

Exploring data #1

Tidyverse and cheatsheets

The “tidyverse”

So far, we have used a number of packages that are part of the *tidyverse*. The tidyverse is a collection of recent and developing packages for R, many written by Hadley Wickham.



The “tidyverse”



"A giant among data nerds"

<https://priceconomics.com/hadley-wickham-the-man-who-revolutionized-r/>

RStudio has several very helpful **cheatsheets**. These are one-page sheets (front and back) that cover many of the main functions for a certain topic or task in R. These cheatsheets cover a lot of the main “tidyverse” functions.

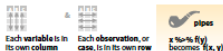
You can access these directly from RStudio. Go to “Help” -> “Cheatsheets” and select the cheatsheet on the topic of interest.

You can find even more of these cheatsheets at <https://www.rstudio.com/resources/cheatsheets/>.

Data Transformation with dplyr : CHEAT SHEET

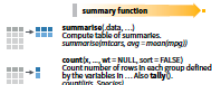


dplyr functions work with pipes and expect tidy data. In tidy data:



Summarise Cases

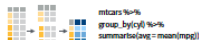
These apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).



VARIATIONS
summarise_all() - Apply funs to every column.
summarise_at() - Apply funs to specific columns.
summarise_if() - Apply funs to all cols of one type.

Group Cases

Use **group_by()** to create a "grouped" copy of a table. dplyr functions will manipulate each "group" separately and then combine the results.



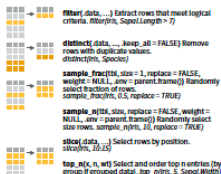
group_by(data, ..., add = FALSE)
 Returns copy of table grouped by ...
 group_by(mtcars, Species)

ungroup(x, ...)
 Returns ungrouped copy of table.
 ungroup(g_iris)

Manipulate Cases

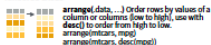
EXTRACT CASES

Row functions return a subset of rows as a new table.

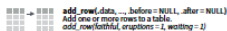


Logical and boolean operators to use with filter()
 < <= is.na() %in% | xor()
 > >= !is.na() ! %&
 See ?base::logic and ?Comparison for help.

ARRANGE CASES



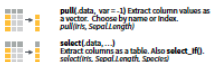
ADD CASES



Manipulate Variables

EXTRACT VARIABLES

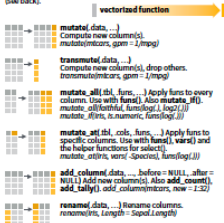
Column functions return a set of columns as a new vector or table.



Use these helpers with 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)

MAKE NEW VARIABLES

These apply **vectorized functions** to columns. Vectorized fns take vectors as input and return vectors of the same length as output (see back).



If you would like more reading and practice on what we've covered so far on transforming data, see chapter 5 of the “R for Data Science” book suggested at the start of the course.

As a reminder, that is available at:

<http://r4ds.had.co.nz>