Manejo de datos en R

Elena Quintero

13/01/2025

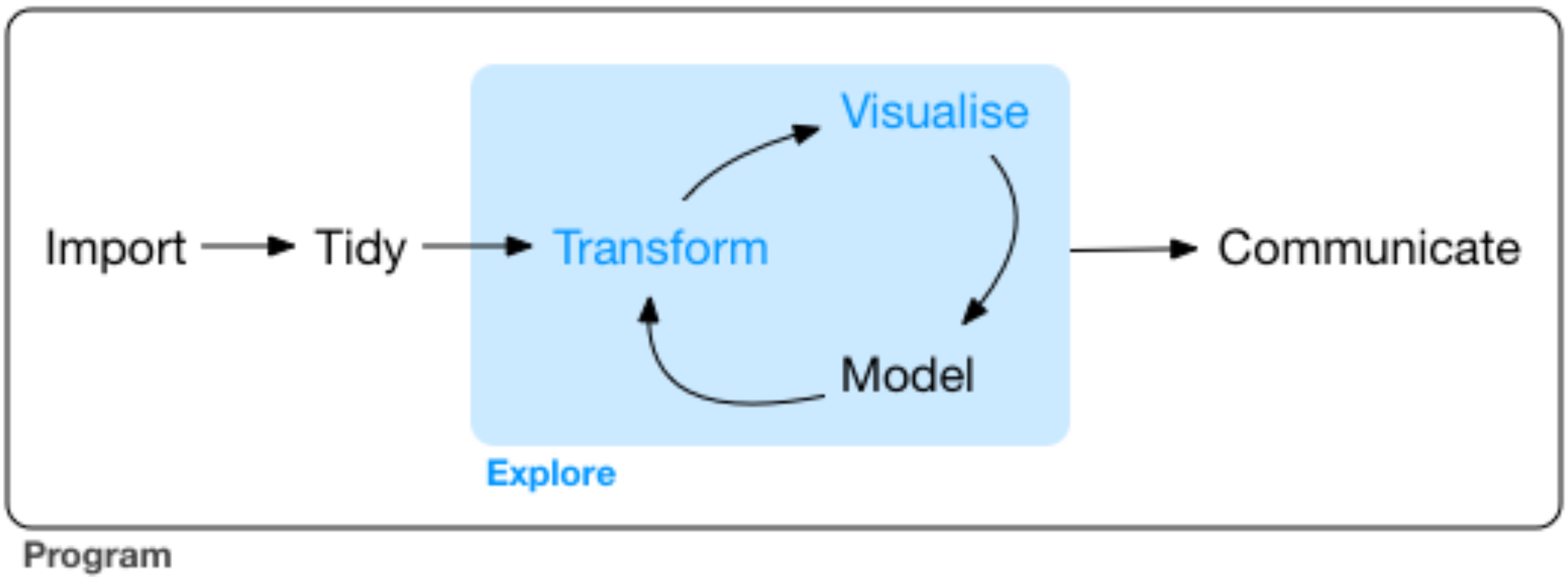
Carpeta con material

https://rstats-courses.github.io/CursoR-AEET-2025/materiales.html

Exploración de datos

La exploración de datos nos permite verificar su calidad, generar y probar hipótesis de forma rápida, identificando pistas prometedoras para analizar más a fondo luego.

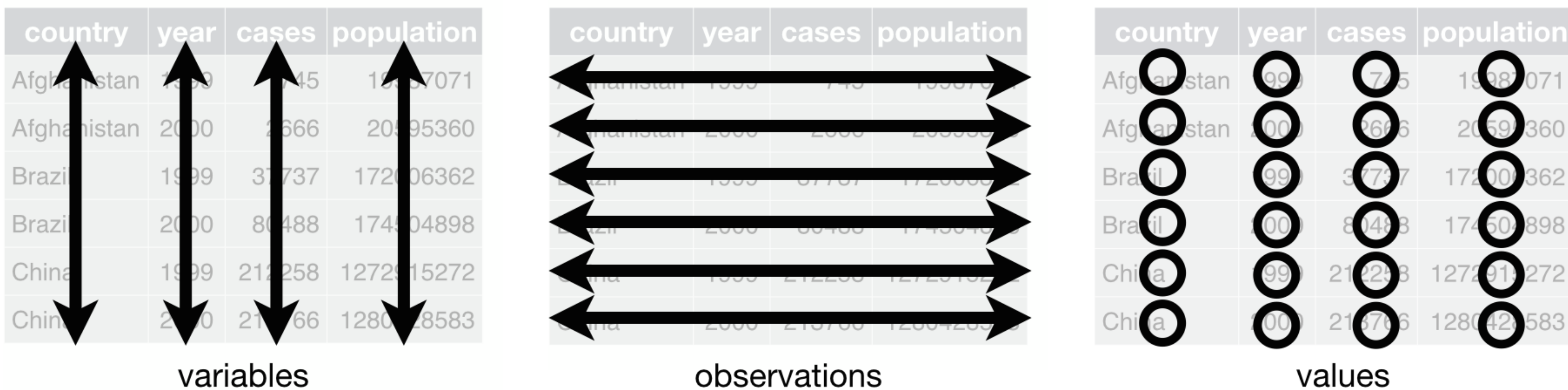
La visualización de los datos es un buen comienzo, pero por sí sola no suele ser suficiente, ya que a menudo requiere transformar los datos previamente.



https://r4ds.had.co.nz/explore-intro.html

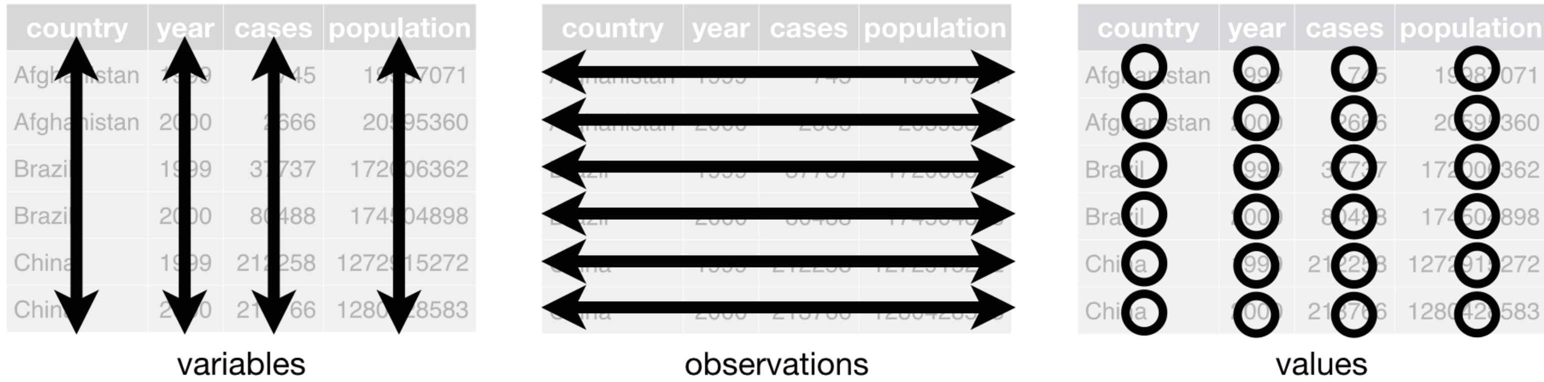
Formato tidy data

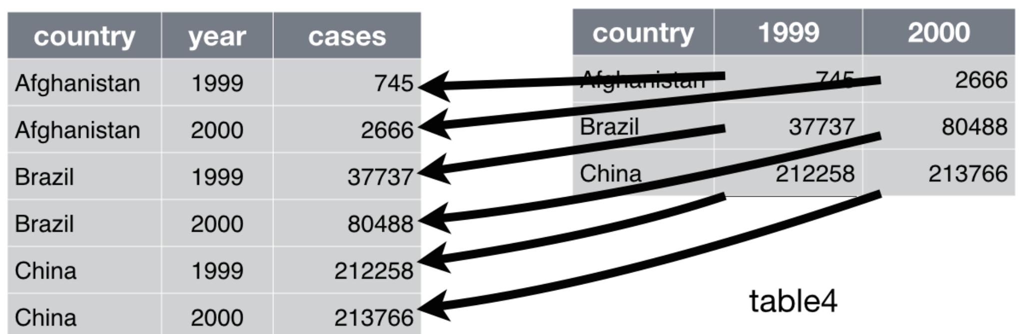
- Cada variable tiene su propia columna
- Cada observación tiene su propia fila
- Cada valor tiene su propia celda



R for Data Science - tidy data

Formato tidy data





Buenas practicas para la recolección de datos

- Poner **variables** en **columnas** (e.g. mediciones: altura, peso, sexo)
- Cada **observación** en una **fila** (e.g. individuos).
- **Evitar** espacios, números, y **caracteres especiales** en los nombres de columnas.
- Siempre **anotar valores de cero**, para diferenciarlos de datos faltantes.
- Usar celdas vacías o con NA para datos faltantes.
- Las fechas incluirlas en columnas separadas como **year, month, day**. O con formato **YYYY-MM-DD** como texto.
- No combinar varias informaciones en una misma celda.
- **No manipular los datos brutos** Realiza todas las manipulaciones de datos mediante código para dejar constancia de los cambios.
- Exporta los datos como texto plano (txt, csv)
- Usar **Data validation** en Excel (or GForms) para limitar la introducción de datos sólo a valores aceptados.
- http://www.datacarpentry.org/spreadsheet-ecology-lesson/
- http://kbroman.org/dataorg/
- Broman & Woo: Data organization in spreadsheets

