

TiposDatos

Silvia Pineda

Ejercicio: Tipos de Datos

Lee el fichero y clasifica las variables en cuantitativas y cualitativas.

```
datos<-read.csv("TiposDatos.csv")
```

Nombre: Variable Cualitativa

id: Variable Cualitativa

Peso: Variable Cuantitativa Continua

Altura: Variable Cuantitativa Continua

Fecha_de_nacimiento: Variable Cuantitativa

estado_civil: Variable Cualitativa nominal

Soltero: Variable Cualitativa dicotomica

Mira si todos los tipos de variables están bien declarados y sino declaralos bien.

```
datos
```

	Nombre	id	Peso	Altura	Fecha_de_nacimiento	estado_civil	Soltero
1	Maria	sc001	58.3	165	10/3/85	casado	FALSE
2	Carlos	sc002	70.2	180	17/10/83	soltero	TRUE
3	Ana	sc003	55.4	160	23/4/90	soltero	TRUE
4	Laura	sc004	90.4	165	15/2/87	casado	FALSE
5	Santiago	sc005	83.2	170	1/1/83	viudo	FALSE

```
str(datos)
```

```
'data.frame':  5 obs. of  7 variables:
 $ Nombre      : chr  "Maria" "Carlos" "Ana" "Laura" ...
 $ id          : chr  "sc001" "sc002" "sc003" "sc004" ...
 $ Peso       : num  58.3 70.2 55.4 90.4 83.2
 $ Altura     : int   165 180 160 165 170
 $ Fecha_de_nacimiento: chr  "10/3/85" "17/10/83" "23/4/90" "15/2/87" ...
 $ estado_civil : chr  "casado" "soltero" "soltero" "casado" ...
 $ Soltero     : logi  FALSE TRUE TRUE FALSE FALSE
```

```
datos$estado_civil<-as.factor(datos$estado_civil)
datos$Fecha_de_nacimiento<-as.Date(datos$Fecha_de_nacimiento,format = "%d/%m/%y")

##Mismo resultado con tidyverse
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.3      v readr      2.1.4
v forcats    1.0.0      v stringr    1.5.0
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.2      v tidyr      1.3.0
v purrr      1.0.2
```

```
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()
```

```
x dplyr::lag()     masks stats::lag()
```

```
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
datos <- datos |>
  mutate (
    estado_civil = as.factor(estado_civil), # Convertir a factor
    Fecha_de_nacimiento = dmy(Fecha_de_nacimiento) # Convertir a fecha
  )
```

Warning: There was 1 warning in `mutate()`.

i In argument: `Fecha_de_nacimiento = dmy(Fecha_de_nacimiento)`.

Caused by warning:

! All formats failed to parse. No formats found.

```
datos
```

	Nombre	id	Peso	Altura	Fecha_de_nacimiento	estado_civil	Soltero
1	Maria	sc001	58.3	165	<NA>	casado	FALSE
2	Carlos	sc002	70.2	180	<NA>	soltero	TRUE
3	Ana	sc003	55.4	160	<NA>	soltero	TRUE
4	Laura	sc004	90.4	165	<NA>	casado	FALSE
5	Santiago	sc005	83.2	170	<NA>	viudo	FALSE

Dime como se llaman los individuos que están solteros.

```
datos$Nombre[which(datos$estado_civil=="soltero")]
```

```
[1] "Carlos" "Ana"
```

```
##Mismo resultado con tidyverse  
datos |>  
  filter(estado_civil == "soltero")
```

	Nombre	id	Peso	Altura	Fecha_de_nacimiento	estado_civil	Soltero
1	Carlos	sc002	70.2	180	<NA>	soltero	TRUE
2	Ana	sc003	55.4	160	<NA>	soltero	TRUE

Calcula la edad de los individuos

```
datos$Edad<-floor(as.numeric(Sys.Date()-datos$Fecha_de_nacimiento)/365.25)  
  
datos <- datos |>  
  mutate(  
    Edad = floor(as.numeric(Sys.Date() - Fecha_de_nacimiento) / 365.25)  
  )  
  
datos
```

	Nombre	id	Peso	Altura	Fecha_de_nacimiento	estado_civil	Soltero	Edad
1	Maria	sc001	58.3	165	<NA>	casado	FALSE	NA
2	Carlos	sc002	70.2	180	<NA>	soltero	TRUE	NA
3	Ana	sc003	55.4	160	<NA>	soltero	TRUE	NA
4	Laura	sc004	90.4	165	<NA>	casado	FALSE	NA
5	Santiago	sc005	83.2	170	<NA>	viudo	FALSE	NA