
R IN DEN BIOWISSENSCHAFTEN

Offene Software zur statistischen Auswertung von Experimentaldaten



Einführung: Was ist R?

- Eine Programmiersprache (imperativ, prozedural, objektorientiert, funktionale Anleihen,...)
- Ein „Framework“ mit Bibliotheken, Mailinglisten, „Standards“, Blogs, Entwicklern, Nutzerkonferenzen, Artikeln in wissenschaftlichen Magazinen
- Hauptziel: Statistische Auswertung, Data Mining
 - Schnell zu implementieren
 - Viele Methoden und Anwendungsgebiete
 - Nicht zu verwenden bei ...
- Historisch: GNU-Implementierung von S, inspiriert von Scheme
 - (i.e., old'n'rusty)
- Freie Software, freie Bibliotheken

Einführung: Wo bekommt man das her?

- R „base system“ für Linux, MacOSX, Windows oder Source von <http://www.r-project.org/>
 - (UNIXer: ./configure; make; make install)
- Mein Tipp: Zuerst mit dem base system anfangen
 - `apt-get install r-base r-base-core r-recommended`

Einführung: Wo bekomme ich Hilfe?

- Google (d'oh)
- Auf R spezialisierte Suchmaschine: <http://www.rseek.org/>
- Reference card: cran.r-project.org/doc/contrib/Short-refcard.pdf
- Fragen und Antworten: <http://stats.stackexchange.com/>
- Howto für verschiedene Statistische Methoden: Quick-R
<http://www.statmethods.net/>
- Blogs: <http://www.r-bloggers.com/>
- Tutorial: <http://www.r-tutor.com/r-introduction>

Erste Schritte

Live

- R ist eine interaktive Konsolenanwendung
- Alternativ können genau so gut R-Skripte direkt ausgeführt werden
- Konsolennutzung, Hello World:

Programmstart → `tibu@cheshire:~$ R`

Begrüßung → `R version 3.1.0 (2014-04-10) -- "Spring Dance"`
`[...]`

Eingabe → `R> print("Hello World")`

Ausgabe → `[1] "Hello World"`

Erste Schritte, Fortsetzung

Live

- Der wichtigste Befehl:

```
R> ?kommando
```

- Z.B.:

```
R> ?print
```

```
tibu@cheshire:~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
print package:base R Documentation  
Print Values  
Description:  
  
  'print' prints its argument and returns it invisibly (via  
  'invisible(x)'). It is a generic function which means that new  
  printing methods can be easily added for new 'class'es.  
  
Usage:  
  
  print(x, ...)  
  
  ## S3 method for class 'factor'  
  print(x, quote = FALSE, max.levels = NULL,  
        width = getOption("width"), ...)  
  
  ## S3 method for class 'table'  
  print(x, digits = getOption("digits"), quote = FALSE,  
        na.print = "", zero.print = "0", justify = "none", ...)  
  
  ## S3 method for class 'function'  
:  
:
```

Erste Schritte, Fortsetzung

Live

■ Auch nützlich:

```
R> help.search("test")  
R> help(package="MASS")
```


Erste Schritte: R-Hilfe

Live

- In jeder R-Hilfeseite finden Sie:
 - Description – eine Einführung
 - Usage – Syntax, Parameter
 - Arguments – Beschreibung der Parameter
 - Details – Details der Implementierung, Best Practice, etc.
 - Value - Rückgabewerte
 - Examples - Beispiele
- Manchmal/oft:
 - References - Quellenliteratur
 - „See Also“ – Vorschläge für andere Hilfeseiten

Erste Schritte: Elementare Datentypen

Live

```
R> # Variablenzuweisung
R> # Integer
R> x <- 1
R> # Numeric
R> y <- 2.0
R> # Character
R> hello <- "World"
R> # Logical
R> foo <- TRUE
R> print(x)
[1] 1
R> print(y)
[1] 2
R> print(hello)
[1] "World"
R> print(foo)
[1] TRUE
```

Erste Schritte: Vektoren

Live

```
R> # Vektoren
R> # Integers
R> vec1 <- c(1,2,15,8)
R> print(vec1)
[1] 1 2 15 8
R> # Logical
R> vec2 <- c(TRUE,FALSE,TRUE,TRUE)
R> print(vec2)
[1] TRUE FALSE TRUE TRUE
```

Erste Schritte: Operationen auf Vektoren

Live

```
R> # vec1 <- c(1,2,15,8)
R> length(vec1)
[1] 4
R> vec1[2]
[1] 2
R> vec1[-2]
[1] 1 15 8
R> vec1[vec1 > 2]
[1] 15 8
R> sum(vec1)
[1] 26
R> max(vec1)
[1] 15
R> min(vec1)
[1] 1
R> sort(vec1)
[1] 1 2 8 15
R> vec1+1
[1] 2 3 16 9
```

Erste Schritte: Listen

```
R> # Lists
R> x <- list(a=23,b="Text",c=c(1,2,3))
R> print(x)
$a
[1] 23

$b
[1] "Text"

$c
[1] 1 2 3
R> print(x$c)
[1] 1 2 3
R> lapply(x,is.numeric)
$a
[1] TRUE

$b
[1] FALSE

$c
[1] TRUE
```

Live

Erste Schritte: Data Frames

Live

```
R> patients <- read.csv("patients.csv")
R> head(patients)
```

	Pat.No	Sex	Copd	smoker	BLOOD_MONOCYTES_ABS	BLOOD_LYMPHOCYTE_ABS	BLOOD_NEUTROPHILS_ABS
1	1	M	Y	Y	0.87	3.26	5.39
2	26	F	Y	Y	0.43	1.87	4.60
3	2	M	Y	Y	0.32	1.85	3.88
4	159	F	Y	Y	0.41	1.08	3.34
5	69	M	N	N	0.45	1.25	3.31
6	65	F	N	N	0.55	1.48	4.58
	SPUTUM_NEUTROPHILS_ABS						
1	1.9587						
2	21.9623						
3	2.2881						
4	0.3055						
5	0.2324						
6	0.3957						

```
R> str(patients)
```

```
'data.frame':  40 obs. of  8 variables:
 $ Pat.No      : int   1 26 2 159 69 65 126 133 143 10 ...
 $ Sex         : Factor w/ 2 levels "F","M": 2 1 2 1 2 1 2 1 1 2 ...
 $ Copd        : Factor w/ 2 levels "N","Y": 2 2 2 2 1 1 1 1 1 2 ...
 $ smoker      : Factor w/ 2 levels "N","Y": 2 2 2 2 1 1 2 2 2 2 ...
 $ BLOOD_MONOCYTES_ABS : num  0.87 0.43 0.32 0.41 0.45 0.55 0.54 0.61 0.51 0.77 ...
 $ BLOOD_LYMPHOCYTE_ABS : num  3.26 1.87 1.85 1.08 1.25 1.48 1.83 1.94 2.89 2.21 ...
 $ BLOOD_NEUTROPHILS_ABS : num  5.39 4.6 3.88 3.34 3.31 4.58 2.75 7.24 5.27 6.34 ...
 $ SPUTUM_NEUTROPHILS_ABS: num  1.959 21.962 2.288 0.305 0.232 ...
```

Erste Schritte: Data Frames

Live

- Andere Funktionen zum einlesen von Data Frames
 - read.table
 - read.delim
 - read.xlsx (xlsx2)
 - read.spss
 - h5read (hdf5)
- Handgestrickt mit Funktion data.frame()

```
R> x <- data.frame(  
+ A = 1:3,  
+ B = c(1,3,5),  
+ C = rep(5,3)  
+ )  
R> print(x)  
  A B C  
1 1 1 5  
2 2 3 5  
3 3 5 5
```

