

# Manejo de datos en R

## Usos de tidyverse

Elena Quintero | Curso R AEET | 19 Sept 2022

# Paquetes que usaremos:

```
install.packages(c("tidyverse",
  "here",
  "readxl",
  "tidylog",
  "summarytools",
  "knitr"))
```

# Paquetes incluidos en tidyverse:

```
library(tidyverse)
tidyverse_packages()
```

```
## [1] "broom"          "cli"            "crayon"         "dbplyr"        "dplyr"
## [6] "dplyr"          "forcats"        "googledrive"   "googlesheets4" "ggplot2"
## [11] "haven"          "hms"            "httr"           "jsonlite"      "lubridate"
## [16] "magrittr"       "modelr"         "pillar"         "purrr"         "readr"
## [21] "readxl"         "reprex"         "rlang"          "rstudioapi"   "rvest"
## [26] "stringr"        "tibble"         "tidyverse"      "xml2"          "tidyverse"
```

```
library(readr)      #leer archivos
library(readxl)    #leer archivos excel
library(dplyr)     #manipular datos
library(tidyr)     #ordenar y trasformar datasets
library(stringr)   #manipular caracteres
library(forcats)   #manipular factores
library(lubridate) #manipular fechas
```

# Otros paquetes que utilizaremos:

```
library(here)          #refiere la ruta a la carpeta del proyecto  
library(tidylog)       #informa sobre operaciones dplyr y tidyr  
library(summarytools)   #resume de forma clara y rápida datos numéricos y categóricos  
library(knitr)          #reportar datos en varios formatos
```

# Leer datos

- Library *base*

`read.table`, `read.csv`, `readRDS`

Argumentos útiles: `sep`, `dec`, `comment.char`, `na.strings`, `stringsAsFactors`

# Leer datos

- Library *base*

`read.table`, `read.csv`, `readRDS`

Argumentos útiles: `sep`, `dec`, `comment.char`, `na.strings`, `stringsAsFactors`

- Library *readr*

`read_delim`, `read_csv`, `read_csv2`, `read_table`

Más rápido, produce "tibbles", no convierte caracteres a factors automáticamente, no usa los nombres de fila.

Argumentos útiles: `delim`, `comment`, `na`, `col_types`, `skip_empty_rows`, `guess_max`

# Leer datos

- Library *base*

`read.table`, `read.csv`, `readRDS`

Argumentos útiles: `sep`, `dec`, `comment.char`, `na.strings`, `stringsAsFactors`

- Library *readr*

`read_delim`, `read_csv`, `read_csv2`, `read_table`

Más rápido, produce "tibbles", no convierte caracteres a factors automáticamente, no usa los nombres de fila.

Argumentos útiles: `delim`, `comment`, `na`, `col_types`, `skip_empty_rows`, `guess_max`

- Library *readxl*

`read_excel`, `read_xls`, `read_xlsx`

Argumentos útiles: `sheet`, `col_types`, `skip`

# Leer datos con `readr`

```
library(readr)
```

La función `read_delim()` lee varios tipos de archivo. El argumento `delim`, especifica el separador.

Además tiene funciones específicas como:

- `read_csv()` usa ',' como campo de separación, y '.' para el punto decimal.
- `read_csv2()` usa ';' como campo de separación, y ',' para el punto decimal.

# library(here)

La función `here()` permite hacer referencia siempre al directorio donde se encuentra el proyecto.



Allison Horst Illustration

# library(here)

Ejemplo de uso.

Usando ruta absoluta:

```
data <- read_csv("C:/Usuarios/Elena/Documentos/Proyectos/Proyecto_peces/datos/medida_peces.csv")
```

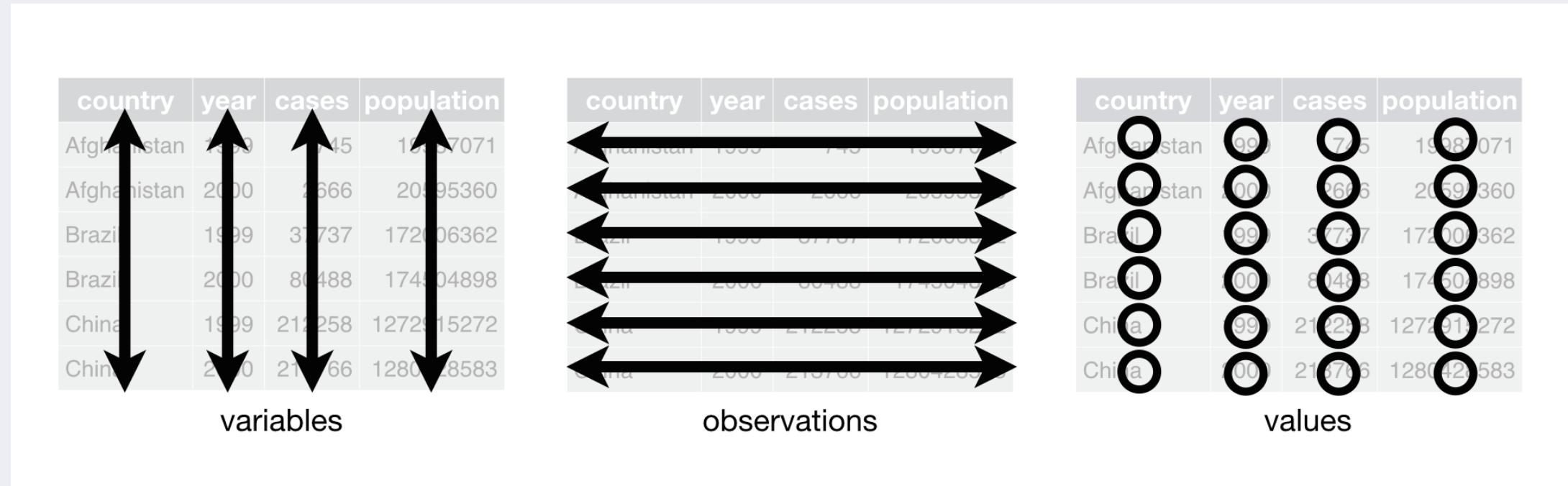
Usando ruta relativa al proyecto:

```
data <- read_csv(here("datos/medida_peces.csv"))
```

# Formato tidy data:

Tres reglas para que los datos estén ordenado:

- Cada variable debe tener su propia columna
- Cada observación debe tener su propia fila
- Cada valor debe tener su propia celda



 THE WORLD BANK | Data Catalog

HOME DATA COLLECTIONS GETTING STARTED FAQS LOGIN

Home / Search Results / Details

# What A Waste Global Database

Metadata last updated on - Jun 28, 2022

What a Waste is a global project to aggregate data on solid waste management from around the world. This database features the statistics collected through the effort, covering nearly all countries and over 330 cities. The metrics included cover all steps from the waste management value chain, including waste generation, composition, collection, and disposal, as well as information on user...

[View More](#)

[Overview](#)

[City level codebook](#) 

[CSV](#) • Last Updated: Mar 4, 2019 • Size: 218.5 KB •  Preview • API Service 

[Country level codebook](#) 

[CSV](#) • Last Updated: Mar 4, 2019 • Size: 1.3 MB •  Preview • API Service 

[Country level dataset](#) 

[CSV](#) • Last Updated: Mar 4, 2019 • Size: 48.0 KB •  Preview • API Service 

 **Data Access and Licensing**

**Classification:** Public

This dataset is classified as **Public** under the Access to Information Classification Policy. Users inside and outside the Bank can access this dataset.

