2_descriptives

Milena Costa

2025-11-04

Helper functions

```
# Color palettes consistent across plots

pal_target <- c(
"self"      = "#E15759",
"climate"   = "#59A14F",
"prosocial" = "#4E79A7"
)

pal_group <- c(
"control"       = "#4E79A7",
"positive_norm" = "#59A14F",
"negative_norm" = "#E15759"
)



pd <- position_dodge(width = 0.7)


group_labels <- c(
  "control"       = "Control Group",
  "positive_norm" = "Positive Norm",
  "negative_norm" = "Negative Norm"
)


# Function to extract key fit indices
fit_table <- function(fit) {
  data.frame(
    ChiSq = fitMeasures(fit, "chisq"),
    df    = fitMeasures(fit, "df"),
    CFI   = fitMeasures(fit, "cfi"),
    TLI   = fitMeasures(fit, "tli"),
    RMSEA = fitMeasures(fit, "rmsea"),
    SRMR  = fitMeasures(fit, "srmr")
  )
}
```

Descriptive Statistics

1. Sample Characteristics

Currently we do not have demographic information, thus the r chunk below does not run. Should any of this information be available in the future, the following entries should be removed from the r chunk below.

```
include = FALSE, eval = FALSE
```

2. Primary Outcome: Choice Behaviour

```r
# Overall choice rates
df_long  |>
  summarise(
    mean_choice = mean(choice, na.rm = TRUE),
    sd_choice = sd(choice, na.rm = TRUE)
  )
```

```
## # A tibble: 1 x 2
##   mean_choice sd_choice
##         <dbl>     <dbl>
## 1       0.924     0.266
```

```r
# By target type
df_long |>
  group_by(target) |>
  summarise(
    mean_choice = mean(choice, na.rm = TRUE),
    sd_choice = sd(choice, na.rm = TRUE),
    n_trials = n()
  )
```

```
## # A tibble: 3 x 4
##   target    mean_choice sd_choice n_trials
##   <fct>           <dbl>     <dbl>    <int>
## 1 self            0.933     0.251     1968
## 2 climate         0.922     0.269     1968
## 3 prosocial       0.916     0.277     1968
```

```r
# By group and block (most important!)
df_long |>
  group_by(group, block) |>
  summarise(
    mean_choice = mean(choice, na.rm = TRUE),
    sd_choice = sd(choice, na.rm = TRUE),
    n_trials = n()
  )
```

```
## # A tibble: 6 x 5
## # Groups:   group [3]
```

```
##   group         block mean_choice sd_choice n_trials
##   <fct>         <fct>       <dbl>     <dbl>    <int>
## 1 control       pre         0.898     0.303     1008
## 2 control       post        0.929     0.257     1008
## 3 positive_norm pre         0.907     0.290      990
## 4 positive_norm post        0.944     0.229      990
## 5 negative_norm pre         0.923     0.266      954
## 6 negative_norm post        0.941     0.235      954
```

```r
# By all three factors (for comprehensive table)
df_long |>
  group_by(target, group, block) |>
  summarise(
    M = mean(choice, na.rm = TRUE),
    SD = sd(choice, na.rm = TRUE)
  ) |>
  arrange(target, group, block)
```

```
## # A tibble: 18 x 5
## # Groups:   target, group [9]
##    target    group         block     M    SD
##    <fct>     <fct>         <fct> <dbl> <dbl>
##  1 self      control       pre   0.896 0.306
##  2 self      control       post  0.934 0.248
##  3 self      positive_norm pre   0.927 0.260
##  4 self      positive_norm post  0.975 0.155
##  5 self      negative_norm pre   0.928 0.259
##  6 self      negative_norm post  0.937 0.243
##  7 climate   control       pre   0.893 0.310
##  8 climate   control       post  0.924 0.265
##  9 climate   positive_norm pre   0.918 0.275
## 10 climate   positive_norm post  0.926 0.262
## 11 climate   negative_norm pre   0.925 0.265
## 12 climate   negative_norm post  0.947 0.225
## 13 prosocial control       pre   0.905 0.294
## 14 prosocial control       post  0.928 0.259
## 15 prosocial positive_norm pre   0.876 0.330
## 16 prosocial positive_norm post  0.932 0.252
## 17 prosocial negative_norm pre   0.918 0.274
## 18 prosocial negative_norm post  0.940 0.237
```

3. Task Controls: Reward and Effort

```r
# Choice rates by reward level
df_long |>
  group_by(reward) |>
  summarise(
    mean_choice = mean(choice, na.rm = TRUE),
    sd_choice = sd(choice, na.rm = TRUE)
  )
```

```
## # A tibble: 3 x 3
```

```
##   reward    mean_choice sd_choice
##   <fct>           <dbl>     <dbl>
## 1 2 points        0.861     0.346
## 2 6 points        0.95      0.218
## 3 10 points       0.960     0.197
```

```r
# Choice rates by effort level
df_long |>
  group_by(effort) |>
  summarise(
    mean_choice = mean(choice, na.rm = TRUE),
    sd_choice = sd(choice, na.rm = TRUE)
  )
```

```
## # A tibble: 2 x 3
##   effort mean_choice sd_choice
##   <fct>        <dbl>     <dbl>
## 1 40%          0.959     0.199
## 2 90%          0.889     0.315
```

```r
# Reward × Effort interaction pattern
df_long |>
  group_by(reward, effort) |>
  summarise(
    mean_choice = mean(choice, na.rm = TRUE),
    sd_choice = sd(choice, na.rm = TRUE)
  )
```

```
## # A tibble: 6 x 4
## # Groups:   reward [3]
##   reward    effort mean_choice sd_choice
##   <fct>     <fct>        <dbl>     <dbl>
## 1 2 points  40%          0.935     0.247
## 2 2 points  90%          0.787     0.409
## 3 6 points  40%          0.966     0.180
## 4 6 points  90%          0.934     0.249
## 5 10 points 40%          0.974     0.158
## 6 10 points 90%          0.945     0.229
```

