

LIMPIEZA Y ANÁLISIS DE DATOS. Práctica 2

Alumno: Pablo Mas Cayuelas

1. Descripción del dataset

El conjunto de datos seleccionado corresponde al censo de adultos de Estados Unidos. Es un repositorio de más de 48000 entradas del año 1994, que nos da información tanto económica como social de los individuos de la muestra, como por ejemplo su raza, su nivel de estudios o los ingresos de los mismos. Así, la base de datos nos resultará útil para realizar predicciones sobre los ingresos que va a tener un individuo con unas características dadas, o bien para ver qué influencia tiene una característica concreta en los ingresos del individuo. La base de datos ha sido escogida de la web “kaggle.com” (enlace: <https://www.kaggle.com/serpilturanyksel/adult-income>). El fichero adjunto se llama “adult11.csv”.

Los atributos de nuestra base de datos son los siguientes:

- age: la edad de un individuo. Es un número natural.
- workclass: término para presentar la situación laboral del individuo. Posibles valores: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- fnlwgt: peso final. Es el número de personas que el censo cree que la entrada representa. Número natural.
- education: nivel de educación más alto alcanzado por el individuo. Posibles valores: Bachelors, Somecollege, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- education-num: el nivel educativo más alto alcanzado en forma numérica. Número natural.
- marital-status: estado civil de un individuo. Posibles valores: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- occupation: trabajo del individuo. Posibles valores: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- relationship: tipo de relación sentimental del individuo. Posibles valores: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- race: raza del individuo. Posibles valores: White, ASian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- sex: sexo biológico del individuo. Posibles valores: Male, Female.
- capital-gain: ganancias del individuo. Número natural.
- capital-loss: pérdidas del individuo. Número natural.
- hours-per-week: horas trabajadas a la semana. Número natural.
- native-country: país de origen del individuo. Posibles valores: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France,

Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad&Tobago, Peru, Hong, Holland-Netherlands.

- salary: salario del individuo (es nuestro target). Dos posibles valores: <=50k o >50k.

1.1. Importancia y objetivos

La idea general de nuestro trabajo es ver qué variables son más importantes para que un individuo gane más o menos de 50000 dólares anuales. Para ello, emplearemos modelos de regresión predictivos, así como tests chi-square, tras haber realizado previamente una visión general del conjunto de datos.

Hemos de decir también que habrá algunas variables que descartaremos para la predicción, bien sea por ser redundantes, o bien porque pueden tomar muchos valores diferentes que dificultan considerablemente el análisis.

2. Limpieza de datos

Antes de comenzar con la limpieza, cargamos el conjunto de datos en formato CSV:

```
# carga de los datos
datos = read.csv("adult11.csv", header = TRUE)
head(datos)

##   age workclass fnlwgt   education education.num marital.status
## 1 25  Private 226802        11th          7    Never-married
## 2 38  Private  89814       HS-grad         9 Married-civ-spouse
## 3 28 Local-gov 336951 Assoc-acdm        12 Married-civ-spouse
## 4 44  Private 160323 Some-college      10 Married-civ-spouse
## 5 18       ? 103497 Some-college      10    Never-married
## 6 34  Private 198693        10th          6    Never-married
##   occupation relationship race gender capital.gain capital.loss
## 1 Machine-op-inspct   Own-child Black  Male        0         0
## 2 Farming-fishing     Husband White  Male        0         0
## 3 Protective-serv     Husband White  Male        0         0
## 4 Machine-op-inspct   Husband Black  Male      7688         0
## 5       ?   Own-child White Female        0         0
## 6 Other-service Not-in-family White  Male        0         0
##   hours.per.week native.country salary
## 1           40 United-States  <=50K
## 2           50 United-States  <=50K
## 3           40 United-States   >50K
## 4           40 United-States   >50K
## 5           30 United-States  <=50K
## 6           30 United-States  <=50K
```

Podemos ahora ver los tipos de datos que tenemos.

```
# tipos de datos
sapply(datos, function(x) class(x))

##            age   workclass      fnlwgt   education education.num
## "integer" "factor" "integer" "factor" "integer"
## marital.status occupation relationship      race      gender
## "factor" "factor" "factor" "factor" "factor"
## capital.gain capital.loss hours.per.week native.country      salary
```

```
##      "integer"      "integer"      "integer"      "factor"      "factor"
```

Quizá en algún momento necesitamos que la variable “salary” sea numérica en lugar de un factor, pero cuando lo necesitemos cambiaremos su tipo. También podemos ver una descripción más general de los datos:

```
# summary  
summary(datos)
```

```
##      age          workclass      fnlwgt  
##  Min.   :17.00   Private       :33906   Min.   : 12285  
##  1st Qu.:28.00  Self-emp-not-inc: 3862   1st Qu.: 117551  
##  Median :37.00  Local-gov     : 3136   Median : 178145  
##  Mean   :38.64  ?             : 2799   Mean   : 189664  
##  3rd Qu.:48.00  State-gov    : 1981   3rd Qu.: 237642  
##  Max.   :90.00  Self-emp-inc : 1695   Max.   :1490400  
##                (Other)        : 1463  
  
##      education      education.num      marital.status  
##  HS-grad       :15784   Min.   : 1.00   Divorced       : 6633  
##  Some-college  :10878   1st Qu.: 9.00   Married-AF-spouse :  37  
##  Bachelors    : 8025   Median :10.00   Married-civ-spouse :22379  
##  Masters       : 2657   Mean   :10.08   Married-spouse-absent: 628  
##  Assoc-voc    : 2061   3rd Qu.:12.00   Never-married  :16117  
##  11th          : 1812   Max.   :16.00   Separated      : 1530  
##  (Other)        : 7625  
  
##      occupation      relationship      race  
##  Prof-specialty : 6172   Husband       :19716   Amer-Indian-Eskimo:  470  
##  Craft-repair   : 6112   Not-in-family :12583   Asian-Pac-Islander: 1519  
##  Exec-managerial: 6086   Other-relative: 1506   Black           : 4685  
##  Adm-clerical   : 5611   Own-child     : 7581   Other           : 406  
##  Sales          : 5504   Unmarried     : 5125   White           :41762  
##  Other-service   : 4923   Wife          : 2331  
##  (Other)         :14434  
  
##      gender      capital.gain      capital.loss      hours.per.week  
##  Female:16192   Min.   : 0   Min.   : 0.0   Min.   : 1.00  
##  Male  :32650   1st Qu.: 0   1st Qu.: 0.0   1st Qu.:40.00  
##  
##  
##  
##  
##  
##  
##  
##      native.country      salary  
##  United-States:43832  <=50K:37155  
##  Mexico       :  951  >50K :11687  
##  ?            :  857  
##  Philippines  :  295  
##  Germany     :  206  
##  Puerto-Rico  :  184  
##  (Other)      : 2517
```

2.1. Ceros y elementos vacíos

Debemos ver si hay ahora algún valor vacío en nuestros datos:

```
# valores vacíos  
sapply(datos, function(x) sum(is.na(x)))
```

```

##          age    workclass      fnlwgt      education education.num
##          0        0            0            0            0            0
## marital.status occupation relationship      race      gender
##          0        0            0            0            0            0
## capital.gain  capital.loss hours.per.week native.country      salary
##          0        0            0            0            0            0

```

También puede darse el caso en el que tengamos ceros que no tengan sentido. Por tanto, veamos cuáles son los atributos con ceros:

```
# ceros
sapply(datos, function(x) sum(x == "0"))
```

```

##          age    workclass      fnlwgt      education education.num
##          0        0            0            0            0            0
## marital.status occupation relationship      race      gender
##          0        0            0            0            0            0
## capital.gain  capital.loss hours.per.week native.country      salary
##          44807    46560          0            0            0            0

```

Tan solo tenemos ceros en dos atributos, correspondientes a las ganancias y pérdidas. Como tiene sentido que estos atributos tomen valores nulos, no debemos realizar ningún cambio en ellos.

Además, viendo de forma general la base de datos, tenemos algunos valores desconocidos que son representados por signos de interrogación. Veámoslos:

```
# interrogaciones
sapply(datos, function(x) sum(x == "?"))
```

```

##          age    workclass      fnlwgt      education education.num
##          0        2799          0            0            0            0
## marital.status occupation relationship      race      gender
##          0        2809          0            0            0            0
## capital.gain  capital.loss hours.per.week native.country      salary
##          0          0          0            857            0

```

Esto quiere decir que hay valores desconocidos en los atributos correspondientes al país de origen, a la clase de trabajo que desempeñan y al trabajo concreto asociado al individuo. Como veremos más adelante, estos tres atributos no los vamos a usar, por lo que no debe ser preocupante: nos quedaremos tan solo con aquellos datos cuyo país de origen es Estados Unidos.

