

WERKZEUGE DER MUSTERERKENNUNG UND DES MASCHINELLEN LERNENS

Vorlesung im Sommersemester 2017

Prof. E.G. Schukat-Talamazzini

Stand: 6. März 2017

Teil III

Text, Datensätze und Grafik in 

Zeichenketten — spezielle Funktionen

Datensätze — spezielle Funktionen

Eingabefunktionen & Ausgabefunktionen

'R'-Programmpakete — installieren & laden

Grundlegende Mechanismen der 'R'-Grafikfunktionen

Univariate und bivariate Darstellungen

Räumliche und vieldimensionale Darstellungen

Das Trellis-Grafiksubsystem

(Vektoren von) Zeichenketten

Zeichen in 'R' \triangleq Zeichenketten der Länge 1

- **Konstruktion und Konversion**

character(5)	"" "" "" "" "
as.character(5:1)	"5" "4" "3" "2" "1"
LETTERS [-(4:24)]	"A" "B" "C" "Y" "Z"
DDR <- c ("Deutsche", "Demokratische", "Republik")	

- **Projektion und Transformation**

nchar (DDR)	8 13 8
c (tolower(DDR)[1], toupper(DDR)[3])	"deutsche" "REPUBLIK"

- **Ersetzung von Zeichen und Ketten**

chartr (old=DDR[1], new=DDR[3], x=DDR[2])	"Rkmokrauiblik"
sub ("t", "TEE", DDR[1])	"DeuTEEsche"
sub (".", "?", DDR[1])	"?eutsche"
gsub (".", "?", DDR[1])	"?????????"

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

Suchen, Zerschneiden und Verketten

- Suche nach Mustervorkommen**

```
grep ("Tisch", DDR)           integer(0)
grep ("Tisch", DDR, ignore.case=TRUE) 2
```

- Zerschneiden von Ketten**

```
strsplit (DDR[1], split="e")      "D" "utsch"
strsplit (DDR[1], split="")       "D" "e" "u" "t" "s" "c" "h"
substr (x=DDR, start=3, stop=5)   "uts" "mok" "pub"
abbreviate (DDR, minlen=4)       "Dtsc" "Dmkr" "Rpbl"
```

- Verkettung (Konkatenation)**

```
paste ("hello", "world", "!")    "hello world !"
paste ("TOP", 1:3)               "TOP 1" "TOP 2" "TOP 3"
paste ("BMW ", seq(1,7,by=2), 30, "i", sep="")
                                "BMW 130i" "BMW 330i" "BMW 530i" "BMW 730i"
paste (letters[1:8], 4, sep="", collapse=".")
                                "a4·b4·c4·d4·e4·f4·g4·h4"
```

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

Partielles String Matching

```
pmatch (x, table, nomatch=NA, duplicates.ok=FALSE)
```

- Index kompletter/partieller Vorkommen**

```
average <- c("mean", "median", "mode")
pmatch (c("mode", "med", "foo"), average) 3 2 NA
```

- Partielle Treffer müssen eindeutig sein!**

```
pmatch (c("me", "mo", ""), average) NA 3 NA
pmatch (c("me", "mo", ""), average, nomatch=0) 0 3 0
```

- Getroffene Einträge werden deaktiviert!**

```
pmatch (c("medi", "med"), average) 2 NA
pmatch (c("medi", "med"), average, dup=TRUE) 2 2
```

- Komplette Treffer zuerst!**

```
pmatch (c("medi", "median"), average) NA 2
```

- Komplette Treffer werden immer akzeptiert!**

```
pmatch(c("L", "L"), c("LOL", "L"), dup=FALSE) 2 1
pmatch(c("L", "L"), c("LOL", "L"), dup=TRUE) 2 2
```

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

Uunausgewertete Ausdrücke als Sprachobjekte

Klassen `expression`, `name` und `call`

- Programmtext ~ Ausdruck**

```
parse (text="1:3+4; iris; NA") -> ex
print (ex)                  expression (1:3+4, iris, NA)
class (ex)                   "expression"
parse (file="prog.R")        Programmtext von Datei lesen
```

- Der Kantorovic-Baum eines Ausdrucks**

```
sapply (ex, class)          "call" "name" "logical"
sapply (ex[[1]], class)     "name" "call" "numeric"
sapply (ex[[1]][[2]], class) "name" "numeric" "numeric"
```

- Auswerten eines (unausgewerteten) Ausdrucks**

```
eval (ex)                   NA
eval (ex[[1]])              5 6 7
eval (ex[[1]][[1]])         function (e1, e2) .Primitive("+")
```

- Ausdruck ~ Programmtext**

```
supply (ex, deparse)        "1:17 + 4" "iris" "NA"
```

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

Zeichenketten — spezielle Funktionen

Datensätze — spezielle Funktionen

Eingabefunktionen & Ausgabefunktionen

'R'-Programmpakete — installieren & laden

Grundlegende Mechanismen der 'R'-Grafikfunktionen

Univariate und bivariate Darstellungen

Räumliche und vieldimensionale Darstellungen

Das Trellis-Grafiksubsystem

?character	?data.frame	E/A	Pakete	Grafik	1D/2D-Plots	3D/HD-Plots	lattice	?character	?data.frame	E/A	Pakete	Grafik	1D/2D-Plots	3D/HD-Plots	lattice																																																																														
<h2>Datensätze in Mustererkennung und Data Mining</h2>																																																																																													
Matrix $\hat{=}$ Objekte (Fälle, Muster) \times Attribute (Variable, Merkmale)																																																																																													
<h3>Beispieldatensatz</h3>																																																																																													
<table border="1"> <thead> <tr><th></th><th>vorbestraft</th><th>Partei</th><th>Abinote</th><th>Geburt</th><th>Spenden</th></tr> </thead> <tbody> <tr><td>Angela</td><td>F</td><td>CDU</td><td>gut</td><td>1954</td><td>$345 \cdot 10^3$</td></tr> <tr><td>Guido</td><td>F</td><td>FDP</td><td>ausreichend</td><td>1961</td><td>$137 \cdot 10^3$</td></tr> <tr><td>Roland</td><td>T</td><td>CDU</td><td>gut</td><td>1958</td><td>$3.6 \cdot 10^6$</td></tr> <tr><td>Gregor</td><td>F</td><td>PDS</td><td>sehr gut</td><td>1948</td><td>NA</td></tr> <tr><td>Linus</td><td>F</td><td>Pirat</td><td>NA</td><td>1969</td><td>0</td></tr> <tr><td>Bill</td><td>F</td><td>Rep</td><td>mangelhaft</td><td>1955</td><td>$-4.2 \cdot 10^9$</td></tr> <tr><td>Roman</td><td>T</td><td>...</td><td>...</td><td>1933</td><td>...</td></tr> <tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr> <tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr> <tr><td>Range</td><td>$\{T, F\}$</td><td>$\{\pi_1, \dots, \pi_9\}$</td><td>$\{\nu_1, \dots, \nu_5\}$</td><td>$\mathbb{Z} \subset \mathbb{R}$</td><td>$\mathbb{R}$</td></tr> <tr><td>Skala</td><td>nominal</td><td>nominal</td><td>ordinal</td><td>relativ</td><td>absolut</td></tr> <tr><td>'R'-Typ</td><td>factor</td><td>factor</td><td>ordered</td><td>numeric</td><td>numeric</td></tr> </tbody> </table>																	vorbestraft	Partei	Abinote	Geburt	Spenden	Angela	F	CDU	gut	1954	$345 \cdot 10^3$	Guido	F	FDP	ausreichend	1961	$137 \cdot 10^3$	Roland	T	CDU	gut	1958	$3.6 \cdot 10^6$	Gregor	F	PDS	sehr gut	1948	NA	Linus	F	Pirat	NA	1969	0	Bill	F	Rep	mangelhaft	1955	$-4.2 \cdot 10^9$	Roman	T	1933	...	:	:	:	:	:	:	:	:	:	:	:	:	Range	$\{T, F\}$	$\{\pi_1, \dots, \pi_9\}$	$\{\nu_1, \dots, \nu_5\}$	$\mathbb{Z} \subset \mathbb{R}$	\mathbb{R}	Skala	nominal	nominal	ordinal	relativ	absolut	'R'-Typ	factor	factor	ordered	numeric	numeric
	vorbestraft	Partei	Abinote	Geburt	Spenden																																																																																								
Angela	F	CDU	gut	1954	$345 \cdot 10^3$																																																																																								
Guido	F	FDP	ausreichend	1961	$137 \cdot 10^3$																																																																																								
Roland	T	CDU	gut	1958	$3.6 \cdot 10^6$																																																																																								
Gregor	F	PDS	sehr gut	1948	NA																																																																																								
Linus	F	Pirat	NA	1969	0																																																																																								
Bill	F	Rep	mangelhaft	1955	$-4.2 \cdot 10^9$																																																																																								
Roman	T	1933	...																																																																																								
:	:	:	:	:	:																																																																																								
:	:	:	:	:	:																																																																																								
Range	$\{T, F\}$	$\{\pi_1, \dots, \pi_9\}$	$\{\nu_1, \dots, \nu_5\}$	$\mathbb{Z} \subset \mathbb{R}$	\mathbb{R}																																																																																								
Skala	nominal	nominal	ordinal	relativ	absolut																																																																																								
'R'-Typ	factor	factor	ordered	numeric	numeric																																																																																								

- **Konstruktorauftruf** $\hat{=}$ Objekt der Klasse `data.frame`
`data.frame(..., row.names=NULL)`
Parameter: $\begin{cases} \text{Vektoren} \\ \text{Matrizen} \end{cases} \begin{cases} \text{mit} \\ \text{ohne} \end{cases} \text{Namen}; \text{ Zeilennamen: } \begin{cases} \text{char(n) explizit} \\ \text{int(1) implizit} \end{cases}$
- **Matrixkonversion (Spalten=Merkmale)**
`as.data.frame(x=diag(5), row.names=LETTERS[1:5])`
- **Laden verfügbarer Datensätze**
 - `data()` listet alle Datensätze
 - `data(package="MASS")` listet `MASS`-Datensätze
 - `data(eagles, package="MASS")` macht `eagles` sichtbar
- **Lesen einer Datensatzdatei**
`read.table, load, source, ...` (je nach Format)

?character	?data.frame	E/A	Pakete	Grafik	1D/2D-Plots	3D/HD-Plots	lattice
<h2>Datensätze laden — auch über das Internet</h2>							

- Als Dateinamen werden auch URLs akzeptiert:

```
myurl <- 'http://stat.ethz.ch/Teaching/Datasets/NDK/sport.dat'
sport <- read.table(myurl)
```

- Datensätze mit benannten Mustern & Merkmalen:

	weit	kugel	hoch	disc	stab	speer	punkte
OBRIEN	7.57	15.66	207	48.78	500	66.90	8824
BUSEMANN	8.07	13.60	204	45.04	480	66.86	8706
DVORAK	7.60	15.82	198	46.28	470	70.16	8664
FRITZ	7.77	15.31	204	49.84	510	65.70	8644
HAMALAINEN	7.48	16.32	198	49.62	500	57.66	8613
NOOL	7.88	14.01	201	42.98	540	65.48	8543
ZMELIK	7.64	13.53	195	43.44	540	67.20	8422
GANIYEV	7.61	14.71	213	44.86	520	53.70	8318
PENALVER	7.27	16.91	207	48.92	470	57.08	8307
HUFFINS	7.49	15.57	204	48.72	470	60.62	8300
PLAZIAT	7.82	14.85	204	45.34	490	52.18	8282
MAGNUSSON	7.28	15.52	195	43.78	480	61.10	8274
SMITH	7.47	16.97	195	49.54	500	64.34	8271
MUELLER	7.25	14.69	195	45.90	510	66.10	8253
CHMARA	7.75	14.51	210	42.60	490	54.84	8249

- Auswahl von Zeilen und Spalten, z.B. `sport[, 'kugel']`

```
[1] 15.66 13.60 15.82 15.31 16.32 14.01 13.53 14.71 16.91 15.57 14.85 15.52
[13] 16.97 14.69 14.51
```

?character	?data.frame	E/A	Pakete	Grafik	1D/2D-Plots	3D/HD-Plots	lattice
<h2>Datensätze inspizieren</h2>							

- **Matrixeigenschaften**

```
names(sport)
rownames(sport)
sport['FRITZ', 'speer']
```

Datensatz, kein Vektor!!

