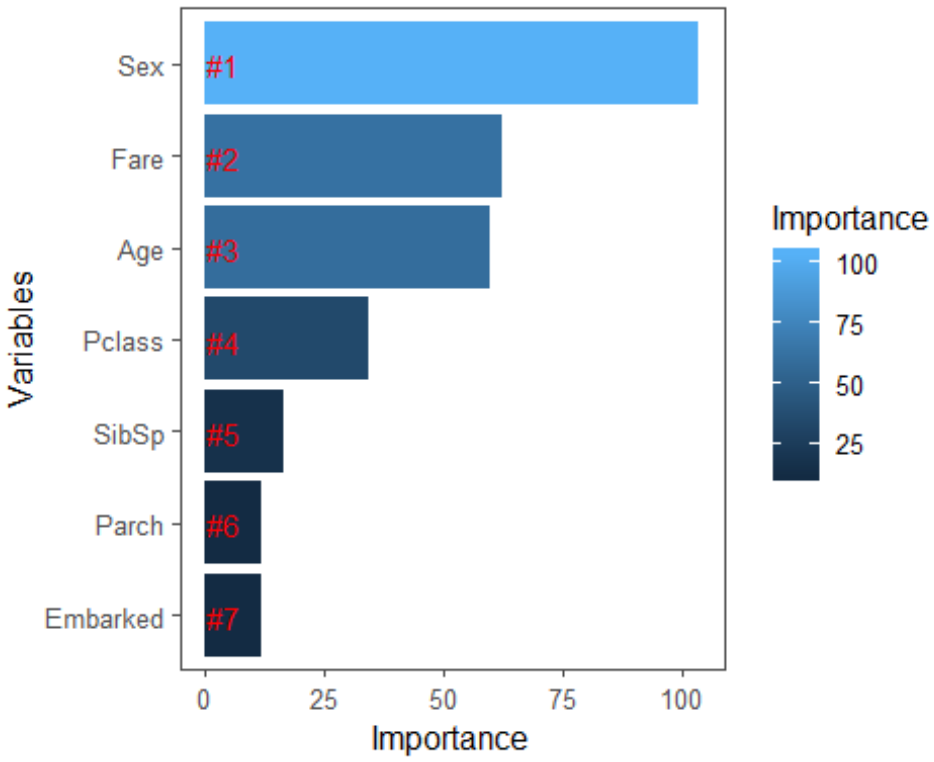
```
# Separamos los datos en su conjunto de entrenamiento y test
train <- Titanic[1:891,]
test <- Titanic[892:1309,]

# Ponemos una semilla
set.seed(754)
# Build the model (note: not all possible variables are used)
modelo_1 <- randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp +
Parch +
                                                Fare + Embarked ,
                                                data = train)

# Show model error
plot(modelo_1, ylim=c(0,0.36))
legend('topright', colnames(modelo_1$err.rate), col=1:3, fill=1:3)
```



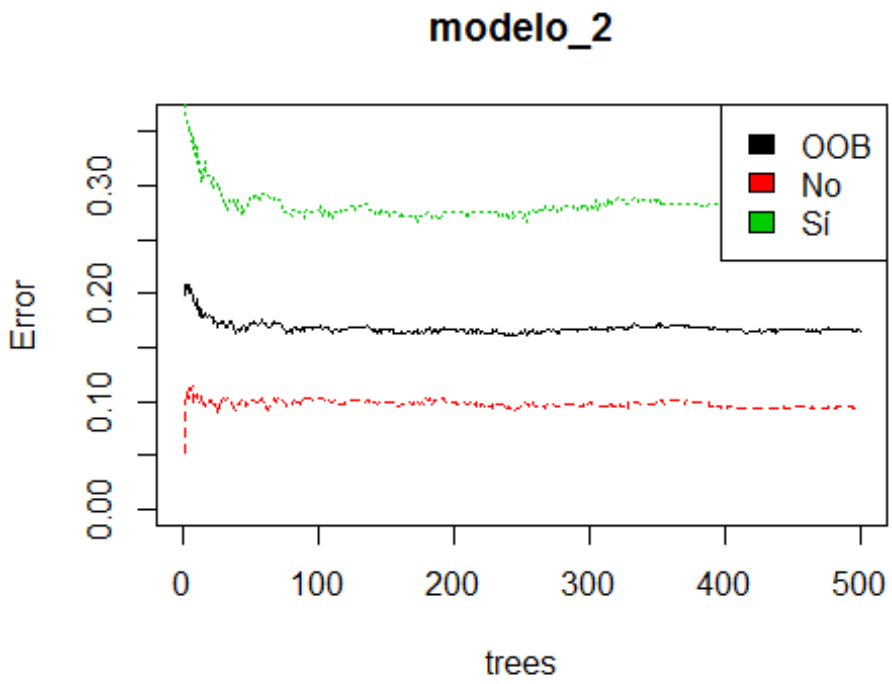
La línea negra muestra la tasa de error general que cae por debajo del 20%. Las líneas roja y verde muestran la tasa de error de “No sobrevivión” y “Sí sobrevivió” respectivamente. Podemos ver que en este momento tenemos mucho más éxito al predecir la muerte que la supervivencia.

