Supplementary Material - On the Trait Like Properties of Emotion Regulation Dynamics

Contents

List of Tables

```
knitr::write_bib(c(.packages(), "bookdown"), "packages.bib")

wkspc = sessionInfo()
packages = wkspc$otherPkgs
package.df = tibble(package = names(packages))
package.df$list = packages
package.df2 = package.df
```

All data cleaning and analyses were completed using R version 4.1.2 (2021-11-01) (Bird Hippie). Table S1 lists the packages (and versions) used in these analyses.

Table S1: Packages (and their versions) used for theses analyses.

Package	Version	Authors and contributors
apa	0.3.3	Daniel Gromer [aut, cre]
arsenal	3.6.3	Ethan Heinzen [aut, cre], Jason Sinnwell [aut], Elizabeth Atkinson [aut], Tina Gunderson [aut], Gregory Dougherty [aut], Patrick Votruba [ctb], Ryan Lennon [ctb], Andrew Hanson [ctb], Krista Goergen [ctb], Emily Lundt [ctb], Brendan Broderick [ctb], Maddie McCullough [art]
colorspace	2.1-0	Ross Ihaka [aut], Paul Murrell [aut] (<https://orcid.org/0000-0002-3224-8858>), Kurt Hornik [aut] (<https://orcid.org/0000-0003-4198-9911>), Jason C. Fisher [aut] (<https://orcid.org/0000-0001-9032-8912>), Reto Stauffer [aut] (<https://orcid.org/0000-0002-3798-5507>), Claus O. Wilke [aut] (<https://orcid.org/0000-0002-7470-9261>), Claire D. McWhite [aut] (<https://orcid.org/0000-0001-7346-3047>), Achim Zeileis [aut, cre] (<https://orcid.org/0000-0003-0918-3766>)
papaja	0.1.0.9999	Frederik Aust [aut, cre] (<https://orcid.org/0000-0003-4900-788X>), Marius Barth [aut] (<https://orcid.org/0000-0002-3421-6665>), Birk Diedenhofen [ctb], Christoph Stahl [ctb], Joseph V. Casillas [ctb], Rudolf Siegel [ctb]
tinylabels	0.2.3	Marius Barth [aut, cre] (<https://orcid.org/0000-0002-3421-6665>)
correlation	0.8.3	Dominique Makowski [aut, inv] (<https://orcid.org/0000-0001-5375-9967>, @Dom_Makowski), Brenton M. Wiernik [aut, cre] (<https://orcid.org/0000-0001-9560-6336>, @bmwiernik), Indrajeet Patil [aut] (<https://orcid.org/0000-0003-1995-6531>, @patilindrajeets), Daniel Lüdecke [aut] (<https://orcid.org/0000-0002-8895-3206>, @strengejacke), Mattan S. Ben-Shachar [aut] (<https://orcid.org/0000-0002-4287-4801>, @mattansb), Mark White [rev], Maximilian M. Rabe [rev] (<https://orcid.org/0000-0002-2556-5644>)
apaTables	2.0.8	David Stanley [aut, cre]
kableExtra	1.3.4	Hao Zhu [aut, cre] (<https://orcid.org/0000-0002-3386-6076>), Thomas Travison [ctb], Timothy Tsai [ctb], Will Beasley [ctb], Yihui Xie [ctb], GuangChuang Yu [ctb], Stéphane Laurent [ctb], Rob Shepherd [ctb], Yoni Sidi [ctb], Brian Salzer [ctb], George Gui [ctb], Yeliang Fan [ctb], Duncan Murdoch [ctb], Bill Evans [ctb]
xtable	1.8-4	David B. Dahl [aut], David Scott [aut, cre], Charles Roosen [aut], Arni Magnusson [aut], Jonathan Swinton [aut], Ajay Shah [ctb], Arne Henningsen [ctb], Benno Puetz [ctb], Bernhard Pfaff [ctb], Claudio Agostinelli [ctb], Claudius Loehnert [ctb], David Mitchell [ctb], David Whiting [ctb], Fernando da Rosa [ctb], Guido Gay [ctb], Guido Schulz [ctb], Ian Fellows [ctb], Jeff Laake [ctb], John Walker [ctb], Jun Yan [ctb], Liviu Andronic [ctb], Markus Loecher [ctb], Martin Gubri [ctb], Matthieu Stigler [ctb], Robert Castelo [ctb], Seth Falcon [ctb], Stefan Edwards [ctb], Sven Garbade [ctb], Uwe Ligges [ctb]

Table S1: Packages (and their versions) used for theses analyses.
(continued)

Package	Version	Authors and contributors
Hmisc	4.6-0	Frank E Harrell Jr [aut, cre] (<https://orcid.org/0000-0002-8271-5493>), Charles Dupont [ctb] (contributed several functions and maintains latex functions)
Formula	1.2-4	Achim Zeileis [aut, cre] (<https://orcid.org/0000-0003-0918-3766>), Yves Croissant [aut]
survival	3.2-13	Terry M Therneau [aut, cre], Thomas Lumley [ctb, trl] (original S->R port and R maintainer until 2009), Atkinson Elizabeth [ctb], Crowson Cynthia [ctb]
lattice	0.20-45	Deepayan Sarkar [aut, cre] (<https://orcid.org/0000-0003-4107-1553>), Felix Andrews [ctb], Kevin Wright [ctb] (documentation), Neil Klepeis [ctb], Johan Larsson [ctb] (colorkey title), Paul Murrell [ctb]
tidybayes	3.0.2	Matthew Kay [aut, cre], Timothy Mastny [ctb]
brms	2.17.0	Paul-Christian Bürkner [aut, cre], Jonah Gabry [ctb], Sebastian Weber [ctb], Andrew Johnson [ctb], Martin Modrak [ctb], Hamada S. Badr [ctb], Frank Weber [ctb], Mattan S. Ben-Shachar [ctb], Hayden Rabel [ctb]
Rcpp	1.0.8.3	Dirk Eddelbuettel, Romain Francois, JJ Allaire, Kevin Ushey, Qiang Kou, Nathan Russell, Inaki Ucar, Douglas Bates and John Chambers
rstan	2.21.5	Jiqiang Guo [aut], Jonah Gabry [aut], Ben Goodrich [cre, aut], Sebastian Weber [aut], Daniel Lee [ctb], Krzysztof Sakrejda [ctb], Modrak Martin [ctb], Trustees of Columbia University [cph], Oleg Sklyar [cph] (R/cxxfunplus.R), The R Core Team [cph] (R/pairs.R, R/dynGet.R), Jens Oehlschlaegel-Akiyoshi [cph] (R/pairs.R), John Maddock [cph] (gamma.hpp), Paul Bristow [cph] (gamma.hpp), Nikhar Agrawal [cph] (gamma.hpp), Christopher Kormanyos [cph] (gamma.hpp), Bronder Steve [ctb]
StanHeaders	2.21.0-7	Ben Goodrich [cre, aut], Joshua Pritikin [ctb], Andrew Gelman [aut], Bob Carpenter [aut], Matt Hoffman [aut], Daniel Lee [aut], Michael Betancourt [aut], Marcus Brubaker [aut], Jiqiang Guo [aut], Peter Li [aut], Allen Riddell [aut], Marco Inacio [aut], Mitzi Morris [aut], Jeffrey Arnold [aut], Rob Goedman [aut], Brian Lau [aut], Rob Trangucci [aut], Jonah Gabry [aut], Alp Kucukelbir [aut], Robert Grant [aut], Dustin Tran [aut], Michael Malecki [aut], Yuanjun Gao [aut], Trustees of Columbia University [cph], Lawrence Livermore National Security [cph] (CVODES), The Regents of the University of California [cph] (CVODES), Southern Methodist University [cph] (CVODES)
psych	2.1.9	William Revelle [aut, cre] (<https://orcid.org/0000-0003-4880-9610>)
rio	0.5.29	Jason Becker [ctb], Chung-hong Chan [aut] (<https://orcid.org/0000-0002-6232-7530>), Geoffrey CH Chan [ctb], Thomas J. Leeper [aut, cre] (<https://orcid.org/0000-0003-4097-6326>), Christopher Gandrud [ctb], Andrew MacDonald [ctb], Ista Zahn [ctb], Stanislaus Stadlmann [ctb], Ruaridh Williamson [ctb], Patrick Kennedy [ctb], Ryan Price [ctb], Trevor L Davis [ctb], Nathan Day [ctb], Bill Denney [ctb] (<https://orcid.org/0000-0002-5759-428X>), Alex Bokov [ctb] (<https://orcid.org/0000-0002-0511-9815>)
here	1.0.1	Kirill Müller [aut, cre] (<https://orcid.org/0000-0002-1416-3412>), Jennifer Bryan [ctb] (<https://orcid.org/0000-0002-6983-2759>)
forcats	0.5.1	Hadley Wickham [aut, cre], RStudio [cph, fnd]
stringr	1.5.0	Hadley Wickham [aut, cre, cph], RStudio [cph, fnd]
dplyr	1.1.2	Hadley Wickham [aut, cre] (<https://orcid.org/0000-0003-4757-117X>), Romain François [aut] (<https://orcid.org/0000-0002-2444-4226>), Lionel Henry [aut], Kirill Müller [aut] (<https://orcid.org/0000-0002-1416-3412>), Davis Vaughan [aut] (<https://orcid.org/0000-0003-4777-038X>), Posit Software, PBC [cph, fnd]

