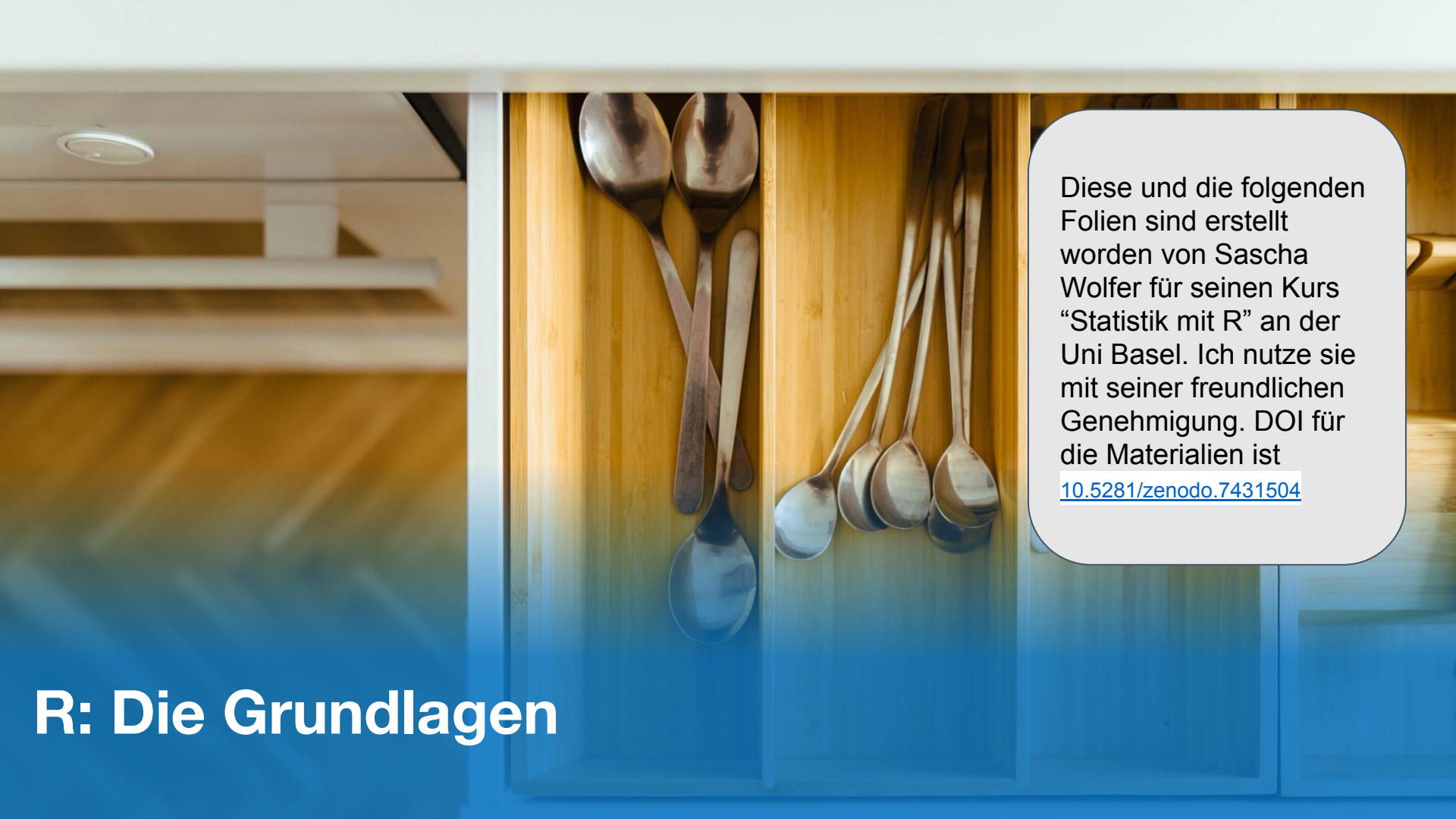
Einführung in die Statistik

Praktische Übung – Jürgen Hermes – IDH – SoSe 2023

R – Die Grundlagen

- **R und RStudio-Setup**
- **Einfache Datentypen in R**
- Komplexe Datentypen in R
- Indizierung
- Funktionen und Prädikate
- Zusatzpakete
- Daten einlesen und speichern
- Daten visualisieren

