W-Rechnung und Statistik für Ingenieure Übung

Christoph Kustosz (kustosz@statistik.tu-dortmund.de)

Mathematikgebäude Raum 715

technische universität dortmund

Organisatorisches

- Übungstermin: Mittwochs 17.00 - 18.00
- Sprechzeiten: Do. 10.00 - 11.00, M 715
- Unterlagen zur Übung und Vorlesung sind online verfügbar (https://www.statistik.tu-dortmund.de/ingu1213.html)
- Hauptklausur: 26.03.13, 14.15 - 16.30
  Raumaufteilung online
- Anmeldung zur Klausur über das BOSS-System
  Bei Studienfächern, bei denen dies nicht möglich ist, erfolgt die Anmeldung über eine Liste
  (am Raum 746, Mathe-Tower)
- Anmeldeschluss ist von jeweiliger Studienordnung abhängig

Was haben wir vor?

- Die Vorlesung behandelt viele Themen der theoretischen und angewandten Statistik
- Themen sind z.B.
  - Wie kann man wichtige Eigenschaften von Daten zu Kennzahlen zusammenfassen?
  - Wie kann man Daten graphisch darstellen?
  - Was sind Wahrscheinlichkeiten, wie rechnet man damit und welche Gesetze gelten?
  - Wie kann man Zufall beschreiben?
  - Kann man trotz zufälliger Schwankungen Aussagen über Kennzahlen belegen?

In der Praxis gibt es analoge Fragestellungen

- Wie hoch ist die mittlere Dicke von produzierten Stahlblechen einer Maschine? Wie stark weichen die einzelnen Bleche von der Norm ab?
- Sind Risse in verschiedenen Stahlbetonplatten unter gleichen Belastungen nach gleicher Zeit in etwa gleich lang?
- Wie wahrscheinlich ist der Defekt eines Produkts, wenn verschiedene Lieferanten Bauteile unterschiedlicher Qualität liefern? Wie wahrscheinlich ist es, dass ein bestimmter Lieferant die Bauteile geliefert hat, wenn das Produkt defekt ist?
- Wieso kann man Flugzeuge überbuchen? Wie plant man ein Lager?
- Kann man belegen, dass ein Holländer holländischen Gouda von deutschem unterscheiden kann, wenn er 11 von 20 mal richtig liegt?

Ziel der Übung

Vertiefung der Vorlesungsinhalte durch die praktische Anwendung auf Daten!

Das Statistik-Programm R

- R ist eine objekt-ortientierte, interaktive Programmiersprache
- R ermöglicht unter Anderem statistische Auswertungen, Erstellung von Graphiken und Simulationen
- In R können Verfahren und Funktionen in Form von Paketen modular verwendet werden, dadurch liefert R das Handwerkszeug für vielfältige statistische Untersuchungen
- Daten und R Ergebnisse können auf verschiedene Arten ein- und ausgegeben werden was die weitere Verwendung mit vielen Programmen ermöglicht
- R kann kostenlos über http://cran.r-project.org heruntergeladen werden

Installation von R

- R ist für Windows, MacOS X und verschiedene Linux Distributionen verfügbar
- Auf der R Homepage können Installationspakete heruntergeladen werden
- Im Folgenden wird die Windows Installation beschrieben

Installation von R - Windows (1)

- R-Homepage aufrufen
- Betriebssystem auswählen

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 MacOS X
- Download R for Windows

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 (2011-09-30): R-2.13.2.tar.gz (read what's new in the latest version).

Installation von R - Windows (2)

- base Paket auswählen

R for Windows

Subdirectories:

base	Binaries for base distribution (managed by Duncan Murdoch). This is what you want if you **install R for the first time**.
contrib	Binaries of contributed packages (managed by Uwe Ligges)

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / sugge
to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Installation von R - Windows (3)

- Dateien herunterladen
- Nach dem Download R-2.15.1-win.exe ausführen, um Installation zu starten
- Bildschirmanweisungen folgen

R-2.15.1 for Windows (32/64 bit)

Download R 2.15.1 for Windows (77 megabytes, 32/64 bit)

Installation and other instructions

New features in this version

If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can
true fingerprint. You will need a version of md5sum for windows: both graphical and command line versions are available.

Frequently asked questions

- How do I install R when using Windows Vista?
- How do I update packages in my previous version of R?
- Should I run 32-bit or 64-bit R?

