

Wissenschaftliches Arbeiten und Forschungsmethoden

Einheit 8: Auswertung von Studien - Teil 1: Datenvorbereitung,
Stichprobenbeschreibung und Deskriptiv-Statistik

19.06.2024 | Dr. Caroline Zygar-Hoffmann

Termine

Einheit	Datum	Hintergrundwissen & Theorieinput Thema	Praxisteil Thema
1	10.04.2024	Präregistrierung im Forschungsprozess & Vorstellung Themenwahl	Gruppenfindung, Open Lab Book in GoogleDoc, Themenauswahl
2	17.04.2024	Forschungsfrage, Literaturrecherche & Studienorganisation	Literaturrecherche, Entwicklung Forschungsfrage
3	24.04.2024	Theoriearbeit & Studiendesign	Ordnerstruktur, Hypothesen, Studiendesign, erste Analysegedanken
-	01.05.2024	entfällt wegen Feiertag	-
4	08.05.2024	Operationalisierung & formr	Operationalisierung & Implementierung & Testdurchlauf
5	15.05.2024	Analyseplan (am Beispiel Interventionsthema)	Analyseplan
6	22.05.2024	Samplingplan & Durchführung von Studien	Poweranalyse & OSF-Projekt & Präregistrierung finalisieren
-	29.05.2024	entfällt wegen Zeitraum Wiederholungsprüfung	Daten sammeln
7	05.06.2024	Publikation von Studien mit Fokus auf Einleitung und Methode	Daten sammeln
8	12.06.2024	Publikation von Studien mit Fokus auf Einleitung und Methode	Daten sammeln
9	19.06.2024	Auswertung von Studien - Teil 1: Datenvorbereitung, Stichprobenbeschreibung und Deskriptiv-Statistik	Code-Review
10	26.06.2024	Auswertung von Studien - Teil 2: Ergebnisse und Datenvisualisierung	Rückmeldung für Teilnehmer
11	03.07.2024	Publikation von Studien mit Fokus auf Ergebnisteil und Diskussion	Feedback zwischen Gruppen
12	10.07.2024	Publikation von Daten & das Wissenschaftssystem	Review innerhalb von Gruppen
13	17.07.2024	Freie Spalte und Fragen	-

Laufende Studien

.small[Hier die Links zu den aktuell schon angelaufenen Datenerhebungen Ihrer Kommiliton:innen. Ich ermuntere alle zur Teilnahme!

<https://tellmi.psy.lmu.de/formr/Grashuepfer>

<https://tellmi.psy.lmu.de/formr/falscherOptimismus>

<https://tellmi.psy.lmu.de/formr/ErkenneDeinenWert>

<https://tellmi.psy.lmu.de/formr/GeschlechtBindungsstile>

<https://tellmi.psy.lmu.de/formr/FuenfFragezeichenRunFinal>

<https://tellmi.psy.lmu.de/formr/BindungsstilSelbstwert>

<https://tellmi.psy.lmu.de/formr/Haustiere-Finalrun>

<https://tellmi.psy.lmu.de/formr/DieMenstruierendenForscher>

<https://tellmi.psy.lmu.de/formr/FriendsAndRelationships>

<https://tellmi.psy.lmu.de/formr/IntThema>

<https://tellmi.psy.lmu.de/formr/Extraversion>

<https://tellmi.psy.lmu.de/formr/StimmungsScouts>

Heutige Themen

Auswertung von Studien: Datenvorbereitung

- Ordnerstruktur
- Datensätze kombinieren
- Bedingungsvariable berechnen
- Daten bereinigen und Variablen berechnen

Auswertung von Studien: Stichprobenbeschreibung und Deskriptivstatistik

Praxisaufgabe

Auswertung von Studien

Ordnerstruktur

Ziel einer guten Ordnerstruktur: Sie können den Ordner mit allen Unterordnern kopieren, jemandem zukommen lassen, und diese Person versteht was gemacht wurde und wie die Analysen wiederholt werden können

Englische Ordnernamen bevorzugt, um breitere Verständlichkeit zu gewährleisten

Readme-Textdatei auf dem obersten Level empfohlen, um alles Nicht-Selbsterklärendes zu erklären

Name
analyses
documentation
literature
manuscript
preprocessing
processed_data
raw_data
results_plots
README.txt

Vorschlag für eine gute Ordnerstruktur

Auswertung von Studien

Ordnerstruktur

Ordner "documentation"

- Hier können Sie Kontextinformationen zu Ihrer Studie ablegen, z.B. die formr-Exceldateien, die Präregistrierung, verwendete Fragebögen, das Open Notebook, Codebooks zu offenen Daten

