

Einführung in R

Daten verarbeiten und analysieren

Tatiana Kvetnaya
Leibniz-Institute for Psychology (ZPID Trier)
Practices and Tools of Open Science

Herzlich willkommen!

Ein kurzes digitales Kennenlernen:

Link: <https://tinyurl.com/r-kennenlernen>



Join at
slido.com
#8783 006

slido



Welcher Gruppe gehörst du an?

slido



Welcher Fachdisziplin gehörst du an?

slido



Wieso nimmst du am heutigen Workshop teil? Was ist deine Motivation, R zu lernen?

Ziele des heutigen Workshops

Was wird in diesem Kurs passieren?

- ✓ Einblick in Grundprinzip und Arbeitsweise von R bekommen
- ✓ Wichtige Basis-Funktionen und Befehle kennenlernen
- ✓ Vorgeschmack auf weitere Möglichkeiten bekommen

Was wird nicht passieren?

- ✗ Workshop als R-Kenner:in wieder verlassen
- ✗ Abgeschlossener Lernprozess

Agenda

<i>Zeit</i>	<i>Abschnitt</i>
16:00	Begrüßung und Kennenlernen
	Teil 1: Grundlagen
16:10	A) Grundlagen: Aufbau von R
16:30	B) Objekte und Aktionen in R
17:00	Übungen zu Teil A und B
17:20	Pause (10-15 Minuten)
	Teil 2: Arbeiten mit Daten
17:35	C) Arbeit mit tabellarischen Daten
18:10	Übungen zu Teil C
18:25	D) Deskriptive Statistik
18:40	Übungen zu Teil D
18:50	Abschlussrunde

Verwendete Literatur und Ressourcen

- Luhmann, M. (2020). **R für Einsteiger:** Einführung in die Statistik-Software für die Sozialwissenschaften. Mit Online-Material (5., überarbeitete Auflage). Beltz.
<http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1809418>
- **pandaR:** Frei verfügbare Lehr- und Lernmaterialien <https://pandar.netlify.app> des Fachbereichs Psychologie der Goethe-Universität Frankfurt



Nutzung von PsychoNotebook

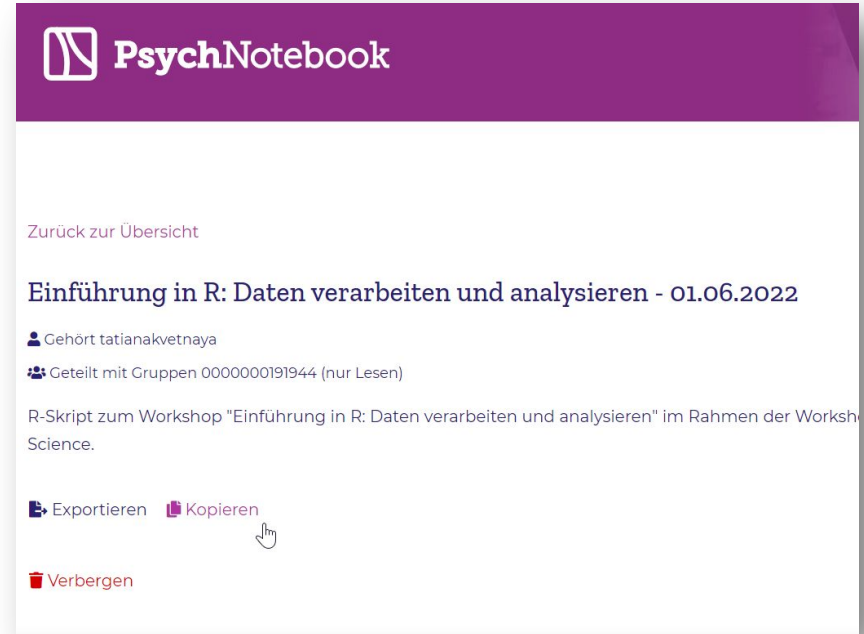
PsychoNotebook ist ein kostenfreies Angebot des ZPID für die Datenanalyse und Planung von Studien.

Darin lassen sich RStudio und andere Software **installationsfrei** und **geräteunabhängig** über den Browser nutzen.

- Account erstellen und einloggen auf psychnotebook.org
- Link zum Projekt: tinyurl.com/r-psychnotebook

RStudio Web in PsychNotebook (1)

- Auf Projektlink klicken
- Das Projekt kopieren (Icon unten rechts anklicken)



RStudio Web in PsychNotebook (2)

- Nun ist eine Kopie des Projekts auf der Projekte-Übersichtsseite im Tab “Meine Projekte” verfügbar.
- Skript öffnen mit Klick auf das kleine R-Zeichen...







Projekte

Hier können Projekte eingerichtet und aufgerufen werden. Projekte sind Sammlungen von Analyseskripten, Daten und anderen Materialien.

[Projekt importieren](#)

[Neues Projekt](#)

[Meine Projekte](#) [Geteilte Projekte](#) [Öffentliche Projekte](#)

Titel	Beschreibung	Anwendungen	Aktionen
Kopie von Einführung in R: ...	R-Skript zum Workshop "Einführung in R: Daten verarbe...	 	   

RStudio Web in PsychNotebook (3)

- ... oder auf den Titel des Workshops klicken, und dann RStudio Web mit dem “Starten”-Button öffnen.