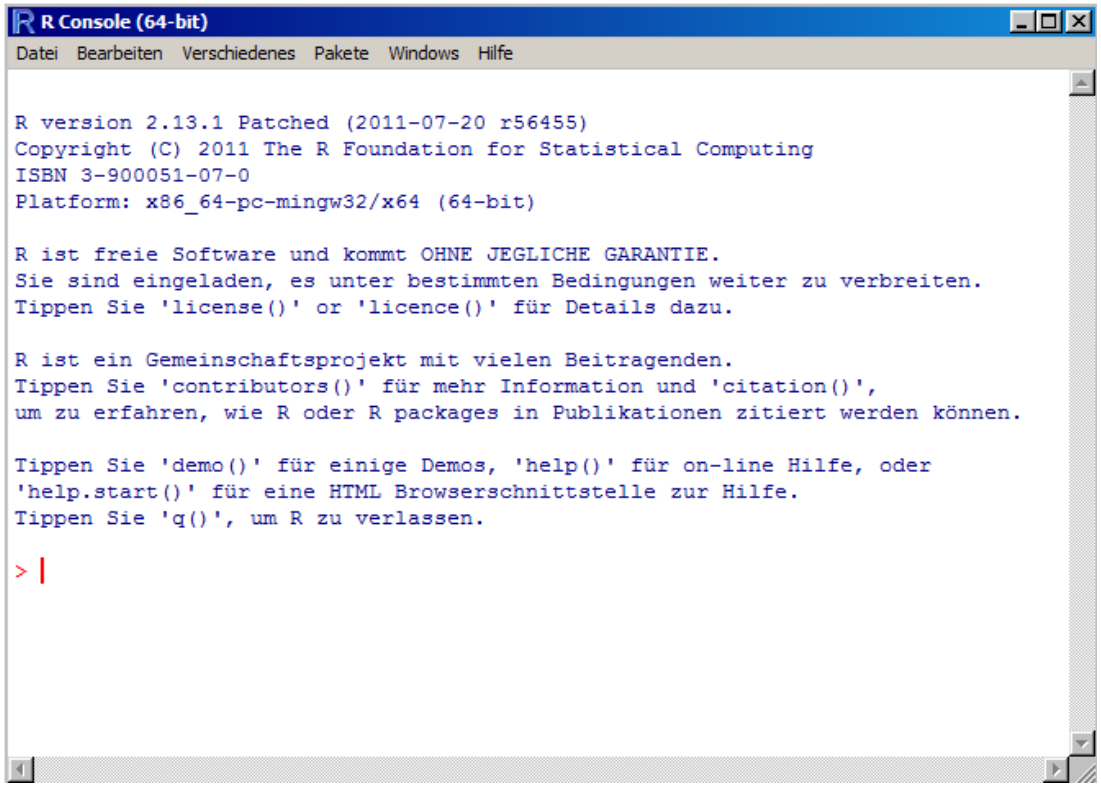
Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information.

Starten von R

- R wird unter Start → Programme → R → R 2.15.1. gestartet
- Alternativ über einen angelegten Shortcut auf dem Desktop
- Bei Windows Vista kann es für volle Funktionalität notwendig sein R im 'Administrator'-Modus auszuführen
  Dazu: Rechtsklick auf Datei → Als Administrator ausführen

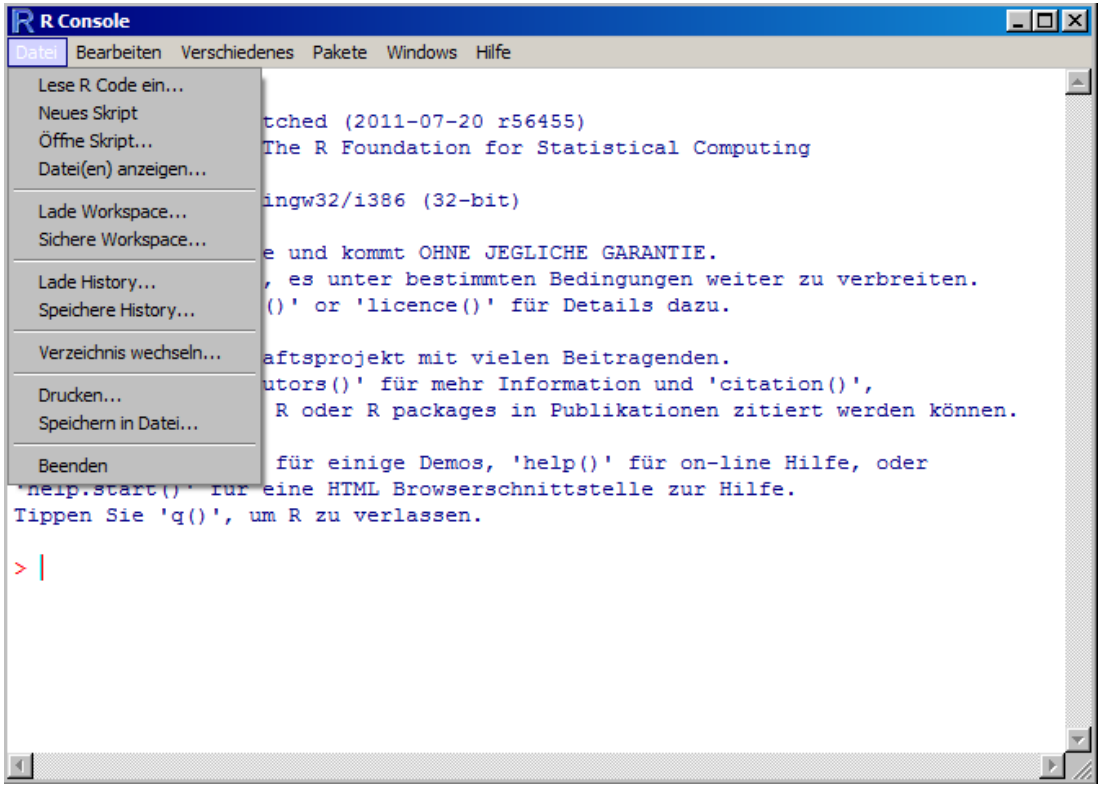
Die Konsole

- R wird in einer Bedienkonsole ausgeführt, in welche Befehle eingegeben werden
- R führt diese nach einer Bestätigung mit der Enter-Taste direkt aus



