Using Command-Line Arguments to Control the Operation of the R Program

Junjia Zhu

Feb. 26, 2007

This article is a brief introduction on how to use command-line arguments to control the operation of the R program.

If only one variable needs to be controlled

This is a the print-out of program "`test.R`". The value of variable `n` will be passed from the command line.

```
$ cat test.R
n1 <- commandArgs()
n <- as.numeric(n1[length(n1)])
a<-1:n a
```

To Run the program:

```
$ R --slave --args 5 <test.R
[1] 1 2 3 4 5
```

The option "texttt–slave" tells R only show the outputs; while the option "`--args`" tells R some arguments will come with the command line.

To see all the R code with the outputs, try "`R --no-save --args 5 <test.R`".

If more than one variables need to be controlled

This is a the print-out of program "`test2.R`". The values of variables `a1,a2,a3` will be passed from the command line.

```
$ cat test2.R
n1 <- commandArgs()
n <- n1[length(n1)]
tmp<-strsplit(n,",")
a1 <- as.numeric(tmp[[1]][1]); a1
a2 <- as.numeric(tmp[[1]][2]); a2
a3 <- as.numeric(tmp[[1]][3]); a3
```

To Run the program:

```
$ R --slave --args 24,25,105 <test2.R
[1] 24 [1] 25 [1] 105
```

Please note that the user needs to be consistent in the length of the inputs (in our example, exactly three numbers are inputed), and the separators to be used (in our example, "," is used).

Web reference

1. http://finzi.psych.upenn.edu/R/Rhelp02a/archive/89984.html
2. https://sws.stat.iastate.edu/resources/programmingExamples/R/R-Command-Line.html
3. http://finzi.psych.upenn.edu/R/Rhelp02a/archive/49465.html
   (The method listed in this link is not discussed here: the author suggested use environment variable in UNIX.)