Tipologia PRA2

2022-05-31

Descripción del dataset:

Nuestros datos se dividen en 3 grupos:

- train.csv
- test.csv
- gender_submission.csv

El set de datos "train", nuestra base de datos de entrenamiento, se utilizará para construir nuestro modelo de predicción. Este set de datos contiene información relativa a 891 pasajeros que estuvieron en el Titanic, así como información sobre si sobrevivieron o no.

En este trabajo, buscaremos analizar las diferentes condiciones y relaciones que pudieron determinar que un pasajero sobreviviera o no a ese suceso. Por ejemplo, analizaremos si ciertos factores como el sexo de la persona, la edad o incluso su estatus social (entre otras) fueron decisivas para su supervivencia. A partir de este análisis, se planteará un modelo capaz de predecir la supervivencia de los pasajeros en función de estas características.

Para probar la consistencia del modelo, se utilizará el set de datos "test". Es por esto que este set de datos no contiene información sobre si el pasajero ha sobrevivido o no, ya que deberemos predecirlo a partir de nuestro modelo.

Finalmente, a partir del archivo gender_submission (que contiene los verdaderos datos de supervivencia del set de datos test), evaluaremos la precisión de nuestro modelo en cuanto a las predicciones obtenidas a partir del set de datos test.

Exploración del dataset:

Carga de la base de datos:

Comenzaremos por cargar la base de datos de entrenamiento:

```
# Cargamos el dataset, y observamos sus primeros registros:
titanic_train <- read.csv("./train.csv", header = TRUE, sep = ",")

# Inspeccionamos sus primeros registros:
head(titanic_train)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
```

```
## 6           6        0      3
##                                                  Name    Sex Age SibSp Parch
## 1                             Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                              Heikkinen, Miss. Laina female  26     0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                            Allen, Mr. William Henry   male  35     0     0
## 6                                    Moran, Mr. James   male  NA     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
## 6           330877  8.4583              Q
```

Inspección:

Ahora, procederemos con la etapa de exploración de los datos, revisando en primer lugar su estructura:

```
# Estructura:
str(titanic_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       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

En el output anterior, observamos las **dimensiones**, nuestro dataset cuenta con 12 **variables** y 891 **registros**, además podemos ver el **tipo** de variables que son y algunos de sus valores.

Aprovechando esto, describimos las variables de nuestro dataset:

- **PassengerId**: identificador único de cada pasajero.
- **Survived**: indica si ha sobrevivido o no.
  - 0: No sobrevive.
  - 1: Sobrevive.
- **Pclass**: ticket indicando la clase en que viajaba.
  - 1: primera clase.
  - 2: segunda clase.
  - 3: tercera clase.

- **Name**: nombre del pasajero/pasajera.
- **Sex**: género del pasajero/pasajera.
- **Age**: edad del pasajero/pasajera.
- **SibSp**: nº de hermanos/hermanas, cónyuges a bordo del Titanic.
- **Parch**: nº de padres/madres, hijos/hijas a bordo del Titanic.
- **Ticket**: nº de ticket.
- **Fare**: tarifa del pasajero.
- **Cabin**: nº de cabina.
- **Embarked**: puerto donde embarcaron.
  - C: Cherbourg.
  - Q: Queenstown.
  - S: Southampton.

Seguimos explorando nuestras variables mediante **estadísticos descriptivos**:

```
# Resumen de los datos:
summary(titanic_train)
```

```
##   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
##
```

Visualizando los estadísticos anteriores, hemos detectado que en la variable "Age" existen 177 casos con valores NA's.

Limpieza de datos:

De todos modos vamos a recomprobar, así como chequear si existen **espacios** en blanco y/o duplicidades:

```
# Comprobamos en qué columnas tenemos NAs:
NAcount <- apply(is.na(titanic_train), 2, sum) >= 1
NAcount[NAcount==TRUE]
```

```
##  Age
## TRUE
```

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

```
# Comprobamos en qué variables tenemos espacios en blanco y cuántos:
WScount <- colSums(titanic_train == "" | titanic_train == " " | titanic_train == "  ")
WScount[WScount >= 1 & !is.na(WScount)]
```

```
##    Cabin Embarked
##      687        2
```

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

```
# Comprobamos registros duplicados:
sum(duplicated(titanic_train))
```

```
## [1] 0
```

Vemos como es cierto que en la variable "Age", existen valores NA y por otro lado, podemos también percatarnos de que las variables "Cabin" y "Embarked" presentan algunos espacios en blanco, no existen registros duplicados.

```
# Número de registros:
n <- nrow(titanic_train)
n
```

```
## [1] 891
```

```
# Comprobamos el porcentaje de NA's:
age_NA <- (sum(is.na(titanic_train$Age)) / n) * 100
cat(sprintf("El porcentaje de NA's en la variable *Age* es de %.2f%%", age_NA), "\n")
```

```
## El porcentaje de NA's en la variable *Age* es de 19.87%
```

```
# Comprobamos el porcentaje de espacios en blanco:
cabin_ws <- (sum(titanic_train$Cabin == "" |
                 titanic_train$Cabin == " " |
                 titanic_train$Cabin == "  ") / n) * 100
cat(sprintf("El porcentaje de espacios en blanco en la variable *Cabin* es de %.2f%%",
            cabin_ws), "\n")
```

```
## El porcentaje de espacios en blanco en la variable *Cabin* es de 77.10%
```

Hemos hallado estos porcentuales, porque es buena costumbre establecer un baremo sobre el que decidir qué hacer con variables que presenten cierto porcentaje de este tipo de valores, por ejemplo:

- Por encima del 95% de valores nulos, descartamos la característica.
- Entre el 50% y el 95% de valores nulos, no rellenamos los datos.
- Por debajo del 50% de valores nulos, estos se rellenan siguiendo alguna técnica de imputación.

Dado esto anterior, llegamos a dos **conclusiones**:

- **Imputaremos** los valores de la variable "Age", al ser pocos en relativo.
- **Eliminaremos** la variable "Cabin", ya que además de contener un número elevado de valores en blanco, es una característica que no aporta valor a nuestro análisis (nº de cabina).

Selección de datos:

Aprovechando esto anterior, también incluiremos en este apartado de **selección de características** la eliminación de otra variable que no aporta al análisis como es el nº de ticket "Ticket", para así reducir la dimensionalidad:

```
# Eliminamos la variable "Cabin":
titanic_train$Cabin <- NULL
titanic_train$Ticket <- NULL
```

Podríamos incluso realizar un **PCA**, pero no lo vemos necesario después de estas eliminaciones.

Imputación de los datos:

Es muy común el uso de kNN a la hora de imputar, pero en nuestro caso haremos uso de un método que además de permitirnos trabajar con juegos de datos mixtos multidimensionales (en cuanto a tipos de variables) resulta ser más robusto como **missForest**.

Mediante la imputación, lograremos no perder información que puede resultar relevante.