Errores comunes en tablas de datos

Más de una variable por columna

Date collected	Plot	Species-Sex	Weight
1/9/78	1	DM-M	40
1/9/78	1	DM-F	36
1/9/78	1	DS-F	135
1/20/78	1	DM-F	39
1/20/78	2	DM-M	43
1/20/78	2	DS-F	144
3/13/78	2	DM-F	51
3/13/78	2	DM-F	44
3/13/78	2	DS-F	146

Date collected	Plot	Species	Sex	Weight
1/9/78	1	DM	M	40
1/9/78	1	DM	F	36
1/9/78	1	DS	F	135
1/20/78	1	DM	F	39
1/20/78	2	DM	M	43
1/20/78	2	DS	F	144
3/13/78	2	DM	F	51
3/13/78	2	DM	F	44
3/13/78	2	DS	F	146

Source: Data Carpentry

Errores comunes en tablas de datos

Múltiples tablas

	A	B	C	D	E	F	G	H
1								
2	lake site May 29 2012						29-May	
3			Bug1	bug2			avr	SEM
4	1	T1	1	1	2	T1	2.6	0.51
5	2	T1	1	2	3	T2	0.2	0.2
6	3	T1	1	3	4	control	0.2	0.2
7	4	T1	1	0	1			
8	5	T1	0	3	3			
9	6	T2	1	0	1			
10	7	T2	0	0	0			
11	8	T2	0	0	0			
12	9	T2	0	0	0			
13	10	T2	0	0	0			
14	11	contro	0	0	0			
15	12	contro	0	0	0			
16	13	contro	0	0	0			
17	14	contro	0	0	0			
18	15	contro	1	0	1			
19								
20								
21	Barn site May 29. 2012							
22		plot	bug1	bug2	general		29-May	
23	1	T1	3	3	6			
24	2	T1	1	4	5		avr	SEM
25	3	T1	0	0	0	T1	2.4	1.288
26	4	T1	0	0	0	T2	0.4	0.245
27	5	T1	0	1	1	control	1	0.316
28	6	T2	0	0	0			
29	7	T2	0	0	0			
30	8	T2	0	1	1			
31	9	T2	0	1	1			
32	10	T2	0	0	0			
33	11	contro	0	0	0			
34	12	contro	0	1	1			
35	13	contro	0	1	1			
36	14	contro	0	1	1			
37	15	contro	0	2	2			
38								
39								

	I	J	K	L	M	N	O	P
1								
2	lake site Jun 12. 2012						12-Jun	
3		plot	bug	bug 2			avr	SEM
4	1	T1	6	85	91	T1	30.4	15.47126
5	2	T1	8	13	21	T2	0.2	0.2
6	3	T1	11	0	11	control	0.6	0.6
7	4	T1	0	6	6			
8	5	T1	3	20	23			
9	6	T2	0	0	0			
10	7	T2	0	0	0			
11	8	T2	1	0	1			
12	9	T2	0	0	0			
13	10	T2	0	0	0			
14	11	contro	0	0	0			
15	12	contro	0	0	0			
16	13	contro	0	0	0			
17	14	contro	0	0	0			
18	15	contro	3	0	3			
19								
20								
21	Barn site Jun 12. 2012							
22		plot	bug	bug2	general		12-Jun	
23	1	T1	21	0	21			
24	2	T1	36	74	110		avr	SEM
25	3	T1	13	0	13	T1	30.6	20.10124
26	4	T1	7	0	7	T2	1	0.774597
27	5	T1	2	0	2	control	2.2	1.714643
28	6	T2	1	0	1			
29	7	T2	0	4	4			
30	8	T2	0	0	0			
31	9	T2	0	0	0			
32	10	T2	0	0	0			
33	11	contro	1	0	1			
34	12	contro	0	0	0			
35	13	contro	0	0	0			
36	14	contro	8	1	9			
37	15	contro	0	1	1			
38								
39								

	Q	R	S	T	U	V	W	X
1								
2		lake site Jun 19. 2012					19-Jun	
3		plot	bug1	bug2	general			
4	1	T1	17	80	97		avr	SEM
5	2	T1	44	136	180	T1	77.8	30.384865
6	3	T1	18	0	18	T2	1.8	1.5620499
7	4	T1	0	14	14	control	0.4	0.244949
8	5	T1	10	70	80			
9	6	T2	1	7	8			
10	7	T2	0	1	1			
11	8	T2	0	0	0			
12	9	T2	0	0	0			
13	10	T2	0	0	0			
14	11	control	0	0	0			
15	12	control	0	0	0			
16	13	control	0	0	0			
17	14	control	0	1	1			
18	15	control	0	1	1			
19								
20								
21	Barn site Jun 19 . 2012							
22		plot	bug1	bug2	general		19-Jun	
23	1	T1	5	0	5			
24	2	T1	65	502	567		avr	SEM
25	3	T1	10	7	17	T1	119.4	111.92882
26	4	T1	0	6	6	T2	5	2.1908902
27	5	T1	0	2	2	control	2.8	0.969536
28	6	T2	0	8	8			
29	7	T2	0	12	12			
30	8	T2	0	0	0			
31	9	T2	3	0	3			
32	10	T2	2	0	2			
33	11	control	0	5	5			
34	12	control	1	1	2			
35	13	control	0	0	0			
36	14	control	0	5	5			
37	15	control	0	2	2			
38								
39								

	Y	Z	AA	AB	AC	AD	AE	AF
1								
2		Lake site Jun 26. 2012					26-Jun	
3		plot	bug1	bug2	general			
4	1	T1	52	191	243		avr	SEM
5	2	T1	50	270	320	T1	141.6	60.313
6	3	T1	6	0	6	T2	0.2	0.2
7	4	T1	0	39	39	control	0	0
8	5	T1	4	96	100			
9	6	T2	0	1	1			
10	7	T2	0	0	0			
11	8	T2	0	0	0			
12	9	T2	0	0	0			
13	10	T2	0	0	0			
14	11	control	0	0	0			
15	12	control	0	0	0			
16	13	control	0	0	0			
17	14	control	0	0	0			
18	15	control	0	0	0			
19								
20								
21	Barn Site Jun 26 . 2012							
22		plot	bug1	bug2	general		26-Jun	
23	1	T1	0	0	0		avr	SEM
24	2	T1	44	2057	2101	T1	431.8	417.33
25	3	T1	12	20	32	T2	0.4	0.4
26	4	T1	0	16	16	control	1.2	0.5831
27	5	T1	0	10	10			
28	6	T2	0	0	0			
29	7	T2	0	0	0			
30	8	T2	0	0	0			
31	9	T2	0	0	0			
32	10	T2	0	2	2			
33	11	control	0	2	2			
34	12	control	1	0	1			
35	13	control	0	0	0			
36	14	control	0	3	3			
37	15	control	1	0	0			
38								
39								