Für beide Ordner ggf. weitere thematische Unterordner sinnvoll, je nach Menge der Dateien

Ordner "literature"

- Hier können Sie die gesammelte Literatur zu Ihrem Thema abspeichern

Ordner "manuscript"

- Hier können Sie die RMarkdown-Datei ablegen, in der Sie den Bericht schreiben (im Ordner Seminar auf studynet -> Vorlagen Prüfungsleistung -> RMarkdown Code -> "template.Rmd")

Auswertung von Studien

Ordnerstruktur

Ordner "raw_data"

- Hier sollen die originalen **Rohdaten** abgespeichert werden, so wie sie aus der Fragebogensoftware (bei uns: formr) rauskommen und ich sie Ihnen zuschicke
- Pro "Survey" in formr gibt es einen Datensatz/eine Datendatei
- **Sobald die Rohdaten hier einmal drinliegen, sollen sie nicht mehr verändert werden.**
- Es ist wichtig, die Rohdaten in ihrer Originalform vor jeglicher Bearbeitung zu behalten (z.B. um nicht versehentlichweise Daten zu überschreiben); zugespitzt gesagt: am besten Sie fassen (klicken) den Ordner danach nicht mehr an.
- Rohdaten sollten auch nicht mit Excel geöffnet werden, sondern direkt in R eingelesen werden -> Excel macht manchmal automatisch etwas, das die Daten zerstört (z.B. Zahlen in ein Datum umwandeln)
- Achtung: Rohdaten sollten nicht einfach irgendwo im Internet hochgeladen werden, sondern erst wenn Sie gemäß der Datenschutzrichtlinien bearbeitet worden sind (z.B. bei uns: Pseudonym gelöscht)

Auswertung von Studien

Ordnerstruktur

Ordner "preprocessing" und "processed_data"

- Im "preprocessing" Ordner liegen die R-Skripte drin, mit denen die Daten für die Analyse vorbereitet werden (siehe restliche Einheit)
- Verschiedene Vorverarbeitungsschritte können (müssen aber nicht) in verschiedene R-Skripte aufgeteilt werden, sollten dann aber nummeriert werden in der Reihenfolge in der sie ausgeführt werden müssen (z.B. "1-combine-data.R", "2-code-missings", "3-recode-variables")
- Die resultierende(n) vorverarbeitete(n) Datendatei(en) können mit dem Befehl `save()` im Ordner "processed_data" abgespeichert werden, z.B.

```
save(processed_data, file = "../processed_data/final_data.RData")
```

Auswertung von Studien

Ordnerstruktur

Ordner "analyses"

- Hier sollen die R-Skripte abgespeichert werden, in denen Sie jegliche Analysen machen (z.B. Deskriptive Statistik, Hypothesen, Explorative Analysen)
- Am Anfang der Analyseskripte in R (und auch im Rmarkdown selbst) können Sie die vorverarbeitete(n) Datendatei(en) mit dem Befehl `load()` einlesen, z.B.:

```
load(file = "../processed_data/final_data.RData")
```

- Idealerweise für jedes Analyse "thema" ein eigenes R-Skript mit sprechendem Namen (z.B. "Descriptives.R", "Hypothesis_1.R", "Exploration_GenderDifferences.R")
- Die Analyseskripte selbst können Sie später auch mit dem Befehl `source()` in Ihrem RMarkdown Skript einlesen, d.h. die Ergebnisse aus den Skripten liegen dann für die weitere Bearbeitung im RMarkdown vor, z.B.:

```
source(file = "../analyses/Descriptives.R")
```

Auswertung von Studien

Ordnerstruktur

Advanced: Projekte in R Studio

- Sie können in R Studio einen Ordner mit all seinen Unterordnern einem Projekt zuordnen (z.B. den Ordner in dem Sie die oben beschriebene Ordnerstruktur angelegt haben), indem sie unter "File -> New Project -> Existing Directory" den gewünschten Ordner auswählen
- Vorteil: Sie müssen das "working directory" nicht immer neu setzen, da es fix der Projektordner ist
 - In den bisherigen Befehlen wo bei `file = "..."` zwei Punkte und ein Slash `../` vorangestellt waren (um auf die obere Ordnerstufe zu wechseln) wären dann überflüssig, weil das working directory des Projekts der gewählte Projektordner ist
 - Im RMarkdown sollten Sie dann auch bei dem Pfeil neben "Knit" (die Wolle) unter "Knit Directory" -> "Project Directory" angeben, damit alle Pfade passen
- Weiterer Vorteil: Es werden beim Öffnen des Projekts alle zuletzt offenen R-Dateien aus dem Projekt geöffnet
- Wenn Sie für die Datenvorbereitung/-analyse zusammenarbeiten, dann benutzen idealerweise entweder alle aus der Kleingruppe die Projekt-Funktion oder gar keiner, weil sonst bei manchen die Pfade funktionieren und bei manchen nicht.

