PRA2_MANZANO_JULEN_CICLO_VIDA

Julen Manzano

4/22/2020

PRACTICA 2 - CICLO DE VIDA DE LOS DATOS

1.- Descripción del dataset. ¿Por qué es importante y que pregunta pretende responde?

El dataset seleccionado es Titanic: Machine Learning from Disaster, competición activa de Kaggle. Es uno de los dataset mas famosos, quizá junto al Iris-Setosa si alguno mas debiéramos mencionar, para la práctica del aprendizaje del procesado y análisis de datos.

El objetivo principal del dataset se centra en determinar mediante el resto de atributos si un pasajero del famoso RMS Titanic sobrevive o muere en el hundimiento aquella fatidica noche del 14 al 15 de abril de 1912.

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

Importamos los datos desde el archivo train proporcionado y realizamos un primer summary que nos permita observar como R ha entendido la información. (tipo de dato)

```
library(readr)
dataTitanic <- read_csv("Desktop/titanic/train.csv")
```

```
## Parsed with column specification:
## cols(
##   PassengerId = col_double(),
##   Survived = col_double(),
##   Pclass = col_double(),
##   Name = col_character(),
##   Sex = col_character(),
##   Age = col_double(),
##   SibSp = col_double(),
##   Parch = col_double(),
##   Ticket = col_character(),
##   Fare = col_double(),
##   Cabin = col_character(),
##   Embarked = col_character()
## )
```

```
summary(dataTitanic)
```

```
##   PassengerId       Survived          Pclass          Name
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000   Length:891
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median :446.0   Median :0.0000   Median :3.000   Mode  :character
##  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
##
##      Sex                 Age            SibSp           Parch
##  Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  Class :character   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##  Mode  :character   Median :28.00   Median :0.000   Median :0.0000
##                     Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                     3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                     Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                     NA's   :177
##     Ticket               Fare           Cabin             Embarked
##  Length:891         Min.   :  0.00   Length:891         Length:891
##  Class :character   1st Qu.:  7.91   Class :character   Class :character
##  Mode  :character   Median : 14.45   Mode  :character   Mode  :character
##                     Mean   : 32.20
##                     3rd Qu.: 31.00
##                     Max.   :512.33
##
```

Los datos descritos son los siguientes:

Survival: 0 = No, 1 = Sí. Determina si el pasajero sobrevivió o no al hundimiento pclass: Clase asociada al ticket, identifica mediante el 1: Primera clase, 2: Segunda clase, 3: Tercera clase. sex: Sexo del pasajero age: Edad en años del pasajero sibsp: número de esposas / hermanos que viajan junto al pasajero parch: número de padres / hijos que viajan junto al pasajero ticket: número de ticket asociado al pasaje del viajero fare: tarifa del pasajero cabin: número de cabina del pasajero embarked: puerto de embarque, C=Cherbourg, Q=Queenstown, S=Southampton

Los datos Survived, pclass, sex y embarked son de tipo factor Los datos PassengerId, Name, Ticket, Cabin son de tipo String Los datos Age, SibSp, Parch son de tipo entero Los datos Fare son de tipo Float.

Ajustamos cada atributo al tipo de dato adecuado

```
colnames(dataTitanic)
```

```
##  [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
##  [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
## [11] "Cabin"       "Embarked"
```

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

```
##   PassengerId    Survived Pclass      Name              Sex
##  Min.   :  1.0   0:549    1:216   Length:891         female:314
##  1st Qu.:223.5   1:342    2:184   Class :character   male  :577
##  Median :446.0            3:491   Mode  :character
##  Mean   :446.0
##  3rd Qu.:668.5
##  Max.   :891.0
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare           Cabin           Embarked
##  Min.   :  0.00   Length:891         C   :168
##  1st Qu.:  7.91   Class :character   Q   : 77
##  Median : 14.45   Mode  :character   S   :644
##  Mean   : 32.20                      NA's:  2
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

Creamos un nuevo dataset con aquellos atributos que nos pueden resultar interesantes para nuestro análisis.

En este caso eliminamos el PassengerId al considerar que no es necesario.

```
library(dplyr) # Cargar la librería de manipulación de dataframes "dplyr"
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
dataSelect <- select(dataTitanic, Survived, Pclass, Sex, Age, SibSp, Parch, Fare, Ticket, Cabin, Embarked)
summary (dataSelect)
```

```
##  Survived Pclass      Sex           Age            SibSp           Parch
##  0:549    1:216   female:314   Min.   : 0.42   Min.   :0.000   Min.   :0.0000
##  1:342    2:184   male  :577   1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000
##           3:491                Median :28.00   Median :0.000   Median :0.0000
##                                Mean   :29.70   Mean   :0.523   Mean   :0.3816
##                                3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##                                Max.   :80.00   Max.   :8.000   Max.   :6.0000
##                                NA's   :177
##       Fare           Ticket             Cabin           Embarked
##  Min.   :  0.00   Length:891         Length:891         C   :168
##  1st Qu.:  7.91   Class :character   Class :character   Q   : 77
##  Median : 14.45   Mode  :character   Mode  :character   S   :644
##  Mean   : 32.20                                        NA's:  2
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

3.- Limpieza de datos.

