

Einführungsstunde

1) What is R

- R ist ein Statistisches Programmpaket, das sowohl auf Unix, Windows als auch Macintosh läuft.
- R ist das open-source Pendant zum kommerziellen S-plus.
- Kostenloses Download von der R-Homepage www.r-project.org ; aktuelle Version 2.2.1
- Neben dem R-Manual "An Introduction to R", den FAQ und den R-Paketen `UsingR` und `TeachingDemos` gibt es auf der Homepage zahlreiche weitere Einführungsunterlagen.
- R ist Software und Sprache, d.h. es stellt umfangreiche statistische Methoden zur Verfügung und erlaubt zusätzliche eigene Funktionen zu schreiben.
- R ist interpretierte Sprache, d.h. Kompilierung wie z.B. bei C/C++ oder Fortran ist nicht nötig.
- R ist nicht über eine Menüleiste zu bedienen, sondern fast ausschließlich über die Kommandozeile. Die Syntax ist jedoch einfach und intuitive.
- R ist case-sensitive, d.h. unterscheidet Klein- und Großbuchstaben.
- R besitzt ein eingebautes Hilfesystem und eine gute Online Hilfe.
- R ermöglicht flexible Grafikerstellung (Formeln und mathematische Symbole; diverse Speicherformate (jpg, ps, pdf, ...)).
- Editor der Wahl für Windows ist Tinn-R (download <http://www.sciviews.org/Tinn-R> und <http://www.sciviews.org/SciViews-R>); für Linux nutze Emacs/ESS (<http://ess.r-project.org/>)

2) Rechnen mit R

- Prompt ">" indiziert, dass R auf Eingabe wartet; "+" bedeutet, dass der aktuelle Befehl nicht vollständig ist (Klammer fehlt) oder noch fortgesetzt wird (z.B. mehrzeilige Funktionen).
- R als Taschenrechner

3) Objekte und Daten

- Mit dem Zuweisungsoperator "<-" können Variablen, Daten, Funktionen, Ergebnisse, etc als **Objekte** gespeichert werden. Der Name eines Objekts muss mit den Buchstaben A bis Z bzw. a bis z anfangen und kann neben weiteren Buchstaben auch Zahlen (0 – 9), Punkte (.), und Unterstriche (_) beinhalten. Auf diese Objekte können Operatoren (arithmetische, logische, vergleichende, ...) und Funktionen angewandt werden.
- Abruf des Objektinhalts mit dem Objektnamen oder über die Funktion `print`.
- Objekte besitzen zwei intrinsische **Attribute**: `mode` fragt den Typ der Objektelemente ab

(numeric, character, complex, logical) und `length` die Anzahl der Objektelemente.

- Mode 'numeric' kann entweder ein integer (32 bits) oder double (64 bits bzw. 8 bytes mit 15 Dezimalstellen Genauigkeit) Datentyp zugrundeliegen. R kennt kein single precision Format (32 bits bzw. 4 bytes und 6 Nachkommastellen Genauigkeit). Siehe hierzu auch `?Machine`
- Für 'character' sind sowohl einfache als auch doppelte Anführungszeichen erlaubt.
- missing data: NA (not available)
- R speichert die Objekte im Arbeitsspeicher (**Working Space**):
`ls`, `ls.str` listet die sich im Speicher befindenden Objekte mit ihren Namen auf.
`rm` entfernt Objekte; `save`, `save.image` speichert den Working Space oder Teile davon, die zu einem anderen Zeitpunkt mit `load` wieder geladen werden können.
- Working Directory: `getwd`, `setwd`
- Wesentliche Arten von Datenobjekten: Vektoren, Faktoren, Matrizen, Arrays, Dataframes und Listen. Die letzten zwei können Daten unterschiedlichen Typs beinhalten, wobei die Variablen eines Dataframes **gleich** lang sein müssen, während der Aufbau von Listen völlig frei ist.
- Beispiele zum **Daten generieren und manipulieren**. Auch Elementezugriff (über Index oder Name) und erste Funktionen wie z.B. `sum` und `exp` oder `rbind` und `merge` werden vorgestellt.
- **Daten lesen und schreiben** mit `read.table/scan` und `write.table/save` (siehe auch R-manual "R Data Import/Export").

4) Operatoren, Funktionen und Pakete

- Operatoren: - vergleichende `<`, `>`, `>=`, `<=`, `==`, `!=`
- logische `&`, `&&`, `|`, `||`
- `anyfun()` zur Ausführung der Funktion `names anyfun`; `anyfun` zum Anzeigen ihres Inhalts
- Verfügbare Funktionen befinden sich im Unterverzeichnis "library" des RHOME auf der Festplatte (RHOME ist das Verzeichnis, in dem R installiert ist). "library" beinhaltet **Pakete** von Funktionen, die selbst wieder in Verzeichnisse gegliedert sind. Das Paket `base` etwa beinhaltet die grundlegenden Funktionen der Sprache, insbesondere zum Daten lesen und manipulieren.
- `install.packages` zur Installation; `library` und `require` fürs Laden in die Liste der momentan geladenen Pakete.
- `detach`, um ein Objekt oder auch ein Paket vom Suchpfad [`search()`] aller in R verfügbaren Objekte zu entfernen.
- `attach`, um z.B. einen Dataframe dem Suchpfad von R hinzuzufügen. Dann können die Komponenten des Dataframe direkt über ihre Variablennamen angesprochen werden. Also z.B.

'group' statt 'daten\$group'.