→ Probieren Sie es mal! Es tut nicht weh und hat Vorteile. Das Projekt im Nachhinein nicht zu benutzen geht immer (aber das ist eine Entscheidung, die Sie aktiv treffen müssen, da Sie dann die Pfade dann wieder entsprechend anpassen müssen).

Auswertung von Studien

Datensätze kombinieren: Schritt 1 - Daten einlesen

- Für den Fall, dass Ihre Studie nicht nur aus einem Survey in formr besteht, ist der erste Vorverarbeitungsschritt, dass Sie Ihre Datensätze in einen einzelnen Datensatz kombinieren
- Dafür müssen Sie zunächst alle Datensätze einlesen:

```
survey1 <- read.csv("../raw_data/survey1.csv")
survey2 <- read.csv("../raw_data/survey2.csv")
```

usw.

Auswertung von Studien

Datensätze kombinieren: Schritt 2 - Leere sessions löschen

- In jedem formr Survey gibt es die Spalte "session" die eindeutig eine Zeile im Datensatz einer Durchführung des Fragebogen-Runs zuordnet
- Da jede Zeile eindeutig eine Person sein soll, können wir mit dieser Spalte also die Datensätze richtig zusammenfügen
- Zunächst müssen wir dafür aber alle Zeilen entfernen, in denen in "session" nichts drin steht (das ist i.d.R. der Fall, wenn der Fragebogensurvey nur geladen wurde, aber sonst nichts auf der Seite gemacht wurde):

```
survey1 <- survey1[survey1$session != "",]  
survey2 <- survey2[survey2$session != "",]
```

usw.

(R-Zeile 1 in Worten: Überschreibe das Objekt "survey1" mit den Zeilen vom Objekt "survey1" in denen in der Spalte "session" *nicht* "" (= nichts) drinsteht, und behalte alle Spalten)

Auswertung von Studien

Datensätze kombinieren: Schritt 3 - Zusammenfügen

- Für das Zusammenfügen der Daten nutzen wir den Befehl `merge()`. Der Befehl muss wissen, welche Spalte er heranziehen soll, um die verschiedenen Datensätze richtig in einen einzelnen Datensatz zusammenzufügen, das ist in unserem Fall die Spalte "session"
- Für weitere Spalten, die es in jedem Survey gibt (z.B. `created` = der Zeitpunkt zu dem das Survey begonnen wurde, oder `ended` = der Zeitpunkt zu dem das Survey abgeschickt wurde), kann man "suffixes" festlegen, das sind Zeichen, die an die Spaltennamen drangehängt werden, um sie eindeutig einem Survey zuzuordnen
- Für die Suffixes bietet sich der Name der surveys an, wobei der erste survey-Name erst im letzten merge angegeben werden sollte, damit alle Spalten am Ende eindeutig benannt sind:

```
data <- merge(survey1, survey2, by = "session", suffixes = c("", "_survey2"), all = TRUE)
data <- merge(data, survey3, by = "session", suffixes = c("", "_survey3"), all = TRUE)
data <- merge(data, survey4, by = "session", suffixes = c("_survey1", "_survey4"), all = TRUE)
# Geschafft! Alle surveys befinden sich jetzt im Objekt `data`
```

- das Argument `all = TRUE` legt fest, dass alle Zeilen behalten werden sollen, auch wenn es in zwei Surveys keine zusammenpassenden session-IDs gibt. Das ist vor allem für die Studien relevant, in denen es unterschiedliche Bedingungen gab, aber ist auch sonst relevant (z.B. weil der Fragebogen abgebrochen wurde)

