

Einführung in R & Prädiktive Analyse-Demo

Thomas Schübel

SQLPASS Regionalgruppe Franken
Nürnberg, 20.2.2018

<http://github.com/schuebelt/sqlpass>

Vorstellung des Referenten

- Akademischer Hintergrund, zuletzt Uni Konstanz
- Langjährige Erfahrung mit prädiktiven Analyseverfahren (z.B. zur Vorhersage von Wahlverhalten)
- Fertigstellung der Dissertation 2017
- Lust auf eine neue Herausforderung im unternehmerischen Data Science-Umfeld

Aufbau der Präsentation

1 R-Grundlagen

2 Predictive Analytics-Beispiel

3 Weitere Aspekte

R-Grundlagen

Grundlagen

- 1992 (auf Basis von S) an der Uni Auckland von den Statistikern Ross Ihaka und Robert Gentleman entwickelt
- Motivation: Entwicklung einer Sprache zur Datenanalyse, welche leistungsfähiger und gleichzeitig nutzerfreundlicher ist als verfügbare Sprachen
- „R Core Team“ besteht heute aus ca. 20 Personen
- Anfänglich v.a. in der (akademischen) Forschung populär, insb. seit etwa 10 Jahren aber auch im Enterprise-Umfeld
- GNU General Public License
- R mit Grundfunktionen (via R-Project, gratis), R Client für SQL-Konnektivität (via Microsoft, gratis)
- **R-Pakete** (gratis)

Bedeutung

Abbildung 1: R (blau) vs. SAS-Skills (orange) in Stellenangeboten



Quelle: <http://i1.wp.com/r4stats.com/wp-content/uploads/2017/02/Fig-1c-R-v-SAS-2017-02-18.png>

Vorteile von R

- Vielzahl (> 10000) von Erweiterungspaketen bieten Flexibilität
 - geprüft (via CRAN)
 - im Entwicklungsstatus (via GitHub)
- Visualisierungsmöglichkeiten
 - ggplot2 (insb. facetting)
 - shiny (Dashboards)
 - maps (Geomapping)
- Data Wrangling
 - plyr, dplyr (vielfältige Join-Möglichkeiten)
 - lubridate (Analysen mit zeitlichen Daten)
- R-Unterstützung in Microsoft-Produkten seit 2015 (Power BI, SQL Server, Azure ML Studio)

IDEs und GUIs

- Code-Unterstützung in MS Produkten (vgl. oben) und Data Mining-Plattformen (KNIME, RapidMiner)
- IDEs
 - **RStudio Desktop** als populärste IDE, RStudio Server (kostenpflichtig)
 - Alternativen
 - Visual Studio
 - Jupyter Notebook
 - Notepad++
- GUIs
 - RCommander
 - Alternativen
 - RKWard
 - Sciviews-R

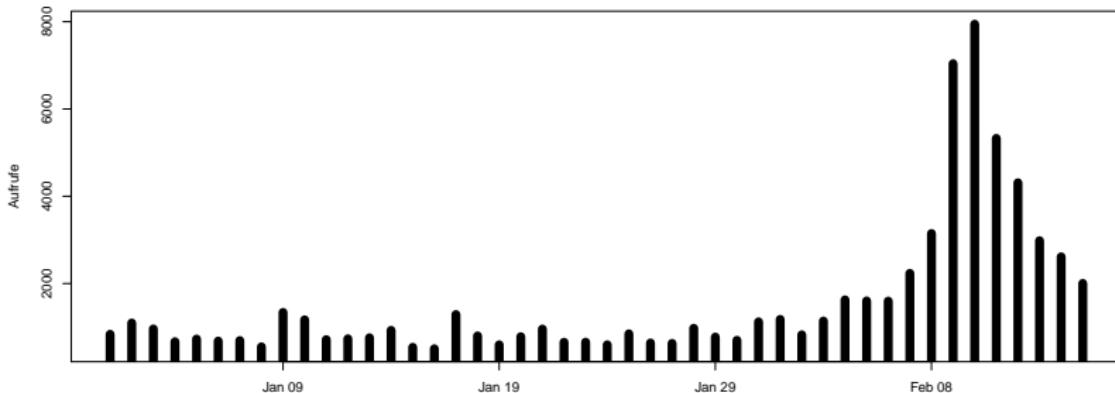
Web Scraping mit htmltab

```
require(htmtab)
u <- "http://www.sportschau.de/fussball/bundesliga2/spieltag/index.html"
htmtab(u,2)[c(1:11),]
```