4. Moderator Variables

```r
sus_scores <- df_wide |>
  select(starts_with("SUS_")) |>
  mutate(across(everything(), as.numeric))

sus_scores
```

```
##     SUS_1 SUS_2 SUS_3 SUS_4 SUS_5 SUS_6 SUS_7 SUS_8
## 1       1     3     1     5     1     3     3     1
## 2       1     4     1     5     1     2     1     1
```

```
## 3     3   5   3   5   3   5   5   3
## 4     1   3   2   4   2   2   1   1
## 5     3   3   2   4   2   3   3   4
## 6     3   2   3   5   5   4   5   3
## 7     3   4   4   5   3   4   3   1
## 8     1   5   2   3   1   2   2   1
## 9     2   4   4   4   3   4   5   1
## 10   NA  NA  NA  NA  NA  NA  NA  NA
## 11    3   2   2   4   3   3   4   2
## 12   NA  NA  NA  NA  NA  NA  NA  NA
## 13    2   1   2   3   1   1   2   3
## 14    3   2   3   3   3   3   2   2
## 15    2   2   1   2   2   3   2   1
## 16    1   2   1   3   2   2   1   1
## 17   NA  NA  NA  NA  NA  NA  NA  NA
## 18    1   1   1   1   1   1   1   1
## 19    2   2   1   4   2   3   2   2
## 20    1   1   1   1   1   1   1   1
## 21    3   3   3   3   3   3   3   3
## 22    3   2   1   4   4   3   4   2
## 23   NA  NA  NA  NA  NA  NA  NA  NA
## 24    3   3   2   3   4   2   3   3
## 25    1   3   2   3   2   2   3   4
## 26    3   3   3   4   3   3   2   4
## 27    2   4   2   4   1   3   1   1
## 28    3   4   2   3   1   2   1   3
## 29    3   4   1   3   1   2   5   1
## 30    3   3   3   3   2   3   4   3
## 31    2   2   2   5   2   3   2   2
## 32    2   4   2   3   2   3   3   2
## 33    5   5   4   5   1   3   5   1
## 34    3   2   2   2   2   2   1   1
## 35    3   4   2   1   2   3   3   2
## 36    4   5   5   2   5   5   4   3
## 37    2   2   2   2   2   2   2   2
## 38    2   3   1   3   1   3   1   1
## 39    1   1   1   3   1   1   3   1
## 40    3   3   3   3   3   3   3   3
## 41    2   2   2   4   2   2   3   2
## 42    3   3   3   4   2   4   3   3
## 43    2   3   2   4   2   4   1   3
## 44    2   4   1   4   1   1   1   1
## 45    3   3   3   4   3   5   3   3
## 46    5   1   5   5   5   5   5   5
## 47    2   1   3   5   3   5   1   2
## 48    2   2   1   2   1   1   3   1
## 49    1   1   1   4   1   1   1   2
## 50    4   3   4   3   4   3   2   2
## 51    1   1   1   1   1   1   1   1
## 52    1   3   1   4   1   1   2   1
## 53    1   3   1   3   1   1   1   1
## 54    1   1   1   3   1   1   1   1
## 55    2   2   2   4   2   2   2   1
## 56    1   1   1   1   1   1   1   1
```

```
## 57      1    1    1    1    1    1    1    1
## 58      1    3    2    3    2    2    3    4
## 59      1    2    2    3    1    1    1    2
## 60      2    2    2    3    3    2    3    2
## 61      2    4    4    3    4    3    4    3
## 62      2    1    2    1    1    2    2    1
## 63      2    2    2    4    2    2    1    1
## 64      3    4    3    4    3    3    4    2
## 65      3    3    3    4    4    5    5    5
## 66      4    2    4    3    4    4    4    3
## 67      1    5    1    4    1    2    1    1
## 68     NA   NA   NA   NA   NA   NA   NA   NA
## 69      2    3    2    4    2    2    2    1
## 70      2    1    1    1    3    3    2    2
## 71      2    3    5    4    1    1    5    5
## 72      2    3    2    2    3    2    2    1
## 73      3    2    3    4    3    2    3    1
## 74      4    3    3    5    3    2    2    1
## 75      3    2    2    2    1    2    2    1
## 76      4    3    5    5    5    4    5    1
## 77      5    3    5    4    5    4    5    2
## 78      5    5    3    5    4    2    3    3
## 79      3    3    1    3    1    1    1    1
## 80      2    1    2    2    1    1    3    1
## 81      2    2    2    2    1    2    2    1
## 82      3    3    4    4    4    5    5    3
## 83      2    3    5    4    5    3    5    4
## 84      3    4    4    5    4    5    4    4
## 85      3    3    4    3    4    3    3    3
## 86      2    3    2    3    2    1    2    1
## 87      2    4    2    4    2    4    4    3
## 88      2    4    1    3    1    2    1    2
## 89      2    2    3    4    4    3    4    3
## 90      3    2    1    4    2    2    4    2
## 91      3    2    3    2    4    5    3    4
## 92     NA   NA   NA   NA   NA   NA   NA   NA
## 93      3    3    3    3    3    3    3    1
## 94      2    3    2    3    2    4    3    2
## 95      2    3    2    4    3    1    2    2
## 96      1    1    1    1    1    1    2    1
## 97      1    1    1    1    1    1    1    1
## 98      3    4    3    4    2    2    3    1
## 99      1    1    2    2    4    3    2    3
## 100     3    4    3    4    3    2    2    2
## 101     3    4    3    5    3    3    4    3
## 102     3    3    2    3    3    3    3    3
## 103     2    3    2    4    2    2    2    1
## 104     2    3    2    1    2    3    1    1
## 105     2    5    4    5    5    4    5    1
## 106     5    4    5    3    5    5    5    5
## 107     2    2    2    3    3    2    3    2
## 108     1    1    1    1    1    1    1    1
## 109     2    2    1    1    2    3    2    1
## 110     3    2    1    1    2    1    1    1
```

```
## 111     2    3    2    4    2    2    1    1
## 112     2    3    2    3    1    3    1    1
## 113     1    4    1    5    1    3    1    1
## 114     1    3    1    4    2    2    1    1
## 115     4    4    4    4    4    4    4    2
## 116     2    2    1    2    2    2    3    1
## 117     4    4    4    4    4    4    3    3
## 118     2    1    1    3    2    2    1    1
## 119     3    3    3    3    3    3    3    3
## 120     2    2    2    3    2    3    2    4
## 121     4    3    3    3    3    2    3    2
## 122     2    2    4    3    4    3    3    3
## 123     2    2    2    4    3    4    3    4
## 124     2    2    3    3    4    4    4    3
## 125     2    3    2    1    1    2    3    2
## 126     4    3    5    3    5    4    5    5
## 127     1    3    3    5    4    3    4    3
## 128     1    1    1    1    1    1    1    1
## 129     1    3    2    5    1    1    3    1
## 130     2    1    1    2    2    2    2    1
## 131     3    4    3    3    3    2    3    3
## 132     1    3    1    3    2    1    4    4
## 133     1    5    4    4    3    3    3    4
## 134     3    3    4    3    4    4    3    4
## 135     3    3    3    4    4    1    1    3
## 136     3    3    1    4    1    1    1    1
## 137     3    4    2    5    4    2    4    3
## 138     1    2    1    2    1    1    2    1
## 139     3    4    3    3    3    2    4    3
## 140     2    3    1    5    1    3    2    3
## 141     3    3    1    1    2    2    1    1
## 142     5    5    4    5    4    5    5    3
## 143     2    4    3    5    3    5    3    5
## 144     3    5    1    5    1    3    2    1
## 145     3    3    4    4    3    3    4    3
## 146     4    5    1    3    1    2    4    5
## 147     2    1    1    1    1    1    1    1
## 148     5    4    2    4    3    5    4    3
## 149     3    5    3    1    3    3    3    1
## 150     2    3    2    2    2    2    3    3
## 151     2    2    2    4    2    2    2    5
## 152     2    2    1    3    2    1    1    2
## 153     2    4    3    4    3    3    3    3
## 154     2    3    3    4    2    2    1    2
## 155     4    3    3    3    3    3    3    3
## 156     2    3    2    3    2    3    2    2
## 157     2    2    1    4    2    3    3    1
## 158     4    3    4    3    4    2    5    3
## 159     1    2    2    4    1    2    1    1
## 160     1    1    1    1    1    1    1    1
## 161     2    2    1    4    1    1    1    2
## 162     4    3    2    5    3    4    2    3
## 163     1    1    1    1    1    1    1    1
## 164     4    3    3    5    4    3    3    2
```