**License:** Creative Commons Attribution 4.0

This dataset is licensed under [Creative Commons Attribution 4.0](#)

**Topics**

- [Environment and Natural Resources](#)
- [Urban Development](#)

 [Collections](#)

# Leer dataset

```
library(readr)
library(here)

waste <- read_csv(here("data/country_level_data.csv"))

## #> Rows: 217 Columns: 28
## #> — Column specification ——————
## #> Delimiter: ","
## #> chr (5): iso3c, region_id, country_name, income_id, where_is_this_data_measured
## #> dbl (23): gdp, population_population_number_of_people, total_msw_total_msw_gener...
## #>
## #> 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.
```

```
#colnames(waste)
dplyr::glimpse(waste)
```

```
## Rows: 217
## Columns: 28
## $ iso3c [3m [38;5;246m<chr> [39m [23m "ABW", "AFG"
## $ region_id [3m [38;5;246m<chr> [39m [23m "LCN", "SAU"
## $ country_name [3m [38;5;246m<chr> [39m [23m "Aruba", "BHR"
## $ income_id [3m [38;5;246m<chr> [39m [23m "HIC", "LIC"
## $ gdp [3m [38;5;246m<dbl> [39m [23m 35563.3125
## $ population_population_number_of_people [3m [38;5;246m<dbl> [39m [23m 103187, 342000
## $ total_msw_total_msw_generated_tons_year [3m [38;5;246m<dbl> [39m [23m 88132.02,
## $ composition_food_organic_waste_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, 51.0
## $ composition_glass_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, 6.0
## $ composition_metal_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, 4.0
## $ composition_other_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, 11.0
## $ composition_paper_cardboard_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, 11.0
## $ composition_plastic_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, 13.0
## $ composition_rubber_leather_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, NA
## $ composition_wood_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, NA
## $ composition_yard_garden_green_waste_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, NA
## $ waste_treatment_anaerobic_digestion_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, NA
## $ waste_treatment_compost_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, NA
## $ waste_treatment_controlled_landfill_percent [3m [38;5;246m<dbl> [39m [23m NA, NA, 14 / 77
```

```
head(waste)
```

```
## # A tibble: 6 × 28
##   iso3c region_id country_name income_id    gdp population_popu... total_msw_total...
##   <chr> <chr>     <chr>       <chr>     <dbl>          <dbl>          <dbl>
## 1 ABW   LCN      Aruba        HIC      35563.         103187        88132.
## 2 AFG   SAS      Afghanistan LIC      2057.         34656032       5628525.
## 3 AGO   SSF      Angola       LMC      8037.         25096150       4213644.
## 4 ALB   ECS      Albania     UMC      13724.        2854191       1087447.
## 5 AND   ECS      Andorra     HIC      43712.        82431          43000
## 6 ARE   MEA      United Arab Em... HIC      67119.        9770529       5617682
## # ... with 21 more variables: composition_food_organic_waste_percent <dbl>,
## #   composition_glass_percent <dbl>, composition_metal_percent <dbl>,
## #   composition_other_percent <dbl>, composition_paper_cardboard_percent <dbl>,
## #   composition_plastic_percent <dbl>, composition_rubber_leather_percent <dbl>,
## #   composition_wood_percent <dbl>,
## #   composition_yard_garden_green_waste_percent <dbl>,
## #   waste_treatment Anaerobic digestion_percent <dbl>, ...
```

```
tail(waste)
```

```
## # A tibble: 6 × 28
##   iso3c region_id country_name income_id     gdp population_population total_msw_total...
##   <chr> <chr>      <chr>       <chr>    <dbl>             <dbl>            <dbl>
## 1 WSM  EAS        Samoa       UMC      6211.           187665          27399.
## 2 XKX  ECS        Kosovo      LMC      9724.           1801800         319000
## 3 YEM  MEA        Yemen, Rep. LIC      8270.           27584212        4836820
## 4 ZAF  SSF        South Africa UMC     12667.           51729344        18457232
## 5 ZMB  SSF        Zambia      LMC      3201.           14264756        2608268
## 6 ZWE  SSF        Zimbabwe   LIC      3191.           12500525        1449752
## # ... with 21 more variables: composition_food_organic_waste_percent <dbl>,
## #   composition_glass_percent <dbl>, composition_metal_percent <dbl>,
## #   composition_other_percent <dbl>, composition_paper_cardboard_percent <dbl>,
## #   composition_plastic_percent <dbl>, composition_rubber_leather_percent <dbl>,
## #   composition_wood_percent <dbl>,
## #   composition_yard_garden_green_waste_percent <dbl>,
## #   waste_treatment Anaerobic_digestion_percent <dbl>, ...
```

# Manejo de datos con tidyverse



Allison Horst Illustration

# Manejo de datos con tidyverse

```
library(tidylog)
```

Da información de las operaciones que se realizan en el dataset.

# El operador 'pipe'

```
library(magrittr)
```

Mecanismo para encadenar funciones.

```
data %>% function(...)
```

Ahora también implementado en R base como |>

```
data |> function(...)
```



# Funciones de **dplyr**

- `arrange()` - Ordenar variable por casos
- `rename()` - Renombrar variables
- `relocate()` - Reordenar variables
- `select()` - Extraer variables

Ayudas de select:

*Selecciona columnas que...*

- `contains()` - *contienen ""*
- `matches()` - *coinciden con ""*
- `starts_with()` - *empiezan por ""*
- `ends_with()` - *acaban por ""*
- `any_of()` - *que estén en el set c("","","")*

Ordernar datos por columnas:

```
waste %>%  
  arrange(population_population_number_of_people)
```

```
## # A tibble: 217 × 28  
##   iso3c region_id country_name income_id    gdp population_popu... total_msw_total...  
##   <chr> <chr>     <chr>       <chr>    <dbl>          <dbl>                <dbl>  
## 1 TUV  EAS        Tuvalu      UMC      3793.         11097            3989.  
## 2 NRU  EAS        Nauru       UMC     11167.         13049            6192.  
## 3 VGB  LCN        British Virgi... HIC      24216.         20645            21099.  
## 4 PLW  EAS        Palau        HIC      18275.         21503            9427.  
## 5 MAF  LCN        St. Martin (F... HIC      30386.         30959            15480.  
## 6 SMR  ECS        San Marino   HIC      58806.         33203            17175.  
## 7 GIB  ECS        Gibraltar   HIC      43712.         33623            16954  
## 8 TCA  LCN        Turks and Cai... HIC      28174.         34900             NA  
## 9 LIE  ECS        Liechtenstein HIC      45727.         36545            32382  
## 10 SXM  LCN       Sint Maarten ... HIC        NA            37685             NA  
## # ... with 207 more rows, and 21 more variables:  
## #   composition_food_organic_waste_percent <dbl>, composition_glass_percent <dbl>,  
## #   composition_metal_percent <dbl>, composition_other_percent <dbl>,  
## #   composition_paper_cardboard_percent <dbl>, composition_plastic_percent <dbl>,  
## #   composition_rubber_leather_percent <dbl>, composition_wood_percent <dbl>,  
## #   composition_yard_garden_green_waste_percent <dbl>,  
## #   waste_treatment Anaerobic_digestion_percent <dbl>, ...
```

## Ordernar datos por columnas

*descendiente:*

```
waste %>%
  arrange(desc(population_population_number_of_people))

## # A tibble: 217 x 28
##   iso3c region_id country_name income_id    gdp population_population_number_of_people total_msw_total...
##   <chr>  <chr>     <chr>       <chr>      <dbl>          <dbl>                    <dbl>
## 1 CHN    EAS        China       UMC        16092.        1400050048            395081376
## 2 IND    SAS        India       LMC        6497.         1352617344            189750000
## 3 USA    NAC        United States HIC        61498.        326687488             265224528
## 4 IDN    EAS        Indonesia   LMC        10531.        261115456             65200000
## 5 BRA    LCN        Brazil      UMC        14596.        208494896             79069584
## 6 PAK    SAS        Pakistan    LMC        4571.         193203472            30760000
## 7 BGD    SAS        Bangladesh  LMC        3196.         155727056            14778497.
## 8 NGA    SSF        Nigeria    LMC        4690.         154402176            27614830.
## 9 RUS    ECS        Russian Feder... UMC        26013.        143201680            60000000
## 10 JPN   EAS        Japan       HIC        41310.        126529104            42720000
## # ... with 207 more rows, and 21 more variables:
## #   composition_food_organic_waste_percent <dbl>, composition_glass_percent <dbl>,
## #   composition_metal_percent <dbl>, composition_other_percent <dbl>,
## #   composition_paper_cardboard_percent <dbl>, composition_plastic_percent <dbl>,
## #   composition_rubber_leather_percent <dbl>, composition_wood_percent <dbl>,
## #   composition_yard_garden_green_waste_percent <dbl>,
## #   waste_treatment_anaerobic_digestion_percent <dbl>, ...
```

