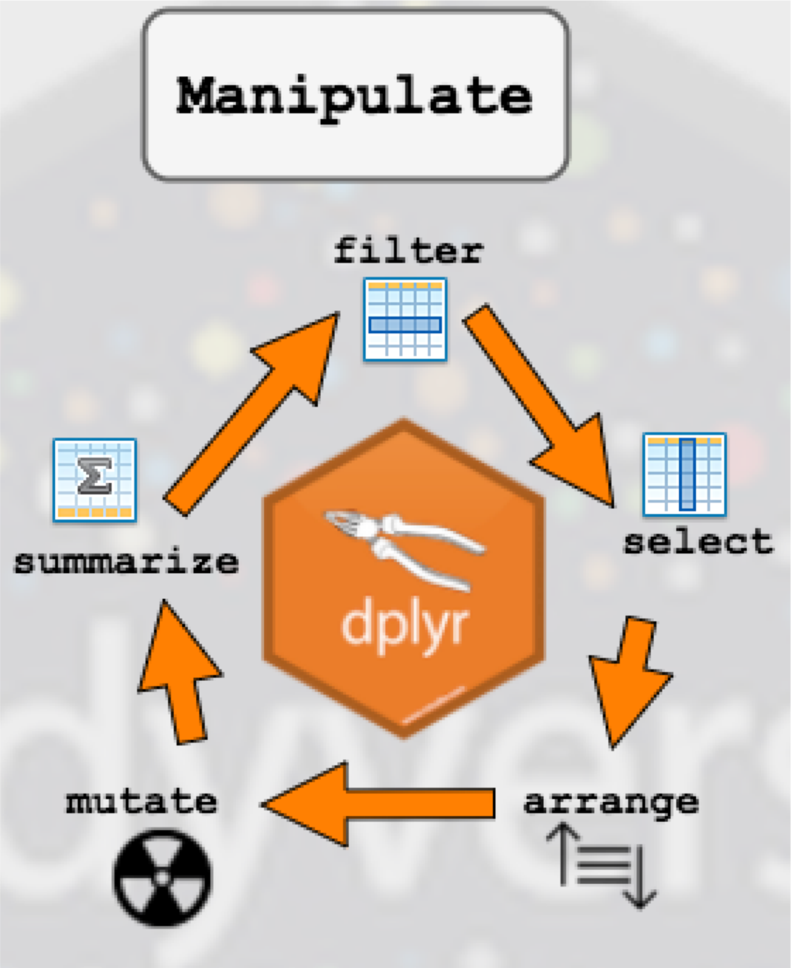
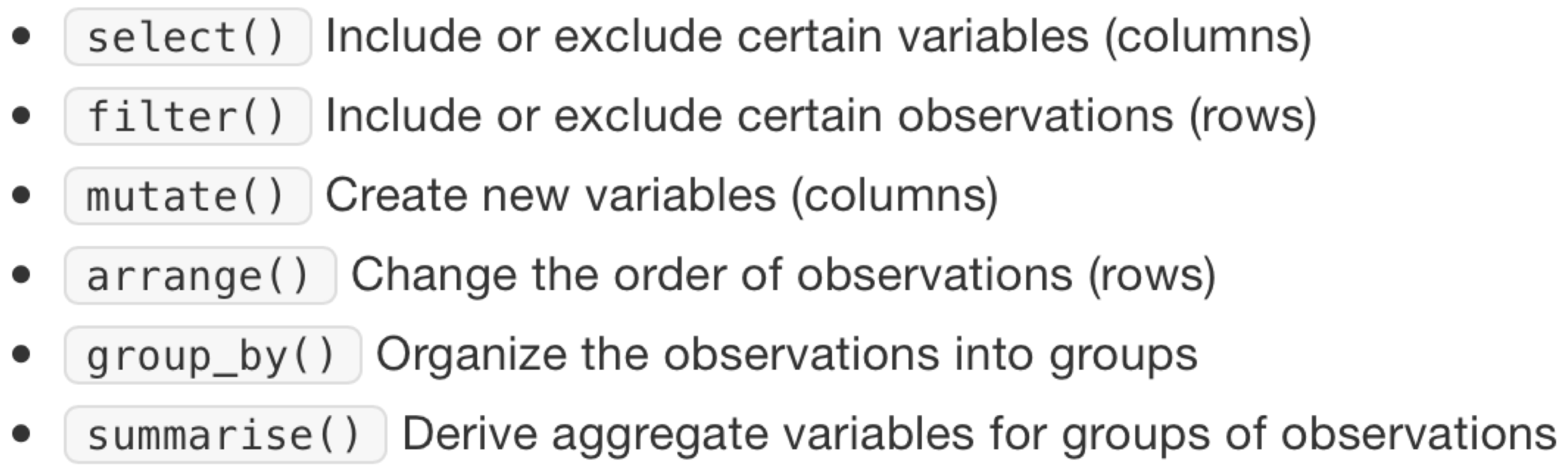
Data wrangling: dplyr and pipes