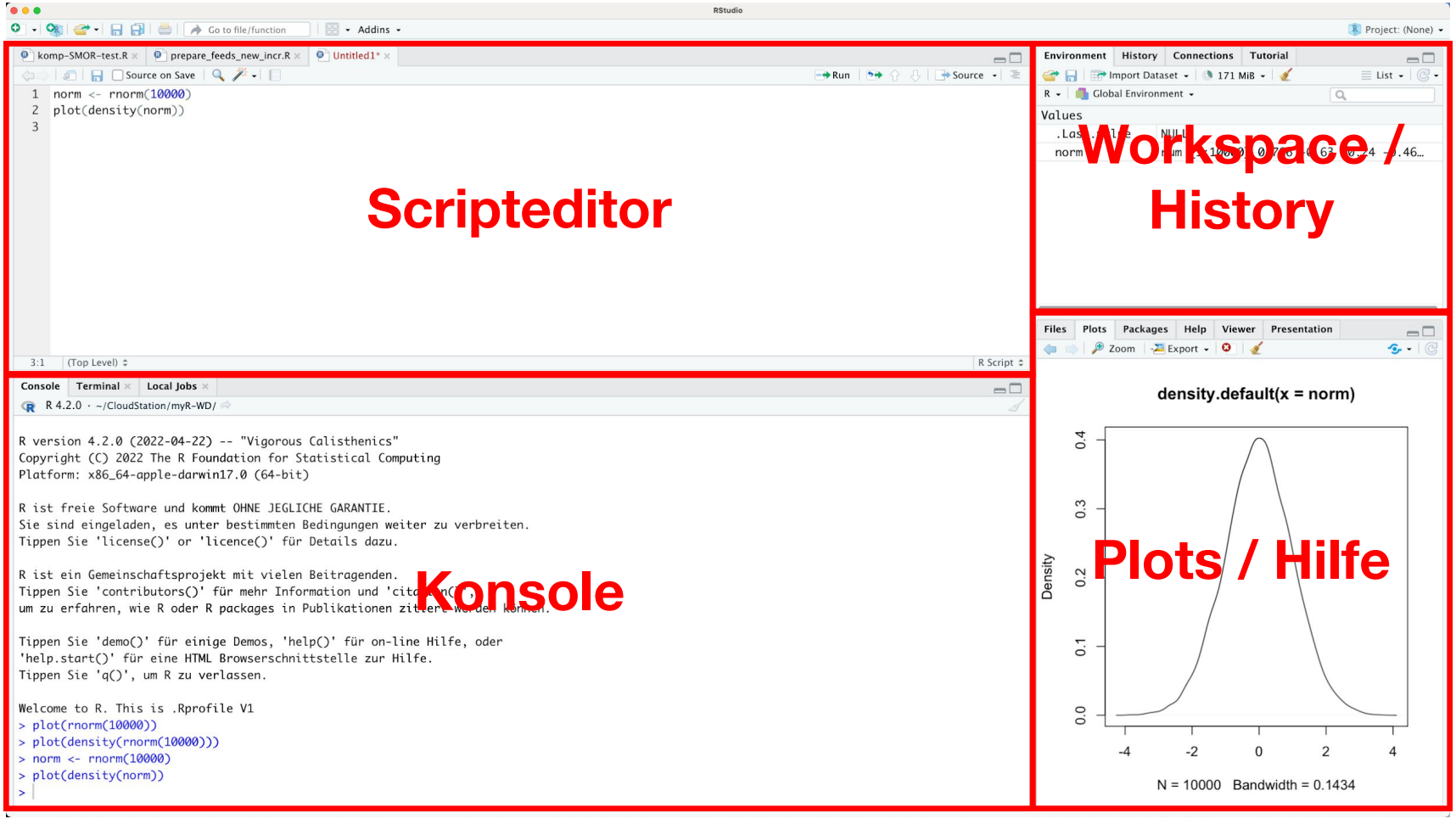


R: Die Grundlagen

Diese und die folgenden Folien sind erstellt worden von Sascha Wolfer für seinen Kurs “Statistik mit R” an der Uni Basel. Ich nutze sie mit seiner freundlichen Genehmigung. DOI für die Materialien ist 10.5281/zenodo.7431504