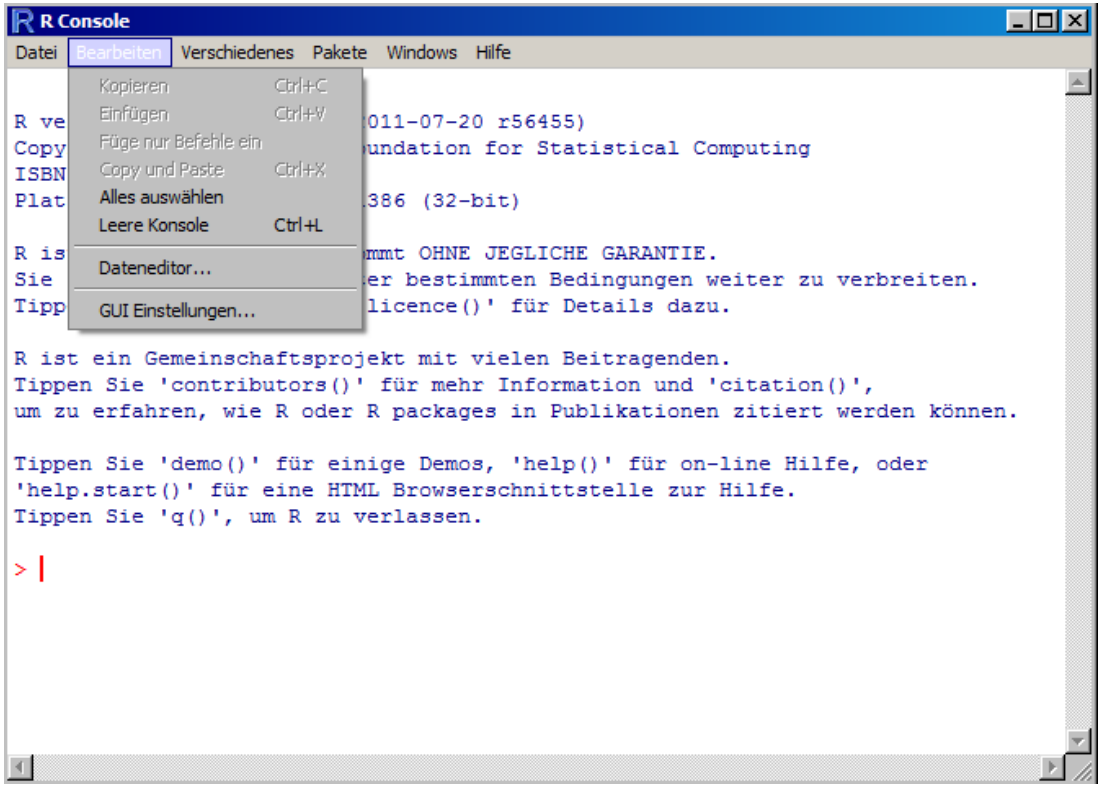
Die Konsole - Datei

- Unter **Datei** können Skripte geladen und gespeichert, eine History eingesehen, das Arbeitsverzeichnis festgelegt und Druckaufträge erteilt werden



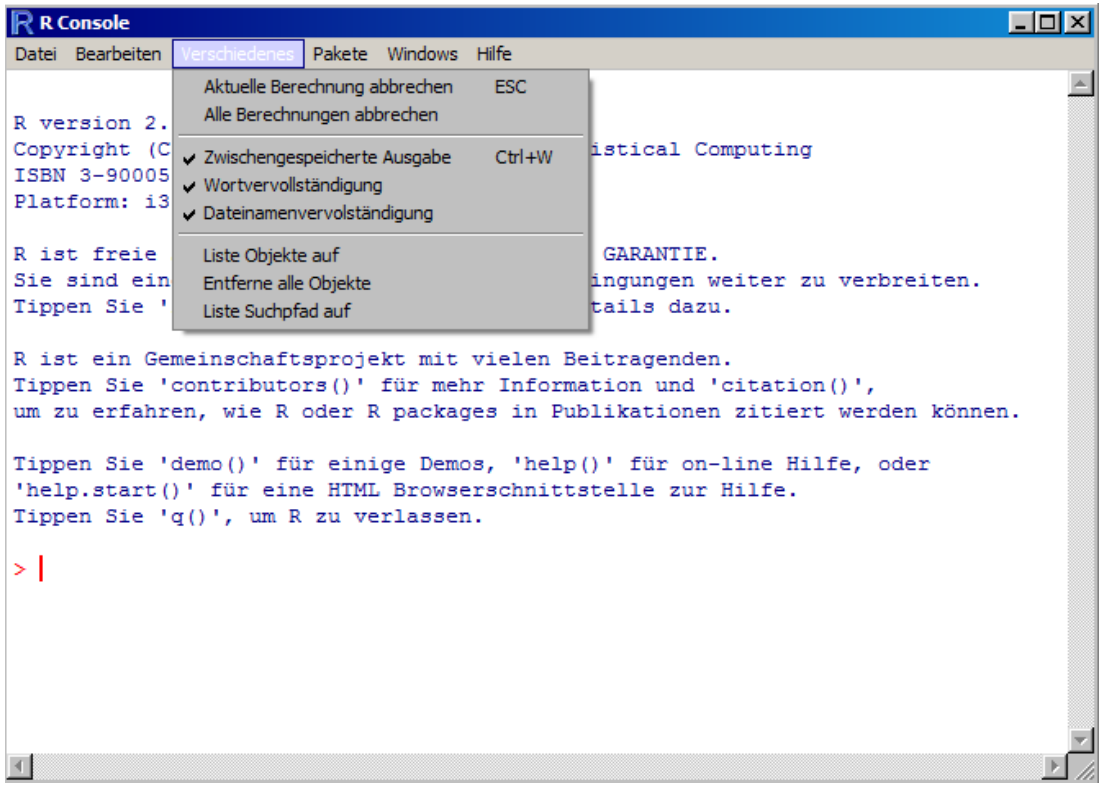
Die Konsole - Bearbeiten

- **Bearbeiten** enthält Funktionen zum Kopieren und Einfügen, einen Dateneditor und Einstellungsmöglichkeiten des User-Interface



