

GESIS-Workshop

"Datenanalyse mit R"

Erste Schritte

Jan-Philipp Kolb

Montag, 15. Februar, 2016



Gliederung

R kam, sah und blieb

Warum R nutzen?

Dein Freund das GUI

Grundlagen im Umgang mit der Sprache R

R ist eine Objekt-orientierte Sprache

Verschiedene Datentypen

Indizieren

Wie bekommt man Hilfe?

Modularer Aufbau

Rein und raus – Datenimport und -export

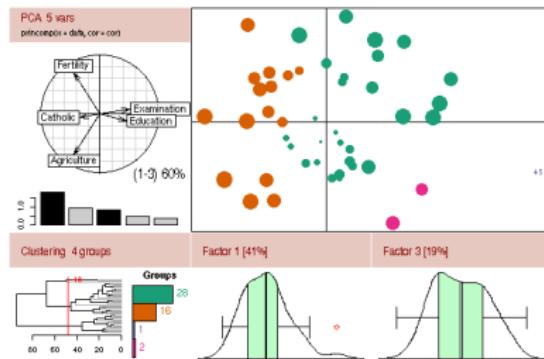
Datenimport

Datenexport

Ein erster Eindruck – Was uns die Daten sagen

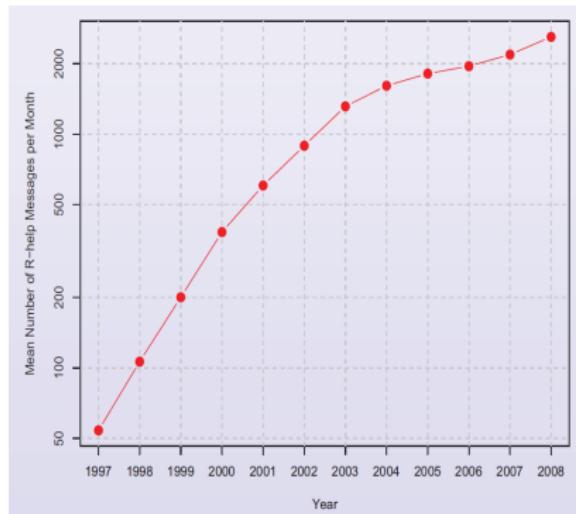
Open Source Programm R

- ▶ R ist eine freie, nicht-kommerzielle Implementierung der Programmiersprache S (von AT&T Bell Laboratories entwickelt)
- ▶ Freie Beteiligung ⇒ modularer Aufbau (immer mehr Erweiterungspakete)



www.r-project.org

Anzahl der Fragen in Hilfeforen zu R



x-Achse: 1997 : 2008

y-Achse: 50 : 2000

Quelle: The R User Conference 2008

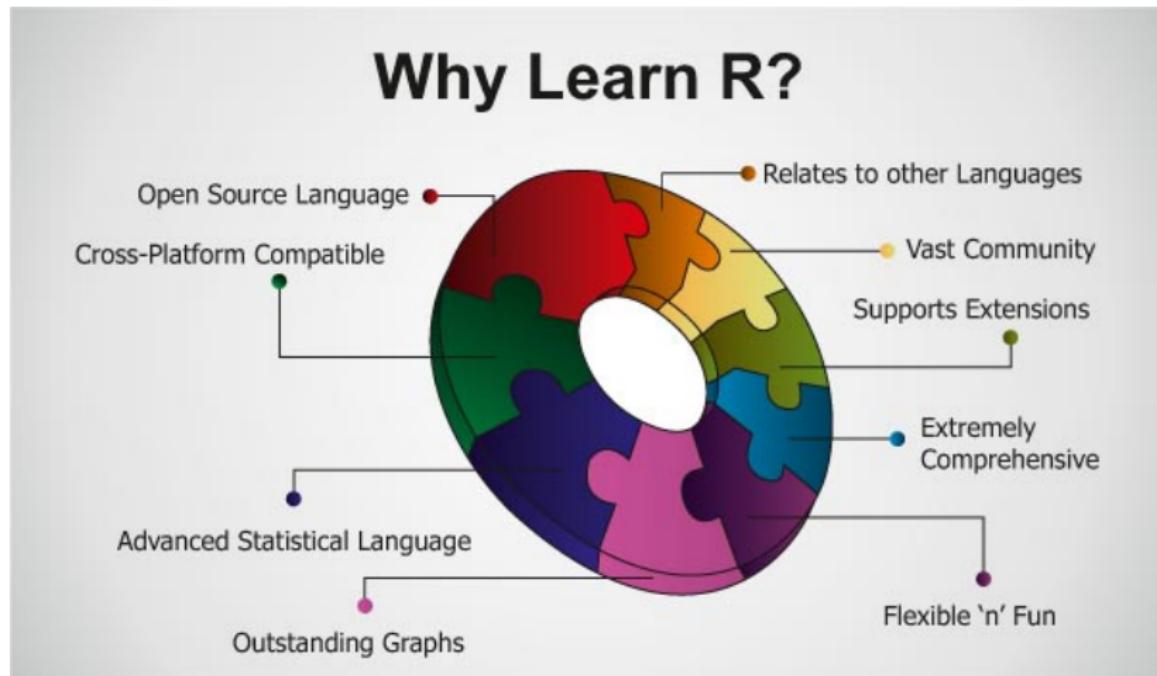
└ R kam, sah und blieb

└ Warum R nutzen?

Warum R nutzen?

- ▶ Als Weg kreativ zu sein ...
- ▶ Graphiken, Graphiken, Graphiken
- ▶ In Kombination mit anderen Programmen nutzbar
- ▶ Zur Verbindung von Datenstrukturen
- ▶ Zum Automatisieren
- ▶ Um die Intelligenz anderer Leute zu nutzen ;-)
- ▶ ...

Warum R nutzen?



└ R kam, sah und blieb

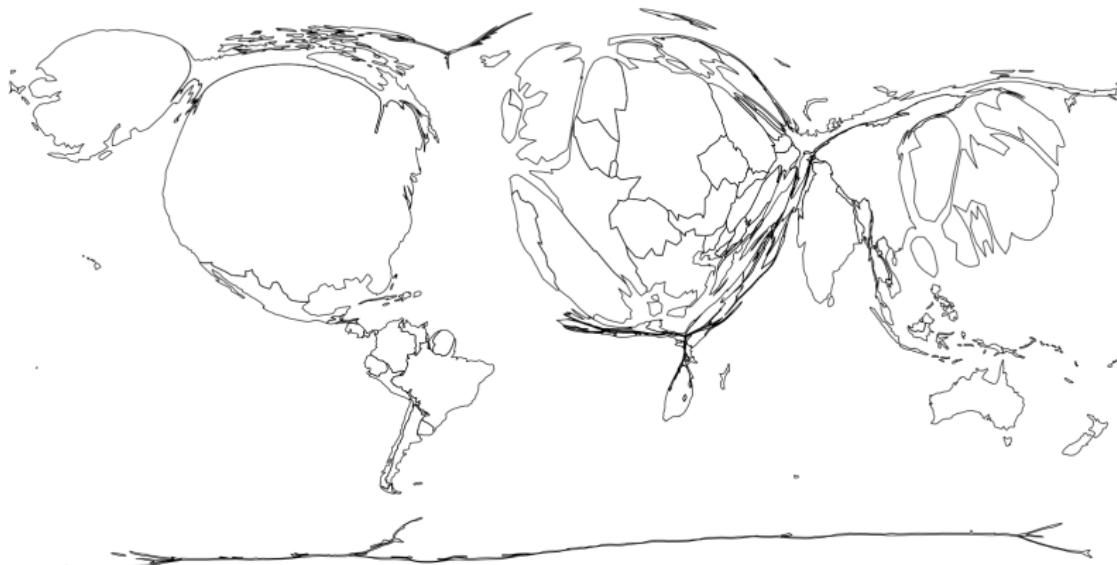
└ Warum R nutzen?

R-Nutzer rund um die Welt



R-Nutzer rund um die Welt

R Activity Around the World

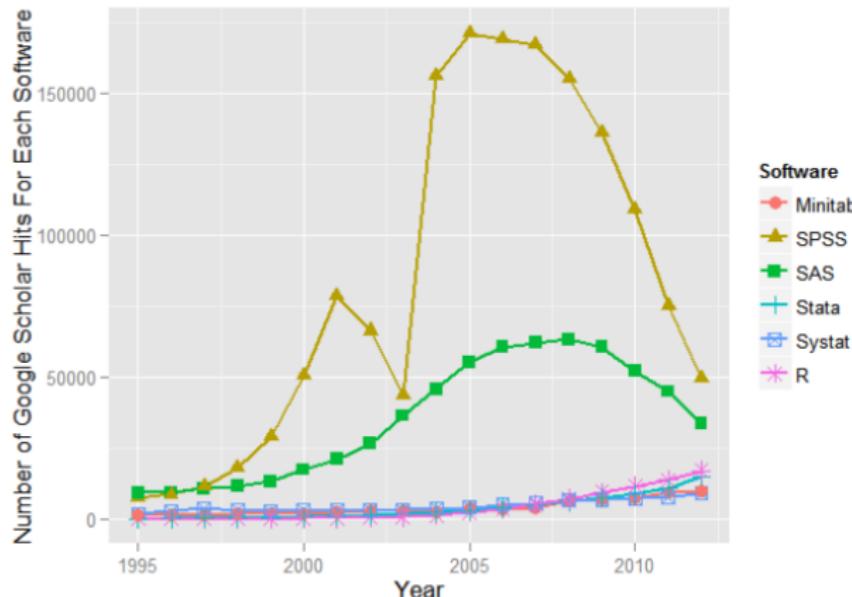


http://spatial.ly/2013/06/r_activity/

└ R kam, sah und blieb

└ Warum R nutzen?

