# Práctica 2

Tipología y ciclo de vida de los datos. Aula 2

Adrián Valls Carbó y Javier Herrero Martín

2023-01-13

# Índice



# 1. Detalles de la actividad

## 1.1. Descripción

En esta actividad se elabora un caso práctico, consistente en el tratamiento de un conjunto de datos (en inglés, dataset), orientado a aprender a identificar los datos relevantes para un proyecto analítico y usar las herramientas de integración, limpieza, validación y análisis de las mismas.

## 1.2. Objetivos

Los objetivos concretos de esta práctica son:

- Aprender a aplicar los conocimientos adquiridos y su capacidad de resoluciónde problemas en entornos nuevos o poco conocidos dentro de contextos más amplios o multidisciplinares

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

## 1.3. Competencias

En esta práctica se desarrollan las siguientes competencias del 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.

## 1.4. Descripción del Dataset

Este conjunto de datos contiene información de una muestra extraída a partir de un censo estadounidense, en el que para cada persona (sin datos personales), se registran los salarios aparte de información personal adicional. Los datos han sido obtenidos en el sitio web de Kaggle. Los datos proceden de la publicación de Kohavi (1996), que fueron obtenidas desde la oficina del censo de EEUU (US Census Bureau) en el año 1996. El conjunto de datos contiene 32.560 registros y 15 variables y se encuentra en formato `.csv`, bajo el nombre `adult.csv`

Las variables de esta muestra son:

- `age`: Edad del individuo. Variable continua expresada en años
- `workclass`: Categorización del individuo en base al perfil laboral. Presenta las categorías: *private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked*
- `fnlwgt`: Peso asignado a cada fila, refleja la proporción de datos que se asimilan dentro de la misma línea (misma información)
- `education`: Nivel de formación educativa del individuo. Contiene las categorías: *Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.*
- `education.num`: Número de años de formación educativa del individuo.
- `marital.status`: Estado civil del individuo. Categorizada en: *Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse*
- `occupation`: Categorización del individuo en base a la tipología de trabajo. Contiene las categorías: *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`: Estado civil del individuo (a diferencia de marital_status, también hace referencia a hijos). Las categorías descritas sonç. *Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried*
- `race`: Grupo racial al que pertenece el individuo. Dentro de ellas se encuentran: *White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black*
- `sex`: Género del individuo: *Female, Male*
- `capital.gain`: Ganancias capitales del individuo €.

- `capital.loss`: Pérdidas capitales del individuo €.
- `native.country`: País de procedencia del individuo, dentro de los que se encuentran los siguientes: *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, Trinadad&Tobago, Peru, Hong, Holand-Netherlands*
- `hours.per.week`: Horas por semana trabajadas por el individuo.
- `income`: Salario (anual) del individuo, en k€, hace referencia a un umbral de salario. Presenta las categorías *>50K, <=50K*

### 1.5. Importancia y objetivos del análisis

La idea original del dataset es analizar y predecir cuáles de dichas variables del censo tienen impacto en la probabilidad de que el individuo gane o no más de 50K de salario anual. Si bien el objetivo de la práctica no es específicamente la predicción de probabilidades, la cantidad de variables nos va a permitir realizar el preprocesado de los datos (tanto dentro de las propias variables como eligiendo qué variables son necesarias para el estudio), así como un análisis de la relevancia de dichas variables.

La importancia de este dataset podría encontrarse en el uso que pudieran hacer desde empresas financieras para conceder créditos a sus clientes en función de saber cuánto llegarán a ganar, así como el hecho de que permite profundizar en las diferencias socieconómicas de diferentes grupos sociales (o al menos, en las existentes en 1996).

## 2. Limpieza de los datos

Leemos el primer lugar el archivo. Para ello tenemos que emplear la función `read.csv` contenida dentro del paquete base de R.

```
# Leemos el archivo
df = read.csv("adult.csv")

# Examinamos los primeros registros
head(df[,1:5])
```

```
##   age workclass fnlwgt    education education.num
## 1  90         ?  77053      HS-grad             9
## 2  82   Private 132870      HS-grad             9
## 3  66         ? 186061 Some-college            10
## 4  54   Private 140359      7th-8th             4
## 5  41   Private 264663 Some-college            10
## 6  34   Private 216864      HS-grad             9
```

Podemos, una vez cargados los datos, examinar cómo R ha leido cada variable y si de forma correcta las ha interpretado.

```
## Llamamos a la funcion str
str(df)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : chr  "?" "Private" "?" "Private" ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013 70037 ...
##  $ education     : chr  "HS-grad" "HS-grad" "Some-college" "7th-8th" ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: chr  "Widowed" "Widowed" "Widowed" "Divorced" ...
```

```
##  $ occupation    : chr  "?" "Exec-managerial" "?" "Machine-op-inspct" ...
##  $ relationship  : chr  "Not-in-family" "Not-in-family" "Unmarried" "Unmarried" ...
##  $ race          : chr  "White" "White" "Black" "White" ...
##  $ sex           : chr  "Female" "Female" "Female" "Female" ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: chr  "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

Vemos en el epígrafe anterior varias cosas. Por un lado podemos ver que los datos perdidos son codificados como '?'. Esto nos conllevará problemas más adelante a la hora de analizar los datos, así que vamos a sustituirlo. En este caso podemos usar R base

```
# Sustituimos los datos
df[df=="?"]<- NA
```

También podemos ver que realmente los datos que son de tipo `chr` deberían serlo del tipo `factor`, por lo que podemos definir una función en la que si la columna es de tipo caracter la transforme en factor

```
# Transformamos todas las columnas que sean caracteres en factor
df[sapply(df, is.character)] <- lapply(df[sapply(df, is.character)],
                                       as.factor)

# Comprobamos que han cambiado
str(df)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : Factor w/ 8 levels "Federal-gov",..: NA 4 NA 4 4 4 4 7 1 4 ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013 70037 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 12 12 16 6 16 12 1 11 12 16 ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 7 7 7 1 6 1 6 5 1 5 ...
##  $ occupation    : Factor w/ 14 levels "Adm-clerical",..: NA 4 NA 7 10 8 1 10 10 3 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 2 2 5 5 4 5 5 3 2 5 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 5 5 5 5 5 5 5 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 2 1 1 2 ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 39 39 39 39 39 NA ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 1 2 ...
```

## 2.1. Examinando los datos perdidos

Tenemos que examinar en nuestro conjunto de datos si disponemos de datos que no estén disponibles (NA o *Not Available*).

