

# GESIS-Workshop

## "Datenanalyse mit R"

### Erste Schritte

Jan-Philipp Kolb

Dienstag, 20. Mai, 2014



## Gliederung

R kam, sah und blieb

Was ist eigentlich R?

Dein Freund das GUI

Modularer Aufbau

Wie bekommt man Hilfe?

Grundlagen im Umgang mit der Sprache R

R ist eine Objekt-orientierte Sprache

Indizieren

Rein und raus – Datenimport und -export

Datemimport

Datenexport

Ein erster Eindruck – Was uns die Daten sagen

Häufigkeiten und gruppierte Kennwerte

Die apply-Familie

Umkodieren, Plausibilisieren

## Gliederung

R kam, sah und blieb

Was ist eigentlich R?

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

Umkodieren, Plausibilisieren

## Erwartungen und Anforderungen

Das kann diese Schulung vermitteln:

- ▶ Eine praxisnahe Einführung in die statistische Programmiersprache **R**
- ▶ Erlernen einer Programmier-Strategie

## Erwartungen und Anforderungen

Das kann diese Schulung vermitteln:

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

## 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

## 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

## Erwartungen und Anforderungen

Das kann sie nicht leisten:

- ▶ Eine Einführungsveranstaltung in die Statistik geben
- ▶ Grundlegende datenanalytische Konzepte vermitteln

## Erwartungen und Anforderungen

Das kann sie nicht leisten:

- ▶ Eine Einführungsveranstaltung in die Statistik geben
- ▶ Grundlegende datenanalytische Konzepte vermitteln
- ▶ Verständnis zementieren

## Erwartungen und Anforderungen

Das kann sie nicht leisten:

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

## 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
- └ Was ist eigentlich R?

## Wo kommen wir her und wo gehen wir hin?

„R is a **language** and environment for **statistical computing** and **graphics**. It is a **GNU** project which is similar to the **S language** and environment (...),“

„R provides a wide variety of **statistical** (...) and **graphical techniques**, and is **highly extensible**. (...) R provides an **Open Source** route to participation in that activity.“

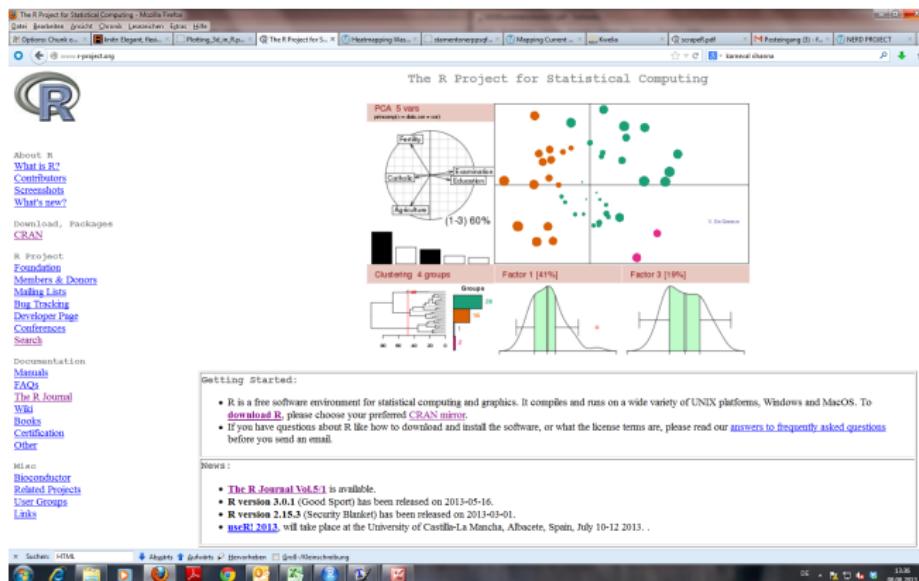
„One of R's strengths is the ease with which well-designed **publication-quality plots** can be produced (...) the user retains full control.“

(<http://www.r-project.org/about.html>)

└ R kam, sah und blieb

└ Was ist eigentlich R?

# R herunterladen:



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

- └ R kam, sah und blieb
- └ Was ist eigentlich R?

## Das ist das Basis-R:

The screenshot shows the RGui (64-bit) application window. The top menu bar includes 'Datei', 'Bearbeiten', 'Pakete', 'Windows', and 'Hilfe'. The left pane, titled 'R Gui (64-bit)', contains the 'R Editor' tab with the script content: '# This is the script'. The right pane, titled 'R Console', displays the R startup message and help text. A red cursor is visible at the bottom of the console window.

```
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

Dein Freund das GUI

Modularer Aufbau

Wie bekommt man Hilfe?

Grundlagen im Umgang mit der Sprache R

Rein und raus – Datenimport und -export

Ein erster Eindruck – Was uns die Daten sagen

Umkodieren, Plausibilisieren

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

## Graphical User Interfaces (GUI)

- ▶ 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/>
- ▶ ...

# Ich nutze Rstudio!

The screenshot shows the RStudio interface with several windows open:

- File Edit Code View Plots Session Project Build Tools Help**
- Source on Site** tab is selected.
- Run** button is visible.
- Source** tab is also present.
- Console** tab shows R code and its output.
- Workspace** tab shows objects in the environment.
- History** tab shows the history of operations.
- Project** tab shows the current project structure.
- File Plot Packages Help** menu bar.
- R Kolmogorov-Smirnov Tests** tab is selected in the bottom-left corner.
- R Documentation** tab is also present.

The code in the source editor is as follows:

```
# Unitled1.R
# DayPart_OLS_MLR.R
# Source on Site
# Run
# Source

8 # -----
9 #
10 # -----
11 # Necessary Libraries
12 # -----
13 # Necessary Libraries
14 # -----
15 #
16 # to read foreign datasets like the stata format .dta
17 #
18 library(foreign)
19 #
20 # -----
21 # Setting the directories
22 # -----
23 #
24 # Name of the working directory is saved in an object
25 # that ensures, that I can refer to it later:
26 #
27 #data.path <- "D:/Dropbox/Samp & Analysis 2013/exercises/datasets"
28 #data.path <- "C:/Users/Stephanie/Documents/My Dropbox/courses/GESIS Summer School/Samp & An
29 #
30 # -----
31 #
```

Console output:

```
> # to read foreign datasets like the stata format .dta
> library(foreign)
> #
> # Setting the directories
> #
> # Name of the working directory is saved in an object
> # that ensures, that I can refer to it later:
> #data.path <- "D:/Dropbox/Samp & Analysis 2013/exercises/datasets"
> #data.path <- "C:/Users/Stephanie/Documents/My Dropbox/courses/GESIS Summer School/Samp &
> #Analysis 2013/exercises/datasets"
```

<http://www.rstudio.com/>

# R-Script - Erste Schritte

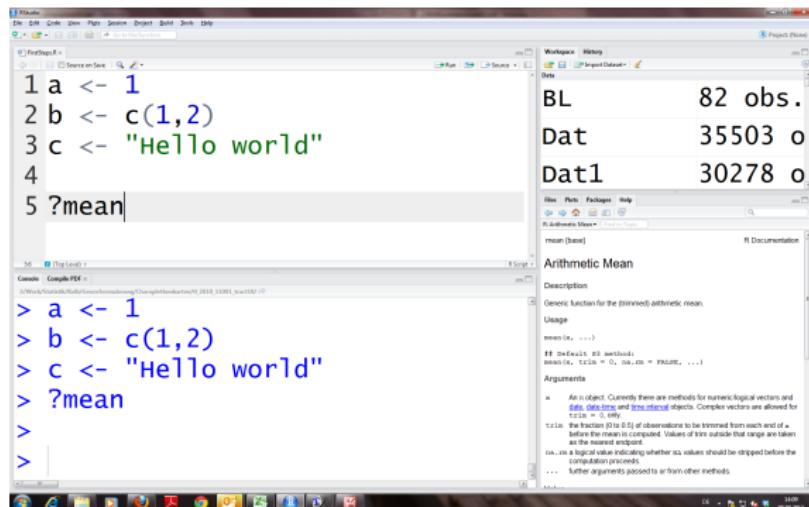
[https://github.com/Japhilko/DLR\\_IntroR](https://github.com/Japhilko/DLR_IntroR)

The screenshot shows a GitHub repository page for 'Japhilko / DLR\_IntroR'. The repository has 1 star, 0 forks, and 0 issues. The file 'DLR\_IntroR\_A\_ErsteSchritte.R' is displayed, containing the following R code:

```
1 #####  
2 # DLR Intro Berlin  
3 # Jan-Philipp Kolb  
4 # 05.11.2014  
5 #  
6 # Erste Schritte  
7 #  
8 #####  
9  
10  
11 #Grundrechenarten  
12  
13 1+1  
14 2-1  
15 2*2  
16 4/2  
17
```

## 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](http://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



## Die verschiedenen Fenster in Rstudio

The screenshot shows the RStudio interface. On the left, the script editor 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 displays the documentation for the `mean` function, specifically the "Arithmetic Mean" section. The help text includes a detailed description, usage examples, and arguments. A note at the bottom states: "This function is part of the base package which is standard on all systems."

## Die verschiedenen Fenster in Rstudio Das Script



## Die verschiedenen Fenster in Rstudio

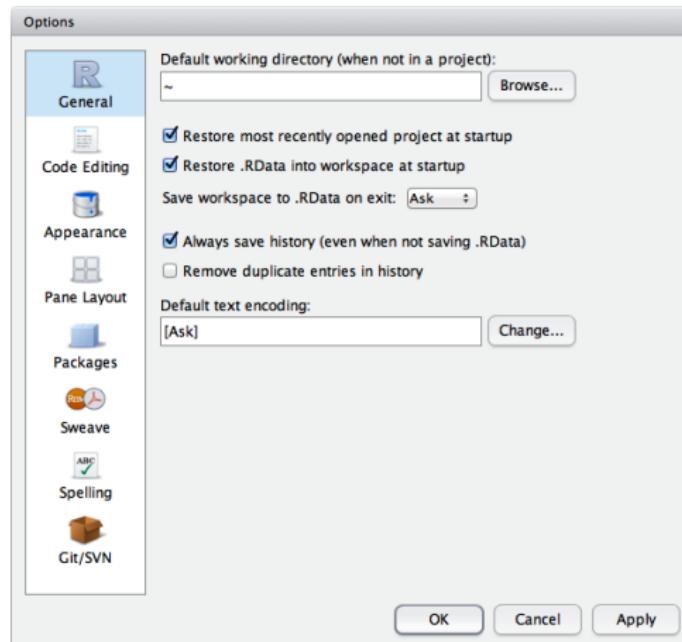
### Konsole



## Die verschiedenen Fenster in Rstudio

### Hilfe

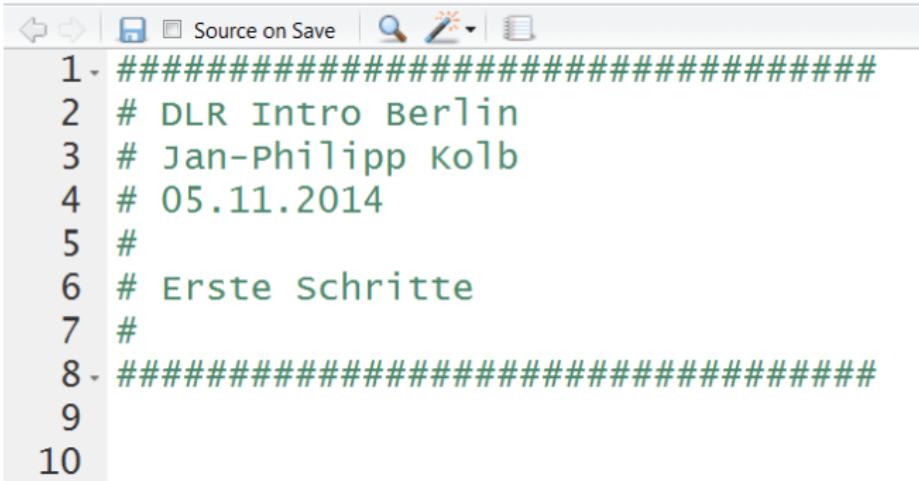
# RStudio anpassen



<http://www.rstudio.com/ide/docs/using/customizing>

## Erste Schritte

Um später noch den Überblick zu behalten, ist es sinnvoll mit vielen Kommentaren zu arbeiten:

A screenshot of a code editor window showing an R script. The window has a toolbar at the top with icons for back, forward, save, search, and edit. The code itself is numbered from 1 to 10 on the left. Lines 1, 8, and 10 consist of a series of hash symbols (#). Lines 2 through 7 are comments describing the script's purpose and author.