R und RStudio

- R: Programmier- / Scriptsprache
- RStudio: IDE (= integrated development environment)
  - erleichtert uns die Arbeit mit R
  - u.a. mit Scripteditor, Workspace-Übersicht, integrierter Hilfe, Speichern von Abbildungen, Syntax Highlighting usw.
- R ist verfügbar auf CRAN (= comprehensive R archive and network): https://cloud.r-project.org
- RStudio: www.rstudio.com ➜ Download


RStudio
Go to file/function
Addins
Project: (None)
komp-SMOR-test.R
prepare_feeds_new_incr.R
Untitled1*
Source on Save
Run
Source
1 norm <- rnorm(10000)
2 plot(density(norm))
3
Scripteditor
3:1
(Top Level)
R Script
Environment
History
Connections
Tutorial
Import Dataset
171 MiB
List
R
Global Environment
Values
Workspace / History
Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
density.default(x = norm)
Density
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
N = 10000 Bandwidth = 0.1434
Plots / Hilfe
Console
Terminal
Local Jobs
R 4.2.0 · ~/CloudStation/myR-WD/
R version 4.2.0 (2022-04-22) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)
R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.
R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()'
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.
Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.
Welcome to R. This is .Rprofile V1
> plot(rnorm(10000))
> plot(density(rnorm(10000)))
> norm <- rnorm(10000)
> plot(density(norm))
>
Konsole

RStudio-Oberfläche

- Sammeln Sie Ihre Befehle im **Scripteditor**!
  - Speichern Sie Ihre Scripts mit der Endung `.R`
  - Schicken Sie Befehle an die Konsole mit Cmd+Enter / Strg+Enter
- **Kommentieren** Sie Ihren Code mit `#`
  - Alles hinter `#` wird nicht von R ausgewertet.
- In der **Konsole** finden Sie die Rückgabewerte von R.
  - Ergebnisse Ihrer Befehle
  - Nicht alle Befehle haben Rückgabewerte.
- Die **Hilfe** rufen Sie mit `?<Befehl>` auf.
  - Beispiel: `?mean`

Übung E02U01

- Berechnen Sie in R das Ergebnis:
  - 1848 geteilt durch 44
  - 55,5 geteilt durch 12 (Dezimaltrennzeichen in R ist der Punkt)
  - Quadratwurzel aus 36 hoch 3 – Quadratwurzel: Funktion `sqrt(x)`
  - Mittelwert der Zahlen 20, 45 und 1, ohne die Funktion `mean(x)`

Datentypen

- Programmiersprachen brauchen fest definierte Datentypen, damit jederzeit klar ist, was mit bestimmten Objekten gemacht werden kann und was nicht.

