

JA_Analyses_Final

T.L

2023-03-27

Packages

```
library(rstatix) # Basic analyses like ANOVA etc  
library(lme4) # For multilevel models  
library(lmerTest) # For lmm p-values  
library(tidyverse) # For pivottting longer etc  
library(sjPlot) #for exporting nice tables of our output  
library(flextable) #another package to make apa tables  
library(robustlmm) #For robust lmm when normality violated  
library(multilevelTools)  
library(JWileymisc) # To use modelDiagnostics  
library(ggpubr) # Plotting  
library(emmeans) # to follow up on sig mixed models interactions  
library(Hmisc) # Correlation analyses  
library(knitr) # For printing nice tables  
library(papaja) # To format figures in APA  
library(reshape2) # Can use functions like melt  
library(chisq.posthoc.test) # For post hoc chisq tests  
library(RVAideMemoire) # Another post hoc test package
```

Load in data

```
load(file = "Judge_Advisor_2022_preprocessed.RData")  
  
#turning off scientific notation  
options(scipen=999)
```

Opinion difference

```
### Put data in a long dataframe  
  
pracs <- select(full_ja_df, ID, AGEGROUP, PRAC1_1n, PRAC1_2n, PRAC1_3n)  
  
pracs$ID <- as.factor(pracs$ID)  
  
# Actual prac jars values  
## Jar 1 = 188  
## Jar 2 = 141  
## Jar 3 = 71  
  
# Create opinion difference scores (initial minus actual coins)  
pracs <- pracs %>%  
  rowwise() %>%  
  mutate(PRAC1_OP = PRAC1_1n-188)  
  
pracs <- pracs %>%  
  rowwise() %>%  
  mutate(PRAC2_OP = PRAC1_2n-141)  
  
pracs <- pracs %>%  
  rowwise() %>%  
  mutate(PRAC3_OP = PRAC1_3n-71)  
  
pracs <- pivot_longer(pracs, cols = c("PRAC1_OP", "PRAC2_OP", "PRAC3_OP"),  
                      names_to = "PRACTICE_TRIAL", values_to = "OPINIONDIFF")  
  
pracs$PRACTICE_TRIAL <- as.factor(pracs$PRACTICE_TRIAL)  
  
levels(pracs$PRACTICE_TRIAL)  
  
## Appears to be some extreme outliers.  
pracs %>%  
  group_by(AGEGROUP) %>%  
  identify_outliers(OPINIONDIFF) # 8 outliers...
```

Adjust outliers +/- 3SDs from M

```

# Put data back to wide

pracs <- spread(pracs, PRACTICE_TRIAL, OPINIONDIFF)

summarystats_op <- pracs %>%
  group_by(AGEGROUP) %>%
  get_summary_stats(PRAC1_OP, PRAC2_OP, PRAC3_OP, type = "mean_sd")

ungroup(pracs)

summarystats_op$M_PLUS_3SD <- summarystats_op$mean + (3*summarystats_op$sd)
summarystats_op$M_MINUS_3SD <- summarystats_op$mean - (3*summarystats_op$sd)

## Split into age groups
young <- subset(pracs, AGEGROUP == "YOUNG")

middle <- subset(pracs, AGEGROUP == "MIDDLE")

older <- subset(pracs, AGEGROUP == "OLDER")

```

Younger adults

```

# Change outliers to 5000 if above M+3*SDs
# Change outliers to -5000 if below M-3*SDs
# Easy to identify and sort out from other values

young <- young %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                          ifelse(PRAC1_OP > 402.353, 5000, PRAC1_OP)), # 1 identified
         PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                           ifelse(PRAC1_OP < -518.353, -5000, PRAC1_OP)))

young <- young %>%
  mutate(PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                          ifelse(PRAC2_OP > 449.818, 5000, PRAC2_OP)), # 1 identified
         PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP < -488.600, -5000, PRAC2_OP)),
         PRAC3_OP= ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP > 237.886, 5000, PRAC3_OP)), # 1 identified
         PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP < -259.460, -5000, PRAC3_OP)))

## Create new df without the outliers

young_exl_out <- select(young, PRAC1_OP, PRAC2_OP, PRAC3_OP)

young_exl_out <- young_exl_out %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                          ifelse(PRAC1_OP == 5000 | PRAC1_OP == -5000,
                                as.integer(NA), PRAC1_OP)),
         PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP == 5000 | PRAC2_OP == -5000,
                                 as.integer(NA), PRAC2_OP)),
         PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP == 5000 | PRAC3_OP == -5000,
                                 as.integer(NA), PRAC3_OP)))

```

```

PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                  ifelse(PRAC2_OP == 5000 | PRAC2_OP == -5000,
                         as.integer(NA), PRAC2_OP)),
PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                  ifelse(PRAC3_OP == 5000 | PRAC3_OP == -5000,
                         as.integer(NA), PRAC3_OP)))

youngmsd <- young_exl_out %>%
  get_summary_stats(PRAC1_OP, PRAC2_OP, PRAC3_OP, type = "mean_sd")

youngmsd$M_PLUS_3SD <- youngmsd$mean + (3*youngmsd$sd)

# Replace the values taken

young <- young %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                           ifelse(PRAC1_OP == 5000, 172.012, PRAC1_OP)),
         PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP == 5000, 213.740, PRAC2_OP)),
         PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP == 5000, 97.689, PRAC3_OP)))

```

Middle-aged adults

```

# Change outliers to 5000 if above M+3*SDs
# Change outliers to -5000 if below M-3*SDs
# Easy to identify and sort out from other values

middle <- middle %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                           ifelse(PRAC1_OP > 973.971, 5000, PRAC1_OP)), # 1 identified
         PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                           ifelse(PRAC1_OP < -1002.711, -5000, PRAC1_OP)),
         PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP > 1007.667, 5000, PRAC2_OP)), # 1 identified
         PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP < -972.705, -5000, PRAC2_OP)),
         PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP > 197.910, 5000, PRAC3_OP)), # 1 identified
         PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP < -211.836, -5000, PRAC3_OP)))

## Create new df without the outliers

middle_exl_out <- select(middle, PRAC1_OP, PRAC2_OP, PRAC3_OP)

middle_exl_out <- middle_exl_out %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                           ifelse(PRAC1_OP == 5000 | PRAC1_OP == -5000,
                                 as.integer(NA), PRAC1_OP)),
         PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP == 5000 | PRAC2_OP == -5000,
                                 as.integer(NA), PRAC2_OP)))

```

```

                        as.integer(NA), PRAC2_OP)),
PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                  ifelse(PRAC3_OP == 5000 | PRAC3_OP == -5000,
                         as.integer(NA), PRAC3_OP)))

middlemsd <- middle_exl_out %>%
  get_summary_stats(PRAC1_OP, PRAC2_OP, PRAC3_OP, type = "mean_sd")

middlemsd$M_PLUS_3SD <- middlemsd$mean + (3*middlemsd$sd)

# Replace the values taken

middle <- middle %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                          ifelse(PRAC1_OP == 5000, 144.812, PRAC1_OP)),
         PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP == 5000, 153.545, PRAC2_OP)),
         PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP == 5000, 81.057, PRAC3_OP)))

```

Older adults

```

# Change outliers to 5000 if above M+3*SDs
# Change outliers to -5000 if below M-3*SDs
# Easy to identify and sort out from other values

older<- older %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                          ifelse(PRAC1_OP > 86.898, 5000, PRAC1_OP)), # No outliers
        PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                           ifelse(PRAC1_OP < -259.734, -5000, PRAC1_OP)),
        PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP > 107.020, 5000, PRAC2_OP)), # 1 identified
        PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP < -218.474, -5000, PRAC2_OP)),
        PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP > 56.977, 5000, PRAC3_OP)), # 1 identified
        PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP < -113.495, -5000, PRAC3_OP)))

## Create new df without the outliers

older_exl_out <- select(older, PRAC1_OP, PRAC2_OP, PRAC3_OP)

older_exl_out <- older_exl_out %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                          ifelse(PRAC1_OP == 5000 | PRAC1_OP == -5000,
                                 as.integer(NA), PRAC1_OP)),
        PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP == 5000 | PRAC2_OP == -5000,

