```
knitr::opts_chunk$set(warning=FALSE)
knitr::opts_chunk$set(message=FALSE)
```

```
# paquetes posiblemente necesarios
# library(readxl)
library(dplyr)
# library(lubridate)
library(tidyr)
# library(stringr)
# library(visdat)# para estudar os NAs
# library(naniar)# para estudar os NAs
# library(lubridate)
```

Exemplo de lectura de datos

Imos traballar cun conxunto de datos gardados nun ficheiro *Rdata*, que é un formato propio de R que permite gardar obxectos de `R` respentando as súas propiedades.

Gárdanse co comando `save` (`save(obxecto1,obxecto2,obxecto3,file="ficheirogardado.Rdata")`) e recupéranse que comando `load` aplicado a un nome de ficheiro (`load("ficheirogardado.Rdata")`).

Neste caso imos abrir un ficheiro con datos da Enquisa de presupostos familiares nos USA, gardados nun ficheiro denominado *adult.Rdata*.

```
load("adult2023.Rdata")
```

Carga en memoria un data.frame *df*, cunha colección de 15 variables.

Por comodidade, por non modificarf o resto do código que teño preparado vou cambiar o nome do data.frame:

```
variables=df
rm(df,df_names)#borrar o obxecto df e outro que non vou usar
```

Como son as variables?

Vémolo con dous posibles comandos:

```
str(variables)
```

```
## 'data.frame':    32566 obs. of  15 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ wrkclass      : chr  " State-gov" " Self-emp-not-inc" " Private" " Private" ...
##  $ fnlweight     : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education_lvl : chr  " Bachelors" " Bachelors" " HS-grad" " 11th" ...
##  $ edu_score     : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital_status: chr  " Never-married" " Married-civ-spouse" " Divorced" " Married-civ-spouse" ...
##  $ occupation    : chr  " Adm-clerical" " Exec-managerial" " Handlers-cleaners" " Handlers-cleaners"
##  $ relationship  : chr  " Not-in-family" " Husband" " Not-in-family" " Husband" ...
##  $ ethnic        : chr  " White" " White" " White" " Black" ...
##  $ gender        : chr  " Male" " Male" " Male" " Male" ...
##  $ cap_gain      : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ cap_loss      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hrs_wk        : int  40 13 40 40 40 40 16 45 50 40 ...
##  $ nationality   : chr  " United-States" " United-States" " United-States" " United-States" ...
##  $ income        : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

```
glimpse(variables)#do paquete dplyr
```

```
## Rows: 32,566
```

```
## Columns: 15
## $ age            <int> 39, 50, 38, 53, 28, 37, 49, 52, 31, 42, 37, 30, 23, 32,~
## $ wrkclass       <chr> " State-gov", " Self-emp-not-inc", " Private", " Privat~
## $ fnlweight      <int> 77516, 83311, 215646, 234721, 338409, 284582, 160187, 2~
## $ education_lvl  <chr> " Bachelors", " Bachelors", " HS-grad", " 11th", " Bach~
## $ edu_score      <int> 13, 13, 9, 7, 13, 14, 5, 9, 14, 13, 10, 13, 13, 12, 11,~
## $ marital_status <chr> " Never-married", " Married-civ-spouse", " Divorced", "~
## $ occupation     <chr> " Adm-clerical", " Exec-managerial", " Handlers-cleaner~
## $ relationship   <chr> " Not-in-family", " Husband", " Not-in-family", " Husba~
## $ ethnic         <chr> " White", " White", " White", " Black", " Black", " Whi~
## $ gender         <chr> " Male", " Male", " Male", " Male", " Female", " Female~
## $ cap_gain       <int> 2174, 0, 0, 0, 0, 0, 0, 0, 14084, 5178, 0, 0, 0, 0, 0, ~
## $ cap_loss       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ hrs_wk         <int> 40, 13, 40, 40, 40, 40, 16, 45, 50, 40, 80, 40, 30, 50,~
## $ nationality    <chr> " United-States", " United-States", " United-States", "~
## $ income         <chr> " <=50K", " <=50K", " <=50K", " <=50K", " <=50K", " <=5~
```

```
# names(variables)
```

Cambiar nomes de variables

Para comezar coa preparación dos datos vou cambiar os nomes das variables por outros algo máis cómodos.

Vou extraer os nomes mediante ***names()***, e como me sae un vector de texto, vou facer sustitucións de fragmentos con ***gsub()***.

Este é un dos comandos de manipulación de texto do ***R*** básico, colle un fragmento dun texto e sustitueo por un fragmento alternativo. Pódese aplicar a un vector de textos, onde sustituira en todos os elementos o mesmo fragmento, se existe.

```
# Cambiar nomes das variables: con R base
names(variables)[1:7]=c("idade","tipo","ponderacion","nivel_edu","valor_edu",
                        "estado_civil","ocupacion")
```

`dplyr` e o Tidyverse

Comezamos aquí a explicar os comandos do paquete `dplyr`, un paquete que forma parte do Tidyverse, unha colección de paquetes de manexo e transformación de datos que teñen como obxectivo facilitar o traballo do *cientifico de datos*, ou sexa, de xente que pasa todo o día traballando con datos, non só analizándoos, se non adicando a maior parte do tempo a "*limpar, pulir e dar esplendor*" para que, cando chegue ao momento de facer a verdadeira análise, os datos teñan a calidade suficiente para unha análise rápida e útil.

Tendo en conta ese obxectivo produciuse unha colección de comandos moi competitivos para manipular datos, penso que incluso poden competir con Excel en facilidade de uso, e seguro que o fan en potencia e capacidades. Podedes botar unha ollada na súa web (www.tidyverse.org).

Nesta parte do seminario adicarémonos a explicar comandos desta colección e desta filosofía, introducindo ademais a súa forma de traballar, que é un dos motivos do gran exito que están acadando.

Preparación de datos con R: `dplyr`

A preparación e limpeza de datos é o proceso que se realiza antes de comezar a análise para obter uns datos utilizables. Soe levar unha cantidade elevada de tempo, pero é inevitable, por que a calidade dos datos vai ter unha enorme influencia sobre a calidade e rapidez coa que realizamos a nosa análise.

O primeiro paso da nosa análise é ***Tabular*** os datos, ou sexa, o que se denomina en *data science* ***obter tidy data***, datos organizados en filas e columnas como unha táboa. Nesta sesión non faremos ningún exemplo, posto que os datos que proceden dunha folla de cálculo xa soen estar organizados previamente en filas e

columnas. Unicamente se podería incluir dentro desta parte o retoque dos nomes das variables, por que se buscan nomes faciles de interpretar e de manexar.

FILTROS

Os filtros consisten en quedarnos só con unha parte das variables (*horizontal*) ou dos individuos analizados (*vertical*).

Filtro horizontal

Escoller que variables queremos usar: `select()`

Usaremos o paquete dplyr, que se deberá instalar (se non o temos instalado), e unha vez instalado cargalo en memoria con `library()`. Instalalo sóunha vez, pero cargalo en memoria unha vez, cada vez que o necesitemos.

O máis razoable é colocar todos os comandos 'library()´ ao inicio do código, para saber onde recargalos en caso de necesidade, e ainda que aquí se fixo así, tamén se colocarán algúns dos paquetes que se expliquen na parte correspondente do código, para lembrar que son os que se utilizan e é necesario cargalos en memoria.

```
# necesítase a library dplyr, que se debe instalar previamente (unha vez)
#pero que se debe cargar en memoria cada vez que a volvamos a necesitar
library(dplyr)#colocamolo aquí para lembrar que é necesario este tipo de cousas,
#pero xa está arriba
```

Comezamos a usar o comando `select()`:

```
borrar=select(variables,idade,valor_edu,income)
head(borrar)
```

```
##   idade valor_edu income
## 1    39        13  <=50K
## 2    50        13  <=50K
## 3    38         9  <=50K
## 4    53         7  <=50K
## 5    28        13  <=50K
## 6    37        14  <=50K
```

iso mesmo tamén se podia conseguir con:

```
borrar=variables[c(1,5,15)]
```

Pero `select` adaptase mellor ao fluxo de traballo, sobre todo se usamos 'pipes' (tubos). Usar un pipe implica pensar no comando como unha frase:

> collo o obxecto 'variables' e entón selecciono as columnas 'id,Nombre,Provincia' e finalmente gardo o resultado en 'borrar'

Este truco funciona identificando `e entón` co simbolo `%>%`, e `finalmente gardo` con `->`

Así, en código, a frase quedaría:

```
variables %>% select(idade,valor_edu,income)->borrar
head(borrar)
```

```
##   idade valor_edu income
## 1    39        13  <=50K
## 2    50        13  <=50K
## 3    38         9  <=50K
## 4    53         7  <=50K
## 5    28        13  <=50K
## 6    37        14  <=50K
```

Algunhas opcións para `select`:

- Selecionar as columas 5,3 e 8:

```
#como imos usar sample_n, que saca unha mostra aleatoria,
#poñemos set.seed para que produza sempres os mesmos valores
set.seed(43123)
```

En lugar de gardalo nun obxecto borrar, finalizamos con `sample_n(k)` que extrae unha mostra de k unidades, con iso xa vemos o que se produce. `sample_n()` é un comando de `dplyr`

```
variables %>% select(3,5,8) %>% sample_n(20)
```

```
##    ponderacion valor_edu   relationship
## 1        51789        10      Own-child
## 2       262978        10  Not-in-family
## 3       196174         6  Not-in-family
## 4       122609         9      Unmarried
## 5       260782        10        Husband
## 6       124244         9  Not-in-family
## 7       290213        10      Own-child
## 8       339163        10      Unmarried
## 9       114691        13        Husband
## 10      308279        13        Husband
## 11      183801        14        Husband
## 12      170301         9           Wife
## 13      347292         9  Not-in-family
## 14      269168         9        Husband
## 15      281315        13  Not-in-family
## 16      102938        13        Husband
## 17      349148        13  Not-in-family
## 18      398988        10  Not-in-family
## 19      596776        10      Own-child
## 20      106812        11  Not-in-family
```

- Seleccionar todas menos algunhas: poñer signo (-) diante das que non queremos

