

PTSD in the year following sexual assault: A meta-analysis of prospective studies

Syntax and Output (Appendix C)

This supplementary file generated by R markdown provides documentation of all statistical analyses conducted in the manuscript, “*PTSD in the year following sexual assault: A meta-analysis of prospective studies.*” Analyses were completed in R version 3.6.1 (2019-07-05).

Variables included in dataset

ID VARIABLES

- **SampleID**: ID variable indicating which effects are drawn from the same sample. Multiple citations may be reflected in a single sample-level ID.
- **citation**: Variable indicating the citation of the primary source article from which effects are drawn.
- **Effect**: ID variable indicating which observations correspond to the same effect (i.e., all observations from the same measure in the same sample over time will share an Effect ID)
- **obs_num**: Indicates the temporal order of observations at which data were obtained on a given measure in a given sample, with the first observation labeled as 1.

EFFECT SIZE VARIABLES

- **N**: Sample size specific to a given time point (reflecting attrition unless a completer analysis was presented). When N was not reported for a time point, we carried forward the last available N from a prior time point
- **n_PTSD**: Number of individuals who met study criteria for PTSD at the time point/observation as reported in the article
- **n_PTSD_calc**: Number of individuals who met study criteria for PTSD at the time point/observation as reported in the article or as calculated, when not available in the article. The n with PTSD was calculated by multiplying N by the proportion reported in the article and rounding to the nearest integer. We then checked this calculation by dividing it by N and comparing to the proportion reported in the article. When this check yielded a different proportion than reported in the article, we retained the unrounded n to 2 decimals.
- **prop_PTSD**: Proportion of individuals with PTSD at a given time point/observation, as reported in the source article. Not used in this syntax; included for reference only.
- **prop_raw**: Proportion of individuals with PTSD at a given time point/observation. If the number of individuals with PTSD was reported in the source article, the proportion of individuals with PTSD was obtained by dividing the number of individuals with PTSD by the observation-specific sample size. If the number of individuals with PTSD was calculated by coders, the proportion of individuals with PTSD was recorded as it was reported in the source article.
- **mean**: Mean score on PTSD measure (raw, not rescaled)
- **sd**: Standard deviation of scores on PTSD measure (raw, not rescaled)
- **PTSD_minimum_score_possible**: Minimum value of PTSD measure (used to rescale means)
- **PTSD_maximum_score_possible**: Maximum value of PTSD measure (used to rescale means)

- **prop_sentbyauthors:** prop_PTSD was obtained from unpublished data sent by study authors
- **mean_sentbyauthors:** mean was obtained from unpublished data sent by study authors
- **prop_is_unpub:** prop_PTSD was obtained from a source other than a peer-reviewed journal article
- **mean_is_unpub:** mean was obtained from a source other than a peer-reviewed journal article

TIME SINCE ASSAULT VARIABLES

- **actual_time_reported_in_article:** The post-assault timing of the observation as reported in the source article. Time is reported in as specific of units as were available in the source article (e.g., when reported in both days and months, days were recorded). Not used in this syntax; included for reference only.
- **months_exact:** Conversion of actual_time_reported_in_article to months. When actual_time_reported_in_article is expressed in a unit smaller than months, the value was converted to months using Google Calculator.
- **calc_months:** Conversion of months_exact to an integer using rounding (e.g., 1.4 -> 1, 1.5 -> 2)

STUDY METHODS VARIABLES

- **publication_type:** The type of source in which the study was published, even when unpublished effect data corresponding to that study were obtained from authors (e.g., journal article, dissertation)
- **country:** The country from which the sample was obtained
- **majority_recruitment_site:** The site from which the majority (at least 67%) of participants were recruited
- **PTSD_measure:** The name of the PTSD measure used to obtain the effect
- **prop_cutscore:** Was PTSD point prevalence determined by a cut score on the PTSD measure? (Y/N)
- **prop_clusters:** Was PTSD point prevalence determined by meeting Diagnostic Statistical Manual of Mental Disorders (DSM) diagnostic criteria (e.g., based on the number of symptoms in clusters B, C, and D in the DSM-IV; APA, 2000)? (Y/N)
- **DSM_version:** The DSM version on which the PTSD measure was based
- **interview_or_self_report:** Whether an interview (e.g., phone interview, structured clinical interview) or self-report measure was used to assess PTSD

SAMPLE AND ASSAULT CHARACTERISTICS VARIABLES

Note. When possible, we coded this information in reference to the baseline sample of sexually assaulted participants followed prospectively. When that information was not available, we coded this variable for the closest-possible sample (e.g., the sample approached about the study regardless of enrollment status).

- **percent_women:** The percent of participants who were women
- **mean_age:** Participant mean age
- **majority_race:** The majority (>67%) racial composition of samples in the United States
- **percent_prior_SA:** The percent of participants with a prior history of sexual assault
- **percent_vic_drugsalc:** The percent of participants presenting after a sexual assault involving alcohol/drug use by the survivor
- **percent_weapon:** The percent of participants presenting after a sexual assault involving weapon use by the perpetrator
- **percent_injured:** The percent of participants presenting after a sexual assault involving physical injury
- **percent_stranger:** The percent of participants presenting after a sexual assault involving a stranger perpetrator (i.e., unknown to the survivor)
- **percent_police_report:** The percent of participants who made a police report about the recent sexual assault

ATTRITION VARIABLES

- **baseline_timepoint**: Does this row represent the baseline time point? (Y/N) The baseline time point is the observation at which participants were enrolled and the baseline N was reported. This will not correspond to obs_num = 1 when studies first reported prevalence/severity of PTSD at a time point subsequent to enrollment.
- **baselineN**: The N at study enrollment
- **attrition**: The % reduction in N at the current assessment relative to baselineN. Attrition is missing for completer analyses and studies that enrolled new participants following baseline.
- **completer_analysis**: Did this study present results only for the subsample of participants who completed all assessments? (Y/N)

STUDY QUALITY

- **quality**: Rating of methodological quality of the study, rescaled to a 0-100 scale

Setup

Run this section before running any other sections. This syntax file has been set up such that all subsequent sections of code will run independently once the setup section has been run.

```
# Install packages
# Remove the #s before these lines of code if these packages have not been installed.
# install.packages("tidyverse") #for data wrangling
# install.packages("dplyr") #for recoding
# install.packages("descr") #for descriptive statistics
# install.packages("metafor") #for meta-analysis
# install.packages("ggplot2") #for plots
# install.packages("car") #for plots
# install.packages('knitr') #to include graphics in the R markdown file
# install.packages('pander') #to include tables in the R markdown file

# Load packages
library(tidyverse)
library(dplyr)
library(descr)
library(metafor)
library(ggplot2)
library(car)
library(knitr)
library(pander)

# Import data
prosptsd_long <- read.csv("Appendix B- Dataset.csv",na.strings=c("", " "))

# Compute variance of raw proportion
prosptsd_long <- escalc(measure="PR", xi = n_PTSD_calc,
                      ni = N, data=prosptsd_long, add=0,
                      var.names=c("prop_PTSD_raw_2", "prop_raw_v"))

# Compute arcsine square root transformed proportion
prosptsd_long <- escalc(measure="PAS", xi = n_PTSD_calc, ni = N, data=prosptsd_long, add=0,
                      var.names=c("prop_pas", "prop_pas_v"))
```

```

prosptsd_long$ci.ub.prop <- prosptsd_long$prop_raw + 1.96*sqrt(prosptsd_long$prop_raw_v)
prosptsd_long$ci.lb.prop <- prosptsd_long$prop_raw - 1.96*sqrt(prosptsd_long$prop_raw_v)
# Rescale means and variances for continuous measures
# Based on formula given by metafor creator here: https://bit.ly/2VItm9C
prosptsd_long$m_rescaled01 <- ((prosptsd_long$mean-prosptsd_long$PTSD_minimum_score_possible)
                               /((prosptsd_long$PTSD_maximum_score_possible
                                   -prosptsd_long$PTSD_minimum_score_possible))
prosptsd_long$v_rescaled01 <- ((prosptsd_long$sd)^2)/((prosptsd_long$N)
                               *(prosptsd_long$PTSD_maximum_score_possible
                                   -prosptsd_long$PTSD_minimum_score_possible)^2)
prosptsd_long$ci.ub.mean <- prosptsd_long$m_rescaled01
+ 1.96*sqrt(prosptsd_long$v_rescaled01)

```

```

## [1] NA NA NA NA 0.03417918 0.04240451
## [7] 0.04100617 0.02590124 NA NA NA NA
## [13] NA NA NA NA NA NA NA
## [19] 0.03833732 0.04188391 0.04443804 0.04521530 0.02741192 0.02659366
## [25] 0.02659366 0.01837348 0.01878013 NA NA NA
## [31] NA NA NA NA NA NA NA
## [37] NA NA 0.07172082 0.10873431 0.11416340 0.02995027
## [43] 0.04220227 NA NA NA NA NA 0.05499941
## [49] 0.05600412 0.03467835 0.04542143 0.03839566 0.04846025 0.05346245
## [55] 0.05132485 0.09582222 NA NA 0.05008889 NA
## [61] NA NA NA NA NA NA NA
## [67] 0.04682222 0.04683131 0.05904817 0.05293974 NA NA
## [73] NA 0.03327052 0.04894063 0.04910699 0.05566651 0.06555111
## [79] 0.09168576 0.07194477 NA NA NA 0.08665722
## [85] 0.10475095 0.05135282 0.08012285 0.04709553 0.06640765 0.05317936
## [91] 0.06635575 0.04496999 0.05358576 0.06231380 0.11070859 0.06155541
## [97] 0.07361076 0.07317730 0.04387928 0.06106610 0.05816742

```

```

prosptsd_long$ci.lb.mean <- prosptsd_long$m_rescaled01
- 1.96*sqrt(prosptsd_long$v_rescaled01)

```

```

## [1] NA NA NA NA -0.03417918
## [6] -0.04240451 -0.04100617 -0.02590124 NA NA
## [11] NA NA NA NA NA NA
## [16] NA NA NA -0.03833732 -0.04188391
## [21] -0.04443804 -0.04521530 -0.02741192 -0.02659366 -0.02659366
## [26] -0.01837348 -0.01878013 NA NA NA
## [31] NA NA NA NA NA NA
## [36] NA NA NA -0.07172082 -0.10873431
## [41] -0.11416340 -0.02995027 -0.04220227 NA NA
## [46] NA NA -0.05499941 -0.05600412 -0.03467835
## [51] -0.04542143 -0.03839566 -0.04846025 -0.05346245 -0.05132485
## [56] -0.09582222 NA NA -0.05008889 NA
## [61] NA NA NA NA NA NA
## [66] NA -0.04682222 -0.04683131 -0.05904817 -0.05293974
## [71] NA NA NA -0.03327052 -0.04894063
## [76] -0.04910699 -0.05566651 -0.06555111 -0.09168576 -0.07194477
## [81] NA NA NA -0.08665722 -0.10475095
## [86] -0.05135282 -0.08012285 -0.04709553 -0.06640765 -0.05317936

```

```
## [91] -0.06635575 -0.04496999 -0.05358576 -0.06231380 -0.11070859
## [96] -0.06155541 -0.07361076 -0.07317730 -0.04387928 -0.06106610
## [101] -0.05816742
```

```
## Create knots to test in linear-linear piecewise models
prosptsd_long <- prosptsd_long %>%
  mutate(
    #Model A: Knot at 1 month
    pre1m = ifelse(months_exact<=1, months_exact, 1),
    post1m= ifelse(months_exact>1, (months_exact-1), 0),
    #Model B: Knot at 2 months
    pre2m = ifelse(months_exact<=2, months_exact, 2),
    post2m= ifelse(months_exact>2, (months_exact-2), 0),
    #Model C: Knot at 3 months
    pre3m = ifelse(months_exact<=3, months_exact, 3),
    post3m= ifelse(months_exact>3, (months_exact-3), 0),
    #Model D: Knot at 4 months
    pre4m = ifelse(months_exact<=4, months_exact, 4),
    post4m= ifelse(months_exact>4, (months_exact-4), 0),
    #Model E: Knot at 5 months
    pre5m = ifelse(months_exact<=5, months_exact, 5),
    post5m= ifelse(months_exact>5, (months_exact-5), 0),
    #Model F: Knot at 6 months
    pre6m = ifelse(months_exact<=6, months_exact, 6),
    post6m= ifelse(months_exact>6, (months_exact-6), 0))

#Create datasets dropping any effects without variances
#(i.e., limit to data used in each model)
meansdata<-prosptsd_long[!is.na(prosptsd_long$v_rescaled01),]
propdata<-prosptsd_long[!is.na(prosptsd_long$prop_pas_v),]
fulldata <- rbind(meansdata, propdata)

### General data setup for descriptives
datafordescriptives <-fulldata[ !(fulldata$SampleID=="23"),]
#exclude ID23 as redundant with 18 and 22 for descriptives

#Create dataset with one row per SampleID for sample-level descriptives
descr <- datafordescriptives %>%
  group_by(SampleID) %>%
  mutate(countpersample = seq(n()),
         N = max(N)) %>%
  filter(countpersample == 1)
```

Study Quality

```
# General data setup for descriptives
# Exclude ID23 as redundant with 18 and 22 for descriptives
datafordescriptives <-fulldata[ !(fulldata$SampleID=="23"),]

# Create dataset with one row per SampleID for sample-level descriptives
descr <- datafordescriptives %>%
```

```

group_by(SampleID) %>%
mutate(countpersample = seq(n()),
       N = max(N)) %>%
filter(countpersample == 1)

# Run descriptives
summary(descr$quality)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    14.00  43.25   56.00   52.59  67.00   78.00

