Tipología y ciclo de vida de los datos. Práctica 2

Mateo Rodríguez Lavado y Eduard Conesa Guerrero

20/12/2021

1 Descripción del dataset

Es conocido por muchos que el 15 de abril de 1912, durante una travesía por mar, el Titanic se hundió tras chocar contra un iceberg en su viaje inaugural. Debido a que no había botes salvavidas para todos los pasajeros, murieron muchos de ellos.

Este dataset contiene información sobre los que iban a bordo del Titanic. La información que contiene es la siguiente:

- PassengerId. Identificador del pasajero.
- survival. Indica si el pasajero sobrevivió o no (0 = No, 1 = Si).
- pclass. Tipo de clase del ticket (1 = primera, 2 = segunda, 3 = tercera).
- sex. Sexo
- Age. Años del pasajero.
- sibsp. Número de hermanos o cónyuges a bordo del titanic.
- parch. Numero de padres o hijos a borod del titanic.
- ticket. Número del Ticket
- fare. Tarifa para el pasajero.
- cabin. Número de cabina.
- embarked. Puerto de embarque.

Este dataset puede ayudar a estudiar esta catástrofe y así reducir el número de víctimas en accidentes similares. La pregunta que intentamos responder es la siguiente: ¿Qué tipo de personas tenían más probabilidades de sobrevivir?

El dataset se lee de la siguientes forma:

```
# Se carga el fichero de datos
people<-read.csv("train.csv")

# Se verifica la estructura del conjunto de datos
str(people)
```

```
## '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)" "
Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...
##  $ 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" ...
```

Se compone de un total de 891 personas que iban a bordo del titanic. Se realiza un análisis rapido de las variables para ver un resumen de las mismas.

```
summary(people)
```

```
##   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
##
```

Este análisis muestra que la carga se ha realizado exitosamente.

2 Integración y selección de los datos de interés a analizar

Despues de describir los campos mostrados en el apartado anterior, se realiza una selección de los atributos de interés para cuando se deseen realizar los diferentes modelos.

Por una parte, los atributos de PassengerId y Name son atributos que solo identifican a la persona, con lo que no aportan información relevante de la supervivencia y podrían generar sobrespecialización del modelo. Por otra parte el atributo Cabin presenta un gran numero de campos vacios con lo que tampoco se tendrá en cuenta.

```
people_red<-people[,c("Survived", "Pclass", "Sex", "Age", "SibSp", "Parch", "Ticket", "Fare", "Embarke
d")]
```

Fijandose en los registros obtenidos, existen dos pasajeros que no disponen de valor de “Embarked” con lo que se eliminaran dichos registros, ya que según se puede observar son dos personas relacionadas que probablemente no subiran a bordo y por tanto no es relevante esta información para estimar la supervivencia de los pasajeros que si lo hicieron.

```
people$Survived[which(people$Embarked == "")]
```

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

```
people_red <- people_red[people_red$Embarked != "", ]
```

Se decide estudiar si se puede generar un nuevo campo con las letras que en ocasiones

contiene el ticket.

```
num_Ticket<-as.numeric(people_red$Ticket)
```

```
## Warning: NAs introduced by coercion
```

```
sum(is.na(num_Ticket))
```

```
## [1] 230
```

```
length(num_Ticket)
```

```
## [1] 889
```

Como se observa, hay muchos registros que no disponen de letras (un 25% aproximadamente), por lo que se descarte generar un campo solo para las letras.

A continuación se estudia si las letras tambien son utiles para la identificación del ticket o basta con los numeros.

```
TicketNum <- sapply(strsplit(people_red$Ticket, " ", fixed=TRUE), tail, 1)
head(people_red)
```

```
##   Survived Pclass    Sex Age SibSp Parch           Ticket    Fare Embarked
## 1        0      3   male  22     1     0        A/5 21171  7.2500        S
## 2        1      1 female  38     1     0         PC 17599 71.2833        C
## 3        1      3 female  26     0     0 STON/O2. 3101282  7.9250        S
## 4        1      1 female  35     1     0           113803 53.1000        S
## 5        0      3   male  35     0     0           373450  8.0500        S
## 6        0      3   male  NA     0     0           330877  8.4583        Q
```

```
length(unique(TicketNum))
```

```
## [1] 678
```

```
length(unique(people_red$Ticket))
```

```
## [1] 680
```

```
people_red$Ticket <- NULL
```

Dado que hay un variación de 2 entre los valores únicos del ticket con letra respecto al ticket sin letra se decide no simplificarlo de manera numérica, ya que es necesaria también. Por tanto, el atributo ticket no se mantiene porque no aporta información útil al conjunto de datos en su totalidad ni descomponiendolo.

3 Limpieza de los datos

3.1 Elementos vacíos

A continuación se estudian los campos que presentan valores vacios o nulos.

```
colSums(is.na(people))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0           0           0
```

```
colSums(people==-1)
```

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

```
colSums(people=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

```
colSums(is.na(people_red))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        0        0        0      177        0        0        0        0
```

Como se ha comentado antes, al haber gran cantidad de registros con Cabin vacio, este atributo ha sido eliminado en la fase anterior, asi como tambien los dos registros con Embarked vacio.

Por otra parte se observa que el atributo de edad tiene 177 registros sin valor, igual que en la version reducida del dataset, probablemente porque sea desconocido. Se pueden imputar los valores utilizando por ejemplo el algoritmo de vecinos cercanos basado en la distancia tomando en cuenta las variables Pclass, Fare, SibSp, Parch. Son de especial utilidad las ultimas variables, ya que contienen información en cuanto a los familiares, lo que nos puede indicar si esa persona es mayor o joven.

