Mineria de datos 2019 PEC1 SOLUCION

Mineria de datos (Universitat Oberta de Catalunya)

Universitat Oberta de Catalunya

Estudios de Informática, Multimedia y Telecomunicaciones

Minería de datos: PEC1

Autor: Ander Lopetegui

Octubre 2019

1 Introducción

1.1 Presentación

Esta prueba de evaluación continuada cubre el módulo 1,2 y 8 del programa de la asignatura.

1.2 Competencias

Las competencias que se trabajan en esta prueba son:

- Uso y aplicación de las TIC en el ámbito académico y profesional
- Capacidad para ir

- Capacidad para evaluar soluciones tecnológicas y elaborar propuestas de proyectos teniendo en cuenta los recursos, las alternativas disponibles y las condiciones de mercado.
- Conocer las tecnologías de comunicaciones actuales y emergentes, así como saberlas aplicar convenientemente para diseñar y desarrollar soluciones basadas en sistemas y tecnologías de la información.
- Aplicación de las técnicas específicas de ingeniería del software en las diferentes etapas del ciclo de vida de un proyecto.
- Capacidad para aplicar las técnicas específicas de tratamiento, almacenamiento y administración de datos.
- Capacidad para proponer y evaluar diferentes alternativas tecnológicas para resolver un problema concreto.
- Capacidad de utilizar un lenguaje de programación.
- Capacidad para desarrollar en una herramienta IDE.
- Capacidad de plantear un proyecto de minería de datos.

1.3 Objetivos

- Asimilar correctamente el módulo 1 y 2.
- Qué es y qué no es MD.
- Ciclo de vida de los proyectos de MD.
- Diferentes tipologías de MD.
- Conocer las técnicas propias de una fase de preparación de datos y objetivos a alcanzar.

1.4 Descripción de la PEC a realizar

La prueba está estructurada en 1 ejercicio teórico/práctico y 1 ejercicio práctico que pide que se desarrolle la fase de preparación en un juego de datos.
Deben responderse todos los ejercicios para poder superar la PEC.

1.5 Recursos

Para realizar esta práctica recomendamos la lectura de los siguientes documentos:

- Módulo 1, 2 y 8 del material didáctico.
- RStudio Cheat Sheet: Disponible en el aula Laboratorio de Minería de datos.
- R Base Cheat Sheet: Disponible en el aula Laboratorio de Minería de datos.

1.6 Criterios de evaluación

Ejercicios teóricos
Todos los ejercicios deben ser presentados de forma razonada y clara, especificando todos y cada uno de los pasos que se hayan llevado a cabo para su resolución. No se aceptará ninguna respuesta que no esté claramente justificada.

Ejercicios prácticos
Para todas las PEC es necesario documentar en cada apartado del ejercicio práctico qué se ha hecho y cómo se ha hecho.

1.7 Formato y fecha de entega

El formato de entrega es: usernameestudiant-PECn.html y rmd
Fecha de Entrega: 30/10/2019
Se debe entregar la PEC en el buzón de entregas del aula

1.8 Nota: Propiedad intelectual

> A menudo es inevitable, al producir una obra multimedia, hacer uso de recursos creados por terceras personas. Es por lo tanto comprensible hacerlo en el marco de una práctica de los estudios de Informática, Multimedia y Telecomunicación de la UOC, siempre y cuando esto se documente claramente y no suponga plagio en la práctica.

> Por lo tanto, al presentar una práctica que haga uso de recursos ajenos, se debe presentar junto con ella un documento en qué se detallen todos ellos, especificando el nombre de cada recurso, su autor, el lugar dónde se obtuvo y su estatus legal: si la obra está protegida por el copyright o se acoge a alguna otra licencia de uso (Creative Commons, licencia GNU, GPL ...). El estudiante deberá asegurarse de que la licencia no impide específicamente su uso en el marco de la práctica. En caso de no encontrar la información correspondiente tendrá que asumir que la obra está protegida por copyright.

> Deberéis, además, adjuntar los ficheros originales cuando las obras utilizadas sean digitales, y su código fuente si corresponde.

2 Enunciado

Como ejemplo, trabajaremos con el conjunto de datos “Titanic” que recoge datos sobre el famoso crucero y sobre el que es fácil realizar tareas de clasificación predictiva sobre la variable “Survived”.

De momento dejaremos para las siguientes prácticas el estudio de algoritmos predictivos y nos centraremos por ahora en el estudio de las variables de una muestra de datos, es decir, haremos un trabajo descriptivo del mismo.

Las actividades que llevaremos a cabo en esta práctica suelen enmarcarse en las fases iniciales de un proyecto de minería de datos y consisten en la selección de caraterísticas o variables y la preparación del los datos para posteriormente ser consumido por un algoritmo.

Las técnicas que trabajaremos son las siguientes:

1. Normalización
2. Discretización
3. Gestión de valores nulos
4. Estudio de correl

5. Reducción de la dimensionalidad
6. Análisis visual del conjunto de datos

3 Ejemplo de estudio visual con el juego de datos Titanic

3.1 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('titanic-test.csv',stringsAsFactors = FALSE)
train <- read.csv('titanic-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)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
...
##  $ 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
```

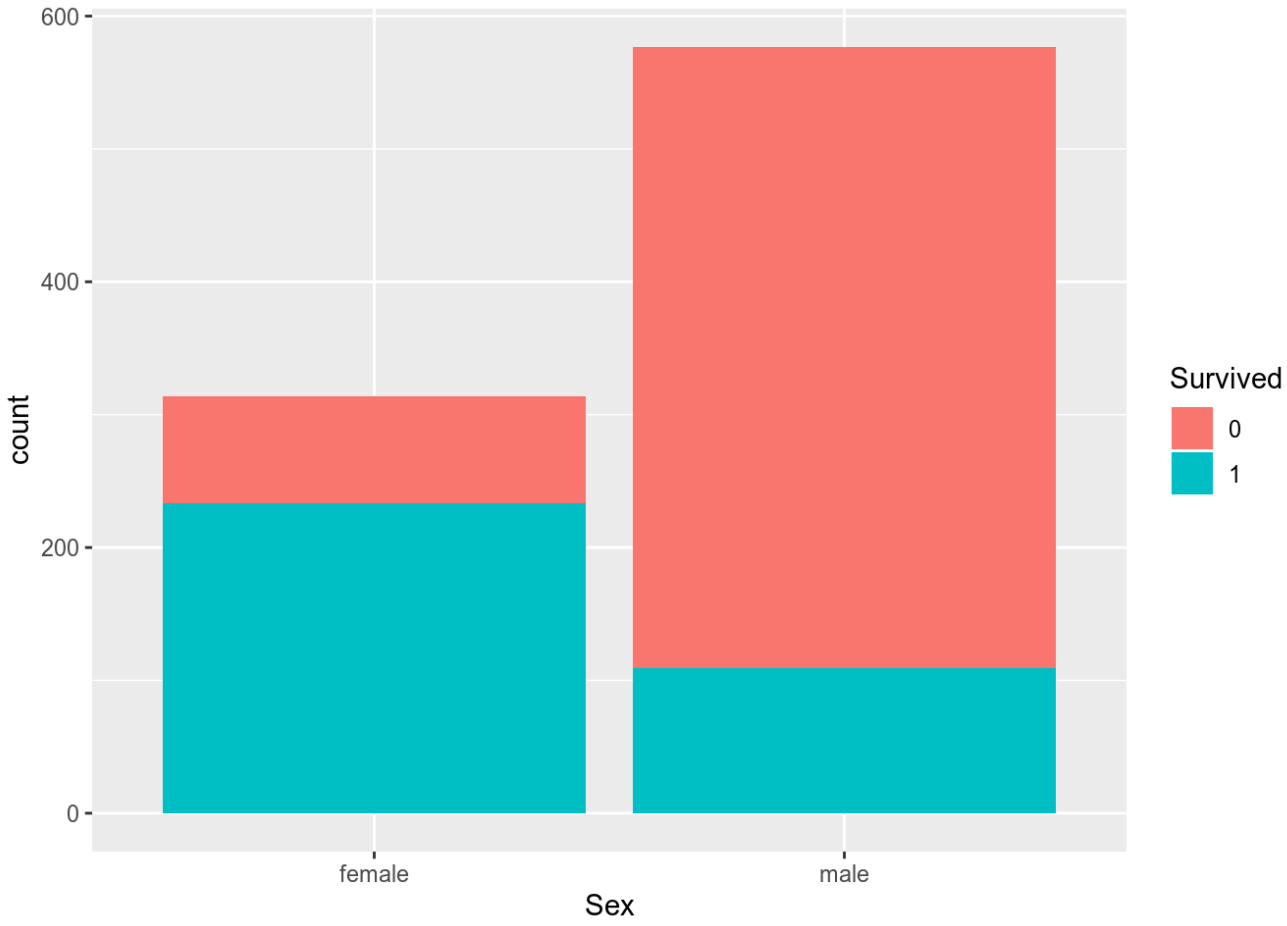
```
# Discretizamos las variables con pocas clases
cols<-c("Survived","Pclass","Sex","Embarked")
for (i in cols){
  totalData[,i] <- as.factor(totalData[,i])
}

# 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)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
...
##  $ 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 ...
```

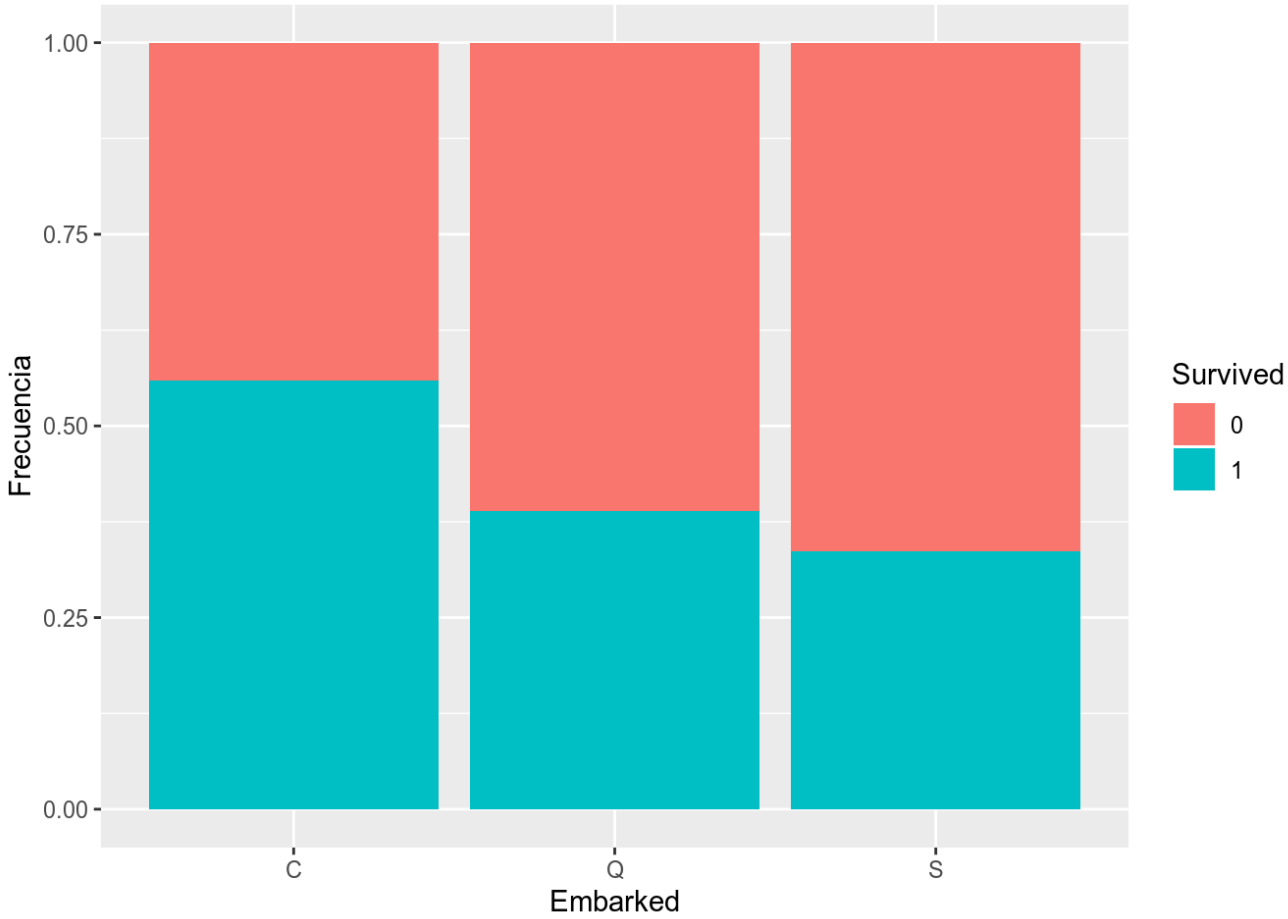
3.2 Procesos de análisis del conjunto de datos

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="fil
l")+ylab("Frecuencia")
```



