Nice to have:

Peter Ruckdeschel

Abteilung Finanzmathematik Peter.Ruckdeschel@itwm.fraunhofer.de

Kaiserslautern, October 1, 2008

ITWM
Fraunhofer Institut
Techno- und
Wirtschaftsmathematik

Some (not so serious) remarks on R:

Early versions of R:

(acc.to Barry Rowlingson, Nov 7, 2003)

`www.jbum.com/idt/r.html`

`www.jbum.com/idt/hyperboreans.html`

. . . and after this talk:

(R-solution to programming quest `99-bottles-of-beer.ls-la.net`
involving S3-classes acc.to Peter Dalgaard, May 14, 2003)

```
f <- function(n) structure(c(paste(c(n,n,n-1),
                   "bottles of beer",c("on the wall"," ")),
                   "take one down, pass it around")[c(1,2,4,3)], class="verse")
print.verse <- function(x,...) cat(x, sep="\n")
lapply(99:1, f)
```


Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Some (not so serious) remarks on R:

Early versions of R:

(acc.to Barry Rowlingson, Nov 7, 2003)

`www.jbum.com/idt/r.html`

`www.jbum.com/idt/hyperboreans.html`

. . . and after this talk:

(R-solution to programming quest `99-bottles-of-beer.ls-la.net` involving S3-classes acc.to Peter Dalgaard, May 14, 2003)

```
f <- function(n) structure(c(paste(c(n,n,n-1),
                   "bottles of beer",c("on the wall"," ")),
                   "take one down, pass it around")[c(1,2,4,3)], class="verse")
print.verse <- function(x,...) cat(x, sep="n")
lapply(99:1, f)
```


Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

R — Statistics for Everyone?

- R is a comprehensive, open source programming environment
- provides tools for statistics and data analysis
- based on well-tested open source code
  - default numerical engine: `BLAS`, `www.netlib.org/blas/faq.html`,
    tunable: `ATLAS`, `ACML`, `Goto BLAS`..., cf. `cran.r-project.org/doc/manuals/R-admin.html`
  - uses large amount of `Statlib` code
- full-fledged programming language;
  paradigms: functional, vectorized, object orientated
- statistical models are implemented as language expressions
- professional graphics
- comprehensive import and export functionality
- see also: `www.r-project.org/about.html`
- shortly mentioned:
  - relation R and S; naming anecdotes
  - organization of R — R Core, R Community, UseR Conference Series

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

R — Statistics for Everyone?

- R is a comprehensive, open source programming environment
- provides tools for statistics and data analysis
- based on well-tested open source code
  - default numerical engine: `BLAS`, `www.netlib.org/blas/faq.html`,
    tunable: `ATLAS`, `ACML`, `Goto BLAS`..., cf. `cran.r-project.org/doc/manuals/R-admin.html`
  - uses large amount of `Statlib` code
- full-fledged programming language;
  paradigms: functional, vectorized, object orientated
- statistical models are implemented as language expressions
- professional graphics
- comprehensive import and export functionality
- see also: `www.r-project.org/about.html`
- shortly mentioned:
  - relation R and S; naming anecdotes
  - organization of R — R Core, R Community, UseR Conference Series

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

R — Statistics for Everyone?

- R is a comprehensive, open source programming environment
- provides tools for statistics and data analysis
- based on well-tested open source code
  - default numerical engine: `BLAS`, `www.netlib.org/blas/faq.html`,
    tunable: `ATLAS`, `ACML`, `Goto BLAS`..., cf. `cran.r-project.org/doc/manuals/R-admin.html`
  - uses large amount of `Statlib` code
- full-fledged programming language;
  paradigms: functional, vectorized, object orientated
- statistical models are implemented as language expressions
- professional graphics
- comprehensive import and export functionality
- see also: `www.r-project.org/about.html`
- shortly mentioned:
  - relation R and `S`; naming anecdotes
  - organization of R — R Core, R Community, UseR Conference Series

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

ad personam: J.M. Chambers

- 1998: ACM Software System Award to JMC, the principal designer of S, for the S system, which has forever altered the way people analyze, visualize, and manipulate data. . .
  S is an elegant, widely accepted, and enduring software system, with conceptual integrity, thanks to the insight, taste, and effort of JMC.
- cf. `http://cm.bell-labs.com/cm/ms/who/jmc/index.html`
- colored books:
  - "Blue Book": *The new S language. [. . .]* (Becker, JMC, Wilks, 1988),
    introduces `S Version 2`
  - "White Book": *Statistical models in* S (JMC, Hastie 1992),
    introduces `S Version 3` — new structures for model formulae in S
  - "Green Book": *Programming with Data* (JMC, 1998),
    introduces `S Version 4`
  - "Yellow Book": *Software for Data Analysis. [...]* (JMC, 2008),
    focus on R

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

ad personam: J.M. Chambers

- 1998: ACM Software System Award to JMC, the principal designer of S, for the S system, which has forever altered the way people analyze, visualize, and manipulate data. . .
  S is an elegant, widely accepted, and enduring software system, with conceptual integrity, thanks to the insight, taste, and effort of JMC.
- cf. `http://cm.bell-labs.com/cm/ms/who/jmc/index.html`
- colored books:
  - "Blue Book": *The new S language. [. . .]* (Becker, JMC, Wilks, 1988),

    introduces `S Version 2`
  - "White Book": *Statistical models in* S (JMC, Hastie 1992),

    introduces `S Version 3` — new structures for model formulae in S
  - "Green Book": *Programming with Data* (JMC, 1998),

    introduces `S Version 4`
  - "Yellow Book": *Software for Data Analysis. [...]* (JMC, 2008),

    focus on R

Obtaining R and contributed packages

Availability: [CRAN]=cran.r-project.org

- source code [CRAN]/src/base/R-2
  → compilable on any machine once standard tools are available