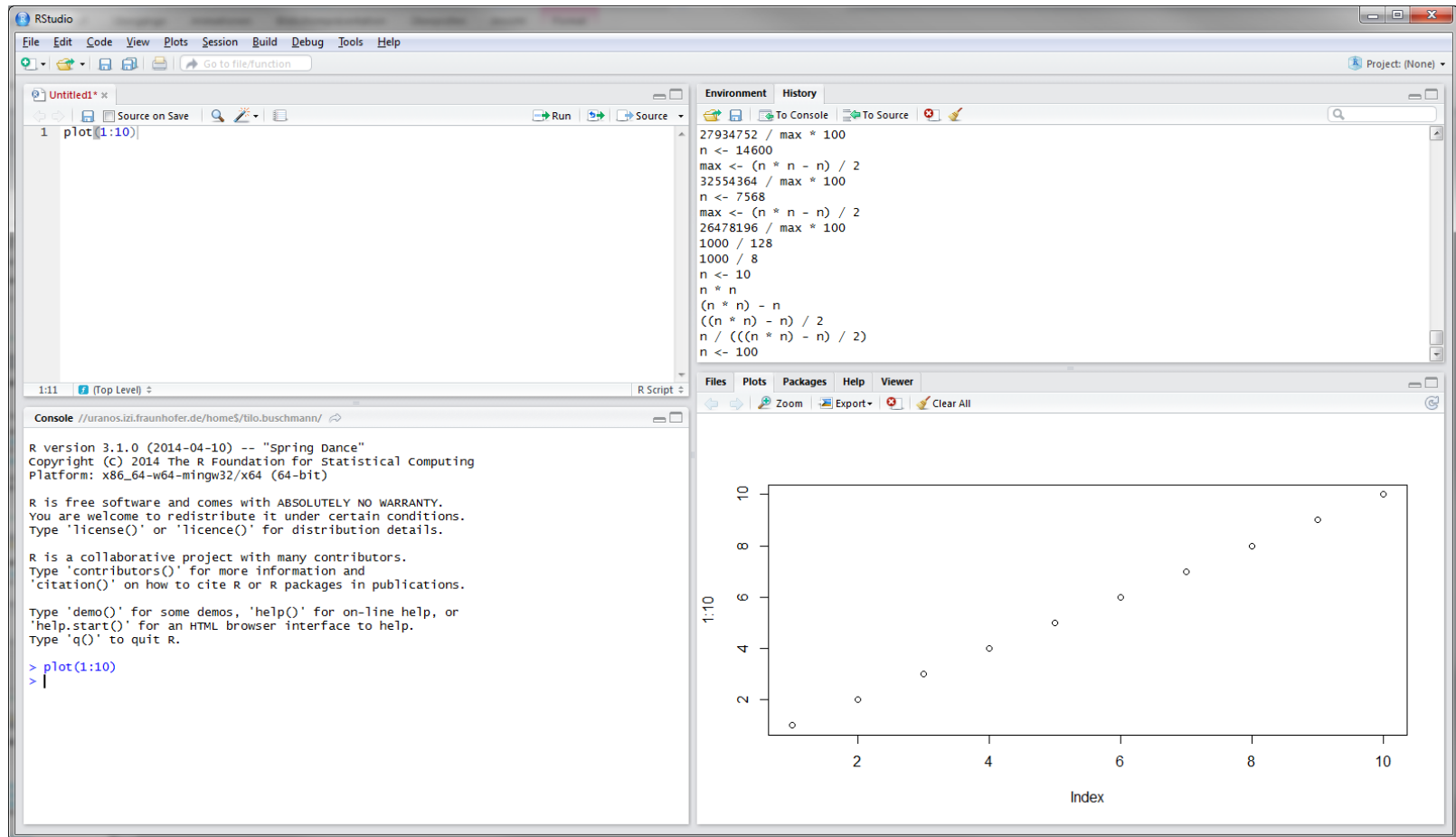
Erste Schritte: Arbeiten mit Data Frames

Live

```
R> dim(patients)
[1] 40  8
R> patients$SPUTUM_NEUTROPHILS_ABS
 [1]  1.95868 21.96229  2.28812  0.30546  0.23238  0.39567  3.07390  1.17182  0.99112  2.2
[11]  1.45405 12.34215 17.41171  1.31371  0.32239  0.35090  0.40708  0.26016  0.67031  4.246
[21] 28.09275  2.88190  0.16614  0.57972  4.50639  1.23610  0.28004  5.28773  2.47152 10.35878
[31] 16.33827  0.04892  1.34417  0.79358  0.36973  0.29812  5.85168  4.53076  6.02017  4.64178
R> mean(patients$SPUTUM_NEUTROPHILS_ABS)
[1] 4.238
R> colnames(patients)
[1] "Pat.No"          "Sex"              "Coptd"
[4] "smoker"          "BLOOD_MONOCYTES_ABS" "BLOOD_LYMPHOCYTE_ABS"
[7] "BLOOD_NEUTROPHILS_ABS" "SPUTUM_NEUTROPHILS_ABS"
R> summary(patients)
      Pat.No      Sex      Coptd      smoker BLOOD_MONOCYTES_ABS BLOOD_LYMPHOCYTE_ABS
Min.   :  1.0   F:16   N:20   N:20   Min.   :0.280   Min.   :1.08
1st Qu.: 20.8   M:24   Y:20   Y:20   1st Qu.:0.410   1st Qu.:1.43
Median : 67.5                Median :0.495   Median :1.86
Mean   : 80.5                Mean   :0.532   Mean   :1.90
3rd Qu.:143.5              3rd Qu.:0.603   3rd Qu.:2.25
Max.   :167.0              Max.   :0.900   Max.   :3.26
BLOOD_NEUTROPHILS_ABS SPUTUM_NEUTROPHILS_ABS
Min.   :2.72          Min.   : 0.049
1st Qu.:3.10          1st Qu.: 0.389
Median :3.30          Median : 1.399
Mean   :3.86          Mean   : 4.238
3rd Qu.:4.61          3rd Qu.: 4.559
Max.   :7.24          Max.   :28.093
```

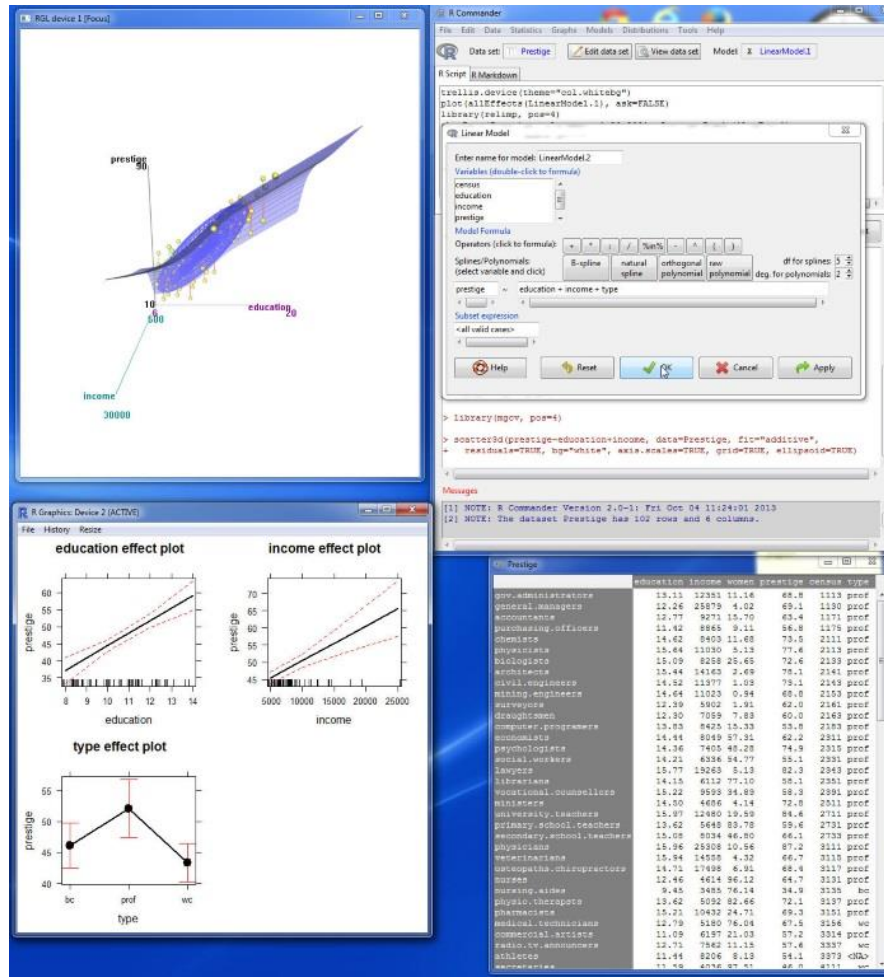

Alternativen zur Kommandozeile

- RStudio: Etwas komfortablere IDE von <http://www.rstudio.com/>
- Plus: Server-Modell ebenfalls möglich.