```
1 - #####  
2 # DLR Intro Berlin  
3 # Jan-Philipp Kolb  
4 # 05.11.2014  
5 #  
6 # Erste Schritte  
7 #  
8 - #####  
9  
10
```

# R als Taschenrechner

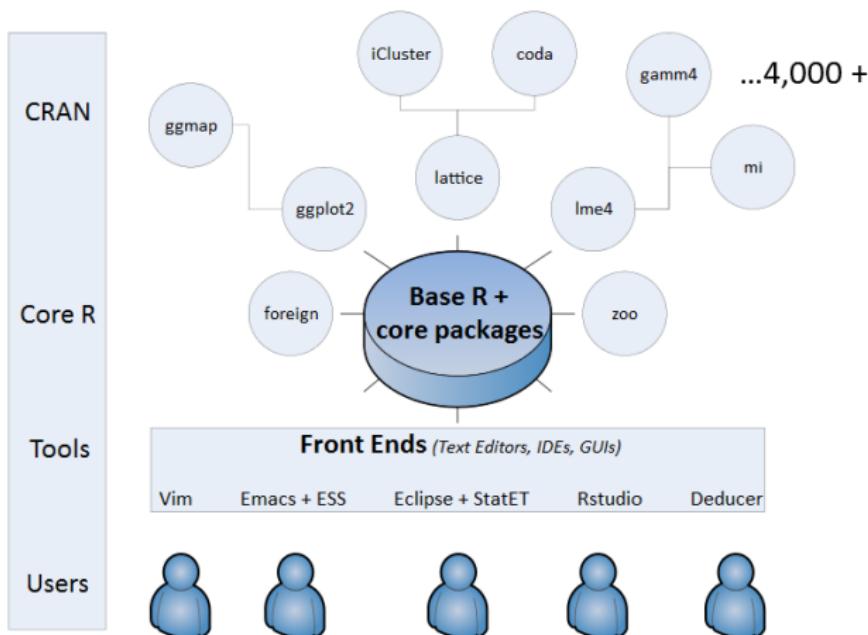
```
#Grundrechenarten
```

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

```
#Mathematische Funktionen
```

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

# Modularer Aufbau



Quelle: <http://www.ats.ucla.edu/stat/r/seminars/intro.htm>

## 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](http://www.bioconductor.org)

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

