Práctica 2: Limpieza y validación de los datos

Beatriz Elena Jaramillo Gallego

08 de Mayo de 2018

Contents

Detalles de la actividad **2**
- Descripción . . . 2
- Competencias . . . 2
- Objetivos . . . 2

Resolución Práctica **3**
1. Descripción del dataset . . . 3
2. Integración y selección . . . 3
3. Limpieza de los datos . . . 6
   - 3.1. Ceros y elementos vacíos . . . 6
   - 3.2. Valores extremos . . . 8
4. Análisis de los datos. . . . 13
   - 4.1. Selección de los grupos de datos a analizar . . . 13
   - 4.2. Normalidad . . . 14
   - 4.3. Pruebas Estadísticas . . . 15
5. Visualización . . . 17
6. Resolución del problema. . . . 21
7. Código . . . 21

8. Referencias **22**

Detalles de la actividad

Descripción

En esta práctica se elabora un caso práctico orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

Competencias

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

- Capacidad de analizar un problema en el nivel de abstracción adecuado a cada situación y aplicar las habilidades y conocimientos adquiridos para abordarlo y resolverlo.
- Capacidad para aplicar las técnicas específicas de tratamiento de datos (integración, transformación, limpieza y validación) para su posterior análisis.

Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares.
- Saber identificar los datos relevantes y los tratamientos necesarios (integración, limpieza y validación) para llevar a cabo un proyecto analítico.
- Aprender a analizar los datos adecuadamente para abordar la información contenida en los datos.
- Identificar la mejor representación de los resultados para aportar conclusiones sobre el problema planteado en el proceso analítico.
- Actuar con los principios éticos y legales relacionados con la manipulación de datos en función del ámbito de aplicación.
- Desarrollar las habilidades de aprendizaje que les permitan continuarestudiando de un modo que tendrá que ser en gran medida autodirigido o autónomo.
- Desarrollar la capacidad de búsqueda, gestión y uso de información y recursos en el ámbito de la ciencia de datos.

Resolución Práctica

1. Descripción del dataset

¿Por qué es importante y qué pregunta/problema pretende responder?

Los datos para el análisis se ha obtenido a partir de este enlace en Kaggle Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic) y está constituido por 12 (variables) que presentan 891 pasajeros(filas o registros) en el archivo de train y 418 pasajeros(filas o registros) en el archivo de test.

Los datos se han dividido en dos grupos:

- Conjunto de entrenamiento (train.csv): El conjunto de entrenamiento se debe usar para construir sus modelos de aprendizaje automático
- Conjunto de prueba (test.csv): El conjunto de prueba se debe usar para ver qué tan bien se desempeña su modelo en datos no vistos.

Variables

- PassengerId: Un identificador numerico del pasajero. Es una variable númerica.
- Survived: Varibale binaria donde se indica si el pasajero sobrevivio o no. (0 = No, 1 = Yes)
- Pclass: La clase en la que viajaba el pasajero. Es una variable númerica. (1 = 1st, 2 = 2nd, 3 = 3rd)
- Name: El nombre del pasajero. Es una variable nominal.
- Sex: El sexo del pasajero. Es una varuable nominal.
- Age: La edad del pasajero. Es una variable númerica.
- SibSp: Numero de familiares cosanguineos de la persona abordo del Titanic. Es una variable númerica
- Parch: Numero de familaires de diferente grado que acompañaban a la persona abordo del Titanic. Es una variable númerica
- Ticket: El ticket correspondiente al pasajero al momento del abordaje. Es una variable nominal.
- Fare: La tarifa del ticket segun la clase en la que abordo el pasajero. Es una variable númerica
- Cabin: El identificador de la cabina que utilizo la persona durante el viaje. Es una variable nominal
- Embarked: Indica el lugar de embarque de la persona. Es una variable nominal. (C = Cherbourg, Q = Queenstown, S = Southampton)

Notas Variables Pclass: un proxy para el estado socio-económico (SES) 1er = superior 2do = Medio Tercero = Más bajo

Sibsp: el conjunto de datos define las relaciones familiares de esta manera ... Hermano = hermano, hermana, hermanastro, hermanastra Cónyuge = esposo, esposa (las amantes y los novios fueron ignorados)

Parch: El conjunto de datos define las relaciones familiares de esta manera ... Padre = madre, padre Niño = hija, hijo, hijastra, hijastro Algunos niños viajaban solo con una niñera, por lo tanto parch = 0 para ellos.

2. Integración y selección

Integración y selección de los datos de interés a analizar.

Inicialmente hare un al tratar de unir las 2 bases de datos (train y test), y para hacer un vistazo genereal de como estan los datos, no he podido porque el archivo train tiene 12 variables y el test 11 variables, la variable que diferencia es Survived.

He tomado la decisón de responder ¿Podria llegarse a determinar si los pasajeros que estan en el dataset test si sobrevivieron o no, basandome en los datos de entrenamiento?

```
#setwd("C:/Users/Admin/Dropbox/Master/Tipologia de Datos/PRACTICA 2")
train <- read.csv2("train.csv",header = TRUE,sep = ";",dec = ".",stringsAsFactors=FALSE)
cat(paste0("Carga fichero train.csv OK.","\n\n"))
```

```
## Carga fichero train.csv OK.
```

```
str(train)
```

```
## '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       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ 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     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : chr  "7.25" "712.833" "7.925" "53.1" ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

```
test <- read.csv2("test.csv",header = TRUE,sep = ";",dec = ".")
cat(paste0("Carga fichero test.csv OK.","\n\n"))
```

```
## Carga fichero test.csv OK.
```

```
str(test)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409 273 414 182 370 85 5
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 262 159 85 101 270 ...
##  $ Fare       : Factor w/ 170 levels "","0","1.356.333",..: 148 122 170 161 15 164 141 76 135 63 ...
##  $ Cabin      : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

Variables Categoricales: Survived, Sex, and Embarked. Ordinal: Pclass.