- **Listeneigenschaften**

```
length(sport) == ncol(sport)
sport[3] versus sport[[3]]
```

Datensatz oder Vektor ?!

- **Attribute in den Namensraum transportieren**

```
attach(sport); mean(kugel); detach(sport)
with(sport, mean(kugel))
```

dieselbe Wirkung

- **Komfortable Ausschnittbildung 1**

```
subset(x=sport, subset=hoch>=200, select=weit:hoch)
Zeilenwahl (logical) und Spaltenwahl (symbol. Indexausdruck)
```


Zoo der E/A-Funktionen in R



„Wer lesen und schreiben kann, ist klar im Vorteil!“

Zuständigkeit

'R'-Objekte
'R'-Klassen
'R'-Atome

Logische Darstellung

Typbezogenes Datenformat
Programmtext ('R')
Objektserialisierung

Abstraktionsebene

Systemschnittstelle
Benutzerschnittstelle

Physische Darstellung

Datei — Strom
ASCII — Binärformat
Redundant — Komprimiert

E/A für elementaren Freitext

Funktionen cat (... , file, sep) und scan (file, what)

- **Textausgabe für Atome und Felder** (Standardausgabe)
cat (1:3, 'Wort', matrix(1:4,2)) 1 2 3 Wort 1 2 3 4

- **Ausgabe mit Trennmuster**

cat (LETTERS[1:5], 'A', sep='=>') A=>B=>C=>D=>E=>A

- **Ausgabe in Datei**

cat (1:4, '\n', letters[1:3], '\n', file='foo.txt')

- **Eingabe typgleicher Werte von Datei**

scan ('foo.txt') Fehler in scan() - erwartete 'a real', bekam 'a'
scan ('foo.txt', character(0)) "1" "2" "3" "4" "a" "b" "c"
scan ('foo.txt', what='xy', nmax=4) "1" "2" "3" "4"
scan ('foo.txt', what=numeric(0), nmax=4) 1 2 3 4

- **Formatierte Eingabe (Zeitkiller!)**

lst <- scan ('foo.txt', what=list (0,0,0,0,'',''))
sapply (lst, length) 1 1 1 1 1 1 1

E/A für Datensätze (und Matrizen)

Funktionen write.table und read.table

- **Formatierte Ausgabe von Datensätzen**

write.table (x, file="", append=FALSE)
Konversion, Standardausgabe, Dateibeginn as.data.frame (x)

- **Formatierte Eingabe von Datensätzen**

read.table (file, header=FALSE, sep=',', dec='.', as.is=, ...)
liefert data.frame-Objekt (Spaltennamen?, Zeilennamen?, Layoutdetails?)
seltener im Einsatz:

- **Formatierte Ausgabe von Matrizen** (alle Typen)

write (x, file, ncolumns, append=FALSE, sep=' ')
vorzugsweise ncolumns=ncol(x), denn:

- **Formatierte Eingabe von Matrizen** (als typhomogener Datensatz)

as.data.frame (read.table (file, header=TRUE))
Standardzeilen- und spaltennamen: 1 2 ... m und V1 V2 ... Vn

Transparente E/A für beliebige 'R'-Objekte

Funktionen save und load speichern Objektstruktur & Objektbestandteile

- **Automatische Serialisierbarkeit aller 'R'-Objekte**

A <- matrix (rnorm(1000), 25, 40)
B <- list (name=c('Amy', 'W.'), age.max=27, alive=F)
C <- function (m, c=3e8) E <- m * c^2

- **Gemeinsames Abspeichern auf Datei**

save (A, "B", C, file="foo.rda")
oder alternativ:
save (list=list ('A', 'B', 'C'), file="foo.rda")

- **Gemeinsames Rückladen von Datei**

z <- load (file="foo.rda")
print (z) "A" "B" "C"
ls () unter anderem: A B C

- **Was passiert am Sitzungsende?** (nach quit('yes'))

save (list=ls (all=TRUE), file=".RData")

Operationale E/A für beliebige 'R'-Objekte

Funktionen `dump` und `source` speichern Bauplan / Herstellungsprozeß des Objekts

- **Objektrepräsentation** \triangleq 'R'-Programmtext

AUS: Name \rightsquigarrow Ausdruck \rightsquigarrow Text

EIN: Text \rightsquigarrow Ausdruck \rightsquigarrow Eval \rightsquigarrow Binden

- **Gemeinsames Abspeichern auf Datei**

`dump (list, file="dumpdata.R", append=FALSE, ...)`

Objektnamen als Liste von Zeichenketten

- **Wiederherstellen aller Objekte**

`source (file, local=FALSE, ...)`

Standardumgebung (Namensraum) ist der globale Arbeitsbereich

- **Basisfunktionen** (Einzelobjekte, ohne Namensbindung)

`dput (x=function(x) sqrt(sum(x^2)), file="foo.R")`

`euno <- dget (file="foo.R")`

`euno (3:4) == 5`

TRUE

E/A-Systemschnittstelle

Datenströme, Atome, Serialisierung, Bytevektoren

Datenströme

Klasse `connection`

Datei `file()`

Adresse `url()`

Quelle/Senke `pipe()`

Filter `fifo()`

`open(), close(), flush()`.

Bytevektoren

Atomarer Typ `raw`

Adreßeinheit \triangleq Byte

Binärrepräsentation der E/A
(`connection=NULL`)

E/A für atomare Vektoren

`readBin()` `writeBin()`

`readChar()` `writeChar()`

`readLines()` `writeLines()`

E/A durch Serialisierung

`serialize()` `saveRDS()`

`unserialize()` `readRDS()`

Zeichenketten — spezielle Funktionen

Datensätze — spezielle Funktionen

Eingabefunktionen & Ausgabefunktionen

'R'-Programmpakete — installieren & laden

Grundlegende Mechanismen der 'R'-Grafikfunktionen

Univariate und bivariate Darstellungen

Räumliche und vieldimensionale Darstellungen

Das Trellis-Grafiksubsystem

R-Programmpakete

'R'-Implementierung

Primitive Funktionen (unzerlegbar)

Eingebaute Funktionen (C++-Bibliotheken)

Deklarierte Funktionen ('R'-Objekte)

Funktionsumfang

Systemkern (alle primitiven Fkt.)

'R'-Basispaket (deklarierte Fkt.)

Standarddistribution (aktuell 27 Pakete)

CRAN Zusatzpakete (>500 Pakete)

Paketbestandteile

Funktionsdeklarationen

Daten(sätze)

C++-Bibliothek

Handbuchseiten, Vignetten

Paketnutzung

(Bauen & Pflegen)

Installation (Rechner)

Laden (Sitzung)

Aufruf (Fkt.) & Bindung (Daten)

Programmpakete der Standarddistribution

Standort: /usr/lib64/R/library o.ä.

base	The R Base Package
boot	Bootstrap Functions (originally by Angelo Canty for S)
class	Functions for Classification
cluster	Cluster Analysis Extended Rousseeuw et al.
codetools	Code Analysis Tools for R
compiler	The R Compiler Package
datasets	The R Datasets Package
foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase
graphics	The R Graphics Package
grDevices	The R Graphics Devices and Support for Colours and Fonts
grid	The Grid Graphics Package
KernSmooth	Functions for kernel smoothing for Wand & Jones (1995)
lattice	Lattice Graphics
MASS	Support Functions and Datasets for Venables and Ripley's MASS
Matrix	Sparse and Dense Matrix Classes and Methods
methods	Formal Methods and Classes
mgcv	GAMs with GCV/AIC/REML smoothness estimation and GAMMs by PQL
nlme	Linear and Nonlinear Mixed Effects Models
nnet	Feed-forward Neural Networks and Multinomial Log-Linear Models
rpart	Recursive Partitioning
spatial	Functions for Kriging and Point Pattern Analysis
splines	Regression Spline Functions and Classes
stats	The R Stats Package
stats4	Statistical Functions using S4 Classes
survival	Survival analysis, including penalised likelihood.
tcltk	Tcl/Tk Interface
tools	Tools for Package Development
utils	The R Utils Package

CRAN Task Views

Standort: http://cran.r-project.org/

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
Finance	Empirical Finance
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
ReproducibleResearch	Reproducible Research
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
Survival	Survival Analysis
TimeSeries	Time Series Analysis
gR	gRaphical Models in R

Pakete installieren & laden

- **Installiertes Paket laden**

```
library (package='MASS') oder require (package='MASS')
```

Paket als Name oder Zeichenkette

- **Paketinformationen**

```
library () listet installierte Pakete
library (help='MASS') Details zu genanntem Paket
```

- **Paket installieren (Internetzugang!)**

```
install.packages (pkgs) Paketnamen in Zeichenkettenvektor
update.packages () spielt aktuelle Paketversionen ein
```

- **Alle Pakete einer Anwendungsklasse „task view“**

```
install.views (views="Econometrics")
update.views (views="Econometrics")
```

Vorher muß Paket `ctv` installiert und geladen werden!

```
available.views () listet verfügbare TVs
```

Zugriff auf Komponenten geladener Pakete

- **Sichtbarkeit in der aktuellen Arbeitsumgebung**

- alle Hilfetexte
- die wichtigsten Funktionsobjekte
- keine Datenobjekte

- **Datensätze ruhen in den Installationsdateien**

```
data (... , list=character(), package=NULL, ...)
```

Vier Datenformate können automatisch gelesen werden:

'R'-Quelltext
Serielles Format
Datensatz-Tabelle
Comma-Separated Values

source() *.{R,r}
load() *.{RData,rda}
read.table() *.{tab,txt,TXT}
read.csv() *.{csv,CSV}

R-Paket rimage

Lesen von JPEG-Dateien (und Manipulieren von Bildobjekten)

- Laden des rimage-Pakets

```
library (rimage)
help (package='rimage')
```

- Einlesen der JPEG-Datei in ein imagematrix-Objekt

```
imo <- read.jpeg ('Rlogo.jpg')
```

- Spezifische Methoden für print und plot

```
print (imo)           size: 77 x 101, type: rgb
plot (imo, main='Das R-Logo')
```

- Grauwertbilder sind Matrizen

```
class (imo); attr (imo,'type') [1] 'imagematrix' [1] 'grey'
dim (imo)                      [1] 77 101
```

Bildpunkte manipulieren mit `imo[i,j]`

- Farbbilder

```
class (imo); attr (imo,'type') [1] 'imagematrix' [1] 'rgb'
dim (imo)                      [1] 77 101 3
```

Bildpunkte manipulieren mit `imo[i,j,channel]`

R-Paket rimage

Schreiben von JPEG-Dateien (?!)