```
#saidas oculta, para recortar espazo. Hai que executar
variables %>% select(-ponderacion) %>% sample_n(10)
variables %>% select(-c(ponderacion,cap_gain,cap_loss)) %>% sample_n(10)
variables %>% select(-ponderacion,-cap_gain,-cap_loss) %>% sample_n(10)
```

- Selecionar as que comezan, rematan ou conteñen algún texto determinado no nome

```
#saidas oculta, para recortar espazo. Hai que executar
variables %>% select(starts_with("cap")) %>% sample_n(10)
variables %>% select(ends_with("edu")) %>% sample_n(10)
variables %>% select(contains("_")) %>% sample_n(10)
#combina 2 condicións: unha e outra
variables %>% select(contains("_")&ends_with("wk")) %>% sample_n(10)
#combina 2 condicións: unha pero non a outra
variables %>% select(contains("_")&-ends_with("wk")) %>% sample_n(10)
#combina 2 condicións: unha OU a outra
variables %>% select(contains("edu")|ends_with("wk")) %>% sample_n(10)
variables %>% select(contains("edu"),ends_with("wk")) %>% sample_n(10)
```

```
# Para practicar
```

Filtro vertical

Escoller que filas queremos usar: `filter()`

Vou facer estes exemplos cunha base de datos máis pequena, menos variables:

```
variables %>% select("idade", "tipo", "nivel_edu","valor_edu","estado_civil","ocupacion",
                "ethnic", "gender", "hrs_wk","nationality" , "income" )->datos
#ademais fago un 1º: elimino as filas de Activo circulante nas que hai NA, (falta o dato)
#collo o dataframe 'borrar' e entón filtro as filas 'que non son NA en A_c_2019' e
# 'que non son NA en A_c_2018', finalmente gardo o resultado en borrar, outra vez
datos %>% filter(!is.na(idade))->datos
summary(datos)
```

```
##      idade           tipo            nivel_edu           valor_edu
##  Min.   :17.00   Length:32561       Length:32561       Min.   : 1.00
##  1st Qu.:28.00   Class :character   Class :character   1st Qu.: 9.00
##  Median :37.00   Mode  :character   Mode  :character   Median :10.00
##  Mean   :38.58                                         Mean   :10.08
##  3rd Qu.:48.00                                         3rd Qu.:12.00
##  Max.   :90.00                                         Max.   :16.00
##  estado_civil        ocupacion            ethnic             gender
##  Length:32561       Length:32561       Length:32561       Length:32561
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      hrs_wk      nationality           income
##  Min.   : 1.00   Length:32561       Length:32561
##  1st Qu.:40.00   Class :character   Class :character
##  Median :40.00   Mode  :character   Mode  :character
##  Mean   :40.44
##  3rd Qu.:45.00
##  Max.   :99.00
```

```
### Como só vou facer probas vou crear un obxecto borrar, que non importa que se estrague
borrar=datos
```

Comandos adaptados da chuleta `5_dplyr.r`:

```
#filas que teñan "valor_edu >10"
borrar%>%filter(valor_edu >10) %>% sample_n(10)
```

```
##    idade       tipo   nivel_edu valor_edu           estado_civil
## 1     22    Private   Assoc-voc        11          Never-married
## 2     62    Private     Masters        14          Never-married
## 3     46    Private   Assoc-voc        11  Married-spouse-absent
## 4     62    Private     Masters        14               Divorced
## 5     62  Local-gov   Doctorate        16                Widowed
## 6     39    Private     Masters        14          Never-married
## 7     39    Private     Masters        14          Never-married
## 8     46    Private  Assoc-acdm        12          Never-married
## 9     71  Local-gov     Masters        14                Widowed
## 10    28    Private   Bachelors        13     Married-civ-spouse
##             ocupacion ethnic  gender hrs_wk    nationality income
## 1     Exec-managerial  White    Male     40  United-States  <=50K
```

```
## 2      Prof-specialty  White    Male     70  United-States  <=50K
## 3   Machine-op-inspct  White    Male     40  United-States  <=50K
## 4      Prof-specialty  White  Female     45  United-States  <=50K
## 5      Prof-specialty  White  Female     40           Iran  <=50K
## 6     Exec-managerial  White  Female     45  United-States  <=50K
## 7       Other-service  White  Female     32  United-States  <=50K
## 8        Adm-clerical  White  Female     40  United-States  <=50K
## 9      Prof-specialty  White  Female     40  United-States  <=50K
## 10    Exec-managerial  White    Male     46  United-States   >50K
```

```
#filas que teñan "nationality igual a ' Ireland' ou ' India'"
#Ollo!!! que antes do nome do país colocaron un espazo en branco:
borrar%>%filter(nationality==" Ireland" | nationality==" India")  %>% sample_n(20)
```

```
##    idade              tipo     nivel_edu valor_edu           estado_civil
## 1     30           Private       HS-grad         9     Married-civ-spouse
## 2     31           Private          11th         7  Married-spouse-absent
## 3     33         State-gov       Masters        14     Married-civ-spouse
## 4     29           Private     Doctorate        16     Married-civ-spouse
## 5     32           Private       HS-grad         9     Married-civ-spouse
## 6     59           Private   Prof-school        15     Married-civ-spouse
## 7     55         State-gov     Doctorate        16     Married-civ-spouse
## 8     21           Private  Some-college        10          Never-married
## 9     29           Private     Bachelors        13     Married-civ-spouse
## 10    49           Private       Masters        14  Married-spouse-absent
## 11    42      Self-emp-inc       Masters        14               Divorced
## 12    34           Private   Prof-school        15          Never-married
## 13    51  Self-emp-not-inc   Prof-school        15     Married-civ-spouse
## 14    27           Private  Some-college        10          Never-married
## 15    27           Private          11th         7  Married-spouse-absent
## 16    26           Private       Masters        14          Never-married
## 17    48           Private  Some-college        10          Never-married
## 18    36           Private       Masters        14                Widowed
## 19    47           Private       Masters        14              Separated
## 20    29  Self-emp-not-inc       HS-grad         9  Married-spouse-absent
##             ocupacion              ethnic  gender hrs_wk nationality income
## 1   Machine-op-inspct               White    Male     60     Ireland   >50K
## 2   Handlers-cleaners  Asian-Pac-Islander    Male     40       India  <=50K
## 3      Prof-specialty  Asian-Pac-Islander    Male     19       India  <=50K
## 4      Prof-specialty  Asian-Pac-Islander    Male     60       India  <=50K
## 5     Exec-managerial               White    Male     40     Ireland  <=50K
## 6      Prof-specialty  Asian-Pac-Islander    Male     40       India   >50K
## 7      Prof-specialty  Asian-Pac-Islander    Male     40       India   >50K
## 8               Sales  Asian-Pac-Islander    Male     30       India  <=50K
## 9        Craft-repair  Asian-Pac-Islander    Male     30       India  <=50K
## 10    Protective-serv  Asian-Pac-Islander    Male     40       India  <=50K
## 11    Exec-managerial  Asian-Pac-Islander    Male     60       India   >50K
## 12       Tech-support  Asian-Pac-Islander    Male     40       India   >50K
## 13     Prof-specialty               Other    Male     70       India   >50K
## 14       Adm-clerical  Asian-Pac-Islander    Male     40       India  <=50K
## 15              Sales  Asian-Pac-Islander    Male     35       India  <=50K
## 16     Prof-specialty  Asian-Pac-Islander    Male     20       India  <=50K
## 17    Exec-managerial               White  Female     40     Ireland   >50K
## 18       Tech-support  Asian-Pac-Islander  Female     40       India  <=50K
## 19  Machine-op-inspct  Asian-Pac-Islander  Female     42       India  <=50K
```

```
## 20   Transport-moving  Asian-Pac-Islander    Male     50       India   >50K
```

```
#filas que "a nationality estea en c(" Laos"," Mexico"," Poland"," Jamaica")
borrar%>%filter(nationality%in%c(" Laos"," Mexico"," Poland"," Jamaica")) %>% sample_n(10)
```

```
##    idade              tipo     nivel_edu valor_edu           estado_civil
## 1     26  Self-emp-not-inc  Some-college        10          Never-married
## 2     24           Private     Bachelors        13          Never-married
## 3     27              <NA>       5th-6th         3     Married-civ-spouse
## 4     38           Private           9th         5  Married-spouse-absent
## 5     29           Private       5th-6th         3          Never-married
## 6     42  Self-emp-not-inc       HS-grad         9     Married-civ-spouse
## 7     38           Private       1st-4th         2  Married-spouse-absent
## 8     59           Private           9th         5             Separated
## 9     24           Private          10th         6     Married-civ-spouse
## 10    34           Private     Bachelors        13     Married-civ-spouse
##             ocupacion ethnic  gender hrs_wk nationality income
## 1      Prof-specialty  White  Female      4      Mexico  <=50K
## 2               Sales  White    Male     55     Jamaica  <=50K
## 3               <NA>  White  Female     40      Mexico  <=50K
## 4   Handlers-cleaners  White    Male     40      Mexico  <=50K
## 5       Other-service  White    Male     25      Mexico  <=50K
## 6        Craft-repair  Black    Male     25     Jamaica  <=50K
## 7        Craft-repair  White    Male     40      Mexico  <=50K
## 8   Machine-op-inspct  White  Female     40      Mexico  <=50K
## 9       Other-service  White    Male     40      Mexico  <=50K
## 10              Sales  White    Male     45     Jamaica  <=50K
```