Dr. Margriet A. Groen

dplyr



dplyr: 6 main functions

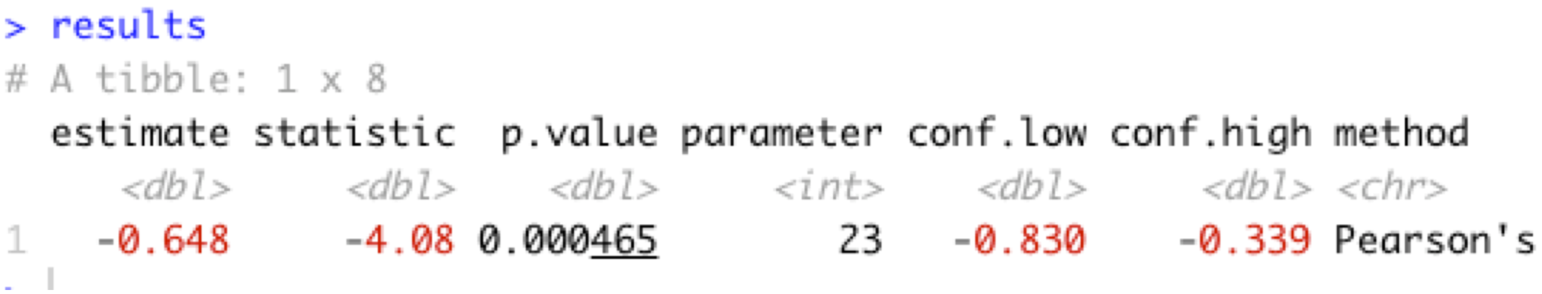


- `select()` Include or exclude certain variables (columns)
- `filter()` Include or exclude certain observations (rows)
- `mutate()` Create new variables (columns)
- `arrange()` Change the order of observations (rows)
- `group_by()` Organize the observations into groups
- `summarise()` Derive aggregate variables for groups of observations

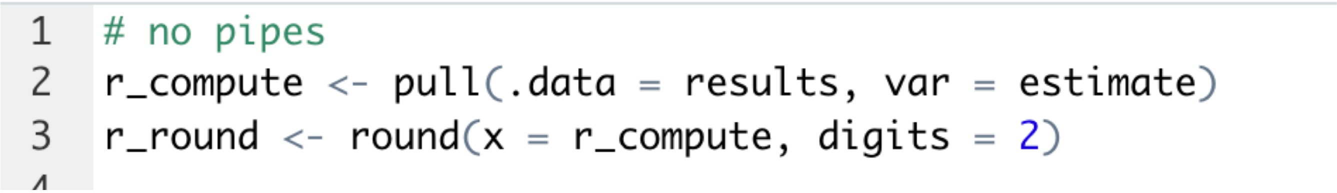
Pipes: basics

- The ‘pipe’ is written as %>%
- You should read it as ‘and then’.
- It allow you to string together ‘sentences’ of code into ‘paragraphs’ so that you don’t have to create intermediary objects.
- It ‘pipes’ the data through to the next function.

Pipes: example



```
> results
# A tibble: 1 x 8
  estimate statistic  p.value parameter conf.low conf.high method
     <dbl>     <dbl>    <dbl>     <int>    <dbl>     <dbl> <chr>
1   -0.648     -4.08 0.000465        23   -0.830    -0.339 Pearson's
```



```
# no pipes
r_compute <- pull(.data = results, var = estimate)
r_round <- round(x = r_compute, digits = 2)
```