```

```

                        as.integer(NA), PRAC2_OP)),
PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                  ifelse(PRAC3_OP == 5000 | PRAC3_OP == -5000,
                         as.integer(NA), PRAC3_OP)))

oldermsd <- older_exl_out %>%
  get_summary_stats(PRAC1_OP, PRAC2_OP, PRAC3_OP, type = "mean_sd")

oldermsd$M_PLUS_3SD <- oldermsd$mean + (3*oldermsd$sd)

# Replace the values taken

older <- older %>%
  mutate(PRAC1_OP = ifelse(is.na(PRAC1_OP), NA,
                           ifelse(PRAC1_OP == 5000, 86.898, PRAC1_OP)),
         PRAC2_OP = ifelse(is.na(PRAC2_OP), NA,
                           ifelse(PRAC2_OP == 5000, 71.314, PRAC2_OP)),
         PRAC3_OP = ifelse(is.na(PRAC3_OP), NA,
                           ifelse(PRAC3_OP == 5000, 29.254, PRAC3_OP)))

```

Combine adjusted value dfs

```

# Combine the age group dfs together

all_ages <- young %>% full_join(middle)

all_ages <- all_ages %>% full_join(older)

pracs <- all_ages

```

Opinion difference between age groups

```

# Put data back into long form
pracs <- pivot_longer(pracs, cols = c("PRAC1_OP", "PRAC2_OP", "PRAC3_OP"),
                      names_to = "PRACTICE_TRIAL", values_to = "OPINIONDIFF")

pracs$PRACTICE_TRIAL <- as.factor(pracs$PRACTICE_TRIAL)

levels(pracs$PRACTICE_TRIAL)

## Analyses

op.aov <- anova_test(
  data = pracs, dv = OPINIONDIFF, wid = ID,
  between = AGEGROUP,
  within = PRACTICE_TRIAL,
  effect.size = "pes"
)

```

```

get_anova_table(op.aov)

pracs %>% group_by(PRACTICE_TRIAL) %>%
  get_summary_stats(OPINIONDIFF, type = "mean_sd")

ungroup(pracs)

# Pairwise comparisons between practice trials

practical <- pracs %>%
  pairwise_t_test(
    OPINIONDIFF ~ PRACTICE_TRIAL, paired = TRUE,
    p.adjust.method = "bonferroni"
  )
practical

# Pairwise comparisons between the age groups

pracages <- pracs %>%
  pairwise_t_test(
    OPINIONDIFF ~ AGEGROUP,
    p.adjust.method = "bonferroni"
  )

pracages

avg_grp_op <- pracs %>% group_by(AGEGROUP) %>%
  get_summary_stats(OPINIONDIFF, type = "mean_sd")

## Create descriptives table of opinion differences by age groups

avg_grp_op <- select(avg_grp_op, AGEGROUP, n, mean, sd)

avg_grp_op$AGEGROUP <- recode_factor(avg_grp_op$AGEGROUP, YOUNG = "Young adults",
                                       MIDDLE = "Middle-aged adults",
                                       OLDER = "Older adults")

## change the descriptives df to 2 decimal places
avg_grp_op$mean <- format(round(avg_grp_op$mean, 2))
avg_grp_op$sd <- format(round(avg_grp_op$sd, 2))

avg_grp_op_table <- flextable(avg_grp_op)

# Make new column names
avg_grp_op_table <- avg_grp_op_table %>%

  set_header_labels(          # Rename the columns in original header row
    AGEGROUP = "Age group",
    n = "N",
    mean = "M",
    sd = "SD")

```

```

avg_grp_op_table <- align(avg_grp_op_table, align = "center", j = 2:4,
                           part = "all")

avg_grp_op_table %>% autofit() # Auto fits the columns

# save_as_docx("age group averages" = avg_grp_op_table, path = "file.docx")

```

Check age group differences

Depressive symptom score

```

levels(full_ja_df$AGEGROUP)

ungroup(full_ja_df)

dep_diff <- full_ja_df %>%
  anova_test(DASS~AGEGROUP)

dep_diff

full_ja_df %>% tukey_hsd(DASS~AGEGROUP)

dep_diff_lm <- lm(DASS~AGEGROUP, data = full_ja_df)

summary(dep_diff_lm) # Less depressive symptoms for older adults

full_ja_df %>%
  group_by(AGEGROUP) %>%
  get_summary_stats(DASS)

```

Pre-Confidence

```

pre_conf_diff <- full_ja_df %>%
  anova_test(PRE_CONFIDENCEn~AGEGROUP)

pre_conf_diff

pre_conf_diff_lm <- lm(PRE_CONFIDENCEn~AGEGROUP, data = full_ja_df)

summary(pre_conf_diff_lm) # No differences

```

Post-Confidence

```
post_conf_diff <- full_ja_df %>%
  anova_test(POST_CONFIDENCEn~AGEGROUP)

post_conf_diff

post_conf_diff_lm <- lm(POST_CONFIDENCEn~AGEGROUP, data = full_ja_df)

summary(post_conf_diff_lm) # Difference between young and older - older less confident

estimates(post_conf_diff_lm)

## Error in estimates(post_conf_diff_lm): could not find function "estimates"
```

Perceived Advice Accuracy

```
advice_acc_diff <- full_ja_df %>%
  anova_test(PERCEIVED_ADVICE_ACCURACYn~AGEGROUP)

advice_acc_diff

advice_acc_diff_lm <- lm(PERCEIVED_ADVICE_ACCURACYn~AGEGROUP, data = full_ja_df)

summary(advice_acc_diff_lm) # Difference between young and older -
#Older perceive the advice as less accurate

estimates(advice_acc_diff_lm)

## Error in estimates(advice_acc_diff_lm): could not find function "estimates"
```

Average Fluid IQ

```
fluidiq_diff <- full_ja_df %>%
  anova_test(AVFLUIDIQ~AGEGROUP)

fluidiq_diff

fluidiq_diff_lm <- lm(AVFLUIDIQ~AGEGROUP, data = full_ja_df)

summary(fluidiq_diff_lm) # No difference

visualize(fluidiq_diff_lm)

## Error in visualize(fluidiq_diff_lm): could not find function "visualize"
```

Mixed Model Analyses

Hypothesis 1: With increasing age, and increasing depressive symptoms, there would be greater advice-taking.

```
# Get dataframe ready

ja_df1 <- full_ja_df

ja_df1 <- pivot_longer(ja_df1, cols = c("JAR01_WOA":"JAR12_WOA"),
                       names_to = "JAR", values_to = "WOA")

ja_df1$JAR <- as.factor(ja_df1$JAR)

ja_df1$ID <- as.factor(ja_df1$ID)

## Center age

ja_df1 <- ja_df1 %>% mutate(GMC_AGE = scale(AGE, scale = FALSE))

## Center DASS

ja_df1 <- ja_df1 %>% mutate(GMC_DASS = scale(DASS, scale = FALSE))

## Basic model with predictors

model1 <- lmer(WOA~GMC_AGE + GMC_DASS + (1|ID), data = ja_df1,
                control=lmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))

# Interaction model

model2 <- lmer(WOA~GMC_AGE*GMC_DASS + (1|ID), data = ja_df1,
                control=lmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))

# Compare main effects and interaction models
anova(model1, model2)

# Retain model 1
summary(model1)
```

