Práctica 2: Limpieza y validación de los datos

Realizado por Álvaro Pavón Díaz (apavond)

1. Intencionalidad del conjunto de datos

La práctica va a ser realizada en el entorno estadístico R.
El dataset que vamos a utilizar en el desarrollo de la práctica contiene datos sobre los pasajeros del Titanic. El objetivo de la práctica es decidir si los factores personales y ecónomicos influyeron en la supervivencia de ellos y la creación de modelos para predecir si una persona que no estaba en el barco en el momento del hundimiento hubiera sobrevivido. Este conjunto de datos esta alojado el la página kaggle en el siguiente enlace: https://www.kaggle.com/c/titanic
Pasamos a importar el archivo con extensión csv llamado train:

```
> Dataset <-
+
+   read.table("F:/Google Drive/Master cuatrimeste 1 de 1819/Tipologia y Ciclo de Vida de los datos/PRA2/all/t
+
+    header=TRUE, sep=",", na.strings="NA", dec=".", strip.white=TRUE)
```

2. Descripción del conjunto de datos

El cojunto de datos que analizaremos a lo largo de esta práctica consta de 10 atributos y con 891 registros. Vamos a describir los diferentes campos:

- **PasangerId**: nos indica el número de registro en el que estamos.
- **Suvirved**: si el pasajero sobrevive como 1 y si el pasajero no sobrevive como 0.
- **Pclase**: con un número nos indica si el pasaje era de primera, segunda o tercera clase.
- **Name**: nombre del pasajero.
- **Sex**: con dos palabras define el sexo del pasajero male para hombre y female para mujer
- **Age**: edad del pasajero en años
- **Sibsp**: el número de hermanos y conyuges abordos del Titanic.
- **Parch**: el número de padres e hihos abordos del Titanic.
- **Ticket**: identificador alfanumérico del ticket con el que embarcaron.
- **Fare**: precio del ticket.
- **Cabin**: identificador alfanumérico de la cabina.
- **Embarked**: puerto de embarcación del pasajero identificado con una letra. Siendo C para Cherbourg, Q para Queenstown y S para Southampton

Veamos un resumen de los datos que se encuentran en el conjunto de datos:

```
> summary(Dataset)
```

```
 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
```

3. Limpieza de los datos.

Vamos a comenzar viendo la clase que tienen tras la importación del conjunto de datos:

```
> sapply( Dataset, class)
```

```
PassengerId    Survived      Pclass        Name         Sex         Age
  "integer"   "integer"   "integer"    "factor"    "factor"   "numeric"
      SibSp       Parch      Ticket        Fare       Cabin    Embarked
  "integer"   "integer"    "factor"   "numeric"    "factor"    "factor"
```

Para el futuro estudio de los datos en R los atributos Survived y Pclass vamos a pasarlas de integer a factor. Las añadiremos en nuevos atributosdentro del data.frame que ya teníamoos creados.

```
> Dataset <- within(Dataset, {
+   SurvivedF <- as.factor(Survived)
+ })
> Dataset <- within(Dataset, {
+   PclassF <- as.factor(Pclass)
+ })
```

Eliminaremos los atributos:

- **PasangerId**: por ser el número de la fila no nos aporta nada.
- **Survived**: se pasa a factor como SurvivedF
- **Pclass**: se pasa a factor como PclassF
- **Name**: no aporta nada estadísticamente hablando
- **Ticket**: el identificador del ticket no nos dice nada.
- **Cabin**: Tampoco es interesante por la cantidad de campos vacíos.

```
> Dataset<-Dataset[, -(1:4)]
> Dataset<-Dataset[, -(5:5)]
> Dataset<-Dataset[, -(6:6)]
```

Vamos a analizar los elementos vacíos que nos encontramos:

```
> sapply(Dataset, function(x) sum(is.na(x)))
```

```
     Sex       Age     SibSp     Parch      Fare  Embarked SurvivedF
       0       177         0         0         0         0         0
  PclassF
       0
```

Se da la situación de que tenemos 177 elementos con vacíos en el atributo Age. Se va a dejar como elementos nulos. Si hubieran sido menos se hubiera intentado rellenarlos con algún método, como podría haber sido el método de los vecinos más próximos.
Vimos en el resumen que había otro elemento de los que quedan que tiene elementos nulos. Es Embarked. Pero no lo contabiliza por ser '' y NA. Así que generaremos una función para contarlas. Que serán 2 y posteriormente se eliminarán tanto los registro como el level del factor Embarked.