Auswertung von Studien

Bedingungsvariable berechnen

- In Studien, in denen es unterschiedliche Bedingungen gab, muss eine neue Variable erstellt werden, die anzeigt in welcher Bedingung eine Versuchsperson war
- Die session-ID und entsprechende Einträge in die "created" oder "ended"-Spalten einer Versuchsperson existieren nur in dem Survey, zu dem sie zugewiesen wurde
- Das bedeutet, nachdem die surveys in einen Datensatz zusammengefügt wurden, ist bekannt in welcher Bedingung eine Versuchsperson war anhand der Tatsache in welcher dieser Spalten eines surveys (siehe suffixes!) Werte drin stehen:

created_condition1	modified_condition1	ended_condition1	expired_condition1	created_condition2	modified_condition2	ended_condition2	expired_condition2	created_condition3
NA	NA	NA	NA	2023-12-11 11:53:13	2023-12-11 11:57:14	2023-12-11 11:57:14	NA	NA
NA	NA	NA	NA	2023-12-10 16:22:12	2023-12-10 16:24:54	2023-12-10 16:24:54	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	2023-12-11 10:46:01
2023-12-10 12:41:06	2023-12-10 12:43:20	2023-12-10 12:43:20	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	2023-12-10 08:43:30	2023-12-10 08:46:24	2023-12-10 08:46:24	NA	NA

Auswertung von Studien

Bedingungsvariable berechnen

- Um eine neue Bedingungsvariable zu erstellen, wird erstmal eine leere Variable angelegt, die sprechend benannt wird (z.B. mit dem Variablenamen **condition**):

```
data$condition <- NA
```

condition
NA

Auswertung von Studien

Bedingungsvariable berechnen

- Wir überschreiben die leeren Werte dieser Variable nun mit einer `ifelse()`-Funktion, welche 3 Argumente braucht:
 - erstes Argument der Funktion ist eine Bedingung ("if" also "wenn"), zu der ein Befehl ausgeführt werden soll. In unserem Fall wollen wir einen Wert einfügen, "wenn die entsprechende `ended`-Spalte nicht leer ist", d.h. in R-Code "nicht leer" = `!is.na()`
 - zweites Argument der Funktion ist der Befehl, der ausgeführt werden soll, wenn die "if"-Bedingung erfüllt ist. In unserem Fall wollen wir eine bestimmte Zahl vergeben, wenn jemand in der entsprechenden Bedingung war, z.B. `1`
 - drittes Argument der Funktion ist der Befehl, der ausgeführt werden soll, wenn die "if"-Bedingung *nicht* erfüllt ist ("else" also "sonst"). In unserem Fall wollen wir den Wert behalten, der bereits drin stand.

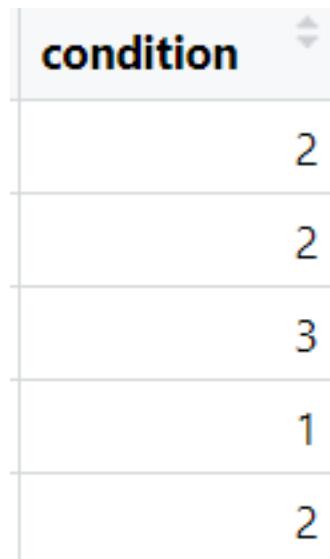
```
data$condition <- ifelse(!is.na(data$ended_condition1), 1, data$condition)
data$condition <- ifelse(!is.na(data$ended_condition2), 2, data$condition)
data$condition <- ifelse(!is.na(data$ended_condition3), 3, data$condition)
```

- Der Befehl in Worten: "Wenn in einer Zeile des Datensatzes die `ended`-Spalte des Surveys von Bedingung 1 nicht leer ist, dann schreibe bitte in die Variable `condition` den Wert 1 rein, und sonst (wenn sie leer ist) behalte den Wert der vorher drin stand."

Auswertung von Studien

Bedingungsvariable berechnen

created_condition1	modified_condition1	ended_condition1	expired_condition1	created_condition2	modified_condition2	ended_condition2	expired_condition2	created_condition3
NA	NA	NA	NA	2023-12-11 11:53:13	2023-12-11 11:57:14	2023-12-11 11:57:14	NA	NA
NA	NA	NA	NA	2023-12-10 16:22:12	2023-12-10 16:24:54	2023-12-10 16:24:54	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	2023-12-11 10:46:01
2023-12-10 12:41:06	2023-12-10 12:43:20	2023-12-10 12:43:20	NA	NA	NA	NA	NA	NA
NA	NA	NA	NA	2023-12-10 08:43:30	2023-12-10 08:46:24	2023-12-10 08:46:24	NA	NA



Auswertung von Studien

Daten bereinigen: Eingabefehler korrigieren

Vor Beginn der Hypothesenprüfung:

- 1) versuchen, Eingabefehler zu identifizieren (nur relevant, falls Eingabe technisch gesehen fehlerhaft sein kann, z.B. bei offenen Antworten oder Eingabe von Paper-Pencil-Fragebögen in eine Datendatei)
- 2) Datensatz um Fehler bereinigen

