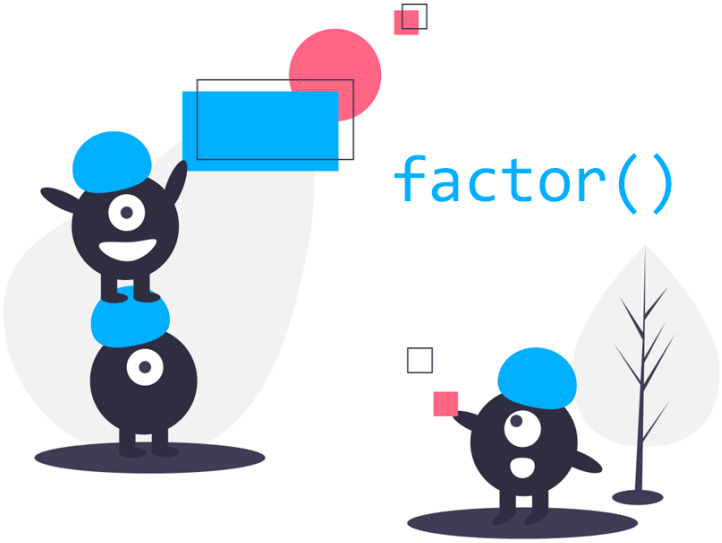


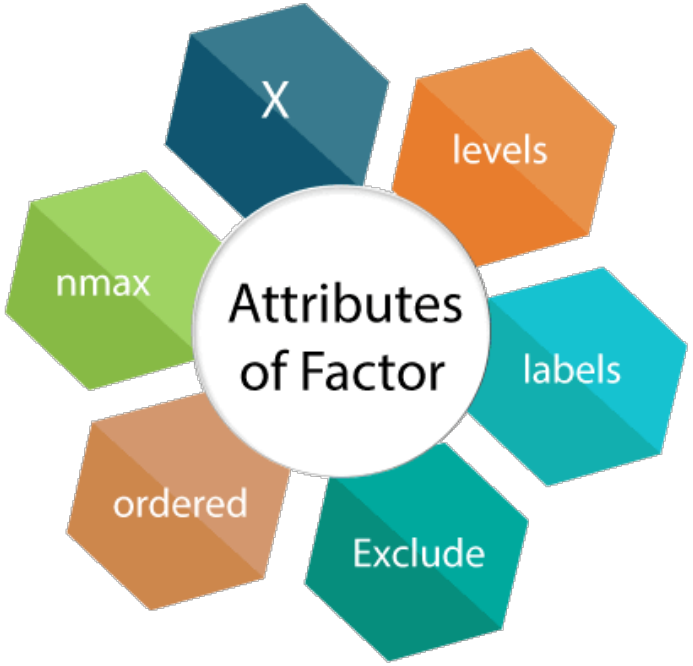
INSP/ESPM
Escuela de Salud Pública de México

R {forcats}, {lubridate}

José Luis Texcalac Sangrador

Procesamiento y visualización de datos espaciales en R

Factores en R



- Se usan para trabajar con variables categóricas, es decir, variables que tienen un conjunto fijo y conocido de valores posibles.
- Son útiles cuando quieres mostrar vectores de caracteres en un orden no alfabético.
- Los valores que un factor puede contener están delimitados por los niveles, es decir, las diferentes categorías presentes en una variable.
- Los niveles pueden estar codificados tanto en valores numéricos como caracteres.

Factores

Vector o variable a categorizar

Niveles que toma la variable

Etiquetas que toma la variable

```
factor(variable,  levels,  labels)
```

```
malla
```

nombre	ojos	pelo	ojos_f	pelo_f
Luisa	azules	1	azules	lacio
Juana	negros	2	negros	rizado
Petra	azules	2	azules	rizado
María	verdes	NA	verdes	NA
Andrea	negros	1	negros	lacio

```
malla <-
  malla %>%
  mutate(ojos_f = factor(ojos,
                         levels = c("azules", "negros",
                                    "verdes", "grises")),
  pelo_f = factor(pelo,
                  levels = c(1, 2),
                  labels = c("lacio", "rizado"))) %>%
  print()
```

```
summary(malla)
```

```
   nombre              ojos                pelo       ojos_f     pelo_f
Length:5           Length:5           Min.   :1.0   azules:2   lacio :2
Class :character   Class :character   1st Qu.:1.0   negros:2   rizado:2
Mode  :character   Mode  :character   Median :1.5   verdes:1   NA's  :1
                                      Mean   :1.5   grises:0
                                      3rd Qu.:2.0
                                      Max.   :2.0
                                      NA's   :1
```