Linux RPMs and similar prepackaged formats for various Linux architectures — 32 and 64 bit [CRAN]/bin/linux

Win installers for Win32 and Win64 [CRAN]/bin/windows

Mac dmg files, [CRAN]/bin/macosx

Important sites

- the R project: www.r-project.org/
- CRAN — the comprehensive R archive network
- official mirrors under [CRAN]/mirrors.html
- infrastructure for collaborative package development: r-forge.r-project.org/

Obtaining R and contributed packages

Availability: [CRAN]=cran.r-project.org

- source code [CRAN]/src/base/R-2
  → compilable on any machine once standard tools are available

Linux RPMs and similar prepackaged formats for various Linux architectures — 32 and 64 bit [CRAN]/bin/linux

Win installers for Win32 and Win64 [CRAN]/bin/windows

Mac dmg files, [CRAN]/bin/macosx

Important sites

- the R project: www.r-project.org/
- CRAN — the comprehensive R archive network
- official mirrors under [CRAN]/mirrors.html
- infrastructure for collaborative package development: r-forge.r-project.org/

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Installing and Configuring R

Installation

- Reference: `[CRAN]/doc/manuals/R-admin.html`

Win use Windows installer
[in Vista, care has to be taken w.r.t. the way accounts and file permissions work]

Linux either RPM-like packages or building from source
[needs packages `blas`, `gcc-fortran`, `glibc`, `libgcc`, `libjpeg`, `libpng`, `xorg-x11-libs`, `zlib`, in suff. recent versions]

Mac for building R from source, certain tools are needed
(cf. r.research.att.com/tools)

Configuration

- environment variables:
  - location of temporary files: `TMPDIR`
  - output specification: `R_RD4PDF`, `R_RD4DVI`, `R_PAPERSIZE`,
  - libraries / where to look for packages `R_LIBS` `R_LIBS_SITE`, `R_LIBS_USER`
  - packages to load at startup `R_DEFAULT_PACKAGES`
  - localization `LANGUAGE` `LC_ALL`, `LC_MESSAGES` `LANG`
- configuration files in installation home directory `R_HOME/etc`:
  `Renviron` [only Unix], `Rconsole`, `Rprofile.site`
- user-individual settings in file `.Renviron`
  in the current or in the user's home directory (in that order)

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Installing and Configuring R

Installation

- Reference: `[CRAN]/doc/manuals/R-admin.html`

Win use Windows installer
[in Vista, care has to be taken w.r.t. the way accounts and file permissions work]

Linux either RPM-like packages or building from source
[needs packages `blas`, `gcc-fortran`, `glibc`, `libgcc`, `libjpeg`, `libpng`, `xorg-x11-libs`, `zlib`, in suff. recent versions]

Mac for building R from source, certain tools are needed
(cf. r.research.att.com/tools)

Configuration

- environment variables:
  - location of temporary files: `TMPDIR`
  - output specification: `R_RD4PDF`, `R_RD4DVI`, `R_PAPERSIZE`,
  - libraries / where to look for packages `R_LIBS` `R_LIBS_SITE`, `R_LIBS_USER`.
  - packages to load at startup `R_DEFAULT_PACKAGES`
  - localization `LANGUAGE` `LC_ALL`, `LC_MESSAGES` `LANG`
- configuration files in installation home directory `R_HOME/etc`:
  `Renviron` [only Unix], `Rconsole`, `Rprofile.site`
- user-individual settings in file `.Renviron`
  in the current or in the user's home directory (in that order)

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Installing Contributed Packages

- glimpse on `CRAN` — 1579 available packages as of Sep.29
- further sources:
  - Omegahat [Distributed Statistical Computing]
  - Bioconductor [Bio-Statistics]
  - Rmetrics [Fin. Market Analysis]
- `rss`-feed on new/updated packages: `CRANberries`
- Taskviews:
  - way of moderated sorting of packages by topic
  - installable and updatable
- local installation vs. download
- `R CMD INSTALL` vs `package.install`

Installing Contributed Packages

- glimpse on `CRAN` — 1579 available packages as of Sep.29
- further sources:
  - Omegahat [Distributed Statistical Computing]
  - Bioconductor [Bio-Statistics]
  - Rmetrics [Fin. Market Analysis]
- `rss`-feed on new/updated packages: `CRANberries`
- Taskviews:
  - way of moderated sorting of packages by topic
  - installable and updatable
- local installation vs. download
- `R CMD INSTALL` vs **package.install**

Running R – Batch mode vs. interactive mode vs. GUI

- interactive mode `R`, `Rgui` [Windows]
- batch mode: `R CMD BATCH`, `Rscript`, `Rterm` [Windows]
- scripting: littleR — hash-bang (#!) capability for the R Project.
- handy in Windows for several parallel R-versions:
  `batchfiles` by Gabor Grothendieck
- GUIs: see `www.sciviews.org/_rgui`
  - very popular and easily extensible: R Commander `Rcmdr` by John Fox
  - award winning: `JGR`
  - popular in data mining: `Rattle`