Check assumptions

```
#Linearity of relationships
linearity_model1 <- plot_model(model1, type = "resid", show.data = TRUE)
linearity_model1
```



```
#Normality of residuals
```

```
model1_diagnostics <- visualize(model1, plot = "residuals")
```

```
## Error in visualize(model1, plot = "residuals"): could not find function "visualize"
```

```
model1_diagnostics
```

```
## Error in eval(expr, envir, enclos): object 'model1_diagnostics' not found
```

```
tab_model(model1)
```

Table of model1

```
tab_model(model1,
  show.re.var = FALSE,
  show.icc = FALSE,
  show.se = TRUE,
  show.r2 = FALSE,
  show.ci = 0.95,
  string.ci = "95% CI",
  p.val = "satterthwaite",
```

```

collapse.se = TRUE,
string.est = "Estimate (SE)",
pred.labels = c("Intercept", "Age"),
dv.labels = c("Weight of Advice"),
file = "model1.html")

```

Emotion regulators

Create variable

```

ja_df1$EMO_REGn <- as.factor(ja_df1$EMO_REGn)
ja_df2 <- ja_df1 %>%
  rowwise() %>%
  mutate(EMO_REGULATOR = ifelse(is.na(EMO_REG_EFFORT1n) & is.na(EMO_REG_EFFORT2n),
                                 "NO", "YES"))

```

LMM Analysis

```

# Age, DSS, Emotion Regulation

model3 <- lmer(WOA~GMC_AGE + GMC_DASS + EMO_REGULATOR + (1|ID), data = ja_df2,
                control=lmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))

model4 <- lmer(WOA~GMC_DASS + GMC_AGE*EMO_REGULATOR + (1|ID), data = ja_df2,
                control=lmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))

anova(model3, model4) # Retain model 3

model5 <- lmer(WOA~GMC_AGE + GMC_DASS*EMO_REGULATOR + (1|ID), data = ja_df2,
                control=lmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))

anova(model3, model5)

# Retain model 5

model6 <- lmer(WOA~ EMO_REGULATOR + GMC_AGE*GMC_DASS + (1|ID), data = ja_df2,
                control=lmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))

anova(model3, model6)

model7 <- lmer(WOA~GMC_AGE*GMC_DASS*EMO_REGULATOR + (1|ID), data = ja_df2,
                control=lmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))

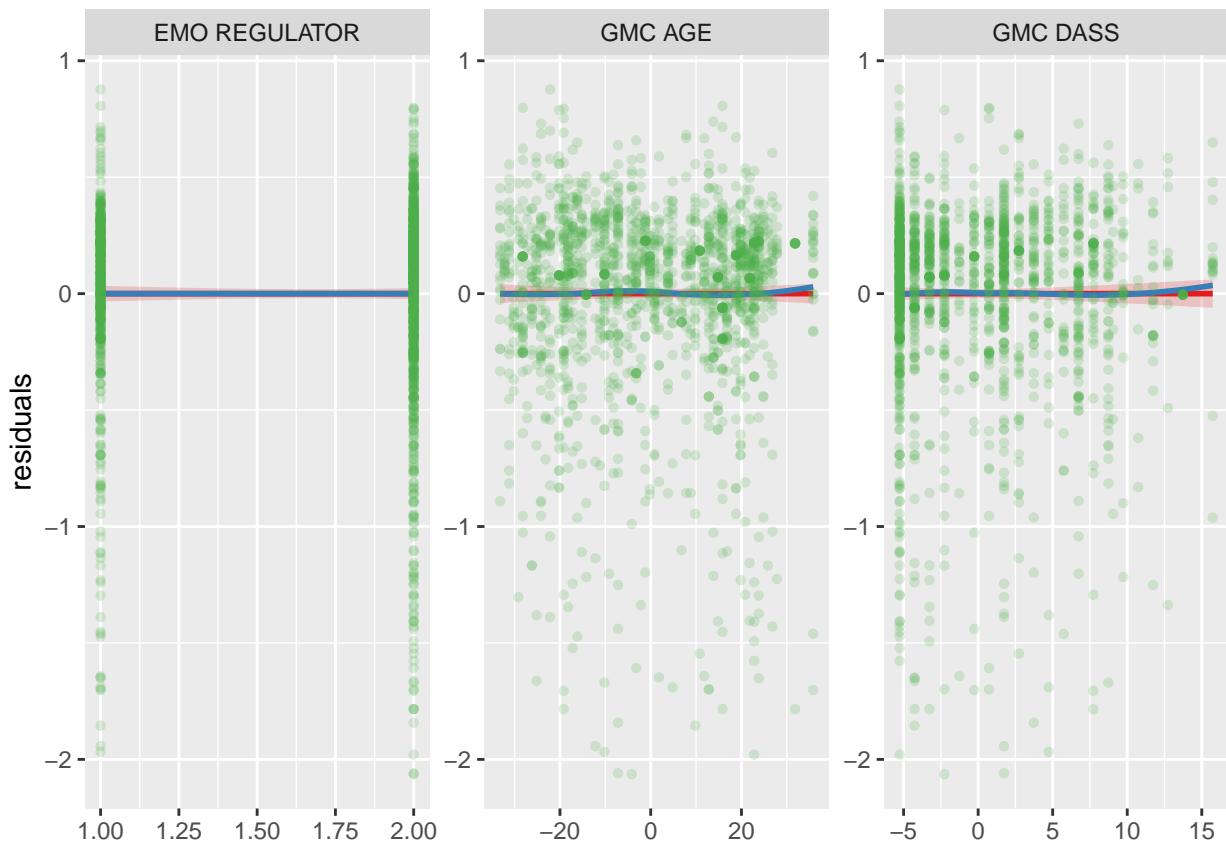
anova(model5, model7)

summary(model5) ### Significant depressive symptoms and emotion regulation interaction

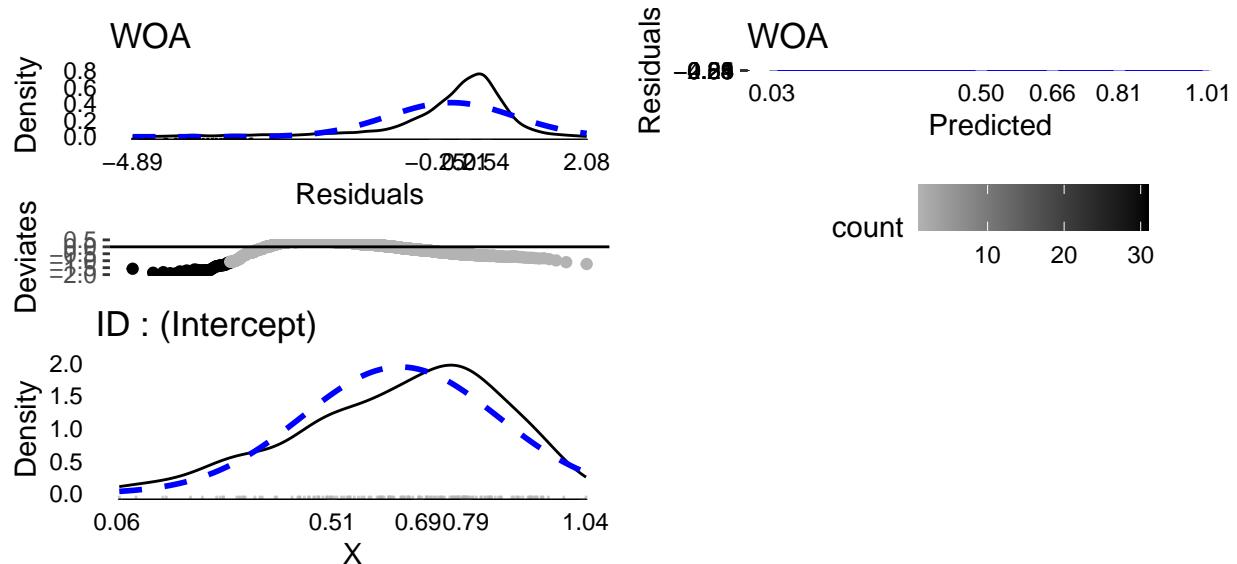
```