Elementare Datentypen

- Zahlen
- Zeichenketten
- Wahrheitswerte

Komplexe Datentypen

Zusammengesetzt aus anderen Elementen

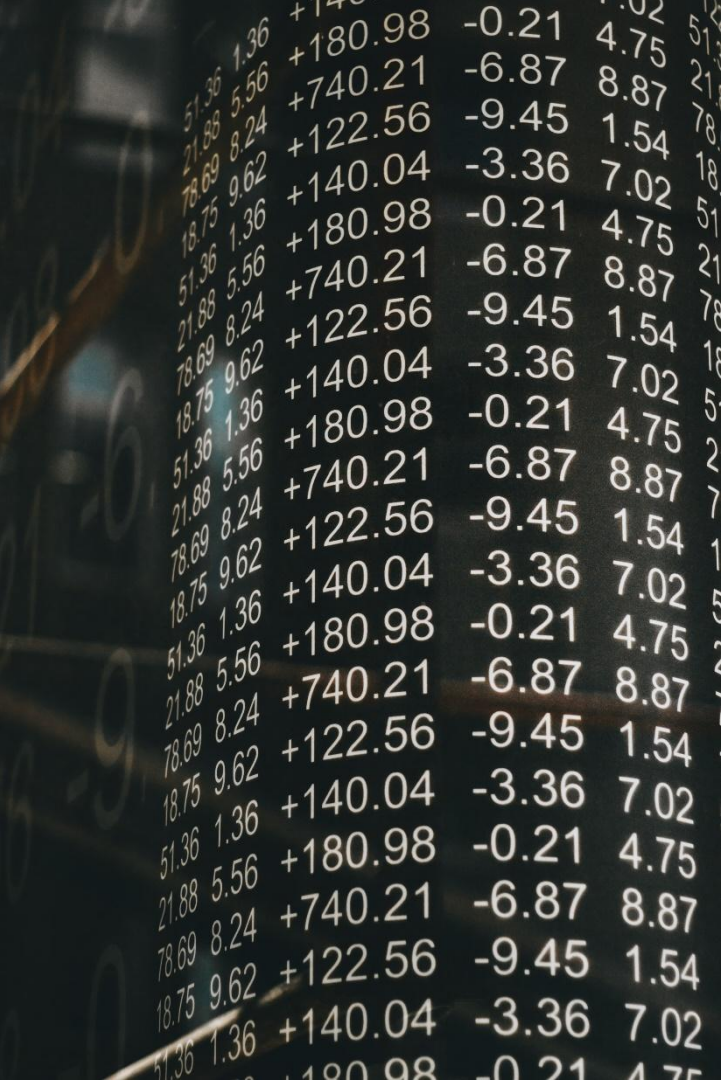
- Vektoren
- Dataframes
- (Matrizen)
- (Listen)

Elementare Datentypen in R

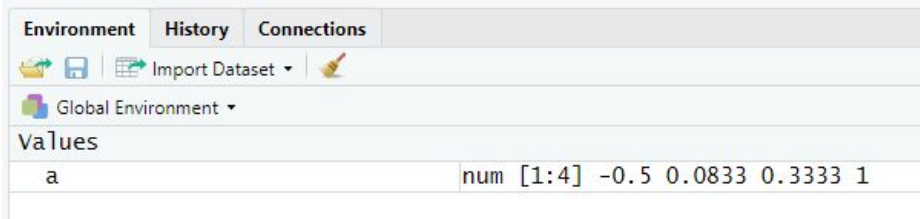
Zahlen
numerical / integer

Zeichenketten
character / string

Wahrheitswerte
logical / boolean



- Alle numerischen Werte
- Spezialfälle (auch Zahlen!):
  - `NaN`: *not a number*
  - `Inf` / `-Inf`: (minus) unendlich
- Abkürzung: `num` für numerical
  - oder: `int` für integer / ganze Zahlen

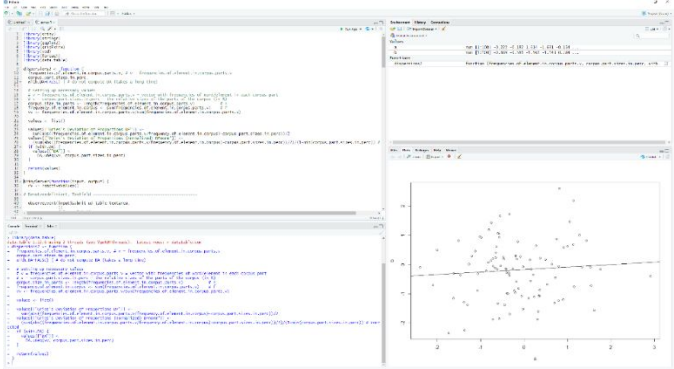


- Alle Zeichen(folgen)
- Stehen in Anführungszeichen
  - `"42"` ist eine Zeichenkette, keine Zahl
    - 42 ist eine Zahl
  - `""` ist eine leere Zeichenkette
  - `" "` ist eine Zeichenkette, die ein Leerzeichen enthält.
- Abkürzung: `chr` *(character)*

Wahrheitswerte

- Auch: *boolean* / *logical* (Abkürzung `logi`)
- `TRUE` (wahr) oder `FALSE` (falsch)
  - Können abgekürzt werden mit `T` und `F`.
- Repräsentieren "ja" und "nein" bzw. "wahr" und "falsch"
  - Werden oft von R als Antworten auf Fragen zurückgegeben.
  - `5 < 10` entspricht TRUE bzw T
- `"T"` ist eine Zeichenkette, `T` ist ein Wahrheitswert.