Ordernar datos por orden jerárquico:

```
waste %>%
  arrange(region_id, country_name)

## # A tibble: 217 × 28
##   iso3c region_id country_name income_id    gdp population_population... total_msw_total...
##   <chr>  <chr>     <chr>       <chr>    <dbl>          <dbl>                <dbl>
## 1 ASM    EAS      American Samoa UMC      11113.        55599            18989.
## 2 AUS    EAS      Australia      HIC      47784.        23789338           13345000
## 3 BRN    EAS      Brunei Daruss... HIC      60866.        423196            216253.
## 4 KHM    EAS      Cambodia      LMC      3364.         15270790            1089000
## 5 CHN    EAS      China         UMC     16092.        1400050048           395081376
## 6 FJI    EAS      Fiji          UMC     10788.        867086            189390.
## 7 PYF    EAS      French Polynesia HIC     60956.        273528            147000
## 8 GUM    EAS      Guam          HIC     59075.        159973            141500
## 9 HKG    EAS      Hong Kong SAR... HIC     57216.        7305700            5679816.
## 10 IDN   EAS      Indonesia     LMC     10531.        261115456           65200000
## # ... with 207 more rows, and 21 more variables:
## #   composition_food_organic_waste_percent <dbl>, composition_glass_percent <dbl>,
## #   composition_metal_percent <dbl>, composition_other_percent <dbl>,
## #   composition_paper_cardboard_percent <dbl>, composition_plastic_percent <dbl>,
## #   composition_rubber_leather_percent <dbl>, composition_wood_percent <dbl>,
## #   composition_yard_garden_green_waste_percent <dbl>,
## #   waste_treatment Anaerobic_digestion_percent <dbl>, ...
```

Cambiar nombre columnas:

```
waste %>%
  rename(population = population_population_number_of_people,
         total_waste = total_msw_total_msw_generated_tons_year)

## # rename: renamed 2 variables (population, total_waste)

## # A tibble: 217 × 28
##   iso3c region_id country_name      income_id     gdp population total_waste
##   <chr>  <chr>    <chr>        <chr>       <dbl>      <dbl>      <dbl>
## 1 ABW    LCN      Aruba          HIC        35563.     103187     88132.
## 2 AFG    SAS      Afghanistan  LIC        2057.     34656032    5628525.
## 3 AGO    SSF      Angola         LMC        8037.     25096150    4213644.
## 4 ALB    ECS      Albania        UMC        13724.    2854191    1087447.
## 5 AND    ECS      Andorra        HIC        43712.    82431      43000
## 6 ARE    MEA      United Arab Emirates HIC        67119.    9770529    5617682
## 7 ARG    LCN      Argentina      HIC        23550.    42981516    17910550
## 8 ARM    ECS      Armenia        UMC        11020.    2906220    492800
## 9 ASM    EAS      American Samoa UMC        11113.    55599      18989.
## 10 ATG   LCN      Antigua and Barbuda HIC       17966.    96777      30585
## # ... with 207 more rows, and 21 more variables:
## #   composition_food_organic_waste_percent <dbl>, composition_glass_percent <dbl>,
## #   composition_metal_percent <dbl>, composition_other_percent <dbl>,
## #   composition_paper_cardboard_percent <dbl>, composition_plastic_percent <dbl>,
## #   composition_rubber_leather_percent <dbl>, composition_wood_percent <dbl>,
## #   composition_yard_garden_green_waste_percent <dbl>,
## #   waste_treatment Anaerobic_digestion_percent <dbl>, ...
```

Organizar columnas:

```
waste %>%
  relocate(country_name, .before = iso3c)

## # A tibble: 217 × 28
##   country_name  iso3c region_id income_id      gdp population_population... total_msw_total...
##   <chr>        <chr>  <chr>    <chr>     <dbl>           <dbl>                <dbl>
## 1 Aruba        ABW    LCN      HIC      35563.          103187              88132.
## 2 Afghanistan  AFG    SAS      LIC      2057.          34656032             5628525.
## 3 Angola       AGO    SSF      LMC      8037.          25096150             4213644.
## 4 Albania      ALB    ECS      UMC      13724.          2854191              1087447.
## 5 Andorra      AND    ECS      HIC      43712.          82431               43000
## 6 United Arab E... ARE    MEA      HIC      67119.          9770529              5617682
## 7 Argentina    ARG    LCN      HIC      23550.          42981516             17910550
## 8 Armenia      ARM    ECS      UMC      11020.          2906220              492800
## 9 American Samoa ASM   EAS      UMC      11113.          55599               18989.
## 10 Antigua and B... ATG   LCN      HIC      17966.          96777               30585
## # ... with 207 more rows, and 21 more variables:
## #   composition_food_organic_waste_percent <dbl>, composition_glass_percent <dbl>,
## #   composition_metal_percent <dbl>, composition_other_percent <dbl>,
## #   composition_paper_cardboard_percent <dbl>, composition_plastic_percent <dbl>,
## #   composition_rubber_leather_percent <dbl>, composition_wood_percent <dbl>,
## #   composition_yard_garden_green_waste_percent <dbl>,
## #   waste_treatment Anaerobic_digestion_percent <dbl>, ...
```

(Des)seleccionar variables:

```
waste %>%
  select(-region_id)

## select: dropped one variable (region_id)

## # A tibble: 217 × 27
##   iso3c country_name      income_id     gdp population_population... total_msw_total...
##   <chr> <chr>          <chr>    <dbl>           <dbl>                <dbl>
## 1 ABW  Aruba            HIC      35563.          103187               88132.
## 2 AFG  Afghanistan     LIC       2057.          34656032              5628525.
## 3 AGO  Angola           LMC      8037.          25096150              4213644.
## 4 ALB  Albania          UMC      13724.          2854191              1087447.
## 5 AND  Andorra          HIC      43712.          82431                43000
## 6 ARE  United Arab Emirates HIC      67119.          9770529              5617682
## 7 ARG  Argentina        HIC      23550.          42981516              17910550
## 8 ARM  Armenia          UMC      11020.          2906220              492800
## 9 ASM  American Samoa   UMC      11113.          55599                18989.
## 10 ATG  Antigua and Barbuda HIC      17966.          96777                30585
## # ... with 207 more rows, and 21 more variables:
## #   composition_food_organic_waste_percent <dbl>, composition_glass_percent <dbl>,
## #   composition_metal_percent <dbl>, composition_other_percent <dbl>,
## #   composition_paper_cardboard_percent <dbl>, composition_plastic_percent <dbl>,
## #   composition_rubber_leather_percent <dbl>, composition_wood_percent <dbl>,
## #   composition_yard_garden_green_waste_percent <dbl>,
## #   waste_treatment Anaerobic_digestion_percent <dbl>, ...
```

Seleccionar sólo variables de interés:

```
waste_select <- waste %>%
  select(iso3c,
         region_id,
         country = country_name,
         income_id,
         gdp,
         population = population_population_number_of_people,
         total_waste = total_msw_total_msw_generated_tons_year,
         starts_with("composition")) %>%
  arrange(desc(total_waste))
```

```
## select: renamed 3 variables (country, population, total_waste) and dropped 12 variables
```

```
glimpse(waste_select)
```

```
## Rows: 217
## Columns: 16
## $ iso3c
## $ region_id
## $ country
## $ income_id
## $ gdp
## $ population
## $ total_waste
## $ composition_food_organic_waste_percent
## $ composition_glass_percent
## $ composition_metal_percent
## $ composition_other_percent
## $ composition_paper_cardboard_percent
## $ composition_plastic_percent
## $ composition_rubber_leather_percent
## $ composition_wood_percent
## $ composition_yard_garden_green_waste_percent
```

```
<chr> "CHN", "USA", "IND", "BRA", "I...
<chr> "EAS", "NAC", "SAS", "LCN", "E...
<chr> "China", "United States", "Ind...
<chr> "UMC", "HIC", "LMC", "UMC", "L...
<dbl> 16092.301, 61498.371, 6496.808...
<dbl> 1400050048, 326687488, 1352617...
<dbl> 395081376, 265224528, 18975000...
<dbl> 61.20000, 14.90000, NA, 51.400...
<dbl> 2.10000, 4.40000, NA, 2.4000...
<dbl> 1.10000, 9.00000, NA, 2.90000,...
<dbl> 13.10000, 3.20000, NA, 16.700...
<dbl> 9.60000, 26.60000, NA, 13.10...
<dbl> 9.80000, 12.90000, NA, 13.50...
<dbl> 1.30, 9.50, NA, NA, NA, NA, NA...
<dbl> 1.80, 6.20, NA, NA, NA, NA, NA...
<dbl> NA, 13.30, NA, NA, NA, NA, NA, ...
```