```
# Transformamos a factor:
titanic_train$Survived <- as.factor(titanic_train$Survived)
titanic_train$Pclass <- as.factor(titanic_train$Pclass)
titanic_train$Sex <- as.factor(titanic_train$Sex)
titanic_train$Embarked <- as.factor(titanic_train$Embarked)

type <- sapply(titanic_train, class)
type
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##   "integer"    "factor"    "factor" "character"    "factor"   "numeric"
##       SibSp       Parch        Fare    Embarked
##   "integer"   "integer"   "numeric"    "factor"
```

Las hemos transformado a tipo *factor* ya que como bien sabemos existen algoritmos que requieren de este tipo de variables para trabajar y además esto nos permite hacer gráficas y por tanto analizar visualmente las variables.

Ahora sí, iremos con la imputación:

```
# Instalamos y cargamos:
if (!require('missForest')) install.packages('missForest'); library('missForest')
```

```
## Loading required package: missForest
```

```
## Warning: package 'missForest' was built under R version 4.1.3
```

```
# Creamos un conjunto sólo con las variables de interés (y escluyendo tipo character):
titanic_train_imp <- titanic_train[, c("Survived",
                                       "Pclass", "Sex",
                                       "Age", "SibSp",
                                       "Parch", "Fare",
                                       "Embarked")]

# Semilla para código reproducible:
set.seed(1000)

# Aplicamos el algoritmo
imp <- missForest(titanic_train_imp , verbose = TRUE, variablewise = TRUE)
```

```
##   missForest iteration 1 in progress...done!
##     estimated error(s): 0 0 0 139.2648 0 0 0 0
##     difference(s): 0.002976575 0
##     time: 0.08 seconds
##
##   missForest iteration 2 in progress...done!
##     estimated error(s): 0 0 0 138.9033 0 0 0 0
##     difference(s): 5.226314e-05 0
##     time: 0.08 seconds
##
##   missForest iteration 3 in progress...done!
##     estimated error(s): 0 0 0 140.8009 0 0 0 0
##     difference(s): 2.708304e-05 0
##     time: 0.11 seconds
##
##   missForest iteration 4 in progress...done!
##     estimated error(s): 0 0 0 141.8477 0 0 0 0
##     difference(s): 4.248775e-05 0
##     time: 0.08 seconds
```

```
# Sustituimos la variable Age por la que hemos logrado imputar:
titanic_train$Age <- imp$ximp$Age

# Vemos cómo ha quedado y el número de variable resultantes:
summary(titanic_train)
```

```
##   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             Fare        Embarked
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Min.   :  0.00    :  2
##  1st Qu.:21.00   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:  7.91   C:168
##  Median :28.75   Median :0.000   Median :0.0000   Median : 14.45   Q: 77
##  Mean   :29.66   Mean   :0.523   Mean   :0.3816   Mean   : 32.20   S:644
##  3rd Qu.:36.73   3rd Qu.:1.000   3rd Qu.:0.0000   3rd Qu.: 31.00
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000   Max.   :512.33
```

```r
str(titanic_train)
```

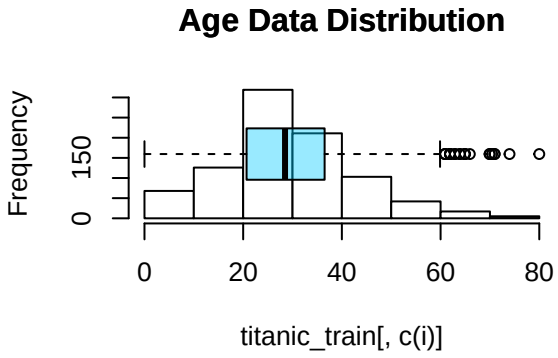
```
## 'data.frame':    891 obs. of  10 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

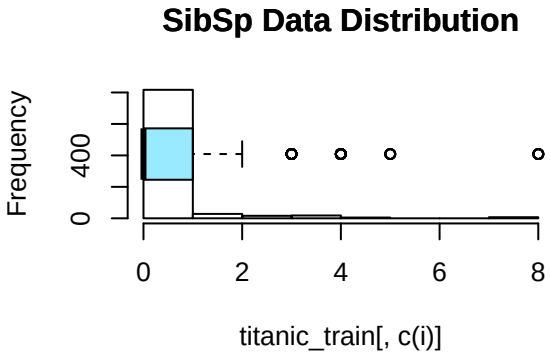
El output anterior, nos muestra como ciertamente, ya no existen registros NA en la variable "Age".

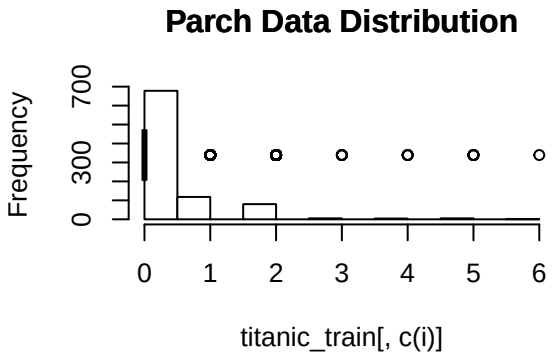
Análisis y graficación (Distribución / Normalidad / Homocedasticidad)

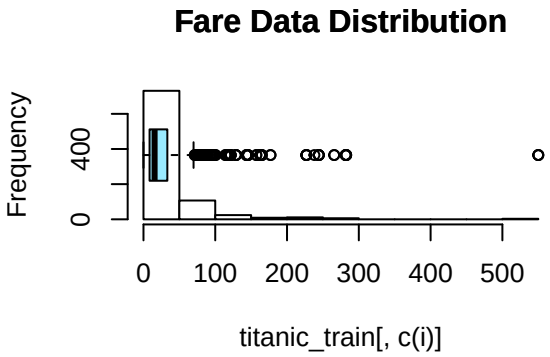
Distribución: Ahora, continuaremos analizando gráficamente **cómo se distribuyen los datos** de nuestras **variables numéricas**:

```r
# Analizaremos la distribución de aquellas que son numéricas:
par(mfrow = c(2,2))

# Aprovechando el siguiente ciclo, realizaremos dos gráficos superpuestos para cada
# variable que son típicos a la hora de visualizar distribuciones de datos,
# como son un Histograma y un BoxPlot:
for (i in c("Age", "SibSp", "Parch", "Fare")) {
  title <- sprintf("%s Data Distribution", i)
  hist(titanic_train[, c(i)], freq = TRUE, main = title, col = "white")
  par(new = TRUE)
  boxplot(titanic_train[, c(i)],
          col = rgb(0, 0.8, 1, alpha = 0.4),
          horizontal = TRUE, main = title,
          axes = FALSE)
}
```