Variablen / Objekte



- In Variablen speichern wir Werte,
  damit wir sie wiederverwenden können.
- Alle Variablen erscheinen im Environment von RStudio.
  - Auch: *workspace*
- Zuweisungsoperator: `<-`
  - `a <- 3` speichert in der Variable `a` die Zahl `3`.
  - `b <- "drei"` speichert in der Variable `b` die Zeichenkette `"drei"`.
- Nennen Sie Variablen, wie Sie wollen, aber…
  - kein Variablenname darf mit einer Zahl beginnen.
  - keinen Bindestrich im Namen verwenden (= minus).

Verändern von Variablen

- Achtung! Eine Variable verändert sich immer nur dann, wenn Sie das explizit schreiben (= "nicht-destruktive" Operationen).

```
a <- 2
a + 4
[1] 6
a
[1] 2
a <- a + 4
a
[1] 6
```

Umwandeln von Objekten

- Manche Objekte können verlustfrei in einen anderen Typ umgewandelt werden.
- Grundsätzlich gilt: Alles kann verlustfrei in eine Zeichenkette umgewandelt werden: `as.character(x)`
- In Zahl umwandeln: `as.numeric(x)`
- In Wahrheitswert: `as.logical(x)`

Übung E02U02

- Beobachten Sie bei jedem Schritt der Übung, was geschieht!
  - Speichern Sie in der Variable `test` die Zeichenkette `"1"`.
  - Addieren Sie `test` mit 3.
  - Wandeln Sie `test` in eine Zahl um.
  - Addieren Sie `test` mit 3 und speichern Sie das Ergebnis wieder in `test`.
  - Ziehen Sie 4 von `test` ab (und speichern Sie das Ergebnis wieder in `test`).
  - Wandeln Sie `test` in einen Wahrheitswert um.

Begriffe

- Scripteditor
- Konsole
- Workspace
- #
- Datentypen
- Elementare D.~
- Komplexe D.~
- Zahlen
- Zeichenketten
- Wahrheitswerte
- Variablen / Objekte

Zusammenfassung

- Schreiben Sie Ihre Scripts in den Scripteditor, kommentieren Sie (viel!) mit `#`.
- In der Konsole finden Sie die Rückgabewerte Ihrer Befehle.
- Elementare Datentypen: Zahlen, Zeichenketten, Wahrheitswerte.
- In Variablen können wir mit `<-` Werte speichern. Wenn Sie eine Variable verändern möchten, müssen Sie neu zuweisen.
- Sie können elementare Datentypen mit den Funktionen `as.` … ineinander umwandeln.
    - Alles kann verlustfrei in Zeichenketten umgewandelt werden.

Fragen?

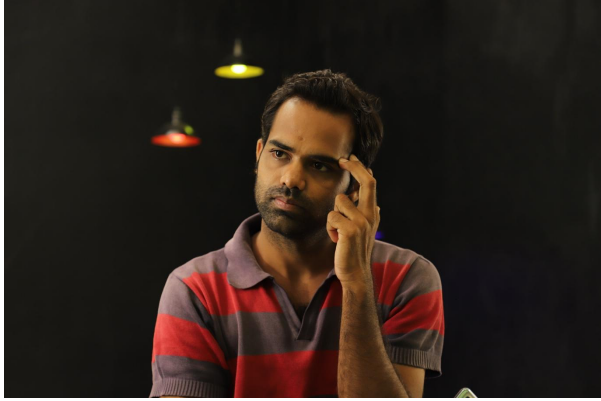
Komplexe Datentypen

- **Vektoren**
- **Dataframes**
- Matrizen
- Listen

Vektoren

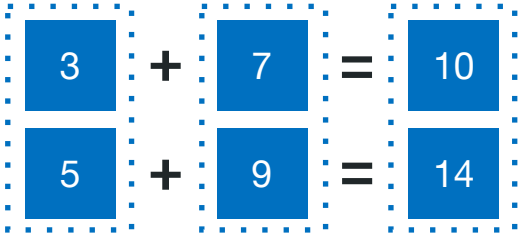
- Vektoren sind Reihen von Elementen **eines** Typs.
  - Numerische Vektoren
  - Character-Vektoren
  - Logische Vektoren
  - + Faktor-Vektoren
- Vektoren sind **geordnet**, die Reihenfolge ist wichtig.