Popularität von R



<http://r4stats.com/articles/popularity/>

└ R kam, sah und blieb

└ Warum R nutzen?

R lässt sich kombinieren...

The image displays five distinct screenshots related to R integration:

- Screenshot 1:** A red header section with the text "Use R!" at the top right, followed by author names "Richard M. Heiberger" and "Erich Neuwirth".
- Screenshot 2:** The "R-Java" logo, which features a stylized "R" above the word "Java".
- Screenshot 3:** The "SASmixed" logo, consisting of the words "SAS" and "mixed" in a bold, sans-serif font.
- Screenshot 4:** A screenshot of the R-Forge website for the "rPython R package". It shows the R-Forge logo and the package name "rPython R package".
- Screenshot 5:** Two stacked headers for a book or software. The top blue header reads "Statistics and Computing". The bottom yellow header reads "R for Stata Users".

Fruchtbare Kombinationen

- ▶ R und C++
[http://dirk.eddelbuettel.com/code/rcpp/
Rcpp-introduction.pdf](http://dirk.eddelbuettel.com/code/rcpp/Rcpp-introduction.pdf)
- ▶ R und github.com



└ R kam, sah und blieb

└ Warum R nutzen?

Kursunterlagen auf [github.com](https://github.com/Japhilko/IntroR)

<https://github.com/Japhilko/IntroR>

GitHub This repository Search Explore Features Enterprise Blog Sign up Sign in

Japhilko / IntroR Watch 1 Star 0 Fork 0

14 commits 1 branch 0 releases 1 contributor

branch: master IntroR +

Rearrange Course ...

Japhilko authored a minute ago Create own package latest commit a8cf1973dd

Rproj user/20FB1B1A/pcs

2015 Rearrange Course 5 days ago

README.md Update First Steps 44 seconds ago

README.md

Einführung in R 2015

Code Issues 1

Pulse

Graphs

HTTPS clone URL

<https://github.com/>

You can clone with HTTPS or Subversion

Clone in Desktop

Download ZIP

Erwartungen und Anforderungen

Das kann diese Schulung vermitteln:

- ▶ Eine praxisnahe Einführung in die statistische Programmiersprache **R**
- ▶ Erlernen einer Programmier-Strategie
- ▶ „Guten Stil“
- ▶ Die Vorzüge grafischer Datenanalyse

└ R kam, sah und blieb

└ Warum R nutzen?

Erwartungen und Anforderungen

Das kann sie nicht leisten:

- ▶ Eine Einführungsveranstaltung in die Statistik geben
- ▶ Grundlegende datenanalytische Konzepte vermitteln
- ▶ Verständnis zementieren
- ▶ Das „Trainieren“ abnehmen

└ R kam, sah und blieb

└ Warum R nutzen?

R herunterladen:



[Home]

[Download](#)

[CRAN](#)

[R Project](#)

[About R](#)

[Contributors](#)

[What's New?](#)

[Mailing Lists](#)

[Bug Tracking](#)

[Conferences](#)

[Search](#)

[R Foundation](#)

[Foundation](#)

[Board](#)

[Members](#)

[Donors](#)

[Donate](#)

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.

News

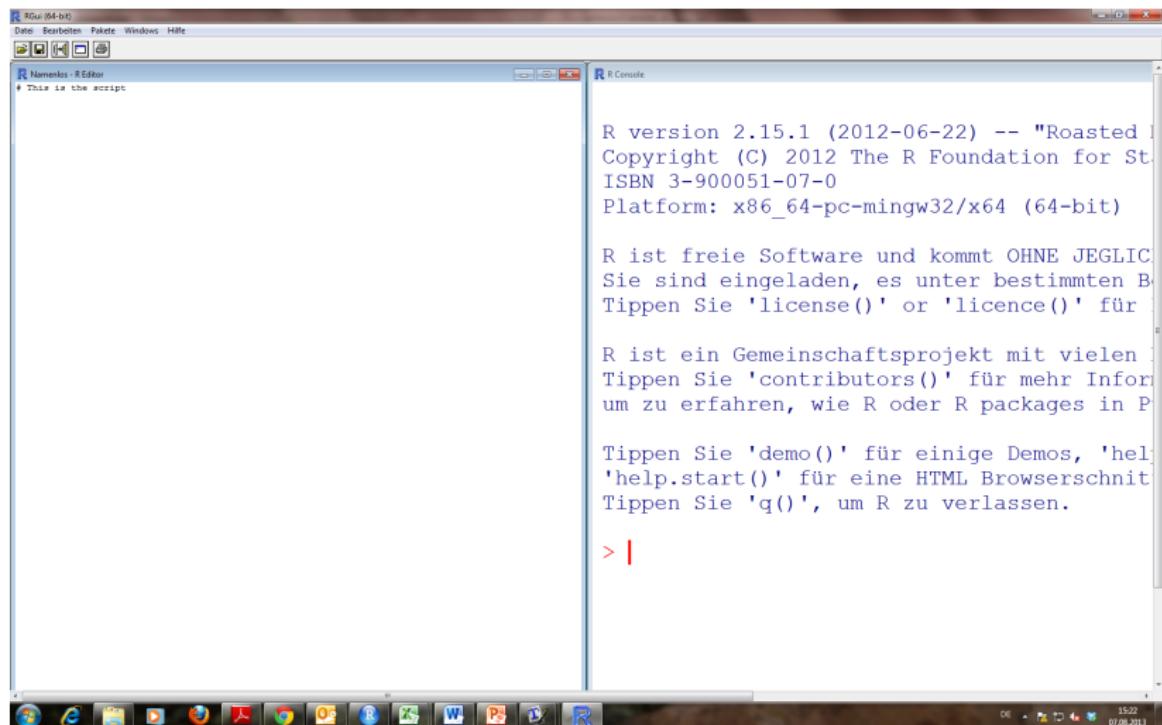
- [R 3.2.0 \(Full of Ingredients\) prerelease versions](#) will appear starting March 19. Final release is scheduled for 2015-04-16.
- R version 3.1.3 (Smooth Sidewalk) has been released on 2015-03-09.
- [The R Journal Volume 6/2](#) is available.
- R version 3.1.2 (Pumpkin Helmet) has been released on 2014-10-31.
- [useR! 2015](#), will take place at the University of Aalborg, Denmark, June 30 - July 3, 2015.
- [useR! 2014](#), took place at the University of California, Los Angeles, USA June 30 - July 3, 2014.

<http://www.r-project.org/>

└ R kam, sah und blieb

└ Warum R nutzen?

Das ist das Basis-R:



The screenshot shows the RGui (64-bit) application window. The title bar says "RGui (64-bit)". The menu bar includes "Datei", "Bearbeiten", "Pakete", "Windows", and "Hilfe". There are two panes: the left pane is titled "R Namenslos - R Editor" with the text "# This is the script.", and the right pane is titled "R Console" showing the R startup message and help text.

```
R version 2.15.1 (2012-06-22) -- "Roasted I
Copyright (C) 2012 The R Foundation for St.
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)

R ist freie Software und kommt OHNE JEGLIC
Sie sind eingeladen, es unter bestimmten B
Tippen Sie 'license()' or 'licence()' für

R ist ein Gemeinschaftsprojekt mit vielen
Tippen Sie 'contributors()' für mehr Infor
um zu erfahren, wie R oder R packages in P

Tippen Sie 'demo()' für einige Demos, 'hel
'help.start()' für eine HTML Browserschnit
Tippen Sie 'q()', um R zu verlassen.

> |
```

Gliederung

R kam, sah und blieb

Warum R nutzen?