```
# Buscamos los datos perdidos
sapply(df, function (x) paste0(sum(is.na(x)),
                               " (", round(sum(is.na(x))/length(x)*100, 2), "%)"))
```

```
##            age      workclass         fnlwgt      education  education.num
##       "0 (0%)" "1836 (5.64%)"       "0 (0%)"       "0 (0%)"       "0 (0%)"
## marital.status     occupation   relationship           race            sex
##       "0 (0%)" "1843 (5.66%)"       "0 (0%)"       "0 (0%)"       "0 (0%)"
```

```
##   capital.gain   capital.loss hours.per.week native.country         income
##       "0 (0%)"       "0 (0%)"       "0 (0%)"  "583 (1.79%)"       "0 (0%)"
```

Vemos que tanto `workclass` como `occupation` tienen 1836 registros perdidos. En el caso de occupation vemos que tiene unos 7 registros perdidos más. Esto supone alrededor de un 6 % de los datos. Por otro lado en `native.country` hay 583 registros perdidos, lo que supone un 1.79 % de los datos perdidos.

Con los datos perdidos podemos realizar varias acciones:

- Etiquetado: Simplemente podríamos asignarles una etiqueta y analizarlos como una categoría más
- Reemplazarlos por una medida de distribución central: podríamos reemplazarlos por la mediana. El problema es que los datos perdidos se agrupan en nuestro caso dentro de variables categóricas, por lo que podríamos sustituirlo en este caso por la moda.
- Imputarlos: es decir, estimar la probabilidad en función a las otras variables de a qué categoría pertence el dato en concreto.
- Omitirlos: es decir, eliminar aquellos registros que contengan datos perdidos o eliminar las columnas que contengan dichos registros.

De cara a imputarlos o no habría que determinar cuál es el mecanismo de generación de los datos perdidos:

- Perdidos completamente aleatorios (MCAR por sus siglas en inglés): esto es que la probabilidad de que los datos estén perdidos es igual para todos los casos. Esto sería que entre todas las categorías la probabilidad de encontrar un dato perdido es constante
- Perdidos aleatorios (MAR por sus siglas en inglés): esto es que la probabilidad de encontrarse perdidos es constante según una categoría observada en los datos. Por ejemplo podría ser que dentro de una categoría concreta los encuestados no quisieran dar su salario, pero tenemos datos de otros de la misma categoría, por lo que podríamos deducir.
- Perdidos no aleatorios (MNAR por sus siglas en inglés): en este caso no sabemos el mecanismo por el que los datos se encuentran perdidos, y este no es debido al azar, por lo que no podemos de hecho deducir las categorias

Si examinamos como se comportan las variables con datos perdidos en función de la variable `income`

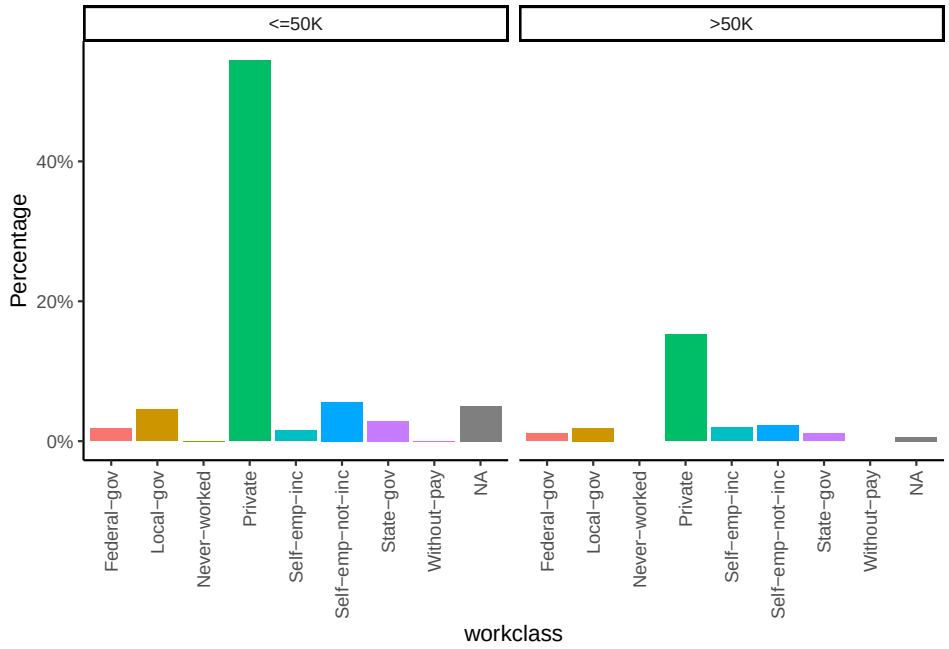


Figura 1: Datos de tipo de trabajo y salario. Se aprecia que los datos perdidos se agrupan más en la categoría de <=50k, por lo que no es totalmente aleatorio. Podríamos decir que los datos perdidos son del tipo MAR, por lo que podemos realizar la imputación de los datos. Esto ocurre también con el resto de las variables, pues es posible que exista un sesgo en el que los encuestados con menor salario tiendan a responder menos a determinados items.

La imputación de datos en este caso no parece estrictamente necesaria, ya que incluso sin tenerlos en cuenta queda una cantidad más que aceptable de datos, pero de cara a la demostración práctica para la actividad que nos ocupa, se va a llevar a cabo. Teniendo en cuenta la distribución de los mismos (MAR), se utilizará el método kNN (k-Nearest Neighbors) que se basa en los k datos vecinos más cercanos para asignar el valor a imputar.

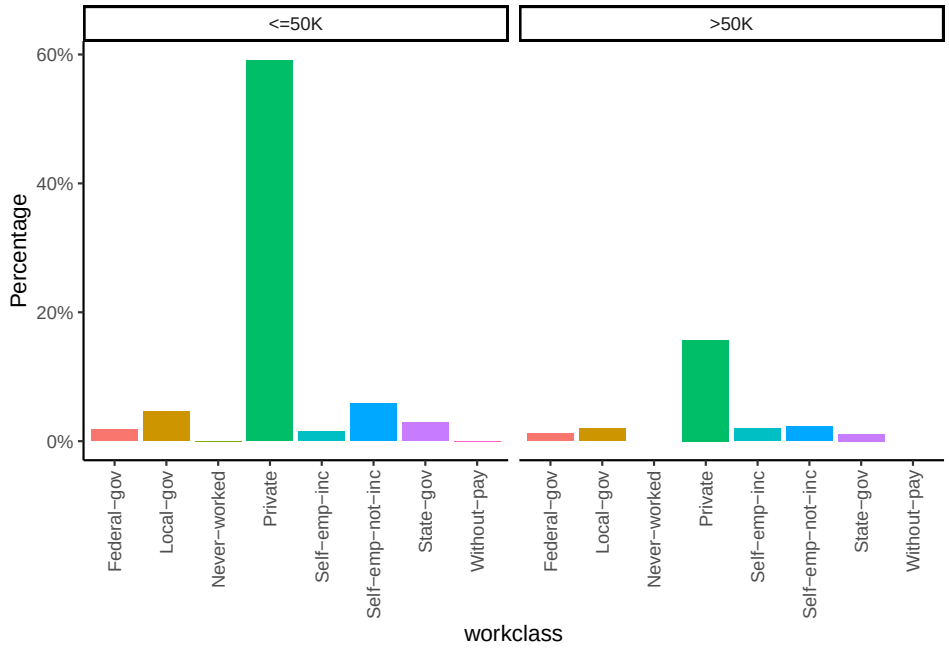


Figura 2: Datos de tipo de trabajo y salario. Se han eliminado los valores NA, siendo sustituidos por valores cercanos a los mismos. A partir de ahora, se trabajará con este nuevo dataframe para preservar el original en caso de que tengamos dudas de si la imputación ha sido correcta

La imputación con la fórmula utilizada puede crear unas variables dummy que indican si el valor de alguna columna para esa fila es real o imputado, pero teniendo el dataframe original no hace falta crear más variables. Sabemos que varias de éstas tienen relación entre sí (`education` y `education.num`, `marital.status` y `relationship`), por lo que vamos a eliminar una de cada pareja para reducir el número de variables.

Por otro lado, `capital.gain` y `capital.loss` hacen referencia a lo mismo, por lo que vamos a combinarlas en una sola variable.

La variable `fnlwgt` hace referencia al peso que tiene cada columna, es decir, un valor asignado por el censo sobre las características demográficas de dicha columna, por lo que no nos aporta información relevante para el estudio (no sirve para hacer una media ponderada or ejemplo)