```
library(lme4)
```

└ Dein Freund das GUI

└ Modularer Aufbau

# Installation von Paketen

The screenshot shows the RStudio interface with several windows open:

- Code Editor:** A script named "paths.R" containing the code `setwd("D:/Projekte/R/packages/germanwebr/Rfunctions")`.
- Environment:** Shows the message "Environment is empty".
- Console:** 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:** A window listing available packages with their descriptions and versions. The packages listed are:

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)	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

└ Dein Freund das GUI

└ Modularer Aufbau

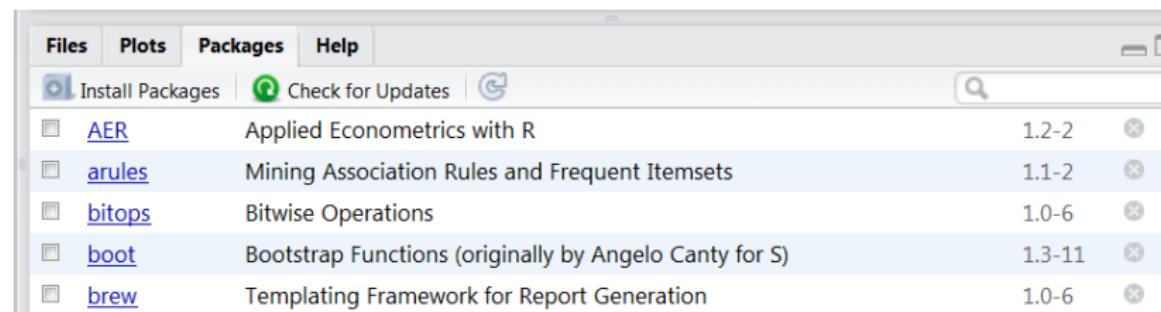
# Installation von Paketen

The screenshot shows the RStudio interface with several windows open:

- Code Editor:** A script named "paths.R" containing the code `setwd("D:/Projekte/R/packages/germanwebr/Rfunctions")`.
- Environment:** Shows the message "Environment is empty".
- Console:** 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:** A window listing available packages with their descriptions and versions. The packages listed are:

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)	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 2 - Interessante Zusatzpakete

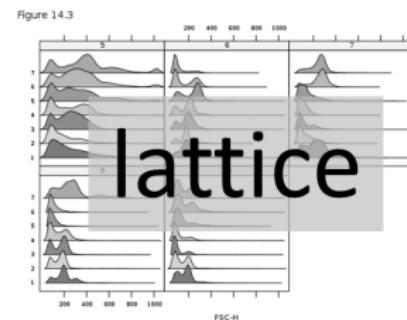
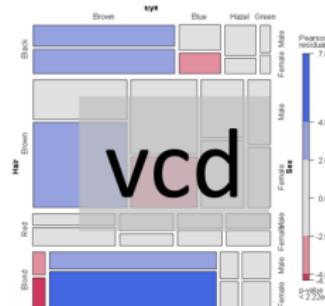
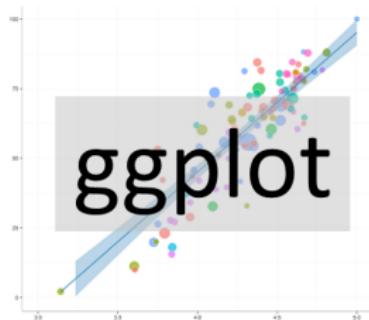
Gehen Sie auf [cran.r-project.org](http://cran.r-project.org) und suchen Sie in dem Bereich, wo die Pakete vorgestellt werden, nach Paketen,...

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

Dein Freund das GUI

Modularer Aufbau

## Pakete - deskriptive Datenanalyse



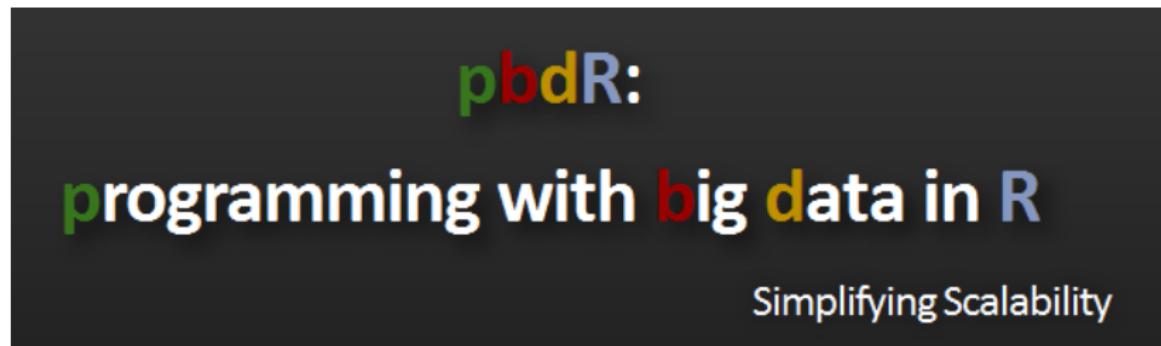
## Pakete - Regression

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

## Paket - big data (pbdR)



<http://r-pbd.org/>

[http://en.wikipedia.org/wiki/Programming\\_with\\_Big\\_Data\\_in\\_R](http://en.wikipedia.org/wiki/Programming_with_Big_Data_in_R)

## Paket - bigdata

`bigdata: Big Data Analytics`

The big data package is a collection of scalable methods for large-scale data analysis.

Version: 0.1

Depends: [glmnet](#), [Matrix](#), [lattice](#)

Published: 2012-04-06

Author: Han Liu, Tuo Zhao

Maintainer: Han Liu <hanliu at cs.jhu.edu>

License: [GPL-2](#)

NeedsCompilation: no

CRAN checks: [bigdata results](#)

---

Weitere Links zum Thema:

[http://blog.revolutionanalytics.com/2013/12/  
tips-on-computing-with-big-data-in-r.html](http://blog.revolutionanalytics.com/2013/12/tips-on-computing-with-big-data-in-r.html)

[http://statistics.org.il/wp-content/uploads/2010/04/  
Big\\_Memory%20V0.pdf](http://statistics.org.il/wp-content/uploads/2010/04/Big_Memory%20V0.pdf)

## 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

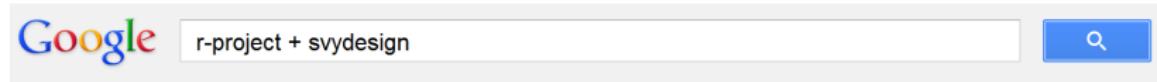
Bibliothek	Thema
xtable	Coerce data to LaTeX and HTML tables
dummies	Expands factors, characters and other eligible classes into dummy/indicator variables.
mvtnorm	Multivariate Normal and t Distributions
maptools	Tools for reading and handling spatial objects

## 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

└ Dein Freund das GUI

└ Wie bekommt man Hilfe?



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

- └ Dein Freund das GUI
- └ Wie bekommt man Hilfe?

The screenshot shows the top navigation bar of Stack Overflow with links for sign up, log in, tour, help, careers 2.0, and search. Below the header is the Stack Overflow logo and navigation links for Questions, Tags, Tour, and Users. A prominent 'Ask Question' button is on the right. The main content area features a large text block about the site's purpose, a 'Take the 2-minute tour' button, and a 'Here's how it works:' section with three icons: a question mark icon, a green 'A' icon, and a green 'A' icon with arrows.

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Here's how it works:

- Anybody can ask a question
- Anybody can answer
- The best answers are voted up and rise to the top

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

## Gliederung

R kam, sah und blieb

Dein Freund das GUI

Grundlagen im Umgang mit der Sprache R

R ist eine Objekt-orientierte Sprache

Indizieren

Rein und raus – Datenimport und -export

Ein erster Eindruck – Was uns die Daten sagen

Umkodieren, Plausibilisieren

## 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

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

## 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.<sup>6</sup>

## Aufgabe 3 - 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

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

## 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
```

