

First R-03

Paul E. Johnson¹ ²

January 18, 2013

Where are you?

- When you start R, and you are at the command prompt, “where are you”?
- This example: I started R in my Linux home directory, /home/pauljohn



```
miterm
pauljohn@p1ap-124:~$ R
R version 2.15.2 (2012-10-26) -- "Trick or Treat"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> getwd()
[1] "/home/pauljohn"
>
```

Working Directory Means...

- Everything R reads (and writes) has to be (or go) “somewhere”
- You CAN specify input and output with full, (long, boring, easy-to-break paths) like “C:/users/your-name-here/Documents/psych790/exercise-1/mydata.dat”
- But it would be much less error prone if you could refer to “mydata.dat” and R would know where to find it.
- So mydata.dat should be in your working directory
- That’s the same place where you keep your R code and where you want to write R output.

Path

- Graphical Computer Interfaces are making it tougher to know “where am I?”.
- You click on a big blue R icon to start R, where does it “put you?”
- If you start R carelessly (click on a menu or icon), it does not know where you are working.
- The `getwd()` command displays the current working directory

```
> getwd()
```

```
[1] "/home/pauljohn/tmp/lyx/tmp/lyx_tmpdir.T14096/lyx_tmpbuf0"
```

- Misty Water Colored Memories of MS Windows
 - Olden days: R would think your working directory was “C:\Program Files\R” or something silly like that.
 - Now, perhaps it guesses “C:\user\your-name-here\Documents” or such.

Be the Boss of R!

- It is OK to ask R where it is working.
- Better than asking, is Telling R where to work. Be assertive!
- For that, you need to develop a little “routine”.
- If you do this, you will never have that horrible problem of not being able to find your R code, your output, or your data. Keep it all together.

Please Be organized

- Be like me. Make a directory structure
 - Macintosh: `/usr/your-name-here/documents/ps706/First-R-02`
 - Windows: `C:\users\your-name-here\Documents\ps706\First-R-02`
- Each separate project is in a separate folder.

Let R know where you want the WD to be (Difficult way)

- This is the hard way. It is also the way that almost all R novices think is most obvious.
- 2 steps
 - Start R carelessly, let it open up anywhere it want, and
 - Use “setwd()” to make R change its working directory
- Use “setwd()”, with a directory name IN QUOTES.
 - Windows:

```
> setwd("c:/users/your-name-here/Documents/ps706/First-R-02")
```
 - YES, I mean forward slashes
- FYI: For other Windows tips, please see the FAQ for Windows that is distributed with R documentation. (don't forget “help.start()”).

Let R know where you want the WD to be (Smarter way)

- Make a directory
- Put an R file in that directory.
- Open that file in an editor (even the R editor window will do, if your OS has one)
- R will figure out where the file was and assume that is the working directory.

Use some editor, like Emacs

- Emacs with ESS (Emacs Speaks Statistics)
- Notepad ++ for Windows with the “NPPPTOR” addon
- When you open an R text file with one of these, and then use that program's R launcher, then R will know where you want to work.

Relative File Locations are OK

- Suppose your working directory is “/home/your-name/ps706/exercise_1”
- You could keep a data file in the directory above that, say “/home/your-name/ps706/mydata.dat”
- If you start R and the working directory is “/home/your-name/ps706/exercise_1”, then when you want to refer to the data file, refer to it as “../mydata.dat”
 - the two periods and a slash mean (on all operating systems) to go “back one level” and get something.
- In R you can also create subfolders and then write output in them.

```
> dir.create(myPlotDir)
```

Just one more thing before finishing

There are many output file formats for graphics, and many ways to create them. Some operating systems have a “pull down” Save As menu to create a PDF file. I don’t do it that way, so try to type this:

```
> x <- rgamma(100, 1.5, 2.2)
> y <- rnorm(100) + 0.2 * x
> pdf("RegPlot1.pdf", height=6, width=6, paper="special")
> plot(y ~ x, main="Here comes a line of best fit!")
> mod1 <- lm(y ~ x)
> abline(mod1)
> dev.off() #turns off pdf output device
```

```
pdf
2
```

```
> x <- rgamma(100, 1.5, 2.2)
> y <- rnorm(100) + 0.2 * x
> pdf("RegPlot1.pdf", height=6, width=6, paper="special")
> plot(y ~ x, main="Here comes a line of best fit!")
> mod1 <- lm(y ~ x)
> abline(mod1)
> dev.off() #turns off pdf output device
```

```
pdf
2
```

Just one more thing before finishing

Here comes a line of best fit!

