Charlotte Fresenius Hochschule

Studiengang: Psychologie (B. Sc.)

Studienort: Köln

Empririsch Wissenschaftliches Arbeiten

Dokumentation der Datenaufbereitung:

‘Dataset 71.txt’

Prof. Dr. Stephan Huber

Gutachter: —

Abgabedatum: 03.07.2024

Zusammenfassung

Dieses Dokument beschreibt die Datenaufbereitung der Datei ‘Dataset 71.txt’ und stellt sicher, dass die Ergebnisse replizierbar sind. Alle Schritte werden mit R durchgeführt.

Datenbeschreibung

Über den Datensatz ist nur wenig bekannt. Es gibt 21 Variablen, die Items einer Umfrage darstellen. Die Antwortmöglichkeiten sind auf einer fünfstufigen Likert-Skala angegeben.

Datenaufbereitung

Vorbereitung

Zunächst werden die benötigten R-Pakete geladen. Hierzu verwende ich das Paket `pacman`. Sollte dieses Paket auf dem verwendeten Computer nicht installiert sein, wird es mit der ersten der folgenden Zeilen installiert. Die zweite Zeile lädt die benötigten Pakete und die dritte bereinigt den aktuellen Arbeitsbereich. Schließlich wird das Arbeitsverzeichnis festgelegt.

```
if (!require(pacman)) install.packages("pacman")
pacman::p_load(tidyverse, janitor, psych, tinytable, ggstats,
               modelsummary, knitr, kableExtra, labelled)
rm(list = ls())
```

Datenimport

Die Datei “Dataset 71.txt” wird mit der Funktion `read.delim` in R eingelesen.

```
df_raw <- read.delim("Dataset 71.txt")
```

Tabelle 1 zeigt einen Ausschnitt des Rohdatensatzes. [1]

Datenexploration

Im Folgenden werde ich die Daten genauer untersuchen, um eventuelle Datenfehler zu identifizieren und diese später zu bereinigen. Zunächst ist festzuhalten, dass folgende Werte im Datensatz vorhanden sind: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 0, NaN. Ohne die Variable `ID`, die offensichtlich eine laufende Nummer von 1 bis 70 ist, sind

[1] Dieses Objekt wird mit der Funktion `tt` aus dem Paket `tinytable` veranschaulicht (Arel-Bundock, 2024).

folgende Werte enthalten: 1, 0, 4, 3, 5, NaN, 2, 22, 44, 33, 6. Das ist seltsam. Eigentlich sollten hier, entsprechend der Likert-Skala, nur Werte von 1 bis 5 enthalten sein. Die folgenden Tabellen sollen einen besseren Einblick in den Datensatz ermöglichen:

- Tabelle 2 zeigt für jede im Datensatz enthaltene Variable die unterschiedlichen Werte an. [2]
- Tabelle 3 zeigt, in wie vielen verschiedenen Items bestimmte Beobachtungen vorkommen. [3]
- Tabelle 4 zeigt einige deskriptive Statistiken. Hier fällt auf, dass `NaN` enthalten sind und einige ungewöhnliche Werte, die außerhalb der Skala von 1 bis 5 liegen. [4]

Diese Anomalien müssen später bereinigt beziehungsweise vermerkt werden.

Datenbereinigung

Zunächst nehme ich einige kosmetische Bereinigungen vor. Dabei passe ich die Variablennamen entsprechend gängigen Konventionen an, indem ich Leerzeichen und Punkte entferne sowie Großbuchstaben vermeide. Dies geschieht mit der Funktion `clean_names`. Darüber hinaus werden die NaN-Werte durch NA ersetzt, und die Beobachtungen gelöscht, die in allen Items ausschließlich fehlende Werte enthalten.

```
df_cosmetic <- df_raw |>
  clean_names() |>
  as_tibble() |>
  # Ersetzen von NaN-Werten durch NA
  mutate(across(everything(), ~ if_else(is.nan(.), NA, .))) |>
  #Entfernen von Zeilen, bei denen alle "item_"-Spalten NA sind
```

[2] Die Datengrundlage wird manuell erstellt. Hierbei kommen verschiedene Funktionen des `dplyr` Pakets zum Einsatz (Wickham et al., 2023). Dargestellt wird der Datensatz mit der Funktion `kable` aus dem `knitr` Paket (Xie, 2024.)