##	R	V	Verein	Sp	S	U	N	Tore	TD	P
## 2	1	(1)	1. FC Nürnberg	23	13	5	5	46:27	+19	44
## 3	2	(2)	Fortuna Düsseldorf	24	13	5	6	38:30	+8	44
## 4	3	(3)	Holstein Kiel (N)	23	9	10	4	43:31	+12	37
## 5	4	(6)	MSV Duisburg (N)	24	10	7	7	37:36	+1	37
## 6	5	(8)	SSV Jahn Regensburg (N)	24	11	3	10	39:34	+5	36
## 7	6	(4)	SV Sandhausen	24	10	5	9	27:21	+6	35
## 8	7	(10)	1. FC Union Berlin	24	9	7	8	41:33	+8	34
## 9	8	(5)	Arminia Bielefeld	24	9	7	8	38:35	+3	34
## 10	9	(7)	FC Ingolstadt 04 (A)	24	9	6	9	33:26	+7	33
## 11	10	(13)	Dynamo Dresden	24	9	5	10	33:35	-2	32
## 12	11	(9)	1. FC Heidenheim	23	9	5	9	34:39	-5	32

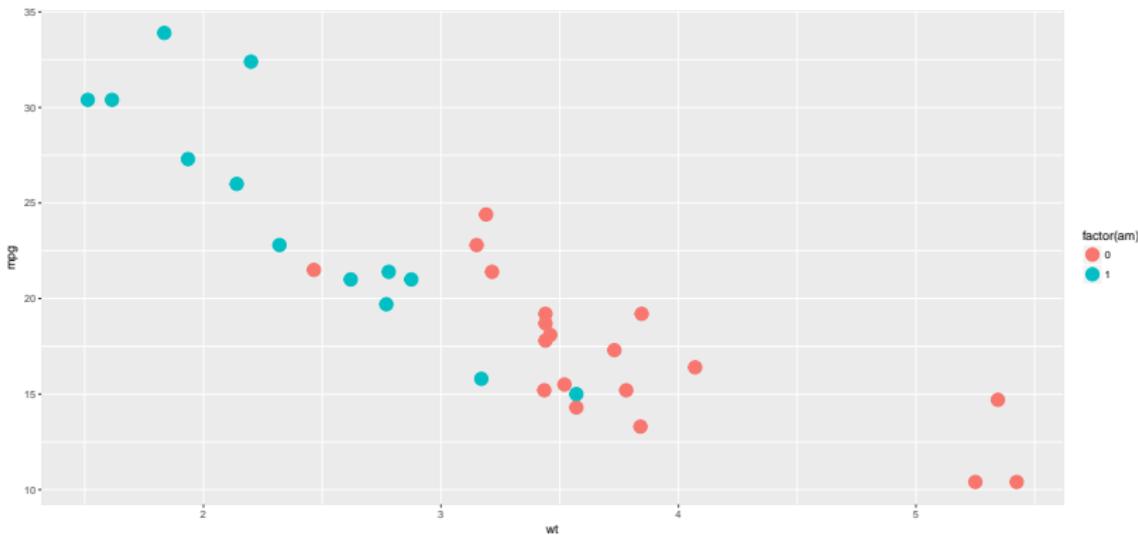
Web Media Trends mit pageviews

```
require(pageviews)
w <- article_pageviews(project = "de.wikipedia",
  article = "Pjöngjang", start = as.Date('2018-01-01'),
  end = as.Date("2018-02-15"))
plot(w$date,w$views,type="h",xlab="Datum",ylab="Aufrufe",lwd=10.5)
```



