Práctica 2: Limpieza y análisis de datos

Autor: Laura Pastor e Yosry Elsayed

7 de junio, 2022

Contents

Introducción

Presentación

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

Descripción del dataset

Trabajaremos con el juego de datos de entrenamiento del reto inicial de Kaggle "Titanic: Machine Learning from Disaster". La idea principal es entrenar un modelo predictivo que nos pueda indicar según las características de un pasajero si ha sobrevivido o no al incidente, lo cual es un conjunto interesante para así entrenar diferentes modelos supervisados y seleccionar el más adecuado sobre estos datos. El conjunto de entrenamiento se compone de los siguientes campos:

- **passengerId:** Valor numérico que especifica la clave primaria de cada pasajero.
- **name:** String con el nombre del pasajero.
- **sex:** Factor con niveles de hombre y mujer (male and female).
- **age:** Valor numérico con la edad de la persona el día del hundimiento. La edad de los bebés (menores de 12 meses) se da como una fracción de un año (1/mes).
- **pclass:** Factor que especifica la clase para los pasajeros o el tipo de servicio a bordo para los miembros de la tripulación.
- **embarked:** Factor con el lugar de embarque de la persona.
- **cabin:** Factor con el número de cabina de cada persona, si tiene.
- **ticket:** Valor numérico que especifica el número de billete de la persona (NA para miembros de la tripulación).
- **fare:** Valor numérico con el precio del billete (NA para tripulantes, músicos y empleados de la empresa astillero)
- **sibsp:** Factor ordenado especificando el número de hermanos/cónyuges a bordo; adoptado del conjunto de datos de Vanderbild.
- **parch:** Factor ordenado que especifica el número de padres/hijos a bordo; adoptado del conjunto de datos de Vanderbild.
- **survived:** Factor con dos niveles (no y sí) que especifica si la persona ha sobrevivido al hundimiento.

Integración y selección de los datos de interés a analizar.

El fichero "train.csv" contiene una serie de información sobre cada pasajero del famoso crucero y si ha sobrevivido a ello. Este primer conjunto de datos nos servirá para relaizar los primero pasos de limpieza y estandarización de los datos y como conjunto de entrenamiento para nuestros modelos predictivos y analizar la eficiencia de cada uno de ellos. Una vez decidido el mejor modelo, se aplicará sobre el conjunto "test.csv" que contiene una serie de información sobre distintos pasajeros al conjunto "train.csv" y no indica si han sobrevivido o no.

Cabe mencionar que las variables que identifican al pasajero como su nombre, identificador, identificador de la cabina o número de billete no son relevantes para el análisis de la supervivencia por lo que los vamos a ignorar.

Limpieza de los datos

Primero, instalamos y cargamos las librerías ggplot2 y dplry.

```
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
if (!require('dplyr')) install.packages('dplyr'); library('dplyr')
```

Cargamos el fichero de datos.

```
train <- read.csv("train.csv", stringsAsFactors = TRUE)
test <- read.csv("test.csv", stringsAsFactors = TRUE)
```

Verificamos la estructura del juego de datos principal.

```
dim(train)
```

```
## [1] 891  12
```

```
str(train)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : Factor w/ 891 levels "Abbing, Mr. Anthony",..: 109 191 358 277 16 559 520 629 417 58
```

```
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ 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     : Factor w/ 681 levels "110152","110413",..: 524 597 670 50 473 276 86 396 345 133 ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : Factor w/ 148 levels "","A10","A14",..: 1 83 1 57 1 1 131 1 1 1 ...
##  $ Embarked   : Factor w/ 4 levels "","C","Q","S": 4 2 4 4 4 3 4 4 4 2 ...
```

Vemos que tenemos 891 registros que se corresponden a los viajeros y tripulación del Titánic y 12 variables que los caracterizan.

Vamos ahora a sacar estadísticas básicas y después trabajamos los atributos con valores vacíos y ceros.

```
summary(train)
```

```
##   PassengerId       Survived          Pclass
##  Min.   :  1.0   Min.   :0.0000   Min.   :1.000
##  1st Qu.:223.5   1st Qu.:0.0000   1st Qu.:2.000
##  Median :446.0   Median :0.0000   Median :3.000
##  Mean   :446.0   Mean   :0.3838   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :891.0   Max.   :1.0000   Max.   :3.000
##
##                                     Name         Sex           Age
##  Abbing, Mr. Anthony                   :  1   female:314   Min.   : 0.42
##  Abbott, Mr. Rossmore Edward           :  1   male  :577   1st Qu.:20.12
##  Abbott, Mrs. Stanton (Rosa Hunt)      :  1                Median :28.00
##  Abelson, Mr. Samuel                   :  1                Mean   :29.70
##  Abelson, Mrs. Samuel (Hannah Wizosky):  1                3rd Qu.:38.00
##  Adahl, Mr. Mauritz Nils Martin        :  1                Max.   :80.00
##  (Other)                               :885                NA's   :177
##      SibSp           Parch             Ticket         Fare
##  Min.   :0.000   Min.   :0.0000   1601    :  7   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   347082  :  7   1st Qu.:  7.91
##  Median :0.000   Median :0.0000   CA. 2343:  7   Median : 14.45
##  Mean   :0.523   Mean   :0.3816   3101295 :  6   Mean   : 32.20
##  3rd Qu.:1.000   3rd Qu.:0.0000   347088  :  6   3rd Qu.: 31.00
##  Max.   :8.000   Max.   :6.0000   CA 2144 :  6   Max.   :512.33
##                                   (Other) :852
##          Cabin     Embarked
##             :687    :  2
##  B96 B98    :  4   C:168
##  C23 C25 C27:  4   Q: 77
##  G6         :  4   S:644
##  C22 C26    :  3
##  D          :  3
##  (Other)    :186
```

Ceros y elementos vacíos

Estadísticas de valores vacíos.

```
colSums(is.na(train))
```

```
## 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(train==0)
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         549           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##         608         678           0          15           0           0
```

Observamos que la única variable con valores vacíos es age. Y las variables survived, sibsp, parch y fare son las que contienen ceros, pero la presencia de ceros en cada una de ellas tiene sentido por la definición de cada una de las variables.

Para los valores vacíos de la variable age aplicamos la imputación por vecinos más cercanos, usando la distancia de Gower, y considerando que la imputación debe hacerse con registros de la misma edad o similares. Para ello usamos la función kNN() de la libreria VIM, donde indicamos nuestra tabla de datos, el nombre de las variables a imputar y la variable que se usa para el cálculo de la distancia de los vecinos.

