dplyr

Grinnell College

February 23, 2024

`dplyr` Package in R

The `dplyr` package in R provides a cohesive framework for working with and manipulating data. The primary functions are presented as verbs:

- Filtering
- Mutating
- Summarizing
- Grouping

Filtering

The process of *filtering* allows us to create subsets of our data based on *logical conditions*:

- Filter data to include only men older than 35
- Filter data to only include those in treatment group
- Filter data to include those with an average BMI greater than 25 and who have family history of cardiac disease

Mutating

The process of *mutating* allows us to create or modify existing variables. We may use mutate if we wish to:

- Standardize quantiative variables
- Change values of categorical from 0/1 to "no"/"yes"
- Take the sum or ratio of multiple variables
- Create a new categorical variable out of an existing quantitative variable

Summarizing

Summarizing data is essentially the process of computing *statistics*; this will condense multiple rows into a single (or several) row with a *summary*:

- Create mean value of variable in dataset
- Create mean value of variable *by group*
- Determine frequency of observations in group
- Essentially compute any other statistic

Grouping

The act of *grouping* our data involves no visible changes, but rather creates a set of internal tags on which rows are a part of which group. This is typically used in conjunction with mutating or summarizing

- Always based on a categorical variable
- Can create groups with one or more categoricals
- Often used in conjunction with `summarize()` or `mutate()`

Examples

We use the pipe operator (%>%)

```
# Fictional data
data %>% filter(Sex == "Male", Age > 35) %>%
  mutate(bmi = weight / height^2) %>%
  group_by(Treatment) %>%
  summarize(meanBMI = mean(bmi),
            sdBMI = sd(bmi))
```