[Zurück zur Übersicht](#)

Kopie von Einführung in R: Daten verarbeiten und analysieren - 01.06.2022

R-Skript zum Workshop "Einführung in R: Daten verarbeiten und analysieren" im Rahmen der Workshop-Reihe Practices and Tools of Open Science.

[✎ Bearbeiten](#)

[📦 Pakete verwalten](#) [↗ Teilen](#) [🌐 Veröffentlichen](#) [📄 Exportieren](#) [📄 Kopieren](#)

 JupyterLab. Eignet sich zur Erstellung von Präsentationen oder Textdokumenten, die ausführbaren Code enthalten.

 RStudio Web. Entwicklungsumgebung für R. Eignet sich zur Inspektion und Analyse von Daten.

[▶ Starten](#)
[🚩 Starten](#)

A

Grundlagen und Aufbau von R

Wieso R?

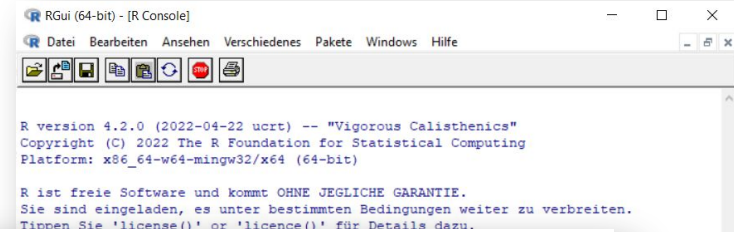
- ist **kostenlos** verfügbar / Open Source
- ist **dynamisch** und flexibel erweiterbar durch zahlreiche **Pakete**
- hat eine lebendige **Community** von Usern, die sich im Internet über R austauschen und das Angebot (z.B. neue Pakete) weiterentwickeln
- ist geeignet, um **reproduzierbare Skripte** zu erstellen und diese mit anderen zu teilen – und damit ein besonders nützliches Open-Science-Tool!

www.r-project.org

R und RStudio

- **R**: Programmiersprache, kommandozeilenbasiert
- **RStudio**: Software-Umgebung bzw. User Interface, das die Arbeit mit R zugänglicher und benutzerfreundlicher macht

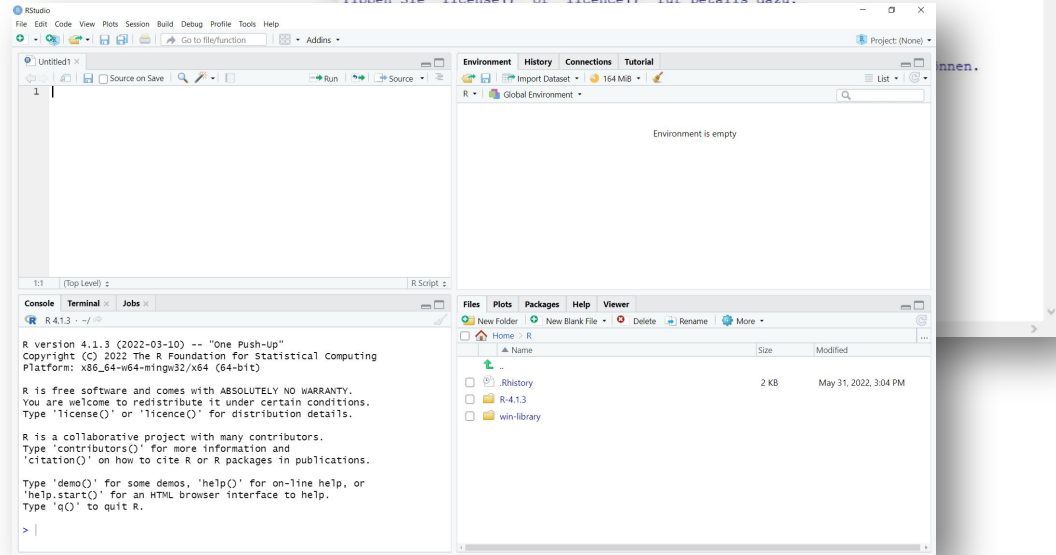
*R-Interface
(Konsole)*



```
RGUi (64-bit) - [R Console]
Datei Bearbeiten Ansehen Verschiedenes Pakete Windows Hilfe

R version 4.2.0 (2022-04-22 ucrt) -- "Vigorous Calisthenics"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.
```