```
# Eliminamos dos de las variables "repetidas"

df2$education<-NULL
df2$relationship<-NULL
df2$capital.change <- df2$capital.gain-df2$capital.loss
df2$capital.gain<-NULL
df2$capital.loss<-NULL
df2$fnlwgt<-NULL
```

Ahora tenemos 11 variables en nuestro dataframe, 4 menos que al principio, pero una de ellas, `native.country`, tiene muchas categorías dentro, por lo que vamos a reducirlas a 9 categorías más relevantes.

```
# Sustituimos los datos para agrupar categorías

Other = c("South","Outlying-US(Guam-USVI-etc)")

North_america = c("Canada", "United-States")

South_america = c("Columbia", "Cuba", "Dominican-Republic", "Ecuador",
```

```
                  "El-Salvador", "Guatemala", "Haiti", "Honduras",
                  "Jamaica", "Mexico", "Nicaragua", "Peru", "Puerto-Rico",
                  "Trinadad&Tobago")

Europe = c("England", "France", "Germany", "Greece", "Holand-Netherlands",
           "Hungary", "Ireland", "Italy", "Poland", "Portugal", "Scotland",
           "Yugoslavia")

Asia = c("Hong", "India", "Iran", "Japan", "Laos", "Cambodia", "China",
         "Philippines","Taiwan", "Thailand", "Vietnam")


# Reclasificamos los niveles
levels(df2$native.country) <- list(NorthAmerica = North_america,
                                   SouthAmerica =  South_america,
                                   Eur = Europe, Asian = Asia, Ot = Other)


# Vemos los niveles de la variable
levels(df2$native.country)
```

```
## [1] "NorthAmerica" "SouthAmerica" "Eur"          "Asian"        "Ot"
```

```
table(df2$native.country)
```

```
##
## NorthAmerica SouthAmerica          Eur        Asian           Ot
##        29832         1414          522          698           95
```

## 2.2. Detección de valores extremos (outliers)

Podemos examinar dentro de las variables cuantitativas si existen datos que podrían ser considerados outliers. Podemos usar el criterio de considerar aquellas observaciones 2 veces por encima de la desviación standar como outliers.

```
# Seleccionamos las variables numéricas
numericas = names(df2)[sapply(df2, is.numeric)]

# Escalamos los datos numéricos
df_num = data.frame(lapply(df2[,numericas], scale))

# Transformamos el dataframe a la forma larga
df_res = reshape(df_num, varying = list(names(df_num)), times =names(df_num),
                 v.names = "value", timevar = "variable", direction = "long")
```

En la figura 2 podemos ver la representación de los registros que sobrepasan las 2 desviaciones estandar. También podemos examinar cuales son estos datos y si tienen sentido, por ejemplo cogiendo las horas trabajadas y la edad

```
# Con este comando podríamos seleccionar los outliers de la edad
# boxplot.stats(df$age)$out
# Sin embargo son muchos registros, por lo que solo seleccionamos el máximo
max(df2$age)
```

```
## [1] 90
```

```
max(df2$hours.per.week)
```

```
## [1] 99
```

Podemos ver que la edad máxima es 90 que puede tener sentido si estamos hablando de una encuesta, aunque estos encuestados realmente no se encuentran en edad de trabajar. Si consideramos las horas trabajadas por semana vemos que existen algunos registros algo incongruentes pues hay individuos que refieren trabajar hasta 99, lo cual excede el máximo de horas semanales permitidas en España, y si contamos que al menos una persona debe de dormir un mínimo de 6 horas diarias, sería estar trabajando un 78.57 % del tiempo que una persona está despierta en una semana. Sin embargo supondremos que esto es correcto, dado que con las leyes laborales de EEUU puede que sean reales.

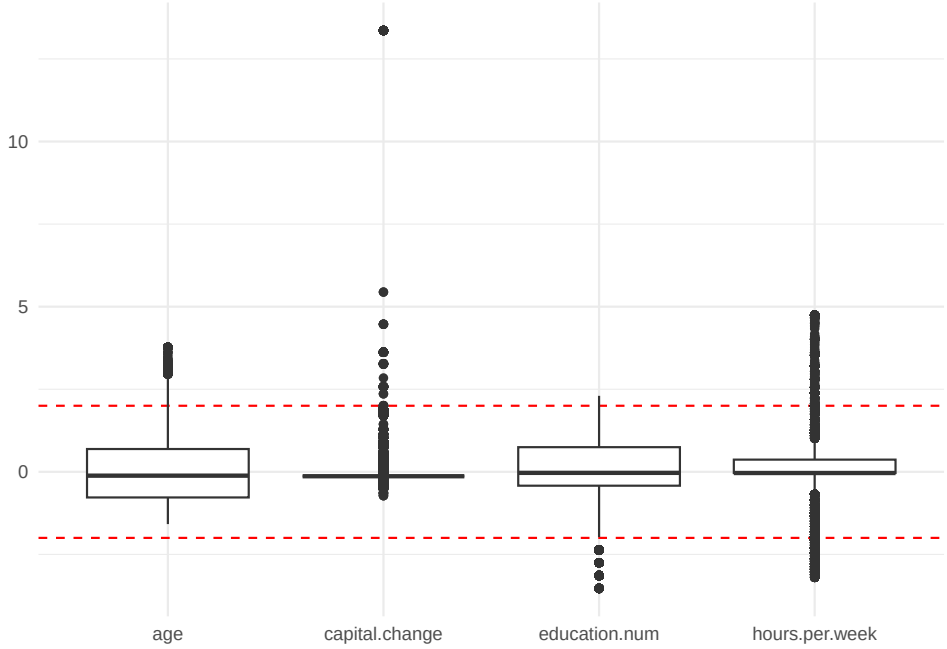


Figura 3: Outliers. Se aprecian los datos escalados para las variables cuantitativas. Las líneas rojas discontinuas representan 2 veces la desviación estandar de la media. Se aprecia que existen muchos registros que podrían ser considerados outliers.

# 3. Análisis de los datos

## 3.1. Valores numéricos

Podemos realizar un breve descriptivo de los valores numéricos presentes en los datos con el comando summary