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

El primer atributo en el que detectamos valores faltantes es Age, un valor entero que contiene 177 NA's. Vamos a procurar dotarles de información imputando con la media (usamos el paquete mice)

```
library(mice)
```

```
##
## Attaching package: 'mice'
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(VIM)
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## Loading required package: data.table
```

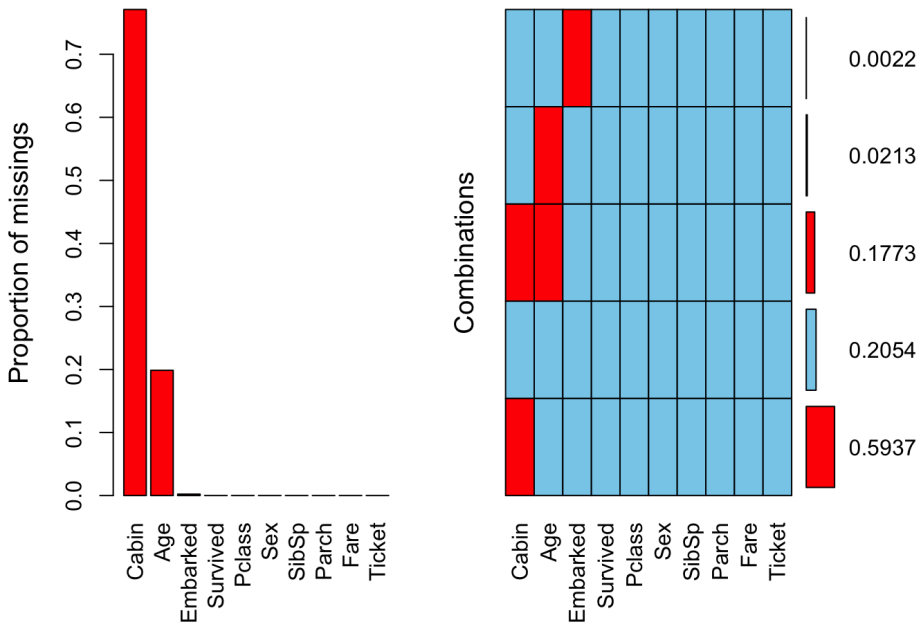
```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues
```

```
aggr(dataSelect, numbers=T,sortVar=T)
```




```
##
##  Variables sorted by number of missings:
##  Variable       Count
##     Cabin 0.771043771
##       Age 0.198653199
##  Embarked 0.002244669
##  Survived 0.000000000
##    Pclass 0.000000000
##       Sex 0.000000000
##     SibSp 0.000000000
##     Parch 0.000000000
##      Fare 0.000000000
##    Ticket 0.000000000
```

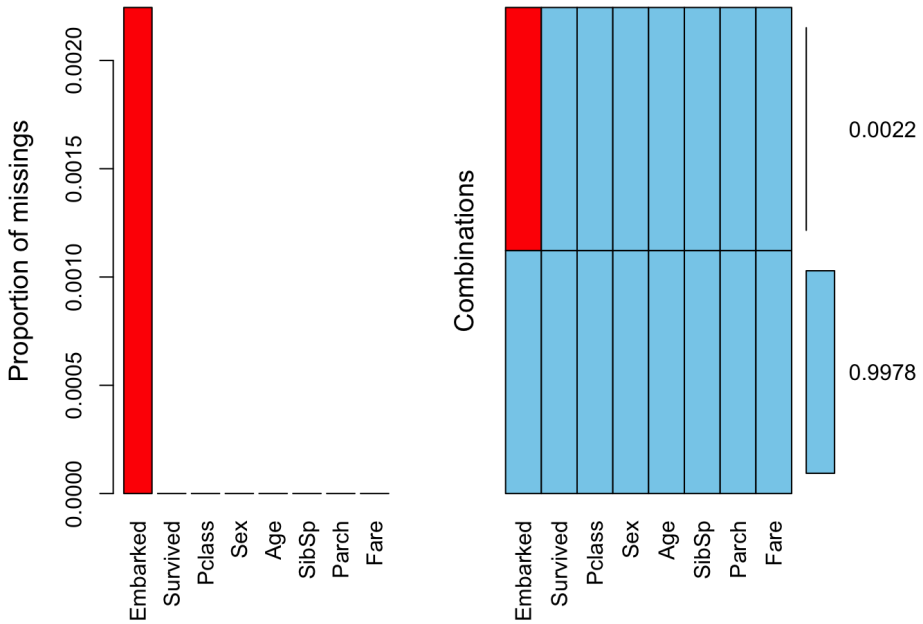
Detectamos los valor nulos en los atributos cabin, age y embarked.

Cabin es un registro poco informado que en principio no parece que nos pueda mostrar información relacionada con el caso. Como mucho podríamos pensar en utilizarla para intentar localizar valores nulos en la clase, intentando buscar la correlación entre los códigos de cabina y su ubicación en el barco. Por lo tanto, dado que pclass se encuentra plenamente informado procederemos a eliminar este atributo de nuestro dataset de estudio.

```
dataSelect <- select(dataTitanic, Survived, Pclass, Sex, Age, SibSp, Parch, Fare,  Embarked)
```

Age es un atributo entero que tiene una serie de valores no informados. Vamos a intentar imputar con la media los resultados faltantes.

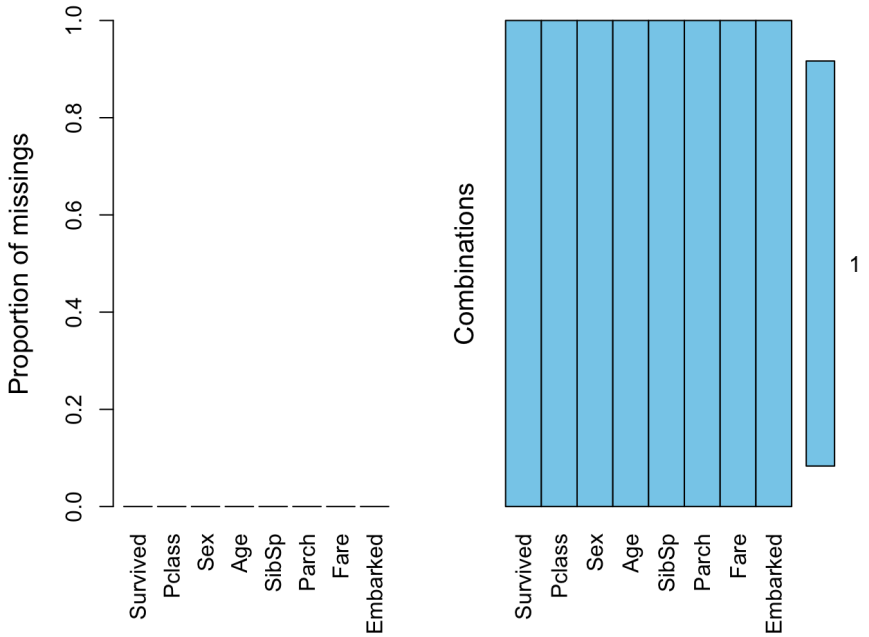
```
dataSelect[is.na(dataSelect$Age), "Age"]<-mean(dataSelect$Age, na.rm = T)
aggr(dataSelect, numbers=T,sortVar=T)
```




```
##
##  Variables sorted by number of missings:
##  Variable       Count
##  Embarked 0.002244669
##  Survived 0.000000000
##    Pclass 0.000000000
##       Sex 0.000000000
##       Age 0.000000000
##     SibSp 0.000000000
##     Parch 0.000000000
##      Fare 0.000000000
```

