

# dplyr y Pokemon

Nayeli Luis

**email:** nayeli.luis@ciencias.unam.mx

## dplyr

Es una librería para la manipulación de datos.

## Pasos iniciales

- Llamar a la librería

```
# Instalar si es el caso
# install.packages("tidyverse")
```

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.7      v dplyr  1.0.9
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

- Importar datos. El dataset fue descargado de Kaggle.

```
pokemon <- read_csv("../00_datasets/pokemon.csv")
```

```
## Rows: 801 Columns: 41
## -- Column specification -----
## Delimiter: ","
## chr  (7): abilities, capture_rate, classification, japanese_name, name, type1...
## dbl  (34): against_bug, against_dark, against_dragon, against_electric, again...
##
## 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.
```

```
pokemon
```

```
## # A tibble: 801 x 41
##   abilities      against_bug against_dark against_dragon against_electric
##   <chr>          <dbl>         <dbl>         <dbl>         <dbl>
## 1 ['Overgrow', 'Chlor~      1             1             1             0.5
## 2 ['Overgrow', 'Chlor~      1             1             1             0.5
## 3 ['Overgrow', 'Chlor~      1             1             1             0.5
## 4 ['Blaze', 'Solar Po~    0.5             1             1             1
```

```
## 5 ['Blaze', 'Solar Po~      0.5          1          1          1
## 6 ['Blaze', 'Solar Po~      0.25         1          1          2
## 7 ['Torrent', 'Rain D~      1          1          1          2
## 8 ['Torrent', 'Rain D~      1          1          1          2
## 9 ['Torrent', 'Rain D~      1          1          1          2
## 10 ['Shield Dust', 'Ru~      1          1          1          1
## # ... with 791 more rows, and 36 more variables: against_fairy <dbl>,
## #   against_fight <dbl>, against_fire <dbl>, against_flying <dbl>,
## #   against_ghost <dbl>, against_grass <dbl>, against_ground <dbl>,
## #   against_ice <dbl>, against_normal <dbl>, against_poison <dbl>,
## #   against_psychic <dbl>, against_rock <dbl>, against_steel <dbl>,
## #   against_water <dbl>, attack <dbl>, base_egg_steps <dbl>,
## #   base_happiness <dbl>, base_total <dbl>, capture_rate <chr>, ...
```

Tipo de objeto que es pokemon:

```
class(pokemon)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"        "data.frame"
```

Conocer las primeras filas: **Nota:** Esto no es necesario cuando trabajamos con tibbles porque por default te arroja las primeras líneas del dataframe.

```
head(pokemon)
```

```
## # A tibble: 6 x 41
##   abilities          against_bug against_dark against_dragon against_electric
##   <chr>              <dbl>         <dbl>         <dbl>         <dbl>
## 1 ['Overgrow', 'Chloro~      1          1          1          0.5
## 2 ['Overgrow', 'Chloro~      1          1          1          0.5
## 3 ['Overgrow', 'Chloro~      1          1          1          0.5
## 4 ['Blaze', 'Solar Pow~      0.5         1          1          1
## 5 ['Blaze', 'Solar Pow~      0.5         1          1          1
## 6 ['Blaze', 'Solar Pow~      0.25         1          1          2
## # ... with 36 more variables: against_fairy <dbl>, against_fight <dbl>,
## #   against_fire <dbl>, against_flying <dbl>, against_ghost <dbl>,
## #   against_grass <dbl>, against_ground <dbl>, against_ice <dbl>,
## #   against_normal <dbl>, against_poison <dbl>, against_psychic <dbl>,
## #   against_rock <dbl>, against_steel <dbl>, against_water <dbl>, attack <dbl>,
## #   base_egg_steps <dbl>, base_happiness <dbl>, base_total <dbl>,
## #   capture_rate <chr>, classification <chr>, defense <dbl>, ...
```

Conocer las dimensiones:

```
dim(pokemon)
```

```
## [1] 801  41
```

Conocer los nombres de las columnas

```
colnames(pokemon)
```

```
## [1] "abilities"      "against_bug"    "against_dark"
## [4] "against_dragon" "against_electric" "against_fairy"
## [7] "against_fight"  "against_fire"    "against_flying"
## [10] "against_ghost"  "against_grass"   "against_ground"
## [13] "against_ice"    "against_normal"  "against_poison"
## [16] "against_psychic" "against_rock"    "against_steel"
## [19] "against_water"  "attack"         "base_egg_steps"
```

```
## [22] "base_happiness"      "base_total"          "capture_rate"
## [25] "classification"      "defense"              "experience_growth"
## [28] "height_m"            "hp"                   "japanese_name"
## [31] "name"                 "percentage_male"      "pokedex_number"
## [34] "sp_attack"           "sp_defense"           "speed"
## [37] "type1"                "type2"                "weight_kg"
## [40] "generation"          "is_legendary"
```

La columna `abilities` es una lista (en sintaxis de Python) de las habilidades que el Pokémon es capaz de tener.