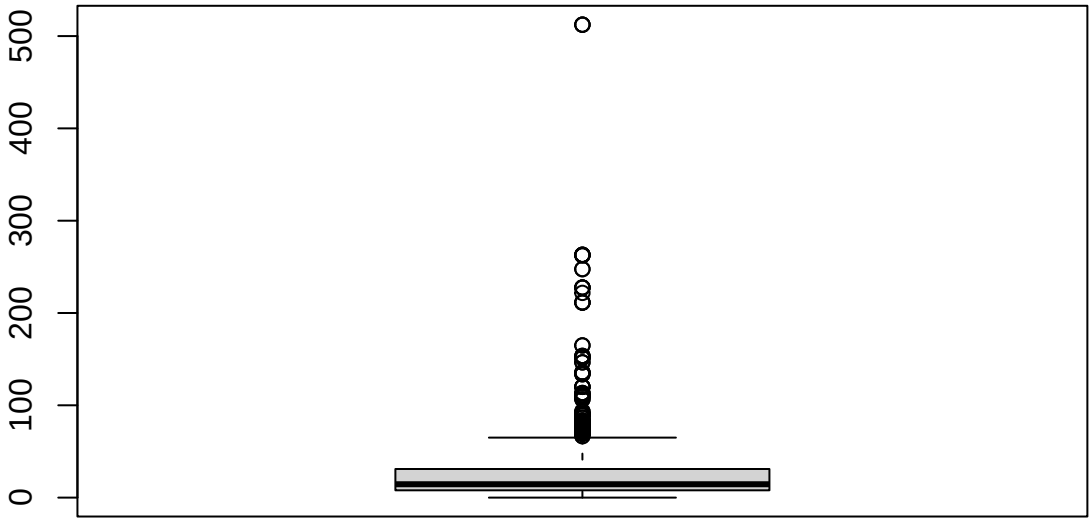
```
if (!require('VIM')) install.packages('VIM'); library('VIM')
#La nueva base imputada por vecinos más cercanos, usando la distancia de Gower
train<-kNN(train, variable = "Age", dist_var = c("Sex", "SibSp", "Parch", "Pclass"), imp_var=FALSE)
#Comprobamos que se han eliminado los valores nulos
colSums(is.na(train))
```

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

Valores extremos

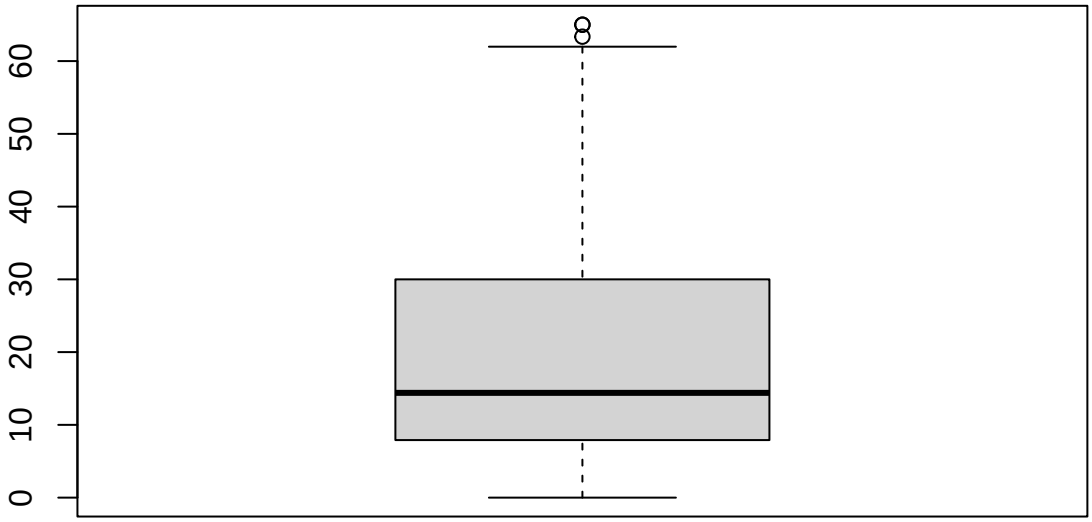
Analizamos los valores extremos (outliers) de la variable numérica “Fare” mediante boxplot, dado que hemos visto anteriormente mediante el summary que el resto de variables numéricas contienen mínimo, máximo y cuartiles que encajan con sus definiciones.

```
Fare_Boxplot<-boxplot(train$Fare)
```



Observamos que existen muchos valores atípicos en la variable, por lo que los convertimos en nulos y después aplicamos KNN mediante la variable PClass dado que el precio de los tickets deben ser similares para cada clase.

```
train$Fare[train$Fare %in% Fare_Boxplot$out]<-NA
#La nueva base imputada por vecinos más cercanos, usando la distancia de Gower
train<-kNN(train, variable = "Fare", dist_var = "Pclass", imp_var=FALSE)
#Comprobamos que se han eliminado los valores extremos
boxplot(train$Fare)
```



Observamos que solo se ven dos valores considerados como extremos según el boxplot, pero por definición de la variable son valores posibles y se encuentran cerca del máximo, por lo que no se modificarán dichos valores.

Con todo esto podemos decir que todos nuestros datos tienen una homogeneidad en la varianza.

Análisis de los datos

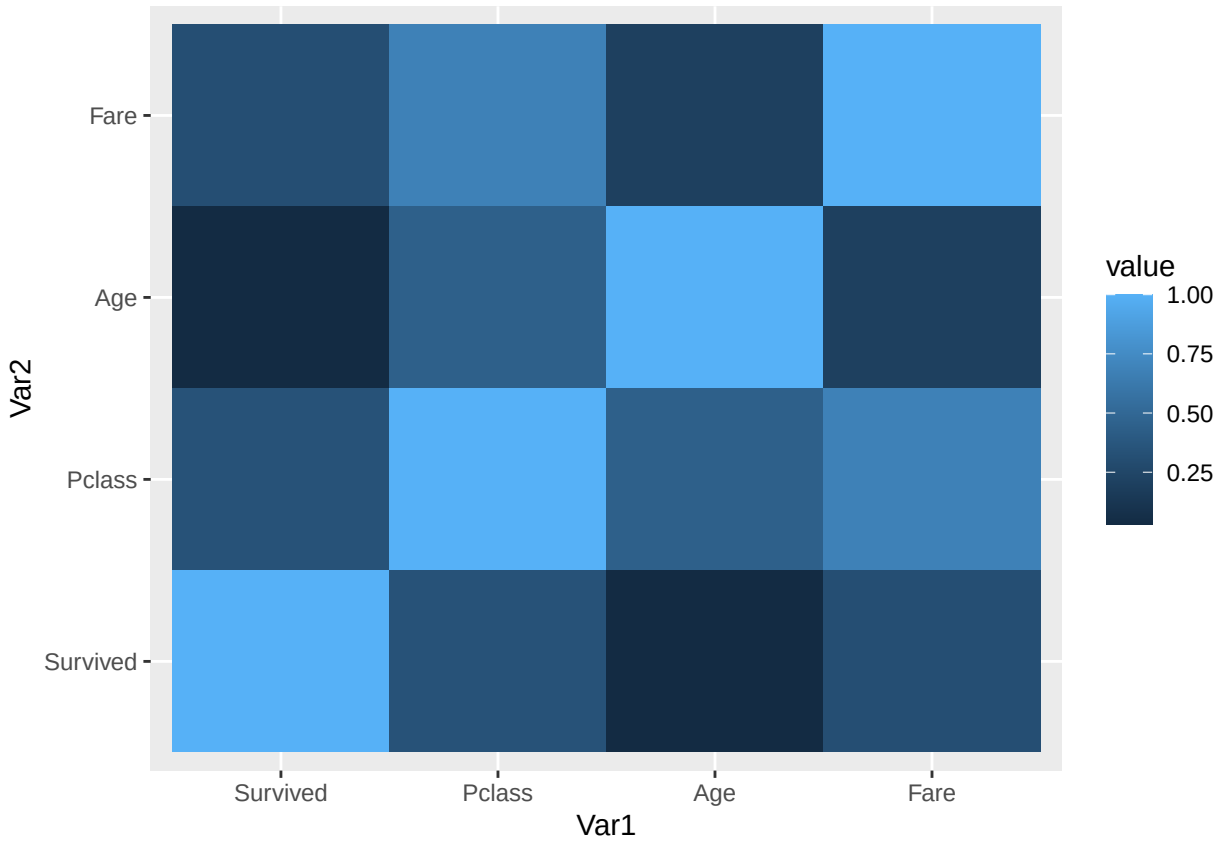
En esta práctica el objetivo es comparar los datos de los pasajeros que han sobrevivido o no al accidente del Titanic con el objetivo de encontrar características en común en cada grupo que nos ayude a generar correctamente nuestro modelo predictivo. Para ello vamos a utilizar el grupo de entrenamiento (train.csv).

