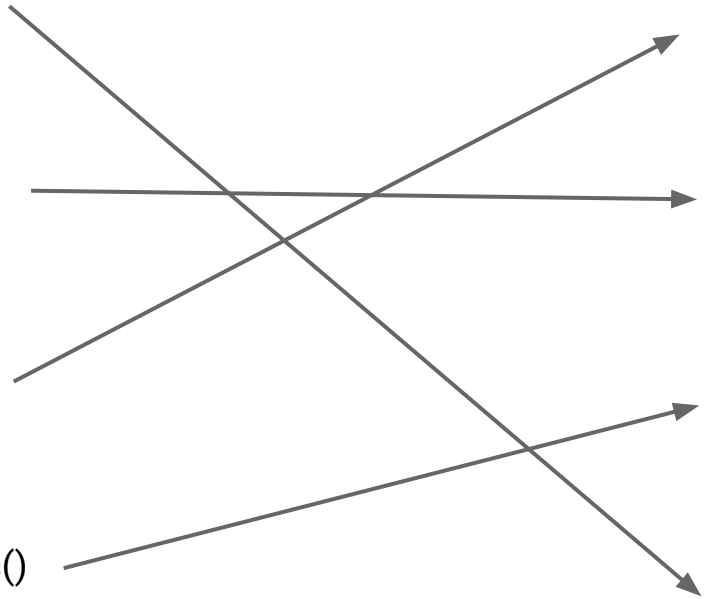
Review of dplyr functions

Naomi Wilcox
9/5/2018

Uses of common functions

1. str()
2. head()
3. dim()
4. names()



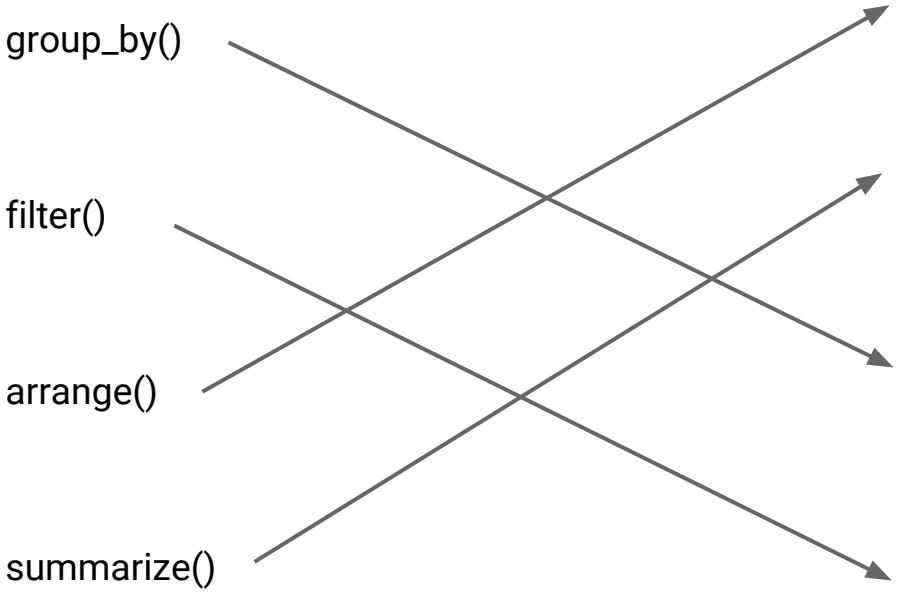
Retrieve or set the dimension of an object. (*m* x *n* = rows x columns)

Returns the first parts of a vector, matrix, table, data frame or function.

Functions to get or set the variable names of an object.

Compactly Displays the Structure of an Arbitrary R Object

Uses of dplyr functions



Order rows by values of a column (low to high).

Summarize Scalars Or Matrices By Cross-Classification

takes an existing tbl and converts it into a grouped tbl where operations are performed "by group".

Return Rows With Matching Conditions

Additional useful functions

View(dataset)

- View data set in spreadsheet-like display (note capital V).

%>%

- Pipe
- Passes object on left hand side as first argument (or . argument) of function on right hand side.
- "Piping" with %>% makes code more readable

Example:

```
iris %>%
    group_by(Species)
    %>% summarise(avg = mean(Sepal.Width))
    %>% arrange(avg)
```

mutate()

- Compute and append one or more new columns.

General syntax:

dataset.name <- dataset.name %>% mutate(name_new_variable = name_original_variable operation)

Where the operation can be any numeric function, such as: *100

Note:

If you use dataset.name <- dataset.name you are writing over the original dataset and adding the new variable as a column

If you want to keep the original (raw) dataset untouched, you can create a new dataset.

ex) new.dataset.name <- original.dataset.name %>% mutate(name_new_variable = name_original_variable operation)

rename()

- Renames variable names in the dataset (i.e. column names)

General syntax:

```
new.dataset.name <- old.dataset.name %>% rename(name_new_variable1 = `name original variable1`,
                                                name_new_variable2 = `name original variable2`)
```

Note: A general syntax rule to follow in R is that variable names should not have spaces. To avoid this, it is common to use an underscore _ in the place of a space. Remember to use backticks when telling R the name of a variable with spaces.

Select & mutate example

How would you go from the first dataset to the second?

len	supp	dose
4.2	VC	0.5
11.5	VC	0.5
7.3	VC	0.5
5.8	VC	0.5
6.4	VC	0.5
10.0	VC	0.5
11.2	VC	0.5
11.2	VC	0.5
5.2	VC	0.5
7.0	VC	0.5
16.5	VC	1.0
16.5	VC	1.0
15.2	VC	1.0
17.3	VC	1.0

len	supp	len_rounded
4.2	VC	4
11.5	VC	11
7.3	VC	7
5.8	VC	5
6.4	VC	6
10.0	VC	10
11.2	VC	11
11.2	VC	11
5.2	VC	5
7.0	VC	7

Select & mutate example

```r
# Load library
library(dplyr)
# Look at raw dataset
View(ToothGrowth)

# Use subset() to create a subset of the original dataset with only
#the "len" and "supp" variables
ToothGrowth2 <- subset(ToothGrowth, select=c(len, supp))
# Subset ToothGrowth2 and include only observations/rows 1-10
ToothGrowth3 <- ToothGrowth2[1:10,]

# Mutate ToothGrowth3, create a new variable len_rounded
ToothGrowth4 <- ToothGrowth3 %>% mutate(len_rounded = floor(len))
```

Note: Here, floor() is a function that rounds a variable down to the nearest whole number. c means “combine”

Common syntax errors

- **‘could not find function’ Error**
- This error arises when an R package is not loaded properly or due to a function being misspelled.
- Fix: include the line – library(package_name) at the start of the code
- **‘object not found’ Error**
- This error occurs when the particular object used in the code is empty.
- Fix: check your Environment to make sure you’ve created the object you’re trying to call.
- **‘Error in eval’ Error**
- This error is caused by references to objects that don't exist
- Fix: check your Environment to make sure you’ve created the object you’re trying to call. Also check for misplaced or missing parentheses or commas

Resources

- Tidyverse style guide

http://style.tidyverse.org/

- Dplyr cheat sheet

https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

- ToothGrowth dataset documentation

https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/ToothGrowth.html

- See Edie’s extra resources

https://github.com/palautatan/ph142