

Using R on beast

The R GUI for Windows is very nice, but often one needs to run a program requiring much more memory and CPU power than a PC can deliver. Linux workstations are a relatively cheap and popular option and we have one, called "beast", physically located in the copy room of the Business and Economics Research Center.

In batch mode you typically will want to run your program, then look at your output, and then go back to your program to edit it before running it again. That sequence of tasks is made easier by creating a batch file in linux. In this course we will use a batch file named "ssr" that contains the following lines:

```
R CMD BATCH $1.R
less $1.Rout
pico $1.R
```

Run the R program *filename.R* by typing `./ssr` followed by the name of your program **without** the extension *R* (i.e., `./ssr filename`). For example, if you have a program called *class2.R*, type the following at your beast prompt:

```
./ssr class2
```

The computer follows these steps:

- Works on your program in R, during which you don't see anything happening on the screen. The output from your program is saved in a file, with the same filename as your program, but with the extension *Rout*. For example, the program *class2.R* generates an output file called *class2.Rout*.
- When R completes its run, your output file (*filename.Rout*) is shown using the utility *less*. The output file shows your program along with information on any problems encountered during execution and whatever printed output requested in your program.
- Type 'q' and you leave *less*, and your input file (*filename.R*) automatically opens in the editor *pico*. After making any changes, type ^x to quit *pico*, making sure to respond affirmatively when asked if you wish to save.

Thus, to use R on beast, you need to know four basic sets of commands:

- 1) *linux* commands. These are used at the beast prompt, and do things like erase files, create directories, copy files, and move you around among directories. A few of these are given below.
- 2) *R* commands. These are written in the body of your *R* program. The online help documents are the best reference. When writing programs, it is easiest to use old programs as a template.
- 3) *less* commands. These are used when viewing your output file. A few commands are given on this sheet; others can be learned by accessing the help screens in *less*.
- 4) *pico* commands. These are used when editing your programs. A few commands are given on this sheet; others can be learned by accessing the help screens in *pico*.

Here are several **less** commands:

- G takes you to the bottom of your file
- g takes you to the top of your file
- -N places line numbers in your file
- /string searches through the file for a specific string (case-sensitive)
- -S chops long lines (rather than wraps)
- space pushing the spacebar scrolls down one page
- b pushing lowercase b scrolls up one page
- h help: gives long list of commands
- q quit
- :n "next"--moves to the next file in the *less* queue
- :p "previous"--moves to the previous file in the *less* queue

Here are some **pico** commands:

- ^k delete current line
- ^u undelete at current cursor point (^k and ^u are used to copy and paste)
- ^v page down

- ^y page up
- ^w ^v end of file
- ^w ^y top of file
- ^w string search for string
- ^a beginning of line
- ^e end of line
- ^d delete current character
- ^x quit
- ^g help

A few **linux** commands:

```

cd               takes you to your own root directory
cd ..            takes you one step toward the root in the directory tree (note the space
                 before the two periods)
ll               (lowercase "ll")lists all of the files and immediate subdirectories in
                 your current directory
pwd              gives the path of your current directory
cd dirname     takes you to the immediate subdirectory named dirname
rm filename     removes the file called filename
mkdir dirname   creates the directory called dirname
rm -r dirname   removes the directory called dirname
cp file1 file2  copies the file called file1 to a new file called file2
mv file1 file2  renames the file called file1 to file2

```

To find out more about any *linux* command, such as *chmod*, simply type the word *man* followed by the command name. For example, for *chmod*:

```
man chmod
```

The resulting output displays the help pages from the *linux* manual for the command. A very useful list of linux commands is given here: <http://www.oreillynet.com/linux/cmd/>