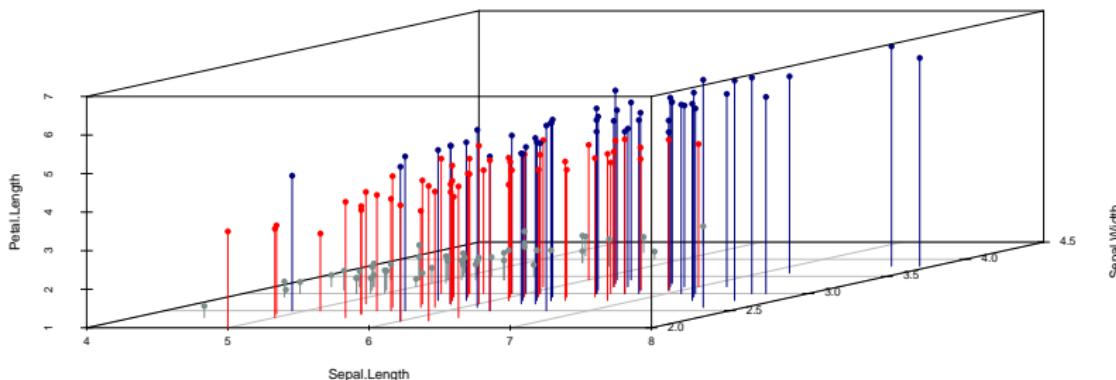
Flexible Streudiagramme mit ggplot2 I

```
require(ggplot2)
ggplot(mtcars, aes(wt,mpg)) +
  geom_point(aes(colour = factor(am)),size=5)
```



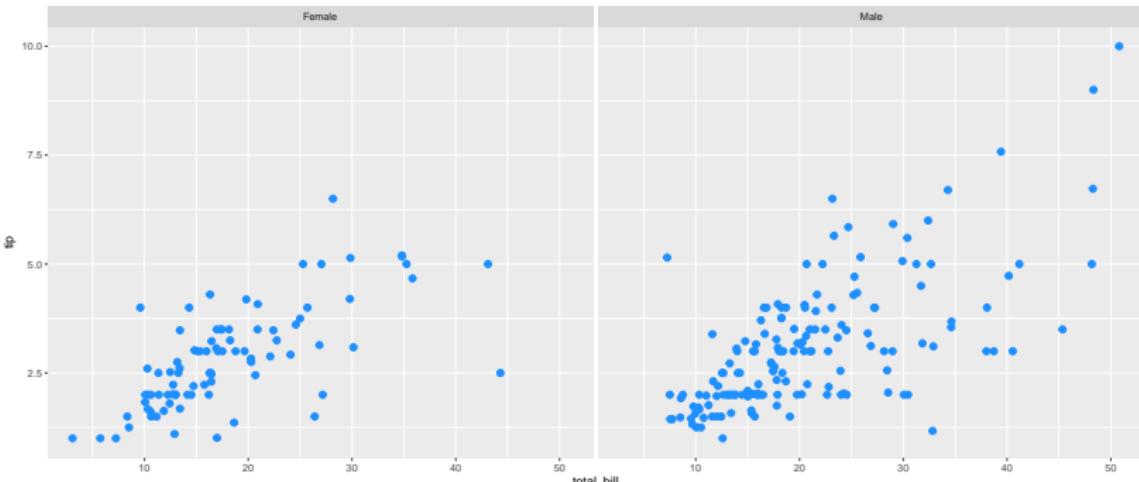
3D-Streudiagramme mit scatterplot3d

```
require(scatterplot3d)
colors <- c("lightcyan4", "red", "navy")
colors <- colors[as.numeric(iris$Species)]
scatterplot3d(iris[,1:3], pch = 16, color=colors, type="h")
```