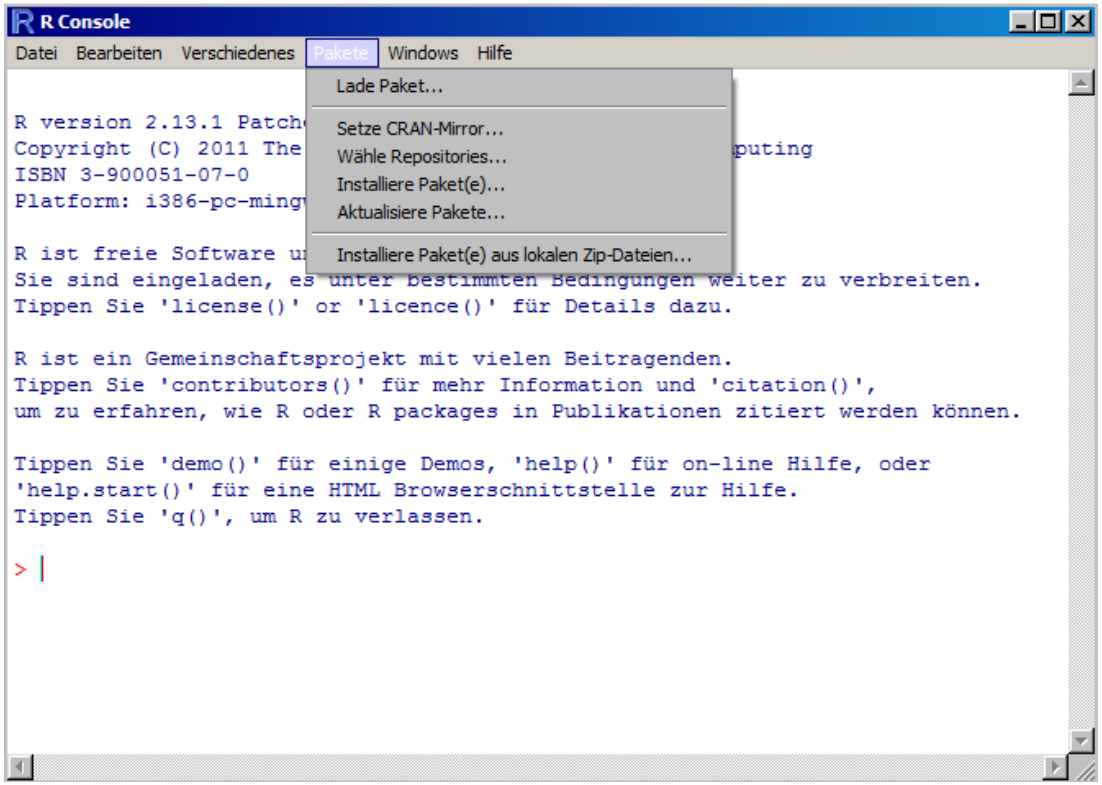
Die Konsole - Bearbeiten

- **Verschiedenes** enthält einige Optionen der Konsole und nützliche Kurzbefehle