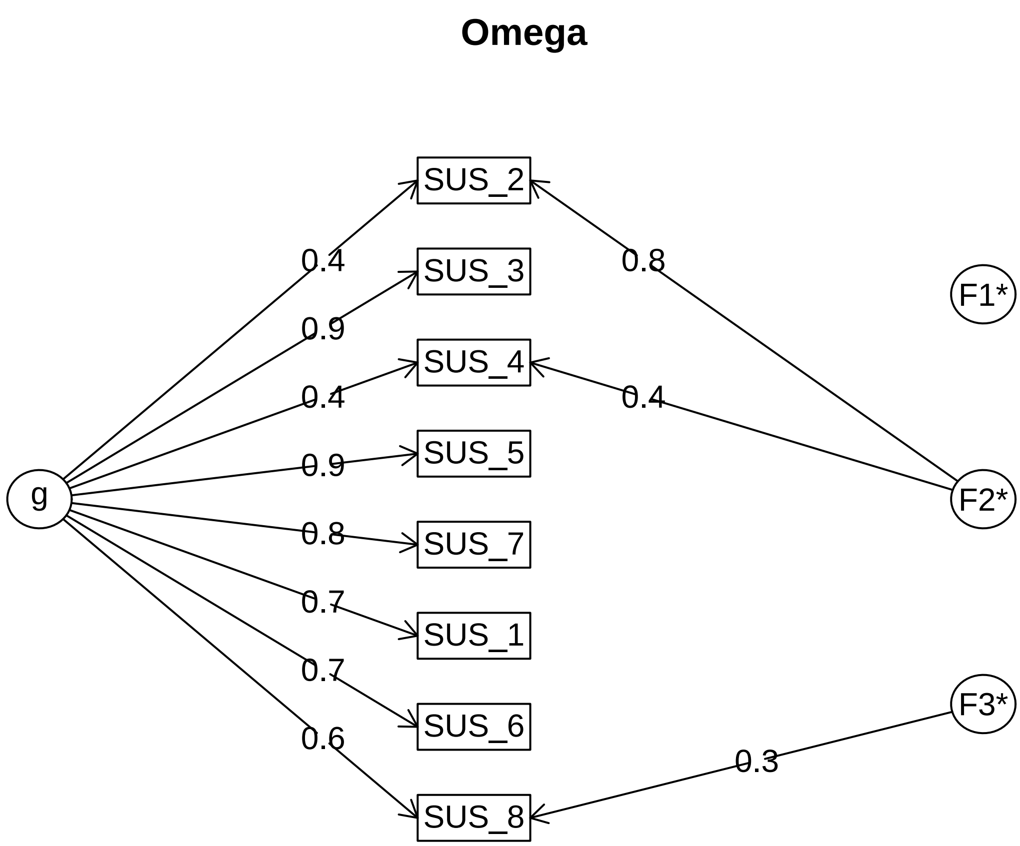
```r
str(sus_scores)
```

```
## 'data.frame':    164 obs. of  8 variables:
##  $ SUS_1: num  1 1 3 1 3 3 3 1 2 NA ...
##  $ SUS_2: num  3 4 5 3 3 2 4 5 4 NA ...
##  $ SUS_3: num  1 1 3 2 2 3 4 2 4 NA ...
##  $ SUS_4: num  5 5 5 4 4 5 5 3 4 NA ...
##  $ SUS_5: num  1 1 3 2 2 5 3 1 3 NA ...
##  $ SUS_6: num  3 2 5 2 3 4 4 2 4 NA ...
##  $ SUS_7: num  3 1 5 1 3 5 3 2 5 NA ...
##  $ SUS_8: num  1 1 3 1 4 3 1 1 1 NA ...
```

```r
# Reliability
psych::alpha(sus_scores, use = "pairwise")$total# alpha
```

```
##  raw_alpha std.alpha   G6(smc) average_r      S/N        ase     mean        sd
##  0.8745036 0.8745255 0.8795407 0.4655888 6.969746 0.01469615 2.548259 0.8676012
##   median_r
##  0.4729983
```

```r
psych::omega(sus_scores, use = "pairwise")$omega.tot # omega total
```




```
## [1] 0.9086112
```

