Tipología de Datos, práctica 2

Autor: Adrián Antón Collado, aantonc@uoc.edu (mailto:aantonc@uoc.edu)

7 de Enero de 2019

1 Descripción de la Práctica a realizar

El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle (https://www.kaggle.com). Algunos ejemplos de dataset con los que podéis trabajar son:

- Red Wine Quality (https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009)

- Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic)

El último ejemplo corresponde a una competición activa de Kaggle de manera que, opcionalmente, podéis aprovechar el trabajo realizado durante la práctica para entrar en esta competición. Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

2 Descripción del dataset

¿Por qué es importante y qué pregunta/problema pretende responder?

El conjunto de datos elegido es Titanic: Machine Learning from Disaster (https://www.kaggle.com/c/titanic). La motivación de elegir este conjunto de datos es porque es una competición activa de Kaggle y ello supone un plus importante.

El problema describe datos sobre los pasajeros del Titanic (https://en.wikipedia.org/wiki/RMS_Titanic) y si han sobrevivido o no a dicho desastre. Tomando como base estos datos se pretende construir un modelo que sea capaz de predecir si un pasajero ha sobrevivido o no.

Mediante este modelo se pueden analizar las causas de la alta tasa de mortalidad para mejorar las medidas de emergencia a aplicar en caso de desastres parecidas.

3 Integración y selección de los datos de interés a analizar.

```
datosEntrenamiento <- read.csv("train.csv",header=T,sep=",")
datosTest <- read.csv("test.csv",header=T,sep=",")
str(datosEntrenamiento)
```

```
## '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
629 417 581 ...
##  $ 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 3
45 133 ...
##  $ 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 ...
```

```
summary(datosEntrenamiento)
```

```
##   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
```

La variable `PassengerId` identifica univocamente al pasajero, por lo que hay 891 valores diferentes. No parece una variable a tener en cuenta para la construcción del modelo, por lo que la eliminaremos.

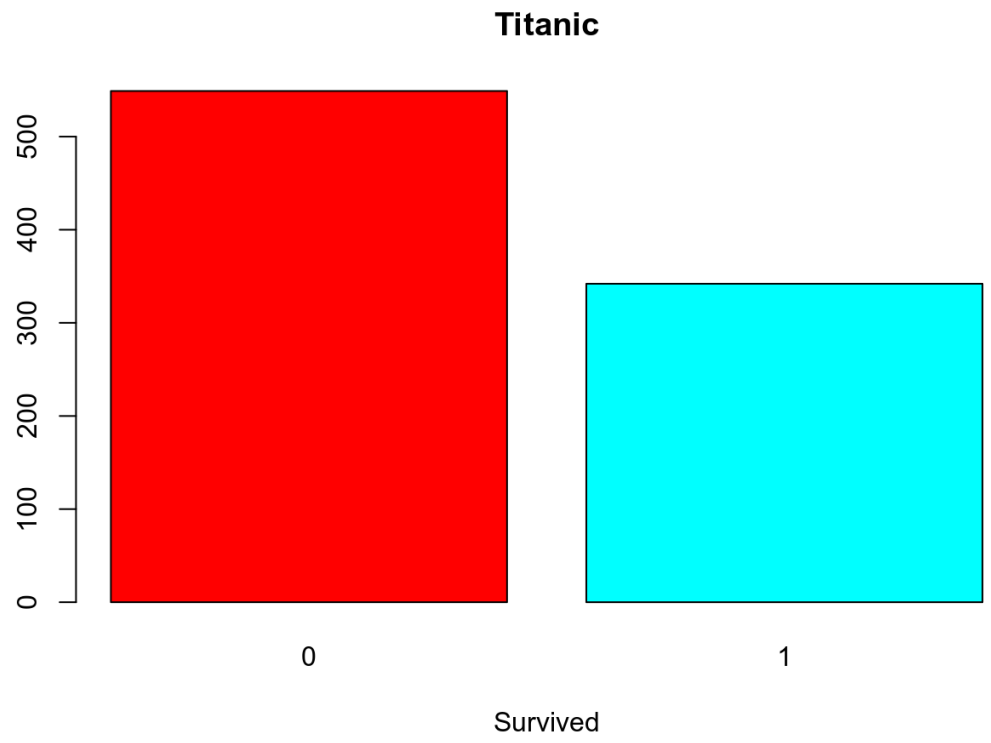
```
length(unique(datosEntrenamiento$PassengerId))
```

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

```
datosEntrenamiento <- subset(datosEntrenamiento, select = -c(PassengerId))
```

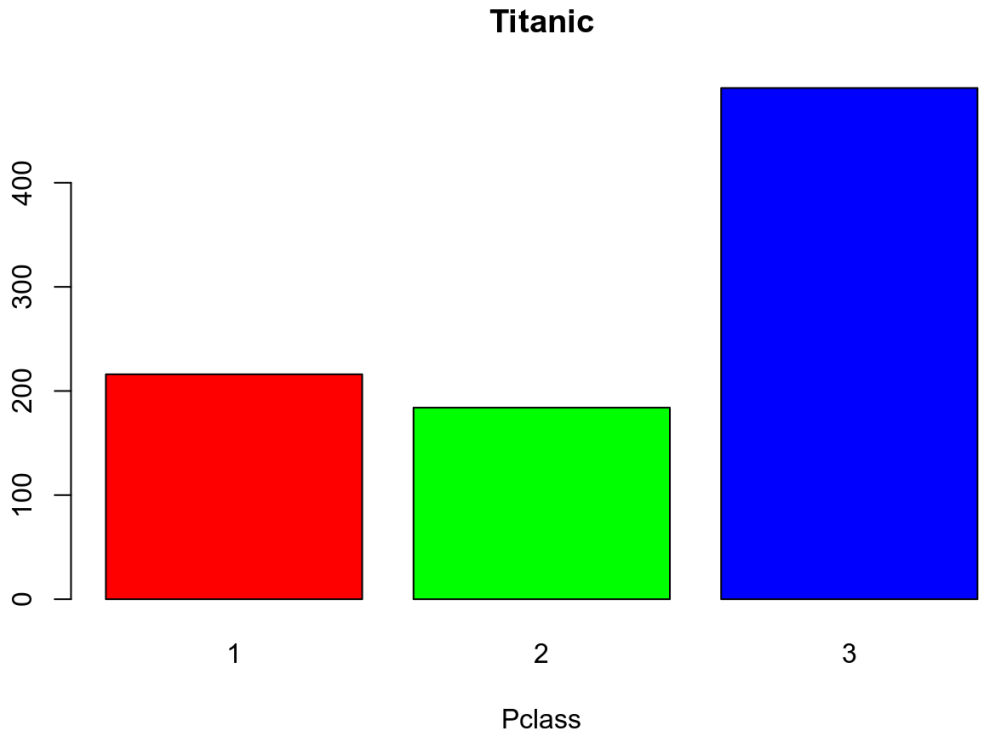
La variable `Survived` identifica si el pasajero sobrevivió (`1`) o no (`0`). En el conjunto de entrenamiento hay 342 supervivientes y 549 no supervivientes.

```
# Lo transformamos a factor
#datosEntrenamiento$Survived <- as.factor(datosEntrenamiento$Survived)
datosEntrenamiento$SurvivedFactor <- as.factor(datosEntrenamiento$Survived)
counts <- table(datosEntrenamiento$SurvivedFactor)
#counts
barplot(counts, main="Titanic",  xlab="Survived", col=rainbow(2))
```

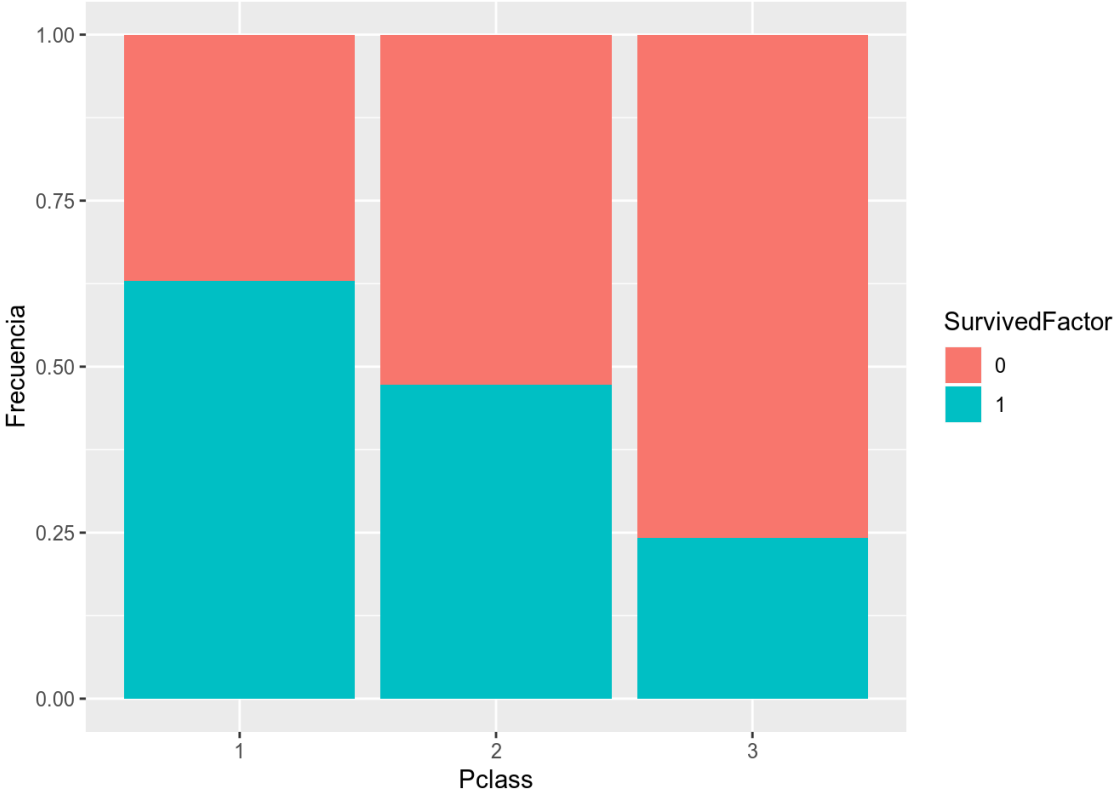


La varialble `Pclass` identifica si el pasajero viajaba en primera clase (`1`), segunda clase (`2`) o tercera clase (`3`). Podemos ver que en el conjunto de entrenamiento hay 216 pasajeros en primera clase, 184 pasajeros en segunda clase y 491 en tercera clase. Además podemos observar que el % de supervivientes es mucho mayor en cuanto mejor es la clase en la que viajaba el pasarejo.

```
# El test de Levene necesita que sea un valor numérico
datosEntrenamiento$Pclass <- as.factor(datosEntrenamiento$Pclass)
datosTest$Pclass <- as.factor(datosTest$Pclass)
counts <- table(datosEntrenamiento$Pclass)
#counts
barplot(counts, main="Titanic",  xlab="Pclass",col=rainbow(3))
```




```
ggplot(data=datosEntrenamiento[,],aes(x=`Pclass`,fill=`SurvivedFactor`))+geom_bar(position="f
ill")+ylab("Frecuencia")
```



La variable `Name` es el nombre del pasajero, por lo que hay 891 valores diferentes. No parece una variable a tener en cuenta para la construcción del modelo.