En el caso de Embarked es un atributo de tipo factor, en este caso sustituiremos por la moda

```
library(modeest)
moda<-mlv(dataSelect$Embarked, method ="mfv")
dataSelect[is.na(dataSelect$Embarked), "Embarked"]<-moda
aggr(dataSelect, numbers=T,sortVar=T)
```

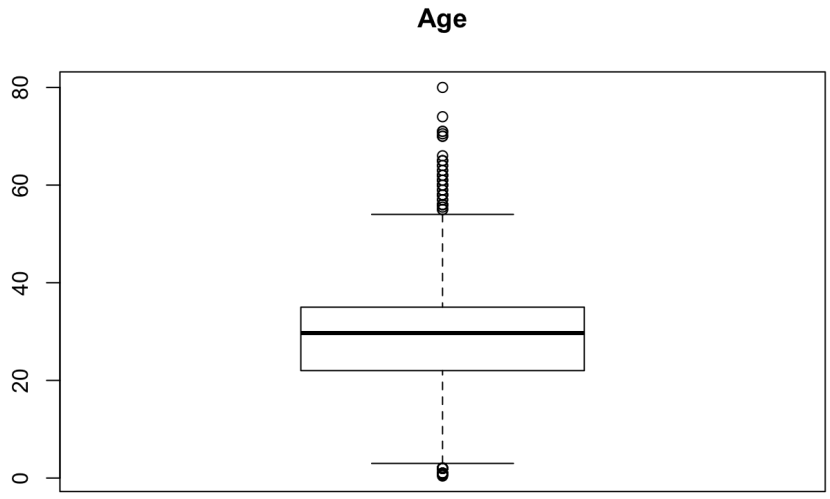



```
##
##  Variables sorted by number of missings:
##  Variable Count
##  Survived     0
##    Pclass     0
##       Sex     0
##       Age     0
##     SibSp     0
##     Parch     0
##      Fare     0
##  Embarked     0
```

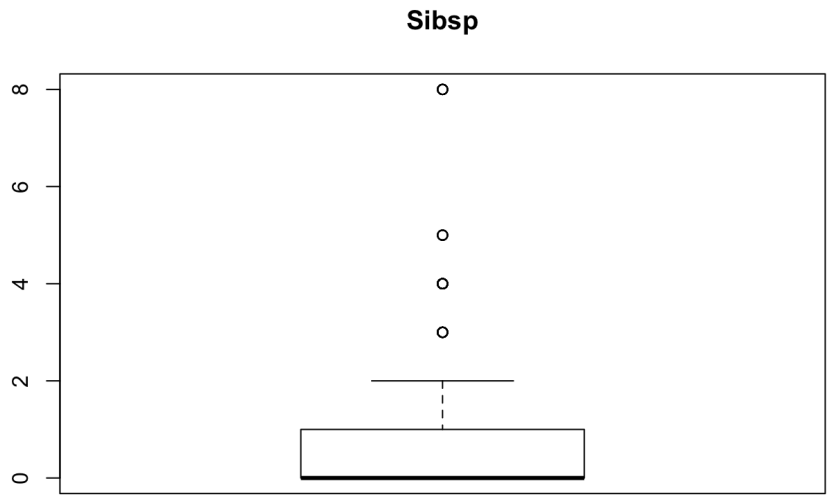
3.2. Identificación y tratamiento de valores extremos

Obseramos aquellos valores numéricos que puedan contener valores extremos mediante el uso de gráficos de cajas

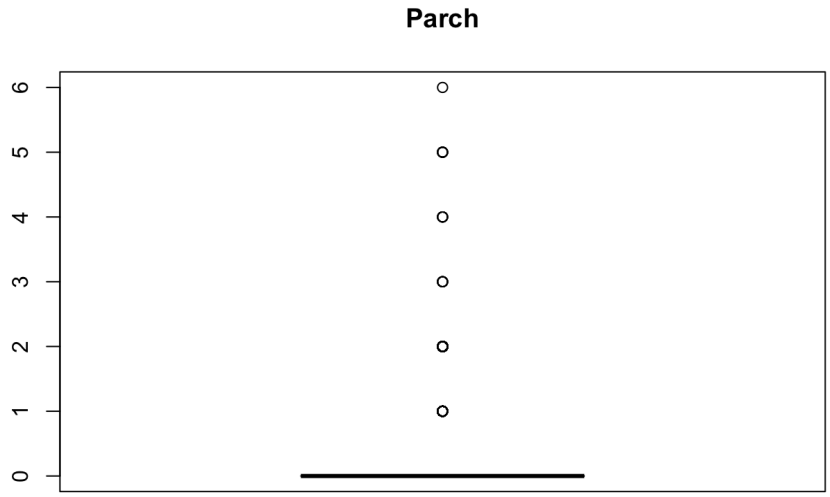
```
library(ggplot2)
boxplot(dataSelect$Age, main = "Age", outline = TRUE)
```



