

# INTRO DATENANALYSE MIT R

Jan-Philipp Kolb

06 Mai, 2019

# WARUM R NUTZEN

# UM REIN ZU KOMMEN

## KLEINE VORSTELLUNGSRUNDE

- ▶ Wo kommt Ihr her?
- ▶ Wo arbeitet und studiert Ihr?
- ▶ Habt Ihr Erfahrungen mit Programmiersprachen / Statistiksoftware? Wenn ja welche?
- ▶ Was sind Eure Erwartungen für diesen Kurs?

# DISCLAIMER/ INFORMATIONEN VORAB

Normalerweise gibt es große Unterschiede bei Vorkenntnissen und Fähigkeiten - bitte gebt Bescheid, wenn es zu schnell oder zu langsam geht oder etwas unklar geblieben ist.

- ▶ Wenn es Fragen gibt - immer fragen
- ▶ In diesem Kurs gibt es viele **Übungen**, denn das Programmieren / die Nutzung von R lernt man am Ende nur allein.
- ▶ Ich habe viele **Beispiele** - probiert sie aus
- ▶ R macht mehr Spaß zusammen - arbeitet zusammen!

# ERWARTUNGEN UND ANFORDERUNGEN

## DAS KANN DIESE SCHULUNG VERMITTELN:

- ▶ Eine praxisnahe Einführung in die statistische Programmiersprache R
- ▶ Erlernen einer Programmier-Strategie
- ▶ Guten Stil
- ▶ Die Vorzüge graphischer Datenanalyse

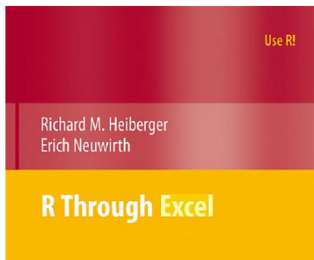
## DAS KANN SIE NICHT LEISTEN:

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

# GRÜNDE R ZU NUTZEN...

- ▶ ... R ist eine **quelloffene Sprache**
- ▶ ... hervorragende **Grafiken, Grafiken, Grafiken**
- ▶ ... **R kann in Kombination mit anderen Programmen verwendet werden** - z.B. zur **Verknüpfung von Daten**
- ▶ ... R kann **zur Automatisierung** verwendet werden
- ▶ ... Breite und aktive Community - **Man kann die Intelligenz anderer Leute nutzen ;-)**

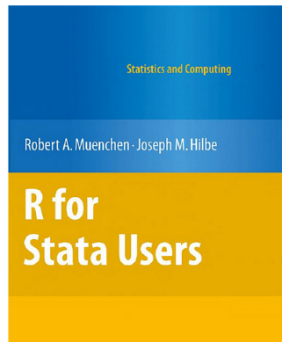
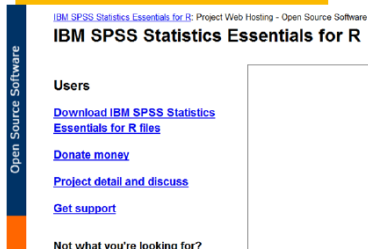
# R KANN IN KOMBINATION MIT ANDEREN PROGRAMMEN GENUTZT WERDEN...



SASmixed



rPython R package



# DIE POPULARITÄT VON R





## R NUTZUNG ...

## WEIL ANDERE PROGRAMME FEHLER PROVOZIEREN KÖNNEN

# FAQ: Reinhart, Rogoff, and the Excel Error That Changed History

By Peter Coy

12,26	38,4
1,4	26,1
29,5	32,1
0,45	+1,41
G	1820
21,73	26,35
3,48	-0,12
12772661	-11,4
22	21,26
565	

# EXCEL BUGS

## DATUM IN EXCEL

BEST PRACTICES, DATA MANAGEMENT, EXCEL, R, TOOLS FOR DATA

# Abandon all hope, ye who enter dates in Excel

by [Kara Woo](#) • April 9, 2014 • Comments Off

## PROBLEME MIT EXCEL



## BMC Bioinformatics

HOME

ABOUT

ARTICLES

SUBMISSION GUIDELINES

CORRESPONDENCE

OPEN ACCESS

Mistaken Identifiers: Gene name errors can be introduced  
inadvertently when using Excel in bioinformatics

# R HERUNTERLADEN:

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



*CRAN*

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

*About R*

[R Homepage](#)

[The R Journal](#)

*Software*

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

## The Comprehensive R Archive Network

### Download and Install R

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

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

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

### Source Code for all Platforms

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

- The latest release (Friday 2017-04-21, You Stupid Darkness)  
[R-3.4.0.tar.gz](#), read [what's new](#) in the latest version.

# LINKS

- ▶ Warum man R für Data Science lernen sollte
- ▶ R Technologie des Jahres
- ▶ Why R is Good for Business
- ▶ Warum R auf r-bloggers
- ▶ Intro R
- ▶ Intro R II
- ▶ Vergleich python und R

# VERGLEICH MIT ANDEREN PROGRAMMEN



INWT Statistics

[HOME](#)

[BUSINESS CASES](#)

[MEET THE TEAM](#)

[TRAINING](#)

[BLOG](#)

[Start a Project!](#)

## Blog

[INWT](#) > [Blog](#) > Statistik-Software: R, SAS, SPSS und STATA im Vergleich

Weitere Teile der Artikelserie zu Stärken und Schwächen gängiger Statistik-Software:

- Checkliste für die Anschaffung von Statistik-Software
- **Statistik-Software: R, SAS, SPSS und STATA im Vergleich**
- Die mächtige Open Source-Lösung: R

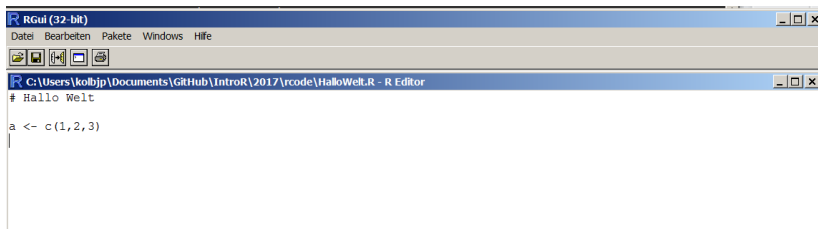
## Statistik-Software: R, SAS, SPSS und STATA im Vergleich

# DEIN FREUND DAS GUI

# 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)
- ▶ Der Download ist auf dieser Seite möglich:

<https://cran.r-project.org/>



# GRAPHISCHES USER INTERFACE

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

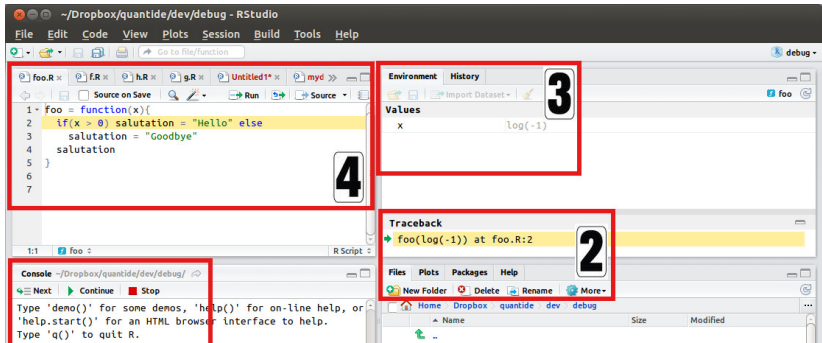
Aus den folgenden Gründen:

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



# VERSCHIEDENE GUIs

- ▶ Gedit mit R-spezifischen Add-ons für Linux
- ▶ Emacs
- ▶ TinnR
- ▶ Ich nutze Rstudio!