Factores

```
star_wars <-
  starwars %>%
  select(name, height, mass, sex, species) %>%
  print()
```

```
# A tibble: 87 × 5
   name               height  mass sex    species
   <chr>               <int> <dbl> <chr>  <chr>
 1 Luke Skywalker        172    77 male   Human
 2 C-3PO                 167    75 none   Droid
 3 R2-D2                  96    32 none   Droid
 4 Darth Vader           202   136 male   Human
 5 Leia Organa           150    49 female Human
 6 Owen Lars             178   120 male   Human
 7 Beru Whitesun Lars    165    75 female Human
 8 R5-D4                  97    32 none   Droid
 9 Biggs Darklighter     183    84 male   Human
10 Obi-Wan Kenobi        182    77 male   Human
# ℹ 77 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Factores

```
summary(star_wars)
```

```
    name                height           mass
Length:87           Min.   : 66.0   Min.   :  15.00
Class :character    1st Qu.:167.0   1st Qu.:  55.60
Mode  :character    Median :180.0   Median :  79.00
                    Mean   :174.6   Mean   :  97.31
                    3rd Qu.:191.0   3rd Qu.:  84.50
                    Max.   :264.0   Max.   :1358.00
                    NA's   :6       NA's   :28
    sex              species
Length:87           Length:87
Class :character    Class :character
Mode  :character    Mode  :character
```

Vector o variable a categorizar

Etiquetas que toma la variable

```
factor(variable, levels, labels)
```

Niveles que toma la variable

Valores distintos en un vector o variable

```
unique(star_wars$sex)
```

```
> unique(star_wars$sex)
[1] "male"           "none"           "female"
[4] "hermaphroditic" NA
```

```
unique(star_wars$species)
```

```
 [1] "Human"          "Droid"          "Wookiee"
 [4] "Rodian"         "Hutt"           NA
 [7] "Yoda's species" "Trandoshan"     "Mon Calamari"
[10] "Ewok"           "Sullustan"      "Neimodian"
[13] "Gungan"         "Toydarian"      "Dug"
[16] "Zabrak"         "Twi'lek"        "Aleena"
[19] "Vulptereen"     "Xexto"          "Toong"
[22] "Cerean"         "Nautolan"       "Tholothian"
[25] "Iktotchi"       "Quermian"       "Kel Dor"
[28] "Chagrian"       "Geonosian"      "Mirialan"
[31] "Clawdite"       "Besalisk"       "Kaminoan"
[34] "Skakoan"        "Muun"           "Togruta"
[37] "Kaleesh"        "Pau'an"
```

Factores