```
# Llamamos a la función summary

summary(df2[,numericas])
```

```
##       age         education.num   hours.per.week  capital.change
##  Min.   :17.00   Min.   : 1.00   Min.   : 1.00   Min.   :-4356.0
##  1st Qu.:28.00   1st Qu.: 9.00   1st Qu.:40.00   1st Qu.:    0.0
##  Median :37.00   Median :10.00   Median :40.00   Median :    0.0
##  Mean   :38.58   Mean   :10.08   Mean   :40.44   Mean   :  990.4
##  3rd Qu.:48.00   3rd Qu.:12.00   3rd Qu.:45.00   3rd Qu.:    0.0
##  Max.   :90.00   Max.   :16.00   Max.   :99.00   Max.   :99999.0
```

Vemos que la edad media es 38.5816468, presentando valores que van desde 17 a 90. Podemos apreciar en la tabla anterior otros parámetros, lo cual nos da cierta informacion a priori sobre las distribuciones de los datos. Por ejemplo vemos que el valor mínimo, la mediana y el 3er cuartil de `capital.change` se encuentran en el 0. Esto debe de ser porque es una variable muy asintótica. Por el contrario vemos que los datos de `hours.per.week` se encuentran en torno a la cifra de 40, lo que indica que presentará una distribución muy leptocúrtica.

Podemos representar las distribuciones de los datos para ver como se distribuyen

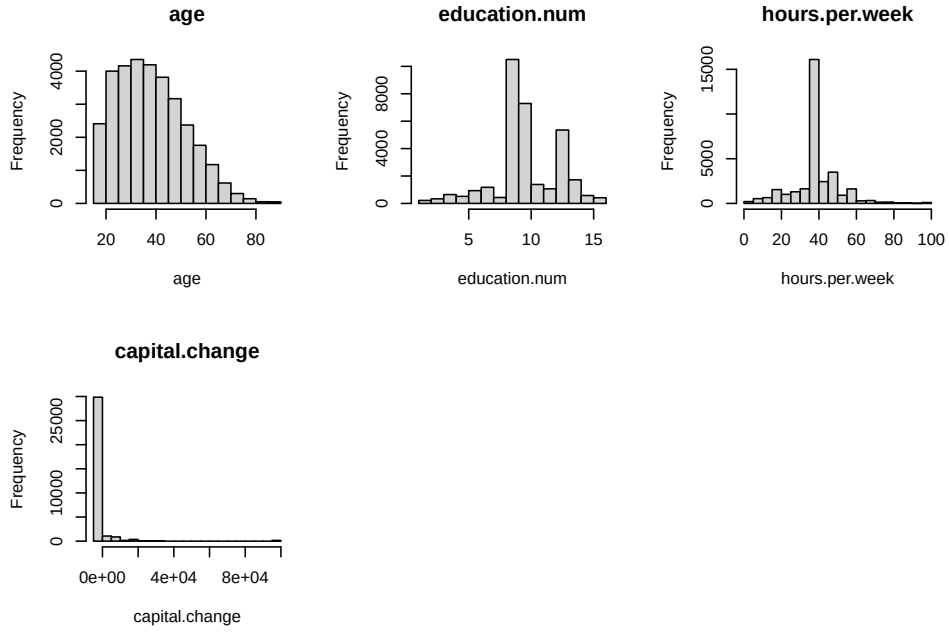


Figura 4: Distribución de los datos numéricos. Se aprecia que los datos de la edad podrían ser normales, aunque se encuentran truncados por debajo de los 18 años. Los datos de educación se encuentran entre 8 y 9 en la mayoría de los casos, así como vemos que en las horas por semana en casi todos los casos se encuentran en 40 horas semanales. La perdida y ganacia de capital se encuentra en 0 en casi todos los casos.

Podemos a continuación aplicar un test de normalidad para cada una de las variables numéricas. Normalmente podríamos emplear el test de Shapiro-Wilk para normalidad, que es un test robusto. Este test se puede llamar en R mediante la función `shapiro.test`. Sin embargo, dado que tenemos más de 5000 observaciones, el test no está implementado en R y debemos acogernos a otras opciones. Entre ellas se encuentra el test de Anderson-Darling. En este test, al igual que en el test de Shapiro-WIlk la hipótesis nula es que los datos presentan una distribución normal. El test de Anderson-Darling se encuentra implementado en el paquete `nortest` mediante la función `ad.test`

```
# Aceptamos como significativo un alpha inferior a 0.05
alpha = .05

testear = function (test, pos){
# Generamos una función para representar la no normalidad
  for (i in 1:length(numericas)) {

    # Cambiamos los strings
    if (pos == ">") {text2 = "SÍ"}
    if (pos == "<") {text2 = "NO"}

    # Seleccionamos el nombre de la variable que deseamos
```

```
    vari = numericas[i]

    # Si en el test especificamos normal, entonces se calculan los test
    # de normalidad de Anderson Darling
    if (test == "normal") {
      texto = "normales"
      p_val = ad.test(df2[,vari])$p.value
    }
    if (test == "homocedasticidad"){
      texto = "homocedásticas"
      p_val = fligner.test(df2[,vari], df2[["income"]])$p.value
    }
    if (i == 1) cat(paste0("Variables que ", text2," son ", texto, ":\n"),
                    "-------------------------------------------------\n")

    if (get(pos)(p_val,alpha)) {
      cat(vari)
    if (i < length(numericas)) cat(", ")
    if (i %% 3 == 0) cat("\n")}
  }
}

# Llamamos a la función que hemos especificado con anterioridad
testear("normal","<")
```

```
## Variables que NO son normales:
##  -------------------------------------------------
## age, education.num, hours.per.week,
## capital.change
```

```
testear("normal",">")
```

```
## Variables que SÍ son normales:
##  -------------------------------------------------
```

Podemos ver que en todas las variables rechazamos la hipótesis de normalidad en todas las variables

Si repetimos el mismo proceso para la homocedasticidad, aplicaremos el test de Fligner, en el que la hipótesis nula es que entre los diferentes grupos las varianzas son constantes. Los grupos en este caso estarán definidos por la variable `income` que es la que deseamos predecir

```
# Llamamos a la función previamente especificada
testear("homocedasticidad","<")
```

```
## Variables que NO son homocedásticas:
##  -------------------------------------------------
## age, education.num, hours.per.week,
## capital.change
```

```
testear("homocedasticidad",">")
```

```
## Variables que SÍ son homocedásticas:
##  -------------------------------------------------
```

Podemos ver por lo tanto que todos las variables numéricas no son normales y no son homocedásticas, por lo que tendremos que usar test no paramétricos para el estudio de las variables

## 3.2. Valores categóricos

Podemos realizar un breve descriptivo de los valores categóricos presentes en los datos con el comando summary

```
# Seleccionamos las variables no numericas
non_num = names(df2)[!names(df2)%in%numericas]