Source: Data Carpentry

Errores comunes en tablas de datos

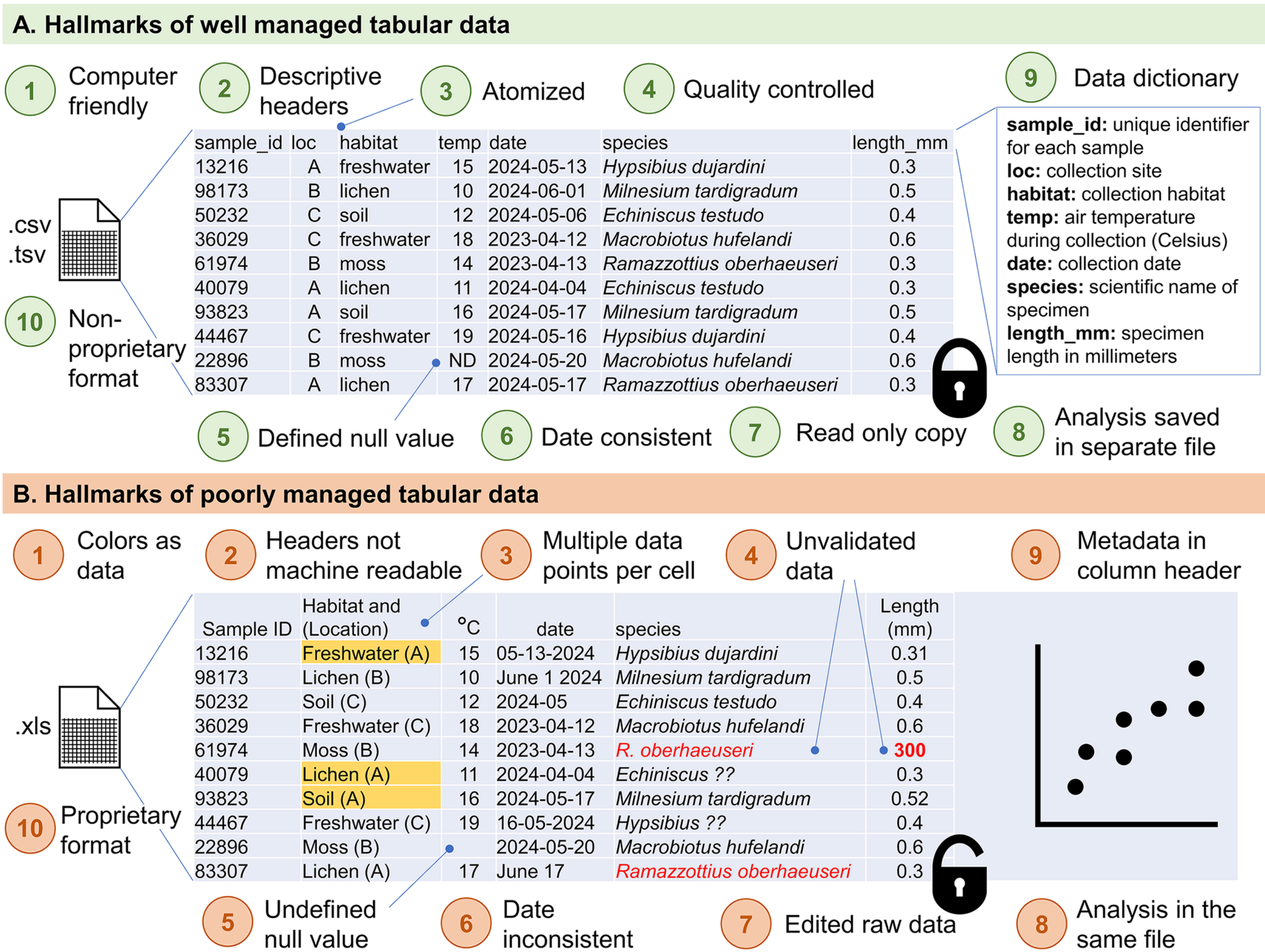
Información en colores

Se puede evitar simplemente añadiendo una columna a la tabla original.

Plot: 2			
Date collect	Species	Sex	Weight
1/8/14	NA		
1/8/14	DM	M	44
1/8/14	DM	M	38
1/8/14	OL		
1/8/14	PE	M	22
1/8/14	DM	M	38
1/8/14	DM	M	48
1/8/14	DM	M	43
1/8/14	DM	F	35
1/8/14	DM	M	43
1/8/14	DM	F	37
1/8/14	PF	F	7
1/8/14	DM	M	45
1/8/14	OT		
1/8/14	DS	M	157
1/8/14	OX		
2/18/14	NA	M	218
2/18/14	PF	F	7
2/18/14	DM	M	52
	measurement device not calibrated		

Date collect	Species	Sex	Weight	Calibrated
1/8/14	NA			
1/8/14	DM	M	44	Y
1/8/14	DM	M	38	Y
1/8/14	OL			
1/8/14	PE	M	22	Y
1/8/14	DM	M	38	Y
1/8/14	DM	M	48	Y
1/8/14	DM	M	43	Y
1/8/14	DM	F	35	Y
1/8/14	DM	M	43	Y
1/8/14	DM	F	37	Y
1/8/14	PF	F	7	Y
1/8/14	DM	M	45	Y
1/8/14	OT			
1/8/14	DS	M	157	N
1/8/14	OX			
2/18/14	NA	M	218	N
2/18/14	PF	F	7	Y
2/18/14	DM	M	52	Y

Recolección de datos



Hertz & McNeill 2024 PLoS Comput Biol

Paquetes que usaremos

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

Paquetes incluidos en tidyverse

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

```
tidyverse::tidyverse_packages()
```

```
 [1] "broom"         "conflicted"    "cli"           "dbplyr"       
 [5] "dplyr"         "dtplyr"        "forcats"       "ggplot2"      
 [9] "googledrive"   "googlesheets4" "haven"         "hms"          
[13] "httr"          "jsonlite"      "lubridate"     "magrittr"     
[17] "modelr"        "pillar"        "purrr"         "ragg"         
[21] "readr"         "readxl"        "reprex"        "rlang"        
[25] "rstudioapi"    "rvest"         "stringr"       "tibble"       
[29] "tidyr"         "xml2"          "tidyverse"    
```

Otros paquetes útiles para el manejo de datos

```r
library(tidylog)
```

Da información de las operaciones que se realizan en el dataset

```r
library(summarytools)
```

Permite hacer resumenes completos de los datasets

Importar datos

```
library(base)
```

`read.table()`, `read.csv()`, `readRDS()`, `read.txt()`

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

```
library(readr)
```

`read_csv()`, `read_csv2()`, `read_table()`, `read_delim()`

Más rapido, produce "tibbles", no convierte characteres a factors automaticamente, 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

Ruta a los datos

```
library(here)
```

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

Ejemplo usando ruta absoluta:

```
data <- read_csv("C:/Usuarios/Elena/Documentos/Mis_proyectos/US/Proyecto_frutos/da
```

Ejemplo usando ruta relativa al proyecto:

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

El operador ‘pipe’

Mecanismo para encadenar funciones:

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

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

Dataset

Received: 30 January 2024 | Revised: 19 July 2024 | Accepted: 28 August 2024

DOI: 10.1002/ecy.4463

DATA PAPER

ECOLOGY
ECOLOGICAL SOCIETY OF AMERICA

Co-mast: Harmonized seed production data for woody plants across US long-term research sites

Katherine M. Nigro[1] | **Jessica H. Barton**[2] | **Diana Macias**[3] | **V. Bala Chaudhary**[4] | **Ian S. Pearse**[5] | **David M. Bell**[6] | **Angel Chen**[7] | **Natalie L. Cleavitt**[8] | **Elizabeth E. Crone**[9] | **David F. Greene**[10] | **E. Penelope Holland**[11] | **Jill F. Johnstone**[12] | **Walter D. Koenig**[13] | **Nicholas J. Lyon**[7] | **Tom E. X. Miller**[14] | **Mark Schulze**[15] | **Rebecca S. Snell**[16] | **Jess K. Zimmerman**[17] | **Johannes M. H. Knops**[18] | **Stacy McNulty**[19] | **Robert R. Parmenter**[20] | **Mark A. Winterstein**[12] | **Roman I. Zlotin**[21] | **Jalene M. LaMontagne**[2,22,23] | **Miranda D. Redmond**[3]

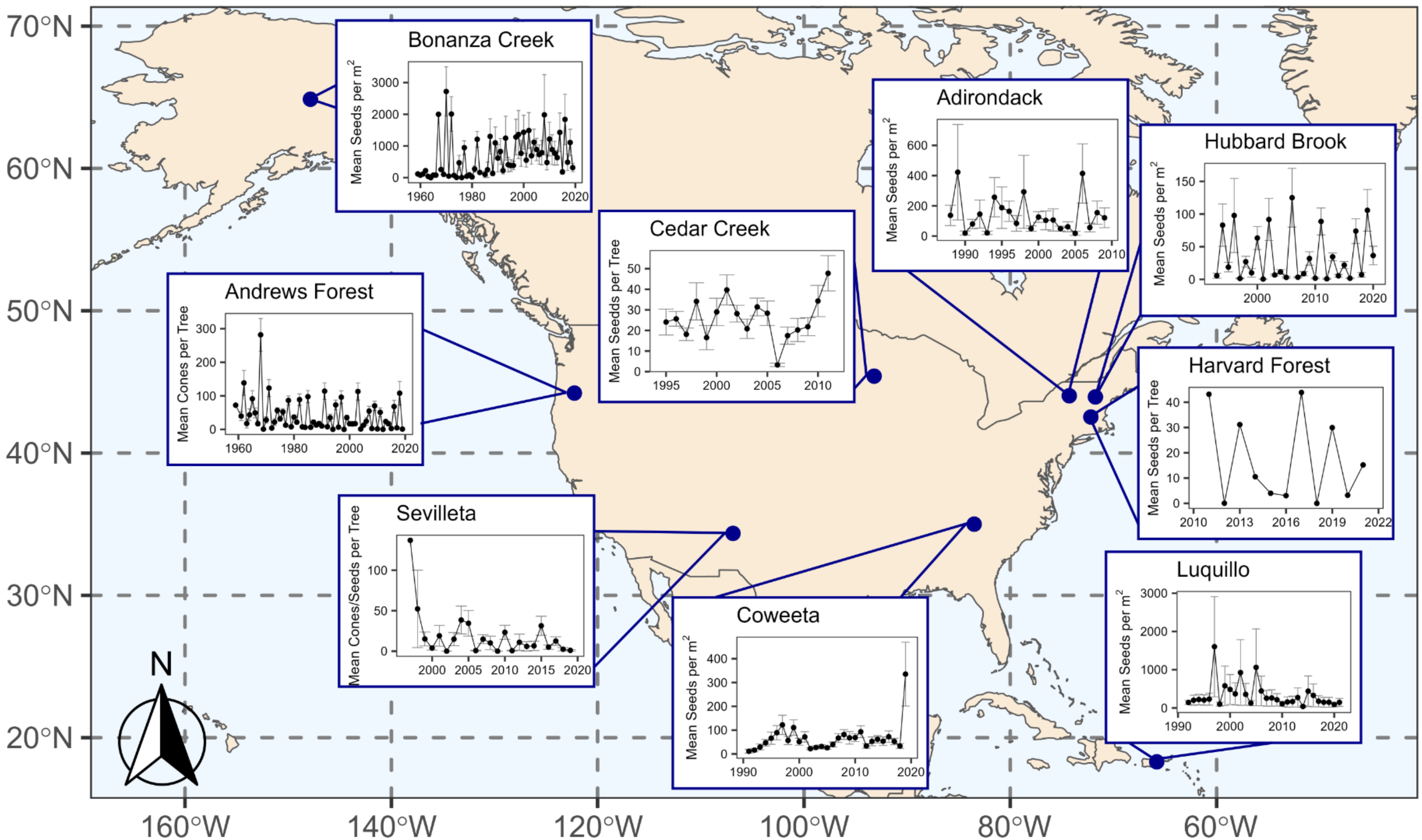
Correspondence
Katherine M. Nigro
Email: katienigro83@gmail.com