## Listen

- ▶ Eine Liste in R entspricht einem geschachtelten Array in anderen Programmiersprachen
- ▶ Listen können alles enthalten
- ▶ Listen können geschachtelt sein
- ▶ Listen sollte man sehr bedacht verwenden

## 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

## 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
```

## Gliederung

R kam, sah und blieb

Dein Freund das GUI

Grundlagen im Umgang mit der Sprache R

Rein und raus – Datenimport und -export

Datemimport

Datenexport

Ein erster Eindruck – Was uns die Daten sagen

Umkodieren, Plausibilisieren

## R-Script - Daten einlesen

The screenshot shows a GitHub repository page for 'Japhilko / DLR\_IntroR'. The repository has 1 star and 0 forks. The file 'DLR\_IntroR\_B\_DatenEinlesen.R' is displayed, which contains an R script for data import. The script includes comments for importing foreign files and defining paths.

```
1 #####  
2 # DLR Intro Berlin  
3 # Jan-Philipp Kolb  
4 # 05.11.2014  
5 #  
6 # Daten einlesen  
7 #  
8 #####  
9  
10 #-----#  
11 # Bibliotheken einlesen  
12 #-----#  
13  
14 library(foreign)  
15 library(xlsx)  
16  
17 #-----#  
18 # Pfade definieren  
19 #-----#  
20  
21 main.path <- "J:/Work/Statistik/Kolb/Workshops/2014/WorkshopIntroBerlin/"
```

## Datenimport



## Datenzugang

### Public-Use-File (PUF)

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

[www.forschungsdatenzentrum.de](http://www.forschungsdatenzentrum.de)

[www.statistik-portal.de](http://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:

- ▶ Forschungsdatenzentrum  
[www.forschungsdatenzentrum.de/](http://www.forschungsdatenzentrum.de/)
- ▶ 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>

## Dateiformate in R

- ▶ Von R werden quelloffene, nicht-proprietäre 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.

## 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"
```

## Die Funktion paste

Die Funktion `paste()` kann verwendet werden um Verzeichnisse zu benennen:

```
main.path <- "J:/WorkshopIntroBerlin/"  
  
data.path <- paste(main.path, "data/", sep = "")  
  
graph.path <- paste(main.path, "graphs/", sep = "")
```

## SPSS-Dateien einlesen

Die `foreign`- Bibliothek ist wichtig um fremde Datensätze einzulesen:

```
library(foreign)
```

Der Pfad muss spezifiziert werden:  
(dieser ist bei jedem Rechner anders)

```
setwd("C:/")
```

`read.spss()` wird dann verwendet um die Daten einzulesen:

## SPSS Dateien einlesen

- ▶ Das (ebenfalls proprietäre) .sav Format lässt sich mit der Funktion `read.spss()` aus dem Paket `foreign` lesen
- ▶ Es müssen bei `read.spss()` allerdings einige Argumente gesetzt werden
  - ▶ `to.data.frame=TRUE` sorgt dafür, dass ein Data Frame entsteht (default ist `FALSE`)
  - ▶ `use.value.labels=FALSE` verwendet numerische Werte anstelle von Labels (default ist `TRUE`)

### SPSS mit `read.spss()`

```
library(foreign)
?read.spss
midcar <- read.spss("MiD2008_PUF_Auto.sav",
                      to.data.frame = T,
                      use.value.labels = F)
```

## SPSS-Dateien einlesen

```
midcar <- read.spss("MiD2008_PUF_Auto.sav",
                      to.data.frame = T,
                      use.value.labels =T)
```

```
midcar2 <- read.spss("MiD2008_PUF_Auto.sav",
                      to.data.frame = T,
                      use.value.labels =F)
```

## 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.

## Die Funktion `read.csv()`

```
sample <- read.csv(  
  "http://www.ats.ucla.edu/stat/examples/ara/angell.txt")
```

- ▶ Ähnlich funktioniert auch `read.table()`
- ▶ Wenn gar nichts sonst geht mit `readLines()` versuchen

## Das Paket xlsx

Der Datensatz wird zunächst als Excel-File abgespeichert, um Beispiel zu haben:

```
library(xlsx)  
write.xlsx(sample,"sample.xlsx")
```

Und das erste Blatt wird wieder eingelesen:

```
sample2 <- read.xlsx("sample.xlsx",1)
```

## STATA Dateien einlesen

Mit dem `foreign` paket können auch STATA (.dta) Dateien eingelesen werden:

STATA mit `read.dta()`

```
?read.dta  
midpers <- read.dta("MiD2008_PUF_Personen.dta")
```

# Datenmanagement wie in SPSS oder Stata

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

Rz · dataset1

File Preferences Help

dataset1 ( spss\_cf\_sohi98.sav )

Variables Descriptive statistics Plot

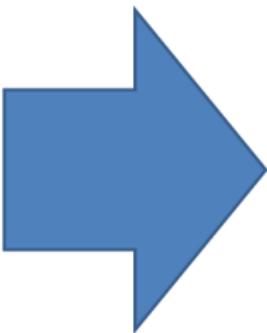
Mea: Names Labels Value Labels Missing

	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 Ausbildun"	
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_situ	besondere soziale Situation, 1: Möglichkeit, 1: Tod eines Par		

## Aufgabe 4 - Datenimport

- ▶ Laden Sie den Personen-Datensatz der Studie Mobilität in Deutschland 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`)

## Einfache Textdateien schreiben

- ▶ Mit Hilfe von `write.table()` lässt sich ein Data Frame sehr einfach in eine Textdatei schreiben
- ▶ Einige Argumente können jedoch angepasst werden
- ▶ Als Trenner sollte ein Tabulator dienen (default ist Leerzeichen)
- ▶ Zeilennamen sollten nicht geschrieben werden (default ist TRUE)