RStudio-Interface

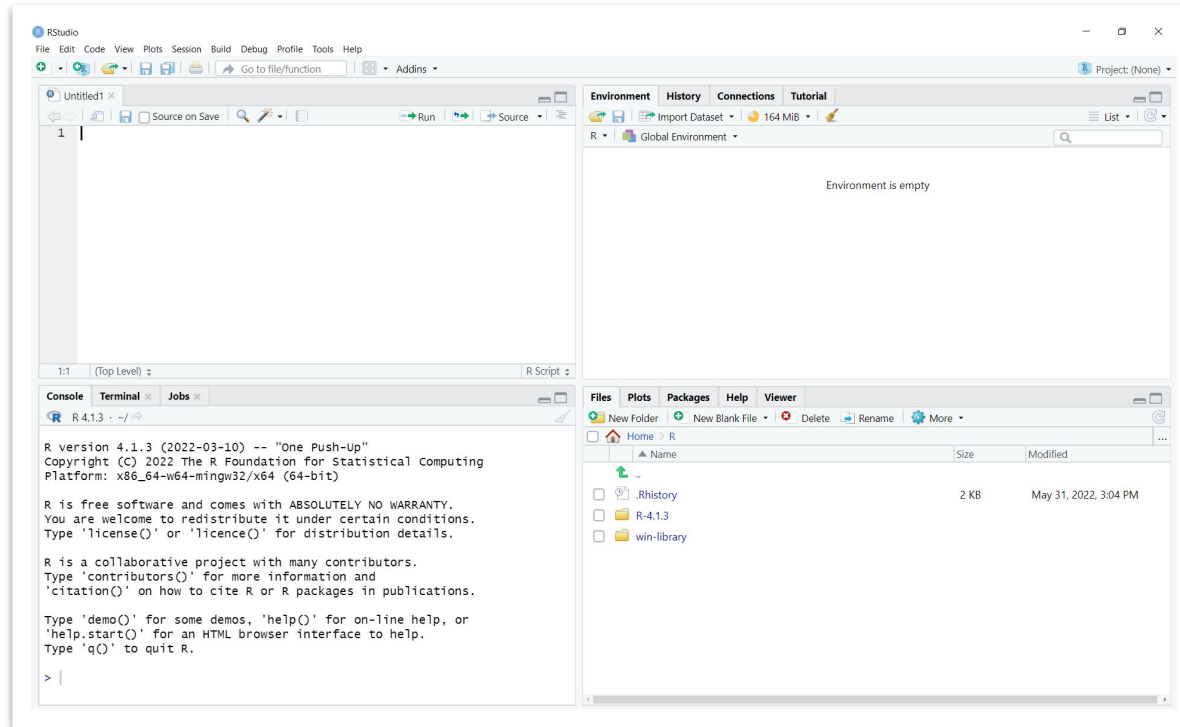
Die RStudio-Oberfläche

Skript

Skript-Dokument
öffnen, schreiben,
Befehle ausführen

Konsole

Ausgabefenster für
ausgeführte Befehle



Environment
Objekte im
Arbeitsspeicher

Files
Dateien im
ausgewählten
Verzeichnis



Help
Anzeigefenster
Hilfe-Datei

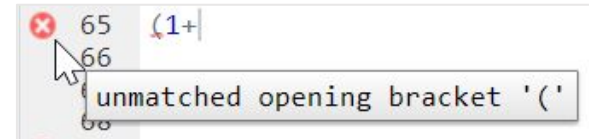
Konsole

Hier können Befehle eingegeben und ihr Ergebnis angezeigt werden.

- Eingabeaufforderung mit Zeichen > am Anfang der Zeile
- Ausführen eines Befehls mit `ENTER`-Taste
- + am Zeilenanfang: Aufforderung, noch etwas einzugeben, da die Befehlseingabe noch nicht abgeschlossen ist.


R-Skripte

- Textdokument mit der Dateiendung .R
- enthält ausführbare R-Befehle
- Ausführen einer Zeile durch `STRG + Enter` oder 
- Ausführen des gesamten Skripts durch 
 - besonders geeignet für reproduzierbare Analysen!
- Unterstützung durch farbliche Hervorhebung der Elemente und möglicher Fehler



R-Skripte effektiv nutzen

Wichtig! Ein übersichtlich geschriebenes und kommentiertes Skript hilft euch und anderen dabei, die Datenanalyse zu einem späteren Zeitpunkt nachvollziehen zu können.

- Kommentare benutzen!
- Das Skript lässt sich in Abschnitte gliedern, die in einem Inhaltsverzeichnis angezeigt werden (abrufbar u.A. unter  im Skriptfenster)

```
# Basis-Rechenoperationen:  
3 + 4 # Addition
```

```
# Ebene 1 ----  
## Ebene 2 ----  
### Ebene 3 ----
```

RStudio Web und Desktop: Unterschiede

Aspekt	RStudio Web (PsychNotebook)	RStudio Desktop (Installation)
Arbeitsverzeichnis	virtuell	lokal setzen mit Befehl <code>setwd("Dateipfad")</code>
Pakete installieren	einmalig installieren über den Dialog <i>Projekt</i> → <i>Pakete installieren</i> (im Workshop-Projekt bereits voreingestellt)	einmalig installieren mit Befehl <code>install.packages("paketname")</code>
Sicherung der Datei und Session	Sicherung automatisch oder manuell	Sicherung manuell (Skript speichern mit STRG + S)

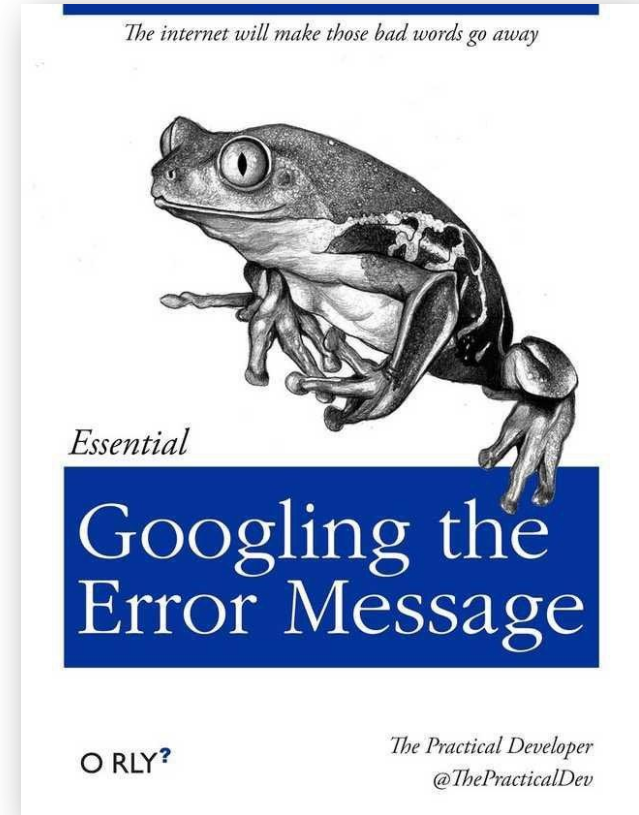
Help?

