Universitat Oberta de Catalunya
Máster Universitario en Ciencia de Datos
Tipología y Ciclo de vida de los Datos

PRAC 2: Limpieza y validación de datos

Profesor(a): Mireia Calvo González
Estudiante: Carlos Alfonso Cuervo Rodríguez
Fecha: 10 de junio de 2018

1: Descripción de la base de datos:

Para esta práctica se ha hecho uso de la base de datos "Titanic", suministrada por la página de competencia de ciencia de datos Kaggle, para machine learning. A continuación, el link de la fuente de información: https://www.kaggle.com/c/titanic/data

Por otra parte, en este ejercicio se pretende crear un modelo predictivo por medio de árboles de decisión y usando el software R, que indique si un tripulante hubiera sobrevivido o no dado sus atributos o características, entre ellas su edad y la clase en la que viajaba.

Kaggle suministra tres bases de datos sobre titanic. Una de entrenamiento, una de test y otra de datos se gender_submission, donde supone que todas las mujeres han sobrevivido. De todas ellas sólo haremos uso de la denominada "train" (entrenamiento), ya que es la única que nos sirve para crear un modelo de árboles de decisión, porque la base de test no indica si el pasajero sobrevive o no, así que no se puede entrenar ni testear un modelo si no se tiene esa información a priori. Y el gender submission, pues ya está sesgado considerando que todas y sólo las mujeres se van a salvar.

Para la base de datos de titanic elegida (train) se tienen los siguientes atributos:

- **PassengerId:** será el id para identificar al pasajero.
- **Survived:** Indica si el pasajero sobrevivió o no. 1 para si, 0 para no.
- **Pclass:** Indica la clase en la que viajaba el pasajero: 1 para primera clase, 2 para segunda clase y 3 para tercera clase.
- **Name:** Nombre del pasajero.
- **Sex:** Sexo del pasajero. Entre masculino y femenino.
- **Age:** Edad aproximada de los pasajeros. Va desde 1 hasta los 70.5 años.
- **SibSp:** Indica el número de hermanos o conyugues del pasajero que se encontraban a bordo.
- **Parch:** Indica el número de padres o niños del pasajero que se encontraban a bordo.
- **Ticket:** Número de billete del pasajero.
- **Fare:** Tarifa del pasajero.
- **Cabin:** Cabina donde se encontraba el pasajero.
- **Embarked:** Puerto donde embarcó el pasajero. C = Cherbourg, Q = Queenstown, S = Southampton.

Cómo se mencionó anteriormente, la idea es hacer preparar los datos para hacer uso de un modelo de clasificación por medio de árboles de decisión que permitan: determinar cuáles variables son

más influyentes a la hora de evaluar la probabilidad de sobrevivencia de un pasajero y segundo, determinar si estas variables más influyentes permiten crear un modelo lo suficientemente robusto para concluir a ciencia cierta la probabilidad de sobrevivencia de una persona en un tipo de accidentes así. Ahora bien, yéndonos al contexto, un resultado de estos podría ayudar a prevenir o al menos mitigar la severidad de un accidente así.

Integración y selección de los datos de interés a analizar:

Para este ejercicio no hay data integration, ya que la fuente de información es única por lo que no es necesario un merge o una homologación de campos. Sin embargo, si se hace un ejercicio de data reduction porque no se hará uso de todas las variables. Sólo se utilizarán (aparte de survived, por obvias razones), "Pclass", "Sex", "Age", "Sibsp" y "Parch". La razón es que se quiere comprobar si las condiciones económicas, de sexo, edad y núcleo familiar influyeron en la probabilidad de sobrevivencia. Claramente estas variables son a priori, ya el modelo determinará si son o no significativas.

Limpieza y preprocesamiento de los datos:

Ya que el objetivo es crear un modelo supervisado de clasificación por medio de árboles de decisión (por me método C50), lo que se debe hacer primeramente es transformar los datos de tal manera que las variables numéricas se conviertan en categóricas. La variable "Pclass" ya tiene sólo tres categorías, así como "sex" sólo tiene dos. La idea es hacer lo mismo con "Age", "SibSp" y "Parch". Sin embargo, antes de esto se debe hacer un análisis de los datos para procesar datos vacíos, nulos y ceros, así como identificar outliers.

Ceros y valores vacíos:
Echando un vistazo a los datos de "sex", se observa que no hay valores ceros ni nulos, pero si hay 177 vacíos que habrá que tratar. Sin embargo, antes de cargar la data a R, fue notorio que todos los pasajeros con título de "Master" son niños, ya que la edad mínima es 1, la máxima es 12 y la media es de 5,2 años. Como el propósito es convertir esta variable en categórica y no la edad específica se puede concluir que todos aquellos pasajeros que tengan el título "Master" y tengan el valor de edad vacío se podrían clasificar como "niño". Así que se hará uso de un método manual [1] (ya que no son muchos registros) para llenar estos vacíos con la media de esta población que es 5,2 años. De esta manera quedamos con 173 datos vacíos que se trataran con otro método.