Problem

Beim Schreiben einer Bildmatrix (oder RGB-Würfel) als JPEG-Datei unterstützt uns das Paket leider nicht!



- Abspeichern auf Datei als imagematrix-Objekt
`save (imo, file='bild.rda')`

- Hardwareplattformunabhängiges Datenformat
`save (imo, file='bild.rda', ascii=TRUE)`

- Exakte Kompression der Bilddaten
`save (imo, file='bild.rda', compress=TRUE)`

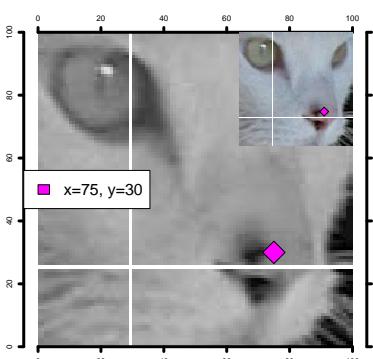
- Wiedereinlesen der (Bild-)Daten
`load (file='bild.rda')`

Auch unter den Einstellungen `ascii=F,compress=T` ist die `.rda`-Datei viel größer als die entsprechende JPEG-Datei.

Allerdings stellt sie auch eine **exakte** Repräsentation dar.

R-Paket rimage

Beispiel zu Bildgeometrie, Farbe und Grauwert



```
require (rimage)
dog <- read.jpeg (
  system.file (
    "images", "cat.jpg",
    package="rimage"))
dog <- imagematrix (dog
  [200+1:100,230+1:100,])
dog[75,] <- 1; dog[,30,] <- 1

layout (matrix (c(1,1,2,1), 2, 2))

plot (rgb2grey (dog))
for (i in 1:4)
  axis (side=i, lwd=5)
points (75, 30,
  pch=23, bg="magenta", cex=5)
legend ("left", legend="x=75, y=30",
  cex=2, bg="white", fill="magenta")

plot (dog)
points (75, 30, pch=23, bg="magenta", cex=2)
```

Bildgeometrie

der Klasse 'imagematrix':

x=75 Zeile 75

y=30 Spalte 30

Zeichenketten — spezielle Funktionen

Datensätze — spezielle Funktionen

Eingabefunktionen & Ausgabefunktionen

'R'-Programmpakete — installieren & laden

Grundlegende Mechanismen der 'R'-Grafikfunktionen

Univariate und bivariate Darstellungen

Räumliche und vieldimensionale Darstellungen

Das Trellis-Grafiksubsystem

Computergrafische Datenvisualisierung

Explorative Data Analysis: [data \mapsto image \mapsto perception \mapsto HCI]*

Zweck der Visualisierung

Gewinne erste Übersicht
Erzähle eine Geschichte
Generiere eine Hypothese
Überprüfe ein Modell

Mittel der Visualisierung

Punkte, Linien
Koordinatensystem, Achsen
Zahlenwerte, Symbole, Text
Farbe, Schattierung

Fakt

Humans are good at discerning subtle patterns that are really there, but equally so at imagining them when they are altogether absent.
„Contact“ (Carl Sagan, 1986)

Mehr Informationen

Skript „Maschinelles Lernen & Data Mining“, Kapitel 3.8 (Visualisierung:EDA)
<http://www.minet.uni-jena.de/fakultaet/schukat/ML/Scriptum/lect03-visual.pdf>

Grafikproduktion in R

... die ersten vier Laufschleifen ...

- **Wohin damit?** (... wenn nicht auf den Bildschirm)

pdf (file="Rplots.pdf", width=7, height=7, ...) [inch]

Alternativ: postscript(), jpeg(), tiff(), svg(), xfig(), pictex(), X11().

- **Leinwandaufteilung** („canvas“, z.B. 2 x 3)

layout (matrix (1:6,2,3)) oder par (mfcol=c(2,3))

- **Primärfunktion** (Koordinatensystem et al.)

plot (x, sin(x), type='b')

Alternativ: boxplot(), barplot(), hist(), contour(), persp()

- **Sekundärfunktionen** (Punkte, Linien, Symbole, Texte)

points(), lines(), polygon(), segments(), ... Daten
abline(), rect(), rug(), symbols(), text(), ... Markierung
legend(), axis(), title(), mtext(), ... Beschriftung

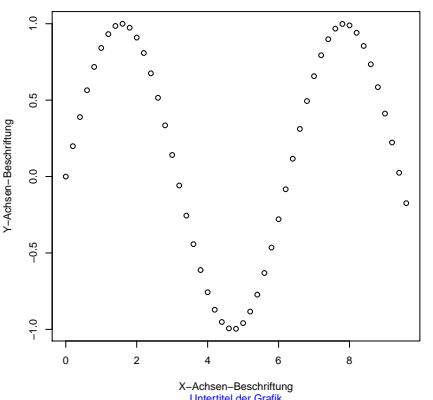
- **Grafikparameter** (siehe ?par)

1. Default · 2. Justieren: par (...) · 3. Übergeben: plot (...)

Beschriftungsparameter und -funktionen

Parameter main, sub, xlab, ylab · Sekundärfunktion title()

Titel der Grafik



```
plot (x <- 0:48/5, sin(x),
      main="Titel der Grafik",
      sub="Untertitel der Grafik",
      xlab="X-Achsen-Beschriftung",
      ylab="Y-Achsen-Beschriftung",
      cex.main=3, col.sub="blue")
```

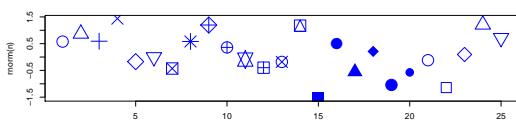
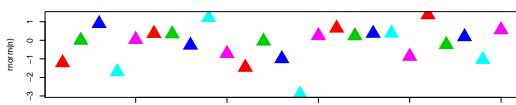
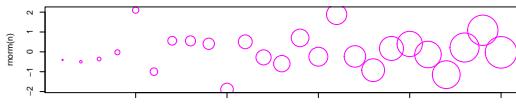
```
# DIESELBE AUSGABE BEWIRKT AUCH ...
plot (x <- 0:48/5, sin(x),
      xlab="", ylab="")
title (main="Titel der Grafik",
       sub="Untertitel der Grafik",
       xlab="X-Achsen-Beschriftung",
       ylab="Y-Achsen-Beschriftung",
       cex.main=3, col.sub="blue")
```

Parameter für Farbe und (relative) Schriftgröße

{col.main}, {col.sub}, {col.xlab}, {col.ylab} und {cex} (Punkt/Linie).
{cex.main}, {cex.sub}, {cex.xlab}, {cex.ylab} und {cex} (Punkt/Linie).

Punktweise Grafikattribute

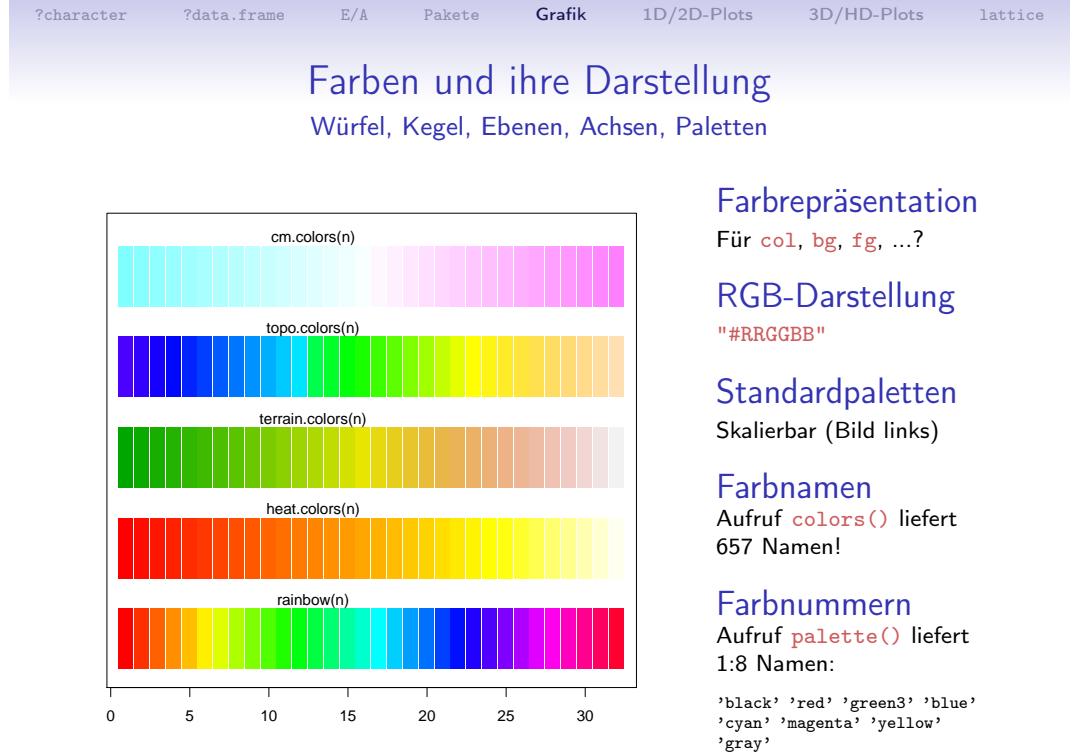
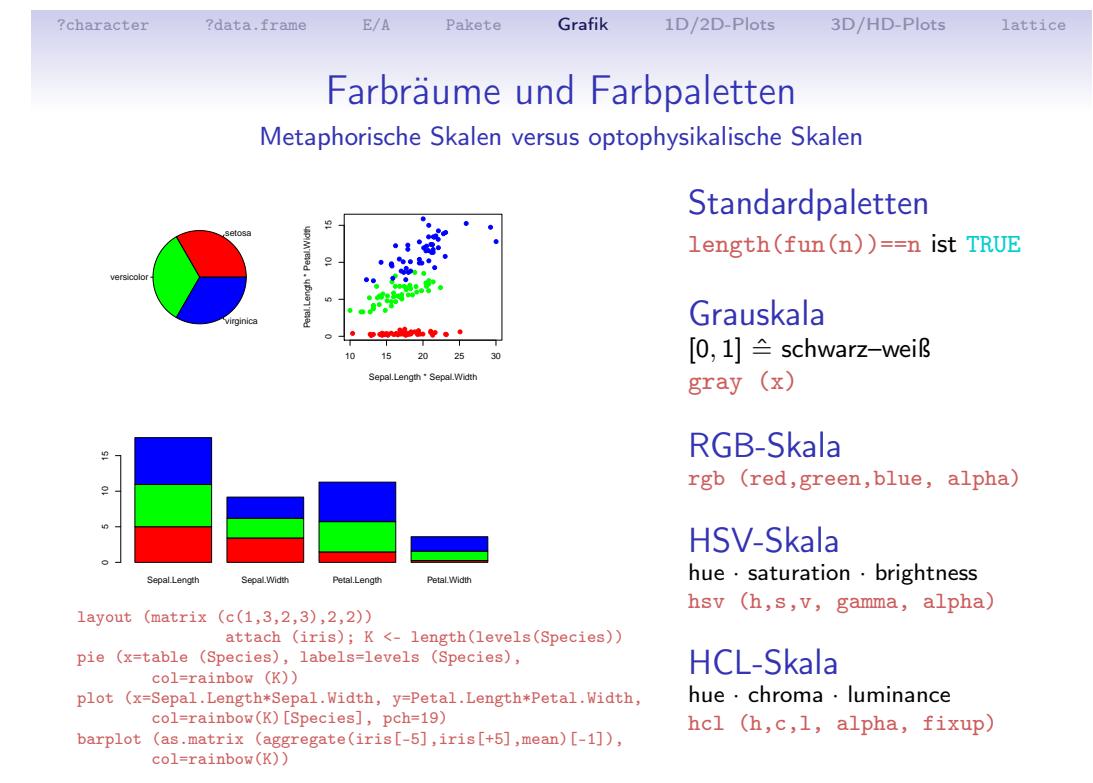
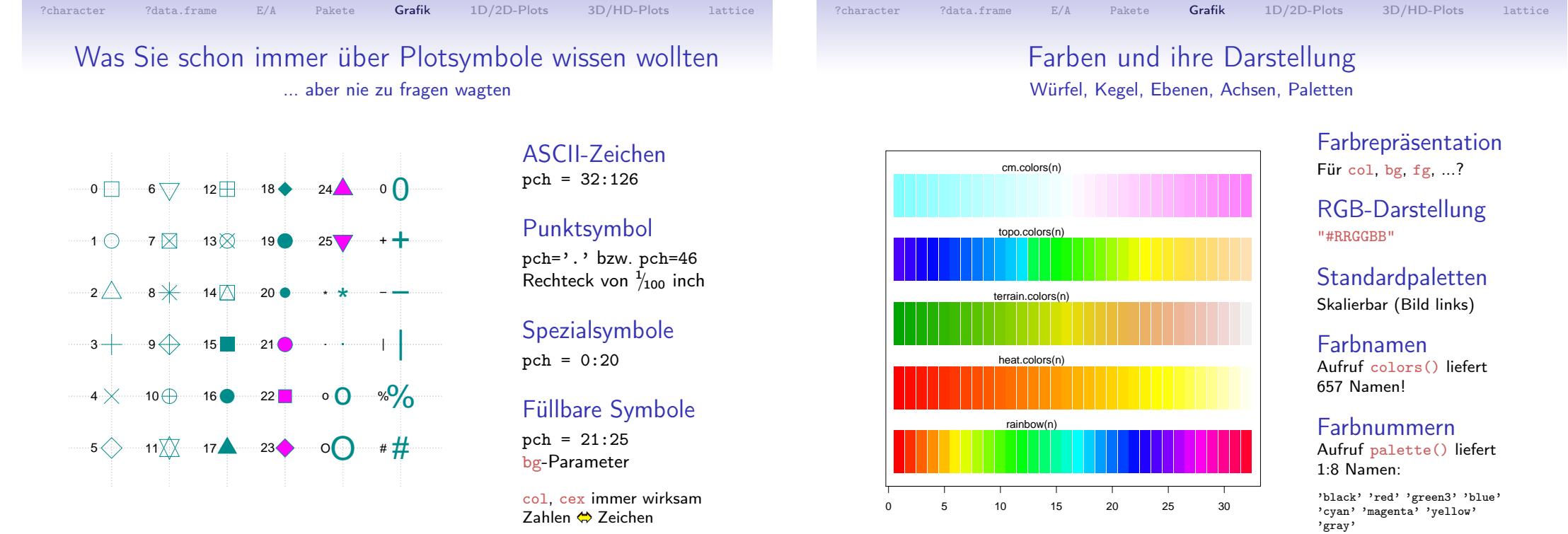
cex = Größe · col = Farbe · pch = Symbol



```
layout (1:3); n=25
plot (rnorm(n),
      col="magenta",
      cex=1:n/3)
plot (rnorm(n),
      cex=3,
      pch=17,
      col=1+1:5)
plot (rnorm(n),
      cex=3,
      col="blue",
      pch=1:n)
```