```
library(VIM)
```

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

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

```
## VIM is ready to use.
```

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

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
people_red <- kNN(people_red , variable=c("Age"), dist_var=c("Pclass","SibSp", "Parch", "Fare"))
sum(is.na(people_red$Age))
```

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

Como se observa ya se han imputado los valores ausentes del atributo Age. Se elimina tambien el ultimo campo añadido al dataset que contiene la información de valores imputados y este no será de utilidad.

```
people_red$Age_imp <- NULL
```

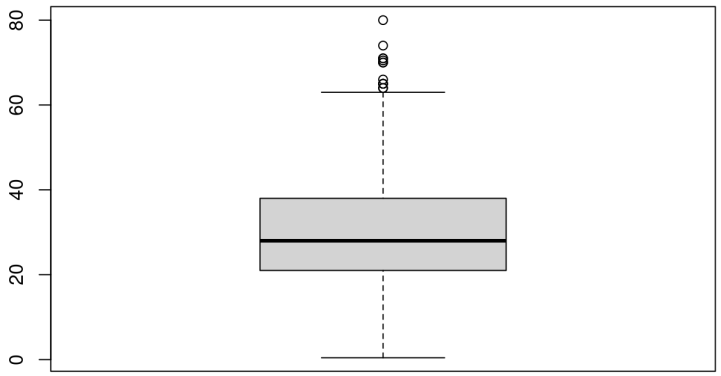
3.2 Valores extremos

Los valores extremos solo se pueden ver en las variables numéricas que en este caso son Age y Fare.

Se procede en primer lugar con la variable Age. Para comprobar los valores extremos se hará un diagrama de caja que muestre si existen estos valores porque tienen una diferencia mayor a tres veces la desviación típica respecto la media.

```
# Boxplot

bpAge <-boxplot(people_red$Age)
```




```
sort(boxplot.stats(people_red$Age)$out)
```

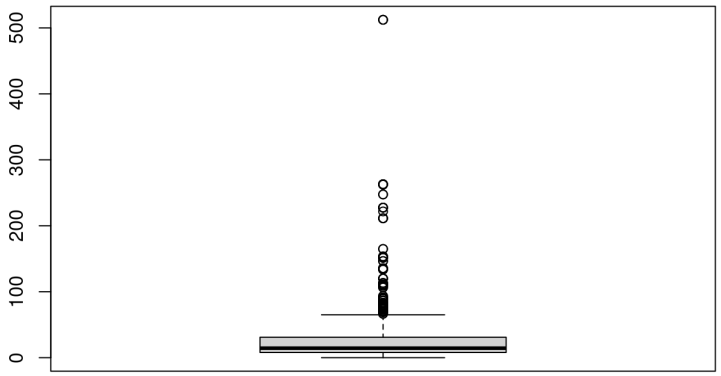
```
##  [1] 64.0 64.0 65.0 65.0 65.0 66.0 70.0 70.0 70.5 71.0 71.0 74.0 80.0
```

Los outliers son personas mayores que 64 años, siendo su máximo 80, pero esto son valores normales dentro de la edad de una persona por lo que no se realiza ningún tratamiento especial.

Si se realiza lo mismo para la variables Fare se tiene:

```
# Boxplot

bpFare <-boxplot(people_red$Fare)
```




```
sort(bpFare$out)
```

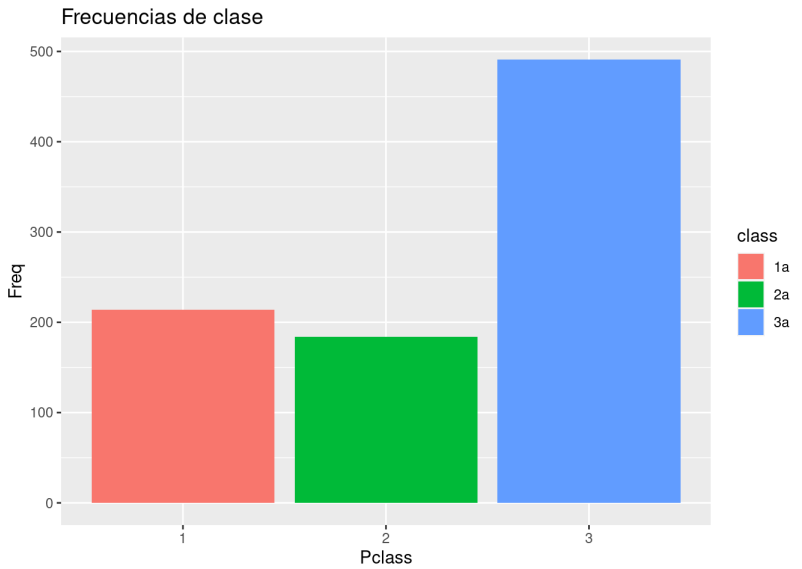
```
##   [1]  66.6000  66.6000  69.3000  69.3000  69.5500  69.5500  69.5500  69.5500
##   [9]  69.5500  69.5500  69.5500  71.0000  71.0000  71.2833  73.5000  73.5000
##  [17]  73.5000  73.5000  73.5000  75.2500  76.2917  76.7292  76.7292  76.7292
##  [25]  77.2875  77.2875  77.9583  77.9583  77.9583  78.2667  78.2667  78.8500
##  [33]  78.8500  79.2000  79.2000  79.2000  79.2000  79.6500  79.6500  79.6500
##  [41]  81.8583  82.1708  82.1708  83.1583  83.1583  83.1583  83.4750  83.4750
##  [49]  86.5000  86.5000  86.5000  89.1042  89.1042  90.0000  90.0000  90.0000
##  [57]  90.0000  91.0792  91.0792  93.5000  93.5000 106.4250 106.4250 108.9000
##  [65] 108.9000 110.8833 110.8833 110.8833 110.8833 113.2750 113.2750 113.2750
##  [73] 120.0000 120.0000 120.0000 120.0000 133.6500 133.6500 134.5000 134.5000
##  [81] 135.6333 135.6333 135.6333 146.5208 146.5208 151.5500 151.5500 151.5500
##  [89] 151.5500 153.4625 153.4625 153.4625 164.8667 164.8667 211.3375 211.3375
##  [97] 211.3375 211.5000 221.7792 227.5250 227.5250 227.5250 227.5250 247.5208
## [105] 247.5208 262.3750 262.3750 263.0000 263.0000 263.0000 263.0000 512.3292
## [113] 512.3292 512.3292
```