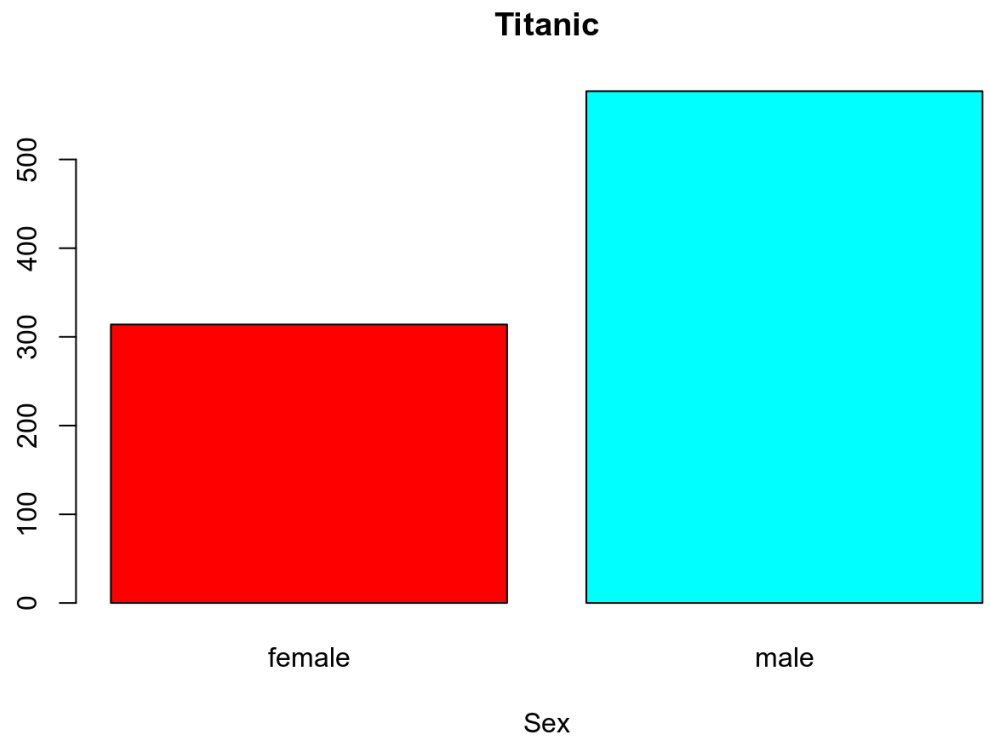
```
length(unique(datosEntrenamiento$Name))
```

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

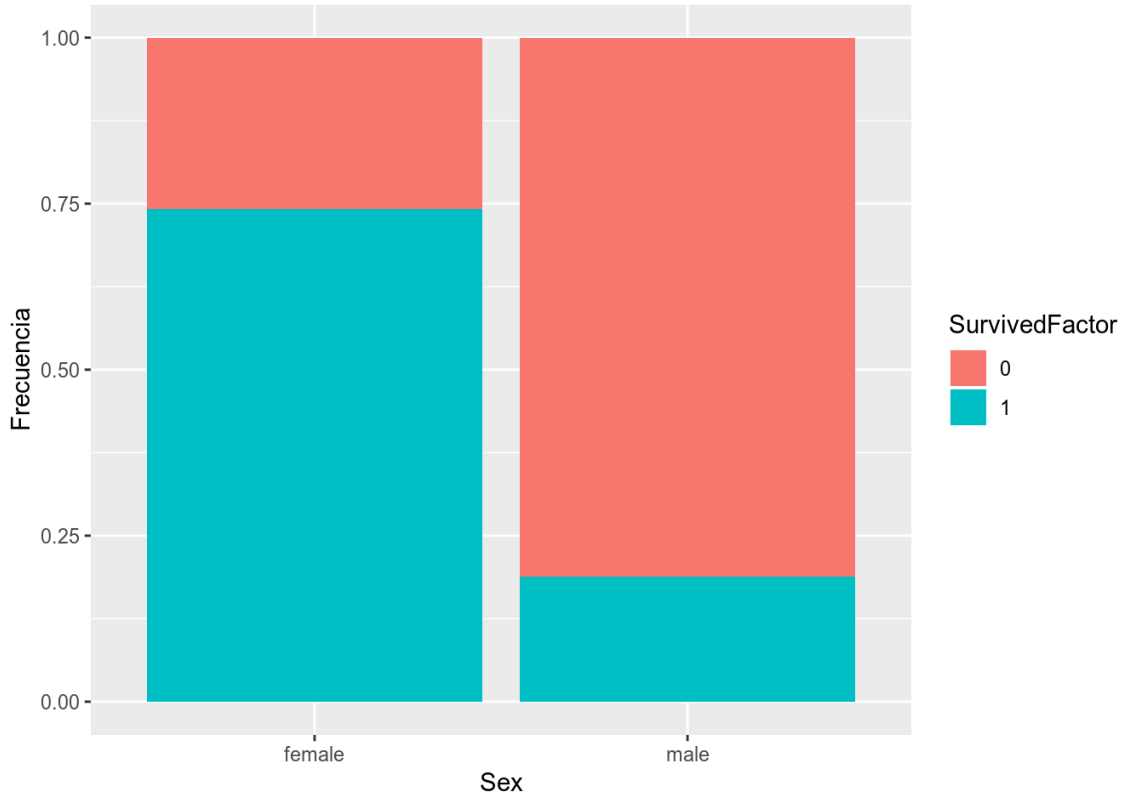
```
datosEntrenamiento <- subset(datosEntrenamiento, select = -c(Name))
```

La variable Sex identifica el género del pasajero. Vemos que hay 577 hombres y 314 mujeres. Además podemos ver que el % de supervivientes es mucho más alto para mujeres que para hombres.

```
counts <- table(datosEntrenamiento$Sex)
#counts
barplot(counts, main="Titanic",  xlab="Sex", col=rainbow(2))
```




```
ggplot(data=datosEntrenamiento[,],aes(x=`Sex`,fill=`SurvivedFactor`))+geom_bar(position="fil
l")+ylab("Frecuencia")
```



La variable `Age` identifica la edad del pasajero. Vemos que el mínimo es de `0.42` años y el máximo de `80` con una mediana de `28` y una media de `29.7`. Además podemos ver que hay `177` pasajeros con edad `NA`, con lo que se desconoce su edad.

```
summary(datosEntrenamiento$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.42   20.12   28.00   29.70   38.00   80.00     177
```

La variable `SibSp` identifica el número de familiares a bordo incluyendo marido/esposa y hermanos. Hay un mínimo de `0` y un máximo de `8` con una mediana de `0.0` y una media de `0.523`.

```
summary(datosEntrenamiento$SibSp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   0.523   1.000   8.000
```

La variable `Parch` identifica el número de familiares a bordo incluyendo padres e hijos. Hay un mínimo de `0` y un máximo de `6` con una mediana de `0.0` y una media de `0.3816`.

```
summary(datosEntrenamiento$Parch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3816  0.0000  6.0000
```

La variable `Ticket` identifica el número de serie del tiquet que utilizó el pasajero para acceder. Hay muchísimos valores diferentes que no parecen aportar gran información por lo que lo eliminaremos.

```
length(unique(datosEntrenamiento$Ticket))
```

```
## [1] 681
```

```
datosEntrenamiento <- subset(datosEntrenamiento, select = -c(Ticket))
```

La variable `Fare` identifica la tarifa pagada por el pasajero. Hay un mínimo de `0.0` que puede identificar tanto polizones como datos desconocidos y que corresponde a 15 pasajeros. El máximo es `512.33` . La mediana es de `14.45` y la media es de `32.20` . Viendo los pasajeros correspondientes a la mayor tarifa podemos ver que pertenecen al mismo ticket.

```
summary(datosEntrenamiento$Fare)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    7.91   14.45   32.20   31.00  512.33
```

```
length(datosEntrenamiento$Fare[datosEntrenamiento$Fare==0.0])
```

```
## [1] 15
```

La variable `Cabin` muestra los camarotes asignados a la persona. Tiene 148 valores diferentes y, además, tiene 687 pasajeros con el valor en blanco, con lo que no parece una variable a tener en cuenta para la construcción de modelos.

```
length(unique(datosEntrenamiento$Cabin))
```

```
## [1] 148
```

```
sum(datosEntrenamiento$Cabin == "")
```

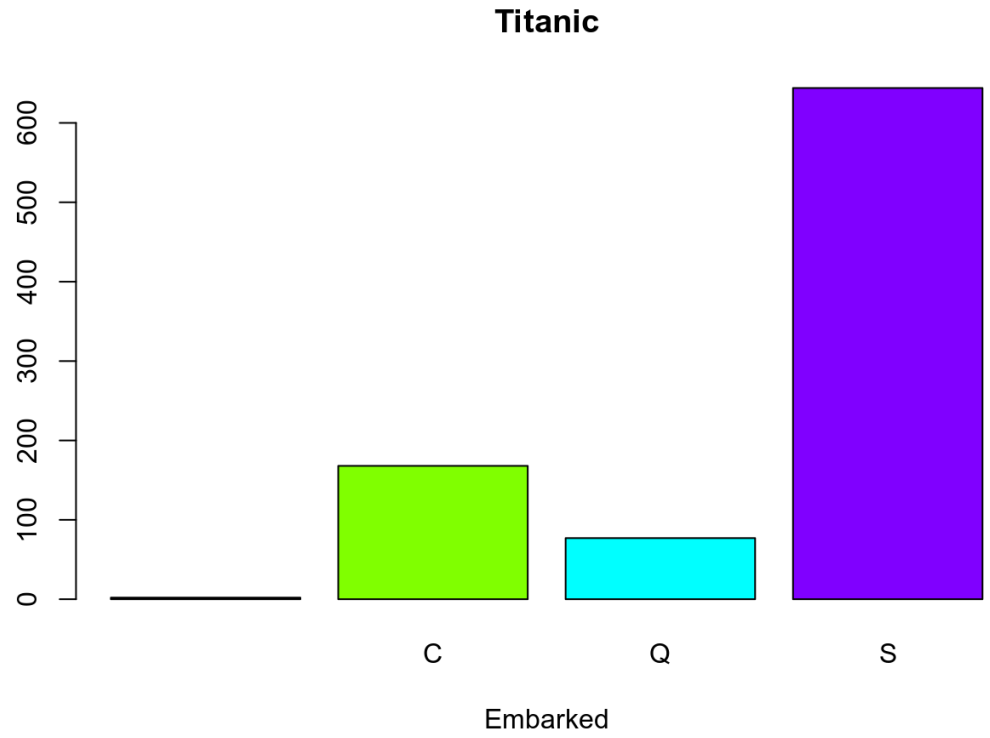
```
## [1] 687
```

```
datosEntrenamiento <- subset(datosEntrenamiento, select = -c(Cabin))
```

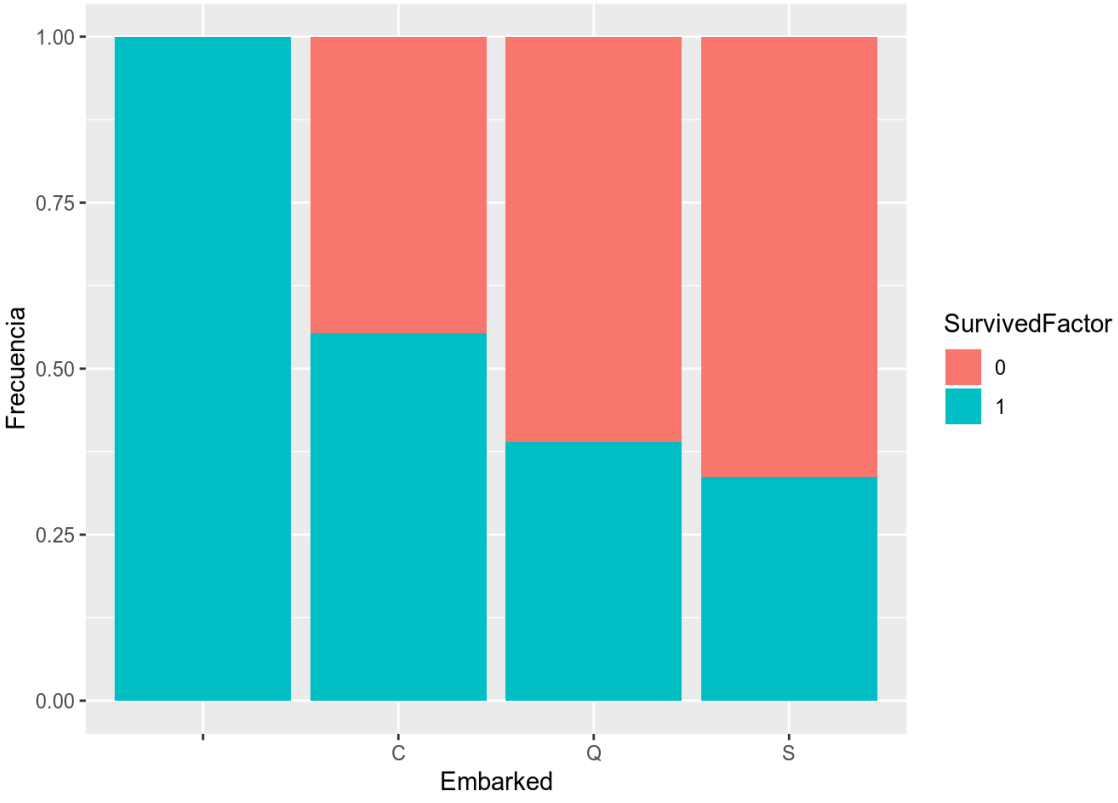
La variable `Embarked` especifica el puerto en el que el pasajero embarcó en el Titanic, siendo `S` el puerto inglés de `Southhampton` con 644 pasajeros embarcados para partir al puerto `C` identificando al puerto francés de `Cherbourg` con 168 pasajeros embarcados para ir finalmente al puerto `Q` identificando el puerto irlandés de `Queenstown` con 77 pasajeros embarcados. Además vemos que hay dos pasajeros de los que se desconoce el puerto de embarque.

Podemos ver que los % de supervivencia varían dependiendo del puerto, por lo que puede ser un valor importante.

```
counts <- table(datosEntrenamiento$Embarked)
#counts
barplot(counts, main="Titanic",  xlab="Embarked", col=rainbow(4))
```




```
ggplot(data=datosEntrenamiento[,],aes(x=`Embarked`,fill=`SurvivedFactor`))+geom_bar(position="fill")+ylab("Frecuencia")
```



4 Limpieza de los datos

¿Los datos contienen ceros o elementos vacíos?