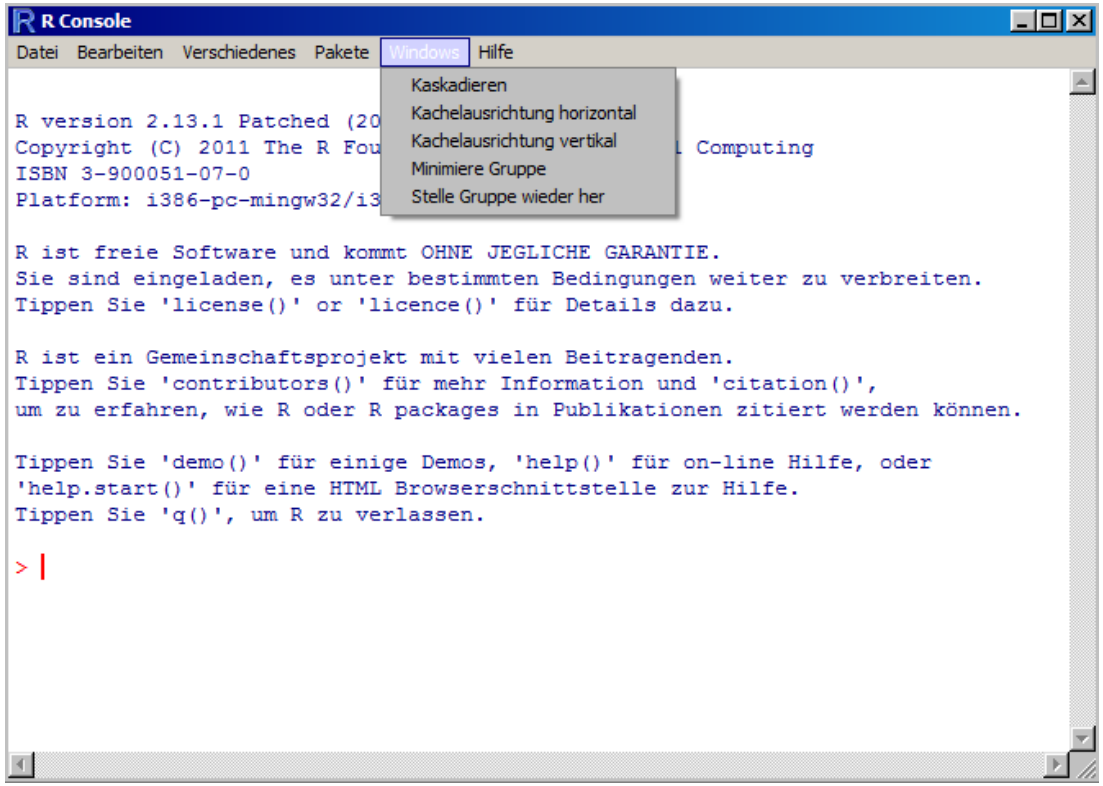
Die Konsole - Pakete

- Unter **Pakete** können R-Pakete mit Daten und Funktionen heruntergeladen, installiert und aktiviert werden



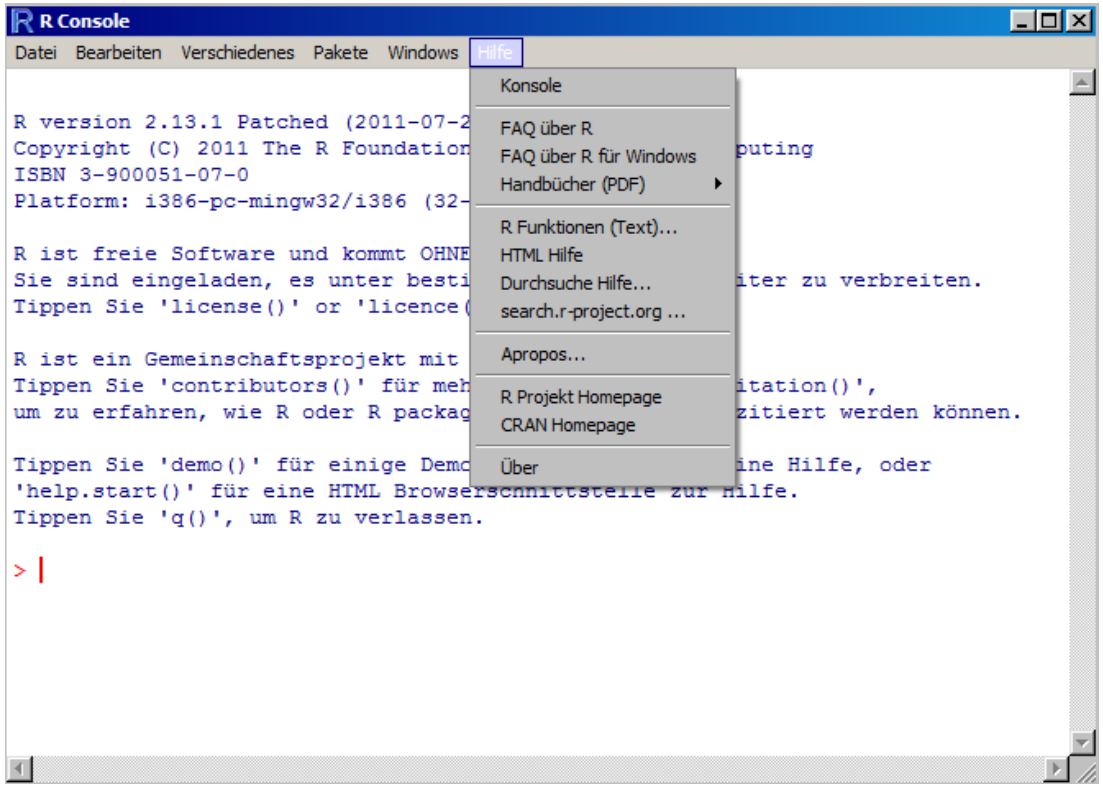
Die Konsole - Windows

- **Windows** beinhaltet Befehle zur Anzeige der R Konole und weiterer R-Fenster



Die Konsole - Windows

- Unter **Hilfe** befinden sich Einführungs- und Hilfetexte zu R



Erste Schritte : Grundrechenarten

- Typische Grundrechenarten
  `+,-,*,/,^`

```
> 3+3
[1] 6
>4*5
[1] 20
>2^3
[1] 8
>(2*3+6)^2
[1] 144
```

Erste Schritte : Mathematische Funktionen

- Beispiele: exp, sqrt, round, sin, cos,...

```
> exp(3)
[1] 20.08554
> sqrt(9)
[1] 3
> round(exp(3),2)
[1] 20.09
> sin(pi)
[1] 1.224647e-16
> cos(pi)
[1] -1
```

Viele weitere sind in R vorimplementiert!