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

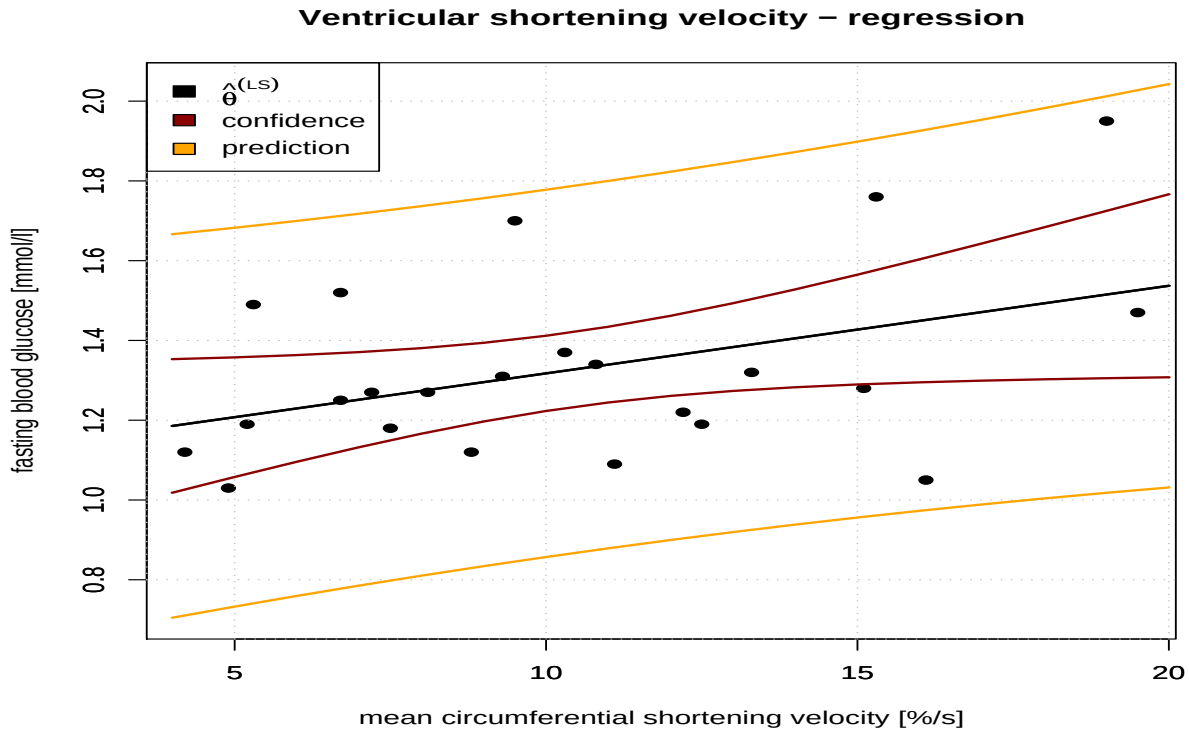
A First Session

```
##########################################################
# a simple regression example
# (adopted from M.Kohl's http://www.stamats.de/Regressionsmodelle.R)
##########################################################
## data from package ISwR (Introductory Statistics with R) by P. Dalgaard.
##########################################################
library(ISwR)

## load data set thuese
data(thuesen)
?thuesen
## attach it to the search path
attach(thuesen)
## a simple x-y plot for sh.velocity against blood.glucose
plot(short.velocity ~ blood.glucose, ylab = "fasting blood glucose [mmol/l]",
     xlab = "mean circumferential shortening velocity [%/s]",
     main = "Ventricular shortening velocity - data", pch = 20)
## a simple linear regression y = a + b*x + eps
## The LS estimate is implemented in function "lm"
fit1 <- lm(short.velocity ~ blood.glucose)
## information about this fit
str(fit1)
summary(fit1)

#######################################
```

A First Session II



further code see also R demonstration / handout

Documentation and References

Slides

- Marlene Müller's slides in the ITWM Wiki pages
- two semester course in German by P.R. and Matthias Kohl as `.pdf`; extra sessions/topics at `www.stamats.de/courses.htm`

Books

- see the lists in Marlene's slides and in our course
- a comprehensive list is `www.r-project.org/doc/bib/R-books.html`
- particular series *UseR* at Springer's

"Final Word" — the manuals

- An Introduction to R
- The R language definition
- Writing R Extensions
- R Data Import/Export
- R Installation and Administration
- R Internals

fun: R Graph Gallery

Documentation and References

Slides

- Marlene Müller's slides in the ITWM Wiki pages
- two semester course in German by P.R. and Matthias Kohl as `.pdf`; extra sessions/topics at `www.stamats.de/courses.htm`

Books

- see the lists in Marlene's slides and in our course
- a comprehensive list is `www.r-project.org/doc/bib/R-books.html`
- particular series *UseR* at Springer's

"Final Word" — the manuals

- An Introduction to R
- The R language definition
- Writing R Extensions
- R Data Import/Export
- R Installation and Administration
- R Internals

fun: R Graph Gallery

Documentation and References

Slides

- Marlene Müller's slides in the ITWM Wiki pages
- two semester course in German by P.R. and Matthias Kohl as `.pdf`; extra sessions/topics at `www.stamats.de/courses.htm`

Books

- see the lists in Marlene's slides and in our course
- a comprehensive list is `www.r-project.org/doc/bib/R-books.html`
- particular series *UseR* at Springer's

"Final Word" — the manuals

- An Introduction to R
- The R language definition
- Writing R Extensions
- R Data Import/Export
- R Installation and Administration
- R Internals

fun: R Graph Gallery

Help Sources

- R help within session by `?`, **help** **help.search** — various formats
- getting started: wiki.r-project.org/rwiki/doku.php?id=getting-started:getting-started
- translations from other languages: wiki.r-project.org/rwiki/doku.php?id=getting-started:translations:translations
- FAQ's at cran.r-project.org/faqs.html
- searching R-web sites e.g. with finzi.psych.upenn.edu/search.html and www.rseek.org
- searching mail archives with maths.newcastle.edu.au/~rking/R
- wikis: wiki.r-project.org
- newsletter: www.r-project.org/doc/Rnews
- mailing lists: www.r-project.org/mail.html