```
#filas que non son " United-States" e cumpren as outras dúas condicións:
borrar%>%filter(nationality!=" United-States",valor_edu >10,idade==21) %>% sample_n(2)
```

```
##   idade     tipo  nivel_edu valor_edu        estado_civil      ocupacion ethnic
## 1    21  Private  Bachelors        13       Never-married   Adm-clerical  White
## 2    21  Private  Assoc-voc        11  Married-civ-spouse  Other-service  White
##    gender hrs_wk nationality income
## 1    Male     40    Columbia  <=50K
## 2  Female     50      Mexico  <=50K
```

```
# https://sebastiansauer.github.io/dplyr_filter/
# (nesta web hai unha moi boa coleccion de exemplos de filter)
library(stringr)#usando esta library, filtrar nationality que comezan por ' L' e rematan en 's'
borrar%>%filter(str_detect(nationality, "^ L"),str_detect(nationality, "s$")) %>% sample_n(10)
```

```
##    idade         tipo     nivel_edu valor_edu        estado_civil
## 1     53      Private       5th-6th         3  Married-civ-spouse
## 2     23      Private     Preschool         1       Never-married
## 3     56  Federal-gov  Some-college        10  Married-civ-spouse
## 4     36      Private  Some-college        10       Never-married
## 5     49      Private       7th-8th         4       Never-married
## 6     44      Private       HS-grad         9       Never-married
## 7     27      Private       HS-grad         9  Married-civ-spouse
## 8     36      Private       HS-grad         9  Married-civ-spouse
## 9     27      Private     Bachelors        13       Never-married
## 10    30      Private       HS-grad         9  Married-civ-spouse
##             ocupacion              ethnic  gender hrs_wk nationality income
## 1   Machine-op-inspct  Asian-Pac-Islander    Male     40        Laos  <=50K
```

```
## 2       Other-service  Asian-Pac-Islander  Female     40        Laos  <=50K
## 3        Adm-clerical  Asian-Pac-Islander    Male     40        Laos  <=50K
## 4        Craft-repair  Asian-Pac-Islander    Male     40        Laos  <=50K
## 5   Machine-op-inspct  Asian-Pac-Islander    Male     45        Laos  <=50K
## 6   Machine-op-inspct  Asian-Pac-Islander  Female     45        Laos  <=50K
## 7   Machine-op-inspct  Asian-Pac-Islander    Male     40        Laos  <=50K
## 8   Machine-op-inspct  Asian-Pac-Islander    Male     40        Laos  <=50K
## 9        Adm-clerical  Asian-Pac-Islander  Female     50        Laos  <=50K
## 10       Craft-repair  Asian-Pac-Islander    Male     40        Laos  <=50K
```

```
#podense enganchar no pipe varios comandos:
#a mesma seleccion de antes, pero que só saque o idade e nivel_edu:
borrar%>%filter(str_detect(nationality, "^ L"),str_detect(nationality, "s$")) %>%
  select(idade,nivel_edu) %>% sample_n(10)
```

```
##    idade     nivel_edu
## 1     42    Assoc-acdm
## 2     27     Bachelors
## 3     49       7th-8th
## 4     53       5th-6th
## 5     29       HS-grad
## 6     23     Preschool
## 7     27       HS-grad
## 8     56  Some-college
## 9     19          11th
## 10    30       HS-grad
```

```
# unha liña con pipe podese separar en duas, sempre que se remate a primeira co pipe,
#ou algo que indique a R que sigue
```

Xestionar datos faltantes (NA's)

Cando a un dos individuos da mostra lle falta un dato R indícao coa expresión `NA`, e nos datos reais é moi frecuente que haxa datos faltantes, ás veces moitos.

Por exemplo, no obxecto orixinal *variables* hai varios.

Vaise comentar algúns comandos que axudan a xestionar estes casos. Primeiro con comandos de exploración, para ver como están distribuidos, e despois con comandos de dplyr que os poden sustituir.

Técnicamente, en Estatística existen técnicas (imputación) que poden propoñer valores sustitutivos, pero é algo que supera con moito o obxectivo deste seminario.

Mirando como son os NA: exploración

R ten un comando para preguntar se un valor é NA: ***is.na()***, e outros máis, por exemplo ***na.omit()*** que eliminan todos os elementos con datos faltantes:

```
## Exemplo
#crear un data.frame para manipular
borrar=variables %>% select(idade,valor_edu,ocupacion,nationality) %>% filter(str_detect(nationality, "
tail(borrar)
```

```
##       idade valor_edu        ocupacion    nationality
## 29170    52         9  Exec-managerial  United-States
## 29171    NA         9  Exec-managerial  United-States
## 29172    NA         9  Exec-managerial  United-States
## 29173    NA         9  Exec-managerial  United-States
## 29174    NA         9  Exec-managerial  United-States
```

```
## 29175    NA         9  Exec-managerial  United-States
```

```
#que filas teñen NA en activo circulante
which(is.na(borrar$idade))
```

```
## [1] 29171 29172 29173 29174 29175
```

```
#borroas
borrar2=na.omit(borrar)
tail(borrar2)
```

```
##       idade valor_edu          ocupacion    nationality
## 29165    22        10    Protective-serv  United-States
## 29166    27        12       Tech-support  United-States
## 29167    40         9  Machine-op-inspct  United-States
## 29168    58         9       Adm-clerical  United-States
## 29169    22         9       Adm-clerical  United-States
## 29170    52         9    Exec-managerial  United-States
```

```
which(is.na(borrar2$idade))
```

```
## integer(0)
```

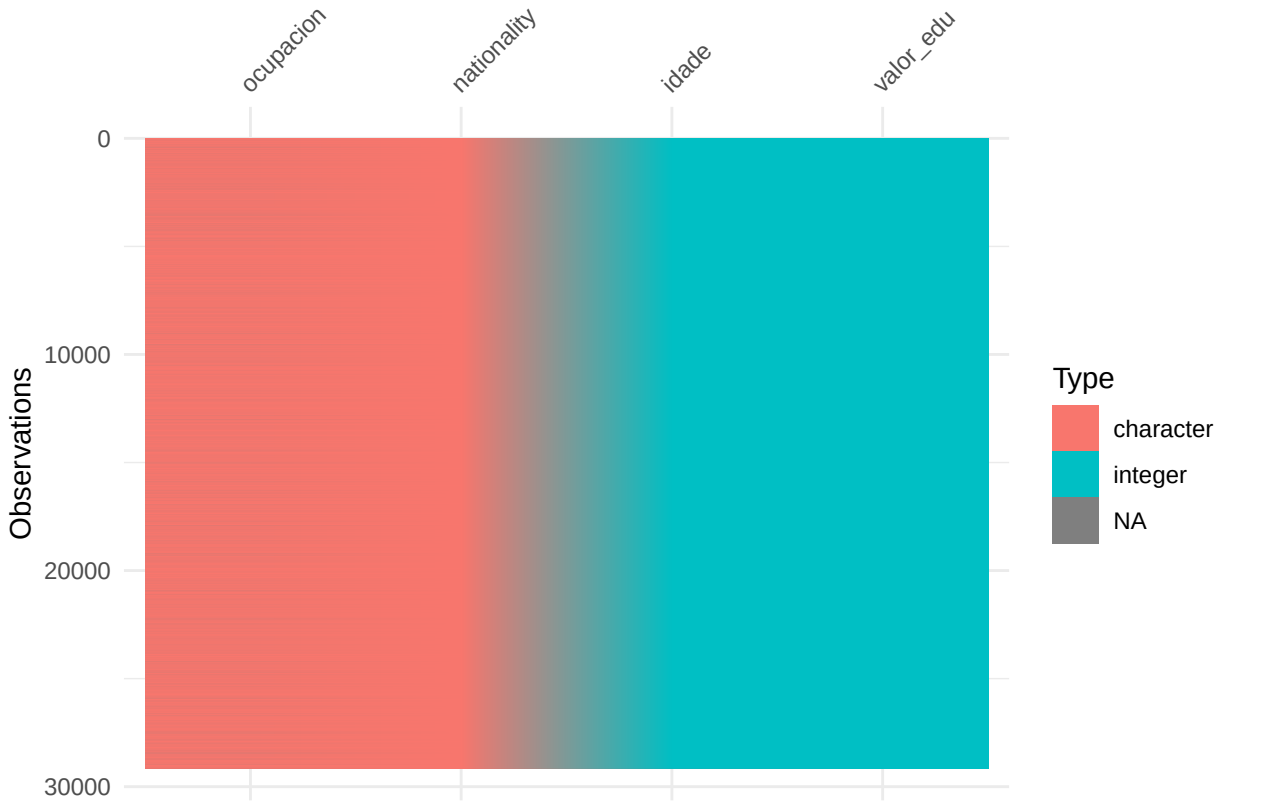
Os comandos ***colSums*** e ***rowSums*** permiten sumar os data.frames por columnas ou filas. Tamén temos que o comando ***is.na()*** produce un valor lóxico (`TRUE`, `FALSE`) como resultado, que cando se suman interprétanse como `1` e `0` respectivamente.

Combinando eses comandos podemos saber cantos NA hai en cada columna ou fila:

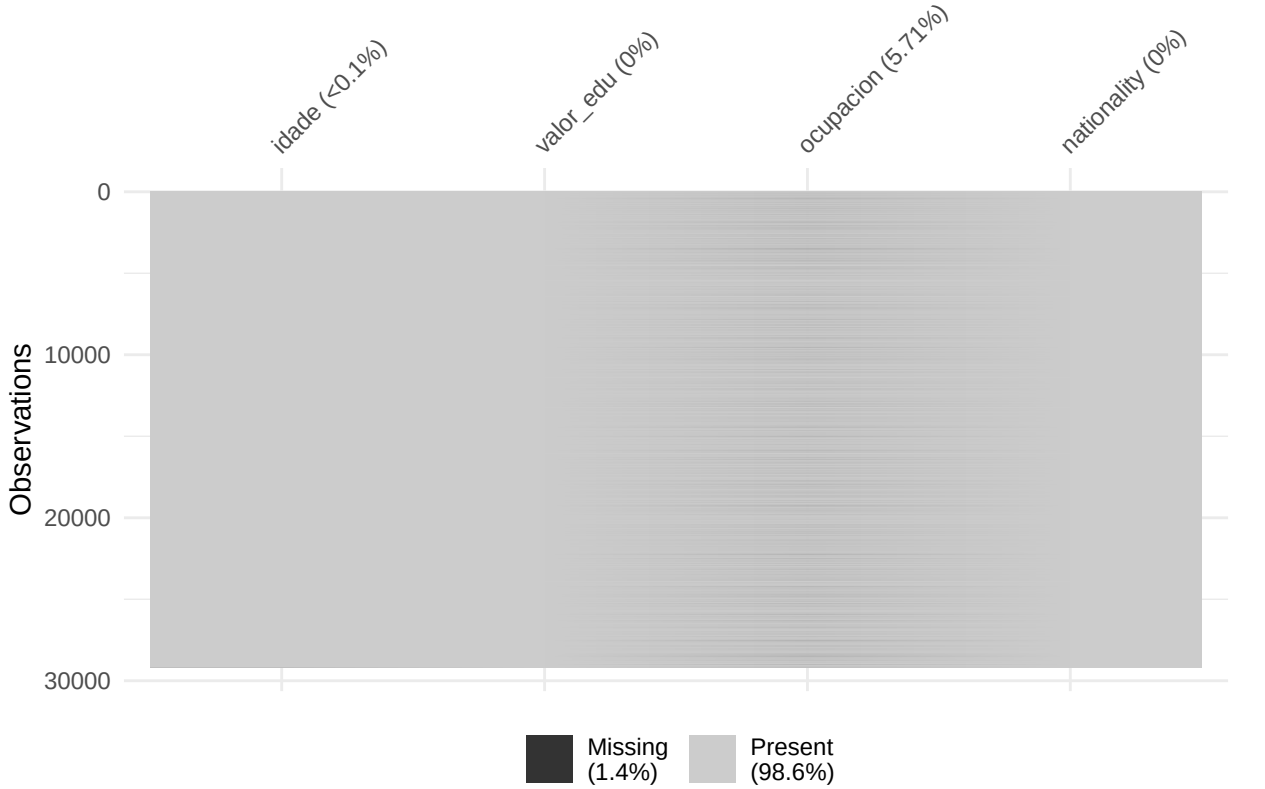
```
#saidas oculta, para recortar espazo. Hai que executar