Table S1: Packages (and their versions) used for theses analyses. *(continued)*

Package	Version	Authors and contributors
purrr	1.0.1	Hadley Wickham [aut, cre] (<https://orcid.org/0000-0003-4757-117X>), Lionel Henry [aut], RStudio [cph, fnd]
readr	2.1.2	Hadley Wickham [aut], Jim Hester [aut], Romain Francois [ctb], Jennifer Bryan [aut, cre] (<https://orcid.org/0000-0002-6983-2759>), Shelby Bearrows [ctb], RStudio [cph, fnd], https://github.com/mandreyel/ [cph] (mio library), Jukka Jylänki [ctb, cph] (grisu3 implementation), Mikkel Jørgensen [ctb, cph] (grisu3 implementation)
tidyr	1.3.0	Hadley Wickham [aut, cre], Davis Vaughan [aut], Maximilian Girlich [aut], Kevin Ushey [ctb], Posit, PBC [cph, fnd]
tibble	3.2.1	Kirill Müller [aut, cre] (<https://orcid.org/0000-0002-1416-3412>), Hadley Wickham [aut], Romain Francois [ctb], Jennifer Bryan [ctb], RStudio [cph, fnd]
ggplot2	3.4.2	Hadley Wickham [aut] (<https://orcid.org/0000-0003-4757-117X>), Winston Chang [aut] (<https://orcid.org/0000-0002-1576-2126>), Lionel Henry [aut], Thomas Lin Pedersen [aut, cre] (<https://orcid.org/0000-0002-5147-4711>), Kohske Takahashi [aut], Claus Wilke [aut] (<https://orcid.org/0000-0002-7470-9261>), Kara Woo [aut] (<https://orcid.org/0000-0002-5125-4188>), Hiroaki Yutani [aut] (<https://orcid.org/0000-0002-3385-7233>), Dewey Dunnington [aut] (<https://orcid.org/0000-0002-9415-4582>), Posit, PBC [cph, fnd]
tidyverse	1.3.1	Hadley Wickham [aut, cre], RStudio [cph, fnd]

1 Descriptive Statistics

1.1 Demographics

```
# Code for table 1
ids <- setdiff(merged %>% filter(wave ==2) %>% select(id),
               merged %>% filter(wave ==1) %>% select(id))

demog_vars <- merged %>%
  select(id, wave, age, gender, hisp = ts.DEM06, race = ts.DEM07) %>%
  filter(wave ==1) %>%
  unique()

missing_demog <- merged[merged$id %in% ids$id,] %>%
  select(id, wave, age, gender, hisp = ts.DEM06, race = ts.DEM07) %>%
  unique()

demog <- rbind(demog_vars, missing_demog)

demog_dat <-    demog%>%
  mutate(ethnic_cat = case_when(hisp == 1  ~ 'Hispanic or Latino',
                       hisp == 0  ~ 'Not Hispanic or Latino',
                       race == 1   ~ 'American Indian or Alaska Native',
                       race == 2  ~ "Asian or Asian-American",
                       race == 3  ~ "Black or African American",
```

```
                      race == 4 ~ "Hawaiian Native or Pacific Islander",
                      race == 5 ~ "White",
                      race == 6 |  race == 7 | race == 8 |
                        race == 9 | race == 10 |
                        race == 11 | race == 12 | race == 13 |
                        race == 14 | race == 15 |
                         race == 16 |  race == 17 ~ "Other",
                      is.na(race) ~ "Not Reported"),
        gender_cat = case_when(gender == 1 ~ "Male",
                               gender == 2 ~ "Female"))



labels(demog_dat)  <- c(age = 'Age', gender_cat = "Gender",
                        ethnic_cat = "Race")
#create demog table for whole sample
demog_tab <- tableby(~ age + gender_cat + ethnic_cat,
                             data = demog_dat, test = FALSE)
summary(demog_tab, text = TRUE)
write2word(demog_tab, here("prior_results", "demog.doc"), title="Demographics")
save(demog_dat, demog_tab,  file = here("prior_results", "demog.rda"))
```

```
outcomes <- merged %>%
  select(id, wave, ts.CESD.tot =ts.CESD_tot,
         starts_with("ts.SQ"),
         starts_with("ts.AGQ"),
         starts_with("ts.PH")) %>%
  unique()


#write.csv(outcomes, file = "data/outcomes.csv")
```

```
outcomes_wide <- outcomes %>%
  select(id, wave, ts.CESD.tot,
         ts.SQ01:ts.SQ03, #family, friends, romantic
         ts.SQ05, #academics
         ts.SQ07, # physical health
         ts.SQ08, #psychological health
         ts.SQ10, #life
         ts.AGQ03,
         ts.PH) %>%
  pivot_wider(names_from = wave, names_sep = "_",
              values_from = c("ts.CESD.tot",
         "ts.SQ01", "ts.SQ02", "ts.SQ03", "ts.SQ05", "ts.SQ07",
         "ts.SQ08", "ts.SQ10", "ts.PH", "ts.AGQ03"))

outcomes_wide$id <- as.character(outcomes_wide$id)
```