Check assumptions

```
# Linearity of relationships  
  
linearity_model5 <- plot_model(model5, type = "resid", show.data = TRUE)  
linearity_model5
```



```
# Normality of residuals  
  
model5_diagnostics1 <- visualize(model5, plot = "residuals")  
  
## Error in visualize(model5, plot = "residuals"): could not find function "visualize"  
  
model5_diagnostics1  
  
## Error in eval(expr, envir, enclos): object 'model5_diagnostics1' not found  
  
tab_model(model5)  
  
model5_diagnostics <- modelDiagnostics(model5, ev.perc = .001)  
  
plot(model5_diagnostics, ask = FALSE, ncol = 2, nrow = 3)
```



```
# Try a robust model
```

Robust lmm

```
model5_robust <- rlmmer(WOA~GMC_AGE + GMC_DASS*EMO_REGULATOR + (1|ID),
                           data = ja_df2, control=lmerControl(optimizer="bobyqa",
                           optCtrl=list(maxfun=2e5)))

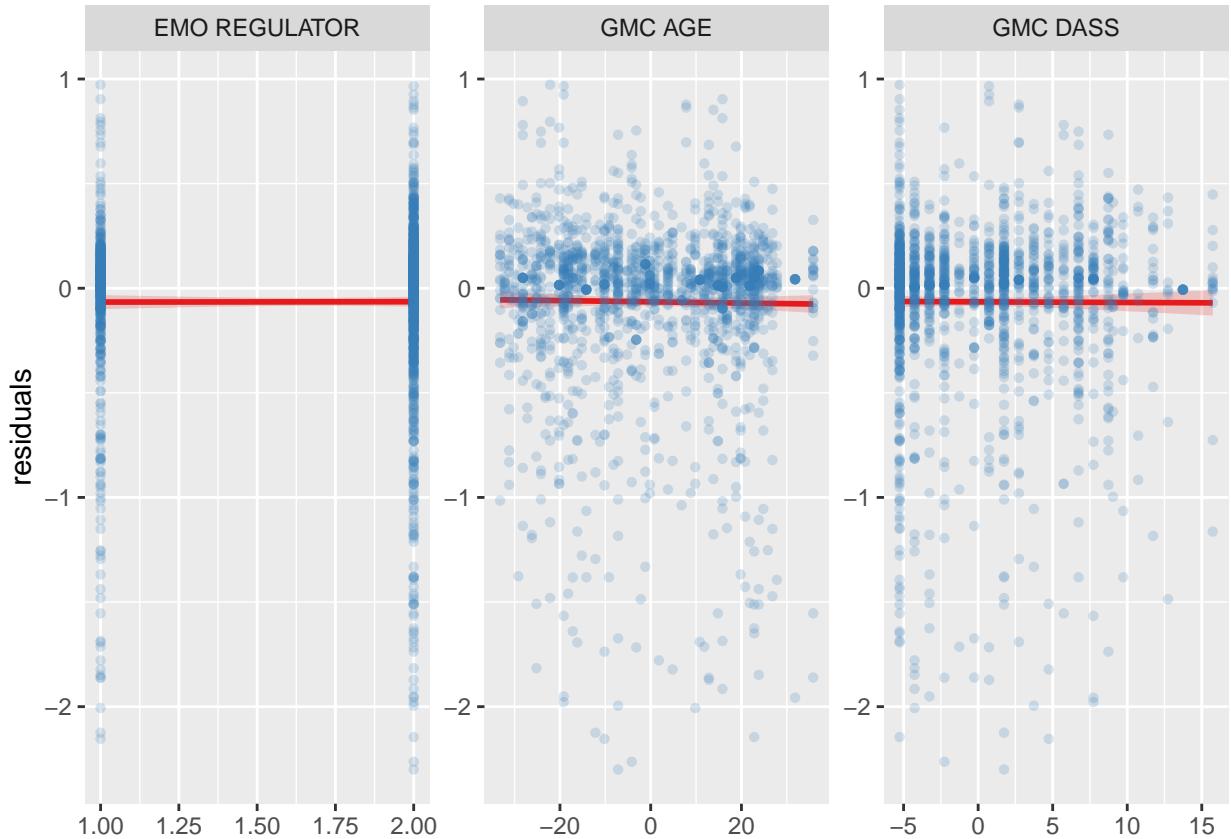
summary(model5_robust)

summary(model5)
```

Diagnostics on robust lmm

```
tab_model(model5_robust)

plot_model(model5_robust, type = "resid", show.data = TRUE, show.loess = FALSE)
```



```
# Use the robust model given normality violations
```

Visualisations and report for robust model

```
## Tab model report
tab_model(model5_robust,
          show.re.var = FALSE,
          show.icc = TRUE, #don't show ICCs
          show.se = TRUE,
          show.ci = 0.95,
          string.ci = "95% CI",
          collapse.se = TRUE,
          string.est = "Estimate (SE)",
          pred.labels = c("Intercept", "Age", "Depressive Symptom Score",
                         "Emotion Regulation Used (Yes)",
                         "Depressive Symptom Score*Emotion Regulation Used (Yes)"),
          dv.labels = c("Weight of Advice"),
          file = "model5_Robust.html")

## Visualisation of model
plot_model(model5_robust)
```