```

```
sd(descr$quality, na.rm=TRUE)
```

```
## [1] 18.9477
```

Descriptives

```
### Descriptives - Sample Characteristics
```

```
# Number of total samples
nrow(descr)
```

```
## [1] 22
```

```
# Number of total participants
sum(descr$N, na.rm=TRUE)
```

```
## [1] 2106
```

```
# Publication type
summary(descr$publication_type)
```

```
## Dissertation or thesis      Journal article      Other
##                2                19                1
```

```
## Proportion data
nrow(propdata) #number of effects
```

```
## [1] 75
```

```
propN <- propdata %>%
  group_by(SampleID) %>%
  summarize(N = max(N))
nrow(propN) #number of samples
```

```
## [1] 20
```

```
freq(propdata$prop_is_unpub,plot=FALSE) #unpublished effects
```

```
## propdata$prop_is_unpub
##      Frequency Percent
## N           49    65.33
## Y           26    34.67
## Total        75   100.00
```

```
# Mean data
```

```
nrow(meansdata) #number of effects
```

```
## [1] 57
```

```
meanN <- meansdata %>%
  group_by(SampleID) %>%
  summarize(N = max(N))
nrow(meanN) #number of samples
```

```
## [1] 17
```

```
freq(meansdata$mean_is_unpub,plot=FALSE) #unpublished effects
```

```
## meansdata$mean_is_unpub
##      Frequency Percent
## N           15    26.32
## Y           42    73.68
## Total        57   100.00
```

```
### Descriptives - Study Methods
```

```
# Country
```

```
summary(descr$country)
```

```
##      Brazil      Denmark      England      France South Africa
##           1           1           1           2           1
##           US
##           16
```

```
# Recruitment site
```

```
summary(descr$majority_recruitment_site)
```

```
##      Community Hospital/SANE/forensic nursing
##           2           14
##      Mixed (no majority)      Police
##           3           1
##      Rape crisis center
##           2
```

```

# Create dataset with PTSD methods information summarized per sample
descrPTSDmeasures <- prosptsd_long %>%
  mutate(interview = dplyr::recode(interview_or_self_report,
                                   "Interview"=1, "Self-report"=0),
         dsm= dplyr::recode(DSM_version, "III"=3, "III-R"=3.5, "IV"=4)) %>%
  group_by(SampleID) %>%
  summarise(propInterview = mean(interview, na.rm=TRUE),
            dsm=min(dsm)) %>%
  filter(SampleID != 23) #ID 23 is redundant with 18 & 22

# Interview or self report
freq(descrPTSDmeasures$propInterview,plot=FALSE)

```

```

## descrPTSDmeasures$propInterview
##      Frequency Percent
## 0           8   36.36
## 1          14   63.64
## Total        22  100.00

```

```

# DSM version
freq(descrPTSDmeasures$dsm, plot=FALSE)

```

```

## descrPTSDmeasures$dsm
##      Frequency Percent
## 3           4   18.18
## 3.5          4   18.18
## 4           14   63.64
## Total        22  100.00

```

```

# Create dataset with PTSD methods information summarized per sample (Proportion only)
descrPTSDmeasures_prop <- propdata %>%
  mutate(cutscore_r = dplyr::recode(prop_cutscore, "N"=0, "Y"=1),
         diagnostic_r = dplyr::recode(prop_clusters,"N"=0, "Y"=1)) %>%
  group_by(SampleID) %>%
  summarise(anicutscore = max(cutscore_r, na.rm=TRUE),
            anydiagnostic = max(diagnostic_r, na.rm=TRUE)) %>%
  filter(SampleID != 18 & SampleID != 22) #ID 23 is redundant with 18 & 22,
                                         #but only 23 has proportion data

```

```

# Cut score
xtabs(~descrPTSDmeasures_prop$anicutscore+descrPTSDmeasures_prop$anydiagnostic)

```

```

##                                descrPTSDmeasures_prop$anydiagnostic
## descrPTSDmeasures_prop$anicutscore  0  1
##                                     0  0 17
##                                     1  2  1

```

```

### Descriptives - Sample Characteristics

```

```

# Race
summary(descr$majority_race)

```



```
## African American Mixed (no majority) White
## 3 6
## NA's
## 7
```

```
# Age
summary(descr$mean_age)
```

```
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## 13.50 25.16 26.59 26.60 30.02 35.60 2
```

```
sd(descr$mean_age, na.rm=TRUE)
```

```
## [1] 5.438857
```

```
# Percent women
freq(descr$percent_women, plot=FALSE)
```

```
## descr$percent_women
## Frequency Percent Valid Percent
## 66.2 1 4.545 4.762
## 84.93 1 4.545 4.762
## 94.81 1 4.545 4.762
## 96.8 1 4.545 4.762
## 100 17 77.273 80.952
## NA's 1 4.545
## Total 22 100.000 100.000
```

```
# Percent with Prior Sexual Assault
summary(descr$percent_prior_SA)
```

```
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## 10.29 34.53 41.67 43.66 53.97 76.92 13
```

```
sd(descr$percent_prior_SA, na.rm=TRUE)
```

```
## [1] 19.20654
```

```
### Descriptives - Sexual Assault Characteristics
```

```
# Assault involved alcohol/drug use by survivor
summary(descr$percent_vic_drugsalc)
```

```
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## 25.84 28.18 37.60 39.72 48.91 60.39 15
```

```
sd(descr$percent_vic_drugsalc, na.rm=TRUE)
```

```
## [1] 14.69159
```

```
# Percent stranger perpetrators
summary(descr$percent_stranger)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##  19.84  28.57   38.35   41.16  54.28   76.12     9
```

```
sd(descr$percent_stranger, na.rm=TRUE)
```

```
## [1] 17.21216
```

```
# Assault involved weapon
summary(descr$percent_weapon)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##   7.70  17.65   24.14   25.68  39.46   39.46    17
```

```
sd(descr$percent_weapon, na.rm=TRUE)
```

```
## [1] 13.87362
```

```
# Assault involved injury
summary(descr$percent_injured)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##  45.24  53.11   58.27   63.60  67.75  100.00    15
```

```
sd(descr$percent_injured, na.rm=TRUE)
```

```
## [1] 18.24938
```

```
# Police report made
summary(descr$percent_police_report)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##  21.01  63.72   90.52   76.14  97.63  100.00    16
```

```
sd(descr$percent_police_report, na.rm=TRUE)
```

```
## [1] 31.77202
```

Publication bias

```
# Examining frequency by months
freq(meansdata$calc_months, plot = FALSE) # 1, 3, and 6 months have at least 10 effects
```

```
## meansdata$calc_months
##      Frequency Percent
## 0          7  12.281
## 1         13  22.807
## 2          5   8.772
## 3         12  21.053
## 4          2   3.509
## 5          1   1.754
## 6         11  19.298
## 12         6  10.526
## Total        57 100.000
```

```
freq(propdata$calc_months, plot = FALSE) # 1, 2, 3, and 6 months have at least 10 effects
```

```
## propdata$calc_months
##      Frequency Percent
## 0          7   9.333
## 1         18  24.000
## 2         12  16.000
## 3         14  18.667
## 4          4   5.333
## 5          1   1.333
## 6         11  14.667
## 8          1   1.333
## 9          1   1.333
## 12         6   8.000
## Total       75 100.000
```

```
# Making datasets containing only months with at least 10 effects
```

```
meansdata1month <-meansdata[ (meansdata$calc_months==1),]
meansdata3month <-meansdata[ (meansdata$calc_months==3),]
meansdata6month <-meansdata[ (meansdata$calc_months==6),]
propdata1month <-propdata[ (propdata$calc_months==1),]
propdata3month <-propdata[ (propdata$calc_months==3),]
propdata6month <-propdata[ (propdata$calc_months==6),]
```

```
# Bivariate tests comparing published to unpublished effects
```

```
t.test(meansdata1month$m_rescaled01-meansdata1month$mean_is_unpub) # ns
```

```
##
## Welch Two Sample t-test
##
## data: meansdata1month$m_rescaled01 by meansdata1month$mean_is_unpub
## t = 0.27695, df = 1.6514, p-value = 0.8125
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.5077253  0.5636926
## sample estimates:
## mean in group N mean in group Y
##      0.4756127      0.4476291
```

```
t.test(meansdata3month$m_rescaled01~meansdata3month$mean_is_unpub) # ns
```

```
##  
## Welch Two Sample t-test  
##  
## data: meansdata3month$m_rescaled01 by meansdata3month$mean_is_unpub  
## t = -0.88579, df = 6.5288, p-value = 0.4072  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.2591334 0.1194037  
## sample estimates:  
## mean in group N mean in group Y  
## 0.3034604 0.3733252
```

```
t.test(meansdata6month$m_rescaled01~meansdata6month$mean_is_unpub) # ns
```

```
##  
## Welch Two Sample t-test  
##  
## data: meansdata6month$m_rescaled01 by meansdata6month$mean_is_unpub  
## t = 0.48127, df = 1.4367, p-value = 0.6934  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.5733553 0.6665600  
## sample estimates:  
## mean in group N mean in group Y  
## 0.3550000 0.3083976
```

```
t.test(propdata1month$prop_pas~propdata1month$prop_is_unpub) # ns
```

```
##  
## Welch Two Sample t-test  
##  
## data: propdata1month$prop_pas by propdata1month$prop_is_unpub  
## t = 0.015733, df = 7.1246, p-value = 0.9879  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.2150854 0.2179769  
## sample estimates:  
## mean in group N mean in group Y  
## 1.014139 1.012693
```

```
t.test(propdata3month$prop_pas~propdata3month$prop_is_unpub) # ns
```

```
##  
## Welch Two Sample t-test  
##  
## data: propdata3month$prop_pas by propdata3month$prop_is_unpub  
## t = -0.64928, df = 9.6866, p-value = 0.5312  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:
```

```
## -0.2453510 0.1350015
## sample estimates:
## mean in group N mean in group Y
##      0.8051718      0.8603466
```

```
t.test(propdata6month$prop_pas~propdata6month$prop_is_unpub) # ns
```

```
##
## Welch Two Sample t-test
##
## data: propdata6month$prop_pas by propdata6month$prop_is_unpub
## t = 0.41065, df = 8.8071, p-value = 0.6911
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.1717416 0.2476125
## sample estimates:
## mean in group N mean in group Y
##      0.8388664      0.8009310
```

```
# Removing unpublished effects
```

```
meansdatanounpub <-meansdata[ !(meansdata$mean_is_unpub=="Y"),]
propdatanounpub <-propdata[ !(propdata$prop_is_unpub=="Y"),]
```

```
# Examining frequency by months
```

```
freq(meansdatanounpub$calc_month, plot = FALSE) # Few published means at any time point
```

```
## meansdatanounpub$calc_month
##      Frequency Percent
## 0           2  13.333
## 1           2  13.333
## 2           1   6.667
## 3           5  33.333
## 4           1   6.667
## 6           2  13.333
## 12          2  13.333
## Total       15 100.000
```

```
freq(propdatanounpub$calc_months, plot = FALSE) # 1 and 3 months have at least 10 effects
```

```
## propdatanounpub$calc_months
##      Frequency Percent
## 0           4   8.163
## 1          13  26.531
## 2           8  16.327
## 3          10  20.408
## 4           3   6.122
## 5           1   2.041
## 6           5  10.204
## 9           1   2.041
## 12          4   8.163
## Total       49 100.000
```

```

# Making datasets containing only 1 and 3 month point prevalence effects
propdata1month_nounpub <- propdatanounpub[ (propdatanounpub$calc_months==1),]
propdata3month_nounpub <- propdatanounpub[ (propdatanounpub$calc_months==3),]

# Run models to examine funnel plot asymmetry
modelforfunnel1month_nounpub <- rma(prop_pas, prop_pas_v, data=propdata1month_nounpub)
modelforfunnel3month_nounpub <- rma(prop_pas, prop_pas_v, data=propdata3month_nounpub)

# Egger's regression test for funnel plot asymmetry
regtest.rma(modelforfunnel1month_nounpub, model="rma", predictor="sei")

```

```

##
## Regression Test for Funnel Plot Asymmetry
##
## model:      mixed-effects meta-regression model
## predictor:  standard error
##
## test for funnel plot asymmetry: z = -0.4841, p = 0.6283

```

```

# no significant asymmetry at 1 month
regtest.rma(modelforfunnel3month_nounpub, model="rma", predictor="sei")

```

```

##
## Regression Test for Funnel Plot Asymmetry
##
## model:      mixed-effects meta-regression model
## predictor:  standard error
##
## test for funnel plot asymmetry: z = 1.0576, p = 0.2902

```

```

# no significant asymmetry at 3 months

```

```

# Visually inspect funnel plots (Appendix D & E)
#Files saved in the default directory; add a filepath to save to a different folder
png(file='funnelplot1month.png', width = 8, height = 8, units = 'in', res = 300)
funnel(modelforfunnel1month_nounpub, digits=2,
       ylab = expression("Standard Error"),
       xlab = expression("PTSD Point Prevalence (Arcsine Square Root Transformed)"),
       main = expression(bold("Funnel Plot for 1 Month Point Prevalence")),
       family = "Times",
       level = c(90, 95, 99), shade = c("white", "gray", "darkgray"), cex=.4)
dev.off()