Alternativen zur Kommandozeile

- Rcmdr: Umfangreiche GUI/IDE von <http://www.rcommander.com/>

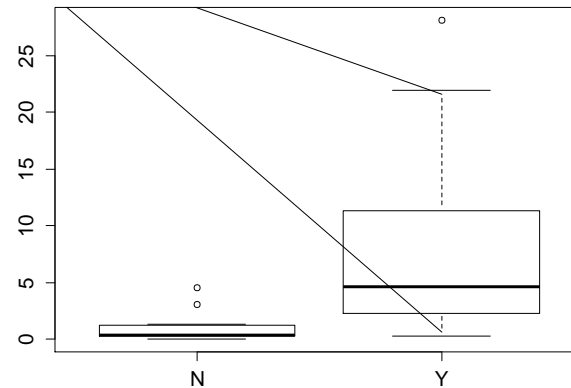
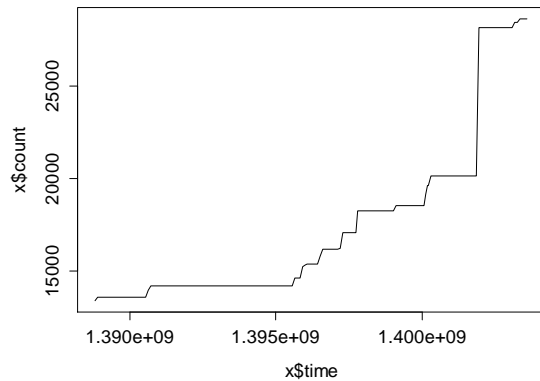
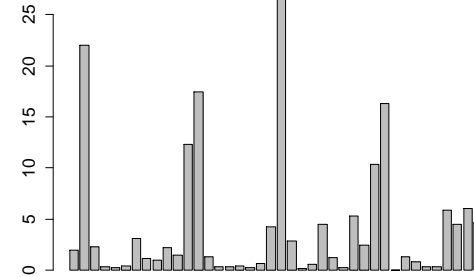
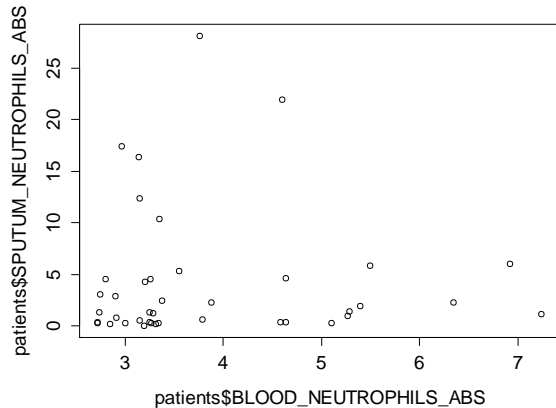


Nächste Schritte: Was kann man mit R alles machen?

- Alles, was mit Daten zu tun hat.
- Grundumfang hat bereits alles, was für Schul- und Universitätsstatistik nötig ist
- Pakete erweitern es ins unendliche
- An Basisoperationen ist alles da
 - Arithmetische Operationen (+, -, *, /, sin, cos, etc.pp.)
 - Vektoroperationen (min, max, which.min, which.max, range, sum, sd, var, mean, median, sort, rev)
 - Matrixoperationen (%*%, solve, rowsum, colsum, rowMeans, colMeans, t, diag)

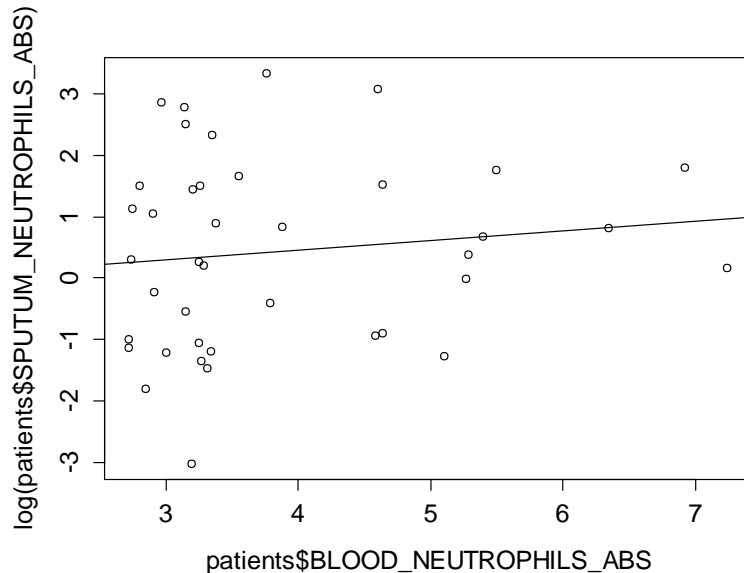
Nächste Schritte: Deskriptive Statistik

■ Plots (Punkte, Linien, Boxplots, Barplots, Kurven,...)



Nächste Schritte: Induktive Statistik

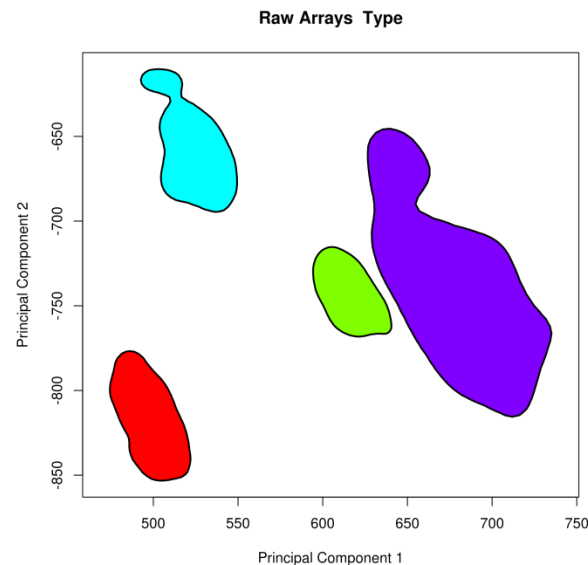
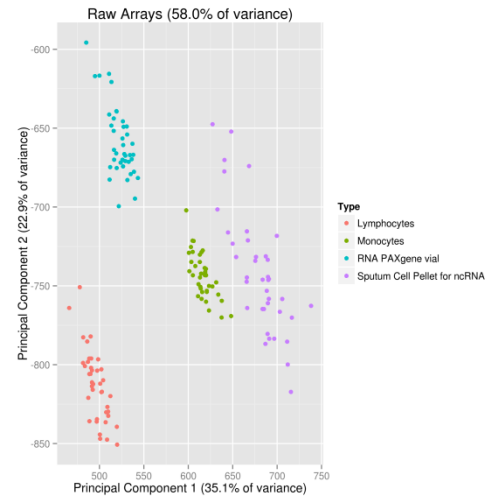
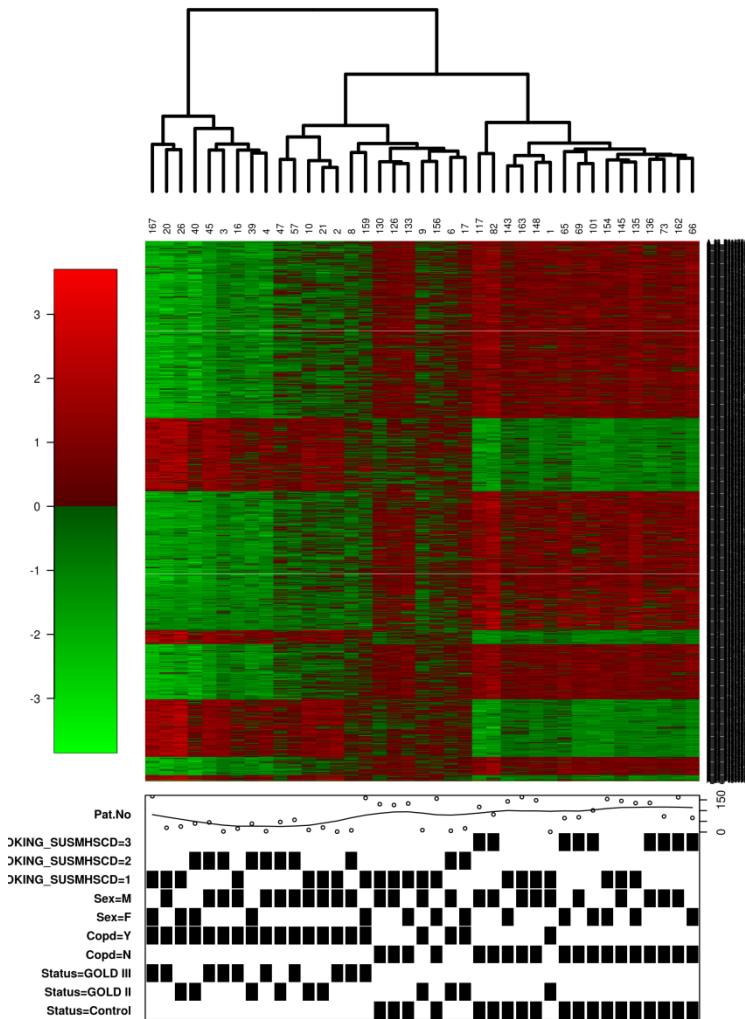
- Einfache univariate Tests (student's t test, chi²-test, etc.)
- Modelle mehrere Variablen (ANOVA, lineare Modelle, generalisierte lineare Modelle)



