

R Programming for Computational Linguists and Similar Creatures

Marco Baroni¹ and Stefan Evert²

¹Center for Mind/Brain Sciences
University of Trento

²Cognitive Science Institute
University of Osnabrück

Potsdam, 3-14 September 2007

General Information

R Basics

Basic functionalities

External files and data-frames

A simple case study: comparing Brown and LOB documents



Goals of the course

- ▶ Learn R basics and basic R programming
- ▶ Learn R implementations of various statistical/data analysis techniques useful in various domains of (computational) linguistics
- ▶ A little bit of background in statistics along the way
- ▶ Practice R skills on real-life data-sets



What this course is *not* about

- ▶ Statistical theory
- ▶ Specific statistical methods
- ▶ Cookbook recipes for specific analyses with R



What you should know

- ▶ Very basic math and statistics (vectors, logarithms, correlation, t-tests...)
- ▶ Some familiarity with programming/scripting and/or with a command-line environment
- ▶ Interest in (computational) linguistics issues



A tentative syllabus 1

Topics we will probably cover

- ▶ Introduction to R: set-up, data manipulation and exploration, plotting, basic statistics, input/output
- ▶ Using an R extension package: frequency distribution modeling with zipfR
- ▶ Co-occurrence statistics and frequency comparisons: contingency tables, association measures, evaluation
- ▶ Unsupervised multivariate data exploration: principal component analysis and clustering



A tentative syllabus 2

Further topics, to be selected depending on time and interests

- ▶ Supervised machine learning
- ▶ Matrix operations and linear algebra: application to the word space model
- ▶ More R programming: functions, list processing, non-interactive use
- ▶ Advanced 2D and 3D plots
- ▶ Generalized linear models, mixed effect models



Some useful R references for linguists

Available on the net, cover the theoretical and cookbook stuff we'll skip

- ▶ Shravan Vasishth, *The foundations of statistics: A simulation-based approach*
<http://www.ling.uni-potsdam.de/~vasishth/SFLS.html>
- ▶ Harald Baayen, *Analyzing Linguistic Data: A practical introduction to statistics*
<http://www.mpi.nl/world/persons/private/baayen/publications/baayenCUPstats.pdf>
 - ▶ (If you print this, you should commit yourself to buying the final published version.)



Some textbooks on statistics & R programming

- ▶ Peter Dalgaard, *Introductory Statistics with R*. New York: Springer, 2002.
- ▶ Morris H. DeGroot and Mark J. Schervish, *Probability and Statistics*, 3rd edition. Boston: Addison Wesley, 2002.
 - ▶ (Stefan's favourite statistics textbook.)
- ▶ Christopher Butler, *Statistics in Linguistics*. Oxford: Blackwell, 1985.
<http://www.uwe.ac.uk/hlss/llas/statistics-in-linguistics/bkindex.shtml>
 - ▶ (Out of print and available online for free download.)



Course materials

- ▶ Handouts, example scripts, data sets available online:
<http://www.ling.uni-potsdam.de/fallschool/z/>
- ▶ Homework assignments
 - ▶ mainly to encourage you to get practice with R :-)
 - ▶ required to get credit for the fall school
 - ▶ hand in solutions as plain text files by e-mail to fallschool.R@gmail.com



Outline

General Information

R Basics

- Basic functionalities
- External files and data-frames
- A simple case study: comparing Brown and LOB documents



R

- ▶ <http://www.r-project.org/>
- ▶ Free, open source development of the S language of Venables and Ripley
- ▶ Available for Linux, Mac and Windows
- ▶ Command-line interface and GUI (for Mac and Windows)
 - ▶ for Windows, we recommend www.sciviews.org GUI
- ▶ Non-interactive use possible via scripting
- ▶ Less user-friendly than other statistical software, but immensely more powerful
- ▶ A wealth of packages implementing impressive range of classic and cutting edge statistical and data analysis techniques available



General Information

R Basics

Basic functionalities

External files and data-frames

A simple case study: comparing Brown and LOB documents

```

> 1+1
[1] 2

> a <- 2      # assignment does not print anything by default

> a * 2
[1] 4

> log(a)      # natural, i.e. base-e logarithm
[1] 0.6931472

> log(a, 2)   # base-2 logarithm
[1] 1