Explorar una columna en específico

```
# pokemon$type1
```

Explorar solo algunos elementos:

```
head(pokemon$type1, 30)
```

```
## [1] "grass"      "grass"      "grass"      "fire"       "fire"       "fire"
## [7] "water"      "water"      "water"      "bug"        "bug"        "bug"
## [13] "bug"        "bug"        "bug"        "normal"     "normal"     "normal"
## [19] "normal"     "normal"     "normal"     "normal"     "poison"     "poison"
## [25] "electric"   "electric"   "ground"     "ground"     "poison"     "poison"
```

Otra columna:

```
head(pokemon$classfication, 20)
```

```
## [1] "Seed Pokémon"      "Seed Pokémon"      "Seed Pokémon"
## [4] "Lizard Pokémon"    "Flame Pokémon"     "Flame Pokémon"
## [7] "Tiny Turtle Pokémon" "Turtle Pokémon"    "Shellfish Pokémon"
## [10] "Worm Pokémon"      "Cocoon Pokémon"    "Butterfly Pokémon"
## [13] "Hairy Pokémon"     "Cocoon Pokémon"    "Poison Bee Pokémon"
## [16] "Tiny Bird Pokémon" "Bird Pokémon"       "Bird Pokémon"
## [19] "Mouse Pokémon"     "Mouse Pokémon"
```

## Manejo de datos

En general, cuando tenemos un dataframe muy largo, no utilizamos todos los datos. Nos concentramos en algunas variables y en algunas observaciones. De manera que generamos *subconjuntos* de datos del dataset original.

Para esto tenemos dos opciones:

1. Seleccionar columnas
2. Filtrar por filas.

### `select()`

Permite seleccionar variables en un dataframe usando un lenguaje conciso e intuitivo. Dicha selección se puede hacer mediante los nombres de las columnas o el tipo de dato que contienen.

Selección por nombres de columnas

```
# Crear un vector con las columnas seleccionadas
columnas <- c('abilities', 'name', 'type1', 'classification', 'is_legendary')
# select()
select(pokemon, all_of(columnas))
```

```
## # A tibble: 801 x 5
##   abilities      name      type1 classification      is_legendary
##   <chr>          <chr>    <chr> <chr>              <dbl>
## 1 ['Overgrow', 'Chlorophyll'] Bulbasaur grass Seed Pokémon      0
## 2 ['Overgrow', 'Chlorophyll'] Ivysaur   grass Seed Pokémon      0
## 3 ['Overgrow', 'Chlorophyll'] Venusaur  grass Seed Pokémon      0
## 4 ['Blaze', 'Solar Power']    Charmander fire Lizard Pokémon      0
## 5 ['Blaze', 'Solar Power']    Charmeleon fire Flame Pokémon      0
## 6 ['Blaze', 'Solar Power']    Charizard fire Flame Pokémon      0
## 7 ['Torrent', 'Rain Dish']    Squirtle  water Tiny Turtle Pokémon 0
## 8 ['Torrent', 'Rain Dish']    Wartortle water Turtle Pokémon    0
## 9 ['Torrent', 'Rain Dish']    Blastoise water Shellfish Pokémon 0
## 10 ['Shield Dust', 'Run Away'] Caterpie   bug   Worm Pokémon          0
## # ... with 791 more rows
```

Seleccionar por un rango de columnas

```
select(pokemon, 10:15)
```

```
## # A tibble: 801 x 6
##   against_ghost against_grass against_ground against_ice against_normal
##   <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1           1           0.25           1             2             1
## 2           1           0.25           1             2             1
## 3           1           0.25           1             2             1
## 4           1           0.5            2             0.5           1
## 5           1           0.5            2             0.5           1
## 6           1           0.25           0             1             1
## 7           1           2             1             0.5           1
## 8           1           2             1             0.5           1
## 9           1           2             1             0.5           1
## 10          1           0.5            0.5           1             1
## # ... with 791 more rows, and 1 more variable: against_poison <dbl>
```