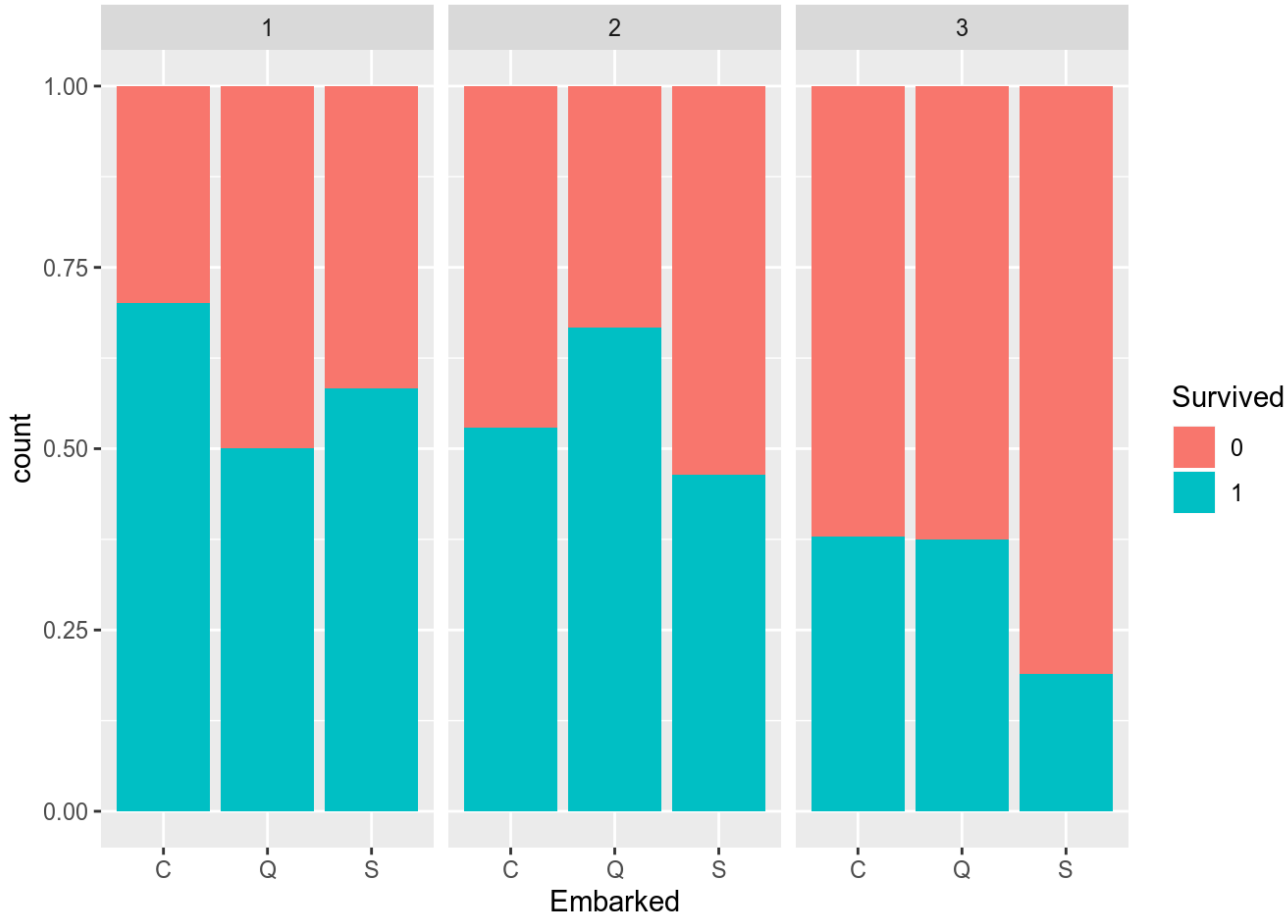
Obtenemos una matriz de porcentages de frecuencia.
Vemos, por ejemplo que la probabilidad de sobrevivir si se embarcó en “C” es de un 55,88%

```
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
```

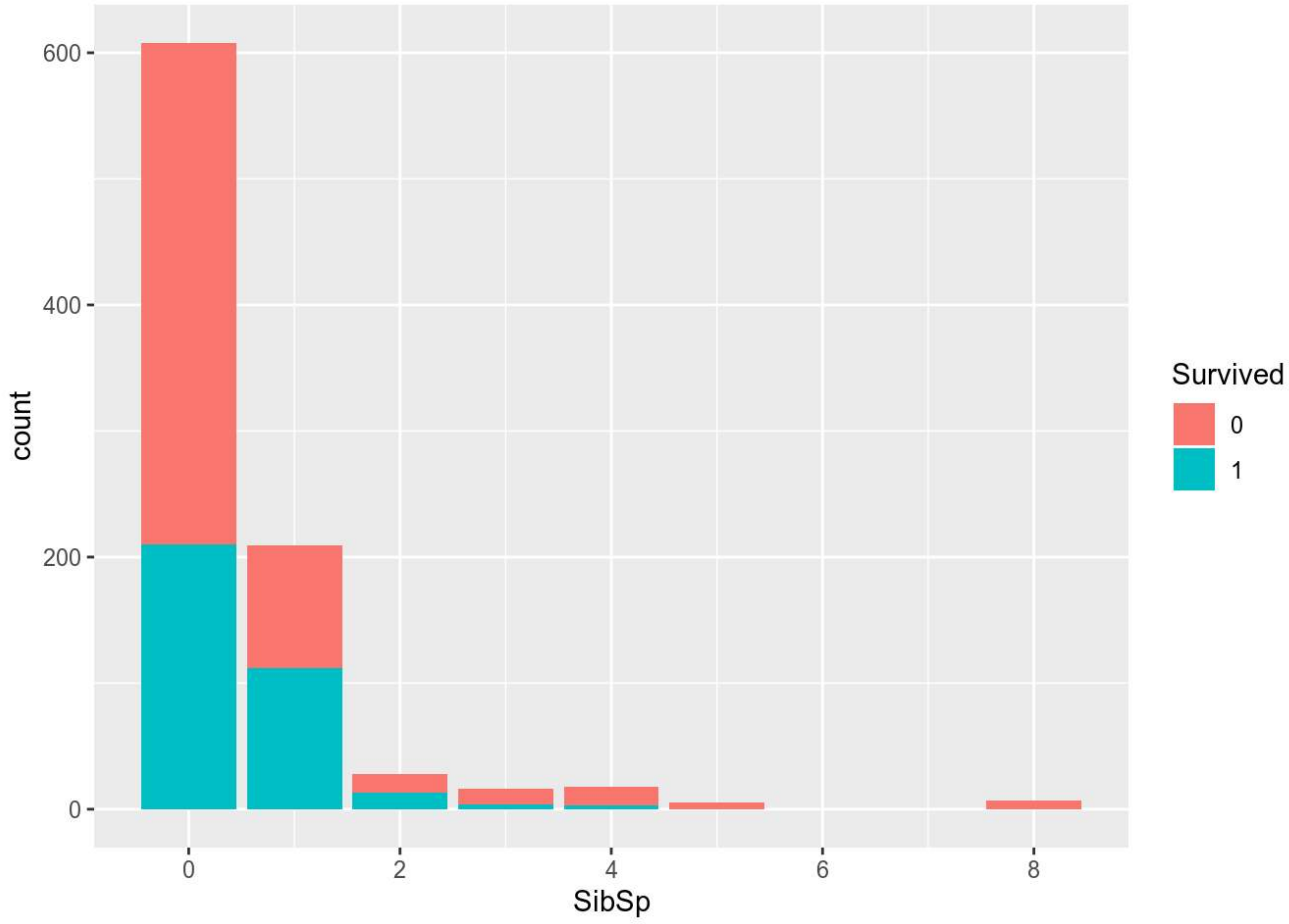
Veamos ahora como en un mismo gráfico de frecuencias podemos trabajar con 3 variables: Embarked, Survived y Pclass.