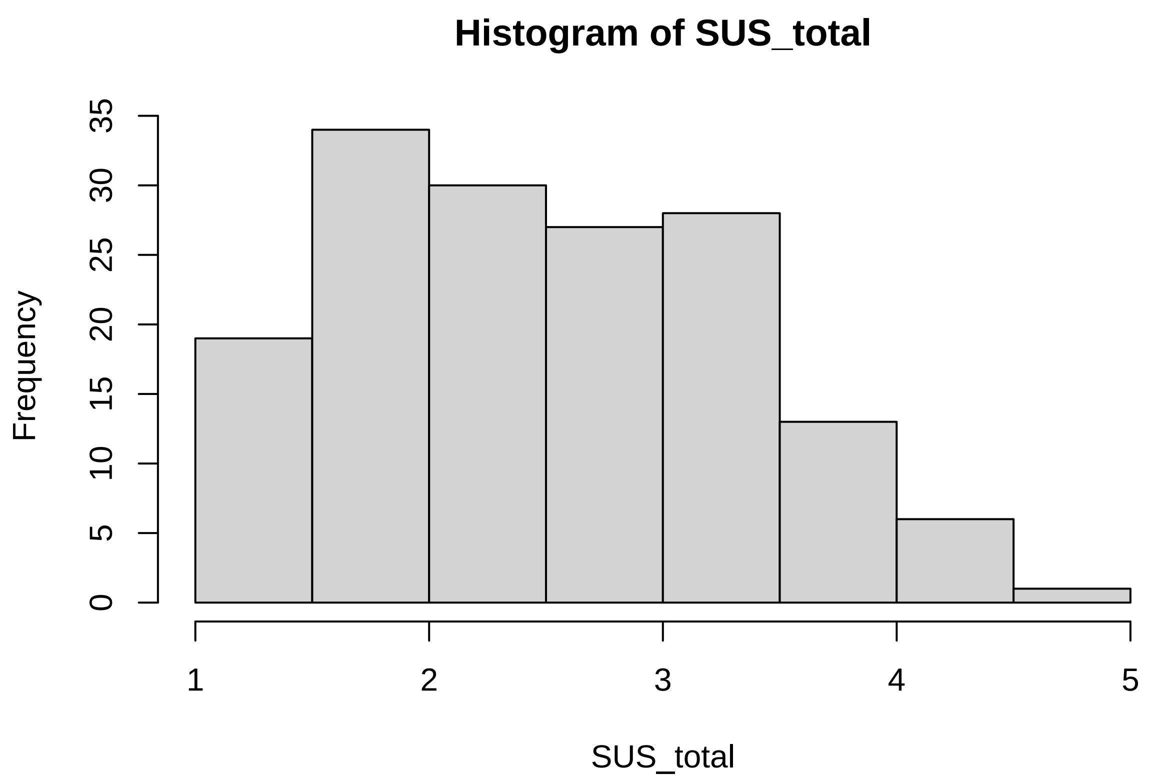
```r
# Total score

SUS_total <- rowMeans(sus_scores, na.rm=TRUE)

describe(SUS_total) # descriptives for SUS
```

```
##    vars   n mean   sd median trimmed  mad min  max range skew kurtosis   se
## X1    1 158 2.55 0.87    2.5    2.54 0.93   1 4.62  3.62 0.14    -0.67 0.07
```

```r
hist(SUS_total)
```