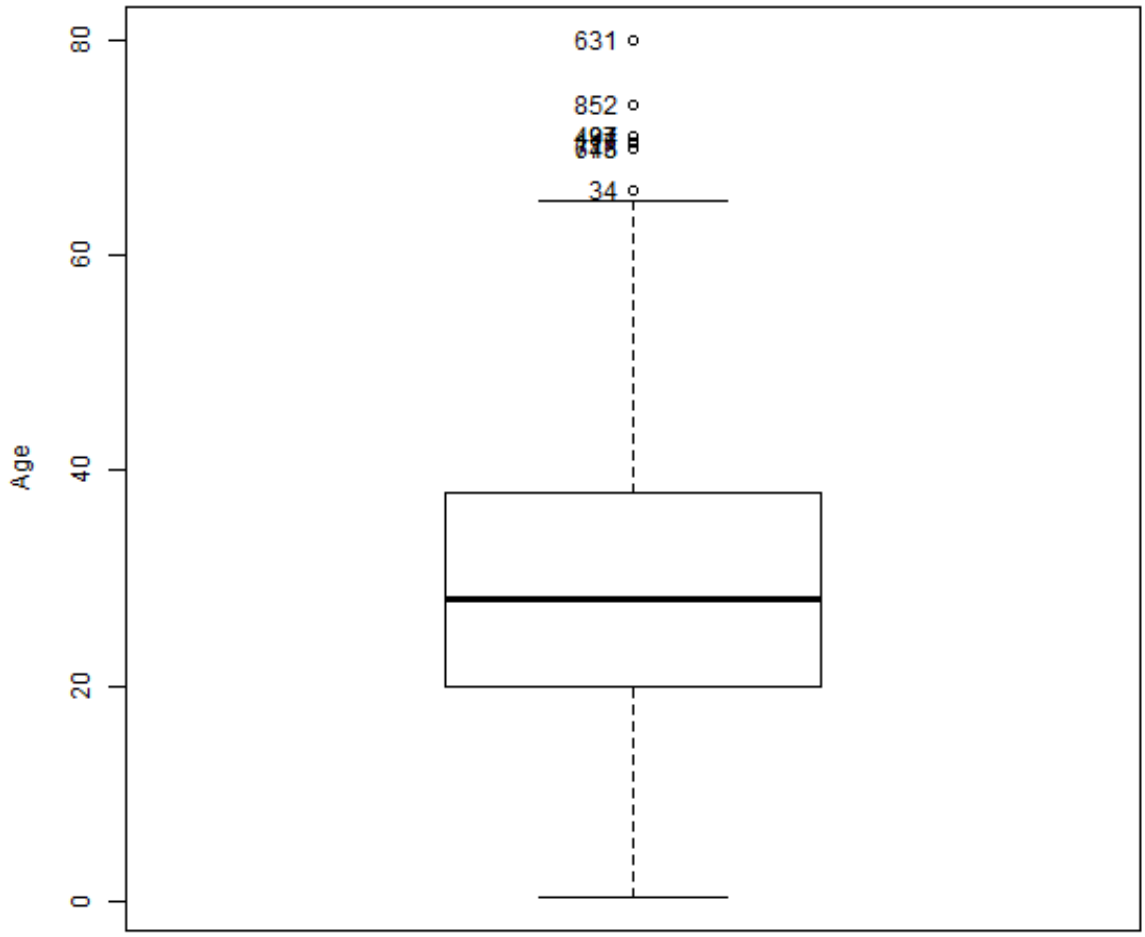
```
> stringEmptyEmbarked <- function (x){
+   result<-0
+   if (x==''|| x==Dataset[62,'Embarked']){
+    result<-result+1
+   }
+   return(result)
+ }
> sum(sapply( Dataset[,'Embarked'], stringEmptyEmbarked))
```

```
[1] 2
```

```
> Dataset <- Dataset[Dataset$Embarked != Dataset[62,'Embarked'],]
> Dataset$Embarked <- Dataset$Embarked[, drop=TRUE]
```