Variables Continuas: Age, Fare. Discrete: SibSp, Parch

He unido los dos dataset (train y test) para pode trabajar con ellos de manera mas sencilla, pero como el dataset test no tiene la variable Survived, he creado la columna “Survived” y la he llenado con NA

```
test$Survived <- NA
titanic <- rbind(train,test)
summary(titanic)
```

```
##   PassengerId      Survived          Pclass          Name
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:1309
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000   Class :character
##  Median : 655   Median :0.0000   Median :3.000   Mode  :character
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##      Sex                 Age            SibSp            Parch
##  Length:1309        Min.   : 0.17   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.000
##  Mode  :character   Median :28.00   Median :0.0000   Median :0.000
##                     Mean   :29.88   Mean   :0.4989   Mean   :0.385
##                     3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.000
##                     Max.   :80.00   Max.   :8.0000   Max.   :9.000
##                     NA's   :263
##     Ticket              Fare              Cabin
##  Length:1309        Length:1309        Length:1309
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    Embarked
##  Length:1309
##  Class :character
##  Mode  :character
##
##
##
##
```

```
titanic$Name <- as.factor(titanic$Name)
titanic$Sex <- as.factor(titanic$Sex)
titanic$Ticket <- as.factor(titanic$Ticket)
titanic$Embarked <- as.factor(titanic$Embarked)

summary(titanic)
```

```
##   PassengerId      Survived          Pclass
##  Min.   :   1   Min.   :0.0000   Min.   :1.000
##  1st Qu.: 328   1st Qu.:0.0000   1st Qu.:2.000
##  Median : 655   Median :0.0000   Median :3.000
##  Mean   : 655   Mean   :0.3838   Mean   :2.295
##  3rd Qu.: 982   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309   Max.   :1.0000   Max.   :3.000
##                 NA's   :418
##                                 Name          Sex           Age
##  Connolly, Miss. Kate            :   2   female:466   Min.   : 0.17
```

```
##  Kelly, Mr. James                :   2   male  :843   1st Qu.:21.00
##  Abbing, Mr. Anthony             :   1                Median :28.00
##  Abbott, Master. Eugene Joseph   :   1                Mean   :29.88
##  Abbott, Mr. Rossmore Edward     :   1                3rd Qu.:39.00
##  Abbott, Mrs. Stanton (Rosa Hunt):   1                Max.   :80.00
##  (Other)                         :1301                NA's   :263
##      SibSp            Parch           Ticket         Fare
##  Min.   :0.0000   Min.   :0.000   CA. 2343:  11   Length:1309
##  1st Qu.:0.0000   1st Qu.:0.000   1601    :   8   Class :character
##  Median :0.0000   Median :0.000   CA 2144 :   8   Mode  :character
##  Mean   :0.4989   Mean   :0.385   3101295 :   7
##  3rd Qu.:1.0000   3rd Qu.:0.000   347077  :   7
##  Max.   :8.0000   Max.   :9.000   347082  :   7
##                                   (Other) :1261
##     Cabin           Embarked
##  Length:1309          :  2
##  Class :character   C:270
##  Mode  :character   Q:123
##                     S:914
##
##
##
```

3. Limpieza de los datos

3.1. Ceros y elementos vacíos

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

Hay 263 datos **NA** en la variable Age, hay 418 NA en la variable **Survived**, que son los valores que anteriormente adicione del dataset test. La variable **Fare** debe ser numerica y se encuentra como factor, la he convertido en numerica.

```
titanic$Fare <- as.numeric(titanic$Fare)
```

```
## Warning: NAs introducidos por coerción
```

```
cat(paste0("Valores vacios Age ",round((263/1309)*100,2)),"%")
```

```
## Valores vacios Age 20.09 %
```

```
cat("\n\n")
```

```
cat(paste0("Valores vacios Fare ",round((36/1309)*100,2)),"%")
```

```
## Valores vacios Fare 2.75 %
```

```
cat("\n\n")
```

```
# Para aquellos pasajeros que se desconoce la cabina, se le cambia su vacio  por un string que indica d
titanic$Cabin[titanic$Cabin==""] <- "Unknown"
titanic$Cabin <- as.factor(titanic$Cabin)

# Hay 2 pasajeros en la columna Embarked que estan vacios y la mayoria embarco por S entonces lo he agr
```

```
errores = which(titanic$Embarked=="")
titanic$Embarked[errores] = "S"
titanic$Embarked <- factor(titanic$Embarked)

# 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          36           0           0
```

```
rm(errores)
```

Para manejar los registros que contienen valores desconocidos para algún campo, una opción podría ser eliminar los registros que incluyen este tipo de valores de la variable Age, pero ello supondría desaprovechar el 20.09% de esta información y para la varaible Fare quitar los NA upondría desaprovechar el 2.75% de esta información

Se empleará el método para imputa de una manera sofisticada para que no toda la matriz de distancia tenga que calcularse: la imputación basada en k vecinos más próximos (en inglés, kNN-imputation). Por lo tanto, la implementación del paquete VIM también es aplicable para conjuntos de datos razonablemente grandes

Como tengo 12 datos en donde la variable Age son números inferiores a 1, por lo tanto he decido multiplicar estos datos por 100, ya que considero que ha sido un error de digitación y no quisiera perderlos, he hecho esto antes de la imputación.

```
#Cambio las observaciones <1 y las multiplico por 100 y luego las vuelvo a unir a titanic
edad_aux <- filter(titanic, titanic$Age<1)
```

```
## Warning: package 'bindrcpp' was built under R version 3.5.0
```

```
edad_aux$Age <- edad_aux$Age*100
titanic <- dplyr::full_join(titanic,edad_aux)
```

```
## Joining, by = c("PassengerId", "Survived", "Pclass", "Name", "Sex", "Age", "SibSp", "Parch", "Ticket
```

```
titanic <-arrange(titanic, PassengerId)
rm(edad_aux)
# Imputación de valores mediante la función kNN() del paquete VIM
suppressWarnings(suppressMessages(library(VIM)))

titanic$Age <- kNN(titanic)$Age
titanic$Fare <- kNN(titanic)$Fare

sapply(titanic, function(x) sum(is.na(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         423           0           0           0           0
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

```
summary(titanic)
```

```
##   PassengerId        Survived          Pclass
##  Min.   :   1.0   Min.   :0.0000   Min.   :1.000
```