Antes de nada vamos a realizar una comprobación de la normalidad de los datos. Esto es un paso imprescindible ya que en la mayoría de los modelos es un requisito que sus datos sean normales. Para ello vamos a aplicar el teorema central del límite que nos dice que si el tamaño de la muestra es lo suficientemente grande (más de 30 registros) tenderá a seguir una distribución normal por lo que como nuestra muestra es lo suficientemente gran de podemos considerar que es normal.

Además como acabamos de ver en el apartado anterior los datos de nuestro conjunto de entrenamiento tienen homocedasticidad.

Antes de comenzar con los modelos vamos a estudiar la correlación de las variables numéricas del juego de datos entre sí.

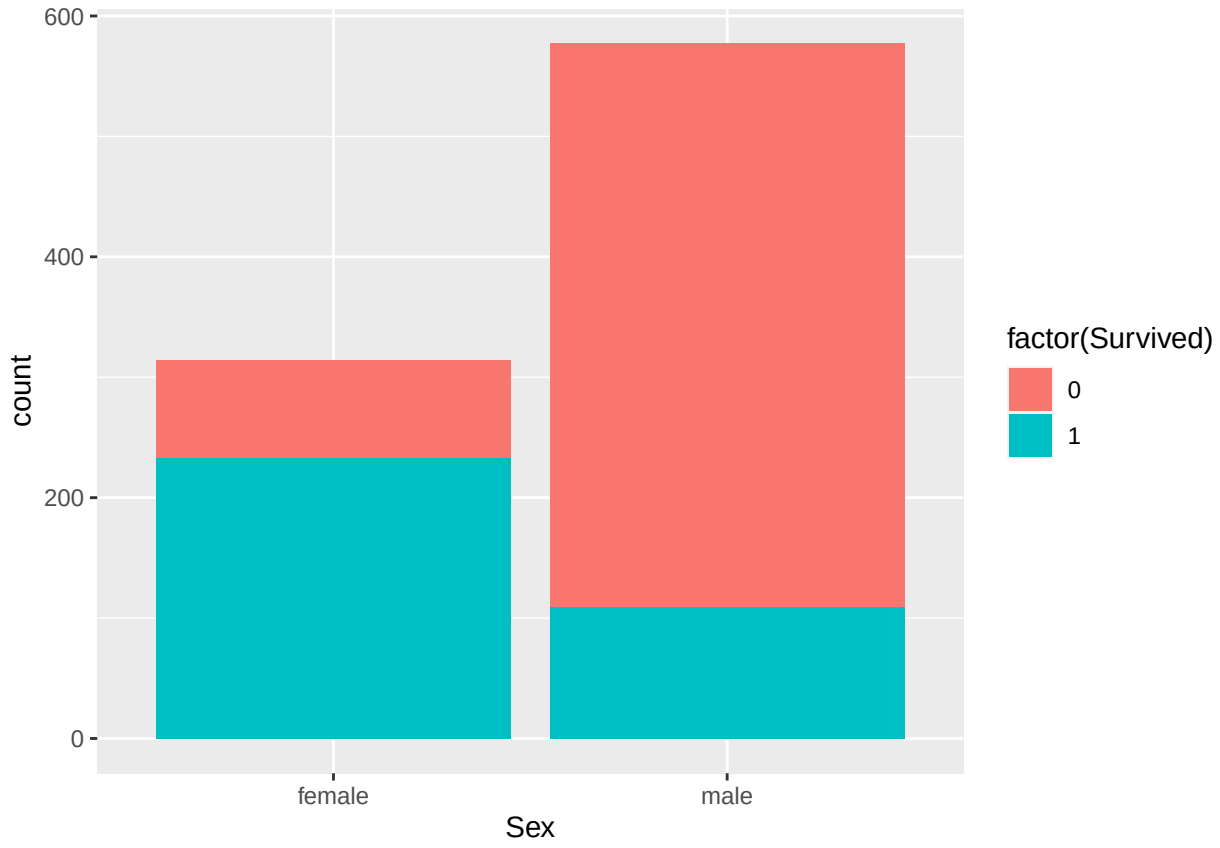
```
if (!require('reshape2')) install.packages('reshape2'); library('reshape2')
if (!require('ggplot2')) install.packages('ggplot2'); library('ggplot2')
ggplot(data = melt(abs(round(cor(train[,c(2,3,6,10)]),2))), aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()
```