# LINKS ZU RSTUDIO

- ▶ Sechs **Gründe** Rstudio zu nutzen.
- ▶ Wie man Rstudio **nutzen kann**.
- ▶ **Rstudio einrichten**
- ▶ **Einführung in RStudio**
- ▶ **RStudio Cheatsheet**

## AUFGABE - VORBEREITUNG

- ▶ Prüfen Sie, ob eine Version von R auf Rechner installiert ist.
- ▶ Falls dies nicht der Fall ist, laden Sie R runter und installieren Sie R.
- ▶ Prüfen Sie, ob Rstudio installiert ist.
- ▶ Falls nicht - Installieren sie Rstudio.
- ▶ Laden Sie die R-Skripte von meinem GitHub-Account
- ▶ Erstellen Sie ein erstes Script und finden Sie das Datum mit dem Befehl `date()` und die R-version mit `sessionInfo()` heraus.

```
date()
```

```
## [1] "Mon May 06 14:52:17 2019"
```

```
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
```

# GRUNDLAGEN IM UMGANG MIT DER SPRACHE R

# R IST EINE OBJEKT-ORIENTIERTE SPRACHE

## 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
```

```
## [1] 1.5
```

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

```
length(b) # b hat die Länge 2
```

```
## [1] 2
```

# OBJEKTSTRUKTUR

```
str(b) # b ist ein numerischer Vektor
```

```
## num [1:2] 1 2
```

# FUNKTIONEN IM BASE-PAKET

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

Diese Funktionen brauchen nur ein Argument.

# FUNKTIONEN MIT MEHR ARGUMENTEN

Andere Funktionen brauchen mehr:

Argument	Bedeutung	Beispiel
quantile()	90 % Quantile	quantile(b,.9)
sample()	Stichprobe ziehen	sample(b,1)



## BEISPIEL - FUNKTIONEN MIT EINEM ARGUMENT

```
max(b)
```

```
## [1] 2
```

```
min(b)
```

```
## [1] 1
```

```
sd(b)
```

```
## [1] 0.7071068
```

```
var(b)
```

```
## [1] 0.5
```

# FUNKTIONEN MIT EINEM ARGUMENT

```
mean(b)
```

```
## [1] 1.5
```

```
median(b)
```

```
## [1] 1.5
```

# FUNKTIONEN MIT MEHR ARGUMENTEN

```
quantile(b,.9)
```

```
## 90%
```

```
## 1.9
```

```
sample(b,1)
```

```
## [1] 2
```

# ÜBERSICHT BEFEHLE

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

## An Introduction to R

### Table of Contents

#### [Preface](#)

#### [1 Introduction and preliminaries](#)

##### [1.1 The R environment](#)

##### [1.2 Related software and documentation](#)

##### [1.3 R and statistics](#)

##### [1.4 R and the window system](#)

##### [1.5 Using R interactively](#)

##### [1.6 An introductory session](#)

##### [1.7 Getting help with functions and features](#)

##### [1.8 R commands, case sensitivity, etc.](#)

##### [1.9 Recall and correction of previous commands](#)

##### [1.10 Executing commands from or diverting output to a file](#)

# AUFGABE - 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

# WIE BEKOMMT MAN HILFE?

# WIE BEKOMME ICH HILFE?

- ▶ Um Hilfe im Allgemeinen zu bekommen:

```
help.start()
```

- ▶ Online-Dokumentation für die meisten Funktionen:

```
help(name)
```

- ▶ Benutze ?, um Hilfe zu bekommen

```
?mean
```

- ▶ `example(lm)` liefert ein Beispiel für die lineare Regression

```
example(lm)
```

# VIGNETTEN

- ▶ Eine Vignette ist ein Papier, das die wichtigsten Funktionen eines Pakets darstellt.
- ▶ Sie enthalten viele reproduzierbare Beispiele.
- ▶ Vignetten sind ein neues Werkzeug, deshalb hat nicht jedes Paket eine Vignette.

```
browseVignettes()
```

- ▶ Um eine Vignette zu bekommen:

```
vignette("osmdata")
```



# EIN BEISPIEL FÜR EINE VIGNETTE - DAS PAKET OSMDATA

<https://cran.r-project.org/web/packages/osmdata/vignettes/osmdata.html>

## 1. Introduction

`osmdata` is an R package for downloading and using data from OpenStreetMap ([OSM](#)). OSM is a global open access mapping project, which is free and open under the [ODbL licence](#) [[@OpenStreetMap](#)]. This has many benefits, ensuring transparent data provenance and ownership, enabling real-time evolution of the database and, by allowing anyone to contribute, encouraging democratic decision making and citizen science [[@johnson\\_models\\_2017](#)]. See the [OSM wiki](#) to find out how to contribute to the world's open geographical data commons.

Unlike the [OpenStreetMap](#) package, which facilitates the download of raster tiles, `osmdata` provides access to the vector data underlying OSM.

`osmdata` can be installed from CRAN with

```
install.packages("osmdata")
```

and then loaded in the usual way:

```
library(osmdata)
```

```
## Data (c) openStreetMap contributors, ODbL 1.0. http://www.openstreetmap.org/copyright
```

The development version of `osmdata` can be installed with the `devtools` package using the following command:

# Demos

- für manche Pakete gibt es Demos:

```
demo() # zeigt alle verfügbaren Demos
demo(package = "httr") # Zeigt alle Demos in einem Paket

# Ein spezifisches Demo laufen lassen:
demo("oauth1-twitter", package = "httr")
```

- Wenn ein Demo gestartet wird, ist der zugehörige Code in der Konsole sichtbar

```
demo(nlm)
```

```
> demo(nlm)
```

```
demo(nlm)
```

# DIE FUNKTION APROPOS

- ▶ durchsucht alles über den angegebenen String:

```
apropos("lm")
```

```
## [1] ".colMeans"          ".lm.fit"             "colMeans"
## [4] "confint.lm"         "contr.helmert"       "dummy.coef.lm"
## [7] "getAllMethods"     "glm"                 "glm.control"
## [10] "glm.fit"           "KalmanForecast"      "KalmanLike"
## [13] "KalmanRun"         "KalmanSmooth"        "kappa.lm"
## [16] "lm"                "lm.fit"              "lm.influence"
## [19] "lm.wfit"           "model.matrix.lm"     "nlm"
## [22] "nlminb"            "predict.glm"         "predict.lm"
## [25] "residuals.glm"     "residuals.lm"        "summary.glm"
## [28] "summary.lm"
```

- ▶ Funktion kann auch mit **regulären Ausdrücken** verwendet werden

# Suchmaschine für die R-Seite