La variable `Survived` contiene valores `0`, identificando aquellos pasajeros que no sobrevivieron.

```
length(datosEntrenamiento$Survived[datosEntrenamiento$Survived==0])
```

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

La variable `Age` contiene 177 valores `NA` para aquellos pasajeros de los que se desconoce su edad.

```
length(datosEntrenamiento$Age[is.na(datosEntrenamiento$Age)])
```

```
## [1] 177
```

La variable `SibSp` contiene valores `0`, identificando aquellos pasajeros con cero familiares a bordo.

```
length(datosEntrenamiento$SibSp[datosEntrenamiento$SibSp==0])
```

```
## [1] 608
```

La variable `Parch` contiene valores `0`, identificando aquellos pasajeros con cero hijos/parientes a bordo.

```
length(datosEntrenamiento$Parch[datosEntrenamiento$Parch==0])
```

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

La variable `Fare` contiene valores `0` señalando a polizones o a pasajeros de los que se desconoce qué precio pagaron.

```
length(datosEntrenamiento$Fare[datosEntrenamiento$Fare==0.0])
```

```
## [1] 15
```

La variable `Embarked` contiene 2 valores en blanco, por lo que se desconoce en qué puerto embarcaron dichos pasajeros.

```
length(datosEntrenamiento$Embarked[datosEntrenamiento$Embarked==""])
```

```
## [1] 2
```

¿Cómo gestionarías cada uno de estos casos?

Para las variables `Survived`, `SibSp` y `Parch` el valor `0` es perfectamente normal con lo que hay que hacer ningún tratamiento.

La variable `Age` contiene 177 pasajeros con valor `NA`. Para tratar este caso se sustituirán dichos valores por la edad calculada mediante KNN.

```
datosEntrenamiento$Age <- kNN(datosEntrenamiento)$Age
```

La variable `Fare` contiene 15 pasajeros con valor `0`, lo que no es habitual. Para tratar este caso se sustituirán dichos valores por el valor calculado por kNN.

```
datosEntrenamiento$Fare[datosEntrenamiento$Fare==0.0] <- NA
datosEntrenamiento$Fare <- kNN(datosEntrenamiento)$Fare
datosTest$Fare[datosTest$Fare==0.0] <- NA
datosTest$Fare <- kNN(datosTest)$Fare
```

La variable `Embarked` contiene 2 valores en blanco que serán sustituidos por el valor más frecuente para dicha variable

```
tableMax <- table(datosEntrenamiento$Embarked)
maxEmbarked <- names(tableMax)[which.max(tableMax)]
datosEntrenamiento$Embarked[datosEntrenamiento$Embarked==""] <- maxEmbarked
length(datosEntrenamiento$Embarked[datosEntrenamiento$Embarked==""])
```

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

Identificación y tratamiento de valores extremos.

La variable `Age` contiene 2 personas que están fuera del rango de la media más tres desviaciones típicas. Aunque estos valores son extremos los valores en si parecen razonables (74-80 años) por lo que los dejaremos tal cual.

```
meanAge <- mean(datosEntrenamiento$Age)
sdAge <- sd(datosEntrenamiento$Age)
length(datosEntrenamiento$Age[ (datosEntrenamiento$Age > (meanAge + (3*sdAge)))|datosEntrenam
iento$Age < (meanAge - (3*sdAge))  ]  )
```

```
## [1] 2
```

```
datosEntrenamiento$Age[ (datosEntrenamiento$Age > (meanAge + (3*sdAge)))|datosEntrenamiento$A
ge < (meanAge - (3*sdAge))  ]
```

```
## [1] 80 74
```

La variable `SibSp` contiene 30 personas que tienen un valor fuera del rango de la media más tres desviaciones típicas (4-8 parientes), siendo los 7 hermanos de la familia `Sage` . Aunque los valores son extremos, no hay razón para modificarlos en este momento.

```
meanSibSp <- mean(datosEntrenamiento$SibSp)
#meanSibSp
sdSibSp <- sd(datosEntrenamiento$SibSp)
#sdSibSp
length(datosEntrenamiento$SibSp[ (datosEntrenamiento$SibSp > (meanSibSp + (3*sdSibSp)))|datos
Entrenamiento$SibSp < (meanSibSp - (3*sdSibSp))  ]  )
```

```
## [1] 30
```

```
datosEntrenamiento$SibSp[ (datosEntrenamiento$SibSp > (meanSibSp + (3*sdSibSp)))|datosEntrena
miento$SibSp < (meanSibSp - (3*sdSibSp))  ]
```

```
##  [1] 4 4 5 4 5 4 8 4 4 8 4 8 4 4 4 4 8 5 5 4 4 5 4 4 8 4 4 8 4 8
```

De igual forma, la variable `Parch` contiene 15 valores que tienen un valor fuera del rango de la media más tres desviaciones típicas (3-6 parientes). Aunque los valores son extremos no hay razón para modificarlos en este momento.

```
meanParch <- mean(datosEntrenamiento$Parch)
#meanParch
sdParch <- sd(datosEntrenamiento$Parch)
#sdParch
length(datosEntrenamiento$Parch[ (datosEntrenamiento$Parch > (meanParch + (3*sdParch)))|datos
Entrenamiento$Parch < (meanParch - (3*sdParch))  ]  )
```

```
## [1] 15
```

```
datosEntrenamiento$Parch[ (datosEntrenamiento$Parch > (meanParch + (3*sdParch)))|datosEntrena
miento$Parch < (meanParch - (3*sdParch))  ]
```

```
##  [1] 5 5 3 4 4 3 4 4 5 5 6 3 3 3 5
```

Igualmente, la variable `Fare` contiene 20 pasajeros que tienen un valor fuera del ranto de la media más tres desviaciones típicas (211.3375-512.3292). Aunque los valores son extremos no hay razón para modificarlos en este momento.

```
meanFare <- mean(datosEntrenamiento$Fare)
#meanFare
sdFare <- sd(datosEntrenamiento$Fare)
#sdFare
length(datosEntrenamiento$Fare[ (datosEntrenamiento$Fare > (meanFare + (3*sdFare)))|datosEntr
enamiento$Fare < (meanFare - (3*sdFare))  ]  )
```

```
## [1] 20
```

```
datosEntrenamiento$Fare[ (datosEntrenamiento$Fare > (meanFare + (3*sdFare)))|datosEntrenamien
to$Fare < (meanFare - (3*sdFare))  ]
```

```
##  [1] 263.0000 263.0000 247.5208 512.3292 247.5208 262.3750 263.0000
##  [8] 211.5000 227.5250 263.0000 221.7792 227.5250 512.3292 211.3375
## [15] 227.5250 227.5250 211.3375 512.3292 262.3750 211.3375
```

5 Análisis de los datos

Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

Parece interesante considerar los siguientes grupos para el análisis.

- Personas divididas por género.

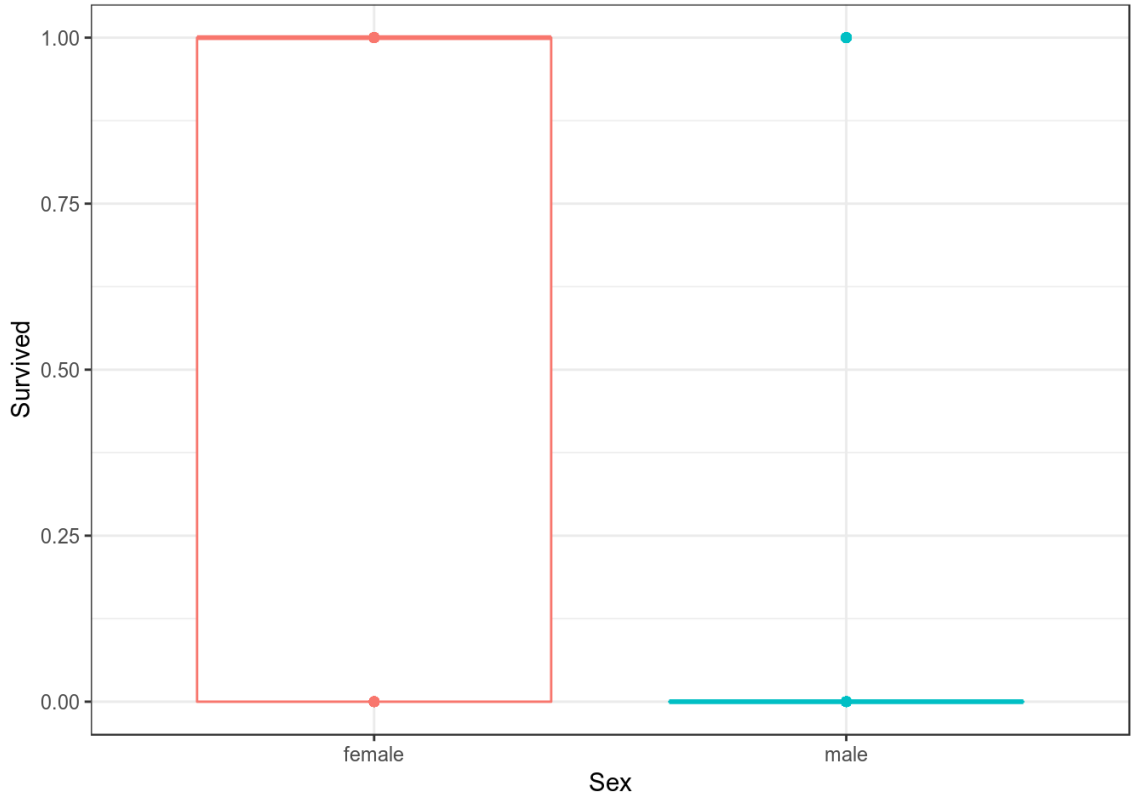
```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Sex=="male"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.1889  0.0000  1.0000
```

```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Sex=="female"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   0.742   1.000   1.000
```

```
ggplot(data = datosEntrenamiento, aes(x = Sex, y = Survived, colour = Sex)) +  geom_boxplot()
+  geom_point() +  theme_bw() +  theme(legend.position = "none")
```



- Personas divididas clase.

```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Pclass==1])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  1.0000  0.6296  1.0000  1.0000
```

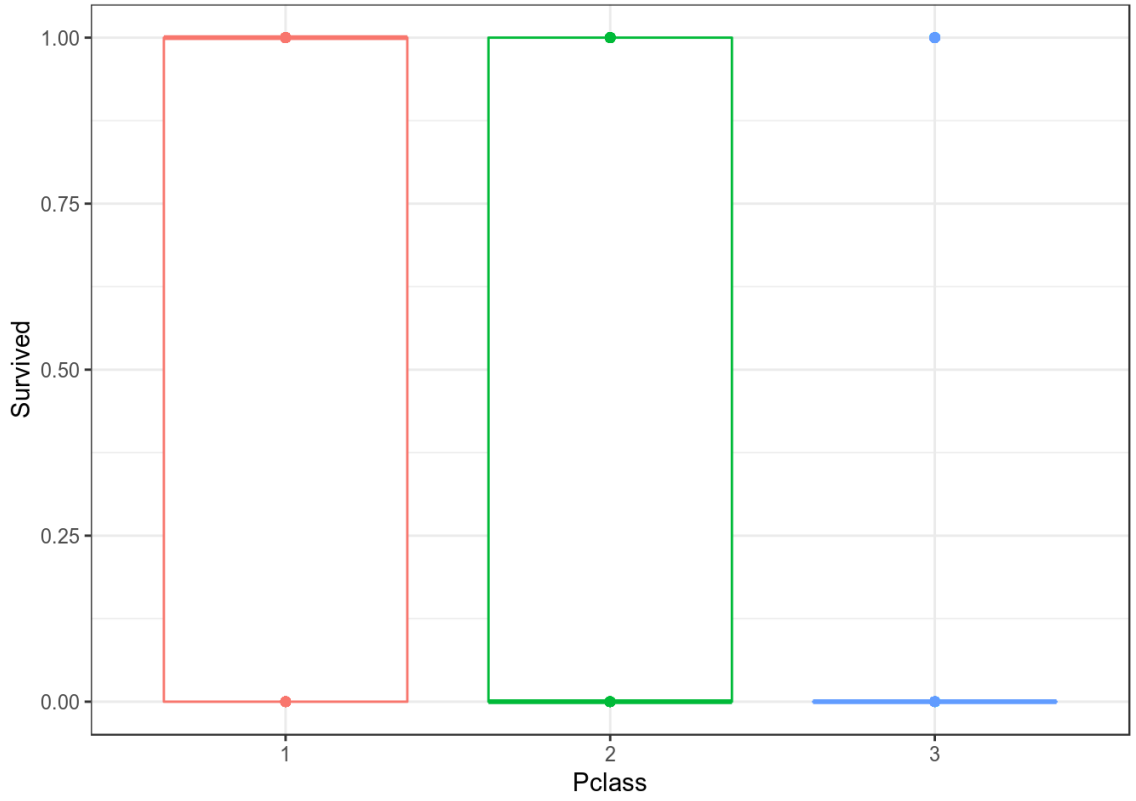
```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Pclass==2])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.4728  1.0000  1.0000
```

```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Pclass==3])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.2424  0.0000  1.0000
```

```
ggplot(data = datosEntrenamiento, aes(x = Pclass, y = Survived, colour = Pclass)) +  geom_box
plot() +  geom_point() +  theme_bw() +  theme(legend.position = "none")
```