# Llamamos a la función summary de estas variables
summary(df2[,non_num])
```

```
##             workclass                   marital.status           occupation
##  Private         :24355   Divorced             : 4443   Prof-specialty :4293
##  Self-emp-not-inc: 2643   Married-AF-spouse    :   23   Craft-repair   :4288
##  Local-gov       : 2122   Married-civ-spouse   :14976   Exec-managerial:4209
##  State-gov       : 1310   Married-spouse-absent:  418   Adm-clerical   :4057
##  Self-emp-inc    : 1134   Never-married        :10683   Sales          :3926
##  Federal-gov     :  967   Separated            : 1025   Other-service  :3700
##  (Other)         :   30   Widowed              :  993   (Other)        :8088
##                  race           sex             native.country    income
##  Amer-Indian-Eskimo:  311   Female:10771   NorthAmerica:29832   <=50K:24720
##  Asian-Pac-Islander: 1039   Male  :21790   SouthAmerica: 1414   >50K : 7841
##  Black             : 3124                  Eur         :  522
##  Other             :  271                  Asian       :  698
##  White             :27816                  Ot          :   95
##
##
```

En este caso solo podemos ver las frecuencias de los datos, aunque vemos que los datos se encuentran muy disbalanceados. Por eemplo vemos que la mayoría de los casos `workclass` es `Private`, que está centrado en EEUU, hombres y con un `income` inferior a 50K. Respecto a su estado marital vemos que tenemos muchos que nunca se han casado

Podemos realizar a continuación una breve representación gráfica de los datos, tal como se aprecia en la figura 4, comprobando que se cumplen los datos de los que hablamos con anterioridad

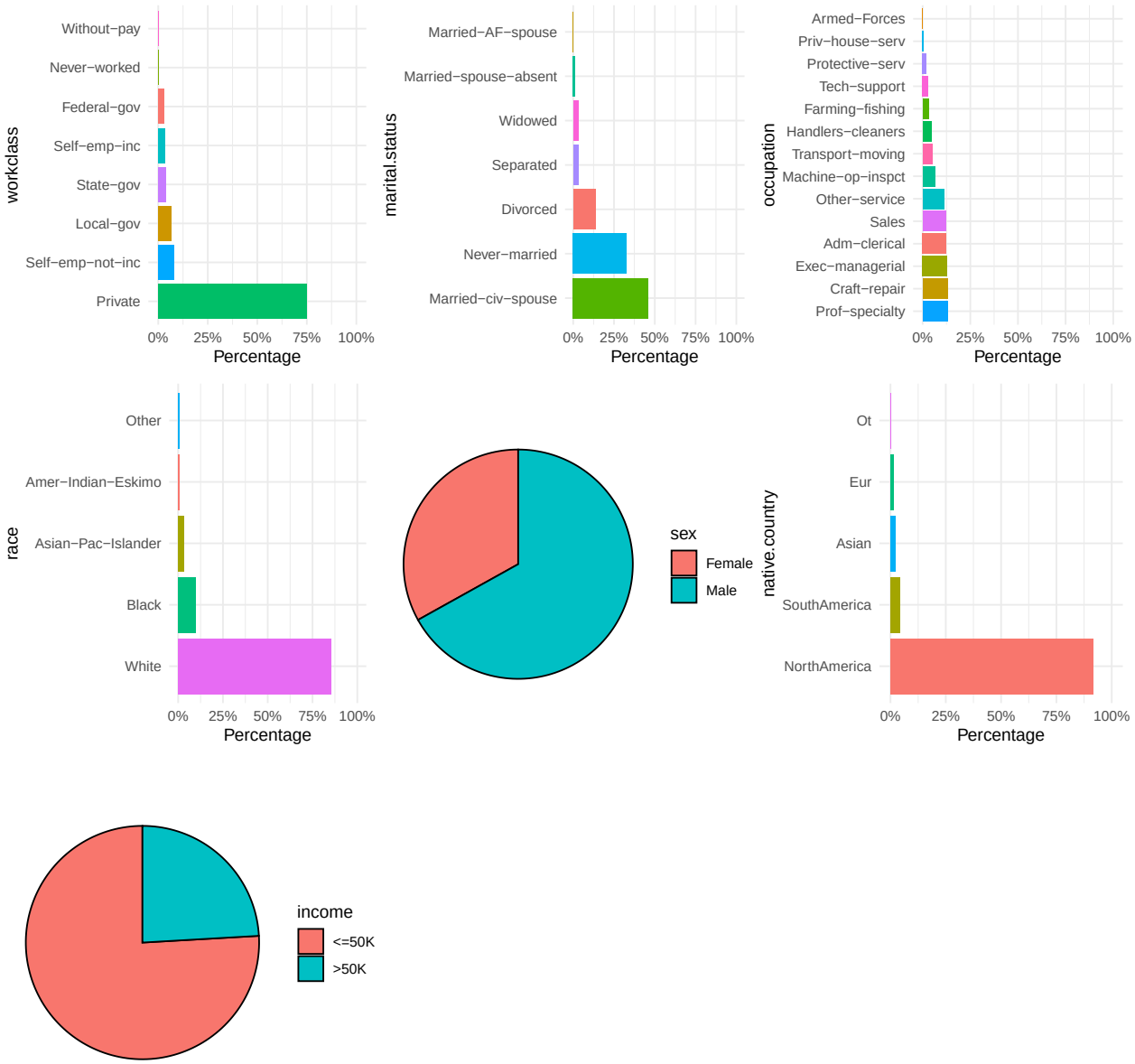


Figura 5: Gráfico de barras de las variables categóricas

# 4. Pruebas estadísticas

Una vez conocidas las variables y sus distribuciones, podemos ver cual es la relación de estas variables con los datos de los ingresos y ver si existe alguna variable que se relacione de forma significativa con los ingresos. Separaremos el análisis por variables cuantitativas y variables cuantitativas.

## 4.1. Variables cuantitativas

Podemos hacer un análisis gráfico para ver si existen diferencias en alguna variable, realizando este análisis gráfico mediante boxplots, tal como se ve en la figura 5.

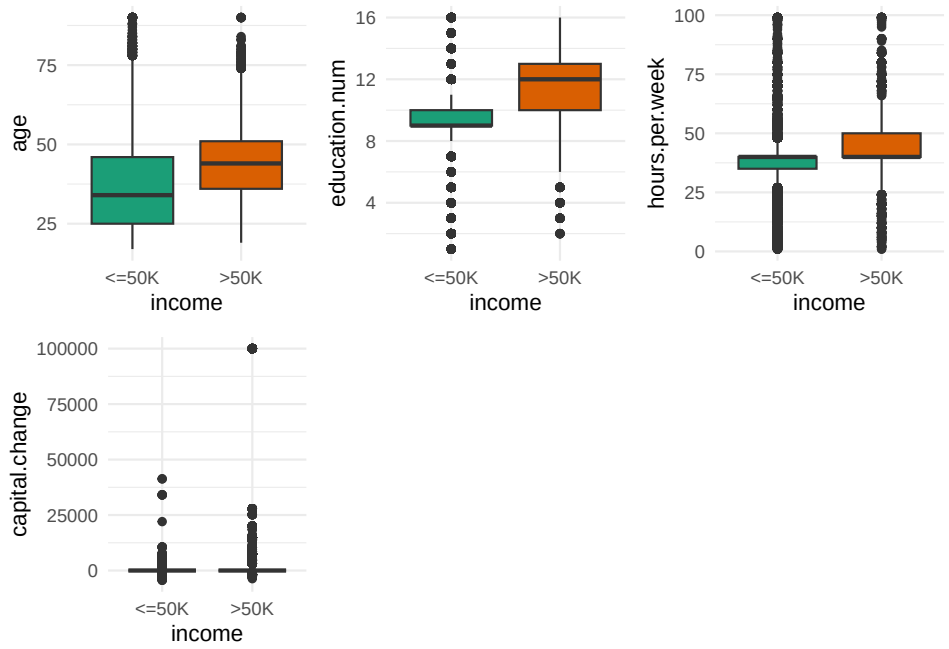


Figura 6: Boxplot de las variables cuantitativas respecto a income. Visualmente parece que la gente más mayor es la que tiene salarios más altos, también los que han estudiado más años y los que trabajan más horas por semana

Sin embargo para el análisis es necesaria la realización de tests estadísticos. En este caso como sabemos que las variables no son normales y además no se cumple la hipótesis de la homocedasticidad, debemos usar test no paramétricos. Para la comparación de dos muestras podríamos usar el test de Wilcoxon. Para representar los datos podemos realizar una tabla en la que mostremos la mediana y rango intercuartílico de cada una de las variables junto al valor de p correspondiente.