2.2. Selección de valores de interés

Como hemos dicho, hay algunos valores que no nos interesan. Como la base de datos es suficientemente grande, nos quedamos con los individuos nacidos en Estados Unidos, que representan a más del noventa porciento de la población. Además, eliminaremos de la base de datos la columna “native.country” una vez que nos hemos quedado solo con los estadounidenses, pues resulta redundante.

Prescindiremos de las ganancias y pérdidas (capital loss y capital gain), debido a que, como hemos visto en el apartado 2.1, la mayoría de los valores de estos atributos son ceros, por lo que no van a resultar útiles para nuestro análisis.

```
# nos quedamos solo con los de estados unidos
data <- datos[datos$native.country == "United-States",]

# quitamos la columna native.country
data <- data[, -14]
```

```
# y las columnas capital.gain  
data <- data[, -11]
```

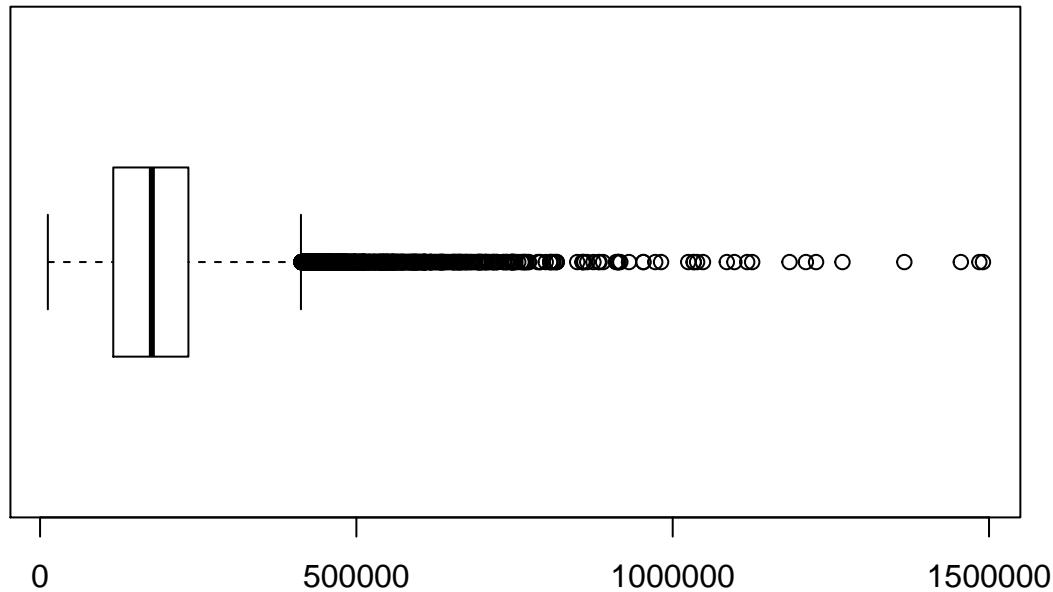
```
# y capital.loss  
data <- data[, -11]
```

Cabe hacer referencia a las variables “occupation” y “workclass”. Podríamos introducirlas en nuestro análisis para la creación de modelos que describan nuestros datos, pero ambas variables, como hemos visto en la descripción del dataset, toman valores muy diferentes y no lo hacen con algún orden establecido (en cambio, la edad o education.num toman valores ordinales). Por este motivo, nuestro modelo se volvería mucho más complejo y opaco si las introdujésemos, por lo que hemos optado por no utilizarlas para el análisis.

2.3. Valores extremos

Tiene también sentido ver qué valores se alejan mucho de la media. Esto lo haremos tan solo para las variables “fnlwgt”, que indica a cuántos individuos representa la información de un individuo concreto, la variable “age” y “hours.per.week”.

```
# realizamos los boxplots  
boxplot(data$fnlwgt, horizontal = TRUE)
```

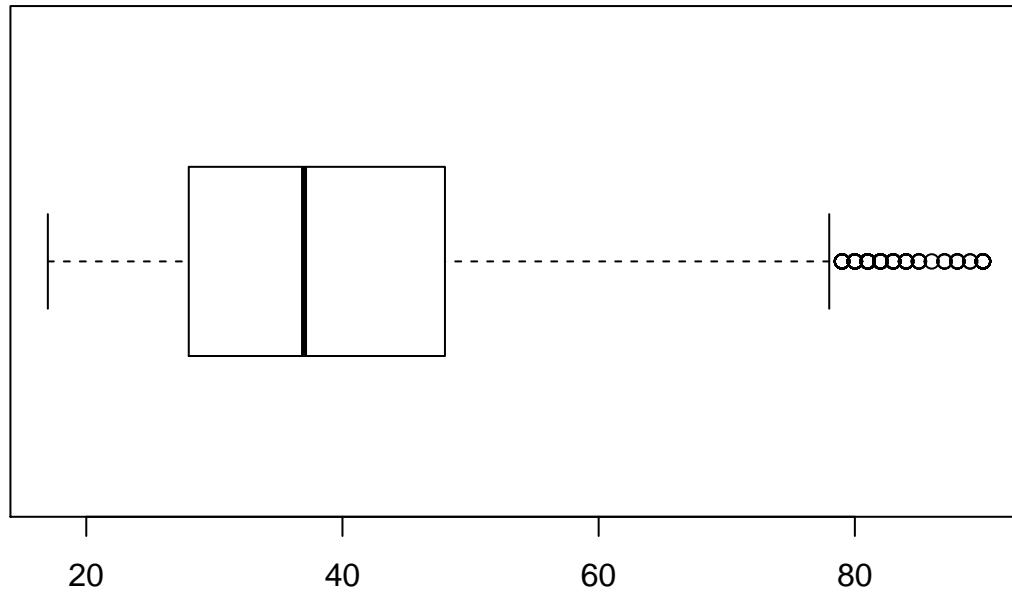


```
# vemos cuántos valores se quedan fuera (valores extremos)  
length(boxplot.stats(data$fnlwgt)$out)
```

```
## [1] 1298
```

Hay 1298 valores que se alejan bastante de la media, pero todos estos valores son razonables, pues la variable indica cuántos ciudadanos representa cada individuo. Vemos ahora el boxplot para la edad:

```
# edad  
boxplot(data$age, horizontal = TRUE)
```

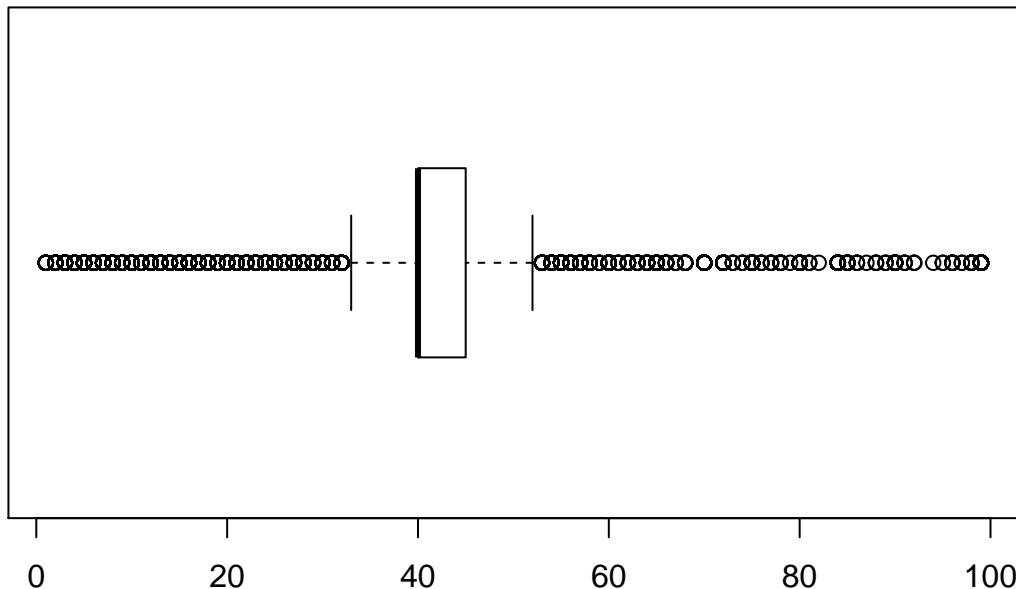


```
# vemos cuántos se han quedado fuera  
length(boxplot.stats(data$age)$out)
```

```
## [1] 192
```

A pesar de que hay 192 datos que se alejan considerablemente de la media, son valores inferiores a 100, y no hay ninguno negativo. Por tanto, teniendo en cuenta la variable que estamos visualizando (edad de los individuos) no debemos realizar ningún cambio en ella. Veamos ahora las horas semanales trabajadas:

```
# horas semanales trabajadas  
boxplot(data$hours.per.week, horizontal = TRUE)
```



```
# veamos cuántos datos se desvían considerablemente de la media
length(boxplot.stats(data$hours.per.week)$out)
```

```
## [1] 12263
```

Este boxplot es el que más valores alejados de la media presenta. Vemos que la media de horas trabajadas a la semana se sitúa en 40, pero tenemos valores que oscilan desde las 0 horas hasta las 100. Es posible que tengamos individuos con casi 100 horas trabajadas a la semana (autónomos o propietarios de negocios que abran todos los días), por lo que no es conveniente modificar estos datos.