[3] Tabelle 3 wird mit Hilfe der Funktion `tabyl` erstellt, die Teil des Pakets `janitor` ist (Firke, 2023).

[4] Tabelle 4 wird mit Hilfe der Funktion `datasummary_skim` erstellt, die Teil des Pakets `modelsummary` ist (Arel-Bundock, 2022).

```
  rowwise() |>
  filter(!all(across(starts_with("item_"), ~ is.na(.)))) |>
  ungroup()
```

Wie bereits erwähnt, gibt es einige fehlende Werte (NA) sowie ungewöhnliche Werte, die außerhalb der Likert-Skala liegen, also nicht im Wertebereich von 1 bis 5. Diese gilt es zu identifizieren. Da ich die genauen Gründe hierfür nicht kenne, werde ich dazu verschiedene Variablen erzeugen. Die genaueren Beschreibungen zu den Variablen befinden sich als Kommentar im folgenden Code-Ausschnitt:

```
df <- df_cosmetic |>
  rowwise() |>
  # Berechnung des größten absoluten Werts in "item_"-Spalten für jede Zeile
  mutate(outlier = max(abs(c_across(starts_with("item_"))), na.rm = TRUE)) |>
  # Markieren, ob ein Ausreißer (> 5 oder gleich 0) vorhanden ist
  mutate(has_outlier = if_else(outlier > 5 | outlier == 0, TRUE, FALSE)) |>
  # Zählen der Werte, die größer als 5 sind, für jede Zeile
  mutate(count_larger_5 =
           sum( c_across(starts_with("item_")) > 5 |
                c_across(starts_with("item_")) == 0, na.rm = TRUE)) |>
  # Zählen der Tippfehler (11, 22, 33, 44, 55) für jede Zeile
  mutate(count_typos = sum(c_across(starts_with("item_")) %in%
                             c(11, 22, 33, 44, 55), na.rm = TRUE)) |>
  # Markieren, ob mehr Werte größer als 5 sind als Tippfehler
  mutate(has_larger_5_notypos = (count_typos < count_larger_5)) |>
  # Markieren, ob Tippfehler vorhanden sind
  mutate(has_typos = count_typos > 0 ) |>
  # Markieren, ob NA-Werte in "item_"-Spalten vorhanden sind
  mutate(has_nas = if_else(anyNA(pick(starts_with("item_"))), TRUE, FALSE)) |>
  # Markieren, ob eine Zeile vollständig ist (keine Ausreißer und keine NAs)
```

```
  mutate(complete = (has_outlier == FALSE & has_nas == FALSE)) |>
  ungroup()
```

Die Variablen `has_typos`, `has_nas`, `has_larger_5_notypos` und `has_outlier` zeigen nun an, ob und welche Probleme in der jeweiligen Beobachtung vorliegen. Diese Variablen sind wie folgt definiert:

- `has_nas`: Ist `TRUE`, wenn mindestens eine Beobachtung ein `NA` ist.
- `has_typos`: Ist `TRUE`, wenn mindestens eine Beobachtung die Werte 11, 22, 33, 44 oder 55 aufweist.
- `has_outlier`: Ist `TRUE`, wenn mindestens eine Beobachtung einen Wert größer als 5 (in absoluten Zahlen) aufweist.
- `has_larger_5_notypos`: Ist `TRUE`, wenn mindestens eine Beobachtung einen Wert größer als 5 (in absoluten Zahlen) aufweist und diese Zahl(en) nicht 11, 22, 33, 44 oder 55 ist.

Die Werte 11, 22, 33, 44 oder 55 könnten Tippfehler sein, bei denen die Zahl versehentlich doppelt eingegeben wurde. Dies werde ich später versuchen zu berücksichtigen und zu bereinigen.