```
##  1st Qu.: 329.0   1st Qu.:0.0000   1st Qu.:2.000
##  Median : 657.0   Median :0.0000   Median :3.000
##  Mean   : 656.4   Mean   :0.3886   Mean   :2.297
##  3rd Qu.: 984.0   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :1.0000   Max.   :3.000
##                   NA's   :423
##                               Name          Sex           Age
##  Aks, Master. Philip Frank     :   2   female:470   Min.   : 0.17
##  Allison, Master. Hudson Trevor:   2   male  :851   1st Qu.:21.00
##  Baclini, Miss. Eugenie        :   2                Median :28.00
##  Baclini, Miss. Helene Barbara :   2                Mean   :29.79
##  Caldwell, Master. Alden Gates :   2                3rd Qu.:37.00
##  Connolly, Miss. Kate          :   2                Max.   :92.00
##  (Other)                       :1309
##      SibSp            Parch            Ticket          Fare
##  Min.   :0.0000   Min.   :0.0000   CA. 2343:  11   Min.   :  0.00
##  1st Qu.:0.0000   1st Qu.:0.0000   1601    :   8   1st Qu.: 11.50
##  Median :0.0000   Median :0.0000   CA 2144 :   8   Median : 29.70
##  Mean   :0.5019   Mean   :0.3944   113781  :   7   Mean   : 95.27
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3101295 :   7   3rd Qu.: 81.12
##  Max.   :8.0000   Max.   :9.0000   347077  :   7   Max.   :910.79
##                                    (Other) :1273
##              Cabin      Embarked
##  Unknown        :1025   C:273
##  C23 C25 C27    :   6   Q:123
##  B57 B59 B63 B66:   5   S:925
##  C22 C26        :   5
##  G6             :   5
##  B96 B98        :   4
##  (Other)        : 271
```

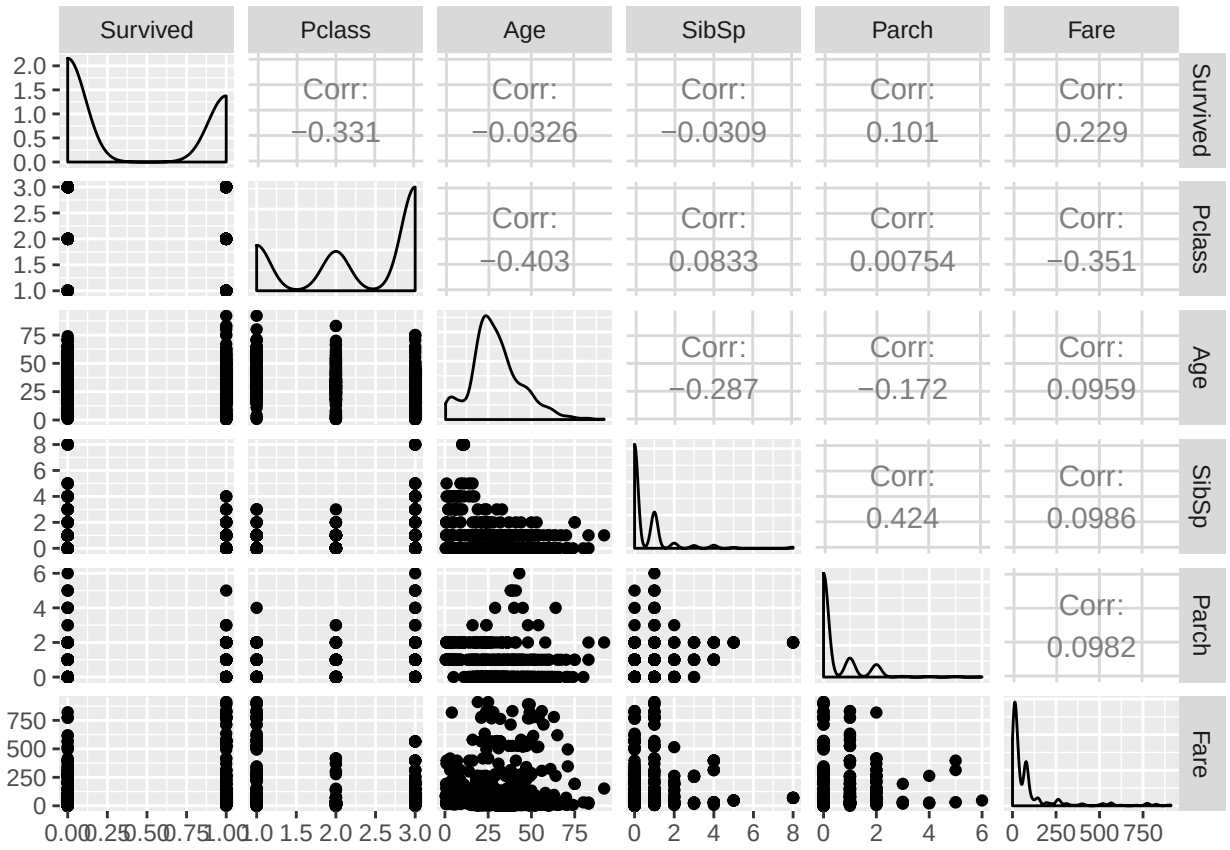
```
#   Listo el preprocesado del dataset, separo en train y test
train <- titanic[1:891,]
test <- titanic[892:1309,]
#Quitamos la columna Survived de test usada para unir los datos
test$Survived = NULL
```

3.2. Valores extremos

Identificación y tratamiento de valores extremos

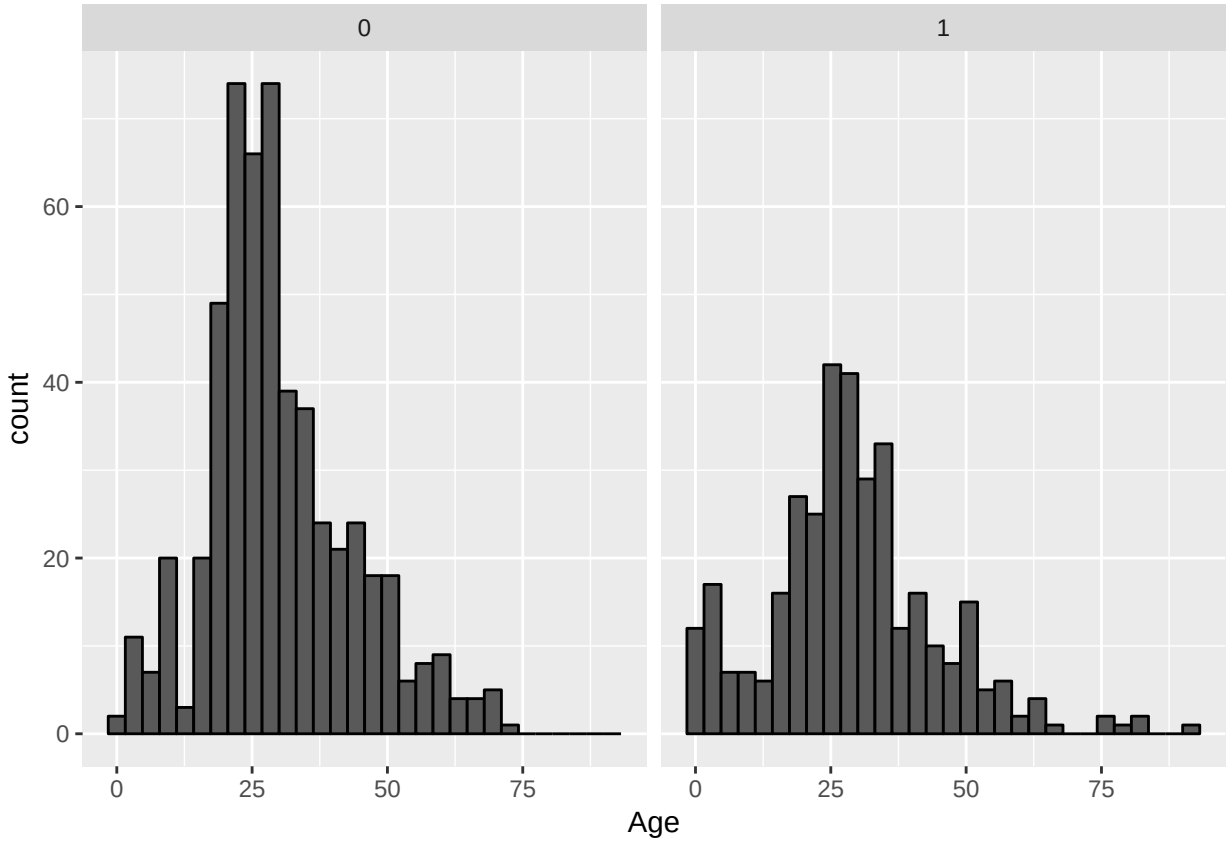
Los valores extremos o outliers son aquellos que parecen no ser congruentes sin los comparamos con el resto de los datos.