```
# Creamos una función para el rango intercuartílico
valor = function (x){
  q = quantile(x, c(0.25, 0.5, 0.75))
  return (paste0(q[2], " [", q[1], ";", q[3], "]"))
}

# Generamos una tabla con el rango y la mediana
tabla = data.frame(t(do.call(cbind, lapply(numericas,
                        function (x) tapply(df2[[x]], df2[["income"]], valor)))))

# Calculamos los valores de p del test de wilcoxon
p = do.call(rbind, lapply(numericas,
       function (x) as.vector(pairwise.wilcox.test(df2[[x]], df2$income)$p.value)))

# Unimos las dos tablas
tabla = cbind(tabla, round(p, 3))
tabla[,3]<-ifelse(tabla[,3]==0, "<0.01", tabla[,3])
rownames(tabla)<-numericas
colnames(tabla)<-c(levels(df2$income), "p valor")
```

Podemos ver que todas las variables estudiadas en este caso muestran diferencias estadísticamente significativas (tabla 1), si bien llama la atención que existan diferencias en variables como `capital.gain` y `capital.loss` en las que presentan una distribución en torno al 0. Probablemente las diferencias en estos grupos se justifiquen

por los outliers, tal como se puede apreciar en los boxplots realizados con anterioridad

Cuadro 1: Tabla de mediana, rango intercuartílico, junto con los valores de p

| | <=50K | >50K | p valor |
|---|---|---|---|
| age | 34 [25;46] | 44 [36;51] | <0.01 |
| education.num | 9 [9;10] | 12 [10;13] | <0.01 |
| hours.per.week | 40 [35;40] | 40 [40;50] | <0.01 |
| capital.change | 0 [0;0] | 0 [0;0] | <0.01 |

## 4.2. Variables cualitativas

Para el estudio de las variables cualitativas podemos realizar de nuevo una representación gráfica, para ver si existen diferencias entre los grupos, para posteriormente realizar los contrastes correspondientes. No mostraremos los porcentajes por cada una de las categorías ya que hay en muchos casos en los que existen variables con muchos grupos.

Esto se puede apreciar claramente en la gráfica 6, donde podemos ver que los hombres blancos con mayor educación y que son autonomos, o que son los maridos, tienen una mayor probabilidad de ganar más de 50 K.

De cara a la realización del análisis estadístico es necesario realizar un test de $\chi^2$ (chi cuadrado), si bien si en alguna de las casillas existe algún valor inferior a 5 se aplicará el test de fisher

```
# Creamos una función que realice el contraste de hipótesis
test_hipo = function (var){
  # Creamos en primer lugar una tabla
  tab = table(df[[var]], df2[["income"]])

  # Si algún valor vale menos de 5 entonces aplicaremos el test de Fisher
  if (any(tab<5)){
    p = fisher.test(tab, simulate.p.value = T)$p.value
  }else{
    # De lo contrario aplicaremos el test de Chi Cuadrado
    p = chisq.test(tab, simulate.p.value = T)$p.value
  }
  # Redondeamos los resultados
  p = round(p, 3)
  p = ifelse(p ==0, "<0.01", p)