Post hoc on interaction

```

mod5_emm <- emtrends(model5_robust, var = "GMC_DASS",
                      pairwise~GMC_DASS*EMO_REGULATOR)

mod5_emm

# Visualisation

## Make apatheme

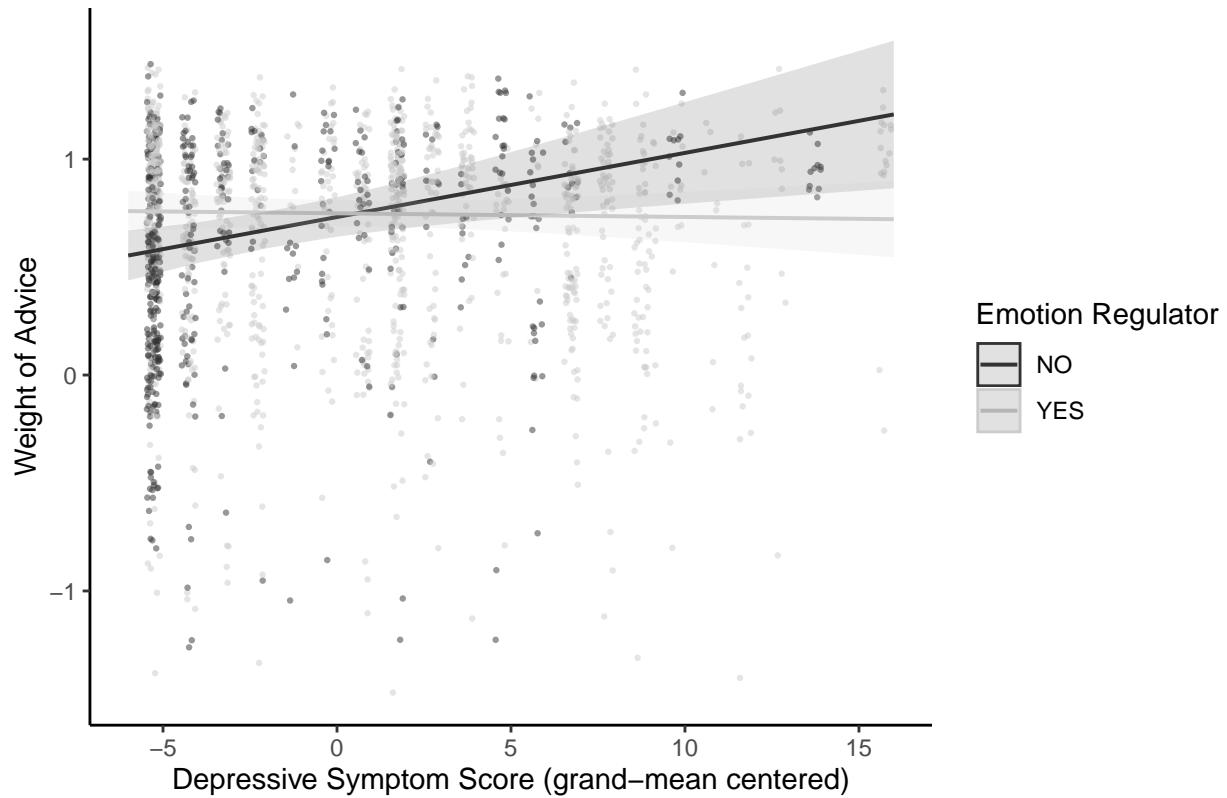
apatheme=theme_bw()+
  theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank(),
        panel.border=element_blank(),
        axis.line=element_line())

plot_model5_int <- plot_model(model5_robust,
                               type = "pred",
                               colors = "gs",
                               terms = c("GMC_DASS", "EMO_REGULATOR"),
                               title = " ",
                               axis.title = c("Depressive Symptom Score (grand-mean centered)",
```

```

  "Weight of Advice"),
legend.title = "Emotion Regulator",
show.data = TRUE, dot.size = 0.8, jitter = 0.2,
ci.lvl = 0.95) + apatheme
plot_model5_int

```



Difference between age groups and emo regulation type

Chi square analysis

```

# Create categorical variable

# Rename emotion regulation effort 1 to effort in change
ja_df2$EMO_CHNG EFFORT <- ja_df2$EMO_REG EFFORT1n

# Rename emotion regulation effort 2 to effort in acceptance
ja_df2$EMO_ACPT EFFORT <- ja_df2$EMO_REG EFFORT2n

# Use case_when instead of ifelse

ja_df3 <- ja_df2 %>%

```

```

mutate(EMO_REG_TYPE = case_when(EMO_CHNG EFFORT >= 0 ~ "CHANGE",
                                EMO_ACPT EFFORT >= 0 ~ "ACCEPT",
                                TRUE ~ "NONE"))

# Set as a factor

ja_df3$EMO_REG_TYPE <- as.factor(ja_df3$EMO_REG_TYPE)

levels(ja_df3$EMO_REG_TYPE)

ja_df3$EMO_REG_TYPE <- relevel(ja_df3$EMO_REG_TYPE, "NONE")

# Create one emotion regulation effort variable

ja_df3 <- ja_df3 %>%
  mutate(EMO_REG EFFORT = case_when(is.na(EMO_CHNG EFFORT) & is.na(EMO_ACPT EFFORT) ~ NA_real_,
                                    !is.na(EMO_CHNG EFFORT) & is.na(EMO_ACPT EFFORT) ~
                                      as.numeric(EMO_CHNG EFFORT),
                                    is.na(EMO_CHNG EFFORT) & !is.na(EMO_ACPT EFFORT) ~
                                      as.numeric(EMO_ACPT EFFORT),
                                    TRUE ~ 0))

# Center emotion regulation effort variable

# manually get the mean
emo_effort_m <- ja_df3 %>%
  get_summary_stats(EMO_REG EFFORT, type = "mean_sd")

ja_df3 <- ja_df3 %>% mutate(GMC_EMO_REG EFFORT = (EMO_REG EFFORT -
                                                       emo_effort_m$mean))

# Chi Square analysis of age groups by each emotion regulation category

# Split by emo regulation types

chi_change <- ja_df4 %>% filter(EMO_CHNG EFFORT >= 1)

## Error in filter(., EMO_CHNG EFFORT >= 1): object 'ja_df4' not found

chi_accept <- ja_df4 %>% filter(EMO_ACPT EFFORT >= 1)

## Error in filter(., EMO_ACPT EFFORT >= 1): object 'ja_df4' not found

chi_none <- ja_df4 %>% filter(EMO_REGULATOR == "NO")

## Error in filter(., EMO_REGULATOR == "NO"): object 'ja_df4' not found

# Change
chi_change <- chi_change %>%
  group_by(ID)

```

```

## Error in group_by(., ID): object 'chi_change' not found

chi_change_test <- chisq.test(table(chi_change$AGEGROUP))

## Error in table(chi_change$AGEGROUP): object 'chi_change' not found

chi_change_test

## Error in eval(expr, envir, enclos): object 'chi_change_test' not found

#calculating frequencies using pro.table

prop.table(table(chi_change$AGEGROUP))

## Error in table(chi_change$AGEGROUP): object 'chi_change' not found

# Accept
chi_accept <- chi_accept %>%
  group_by(ID)

## Error in group_by(., ID): object 'chi_accept' not found

chi_accept_test <- chisq.test(table(chi_accept$AGEGROUP))

## Error in table(chi_accept$AGEGROUP): object 'chi_accept' not found

chi_accept_test

## Error in eval(expr, envir, enclos): object 'chi_accept_test' not found

#calculating frequencies using pro.table

prop.table(table(chi_accept$AGEGROUP))

## Error in table(chi_accept$AGEGROUP): object 'chi_accept' not found

# None

chi_none <- chi_none %>%
  group_by(ID)

## Error in group_by(., ID): object 'chi_none' not found

chi_none_test <- chisq.test(table(chi_none$AGEGROUP))

## Error in table(chi_none$AGEGROUP): object 'chi_none' not found

```

```

chi_none_test

## Error in eval(expr, envir, enclos): object 'chi_none_test' not found

#calculating frequencies using prop.table

prop.table(table(chi_none$AGEGROUP))

## Error in table(chi_none$AGEGROUP): object 'chi_none' not found

```

Correlation

Hypothesis 2: Depressive symptoms in older age either associated directly with a decline in cognitive ability, or via increased emotion regulation.

###In both cases, reduced cognitive capacity was expected to be associated with increased advice-taking

Prepare data for correlations

```

# Pivot back wider
ja_df5 <- spread(ja_df4, JAR, WOA)

## Error in spread(ja_df4, JAR, WOA): object 'ja_df4' not found

# Create Grand Mean Centered emo effort scores

ja_df5 <- ja_df5 %>%
  mutate(GMC_EMO_CHNG EFFORT = scale(EMO_CHNG EFFORT, center = TRUE, scale = FALSE))

## Error in mutate(., GMC_EMO_CHNG EFFORT = scale(EMO_CHNG EFFORT, center = TRUE, : object 'ja_df5' not
## found

ja_df5 <- ja_df5 %>%
  mutate(GMC_EMO_ACPT EFFORT = scale(EMO_ACPT EFFORT, center = TRUE, scale = FALSE))

## Error in mutate(., GMC_EMO_ACPT EFFORT = scale(EMO_ACPT EFFORT, center = TRUE, : object 'ja_df5' not
## found

# Make average WOA for each person

ja_df5 <- ja_df5 %>%
  rowwise() %>%
  mutate(AVG_WOA = mean(c(JAR01_WOA, JAR02_WOA, JAR03_WOA, JAR04_WOA,
                         JAR05_WOA, JAR06_WOA, JAR07_WOA, JAR08_WOA,
                         JAR09_WOA, JAR10_WOA, JAR11_WOA, JAR12_WOA),
                        na.rm=TRUE))

## Error in rowwise(.): object 'ja_df5' not found