Die **R-Hilfe** benutzen:

- `help(funktion)` oder `?funktion`
öffnet die Hilfedatei zu einer Funktion
- `??funktion` sucht in der Dokumentation für
alle (nicht nur geladene) Pakete

Keine Angst vor Fehlermeldungen! Sie sind
Teil des Arbeitsprozesses.

Tipp: <https://stackoverflow.com/questions>



B

Objekte und Aktionen in R

R als Taschenrechner: Rechenoperationen und logische Operationen

Zeichen	Bedeutung
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
^	Potenz
<code>sqrt(x)</code>	Wurzel

Zeichen	Bedeutung
==	gleich
!=	ungleich
>, <	kleiner, größer
>=, <=	kleiner/gleich, größer/gleich
&	logisches UND
	logisches ODER

Funktionen bzw. Befehle

Funktionen sind in R stets nach dem gleichen Muster aufgebaut:

```
funktion(argument1 = wert1, argument2 = wert2, ...)
```

Solange die Reihenfolge der Argumente gleich bleibt, muss man sie nicht gesondert nennen.

Beispiel:

- `round(x = 2.567, digits = 2)` ist das gleiche wie
- `round(2.567, 2)`

Objekte selbst erzeugen

Mit dem Zuweisungspfeil <- kann ein neues Objekt erzeugt werden

```
> zahlen <- c(1, 2, 3, 4)
> zahlen
[1] 1 2 3 4
```

Objekte benennen

- Objektnamen können Buchstabe, Zahlen, "." und "_" enthalten
- Vorsicht: R berücksichtigt **Groß- und Kleinschreibung!**

Tipp: Objektnamen wählen, die so kurz wie möglich, aber so lang wie nötig sind! Am besten sind nachvollziehbare Namen, die möglichst wenig Tippen erfordern!

✗ `datensatz_2_doktorarbeit` → Nicht optimal...

✓ `dat2` → Besser!

Daten- und Objektklassen

- **numeric**: Zahlen, z.B. 1.5, 2.14, 7,...
 - **integer** (Unterkategorie von *numeric*): Ganze Zahlen, z.B., 1, 3, 5...
 - **character**: Zeichen (Buchstaben) und Zeichenketten (Wörter)
 - **logical**: TRUE, FALSE
-
- prüfbar mit dem Befehl `class (objekt)`

Vektoren

Vektoren sind eindimensionale Objekte, die eine Reihe von Elementen enthalten.

- `class()` ist auch auf Vektoren anwendbar. Alle Elemente in einem Vektor haben die gleiche Klasse.
- `str()` zeigt die Struktur und die Eigenschaften eines Vektors

```
> str(zahlen)
num [1:4] 1 2 3 4
```

- Mit numerischen Vektoren kann man ebenso wie mit einzelnen Elementen rechnen:

```
> zahlen - 1.5
[1] -0.5  0.5  1.5  2.5
> zahlen + zahlen
[1] 2 4 6 8
```

Data Frames

- Ein Dataframe ist ein zweidimensionales rechteckiges Datenobjekt, das Zeilen (Beobachtungen) und Spalten (Variablen) enthält.
- Die Spaltenvariablen können verschiedene Dateiklassen (numerisch, nicht-numerisch) haben.
- Wir können ein Dataframe erzeugen, indem wir mehrere Vektoren kombinieren:

```
> dat <- data.frame(alter, geschl)
> dat
  alter geschl
1    21   mann
2    29   frau
3    35   mann
4    25   frau
5    52 divers
6    20   mann
7    19   frau
```

Übungen zu Teil A+B

Aufgabe 1.

Rufe die R-Hilfeseite für die Funktion `c()` auf. In der Sektion "Examples" befinden sich mehrere Anwendungsbeispiele der Funktion. Führe das zweite Beispiel aus. Wieviele Elemente befinden sich in dem Objekt? Welche Klasse hat das Objekt?

Aufgabe 2.

- a) Führe diesen Code aus. Wieso funktioniert er nicht? Welche Korrektur müsste man machen, damit er ausgeführt werden kann?

```
altersunterschied <- 20 - 10
atlersunterschied
```

- b) Modifiziere diese Funktion so, dass die Funktion ausgeführt werden kann.
Tipp: Zwei Korrekturen sind erforderlich.

```
dat_neu <- data.frame(code = c(1, 2, 3, 4,
                                name = c("julia" "denis", "miriam", "kim"))
```

Übungen zu Teil A+B

Aufgabe 3.