Erste Schritte : Vektoren und Matrizen

- Man kann in R unter anderem Zahlen, Vektoren, Matrizen, Listen und Tabellen erstellen
- Diese können in Variablen im Arbeitsbereich gespeichert werden
  - Anzeigen aller vorhandenen Variablen: ls()
  - Löschen einer Variable: rm(VarName)

```
> num<-5
> vec<-c(1,2,3)
> mat<-matrix(data=c(num,vec),ncol=2,byrow=T)
> mat
[,1] [,2]
[1,] 5 1
[2,] 2 3
> ls()
[1] "num" "mat" "vec"
```

Erste Schritte : Rechnen mit Variablen

- Mit Variablen kann wie mit Zahlen gerechnet werden
- Dabei ist auf den jeweiligen Datentyp zu achten
  Beispiel:
  - %*% ist Matrix-Multiplikation
  - * ist komponentenweise Multiplikation

```
> num*vec
[1] 5 10 15
> vec2<-c(1,2)
> mat%*%vec2
[,1]
[1,] 7
[2,] 8
```

Erste Schritte : Multiplikation (WARNUNG)

- Es ist wichtig, die Datentypen und Rechenvorschriften zu beachten

```
> mat%*%mat
[,1] [,2]
[1,] 27 8
[2,] 16 11
> mat*mat
[,1] [,2]
[1,] 25 1
[2,] 4 9
```

Erste Schritte : Mehr Funktionen

- In R gibt es viele weitere Funktionen
  Beispiele:
  - Anzahl der Elemente: `length(Vektor)`
  - Minimum: `min(Vektor)` oder `min(Matrix)`
  - Dimension: `dim(Matrix)`
  - Transponieren: `t(Vektor)` oder `t(Matrix)`

```
> length(vec)
[1] 3
> min(mat)
[1] 1
> dim(mat)
[1] 2 2
> t(mat)
[,1] [,2]
[1,] 5 2
[2,] 1 3
```

Erste Schritte : Erzeugen von Daten

- Daten und Vektoren können mit hilfreichen Funktionen erstellt werden
  Beispiele:
  - Einlesen einer dat-Datei: `scan(Quelle)`
  - Wiederholen von Werten: `rep(Werte,Wiederholungen)`
  - Folge: `Anfangswert:Endwert`
  - Folge mit gegebenen Abständen:
    `seq(Anfangswert,Endwert,Inkrement)`
  - Initialisierung von leerem Vektor: `numeric(Länge)`

```
> dat<-scan("Daten.dat")
> rep(c(1,2),3)
[1] 1 2 1 2 1 2
> 1:4
[1] 1 2 3 4
> seq(-1,1,0.4)
[1] -1.0 -0.6 -0.2 0.2 0.6 1.0
```

Erste Schritte : Komponentenzugriff

- Auf einzelne Komponenten von Variablen kann zugegriffen werden

```
> vec[2]
[1] 2
> mat[2,2]
[1] 3
> mat[,2]
[1] 1 3
>vec[2:3]
[1] 2 3
> test<-sin(seq(-pi,pi,0.5))
> test[test>0]
[1] 0.3507832 0.7568025 0.9775301 0.9589243
[5] 0.7055403 0.2794155
> which(test>0)
[1] 8 9 10 11 12 13
```

Erste Schritte : Kombinieren von Daten

- Man kann Spalten oder Zeilen zu Matrizen kombinieren
  - Spalten: `cbind(Vektor1,Vektor2,...)`
  - Zeiten: `rbind(Vektor1,Vektor2,...)`

```
> v1<-c(1,1,1)
> v2<-c(2,1,2)
> cbind(v1,v2)
v1 v2
[1,] 1 2
[2,] 1 1
[3,] 1 2
> rbind(v1,v2)
[,1] [,2] [,3]
v1 1 1 1
v2 2 1 2
```

Erste Schritte : Schleifen

- In R sind verschiedene Arten von Schleifen und Bedingungen möglich
  Beispiele:
  - for-Schleifen
  - while-Schleifen
  - if-else-Bedingungen

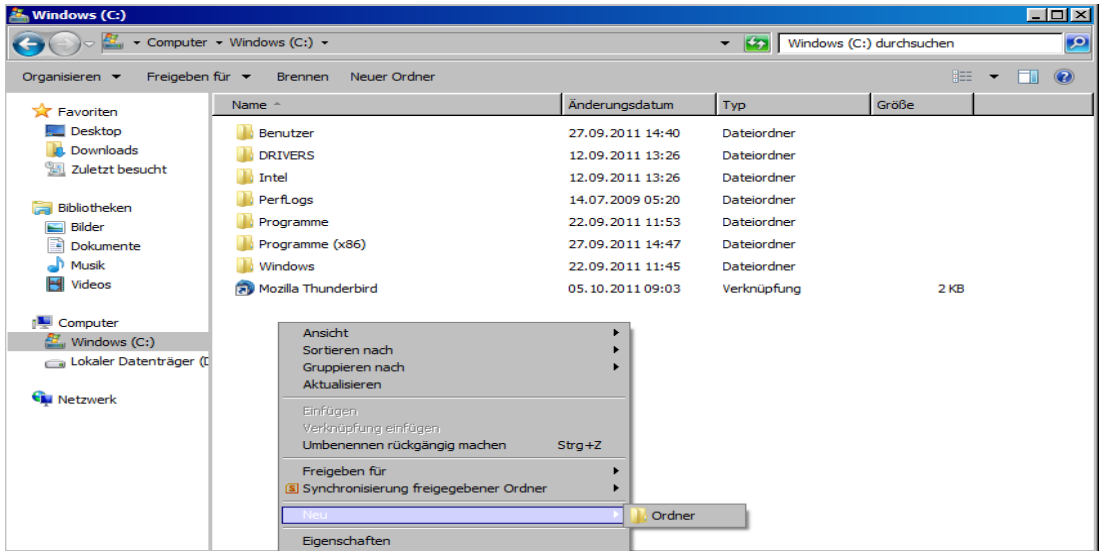
```
> v<-seq(-1,1,0.5)
> v
[1] -1.0 -0.5 0.0 0.5 1.0
> s<-numeric(length(v))
> for(i in 1:length(v)){s[i]<-v[i]^2}
> s
[1] 1.00 0.25 0.00 0.25 1.00
>s2<-v^2
> s2
[1] 1.00 0.25 0.00 0.25 1.00
```