Dein Freund das GUI

Grundlagen im Umgang mit der Sprache R

Rein und raus – Datenimport und -export

Ein erster Eindruck – Was uns die Daten sagen

Aber die meisten Menschen nutzen einen Editor oder ein *graphical user interface (GUI)*

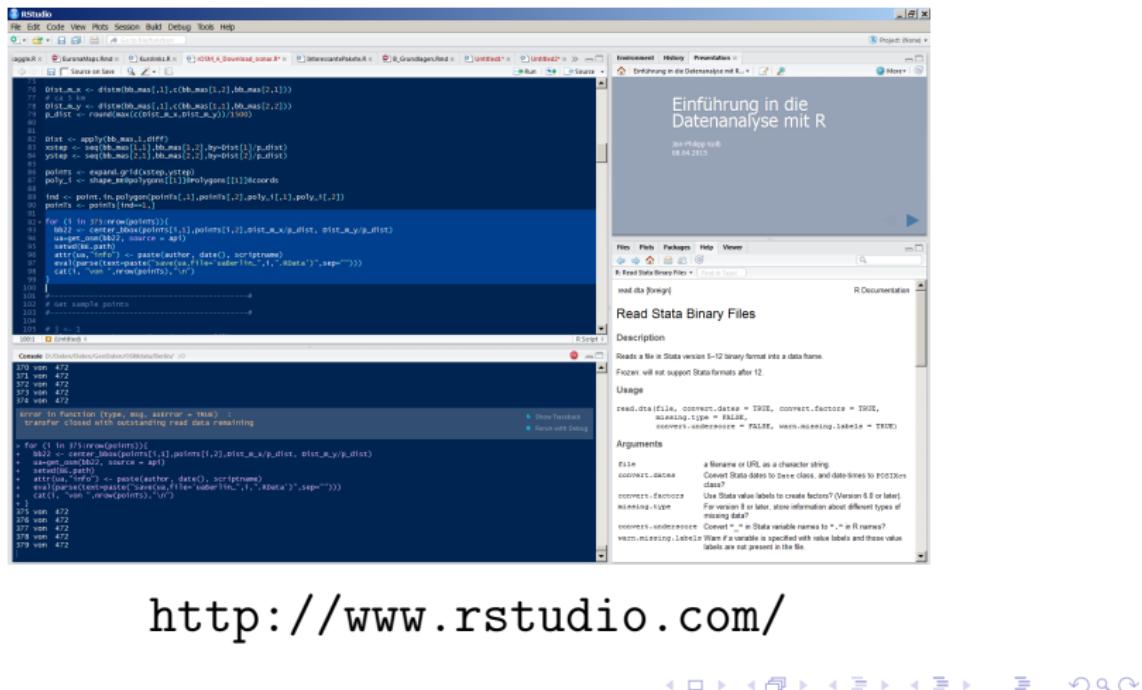
Aus den folgenden Gründen:

- ▶ Syntax highlighting
- ▶ Auto-Vervollständigung
- ▶ Bessere Übersicht über Graphiken, Bibliotheken

- ▶ Gedit mit R-spezifischen Add-ons für Linux
<https://projects.gnome.org/gedit/>
- ▶ Emacs
<http://www.gnu.org/software/emacs/>
- ▶ TinnR
<http://www.sciviews.org/Tinn-R/>
- ▶ ...

- └ R kam, sah und blieb
 - └ Dein Freund das GUI

Ich nutze Rstudio!



- └ R kam, sah und blieb
- └ Dein Freund das GUI

Download der Unterlagen

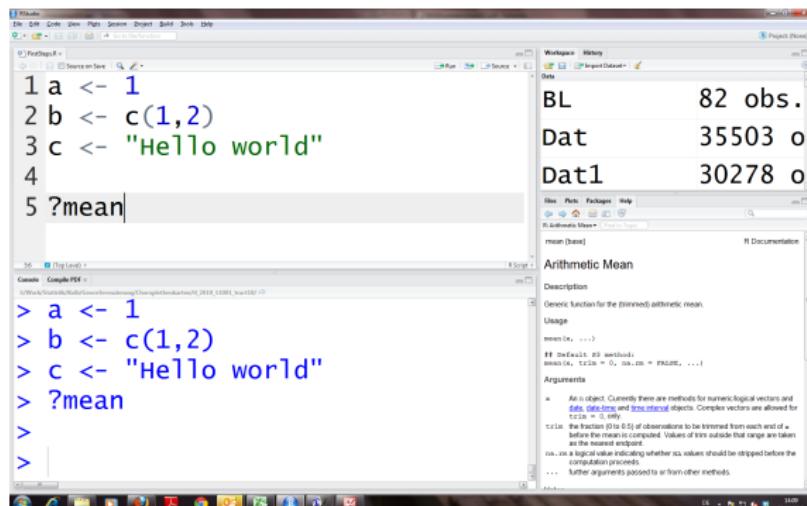
github.com/Japhilko/IntroR/

The screenshot shows the GitHub repository page for 'Japhilko / IntroR'. At the top, there's a navigation bar with 'GitHub' logo, search bar, and links for 'Explore', 'Features', 'Enterprise', and 'Blog'. On the right are 'Sign up' and 'Sign in' buttons. Below the header, the repository name 'Japhilko / IntroR' is displayed with a fork icon. To the right are buttons for 'Watch 1', 'Star 0', and 'Fork 0'. The main content area shows summary statistics: 13 commits, 1 branch, 0 releases, and 1 contributor. A red horizontal bar highlights the '13 commits' section. Below this, a dropdown menu shows the current branch is 'master'. The commit history table lists several commits:

Author	Message	Date
Japhilko	authored a day ago	latest commit d8086d3d4f
Rproj user/20FB1B1A/pcs	Create own package	4 days ago
RintroMA	update presentation	4 days ago
data	Data import	a day ago
figure	Import Export	a day ago
slides	Import Export	a day ago
Rhistory	Create own package	4 days ago

On the right sidebar, there are sections for 'Code', 'Issues (1)', 'Pull requests (0)', 'Pulse', 'Graphs', and a 'HTTPS clone URL' field containing the URL <https://github.com/Japhilko/IntroR>. There's also a note about cloning with HTTPS or Subversion.

- └ R kam, sah und blieb
- └ Dein Freund das GUI



Die verschiedenen Fenster in Rstudio

- └ R kam, sah und blieb
- └ Dein Freund das GUI

The screenshot shows the RStudio interface. On the left, the 'Script Editor' window contains the following R code:

```
1 a <- 1
2 b <- c(1,2)
3 c <- "Hello world"
4
5 ?mean
```

The line `?mean` is highlighted with a red rectangle. On the right, the 'Help Viewer' window displays the documentation for the `mean` function, titled 'Arithmetic Mean'. It includes sections for 'Description', 'Details', 'Usage', 'Value', 'Author', 'References', and 'Arguments'. The 'Details' section notes that `mean(x)` is equivalent to `mean.default(x)`.

Die verschiedenen Fenster in Rstudio Das Script

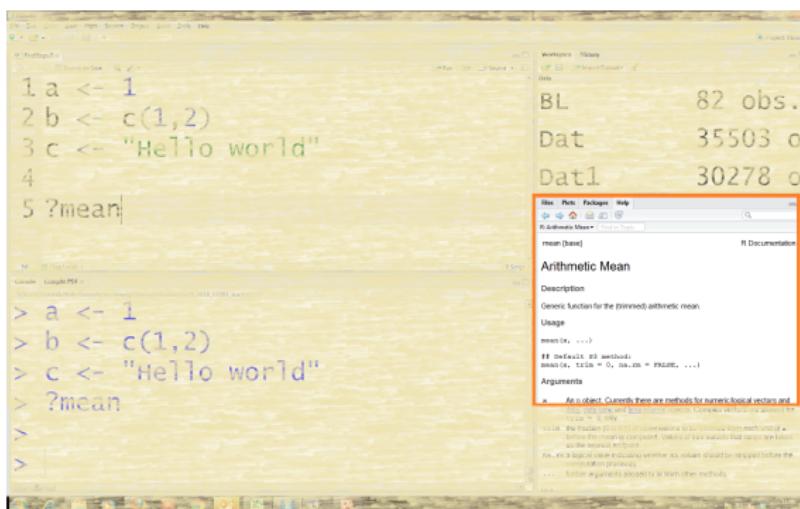
- └ R kam, sah und blieb
- └ Dein Freund das GUI



Die verschiedenen Fenster in Rstudio

Konsole

- └ R kam, sah und blieb
- └ Dein Freund das GUI



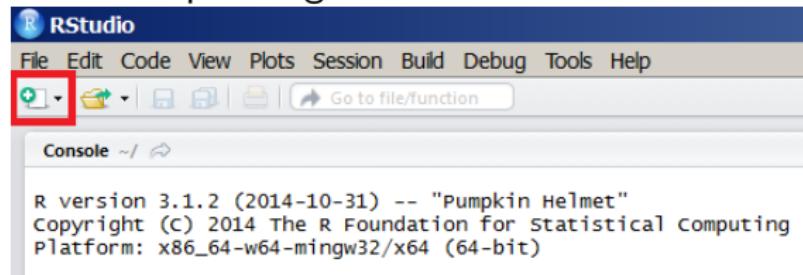
Die verschiedenen Fenster in Rstudio

Hilfe

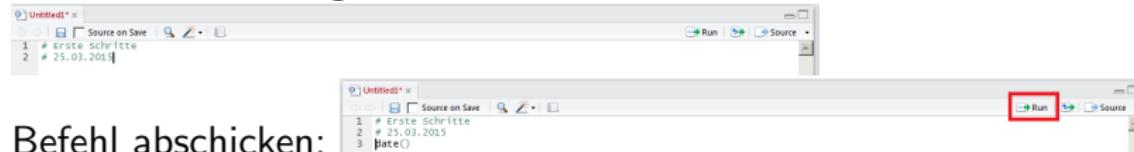
- └ R kam, sah und blieb
- └ Dein Freund das GUI

Erste Schritte - Rstudio

Neues script anlegen:



Kommentare einfügen:



Befehl abschicken:

Aufgabe 1 - Vorbereitung

- ▶ Prüfen Sie, ob eine Version von R auf Rechner installiert ist.
- ▶ Falls dies nicht der Fall ist, laden Sie R unter r-project.org runter und installieren Sie R.
- ▶ Prüfen Sie, ob Rstudio installiert ist.
- ▶ Falls nicht → Installieren: <http://www.rstudio.com/>.
- ▶ Laden Sie die R-Skripte von meinem GitHub-Account
- ▶ Erstellen Sie ein erstes Script und finden Sie das Datum mit dem Befehl `date()` heraus.