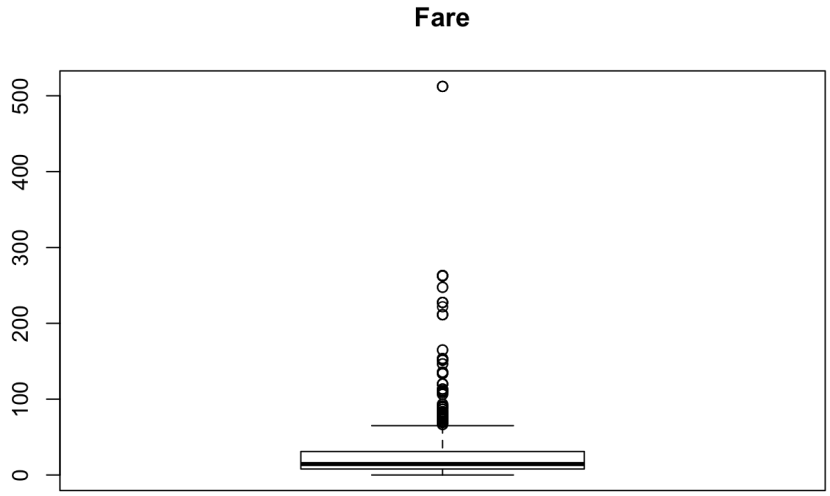

```
boxplot(dataSelect$SibSp, main = "Sibsp", outline = TRUE)
```




```
boxplot(dataSelect$Parch, main = "Parch", outline = TRUE)
```




```
boxplot(dataSelect$Fare, main = "Fare", outline = TRUE)
```




```
ggplot(dataSelect, aes(x = as.numeric(Fare))) +
  geom_density() +
  scale_x_continuous("Fare") +
  scale_y_continuous("Frecuencia") +
  labs(title = "Histograma de frecuencias",
       subtitle = "Precio del pasaje")
```

No vamos a considerar ningun valor como outlier. Todos se encuentran dentro de valores lógicos y si bien el precio del billete ha podido despertar sospechas por sus valores a cero o sus valores superiores a 500 no destacan por una situación que no pudiera darse dentro de un sentido lógico, o de un precio muy caro por un camarote o servicio extraordinario, como de precios de billete sin coste asociados a acciones promocionales o clientes corporativos.

4.- Análisis de los datos

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

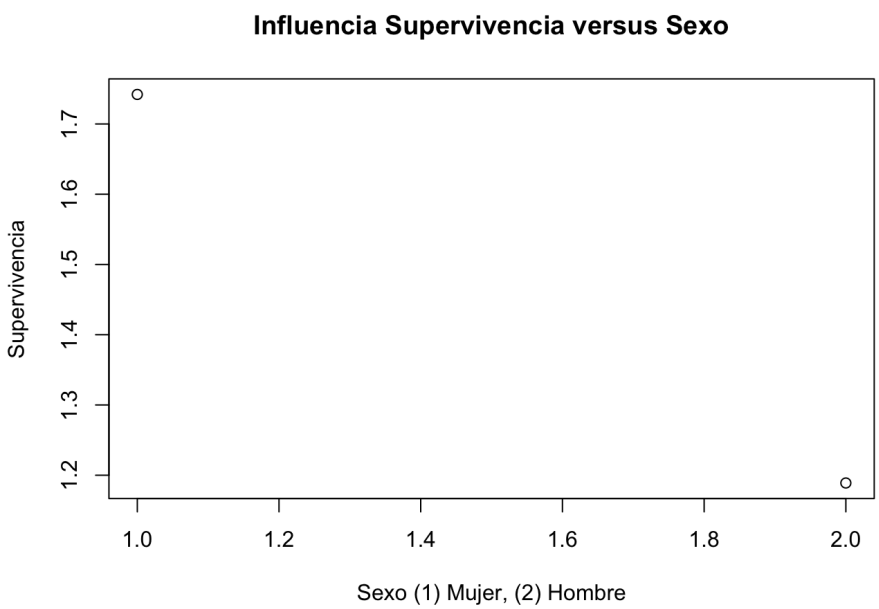
Vamos a observar la incidencia de cada atributo con el resultado de supervivencia final mediante la aplicación de gráficas que muestren en el eje x el atributo a estudiar versus el eje y que almacenará el valor de la supervivencia. Es necesario que transformemos el dataset para poder trabajar adecuadamente en estas tareas

```
newdata = dataSelect
newdata$Survived = as.integer(newdata$Survived)
newdata$Pclass = as.integer(newdata$Pclass)
newdata$Sex = as.integer(newdata$Sex)
newdata$Age = as.integer(newdata$Age)
newdata$SibSp = as.integer(newdata$SibSp)
newdata$Parch = as.integer(newdata$Parch)
newdata$Fare = as.integer(newdata$Fare)
newdata$Embarked = as.character(newdata$Embarked)
newdata$Embarked[newdata$Embarked =="C"]<-1
newdata$Embarked[newdata$Embarked =="Q"]<-2
newdata$Embarked[newdata$Embarked =="S"]<-3
newdata$Embarked = as.integer(newdata$Embarked)
summary(newdata)
```

```
##     Survived         Pclass           Sex             Age
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   : 0.00
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:22.00
##  Median :1.000   Median :3.000   Median :2.000   Median :29.00
##  Mean   :1.384   Mean   :2.309   Mean   :1.648   Mean   :29.54
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:35.00
##  Max.   :2.000   Max.   :3.000   Max.   :2.000   Max.   :80.00
##      SibSp           Parch             Fare           Embarked
##  Min.   :0.000   Min.   :0.0000   Min.   :  0.00   Min.   :1.000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:  7.00   1st Qu.:2.000
##  Median :0.000   Median :0.0000   Median : 14.00   Median :3.000
##  Mean   :0.523   Mean   :0.3816   Mean   : 31.79   Mean   :2.536
##  3rd Qu.:1.000   3rd Qu.:0.0000   3rd Qu.: 31.00   3rd Qu.:3.000
##  Max.   :8.000   Max.   :6.0000   Max.   :512.00   Max.   :3.000
```