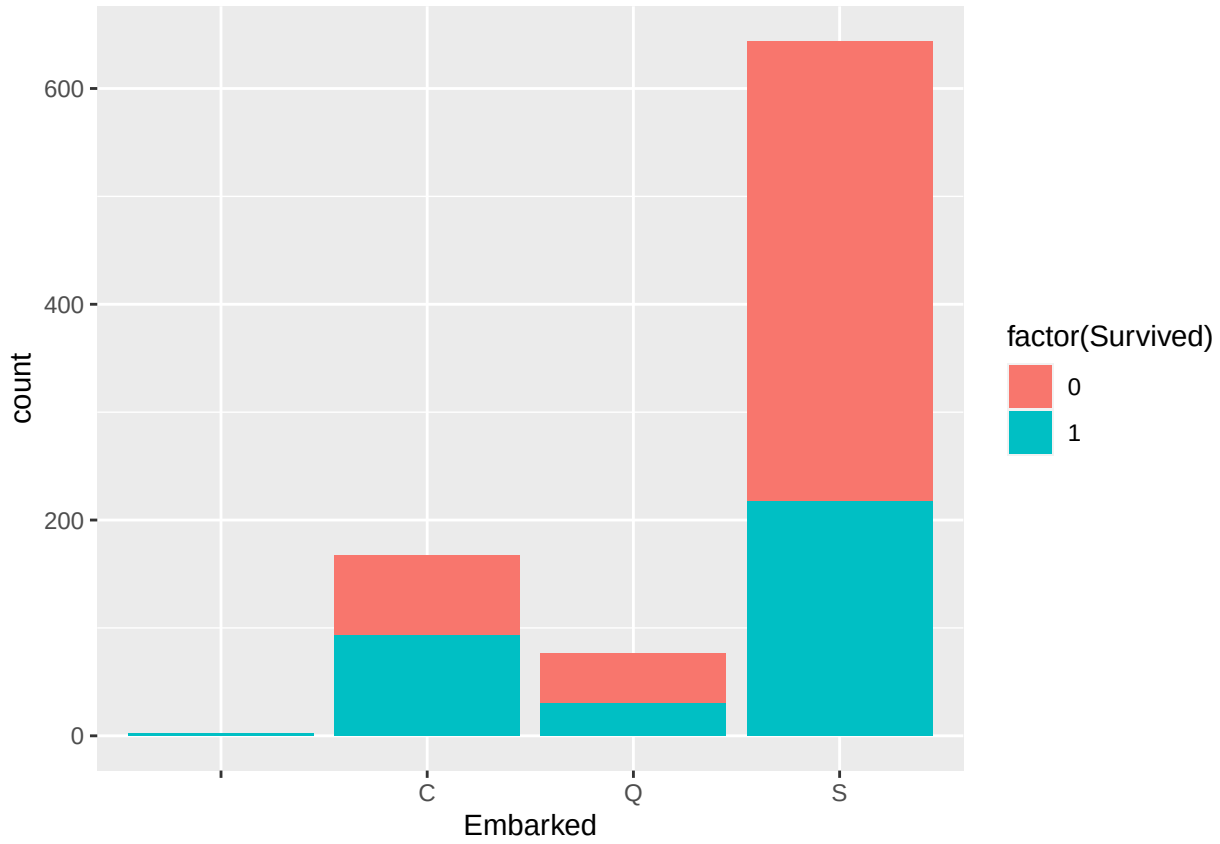
Observamos que las variables numéricas que tienen un mayor impacto en la supervivencia del accidente es la variable PClass y Fare, pero como están muy relacionadas entre ellas no tiene sentido utilizar ambas por lo que nos vamos a quedar a partir de ahora con la variable PClass.

Para las variables categóricas Sex y Embarked, analizamos si tienen impacto sobre la supervivencia de cada pasajero mediante gráficos.

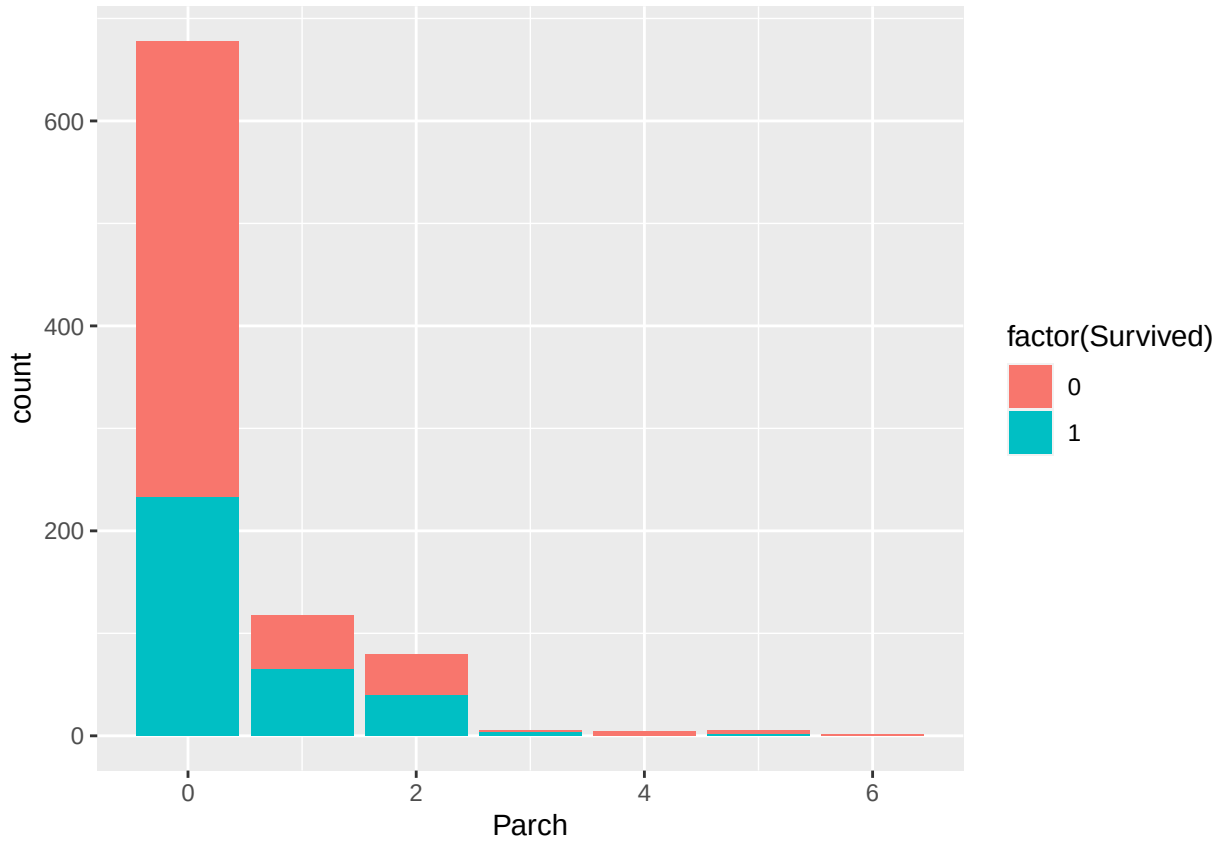
```
#Gráfico para la variable género
ggplot(data = train, aes(x = Sex, fill = factor(Survived))) +
    geom_bar()
```