Importancia de las variables en el modelo:

```
# Get importance
importancia_Mod_1    <- importance(modelo_1)
varImportance_Mod_1 <- data.frame(Variables =
row.names(importancia_Mod_1),
                                  Importance = round(importancia_Mod_1[
,'MeanDecreaseGini'],2))

# Create a rank variable based on importance
rankImportance_Mod_1 <- varImportance_Mod_1 %>%
  mutate(Rank = paste0('#',dense_rank(desc(Importance))))

# Use ggplot2 to visualize the relative importance of variables
ggplot(rankImportance_Mod_1, aes(x = reorder(Variables, Importance),
    y = Importance, fill = Importance)) +
  geom_bar(stat='identity') +
  geom_text(aes(x = Variables, y = 0.5, label = Rank),
    hjust=0, vjust=0.55, size = 4, colour = 'red') +
  labs(x = 'Variables') +
  coord_flip() +
  theme_few()
```

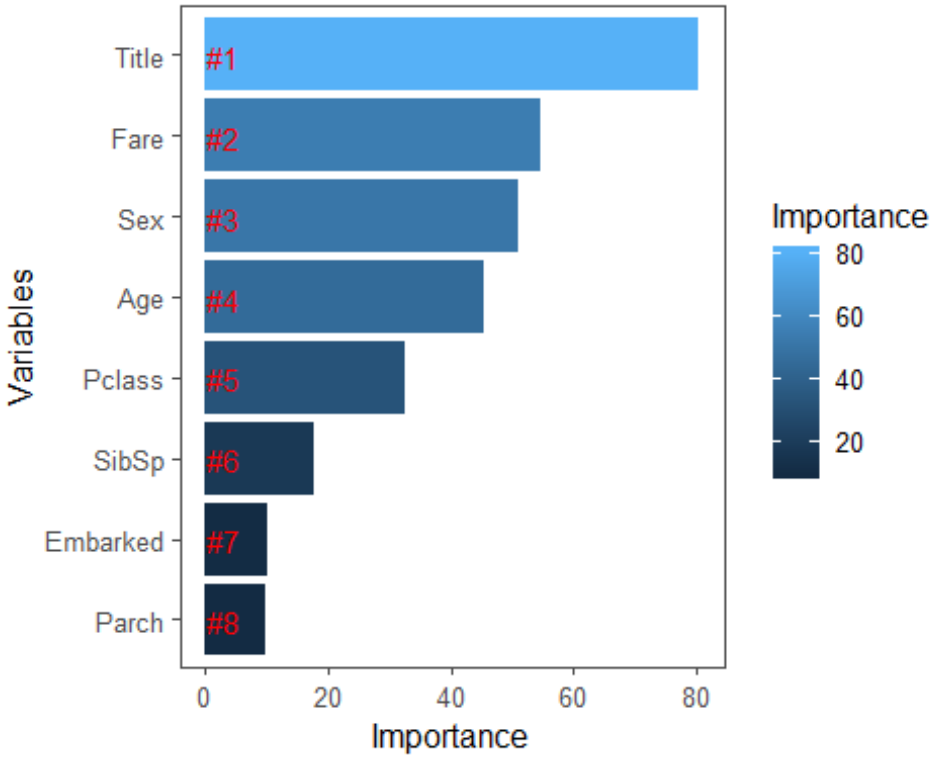


Modelo del grupo 2.