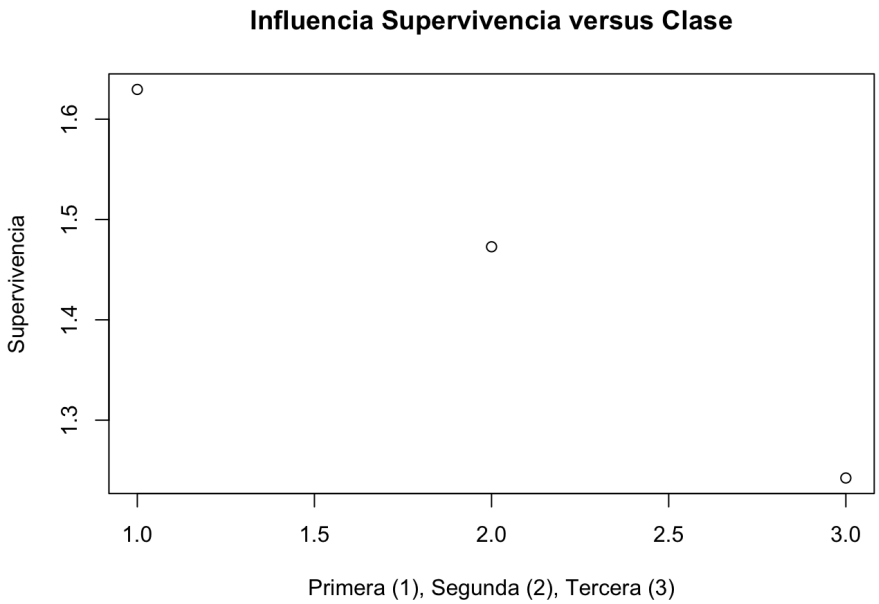
Una vez tenemos los datos con caracter numérico comenzamos a compararlas

```
vector = c(0,0)
vector[1] = mean(newdata$Survived[newdata$Sex==1])
vector[2] = mean(newdata$Survived[newdata$Sex==2])
plot(vector, main="Influencia Supervivencia versus Sexo", xlab="Sexo (1) Mujer, (2) Hombre", ylab="Supervivencia"
)
```



Como conocemos por la historia del titanic y comprobamos en los datos, el ser hombre no era buena opción para sobrevivir al incidente

```
vector = c(0,0,0)
vector[1] = mean(newdata$Survived[newdata$Pclass==1])
vector[2] = mean(newdata$Survived[newdata$Pclass==2])
vector[3] = mean(newdata$Survived[newdata$Pclass==3])
plot(vector, main="Influencia Supervivencia versus Clase", xlab="Primera (1), Segunda (2), Tercera (3)", ylab="Su
pervivencia" )
```



Respeto a la categoría del billete comprado vemos como efectivamente la supervivencia estaba fuertemente ligada a la posición económica

4.2 Comprobación de la normalidad y homogeneidad de la varianza

Utilizamos el test de Shapiro-Wilk asumiendo como hipótesis nula que la población está distribuida normalmente

```
shapiro.test(newdata$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  newdata$Age
## W = 0.95714, p-value = 1.817e-15
```

```
shapiro.test(newdata$SibSp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  newdata$SibSp
## W = 0.51297, p-value < 2.2e-16
```

```
shapiro.test(newdata$Fare)
```

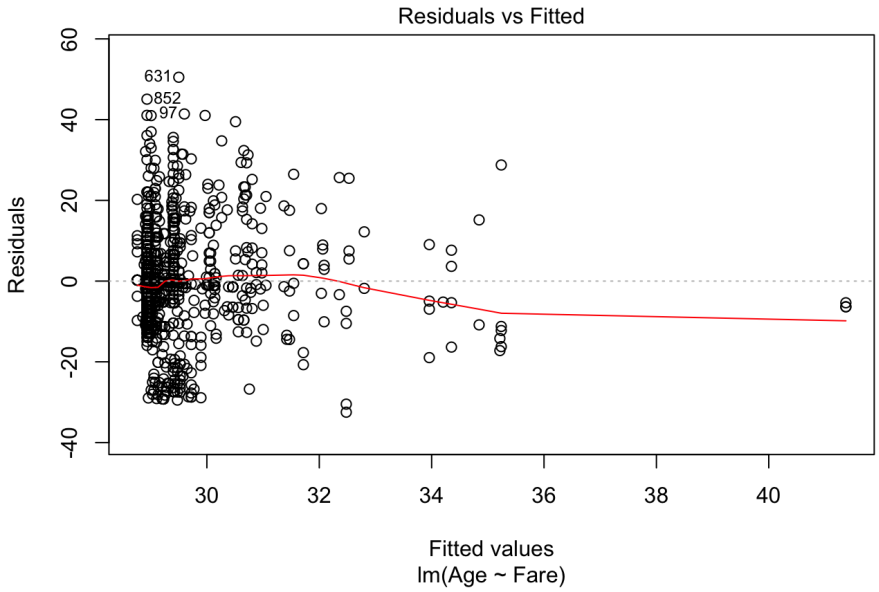
```
##
##  Shapiro-Wilk normality test
##
## data:  newdata$Fare
## W = 0.52232, p-value < 2.2e-16
```