Lo más lógico es que todos estos precios altos correspondieran a billetes de primera clase muy selectos o a subidas de precio debido a la oferta y demanda. Lo primero es ver es como se distribuyen estos billetes según la clase.

```
library(ggplot2)
people_red_outFare <- people_red[people_red$Fare %in% bpFare$out,]
class <- c("1a","2a","3a")
freq_pclass_outFare <- as.data.frame(table(people_red_outFare$Pclass))
ggplot(freq_pclass_outFare,aes(x=Var1,y=Freq, fill=class))+geom_bar(stat = "identity", position = "dodg
e") + labs(x="Pclass")+ggtitle("Frecuencias de clase de los outliers")
```

Comparando con la distribución original:

```
freq_pclass <- as.data.frame(table(people_red$Pclass))
ggplot(freq_pclass,aes(x=Var1,y=Freq, fill=class))+geom_bar(stat = "identity", position = "dodge") + labs
(x="Pclass")+ggtitle("Frecuencias de clase")
```



Se observa que la mitad de los pasajeros de primera clase tienen precios mayores de lo esperado. Sin embargo, para las clases segunda y tercera hay muy poco que tengan outliers por lo tanto podremos reemplazar sus valores por la media del valor del ticket por cada clase.

```
#Cojo solo los valores que no sean outliers y calculo la media
people_red_wo_out <- people_red[!(people_red$Fare %in% bpFare$out),]
mean_class <- aggregate(people_red_wo_out$Fare, list(people_red_wo_out$Pclass), mean)
colnames(mean_class) <- c("Pclass", "mean")
#Sustituyo para la clase 2
people_red[(people_red$Fare %in% bpFare$out) & (people_red$Pclass == 2),]$Fare <- mean_class[mean_class$P
class == 2,]$mean
#Sustituyo para la clase 3
people_red[(people_red$Fare %in% bpFare$out) & (people_red$Pclass == 3),]$Fare <- mean_class[mean_class$P
class == 3,]$mean
```

Si se observa nuevamente el diagrama de cajas se aprecia que hay una separación grande entre tickets con valor menor que 200 y mayor que 200, por lo tanto, estos valores también se sustituyen con la media.

```
#Sustituyo para la clase 1
people_red[(people_red$Fare %in% bpFare$out) & (people_red$Pclass == 1) & (people_red$Fare >= 200),]$Fare
<- mean_class[mean_class$Pclass == 1,]$mean
```

Por ultimo, se convierten las clases con valor categórico a factor, para su correcto procesado posterior.

```
people_red$Sex <- as.factor(people_red$Sex)
people_red$Pclass <- as.factor(people_red$Pclass)
people_red$Embarked <- as.factor(people_red$Embarked)
people_red$Survived <- factor(people_red$Survived, levels=c("0", "1"), labels=c("Fallece", "Sobrevive"))
head(people_red)
```

```
##    Survived Pclass    Sex Age SibSp Parch    Fare Embarked
## 1   Fallece      3   male  22     1     0  7.2500        S
## 2 Sobrevive      1 female  38     1     0 71.2833        C
## 3 Sobrevive      3 female  26     0     0  7.9250        S
## 4 Sobrevive      1 female  35     1     0 53.1000        S
## 5   Fallece      3   male  35     0     0  8.0500        S
## 6   Fallece      3   male  21     0     0  8.4583        Q
```

4 Análisis de los datos.

4.1 Selección de los grupos de datos

A continuación, se generan muestras de datos en función de las características más interesantes para investigar y analizar, como el sexo, la clase del pasajero y por supervivencia

```
#Agrupación por sexo
people_red.male <- people_red[people_red$Sex == "male",]
people_red.female <- people_red[people_red$Sex == "female",]

#Agrupación por clase
people_red.first <- people_red[people_red$Pclass == 1,]
people_red.second <- people_red[people_red$Pclass == 2,]
people_red.third <- people_red[people_red$Pclass == 3,]

#Agrupación por Fallecimiento o Supervivencia
people_red.fallecidos <- people_red[people_red$Survived == "Fallece",]
people_red.supervivientes<- people_red[people_red$Survived == "Sobrevive",]
```

Como se observa, ninguna de las muestras contiene menos de 30 elementos, con lo que se puede asumir normalidad en la distribución de las medias según el teorema central del límite para los futuros analisis.

```
nrow(people_red.male); nrow(people_red.female); nrow(people_red.first); nrow(people_red.second); nrow(peo
ple_red.third);nrow(people_red.fallecidos);nrow(people_red.supervivientes)
```

```
## [1] 577
```

```
## [1] 312
```

```
## [1] 214
```

```
## [1] 184
```

```
## [1] 491
```

```
## [1] 549
```

```
## [1] 340
```

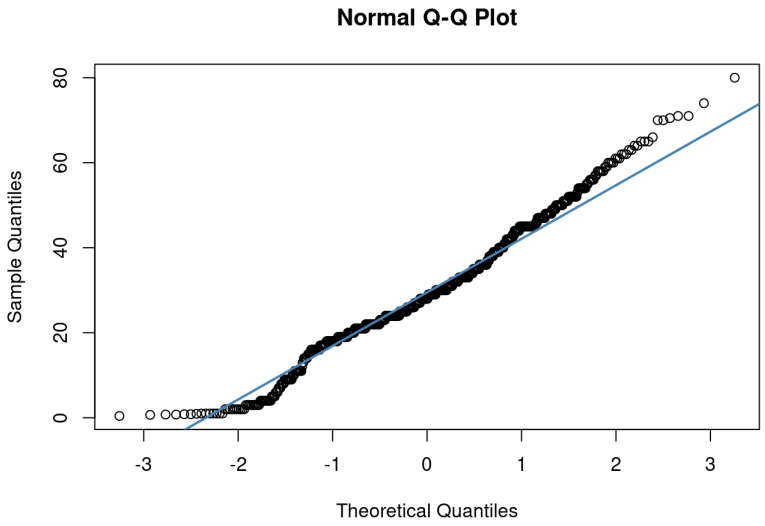
4.2 Comprobación de normalidad y homogeneidad