- Nicht-parametrische Statistik (GAM, Wilcoxon, Mann–Whitney U)
- Multivariate Statistik (MANOVA)

Nächste Schritte: Machine Learning

■ Unbegrenzte Möglichkeiten (SVM, Clustering, etc. pp.)



Nächste Schritte: Funktionalität erweitern

- Zusatzfunktionalität in Form von „Packages“ die in einer Bibliothek hinterlegt werden
- Sammlung von zusammengehörigen Funktionen, Klassen, Daten
- 2 (+1)Hauptquellen:
 - CRAN, in R eingebaut: <http://cran.r-project.org/>
 - Bioconductor: <http://www.bioconductor.org/>
 - Alpha / In-Development: <https://r-forge.r-project.org/>

Nächste Schritte: Funktionalität erweitern

■ Installation von CRAN-Paketen:

```
R> install.packages("ggplot2")
Installing package into '/home/tibu/R/x86_64-pc-linux-gnu-library/3.1'
[...]
* DONE (ggplot2)
```

■ Updates:

```
R> update.packages()
R> update.packages("ggplot2")
```

■ Liste:

```
R> installed.packages()
```

■ Installation von Bioconductor-Paketen:

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

■ Laden von Paketen:

```
library("ggplot2")
library("GenomicFeatures")
```


Nächste Schritte: Alternative: Task Views

■ Installation von CRAN Task Views:

```
R> install.packages("ctv")  
R> library("ctv")  
R> install.views("Cluster")
```

■ Andere Beispielsviews:

- Bayesian
- Econometrics
- Genetics

■ Update

```
R> update.views()
```

Live

Nächste Schritte: Funktionalität erweitern

- Größter Unterschied zu jar, CPAN, CTAN, etc: Wissenschaftliche Veröffentlichungen

BMC Bioinformatics



Methodology article

Open Access

A unified approach to false discovery rate estimation

Korbinian Strimmer

```
fdrtool                package:fdrtool                R Documentation
Estimate (Local) False Discovery Rates For Diverse Test Statistics
Description:
'fdrtool' takes a vector of z-scores (or of correlations,
p-values, or t-statistics), and estimates for each case both the
tail area-based Fdr as well as the density-based fdr (=q-value
resp. local false discovery rate). The parameters of the null
```

```
R> citation("fdrtool")
```

To cite package 'fdrtool' in publications use:

Bernd Klaus and Korbinian Strimmer. (2014). fdrtool: Estimation of (Local) False Discovery Rates and Higher Criticism. R package version 1.2.12. <http://CRAN.R-project.org/package=fdrtool>

Nächste Schritte: Beliebte Pakete

- ggplot2 – Graphiken für Dunkle Magier
- Rcpp – C/C++-Integration
- Hmisc - Harrell Miscellaneous
- zoo – Zeitserien
- plyr – „splitting, applying, combining data“
- reshape2 – Daten umformen (normalisieren, denormalisieren, etc)
- stringr – einfacher mit Strings arbeiten

Erste Statistik

- Ablauf einer statistischen Erhebung:
 - 1. Hypothesen überlegen
 - 2. Überdenken Sie ihre Variablen
 - Welche Skala? (Nominalskala, Ordinalskala, Intervalskala/Verhältnisskala)
 - Welche Variable ist unabhängig, welche abhängig?
 - Welche Verteilung? (diskret, kontinuierlich, normalverteilt, binomial, etc.)
 - 3. Legen sie sich ihre statistischen Hilfsmittel zurecht: Welche Tests? Mit welchen Daten? Was erwarte ich?
 - 4. Erheben Sie Ihre Daten/ Führen Sie ihr Experiment durch.
 - 5. Beginnen Sie ihre Auswertung mit deskriptiver Statistik
 - 6. Danach beginnen sie mit explorativer Statistik oder Hypothesentesten (Sie hatten vorher Hypothesen, nicht wahr?)

Erste Statistik: Deskriptive Statistik

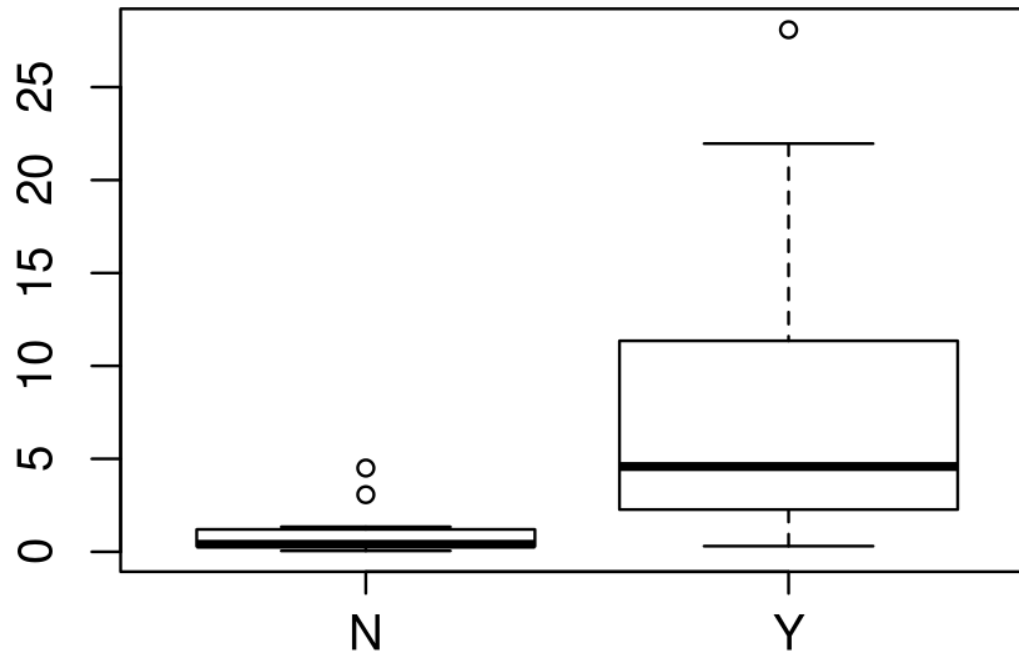
Live

```
R> mean(patients$SPUTUM_NEUTROPHILS_ABS)
[1] 4.238
R> var(patients$SPUTUM_NEUTROPHILS_ABS)
[1] 41.33
R> mean(patients[patients$Copd == "Y", "SPUTUM_NEUTROPHILS_ABS"])
[1] 7.568
R> mean(patients[patients$Copd == "N", "SPUTUM_NEUTROPHILS_ABS"])
[1] 0.9071
R> var(patients[patients$Copd == "Y", "SPUTUM_NEUTROPHILS_ABS"])
[1] 60.29
R> var(patients[patients$Copd == "N", "SPUTUM_NEUTROPHILS_ABS"])
[1] 1.194
```