Vektoren



- Erstellen Sie Vektoren mit `c()` – *construct*
  - `c("a", "b", 3)` → `[1] "a" "b" "3"`
- Andere Funktionen/Operatoren geben ebenfalls Vektoren zurück.
  - `5:9` → `[1] 5 6 7 8 9`
  - `seq(5, 8, 0.5)` → `[1] 5.0 5.5 6.0 6.5 7.0 7.5 8.0`
  - `rep(T, 5)` → `[1] TRUE TRUE TRUE TRUE TRUE`
- Sie können Vektoren miteinander verrechnen.
  - `c(3, 5) + c(7, 9)` → `[1] 10 14`

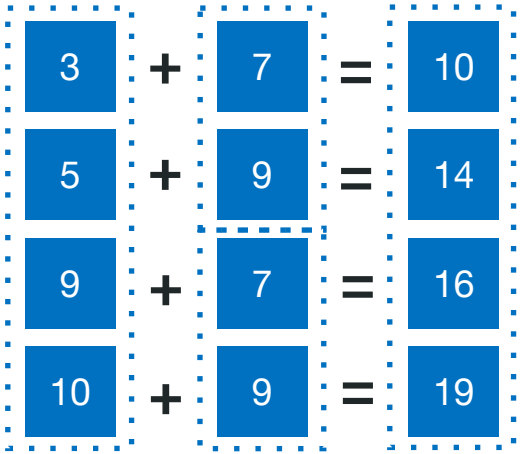
Recycling



`c(3, 5) + c(7, 9)`

Vektoren werden – falls nötig – *recyclet*. Sie werden also so lange wiederholt, wie es nötig ist.

`c(3, 5, 9, 10) + c(7, 9)`



Recycling: Achtung!

Das ist keine Fehlermeldung!

```
> c(3,5,9,10) + c(7,9,10)
[1] 10 14 19 17
Warnmeldung:
In c(3, 5, 9, 10) + c(7, 9, 10) : Länge des längeren Objektes
        ist kein Vielfaches der Länge des kürzeren Objektes
```

- R möchte Vektoren gerne komplett recyclen.
  - Hier kann der kürzere Vektor aber nur zum Teil recyclet werden.
- Wenn das geschieht, wird eine Warnmeldung ausgegeben.
  - Bei **Warn**meldungen wird trotzdem ein Ergebnis zurückgegeben, bei **Fehler**meldungen nicht.

Faktor-Vektoren

Um zu verstehen, was Faktor-Vektoren sind und wozu sie da sind, benötigen wir Kenntnisse über **Skalenniveaus**.

Skalenniveaus

Verhältnisskala

Intervallskala

Metrische Skala

Ordinalskala

Nominalskala

Nominalskala

- Notiert werden die **Gleich- und Verschiedenheit** von Elementen.
  - Keine Rangfolge
  - Keine Abstände
- Möglich: Auszählen von Häufigkeiten
- Beispiele: Wort-Type, Erstsprache, Korpusquelle
- Sonderfall: **Binäre** Skala
  (richtig/falsch, vorhanden/nicht vorhanden)

Ordinalskala

- **Zusätzlich** ablesbar: Rangfolge von Elementen
  - Keine Abstände
- Wir können zusätzlich alle Analysen/Maßzahlen anwenden, die auf der Rangfolge beruhen (z.B. **Median**).
- Beispiele: Ränge beim Militär (General > Oberst > Feldwebel > Gefreiter),
  Beurteilungen (sehr unzufrieden < unzufrieden < zufrieden < sehr z.)
  Schulnoten (sehr gut > gut > ... > ungenügend)