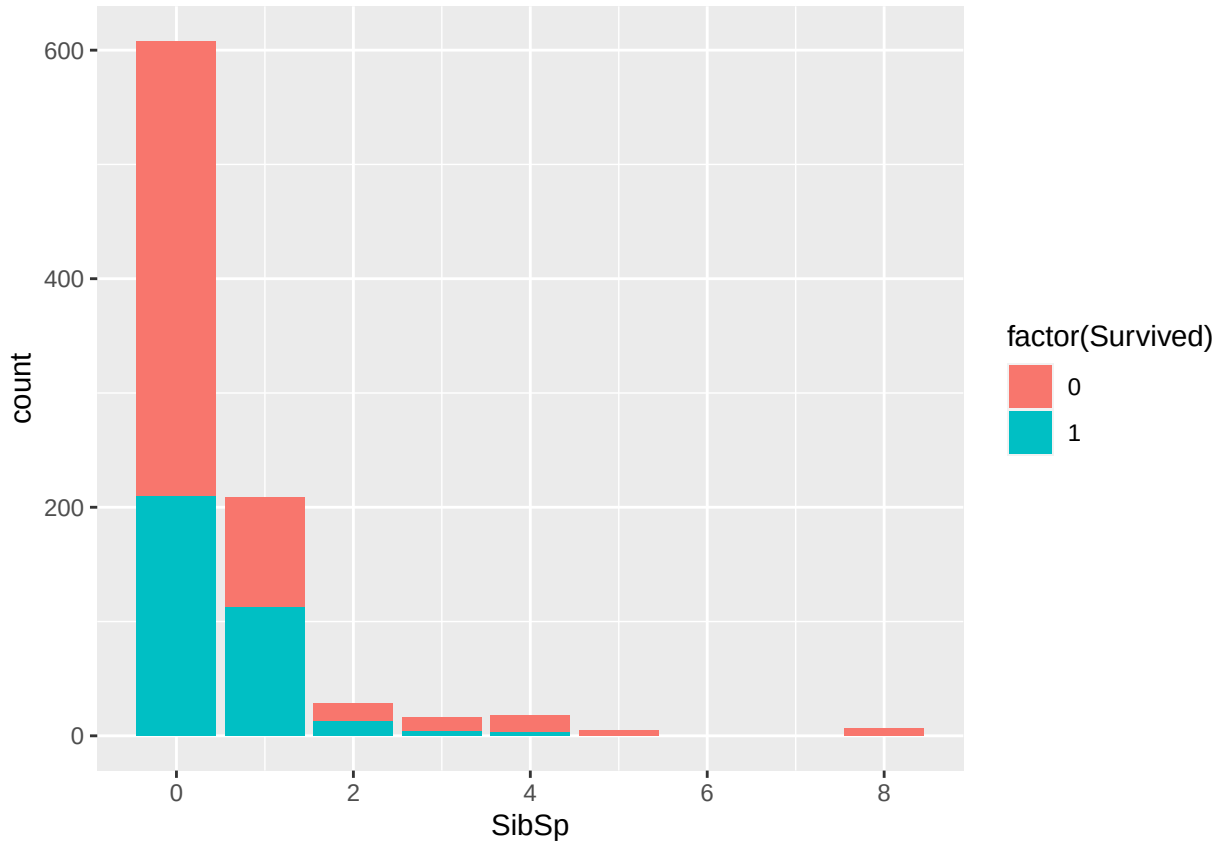

```
#Gráfico para la variable de embarque
ggplot(data = train, aes(x = Embarked, fill = factor(Survived))) +
    geom_bar()
```



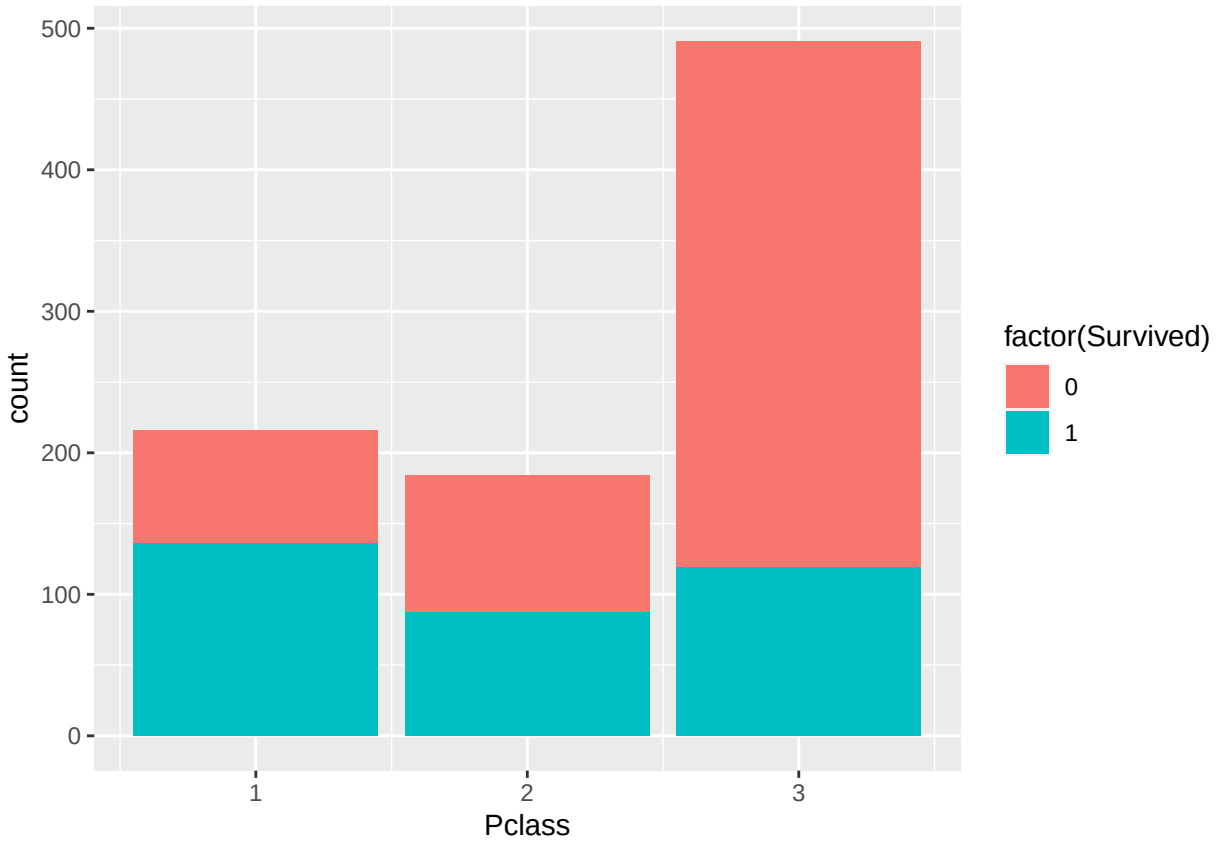

```
#Gráfico para la variable de Parch
ggplot(data = train, aes(x = Parch, fill = factor(Survived))) +
    geom_bar()
```




```
#Gráfico para la variable de SibPs
ggplot(data = train, aes(x = SibSp, fill = factor(Survived))) +
    geom_bar()
```




```
#Gráfico para la variable de Clase
ggplot(data = train, aes(x = Pclass, fill = factor(Survived))) +
    geom_bar()
```



En el primer gráfico vemos que las mujeres tienen una mayor tasa de supervivencia que los hombres. En el segundo gráfico vemos que la variable de embarque no parece tener mucha relevancia sobre la supervivencia de los pasajeros al accidente. En el tercer gráfico también podemos ver que la variable en este caso PArch no tiene relación con la supervivenvcia de los pasajeros y lo mismo ocurre con la variable SibPs (cuarto gráfico). Y por último el quinto gráfico confirma lo visto en la matriz de correlación anterior, que la clase influye bastante en la supervivencia de los pasajeros.

Con todo este análisis preliminar podemos determinar que las variables que más relación tienen con la supervivencia de los pasajeros son el Género (Sex), la clase (PClass) y en menor medida la edad (Age).

Modelo de regresión logística.

