Stats Club

DPLYR WORKSHOP!

Step one: Load the dplyr library!

Dplyr is a package in R that makes dataframe manipulation super simple! It is part of a collection of packages called the **tidyverse.**

You can install and load the package from CRAN with the usual:

- install.packages(“dplyr”)
- library(dplyr)

Or, you can install all the tidyverse packages at once using:

- install.packages(“tidyverse”)
- library(“tidyverse”)

Data frames vs. tibbles and the pipe function

- When using dplyr, data frames will be automatically converted to "**tibbles**" (no that's not a misspelling of table!). You don't have to worry too much about converting between tibbles and data frames -- this happens automatically!
  - When you print a tibble in R, only the first 10 rows and columns that fit will print by default, making tibbles easier to view in the Console directly
  - Also: subsetting a tibble with a single bracket always returns another tibble, subsetting with double brackets produces a vector (vs. needing drop=FALSE for data frames)

- Another feature of dplyr is the pipe function: %>%
  - The pipe allows you to take output from one function and pipe it in as the first argument in the next function, allowing you to create a clear data flow. For example, these two lines do the same thing (note: you can use the pipe with any functions, including those in base R!):
    - iris %>% subset(iris$Species=="setosa") %>% na.omit()
    - na.omit(subset(iris, iris$Species=="setosa"))

There are five main “verbs”

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

The best part about dplyr functions: you can reference variable names directly and don’t have to always use the $ operator!

Example: filter(iris, Petal.Length<1.5) vs. subset(iris, iris$Petal.Length < 1.5)

Other functions that get used a lot:

- group_by() allows you to group the data by specific variables or sets of variables
- n() counts number of values/rows, n_distinct() counts number of distinct values
- left_join(), right_join(), inner_join(), full_join()
  - allow you to combine two datasets in various ways
- Cheatsheet: https://github.com/rstudio/cheatsheets/blob/master/data-transformation.pdf