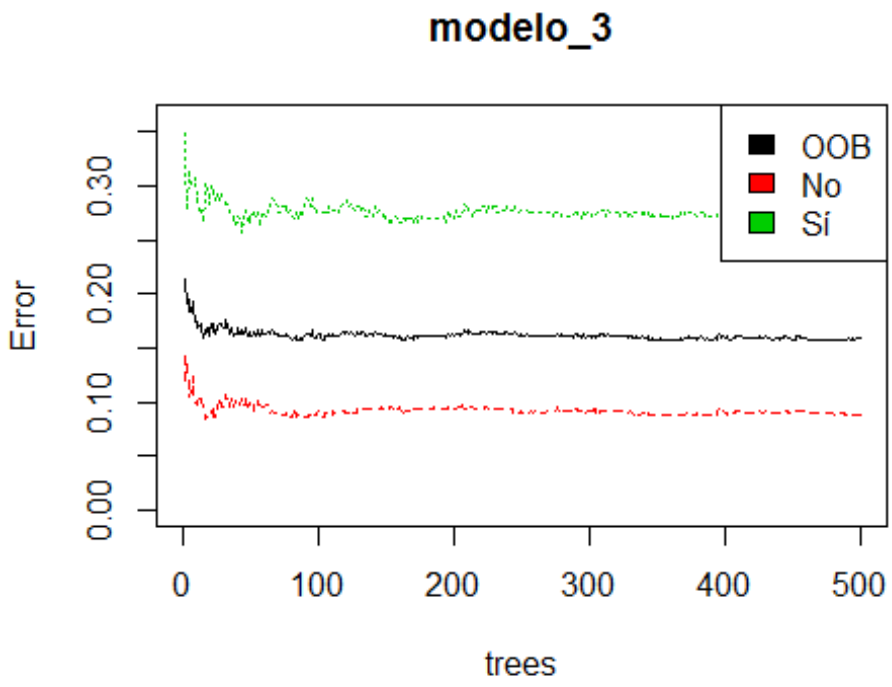
La línea negra muestra la tasa de error general que cae por debajo del 20%. Las líneas roja y verde muestran la tasa de error de “No sobrevivión” y “Sí sobrevivió” respectivamente. Podemos ver que en este momento tenemos mucho más éxito al predecir la muerte que la supervivencia.

Importancia de las variables en el modelo:



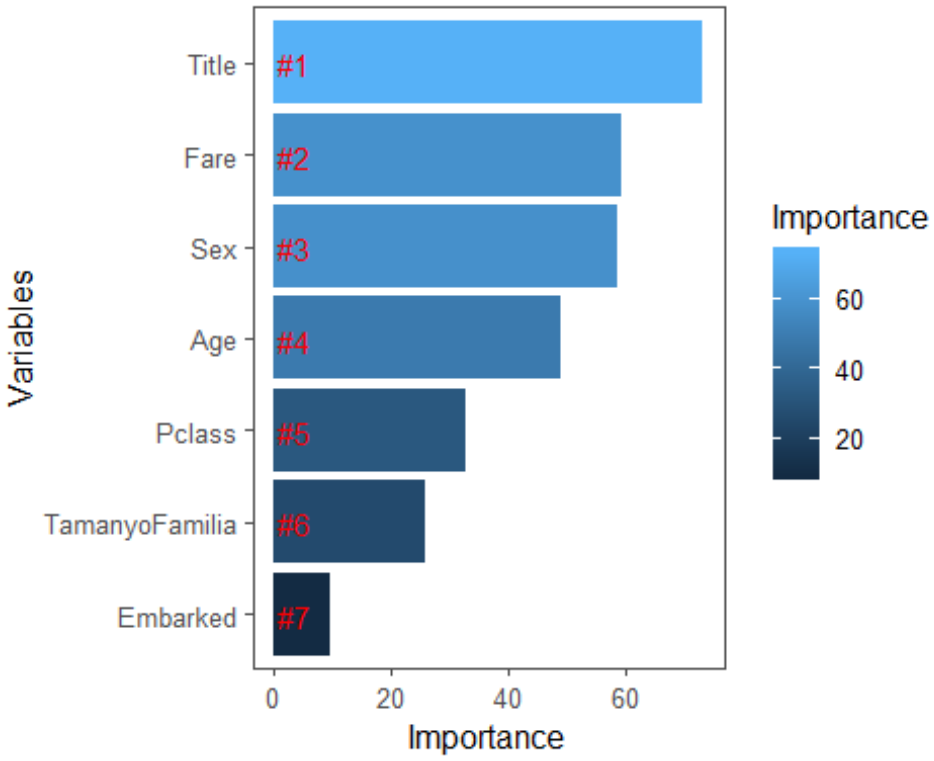
Modelo del grupo 3.

```
# Ponemos una semilla
set.seed(754)
# Build the model (note: not all possible variables are used)

modelo_3 <- randomForest(factor(Survived) ~ Pclass + Sex + Age + Fare +
Embarked + Title + TamanyoFamilia,
                                                data = train)

# Show model error
plot(modelo_3, ylim=c(0,0.36))
legend('topright', colnames(modelo_3$err.rate), col=1:3, fill=1:3)
```