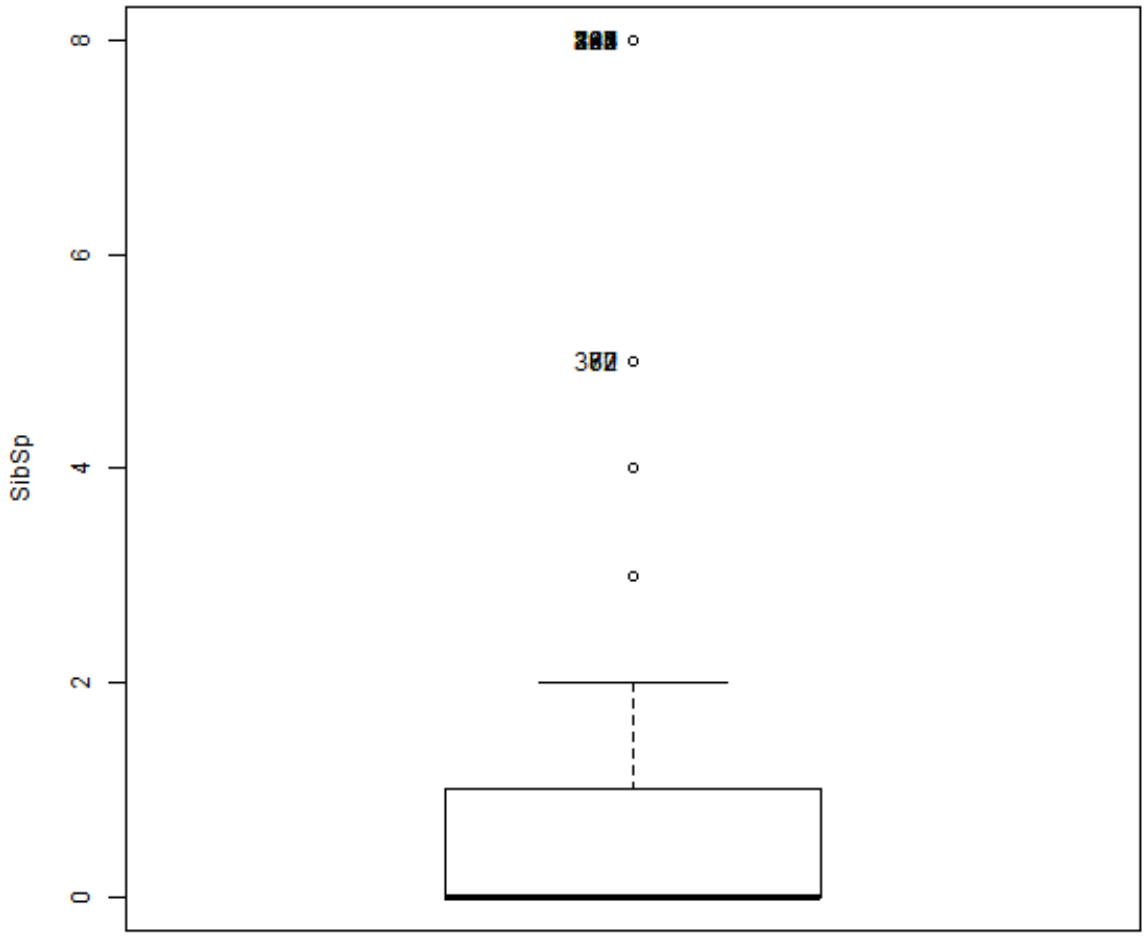
Veamos los diagramas de cajas de los atributos numéricos :

```
> Boxplot( ~ Age, data=Dataset, id=list(method="y"))
```