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

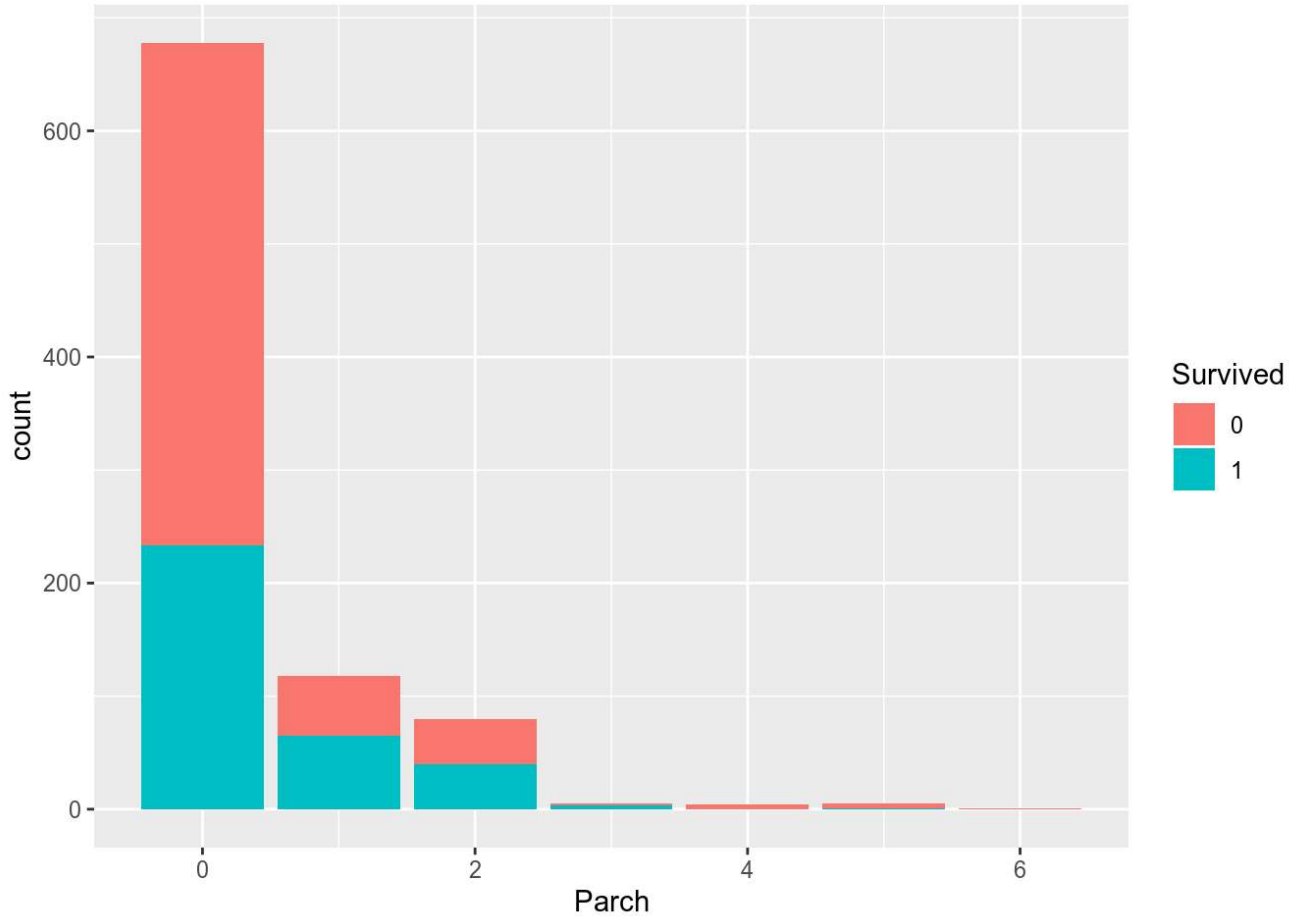


Comparemos ahora dos gráficos de frecuencias: Survived-SibSp y Survived-Parch

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



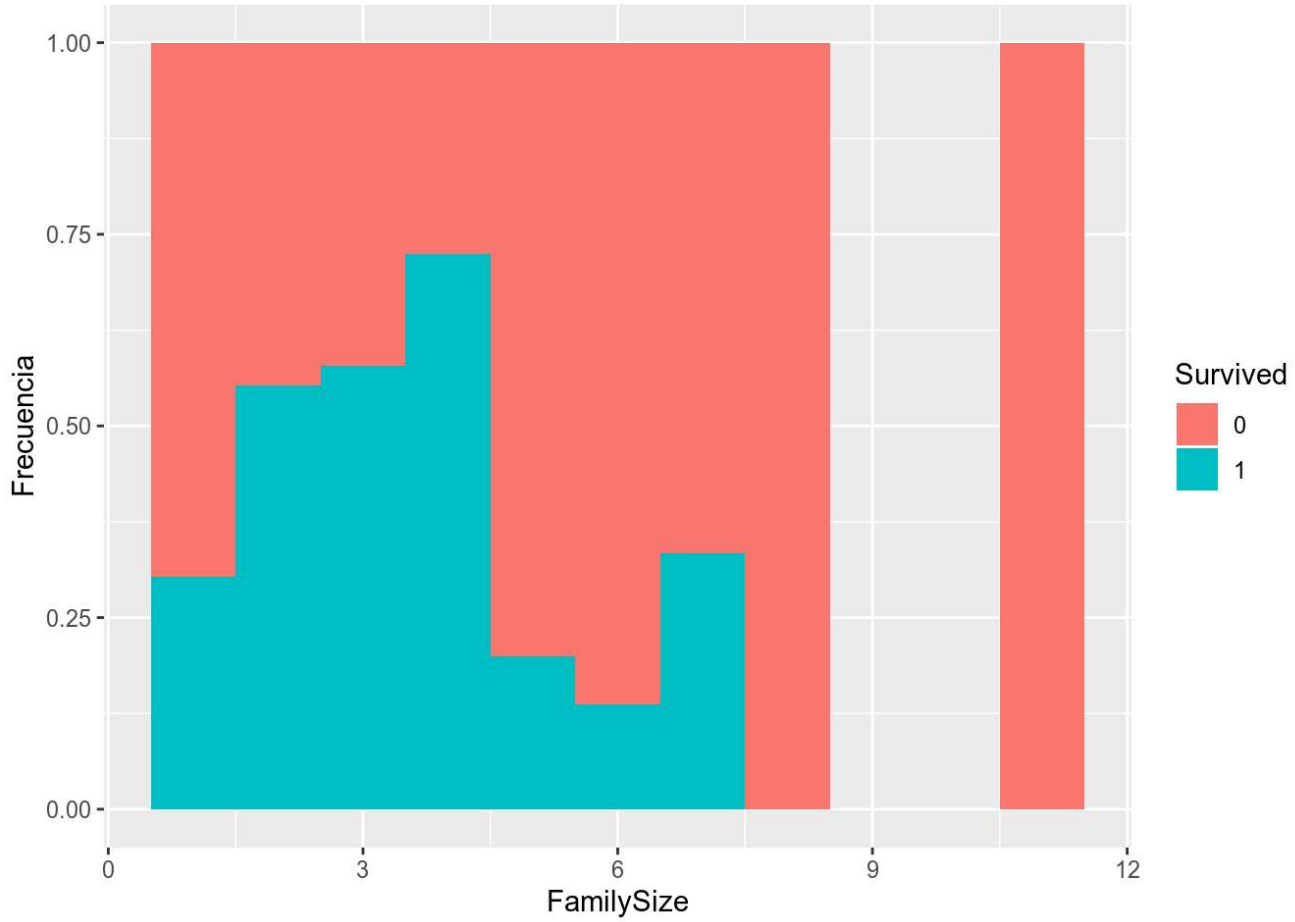

```
ggplot(data = totalData[1:filas,],aes(x=Parch,fill=Survived))+geom_bar()
```



```
# Vemos como las forma de estos dos gráficos es similar. Este hecho nos puede indicar pr
esencia de correlaciones altas.
```

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

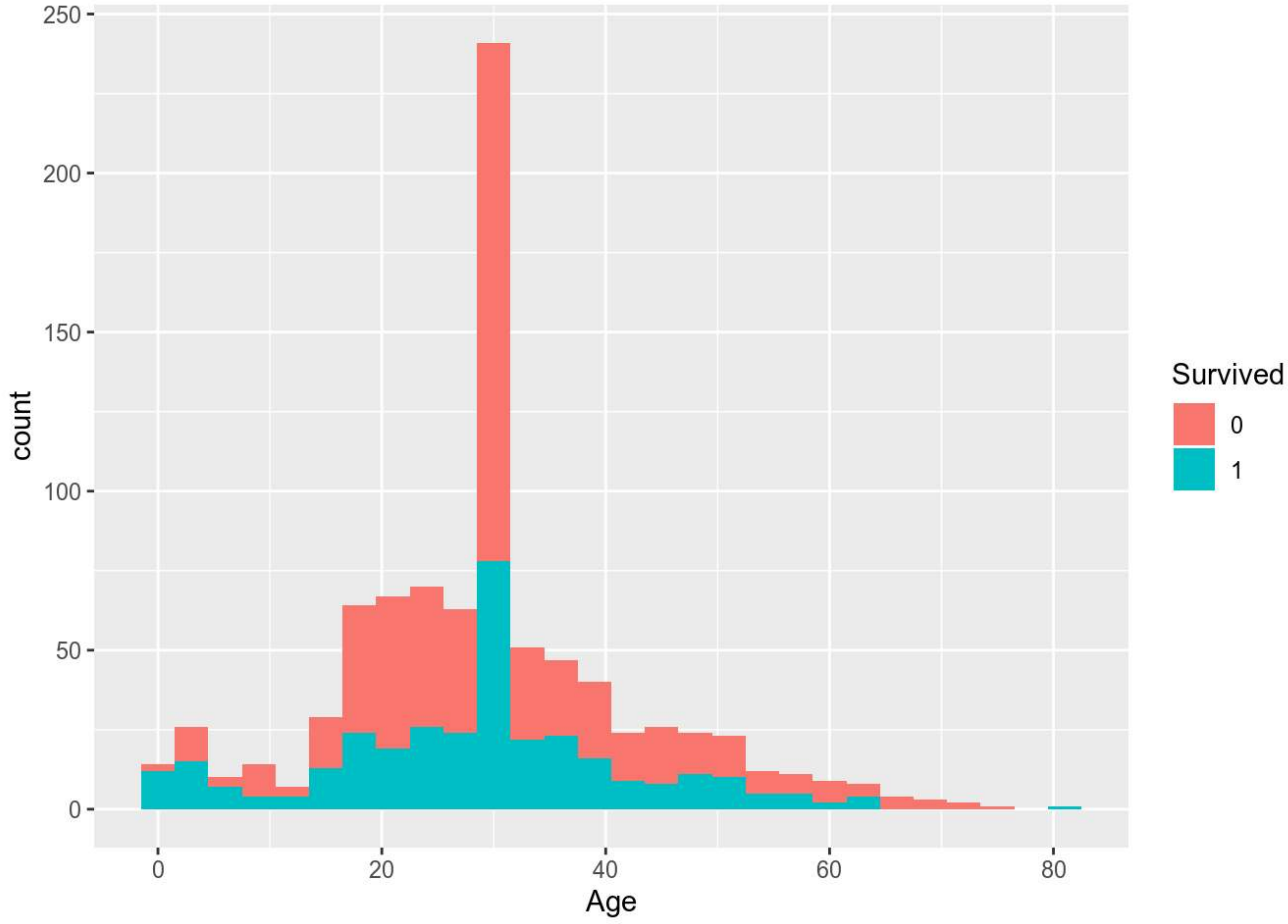
```
# Construimos un atributo nuevo: family size.
totalData$FamilySize <- totalData$SibSp + totalData$Parch +1;
totalData1<-totalData[1:filas,]
ggplot(data = totalData1[!is.na(totalData[1:filas,]$FamilySize),],aes(x=FamilySize,fill=
Survived))+geom_histogram(binwidth =1,position="fill")+ylab("Frecuencia")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



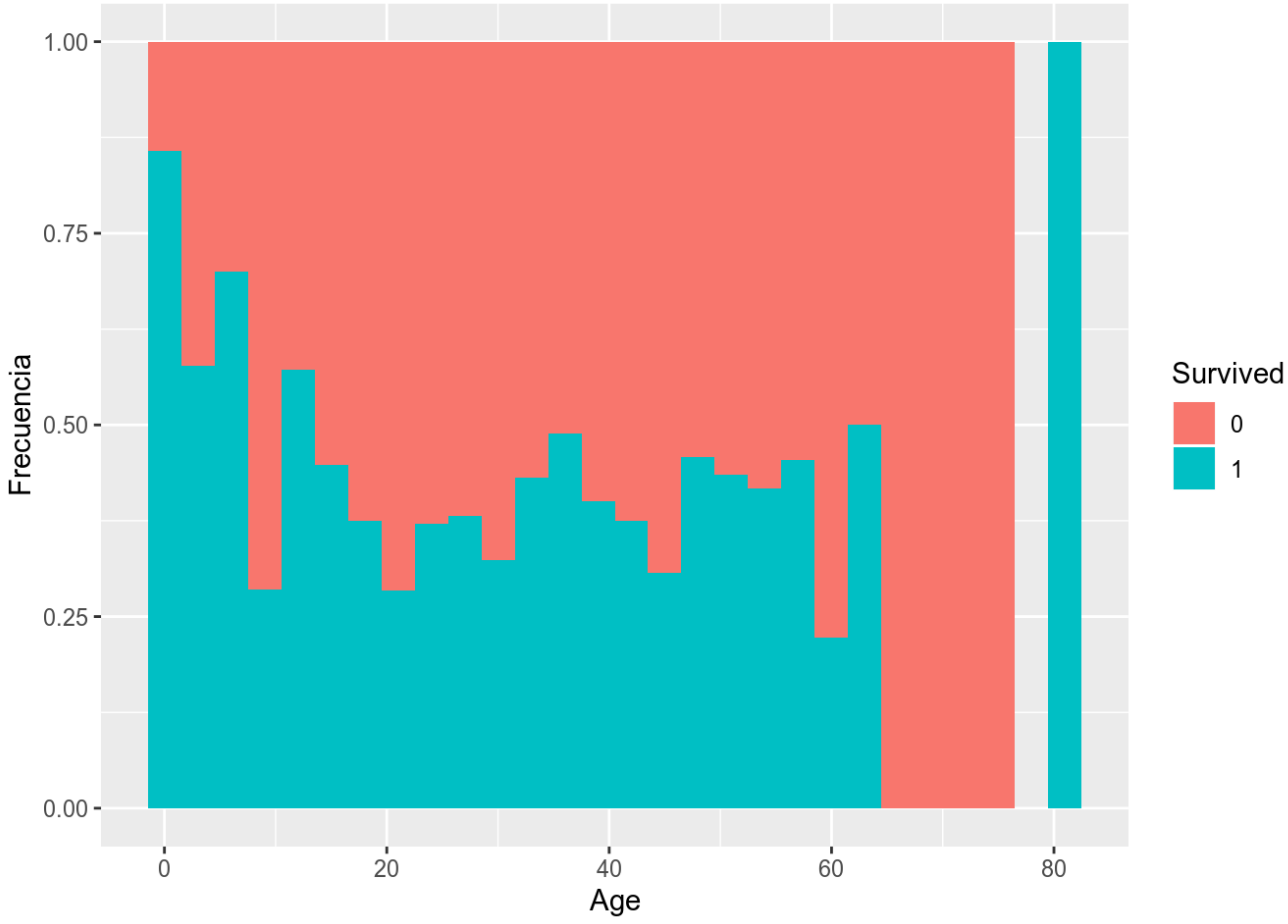