1.2 ESM

Here we calculate descriptive statistics for the ESM and cross-sectional data (included as Table 2).

```
esm_descriptives <- merged %>%
  select(esm.reappraisal, esm.suppression, PosA = esm.NQ11,
         NegA = esm.NQ12, social = esm.ST13) %>%
  psych::describe(. , fast = TRUE)

esm_descriptives_wave <-  merged %>%
  select(esm.reappraisal, esm.suppression, PosA = esm.NQ11,
         NegA = esm.NQ12, social = esm.ST13) %>%
  psych::describeBy(. , group = merged$wave, fast = TRUE) %>%
  map(., data.frame) %>%
  map(., rownames_to_column)

esm_descriptives_wave<- rbind(esm_descriptives_wave$`1`, esm_descriptives_wave$`2`)
esm_descriptives_wave$wave <- c(rep(1, 5), rep(2, 5))

aff_cor <- correlation(merged, select = c("esm.NQ12"),
                       select2 = c("esm.ST13"), method = "pearson",
                       bayesian = TRUE)


esm_descriptives_wave2 <- esm_descriptives_wave %>%
   select(mean, sd, wave, item1 = rowname) %>%
  pivot_wider(., names_from = wave, values_from = c(mean, sd),
              names_glue = "{.value}_{wave}")%>%
  mutate(from = "esm")


wave_descriptives <- describeBy(outcomes %>% select(ts.CESD.tot,
         starts_with("ts.SQ"),
         starts_with("ts.AGQ"),
         starts_with("ts.PH")), outcomes$wave, fast = TRUE) %>%
  map(., data.frame) %>%
  map(., rownames_to_column)
wave_descriptives<- rbind(wave_descriptives$`1`, wave_descriptives$`2`)
wave_descriptives$wave <- c(rep(1, 13), rep(2, 13))

wave_descriptives2 <- wave_descriptives %>%
  select(mean, sd, wave, item1 = rowname) %>%
  pivot_wider(., names_from = wave, values_from = c(mean, sd),
              names_glue = "{.value}_{wave}") %>%
  mutate(from = "wave")
```

1.3 Wave/ESM Descriptive Correlations

Here we calculate the correlations across wave and conduct t-tests of wave 1 data between those who did and did not return for wave 2.

```
esm_wide <- merged %>%
    select(id, wave, esm.reappraisal, esm.suppression,
           PosA = esm.NQ11, NegA = esm.NQ12, social = esm.ST13) %>%
  group_by(id, wave) %>%
  summarize_all(., ~mean(.x, na.rm = TRUE)) %>%
```

```
  pivot_wider(., names_from = wave,
              values_from = c(esm.suppression,
                              esm.reappraisal,PosA, NegA, social),
              names_glue = "{.value}_{wave}") %>%
  ungroup()

esm_compare <- merged %>%
    select(id, wave, esm.reappraisal, esm.suppression,
           PosA = esm.NQ11, NegA = esm.NQ12, social = esm.ST13) %>%
  group_by(id, wave) %>%
  summarize_all(., ~mean(.x, na.rm = TRUE)) %>%
  group_by(id) %>%
  mutate(max = max(wave)) %>% # separate into those who have both waves or just wave 1
  filter(wave == 1) %>%
  ungroup



esm_descriptives_comp <-  esm_compare %>%
  group_by(max) %>%
  select(esm.reappraisal, esm.suppression, PosA, NegA, social) %>%
  psych::describeBy(. , group = esm_compare$max, fast = TRUE) %>%
  map(., data.frame) %>%
  map(., rownames_to_column) %>%
  bind_cols() %>%
  select(name = `rowname...1`, everything(), -contains("min"), -contains("max"),
         -contains("range"), -contains("vars"), -rowname...10, -contains("se")) %>%
  filter(name != "max")
```

```
## Adding missing grouping variables: `max`
## New names:
```

```
t_compare<- esm_compare %>%
  select(-id, -wave) %>%
        summarize_at(
            vars(-max),
            ~list(broom::tidy(t.test(.x ~ max)))) %>%
  pivot_longer(everything(), names_to = "name", values_to = "data") %>%
  unnest(cols = data) %>%
  select(name, estimate, statistic, p.value, contains("conf")) %>%
  mutate_if(is.numeric, apa_num) %>%
  mutate(estimate = paste0(estimate, " [", conf.low, ", ", conf.high, "]")) %>%
  select(name, estimate, statistic, p.value)


compare <- left_join(esm_descriptives_comp, t_compare)
```

```
## Joining with `by = join_by(name)`
```

```
compare %>%
  mutate(., name = case_when(
    str_detect(name, "esm.suppression")  ~  "Suppression",
```

```
    str_detect(name, "esm.reappraisal")  ~  "Reappraisal",
    str_detect(name, "social")  ~  "Sociality",
    str_detect(name, "PosA")  ~  "Positive Affect",
    str_detect(name, "NegA")  ~  "Negative Affect")) %>%
  kable(
    booktabs = TRUE,
    col.names = c("Measure","N",  "Mean", "SD", "N", "Mean", "SD",
                  "Mean Diff [CI]", "t", "p"),
    caption = "Comparisons of Returners vs. Non-Returners at Wave 1"
    ) %>%
    landscape() %>%
    kable_styling(font_size = 7,
                  latex_options = c("scale_down", "repeat_header")) %>%
    kable_classic() %>%
    add_header_above(c(" " = 1, "Non-Returners" = 3, "Returners" = 3, " " = 3))
```

