Manipulación de datos

Objetivo de la sesión

Conocer las herramientas para la manipular mis datos

Ejercicios pendientes (favor de ir a la sala que les manda zoom)

Crea la variable X con el valor de 12

Modifica el valor de la variable X con 13

A continuación se muestran los precios de 5 productos en dólares,

precios_productos_dolar <- [12, 13, 16, 15, 0]

Crea la variable precio_productos_pesos con el precio en pesos

Imprime tu nombre 10 veces

Crea una función que reciba la calificación de un alumno (c_alumno) y la calificación mínima para aprobar la materia (c_min) devuelve el mensaje: “Aprobado” en caso pasar la calificación mínima, “No Aprobado” en caso de no tenerla.

¿Qué da como resultado 1:10 + 5, si quisiera una secuencia de 5 a 15, qué podría poner

Transformar datos

- Cargar Datos
- Conversión de datos
- Visualizarlos
- Transformarlos
- Recursos en español

Carga de datos

Abrir R scripts:

Lectura_archivos_version_base.R

Lectura_archivos_tidyverse.R

Cargar datos (tidy verse)
Ejercicios

1 ¿Qué función utilizarías para leer un archivo donde los campos están separados con “|”?

2 Además de file, skip y comment, ¿qué otros argumentos tienen en común read_csv() y read_tsv()?

3 Algunas veces las cadenas de caracteres en un archivo csv contienen comas. Para evitar que causen problemas, deben estar rodeadas por comillas, como " o '. Por convención, read_csv() asume que el caracter de separación será ".¿Qué argumentos debes especificar para leer el siguiente texto en un data frame?

"x,y\n1,'a,b"'

4 Identifica qué está mal en cada una de los siguientes archivos csv en línea (inline). ¿Qué pasa cuando corres el código?

read_csv("a,b\n1,2,3\n4,5,6")

read_csv("a,b,c\n1,2\n1,2,3,4")

read_csv("a,b\n\"1")

read_csv("a,b\n1,2\na,b")

Conversión de datos

parse_logical() y parse_integer() analizan valores lógicos y números enteros respectivamente. No hay prácticamente nada que pueda salir mal con estos segmentadores, así que no los describiremos con detalle aquí.

parse_double() es un segmentador numérico estricto, y parse_number() es un segmentador numérico flexible. Son más complicados de lo que podrías esperar debido a que los números se escriben de diferentes formas en distintas partes del mundo.

parse_character() parece tan simple que no debiera ser necesario. Pero una complicación lo hace bastante importante: la codificación de caracteres (el encoding).

parse_factor() crea factores, la estructura de datos que R usa para representar variables categóricas con valores fijos y conocidos.

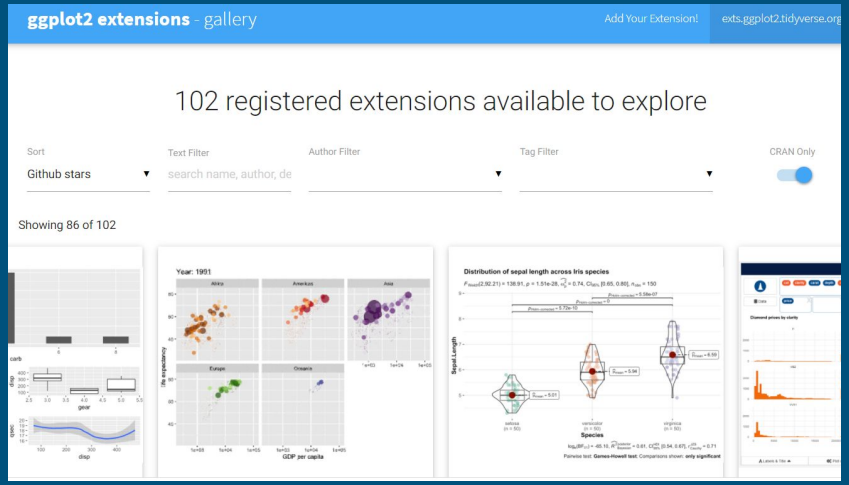
parse_datetime(), parse_date() y parse_time() te permiten analizar diversas especificaciones de fechas y horas. Estos son los más complicados, ya que hay muchas formas diferentes de escribir las fechas.

Conversión de datos

```
str(parse_logical(c("TRUE", "FALSE", "NA")))
#>  logi [1:3] TRUE FALSE NA
str(parse_integer(c("1", "2", "3")))
#>  int [1:3] 1 2 3
str(parse_date(c("2010-01-01", "1979-10-14")))
#>  Date[1:2], format: "2010-01-01" "1979-10-14"
```

Visualizarlos

```
ggplot(data = <DATOS>) +
 <GEOM_FUNCIÓN>(
   mapping = aes(<MAPEOS>),
   stat = <ESTADÍSTICAS>,
   position = <POSICIÓN>
 ) +
 <FUNCIÓN_COORDENADAS> +
 <FUNCIÓN_FACETAS>
```



https://exts.ggplot2.tidyverse.org/gallery/

Stat en gráficas

1. **`geom_bar()`** comienza con el conjunto de datos **`diamantes`**

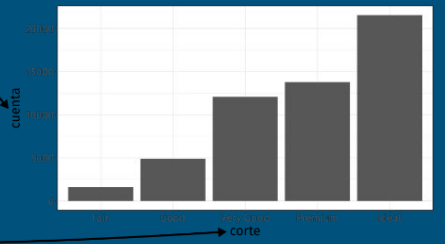
quilate	corte	color	claridad	profundidad	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Bueno	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Bueno	J	SI2	63.3	58	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...