Julia und Denis haben fünf Statistik-Aufgaben gelöst. Ihre Ergebnisse (korrekt = `TRUE`, falsch = `FALSE`) wurden in diesen beiden Vektoren eingetragen.

```
results_julia <- c(5, 4, 2, 1, 0)
results_denis <- c(5, 3, 2, 1, 2)
```

In welchen Aufgaben haben Julia und Denis die gleichen Ergebnisse erzielt? In welchen Aufgaben hatte Julia eine höhere Punktzahl als Denis? Und wo war Denis besser?

Aufgabe 4.

In der `mean`-Funktion kann man das Argument `na.rm` ergänzen. Finde heraus, was dieses Argument bedeutet, indem du die Hilfedatei für die `mean`-Funktion aufrufst. Erinnerung: `NA` ("not available") kennzeichnet einen fehlenden Wert in der Datenreihe.

C

Arbeiten mit tabellarischen Daten

Daten importieren

Um die Datenbearbeitung und Analyse zu starten, müssen wir in der Regel einen Datensatz in R importieren.

Es gibt Möglichkeiten für verschiedene Dateitypen:

- `load("filename.rda")` für R-eigenes Dateiformat `.rda` oder `.rdata`
- `read.table("filename.txt")` für Dateien mit Endung `.txt` oder `.csv`
- `read.csv("filename")` für Dateien mit der Endung `.csv`
- `read.csv2("filename")` für Dateien mit der Endung `.csv`, die ";" als Trennzeichen und "," als Dezimalzeichen verwenden. (Beispiel: 1,5 statt 1.5). **Vorsicht:** Das ist bei in Deutschland verwendeten Tabellendaten häufig der Fall!

readxl-Paket für den Import von Excel-Dateien

- `read_excel("filename")` für das Einlesen von Excel (.xlsx)-Dateien
- da diese Funktion in Basis-R nicht enthalten ist, müssten wir zunächst das Paket **readxl** mit dem Befehl `library(paketname)` laden, um es nutzen zu können.

```
> # Im lokalen R muss vorher einmalig das richtige Paket installiert werden:  
> # install.packages("readxl")  
> library(readxl)
```

Wenn wir den Befehl ausführen, wird uns in der Konsole nichts weiter angezeigt. Im Tab "Packages" sehen wir aber, dass das Paket nun "aktiviert" ist.

Files	Plots	Packages	Help	Viewer
		Name	Description	
<input type="checkbox"/>		RcppEigen	'Rcpp' Integration for the 'Eigen' Templated Linear Algebra Library	
<input checked="" type="checkbox"/>		readxl	Read Excel Files	

foreign-Paket für den Import von SPSS-Dateien

- Auch Dateien aus SPSS (z.B. mit der Dateiendung .sav) können in R mithilfe des Pakets **foreign** verarbeitet werden.
- Nach dem Laden des Pakets mit `library(foreign)` können wir die Funktion `read.spss("filename")` nutzen.
- Beispiele:

```
dat  <- read.spss("studie1.sav")  
dat2 <- read.spss("studie2.sav", to.data.frame = TRUE)
```

Tipp: Auch Stata-Dateien und Dateien vieler weiterer "externer" Statistik-Programme lassen sich mit dem `foreign`-Paket importieren.

Beispieldatensatz: Erstsemester (*erstis*)

Daten von 109 Psychologie-Studierenden im 1. Studienjahr, die im Rahmen einer Online-Befragen erhoben wurden:

Variable	Inhalt	Kodierung
<i>alter</i>	Alter	<i>Jahre</i>
<i>geschl</i>	Geschlecht	1 = weiblich, 2 = männlich, 3 = divers
<i>ort</i>	Derzeitiger Wohnort	1 = Studienort, 2 = anderer
<i>job</i>	Nebentätigkeit	1 = nein, 2 = ja
<i>neuro</i>	Neurotizismus	<i>Skalenwert (1–5)</i>
<i>lz</i>	Lebenszufriedenheit	<i>Skalenwert (1–7)</i>
<i>prok1</i> – <i>prok3</i>	Items zur Prokrastinationstendenz	<i>Skalenwert</i> (1 = stimmt nicht, 4= stimmt genau)

Quelle: Frei adaptiert nach <https://pandar.netlify.app/post/datensaetze>.

Die fiktive Variable "Alter" wurde dem Datensatz nachträglich zu Lehrzwecken hinzugefügt und war nicht Teil des Originaldatensatzes.



Daten ansehen: Eigenschaften eines Data Frame

Nützliche Funktionen:

Funktion	Bedeutung
<code>View(dat)</code>	Dataframe in neuem Fenster als Tabelle anzeigen (praktisch zum Betrachten eines großen Datensatzes)
<code>names(dat)</code> <code>colnames(dat)</code>	Namen der Spalten/Variablen ausgeben
<code>dim(dat)</code>	Dimensionen (Anzahl Zeilen, Spalten) eines Dataframe
<code>nrow(dat)</code>	Anzahl der Reihen
<code>ncol(dat)</code>	Anzahl der Spalten
<code>str(dat)</code>	Überblick über die Struktur des Dataframe
<code>summary(dat)</code>	Deskriptive Zusammenfassung der Variablen im Dataframe
<code>head(dat)</code>	Die ersten sechs Zeilen eines Dataframe ausgeben

Bild© Eren Li / pexels.com

Daten auswählen

Methoden, um aus einem Dataframe interessierende Teildaten und Variablen zu extrahieren:

- Der **Dollarzeichen-Operator**:
Auswahl einer einzelnen Variable

```
> erstis$alter
```

- Auswählen von Zeilen und Spalten
mit **eckigen Klammern**

```
> erstis[1,]  
> erstis[,1]  
> erstis[, "alter"]
```

- Auswählen von Variablen
mit der **Funktion subset()**

```
> subset(erstis, alter < 18)
```

Auswählen mit eckigen Klammern

Das Indizieren eines Dataframe mit eckigen Klammern folgt dem

Muster: `dataframe[Zeilennummer, Spaltennummer]`

- `erstis[1,]` wählt 1. Zeile/Beobachtung, alle Spalten aus
- `erstis[, 2]` wählt alle Zeilen und 2. Variable/Spalte aus

Mit dieser Methode lassen sich auch mehrere Zeilen- oder Spaltenvariablen gleichzeitig auswählen.

