

Statistics Environment R

Introduction, Overview, Applications

Hans Werner Borchers
Duale Hochschule Mannheim

Universität Kassel
February 11, 2013



What is R?

- R is “a **language** and **environment** for statistical computing and graphics”.
- R is Free Software under the terms of the GNU General Public License (GPL).
- **R Foundation of Statistical Computing**
- As an integrated suite for data analysis it includes:
 - well-developed high-level programming language.
 - extensible through packages and C/C++/Fortran codes
 - effective data handling and storage facilities,
 - large, integrated collection of tools for data analysis
 - graphical facilities for statistical (and other) plots
 - elaborate help system (text, PDF, HTML, LaTeX, ...)



History of R (as Open Source project)

History of R (as Open Source project)

- John Chambers: "S Programming Language"(1985/88)
ACM Software Award to J. Chambers (1998)

History of R (as Open Source project)

- John Chambers: “S Programming Language”(1985/88)
ACM Software Award to J. Chambers (1998)
- *S-PLUS* – commercial implementation of S (1988–2008)

History of R (as Open Source project)

- John Chambers: "S Programming Language"(1985/88)
ACM Software Award to J. Chambers (1998)
- *S-PLUS* – commercial implementation of S (1988–2008)
- Ross Ihaka and Robert Gentleman, Sidney (1995)
Environment for statistical computing (Mac)
"R: Language for Data Analysis and Graphics"(1996)

History of R (as Open Source project)

- John Chambers: "S Programming Language"(1985/88)
ACM Software Award to J. Chambers (1998)
- *S-PLUS* – commercial implementation of S (1988–2008)
- Ross Ihaka and Robert Gentleman, Sidney (1995)
Environment for statistical computing (Mac)
"R: Language for Data Analysis and Graphics"(1996)
- Martin Mächler hosts R on a server at the ETH Zürich
R Core Development Team (1998)
R becomes an official part of the GNU project

History of R (as Open Source project)

- John Chambers: "S Programming Language"(1985/88)
ACM Software Award to J. Chambers (1998)
- *S-PLUS* – commercial implementation of S (1988–2008)
- Ross Ihaka and Robert Gentleman, Sidney (1995)
Environment for statistical computing (Mac)
"R: Language for Data Analysis and Graphics"(1996)
- Martin Mächler hosts R on a server at the ETH Zürich
R Core Development Team (1998)
R becomes an official part of the GNU project
- R version 1.0.0 stable for production use (2000)

History of R (as Open Source project)

- John Chambers: "S Programming Language"(1985/88)
ACM Software Award to J. Chambers (1998)
- *S-PLUS* – commercial implementation of S (1988–2008)
- Ross Ihaka and Robert Gentleman, Sidney (1995)
Environment for statistical computing (Mac)
"R: Language for Data Analysis and Graphics"(1996)
- Martin Mächler hosts R on a server at the ETH Zürich
R Core Development Team (1998)
R becomes an official part of the GNU project
- R version 1.0.0 stable for production use (2000)

- R version 3.0 (April 2013)
> 5000 user contributed packages on CRAN

History of R (as Open Source project)

- John Chambers: "S Programming Language"(1985/88)
ACM Software Award to J. Chambers (1998)
- *S-PLUS* – commercial implementation of S (1988–2008)
- Ross Ihaka and Robert Gentleman, Sidney (1995)
Environment for statistical computing (Mac)
"R: Language for Data Analysis and Graphics"(1996)
- Martin Mächler hosts R on a server at the ETH Zürich
R Core Development Team (1998)
R becomes an official part of the GNU project
- R version 1.0.0 stable for production use (2000)
- ... CRAN ... R-Forge ... RStudio ... Rcpp ...
- R version 3.0 (April 2013)
> 5000 user contributed packages on CRAN