Erste Schritte : Arbeitsverzeichnis

- Später werden wir vorbereitete Daten laden und bereits erstellte R-Codes ausführen, welche in zusätzlichen Dateien hinterlegt sind
- R ruft diese Dateien in einem **Arbeitsverzeichnis** auf
- Dieses kann festgelegt werden mit Datei → Verzeichnis wechseln...
  Alternativ mit dem Befehl: > `setwd(Pfad)`
- Damit können Dateien in diesem Verzeichnis ohne weitere Pfadangabe verwendet werden

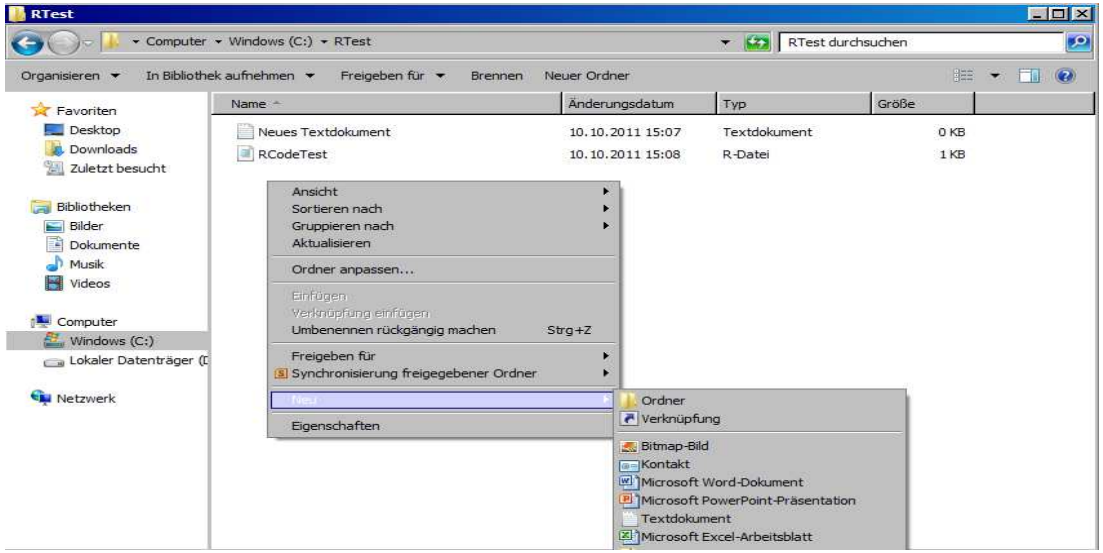
Erste Schritte : Arbeitsverzeichnis - Beispiel (1)

- Erstelle auf der Festplatte einen Ordner: RTest



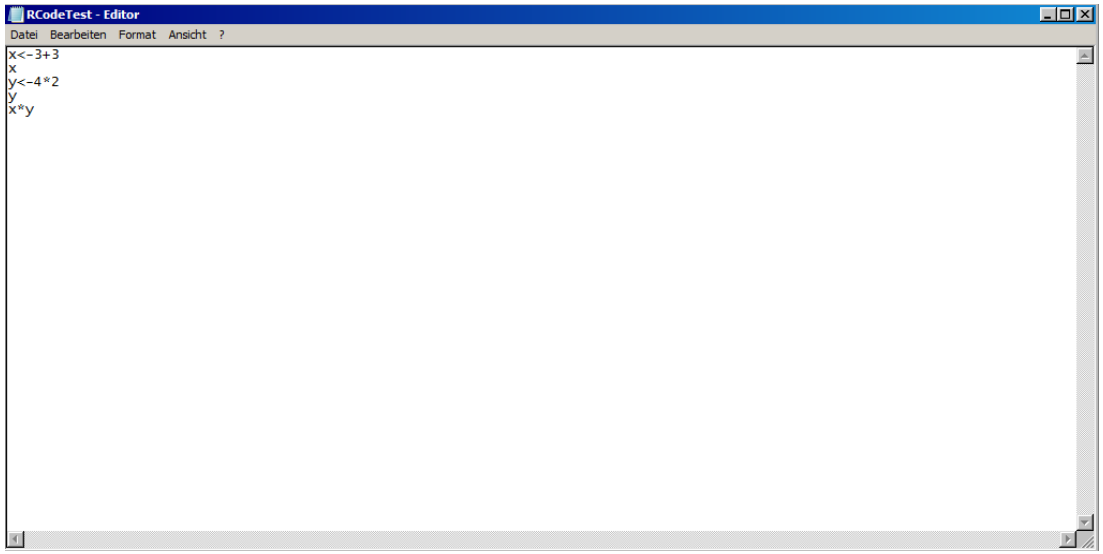
Erste Schritte : Arbeitsverzeichnis - Beispiel (2)