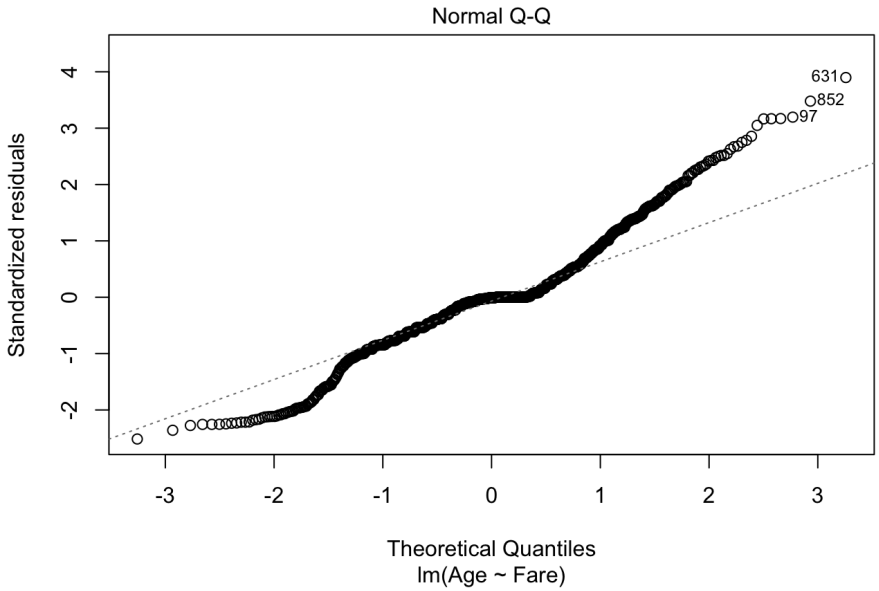
Observamos que los datos no reflejan una distribución normal

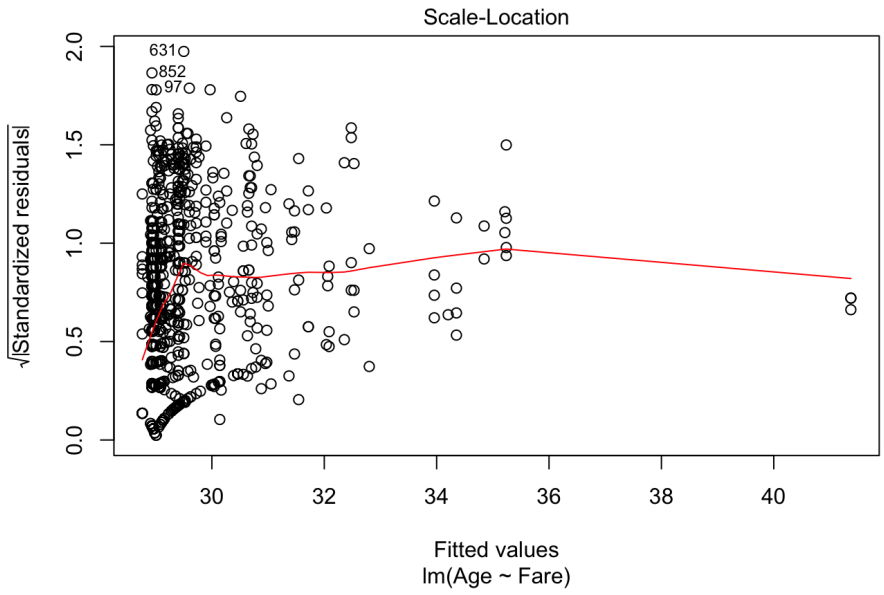
4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio aplicar pruebas de contrastes de hipótesis, correlaciones, regresiones, etc.

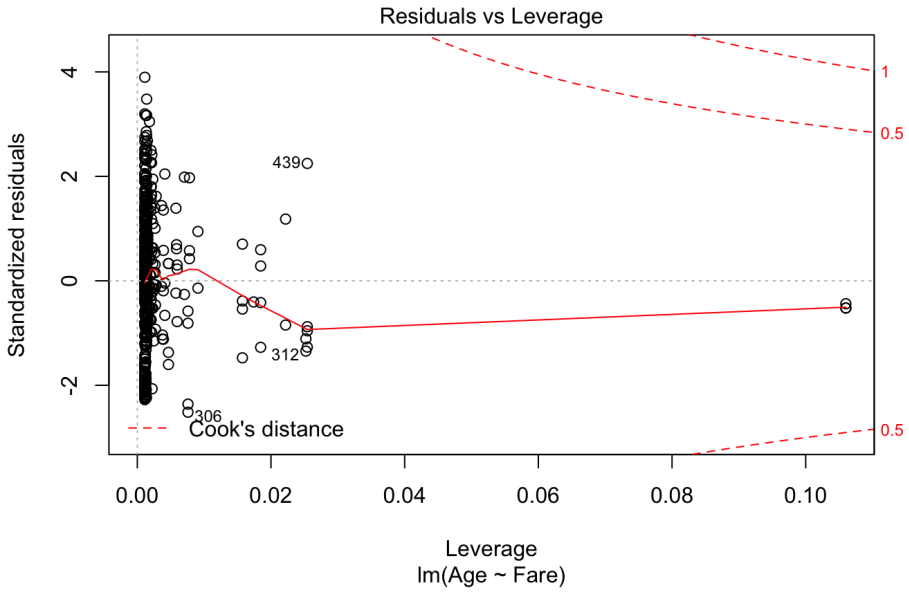
a. Regresión: Probaremos la relación entre la edad y el coste del billete

```
modelo <-lm(Age~Fare, data=newdata)
plot(modelo)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Age ~ Fare)
631
852
97


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Age ~ Fare)
631
852
97


Scale-Location
√|Standardized residuals|
Fitted values
lm(Age ~ Fare)
631
852
97