4.2.1 Normalidad

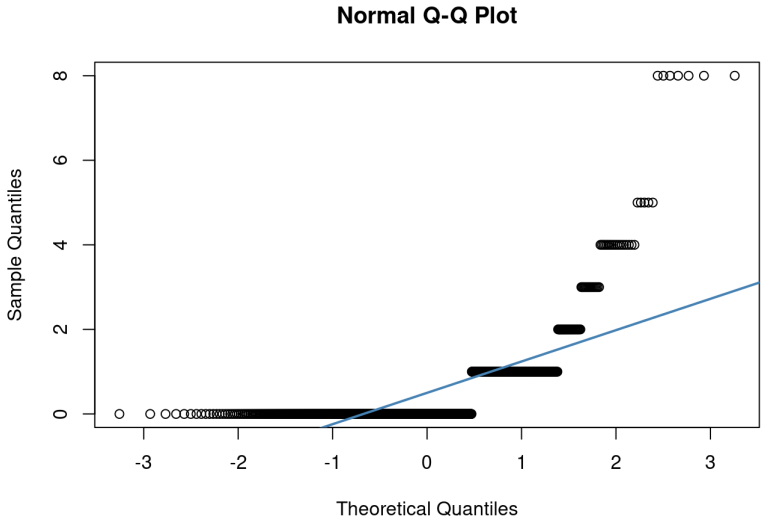
Se analiza si la variables numéricas siguen una distribución normal. Para ello se aplica el test de Shapiro-Wilk asi como se muestra tambien el gráfico cuartil-cuartil.

```
for(c in names(people_red)) {
  if(is.numeric(people_red[[c]])) {
    print(c)
    print(shapiro.test(people_red[[c]]))
    qqnorm(people_red[[c]])
    qqline(people_red[[c]], col = "steelblue", lwd = 2)
  }
}
```

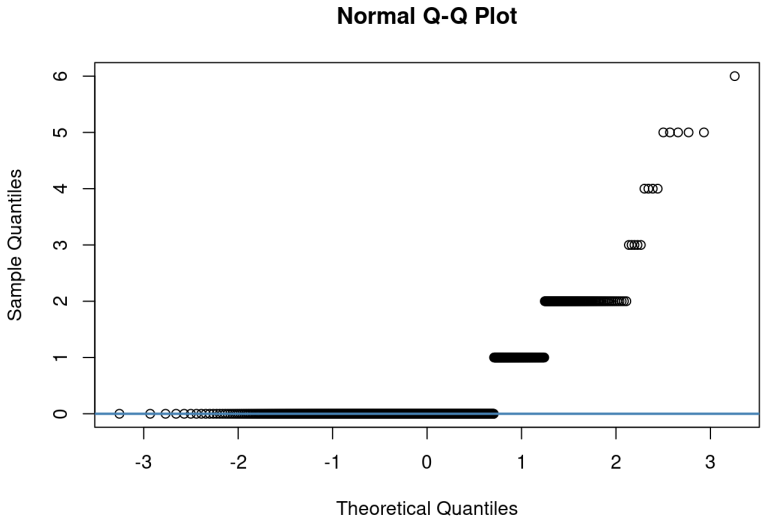
```
## [1] "Age"
##
##  Shapiro-Wilk normality test
##
## data:  people_red[[c]]
## W = 0.98205, p-value = 5.452e-09
```



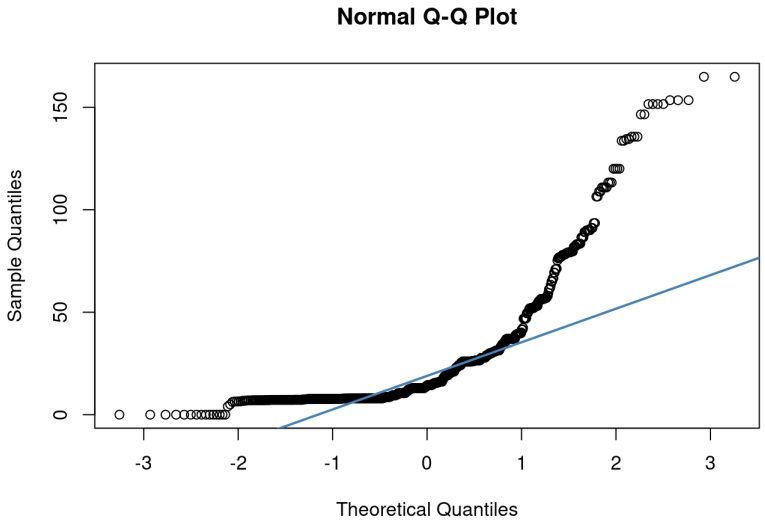

```
## [1] "SibSp"
##
##  Shapiro-Wilk normality test
##
## data:  people_red[[c]]
## W = 0.51353, p-value < 2.2e-16
```




```
## [1] "Parch"
##
##  Shapiro-Wilk normality test
##
## data:  people_red[[c]]
## W = 0.53345, p-value < 2.2e-16
```




```
## [1] "Fare"
##
##  Shapiro-Wilk normality test
##
## data:  people_red[[c]]
## W = 0.69244, p-value < 2.2e-16
```



Se observa que la distribución se aleja mucho de la distribución normal en el gráfico. Esto se corrobora al ver que el p-valor es muy inferior a 0.05 y se rechaza la hipótesis nula. Por lo tanto, no se puede suponer normalidad en las variables.

4.2.2 Homocedasticidad

Ahora se estudiará la homogeneidad de varianzas mediante la aplicación del test de Fligner-Kileen, ya que las variables no cumplen la condición de normalidad.

```
#Age
fligner.test(Age ~ as.factor(Pclass), data = people_red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by as.factor(Pclass)
## Fligner-Killeen:med chi-squared = 16.343, df = 2, p-value = 0.0002826
```

```
fligner.test(Age ~ as.factor(Embarked), data = people_red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by as.factor(Embarked)
## Fligner-Killeen:med chi-squared = 0.53805, df = 2, p-value = 0.7641
```

```
fligner.test(Age ~ as.factor(Sex), data = people_red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Age by as.factor(Sex)
## Fligner-Killeen:med chi-squared = 0.051639, df = 1, p-value = 0.8202
```

Se observa que con Age, tanto Embarked como Sex, tienen un p-valor superior a 0,05 y, por lo tanto, no se puede rechazar la hipótesis nula lo que implica homogeneidad en la varianza.

```
#Fare
fligner.test(Fare ~ as.factor(Pclass), data = people_red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by as.factor(Pclass)
## Fligner-Killeen:med chi-squared = 319, df = 2, p-value < 2.2e-16
```

```
fligner.test(Fare ~ as.factor(Embarked), data = people_red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by as.factor(Embarked)
## Fligner-Killeen:med chi-squared = 111.07, df = 2, p-value < 2.2e-16
```

```
fligner.test(Fare ~ as.factor(Sex), data = people_red)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Fare by as.factor(Sex)
## Fligner-Killeen:med chi-squared = 54.939, df = 1, p-value = 1.243e-13
```

Se observa que para las tres pruebas resulta un p-valor inferior al nivel de significancia (< 0,05). Por lo tanto, se rechaza la hipótesis nula de homocedasticidad y se concluye que la variable Fare presenta varianzas estadísticamente diferentes para los diferentes grupos de Pclass, Embarked y Sex.