```

```

## pdf
## 2

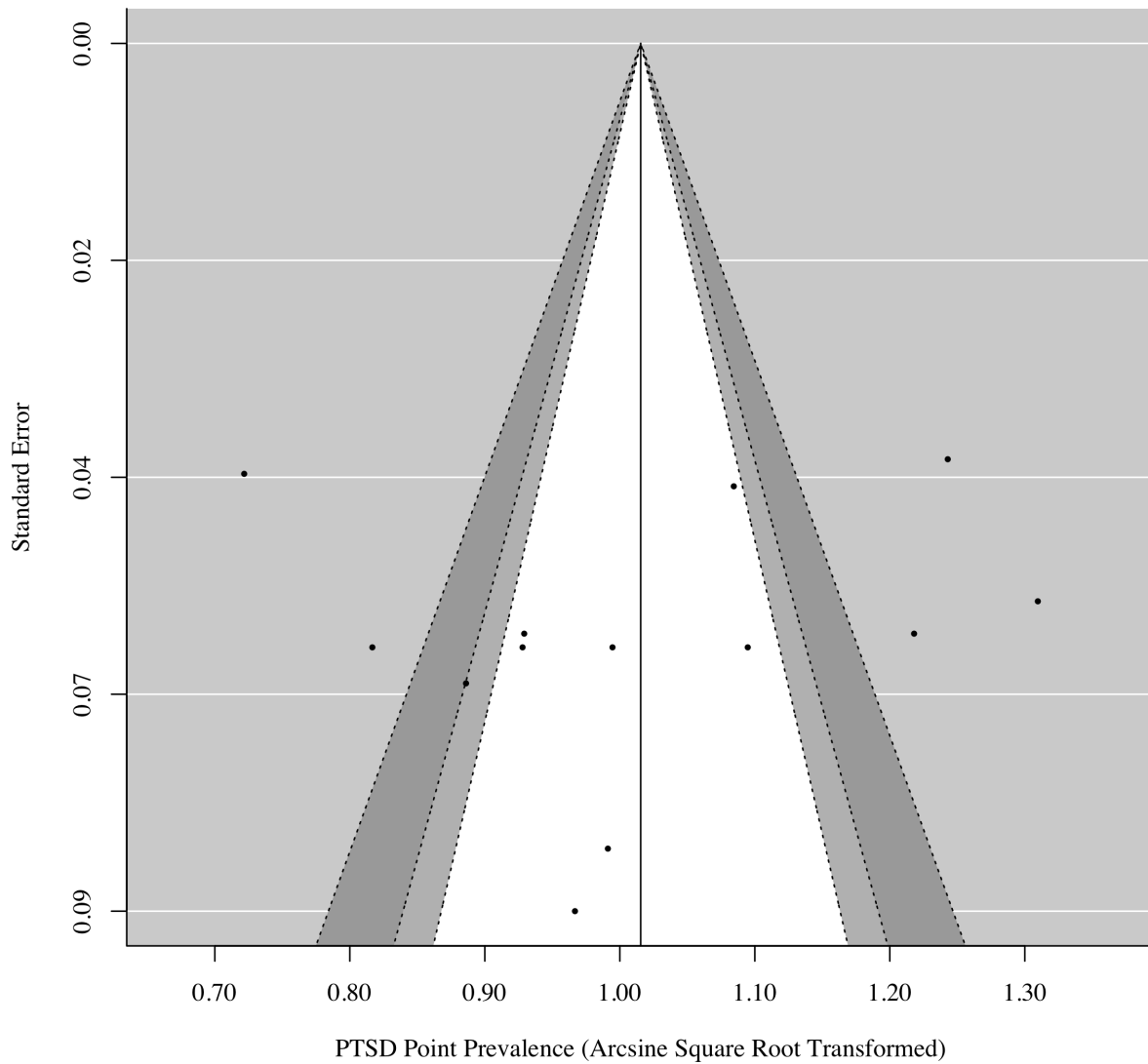
```

```

include_graphics('funnelplot1month.png')

```

Funnel Plot for 1 Month Point Prevalence

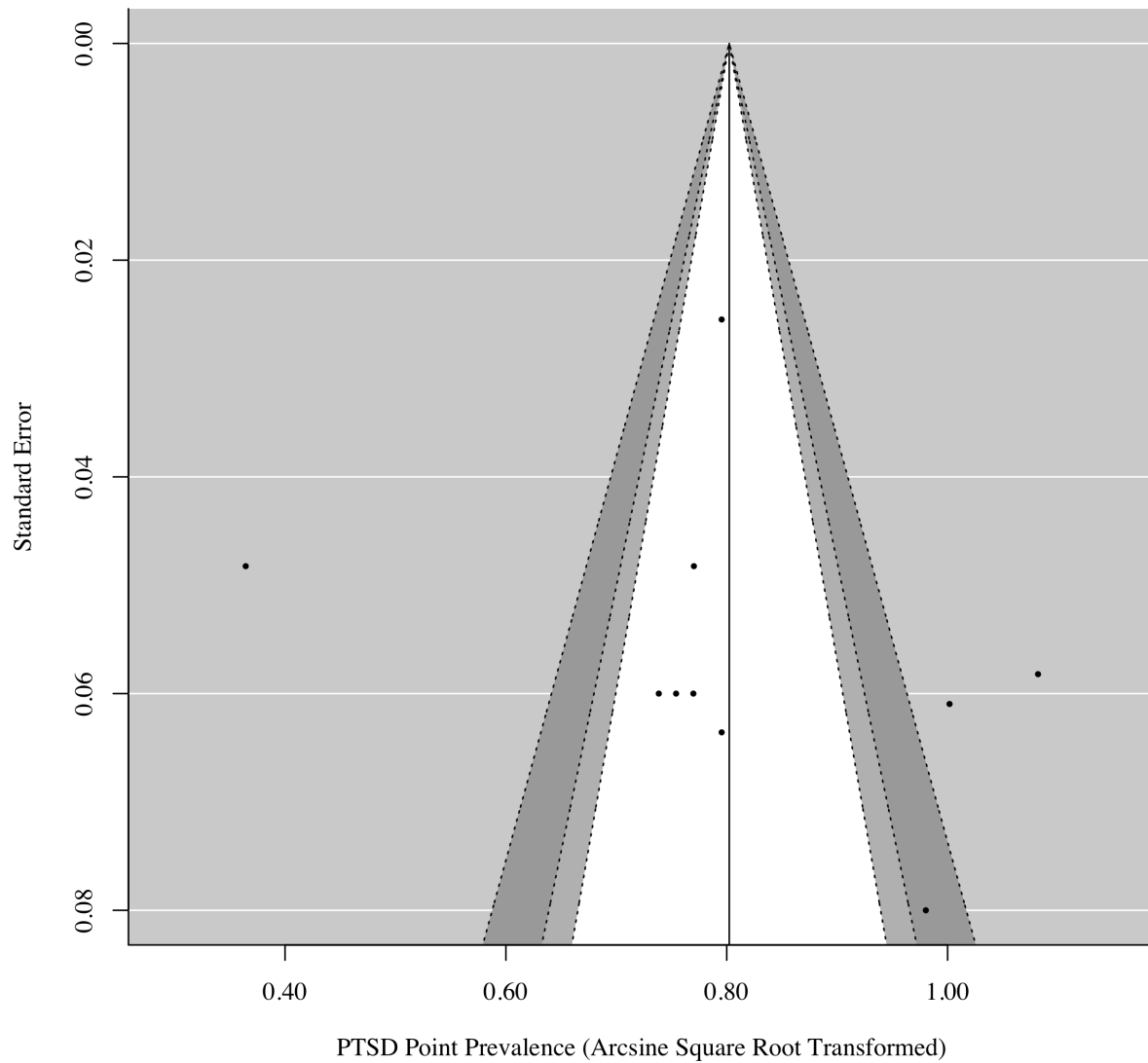


```
png(file='funnelplot3month.png', width = 8, height = 8, units = 'in', res = 300)
funnel(modelforfunnel3month_nounpub, digits=2,
  ylab = expression("Standard Error"),
  xlab = expression("PTSD Point Prevalence (Arcsine Square Root Transformed)"),
  main = expression(bold("Funnel Plot for 3 Month Point Prevalence")),
  family = "Times",
  level = c(90, 95, 99), shade = c("white", "gray", "darkgray"), cex=.4)
dev.off()
```

```
## pdf
## 2
```

```
include_graphics('funnelplot3month.png')
```

Funnel Plot for 3 Month Point Prevalence



Modeling Change Over Time

Testing appropriate random effect structure for overall model - Proportions

```
# Fixed effects model  
ptsdpropfixed <- rma.mv(prop_pas, prop_pas_v,  
  mods = ~months_exact,  
  data = prosptsd_long)
```



```

# 3-level model (effect sizes nested in measures nested in samples)
ptsdpropfull <- rma.mv(prop_pas, prop_pas_v,
                     mods = ~months_exact,
                     random = ~1 | SampleID/Effect, data = prosptsd_long)

# 2-level model (effect sizes nested in measures)
ptsdpropmeasurenest <- rma.mv(prop_pas, prop_pas_v,
                              mods = ~months_exact,
                              random = ~1 | Effect, data = prosptsd_long)

# 2-level model (effect sizes nested in samples)
ptsdpropsamplenest <- rma.mv(prop_pas, prop_pas_v,
                             mods = ~months_exact,
                             random = ~1 | SampleID, data = prosptsd_long)

# random effect for sample nest better than no random effect
anova(ptsdpropfixed,ptsdpropsamplenest)

##
##          df          AIC          BIC          AICc          logLik          LRT          pval          QE
## Full      3 175.0366 181.9079 175.3844 -84.5183
## Reduced  2 708.2370 712.8179 708.4084 -352.1185 535.2004 <.0001 970.2573

# random effect for measure nest better than no random effect
anova(ptsdpropfixed,ptsdpropmeasurenest)

##
##          df          AIC          BIC          AICc          logLik          LRT          pval          QE
## Full      3  47.1967  54.0681  47.5445 -20.5984
## Reduced  2 708.2370 712.8179 708.4084 -352.1185 663.0403 <.0001 970.2573

# -2LL for measure nest is preferable to the sample nest
logLik.rma(ptsdpropsamplenest)

## 'log Lik.' -84.51828 (df=3)

logLik.rma(ptsdpropmeasurenest)

## 'log Lik.' -20.59835 (df=3)

# no difference between measure nest and full model
anova(ptsdpropmeasurenest,ptsdpropfull)

##
##          df          AIC          BIC          AICc          logLik          LRT          pval          QE
## Full      4  48.9875  58.1493  49.5757 -20.4937
## Reduced  3  47.1967  54.0681  47.5445 -20.5984 0.2092 0.6474 970.2573

```

Testing appropriate random effect structure for overall model - Means

```

# Testing for need for random effects in means
ptsdmeanfixed <- rma.mv(m_rescaled01, v_rescaled01,
                      mods = ~months_exact,
                      data = prosptsd_long) #Fixed effects model
# 3-level model (effects nested in samples)
ptsdmeanfull <- rma.mv(m_rescaled01, v_rescaled01,
                     mods = ~months_exact,
                     random = ~1 | SampleID/Effect, data = prosptsd_long)
# 2-level model, nested in effects
ptsdmeaneffectnest <- rma.mv(m_rescaled01, v_rescaled01,
                             mods = ~months_exact,
                             random = ~1 | Effect, data = prosptsd_long)
# 2-level model, nested in samples
ptsdmeansamplenest <- rma.mv(m_rescaled01, v_rescaled01,
                              mods = ~months_exact,
                              random = ~1 | SampleID, data = prosptsd_long)

# random effect for sample nest better than no random effect
anova(ptsdmeanfixed,ptsdmeansamplenest)

```

```

##
##          df          AIC          BIC          AICc          logLik          LRT          pval
## Full      3 1490.7271 1496.7491 1491.1977 -742.3636
## Reduced  2 3147.4264 3151.4411 3147.6572 -1571.7132 1658.6993 <.0001
##
##          QE
## Full      3442.8309
## Reduced  3442.8309

```

```

# random effect for effect nest better than no random effect,
# -2LL is preferable to the sample nest
anova(ptsdmeanfixed,ptsdmeaneffectnest)

```

```

##
##          df          AIC          BIC          AICc          logLik          LRT          pval
## Full      3  144.2802  150.3022  144.7508   -69.1401
## Reduced  2 3147.4264 3151.4411 3147.6572 -1571.7132 3005.1462 <.0001
##
##          QE
## Full      3442.8309
## Reduced  3442.8309

```

```

# no difference between effect nest and full model
anova(ptsdmeaneffectnest,ptsdmeanfull)

```

```

##
##          df          AIC          BIC          AICc          logLik          LRT          pval          QE
## Full      4 146.2802 154.3095 147.0802 -69.1401                    3442.8309
## Reduced  3 144.2802 150.3022 144.7508 -69.1401 0.0000 1.0000 3442.8309

```

Obtaining I2

```

# Create function to compute i2 for multilevel models based on this site:
# http://www.metafor-project.org/doku.php/tips:i2_multilevel_multivariate
i2 <- function(variance, modelname) {
  W <- diag(1/variance)
  X <- model.matrix(modelname)
  P <- W - W %*% X %*% solve(t(X) %*% W %*% X) %*% t(X) %*% W
  value <- 100 * sum(modelname$sigma2) /
    (sum(modelname$sigma2)
     + (modelname$k-modelname$p)/sum(diag(P)))
  return(value)
}

# Reduce the dataset to the cases in the model for matrix multiplication
propdata<-prosptsd_long[!is.na(prosptsd_long$prop_pas_v),]
meansdata<-prosptsd_long[!is.na(prosptsd_long$v_rescaled01),]

# Compute i2 based on effects nest model
ptsdpropeffectnest <- rma.mv(prop_pas, prop_pas_v,
                             random = ~1 | Effect, data = propdata)
ptsdmeaneffectnest <- rma.mv(m_rescaled01, v_rescaled01,
                             random = ~1 | Effect, data = meansdata)

i2(propdata$prop_pas_v, ptsdpropeffectnest)

```

```
## [1] 90.1209
```

```
i2(meansdata$v_rescaled01, ptsdmeaneffectnest)
```

```
## [1] 98.16067
```

Testing linear vs. quadratic effects - Proportions

```

ptsdproplin <- rma.mv(prop_pas, prop_pas_v,
                     mods = ~months_exact, random = ~1 | Effect,
                     data = prosptsd_long, method = "ML")
ptsdpropquad <- rma.mv(prop_pas, prop_pas_v,
                      mods = ~months_exact+I(months_exact^2), random = ~1 | Effect,
                      data = prosptsd_long, method = "ML")

anova(ptsdproplin,ptsdpropquad) # -2LL is significantly lower with quadratic effect

```

```

##
##          df          AIC          BIC          AICc      logLik      LRT      pval      QE
## Full      4 -16.0727 -6.8027 -15.5012  12.0363          915.0019
## Reduced   3  43.5405 50.4930  43.8786 -18.7703  61.6132 <.0001  970.2573

