Práctica 2: Limpieza y validación de los datos

Gonzalo Mellizo-Soto

7 de enero 2019

Contents

1. Información sobre la actividad

1.1 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.

1.2 Objetivos

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

1.3 Competencias

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

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

2. Desarrollo

2.1 Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

En este caso se trata de uno de los datasets propuestos en la práctica, concretamente el dataset del Titanic.Para ello lo primero será cargarlo e identificar sus características.

El número de observaciones es 891 y consta de 12 variables. Podemos realizar un pequeño resumen de las mismas:

- survival: variable binaria que indica si la persona sobrevivió o no al incidente del titanic
- pclass: clase del billete comprado
- sex: sexo de la persona
- Age: edad en años
- sibsp: número de hermanos y/o esposas en el barco
- parch: número de padres y/o hijos a bordo del titanic
- ticket: número del ticket

- fare: precio del ticket en el momento de la compra
- cabin: identificador de la cabina
- embarked: puerto desde el cual se embarcó al titanic

Se trata de un problema muy utilizado para adentrarse dentro del Machine Learning gracias a la simplicidad de las variables y al tratarse de un acontecimiento muy conocido. Se busca predecir las personas que sobrevivieron o murieron en la catástrofe de titanic, por ello en la competición de *Kaggle* se proporcionan dos datasets, *train* para entrenar el modelo y *test* para evaluar la predicción realizada.

Se puede realizar un pequeño resumen utilizando las funciones `summary` y `str`:

```
# Resumen de las variables
summary(titanic)
```

```
##   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
##
```

```
# Estructura de las variables
str(titanic)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
```

```
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  NA "C85" NA "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

2.2 Integración y selección de los datos de interés a analizar

En este caso los datos se encuentran completamente integrados en el propio dataset, por lo que no hace falta realizar ningún tipo de operación para formar el dataset completo, por lo que por defecto se encuentran todos los datos de interes en el propio dataset.

El único cambio que se va a realizar va a ser cambiar el sexo a una variable binaria:

```
titanic$Sex[titanic$Sex == 'male'] <- 1
titanic$Sex[titanic$Sex == 'female'] <- 0
titanic$Sex <- as.numeric(titanic$Sex)
```

3. Limpieza de los datos

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

Para ello se debe de analizar cada una de las variables y buscar por valores vacíos `NA`:

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

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0           0           0           0           0         177
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           0         687           2
```

Las acciones a realizar para cada variable son las siguientes:

- Age: se encuentran 177 valores vacíos, se procederá a imputar estos valores creando un pequeño modelo con las variables iniciales
- Embarked: se trata de un variable categórica de la que solo faltan dos registros, por ello se va a imputar utilizando la moda
- Cabin: con 687 registros sin información, la imputación o consideración como una variable útil es más bien nula, por ello se va a descartar y se va a eliminar del conjunto.

Imputación Embarked

```
# Observamos el valor más común
table(titanic$Embarked)
```

```
##
##   C   Q   S
## 168  77 644
```

```r
# Se imputa a este valor los NA
titanic$Embarked[is.na(titanic$Embarked)] <- 'S'
```

Eliminación de Cabin

```r
titanic$Cabin <- NULL
```

Imputación de Age

```r
# Utilizamos un randomForest rápido para generar un modelo para imputar la edad
set.seed(101)
modelVars <- c('Survived', 'Pclass', 'Sex', 'SibSp',
                                        'Parch')
ageModel <- randomForest(formula = Age ~ Survived + Pclass + Sex + SibSp + Parch,
                         data = titanic[!is.na(titanic$Age), c('Age',modelVars)])
predictedAge <- predict(ageModel, titanic[is.na(titanic$Age),
                                      modelVars])

titanic$Age[is.na(titanic$Age)] <- round(predictedAge)
```

Comprobamos si se han eliminado los NA

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

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

Ahora vamos a revisar que variables contienen ceros:

```r
sapply(titanic, function(x) sum(x == 0))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         549           0           0         314           0
##       SibSp       Parch      Ticket        Fare    Embarked
##         608         678           0          15           0
```

Los pasos a realizar son los siguientes:

- Survived: los ceros en esta variable indican que la persona falleció en el incidente, por lo que no hay que realizar ninguna operación
- Sex: indica que se trata de una mujer, por lo que no se realiza ninguna operación
- SibSp: el valor indica que en el viaje el pasajero no tenía ningún hermano y/o esposa/marido
- Parch: el pasajero no constaa de padres y/o hijos
- Fare: para este caso en concreto, las observaciones no concuerdan con el precio pagado, por lo que se va a modificar el valor imputandolo a la mediana de la clase (Pclass)

```
titanic[titanic$Fare == 0,]
```

```
##     PassengerId Survived Pclass                              Name Sex Age
## 180         180        0      3               Leonard, Mr. Lionel   1  36
## 264         264        0      1             Harrison, Mr. William   1  40
## 272         272        1      3      Tornquist, Mr. William Henry   1  25
## 278         278        0      2       Parkes, Mr. Francis "Frank"   1  32
## 303         303        0      3  Johnson, Mr. William Cahoone Jr   1  19
## 414         414        0      2   Cunningham, Mr. Alfred Fleming   1  32
## 467         467        0      2             Campbell, Mr. William   1  32
## 482         482        0      2 Frost, Mr. Anthony Wood "Archie"   1  32
## 598         598        0      3               Johnson, Mr. Alfred   1  49
## 634         634        0      1    Parr, Mr. William Henry Marsh   1  38
## 675         675        0      2       Watson, Mr. Ennis Hastings   1  32
## 733         733        0      2             Knight, Mr. Robert J   1  32
## 807         807        0      1           Andrews, Mr. Thomas Jr   1  39
## 816         816        0      1                Fry, Mr. Richard   1  38
## 823         823        0      1  Reuchlin, Jonkheer. John George   1  38
##     SibSp Parch Ticket Fare Embarked
## 180     0     0   LINE    0        S
## 264     0     0 112059    0        S
## 272     0     0   LINE    0        S
## 278     0     0 239853    0        S
## 303     0     0   LINE    0        S
## 414     0     0 239853    0        S
## 467     0     0 239853    0        S
## 482     0     0 239854    0        S
## 598     0     0   LINE    0        S
## 634     0     0 112052    0        S
## 675     0     0 239856    0        S
## 733     0     0 239855    0        S
## 807     0     0 112050    0        S
## 816     0     0 112058    0        S
## 823     0     0  19972    0        S
```