#Cantos NA ten cada variable (columna)
colSums(is.na(datos))
#Cantos NA ten cada individuo?, gárdoo en auxiliar para aproveitalo máis adiante
(auxiliar=rowSums(is.na(datos)))#encerrar todo o comando entre parentese fai que o garde e o amose á ve
```

Vou amosar agora dous paquetes específicos para xestionar e revisar NAs. Primeiro `visdat`, que me aporta unha visión gráfica.

```
#Visualizar os NA por variable
library(visdat)# paquete para revisar os NAs
vis_dat(borrar) #visualmente
```

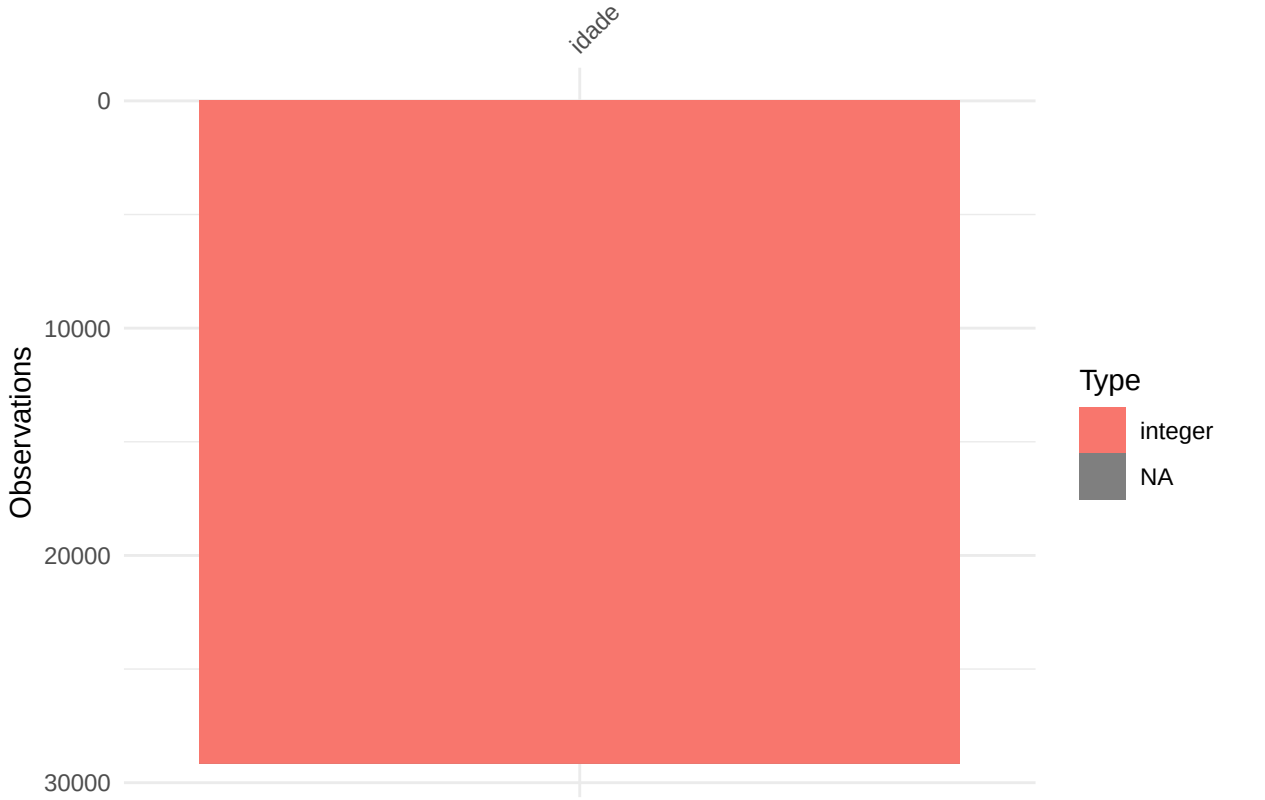