2.4. Exportación de los datos procesados

Una vez que hemos realizado el proceso de limpieza de datos sobre los datos iniciales, guardaremos en un nuevo archivo los datos modificados:

```
# exportación de los datos limpios como csv
write.csv(data, "adult_data_clean.csv")
```

3. Análisis de los datos

A continuación, seleccionamos los grupos dentro de nuestra base de datos que pueden ser utilizados para analizar o compararlos entre sí. Quizá no los usemos todos, pero es conveniente realizarlo para tener una visión más general de la base de datos:

```

# grupos de datos:

# sexo femenino y masculino:
data.Male <- data[data$gender == "Male",]
data.Female <- data[data$gender == "Female",]

# diferentes razas:
data.Black <- data[data$race == "Black",]
data.White <- data[data$race == "White",]
data.AIE <- data[data$race == "Amer-Indian-Eskimo",]
data.API <- data[data$race == "Asian-Pac-Islander",]
data.Other <- data[data$race == "Other",]

```

3.1. Revisión de datos normalizados

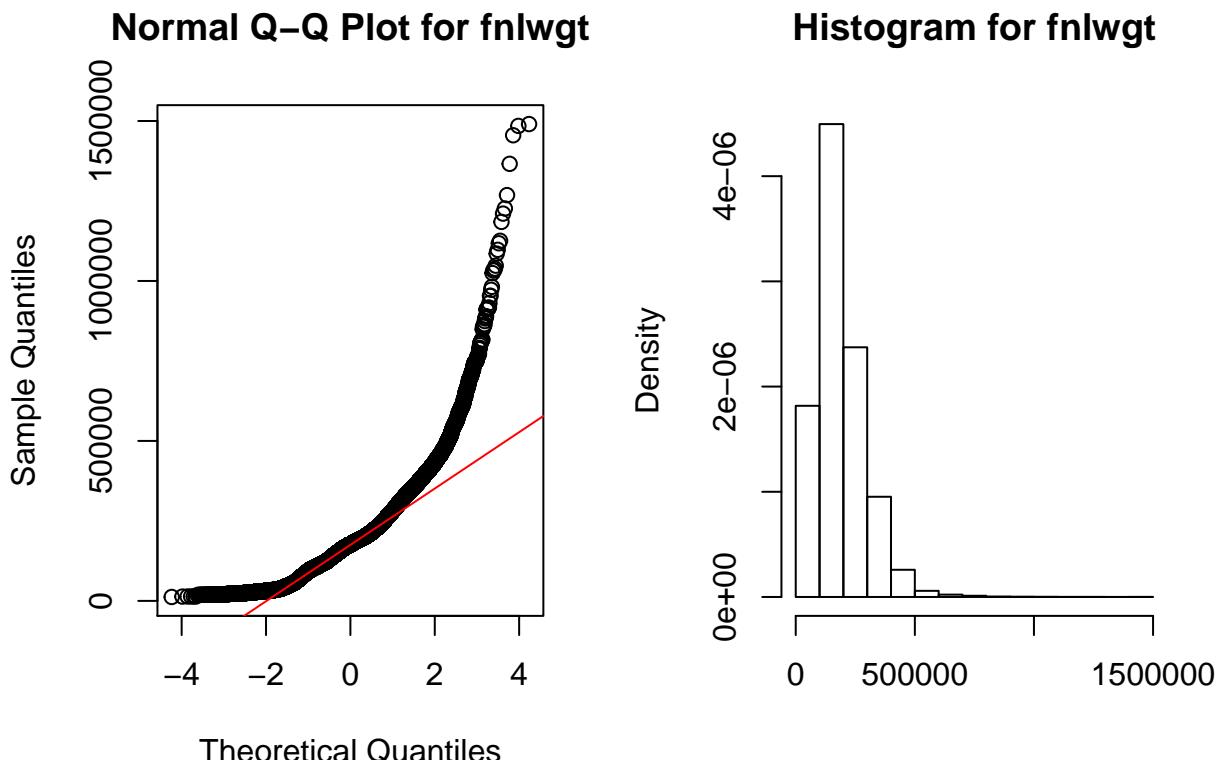
Para revisar si las variables pueden ser candidatas a la normalización miramos las graficas de quantile-quantile plot y el histograma.

```

# veamos primero la variable "fnlwgt"
par(mfrow=c(1,2))
{qqnorm(data[, "fnlwgt"], main = paste("Normal Q-Q Plot for fnlwgt"))
qqline(data[, "fnlwgt"], col="red")}

hist(data[, "fnlwgt"], main=paste("Histogram for fnlwgt"), xlab=colnames(data)[ "fnlwgt"], freq = FALSE)

```



```

ks.test(x=data[, "fnlwgt"], y='pnorm', alternative='two.sided')

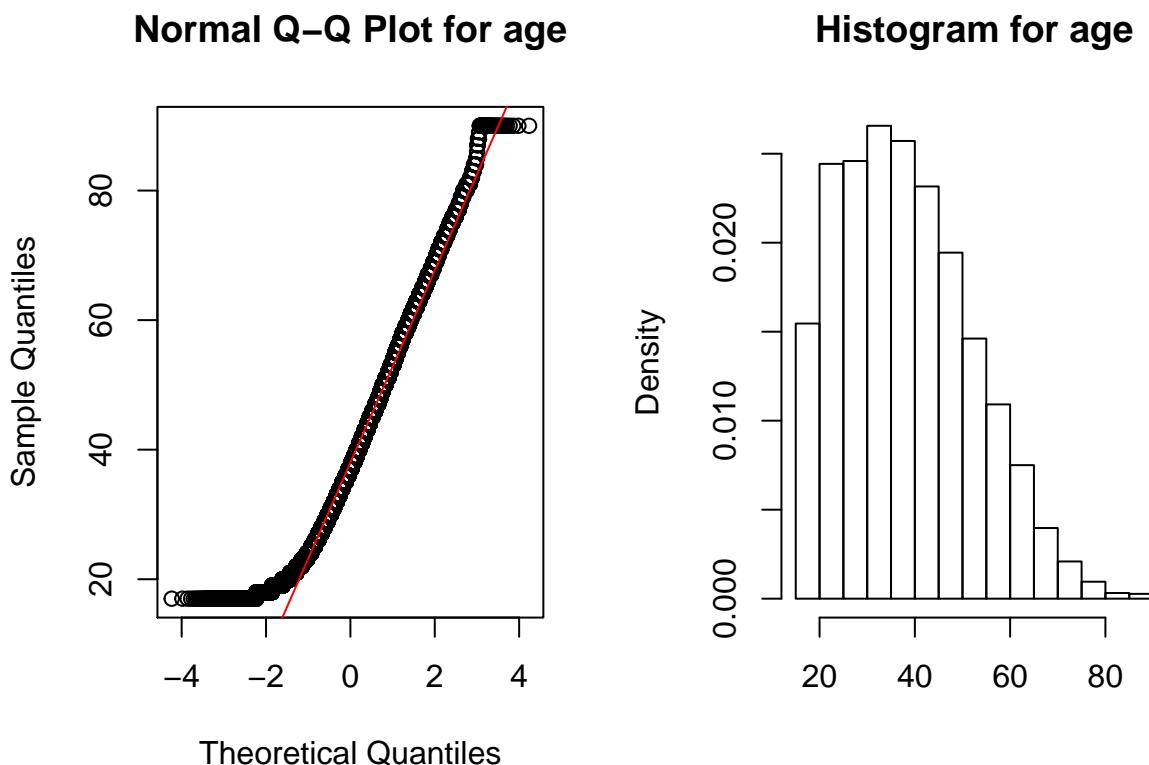
## Warning in ks.test(x = data[, "fnlwgt"], y = "pnorm", alternative =
## "two.sided"): ties should not be present for the Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: data[, "fnlwgt"]
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided

# veamos la variable "age"
par(mfrow=c(1,2))
{qqnorm(data[, "age"], main = paste("Normal Q-Q Plot for age"))
qqline(data[, "age"], col="red")}

hist(data[, "age"], main=paste("Histogram for age"),
xlab=colnames(data)[ "age"], freq = FALSE)