Si se coloca un el rango menor al final, hace una selección de manera invertida

```
select(pokemon, 20:1)
```

```
## # A tibble: 801 x 20
##   attack against_water against_steel against_rock against_psychic
##   <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1     49           0.5            1             1             2
## 2     62           0.5            1             1             2
## 3    100           0.5            1             1             2
## 4     52           2             0.5            2             1
## 5     64           2             0.5            2             1
## 6    104           2             0.5            4             1
## 7     48           0.5            0.5            1             1
## 8     63           0.5            0.5            1             1
## 9    103           0.5            0.5            1             1
## 10    30           1             1             2             1
## # ... with 791 more rows, and 15 more variables: against_poison <dbl>,
## #   against_normal <dbl>, against_ice <dbl>, against_ground <dbl>,
## #   against_grass <dbl>, against_ghost <dbl>, against_flying <dbl>,
## #   against_fire <dbl>, against_fight <dbl>, against_fairy <dbl>,
## #   against_electric <dbl>, against_dragon <dbl>, against_dark <dbl>,
```

```
## #   against_bug <dbl>, abilities <chr>
```

select() a la antigua:

```
pokemon[1:10,columnas]
```

```
## # A tibble: 10 x 5
```

	abilities	name	type1	classification	is_legendary
	<chr>	<chr>	<chr>	<chr>	<dbl>
## 1	['Overgrow', 'Chlorophyll']	Bulbasaur	grass	Seed Pokémon	0
## 2	['Overgrow', 'Chlorophyll']	Ivysaur	grass	Seed Pokémon	0
## 3	['Overgrow', 'Chlorophyll']	Venusaur	grass	Seed Pokémon	0
## 4	['Blaze', 'Solar Power']	Charmander	fire	Lizard Pokémon	0
## 5	['Blaze', 'Solar Power']	Charmeleon	fire	Flame Pokémon	0
## 6	['Blaze', 'Solar Power']	Charizard	fire	Flame Pokémon	0
## 7	['Torrent', 'Rain Dish']	Squirtle	water	Tiny Turtle Pokémon	0
## 8	['Torrent', 'Rain Dish']	Wartortle	water	Turtle Pokémon	0
## 9	['Torrent', 'Rain Dish']	Blastoise	water	Shellfish Pokémon	0
## 10	['Shield Dust', 'Run Away']	Caterpie	bug	Worm Pokémon	0

Seleccionar utilizando un patrón de caracteres:

```
# Opción 1
```

```
select(pokemon, contains("against"))
```

```
## # A tibble: 801 x 18
```

	against_bug	against_dark	against_dragon	against_electric	against_fairy
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
## 1	1	1	1	0.5	0.5
## 2	1	1	1	0.5	0.5
## 3	1	1	1	0.5	0.5
## 4	0.5	1	1	1	0.5
## 5	0.5	1	1	1	0.5
## 6	0.25	1	1	2	0.5
## 7	1	1	1	2	1
## 8	1	1	1	2	1
## 9	1	1	1	2	1
## 10	1	1	1	1	1

## # ... with 791 more rows, and 13 more variables: against\_fight <dbl>,  
 ## # against\_fire <dbl>, against\_flying <dbl>, against\_ghost <dbl>,  
 ## # against\_grass <dbl>, against\_ground <dbl>, against\_ice <dbl>,  
 ## # against\_normal <dbl>, against\_poison <dbl>, against\_psychic <dbl>,  
 ## # against\_rock <dbl>, against\_steel <dbl>, against\_water <dbl>

```
# Opción 2
```

```
select(pokemon, matches("against"))
```

```
## # A tibble: 801 x 18
```

	against_bug	against_dark	against_dragon	against_electric	against_fairy
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
## 1	1	1	1	0.5	0.5
## 2	1	1	1	0.5	0.5
## 3	1	1	1	0.5	0.5
## 4	0.5	1	1	1	0.5
## 5	0.5	1	1	1	0.5
## 6	0.25	1	1	2	0.5
## 7	1	1	1	2	1

```
## 8      1      1      1      2      1
## 9      1      1      1      2      1
## 10     1      1      1      1      1
## # ... with 791 more rows, and 13 more variables: against_fight <dbl>,
## #   against_fire <dbl>, against_flying <dbl>, against_ghost <dbl>,
## #   against_grass <dbl>, against_ground <dbl>, against_ice <dbl>,
## #   against_normal <dbl>, against_poison <dbl>, against_psychic <dbl>,
## #   against_rock <dbl>, against_steel <dbl>, against_water <dbl>
```

Por tipo de dato:

```
select(pokemon, where(is.numeric))
```

```
## # A tibble: 801 x 34
##   against_bug against_dark against_dragon against_electric against_fairy
##   <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1      1      1      1      0.5      0.5
## 2      1      1      1      0.5      0.5
## 3      1      1      1      0.5      0.5
## 4      0.5      1      1      1      0.5
## 5      0.5      1      1      1      0.5
## 6      0.25      1      1      2      0.5
## 7      1      1      1      2      1
## 8      1      1      1      2      1
## 9      1      1      1      2      1
## 10     1      1      1      1      1
## # ... with 791 more rows, and 29 more variables: against_fight <dbl>,
## #   against_fire <dbl>, against_flying <dbl>, against_ghost <dbl>,
## #   against_grass <dbl>, against_ground <dbl>, against_ice <dbl>,
## #   against_normal <dbl>, against_poison <dbl>, against_psychic <dbl>,
## #   against_rock <dbl>, against_steel <dbl>, against_water <dbl>, attack <dbl>,
## #   base_egg_steps <dbl>, base_happiness <dbl>, base_total <dbl>,
## #   defense <dbl>, experience_growth <dbl>, height_m <dbl>, hp <dbl>, ...
```