Confirmatory Factor Analysis

One factor model

```r
# Single factor model


model <- '
 SUS =~ SUS_1 + SUS_2 + SUS_3 + SUS_4 + SUS_5 + SUS_6 + SUS_7 + SUS_8
'
```

```r
fit1 <- cfa(
  model,
  data = sus_scores,
  std.lv = TRUE,
  missing = "fiml"
)
```

```r
summary(fit1, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-20 ended normally after 17 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        24
##
##                                                   Used       Total
##   Number of observations                           158         164
##   Number of missing patterns                         1
##
## Model Test User Model:
##
##   Test statistic                                51.300
##   Degrees of freedom                                20
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                               609.286
##   Degrees of freedom                                28
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.946
##   Tucker-Lewis Index (TLI)                       0.925
##
##   Robust Comparative Fit Index (CFI)             0.946
##   Robust Tucker-Lewis Index (TLI)                0.925
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1725.164
##   Loglikelihood unrestricted model (H1)      -1699.514
##
##   Akaike (AIC)                                3498.328
##   Bayesian (BIC)                              3571.831
##   Sample-size adjusted Bayesian (SABIC)       3495.859
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.100
```

```
##   90 Percent confidence interval - lower         0.066
##   90 Percent confidence interval - upper         0.134
##   P-value H_0: RMSEA <= 0.050                    0.009
##   P-value H_0: RMSEA >= 0.080                    0.845
##
##   Robust RMSEA                                   0.100
##   90 Percent confidence interval - lower         0.066
##   90 Percent confidence interval - upper         0.134
##   P-value H_0: Robust RMSEA <= 0.050             0.009
##   P-value H_0: Robust RMSEA >= 0.080             0.845
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.053
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SUS =~
##     SUS_1             0.727    0.076    9.585    0.000    0.727    0.691
##     SUS_2             0.498    0.089    5.566    0.000    0.498    0.442
##     SUS_3             1.009    0.076   13.341    0.000    1.009    0.867
##     SUS_4             0.544    0.097    5.616    0.000    0.544    0.444
##     SUS_5             1.040    0.080   12.963    0.000    1.040    0.851
##     SUS_6             0.886    0.083   10.688    0.000    0.886    0.749
##     SUS_7             1.007    0.090   11.202    0.000    1.007    0.773
##     SUS_8             0.726    0.089    8.159    0.000    0.726    0.610
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SUS_1             2.380    0.084   28.405    0.000    2.380    2.260
##    .SUS_2             2.785    0.090   31.060    0.000    2.785    2.471
##    .SUS_3             2.285    0.093   24.667    0.000    2.285    1.962
##    .SUS_4             3.266    0.097   33.529    0.000    3.266    2.667
##    .SUS_5             2.392    0.097   24.623    0.000    2.392    1.959
##    .SUS_6             2.551    0.094   27.103    0.000    2.551    2.156
##    .SUS_7             2.595    0.104   25.041    0.000    2.595    1.992
##    .SUS_8             2.133    0.095   22.506    0.000    2.133    1.791
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SUS_1             0.580    0.072    8.091    0.000    0.580    0.523
##    .SUS_2             1.022    0.118    8.648    0.000    1.022    0.805
##    .SUS_3             0.337    0.054    6.214    0.000    0.337    0.248
##    .SUS_4             1.203    0.139    8.663    0.000    1.203    0.803
##    .SUS_5             0.410    0.063    6.547    0.000    0.410    0.275
##    .SUS_6             0.615    0.080    7.719    0.000    0.615    0.439
##    .SUS_7             0.682    0.090    7.560    0.000    0.682    0.402
##    .SUS_8             0.892    0.107    8.370    0.000    0.892    0.628
```

```
##     SUS                 1.000                                     1.000    1.000
```