```
RSiteSearch("glm")
```

## R Site Search

Query:   [\[How to search\]](#)

Display:  Description:  Sort:

### Target:

- ☒ Functions
- ☒ Task views

For problems WITH THIS PAGE (not with R) contact [baron@upenn.edu](mailto:baron@upenn.edu).

## Results:

### References:

- **views:** [ glm: 11 ]
- **vignettes:** [ (can't open the index) ]
- **functions:** [ glm: 4391 ]

**Total 4402 documents matching your query.**

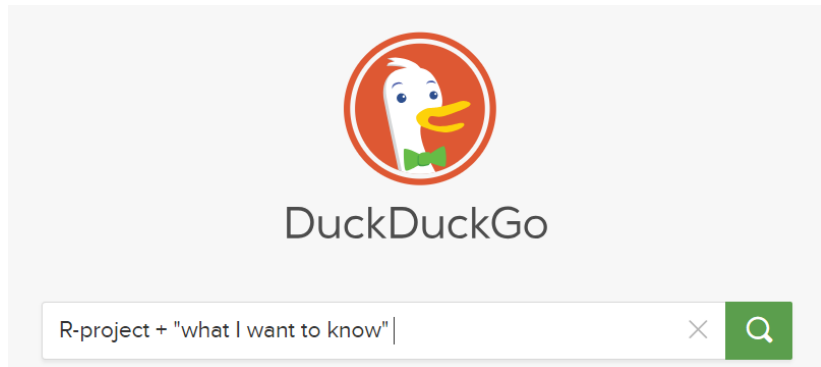
1. [R: Bias reduction in Binomial-response GLMs](#) (score: 299)

# NUTZUNG VON SUCHMASCHINEN

- ▶ Ich nutze **duckduckgo.de**:

R-project + "was ich schon immer wissen wollte"

- ▶ das funktioniert natürlich für alle Suchmaschinen!



# Stackoverflow

- ▶ Für alle Fragen zum programmieren
- ▶ Ist nicht auf R fokussiert - aber es gibt **viele Diskussionen zu R-Fragen**
- ▶ Sehr detaillierte Diskussionen

The screenshot shows the Stackoverflow homepage. At the top, there's a navigation bar with 'Questions', 'Jobs', 'Documentation', 'Tags', and 'Users'. A search bar contains '[r]'. On the right, there are links for 'Log In' and 'Sign Up'. Below the navigation bar, the 'Tagged Questions' section is active, showing filters for 'info', 'newest', 'featured', 'frequent', 'votes', 'active', and 'unanswered'. The 'frequent' filter is selected. A question titled 'How to make a great R reproducible example?' is displayed, with 1776 votes, 22 answers, and 147k views. The question text asks for guidance on creating reproducible examples. To the right of the question, there's a box for 'R Language DOCUMENTATION' with a link to 'Find a request to handle or browse 121 topics.' Below this, a 'Related Tags' section lists 'ggplot2' (2875), 'dataframe' (1351), and 'plot' (1105).

stackoverflow Questions Jobs Documentation BETA Tags Users  ? Log In Sign Up

Tagged Questions info newest featured frequent votes active unanswered

R is a free, open-source programming language and software environment for statistical computing, bioinformatics, and graphics. Please supplement your question with a minimal reproducible example. Use `dput()` for data and specify all non-base packages with library calls. For statistical questions ...

[learn more...](#) [top users](#) [synonyms \(2\)](#) [r jobs](#)

1776 votes

22 answers

147k views

**How to make a great R reproducible example?**

When discussing performance with colleagues, teaching, sending a bug report or searching for guidance on mailing lists and here on SO, a reproducible example is often asked and always helpful. What ...

[r](#) [r-faq](#)

community wiki  
11 revs, 8 users 54%  
Hack-R

22,187 frequent questions tagged

[r](#) [about »](#)

**R Language**  
DOCUMENTATION

[Find a request to handle](#) or [browse 121 topics](#).

**Related Tags**

[ggplot2](#) × 2875

[dataframe](#) × 1351

[plot](#) × 1105

FIGURE 3: Stackoverflow Beispiel

# EIN SCHUMMELZETTEL FÜR BASIS R

<https://www.rstudio.com/resources/cheatsheets/>

## Base R Cheat Sheet

### Getting Help

#### Accessing the help files

##### ?mean

Get help of a particular function.

**help.search('weighted mean')**

Search the help files for a word or phrase.

**help(package = 'dplyr')**

Find help for a package.

#### More about an object

##### str(iris)

Get a summary of an object's structure.

##### class(iris)

Find the class an object belongs to.

### Using Packages

#### install.packages('dplyr')

Download and install a package from CRAN.

#### library(dplyr)

Load the package into the session, making all its functions available to use.

#### dplyr::select

Use a particular function from a package.

#### data(iris)

Load a built-in dataset into the environment.

### Vectors

#### Creating Vectors

c(2, 4, 6)	2 4 6	Join elements into a vector
2:6	2 3 4 5 6	An integer sequence
seq(2, 3, by=0.5)	2.0 2.5 3.0	A complex sequence
rep(1:2, times=3)	1 2 1 2 1 2	Repeat a vector
rep(1:2, each=3)	1 1 1 2 2 2	Repeat elements of a vector

#### Vector Functions

<b>sort(x)</b> Return x sorted.	<b>rev(x)</b> Return x reversed.
<b>table(x)</b> See counts of values.	<b>unique(x)</b> See unique values.

#### Selecting Vector Elements

##### By Position

<b>x[4]</b>	The fourth element.
<b>x[-4]</b>	All but the fourth.
<b>x[2:4]</b>	Elements two to four.
<b>x[-(2:4)]</b>	All elements except two to four.
<b>x[c(1, 5)]</b>	Elements one and five.

### Programming

#### For Loop

```
for (variable in sequence){  
  Do something  
}
```

##### Example

```
for (i in 1:4){  
  j <- i + 10  
  print(j)  
}
```

#### While Loop

```
while (condition){  
  Do something  
}
```

##### Example

```
while (i < 5){  
  print(i)  
  i <- i + 1  
}
```

#### If Statements

```
if (condition){  
  Do something  
} else {  
  Do something different  
}
```

##### Example

```
if (i > 3){  
  print('Yes')  
} else {  
  print('No')  
}
```

#### Functions

```
function_name <- function(var){  
  Do something  
  return(new_variable)  
}
```

##### Example