## filter()

Función que se utiliza para generar subconjuntos de datos, reteniendo las **filas** que cumplen una condición. Para hacer el filtrado, se evalúa una expresión que deber ser **TRUE** para generar las filas. Cuando existen **NAs** se eliminan.

Para generar la condición se utilizan operadores relacionales y lógicos.

### Operadores relacionales

- `>`, `<`: mayor que y menor que
- `=` mayor o igual que
- `<=` menor o igual que
- `!=` diferente de
- `==` igual a

### Operadores lógicos o booleanos

**AND (&)** TRUE and TRUE -> TRUE TRUE and FALSE -> FALSE FALSE and FALSE -> FALSE

**OR (|)** TRUE or TRUE -> TRUE TRUE or FALSE -> TRUE FALSE or FALSE -> FALSE

## NOT (!)

En R se puede utilizar la ley de Morgan

$!(x \& y) = (!x) \mid (!y)$  : Negar  $x$  y  $y$  es igual que negar  $x$  o  $y$ .  $!(x \mid y) = (!x) \& (!y)$ : Negar  $x$  o  $y$  es igual que negar  $x$  y  $y$ .

Del dataset de pokemon filtrar todos los pokemones que sean de fuego:

```
filter(pokemon, type1 == "fire")
```

```
## # A tibble: 52 x 41
##   abilities          against_bug against_dark against_dragon against_electric
##   <chr>              <dbl>      <dbl>      <dbl>      <dbl>
## 1 ['Blaze', 'Solar Po~ 0.5        1        1        1
## 2 ['Blaze', 'Solar Po~ 0.5        1        1        1
## 3 ['Blaze', 'Solar Po~ 0.25       1        1        2
## 4 ['Flash Fire', 'Dro~ 0.5        1        1        1
## 5 ['Flash Fire', 'Dro~ 0.5        1        1        1
## 6 ['Intimidate', 'Fla~ 0.5        1        1        1
## 7 ['Intimidate', 'Fla~ 0.5        1        1        1
## 8 ['Run Away', 'Flash~ 0.5        1        1        1
## 9 ['Run Away', 'Flash~ 0.5        1        1        1
## 10 ['Flame Body', 'Vit~ 0.5        1        1        1
## # ... with 42 more rows, and 36 more variables: against_fairy <dbl>,
## #   against_fight <dbl>, against_fire <dbl>, against_flying <dbl>,
## #   against_ghost <dbl>, against_grass <dbl>, against_ground <dbl>,
## #   against_ice <dbl>, against_normal <dbl>, against_poison <dbl>,
## #   against_psychic <dbl>, against_rock <dbl>, against_steel <dbl>,
## #   against_water <dbl>, attack <dbl>, base_egg_steps <dbl>,
## #   base_happiness <dbl>, base_total <dbl>, capture_rate <chr>, ...
```

Ahora, todos los pokemones que no sean de fuego:

```
filter(pokemon, type1 != "fire")
```

```
## # A tibble: 749 x 41
##   abilities          against_bug against_dark against_dragon against_electric
##   <chr>              <dbl>      <dbl>      <dbl>      <dbl>
## 1 ['Overgrow', 'Chlor~ 1        1        1        0.5
## 2 ['Overgrow', 'Chlor~ 1        1        1        0.5
## 3 ['Overgrow', 'Chlor~ 1        1        1        0.5
## 4 ['Torrent', 'Rain D~ 1        1        1        2
## 5 ['Torrent', 'Rain D~ 1        1        1        2
## 6 ['Torrent', 'Rain D~ 1        1        1        2
## 7 ['Shield Dust', 'Ru~ 1        1        1        1
## 8 ['Shed Skin']      1        1        1        1
## 9 ['Compoundeyes', 'T~ 0.5       1        1        2
## 10 ['Shield Dust', 'Ru~ 0.5       1        1        1
## # ... with 739 more rows, and 36 more variables: against_fairy <dbl>,
## #   against_fight <dbl>, against_fire <dbl>, against_flying <dbl>,
## #   against_ghost <dbl>, against_grass <dbl>, against_ground <dbl>,
## #   against_ice <dbl>, against_normal <dbl>, against_poison <dbl>,
## #   against_psychic <dbl>, against_rock <dbl>, against_steel <dbl>,
## #   against_water <dbl>, attack <dbl>, base_egg_steps <dbl>,
## #   base_happiness <dbl>, base_total <dbl>, capture_rate <chr>, ...
```