### `write.table()`

```
?write.table
write.table(sample,file="sample.txt",
            sep="\t", na="", dec=",", row.names=FALSE)
?write.csv
```

## Stata-dateien schreiben

Analog zu dem Befehl `read.dta()` gibt es auch `write.dta()`:

```
write.dta(sample, file="sample.dta")
```

## SPSS-Dateien schreiben

Für SPSS-Daten gibt es den Befehl `wirte.foreign`:

```
write.foreign(sample, "sample.txt", "sample.sps",  
package="SPSS")
```

# QuickR Daten-Export

<http://www.statmethods.net/input/exportingdata.html>

The screenshot shows the Quick-R website with a blue header bar containing navigation links: Home, Interface, Input, Manage, Stats, Adv Stats, Graphs, Adv Graphs, and Blog. Below the header is a search bar. The main content area has a blue sidebar on the left with the following menu items: Data Input (selected), Data types, Importing Data, Keyboard Input, Database Input, Exporting Data (selected), Viewing Data, Variable Labels, Value Labels, Missing Data, and Date Values. The main content area features a section titled "Exporting Data" with the sub-section "To A Tab Delimited Text File". It contains the R code: 

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

. Below this is another section titled "To an Excel Spreadsheet" with the R code: 

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

.

## Objekt(e) in R-Workspaces sichern

- ▶ In einem Workspace kann ein oder mehrere Objekte gespeichert werden
- ▶ Workspaces sind komprimierte Binärdateien (sehr klein)
- ▶ Der Befehl `save.image()` speichert alle Objekte in einem Workspace
- ▶ Dem Befehl `save()` kann eine Liste von Objekten (auch nur eines) übergeben werden

### `save.image()` und `save()`

```
?save.image  
save.image("Day01.RData")  
  
?save  
save(midpers,file="midpers.RData")
```

# Überblick Daten Import/Export

## 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

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  
Häufigkeiten und gruppierte Kennwerte  
Die apply-Familie

Umkodieren, Plausibilisieren

## Mittelwert

Den Mittelwert erhält man mit der Funktion `mean()`

```
mean(midpers$wege3)
```

Allerdings befinden sich oft fehlende Werte in den Daten:

```
mean(midpers$wege3, na.rm=T)
```

Den gewichteten Mittelwert erhält man mit:

```
mean(midpers$wege3 * midpers$p_gew, na.rm=T)
```

## Auswertungen SPSS

Versuchen Sie diese Ergebnisse in R nachzuvollziehen:

MiD2008-Personendatensatz, gewichtet mit p_gew	Mittelwert
wege3 Wege insgesamt inkl. weiterer Wege und rbW (wege2 + rbw04)	3,8
anzkm Tagesstrecke [km] (Summe wegkm_k)	44
anzmin Unterwegszeit [min] (Summe wegmin_k)	88

## Auswertungen SPSS mit Filter

mit Filter: **mobil = 1** (d.h. hier sollen nur Personen betrachtet werden, die am Stichtag außer Haus waren)

MiD2008-Personendatensatz, gewichtet mit p_gew	Mittelwert
<b>wege3</b> Wege insgesamt inkl. weiterer Wege und rbW (wege2 + rbw04)	3,8
<b>anzkm</b> Tagesstrecke [km] (Summe wegkm_k)	44
<b>anzmin</b> Unterwegszeit [min] (Summe wegmin_k)	88

## SPSS Ergebnis mit R

- ▶ Mittelwert mit Filter: `mobil = 1`
- ▶ nur Personen werden betrachtet, die am Stichtag außer Haus waren

```
mobilTrue <- midpers[midpers$mobil=="mobil",]  
mean(mobilTrue$wege3*mobilTrue$p_gew , na.rm=T)
```

## 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(midpers$wege3)  
var(midpers$wege3, 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(midpers$mobil)  
  
table(midpers$mobil, midpers$hp_sex)
```

- └ Ein erster Eindruck – Was uns die Daten sagen
- └ Häufigkeiten und gruppierte Kennwerte

## Häufigkeitstabellen

	V1
nicht mobil	5862
mobil	51333
Mobilität unbekannt	760
Erwachsener > 13 J. (Selbst) o. Wegeprot. (StTag > 3 Tage)	2679
Kind 10-13 J. (Selbst) o. Wegeprot. (StTag > 3 Tage)	79

## 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(midpers$mobil)
# oben u unten Erg gleich
with(midpers,table(mobil))

?prop.table
prop.table(table(midpers$mobil))
```

## Mehrdimensionale Häufigkeitstabellen

```
table(midpers$mobil, midpers$hp_sex)
```

	männlich	weiblich	verweigert	weiß nicht	keine Angabe
nicht mobil	2618	3244	0	0	0
mobil	25650	25679	0	0	4
Mobilität	388	372	0	0	0
unbekannt					

## 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

- └ Ein erster Eindruck – Was uns die Daten sagen
  - └ Häufigkeiten und gruppierte Kennwerte

## Beispiel aggregate() Funktion

```
aggregate(midpers$anzkm, list(midpers$hp_sex), mean)
```

	Group.1	x
1	männlich	662.65
2	weiblich	699.68
3	keine Angabe	20.09

## Exkurs - die Funktion `sample()`

`sample {base}`

R Documentation

### Random Samples and Permutations

#### Description

`sample` takes a sample of the specified size from the elements of `x` using either with or without replacement.

#### Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

x: From what do we want  
to sample ?

```
sample(x=1:10,n=1,replace=T)
```

## Exkurs - die Funktion `sample()`

`sample {base}`

R Documentation

### Random Samples and Permutations

#### Description

`sample` takes a sample of the specified size from the elements of `x` using either with or without replacement.

#### Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

n: How many elements  
do we want to draw?