`stat_count()`

2. **`geom_bar()`** transforma los datos con el estadístico "cuenta", que entrega un conjunto de datos de valores de corte y cuenta

corte	count	prop
Regular	1610	1
Bueno	4906	1
Muy Bueno	12082	1
Premium	13791	1
Ideal	51551	1

3. **`geom_bar()`** usa los datos transformados para construir el gráfico, `corte` se mapea al eje x, `cuenta` se mapea al eye y.

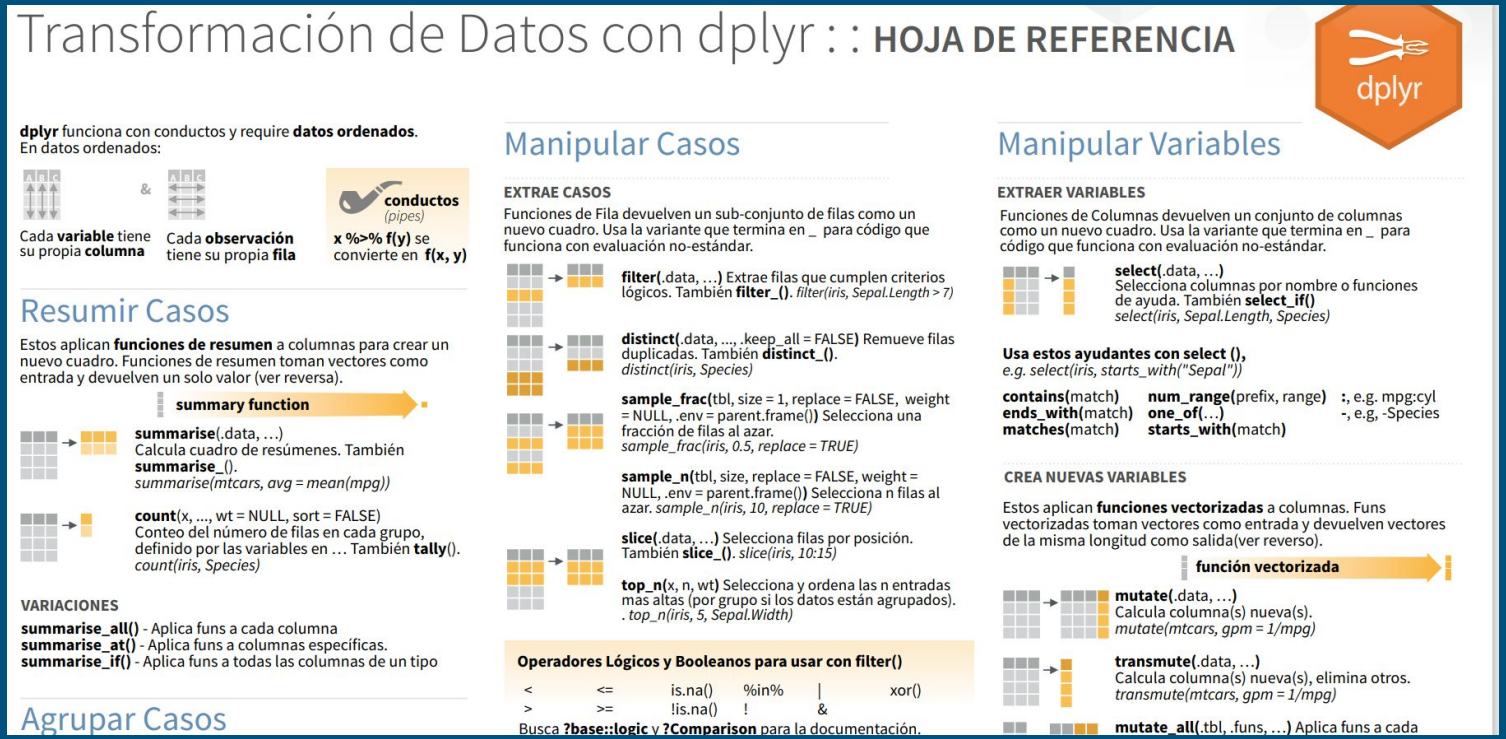


Visualizarlos

Abrir R scripts:

Visualización.R

Transformarlos



Transformación de Datos con dplyr : : HOJA DE REFERENCIA

dplyr

dplyr funciona con conductos y requiere **datos ordenados**.
En datos ordenados:

Cada **variable** tiene su propia **columna** & Cada **observación** tiene su propia **fila**

conductos *(pipes)*
x %>% f(y) se convierte en **f(x, y)**

Resumir Casos

Estos aplican **funciones de resumen** a columnas para crear un nuevo cuadro. Funciones de resumen toman vectores como entrada y devuelven un solo valor (ver reversa).