```
vis_miss(borrar)
```

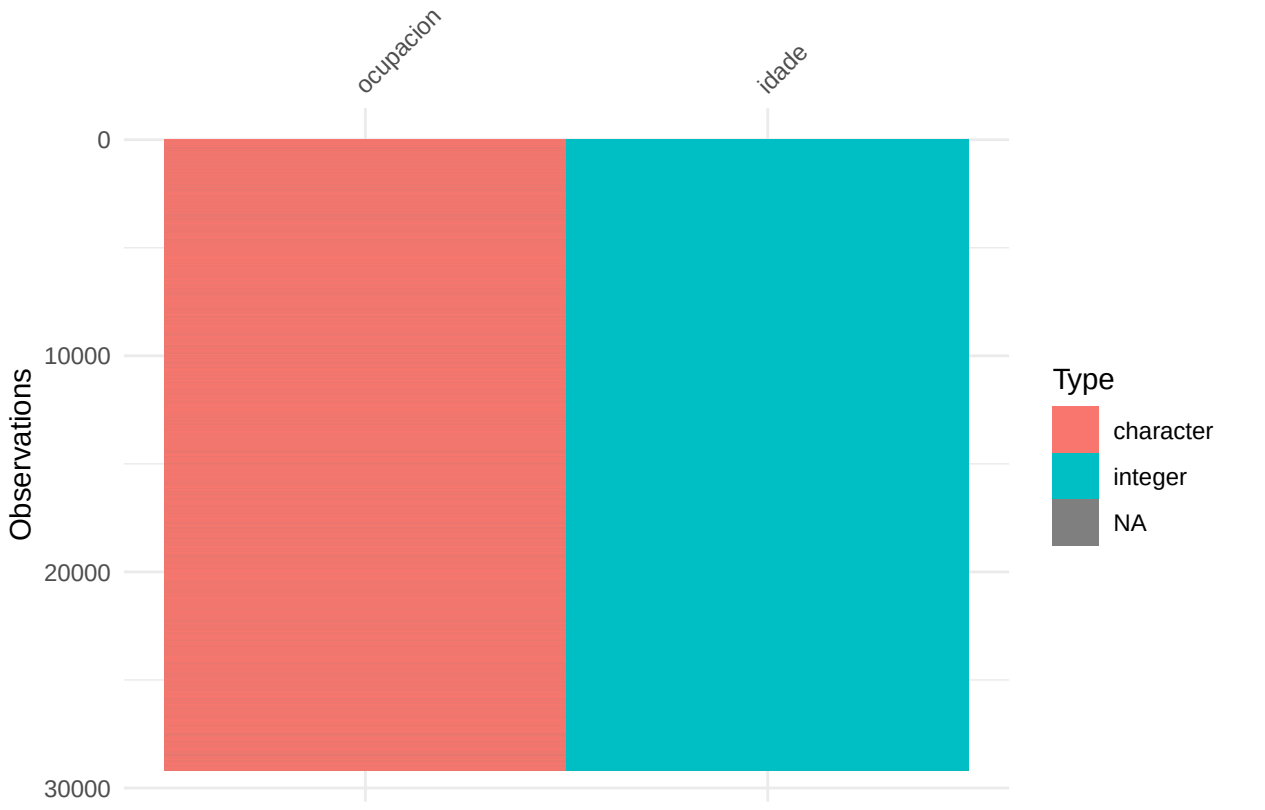


Vese mellor partindo as variables en duas partes.

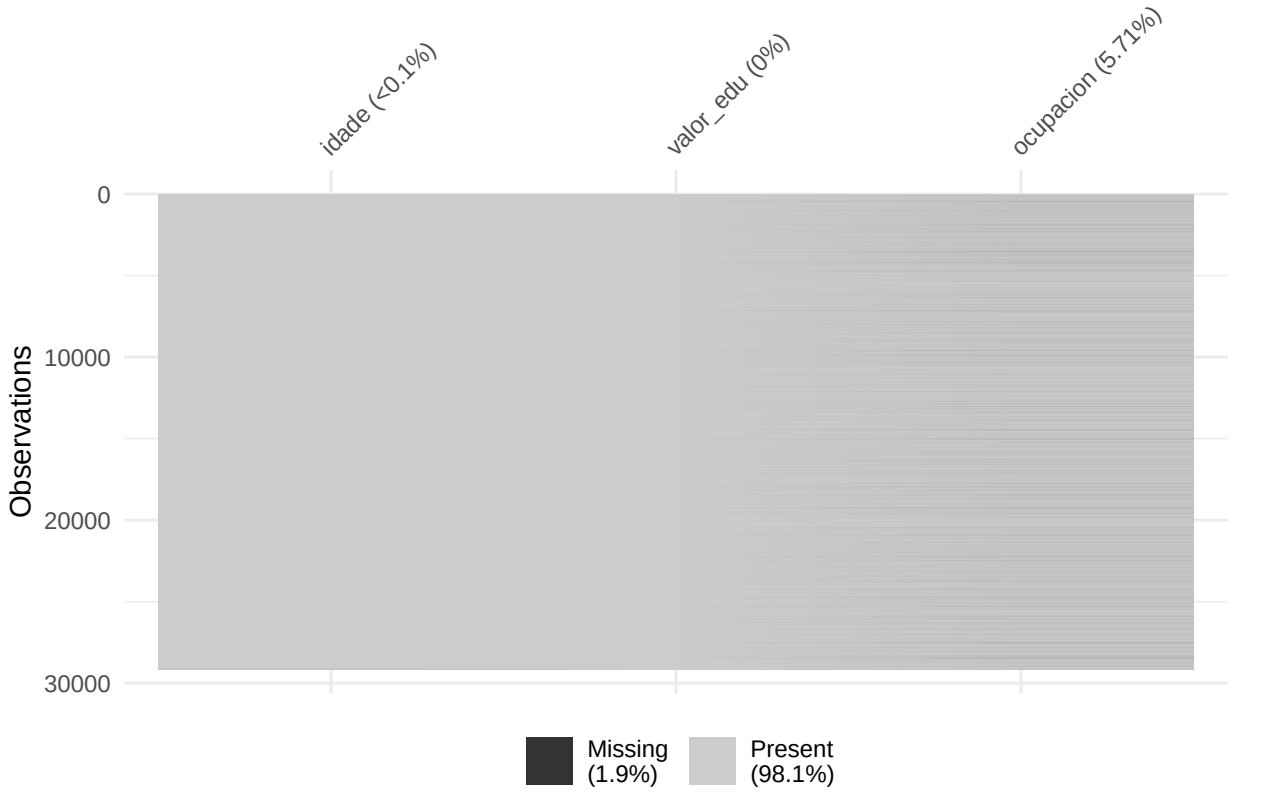
```r
# pódese ver co zoom, pero
vis_dat(borrar[1]) #visualmente
```




```r
vis_dat(borrar[c(1,3)])
```




```
vis_miss(borrar[c(1:3)])
```



Vense 4 grupos de variables que acumulan a maioría dos NAs: as relacionadas con *Deudas financieras*, con *Materiales*, con *Pasivo fijo*, e en menor medida con *Numero de empleados*

O paquete `naniar` tamén analiza os NAs, pero con táboas:

```
library(naniar)# outro paquete para revisar os NAs
#taboa de individuos con numero de NAs
miss_case_table(borrar)
```

```
## # A tibble: 2 x 3
##   n_miss_in_case n_cases pct_cases
##            <int>   <int>     <dbl>
## 1              0   27504     94.3
## 2              1    1671      5.73
```

```
# resume: nº de NAs por variable
miss_var_summary(borrar)
```

```
## # A tibble: 4 x 3
##   variable    n_miss pct_miss
##   <chr>        <int>    <dbl>
## 1 ocupacion     1666   5.71
## 2 idade            5   0.0171
## 3 valor_edu        0   0
## 4 nationality      0   0
```

Claramente *Ocupacion* acumula mais NAs

Que facer?

As solucións buscanse en función do coñecemento do tema que se trate. Neste caso o NA de ocupación significa simplemente que non esta a traballar. Podemos deixalo así ou máis adiante darlle unha etiqueta de "Non activo".

Transformación de variables (crear variables novas)

`dplyr` crea novas variables con `mutate()`

Previamente vou facer algun exemplo de cambio de nomes de variables, co comando `rename()`:

```
#volvo a borrar con datos
borrar=datos
#######CAMBIAR OS NOMES DAS VARIABLES: rename(new_name = old_name)
borrar%>%  rename(gana=hrs_wk)  %>% sample_n(10)
```

```
##    idade          tipo     nivel_edu valor_edu        estado_civil
## 1     19       Private          11th         7       Never-married
## 2     67          <NA>       HS-grad         9  Married-civ-spouse
## 3     21       Private  Some-college        10       Never-married
## 4     90       Private       HS-grad         9             Widowed
## 5     37  Self-emp-inc     Assoc-voc        11  Married-civ-spouse
## 6     56       Private       HS-grad         9            Divorced
## 7     38       Private    Assoc-acdm        12  Married-civ-spouse
## 8     44  Self-emp-inc     Bachelors        13  Married-civ-spouse
## 9     21       Private  Some-college        10       Never-married
## 10    50     Local-gov       HS-grad         9  Married-civ-spouse
##             ocupacion ethnic  gender gana    nationality income
## 1       Other-service  Black    Male   40  United-States  <=50K
## 2                <NA>  Black    Male   20  United-States  <=50K
## 3   Handlers-cleaners  White    Male   40  United-States  <=50K
## 4    Transport-moving  White    Male   99  United-States  <=50K
## 5               Sales  White    Male   60  United-States   >50K
```

```
## 6        Adm-clerical  White  Female   32  United-States  <=50K
## 7      Prof-specialty  White    Male   40  United-States   >50K
## 8     Exec-managerial  White    Male   45  United-States   >50K
## 9       Other-service  White  Female   20           <NA>  <=50K
## 10    Exec-managerial  Black  Female   40  United-States   >50K
```

```
borrar%>%  rename(gana=hrs_wk,educacion=nivel_edu)  %>% sample_n(10)
```

```
##    idade              tipo     educacion valor_edu        estado_civil
## 1     61           Private       7th-8th         4  Married-civ-spouse
## 2     32           Private          11th         7       Never-married
## 3     36           Private       HS-grad         9            Divorced
## 4     33           Private       HS-grad         9  Married-civ-spouse
## 5     29           Private  Some-college        10  Married-civ-spouse
## 6     55         Local-gov   Prof-school        15            Divorced
## 7     36             <NA>       HS-grad         9  Married-civ-spouse
## 8     63           Private          12th         8  Married-civ-spouse
## 9     47           Private       HS-grad         9  Married-civ-spouse
## 10    50  Self-emp-not-inc       Masters        14  Married-civ-spouse
##            ocupacion               ethnic  gender gana    nationality income
## 1       Craft-repair                White    Male   40  United-States  <=50K
## 2       Craft-repair  Asian-Pac-Islander    Male   40          India  <=50K
## 3   Transport-moving                White    Male   43           <NA>  <=50K
## 4    Exec-managerial                White    Male   50  United-States  <=50K
## 5       Adm-clerical  Amer-Indian-Eskimo    Male   40  United-States  <=50K
## 6     Prof-specialty                White  Female   39  United-States  <=50K
## 7               <NA>                White    Male   40  United-States   >50K
## 8   Transport-moving                White    Male   40           Cuba  <=50K
## 9             Sales                 White    Male   55  United-States  <=50K
## 10            Sales                 White    Male   55  United-States   >50K
```

Manexar NAs con tidyr e dplyr

Agora vamos modificar as variables que tiñan NAs, segundo os criterios comentados antes. Usaremos unha función, pero amosaremos outra función de dplyr relacionada con NAs:

- reemprazar NAs por un valor específico: `replace_NA()` (paquete tidyr)

Este comando sustitúe os NAs polo valor que lle indiquemos.

Fágoo con *ocupacion*

```
#Para unha unica variable:ocupacion
borrar %>%  mutate(ocupacion=replace_na(ocupacion,"inactiv@")) %>% sample_n(12)
```

```
##    idade          tipo     nivel_edu valor_edu        estado_civil
## 1     36  Self-emp-inc     Bachelors        13            Divorced
## 2     38       Private          10th         6           Separated
## 3     27       Private  Some-college        10  Married-civ-spouse
## 4     29       Private       HS-grad         9  Married-civ-spouse
## 5     17          <NA>          10th         6       Never-married
## 6     32       Private       HS-grad         9       Never-married
## 7     45   Federal-gov     Bachelors        13  Married-civ-spouse
## 8     33       Private  Some-college        10  Married-civ-spouse
## 9     38       Private   Prof-school        15  Married-civ-spouse
## 10    40     Local-gov       HS-grad         9       Never-married
## 11    23       Private    Assoc-acdm        12  Married-civ-spouse
## 12    32       Private  Some-college        10  Married-civ-spouse
```

