Limpieza y validación de datos - Práctica 2

Autor: Angel Vicente Sanjuan Martin

Junio 2021

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. Para hacer esta práctica tendréis que trabajar en grupos de 2 personas. Tendréis que entregar un solo archivo con el enlace Github (https://github.com) donde se encuentren las soluciones incluyendo los nombres de los componentes del equipo. Podéis utilizar la Wiki de Github para describir vuestro equipo y los diferentes archivos que corresponden a vuestra entrega. Cada miembro del equipo tendrá que contribuir con su usuario Github. Aunque no se trata del mismo enunciado, los siguientes ejemplos de ediciones anteriores os pueden servir como guía: * Ejemplo: https://github.com/Bengis/nba-gap-cleaning * Ejemplo complejo (archivo adjunto).

Competencias

En esta práctica se desarrollan las siguientes competencias del Máster 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.

Objetivos

Los objetivos concretos de esta práctica son:

- 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.

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: 1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder? 2. Integración y selección de los datos de interés a analizar. 3. Limpieza de los datos. 3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos? 3.2. Identificación y tratamiento de valores extremos. 4. Análisis de los datos. 4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar). 4.2. Comprobación de la normalidad y homogeneidad de la varianza. 4.3. 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. Aplicar al menos tres métodos de análisis diferentes. 5. Representación de los resultados a partir de tablas y gráficas. 6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema? 7. 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.

Recursos

Los siguientes recursos son de utilidad para la realización de la práctica: * Calvo M., Subirats L., Pérez D. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC. * Megan Squire (2015). Clean Data. Packt Publishing Ltd. * Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann. * Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369 . * Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media. * Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc. * Tutorial de Github https://guides.github.com/activities/hello-world.

Criterios de evaluación

Todos los apartados son obligatorios. La ponderación de los ejercicios es la siguiente: * Los apartados 1, 2 y 6 valen 0,5 puntos. * Los apartados 3, 5 y 7 valen 2 puntos. * El apartado 4 vale 2,5 puntos. Se valorará la idoneidad de las respuestas, que deberán ser claras y completas. Las diferentes etapas deberán justificarse y acompañarse del código correspondiente. También se valorará la síntesis y claridad, a través del uso de comentarios, del código resultante, así como la calidad de los datos finales analizados.

Formato y fecha de entega

Durante la semana del 24 al 28 de mayo el grupo podrá entregar al profesor una entrega parcial opcional. Esta entrega parcial es muy recomendable para recibir asesoramiento sobre la práctica y verificar que la dirección tomada es la correcta. Se entregarán comentarios a los estudiantes que hayan efectuado la entrega parcial pero no contará para la nota de la práctica. En la entrega parcial los estudiantes deberán entregar por correo electrónico, al profesor encargado del aula, el enlace al repositorio Github con el que hayan avanzado.

En referente a la entrega final, hay que entregar un único fichero que contenga el enlace Github, el cual no se podrá modificar posteriormente a la fecha de entrega, donde haya: 1. Una Wiki con los nombres de los componentes del grupo y una descripción de los ficheros. 2. Un documento PDF con las respuestas a las preguntas y los nombres de los componentes del grupo. Además, al final del documento, deberá aparecer la siguiente tabla de contribuciones al trabajo, la cual debe firmar cada integrante del grupo con sus iniciales. Las iniciales representan la confirmación de que el integrante ha participado en dicho apartado. Todos los integrantes deben participar en cada apartado, por lo que, idealmente, los apartados deberían estar firmados por todos los integrantes. 3. Una carpeta con el código generado para analizar los datos. 4. El fichero CSV con los datos originales. 5. El fichero CSV con los datos finales analizados. Este documento de entrega final de la Práctica 2 se debe entregar en el espacio de Entrega y Registro de AC del aula antes de las 23:59 del día 8 de junio. No se aceptarán entregas fuera de plazo

Procesos de limpieza del conjunto de datos

Primer contacto con el conjunto de datos, visualizamos su estructura.

```
# Cargamos los paquetes R que vamos a usar
library(ggplot2)
library(dplyr)

# Guardamos el conjunto de datos test y train en un único dataset
test <- read.csv('test.csv',stringsAsFactors = FALSE)
train <- read.csv('train.csv', stringsAsFactors = FALSE)

# Unimos los dos conjuntos de datos en uno solo
totalData <- bind_rows(train,test)
filas=dim(train)[1]

# Verificamos la estructura del conjunto de datos
str(totalData)
```

```
## 'data.frame':    1309 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  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Trabajamos los atributos con valores vacíos.