```
square <- function(x){  
  squared <- x*x  
  return(squared)  
}
```

### Reading and Writing Data

Also see the **readr** package.

Input	Output	Description
<code>df &lt;- read.table('file.txt')</code>	<code>write.table(df, 'file.txt')</code>	Read and write a delimited text file.

FIGURE 4: Cheatsheet BaseR

# MEHR SCHUMMELZETTEL

## Regular Expressions



Basics of regular expressions and pattern matching in R by Ian Kopacka. Updated 09/16.

DOWNLOAD

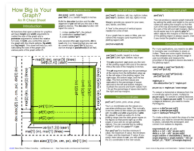
## The leaflet package



Interactive maps in R with leaflet, by Kejia Shi. Updated 05/17.

DOWNLOAD

## How big is your graph?



Graph sizing with base R by Stephen Simon. Updated 10/16.

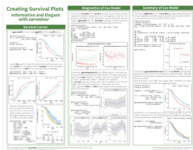
DOWNLOAD

## The eurostat package



R tools to access the eurostat database, by rOpenGov. Updated 03/17.

## The survminer package



Elegant survival plots, by Przemyslaw Biecek. Updated 03/17.

## The sjmisc package



dplyr friendly Data and Variable Transformation, by Daniel Lüdtke. Updated 08/17.



# Quick R

- ▶ Immer mit vielen Beispielen und Hilfen bezüglich eines Themas
- ▶ Beispiel: **Quick R - Getting Help**



R Tutorial | R Interface | Data Input | Data Management | Statistics | Advanced Statistics | Graphs | Advanced Graphs

## < R Interface

Getting Help

The Workspace

Input/Output

Packages

Graphic User Interfaces

Customizing Startup

Publication Quality Output

Batch Processing

Reusing Results

## Getting Help

Once R is installed, there is a comprehensive built-in help system. At the program's command prompt you can use any of the following:

```
help.start()  # general help
help(foo)     # help about function foo
?foo         # same thing
apropos("foo") # list all functions containing string foo
example(foo)  # show an example of function foo
```

# WEITERE LINKS

- **Überblick - wie bekommt man Hilfe in R**



[\[Home\]](#)

**Download**

[CRAN](#)

## Getting Help with R

### Helping Yourself

Before asking others for help, it's generally a good idea for you to try to help yourself. R includes extensive facilities for accessing documentation and searching for help. There are also specialized search engines for accessing information about R on the internet, and general internet search engines can also prove useful ([see below](#)).

- **Eine Liste mit HowTo's**
- **Eine Liste mit den wichtigsten R-Befehlen**

# AUFGABE Hilfe bekommen

## HILFE FÜR `WHICH.MIN`

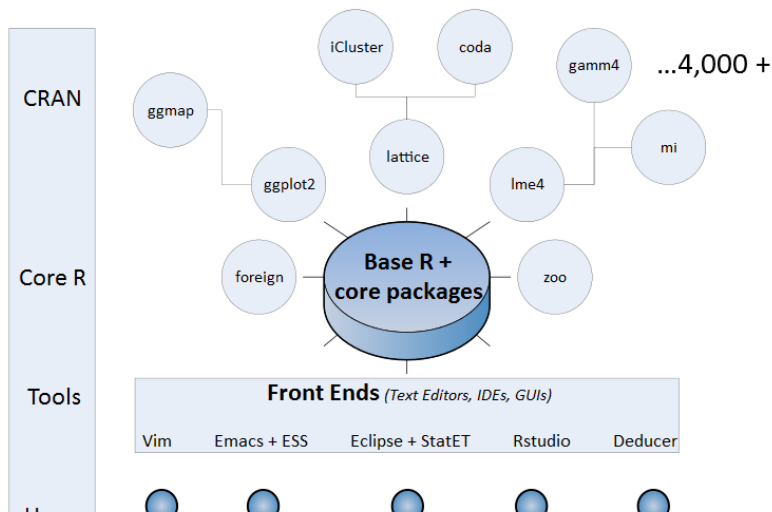
- ▶ Versuchen Sie den Befehl `?which.min`. Dies öffnet eine Hilfeseite im unteren rechten Fenster von RStudio. Was macht die Funktion?
- ▶ Sie müssen den Namen der Funktion kennen, um die Hilfeseite wie oben beschrieben zu öffnen. Manchmal (oft, sogar) kennen Sie den Namen der R-Funktionen nicht; dann kann Ihnen eine **Suchmaschine** helfen. Versuchen Sie zum Beispiel, den Text `R minimum vector` zu suchen.
- ▶ Quelle: - LABORATORY FOR APPLIED STATISTICS: Intro to R - **Exercises**

# MODULARER AUFBAU VON R

# Wo man Routinen findet

- ▶ Viele Funktionen sind in Basis-R enthalten.
- ▶ Viele spezifische Funktionen sind in zusätzliche Bibliotheken integriert.
- ▶ R kann modular durch sogenannte Pakete oder Bibliotheken erweitert werden.
- ▶ Die wichtigsten Pakete, die auf CRAN gehostet werden (14106 at Mo Mai 06)
- ▶ Weitere Pakete findet man z.B. unter **bioconductor**

# ÜBERSICHT R-PAKETE



# INSTALLATION VON PAKETEN

- ▶ Die Anführungszeichen um den Paketnamen herum sind für den Befehl `install.packages` notwendig.
- ▶ Sie sind optional für den Befehl `library`.
- ▶ Man kann auch `require` anstelle von `library` verwenden.

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

```
library(lme4)
```

# INSTALLATION VON PAKETEN MIT RSTUDIO

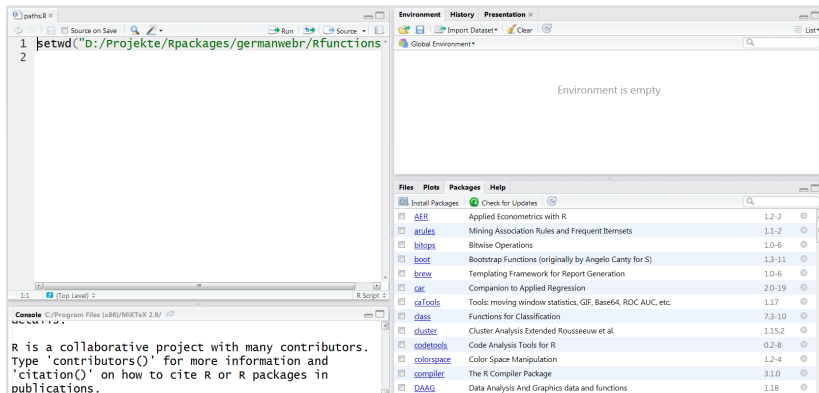


FIGURE 6: Package installation with Rstudio



# BESTEHENDE PAKETE UND INSTALLATION

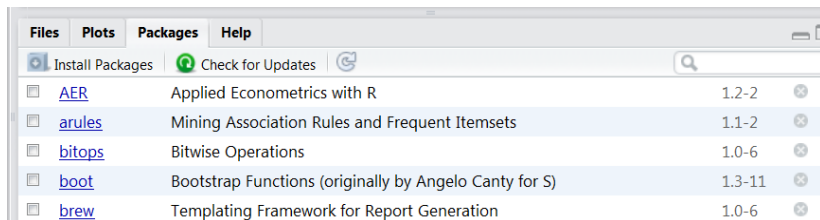


FIGURE 7: Existing packages

# ÜBERSICHT ÜBER VIELE NÜTZLICHE PAKETE:

- ▶ Luhmann - **Table with many useful packages**

## WEITERE INTERESSANTE PAKETE:

- ▶ Paket für Import/Export - `foreign`
- ▶ **sampling-Paket für die Stichprobenziehung**
- ▶ `xtable` Paket zur Integration von LaTeX in R (**xtable Galerie**)
- ▶ `dummies` - **Paket zur Erstellung von Dummies**
- ▶ **Paket `mvtnorm` um eine multivariate Normalverteilung zu erhalten.**
- ▶ **Paket `maptools` um Karten zu erzeugen**

# PAKETE AUS VERSCHIEDENEN QUELLEN INSTALLIEREN

## PAKETE VOM CRAN SERVER INSTALLIEREN

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

## PAKETE VOM BIOCONDUCTOR SERVER INSTALLIEREN