```
summary(modelo)
```

```
##
## Call:
## lm(formula = Age ~ Fare, data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.481  -6.958  -0.131   5.204  50.500
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.761456   0.515584  55.784  < 2e-16 ***
## Fare         0.024630   0.008742   2.817  0.00495 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.96 on 889 degrees of freedom
## Multiple R-squared:  0.008849,   Adjusted R-squared:  0.007734
## F-statistic: 7.937 on 1 and 889 DF,  p-value: 0.004951
```

b. Estudio de correlaciones

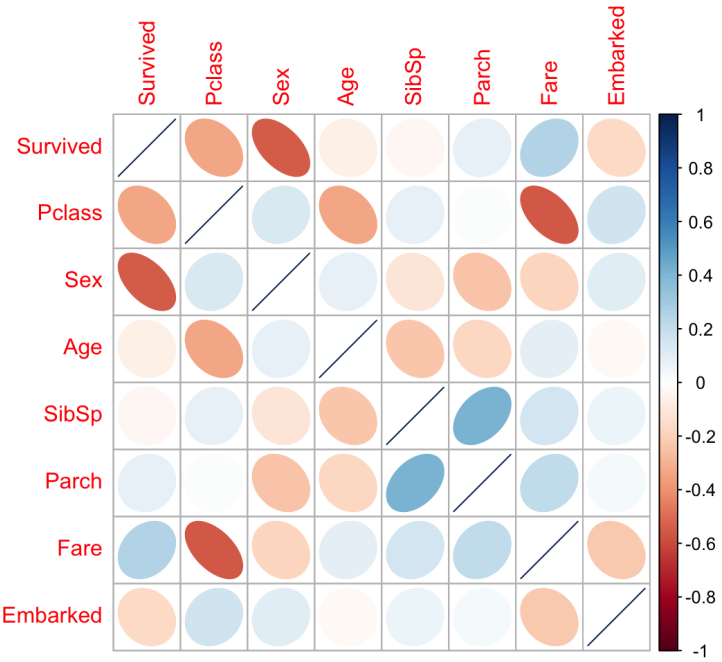
```
cor.test (newdata$Survived, newdata$Pclass)
```

```
##
##  Pearson's product-moment correlation
##
## data:  newdata$Survived and newdata$Pclass
## t = -10.725, df = 889, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3953692 -0.2790061
## sample estimates:
##       cor
## -0.338481
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
correlacion <- cor(newdata)
corrplot(correlacion, method = "ellipse")
```



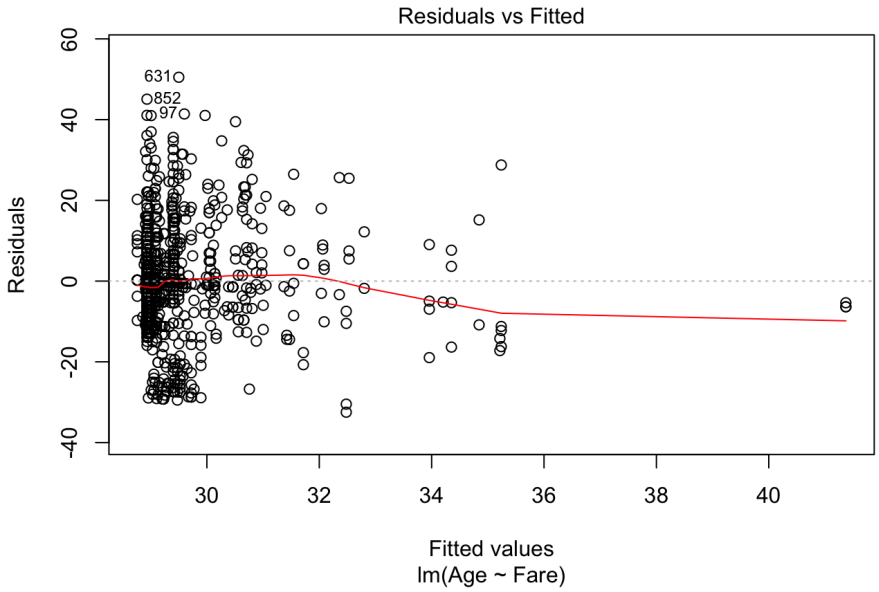
c. Uso de KMEANS para realizar agrupaciones basadas en supervivencia.

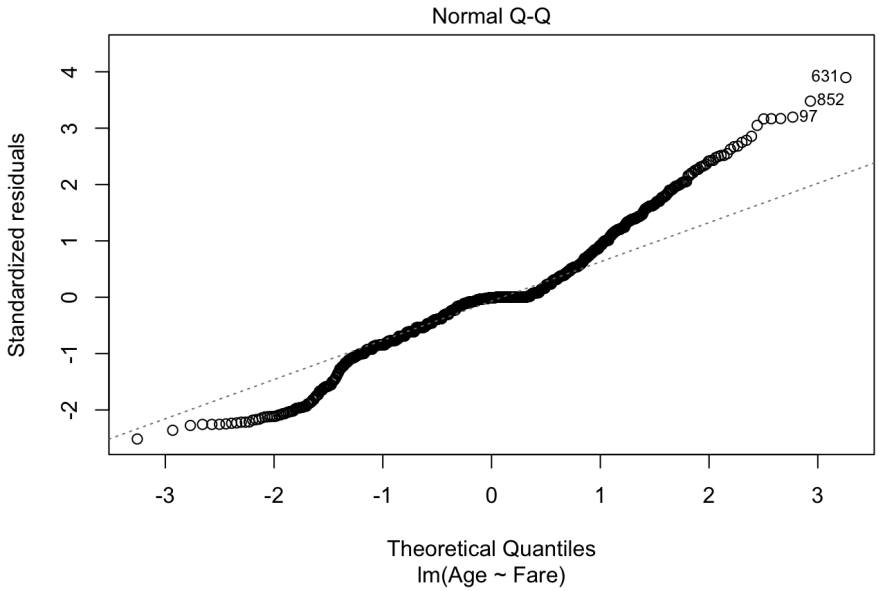
```
newdata.cl<-newdata
newdata.cl$Survived<-NULL
kmeans.res<-kmeans(newdata.cl,3)
table(newdata$Survived,kmeans.res$cluster)
```

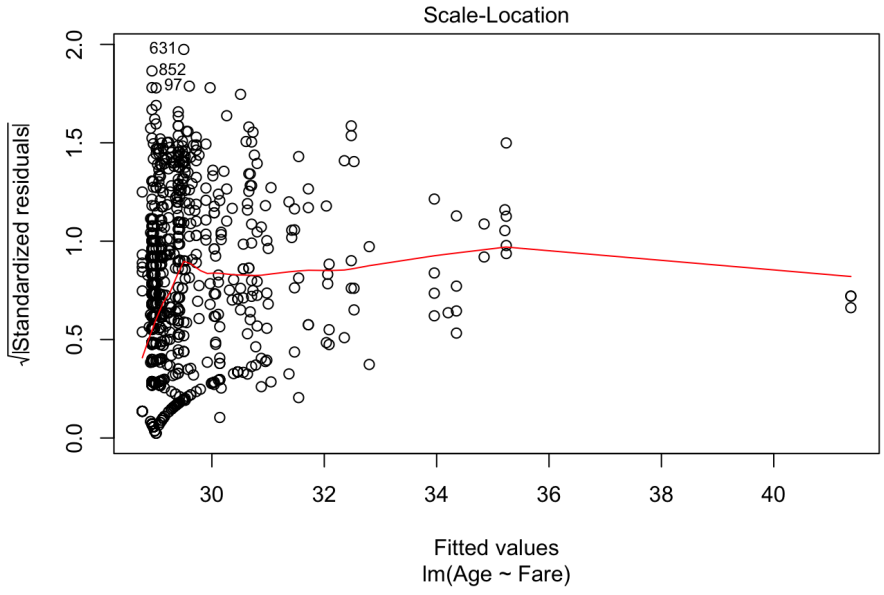
```
##
##        1   2   3
##    1  47   6 496
##    2  95  14 233
```

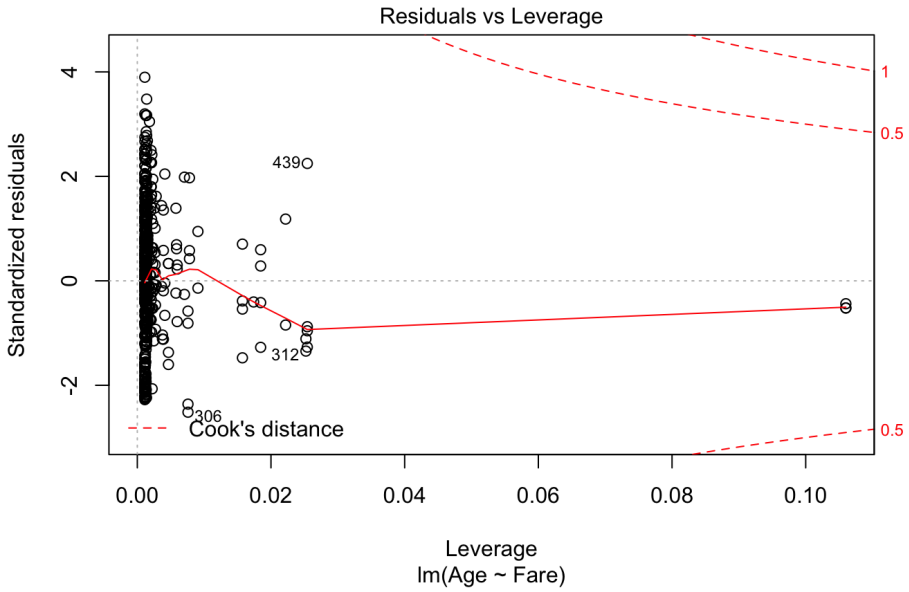
5. Representación de los resultados a partir de las tablas y gráficas