Wirkung Länge

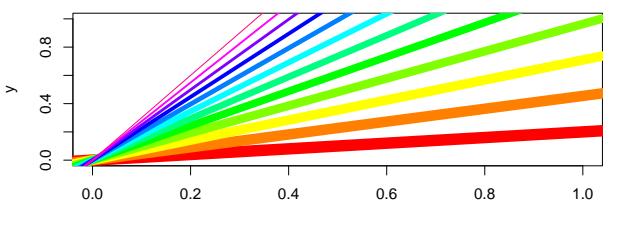
- = 1 \rightsquigarrow global
- = n \rightsquigarrow lokal
- < n \rightsquigarrow repetitiv



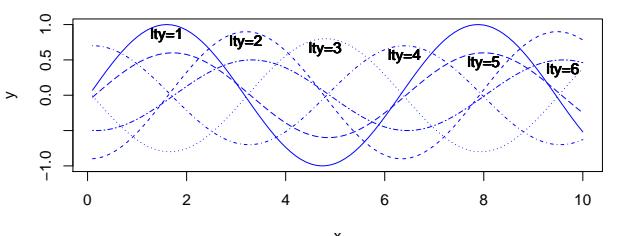
Linienattribute

Linientyp lty, Liniendicke lwd und Farbe col

Linienfarbe & Liniendicke



Linientyp



Liniendicke

$$lwd \in \mathbb{R}_0^+ \quad (\text{Standard}=1)$$

Linientyp

lty ist Zahlencode,
Schlüsselwort oder
Hexasequenz

0	"blank"	"_"
1	"solid"	"11"
2	"dashed"	"44"
3	"dotted"	"13"
4	"dotdash"	"1343"
5	"longdash"	"73"
6	"twodash"	"2262"
-	"_"	"AFFE"

Textinschrift und Abbildungslegende

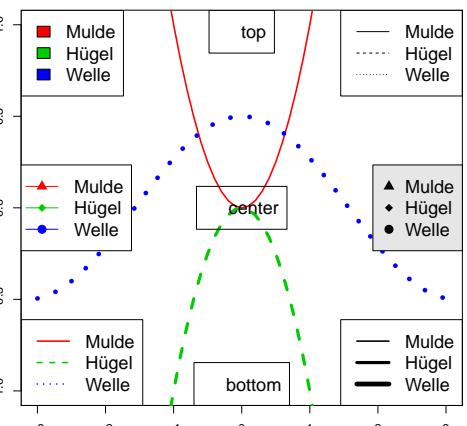
legend (x, y=NULL, legend, fill=NULL, col, lty, lwd, pch, bg, ...)

```
x <- seq (-3,+3,0.1)
txt <- c("Mulde", "Hügel", "Welle")
Y <- list (x^2, -x^2, cos(x)/2)
clr <- 1:3+1
typ <- 1:3
wid <- 1:3*2
cha <- 1:3+16

plot (x, sin(x),
      type="n", xlab="", ylab="")
for (i in seq(along=Y))
  lines (x, Y[[i]],
         col=clr[i], lty=typ[i], lwd=wid[i])
par (cex=1.4)

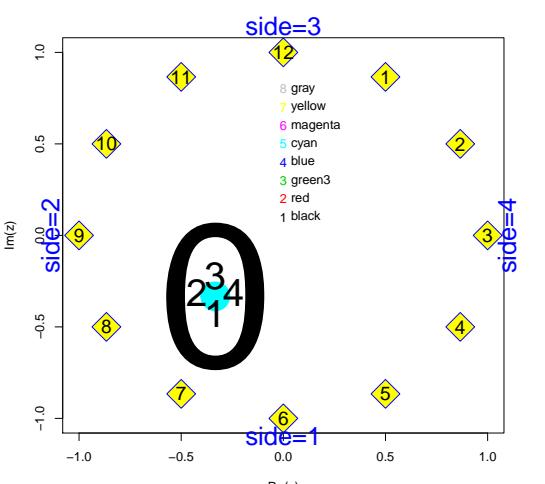
legend ("topleft", legend=txt, fill=clr)
legend ("topright", legend=txt, lty=typ)
legend ("bottomright", legend=txt, lwd=wid)
legend ("bottomleft", legend=txt,
       lwd=2, lty=typ, col=clr)
legend ("right", legend=txt,
       pch=cha, bg=gray(.9))
legend ("left", legend=txt,
       pch=cha, col=clr, lwd=1, bg=gray(1))

for (pos in c("top", "bottom", "center"))
  legend (pos, legend=pos)
```



Textinschrift und Abbildungslegende

Funktionen $\left\{ \begin{array}{l} \text{text (x, y=NULL, labels, pos, cex, ...)} \\ \text{mtext (text, side=3, cex, col, ...)} \end{array} \right\}$



```
z <- exp (2i*pi/12 * 1:12)
plot (Re(z), Im(z),
      pch=23, cex=4,
      col="blue", bg="yellow")
text (z,
      labels=(12:1+1)%/%12+1, cex=1.5)

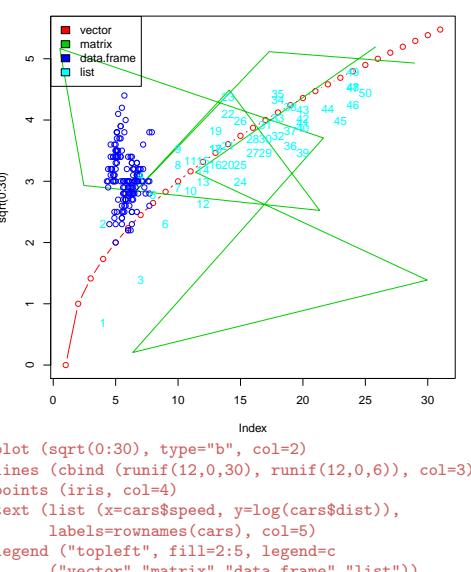
text (x=0, y=1:8/10,
      labels=1:8, col=palette())
text (x=0, y=1:8/10,
      labels=palette(), pos=4)

points (x=-1/3, y=-1/3,
       pch=19, cex=5, col="cyan")
text (x=-1/3, y=-1/3,
      labels=1:4, pos=1:4, cex=3)
text (x=-1/3, y=-1/3,
      labels=0, cex=16)

mtext (
      text=paste ("side",1:4,sep=""),
      side=1:4, col="blue", cex=2)
```

Formulierungsvarianten für 2D-Koordinaten

xy.coords (x, y=NULL, xlab=, ylab=, log=, recycle=FALSE)



```
plot (sqrt(0:30), type="b", col=2)
lines (cbind (runif(12,0,30), runif(12,0,6)), col=3)
points (iris, col=4)
text (list (x=cars$speed, y=log(cars$dist)),
      labels=rownames(cars), col=5)
legend ("topleft", fill=2:5, legend=c
("vector","matrix","data.frame","list"))
```

Schnittstelle x, y-Args

xy.coords()-Aufruf liefert
Liste mit Vektoren **x**, **y** und
Zeichenketten **xlab**, **ylab**.

Kritischer Fall y=NULL

vector:	Index 1:n und x
complex:	Re(x) und Im(x)
matrix:	Spalten 1,2
data.frame:	Spalten 1,2
list:	x\$x und x\$y
ts:	time(x) und x
formula:	Form yvar ~ xvar

Standardfall recycle=F

Die Argumente **x** und **x** müssen
gleiche Länge besitzen!