Generamos nuestro modelo predictivo mediante regresión logística con variable respuesta Survived y variables explicativas Sex, Fare y Parch.

```
model <- glm(Survived ~Sex+Fare+Pclass,family=binomial(link='logit'),data=train)
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ Sex + Fare + Pclass, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2007  -0.6714  -0.4441   0.6686   2.2100
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  2.786012   0.485178   5.742 9.34e-09 ***
## Sexmale     -2.613020   0.184790 -14.140  < 2e-16 ***
## Fare         0.010811   0.008252   1.310     0.19
## Pclass      -0.841330   0.138877  -6.058 1.38e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1186.7  on 890  degrees of freedom
## Residual deviance:  825.5  on 887  degrees of freedom
## AIC: 833.5
##
## Number of Fisher Scoring iterations: 4
```

```
anova(model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Survived
##
## Terms added sequentially (first to last)
##
##
##        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                     890    1186.66
## Sex     1  268.851       889     917.80 < 2.2e-16 ***
## Fare    1   53.174       888     864.63 3.053e-13 ***
## Pclass  1   39.134       887     825.50 3.958e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Gracias al summary, observamos que todas las variables explicativas del modelo son significativas.

Mediante el test anova, sabemos que la diferencia entre la desviación nula y la desviación residual muestra como nuestro modelo se compara con el modelo nulo (un modelo con solo la intersección). Cuanto más amplia sea esta brecha, mejor. Al analizar la tabla, podemos ver como al agregar Sex reduce significativamente la desviación residual. Además, un valor p grande aquí indica que el modelo sin la variable explica más o menos la misma cantidad de variación. Por lo que el test anova nos indica que la variable con mayor impacto sobre la variable respuesta es Sex.

Analizamos los residuos del modelo para ver si cumplen:

1. Media igual a 0.
2. Igualdad de varianzas.
3. Normalidad.

Empezamos con la media de los residuos:

```
round(mean(model$residuals),2)
```

```
## [1] 0.15
```

La media de los residuos es aproximadamente 0.15 por lo que podemos suponer que se cumple la primera condición.

Vemos la homocedasticidad de los residuos mediante la prueba de Breush-Pagan, que es un test que tiene como hipótesis nula que existe homocedasticidad y como hipótesis alternativa que existe heterocedasticidad.

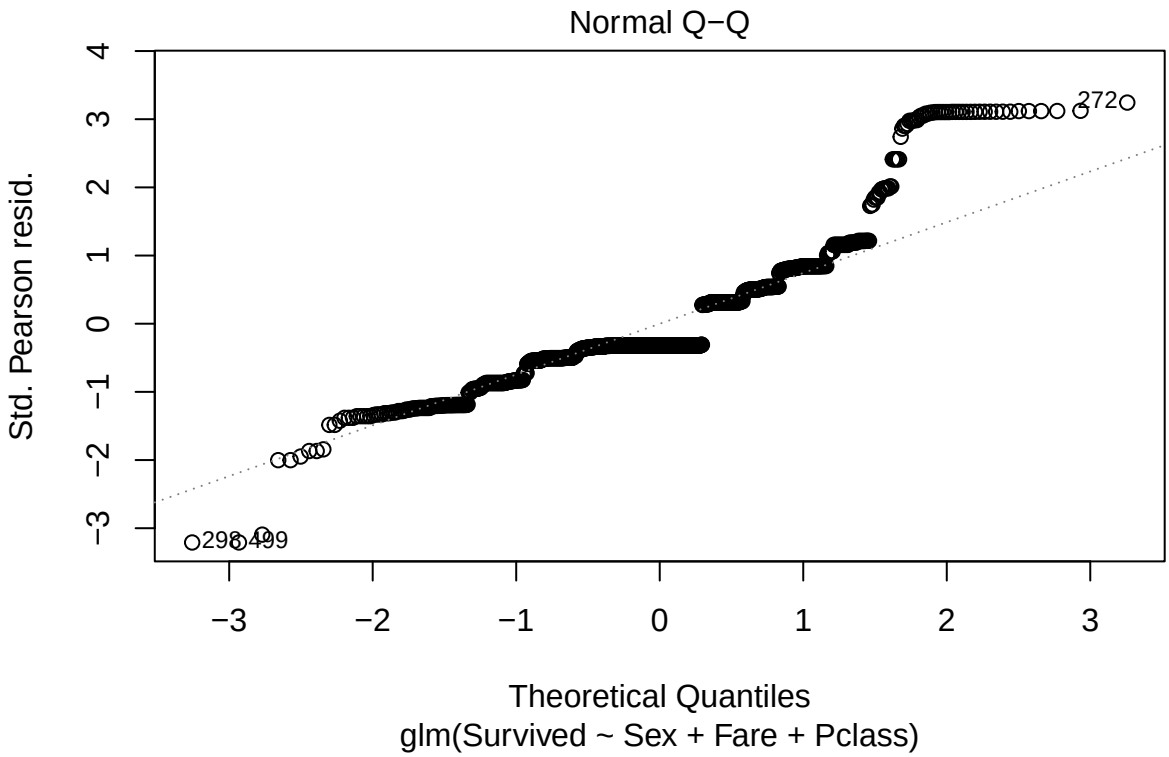
```
if (!require('lmtest')) install.packages('lmtest'); library('lmtest')
bptest(model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 11.907, df = 3, p-value = 0.007709
```

Se obtiene un p-valor menor que 0.05, por lo que se rechaza la hipótesis nula. Por lo que implica la heterocedasticidad de los residuos.

Por lo último comprabamos la normalidad de los residuos mediante un QQ-plot, a pesar de no tener la igualdad de varianzas. QQ-plot es un diagrama de dispersión que permite comparar distribución de probabilidades. Básicamente la lectura del gráfico es si los puntos del gráfico forman una línea recta sobre la línea marcada, implica que los residuos están distribuidos de forma normal.

```
plot(model,2)
```



Vemos que los puntos se desvían de la línea marcada, por lo que implica que los residuos no siguen una distribución normal.

Los residuos no cumplen las tres condiciones, por lo que nuestro modelo no se ajusta del todo bien a nuestros datos y es mejorable.

Una vez dicho esto vamos a estudiar la predicción del modelo.

```
#Sacamos las predicciones del modelo de regresión logística
predict_glm <- predict (model, train[,c(5,10,3)],type='response')
#Convertimos las probabilidades en los resultados de supervivencia 0 y 1
binary_predict_glm <- ifelse(predict_glm > 0.5,1,0)
```

```r
#Matriz de confusión
table(Real=as.factor(train[,2]), Predicted=as.factor(binary_predict_glm))
```