```
##             ocupacion ethnic  gender hrs_wk    nationality income
## 1               Sales  White    Male     60  United-States  <=50K
## 2        Adm-clerical  White    Male     40  United-States  <=50K
## 3      Prof-specialty  White    Male     50  United-States   >50K
## 4   Machine-op-inspct  White    Male     40  United-States  <=50K
## 5            inactiv@  Black    Male     20  United-States  <=50K
## 6               Sales  White  Female     45  United-States  <=50K
## 7     Exec-managerial  White    Male     40  United-States  <=50K
## 8   Handlers-cleaners  White    Male     50  United-States   >50K
## 9      Prof-specialty  White    Male     40  United-States   >50K
## 10      Other-service  White    Male     40  United-States  <=50K
## 11              Sales  White  Female     25  United-States  <=50K
## 12    Exec-managerial  White    Male     50        Germany  <=50K
```

Falta por explicar outro comando de dplyr para NAs:

- transformar un valor específico en NA: `na_if()`

Fai o contrario de `replace_na()` colle un valor e cambiao por NA. Para velo vou probar a cambiar os " >50K de *invome* por NAs.

```
# manexar NAs
borrar %>% select(idade,tipo,income)%>% mutate(income=na_if(income," >50K")) %>%  sample_n(25)
```

```
##    idade             tipo income
## 1     72          Private  <=50K
## 2     28          Private  <=50K
## 3     48          Private  <=50K
## 4     48          Private  <=50K
## 5     51          Private  <=50K
## 6     36          Private   <NA>
## 7     21          Private  <=50K
## 8     48          Private  <=50K
## 9     38  Self-emp-not-inc   <NA>
## 10    18          Private  <=50K
## 11    27          Private  <=50K
## 12    24             <NA>  <=50K
## 13    45       Federal-gov  <=50K
## 14    29             <NA>  <=50K
## 15    22          Private  <=50K
## 16    28        Local-gov  <=50K
## 17    37          Private  <=50K
## 18    21          Private  <=50K
## 19    67             <NA>  <=50K
## 20    28          Private  <=50K
## 21    45          Private  <=50K
## 22    32             <NA>  <=50K
## 23    26          Private  <=50K
## 24    44       Federal-gov   <NA>
## 25    25             <NA>  <=50K
```

Transformación de variables (crear variables novas): CUANTITATIVAS

- `mutate()`

Permite crear novas columnas (variables) ou transformar as antigas, se asignamos o mesmo nome á nova variable

```
#######CREAR NOVAS COLUMNAS CON NOVAS VARIABLES: mutate()
#crea unha variable nova facendo calculos con variables que xa existen:
borrar %>% select(idade,tipo,income,valor_edu)%>%
  mutate(cadrado=valor_edu^2)%>% sample_n(10)
```

```
##    idade              tipo income valor_edu cadrado
## 1     52           Private  <=50K        14     196
## 2     43           Private   >50K        10     100
## 3     33  Self-emp-not-inc  <=50K        11     121
## 4     41           Private  <=50K         9      81
## 5     26           Private  <=50K        10     100
## 6     34         Local-gov  <=50K        12     144
## 7     29           Private  <=50K        13     169
## 8     39           Private  <=50K        14     196
## 9     43  Self-emp-not-inc  <=50K         9      81
## 10    59           Private   >50K        10     100
```

```
#crea duas variables novas
borrar %>% select(idade,tipo,income,valor_edu)%>%
  mutate(cadrado=valor_edu^2, diferencia=cadrado-valor_edu) %>% sample_n(10)
```

```
##    idade              tipo income valor_edu cadrado diferencia
## 1     46           Private  <=50K         7      49         42
## 2     42           Private  <=50K         9      81         72
## 3     18           Private  <=50K         7      49         42
## 4     25           Private  <=50K        10     100         90
## 5     50  Self-emp-not-inc   >50K        15     225        210
## 6     39           Private   >50K        13     169        156
## 7     44           Private  <=50K         9      81         72
## 8     37      Self-emp-inc   >50K         9      81         72
## 9     30           Private  <=50K         9      81         72
## 10    37  Self-emp-not-inc   >50K        13     169        156
```

```
#modifica unha variable
borrar %>% select(idade,tipo,income,valor_edu)%>%
  mutate(idade=idade+1) %>% sample_n(10)
```

```
##    idade              tipo income valor_edu
## 1     47         Local-gov  <=50K        11
## 2     23           Private  <=50K        10
## 3     41           Private   >50K        10
## 4     40  Self-emp-not-inc   >50K        16
## 5     27           Private  <=50K        10
## 6     41           Private  <=50K         7
## 7     40  Self-emp-not-inc   >50K        15
## 8     73              <NA>  <=50K         7
## 9     23           Private  <=50K        11
## 10    24           Private  <=50K         9
```

manexar datas:`lubridate`

Imos crear variables, manexando datas. Para o exemplo vou extraer datos dun ficheiro *csv*, chamado *Face2.csv*

```
Fake=read.csv2("Fake2.csv")
str(Fake)
```

```
## 'data.frame':    631 obs. of  6 variables:
```

```
##  $ Nombre                 : chr  "Gy144" "Ft3553" "Yi1372" "Jb2931" ...
##  $ Provincia              : chr  "Madrid" "Madrid" "Cantabria" "Madrid" ...
##  $ Comunidad.autónoma     : chr  "Comunidad de Madrid" "Comunidad de Madrid" "Cantabria" "Comunidad
##  $ Código.postal          : int  28020 28002 39002 28020 28005 28006 46011 42306 8034 36346 ...
##  $ Fecha                  : chr  "19/06/2010" "15/11/2007" "26/10/1999" "16/05/1990" ...
##  $ Importador...Exportador: chr  "Exportador" "Exportador" "Exportador" "No realiza actividad exteri
```

Temos unha columna "*Fecha*", que corresponde a unha data, ainda que aparece como tipo `character`. O primeiro paso será asignarlle un tipo `date`, para que identifique esa variable como unha data.

`R` basico ten comandos para xestionar datas, pero son liosos. Son moito máis manexables os comandos do paquete `lubridate`, un paquete para manexo de datas e que tamén forma parte do `tidyverse`. Podedes explorar os seus comandos na súa web

```
library(lubridate)
# "19/06/2010" indica dia, mes, ano polo que hai que usar un comando dmy() para darlle
# o formato de data
dmy("19/06/2010")
```

```
## [1] "2010-06-19"
```

```
#agora para toda a columna. Vouno facer con R básico por que é facil
Fake$Fecha=dmy(Fake$Fecha)
#pero tamén se podería ter feito con dplyr. O que vos sexa máis fácil
#### Fake %>% mutate(Fecha=dmy(Fecha))->Fake
```

Agora, empregando comandos de `lubridate` vou crear variables novas que se recollan o dia, o mes ou o ano da columna *Fecha*.

```
Fake %>% mutate(dia=day(Fecha),mes=month(Fecha),ano=year(Fecha),sdia=wday(Fecha,label=TRUE))->Fake
```

Discretizar: `case_when()`

Discretizar é unha operación con datos que consiste en transformar unha variable continua en discreta (ou non atributo). É o proceso que se aplica cando agrupamos unha variable en intervalos, só que a cada intervalo asignariaselle un valor (discreta) ou un nome (atributo).

O comando que se pode usar é `case_when()` que asigna un resultado a cada individuo en función do seu valor.

```
#crease un atributo tempo
borrar %>% mutate(tempo=case_when(
                hrs_wk<=10~"escaso",
                hrs_wk<=20~"baixo",
                hrs_wk<=30~"esta ben",
                hrs_wk<=40~"normal",
                hrs_wk<=50~"alto",
                hrs_wk<=60~"excesivo",
                TRUE~"imposible"))->borrar #TRUE significa que todos os demais casos son veterana
table(borrar$tempo)
```

```
##
##      alto     baixo    escaso  esta ben  excesivo imposible    normal
##      5938      2192       736      2317      2533      1110     17735
```

Nun `case_when()` poden combinarse duas ou máis variables, por exemplo:

```
borrar %>% mutate(nada=case_when(
  (ethnic==" White"& nationality==" United-States")~"Branco_USA",
  (ethnic==" Black"& nationality==" United-States")~"Afro_USA",
```

```
  (nationality==" United-States")~"Outro_USA",
  TRUE~"O resto"))->borrar
```

Transformación de variables (crear variables novas): CUALITATIVAS

Vanse ver agora algunha das operacións a realizar con variables cuanlitativas.

Recodificar

Para un factor (ou un vector de caracteres) consiste en cambiar as etiquetas (nomes) que se lle asignan a cada elemento. No noso caso, por exemplo, podemos abreviar os nomes de `tipo`.

