PEC 4

Adrián Valls Carbó y Javier Herrero Martín

2023-01-08

Índice

1. Detalles de la actividad

1.1. Descripción

1.2. Objetivos

1.3. Competencias

2. Resolución

2.1. 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*

2.2. 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

2.3. Limpieza y cargado 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 ...
```

2.4. 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`

Para este caso concreto analizaremos los datos perdidos como una categoría más de los datos, sin llegar a eliminarla. Si en el futuro para el desarrollo de un modelo necesitamos que no existan valores perdidos lo imputaremos

2.5. 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(df)[sapply(df, is.numeric)]
```

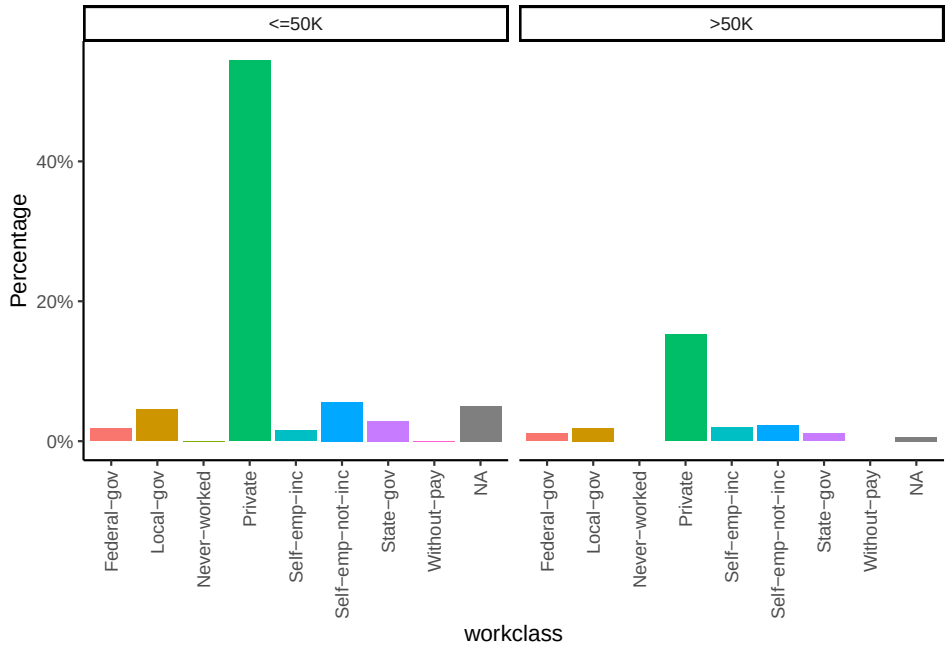

Figura 1: Datos de tipo de trabajo y salario. Se aprecia que los datos perdidos se agrupan más en la categoría de <=50k, por lo que no es totalmente aleatorio. Podríamos decir que los datos perdidos son del tipo MAR, por lo que podemos realizar la imputación de los datos. Esto ocurre también con el resto 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.

```
# Escalamos los datos numéricos
df_num = data.frame(lapply(df[,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(df$age)
```

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

```
max(df$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, pues contiene los

datos de muchos países que puede que sean reales.

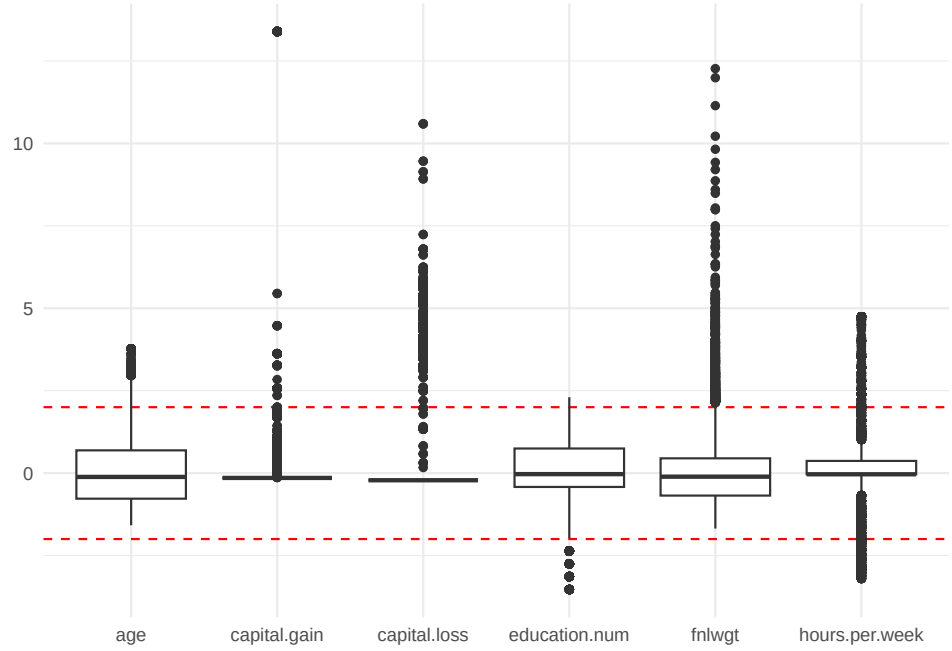

Figura 2: Outliers. Se aprecian los datos escalados para las variables cuantitativas. Las líneas rojas discontinuas representan 2 veces la desviación estandar de la media. Se aprecia que existen muchos registros que podrían ser considerados outliers, especialmente dentro de la variable 'fnlwgt'.

2.6. Descripción de los valores

2.6.1. 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(df[,numericas])
```

```
##       age            fnlwgt         education.num    capital.gain
##  Min.   :17.00   Min.   :  12285   Min.   : 1.00   Min.   :    0
##  1st Qu.:28.00   1st Qu.: 117827   1st Qu.: 9.00   1st Qu.:    0
##  Median :37.00   Median : 178356   Median :10.00   Median :    0
##  Mean   :38.58   Mean   : 189778   Mean   :10.08   Mean   : 1078
##  3rd Qu.:48.00   3rd Qu.: 237051   3rd Qu.:12.00   3rd Qu.:    0
##  Max.   :90.00   Max.   :1484705   Max.   :16.00   Max.   :99999
##   capital.loss    hours.per.week
##  Min.   :   0.0   Min.   : 1.00
##  1st Qu.:   0.0   1st Qu.:40.00
##  Median :   0.0   Median :40.00
##  Mean   :  87.3   Mean   :40.44
##  3rd Qu.:   0.0   3rd Qu.:45.00
##  Max.   :4356.0   Max.   :99.00
```