```
# Estadísticas de valores vacíos
colSums(is.na(totalData))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0         418           0           0           0         263
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0           1           0           0
```

```
colSums(totalData=="")
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##           0          NA           0           0           0          NA
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           0           0           0          NA        1014           2
```

```
# Tomamos valor "C" para los valores vacíos de la variable "Embarked"
totalData$Embarked[totalData$Embarked==""]="C"

# Tomamos la media para valores vacíos de la variable "Age"
totalData$Age[is.na(totalData$Age)] <- mean(totalData$Age,na.rm=T)
```

Discretizamos cuando tiene sentido y en función de cada variable.

```
# ¿Con qué variables tendría sentido un proceso de discretización?
apply(totalData,2, function(x) length(unique(x)))
```

```
## PassengerId    Survived      Pclass        Name         Sex         Age
##        1309           3           3        1307           2          99
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked
##           7           8         929         282         187           3
```

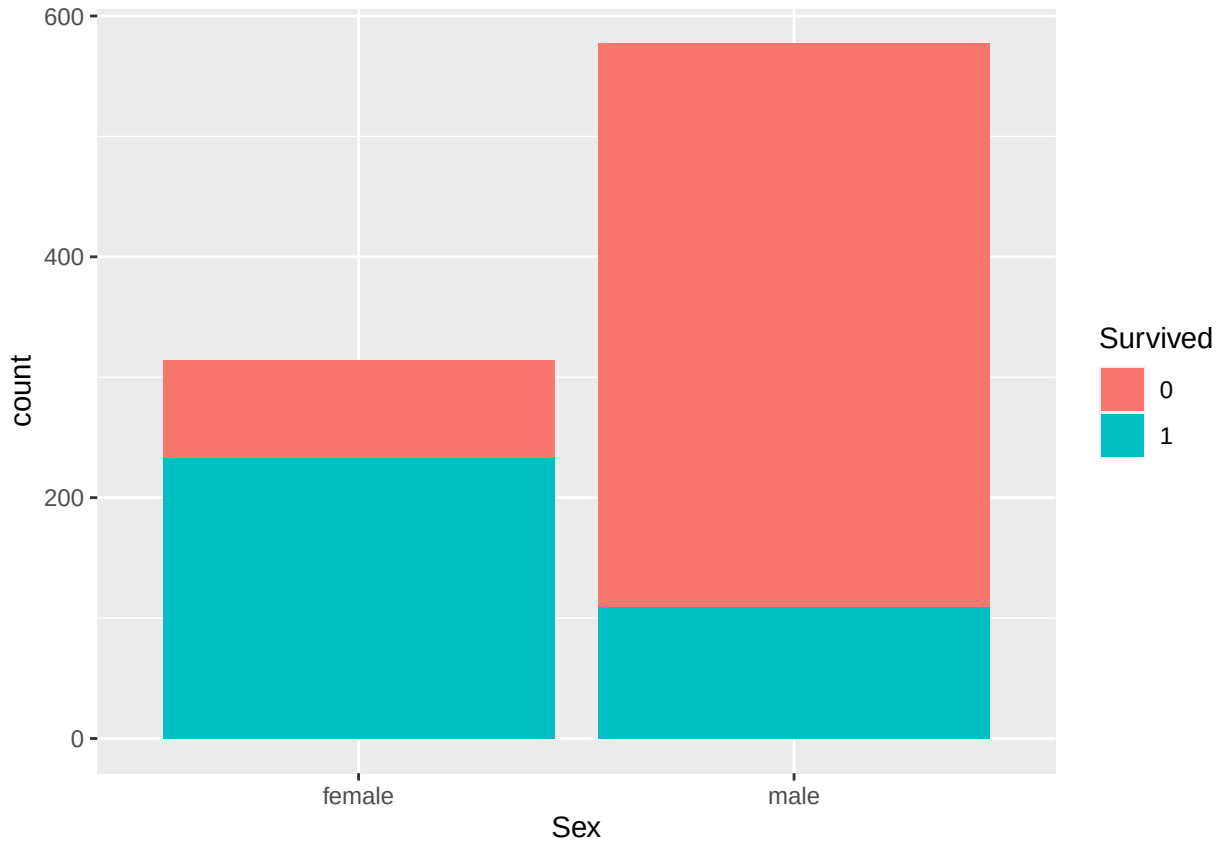
```
# Discretizamos las variables con pocas clases
cols<-c("Survived","Pclass","Sex","Embarked")
for (i in cols){
  totalData[,i] <- as.factor(totalData[,i])
}