Table S2: Comparisons of Returners vs. Non-Returners at Wave 1

	Non-Returners			Returners					
Measure	N	Mean	SD	N	Mean	SD	Mean Diff [CI]	t	p
Reappraisal	206	2.7514113	0.8599104	163	2.8480738	0.7816844	-0.10 [-0.27, 0.07]	-1.13	0.26
Suppression	206	2.7857674	0.7503821	163	2.8704174	0.7225089	-0.08 [-0.24, 0.07]	-1.10	0.27
Positive Affect	187	3.4208292	0.5513199	152	3.3924594	0.5503249	0.03 [-0.09, 0.15]	0.47	0.64
Negative Affect	187	2.1580075	0.5879769	152	2.1211937	0.5546420	0.04 [-0.09, 0.16]	0.59	0.55
Sociality	223	0.7560294	0.1891479	170	0.7510418	0.1824534	0.00 [-0.03, 0.04]	0.26	0.79

```
esm_corrs <- correlation(esm_wide,  method = "pearson", bayesian = TRUE)
esm_corrs2 <- esm_corrs %>%
  mutate(item1 = str_sub(Parameter1,1, nchar(Parameter1)-2),
         item2 = str_sub(Parameter2,1, nchar(Parameter2)-2)) %>%
  filter(., item1 == item2) %>%
  mutate(across(c("rho", "CI_low", "CI_high"), papaja::printnum)) %>%
  mutate(estimate = paste0(rho, " [", CI_low, ", ", CI_high, "]")) %>%
  data.frame() %>%
  select(., item1, estimate)




wave_corrs <- correlation(outcomes_wide,  method = "pearson", bayesian = TRUE)

wave_corrs2 <- wave_corrs %>%
  mutate(item1 = str_sub(Parameter1,1, nchar(Parameter1)-2),
         item2 = str_sub(Parameter2,1, nchar(Parameter2)-2)) %>%
  filter(., item1 == item2) %>%
  mutate(across(c("rho", "CI_low", "CI_high"), papaja::printnum)) %>%
  mutate(estimate = paste0(rho, " [", CI_low, ", ", CI_high, "]")) %>%
  data.frame() %>%
  select(., item1, estimate)
```

```
# Code for Table 2
corrs <- rbind(esm_corrs2, wave_corrs2)
descriptives <- rbind(esm_descriptives_wave2, wave_descriptives2)

all_descriptives <- left_join(descriptives, corrs) %>%
  data.frame() %>%
  filter(., item1 != "ts.SQ04", item1 != "ts.SQ06", item1 != "ts.SQ09") %>%
  mutate(., item1 = case_when(
    str_detect(item1, "esm.suppression")  ~  "Suppression",
    str_detect(item1, "esm.reappraisal")  ~  "Reappraisal",
    str_detect(item1, "ts.CESD.tot")  ~  "CESD",
    str_detect(item1, "social")  ~  "Sociality",
    str_detect(item1, "PosA")  ~  "Positive Affect",
    str_detect(item1, "NegA")  ~  "Negative Affect",
    str_detect(item1, "ts.SQ01")  ~  "Satisfaction - Family",
    str_detect(item1, "ts.SQ02")  ~  "Satisfaction - Friends",
    str_detect(item1, "ts.SQ03")  ~  "Satisfaction - Romantic",
    str_detect(item1, "ts.SQ05")  ~  "Satisfaction - Academics",
    str_detect(item1, "ts.SQ07")  ~  "Satisfaction - Physical Health",
    str_detect(item1, "ts.SQ08")  ~  "Satisfaction - Mental Health",
    str_detect(item1, "ts.SQ10")  ~  "Satisfaction - Life",
    str_detect(item1, "ts.AGQ03")  ~  "GPA",
    str_detect(item1, "ts.PH")  ~  "Physical Health") ,
    across(c(mean_1:sd_2), papaja::printnum)
    ) %>%
  mutate(from = factor(from,
                       levels = c("ESM", "Outcomes"))) %>%
  arrange(from) %>%
  select(item1, mean_1, sd_1, mean_2, sd_2, estimate)
```

```r
desc_table <- all_descriptives %>%
kable(
  booktabs = TRUE,
  longtable = TRUE,
  #escape = FALSE,
  col.names = c("Measure", "Mean", "SD", "Mean", "SD", "r [CI]"),
  caption = "Item Descriptives"
  ) %>%
  landscape() %>%
  kable_styling(font_size = 7,
                latex_options = c("scale_down", "repeat_header")) %>%
  kable_classic() %>%
  add_header_above(c(" " = 1, "Wave 1" = 2, "Wave 2" = 2, " " = 1)) %>%
  group_rows("ESM", 1, 5) %>%
  group_rows("Outcomes", 6, 15)

save(desc_table, file = here("prior_results", "descriptives_table.rda"))

save(esm_descriptives, esm_descriptives_wave, wave_descriptives,
     all_descriptives, file = here("prior_results", "descriptives.rda"))
```

2 The Models

Here we include the formal code for our various models.

```r
prior_var <- c(prior(normal(2.5, 1.5), class = Intercept),
           prior(normal(0, 2), class = b),
           prior(normal(2.5, 1.5), class = sd, coef = Intercept, group = id),
           prior(normal(0, 2), class = sd, coef = age_calc_c, group = id),
           prior(normal(0, 1), class = Intercept, dpar = sigma),
           prior(normal(0, 1), class = b, dpar = sigma),
           prior(exponential(1), class = sd, dpar = sigma),
           prior(lkj(2), class = cor))



fit_var_sup <- brm(bf(

                   esm.suppression ~ 1  + age_calc_c +
                     (1 + age_calc_c | c | id),
                   sigma ~ 1  + age_calc_c +  (1 + age_calc_c | c | id)),
                   data=merged, inits = 0,
                   cores = 4, sample_prior = T,
                   iter = 5000, warmup = 2500,
                   prior = prior_var,
                   file = here("prior_results", "fit_var_sup"))

summary(fit_var_sup)
mcmc_plot(fit_var_sup_prior, type = 'dens')

rm(list = setdiff(ls(), c("merged", "prior_var")))
```

```
fit_var_app <- brm(bf(

                    esm.reappraisal ~ 1  + age_calc_c +
                      (1 + age_calc_c | c | id) ,
                    sigma ~ 1  + age_calc_c +  (1 + age_calc_c | c | id)),
                    data=merged, inits = 0,
                    cores = 4, sample_prior = T,
                    iter = 5000, warmup = 2500,
                    control= list(adapt_delta = .99, max_treedepth = 15,
                                  stepsize = .1),
                    prior = prior_var,
                    file = here("prior_results", "fit_var_app"))



mcmc_plot(
  fit_var_app,
  pars = c('b_age_calc_c',  "b_sigma_age_calc_c", "b_Intercept",
           "b_sigma_intercept"),
  type = 'pairs',
  diag_fun = 'dens',
  off_diag_fun = 'hex',
  fixed = TRUE
)
prior_summary(fit_var_app, data = merged)



summary(fit_var_app)
mcmc_plot(fit_var_app, type = 'dens')

rm(list = setdiff(ls(), c("merged")))
```

```
nest_var_prior <-  c(prior(normal(2.5, 1.5), class = Intercept),# group intercept
           prior(normal(2.5, 1.5), class = sd,
                 coef = Intercept, group = id:wave),  # individual intercept
           prior(normal(0, 1), class = Intercept, dpar = sigma),
           prior(exponential(1), class = sd, dpar = sigma), #
           prior(lkj(2), class = cor),                      # correlation
           prior(lkj(2), class = cor, group = id:wave)      # correlation by wave

   )




nest_var_sup <- brm(bf(esm.suppression ~ 1  +  (1  | c | id:wave),
                    sigma ~ 1 + (1 | c | id:wave)) ,
                    data=merged, inits = 0,
                    cores = 4, sample_prior = T,
                    iter = 7500, warmup = 3750,
                    prior = nest_var_prior,
```

```
                  file = here("prior_results", "nest_var_sup"))

mcmc_plot(nest_var_sup, type = 'dens')

summary(nest_var_sup)

rm(list = setdiff(ls(), c("merged", "nest_var_prior")))



nest_var_app <- brm(bf(

                  esm.reappraisal ~ 1  +  (1  | c | id:wave) ,
                  sigma ~ 1 + (1 | c | id:wave)) ,
                  data=merged, inits = 0,
                  cores = 4, sample_prior = T,
                  iter = 7500, warmup = 3750,
                  prior = nest_var_prior,
                  control= list(adapt_delta = .99),
                  file = here("prior_results", "nest_var_app"))

summary(nest_var_app)
mcmc_plot(nest_var_app, type = 'dens')
mcmc_plot(nest_var_app, type = 'pairs', diag_fun = 'dens', off_diag_fun = 'hex')


rm(list = setdiff(ls(), c("merged")))
```

```
nest_flex_formula <- bf(esm.suppression ~ 1  + PA_avg + NA_avg +
                          social_avg + PA_centered + NA_centered +
                          social_centered +
                          (1 + PA_centered + NA_centered +  social_centered
                           | c | id:wave),
                  sigma ~ 1 + (1 | c | id:wave))
get_prior(nest_flex_formula, data = merged)

nest_flex_prior <-  c(prior(normal(2.5, 1.5), class = Intercept),  # group int
                      prior(normal(0, 2), class = b),# global slope prior
                      prior(normal(2.5, 1.5), class = sd,
                            coef = Intercept, group = id:wave), # individual int
                      prior(normal(0, 1), class = sd, group = id:wave),
                      prior(normal(0, 1), class = Intercept, dpar = sigma),
                      prior(exponential(1), class = sd, dpar = sigma),
                      prior(lkj(2), class = cor),                  # correlation
                      prior(lkj(2), class = cor, group = id:wave) #corr by wave

   )




flex_rank_sup <- brm(bf(esm.suppression ~ 1  + PA_avg + NA_avg + social_avg +
```

```
                        PA_centered + NA_centered +  social_centered +
                        (1 + PA_centered + NA_centered +  social_centered
                         | c | id:wave),
                    sigma ~ 1 + (1 | c | id:wave)) ,
                    data=merged, inits = 0,
                    cores = 4, sample_prior = T,
                    iter = 15000, warmup = 7500,
                    prior = nest_flex_prior,
                    control= list(adapt_delta = .99),
                    file = here("prior_results", "flex_rank_sup_wave"))



rm(list = setdiff(ls(), c("merged", "nest_flex_prior")))
#adapt
flex_rank_app <- brm(bf(

                    esm.reappraisal ~ 1  + PA_avg + NA_avg + social_avg +
                      PA_centered + NA_centered +  social_centered +
                      (1 + PA_centered + NA_centered +  social_centered
                       | c | id:wave),
                    sigma ~ 1 + (1 | c | id:wave)) ,
                    data=merged, inits = 0,
                    cores = 4, sample_prior = T,
                    iter = 15000, warmup = 7500,
                    prior = nest_flex_prior,
                    control= list(adapt_delta = .99),

                    file = here("prior_results", "flex_rank_app_wave"))


summary(flex_rank_app)
mcmc_plot(flex_rank_app, type = 'dens')


rm(list = setdiff(ls(), c("merged")))
```

```
prior_flex_change <- c(prior(normal(2.5, 1.5), class = Intercept),
           prior(normal(0, 2), class = b),
           prior(normal(2.5, 1.5), class = sd, coef = Intercept, group = id),
           prior(normal(0, 2), class = sd, coef = age_calc_c, group = id),
           prior(normal(0, 1), class = Intercept, dpar = sigma),
           prior(normal(0, 1), class = b, dpar = sigma),
           prior(exponential(1), class = sd, dpar = sigma),
           prior(lkj(2), class = cor))



flex_change_sup <- brm(bf(esm.suppression ~ 1  + PA_avg + NA_avg + social_avg +
                            PA_centered + NA_centered +  social_centered +
                            age_calc_c +
```

```
                          (1 + PA_centered + NA_centered +  social_centered +
                             age_calc_c | c | id),
                    sigma ~ 1 + age_calc_c +
                      (1 + age_calc_c | c | id)) ,
                    data=merged, inits = 0,
                    cores = 4, sample_prior = T,
                    iter = 15000, warmup = 7500,
                    prior = prior_flex_change,
                    control= list(adapt_delta = .99),
                    file = here("prior_results", "flex_change_sup"))

summary(flex_change_sup)
mcmc_plot(flex_change_sup, type = 'dens')



rm(list = setdiff(ls(), c("merged", "prior_flex_change")))


flex_change_app <- brm(bf(

                    esm.reappraisal ~ 1  + PA_avg + NA_avg + social_avg +
                      PA_centered + NA_centered +  social_centered + age_calc_c +
                      (1 + PA_centered + NA_centered +  social_centered +
                         age_calc_c | c | id),
                    sigma ~ 1 + age_calc_c + (1 + age_calc_c | c | id)) ,
                    data=merged, inits = 0,
                    cores = 4, sample_prior = T,
                    iter = 15000, warmup = 7500,
                    prior = prior_flex_change,
                    control= list(adapt_delta = .99),
                    file = here("prior_results", "flex_change_app"))

summary(flex_change_app)
mcmc_plot(flex_change_app, type = 'dens')
```