```
star_wars <-
  star_wars %>%
  mutate(sexo = factor(sex,
                       levels = c("male", "female",
                                  "hermaphroditic", "none")),
         especie = factor(species)) %>%
  print()
```

summary(star_wars)

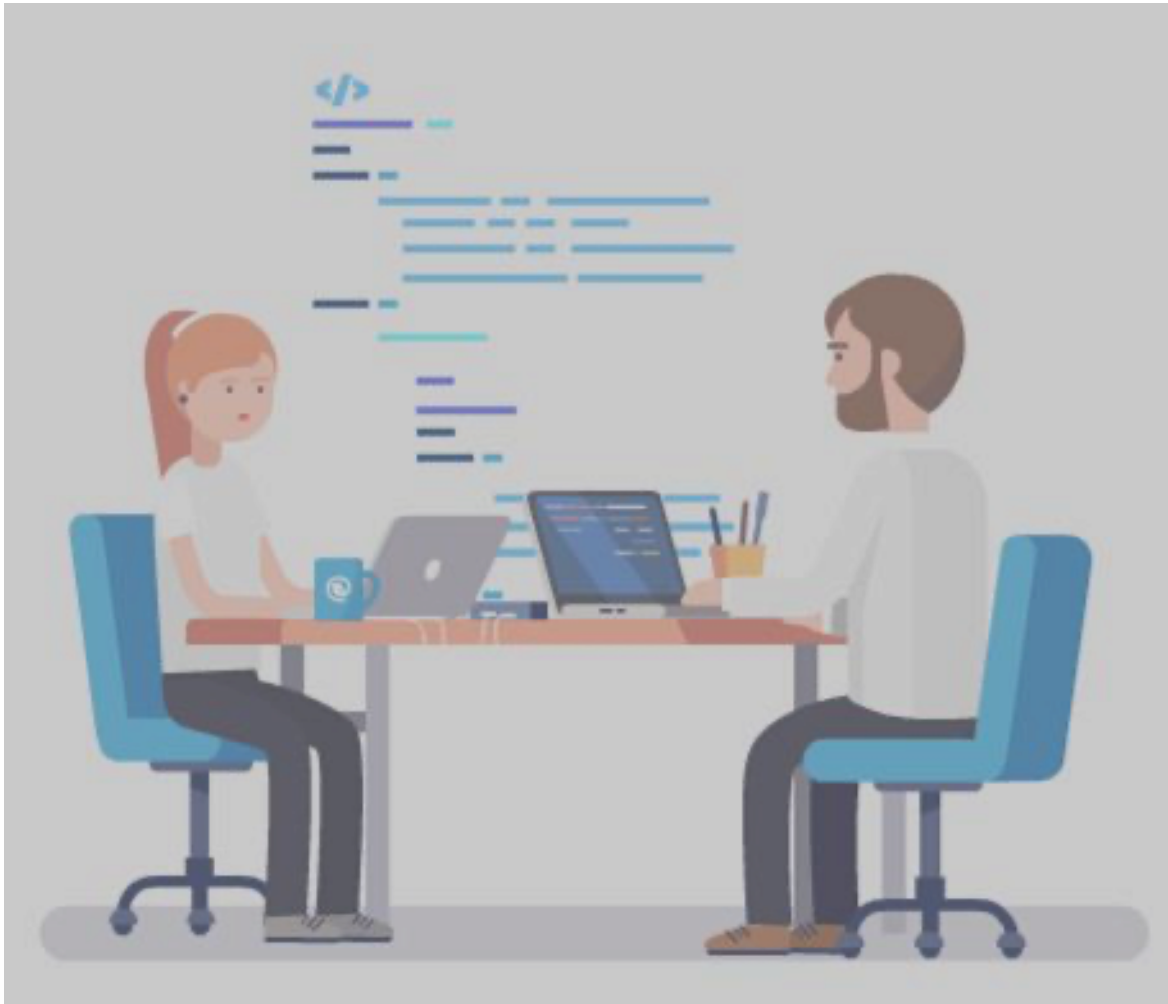
```
     name               height           mass              sex
 Length:87          Min.   : 66.0   Min.   :  15.00   Length:87
 Class :character   1st Qu.:167.0   1st Qu.:  55.60   Class :character
 Mode  :character   Median :180.0   Median :  79.00   Mode  :character
                    Mean   :174.6   Mean   :  97.31
                    3rd Qu.:191.0   3rd Qu.:  84.50
                    Max.   :264.0   Max.   :1358.00
                    NA's   :6       NA's   :28
   species                  sexo         especie
 Length:87          male         :60   Human   :35
 Class :character   female       :16   Droid   : 6
 Mode  :character   hermaphroditic: 1   Gungan  : 3
                    none         : 6   Kaminoan: 2
                    NA's         : 4   Mirialan: 2
                                       (Other) :35
                                       NA's    : 4
```

Trabajando con factores en R {forcats}

- Reordenar niveles en malla de datos y/o gráficos
- Rectificar niveles
- Agregar niveles

Su turno...

- Importe la malla promedios_2021_ps.csv
- Nombre a su objeto como **contam**
- Columnas en formato **snake case**