Erste Statistik: Deskriptive Statistik

Live

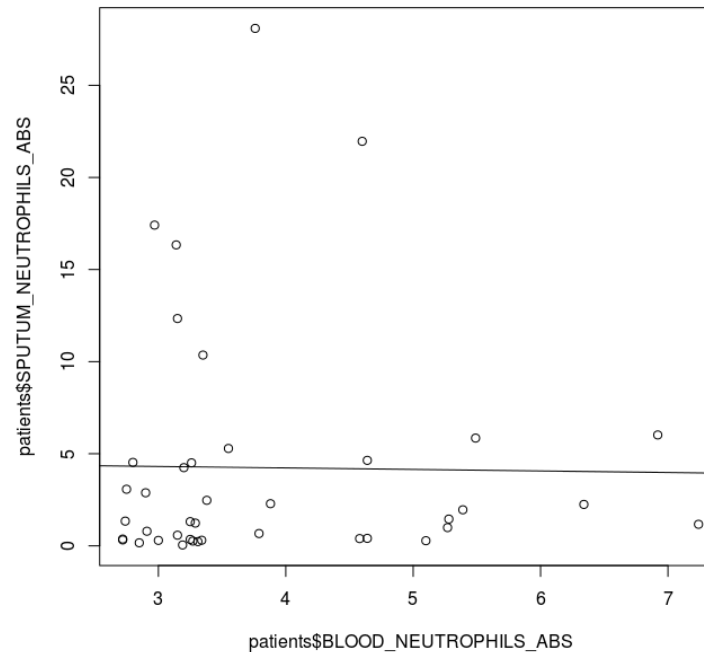
```
R> boxplot(SPUTUM_NEUTROPHILS_ABS ~ Copd, data=patients)
```



Erste Statistik: Deskriptive Statistik

Live

```
R> cor(patients$BLOOD_NEUTROPHILS_ABS,patients$SPUTUM_NEUTROPHILS_ABS)
[1] -0.0152
R> plot(patients$BLOOD_NEUTROPHILS_ABS,patients$SPUTUM_NEUTROPHILS_ABS)
R> reg1 <- lm(SPUTUM_NEUTROPHILS_ABS ~ BLOOD_NEUTROPHILS_ABS, data=patients)
R> abline(reg1)
```



Erste Statistik: Skripting

Live

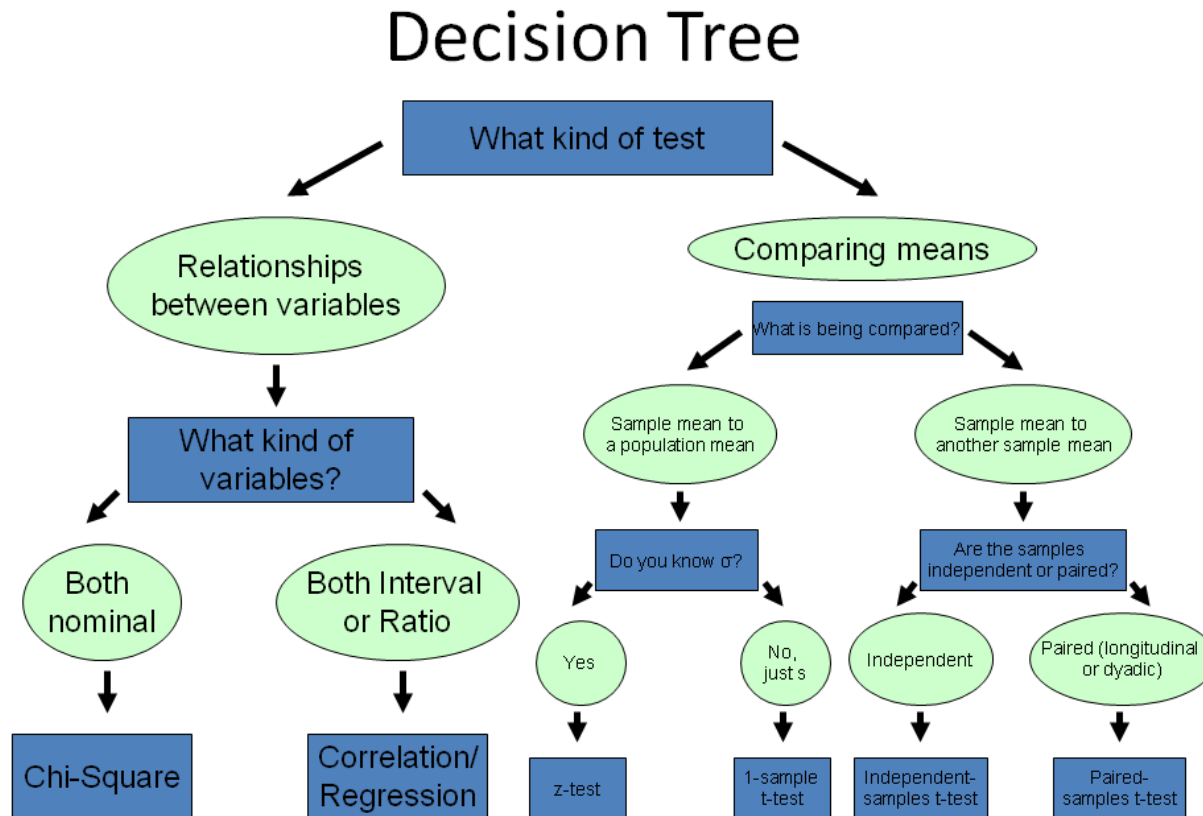
```
R> png("correlation1.png")
R> plot(patients$BLOOD_NEUTROPHILS_ABS,patients$SPUTUM_NEUTROPHILS_ABS)
R> reg1 <- lm(SPUTUM_NEUTROPHILS_ABS ~ BLOOD_NEUTROPHILS_ABS, data=patients)
R> abline(reg1)
R> dev.off()
```


Erste Statistik: Skripting

```
tibu@cheshire:~$ ./test.R
R>
R> patients <- read.csv("patients.csv")
R> png("correlation1.png")
R> plot(patients$BLOOD_NEUTROPHILS_ABS,patients$SPUTUM_NEUTROPHILS_ABS)
R> reg1 <- lm(SPUTUM_NEUTROPHILS_ABS ~ BLOOD_NEUTROPHILS_ABS, data=patients)
R> abline(reg1)
R> dev.off()
null device
      1
R>
tibu@cheshire:~$ ls correlation1.png
correlation1.png
```