Personal Relations with R

- R user since 2001 on Windows, Mac OSX, Linux ... systems
- Answering questions on the R-help mailing list since 2001
- Several packages on R-Forge and CRAN ('pracma', 'numbers', 'adagio', 'nloptwrap', 'specfun', 'LowRankQP', 'linprog')
- Maintainer of the "Numerical Math" task view on CRAN
- Mentor of two R-GSoC 'Google Summer of Code' projects ("RMaxima"2012, "rHanso"2013)
- Reviewer (and author) for the Journal of Statistical Software (JSS) special issue on "Optimization with R"(2014)
- Contributor to a book on "Using R for Numerical Analysis", to appear in 2014, Chapman & Hall/CRC Press

R Resources

- R home and download pages (95 mirrors worldwide)
`http://www.r-project.org`, `cran.r-project.org`
- Contributed packages (sorted by name or date of publication)
`cran.r-project.org/web/packages/`
- Mailing lists (searchable)
`https://stat.ethz.ch/pipermail/r-help/`,
`stackoverflow.com/questions/tagged/r/`
- Task Views
`cran.r-project.org/web/views/`
- R Journal and the Journal of Statistical Software
`journal.r-project.org/`, `www.jstatsoft.org/`
- Books related to R (a.o., Springer UseR Series, CRC R Series)
`www.R-project.org/doc/bib/R-books.html`

Graphical User Interface: RStudio

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for a user analysis script. Lines 10-15 are highlighted in blue.


```

1 # User Analysis
2
3 setwd("~/analysis")
4 source("prep.R")
5
6 library(plyr)
7 library(lattice)
8 library(ggplot2)
9
10 # Import data set
11 rawdata <- read.csv("stats.csv")
12 dim(rawdata)
13
14 # Clean data set
15 clean <- prepareStats(rawdata)
16
17 # Subset of active users
18 active <- subset(clean, active == 1)
19 count(active, "daysSinceAccountCreated < 30")[2,2]
20 mean(active$age)
21
22
23
24

```
- Run Button:** A tooltip indicates: "Run the current line or selection (Ctrl+Enter)".
- Console:** Shows the execution output:


```

> library(plyr)
> library(lattice)
> library(ggplot2)
> # Import data set
> rawdata <- read.csv("stats.csv")
> dim(rawdata)
[1] 530750 35
> # Clean data set
> clean <- prepareStats(rawdata)
>
>

```
- Workspace:** Lists variables:
 - clean: 360404 obs. of 35 variables
 - rawdata: 530750 obs. of 35 variables
- Files:** Lists installed packages:
 - compiler
 - datasets
 - dichromat
 - dquest
 - evaluate
 - foreign
 - formatR
 - ggplot2
 - graphics
 - qrDevices

Reproducible Research

- “Combine technical report, data analysis, and experimental data s.t. research can be recreated, better understood and verified.”
- RStudio supports document preparation by
 - Markup languages: Markdown (HTML) or LaTeX (PDF)
 - knitr: literate programming
Presentation + analysis + executable code
 - Unix-like shell programs, e.g., GNU make, pandoc
 - Storing data, code and text on Git/Github repositories
- Minimal requirements for reproducible research: fingerprinting of data, reproducing random numbers, identification of R and package versions, reproducing system settings, etc., are not covered in this framework!

Help System

- Getting help for functions and operators
`?lm`, `?["["`, `??solve`, `help.start()`
- **Extended/-able help system** for package authors
 - structured help templates (Rd format)
 - automatic tests for correctness, checking examples
 - formatted inline help
 - automatic conversion to other formats: HTML, LaTeX, ...
 - generation of package manual in PDF format
- User and reference manuals easily available
- Web-based online help
Search facilities for package manuals and mailing lists
Online documentation: <http://www.Rdocumentation.org/>

Object Orientation

“Everything in R is an object.”