Se muestra a continuación lo procesado hasta el momento:

```
1
2  #Traer los paquetes requeridos:
3  library(readxl)
4  library(C50)
5  library(VIM)
6
7  #Leer la data necesaria:
8  titanic <- read_excel("C:/Users/cuervoca/Desktop/MasterDataScience/Semestre 2/TipologiaDatos/PRA's/PRA 2/Data/train.xlsx",
9        col_types = c("numeric", "numeric", "numeric", "text", "text", "numeric", "numeric","numeric", "text", "text","text","text"))
10
11 #Quedarse con los campos deseados:
12 titanic <- titanic[c(2,3,5:8)]
13
14 #conocer los datos
15 summary(titanic)
16
```

Se han cargado las librerías para cargar el Excel y para usar más adelante los métodos descritos líneas arriba (C50 para el árbol de decisión y VIM para manejar missing values). Después se cargó la data, indicando la naturaleza de cada variable. Posteriormente se seleccionan sólo las variables deseadas y por último se presenta un resumen del valor de dichas variables:

```
> summary(titanic)
    Survived          Pclass          Sex                 Age             SibSp           Parch
 Min.   :0.0000   Min.   :1.000   Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:2.000   Class :character   1st Qu.:20.00   1st Qu.:0.000   1st Qu.:0.0000
 Median :0.0000   Median :3.000   Mode  :character   Median :28.00   Median :0.000   Median :0.0000
 Mean   :0.3838   Mean   :2.309                      Mean   :29.56   Mean   :0.523   Mean   :0.3816
 3rd Qu.:1.0000   3rd Qu.:3.000                      3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
 Max.   :1.0000   Max.   :3.000                      Max.   :80.00   Max.   :8.000   Max.   :6.0000
                                                     NA's   :173
```

Cómo se había mencionado, se tienen 173 datos vacíos en la variable edad, lo cual representa el 19,4% de la información, considerando que nuestra muestra total para este caso es de 891 registros. Así entonces se vuelve de suma importancia tratar de rellenar estos valores, ya que si se eliminan o descartan se perdería demasiada información.

Para tratar los 173 registros perdidos, se hará uso del método KNN de la librería VIM, que permite imputar valores dados los vecinos más cercanos[1]

```
17
18  #Rellenar los valores vacios de edad:
19  titanic <- kNN(titanic, variable = c("Age"),k=10)
20
21
```

Se indica el dataset, que para este caso es “titanic”, se le indica la variable a la que se le imputaran los valores, y se le indica la cantidad de vecinos a tener en cuenta, en este caso se escogieron 10. De esta manera si se vuelve a obtener el resumen se tiene lo siguiente:

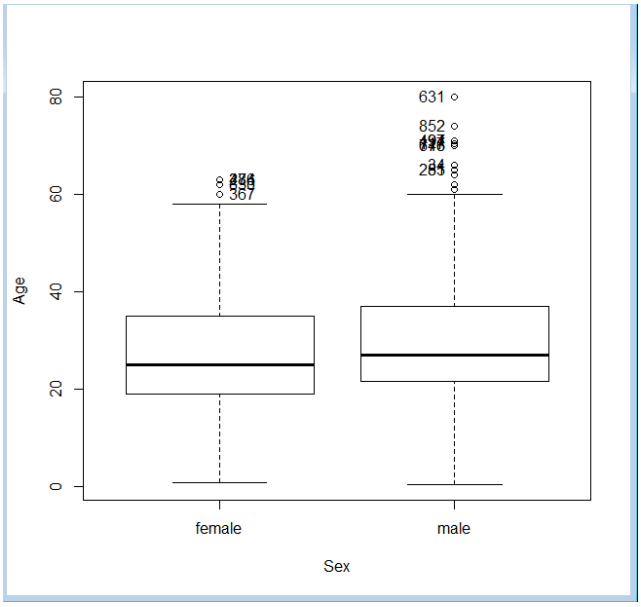
```
> summary(titanic)
    Survived          Pclass          Sex                 Age             SibSp           Parch          Age_imp
 Min.   :0.0000   Min.   :1.000   Length:891         Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Mode :logical
 1st Qu.:0.0000   1st Qu.:2.000   Class :character   1st Qu.:21.00   1st Qu.:0.000   1st Qu.:0.0000   FALSE:718
 Median :0.0000   Median :3.000   Mode  :character   Median :27.00   Median :0.000   Median :0.0000   TRUE :173
 Mean   :0.3838   Mean   :2.309                      Mean   :28.90   Mean   :0.523   Mean   :0.3816
 3rd Qu.:1.0000   3rd Qu.:3.000                      3rd Qu.:36.00   3rd Qu.:1.000   3rd Qu.:0.0000
 Max.   :1.0000   Max.   :3.000                      Max.   :80.00   Max.   :8.000   Max.   :6.0000
```

Así entonces la variable edad ya no tiene valores vacíos y además se crea la nueva variable “Age_imp” que indica con un true o false, si el valor del registro fue imputado o no. En efecto indica que 173 valores fueron imputados, lo cual era lo que se esperaba. Esta última variable se borrará posteriormente.