```



Basic session management

Some of it is not necessary if you only use the GUI

to start R on command line, simply type R

setwd("path/to/data") # or use GUI menus

ls() # probably empty for now

ls # notice difference with previous line

quit() # or use GUI menus

quit(save="yes")

quit(save="no")

NB: at least some interfaces support history recall, tab completion

Vectorial math

```

> a <- c(1,2,3) # c (for combine) creates vectors

> a * 2      # operators are applied to each element of a vector
[1] 2 4 6

> log(a)     # also works for most standard functions
[1] 0.0000000 0.6931472 1.0986123

> sum(a)     # basic vector operations: sum, length, product, ...
[1] 6

> length(a)
[1] 3

> sum(a)/length(a)
[1] 2

```



Help!

```
> help("hist") # R has excellent online documentation
> ?hist      # short, convenient form of the help command

> help.search("histogram")

> ?help.search

> help.start() # searchable HTML documentation
```

or use GUI menus to access & search documentation

Your first R script

- ▶ Simply type R commands into a text file & save it
- ▶ Use built-in GUI functionality or external text editor
 - ▶ Microsoft Word is *not* a text editor!
 - ▶ nor is Apple's TextEdit application ...
- ▶ Execute R script from GUI editor or by typing

```
> source("my_script.R") # more about files later
> source(file.choose()) # select with file dialog box
```
- ▶ Just typing a variable name will not automatically print value in scripts: use `print(sd(a))` instead of `sd(a)`

Outline

General Information

R Basics

Basic functionalities

External files and data-frames

A simple case study: comparing Brown and LOB documents

Input from an external file

- ▶ We like to keep our data in space/tab delimited text files with a first row ("header") labeling the fields, like so:

```
word frequency cat
dog 15 noun
bark 10 verb
```
- ▶ This is an easy format to import into R, and it is easy to convert from other formats into this one using other tools
- ▶ We assume that external input is always in this format (or can easily be converted to it)
 - ▶ spreadsheet applications prefer CSV format (comma-separated values)

Procedure

- ▶ Collect basic summary statistics for the two corpora
- ▶ Check if there is significant difference in the token counts (since document length in tokens was controlled by corpus builders)
- ▶ If difference is significant (we will see that it is), then types are not truly comparable on doc-by-doc basis, and sentence lengths should be normalized (dividing by token count)
- ▶ Is word length correlated to document length? (in which case, corpus comparison would also not be appropriate)



Procedure

- ▶ Collect basic summary statistics for the two corpora
- ▶ Check if there is significant difference in the token counts (since document length in tokens was controlled by corpus builders)
- ▶ If difference is significant (we will see that it is), then types are not truly comparable on doc-by-doc basis, and sentence lengths should be normalized (dividing by token count)
- ▶ Is word length correlated to document length? (in which case, corpus comparison would also not be appropriate)
- ▶ Please read in the LOB data-set in a LOB data-frame and look at basic statistics
- ▶ Also, plot the data-frame for a quick look at relations between variables



Comparing token counts

```
> boxplot(brown$to,lob$to)
> boxplot(brown$to,lob$to,names=c("brown","lob"))
> boxplot(brown$to,lob$to,names=c("brown","lob"),
  ylim=c(1500,3000))
> ?boxplot

> t.test(brown$to,lob$to)
> wilcox.test(brown$to,lob$to)

> brown.to.center <- brown$to[brown$to > 2200
  & brown$to < 2400]
> lob.to.center <- lob$to[lob$to > 2200
  & lob$to < 2400]

> t.test(brown.to.center, lob.to.center)
```

how about sentence length?



Is word length correlated with token count?

token and type wl are almost identical:

```
> plot(brown$to,lob$to)
> cor.test(brown$to,lob$to)
> cor.test(brown$to,lob$to,
  method="spearman")
```

correlation with token count

```
> plot(brown$to,lob$to)
> cor.test(brown$to,lob$to)
```