  return(p)
}

# Generamos la tabla
resultado = data.frame(do.call(rbind, lapply(non_num[non_num!="income"], test_hipo)))
rownames(resultado)<-non_num[non_num!="income"]
colnames(resultado) = "p valor"
```

Cuadro 2: Tabla de los valores de p para las variables cualitativas. Se aprecia que son todas significativas

| | p valor |
|---|---|
| workclass | <0.01 |
| marital.status | <0.01 |
| occupation | <0.01 |
| race | <0.01 |
| sex | <0.01 |
| native.country | <0.01 |

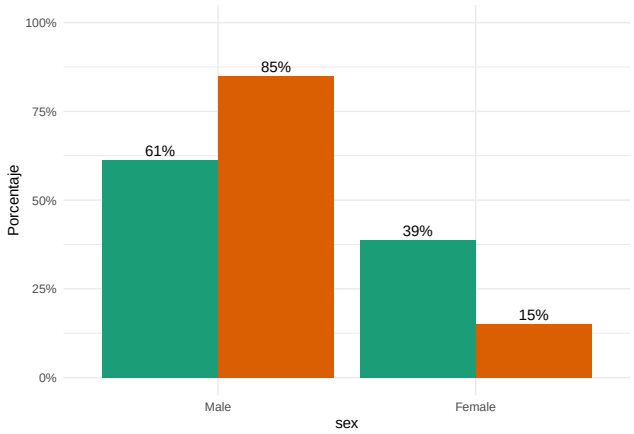


Figura 7: Gráfico de barras del sexo. Se aprecia la brecha salarial debida al género, en la que el 85 % de los que ganan >50K son hombres, mientras que solo el 15 % de las mujeres lo son

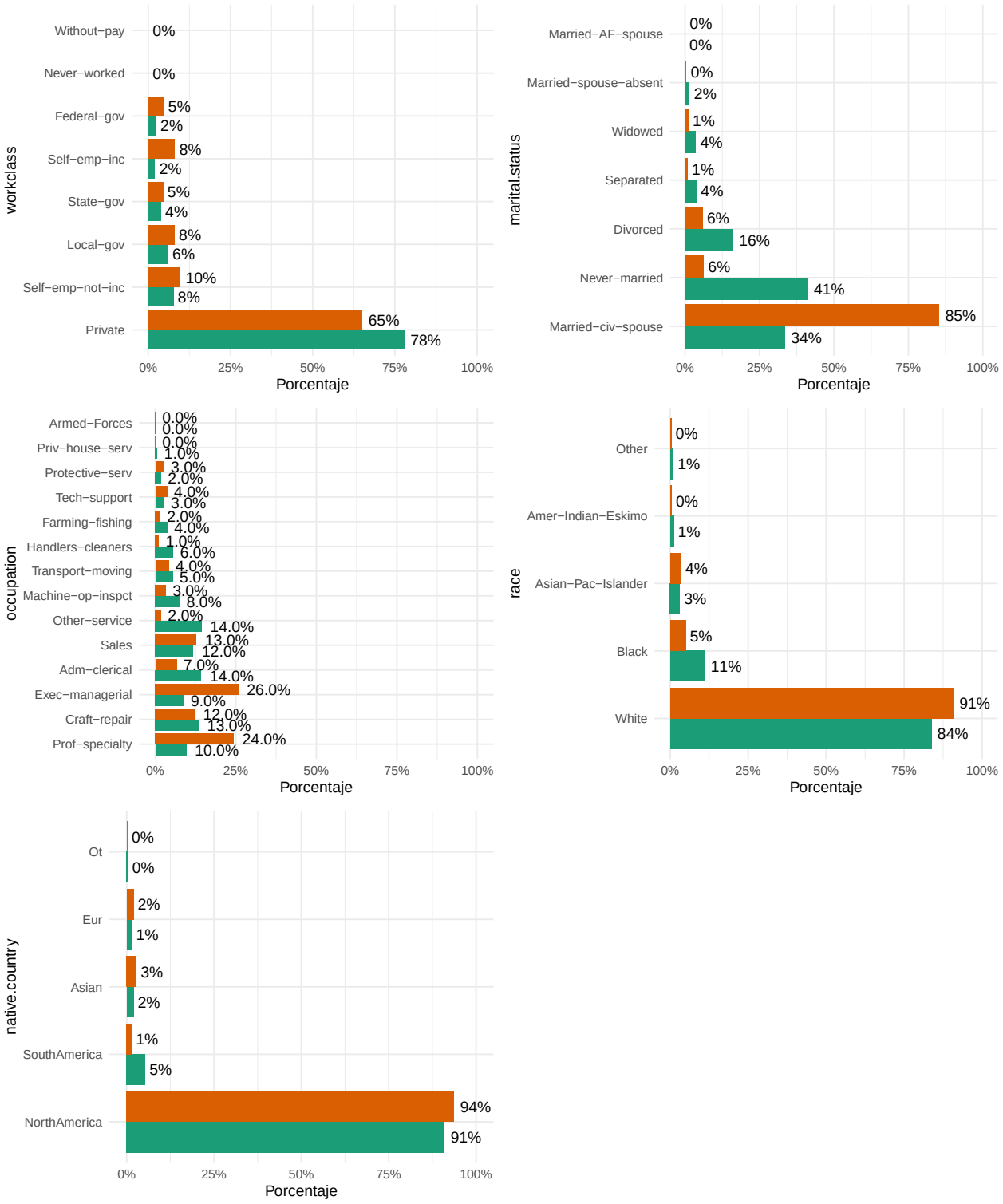


Figura 8: Gráfico de barras de las variables categóricas. En verde se muestran los encuestados que ganan <=50K y en marrón los que ganan más de 50K

## 4.3. Modelo de regresión logística

Podemos intentar ajustar un modelo de cara a predecir con las variables anteriores el sueldo. Un acercamiento simple al problema puede ser el ajuste de una regresión logística binomial para la predicción de la variable `income`.

Se crea el modelo predictivo utilizando todas las variables del set, creando una variable binomial a partir del income y exceptuando income (ya que el programa ve la relación directa que hay con esa variable).

```
# Reordenamos los niveles de los factores
invers = function(y) {-length(y)}
df2[,non_num] = lapply(non_num, function(x) reorder(df2[[x]], df2[[x]], FUN= invers))

# Creamos una variable dicotómica
df2$Less50 <- ifelse(df2$income == "<=50K", 0, 1)

# Obtenemos una muestra del 80% de los datos
sampling<-sort(sample(nrow(df2),nrow(df2)*0.8))

# Seleccionamos un grupo para ajustar el modelo
training_set<-df2[sampling,]

# Seleccionamos otro grupo para validarlo
testing_set<-df2[-sampling,]

# Ajustamos una regresión logística binomial
modeloLR<-glm(Less50~. - income, data = training_set, family = binomial)

# Mostramos en pantalla el modelo
summary(modeloLR)
```

```
##
## Call:
## glm(formula = Less50 ~ . - income, family = binomial, data = training_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4325  -0.5149  -0.2141  -0.0430   3.8215
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -5.271e+00  1.854e-01 -28.431  < 2e-16 ***
## age                                2.545e-02  1.723e-03  14.766  < 2e-16 ***
## workclassSelf-emp-not-inc         -4.436e-01  6.893e-02  -6.436 1.22e-10 ***
## workclassLocal-gov                -1.444e-01  7.711e-02  -1.873 0.061069 .
## workclassState-gov                -2.789e-01  9.797e-02  -2.846 0.004420 **
## workclassSelf-emp-inc              2.662e-01  9.305e-02   2.861 0.004229 **
## workclassFederal-gov               5.615e-01  1.000e-01   5.613 1.99e-08 ***
## workclassNever-worked             -9.482e+00  1.458e+02  -0.065 0.948154
## workclassWithout-pay              -1.206e+01  1.412e+02  -0.085 0.931952
## education.num                      2.814e-01  1.017e-02  27.653  < 2e-16 ***
## marital.statusNever-married       -2.645e+00  6.788e-02 -38.962  < 2e-16 ***
## marital.statusDivorced            -2.165e+00  7.321e-02 -29.570  < 2e-16 ***
## marital.statusSeparated           -2.317e+00  1.690e-01 -13.711  < 2e-16 ***
## marital.statusWidowed             -2.257e+00  1.596e-01 -14.136  < 2e-16 ***
## marital.statusMarried-spouse-absent -2.240e+00  2.327e-01  -9.626  < 2e-16 ***
```