- `erstis[10:15,]` wählt Beobachtungen 10 bis 15 aus
- `erstis[, c("alter", "ort")]` wählt Variablen Alter und Wohnort aus

Logisches Auswählen

Auch hier sind die logischen Operatoren anwendbar, die wir zuvor kennengelernt haben, um uns **Teildatensätze** anzuschauen:

```
> erstis[erstis$geschl == "mann", ]
```

	alter	geschl	ort	job	neuro	lz	pork1	prok2	prok3
1	20	mann	anderer	ja	3.00	3.0	3	2	2
14	23	mann	Studienort	nein	2.75	5.0	2	2	3
18	21	mann	anderer	nein	3.00	6.8	2	2	1
19	23	mann	anderer	ja	3.25	5.8	3	3	4

wähle alle Zeilen aus, für die gilt:
"Geschlecht = Mann", und zeige alle
Spalten dieser Bedingung.

```
> erstis[erstis$alter > 30 | erstis$alter < 18, ]
```

	alter	geschl	ort	job	neuro	lz	pork1	prok2	prok3
7	54	frau	Studienort	nein	2.25	5.8	1	2	2
20	17	frau	Studienort	ja	4.00	4.4	2	2	3
90	40	frau	Studienort	ja	3.00	6.6	2	4	3
96	17	frau	anderer	ja	3.00	5.4	2	3	2

wähle alle Zeilen aus, für die gilt: Alter ist
größer als 30 ODER kleiner als 18, und zeige
alle Spalten dieser Bedingung.

Auswählen mit der subset()-Funktion

Eine weitere bequeme Funktion, die für das Auswählen von Teildatensätzen geeignet ist, ist die `subset()`-Funktion.

Die Auswahl funktioniert nach dem Muster: `subset(x = Objekt, subset = Spaltenbedingung)` und erzielt die ähnliche Resultate wie das logische Auswählen mit `[,]`.*

```
> subset(erstis, geschl == "mann")
```

	alter	geschl	ort	job	neuro	lz	pork1	prok2	prok3
1	20	mann	anderer	ja	3.00	3.0	3	2	2
14	23	mann	Studienort	nein	2.75	5.0	2	2	3
18	21	mann	anderer	nein	3.00	6.8	2	2	1
19	23	mann	anderer	ja	3.25	5.8	3	3	4

```
> subset(erstis, alter > 30 | alter < 18)
```

	alter	geschl	ort	job	neuro	lz	pork1	prok2	prok3
7	54	frau	Studienort	nein	2.25	5.8	1	2	2
20	17	frau	Studienort	ja	4.00	4.4	2	2	3
90	40	frau	Studienort	ja	3.00	6.6	2	4	3
96	17	frau	anderer	ja	3.00	5.4	2	3	2

* In diesem Fall erzielen sowohl eckige Klammern als auch `subset()` das gleiche Ergebnis. Jedoch hat `subset()` besonders hilfreiche Eigenschaften in anderen Bereichen, z.B. dem Umgang mit fehlenden Werten. Daher ist es nützlich, diese Funktion jetzt schon zu kennen.

Daten bearbeiten:

Variable umbenennen

- Mit der Funktion `names(dat)` können wir uns einen Vektor aller Spaltennamen ausgeben lassen.
- Mit dem Zuweisungspfeil einzelne Elemente in dem Spaltennamen-Vektor durch neue ersetzen:

```
> names(erstis)[7] <- "prok1"  
> names(erstis)  
[1] "alter" "geschl" "ort"    "job"    "neuro"  "lz"    "prok1"  "prok2"  
[9] "prok3"
```

Beispiel: Ersetze das 7. Element im Spaltennamen-Vektor durch die Zeichenkette "prok1".

Vorsicht! Dafür immer prüfen, ob die Nummer des Elements, das wir auswählen, auch wirklich die richtige ist.

Daten bearbeiten:

Variablen hinzufügen / transformieren

Wollen wir – z.B. durch Transformation – dem Dataframe eine neue Variable hinzufügen, ist auch dafür der Zuweisungspfeil nutzbar.

