Introduction to R package

Thanks to Professor Yan Yu

Why R?

- free !!!
- available for Unix, Linux and Windows
- Nice plots

- Other Statistical Softwares:

SAS (expensive)

S-plus (close relative of R)

Minitab

SPSS

What is R?

- A programming language for statistical data analysis called S developed at Bell Labs. Later extended to S+.
- R is a free, open source version of S/S+.
- Under active cooperative development; versions change frequently.
- Very popular tool in statistics community: New methods often now implemented there.

Installing R

- CRAN: http://cran.r-project.org/
  - (R homepage: http://www.r-project.org/)
  - Windows 95 and later → Base
  - the latest release (R-2.15.2, 2012-10-26): http://cran.case.edu/
- Statlib http://lib.stat.cmu.edu/
- UCI Machine Learning: http://archive.ics.uci.edu/ml/
- Nice data mining course notes from Penn State:

http://sites.stat.psu.edu/~jiali/course/stat557/

Basics of R

- Command-driven language
- Data (and functions) stored as named objects
- Objects can be fairly simple (vectors, matrices) or more complex (assembled from other objects)
- RStudio(http://www.rstudio.com/): Powerful and Free IDE for R
  - Syntax highlighting, code completion, and smart indentation
  - Workspace browser and data viewer
  - Plot history, zooming, and flexible image and PDF export
  - Integrated R help and documentation

Install R, R Studio, R Rattle (clips to follow – thanks to Mai, Feng)

- Install R: http://www.youtube.com/watch?v=SJ9sVyqWJn8&hd=1
- Install RStudio: http://www.youtube.com/watch?v=6aTRbo7kdGk&hd=1
- Install Rattle and Explore Iris: http://www.youtube.com/watch?v=7P16fj0tqa4&hd=1
- Install R and introduction to R (DSI) http://www.youtube.com/watch?v=ZmtkqaRVTDc

Help and documentation

- Use ″An introduction to R″ (http://cran.r-project.org/doc/manuals/R-intro.pdf)
- Some recommended sections: 1.1, 1.7-1.11, 2, 5.1-5.4, 5.8, 6, 7.1, 10.1, 12.1, 12.3
- help(mean)
- help.search(″<subject>″)
- help.start()

Organizing your computations

- R has a ”current directory” (Working Directory)
  - To set Working Directory in RStudio: Session -> Set Working Directory
- Your objects (loaded datasets, variables, functions, etc.) are contained in your ”current workspace”, which can be saved any time
  - In Rstudio: Session -> Load Workspace/Save Workspace As
- Keep it tidy!
  - Simple way: Keep separate projects (code, data files) in separate workspaces/directories
  - More advanced: You can use Projects in Rstudio which comes with integration with version control tools such as Git or SVN (http://www.rstudio.com/ide/docs/using/projects)

More about R

- R workspace
  - Show what's been stored: ls() or objects()
  - Remove individual objects: rm(a, b, etc.)
  - Clear entire workspace: rm(list=ls())
    - In RStudio: Session -> Clear Workspace

- Quit: q() or file → quit R

- Up arrow repeats prior commands

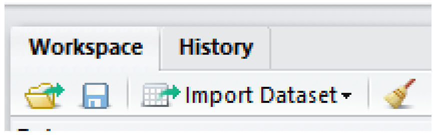
- R is case sensitive

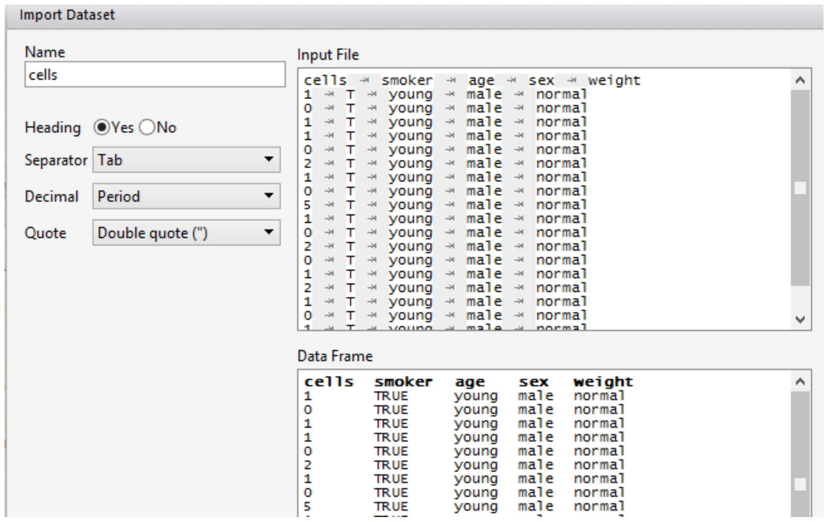
Assignments and some basic math

- The assignment operator consists of '<' and '-' and points from the expression to the name given to that expression
    - x <- 10 assigns the number 10 to the letter x
    - x = 10 does the same thing
- R as an over-qualified calculator
- log, exp
- mean, median, mode, max, min, sd
- Trigonometry:cos(x)
- Set operations: union(x, y)
- Logical operators: <, <=, >, >=, ==, !=