4.3 Análisis

4.3.1 Clustering

A continuación, se desea observar si es posible obtener una caracterización de los supervivientes o pasajeros del Titanic, por lo que se aplica un metodo de clusterización basado en k-medoides, utilizando la distancia de Gower, que permite combinar atributos numericos con categóricos. Para hacer esta caracterización se tendrán en cuenta todos los atributos excepto Survived.

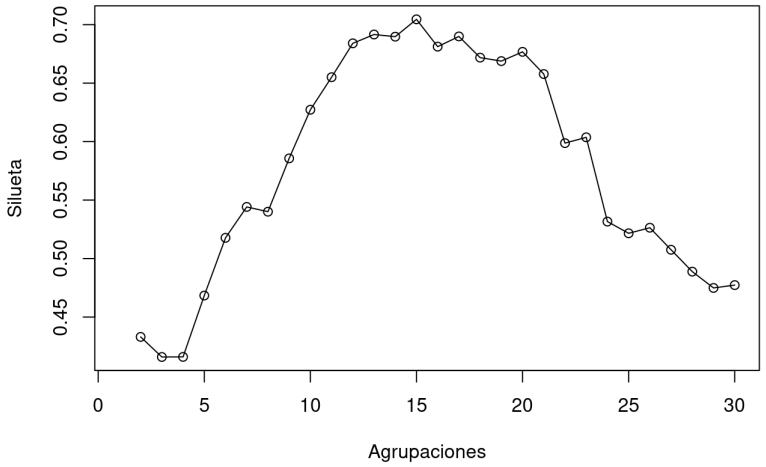
```
library(cluster)
# Se establece el valor de la semilla para la repetibilidad de los datos en el informe
set.seed(1)

#Calculo de métrica de distancia
gower_dist <- daisy(people_red[ , -which(names(people_red) %in% c("Survived"))],
                    metric = "gower",
                    type = list(logratio = 3))
```

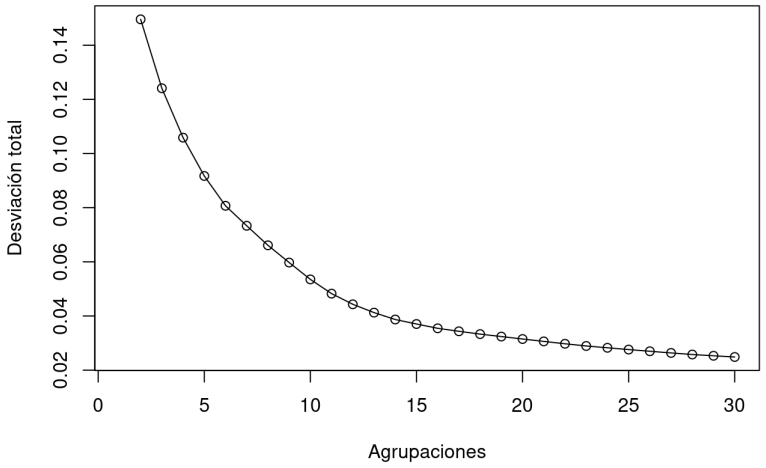
A continuación, se determina el número óptimo de grupos según los criterios de máxima silueta y deviación total.

```
sil_width <- c(NA)
td <- c(NA)
for(i in 2:30){
  pam_fit <- pam(gower_dist,
                 diss = TRUE,
                 k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
  td[i] <- pam_fit$objective
}

plot(1:30, sil_width,
     xlab = "Agrupaciones",
     ylab = "Silueta")
lines(1:30, sil_width)
```




```
plot(1:30, td,
     xlab = "Agrupaciones",
     ylab = "Desviación total")
lines(1:30, td)
```



Como se puede contemplar en las gráficas, el punto óptimo esta en el codo de la gráfica de desviación total, donde existe un máximo de silueta con 12 grupos. Se conforman a continuación los 12 grupos.

```
library(dplyr)
pam_fit <- pam(gower_dist, diss = TRUE, k = 12)

pam_results <- people_red[ , -which(names(people_red) %in% c("Survived"))] %>%
  mutate(cluster = pam_fit$clustering) %>%
  group_by(cluster) %>%
  do(the_summary = summary(.))
```