Datensatzerstellung

In diesem Schritt erstelle ich Datensätze, die ich zur Analyse verwenden kann. Hierbei werde ich zwei verschiedene Datensätze erstellen. Einen Datensatz, in dem ich ausschließlich Beobachtungen berücksichtige, die scheinbar frei von Fehleingaben und fehlenden Werten sind. Dieser Datensatz wird als `df_complete` bezeichnet. Darüber hinaus speichere ich alle Variablen, entsprechend der Likert-Skala, als Faktorvariablen ab und versehe sie mit einem entsprechenden Label.

Die Variablen `has_typos`, `has_nas`, `has_larger_5_notypos` und `has_outlier` indizieren nun, ob und welche Probleme in der jeweiligen Beobachtung vorliegen. Die Variablen sind wie folgt definiert:

- `has_nas`: Ist `TRUE`, wenn mindestens eine Beobachtung ein `NA` ist.

- `has_typos`: Ist `TRUE`, wenn mindestens eine Beobachtung die Werte 11, 22, 33, 44, oder 55 aufweist.
- `has_outlier`: Ist `TRUE`, wenn mindestens eine Beobachtung die Werte in absoluten Zahlen größer als 5 ist
- `has_larger_5_notypos`: Ist `TRUE`, wenn mindestens eine Beobachtung die Werte in absoluten Zahlen größer als 5 ist und diese Zahl(en) nicht 11, 22, 33, 44, oder 55 ist.

Die Werte 11, 22, 33, 44, oder 55 könnten besonders sein, denn hier könnte man vermuten, dass hier schlicht ein Tippfehler vorliegt. Also die Zahl versehentlich doppelt eingegeben wurde. Dies werde ich später versuchen, zu berücksichtigen und zu bereinigen.

```
# Labels definieren
likert_levels <- c(
  "Stimme überhaupt nicht zu",
  "Stimme nicht zu",
  "Neutral",
  "Stimme zu",
  "Stimme voll und ganz zu"
)

# Faktorisierung der Items und hinzufügen eines Labels
df_chr <- df |>
  mutate(across(starts_with("item_"),
                ~ case_when(
                  . == 1 ~ "Stimme überhaupt nicht zu",
                  . == 2 ~ "Stimme nicht zu",
                  . == 3 ~ "Neutral",
                  . == 4 ~ "Stimme zu",
                  . == 5 ~ "Stimme voll und ganz zu",
                  TRUE ~ as.character(.)
```

```
                ))) |>
  mutate(across(starts_with("item_"), ~ factor(.x, levels = likert_levels)))


df_complete <- df_chr |>
  filter(complete == TRUE)
```

Der Datensatz `df_complete` hat 48 Beobachtungen.

Der zweite Datensatz ist als `df_cleaned` betitelt. Hierbei unterstelle ich, dass es sich bei den Eingaben mit den Werten 11, 22, 33, 44 oder 55 um Tippfehler handelt. Diese rekodiere ich entsprechend in 1, 2, 3, 4 und 5 um. Alle übrigen Werte außerhalb des Wertebereichs 1 bis 5 bezeichne ich als `NA`.

```
df_cleaned <- df |>
  # Ersetzen von bestimmten Werten (11, 22, 33, 44, 55) in "item_"-Spalten
  mutate(across(starts_with("item_"), ~ case_when(
    . == 11 ~ 1,
    . == 22 ~ 2,
    . == 33 ~ 3,
    . == 44 ~ 4,
    . == 55 ~ 5,
    TRUE ~ .
  ))) |>
  # Ersetzen von Werten größer als 5 durch NA in "item_"-Spalten
  mutate(across(starts_with("item_"), ~ if_else(. > 5 | . == 0, NA, .))) |>
  mutate(across(starts_with("item_"),
                ~ case_when(
                  . == 1 ~ "Stimme überhaupt nicht zu",
                  . == 2 ~ "Stimme nicht zu",
                  . == 3 ~ "Neutral",
                  . == 4 ~ "Stimme zu",
```

```
                    . == 5 ~ "Stimme voll und ganz zu",
                    TRUE ~ as.character(.)
                  ))) |>
  mutate(across(starts_with("item_"), ~ factor(.x, levels = likert_levels)))
```

Der Datensatz `df_cleaned` hat 69 Beobachtungen.