Por otra parte, del anterior resumen se puede ver que las variables “SibSp” y “Parch” toman valores de cero, pero esto no representa un problema o un error, porque en efecto la persona podría estar viajando sola o en el caso de los niños estar viajando con una niñera por lo que el valor de Parch es cero. Así que no se les hará ningún tratamiento especial a estas variables por ahora.

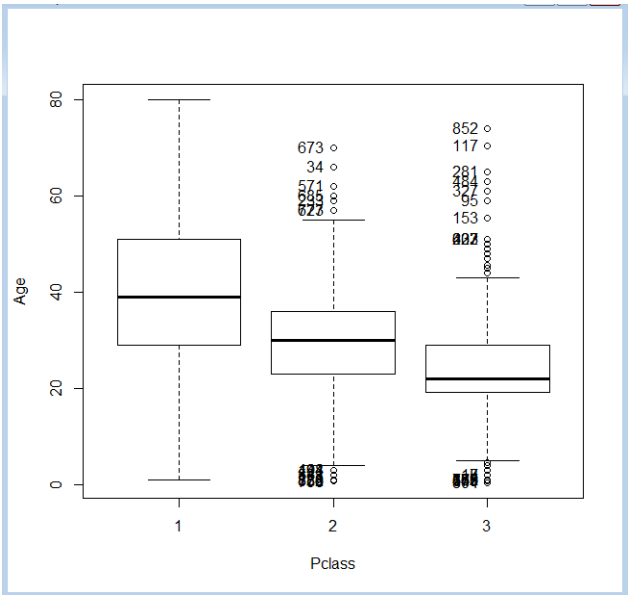
[1] La documentación CRAN se puede encontrar en: https://cran.r-project.org/web/packages/VIM/VIM.pdf

Valores extremos:
Ahora bien, como ya se mencionó los valores de las variables “SibSp” y “Parch” pueden ir des cero hasta valores de 6 y 8, así que no hay problemas con sus valores, por lo cual nos queda la variable “Age” para analizarla por sexo y por clase:



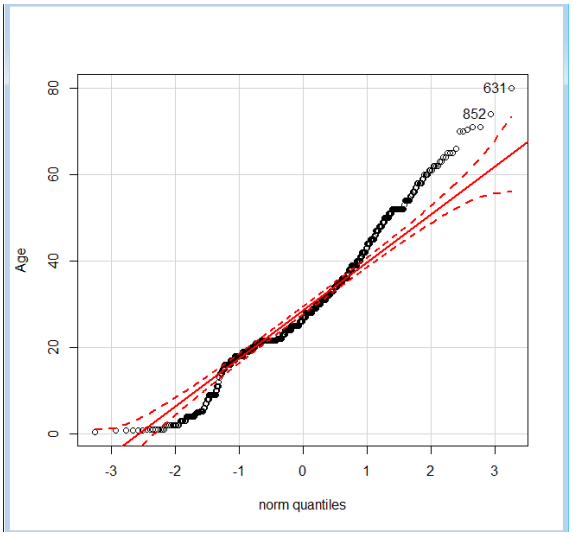
En efecto se evidencia que tanto para hombre como para mujeres hay valores de edad que están muy por encima del promedio y de la desviación, es decir por encima del cuarto cuartil. Sin embargo, se puede observar que son valores de edad factibles, es decir, la mayor edad es de 80 años, lo que se puede considerar posible. No se observan valores de una edad de 200 años, por ejemplo, así que se decide seguir trabajando con ellos y no sacarlos de la muestra [3]. Por otra parte, esta gráfica nos indica que el promedio de edad entre hombre y mujeres realmente era muy parecido y que presenta diferencias mínimas.

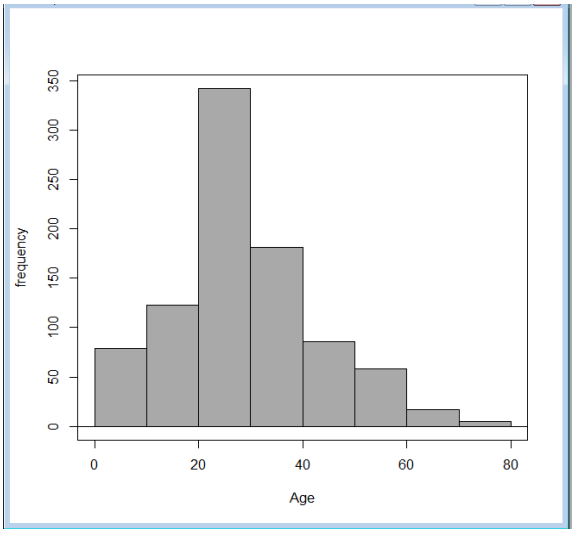
Ahora se analiza el comportamiento de la edad, dada la clase en la que viajaban los pasajeros. Primeramente, se observan bastantes outliers en la segunda y en la tercera clase, pero como se mencionaba anteriormente, son valores de edades factibles. Por otra parte, de esta gráfica se puede concluir que las personas de primera clase eran las personas con el promedio de edad más alta, sin decir esto que no hubiera personas mayores en las demás clases, por esta razón los outliers. Se puede ver que las clases dos y tres tenía personas muy mayores y así mismo, niños muy pequeños. De esto se puede concluir que las personas de más bajos recursos son los que más número de niño tenía y de muy baja edad. Esta información nos servirá para analizar más adelante los resultados del modelo.



