Computers and Computer Languages

Frank E Harrell Jr
Department of Biostatistics
Vanderbilt University School of Medicine
f.harrell@vanderbilt.edu
biostat.mc.vanderbilt.edu
Updated February 1, 2004

Outline

1. File Systems
2. Operating Systems
3. Applications
4. Data types and variables
5. Languages
6. Functions
7. Auditing and Reproducible Analysis
8. Utilities, Import/Export/Conversions

File Systems

- Managing files & directories is important function of operating system
- Directories, sub-directories, . . .
- Types of files areas
  - System files (executables, tables)
  - Applications (executables, templates, etc.)
  - User files
- Types of user files
  - Text — documents, simple data
  - Binary — documents, datasets
  - Graphics files
- Organize your work — project directories; not by file type

- User-interaction: mouse & menus, commands, mixture
- McIntosh: ultimate in non-programming
  learning curve +; flexibility -
- Windows 95/98/ME: mainly mouse/menus; 95/98 had DOS for commands; many language tools for extending system
  learning curve moderate; flexibility moderate
- Windows NT/2000: mixture, better security
- All Microsoft OS: closed system, monopoly and collusion, inefficient use of RAM, unreliable, slow, major security problems.
  Cannot remotely use RAM on another machine.

- UNIX/Linux: command based, graphical user interface added
  Most flavors are open; fast, efficient, reliable, better security, more logical window layout with quicker user interaction; many system tools and utilities; fewer applications overall than Windows (so far)
  learning curve -; flexibility +; hassle in installing new devices
  can use remote CPUs with large RAM (compute servers)

Example Commands to OS

Synchronizing home and office computers. Marker file `IMPORTED` on Iomega Zip drive `Z:` is for bookkeeping. The file date for this file will always be set to the last date that the data were imported.

1. Get date of `IMPORTED`
2. If `IMPORTED` does not exist use today minus 3 days
3. Put in a compressed zip archive on `Z:` all project and doc files that have changed since this date
4. On other computer, import zip archive on `Z:`
5. 'touch' `IMPORTED` to set its file date to date of this import
6. Next export on current computer will include files created/modified since date of `IMPORTED`

Applications

- WYSIWYG word processing learning curve +; speed of use -; distortion of graphics files -; assembling large documents whose components are updated separately -; appearance not journal/book quality; spend time worrying how document appears on screen, which takes away from substantive aspects of composition

- Markup language
  learning curve -; speed of use +; other +; minimal use of GUI/menus

Example: LaTeX— Typesetting Language

```
\documentclass{report}
\def\long\1   % 1=true here
\begin{document}
Here is a sentence.  This word is
\textbf{boldface}.
$\sqrt{X^{2}} = X$. $A = \pi \times r^2$.
\begin{itemize}
\item Bullet item one
\item Bullet item two
\end{itemize}
\ifthenelse{\long = 1}{\input{table1}}{}
% optionally insert another file or nothing
\end{document}
```

More Applications: Analytical

- GUI/Menu based: learning curve +; flexibility -; extendability -
- Example: click `Statistics` menu then click names of analysis (`age`) and grouping variable (`sex`), click on which statistics to print
- Command-based: learning curve -; flexibility +; extendability +
- Example (S): `tapply(age, sex, mean)`
- Analytical packages based on GUI, functions, procedures

```
mean(age)                  % S
PROC MEANS; VAR age;       % SAS
```

Basic Scalar (Single Valued) Data Types

- integer
- floating point (scientific notation, e.g. 1.2e5 = 1.2×10^5)
  - single precision (7 significant digits)
  - double precision (15 digits)
- character string, e.g. ‘Jim’
- categorical (“choice”), e.g., 1=good 2=better 3=best
- logical: `TRUE, FALSE, T, F, 1, 0`
- missing values: blank, `?, NA, .`

Complex Data Types

- vector
- matrix: $r \times c$
- multi-dimensional array: $r \times c \times p, p =$ pages
- irregular structure (“list” or “tree”) e.g. states having variable number of counties having varying number of cities, data = population of city in 2000, population in 1990.

Variables

- Name of variable can be more than one letter; rules for names depend on language being used, e.g. `age.yrs, X2, cholesterol, Age`; may be case-sensitive
- Depending on language, variable name may stand for only one value of the variable at a time or it may stand for complex objects such as vectors, matrices, lists
- Variables vs. literals: ‘Jim’ is a particular value. `Jim` might be a variable containing a series of values.
- Examples

```
sex <- ’female’
x   <- 7+2
age.yrs <- age.days / 365.25
```

- Mode of inputting commands in the language
  - Read from previously created file
  - User input one line at a time
- How commands are processed
  - Compiled: read file as a whole, convert into machine language
    Fast, harder to debug, inflexible
    C, Fortran, Basic, Tcl/Tk
  - Interpretive: every command or compound command is executed as it is inputted
    Slower, easier to debug, and change course based on previous results
    Java, Perl, Python, S, Basic, Tcl/Tk

- Arithmetic: `+ - * / ^ ( )`
  $3 \times \frac{(y-5)^2}{z}$ = `3*(y-5)^2/z`
- Precedence of operators: `^`, unary `-`,
  `* / + - < > <= >= == != ! & |`
  Left to right; parentheses rule
- Logical expressions: intersection `a & b`; union `a | b`
- Negation: `!a`; equality, inequality, directional:
  `x==y, x!=y, x>y, x>=y, x<y, x<=y`
  `age > 50 | sex=='male'`
- Are not logical assertions; are tests of conditions
- Assignment: `x <- y,  x = y`