# Después de los cambios, analizamos la nueva estructura del conjunto de datos
str(totalData)
```

```
## 'data.frame':    1309 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 3 levels "C","Q","S": 3 1 3 3 3 2 3 3 3 1 ...
```

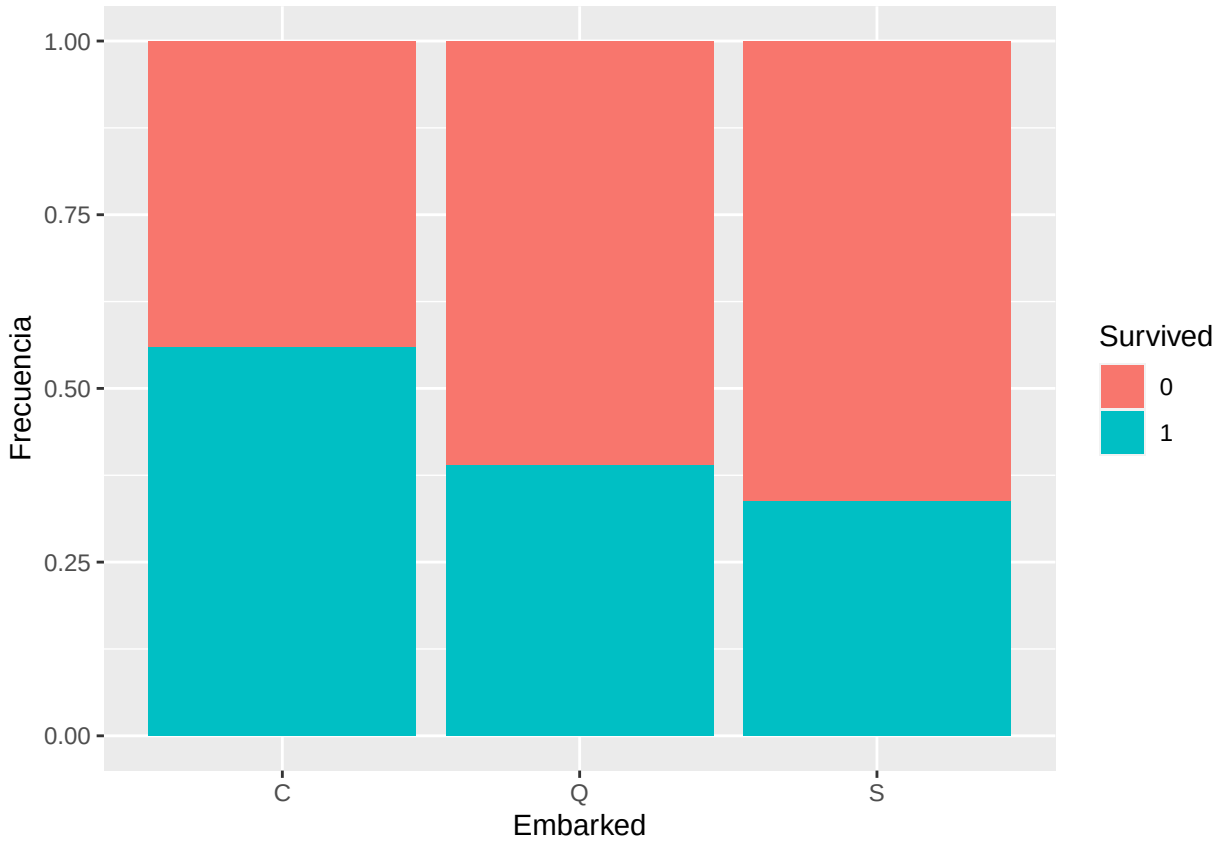
Procesos de análisis del conjunto de datos

Nos proponemos analizar las relaciones entre las diferentes variables del conjunto de datos.

```
# Visualizamos la relación entre las variables "sex" y "survival":
ggplot(data=totalData[1:filas,],aes(x=Sex,fill=Survived))+geom_bar()
```