```



```

ks.test(x=data[, "age"], y='pnorm', alternative='two.sided')

## Warning in ks.test(x = data[, "age"], y = "pnorm", alternative = "two.sided"):
## ties should not be present for the Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##

```

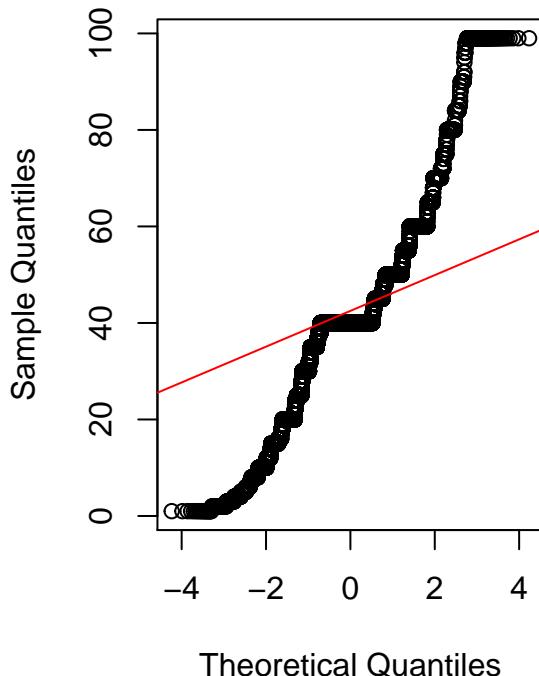
```

## data: data[, "age"]
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
# la variable horas trabajadas por semana
par(mfrow=c(1,2))
{qqnorm(data[,"hours.per.week"],main = paste("Normal Q-Q Plot for hours.per.week"))
qqline(data[,"hours.per.week"],col="red")}

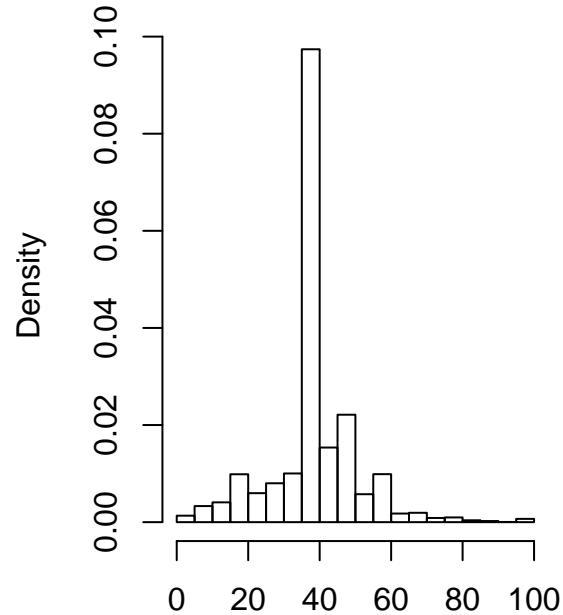
hist(data[,"hours.per.week"], main=paste("Histogram for hours.per.week"),
xlab=colnames(data)[["hours.per.week"]], freq = FALSE)

```

Normal Q-Q Plot for hours.per.we



Histogram for hours.per.week



```

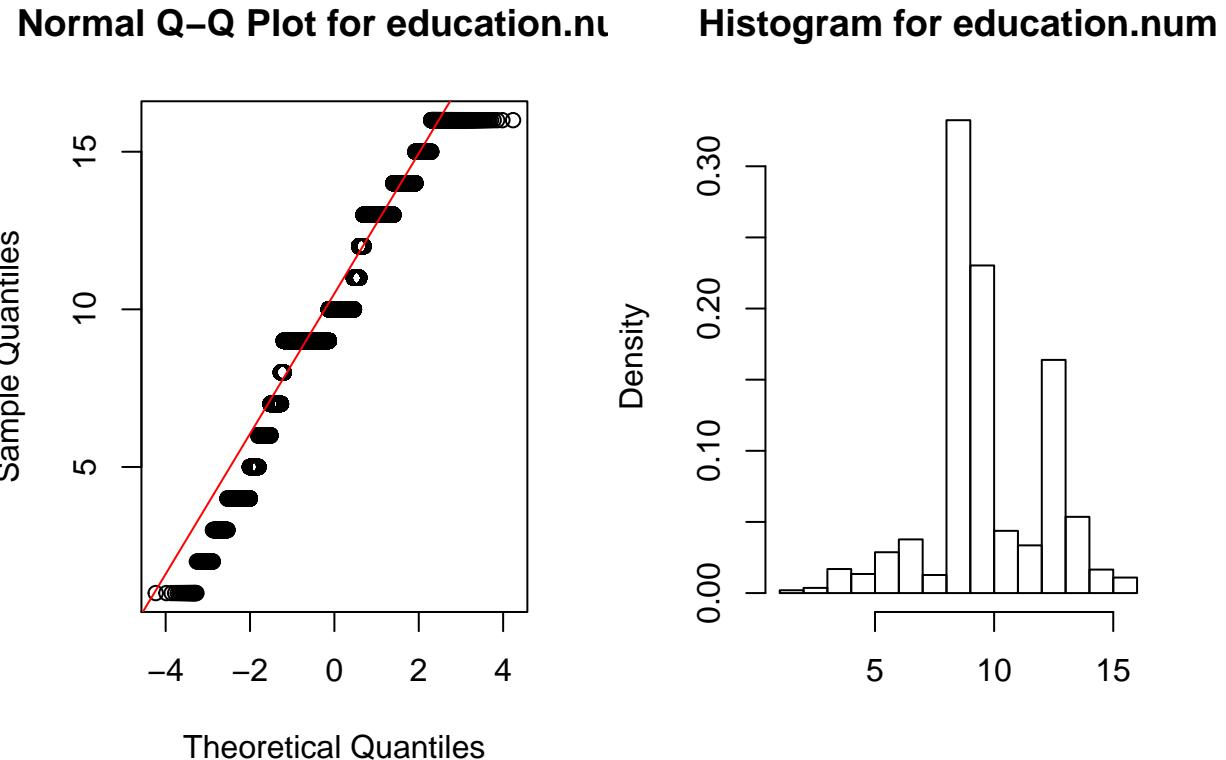
ks.test(x=data[,"hours.per.week"],y='pnorm',alternative='two.sided')

## Warning in ks.test(x = data[, "hours.per.week"], y = "pnorm", alternative =
## "two.sided"): ties should not be present for the Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: data[, "hours.per.week"]
## D = 0.99705, p-value < 2.2e-16
## alternative hypothesis: two-sided

# y la variable education num
par(mfrow=c(1,2))
{qqnorm(data[,"education.num"],main = paste("Normal Q-Q Plot for education.num"))
qqline(data[,"education.num"],col="red")}
```

```
hist(data[, "education.num"], main=paste("Histogram for education.num"),
xlab=colnames(data)[ "education.num"], freq = FALSE)
```



```
ks.test(x=data[, "education.num"], y='pnorm', alternative='two.sided')
```

```
## Warning in ks.test(x = data[, "education.num"], y = "pnorm", alternative =
## "two.sided"): ties should not be present for the Kolmogorov-Smirnov test
##
## One-sample Kolmogorov-Smirnov test
##
## data: data[, "education.num"]
## D = 0.99667, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Los resultados nos indican, a priori, que ninguna de las variables tienen valores numéricos pueden ser candidatas a la normalización, pero como todas las observaciones cuentan con más de 30 elementos, podemos aproximarla como una distribución normal de media 0 y desviación estandar 1 por el teorema del límite central.

Podemos ver ahora si nuestros atributos cumplen la homocedasticidad, para ver si la varianza del error de cada variable es constante para todos sus datos, con el test “fligner”:

```
# en primer lugar es necesario convertir como numérica la variable "salary"
data[ "salary" ] = lapply(data[ "salary" ], as.numeric)

# aplicamos el test a las cuatro variables cualitativas
fligner.test(salary ~ fnlwgt, data = data)
```

```

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: salary by fnlwgt
## Fligner-Killeen:med chi-squared = 22405, df = 26235, p-value = 1
fligner.test(salary ~ age, data = data)