```
ggpairs(train[c(2,3,6,7,8,10)])
```




```
#gráfico la variable Age, sin NA y valores <1 corregidos
ggplot(train, aes(x = Age)) +
    geom_histogram(fill = "darkblue", alpha = .5) +
    geom_histogram(colour = "black")+
    facet_wrap(~ Survived)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

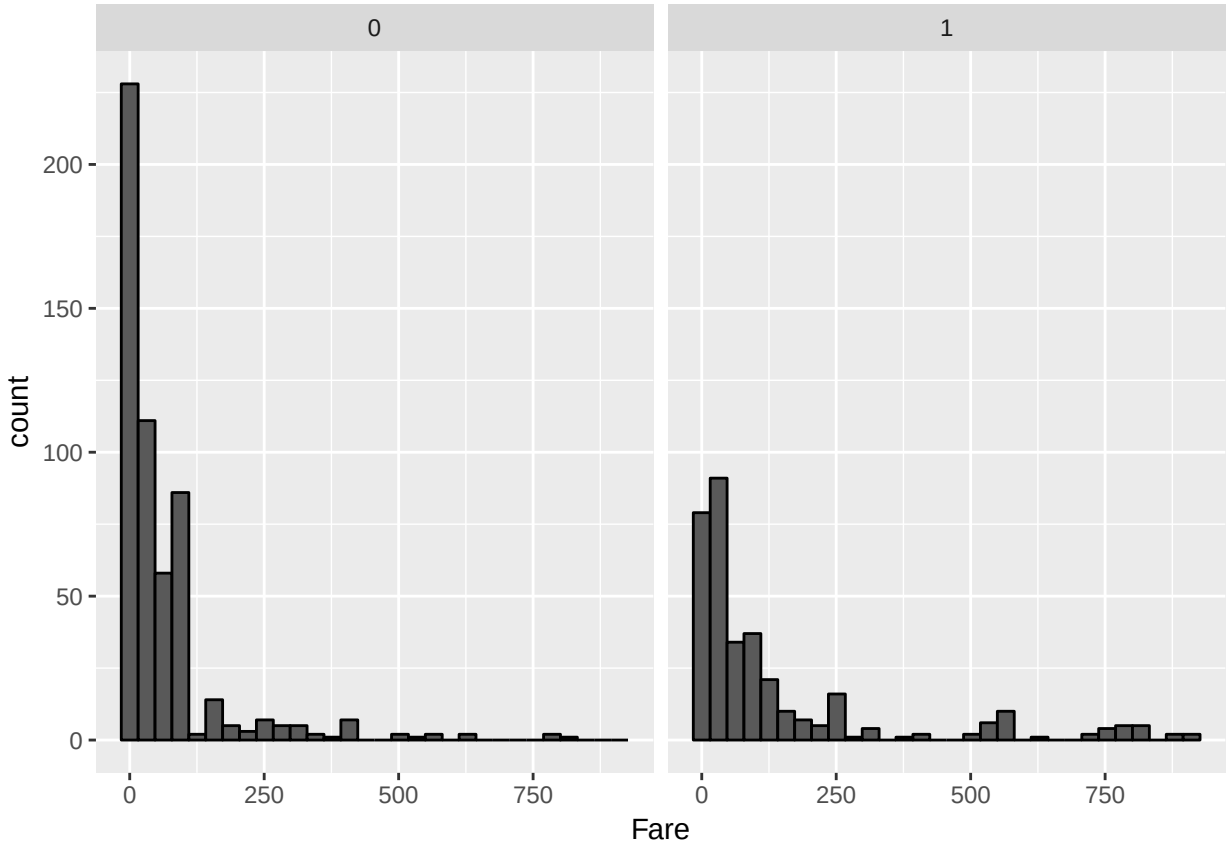



```
boxplot.stats(train$Age)$out
```

```
##  [1] 66.0 65.0 83.0 71.0 70.5 62.0 63.0 65.0 92.0 64.0 65.0 75.0 63.0 71.0
## [15] 64.0 62.0 62.0 80.0 75.0 70.0 70.0 67.0 62.0 83.0 74.0
```

```
#gráfico la variable Fare
ggplot(train, aes(x = Fare)) +
    geom_histogram(fill = "darkblue", alpha = .5) +
    geom_histogram(colour = "black")+
    facet_wrap(~ Survived)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

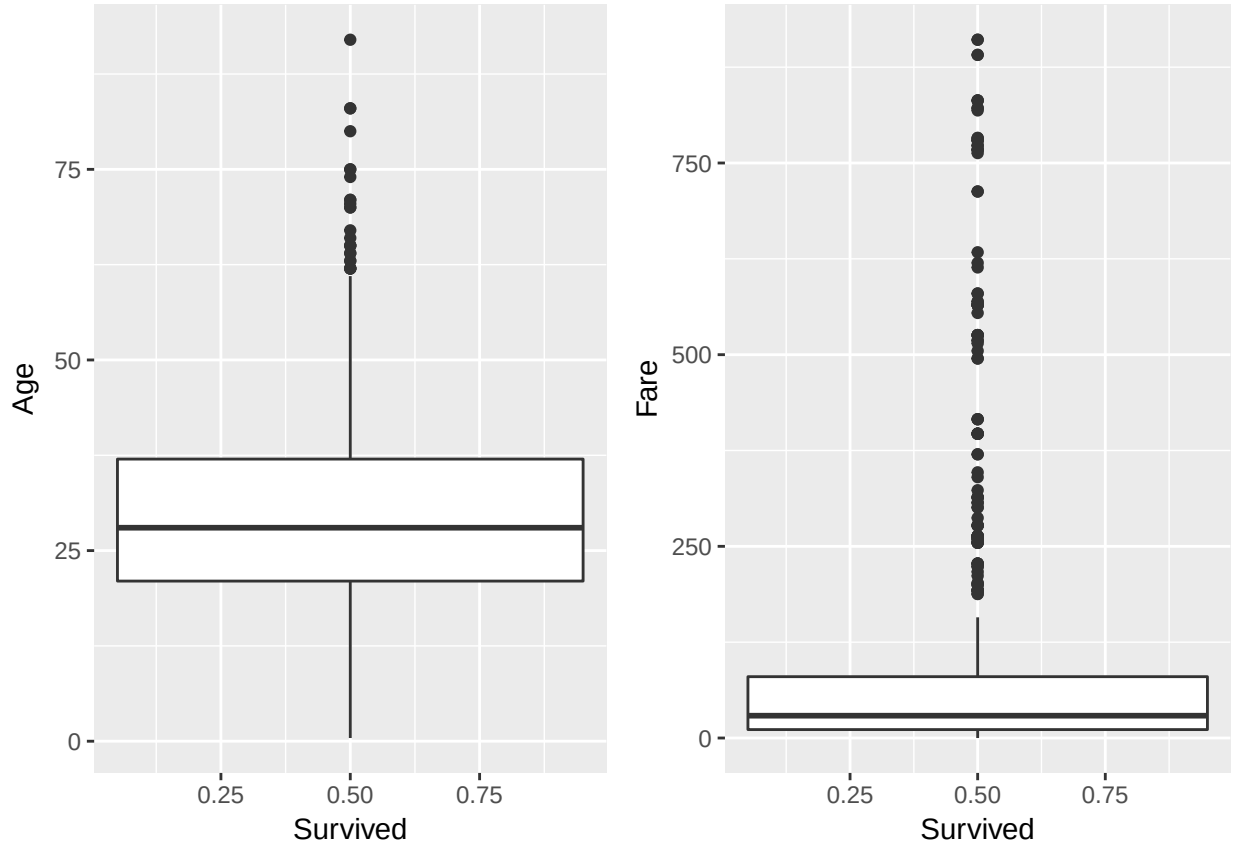