```
plot(modelo)
```

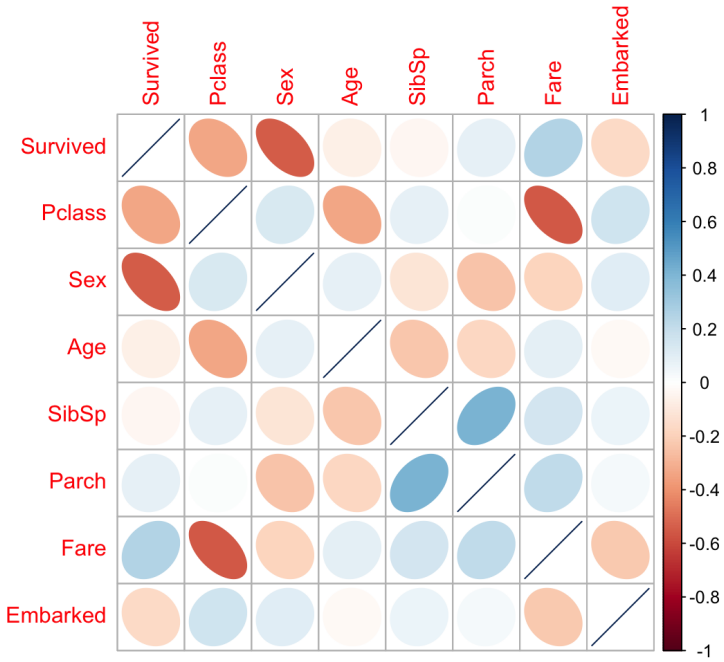

Residuals vs Fitted
Residuals
Fitted values
lm(Age ~ Fare)
631
852
97


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Age ~ Fare)
631
852
97


Scale-Location
√|Standardized residuals|
Fitted values
lm(Age ~ Fare)
631
852
97




```
corrplot(correlacion, method = "ellipse")
```



6. Resolución del problema. A partir de lois resultados obtenidos, ¿Cuales son las conclusiones?

Los datos no muestran una distribución normal. Existe una relación directa entre los supervivientes y el sexo o la clase en la que los pasajeros viajaban tal y como a priori podemos imaginar por el conocimiento previo de la historia del hundimiento.

Los datos pueden por sus relaciones posibilitar la generación de modelos de comportamiento que en base al resto de parámetros nos permita conocer la capacidad de un viajero de sobrevivir o de no hacerlo.

Los datos tenían una corrrecta calidad para su preparación y uso en la analítica de datos.