Present address
Katherine M. Nigro, Oak Ridge Institute for Science and Education, USA Forest

Abstract

Plants display a range of temporal patterns of inter-annual reproduction, from relatively constant seed production to "mast seeding," the synchronized and highly variable interannual seed production of plants within a population. Previous efforts have compiled global records of seed production in long-lived

Dataset



https://doi.org/10.1002/ecy.4463

Cargar paquetes

```
library(here)
library(tidyverse)
library(tidylog)
library(summarytools)
```

Importar datos

```
dt_raw <- read_csv(here("data/individual_seed_production.csv"))
```

```
Rows: 213062 Columns: 14
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (8): site_name, megaplot, plot, plant_ID, species_name, height_diameter_...
dbl (6): trap, year, count, stem_diameter_cm, trap_area_m2, burned

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.
```

```r
glimpse(dt_raw)
```

```
Rows: 213,062
Columns: 14
$ site_name              <chr> "AND", "AND", "AND", "AND", "AND", "AND", "AND",…
$ megaplot               <chr> "Bare Mountain", "Bare Mountain", "Bare Mountain…
$ plot                   <chr> "CNCT_01", "CNCT_01", "CNCT_01", "CNCT_01", "CNC…
$ trap                   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
$ plant_ID               <chr> "CNCT_01ABAM1", "CNCT_01ABAM1", "CNCT_01ABAM1", …
$ species_name           <chr> "Abies_amabilis", "Abies_amabilis", "Abies_amabi…
$ year                   <dbl> 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, …
$ count                  <dbl> 22, 0, 0, 2, 0, 2, 108, 0, 0, 7, 0, 0, 2, 0, 12,…
$ stem_diameter_cm       <dbl> 56.6, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
$ trap_area_m2           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
$ height_diameter_taken <chr> "Breast Height", "Breast Height", "Breast Height…
```

```r
head(dt_raw)
```

```
# A tibble: 6 × 14
  site_name megaplot      plot     trap plant_ID     species_name    year count
  <chr>     <chr>         <chr>   <dbl> <chr>        <chr>          <dbl> <dbl>
1 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis  1962    22
2 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis  1963     0
3 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis  1964     0
4 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis  1965     2
5 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis  1966     0
6 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis  1967     2
# ℹ 6 more variables: stem_diameter_cm <dbl>, trap_area_m2 <dbl>,
#   height_diameter_taken <chr>, burned <dbl>, general_method <chr>,
#   methods_notes <chr>
```

Funciones de dplyr (parte 1)

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

Ordernar datos por columnas

```r
dt_raw |>
  arrange(count)
```

```
# A tibble: 213,062 × 14
   site_name megaplot      plot     trap plant_ID     species_name     year count
   <chr>     <chr>         <chr>   <dbl> <chr>        <chr>           <dbl> <dbl>
 1 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1963     0
 2 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1964     0
 3 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1966     0
 4 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1969     0
 5 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1970     0
 6 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1972     0
 7 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1973     0
 8 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1975     0
 9 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1977     0
10 AND       Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   1981     0
```

Ordernar datos por columnas

De mayor a menor:

```
dt_raw |>
  arrange(desc(count))
```

```
# A tibble: 213,062 × 14
   site_name megaplot plot   trap plant_ID species_name            year   count
   <chr>     <chr>    <chr> <dbl> <chr>    <chr>                  <dbl>   <dbl>
 1 LUQ       1        1        92 <NA>     Cecropia_schreberiana   1997 1114340
 2 LUQ       1        1        93 <NA>     Ficus_trigonata         2015  106650
 3 LUQ       1        1        92 <NA>     Ficus_trigonata         2013   69450
 4 LUQ       1        1        93 <NA>     Ficus_trigonata         2018   44090
 5 LUQ       1        1       109 <NA>     Ficus_trigonata         2015   39670
 6 LUQ       1        1        92 <NA>     Ficus_trigonata         2015   35075
 7 LUQ       1        1        93 <NA>     Ficus_trigonata         2016   33500
 8 LUQ       1        1        92 <NA>     Ficus_trigonata         2008   33000
 9 LUQ       1        1       107 <NA>     Ficus_trigonata         2011   32689
10 LUQ       1        1        99 <NA>     Cecropia_schreberiana   2009   30594
```

Ordernar datos por columnas

Por orden jerárquico:

```
dt_raw |>
  arrange(site_name, species_name, desc(count))
```

```
# A tibble: 213,062 × 14
   site_name megaplot   plot        trap plant_ID species_name  year count
   <chr>     <chr>      <chr>      <dbl> <chr>    <chr>        <dbl> <dbl>
 1 AEC       adirondack adirondack   971 <NA>     Acer_rubrum   2009   191
 2 AEC       adirondack adirondack   971 <NA>     Acer_rubrum   2004   171
 3 AEC       adirondack adirondack   971 <NA>     Acer_rubrum   1995   141
 4 AEC       adirondack adirondack   941 <NA>     Acer_rubrum   1994   105
 5 AEC       adirondack adirondack   941 <NA>     Acer_rubrum   1995    85
 6 AEC       adirondack adirondack   972 <NA>     Acer_rubrum   2007    82
 7 AEC       adirondack adirondack   971 <NA>     Acer_rubrum   2008    81
 8 AEC       adirondack adirondack   971 <NA>     Acer_rubrum   1993    79
 9 AEC       adirondack adirondack   941 <NA>     Acer_rubrum   1991    77
10 AEC       adirondack adirondack   938 <NA>     Acer_rubrum   2004    72
```

Renombrar variables

```
dt_raw |>
  rename(site = site_name)
```

```
rename: renamed one variable (site)

# A tibble: 213,062 × 14
   site  megaplot plot   trap plant_ID species_name  year count stem_diameter_cm
   <chr> <chr>    <chr> <dbl> <chr>    <chr>        <dbl> <dbl>            <dbl>
 1 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1962    22             56.6
 2 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1963     0             NA
 3 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1964     0             NA
 4 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1965     2             NA
 5 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1966     0             NA
 6 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1967     2             NA
 7 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1968   108             NA
 8 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1969     0             NA
 9 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1970     0             NA
10 AND   Bare Mo… CNCT…    NA CNCT_01… Abies_amabi…  1971     7             NA
```

Organizar columnas

```r
dt_raw |>
  relocate(year, .before = megaplot)
```

```
relocate: columns reordered (site_name, year, megaplot, plot, trap, …)
```

```
# A tibble: 213,062 × 14
   site_name  year megaplot      plot     trap plant_ID     species_name   count
   <chr>     <dbl> <chr>         <chr>   <dbl> <chr>        <chr>          <dbl>
 1 AND        1962 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis    22
 2 AND        1963 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis     0
 3 AND        1964 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis     0
 4 AND        1965 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis     2
 5 AND        1966 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis     0
 6 AND        1967 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis     2
 7 AND        1968 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis   108
 8 AND        1969 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis     0
 9 AND        1970 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis     0
10 AND        1971 Bare Mountain CNCT_01    NA CNCT_01ABAM1 Abies_amabilis     7
```

Seleccionar variables de interés

```r
dt_raw |>
  select(site_name, year, species_name, count)
```

```
select: dropped 10 variables (megaplot, plot, trap, plant_ID, stem_diameter_cm, …)
# A tibble: 213,062 × 4
   site_name  year species_name   count
   <chr>     <dbl> <chr>          <dbl>
 1 AND        1962 Abies_amabilis    22
 2 AND        1963 Abies_amabilis     0
 3 AND        1964 Abies_amabilis     0
 4 AND        1965 Abies_amabilis     2
 5 AND        1966 Abies_amabilis     0
 6 AND        1967 Abies_amabilis     2
 7 AND        1968 Abies_amabilis   108
 8 AND        1969 Abies_amabilis     0
 9 AND        1970 Abies_amabilis     0
10 AND        1971 Abies_amabilis     7
```