- Personas divididas por puerto de embarque.

```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Embarked=="C"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  1.0000  0.5536  1.0000  1.0000
```

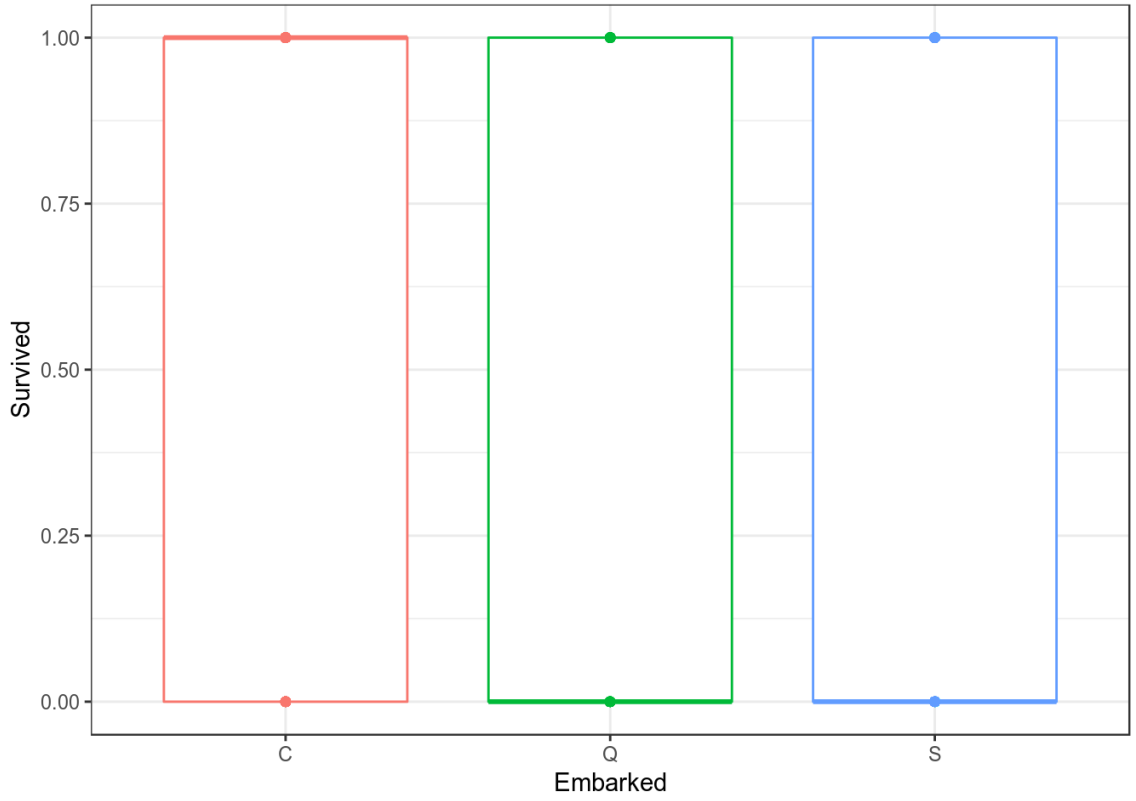
```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Embarked=="Q"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3896  1.0000  1.0000
```

```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Embarked=="S"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   0.339   1.000   1.000
```

```
ggplot(data = datosEntrenamiento, aes(x = Embarked, y = Survived, colour = Embarked)) +  geom
_boxplot() +  geom_point() +  theme_bw() +  theme(legend.position = "none")
```



- Personas divididas por tener o no familia.

```
datosEntrenamiento$Familia <- 0
datosEntrenamiento$Familia[datosEntrenamiento$Parch>0] <- 1
datosEntrenamiento$Familia[datosEntrenamiento$SibSp>0] <- 1
datosEntrenamiento$Familia <- as.integer(datosEntrenamiento$Familia)
datosEntrenamiento$FamiliaFactor <- as.factor(datosEntrenamiento$Familia)
#str(datosEntrenamiento)
datosTest$Familia <- 0
datosTest$Familia[datosTest$Parch>0] <- 1
datosTest$Familia[datosTest$SibSp>0] <- 1
datosTest$Familia <- as.integer(datosTest$Familia)
```

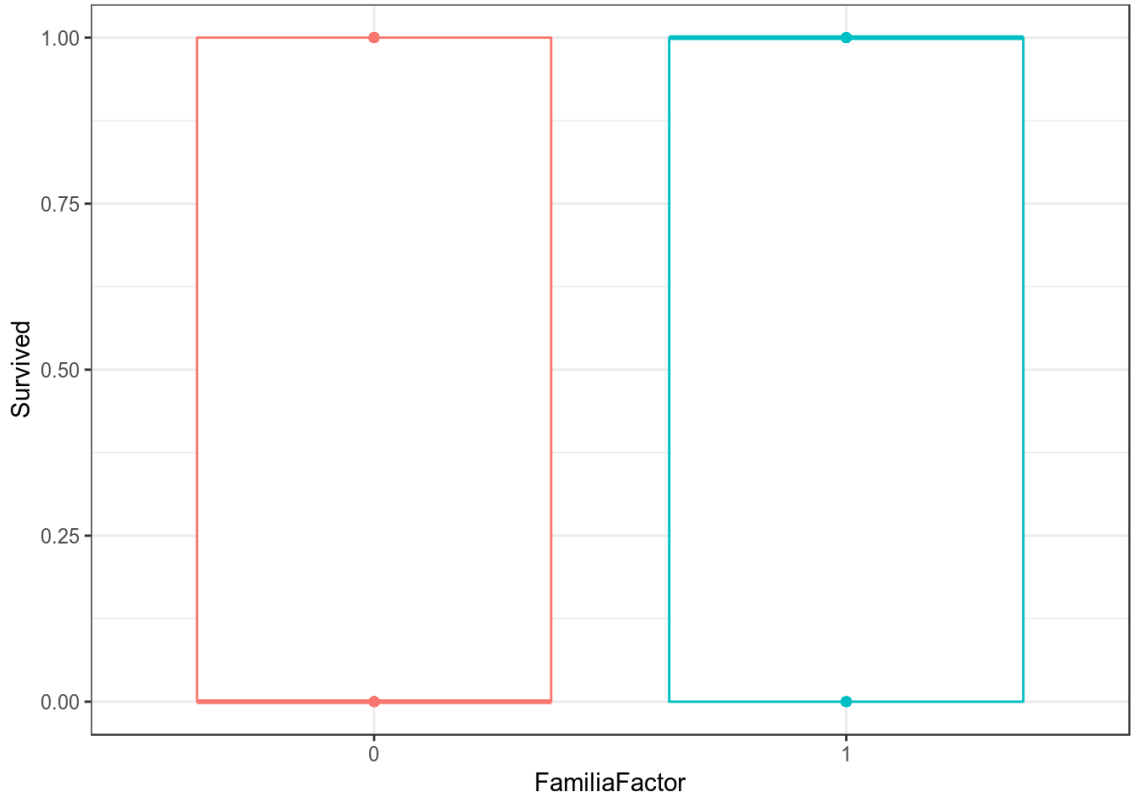
```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Familia==0])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3035  1.0000  1.0000
```

```
summary(datosEntrenamiento$Survived[datosEntrenamiento$Familia==1])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  1.0000  0.5056  1.0000  1.0000
```

```
ggplot(data = datosEntrenamiento, aes(x = FamiliaFactor, y = Survived, colour = FamiliaFacto
r)) +  geom_boxplot() +  geom_point() +  theme_bw() +  theme(legend.position = "none")
```



Comprobación de la normalidad y homogeneidad de la varianza.

En primer lugar comprobaremos si las variables cuantitativas provienen de una población distribuida normalmente. Para ello se usará el test de Anderson-Darling. De esta forma se comprobará que cada variable obtiene un p-valor superior al nivel de significación $\alpha = 0.05$. Si así fuera la variable sigue una distribución normal

```
alpha = 0.05
col.names = colnames(datosEntrenamiento)

for (i in 1:ncol(datosEntrenamiento)) {
    if (is.integer(datosEntrenamiento[,i]) | is.numeric(datosEntrenamiento[,i])) {
        p_val = ad.test(datosEntrenamiento[,i])$p.value
        if (p_val < alpha) {
            cat(col.names[i])
            cat(" NO sigue una distribución normal\n")
        }else{
            cat(col.names[i])
            cat(" SI sigue una distribución normal\n")
        }
    }
}
```

```
## Survived NO sigue una distribución normal
## Age NO sigue una distribución normal
## SibSp NO sigue una distribución normal
## Parch NO sigue una distribución normal
## Fare NO sigue una distribución normal
## Familia NO sigue una distribución normal
```

Es decir, ninguna variable sigue una distribución normal.

Además comprobaremos la homogeneidad de varianzas entre poblaciones usando el test de Fligner-Killeen. En primer lugar comprobaremos si la varianza de Survived en la población masculina es la misma que la población femenina, siendo esta la hipótesis nula.

```
fligner.test(Survived ~ Sex, data = datosEntrenamiento)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Sex
## Fligner-Killeen:med chi-squared = 5.7729, df = 1, p-value =
## 0.01627
```

Obtenemos un valor de `0.01627` , inferior a $\alpha =$ `0.05` por lo que rechazamos la hipótesis de que las varianzas de ambas muestras son homogéneas.

De igual forma comprobaremos si la varianza de las poblaciones divididas por la clase del camarote es la misma, siendo esta la hipótesis nula.

```
fligner.test(Survived ~ Pclass, data = datosEntrenamiento)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Pclass
## Fligner-Killeen:med chi-squared = 35.766, df = 2, p-value =
## 1.712e-08
```

Obtenemos un valor de `1.712e-08` , inferior a $\alpha =$ `0.05` por lo que rechazamos la hipótesis de que las varianzas de las muestras son homogéneas.

Repetiremos el proceso para comprobar si la varianza de las poblaciones divididas por el puerto de embarque es la misma, siendo esta la hipótesis nula.

```
fligner.test(Survived ~ Embarked, data = datosEntrenamiento)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by Embarked
## Fligner-Killeen:med chi-squared = 6.8863, df = 2, p-value =
## 0.03196
```

Obtenemos un valor de `0.03196` , inferior a $\alpha =$ `0.05` por lo que rechazamos la hipótesis de que las varianzas de las muestras son homogéneas.

Por ultimo analizaremos la varianza de las poblaciones divididas dependiendo de si tienen o no familia. La hipótesis nula será que son iguales.

```
fligner.test(Survived ~ FamiliaFactor, data = datosEntrenamiento)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Survived by FamiliaFactor
## Fligner-Killeen:med chi-squared = 32.956, df = 1, p-value =
## 9.426e-09
```

Obtenemos un valor de `9.426e-09` , inferior a $\alpha =$ `0.05` por lo que rechazamos la hipótesis de que las varianzas de las muestras son homogéneas.

Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc.

Para empezar estudiaremos la correlación entre las variables usando el coeficiente de correlación de Spearman debido a que hemos visto que los datos no se ajustan a una distribución normal.

```
datosEntrenamiento$SexInt <- as.numeric(datosEntrenamiento$Sex)
datosEntrenamiento$PclassInt <- as.numeric(datosEntrenamiento$Pclass)
```

```
corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")

for (i in 1:(ncol(datosEntrenamiento))) {
    if (is.integer(datosEntrenamiento[,i]) | is.numeric(datosEntrenamiento[,i]) ) {
        spearman_test = cor.test(datosEntrenamiento[,i],datosEntrenamiento$Survived,method =
"spearman",exact=FALSE)
        corr_coef = spearman_test$estimate
        p_val = spearman_test$p.value
        pair = matrix(ncol = 2, nrow = 1)
        pair[1][1] = corr_coef
        pair[2][1] = p_val
        corr_matrix <- rbind(corr_matrix, pair)
        rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(datosEntrenamiento)[i]
    }
}
corr_matrix
```

```
##              estimate      p-value
## Survived   1.00000000 0.000000e+00
## Age       -0.08232837 1.396373e-02
## SibSp      0.08887948 7.941431e-03
## Parch      0.13826563 3.453591e-05
## Fare       0.31428332 7.067913e-22
## Familia    0.20336709 9.009490e-10
## SexInt    -0.54335138 1.406066e-69
## PclassInt -0.33966794 1.687608e-25
```

Por ello podemos ver que las variables que más correlación tienen con `Survived` son, por este orden:

- `Sex`
- `Pclass`
- `Fare`
- `Familia`

A continuación ejecutaremos una serie de contrastes. En primer lugar dividiremos los datos en dos muestras dependiendo del género del pasajero. La hipótesis nula es que la media de `Survived` es la misma, siendo la hipótesis alternativa que la media de `Survived` es inferior.

```
t.test(datosEntrenamiento$Survived[datosEntrenamiento$Sex=="male"],datosEntrenamiento$Survive
d[datosEntrenamiento$Sex=="female"],alternative="less")
```

```
##
##  Welch Two Sample t-test
##
## data:  datosEntrenamiento$Survived[datosEntrenamiento$Sex == "male"] and datosEntrenamient
o$Survived[datosEntrenamiento$Sex == "female"]
## t = -18.672, df = 584.43, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.5043259
## sample estimates:
## mean of x mean of y
## 0.1889081 0.7420382
```

Se obtiene un p-valor de `2.2e-16`, inferior al nivel de significación $\alpha =$ `0.05` por lo que rechazamos la hipótesis nula y concluimos que la media de supervivencia de los pasajeros de género masculino es inferior.