```
source("https://bioconductor.org/biocLite.R")  
biocLite(c("GenomicFeatures", "AnnotationDbi"))
```

## PAKETE VON GITHUB INSTALLIEREN

```
install.packages("devtools")  
library(devtools)  
  
install_github("hadley/ggplot2")
```

# WIE BEKOMME ICH EINEN ÜBERBLICK?

- ▶ Entdecke Pakete, die kürzlich auf den **CRAN** Server hochgeladen wurden
- ▶ Nutze eine Shiny Web-App, die **Pakete anzeigt, die kürzlich von CRAN** heruntergeladen wurden.
- ▶ Werfe einen Blick auf eine **Quick-Liste nützlicher Pakete**
- ▶ . . . ., oder auf eine Liste mit den **besten Paketen für die Datenverarbeitung und -analyse**, . . . .
- ▶ . . . ., oder schaue unter **die 50 meistgenutzten Pakete**

# CRAN TASK VIEWS

- ▶ Bezüglich mancher Themen gibt es einen Überblick über alle wichtigen Pakete - (**CRAN Task Views**)
- ▶ Momentan gibt es 35 Task Views.
- ▶ Alle Pakete einer Task-View können mit folgendem Befehl installiert werden: **command:**

```
install.packages("ctv")  
library("ctv")  
install.views("Bayesian")
```

## CRAN Task Views

[Bayesian](#)

Bayesian Inference

[ChemPhys](#)

Chemometrics and Computational Physics

[ClinicalTrials](#)

Clinical Trial Design, Monitoring, and Analysis

[Cluster](#)

Cluster Analysis & Finite Mixture Models

[DifferentialEquations](#)

Differential Equations

[Distributions](#)

Probability Distributions

[Econometrics](#)

Econometrics

[Environmetrics](#)

Analysis of Ecological and Environmental Data

# A1C ÜBUNG - ZUSÄTZLICHE PAKETE

Geht auf <https://cran.r-project.org/> und sucht nach Paketen...

- ▶ die sich für die deskriptive Datenanalyse eignen.
- ▶ mit denen man fremde Datensätze einlesen kann (z.B. SPSS data)
- ▶ mit denen man Lasso Regressionen rechnen kann
- ▶ mit denen man große Datenmengen bearbeiten kann

# AUFGABE - ZUSATZPAKETE

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

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

# DATENIMPORT



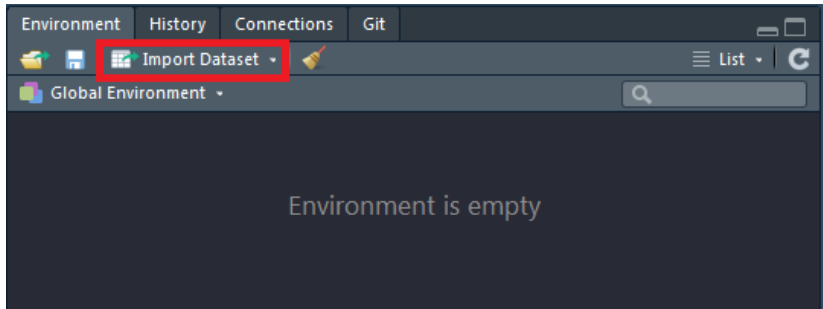
# DATENIMPORT



# DATEN MIT RSTUDIO IMPORTIEREN

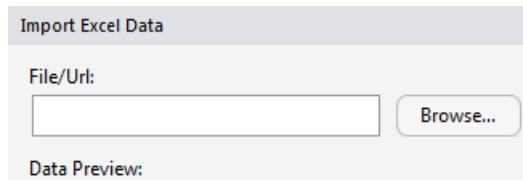
## RSTUDIO FUNKTIONALITÄT UM DATEN ZU IMPORTIEREN

- Environment - Import Dataset - Filetyp auswählen



# WO FINDET MAN DIE DATEN?

## BROWSE BUTTON IN RSTUDIO



The screenshot shows the 'Import Excel Data' dialog box in RStudio. It has a title bar 'Import Excel Data'. Below the title bar, there is a label 'File/Url:' followed by a text input field. To the right of the input field is a button labeled 'Browse...'. Below the input field and button is a label 'Data Preview:'.

## CODE VORSCHAU IN RSTUDIO



The screenshot shows the 'Code Preview' window in RStudio. It has a title bar 'Code Preview:'. Below the title bar, there is a text area containing the following R code:

```
library(readxl)
ee_recode_questionnaire_coded <- read_excel("data/ee_recode_questionnaire_coded.xls")
View(ee_recode_questionnaire_coded)
```

At the bottom right of the window, there are two buttons: 'Import' and 'Cancel'.

## CSV DATEN IMPORTIEREN

- ▶ `read.csv` ist ein Befehl, der im Basispaket verfügbar ist.
- ▶ Excel-Daten können als `.csv` in Excel gespeichert werden.
- ▶ Dann kann `read.csv()` zum Einlesen der Daten verwendet werden.
- ▶ Für Deutsche Daten benötigt man eventuell `read.csv2()` wegen der Komma-Trennung.