3 Results

3.1 Load Model Output

```
fit_var_app <- import(here("prior_results", "fit_var_app.rds"))
fit_var_sup <- import(here("prior_results", "fit_var_sup.rds"))
flex_change_app <- import(here("prior_results", "flex_change_app.rds"))
flex_change_sup <- import(here("prior_results", "flex_change_sup.rds"))
```

```
# Code for Table 4
df = data.frame(
  ER = c("Suppression", "Reappraisal","Suppression", "Reappraisal"),
  type = c("fit", "fit", "flex", "flex")
)
```

```
df$model = list(fit_var_sup, fit_var_app, flex_change_sup, flex_change_app)

df = df %>%
  mutate(slope_draws = map(model, gather_draws, b_age_calc_c,
                           b_sigma_age_calc_c)) %>%
  mutate(estimates = map(slope_draws, median_qi)) %>%
  select(ER, type, estimates) %>%
  unnest(cols = c(estimates)) %>%
  mutate(across(where(is.numeric), papaja::printnum)) %>%
  mutate(estimate = paste0(.value, " [", .lower, ", ", .upper, "]"))

save(df, file = here("prior_results", "all_changemodel_estimates.rda"))

change_table <- df %>%
  mutate(.variable = case_when(
    str_detect(.variable, "sigma") & type == "fit" ~  "Change in Variability",
    str_detect(.variable, "sigma") & type == "flex" ~  "Change in Flexibility",
    type == "flex" ~  "Change in Tendency Controlling for Situation",
    TRUE ~ "Change in Tendency"
  )) %>%
  mutate(.variable = factor(.variable,
                            levels = c("Change in Tendency",
                                       "Change in Tendency Controlling for Situation",
                                       "Change in Variability",
                                       "Change in Flexibility"))) %>%
  select(ER, .variable, estimate) %>%
  pivot_wider(names_from = ER, values_from =  estimate) %>%
  arrange(.variable) %>%
kable(
  booktabs = TRUE,
  #longtable = TRUE,
  #escape = FALSE,
  col.names = c("Measure", "estimate", "estimate"),
  caption = "Age Related Change in ER"
  ) %>%
  landscape() %>%
  kable_styling(font_size = 7,
                latex_options = c("scale_down", "repeat_header")) %>%
  kable_classic() %>%
  add_header_above(c(" " = 1, "Suppression" = 1, "Reappraisal" = 1))

save(change_table, file = here("prior_results", "change_table.rda"))

#merged %>%
#  group_by(id, wave) %>%
#  slice(1) %>%
#  pivot_wider(id_cols = id, names_from = wave, values_from = StartDate) %>%
#  ungroup() %>%
#  mutate(datediff = difftime(`1`, `2`)) %>%
#  summarise(mean(datediff, na.rm = TRUE))
```

3.2 Mean/Var Correlation Suppression

```
nest_var_sup <- import(here("prior_results", "nest_var_sup.rds"))


person_draws_sup_mean = nest_var_sup %>%
  gather_draws(`r_id:wave`[id, term])

person_draws_sup2_mean <- person_draws_sup_mean
person_draws_sup_mean <- person_draws_sup2_mean

person_draws_sup_mean <-  person_draws_sup_mean %>%
  group_by(id, term) %>%
  median_qi(.value) %>% # get median estimate for each term and each person
  select(id, term, .value) %>%
  # put into wide form or one row per person
  spread(term, .value) %>%
  separate(id, c("id", "time"), sep = "_")

person_draws_sup_wide_mean <- person_draws_sup_mean %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "Intercept")


set.seed(1)
mean_sup_corr <- correlation(person_draws_sup_wide_mean,
                             select = '1', select2 = '2',
                             method = "pearson", bayesian = TRUE)


person_draws_sup_nest_sigma = nest_var_sup %>%
  gather_draws(`r_id:wave__sigma`[id, term])

person_draws_sup_nest_sigma2 <- person_draws_sup_nest_sigma
person_draws_sup_nest_sigma <- person_draws_sup_nest_sigma2

person_draws_sup_wide_sigma = person_draws_sup_nest_sigma %>%
  group_by(id, term) %>%
  median_qi(.value) %>% # get median estimate for each term and each person
  select(id, term, .value) %>%
  # put into wide form or one row per person
  spread(term, .value) %>%
  separate(id, c("id", "time"), sep = "_")

  person_draws_sup_wide_sigma <- person_draws_sup_wide_sigma %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "Intercept")

set.seed(1)
var_sup_corr <- correlation(person_draws_sup_wide_sigma,
                            select = '1', select2 = '2',
                            method = "pearson", bayesian = TRUE)


rm(list = setdiff(ls(), c("merged", "mean_sup_corr",
```

```
                          "var_sup_corr", "person_draws_sup_wide_mean",
                          "person_draws_sup_wide_sigma")))
```