Two factor model

```r
# Two factor model

model2 <- '
 Anxiety =~ SUS_1 + SUS_3 + SUS_5 + SUS_7
 Selfesteem =~ SUS_2 + SUS_4 + SUS_6 + SUS_8
'


fit2 <- cfa(
  model2,
  data = sus_scores,
  std.lv = TRUE,
  missing = "fiml"
)



summary(fit2, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-20 ended normally after 20 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        25
##
##                                                   Used       Total
##   Number of observations                           158         164
##   Number of missing patterns                         1
##
## Model Test User Model:
##
##   Test statistic                                45.589
##   Degrees of freedom                                19
##   P-value (Chi-square)                           0.001
##
## Model Test Baseline Model:
##
##   Test statistic                               609.286
##   Degrees of freedom                                28
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.954
##   Tucker-Lewis Index (TLI)                       0.933
##
##   Robust Comparative Fit Index (CFI)             0.954
##   Robust Tucker-Lewis Index (TLI)                0.933
```

```
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1722.309
##   Loglikelihood unrestricted model (H1)      -1699.514
##
##   Akaike (AIC)                                3494.617
##   Bayesian (BIC)                              3571.182
##   Sample-size adjusted Bayesian (SABIC)       3492.045
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.094
##   90 Percent confidence interval - lower         0.059
##   90 Percent confidence interval - upper         0.129
##   P-value H_0: RMSEA <= 0.050                    0.021
##   P-value H_0: RMSEA >= 0.080                    0.768
##
##   Robust RMSEA                                   0.094
##   90 Percent confidence interval - lower         0.059
##   90 Percent confidence interval - upper         0.129
##   P-value H_0: Robust RMSEA <= 0.050             0.021
##   P-value H_0: Robust RMSEA >= 0.080             0.768
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.049
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Observed
##   Observed information based on                Hessian
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Anxiety =~
##     SUS_1             0.728    0.076    9.588    0.000    0.728    0.692
##     SUS_3             1.018    0.075   13.495    0.000    1.018    0.874
##     SUS_5             1.049    0.080   13.108    0.000    1.049    0.859
##     SUS_7             1.003    0.090   11.113    0.000    1.003    0.770
##   Selfesteem =~
##     SUS_2             0.537    0.092    5.845    0.000    0.537    0.477
##     SUS_4             0.601    0.100    6.025    0.000    0.601    0.491
##     SUS_6             0.945    0.086   11.045    0.000    0.945    0.799
##     SUS_8             0.744    0.091    8.133    0.000    0.744    0.625
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Anxiety ~~
##     Selfesteem        0.914    0.039   23.326    0.000    0.914    0.914
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
```

```
##    .SUS_1             2.380    0.084   28.405    0.000    2.380    2.260
##    .SUS_3             2.285    0.093   24.667    0.000    2.285    1.962
##    .SUS_5             2.392    0.097   24.623    0.000    2.392    1.959
##    .SUS_7             2.595    0.104   25.041    0.000    2.595    1.992
##    .SUS_2             2.785    0.090   31.060    0.000    2.785    2.471
##    .SUS_4             3.266    0.097   33.529    0.000    3.266    2.667
##    .SUS_6             2.551    0.094   27.103    0.000    2.551    2.156
##    .SUS_8             2.133    0.095   22.506    0.000    2.133    1.791
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .SUS_1             0.578    0.072    8.057    0.000    0.578    0.521
##    .SUS_3             0.319    0.054    5.904    0.000    0.319    0.235
##    .SUS_5             0.392    0.062    6.292    0.000    0.392    0.263
##    .SUS_7             0.690    0.092    7.521    0.000    0.690    0.407
##    .SUS_2             0.981    0.117    8.378    0.000    0.981    0.773
##    .SUS_4             1.137    0.137    8.312    0.000    1.137    0.759
##    .SUS_6             0.506    0.089    5.711    0.000    0.506    0.362
##    .SUS_8             0.866    0.110    7.901    0.000    0.866    0.610
##     Anxiety           1.000                               1.000    1.000
##     Selfesteem        1.000                               1.000    1.000
```

Fit Measures

```r
fit_table(fit1)
```

```
##           ChiSq df       CFI       TLI      RMSEA       SRMR
## chisq 51.30013 20 0.9461536 0.9246151 0.09952439 0.05298635
```

```r
fit_table(fit2)
```

```
##           ChiSq df       CFI       TLI      RMSEA       SRMR
## chisq 45.58885 19 0.9542586 0.9325916 0.09411187 0.04908875
```

5. Ceiling Effects

```r
# By participant overall
ceiling_participants <- df_long |>
  group_by(ppn) |>
  summarise(
    prop_high_effort = mean(choice, na.rm = TRUE)
  ) |>
  summarise(
    at_90 = sum(prop_high_effort >= 0.90),
    pct_90 = mean(prop_high_effort >= 0.90) * 100,
    at_95 = sum(prop_high_effort >= 0.95),
    pct_95 = mean(prop_high_effort >= 0.95) * 100,
    at_100 = sum(prop_high_effort == 1.00),
    pct_100 = mean(prop_high_effort == 1.00) * 100
```

```r
  )

print(ceiling_participants)
```

```
## # A tibble: 1 x 6
##   at_90 pct_90 at_95 pct_95 at_100 pct_100
##   <int>  <dbl> <int>  <dbl>  <int>   <dbl>
## 1   118   72.0    84   51.2     52    31.7
```

```r
# By participant and block
ceiling_by_block <- df_long |>
  group_by(ppn, block) |>
  summarise(
    prop_high_effort = mean(choice, na.rm = TRUE),
    .groups = "drop"
  ) |>
  group_by(block) |>
  summarise(
    n_at_90 = sum(prop_high_effort >= 0.90),
    pct_at_90 = mean(prop_high_effort >= 0.90) * 100,
    n_at_95 = sum(prop_high_effort >= 0.95),
    pct_at_95 = mean(prop_high_effort >= 0.95) * 100,
    n_at_100 = sum(prop_high_effort == 1.00),
    pct_at_100 = mean(prop_high_effort == 1.00) * 100
  )

print(ceiling_by_block)
```

```
## # A tibble: 2 x 7
##   block n_at_90 pct_at_90 n_at_95 pct_at_95 n_at_100 pct_at_100
##   <fct>   <int>     <dbl>   <int>     <dbl>    <int>      <dbl>
## 1 pre       100      61.0      59      36.0       59       36.0
## 2 post       NA      NA        NA      NA         NA       NA
```

```r
# Calculate proportion of high-effort choices per participant
participant_props <- df_long |>
  group_by(ppn) |>
  summarise(
    prop_high_effort = mean(choice, na.rm = TRUE)
  )

# Descriptives
summary(participant_props$prop_high_effort)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4444  0.8889  0.9722  0.9238  1.0000  1.0000
```

```r
quantile(participant_props$prop_high_effort, probs = c(0.25, 0.5, 0.75))
```

```
##       25%       50%       75%
## 0.8888889 0.9722222 1.0000000
```

```r
# How many chose high effort 100% of the time?
sum(participant_props$prop_high_effort == 1.0)
```

