

Scaling

```
# Load any necessary R packages
library(dplyr)
library(modelsummary)
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

For illustration, we will use the open data made available by the authors of this paper: <https://www.nature.com/articles/s41467-021-24786-2>. Specifically, we will focus on the 4-item measure of EBEP (extreme behavioral expressions of prejudice) for Study 3, with detailed description in this Supplementary Information file (p. 10): https://static-content.springer.com/esm/art%3A10.1038%2Fs41467-021-24786-2/MediaObjects/41467_2021_24786_MOESM1_ESM.pdf.

Data Import

```
ebep_dat <- read.csv("https://osf.io/download/mkx6z/")
ebep_dat |>
  select(starts_with("ebep")) |>
  head(n = 10L)
```

	ebep_fb_justified	ebep_flyer_justified	ebep_yell_justified
1	2	2	2
2	1	1	1
3	1	1	1
4	5	5	2
5	1	1	1
6	1	1	1
7	1	1	1
8	2	2	2
9	3	2	1

	3	2	1
10	ebep_punch_justified		
1	1		
2	1		
3	1		
4	7		
5	1		
6	1		
7	1		
8	2		
9	1		
10	1		

Guttman and Likert Scaling

Q1 & Q2

Q1: Create a summary table for the EBEP items. You may use the `modelsummary::datasummary_skim()` function, but other functions are fine too.

Q2: How many scale points are there for the EBEP items? What are the response labels?

Errors in Guttman Scaling

Based on the item means and the item wordings, one may suspect that the items are in the order of increasing prejudice (disclaimer: this may not be the intention of the authors), which would be similar to the idea of Guttman scaling. However, a Guttman scale usually has binary responses, so we can recode 1 (not at all justified) to 0 and 2 or above (slightly justified) to 1.

```
ebep_bin <- ebep_dat |>
  select(starts_with("ebep"))
ebep_bin[] <- apply(ebep_bin, 2, function(x) as.integer(x >= 2))
ebep_bin |> head(n = 10L)
```

	ebep_fb_justified	ebep_flyer_justified	ebep_yell_justified
1	1	1	1
2	0	0	0
3	0	0	0
4	1	1	1
5	0	0	0
6	0	0	0

7	0	0	0
8	1	1	1
9	1	1	0
10	1	1	0

	ebep_punch_justified
1	0
2	0
3	0
4	1
5	0
6	0
7	0
8	1
9	0
10	0

We can then check how many respondents violate this order by answering a 0 in a lower-numbered item but a 1 in a higher-numbered item (e.g., they think that punching is justified, but distributing fliers is not).

```
# Check for participant 119
ebep_bin[119, ]
```

```
      ebep_fb_justified ebep_flyer_justified ebep_yell_justified
119              0              0              0
      ebep_punch_justified
119              1
```

```
# The `diff()` function computes the difference of item 2 - item 1, item 3 - item 2, and so on
diff(as.numeric(ebep_bin[119, ]))
```

```
[1] 0 0 1
```

```
# Find all participants with a 1 followed by a 0
errors <- apply(ebep_bin, 1, function(x) any(diff(x) == 1))
```

Q3 to Q5

Q3: What is the proportion of respondents violating the Guttman scale ordering?

Q4: The following computes the sum scores for the EBEP items on the original 7-point scale. Obtain the sum scores for the recoded binary items, and show a scatterplot of the

two sum scores.

```
# Sum scores of the original 7-point scale
ebep_sum <- ebep_dat |>
  select(starts_with("ebep")) |>
  rowSums()
# Sum scores of the recoded binary items

# Scatterplot
```

Q5: Which set of sum scores do you think is more accurate? Why?

Missing Items

While the data set we used have complete data on the EBEP items, when a participant misses some items, it is common to compute the mean of all answered items (which is known as the mean imputation method). However, this method may not always be appropriate.

Consider the responses on the EBEP items of the following two hypothetical participants:

- Participant A: 5, 3, NA, NA
- Participant B: NA, NA, NA, 4

Q6

Q6: What are the sum scores for the two participants? What is the problem of sum scores in the presence of missing items?

Q7: What are the mean scores for the two participants? What is a potential problem of mean scores in the presence of missing items?