```r
contam <-
  read_csv("./data/promedios_2021_ps.csv", skip = 8) %>%
  clean_names() %>%
  print()
```

```
Rows: 6692 Columns: 5
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (3): date, id_station, id_parameter
dbl (2): value, unit

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# A tibble: 6,692 × 5
   date       id_station id_parameter value  unit
   <chr>      <chr>      <chr>        <dbl> <dbl>
 1 01/01/2021 ACO        PM10            70     2
 2 01/01/2021 ATI        PM10            44     2
 3 01/01/2021 CUT        PM10            73     2
 4 01/01/2021 FAC        PM10            43     2
 5 01/01/2021 MER        PM10            56     2
 6 01/01/2021 MER        PM2.5           40     2
 7 01/01/2021 NEZ        PM2.5           29     2
 8 01/01/2021 PED        PM10            34     2
 9 01/01/2021 PED        PM2.5           25     2
10 01/01/2021 SAG        PM10            59     2
# ℹ 6,682 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Su turno...

```
summary(contam)
```

```
    date            id_station        id_parameter          value             unit
Length:6692        Length:6692        Length:6692        Min.   :  2.00   Min.   :2
Class :character   Class :character   Class :character   1st Qu.: 17.00   1st Qu.:2
Mode  :character   Mode  :character   Mode  :character   Median : 25.00   Median :2
                                                         Mean   : 30.46   Mean   :2
                                                         3rd Qu.: 39.00   3rd Qu.:2
                                                         Max.   :158.00   Max.   :2
```

- Identifique en la malla contam las variables que requieran codificarse como factor
- Convierta a factor las variables identificadas.
- Nombre a las nuevas columnas como site y pollutant.
- Ejecute el comando summary.
- Elimine las columnas id_station, id_parameter y unit.
- Renombre a la columna value como concentration.


```
contam <-
 contam %>%
 mutate(site = factor(id_station),
        pollutant = factor(id_parameter)) %>%
 print()
```

```
summary(contam)
```

```
    date            id_station        id_parameter          value              unit         site      pollutant
Length:6692        Length:6692        Length:6692        Min.   :  2.00   Min.   :2   BJU    : 482   PM10 :3529
Class :character   Class :character   Class :character   1st Qu.: 17.00   1st Qu.:2   MER    : 478   PM2.5:3163
Mode  :character   Mode  :character   Mode  :character   Median : 25.00   Median :2   PED    : 470
                                                         Mean   : 30.46   Mean   :2   TLA    : 432
                                                         3rd Qu.: 39.00   3rd Qu.:2   SFE    : 386
                                                         Max.   :158.00   Max.   :2   CAM    : 364
                                                                                      (Other):4080
```

```
contam <-
 contam %>%
 select(-c(id_station, id_parameter, unit)) %>%
 rename(concentration = value)
 print()
```

```
summary(contam)
```

```
     date           concentration         site      pollutant
Length:6692        Min.   :  2.00   BJU    : 482   PM10 :3529
Class :character   1st Qu.: 17.00   MER    : 478   PM2.5:3163
Mode  :character   Median : 25.00   PED    : 470
                   Mean   : 30.46   TLA    : 432
                   3rd Qu.: 39.00   SFE    : 386
                   Max.   :158.00   CAM    : 364
                                    (Other):4080
```

Fechas con código base

Columna o vector con fechas
en algún formato consistente
- mes-día-año
- año/mes/dia

Formato de la fecha

```
as.Date(fecha,  format)
```

malla

fecha	valor
12/11/19	12.1
31/01/21	13.4
28/02/17	11.8
03/09/18	14.4
22/11/20	12.7

%d/%m/%y

¡Importante!
El formato se refiere a como está capturado en la columna original, NO al resultado final deseado

```
malla %>%
 mutate(fecha = as.Date(fecha, format = "%d/%m/%y"))
```

fecha	valor
2019-11-12	12.1
2021-01-31	13.4
2017-02-28	11.8
2018-09-03	14.4
2020-11-22	12.7