Mathematik-Layout im L^AT_EX-Stil

'R'-Objekte der Klasse **expression** als Textargumente

Arithmetic Operators	Relations	Juxtaposition
$x + y$	$x + y$	$x == y$
$x - y$	$x != y$	$x = y$
$x * y$	$x < y$	$x < y$
x/y	$x <= y$	$x \leq y$
$x \%+-% y$	$x \pm y$	$x > y$
$x \%/% y$	$x \geq y$	$x \geq y$
$x \%*% y$	$x \sim y$	$x \approx y$
$x \%.% y$	$x \equiv y$	$x \equiv y$
$-x$	$x == y$	$x \equiv y$
$+x$	$x \prop y$	$x \approx y$

Sub/Superscripts
$x[i]$
x^2
Radicals
\sqrt{x}
$\sqrt[n]{x}$
Ellipsis
$list(x[1], \dots, x[n])$
$x[1] + \dots + x[n]$
$list(x[1], \dots, x[n])$
$x[1] + \dots + x[n]$

Set Relations	Arrows
$x \%subset% y$	$x \subset y$
$x \%subseteq% y$	$x \subseteq y$
$x \%supset% y$	$x \supset y$
$x \%supseteq% y$	$x \supseteq y$
$x \%notsubset% y$	$x \not\subset y$
$x \%in% y$	$x \in y$
$x \%notin% y$	$x \notin y$
	$x \%lt% y$
	$x \%gt% y$
	$x \%leq% y$
	$x \%geq% y$
	$x \%eq% y$
	$x \%neq% y$
	$x \%lneq% y$
	$x \%gtrless% y$
	$x \%lneq% y$
	$x \%gtrless% y$

Accents
\hat{x}
\tilde{x}
\bar{x}
\overline{xy}
\widehat{xy}
\widetilde{xy}

Mathematik-Layout im L^AT_EX-Stil

Hilfeseiten mit Kommando **?plotmath**

Symbolic Names	
Alpha – Omega	$A - \Omega$
alpha – omega	$\alpha - \omega$
phi1 + sigma1	$\phi + \varsigma$
Upsilon1	Υ
infinity	∞
32 * degree	32°
60 * minute	$60'$
30 * second	$30''$

Style
<code>displaystyle(x)</code>
<code>textstyle(x)</code>
<code>scriptstyle(x)</code>
<code>scriptscriptstyle(x)</code>
Spacing
$x \sim y$
$x + phantom(0) + y$
$x + over(1, phantom(0))$
$\frac{1}{x + -}$
Fractions
$\frac{x}{y}$
$\overline{x}{y}$
$atop(x, y)$

Big Operators
$\sum_i^n x_i$
$\prod_x P(X == x)$
$\int_a^b f(x) dx$
$\bigcup_{i=1}^n A_i$
$\bigcap_{i=1}^n A_i$
$\lim_{x \rightarrow 0} f(x)$
$\min_{x \geq 0} g(x)$
$\inf S$
$\sup S$

Grouping
$(x + y) * z$
$(x + y)z$
$x^y + z$
x^{y+z}
x^{y+z}
$group("(", list(a, b), ")")$
$(a, b]$
$bgroup("(", atop(x, y), ")")$
$\binom{x}{y}$
$[x]$
$ x $
$ x $

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

Zeichenketten — spezielle Funktionen

Datensätze — spezielle Funktionen

Eingabefunktionen & Ausgabefunktionen

'R'-Programmpakete — installieren & laden

Grundlegende Mechanismen der 'R'-Grafikfunktionen

Univariate und bivariate Darstellungen

Räumliche und vieldimensionale Darstellungen

Das Trellis-Grafiksubsystem

Histogramme — Ereignishäufigkeiten in Schubladen

hist (x, breaks, freq, density, angle, x/ylim, axes, plot, labels, ...)

Histogram of log(USairpollution\$popul)

Schubladengeometrie
± äquidistante Grenzen

Parameter 'breaks'

Grenzen numeric(k)
Anzahl integer(1)
Methode function
Strategie character(1)
Sturges-Algorithmus default

Ergebnisobjekt

list (breaks,
counts, density)

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

Box-Whisker-Diagramm

boxplot (x, ..., notch, out, names, log, horizontal, add, ...)

US air pollution

Übersichtsstatistik

Taille	Median
Kasten	($\frac{1}{4}, \frac{3}{4}$)-Quantile
Strichel	Wertebereich*
Kringel	Ausreißer
Kerbe	Signifikanz

Datenargumente

- ein Vektor x
- mehrere Vektoren x, ...
- eine Matrix x
- ein Datensatz x
- Modellformel data

```
require (HSAUR2)
boxplot (USairpollution, log="y", notch=TRUE,
         col=cm.colors (length (USairpollution)), main="US air pollution")
```

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

Histogramme — Ereignishäufigkeiten in Schubladen

hist (x, breaks, freq, density, angle, x/ylim, axes, plot, labels, ...)

Histogram of log(USairpollution\$popul)

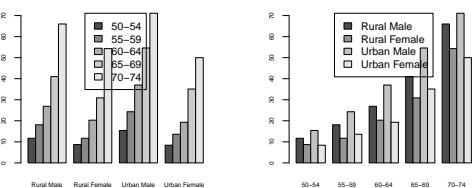
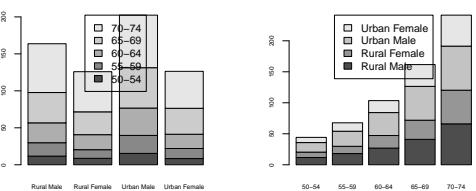
Histogram of rnorm(1000)

hist (log (USairpollution\$popul), col="blue", labels=TRUE, xlab="")
hist (rnorm (1000), col="red", breaks=32, xlab="")

?character ?data.frame E/A Pakete Grafik 1D/2D-Plots 3D/HD-Plots lattice

Balkendiagramm für Vektor- und Matrixdaten

barplot (height, width=1, legend.text, beside, horiz, col, axes, ...)



```
for(tb in list (VADeaths, t(VADeaths)))
  for (bs in c(FALSE,TRUE))
    barplot (height=tb, legend=TRUE, beside=bs, args.legend=list(x="top"))
```

Eingabedaten
Zahlenmatrix/vektor
benannte Zeilen
benannte Zeilen
positive Werte (s.u.)
moderate Anzahl

Balkengrafik

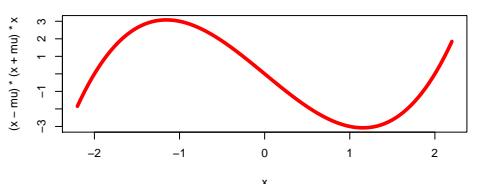
Hochstapler beside=FALSE

Orgelpfeifen beside=TRUE

VADeaths			
RM	RF	UM	UF
11.7	8.7	15.4	8.4
18.1	11.7	24.3	13.6
26.9	20.3	37.0	19.3
41.0	30.9	54.6	35.1
66.0	54.3	71.1	50.0

Kurvenverlauf (Ausdrucksobjekt)

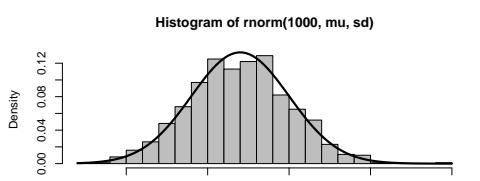
curve (expr, from, to, n=101, add=FALSE, type='l', ylab, log, ...)



Ausdruck in 'x'

Analytisch \rightsquigarrow Stützwerte:

```
x <- seq (from,to,len=n)
y <- eval (expr)
```



Sekundärfunktion

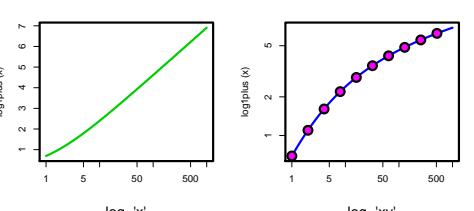
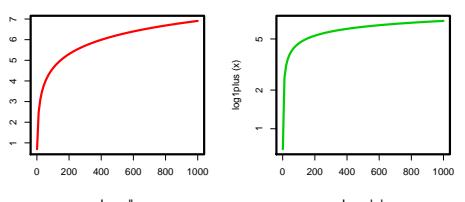
Setze add=TRUE
(Koordinatenübernahme)

```
mu=2; sd=3;
curve ((x-mu)*(x+mu)*x, from=-1.1*mu, to=+1.1*mu, col="red", lwd=5)

hist (rnorm (1000,mu,sd), freq=FALSE, col="gray", breaks=32, xlab="")
curve (dnorm (x,mu,sd), add=TRUE, lwd=3)
```

Kurvenverlauf (Funktionsobjekt)

curve (x, y=0, to=1, from=y, xlim=NULL, ...) oder gleichwertig plot (< dto.>)



```
for (ll in c("", "x", "y", "xy"))
  curve (log1plus, 1, 1000, log=ll, col=2+nchar(ll), xlab="",
         sub=paste ("log=", ll, "", sep=""))
z <- 2^(0:9);   points (z, log1plus(z), pch=21, cex=2, bg="magenta")
```

Funktionsobjekt

mit einem Argument:
log1plus <- function(a)
 log(1+a)

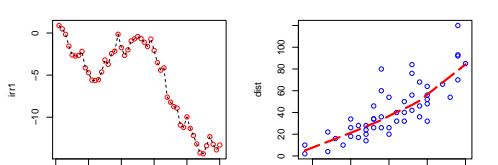
Logarithm. Achsen

Argument log = {
'x'
'y'
'xy'}

Skalierungsweise ist transparent für sukzessives Zeichnen!

Punkte & Linien in der Koordinatenebene

points (x, y=NULL, type='p', pch, cex, bg, ...) oder
lines (x, y=NULL, type='l', lwd, lty, ...)