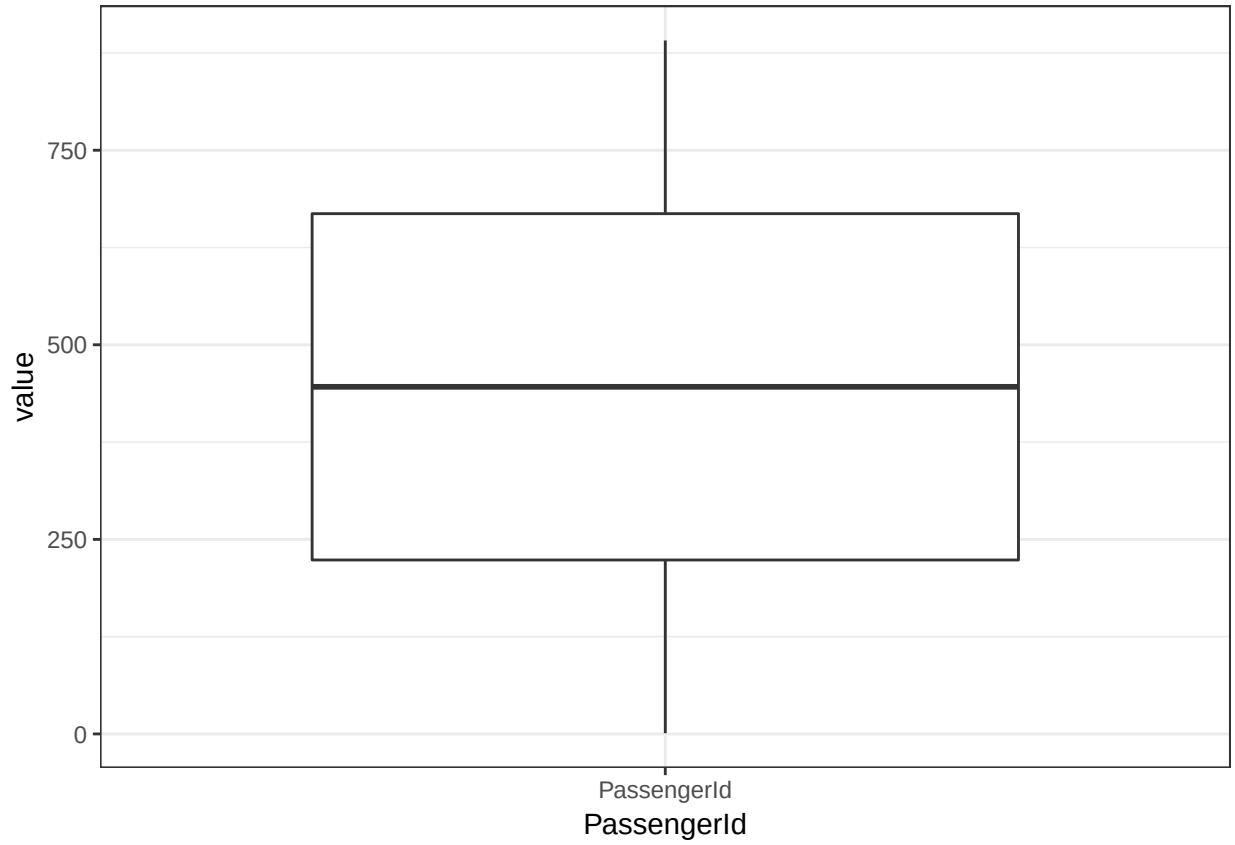
```
pclassMedianas <- sapply(split(titanic[titanic$Fare != 0,'Fare'],
                               f = titanic$Pclass[titanic$Fare != 0]),
                         median)

titanic[titanic$Fare == 0 & titanic$Pclass == 1,'Fare'] <- pclassMedianas[1]
titanic[titanic$Fare == 0 & titanic$Pclass == 2,'Fare'] <- pclassMedianas[2]
titanic[titanic$Fare == 0 & titanic$Pclass == 3,'Fare'] <- pclassMedianas[3]
```

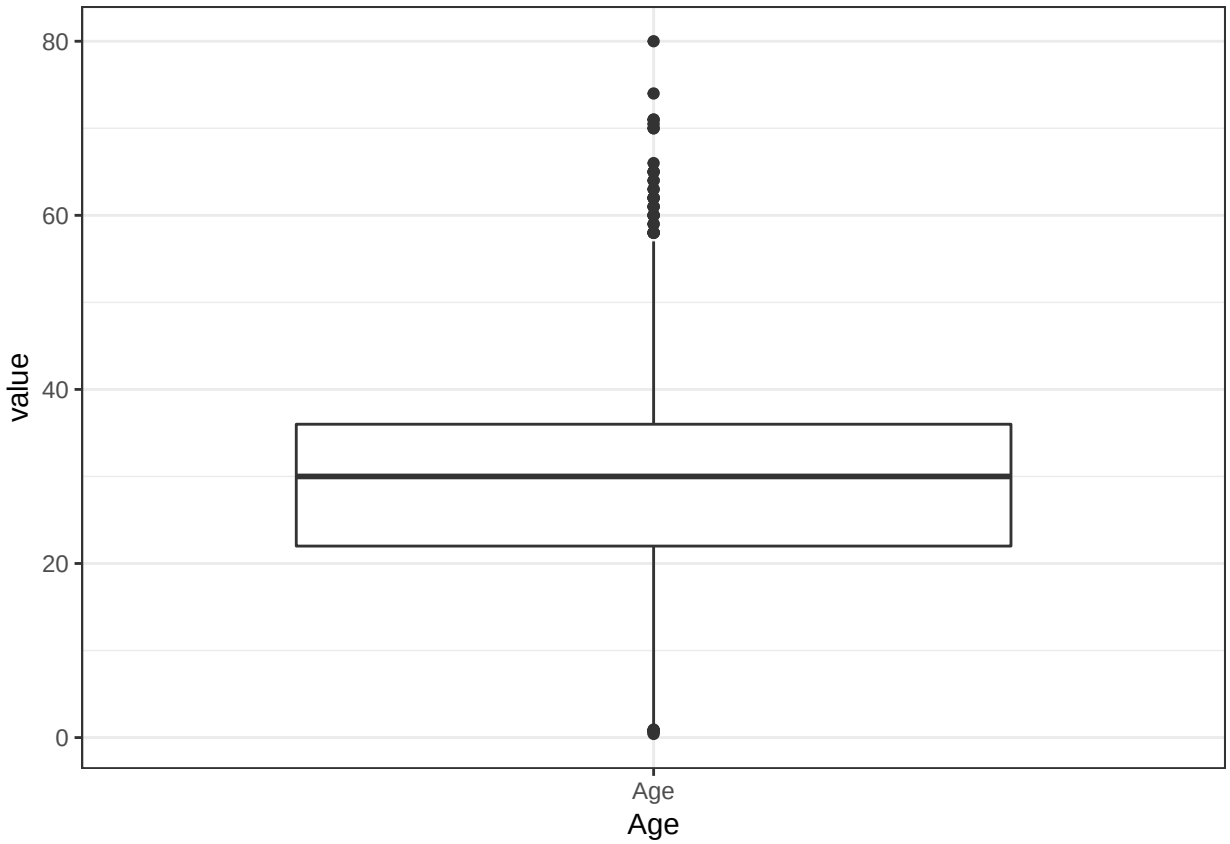
3.2 Identificación y tratamiento de valores extremos