`sample(x=1:10, n=1, replace=T)`

## Exkurs - die Funktion `sample()`

`sample {base}`

R Documentation

### Random Samples and Permutations

#### Description

`sample` takes a sample of the specified size from the elements of `x` using either with or without replacement.

#### Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

Do we want to draw with  
or without replacement?

```
sample(x=1:10, n=1, replace=T)
```

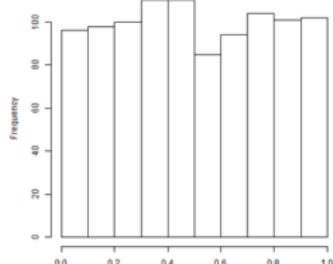
## Exkurs - Zufallszahlen in R

Funktion	Verteilung	Wichtige Parameter
runif	Gleichverteilung	n, min, max
rnorm	Normalverteilung	n, mean, sd
rpois	Poisson-Verteilung	n, lambda
...	...	...

## Exkurs - Zufallszahlen in R

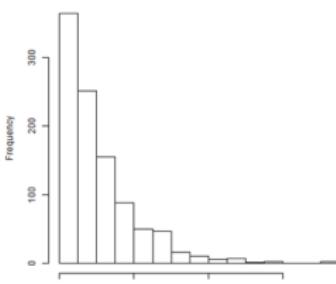
```
png("Histx2.png")
hist(x2,main="")
dev.off()
```

```
x3 <- rexp(1000)
```



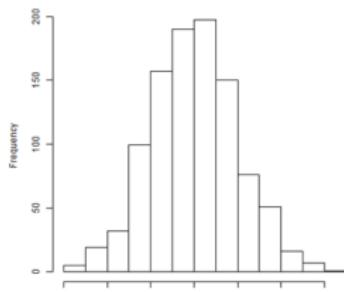
A

```
x1 <- runif(1000)
```



B

```
x2 <- rnorm(1000)
```

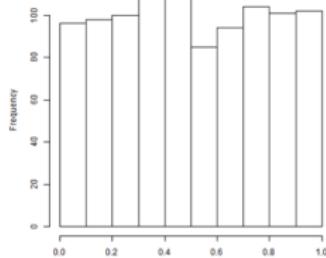


C

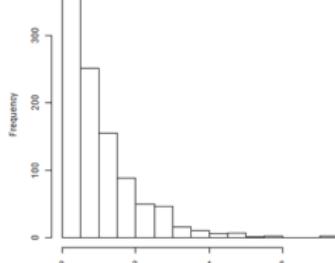
## Exkurs - Zufallszahlen in R

```
png("Histx2.png")
hist(x2,main="")
dev.off()
```

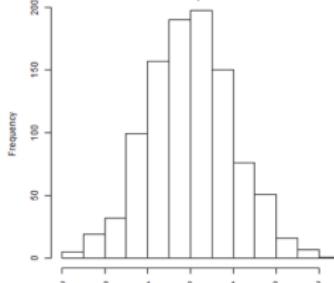
x3 <- rexp(1000)



x1 <- runif(1000)



x2 <- rnorm(1000)



# 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

```
x <- cbind(x1 = 3, x2 = c(4:1, 2:5))  
x  
  
apply(x, 2, mean)  
apply(x, 1, mean)
```

## 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.

# Auswertungen SPSS

## Mobilitätsquote nach Geschlecht

MiD2008-Personendatensatz, gewichtet mit p_gew		hp_sex Geschlecht (fehlende Angaben ergänzt aus HH- Interview)	
		1 männlich	2 weiblich
mobil Mobilität 0 nicht mobil am Stichtag	Anzahl der Spalten (%)	Anzahl der Spalten (%)	
	9%	11%	
	91%	89%	
Gesamt	100%	100%	

## SPSS Ergebnis mit R

```
with(midpers, tapply(p_gew, list(hp_sex, mobil), sum))
```

Zwei Probleme:

- ▶ Kategorie 9 ist jeweils noch enthalten
- ▶ Absolute und keine relativen Werte

## SPSS Ergebnis mit R

9 wird als fehlender Wert kodiert:

```
Pgew <- midpers$p_gew
sex <- midpers$hp_sex
sex[sex==9] <- NA
mob <- midpers$mobil
mob[mob==9] <- NA
```

Jetzt kann die korrekte Häufigkeit berechnet werden:

```
Tab1 <- tapply(Pgew, list(sex,mob), sum)
apply(Tab1, 1, prop.table)
```

## SPSS Ergebnis mit R

	männlich	weiblich
nicht mobil	0.09	0.11
mobil	0.90	0.87
Mobilität unbekannt	0.01	0.01

- └ Ein erster Eindruck – Was uns die Daten sagen
- └ Die apply-Familie

## Auswertungen SPSS

MiD2008-Wegedatensatz, gewichtet mit w_gew		ktyp_zsg Zusammengefasster Kreistyp nach ROB2005			
		1 Kernstädte	2 Verdichtete Kreise	3 Ländliche Kreise	Gesamt
		Anzahl der Spalten (%)	Anzahl der Spalten (%)	Anzahl der Spalten (%)	Anzahl der Spalten (%)
hvm Hauptverkehr smittel	1 zu Fuß	27%	22%	23%	24%
	2 Fahrrad	10%	10%	10%	10%
	3 MIV (Mitfahrer)	13%	16%	16%	15%
	4 MIV (Fahrer)	36%	45%	46%	43%
	5 ÖPV	15%	6%	5%	8%
	Gesamt	100%	100%	100%	100%

## 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>

## Gliederung

R kam, sah und blieb

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

Umkodieren, Plausibilisieren

# Werte umkodieren - SPSS

tempo: Wegegeschwindigkeit [km/h] korrigiert

- Quotient aus Wegelänge und Wegedauer und Runden auf 2 Dezimalstellen

```
COMPUTE tempo = wegkm_k / (wegmin_k / 60).
COMPUTE tempo = RND(tempo * 100) / 100.
EXECUTE.
```