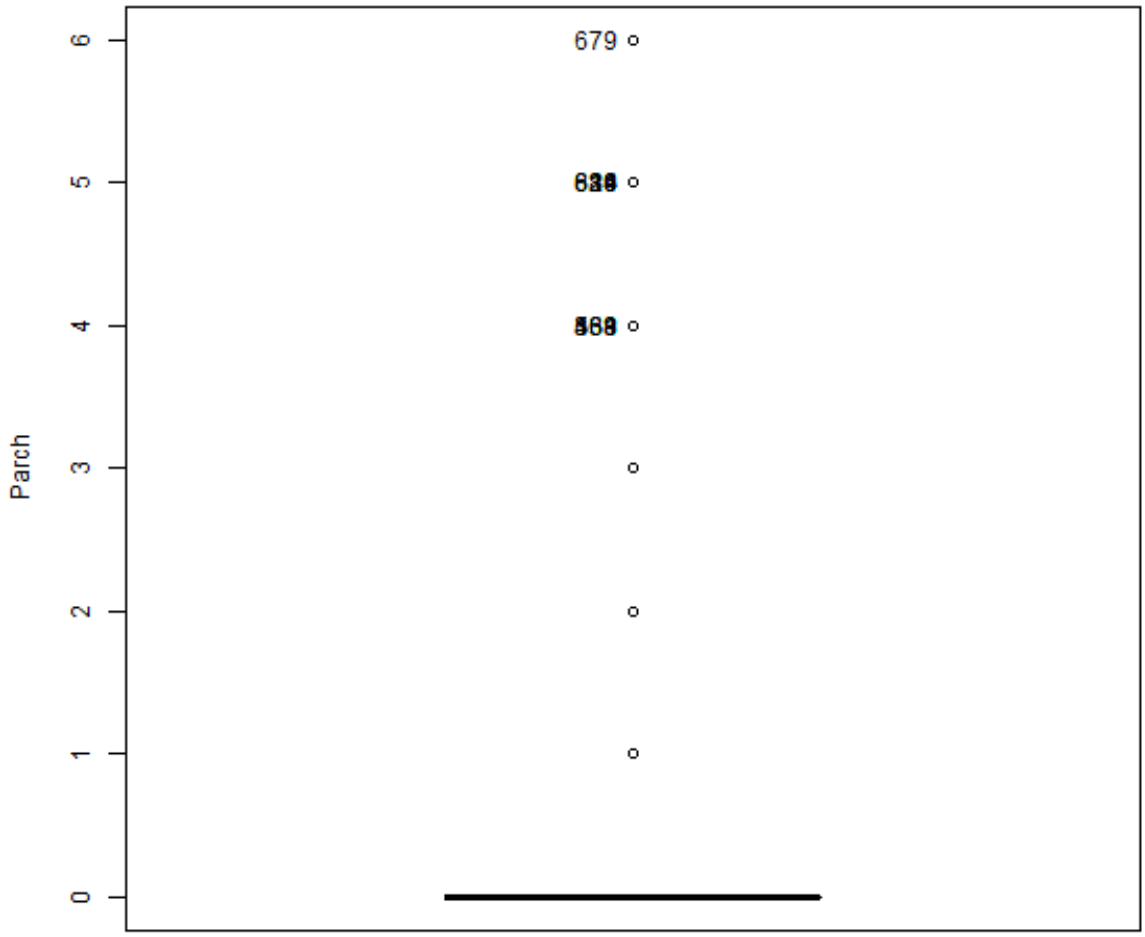

```
[1] "34"  "97"  "117" "494" "631" "673" "746" "852"
```

```
> Boxplot( ~ SibSp, data=Dataset, id=list(method="y"))
```



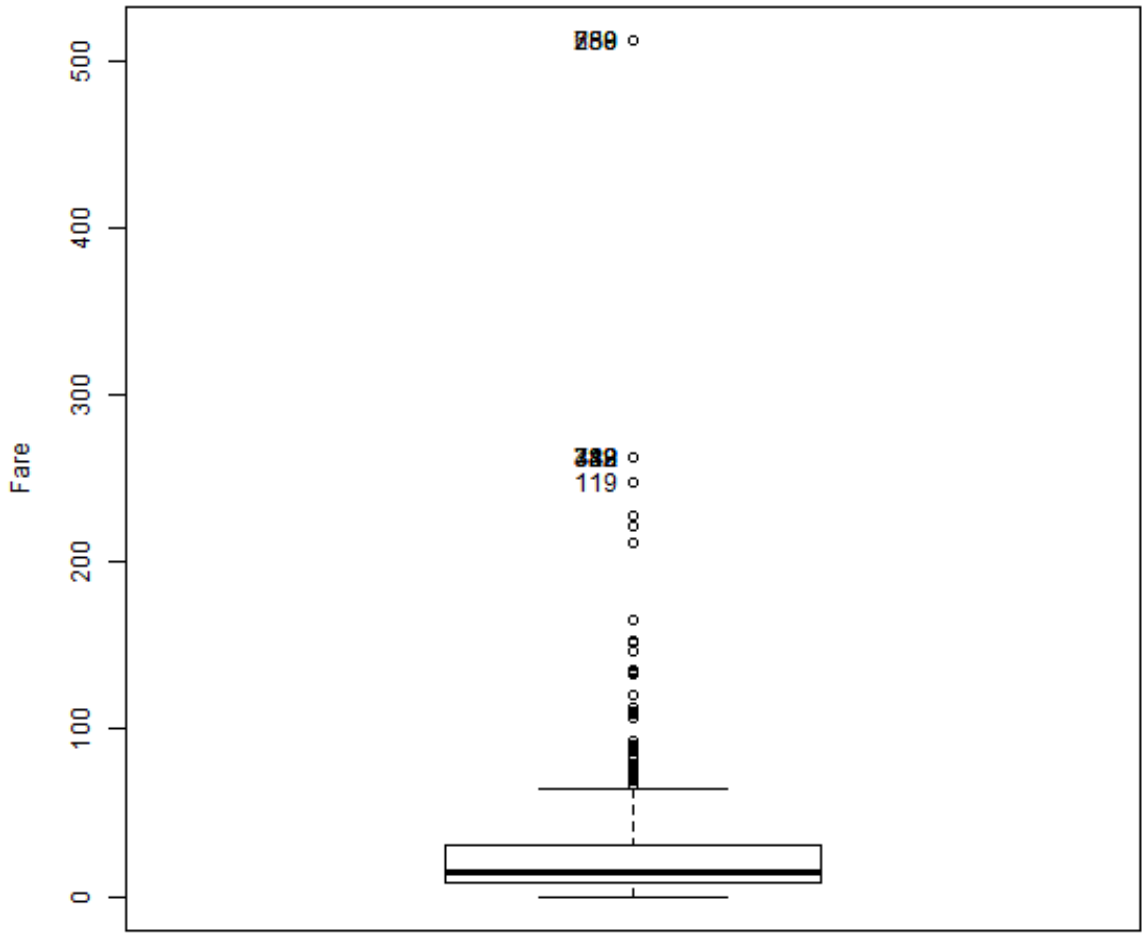

```
 [1] "160" "181" "202" "325" "793" "847" "864" "60"  "72"  "387"
```

```
> Boxplot( ~ Parch, data=Dataset, id=list(method="y"))
```




```
 [1] "679" "14"  "26"  "611" "639" "886" "168" "361" "439" "568"
```

```
> Boxplot( ~ Fare, data=Dataset, id=list(method="y"))
```




```
 [1] "259" "680" "738" "28"  "89"  "342" "439" "312" "743" "119"
```

