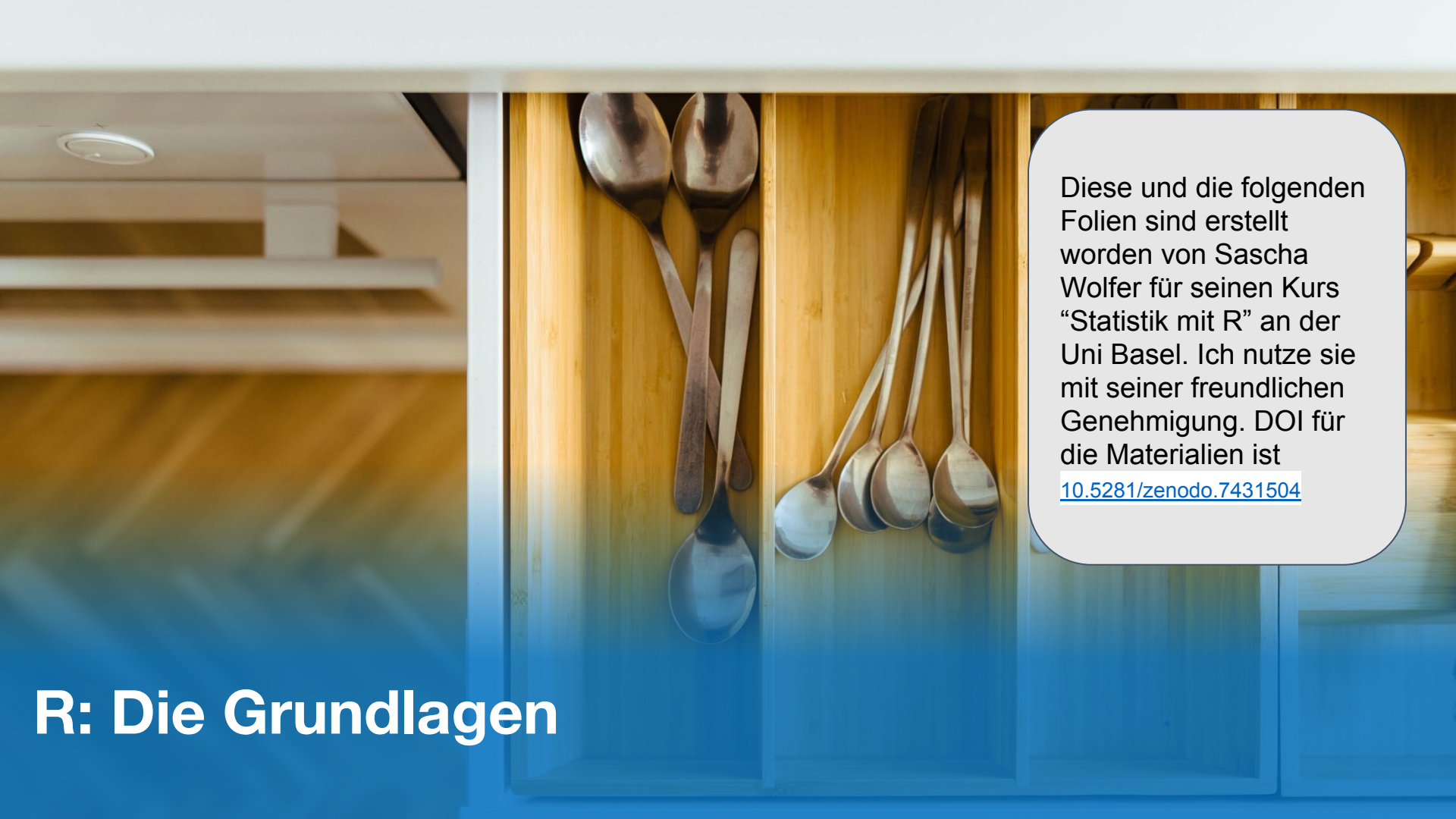
Einführung in die Statistik

Praktische Übung – Jürgen Hermes – IDH – SoSe 2023



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: Die Grundlagen (2. Teil)

- **Dataframes**
- **Funktionen und Prädikate**
- Daten einlesen
- Daten speichern
- Daten visualisieren

Die Programmiersprache R

- Freie, vollständige Programmiersprache, die meist für statistische Berechnungen und Visualisierungen eingesetzt wird.
- Publiziert 1993, aktuelle Version 4.1.3
- Benutzerfreundliche integrierte Entwicklungsumgebung RStudio
- Besonderheiten: Für statistische Berechnungen optimierte Datenstrukturen und Funktionen, besonderes Potential in der einfachen Grafikerzeugung.
- “To understand computations in R, two slogans are helpful: Everything that exists is an object. Everything that happens is a function call.” (John Chambers)
- Extrem große Nutzer:innen-Community, die viele vorgefertigte, nutzbare Pakete zur Verfügung stellt.
- Extrem große Verbreitung innerhalb der Forschungsrichtung Data Science → Das Erlernen von R hilft bei einer Vielzahl potentieller Berufe.