Gliederung

R kam, sah und blieb

Grundlagen im Umgang mit der Sprache R

R ist eine Objekt-orientierte Sprache

Verschiedene Datentypen

Indizieren

Wie bekommt man Hilfe?

Modularer Aufbau

Rein und raus – Datenimport und -export

Ein erster Eindruck – Was uns die Daten sagen

R als Taschenrechner

```
#Grundrechenarten
```

```
1+1  
2-1  
2*2  
4/2
```

```
#Mathematische Funktionen
```

```
log(4)  
log(4, base=2)  
exp(4)  
sqrt(4)  
2^4
```

Vektoren und Zuweisungen

- ▶ R ist eine Objekt-orientierte Sprache
- ▶ `<-` ist der Zuweisungsoperator

```
b <- c(1,2) # erzeugt ein Objekt mit den Zahlen 1 und 2
```

- ▶ Eine Funktion kann auf dieses Objekt angewendet werden:
- ▶ `mean(b)` berechnet den Mittelwert

Mit den folgenden Funktionen können wir etwas über die Eigenschaften des Objekts lernen:

- ▶ `length(b)` - b hat die Länge 2
- ▶ `str(b)` - b ist ein numerischer Vektor

Funktionen im base-Paket

Funktion	Bedeutung	Beispiel
<code>length()</code>	Länge	<code>length(b)</code>
<code>max()</code>	Maximum	<code>max(b)</code>
<code>min()</code>	Minimum	<code>min(b)</code>
<code>sd()</code>	Standardabweichung	<code>sd(b)</code>
<code>var()</code>	Varianz	<code>var(b)</code>
<code>mean()</code>	Mittelwert	<code>mean(b)</code>
<code>median()</code>	Median	<code>median(b)</code>

Diese Funktionen brauchen nur ein Argument.

Andere Funktionen brauchen mehr:

<code>quantile()</code>	90 % Quantile	<code>quantile(b,.9)</code>
<code>sample()</code>	Stichprobe ziehen	<code>sample(b,1)</code>

Eine einführende Übersicht findet man unter:

<http://cran.r-project.org/doc/manuals/R-intro.html>

2.1 Vectors and assignment

R operates on named *data structures*. The simplest such structure is the numeric *vector*, which is a single entity consisting of an ordered collection of numbers. To set up a vector named `x`, say, consisting of five numbers, namely 10.4, 5.6, 3.1, 6.4 and 21.7, use the R command

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```

This is an *assignment* statement using the *function* `c()` which in this context can take an arbitrary number of vector *arguments* and whose value is a vector got by concatenating its arguments end to end.⁶

- └ Grundlagen im Umgang mit der Sprache R
- └ R ist eine Objekt-orientierte Sprache

Reference cards

Ausdrucken und neben den Bildschirm hängen!

R Reference Card 2.0

Public domain, v2.0 2012-12-24.
 V 2 by Matt Baggett, matt@baggett.net
 V 1 by Tom Short, t.short@ieee.org
 Material from *R for Beginners* by permission of
 Emmanuel Paradis.

Getting help and info

`help(topic)` documentation on topic
`?topic` same as above; special chars need quotes: for example `?&&`
`help.search("topic")` search the help system; same as `?topic`
`apropos("topic")` the names of all objects in the search list matching the regular expression `"topic"`
`help.start()` start the HTML version of help
`summary(x)` generic function to give a "summary" of x, often a statistical one
`str(x)` display the internal structure of an R object
`is()` show objects in the search path; specify `pat="pat"` to search on a pattern
`is.str()` str for each variable in the search path
`dir()` show files in the current directory
`methods(x)` shows S3 methods of x
`methods(class=class(x))` lists all the methods to handle objects of class x
`findFn()` searches a database of help packages for functions and returns a data.frame (`sos`)

Operators

<code><-</code>	Left assignment, binary
<code>-></code>	Right assignment, binary
<code>=</code>	Left assignment, but not recommended
<code><-<</code>	Left assignment in outer lexical scope; not for beginners
<code>\$</code>	List subset, binary
<code>-</code>	Minus, can be unary or binary
<code>+</code>	Plus, can be unary or binary
<code>-~</code>	Tilde, used for model formulae
<code>:</code>	Sequence, binary (in model formulae: interaction)
<code>::</code>	Refer to function in a package, i.e., <code>pkg::function</code> ; usually not needed
<code>*</code>	Multiplication, binary
<code>/</code>	Division, binary
<code>^</code>	Exponentiation, binary
<code>%*%</code>	Special binary operators, x can be replaced by any valid name
<code>%o%</code>	Modulus, binary
<code>%/%</code>	Integer divide, binary
<code>%*%*</code>	Matrix product, binary
<code>%o*%</code>	Outer product, binary
<code>%ox%</code>	Kronecker product, binary
<code>%in%</code>	Matching operator, binary (in model formulae: nesting)
<code>! x</code>	logical negation, NOT x
<code>x & y</code>	elementwise logical AND
<code>x && y</code>	vector logical AND
<code>x y</code>	elementwise logical OR
<code>x y</code>	vector logical OR

Indexing vectors

<code>x[n]</code>	nth element
<code>x[-n]</code>	all but the nth element
<code>x[1:n]</code>	first n elements
<code>x[-(1:n)]</code>	elements from n+1 to end
<code>x[c(1,4,2)]</code>	specific elements
<code>x["name"]</code>	element named "name"
<code>x[x > 3]</code>	all elements greater than 3
<code>x[x > 3 & x < 5]</code>	all elements between 3 and 5
<code>x[x %in% c("a","if")]</code>	elements in the given set

Indexing lists

<code>x[n]</code>	list with elements n
<code>x[[n]]</code>	nth element of the list
<code>x[["name"]]</code>	element named "name"
<code>x\$name</code>	as above (w. partial matching)

Indexing matrices

<code>x[i,j]</code>	element at row i, column j
<code>x[i,]</code>	row i
<code>x[,j]</code>	column j
<code>x[,c(1,3)]</code>	columns 1 and 3
<code>x[["name",]]</code>	row named "name"

Indexing matrices data frames (same as matrices plus the following)

<code>X[["name"]]</code>	column named "name"
<code>x\$name</code>	as above (w. partial matching)

Input and output (I/O)

Reference cards

- ▶ Die bekannteste Version:

[http://cran.r-project.org/doc/contrib/
Short-refcard.pdf](http://cran.r-project.org/doc/contrib/Short-refcard.pdf)

- ▶ Eine Karte für Data Mining

[http://cran.r-project.org/doc/contrib/
YanchangZhao-refcard-data-mining.pdf](http://cran.r-project.org/doc/contrib/YanchangZhao-refcard-data-mining.pdf)

Aufgabe 2 - Zuweisungen und Funktionen

Erzeugen Sie einen Vektor b mit den Zahlen von 1 bis 5 und berechnen Sie...

1. den Mittelwert
2. die Varianz
3. die Standardabweichung
4. die quadratische Wurzel aus dem Mittelwert

Verschiedene Datentypen

Datentyp	Beschreibung	Beispiel
numeric	ganze und reelle Zahlen	5, 3.462
logical	logische Werte	FALSE, TRUE
character	Buchstaben und Zeichenfolgen	"Hallo"

Quelle: R. Münnich und M. Knobelspieß (2007): Einführung in das statistische Programm Paket R

Verschiedene Datentypen

```
b <- c(1,2) # numeric
log <- c(T,F) # logical
char <-c("A","b") # character
fac <- as.factor(c(1,2)) # factor
```

Mit `str()` bekommt man den Objekttyp.

```
> str(fac)
Factor w/ 2 levels "1","2": 1 2
|
```

Indizieren

Indizieren eines Vektors:

```
> A1 <- c(1,2,3,4)
> A1
[1] 1 2 3 4
> A1[1]
[1] 1
> A1[4]
[1] 4
> A1[1:3]
[1] 1 2 3
> A1[-4]
[1] 1 2 3
```

data.frames

Beispieldaten generieren:

```
AGE <- c(20,35,48,12)
SEX <- c("m","w","w","m")
```

Diese beiden Vektoren zu einem data.frame verbinden:

```
Daten <- data.frame(Alter=AGE, Geschlecht=SEX)
```

Anzahl der Zeilen/Spalten herausfinden

```
nrow(Daten) # Zeilen
ncol(Daten) # Spalten
```

Indizieren

Indizieren eines dataframe:

```
> AA <- 4:1
> A2 <- cbind(A1,AA)
> A2[1,1]
A1
 1
> A2[2,]
A1 AA
 2  3
> A2[,1]
[1] 1 2 3 4
> A2[,1:2]
      A1 AA
[1,]  1  4
[2,]  2  3
[3,]  3  2
[4,]  4  1
```