```
boxplot.stats(train$Fare)$out
```

```
##   [1] 712.833 518.625 300.708 313.875 263.000 277.208 712.833 821.708
##   [9] 415.792 216.792 396.875 767.292 619.792 277.208 564.958 263.000
##  [17] 346.542 633.583 772.875 300.708 772.875 223.583 262.833 613.792
##  [25] 396.875 564.958 306.958 254.667 287.125 313.875 277.208 187.875
##  [33] 762.917 254.667 313.875 525.542 202.125 313.875 396.875 779.583
##  [41] 910.792 277.208 262.875 569.292 831.583 262.375 262.875 579.792
##  [49] 263.000 277.208 554.417 821.708 211.500 227.525 254.667 202.125
##  [57] 263.000 818.583 192.583 199.667 891.042 518.625 192.583 192.583
##  [65] 910.792 254.667 199.667 495.042 782.667 564.958 262.875 340.208
##  [73] 579.792 223.583 227.525 514.792 263.875 525.542 782.667 569.292
##  [81] 415.792 525.542 323.208 779.583 396.875 564.958 192.583 192.583
##  [89] 767.292 255.875 262.875 767.292 415.792 396.875 564.958 227.525
##  [97] 262.875 262.875 495.042 227.525 262.875 187.875 262.375 779.583
## [105] 262.875 306.958 259.292 370.042 396.875 564.958 370.042 831.583
## [113] 564.958 891.042 525.542 192.583 259.292 504.958 525.542 831.583
```

```
p <- ggplot(train, aes(x=Survived, y=Age,fill=Survived)) +
    geom_boxplot()
q <- ggplot(train, aes(x=Survived, y=Fare,fill=Survived)) +
    geom_boxplot()

grid.arrange(p,q,ncol = 2)
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

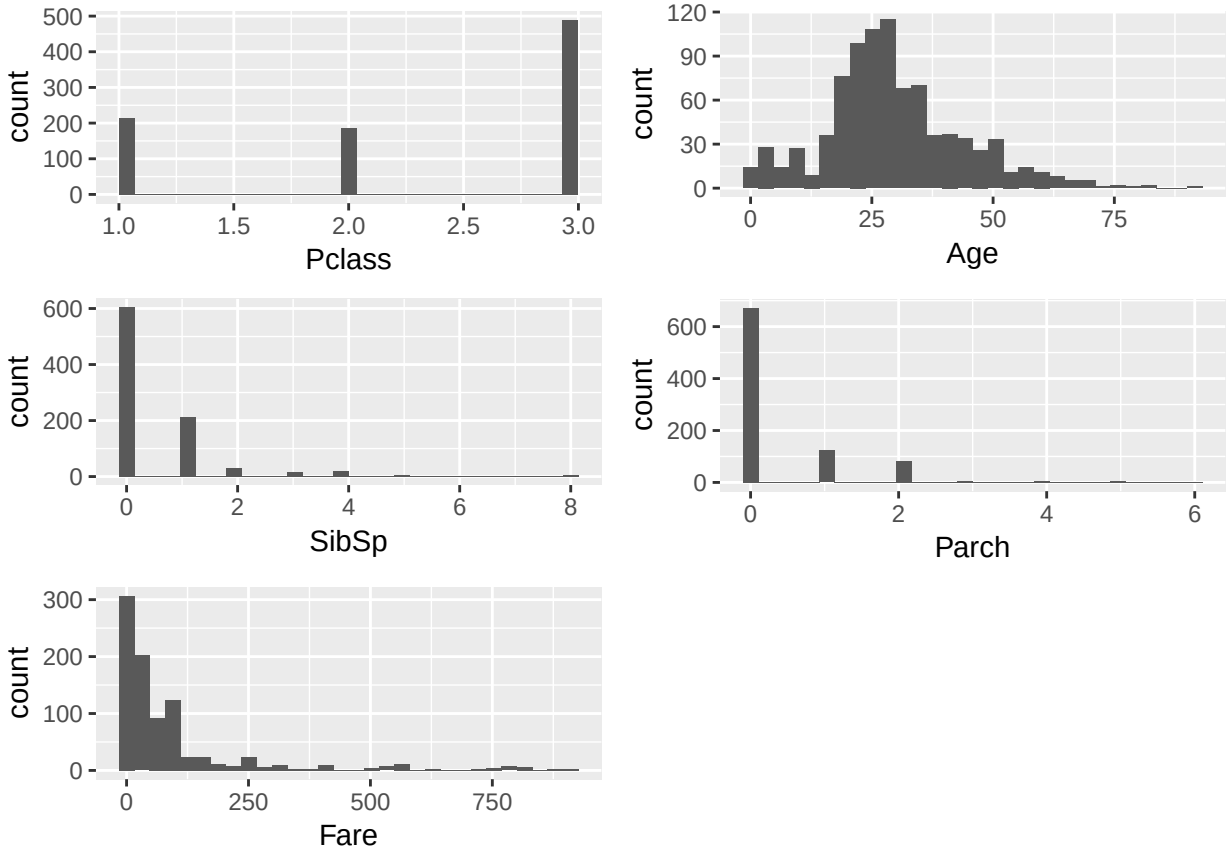


Otras variables númericas que se utilizan en el problema.

```
# La cantidad de personas en cada clase
p <- ggplot(train, aes(x=Pclass)) +
  geom_histogram()

# La edad de las personas
q <- ggplot(train, aes(x=Age)) +
  geom_histogram()

# La cantidad hermanos/conyuges de la personas
r <- ggplot(train, aes(x=SibSp)) +
  geom_histogram()

# La cantidad de parientes y niños abordo
s <- ggplot(train, aes(x=Parch)) +
  geom_histogram()

# La tarifas de los tickets por personas
t <- ggplot(train, aes(x=Fare)) +
  geom_histogram()
```

```
grid.arrange(p,q,r,s,t,ncol = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



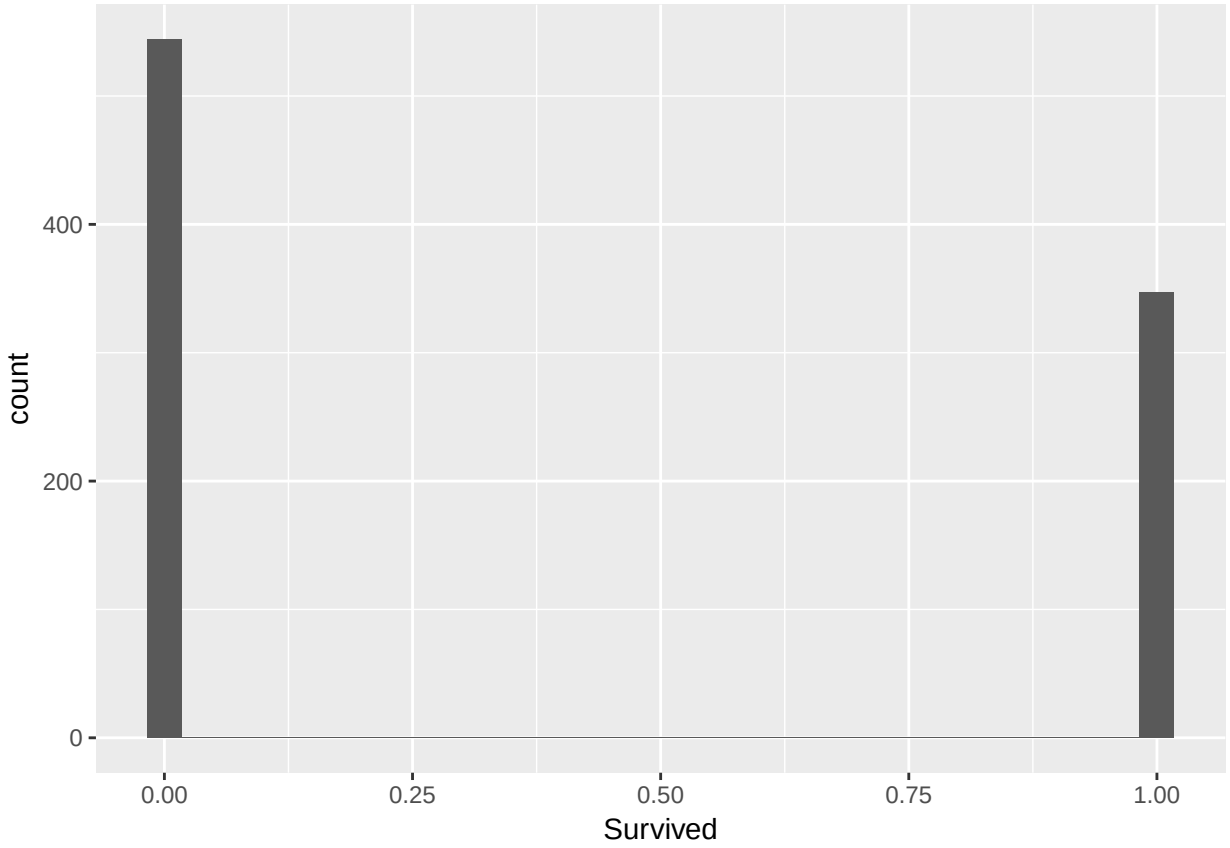
4. Análisis de los datos.

4.1. Selección de los grupos de datos a analizar

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