Eingabefehler:

- oft Werte, die außerhalb des zulässigen Wertebereichs einer Variablen liegen
- hat eine Variable nur wenige Stufen (z.B. Geschlecht: 0, 1), lässt man sich mit einem geeigneten Befehl ausgeben, wie oft die Werte der betrachteten Variablen vorkommen
- Bei Variablen ohne exakt festgelegten Wertebereich (z.B. Alter) ist auf Extremwerte zu achten; so sind Altersangaben größer als 100 z. B. sehr unwahrscheinlich und sollten überprüft werden.
- Extremwerte springen auch bei graphischen Darstellungen ins Auge (Histogramm, Boxplot)

Auswertung von Studien

Daten bereinigen: Eingabefehler korrigieren

Beispiel: Eingabefehler

- Häufigkeiten anzeigen lassen (z.B. Geschlecht: 0, 1) mit dem Befehl `table()`, z.B.:

```
table(data$Geschlecht)
```

- Erhielt man nun die Angabe, dass der Wert »0« (für männlich) 456-mal vorkommt, der Wert »1« 435-mal und der Wert »9« 3-mal, hat man damit bereits 3 Eingabefehler identifiziert.
- Nun lässt man sich die Nummern all derjenigen Fälle ausgeben, bei denen »Geschlecht=9« auftaucht mit dem Befehl `which()`, z.B.:

```
data[which(data$Geschlecht == 9), ]
```

- Bei diesen Personen muss man in den Originalfragebögen nachschauen, welches Geschlecht sie angegeben haben und die entsprechenden Angaben in der Datendatei ändern.

Auswertung von Studien

Daten bereinigen: Missing-Data (unvollständige Datensätze)

Probleme:

- kleineres N → weniger Power
- größere Standardfehler (d.h. ungenauere Schätzungen)
- Verzerrungen, wenn fehlende Werte mit anderen Merkmalen zusammenhängen
- manche Tests lassen sich nicht rechnen

→ Je nach statistischem Test muss ggf. ein Datensatz ohne fehlende Werte (auf den Variablen, die in das Modell eingehen) erstellt werden

```
data_all <- data
data_H1 <- data[!is.na(data$skala1),]
```

(R-Zeile 2 in Worten: Erstelle das Objekt "data_H1" und speichere darin vom Objekt "data" alle Zeilen die in der Spalte "skala1" *nicht* leer (= `!is.na()`) sind, und behalte alle Spalten)

Auswertung von Studien

Daten bereinigen: ggf. nur für eigene Studie relevante Bedingungen auswählen

- Die Gruppen mit dem Interventionsthema vergleichen immer nur 2 Bedingungen (zumindest konfirmatorisch), d.h. es macht Sinn den Datensatz auf die Bedingungen zu kürzen, die für die eigene Forschungsfrage relevant sind
- Hinweis: Auch als Stichprobe sollten primär die Personen beschrieben werden, die an den Bedingungen teilgenommen haben, die für die eigene Forschungsfrage relevant sind
- Wenn man mit 1 und 2 die Bedingungen benannt hat, die für die eigene Forschungsfrage relevant sind, dann kann man den Datensatz mit folgendem Befehl kürzen:

```
data <- data[!is.na(data$condition) & data$condition == 1 | !is.na(data$condition) & data$condition
```

Der Befehl in Worten: Überschreibe das Objekt "data" mit den Zeilen vom Objekt "data" in denen die Spalte "condition" nicht leer ist und entweder 1 oder 2 steht, und behalte alle Spalten)

Auswertung von Studien

Daten bereinigen: Ausschlusskriterien

- Beispiel: Manipulationscheck nicht erfüllt

```
data <- data[data$mc == "Ja, ich habe die Instruktionen befolgt",]
```

(In Worten: Überschreibe das Objekt "data" mit den Zeilen vom Objekt "data" in denen in der Spalte "mc" "Ja, ich habe die Instruktionen befolgt" drinst steht, und behalte alle Spalten)

- In manchen Fällen macht es Sinn die Daten nicht direkt zu überschreiben, sondern zwei (in manchen Fällen auch mehr als zwei) Datensatz-Varianten zu erstellen, z.B. einen Datensatz *mit* den Personen, die den Manipulationscheck nicht erfüllt haben und einen *ohne* diese Personen (diese Datensätze dann entsprechend benennen, z.B. `data_all` und `data_mc_passed`)

```
data_all <- data
data_mc_passed <- data[data$mc == "Ja, ich habe die Instruktionen befolgt",]
```