```
# Otro punto de vista. Survival como función de Embarked:
ggplot(data = totalData[1:filas,],aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+ylab("Frecue
```



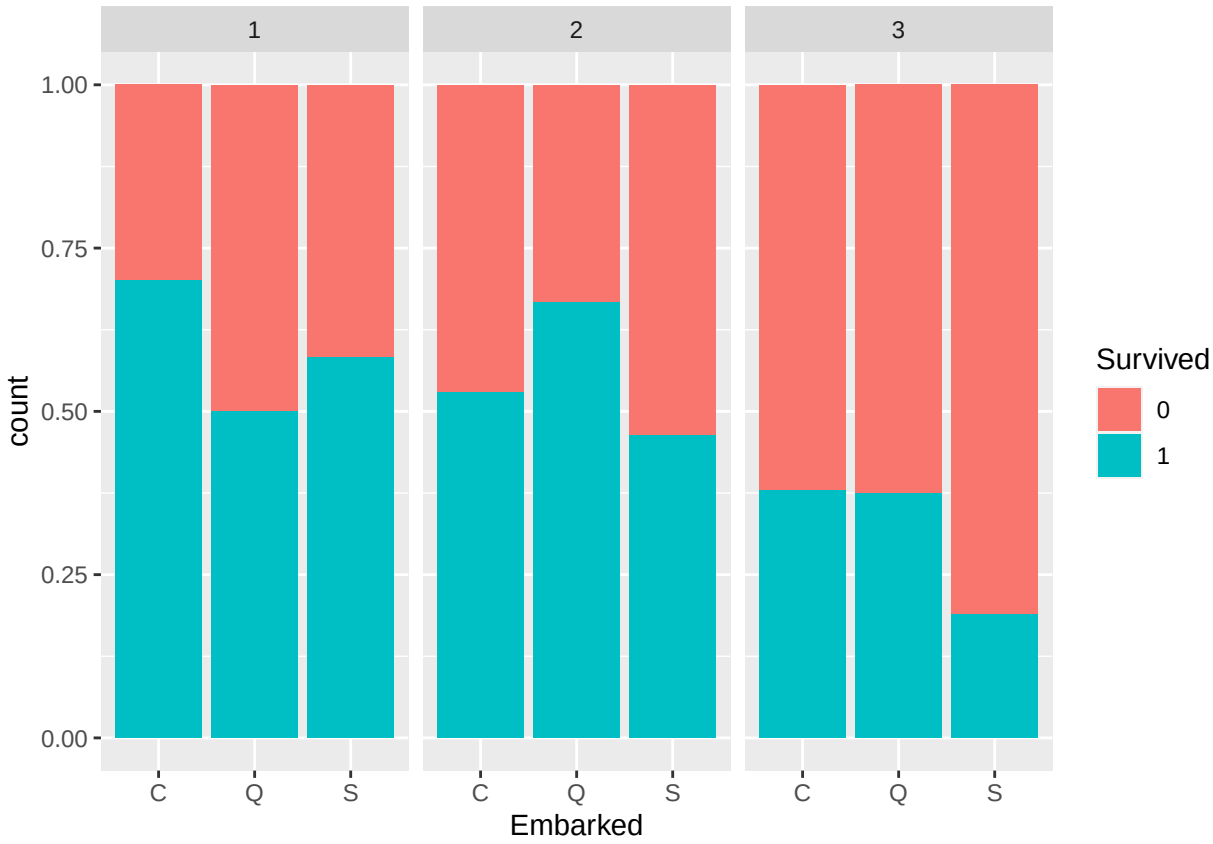
Obtenemos una matriz de porcentajes de frecuencia.
Vemos, por ejemplo que la probabilidad de sobrevivir si se embarcó en "C" es de un 55,88%

```
t<-table(totalData[1:filas,]$Embarked,totalData[1:filas,]$Survived)
for (i in 1:dim(t)[1]){
    t[i,]<-t[i,]/sum(t[i,])*100
}
t
```