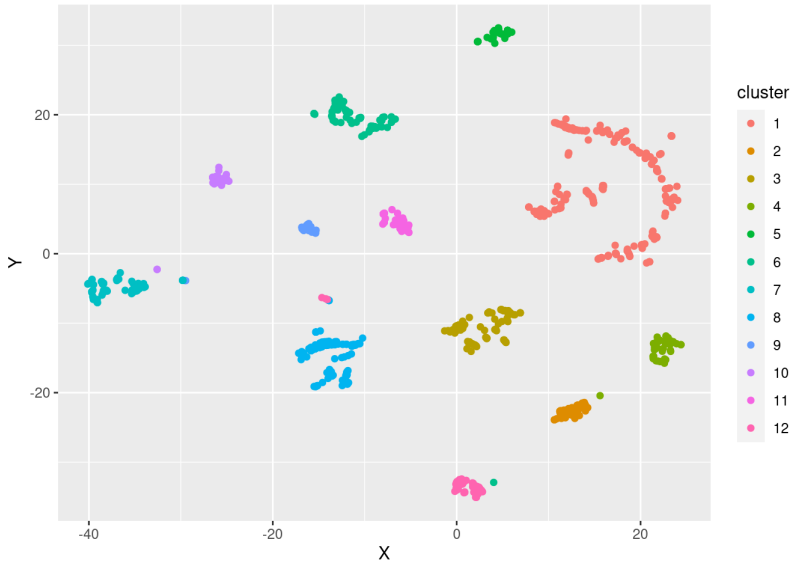
De estos grupos se puede resumir sus principales características basadas en el medoide.

```
people_red[pam_fit$medoids, ]
```

```
##      Survived Pclass    Sex Age SibSp Parch    Fare Embarked
## 794   Fallece      3   male  25     0     0  7.8958        S
## 218 Sobrevive      1 female  32     0     0 76.2917        C
## 142 Sobrevive      3 female  24     1     0 15.8500        S
## 230 Sobrevive      1 female  35     1     0 83.4750        S
## 510 Sobrevive      3   male  29     0     0  7.7500        Q
## 339   Fallece      1   male  45     0     0 35.5000        S
## 211 Sobrevive      2 female  35     0     0 21.0000        S
## 178   Fallece      2   male  30     0     0 13.0000        S
## 702   Fallece      3 female  18     0     1 14.4542        C
## 359 Sobrevive      3 female  28     0     0  7.8792        Q
## 693   Fallece      3   male  25     0     0  7.2250        C
## 583   Fallece      1   male  36     0     0 40.1250        C
```

La siguiente gráfica permite ver de manera simple la separación realizada por el algoritmo para 12 grupos.

```
library(Rtsne) # for t-SNE plot
tsne_obj <- Rtsne(gower_dist, is_distance = TRUE)
tsne_data <- tsne_obj$Y %>%
  data.frame() %>%
  setNames(c("X", "Y")) %>%
  mutate(cluster = factor(pam_fit$clustering),
         name = people_red$Survived)
ggplot(aes(x = X, y = Y), data = tsne_data) +
  geom_point(aes(color = cluster))
```



Como no se pretende utilizar el modelo para predecir los resultados, no será necesario ningún procedimiento adicional.

4.3.2 Arboles de decision

A continuación, se generara un modelo para predecir los supervivientes mediante arboles de decisión.

Se genera el arbol de decisión con ctree y Random forest. Para entrenar ambos arboles se realizará por validación cruzada con el metodo k-folds haciendo un conjunto de 10 submuestras.

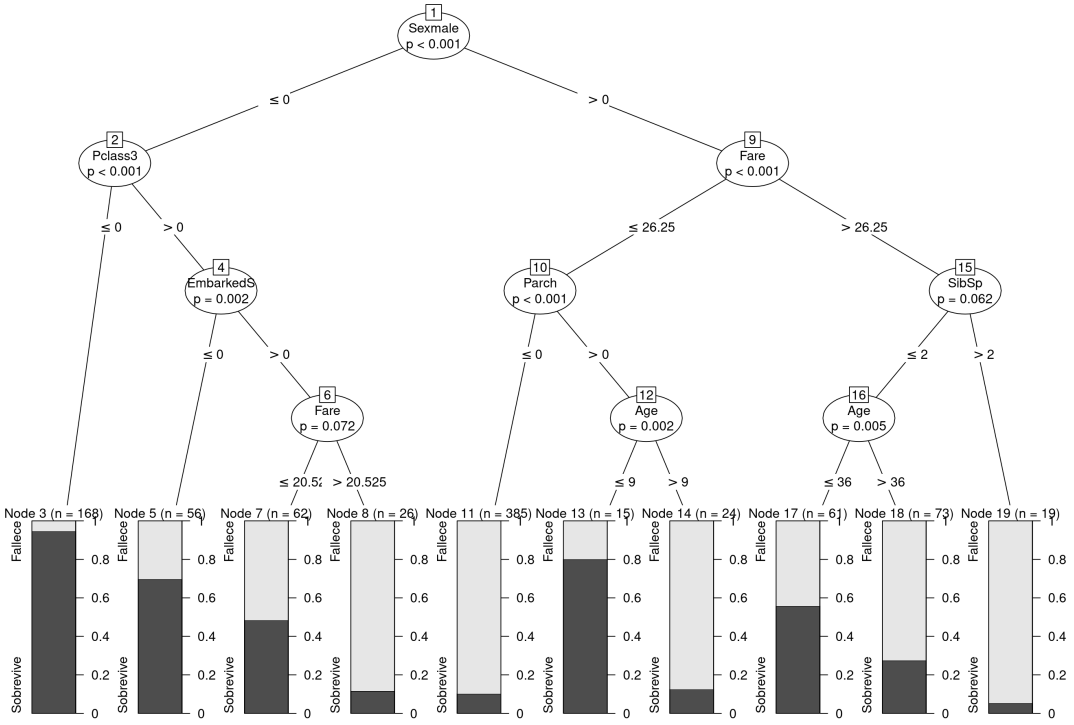
```
library(caret)
library(party)
library(partykit)
train_control <- trainControl(method = "cv", number = 10, savePredictions=TRUE)

ctree_ <- train(Survived ~ ., data = people_red, method = "ctree", trControl=train_control)

forest.mod <- train(Survived ~ ., data = people_red, method = "rf", trControl=train_control)
```

Se muestra el arbol de decisión generado con el algoritmo ctree.

```
plot(ctree_$finalModel)
```



Como resultado, se observa que los atributos que más peso tienen en la clasificación son el sexo, la clase y el precio del ticket.