# Tablas de resumen de datos:

```
library(summarytools)
dfSummary(waste_select$region_id)

## waste_select$region_id was converted to a data frame

## Data Frame Summary
## waste_select
## Dimensions: 217 x 1
## Duplicates: 210
##
## -----
## No Variable Stats / Values Freqs (% of Valid) Graph Valid Missing
## -----
## 1 region_id 1. EAS 37 (17.1%) III 217 0
## [character] 2. ECS 58 (26.7%) IIIZ (100.0%) (0.0%)
## 3. LCN 42 (19.4%) III
## 4. MEA 21 ( 9.7%) I
## 5. NAC 3 ( 1.4%)
## 6. SAS 8 ( 3.7%)
## 7. SSF 48 (22.1%) IIIZ
## -----
```

```
waste_select %>%  
  select(population, gdp) %>%  
  dfSummary()
```

```
## select: dropped 14 variables (iso3c, region_id, country, income_id, total_waste, ...)  
  
## Data Frame Summary  
## waste_select  
## Dimensions: 217 x 2  
## Duplicates: 0  
##  
## -----  
##   No    Variable      Stats / Values          Freqs (% of Valid)  Graph  Valid  Mis  
##   --  
## 1 population  Mean (sd) : 33643890 (136583825)  217 distinct values : 217  0  
##           [numeric] min < med < max:  
##                      11097 < 5737723 < 1400050048  
##                      IQR (CV) : 20456779 (4.1)  
##  
## 2     gdp       Mean (sd) : 22645.9 (22663.6)  213 distinct values : 216  1  
##     [numeric] min < med < max:  
##                      822.6 < 13465.9 < 117335.6  
##                      IQR (CV) : 31111.8 (1)  
##  
##   -----
```

# Más funciones de `dplyr`

- `distinct()` - Extraer valores únicos
- `recode()` - Recodificar casos de una variable
- `group_by()` - Agrupar datos por casos
- `summarise()` - Resumir datos por casos
- `mutate()` - Crear nuevas variables
- `filter()` - Filtrar datos por casos
- `case_when()` - Filtrar datos por casos

Extraer valores únicos (niveles) de una(s) variable(s):

```
waste_select %>%  
  distinct(income_id)
```

```
## distinct: removed 213 rows (98%) , 4 rows remaining
```

```
## # A tibble: 4 × 1  
##   income_id  
##   <chr>  
## 1 UMC  
## 2 HIC  
## 3 LMC  
## 4 LIC
```

Igual a:

```
base::unique(waste_select$income_id)
```

```
## [1] "UMC" "HIC" "LMC" "LIC"
```

LIC = Low income; LMC = Lower middle income; UMC = Upper middle income; HIC = High income

Recodificar niveles de una variable:

```
waste_select %>%
  distinct(region_id)

## distinct: removed 210 rows (97%), 7 rows remaining

## # A tibble: 7 × 1
##   region_id
##   <chr>
## 1 EAS
## 2 NAC
## 3 SAS
## 4 LCN
## 5 ECS
## 6 SSF
## 7 MEA
```

- LCN: Latin America & Caribbean
- SAS: South Asia
- SSF: Sub-Saharan Africa
- ECS: Europe & Central Asia
- MEA: Middle East & North Africa
- EAS: East Asia & Pacific
- NAC: North America

Recodificar niveles de una variable:

```
waste_regions <- waste_select %>%
  mutate(region_id = recode(region_id,
    "LCN" = "Latin_America",
    "SAS" = "South_Asia",
    "SSF" = "Sub-Saharan_Africa",
    "ECS" = "Europe_Central_Asia",
    "MEA" = "Middle_East_North_Africa",
    "EAS" = "East_Asia_Pacific",
    "NAC" = "North_America"))
```

```
## # A tibble: 7 × 1
##   region_id
##   <chr>
## 1 East_Asia_Pacific
## 2 North_America
## 3 South_Asia
## 4 Latin_America
## 5 Europe_Central_Asia
## 6 Sub-Saharan_Africa
## 7 Middle_East_North_Africa
```

Agrupar datos y resumir:

```
waste_regions %>%  
  group_by(region_id) %>%  
  summarise(total_waste = sum(total_waste, na.rm = TRUE))
```

```
## group_by: one grouping variable (region_id)  
  
## summarise: now 7 rows and 2 columns, ungrouped  
  
## # A tibble: 7 × 2  
##   region_id          total_waste  
##   <chr>                <dbl>  
## 1 East_Asia_Pacific    630827137.  
## 2 Europe_Central_Asia  407170316.  
## 3 Latin_America         224893129.  
## 4 Middle_East_North_Africa 124442193.  
## 5 North_America        290409562  
## 6 South_Asia           245640470.  
## 7 Sub-Saharan_Africa   149000010.
```

## Crear nueva variable - Ej: transformar basura a millones de toneladas

```
waste_regions %>%
  group_by(region_id) %>%
  summarise(total_waste = sum(total_waste, na.rm = TRUE)) %>%
  mutate(waste_mttons = total_waste/1000000)

## group_by: one grouping variable (region_id)

## summarise: now 7 rows and 2 columns, ungrouped

## mutate: new variable 'waste_mttons' (double) with 7 unique values and 0% NA

## # A tibble: 7 × 3
##   region_id      total_waste waste_mttons
##   <chr>            <dbl>        <dbl>
## 1 East_Asia_Pacific 630827137.     631.
## 2 Europe_Central_Asia 407170316.    407.
## 3 Latin_America       224893129.    225.
## 4 Middle_East_North_Africa 124442193.  124.
## 5 North_America        290409562     290.
## 6 South_Asia          245640470.    246.
## 7 Sub-Saharan_Africa  149000010.    149.
```

Filtrar datos:

```
waste_regions %>%  
  filter(region_id == "Latin_America")  
  
## filter: removed 175 rows (81%), 42 rows remaining  
  
## # A tibble: 42 × 16  
##   iso3c region_id country income_id     gdp population total_waste composition_foo...  
##   <chr> <chr>      <chr>    <chr>     <dbl>      <dbl>       <dbl>                <dbl>  
## 1 BRA  Latin_Ame... Brazil   UMC      14596.  208494896  79069584               51.4  
## 2 MEX  Latin_Ame... Mexico   UMC      19332.  125890952  53100000               52.4  
## 3 ARG  Latin_Ame... Argent... HIC      23550.  42981516   17910550               38.7  
## 4 COL  Latin_Ame... Colomb... UMC      12523.  46406648  12150120               59.6  
## 5 VEN  Latin_Ame... Venezu... UMC      14270.  29893080  9779093.                NA  
## 6 PER  Latin_Ame... Peru     UMC      11877.  30973354  8356711.                50.4  
## 7 CHL  Latin_Ame... Chile    HIC      20362.  16829442  6517000               53.3  
## 8 ECU  Latin_Ame... Ecuador  UMC      11896.  16144368  5297211.               58.7  
## 9 PRI  Latin_Ame... Puerto... HIC      34311.  3473181   4170953                13.1  
## 10 DOM Latin_Ame... Domini... UMC     15328.  10528394  4063910                51  
## # ... with 32 more rows, and 8 more variables: composition_glass_percent <dbl>,  
## #   composition_metal_percent <dbl>, composition_other_percent <dbl>,  
## #   composition_paper_cardboard_percent <dbl>, composition_plastic_percent <dbl>,  
## #   composition_rubber_leather_percent <dbl>, composition_wood_percent <dbl>,  
## #   composition_yard_garden_green_waste_percent <dbl>
```