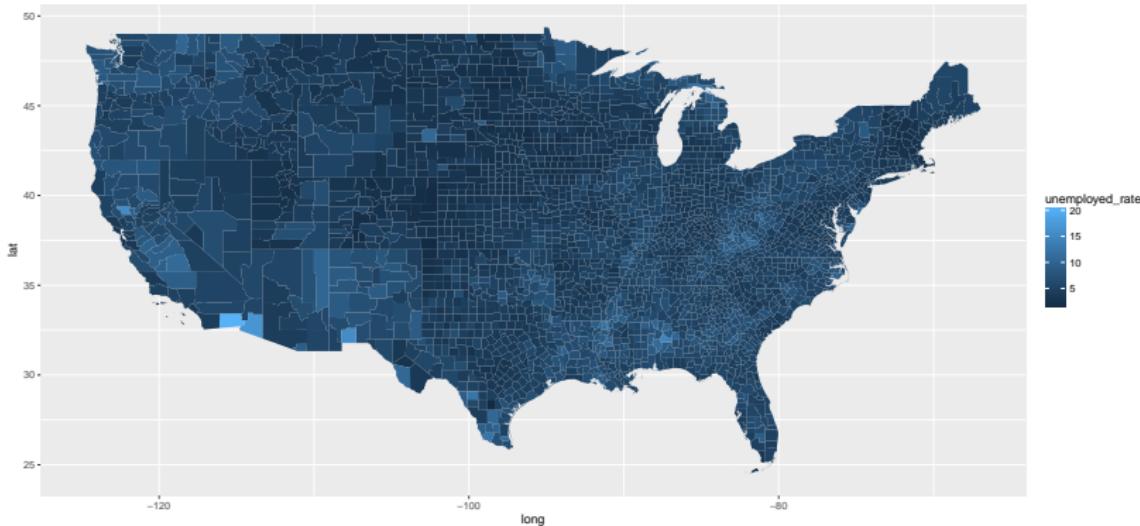
Flexible Streudiagramme mit ggplot2 II

```
require(reshape2)
ggplot(tips, aes(total_bill,tip)) +
  geom_point(size=2.5, color="dodgerblue") +
  facet_grid(. ~ sex)
```



Geomapping mit ggplot2

```
require(ggplot2)
map <- url("http://sharpsightlabs.com/wp-content/datasets/unemployment_map_data_2016_nov.RData")
load(map)
ggplot() +
  geom_polygon(data = map$county_unemp, aes(x = long, y = lat, group = group, fill = unemployed_rate))
```