Schließlich speichere ich die aktuelle Arbeitsumgebung in einer `.RData`-Datei.

```
save.image("data_71.RData")
```

Auswertung

Antwortverteilung zu den gestellten Fragen

Abbildung 1 und Abbildung 2 zeigen die Verteilung der Antworten. Die erste Abbildung verwendet den Datensatz `df_complete`, bei dem nur die Befragungen berücksichtigt wurden, bei denen keine Auffälligkeiten gefunden wurden. Die zweite Abbildung verwendet den Datensatz `df_cleaned`, bei dem einige Bereinigungen durchgeführt wurden und einige Fragen nicht verfügbar waren. [5]

Überprüfung der neu erstellten Datensätze

Tabelle 5 und Tabelle 6 enthalten die unterschiedliche Werte in den Variablen für die Datensätze `df_complete` und `df_cleaned`.

[5] Beide Abbildungen werden mit der Funktion `gglikert` erstellt, welche Teil des `ggstats`-Pakets ist (Larmarange, 2024).

Literaturverzeichnis

Arel-Bundock, V. (2022). modelsummary: Data and Model Summaries in R. *Journal of Statistical Software*, *103*(1), 1–23. https://doi.org/10.18637/jss.v103.i01

Arel-Bundock, V. (2024). *tinytable: Simple and Configurable Tables in 'HTML', 'LaTeX', 'Markdown', 'Word', 'PNG', 'PDF', and 'Typst' Formats*. https://CRAN.R-project.org/package=tinytable

Firke, S. (2023). *janitor: Simple Tools for Examining and Cleaning Dirty Data*. https://CRAN.R-project.org/package=janitor

Larmarange, J. (2024). *ggstats: Extension to 'ggplot2' for Plotting Stats*. https://CRAN.R-project.org/package=ggstats

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr

Xie, Y. (2024). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://yihui.org/knitr/

Tabelle 1

Ausschnitt des Rohdatensatz

ID	Group	Item.1	Item.2	Item.3	Item.4	Item.5	Item.6	Item.7	Item.8	Item.9
1	1	4	3	4	5	2	3	2	2	3
2	0	4	4	3	4	4	5	3	3	4
3	1	1	3	4	3	3	4	4	3	4
4	1	3	3	5	3	2	3	1	3	2
5	1	3	3	2	5	4	4	4	3	4
6	0	3	3	2	2	2	3	2	2	2

Tabelle 2

Unterschiedliche Werte in den Variablen

Attribute	Values
Group	0, 1
Item.1	1, 2, 3, 4, 5
Item.2	1, 2, 3, 4, 5, 22
Item.3	1, 2, 3, 4, 5, 44
Item.4	1, 2, 3, 4, 5, 44
Item.5	1, 2, 3, 4, 5
Item.6	1, 2, 3, 4, 5
Item.7	1, 2, 3, 4, 5
Item.8	1, 2, 3, 4, 5, 6, 33
Item.9	1, 2, 3, 4, 5, 33
Item.10	1, 2, 3, 4, 5, 44
Item.11	1, 2, 3, 4, 5, 6
Item.12	1, 2, 3, 4, 5
Item.13	1, 2, 3, 4, 5
Item.14	1, 2, 3, 4, 5, 33
Item.15	1, 2, 3, 4, 5, 6
Item.16	1, 2, 3, 4, 5, 6
Item.17	1, 2, 3, 4, 5, 6
Item.18	1, 2, 3, 4, 5
Item.19	1, 2, 3, 4, 5, 6
Item.20	1, 2, 3, 4, 5
Item.21	1, 2, 3, 4, 5, 22

Tabelle 3

Häufigkeitstabelle der unterschiedlichen Werte (pro item)

long$count	n	percent	valid_percent
0	1	0.0070423	0.0082645
1	22	0.1549296	0.1818182
2	21	0.1478873	0.1735537
3	21	0.1478873	0.1735537
4	21	0.1478873	0.1735537
5	21	0.1478873	0.1735537
6	6	0.0422535	0.0495868
22	2	0.0140845	0.0165289
33	3	0.0211268	0.0247934
44	3	0.0211268	0.0247934
NaN	21	0.1478873	NA

Anmerkung. Die Tabelle zeigt an, in wie vielen Items die jeweiligen Werte vorkommen.