```

Testing linear vs. quadratic effects - Means

```
ptsdmeanlin <- rma.mv(m_rescaled01, v_rescaled01,
  mods = ~months_exact, random = ~1 | Effect,
  data = prosptsd_long, method = "ML")
ptsdmeanquad <- rma.mv(m_rescaled01, v_rescaled01,
  mods = ~months_exact+I(months_exact^2), random = ~1 | Effect,
  data = prosptsd_long, method = "ML")

anova(ptsdmeanlin,ptsdmeanquad) # -2LL is significantly lower with quadratic effect
```

```
##
##          df      AIC      BIC      AICc  logLik      LRT  pval      QE
## Full      4 -39.0552 -30.8830 -38.2859  23.5276      178.8291 <.0001 3331.4419
## Reduced   3 137.7740 143.9031 138.2268 -65.8870      178.8291 <.0001 3442.8309
```

Linear-linear piecewise models

Testing knots in proportions

```
# Base model (proportions)
ptsdproplin_base <- rma.mv(prop_pas, prop_pas_v, mods = ~months_exact,
  random = ~1 |Effect, data = prosptsd_long)
summary(ptsdproplin_base)
```

```
##
## Multivariate Meta-Analysis Model (k = 75; method: REML)
##
##   logLik Deviance      AIC      BIC      AICc
## -20.5984  41.1967  47.1967  54.0681  47.5445
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0329 0.1813    21     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 73) = 970.2573, p-val < .0001
##
## Test of Moderators (coefficient 2):
## QM(df = 1) = 191.4714, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt           1.0359 0.0412  25.1259 <.0001  0.9551  1.1167 ***
## months_exact     -0.0363 0.0026 -13.8373 <.0001 -0.0415 -0.0312 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Model A: Knot at 1 month (proportions)
ptsdproplinA <- rma.mv(prop_pas, prop_pas_v, mods = ~pre1m+post1m,
  random = ~1 |Effect, data = prosptsd_long)
summary(ptsdproplinA)
```

```
##
## Multivariate Meta-Analysis Model (k = 75; method: REML)
##
##   logLik Deviance      AIC      BIC      AICc
## -6.9098  13.8196  21.8196  30.9263  22.4166
##
## Variance Components:
##
##           estim      sqrt  nlvls  fixed  factor
## sigma^2    0.0343  0.1853    21     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 72) = 959.2527, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 222.5533, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      1.1857  0.0499  23.7511 <.0001  1.0878  1.2835 ***
## pre1m        -0.2279  0.0345  -6.6095 <.0001 -0.2955 -0.1603 ***
## post1m       -0.0296  0.0029 -10.2499 <.0001 -0.0353 -0.0240 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Model B: Knot at 2 months (proportions)
ptsdproplinB <- rma.mv(prop_pas, prop_pas_v, mods = ~pre2m+post2m,
  random = ~1 |Effect, data = prosptsd_long)
summary(ptsdproplinB)
```

```
##
## Multivariate Meta-Analysis Model (k = 75; method: REML)
##
##   logLik Deviance      AIC      BIC      AICc
## 15.9842 -31.9683 -23.9683 -14.8617 -23.3713
##
## Variance Components:
##
##           estim      sqrt  nlvls  fixed  factor
## sigma^2    0.0339  0.1841    21     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 72) = 910.3081, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
```

```
## QM(df = 2) = 268.3059, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      1.1773  0.0448  26.2680 <.0001    1.0895    1.2652 ***
## pre2m       -0.1520  0.0135 -11.2956 <.0001   -0.1784   -0.1257 ***
## post2m      -0.0195  0.0033  -5.9712 <.0001   -0.0258   -0.0131 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Model C: Knot at 3 months (proportions)
ptsdproplinC <- rma.mv(prop_pas, prop_pas_v, mods = ~pre3m+post3m,
                      random = ~1 |Effect, data = prosptsd_long)
summary(ptsdproplinC)
```

```
##
## Multivariate Meta-Analysis Model (k = 75; method: REML)
##
##      logLik Deviance      AIC      BIC      AICc
## 23.3240 -46.6480 -38.6480 -29.5414 -38.0510
##
## Variance Components:
##
##      estim      sqrt  nlvls  fixed  factor
## sigma^2  0.0335  0.1831    21     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 72) = 892.0254, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 283.0106, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      1.1528  0.0434  26.5849 <.0001    1.0678    1.2377 ***
## pre3m       -0.1105  0.0082 -13.5024 <.0001   -0.1265   -0.0944 ***
## post3m      -0.0135  0.0035  -3.8054  0.0001   -0.0205   -0.0065 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Model D: Knot at 4 months (proportions)
ptsdproplinD <- rma.mv(prop_pas, prop_pas_v, mods = ~pre4m+post4m,
                      random = ~1 |Effect, data = prosptsd_long)
summary(ptsdproplinD)
```

```
##
## Multivariate Meta-Analysis Model (k = 75; method: REML)
##
##      logLik Deviance      AIC      BIC      AICc
```

```

## 18.5097 -37.0194 -29.0194 -19.9127 -28.4223
##
## Variance Components:
##
##          estim  sqrt  nlvls  fixed  factor
## sigma^2  0.0330  0.1816    21    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 72) = 897.1660, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 273.3142, p-val < .0001
##
## Model Results:
##
##          estimate      se      zval    pval    ci.lb    ci.ub
## intrcpt      1.1333  0.0427  26.5562 <.0001  1.0496  1.2169 ***
## pre4m       -0.0920  0.0067 -13.7530 <.0001 -0.1051 -0.0789 ***
## post4m      -0.0054  0.0043  -1.2521  0.2105 -0.0138  0.0031
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# Model E: Knot at 5 months (proportions)
ptsdproplinE <- rma.mv(prop_pas, prop_pas_v, mods = ~pre5m+post5m,
                    random = ~1 |Effect, data = prosptsd_long)
summary(ptsdproplinE)

```

```

##
## Multivariate Meta-Analysis Model (k = 75; method: REML)
##
##   logLik Deviance      AIC      BIC      AICc
##   7.8315 -15.6631  -7.6631   1.4436  -7.0661
##
## Variance Components:
##
##          estim  sqrt  nlvls  fixed  factor
## sigma^2  0.0333  0.1824    21    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 72) = 926.4562, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 251.9593, p-val < .0001
##
## Model Results:
##
##          estimate      se      zval    pval    ci.lb    ci.ub
## intrcpt      1.1073  0.0425  26.0710 <.0001  1.0241  1.1906 ***
## pre5m       -0.0733  0.0054 -13.4969 <.0001 -0.0840 -0.0627 ***
## post5m      -0.0032  0.0050  -0.6346  0.5257 -0.0130  0.0066
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
# Model F: Knot at 6 months(proportions)
ptsdproplinF <- rma.mv(prop_pas, prop_pas_v, mods = ~pre6m+post6m,
  random = ~1 |Effect, data = prosptsd_long)
summary(ptsdproplinF)
```

```
##
## Multivariate Meta-Analysis Model (k = 75; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
## -0.3289   0.6579   8.6579  17.7645   9.2549
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0329 0.1815    21    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 72) = 936.8233, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 235.5985, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      1.0864  0.0420  25.8867 <.0001   1.0042   1.1687 ***
## pre6m       -0.0596  0.0044 -13.6142 <.0001  -0.0682  -0.0510 ***
## post6m      -0.0040  0.0055  -0.7260  0.4679  -0.0149   0.0068
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Table results
knot_summary_table_prop <- data.frame("knot"=seq(0,6, by=1)) %>%
  mutate(int= ifelse(knot==0, round(ptsdproplin_base$b[1], digits=3),
    ifelse(knot==1, round(ptsdproplinA$b[1], digits=3),
    ifelse(knot==2, round(ptsdproplinB$b[1], digits=3),
    ifelse(knot==3, round(ptsdproplinC$b[1], digits=3),
    ifelse(knot==4, round(ptsdproplinD$b[1], digits=3),
    ifelse(knot==5, round(ptsdproplinE$b[1], digits=3),
    ifelse(knot==6, round(ptsdproplinF$b[1], digits=3), NA)))))),
  preslp = ifelse(knot==1, round(ptsdproplinA$b[2], digits=3),
    ifelse(knot==2, round(ptsdproplinB$b[2], digits=3),
    ifelse(knot==3, round(ptsdproplinC$b[2], digits=3),
    ifelse(knot==4, round(ptsdproplinD$b[2], digits=3),
    ifelse(knot==5, round(ptsdproplinE$b[2], digits=3),
    ifelse(knot==6, round(ptsdproplinF$b[2], digits=3), NA)))))),
  postslp = ifelse(knot==0, round(ptsdproplin_base$b[2], digits=3),
    ifelse(knot==1, round(ptsdproplinA$b[3], digits=3),
    ifelse(knot==2, round(ptsdproplinB$b[3], digits=3),
    ifelse(knot==3, round(ptsdproplinC$b[3], digits=3),
    ifelse(knot==4, round(ptsdproplinD$b[3], digits=3),
    ifelse(knot==5, round(ptsdproplinE$b[3], digits=3),
```



```

        ifelse(knot==6, round(ptsdproplinF$b[3], digits=3), NA))))),
AICval= ifelse(knot==0, round(fitstats(ptsdproplin_base)[3], digits=2),
        ifelse(knot==1, round(fitstats(ptsdproplinA)[3], digits=2),
        ifelse(knot==2, round(fitstats(ptsdproplinB)[3], digits=2),
        ifelse(knot==3, round(fitstats(ptsdproplinC)[3], digits=2),
        ifelse(knot==4, round(fitstats(ptsdproplinD)[3], digits=2),
        ifelse(knot==5, round(fitstats(ptsdproplinE)[3], digits=2),
        ifelse(knot==6, round(fitstats(ptsdproplinF)[3], digits=2), NA))))),
BICval= ifelse(knot==0, round(fitstats(ptsdproplin_base)[4], digits=2),
        ifelse(knot==1, round(fitstats(ptsdproplinA)[4], digits=2),
        ifelse(knot==2, round(fitstats(ptsdproplinB)[4], digits=2),
        ifelse(knot==3, round(fitstats(ptsdproplinC)[4], digits=2),
        ifelse(knot==4, round(fitstats(ptsdproplinD)[4], digits=2),
        ifelse(knot==5, round(fitstats(ptsdproplinE)[4], digits=2),
        ifelse(knot==6, round(fitstats(ptsdproplinF)[4], digits=2), NA))))),
# Output table for manuscript (Appendix F)
panderOptions('keep.trailing.zeros',TRUE)
pander(knot_summary_table_prop)

```

knot	int	preslp	postslp	AICval	BICval
0	1.036	NA	-0.036	47.20	54.07
1	1.186	-0.228	-0.030	21.82	30.93
2	1.177	-0.152	-0.019	-23.97	-14.86
3	1.153	-0.110	-0.014	-38.65	-29.54
4	1.133	-0.092	-0.005	-29.02	-19.91
5	1.107	-0.073	-0.003	-7.66	1.44
6	1.086	-0.060	-0.004	8.66	17.76

Testing knots in means

```

# Base model (means)
ptsdmeanlin_base <- rma.mv(m_rescaled01, v_rescaled01, mods = ~months_exact,
                        random = ~1 |Effect, data = prosptsd_long, method = "ML")
summary(ptsdmeanlin_base)

```

```

##
## Multivariate Meta-Analysis Model (k = 57; method: ML)
##
##   logLik  Deviance      AIC      BIC      AICc
## -65.8870  443.0518  137.7740  143.9031  138.2268
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0206  0.1435    18    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 55) = 3442.8309, p-val < .0001
##

```

```

## Test of Moderators (coefficient 2):
## QM(df = 1) = 256.2766, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt          0.4588  0.0342  13.4232 <.0001   0.3918   0.5258 ***
## months_exact    -0.0143  0.0009 -16.0086 <.0001  -0.0161  -0.0126 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Model A: Knot at 1 month (means)
ptsdmeanlinA <- rma.mv(m_rescaled01, v_rescaled01, mods = ~pre1m+post1m,
                      random = ~1 |Effect, data = prosptsd_long, method = "ML")
summary(ptsdmeanlinA)

##
## Multivariate Meta-Analysis Model (k = 57; method: ML)
##
##   logLik Deviance      AIC      BIC      AICc
## -16.1206 343.5190  40.2412  48.4134  41.0104
##
## Variance Components:
##
##           estim      sqrt  nlvls  fixed  factor
## sigma^2    0.0210  0.1450    18     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 54) = 3405.9913, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 355.7448, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt          0.5701  0.0363  15.7118 <.0001   0.4990   0.6412 ***
## pre1m            -0.1559  0.0142 -10.9637 <.0001  -0.1838  -0.1280 ***
## post1m           -0.0107  0.0010 -11.0145 <.0001  -0.0125  -0.0088 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Model B: Knot at 2 months (means)
ptsdmeanlinB <- rma.mv(m_rescaled01, v_rescaled01, mods = ~pre2m+post2m,
                      random = ~1 |Effect, data = prosptsd_long, method = "ML")
summary(ptsdmeanlinB)

##
## Multivariate Meta-Analysis Model (k = 57; method: ML)
##
##   logLik Deviance      AIC      BIC      AICc

```

```

## 24.2227 262.8324 -40.4455 -32.2733 -39.6762
##
## Variance Components:
##
##      estim  sqrt  nlvls  fixed  factor
## sigma^2  0.0200 0.1415   18    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 54) = 3264.0074, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 436.5968, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      0.5571 0.0345  16.1461 <.0001  0.4895  0.6248 ***
## pre2m       -0.0914 0.0058 -15.7299 <.0001 -0.1028 -0.0800 ***
## post2m      -0.0074 0.0010  -7.2115 <.0001 -0.0095 -0.0054 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# Model C: Knot at 3 months (means)
ptsdmeanlinC <- rma.mv(m_rescaled01, v_rescaled01, mods = ~pre3m+post3m,
                    random = ~1 |Effect, data = prosptsd_long, method = "ML")
summary(ptsdmeanlinC)