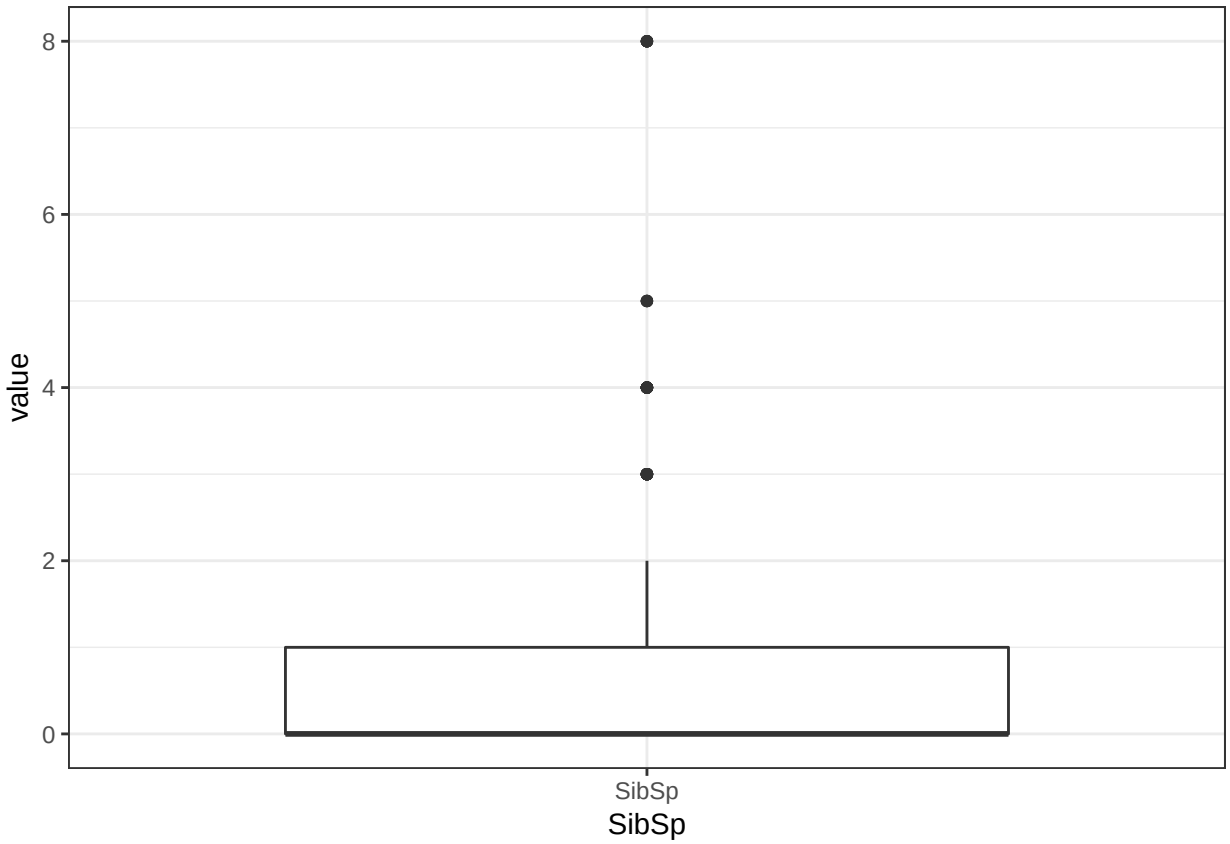
En este caso se van a utilizar solo las variables númericas para identificar los outliers y se van a identificar estos utilizando boxplots.

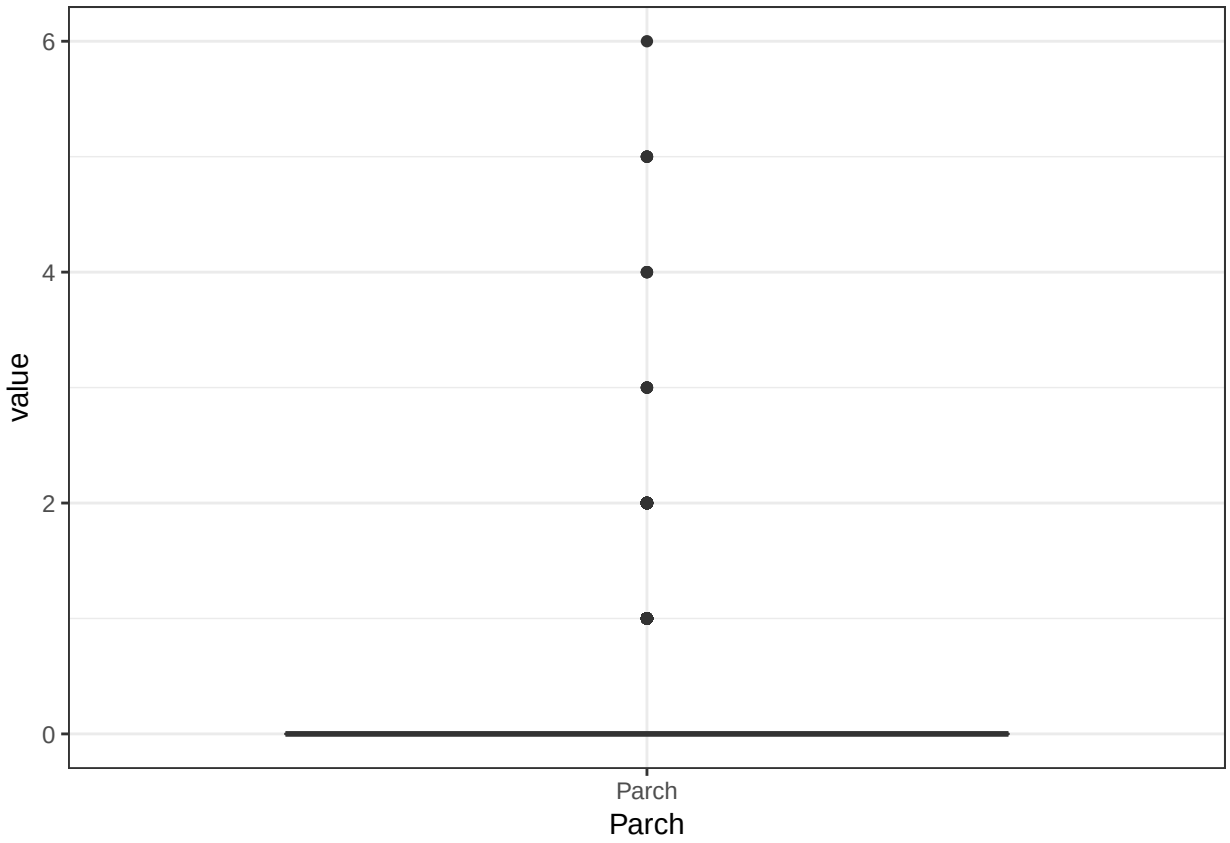
```
varsNum <- c('PassengerId', 'Age', 'SibSp', 'Parch', 'Fare')

plotList <- vector(mode = 'list', length = length(varsNum))
```