Matrizen und Arrays

- ▶ In Matrizen und Arrays stehen meist nur numerische Werte.
- ▶ Dadurch wird beispielsweise Matrix Multiplikation möglich.
- ▶ Anders als beim data.frame sind mehr als zwei Dimensionen möglich.

```
A <- matrix(seq(1,100), nrow = 4)
dim(A)
```

Indizieren

Indizieren eines array:

```
> A3 <- array(1:8,c(2,2,2))
> A3
, , 1

[,1] [,2]
[1,]    1    3
[2,]    2    4

, , 2

[,1] [,2]
[1,]    5    7
[2,]    6    8

> A3[, , 2]
[,1] [,2]
[1,]    5    7
[2,]    6    8
```

Indizieren

Indizieren einer Liste:

```
> A4 <- List(A1,1)
> A4
[[1]]
[1] 1 2 3 4

[[2]]
[1] 1

> A4[[2]]
[1] 1
```

Logische Operatoren

Operator	Operation
>	größer als
<	kleiner als
==	genau gleich
!=	ungleich (Negation)
>=	größer gleich
<=	kleiner gleich
&	und
	oder
!	Negation
xor	entweder oder

Quelle: R. Münnich und M. Knobelspieß (2007): Einführung in das statistische Programm Paket R

Indizieren mit logischen Operatoren

Beispieldaten generieren:

```
AGE <- c(20,35,48,12)
SEX <- c("m","w","w","m")
```

Logische Operatoren verwenden:

```
AGE [SEX == "m"]
AGE [SEX != "m"]

SEX [AGE >= 35]
```

Sequenzen

```
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
> rep(1,10)
[1] 1 1 1 1 1 1 1 1 1 1
> rep("A",10)
[1] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
> seq(-2,8,by=1.5)
[1] -2.0 -0.5 1.0 2.5 4.0 5.5 7.0
```

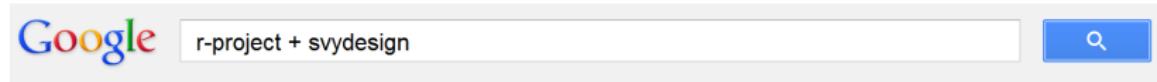
Die Funktion paste

```
?paste  
paste(1:4)  
paste("A", 1:6, sep = "")
```

```
> paste(1:4)  
[1] "1" "2" "3" "4"  
> paste("A", 1:6, sep = "")  
[1] "A1" "A2" "A3" "A4" "A5" "A6"
```

Wie bekommt man Hilfe?

- ▶ Um generell Hilfe zu bekommen: `help.start()`
- ▶ Online Dokumentation für die meisten Funktionen:
`help(name)`
- ▶ Nutze `?` um Hilfe zu bekommen.
Beispiel: `?mean`
- ▶ `example(lm)` gibt ein Beispiel für die lineare Regression

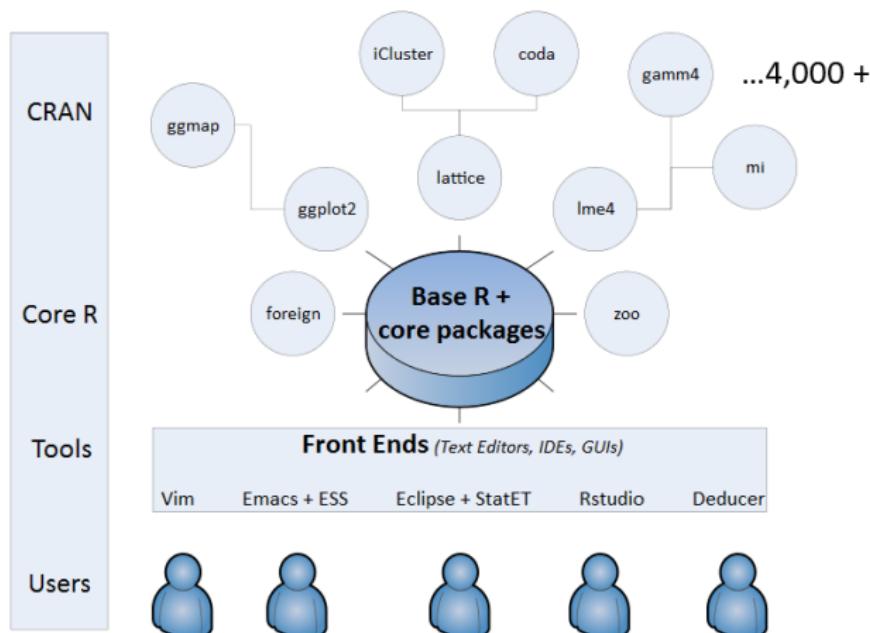


- ▶ Ich nutze meistens google
- ▶ Tippe:
R-project + Was ich schon immer wissen wollte
- ▶ Das funktioniert natürlich mit jeder Suchmaschine!

The screenshot shows the homepage of Stack Overflow. At the top, there's a navigation bar with links for 'sign up', 'log in', 'tour', 'help', 'careers 2.0', and a search bar. Below the navigation is the Stack Overflow logo and a navigation menu with 'Questions', 'Tags', 'Tour', and 'Users' buttons. A large 'Ask Question' button is on the right. The main content area has a light gray background. On the left, a text box says: 'Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.' Below it is a 'Take the 2-minute tour' button. In the center, under the heading 'Here's how it works:', there are three sections: 'Anybody can ask a question' (illustrated with a 'Q' icon), 'Anybody can answer' (illustrated with 'A' icons), and 'The best answers are voted up and rise to the top' (illustrated with 'A' icons and arrows). There are also small icons for a profile picture and a 'Tour' link.

- ▶ <http://stackoverflow.com/>
- ▶ Für Fragen zum Programmieren
- ▶ Ist nicht auf R fokussiert
- ▶ Sehr detaillierte Diskussionen

Modularer Aufbau



Modularer Aufbau

- ▶ Viele Funktionen sind im Basis-R enthalten
- ▶ Viele spezifische Funktionen sind in zusätzlichen Bibliotheken integriert
- ▶ R kann modular erweitert werden durch sog. **packages** bzw. **libraries**
- ▶ Auf **CRAN** werden die wichtigsten packages gehostet (im Moment 4567)
- ▶ Weitergehende Pakete finden sich z.B. bei www.bioconductor.org

```
install.packages("lme4")
```

```
library(lme4)
```

Installation von Paketen

The screenshot shows the RStudio interface with several windows open:

- Code Editor (top-left):** A script named "paths.R" containing the code:

```
1 setwd("D:/Projekte/R/packages/germanwebr/Rfunctions")  
2
```
- Environment (top-right):** Shows the message "Environment is empty".
- Console (bottom-left):** Displays the message: "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."
- Packages (bottom-right):** A table showing installed packages and their versions:

Package	Description	Version
AER	Applied Econometrics with R	12.2
arules	Mining Association Rules and Frequent Itemsets	1.1-2
bitops	Bitwise Operations	10-6
boot	Bootstrap Functions (originally by Angelo Canty for S)	13-11
brew	Templating Framework for Report Generation	1.0-6
car	Companion to Applied Regression	2.0-19
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17
class	Functions for Classification	7.3-10
cluster	Cluster Analysis Extended Rousseeuw et al.	1.15.2
codetools	Code Analysis Tools for R	0.2-8
colorspace	Color Space Manipulation	1.2-4
compiler	The R Compiler Package	3.1.0
DAAG	Data Analysis And Graphics data and functions	1.18

Installation von Paketen

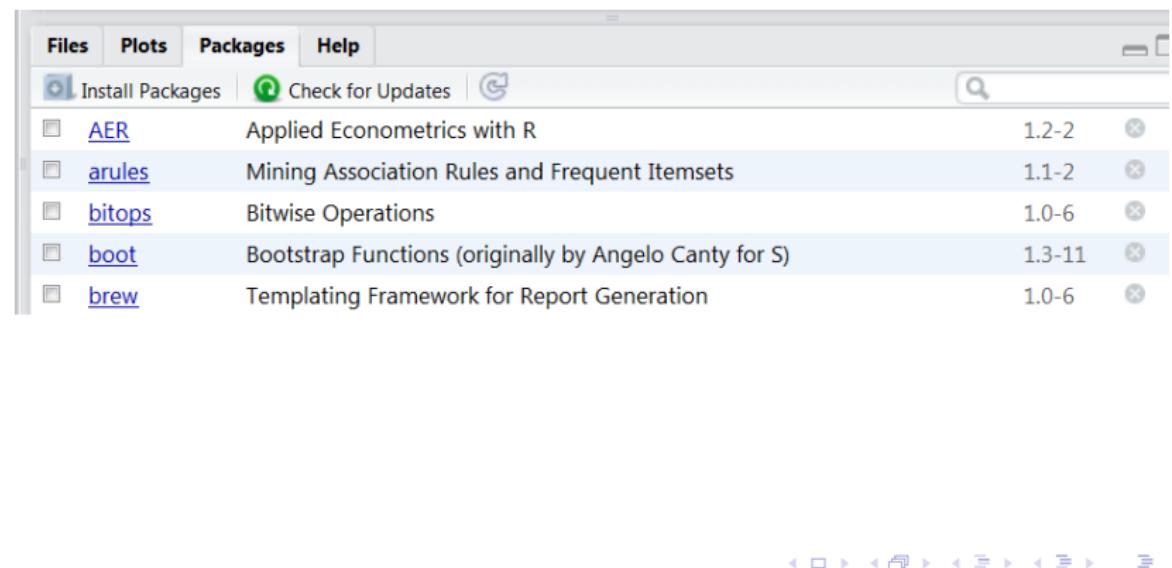
The screenshot shows the RStudio interface with several windows open:

- Code Editor (top-left):** A script named "paths.R" containing the code:

```
1 setwd("D:/Projekte/R/packages/germanwebr/Rfunctions")
2
```
- Environment (top-right):** Shows the message "Environment is empty".
- Console (bottom-left):** Displays the R startup message and a note about citation practices.
- Packages (bottom-right):** A table showing installed packages and their versions. The table includes columns for package name, description, version, and status (indicated by a circular icon).