Podemos observar que existen **outliers** en algunas de ellas, pero debemos dimensionar si su número es crítico, o no, por eso veremos qué porcentaje suponen para cada variable:

```
# Analizamos qué porcentaje suponen los outliers de cada variable sobre el total de registros:
for (i in c("Age", "SibSp", "Parch", "Fare")) {
  d <- length(boxplot.stats(titanic_train[, c(i)])$out) / nrow(titanic_train) * 100
  text <- sprintf("Los outliers para %s suponen un %f", i, d)
  print(text)
}
```

```
## [1] "Los outliers para Age suponen un 2.469136"
## [1] "Los outliers para SibSp suponen un 5.162738"
## [1] "Los outliers para Parch suponen un 23.905724"
## [1] "Los outliers para Fare suponen un 13.019080"
```

De estas visualizaciones y porcentajes, podemos extraer varias conclusiones:

- La variable **Age**, presenta algún valor atípico a partir de 60 años con un máximo de 80, algo que podría darse perfectamente en la realidad. El % de outliers es ínfimo.
- En cuanto a **SibSp**, sí que presenta atípicos, ocurre como el en caso anterior, puede darse que un pasajero/pasajera contara con 8 hermanos más cónyuges (el máximo es 8). El % es muy bajo.
- Respecto a **Parch**, mismo caso que el anterior aunque contamos con un mayor porcentaje (casi 24%).
- En **Fare** contamos con un % no muy elevado de outliers, pero sí parece que existe un caso atípico muy desmarcado del resto.

- Como **conclusión general** a todas ellas, vemos que se encuentran en diferentes **escalas**, por lo que esto complica su análisis e interpretabilidad.

Análisis de Correlaciones, Wilcoxon y selección variables/grupos:

Correlaciones variables numéricas: A continuación, vamos a realizar un análisis de **correlaciones** entre las variables numéricas del dataset.

A priori, no conocemos cómo se distribuyen las variables pero sí nos podemos hacer una idea bastante precisa interpretando el análisis gráfico anterior, deduciendo que no se distribuyen como variables normales, a lo sumo en la única que podríamos tener duda en en "Age".

Aún así, lo que haremos será usar tanto el método de correlación de **Spearman** (no paramétrico) como el de **Pearson** (paramétrico) por curiosidad de comparar ambos resultados y porque consideramos que se podrían aplicar test paramétricos adheriéndonos al TCL (N>30), aún así **Spearman** sería el que prevalecería en este caso dado lo comentado.

```
# Semilla:
set.seed(333)

# Instalamos y cargamos:
if (!require('BBmisc')) install.packages('BBmisc'); library('BBmisc')
```

```
## Loading required package: BBmisc

## Warning: package 'BBmisc' was built under R version 4.1.3

##
## Attaching package: 'BBmisc'

## The following object is masked from 'package:base':
##
##     isFALSE
```

```
# Dividiremos los datasets en numérico y factor, para analizar las
# correlaciones según el tipo de variable, en este caso las numéricas:
titanic_train_num <- titanic_train[, c("Age", "SibSp", "Parch", "Fare")]

# Comprobamos el tipo de variables:
sapply(titanic_train_num, class)
```

```
##       Age     SibSp     Parch      Fare
## "numeric" "integer" "integer" "numeric"
```

```
# Normalizamos a la misma escala según Z-score:
titanic_train_num_norm <- normalize(titanic_train_num)

# Paramétrico:
p <- cor(titanic_train_num_norm, method = "pearson")

# No paramétrico:
s <- cor(titanic_train_num_norm, method = "spearman")
```

Ahora, analizaremos las correlaciones entre los pares de variables:

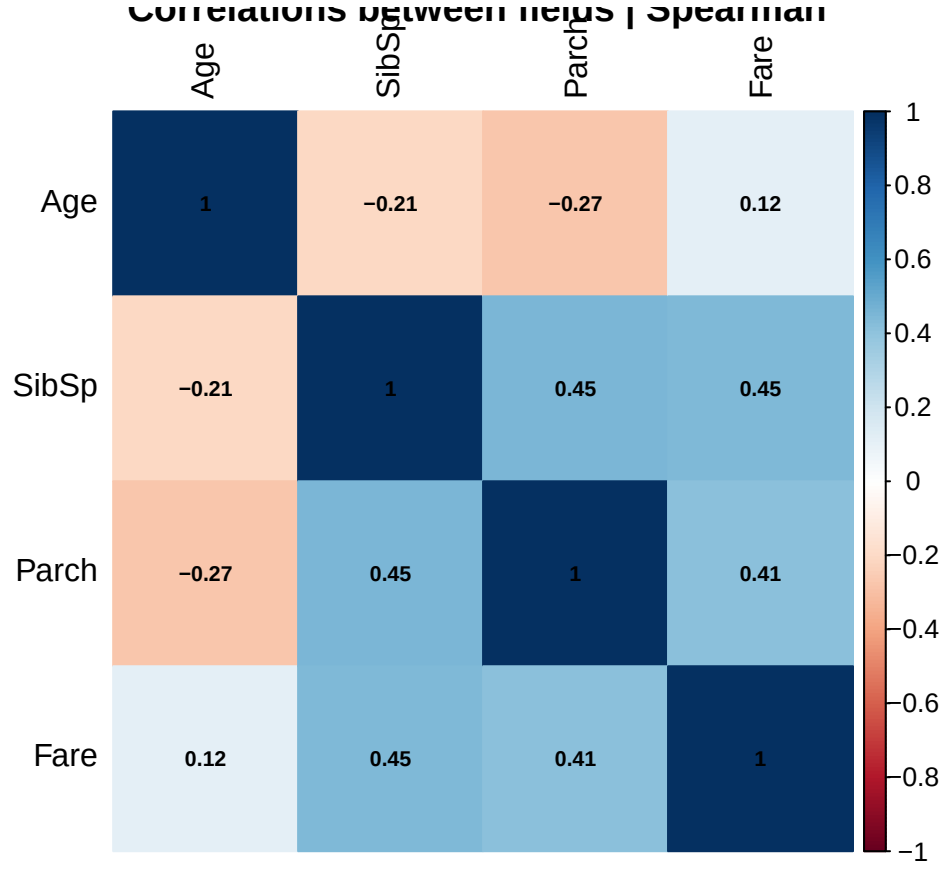
```
# Instalamos y cargamos:
if (!require('corrplot')) install.packages('corrplot'); library('corrplot')
```

```
## Loading required package: corrplot
```