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: salary by age
## Fligner-Killeen:med chi-squared = 4805.9, df = 73, p-value < 2.2e-16
fligner.test(salary ~ education.num, data = data)

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: salary by education.num
## Fligner-Killeen:med chi-squared = 3432.7, df = 15, p-value < 2.2e-16
fligner.test(salary ~ hours.per.week, data = data)

## 
## Fligner-Killeen test of homogeneity of variances
## 
## data: salary by hours.per.week
## Fligner-Killeen:med chi-squared = 3516.4, df = 94, p-value < 2.2e-16

```

Todas las variables menos “fnlwgt” nos dan valores por debajo de $p = 0.05$, por lo que no cumplen la homocedasticidad en ellas. Por tanto, para “fnlwgt” tendremos que realizar la prueba t-test, y para el resto tendremos que recurrir a otras pruebas como la Mann Whitney.

3.2. Análisis de las variables discretizadas

Tenemos dos variables discretizadas: género y raza. Veamos una representación de cada una de estas variables en función del salario para saber si éste es un factor clave. Además, realizando un test chi cuadrado confirmaremos la tendencia descrita en la representación. En caso de que alguna variable condicione el salario, podremos ver el valor del odds ratio, que lo definiremos más adelante.

En primer lugar, representaremos la variable “gender”:

```

# cargamos las librerías necesarias para la representación:
library(ggplot2)

```

```

## Warning: package 'ggplot2' was built under R version 3.6.3
library(cowplot)

## Warning: package 'cowplot' was built under R version 3.6.3
## ****
## Note: As of version 1.0.0, cowplot does not change the
## default ggplot2 theme anymore. To recover the previous

```

```

##   behavior, execute:
##   theme_set(theme_cowplot())

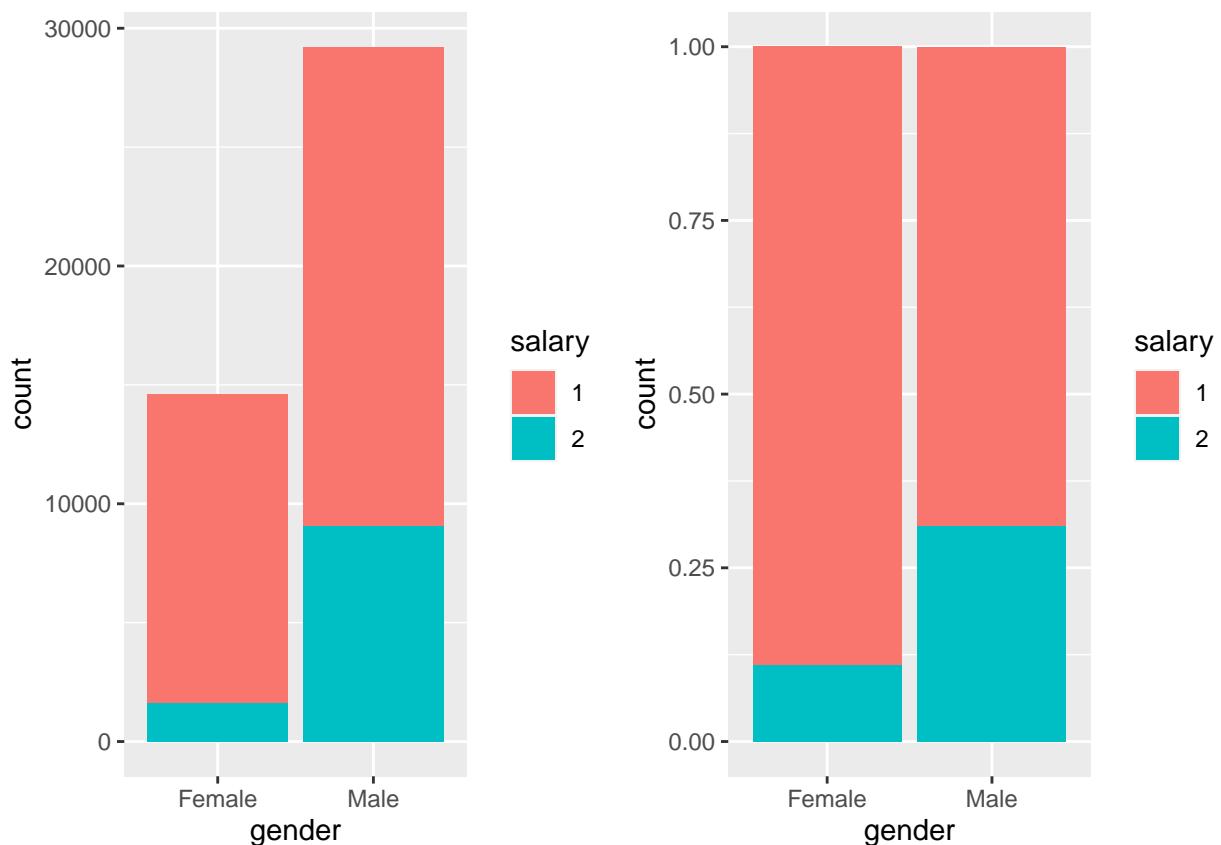
## ****
# cambiamos a factor la columna "salary":
data["salary"] = lapply(data["salary"], as.factor)

# representación de los datos sin normalizar:
gen=ggplot(data, aes(x=gender,fill=salary))+geom_bar()

# representación de los datos normalizados:
gen_n=ggplot(data,aes(x=gender,fill=salary))+geom_bar(position="fill")

# veamos ambas:
plot_grid(gen, gen_n)

```



Como se puede observar, el sexo biológico es un factor bastante influyente para que el salario esté por encima (rojo) o por debajo (azul) de 50k dólares anuales. Veamos el test chi:

```

# creamos la tabla:
stgen <-table(data$salary, data$gender)

# le realizamos el test a la tabla:
chisq.test(stgen, 95)

##
## Pearson's Chi-squared test with Yates' continuity correction

```

```

## 
## data: stgen
## X-squared = 2118.2, df = 1, p-value < 2.2e-16

```

Como el valor de p está por debajo del límite 0.05, rechazamos la hipótesis nula y podemos decir que la variable “gender” es un factor clave para que el sueldo sea alto o bajo para un individuo. Podemos ver el odds ratio:

```

library(questionr)

## Warning: package 'questionr' was built under R version 3.6.3
odds.ratio(stgen)

##          OR  2.5 % 97.5 %      p
## Fisher's test 3.6325 3.4289 3.849 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Este valor, en términos generales, se define como la probabilidad que una condición se presente en un individuo (por ejemplo, que el sueldo sea alto) frente al riesgo que ocurra otra (que el sexo biológico sea femenino o masculino). El valor de referencia para el Odd Ratio es “1”, puesto que valores por debajo del mismo (hasta cero) indican que la asociación entre las variables estudiadas es negativa, y valores por encima de éste (hasta infinito) indican que la asociación es positiva. En nuestro caso, tenemos $OR = 3.6325$, lo cual quiere decir que la probabilidad de que un hombre tenga un sueldo superior a 50k anuales es 3.6325 veces la de que una mujer lo tenga.

Hacemos lo propio con las razas de los individuos (variable race):