Repetiremos este test para las muestras dependiendo de si pertenecen a la `Pclass 3` o no. La hipotesis nula es que la media de `Survived` es la misma, siendo la hipotesis alternativa que la media de `Survived` es inferior.

```
t.test(datosEntrenamiento$Survived[datosEntrenamiento$Pclass==3],datosEntrenamiento$Survived
[datosEntrenamiento$Pclass!=3],alternative="less")
```

```
##
##  Welch Two Sample t-test
##
## data:  datosEntrenamiento$Survived[datosEntrenamiento$Pclass == 3] and datosEntrenamiento
$Survived[datosEntrenamiento$Pclass != 3]
## t = -10, df = 792.25, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.2632438
## sample estimates:
## mean of x mean of y
## 0.2423625 0.5575000
```

Se obtiene un p-valor de `2.2e-16`, inferior al nivel de significación $\alpha =$ `0.05` por lo que rechazamos la hipótesis nula y concluimos que la media de supervivencia de los pasajeros de tercera clase es inferior al resto.

De nuevo comprobaremos si la supervivencia de los pasajeros de segunda clase es inferior a la supervivencia de los pasajeros de primera clase. La hipótesis nula es que la media de `Survived` es la misma, siendo la hipótesis alternativa que es inferior.

```
t.test(datosEntrenamiento$Survived[datosEntrenamiento$Pclass==2],datosEntrenamiento$Survived
[datosEntrenamiento$Pclass==1],alternative="less")
```

```
##
##  Welch Two Sample t-test
##
## data:  datosEntrenamiento$Survived[datosEntrenamiento$Pclass == 2] and datosEntrenamiento
$Survived[datosEntrenamiento$Pclass == 1]
## t = -3.17, df = 383.5, p-value = 0.0008233
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##         -Inf -0.07524497
## sample estimates:
## mean of x mean of y
## 0.4728261 0.6296296
```

Se obtiene un p-valor de `0.0008233`, inferior al nivel de significación $\alpha =$ `0.05` por lo que rechazamos la hipótesis nula y concluimos que la media de supervivencia de los pasajeros de segunda clase es inferior a la supervivencia de los pasajeros de primera clase.

Además comprobaremos si la supervivencia de los pasajeros embarcados en el puerto de `Cherbourg` es igual o no al resto de pasajeros. Por tanto la hipótesis nula es que la media de `Survived` es la misma, siendo la hipótesis alterativa que es inferior.

```
t.test(datosEntrenamiento$Survived[datosEntrenamiento$Embarked=="C"],datosEntrenamiento$Survi
ved[datosEntrenamiento$Embarked!="C"],alternative="less")
```

```
##
##  Welch Two Sample t-test
##
## data:  datosEntrenamiento$Survived[datosEntrenamiento$Embarked == "C"] and datosEntrenamie
nto$Survived[datosEntrenamiento$Embarked != "C"]
## t = 4.9405, df = 242.54, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 0.2790807
## sample estimates:
## mean of x mean of y
## 0.5535714 0.3443983
```

Se obtiene un p-valor de `1` , superior al nivel de significación $\alpha =$ `0.05` por lo que no podemos rechazar la hipótesis nula y no podemos decir que la supervivencia de los pasajeros embarcados en el puerto de `Cherbourg` sea superior a la supervivencia de el resto de pasajeros.

Por último comprobaremos si la supervivencia de aquellos pasajeros sin familia es inferior a la supervivencia de los pasajeros con familia. Por ellos la hipótesis nula es que la media de `Survived` es la misma para los pasajeros con y sin familia y la hipótesis alternativa es que la media de `Survived` es inferior para los pasajeros sin familia.

```
t.test(datosEntrenamiento$Survived[datosEntrenamiento$Familia==0],datosEntrenamiento$Survived
[datosEntrenamiento$Familia!=0],alternative="less")
```

```
##
##  Welch Two Sample t-test
##
## data:  datosEntrenamiento$Survived[datosEntrenamiento$Familia == 0] and datosEntrenamiento
$Survived[datosEntrenamiento$Familia != 0]
## t = -6.0869, df = 710.56, p-value = 9.414e-10
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.1474238
## sample estimates:
## mean of x mean of y
## 0.3035382 0.5056497
```

Se obtiene un p-valor de `9.414e-10` , inferior al nivel de significación $\alpha =$ `0.05` por lo que podemos rechazar la hipótesis nula y podemos decir que la supervivencia de los pasajeros sin familia es inferior a la de los pasajeros con familia.

A continuación se generarán modelos basados en regresión. El primero, usado únicamente la variable `Sex` .

```
threshold <- 0.4
modelo1 <- lm(Survived ~ Sex ,data=datosEntrenamiento)
summary(modelo1)
```

```
##
## Call:
## lm(formula = Survived ~ Sex, data = datosEntrenamiento)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7420 -0.1889 -0.1889  0.2580  0.8111
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.74204    0.02307   32.17   <2e-16 ***
## Sexmale     -0.55313    0.02866  -19.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4087 on 889 degrees of freedom
## Multiple R-squared:  0.2952, Adjusted R-squared:  0.2944
## F-statistic: 372.4 on 1 and 889 DF,  p-value: < 2.2e-16
```

```
datosTest$valueModelo1 <- predict(modelo1,datosTest)
datosTest$resultadoModelo1 <- NA
datosTest$resultadoModelo1[datosTest$valueModelo1 > threshold] <- 1
datosTest$resultadoModelo1[datosTest$valueModelo1 <= threshold] <- 0
datosTest$resultadoModelo1Factor <- as.factor(datosTest$resultadoModelo1)
```

El segundo usando `Sex` y `Pclass`.

```
modelo2 <- lm(Survived ~ Sex + Pclass ,data=datosEntrenamiento)
summary(modelo2)
```

```
##
## Call:
## lm(formula = Survived ~ Sex + Pclass, data = datosEntrenamiento)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.92123 -0.25958 -0.09095  0.22415  0.90905
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.92123    0.03059  30.118  < 2e-16 ***
## Sexmale     -0.51627    0.02744 -18.814  < 2e-16 ***
## Pclass2     -0.14537    0.03889  -3.739 0.000197 ***
## Pclass3     -0.31401    0.03188  -9.849  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3876 on 887 degrees of freedom
## Multiple R-squared:  0.3678, Adjusted R-squared:  0.3656
## F-statistic:   172 on 3 and 887 DF,  p-value: < 2.2e-16
```

```
datosTest$valueModelo2 <- predict(modelo2,datosTest)
datosTest$resultadoModelo2 <- NA
datosTest$resultadoModelo2[datosTest$valueModelo2 > threshold] <- 1
datosTest$resultadoModelo2[datosTest$valueModelo2 <= threshold] <- 0
datosTest$resultadoModelo2Factor <- as.factor(datosTest$resultadoModelo2)
```

El tercero usando `Sex`, `Pclass` y `Fare`.

```
modelo3 <- lm(Survived ~ Sex + Pclass + Fare ,data=datosEntrenamiento)
summary(modelo3)
```

```
##
## Call:
## lm(formula = Survived ~ Sex + Pclass + Fare, data = datosEntrenamiento)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93621 -0.25883 -0.09036  0.22472  0.90984
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8985009  0.0429483  20.920  < 2e-16 ***
## Sexmale     -0.5132165  0.0277447 -18.498  < 2e-16 ***
## Pclass2     -0.1296851  0.0441101  -2.940  0.00337 **
## Pclass3     -0.2968611  0.0391662  -7.580 8.74e-14 ***
## Fare         0.0002488  0.0003300   0.754  0.45107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3877 on 886 degrees of freedom
## Multiple R-squared:  0.3682, Adjusted R-squared:  0.3653
## F-statistic: 129.1 on 4 and 886 DF,  p-value: < 2.2e-16
```

```
datosTest$valueModelo3 <- predict(modelo3,datosTest)
datosTest$resultadoModelo3 <- NA
datosTest$resultadoModelo3[datosTest$valueModelo3 > threshold] <- 1
datosTest$resultadoModelo3[datosTest$valueModelo3 <= threshold] <- 0
datosTest$resultadoModelo3Factor <- as.factor(datosTest$resultadoModelo3)
```

El cuarto usando `Sex` , `Pclass` , `Fare` y `Familia` .

```
modelo4 <- lm(Survived ~ Sex + Pclass + Fare + Familia ,data=datosEntrenamiento)
summary(modelo4)
```

```
##
## Call:
## lm(formula = Survived ~ Sex + Pclass + Fare + Familia, data = datosEntrenamiento)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93797 -0.25588 -0.08798  0.22087  0.91222
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8943300  0.0447011  20.007  < 2e-16 ***
## Sexmale     -0.5106283  0.0287920 -17.735  < 2e-16 ***
## Pclass2     -0.1307322  0.0442404  -2.955  0.00321 **
## Pclass3     -0.2974810  0.0392286  -7.583 8.51e-14 ***
## Fare         0.0002239  0.0003383   0.662  0.50826
## Familia      0.0097160  0.0286992   0.339  0.73503
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3878 on 885 degrees of freedom
## Multiple R-squared:  0.3683, Adjusted R-squared:  0.3647
## F-statistic: 103.2 on 5 and 885 DF,  p-value: < 2.2e-16
```

```
datosTest$valueModelo4 <- predict(modelo4,datosTest)
datosTest$resultadoModelo4 <- NA
datosTest$resultadoModelo4[datosTest$valueModelo4 > threshold] <- 1
datosTest$resultadoModelo4[datosTest$valueModelo4 <= threshold] <- 0
datosTest$resultadoModelo4Factor <- as.factor(datosTest$resultadoModelo4)
```

Y el quinto usando `Pclass` , `Fare` y `Familia` .

```
modelo5 <- lm(Survived ~ Pclass + Fare + Familia ,data=datosEntrenamiento)
summary(modelo5)
```

```
##
## Call:
## lm(formula = Survived ~ Pclass + Fare + Familia, data = datosEntrenamiento)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8279 -0.3457 -0.1889  0.4419  0.8117
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.4986911  0.0450724  11.064  < 2e-16 ***
## Pclass2     -0.1036310  0.0514457  -2.014   0.0443 *
## Pclass3     -0.3152465  0.0456300  -6.909 9.33e-12 ***
## Fare         0.0007011  0.0003923   1.787   0.0743 .
## Familia      0.1448612  0.0321946   4.500 7.72e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4513 on 886 degrees of freedom
## Multiple R-squared:  0.1437, Adjusted R-squared:  0.1399
## F-statistic: 37.18 on 4 and 886 DF,  p-value: < 2.2e-16
```

```
datosTest$valueModelo5 <- predict(modelo5,datosTest)
datosTest$resultadoModelo5 <- NA
datosTest$resultadoModelo5[datosTest$valueModelo5 > threshold] <- 1
datosTest$resultadoModelo5[datosTest$valueModelo5 <= threshold] <- 0
datosTest$resultadoModelo5Factor <- as.factor(datosTest$resultadoModelo5)
```