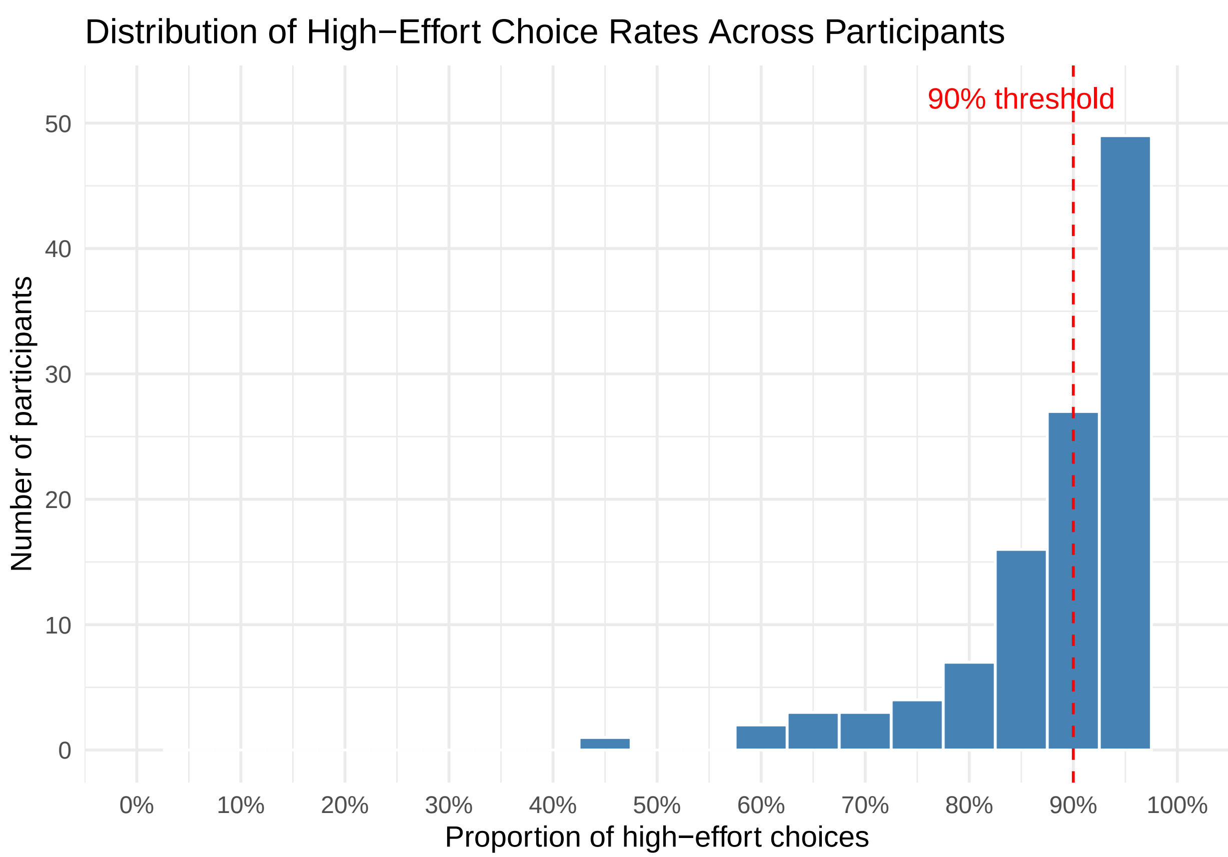
```
## [1] 52
```

```r
mean(participant_props$prop_high_effort == 1.0) * 100
```

```
## [1] 31.70732
```

```r
# Histogram

ggplot(participant_props, aes(x = prop_high_effort)) +
  geom_histogram(binwidth = 0.05, fill = "steelblue", color = "white") +
  scale_x_continuous(limits = c(0, 1),
                     breaks = seq(0, 1, 0.1),
                     labels = scales::percent) +
  labs(x = "Proportion of high-effort choices",
       y = "Number of participants",
       title = "Distribution of High-Effort Choice Rates Across Participants") +
  theme_minimal() +
  geom_vline(xintercept = 0.90, linetype = "dashed", color = "red") +
  annotate("text", x = 0.85, y = max(table(cut(participant_props$prop_high_effort, breaks = 20))),
           label = "90% threshold", color = "red")
```



```r
# Standard deviation of choices (within-person)
within_person_variability <- df_long|>
  group_by(ppn)|>
  summarise(
    sd_choice = sd(choice, na.rm = TRUE),
    var_choice = var(choice, na.rm = TRUE)
  )|>
  summarise(
    mean_sd = mean(sd_choice, na.rm = TRUE),
    median_sd = median(sd_choice, na.rm = TRUE),
    n_zero_var = sum(sd_choice == 0, na.rm = TRUE),
    pct_zero_var = mean(sd_choice == 0, na.rm = TRUE) * 100
  )

print(within_person_variability)
```

```
## # A tibble: 1 x 4
##   mean_sd median_sd n_zero_var pct_zero_var
##     <dbl>     <dbl>      <int>        <dbl>
## 1   0.195     0.167         52         31.7
```

```r
# Ceiling by experimental conditions
ceiling_by_group <- df_long|>
  group_by(ppn, group)|>
  summarise(
    prop_high_effort = mean(choice, na.rm = TRUE),
    .groups = "drop"
  )|>
  group_by(group)|>
  summarise(
    n_at_90 = sum(prop_high_effort >= 0.90),
    pct_at_90 = mean(prop_high_effort >= 0.90) * 100,
    mean_prop = mean(prop_high_effort),
    sd_prop = sd(prop_high_effort),
    median_prop = median(prop_high_effort)
  )

print(ceiling_by_group)
```

```
## # A tibble: 3 x 6
##   group         n_at_90 pct_at_90 mean_prop sd_prop median_prop
##   <fct>           <int>     <dbl>     <dbl>   <dbl>       <dbl>
## 1 control            36      64.3     0.913  0.107        0.972
## 2 positive_norm      43      78.2     0.926  0.0990       0.972
## 3 negative_norm      39      73.6     0.932  0.0840       0.972
```

```r
# table
ceiling_summary <- df_long |>
  group_by(ppn)|>
  summarise(
    prop = mean(choice, na.rm = TRUE),
    sd = sd(choice, na.rm = TRUE)
```

```
  )|>
  summarise(
    mean = mean(prop),
    sd = sd(prop),
    median = median(prop),
    q25 = quantile(prop, 0.25),
    q75 = quantile(prop, 0.75),
    min = min(prop),
    max = max(prop),
    n_90 = sum(prop >= 0.90),
    pct_90 = mean(prop >= 0.90) * 100,
    n_95 = sum(prop >= 0.95),
    pct_95 = mean(prop >= 0.95) * 100,
    n_100 = sum(prop == 1.00),
    pct_100 = mean(prop == 1.00) * 100,
    mean_within_sd = mean(sd, na.rm = TRUE),
    n_zero_var = sum(sd == 0, na.rm = TRUE),
    pct_zero_var = mean(sd == 0, na.rm = TRUE) * 100
  )

print(ceiling_summary)
```

```
## # A tibble: 1 x 16
##    mean     sd median   q25   q75   min   max  n_90 pct_90  n_95 pct_95 n_100
##   <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <int>  <dbl> <int>  <dbl> <int>
## 1 0.924 0.0970  0.972 0.889     1 0.444     1   118   72.0    84   51.2    52
## # i 4 more variables: pct_100 <dbl>, mean_within_sd <dbl>, n_zero_var <int>,
## #   pct_zero_var <dbl>
```