```

```

# Change factored levels of emotion regulator

ja_df5$EMO_REGULATOR <- gsub("NO", "0", ja_df5$EMO_REGULATOR)

## Error in is.factor(x): object 'ja_df5' not found

ja_df5$EMO_REGULATOR <- gsub("YES", "1", ja_df5$EMO_REGULATOR)

## Error in is.factor(x): object 'ja_df5' not found

ja_df5$EMO_REGULATOR <- as.numeric(ja_df5$EMO_REGULATOR)

## Error in eval(expr, envir, enclos): object 'ja_df5' not found

structure(ja_df5)

## Error in structure(ja_df5): object 'ja_df5' not found

# Select variables for correlation
# Split by emo regulation types

cor_df_chg <- ja_df5 %>% filter(EMO_CHNG EFFORT >= 1)

## Error in filter(., EMO_CHNG EFFORT >= 1): object 'ja_df5' not found

cor_df_chg <- select(cor_df_chg, AGEen, DASS, EMO_CHNG EFFORT, PRE_CONFIDENCEn,
                      POST_CONFIDENCEn,
                      PERCEIVED_ADVICE_ACCURACYn, AVFLUIDIQ, AVG_WOA)

## Error in select(cor_df_chg, AGEen, DASS, EMO_CHNG EFFORT, PRE_CONFIDENCEn, : object 'cor_df_chg' not found

cor_df_acpt <- ja_df5 %>% filter(EMO_ACPT EFFORT >= 1)

## Error in filter(., EMO_ACPT EFFORT >= 1): object 'ja_df5' not found

cor_df_acpt <- select(cor_df_acpt, AGEen, DASS, EMO_ACPT EFFORT, PRE_CONFIDENCEn,
                       POST_CONFIDENCEn,
                       PERCEIVED_ADVICE_ACCURACYn, AVFLUIDIQ, AVG_WOA)

## Error in select(cor_df_acpt, AGEen, DASS, EMO_ACPT EFFORT, PRE_CONFIDENCEn, : object 'cor_df_acpt' not found

```

Functions to create APA matrix table

```

# mat A matrix or data frame
# p A matrix with the same dimension as `mat`
# f A function to apply
#' @return `mat` with `f` applied to each cell where `p` is TRUE.
#' @examples
#' x <- rbind(c(1,2,3), c(4,5,6), c(7,8,9))
#' apply_if(x, upper.tri(x), function(x) x + 5)
#'
apply_if <- function(mat, p, f) {
  # Fill NA with FALSE
  p[is.na(p)] <- FALSE
  mat[p] <- f(mat[p])
  mat
}

#' @param mat an rcorr object or a double matrix
#' @param corrtype is either pearson or spearman. Will be passed into
#' Hmsic::rcorr if mat is not already an rcorr object
#' @return `mat` with stars appended for each level of significants
#' (p < 0.05, p < 0.01, p < 0.001)
apaCorr <- function(mat, corrtype = "pearson") {
  matCorr <- mat
  if (class(matCorr) != "rcorr") {
    matCorr <- rcorr(mat, type = corrtype)
  }

  # Add one star for each p < 0.05, 0.01, 0.001
  stars <- apply_if(round(matCorr$r, 2), matCorr$P < 0.05, function(x) paste0(x, "*"))
  stars <- apply_if(stars, matCorr$P < 0.01, function(x) paste0(x, "*"))
  stars <- apply_if(stars, matCorr$P < 0.001, function(x) paste0(x, "*"))
  # Put - on diagonal and blank on upper diagonal
  stars[upper.tri(stars, diag = T)] <- "-"
  stars[upper.tri(stars, diag = F)] <- ""
  n <- length(stars[1,])
  colnames(stars) <- 1:n
  # Remove _ and convert to title case
  row.names(stars) <- tools::toTitleCase(sapply(row.names(stars), gsub,
                                                pattern = "_", replacement = " "))
  # Add index number to row names
  row.names(stars) <- paste(paste0(1:n, ". "), row.names(stars))
  stars
}

```

```
rcorr(as.matrix(cor_df_chg), type = "spearman")
```

Correlation matrix for change emotions regulators

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f
```

```

chg_corr <- apaCorr(as.matrix(cor_df_chg), corrtype = "spearman")

## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f

chg_corr

## Error in eval(expr, envir, enclos): object 'chg_corr' not found

chg_corr_out <- kable(chg_corr, format = "html")

## Error in kable(chg_corr, format = "html"): object 'chg_corr' not found

# Save the table:
# readr::write_file(chg_corr_out, "chg_corr_out.html")

# Visualisation of some sig relationships

# Age and perceived advice accuracy
ggscatter(cor_df_chg, x = "AGEn", y = "PERCEIVED_ADVICE_ACCURACYn",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman",
          xlab = "Age", ylab = "Perceived Advice Accuracy")

## Error in ggscatter(cor_df_chg, x = "AGEn", y = "PERCEIVED_ADVICE_ACCURACYn", : object 'cor_df_chg' n

# Emotion regulation change effort and depressive symptoms
ggscatter(cor_df_chg, x = "EMO_CHNG EFFORT", y = "DASS",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman")

## Error in ggscatter(cor_df_chg, x = "EMO_CHNG EFFORT", y = "DASS", add = "reg.line", : object 'cor_df_ch

# Depressive symptoms and fluid IQ
ggscatter(cor_df_chg, x = "DASS", y = "AVFLUIDIQ",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman")

## Error in ggscatter(cor_df_chg, x = "DASS", y = "AVFLUIDIQ", add = "reg.line", : object 'cor_df_chg' n

# Advice accuracy and WOA
ggscatter(cor_df_chg, x = "PERCEIVED_ADVICE_ACCURACYn", y = "AVG_WOA",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman")

## Error in ggscatter(cor_df_chg, x = "PERCEIVED_ADVICE_ACCURACYn", y = "AVG_WOA", : object 'cor_df_chg'

```

```
cor_df_chg %>%  
  get_summary_stats()
```

```
## Error in is_grouped_df(data): object 'cor_df_chg' not found
```

```
rcorr(as.matrix(cor_df_acpt), type = "spearman")
```

Correlation matrix for accept emotions regulators

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f  
acpt_corr <- apaCorr(as.matrix(cor_df_acpt), corrtype = "spearman")
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f
```

```
acpt_corr
```

```
## Error in eval(expr, envir, enclos): object 'acpt_corr' not found
```

```
acpt_corr_out <- kable(acpt_corr, format = "html")
```

```
## Error in kable(acpt_corr, format = "html"): object 'acpt_corr' not found
```

```
# Save the table:  
# readr::write_file(acpt_corr_out, "acpt_corr_out.html")
```

```
# Visualisation of some sig relationships
```

```
# Age and depressive symptoms  
ggscatter(cor_df_acpt, x = "AGEn", y = "DASS",  
         add = "reg.line", conf.int = TRUE,  
         cor.coef = TRUE, cor.method = "spearman")
```

```
## Error in ggscatter(cor_df_acpt, x = "AGEn", y = "DASS", add = "reg.line", : object 'cor_df_acpt' not
```

```
# Depressive symptoms and perceived advice accuracy  
ggscatter(cor_df_acpt, x = "DASS", y = "PERCEIVED_ADVICE_ACCURACYn",  
         add = "reg.line", conf.int = TRUE,  
         cor.coef = TRUE, cor.method = "spearman")
```

```
## Error in ggscatter(cor_df_acpt, x = "DASS", y = "PERCEIVED_ADVICE_ACCURACYn", : object 'cor_df_acpt'
```

```
cor_df_acpt %>%  
  get_summary_stats()
```

```
## Error in is_grouped_df(data): object 'cor_df_acpt' not found
```