```
# Observamos como familias de entre 2 y 6 miembros tienen más del 50% de posibilidades d
e supervivencia.
```

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))+g
eom_histogram(binwidth =3)
```




```
ggplot(data = totalData1[!is.na(totalData[1:filas,]$Age),],aes(x=Age,fill=Survived))+geo
m_histogram(binwidth = 3,position="fill")+ylab("Frecuencia")
```



4 Ejercicios

4.1 Ejercicio 1:

Estudia los tres casos siguientes y contesta, de forma razonada la pregunta que se realiza:

- Disponemos de un conjunto de variables referentes a vehículos, tales como la marca, modelo, año de matriculación, etc. También se dispone del precio al que se vendieron. Al poner a la venta a un nuevo vehículo, se dispone de las variables que lo describen, pero se desconoce el precio. ¿Qué tipo de algoritmo se debería aplicar para predecir de forma automática el precio?
- En un almacén de naranjas se tiene una máquina, que de forma automática obtiene un conjunto de variables de cada naranja, como su tamaño, acidez, grado maduración, etc. Si se desea estudiar las naranjas por tipos, según las variables obtenidas, ¿qué tipo de algoritmo es el más adecuado?
- Un servicio de música por internet dispone de los historiales de audición de sus clientes: Qué canciones y qué grupos eligen los clientes a lo largo del tiempo de sus escuchas. La empresa desea crear un sistema que proponga la siguiente canción y grupo en función de la canción que se ha escuchado antes. ¿Qué tipo de algoritmo es el más adecuado?

4.1.1 Respuesta 1:

A. Algoritmo de regresión lineal múltiple. La regresión lineal es una técnica estadística que intenta construir un modelo para los datos analizados, y a través de éste predecir los datos futuros. Cuando la salida que perseguimos predecir depende de más de una variable, se puede utilizar un modelo más complejo que tenga en cuenta las dimensiones adicionales. Considerando si son relevantes o no para abordar el problema planteado, el uso de más variables puede contribuir a conseguir mejores predicciones. En este caso disponemos de las variable marca, modelo, año de matriculacion del coche, etc. que nos proporcionará los datos para elvaborar un algorítmo de regresión lineal para poder predecir su precio.

B. Algoríto de agregación (Clustering). Éste modelo consiste en encontrar similitudes y agrupar objetos parecidos. Corresponde a un proyecto en el que tenemos “poca” información del dominio y queremos empezar a tener una idea más clara al respecto. En este caso, tenemos un conjunto de variables relativos a las naranjas, para clasificarlos por tippo, si tener una idea inicial de como clasificarlos. Por ello, los modelos típicos para alcanzar estos objetivos son los modelos de agregación (clustering) procedentes del análisis de datos o del aprendizaje automático y los modelos asociativos.

c. Algorítmo de Arbol de decisión Los árboles de decisión ofrecen una estructura en la que en cada nodo se hace una pregunta sobre un atributo determinado. El valor que tome indica que hay que seguir la rama correspondiente al atributo. Los nodos finales corresponden a conjuntos de ejemplos que pertenecen a la misma clase. Si seguimos las ramas desde la raíz hasta las hojas, se obtiene una serie de condiciones que permiten clasificar las nuevas observaciones. En este caso, sabemos a traves del historial de audición de los clientes, la ruta que usan a la hora de elegir las canciones, por ello podemos anañizar qué caminos son los más habituales y así predecir qué canción puede querer un cliente después de una canción y así proponérsela.

4.2 Ejercicio 2:

A partir del conjunto de datos disponible en el siguiente enlace http://archive.ics.uci.edu/ml/datasets/Adult (http://archive.ics.uci.edu/ml/datasets/Adult) , realiza un estudio tomando como propuesta inicial al que se ha realizado con el conjunto de datos “Titanic”. Amplia la propuesta generando nuevos indicadores o solucionando

otros problemas expuestos en el módulo 2. Explica el proceso que has seguido, qué conocimiento obtienes de los datos, qué objetivo te has fijado y detalla los pasos, técnicas usadas y los problemas resueltos.

Nota: Si lo deseas puedes utilizar otro conjunto de datos propio o de algun repositorio open data siempre que sea similar en diversidad de tipos de variables al propuesto.

4.2.1 Respuesta 2:

```
# Cargamos el juego de datos
datosAdult <- read.csv('http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data',sep = ',', fill = F, strip.white = T)

# Nombres de los atributos
names(datosAdult) <- c("age","workclass","fnlwgt","education","education-num","marital-status","occupation","relationship","race","sex","capital-gain","capital-loss","hour-per-week","native-country","income")
```

4.3 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)
library(plyr)

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

```
## 'data.frame':    32560 obs. of  15 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : Factor w/ 9 levels "?","Federal-gov",..: 7 5 5 5 5 5 7 5 5 5 ...
##  $ fnlwgt        : int  83311 215646 234721 338409 284582 160187 209642 45781 159449
280464 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 10 12 2 10 13 7 12 13 10 16
...
##  $ education-num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital-status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 3 1 3 3 3 4
3 5 3 3 ...
##  $ occupation    : Factor w/ 15 levels "?","Adm-clerical",..: 5 7 7 11 5 9 5 11 5 5
...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 1 2 1 6 6 2 1 2 1
1 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 3 5 3 5 5 5 3
...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 1 2 2 ...
##  $ capital-gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital-loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hour-per-week : int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native-country: Factor w/ 42 levels "?","Cambodia",..: 40 40 40 6 40 24 40 40 40 4
0 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 2 2 2 2 ...
```

```
#El conjunto de datos contiene información anónima como la edad, ocupación, aducación, w
orking class, etc. El objetivo será construir un algoritmo para predecir el income de ca
da individuo, que tiene dos valores posible '>50K' y '<50K'. Hay 32561 instancias y 15 a
tributos en el dataset. Los datos contienen datos categóricos, numéricos y valores desco
nocidos.
```

Descartamos los campos que no son útiles para el análisis

```
#Para simplificar el análisis, vamos a descartar algunos campos que no son útiles. Educa
tion se puede sustituir por la columna education-num, ya que significa el número de curs
o obtenido y son equivalentes.
#fnlwgt es el peso de la muestra, no está relacionado con el la variable objetivo incom
e. Por lo tanto descartamos esas columnas

datosAdult$education <- NULL
datosAdult$fnlwgt <- NULL
datosAdult$relationship <- NULL

str(datosAdult)
```

```
## 'data.frame':    32560 obs. of  12 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : Factor w/ 9 levels "?","Federal-gov",..: 7 5 5 5 5 5 7 5 5 5 ...
##  $ education-num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital-status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 3 1 3 3 3 4
3 5 3 3 ...
##  $ occupation    : Factor w/ 15 levels "?","Adm-clerical",..: 5 7 7 11 5 9 5 11 5 5
...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 3 5 3 5 5 5 3
...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 1 2 2 ...
##  $ capital-gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital-loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hour-per-week : int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native-country: Factor w/ 42 levels "?","Cambodia",..: 40 40 40 6 40 24 40 40 40 4
0 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 2 2 2 2 ...
```

Trabajamos los atributos con valores vacíos.

```
# Estadísticas de valores vacíos/desconocidos
colSums(datosAdult=="?")
```

```
##            age      workclass  education-num marital-status     occupation
##              0           1836              0              0           1843
##           race            sex   capital-gain   capital-loss hour-per-week
##              0              0              0              0              0
## native-country         income
##            583              0
```

```
#Función para calcular el valor más común (poner Referencias)
common_value <- function(x) {
  uniqx <- unique(na.omit(x))
  uniqx[which.max(tabulate(match(x, uniqx)))]
}
#Creamos variables para obtener el valor más column de las 3 columnas que tienen valores
desconocidos
workclass_df <- common_value(datosAdult$workclass)
occupation_df <- common_value(datosAdult$occupation)
country_df <- common_value(datosAdult$`native-country`)

#Sustituir los datos desconocidos
datosAdult$occupation[datosAdult$occupation=="?"]=occupation_df
datosAdult$workclass[datosAdult$workclass=="?"]=workclass_df
datosAdult$`native-country`[datosAdult$`native-country`=="?"]=country_df
```

Discretizamos cuando tiene sentido y en función de cada variable.

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

```
##            age      workclass  education-num marital-status     occupation
##             73              8             16              7             14
##           race            sex   capital-gain   capital-loss  hour-per-week
##              5              2            119             92             94
## native-country         income
##             41              2
```

```
# Discretizamos las variables con pocas clases
cols<-c("native-country","race","sex","income")
for (i in cols){
  datosAdult[,i] <- as.factor(datosAdult[,i])
}

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