Editors and Literate Programming

Popular Editors

- (minimum) requirements:
  - pattern matching (braces, brackets,. . .)
  - search and replace with regular expressions
  - R syntax highlighting
  - possibly: code folding
  - spell checking
  - column / block marking
  - for `Sweave`: recognize `Schunks` & simultaneous highlighting of LaTeX commands
- most popular amongst developers: Emacs
  — `ESS` = "Emacs speaks Statistics" — also for Windows
- Windows: `WinEdt` with addon package `R-WinEdt`
- open source for Windows: `Tinn-R`
- further ones: SciTE, vim, context

Literate Programming

- `noweb` idea: code and report in one document
- realized by `Sweave` resp. `odfWeave`
- pretty printing / literate programming with R syntaxhighlighting with LaTeX package `listings`

Editors and Literate Programming

Popular Editors

- (minimum) requirements:
  - pattern matching (braces, brackets,. . .)
  - search and replace with regular expressions
  - R syntax highlighting
  - possibly: code folding
  - spell checking
  - column / block marking
  - for `Sweave`: recognize `Schunks` & simultaneous highlighting of LaTeX commands
- most popular amongst developers: Emacs
  — `ESS` = "Emacs speaks Statistics" — also for Windows
- Windows: `WinEdt` with addon package `R-WinEdt`
- open source for Windows: `Tinn-R`
- further ones: SciTE, vim, context

Literate Programming

- `noweb` idea: code and report in one document
- realized by `Sweave` resp. `odfWeave`
- pretty printing / literate programming with R syntaxhighlighting with LaTeX package `listings`

Package Concept in R

- goal: "packaging"/ assembling functions, data structures and data together with documentation
- standardized, platform-independent format in R

⇝ simple distribution and installation (with dependency checking)

- R provides tools for validation / checking
- documentation (in standard format) and examples enforced
- data sets are also documented acc. to scientific standards
- also: good idea to capsulate own work on a project for archiving
- References:
  - Writing R Extensions
  - Yellow book, ch. 4
  - our course 8.2
- larger documentation unit `vignette`,see our course, 8.2.6(j)

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Package Concept in R

- goal: "packaging"/ assembling functions, data structures and data together with documentation
- standardized, platform-independent format in R

⇝ simple distribution and installation (with dependency checking)

- R provides tools for validation / checking
- documentation (in standard format) and examples enforced
- data sets are also documented acc. to scientific standards
- also: good idea to capsulate own work on a project for archiving
- References:
  - Writing R Extensions
  - Yellow book, ch. 4
  - our course 8.2
- larger documentation unit `vignette`,see our course, 8.2.6(j)

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Preparations for Package Building

- mostly ready for start in Linux and Mac world, preparations needed in Windows, see

  `www.murdoch-sutherland.com/Rtools/`

- needed items
  - `Perl`
  - command line tools
  - MinGW compilers
  - LaTeX / `pdf`LaTeX
  - for Windows help: Microsoft HTML Help Workshop
  - (for building a distributable installer: Inno Setup installer)
- helper functions in R for template generation
  - package generation: **package**.skeleton
  - documentation: **prompt**, promptClass, promptMethods

Preparations for Package Building

- mostly ready for start in Linux and Mac world, preparations needed in Windows, see
  `www.murdoch-sutherland.com/Rtools/`
- needed items
  - `Perl`
  - command line tools
  - MinGW compilers
  - LaTeX / `pdf`LaTeX
  - for Windows help: Microsoft HTML Help Workshop
  - (for building a distributable installer: Inno Setup installer)
- helper functions in R for template generation
  - package generation: **package**.skeleton
  - documentation: **prompt**, promptClass, promptMethods

Preparations for Package Building

- mostly ready for start in Linux and Mac world, preparations needed in Windows, see
  `www.murdoch-sutherland.com/Rtools/`
- needed items
  - `Perl`
  - command line tools
  - MinGW compilers
  - LaTeX / `pdf`LaTeX
  - for Windows help: Microsoft HTML Help Workshop
  - (for building a distributable installer: Inno Setup installer)
- helper functions in R for template generation
  - package generation: **package**.skeleton
  - documentation: **prompt**, promptClass, promptMethods

Object Orientation in S/R

- generalties to OOP not necessary: You are the experts!
- OOP in S/R — a different paradigm:
  *F*unction-orientated- *FOOP* as opposed to *COOP*
  - methods *not* part of object but managed by *generic functions*
  - depending on the arguments different methods are dispatched
  - example: **plot**

advantages:

OOP in R general interfaces (c.f. **lm**, **glm**, rlm) possible

OOP in R by dispatching mechanism on run-time: general code using particularized methods

OOP in R code (may / will) be:
less redundant, better maintainable, better readable, better extensible

FOOP good for collaborative programming

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Object Orientation in S/R

- generalties to OOP not necessary: You are the experts!
- OOP in S/R — a different paradigm:
  *F*unction-orientated- *FOOP* as opposed to *COOP*
  - methods *not* part of object but managed by *generic functions*
  - depending on the arguments different methods are dispatched
  - example: **plot**

advantages:

OOP in R general interfaces (c.f. **lm, glm,** rlm) possible

OOP in R by dispatching mechanism on run-time: general code using particularized methods

OOP in R code (may / will) be:
less redundant, better maintainable, better readable, better extensible

FOOP good for collaborative programming

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Object Orientation in S/R

- generalties to OOP not necessary: You are the experts!
- OOP in S/R — a different paradigm:
  *F*unction-orientated- *FOOP* as opposed to *COOP*
  - methods *not* part of object but managed by *generic functions*
  - depending on the arguments different methods are dispatched
  - example: **plot**

advantages:

OOP in R general interfaces (c.f. **lm, glm,** rlm) possible

OOP in R by dispatching mechanism on run-time: general code using particularized methods

OOP in R code (may / will) be:
less redundant, better maintainable, better readable, better extensible

FOOP good for collaborative programming

Doing it the old way: S3 classes

- very informal
- basically: an object (normally a list) may be assigned a class attribute by

  class(<myobj>) ⟵ myClass
- gentleman's agreement:

  all objects of the class share the same structure
- object can be of several classes at one time
- generic functions like **print** defined e.g. as **function** (x, ...) **UseMethod**("print")
- dispatch by class attribute of first argument
- actual methods by naming convention `<name of generic>.<class name>`
- general method like **print.default**, particular method like **print.table**
- important examples: htest, **lm**

Doing it the old way: S3 classes

- very informal
- basically: an object (normally a list) may be assigned a class attribute by

  class(<myobj>) ⟵ myClass
- gentleman's agreement:

  all objects of the class share the same structure
- object can be of several classes at one time

- generic functions like **print** defined e.g. as **function** (x, ...) **UseMethod**("print")
- dispatch by class attribute of first argument
- actual methods by naming convention `<name of generic>.<class name>`
- general method like **print.default**, particular method like **print.table**

- important examples: htest, **lm**

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Doing it the old way: S3 classes

- very informal
- basically: an object (normally a list) may be assigned a class attribute by

  class(<myobj>) ⟵ myClass
- gentleman's agreement:

  all objects of the class share the same structure
- object can be of several classes at one time

- generic functions like **print** defined e.g. as **function** (x, ...) **UseMethod**("print")
- dispatch by class attribute of first argument
- actual methods by naming convention `<name of generic>.<class name>`
- general method like **print.default**, particular method like **print.table**

- important examples: htest, **lm**

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Let's be "modern": S4 classes

- more formal
- classes are defined by `setClass()` directive, e.g.

```
setClass("myS4class", representation(a="numeric", b="list"))
```

- there is type checking
- inheritance by `contains` argument in `setClass`
- generic functions are formally registered by `setGeneric()`, e.g.

```
setGeneric("myMethod", function(object) standardGeneric("myMethod"))
```

- actual methods are formally registered by `setMethod()`, e.g.

```
setMethod("myMethod", signature(object="ANY"), function(object) print(object))
```

- dispatch on more than on argument is possible
- for examples see the `distrXXX` family of packages

References:

- Yellow book, ch. 9, 10
- our course 8.1

Let's be "modern": S4 classes

- more formal
- classes are defined by `setClass()` directive, e.g.

```
setClass("myS4class", representation(a="numeric", b="list"))
```

- there is type checking
- inheritance by `contains` argument in `setClass`
- generic functions are formally registered by `setGeneric()`, e.g.

```
setGeneric("myMethod", function(object) standardGeneric("myMethod"))
```

- actual methods are formally registered by `setMethod()`, e.g.

```
setMethod("myMethod", signature(object="ANY"), function(object) print(object))
```

- dispatch on more than on argument is possible
- for examples see the `distrXXX` family of packages

References:

- Yellow book, ch. 9, 10
- our course 8.1

Let's be "modern": S4 classes

- more formal
- classes are defined by `setClass()` directive, e.g.

```
setClass("myS4class", representation(a="numeric", b="list"))
```

- there is type checking
- inheritance by `contains` argument in `setClass`
- generic functions are formally registered by `setGeneric()`, e.g.

```
setGeneric("myMethod", function(object) standardGeneric("myMethod"))
```

- actual methods are formally registered by `setMethod()`, e.g.

```
setMethod("myMethod", signature(object="ANY"), function(object) print(object))
```

- dispatch on more than on argument is possible
- for examples see the `distrXXX` family of packages

References:

- Yellow book, ch. 9, 10
- our course 8.1

Limitations of R: Speed

- R/S is interpreted $\longrightarrow$ tends to get slow in loops

$\Longrightarrow$ vectorize where possible — hints: **apply**&friends, Vectorize, mapply, rollapply

- analysis of runtime — profiling with
  Rprof, Rprofmem, `R CMD Rprof`, package `proftools`

References

- Writing R Extensions, Sec.3
  - our course 8.3.2
- delegate time-consuming parts to compiled languages
  like `FORTRAN` / `C` / `C++`
- alternative: compiling R-code itself
  - byte compiler (Luke Tierney,
    `www.stat.uiowa.edu/~luke/R/bytecode.html`)
  - just in time compiler (Stephen Milborrow,
    `www.milbo.users.sonic.net/ra/`)

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Limitations of R: Speed

- R/S is interpreted $\longrightarrow$ tends to get slow in loops

$\Longrightarrow$ vectorize where possible — hints: **apply**&friends, Vectorize, mapply, rollapply

- analysis of runtime — profiling with
  Rprof, Rprofmem, `R CMD Rprof`, package `proftools`

  References
  - Writing R Extensions, Sec.3
  - our course 8.3.2
- delegate time-consuming parts to compiled languages
  like `FORTRAN` / `C` / `C++`
- alternative: compiling R-code itself
  - byte compiler (Luke Tierney,
    `www.stat.uiowa.edu/~luke/R/bytecode.html`)
  - just in time compiler (Stephen Milborrow,
    `www.milbo.users.sonic.net/ra/`)

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Limitations of R: Speed

- R/S is interpreted $\longrightarrow$ tends to get slow in loops

$\Longrightarrow$ vectorize where possible — hints: **apply**&friends, Vectorize, mapply, rollapply