Ahora veamos los elementos que se quedan fuera de los diagramans de cajas.

```
> boxplot.stats(Dataset$Age)$out
```

```
[1] 66.0 71.0 70.5 71.0 80.0 70.0 70.0 74.0
```

```
> boxplot.stats(Dataset$SibSp)$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
```

```
> boxplot.stats(Dataset$Parch)$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
```

```
> boxplot.stats(Dataset$Fare)$out
```

```
  [1]  71.2833 263.0000 146.5208  82.1708  76.7292  83.4750  73.5000
  [8] 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000  66.6000
 [15]  69.5500  69.5500 146.5208  69.5500 113.2750  76.2917  90.0000
 [22]  83.4750  90.0000  79.2000  86.5000 512.3292  79.6500 153.4625
 [29] 135.6333  77.9583  78.8500  91.0792 151.5500 247.5208 151.5500
 [36] 110.8833 108.9000  83.1583 262.3750 164.8667 134.5000  69.5500
 [43] 135.6333 153.4625 133.6500  66.6000 134.5000 263.0000  75.2500
 [50]  69.3000 135.6333  82.1708 211.5000 227.5250  73.5000 120.0000
 [57] 113.2750  90.0000 120.0000 263.0000  81.8583  89.1042  91.0792
 [64]  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000 221.7792
 [71] 106.4250  71.0000 106.4250 110.8833 227.5250  79.6500 110.8833
 [78]  79.6500  79.2000  78.2667 153.4625  77.9583  69.3000  76.7292
 [85]  73.5000 113.2750 133.6500  73.5000 512.3292  76.7292 211.3375
 [92] 110.8833 227.5250 151.5500 227.5250 211.3375 512.3292  78.8500
 [99] 262.3750  71.0000  86.5000 120.0000  77.9583 211.3375  79.2000
[106]  69.5500 120.0000  93.5000  83.1583  69.5500  89.1042 164.8667
[113]  69.5500  83.1583
```

A pesar de que puedan parecer muchos los elementos que quedan fueras de los diagramas de caja no parecen que sean datos que esten fuera de rango según los datos analizados.
Para finalizar el estudio de la limpieza de datos veamos el resumende como han quedado los datos del conjunto de datos y exportemos el archivo limpio **como Titanic_clean.csv**.

```
> summary(Dataset)
```

```
    Sex           Age            SibSp            Parch
 female:312   Min.   : 0.42   Min.   :0.0000   Min.   :0.0000
 male  :577   1st Qu.:20.00   1st Qu.:0.0000   1st Qu.:0.0000
              Median :28.00   Median :0.0000   Median :0.0000
              Mean   :29.64   Mean   :0.5242   Mean   :0.3825
              3rd Qu.:38.00   3rd Qu.:1.0000   3rd Qu.:0.0000
              Max.   :80.00   Max.   :8.0000   Max.   :6.0000
              NA's   :177
     Fare          Embarked SurvivedF PclassF
 Min.   :  0.000   C:168    0:549     1:214
 1st Qu.:  7.896   Q: 77    1:340     2:184
 Median : 14.454   S:644              3:491
 Mean   : 32.097
 3rd Qu.: 31.000
 Max.   :512.329
```

```
> write.csv(Dataset, "Titanic_clean.csv")
```