```
##     Predicted
## Real   0   1
##    0 467  82
##    1 108 234
```

```r
#Matriz de confusión en porcentaje
table(Real=as.factor(train[,2]), Predicted=as.factor(binary_predict_glm))/nrow(train)
```

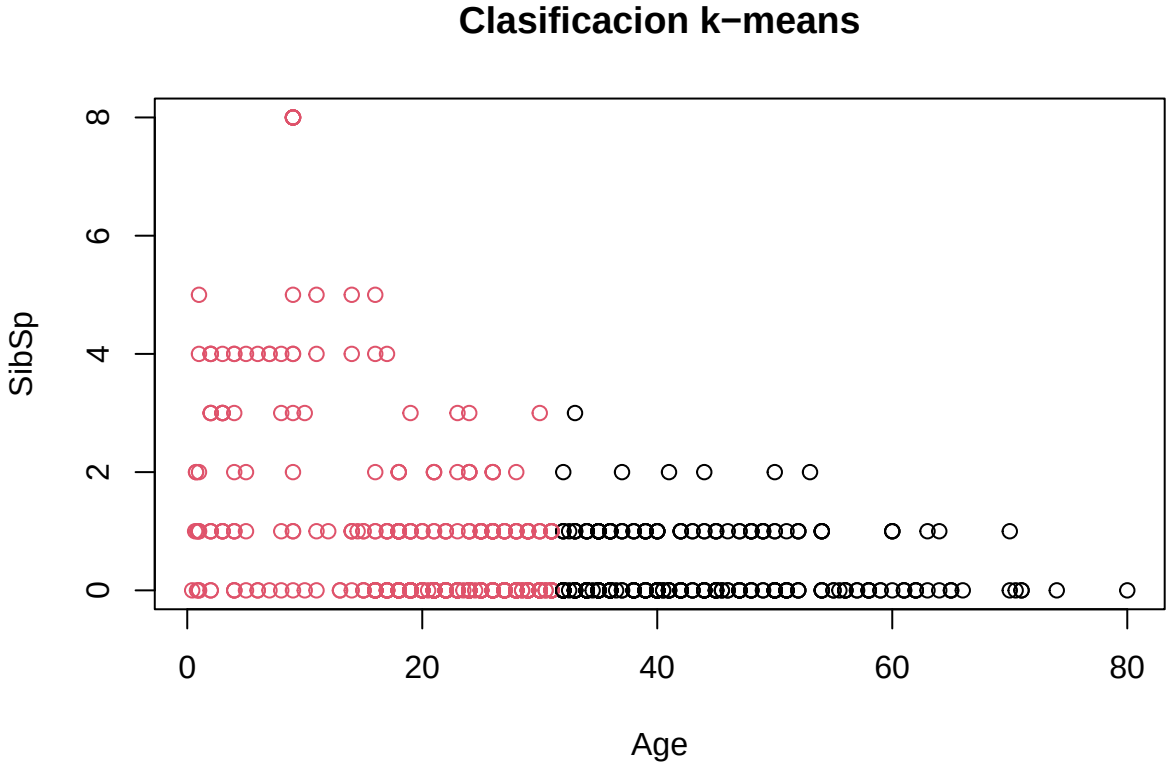
```
##     Predicted
## Real          0          1
##    0 0.52413019 0.09203143
##    1 0.12121212 0.26262626
```

A pesar de que los residuos indican que el modelo es mejorable, observamos que el modelo acierta con un 78% de exactitud. Por lo que podemos considerar que es un modelo bueno y aceptable para predecir la supervivencia de los pasajeros del Titanic.

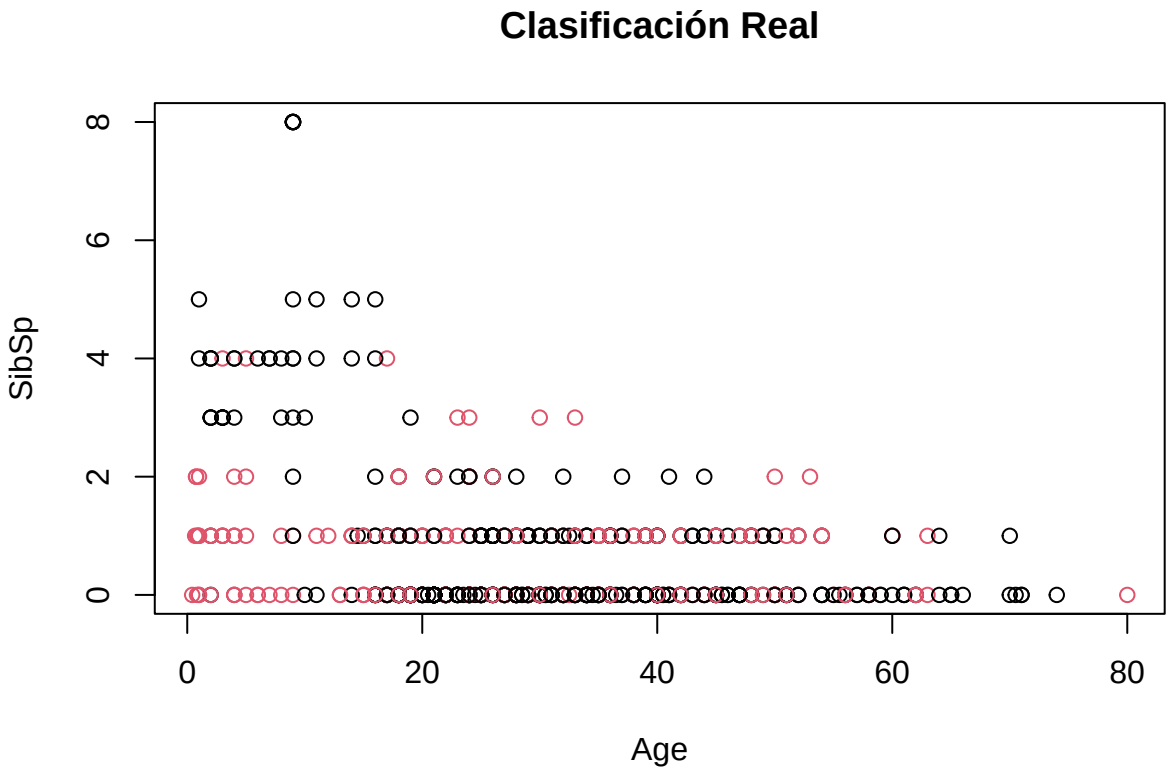
Modelo de clasificación

Vamos a saltarnos el proceso de elección de cual es el número de grupos óptimo para este conjunto de datos ya que nosotros sabemos que son 2.

```r
subtrain <- train[c(6,7)]
if (!require('fpc')) install.packages('fpc'); library('fpc')
model2 <-  kmeans(subtrain, 2)
plot(subtrain[c(1,2)],col=model2$cluster, main="Clasificacion k-means")
```