Tabelle 4

Deskriptive Statistiken zum Rohdatensatz

	Unique	Missing Pct.	Mean	SD	Min	Median	Max	Histogram
ID	70	0	35.5	20.4	1.0	35.5	70.0	
Group	2	0	0.5	0.5	0.0	0.0	1.0	
Item.1	6	4	2.9	1.0	1.0	3.0	5.0	
Item.2	7	1	3.1	2.5	1.0	3.0	22.0	
Item.3	7	3	3.4	5.1	1.0	3.0	44.0	
Item.4	7	3	3.8	5.1	1.0	3.0	44.0	
Item.5	6	3	3.0	1.0	1.0	3.0	5.0	
Item.6	6	1	3.1	1.1	1.0	3.0	5.0	
Item.7	6	1	2.9	1.1	1.0	3.0	5.0	
Item.8	8	1	3.4	3.7	1.0	3.0	33.0	
Item.9	7	3	3.4	3.8	1.0	3.0	33.0	
Item.10	7	1	3.9	5.0	1.0	3.0	44.0	
Item.11	7	3	2.8	1.2	1.0	3.0	6.0	
Item.12	6	1	2.9	1.0	1.0	3.0	5.0	
Item.13	6	4	2.9	1.0	1.0	3.0	5.0	
Item.14	7	1	3.4	3.8	1.0	3.0	33.0	
Item.15	7	3	3.2	1.2	1.0	3.0	6.0	
Item.16	7	4	2.9	1.1	1.0	3.0	6.0	
Item.17	7	1	3.0	1.1	1.0	3.0	6.0	
Item.18	6	1	3.1	1.0	1.0	3.0	5.0	
Item.19	7	1	3.0	1.1	1.0	3.0	6.0	
Item.20	6	1	3.1	1.1	1.0	3.0	5.0	
Item.21	7	1	3.1	2.6	1.0	3.0	22.0	

Tabelle 5

Unterschiedliche Werte in den Variablen (df_complete)

Attribute	Values
group	0, 1
item_1	1, 2, 3, 4, 5
item_2	1, 2, 3, 4, 5
item_3	1, 2, 3, 4, 5
item_4	1, 2, 3, 4, 5
item_5	1, 2, 3, 4, 5
item_6	1, 2, 3, 4, 5
item_7	1, 2, 3, 4, 5
item_8	1, 2, 3, 4, 5
item_9	1, 2, 3, 4, 5
item_10	1, 2, 3, 4, 5
item_11	1, 2, 3, 4, 5
item_12	1, 2, 3, 4, 5
item_13	1, 2, 3, 4, 5
item_14	1, 2, 3, 4, 5
item_15	1, 2, 3, 4, 5
item_16	1, 2, 3, 4, 5
item_17	1, 2, 3, 4, 5
item_18	1, 2, 3, 4, 5
item_19	1, 2, 3, 4, 5
item_20	1, 2, 3, 4, 5
item_21	1, 2, 3, 4, 5

Tabelle 6

Unterschiedliche Werte in den Variablen (df_cleaned)

Attribute	Values
group	0, 1
item_1	1, 2, 3, 4, 5
item_2	1, 2, 3, 4, 5
item_3	1, 2, 3, 4, 5
item_4	1, 2, 3, 4, 5
item_5	1, 2, 3, 4, 5
item_6	1, 2, 3, 4, 5
item_7	1, 2, 3, 4, 5
item_8	1, 2, 3, 4, 5
item_9	1, 2, 3, 4, 5
item_10	1, 2, 3, 4, 5
item_11	1, 2, 3, 4, 5
item_12	1, 2, 3, 4, 5
item_13	1, 2, 3, 4, 5
item_14	1, 2, 3, 4, 5
item_15	1, 2, 3, 4, 5
item_16	1, 2, 3, 4, 5
item_17	1, 2, 3, 4, 5
item_18	1, 2, 3, 4, 5
item_19	1, 2, 3, 4, 5
item_20	1, 2, 3, 4, 5
item_21	1, 2, 3, 4, 5

