

Intro to the Tidyverse



“The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.”

tibble: dataframes... but a bit different

“Keeping what time has proven to be effective, and throwing out what is not”.

Key points:

- Displays data differently in console window
- Allows non-syntactic column names
- No partial matching of column names
- Always returns a tibble when subsetting



Pipes %>%

- Let you pass an intermediate result onto the next function

Alternatives:

- Intermediate steps
- Overwrite original
- Sandwich functions

Benefits of pipes:

- Avoids nested function calls
- Minimises need for local variables
- Easy to read

Avoid when:

- Manipulating multiple objects
- There are meaningful intermediate objects

readr + readxl + haven: data import

- Create tibbles instead of dataframes
- Alternatives: base R and data.table

Key points:

- Consistent naming scheme for functions + parameters
- Faster than base, slower than data.table::fread()
- Guesses column types and converts where appropriate (but does NOT automatically convert strings to factors)
- Automatically parses common date/time formats



lubridate: date-time data

- Makes basic date-time manipulations straightforward
- Works with wide range of object classes



stringr: string manipulation

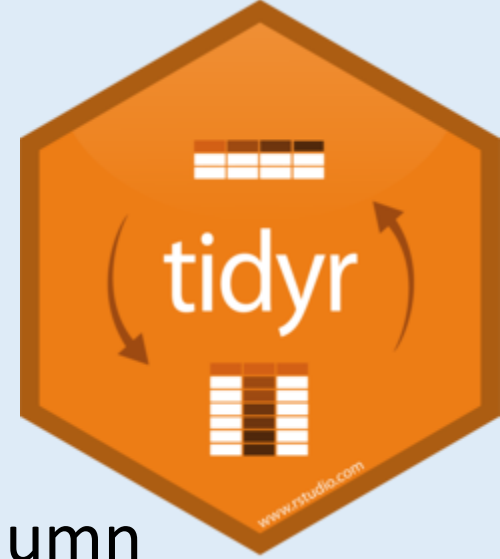
Advantages:

- Consistent naming scheme – str_ prefix
- Intuitively named functions – e.g. str_replace_all() vs. gsub(), str_length() vs. nchar()



tidyr: tidy data

- Replacement for reshape and reshape2
- Easy creation of “tidy” data – i.e. each variable in its own column and each observation in its own row
- 2 main functions: `gather()` and `spread()` → now updated to `pivot_longer()` and `pivot_wider()`
- Alternative: `dcast()` and `melt()` from `data.table`

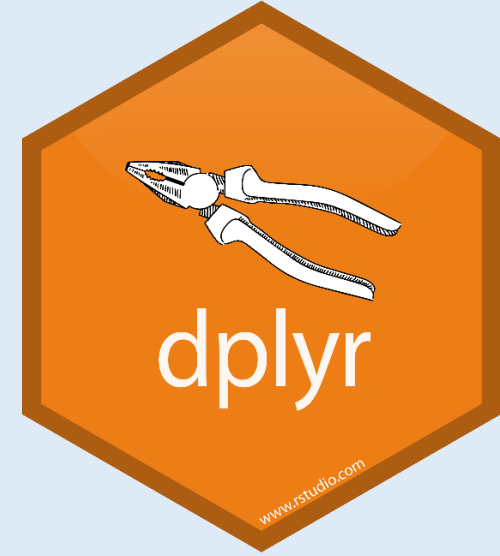


dplyr: data manipulation

- Reduced use of [,] and \$ indexing
- Syntax simplicity and readability

Primary functions:

- `select()`: select columns
- `filter()`: to rows which satisfy certain conditions
- `mutate()`: add a new variable
- `summarise()`
- `arrange()`: change order of rows



googledrive: interact with files on Google Drive

- Upload
- Share
- Download



purrr: functional programming

- Similar use to `apply()`
- Designed to improve readability
- Slightly slower(?)



forcats: working with categorical factors

- Modify factor order
- Modify factor levels



Useful resources

- R for Data Science, Hadley Wickham: <https://r4ds.had.co.nz/>
- Cheat sheets (will put in folder on Teams)
- Transitioning into the Tidyverse tutorial:
<http://www.rebeccabarter.com/blog/2019-08-05-base-r-to-tidyverse/> [part 1]
<http://www.rebeccabarter.com/blog/2019-08-05-base-r-to-tidyverse-pt2/> [part 2]
- Master the Tidyverse course: <https://github.com/rstudio/master-the-tidyverse>

Thanks for listening!

- Any questions email me on hrisser@ceh.ac.uk or post them on the Teams group 😊