Seleccionar variables de interés

Quitar variables:

```
dt_raw |>
  select(-c(megaplot, plot, trap))
```

```
select: dropped 3 variables (megaplot, plot, trap)
```

```
# A tibble: 213,062 × 11
   site_name plant_ID     species_name   year count stem_diameter_cm trap_area_m2
   <chr>     <chr>        <chr>         <dbl> <dbl>            <dbl>        <dbl>
 1 AND       CNCT_01ABAM1 Abies_amabi…   1962    22             56.6           NA
 2 AND       CNCT_01ABAM1 Abies_amabi…   1963     0             NA             NA
 3 AND       CNCT_01ABAM1 Abies_amabi…   1964     0             NA             NA
 4 AND       CNCT_01ABAM1 Abies_amabi…   1965     2             NA             NA
 5 AND       CNCT_01ABAM1 Abies_amabi…   1966     0             NA             NA
 6 AND       CNCT_01ABAM1 Abies_amabi…   1967     2             NA             NA
 7 AND       CNCT_01ABAM1 Abies_amabi…   1968   108             NA             NA
 8 AND       CNCT_01ABAM1 Abies_amabi…   1969     0             NA             NA
 9 AND       CNCT_01ABAM1 Abies_amabi…   1970     0             NA             NA
10 AND       CNCT_01ABAM1 Abies_amabi…   1971     7             NA             NA
```

Seleccionar variables de interés

La función `select()` nos permite seleccionar, renombrar y recolocar - **todo a la vez**!

```
dt <- dt_raw |>
  select(site = site_name,
         year,
         species_name,
         plant_ID,
         count,
         method = general_method,
         stem_cm = stem_diameter_cm,
         trap_area_m2)
```

```
select: renamed 3 variables (site, method, stem_cm) and dropped 6 variables
```

Seleccionar variables de interés

La función `select()` nos permite seleccionar, renombrar y recolocar - **todo a la vez**!

```r
glimpse(dt)
```

```
Rows: 213,062
Columns: 8
$ site         <chr> "AND", "AND", "AND", "AND", "AND", "AND", "AND", "AND", "…
$ year         <dbl> 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, 197…
$ species_name <chr> "Abies_amabilis", "Abies_amabilis", "Abies_amabilis", "Ab…
$ plant_ID     <chr> "CNCT_01ABAM1", "CNCT_01ABAM1", "CNCT_01ABAM1", "CNCT_01A…
$ count        <dbl> 22, 0, 0, 2, 0, 2, 108, 0, 0, 7, 0, 0, 2, 0, 12, 0, 21, 1…
$ method       <chr> "PARTIALCONECOUNT", "PARTIALCONECOUNT", "PARTIALCONECOUNT…
$ stem_cm      <dbl> 56.6, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
$ trap_area_m2 <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
```

Resumen datos

```
summary(dt$species_name)
```

```
   Length     Class      Mode 
   213062 character character 
```

Resumen datos

```r
dfSummary(dt$species_name)
```

```
Data Frame Summary
dt
Dimensions: 213062 x 1
Duplicates: 212921

--------------------------------------------------------------------------------------------------
No   Variable       Stats / Values               Freqs (% of Valid)   Graph
Valid      Missing
---- -------------- ---------------------------- -------------------- ------------- -
--------- ---------
1    species_name   1. Abies_amabilis             15488 ( 7.3%)       I
213062     0
```

Resumen datos

```
summary(dt$count)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's 
      0.0       0.0       0.0      34.6       4.0 1114340.0      4015 
```

Resumen datos

```
dfSummary(dt$count)
```

```
Data Frame Summary
dt
Dimensions: 213062 x 1
Duplicates: 211712

----------------------------------------------------------------------------------------
-------------
No   Variable    Stats / Values              Freqs (% of Valid)     Graph   Valid
Missing
---- ----------- --------------------------- ---------------------- ------- ------
--- ---------
1    count       Mean (sd) : 34.6 (2481.6)   1349 distinct values   :       209047
4015
```

Funciones de dplyr (parte 2):

- `distinct()` - Extraer valores únicos
- `mutate()` - Crear nuevas variables
- `filter()` - Filtrar datos por casos
- `group_by()` - Agrupar datos por casos
- `summarise()` - Resumir datos por casos
- `case_when()` - Categorizar datos

Extraer valores únicos

Niveles de una variable:

```
dt |>
  distinct(site)
```

```
# A tibble: 9 × 1
  site
  <chr>
1 AND
2 SEV
3 CDR
4 HFR
5 AEC
6 HBR
7 BNZ
8 CWT
9 LUQ
```

Extraer valores únicos

Equivalente en `library(base)`:

```
unique(dt$site)
```

```
[1] "AND" "SEV" "CDR" "HFR" "AEC" "HBR" "BNZ" "CWT" "LUQ"
```

Extraer valores únicos

Niveles de una variable:

```
dt |>
  distinct(site, method)
```

```
# A tibble: 10 × 2
   site  method
   <chr> <chr>
 1 AND   PARTIALCONECOUNT
 2 SEV   ESTIMATEDSEEDCOUNT
 3 SEV   CONECOUNT
 4 CDR   TIMEDSEEDCOUNT
 5 HFR   TIMEDSEEDCOUNT
 6 AEC   TRAP
 7 HBR   TRAP
 8 BNZ   TRAP
 9 CWT   TRAP
10 LUQ   TRAP
```

Crear nuevas variables

Ej: transformar frutos a frutos/m2

```
dt |>
  mutate(fruits_per_m2 = count/trap_area_m2)
```

```
mutate: new variable 'fruits_per_m2' (double) with 2,116 unique values and 35% NA

# A tibble: 213,062 × 9
   site   year species_name    plant_ID      count method      stem_cm trap_area_m2
   <chr> <dbl> <chr>           <chr>         <dbl> <chr>         <dbl>        <dbl>
 1 AND    1962 Abies_amabilis CNCT_01ABAM1     22 PARTIALCO…     56.6           NA
 2 AND    1963 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…    NA              NA
 3 AND    1964 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…    NA              NA
 4 AND    1965 Abies_amabilis CNCT_01ABAM1      2 PARTIALCO…    NA              NA
 5 AND    1966 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…    NA              NA
 6 AND    1967 Abies_amabilis CNCT_01ABAM1      2 PARTIALCO…    NA              NA
 7 AND    1968 Abies_amabilis CNCT_01ABAM1    108 PARTIALCO…    NA              NA
 8 AND    1969 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…    NA              NA
 9 AND    1970 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…    NA              NA
10 AND    1971 Abies_amabilis CNCT_01ABAM1      7 PARTIALCO…    NA              NA
```

Filtrar datos

```
dt |>
  filter(site == "BNZ")
```

```
filter: removed 208,641 rows (98%), 4,421 rows remaining

# A tibble: 4,421 × 8
   site   year species_name plant_ID count method stem_cm trap_area_m2
   <chr> <dbl> <chr>        <chr>    <dbl> <chr>    <dbl>        <dbl>
 1 BNZ    1957 Picea_glauca <NA>         3 TRAP        NA         0.25
 2 BNZ    1957 Picea_glauca <NA>         1 TRAP        NA         0.25
 3 BNZ    1957 Picea_glauca <NA>         2 TRAP        NA         0.25
 4 BNZ    1957 Picea_glauca <NA>         3 TRAP        NA         0.25
 5 BNZ    1957 Picea_glauca <NA>         3 TRAP        NA         0.25
 6 BNZ    1957 Picea_glauca <NA>         0 TRAP        NA         0.25
 7 BNZ    1957 Picea_glauca <NA>         2 TRAP        NA         0.25
 8 BNZ    1957 Picea_glauca <NA>         3 TRAP        NA         0.25
 9 BNZ    1957 Picea_glauca <NA>         0 TRAP        NA         0.25
10 BNZ    1957 Picea_glauca <NA>         1 TRAP        NA         0.25
```

Filtrar datos

```
dt |>
  filter(site %in% c("AEC", "AND", "BNZ")) |>
  filter(count >= 10)
```

```
filter: removed 150,475 rows (71%), 62,587 rows remaining

filter: removed 41,369 rows (66%), 21,218 rows remaining

# A tibble: 21,218 × 8
   site   year species_name   plant_ID     count method      stem_cm trap_area_m2
   <chr> <dbl> <chr>          <chr>        <dbl> <chr>         <dbl>        <dbl>
 1 AND    1962 Abies_amabilis CNCT_01ABAM1    22 PARTIALCO…     56.6           NA
 2 AND    1968 Abies_amabilis CNCT_01ABAM1   108 PARTIALCO…     NA             NA
 3 AND    1976 Abies_amabilis CNCT_01ABAM1    12 PARTIALCO…     NA             NA
 4 AND    1978 Abies_amabilis CNCT_01ABAM1    21 PARTIALCO…     NA             NA
 5 AND    1980 Abies_amabilis CNCT_01ABAM1    30 PARTIALCO…     NA             NA
 6 AND    1982 Abies_amabilis CNCT_01ABAM1    61 PARTIALCO…     NA             NA
 7 AND    1985 Abies_amabilis CNCT_01ABAM1    76 PARTIALCO…     NA             NA
 8 AND    1991 Abies_amabilis CNCT_01ABAM1    42 PARTIALCO…     NA             NA
 9 AND    1995 Abies_amabilis CNCT_01ABAM1    75 PARTIALCO…     NA             NA
10 AND    1997 Abies_amabilis CNCT_01ABAM1    52 PARTIALCO…     NA             NA
```