- Beispiel: Dieser Befehl erzeugt eine neue Spaltenvariable `erstis$gebjahr`:

```
> erstis$gebjahr <- 2021 - erstis$alter
> erstis$gebjahr
[1] 2001 2001 2002 2000 1997 1998 1967 1998 2000
[10] 2001 1999 2002 2000 1998 2002 1997 1996 2000
```

- Ausführbar auch mit mehreren Spaltenvariablen (vgl. Vektorrechnung), z.B:

```
> # Summenscore Prokrastination
> erstis$prok_sum <- erstis$prok1 + erstis$prok2 + erstis$prok3
```

Daten bearbeiten:

Beobachtungen und Variablen entfernen

Eine oder mehrere **Zeilen** aus dem Dataframe entfernen mit dem Minus-Zeichen:

- `erstis2 <- erstis[-5,]` erzeuge neuen Dataframe ohne die 5. Zeile
- `erstis2 <- erstis[-(5:10),]` erzeuge neuen Dataframe ohne die Zeilen 5-10

Eine oder mehrere **Spalten** aus dem Dataframe entfernen:

- `erstis 3 <- erstis2[, -(7:9)]`

Tipp: Beim Entfernen von Elementen immer eine **neue Version** des Objekt erzeugen, statt das Objekt zu überschreiben, das wir verändern. Das spart Arbeit und Nerven, falls beim Entfernen doch ein Fehler passiert sein sollte.

Daten bearbeiten:

Beobachtungen und Variablen entfernen (2)

Mit dem Minus-Operator lassen sich leider keine Spalten anhand ihres Namens entfernen – nur anhand ihrer Nummer. Um Verwechslungen vorzubeugen, kann die `subset()`-Funktion auch hier zum Auswählen (`select`) genutzt werden:

- `erstis3 <- subset(erstis2, select = -(prok1:prok3))`

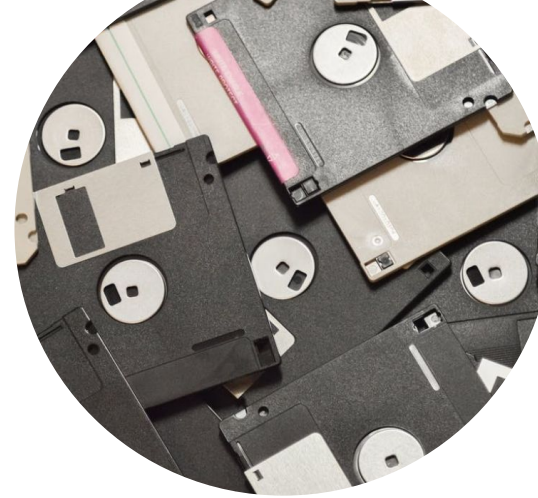
Daten speichern

Dataframe im R-eigenen Datenformat `.rda` oder `.rdata` speichern, das alle Variableneigenschaften originalgetreu erhält mit:

- `save(objekt, file = "filename.rda")`

Speichern in anderen Dateiformaten nach gleichem Muster:

- `write.table()` für Dateierendungen `.txt`, `.csv`
- `write.csv()` besonders geeignet für `.csv`
- `write.csv2()` für `.csv` mit Trennzeichen ";" und Dezimalzeichen ","
- `write_excel()` für `.xls` und `.xlsx` (benötigt das Paket *writexl*)



Bild© SJ / pexels.com

Beispieldatensatz: Zweitsemester (zweitis)

Daten von $N = 150$
Psychologie-Studierenden
im 1. Studienjahr, die im
Rahmen einer Online-
Befragung erhoben
wurden

Quelle: Adaptiert nach
https://www.beltz.de/fachmedien/psychologie/online_material/r_fuer_einsteiger_5_auflage.html

Variable	Inhalt	Kodierung
code	Versuchspersonennummer	<i>Nummer</i>
gruppe	Seminarkurs-Gruppe	<i>Kurs 1 – Kurs 4</i>
geschl	Geschlecht	1 = weiblich, 2 = männlich
alter	Alter	<i>Jahre</i>
kinder	Elternschaft	1 = ja, 2 = nein
job	Nebentätigkeit	1 = nein, 2 = ja
prok	Prokrastinationstendenz	<i>Skalenmittelwert (1–4)</i>
extra	Extraversion	<i>Skalenwert (1–5)</i>
vertraeg	Verträglichkeit	<i>Skalenwert (1–5)</i>
gewiss	Gewissenhaftigkeit	<i>Skalenwert (1–5)</i>
neuro	Neurotizismus	<i>Skalenwert (1–5)</i>
offen	Offenheit für Neues	<i>Skalenwert (1–5)</i>

Übungen zu Teil C

Aufgabe 1.

Lies den Datensatz `zweitis.csv` mit dem Befehl `read.table("zweitis.csv")` ein. Sieh ihn dir danach an. Wo könnte etwas schiefgelaufen sein? Durch welchen Befehl lässt sich das Problem lösen?

Aufgabe 2.

Lies für die nächsten Aufgaben den Datensatz `zweitis.rda` ein, indem du `load("zweitis.rda")` ausführst. `zweitis` befindet sich nun in deinem Environment.

Die kategoriale Variable `kinder` (1 = ja, 2 = nein) hat im Datensatz noch fälschlicherweise die Klasse numerisch. Wandle die Variable in einen **Faktor** um. Achte auf eine korrekte Benennung der Faktorstufen.

Aufgabe 3.

Was ist die 5. Beobachtung im Datensatz `zweitis`? Welche Eigenschaften weist diese Person auf den demografischen Variablen (Geschlecht, Alter, Kinder, Job) auf?

Zusatz: Wieviele Personen in dem Datensatz sind 18 oder jünger? Wieviele männliche Studierende gibt es in Kurs 3?

D

Deskriptive Statistik

Lage- und Streuungsmaße

`summary()` bietet einen Gesamtüberblick über deskriptive Maße des gesamten Datensatzes.

