Datos ordenados y fechas

Ronny A. Hernández Mora.

@RonnyHdezM
ronnyhdez
http://ronnyhdez.rbind.io/

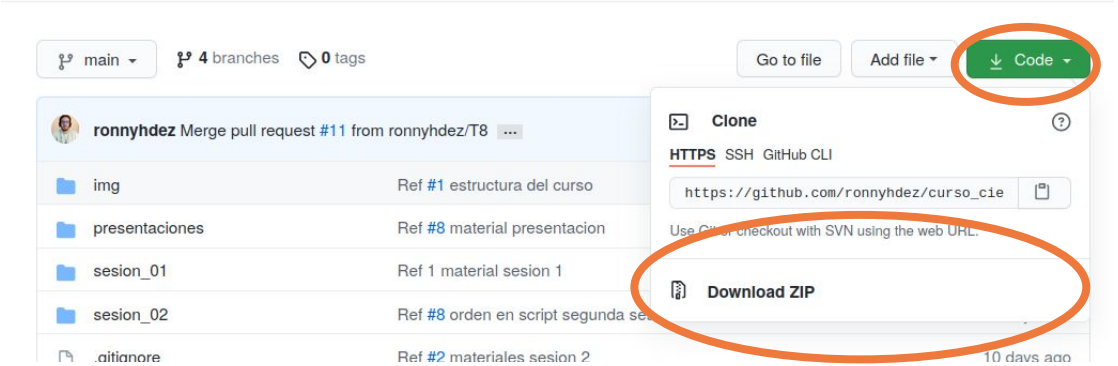
¡Grabar!

Sesión	Temas	Horario
1. Introducción a R	- ¿Qué es R? - ¿Qué es RStudio? - Sintaxis R	Miércoles de 19:00 a 21:00 26 de mayo 2021
2. Estructuras de datos en R	- Clases - Estructuras de datos - Coerción	Miércoles de 19:00 a 21:00 2 de junio 2021
3. Introducción a Rmarkdown	- ¿Qué es markdown? - Estructura de archivos .Rmd - Informes -	Miércoles de 19:00 a 21:00 9 de junio 2021
4. Manejo de datos con dplyr	- Importar datos - Funciones básicas de dplyr	Miércoles de 19:00 a 21:00 16 de junio 2021
5. Manejo de datos con tidyr y fechas con lubridate	- Concepto de tidydata - Funciones en tidyr - Fechas en R con lubridate	Miércoles de 19:00 a 21:00 23 de junio 2021
6. Visualización de datos con ggplot2	- Gramática de gráficos - Funciones de ggplot2	Miércoles de 19:00 a 21:00 30 de junio 2021
7. Presentaciones de proyectos	- Estudiantes presentan un análisis de datos en un informe construido con Rmarkdown.	Miércoles de 19:00 a 21:00 7 de julio 2021