3.3 Mean/Var Correlation Reappraisal

```
nest_var_app <- import(here("prior_results", "nest_var_app.rds"))
#summary(nest_mean_app)


person_draws_app_mean = nest_var_app %>%
  gather_draws(`r_id:wave`[id, term])

person_draws_app2_mean <- person_draws_app_mean
person_draws_app_mean <- person_draws_app2_mean

person_draws_app_mean <-  person_draws_app_mean %>%
  group_by(id, term) %>%
  median_qi(.value) %>% # get median estimate for each term and each person
  select(id, term, .value) %>%
  # put into wide form or one row per person
  spread(term, .value) %>%
  separate(id, c("id", "time"), sep = "_")

person_draws_app_wide_mean <- person_draws_app_mean %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "Intercept")


set.seed(1)
mean_app_corr <-correlation(person_draws_app_wide_mean, select = '1',
                            select2 = '2',  method = "pearson", bayesian = TRUE)


person_draws_app_nest_sigma = nest_var_app %>%
  gather_draws(`r_id:wave__sigma`[id, term])

person_draws_app_nest_sigma2 <- person_draws_app_nest_sigma
person_draws_app_nest_sigma <- person_draws_app_nest_sigma2

person_draws_app_nest_sigma = person_draws_app_nest_sigma %>%
  group_by(id, term) %>%
  median_qi(.value) %>% # get median estimate for each term and each person
  select(id, term, .value) %>%
  # put into wide form or one row per person
  spread(term, .value) %>%
  separate(id, c("id", "time"), sep = "_")

  person_draws_app_wide_sigma <- person_draws_app_nest_sigma %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "Intercept")

set.seed(1)
var_app_corr <- correlation(person_draws_app_wide_sigma,
```

```
                            select = '1', select2 = '2',
                            method = "pearson", bayesian = TRUE)

rm(list = setdiff(ls(), c("merged", "mean_sup_corr", "var_sup_corr",
                          "person_draws_sup_wide_mean",
                          "person_draws_sup_wide_sigma",
                          "mean_app_corr", "var_app_corr",
                          "person_draws_app_wide_mean",
                          "person_draws_app_wide_sigma")))
```

3.4 Flexibility Correlation Suppression

```
flex_rank_sup <- import(here("prior_results", "flex_rank_sup_wave.rds"))
#summary(flex_rank_sup)
#sup_vars <- get_variables(flex_rank_sup)


# get participant level intercept and slopes for each iteration
person_draws_flex_rank_sup_sigma = flex_rank_sup %>%
  gather_draws(`r_id:wave__sigma`[id, term])


person_draws_flex_rank_sup2_sigma <- person_draws_flex_rank_sup_sigma
person_draws_flex_rank_sup_sigma <- person_draws_flex_rank_sup2_sigma

person_draws_flex_rank_sup_sigma <-  person_draws_flex_rank_sup_sigma %>%
  group_by(id, term) %>%
  median_qi(.value) %>% # get median estimate for each term and each person
  select(id, term, .value) %>%
  spread(term, .value) %>%
  separate(id, c("id", "time"), sep = "_")

  person_draws_flex_rank_sup_wide_sigma <- person_draws_flex_rank_sup_sigma %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "Intercept")

set.seed(1)
flex_sup_corr <- correlation(person_draws_flex_rank_sup_wide_sigma,
                             select = '1', select2 = '2',
                             method = "pearson", bayesian = TRUE)




rm(list = setdiff(ls(), c("merged", "mean_sup_corr", "var_sup_corr",
                          "flex_sup_corr", "flex_rank_sup",
                          "mean_app_corr", "var_app_corr",
                          "person_draws_sup_wide_mean",
                          "person_draws_sup_wide_sigma",
                          "person_draws_app_wide_mean",
                          "person_draws_app_wide_sigma",
                          "person_draws_flex_rank_sup_wide_sigma")))
```

3.5 Reactivity Correlation Suppression

```
# get random slopes
person_draws_flex_rank_sup_slope = flex_rank_sup %>%
  gather_draws(`r_id:wave`[id, term])

person_draws_flex_rank_sup_slope2 <- person_draws_flex_rank_sup_slope
person_draws_flex_rank_sup_slope <- person_draws_flex_rank_sup_slope2

person_draws_flex_rank_sup_slope <-  person_draws_flex_rank_sup_slope %>%
  group_by(id, term) %>%
  median_qi(.value) %>% # get median estimate for each term and each person
  select(id, term, .value) %>%
  # put into wide form or one row per person
  spread(term, .value) %>%
  separate(id, c("id", "time"), sep = "_")


rank_slopes_sup <- person_draws_flex_rank_sup_slope %>%
  select(id, time, Intercept, social_centered, NA_centered, PA_centered) %>%
  pivot_wider(names_from = time, names_sep = "_",
              values_from = c("Intercept", "social_centered",
                              "PA_centered", "NA_centered"))



intercept_rank_sup <- person_draws_flex_rank_sup_slope %>%
  select(id, time, Intercept) %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "Intercept")

set.seed(1)
tend_rank_sup <- correlation(intercept_rank_sup, select = '1', select2 = '2',
                             method = "pearson", bayesian = TRUE)


na_rank_slopes_sup <- person_draws_flex_rank_sup_slope %>%
  select(id, time, NA_centered) %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "NA_centered")

set.seed(1)
na_rank_sup <- correlation(na_rank_slopes_sup, select = '1', select2 = '2',
                           method = "pearson", bayesian = TRUE)


pa_rank_slopes_sup <- person_draws_flex_rank_sup_slope %>%
  select(id, time, PA_centered) %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "PA_centered")

set.seed(1)
pa_rank_sup <- correlation(pa_rank_slopes_sup, select = '1', select2 = '2',
```

```
                          method = "pearson", bayesian = TRUE)


social_rank_slopes_sup <- person_draws_flex_rank_sup_slope %>%
  select(id, time, social_centered) %>%
  pivot_wider(names_from = time, names_sep = "_",
              values_from = "social_centered")

set.seed(1)
soc_rank_sup <- correlation(social_rank_slopes_sup, select = '1', select2 = '2',
                            method = "pearson", bayesian = TRUE)


rm(list = setdiff(ls(), c("merged", "mean_sup_corr", "var_sup_corr",
                          "flex_sup_corr",
                         "tend_rank_sup", "na_rank_sup", "pa_rank_sup",
                         "soc_rank_sup", "rank_slopes_sup",
                          "mean_app_corr", "var_app_corr",
                          "person_draws_sup_wide_mean",
                         "person_draws_sup_wide_sigma",
                          "person_draws_app_wide_mean",
                         "person_draws_app_wide_sigma",
                          "person_draws_flex_rank_sup_wide_sigma")))
```