```
dat <- read.csv("../data/ZA5666_v1-0-0.csv")
```

Wenn es Deutsche Daten sind:

```
datd <- read.csv2("../data/ZA5666_v1-0-0.csv")
```

# EXCEL-DATENSATZ IMPORTIEREN - MIT XLSX.

## PAKET XLSX

- ▶ Titel: Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files
- ▶ Autoren: Adrian A. Dragulescu, Cole Arendt

```
install.packages("xlsx")
```

```
library("xlsx")  
ab_xlsx <- read.xlsx("../data/ab.xlsx",1)
```

- ▶ Das Paket xlsx benötigt Java - wenn das nicht verfügbar ist, verwenden Sie den Befehl read\_excel aus dem Paket readxl.

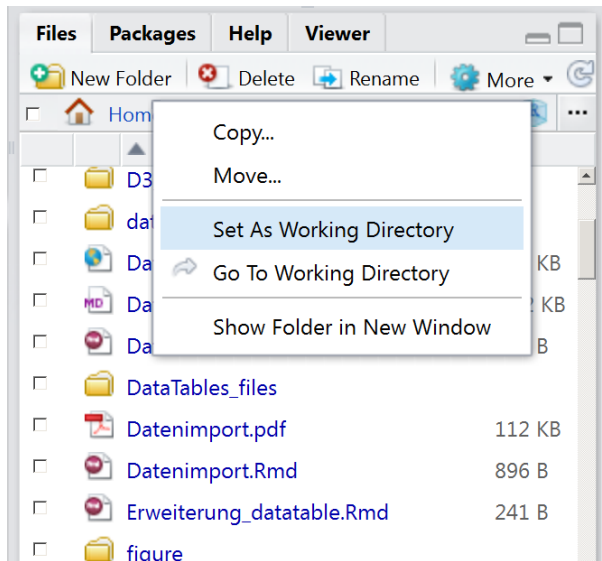
# DAS PAKET READXL

```
install.packages("readxl")
```

- ▶ **readxl hat keine externen Abhängigkeiten**
- ▶ readxl unterstützt sowohl das alte .xls Format als auch das moderne xml-basierte .xlsx Format.

```
library(readxl)  
ab <- read_excel("../data/ab.xlsx")  
head(ab)
```

# ARBEITSSPEICHER



# 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)
```



# IMPORTIEREN VON STATA DATEIEN

- ▶ Mit `read.dta13` können Stata-Dateien ab Version 13 (und höher) importiert werden.

```
library(readstata13)
dat_stata <- read.dta13("../data/ZA5666_v1-0-0_Stata14.dta")
```

## IMPORT VON STATA DATEIEN - ÄLTERE VERSIONEN

```
library(foreign)
dat_stata12 <- read.dta("../data/ZA5666_v1-0-0_Stata12.dta")
```

- ▶ Einführung in den Import mit R (**is.R**)

# DIE BIBLIOTHEK READSTATA13

```
readstata13 {readstata13}
```

R Documentation

## Import Stata Data Files

### Description

Function to read the Stata file format into a data.frame.

### Note

If you catch a bug, please do not sue us, we do not have any money.

### Author(s)

Marvin Garbuszus [jan.garbuszus@ruhr-uni-bochum.de](mailto:jan.garbuszus@ruhr-uni-bochum.de)

Sebastian Jeworutzki [sebastian.jeworutzki@ruhr-uni-bochum.de](mailto:sebastian.jeworutzki@ruhr-uni-bochum.de)

### See Also

[read.dta](#) and [memisc](#) for dta files from Stata Versions < 13

# Die Bibliothek rio

```
install.packages("rio")
```

```
library("rio")  
x <- import("../data/ZA5666_v1-0-0.csv")  
y <- import("../data/ZA5666_v1-0-0_Stata12.dta")  
z <- import("../data/ZA5666_v1-0-0_Stata14.dta")
```

- **rio: Ein Schweizer Offiziersmesser für Data I/O**

# SICH EINEN ERSTEN ÜBERBLICK VERSCHAFFEN

```
View(datf)
```

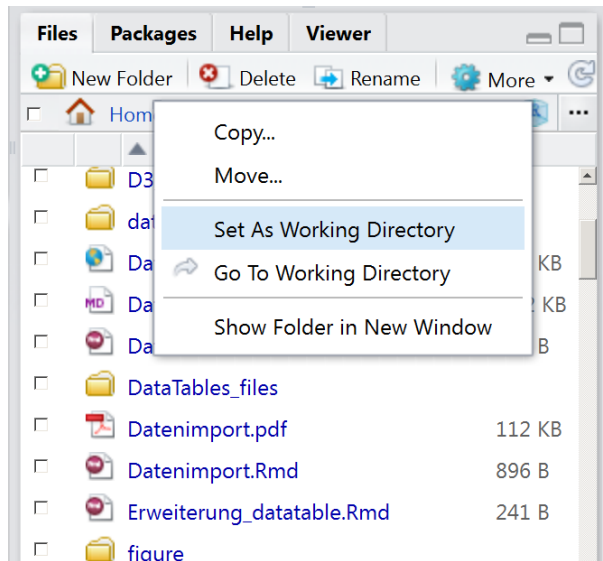


	z000001z Personen ID - Campus File	z000002z Studiennummer des Archivs	z000003z Versionskennung und -datum des Archivs	z000005z doi	a11c019a Zufriedenheit Leben in Wohnort	a11c020a Zufriedenheit
1	198431880	ZA5666	1-0-0 2017-06-20	10.4232/1.12749		1
2	436122330	ZA5666	1-0-0 2017-06-20	10.4232/1.12749		1
3	856844220	ZA5666	1-0-0 2017-06-20	10.4232/1.12749		2
4	117346660	ZA5666	1-0-0 2017-06-20	10.4232/1.12749		1
5	943433330	ZA5666	1-0-0 2017-06-20	10.4232/1.12749		1

FIGURE 8: Rstudio Viewer

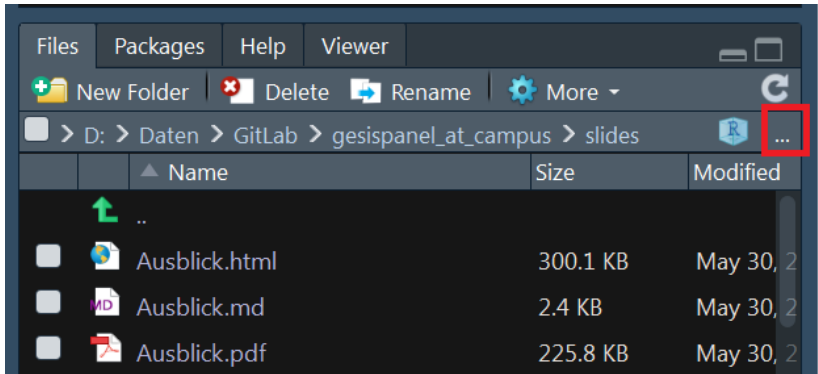
- ▶ Das gleiche kann man mit RStudio erreichen, wenn man auf das Datensatzsymbol im Umgebungsmenü klickt.

# DAS ARBEITSVERZEICHNIS



...

- ▶ Wenn sich die Daten auf einem anderen Laufwerk in Windows befinden



## DAS ARBEITSVERZEICHNIS II

Auf diese Weise können Sie herausfinden, in welchem Verzeichnis Sie sich gerade befinden.

```
getwd()
```

So können Sie das Arbeitsverzeichnis ändern:

Sie legen ein Objekt an, in dem Sie den Pfad speichern:

```
main.path <- "C:/" # Example for Windows  
main.path <- "/users/Name/" # Example for Mac  
main.path <- "/home/user/" # Example for Linux
```

Und dann ändert man den Pfad mit `setwd()`.

