

Einführung in R

(Dank an Markus Kreuz)

Was ist R:

- Freie Statistik Software (GNU-Lizenz)
 - Orientiert sich an S/S-Plus
- Für viele Plattformen verfügbar
 - Windows
 - Linux/Unix
 - Macintosh
- Entwickler: Ross Ihaka und Robert Gentleman
- Aktuelle Version: 2.8.0
 - 2 neue Versionen pro Jahr entwickelt

Vorteile von R:

- Kostenlos
- Umfangreiche Funktionalität zur graphischen Darstellung
 - Export in verschiedenste Graphikformate
- Entwicklung eigener Pakete möglich
 - Vielzahl verschiedener Pakete zu unterschiedlichsten Anwendungsgebieten verfügbar
 - Quellen für Pakete/Dokumentation:
 - <http://www.r-project.org/>
 - <http://www.bioconductor.org/>
 - <http://rosuda.org/lehre/WS05/R-Vorlesung.shtml>

Grundlagen R:

- R starten: [Link](#)
- Windows Version enthält Standard GUI
 - Bessere GUI oder Editor (Emacs/WinEdt) für komplexere Analysen notwendig
- R arbeitet mit REPL (read, evaluate, print-loop)

Datentypen / Konstanten:

- Skalar

```
x <- 1
```

```
x = 1
```

- Vektor

```
x <- c(1, 2, 3, 4, 5)
```

- Typische Operatoren können auf Vektoren angewendet werden:

```
y <- x * 2
```

```
z <- x + y
```

- Wichtige Konstanten:

```
TRUE, FALSE, NA, NaN, pi, Inf, NULL
```

Operatoren:

- +
- -
- *
- /
- ^ (Potenz)
- %/% (Ganzzahlige Division)
- %% (Modulo)
- == (Gleichheit)
- != (Ungleichheit)
- &, |, ! (AND, OR, Negation)
- sqrt(x) (Quadratwurzel = $x^{1/2}$)
- abs(x) (Betrag x)
- choose(n,k) (n über k)
- factorial(x) (x!)

Grundfunktionen:

- `mean()` (Mittelwert)
- `median()` (Median)
- `sd()` (Standardabweichung)
- `quantile()` (Quantile)
- `cor()` (Korrelation)
- `round()`, `floor()`, `ceiling()` (Runden, Ab/Aufrunden)
- `exp(x)` (e^x)
- `log()`, `log10()`, `log2()` (Logarithmieren)
- `sin()`, `cos()`, `tan()` (Trig. Funktionen)
`asin()`, `acos()`, `atan()`
- `sum()`, `max()`, `min()` (Summe, Maximum, Minimum)
- `length()` (Länge des Vektors)
- `sort()` / `order()` (Sortierung Vektor)

Operationen mit Vektoren:

- Erzeugen von Vektoren:

- `c(1,2,3)` (Erzeugt Vektor: {1,2,3})
- `rep(x,y)` (Wiederholt x y-mal)
- `seq()` (Erzeuge Sequenz)
`x:y`

- Verknüpfung:

- `x<-c(1,2,3); c(x,x)`
- `x*x, ...`

- Selektion von Elementen:

- `x[4]` (ein Element)
- `x[2:3]` (mehrere Elemente)
- `x[x>5] / x[!is.na(x)]` (logische Selektion)
- `x[-1]` (Exklusion)

Funktionen für Verteilungen:

- Erzeugen von Vektoren:
 - Dichte: `dVerteilung()` (z.B. `dnorm()`)
 - Verteilung: `pVerteilung()` (z.B. `pnorm()`)
 - Stichprobe: `rVerteilung()` (z.B. `rnorm()`)
 - Quantile `qVerteilung()` (z.B. `qnorm()`)
- Verfügbare Verteilungen:
 - `norm` (Normalverteilung)
 - `unif` (Gleichverteilung)
 - `binom` (Binomialverteilung)
 - `gamma` (Gammaverteilung)
 - `pois` (Poissonverteilung)
 - `mvnorm` (Multivariate Normalverteilung)
 - `chisq` (Chi²-Verteilung)
 - `t` (t-Verteilung)
 - ...

Graphikfunktionen:

- Standardmäßig wird in interaktives Bildschirm Device geplottet
- `png()` / `bmp()` / `pdf()` / `postscript()` / ... (Datei Device)
 - `dev.off()` (Schließt Device)
- `plot()` (Allgemeine Zeichenfunktion)
 - `lines()`
 - `points()`
 - `legend()`
- `hist()` (Histogramm)
- `pie()` (Kreisdiagramm)
- `boxplot()` (Boxplot)

Weitere wichtige Operatoren/Funktionen:

- `is.na()` (Prüft ob Werte = NA sind)
- `is.null()` (Prüft ob Werte = NULL sind)
- `density()` (Dichte von Vektor Schätzen)
- `#` (Kommentarzeichen)
- `if (x==y){` (If / else Schleife)
 `...`
 `}else{...}`
- `for (i in x){` (For-Schleife)
 `}`
- `matrix()` (Erzeugt Matrix)
- `dim()` (Dimension von Objekten zurückgeben)
- `apply()` (Funktion auf Zeilen/Spalten von Matrix anwenden)
- `attributes()` (Elemente von zusammengesetzten Datentypen zeigen)
- `$` (Zugriff auf Elemente von zusammengesetzten Objekten)

Weitere wichtige Operatoren/Funktionen (2):

- `?Funktion` (Hilfe zu Funktion)
- `ls()` (Zeige Objekte im Workspace)
- `rm()` (Lösche Objekt aus Workspace)
- `save()` (Speichere Objekte ab)
- `save.image()` (Speichere kompletten Workspace)
- `read.table()` (Lese Daten aus Textdatei z. B. tab-separiert)
- `write.table()` (Schreibe Daten in Textdatei)
- `library()` (Lade Bibliothek (muss bereits installiert sein))
- `sample()` (Ziehe Stichprobe aus Vektor (mit/ohne replace))
- `function()` (Definition von eigenen Funktionen)
 - `testfunc<-function(x,y){`
 ...
 `return(z)`
 `}`

Statistische Tests:

- `t.test()` (t-Test)
- `fisher.test()` (Fisher-Test)
- `chisq.test()` (Chi²-Test)
- `binom.test()` (Binomialtest)
- `wilcox.test()` (Wilcoxon-Test, Ein- und Zweistichproben-Fall)
- `kruskal.test()` (Kruskal-Wallis-Rangsummen-Test)
- `var.test()` (F –Test für den Vergleich von Varianzen)
- ...

Aufgaben:

- (1) Schreiben Sie eine Funktion, die den Median ermittelt ohne die Funktion `median()` zu verwenden.
- (2) Schreiben Sie ein Skript, dass N Stichproben $\sim N(0, \gamma)$ der Größe m zieht und jeweils einen t-Test gegen Null durchführt. Stellen Sie die p -Werte graphisch dar und ermitteln Sie die Anzahl signifikanter ($p < 0.01$) Stichproben. Erzeugen Sie die Boxplots zu den signifikanten Stichproben.
- (3) Schreiben Sie ein R-Skript, mit dem Sie Zufallszahlen einer Binomialverteilung mit den Parametern $n = 15$ und $p = 0.2$ erzeugen können, ohne dabei die Funktion `rbinom()` zu verwenden.