Einführung in R (I)

- Was ist R?
  - frei verfügbares Statistikprogramm für die gesamte Bandbreite statistischer Verfahren
  - extreme Flexibilität und objektorientierte Programmiersprache (fast identisch zur Sprache S)
  - spezielle Stärken bei Grafiken und der nahezu unendlichen Erweiterbarkeit
- Nützliche Literatur:
  - Monographien:
    – Dalgaard, P. (2002), Introductory Statistics with R, New York: Springer.
    – Venables, W.N., Ripley, B.D. (2002), Modern Applied Statistics with S, 4. Aufl., New York: Springer. [Standardwerk für statistische Anwendungen]
  - Tutorien in PDF-Format:
    – Farnsworth, G.V. (2006), Econometrics in R. [speziell auf Ökonometrie ausgerichtet]
    – Paradis, E. (2005), R for Beginners. [konziser und einigermaßen kompletter Kurs]
    – Venables, W.N., Smith, D.M. (2006), An Introduction to R. [wird mit R installiert]
    – Verzani, J. (2001), simpleR – Using R for Introductory Statistics. [wunderschön aufbereitet]
  - Web-Seiten:
    – http://www.r-project.org [die R-Seite page mit Diskussionsgruppen und Zugang zu packages]
    – http://www.uni-koeln.de/themen/statistik/software/s [viel deutschsprachiges Material]

Einführung in R (II)

- Vektoren und Matrizen:
  - Erzeugung von Vektoren: `c()`, `rep()`, `seq()`
  - Erzeugung von Matrizen: `matrix()`, `cbind()`, `rbind()`
  - Zuordnung zu Objekten: `A <- matrix(c(1:12),4,3)`
  - Extrahierung: dritte Zeile `A[3,]`, zweite Spalte `A[,2]`, Element in Zeile 2, Spalte 3 `A[2,3]`, 2×2 Teilmatrix oben-links `A[1:2,1:2]`
  - Abfragen: `is.vector()`, `is.matrix()`, `length()`, `dim()`, `nrow()` , `ncol()`
  - Benennung: `dimnames()`, `rownames()`, `colnames()`
    [Zeilen und Spalten können auch über ihre Namen angesprochen werden]
  - Einträge könne logisch (`TRUE` oder `FALSE`), ganzzahlig, real oder komplex sein bzw. Strings (Abfragen `is.logical`, `is.integer`, `is.numeric`, `is.complex`, `is.character`)
- Arrays:
  - multidimensionale Erweiterungen von Matrizen: `array()`
  - `dim()` und `dimnames()` sind hier ebenfalls anwendbar
  - sehr effiziente Möglichkeiten bestimmte Teile zu extrahieren

Einführung in R (III)

- Data Frames:
  - Matrizes mit verschiedenen Datentypen in den Spalten: `data.frame()`
  - Umwandlung in eine Matrix (manchmal erforderlich): `as.matrix()`
- Faktoren:
  - vektor-ähnliche Zusammenfassungen kategorialer Variablen, die entweder
    - ungeordnet `factor()` oder
    - geordnet `ordered()` sein können, wobei `levels` die Reihung spezifiziert
- Listen:
  - nützliche Sammelobjekte verschiedenster Objekttypen: `list()`, `unlist()` [Vorsicht!]
  - z.B. Kombination einer Matrix `A` und eines Vektors `b` in einer Liste mit `L <- list(A,b)` und Extrahierung der Teile mit `L[[1]]` und `L[[2]]` (alternativ: `L <- list(A=A,b=b)`, Extrahierung mit `L$A` und `L$b`)
- Strings:
  - alphanumerische Zeichenfolgen („Worte und Sätze")
  - viele nützliche Befehle verfügbar, z.B. `paste()`, `substring()`, `nchar()`

Einführung in R (IV)

- Basisoperationen:
  - Aufruf der Hilfefunktion mit `?command`
  - Zuweisung zu Variablen mit `<-` oder `=` (`<-` ist üblicher in in R)
  - arithmetische Operationen `+`, `-`, `*`, `/`, `^`, `sum()`, `exp()`, `log()`, `sqrt()`, `min()`, `max()`
  - Mengenbefehle: `union()`, `intersect()`, `setdiff()`, `setequal()`, `is.element()`
  - die meisten Befehle sind matrixorientiert, d.h. beziehen sich auf alle Matrixelemente; `apply()` ist hier nützlich, z.B. `apply(A,2,sum)` berechnet die Spaltensummen von `A`
- Programmkontrolle:
  - for-Schleife: `for(i in 1:n)` { ... commands ... } [zu vermeiden, setzt Geschwindigkeit herab]
  - repeat-Schleife: `repeat` { ... commands ... }, wobei die Schleife mit `break` verlassen wird
  - while-Schleife: `while(condition)` { ... commands ... }
  - if-then-else: `if(condition)` { ... then part ... } `else` { ... else part ... }, `ifelse()`
- Matrix- und statistische Operationen:
  - umfangreiche und flexible Befehle für Matrixoperationen verfügbar (siehe Matrixalgebra später)
  - ebenfalls viele statistische Kommandos verfügbar (siehe Teil statische Grundlagen später)