A continuación, se obtienen sus métricas.

```
confusionMatrix(ctree_$pred$pred, ctree_$pred$obs)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Fallece Sobrevive
##   Fallece      1494       363
##   Sobrevive     153       657
##
##               Accuracy : 0.8065
##                 95% CI : (0.791, 0.8214)
##    No Information Rate : 0.6175
##    P-Value [Acc > NIR] : < 2.2e-16
##
##                  Kappa : 0.5737
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9071
##             Specificity : 0.6441
##          Pos Pred Value : 0.8045
##          Neg Pred Value : 0.8111
##              Prevalence : 0.6175
##          Detection Rate : 0.5602
##    Detection Prevalence : 0.6963
##       Balanced Accuracy : 0.7756
##
##        'Positive' Class : Fallece
##
```

```
confusionMatrix(forest.mod$pred$pred, forest.mod$pred$obs)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  Fallece Sobrevive
##   Fallece      1481       301
##   Sobrevive     166       719
##
##                Accuracy : 0.8249
##                  95% CI : (0.8099, 0.8391)
##     No Information Rate : 0.6175
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6197
##
##  Mcnemar's Test P-Value : 5.618e-10
##
##             Sensitivity : 0.8992
##             Specificity : 0.7049
##          Pos Pred Value : 0.8311
##          Neg Pred Value : 0.8124
##              Prevalence : 0.6175
##          Detection Rate : 0.5553
##    Detection Prevalence : 0.6682
##       Balanced Accuracy : 0.8021
##
##        'Positive' Class : Fallece
##
```

La precisión global del modelo obtenido por random forest es mejor, aunque el modelo de ctree permite determinar los fallecidos con un poco más de precisión.

4.3.3 Regresión Logística

En este caso, se realiza una regresión logística para predecir los datos. Se aplica una regresión logística utilizando todas las variables, tanto cuantitativas como cualitativas. Se realiza validación cruzada K-fold sobre el modelo de regresión con 10 iteraciones, como se ha realizado en el apartado anterior. Para saber que covariables insertar al modelo, se debe observar si hay variables de confusión, esto es, variables que al tomarlas en cuenta en el modelo, cambien significativamente el factor de otras.

```
logist <- train(Survived ~ .,
                data = people_red,
                trControl = train_control,
                method = "glm",
                family=binomial())

summary(logist, maxsum=1)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8401  -0.6271  -0.4044   0.6142   2.4458
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.728663   0.531452   7.016 2.28e-12 ***
## Pclass2     -0.755175   0.332207  -2.273 0.023014 *
## Pclass3     -1.960698   0.350064  -5.601 2.13e-08 ***
## Sexmale     -2.673995   0.202518 -13.204  < 2e-16 ***
## Age         -0.036576   0.007795  -4.692 2.70e-06 ***
## SibSp       -0.383427   0.113718  -3.372 0.000747 ***
## Parch       -0.129656   0.121259  -1.069 0.284957
## Fare         0.009429   0.005383   1.751 0.079867 .
## EmbarkedQ    0.074674   0.382580   0.195 0.845249
## EmbarkedS   -0.426656   0.238794  -1.787 0.073984 .
## ---
## 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:  784.01  on 879  degrees of freedom
## AIC: 804.01
##
## Number of Fisher Scoring iterations: 5
```

Se observa como las variables de PClass, Sex, Age y SibSp son significativas ya que el p-valor proporcionado por el estadístico de Wald es inferior a 0,05. De estos regresores, vamos a ver cuál tiene mayor impacto. Para ello utilizamos el criterio de información de Akaike, es decir, se observa que valor tiene el AIC con cada una de las variables predictoras por separado.

```
#Pclass
summary(train(Survived ~ Pclass,
                data = people_red,
                trControl = train_control,
                method = "glm",
                family=binomial(),
                metric="Accuracy"))
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4028  -0.7450  -0.7450   0.9676   1.6836
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5158     0.1413   3.651 0.000261 ***
## Pclass2      -0.6246     0.2044  -3.056 0.002241 **
## Pclass3      -1.6556     0.1762  -9.395  < 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: 1182.8  on 888  degrees of freedom
## Residual deviance: 1081.2  on 886  degrees of freedom
## AIC: 1087.2
##
## Number of Fisher Scoring iterations: 4
```

```
#Sex
summary(train(Survived ~ Sex,
                data = people_red,
                trControl = train_control,
                method = "glm",
                family=binomial(),
                metric="Accuracy"))
```

```
##
## Call:
## NULL
##
## 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 ***
## Sexmale      -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
```

```
#Age
summary(train(Survived ~ Age,
                data = people_red,
                trControl = train_control,
                method = "glm",
                family=binomial(),
                metric="Accuracy"))
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0625  -0.9943  -0.9475   1.3691   1.5479
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.269204   0.163069  -1.651   0.0988 .
## Age         -0.007120   0.005037  -1.413   0.1575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.8  on 888  degrees of freedom
## Residual deviance: 1180.8  on 887  degrees of freedom
## AIC: 1184.8
##
## Number of Fisher Scoring iterations: 4
```

```
#SibSp
summary(train(Survived ~ SibSp,
               data = people_red,
               trControl = train_control,
               method = "glm",
               family=binomial(),
               metric="Accuracy"))
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9951  -0.9951  -0.9694   1.3714   1.4896
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.44525    0.07643  -5.826 5.69e-09 ***
## SibSp       -0.06604    0.06532  -1.011    0.312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.8  on 888  degrees of freedom
## Residual deviance: 1181.8  on 887  degrees of freedom
## AIC: 1185.8
##
## Number of Fisher Scoring iterations: 4
```

Si se compara los valores de AIC de cada modelo, se observa que el de Sex es el mas bajo luego esa sera la variable más representativa. Por último, se crea la matriz de confusión para ver cuál es el valor predictivo de nuestro modelo.

```
logist <- train(Survived ~ .,
         data = people_red,
         trControl = train_control,
         method = "glm",
         family=binomial())

confusionMatrix(logist$pred$pred, logist$pred$obs)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Fallece Sobrevive
##   Fallece       477       103
##   Sobrevive      72       237
##
##                Accuracy : 0.8031
##                  95% CI : (0.7755, 0.8288)
##     No Information Rate : 0.6175
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.5759
##
##  Mcnemar's Test P-Value : 0.02334
##
##             Sensitivity : 0.8689
##             Specificity : 0.6971
##          Pos Pred Value : 0.8224
##          Neg Pred Value : 0.7670
##              Prevalence : 0.6175
##          Detection Rate : 0.5366
##    Detection Prevalence : 0.6524
##       Balanced Accuracy : 0.7830
##
##        'Positive' Class : Fallece
##
```

Se observa que el modelo tiene una precisión decente aunque mejorable. Sobre todo, el modelo identifica bien a los fallecidos.