Materiales del curso

GitHub

https://github.com/ronnyhdez/curso_ciencia_datos_r

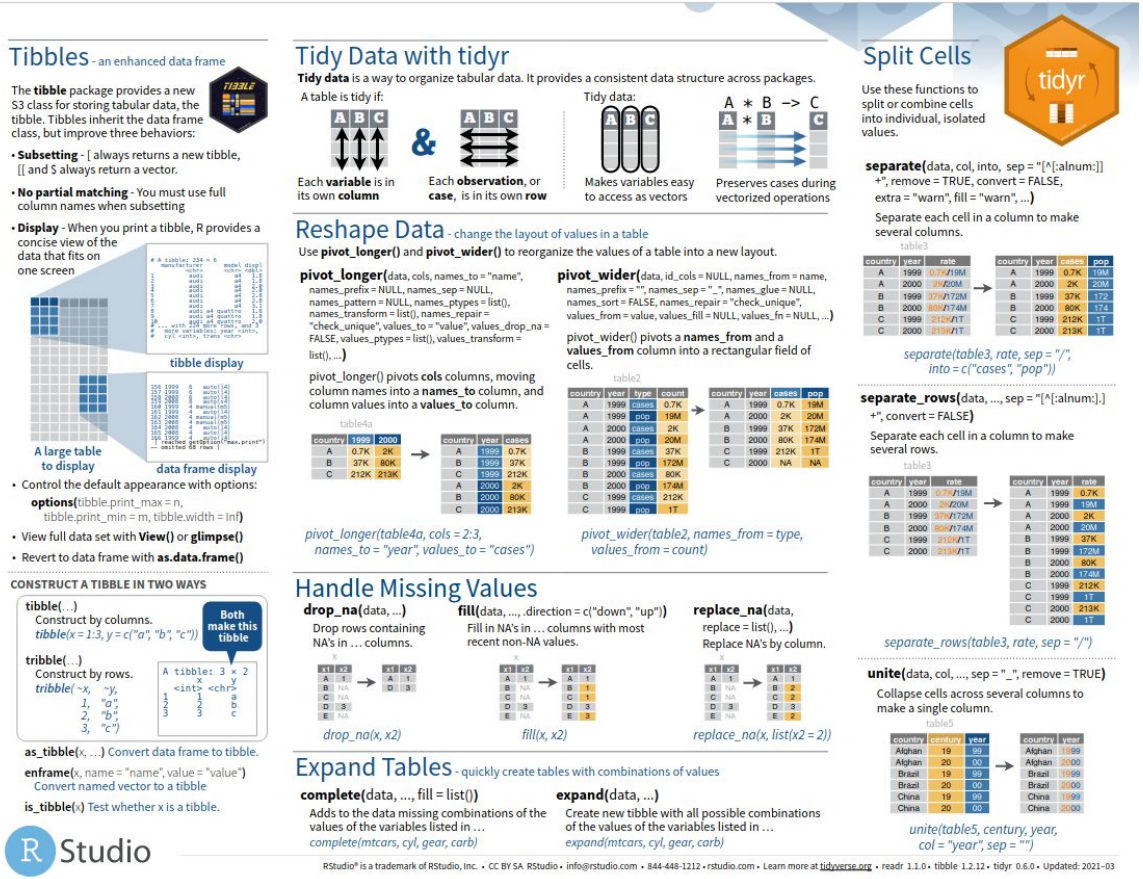


¿Qué queremos de la sesión de hoy?

- Entender qué son datos ordenados
- Saber usar funciones de tidyr para ordenar datos
- Comprender el uso de fechas

Datos ordenados

Hoja de referencia



https://www.rstudio.com/resources/cheatsheets/

¿Qué son datos ordenados?

curso	fecha	calificacion	estudiantes
matematica	2020-05-28	excelente	34
historia del arte	2020-06-04	regular	20
computacion	2020-06-12	bueno	28

cada **fila** es una observación

cada **columna** es una variable

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

¿Estos son datos ordenados?

curso	fecha	calificacion	estudiantes/respuestas
matematica	2020-05-28	excelente	34/20
historia del arte	2020-06-04	regular	20/18
computacion	2020-06-12	bueno	28/12

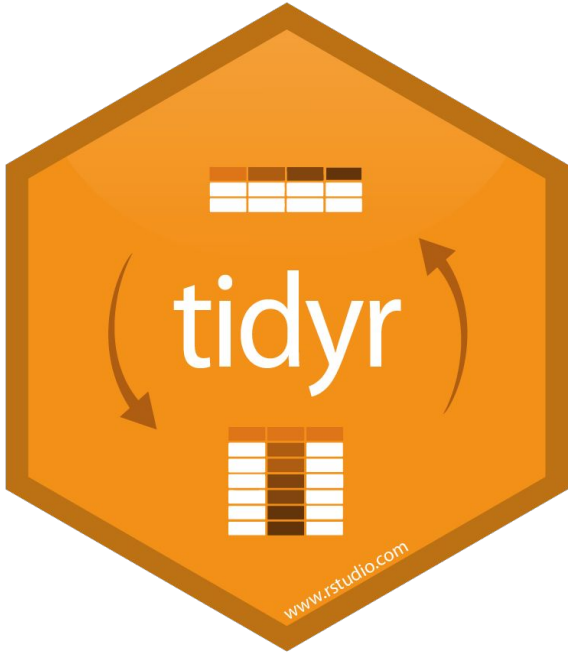
¿Estos son datos ordenados?

curso	fecha	calificacion	tipo	total
matematica	2020-05-28	excelente	estudiantes	34
matematica	2020-05-28	excelente	respuestas	20
historia del arte	2020-06-04	regular	estudiantes	20
historia del arte	2020-06-04	regular	respuestas	18
computacion	2020-06-12	bueno	estudiantes	28
computacion	2020-06-12	bueno	respuestas	12