```
#Analisis de supervivencia
#   Con el caso de los sobrevientes o no
ggplot(train, aes(x=Survived)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
table(as.factor(train$Survived)) #  Cantidades en una tabla
```

```
##
##   0   1
## 544 347
```

Podemos observar que alrededor del 55.45% no sobrevivio y el 44.55% sobrevivio.

4.2. Normalidad

Comprobación de la normalidad y homogeneidad de la varianza.

Para la comprobación de que los valores que toman nuestras variables cuantitativas provienen de una población con distribución normal, utilizaremos la prueba de normalidad de Anderson-Darling.

Con esto, comprobaremos que para que cada prueba se obtiene un p-valor superior al nivel de significación prefijado $\alpha = 0,05$. Si esto se cumple, entonces se considera que la variable se distribuye normalmente.

```
library(nortest)
alpha = 0.05
col.names = colnames(train)
for (i in 1:ncol(train)) {
    if (i == 1) cat("Variables que no se distribuyen normalmente:\n")
    if (is.integer(train[,i]) | is.numeric(train[,i])) {
        p_val = ad.test(train[,i])$p.value
        if (p_val < alpha) {
            cat(col.names[i])
```

```
            # Format output
    if (i < ncol(train) - 1) cat(", ")
        }
    }
}
```

```
## Variables que no se distribuyen normalmente:
## PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare,
```

```
fligner.test(Age ~ Survived, data = train)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by Survived
## Fligner-Killeen:med chi-squared = 4.7838, df = 1, p-value =
## 0.02873
```

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

4.3. Pruebas Estadísticas

Aplicación de pruebas estadísticas para comparar los grupos de datos

En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

Ahora poseemos un mejor conocimiento de los elementos y atributos de las variables, pero esta no nos dicen la manera en que se relacionan. Para ello apliacaremos un analisis de componentes principales al conjunto de datos.

```
#   Observemos el Analisis de Componentes Principales (ACP) de las variables cuantitativas
cuant <- data.frame(
    train$Survived,train$Pclass,train$Age,train$Parch,train$SibSp,train$Fare)
train.pca <- pca(cuant)
train.pca
```

```
## Principal Components Analysis
## Call: principal(r = r, nfactors = nfactors, residuals = residuals,
##     rotate = rotate, n.obs = n.obs, covar = covar, scores = scores,
##     missing = missing, impute = impute, oblique.scores = oblique.scores,
##     method = method)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                   PC1    h2   u2 com
## train.Survived  0.43 0.185 0.82   1
## train.Pclass   -0.80 0.633 0.37   1
## train.Age       0.68 0.459 0.54   1
## train.Parch    -0.28 0.081 0.92   1
## train.SibSp    -0.45 0.198 0.80   1
## train.Fare      0.47 0.225 0.77   1
##
##                 PC1
```

```
## SS loadings    1.78
## Proportion Var 0.30
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.2
##  with the empirical chi square  1090.26  with prob <  5.8e-229
##
## Fit based upon off diagonal values = 0.22
```

Obtenido el PCA, observemos claramente la correlación de las variables:

```
#   Viendo las correlaciones de las variables involucradas en el PCA, procedemos
#   a ver aquellas variables que parecieran tener alguna relacion entre si
M <- cor(cuant)
round(M,2)
```

```
##                train.Survived train.Pclass train.Age train.Parch
## train.Survived           1.00        -0.33     -0.03        0.10
## train.Pclass            -0.33         1.00     -0.40        0.01
## train.Age               -0.03        -0.40      1.00       -0.17
## train.Parch              0.10         0.01     -0.17        1.00
## train.SibSp             -0.03         0.08     -0.29        0.42
## train.Fare               0.23        -0.35      0.10        0.10
##                train.SibSp train.Fare
## train.Survived       -0.03       0.23
## train.Pclass          0.08      -0.35
## train.Age            -0.29       0.10
## train.Parch           0.42       0.10
## train.SibSp           1.00       0.10
## train.Fare            0.10       1.00
```

Podemos observar que entre las variables cuantitativas, hay una serie de resultados interesantes:

- Existe una correlacion negativa significativa entre la clase en la que abordo el pasajero y su tarifa y su supervivencia. Esto se explica dado que las personas de primera clase pagaban una tarifa mucho mayor por abordar y tambien podemos decir que la mayoria de las personas al ser menor la cantidad de personas en primera clase con respecto a las otras dos, estos tenian una mayor posibilidad de sobrevivir, mientras que la segunda y la tercera clase ocurre lo contrario.

```
#   Con respecto al Sexo de las personas podemos ver la siguiente distribucion
table(train$Sex)
```

```
##
## female   male
##    313    578
```

Vemos que habian 313 mujeres y 578 hombres lo que representa un 35,13% de mujeres y un 64,87% de hombres en el conjunto de entrenamiento. Ahora relacionemos esta información con la sobrevivencia de los pasajeros y tambien la tasa de sobrevivencia de los niños.

```
#   Siguiendo el contexto del problema examinemos el sexo de los sobrevivientes
table(train$Sex,train$Survived)
```

```
##
##            0   1
##   female  79 234
##   male   465 113
```

```
Surv_age <- ifelse(train$Age<=18,"Niño","Adulto")
table(Surv_age,train$Survived)
```

```
##
## Surv_age   0   1
##   Adulto 464 270
##   Niño    80  77
```

```
table(Surv_age,train$Survived, by=train$Sex) # Sobrevivieron mas niñas que niños
```

```
## , , by = female
##
##
## Surv_age   0   1
##   Adulto  51 183
##   Niño    28  51
##
## , , by = male
##
##
## Surv_age   0   1
##   Adulto 413  87
##   Niño    52  26
```

Ahora comparemos usando arboles decision como metodo para generar un modelo predictivo.