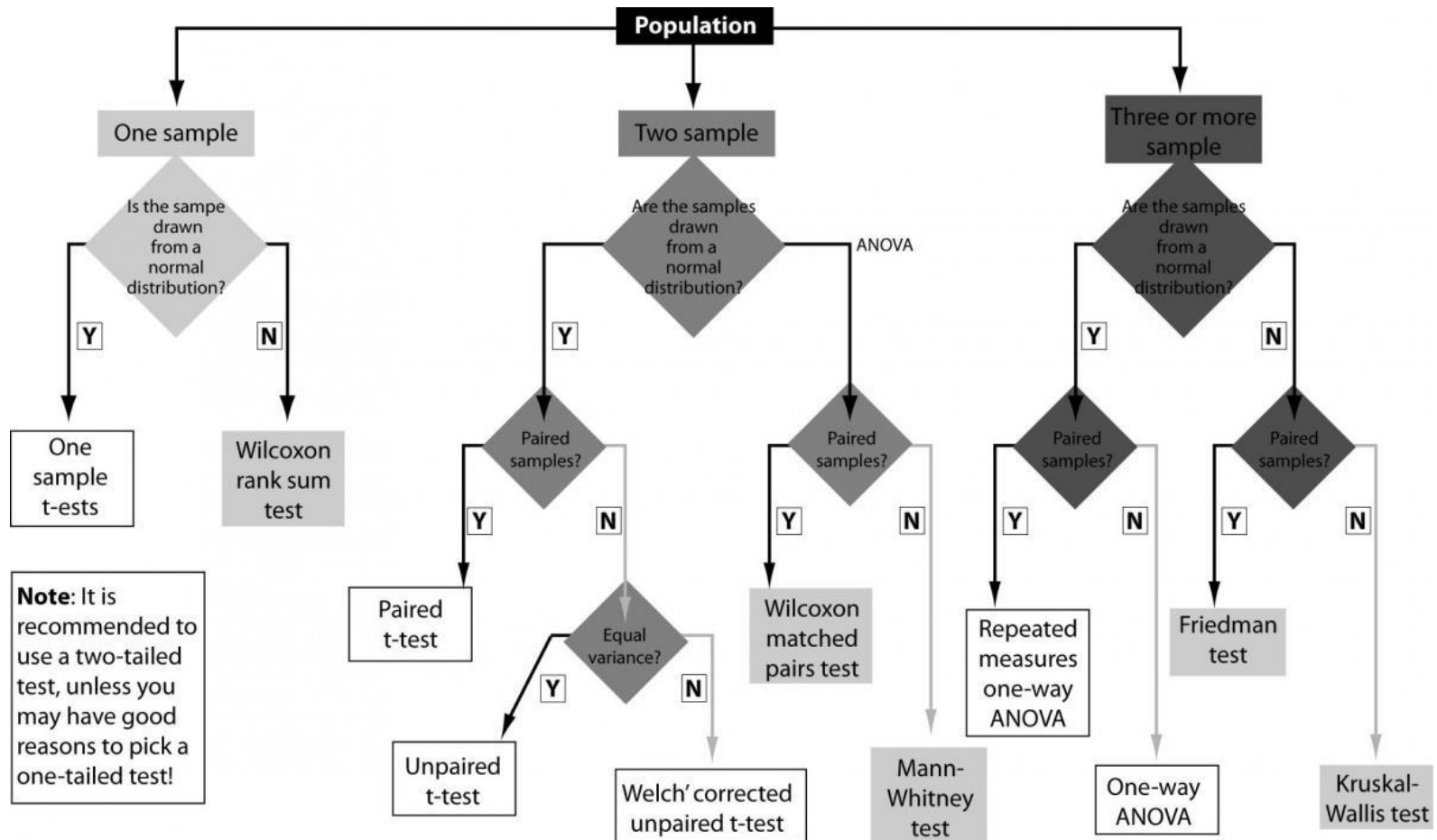
Erste Statistik: Hypothesentest - Mittelwertsvergleich

- Welcher ist der richtige für mich?



Erste Statistik: Hypothesentest - Mittelwertsvergleich

■ Welcher ist der richtige für mich?



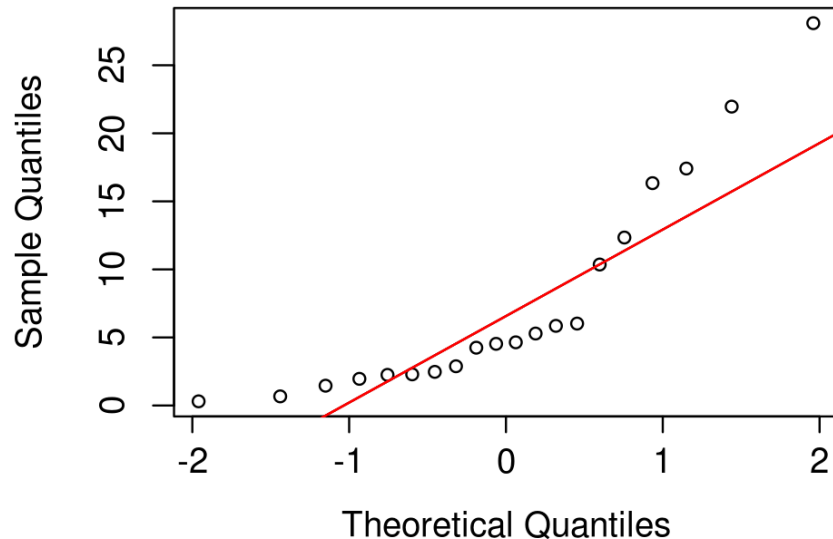
Erste Statistik: Hypothesentest - Mittelwertsvergleich

Live

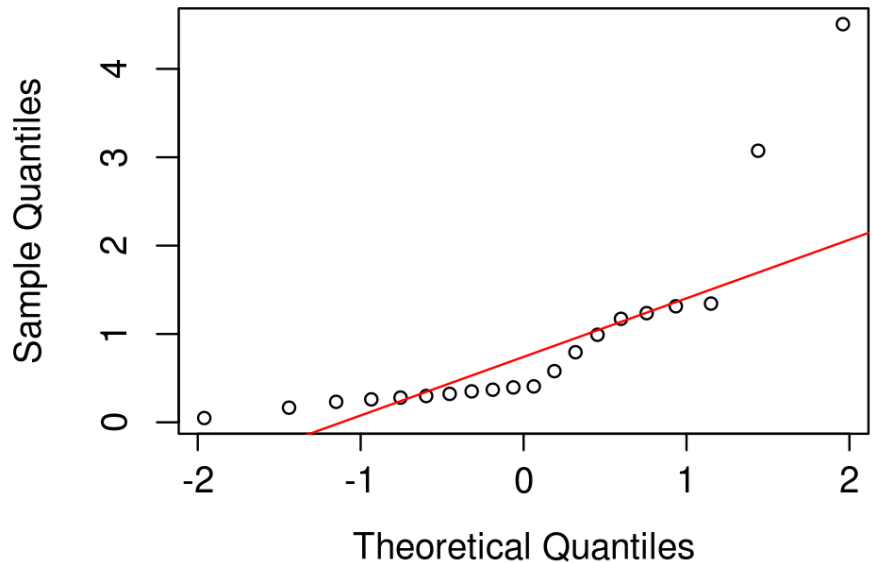
- Normalerweise würde man hier den t-Test anwenden (`t.test(...)`)
- Grundannahmen prüfen (hier z.B.: Normalität der Werte)

```
R> qqnorm(patients[patients$Copd == "Y", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot COPDists")
R> qqline(patients[patients$Copd == "Y", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot COPDists", col=2)
R> qqnorm(patients[patients$Copd == "N", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot Healthy")
R> qqline(patients[patients$Copd == "N", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot Healthy", col=2)
```

Q-Q Plot COPDists



Q-Q Plot Healthy



Erste Statistik: Hypothesentest – Verteilungs(un)gleich

Live

- Grundannahme des Wilcoxon-Mann-Whitney-Test ist das Vorhandensein zweier unabhängiger Stichproben, die zumindest einer Ordinalskala entsprechen

```
R> wilcox.test(SPUTUM_NEUTROPHILS_ABS ~ Copd, data=patients)
```