```

```

##
## Multivariate Meta-Analysis Model (k = 57; method: ML)
##
##      logLik  Deviance      AIC      BIC      AICc
## 40.7132 229.8515 -73.4263 -65.2541 -72.6571
##
## Variance Components:
##
##      estim  sqrt  nlvls  fixed  factor
## sigma^2  0.0192 0.1387   18    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 54) = 3156.4331, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 469.7526, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      0.5426 0.0336  16.1691 <.0001  0.4769  0.6084 ***
## pre3m       -0.0633 0.0035 -18.2365 <.0001 -0.0701 -0.0565 ***
## post3m      -0.0060 0.0011  -5.6186 <.0001 -0.0080 -0.0039 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
# Model D: Knot at 4 months (means)
ptsdmeanlinD <- rma.mv(m_rescaled01, v_rescaled01, mods = ~pre4m+post4m,
                      random = ~1 |Effect, data = prosptsd_long, method = "ML")
summary(ptsdmeanlinD)
```

```
##
## Multivariate Meta-Analysis Model (k = 57; method: ML)
##
##   logLik Deviance      AIC      BIC      AICc
## 35.3841 240.5097 -62.7681 -54.5959 -61.9989
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0196 0.1401    18    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 54) = 3271.0823, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 459.0028, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      0.5326 0.0338  15.7617 <.0001  0.4664  0.5988 ***
## pre4m        -0.0532 0.0029 -18.5088 <.0001 -0.0588 -0.0476 ***
## post4m       -0.0007 0.0013  -0.5268 0.5983 -0.0033  0.0019
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Model E: Knot at 5 months (means)
ptsdmeanlinE <- rma.mv(m_rescaled01, v_rescaled01, mods = ~pre5m+post5m,
                      random = ~1 |Effect, data = prosptsd_long, method = "ML")
summary(ptsdmeanlinE)
```

```
##
## Multivariate Meta-Analysis Model (k = 57; method: ML)
##
##   logLik Deviance      AIC      BIC      AICc
## 19.5989 272.0801 -31.1977 -23.0255 -30.4285
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0199 0.1412    18    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 54) = 3356.5201, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
```

```
## QM(df = 2) = 427.3650, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt    0.5175  0.0340  15.2429 <.0001    0.4510    0.5841 ***
## pre5m     -0.0422  0.0023 -18.2574 <.0001   -0.0467   -0.0376 ***
## post5m     0.0022  0.0015   1.3954  0.1629   -0.0009    0.0052
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Model F: Knot at 6 months (means)
ptsdmeanlinF <- rma.mv(m_rescaled01, v_rescaled01, mods = ~pre6m+post6m,
                    random = ~1 |Effect, data = prosptsd_long, method = "ML")
summary(ptsdmeanlinF)
```

```
##
## Multivariate Meta-Analysis Model (k = 57; method: ML)
##
##      logLik Deviance      AIC      BIC      AICc
##      5.7638 299.7503  -3.5275   4.6447  -2.7583
##
## Variance Components:
##
##      estim      sqrt  nlvls  fixed  factor
## sigma^2  0.0197  0.1405    18    no    Effect
##
## Test for Residual Heterogeneity:
## QE(df = 54) = 3380.2946, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 399.7371, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt    0.5049  0.0337  14.9793 <.0001    0.4388    0.5710 ***
## pre6m     -0.0337  0.0018 -18.2220 <.0001   -0.0373   -0.0301 ***
## post6m     0.0036  0.0017   2.0439  0.0410    0.0001    0.0070 *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Table results
knot_summary_table_mean <- data.frame("knot"=seq(0,6, by=1)) %>%
  mutate(int= ifelse(knot==0, round(ptsdmeanlin_base$b[1], digits=3),
                    ifelse(knot==1, round(ptsdmeanlinA$b[1], digits=3),
                    ifelse(knot==2, round(ptsdmeanlinB$b[1], digits=3),
                    ifelse(knot==3, round(ptsdmeanlinC$b[1], digits=3),
                    ifelse(knot==4, round(ptsdmeanlinD$b[1], digits=3),
                    ifelse(knot==5, round(ptsdmeanlinE$b[1], digits=3),
                    ifelse(knot==6, round(ptsdmeanlinF$b[1], digits=3), NA)))))))))
```

```

preslp = ifelse(knot==1, round(ptsdmeanlinA$b[2], digits=3),
               ifelse(knot==2, round(ptsdmeanlinB$b[2], digits=3),
               ifelse(knot==3, round(ptsdmeanlinC$b[2], digits=3),
               ifelse(knot==4, round(ptsdmeanlinD$b[2], digits=3),
               ifelse(knot==5, round(ptsdmeanlinE$b[2], digits=3),
               ifelse(knot==6, round(ptsdmeanlinF$b[2], digits=3, NA)))))),
postsplp= ifelse(knot==0, round(ptsdmeanlin_base$b[2], digits=3),
                 ifelse(knot==1, round(ptsdmeanlinA$b[3], digits=3),
                 ifelse(knot==2, round(ptsdmeanlinB$b[3], digits=3),
                 ifelse(knot==3, round(ptsdmeanlinC$b[3], digits=3),
                 ifelse(knot==4, round(ptsdmeanlinD$b[3], digits=3),
                 ifelse(knot==5, round(ptsdmeanlinE$b[3], digits=3),
                 ifelse(knot==6, round(ptsdmeanlinF$b[3], digits=3, NA)))))),
AICval= ifelse(knot==0, round(fitstats(ptsdmeanlin_base)[3], digits=2),
               ifelse(knot==1, round(fitstats(ptsdmeanlinA)[3], digits=2),
               ifelse(knot==2, round(fitstats(ptsdmeanlinB)[3], digits=2),
               ifelse(knot==3, round(fitstats(ptsdmeanlinC)[3], digits=2),
               ifelse(knot==4, round(fitstats(ptsdmeanlinD)[3], digits=2),
               ifelse(knot==5, round(fitstats(ptsdmeanlinE)[3], digits=2),
               ifelse(knot==6, round(fitstats(ptsdmeanlinF)[3], digits=2, NA)))))),
BICval= ifelse(knot==0, round(fitstats(ptsdmeanlin_base)[4], digits=2),
               ifelse(knot==1, round(fitstats(ptsdmeanlinA)[4], digits=2),
               ifelse(knot==2, round(fitstats(ptsdmeanlinB)[4], digits=2),
               ifelse(knot==3, round(fitstats(ptsdmeanlinC)[4], digits=2),
               ifelse(knot==4, round(fitstats(ptsdmeanlinD)[4], digits=2),
               ifelse(knot==5, round(fitstats(ptsdmeanlinE)[4], digits=2),
               ifelse(knot==6, round(fitstats(ptsdmeanlinF)[4], digits=2, NA)))))))
# Output table for manuscript (Appendix F)
panderOptions('keep.trailing.zeros', TRUE)
pander(knot_summary_table_mean)

```

knot	int	preslp	postsplp	AICval	BICval
0	0.459	NA	-0.014	137.77	143.90
1	0.570	-0.156	-0.011	40.24	48.41
2	0.557	-0.091	-0.007	-40.45	-32.27
3	0.543	-0.063	-0.006	-73.43	-65.25
4	0.533	-0.053	-0.001	-62.77	-54.60
5	0.518	-0.042	0.002	-31.20	-23.03
6	0.505	-0.034	0.004	-3.53	4.64

Generating predicted values- Proportions

```

# Set up dataframe for values to predict
months <- data.frame("months"=seq(0,12, by=0.25))
months$pre3m <- ifelse(months$months<=3, months$months, 3)
months$post3m<- ifelse(months$months>3, (months$months-3), 0)

# Run model
ptsdprop <- rma.mv(prop_pas, prop_pas_v,

```

```

      mods = ~pre3m+post3m, random = ~1 | Effect, data = prosptsd_long)
summary(ptsdprop)

```

```

##
## Multivariate Meta-Analysis Model (k = 75; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
## 23.3240 -46.6480 -38.6480 -29.5414 -38.0510
##
## Variance Components:
##
##      estim  sqrt  nlvls  fixed  factor
## sigma^2  0.0335 0.1831   21    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 72) = 892.0254, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 283.0106, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      1.1528 0.0434  26.5849 <.0001  1.0678  1.2377 ***
## pre3m       -0.1105 0.0082 -13.5024 <.0001 -0.1265 -0.0944 ***
## post3m      -0.0135 0.0035  -3.8054 0.0001 -0.0205 -0.0065 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# Generate predicted values and putting into dataframe
pprop<-as.data.frame(predict.rma
                      (ptsdprop,
                       newmods = cbind(months$pre3m,months$post3m),
                       intercept=TRUE))
# Add months to dataset
pprop$month<-seq(0,12, by=0.25)
# Back-transforming proportions
# from formula here:
# https://bit.ly/36J4S6a
# and multiply by 100
pprop$pred<-((sin(pprop$pred))^2)*100
pprop$ci.lb<-((sin(pprop$ci.lb))^2)*100
pprop$ci.ub<-((sin(pprop$ci.ub))^2)*100
# Printing predicted values
pprop<-pprop[c("month","pred","ci.lb","ci.ub")]
# Output week-by-week version of table for manuscript (Table 2)
panderOptions('digits',4)
pander(pprop)

```

month	pred	ci.lb	ci.ub
0.00	83.52	76.76	89.31

month	pred	ci.lb	ci.ub
0.25	81.42	74.49	87.47
0.50	79.23	72.14	85.51
0.75	76.94	69.71	83.45
1.00	74.58	67.21	81.29
1.25	72.13	64.64	79.05
1.50	69.63	62.00	76.74
1.75	67.06	59.31	74.36
2.00	64.44	56.58	71.92
2.25	61.77	53.80	69.44
2.50	59.07	50.99	66.91
2.75	56.34	48.16	64.36
3.00	53.60	45.31	61.78
3.25	53.26	45.01	61.42
3.50	52.92	44.70	61.07
3.75	52.59	44.38	60.72
4.00	52.25	44.06	60.37
4.25	51.91	43.74	60.03
4.50	51.57	43.42	59.69
4.75	51.24	43.09	59.35
5.00	50.90	42.76	59.01
5.25	50.56	42.43	58.68
5.50	50.22	42.09	58.35
5.75	49.89	41.75	58.03
6.00	49.55	41.40	57.70
6.25	49.21	41.06	57.39
6.50	48.87	40.71	57.07
6.75	48.54	40.35	56.76
7.00	48.20	40.00	56.45
7.25	47.86	39.64	56.14
7.50	47.52	39.27	55.84
7.75	47.19	38.91	55.54
8.00	46.85	38.54	55.25
8.25	46.51	38.17	54.95
8.50	46.18	37.80	54.66
8.75	45.84	37.42	54.38
9.00	45.50	37.04	54.09
9.25	45.17	36.66	53.81
9.50	44.83	36.28	53.54
9.75	44.50	35.90	53.26
10.00	44.16	35.51	52.99
10.25	43.82	35.12	52.72
10.50	43.49	34.73	52.46
10.75	43.16	34.34	52.19
11.00	42.82	33.95	51.93
11.25	42.49	33.55	51.67
11.50	42.15	33.16	51.42
11.75	41.82	32.76	51.17
12.00	41.49	32.36	50.92

Generating predicted values- Means

```

# Set up dataframe for values to predict
months <- data.frame("months"=seq(0,12, by=0.25))
months$pre3m <- ifelse(months$months<=3,
                        months$months, 3)
months$post3m <- ifelse(months$months>3,
                        (months$months-3), 0)

# Run model
ptsdmean <- rma.mv(m_rescaled01, v_rescaled01,
                  mods = ~pre3m+post3m, random = -1 | Effect,
                  data = prosptsd_long)

# Generate predicted values and putting into dataframe
pmean<-as.data.frame(predict.rma
                      (ptsdmean,
                       newmods = cbind(months$pre3m,months$post3m),
                       intercept=TRUE))

# Multiply by 100 for presentation
pmean$pred<-pmean$pred*100
pmean$ci.lb<-pmean$ci.lb*100
pmean$ci.ub<-pmean$ci.ub*100
# Add months to dataset
pmean$month<-seq(0,12, by=0.25)
# Print predicted values
pmean<-pmean[c("month","pred","ci.lb","ci.ub")]
# Output week-by-week version of table for manuscript (Table 2)
panderOptions('digits',4)
pander(pmean)

```

month	pred	ci.lb	ci.ub
0.00	54.27	47.51	61.03
0.25	52.68	45.95	59.42
0.50	51.10	44.39	57.81
0.75	49.52	42.83	56.21
1.00	47.94	41.27	54.61
1.25	46.36	39.70	53.01
1.50	44.77	38.12	51.42
1.75	43.19	36.54	49.84
2.00	41.61	34.96	48.26
2.25	40.03	33.37	46.68
2.50	38.44	31.78	45.11
2.75	36.86	30.19	43.54
3.00	35.28	28.58	41.97
3.25	35.13	28.44	41.82
3.50	34.98	28.29	41.67
3.75	34.83	28.15	41.52
4.00	34.68	28.00	41.37
4.25	34.53	27.85	41.22
4.50	34.39	27.71	41.06
4.75	34.24	27.56	40.91

month	pred	ci.lb	ci.ub
5.00	34.09	27.41	40.76
5.25	33.94	27.26	40.61
5.50	33.79	27.11	40.47
5.75	33.64	26.96	40.32
6.00	33.49	26.81	40.17
6.25	33.34	26.66	40.02
6.50	33.19	26.51	39.87
6.75	33.04	26.36	39.73
7.00	32.89	26.21	39.58
7.25	32.75	26.06	39.43
7.50	32.60	25.91	39.29
7.75	32.45	25.75	39.14
8.00	32.30	25.60	39.00
8.25	32.15	25.45	38.85
8.50	32.00	25.29	38.71
8.75	31.85	25.14	38.56
9.00	31.70	24.98	38.42
9.25	31.55	24.83	38.28
9.50	31.40	24.67	38.13
9.75	31.25	24.52	37.99
10.00	31.11	24.36	37.85
10.25	30.96	24.20	37.71
10.50	30.81	24.05	37.57
10.75	30.66	23.89	37.43
11.00	30.51	23.73	37.29
11.25	30.36	23.57	37.15
11.50	30.21	23.42	37.01
11.75	30.06	23.26	36.87
12.00	29.91	23.10	36.73