```
## 'data.frame':    32560 obs. of  12 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : Factor w/ 9 levels "?","Federal-gov",..: 7 5 5 5 5 5 7 5 5 5 ...
##  $ education-num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital-status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 3 1 3 3 3 4
3 5 3 3 ...
##  $ occupation    : Factor w/ 15 levels "?","Adm-clerical",..: 5 7 7 11 5 9 5 11 5 5
...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 3 5 3 5 5 5 3
...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 1 2 2 ...
##  $ capital-gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital-loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hour-per-week : int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native-country: Factor w/ 42 levels "?","Cambodia",..: 40 40 40 6 40 24 40 40 40 4
0 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 2 2 2 2 ...
```

La variable education-num tiene demasiados valores para el análisis, con lo que vamos a agrupar los 16 niveles de estudio en 4 niveles.

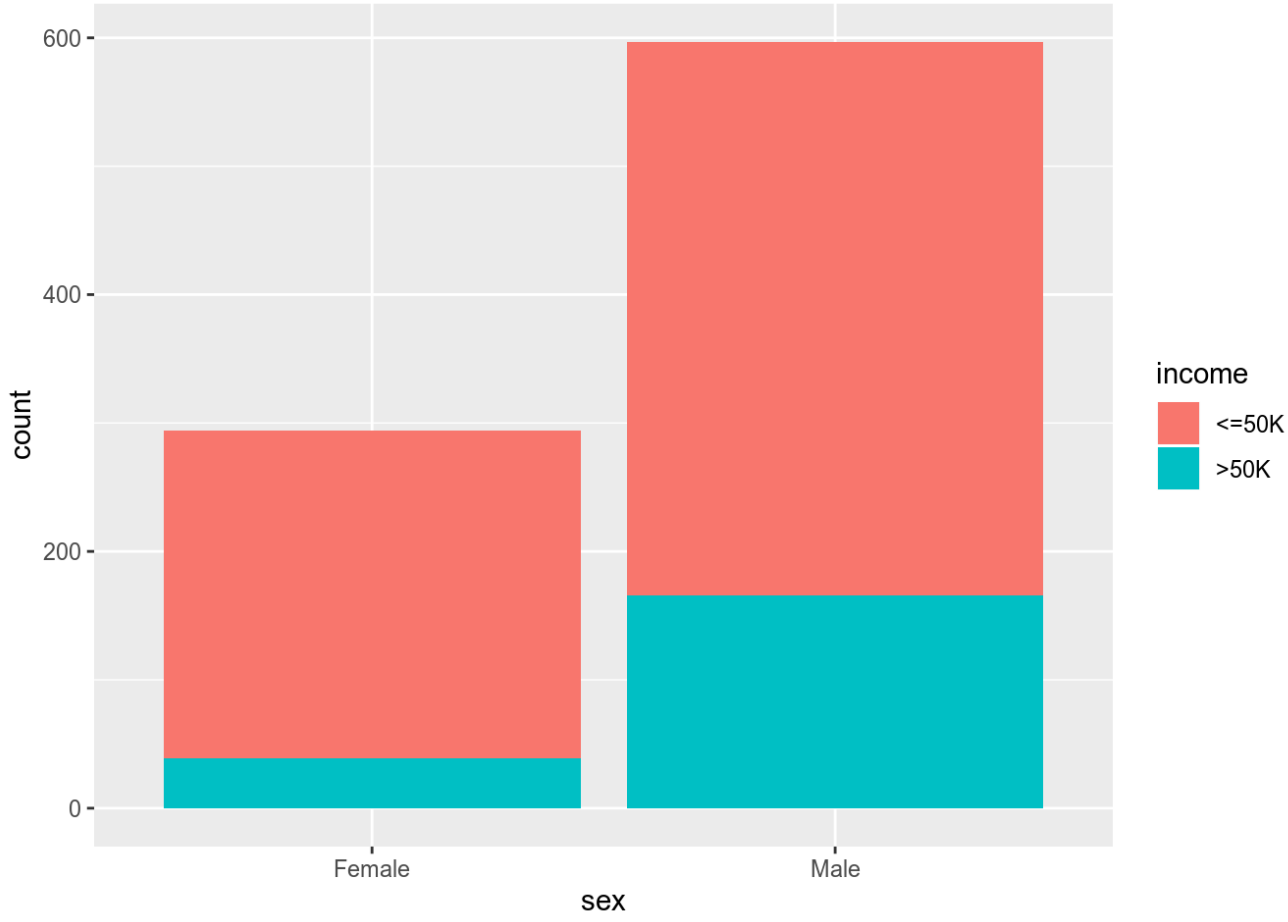
```
# 1-4: "No Education"; 5-8: "Basic"; 9-12:"Medium"; 13-16:"High"
datosAdult$`education-num`[between(datosAdult$`education-num`,1,4)] <- 1
datosAdult$`education-num`[between(datosAdult$`education-num`,5,8)] <- 2
datosAdult$`education-num`[between(datosAdult$`education-num`,9,12)] <- 3
datosAdult$`education-num`[between(datosAdult$`education-num`,13,16)] <- 4

str(datosAdult)
```

```
## 'data.frame':    32560 obs. of  12 variables:
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : Factor w/ 9 levels "?","Federal-gov",..: 7 5 5 5 5 5 7 5 5 5 ...
##  $ education-num : num  4 3 2 4 4 2 3 4 4 3 ...
##  $ marital-status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 3 1 3 3 3 4
3 5 3 3 ...
##  $ occupation    : Factor w/ 15 levels "?","Adm-clerical",..: 5 7 7 11 5 9 5 11 5 5
...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 3 5 3 5 5 5 3
...
##  $ sex           : Factor w/ 2 levels "Female","Male": 2 2 2 1 1 1 2 1 2 2 ...
##  $ capital-gain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ capital-loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hour-per-week : int  13 40 40 40 40 16 45 50 40 80 ...
##  $ native-country: Factor w/ 42 levels "?","Cambodia",..: 40 40 40 6 40 24 40 40 40 4
0 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 2 2 2 2 ...
```

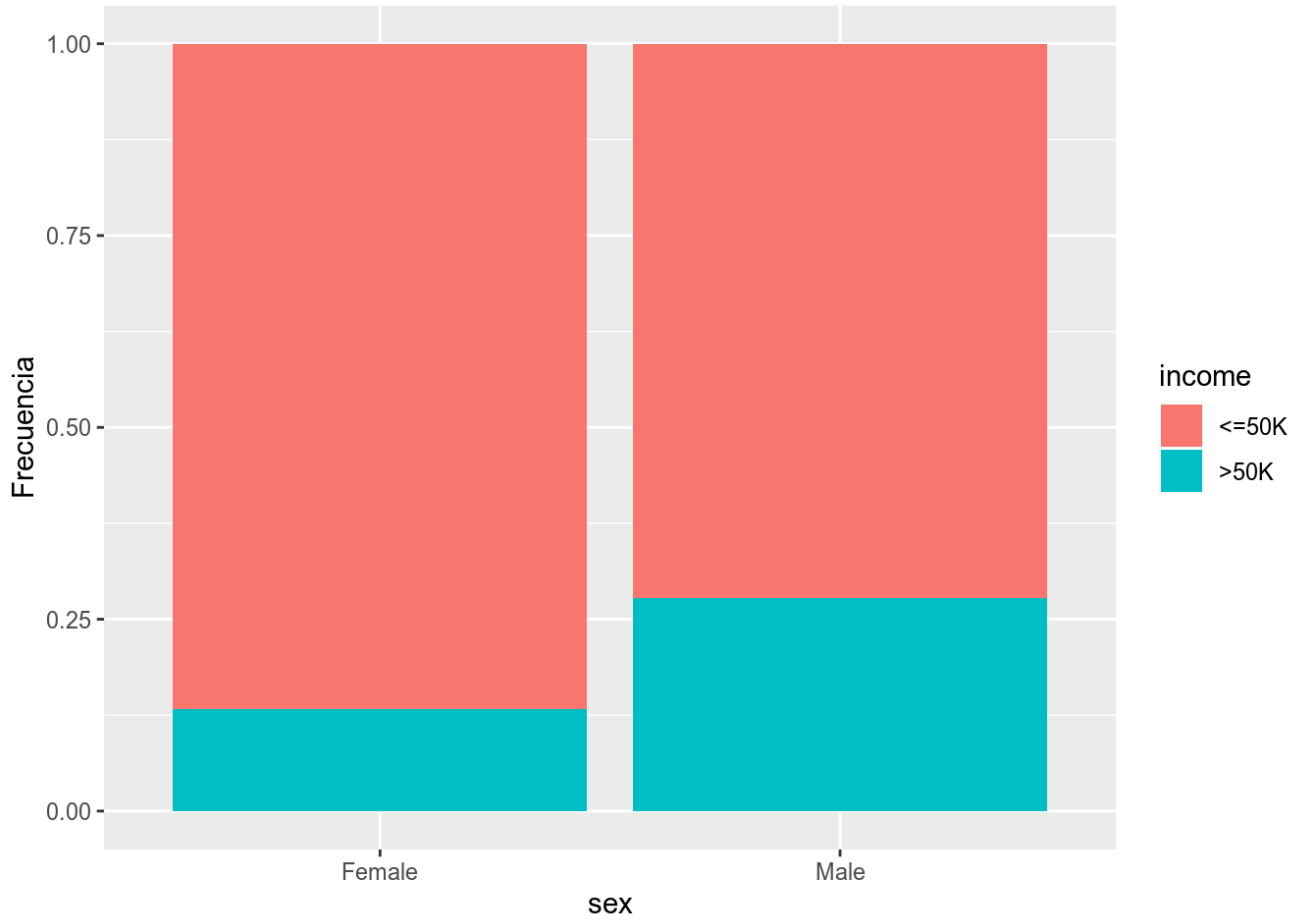
4.4 Procesos de análisis del conjunto de datos

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

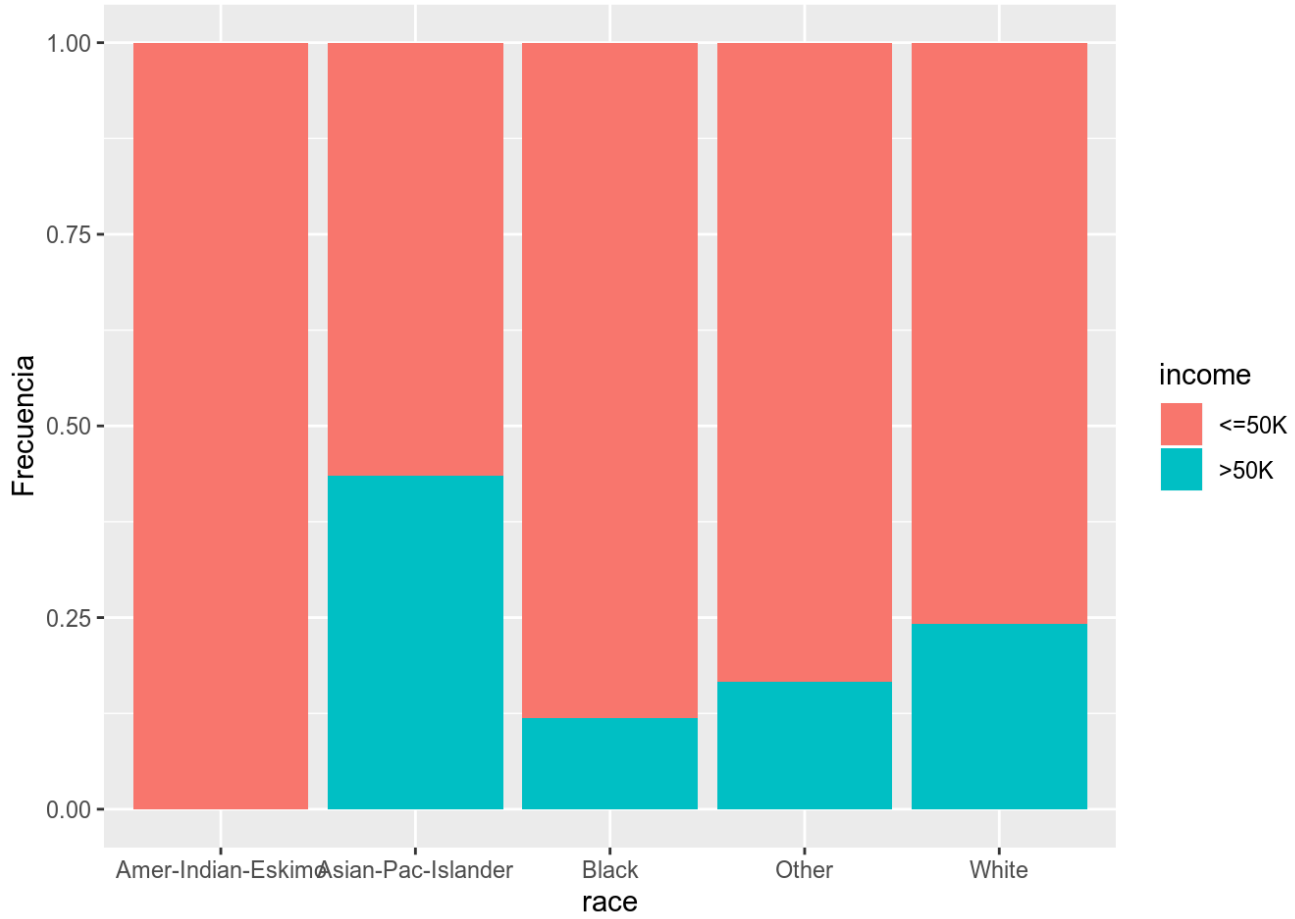
```
# Visualizamos la relación entre las variables "sex" y "income"
ggplot(data=datosAdult[1:filas,],aes(x=sex,fill=income))+geom_bar()
```