- **S3 classes**

‘informal classes’, using method dispatching on polymorphic functions
e.g., generic functions `summary` or `plot`: `methods(plot)`

- S4 classes

newer, fully object-oriented system [*seldom used*]

- OO in packages

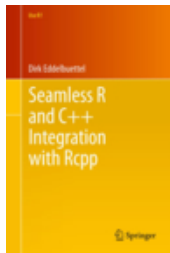
prototype-based: `proto`, R.OO; multiple inheritance: `mutatr`

- Reference Classes (Chambers)

mutable objects, message-passing mechanism (\leftrightarrow C++)

Integrating Fortran / C / C++

- Dynamically linked Fortran / C / C++ code can be directly used in R and is fully supported by package management and CRAN facilities
- 'Rcpp' provides an interface for seamlessly accessing, extending or modifying R objects at the C++ level
'Rinline': use uncompiled C++ code in R programs
- Other languages:
 - 'rJava': low-level R to Java interface
 - 'rPython': permits calls from R to Python
 - 'R.matlab': TCP/IP interface with the Matlab process



Package Management

- Packages are 'bundles' of functions, data files, help files, source files (Fortran, C, C++), vignettes, and additional information such as DESCRIPTION, NAMESPACE, NEWS
- **R builds and checks packages** with 50(!) different checks
- Source files are compiled and added as dynamically linked libraries
- User-contributed packages are stored on CRAN (> 5000) and are installed and updated from there:

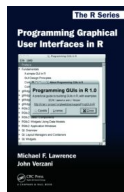
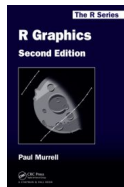
```
install.packages("pracma") ... update.packages()  
library(pracma)
```
- Packages can be stored and checked on R-Forge (subversion repository)

Data Sources

- Reading in data files: Text, CSV files (URL addresses)
`read.table(<file>, header=TRUE, sep='\t', ...)`
- Data files from other systems: *.mat (Matlab), Excel, ODS, SAS, SPSS, Stata, Systat, IDL(?), ...
- Database connections: DBI, ODBC, JDBC
Oracle, MySQL, PostgreSQL, **SQLite**
(non-SQL:) MongoDB, ArangoDB
- Internet data sources: Statistical databases, data archives
Finance, economics, ecological, census data, geographic, demographic, weather, medical, forensic, genomic/sequencing, ...
- Example:

Statistical Graphs

- Statistical Graphs are completely customized scatter plots, function graphs, density diagrams, pie charts, histograms, box and mosaic plots, spine/spinograms, ... advanced: grid graphics, 'pairs', 'coplots'
- Dynamic graphics: 'animation' package, 'rgobi'
- Importing and exporting graphics formats
- '**ggplot2**' (Grammar of Graphics) for professional plots
- *Missing Links*: high-quality 3D-Graphs, interactive graphs
- Building Graphical User Interfaces
Packages: '**gWidgets**', 'RGtk2', 'qtbase', 'tkltk'



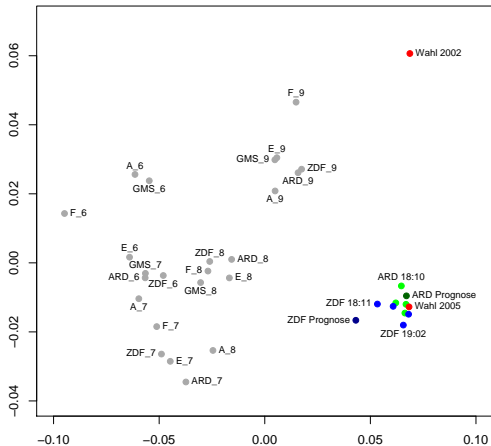
Computational Statistics

- Classical univariate Statistics:
Distributions, summaries, random data, robust methods, density estimation, hypotheses testing, ...
- **Linear regression**, generalized/additive linear models
Formula language for linear models: $y \sim x_1 + x_2 - 1$
- Nonlinear and smooth regression
- Multivariate Statistics:
cluster/factor/discriminant analysis, classification
visualization methods: PCA, MDS, SOM, ICA, etc.
- Tree-based methods, random and mixed effects
- Time series, survival analysis, spatial analysis



Example: Multidimensional Scaling

Bundestagswahl 2005



Multi-Dimensional Scaling (MDS)