Agrupar datos y resumir

```r
dt |>
  group_by(site) |>
  summarise(fruits = sum(count))
```

```
group_by: one grouping variable (site)
```

```
summarise: now 9 rows and 2 columns, ungrouped
```

```
# A tibble: 9 × 2
  site    fruits
  <chr>    <dbl>
1 AEC     55731.
2 AND        NA
3 BNZ    915902
4 CDR     85431
5 CWT        NA
6 HBR     24556.
7 HFR      3683
8 LUQ   3653588
9 SEV        NA
```

Agrupar datos y resumir

```r
dt |>
  group_by(site) |>
  summarise(fruits = sum(count, na.rm = TRUE))
```

```
group_by: one grouping variable (site)
summarise: now 9 rows and 2 columns, ungrouped
# A tibble: 9 × 2
  site    fruits
  <chr>    <dbl>
1 AEC     55731.
2 AND   1968048
3 BNZ    915902
4 CDR     85431
5 CWT    292939
6 HBR     24556.
7 HFR      3683
8 LUQ   3653588
9 SEV    231905.
```

Agrupar datos y resumir

```r
dt |>
  group_by(site) |>
  summarise(max_fruit = max(count, na.rm = TRUE),
            min_fruit = min(count, na.rm = TRUE))
```

```
group_by: one grouping variable (site)

summarise: now 9 rows and 3 columns, ungrouped

# A tibble: 9 × 3
  site  max_fruit min_fruit
  <chr>     <dbl>     <dbl>
1 AEC         591         0
2 AND        5000         0
3 BNZ        7230         0
4 CDR         151         0
5 CWT        1383         0
6 HBR         244         0
7 HFR          77         0
8 LUQ     1114340         0
9 SEV        1100         0
```

Agrupar datos y resumir

Crear dataset con media de frutos de cada especie de árbol por sitio y por año:

```
dt |>
  group_by(site, species_name, year) |>
  summarise(mean_fruits = mean(count, na.rm = TRUE)) |>
  ungroup()
```

```
group_by: 3 grouping variables (site, species_name, year)

summarise: now 3,212 rows and 4 columns, 2 group variables remaining (site,
species_name)

ungroup: no grouping variables

# A tibble: 3,212 × 4
   site  species_name  year mean_fruits
   <chr> <chr>        <dbl>       <dbl>
 1 AEC   Acer_rubrum   1988        0
 2 AEC   Acer_rubrum   1989        3.1
 3 AEC   Acer_rubrum   1990        0.44
 4 AEC   Acer_rubrum   1991        9.36
 5 AEC   Acer_rubrum   1992        3.90
 6 AEC   Acer_rubrum   1993        4.45
 7 AEC   Acer_rubrum   1994        9.75
 8 AEC   Acer_rubrum   1995        6.52
 9 AEC   Acer_rubrum   1996        6.86
```

Crear categorias

Crear una nueva variable en base a diferentes niveles de frutos.

Ej - un factor de 3 niveles de cantidad frutos:

```
dt |>
  filter(!is.na(count)) |>
  filter(count != 0) |>
  select(count) |>
  summary()
```

```
filter: removed 4,015 rows (2%), 209,047 rows remaining
filter: removed 126,766 rows (61%), 82,281 rows remaining
select: dropped 7 variables (site, year, species_name, plant_ID, method, …)
     count
 Min.   :      0.1
 1st Qu.:      2.0
 Median :      8.0
 Mean   :     87.9
 3rd Qu.:     35.0
 Max.   :1114340.0
```

Crear categorias

Crear una nueva variable en base a diferentes niveles de frutos.

Ej - un factor de 3 niveles de cantidad frutos:

```
dt |>
  mutate(nivel_frutos = case_when(
    count <= 100 ~ "bajo",
    count > 100 & count <= 1000 ~ "medio",
    count > 1000 ~ "alto"))
```

```
mutate: new variable 'nivel_frutos' (character) with 4 unique values and 2% NA

# A tibble: 213,062 × 9
   site   year species_name   plant_ID      count method      stem_cm trap_area_m2
   <chr> <dbl> <chr>          <chr>         <dbl> <chr>         <dbl>        <dbl>
 1 AND    1962 Abies_amabilis CNCT_01ABAM1     22 PARTIALCO…     56.6           NA
 2 AND    1963 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…     NA             NA
 3 AND    1964 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…     NA             NA
 4 AND    1965 Abies_amabilis CNCT_01ABAM1      2 PARTIALCO…     NA             NA
 5 AND    1966 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…     NA             NA
 6 AND    1967 Abies_amabilis CNCT_01ABAM1      2 PARTIALCO…     NA             NA
 7 AND    1968 Abies_amabilis CNCT_01ABAM1    108 PARTIALCO…     NA             NA
 8 AND    1969 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…     NA             NA
 9 AND    1970 Abies_amabilis CNCT_01ABAM1      0 PARTIALCO…     NA             NA
10 AND    1971 Abies_amabilis CNCT_01ABAM1      7 PARTIALCO…     NA             NA
```

Crear categorias

Contar numero de arboles con distintos niveles de frutos:

```
dt |>
  mutate(nivel_frutos = case_when(
    count <= 100 ~ "bajo",
    count > 100 & count <= 1000 ~ "medio",
    count > 1000 ~ "alto")) |>
  group_by(nivel_frutos) |>
  summarise(trees = n())
```

```
mutate: new variable 'nivel_frutos' (character) with 4 unique values and 2% NA
group_by: one grouping variable (nivel_frutos)
summarise: now 4 rows and 2 columns, ungrouped
# A tibble: 4 × 2
  nivel_frutos  trees
  <chr>         <int>
1 alto            698
2 bajo         200157
3 medio          8192
4 <NA>           4015
```

Funciones vistas de dplyr

Funciones de dplyr (parte 1 y 2)

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

Corregir datos

Función `if_else()`:

```
dt_fix <- dt |>
  # quitar un valor equivocado
  mutate(count = if_else(count > 200000, NA, count))
```

```
mutate: changed one value (<1%) of 'count' (1 new NA)
```

Modificar datos

Función `if_else()`:

```
dt_fix <- dt |>
  # quitar un valor equivocado
  mutate(count = if_else(count > 200000, NA, count)) |>
  # calcular número de frutos por m2
  mutate(fruits_per_m2 = count/trap_area_m2) |>
  # crear variable con la cantidad de frutos de count o corregida
  mutate(fruits = if_else(is.na(fruits_per_m2), count, fruits_per_m2))
```

```
mutate: changed one value (<1%) of 'count' (1 new NA)
mutate: new variable 'fruits_per_m2' (double) with 2,115 unique values and 35% NA
mutate: new variable 'fruits' (double) with 2,305 unique values and 2% NA
```

Modificar datos

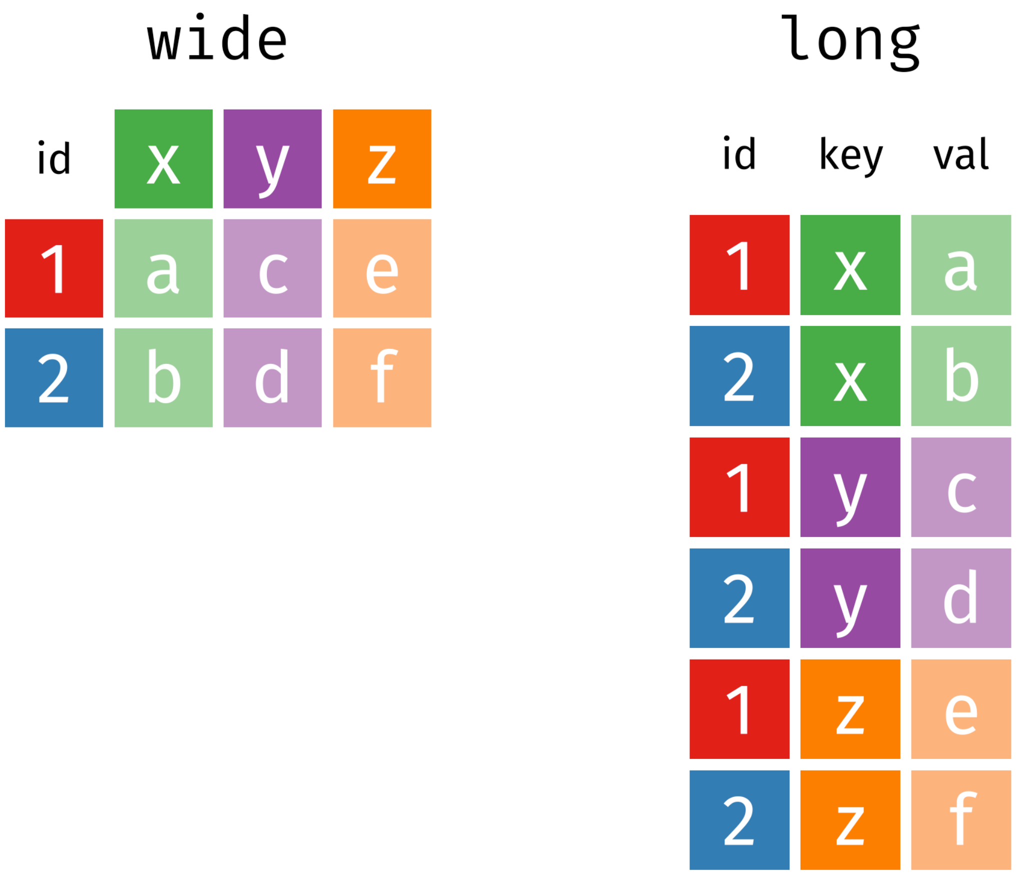
Función `if_else()`:

```
dt_fix <- dt |>
  # quitar un valor equivocado
  mutate(count = if_else(count > 200000, NA, count)) |>
  # calcular número de frutos por m2
  mutate(fruits_per_m2 = count/trap_area_m2) |>
  # crear variable con la cantidad de frutos de count o corregida
  mutate(fruits = if_else(is.na(fruits_per_m2), count, fruits_per_m2)) |>
  # quitar valores de 0 o NA
  filter(count != 0)
```

```
mutate: changed one value (<1%) of 'count' (1 new NA)
mutate: new variable 'fruits_per_m2' (double) with 2,115 unique values and 35% NA
mutate: new variable 'fruits' (double) with 2,305 unique values and 2% NA
filter: removed 130,782 rows (61%), 82,280 rows remaining
```