Predictive Analytics-Beispiel

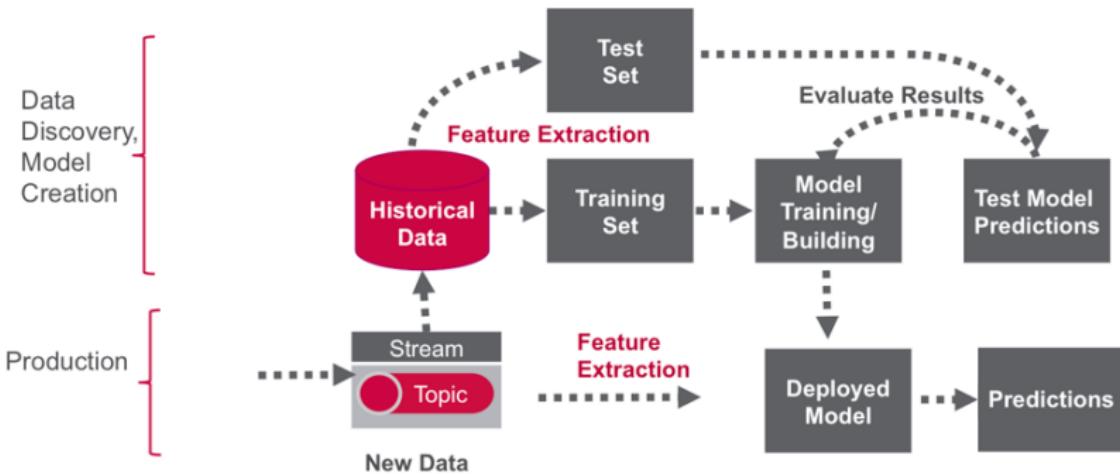
Szenario

- Ausgangslage
 - Sinkende Kundenloyalität
 - Wechselbarrieren in B2C-Märkten (z.B. Telekommunikation) besonders gering
 - Wechselbereitschaft hängt von Personenmerkmalen ab
- Problem
 - Kosten der Neukundenakquise hoch
 - Abwanderung von Kunden kann für Unternehmen ein existenzielles Problem darstellen
 - Maßnahmen zur Kundenbindung (z.B. Preisnachlässe) müssen aus Kostengründen minimiert werden
- Ziel
 - Unzufriedene Kunden frühzeitig identifizieren
 - Unzufriedene Kunden durch Anreize von Abwanderung abhalten

Churn Prediction

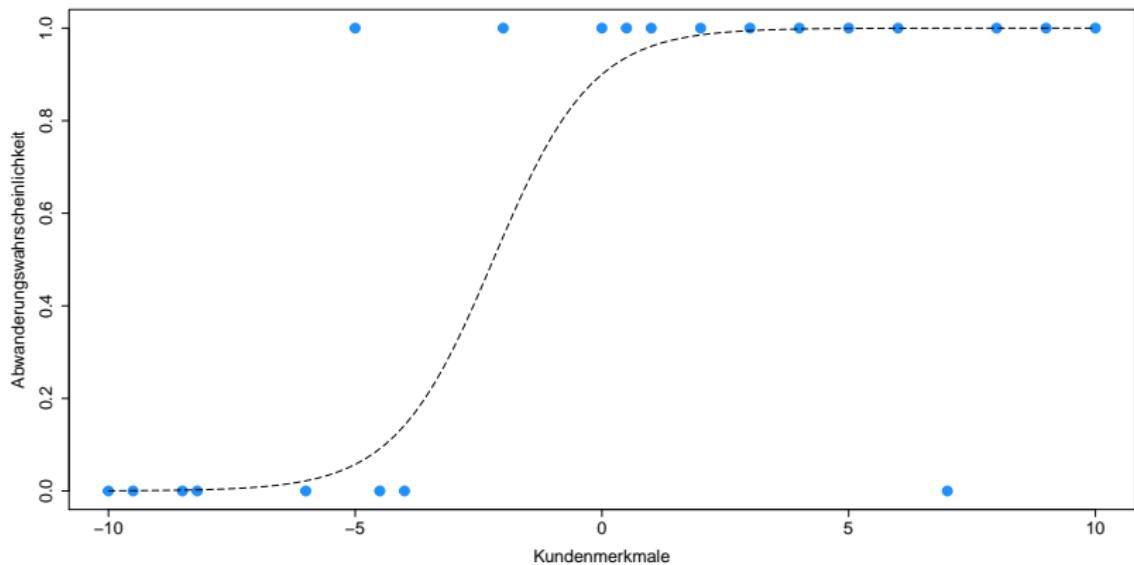
- Lösung: Wahrscheinlichkeit einer Abwanderung („churn“) pro Kunde vorhersagen
- Aus vorhandenen Kundendaten mittels ML-Algorithmen eine Prognose-Funktion ableiten
- Daten (Kundendatenbank Telekommunikationsunternehmen)
 - Hat Kunde während des letzten Monats gekündigt?
 - Dienste, die der Kunde in Anspruch genommen hat (z.B. Telefon, Internet, Streaming)
 - Vertrags-/ Abrechnungsmerkmale (z.B. Vertragsdauer, Zahlungsmethode, Rechnungsbeträge)
 - Demographische Kundenmerkmale (Geschlecht, Alter, Partnerschaft etc.)
- R-Paket caret als Framework

Machine Learning-Anwendung



Quelle: <http://mapr.com/blog/fast-data-processing-pipeline-predicting-flight-delays-using-apache-apis-pt-1> (mit eigenen Änderungen)