Correlation matrix for non-emotion regulators

```
# Change back to factored levels of emotion regulator

ja_df5$EMO_REGULATOR <- gsub("0", "NO", ja_df5$EMO_REGULATOR)

## Error in is.factor(x): object 'ja_df5' not found

ja_df5$EMO_REGULATOR <- gsub("1", "YES", ja_df5$EMO_REGULATOR)

## Error in is.factor(x): object 'ja_df5' not found

ja_df5$EMO_REGULATOR <- as.factor(ja_df5$EMO_REGULATOR)

## Error in is.factor(x): object 'ja_df5' not found

# Select variables for correlation

no_er_cor_df <- select(ja_df5, AGE_n, DASS, EMO_REGULATOR, PRE_CONFIDENCE_n, POST_CONFIDENCE_n,
PERCEIVED_ADVICE_ACCURACY_n, AVFLUIDIQ, AVG_WOA)

## Error in select(ja_df5, AGE_n, DASS, EMO_REGULATOR, PRE_CONFIDENCE_n, POST_CONFIDENCE_n, : object 'ja_d

no_er_cor_df <- no_er_cor_df %>% filter(EMO_REGULATOR == "NO")

## Error in filter(., EMO_REGULATOR == "NO"): object 'no_er_cor_df' not found

# Select variables again - exclude emo regulator

no_er_cor_df <- select(no_er_cor_df, AGE_n, DASS, PRE_CONFIDENCE_n, POST_CONFIDENCE_n,
PERCEIVED_ADVICE_ACCURACY_n, AVFLUIDIQ, AVG_WOA)

## Error in select(no_er_cor_df, AGE_n, DASS, PRE_CONFIDENCE_n, POST_CONFIDENCE_n, : object 'no_er_cor_df'

rcorr(as.matrix(no_er_cor_df), type = "spearman")

## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f

no_er_corr <- apaCorr(as.matrix(no_er_cor_df), corrtype = "spearman")

## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f

no_er_corr_out <- kable(no_er_corr, format = "html")

## Error in kable(no_er_corr, format = "html"): object 'no_er_corr' not found
```

```

# Save the table:
#readr::write_file(no_er_corr_out, "no_er_corr_out.html")

# Visualisation of some sig relationships

# Depressive symptoms and WOA
ggscatter(no_er_cor_df, x = "DASS", y = "AVG_WOA",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "spearman")

## Error in ggscatter(no_er_cor_df, x = "DASS", y = "AVG_WOA", add = "reg.line", : object 'no_er_cor_df'

no_er_cor_df %>%
  get_summary_stats()

## Error in is_grouped_df(data): object 'no_er_cor_df' not found

```

Exploratory analyses

Prep confidence & advice accuracy variables

```

# Grand mean center pre-confidence

ja_df6 <- ja_df5 %>% ungroup()

## Error in ungroup(.): object 'ja_df5' not found

ja_df6 <- ja_df6 %>%
  mutate(GMC_PRE_CONF = scale(PRE_CONFIDENCEn, center = TRUE, scale = FALSE))

## Error in mutate(., GMC_PRE_CONF = scale(PRE_CONFIDENCEn, center = TRUE, : object 'ja_df6' not found

ja_df6 <- ja_df6 %>%
  mutate(GMC_POST_CONF = scale(POST_CONFIDENCEn, center = TRUE, scale = FALSE))

## Error in mutate(., GMC_POST_CONF = scale(POST_CONFIDENCEn, center = TRUE, : object 'ja_df6' not found

ja_df6 <- ja_df6 %>%
  mutate(GMC_PERCEIVED_ACC = scale(PERCEIVED_ADVICE_ACCURACYn, center = TRUE,
                                    scale = FALSE))

## Error in mutate(., GMC_PERCEIVED_ACC = scale(PERCEIVED_ADVICE_ACCURACYn, : object 'ja_df6' not found

# Pivot longer by WOA and Jar

ja_df6 <- pivot_longer(ja_df6, cols = c("JAR01_WOA":"JAR12_WOA"),
                       names_to = "JAR", values_to = "WOA")

```

```

## Error in pivot_longer(ja_df6, cols = c("JAR01_WOA":"JAR12_WOA"), names_to = "JAR", : object 'ja_df6'

ja_df6$JAR <- as.factor(ja_df6$JAR)

## Error in is.factor(x): object 'ja_df6' not found

```

Make fluid IQ a GMC variable

```

ja_df6 <- ja_df6 %>%
  mutate(GMC_FLUIDIQ = scale(AVFLUIDIQ, center = TRUE, scale = FALSE))

```

```

## Error in mutate(., GMC_FLUIDIQ = scale(AVFLUIDIQ, center = TRUE, scale = FALSE)): object 'ja_df6' no

```

Exploratory LMM

```

# Baseline LMM will have age, DSS, Emo reg type, fluid IQ, pre-conf, and perceived advice accuracy.

model8 <- lmer(WOA~GMC_AGE + GMC_DASS + EMO_REG_TYPE + GMC_FLUIDIQ +
  GMC_PRE_CONF + GMC_PERCEIVED_ACC +
  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
  optCtrl=list(maxfun=2e5)))

```

```

## Error: bad 'data': object 'ja_df6' not found

```

```

model9 <- lmer(WOA~GMC_AGE*GMC_DASS + EMO_REG_TYPE + GMC_FLUIDIQ + GMC_PRE_CONF +
  GMC_PERCEIVED_ACC +
  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
  optCtrl=list(maxfun=2e5)))

```

```

## Error: bad 'data': object 'ja_df6' not found

```

```

anova(model8, model9) # Retain model 8

```

```

## Error in anova(model8, model9): object 'model8' not found

```

```

model10 <- lmer(WOA~GMC_AGE + GMC_DASS*EMO_REG_TYPE + GMC_FLUIDIQ + GMC_PRE_CONF +
  GMC_PERCEIVED_ACC +
  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
  optCtrl=list(maxfun=2e5)))

```

```

## Error: bad 'data': object 'ja_df6' not found

```

```

anova(model8, model10) # Retain model 10

```

```

## Error in anova(model8, model10): object 'model8' not found

```

```

model11 <- lmer(WOA~GMC_AGE + GMC_DASS*EMO_REG_TYPE*GMC_FLUIDIQ + GMC_PRE_CONF +
                  GMC_PERCEIVED_ACC +
                  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
                                                optCtrl=list(maxfun=2e5)))

## Error: bad 'data': object 'ja_df6' not found

anova(model10, model11) # Retain model 10

## Error in anova(model10, model11): object 'model10' not found

model12 <- lmer(WOA~GMC_AGE + GMC_DASS*EMO_REG_TYPE + GMC_FLUIDIQ + GMC_PRE_CONF*GMC_DASS +
                  GMC_PERCEIVED_ACC +
                  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
                                                optCtrl=list(maxfun=2e5)))

## Error: bad 'data': object 'ja_df6' not found

anova(model10, model12) # Retain model 10

## Error in anova(model10, model12): object 'model10' not found

model13 <- lmer(WOA~GMC_AGE*GMC_DASS*EMO_REG_TYPE + GMC_FLUIDIQ + GMC_PRE_CONF +
                  GMC_PERCEIVED_ACC +
                  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
                                                optCtrl=list(maxfun=2e5)))

## Error: bad 'data': object 'ja_df6' not found

anova(model10, model13) # Retain model 10

## Error in anova(model10, model13): object 'model10' not found

model14 <- lmer(WOA~GMC_AGE +GMC_DASS*EMO_REG_TYPE*GMC_PERCEIVED_ACC + GMC_FLUIDIQ +
                  GMC_PRE_CONF +
                  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
                                                optCtrl=list(maxfun=2e5)))

## Error: bad 'data': object 'ja_df6' not found

anova(model10, model14) # Retain model 10

## Error in anova(model10, model14): object 'model10' not found

model15 <- lmer(WOA~GMC_AGE + GMC_DASS*EMO_REG_TYPE + GMC_FLUIDIQ +
                  GMC_PRE_CONF*GMC_PERCEIVED_ACC +
                  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
                                                optCtrl=list(maxfun=2e5)))

## Error: bad 'data': object 'ja_df6' not found