Auswertung von Studien

Variablen berechnen: Variablen umkodieren und Skalen berechnen

- siehe Einheit zur Operationalisierung
- ggf. einzelne Items in die richtige Richtung umpolen, Beispiel-Code bei einer 7-stufigen Skala:

```
data$skala1_item1_r <- (data$skala1_item1-8)*-1
```

- Skalenwerte für Subskalen und Gesamtwerte bilden
 - Prüfen, ob Summenwert mit `rowSums()`, Mittelwert mit `rowMeans()` oder was anderes nötig ist
 - Dabei unbedingt darauf achten, dass bei Personen, bei denen es fehlende Werte für eine Skala gibt gemäß der Präregistrierung vorgegangen wird (z.B. die Skala dann nicht gebildet wird z.B. indem `na.rm = FALSE` als Argument übergeben wird)

```
data$skala1 <- rowSums(data[,c("skala1_item1_r", "skala1_item2", "skala1_item3")], na.rm = FALSE)
data$skala2 <- rowMeans(data[,c("skala2_item1", "skala2_item2", "skala2_item3")], na.rm = FALSE)
```

Auswertung von Studien

Variablen berechnen: Variablen umkodieren und Skalen berechnen

- Überprüfung, ob alles geklappt hat (z.b. hinsichtlich Wertebereichen), z.B. mit dem Befehl `describe()` aus dem `psych` package

```
library(psych)
describe(data$item1)
describe(data$item1_r)
describe(data$skala1)
```

- Der Befehl bietet sich auch bei nicht selbst-erstellten Variablen zur Überprüfung an, ob alles normal aussieht

Auswertung von Studien

Variablen berechnen: z-Standardisierung

- Variablen ggf. zur besseren Interpretierbarkeit von Modellparametern z-standardisieren
- Konsequenz: Mittelwert der Variable auf 0 verschoben, Standardabweichung auf 1 verschoben ("normiert")
- Ist sinnvoll bei Variablen die keine natürliche 0 haben und/oder bei denen ein Abstand von 1 nicht natürlich ist (z.B. bei den meisten Likert Skalen)
- Beispiel:

```
data$skala1_z <- (data$skala1 - mean(data$skala1, na.rm = TRUE))/sd(data$skala1, na.rm = TRUE)
```

Auswertung von Studien

Variablen berechnen: Prä-Post-Differenz bilden

- Macht man einen Prä-Post-Vergleich, den man als AV vorhersagen möchte, muss man die entsprechende Differenz bilden:

Beispiel:

```
data$RS1_diff <- data$RS1_post - data$RS1_pre
```

Auswertung von Studien

Datenvorbereitung

- ggf. sind noch weitere Schritte in der Datenvorbereitung nötig, z.B. Faktorvariablen als solche zu kodieren und ihnen sinnvolle Labels (= Gruppenbezeichnungen) zu vergeben
- Man sollte sich vor Beginn der Analysen einen Überblick darüber gemacht haben, dass möglichst alle Variablen so aussehen wie sie aussehen sollen → die Datenvorbereitung sollte abgeschlossen sein, bevor mit den Hypothesenprüfungen begonnen wird
 - Stellt sich erst im Nachhinein heraus, dass noch gravierende Kodierungs- oder Eingabefehler in den Daten stecken, müssen die Analysen wiederholt werden
 - Zudem bestünde die Gefahr, beim Bereinigen der Daten bewusst oder unbewusst im Sinne der eigenen Hypothesen vorzugehen
 - Dies betrifft auch die Frage, welche Fälle wegen fehlender oder fragwürdiger Angaben ggf. ganz aus den Analysen ausgeschlossen werden sollen

Auswertung von Studien

Datenvorbereitung

- Sie stellen bei der Datenvorbereitung fest, dass es total sinnvoll ist, bestimmte Angaben oder Personen auszuschließen, obwohl Sie das nicht präregistriert haben? Kein Problem! Erklären Sie, warum das so ist.
- Bei jeder Abweichung von der Präregistrierung kann man sich überlegen zu berichten, inwiefern sich die Ergebnisse dadurch ändern (z.B. von signifikant zu nicht signifikant, Vorzeichen ändert sich oder bleibt gleich)
- Beispiel aus einem Paper:

Analytical Strategy

As preregistered, we winsorized outliers with scores outside $M \pm 3 \times SD$ to the respective lower or upper bound. This procedure was used for 15 observations of PA and 58 observations of NA. We repeated analyses without outlier correction and found almost identical results. We used multilevel modeling (Hoffman, 2015)