Modellierung - Logistische Regression

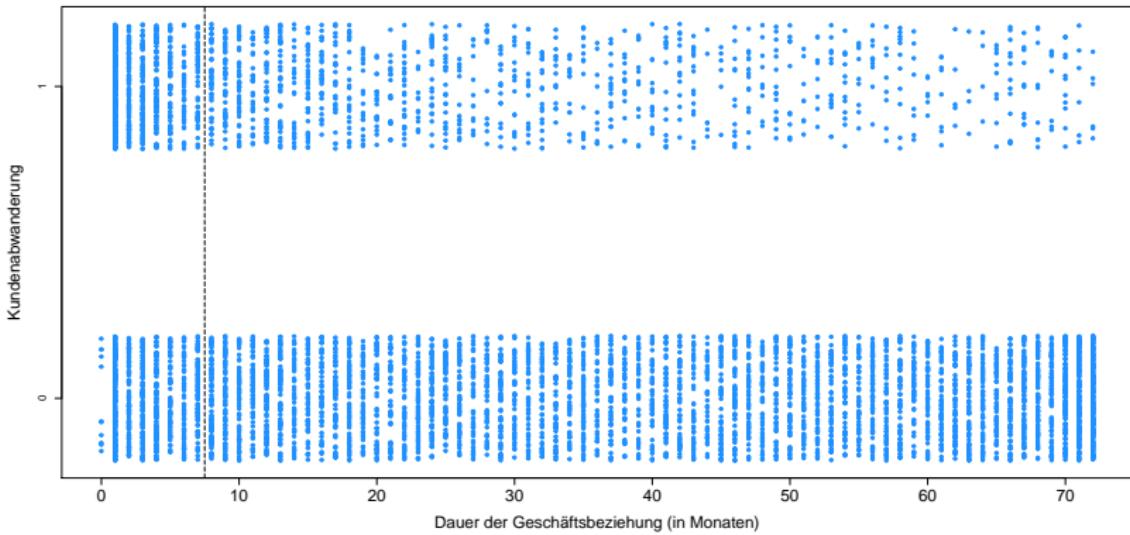


Ergebnisse

- Akkurateitwert der Vorhersage ist mit rund .80 für den ersten Versuch akzeptabel, aber nicht „state of the art“
- Optimierungsmöglichkeiten
 - „feature engineering“
 - Verwendung von Polynomen höherer Ordnung (z.B. x_1, x_1^2, x_1x_2)
 - Verwendung anderer ML-Algorithmen (z.B. SVM, RF, DNN)
- Anwendungsfall verlangt Berücksichtigung der spezifischen Kosten der Zellen der Konfusionsmatrix
- Deployment in SQL Server

- $P(\text{Kundenabwanderung} = 1) = \frac{\exp(.09 - .03 \times \text{Mann} + \dots)}{1 + \exp(\exp(.09 - .03 \times \text{Mann} + \dots))}$
- UPDATE dbo.Customers SET prob = exp ...

Optimierung: Feature Engineering



Kostenberücksichtigung: Konfusionsmatrix I

Tabelle 1: Konfusionsmatrix (Limit = .5)

		Realität	
		Keine Abwanderung	Abwanderung
Vorhersage	Keine Abwanderung	918 (.65)	173 (.12)
	Abwanderung	116 (.08)	200 (.14)

Kostenberücksichtigung: Konfusionsmatrix II

Tabelle 2: Kosten = 577526 EUR (Limit = .5)

		Realität	
		Keine Abwanderung	Abwanderung
Vorhersage	Keine Abwanderung	.65 × 7043 × 0 EUR	.12 × 7043 × 500 EUR
	Abwanderung	.08 × 7043 × 100 EUR	.14 × 7043 × 100 EUR

Kostenberücksichtigung: Konfusionsmatrix III

Tabelle 3: Kosten = 950805 EUR (Limit = 0)