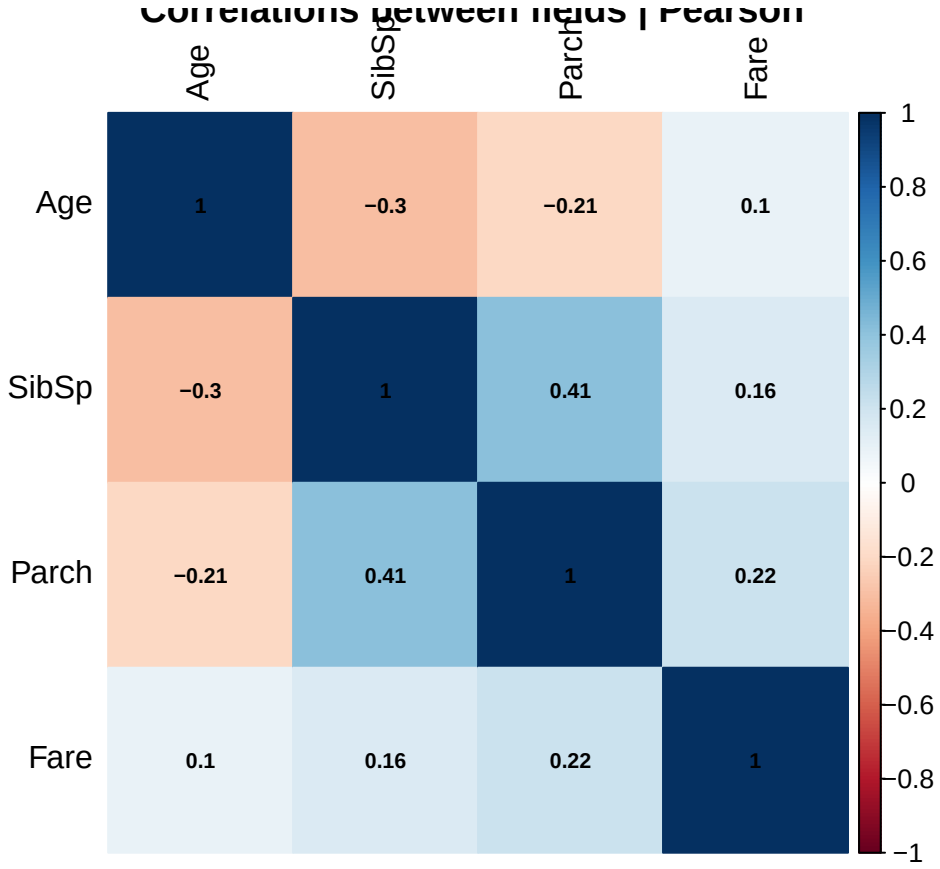
```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
# No paramétrico:
corrplot(s, method = "color", type = "full", diag = TRUE,
         title = "Correlations between fields | Spearman",
         tl.col = "black", number.cex=0.7, sig.level = 0.05,
         addCoef.col = "black")
```




```
# Paramétrico:
corrplot(p, method = "color", type = "full", diag = TRUE,
         title = "Correlations between fields | Pearson",
         tl.col = "black", number.cex=0.7, sig.level = 0.05,
         addCoef.col = "black")
```



Vamos a comentar las conclusiones de este análisis:

Spearman:

Vemos que las mayores correlaciones (que además son directas/positivas) aunque no muy significativas (máx de 0.45) se dan para la variable "Fare" (tarifa del pasajero), concretamente con "SibSP" (nº de hermanos/hermanas, cónyuges) y "Parch" (nº de padres/madres, hijos/hijas), lo cuál parece tener sentido ya que podría darse que a mayor número de integrantes de la familia, la tarifa fuese más elevada.

Por otro lado, en cuanto a las correlaciones negativas estas son menos significativas, y se dan para "Age" (edad) y especificamente con las variables "SibSP" y "Parch". Quizá esto podría indicarnos que las personas mayores, viajan con menor número de familiares, podría ser lógico por el perfil de edad.

Comentar que, hemos valorado discretizar edad generando grupos como niños, adultos, 3ª edad... pero hemos creído que extenderíamos demasiado la PRA, a pesar que es un punto muy interesante a tratar en este caso.

Pearson:

Usando este método y al contrario de Spearman, las correlaciones se dan entre menos número de variables y con menor grado, así pues podemos ver como relativamente significativa (que no lo es realmente) la correlación positiva entre "SibSp" y "Parch", quizá también entre "Fare" y "Parch". Respecto a correlaciones negativas, podemos ver que hay una escasa correlación entre "Fare" y "Parch".

Como último punto a este análisis, comentar que si hubiéramos encontrado una gran correlación entre variables, además de haber llegado a conclusiones más robustas podríamos incluso haber planteado un PCA para la reducción de la dimensionalidad del dataset.

Wilcoxon: Dado que intuimos que las variables no se distribuyen como una normal, aplicaremos el test no paramétrico de **Wilcoxon** que nos ayudará a comparar el rango medio entre dos muestras y determinar

si existen diferencias entre ellas, en nuestro caso las muestras se dividirán según nuestra variable objetivo "Survived".

Por otro lado, si nuestras variables se distribuyesen según una normal, el **t-test** sería el test estadístico paramétrico que parmitiría contrastar la hipótesis nula de que las medias de dos poblaciones son iguales, frente a la hipótesis alternativa de que no lo son.

Vemos con **Wilcoxon**:

```
# Semilla:
set.seed(333)

# Incluimos la variable objetivo:
titanic_train_num_norm$Survived <- titanic_train$Survived

# Aplicamos el test a nuestras variables numéricas normalizadas:
for (i in colnames(titanic_train_num_norm[,c("Age", "SibSp", "Parch", "Fare")])) {
  test <- wilcox.test(titanic_train_num_norm[, i] ~ titanic_train_num_norm$Survived)
  title <- sprintf("Test de Wilcoxon para %s vs Survived", i)
  print(title)
  print(test)
  cat("\n\n")
}
```

```
## [1] "Test de Wilcoxon para Age vs Survived"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  titanic_train_num_norm[, i] by titanic_train_num_norm$Survived
## W = 102202, p-value = 0.02587
## alternative hypothesis: true location shift is not equal to 0
##
##
##
## [1] "Test de Wilcoxon para SibSp vs Survived"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  titanic_train_num_norm[, i] by titanic_train_num_norm$Survived
## W = 85775, p-value = 0.008017
## alternative hypothesis: true location shift is not equal to 0
##
##
##
## [1] "Test de Wilcoxon para Parch vs Survived"
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  titanic_train_num_norm[, i] by titanic_train_num_norm$Survived
## W = 82385, p-value = 3.712e-05
## alternative hypothesis: true location shift is not equal to 0
##
##
##
## [1] "Test de Wilcoxon para Fare vs Survived"
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  titanic_train_num_norm[, i] by titanic_train_num_norm$Survived
## W = 57807, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Comentaremos los resultados del análisis anterior:

- **Age vs Survived**: el p-valor es menos a 0.05, por lo que rechazamos la hipótesis nula de que no existen diferencias.
- **SibSp vs Survived**: el p-valor es menos a 0.05, por lo que rechazamos la hipótesis nula de que no existen diferencias.
- **Parch vs Survived**: el p-valor es menos a 0.05, por lo que rechazamos la hipótesis nula de que no existen diferencias.
- **Fare vs Survived**: el p-valor es menos a 0.05, por lo que rechazamos la hipótesis nula de que no existen diferencias.

Vemos como todas ellas, presentan diferencias según si han sobrevivido o no, pero para nuestro proyecto hemos decidido seleccionar únicamente "Age" y "Fare" al resultarnos variables interesantes de estudiar.

Correlaciones variables categóricas: A seguir, realizaremos el análisis de correlaciones pero en este caso sobre las variables categóricas, o más bien tipo factor en nuestro caso:

```r
# Dataset factores:
titanic_train_fac <- titanic_train[, c("Survived", "Pclass", "Sex", "Embarked")]

# Chequeamos el tipo de variable:
sapply(titanic_train_fac, class)
```

```
## Survived   Pclass      Sex Embarked
## "factor" "factor" "factor" "factor"
```

```r
# Estructura:
str(titanic_train_fac)
```

```
## 'data.frame':    891 obs. of  4 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

Lo que haremos será ir comparando 1 por 1 las variables contra nuestra variable objetivo, haciendo uso de los test de significancia de **Phi** y de **Cramer** así como mostrando las tablas de contingencia en absoluto y en relativo.

```r
# Semilla:
set.seed(333)

# Instalamos y cargamos:
if (!require('DescTools')) install.packages('DescTools'); library('DescTools')
```

```
## Loading required package: DescTools
```

```
## Warning: package 'DescTools' was built under R version 4.1.3
```

```
##
## Attaching package: 'DescTools'
```

```
## The following object is masked from 'package:BBmisc':
##
##     %nin%
```

```
# Generaremos bloques de información por cada pareja de variables:
for (i in colnames(titanic_train_fac)){
  t1 <- table(titanic_train_fac[, c(i)], titanic_train_fac$Survived)
  t2 <- prop.table(t1, margin = 1)

  title1 <- sprintf("%s vs Survived | Phi: ", i)
  title2 <- sprintf("%s vs Survived | V Cramer: ", i)

  cat(title1, Phi(t1), "\n")
  cat(title2, CramerV(t1), "\n")
  print(t1)
  print(t2)
  cat("\n --------\n\n")
}
```

```
## Survived vs Survived | Phi:  1
## Survived vs Survived | V Cramer:  1
##
##       0   1
##   0 549   0
##   1   0 342
##
##     0 1
##   0 1 0
##   1 0 1
##
##  --------
##
## Pclass vs Survived | Phi:  0.3398174
## Pclass vs Survived | V Cramer:  0.3398174
##
##       0   1
##   1  80 136
##   2  97  87
##   3 372 119
##
##             0         1
##   1 0.3703704 0.6296296
##   2 0.5271739 0.4728261
##   3 0.7576375 0.2423625
##
##  --------
```

```
##
## Sex vs Survived | Phi:  0.5433514
## Sex vs Survived | V Cramer:  0.5433514
##
##            0   1
##   female  81 233
##   male   468 109
##
##                   0         1
##   female 0.2579618 0.7420382
##   male   0.8110919 0.1889081
##
##  --------
##
## Embarked vs Survived | Phi:  0.1824838
## Embarked vs Survived | V Cramer:  0.1824838
##
##       0   1
##       0   2
##   C  75  93
##   Q  47  30
##   S 427 217
##
##             0         1
##     0.0000000 1.0000000
##   C 0.4464286 0.5535714
##   Q 0.6103896 0.3896104
##   S 0.6630435 0.3369565
##
##  --------
```

Antes de tratar las conclusiones, comentamos cómo interpretar Phi y Cramer:

- **Phi**: El coeficiente phi es el valor de la chi cuadrado entre el número de observaciones, un valor próximo a 0 indica independencia entre los factores, valores próximos a 1 implican relación entre los factores.
- **CramerV o V de Cramer**: Muy habitual para medir la relación entre factores. También 0 implica independencia y 1 una relación perfecta entre los factores.

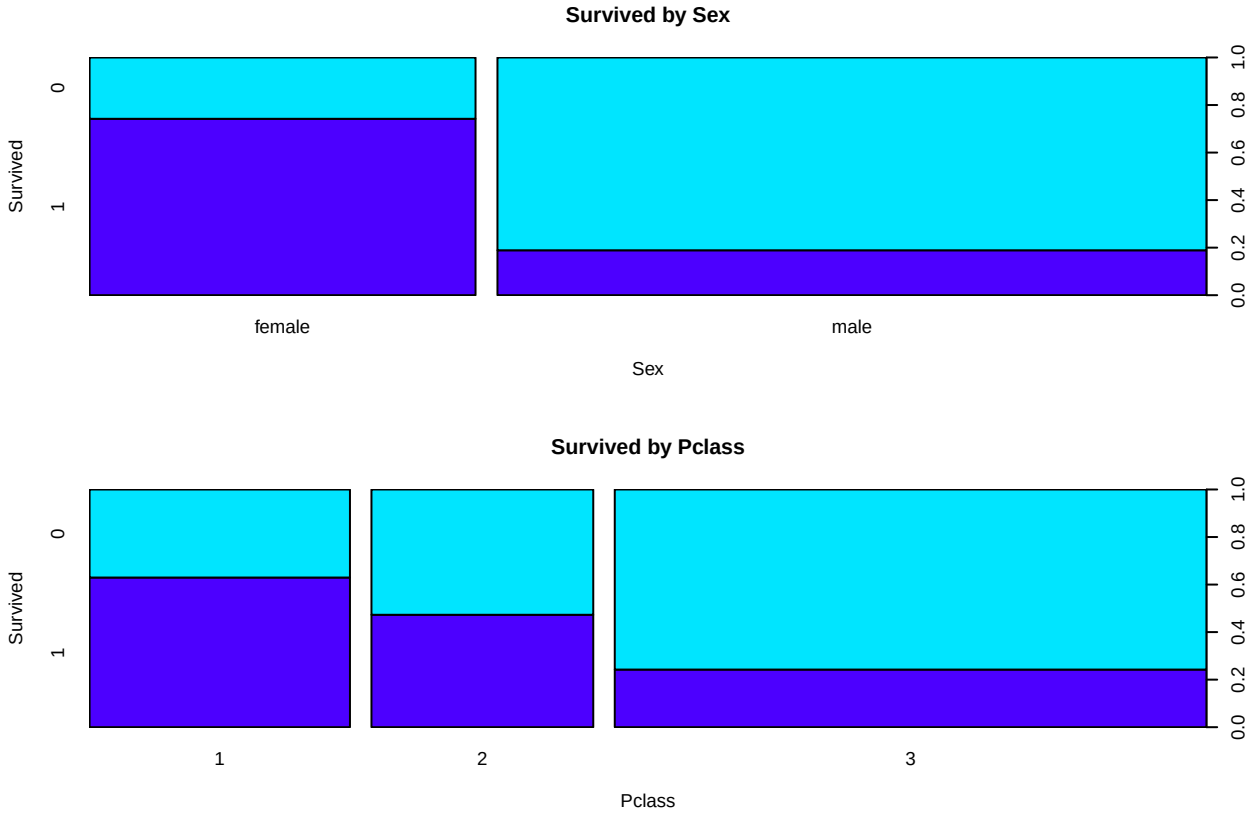
Para ambos test, valores superiores a 0,30 nos indican que hay una posible relación entre las variables.