Example: R code for MDS

```
umf <- read.table("umfragen.txt", sep="\t", header=T, row.names=1)
umf <- umf[,-6]

dumf <- dist(umf)
cumf <- cmdscale(dumf)

# plotting
clrs <- c(rep("darkgray", 24), "darkgreen", rep("green", 4),
          "darkblue", rep("blue", 4), rep("red", 2))
plot(cumf, pch=19, col=clrs,
     xlim=c(-0.10,0.10), ylim=c(-0.04,0.07),
     xlab=, ylab=,
     main="Bundestagswahl 2005",
     sub="Multi-Dimensional Scaling (MDS)")
identify(cumf[,1], cumf[,2], labels=row.names(umf), cex=0.75)
```

Time Series

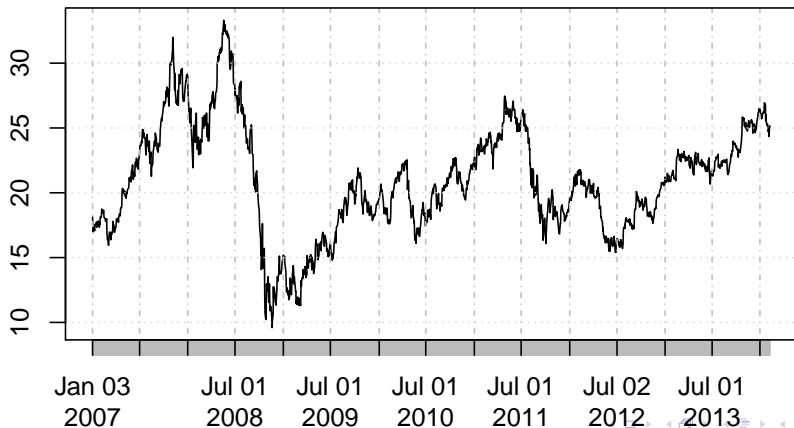
- Reading time series from different sources
 - text files, spreadsheets
 - Internet time series databases
 - PostgreSQL time series ('TSdbi')
 - Handling of time and date
- **Forecasting** and Modeling
 - Decomposition, seasonal trends, ARMA/ARIMA
 - spectral analysis, state space identification
- 'signal' package, but no 'control' package
- **Time Series Data Mining**
 - Clustering and classification
 - Dynamic time warping
 - Functional Data Analysis (FDA)



Example: Financial Time Series

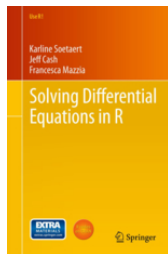
```
library(quantmod); getSymbols("ABB"); plot(ABB)
```

ABB

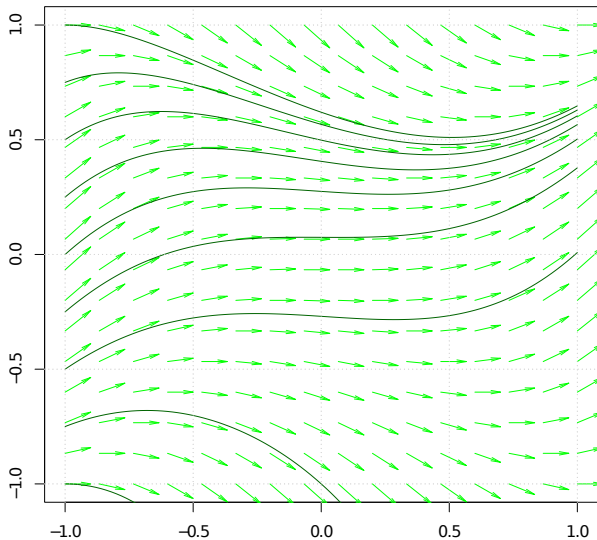


Differential Equations

- Several packages offer to solve differential equations, e.g. 'deSolve' or 'bvpSolve', mostly by integrating known free solvers written in C or Fortran:
 - Ordinary differential equations (ODEs)
 - Partial differential equations (PDEs)
 - Boundary value problems, e.g. reactive transport equations
 - Differential algebraic equations (DAEs)
 - Delay differential equations (DDEs)