```
#   Tomamos un subconjunto de los mismos
training <- subset(train,select=-c(PassengerId,Name,Ticket,Cabin))
modelo_arbol <- rpart(Survived ~ ., data=training,method="class")

prediction <- predict(modelo_arbol,training[,-1])
prediction <- ifelse(prediction<0.5,0,1)
matriz_conf <- table(prediction = prediction[,2], true = training[,1])
matriz_conf
```

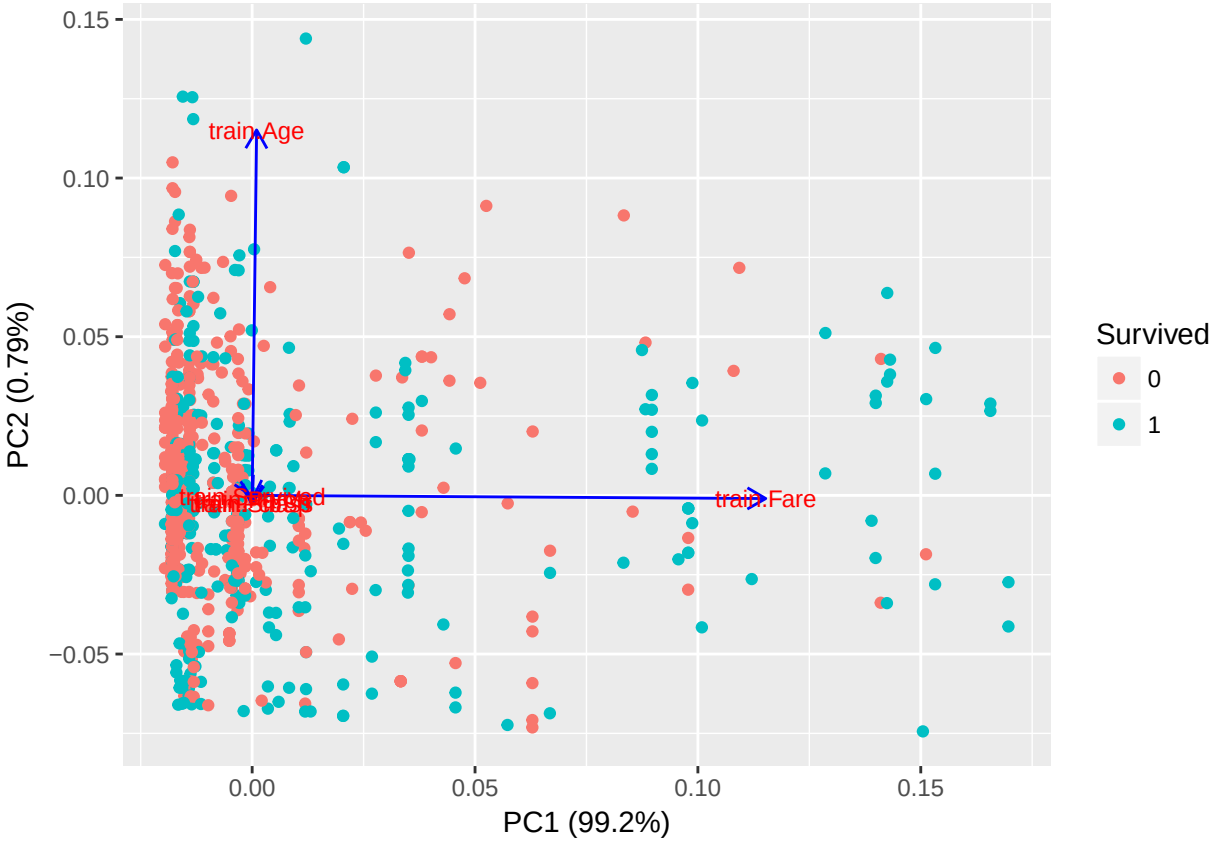
```
##           true
## prediction   0   1
##          0 504 107
##          1  40 240
```

De acuerdo a la matriz de confusión tenemos el mismo indice de acierto y de desacierto que teniamos inicialmente, con una pequeña variacion entre falsos positivos y falsos negativos.

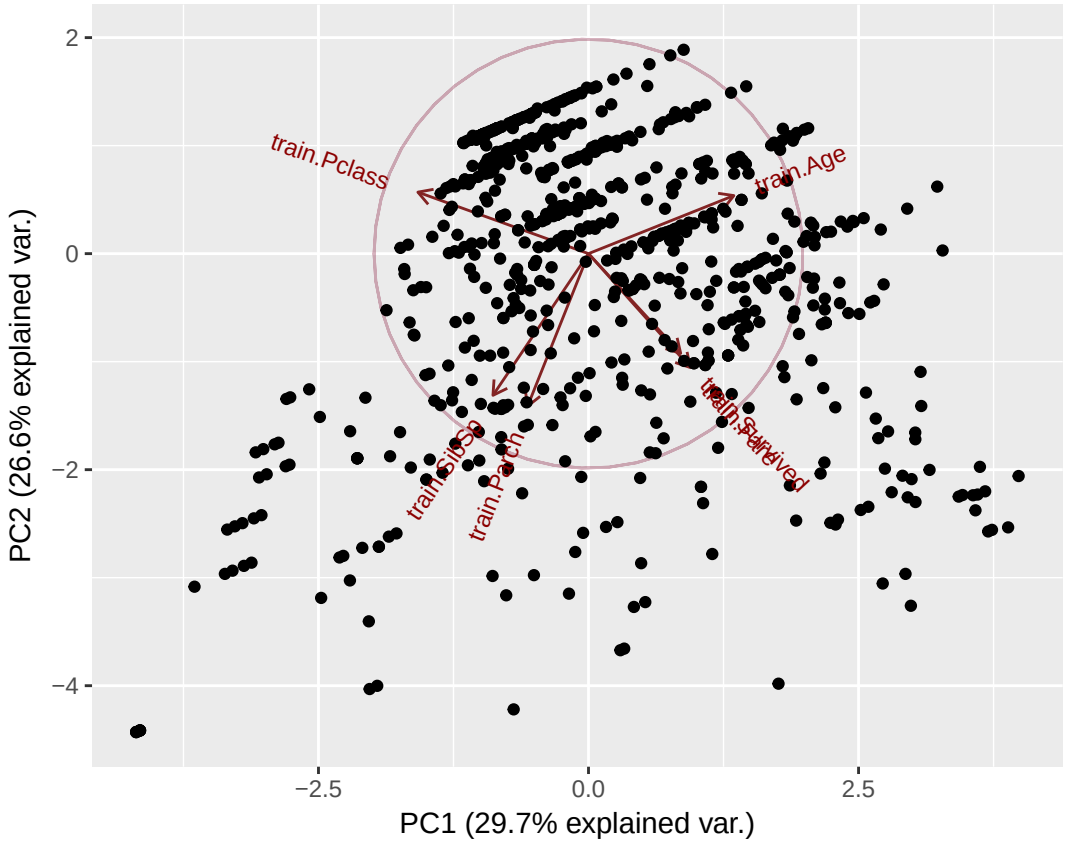
5. Visualización

Representación de los resultados a partir de tablas y gráficas

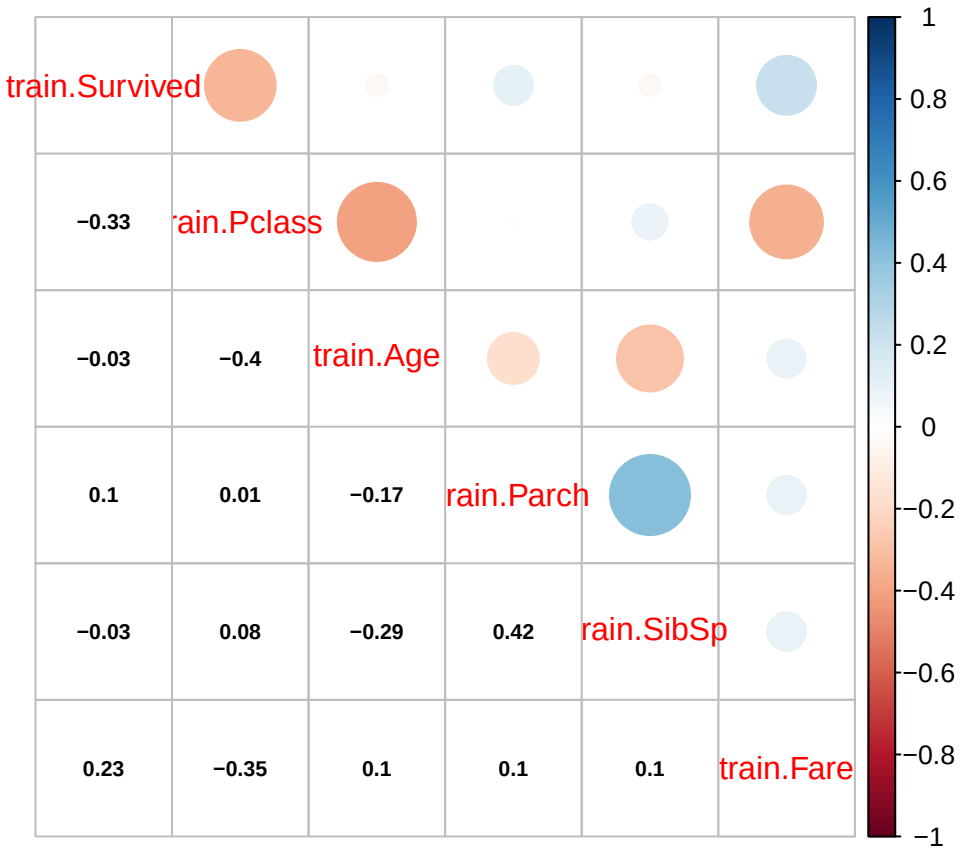
```r
#acp
# autoplot(prcomp(cuant))
train$Survived <- as.factor(train$Survived)
autoplot(prcomp(cuant), data = train, colour = 'Survived',
         loadings = TRUE, loadings.colour = 'blue',
         loadings.label = TRUE, loadings.label.size = 3)
```



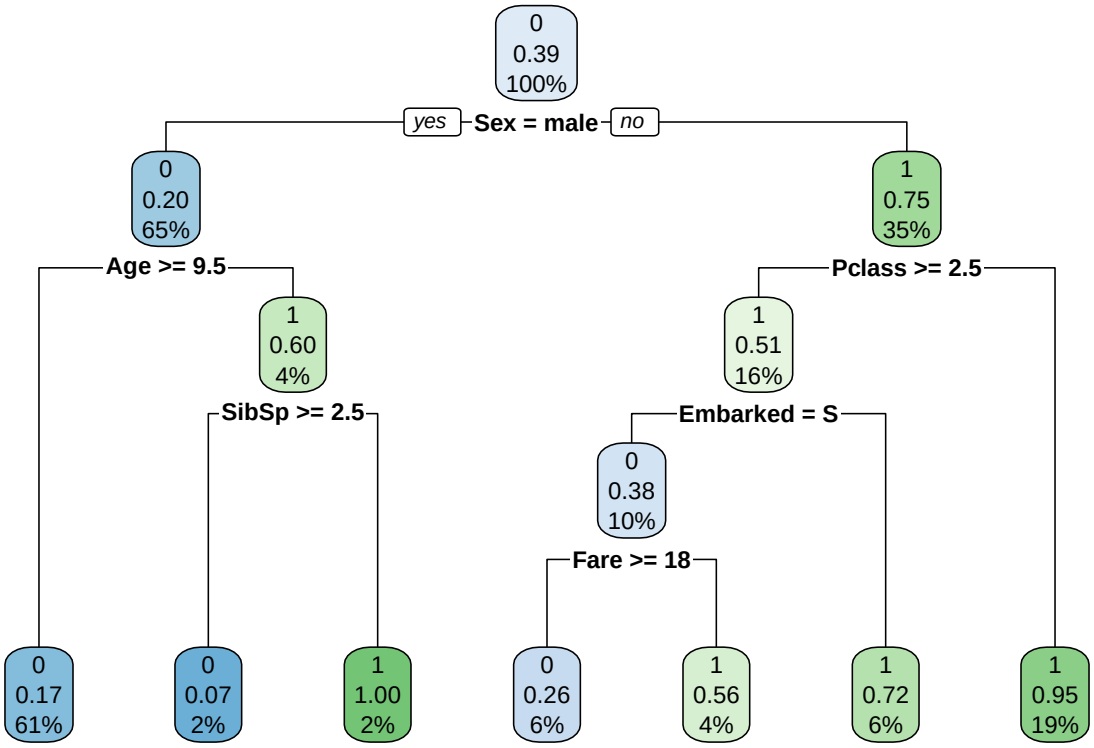