Filtrar datos:

```
waste_regions %>%  
  filter(region_id == "Europe_Central_Asia" & population <= 1000000)  
  
## filter: removed 205 rows (94%), 12 rows remaining  
  
## # A tibble: 12 × 16  
##   iso3c region_id country income_id    gdp population total_waste composition_foo...  
##   <chr> <chr>     <chr>   <chr>    <dbl>      <dbl>        <dbl>                <dbl>  
## 1 LUX Europe_Ce... Luxemb... HIC     1.14e5     619896     490338.                 30  
## 2 MNE Europe_Ce... Monten... UMC     2.08e4     622227     329780.                 33.8  
## 3 ISL Europe_Ce... Iceland HIC     5.53e4     343400     225270.                 10  
## 4 CHI Europe_Ce... Channe... HIC     4.67e4     164541     178933                  NA  
## 5 FRO Europe_Ce... Faeroe... HIC     4.44e4     48842      61000                  NA  
## 6 IMN Europe_Ce... Isle o... HIC     4.42e4     80759      50551                  NA  
## 7 GRL Europe_Ce... Greenl... HIC     4.39e4     56905      50000                 42.8  
## 8 MCO Europe_Ce... Monaco  HIC     4.37e4     37783      46000                  NA  
## 9 AND Europe_Ce... Andorra HIC     4.37e4     82431      43000                 31.2  
## 10 LIE Europe_Ce... Liecht... HIC     4.57e4     36545      32382                 37.6  
## 11 SMR Europe_Ce... San Ma... HIC     5.88e4     33203     17175.                  5.35  
## 12 GIB Europe_Ce... Gibralt... HIC     4.37e4     33623      16954                 24.6  
## # ... with 8 more variables: composition_glass_percent <dbl>,  
## #   composition_metal_percent <dbl>, composition_other_percent <dbl>,  
## #   composition_paper_cardboard_percent <dbl>, composition_plastic_percent <dbl>,  
## #   composition_rubber_leather_percent <dbl>, composition_wood_percent <dbl>,  
## #   composition_yard_garden_green_waste_percent <dbl>
```

Crear nuevo factor:

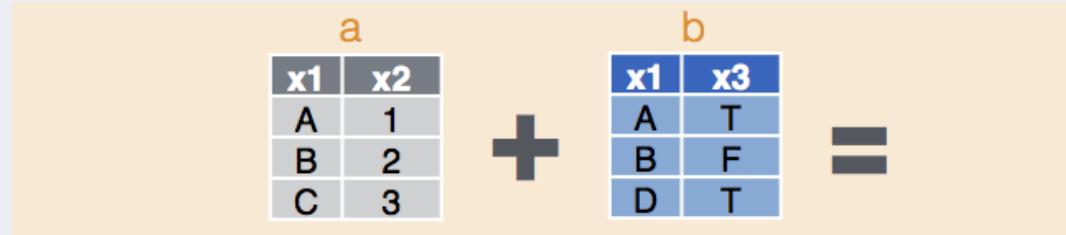
```
waste_regions %>%
  mutate(pop_size = case_when(
    population >= 1000000 ~ "big",
    population < 1000000 & population > 500000 ~ "medium",
    population <= 500000 ~ "small")) %>%
  relocate(pop_size, .before = population)

## # ... with 207 more rows, and 9 more variables:
## #   composition_food_organic_waste_percent <dbl>, composition_glass_percent <dbl>,
```

# Funciones de **dplyr**

- `arrange()` - Ordenar variable por casos
- `rename()` - Renombrar variables
- `relocate()` - Reordenar variables
- `select()` - Extraer variables
- `distinct()` - Extraer valores únicos
- `recode()` - Recodificar casos de una variable
- `group_by()` - Agrupar datos por casos
- `summarise()` - Resumir datos por casos
- `mutate()` - Crear nuevas variables
- `filter()` - Filtrar datos por casos
- `case_when()` - Filtrar datos por casos

# Combinar bases de datos con **join**:



Mutating Joins

**dplyr::left\_join(a, b, by = "x1")**  
Join matching rows from b to a.

x1	x2	x3
A	1	T
B	2	F
C	3	NA

**dplyr::right\_join(a, b, by = "x1")**  
Join matching rows from a to b.

x1	x3	x2
A	T	1
B	F	2
D	T	NA

**dplyr::inner\_join(a, b, by = "x1")**  
Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F

**dplyr::full\_join(a, b, by = "x1")**  
Join data. Retain all values, all rows.

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T

# Combinar bases de datos con **join**:

Vamos a darle a nuestro dataset información sobre el continente donde se encuentra cada país y su localización.

Leer nuevo dataset con información sobre el continente:

```
world_data <- read_csv2(here("data/world_data.csv"))

## i Using ',',',' as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.

## Rows: 241 Columns: 16
## — Column specification ——————
## Delimiter: ";"
## chr (12): name, name_long, sovereign, type, abbrev, continent, formal_en, gdp_m...
## dbl (4): pop_est, pop_year, lastcensus, gdp_year
##
## 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.
```

```
glimpse(world_data)
```

```
## Rows: 241
## Columns: 16
## $ name      <chr> "Aruba", "Afghanistan", "Angola", "Anguilla", "Albania", "Aland...
## $ name_long  <chr> "Aruba", "Afghanistan", "Angola", "Anguilla", "Albania", "Aland...
## $ sovereignt <chr> "Netherlands", "Afghanistan", "Angola", "United Kingdom", "Alba...
## $ type       <chr> "Country", "Sovereign country", "Sovereign country", "Dependenc...
## $ abbrev     <chr> "Aruba", "Afg.", "Ang.", "Ang.", "Alb.", "Aland", "And.", "U.A...
## $ continent  <chr> "North America", "Asia", "Africa", "North America", "Europe", ...
## $ formal_en   <chr> "Aruba", "Islamic State of Afghanistan", "People's Republic of ...
## $ pop_est    <dbl> 103065, 28400000, 12799293, 14436, 3639453, 27153, 83888, 47984...
## $ gdp_md_est <chr> "2258", "22270", "110300", "108.9", "21810", "1563", "3660", "1...
## $ pop_year   <dbl> NA, ...
## $ lastcensus <dbl> 2010, 1979, 1970, NA, 2001, NA, 1989, 2010, 2010, 2001, 2010, N...
## $ gdp_year   <dbl> NA, ...
## $ economy    <chr> "6. Developing region", "7. Least developed region", "7. Least ...
## $ income_grp  <chr> "2. High income: nonOECD", "5. Low income", "3. Upper middle in...
## $ iso_a3     <chr> "ABW", "AFG", "AGO", "AIA", "ALB", "ALA", "AND", "ARE", "ARG", ...
## $ region_un  <chr> "Americas", "Asia", "Africa", "Americas", "Europe", "Europe", "...
```

## Seleccionar variables de interés:

```
continent <- world_data %>%
  select(iso_a3,
         country_name = name_long,
         continent)

## select: renamed one variable (country_name) and dropped 13 variables

glimpse(continent)

## #> #> Rows: 241
## #> #> Columns: 3
## #> #> $ iso_a3      <chr> "ABW", "AFG", "AGO", "AIA", "ALB", "ALA", "AND", "ARE", "ARG"...
## #> #> $ country_name <chr> "Aruba", "Afghanistan", "Angola", "Anguilla", "Albania", "Ala...
## #> #> $ continent    <chr> "North America", "Asia", "Africa", "North America", "Europe",...
```

# Combinar datasets:

Usando `full_join()`

```
waste_world <- waste_regions %>%
  rename(iso_a3 = iso3c) %>%
  full_join(continent, by = "iso_a3")

## rename: renamed one variable (iso_a3)

## full_join: added 2 columns (country_name, continent)

##           > rows only in x      5
##           > rows only in y     29
##           > matched rows    212
##           >                   =====
##           > rows total       246
```