Análisis de los datos:
Como se había mencionado la idea es hacer un modelo de minería de datos, haciendo uso de árboles de clasificación. Como se sabe para estos modelos no se deben asumir supuestos de normalidad ni de homocedasticiadad. Sin embargo, se realizan los análisis sobre la variable “Age” para tener más conocimiento de los datos usados.

Se empieza por el análisis de normalidad, primeramente, con un Q-Q plot, seguido de un histograma de frecuencia:





Como se pude observar en el Q-Q plot el segundo y el tercer cuartil se alejan demasiado de lo que debería ser una distribución normal, lo que a primera vista estaría indicando que no se distribuye de forma normal. Esto se refuerza al ver el histograma, donde se ve una buena concentración entre los 20 y los 30 años, lo que hace que el primer y segundo cuartil tenga la mayor concentración.

Sin embargo, se realiza la prueba de normalidad de Shapiro- Wilk para comprobar estas primeras hipótesis:

```
> normalityTest(~Age, test="shapiro.test", data=titanic)

        Shapiro-Wilk normality test

data:  Age
W = 0.97034, p-value = 1.702e-12
```

En efecto el P-Valor es bastante pequeño, considerando un Alpha de 5%. Por lo cual no hay suficiente evidencia estadística para no rechazar la hipótesis nula. Por ende, se concluye que estos datos no siguen una distribución normal.

Ahora se realiza una prueba entre los hombres y mujeres, para determinar si la varianza de sus edades es homogénea o no:

```
> bartlett.test(Age ~ Sex, data=titanic)

        Bartlett test of homogeneity of variances

data:  Age by Sex
Bartlett's K-squared = 0.89328, df = 1, p-value = 0.3446
```

Ya que el p-valor es superior al Alpha estimado del 5%, se concluye que las varianzas no presentan diferencias significativas.

Sin embargo, como se mencionaba anteriormente, estas condiciones estadísticas no tendrían por qué afectar un modelo de machine learning, así que se procede con el modelo descrito.

Aplicación del modelo:

Para esto lo primero que se va a hacer es volver a la variable "Survived" un factor, para que el modelo lo tome como categórica, por lo que también se cambiará su nomenclatura. Ahora el pasajero que sobrevive será identificado con una "S" y el que muere con una "M":

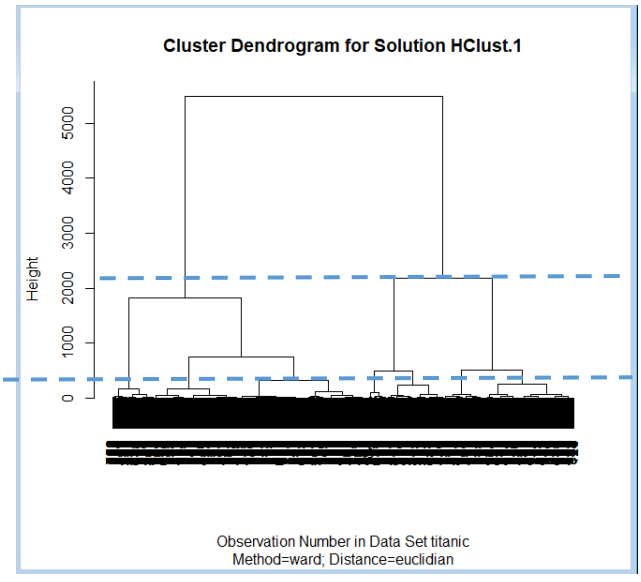
```
35
36  #Aplicación del modelo:
37  #Obtener vector de la dimensión de las filas
38  vive<-vector(length = dim(titanic)[1])
39  #cambiar los nombre de variables
40  vive[titanic$Survived== 0]<-"M"
41  vive[titanic$Survived== 1]<-"S"
42  #Ahora se incorpora esta información a los datos, reemplazando los valores y convirtiendolo en un factor
43  titanic$Survived<-factor(vive)
44
```

Ahora se realiza una discretización de la variable edad, para poder así tener rangos de edad y no edades puntuales, para esto se hará uso del método jerárquico y el de K-means:

Primeramente, se hace uso de un modelo no supervisado para analizar a priori cual debería ser el número adecuado de grupos que se debería tener para la variable "Age". Así entonces se hace uso del modelo jerárquico y se obtiene lo siguiente:

```
45
46  #Discretizar variables:
47  #Analisis jerarquico de edad:
48  HClust.1 <- hclust(dist(model.matrix(~-1 + Age, titanic)) , method= "ward")
49  plot(HClust.1, main= "Cluster Dendrogram for Solution HClust.1",
50               xlab= "Observation Number in Data Set titanic", sub="Method=ward; Distance=euclidian")
51
```

Con el presente dendograma se podría llegar a pensar en realizar dos grandes grupos, donde seguramente se haga la agrupación entre adultos y niños, o se podría pensar en 4 rangos de edad. Se realizarán las pruebas y se evaluará la pertinencia del número de grupos.