```r
plot(subtrain[c(1,2)],col=train$Survived+1, main="Clasificación Real")
```



El modelo necesita de variables numéricas para su predicción lo que hace que limita el número de variables que se pueden usar para entrenar al modelo. Además las variables numéricas no son las más influyentes en la supervivencia de las personas a bordo del Titanic con lo que podemos decir que este modelo no es muy útil para este conjunto de datos.

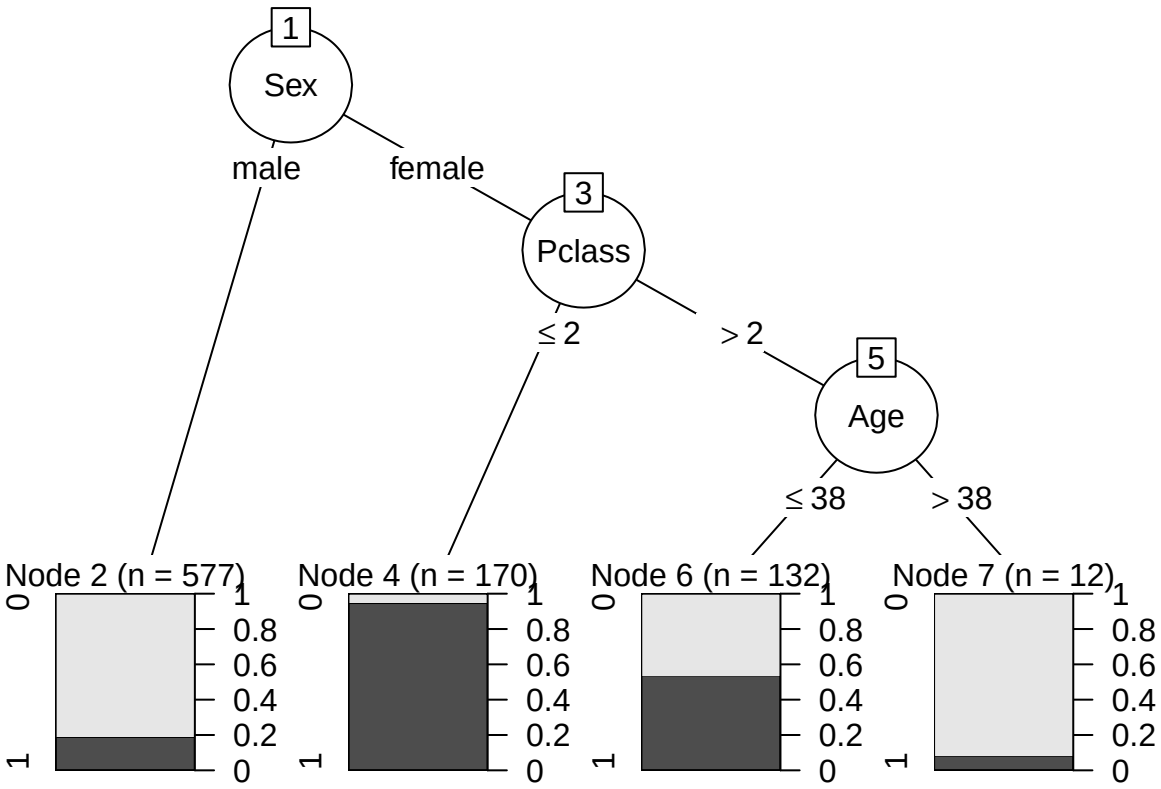
De todas formas podemos ver que para el caso de la relación de la edad con el precio del billete podemos ver que el modelo clasifica los datos en dos grupos. Los jóvenes que pagaron poco por su billete y por otro lado los más mayores que pagaron más por su billete. Como se puede ver en la comparación con la gráfica real esta clasificación es independiente de que sobrevivan o no. Como ya hemos dicho este modelo es insuficiente.

Modelo del diagrama de árbol

Ahora vamos a utilizar un modelo de decisión que se llama "Diagrama del Árbol" donde mediante separación de ramas se irá viendo la probabilidad de que los pasajeros vivan o mueran. Para este tipo de modelos es recomendable un reducido número de variables con un alto nivel de significacia por lo que una opción para mejorar este modelo es simplemente seleccionar las variables más influeyentes en la supervivencia de los pasajeros. Estas serían la edad, el género y la clase a la que pertenecen los pasajeros.La significancia de dichas variables con respecto a la supervivencia se vió al principio de este apartado.

Con todo esto vamos a aplicar el modelo del diagrama del árbol:

```
if (!require('C50')) install.packages('C50'); library('C50')
train3y <- as.factor(train$Survived)
train3x <- train[c(3,6,5)]
arbol1plot <- C50::C5.0(train3x, train3y)
plot(arbol1plot)
```



Con este árbol podemos concluir lo siguiente:

- Si es hombre muere con una probabilidad del 80%. Los hombres representan un 65% del total.
- Si es mujer sigue el árbol:
  - Si además de ser mujer pertenece a primera o segunda clase entonces sobrevive casi con una probabilidad del 100%. Esta casuística supone un 20% del total de los pasajeros.
  - En cambio si es una mujer pero de tercera clase entonces influye la edad.
    - Si es menor de 38 años entonces sobrevive con una probabilidad del casi 60%. Las muejeres de tercera clase menores de 28 años en el grupo de prueba son el 15%.
    - Por el contrario si se trata de un mujer de tercera clase mayor de 38 años (1.3% de los pasajeros del grupo de prueba) muere con una probabilidad superior al 90%.

Se puede ver que la variable Age que aparantemente no tenía mucha relación sobre la supervivencia de los pasajeros ha formado parte de una regla de decisión del modelo.

Una vez analizado el modelo de prueba vamos a predecir los datos del conjunto de prueba.

```
test3y <- as.factor(train$Survived)
test3x <- train[c(3,6,5)]
predict3 <- predict (arbol1plot, test3x, type="class")
mat_conf3 <- table(test3y, Predicted=predict3)
mat_conf3
```

```
##       Predicted
## test3y   0   1
##      0 479  70
##      1 110 232
```

Con lo que el modelo es capaz de predecir con un 80% de exactitud si un pasajero vive o muere analizando exclusivamente su edad, género y clase.

Resolución del probelma

Gracias a los modelos anteriores hemos podido determinar como era la supervivencia de los pasajeros del Titanic con bastante exactitud. Claramente los hombres lo tuvieron muy complicado ya que murió alrededor de un 80%. En cambio las mujeres tenían más oportunidades sobretodo si eran jóvenes, tal y como ha demostrado el modelo de decisión. Todo lo anterior se puede resumir en la famosa frase del desastre "Primero mujeres y niños", por lo que ellos fueron los que más oportunidades de salvarse tuvieron.

Además este problema nos a ayudado a ver cuales son los módelos más adecuados a cada tipo de datos. Para un conjunto de datos con muchas variables factoriales es recomendable utilizar un árbol de decisión. En cambio para los conjuntos de datos con más variables numéricas es más útil un modelo de clasificación. Para ambos casos los modelos de regresión logística dan buenos resultados aunque no son tan visuales como para los dos modelos anteriores.

Código

El código utilizado para trabajar con el conjunto de datos se ha ido mostrando dividido en cada uno de los apartados a lo largo de toda la práctica. Pero además se a recopilado todo en un único archivo de R que se adjunta también al repositorio de GitHub.

Generamos el output resultante del conjunto de entrenamiento después de su limpieza.

```
write.csv(train, file = "train_clean.csv")
```

Contribución al trabajo

Una tabla donde se muestra que cada uno de los autores/integrantes han participado en cada uno de los apartados del trabajo presentado.

Contribuciones	Firma
Investigación previa	Laura y Yosry
Redacción de las respuestas	Laura y Yosry
Desarrollo del código	Laura y Yosry

Enlaces

- Video: https://drive.google.com/file/d/1BdSjSFXOMuf_NkFyuspu0iFnasZZtFQN/view?usp=sharing
- GitHub (Yosry): https://github.com/Yoyazoooo20/PRA2-Titanic
- GitHub (Laura): https://github.com/lpastoram/PRA2-Titanic