# Combinar datasets:

Usando `left_join()`

```
waste_world <- waste_regions %>%
  rename(iso_a3 = iso3c) %>%
  left_join(continent, by = "iso_a3")

## rename: renamed one variable (iso_a3)

## left_join: added 2 columns (country_name, continent)

##           > rows only in x      5
##           > rows only in y  ( 29)
##           > matched rows     212
##           >                   =====
##           > rows total       217
```

¿Qué países se han quedado sin identificar?

```
waste_world %>%
  filter(is.na(continent)) %>%
  pull(country, iso_a3)
```

```
## filter: removed 212 rows (98%), 5 rows remaining
```

```
##          TWN           XKX           CHI           GIB
##          NA   "Kosovo"  "Channel Islands"  "Gibraltar"
##          TUV
##  "Tuvalu"
```

Buscar los países que faltan en el dataset de continente:

```
continent %>%
  filter(country_name %in%
        c("Channel Islands", "Gibraltar", "Tuvalu", "Kosovo", "Taiwan"))
```

```
## filter: removed 239 rows (99%), 2 rows remaining
```

```
## # A tibble: 2 × 3
##   iso_a3 country_name continent
##   <chr>   <chr>       <chr>
## 1 <NA>    Kosovo      Europe
## 2 <NA>    Taiwan      Asia
```

# library(stringr)

## stringr::str\_squish()

remove leading, trailing, &  
repeated interior whitespace  
from strings.



@allison\_horst

# Utilidades de `library(stringr)`

- `str_length()` - Longitud de una cadena
- `str_detect()` - Detecta un determinado patrón
- `str_extract()` - Extrae un determinado patrón
- `str_c()` - Encadena caracteres (similar a `paste0()`)
- `str_sub()` - Extrae sub-caracteres de una cadena
- `str_replace()` - Reemplaza carácter(es) por otro(s)
- `str_to_lower()`, `str_to_upper()`, `str_to_title()` - transformar en mayúsculas o minúsculas

# Usando `library(stringr)`

Alternativa para buscar los países que faltan en el dataset de continente:

```
continent %>%
  filter(str_detect(country_name, "Kosovo|Gibraltar|Tuvalu|Channel Islands|Taiwan"))
```

```
## filter: removed 239 rows (99%), 2 rows remaining
```

```
## # A tibble: 2 × 3
##   iso_a3 country_name continent
##   <chr>   <chr>       <chr>
## 1 <NA>    Kosovo      Europe
## 2 <NA>    Taiwan      Asia
```

Otro ejemplo - buscar Islas:

```
continent %>%
  filter(str_detect(country_name, "Island"))

## filter: removed 224 rows (93%), 17 rows remaining

## # A tibble: 17 × 3
##   iso_a3 country_name      continent
##   <chr>   <chr>        <chr>
## 1 ALA     Aland Islands    Europe
## 2 <NA>   Ashmore and Cartier Islands Oceania
## 3 COK     Cook Islands     Oceania
## 4 CYM     Cayman Islands   North America
## 5 <NA>   Falkland Islands <NA>
## 6 FRO     Faeroe Islands   Europe
## 7 HMD     Heard I. and McDonald Islands Seven seas (open ocean)
## 8 MHL     Marshall Islands Oceania
## 9 MNP     Northern Mariana Islands Oceania
## 10 NFK    Norfolk Island   Oceania
## 11 PCN    Pitcairn Islands Oceania
## 12 SGS    South Georgia and South Sandwich Islands Seven seas (open ocean)
## 13 SLB    Solomon Islands  Oceania
## 14 TCA    Turks and Caicos Islands North America
## 15 VGB    British Virgin Islands North America
## 16 VIR    United States Virgin Islands North America
```

Corregir un dato:

```
continent_corrected <- continent %>%
  mutate(iso_a3 = ifelse(country_name == "Kosovo", "XKX", iso_a3)) %>%
  mutate(iso_a3 = ifelse(country_name == "Taiwan", "TWN", iso_a3))
```

```
## mutate: changed one value (<1%) of 'iso_a3' (1 fewer NA)
## mutate: changed one value (<1%) of 'iso_a3' (1 fewer NA)
```

Volver a combinar dataset:

```
waste_world <- waste_regions %>%
  rename(iso_a3 = iso3c) %>%
  left_join(continent_corrected, by = "iso_a3")
```

```
## rename: renamed one variable (iso_a3)

## left_join: added 2 columns (country_name, continent)
```

```
##           > rows only in x      3
```

```
##           > rows only in y  ( 27)
```

```
##           > matched rows     214
```

```
##           >                      =====
```

```
##           > rows total        217
```

```
glimpse(waste_world)
```

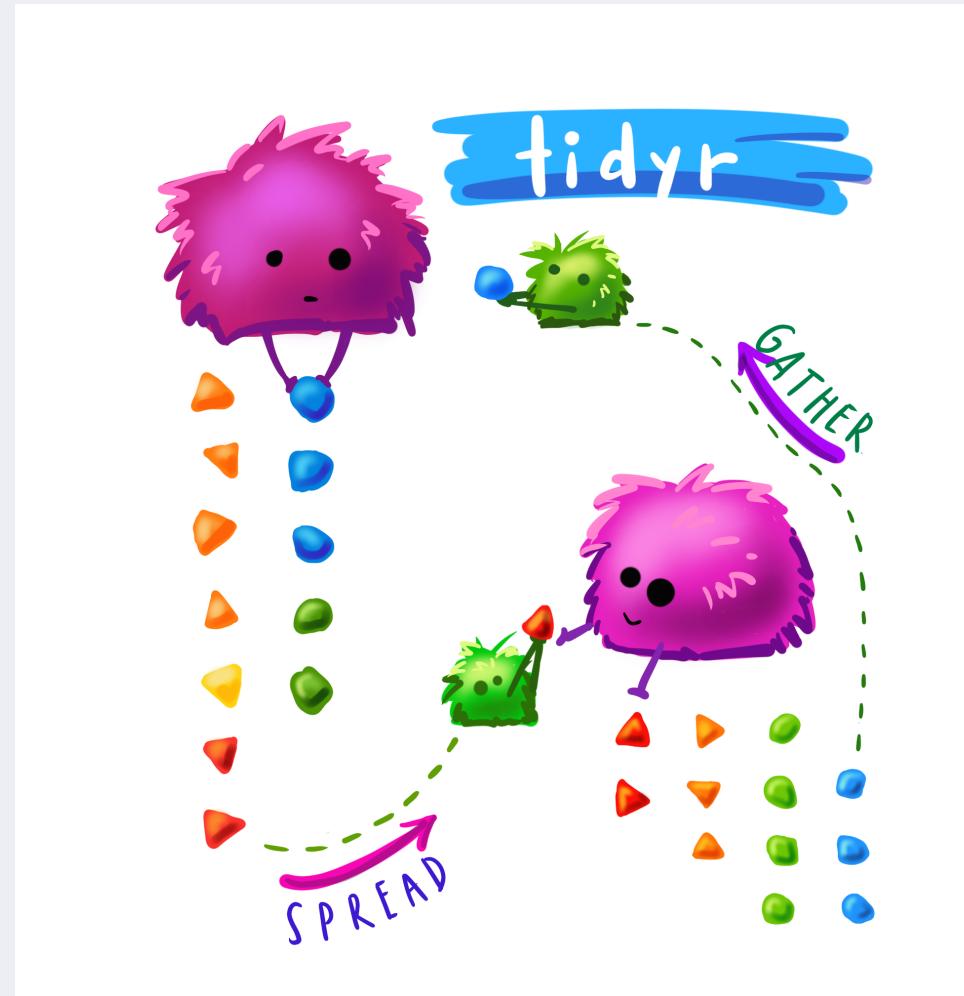
```
## Rows: 217
## Columns: 18
## $ iso_a3
## $ region_id
## $ country
## $ income_id
## $ gdp
## $ population
## $ total_waste
## $ composition_food_organic_waste_percent
## $ composition_glass_percent
## $ composition_metal_percent
## $ composition_other_percent
## $ composition_paper_cardboard_percent
## $ composition_plastic_percent
## $ composition_rubber_leather_percent
## $ composition_wood_percent
## $ composition_yard_garden_green_waste_percent
## $ country_name
## $ continent
```

```
<chr> "CHN", "USA", "IND", "BRA", "I...
<chr> "East_Asia_Pacific", "North_Am...
<chr> "China", "United States", "Ind...
<chr> "UMC", "HIC", "LMC", "UMC", "L...
<dbl> 16092.301, 61498.371, 6496.808...
<dbl> 1400050048, 326687488, 1352617...
<dbl> 395081376, 265224528, 18975000...
<dbl> 61.20000, 14.90000, NA, 51.400...
<dbl> 2.100000, 4.400000, NA, 2.4000...
<dbl> 1.10000, 9.00000, NA, 2.90000,...
<dbl> 13.10000, 3.20000, NA, 16.7000...
<dbl> 9.600000, 26.600000, NA, 13.10...
<dbl> 9.800000, 12.900000, NA, 13.50...
<dbl> 1.30, 9.50, NA, NA, NA, NA, NA...
<dbl> 1.80, 6.20, NA, NA, NA, NA, NA...
<dbl> NA, 13.30, NA, NA, NA, NA, NA, ...
<chr> "China", "United States", "Ind...
<chr> "Asia", "North America", "Asia..."
```