Einführung in R (V)

- Input:
  - über die Tastatur: `data.entry()`, `edit()`
  - aus der Zwischenablage: `read.table("clipboard")`
  - aus Dateien:
    - `read.table()`, z.B. einlesen einer csv-Datei mit Benennung in den Dataframe `D`:
      ```
      D <- read.table("C:\\d.csv",header=T,row.names=1,sep=";")
      ```
    - `scan()` [für einzelne Datenreihen, Geschmackssache]
    - `load()` [lädt alle Objekte, die vorher mit `save()` in eine Datei gespeichert wurden]
  - fehlende Werte kodiert mit `NA` (nicht `"NA"`!), Abfrage mit `is.na()`
- Output:
  - auf den Bildschirm: Objektname eingeben, `print()`, `cat()`, `fix()`
  - in die Zwischenablage: `write.table(D,file="clipboard",sep="\t",eol="\n")`
    [sehr nützlich wenn man mit Excel arbeitet]
  - in Dateien:
    - `write.table(D,file="C:\\d.csv",sep=";",eol="\n")` [für größere Dateien]
    - `save(A,b,file="C:\\d.csv")` [speichert alle Objekte in eine Datei]
  - weitere Information im Manual „R Data Import/Export” (wird automatisch mit R installiert)

Einführung in R (VI)

- Funktionen:
  - eigene Funktionen leicht programmierbar über den Befehl `function()`
  - Beispiel: eine Funktion, die die Spur einer Matrix berechnet (Summe der Hauptdiagonalelemente)
    ```
    trace <- function(X) { return(sum(diag(X))) }
    ```
    oder besser mit Prüfung, ob die Matrix quadratisch ist:
    ```
    trace <- function(X)
             {
                if(nrow(X)==ncol(X)) return(sum(diag(X)))
                else stop("Matrix ist nicht quadratisch!”)
             }
    ```
  - Beispiel: eine Funktion, die Zeilen- und Spaltensummen einer Matrix als Liste wiedergibt
    ```
    rcs <- function(X)
           { return(list(rs=apply(X,1,sum),cs=apply(X,2,sum))) }
    ```
  - für die Integration von Funktionen, die in C oder FORTRAN programmiert sind siehe das Manual „Writing R Extensions” (wird automatisch mit R installiert)
- Packages:
  - hinzuladen eines vorher installierten R-Packages mit `library()`
  - einlesen und ausführen eines Programms aus einer Textdatei mit `source()`

Einführung in R (VII)

- Graphiken:
  - öffnen eines Graphikfensters mit `windows(height,width)` [nicht unbedingt erforderlich]
  - grundlegenden Graphikbefehle: `plot()`, `points()`, `lines()`, `curve()`, `barplot()`
  - plotten von mehrerer Datenreihen mit `matplot()`
  - dreidimensionale Graphiken können mit `persp()` erstellt werden
  - Axen, Striche und Beschriftung werden automatisch angefügt, aber modifizierbar mit `axis()`
  - eine Legende kann mit `legend()` hinzugefügt und vielfach gestaltet werden
  - ein Titel und weitere Beschriftung kann mit `title()` und `text()` hinzugefügt werden
  - sogar mathematische Ausdrücke können mit `expression()` oder `substitute()` hinzugefügt werden [`demo(plotmath)` bietet eine Demonstration der Möglichkeiten]
  - verschiedene Parameter können gesetzt werden: z.B. Schriftart mit `font` , Textgröße mit `cex`, Linienart und -breite mit `lty` und `lwd`, Farbe mit `col`, ...
  - der Befehl `par()` bietet vielfältige Möglichkeiten zur Modifikation der Details von Graphiken [z.B. Anordnung von 4 Graphiken zeilenweise als 2×2-Matrix mit `par(mfrow=c(2,2))`]
  - Export von Graphiken leicht über die Zwischenablage als Metafile oder Bitmap bzw. über die Speicherung als Metafile, Pdf, Postscript, Jpeg, usw.