Package	Description	Version	Status
AER	Applied Econometrics with R	12.2	○
arules	Mining Association Rules and Frequent Itemsets	1.1-2	○
bitops	Bitwise Operations	10-6	○
boot	Bootstrap Functions (originally by Angelo Canty for S)	1.3-11	○
brew	Templating Framework for Report Generation	1.0-6	○
car	Companion to Applied Regression	2.0-19	○
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17	○
class	Functions for Classification	7.3-10	○
cluster	Cluster Analysis Extended Rousseeuw et al.	1.15.2	○
codetools	Code Analysis Tools for R	0.2-8	○
colorspace	Color Space Manipulation	1.2-4	○
compiler	The R Compiler Package	3.1.0	○
DAAG	Data Analysis And Graphics data and functions	1.18	○

Installation von Paketen

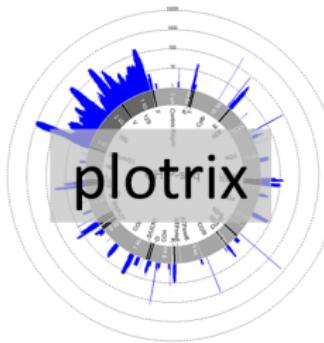
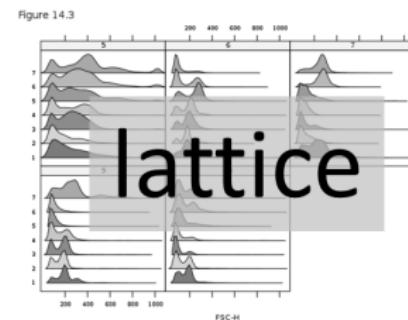
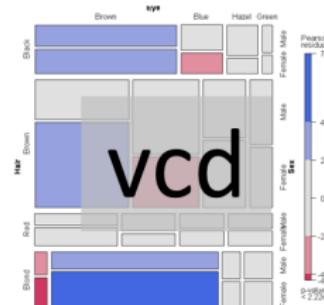
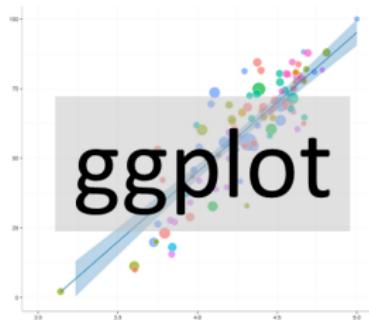


Aufgabe 3 - Interessante Zusatzpakete

Gehen Sie auf cran.r-project.org und suchen Sie in dem Bereich, wo die Pakete vorgestellt werden, nach Paketen,...

1. die für die deskriptive Datenanalyse geeignet sind.
2. um Regressionen zu berechnen
3. um fremde Datensätze einzulesen (z.B. SPSS-Daten)
4. um mit großen Datenmengen umzugehen

Pakete - deskriptive Datenanalyse



Pakete - Regression

Paket	Für was?
base{lm}	Einfache lineare Regression
base{glm}	Generalisierte Lineare Modelle
tsDyn	Autoregressive Modelle (Zeitreihen)
robustbase	Robuste Regressionen
crs	Nichtparametrische Regression
glmnet	Lasso Verfahren

Paket - fremde Datensätze (foreign)

read.spss

read.ssd
write.dta
read.mtp
read.epiinfo
write.arff
read.octave
write.dbf
read.arff
lookup.xport
read.xport

read.dta

write.foreign

Large memory and out-of-memory data

Large memory and out-of-memory data

- The [biglm](#) package by Lumley uses incremental computations to offer `lm()` and `glm()` functionality to data sets stored outside of R's main memory.
- The [ff](#) package by Adler et al. offers file-based access to data sets that are too large to be loaded into memory, along with a number of higher-level functions.
- The [bigmemory](#) package by Kane and Emerson permits storing large objects such as matrices in memory (as well as via files) and uses external pointer objects to refer to them. This permits transparent access from R without bumping against R's internal memory limits. Several R processes on the same computer can also share big memory objects.
- A large number of database packages, and database-alike packages (such as [sqldf](#) by Grothendieck and [data.table](#) by Dowle) are also of potential interest but not reviewed here.
- The [HadoopStreaming](#) package provides a framework for writing map/reduce scripts for use in Hadoop Streaming; it also facilitates operating on data in a streaming fashion which does not require Hadoop.
- The [speedglm](#) package permits to fit (generalised) linear models to large data. For in-memory data sets, `speedlm()` or `speedglm()` can be used along with `update.speedlm()` which can update fitted models with new data. For out-of-memory data sets, `shglm()` is available; it works in the presence of factors and can check for singular matrices.
- The [biglars](#) package by Seligman et al can use the [ff](#) to support large-than-memory datasets for least-angle regression, lasso and stepwise regression.
- The [bigrf](#) package provides a Random Forests implementation with support for parallel execution and large memory.
- The [MonetDB_R](#) package allows R to access the MonetDB column-oriented, open source database system as a backend.
- The [ffbase](#) package by de Jonge et al adds basic statistical functionality to the [ff](#) package.

[http://cran.r-project.org/web/views/
HighPerformanceComputing.html](http://cran.r-project.org/web/views/HighPerformanceComputing.html)

Wichtige Bibliotheken

Bibliothek	Thema
foreign	Functions for reading and writing data stored by statistical packages
sampling	Functions for drawing and calibrating samples.
survey	Analysis of complex survey samples
MASS	Functions and Datasets for Venables and Ripley's Modern Applied Statistics with S'

Weitere nützliche Bibliotheken

Bibliothek	Thema
xtable	Coerce data to LaTeX and HTML tables
dummies	Expands factors, characters and other eligible classes into dummy/indicator variables.
mvtnorm	Multivariate Normal and t Distributions
maptools	Tools for reading and handling spatial objects

Gliederung

R kam, sah und blieb

Grundlagen im Umgang mit der Sprache R

Rein und raus – Datenimport und -export

Datemimport

Datenexport

Ein erster Eindruck – Was uns die Daten sagen

Datenimport



Datenzugang

Public-Use-File (PUF)

Datei zur öffentlichen Nutzung -
meist stark anonymisierte Daten
viele Beispiele unter:

www.forschungsdatenzentrum.de

www.statistik-portal.de

[www.infothek.statistik.rlp.de/lis/MeineRegion/
index.asp](http://www.infothek.statistik.rlp.de/lis/MeineRegion/index.asp)

Scientific-Use-File (SUF)

Datei zur wissenschaftlichen Nutzung - anonymisierte
Daten, die zu wissenschaftlichen Zwecken und zur
Sekundäranalyse genutzt werden können.

On-Site-Nutzung

- ▶ Arbeitsplätze für Gastwissenschaftler
- ▶ Kontrollierte Datenfernverarbeitung



Datenquellen:

- ▶ Datahub
<http://datahub.io/>
- ▶ GDELT: Global Data on Events, Location and Tone
<http://gdelt.utdallas.edu/>
- ▶ Social security administration puf
<http://www.ssa.gov/policy/docs/data/index.html>
- ▶ National health and nutrition examination survey
library(survey) und data(nhanes)
- ▶ FAO Datenbank
<http://cran.r-project.org/web/packages/FAOSTAT/index.html>

Download Daten - Forschungsdatenzentrum

The screenshot shows the homepage of the Statistische Ämter des Bundes und der Länder Forschungsdatenzentren. The header features the logo of the Federal Statistical Office of Germany (Destatis) and the text "STATISTISCHE ÄMTER DES BUNDES UND DER LÄNDER FORSCHUNGSDATENZENTREN". Below the header is a navigation bar with links to "Startseite", "Impressum", "Wir über uns", and "English".

[Datenangebot](#)

Datenangebot | Mikrozensus 2002

[Datenzugang](#)

[Metadaten | Ansprechpartner](#)

[Nutzungsbedingungen](#)

[Amtliche Firmendaten](#)

CAMPUS-File

Das CAMPUS-File zum Mikrozensus 2002 ist eine 3,5%-Wohnungssstichprobe des Originalmaterials des Mikrozensus 2002, der speziell für Lehr- und Übungszwecke erstellt wurde. Die Daten wurden durch Vergrößerung und Löschung einzelner Merkmale anonymisiert. Weitere Informationen zur Methodik und zur Anonymisierung sind in der Methodenbeschreibung aufbereitet. Das CAMPUS-File wird in den Formaten SPSS, SAS und STATA sowie als ASCII-CSV angeboten.