- analysis of runtime — profiling with
  Rprof, Rprofmem, `R CMD Rprof`, package `proftools`

  References
  - Writing R Extensions, Sec.3
  - our course 8.3.2
- delegate time-consuming parts to compiled languages
  like `FORTRAN` / `C` / `C++`
- alternative: compiling R-code itself
  - byte compiler (Luke Tierney,
    `www.stat.uiowa.edu/~luke/R/bytecode.html`)
  - just in time compiler (Stephen Milborrow,
    `www.milbo.users.sonic.net/ra/`)

Limitations of R: Speed

- R/S is interpreted $\longrightarrow$ tends to get slow in loops

$\Longrightarrow$ vectorize where possible — hints: **apply**&friends, Vectorize, mapply, rollapply

- analysis of runtime — profiling with
  Rprof, Rprofmem, `R CMD Rprof`, package `proftools`

  References
  - Writing R Extensions, Sec.3
  - our course 8.3.2
- delegate time-consuming parts to compiled languages
  like `FORTRAN` / `C` / `C++`
- alternative: compiling R-code itself
  - byte compiler (Luke Tierney,
    `www.stat.uiowa.edu/~luke/R/bytecode.html`)
  - just in time compiler (Stephen Milborrow,
    `www.milbo.users.sonic.net/ra/`)

Examples of Interfaces

- References:
  - Writing R Extensions, Sec.6
  - Yellow Book, ch. 12
  - our course 8.3

FORTRAN: .Fortran — interface to FORTRAN subroutine

C/C++: .C — interface to C/C++ function with return value void and only pointers to simple types as arguments

C/C++: .Call —interface to C/C++ function more flexible than .C, interfacing by R data type raw, in C/C++ type SEXP available; recommended for more complex objects as arguments/return values

CAVEAT
- matching of R and C/C++/FORTRAN variable types not obvious but necessary, see table in sec. 5.2 in Writing R Extensions
- explicit care of memory management

- enhanced comfort:
  in-lined FORTRAN / C / C++ code in package inline

pkg's to Perl: RSPerl
to Python: RSPython, Rpy [to R]
to Java: JRI [to R], rJava [from R], RSJava

Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik

Examples of Interfaces

- References:
  - Writing R Extensions, Sec.6
  - Yellow Book, ch. 12
  - our course 8.3

FORTRAN: **.Fortran** — interface to FORTRAN subroutine

C/C++: **.C** — interface to C/C++ function with return value `void` and only pointers to simple types as arguments

C/C++: **.Call** —interface to C/C++ function more flexible than **.C**, interfacing by R data type raw, in C/C++ type `SEXP` available; recommended for more complex objects as arguments/return values

CAVEAT
- matching of R and C/C++/FORTRAN variable types not obvious but necessary, see table in sec. 5.2 in Writing R Extensions
  explicit care of memory management
- enhanced comfort:
  in-lined FORTRAN / C / C++ code in package `inline`

pkg's to Perl: RSPerl
to Python: RSPython, Rpy [to R]
to Java: JRI [to R], rJava [from R], RSJava

Examples of Interfaces

- References:
  - Writing R Extensions, Sec.6
  - Yellow Book, ch. 12
  - our course 8.3

FORTRAN: **.Fortran** — interface to FORTRAN subroutine

C/C++: **.C** — interface to C/C++ function with return value `void` and only pointers to simple types as arguments

C/C++: **.Call** —interface to C/C++ function more flexible than **.C**, interfacing by R data type raw, in C/C++ type `SEXP` available; recommended for more complex objects as arguments/return values

CAVEAT
- matching of R and C/C++/FORTRAN variable types not obvious but necessary, see table in sec. 5.2 in Writing R Extensions
  explicit care of memory management
- enhanced comfort:
  in-lined FORTRAN / C / C++ code in package `inline`

pkg's to Perl: RSPerl
to Python: RSPython, Rpy [to R]
to Java: JRI [to R], rJava [from R], RSJava

Loading and Registering Object Code / Dynamic Libraries

Generation of Shared Libraries [shlib's](`.dll`, `.so`)

- recommendation: use `R CMD SHLIB` — do not try to do it yourself
- `R CMD SHLIB` accepts/treats object and source files acc. to ending

  `.o` [object files], `.c`, `.cc` [C], `.C`, `.cpp` [C++], `.f`, [FORTRAN77], `.f90`, `f95` [FORTRAN9x], `.m`, `.mm`, `.M` [Objective C++]
- use `Makevars`, `Makevars.win` to specify compiler flags
- before interfacing, shared library must be loaded in standard R code done by **dyn.load**, **dyn**.unload

Integration of Foreign Code to Packages

- special sub-directory `src` for foreign source code
- shlib's generated automatically by `R CMD build`/`R CMD INSTALL`
- within R packages, shlib's are usually loaded during package initialization by
  - **library**.dynam in .First.Lib (packages without name space)
  - useDynLib in .onLoad (packages with name space)

Registration of Foreign Code

- registration allows for management of entry points
- adds level of protection
- details see Writing R Extensions, "Registering native routines"

Loading and Registering Object Code / Dynamic Libraries

Generation of Shared Libraries [shlib's](`.dll`, `.so`)

- recommendation: use `R CMD SHLIB` — do not try to do it yourself
- `R CMD SHLIB` accepts/treats object and source files acc. to ending

  `.o` [object files], `.c`, `.cc` [`C`], `.C`, `.cpp` [`C++`], `.f`, [`FORTRAN77`], `.f90`, `f95` [`FORTRAN9x`], `.m`, `.mm`, `.M` [`Objective C++`]
- use `Makevars`, `Makevars.win` to specify compiler flags
- before interfacing, shared library must be loaded in standard R code done by **dyn.load**, **dyn**.unload