¿Cuántas categorías hay en la columna type1?

```
dplyr::count(pokemon, type1)
```

```
## # A tibble: 18 x 2
##   type1      n
##   <chr>    <int>
## 1 bug       72
## 2 dark      29
## 3 dragon    27
## 4 electric  39
## 5 fairy     18
## 6 fighting  28
## 7 fire      52
## 8 flying     3
## 9 ghost     27
## 10 grass     78
## 11 ground    32
## 12 ice       23
## 13 normal   105
## 14 poison    32
## 15 psychic   53
## 16 rock      45
## 17 steel     24
## 18 water    114
```

Existen 18 categorías, o sea, 18 tipos de pokemones. Hacer un subconjunto de datos que elija solo a los de roca, agua, pasto y fuego.

¿Usamos AND u OR?

```
# and
filter(pokemon, type1 == "rock" & type1 == "water" & type1 == "grass" & type1 == "fire")
```

```
## # A tibble: 0 x 41
## # ... with 41 variables: abilities <chr>, against_bug <dbl>,
## #   against_dark <dbl>, against_dragon <dbl>, against_electric <dbl>,
## #   against_fairy <dbl>, against_fight <dbl>, against_fire <dbl>,
## #   against_flying <dbl>, against_ghost <dbl>, against_grass <dbl>,
## #   against_ground <dbl>, against_ice <dbl>, against_normal <dbl>,
## #   against_poison <dbl>, against_psychic <dbl>, against_rock <dbl>,
## #   against_steel <dbl>, against_water <dbl>, attack <dbl>, ...
```

```
# or
filter(pokemon, type1 == "rock" | type1 == "water" | type1 == "grass" | type1 == "fire")
```

```
## # A tibble: 289 x 41
##   abilities      against_bug against_dark against_dragon against_electric
##   <chr>          <dbl>          <dbl>          <dbl>          <dbl>
## 1 ['Overgrow', 'Chlor~    1              1              1              0.5
## 2 ['Overgrow', 'Chlor~    1              1              1              0.5
## 3 ['Overgrow', 'Chlor~    1              1              1              0.5
## 4 ['Blaze', 'Solar Po~  0.5              1              1              1
## 5 ['Blaze', 'Solar Po~  0.5              1              1              1
## 6 ['Blaze', 'Solar Po~  0.25            1              1              2
## 7 ['Torrent', 'Rain D~    1              1              1              2
## 8 ['Torrent', 'Rain D~    1              1              1              2
## 9 ['Torrent', 'Rain D~    1              1              1              2
```



```
## 10 ['Flash Fire', 'Dro~          0.5          1          1          1
## # ... with 279 more rows, and 36 more variables: against_fairy <dbl>,
## #   against_fight <dbl>, against_fire <dbl>, against_flying <dbl>,
## #   against_ghost <dbl>, against_grass <dbl>, against_ground <dbl>,
## #   against_ice <dbl>, against_normal <dbl>, against_poison <dbl>,
## #   against_psychic <dbl>, against_rock <dbl>, against_steel <dbl>,
## #   against_water <dbl>, attack <dbl>, base_egg_steps <dbl>,
## #   base_happiness <dbl>, base_total <dbl>, capture_rate <chr>, ...
```

Para no hacer la expresión tan larga, podemos utilizar un operador de pertenencia:

```
tipos_pokemones <- c("rock", "water", "grass", "fire")
filter(pokemon, type1 %in% tipos_pokemones)
```

```
## # A tibble: 289 x 41
##   abilities          against_bug against_dark against_dragon against_electric
##   <chr>              <dbl>         <dbl>         <dbl>         <dbl>
## 1 ['Overgrow', 'Chlor~      1           1           1           0.5
## 2 ['Overgrow', 'Chlor~      1           1           1           0.5
## 3 ['Overgrow', 'Chlor~      1           1           1           0.5
## 4 ['Blaze', 'Solar Po~    0.5           1           1           1
## 5 ['Blaze', 'Solar Po~    0.5           1           1           1
## 6 ['Blaze', 'Solar Po~    0.25          1           1           2
## 7 ['Torrent', 'Rain D~      1           1           1           2
## 8 ['Torrent', 'Rain D~      1           1           1           2
## 9 ['Torrent', 'Rain D~      1           1           1           2
## 10 ['Flash Fire', 'Dro~    0.5           1           1           1
## # ... with 279 more rows, and 36 more variables: against_fairy <dbl>,
## #   against_fight <dbl>, against_fire <dbl>, against_flying <dbl>,
## #   against_ghost <dbl>, against_grass <dbl>, against_ground <dbl>,
## #   against_ice <dbl>, against_normal <dbl>, against_poison <dbl>,
## #   against_psychic <dbl>, against_rock <dbl>, against_steel <dbl>,
## #   against_water <dbl>, attack <dbl>, base_egg_steps <dbl>,
## #   base_happiness <dbl>, base_total <dbl>, capture_rate <chr>, ...
```