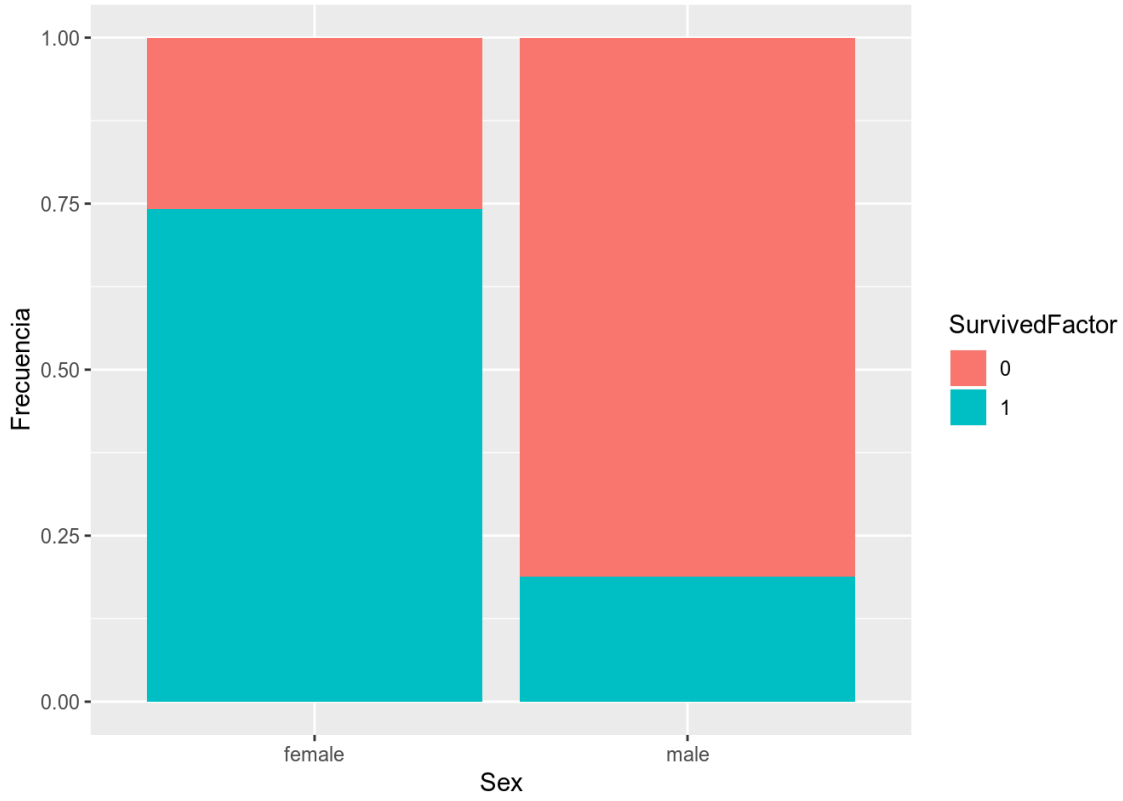
Se generan los ficheros para kaggle.

```
resultadosFileModelo1 = data.frame(PassengerId=datosTest$PassengerId,Survived=datosTest$resul
tadoModelo1)
resultadosFileModelo2 = data.frame(PassengerId=datosTest$PassengerId,Survived=datosTest$resul
tadoModelo2)
resultadosFileModelo3 = data.frame(PassengerId=datosTest$PassengerId,Survived=datosTest$resul
tadoModelo3)
resultadosFileModelo4 = data.frame(PassengerId=datosTest$PassengerId,Survived=datosTest$resul
tadoModelo4)
resultadosFileModelo5 = data.frame(PassengerId=datosTest$PassengerId,Survived=datosTest$resul
tadoModelo5)
write.csv(resultadosFileModelo1,file = "submission_modelo1.csv",row.names = F)
write.csv(resultadosFileModelo2,file = "submission_modelo2.csv",row.names = F)
write.csv(resultadosFileModelo3,file = "submission_modelo3.csv",row.names = F)
write.csv(resultadosFileModelo4,file = "submission_modelo4.csv",row.names = F)
write.csv(resultadosFileModelo5,file = "submission_modelo5.csv",row.names = F)
```

6 Representación de los resultados a partir de tablas y gráficas

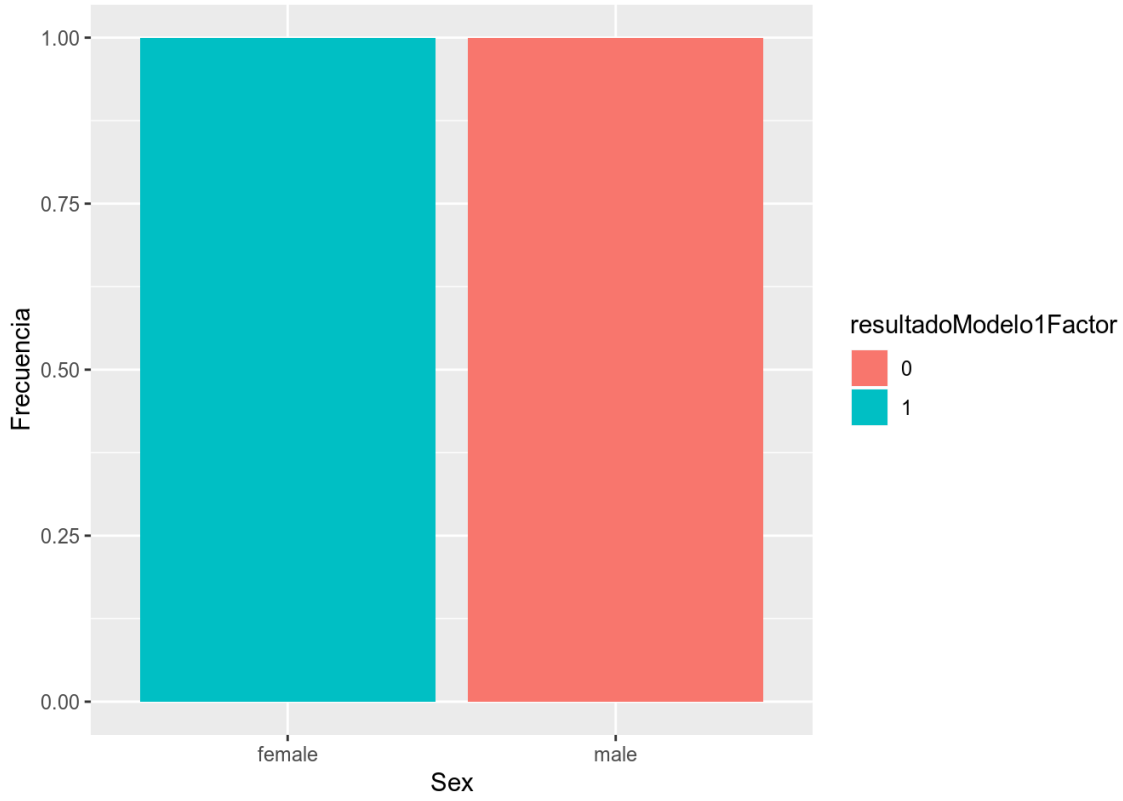
En primer lugar se repite la distribución de supervivientes en base a su genero.

```
ggplot(data=datosEntrenamiento[,],aes(x=`Sex`,fill=`SurvivedFactor`))+geom_bar(position="fil
l")+ylab("Frecuencia")
```

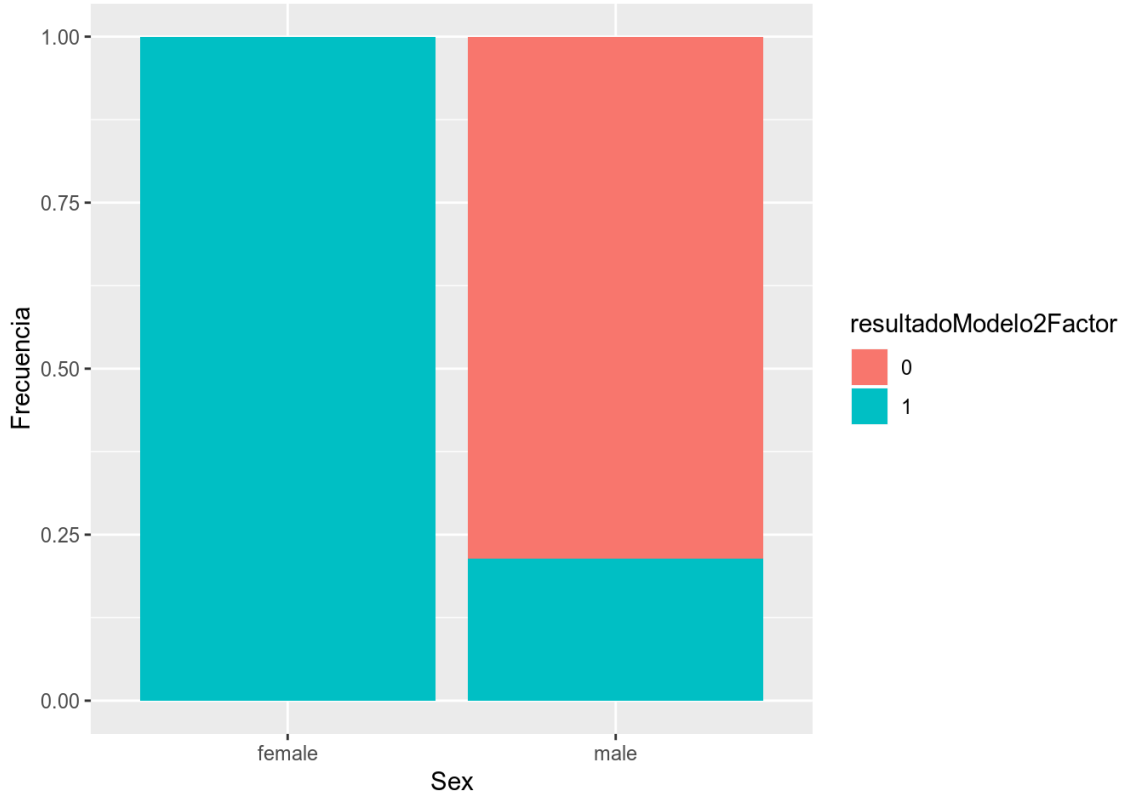


Para luego compararlo con los modelos generados. Los modelos generados no parecen seguir la misma tendencia.

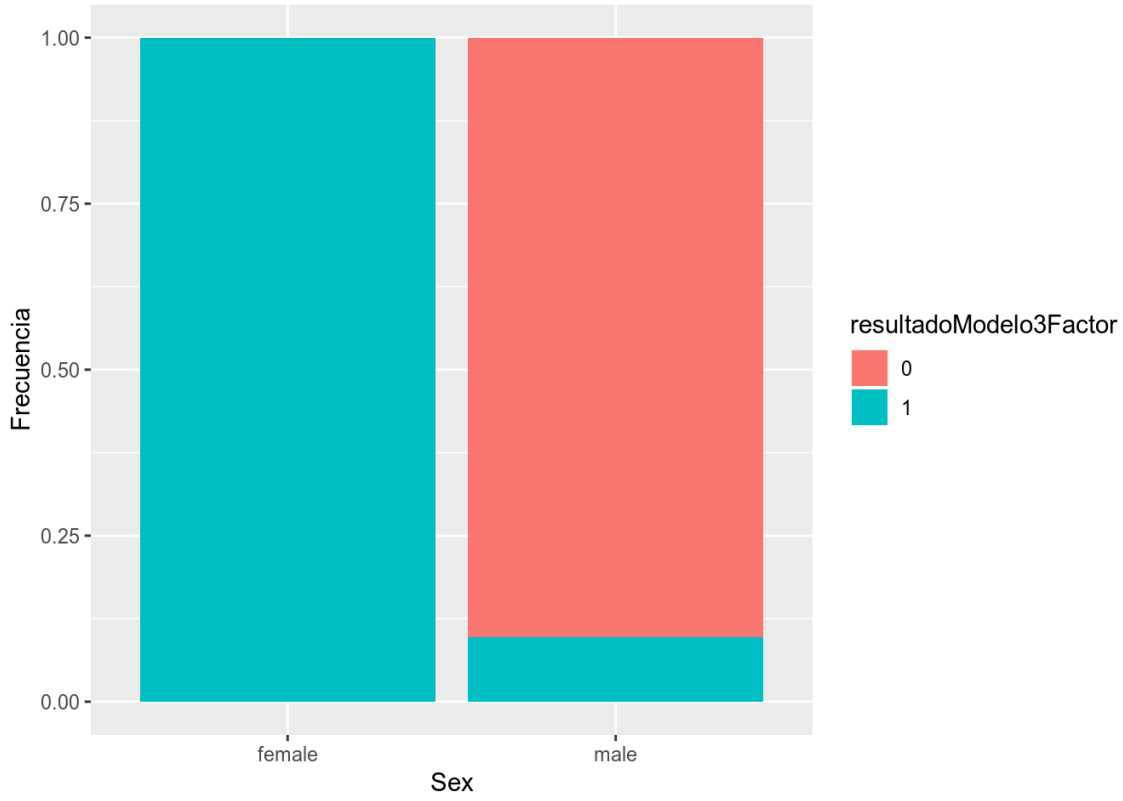
```
ggplot(data=datosTest[,],aes(x=`Sex`,fill=`resultadoModelo1Factor`))+geom_bar(position="fill"
)+ylab("Frecuencia")
```