```
# Otro punto de vista. income como función de sex: Observamos que las mujeres cobran en
 porcentaje menos que los hombres
ggplot(data=datosAdult[1:filas,],aes(x=sex,fill=income))+geom_bar(position="fill")+ylab(
"Frecuencia")
```




```
# Otro punto de vista. income como función de race: Observamos diferencias significativa
s en función de raza, Los de Asia pacífica son los que más income tienen mientras que de
raza india americana ninguno supera 50K.
ggplot(data = datosAdult[1:filas,],aes(x=race,fill=income))+geom_bar(position="fill")+yl
ab("Frecuencia")
```



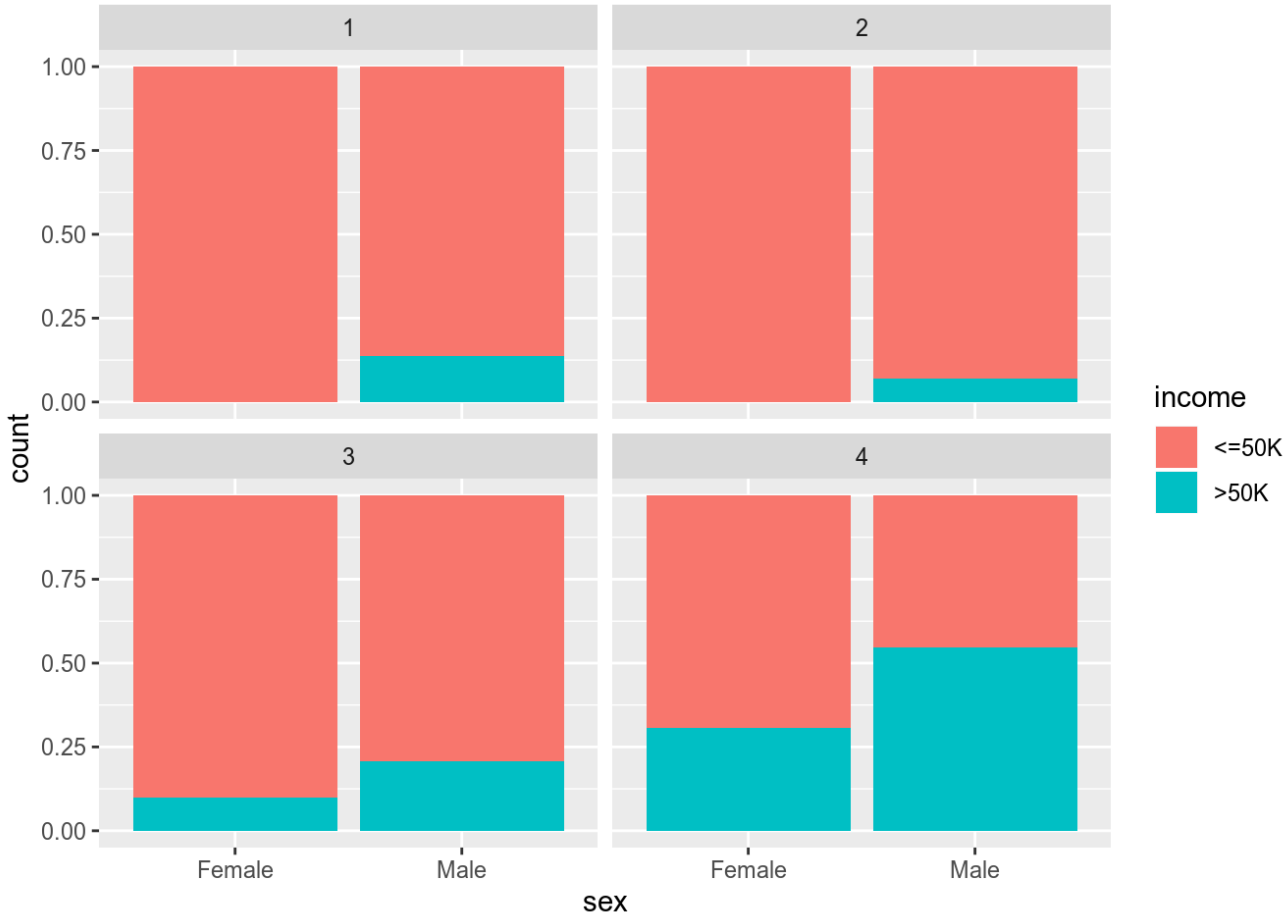
Obtenemos una matriz de porcentages de frecuencia.
Vemos, por ejemplo que la probabilidad de tener un income >50K siendo mujer es 13,26%

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

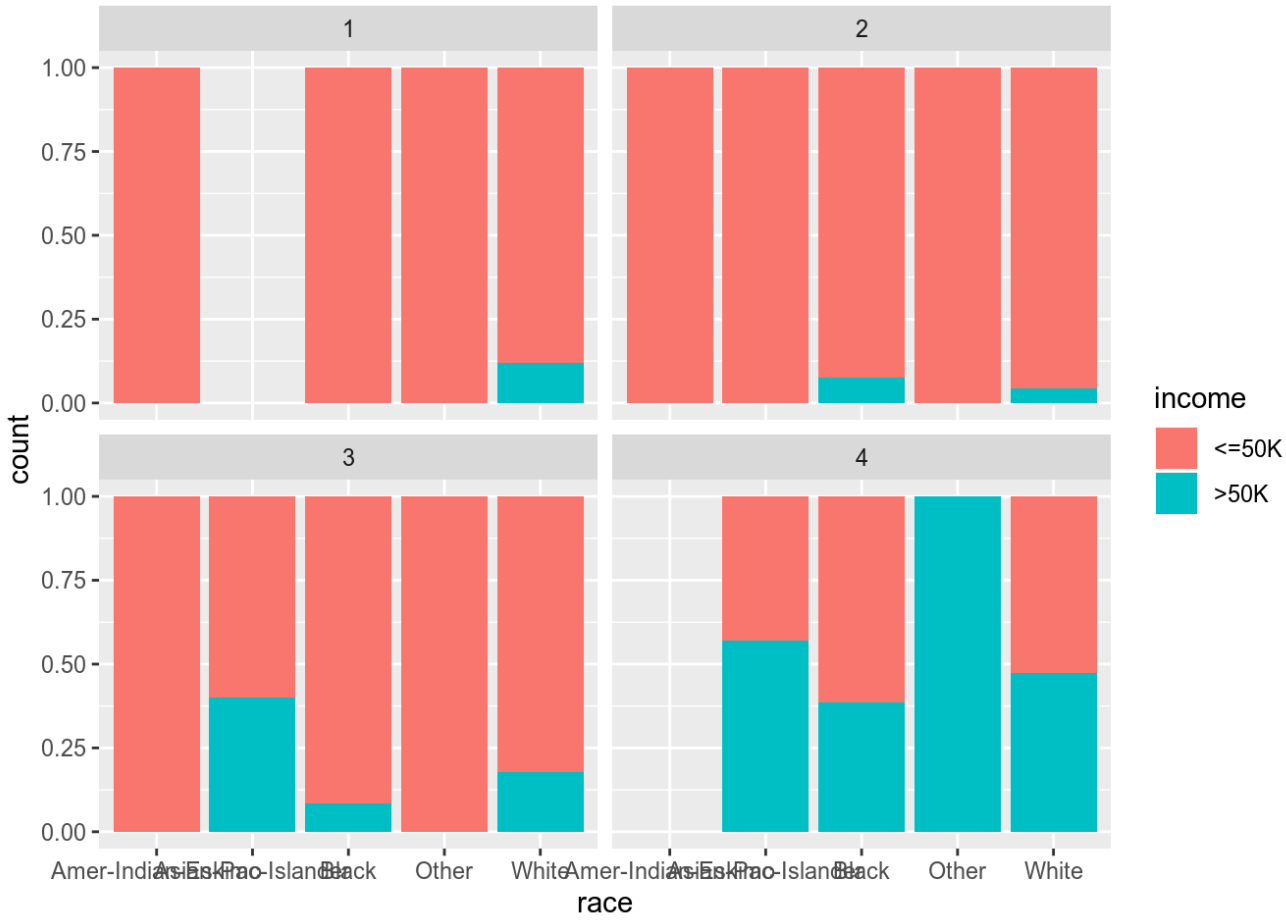
```
##
##             <=50K     >50K
##   Female 86.73469 13.26531
##   Male   72.19430 27.80570
```

Veamos ahora como en un mismo gráfico de frecuencias podemos trabajar con 3 variables: sex, income y education-num

```
# Ahora, podemos dividir el gráfico de Embarked por Pclass:
ggplot(data = datosAdult[1:filas,],aes(x=sex,fill=income))+geom_bar(position="fill")+fac
et_wrap(~`education-num`)
```




```
ggplot(data = datosAdult[1:filas,],aes(x=race,fill=income))+geom_bar(position="fill")+facet_wrap(~`education-num`)
```



```
#Podemos observar que el atributo sexo y raza influye claramente en el income, ya que las mujeres tienen menor income incluso con el mismo nivel de educación.
#Lo mismo ocurre con la raza, con un mismo nivel de educación hay diferencias según la raza.
```

Veamos un ejemplo de construcción de una variable nueva: Industry

```
# Construimos un atributo nuevo: industry Es la workclass agrupados en 4 grupos. Government, Self-Employed, Private, Other/Unknown

# Agrupamos los trabajos de gobierno

datosAdult$industry[datosAdult$workclass == "Federal-gov"]  <- "Government"
datosAdult$industry[datosAdult$workclass == "Local-gov"]  <- "Government"
datosAdult$industry[datosAdult$workclass == "State-gov"]  <- "Government"

# Agrupamos los Sele-Employed
datosAdult$industry[datosAdult$workclass == "Self-emp-inc"]  <- 'Self-Employed'
datosAdult$industry[datosAdult$workclass == "Self-emp-not-inc"]  <- 'Self-Employed'

# Agrupamos Otros y desconocidos Other/Unknown
datosAdult$industry[datosAdult$workclass == "Never-worked"]  <- 'Other/Unknown'
datosAdult$industry[datosAdult$workclass == "Without-pay"]  <- 'Other/Unknown'
datosAdult$industry[datosAdult$workclass == "Other"]  <- 'Other/Unknown'
datosAdult$industry[datosAdult$workclass == "Other/Unknown"]  <- 'Other/Unknown'