```

```

anova(model10, model15) # Retain model 10

## Error in anova(model10, model15): object 'model10' not found

model16 <- lmer(WOA~GMC_AGE*GMC_PRE_CONF + GMC_DASS*EMO_REG_TYPE + GMC_FLUIDIQ +
  GMC_PERCEIVED_ACC +
  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
  optCtrl=list(maxfun=2e5)))

## Error: bad 'data': object 'ja_df6' not found

anova(model10, model16) # Retain model 10

## Error in anova(model10, model16): object 'model10' not found

model17 <- lmer(WOA~GMC_AGE*GMC_FLUIDIQ + GMC_PRE_CONF + GMC_DASS*EMO_REG_TYPE +
  GMC_FLUIDIQ + GMC_PERCEIVED_ACC +
  (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
  optCtrl=list(maxfun=2e5)))

## Error: bad 'data': object 'ja_df6' not found

anova(model10, model17) # Retain model 10

## Error in anova(model10, model17): object 'model10' not found

summary(model10)

## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method : 
```

Check assumptions

```

# Linearity of relationships
linearity_model10 <- plot_model(model10, type = "resid", show.data = TRUE)

## Error in insight::model_info(model): object 'model10' not found

linearity_model10

## Error in eval(expr, envir, enclos): object 'linearity_model10' not found

# Normality of residuals

model10_diagnostics1 <- visualize(model10, plot = "residuals")

## Error in visualize(model10, plot = "residuals"): could not find function "visualize"
```

```

model10_diagnostics1

## Error in eval(expr, envir, enclos): object 'model10_diagnostics1' not found

tab_model(model10)

## Error in tab_model(model10): object 'model10' not found

model10_diagnostics2 <- modelDiagnostics(model10, ev.perc = .001)

## Error in modelDiagnostics(model10, ev.perc = 0.001): object 'model10' not found

plot(model10_diagnostics2, ask = FALSE, ncol = 2, nrow = 3)

## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f

model10_robust <- rlm(WOA~GMC_AGE + GMC_DASS*EMO_REG_TYPE + GMC_FLUIDIQ +
                      GMC_PRE_CONF + GMC_PERCEIVED_ACC +
                      (1|ID), data = ja_df6, control=lmerControl(optimizer="bobyqa",
                                                    optCtrl=list(maxfun=2e5)))

```

Robust LMM

```

## Error: bad 'data': object 'ja_df6' not found

summary(model10_robust)

## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method : 

summary(model10)

## Error in h(simpleError(msg, call)): error in evaluating the argument 'object' in selecting a method : 

plot(model10_robust)

```

Diagnostics on robust lmm

```

## Error in h(simpleError(msg, call)): error in evaluating the argument 'x' in selecting a method for f

tab_model(model10_robust)

## Error in tab_model(model10_robust): object 'model10_robust' not found

```

```
plot_model(model10_robust, type = "resid", show.data = TRUE, show.loess = FALSE)
```

```
## Error in insight::model_info(model): object 'model10_robust' not found
```

```
# Use the robust model given normality violations
```

Visualisations and report for robust model

```
## Tab model report
tab_model(model10_robust,
          show.re.var = FALSE,
          show.icc = TRUE,
          show.se = TRUE,
          show.ci = 0.95,
          string.ci = "95% CI",
          collapse.se = TRUE,
          string.est = "Estimate (SE)",
          #pred.labels = c("Intercept", "Age", "Emotion Regulation (Accept)",
          #               # "Emotion Regulation (Change)",
          #               # "Depressive Symptom Score", "Pre-Confidence Score",
          #               # "Perceived Advice Accuracy",
          #               # "Emotion Regulation (Accept)*DSS",
          #               # "Emotion Regulation (Change)*DSS"),
          dv.labels = c("Weight of Advice"),
          file = "model10_Robust.html")
```

```
## Error in tab_model(model10_robust, show.re.var = FALSE, show.icc = TRUE, : object 'model10_robust' n
```

```
## Visualisation of model
plot_model(model10_robust)
```

```
## Error in insight::model_info(model): object 'model10_robust' not found
```

Post hoc on interaction

```
mod10_emm <- emtrends(model10_robust, var = "GMC_DASS",
                        pairwise ~ GMC_DASS * EMO_REG_TYPE)
```

```
## Error in emtrends(model10_robust, var = "GMC_DASS", pairwise ~ GMC_DASS * : object 'model10_robust' n
```

```
mod10_emm
```

```
## Error in eval(expr, envir, enclos): object 'mod10_emm' not found
```

```

## Visualisation

# Make apatheme

apatheme=theme_bw()+
  theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank(),
        panel.border=element_blank(),
        axis.line=element_line())

plot_model10_int <- plot_model(model10_robust,
                                colors = "gs",
                                type = "pred",
                                terms = c("GMC_DASS", "EMO_REG_TYPE"),
                                title = " ",
                                axis.title = c("Depressive Symptom Score (grand-mean centered)",
                                              "Weight of Advice"),
                                legend.title = "Emotion Regulation Method",
                                show.data = TRUE, dot.size = 0.8, jitter = 0.2,
                                ci.lvl = 0.95) + apatheme

```

```
## Error in insight::model_info(model): object 'model10_robust' not found
```

```
plot_model10_int
```

```
## Error in eval(expr, envir, enclos): object 'plot_model10_int' not found
```

Main effects visualisations

```

apatheme=theme_bw()+
  theme(panel.grid.major=element_blank(),
        panel.grid.minor=element_blank(),
        panel.border=element_blank(),
        axis.line=element_line())

# Pre-confidence plot
pre_conf_plot <- plot_model(model10_robust,
                             colors = "gs",
                             type = "pred",
                             terms = "GMC_PRE_CONF",
                             title = " ",
                             axis.title = c("Pre-Advice Confidence (grand-mean centered)",
                                           "Weight of Advice"),
                             show.data = TRUE, dot.size = 0.8, jitter = 0.2,
                             ci.lvl = 0.95) + apatheme

```

```
## Error in insight::model_info(model): object 'model10_robust' not found
```

```
pre_conf_plot

## Error in eval(expr, envir, enclos): object 'pre_conf_plot' not found

# Perceived advice accuracy plot

per_advice_acc_plot <- plot_model(model10_robust,
                                      colors = "gs",
                                      type = "pred",
                                      terms = "GMC_PERCEIVED_ACC",
                                      title = " ",
                                      axis.title = c("Perceived Advice Accuracy (grand-mean centered)",
                                                    "Weight of Advice"),
                                      show.data = TRUE, dot.size = 0.8, jitter = 0.2,
                                      ci.lvl = 0.95) + apatheme

## Error in insight::model_info(model): object 'model10_robust' not found

per_advice_acc_plot

## Error in eval(expr, envir, enclos): object 'per_advice_acc_plot' not found
```