```
##
##            0        1
##   C 44.11765 55.88235
##   Q 61.03896 38.96104
##   S 66.30435 33.69565
```

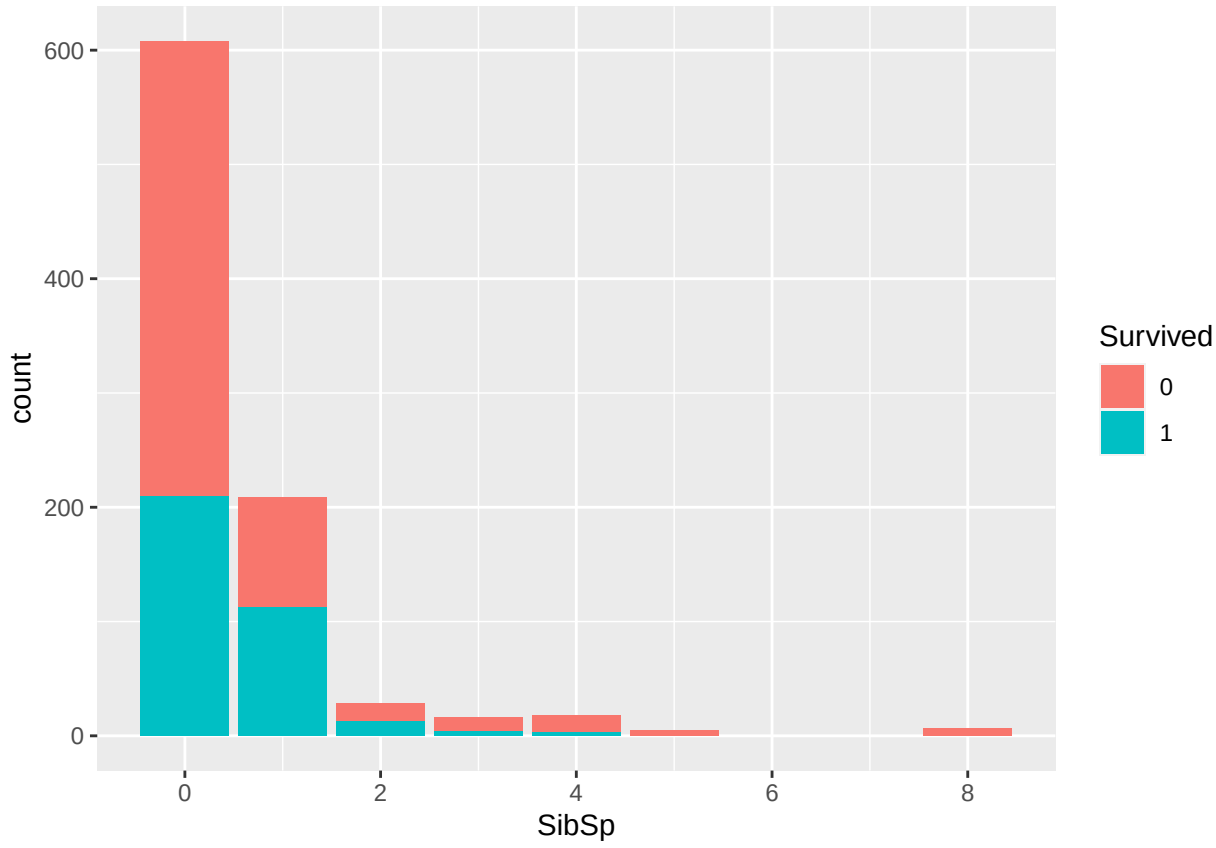
Veamos ahora como en un mismo gráfico de frecuencias podemos trabajar con 3 variables: Embarked, Survived y Pclass.

```
# Ahora, podemos dividir el gráfico de Embarked por Pclass:
ggplot(data = totalData[1:filas,],aes(x=Embarked,fill=Survived))+geom_bar(position="fill")+facet_wrap(~
```

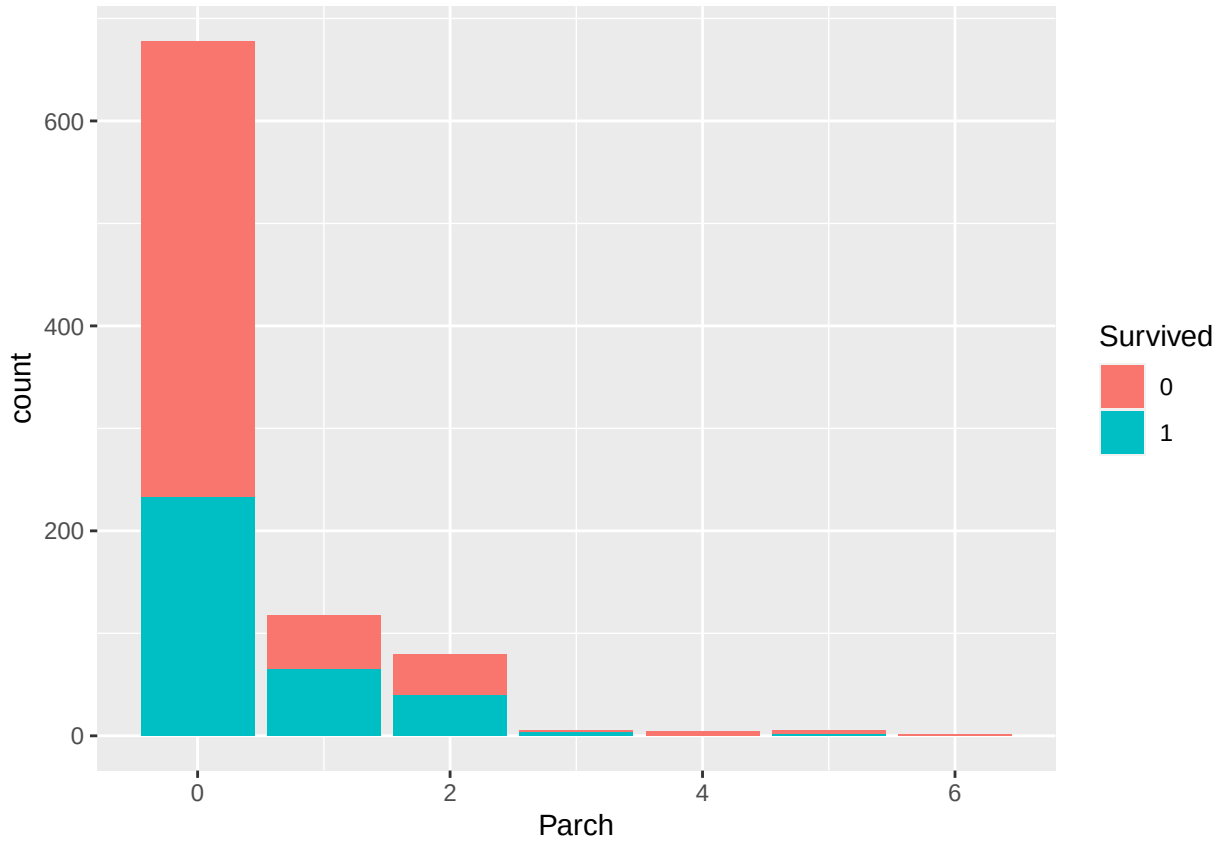


Comparemos ahora dos gráficos de frecuencias: Survived-SibSp y Survived-Parch