Para el caso de dos rangos de edad (K=2):

Se espera que haya alta cohesión entre los miembros de cada grupo, es decir que la suma de cuadrados del error de la distancia de cada punto al centroide del grupo sea la más baja posible, y que la suma de cuadrados del error de la distancia entre los centroides de los grupos (también se puede entre los miembros más cercanos o los más lejanos) sea lo más alta posible [2].

Indicadores:

Haciendo uso de las sumas del cuadrado del error, se espera obtener los siguientes indicadores para medir la cohesión y separación de los diferentes modelos [2]:

$$Ball\ \&\ Hall\ \frac{SSW}{K}$$

$$Caliski\&Harabasz\ \frac{\left(\frac{SSB}{k-1}\right)}{\left(\frac{SSW}{n-k}\right)}$$

Donde:

SSW= suma de cuadrados del error dentro de los grupos.
SSB = suma de cuadrados del error total entre los grupos.
n = número total de datos.
k= número de clusters.

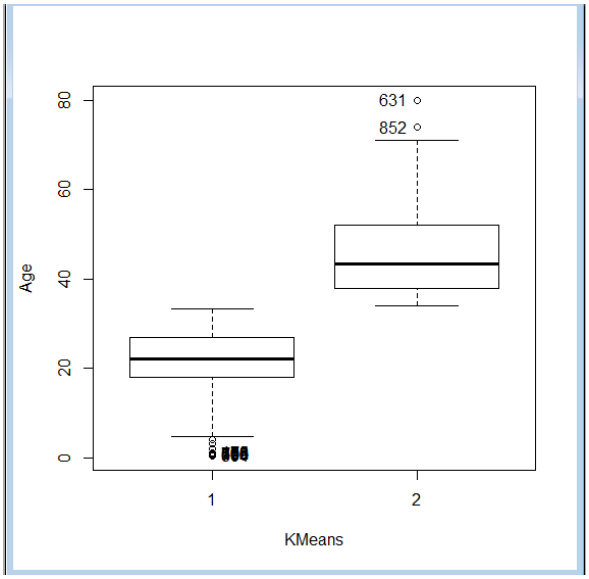
Datos obtenidos:

```
> .cluster <-  KMeans(model.matrix(~-1 + Age, titanic), centers = 2, iter.max = 10, num.seeds = 10)
> .cluster$size # Cluster Sizes
[1] 611 280
> .cluster$centers # Cluster Centroids
       Age
1 21.38211
2 45.30000
> .cluster$withinss # Within Cluster Sum of Squares
[1] 39092.58 24421.80
> .cluster$tot.withinss # Total Within Sum of Squares
[1] 63514.38
> .cluster$betweenss # Between Cluster Sum of Squares
[1] 109841.7
```

Con los datos obtenidos los indicadores serán los siguientes:

B&H = (63514.38/2) = **31,757.19**
C&H = (109841.7/1) / (63514.38/889) = **1,537.45**

Ahora se observa un diagrama de cajas con la partición anteriormente hecha. Se observa que evidentemente hay una diferencia significativa en rangos de edad: una desde los 0 hasta los 38 años aproximadamente y otra desde los 38 hasta los 80 años.



Para el caso de cuatro rangos de edad (K=4):

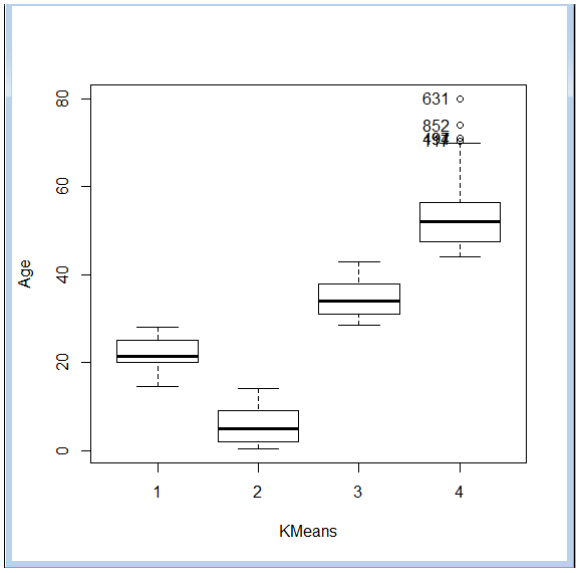
Para este caso se obtuvo el siguiente resultado:

```
> .cluster <-  KMeans(model.matrix(~-1 + Age, titanic), centers = 4, iter.max = 10, num.seeds = 10)
> .cluster$size # Cluster Sizes
[1] 404  92 255 140
> .cluster$centers # Cluster Centroids
       Age
1 22.21163
2  5.84750
3 34.63922
4 52.88571
> .cluster$withinss # Within Cluster Sum of Squares
[1] 4556.655 1429.875 4178.933 7284.171
> .cluster$tot.withinss # Total Within Sum of Squares
[1] 17449.63
> .cluster$betweenss # Between Cluster Sum of Squares
[1] 155906.4
```