Fechas con código base

```
malla
```

fecha	valor
12-11-2019	12.1
31-01-2021	13.4
28-02-2017	11.8
03-09-2018	14.4
22-11-2020	12.7

%d-%m-%Y

```
malla %>%
  mutate(fecha = as.Date(fecha, format = "%d-%m-%Y"))
```

fecha	valor
2019-11-12	12.1
2021-01-31	13.4
2017-02-28	11.8
2018-09-03	14.4
2020-11-22	12.7

Code	Value
%d	Day of the month (decimal number)
%m	Month (decimal number)
%b	Month (abbreviated)
%B	Month (full name)
%y	Year (2 digit)
%Y	Year (4 digit)

https://www.stat.berkeley.edu/~s133/dates.html

Fechas usando el paquete {lubridate}

```
malla
```

fecha	valor
12-11-2019	12.1
31-01-2021	13.4
28-02-2017	11.8
03-09-2018	14.4
22-11-2020	12.7

```
malla %>%
  mutate(fecha = dmy(fecha))
```

fecha	valor
2019-11-12	12.1
2021-01-31	13.4
2017-02-28	11.8
2018-09-03	14.4
2020-11-22	12.7

fecha	valor
12-2019-11	12.1
31-2021-01	13.4
28-2017-02	11.8
03-2018-09	14.4
22-2020-11	12.7

```
malla %>%
  mutate(fecha = dym(fecha))
```

```r
today()                          # devuelve la fecha actual

year("2020-11-05")               # devuelve el año de la fecha indicada

month("2020-11-05")              # devuelve el mes de la fecha indicada

day("2020-11-05")                # devuelve el día de la fecha indicada

wday("2020-11-05")               # devuelve el número de la semana de la fecha indicada

yday("2020-11-05")               # devuelve el número de día del año de la fecha indicada
```

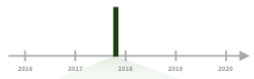
```r
malla %>%
  mutate(mes = month(fecha, label = TRUE))
```

fecha	valor	mes
2019-11-12	12.1	nov
2021-01-31	13.4	ene
2017-02-28	11.8	feb

Dates and times with lubridate : : CHEAT SHEET

https://lubridate.tidyverse.org

Date-times



`2017-11-28 12:00:00`

`2017-11-28 12:00:00`
A **date-time** is a point on the timeline, stored as the number of seconds since 1970-01-01 00:00:00 UTC

dt <- ***as_datetime****(1511870400)*
"2017-11-28 12:00:00 UTC"

`2017-11-28`
A **date** is a day stored as the number of days since 1970-01-01

d <- ***as_date****(17498)*
"2017-11-28"

`12:00:00`
An hms is a **time** stored as the number of seconds since 00:00:00

*t <- hms::****as.hms****(85)*
00:01:25

PARSE DATE-TIMES (Convert strings or numbers to date-times)

1. Identify the order of the year (**y**), month (**m**), day (**d**), hour (**h**), minute (**m**) and second (**s**) elements in your data.
2. Use the function below whose name replicates the order. Each accepts a wide variety of input formats.

`2017-11-28T14:02:00` **ymd_hms**(), **ymd_hm**(), **ymd_h**(). *ymd_hms("2017-11-28T14:02:00")*

`2017-22-12 10:00:00` **ydm_hms**(), **ydm_hm**(), **ydm_h**(). *ydm_hms("2017-22-12 10:00:00")*

`11/28/2017 1:02:03` **mdy_hms**(), **mdy_hm**(), **mdy_h**(). *mdy_hms("11/28/2017 1:02:03")*

`1 Jan 2017 23:59:59` **dmy_hms**(), **dmy_hm**(), **dmy_h**(). *dmy_hms("1 Jan 2017 23:59:59")*

`20170131` **ymd**(), **ydm**(). *ymd(20170131)*

`July 4th, 2000` **mdy**(), **myd**(). *mdy("July 4th, 2000")*

`4th of July '99` **dmy**(), **dym**(). *dmy("4th of July '99")*

`2001: Q3` **yq**() Q for quarter. *yq("2001: Q3")*

`2:01` hms::**hms**() Also lubridate::**hms**(), **hm**() and **ms**(), which return periods.* *hms::hms(sec = 0, min= 1, hours = 2)*

