22S:166
Computing in Statistics

Introduction to R

Lecture 5
September 4, 2009

Kate Cowles
374 SH, 335-0727
kcowles@stat.uiowa.edu

What R is

- "an integrated suite of software facilities for data manipulation, calculation, and graphics display" (*An Introduction to R*, Venables, Ripley, and the R Core team)
  - data handling and storage capabilities
  - operators for calculations on arrays and matrices
  - data analysis tools
  - graphical capabilities
  - programming language
  - planned and coherent system

- an implementation of S language
  - S language was developed at AT&T-Bell Labs
    - first version 1976
  - S-Plus is a commercial version of S (begin in 1987)
    - sold and supported by Insightful Corp.
    - GUI
    - many formats supported for graphics export and data input/output
    - runs on Windows, UNIX, Linux (not Macintosh)

- advantages of S
  - extendible
    - users write new functions in S language — just as developers do
    - excellent documentation for adding functions to system
    - users can create their own data types
    - huge international community of users constantly contribute new capabilities
    - contrast with SAS
      - very hard to write new SAS procedures
      - users write in different language (SAS macro or IML) than developers
  - high-level language
    - only a few commands required to do complex things

– language is connected to data while executing
example (from *Statistical Computing and Graphics* course notes by Frank Harrell)

```
if(is.factor(x) | is.character(x) |
  (is.numeric(x) & length(unique(x)) < 20))
  table (x) else quantile(x)
```

computes quantiles of `x` if `x` is numeric and has at least 20 distinct values, requency table otherwise

– object-oriented
  * fewer commands to learn because the same command can be applied to different types of objects
  – Harrell: "best scientific graphics available"
    * Harrell: "SAS graphics are ugly, inflexible, have poor defaults, difficult to program"

R

- international team of statisticians started developing R in early 1990's
  – to provide open source alternative to S-Plus
  – to provide S implementation on Linux (not supported by S-Plus then)
- easy to download and install from web sites
- excellent documentation
- user-contributed libraries called packages expand capabilities
- runs on Windows, UNIX, Linux, Macintosh
- no GUI on most platforms
- fewer data import/export capabilities than S-Plus
  – although add-on packages provide more
  – no export specifically to Powerpoint

Starting and running R interactively on Linux

- recommendation: use a separate subdirectory for each major project you do with R
- in a terminal window, get into the desired subdirectory and start R by entering

```
R
```

- R commands may be issued interactively
- to quit

```
q()
```

  – follow prompts as to whether you want to save *workspace*
  – if you don't save it, any new objects (data, functions, results) created during the current R session will be lost

Starting and running R interactively on Linux

- strongly recommended to use a separate subdirectory for each major R project. You might want one subdirectory for your homework assignments, and another for your group project.
- begin by creating the subdirectory
- copy in or download any needed data files
- then invoke R in that subdirectory

```
[kcowles@p-lnx402 ~]$ mkdir examples166
[kcowles@p-lnx402 ~]$ cd examples166
[kcowles@p-lnx402 ~/examples166]$ ls -a
.  ..
[kcowles@p-lnx402 ~/examples166]$
```

Reading in data from external files

- Use Firefox to download `Cars.dat` from "Datasets" section of course web page into this directory.

```
[kcowles@p-lnx402 ~/examples166]$ ls
Cars.dat
```

- Use a text editor to look at this file. Note that the separators between columns are tabs (You can tell because the cursor jumps) and that the decimal point in numbers is indicated by periods.
- We need to read this file into an `object` in R to analyze it. R has several functions that read in data files in different formats.
- We will use R's built-in help facility to figure out which one to use.


```
[kcowles@p-lnx402 ~/examples166]$ R

R version 2.7.1 (2008-06-23)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

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.
```



```
> help(read.delim)

read.table                package:utils                R Documentation

Data Input

Description:

     Reads a file in table format and creates a data frame from it,
     with cases corresponding to lines and variables to fields in the
     file.

Usage:

     read.table(file, header = FALSE, sep = "", quote = "\"'",
                dec = ".", row.names, col.names,
                as.is = !stringsAsFactors,
                na.strings = "NA", colClasses = NA, nrows = -1,
                skip = 0, check.names = TRUE, fill = !blank.lines.skip,
                strip.white = FALSE, blank.lines.skip = TRUE,
                comment.char = "#",
                allowEscapes = FALSE, flush = FALSE,
                stringsAsFactors = default.stringsAsFactors(),
                encoding = "unknown")
     read.csv(file, header = TRUE, sep = ",", quote="\"", dec=".",
              fill = TRUE, comment.char="", ...)

     read.csv2(file, header = TRUE, sep = ";", quote="\"", dec=",",
               fill = TRUE, comment.char="", ...)

     read.delim(file, header = TRUE, sep = "\t", quote="\"", dec=".",
                fill = TRUE, comment.char="", ...)

     read.delim2(file, header = TRUE, sep = "\t", quote="\"", dec=",",
                 fill = TRUE, comment.char="", ...)


      .......   lots of additional detail ........
```



```
> Cars <- read.delim("Cars.dat")      #  <- is assignment operator

> str(Cars)               # find out structure of the object
'data.frame':   38 obs. of  8 variables:
 $ Country     : Factor w/ 6 levels "France","Germany",..: 6 6 6 6 6 4 4 6 2 5 ...
 $ Car         : Factor w/ 38 levels "AMC Concord D/L",..: 6 21 11 12 8 34 14 18 3 35 ...
 $ MPG         : num  16.9 15.5 19.2 18.5 30 27.5 27.2 30.9 20.3 17 ...
 $ Weight      : num  4.36 4.05 3.60 3.94 2.15 ...
 $ Drive_Ratio : num  2.73 2.26 2.56 2.45 3.7 3.05 3.54 3.37 3.9 3.5 ...
 $ Horsepower  : int  155 142 125 150 68 95 97 75 103 125 ...
 $ Displacement: int  350 351 267 360 98 134 119 105 131 163 ...
 $ Cylinders   : int  8 8 8 8 4 4 4 4 5 6 ...
```