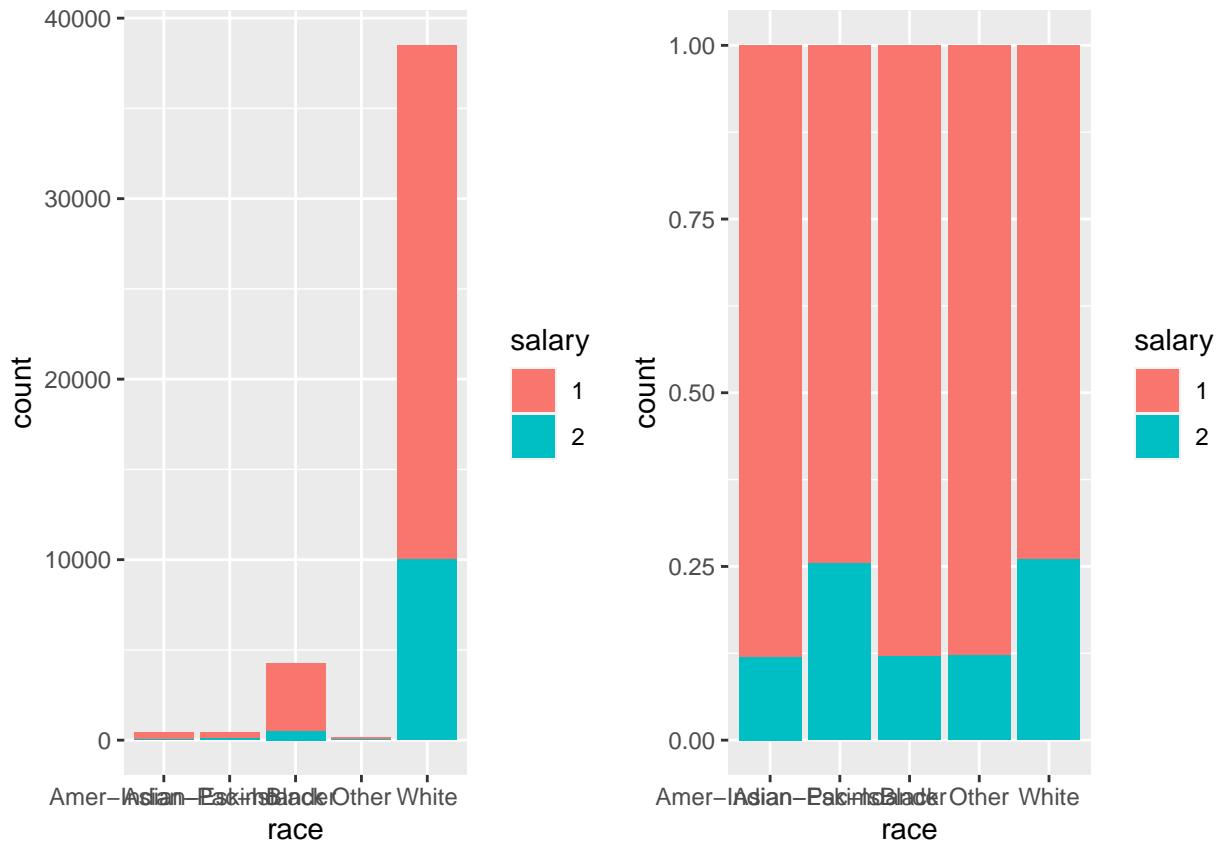
```

# distribución sin normalizar
rac=ggplot(data, aes(x=race,fill=salary))+geom_bar()

# distribución normalizada
rac_n=ggplot(data,aes(x=race,fill=salary))+geom_bar(position="fill")

# vemos ambas representaciones
plot_grid(rac, rac_n)

```



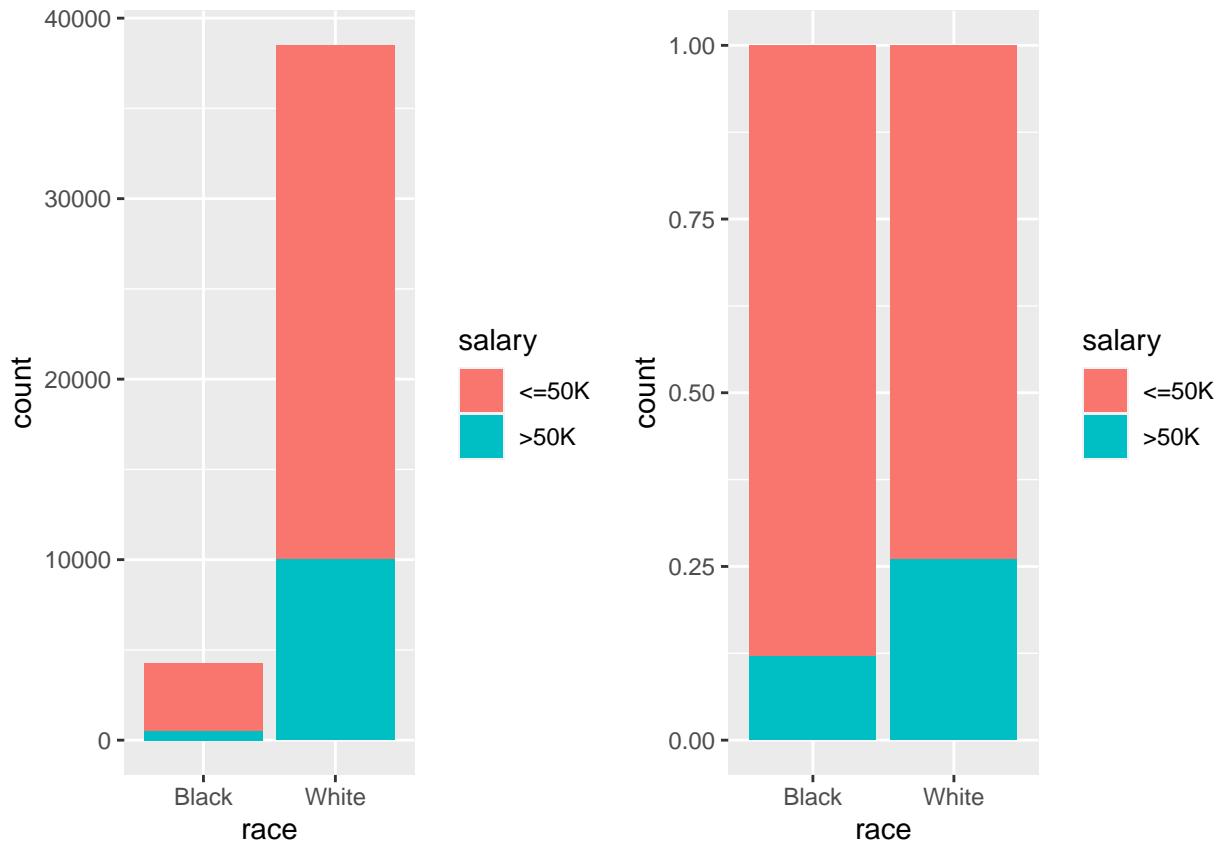
Vemos que la mayoría de datos se corresponden a las razas blanca y negra (lo cual tiene sentido porque nos hemos quedado solo con los individuos de origen estadounidense). Por tanto, no es muy descriptivo mezclar otras razas debido a que el conjunto representado por ellas es mínimo. Veamos el mismo gráfico pero solo para razas negra y blanca:

```
# cogemos solo los datos de razas blanca y negra:
datosraza <- rbind(data.Black, data.White)
datosraza$race <- as.factor(as.character(datosraza$race))
datosraza["salary"] = lapply(datosraza["salary"], as.factor)

# distribución sin normalizar
rac2=ggplot(datosraza, aes(x=race, fill=salary))+geom_bar()

# distribución normalizada
rac2_n=ggplot(datosraza,aes(x=race,fill=salary))+geom_bar(position="fill")

# vemos ambas representaciones
plot_grid(rac2, rac2_n)
```



Podemos hacer el test chi cuadrado y calcular el valor del odds ratio:

```
# tabla de contingencia
strac <-table(datosraza$salary, datosraza$race)
strac

##
##          Black White
##  <=50K    3753 28501
##  >50K      516   9992

# test chi square
chisq.test(strac, 95)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
##  data: strac
##  X-squared = 398.15, df = 1, p-value < 2.2e-16
```

Como el valor de p está por debajo del límite 0.05 rechazamos la hipótesis nula y podemos decir que la variable "race" es un factor clave para que el sueldo sea alto o bajo para un individuo. Podemos ver el odds ratio:

```
# odds ratio
odds.ratio(strac)

##
##          OR  2.5 % 97.5 %
##  Fisher's test 2.5498 2.3182 2.8089 < 2.2e-16 ***
```

```
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nos ha salido un OR = 2.55 aproximadamente. Esto quiere decir que la probabilidad de tener un sueldo por encima de 50k anuales es 2.55 veces mayor en los hombres de raza blanca que negra.

3.3. Contraste de las variables cuantitativas

Dependiendo de si una variable está distribuida normalmente y de si cumple homocedasticidad o no, podemos utilizar varios tests de contraste de hipótesis para ver si dos variables están relacionadas: cuando la normalidad y la homocedasticidad se cumplen (p-valores mayores al nivel de significancia), se podrán aplicar pruebas por contraste de hipótesis de tipo paramétrico, como la prueba t de Student. Esto lo haremos para la variable “fnlwgt”. En los casos en los que no se cumplen, se deberán aplicar pruebas no paramétricas como Wilcoxon (cuando se comparan datos dependientes) o Mann-Whitney (cuando los grupos de datos sean independientes): lo haremos para “age”, “education.num” y “hours.per.week”.

Contraste entre el finalweight medio de los salarios superiores a 50k e inferiores a 50k

```
# separamos la base de datos:
data.salary1.fnlwgt <- data[data$salary == 1,]$fnlwgt
data.salary2.fnlwgt <- data[data$salary == 2,]$fnlwgt
```

Realizamos el test t-student:

```
# test
t.test(data.salary1.fnlwgt, data.salary2.fnlwgt, alternative = "less")

##
## Welch Two Sample t-test
##
## data: data.salary1.fnlwgt and data.salary2.fnlwgt
## t = -0.78355, df = 18626, p-value = 0.2167
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf 990.7233
## sample estimates:
## mean of x mean of y
## 186922.1 187823.3
```

Vemos que las medias son muy parecidas: ambas están por encima de 18000 personas y por debajo de 19000. El valor de $p = 0.2167$ está por encima del límite fijado (0.05), por lo que no podemos rechazar la hipótesis nula: es decir, no podemos decir que las medias de las personas que representa cada individuo es diferente para las personas que tienen un sueldo superior a los 50k anuales de las que tienen un sueldo por debajo de esta cifra.

Contraste entre la edad media de los salarios superiores a 50k e inferiores a 50k

```
# separamos los datos
data.salary1.age <- data[data$salary == 1,]$age
data.salary2.age <- data[data$salary == 2,]$age
```

Realizamos la prueba wilcox.test:

```
wilcox.test(data.salary1.age, data.salary2.age, alternative = "less", mu = 0, paired = FALSE, conf.int = TRUE)

##
## Wilcoxon rank sum test with continuity correction
##
## data: data.salary1.age and data.salary2.age
## W = 112639577, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0
## 95 percent confidence interval:
##      -Inf -8.000021
## sample estimates:
## difference in location
##      -8.999951
```

Interpretando el p-valor obtenido, inferior a 0.05, procedemos al rechazo de la hipótesis nula. Por tanto, podemos concluir que, efectivamente, la edad media para quienes cobran más de 50k anuales es diferente (mayor) de la edad media de aquellos que cobran menos de 50k anuales.

Contraste entre el grado de educación medio de los salarios superiores a 50k e inferiores a 50k

```
# separamos los datos:
data.salary1.education.num <- data[data$salary == 1,]$education.num
data.salary2.education.num <- data[data$salary == 2,]$education.num
```

Realizamos el wilcox.test para las horas trabajadas a la semana:

```
# test
library(stats)
wilcox.test(data.salary1.education.num, data.salary2.education.num, alternative = "less", mu = 0, paired = FALSE)

##
## Wilcoxon rank sum test with continuity correction
##
## data: data.salary1.education.num and data.salary2.education.num
## W = 102623407, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0
## 95 percent confidence interval:
##      -Inf -1.999977
## sample estimates:
## difference in location
##      -1.999964
```

Interpretando el p-valor obtenido, inferior a 0.05, procedemos al rechazo de la hipótesis nula. Por tanto, podemos concluir que, efectivamente, el grado de educación para quienes cobran más de 50k anuales es diferente (mayor) del grado de educación medio de aquellos que cobran menos de 50k anuales.

Contraste entre las horas medias trabajadas a la semana para los salarios superiores a 50k e inferiores a 50k

Hagamos lo propio para las horas trabajadas a la semana:

```
#separamos los datos:
data.salary1.hours.per.week <- data[data$salary == 1,]$hours.per.week
data.salary2.hours.per.week <- data[data$salary == 2,]$hours.per.week
```

Realizamos el wilcox.test para las horas trabajadas a la semana:

```
#test
wilcox.test(data.salary1.hours.per.week, data.salary2.hours.per.week, alternative = "less", mu = 0, pai
## 
## Wilcoxon rank sum test with continuity correction
##
## data: data.salary1.hours.per.week and data.salary2.hours.per.week
## W = 116151090, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0
## 95 percent confidence interval:
##       -Inf -4.999925
## sample estimates:
## difference in location
##                  -4.999933
```

Interpretando el p-valor obtenido, inferior a 0.05, procedemos al rechazo de la hipótesis nula. Por tanto, podemos concluir que, efectivamente, las horas medias trabajadas para quienes cobran más de 50k anuales es diferente (mayor) de las horas medias trabajadas de aquellos que cobran menos de 50k anuales.

Por último, podemos realizar análisis cruzados de otras variables, sin ligarlas necesariamente con la variable “salary”, si esto nos interesa. Podemos ver que las variables “age”, “hours.per.week” y “fnlwgt” no están prácticamente relacionadas entre sí, debido a los valores de los coeficientes de correlación que se muestran a continuación, lo cual nos aporta más información sobre la base de datos.

```
# correlación entre las variables cuantitativas
cor(data$age, data$hours.per.week, method = "spearman")
```

```
## [1] 0.1483822
```

```
cor(data$age, data$fnlwgt, method = "spearman")
```

```
## [1] -0.07369951
```

```
cor(data$fnlwgt, data$hours.per.week, method = "spearman")
```

```
## [1] -0.01514969
```

3.4. Modelos de regresión logística y curva ROC

Podemos estimar modelos de regresión logística tomando como variable dependiente el hecho de ganar más o menos de 50k anuales. En base a estos modelos, también podremos predecir qué factores o atributos son más decisivos para el valor de esta variable. Separaremos el conjunto de datos en dos: train y test:

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 3.6.3
```

```
# Semilla
```

```
set.seed(5)
```

```
# Dividimos aleatoriamente el dataset en dos grupos del 67% y 33 %
sample = sample.split(data, SplitRatio = .67)
```

```
# Conjunto de entrenamiento
```

```
train = subset(data, sample == TRUE)
```

```

# Conjunto de test
test = subset(data, sample == FALSE)

Así, en primer lugar haremos un modelo tan solo con la variable “gender”, y crearemos otros añadiendo el resto de variables a éste (race, education.num y age).

train["salary"] = lapply(train[["salary"]], as.factor)

# primer modelo
mlog1.glm <- glm(salary ~ gender, train, family = "binomial")

# segundo modelo
mlog2.glm <- glm(salary ~ gender + race, train, family = "binomial")

# tercer modelo
mlog3.glm <- glm(salary ~ gender + education.num + race, train, family = "binomial")

# cuarto modelo
mlog4.glm <- glm(salary ~ age + education.num + race + gender, train, family = "binomial")

```

Podemos extraer el coeficiente AIC de los modelos. Este coeficiente es una medida de la calidad relativa de un modelo estadístico para un conjunto dado de datos, y cuanto menor sea, más calidad tendrá nuestro modelo:

```

# coeficientes de cada modelo
extractAIC(mlog1.glm) [2]

```

```
## [1] 30939.29
```

```
extractAIC(mlog2.glm) [2]
```

```
## [1] 30737.92
```

```
extractAIC(mlog3.glm) [2]
```

```
## [1] 27404.52
```

```
extractAIC(mlog4.glm) [2]
```

```
## [1] 26141.37
```

El coeficiente más bajo se corresponde con el cuarto modelo, por lo que nos quedaremos con éste como el que mejor predice la variable “salary”.

```

# Seleccionamos las variables del modelo
test_model= test[, c(1, 5, 9, 10)]

# Guardamos la variable categórica
test_class= test[,12]

# redecimos el resultado
pred=predict(mlog4.glm, test_model, type="response", se.fit = FALSE)

# creamos una nueva variable
pred2=pred

# Clasificamos como "2" las variables con una probabilidad mayor de 0.5
pred2[pred > 0.5]="2"

```

```

# Clasificamos como "1" las variables con una probabilidad menor de 0.5
pred2[pred <= 0.5] = "1"

#Convertimos a factor nuestras predicciones
pred2 = as.factor(pred2)

#Calculamos la tabla de contingencia para las predicciones
prediction_table = table(pred2, test_class)
prediction_table

##      test_class
## pred2     1     2
##       1 10348 2323
##       2   688 1252

```

La tabla nos indica que han sido bien clasificados un total de 11600 casos, y mal clasificados 3011. Veamos la precisión del modelo a partir de la matriz de confusión:

```

# cargamos las librerías
library(caret)

## Warning: package 'caret' was built under R version 3.6.3
## Loading required package: lattice
# Mostramos la precisión
confusionMatrix(prediction_table)$overall[1]

## Accuracy
## 0.7939224

```

Obtenemos una precisión del 79.4 % para nuestro modelo logístico, por lo que podemos afirmar que la clasificación es correcta, aunque con limitaciones.