`if then` command to a computer is a conditional command and not a logical assertion.

```
if(sex=='male' & pregnant) error <- T else
   error <- F
if(analysis.type=='concise') {
  print(mean(age))
  print(median(age))
} else {
  print(mean(age)); print(median(age))
  histogram(age); cdf(age)
}
```

Using Logical Values in Arithmetic

- Most languages treat T/F as 1/0 in arithmetic calculations

```
male <- sex=='male'       T/F values
male <- 1*(sex=='male')   1/0 values
k <- 2.3*(sex=='female') + 2.1*(sex=='male')
k is 2.3 for females, 2.1 for males

x <- age*(1 + .2*(sex=='female'))
age for males, 1.2*age for females
```

Operations on Vectors

```
age <- [10, 11, 20, 30]
min(age)       10
max(age)       30
sex <- [’male’, ’female’, ’male’, ’female’]
sex==’female’ F T F T
age >= 20      F F T T
age >= 20 & sex==’female’  F F F T
```

```
NA + 3          NA
3*NA            NA
x <- [3,4,NA]
min(x)          NA
mean(x)         NA
x < 4           T F NA
T | NA          T
F | NA          NA
F & NA          F
T & NA          NA
```

Functions

- Functions for computing and returning something, e.g. `sqrt(4)`
- Functions for doing something, e.g. `print( ), plot( )`
- Function of scalars returning vectors:

```
runif(5)   % compute 5 random 0-1 numbers
0.213  0.876  0.401  0.772  0.101
```

- Function of vector returning scalar:

```
min([11,12,13])          11
```

- Function of vector returning vector:

```
sqrt([1,0,0] + [3,9,25])  2  3  5
```

Common Functions

Algebraic Form	Computer Language
$\lvert x \rvert$	`abs(x)`
$\ln\ x$	`log(x)`
e^x	`exp(x)`
$\sqrt{x}$	`sqrt(x)`
$\min(x_1, x_2, x_3)$	`min(x1,x2,x3)` or
	`min(x),` x a vector

Arguments to Functions

- Symbols for passing values to be operated on
- Syntax very language-dependent
- All allow passing arguments by position
- Some interpretive languages allow default values and passing arguments by name
- Examples

```
function z(x, power) x^power
z(2)          ERROR (power omitted)
function z(x, power=2) x^power
z(2)             4  (using default for power)
z(2,3)           8
z(power=3,x=2) 8  (by name)
```

Auditing and Reproducible Analysis

- Longer analyses requiring many mouse clicks are tedious to re-do when data are updated
- Menu clicks in some systems do not generate an audit trail
- Those that do: re-issuing the analytical steps may not work; audit trail often hard to read or change
- Commands easy to re-issue, correct
- Programs are developed by trial and error; discard errors, save what works (in the right order) in script file
- If use batch-oriented document processing system such as LaTeX, can regenerate entire report when any component changes
- Can use separate graphics file for each plot (e.g., postscript or pdf file)

Utilities, Import/Export/Conversions

- File conversions
  - Data, e.g. DBMS/COPY to convert SAS to S
  - Graphics, e.g. postscript to pdf, jpeg, gif
  - Reports, e.g. Word or LaTeX to html
- File compression and archiving: zip, WinZip for compression and putting many files in one
- Example: zip `my.s, my.txt, my.dat` into `my.zip`
  60% shorter, one file, split back into 3 when unzip

Exercises

Except for Problem 11, write generic “open code”, i.e., code that is not a function.

1. Write two commands that will compute $\frac{1}{a^n}$.
2. Program $w^{2k}y^{-3u}$.
3. Write a command that will result in e, the base of the natural log.
4. Program $\frac{1}{1+e^{-x}}$.
5. Program $7x + 2x^2y^3 + 4x^2y^4$ efficiently.
6. Program $(1 + e^x)(1 + e^y)$ concisely and efficiently.
7. Program $\frac{e^x}{e^x+e^y}$ efficiently.
8. In the quadratic equation $ax^2 + bx + c$ one of the roots is $\frac{-b+\sqrt{b^2-4ac}}{2a}$. Write a command to compute this root.

9. What is the value of `b` at the end of the following program?

```
a <- 3
b <- -1
b <- a + a/3 + b
b <- b/2
```

10. Using a generic computer language, write the negation of `age > 50 & sex=='male'` two ways.

11. Make a truth table for the logical expression `a | !b` that allows for `NA` for `a, b`. Do this by writing a function and invoking it 9 times or 3 times by making one of the arguments a vector.

12. Without using an `if` statement, given a vector of heights called `height` and a parallel vector of subject sexes called `sex` having values of `'female'` and `'male'`, create a new vector whose

values are $1.5 + (0.9 \times$ height) for males and height for females. Assume that the language treats logical true values as one and false values as zero in arithmetic.

13. A parking garage charges $2.75 for parking under 3 hours and $0.75 for additional hours. Write two different commands for computing the total charge given the duration of parking in hours (a floating point number).

14. Salespersons whose weekly sales total more than $5000 receive a bonus. If their travel expense is below $600 the bonus is $100, otherwise it is $50. Compute the bonus two ways.

The last two problems are from Roy Ageloff and Richard Mojena, *Applied Structured BASIC*, Belmont: Wadsworth; 1985, p. 143, 181.