Introductory statistical computing with R

Thomas Lumley

Biostatistics

2005-9-29

Objectives

- Basic computations in R that will be needed in other courses
- Statistical methods that require some programming and so are not covered thoroughly in 517/518
- Programming techniques and concepts useful in statistical computing

Languages

There are three reasons to learn a programming language

- It is a good tool for problems you will have to solve
- You are going to be forced to use it
- It provides a way to learn new programming abstractions

R qualifies on all of these.

You should also find time to learn C while you are a student and consider learning Ruby or Java (or some other sufficiently different language).

R and S

S is a statistical and graphical programming language developed at Bell Labs over the past 30 years.

S-PLUS is a commercial version of S with many additional features. It is sold by Insightful Co., a Seattle company. The University has a site license, and you can buy copies from the University bookstore.

R is a free implementation of a language very similar to S. You can download it from http://www.r-project.org. I am one of the authors.

R and S

S-PLUS, and now R, are the favored systems for statistical research and teaching, because of

- easy programmability
- interfaces to other languages
- presentation-quality graphics
- interactive data analysis
- R package system for distribution and quality control

GUI for R

R does not have any built-in GUI. It can be run under Emacs, for which you can see the course notes from this summer at https://portal.biostat.washington.edu/computing/scs/

There is an add-on Java GUI available from http://www.rosuda.org/JGR/down.shtml

Other statistical software

The first-year courses and the second-year MS courses use Stata instead of S, and some more advanced courses use Stata as well as S. Health Sciences has a discount plan. You can buy online at `http://www.stata.com` or by phone, and then collect the software from the library.

SAS makes a brief appearance in BIOST 571. Few or none of the teaching faculty use it regularly. MS students interested in working in industry might want to teach themselves SAS. You can get SAS from the bookstore for $100/year. UW Computing and Communications offers short courses in SAS.

Reading data

- Text files
- Stata datasets
- Web pages
- (Databases)

Much more information is in the Data Import/Export manual.

Reading text data

The easiest format has variable names in the first row

case	id	gender	deg	yrdeg	field	startyr	year	rank	admin
1	1	F	Other	92	Other	95	95	Assist	0
2	2	M	Other	91	Other	94	94	Assist	0
3	2	M	Other	91	Other	94	95	Assist	0
4	4	M	PhD	96	Other	95	95	Assist	0

and fields separated by spaces. In R, use

```
salary <- read.table("salary.txt", header=TRUE)
```

to read the data from the file salary.txt into the data frame salary.

http://courses.washington.edu/b570/datasets/salary.dat

Syntax notes

- Spaces in commands don't matter (except for readability), but Capitalisation Does Matter.

- TRUE (and FALSE) are logical constants

- Unlike many systems, R does not distinguish between commands that do something and commands that compute a value. Everything is a function: ie returns a value.

- Arguments to functions can be named (header=TRUE) or unnamed ("salary.txt")

- A whole data set (called a data frame is stored in a variable (salary), so more than one dataset can be available at the same time.

Reading text data

Sometimes columns are separated by commas (or tabs)

```
Ozone,Solar.R,Wind,Temp,Month,Day
41,190,7.4,67,5,1
36,118,8,72,5,2
12,149,12.6,74,5,3
18,313,11.5,62,5,4
NA,NA,14.3,56,5,5
```

Use

```
ozone <- read.table("ozone.csv", header=TRUE, sep=",")
```

or

```
ozone <- read.csv("ozone.csv")
```

Syntax notes

- Functions can have optional arguments (sep wasn't used the first time). Use help(read.table) for a complete description of the function and all the arguments.

- There's more than one way to do it.

- NA is the code for missing data. Think of it as "Don't Know". R handles it sensibly in computations: eg 1+NA, NA & FALSE, NA & TRUE.

Reading text data

Sometime the variable names aren't included

```
 1      0.2      115       90      1         3        68     42 yes
 2      0.7      193       90      3         1        61     48 yes
 3      0.2       58       90      1         3        63     40 yes
 4      0.2        5       80      2         3        65     75 yes
```

and you have to supply them

```
psa <- read.table("psa.txt", col.names=c("ptid","nadirpsa",
       "pretxpsa", "ps","bss","grade","age",
       "obstime","inrem"))
```

or

```
psa <- read.table("psa.txt")
names(psa) <- c("ptid","nadirpsa","pretxpsa", "ps",
           "bss","grade","age","obstime","inrem"))
```

https://courses.washington.edu/b518/datasets/psa.txt

Syntax notes

- c() is a function that makes a single vector from its arguments.