```
# recodificar
borrar%>%mutate(tipo = recode(tipo," State-gov"="F_estado"," Self-emp-not-inc"="Autonomo empresa",
                              " Private"="Privada", " Federal-gov"="F_federal",
                              " Local-gov"="F_local", " Self-emp-inc"="Autonomo non empresa",
                              " Without-pay"="Gratis"," Never-worked"="Nunca" ))-> borrar
#recodificar varios a un
borrar%>%mutate(tipo2=recode(tipo," State-gov"="Funcionario"," Self-emp-not-inc"="Autonomo",
                              " Private"="Privada", " Federal-gov"="Funcionario",
                              " Local-gov"="Funcionario", " Self-emp-inc"="Autonomo",
                              " Without-pay"="Outros"," Never-worked"="Outros"  ))->borrar

# Tamén se pode crear unha recodificación a partir de varias variables
# este exemplo non ten sentido polo que non se gardará, pero sirve para ver como sería
borrar %>% mutate(nadaqufacer=case_when(
  ethnic==" White"& nationality==" United-States"~"Branco_USA",
  nationality==" United-States"~"Outro_USA",
  TRUE ~ "Outro"
))%>%sample_n(10)
```

```
##    idade             tipo     nivel_edu valor_edu        estado_civil
## 1     19         F_estado  Some-college        10       Never-married
## 2     29          Privada       HS-grad         9  Married-civ-spouse
## 3     26 Autonomo empresa       HS-grad         9       Never-married
## 4     27          Privada     Assoc-voc        11            Divorced
## 5     38          Privada     Assoc-voc        11            Divorced
## 6     21          Privada       HS-grad         9       Never-married
## 7     38          Privada  Some-college        10            Divorced
## 8     42 Autonomo empresa     Bachelors        13  Married-civ-spouse
## 9     63             <NA>     Bachelors        13             Widowed
## 10    22             <NA>       7th-8th         4       Never-married
##             ocupacion              ethnic  gender hrs_wk    nationality income
## 1      Prof-specialty               White    Male     10  United-States  <=50K
## 2   Machine-op-inspct               White    Male     30  United-States   >50K
## 3      Prof-specialty               Black  Female     40  United-States  <=50K
## 4   Machine-op-inspct               White    Male     40  United-States  <=50K
## 5        Craft-repair               White    Male     40  United-States  <=50K
## 6     Farming-fishing               White    Male     20  United-States  <=50K
## 7     Exec-managerial               White  Female     42  United-States  <=50K
## 8     Farming-fishing               White    Male     65  United-States  <=50K
## 9                <NA>  Amer-Indian-Eskimo  Female     56  United-States  <=50K
## 10               <NA>               White    Male     40  United-States  <=50K
##        tempo       nada            tipo2 nadaqufacer
## 1     escaso Branco_USA         F_estado  Branco_USA
```

```
## 2   esta ben Branco_USA          Privada  Branco_USA
## 3     normal   Afro_USA Autonomo empresa   Outro_USA
## 4     normal Branco_USA          Privada  Branco_USA
## 5     normal Branco_USA          Privada  Branco_USA
## 6      baixo Branco_USA          Privada  Branco_USA
## 7       alto Branco_USA          Privada  Branco_USA
## 8  imposible Branco_USA Autonomo empresa  Branco_USA
## 9   excesivo  Outro_USA             <NA>   Outro_USA
## 10    normal Branco_USA             <NA>  Branco_USA
```

Reordear

Asignarlle unha ordenación a un atributo, para que apareza con ela en táboas e gráficos e non apareza por orde alfabética.

Un truco é converter o atributo en factor, indicándolle a orde que deben levar os niveis.

```
table(borrar$tempo)
```

```
##
##      alto     baixo    escaso  esta ben  excesivo imposible    normal
##      5938      2192       736      2317      2533      1110     17735
```

```
borrar$tempo=factor(borrar$tempo,levels = c("escaso", "baixo","esta ben","normal","alto", "excesivo",
                                            "imposible" ))
table(borrar$tempo)
```

```
##
##    escaso     baixo  esta ben    normal      alto  excesivo imposible
##       736      2192      2317     17735      5938      2533      1110
```

AGREGACIÓN `group_by()`,`summarise()`

Consiste en calcular valores resume a partir de submostras, por exemplo, podería ser o nº medio de horas traballadas por ocupacion. Teríamos un data.frame novo con menos elementos (1 por tipo de ocupacion), e o valor sería a media de horas de cada ocupación.

Necesitanse dúas accións. En primeiro lugar crear un agrupamento con `group_by`, despois facer os cálculos con `summarise()`

```
#exemplo: calcular un dato de hrs_wk medio por ocupacion.
borrar %>% group_by(ocupacion) %>%
  summarise(hrs_wk=mean(hrs_wk,na.rm=TRUE))#na.rm=TRUE é por se hai NAs
```

```
## # A tibble: 15 x 2
##    ocupacion            hrs_wk
##    <chr>                 <dbl>
##  1 " Adm-clerical"        37.6
##  2 " Armed-Forces"        40.7
##  3 " Craft-repair"        42.3
##  4 " Exec-managerial"     45.0
##  5 " Farming-fishing"     47.0
##  6 " Handlers-cleaners"   37.9
##  7 " Machine-op-inspct"   40.8
##  8 " Other-service"       34.7
##  9 " Priv-house-serv"     32.9
## 10 " Prof-specialty"      42.4
## 11 " Protective-serv"     42.9
```

```
## 12 " Sales"                40.8
## 13 " Tech-support"         39.4
## 14 " Transport-moving"     44.7
## 15  <NA>                   31.9
```

```
#Poden facerse máis calculos á vez, e obter unha táboa de parámetros
borrar %>% group_by(ocupacion) %>% summarise(N=n(),media=mean(hrs_wk,na.rm=TRUE),
                                             dt=sd(hrs_wk,na.rm=TRUE),
                                             Minimo=min(hrs_wk,na.rm=TRUE),
                                             C1=quantile(hrs_wk,0.25,na.rm=TRUE),
                                             C3=quantile(hrs_wk,0.75,na.rm=TRUE),
                                             Maximo=max(hrs_wk,na.rm=TRUE))
```

```
## # A tibble: 15 x 8
##    ocupacion                N media    dt Minimo    C1    C3 Maximo
##    <chr>                <int> <dbl> <dbl>  <int> <dbl> <dbl>  <int>
##  1 " Adm-clerical"       3770  37.6  9.59      1    37    40     80
##  2 " Armed-Forces"          9  40.7 14.1       8    40    48     60
##  3 " Craft-repair"       4099  42.3  9.05      1    40    45     99
##  4 " Exec-managerial"    4066  45.0 11.1       1    40    50     99
##  5 " Farming-fishing"     994  47.0 17.3       2    40    60     99
##  6 " Handlers-cleaners"  1370  37.9 10.6       2    36    40     95
##  7 " Machine-op-inspct"  2002  40.8  7.59      1    40    40     96
##  8 " Other-service"      3295  34.7 12.7       1    25    40     99
##  9 " Priv-house-serv"     149  32.9 16.2       4    22    40     99
## 10 " Prof-specialty"     4140  42.4 12.5       1    40    50     99
## 11 " Protective-serv"     649  42.9 12.3       3    40    45     99
## 12 " Sales"              3650  40.8 13.2       2    35    50     99
## 13 " Tech-support"        928  39.4 10.6       3    40    40     99
## 14 " Transport-moving"   1597  44.7 12.7       5    40    50     99
## 15  <NA>                 1843  31.9 14.9       1    20    40     99
```

Tamén se poden facer os cálculos agrupando dúas categorias:

```
#exemplo: calcular un dato de hrs_wk medio por ocupacion e xenero
borrar %>% group_by(ocupacion,gender) %>%
  summarise(hrs_wk=mean(hrs_wk,na.rm=TRUE))->auxiliar#na.rm=TRUE é por se hai NAs
auxiliar
```

```
## # A tibble: 29 x 3
## # Groups:   ocupacion [15]
##    ocupacion            gender    hrs_wk
##    <chr>                <chr>      <dbl>
##  1 " Adm-clerical"      " Female"   36.7
##  2 " Adm-clerical"      " Male"     39.2
##  3 " Armed-Forces"      " Male"     40.7
##  4 " Craft-repair"      " Female"   39.9
##  5 " Craft-repair"      " Male"     42.4
##  6 " Exec-managerial"   " Female"   41.5
##  7 " Exec-managerial"   " Male"     46.4
##  8 " Farming-fishing"   " Female"   37.8
##  9 " Farming-fishing"   " Male"     47.6
## 10 " Handlers-cleaners" " Female"   36.1
## # ... with 19 more rows
```

```
#Poden facerse máis calculos á vez, e obter unha táboa de parámetros
borrar %>% group_by(ocupacion,gender) %>% summarise(N=n(),media=mean(hrs_wk,na.rm=TRUE),
```

```
                                                dt=sd(hrs_wk,na.rm=TRUE),
                                                Minimo=min(hrs_wk,na.rm=TRUE),
                                                C1=quantile(hrs_wk,0.25,na.rm=TRUE),
                                                C3=quantile(hrs_wk,0.75,na.rm=TRUE),
                                                Maximo=max(hrs_wk,na.rm=TRUE))->auxiliar2
auxiliar2
```

```
## # A tibble: 29 x 9
## # Groups:   ocupacion [15]
##    ocupacion            gender        N media    dt Minimo    C1    C3 Maximo
##    <chr>                <chr>     <int> <dbl> <dbl>  <int> <dbl> <dbl>  <int>
##  1 " Adm-clerical"      " Female"  2537  36.7  9.59      1    35    40     80
##  2 " Adm-clerical"      " Male"    1233  39.2  9.37      3    40    40     80
##  3 " Armed-Forces"      " Male"       9  40.7 14.1       8    40    48     60
##  4 " Craft-repair"      " Female"   222  39.9  8.28      8    40    40     80
##  5 " Craft-repair"      " Male"    3877  42.4  9.08      1    40    45     99
##  6 " Exec-managerial"   " Female"  1159  41.5 10.4       3    40    45     99
##  7 " Exec-managerial"   " Male"    2907  46.4 11.1       1    40    50     99
##  8 " Farming-fishing"   " Female"    65  37.8 14.2       8    30    45     82
##  9 " Farming-fishing"   " Male"     929  47.6 17.3       2    40    60     99
## 10 " Handlers-cleaners" " Female"   164  36.1 10.8       6    30    40     84
## # ... with 19 more rows
```

Usando `summarise_if`pódese aplicar o cálculo a todas as variables que verifiquen unha condición.