5) The Help System

- `?name` bzw. `help(name)` für bestimmte Funktionen
- `?"name"` bzw. `help("name")` für Hilfe bei nicht-konventionellen Zeichen, z.B. `?"+`
- Default der Hilfsfunktion `help` ist es, nur in geladenen Paketen zu suchen. Mit der Option `try.all.packages=TRUE` kann in allen installierten Paketen gesucht werden. Die Hilfeseite eines noch nicht geladenen Pakets kann dann mit der Option `package` angezeigt werden.
- `help.search("name")` für Volltextsuche
- PDF-Handbücher über das Hilfemenü
- `help.start()` für Hilfe im html-Format
- `RSiteSearch("name")` durchsucht online Rhelp-Archives
- `apropos(name)` liefert alle Funktionen, deren Namen `name` beinhalten (Topiksuche). Nur geladene Pakete werden durchsucht.
- `example(name)`

6) Programmieren

- Verschiedene statistische Funktionen können zu einem einzigen Programm zusammengestellt werden, das einerseits komplexere Analysen erlaubt und zum Anderen wiederholt aufrufbar ist.
- Speicherformat ist ASCII mit Dateierdung `'.R'` oder auch `'.r'` (=> Syntax Highlighting)
- `"#"` kennzeichnet Kommentare
- Genereller Aufbau:

```
anyfun <- function(arguments) {  
  ## function body ##  
}
```
- Die Argumente können Objekte sein, deren Übergabe per Position oder Namen erfolgt. Die Übergabe von vordefinierten Argumenten ist optional.
- zum Gültigkeitsbereich von Variablen siehe `RSiteSearch("lexical scoping")`
- Rückgabe von Werten über `return` oder `invisible`
- Schleifen: `for`, `while`; => effizientere Programmierung über Nutzung der Vektorstruktur
- Conditional Expressions: `if`, `if ... else`
- `source` zum Laden eines selbstgeschriebenen Programms
- Paket `debug` zur komfortablen Fehlersuche und -bereinigung

7) Statistische Analysen

- (nicht-) lineare Modelle, klassische Tests, Zeitreihenanalyse, Klassifikation, Clustering, etc.
- Beispiel für lineares Modell `lm` mit Formelobjekt (`?formula`)

8) Grafiken

- wesentliche Grafikfunktionen: `plot`, `persp`, `image`, `contour`
- Ergänzungen mit `points`, `lines`, `abline`
- verschiedene Grafikformate wie JPEG, Postscript oder PDF sind möglich (`?Devices`)
- grafische Parameter über `?par` abfragen sowie zur Strukturierung `?layout`
- `mtext`, `text` fügt zusätzlichen Text ein, `legend` eine Legende; flexible Gestaltung der Achsen über `axis`
- Anwendung von `plot` auf das lineare Modell Beispiel
- Symbole und mathematische Formeln mit `expression` darstellbar (Beispiele dazu in `?text`)

9) Quit R

- save working space; `.Rdata`; `.Rhistory`

10) Einstellungen der RGUI verändern

R/Edit/RGUI/Einstellungen:

- SDI (single document interface) reduziert die GUI auf das Fenster der R Console
- `rows`, `columns` erlauben Spezifikation der Fensterhöhe und -breite
- alle Modifikationen speichern, und zwar unter dem vorgeschlagenen Dateinamen
- mit `ok` beenden

R mit `quit()` verlassen, da die Einstellungen erst beim Neustart berücksichtigt werden.

Anm.: siehe `?Rprofile` für dauerhafte Startup-Spezifikationen

11) Einstellungen von tinn-R

tinn-R muss im CIP-Pool wegen installationstechnischer Probleme zweimal infolge gestartet werden.

File/Open:

- zum Laden des R Codes ins Editorfenster mit unmittelbarem Highlighting der Syntax;
- Highlighting von C/C++ Code und vielen andere Sprachen (html, latex) ist ebenso möglich.

Options/Main/Application/Main:

- `active line highlighted`

Options/Main/Editor/Display:

- `Edge Columns` zur Anpassung der Breite des Editorfensters
- `show line numbers`

R/Send to R:

- Falls diese Funktionalität von `tinn-R` noch nicht aktiv ist – etwa weil `tinn-R` vor `R` aufgerufen wurde – gilt es den Pfad zur `RGUI` zu spezifizieren. Hierzu entweder über `R/RGUI/Initiate Preferred` oder `Options/Main/Application/R` gehen und `set path to preferred RGUI` spezifizieren (L:/R_2/R-2.2.0/bin/RGUI.exe im CIP-Pool).

R/Hotkeys of R:

- Bsp: Shortcut für `send line`
Cursorfeld auf `send line` bewegen; `Ctrl-l` auswählen, auf Verfügbarkeit prüfen und gegebenenfalls hinzufügen; `Active` und mit `ok` beenden

Weitere Funktionalität:

- `toggle tools (in)visible` - Icon erlaubt das Ein- bzw. Ausblenden des Zusatzfensters "Computer/Project/RCard"
- `return focus after sending to R` - Icon belässt den Cursor in `tinn-R` bzw. bei nicht aktiviertem Icon springt der Cursor in die `R Console`
- `on top` - Icon lässt `tinn-R` über allen anderen Programmfenstern sichtbar
- U.a. bietet `tinn-R` auch einen Versionsvergleich von Dateien an (`Tools/Differences`).
- Die persönlichen Einstellungsvorlieben können unter `Tools/Backup` gespeichert und bei der nächsten Sitzung wiederhergestellt werden.