```
## marital.statusMarried-AF-spouse      9.184e-01  5.396e-01   1.702 0.088770 .
## occupationCraft-repair              -5.745e-01  7.727e-02  -7.435 1.05e-13 ***
## occupationExec-managerial            1.582e-01  6.745e-02   2.345 0.019040 *
## occupationAdm-clerical              -5.992e-01  8.372e-02  -7.157 8.27e-13 ***
## occupationSales                     -3.950e-01  7.653e-02  -5.161 2.46e-07 ***
## occupationOther-service             -1.378e+00  1.181e-01 -11.664  < 2e-16 ***
## occupationMachine-op-inspct         -9.307e-01  1.054e-01  -8.829  < 2e-16 ***
## occupationTransport-moving          -6.928e-01  1.003e-01  -6.907 4.96e-12 ***
## occupationHandlers-cleaners         -1.417e+00  1.540e-01  -9.199  < 2e-16 ***
## occupationFarming-fishing           -1.768e+00  1.462e-01 -12.094  < 2e-16 ***
## occupationTech-support               6.764e-02  1.095e-01   0.618 0.536890
## occupationProtective-serv           -9.138e-02  1.292e-01  -0.707 0.479509
## occupationPriv-house-serv           -4.282e+00  1.278e+00  -3.352 0.000803 ***
## occupationArmed-Forces              -1.207e+01  2.274e+02  -0.053 0.957675
## raceBlack                           -1.664e-01  8.015e-02  -2.076 0.037877 *
## raceAsian-Pac-Islander              -9.051e-03  1.653e-01  -0.055 0.956324
## raceAmer-Indian-Eskimo              -7.305e-01  2.512e-01  -2.908 0.003638 **
## raceOther                           -5.379e-01  2.952e-01  -1.822 0.068398 .
## sexFemale                           -1.687e-01  5.706e-02  -2.956 0.003117 **
## hours.per.week                       3.274e-02  1.746e-03  18.752  < 2e-16 ***
## native.countrySouthAmerica          -5.383e-01  1.398e-01  -3.850 0.000118 ***
## native.countryAsian                 -9.095e-02  1.895e-01  -0.480 0.631198
## native.countryEur                    1.826e-01  1.431e-01   1.276 0.201948
## native.countryOt                    -8.296e-01  4.280e-01  -1.938 0.052605 .
## capital.change                       2.451e-04  9.383e-06  26.126  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28816  on 26047  degrees of freedom
## Residual deviance: 17282  on 26008  degrees of freedom
## AIC: 17362
##
## Number of Fisher Scoring iterations: 12
```

En lo que respecta a las variables explicativas, las variables continuas (`age`, `hours.per.week`, `capital.change` y el `education.level`) influyen significativamente en la probabilidad de tener un salario. En cuanto al resto de variables, se puede apreciar la contribución del `sex` (Male > Female), `race` (White> todas menos islander pacific que no es significativo), `work-class` (Private, aunque federal gov es significativamente mayor), `marital_status` (Married mayor que todas) y `ocuppation` (Exec-managerial mayor que todas). Además se aprecia que los norteamericanos tienen sueldos superiores al resto de nacionalidades, excepto que asiaticos y europeos, donde no se aprecian diferencias significativas

Ahora, vamos a comprobar el modelo para predecir el resultado del testing_set, es decir, con datos no utilizados para entrenar el modelo. Luego, factorizamos la probabilidad optenida como 1 o 0, de manera que cuadre con el formato esperado

```
# Predecimos los datos
prediccion_test<-predict(modeloLR, testing_set)

# Si la probabilidad es mayor o igual al 50% entonces valdrá 1
prediccion_test<-ifelse(prediccion_test>=0.5,1,0)
```

Una vez se tiene la predicción, se crea la matriz de confusión (se pasan tanto los datos como la referencia como factor, para evitar problemas de formato)

```
table(as.factor(prediccion_test), as.factor(testing_set$Less50))
```

```
##
##        0    1
##   0 4769  841
##   1  201  702
```

Se puede observar que el modelo hace un buen trabajo a priori, siendo la mayor 0 predichos como les corresponde. Para analizar de manera cuantificable la predicción del modelo, se calculan la sensitividad y sensibilidad. No utilizaremos las funciones por defecto para realizar estos cálculos ya que toman el 0 como positivo y el 1 como negativo.

La sensibilidad es el ratio de los positivos, es decir positivos verdaderos / (positivos verdaderos + falsos negativos), es decir, cuantos casos negativos se asocian erróneamente con un positivo (en este caso, solo 0.8410584 de los casos asignados como negativos por el modelo son en realidad positivos).

La especificidad es el ratio de los negativos, es decir negativos verdaderos / (negativos verdaderos + falsos positivos), es decir, cuantos casos positivos se asocian erróneamente con un negativo (en este caso, solo 0.792517 de los casos asignados como positivos por el modelo son en realidad negativos). En este caso se comprueba que el modelo no hace tan buen trabajo, por lo que seguramente habría que revisar de nuevo qué datos no han sido significativos en el modelo, y realizar alguna transformación para mejorar el resultado obtenido.

# 5. Conclusiones

Se han sometido los datos a un preprocesamiento para manejar los casos de elementos vacíos y valores extremos (outliers). Para el caso delos primeros, tras analizar su distribución (MAR), se ha elegido la imputación de los valores basándonos en los valores similares a los perdidos (o más próximos), de manera que no se eliminen los registros, pero manteniendo la base de datos sin alterar por si hubiese hecho falta una revisión de los valores imputados. Para el caso del segundo, debido al estado de las leyes laborales de EEUU, se ha optado por incluirlos, ya que no tenemos la certeza de que esos extremos sean imposibles, y eliminar los valores sin un respaldo no parece la opción más correcta.

Posteriormente, se ha llevado a cabo un análisis de las variables, reduciendo la dimensionalidad eliminando aquellas que eran redundantes o no aportaban información relevante y agrupando las categorías de las variables no numéricas para reducirlas sin perder información relevante. También se han explorado las variables de valores numéricos, estudiando su normalidad y homocedasticidad, así como la distribución de las variables categóricas mediante gráficas.

Una vez conocidas las variables y sus distribuciones, se ha procedido a estudiar la significancia de ambos tipos de variables con respecto a `income`. Para las cuantitativas, dada la falta de normalidad y homocedasticidad, se ha utilizado el test de Wilcoxon, un test no paramétrico, para la comparación de dos muestras. Para las cualitativas, se ha realizado un test de $\chi^2$ (chi cuadrado), si bien si en alguna de las casillas existe algún valor inferior a 5 se ha aplicado el test de fisher.

Finalmente, se ha utilizado un modelo de regresión logística primero para adaptarlo al set de entrenamiento, y luego se ha comprobado su eficacia prediciendo los resultados esperados con los datos de test, así como se ha comprobado dichos resultados midiendo tanto su sensibilidad como sensitividad. En el caso de estudio, no tiene porque haber ninguna preferencia de cara a mejorar una de estas estadísticas en detrimento de la otra, pero dependiendo del uso que se le vaya a dar al modelo, puede ser interesante. Por ejemplo, si se van a usar para la concesión o no de créditos solamente a las pesonas con un salario anual mayor que 50k, deberíamos ponderar cuanto nos interesa la cantidad de créditos que otorgamos frente a otorgarlos a gente que en realidad no cumpla la condición (en otras utilidades, como en datos utilizados para predecir enfermedades, las consecuencias de un falso negativo y de un falso positivo pueden tener consecuencias muy diferentes).

## 6. Contribuciones

| Contribuciones | Firma |
|---|---|
| Investigación Previa | Adrián Valls, Javier Herrero |
| Redacción de las respuestas | Adrián Valls, Javier Herrero |
| Desarrollo del código | Adrián Valls, Javier Herrero |
| Participación en el video | Adrián Valls, Javier Herrero |

## 7. Bibliografía

Kohavi, Ron. 1996. "Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid." In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 202–7.