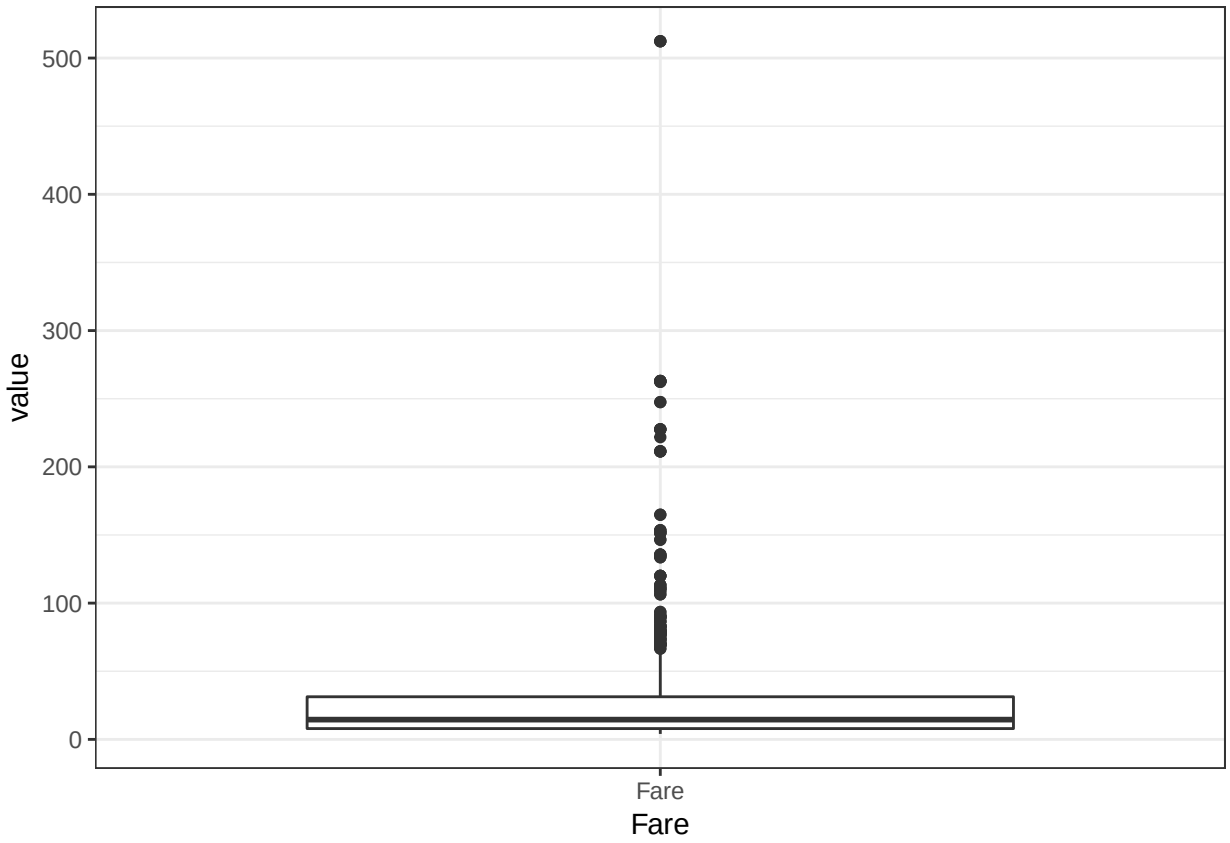
```r
for(variable in varsNum){
   p <- ggplot(titanic, aes(y = titanic[[variable]], x = variable))+
          geom_boxplot()+
          theme_bw()+
          ylab('value')+
          xlab(variable)
   plot(p)
}
```




80
60
40
20
0
value
Age
Age


8
6
4
2
0
value
SibSp
SibSp


6
4
2
0
value
Parch
Parch



En este caso se realiza un identificación visual para comprender en mayor profundidad si se trata de un valor atípico real (tiene sentido) o se trata de un valor producido por algún tipo de error.

Con las gráficas se aprecia lo siguiente:

- PassengerId: se trata de una variable identificativa con valores únicos
- Age: contiene valores atípicos reales (mayores a 1.5*IQR desde los cuartiles) con valores con sentido, personas mayores (entro 60 y 80) y niños pequeños (cerca de los cero años) y no se encuentran valores negativos
- SibSp y Parch se podrían tratar más como una variable categórica que como una variable numérica y como se aprecia se puede tener entre hijos y marido/esposa un valor de 8 y menor.
- Fare: para este caso si que nos encontramos con outliers que pueden ser a causa de un error y nos encontramos con un valor muy por encima del resto (hasta 200$ de diferencia) 512.3292. Por ello se va a cambar a un valor más normal, la mediana de la clase correspondiente (primera clase).

```
titanic[titanic$Fare == max(titanic$Fare),'Fare'] <- pclassMedianas[1]
```

Podría haberse utilizado la función `boxplot.stats`, sin embargo gracias a la visualización directa del boxplot facilita la interpretación del rango intercuartílico y la detección de outliers.

4. Análisis de los datos

4.1 Selección de los grupos de datos que se quieren analizar/comparar

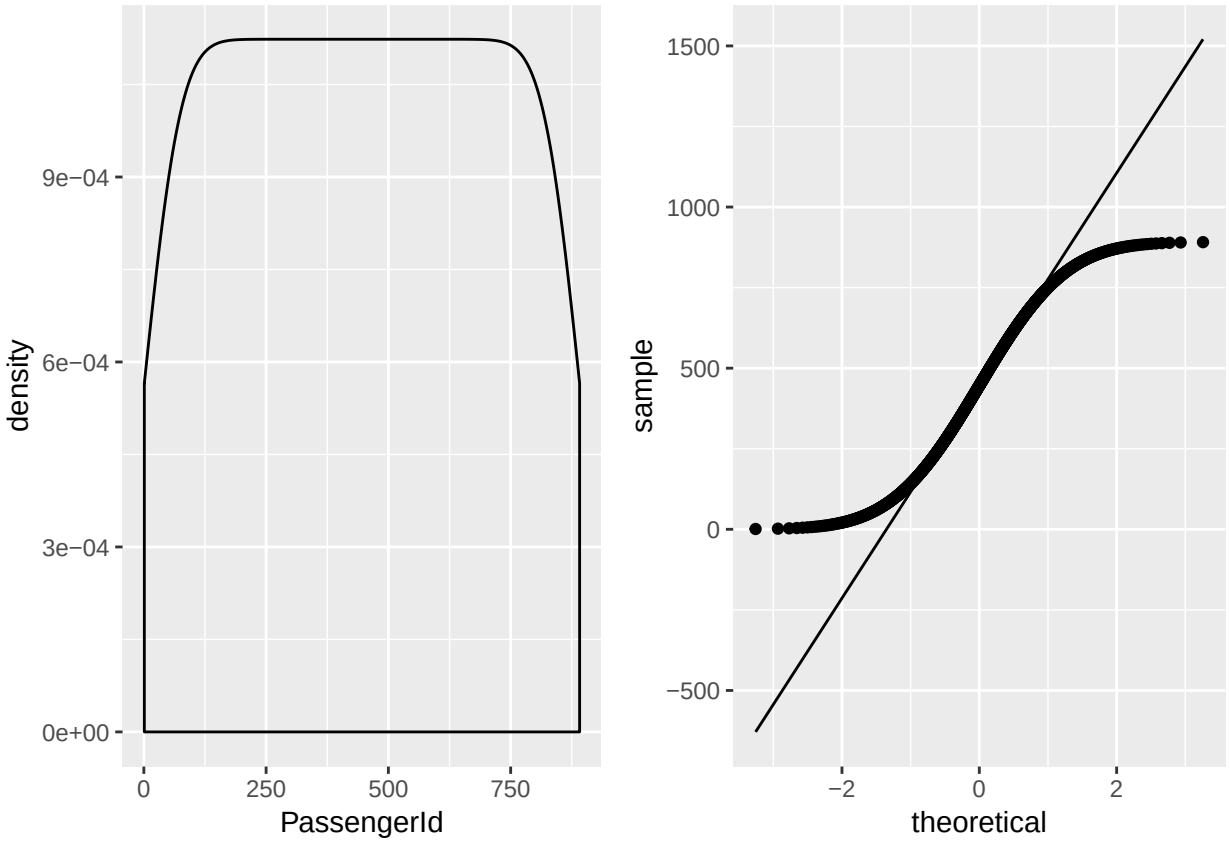
Para este caso en concreto se van a considerar las sigueintes pruebas:

- El efecto de sexo en la supervivencia de los pasajeros, las mujeres sobreviven más o menos
- La influencia de la clase del billete en la supervivencia del pasajero
- La supervivencia según la edad del pasajero
- Pruebas generales sobre las variables con respecto a la supervivencia

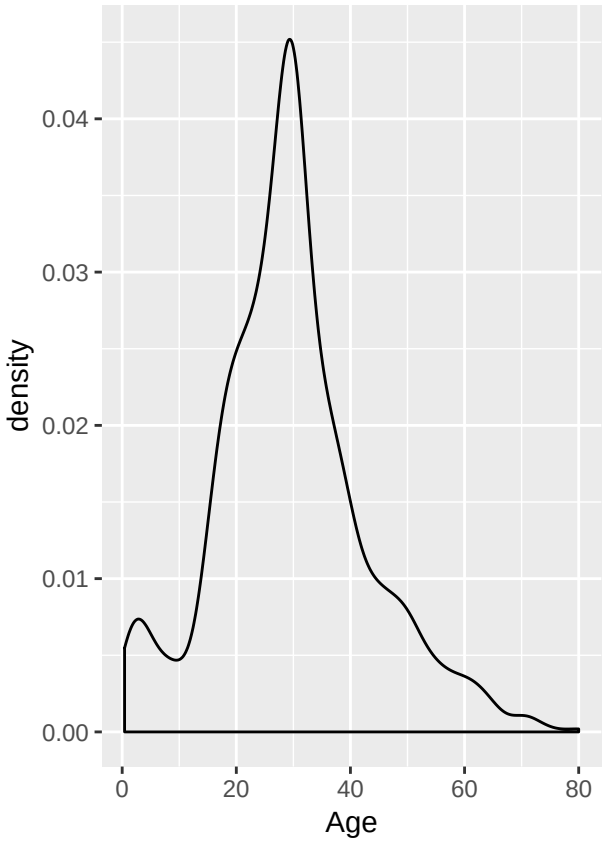
4.2 Comprobación de la normalidad y homogeneidad de la varianza.

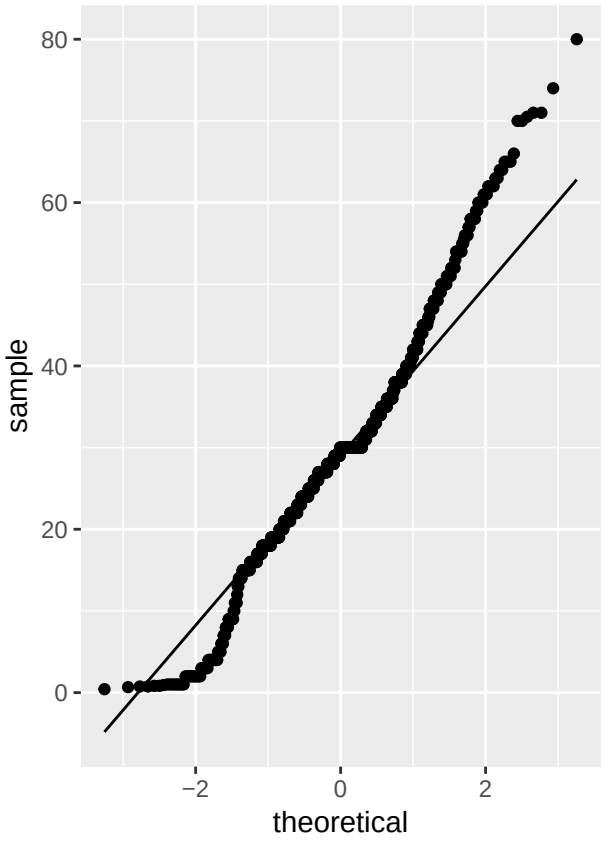
Para la comprobación de la normalidad se va a proceder a un análisis visual utilizando gráficas de densidad y q-q y por otro lado un análisis teórico utilizando el método *Shapiro-Wilk's*:

```
for(variable in varsNum){

  densityplot <- ggplot(titanic, aes(x = titanic[[variable]]))+
                 geom_density()+
                 xlab(variable)
  qq <- ggplot(titanic, aes(sample = titanic[[variable]]))+
            geom_qq()+
            geom_qq_line()
  grid.arrange(densityplot, qq, ncol=2)

  print(paste0("Análisis utilizando el método Shapiro-Wilk's para la variable",
               variable))

  print(shapiro.test(titanic[[variable]]))
}
```

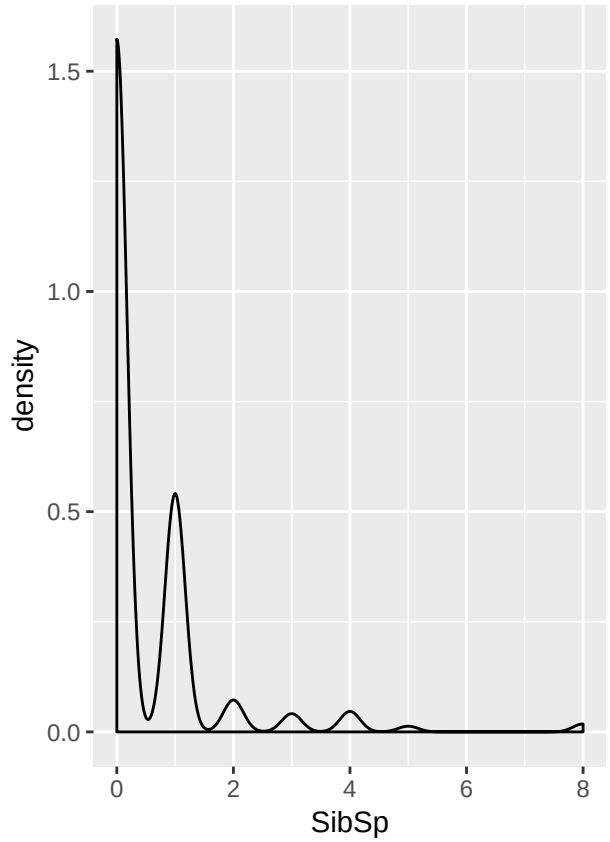


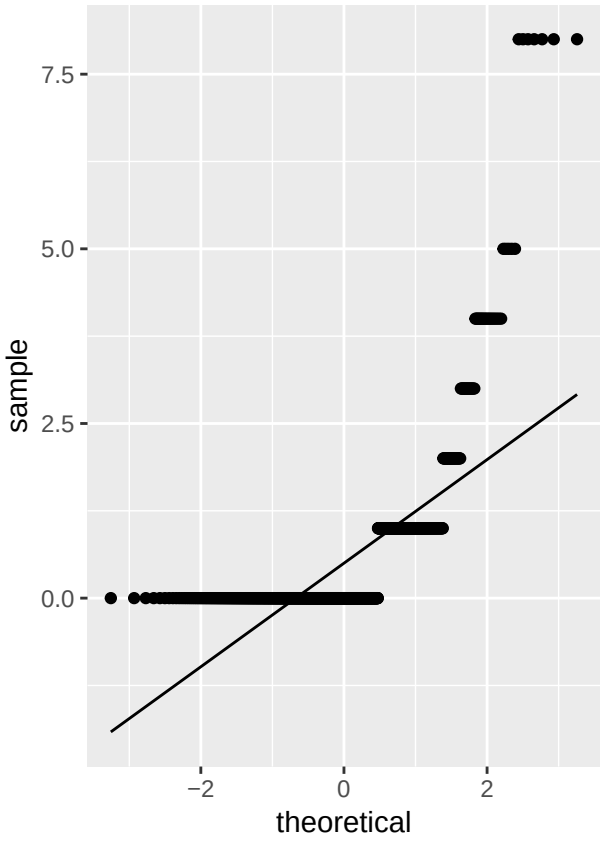

```
## [1] "Análisis utilizando el método Shapiro-Wilk's para la variablePassengerId"
##
##  Shapiro-Wilk normality test
##
## data:  titanic[[variable]]
## W = 0.9548, p-value = 6.308e-16
```