Auswertung von Studien

Code-Review

- Macht es Sinn, dass nur einer aus Ihrer Gruppe die komplette Datenvorbereitung und Analyse macht? **Nein.**
 - Erstens, lernt dann nur einer wie Datenvorbereitung und Analysen funktionieren
 - Zweitens, machen wir alle Fehler (ich auch)
- **Idee vom Code-Review:** 4-Augen-Prinzip
 - Einer prüft den Code vom anderen
 - Voraussetzung: Code ist gut kommentiert und kann vom anderen nachvollzogen werden
 - mit Hashtags kann man in R Kommentare oder Überschriften schreiben, z.B.

```
# Summenwert für Skala1 bilden
data$skala1 <- rowSums(data[,c("skala1_item1", "skala1_item2", "skala1_item3")], na.rm = FALSE)
```

Auswertung von Studien

Ergebnis-Abgleich

- **Ein Abgleich von Ergebnissen ist noch besser:** Jeder macht die Datenvorbereitung, und am Ende wird verglichen, ob alle zum gleichen Ergebnis kommen, d.h. hier zu den gleichen vorverarbeiteten Datensätzen und im nächsten Schritt zu gleichen Ergebnissen (der Weg muss nicht gleich sein - Hauptsache es kommt dasselbe raus)
 - Vergleich zweier Datensätze mit dem Befehl `all.equal()`
 - Zuvor müssen beide Datensätze mit `load()` eingelesen werden

```
load(file = "../processed_data/final_data_musterfrau.RData")
load(file = "../processed_data/final_data_mustermann.RData")

all.equal(final_data_musterfrau, final_data_mustermann)
```

→ R gibt Ihnen dann aus, an welchen Stellen sich Ihre Datensätze ggf. unterscheiden

Stichprobenbeschreibung und Deskriptivstatistik

Methodenteil: Stichprobenbeschreibung

- Hat man die Vorverarbeitung der Daten durchlaufen, erstellt man üblicherweise zunächst eine Stichprobenbeschreibung für den Methodenteil und eine Tabelle mit Deskriptivstatistik für den Ergebnisteil, bevor man zu den Hypothesentests übergeht
- Bei der Stichprobenbeschreibung berichtet man für die gängigen sozialstatistischen bzw. soziodemographischen Merkmale ...
 - Geschlecht
 - Alter
 - Familienstand
 - Bildungsgrad, Tätigkeit
 - Einkommen
 - Wohnort
 - ...
- Auswertung
 - **Numerische Merkmale:** empirische Range und M (SD) oder Med (IQR) mit den Befehlen `range()`, `mean()`, `sd()`, `median()`, `IQR()`
 - **Kategoriale Merkmale:** N (%) mit den Befehlen `table()` und `prop.table(table())`
 - In die Klammern der Befehle die relevante(n) Spalte(n) im Datensatz auswählen, z.B. `mean(data$variable1)`
- Gruppen mit eigenem Thema werden aufgrund der Anonymitätsanforderung nicht sehr viele soziodemographische Informationen erhoben haben. Dann bleibt es bei der Stichprobengröße (ggf. pro Gruppe) und eben den stichprobenbezogenen Variablen die Sie erhoben haben.

Stichprobenbeschreibung und Deskriptivstatistik

Methodenteil: Stichprobenbeschreibung

- Neben allgemeinen soziodemografischen Variablen werden im Rahmen der Stichprobenbeschreibung auch weitere für das Studienthema relevante Merkmale beschrieben
- Beispiel: Studie über Computerspiele
 - die Computererfahrungen der Probanden
 - durchschnittliche Spielzeit/Woche
 - ...

Stichprobenbeschreibung und Deskriptivstatistik

Ergebnisteil: Deskriptivstatistik

- Die deskriptiven Ergebnisse werden in einer Tabelle zusammengefasst
- Werden im Studiendesign mehrere Gruppen untersucht sollte es eine Spalte pro Gruppe, sowie eine "Gesamt" Spalte geben
 - Ggf. kann dann auch für jede Variable ein t-test berechnet werden, um zu prüfen, ob es Gruppenunterschiede gibt (wären dann Störeinflüsse)
 - Debatte: Viele Signifikanztests haben hohe α -Fehler Kumulierung (daher oft nur deskriptive Statistik)
- **In der Regel gilt es alle Fragebögen deskriptiv darzustellen (und im Methodenteil zu beschreiben), die Sie im Rahmen der Studie erhoben haben**, da Sie diese ja in irgendeiner Form für relevant für das Studienthema erachtet haben (Ausnahme sind große Studien, wo es ggf. den Rahmen sprengen würde alle Fragebögen zu berichten)
- Außerdem ist es üblich die Korrelationen zwischen allen erhobenen Variablen darzustellen (in einer zweiten Tabelle oder zusammen mit der ersten Tabelle); wenn eine Korrelation Teil einer Hypothese ist, wird sie meist trotzdem auch in der Gesamttabelle aufgeführt, aber im Fließtext separat angesprochen
- Manipulationschecks werden meist vor den Hypothesentests im Fließtext berichtet