4. Análisis de los datos

El conjunto de los datos que nos queda se puede dividir por según tres criterios: por si sobrevivieron, por la clase en la que viajaban y por el lugar donde embarcaron. Aunque en la práctiva solo nos centraremos en el primer grupo que se puede agrupar de la siguiente forma:

```
> sobreviven <- Dataset[Dataset$SurvivedF == "0",]
> mueren <- Dataset[Dataset$SurvivedF == "1",]
```

Vamos a comprobar la normalidad de los atributos numéricos por medio de la prueba de normalidad de Anderson-Darling. Los siguientes atributos según esta prueba y un nivel de significacion de $\alpha = 0{,}05$ no siguen una distribución normal:

```
> library(nortest)
> alpha = 0.05
> col.names = colnames(Dataset)
> for (i in 1:ncol(Dataset)) {
+   if (is.integer(Dataset[,i]) | is.numeric(Dataset[,i])) {
+       p_val = ad.test(Dataset[,i])$p.value
+       if (p_val < alpha) {
+           cat(col.names[i])
+           if (i < ncol(Dataset) - 1) cat(", ")
+       }
+   }
+ }
```

```
Age, SibSp, Parch, Fare,
```

Para comprobar la homogeneidad de las varianza se aplicara un test de Fligner-Killen. Para la relación entre Age y el Fare dando que no son homogéneas porque dan de resultado un p-valor inferior al 0.05.

```
> fligner.test(Fare ~ Age, data = Dataset)
```

```

    Fligner-Killeen test of homogeneity of variances

data:  Fare by Age
Fligner-Killeen:med chi-squared = 137.73, df = 87, p-value =
0.0004352
```

Ofreceremos ahora un estudio de diferentes posibles modelos de rregresión logística.

```
> modeloRLSuvirved1 <- glm(SurvivedF ~ Sex , family=binomial, data=Dataset)
> summary(modeloRLSuvirved1)
```

```

Call:
glm(formula = SurvivedF ~ Sex, family = binomial, data = Dataset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6423  -0.6471  -0.6471   0.7753   1.8256

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.0480     0.1291   8.116 4.83e-16 ***
Sex[T.male]  -2.5051     0.1673 -14.975  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.82  on 888  degrees of freedom
Residual deviance:  916.61  on 887  degrees of freedom
AIC: 920.61

Number of Fisher Scoring iterations: 4
```

```
> modeloRLSuvirved2 <- glm(SurvivedF ~ Sex + PclassF + SibSp + Parch, family=binomial, data=Dataset)
> summary(modeloRLSuvirved2)
```

```
Call:
glm(formula = SurvivedF ~ Sex + PclassF + SibSp + Parch, family = binomial,
    data = Dataset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2641  -0.6848  -0.4731   0.5955   2.5299

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.48295    0.23946  10.369  < 2e-16 ***
Sex[T.male]  -2.75521    0.19569 -14.079  < 2e-16 ***
PclassF[T.2] -0.84321    0.24674  -3.417 0.000632 ***
PclassF[T.3] -1.86146    0.21526  -8.648  < 2e-16 ***
SibSp        -0.23181    0.09925  -2.336 0.019512 *
Parch        -0.04885    0.11043  -0.442 0.658219
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.82  on 888  degrees of freedom
Residual deviance:  818.63  on 883  degrees of freedom
AIC: 830.63

Number of Fisher Scoring iterations: 4
```

```
> modeloRLSuvirved3 <- glm(SurvivedF ~ Sex + PclassF + SibSp + Parch + Fare, family=binomial, data=Dataset)
> summary(modeloRLSuvirved3)
```

```
Call:
glm(formula = SurvivedF ~ Sex + PclassF + SibSp + Parch + Fare,
    family = binomial, data = Dataset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2146  -0.6924  -0.4762   0.6004   2.5581

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.215530   0.291822   7.592 3.15e-14 ***
Sex[T.male]  -2.748118   0.196523 -13.984  < 2e-16 ***
PclassF[T.2] -0.634097   0.278635  -2.276   0.0229 *
PclassF[T.3] -1.615216   0.263836  -6.122 9.24e-10 ***
SibSp        -0.255772   0.100976  -2.533   0.0113 *
Parch        -0.091889   0.113409  -0.810   0.4178
Fare          0.003868   0.002468   1.567   0.1170
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.82  on 888  degrees of freedom
Residual deviance:  815.81  on 882  degrees of freedom
AIC: 829.81

Number of Fisher Scoring iterations: 5
```

```
> modeloRLSuvirved4 <- glm(SurvivedF ~ Sex + PclassF + SibSp + Parch + Age, family=binomial, data=Dataset)
> summary(modeloRLSuvirved4)
```

```
Call:
glm(formula = SurvivedF ~ Sex + PclassF + SibSp + Parch + Age,
    family = binomial, data = Dataset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7704  -0.6375  -0.3879   0.6354   2.4516

Coefficients:
             Estimate Std. Error z value     Pr(>|z|)
(Intercept)   4.346827   0.456049   9.531      < 2e-16 ***
Sex[T.male]  -2.633042   0.220028 -11.967      < 2e-16 ***
PclassF[T.2] -1.407361   0.284827  -4.941 0.0000007768 ***
PclassF[T.3] -2.642714   0.285800  -9.247      < 2e-16 ***
SibSp        -0.367175   0.126678  -2.898      0.00375 **
Parch        -0.036995   0.119580  -0.309      0.75704
Age          -0.045009   0.008241  -5.462 0.0000000472 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 960.90  on 711  degrees of freedom
Residual deviance: 635.94  on 705  degrees of freedom
  (177 observations deleted due to missingness)
AIC: 649.94

Number of Fisher Scoring iterations: 5
```

```
> modeloRLSuvirved5 <- glm(SurvivedF ~ Sex + PclassF + SibSp + Parch + Fare + Age, family=binomial,
+   data=Dataset)
> summary(modeloRLSuvirved5)
```

```
Call:
glm(formula = SurvivedF ~ Sex + PclassF + SibSp + Parch + Fare +
    Age, family = binomial, data = Dataset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7984  -0.6437  -0.3886   0.6353   2.4422

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   4.172041   0.503349   8.289  < 2e-16 ***
Sex[T.male]  -2.628356   0.220355 -11.928  < 2e-16 ***
PclassF[T.2] -1.285746   0.321735  -3.996 6.43e-05 ***
PclassF[T.3] -2.495085   0.338640  -7.368 1.73e-13 ***
SibSp        -0.375639   0.127328  -2.950  0.00318 **
Parch        -0.059443   0.122898  -0.484  0.62861
Fare          0.002022   0.002557   0.791  0.42904
Age          -0.044334   0.008276  -5.357 8.46e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 960.90  on 711  degrees of freedom
Residual deviance: 635.27  on 704  degrees of freedom
  (177 observations deleted due to missingness)
AIC: 651.27

Number of Fisher Scoring iterations: 5
```