```
# Un truco interesante, calcular a mediana de todas as variables numericas para cada empresa
borrar %>% group_by(ocupacion) %>%
  summarise_if(is.numeric, median, na.rm = TRUE)
```

```
## # A tibble: 15 x 4
##    ocupacion            idade valor_edu hrs_wk
##    <chr>                <dbl>     <dbl>  <dbl>
##  1 " Adm-clerical"         35        10     40
##  2 " Armed-Forces"         29         9     40
##  3 " Craft-repair"         38         9     40
##  4 " Exec-managerial"      41        12     40
##  5 " Farming-fishing"      39         9     40
##  6 " Handlers-cleaners"    29         9     40
##  7 " Machine-op-inspct"    36         9     40
##  8 " Other-service"        32         9     40
##  9 " Priv-house-serv"      40         9     35
## 10 " Prof-specialty"       40        13     40
## 11 " Protective-serv"      36        10     40
## 12 " Sales"                35        10     40
## 13 " Tech-support"         36        11     40
## 14 " Transport-moving"     39         9     40
## 15  <NA>                   35         9     36
```

PIVOTAR: Pivot ancho

https://dcl-wrangle.stanford.edu/pivot-basic.html

No obxecto auxiliar aparece o nº medio de horas traballadas por ocupación e xénero, pero apreciaríanse mellor as diferencias entre os xéneros se colocásemos home e mullar en columnas diferenciadas.

Esta operación chámase *pivotar*, é o *pivote ancho*, formar variables novas usando as categorías de unha columna:

```
auxiliar %>% pivot_wider(names_from = gender, values_from = hrs_wk) ->auxiliar3
auxiliar3
```

```
## # A tibble: 15 x 3
## # Groups:   ocupacion [15]
##    ocupacion            ` Female` ` Male`
##    <chr>                   <dbl>   <dbl>
##  1 " Adm-clerical"          36.7    39.2
##  2 " Armed-Forces"          NA      40.7
##  3 " Craft-repair"          39.9    42.4
##  4 " Exec-managerial"       41.5    46.4
##  5 " Farming-fishing"       37.8    47.6
##  6 " Handlers-cleaners"     36.1    38.2
##  7 " Machine-op-inspct"     38.9    41.4
##  8 " Other-service"         33.4    36.2
##  9 " Priv-house-serv"       32.5    39.9
## 10 " Prof-specialty"        39.4    44.1
## 11 " Protective-serv"       38.5    43.4
## 12 " Sales"                 34.3    44.2
## 13 " Tech-support"          37.3    40.7
## 14 " Transport-moving"      36.7    45.1
## 15  <NA>                    30.0    33.5
```

Tamén se pode aplicar a separación a varias columnas:

```
auxiliar2 %>% pivot_wider(names_from = gender, values_from = 3:9) ->auxiliar4
auxiliar4
```

```
## # A tibble: 15 x 15
## # Groups:   ocupacion [15]
##    ocupacion     N_ Fe~1 N_ Ma~2 media~3 media~4 dt_ F~5 dt_ M~6 Minim~7 Minim~8
##    <chr>           <int>   <int>   <dbl>   <dbl>   <dbl>   <dbl>   <int>   <int>
##  1 " Adm-cleric~    2537    1233    36.7    39.2    9.59    9.37       1       3
##  2 " Armed-Forc~      NA       9    NA      40.7   NA      14.1       NA       8
##  3 " Craft-repa~     222    3877    39.9    42.4    8.28    9.08       8       1
##  4 " Exec-manag~    1159    2907    41.5    46.4   10.4    11.1        3       1
##  5 " Farming-fi~      65     929    37.8    47.6   14.2    17.3        8       2
##  6 " Handlers-c~     164    1206    36.1    38.2   10.8    10.5        6       2
##  7 " Machine-op~     550    1452    38.9    41.4    8.03    7.30       7       1
##  8 " Other-serv~    1800    1495    33.4    36.2   12.7    12.5        1       3
##  9 " Priv-house~     141       8    32.5    39.9   15.6    24.6        4      25
## 10 " Prof-speci~    1515    2625    39.4    44.1   12.5    12.3        2       1
## 11 " Protective~      76     573    38.5    43.4    8.97   12.6        6       3
## 12 " Sales"         1263    2387    34.3    44.2   12.1    12.5        3       2
## 13 " Tech-suppo~     348     580    37.3    40.7   10.3    10.6        3       8
## 14 " Transport-~      90    1507    36.7    45.1   10.0    12.7       11       5
## 15  <NA>             841    1002    30.0    33.5   13.5    15.8        1       1
## # ... with 6 more variables: `C1_ Female` <dbl>, `C1_ Male` <dbl>,
## #   `C3_ Female` <dbl>, `C3_ Male` <dbl>, `Maximo_ Female` <int>,
## #   `Maximo_ Male` <int>, and abbreviated variable names 1: `N_ Female`,
## #   2: `N_ Male`, 3: `media_ Female`, 4: `media_ Male`, 5: `dt_ Female`,
## #   6: `dt_ Male`, 7: `Minimo_ Female`, 8: `Minimo_ Male`
```

PIVOTAR: Pivot longo

Neste caso combínanse os valores de varias columnas nunha columna nova

```
auxiliar3 %>%
   pivot_longer(
     cols = 2:3, #columnas que se queren combinar
     names_to = "gender", #nome que recibira a columna coas categorias
     values_to = "hrs_wk")  #nome columna dos valores
```

```
## # A tibble: 30 x 3
## # Groups:   ocupacion [15]
##    ocupacion          gender    hrs_wk
##    <chr>              <chr>      <dbl>
##  1 " Adm-clerical"    " Female"   36.7
##  2 " Adm-clerical"    " Male"     39.2
##  3 " Armed-Forces"    " Female"   NA
##  4 " Armed-Forces"    " Male"     40.7
##  5 " Craft-repair"    " Female"   39.9
##  6 " Craft-repair"    " Male"     42.4
##  7 " Exec-managerial" " Female"   41.5
##  8 " Exec-managerial" " Male"     46.4
##  9 " Farming-fishing" " Female"   37.8
## 10 " Farming-fishing" " Male"     47.6
## # ... with 20 more rows
```

Facer pivot longo con varias columnas, pode ser máis complexo. Primeiro un caso suave, meter todos os valores nunha unica columna nova

```
auxiliar4 %>%
   pivot_longer(
     cols = !ocupacion, #columnas todas menos ocupacion
     names_to =  "gender_pmtro", #nome que recibira a columna coas categorias
     values_to = "valor")  #nome columna dos valores
```

```
## # A tibble: 210 x 3
## # Groups:   ocupacion [15]
##    ocupacion       gender_pmtro     valor
##    <chr>           <chr>            <dbl>
##  1 " Adm-clerical" N_ Female      2537
##  2 " Adm-clerical" N_ Male        1233
##  3 " Adm-clerical" media_ Female    36.7
##  4 " Adm-clerical" media_ Male      39.2
##  5 " Adm-clerical" dt_ Female        9.59
##  6 " Adm-clerical" dt_ Male          9.37
##  7 " Adm-clerical" Minimo_ Female    1
##  8 " Adm-clerical" Minimo_ Male      3
##  9 " Adm-clerical" C1_ Female       35
## 10 " Adm-clerical" C1_ Male         40
## # ... with 200 more rows
```

O que si se complica é un pivot longo para que me separe os diferentes parámetros en cadansua columna, e unha columna única para *genre*

Pero pode facerse usando *comodins*, por exemplo *".value"*, que aplicado en `names_to` construe unha columna nova por cada un dos diferentes nomes das columnas implicadas:

> os catro nomes de columna "N_ Female" "N_ Male" "media_ Female" "media_ Male"
>
> producen 3 columnas novas: "N", "media", "genre", usando ".value" de forma apropiada

Outra parte do truco será construir un modelo adecuado para os nomes que estamos separando: `names_pattern = "(.*)_ (.*)"`; esta combinacion propon

duas palabras (os simbolos (.*))

separadas por "_"

En "N_ Female" as duas palabras serían "N" e "Female".

```
auxiliar4 %>%
pivot_longer(!ocupacion,#columnas todas menos ocupacion
names_to = c(".value", "gender"),
 #cada nome de columna produce dous nomes novos:
# N_ Male  produce N (.value) e gender
names_pattern = "(.*)_ (.*)")
```

```
## # A tibble: 30 x 9
## # Groups:   ocupacion [15]
##    ocupacion          gender     N media    dt Minimo    C1    C3 Maximo
##    <chr>              <chr>  <int> <dbl> <dbl>  <int> <dbl> <dbl>  <int>
##  1 " Adm-clerical"    Female  2537  36.7  9.59      1    35    40     80
##  2 " Adm-clerical"    Male    1233  39.2  9.37      3    40    40     80
##  3 " Armed-Forces"    Female    NA  NA   NA        NA    NA    NA     NA
##  4 " Armed-Forces"    Male       9  40.7 14.1       8    40    48     60
##  5 " Craft-repair"    Female   222  39.9  8.28      8    40    40     80
##  6 " Craft-repair"    Male    3877  42.4  9.08      1    40    45     99
##  7 " Exec-managerial" Female  1159  41.5 10.4       3    40    45     99
##  8 " Exec-managerial" Male    2907  46.4 11.1       1    40    50     99
##  9 " Farming-fishing" Female    65  37.8 14.2       8    30    45     82
## 10 " Farming-fishing" Male     929  47.6 17.3       2    40    60     99
## # ... with 20 more rows
```