Whats’ beyond Concerto: An introduction to the R package \textit{catR}

Session 2:

Overview of R software and \textit{catR} package

The Psychometrics Centre, Cambridge, June 10th, 2014
Outline:

1. The R software
2. R GUI and RStudio
3. R language
4. R packages
1. The R software

R was created in 1997
Similar to the S language (used in e.g., S-PLUS)...
... but is open source and freely accessible
Network of developers and researchers in R
R website: http://www.r-project.org
Specific meetings and conferences (useR!)
Regular updates and personal improvements:
  • first version R-0.49 released April 23, 1997
  • last version R-3.1.0 released April 10, 2014
Increasing literature about R and its applications
1. The R software

R can handle a variety of tasks:

- **Descriptive statistics**
- Basic and advanced **figures**
- Basic and advanced **statistical methods** (ANOVA, PCA, factor analysis, GLM, bootstrap, neural networks, time series, ...)
- Easy data generation processes and design of simulation studies
- Interaction with other statistical **software** (e.g. WinBUGS)
- Easy loading and management of **data sets** from Excel, SAS, SPSS, ...
- Personal implementation and optimization possible (through e.g. packages)
1. The R software

Not all these features will be looked at today!

Some useful references and reading:

- ... and many other nice books from the *UseR!* series

Check also my slides from former workshop on R (Trier, Germany, 2013):

http://hdl.handle.net/2268/147096
2. R GUI and RStudio

If you install R from the CRAN (Comprehensive R Archive Network) website:

http://www.r-project.org

then you will install the GUI (graphical user interface) of R
R GUI is the basic interface to program in R...
... so it is not necessarily the most fancy interface :-}
2. R GUI and RStudio

R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

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.

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.

>
2. R GUI and RStudio

Fortunately (for you) it exists a more user-friendly interface: the RStudio software. RStudio can be downloaded and installed freely from https://www.rstudio.com/

RStudio works similarly to R GUI but optimizes windows and display information in a much more enhanced way. RStudio definitely to be preferred for non-crazy R users...

... though I will illustrate \texttt{catR} with R GUI :-)

2. R GUI and RStudio

Welcome to RStudio
Open source and enterprise-ready professional software for the R community

Powerful IDE for R
RStudio IDE is a powerful and productive user interface for R. It's free and open source, and works great on Windows, Mac, and Linux.

Web framework for R
Shiny is an elegant and powerful web framework for building interactive reports and visualizations using R — with or without web development skills.

Open source R packages
Our developers and expert trainers are the authors of several popular R packages, including ggplot2, plyr, lubridate, and others.
2. R GUI and RStudio

R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

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.

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()', 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
3. R language

R has a specific syntax and all instructions written in the console will be executed in a line-by-line approach.
Errors will be mentioned and execution will stop.

Basic example: let’s write ”3+2” in the console:

R> 3+2
   and execute it by pressing ENTER
(Important: R> must NOT be written in the console!)
Let’s see how R reacts...
3. R language

The following appears in the console:

```
> 3+2
[1] 5
> |
```

Surprisingly, R computed $3+2$ and returned value 5 :-)

Preceding symbol [1] helps in identifying the component number when a vector is returned.
3. R language

Now, let’s execute “3+a” in the console

R> 3+a

Output is

> 3+a

Error: object 'a' not found

Since object a was not yet created in R, it doesn’t exist and R detects an error.

Error message is returned, process stops and nothing more is done.
3. R language

Most often, analyses in R require several successive steps of calculation.
One mistake stops the whole process and all steps must be written once again...

More efficient to write all R code in a script file first, then copy-paste (or execute it) in the console.
Any text editing software (Word, Notepad etc.) is convenient, but R has an internal script editor (and RStudio opens automatically an R script window).
3. R language
3. R language

R version 3.0.0 (2013-04-03) -- "Masked Marvel"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTE
You are welcome to redistribute it under o
Type 'license()' or 'licence()' for distri
R is a collaborative project with many con
Type 'contributors()' for more informa
'citation()' on how to cite R or R packag
Type 'demo()' for some demos, 'help()' for
'help.start()' for an HTML browser interfa
Type 'q()' to quit R.

> 3 + 2
[1] 5
> 3 + a
Error: object 'a' not found
>
3. R language

Within the script window, any R code can be written
Nothing is being executed at once in the console
To execute some R code written in the script:

1. select the R code with the mouse
2. Click-right and select “Run line or selection”
   (this step can be replaced by CTRL+R)
3. R language

R language is based on several elements:

- predefined functions that are available internally
- additional functions available in packages (from CRAN or elsewhere)
- internal variables (to store data or output)
- vectors and matrices
- data frames and lists
3. R language

Functions are called by their name, and one or several arguments must be specified (in parentheses)

Example internal functions are:

- basic calculations: `exp()`, `log()`, `abs()`, `sqrt()`, `cos()`, `sin()`, ...
- graphics: `plot()`, `hist()`, `pie()`, `boxplot()`, ...
- basic statistics: `mean()`, `sd()`, `var()`, `quantile()`, `cor()`, ...
- distributions: `dnorm()`, `dchisq()`, `dpois()`, `dbinom()`, `dt()`, ...
- standard tests: `t.test()`, `chisq.test()`, `ks.test()`, `wilcox.test()`, ...
- advanced statistical models and methods: `glm()`, `prcomp()`, `factanal()`, `anova()`, `ksmooth()`, ...
3. R language

Some functions may not be available by default (see later)

Before going on, two important remarks:

- R makes distinction between lowercase and uppercase letters: `abs()` works, but not `Abs()`!
- Each function in R (either internally or from external package) has a help file with (sometimes) useful information
  – can be accessed by typing first
    ```
    ?
    ```
  – directly followed by the function name, without ()
  – for instance,
    ```
    R> ?anova
    ```
  opens the help file related to `anova()` function
3. **R language**

Internal variables may be created to store data and provisional or final results.

Variable name is up to you (remember that lowercase and uppercase are not the same!)

Storing some value in a variable is done by means of the logical symbol `<-`

Example: let’s write

```
R> x <- 5 + 2
```

R will create an internal variable `x` with value 7 (to see it, write down `x` in the console)
3. **R language**

**Matrices** can also be created and handled in R. Matrices are arrays with rows and columns. Elements of a matrix identified by means of two indicators, one for row number, one for column number (in this order).

Consider a $I \times J$ matrix stored in the R variable `mat`. Element $(i, j)$ of `mat` is obtained by

$$\text{mat}[i,j]$$

$i$-th row and $j$-th column of `mat` are obtained by

$$\text{mat}[i,\] \text{ and } \text{mat}[,.j]$$
3. R language

Matrices should contain only *numeric* values

However, data sets might hold mix of numeric and character / qualitative variables (such as category names)

Handled in R by means of *data frames*

Data frames are *lists* of R objects of different formats, most often vectors of numeric or character values of the *same length*

Display in a “data set” format with column names

Each column is an element of the list (i.e. a vector) and the column name is the variable name

*Hint*: most often, data sets loaded in R have the data frame format (see later)
4. R packages

R holds a lot of predefined functions for basic statistics and graphics.

However, R can also be used to develop specific routines... such as CAT applications.

Easiest way to get stable and portable R code: creation of an R package.

Packages hold several functions, with a full description (help file).

Most of them available from CRAN:

- 4465 packages on April 27th, 2013
- 5540 packages on May 14th, 2014
- ... how many on June 10th, 2014?
4. R packages

The R Project for Statistical Computing

Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred CRAN mirror.
- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.
4. R packages

The Comprehensive R Archive Network

Download and install R

Precompiled binary distributions of the base system and contributed packages, Windows and Mac users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- Sources of R, alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.
4. R packages

Contributed Packages

Available Packages

Currently, the CRAN package repository features 4465 available packages.

Table of available packages, sorted by date of publication

Table of available packages, sorted by name

Installation of Packages

Please type help("INSTALL") or help("install.packages") in R for information on how to install packages from this repository. The manual R Installation and Administration [PDF] (also contained in the R base sources) explains the process in detail.

CRAN Task Views allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 31 views are available.

Package Check Results

All packages are tested regularly on machines running Debian GNU/Linux, Fedora and Solaris. Packages are also checked under Mac OS X and Windows, but typically only on the day they appear on CRAN.

The results are summarized in the check summary (some timings are also available). Additional details for Windows checking and building can be found in the Windows check summary.

Writing Your Own Packages

The manual Writing R Extensions [PDF] (also contained in the R base sources) explains how to write new packages and how to contribute them to CRAN.

Repository Policies

The manual CRAN Repository Policy [PDF] describes the policies in place for the CRAN package repository.
4. R packages

Available CRAN Packages By Name

A: A3: Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
A: Tools for Approximate Bayesian Computation (ABC)
A: ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package
A: Approximate Bayesian Computational model for estimating P2
A: Algorithms for ABC summary statistics selection
A: The Analysis of Biological Data
A: Combine multi-dimensional arrays
A: A Biomarker Validation Suit for predicting Survival using gene signature
A: Data Modeling with Additive Bayesian Networks
A: Creation and evaluation of Acceptance Sampling Plans
A: ACC & LMA Graph Plotting
A: Categorical data analysis with complete or missing responses
A: Assay-based Cross-sectional Estimation of incidence rates
A: ace() and avas() for selecting regression transformations
A: The ACER Method for Extreme Value Estimation
A: Robust spline interpolation for dual color array comparative genomic hybridisation data
A: Affymetrix SNP probe-summation using non-negative matrix factorization
A: Convenience functions for analytical Customer Relationship Management
A: Download and manipulate data from the US Census American Community Survey
A: Actigraphy Data Analysis
A: Actuarial functions
A: Functions for actuarial scientists
A: ada: an R package for stochastic boosting
4. R packages

Best way to install an R package: make use of “Install” menu in R GUI / RStudio

Packages must be installed once and loaded with each new session of R

Packages must be re-installed after each update of R itself

To load an R package (for instance *catR*):

\[
R> \text{require(catR)}
\]

This allows R to access to all functions within *catR* package