# Merging Multiple Data Files into One Data Frame

*April 24, 2011*

By Tony Cookson



(This article was first published on **Coffee and Econometrics in the Morning**, and kindly contributed to R-bloggers)

We often encounter situations where we have data in multiple files, at different frequencies and on different subsets of observations, but we would like to match them to one another as completely and systematically as possible. In R, the merge() command is a great way to match *two* data frames together.

Just read the two data frames into R

```
mydata1 = read.csv(path1, header=T)
mydata2 = read.csv(path2, header=T)
```

Then, merge

```
myfulldata = merge(mydata1, mydata2)
```

As long as `mydata1` and `mydata2` have at least one common column with an identical name (that allows matching observations in `mydata1` to observations in `mydata2`), this will work like a charm. It also takes three lines.

What if I have 20 files with data that I want to match observation-to-observation? Assuming they all have a common column that allows merging, I would still have to read 20 files in (20 lines of code) and `merge()` works two-by-two... so I could merge the 20 data frames together with 19 merge statements like this:

```
mytempdata = merge(mydata1, mydata2)
mytempdata = merge(mytempdata, mydata3)
.
.
.
mytempdata = merge(mytempdata, mydata20)
```

That's tedious. You may be looking for a simpler way. If you are, I wrote a function to solve your woes called `multmerge()`.* Here's the code to define the function:

```
multmerge = function(mypath){
filenames=list.files(path=mypath, full.names=TRUE)
datalist = lapply(filenames,
function(x){read.csv(file=x,header=T)})
Reduce(function(x,y) {merge(x,y)}, datalist)
```

After running the code to define the function, you are all set to use it. The function takes a path. This path should be the name of a folder that contains **all of the files** you would like to read and merge together **and only those files you would like to merge**. With this in mind, I have two tips:

amateur's dabbling
SYTYCD — where are these terrific dancers come from?
Wrap-up on Blogging with R Markdown and tumblr
igraph and structured text exploration
What kind of profitable financial forecasting models and algorithms do R professional users focus on?
Trying for a baby? Here's how long it might take.
Horizon Plot Already Available
multiplicitiesoffreedom demonstrates Chaos Theory in Excel. If...
Bayesian credible intervals in the mainstream medical literature
Impact of hardware characteristics on detectable fault behavior
Decline Effect in Linguistics?
The inner workings of R objects

OTHER SITES

Forex blogs
SAS blogs

POPULAR SEARCHES

ggplot
RStudio
hadoop
ggplot2
googleVis
heatmap
quantmod
trading
sweave
animation
Web Scraping
PCA
finance
knitr
map
maps
Tutorial
Excel
boxplot
Latex
GIS
Rcmdr
eclipse
twitter
ecdf
SQL
time series
ANOVA

1. Before you use this function, my suggestion is to create a new folder in a short directory (for example, the path for this folder could be "`C://R//mergeme`") and save all of the files you would like to merge in that folder.
2. In addition, make sure that the column that will do the matching is formatted the same way (and has the same name) in each of the files.

Suppose you saved your 20 files into the mergeme folder at "`C://R//mergeme`" and you would like to read and merge them. To use my function, you use the following syntax:

```
mymergeddata = multmerge("C://R//mergeme")
```

After running this command, you have a fully merged data frame with all of your variables matched to each other. Of course, most of the details in matching and merging data come down to making sure that the common column is specified correctly, but given that, this function can save you a lot of typing.

*Maybe a function like this exists out there already, but I think it is entertaining to write helpful functions. I also trust the functions more if I write them myself and test them on my own problems.



**Tags:** Data, useful functions

Comments are closed.

regression
Time%20Series