```r
train.pca2 <- prcomp(cuant,scale.=TRUE)
g<-ggbiplot(train.pca2, obs.scale=1, var.scale=1,  ellipse=TRUE, circle=TRUE)
g<-g+scale_color_discrete(name="")
print(g)
```




```
#Correlación
corrplot.mixed(M, lower.col = "black", number.cex = .7)
```




```
#arbol de decisión
rpart.plot(modelo_arbol)
```



Las variables Survived y Fare estan altamente correlacionadas.

6. Resolución del problema.

A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

7. Código

Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

8. Referencias

Los siguientes recursos son de utilidad para la realización de la práctica:

- 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"
- Ejemplos de contrastes de hipótesis con R: " https://rstudio-pubs-static.s3.amazonaws.com/65042_a1784120e81a430f9de400ed9b899b0b.html"
- Tutorial dplyr: "https://github.com/fdelaunay/tutorial-dplyr-es/blob/master/R/tutorial-dplyr.md"
- Test de Shapiro-Wilk: "https://rpro.wikispaces.com/Test+de+Shapiro-Wilk"
- Estadística descriptiva: "Introducción al análisis de datos", Àngel J. Gil Estallo
- Intervalos de confianza, Àngel J. Gil Estallo
- Contrastes de hipótesis, Carles Rovira Escofet
- Contraste de dos muestras, Josep Gibergans Bàguena