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

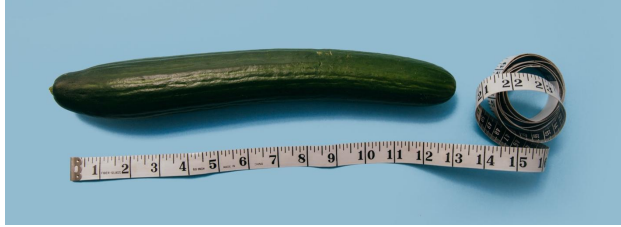
Nominalskala

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

Ordinalskala

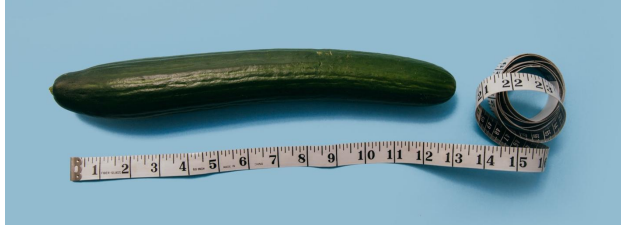
- **Zusätzlich** ablesbar: Rangfolge von Elementen
  - Keine Abstände
- Wir können zusätzlich alle Analysen/Maßzahlen anwenden, die auf der Rangfolge beruhen (z.B. **Median**).
- Beispiele: Ränge beim Militär (General > Oberst > Feldwebel > Gefreiter),
  Beurteilungen (sehr unzufrieden < unzufrieden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

 stark modifizierte Version (Jürgen Hermes)

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

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	Skalenniveau
Faktor-Vektoren	Nominalskala
Recycling	Ordinalskala
Warnmeldung	Metrische Skala
Fehlermeldung	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?

Dataframes

Dataframes kann man als nebeneinander gestapelte Vektoren begreifen.

- Datentabellen: Ein Fall pro Zeile
  - Fall: Versuchsdurchlauf, Wort, Studierender, ...
- In den Spalten stehen Informationen zu jedem Fall.

Gender	Age	Grade	2ndStudyProg
m	19	1.3	History
w	24	1.0	Linguistics
m	23	2.0	Art Hist.
m	18	2.3	History

leicht modifizierte Version (Jürgen Hermes)

Dataframes

- Dataframes sind für alle wichtigen Operationen der Datenanalyse der Ausgangspunkt.
  - Gruppenvergleiche
  - Zusammenhänge
  - Aggregationen
  - Visualisieren
  - ...
- Man kann Dataframes mit `data.frame()` erstellen, meist sind sie aber Ergebnisse von **Einleseoperationen**.
- Mit `head(<dataframe>)` werden die ersten 6 Zeilen eines Dataframes ausgegeben.

Matrizen

- Matrizen (Singular: Matrix) sind **mehrdimensionale Vektoren**.
  - Auch eine Matrix kann *nur einen Datentyp* enthalten!
- Zweidimensionale Matrizen haben auch Zeilen und Spalten.
  - Matrizen können aber beliebig viele Dimensionen haben.

```
> matrix(1:15, ncol = 3)
     [,1] [,2] [,3]
[1,]    1    6   11
[2,]    2    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```

```
> matrix(1:15, ncol = 4)
```

Listen

- Listen sind geordnete Sequenzen von beliebigen Datentypen.
  - Im Gegensatz zu Vektoren können Listen auch komplexe Datentypen enthalten. Listen können also auch wiederum Listen enthalten.

```
> list(5:10, data.frame(A = 1:3, B = c("a", "b", "c")), rep(T,
4))
[[1]]
[1]  5  6  7  8  9 10

[[2]]
  A B
1 1 a
2 2 b
3 3 c

[[3]]
[1] TRUE TRUE TRUE TRUE
```

1. Element: Numerischer Vektor

2. Element: Zweispaltiger Dataframe

3. Element: Vektor mit Wahrheitswerten

Indizierung

Indizierung

- Indizierung = Zugriff; wir greifen auf komplexe Datentypen zu, um Teile von ihnen zu extrahieren oder zu verändern.
- Vektoren: `[n]`
  - `vec[2]`: Zweites Element des Vektors `vec`
  - `vec[3:5]`: Drittes bis fünftes Element von `vec`
- Dataframes: `[<Zeile n>, <Spalte m>]` oder `$<Spaltenname>`
  - `df[5,]`: Fünfte Zeile von Dataframe `df`
  - `df[, 4]`: Vierte Spalte von Dataframe `df`
  - `df[2, 3]`: Element in der zweiten Zeile in der dritten Spalte
  - `df$wort`: Komplette Spalte `wort` (= Vektor)

Indizierung

- Wir können auch über **Namen** zugreifen (wie in `df$wort`):
  - `df[4, "freq"]`: Der vierte Eintrag in der Spalte `freq`
- Wir können außerdem über **Wahrheitswerte** zugreifen (das wird später nochmal wichtig!):

```
> vec <- c("das", "ist", "ein", "vektor")
> vec[c(T, T, F, T)]
[1] "das"    "ist"    "vektor"
```

Hausaufgabe

- Die beste Übung, sich auf das Zwischentestat vorzubereiten, ist die Bearbeitung der Hausaufgabe (Stat_HA_2.R) in ILIAS. Die Aufgaben sind dieses Mal nur zum Teil dem R-Kurs vom HPI entnommen.
- 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!]