De esta manera los indicadores son los siguientes:

B&H = (17449.63/4) = **4,362.4**
C&H = (155906.4/3) / (17449.63/887) = **2,641.68**

Ahora se observa un diagrama de cajas con la partición anteriormente hecha de 4 rangos. Se observa que ahora hay una mejor división entre niños, pasajeros jóvenes, adultos y adultos mayores.



Después de comparar los indicadores donde se observa que de 2 a 4 grupos se pasa de un error dentro de grupos de 31757 a 4362, pues esto significa que hay mayor cohesión entre los miembros de los grupos. Y se pasa de un error entre grupos de 1537 a 2641, lo que significa que hay mayor diferencia entre los centroides de los grupos. Por esta razón se hace una discretización de los rangos de edad en 4 grupos.

Sin embargo, esta discretización se ve en el resultado como la variable Kmeans que toma el valor de 1 a 4 dependiendo del grupo. Sin embargo, cada vez que se corre el modelo, no siempre el grupo con menor edad será el 1, ni el de mayor edad será el 4, ya que esto se da de manera aleatoria, por esta razón lo que se hará es tratar de poner las categorías según los rangos dados por el boxplot, para tener una aproximación y asegurar que sin importar la corrida siempre de lo mismo.

```
59
60  #Discretización de la edad:
61  edad<-vector(length = dim(titanic)[1])
62  edad[titanic$Age >= 0 & titanic$Age < 15]<-"menor"
63  edad[titanic$Age >= 15 & titanic$Age < 29]<-"joven"
64  edad[titanic$Age >= 29 & titanic$Age < 45]<-"adulto"
65  edad[titanic$Age >= 45]<-"adulto_mayor"
66  titanic$Age<-factor(edad)
67
```

Discretización de "SibSp" y "Parch":

Ahora se procede a discretizar las otras dos variables que dan información de las condiciones en las que viajaba el pasajero. Se podrían unir estas variables y tratar de crear otra donde se indique si el tamaño de la familia era pequeña, mediana o grande, pero haciendo se pierde información valiosa, como por ejemplo si tenía o no hijos, si tenía o no pareja y si tenía o no padres o si viajaba completamente solo.

Así entonces se hace una discretización arbitraria donde:

SibSp	Categoría
0	solo
1	en_pareja
2,3	pocos_hermanos
4,5,6,7,8	muchos_hermanos

Parch	Categoría
0	sin_padres
1,2	con_padres
3,4,5,6	con_hijos

Se asumirá que si "SibSp" es cero es porque viajaba sólo el pasajero, pero si es 1 viajaba con una pareja, si era 2 o 3, es porque tenía hermanos y eran pocos, pero si es mayor significa que tenía hermanos y eran varios. Así mismo si "Parch" es cero, es porque no viajaba con padres ni hijos, si es 1 o 2, es porque viajaba con padres (aunque también cabe la probabilidad de hijos, pero se hace la suposición) y si era mayor, es porque viajaba con hijos.

Dentro del código se hace de la siguiente manera:

```
60
61  #Discretización del resto de variables:
62  conyugues<-vector(length = dim(titanic)[1])
63  conyugues[titanic$SibSp== 0]<-"solo"
64  conyugues[titanic$SibSp== 1]<-"pareja"
65  conyugues[titanic$SibSp== 2|titanic$SibSp== 3]<-"pocos_hermanos"
66  conyugues[titanic$SibSp== 4|titanic$SibSp== 5|titanic$SibSp== 8]<-"muchos_hermanos"
67  titanic$SibSp<-factor(conyugues)
```

Para el caso de Parch:

```
69
70  #Discretización de Parch:
71  hijos<-vector(length = dim(titanic)[1])
72  hijos[titanic$Parch== 0]<-"sin_padres"
73  hijos[titanic$Parch== 1|titanic$Parch== 2]<-"con_padres"
74  hijos[titanic$Parch >= 3]<-"con_hijos"
75  titanic$SibSp<-factor(hijos)
76
```

Ahora si se hace un summary para ver lo que se ha hecho hasta el momento se obtiene:

```
> summary(titanic)
 Survived     Pclass          Sex                     Age                 SibSp           Parch      Age_imp        KMeans
 M:549    Min.   :1.000   Length:891         adulto      :262   muchos_hermanos: 30   con_hijos : 15   Mode :logical   1: 86
 S:342    1st Qu.:2.000   Class :character   adulto_mayor:131   pareja          :209   con_padres:198   FALSE:718       2:140
          Median :3.000   Mode  :character   joven       :405   pocos_hermanos : 44   sin_padres:678   TRUE :173       3:376
          Mean   :2.309                      menor       : 93   solo            :608                                    4:289
          3rd Qu.:3.000
          Max.   :3.000
```

Se observa que aún hace falta convertir a factor la variable de clase, ya que no tiene sentido que tenga un valor numérico. Además, sexo a pesar de ser categórica no la considera como un factor por lo cual no muestra el conteo de hombres y mujeres, como si lo hace por ejemplo con la variable survived entre sobrevivientes y muertos. Por otra parte, se deben eliminar las variables que se crearon en el camino: Age_imp, y KMeans.