`2017.5` **date_decimal**(decimal, tz = "UTC") *date_decimal(2017.5)*

now(tzone = "") Current time in tz (defaults to system tz). *now()*

today(tzone = "") Current date in a tz (defaults to system tz). *today()*

fast_strptime() Faster strptime. *fast_strptime('9/1/01', '%y/%m/%d')*

parse_date_time() Easier strptime. *parse_date_time("9/1/01", "ymd")*

GET AND SET COMPONENTS

Use an accessor function to get a component. *d ## "2017-11-28"* *day(d) ## 28*

Assign into an accessor function to change a component in place. *day(d) <- 1* *d ## "2017-11-01"*

`2018-01-31 11:59:59` **date**(x) Date component. *date(dt)*

`2018-01-31 11:59:59` **year**(x) Year. *year(dt)*
isoyear(x) The ISO 8601 year.
epiyear(x) Epidemiological year.

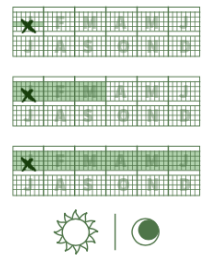
`2018-01-31 11:59:59` **month**(x, label, abbr) Month. *month(dt)*

`2018-01-31 11:59:59` **day**(x) Day of month. *day(dt)*
wday(x,label,abbr) Day of week.
qday(x) Day of quarter.

`2018-01-31 11:59:59` **hour**(x) Hour. *hour(dt)*

`2018-01-31 11:59:59` **minute**(x) Minutes. *minute(dt)*

`2018-01-31 11:59:59` **second**(x) Seconds. *second(dt)*



week(x) Week of the year. *week(dt)*
isoweek() ISO 8601 week.
epiweek() Epidemiological week.

quarter(x, with_year = FALSE) Quarter. *quarter(dt)*

semester(x, with_year = FALSE) Semester. *semester(dt)*

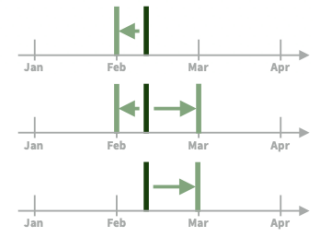
am(x) Is it in the am? *am(dt)*
pm(x) Is it in the pm? *pm(dt)*

dst(x) Is it daylight savings? *dst(d)*

leap_year(x) Is it a leap year? *leap_year(d)*

update(object, ..., simple = FALSE) *update(dt, mday = 2, hour = 1)*

Round Date-times



floor_date(x, unit = "second") Round down to nearest unit. *floor_date(dt, unit = "month")*

round_date(x, unit = "second") Round to nearest unit. *round_date(dt, unit = "month")*

ceiling_date(x, unit = "second", change_on_boundary = NULL) Round up to nearest unit. *ceiling_date(dt, unit = "month")*

rollback(dates, roll_to_first = FALSE, preserve_hms = TRUE) Roll back to last day of previous month. *rollback(dt)*

Stamp Date-times

stamp() Derive a template from an example string and return a new function that will apply the template to date-times. Also **stamp_date**() and **stamp_time**().

1. Derive a template, create a function
   sf <- stamp("Created Sunday, Jan 17, 1999 3:34")
2. Apply the template to dates
   sf(ymd("2010-04-05"))
   ## [1] "Created Monday, Apr 05, 2010 00:00"

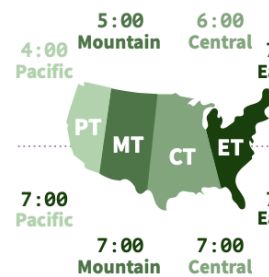
Tip: use a date with day > 12

Time Zones

R recognizes ~600 time zones. Each encodes the time zone, Daylight Savings Time, and historical calendar variations for an area. R assigns *one* time zone per vector.

Use the **UTC** time zone to avoid Daylight Savings.