```
# Survivial como función de SibSp y Parch
ggplot(data = totalData[1:filas,],aes(x=SibSp,fill=Survived))+geom_bar()
```



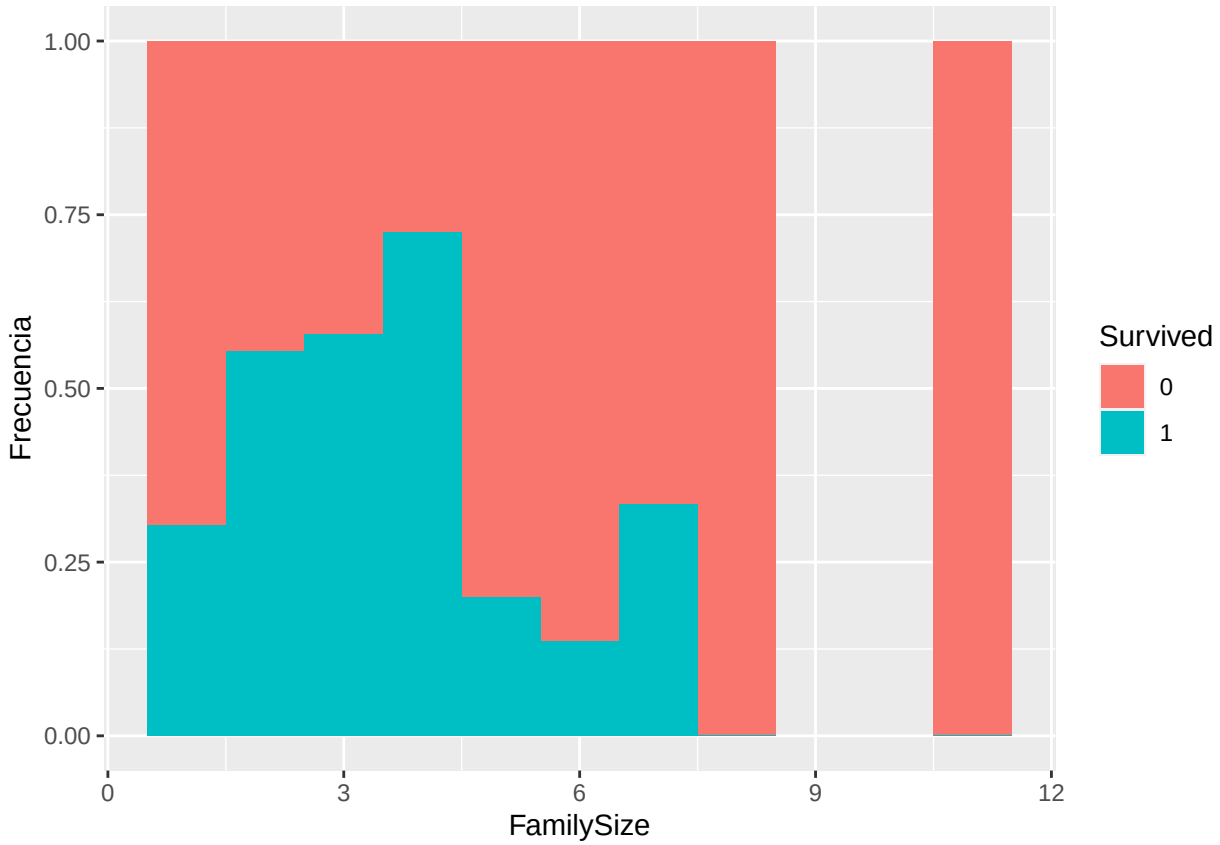