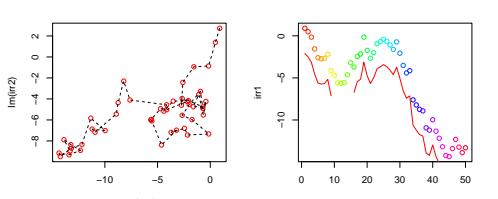


Punkte

Attributwechsel Punkt zu Punkt

Linien

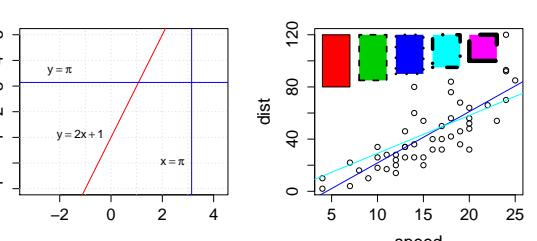
Unterbrechungen via NA
kein Attributwechsel



```
x <- rnorm(50); y <- rnorm(50); irr1 <- cumsum (x); irr2 <- cumsum (x + i*i*y)
plot (irr1, col="red"); lines (irr1, lty=2)
plot (irr2, col="red"); lines (irr2, lty=2)
plot (cars, col="blue"); lines (lowess (cars), col="red", lwd=3, lty="82")
plot (irr1, col=rainbow(50)); irr1[10:15] <- NA; lines (irr1-3, col=rainbow(50))
```

Geraden und Rechtecke

abline (a, b, h, v, reg, coef, ...) oder
rect (xleft, ybottom, xright, ytop, col, border, lty, lwd, ...)



```
plot (c(-2,3), c(-1,5), type="n", xlab="", ylab="", asp=1)
abline (h=-1:5, v=-2:3, col="lightgray", lty=3)
abline (a=1, b=2, col="red")
abline (h=pi, v=pi, col="blue")

attach (cars)
plot (cars)
abline (lsfit (speed, dist), col="blue")
abline (lsfit (speed, dist, intercept=FALSE), col="cyan")

N <- 1:5
rect (xleft=0+4*N, ybottom=75+5*N, xright=3+4*N, ytop=120,
      col=1:N, lty=N, lwd=N)
```

Geradenspezifikation

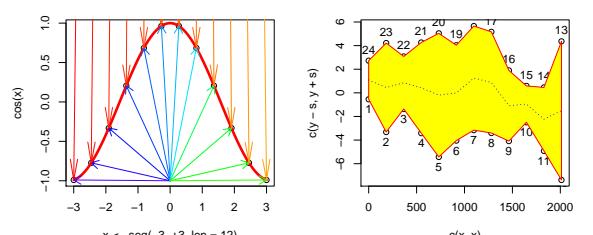
Koeffizienten a und b
Horizontale h
Vertikale v
Linearmodell reg
Koeffiz.vektor coef
 $(\geq 1$ Geraden ok für h/v)

Boxenspezifikation

links unten xleft ybottom
rechts oben xright ytop
Füllfarbe(n) col
Randfarbe(n) border
Randattribut(e) lty lwd

Pfeile und Polygone

arrows (x0, y0, x1=x0, y1=y0, length, angle, code, ...) oder
polygon (x, y, density, angle, border, col, lty, ...)



```
plot (x <- seq (-3, +3, len=12), cos(x))
curve (cos, add=TRUE, col="red", lwd=3)
arrows (0, -1, x, cos(x), length=0.1, col=topo.colors(24))
arrows (0, +100, x, cos(x), angle=10, col=heat.colors(24))

x <- seq (0, 2012, len=(n<-12))
y <- cumsum (rnorm (n))
s <- abs (rnorm (n,3))
plot (c(x,x), c(y-s,y+s))
polygon (c(x,rev(x)), c(y-s,rev(y+s)), col="yellow", border="red")
lines (x, y, lty=3, col="blue")
```

Pfeilparameter

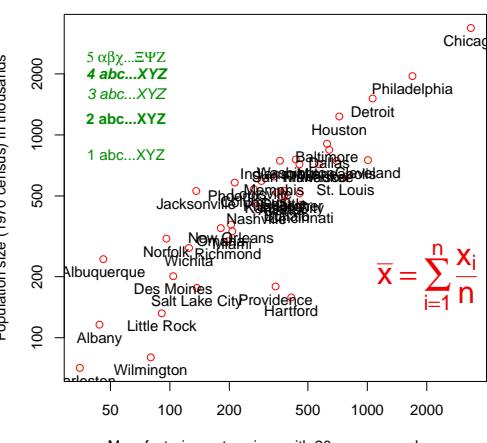
Startposition	x0 y0
Zielposition	x1 y1
Pfeilgeometrie	length angle
Pfeiltyp	{ \leftarrow , \rightarrow , \leftrightarrow } code {1, 2, 3}

Polygonparameter

Ankerpunkte	x y
Schraffur	density angle
Füllfarbe	col
Randfarbe	border
Randattribute	lty lwd
Füllstrategie	fillOddEven

Textmarkierungen in der Koordinatenebene

text (x, y, labels, adj, pos, offset, vfont, cex, col, font, ...)



Textposition
pos, adj, und offset

Schriftattribut
cex, col und font

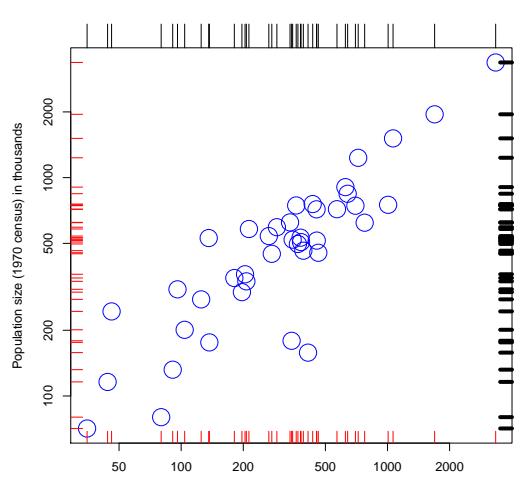
1	= normal
2	= fettig
3	= kursiv
4	= beides
5	= symbol

Texte abkürzen!
abbreviate (names, min=4, strict=FALSE)

```
plot (popul ~ manu, data=USairpollution, xlab=mlab, ylab=plab, log="xy", col="red")
text (popul ~ manu, data=USairpollution, pos=1, labels=rownames (USairpollution))
text (rep(60,5), 400*2:6, labels=paste (1:5, "abc...XYZ"), font=1:5, col="green4")
text (2000, 200, cex=2, col="red2", labels=expression (bar(x) == sum(frac(x[i], n), i==1, n)))
```

Stützstellenmarkierungen an Koordinatenachse

rug (x, ticksize=0.03, side=1, lwd=0.5, col=par('fg'), quiet=, ...)



Achsenwahl

1=unten 2=links 3=oben
4=rechts

Koordinaten

Übernahme, auch log/log

Markierungsgrößen

Papierskala, nicht Datenskala

Doubletten schütteln!

jitter (x,
factor=1, amount=NULL)

Zeichenketten — spezielle Funktionen

Datensätze — spezielle Funktionen

Eingabefunktionen & Ausgabefunktionen

'R'-Programmpakete — installieren & laden

Grundlegende Mechanismen der 'R'-Grafikfunktionen

Univariate und bivariate Darstellungen

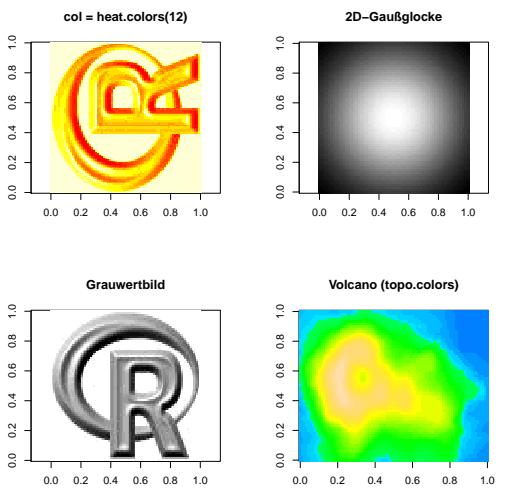
Räumliche und vieldimensionale Darstellungen

Das Trellis-Grafiksubsystem

```
plot (popul ~ manu, data=USairpollution, xlab=mlab, ylab=plab, log="xy", col="blue", cex=3)
rug (USairpollution$manu, side=1, col="red");
rug (USairpollution$manu, side=3, ticks=-0.06)
rug (USairpollution$popul, side=2, col="red");
rug (USairpollution$popul, side=4, lwd=5)
```

Rastergrafik für $f(x, y)$

image (x, y, z, zlim, xlim, ylim, col=heat.colors(12), add, breaks, ...)



Geometrie

Definitionsbereich x
Wertematrix y
Pixel quadratisch $asp=1$
Gebietsgrenzen $xlim$ $ylim$ $zlim$
neue Koordinaten? $add=FALSE$

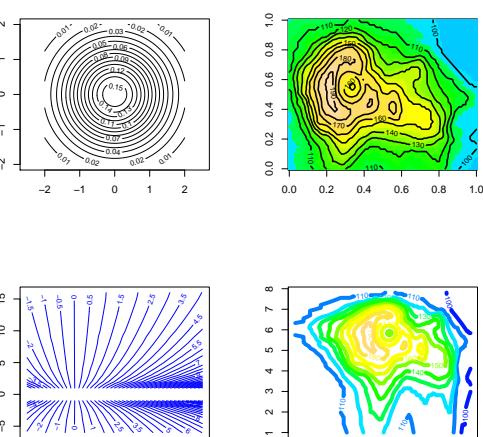
Färbung

Farbpalette col
Farbgrenzen $breaks$

```
library (rimage); data (logo);
image (logo, asp=1, main="col = heat.colors(12)")
image (t(logo[nrow(logo):1,]), asp=1, col=gray (0:20/20), main="Grauwertbild")
x <- dnorm (-30:+30/20)
image (z=x%/%x, col=gray (0:50/50), asp=1, main="2D-Gaußglocke")
image (volcano, col=topo.colors(48)[-1:10], main="Volcano (topo.colors)")
```

Konturgrafik für $f(x, y)$

contour (x, y, z, nlevels=10, levels, labels, method, col, lty, add, ...)



Geometrie

Definitionsbereich x
Wertematrix y
Anzahl Schichten $nlevels$
Konturhöhenvektor $levels$
Gebietsgrenzen $xlim$ $ylim$ $zlim$
neue Koordinaten? $add=FALSE$

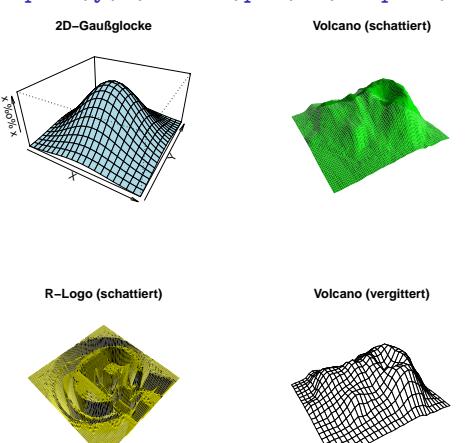
Beschriftung

Konturmarkierungen $labels$
Schriftattribute $vfont$ $labcex$
Linienattribute col lty lwd

```
x <- seq (-2, +2, len=100); contour (x=x, y=y, z=outer (dnorm(x), dnorm(y)), nlevels=12, asp=1)
x <- -6:16; z <- outer(x, sqrt(abs(x)), FUN="/"); contour(x,x,z, col="blue", method="edge", nlev=32)
image (z=volcano, col=topo.colors(32)[-1:10]); contour (z=volcano, add=TRUE, lwd=2)
contour (x=sqrt(1:nrow(volcano)), y=sqrt(1:ncol(volcano)), z=volcano, col=topo.colors(12), lwd=5)
```

Perspektivgrafik für $f(x, y)$

persp (x,y,z, theta,phi, r, expand, col, border, shade, box, ...)



Geometrie

Definitionsbereich x
Wertematrix y
Azimuthwinkel $theta$
Colatitudewinkel phi
Auge-Box-Distanz r
Aspekttreue $scale=FALSE$
z-Expansion $expand$
Gebietsgrenzen $xlim$ $ylim$ $zlim$

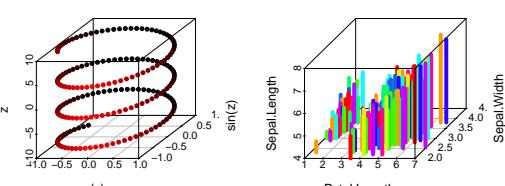
Licht und Farbe

Facettenfarbe(n) col
Gitterfarbe(n) $border$
Schattierungskoeff. $shade$
Lichtquelle $ltheta$ $lphi$
3D-Beleuchtung? box

```
x <- dnorm (-10:+10/5)
persp (z=x%/%x, theta=30, phi=30, expand=0.5, col="lightblue", main=
persp (z=sqrt(logo), theta=45, phi=65, shade=0.75, col="yellow", border=NA, box=FALSE, main=
persp (z=volcano, theta=150, phi=30, expand=0.4, col="green", shade=0.75, box=F, border=NA, main=
persp (z=volcano [1:nrow(volcano)%3==0, 1:ncol(volcano)%3==0],
      theta=150, phi=30, expand=0.4, box=FALSE, main=
```

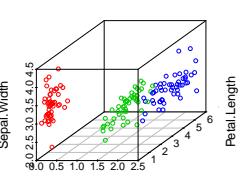
3D Scatterplot für (x, y, z) -Punktwolken

scatterplot3d (x,y,z, color, highlight.3d, x/y/zlim, x/y/zlab, scale, angle, grid, box, type, col.*., cex.*., font.*., lty.*., ...)



