The Fourth R: Using R in STEM Courses

Brian Dennis
Dept Fish and Wildlife
and
Dept Statistical Science
University of Idaho

What is "R"?

R is a free, open source computer package for scientific graphs, calculations, and statistics.

R was written by scientists and statisticians, for scientists to use in their work.

R runs on Windows, Mac, and Linux/Unix operating systems.

R is free. Download from the "R Project" website:
www.r-project.org

R is spreading rapidly in the scientific/technology world

R is taught at many colleges and universities.

Many corporations and government agencies use R.

Many free tutorials posted online.

The library of routines contributed by scientsts for specialty analyses is growing rapidly.

Welcome to TimesPeople
What's this?

TimesPeople Lets You Share and Discover the Best of NYTimes.com 9:49 AM

HOME PAGE | TODAY'S PAPER | VIDEO | MOST POPULAR | TIMES TOPICS

Get Home D

The New York Times

Business Computing

Search

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION | ARTS | STYLE | TRAVEL | JO

AUTOS

Search Technology
Go

Inside Technology
Internet | Start-Ups | Business Computing | Companies

Bits Blog »

Personal Tech »
Cellphones, Camera

Data Analysts Captivated by R's Power

More Articles in Technology »

Stuart Isett for The New York Times

R first appeared in 1996, when the statistics professors Robert Gentleman, left, and Ross Ihaka released the code as a free software package.

By ASHLEE VANCE
Published: January 6, 2009

SIGN IN TO E-MAIL OR SAVE THIS
PRINT
SINGLE PAGE
REPRINTS

To some people R is just the 18th letter of the alphabet. To others, it's the rating on racy movies, a measure of an attic's insulation or what pirates in movies say.

Let's do some R!

(Web-based R at:

www.stats4stem.org/rweb.html

The site works on your smart phone!)

The simplest way to use R is as an ultra-powerful calculator. At the prompt, type: 6+8 and hit the Enter key:

```
> 6+8
[1] 14
```

Row 1 of the answer is 14. Answers to complicated questions often come in many rows (matrix calculations, for instance), and so R prints row numbers along with the answers on the screen.

Try subtraction:

```
> 6-8
[1] -2
```

Negative numbers:

```
> 6+-3
[1] 3
```

Multiplication is an asterisk:

```
> 6*8
[1] 48
```

Division is a forward slash, so 6 divided by 8 is

```
> 6/8
[1] 0.75
```

Raising to a power is the carat symbol "^", so 6^8 is

```
> 6^8
[1] 1679616
```

Many calculations can appear in one statement. The calculations with `^` are done first, followed by `*` and `/` , and then `+` and `-`, and the calculations are done left to right:

```
> 6+8*2-12/4-1
[1] 18

> 2+5*3^2
[1] 47
```

The order of operations can be altered by using parentheses:

```
> (6+8)*2-(12/4-1)
[1] 26
```

Parentheses inside of parentheses are done first. Make sure that every left paren "(" is followed by a right paren ")":

```
> (6+8)*(2-12/(4-1))
[1] -28
```

R will store everything. Just give your calculations names:

```
> deb=6+8
> tom=5-1
> deb-tom
[1] 10
```

R names are sensitive to lower and upper case. In R, the name Deb is different from deb. Are deb and tom still there?

```
> deb
[1] 14
> tom
[1] 4
```

If you reuse a name for something different, R will erase the old value and replace it with the new value:

```
> tom=9
> tom
[1] 9
```

The equals symbol "`=`" in R (and in most computer programming languages) does not mean "equals". Instead, it means:

calculate what is on the right, and store the result using the name on the left.

So, `deb=deb+1` means:

take the old value of deb, add 1, and store the result as the new value of deb.

A statement with an equals sign is called an "assignment statement," assigning the value resulting from the calculation on the right to the storage location on the left. Try it:

```
> deb=deb+1
> deb
[1] 15
```

R has many scientifically cool properties

A. Vectors! (whole lists of numbers):

```
> x=c(4,-6,3,5,2,-1,0)
> x
[1]  4 -6  3  5  2 -1  0
```

Add a single number to every element of x:

```
> y=4
> x+y
[1] 8  -2  7  9  6  3  4
```

Add or multiply the elements of two vectors:

```
> y=c(1,2,3,4,5,6,7)
> v=x+y
> v
[1]  5 -4  6  9  7  5  7
> w=x*y
> w
[1]   4 -12   9  20  10  -6   0
```

Answers with very large or very small numbers are printed by R in computer-scientific notation: `3.303069e+11` means 3.303069×10^{11} (the "e" signifies "exponent").