¿Cómo llevar datos de un archivo a mi sesión de R?

```
pivot_longer( argumentos )
```

```
pivot_wider( argumentos )
```



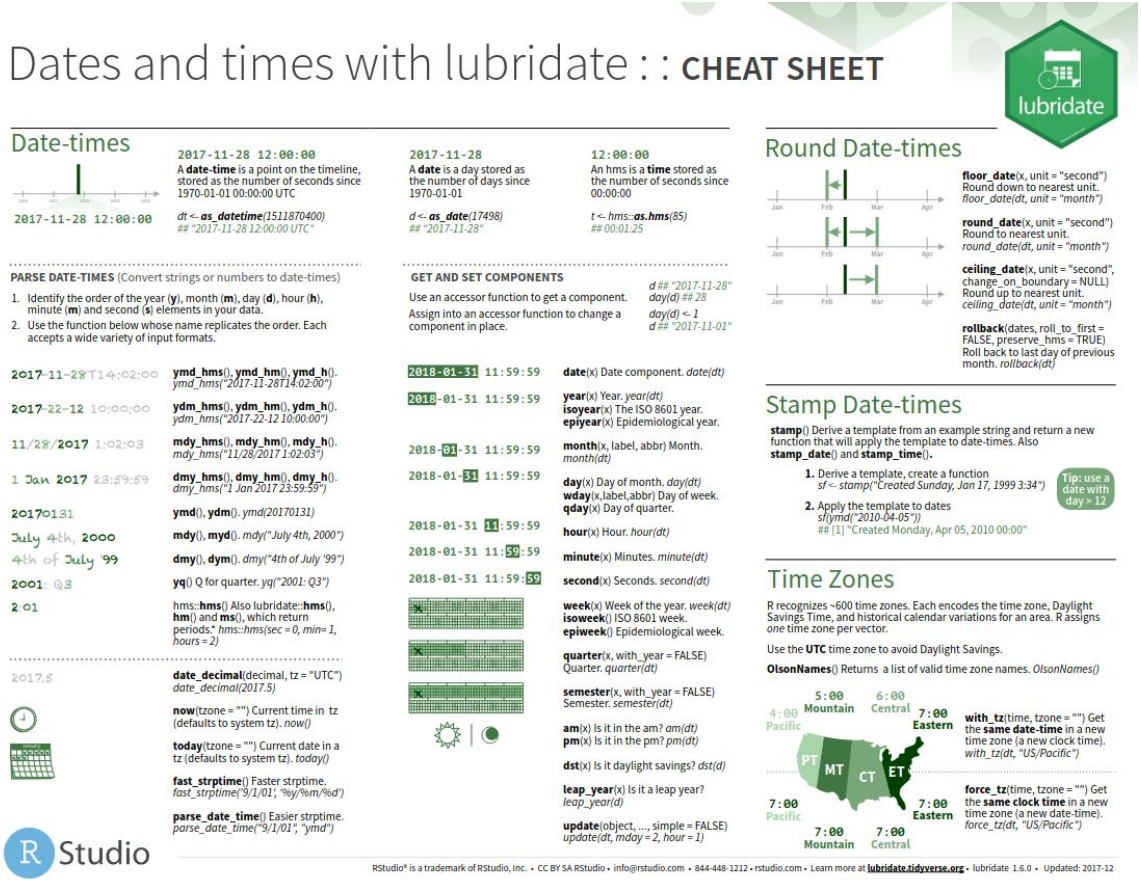
https://tidyr.tidyverse.org

https://www.garrickadenbuie.com/project/tidyexplain/

Café 20:00 a 20:10

Fechas con lubridate

Hoja de referencia



Dates and times with lubridate : : **CHEAT SHEET**

lubridate

Date-times

2017-11-28 12:00:00

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

GET AND SET COMPONENTS

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

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

2018-01-31 11:59:59

date(x) Date component. *date(dt)*

2018-01-31 11:59:59

year(x) Year. *year(dt)*
isoyear(x) The ISO 8601 year.
epiyear(x) Epidemiological year.