## Filtrar una variable numérica

Conocer el valor mínimo y máximo de una variable numérica.

```
summary(pokemon$weight_kg)
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##   0.10   9.00   27.30   61.38   64.80   999.90    20
```

Ahora...¿Usamos OR o AND?

```
# or
filter(pokemon, weight_kg >= 100 | weight_kg <= 300 )
```

```
## # A tibble: 781 x 41
##   abilities          against_bug against_dark against_dragon against_electric
##   <chr>              <dbl>         <dbl>         <dbl>         <dbl>
## 1 ['Overgrow', 'Chlor~      1           1           1           0.5
## 2 ['Overgrow', 'Chlor~      1           1           1           0.5
## 3 ['Overgrow', 'Chlor~      1           1           1           0.5
## 4 ['Blaze', 'Solar Po~    0.5           1           1           1
## 5 ['Blaze', 'Solar Po~    0.5           1           1           1
```

```
## 6 ['Blaze', 'Solar Po~      0.25      1      1      2
## 7 ['Torrent', 'Rain D~      1      1      1      2
## 8 ['Torrent', 'Rain D~      1      1      1      2
## 9 ['Torrent', 'Rain D~      1      1      1      2
## 10 ['Shield Dust', 'Ru~      1      1      1      1
## # ... with 771 more rows, and 36 more variables: against_fairy <dbl>,
## #   against_fight <dbl>, against_fire <dbl>, against_flying <dbl>,
## #   against_ghost <dbl>, against_grass <dbl>, against_ground <dbl>,
## #   against_ice <dbl>, against_normal <dbl>, against_poison <dbl>,
## #   against_psychic <dbl>, against_rock <dbl>, against_steel <dbl>,
## #   against_water <dbl>, attack <dbl>, base_egg_steps <dbl>,
## #   base_happiness <dbl>, base_total <dbl>, capture_rate <chr>, ...

# and
filter(pokemon, weight_kg >= 100 & weight_kg <= 300 )

## # A tibble: 97 x 41
##   abilities      against_bug against_dark against_dragon against_electric
##   <chr>          <dbl>      <dbl>      <dbl>      <dbl>
## 1 ['Overgrow', 'Chlor~      1      1      1      0.5
## 2 ['Intimidate', 'Fla~      0.5      1      1      1
## 3 ['Guts', 'No Guard'~      0.5      0.5      1      1
## 4 ['Thick Fat', 'Hydr~      1      1      1      2
## 5 ['Shell Armor', 'Sk~      1      1      1      2
## 6 ['Rock Head', 'Stur~      1      1      1      0
## 7 ['Lightningrod', 'R~      1      1      1      0
## 8 ['Lightningrod', 'R~      1      1      1      0
## 9 ['Intimidate', 'Mox~      0.5      1      1      4
## 10 ['Water Absorb', 'S~      1      1      1      2
## # ... with 87 more rows, and 36 more variables: against_fairy <dbl>,
## #   against_fight <dbl>, against_fire <dbl>, against_flying <dbl>,
## #   against_ghost <dbl>, against_grass <dbl>, against_ground <dbl>,
## #   against_ice <dbl>, against_normal <dbl>, against_poison <dbl>,
## #   against_psychic <dbl>, against_rock <dbl>, against_steel <dbl>,
## #   against_water <dbl>, attack <dbl>, base_egg_steps <dbl>,
## #   base_happiness <dbl>, base_total <dbl>, capture_rate <chr>, ...
```

Se puede utilizar `between()` para los rangos:

```
filter(pokemon, between(weight_kg, 100, 300))

## # A tibble: 97 x 41
##   abilities      against_bug against_dark against_dragon against_electric
##   <chr>          <dbl>      <dbl>      <dbl>      <dbl>
## 1 ['Overgrow', 'Chlor~      1      1      1      0.5
## 2 ['Intimidate', 'Fla~      0.5      1      1      1
## 3 ['Guts', 'No Guard'~      0.5      0.5      1      1
## 4 ['Thick Fat', 'Hydr~      1      1      1      2
## 5 ['Shell Armor', 'Sk~      1      1      1      2
## 6 ['Rock Head', 'Stur~      1      1      1      0
## 7 ['Lightningrod', 'R~      1      1      1      0
## 8 ['Lightningrod', 'R~      1      1      1      0
## 9 ['Intimidate', 'Mox~      0.5      1      1      4
## 10 ['Water Absorb', 'S~      1      1      1      2
## # ... with 87 more rows, and 36 more variables: against_fairy <dbl>,
## #   against_fight <dbl>, against_fire <dbl>, against_flying <dbl>,
```

```
## #   against_ghost <dbl>, against_grass <dbl>, against_ground <dbl>,
## #   against_ice <dbl>, against_normal <dbl>, against_poison <dbl>,
## #   against_psychic <dbl>, against_rock <dbl>, against_steel <dbl>,
## #   against_water <dbl>, attack <dbl>, base_egg_steps <dbl>,
## #   base_happiness <dbl>, base_total <dbl>, capture_rate <chr>, ...
```