summary function

summarise(.data, …)
Calcula cuadro de resúmenes. También **summarise_**().
summarise(mtcars, avg = mean(mpg))

count(x, ..., wt = NULL, sort = FALSE)
Conteo del número de filas en cada grupo, definido por las variables en … También **tally**().
count(iris, Species)

VARIACIONES

summarise_all() - Aplica funs a cada columna
summarise_at() - Aplica funs a columnas específicas.
summarise_if() - Aplica funs a todas las columnas de un tipo

Agrupar Casos

Manipular Casos

EXTRAE CASOS

Funciones de Fila devuelven un sub-conjunto de filas como un nuevo cuadro. Usa la variante que termina en _ para código que funciona con evaluación no-estándar.

filter(.data, …) Extrae filas que cumplen criterios lógicos. También **filter_()**. *filter(iris, Sepal.Length > 7)*

distinct(.data, ..., .keep_all = FALSE) Remueve filas duplicadas. También **distinct_()**.
distinct(iris, Species)

sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame()) Selecciona una fracción de filas al azar.
sample_frac(iris, 0.5, replace = TRUE)

sample_n(tbl, size, replace = FALSE, weight = NULL, .env = parent.frame()) Selecciona n filas al azar. *sample_n(iris, 10, replace = TRUE)*

slice(.data, …) Selecciona filas por posición. También **slice_()**. *slice(iris, 10:15)*

top_n(x, n, wt) Selecciona y ordena las n entradas mas altas (por grupo si los datos están agrupados). *. top_n(iris, 5, Sepal.Width)*

Operadores Lógicos y Booleanos para usar con filter()

<	<=	is.na()	%in%	|	xor()
>	>=	!is.na()	!	&	

Busca **?base::logic** y **?Comparison** para la documentación.

Manipular Variables

EXTRAER VARIABLES

Funciones de Columnas devuelven un conjunto de columnas como un nuevo cuadro. Usa la variante que termina en _ para código que funciona con evaluación no-estándar.

select(.data, …)
Selecciona columnas por nombre o funciones de ayuda. También **select_if()**
select(iris, Sepal.Length, Species)

Usa estos ayudantes con select (),
e.g. select(iris, starts_with("Sepal"))

contains(match) **num_range**(prefix, range) **:**, e.g. mpg:cyl
ends_with(match) **one_of**(…) **-**, e.g, -Species
matches(match) **starts_with**(match)

CREA NUEVAS VARIABLES

Estos aplican **funciones vectorizadas** a columnas. Funs vectorizadas toman vectores como entrada y devuelven vectores de la misma longitud como salida(ver reverso).

función vectorizada

mutate(.data, …)
Calcula columna(s) nueva(s).
mutate(mtcars, gpm = 1/mpg)

transmute(.data, …)
Calcula columna(s) nueva(s), elimina otros.
transmute(mtcars, gpm = 1/mpg)

mutate_all(.tbl, .funs, …) Aplica funs a cada

https://github.com/rstudio/cheatsheets/raw/master/translations/spanish/data-transformation_es.pdf

Transformarlos

Abrir R scripts:

Manipulación.R

Recursos en Español

- Apply functions with purrr translated by Carlos Ortega of the Grupo de Usuarios de R de Madrid. Updated October 2017.
- caret translated by Carlos Ortega of the Grupo de Usuarios de R de Madrid. Updated September 2017.
- Data import translated by Yanina Bellini Saibene. Updated December 2019.
- Data science in Spark with sparklyr translated by DaniPrina. Updated December 2019.
- Data transformation with dplyr translated by Frans van Dunné of ixpantia. Updated March 2018.
- Data visualization with ggplot2 translated by Carolina Mengoni Goñalons. Updated December 2019.
- Dates and times with lubridate translated by Yanina Bellini Saibene. Updated December 2019.
- Deep learning with Keras translated by Carlos Ortega of the Grupo de Usuarios de R de Madrid. Updated January 2018.
- Dynamic documents with rmarkdown translated by Jessica Formoso. Updated January 2020.
- Entorno de desarrollo RStudio translated by Monica Alonso. Updated December 2019.
- Estadística descriptiva con R translated by Rosana Ferraro of Máxima Formación. Updated June 2018.
- Factors with forcats translated by Laura Acion. Updated January 2020.
- Interactive web apps with shiny translated by Frans van Dunné of ixpantia. Updated May 2018.
- Introducción a R translated by Rosana Ferraro of Máxima Formación. Updated June 2018.
- Package development with devtools translated by Paola Corrales. Updated December 2019.
- Python with R and reticulate translated by Vanesa Maribel. Updated January 2020.
- String manipulation with stringr translated by Carlos Ortega of the Grupo de Usuarios de R de Madrid (updated by L. Paloma Rojas Saunero). Updated December 2019.
- survminer translated by Maria Dermit. Updated March 2021.
- Syntax comparison translated by Riva Quiroga. Updated June 2020.