Abbildung 1

Antwortverteilung zu den gestellten Fragen (df_complete)

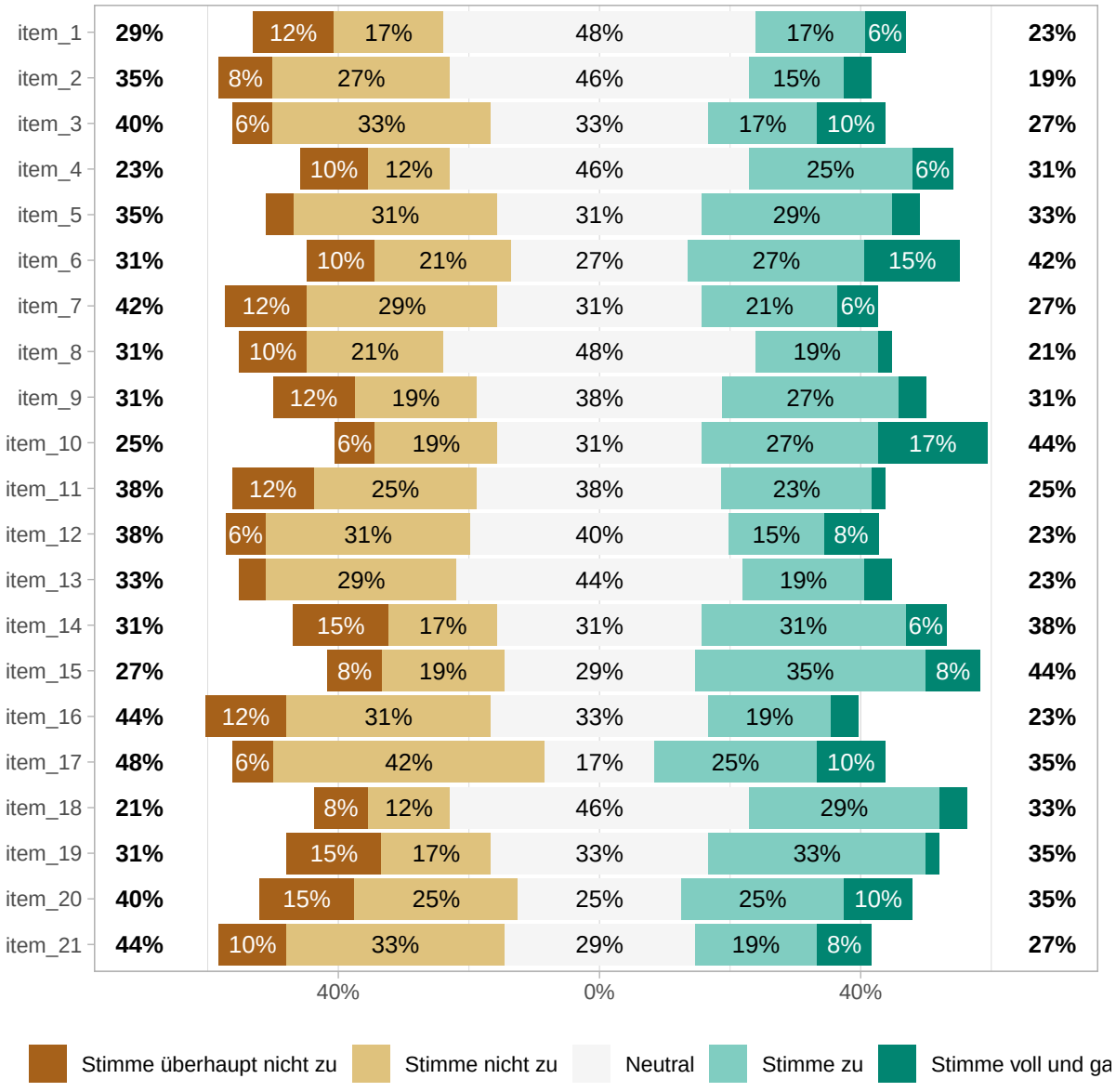

Abbildung 2

Antwortverteilung zu den gestellten Fragen (df_cleaned)