Plots

```

prosptsd_long$range_months <- car::recode(prosptsd_long$months_exact,
  "0:0.99=0;
  0.99001:1.99=1;
  1.99001:2.99=2;
  2.99001:3.99=3;
  3.99001:4.999=4;
  4.99001:5.99=5;
  5.99001:6.99=6;
  6.99001:7.99=7;
  7.99001:8.99=8;
  8.99001:9.99=9;
  9.99001:10.99=10;
  10.99001:11.99=11;
  11.99001:12.99=12")
prosptsd_mon0 <- prosptsd_long[ (prosptsd_long$range_months==0),]
prosptsd_mon1 <- prosptsd_long[ (prosptsd_long$range_months==1),]
prosptsd_mon2 <- prosptsd_long[ (prosptsd_long$range_months==2),]

```

```

prosptsd_mon3 <- prosptsd_long[ (prosptsd_long$range_months==3),]
prosptsd_mon4 <- prosptsd_long[ (prosptsd_long$range_months==4),]
prosptsd_mon5 <- prosptsd_long[ (prosptsd_long$range_months==5),]
prosptsd_mon6 <- prosptsd_long[ (prosptsd_long$range_months==6),]
prosptsd_mon8 <- prosptsd_long[ (prosptsd_long$range_months==8),]
prosptsd_mon9 <- prosptsd_long[ (prosptsd_long$range_months==9),]
prosptsd_mon12 <- prosptsd_long[ (prosptsd_long$range_months==12),]
prosptsd_mon0<-prosptsd_mon0[!is.na(prosptsd_mon0$SampleID), ]
prosptsd_mon1<-prosptsd_mon1[!is.na(prosptsd_mon1$SampleID), ]
prosptsd_mon2<-prosptsd_mon2[!is.na(prosptsd_mon2$SampleID), ]
prosptsd_mon3<-prosptsd_mon3[!is.na(prosptsd_mon3$SampleID), ]
prosptsd_mon4<-prosptsd_mon4[!is.na(prosptsd_mon4$SampleID), ]
prosptsd_mon5<-prosptsd_mon5[!is.na(prosptsd_mon5$SampleID), ]
prosptsd_mon6<-prosptsd_mon6[!is.na(prosptsd_mon6$SampleID), ]
prosptsd_mon8<-prosptsd_mon8[!is.na(prosptsd_mon8$SampleID), ]
prosptsd_mon9<-prosptsd_mon9[!is.na(prosptsd_mon9$SampleID), ]
prosptsd_mon12<-prosptsd_mon12[!is.na(prosptsd_mon12$SampleID), ]

# Set up dataframe for values to predict
plotting_m3 <- data.frame("months"=seq(0,12, by=0.5))
plotting_m3$pre3m <- ifelse(plotting_m3$months<=3,
                           plotting_m3$months, 3)
plotting_m3$post3m <- ifelse(plotting_m3$months>3,
                             (plotting_m3$months-3), 0)

#Prepare linetypes for both plots
linetypes = c(apply(expand.grid(c(2,4), c(1,2,4,8,"A")), 1, paste, collapse=""),
              apply(expand.grid(c(2,4,8), c(2,4), c(5,"F"), 2), 1, paste, collapse=""),
              "4284B4F4", "228F61A4")

### Plot Proportions
ptsdprop0 <- rma.mv(prop_pas, prop_pas_v,
                  random = ~1 | Effect, data = prosptsd_mon0)
ptsdprop1 <- rma.mv(prop_pas, prop_pas_v,
                  random = ~1 | Effect, data = prosptsd_mon1)
ptsdprop2 <- rma.mv(prop_pas, prop_pas_v,
                  random = ~1 | Effect, data = prosptsd_mon2)
ptsdprop3 <- rma.mv(prop_pas, prop_pas_v,
                  random = ~1 | Effect, data = prosptsd_mon3)
ptsdprop6 <- rma.mv(prop_pas, prop_pas_v,
                  random = ~1 | Effect, data = prosptsd_mon6)
ptsdprop12 <- rma.mv(prop_pas, prop_pas_v,
                   random = ~1 | Effect, data = prosptsd_mon12)

p0s<-as.data.frame(predict.rma(ptsdprop0, intercept=TRUE))
p1s<-as.data.frame(predict.rma(ptsdprop1, intercept=TRUE))
p2s<-as.data.frame(predict.rma(ptsdprop2, intercept=TRUE))
p3s<-as.data.frame(predict.rma(ptsdprop3, intercept=TRUE))
p6s<-as.data.frame(predict.rma(ptsdprop6, intercept=TRUE))
p12s<-as.data.frame(predict.rma(ptsdprop12, intercept=TRUE))

# Back-transform (based on back transformation here):
# https://bit.ly/36J4S6a

```

```

p0s<-(sin(p0s))^2
p1s<-(sin(p1s))^2
p2s<-(sin(p2s))^2
p3s<-(sin(p3s))^2
p6s<-(sin(p6s))^2
p12s<-(sin(p12s))^2

# Merge predicted values
proppredmerged_sub <- do.call("rbind", list(p0s,p1s,p2s,p3s,p6s,p12s))
# Create a column to add time points to
proppredmerged_sub$range_months <- NA
# Use the midpoint number of months
proppredmerged_sub$range_months <-c(0,1,2,3,6,12)
proppredmerged_sub$daysgroupedintomonths <- proppredmerged_sub$range_months*30.4167

# Plotting

ptsdprop<- rma.mv(prop_pas, prop_pas_v, mods = ~pre3m+post3m,
                 random = ~1 | Effect, data = prosptsd_long)
pcont<-as.data.frame(predict
                       (ptsdprop,
                        newmods = cbind(plotting_m3$pre3m,plotting_m3$post3m),
                        intercept=TRUE))
pcont<-(sin(pcont))^2
# Add months to dataset
pcont$months_new<-seq(0,12, by=0.5)

# Black and white plot (Figure 1)
proptime_bw<-ggplot() +
  labs(x= "Months Post-Assault", y="Percent Meeting Study PTSD Criteria"
       , linetype="Sample", color="Sample",size="Sample Size") +
  geom_line(data=pcont, aes(x=months_new, y=pred*100),size=5)+
  geom_point(data=prosptsd_long[!is.na(prosptsd_long$prop_pas_v),],
            aes(x=months_exact, y=prop_raw*100, color=citation, size=N))+
  scale_color_manual(values=rep(c("gray80", "gray50", "gray0"),ceiling(
    length(prosptsd_long$citation)/2))[1:length(prosptsd_long$citation)])+
  scale_size_continuous(range=c(2,25),
                        breaks = c(0,50,100,150,200,250,300,350,400,450,500,550))+
  geom_line(data=prosptsd_long[!is.na(prosptsd_long$prop_pas_v),],
            aes(x=months_exact, y=prop_raw*100,
                color=citation, linetype=citation),size=1.5)+
  scale_linetype_manual(values=linetypes) +
  theme_classic() +
  theme(text = element_text(family = "Times", size=48)) +
  theme(axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) +
  theme(axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) +
  theme(legend.key.height=unit(2.8,"line"))+
  theme(legend.key.width = unit(5,"cm"))+
  theme(plot.margin=unit(c(2,2,2,2),"cm"))+
  theme(legend.box.margin=unit(c(2,2,2,2),"cm"))+
  guides(color=guide_legend(ncol=1,override.aes = list(size = 1.5, shape = NA)))+
  ylim(0,100.05)+

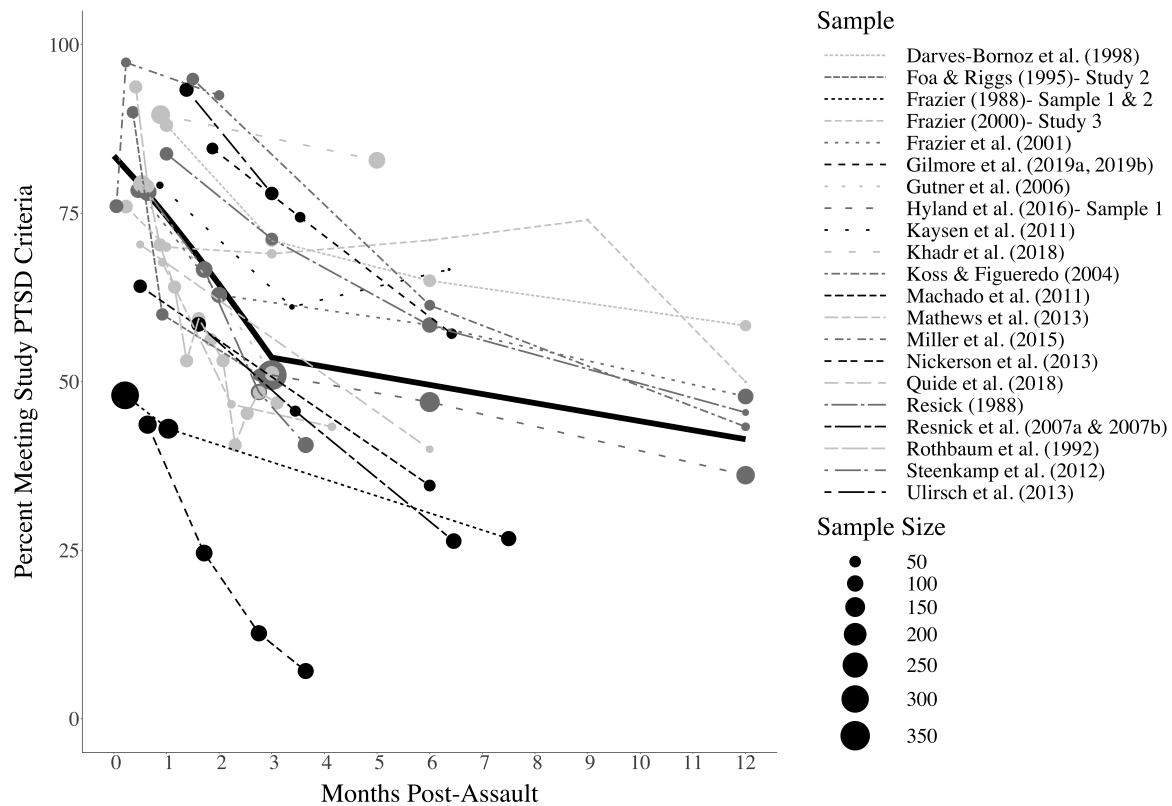
```

```

coord_cartesian(clip = "off")+
scale_x_discrete(limits=c(0,1,2,3,4,5,6,7,8,9,10,11,12))
ggsave(proptime_bw, file='figure_prevalence_bw.png', width = 32, height = 22, dpi=300)

include_graphics('figure_prevalence_bw.png')

```



```

# Color plot (Appendix I)
proptime_color<-ggplot() +
  labs(x= "Months Post-Assault", y="Percent Meeting Study PTSD Criteria",
       color="Sample",size="Sample Size")+
  geom_line(data=pcont, aes(x=months_new, y=pred*100),size=5)+
  geom_ribbon(data=pcont, aes(x=months_new, ymin=ci.lb*100,ymax=ci.ub*100), alpha=0.3)+
  geom_point(data=prosptsd_long[!is.na(prosptsd_long$prop_pas_v)],
            aes(x=months_exact, y=prop_raw*100, color=citation, size=N))+
  scale_color_manual(values=c("#F2C80F", "#8250C4", "#FD625E", "#71E8AC",
                              "#86C2DB", "#CDA470", "#74EA5B", "#D9E5CE",
                              "#D79ADD", "#D7C8D6", "#80ECE4", "#CEEADB",
                              "#7F7E8B", "#A43B76", "#71AB94", "#DEC658",
                              "#E19899", "#8A8666", "#26B0D2", "#E0663C", "#FE6DB6"))+
  scale_size_continuous(range=c(2,25),
                        breaks = c(0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550))+
  geom_line(data=prosptsd_long[!is.na(prosptsd_long$prop_pas_v)],
            aes(x=months_exact, y=prop_raw*100, color=citation),size=1.5)+
  theme_classic() +
  theme(text = element_text(family = "Times", size=48)) +

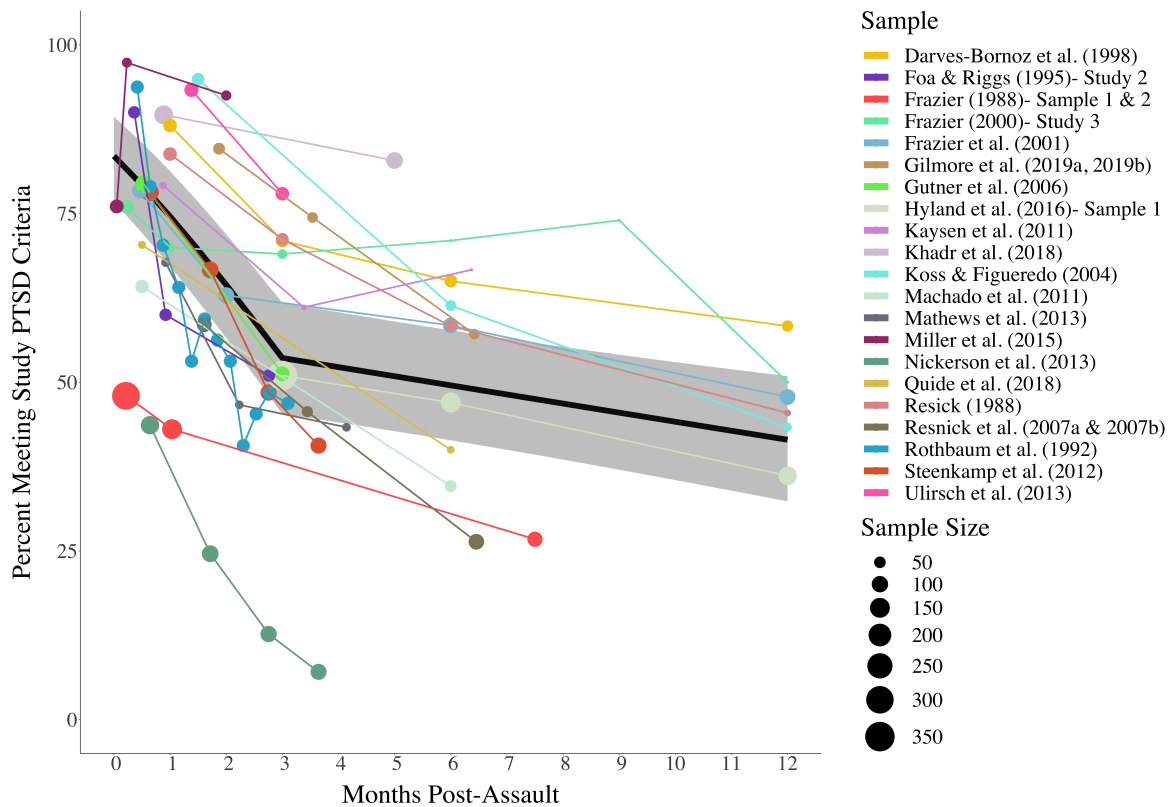
```

```

theme(axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) +
theme(axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) +
theme(legend.key.height=unit(2.8,"line"))+
theme(legend.key.width = unit(2,"cm"))+
theme(plot.margin=unit(c(2,2,2,2),"cm"))+
theme(legend.box.margin=unit(c(2,2,2,2),"cm"))+
guides(color=guide_legend(ncol=1,override.aes = list(size = 6)))+
ylim(0,100.05)+
coord_cartesian(clip = "off")+
scale_x_discrete(limits=c(0,1,2,3,4,5,6,7,8,9,10,11,12))
ggsave(proptime_color, file='figure_prevalence_color.png',
width = 32, height = 22, dpi=300)

include_graphics('figure_prevalence_color.png')