La línea negra muestra la tasa de error general que cae por debajo del 20%. Las líneas roja y verde muestran la tasa de error de “No sobrevivión” y “Sí sobrevivió” respectivamente. Podemos ver que en este momento tenemos mucho más éxito al predecir la muerte que la supervivencia.

Importancia de las variables en el modelo:

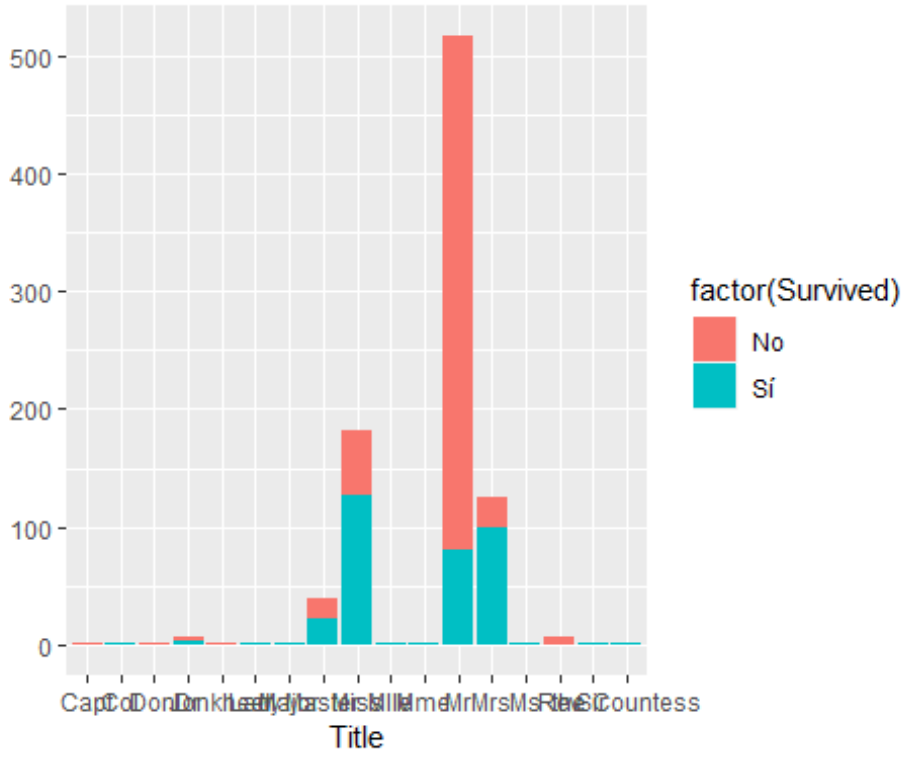


5.-Representación de los resultados a partir de tablas y gráficas

Hemos visto que las variables más importantes que deciden quien sobrevive y quien no en el naufragio del Titanic son Title, Fare, Sex and Age.