- names is a function that accesses the variable names of a data frame

- Some functions (such as names) can be used on the LHS of an assignment.

Reading Stata data

Stata saves data in files with a `.dta` extension.

```
library(foreign)
salary <- read.dta("salary.dta")
```

Notes:

- Many functions in R live in optional packages. The library() function lists packages, shows help, or loads packages from the package library.

- The foreign package is in the standard distribution. It handles import and export of data. Hundreds of extra packages are available at http://cran.us.r-project.org.

The web

Files for read.table can live on the web

```
antibiotics <- read.csv("http://courses.washington.edu/b518/
datasets/hospital.csv")
```

It's also possible to read from more complex web databases (such as the genome databases)

Operating on data

As R can have more than one data frame available you need to specify where to find a variable. The syntax antibiotics$duration means the variable duration in the data frame antibiotics.

```
## This is a comment
## Convert temperature to real degrees
antibiotics$tempC <- (antibiotics$temp-32)*5/9
## display mean, quartiles of all variables
summary(antibiotics)
```

Subsets

Everything in R is a vector (but some have only one element).
Use [] to extract subsets

```
## First element
antibiotics$temp[1]
## All but first element
antibiotics$temp[-1]
## Elements 5 through 10
antibiotics$temp[5:10]
## Elements 5 and 7
antibiotics$temp[c(5,7)]
## People who received antibiotics (note ==)
antibiotics$temp[ antibiotics$antib==1 ]
## or
with(antibiotics, temp[antib==1])
```

Subsets

For data frames you need two indices

```
## First row
antibiotics[1,]
## Second column
antibiotics[,2]
## Some rows and columns
antibiotics[3:7, 2:4]
## Columns by name
antibiotics[, c("id","temp","wbc")]
## People who received antibiotics
antibiotics[antibiotics$antib==1, ]
## Put this subset into a new data frame
yes <- antibiotics[antibiotics$antib==1,]
```

Computations

```
mean(antibiotics$temp)
median(antibiotics$temp)
var(antibiotics$temp)
sd(antibiotics$temp)
mean(yes$temp)
mean(antibiotics$temp[antibiotics$antib==1]
with(antibiotics, mean(temp[sex==2]))
toohot <- with(antibiotics, temp>99)
mean(toohot)
```

Factors

Factors represent categorical variables. You can't do mathematical operations on them (except for ==)

```
> table(salary$rank,salary$field)

         Arts Other Prof
  Assist  668 2626   754
  Assoc  1229 4229  1071
  Full    942 6285  1984
> antibiotics$antib<-factor(antibiotics$antib, levels=c(1,2),
                            labels=c("Yes","No"))
> antibiotics$agegp<-cut(antibiotics$age, c(0,18,65,100))
> table(antibiotics$agegp)
  (0,18]  (18,65] (65,100]
       2       19        4
```

More tables

```
> with(salary, addmargins(table(rank, field)))
        field
rank     Arts Prof Other   Sum
  Assist  668  754  2626  4048
  Assoc  1229 1071  4229  6529
  Full    942 1984  6285  9211
  Sum    2839 3809 13140 19788
> with(salary, table(rank,field, deg))
, , deg = PhD

        field
rank     Arts Prof Other
  Assist  534  692  2236
  Assoc   989 1002  3418
  Full    623 1608  5701
```

More tables

```
, , deg = Prof

        field
rank     Arts Prof Other
  Assist    0   50   183
  Assoc     0   30   344
  Full      0  345   394

, , deg = Other

        field
rank     Arts Prof Other
  Assist  134   12   207
  Assoc   240   39   467
  Full    319   31   190
```

More tables

```
> with(salary, ftable(table(rank, field,deg)))
                deg  PhD Prof Other
rank   field
Assist Arts          534    0   134
       Prof          692   50    12
       Other        2236  183   207
Assoc  Arts          989    0   240
       Prof         1002   30    39
       Other        3418  344   467
Full   Arts          623    0   319
       Prof         1608  345    31
       Other        5701  394   190
```

Help

- help(fn) for help on fn
- help.search("topic") for help pages related to "topic
- apropos("tab") for functions whose names contain "tab"
- Search function on the http://www.r-project.org web site.

Asking for help

When something doesn't work the best way to ask for help looks like

> I did X. Y happened. I thought Z would happen.

with as much detail as possible about X. Questions like this are easy to answer, so they get answered faster.