# Asignamos Private a la nueva columna
datosAdult$industry[datosAdult$workclass == "Private"]  <- "Private"

datosAdult$industry <- as.factor(datosAdult$industry)

summary(datosAdult$industry)
```

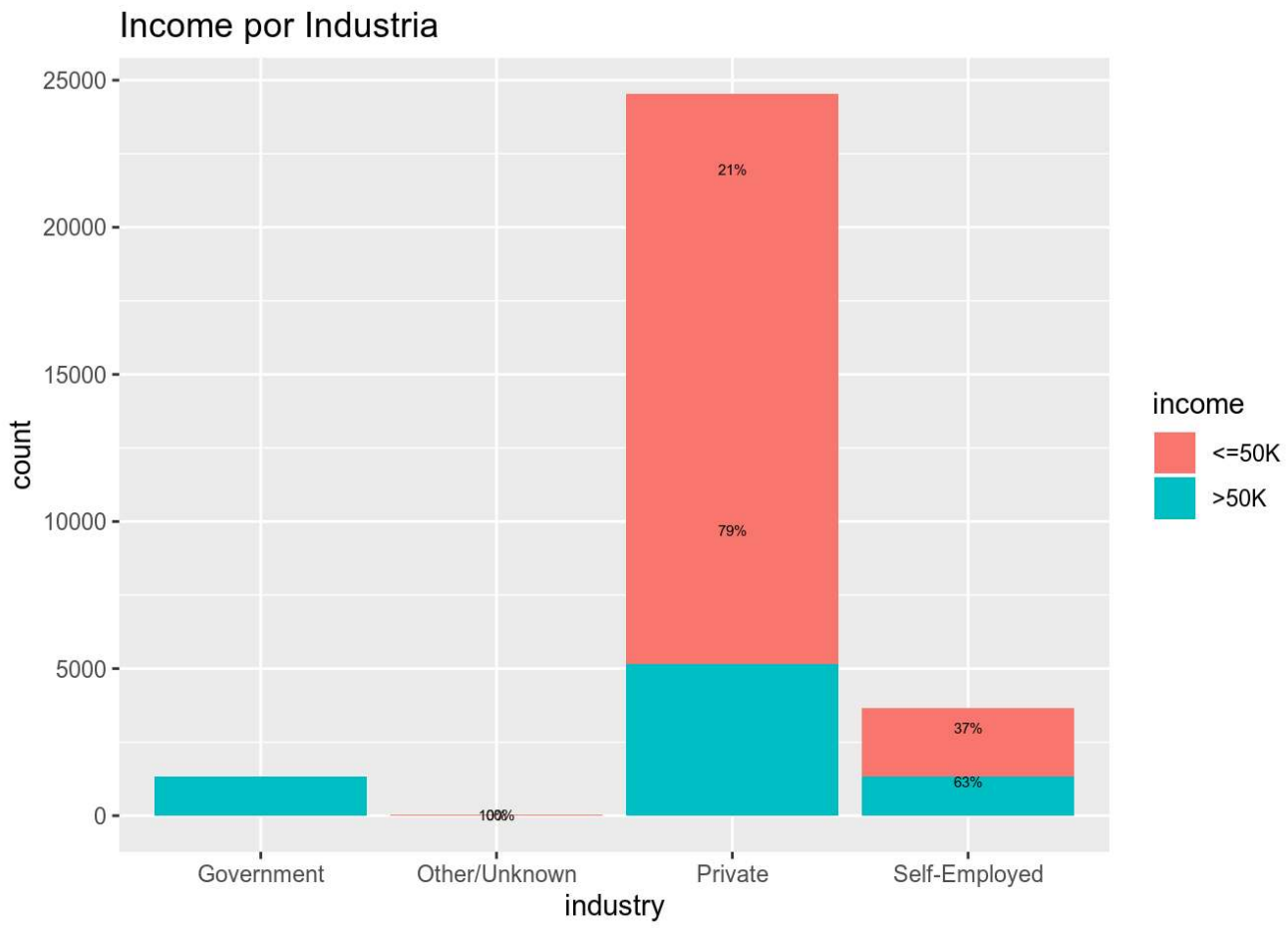
```
##    Government Other/Unknown       Private Self-Employed
##          4350            21         24532          3657
```

Para explorar la relación entre industria y income, hacemos la cuenta para de cada una de las dos categorías income en las distintas industrias. Los resultados están representados en un bar plot

```
# barplot de tipo de trabajo por tipo de income
# Hacemos la cuenta de industria por tipo de income
count <- table(datosAdult[datosAdult$industry == 'Government',]$income)["<=500K"]
count <- c(count, table(datosAdult[datosAdult$industry == 'Government',]$income)[">50K"
])
count <- c(count, table(datosAdult[datosAdult$industry == 'Other/Unknown',]$income)["<=5
0K"])
count <- c(count, table(datosAdult[datosAdult$industry == 'Other/Unknown',]$income)[">50
K"])
count <- c(count, table(datosAdult[datosAdult$industry == 'Private',]$income)["<=50K"])
count <- c(count, table(datosAdult[datosAdult$industry == 'Private',]$income)[">50K"])
count <- c(count, table(datosAdult[datosAdult$industry == 'Self-Employed',]$income)["<=5
0K"])
count <- c(count, table(datosAdult[datosAdult$industry == 'Self-Employed',]$income)[">50
K"])
count <- as.numeric(count)

# Creamos un dataframe
industry <- rep(levels(datosAdult$industry), each = 2)
income <- rep(c('<=50K', '>50K'), 4)
df <- data.frame(industry, income, count)
df
```

```
##        industry income count
## 1    Government  <=50K    NA
## 2    Government   >50K  1341
## 3 Other/Unknown  <=50K    21
## 4 Other/Unknown   >50K     0
## 5       Private  <=50K 19378
## 6       Private   >50K  5154
## 7 Self-Employed  <=50K  2311
## 8 Self-Employed   >50K  1346
```

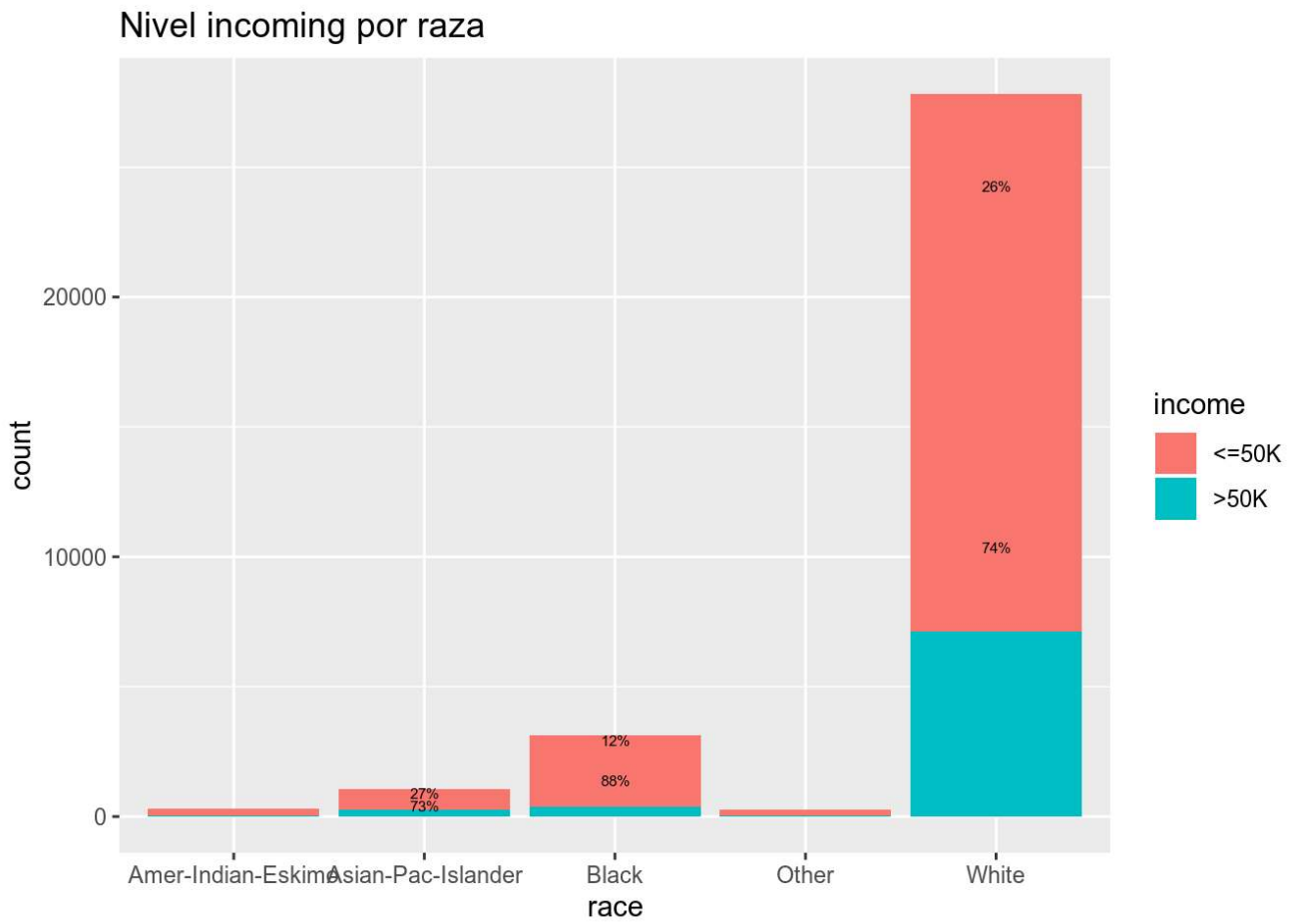
```
# Calculamos los porcentajes

df <- ddply(df, .(industry), transform, percent = count/sum(count) * 100)

# Formateamos las etiquetas y calculamos sus posiciones
df <- ddply(df, .(industry), transform, pos = (cumsum(count) - 0.5 * count))
df$label <- paste0(sprintf("%.0f", df$percent), "%")

# bar plot Income por industia
ggplot(df, aes(x = industry, y = count, fill = income)) +
  geom_bar(stat = "identity") +
  geom_text(aes(y = pos, label = label), size = 2) +
  ggtitle('Income por Industria')
```




```
#Podemos apreciar que el 100% de los que trabajan para el gobierno ganan más de 50K, des
pués los autónomos tienen los que más probabilidad de tener un income > 50K
```

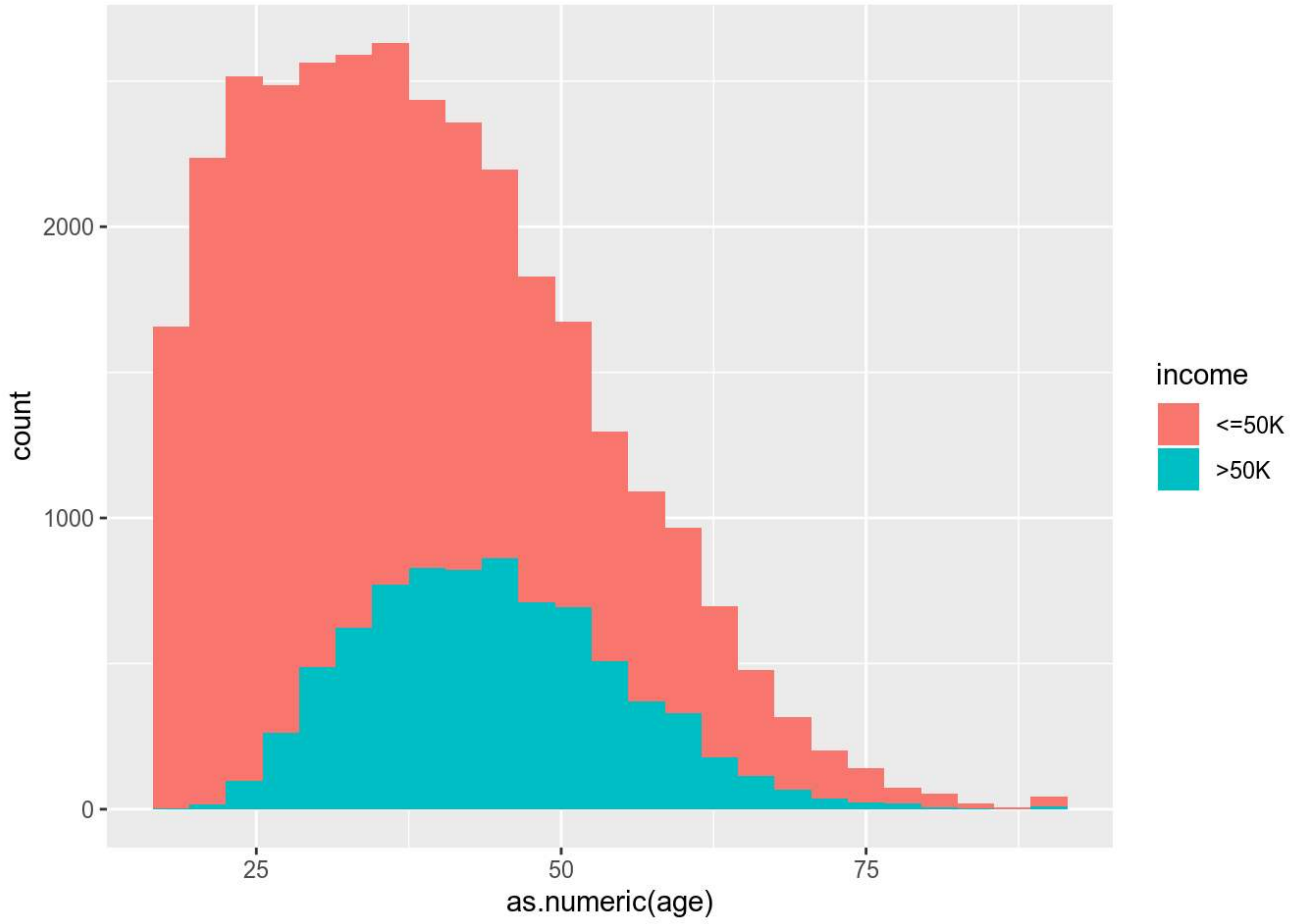
race es una variable categorica. El gráfico muestra que los bñancos y los de Asia-pacçifica tienen más potencial de ganar dinero, porque sobre el 25% de esas razas generan más de 50K anualmente.