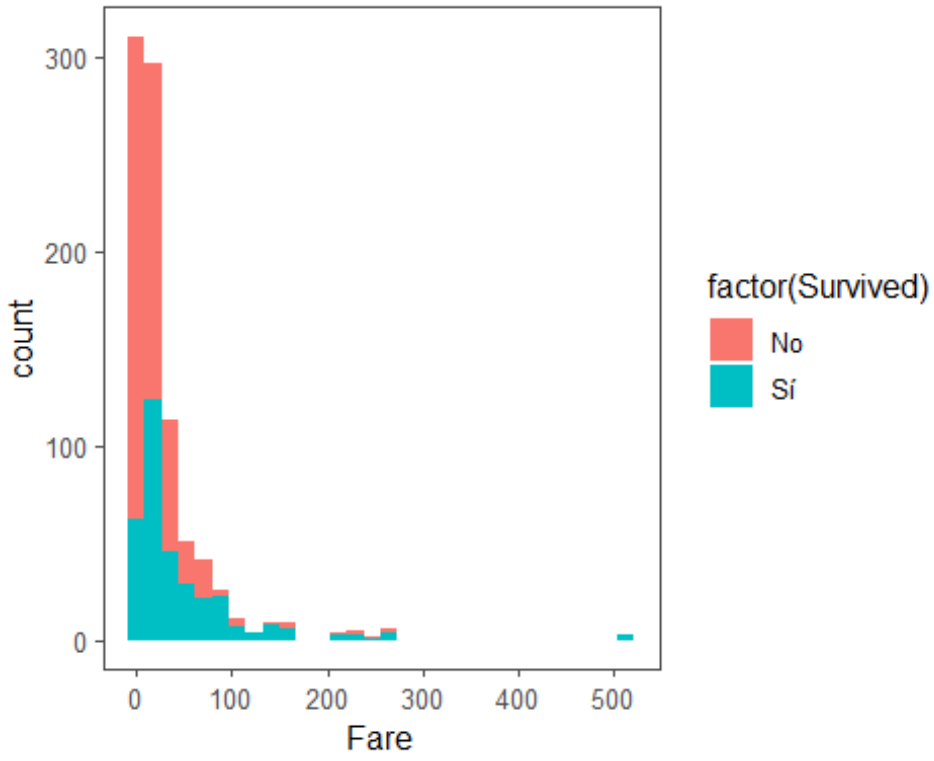
TITLE

```
qplot(Title, data = train, geom="bar",fill=factor(Survived))
```



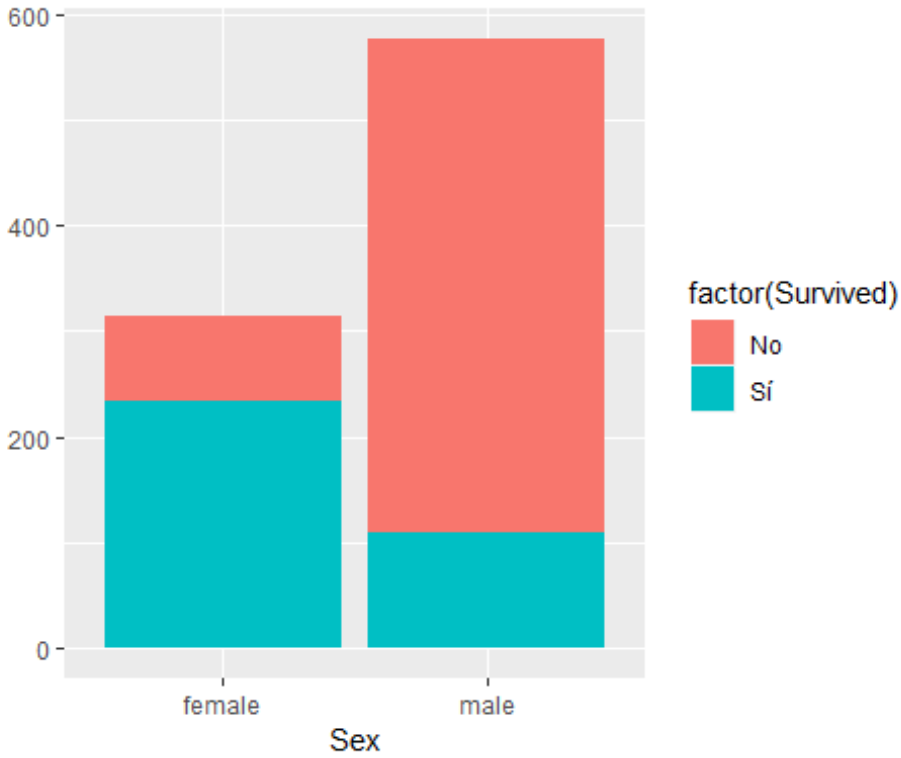
FARE

```
ggplot(train, aes(x = Fare, fill = factor(Survived))) +
  geom_histogram(bins =30) +
  theme_few()
```