Guardar dataset para el próximo día:

```
write_csv(waste_world, here("data/waste_world.csv"))
```

# library(tidyr)



Allison Horst Illustration

# Reestructurar dataset con `library(tidyr)`

- `pivot_wider()` o `spread()`
- `pivot_longer()` o `gather()`

		wide		long		
		x	y	id	key	val
1	a	c	e	1	x	a
2	b	d	f	2	x	b
				1	y	c
				2	y	d
				1	z	e
				2	z	f

# Reestructurar el dataset con **library(tidyr)**

```
glimpse(waste_world)
```

```
## Rows: 217
## Columns: 18
## $ iso_a3
## $ region_id
## $ country
## $ income_id
## $ gdp
## $ population
## $ total_waste
## $ composition_food_organic_waste_percent
## $ composition_glass_percent
## $ composition_metal_percent
## $ composition_other_percent
## $ composition_paper_cardboard_percent
## $ composition_plastic_percent
## $ composition_rubber_leather_percent
## $ composition_wood_percent
## $ composition_yard_garden_green_waste_percent
## $ country_name
## $ continent
```

```
<chr> "CHN", "USA", "IND", "BRA", "I...
<chr> "East_Asia_Pacific", "North_Am...
<chr> "China", "United States", "Ind...
<chr> "UMC", "HIC", "LMC", "UMC", "L...
<dbl> 16092.301, 61498.371, 6496.808...
<dbl> 1400050048, 326687488, 1352617...
<dbl> 395081376, 265224528, 18975000...
<dbl> 61.20000, 14.90000, NA, 51.400...
<dbl> 2.100000, 4.400000, NA, 2.4000...
<dbl> 1.10000, 9.00000, NA, 2.90000...
<dbl> 13.10000, 3.20000, NA, 16.7000...
<dbl> 9.600000, 26.600000, NA, 13.10...
<dbl> 9.800000, 12.900000, NA, 13.50...
<dbl> 1.30, 9.50, NA, NA, NA, NA, NA...
<dbl> 1.80, 6.20, NA, NA, NA, NA, NA...
<dbl> NA, 13.30, NA, NA, NA, NA, NA, ...
<chr> "China", "United States", "Ind...
<chr> "Asia", "North America", "Asia..."
```

# Reestructurar el dataset con `library(tidyr)`

```
composition <- waste_world %>%
  pivot_longer(cols = starts_with("composition"), names_to = "composition", values_to = "percent")
```

```
composition %>%
  select(country, composition, percent)

## select: dropped 8 variables (iso_a3, region_id, income_id, gdp, population, ...)

## # A tibble: 1,953 × 3
##   country      composition      percent
##   <chr>        <chr>            <dbl>
## 1 China        composition_food_organic_waste_percent 61.2
## 2 China        composition_glass_percent             2.1
## 3 China        composition_metal_percent            1.1
## 4 China        composition_other_percent           13.1
## 5 China        composition_paper_cardboard_percent 9.6
## 6 China        composition_plastic_percent          9.8
## 7 China        composition_rubber_leather_percent 1.3
## 8 China        composition_wood_percent            1.8
## 9 China        composition_yard_garden_green_waste_percent NA
## 10 United States composition_food_organic_waste_percent 14.9
## # ... with 1,943 more rows
```

## Simplificar variables con `library(stringr)`:

```
composition_fix <- composition %>%  
  mutate(composition=str_remove(composition, "composition_")) %>%  
  mutate(composition=str_remove(composition, "_percent"))
```

```
## mutate: changed 1,953 values (100%) of 'composition' (0 new NA)
## mutate: changed 1,953 values (100%) of 'composition' (0 new NA)
```

`distinct(composition_fix, composition)`

```
## distinct: removed 1,944 rows (>99%), 9 rows remaining
```

```
## # A tibble: 9 × 1
##   composition
##   <chr>
## 1 food_organic_waste
## 2 glass
## 3 metal
## 4 other
## 5 paper_cardboard
## 6 plastic
## 7 rubber_leather
## 8 wood
## 9 yard_garden_green
```

```
glimpse(composition_fix)
```

```
## Rows: 1,953
## Columns: 11
## $ iso_a3      <chr> "CHN", "CHN", "CHN", "CHN", "CHN", "CHN", "CHN", "CHN"...
## $ region_id   <chr> "East_Asia_Pacific", "East_Asia_Pacific", "East_Asia_Pacific"...
## $ country     <chr> "China", "China", "China", "China", "China", "China", "China"...
## $ income_id   <chr> "UMC", "UMC", "UMC", "UMC", "UMC", "UMC", "UMC", "UMC"...
## $ gdp         <dbl> 16092.301, 16092.301, 16092.301, 16092.301, 16092.301, 16092...
## $ population  <dbl> 1400050048, 1400050048, 1400050048, 1400050048, 1400050048, 1...
## $ total_waste <dbl> 395081376, 395081376, 395081376, 395081376, 395081376, 395081...
## $ country_name <chr> "China", "China", "China", "China", "China", "China", "China"...
## $ continent    <chr> "Asia", "Asia", "Asia", "Asia", "Asia", "Asia", "Asia", "Asia"...
## $ composition  <chr> "food_organic_waste", "glass", "metal", "other", "paper_cardb...
## $ percent      <dbl> 61.2, 2.1, 1.1, 13.1, 9.6, 9.8, 1.3, 1.8, NA, 14.9, 4.4, 9.0,...
```

## Comprobar datos: asegurar que los porcentajes suman 100%

```
composition %>%  
  group_by(country) %>%  
  summarise(per_sum = sum(percent, na.rm = TRUE))
```

```
## group_by: one grouping variable (country)  
  
## summarise: now 217 rows and 2 columns, ungrouped  
  
## # A tibble: 217 × 2  
##   country      per_sum  
##   <chr>        <dbl>  
## 1 Afghanistan     0  
## 2 Albania       100.  
## 3 Algeria        100  
## 4 American Samoa 100  
## 5 Andorra         100  
## 6 Angola          99.8  
## 7 Antigua and Barbuda 100  
## 8 Argentina       100.  
## 9 Armenia         100  
## 10 Aruba          0  
## # ... with 207 more rows
```

## Comprobar datos: Lista de países cuyos porcentajes no suman 100%

```
composition %>%
  group_by(country) %>%
  summarise(per_sum = sum(percent, na.rm = TRUE)) %>%
  filter(per_sum < 99.9 | per_sum > 100.1) %>% # arrange(per_sum) %>%
  pull(per_sum, country)
```

##	Afghanistan	Angola	Aruba
##	0.00	99.80	0.00
##	Barbados	Belgium	Botswana
##	100.10	99.86	100.10
##	Cabo Verde	Canada	Central African Republic
##	0.00	101.00	0.00
##	Channel Islands	Congo, Dem. Rep.	Congo, Rep.
##	0.00	0.00	0.00
##	Côte d'Ivoire	Curacao	Djibouti
##	0.00	0.00	0.00
##	Equatorial Guinea	Eritrea	Eswatini
##	0.00	0.00	0.00
##	Faeroe Islands	Gabon	Gibraltar
##	0.00	0.00	97.25
##	Guinea-Bissau	India	Indonesia
##	0.00	0.00	99.90
##	Iran, Islamic Rep.	Iraq	Isle of Man
##	100.30	99.74	0.00
##	Kazakhstan	Korea, Rep.	Kuwait
##	97.50	100.40	101.00