Import and export of data

- Useful functions: read.table, write.table, scan
- Works with mixed-type data (numbers and text columns)
- Works well with tab-separated data (connection to Excel)
- Easiest way: Import Dataset in RStudio





Name: cells

Heading: [✓ Yes] No

Separator: Tab

Decimal: Period

Quote: Double quote (")

Input File

```
cells ⇥ smoker ⇥ age ⇥ sex ⇥ weight
1 ⇥ T ⇥ young ⇥ male ⇥ normal
0 ⇥ T ⇥ young ⇥ male ⇥ normal
1 ⇥ T ⇥ young ⇥ male ⇥ normal
1 ⇥ T ⇥ young ⇥ male ⇥ normal
0 ⇥ T ⇥ young ⇥ male ⇥ normal
2 ⇥ T ⇥ young ⇥ male ⇥ normal
1 ⇥ T ⇥ young ⇥ male ⇥ normal
0 ⇥ T ⇥ young ⇥ male ⇥ normal
5 ⇥ T ⇥ young ⇥ male ⇥ normal
1 ⇥ T ⇥ young ⇥ male ⇥ normal
0 ⇥ T ⇥ young ⇥ male ⇥ normal
2 ⇥ T ⇥ young ⇥ male ⇥ normal
0 ⇥ T ⇥ young ⇥ male ⇥ normal
1 ⇥ T ⇥ young ⇥ male ⇥ normal
2 ⇥ T ⇥ young ⇥ male ⇥ normal
1 ⇥ T ⇥ young ⇥ male ⇥ normal
0 ⇥ T ⇥ young ⇥ male ⇥ normal
```

Data Frame

cells	smoker	age	sex	weight
1	TRUE	young	male	normal
0	TRUE	young	male	normal
1	TRUE	young	male	normal
1	TRUE	young	male	normal
0	TRUE	young	male	normal
2	TRUE	young	male	normal
1	TRUE	young	male	normal
0	TRUE	young	male	normal
5	TRUE	young	male	normal

Functions

- Most of R consists of functions
- The arguments can either be input in the right order, or using argument names
- Use help(...) !
- In Rstudio, hit tab after function name gives help about arguments

Vectors, matrices and data frames

- Calculations often done on vectors, matrices or data frames
  - Vectors: x = c(10,20,30,40)
  - Matrix: A = matrix(c(1,2,3,4), nrow = 2)
  - Data frames are the usual datasets, with variables names, can have both categorical and numerical variables

    data.frame(A) converts A into a dataframe
- Elementwise operations
- Subsetting, indexing
- Logical indexing

Vectors

- Vectors have length, but no dimension
    - >length(vector)
- >c() concatenates or combines numbers inside () into a vector or list
- >seq() generates sequence
    - Can specify length of sequence and increment
- >sort(vector) sorts items in vector
- Can use usual arithmetic and mathematical functions on vectors
- Logical vectors
- Character vectors
- Indexing a vector:
    - >vector[i]
    - >vector[-i]

Matrices

- Matrices have length and dimensions
  - >length(M); dim(M)
- Generating matrices
  - By combining vectors: rbind(); cbind()
  - >matrix() command
- Transpose
  - >t(M)
- Diagonal
  - >diag(M)
- Inverse
  - >solve(M)
- Multiplying matrices: M%*%N
- Indexing matrices

Data Frame

- >summary(data)
- >names(data); >attributes(data)
- Editing data
    - >fix(data) or >edit(data)
- >data$var select *var* from *data*
    - In RStudio, hit tab after data$ allows you to select/autocomplete variable names in *data*
    - Need to select more than 1 variable?

      >myvars = c('var1', 'var2')

      >data[,myvars]
    - You can also use numerical index as in matrix

      >data[,c(1,3)]

Lists

- ″Compound objects″ can often be constructed as lists
- Using $ to access parts of lists
- data.frame
- names

R packages

- A package: collection of functions (and data) concerning special application
- Contributed from different sources/persons
- Can be downloaded from CRAN or BioConductor
- must be ”loaded” with library()
- search()
- help(package = graphics)
- Also: Documentation from help.start()

Probability distributions

- norm, binom, beta, cauchy, chisq, exp, f, gamma, geom, hyper, lnorm, logis, nbinom, t, unif, weibull, wilcox
- Four prefixes:
  - ‘d’ for density (PDF)
    ```
    >dbinom(x=4,size=10,prob=0.5)
    ```
  - ‘p’ for distribution (CDF)
    ```
    >pnorm(1.86)
    ```
  - ‘q’ for quantile (percentiles)
    ```
    >qnorm(0.975)
    ```
  - ‘r’ for random generation (simulation)
    ```
    >rnorm(10)
    ```
- Each distribution has arguments that need to be specified
  - >rnorm(n=10,mean=100,sd=20)