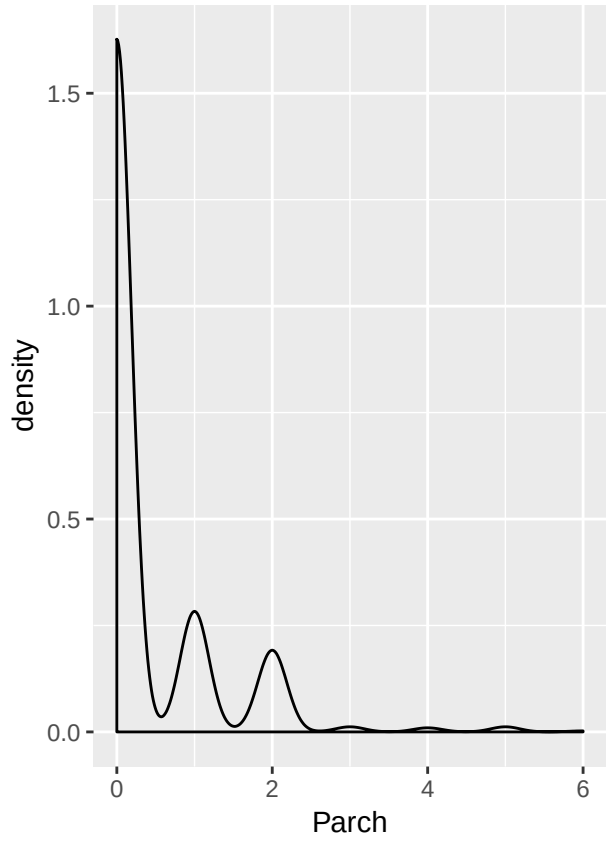


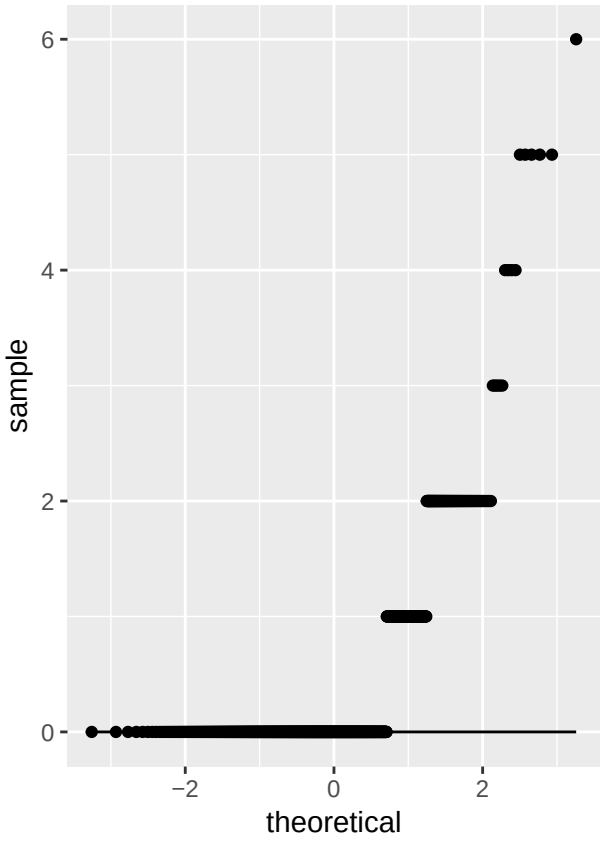

```
## [1] "Análisis utilizando el método Shapiro-Wilk's para la variableAge"
##
##  Shapiro-Wilk normality test
##
## data:  titanic[[variable]]
## W = 0.97325, p-value = 1.005e-11
```



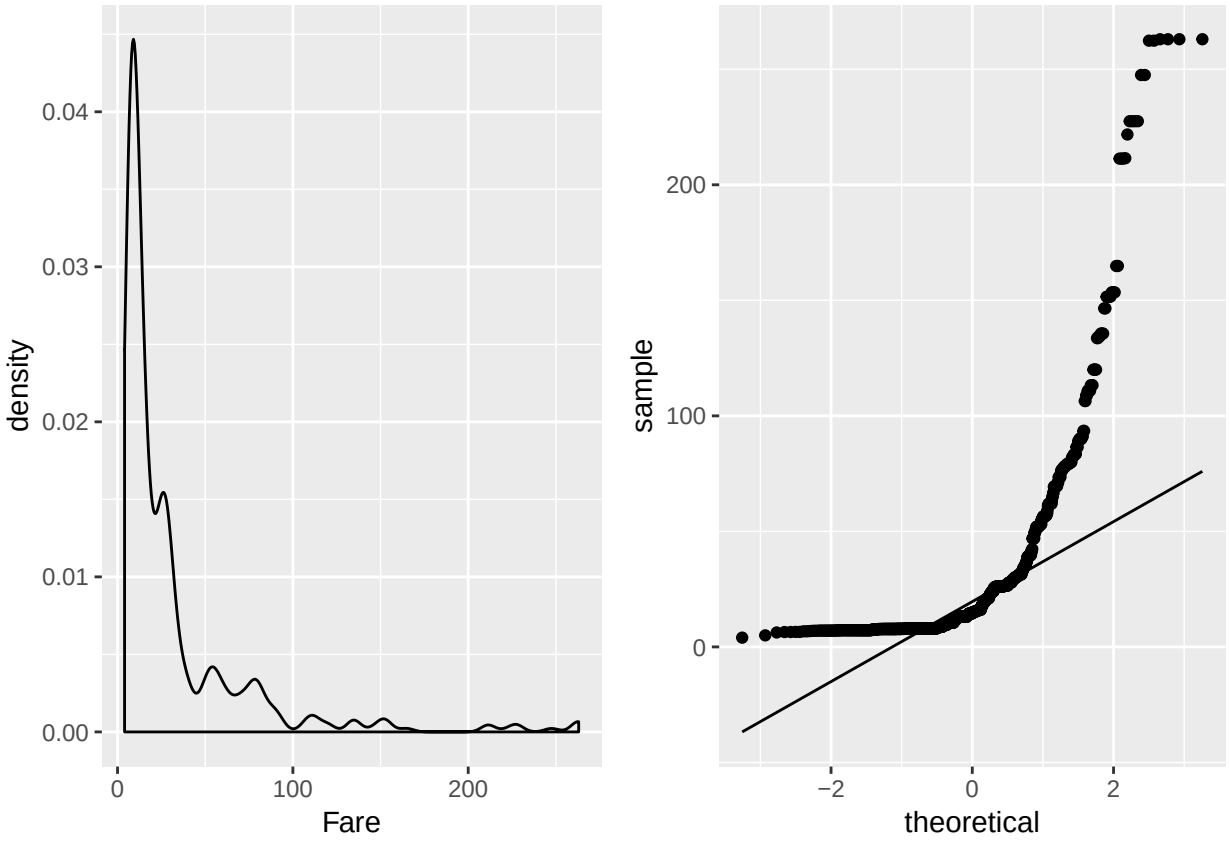



```
## [1] "Análisis utilizando el método Shapiro-Wilk's para la variableSibSp"
##
##  Shapiro-Wilk normality test
##
## data:  titanic[[variable]]
## W = 0.51297, p-value < 2.2e-16
```






```
## [1] "Análisis utilizando el método Shapiro-Wilk's para la variableParch"
##
##  Shapiro-Wilk normality test
##
## data:  titanic[[variable]]
## W = 0.53281, p-value < 2.2e-16
```




```
## [1] "Análisis utilizando el método Shapiro-Wilk's para la variableFare"
##
##  Shapiro-Wilk normality test
##
## data:  titanic[[variable]]
## W = 0.60434, p-value < 2.2e-16
```

Se puede apreciar como, tanto visualmente como analíticamente (p-valor menor de 0.05), los valores no siguen una distribución normal.