- Erstelle in diesem Ordner ein Textdokument



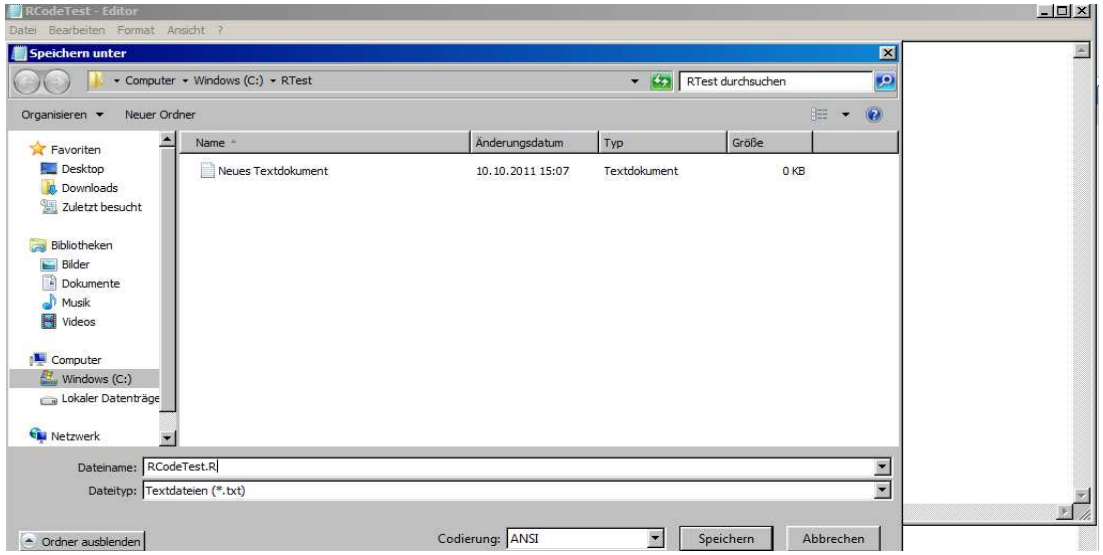
Erste Schritte : Arbeitsverzeichnis - Beispiel (3)

- Schreibe in das Dokument R-Anweisungen



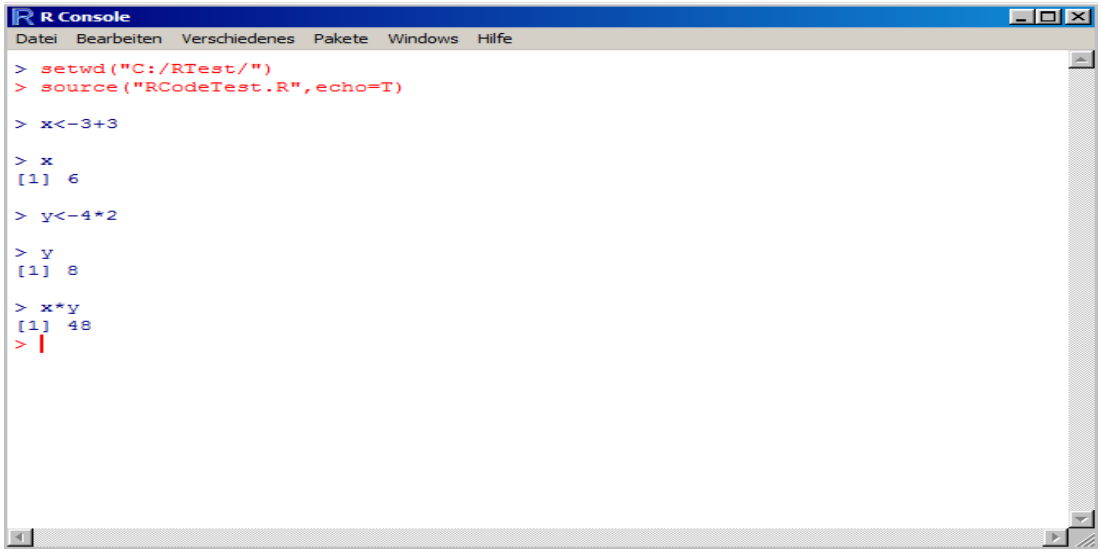
Erste Schritte : Arbeitsverzeichnis - Beispiel (4)

- Speichere das Textdokument als RTestCode.R



Erste Schritte : Arbeitsverzeichnis - Beispiel (5)

- In R definiere das Arbeitsverzeichnis durch `setwd('C:/RTest/')`
- Führe den Code aus mit `source('RCodeTest.R',echo=T)`



Erste Schritte :Hilfequellen

- Der Menüeintrag 'Hilfe' liefert den Zugang zu vielen Texten über R
- Benötigt man Hilfe zu einer Funktion, so kann man diese direkt durch die Eingabe von ?Funktionsname erreichen
- Einige weitere Hilfequellen:
  - Skript zur Einführung in R von Prof. Dr. Müller auf der Homepage der Vorlesung
  - Buch: Programmieren mit R (Statistik und ihre Anwendungen), Uwe Ligges, Springer

Ende

Vielen Dank für Ihre Aufmerksamkeit!
Viel Spaß und Erfolg beim Lösen der Aufgaben!