El resultado que nos arroja un mejor resultado por medio de valorar de que un modelo es mejor cuanto menor es el valor AIC. En este caso es el **modeloRLSuvirved5**. Problemas que nos da este modelo, ignora los registros que contine el atributo Age como NA. Los valores significativos en este modelos para un valor inferior a 0.001 son Sex, PclassF y Age.
Veamos como predice alguno de los elementos del test:

```
> pred1 <- data.frame(SuvirvedF=0, Sex='male', PclassF="3", SibSp=0, Parch=0, Age=34.5)
> predict(modeloRLSuvirved4, pred1, type= "response")
```

```
       1
0.0771475
```

```
> pred2 <- data.frame(SuvirvedF='1', Sex='female', PclassF='3' , SibSp=1, Parch=0, Age=47)
> predict(modeloRLSuvirved4, pred2, type= "response")
```

```
       1
0.3146432
```

```
> pred3 <- data.frame(SuvirvedF='0', Sex='male', PclassF='2' , SibSp=0, Parch=0, Age=62)
> predict(modeloRLSuvirved4, pred3, type= "response")
```

```
        1
0.07697652
```

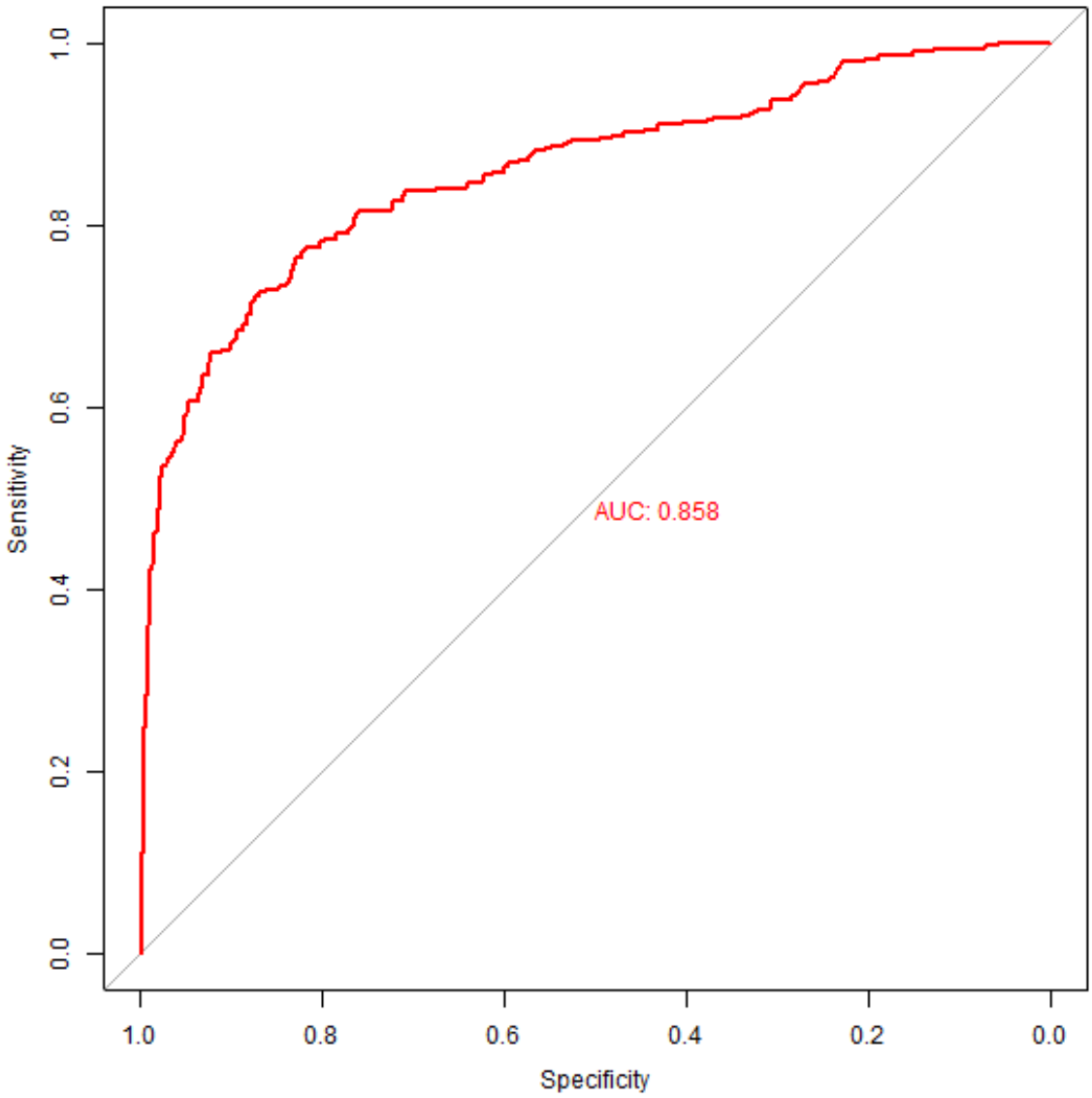
En este caso predice en los tres casos que fallece las personas en caso de tomar que solo sobrevive si el resultado es superior a 0.7.
Veamos la calidad del ajuste por medio de la matriz de confusión. Hay 129 falsos positivos que han muerto pero estan vivas, pero se consideran como vivos. Hay falsos negativos 15 que estaban vivas pero se han considerado muertas. Es bueno que los falsos negativos sea bastante mas reducido que los falsos positivos.

```
> prob_Survived <- predict(modeloRLSuvirved4, Dataset, type="response")
> pred_Survived <- ifelse(prob_Survived > 0.70,1,0)
> table (Dataset$SurvivedF, pred_Survived)
```

```
   pred_Survived
      0   1
  0 409  15
  1 129 159
```

Valoremos por otro lado la calidad del ajuste por medio de la curva ROC, como da un valor de AUC de 0.858 tiene un buen valor predictivo según esta curva.

```
> library(pROC)
> rocSuvirved <- with(Dataset,roc(Dataset$SurvivedF,prob_Survived))
> plot(rocSuvirved, col="red", print.auc=TRUE)
```



5. Conclusiones

Se puede definir un modelo a través de los datos del hundimiento para determinar las posibilidades de sobrevivir en una catastrofe de las dimensiones similares. Cuales son las variables de una personas son mas significativas a la hora de sobrevivir y hacer predicciones que pueden ayudar a la hora de fijar tarifas por según las carácterísticas de los individuos que suelen viajar en los transportes.