```
ggplot(data = totalData[1:filas,],aes(x=Parch,fill=Survived))+geom_bar()
```




```
# Vemos como las forma de estos dos gráficos es similar. Este hecho nos puede indicar presencia de corr
```

Veamos un ejemplo de construcción de una variable nueva: Tamaño de familia

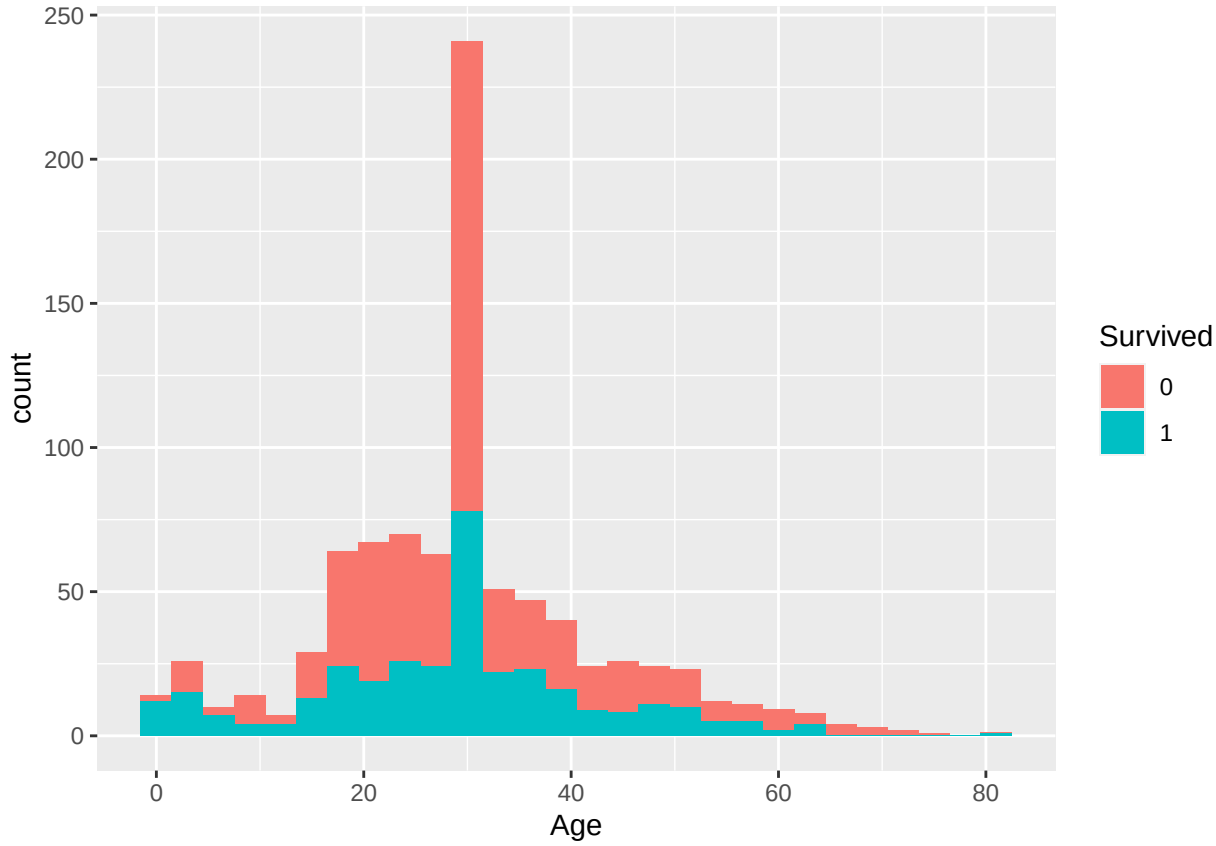
```
# Construimos un atributo nuevo: family size.
totalData$FamilySize <- totalData$SibSp + totalData$Parch +1;
totalData1<-totalData[1:filas,]
ggplot(data = totalData1[!is.na(totalData[1:filas,]$FamilySize),],aes(x=FamilySize,fill=Survived))+geom
```



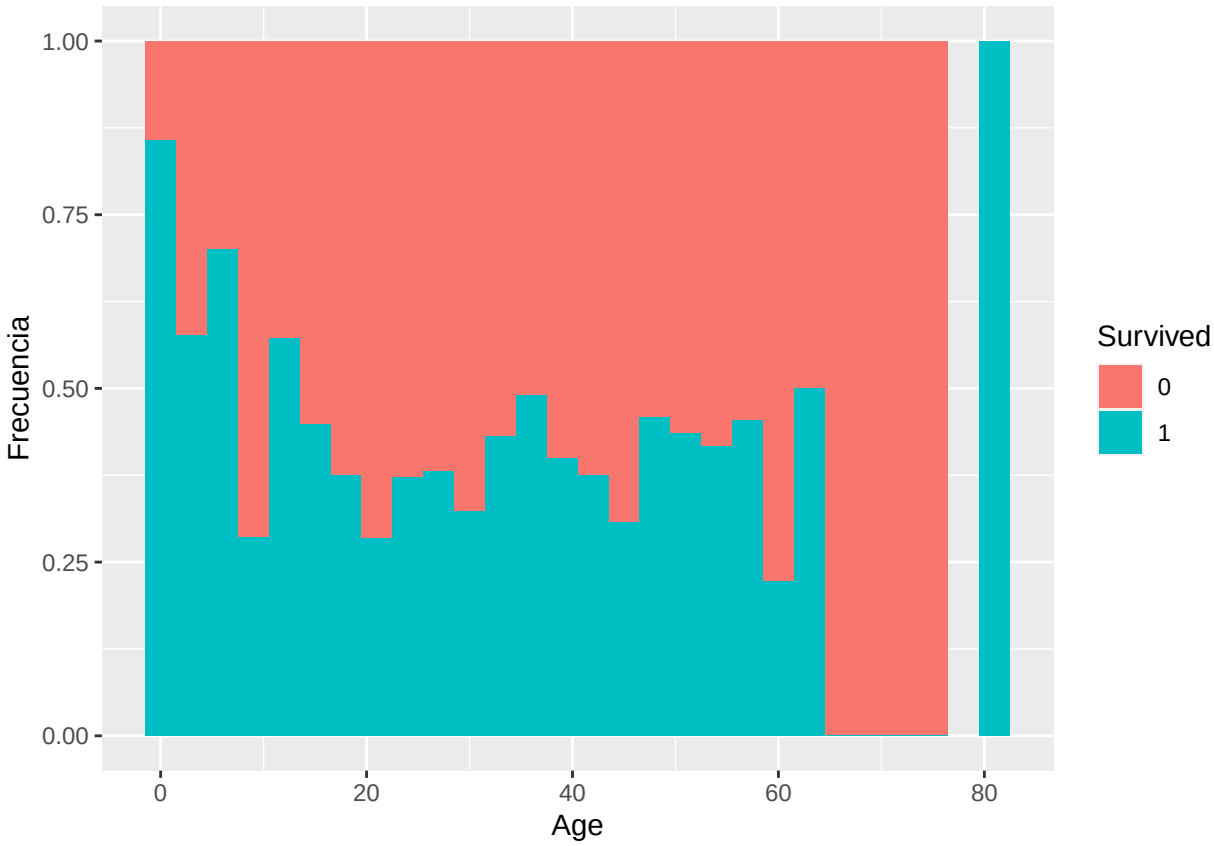

```
# Observamos como familias de entre 2 y 6 miembros tienen más del 50% de posibilidades de supervivencia
```

Veamos ahora dos gráficos que nos compara los atributos Age y Survived.
Observamos como el parámetro position="fill" nos da la proporción acumulada de un atributo dentro de otro

```
# Survival como función de age:
ggplot(data = totalData1[!(is.na(totalData[1:filas,]$Age)),],aes(x=Age,fill=Survived))+geom_histogram(bi
```




```
ggplot(data = totalData1[!is.na(totalData[1:filas,]$Age),],aes(x=Age,fill=Survived))+geom_histogram(bin
```


1.00
0.75
0.50
0.25
0.00
Frecuencia
0
20
40
60
80
Age
Survived
0
1