2018-01-31 11:59:59

month(x, label, abbr) Month. *month(dt)*

2018-01-31 11:59:59

day(x) Day of month. *day(dt)*
wday(x,label,abbr) Day of week.
qday(x) Day of quarter.

2018-01-31 11:59:59

hour(x) Hour. *hour(dt)*

2018-01-31 11:59:59

minute(x) Minutes. *minute(dt)*

2018-01-31 11:59:59

second(x) Seconds. *second(dt)*

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

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

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

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

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

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

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

Round Date-times

Jan Feb Mar Apr

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

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

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

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

Stamp Date-times

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

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

Tip: use a date with day > 12

Time Zones

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

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

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

4:00 Pacific — 5:00 Mountain — 6:00 Central — 7:00 Eastern
PT MT CT ET
7:00 Pacific — 7:00 Mountain — 7:00 Central — 7:00 Eastern

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

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

RStudio

https://www.rstudio.com/resources/cheatsheets/

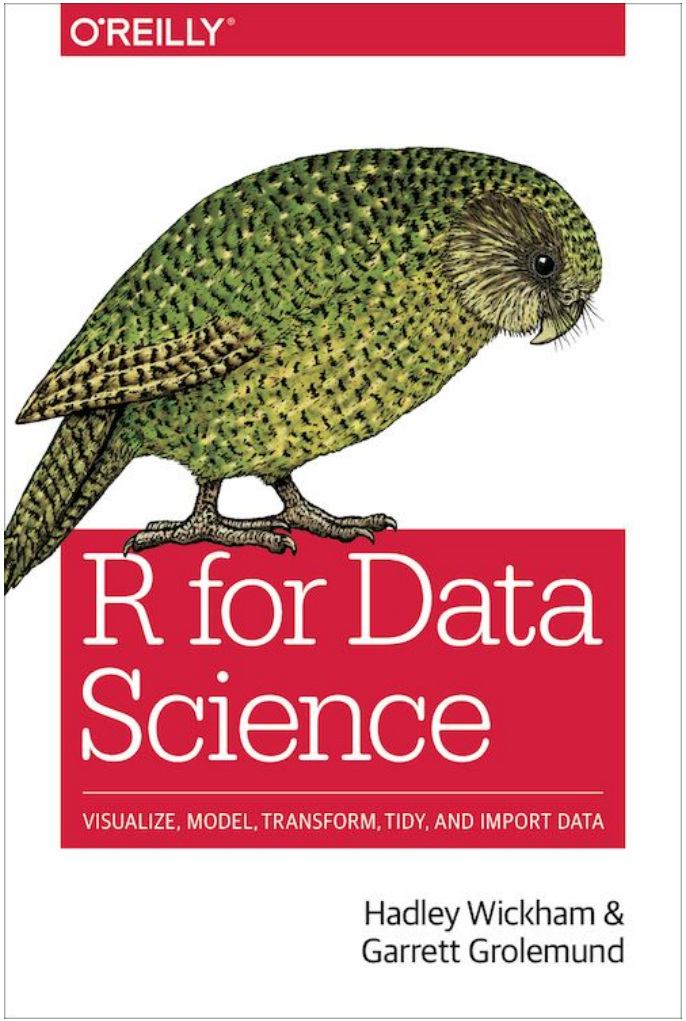
Proyecto

Exploración de datos abiertos de la UNA sobre evaluación del desempeño

https://www.transparencia.una.ac.cr/index.php?option=com_content&view=article&id=333&Itemid=787

Recursos


Programación en C++
Algoritmos, estructuras de datos y objetos
L. Joyanes Aguilar
Learning Python
FIFTH EDITION
LINUX BASICS FOR HACKERS
Fundamentals of Data Visualization
Wilke
Mastering Ubuntu Server
Third Edition
Jay LaCroix
R for Data Science
O'REILLY
THINK LIKE A PROGRAMMER
The Pragmatic Programmer
20th ANNIVERSARY EDITION
Zuur • Ieno • Walker
Saveliev • Smith
Mixed Effects Models
Extensions in Ecology with R
Advanced R
Wickham
Raspberry Pi 3
Model B+
E1385



https://es.r4ds.hadley.nz/

¡Gracias!

Ronny A. Hernández Mora.

@RonnyHdezM
ronnyhdez
http://ronnyhdez.rbind.io/

ronny.hernandezm@gmail.com