Integration of Foreign Code to Packages

- special sub-directory `src` for foreign source code
- shlib's generated automatically by `R CMD build`/`R CMD INSTALL`
- within R packages, shlib's are usually loaded during package initialization by
  - **library**.dynam in .First.Lib (packages without name space)
  - useDynLib in .onLoad (packages with name space)

Registration of Foreign Code

- registration allows for management of entry points
- adds level of protection
- details see Writing R Extensions, "Registering native routines"

Loading and Registering Object Code / Dynamic Libraries

Generation of Shared Libraries [shlib's](`.dll`, `.so`)

- recommendation: use `R CMD SHLIB` — do not try to do it yourself
- `R CMD SHLIB` accepts/treats object and source files acc. to ending

  `.o` [object files], `.c`, `.cc` [`C`], `.C`, `.cpp` [`C++`], `.f`, [`FORTRAN77`], `.f90`, `f95` [`FORTRAN9x`], `.m`, `.mm`, `.M` [`Objective C++`]
- use `Makevars`, `Makevars.win` to specify compiler flags
- before interfacing, shared library must be loaded in standard R code done by **dyn**.**load**, **dyn**.unload

Integration of Foreign Code to Packages

- special sub-directory `src` for foreign source code
- shlib's generated automatically by `R CMD build`/`R CMD INSTALL`
- within R packages, shlib's are usually loaded during package initialization by
  - **library**.dynam in .First.Lib (packages without name space)
  - useDynLib in .onLoad (packages with name space)

Registration of Foreign Code

- registration allows for management of entry points
- adds level of protection
- details see Writing R Extensions, "Registering native routines"

Limitations of R: Large Data Sets

- Memory limitations
  - on "usual machines" no more memory than 2 GB
  - virtual memory on Win32 limited to 4 GB
  - necessary to keep all data in memory at same time?
  - way out: split data into chunks;
    entire data set only available on persistant storage medium (hard disk)
- Addressing limitations
  - clue: $2^{32} = 4\text{GB} \Longrightarrow$ impossible to address more than 4 GB (at "once")...
  - usual R integer arithmetic is limited to 32bit
  - way out: multi-indices
- packages realizing these ideas
  `ff` and `R.ff`, `bigmemory`, `R.huge`, `biglm`

Limitations of R: Large Data Sets

- Memory limitations
  - on "usual machines" no more memory than 2 GB
  - virtual memory on Win32 limited to 4 GB
  - necessary to keep all data in memory at same time?
  - way out: split data into chunks;
    entire data set only available on persistant storage medium (hard disk)
- Addressing limitations
  - clue: $\mathbf{2^{32} = 4GB} \Longrightarrow$ impossible to address more than 4 GB (at "once")...
  - usual R integer arithmetic is limited to 32bit
  - way out: multi-indices
- packages realizing these ideas
  `ff` and `R.ff`, `bigmemory`, `R.huge`, `biglm`

Limitations of R: Large Data Sets

- Memory limitations
  - on "usual machines" no more memory than 2 GB
  - virtual memory on Win32 limited to 4 GB
  - necessary to keep all data in memory at same time?
  - way out: split data into chunks;
    entire data set only available on persistant storage medium (hard disk)
- Addressing limitations
  - clue: $\mathbf{2^{32} = 4GB} \Longrightarrow$ impossible to address more than 4 GB (at "once")...
  - usual R integer arithmetic is limited to 32bit
  - way out: multi-indices
- packages realizing these ideas
  `ff` and `R.ff`, `bigmemory`, `R.huge`, `biglm`

Way out: Database Connections

- in analogy to
  - Java's Database Connectivity (JDBC)
  - in C through the OpenDatabase Connectivity (ODBC)
  - Python's Database Application Programming Interface
  - Perl's Database Interface

  there is a common "frontend" package `DBI`, which interfaces to driver packages for interfaces to several data base architectures

  `RMySQL` [MySQL], `ROracle` [Oracle], `RSQLite` [SQLite],
  `RODBC` [ODBC, e.g. `MS Access`], `RPgSQL` [PostgreSQL] [yet to come]
- here somewhat more details just for `MySQL`
- installation: easy in Linux; in Windows: (no `.zip` file hosted on `CRAN`)

  installation of `MySQL`, generation of files `LIBMYSQL.def` and `libmySQL.a`; installation of `RMySQL` with `R CMD INSTALL`; see `README.windows` in `RMySQL` package, resp. our course, 8.3.4(b)
- use of `SQL` queries for data frames in R [without docking to data base]
  easy with package `sqldf`

Way out: Database Connections

- in analogy to
  - Java's Database Connectivity (JDBC)
  - in C through the OpenDatabase Connectivity (ODBC)
  - Python's Database Application Programming Interface
  - Perl's Database Interface

  there is a common "frontend" package `DBI`, which interfaces to driver packages for interfaces to several data base architectures

  `RMySQL` [MySQL], `ROracle` [Oracle], `RSQLite` [SQLite],
  `RODBC` [ODBC, e.g. `MS Access`], `RPgSQL` [PostgreSQL] [yet to come]
- here somewhat more details just for `MySQL`
- installation: easy in Linux; in Windows: (no `.zip` file hosted on `CRAN`)

  installation of `MySQL`, generation of files `LIBMYSQL.def` and `libmySQL.a`; installation of `RMySQL` with `R CMD INSTALL`; see `README.windows` in `RMySQL` package, resp. our course, 8.3.4(b)
- use of `SQL` queries for data frames in R [without docking to data base]
  easy with package `sqldf`