[CAMPUS-Files](#)

[Veranstaltungen](#)

[Veröffentlichungen](#)

[Kontakt](#)

Die Nutzung dieses CAMPUS-Files ist unentgeltlich.

Fälle	Variablen	Hochrechnung
25.137 Personen 11.655 Haushalte 11.788 Wohnungen	335	Bund, Länder

Metadaten zum Download

Datei	Format	Größe
Daten: SAS (mit Labels)	zip	3.050 KB



Dateiformate in R

- ▶ Von R werden quelloffene, nicht-proprietary Formate bevorzugt
- ▶ Es können aber auch Formate von anderen Statistik Software Paketen eingelesen werden
- ▶ R-user speichern Objekte gerne in sog. Workspaces ab
- ▶ Auch hier jedoch gilt: (fast) alles andere ist möglich

Formate - base package

- ▶ R unterstützt von Haus aus schon einige wichtige Formate:
 - ▶ CSV (Comma Separated Values): `read.csv()`
 - ▶ FWF (Fixed With Format): `read.fwf()`
 - ▶ Tab-getrennte Werte: `read.delim()`

Der Arbeitsspeicher

So findet man heraus, in welchem Verzeichnis man sich gerade befindet

```
getwd()
```

So kann man das Arbeitsverzeichnis ändern:

Man erzeugt ein Objekt in dem man den Pfad abspeichert

```
main.path <- "C:/"
```

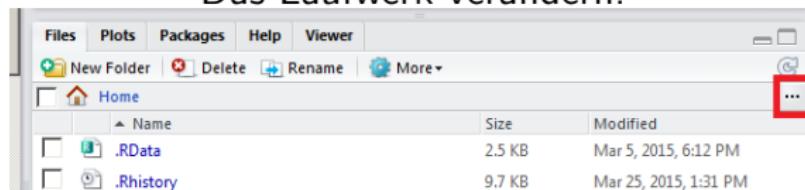
Und ändert dann den Pfad mit setwd():

```
setwd(main.path)
```

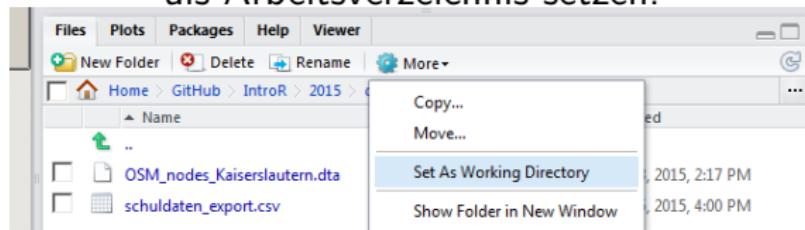
Wichtig ist es Slashes anstelle von Backslashes zu verwenden.

Arbeitsverzeichnis setzen

Das Laufwerk verändern:



Das aktuell angezeigte Verzeichnis
als Arbeitsverzeichnis setzen:



Der Arbeitsspeicher

So findet man heraus, in welchem Verzeichnis man sich gerade befindet

```
getwd()
```

So kann man das Arbeitsverzeichnis ändern:

Man erzeugt ein Objekt in dem man den Pfad abspeichert

```
main.path <- "C:/"
```

Und ändert dann den Pfad mit setwd():

```
setwd(main.path)
```

Wichtig ist es Slashes anstelle von Backslashes zu verwenden.

Import von Excel-Daten

- ▶ `library(foreign)` ist für den Import von fremden Datenformaten nötig
- ▶ Wenn Excel-Daten vorliegen - als .csv abspeichern
- ▶ Dann kann `read.csv()` genutzt werden um die Daten einzulesen.
- ▶ Bei Deutschen Daten kann es sein, dass man `read.csv2()` wegen der Komma-Separierung braucht.

CSV Dateien einlesen

Zunächst muss das Arbeitsverzeichnis gesetzt werden, in dem sich die Daten befinden:

```
Dat <- read.csv("schuldaten_export.csv")
```

Wenn es sich um Deutsche Daten handelt:

```
Dat <- read.csv2("schuldaten_export.csv")
```

SPSS Dateien einlesen

Dateien können auch direkt aus dem Internet geladen werden:

```
link<- "http://www.statistik.at/web_de/static/  
mz_2013_sds_-_datensatz_080469.sav"  
  
?read.spss  
Dat <- read.spss(link,to.data.frame=T)
```

stata Dateien einlesen

```
MZ02 <- read.dta("MZ02.dta")
```

[http://is-r.tumblr.com/post/37181850668/
reading-writing-stata-dta-files-with-foreign](http://is-r.tumblr.com/post/37181850668/reading-writing-stata-dta-files-with-foreign)

Datenmanagement ähnlich wie in SPSS oder Stata

```
install.packages("Rz")
library(Rz)
```

Screenshot of the Rz software interface showing a dataset named "dataset1 (spss_cf_sohi98.sav)".

The interface includes a menu bar (File, Preferences, Help) and a toolbar with icons for file operations.

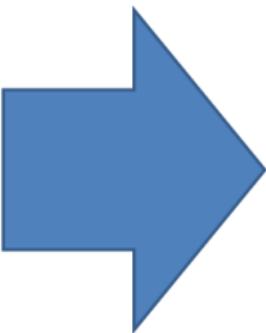
The main window displays a table of variables:

Meas	Names	Labels	Value Labels	Missing
1	gebiet	Gebietsstand	1 "Früh, Bundes"	
2	traeger	Art des Trägers	1 "örtlich", 2 "üb."	
3	lfd_p_nr	Nummer der Person		
4	stellung	Stellung zum Haushaltsvorstand	1 "Haushaltsvors"	
5	geschl	Geschlecht	1 "männlich", 2 "	
6	geb_jahr	Geburtsjahr		
7	staat	Personengruppe	1 "Deutsche(r)",	
8	zuschl1	Mehrbedarfszuschlag, 1. Art	1 "an Personen",	
9	zuschl2	Mehrbedarfszuschlag, 2. Art	1 "an Personen",	
10	zuschl3	Mehrbedarfszuschlag, 3. Art	1 "an Personen",	
11	zuschl4	Mehrbedarfszuschlag, 4. Art	1 "an Personen",	
12	erv_sta	Erwerbsstatus	1 "vollzeiterwerb",	
13	schule	höchster allgemeinbildender Schulabschluss	1 "in schulischer",	
14	beruf	höchster Berufsausbildungsabschluss	1 "kein Ausbildung",	
15	dauer_e	bisherige Dauer der Arbeitslosigkeit		
16	lfd_hhnr	Lfd. Nr. des Haushalts		
17	personen	Anzahl der Personen in der Bedarfsgem.		
18	einricht	Lfd. HLU wird gewährt	1 "ausserhalb vo",	
19	bedarf	Bruttobedarf der Bedarfsgem. in DM/Monat		
20	miete	anerkannte Bruttokaltmiete in DM/Monat		
21	anspruc	Anspruch der Bedarfsgem. Jp. DM/Monat (W)		
22	soz_sit1	besondere soziale Situation, 1 Möglichkeit, 1 Tod eines Par		

Aufgabe 4 - Datenimport

- ▶ Gehen Sie auf die Seite des Forschungsdatenzentrums und laden Sie den Campusfile des Mikrozensus 2002 herunter
- ▶ Laden Sie den Datensatz mit einer geeigneten Funktion in Ihren Workspace.
- ▶ Finden Sie heraus, wieviele Beobachtungen und Variablen der Datensatz umfasst.

Datenexport



R's Exportformate

- ▶ In R werden offene Dateiformate bevorzugt
- ▶ Als Äquivalenz zu den `read.X()` Funktionen stehen viele `write.X()` Funktionen zur Verfügung
- ▶ Das eigene Format von R sind sog. Workspaces (`.RData`)

Export

The screenshot shows the homepage of Quick-R. At the top is a blue header bar with a logo of a stylized 'R' on the left. To its right is a horizontal menu with links: Home, Interface, Input, Manage, Stats, Adv Stats, Graphs, Adv Graphs, and Blog. Below the menu is a search bar with a placeholder and a 'Search' button. The main content area has a light blue background with diagonal stripes. On the left, there's a sidebar with a dark blue header labeled 'Data Input' containing a list of links: Data types, Importing Data, Keyboard Input, Database Input, Exporting Data, Viewing Data, Variable Labels, Value Labels, Missing Data, and Date Values.

Exporting Data

There are numerous methods for exporting R objects into other formats . For SPSS, SAS and Stata. you will need to load the [foreign](#) packages. For Excel, you will need the [xlsReadWrite](#) package.