```
df2 <- data.frame(table(datosAdult$income, datosAdult$race))
names(df2) <- c('income', 'race', 'count')

# Calculamos los porcentajes
df2 <- ddply(df2, .(race), transform, percent = count/sum(count) * 100)

# Formateamos las etiquetas y calculamos sus posiciones
df2 <- ddply(df2, .(race), transform, pos = (cumsum(count) - 0.5 * count))
df2$label <- paste0(sprintf("%.0f", df2$percent), "%")

# No mostramos porcentajes para categorías con pocas unidades
df2$label[df2$race == 'Other'] <- NA
df2$label[df2$race == 'Amer-Indian-Eskimo'] <- NA

# bar plot
ggplot(df2, aes(x = race, y = count, fill = income)) +
  geom_bar(stat = "identity") +
  geom_text(aes(y = pos, label = label), size = 2) +
  ggtitle('Nivel incoming por raza')
```



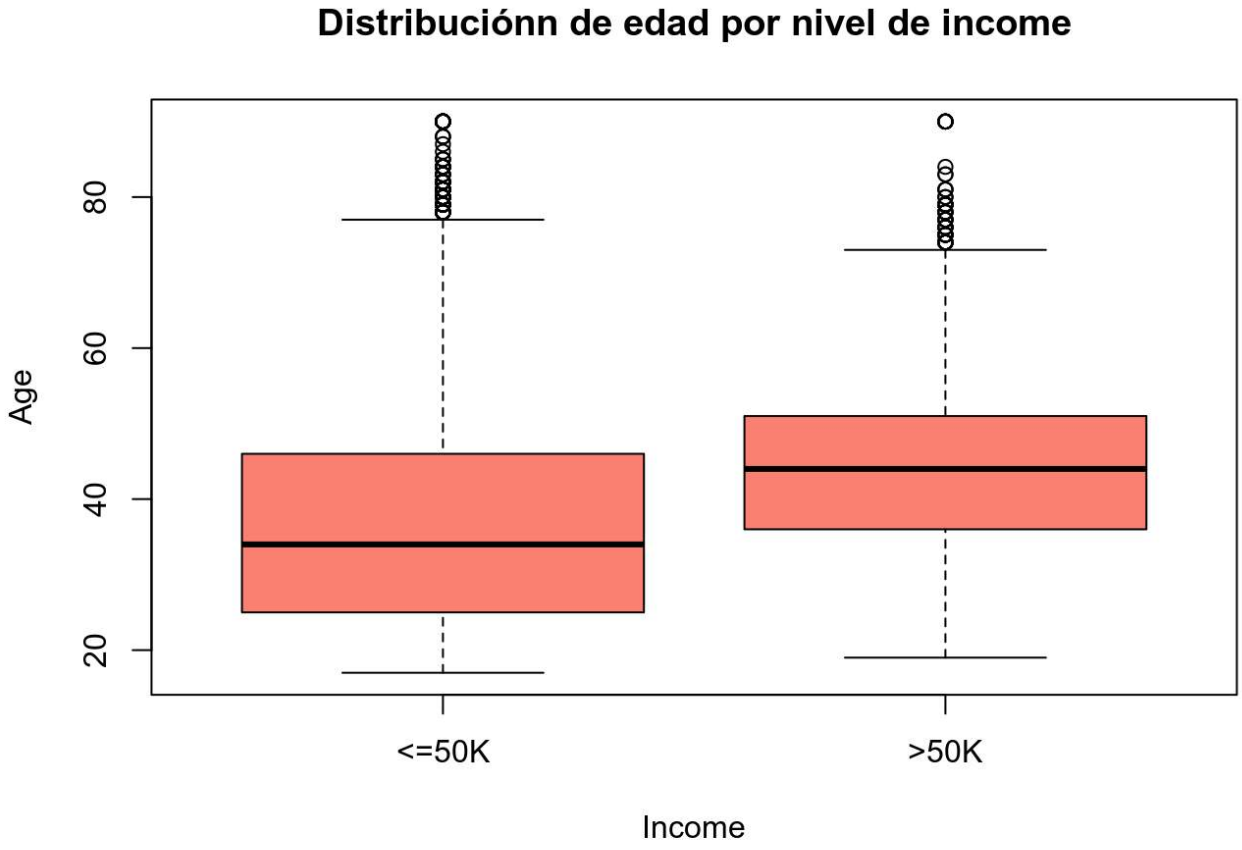
Vamos ahora a analizar la variable continua edad.

```
# histograma edad por income
ggplot(datosAdult) + aes(x=as.numeric(age), group=income, fill=income) + geom_histogram
(binwidth = 3)
```



Ahora realizamos el mismo estudio representado con un diagrama de cajas

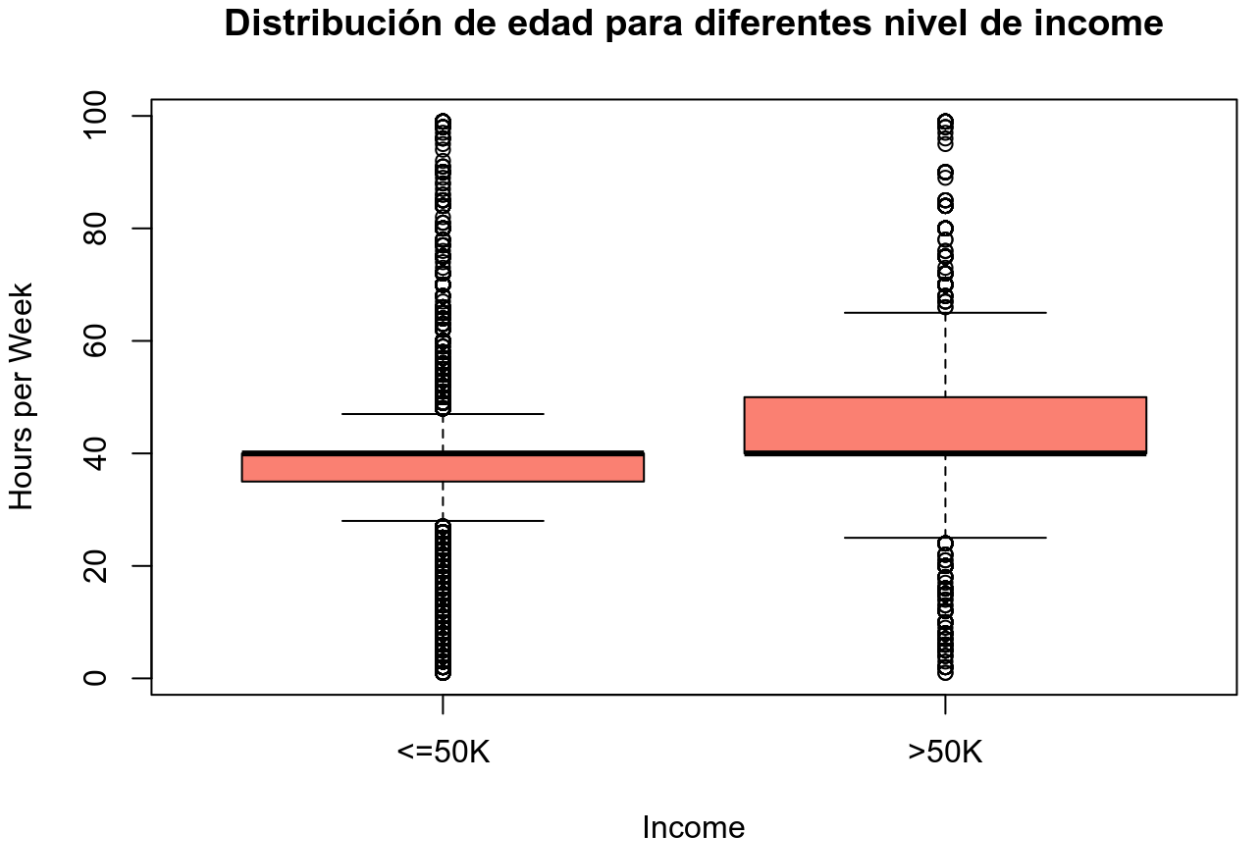
```
# Diagrama de Cajas age  por income
boxplot (age ~ income, datosAdult,
     main = "Distribucióonn de edad por nivel de income",
     xlab = "Income", ylab = "Age", col = "salmon")
```




```
# La mayoría de los adultos que trabajan están entre 25 y 65 años. Segú el diagrama, pod
emos observar que los adultos menores de 30 años ganan menos de 50K mientras que los de
 más de 43 años ganan más de 50K mayormente. Esto puede demostrar que el tener más exper
iencia afecta a la hora de ganar más dinero.
```

Realizamos el mismo tipo de gráfica para estudiar la variable horas trabajadas.

```
# Diagrama de Cajas hoursperweek por income
boxplot (`hour-per-week` ~ income, datosAdult,
     main = "Distribución de edad para diferentes nivel de income",
     xlab = "Income", ylab = "Hours per Week", col = "salmon")
```




```
# Como es evidente aquellos que invierten más tiempo en el lugar de trabajo tienden a ga
nar más y viceversa
```

4.5 Conclusión

En este estudio el objetivo a sido predecir el income de una persona en función de su educación, edad, sexo, raza y otros valores. Hemos comprobado datos curiosos como que el sexo y la raza afectan a la hora de tener más o menos ingresos a pesar de tener un mismo nivel de educación. También hemos analizado otros datos más intuitivos como que a más edad mayor probabilidad de tener un income >50K probablemente debido al valor añadido de la experiencia, así como que a más horas trabajadas más se gana. Para ello hemos tratado los datos nulos, hemos discretizado algunas variables, creado nuevos atributos y quitando otros que no son relevantes con el atributo objetivo income. En este estudio faltaría un último paso que es el ajuste del modelo con técnicas de machine learning, como podrían ser Regresión Lógica, Redes neuronales o Random Forest,etc. Estudiando esas téncicas podríamos analizar cual es el mejor modelo para una predicción de income con el resto de atributos.