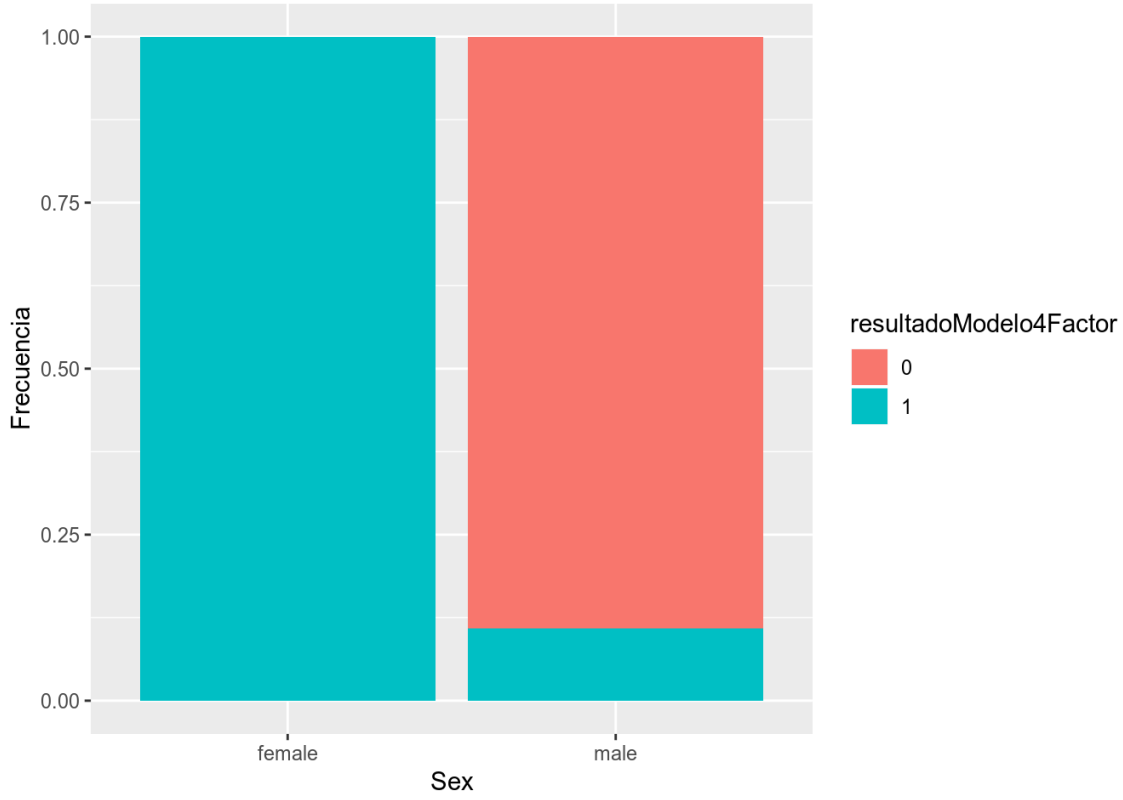

```
ggplot(data=datosTest[,],aes(x=`Sex`,fill=`resultadoModelo2Factor`))+geom_bar(position="fill"
)+ylab("Frecuencia")
```



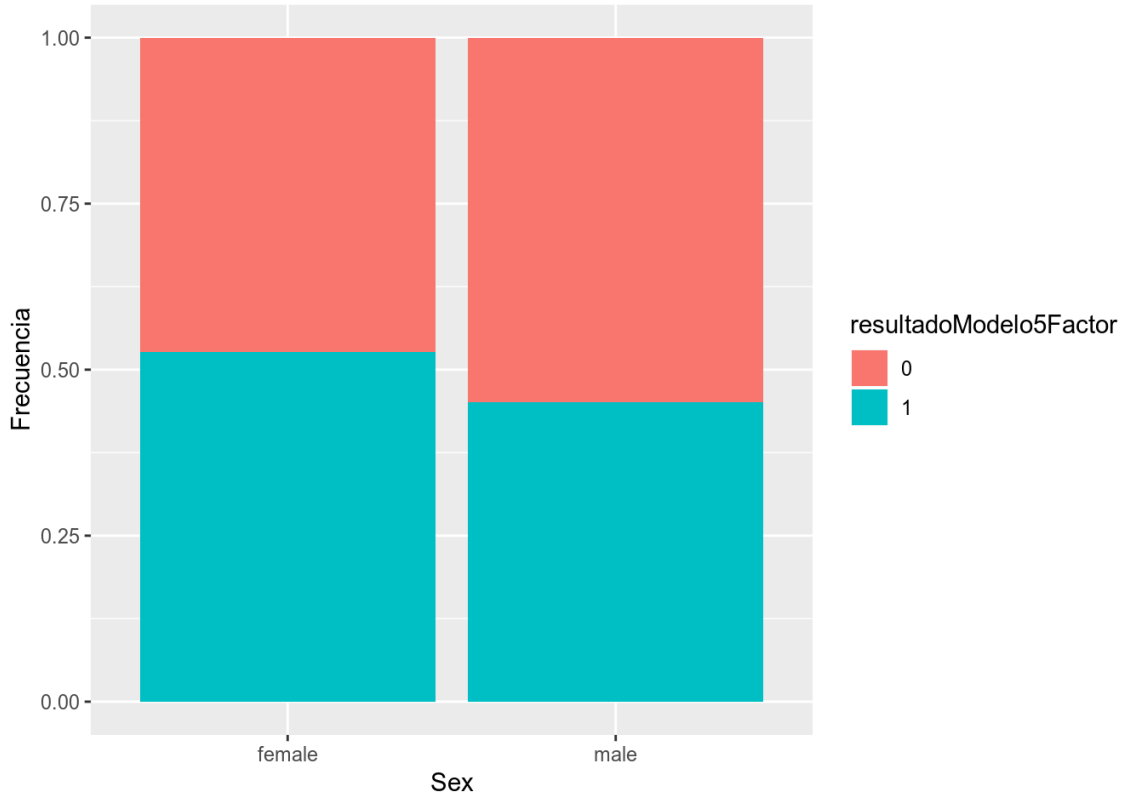

```
ggplot(data=datosTest[,],aes(x=`Sex`,fill=`resultadoModelo3Factor`))+geom_bar(position="fill"
)+ylab("Frecuencia")
```




```
ggplot(data=datosTest[,],aes(x=`Sex`,fill=`resultadoModelo4Factor`))+geom_bar(position="fill"
)+ylab("Frecuencia")
```

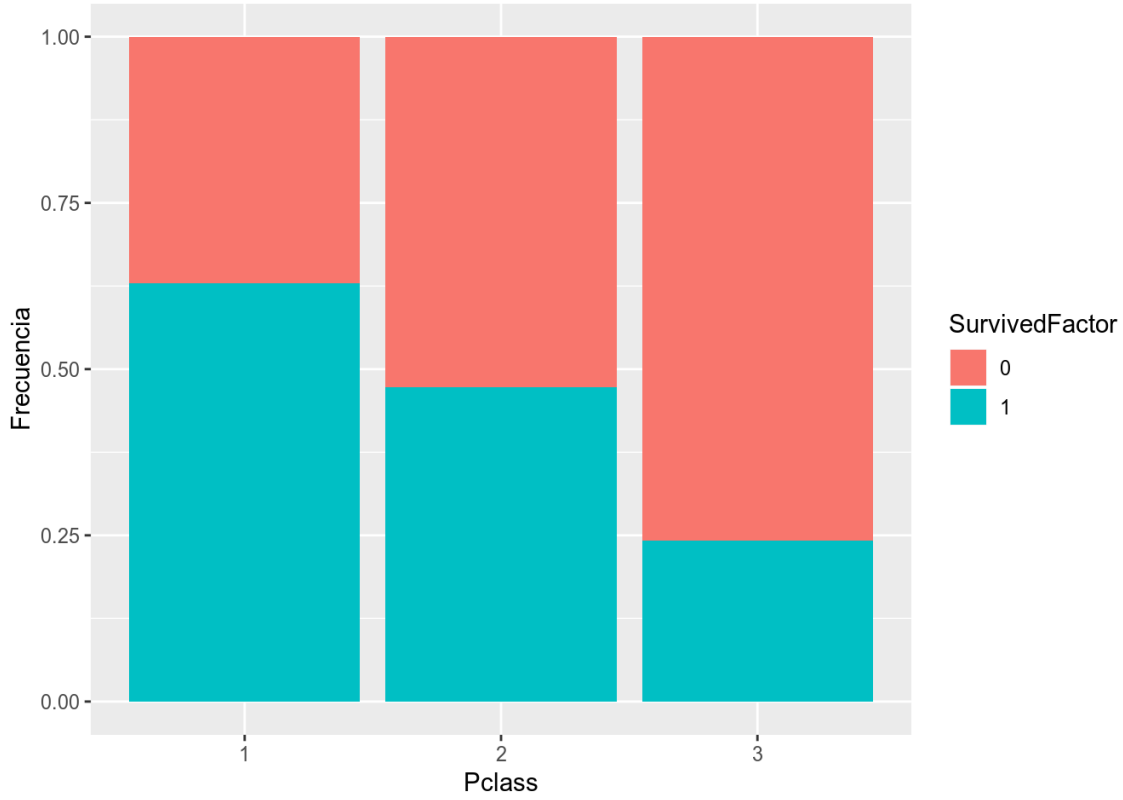



```
ggplot(data=datosTest[,],aes(x=`Sex`,fill=`resultadoModelo5Factor`))+geom_bar(position="fill"
)+ylab("Frecuencia")
```



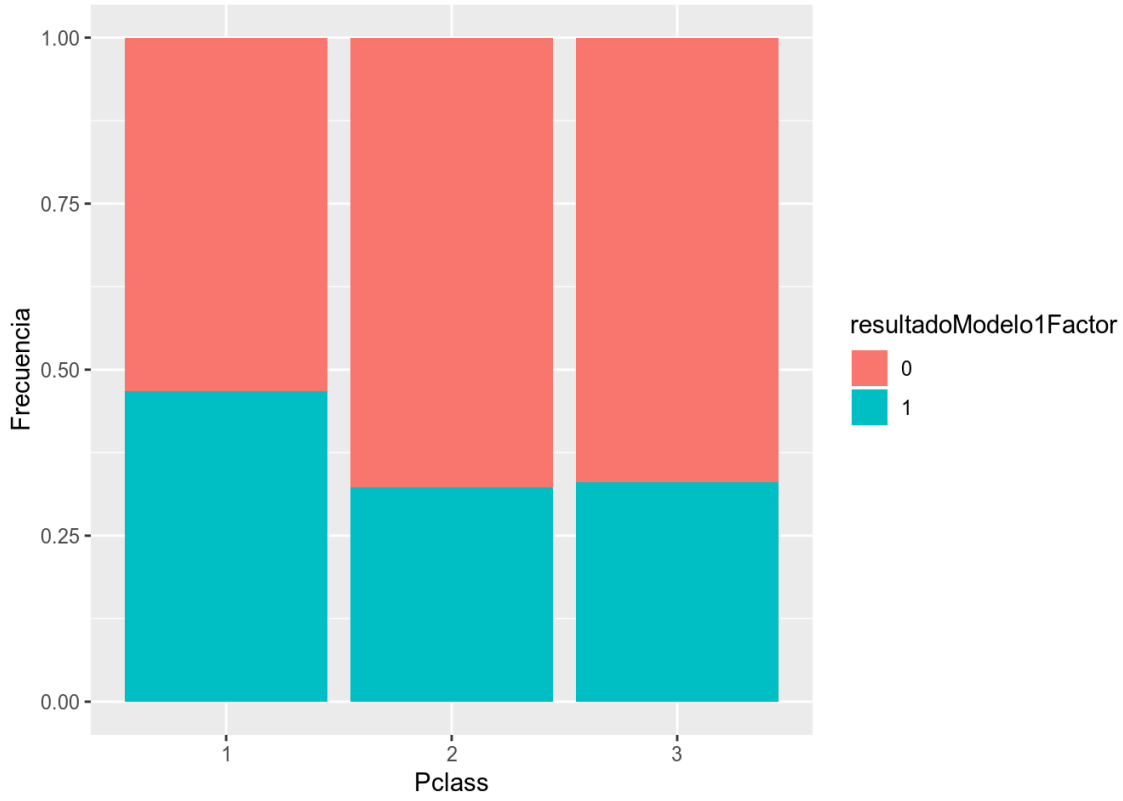
De igual forma se repite la distribución de supervivientes en base a la clase de su camarote.

```
ggplot(data=datosEntrenamiento[,],aes(x=`Pclass`,fill=`SurvivedFactor`))+geom_bar(position="f
ill")+ylab("Frecuencia")
```