Wilcoxon rank sum test

data: SPUTUM_NEUTROPHILS_ABS by Copd

W = 35, p-value = 1.126e-06

alternative hypothesis: true location shift is not equal to 0

Erste Statistik: Hypothesentest – Ein paar Worte zum p-Wert

■ Was ist der

■ $p = P(D$

■ $\rightarrow ? P(H_1$

■ $p < 0.05$
Nullhypoc

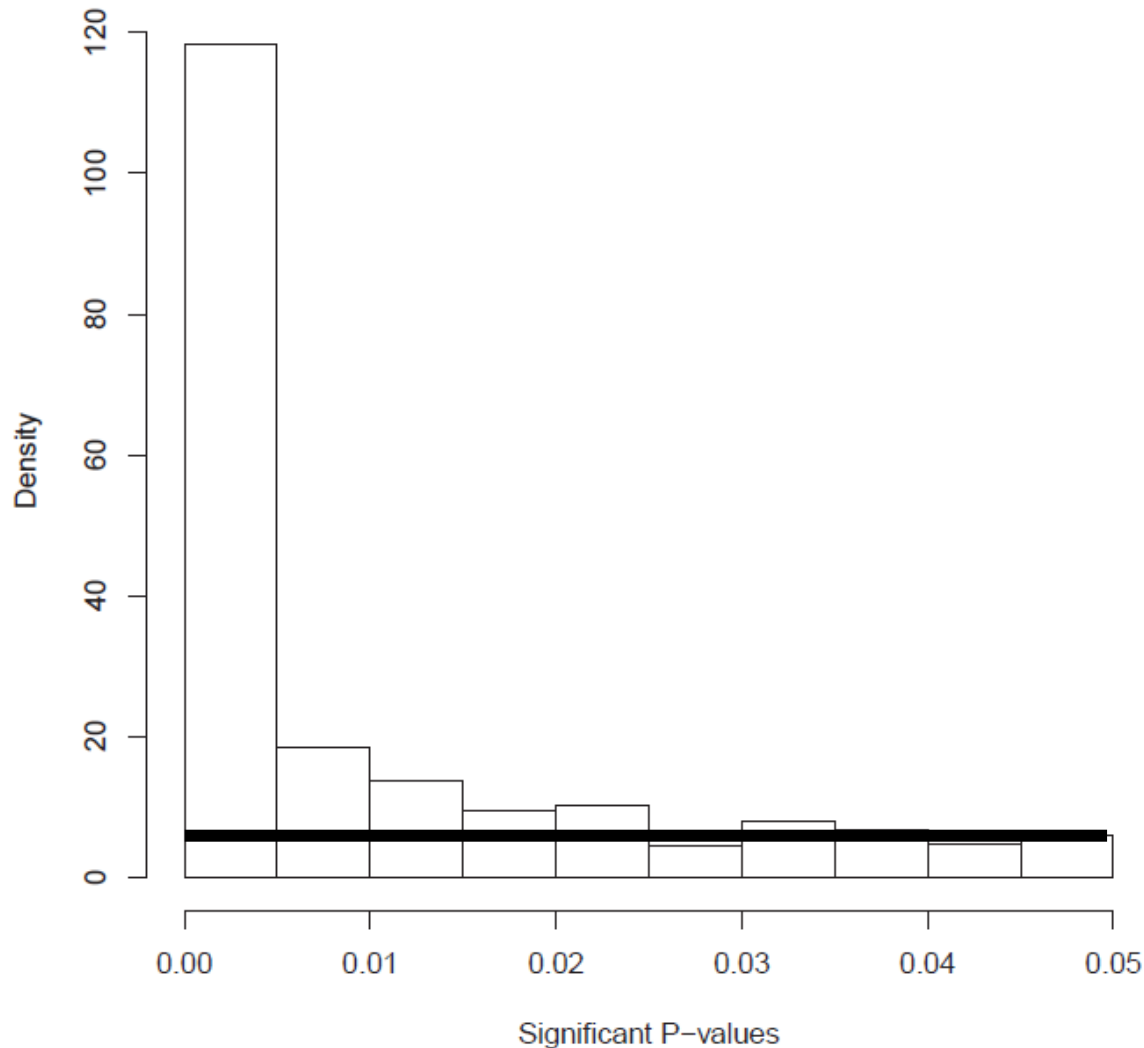
Revised

Valen E. Johnsc

Department of Stati

Edited by Adrian E.

Recent advances
development of



1, dass die

13)

f the observed data
ted with respect to

Erste Statistik: Hypothesentest – Ein paar Worte zum p-Wert

■ Vorschlag:

- $p\text{-Wert} < 0.01$ oder < 0.005
- Immer Effektgröße mit angeben, z.B. für t-test:

- $$d = \frac{|\mu_1 - \mu_2|}{\sigma}$$

■ Korrigieren sie für multiples Tests:

```
R> pvals <- c(0.013, 0.025, 0.05, 0.5, 0.75)
R> p.adjust(pvals,method="bonferroni")
[1] 0.065 0.125 0.250 1.000 1.000
R> p.adjust(pvals,method="BH")
[1] 0.06250 0.06250 0.08333 0.62500 0.75000
```

- Falls Sie das mehr interessiert: Schauen Sie in baysianische Statistik (z.B. R-Paket „BEST“)

Erste Statistik: Vorhersage

Live

- Um es weniger komplex zu machen, benutzen wir ein parametrisches Modell, logarithmieren aber die Daten

```
R> fit <- glm(Copd ~ log(SPUTUM_NEUTROPHILS_ABS), data=patients, family="binomial")  
R> summary(fit)
```

Call:

```
glm(formula = Copd ~ log(SPUTUM_NEUTROPHILS_ABS), family = "binomial",  
     data = patients)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.9785	-0.4532	0.0219	0.5419	2.3381

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.696	0.503	-1.38	0.1662
log(SPUTUM_NEUTROPHILS_ABS)	1.661	0.496	3.35	0.0008

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 55.452 on 39 degrees of freedom
Residual deviance: 30.294 on 38 degrees of freedom
AIC: 34.29

Number of Fisher Scoring iterations: 5

Erste Statistik: Vorhersage

Live

■ Eine Cross-Validierung ist sinnvoll

```
R> cv.fit <- cv.glm(patients, fit)
R> cv.fit$delta[2]
[1] 0.1319
```

Erste Statistik: Vergleich mit Zufall

Live

```
R> patients$RandGroup <- sample(patients$Copd,nrow(patients))
R> fit.random <- glm(RandGroup ~ log( Sputum_NEUTROPHILS_ABS),data=patients,family="binomial")
R> summary(fit.random)
```

[...]

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-0.105	0.335	-0.31	0.75
log(Sputum_NEUTROPHILS_ABS)	0.242	0.218	1.11	0.27

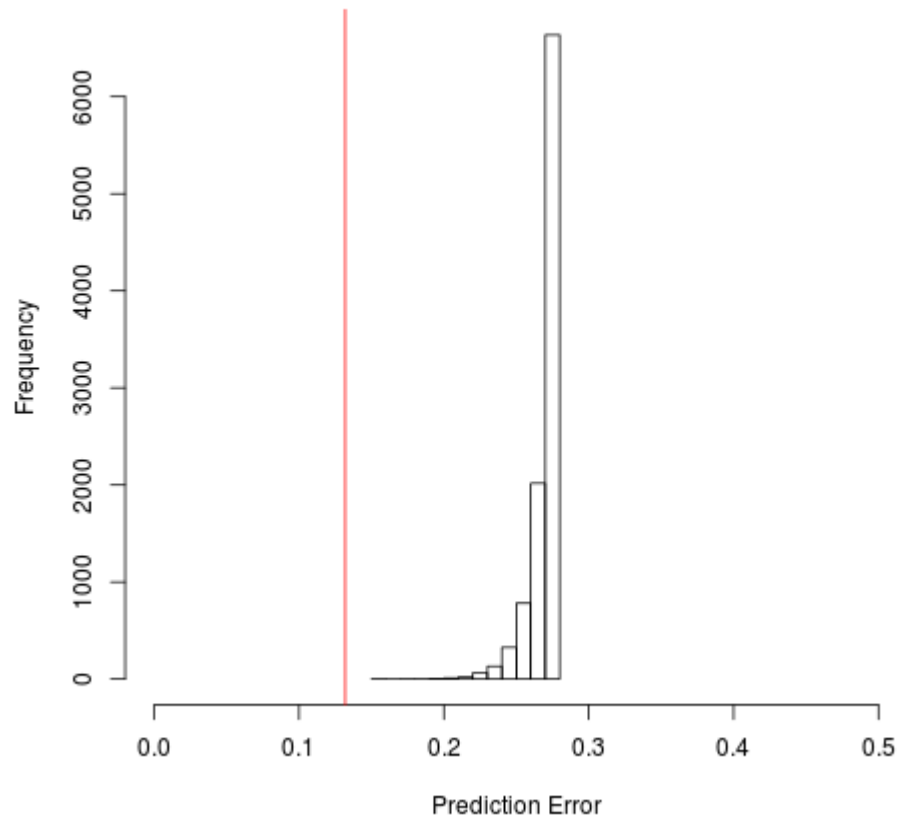
[...]