Way out: Database Connections II

- example: (taken from `RMySQL` package)

```
# create a MySQL instance and create one connection.
m ← dbDriver("MySQL")  ## or MySQL()
# open the connection using user, passsword, etc.,
# as specified in the "[iptraffic]" section of the
# configuration file \file{\$HOME/.my.cnf}
con ← dbConnect(m, group = "iptraffic")
rs ← dbSendQuery(con,
      "select * from HTTP_ACCESS where IP_ADDRESS = '127.0.0.1'")
df ← fetch(rs, n = 50)
dbHasCompleted(rs)
df2 ← fetch(rs, n = -1)
dbHasCompleted(rs)
dbClearResult(rs)
dim(dbGetQuery(con, "show tables"))
dbListTables(con)
```

Limitations of R: Computation Resources

- Herb Sutter: "CPU performance growth as we have known it hit a wall two years ago" (Moore's law in trouble!)
- easy/cheap way out: instead, processor companies add cores
- problem so far: R is single-threaded
- way out: (experimental) packages

`pnmath` [uses OpenMP compiler directives], `pnmath0` [uses pthreads]

by Luke Tierney (for use in Windows, see `README` files)

Network Computing

- even cheaper: do network/grid computing
- parallelization in R?
- packages for parallel/network/grid computing
- common architectures: `PVM`, `MPI`

Limitations of R: Computation Resources

- Herb Sutter: "CPU performance growth as we have known it hit a wall two years ago" (Moore's law in trouble!)
- easy/cheap way out: instead, processor companies add cores
- problem so far: R is single-threaded
- way out: (experimental) packages

`pnmath` [uses OpenMP compiler directives], `pnmath0` [uses pthreads]

by Luke Tierney (for use in Windows, see `README` files)

Network Computing

- even cheaper: do network/grid computing
- parallelization in R?
- packages for parallel/network/grid computing
- common architectures: `PVM`, `MPI`

Way out: Network Computations — R Packages for Parallelization

- `rpvm`
  - preparation: installation of PVM, setting paths
  - provides shell-skripts for communication between PVM and R
  - processes are started in R
  - both low-level and high-level tools for parallelization
- `Rmpi`
  - preparation: generation of a Beowulf cluster; installation of LAM-MPI
  - R-slaves is started from R
  - implements several MPI-functions for parallelization in R
- `snow` [breakthrough in R community] comfortable user-interface
  - based on either `rpvm` or `Rmpi`
  - at session start: initialization of PVM / MPI
  - important functions in `snow`
    - administration: makeCluster, stopCluster, clusterSetupSPRNG
    - high level routines: parLapply, parSapply, parApply
    - basic routines: clusterExport, clusterCall, clusterApply, clusterApplyLB [load balanced], clusterEvalQ, clusterSplit
- connector to `sfCluster`: `snowfall` — based on `snow`
- Grid-Computing with R: `GridR` [Fraunhofer!] (uses `Globus CoG` and `Gridge` toolkits), see also slides to tutorial
- `multiR` aims at heterogeneous architectures in High Throughput Distributed Computing (HTDC)

Way out: Network Computations — R Packages for Parallelization

- `rpvm`
  - preparation: installation of PVM, setting paths
  - provides shell-skripts for communication between PVM and R
  - processes are started in R
  - both low-level and high-level tools for parallelization
- `Rmpi`
  - preparation: generation of a Beowulf cluster; installation of LAM-MPI
  - R-slaves is started from R
  - implements several MPI-functions for parallelization in R
- `snow` [breakthrough in R community] comfortable user-interface
  - based on either `rpvm` or `Rmpi`
  - at session start: initialization of PVM / MPI
  - important functions in `snow`
    - administration: makeCluster, stopCluster, clusterSetupSPRNG
    - high level routines: parLapply, parSapply, parApply
    - basic routines: clusterExport , clusterCall , clusterApply, clusterApplyLB [load balanced], clusterEvalQ , clusterSplit
- connector to `sfCluster`: `snowfall` — based on `snow`
- Grid-Computing with R: `GridR` [Fraunhofer!] (uses `Globus CoG` and `Gridge` toolkits), see also slides to tutorial
- `multiR` aims at heterogeneous architectures in High Throughput Distributed Computing (HTDC)

Way out: Network Computations II

- further interesting projects:
  - R on Biowulf
  - Biocep-R — "Cloud Computing"
  - NetWorkSpaces, `nws`, see also manual
- Example for use of `snow`

```
require(boot); require(snow)
# generates cluster of 4 processors
cl ← makeCluster(4, type = "MPI")
# loads library(boot) to all R processes
clusterEvalQ(cl, library(boot))
# definition of the function to be called
ratio ← function(d, w) sum(d$x * w)/sum(d$u * w)
# do the bootstrap on the cluster
clusterCall(cl,boot,city,ratio, R=999,stype="w")
# stops the cluster
stopCluster(cl)
```

Way out: Network Computations II

- further interesting projects:
  - R on Biowulf
  - Biocep-R — "Cloud Computing"
  - NetWorkSpaces, `nws`, see also manual
- Example for use of `snow`

```
require(boot); require(snow)
# generates cluster of 4 processors
cl ← makeCluster(4, type = "MPI")
# loads library(boot) to all R processes
clusterEvalQ(cl, library(boot))
# definition of the function to be called
ratio ← function(d, w) sum(d$x * w)/sum(d$u * w)
# do the bootstrap on the cluster
clusterCall(cl,boot,city,ratio, R=999,stype="w")
# stops the cluster
stopCluster(cl)
```


Fraunhofer ITWM Institut Techno- und Wirtschaftsmathematik