Stichprobenbeschreibung und Deskriptivstatistik

Methodenteil oder Ergebnisteil: Reliabilität

- Angaben zur Reliabilität (α oder ω) sind meist im Fließtext im Methodenteil bei der Beschreibung der Instrumente, manchmal findet man sie aber auch in der Tabelle der Deskriptivstatistik
- Berechnung der Reliabilität auf eigenen Daten z.B. mit `ci.reliability()` aus dem package **MBESS** (relevantes Ergebnis: "est" = "estimate" = Schätzwert der Reliabilität, am Besten zusammen mit Konfidenzintervall "ci.lower" und "ci.upper")

```
library(MBESS)
ci.reliability(data = data[, c("item1", "item2", "item3")], type = "omega")
```

Stichprobenbeschreibung und Deskriptivstatistik

Beispiel für eine Deskriptivstatistik-Tabelle

- Hier wurden Mittelwert, Standardabweichungen, Range und Korrelationen gemeinsam dargestellt

Table 1. Descriptive statistics and correlations for trait measures

Variables	<i>M</i> (<i>SD</i>)	Range	ω_t	1	2	3	4	5
1. Implicit pnCommunion	5.34 (2.12)	1 to 12		.23				
2. Explicit desire for closeness	6.16 (0.69)	3.5 to 7	.86	.20*	.10			
3. Couple satisfaction index	66.30 (10.23)	32 to 81	.92	.27***	.61***	.44***		
4. Positive relationship quality	6.11 (1.05)	1 to 7	.92	.17*	.44***	.47***	.23	
5. Negative relationship quality	2.01 (1.27)	1 to 7	.91	-.12	-.26**	-.45***	-.17*	.18

Note. *N* = 152 individuals from 77 couples. pnCommunion = partner-related need for Communion. The reliability coefficient ω_t refers to McDonald's omega total, calculated with the MBESS package (Kelley, 2016). Cronbach's α was equal to ω_t for all measures, except for the explicit need for closeness, α was .87 (calculated with the psych package, Revelle, 2016). Correlations below the diagonal refer to associations between individuals. Correlations on the diagonal refer to dyadic associations. *M* (*SD*) of pnCommunion refer to raw motive scores (number of motive categories). Correlations of pnCommunion were calculated with motive scores corrected for word count. * $p < .05$, ** $p < .01$, *** $p < .001$.

Stichprobenbeschreibung und Deskriptivstatistik

Bewertungsschema

Methodenteil:

Beschreibung Stichprobe	0	keine/unzureichende Beschreibung der Stichprobe
	1	Stichprobe ist mit allen relevanten Merkmalen beschrieben

Ergebnisteil:

Deskriptive Statistiken (MW, SD, Korrelationen etc.)	0	Deskriptive Statistiken der Variablen, die in irgendeine Analyse eingegangen sind: fehlen überwiegend oder ganz
	1	Deskriptive Statistiken der Variablen, die in irgendeine Analyse eingegangen sind: fehlen teilweise
	2	Deskriptive Statistiken der Variablen, die in irgendeine Analyse eingegangen sind: sind vorhanden

Schritt 1: Ordnerstruktur nutzen

- Bereits vorhandene Dateien einsortieren

Schritt 2: Code-Review besprechen

- 4-Augen-Prinzip, oder jeder separat und dann vergleichen?
- Bei 4-Augen-Prinzip: Wer macht welchen Teil, und wer reviewt welchen Teil?

CFH Exkursion Wien



Start: 24.10. gegen Mittag; Ende: 26.10. am späteren Nachmittag; Übrigens können selbstverständlich alle Teilnehmenden eigenständig auch früher anreisen und/ oder länger vor Ort bleiben.

Folgende Programmpunkte sind geplant (und die Eintrittspreise/ Leistungen im Teilnahmepreis inbegriffen):

- Besuch & Workshop an der Charlotte Fresenius Privat Universität Wien (Freitag)
- Gemeinsames Abendessen (Freitag)
- Stadtführung
- Besuch Freud Museum
- Besuch Viktor Frankl-Museum
- Besuch Naschmarkt