

# A Very Quick Intro to R

November 24, 2013

R is a powerful open source environment for statistical computing, very widely used for data analytics. As such, some familiarity with it is likely to be valuable to CSE students. Additionally, the built-in support for probability and statistics, plus simple graphics, make it a good tool to help reinforce concepts in this course, so you will be doing a few short, simple programs in R as part of your homework. This document is meant to outline the basics, probably all you'll need for the homework, although I encourage you to explore more deeply.

**Setup:** The R system lives is here: <http://www.r-project.org>. It is very easy to download and install. If you prefer not to install your own copy, it is preinstalled on the CSE lab machines. You can remotely run it there. If so, X-windows will be convenient for plotting and the help system, but isn't necessary. (X-windows support is built into modern Linux systems and Mac OS X; although on a Mac you may need to download and install it on first use, it's a very simple process. I don't know about recent versions of Windows; if someone can give me a pointer to a web page with simple instructions, I'll add a link here.)

There are many good online R tutorials, and it has good built-in help, but you don't need to learn much to do any of the assignments. R has (surprise!) variables, loops, and print statements. E.g., from the command prompt on your computer:

```
mylaptop% ssh attu                # omit if R is installed on your computer
... or ...
mylaptop% ssh -X attu             # '-X' if your computer supports X-windows
```

Then run R:

```
attu4% R                          # start R
R version 3.0.2 ... blah blah blah  # R wakes up talking...
> 2+2                               # it's a calculator...
[1] 4
> choose(4,2)*factorial(2)         # it knows basic math functions...
[1] 12
```

**Help:**

```
> ?choose                          # get help on builtin-functions,
> ?'*'                              # ... operators,
> ?'for'                            # ... or language keywords.
```

(The examples above should look just like that on your terminal; the ones below are formatted differently just for legibility, using the excellent [knitr](#) package.)

**Variables and Assignment:** Variables don't need declarations needed. “<-” is the assignment operator:

```
myvariable <- 42
```

**Printing:** Three examples:

```
myvariable

## [1] 42

print(myvariable)

## [1] 42

cat("My variable =", myvariable, ".\n")

## My variable = 42 .
```

**Loops:**

```
for (i in 1:3) print(i^2)

## [1] 1
## [1] 4
## [1] 9

while (i < 5) {
  print(i^2 - 1)
  i <- i + 1
}

## [1] 8
## [1] 15
```

**Vectors:** Actually, you will use loops less frequently than in Java, once you figure out vectors:

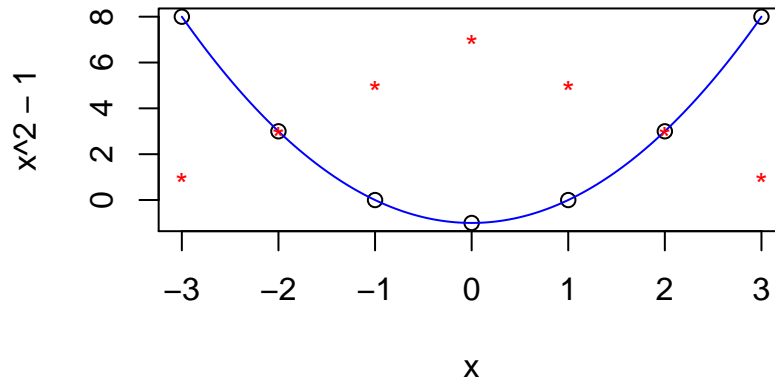
```
x <- -3:3
print(x^2 - 1)

## [1] 8 3 0 -1 0 3 8
```

But for homework (unless otherwise stated), suit yourself.

**Plotting:** Plotting, which may be needed in later assignments, is also easy, including the ability to overlay additional points or lines on a plot:

```
plot(x, x^2 - 1)
xsmooth <- (-300:300)/100
lines(xsmooth, xsmooth^2 - 1, col = "blue")
points(x, 7 - 2 * abs(x), pch = "*", col = "red")
```



Other tricks: `par(mfrow=c(2,2))` before the first plot will put 4 plots (2 rows of 2) on one page. For final output, putting `pdf('path/to/myplot.pdf', onefile=T, width=8.5, height=11)` in front of your first `par()` or `plot()` call and `dev.off()` after the last one will write your plot(s) to a .pdf file. Copy/paste from the graphics window may also work. There are *many* shared optional parameters to the plot routines, most described on the help page for `par()`.

**Functions:** Define and call functions like this:

```
myfact <- function(n) {
  return(prod(1:n))
}
myfact(5)

## [1] 120
```

**Other:** The above examples are probably more than enough “programming” to do most assignments. R has an extensive built in library of functions for math/probability/statistics that will simplify what you need to do in particular cases.

**The “Hashing” Problem in HW#4:** The R help pages for “for”, `choose()`, `factorial()`, `lfactorial()`, `exp()` and `log()` may be useful (and `runif()`, `unique()`, `length()`, and the plot functions above if you want to tackle some extra credit.)

**The “Coin Flipping” Problem in HW#5:** Look at R help for `rbinom()` to sample the binomial distribution, `plot(x, y)` to make a scatter plot from vectors  $x$  and  $y$ , and the plotting examples above.

**“Voters” and “Mint” in HW#6:** Look at `dnorm()/pnorm()` for the PDF/CDF, resp., of the normal distribution, and similarly `dbinom()/pbinom()` for the binomial. For the extra credit parts, also look at `rnorm()/rbinom()` for random sampling from those distributions, as well as, perhaps, `hist()` to plot a summary histogram.