4.3.4 Contrastes

A continuación, se realizarán algunos contrastes de hipótesis entre los grupos creados anteriormente.

En primer lugar, se realiza un test de proporción para definir si es diferente el numero de fallecidos en función del sexo.

```
# Test proporcion muertes por genero
n1<-nrow(people_red.male)
n2<-nrow(people_red.female)
p1 <- sum(people_red.male$Survived=="Fallece")/nrow(people_red.male);
p2 <- sum(people_red.female$Survived=="Fallece")/nrow(people_red.female);
success<-c( p1*n1, p2*n2)
nn<-c(n1,n2)
prop.test(success, nn, alternative="two.sided", correct=FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  success out of nn
## X-squared = 260.76, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.4932809 0.6096720
## sample estimates:
##    prop 1    prop 2
## 0.8110919 0.2596154
```

El test permite determinar que la proporcion de muertes según el sexo no es la misma.

Continuando con el mismo tipo de análisis, se analizará si hay similitud entre el numero de fallecimientos entre la clase primera y la tercera.

```
# Test proporcion muertes por clase
n1<-nrow(people_red.first)
n2<-nrow(people_red.third)
p1 <- sum(people_red.first$Survived=="Fallece")/nrow(people_red.first);
p2 <- sum(people_red.third$Survived=="Fallece")/nrow(people_red.third);
success<-c( p1*n1, p2*n2)
nn<-c(n1,n2)
prop.test(success, nn, alternative="two.sided", correct=FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  success out of nn
## X-squared = 95.422, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.4588960 -0.3087154
## sample estimates:
##    prop 1    prop 2
## 0.3738318 0.7576375
```

Se estima de esta manera, que la proporcion de muertes según clase tampoco es la misma, es decir, tiene un cierto efecto la clase a la que se pertenece para determinar la supervivencia.

Para el siguiente test, se analiza en primer lugar si se produce homocedasticidad entre las variables.

```
var.test(people_red.first$Age, people_red.third$Age)
```

```
##
##  F test to compare two variances
##
## data:  people_red.first$Age and people_red.third$Age
## F = 1.4182, num df = 213, denom df = 490, p-value = 0.002024
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.134954 1.790572
## sample estimates:
## ratio of variances
##           1.418186
```

Como las varianzas no son iguales debido a un valor p inferior a 0.05, se realiza el test para heterocedasticidad para determinar si la media de edad de la 1a clase es mayor que la de la 3a.

```
# Test media edad en funcion de clase
t.test(people_red.first$Age, people_red.third$Age,alternative="greater", var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  people_red.first$Age and people_red.third$Age
## t = 11.618, df = 349.68, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  10.96389      Inf
## sample estimates:
## mean of x mean of y
##  38.42720  25.64953
```

Dado un valor p tan pequeño, no se confirma la hipotesis nula y se estima entonces que la media de edad de la primera clase era más elevada que la de la tercera.

Nuevamente se analiza si se produce homocedasticidad entre las variables antes de realizar el analisis de edad entre fallecidos.

```
var.test(people_red.fallecidos$Age, people_red.supervivientes$Age)
```

```
##
##  F test to compare two variances
##
## data:  people_red.fallecidos$Age and people_red.supervivientes$Age
## F = 0.89778, num df = 548, denom df = 339, p-value = 0.2649
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7394686 1.0852144
## sample estimates:
## ratio of variances
##          0.8977827
```

Se observa que la varianzas son iguales. Se procede a realizar el test para determinar si la media de la edad de los fallecidos es mayor que la de los supervivientes.

```
# Test media edad en funcion de fallecimiento o supervivencia
t.test(people_red.fallecidos$Age, people_red.supervivientes$Age,alternative="greater", var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  people_red.fallecidos$Age and people_red.supervivientes$Age
## t = 1.4151, df = 887, p-value = 0.07869
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -0.220537       Inf
## sample estimates:
## mean of x mean of y
##  30.16667  28.81815
```

Según los resultados del test, la edad de los fallecidos no era mayor que la de los supervivientes, al no rechazar la hipotesis nula por un valor p mayor que el valor de significancia 0.05.

Para el ultimo test, se analiza la homocedasticidad tambien.

```
var.test(people_red.first$Fare, people_red.third$Fare)
```

```
##
##  F test to compare two variances
##
## data:  people_red.first$Fare and people_red.third$Fare
## F = 14.822, num df = 213, denom df = 490, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  11.86147 18.71338
## sample estimates:
## ratio of variances
##           14.82155
```

Dada la heterocedasticidad entre las muestras, se analiza si la media del precio del ticket de 1a clase es significativamente mayor que el de la 3a.

```
# Test media precio por clase
t.test(people_red.first$Fare, people_red.third$Fare,alternative="greater", var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  people_red.first$Fare and people_red.third$Fare
## t = 18.858, df = 225.63, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  44.42146      Inf
## sample estimates:
## mean of x mean of y
##  61.55281  12.86745
```

Este ultimo test, rechaza la hipotesis nula de las tarifas, aceptando que los precios de primera clase son más caros, al disponer de un valor p tan pequeño, tal como se esperaba.

```
write.csv(people_red,"titanic_clean.csv", row.names = FALSE)
```

5 Conclusiones

Los modelos obtenidos permiten obtener gran cantidad de información sobre los pasajeros y su supervivencia.

Por una parte, se ha observado que mediante clustering existen 12 perfiles de pasajeros, de los cuales los hombres suelen fallecer en la mayoría, mientras que los perfiles de mujeres suelen sobrevivir.

Los modelos destinados a predición poseen una precisión similar, siendo el de Random Forest el mejor, con un 82% aproximadamente. Todos los modelos, muestran una sensibilidad más elevada que la especificidad, con lo que es más facil detectar a los que fallecen que a los que sobreviven. El modelo de regresión logística ha mostrado, de manera acorde con los otros modelos y con los contrastes de hipótesis realizados sobre los diferentes grupos, que las variables más influyentes en la supervivencia son el sexo y la clase del pasajero.

6 Tabla de contribuciones

Contribuciones	Firma
Investigación Previa	E.C.G. M.R.L
Redacción de las respuestas	E.C.G. M.R.L
Desarrollo código	E.C.G. M.R.L