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

Unter Windows ist es wichtig, Slashes anstelle von Backslashes zu

# ARBEITSVERZEICHNIS WECHSELN

- ▶ Man kann auch die Tabulatortaste verwenden, um die automatische Vervollständigung zu erhalten.

```
getwd()
```

```
## [1] "D:/github/IntroR/2019/slides"
```

```
setwd("../")  
getwd()
```

```
## [1] "D:/github/IntroR/2019"
```



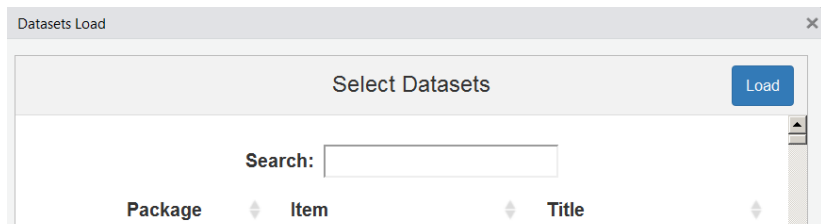
# INGEBAUTE DATENSÄTZE

- ▶ Häufig wird ein Beispieldatensatz zur Verfügung gestellt, um die Funktionalität eines Pakets zu zeigen.
- ▶ Diese Datensätze können mit dem Befehl `data` geladen werden.

```
data(iris)
```

- ▶ Es gibt auch ein **RStudio-Add-In**, das hilft, einen Datensatz zu finden.

```
install.packages("datasets.load")
```



# Daten einfügen

## ► RStudio Addin um Daten einzufügen

```
devtools::install_github("lbusett/insert_table")
```

Insert Table Add-In

Cancel Select output format and edit the Table if you wish so Done

Select Table Name Select Output Format

my\_tbl None

Edit Table or cut and paste from spreadsheet

- \* The first row will be used as column names.
- \* Right click to add more lines or columns

☒ Use first row as column names. (If unchecked, 'Col\_1', 'Col\_2', etc. are used)

1	a	v		c
---	---	---	--	---

# ÜBUNG - IMPORTIEREN VON DATEN

- ▶ Importiere die Daten des österreichischen Mikrozensus und verschaffe Dir einen ersten Überblick über die Daten.

# DATENEXPORT

# 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`)

# BEISPIELDATENSATZ ERZEUGEN

```
A <- c(1,2,3,4)
B <- c("A","B","C","D")

mydata <- data.frame(A,B)
```

# ÜBERBLICK DATEN IMPORT/EXPORT

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



# DATEN IN EXCEL FORMAT ABSPEICHERN

```
write.csv(mydata, file="mydata.csv")
```

```
library(xlsx)  
write.xlsx(mydata, file="mydata.xlsx")
```

# DATEN IN STATA FORMAT ABSPEICHERN

```
library(foreign)  
write.dta(mydata, file="mydata.dta")
```

## AUCH ZUM EXPORT EIGNET SICH DAS `rio` PAKET

```
library("rio")  
  
export(mtcars, "mtcars.csv")  
export(mtcars, "mtcars.rds")  
export(mtcars, "mtcars.dta")
```

# LINKS EXPORT

- ▶ Quick R für das Exportieren von Daten:
- ▶ Hilfe zum Export auf dem CRAN Server

# INDIZIEREN

# AUFGABE IMPORT VON DATEN

- ▶ Gehe auf das Portal für **offene Daten der Stadt Frankfurt**



- ▶ Importiere den Datensatz für Beschäftigte mit einer geeigneten Funktion.

# DIE DATEN EINLESEN

```
link <- "http://offenedaten.frankfurt.de/dataset/50968551-h  
dat <- rio::import(link)
```

# EINEN ÜBERBLICK ÜBER DIE DATEN BEKOMMEN

```
head(dat)
```

Codes	Stadtteil	Wirtschaft Betriebe 2009	Wirtschaft Betriebe 2010
1	Altstadt		708
2	Innenstadt		2733
3	Bahnhofsviertel		1267
4	Westend-Süd		3190
5	Westend-Nord		747
6	Nordend-West		2325



# EINEN ERSTEN EINDRUCK DER DATEN BEKOMMEN

```
library(dplyr)
glimpse(dat)
```

```
## Observations: 46
## Variables: 54
## $ Codes
## $ Stadtteil
## $ `Wirtschaft Betriebe 2009`
## $ `Wirtschaft Betriebe 2010`
## $ `Wirtschaft Beschäftigte in Betrieben 2009`
## $ `Wirtschaft Beschäftigte in Betrieben 2010`
## $ `Wirtschaft Beschäftigte in Betrieben im Produzierende
## $ `Wirtschaft Beschäftigte in Betrieben im Produzierende
## $ `Wirtschaft Beschäftigte in Betrieben im Dienstleistun
## $ `Wirtschaft Beschäftigte in Betrieben im Dienstleistun
## $ `Wirtschaft Coverbeanmeldungen 2011`
```

# INDIZIEREN

## NUR DIE ERSTE SPALTE

```
dat[,2]
```

```
## [1] "Altstadt"      "Innenstadt"     "Bahnhofsvierte  
## [5] "Westend-Nord"  "Nordend-West"
```

## GLEICHES ERGEBNIS

```
dat$Stadtteil
```

```
## [1] "Altstadt"      "Innenstadt"     "Bahnhofsvierte  
## [5] "Westend-Nord"  "Nordend-West"
```

# EINE BEOBACHTUNG ANSCHAUEN

## NUR DIE ERSTE REIHE

```
dat[1,]
```

```
## Codes Stadtteil Wirtschaft Betriebe 2009 Wirtschaft B
## 1      1 Altstadt                      708
## Wirtschaft Beschäftigte in Betrieben 2009
## 1                                           11111
## Wirtschaft Beschäftigte in Betrieben 2010
## 1                                           21850
## Wirtschaft Beschäftigte in Betrieben im Produzierenden
## 1
## Wirtschaft Beschäftigte in Betrieben im Produzierenden
## 1
## Wirtschaft Beschäftigte in Betrieben im Dienstleistung
## 1
```

# DIE DATEN ZUSAMMENFASSEN

```
summary(dat[3])
```

```
##  Wirtschaft Betriebe  2009  
##  Min.      : 118.0  
##  1st Qu.: 429.8  
##  Median : 695.0  
##  Mean    : 1810.5  
##  3rd Qu.: 1090.0  
##  Max.    :42126.0
```

# EINE AUSWAHL TREFFEN

## WIRTSCHAFT BETRIEBE 2009 - MEHR ALS 3000

```
dat$Stadtteil[dat[,3]>2000]
```

```
## [1] "Innenstadt"          "Westend-Süd"          "Nordend-West"
## [4] "Ostend"              "Bockenheim"           "Sachsenhausen"
## [7] "Frankfurt am Main"
```