OlsonNames() Returns a list of valid time zone names. *OlsonNames()*



with_tz(time, tzone = "") Get the **same date-time** in a new time zone (a new clock time). *with_tz(dt, "US/Pacific")*

force_tz(time, tzone = "") Get the **same clock time** in a new time zone (a new date-time). *force_tz(dt, "US/Pacific")*

Su turno...

- Convierta a formato de fecha la columna date
- Genere columnas day, month (con etiqueta), year
- Guarde el objeto con el mismo nombre
- Genere el summary de la malla de datos


```r
contam <-
  contam %>%
  mutate(date = as.Date(date, format = "%d/%m/%Y"),
         day = day(date),
         month = month(date, label = TRUE),
         year = year(date)) %>%
  print()
```

```
# A tibble: 6,692 × 7
   date       concentration site  pollutant   day month  year
   <date>             <dbl> <fct> <fct>     <int> <ord> <dbl>
 1 2021-01-01            70 ACO   PM10          1 ene    2021
 2 2021-01-01            44 ATI   PM10          1 ene    2021
 3 2021-01-01            73 CUT   PM10          1 ene    2021
 4 2021-01-01            43 FAC   PM10          1 ene    2021
 5 2021-01-01            56 MER   PM10          1 ene    2021
 6 2021-01-01            40 MER   PM2.5         1 ene    2021
 7 2021-01-01            29 NEZ   PM2.5         1 ene    2021
 8 2021-01-01            34 PED   PM10          1 ene    2021
 9 2021-01-01            25 PED   PM2.5         1 ene    2021
10 2021-01-01            59 SAG   PM10          1 ene    2021
# … with 6,682 more rows
```

```r
summary(contam)
```

```
     date             concentration        site          pollutant        day              month          year
Min.   :2021-01-01   Min.   :  2.00   BJU     : 482   PM10 :3529   Min.   : 1.00   jul     : 919   Min.   :202
1st Qu.:2021-03-06   1st Qu.: 17.00   MER     : 478   PM2.5:3163   1st Qu.: 8.00   may     : 917   1st Qu.:202
Median :2021-05-07   Median : 25.00   PED     : 470                Median :16.00   jun     : 851   Median :202
Mean   :2021-05-04   Mean   : 30.46   TLA     : 432                Mean   :15.69   mar     : 843   Mean   :202
3rd Qu.:2021-07-03   3rd Qu.: 39.00   SFE     : 386                3rd Qu.:23.00   ene     : 834   3rd Qu.:202
Max.   :2021-08-31   Max.   :158.00   CAM     : 364                Max.   :31.00   ago     : 831   Max.   :202
                                      (Other):4080                                 (Other):1497
```

Malla incial

```
    date            id_station        id_parameter          value              unit
Length:6692        Length:6692        Length:6692        Min.   :  2.00   Min.   :2
Class :character   Class :character   Class :character   1st Qu.: 17.00   1st Qu.:2
Mode  :character   Mode  :character   Mode  :character   Median : 25.00   Median :2
                                                         Mean   : 30.46   Mean   :2
                                                         3rd Qu.: 39.00   3rd Qu.:2
                                                         Max.   :158.00   Max.   :2
```

Malla procesada

```
      date            concentration          site          pollutant          day              month          year
Min.   :2021-01-01   Min.   :  2.00   BJU     : 482   PM10 :3529   Min.   : 1.00   jul     : 919   Min.   :2021
1st Qu.:2021-03-06   1st Qu.: 17.00   MER     : 478   PM2.5:3163   1st Qu.: 8.00   may     : 917   1st Qu.:2021
Median :2021-05-07   Median : 25.00   PED     : 470                Median :16.00   jun     : 851   Median :2021
Mean   :2021-05-04   Mean   : 30.46   TLA     : 432                Mean   :15.69   mar     : 843   Mean   :2021
3rd Qu.:2021-07-03   3rd Qu.: 39.00   SFE     : 386                3rd Qu.:23.00   ene     : 834   3rd Qu.:2021
Max.   :2021-08-31   Max.   :158.00   CAM     : 364                Max.   :31.00   ago     : 831   Max.   :2021
                                      (Other):4080                                 (Other):1497
```