Wollen wir spezifische Werte einzeln ausgeben lassen, geht das auch durch die folgenden Funktionen:

- `mean()` Mittelwert
- `median()` Median
- `var()` Varianz
- `sd()` Standardabweichung
- `range()` Minimum und Maximum

Statistiken für Subgruppen mit *describeBy()*

Mit den Funktionen `describe()` und `describeBy()` aus dem Paket "psych" gibt es eine übersichtliche Möglichkeit, deskriptive Statistiken für numerische Variablen zu erzeugen.

- `describe(dat)`
- `describe(dat, omit = TRUE)`: Das Argument `omit` entfernt eventuelle kategoriale Variablen aus der Ansicht.
- `describeBy(dat, group = "variable")` zeigt Statistiken für die Subgruppen, die den Kategorien der ausgewählten Variable entsprechen.

Deskriptive Tabellen für kategoriale Variablen

`table()` erzeugt eine Tabelle mit absoluten Häufigkeiten für eine oder mehrere Variablen.

```
> table(erstis$geschl)
```

frau	mann	divers
78	30	1

*Tabelle mit absoluten Häufigkeiten
der Kategorien von "Geschlecht"*

```
> table(erstis$job, erstis$geschl)
```

	frau	mann	divers
nein	41	19	0
ja	37	11	1

*Kreuztabelle mit absoluten Häufigkeiten der
Variablenkategorien "Job" (Zeilen) je nach
Geschlecht (Spalten).*

Deskriptive Tabellen für kategoriale Variablen

`prop.table()` erzeugt relative Häufigkeitstabellen aus einem anderen Tabellenobjekt:

```
> tab1 <- table(erstis$geschl)
> round(prop.table(tab1), digits = 2)
```

frau	mann	divers
0.72	0.28	0.01

Tabelle mit **relativen** Häufigkeiten
der Kategorien von "Geschlecht"

```
> tab2 <- table(erstis$job, erstis$geschl)
> round(prop.table(tab2), digits = 2)
```

	frau	mann	divers
nein	0.38	0.17	0.00
ja	0.34	0.10	0.01

Kreuztabelle mit **relativen** Häufigkeiten der
Variablenkategorien "Job" (Zeilen) je nach
Geschlecht (Spalten).

Deskriptive Tabellen für numerische Variablen

Durch das Argument `margin` (Randsummen) können wir der Funktion `prop.table()` übergeben, ob die relativen Häufigkeiten über die Zeilen (`margin = 1`) oder die Spalten (`margin = 2`) auf 1 summiert werden sollen.

```
> prop_tab1 <- prop.table(tab2, margin = 1)
> prop_tab1 <- round(prop_tab1, 2)
> addmargins(prop_tab1)
```

	frau	mann	divers	Sum
nein	0.68	0.32	0.00	1.00
ja	0.76	0.22	0.02	1.00
Sum	1.44	0.54	0.02	2.00

Mit `margin = 1` ergeben die
Zeilensummen 1.

```
> prop_tab2 <- prop.table(tab2, margin = 2)
> prop_tab2 <- round(prop_tab2, 2)
> addmargins(prop_tab2)
```

	frau	mann	divers	Sum
nein	0.53	0.63	0.00	1.16
ja	0.47	0.37	1.00	1.84
Sum	1.00	1.00	1.00	3.00

Mit `margin = 2` ergeben die
Spaltensummen 1.

Korrelation

Einer der vielen statistischen Zusammenhangsmaße, die sich in R berechnen lassen, ist der Korrelationskoeffizient:

- `cor(dat$variable1, dat$variable2)`: Pearson'scher Korrelationskoeffizient für die ausgewählten Variablen
- `cor.test(dat$variable1, dat$variable2)`: Inferenzstatistischer Test des Korrelationskoeffizienten

Übungen zu Teil D) Deskriptivstatistik

Aufgabe 1.

Erzeuge deskriptive Statistiken für die numerischen Variablen aus dem Datensatz `zweitis`, gruppiert nach Kurszugehörigkeit (Kurs 1 – Kurs 4).

Aufgabe 2.

Erstelle eine Kreuztabelle für die Variable `gruppe` als Zeilen- und `job` als Spaltenvariable. In welchem Kurs sind die meisten Personen mit Kindern?

Aufgabe 3.

Entscheide dich für zwei numerische Variablen aus dem Datensatz, die dich besonders interessieren, und berechne den Zusammenhang zwischen diesen.



Wie weitermachen?

- R-Software hier herunterladen:
<https://cran.r-project.org>
- RStudio Download:
www.rstudio.com/products/rstudio/download

Bild © Patrick Perkins / Unsplash

Wo weiterlernen?

- Noch mehr Einführungen zu R, inkl. **Grafiken**: "Quick-R" www.statmethods.net
- Lektionen zu statistischen **Datenanalyse-Verfahren**: <https://pandar.netlify.app/lehre>

Veranstaltungen in "*Practices and Tools of Open Science*" mit R-Bezug:

- 15.06.2022: **Poweranalyse**
- 02.11.2022: **Reproduzierbare Textanalysen** mit Topic Modeling
- 16.11.2022: **Friends don't let friends copy and paste.** Analytisch reproduzierbare, APA-konforme Manuskripte mit dem R-Paket *papaja*

<https://leibniz-psychology.org/ptos>

Abschlussrunde

Vielen Dank für Eure Aufmerksamkeit!

... und viel Spaß mit R!