Conclusiones:

- Los coeficientes son iguales, al ser tablas de contingencia de 2x2.
- Las correlaciones relevantes encontrada se dan entre nuestra variable objetivo y "Sex" y en menor medida "Pclass" (al presentar un coeficiente mayor a 0.3). Esto además se puede intuir/comprobar en la tabla de contingencia en relativo, por ejemplo para la variable "Sex", se observa como el 81% de los hombres no sobreviven, mientras que el 75% de las mujeres sí lo hacen.

Para continuar con nuestro análisis, seleccionaremos "Pclass" y "Sex" al ser las que realmente están correlacionadas, y procedemos a graficar las tablas de contingencia de cada una de las variables vs nuestra variable objetivo "Survived":

```r
# Analizaremos la distribución de aquellas que son numéricas:
par(mfrow = c(2,1), cex=0.55, bty='o')

# Aprovechando el siguiente ciclo,
# realizaremos un gráfico comparando default
# con cada variable tipo factor:
for (i in colnames(titanic_train_fac[, c("Sex", "Pclass")])) {
  title <- sprintf("Survived by %s", i)
  plot(x = titanic_train_fac[, c(i)], y = titanic_train_fac[, c("Survived")], las = 3,
       col = topo.colors(2), main = title,
       xlab = i, ylab = "Survived")
}
```

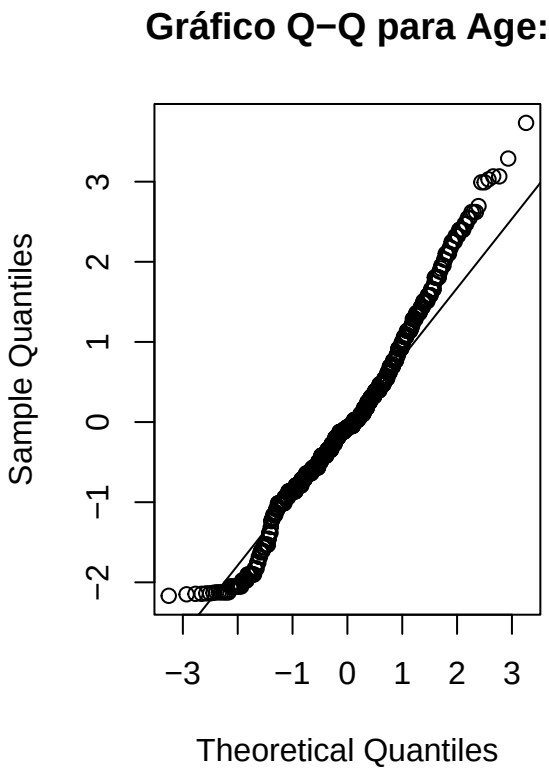


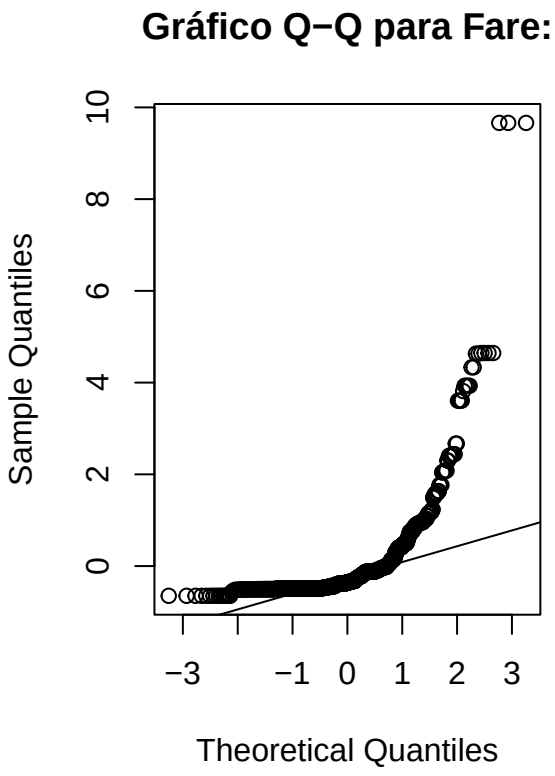
Normalidad: Para verificar que siguen una **distribución normal**, usaremos una de las pruebas más habituales y robustas como es la de *Shapiro-Wilk*.

```r
# Semilla:
set.seed(333)

# Para mostrar gráficos:
par(mfrow = c(1,2))

# Usaremos además de Shapiro uno de los típicos
# gráficos de comprobación, como es el de quantiles Q-Q:
for (i in c("Age", "Fare")) {
```

```r
  norm <- shapiro.test(titanic_train_num_norm[, c(i)])
  text <- sprintf("Test de Shapiro para %s:", i)
  text_2 <- sprintf("Gráfico Q-Q para %s:", i)
  print(text)
  print(norm)
  qqnorm(titanic_train_num_norm[, c(i)], main = text_2)
  qqline(titanic_train_num_norm[, c(i)])
}
```

```
## [1] "Test de Shapiro para Age:"
##
##  Shapiro-Wilk normality test
##
## data:  titanic_train_num_norm[, c(i)]
## W = 0.98025, p-value = 1.284e-09

## [1] "Test de Shapiro para Fare:"
##
##  Shapiro-Wilk normality test
##
## data:  titanic_train_num_norm[, c(i)]
## W = 0.52189, p-value < 2.2e-16
```





Observamos que, para todos los casos **rechazamos la hipótesis nula H0** que defiende la normalidad de la variable, además podemos observar y corroborar esto en los gráficos de cuantiles, donde para poder afirmar

o conjeturizar con que los datos NO se distribuyen como una normal, los puntos deberían ceñirse a la línea representada.