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.gain` y `capital.loss` se

encuentran en el 0. Esto debe de ser porque son variables muy asintóticas. 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

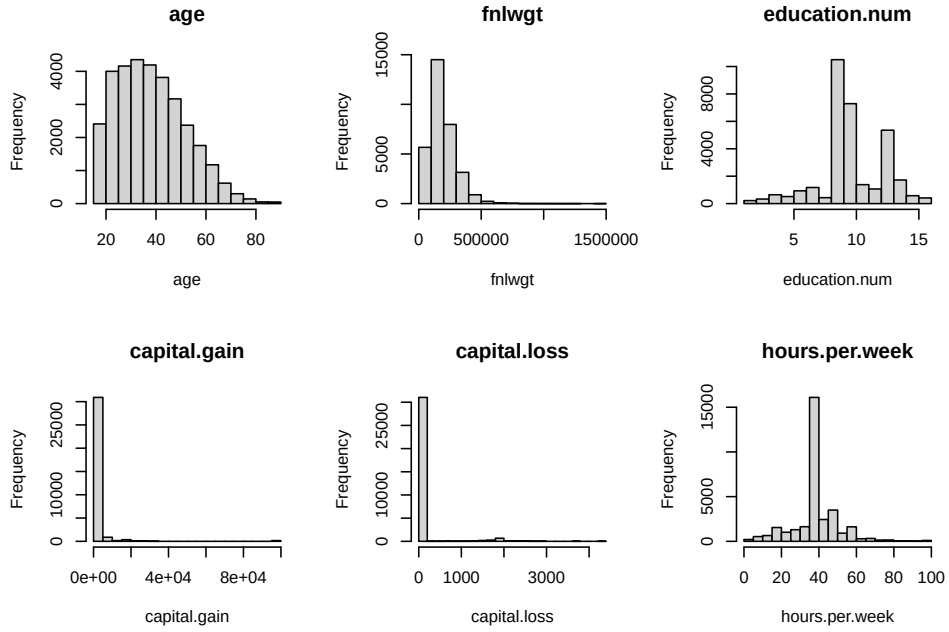

Figura 3: Distribución de los datos numéricos. Se aprecia que los datos de la edad podrían ser normales, aunque se encuentran truncados por debajo de los 18 años. Los datos de educación se encuentran entre 8 y 9 en la mayoría de los casos, así como vemos que en las horas por semana en casi todos los 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(df[,vari])$p.value
    }
    if (test == "homocedasticidad"){
      texto = "homocedásticas"
      p_val = fligner.test(df[,vari], df[["income"]])$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, fnlwgt, education.num,
## capital.gain, capital.loss, hours.per.week
```

```
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, fnlwgt, education.num,
## capital.gain, capital.loss, hours.per.week
```

```
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

2.6.2. Valores categóricos

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

```
non_num = names(df)[!names(df)%in%numericas]
# Llamamos a la función summary

summary(df[,non_num])
```

```
##             workclass            education                   marital.status
##  Private         :22696   HS-grad     :10501   Divorced              : 4443
##  Self-emp-not-inc: 2541   Some-college: 7291   Married-AF-spouse     :   23
##  Local-gov       : 2093   Bachelors   : 5355   Married-civ-spouse    :14976
##  State-gov       : 1298   Masters     : 1723   Married-spouse-absent:  418
##  Self-emp-inc    : 1116   Assoc-voc   : 1382   Never-married         :10683
##  (Other)         :  981   11th        : 1175   Separated             : 1025
##  NA's            : 1836   (Other)     : 5134   Widowed               :  993
##            occupation            relationship                    race
##  Prof-specialty : 4140   Husband       :13193   Amer-Indian-Eskimo:  311
##  Craft-repair   : 4099   Not-in-family : 8305   Asian-Pac-Islander: 1039
##  Exec-managerial: 4066   Other-relative:  981   Black             : 3124
##  Adm-clerical   : 3770   Own-child     : 5068   Other             :  271
##  Sales          : 3650   Unmarried     : 3446   White             :27816
##  (Other)        :10993   Wife          : 1568
##  NA's           : 1843
##      sex              native.country    income
##  Female:10771   United-States:29170   <=50K:24720
##  Male  :21790   Mexico       :  643   >50K : 7841
##                 Philippines  :  198
##                 Germany      :  137
##                 Canada       :  121
##                 (Other)      : 1709
##                 NA's         :  583
```

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

3. 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.