To A Tab Delimited Text File

```
write.table(mydata, "c:/mydata.txt", sep="\t")
```

To an Excel Spreadsheet

```
library(xlsx)
write.xlsx(mydata, "c:/mydata.xlsx")
```



Überblick Daten Import/Export

```
save(Dat, file="Dat.RData")
```

R Data Import/Export

Version 3.1.0 (2014-04-10)

<http://cran.r-project.org/doc/manuals/r-release/R-data.pdf>

Gliederung

R kam, sah und blieb

Grundlagen im Umgang mit der Sprache R

Rein und raus – Datenimport und -export

Ein erster Eindruck – Was uns die Daten sagen

Streuungsmaße

Im base Package sind die wichtigsten Streuungsmaße enthalten:

- ▶ Varianz: `var()`
- ▶ Standardabweichung: `sd()`
- ▶ Minimum und Maximum: `min()` und `max()`
- ▶ Range: `range()`

Fehlende Werte

- Sind NAs vorhanden muss dies der Funktion mitgeteilt werden

```
?var  
var(x)  
var(xNA)  
var(xNA, na.rm=TRUE)
```

Häufigkeitstabellen

- ▶ Eine Auszählung der Häufigkeiten der Merkmale einer Variable liefert `table()`
- ▶ Mit `table()` sind auch Kreuztabellierungen möglich indem zwei Variablen durch Komma getrennt werden: `table(x,y)` liefert Häufigkeiten von `y` für gegebene Ausprägungen von `x`

Die Funktion `table()`

```
?table  
table(x)  
table(x, musician)  
data(esoph)  
table(esoph$agegp)
```

Häufigkeitstabellen

- ▶ `prop.table()` liefert die relativen Häufigkeiten
- ▶ Wird die Funktion außerhalb einer `table()` Funktion geschrieben erhält man die relativen Häufigkeiten bezogen auf alle Zellen

Die Funktion `prop.table()`

```
table(esoph$agegp,esoph$alcgp)
?prop.table
prop.table(table(esoph$agegp,
esoph$alcgp),1)
```

Die aggregate() Funktion

- ▶ Mit der aggregate() Funktion können Kennwerte für Untergruppen erstellt werden
- ▶ `aggregate(x, by, FUN)` müssen mindestens drei Argumente übergeben werden:
 - x:** ein oder mehrere Beobachtungsvektor(en) für den der Kennwert berechnet werden soll
 - by:** eine oder mehrere bedingende Variable(n)
 - FUN:** die Funktion welche den Kennwert berechnet (z.B. `mean` oder `sd`)
- ▶ Die Ausgabe kann mit Hilfe von `xtabs()` in eine schöne zweidimensionale Tabelle überführt werden

Die Funktion apply

apply {base}

R Documentation

Apply Functions Over Array Margins

Description

Returns a vector or array or list of values obtained by applying a function to margins of an array or matrix.

Usage

```
apply(X, MARGIN, FUN, ...)
```

Arguments

X an array, including a matrix.

MARGIN a vector giving the subscripts which the function will be applied over. E.g., for a matrix 1 indicates rows, 2 indicates columns, `c(1, 2)` indicates rows and columns. Where **x** has named dimnames, it can be a character vector selecting dimension names.

FUN the function to be applied: see 'Details'. In the case of functions like `+`, `%*%`, etc., the function name must be backquoted or quoted.

... optional arguments to **FUN**.

Die Funktion apply

Für `margin=1` die Funktion `mean` auf die Reihen angewendet,
Für `margin=2` die Funktion `mean` auf die Spalten angewendet,

```
> ApplyDat <- cbind(1:4,runif(4),rnorm(4))
> apply(ApplyDat,1,mean)
[1] 0.4798562 0.9655396 1.0619841 1.8760724
> apply(ApplyDat,2,mean)
[1] 2.5000000 0.4251074 0.3624818
```

Anstatt `mean` können auch andere Funktionen wie `var`, `sd` oder `length` verwendet werden.

Die Funktion tapply

tapply {base}

R Documentation

Apply a Function Over a Ragged Array

Description

Apply a function to each cell of a ragged array, that is to each (non-empty) group of values given by a unique combination of the levels of certain factors.

Usage

```
tapply(X, INDEX, FUN = NULL, ..., simplify = TRUE)
```

Arguments

- X** an atomic object, typically a vector.
- INDEX** list of one or more factors, each of same length as **x**. The elements are coerced to factors by [as.factor](#).
- FUN** the function to be applied, or **NULL**. In the case of functions like **+**, **%*%**, etc., the function name must be backquoted or quoted. If **FUN** is **NULL**, **tapply** returns a vector which can be used to subscript the multi-way array **tapply** normally produces.
- ...** optional arguments to **FUN**: the Note section.

Die Funktion tapply

```
> ApplyDat <- data.frame(Income=rnorm(5,1400,200),  
+                           Sex=sample(c(1,2),5,replace=T))  
> ApplyDat  
   Income Sex  
1 1230.7846  1  
2  871.6304  1  
3 1454.7069  2  
4 1495.1007  2  
5 1348.5055  1  
> tapply(ApplyDat$Income,ApplyDat$Sex,mean)  
      1       2  
1150.307 1474.904
```

Die Funktion tapply

Auch andere Funktionen können eingesetzt werden....

Auch selbst programmierte Funktionen

```
> ApplyDat
  Income Sex
1 1230.7846  1
2  871.6304  1
3 1454.7069  2
4 1495.1007  2
5 1348.5055  1
> tapply(ApplyDat$Income,ApplyDat$Sex,function(x)x)
$`1`
[1] 1230.7846 871.6304 1348.5055

$`2`
[1] 1454.707 1495.101
```

Im Beispiel wird die einfachste eigene Funktion angewendet.

Aufgabe 5 - Apply Funktion verwenden

- Erstellen Sie eine Matrix A mit 4 Zeilen und 25 Spalten, die die Werte 1 bis 100 enthält. Analog dazu erstellen Sie eine Matrix B mit 25 Zeilen und 4 Spalten, die die Werte 1 bis 100 enthält.
- Berechnen Sie mittels dem `apply()`-Befehl den Mittelwert und die Varianz für jede Zeile von A bzw. B.
- Berechnen Sie mittels dem `apply()`-Befehl den Mittelwert und die Varianz für jede Spalte von A bzw. B.
- Standardisieren ist eine häufige Transformation von Daten; dafür wird der Mittelwert von der entsprechenden Zeile oder Spalte abgezogen und durch die entsprechende Standardabweichung geteilt. Somit besitzen die Daten einen Mittelwert von 0 und eine Standardabweichung von 1.

Standardisieren Sie die Spalten der Matrix A. Abschließend überprüfen Sie, ob die Spalten richtig standardisiert wurden.

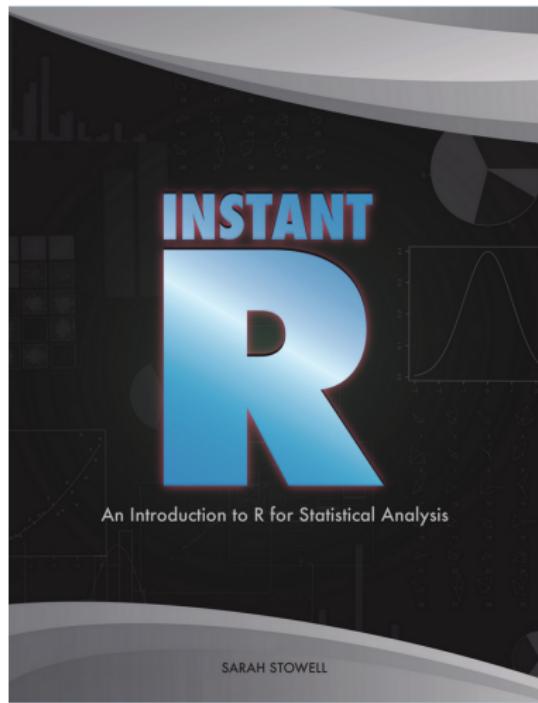
Quelle: <http://www.uni-leipzig.de/~zuber/teaching/ws12/r-kurs/praxis/U2.pdf>

Literatur



- ▶ Ligges, U. (2008):
Programmieren mit R.
Springer.
- ▶ Gut für Anfänger

Literatur



- ▶ Import und Export von Daten
- ▶ Daten editieren
- ▶ Graphiken

Mehr Hilfe für Anfänger



Help the Stat Consulting Group by

[Google™ Custom Search](#)

[giving a gift](#)



stat > r > sk

R Starter Kit

This page is intended for people who:

Are just starting	Have a question or two about	Want a quick refresher
<ul style="list-style-type: none">• to learn R• to utilize basic statistical procedures	<ul style="list-style-type: none">• how to do a simple task in R• how to interpret the output from commonly used procedures	<ul style="list-style-type: none">• on how to do basic tasks in R• on frequently used statistical procedures and the interpretation of their output.

These materials have been collected from various places on our website and have been ordered so that you can, in step-by-step fashion, develop the skills needed to conduct common analyses in R.

Getting familiar with R

- [Class notes](#): There is no point in waiting to take an introductory class on how to use R. Instead, we have notes of our introductory class that you can download and view.
- [Learning modules](#): We have developed a set of web pages called learning modules which show you how to accomplish basic data management tasks in R, including how to get data into R, how to recode variable and how to subset data. The R code and the output produced are shown, as well as tips on things to look out for.

<http://www.ats.ucla.edu/stat/r/sk/>



Hilfe und erste Schritte:

- ▶ Skript für die ersten Schritte:
www.stamats.de/InstallationUndErsteSchritteMitR.pdf

- ▶ Einführung direkt auf CRAN:
cran.r-project.org/doc/contrib/Sawitzki-Einfuehrung.pdf