- Como última conclusión en este punto, a pesar de que según lo anterior no se distribuirían como una normal, el **TCL** expresa que si el nº de registros es grande (N>30), su media muestral se aproxima a una normal, pudiendo aplicar test paramétricos. A lo sumo, podríamos hacer uso de una transformación **Box-Cox** para intentar que se distribuyesen como variables normales, pero como se ha podido observar ya hemos hecho uso de test no paramétricos.

Homocedasticidad: Después de realizar un test de Shapiro-Wilks sobre los datos y ver que estos no se distribuyen siguiendo una normal, se realizará un test para comprobar la homocedasticidad de los datos. El supuesto de homocedasticidad considera que la varianza es constante, es decir, no varía entre diferentes grupos. Como no se cumple la condición de normalidad en los datos, se opta por realizar un test de Fligner-Killeen, un test no paramétrico que compara las varianzas basándose en la mediana.

```
# Semilla:
set.seed(333)

# Homocedasticidad:
for (i in colnames(titanic_train_fac[, c("Sex", "Pclass")])) {
  norm <- fligner.test(as.integer(titanic_train_fac$Survived) ~ titanic_train_fac[, i])
  # LeveneTest(as.integer(titanic_train$Survived) ~ titanic_train[, i])
  text <- sprintf("Test de Fligner para %s:", i)
  print(text)
  print(norm)
}
```

```
## [1] "Test de Fligner para Sex:"
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  as.integer(titanic_train_fac$Survived) by titanic_train_fac[, i]
## Fligner-Killeen:med chi-squared = 5.7729, df = 1, p-value = 0.01627
##
## [1] "Test de Fligner para Pclass:"
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  as.integer(titanic_train_fac$Survived) by titanic_train_fac[, i]
## Fligner-Killeen:med chi-squared = 35.766, df = 2, p-value = 1.712e-08
```

```
str(titanic_train_fac)
```

```
## 'data.frame':    891 obs. of  4 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Embarked: Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

Podemos observar a partir del test de homocedasticidad, que el resultado del p-valor tanto para los diferentes grupos de sexo como el de los diferentes clases sociales es estadísticamente significativa. Esto quiere decir que el p-valor para ambas pruebas es inferior a un nivel de confianza del 5% por lo que no hay evidencias para afirmar que se cumpla el supuesto de homocedasticidad.

Regresión logística:

La regresión logística no requiere de ciertas condiciones como linealidad, normalidad y homocedasticidad de los residuos que sí lo son para la regresión lineal. Las principales condiciones que este modelo requiere son: Respuesta binaria: La variable dependiente ha de ser binaria

```
# Generaremos el dataset final:
titanic_train_final <- titanic_train_fac[, c("Survived", "Sex", "Pclass")]
titanic_train_final[, c("Age", "Fare")] <- titanic_train_num_norm[, c("Age", "Fare")]

# Comprobamos:
str(titanic_train_final)
```

```
## 'data.frame':    891 obs. of  5 variables:
##  $ Survived: Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Sex     : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Pclass  : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Age     : num  -0.568 0.619 -0.271 0.396 0.396 ...
##  $ Fare    : num  -0.502 0.786 -0.489 0.42 -0.486 ...
```

```
# Semilla:
set.seed(333)

rl <- glm(Survived ~ Sex + Pclass + Age + Fare,
          data = titanic_train_final, family = "binomial")

# Mostramos el resumen:
summary(rl)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + Age + Fare, family = "binomial",
##     data = titanic_train_final)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7436  -0.6696  -0.4037   0.6381   2.4874
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.6504159  0.2669876   9.927  < 2e-16 ***
## Sexmale     -2.5544456  0.1881746 -13.575  < 2e-16 ***
## Pclass2     -1.2465151  0.2941373  -4.238 2.26e-05 ***
## Pclass3     -2.5378777  0.2990159  -8.487  < 2e-16 ***
## Age         -0.5323456  0.1027323  -5.182 2.20e-07 ***
## Fare        -0.0007734  0.1047330  -0.007    0.994
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  797.37  on 885  degrees of freedom
```

```
## AIC: 809.37
##
## Number of Fisher Scoring iterations: 5
```

```
rl$coefficients
```

```
##   (Intercept)       Sexmale       Pclass2       Pclass3           Age
##  2.6504159245 -2.5544456236 -1.2465151100 -2.5378776802 -0.5323455864
##          Fare
## -0.0007734448
```

Los resultados del p-valor nos indican que la relación de supervivencia puede explicarse a partir de la clase social, el sexo y la edad ya que los p-valores de estas variables son estadísticamente significativos para el modelo (todos son inferiores a 0.05). Hay que destacar que la tarifa no es significativa para predecir la variable supervivencia, por lo que podríamos excluirla del modelo.

Además, podemos hacer varias reflexiones:

- Sobrevivir siendo hombre versus sobrevivir siendo mujer, cambia el logarítmo de probabilidades de sobrevivir en -2.55.
- Sobrevivir siendo de la clase 2 versus sobrevivir siendo de la clase 1, cambia el logaritmo de probabilidades de sobrevivir en -1.24, mientras que ser de la clases tres versus ser de la clase 1 lo cambia en -2.53.
- Por cada año de edad del pasajero, el logaritmo de probabilidades de su supervivencia disminuye en -0.5323456.

Nuestro modelo final sería:

```
rl2 <- glm(Survived ~ Sex + Pclass + Age, data = titanic_train_final, family = binomial())
# Mostramos el resumen:
summary(rl2)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Pclass + Age, family = binomial(),
##     data = titanic_train_final)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7439  -0.6697  -0.4037   0.6381   2.4874
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.6496     0.2399  11.046  < 2e-16 ***
## Sexmale      -2.5543     0.1874 -13.634  < 2e-16 ***
## Pclass2      -1.2455     0.2634  -4.728 2.27e-06 ***
## Pclass3      -2.5367     0.2567  -9.881  < 2e-16 ***
## Age          -0.5322     0.1014  -5.248 1.54e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.66  on 890  degrees of freedom
## Residual deviance:  797.37  on 886  degrees of freedom
## AIC: 807.37
##
## Number of Fisher Scoring iterations: 5
```

Vemos como el valor de AIC se reduce de 809.37 a 807.37, por lo que nos quedamos con el modelo que no contiene la variable Fare.

Predicciones:

Antes de nada y al igual que hemos realizado anteriormente, necesitamos imputar los valores de la variable "Age" en este caso sobre el conjunto de test:

```
titanic_test <- read.csv("./test.csv", header = TRUE, sep = ",")
titanic_test$Pclass <- as.factor(titanic_test$Pclass)
titanic_test$Sex <- as.factor(titanic_test$Sex)
titanic_test$Embarked <- as.factor(titanic_test$Embarked)

# Creamos un conjunto sólo con las variables de interés (y escluyendo tipo character):
titanic_test_imp <- titanic_test[, c("Pclass", "Sex",
                                     "Age", "SibSp",
                                     "Parch", "Fare",
                                     "Embarked")]

# Semilla para código reproducible:
set.seed(1000)

# Aplicamos el algoritmo
imp <- missForest(titanic_test_imp , verbose = TRUE, variablewise = TRUE)
```

```
##   missForest iteration 1 in progress...done!
##     estimated error(s): 0 0 149.8306 0 0 1775.478 0
##     difference(s): 0.001640706 0
##     time: 0.07 seconds
##
##   missForest iteration 2 in progress...done!
##     estimated error(s): 0 0 151.5891 0 0 1780.226 0
##     difference(s): 3.300871e-05 0
##     time: 0.08 seconds
##
##   missForest iteration 3 in progress...done!
##     estimated error(s): 0 0 152.4332 0 0 1798.635 0
##     difference(s): 1.921708e-05 0
##     time: 0.08 seconds
##
##   missForest iteration 4 in progress...done!
##     estimated error(s): 0 0 149.941 0 0 1780.13 0
##     difference(s): 2.611032e-05 0
##     time: 0.08 seconds
```

```
# Sustituimos la variable Age por la que hemos logrado imputar:
titanic_test$Age <- imp$ximp$Age

# Vemos cómo ha quedado y el número de variable resultantes:
summary(titanic_test)
```

```
##   PassengerId     Pclass      Name               Sex           Age
##  Min.   : 892.0   1:107   Length:418         female:152   Min.   : 0.17
##  1st Qu.: 996.2   2: 93   Class :character   male  :266   1st Qu.:22.00
##  Median :1100.5   3:218   Mode  :character                Median :26.41
##  Mean   :1100.5                                           Mean   :29.71
##  3rd Qu.:1204.8                                           3rd Qu.:36.38
##  Max.   :1309.0                                           Max.   :76.00
##
##      SibSp            Parch           Ticket               Fare
##  Min.   :0.0000   Min.   :0.0000   Length:418         Min.   :  0.000
##  1st Qu.:0.0000   1st Qu.:0.0000   Class :character   1st Qu.:  7.896
##  Median :0.0000   Median :0.0000   Mode  :character   Median : 14.454
##  Mean   :0.4474   Mean   :0.3923                      Mean   : 35.627
##  3rd Qu.:1.0000   3rd Qu.:0.0000                      3rd Qu.: 31.500
##  Max.   :8.0000   Max.   :9.0000                      Max.   :512.329
##                                                       NA's   :1
##     Cabin           Embarked
##  Length:418         C:102
##  Class :character   Q: 46
##  Mode  :character   S:270
##
##
##
##
```

```
str(titanic_test)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis
##  $ 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     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
```

Continuaremos normalizando los valores de esta variable:

```
# Normalizamos a la misma escala según Z-score:
titanic_test[, c("Age")] <- normalize(titanic_test[, c("Age")])

titanic_test <- titanic_test[, c("Sex", "Pclass", "Age")]
```

```r
titanic_test$Pclass <- as.factor(titanic_test$Pclass)
titanic_test$Sex <- as.factor(titanic_test$Sex)

expected_value <- read.csv("./gender_submission.csv", header = TRUE, sep = ",")
expected_value <- expected_value[, 'Survived']
```

Realizamos la predicción:

```r
# Predicción:
predicion_1 <- ifelse(predict(rl2,newdata = titanic_test, type="response") >=0.5,1,0)

# Cargamos la siguiente librería para generar la matriz de confusión:
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 4.1.3
```

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked from 'package:DescTools':
##
##     MAE, RMSE
```

```r
confusionMatrix(as.factor(expected_value), as.factor(predicion_1))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 251  15
##          1  10 142
##
##                Accuracy : 0.9402
##                  95% CI : (0.913, 0.9609)
##     No Information Rate : 0.6244
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8717
##
##  Mcnemar's Test P-Value : 0.4237
##
##             Sensitivity : 0.9617
```

```
##             Specificity : 0.9045
##          Pos Pred Value : 0.9436
##          Neg Pred Value : 0.9342
##              Prevalence : 0.6244
##          Detection Rate : 0.6005
##    Detection Prevalence : 0.6364
##       Balanced Accuracy : 0.9331
##
##        'Positive' Class : 0
##
```

- La **precisión** del modelo o nº de casos correctamente clasificados es del 94.02% es decir, una tasa de error del 5.98%, se han clasificado correctamente 393 casos de los 418, por lo que nuestro modelo tiene una precisión bastante elevada.
- En cuanto a la **Sensibilidad** (tasa de verdaderos positivos, en este caso proporción de individuos que NO sobreviven clasificados correctamente) y la **Especificidad** (tasa de verdaderos negativos o proporción de individuos que sobreviven clasificados correctamente), los valores son de 96.17% y de 90.45% respectivamente.

Exportamos las predicciones

```
# Generamos un CSV:
write.csv(predicion_1,"./predicctions.csv", row.names = FALSE)
```

Conclusiones finales

Las conclusioens finales nos indican que la edad, el sexo y la clase de una persona son características decisivas para determinar la supervivencia de los pasajeros del Titanic. Las mejores características para sobrevivir son ser mujer ante ser hombre, ser de la clase 1 ante ser de la clase 2 o la 3, y no ser una persona muy mayor de edad. A menos edad, más probabilidades de sovrevivir.

El modelo de regresión logística generado anteriomente, ha logrado clasificar correctamente la gran mayoría de los casos al aplicarlo sobre el conjunto de test como hemos podido ver a través de la matriz de confusión, por lo que dicho modelo podría predecir con una precisión de más del 90% si un individuo sobreviviría o no a bordo del Titanic en base a las variables que seleccionamos en el modelo.

Todas estas conclusiones tienen sentido lógico si se analizan con los hechos históricos que se recogen de la tragedia:

- Había un plan de accidentes ante el cual, los primeros en ser evacuados serían mujeres y niños (de ahí que tuvieran más posibilidades de sobrevivir).
- Es muy probable que las personas que fuesen de la clase 1, tuviesen más poder para ser evacuadas antes que las personas de las otras clases, ya que tenían poder adquisitivo suficiente para podero permitirse ese privilegio.
- Ante una situación de supervivencia, las personas más jovenes tienen más resistencia y fuerza que las personas más mayores.

Contribuciones:

Investigación Previa: DCM & NCM

Redacción de las respuestas: DCM & NCM

Desarrollo código: DCM & NCM