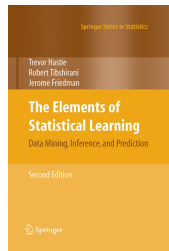


Example: Vectorfield



Statistical Learning

- Linear regression and classification
- Kernel smoothing methods (e.g., radial basis functions)
- **CART**, Decision Trees, **MARS**, Earth
- Neural Networks, Kohonen maps
- Support Vector Machines (classification, regression)
- **Random Forests**, ensemble learning
- Graphical models [, Bayesian networks]
- Boosting, AdaBoost, JackKnife methods
- Model assessment and selection



T. Hastie, R. Tibshirani, J. Freedman: *The Elements of Statistical Learning*
James, Witten, Hastie & Tibshirani: *An Introduction to Statistical Learning – With Applications in R*.

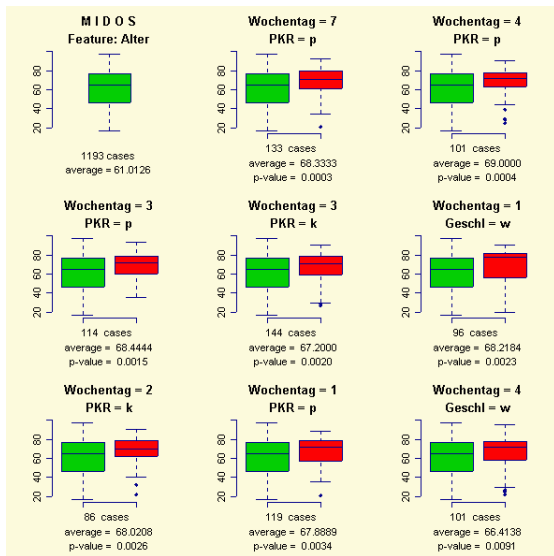
Coursera Course on “Statistical Learning” online since Jan 21, 2014

Data Mining with R

Database Mining...Data Mining...Predictive Analytics...Big Data...Data Science

- Packages providing Data Mining functionality:
 - **Weka**: open source machine learning software
 - Decision trees: Cubist, **C5.0** (Quinlan) [GritBot ?]
 - Random Forest (Breiman): ensemble learning
 - FactorMineR: sequence analysis
- Rattle: graphical user interface for data mining in R
- RODM: interface to Oracle Data Mining
- Vikamine: open source Subgroup Discovery
- Integration of R in Data Mining environments
(free) Weka, KNIME, RapidMiner; (comm.) SAS Enterprise Miner

Example: MIDOS Graph



Optimization with R (I)

- Optimization in Base R
`optim(par, fn, gr, lower, upper, method="Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN")`
- (Nonlinear) Least-Squares optimization: 'nls', 'nls', 'minpack.lm', ...
- Derivative-free optimization: 'dfoptim'
- (Mixed-integer) Linear Programming:
'lpSolve', 'Rglpk', 'Rsymphony', 'rneos' (NEOS server)
- Nonlinear Optimization: 'nloptr' (NLOpt library)
- Constraint Optimization: 'alabama', 'Rsolnp'
(using "augmented Lagrangian")

Optimization with R (II)

- Global Optimization: 'GenSA', 'PSO', 'soma', 'cmaes', 'DEoptim', 'RcppDE', 'DEoptimR' (differential evolution)
- Convex Optimization: 'Rcsdp' (semi-definite programming) CVX ?? ("disciplined convex programming") [Matlab toolbox]
- QP-QC (w/ quadratic constraints) ?? – MIQP ??
- Non-smooth Optimization: ?? (e.g., minimax problems)
- Discrete Optimization: 'knapsack', 'TSP'
- Modeling language ?? [MathProg, AMPL, GAMS, ZIMPL]

- Recommendations: See the "Optimization" task view !, or Special Issue "Optimization in R", J. of Stat. Software, 2014