Una de medir la efectividad de nuestro modelo es por medio de la curva ROC: Lo que haremos por medio de ella es ver una representación gráfica de la sensibilidad frente a la especificidad de nuestro modelo. Así, cuanta mayor sea el área bajo nuestra curva ROC (desde 0 hasta 1), mejor será el ajuste del modelo.

```

# cargamos la librería
library(pROC)

## Warning: package 'pROC' was built under R version 3.6.3
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
## 
## cov, smooth, var
# creamos la curva y vemos el área bajo la curva
resRoc <- roc(test_class ~ pred)

## Setting levels: control = 1, case = 2
## Setting direction: controls < cases
resRoc

##

```

```

## Call:
## roc.formula(formula = test_class ~ pred)
##
## Data: pred in 11036 controls (test_class 1) < 3575 cases (test_class 2).
## Area under the curve: 0.8072

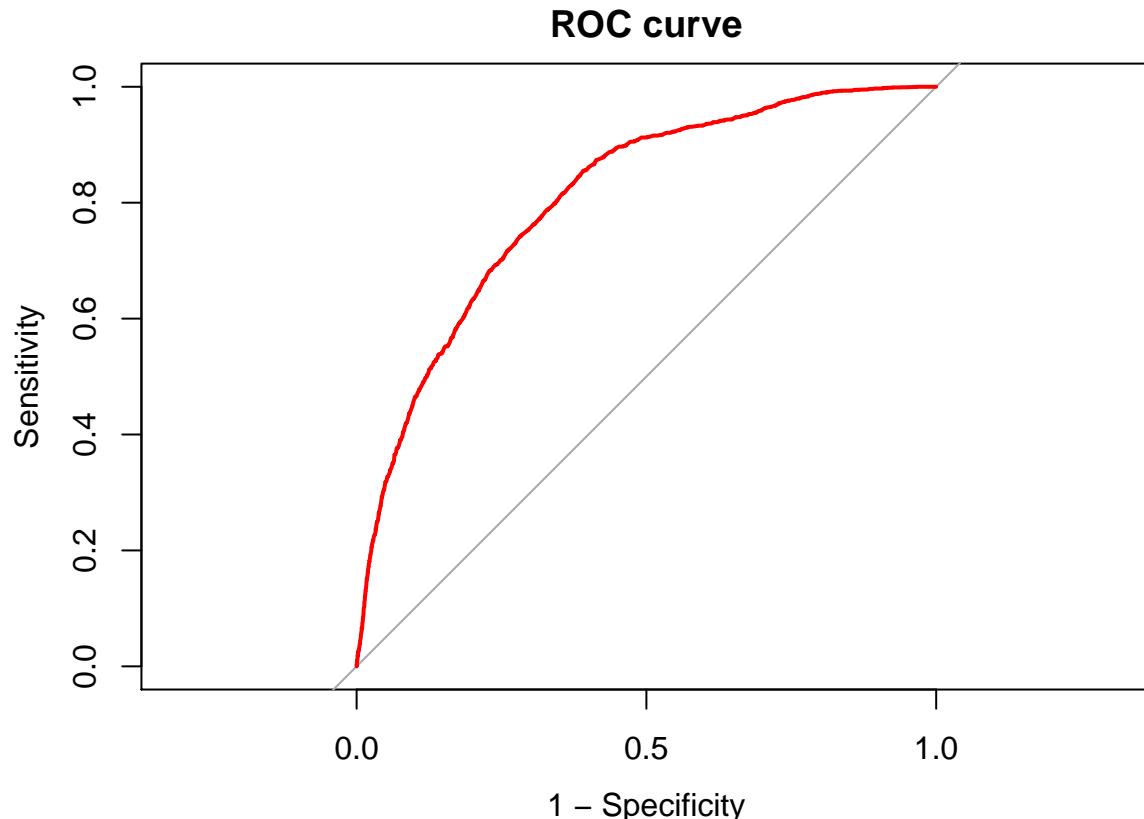
```

El área bajo la curva es 0.8072, lo cual quiere decir que el modelo ajusta notablemente la realidad. A pesar de ello, habrá muchos datos que no describa a la perfección. Veamos la representación de la curva:

```

# representamos la curva
plot(resRoc, legacy.axes = TRUE, col='red', main= 'ROC curve')

```



Por último, podemos realizar una predicción para nuestro modelo. Veamos la probabilidad de ganar más de 50k en dos casos: en el primero, si el individuo es varón, tiene 40 años, su educación es alta y su raza blanca:

```

# predicción
data2 = data.frame(gender="Male", age = 40, education.num = 12, race = "White")
predict(mlog4.glm, data2, type="response")

```

```

##           1
## 0.4335624

```

Nos sale que la probabilidad de ganar más de 50k es considerable (más de un cuarenta por ciento). Veamos ahora lo que sucede si nuestro individuo es mujer, tiene 70 años, su educación baja y su raza negra:

```

data3 = data.frame(gender="Female", age = 70, education.num = 2, race = "Black")
predict(mlog4.glm, data3, type="response")

```

```

##           1
## 0.01120863

```

Tenemos que la probabilidad de ganar más de 50k anuales está sobre el 1 %. Estos dos resultados son acorde con las gráficas expuestas anteriormente.

3.5. Conclusión

En primer lugar, tras haber descrito los datos y visto los principales objetivos de esta práctica, se ha procedido a la limpieza de datos: hemos observado que no hay valores vacíos, y que los signos de interrogación que salían y los valores nulos no era necesario modificarlos; bien porque eran valores correctos, bien porque las variables a las que correspondían iban a ser eliminadas o no iban a ser usadas. Como última parte de la limpieza de datos, hemos eliminado columnas que no íbamos a utilizar, nos hemos quedado solo con los individuos cuyo país de origen es Estados Unidos, y hemos observado que todos los valores extremos eran razonables, por lo que no debíamos eliminar datos.

La parte fundamental de la práctica la ocupa el análisis de datos:

- Primero hemos revisado los datos normalizados, donde los resultados nos indican que ninguna de las variables que tienen valores numéricos pueden ser candidatas a la normalización. Además, con el test Fligner hemos certificado que todas las variables menos “fnlwgt” nos dan valores por debajo de $p=0.05$, por lo que no cumplen la homocedasticidad en ellas.
- En segundo lugar, hemos realizado un análisis de las variables discretizadas “gender” y “race”. Tanto visualmente como por medio del valor p del test *chi square*, podemos afirmar que ambas influyen en el hecho de tener un salario por encima de 50k. Esto lo hemos certificado con los valores del odds ratio, con las siguientes conclusiones:

La probabilidad de que un hombre tenga un sueldo superior a 50k anuales es 3.6325 veces la de que una mujer lo tenga.

La probabilidad de tener un sueldo por encima de 50k anuales es 2.5498 veces mayor en los hombres de raza blanca que negra.

- En tercer lugar, hemos realizado para las variables cuantitativas la prueba t-student (*t.test*) o la prueba wilcox-test, para comprobar la igualdad o no de sus medias para salarios por encima o por debajo de 50k. Para la variable “fnlwgt” no encontramos diferencias significativas entre estas medias: obtenemos $p = 0.2167$, que está por encima del límite fijado (0.05), por lo que no podemos decir que las medias de las personas que representa cada individuo es diferente para las personas que tienen un sueldo superior a los 50k anuales de las que tienen un sueldo por debajo de esta cifra. Para la media de las edades, podemos concluir que la edad media para quienes cobran más de 50k anuales es diferente (mayor) de la edad media de aquellos que cobran menos de 50k anuales. También encontramos diferencias significativas en la variable “hours.per.week”: la media de las horas trabajadas para las personas que ganan menos de 50k es menor que la media de aquellos que ganan más, al igual que para la variable “education.num” (el grado de educación para quienes cobran más de 50k es mayor que el grado medio para los que cobran menos). Estas diferencias la certifican los valores de p de la prueba.
- En cuarto y último lugar, hemos realizado modelos de regresión logística y representado la curva ROC, con los siguientes resultados: el modelo que mejor ajusta los datos es aquél que tiene en cuenta las cuatro variables: “age”, “education.num”, “race” y “gender”. Además, el área bajo la curva ROC para este modelo es 0.8072. También hemos realizado las siguientes predicciones:

La probabilidad de ganar más de 50k si el individuo es varón, tiene 40 años, su nivel de educación es alto y su raza blanca es 0.4336.

La probabilidad de ganar más de 50k si el individuo es mujer, tiene 70 años, su nivel de educación es bajo y su raza negra es 0.0112.

- En general, queríamos ver qué variables son más importantes para que un individuo gane más o menos de 50000 dólares anuales, y esto lo hemos conseguido responder con nuestro modelo de forma satisfactoria. Respecto a su aplicación, podremos utilizarlo, por ejemplo, en empresas que quieran

enfocar su publicidad a personas que tengan un sueldo elevado o bajo, como por ejemplo entidades bancarias.