## Ejercicio:

Hacer un subconjunto de datos de las columnas: nombre, tipo1, clasificación, habilidades, peso y si es legendario de los pokemones de agua, fuego, hielo y electricos.

## Solución

Ver de nuevo los nombres de las columnas:

```
colnames(pokemon)
```

```
## [1] "abilities"      "against_bug"    "against_dark"
## [4] "against_dragon" "against_electric" "against_fairy"
## [7] "against_fight"  "against_fire"    "against_flying"
## [10] "against_ghost"  "against_grass"   "against_ground"
## [13] "against_ice"    "against_normal"  "against_poison"
## [16] "against_psychic" "against_rock"    "against_steel"
## [19] "against_water"  "attack"          "base_egg_steps"
## [22] "base_happiness" "base_total"      "capture_rate"
## [25] "classification" "defense"          "experience_growth"
## [28] "height_m"       "hp"              "japanese_name"
## [31] "name"           "percentage_male" "pokedex_number"
## [34] "sp_attack"      "sp_defense"      "speed"
## [37] "type1"          "type2"           "weight_kg"
## [40] "generation"     "is_legendary"
```

Ver de nuevo los tipos:

```
table(pokemon$type1)
```

```
##
##      bug      dark  dragon electric    fairy fighting    fire    flying
##      72       29      27       39      18       28       52       3
##    ghost    grass  ground      ice   normal    poison  psychic    rock
##      27       78      32       23     105      32       53      45
##    steel    water
##      24      114
```

```
# Conocer la proporción
# prop.table(table(pokemon$type1))
```

Entonces:

```
columnas2 <- c("name", 'type1', 'classification', 'abilities', 'weight_kg', 'is_legendary')
tipos <- c('water', 'ice', 'fire', 'electric')
pokemon2 <- pokemon %>%
  select(all_of(columnas2)) %>%
  filter(type1 %in% tipos)

pokemon2
```

```
## # A tibble: 228 x 6
```

```
##   name      type1   classfication   abilities      weight_kg is_legendary
##   <chr>     <chr>   <chr>         <chr>          <dbl>      <dbl>
## 1 Charmander fire    Lizard Pokémon   ['Blaze', 'So~    8.5        0
## 2 Charmeleon fire    Flame Pokémon    ['Blaze', 'So~    19         0
## 3 Charizard  fire    Flame Pokémon    ['Blaze', 'So~    90.5       0
## 4 Squirtle   water   Tiny Turtle Pokémon ['Torrent', '~    9          0
## 5 Wartortle  water   Turtle Pokémon    ['Torrent', '~    22.5       0
## 6 Blastoise  water   Shellfish Pokémon ['Torrent', '~    85.5       0
## 7 Pikachu    electric Mouse Pokémon    ['Static', 'L~    6          0
## 8 Raichu      electric Mouse Pokémon    ['Static', 'L~    NA         0
## 9 Vulpix      fire    Fox Pokémon      ['Flash Fire'~    NA         0
## 10 Ninetales fire    Fox Pokémon      ['Flash Fire'~    NA         0
## # ... with 218 more rows
```