All of the arithmetic operations, addition, subtraction, multiplication, division, and power, are "vectorized" in R. The order of operations is the same for vector arithmetic, and parentheses may be used in the usual way to indicate which calculations to perform first:

```
> mitt=c(1,2,3)
> barry=c(-1,1,.5)
> 2*(mitt+barry)
[1] 0  6  7
> 2*mitt+barry
[1] 1.0  5.0  6.5
```

Continue incomplete statements on the next line:

```
> mitt=c(-1,1,
+ .5)
> mitt
[1] -1.0  1.0  0.5
```

Special vectors containing sequences of numbers can be built with a colon ":" or with the "seq" function:

```
> j=0:10
> j
[1] 0  1  2  3  4  5  6  7  8  9  10

> k=5:-5
> k
[1] 5  4  3  2  1  0  -1  -2  -3  -4  -5

> seq(10,15,.5)
 [1] 10.0  10.5  11.0  11.5  12.0  12.5  13.0
13.5  14.0  14.5  15.0
```

Want to see the powers of 2 from 0 to 20?

```
> j=0:20
> 2^j
[1] 1  2  4  8  16  32  64  128  256
[10] 512 1024 2048 4096 8192 16384 32768 65536 131072
[19] 262144  524288  1048576
```

B. Functions!

```
> x=0:10
> sum(x)
[1] 55

> mean(x)
[1] 5

> sqrt(x)
[1] 0.000000  1.000000  1.414214  1.732051
2.000000  2.236068  2.449490  2.645751
[9] 2.828427  3.000000  3.162278

> x^(1/2)
```

```
[1] 0.000000  1.000000  1.414214  1.732051
2.000000  2.236068  2.449490  2.645751
[9] 2.828427  3.000000  3.162278

> sin(x)
[1] 0.0000000  0.8414710  0.9092974
0.1411200  -0.7568025  -0.9589243
[7] -0.2794155  0.6569866  0.9893582
0.4121185  -0.5440211

> exp(x)
[1] 1.000000  2.718282  7.389056  20.085537
54.598150
[6] 148.413159  403.428793  1096.633158
2980.957987  8103.083928
```

```
[11] 22026.465795

> log(x)
[1] -Inf  0.0000000  0.6931472  1.0986123
1.3862944  1.6094379  1.7917595
[8] 1.9459101  2.0794415  2.1972246
2.3025851

> log10(x)
[1] -Inf  0.0000000  0.3010300  0.4771213
0.6020600  0.6989700  0.7781513
[8] 0.8450980  0.9030900  0.9542425
1.0000000
```

In the above, `sin(x)` is the trigonometric sine function, with the angle measure `x` in radians, where 1 radian is around 57 degrees (360 degrees is 2π radians), and `exp(x)` is the exponential function e^x, where e is the special irrational number $e =$ 2.71828... from calculus. Also, `log(x)` and `log10(x)` are, respectively, natural (base e) logarithm and base 10 logarithm of `x`. The logarithm of zero is undefined at minus infinity (printed as the code `-Inf` in R).

The arguments of functions can themselves be R calculations:

```
> y=.2*exp(-.2*x)
> y
[1] 0.20000000  0.16374615  0.13406401
0.10976233  0.08986579  0.07357589
[7] 0.06023884  0.04931939  0.04037930
0.03305978  0.02706706
```

C. Graphs!

```
> plot(x,y,type="l")
```

Exponential decay. The type="l" option in the plot() statement specifies a line plot, with lines drawn connecting points but points not drawn.

```
> z=rnorm(10000,500,100)
> hist(z)
```

Histogram of 10,000 random numbers from a normal distribution with mean of 500 and a standard deviation of 100.

D. Other cool properties

Define your own functions.

Bring files of data in for analysis (columns become vectors); write files of data as output.

Loops to repeat blocks of R statements many times.

Random numbers & simulation.

Matrix calculations.

Every statistical & graphical analysis method.

Edit, save, & run list of R statements (script).

Why use R in STEM classes?

(1) With R, scientific calculations and graphs are fun and easy to produce.

(2) R could be used across a wide variety of STEM courses, promoting the integration of STEM subjects. R skills would follow a student from course to course.