5. Missing Data Summary

1) Helper functions

```
# Color palettes consistent across plots

pal_target <- c(
"self"      = "#E15759",
"climate"   = "#59A14F",
"prosocial" = "#4E79A7"
)

pal_group <- c(
"control"       = "#4E79A7",
"positive_norm" = "#59A14F",
"negative_norm" = "#E15759"
)



pd <- position_dodge(width = 0.7)
```

```r
group_labels <- c(
  "control"       = "Control Group",
  "positive_norm" = "Positive Norm",
  "negative_norm" = "Negative Norm"
)
```

Susceptibility

```r
susceptibility_descriptives <- df_long |>
summarise(
M   = mean(susceptibility, na.rm = TRUE),
SD  = sd(susceptibility, na.rm = TRUE),
Min = min(susceptibility, na.rm = TRUE),
Max = max(susceptibility, na.rm = TRUE)
)

susceptibility_descriptives
```

```
## # A tibble: 1 x 4
##       M    SD   Min   Max
##   <dbl> <dbl> <dbl> <dbl>
## 1  2.54 0.852     1  4.62
```

Visualizations

a) Reward × Effort interaction

```r
plot_a_data <- df_long |>
  group_by(reward, effort) |>
  summarise(p = mean(choice == 1, na.rm = TRUE), .groups = "drop")


reward_effort_plot <- plot_a_data |>
  ggplot(aes(x = reward, y = p, color = effort, group = effort)) +
  geom_point(size = 3) +
  geom_line(size = 1) +
  scale_color_manual(values = c(
    "40%" = "#E15759",
    "90%" = "#4E79A7"
  )) +
  scale_y_continuous(limits = c(0, 1)) +
  labs(
    x = "Reward",
    y = "Proportion of High-Effort Choices",
    color = "Effort"
  ) +
  theme_bw()
```

```r
ggsave("figures/descriptives/reward_effort_plot.png",
       plot = reward_effort_plot,
       width = 7, height = 5, dpi = 300)
```

b) Target × Block × Group

```r
plot_b_data <- df_long |>
  group_by(group, block, target) |>
  summarise(
    p_choice = mean(choice == 1, na.rm = TRUE),
    n        = sum(!is.na(choice)),
    .groups = "drop"
  )


target_block_group_plot <- plot_b_data |>
  ggplot(aes(x = block,
             y = p_choice,
             color = target,
             group = target)) +
  geom_point(size = 3) +
  geom_line(size = 1) +
  facet_wrap(~ group, labeller = labeller(group = group_labels)) +
  labs(
    x     = "Block",
    y     = "Proportion of High-Effort Choices",
    color = "Target"
  ) +
  scale_y_continuous(limits = c(0, 1)) +
  scale_color_manual(values = pal_target) +
  theme_bw() +
  theme(
    legend.position = "top",
    legend.title    = element_blank()
  )

ggsave("figures/descriptives/target_block_group_plot.png",
       plot = target_block_group_plot,
       width = 7, height = 5, dpi = 300)
```