Intervallskala

- **Zusätzlich** können wir ablesen, wie weit Datenpunkte auf der Skala voneinander entfernt sind.
  - Kein absoluter Nullpunkt ➔ keine Verhältnisse
- Wir können alle relevanten statistischen Verfahren anwenden.
- Beispiele: IQ, Temperatur in Celsius

Verhältnisskala

- Einziger Unterschied zur Intervallskala: Absoluter Nullpunkt
- **Zusätzlich** können wir Aussagen über Verhältnisse machen.
  - "X ist doppelt so schwer wie Y."
- Unterschied in der Praxis (fast) nie relevant, daher Zusammenfassung von Intervall- und Verhältnisskala als **metrische Skala**
- Beispiel: Gewicht, Reaktionszeit, Temperatur in Kelvin

Nominalskala / Ordinalskala / Metrische Skala?

- Länge des Weges von der Haustür zur Uni
- Sprachkenntnisse (Niveau des europäischen Referenzrahmens)
- Studienfachkombination im BA
- Gewicht des Rucksacks, den Sie mit sich tragen
- Zeit, die für das Lösen einer Programmieraufgabe benötigt wird
- Zufriedenheit mit einem bestimmten Seminar
- Anteil des Studiums an der wöchentlichen Lebenszeit

Jetzt aber: Faktor-Vektoren

```
> factor(c("A", "A", "B", "C", "C", "C"))
[1] A A B C C C
Levels: A B C
```

- Faktorvektoren werden in R dazu verwendet, **nominal-** und **ordinalskalierte** Daten zu notieren.
- Ordinal: Sonderfall *ordered factors* – erstellen mit `ordered(x)`
  - *Ordered factors* braucht man sehr selten.
- *Levels*: Alle Stufen eines Faktors

Übung E02U03 - Faktor-Vektoren: Achtung!

- Weisen Sie einer Variable den folgenden numerischen Vektor zu:
  - `[42, -1, 8, 99]`
- Wandeln Sie die Variable in einen Faktor-Vektor um. Sie benötigen dazu die Funktion `as.factor()`. Geben Sie die Variable aus.
- Wandeln Sie die Variable mit `as.numeric()` wieder zurück in einen numerischen Vektor um.
- Geben Sie die Variable wieder aus. Was fällt Ihnen auf?

Begriffe

Vektoren

Faktor-Vektoren

Recycling

Warnmeldung

Fehlermeldung

Skalenniveau

Nominalskala

Ordinalskala

Metrische Skala

Binäre Skala

Vektoren: Zusammenfassung

- Vektoren sind ein komplexer Datentyp.
- Vektoren sind geordnete Reihen **eines** elementaren Datentyps.
- Vektoren werden Element für Element miteinander verrechnet.
  - u.U. werden Vektoren dabei recyclet.
  - Nebenbei: Wir müssen unterscheiden zwischen **Fehlermeldungen** und **Warnmeldungen**.
- Faktor-Vektoren kodieren nominal- und ordinalskalierte Variablen und haben Stufen (*levels*).

Fragen?

Hausaufgabe

- Die beste Übung, sich auf das Zwischentestat vorzubereiten, ist die Bearbeitung der Hausaufgabe (Stat_HA_1.R) in ILIAS. Die Aufgaben sind dem R-Kurs vom HPI entnommen. Dies habe ich aus dem Grund gemacht, als dass Sie so auch die dortige Umgebung nutzen können, um Ihre Lösungen zu überprüfen (CodeOcean).
- Die Abgabe wird nicht kontrolliert, Sie sind aber herzlich eingeladen, ihr R-Script auch in ILIAS einzureichen.
- Die Studienleistung wird vergeben aufgrund der Ergebnisse in den Testaten. Sollten Sie *eines* der Testate entschuldigt (Attest!) versäumen, *könnte* ggfs. die regelmäßige, rechtzeitige, vollständige und korrekte Abgabe der HA in die Bewertung mit einbezogen werden statt ein Nachtestat anzusetzen. [Hinweis: Das ist kein Automatismus!]