Reestructurar datos

Reestructurar datos con `library(tidyr)`



- Función `pivot_wider()`
- Función `pivot_longer()`

Fuente: Garrick Aden-Buie's - Tidyexplained Verbs

Reestructurar datos

```r
head(dt_fix)
```

```
# A tibble: 6 × 10
  site   year species_name   plant_ID     count method       stem_cm trap_area_m2
  <chr> <dbl> <chr>          <chr>        <dbl> <chr>          <dbl>        <dbl>
1 AND    1962 Abies_amabilis CNCT_01ABAM1    22 PARTIALCON…     56.6           NA
2 AND    1965 Abies_amabilis CNCT_01ABAM1     2 PARTIALCON…    NA              NA
3 AND    1967 Abies_amabilis CNCT_01ABAM1     2 PARTIALCON…    NA              NA
4 AND    1968 Abies_amabilis CNCT_01ABAM1   108 PARTIALCON…    NA              NA
5 AND    1971 Abies_amabilis CNCT_01ABAM1     7 PARTIALCON…    NA              NA
6 AND    1974 Abies_amabilis CNCT_01ABAM1     2 PARTIALCON…    NA              NA
# ℹ 2 more variables: fruits_per_m2 <dbl>, fruits <dbl>
```

Reestructurar datos

Primero creamos dataset reducido:

```
dt_fix |>
  group_by(site, year) |>
  summarise(fruits = mean(fruits, na.rm. = TRUE))
```

```
group_by: 2 grouping variables (site, year)
```

```
summarise: now 280 rows and 3 columns, one group variable remaining (site)
```

```
# A tibble: 280 × 3
# Groups:   site [9]
   site   year fruits
   <chr> <dbl>  <dbl>
 1 AEC    1988  252.
 2 AEC    1989  656.
 3 AEC    1990   67.3
 4 AEC    1991  148.
 5 AEC    1992  279.
 6 AEC    1993   66.1
 7 AEC    1994  375.
 8 AEC    1995  343.
 9 AEC    1996  250.
```

Reestructurar datos

Convertir a formato corto:

```
dt_short <- dt_fix |>
  group_by(site, year) |>
  summarise(fruits = mean(fruits, na.rm. = TRUE)) |>
  pivot_wider(names_from = "site",
              values_from = "fruits")
```

```
group_by: 2 grouping variables (site, year)

summarise: now 280 rows and 3 columns, one group variable remaining (site)

pivot_wider: reorganized (site, fruits) into (AEC, AND, BNZ, CDR, CWT, …) [was
280x3, now 65x10]
```

```
head(dt_short)
```

```
# A tibble: 6 × 10
   year   AEC   AND   BNZ   CDR   CWT   HBR   HFR   LUQ   SEV
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  1988 252.   27.0  198.    NA    NA  NA      NA    NA    NA
2  1989 656.   46.6 1406.    NA    NA  NA      NA    NA    NA
3  1990  67.3  26.2  909.    NA    NA  NA      NA    NA    NA
4  1991 148.  140.  1318.    NA  154.  NA      NA    NA    NA
5  1992 279.   28.4  357.    NA  101.  NA      NA  249.    NA
6  1993  66.1  65.1 1683.    NA  124.  43.9    NA  269.    NA
```

Reestructurar datos

Convertir a formato largo:

```
dt_short |>
  pivot_longer(cols = c(AEC:SEV),
               names_to = "site",
               values_to = "fruits")
```

```
pivot_longer: reorganized (AEC, AND, BNZ, CDR, CWT, …) into (site, fruits) [was
65x10, now 585x3]
```

```
# A tibble: 585 × 3
    year site  fruits
   <dbl> <chr>  <dbl>
 1  1988 AEC    252. 
 2  1988 AND     27.0
 3  1988 BNZ    198. 
 4  1988 CDR     NA  
 5  1988 CWT     NA  
 6  1988 HBR     NA  
 7  1988 HFR     NA  
 8  1988 LUQ     NA  
 9  1988 SEV     NA  
10  1989 AEC    656. 
```

Combinar bases de datos

Combinar bases de datos con `join`

a

x1	x2
A	1
B	2
C	3

\+

b

x1	x3
A	T
B	F
D	T

=

Mutating Joins

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

dplyr::left_join(a, b, by = "x1")

Join matching rows from b to a.

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

dplyr::right_join(a, b, by = "x1")

Join matching rows from a to b.

x1	x2	x3
A	1	T
B	2	F

dplyr::inner_join(a, b, by = "x1")

Join data. Retain only rows in both sets.

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

dplyr::full_join(a, b, by = "x1")

Join data. Retain all values, all rows.

Combinar bases de datos

Leemos un nuevo dataset con información de atributos para las especies de árboles:

```r
sp_info <- read_csv(here("data/species_attributes.csv"))
```

```
Rows: 104 Columns: 17
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (15): species_name, family, genus, epithet, pollinator_code, mycorrhiza_...
dbl  (2): seed_development_years, seed_mass_mg

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.
```

Combinar bases de datos

```
glimpse(sp_info)
```

```
Rows: 104
Columns: 17
$ species_name           <chr> "Abies_amabilis", "Abies_concolor", "Abies_gran…
$ family                 <chr> "Pinaceae", "Pinaceae", "Pinaceae", "Pinaceae",…
$ genus                  <chr> "Abies", "Abies", "Abies", "Abies", "Abies", "A…
$ epithet                <chr> "amabilis", "concolor", "grandis", "lasiocarpa"…
$ seed_development_years <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,…
$ pollinator_code        <chr> "wind", "wind", "wind", "wind", "wind", "wind",…
$ mycorrhiza_type        <chr> "EM", "EM", "EM", "EM", "EM", "EM", "AM", "AM",…
$ needleleaf_broadleaf   <chr> "needleleaf", "needleleaf", "needleleaf", "need…
$ deciduous_evergreen    <chr> "evergreen", "evergreen", "evergreen", "evergre…
$ seed_maturation_timing <chr> "late summer", "fall", "late summer", "late sum…
$ seed_mass_mg           <dbl> 46.2063354, 34.2847056, 21.0800075, 13.7327226,…
```

Combinar bases de datos

La función `count` cuenta el número de casos para una variable categórica

```
sp_info |> count(pollinator_code)
```

```
# A tibble: 2 × 2
  pollinator_code     n
  <chr>           <int>
1 animal             73
2 wind               31
```

```
sp_info |> count(family)
```

```
# A tibble: 41 × 2
   family            n
   <chr>         <int>
 1 Aceraceae         2
 2 Annonaceae        2
 3 Aquifoliaceae     1
 4 Araliaceae        2
 5 Arecaceae         1
 6 Betulaceae        4
 7 Bignoniaceae      2
 8 Boraginaceae      2
 9 Burseraceae       2
10 Cecropiaceae      1
```

Combinar bases de datos

Usando `left_join()`

```
dt_sp <- dt_fix |>
  left_join(sp_info, by = c("species_name"))
```

```
left_join: added 16 columns (family, genus, epithet, seed_development_years,
pollinator_code, …)
           > rows only in x      293
           > rows only in y  (      1)
           > matched rows     81,987
           >                 ========
           > rows total       82,280
```

Combinar bases de datos

```r
setdiff(sp_info$species_name, dt_fix$species_name)
```

```
[1] "Myrcia_amazonica"
```

