

1 Basics

1. (1) Intro
2. (2) Operatoren
3. (4) Variablen (typen, gültige Namen)
4. (6) atomare vektoren (seq, rep, indizierung, NA)
5. (11) text (print, cat)
6. (12) funktionen
7. (13) listen (str, c, unlist, as.list)
8. (16) if, while, for (seq_along, letters)
9. (18) Zufall
10. (20) summary statistics (mean, median, var, sd, max, which.max, summary)
11. (21) Matrizen
12. (23) basegraphics plot
13. (26) pakete
14. (27) tibble (erstellung, indizierung,
15. (28) apply Funktionen (lapply, sapply, apply(mat, 1/2, f)
16. (29) Statistik

2 Datenstrukturen

1. (1) Objekte (typeof, class, length)
2. (2) Vektoren
 - atomare (basistypen, initialisierung BASETYPE(n), typeof)
 - logical(3), integer(4), double(4), character(6) (letters, month.name)
 - coercion(8)
 - listen(8) (initialisierung list())
3. (10) Attribute (attr, attributes, structure, str)
names (11) (atomare vektoren oder listen)
4. (14) Matrizen und Arrays (dim, nrow, ncol, rbind, cbind, abind::abind())
5. (19) S3-Objekte (factor, date time zeug)
6. (22) Tibbles
7. (25) Na zeug

3 Subsetting

1. (1) Intro
2. (1) mehrere Elemente
 - 2.1(1) atomare Vektoren
 - 2.2(4) Arrays
 - 2.3(8) Tibbles
3. (9) einzelne Elemente ([[, \$]])
4. (11) fehlende indizes
5. (12) subsetting und zuweisung, out-of-bounds-zuweisung
6. (15) Anwendungen
 - (15) Matching und Merging
 - (16) Random Samples
 - (17) Sortieren (sort, order, rank)
 - (18) Aggregierte Zeilen expandieren
 - (18) Spalten aus Tibble entfernen
 - (19) Konditionales Subsetting
 - (19) Boolean/Integer Subsetting (which)

4 Funktionen

1. (1) nomenklatur, syntax
2. (3) Funktionsaufruf (präfix, infix, ersetzungsfunktionen, vordefinierte spezialfunktionen, do.call)
3. (7) Klassifizierung (class function, typeof closure/builtin/special)
4. (8) Komponenten
5. (9) Funktionsargumente (lazy evaluation, default arguments, missing arguments, ...)
6. (12) Beenden einer Funktion

5 stringr

1. (1) Intro
2. (1) Basics (str_length, str_c, str_sub, str_trim, str_squish, str_to_upper, ...)
3. (3) Formatieren (str_glue, str_glue_data, ...)
4. (4) regexp (str_view, str_view_all)
 - (5) Funktionen (str_detect, str_subset, str_which, str_count)
 - (7) Syntax (siehe stringr.cheatsheet.pdf)

6 tibble

1. (1) Tabellen (erstellen, as_tibble, add_column, add_row, subsetting, bind_rows, bind_cols)
2. (6) Relationale Datenbanken (union, intersect, setdiff, crossing)

7 dplyr

1. (1) Verben für eine Tabelle (filter, arrange, select, mutate, summarize, pipe, group_by, column/row-wise operations: across/rowwise, non-standard-evaluation)
2. (23) Verben für zwei Tabellen (mutating: left/right/full/inner_join, filtering: semi/anti_join)
3. (28) Relationale Datenbanken

8 tidyr

- 1.1(3) pivoting
- 1.2(5) separate and unite
- 1.3(7) Fehlende Werte
- 2(9) Normalisierung Relationaler Datenbanken (mega unnötig)

9 Umgebungen

1. (2) Umgebungsobjekte (rlang ist auch klausurrelevant)
2. (6) Umgebungsdiagramme
3. (8) Funktionsumgebungen (funktions-, aufrufende, ausführungsumgebungen)
4. (14) Scoping-Prinzipien (überdeckung, dynamic lookup, ausführungsumgebung)
5. (17) Zuweisungsoperatoren (i-, ii-, -i, -ii, =)
6. (18) Elternumgebungen, suchpfad
7. (19) paketumgebungen
8. (22) Übersicht
9. (22) Lazy evaluation

10 Funktionale Programmierung

1. (1) Funktionale Programmierung, rein(3),
2. (5) Funktionale (apply-funktionen, rep, outer(12), integrate, optim)

3. (15) Funktionsfabriken (Counter, MLS, Interpolation)
4. (22) Funktionsoperatoren

11 Objektorientierte Programmierung

1. (1) Intro
2. (3) S3
3. (13) S4
4. (18) R6

12 Meta-Programmierung

1. (2) Ausdrücke, Syntaxbaum, parse, base-R
2. (7) Ausdrücke verwenden, enexpr, eval (mit Umgebung), quosures(9)
3. (11) Anwendung
4. (12) tidy-eval-framework, !!, !!!, , :=