the Tarzan

[R] + applied economics.

R: apply() + function = no need for loops

In my research, I am constantly running the same computation over every combination of month-day-year-hour in a given sample's time period. Traditionally, this can be done using loops, like so:

R:

```r
k = 2008       # year start
j = 1          # month start
i = 1          # day start
h = 1          # hour start

# start nested loops:
for (k in 2008:2010) {
for (j in 1:12) {
for (i in 1:31) {
for (h in 1:24) {

print(paste('The date is ',paste(j,i,k,sep='/'),' hour ',h,sep=''))

}}}}
```

However, there is a cleaner, more efficient way to go. That is, to write a function that takes the day, month, year, etc. as input parameters, and call it using `apply()`. For a great explanation and introduction to using `apply()`, `sapply()`, `lapply()`, and other derivatives of `apply()`, see this excellent poston Neil Saunders blog: "What You're Doing is Rather Desperate".

To follow our silly example from above, we could create a function that prints the date and hour:

```r
dateprint = function(MM,DD,YR,HR) {
print(paste('The date is ',paste(MM,DD,YR,sep='/'),' hour ',HR,sep=''))
 }
```

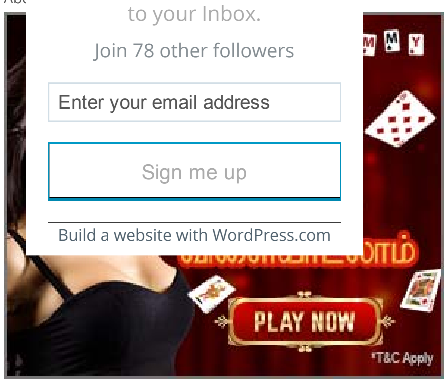
Then we could call the function as follows:

```r
k = c(2008:2010)       # year range
j = c(1:12)            # month range
i = c(1:31)            # day range
h = c(1:24)            # hour range

# Call function using apply() and defined parameters
output = apply(expand.grid(j,i,k,h), 1,
         function(x,y,z,a) dateprint(x[1],x[2],x[3],x[4]))

# Apply stores the output as a list
# I like to convert it to a dataframe for easier viewing and manipulation.
output=data.frame(output)
```

Notice that you are essentially giving `apply()` an "input matrix" created by `expand.grid()`; apply() takes parameters from each row of that "input matrix" and feeds them to our `dateprint()` function. You can tell `apply()` to take parameters from each column by changing the "1" to a "2" within your call of `apply()`.

I am not too close with the back end of R, so I am not certain that [illegible] will increase the computational efficiency of your code. That said, it is another approach [illegible] I use often. Furthermore, it cleans up your code a scintilla.

Clean code = happy code.



Search this blog

Search...

Contributors

Kevin Goulding

Categories

Econometrics
Econometrics with R
Numpy
Python
R tips & tricks
Surviving Graduate Econometrics with R
TikZ for Economists
Visualizing Data with R
White Papers

Twitter feed

RT @gappy3000: This post, apparently about #julialang and #pydata, explains why #rstats has become the standard of data analysis http:// ... 3 years ago

RT @justinwolfers: "if prediction markets are really as valuable as economists think, then..more experimentation could prove worthwhile. ... 3 years ago

RT @vsbuffalo: For me the biggest victory is for statistics and empiricism. Go Nate Silver and @fivethirtyeight for a brilliant forecast ... 3 years ago

Follow @baha_kev

Tag Cloud

cluster-robust Econometrics heteroskedasticity LaTeX Numpy Parallel Computing plots Python R STATA tex TikZ

Related

Parallel computing with package 'snowfall'
In "R tips & tricks"

TikZ diagrams with R: loops with tikzDevice
In "Visualizing Data with R"

Clustered Standard Errors in R
In "Econometrics with R"

Posted on July 13, 2011 at 5:24 pm in R tips & tricks | RSS feed | Reply | Trackback URL

Tags: R

2 Responses to "R: apply() + function = no need for loops"

Cameron
February 24, 2014 at 7:53 pm
Wouldnt using i(1:31) make all days for each month 31 I think seq_along in a for loop works better as it accounts for leap years and days of the month

Reply

Trackbacks

Parallel computing with package 'snowfall' | the Tarzan
February 21, 2012 at 4:21 pm

Leave a Reply

Enter your comment here...

Tags

cluster-robust econometrics heteroskedasticity latex numpy parallel computing plots python r stata tex tikz

Calendar

July 2011

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

« Jun Feb »

Archives

- October 2012
- February 2012
- July 2011
- June 2011
- May 2011

Blogroll

- Documentation
- Plugins
- Suggest Ideas
- Support Forum
- Themes
- WordPress Blog
- WordPress Planet