Application Areas / Related Projects

- Econometrics / Financial Engineering: **Rmetrics**
“Financial Market Analysis w/ R” <<https://www.rmetrics.org/>>
- Bioinformatics: **Bioconductor** <www.bioconductor.org/>
“Analysis and comprehension of high-throughput genomic data”
- Spatial Statistics: Rgeo <www.R-project.org/Rgeo>
- Robust Statistics: <www.statistik.tuwien.ac.at/rsr/>
- Social Network Analysis: ‘igraph’, ‘sna’
- Optimization “community”

- See the ‘R task View’s <cran.r-project.org/web/views/>

High Performance / Parallel Computing

- High Performance Computing
 - 'Rcpp' family of packages, e.g., RcppArmadillo, RcppEigen
 - 'RcppOctave' – interface to Octave *and* Matlab
- **Parallel Computing**
 - Message Passing Interface (MPI)
 - Packages for explicit / implicit parallelism
 - Grid computing, GPUs
- Apache **Hadoop** framework (HDFS, MapReduce)
- Cloud Computing, e.g., Amazon Web / EC2 services
- Large memory / big data support / profiling tools

Matlab Advantages

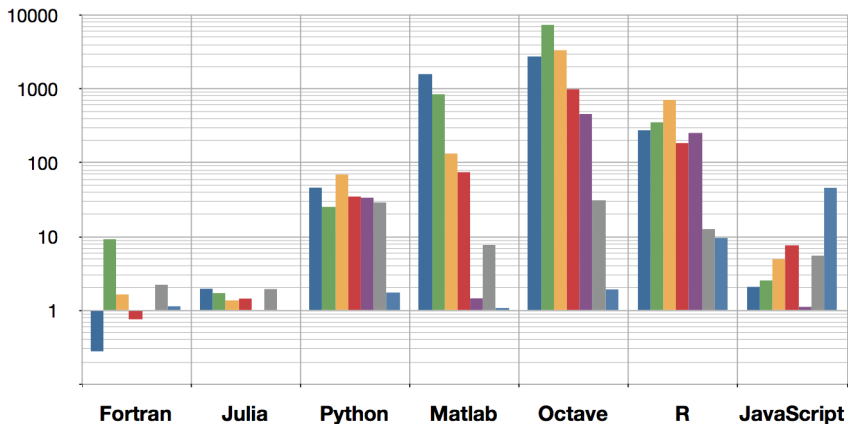
Matlab als Umgebung für techn.-wissenschaftl. Rechnen

- Graphische Bedienoberfläche
- Mathematische Operationen / Funktionen
- Eigene Programmiersprache
- 'Just-In-Time' (JIT) compiler
- Toolboxen, z.B. die 'Symbolic' Toolbox
- 2- und 3-dimensionale Visualisierungen
- Integration mit MS Office Paketen
- Erzeugung graphischer Benutzerschnittstellen
- Simulink: Simulation dynamischer Systeme
- C/C++ Compiler bzw. .NET Integration

Gang of Forty

Matlab Maple Mathematica SciPy SciLab IDL R
Octave S-PLUS SAS J APL Maxima Mathcad
Axiom Sage Lush Ch LabView O-Matrix PV-WAVE
Igor Pro OriginLab FreeMat Yorick GAUSS MuPad
Genius SciRuby Ox Stata JLab Magma Euler Rlab
Speakeasy GDL Nickle gretl ana Torch7

Obligatory Performance Slide



Execution time relative to C++

Some Examples of R and Matlab Syntax

Matlab

```
x = [1, 3, 5, 7, 9]
y = A(:, 2)
```

```
% defined in 'myfun.m'
function [a,b] = myfun(x,y,z)
...
end
```

```
if x == 1
    y = 0
else
    y = x
end
```

R

```
x <- c(1, 3, 5, 7, 9)
y <- A[, 2]
```

```
# defined interactively
myfun <- function(x,y,z=1e-0.7) {
    ...
}
```

```
if (x == 1) {
    y <- 0
} else {
    y <- x
}
```

Conclusions ?