```



```

### Plot means
ptsdmean0 <- rma.mv(m_rescaled01, v_rescaled01,
random = ~1 | Effect, data = prosptsd_mon0)
ptsdmean1 <- rma.mv(m_rescaled01, v_rescaled01,
random = ~1 | Effect, data = prosptsd_mon1)
ptsdmean3 <- rma.mv(m_rescaled01, v_rescaled01,
random = ~1 | Effect, data = prosptsd_mon3)
ptsdmean6 <- rma.mv(m_rescaled01, v_rescaled01,
random = ~1 | Effect, data = prosptsd_mon6)
ptsdmean12 <- rma.mv(m_rescaled01, v_rescaled01,

```

```

        random = ~1 | Effect, data = prosptsd_mon12)

m0s<-as.data.frame(predict.rma(ptsdmean0, intercept=TRUE))
m1s<-as.data.frame(predict.rma(ptsdmean1, intercept=TRUE))
m3s<-as.data.frame(predict.rma(ptsdmean3, intercept=TRUE))
m6s<-as.data.frame(predict.rma(ptsdmean6, intercept=TRUE))
m12s<-as.data.frame(predict.rma(ptsdmean12, intercept=TRUE))

meanspredmerged_sub <- do.call("rbind", list(m0s,m1s,m3s,m6s,m12s)) # Merge predicted values
meanspredmerged_sub$range_months <- NA # Create a column to add time points to
meanspredmerged_sub$range_months <-c(0,1,3,6,12) # Used the midpoint # of months
meanspredmerged_sub$daysgroupedintomonths <- meanspredmerged_sub$range_months*30.4167

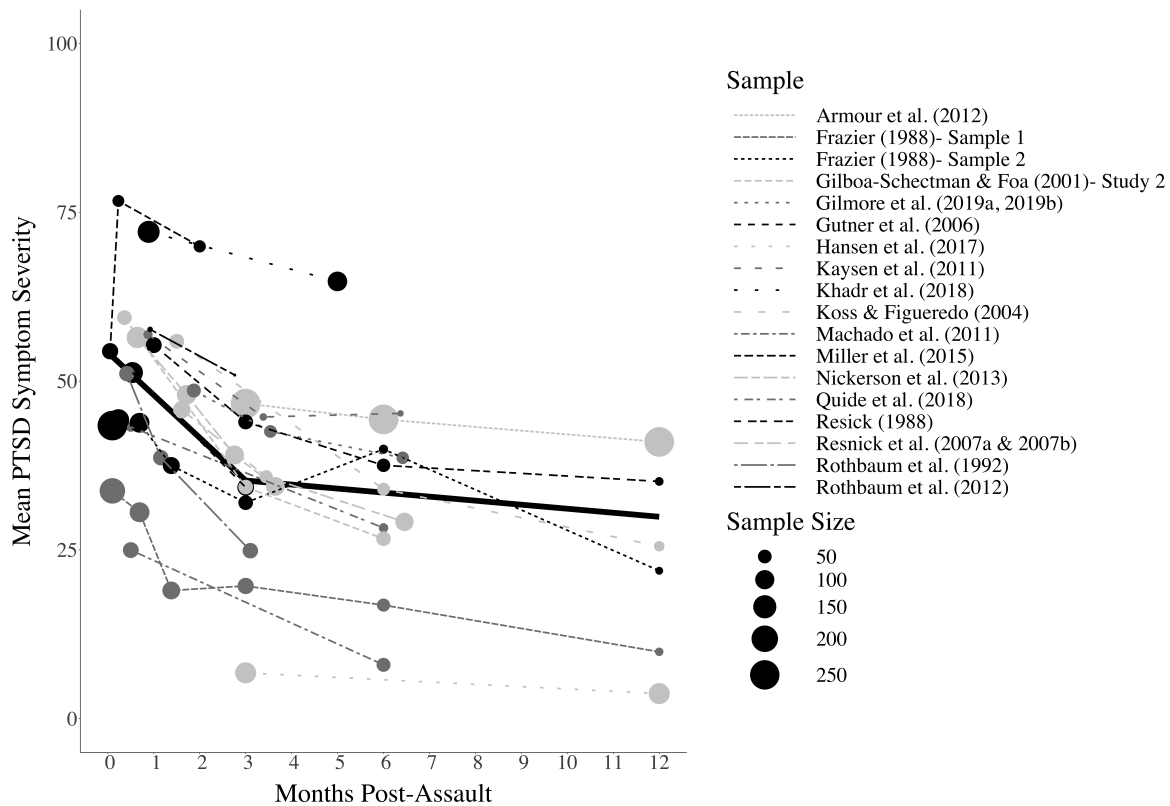
# Run model
ptsdmean <- rma.mv(m_rescaled01, v_rescaled01,
                  mods = ~pre3m+post3m, random = ~1 | Effect, data = prosptsd_long)
# Generate predicted values and put into dataframe
pmean<-as.data.frame(predict.rma
                     (ptsdmean,
                      newmods = cbind(plotting_m3$pre3m,plotting_m3$post3m),
                      intercept=TRUE))

# Add months to dataset
pmean$months_new<-seq(0,12, by=0.5)

# Black and white plot (Figure 2)
meanplot_bw<-ggplot() +
  labs(x= "Months Post-Assault", y="Mean PTSD Symptom Severity", linetype="Sample",
       color="Sample",size="Sample Size") +
  geom_line(data=pmean, aes(x=months_new, y=pred*100),size=5)+
  geom_point(data=prosptsd_long[!is.na(prosptsd_long$v_rescaled01),],
            aes(x=months_exact, y=m_rescaled01*100, color=citation,size=N))+
  scale_color_manual(values=rep(c("gray80", "gray50", "gray0"),ceiling(
    length(prosptsd_long$citation)/2))[1:length(prosptsd_long$citation)]+
  scale_size_continuous(range=c(2,25),
                        breaks = c(0,50,100,150,200,250,300,350,400,450,500,550))+
  geom_line(data=prosptsd_long[!is.na(prosptsd_long$v_rescaled01),],
            aes(x=months_exact, y=m_rescaled01*100, color=citation,
                linetype=citation),size=1.5)+
  scale_linetype_manual(values=linetypes) +
  theme_classic() +
  theme(text = element_text(family = "Times", size=48)) +
  theme(axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) +
  theme(axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) +
  theme(legend.key.height=unit(2.8,"line"))+
  theme(legend.key.width = unit(5,"cm"))+
  theme(plot.margin=unit(c(2,2,2,2),"cm"))+
  theme(legend.box.margin=unit(c(2,2,2,2),"cm"))+
  guides(color=guide_legend(ncol=1,override.aes = list(size = 1.5, shape = NA)))+
  ylim(0,100)+
  expand_limits(y = 900000) +
  coord_cartesian(clip = "off")+
  scale_x_discrete(limits=c(0,1,2,3,4,5,6,7,8,9,10,11,12))
ggsave(meanplot_bw, file='figure_severity_bw.png', width = 32, height = 22, dpi=300)

```

```
include_graphics('figure_severity_bw.png')
```



```
# Color plot (Appendix J)
meanplot_color<-ggplot() +
  labs(x= "Months Post-Assault", y="Mean PTSD Symptom Severity",
       color="Sample",size="Sample Size") +
  geom_line(data=pmean, aes(x=months_new, y=pred*100),size=5)+
  geom_ribbon(data=pmean, aes(x=months_new, ymin=ci.lb*100,ymax=ci.ub*100), alpha=0.3)+
  geom_point(data=prosptsd_long[!is.na(prosptsd_long$v_rescaled01)],,
            aes(x=months_exact, y=m_rescaled01*100, color=citation, size=N))+
  scale_color_manual(values=c("#F2C80F", "#8250C4", "#FD625E", "#71E8AC",
                              "#86C2DB", "#CDA470", "#74EA5B", "#D9E5CE",
                              "#D79ADD", "#FE6DB6", "#80ECE4", "#CEEADB",
                              "#7F7E8B", "#A43B76", "#71AB94", "#DEC658",
                              "#E19899", "#8A8666"))+
  scale_size_continuous(range=c(2,25),
                       breaks = c(0,50,100,150,200,250,300,350,400,450,500,550))+
  geom_line(data=prosptsd_long[!is.na(prosptsd_long$v_rescaled01)],,
            aes(x=months_exact, y=m_rescaled01*100, color=citation),size=1.5)+
  theme_classic() +
  theme(text = element_text(family = "Times", size=48)) +
  theme(axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) +
  theme(axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) +
  theme(legend.key.height=unit(2.8,"line"))+
```

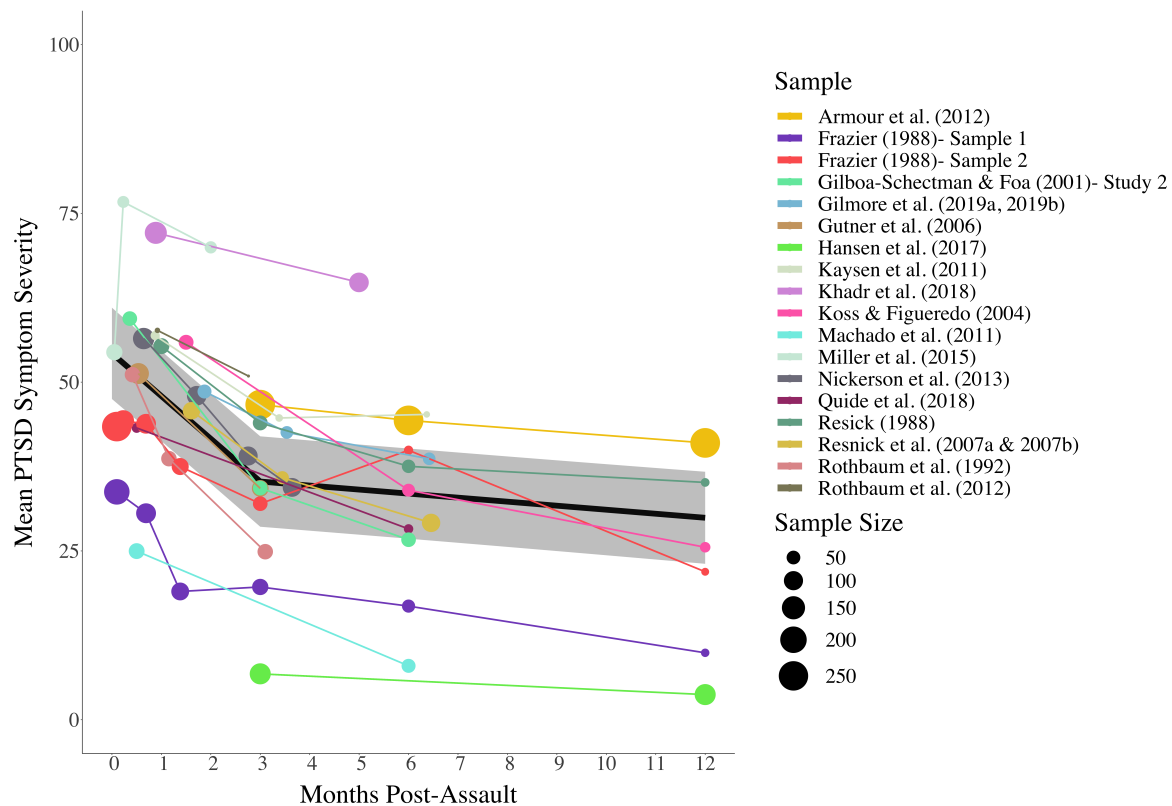


```

theme(legend.key.width = unit(2,"cm"))+
theme(plot.margin=unit(c(2,2,2,2),"cm"))+
theme(legend.box.margin=unit(c(2,2,2,2),"cm"))+
guides(color=guide_legend(ncol=1,override.aes = list(size = 6)))+
ylim(0,100)+
expand_limits(y = 900000) +
coord_cartesian(clip = "off")+
scale_x_discrete(limits=c(0,1,2,3,4,5,6,7,8,9,10,11,12))
ggsave(meanplot_color, file='figure_severity_color.png', width = 32, height = 22, dpi=300)

include_graphics('figure_severity_color.png')

```



Effects of Attrition on Point Prevalence

```

### Descriptives

# Creating dataset to examine attrition during follow-ups
prosptsd_long_attrition <- prosptsd_long %>%
  filter(baseline_timepoint != "Y" #drop baseline time points
         & completer_analysis != "Y" #drop samples with completer analyses
         & SampleID != 23) #exclude ID 23 as redundant with 18 and 22 for descriptives