## WIRTSCHAFT BETRIEBE 2010 - WENIGER ALS 1000

```
dat$Stadtteil[dat[,4]<1000]
```

```
## [1] "Altstadt"          "Westend-Nord"          "Gutleutviertel"
## [4] "Oberrad"          "Niederrad"            "Schwanheim"
## [7] "Griesheim"        "Rödelheim"            "Hausen"
## [10] "Praunheim"        "Heddernheim"          "Niederursel"
## [13] "Ginnheim"         "Dornbusch"            "Eschersheim"
## [16] "Sachsenhausen"    "Bockenheim"           "Nordend-West"
## [19] "Ostend"            "Innenstadt"           "Frankfurt am Main"
```

# MEHRERE BEDINGUNGEN MITEINANDER VERKNÜPFEN

## SPALTE 23 - WIRTSCHAFT GEWERBEABMELDUNGEN 2011

```
dat$Stadtteil[dat[,4]<1000 & dat[,23] > 300]
```

```
## [1] "Gutleutviertel" "Griesheim"      "Fechenheim"      "
```

# DATENANALYSE

# STREUUNGSMASSE

In Basis R sind die wichtigsten Streuungsmaße enthalten:

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

```
ab <- rnorm(100)
var(ab)
```

```
## [1] 1.347541
```

```
sd(ab)
```

```
## [1] 1.160836
```

```
range(ab)
```



# EXTREMWERTE

```
min(ab)
```

```
## [1] -2.445036
```

```
max(ab)
```

```
## [1] 3.071773
```

# FEHLENDE WERTE

- Sind NAs vorhanden muss dies der Funktion mitgeteilt werden

```
ab[10] <- NA  
var(ab)
```

```
## [1] NA
```

Bei fehlenden Werten muss ein weiteres Argument mitgegeben werden:

```
var(ab, na.rm=T)
```

```
## [1] 1.35953
```

# HÄUFIGKEITEN UND GRUPPIERTE KENNWERTE

- ▶ 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`

```
x <- sample(1:10,100,replace=T)
table(x)
```

```
## x
##  1  2  3  4  5  6  7  8  9 10
## 12 10  6  8 12  6 14 10  9 13
```

## TABELLIEREN - WEITERES BEISPIEL

```
musician <- sample(c("yes", "no"), 100, replace=T)
```

```
?table
```

```
table(x)
```

```
## x
```

```
##  1  2  3  4  5  6  7  8  9 10
```

```
## 12 10  6  8 12  6 14 10  9 13
```

```
table(x, musician)
```

```
##      musician
```

```
## x      no yes
```

```
##  1      6  6
```

```
##  2      5  5
```

```
##  3      2  4
```

# EINE WEITERE TABELLE

```
data(esoph)
table(esoph$agegp)
```

```
##
## 25-34 35-44 45-54 55-64 65-74 75+
##    15    15    16    16    15    11
```

# 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)
```

```
##
##           0-39g/day 40-79 80-119 120+
## 25-34             4      4      3      4
## 35-44             4      4      4      3
## 45-54             4      4      4      4
## 55-64             4      4      4      4
## 65-74             4      3      4      4
## 75+              3      4      2      2
```

# DIE FUNKTION `PROP.TABLE`

```
?prop.table
```

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

```
##
##           0-39g/day      40-79      80-119      120+
## 25-34 0.2666667 0.2666667 0.2000000 0.2666667
## 35-44 0.2666667 0.2666667 0.2666667 0.2000000
## 45-54 0.2500000 0.2500000 0.2500000 0.2500000
## 55-64 0.2500000 0.2500000 0.2500000 0.2500000
## 65-74 0.2666667 0.2000000 0.2666667 0.2666667
## 75+   0.2727273 0.3636364 0.1818182 0.1818182
```

# DIE AGGREGATE FUNKTION

- ▶ Mit der `aggregate()` Funktion können Kennwerte für Untergruppen erstellt werden
- ▶ `aggregate(x,by,FUN)` müssen mindestens drei Argumente übergeben werden:

```
aggregate(state.x77,by=list(state.region),mean)
```

```
##           Group.1 Population    Income Illiteracy Life Exp
## 1      Northeast   5495.111  4570.222    1.000000  71.26444
## 2           South   4208.125  4011.938    1.737500  69.70625
## 3 North Central   4803.000  4611.083    0.700000  71.76667
## 4           West   2915.308  4702.615    1.023077  71.23462
##           Frost      Area
## 1 132.7778  18141.00
## 2  64.6250  54605.12
## 3 138.8333  62652.00
```



# BEISPIELDATENSATZ - APPLY FUNKTION

```
ApplyDat <- cbind(1:4,runif(4),rnorm(4))
```

```
apply(ApplyDat,1,mean)
```

```
## [1] 0.6703499 0.5683878 1.0292335 1.1017736
```

```
apply(ApplyDat,2,mean)
```

```
## [1] 2.5000000 0.7541102 -0.7268016
```

# DIE FUNKTION APPLY

```
apply(ApplyDat,1,var)
```

```
## [1] 0.2260003 2.4823473 3.4941628 7.0173564
```

```
apply(ApplyDat,1,sd)
```

```
## [1] 0.4753949 1.5755467 1.8692680 2.6490293
```

```
apply(ApplyDat,1,range)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.1253954 -1.119639 -0.7185291 -1.194434
## [2,] 1.0000000  2.000000  3.0000000  4.000000
```

```
apply(ApplyDat,1,length)
```

```
## [1] 3 3 3 3
```

# ARGUMENTE DER FUNKTION APPLY

- ▶ Für `margin=1` die Funktion `mean` auf die Reihen angewendet,
- ▶ Für `margin=2` die Funktion `mean` auf die Spalten angewendet,
- ▶ Anstatt `mean` können auch andere Funktionen wie `var`, `sd` oder `length` verwendet werden.

# DIE FUNKTION TAPPLY

```
ApplyDat <- data.frame(Income=rnorm(5,1400,200),  
                        Sex=sample(c(1,2),5,replace=T))
```

- ▶ Auch andere Funktionen können eingesetzt werden.... - Auch selbst programmierte Funktionen
- ▶ Im Beispiel wird die einfachste eigene Funktion angewendet.

```
ApplyDat
```

##		Income	Sex
##	1	1278.451	2
##	2	1453.014	1
##	3	1668.245	1
##	4	1306.011	2
##	5	1235.319	2

## BEISPIEL FUNKTION TAPPLY

```
tapply(ApplyDat$Income, ApplyDat$Sex, mean)
```

```
##           1           2  
## 1560.629 1273.260
```

```
tapply(ApplyDat$Income,  
       ApplyDat$Sex, function(x)x)
```

```
## $`1`  
## [1] 1453.014 1668.245  
##  
## $`2`  
## [1] 1278.451 1306.011 1235.319
```

# LINKS DATENANALYSE

- ▶ Die Benutzung von `apply`, `tapply`, etc. (Artikel bei R-bloggers)
- ▶ Quick-R zu deskriptiver Statistik
- ▶ Quick-R zur Funktion `aggregate`