3.6 Flexibility Reliability Reappraisal

```
flex_rank_app <- import(here("prior_results", "flex_rank_app_wave.rds"))
#summary(flex_rank_app)


# get participant level intercept and slopes for each iteration
person_draws_flex_rank_app_sigma = flex_rank_app %>%
  gather_draws(`r_id:wave__sigma`[id, term])


person_draws_flex_rank_app2_sigma <- person_draws_flex_rank_app_sigma
person_draws_flex_rank_app_sigma <- person_draws_flex_rank_app2_sigma

person_draws_flex_rank_app_sigma <-  person_draws_flex_rank_app_sigma %>%
  group_by(id, term) %>%
  median_qi(.value) %>% # get median estimate for each term and each person
  select(id, term, .value) %>%
  spread(term, .value) %>%
  separate(id, c("id", "time"), sep = "_")

  person_draws_flex_rank_app_wide_sigma <- person_draws_flex_rank_app_sigma %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "Intercept")

set.seed(1)
flex_app_corr <- correlation(person_draws_flex_rank_app_wide_sigma,
                             select = '1', select2 = '2',
```

```
                            method = "pearson", bayesian = TRUE)



rm(list = setdiff(ls(), c("merged", "mean_sup_corr", "var_sup_corr", "flex_sup_corr",
                          "tend_rank_sup", "na_rank_sup", "pa_rank_sup",
                          "soc_rank_sup", "rank_slopes_sup",
                           "mean_app_corr", "var_app_corr", "flex_app_corr",
                          "flex_rank_app",
                           "person_draws_sup_wide_mean",
                          "person_draws_sup_wide_sigma",
                           "person_draws_app_wide_mean",
                          "person_draws_app_wide_sigma",
                           "person_draws_flex_rank_sup_wide_sigma",
                          "person_draws_flex_rank_app_wide_sigma")))
```

3.7 Reactivity Reliability Reappraisal

```
# get random slopes
person_draws_flex_rank_app_slope = flex_rank_app %>%
  gather_draws(`r_id:wave`[id, term])

person_draws_flex_rank_app_slope2 <- person_draws_flex_rank_app_slope
person_draws_flex_rank_app_slope <- person_draws_flex_rank_app_slope2

person_draws_flex_rank_app_slope <-  person_draws_flex_rank_app_slope %>%
  group_by(id, term) %>%
  median_qi(.value) %>% # get median estimate for each term and each person
  select(id, term, .value) %>%
  # put into wide form or one row per person
  spread(term, .value) %>%
  separate(id, c("id", "time"), sep = "_")


rank_slopes_app <- person_draws_flex_rank_app_slope %>%
  select(id, time, Intercept, social_centered, NA_centered, PA_centered) %>%
  pivot_wider(names_from = time, names_sep = "_",
              values_from = c("Intercept", "social_centered",
                              "PA_centered", "NA_centered"))


intercept_rank_app <- person_draws_flex_rank_app_slope %>%
  select(id, time, Intercept) %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "Intercept")

set.seed(1)
tend_rank_app <- correlation(intercept_rank_app, select = '1', select2 = '2',
                             method = "pearson", bayesian = TRUE)
```

```
na_rank_slopes_app <- person_draws_flex_rank_app_slope %>%
  select(id, time, NA_centered) %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "NA_centered")

set.seed(1)
na_rank_app <- correlation(na_rank_slopes_app, select = '1', select2 = '2',
                           method = "pearson", bayesian = TRUE)



pa_rank_slopes_app <- person_draws_flex_rank_app_slope %>%
  select(id, time, PA_centered) %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "PA_centered")

set.seed(1)
pa_rank_app <- correlation(pa_rank_slopes_app, select = '1', select2 = '2',
                           method = "pearson", bayesian = TRUE)


social_rank_slopes_app <- person_draws_flex_rank_app_slope %>%
  select(id, time, social_centered) %>%
  pivot_wider(names_from = time, names_sep = "_", values_from = "social_centered")

set.seed(1)
soc_rank_app <- correlation(social_rank_slopes_app, select = '1', select2 = '2',
                            method = "pearson", bayesian = TRUE)




rm(list = setdiff(ls(), c("merged", "mean_sup_corr", "var_sup_corr",
                          "flex_sup_corr",
                         "tend_rank_sup", "na_rank_sup", "pa_rank_sup",
                         "soc_rank_sup", "rank_slopes_sup",
                          "mean_app_corr", "var_app_corr", "flex_app_corr",
                          "na_rank_app", "pa_rank_app", "soc_rank_app",
                         "tend_rank_app", "rank_slopes_app",
                          "person_draws_sup_wide_mean",
                         "person_draws_sup_wide_sigma",
                          "person_draws_app_wide_mean",
                         "person_draws_app_wide_sigma",
                          "person_draws_flex_rank_sup_wide_sigma",
                         "person_draws_flex_rank_app_wide_sigma")))
```

3.8 Correlation Tables

```
corrs <- data.frame(rbind(mean_sup_corr, var_sup_corr, flex_sup_corr,
                          tend_rank_sup,
                               na_rank_sup, pa_rank_sup, soc_rank_sup,
                               mean_app_corr, var_app_corr,
```

```
                          flex_app_corr,tend_rank_app,
                               na_rank_app, pa_rank_app, soc_rank_app))%>%
  select(rho, CI_low, CI_high, pd, BF)

corrs$Measures <- c("Tendency", "Variability", "Flexibility",
                    "Tendency - Controlling for Context",
                        "NA Reactivity", "PA Reactivity", "Social Reactivity",
                          "Tendency", "Variability", "Flexibility",
                    "Tendency - Controlling for Context",
                        "NA Reactivity", "PA Reactivity", "Social Reactivity")
corrs$Strategy <- c(rep("Suppression", 7), rep("Reappraisal", 7))


load(here("prior_results", "all_reliability_estimates.rda" ))

corrs_tab <- corrs %>%
 mutate(across(where(is.numeric), papaja::printnum)) %>%
  mutate(estimate = paste0(rho, " [", CI_low, ", ", CI_high, "]"))


corrs_tab <- corrs_tab %>%
  select(Strategy, Measures, estimate) %>%
  pivot_wider(names_from = Strategy, values_from =  estimate)

corrs_tab %>%
  kable(
  booktabs = TRUE,
  col.names = c("Measure", "Suppression", "Reappraisal"),
  caption = "Reliability of ER"
  ) %>%
  landscape() %>%
  kable_styling(font_size = 7,
                latex_options = c("scale_down", "repeat_header")) %>%
  kable_classic()

save(corrs, file = here("prior_results", "all_reliability_estimates.rda" ))

write.csv(corrs_tab, file = here("new_plots2022", "rel_corrs_tab.csv"))

write.csv(corrs, file = here("new_plots2022", "rel_corrs.csv"))
```

```
names(person_draws_app_wide_mean) <- c("id", "tend_app_1", "tend_app_2")
names(person_draws_sup_wide_mean) <- c("id", "tend_sup_1", "tend_sup_2")

names(person_draws_app_wide_sigma) <- c("id", "var_app_1", "var_app_2")
names(person_draws_sup_wide_sigma) <- c("id", "var_sup_1", "var_sup_2")

names(person_draws_flex_rank_app_wide_sigma) <- c("id", "flex_app_1" ,"flex_app_2")
names(person_draws_flex_rank_sup_wide_sigma) <- c("id", "flex_sup_1" ,"flex_sup_2")



suppression <- full_join(person_draws_sup_wide_mean, person_draws_sup_wide_sigma)
```

```
suppression <- full_join(suppression, person_draws_flex_rank_sup_wide_sigma)
suppression <- full_join(suppression, rank_slopes_sup)


reappraisal <- full_join(person_draws_app_wide_mean, person_draws_app_wide_sigma)
reappraisal <- full_join(reappraisal, person_draws_flex_rank_app_wide_sigma)
reappraisal <- full_join(reappraisal, rank_slopes_app)

save(reappraisal, suppression, file = here("data", "model_results.Rda"))

rm(list = setdiff(ls(), c("merged", "suppression", "reappraisal")))
```