(3) For students, R is probably the most universally available computational tool (aside from counting on fingers). R runs on most platforms, so R can be made instantly available to every student in every course. No additional proprietary software or calculators have to be purchased. Once R is installed on a machine, the internet is not needed to use R.

(4) R invites collaboration among students.

Graphical displays are big, and easily saved as files.

R expressions are easily shared and texted.

R scripts (lists of expressions) are easily saved, modified, shared, emailed. *R builds on itself*.

(5) R skills follow a student to college and professional life.

(6) There is excellent, free documentation available online (including videos).

(7) Inexpensive tutorial and reference books are available.

Calculators

Graphing calculators are hard to learn: a student rarely actually masters the daunting sequence of keystrokes needed to produce even the simplest graph.

The graphs themselves are low-resolution toys that are near-useless.

Graphing calculators are expensive.

Spreadsheets

The most widely used spreadsheet is proprietary.

The quality of the graphical displays produced by the proprietary spreadsheet is reviled by many working scientists, because the common default graphing options are unacceptable in scientific journals (axis label sizes and fonts, line thicknesses, tic marks, plotting symbols, and so on).

For many years, the proprietary spreadsheet has had statistical routines known to be poorly programmed (see http://practicalstats.com/xlsstats/excelstats.html).

One can use an open source spreadsheet, but then one must spend time dealing with installation, compatibility issues and uneven documentation.

On the web

A self-learn tutorial at the National Center for Ecological Analysis and Synthesis:
http://www.nceas.ucsb.edu/files/scicomp/Dloads/RProgramming/BestFirstRTutorial.pdf

R for biologists at U Tennessee:
http://www.nimbios.org/products/RforBiologistv1.1.pdf

R tutorial at Illinois State U:
http://math.illinoisstate.edu/dhkim/rstuff/rtutor.html

R resource website at UCLA:
http://www.ats.ucla.edu/stat/r/

R tutorial (with videos) at U Colorado/Denver:
http://math.ucdenver.edu/RTutorial/

Intro to R at the R Project website:
http://cran.r-project.org/doc/manuals/R-intro.html

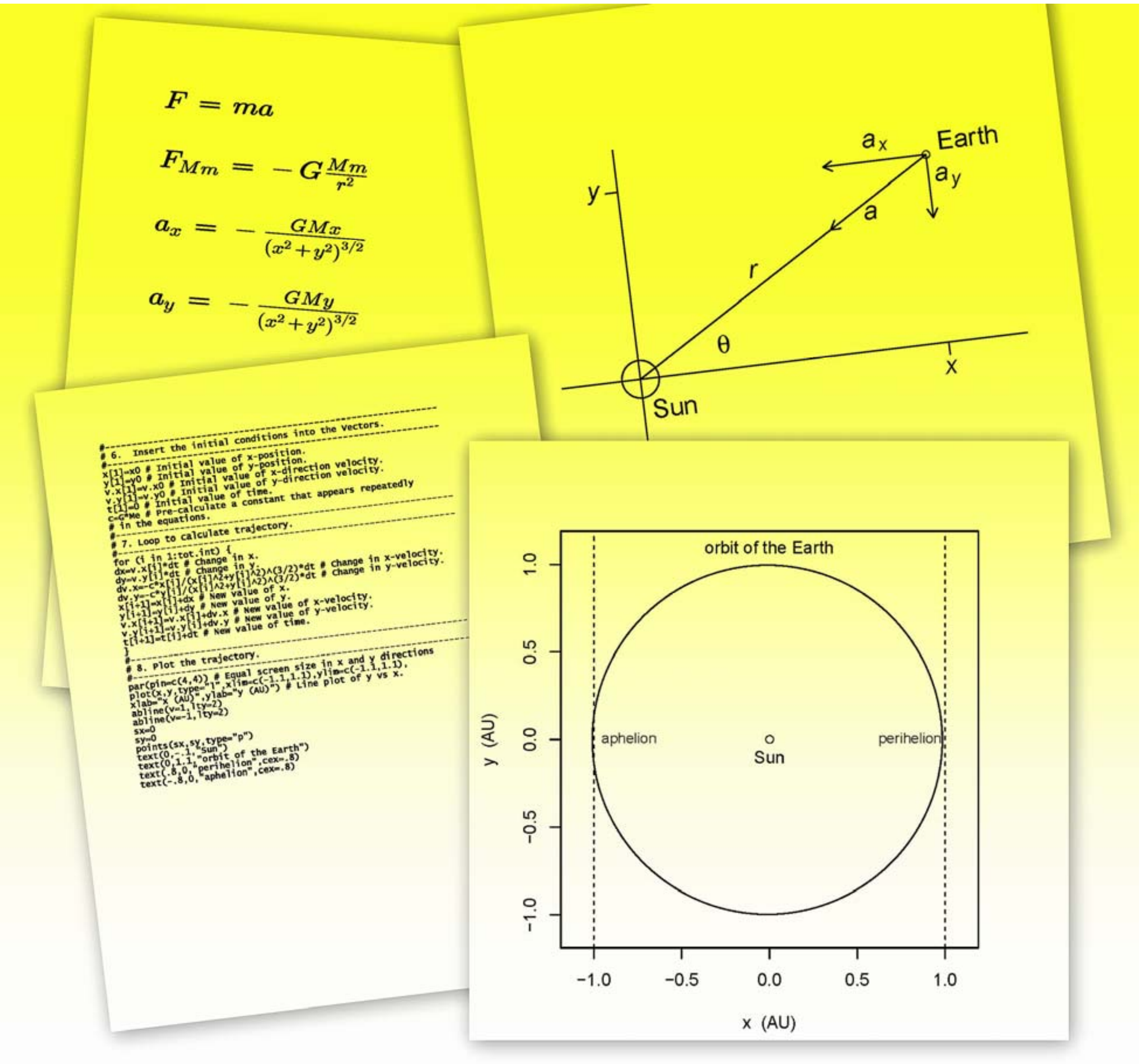
Introductory-level instructional books about R