Geometrie

Punktwolke x y z
 (x, y) -Winkel $angle$
y-Achsenfaktor $scale.y$
Gebietsgrenzen $xlim$ $ylim$ $zlim$



```
z <- seq (-10, +10, 0.1); scatterplot3d (x=cos(z), y=sin(z), z=z, pch=19, highlight.3d=TRUE)
scatterplot3d (iris[4:2], color=1+unclass(iris$Species))
scatterplot3d (iris[3:1], type="h", pch=" ", lwd=5, color=rep(rainbow(10),15))
rb <- rainbow (151); rb3 <- t (col2rgb (rb)); scatterplot3d (rb3, color=rb, box=FALSE, axis=F) -> geo
tp <- cm.colors (151); tp3 <- t (col2rgb (tp)); geo$points3d (tp3, pch="X", col=tp)
```

Beschriftung

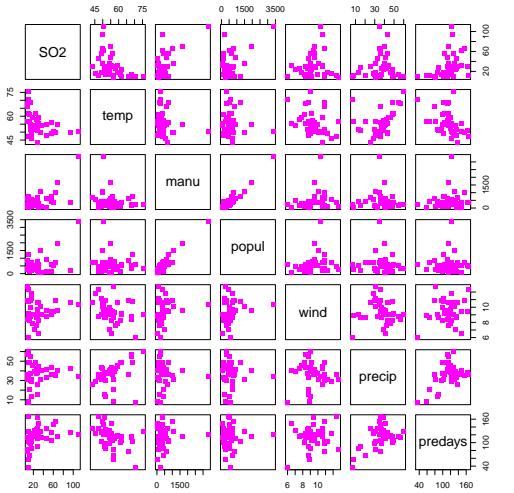
Punktfarbe(n) $color$
3D-Effekt $highlight.3d$
Achsen etc.? $axis$ $grid$ box

Sekundärfunktionen

in der Rückgabeliste:
points3d, plane3d, box3d, xyz.convert.

Scatterplot: $n \times n$ 2D-Wolken

```
pairs (x, labels, panel, ..., cex.labels, font.labels, gap=1, ...)
```



pairs-Parameter

Matrix/Datensatz **x**
Variablennamen **labels**
Zwischenraum **gap**
points-Argumente **...**

USairpollution

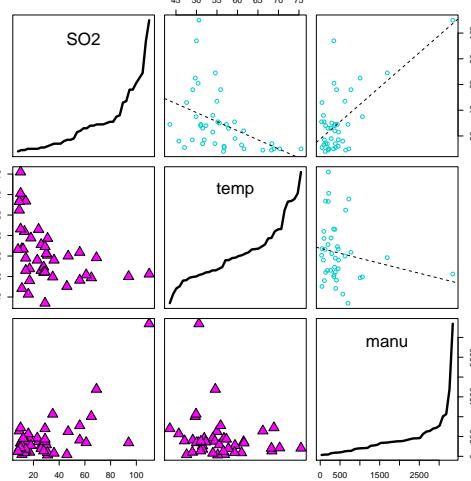
Luftverschmutzung in 41 US-Metropolen

SO_2 -Gehalt in $\mu g/m^3$
 \varnothing -Temperatur in deg F
Firmen (≥ 20 Mitarb.)
Einwohner (Tsd., 1970)
Windgeschwindigkeit mph
Niederschlag inches
Regentage 1/J

```
library (HSAUR2)
pairs (USairpollution, cex=8, pch=". ", col="magenta")
```

Scatterplot mit individuellem Layout

```
pairs (x, labels, panel, lower.panel, upper.panel, diag.panel,
text.panel, ...)
```

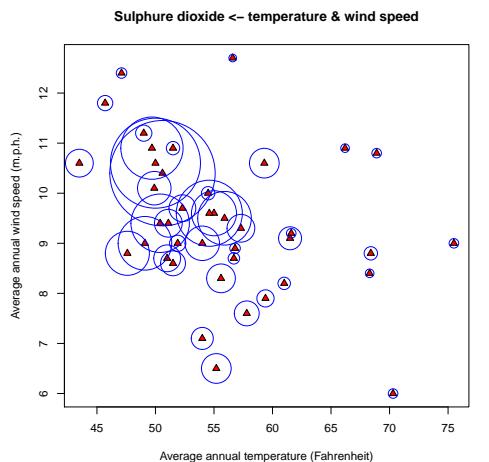


library (HSAUR2)

```
pairs (USairpollution[c(1,2,3)],
lower.panel=points,
upper.panel=function (x,y,...)
{
  points (x, y, col="cyan3")
  abline (lsfit(x,y), lty=2)
},
diag.panel=function (x,...)
  lines (
    seq (
      min(x),
      max(x),
      len=length(x)
    ),
    sort(x), lwd=3, ...
  ),
cex=2, pch=24, bg="magenta")
```

Ikonische 3D-Darstellung: „bubble plot“

```
symbols (x, y, circles, squares, inches, add=FALSE, fg, bg, ...)
```



Dritte Dimension

als Symbolgröße kodiert
 $(z \geq 0)$

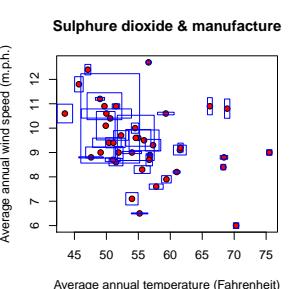
Funktionsargumente

Punktepositionen **x y**
z-Koordinate **circles**
oder **squares**
Maximalgröße **inches**
(automatisch) **inches=FALSE**
Berandungsfarbe **fg**
Hintergrundfarbe **fg**

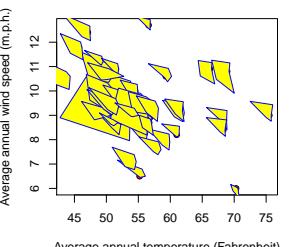
```
with (USairpollution,
plot (temp, wind, main=slab, xlab=tlab, ylab=wlab, pch=24, bg="red"))
with (USairpollution,
symbols (temp, wind, add=TRUE, circles=S02, inches=0.75, fg="blue"))
```

Ikonische Darstellung von $\geq 3D$ -Punktwolken

```
symbols (x, y, rectangles, stars, thermometers, boxplots, ...)
```



Sulphure dioxide & manufacture



SO2 & precip & predays & manu & temp

Zusatzdimensionen
als Symbolattribute kodiert
 $(z_i \geq 0)$

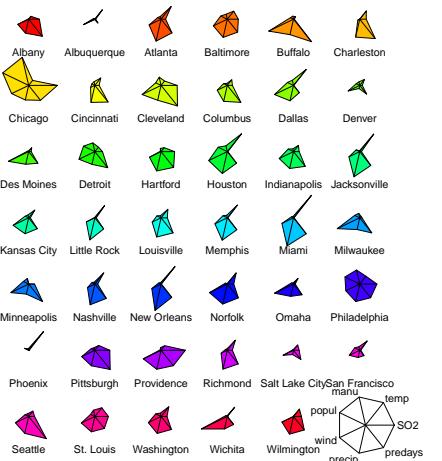
Geometrie

Punktepositionen **x y**
 z_1, \dots, z_2 **rectangles**
 z_1, \dots, z_3 **thermometers**
 z_1, \dots, z_4 **thermometers**
 z_1, \dots, z_5 **boxplots**
 z_1, \dots, z_3, \dots **stars**

Alle z_i -Argumente als Matrizen
oder Datensätze passender
Dimension

Sterngrafik

```
stars (x, full, scale, radius, labels, location, nrow, ncol, len,
      key.loc, draw.segments=FALSE, col.segments, col.stars, ...)
```



Funktionsargumente

Matrix, Datensatz	x
Voll/halbkreis?	full
Radiallinien?	radius
Objektnamen	labels
Legendenposition	key.loc
Attributnamen	key.labels
Attributfarben	col.segments
Objektfarben	col.stars
Skalenfaktor	len

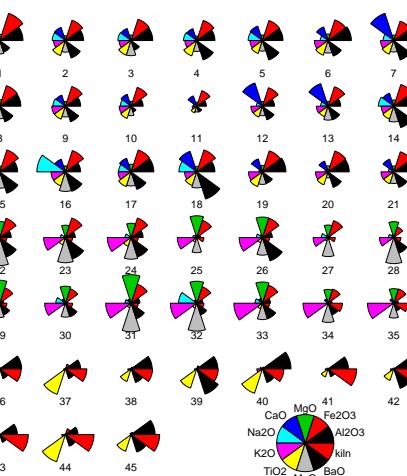
Objektgeometrie

2D-Positionen	locations
Gitterdimensionen	nrow ncol

```
stars (USairpollution, draw.segments=FALSE,
       col.stars=rainbow(41), key.loc=c(14,2), flip.labels=FALSE, cex=0.8)
```

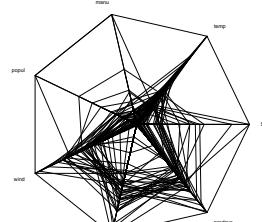
Radargrafik und Spinnennetz

```
stars (x, full, scale, radius, labels, location, nrow, ncol, len,
      key.loc, draw.segments=TRUE, col.segments, col.stars, ...)
```



Funktionsargumente

Matrix, Datensatz	x
Attributfarben	col.segments
Spinnennetz	loc=c(0,0)

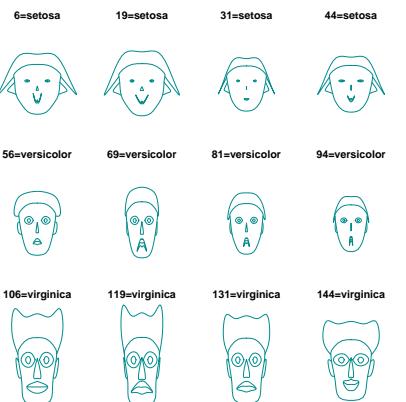


```
stars (USairpollution,
       draw.segments=FALSE,
       locations=c(0,0), key.loc=c(0,0))
```

```
require (HSAUR2); stars (pottery, draw.segments=TRUE, key.loc=c(13,2))
```

Chernoff-Gesichter: physiognomische Darstellung

```
faces (xy, fill=FALSE, nrow, ncol, scale=TRUE, labels)
```



Chernoff-Merkmale

1-height of face · 2-width of face ·
 3-shape of face · 4-height of mouth ·
 5-width of mouth · 6-curve of
 smile · 7-height of eyes · 8-width of
 eyes · 9-height of hair · 10-width of
 hair · 11-styling of hair · 12-height
 of nose · 13-width of nose ·
 14-width of ears · 15-height of ears.