Crear dataset solo para los países con información completa:

```
composition_complete <- composition %>%
  group_by(country) %>%
  mutate(per_sum = sum(percent, na.rm = TRUE)) %>%
  filter(per_sum >= 99.9) %>%
  filter(per_sum <= 100.1)

## group_by: one grouping variable (country)

## mutate (grouped): new variable 'per_sum' (double) with 42 unique values and 0% NA

## filter (grouped): removed 477 rows (24%) , 1,476 rows remaining

## filter (grouped): removed 135 rows (9%) , 1,341 rows remaining
```

Atención, en este caso para filtrar el dataset usamos `mutate()` y no `summarise()`, ya que queremos toda la información desagregada.

```
glimpse(composition_complete)
```

```
## Rows: 1,341
## Columns: 12
## Groups: country [149]
## $ iso_a3      <chr> "CHN", "CHN", "CHN", "CHN", "CHN", "CHN", "CHN", "CHN"...
## $ region_id   <chr> "East_Asia_Pacific", "East_Asia_Pacific", "East_Asia_Pacific"...
## $ country     <chr> "China", "China", "China", "China", "China", "China"...
## $ income_id   <chr> "UMC", "UMC", "UMC", "UMC", "UMC", "UMC", "UMC", "UMC"...
## $ gdp         <dbl> 16092.30, 16092.30, 16092.30, 16092.30, 16092.30, 16092.30, 1...
## $ population  <dbl> 1400050048, 1400050048, 1400050048, 1400050048, 1400050048, 1...
## $ total_waste <dbl> 395081376, 395081376, 395081376, 395081376, 395081376, 395081...
## $ country_name <chr> "China", "China", "China", "China", "China", "China"...
## $ continent    <chr> "Asia", "Asia", "Asia", "Asia", "Asia", "Asia", "Asia", "Asia"...
## $ composition  <chr> "composition_food_organic_waste_percent", "composition_glass_...
## $ percent      <dbl> 61.20, 2.10, 1.10, 13.10, 9.60, 9.80, 1.30, 1.80, NA, 14.90, ...
## $ per_sum      <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 1...
```

# Recursos

- Tidyverse packages
- R for Data Science Book - Wrangle Chapter
- RStudio CheatSheets
  - Data import with `readr`, `readxl`, and `googlesheets4`
  - Data Transformation with `dplyr`
  - Data tidying with `tidyr`
  - String manipulation with `stringr`
  - Factors with `forcats`
  - Dates and times with `lubridate`



# Ejercicio 1:

Calcular composición de basura en España:

# Ejercicio 1:

Calcular composición de basura en España:

```
composition_complete %>%  
  filter(country == "Spain") %>%  
  select(composition, percent)
```

```
## # A tibble: 9 × 3  
## # Groups:   country [1]  
##   country      composition    percent  
##   <chr>        <chr>           <dbl>  
## 1 Spain       composition_food_organic_waste_percent     49  
## 2 Spain       composition_glass_percent                  8  
## 3 Spain       composition_metal_percent                 3  
## 4 Spain       composition_other_percent                14  
## 5 Spain       composition_paper_cardboard_percent    15  
## 6 Spain       composition_plastic_percent                9  
## 7 Spain       composition_rubber_leather_percent     NA  
## 8 Spain       composition_wood_percent                  2  
## 9 Spain       composition_yard_garden_green_waste_percent NA
```

## Ejercicio 2:

Usando el dataset `waste_world` - calcular la población media por regiones en millones de habitantes.

# Ejercicio 2:

Usando el dataset `waste_world` - calcular la población media por regiones en millones de habitantes.

```
waste_world %>%
  group_by(region_id) %>%
  summarise(pop = mean(population, na.rm = TRUE)) %>%
  mutate(pop = pop / 1000000)
```

```
## # A tibble: 7 × 2
##   region_id          pop
##   <chr>              <dbl>
## 1 East_Asia_Pacific    61.6
## 2 Europe_Central_Asia   15.7
## 3 Latin_America        15.0
## 4 Middle_East_North_Africa 20.2
## 5 North_America         121.
## 6 South_Asia            223.
## 7 Sub-Saharan_Africa    18.9
```

# Ejercicio 3:

Crear una variable basada en el nivel de basura per cápita y contar el numero de países en cada grupo.

Ejemplo:

- high\_waste = >0.6 toneladas de basura por persona al año
- medium\_waste = 0.2 a 0.6 toneladas de basura por persona al año
- low\_waste = <0.2 toneladas de basura por persona al año

# Ejercicio 3:

Crear una variable basada en el nivel de basura per cápita y contar el número de países en cada grupo.

- high\_waste = >0.6 toneladas de basura por persona al año
- medium\_waste = 0.2 a 0.6 toneladas de basura por persona al año
- low\_waste = <0.2 toneladas de basura por persona al año

```
waste_world %>%
  select(country, population, total_waste) %>%
  mutate(waste_per_pers = total_waste / population) %>%
  mutate(waste_levels = case_when(
    waste_per_pers >= 0.6 ~ "high_waste",
    waste_per_pers <= 0.2 ~ "low_waste",
    waste_per_pers < 0.6 & waste_per_pers > 0.2 ~ "medium_waste")) %>%
  group_by(waste_levels) %>%
  summarise(n_countries = n())
```

```
## # A tibble: 4 × 2
##   waste_levels n_countries
##   <chr>          <int>
## 1 high_waste      34
## 2 low_waste       65
## 3 medium_waste    116
## 4 <NA>              2
```

# Ejercicio 4:

Usando la nueva categoria de niveles de basura, contar paises por region y crear una tabla como esta:

region_id	high_waste	low_waste	medium_waste
East_Asia_Pacific	6	10	21
Europe_Central_Asia	14	5	39
Latin_America	8	4	28
Middle_East_North_Africa	3	3	15
North_America	3	0	0
South_Asia	0	7	1
Sub-Saharan_Africa	0	36	12

# Ejercicio 4:

Usando la nueva categoria de niveles de basura, contar paises por region y crear una tabla como esta:

```
waste_world %>%
  select(region_id, country, population, total_waste) %>%
  mutate(waste_per_pers = total_waste/population) %>%
  filter(!is.na(waste_per_pers)) %>%
  mutate(waste_levels = case_when(
    waste_per_pers >= 0.6 ~ "high_waste",
    waste_per_pers <= 0.2 ~ "low_waste",
    waste_per_pers < 0.6 & waste_per_pers > 0.2 ~ "medium_waste")) %>%
  group_by(region_id, waste_levels) %>%
  summarise(n_countries=n()) %>%
  pivot_wider(names_from=waste_levels, values_from=n_countries) %>%
  replace(is.na(.), 0) %>%
  kable()
```

## Ejercicio 5:

Usando el dataset `waste_world` - calcular el residuo de basura plástica en millones de toneladas (`composition_plastic_percent * total_waste`) por continente y ordenarlo de mayor a menor.

# Ejercicio 5:

Usando el dataset `waste_world` - calcular el residuo de basura plástica en millones de toneladas (`composition_plastic_percent * total_waste`) por continente y ordenarlo de mayor a menor.

```
waste_world %>%
  select(continent,
         total_waste,
         plastic_per=composition_plastic_percent) %>%
  mutate(plastic_waste=total_waste*plastic_per) %>%
  group_by(continent) %>%
  summarise(plastic = sum(plastic_waste, na.rm=T) / 1000000) %>%
  arrange(desc(plastic))
```

```
## # A tibble: 8 × 2
##   continent          plastic
##   <chr>              <dbl>
## 1 Asia                8565.
## 2 North America        4387.
## 3 Europe               4261.
## 4 South America        1735.
## 5 Africa                1266.
## 6 Oceania                 152.
## 7 Seven seas (open ocean)  6.54
## 8 <NA>                  0.335
```