Load External Files

- Stata, SPSS, SAS files
  - Library(foreign)
    - Stata: read.dta
    - SPSS: read.spss
    - SAS: read.xport (must first create export file in SAS)
- Excel files
  - Files must be saved as comma separated value or .csv
  - read.table, read.csv, read.csv2: identical except for defaults
- Watch the direction of '/'!
- >load(".Rdata")
- Loading and running R programs
    - >source(".R")

NA, NaN, and Null

- NA or “Not Available”
  - Applies to many modes – character, numeric, etc.
  - Missing values are NAs
- NaN or “Not a Number”
  - Applies only to numeric modes
- NULL
  - Lists with zero length

Lm, glm, nlm, optim

- Linear models
  - \>lm(y~x1+x2, data = dataset)
  - Example:

    \>data(cars)

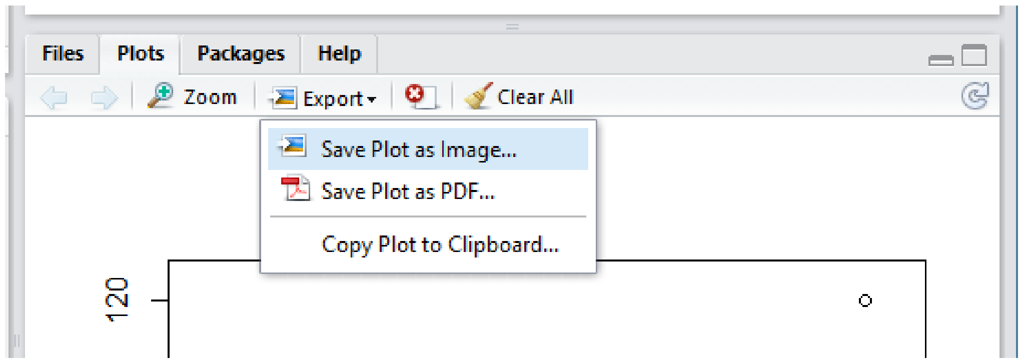
    \>lm(dist~speed, data=cars)
- Generalized linear models
  - \>glm(y~x1+x2, family="", data = dataset)
  - See help(family) for family options; family specifies error distribution
- Non-linear minimization
  - \>nlm(f, p) where f is the function to be minimized and p are the starting parameters
- Optimization
  - \>optim(par, fn) where par are the initial values and fn is the function to be minimized

Graphical visualization

- A ”generic” function: plot()
  - Producing Simple Graphs with R(http://www.harding.edu/fmccown/r/)
- High level commands, like pairs, image, contour…
- Lower level commands, adding stuff: points(), lines(), text(), title(), legend()…
- plotting characters pch, colors col…
- More advanced packages:
  - ggplot2 (http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/)
  - Lattice (http://www.isid.ac.in/~deepayan/R-tutorials/labs/04_lattice_lab.pdf)

Graphics Options

- >par() is used to set graphical options
  - mfrow and mfcol
- >plot(x,y)
  - >identify(); >abline()
  - Arguments: type, point style, labels, etc.
- Boxplot, histograms, sunflowerplots, piecharts, etc.
- Saving them
  - >dev.off() when things go wrong
  - Know the current directory
  - In RStudio



Writing your own functions

- Collecting commands into a function
- Arguments to function
- Returning result as a list
- Assignments within functions
- programming: conditional statements, loops (be careful, they are slow in R) etc…
- fix(<myfunction>) to edit a function

Write simple functions

- Writing a user function
  - >function(argument list){expressions}
  - Expressions or commands can be grouped together in {}, as they often are when writing functions.
  - Example:

```
abs_val = function(x){
 if(x >= 0){
  return(x)
 }
 else{
  return(-x)
 }
}
abs_val(-5)
```

- Conditionals
  - ‘if’, ‘else’
- Loops
  - ‘for’

Help!

- >help(package=stats)
  - Lists functions in the specified package
- >help(glm) or >?glm
- >help(“{“) for characters
- Help → Html help → Search Engine & Keywords
  - Potentially most helpful
- R documentation
  - command{package}
- More help:
  - Stackoverflow: http://stackoverflow.com/questions/tagged/r
  - R Help Mailing List: http://r.789695.n4.nabble.com/R-help-f789696.html

Other Resources

- The R Manuals under Documentation at http://www.r-project.org
- Mark Handcock's website at http://www.stat.washington.edu/handcock/567/links.html
- Jonathan Barron's R resources at http://finzi.psych.upenn.edu/
- Fox, John. 2002. *An R and S-PLUS Companion to Applied Regression*