R in Action
http://www.manning.com/kabacoff/

R in a Nutshell
http://shop.oreilly.com/product/0636920022008.do

R for Dummies
http://www.dummies.com/store/product/R-For-Dummies.productCd-1119962846.html

The R Student Companion (for high school!)
http://www.crcpress.com/product/isbn/9781439875407



$$F = ma$$

$$F_{Mm} = -G\frac{Mm}{r^2}$$

$$a_x = -\frac{GMx}{(x^2+y^2)^{3/2}}$$

$$a_y = -\frac{GMy}{(x^2+y^2)^{3/2}}$$

```
#-----------------------------------------------------------
# 6.  Insert the initial conditions into the vectors.
#-----------------------------------------------------------
x[1]=x0 # Initial value of x-position.
y[1]=y0 # Initial value of y-position.
v.x[1]=v.x0 # Initial value of x-direction velocity.
v.y[1]=v.y0 # Initial value of y-direction velocity.
t[1]=0 # Initial value of time.
c=G*Me # Pre-calculate a constant that appears repeatedly
# in the equations.
#-----------------------------------------------------------
# 7. Loop to calculate trajectory.
#-----------------------------------------------------------
for (i in 1:tot.int) {
dx=v.x[i]*dt # Change in x.
dy=v.y[i]*dt # Change in y.
dv.x=-c*x[i]/(x[i]^2+y[i]^2)^(3/2)*dt # Change in x-velocity.
dv.y=-c*y[i]/(x[i]^2+y[i]^2)^(3/2)*dt # Change in y-velocity.
x[i+1]=x[i]+dx # New value of x.
y[i+1]=y[i]+dy # New value of y.
v.x[i+1]=v.x[i]+dv.x # New value of x-velocity.
v.y[i+1]=v.y[i]+dv.y # New value of y-velocity.
t[i+1]=t[i]+dt # New value of time.
}
#-----------------------------------------------------------
# 8. Plot the trajectory.
#-----------------------------------------------------------
par(pin=c(4,4)) # Equal screen size in x and y directions
plot(x,y,type="l",xlim=c(-1.1,1.1),ylim=c(-1.1,1.1),
xlab="x (AU)",ylab="y (AU)") # Line plot of y vs x.
abline(v=1,lty=2)
abline(v=-1,lty=2)
sx=0
sy=0
points(sx,sy,type="p")
text(0,-.1,"Sun")
text(0,1.1,"orbit of the Earth")
text(.8,0,"perihelion",cex=.8)
text(-.8,0,"aphelion",cex=.8)
```

Happy R-ing!
These slides can be downloaded at:

webpages.uidaho.edu/~brian/reprints/BDennis_reprint_list.htm

Brian Dennis
brian@uidaho.edu