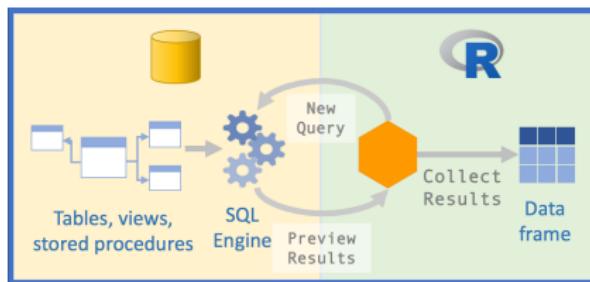
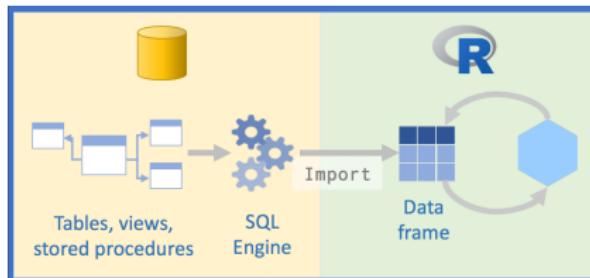
		Realität	
		Keine Abwanderung	Abwanderung
Vorhersage	Keine Abwanderung	.73 × 7043 × 0 EUR	.27 × 7043 × 500 EUR
	Abwanderung		

Weitere Aspekte

Optionen

- Konnektivität von R und SQL Server bzw. Azure SQL
 - SQL Server-Compute Context in R (RevoScaleR)
 - Ausführen von R-Skripts in SQL Server 2016/2017 mittels T-SQL-Prozeduren
- Big Data (RevoScaleR)
 - XDF-Datenformat
 - HDFS-Konnektivität (Hadoop bzw. Spark-Compute Context)

Option: SQL Server-Compute Context



Quelle: <https://rviews.rstudio.com/2017/05/17/databases-using-r> (mit eigenen Änderungen)

Einschränkungen

- Lernkurve flacher als bei anderen Statistikprogrammen (z.B. Stata, SPSS): Lernfortschritt ist zu Beginn meist geringer
 - Fehlen einer vollständigen grafischen Benutzeroberfläche wie in anderen Statistikprogrammen
 - Grundlegende Programmiererfahrung hilfreich
- Syntax weniger stringent als bei „Low Level“-Sprachen
- Pakete sind nicht immer aufeinander abgestimmt
- Performance-Optimierung spielt zunächst nur eine nachgeordnete Rolle, aber Optimierung durch
 - Byte Code-Compiling (`compiler`)
 - Parallel Processing (`doParallel`)
 - Cluster Computing (`sparklyr`, `RevoScaleR`, vgl. oben)
- Steigende Anzahl an Paketen mit Data Science-Bezug beim Konkurrenten Python

Weitere Informationen

- Internet
 - Quick-R
 - RStudio Cheat Sheets
 - R-bloggers
- Kontakt: thomasschuebel(at)web.de

Frage?



Nachfrage: Kann man für die Arbeit mit großen Dateien HDD-Speicher nutzen?

```
big_csv <- "C:/temp/tips.csv"

# Option 1: Base R
memory.limit(size = 100000) # ca. 100 GB als virt. RAM
df <- read.csv(big_csv)

# Option 2: R Client
big_xdf <- "c:/temp/tips.xdf"
rxImport(inData = big_csv, outFile = big_xdf)
rxLinMod(tip ~ total_bill, data = big_xdf)
```

Nachfrage: Was tun, wenn mehrere Funktionen den gleichen Namen tragen?

```
require(plyr)
require(Hmisc)

plyr::summarize(mtcars, avgwt = mean(wt))

##      avgwt
## 1  3.21725

Hmisc::summarize(mtcars$wt, mtcars$am, median)

##      mtcars$am mtcars$wt
## 1          0     3.52
## 2          1     2.32
```

Nachfrage: Wie heißt die Evaluierungsfunktion?

```
a <- 2  
b <- 5  
eval(a*b)
```

```
## [1] 10
```

```
eval(parse(text="17+3"))
```

```
## [1] 20
```

```
eval(parse(text="17+3+a*b"))
```

```
## [1] 30
```