Combinar bases de datos

```r
glimpse(dt_sp)
```

```
Rows: 82,280
Columns: 26
$ site                        <chr> "AND", "AND", "AND", "AND", "AND", "AND", "AND"…
$ year                        <dbl> 1962, 1965, 1967, 1968, 1971, 1974, 1976, 1978,…
$ species_name                <chr> "Abies_amabilis", "Abies_amabilis", "Abies_amab…
$ plant_ID                    <chr> "CNCT_01ABAM1", "CNCT_01ABAM1", "CNCT_01ABAM1",…
$ count                       <dbl> 22, 2, 2, 108, 7, 2, 12, 21, 1, 30, 61, 76, 5, …
$ method                      <chr> "PARTIALCONECOUNT", "PARTIALCONECOUNT", "PARTIA…
$ stem_cm                     <dbl> 56.6, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
$ trap_area_m2                <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
$ fruits_per_m2               <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
$ fruits                      <dbl> 22, 2, 2, 108, 7, 2, 12, 21, 1, 30, 61, 76, 5, …
$ family                      <chr> "Pinaceae", "Pinaceae", "Pinaceae", "Pinaceae",…
```

Guardar dataset

```r
write_csv(dt_sp, here("data/clean_data.csv"))
#write_csv2(dt_sp, here("data/clean_data.csv"))
```

- `write_csv` - usa separador de ","
- `write_csv2` - usa separador de ";"
- `write_delim` - usa cualquier separador de datos (ej. delim = "|")

Guardar dataset

```
#install.packages("arrow")
library(arrow)

write_parquet(dt_sp, here("data/clean_data.parquet"))

dt_sp |>
  group_by(site) |>
  arrow::write_dataset(path = "data/clean_data", format = "parquet")
```

El formato `parquet` para guardar datos es una forma muy eficiente de manejar grandes bases de datos.

Este formato archiva los datos en forma de columnas, ofrece una compresion mayor que .csv incluso mayor que .rds y es más rapido para trabajar.

Además permite el particionado de datos en diferentes ficheros.

Recursos

- Tidyverse packages
- R for Data Science Book - Capítulo Wrangle
- 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:

Usando la base de datos final (dt_sp), seleccionar datos con información para diámetro de tronco (stem_cm) y ordernar de mayor a menor:

Ejercicio 1:

Usando la base de datos final (dt_sp), seleccionar datos con información para diámetro de tronco (stem_cm) y ordernar de mayor a menor:

```
dt_sp |>
  filter(!is.na(stem_cm)) |>
  arrange(desc(stem_cm))
```

```
filter: removed 80,777 rows (98%), 1,503 rows remaining

# A tibble: 1,503 × 26
   site   year species_name  plant_ID      count method      stem_cm trap_area_m2
   <chr> <dbl> <chr>         <chr>         <dbl> <chr>         <dbl>        <dbl>
 1 AND    1993 Abies_procera CNCT_37ABPR17   250 PARTIALCO…     221.           NA
 2 AND    1962 Abies_procera CNCT_15ABPR15    50 PARTIALCO…     198.           NA
 3 AND    1962 Abies_procera CNCT_15ABPR8     68 PARTIALCO…     196.           NA
 4 AND    1993 Abies_procera CNCT_37ABPR1    350 PARTIALCO…     186.           NA
 5 AND    1961 Abies_procera CNCT_37ABPR3    231 PARTIALCO…     186.           NA
 6 AND    1993 Abies_procera CNCT_37ABPR3    410 PARTIALCO…     184.           NA
 7 AND    1962 Abies_procera CNCT_15ABPR16   130 PARTIALCO…     183.           NA
 8 AND    1992 Abies_procera CNCT_02ABPR35    54 PARTIALCO…     180.           NA
 9 AND    1962 Abies_procera CNCT_15ABPR3    300 PARTIALCO…     178.           NA
10 AND    1993 Abies_procera CNCT_37ABPR10   250 PARTIALCO…     174.           NA
```

Ejercicio 2:

Usando la base de datos final (dt_sp), calcular diámetro medio y SD para cada especie de árbol.

Ejercicio 2:

Usando la base de datos final (dt_sp), calcular diámetro medio y SD para cada especie de árbol.

```
dt_sp |>
  filter(!is.na(stem_cm)) |>
  group_by(species_name) |>
  summarise(mean = mean(stem_cm),
            sd = sd(stem_cm))
```

```
filter: removed 80,777 rows (98%), 1,503 rows remaining
group_by: one grouping variable (species_name)
summarise: now 10 rows and 3 columns, ungrouped
# A tibble: 10 × 3
   species_name       mean    sd
   <chr>             <dbl> <dbl>
 1 Abies_amabilis     65.0  19.7
 2 Abies_concolor     63.1  18.6
 3 Abies_grandis      74.5  14.8
 4 Abies_lasiocarpa   45.0  16.2
 5 Abies_magnifica    87.9  19.9
 6 Abies_procera     104.   34.4
 7 Picea_engelmannii  80.2  16.5
 8 Pinus_lambertiana 114.   27.7
```

```
 9 Pinus_monticola    63.4  22.4
10 Tsuga_mertensiana  56.5  12.5
```

Ejercicio 3:

Usando la base de datos final (dt_sp), calcular el número de árboles y número de especies mayores de 40cm de diámetro y menores de 40cm de diámetro.

Ejercicio 3:

Usando la base de datos final (dt_sp), calcular el número de árboles y número de especies mayores de 40cm de diámetro y menores de 40cm de diámetro.

```
dt |>
  filter(!is.na(stem_cm)) |>
  mutate(tree_size = case_when(stem_cm >= 40 ~ "big",
                               stem_cm < 40 ~ "small")) |>
  group_by(tree_size) |>
  summarise(n_trees = n(),
            n_species = n_distinct(species_name))
```

```
filter: removed 210,780 rows (99%), 2,282 rows remaining

mutate: new variable 'tree_size' (character) with 2 unique values and 0% NA

group_by: one grouping variable (tree_size)

summarise: now 2 rows and 3 columns, ungrouped

# A tibble: 2 × 3
  tree_size n_trees n_species
  <chr>       <int>     <int>
1 big          2144        10
2 small         138         6
```

Ejercicio 4:

Usando la base de datos final (dt_sp), seleccionar sitios con método de conteo tipo “TRAP” y calcular cantidad máxima y mínima de frutos por m2 para cada sitio.

Ejercicio 4:

Usando la base de datos final (dt_sp), seleccionar sitios con método de conteo tipo “TRAP” y calcular cantidad máxima y mínima de frutos por m2 para cada sitio.

```
dt_sp |>
  filter(method == "TRAP") |>
  group_by(site) |>
  summarise(max_fruit = max(fruits_per_m2),
            min_fruit = mean(fruits_per_m2))
```

```
filter: removed 33,191 rows (40%), 49,089 rows remaining

group_by: one grouping variable (site)

summarise: now 5 rows and 3 columns, ungrouped

# A tibble: 5 × 3
  site  max_fruit min_fruit
  <chr>     <dbl>     <dbl>
1 AEC       8107.     296.
2 BNZ      28920     1088.
3 CWT      12207.     197.
4 HBR       2440       93.4
5 LUQ     213300      299.
```

Ejercicio 5:

Usando la base de datos final (dt_sp), crear una tabla que compare la suma de frutos contados en los sitios CWT y HFR (en columnas), para los años entre 2000-2010 (filas).

Ejercicio 5:

Usando la base de datos final (dt_sp), crear una tabla que compare la suma de frutos contados en los sitios CWT y HFR (en columnas), para los años entre 2000-2010 (filas).

```
dt_sp |>
  filter(site %in% c("CWT", "SEV")) |>
  filter(year %in% c(2000:2010)) |>
  group_by(site, year) |>
  summarise(fruits = sum(fruits)) |>
  pivot_wider(names_from = site, values_from = fruits)
```

```
# A tibble: 11 × 3
   year     CWT   SEV
  <dbl>   <dbl> <dbl>
1  2000  92776.  5561.
2  2001 133160. 28243.
3  2002  45746.   302.
4  2003  63213. 13646.
5  2004  67092. 23964.
6  2005  47034. 20558.
7  2006  84603.   726.
8  2007 114387. 11630.
9  2008 147617. 14634
```

Ejercicio 6:

Usando la base de datos final (dt_sp), crear una tabla que compare la suma de frutos contados entre los años 2001 y 2005 (en columnas), para las especies de Abies (filas).

Ejercicio 6:

Usando la base de datos final (dt_sp), crear una tabla que compare la suma de frutos contados entre los años 2001 y 2005 (en columnas), para las especies de Abies (filas).

```
dt_sp |>
  filter(year %in% c(2001:2005)) |>
  filter(str_detect(species_name, "Abies")) |>
  group_by(year, species_name) |>
  summarise(fruits = sum(fruits)) |>
  pivot_wider(names_from = year, values_from = fruits)
```

```
# A tibble: 6 × 6
  species_name      `2001` `2002` `2003` `2004` `2005`
  <chr>              <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 Abies_amabilis       721   2819   5907     54    864
2 Abies_concolor      1429     92   3032     NA    136
3 Abies_grandis       3509    238   4414     17   1119
4 Abies_magnifica       52   1374   6324      8    570
5 Abies_procera       3308   7772  10485    957   1588
6 Abies_lasiocarpa      NA    443   1974     17     73
```