Para la prueba de homogeneidad se utilizará el test de Levene el cual es menos sensible a desviaciones de la normalidad. Se va a realizar utilizando la supervivencia como el grupo a comparar.

```
for(variable in names(titanic[,sapply(titanic, is.numeric)])){

  print(paste0("Análisis de homogeneidad de variancias con el test de Levene para la variable ",
               variable))
  print(levene.test(titanic[[variable]], titanic$Survived))

}
```

```
## [1] "Análisis de homogeneidad de variancias con el test de Levene para la variable PassengerId"
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
```

```
##
## data:  titanic[[variable]]
## Test Statistic = 1.665, p-value = 0.1973
##
## [1] "Análisis de homogeneidad de variancias con el test de Levene para la variable Survived"
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  titanic[[variable]]
## Test Statistic = NaN, p-value = NA
##
## [1] "Análisis de homogeneidad de variancias con el test de Levene para la variable Pclass"
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  titanic[[variable]]
## Test Statistic = 39.897, p-value = 4.226e-10
##
## [1] "Análisis de homogeneidad de variancias con el test de Levene para la variable Sex"
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  titanic[[variable]]
## Test Statistic = 38.301, p-value = 9.243e-10
##
## [1] "Análisis de homogeneidad de variancias con el test de Levene para la variable Age"
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  titanic[[variable]]
## Test Statistic = 4.0946, p-value = 0.04332
##
## [1] "Análisis de homogeneidad de variancias con el test de Levene para la variable SibSp"
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  titanic[[variable]]
## Test Statistic = 1.1106, p-value = 0.2922
##
## [1] "Análisis de homogeneidad de variancias con el test de Levene para la variable Parch"
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  titanic[[variable]]
## Test Statistic = 5.9635, p-value = 0.0148
##
## [1] "Análisis de homogeneidad de variancias con el test de Levene para la variable Fare"
##
```

```
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  titanic[[variable]]
## Test Statistic = 43.477, p-value = 7.351e-11
```

Observando los resultados, se concluye que la úncia variable con lo que se puede asumir una homogeneidad de varianzas es con *SibSp* dado que el p-valor es mayor a 0.05, mientras que en el resto no se puede asumir.

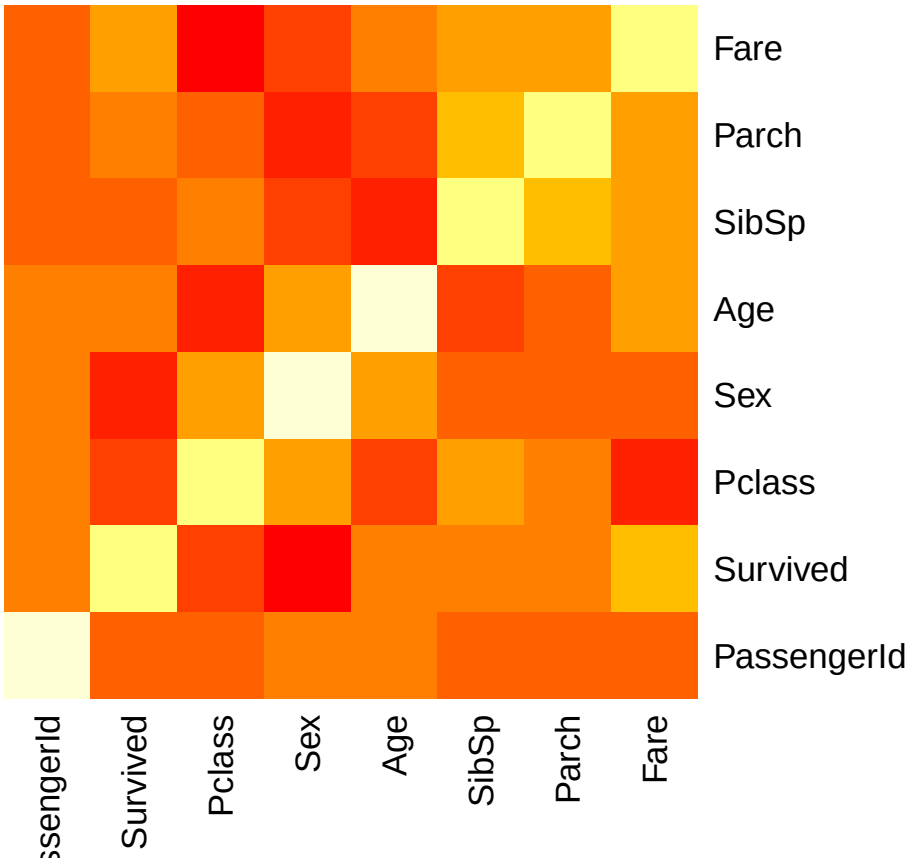
4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos

El primer paso va a ser realizar las correlaciones entre las distintas variables para obtener más información sobre como afectan a la variable independiente y si entre las dependientes existe alguna correlación que nos ayude a seleccionar las variables.

```
corrVar <- cor(titanic[,sapply(titanic, is.numeric)])
print(as.data.frame(corrVar))
```

```
##              PassengerId     Survived      Pclass         Sex         Age
## PassengerId  1.000000000 -0.005006661 -0.03514399  0.04293888  0.03813071
## Survived    -0.005006661  1.000000000 -0.33848104 -0.54335138 -0.07242633
## Pclass      -0.035143994 -0.338481036  1.00000000  0.13190049 -0.37225112
## Sex          0.042938880 -0.543351381  0.13190049  1.00000000  0.10688274
## Age          0.038130708 -0.072426333 -0.37225112  0.10688274  1.00000000
## SibSp       -0.057526834 -0.035322499  0.08308136 -0.11463081 -0.30728755
## Parch       -0.001652012  0.081629407  0.01844267 -0.24548896 -0.20936927
## Fare         0.006653859  0.255676329 -0.62053156 -0.21339335  0.10549405
##                   SibSp        Parch         Fare
## PassengerId -0.05752683 -0.001652012  0.006653859
## Survived    -0.03532250  0.081629407  0.255676329
## Pclass       0.08308136  0.018442671 -0.620531565
## Sex         -0.11463081 -0.245488960 -0.213393351
## Age         -0.30728755 -0.209369269  0.105494049
## SibSp        1.00000000  0.414837699  0.205113305
## Parch        0.41483770  1.000000000  0.258263731
## Fare         0.20511330  0.258263731  1.000000000
```

```
heatmap(corrVar, Rowv=NA, Colv=NA)
```



Con la correlación se puede observar como las variables SibSp y Parch no tienen mucha correlación con la variable a predecir, por lo que se podría considerar eliminarlas del dataset. Por otro lado, nos encontramos con una correlación significativa entre Pclass y Fare, teniendo la primera una correlación más alta con la variable independiente, por lo que se puede considerar eliminarla para evitar introducir ruido al modelo.