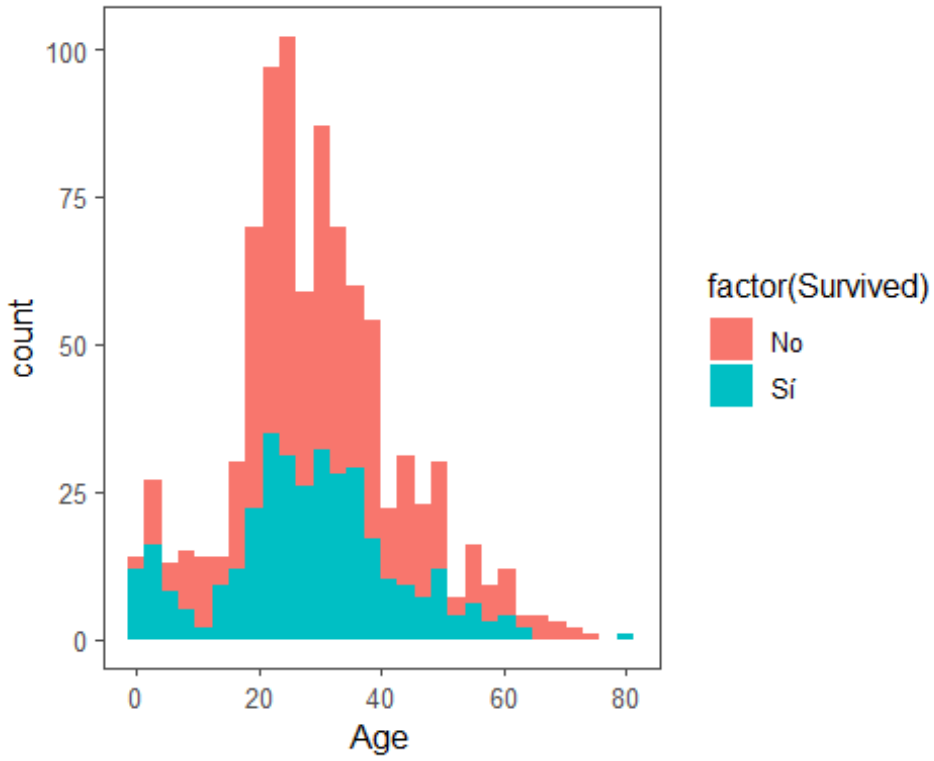
SEX

```
qplot(Sex, data = train, geom="bar",fill=factor(Survived))
```



AGE

```
ggplot(train, aes(x = Age, fill = factor(Survived))) +
  geom_histogram(bins =30) +
  theme_few()
```




```
write.csv(Titanic, file = "titanic_out.csv")
```

6.-Resolución del problema.

Con la finalidad de obtener el conjunto de datos idóneo para predecir qué pasajeros sobrevivirán o no sobre los datos proporcionados por Kaggle, hemos realizado las siguientes acciones: hemos dado un tipo adecuado a las variables tras su carga, los hemos limpiado de valores vacíos y hemos estudiados sus valores extremos(outliers). Posteriormente, hemos includio dos variable nuevas a partir de las variables existentes, Title y TamanyoFamilia. La inclusión de la variable Title, con información extraída del nombre, ha resultado decisiva, convirtiéndola en la variable más influyente. En cambio, la nueva variable TamanyoFamilia obtenida de la suma de las variables SibSp y Parch no ha influído mucho en su posición con respecto a los modelos anteriores.

7.-Bibliografía

Megan Squire (2015). Clean Data. Packt Publishing Ltd. Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann. Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369. Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media. Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc. Tutorial de Github https://guides.github.com/activities/hello-world.

Práctica 2: Limpieza y validación de los datos. Teguayco Gutiérrez González. 6 de diciembre de 2017.

PRACTICA 2: LIMPIEZA Y VALIDACIÓN DE LOS DATOS Jose Ignacio Bengoechea Isasa .7 de enero 2018.

PEC 2: Limpieza y validación de los datos. M2.851 - Tipología y ciclo de vida de los datos. Diciembre 2018. Ernesto Peralta.

Exploring Survival on the Titanic. Megan L. Risdal. 6 March 2016. https://www.kaggle.com/mrisdal/exploring-survival-on-the-titanic