3.9 Sensitivity Analyses

```
merged2 <- left_join(merged, outcomes)

intra <- merged2 %>%
  group_by(id, wave) %>%
  mutate(sup_iM = mean(esm.suppression, na.rm = TRUE),
         sup_iSD = sd(esm.suppression, na.rm = TRUE),
         app_iM = mean(esm.reappraisal, na.rm = TRUE),
         app_iSD = sd(esm.reappraisal, na.rm = TRUE),
         ) %>%
  select(id, wave, starts_with("sup"), starts_with("app")) %>%
  unique() %>%
  pivot_wider( id_cols = id, values_from = c(sup_iM, sup_iSD, app_iM, app_iSD),
               names_from = wave, names_glue = "{.value}_{wave}") %>%
  ungroup()


corrs <- intra %>%
  select(-id) %>%
  correlation(.,  bayesian = TRUE, method = "pearson") %>%
               separate(., Parameter1,
                        into = c("ER1", "Parameter1", "Time1"), sep = "_") %>%
               separate(., Parameter2,
                        into = c("ER2", "Parameter2", "Time2"), sep = "_") %>%
               filter(., Parameter1 == Parameter2 & ER1 == ER2)

save(corrs, file = here("prior_results", "intra_corrs.rda"))

corrs <- corrs %>%
  mutate(across(c("rho", "CI_low", "CI_high"), papaja::printnum)) %>%
  mutate(estimate = paste0(rho, " [", CI_low, ", ", CI_high, "]"),
         Term = paste(ER1, Parameter1, sep = "_"))

corr_table <- corrs%>%
  select(Term, estimate) %>%
  as.data.frame()

save(corr_table, file = here("prior_results", "intra_corr_table.rda"))
```

```r
outcomes_wide$id <- as.numeric(outcomes_wide$id)
intra_outs <- left_join(intra, outcomes_wide) %>% data.frame()

intra_corrs <- intra_outs %>%
  correlation(., bayesian = TRUE, method = "pearson") %>%
  filter (., Parameter1 == "sup_iM_1" | Parameter1 == "sup_iSD_1"|
             Parameter1 == "sup_iM_2" | Parameter1 == "sup_iSD_2"|
             Parameter1 == "app_iM_1" | Parameter1 == "app_iSD_1"|
             Parameter1 == "app_iM_2" | Parameter1 == "app_iSD_2") %>%
  group_by(Parameter1) %>%
  filter(., Parameter2 == "ts.CESD.tot_1" | Parameter2 == "ts.PC.tot_1"|
            Parameter2 =="ts.SQ01_1" | Parameter2 == "ts.SQ02_1" |
            Parameter2 == "ts.SQ03_1" | Parameter2 == "ts.SQ05_1" |
            Parameter2 =="ts.SQ07_1" | Parameter2 == "ts.SQ08_1"|
            Parameter2 == "ts.SQ10_1"| Parameter2 =="ts.PH_1"|
            Parameter2 =="ts.AGQ03_1"|
            Parameter2 == "ts.CESD.tot_2" | Parameter2 == "ts.PC.tot_2"|
            Parameter2 =="ts.SQ01_2" | Parameter2 == "ts.SQ02_2" |
            Parameter2 == "ts.SQ03_2" | Parameter2 == "ts.SQ05_2" |
            Parameter2 =="ts.SQ07_2" | Parameter2 == "ts.SQ08_2"|
            Parameter2 == "ts.SQ10_2"| Parameter2 =="ts.PH_2"|
            Parameter2 =="ts.AGQ03_2") %>%
    mutate(across(c("rho", "CI_low", "CI_high"), papaja::printnum)) %>%
  mutate(estimate = paste0(rho, " [", CI_low, ", ", CI_high, "]")) %>%
  separate(., Parameter1, into = c("ER", "Parameter1", "Time1"), sep = "_") %>%
  separate(., Parameter2, into = c("Parameter2", "Time2"), sep = "_") %>%
  filter(., Time1 == Time2) %>%
  mutate(Parameter1 = paste(ER, Parameter1, sep = "_"))


save(intra_corrs, file = here("prior_results", "intra_out_corrs.rda"))
```

```r
intra_corrtable <- intra_corrs %>%
  arrange(Parameter2) %>%
  select(., Parameter1, Parameter2, estimate, Time1) %>%
  pivot_wider(names_from = c(Parameter2, Time1), names_sep = "-",
              values_from =  estimate, id_cols = Parameter1) %>%
  select(Parameter1, starts_with("ts.CESD"), starts_with("ts.SQ"),
         starts_with("ts.PH"), starts_with("ts.AC"))


colnames(intra_corrtable) <- c("", "Depression",
                               "", "Satisfaction - Family",
```

```
                                "", "Satisfaction - Friends",
                                "", "Satisfaction - Romantic",
                                "", "Satisfaction - Academics",
                                "", "Satisfaction - Physical Health",
                                "", "Satisfaction - Mental Health",
                                "", "Satisfaction - Life",
                                "", "Physical Health",
                                "", "GPA", "")


save(intra_corrtable, file = here("prior_results", "intra_outcome_corrs.rda"))
```

```
load(here("data", "model_results.Rda"))# has merged files of all relevant metrics
```

4 Plots

```
reappraisal_long <- reappraisal %>%
  select(id, contains("_app") | contains("_sup")) %>%
  pivot_longer(!id, names_to = "var", values_to = "estimate") %>%
 separate(var, sep = "_", into = c("var", "er", "time")) %>%
na.omit


suppression_long <- suppression %>%
  select(id, contains("_app") | contains("_sup")) %>%
  pivot_longer(!id, names_to = "var", values_to = "estimate") %>%
 separate(var, sep = "_", into = c("var", "er", "time")) %>%
na.omit

estimates_long <- rbind(reappraisal_long, suppression_long) %>%
  arrange(er) %>%
  mutate(var =
           case_when(
    var == "tend" ~ "Tendency",
    var == "flex" ~ "Flexibility",
    var == "var" ~ "Variability"),
    er =
      case_when(
        er == "sup" ~ "Suppression",
        er == "app" ~ "Reappraisal"
))

estimates_long %>%
mutate(var = factor(var, levels =
                      c("Tendency", "Variability", "Flexibility"))) %>%
  arrange(var, er) %>%
  mutate(time = as.numeric(time)) %>%
  ggplot(aes(x = time, y = estimate)) +
  geom_point(alpha = .3) +
  geom_line(aes(group = id), alpha = .1) +
```

```
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_continuous(breaks = c(1, 2))+
  theme(axis.title.y=element_blank(),
        axis.title.x=element_blank())+
    facet_grid(er~var, switch = "both")

ggsave("new_plots2022/reliability.png",
       width = 6.25, height = 6.25, units = "in")
```