También se puede observar la correlación entre el sexo y clase del billete con lo supervivencia. Sobre la primera aparece una correlación negativo lo que indica que al tener un valor de 1 (ser hombre) influye negativamente en la supervivencia, con lo que se puede asumir que las mujeres viven más. Asimismo, la clase tiene un caorrelación negativa, a mayor nivel de clase (la peor clase es la 3) menos supervivientes.

Por último, la edad no aporta una información significativa frente a la supervivencia de los pasajeros.

Con esto podemos pasar a realizar los modelos, para este caso se van a utilizar varios modelos de randomForest:

```
set.seed(101)
titanic$Name <- NULL
titanic$Embarked <- factor(titanic$Embarked)
titanic$Ticket <- NULL
titanic$Survived <- factor(titanic$Survived)
titanic$Pclass <- factor(titanic$Pclass)
titanic$Sex <- factor(titanic$Sex)
rf1 <- randomForest(factor(Survived) ~ ., titanic)
rf2 <- randomForest(factor(Survived) ~ Pclass + Sex + Age, titanic)
rf3 <- randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch, titanic)
rf4 <- randomForest(factor(Survived) ~ Pclass + Sex + Age + Fare + Embarked, titanic)

for(i in 1:4){
```

```
  print(paste("Matriz de confusión para el modelo rf", i))
  print(eval(parse(text = paste0('rf',i,'$confusion'))))
}
```

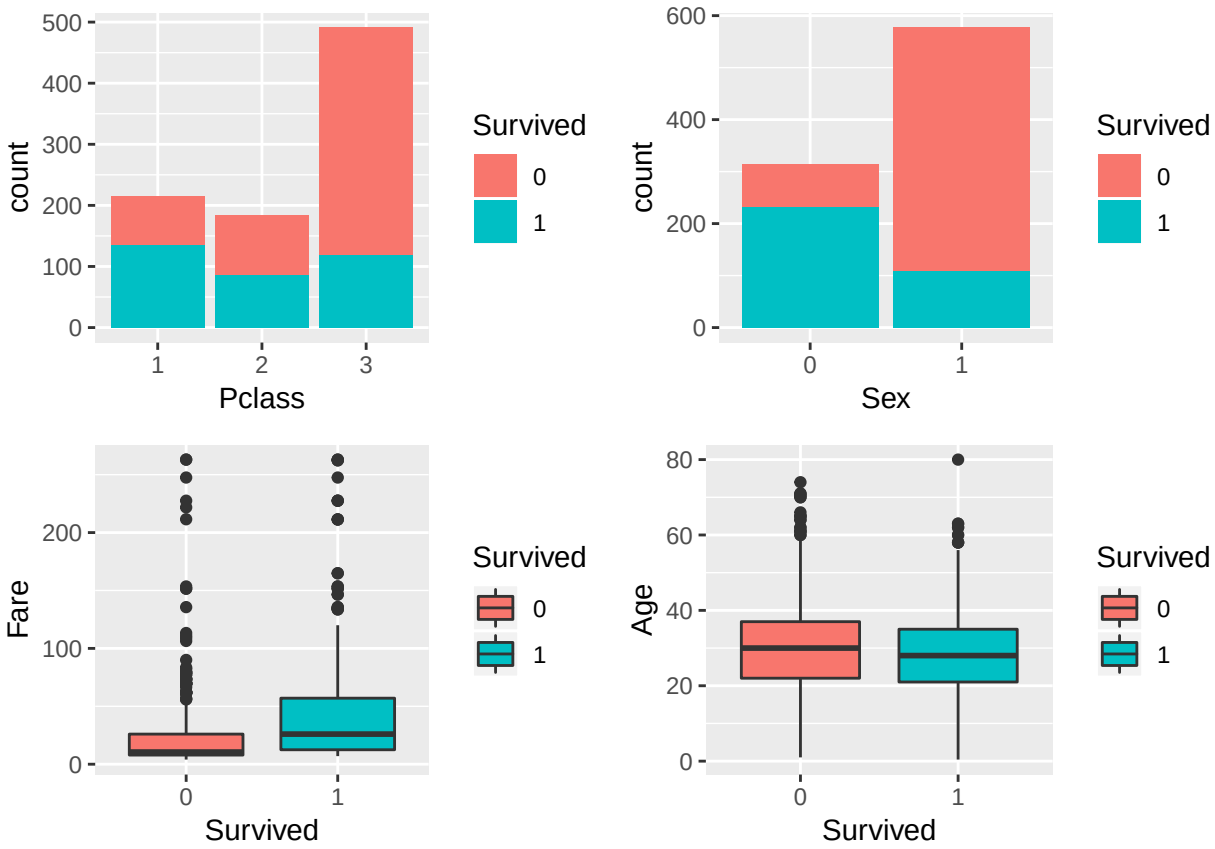
```
## [1] "Matriz de confusión para el modelo rf 1"
##     0   1 class.error
## 0 509  40  0.07285974
## 1  97 245  0.28362573
## [1] "Matriz de confusión para el modelo rf 2"
##     0   1 class.error
## 0 507  42  0.07650273
## 1 116 226  0.33918129
## [1] "Matriz de confusión para el modelo rf 3"
##     0   1 class.error
## 0 504  45  0.08196721
## 1  99 243  0.28947368
## [1] "Matriz de confusión para el modelo rf 4"
##     0   1 class.error
## 0 513  36  0.06557377
## 1  94 248  0.27485380
```

Para seleccionar el modelo, se puede pensar en que tipo de error se pretende disminuir, los errores de tipo uno o los errores de tipo dos. Concretamente en este caso podemos seleccionar el cuarto modela, dado que tiene el mayor *accuracy* de los cuatro.

5. Representación de los resultados a partir de tablas y gráficas

Las siguientes gráficas nos permiten entender en mayor medida los datos con los que estamos trabajando y las más significativas al realizar los modelos.

```
b1 <- ggplot(titanic, aes(x = Pclass, fill = Survived))+
        geom_bar()

b2 <- ggplot(titanic, aes(x = Sex, fill = Survived))+
        geom_bar()

b3 <- ggplot(titanic, aes(x = Survived, y = Fare, fill=Survived))+
       geom_boxplot()

b4 <- ggplot(titanic, aes(x = Survived, y = Age, fill=Survived))+
      geom_boxplot()

grid.arrange(b1, b2, b3, b4, ncol=2)
```



En las gráficas su pueden observar ciertas conclusiones a las que hemos llegado, como el sexo influye, más mujeres se salvaron, a más alta clase (menor valor) más proporción de supervivientes y su relación con el coste del billete.

6. Conclusiones

Según la limpieza y transformaciones empleadas, podemos llegar a las siguientes conclusiones:

- Las mujeres tenían más posibilidades de sobrevivir en el titanic que los hombres.
- Las clases más altas sobrevivieron proporcionalmente más que las clases bajas, siendo la primera y segunda clase en su conjunto menor que la tercera en número de pasajeros.
- No se aprecia una gran significación en la edad del pasajero o la cantidad de familiares.