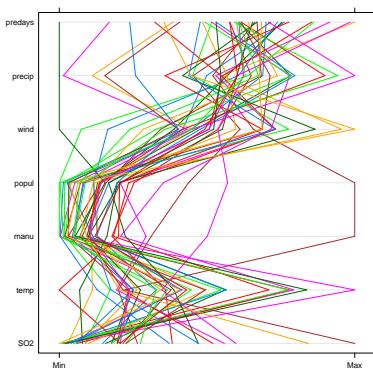
Funktionsargumente

Numerische Datenmatrix	xy
Gitterlayout	ncol nrow
Attribute normieren?	scale
Attribute ~ PCA?	fill

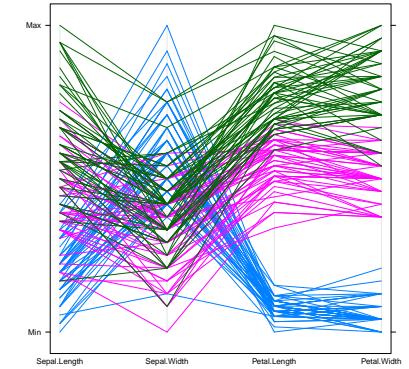
```
require (TeachingDemos)
N=150; M=4; K=3; idx <- round ((1:(K*M)-.5)*N/K/M)
lab <- paste (idx, iris$Species[idx], sep="=")
faces (iris[idx,-5], nrow=K, ncol=M, labels=lab, fill=FALSE)
```

Wäscheleinengrafik: parallele Koordinaten

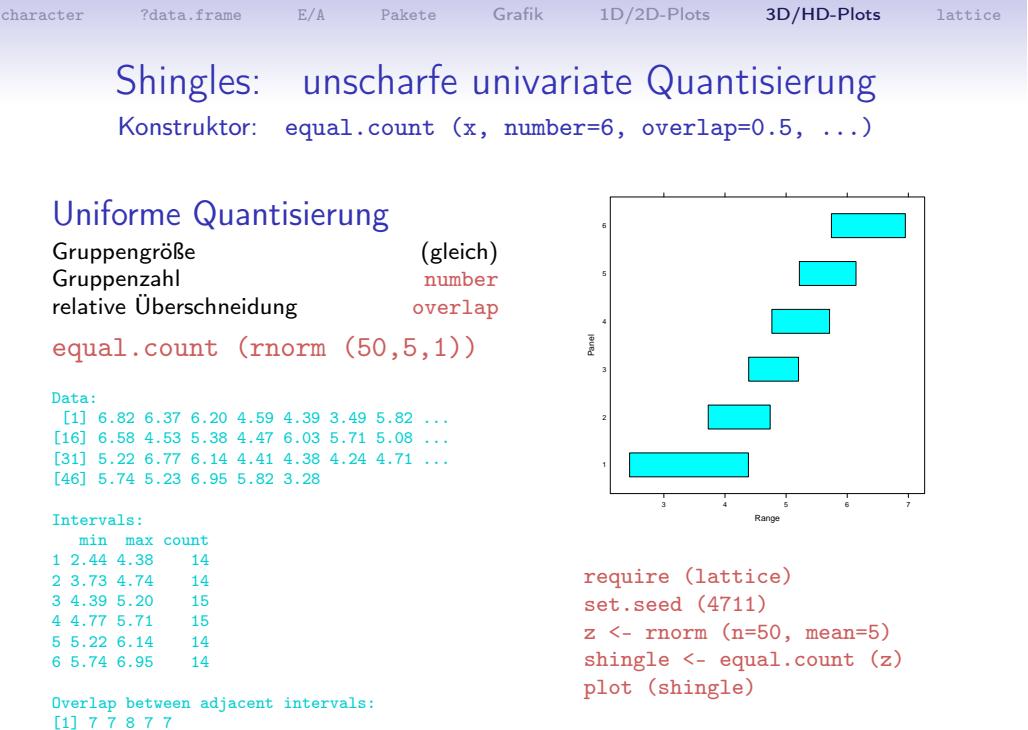
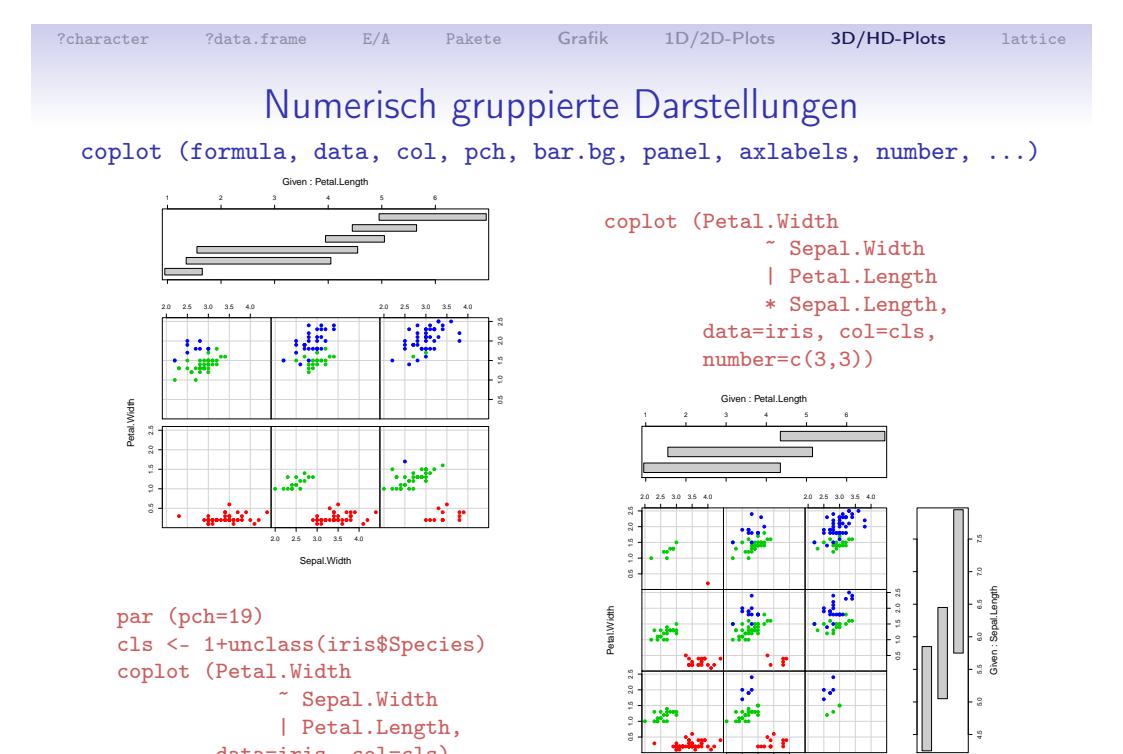
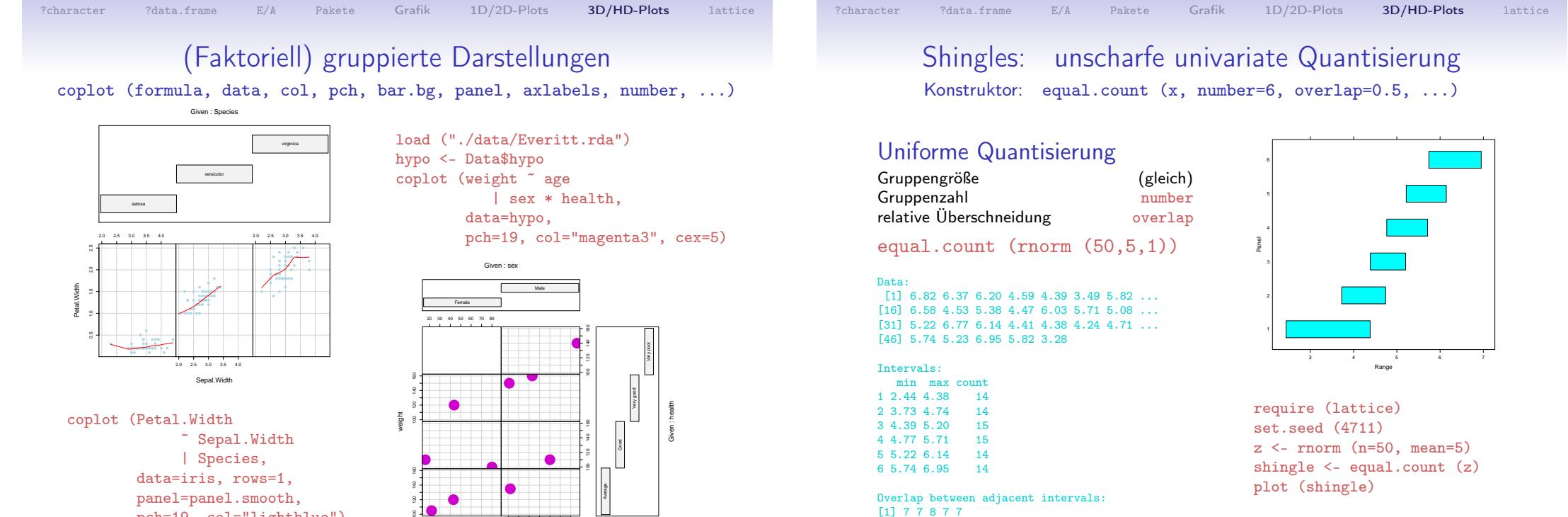
```
parallel (x, data=NULL, groups=NULL, subset=TRUE, ...)
```



```
require (lattice)
parallel (x=iris[1:4],
          groups=iris$Species,
          horizontal.axis=FALSE
        )
```



```
require (lattice)
require (HSAUR2)
parallel (x=USairpollution)
```



- Zeichenketten — spezielle Funktionen
- Datensätze — spezielle Funktionen
- Eingabefunktionen & Ausgabefunktionen
- 'R'-Programmpakete — installieren & laden
- Grundlegende Mechanismen der 'R'-Grafikfunktionen
- Univariante und bivariate Darstellungen
- Räumliche und vieldimensionale Darstellungen
- Das Trellis-Grafiksubsystem

Zusammenfassung (3)

1. Zeichenketten sind **atomare** 'R'-Typen; es gibt also keine expliziten Einzelzeichen.
2. **Datensätze** werden durch die Klasse `data.frame` repräsentiert.
Zeilen enthalten die **Objekte** (Fälle) und Spalten die numerischen, nominalen oder ordinalen **Attribute** (Variable, Merkmale).
3. Die **E/A** geschieht mit `save()` und `load()` (**komplexe** Typen) oder mit `write()` und `scan()` (**atomare** Typen).
4. **Grafiken** werden **sukzessiv** erzeugt; ein Primäraufruf baut (u.a.) das **Koordinatensystem**, nachfolgende Aufrufe vervollständigen die Grafik mit Punkten, Linien, Flächen, Texten und Ikonen.
5. Mehrere Grafiken können auf einer **Leinwand** arrangiert werden.
6. Wir unterscheiden Darstellungstechniken für **Verteilungseigenschaften** (z.B. Histogramme), für **Punktwolken** (Scatterplots) und für **Funktionsverläufe** (Linien, Flächen).
7. Die Visualisierung **Höherdimensionale** Daten ($dim \geq 4$) erfolgt durch **Ikonenbildung** (Sterne, Gesichter), **Staffelung** (Wäscheleinen) oder **Schachtelung** (`pairs()`, `coplot()`).