## arrange()

Permite ordenar el dataframe en funcion de los valores que hay en una columna. De manera predeterminada lo hace de menor a mayor.

```
# Ordena los nombres de los pokemones en orden alfabético
arrange(pokemon2, name)
```

```
## # A tibble: 228 x 6
##   name      type1   classfication   abilities      weight_kg is_legendary
##   <chr>     <chr>   <chr>         <chr>          <dbl>      <dbl>
## 1 Alomomola water    Caring Pokémon   ['Healer', 'H~    31.6        0
## 2 Ampharos  electric Light Pokémon    ['Static', 'P~    61.5        0
## 3 Araquanid water    Water Bubble Pokémon ['Water Bubbl~    82         0
## 4 Arcanine  fire    Legendary Pokémon ['Intimidate'~    155        0
## 5 Articuno  ice     Freeze Pokémon    ['Pressure', ~    55.4        1
## 6 Avalugg   ice     Iceberg Pokémon    ['Own Tempo',~    505        0
## 7 Azumarill water    Aquarabbit Pokémon ['Thick Fat',~    28.5        0
## 8 Barboach  water    Whiskers Pokémon    ['Oblivious',~    1.9         0
## 9 Basculin  water    Hostile Pokémon    ['Reckless', ~    18         0
## 10 Beartic  ice     Freezing Pokémon    ['Snow Cloak'~    260        0
## # ... with 218 more rows
```

```
# Ordenar los nombres en orden alfabético pero el peso de mayor a menor
arrange(pokemon2, name, desc(weight_kg))
```

```
## # A tibble: 228 x 6
##   name      type1   classfication   abilities      weight_kg is_legendary
##   <chr>     <chr>   <chr>         <chr>          <dbl>      <dbl>
## 1 Alomomola water    Caring Pokémon   ['Healer', 'H~    31.6        0
## 2 Ampharos  electric Light Pokémon    ['Static', 'P~    61.5        0
## 3 Araquanid water    Water Bubble Pokémon ['Water Bubbl~    82         0
## 4 Arcanine  fire    Legendary Pokémon ['Intimidate'~    155        0
## 5 Articuno  ice     Freeze Pokémon    ['Pressure', ~    55.4        1
## 6 Avalugg   ice     Iceberg Pokémon    ['Own Tempo',~    505        0
## 7 Azumarill water    Aquarabbit Pokémon ['Thick Fat',~    28.5        0
## 8 Barboach  water    Whiskers Pokémon    ['Oblivious',~    1.9         0
## 9 Basculin  water    Hostile Pokémon    ['Reckless', ~    18         0
## 10 Beartic  ice     Freezing Pokémon    ['Snow Cloak'~    260        0
## # ... with 218 more rows
```

## mutate()

Agrega nuevas variables y preserva las existentes.

Sumar el total de las variables `against`

```
# Opcion 1
pokemon %>%
  select(name, contains("against")) %>%
  mutate(total = rowSums(select(., -name))) %>%
  select(name, total)
```

```
## # A tibble: 801 x 2
##   name      total
##   <chr>     <dbl>
## 1 Bulbasaur  19.2
## 2 Ivysaur   19.2
## 3 Venusaur  19.2
## 4 Charmander 18
## 5 Charmeleon 18
## 6 Charizard 18.5
## 7 Squirtle   18
## 8 Wartortle  18
## 9 Blastoise  18
## 10 Caterpie  19.5
## # ... with 791 more rows
```

```
# opcion2
pokemon %>%
  select(name, contains("against")) %>%
  mutate(total = reduce(select(., -name), `+`))
```

```
## # A tibble: 801 x 20
##   name      against_bug against_dark against_dragon against_electric against_fairy
##   <chr>         <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1 Bulba~         1             1             1             0.5           0.5
## 2 Ivysa~         1             1             1             0.5           0.5
## 3 Venus~         1             1             1             0.5           0.5
## 4 Charm~         0.5           1             1             1             0.5
## 5 Charm~         0.5           1             1             1             0.5
## 6 Chari~         0.25          1             1             2             0.5
## 7 Squir~         1             1             1             2             1
## 8 Wartos~         1             1             1             2             1
## 9 Blast~         1             1             1             2             1
## 10 Cater~         1             1             1             1             1
## # ... with 791 more rows, and 14 more variables: against_fight <dbl>,
## #   against_fire <dbl>, against_flying <dbl>, against_ghost <dbl>,
## #   against_grass <dbl>, against_ground <dbl>, against_ice <dbl>,
## #   against_normal <dbl>, against_poison <dbl>, against_psychic <dbl>,
## #   against_rock <dbl>, against_steel <dbl>, against_water <dbl>, total <dbl>
```

## group\_by() y summarise()

Funciones que nos permiten conocer alguna medida de estadística descriptiva, a partir de las categorías de un grupo.

Ejemplo: ¿Cuál es la media del peso de los pokemones en función del tipo de pokemon?

```
pokemon2 %>%
  group_by(type1) %>%
  summarise(across(weight_kg, .fns = list(media = mean)))
```

```
## # A tibble: 4 x 2
##   type1    weight_kg_media
##   <chr>          <dbl>
## 1 electric         NA
## 2 fire             NA
## 3 ice             103.
## 4 water           51.1
```

Indica que hay NAs en nuestro dataframe.

```
pokemon2 %>%
  group_by(type1) %>%
  summarise(across(weight_kg, .fns = list(media = mean), na.rm = T))
```

```
## # A tibble: 4 x 2
##   type1    weight_kg_media
##   <chr>          <dbl>
## 1 electric        37.9
## 2 fire            66.1
## 3 ice            103.
## 4 water           51.1
```