- Fehlende Werte kodieren

➤ Bei rbW sind keine Start- und Ankunftszeiten bekannt, entsprechend existiert keine Wegedauer, und somit lässt sich auch keine Geschwindigkeit berechnen.



rbW

```
DO IF (wsource = 1).
RECODE tempo (SYSMIS=30301).
END IF.
EXECUTE.
```

```
RECODE tempo (SYSMIS=99995).
EXECUTE.
```

## Werte umkodieren - SPSS/R

tempo: Wegegeschwindigkeit [km/h] korrigiert

- Quotient aus Weglänge und Wegdauer und Runden auf 2 Dezimalstellen

```
COMPUTE tempo = wegkm_k / (wegmin_k / 60).
COMPUTE tempo = RND(tempo * 100) / 100.
EXECUTE.
```

```
attach(midway)
tempo=wegkm_k/(wegmin_k/60)

tempo=round(tempo/100,digits=2)/100
```

# Fehlende Werte umkodieren - SPSS/R

## ■ Fehlende Werte kodieren

- Bei rbW sind keine Start- und Ankunftszeiten bekannt, entsprechend existiert keine Wegedauer, und somit lässt sich auch keine Geschwindigkeit berechnen.



```
DO IF (wsource = 1).
RECODE tempo (SYSMIS=30301).
END IF.
EXECUTE.

RECODE tempo (SYSMIS=99995).
EXECUTE.
```

```
tempo [wsource ==1] <- NA
```

## Fehlende Werte umkodieren - SPSS/R

### ■ Plausibilisierung

\*\*\*minimale Geschwindigkeit beträgt 0,5km/h.

**RECODE tempo (Lowest thru 0.49=99994).**

**EXECUTE.**

```
tempo [tempo < .5] <- .5
```

## Werte umkodieren - R

\*\*\*Fußweg max. 20 km/h.

**DO IF** (w05 = 1).

**RECODE** tempo (20.01 thru 30300=99994).

**END IF.**

**EXECUTE.**

Bei mehreren Bedingungen:

```
tempo [w05_1==1 & tempo>20] <- NA
```

# Werte umkodieren - SPSS

## Tempo - Plausibilisierung

Mobilität in Deutschland

38

### Plausibilisierung

\*\*\*minimale Geschwindigkeit beträgt 0,5km/h.  
**RECODE tempo (Lowest thru 0.49=99994).**  
**EXECUTE.**

\*\*\*Fußweg max. 20 km/h.  
**DO IF (w05 = 1).**  
**RECODE tempo (20.01 thru 30300=99994).**  
**END IF.**  
**EXECUTE.**

### Analoge Syntax zur Plausibilisierung für die anderen Verkehrsmittel:

- Fahrrad: max. 50 km/h
- ÖPNV: max. 150 km/h
- Motorrad, Pkw, sonstiges: max: 250 km/h
- ÖPFV: max 900 km/h

### Anpassen des Formats der Missing-Codes, da nur noch max. 900 auftritt

**RECODE tempo (30301=3301) (99994=994) (99995=995).**  
**EXECUTE.**

**VALUE LABELS tempo**

994 'unplausibler Wert'  
 995 'Wert nicht zu berechnen'

## Plausibilisierungen - R

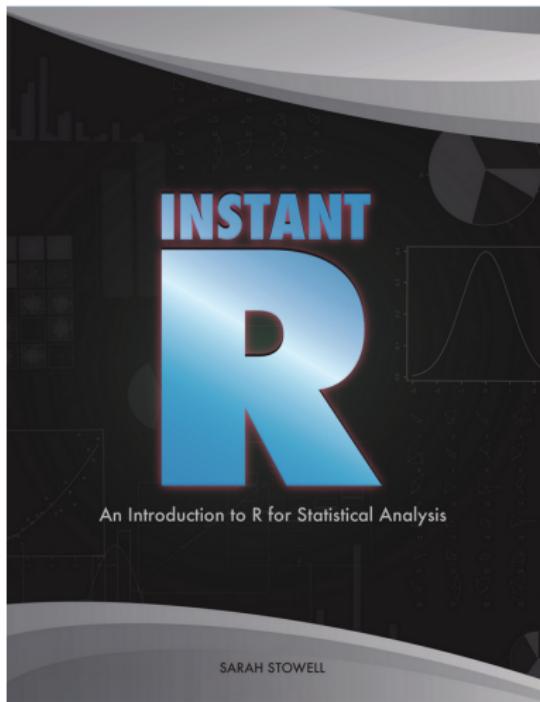
```
min(tempo,na.rm=T)  
  
tempo[tempo<.5] <- .5  
min(tempo,na.rm=T)
```

## Literatur



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

## Literatur



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

# Mehr Hilfe für Anfänger



INSTITUTE FOR DIGITAL RESEARCH AND EDUCATION  
UCLA

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"><li>• to learn R</li><li>• to utilize basic statistical procedures</li></ul>	<ul style="list-style-type: none"><li>• how to do a simple task in R</li><li>• how to interpret the output from commonly used procedures</li></ul>	<ul style="list-style-type: none"><li>• on how to do basic tasks in R</li><li>• on frequently used statistical procedures and the interpretation of their output.</li></ul>

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](http://www.stamats.de/InstallationUndErsteSchritteMitR.pdf)
- ▶ Einführung direkt auf CRAN:  
[cran.r-project.org/doc/contrib/Sawitzki-Einfuehrung.pdf](http://cran.r-project.org/doc/contrib/Sawitzki-Einfuehrung.pdf)

# Vielen Dank für die Aufmerksamkeit

MY HOBBY: EXTRAPOLATING