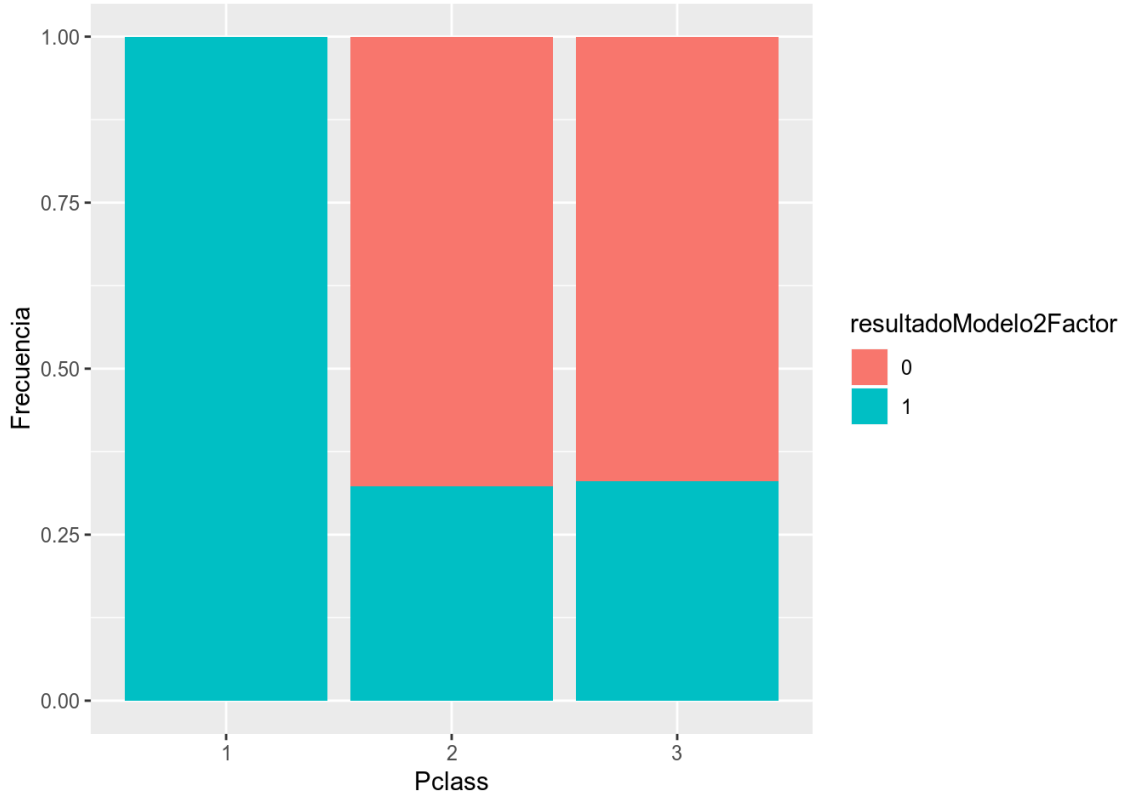


Para luego compararlo con los modelos generados. Los modelos generados tampoco parecen seguir la misma tendencia por calidad del camarote.

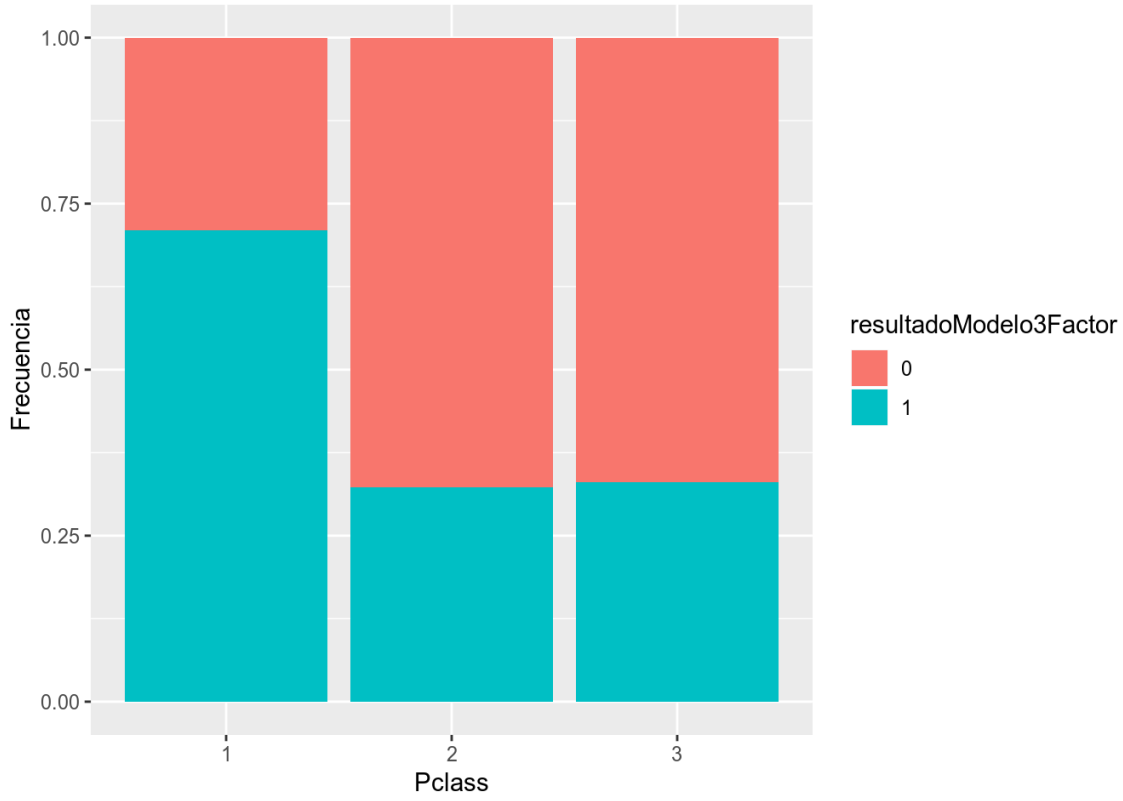
```
ggplot(data=datosTest[,],aes(x=`Pclass`,fill=`resultadoModelo1Factor`))+geom_bar(position="fi
ll")+ylab("Frecuencia")
```



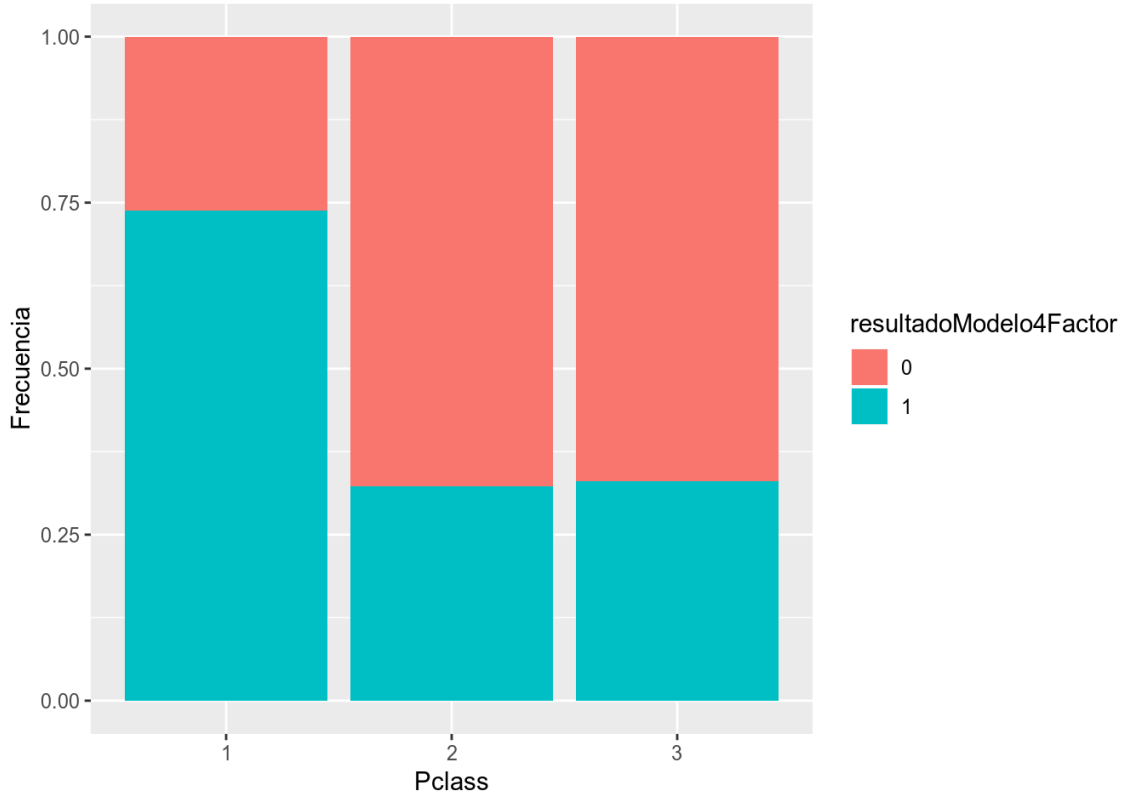

```
ggplot(data=datosTest[,],aes(x=`Pclass`,fill=`resultadoModelo2Factor`))+geom_bar(position="fi
ll")+ylab("Frecuencia")
```



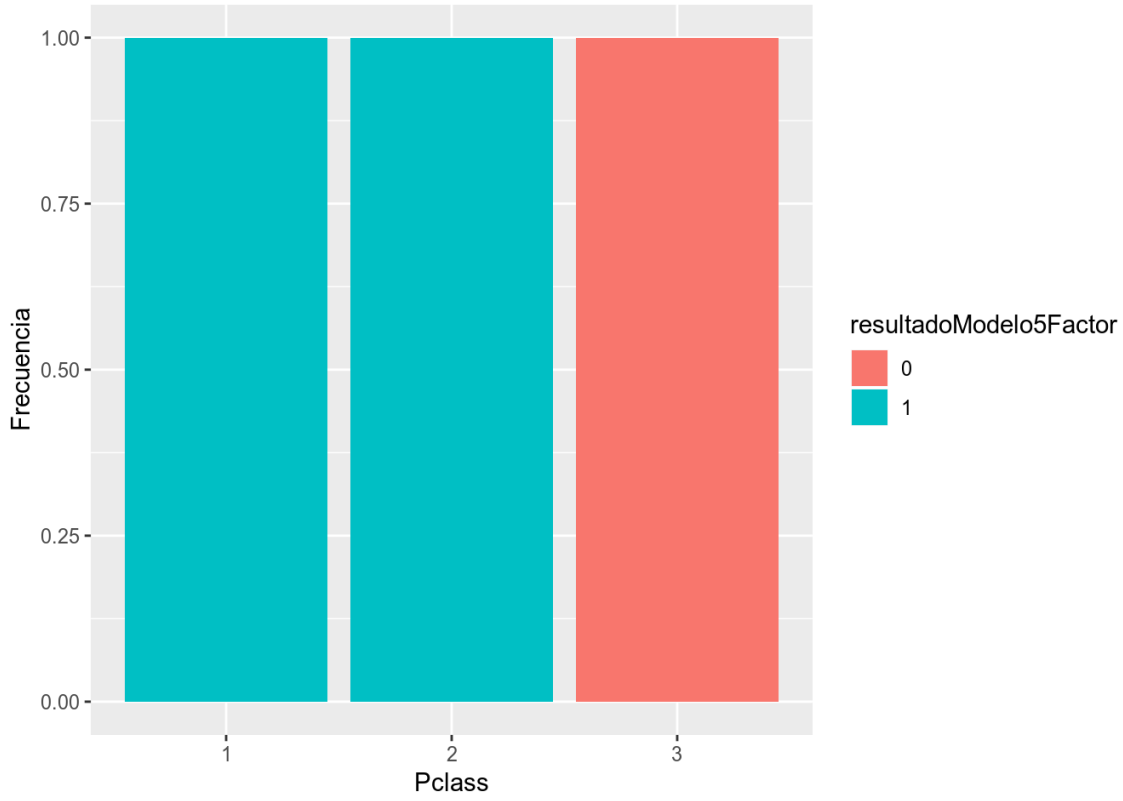

```
ggplot(data=datosTest[,],aes(x=`Pclass`,fill=`resultadoModelo3Factor`))+geom_bar(position="fi
ll")+ylab("Frecuencia")
```




```
ggplot(data=datosTest[,],aes(x=`Pclass`,fill=`resultadoModelo4Factor`))+geom_bar(position="fi
ll")+ylab("Frecuencia")
```




```
ggplot(data=datosTest[,],aes(x=`Pclass`,fill=`resultadoModelo5Factor`))+geom_bar(position="fi
ll")+ylab("Frecuencia")
```



7 Resolución del problema

A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

La conclusiones son:

- Los pasajeros de género femenino tienen muchas más posibilidades de sobrevivir que los pasajeros de género masculino.
- Los pasajeros de camarotes de primera clase tienen más posibilidades de sobrevivir que el resto. Además, los pasajeros de segunda clase tienen también más posibilidades que los de tercera clase.
- Los pasajeros con familia tienen más posibilidades de sobrevivir.
- El puerto de embarque usado no afecta significativamente a la supervivencia del pasajero.
- La edad no afecta significativamente a la supervivencia del pasajero.

Por tanto las causas de la alta mortandad quedan analizadas y explicadas, aportando respuestas al problema planteado.

8 Código

Hay que adjuntar el código, preferiblemente en R, con el que se ha realizado la limpieza, análisis y representación de los datos. Si lo preferís, también podéis trabajar en Python.

El código está disponible en el fichero aantonc.Rmd disponible en GitHub.