Convertir en factor las variables que faltan:

```
84
85
86  #Convertir a factor las variables que faltan y borrar variables no necesarias:
87  titanic$Sex<-factor(titanic$Sex)
88  titanic$Pclass<-factor(titanic$Pclass)
89  titanic <- titanic[c(1:6)]
90
```

Con esto la base de datos con la que se va a trabajar es la siguiente:

```
> summary(titanic)
 Survived Pclass      Sex                Age                   SibSp            Parch
 M:549    1:216   female:314   adulto      :262   muchos_hermanos: 30   con_hijos : 15
 S:342    2:184   male  :577   adulto_mayor:131   pareja          :209   con_padres:198
          3:491                joven       :405   pocos_hermanos : 44   sin_padres:678
                               menor       : 93   solo            :608
```

Ahora, ya estamos listos para generar el modelo de minería de datos de árboles de decisión por medio del método C50.

Implementación del modelo y resultados:

```
101
102  ## Implementación del modelo:
103  #Se crea el conjunto de datos para hacer el modelo y para realizar el test
104  set.seed(891)
105  vive.indice<-sample(1:nrow(titanic),size=(round(nrow(titanic)*0.7)))
106  #Se espera hacer la prueba con una muestra aleatoria que represente el 70% de los datos completos:
107  vive.entrenar<-titanic[vive.indice,]
108  #El 30% restante será para el test
109  vive.test<-titanic[-vive.indice,]
110
```

En la anterior implementación, lo que se hace primero es establecer los valores para que no varíen cada vez que se realiza una corrida. Acto seguido se crea una muestra de los datos que represente el 70% aproximadamente de los datos para entrenar al modelo. Estos datos de entrenar se guardan en el arreglo vive.entrenar. Ahora se crea la base de test, la cual será el complemento de la muestra obtenida en el primer paso. (En Kaggle dan una base para el test, pero como se mencionó en las primeras líneas del trabajo, esta no se tomó en cuenta porque no contiene la variable survived por lo cual no sería útil para hacer pruebas).

```
111  #Ahora se crea el modelo
112  modelotitanic<-C5.0(Survived ~ ., data = vive.entrenar)
113  summary(modelotitanic)
114  plot(modelotitanic)
115
```

Se crea el modelo con el método anteriormente mencionado y se obtiene el siguiente resultado:

```
Class specified by attribute `outcome'

Read 624 cases (6 attributes) from undefined.data

Decision tree:

Sex = female:
:...Pclass in {1,2}: S (121/8)
:   Pclass = 3:
:   :...Parch in {con_hijos,con_padres}: M (40/15)
:       Parch = sin_padres: S (58/20)
Sex = male:
:...Age in {adulto,adulto_mayor,joven}: M (374/51)
    Age = menor:
    :...SibSp in {muchos_hermanos,pocos_hermanos}: M (14/1)
        SibSp in {pareja,solo}: S (17/1)
```

```
           (a)    (b)    <-classified as
          ----   ----
           361     29    (a): class M
            67    167    (b): class S


        Attribute usage:

        100.00% Sex
         64.90% Age
         35.10% Pclass
         15.71% Parch
          4.97% SibSp


Time: 0.0 secs
```

Estos resultados indican la manera en que los datos fueron clasificados. Lo primero es que los datos se clasificaron primero por sexo, esto indica que es un atributo que discrimina fuertemente. Luego son discriminados por clase, pero no cada clase, se agrupan las clases: 1 y 2, porque que son similares, pero la clase 3 si es clasificada por separado.

Esto también indica que de las 121 iteraciones realizadas en la primera clasificación para determinar si una mujer sobrevivía se obtuvo 8 errores, lo que indica un poder de predicción cercano al 93.3%, lo cual podría ser un buen indicador. Sin embargo, en la clasificación de las mujeres de la clase 3, tanto para con padres como sin padres, se obtiene un poder de predicción cercano al 62.5% y 65.5% respectivamente, lo cual no es muy bueno, pero supone útil si es mayor al 50%. El resto de clasificaciones son muy buenas con un poder de predicción por encima del 86%.

Por otra parte, se observa que todas las variables se utilizaron, sin embargo, la de menor participación fue SibSp con apenas un 4,9%.

Se crea la tabla de confusión para determinar cuántos falsos positivos se tuvieron y así determinar además cual será la capacidad global de predicción del modelo.