```
R> cv.fit.random <- cv.glm(patients,fit.random)
R> cv.fit.random$delta[2]
[1] 0.2688
```

```
R> histogram <- sample(1:10000,function(i) {
+ patients$RandGroup <- sample(patients$Copd,nrow(patients))
+ fit.random <- glm(RandGroup ~ log( Sputum_NEUTROPHILS_ABS),data=patients,family="binomial")
+ cv.fit.random <- cv.glm(patients,fit.random)
+ return(cv.fit.random$delta[2]) })
R> hist(histogram,xlim=c(0,0.5),xlab="Prediction Error",ylab="Frequency",main="")
R> abline(v=0.1319,col="red")
```

Erste Statistik: Vergleich mit Zufall

Live



■ ($p < 1e6$)

Weitere Features: C-Anbindung

- Warum C/C++?
 - Schneller
 - OpenMP
 - Bessere Datenstrukturen
- Native Interface: Clumsy, 80er-Jahre, un-ästhetisch
- Besser: Rcpp + Inline

Weitere Features: C-Anbindung

■ Inline (C in R):

```
R> library("Rcpp")
R> func_str <- 'int add(int x, int y, int z) {
+   int sum = x + y + z;
+   return sum;
+ }'
R> add <- cppFunction(func_str)
R> add(1,2,3)
[1] 6
```

Weitere Features: C-Anbindung

■ Rcpp – Beispiel C in R

■ sumC.cpp:

```
#include <Rcpp.h>
using namespace Rcpp;
// [[Rcpp::export]]
double sumC(NumericVector x) {
    int n = x.size();
    double sum = 0;

    for(int i(0); i < n; ++i) {
        sum += x[i];
    }
    return sum;
}
```

■ In R:

```
R> library("Rcpp")
R> sourceCpp("sumC.cpp")
R> sumC(c(1,2,3))
[1] 6
```

Weitere Features: C-Anbindung

- R aus C/C++ geht auch
- Erstellung dynamischer Objekte geht auch -> Besser man baut ein ganzes R-Paket

Weitere Features: Latex-Integration

- Pakete „Sweave“ oder „knitr“
- Ich ziehe knitr vor (kann Plots in Schleifen)
- Beispiel-Latex:

```
\documentclass[11pt, a4paper]{article}
\title{Beispiel-Knitr}
\begin{document}
\maketitle
```

```
<<>>=
patients <- read.csv("patients.csv")

qqnorm(patients[patients$Copd == "Y", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot
COPDists") >
qqline(patients[patients$Copd == "Y", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot
COPDists", col=2) >
qqnorm(patients[patients$Copd == "N", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot
Healthy") >
qqline(patients[patients$Copd == "N", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot
Healthy", col=2)
@
```

```
\end{document}
```


Weitere Features: Latex-Integration

Live

■ In R:

```
R> library("knitr")
R> knit("beispiel.Rnw")
```

```
processing file: beispiel.Rnw
|.....| 33%
ordinary text without R code

|.....| 67%
label: unnamed-chunk-1
|.....| 100%
ordinary text without R code
```

```
output file: beispiel.tex
```

```
[1] "beispiel.tex"
```

Weitere Features: Latex-Integration

■ Ergebnis-PDF

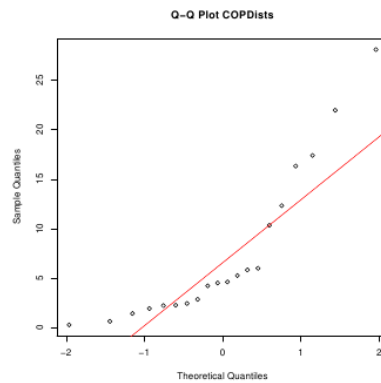
Live

Beispiel-Knitr

June 19, 2014

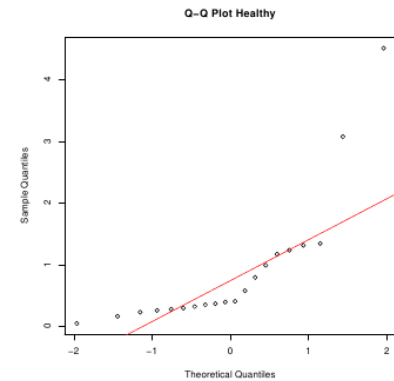
```
patients <- read.csv("patients.csv")
```

```
qqnorm(patients[patients$COPD == "Y", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot COPD")  
qqline(patients[patients$COPD == "Y", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot COPD")
```

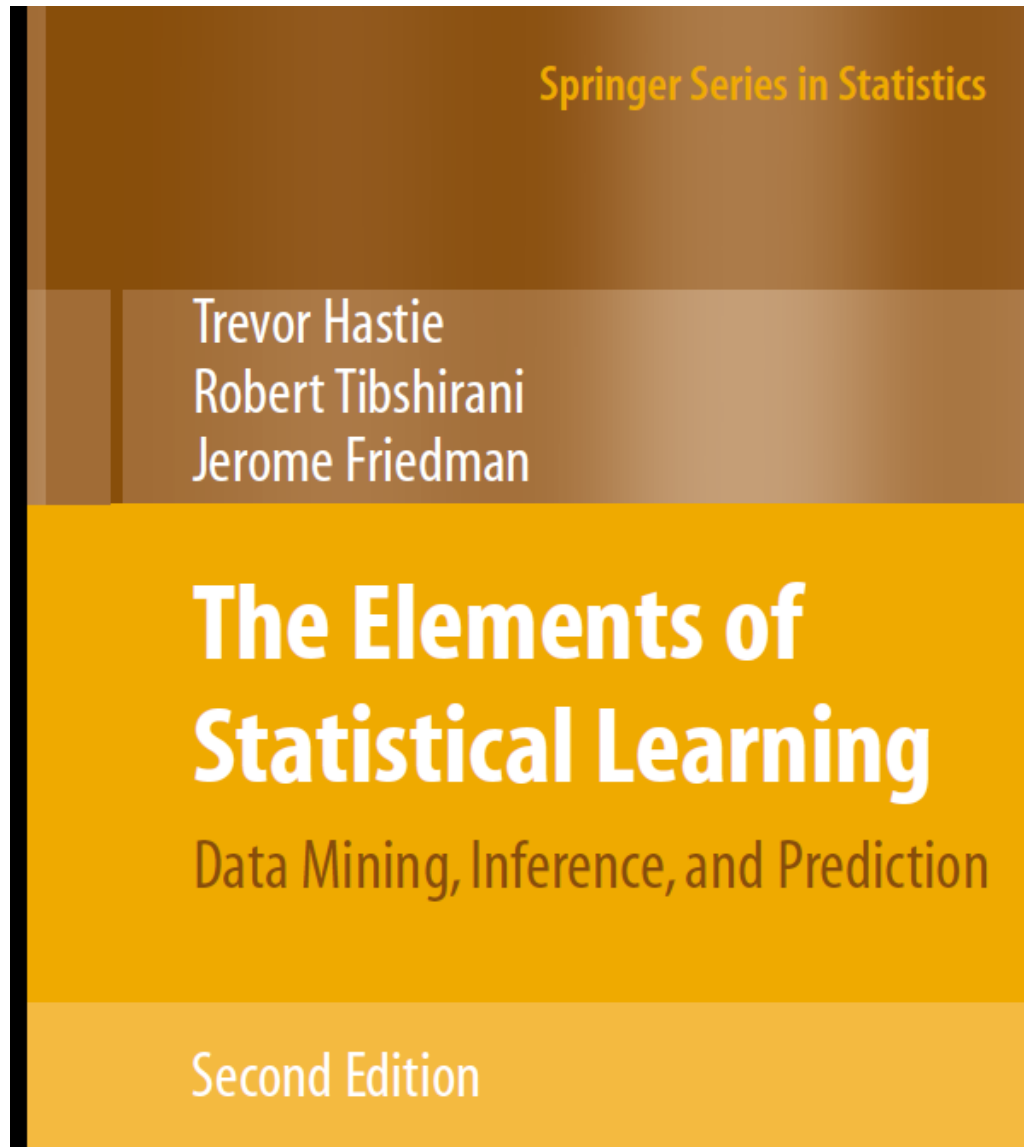


1

```
qqnorm(patients[patients$COPD == "N", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot Healthy")  
qqline(patients[patients$COPD == "N", "SPUTUM_NEUTROPHILS_ABS"], main="Q-Q Plot Healthy")
```



2



<http://www-stat.stanford.edu/~tibs/ElemStatLearn/download.html>

Ende

**Vielen Dank fürs
Zuhören!**