Manipulating Data in R

John Muschelli

January 4, 2016

Overview

In this module, we will show you how to:

1. Reshaping data from long (tall) to wide (fat)
2. Reshaping data from wide (fat) to long (tall)
3. Merging Data
4. Perform operations by a grouping variable

Setup

We will show you how to do each operation in base R then show you how to use the dplyr or tidyr package to do the same operation (if applicable).

See the "Data Wrangling Cheat Sheet using dplyr and tidyr":

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

Load the packages/libraries

Reshaping data from wide (fat) to long (tall)

Resources

See `http://www.cookbook-r.com/Manipulating_data/`
`Converting_data_between_wide_and_long_format/`

Reshaping data from wide (fat) to long (tall): base R

The `reshape` command exists. It is a **confusing** function. Don't use it.

Reshaping data from wide (fat) to long (tall): tidyr

In `tidyr`, the `gather` function gathers columns into rows.

We want the column names into "var" variable in the output dataset and the value in "number" variable. We then describe which columns we want to "gather:"

```
long = gather(ex_data, "var", "number",
              starts_with("orange"),
              starts_with("purple"), starts_with("green"),
              starts_with("banner"))
```

Reshaping data from long (tall) to wide (fat)

Reshaping data from long (tall) to wide (fat): tidyr

In tidyr, the spread function spreads rows into columns. Now we have a long data set, but we want to separate the Average, Alightings and Boardings into different columns:

```
# have to remove missing days
wide = filter(long, !is.na(date))
wide = spread(wide, type, number)
head(wide)
```

```
     day       date   line Alightings Average Boardings
1 Friday 2010-01-15 banner         NA      NA        NA
2 Friday 2010-01-15  green         NA      NA        NA
3 Friday 2010-01-15 orange       1643    1644      1645
4 Friday 2010-01-15 purple         NA      NA        NA
5 Friday 2010-01-22 banner         NA      NA        NA
6 Friday 2010-01-22  green         NA      NA        NA
```

Merging Data

Data Merging/Append in Base R

- ▶ Merging - joining data sets together - usually on key variables, usually "id"
- ▶ merge() is the most common way to do this with data sets
- ▶ rbind/cbind - row/column bind, respectively
  - ▶ rbind is the equivalent of "appending" in Stata or "setting" in SAS
  - ▶ cbind allows you to add columns in addition to the previous ways
- ▶ t() is a function that will transpose the data

Merging

```
base <- data.frame(id = 1:10, Age= seq(55,60, length=10))
base[1:2,]
```

Perform Operations By Groups of Variables
Perform Operations By Groups: base R

The `tapply` command will take in a vector (`X`), perform a function
(`FUN`) over an index (`INDEX`):

```
args(tapply)
```

```
function (X, INDEX, FUN = NULL, ..., simplify = TRUE)
NULL
```

Perform Operations By Groups: base R

Let's get the mean Average ridership by line:

```
tapply(wide$Average, wide$line, mean, na.rm = TRUE)
```

```
    banner      green     orange     purple
```