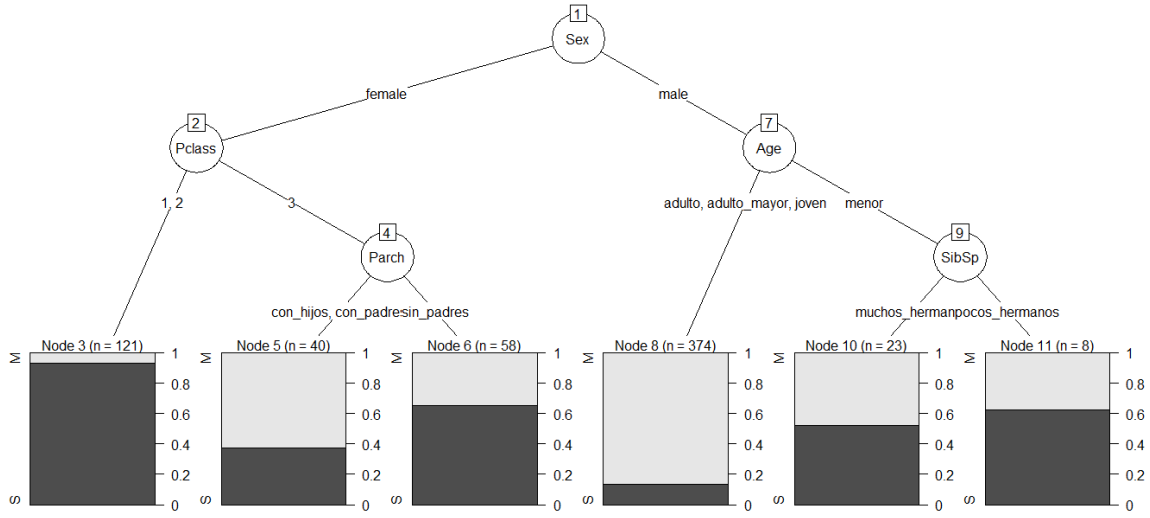
Se obtiene que en general se acertará en un 77,9% de las oportunidades.

```
> tablapred

predit   M   S
     M 143  43
     S  16  65

> (sum(diag(tablapred))/sum(tablapred))*100
[1] 77.90262
```

Árbol de clasificación y conclusiones:



No fue posible obtener una clasificación homogénea (donde cada una de las hojas fuera determinante), a pesar de hacer uso de todos los atributos, así que se debe hablar en términos de probabilidad de ocurrencia.

Gracias a los datos del resumen, cerca del 31,4% de los datos son clasificados como que sobreviven del total de datos. Por otra parte, se puede apreciar el peso de las diferentes hojas, a priori es posible ver que ser hombre, era una condición para tener una probabilidad de vida más baja, ya que sólo se requería esta condición y ser joven, adulto o adulto mayor para obtener una probabilidad de muerte cercana al 90%. Mientras que las mujeres de la clase 1 y 2 tenían esta misma probabilidad, pero de sobrevivir, mientras que las de clase 3 tenían en promedio una probabilidad del 50% dependiendo de si tenía hijos o no.

Así entonces se pueden tener las siguientes reglas:

Si el pasajero era hombre y no era menor tenía una probabilidad aproximada del 86% de morir.

Si el pasajero era hombre y era menor tenía una probabilidad aproximada del 60% de sobrevivir. Varía un poco si viajaba con muchos hermanos (50%) o con pocos o un hermano (61%).

Si el pasajero era mujer y era de clase 1 o 2, tenía una probabilidad del 6.6% de morir.

Si el pasajero era mujer y era de clase 3, y viajaba con hijos o padres, tenía una probabilidad de sobrevivencia del 38%.

Si el pasajero era mujer y era de clase 3, no viajaba con hijos o padres, tenía una probabilidad de sobrevivencia del 61%.

Como se mencionó anteriormente, se puede deducir que la gran mayoría de vidas que se salvaron en este accidente son de género femenino en términos relativos, ya que a bordo estaban muchas menos mujeres que hombres. Pero en términos absolutos también fueron más víctimas masculinas que femeninas.

Otro aspecto importante es que en la edad si influye si el pasajero era hombre. Ya que si se era menor se tenía mucha más probabilidad de sobrevivir de que si se fuera joven o adulto.

Código:

El código del presente trabajo, así como este mismo trabajo y la base de datos usada se puede encontrar en el siguiente enlace de GitHub para la realización de cualquier consulta. https://github.com/ccuervor/PRA2_ProyectoAnalitico

Referencias:

[1] Megan Squire (2015). Clean Data. Packt Publishing Ltd. Capitulos 1 y 2.

[2]León, E (2017). *Validación de Clusters. Minería de Datos.* Universidad Nacional de Colombia, Ingeniería de sistemas y computación. Recuperado el 10 de junio de 2018 de la página web: http://www.disi.unal.edu.co/profesores/eleonguz/cursos/md/presentaciones/Sesion13_validacion_Clustering.pdf

[3] Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing withExtreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.