```

```

# Aggregate maximum attrition by sample
attr <- prosptsd_long_attrition %>%
  group_by(SampleID) %>%
  summarise(max_attr = max(attrition),
            max_months = max(months_exact))

# Maximum attrition descriptives
nrow(attr) # Number of studies included in analysis

## [1] 19

summary(attr$max_attr) # Minimum, maximum, mean of study-mean attrition

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03226 0.23099 0.42105 0.45495 0.64037 0.92490

sd(attr$max_attr)

## [1] 0.2654284

# Association between attrition and follow-up time,
# considering all post-baseline timepoints across samples
cor.test(prosptsd_long_attrition$attrition, prosptsd_long_attrition$months_exact)

##
## Pearson's product-moment correlation
##
## data: prosptsd_long_attrition$attrition and prosptsd_long_attrition$months_exact
## t = 3.5836, df = 54, p-value = 0.0007279
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1982680 0.6287659
## sample estimates:
##      cor
## 0.4383211

# Calculating average attrition before and after 6 months post-assault
attr_6month <- prosptsd_long_attrition %>%
  mutate(post6mo = ifelse(months_exact<6, 0, 1)) %>%
  group_by(post6mo) %>%
  summarise(mean_attr = mean(attrition))
as.data.frame(attr_6month) # Output average at 0 (pre 6 months) and 1 (post 6 months)

##   post6mo mean_attr
## 1         0 0.3373015
## 2         1 0.5457789

```

```
### Correlations between PTSD at first assessment and attrition
```

```
# Set up data for bivariate analyses
```

```
prosptsd_long_attrition_obs1 <- prosptsd_long %>%  
  # Select first observation when PTSD data was obtained  
  filter(obs_num ==1) %>%  
  select(SampleID,Effect,prop_PTSD,m_rescaled01)
```

```
prosptsd_long_attrition_mo3 <- prosptsd_long %>%  
  # Select 3 month attrition  
  filter(calc_months ==3) %>%  
  group_by(SampleID) %>%  
  # Take average of 3-month attritions if multiple in a study  
  summarise(attr3 = mean(attrition, na.rm=TRUE))
```

```
prosptsd_long_attrition_mo6 <- prosptsd_long %>%  
  # Select 6 month attrition  
  filter(calc_months ==6) %>%  
  group_by(SampleID) %>%  
  # Take average of 6-month attritions if multiple in a study  
  summarise(attr6 = mean(attrition, na.rm=TRUE))
```

```
# Merge datasets
```

```
prosptsd_long_attrition<-full_join(  
  prosptsd_long_attrition_obs1,prosptsd_long_attrition_mo3)
```

```
## Joining, by = "SampleID"
```

```
prosptsd_long_attrition<-full_join(  
  prosptsd_long_attrition,prosptsd_long_attrition_mo6)
```

```
## Joining, by = "SampleID"
```

```
# Testing associations between baseline prevalence and 3- and 6-month attrition
```

```
cor.test(prosptsd_long_attrition$prop_PTSD, prosptsd_long_attrition$attr3)
```

```
##
```

```
## Pearson's product-moment correlation
```

```
##
```

```
## data: prosptsd_long_attrition$prop_PTSD and prosptsd_long_attrition$attr3
```

```
## t = -1.0012, df = 8, p-value = 0.346
```

```
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
```

```
## -0.7960670 0.3746419
```

```
## sample estimates:
```

```
## cor
```

```
## -0.3336987
```

```
cor.test(prosptsd_long_attrition$prop_PTSD, prosptsd_long_attrition$attr6)
```

```
##  
## Pearson's product-moment correlation  
##  
## data: prosptsd_long_attrition$prop_PTSD and prosptsd_long_attrition$attr6  
## t = -0.51485, df = 8, p-value = 0.6206  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.7267635 0.5077995  
## sample estimates:  
## cor  
## -0.1790843
```

```
# Testing associations between baseline symptom severity and 3- and 6-month attrition
```

```
cor.test(prosptsd_long_attrition$m_rescaled01, prosptsd_long_attrition$attr3)
```

```
##  
## Pearson's product-moment correlation  
##  
## data: prosptsd_long_attrition$m_rescaled01 and prosptsd_long_attrition$attr3  
## t = -0.74349, df = 8, p-value = 0.4784  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.7618982 0.4469395  
## sample estimates:  
## cor  
## -0.2542277
```

```
cor.test(prosptsd_long_attrition$m_rescaled01, prosptsd_long_attrition$attr6)
```

```
##  
## Pearson's product-moment correlation  
##  
## data: prosptsd_long_attrition$m_rescaled01 and prosptsd_long_attrition$attr6  
## t = -0.92078, df = 9, p-value = 0.3812  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.7595933 0.3719213  
## sample estimates:  
## cor  
## -0.2934169
```

```
### Setting up data to re-calculate model-implied effects under various assumptions
```

```
prosptsd_long <- prosptsd_long %>%  
  filter(!is.na(n_PTSD_calc)) %>%  
  arrange(SampleID, Effect, obs_num) %>%  
  group_by(Effect) %>%  
  mutate(N_maxmin = ifelse((SampleID==10 | (SampleID==25 & obs_num==3)),
```

```

      N,
      N[obs_num==1])) %>%
# Creating a new denominator representing the N at the first observation,
# except when new participants were enrolled post BL, in which case just using N
mutate(ndrop_sincefirstobs =
      ifelse((SampleID==10|(SampleID==25&obs_num==3)|completer_analysis=="Y"),
      NA, N_maxmin-N)) %>%
# Calculating number of people who dropped out since first observation.
# For studies that enrolled new participants post BL, this is NA.
mutate(n_PTSD_calc_new=ifelse(completer_analysis=="Y",
      n_PTSD_calc,n_PTSD_calc+
      (ndrop_sincefirstobs*prop_raw[obs_num==1])) %>%
# Creating new numerator representing the hypothetical number of people with PTSD
# if the proportion of people with PTSD was the same as the proportion at
# baseline- same as Hiller et al approach
# For completer analyses, retaining the original number of people with PTSD
mutate(n_PTSD_calc_MAX=ifelse(completer_analysis=="Y",
      n_PTSD_calc,n_PTSD_calc+ndrop_sincefirstobs)) %>%
# Creating new numerator representing the hypothetical number of people with PTSD
# if everyone who dropped out had PTSD.
# For completer analyses, retaining the original number of people with PTSD
# Preventing the number of people with PTSD at follow-up time points
# from being larger than the number of people with PTSD at baseline
filter(!(SampleID==10|(SampleID==25&obs_num==3))) %>%
# Removing studies that enrolled new people
mutate(n_PTSD_calc_MAX= if_else(obs_num==1,n_PTSD_calc_MAX,
# Maintain the value for the first observation
ifelse(n_PTSD_calc_MAX>n_PTSD_calc_MAX[obs_num==1],
      # For all other observations, check if the baseline value is smaller
      n_PTSD_calc_MAX[obs_num==1], # If so, replace with the baseline value
      n_PTSD_calc_MAX))) # If not, keep value

# Computing variance of raw proportion - STANDARD ASSUMPTION
prosptsd_long <- escalc(measure="PR", xi = n_PTSD_calc,
      ni = N, data=prosptsd_long, add=0,
      var.names=c("prop_PTSD_raw_2", "prop_raw_v"))

# Computing arcsine square root transformed proportion - STANDARD ASSUMPTION
prosptsd_long <- escalc(measure="PAS", xi = n_PTSD_calc,
      ni = N, data=prosptsd_long, add=0,
      var.names=c("prop_pas", "prop_pas_v"))
prosptsd_long$ci.ub.prop <- prosptsd_long$prop_raw +
      1.96*sqrt(prosptsd_long$prop_raw_v)
prosptsd_long$ci.lb.prop <- prosptsd_long$prop_raw -
      1.96*sqrt(prosptsd_long$prop_raw_v)

# Computing variance of raw proportion - ASSUMPTION 1 (Attrition less likely w/ PTSD)
prosptsd_long <- escalc(measure="PR", xi = n_PTSD_calc,
      ni = N_maxmin, data=prosptsd_long, add=0,
      var.names=c("prop_PTSD_raw_2_MIN", "prop_raw_v_MIN"))
# Computing arcsine square root transformed proportion - ASSUMPTION 1 (Attrition less likely w/ PTSD)

```

```

prosptsd_long <- escalec(measure="PAS", xi = n_PTSD_calc, ni = N_maxmin,
                        data=prosptsd_long, add=0,
                        var.names=c("prop_pas_MIN", "prop_pas_v_MIN"))
prosptsd_long$ci.ub.prop_MIN <- prosptsd_long$prop_PTSD_raw_2_MIN
+ 1.96*sqrt(prosptsd_long$prop_raw_v_MIN)

```

```

## [1] 0.05645445 0.10304180 0.05142948 0.08074872 0.11481106 0.10613977
## [7] 0.10077294 0.11361084 0.11325901 0.08524672 0.08669284 0.07344304
## [13] 0.07424441 0.08842655 0.08814251 0.08433599 0.08659858 0.07049061
## [19] 0.05233328 0.03949790 0.05200260 0.04191064 0.03525639 0.07764525
## [25] 0.11369310 0.11810144 0.11810144 0.16248041 0.19934586 0.19724374
## [31] 0.07214125 0.08433638 0.16455958 0.17518705 0.17370866 0.17223929
## [37] 0.18215239 0.05930506 0.09979052 0.11193584 0.11755523 0.12226051
## [43] 0.12032741 0.12153920 0.12226051 0.12032741 0.12196048 0.12244017
## [49] 0.12226051 0.07788250 0.12718159 0.12977821 0.05588679 0.04569965
## [55] 0.02944907 0.10663392 0.09447347 0.09926357 0.11620073 0.11620073
## [61] 0.08752386 0.11550586 0.11697733 0.08417957 0.09806689 0.13223336
## [67] 0.13549888 0.05605737 0.12712616 0.10576175

```

```

prosptsd_long$ci.lb.prop_MIN <- prosptsd_long$prop_PTSD_raw_2_MIN
- 1.96*sqrt(prosptsd_long$prop_raw_v_MIN)

```

```

## [1] -0.05645445 -0.10304180 -0.05142948 -0.08074872 -0.11481106
## [6] -0.10613977 -0.10077294 -0.11361084 -0.11325901 -0.08524672
## [11] -0.08669284 -0.07344304 -0.07424441 -0.08842655 -0.08814251
## [16] -0.08433599 -0.08659858 -0.07049061 -0.05233328 -0.03949790
## [21] -0.05200260 -0.04191064 -0.03525639 -0.07764525 -0.11369310
## [26] -0.11810144 -0.11810144 -0.16248041 -0.19934586 -0.19724374
## [31] -0.07214125 -0.08433638 -0.16455958 -0.17518705 -0.17370866
## [36] -0.17223929 -0.18215239 -0.05930506 -0.09979052 -0.11193584
## [41] -0.11755523 -0.12226051 -0.12032741 -0.12153920 -0.12226051
## [46] -0.12032741 -0.12196048 -0.12244017 -0.12226051 -0.07788250
## [51] -0.12718159 -0.12977821 -0.05588679 -0.04569965 -0.02944907
## [56] -0.10663392 -0.09447347 -0.09926357 -0.11620073 -0.11620073
## [61] -0.08752386 -0.11550586 -0.11697733 -0.08417957 -0.09806689
## [66] -0.13223336 -0.13549888 -0.05605737 -0.12712616 -0.10576175

```

Computing variance of raw proportion - ASSUMPTION 2 (Attrition more likely with PTSD)

```

prosptsd_long <- escalec(measure="PR", xi = n_PTSD_calc_MAX,
                        ni = N_maxmin, data=prosptsd_long, add=0,
                        var.names=c("prop_PTSD_raw_2_MAX", "prop_raw_v_MAX"))

```

Computing arcsine square root transformed proportion - ASSUMPTION 2 (Attrition more likely w/ PTSD)

```

prosptsd_long <- escalec(measure="PAS", xi = n_PTSD_calc_MAX,
                        ni = N_maxmin, data=prosptsd_long, add=0,
                        var.names=c("prop_pas_MAX", "prop_pas_v_MAX"))
prosptsd_long$ci.ub.prop_MAX <- prosptsd_long$prop_PTSD_raw_2_MAX
+ 1.96*sqrt(prosptsd_long$prop_raw_v_MAX)

```

```

## [1] 0.05645445 0.09052852 0.05142948 0.05734355 0.11481106 0.11971272
## [7] 0.10077294 0.10077294 0.10077294 0.10077294 0.10077294 0.10077294
## [13] 0.07424441 0.08186708 0.08891503 0.08975712 0.08659858 0.08438061
## [19] 0.08104467 0.07940759 0.05200260 0.05200260 0.05200260 0.07764525

```

```
## [25] 0.10613977 0.11107912 0.10957407 0.16248041 0.18184974 0.16248041
## [31] 0.07214125 0.08327550 0.16455958 0.17592159 0.17518705 0.17223929
## [37] 0.18743327 0.05930506 0.09979052 0.11193584 0.11755523 0.12226051
## [43] 0.12032741 0.12153920 0.12226051 0.12032741 0.12196048 0.12244017
## [49] 0.12226051 0.07788250 0.12718159 0.12977821 0.05588679 0.05588679
## [55] 0.05588679 0.10663392 0.10663392 0.09926357 0.09926357 0.09926357
## [61] 0.08752386 0.10292059 0.10829997 0.09061025 0.09806689 0.11100418
## [67] 0.12930841 0.05605737 0.11556522 0.11556522
```

```
prosptsd_long$ci.lb.prop_MAX <- prosptsd_long$prop_PTSD_raw_2_MAX
- 1.96*sqrt(prosptsd_long$prop_raw_v_MAX)
```

```
## [1] -0.05645445 -0.09052852 -0.05142948 -0.05734355 -0.11481106
## [6] -0.11971272 -0.10077294 -0.10077294 -0.10077294 -0.10077294
## [11] -0.10077294 -0.10077294 -0.07424441 -0.08186708 -0.08891503
## [16] -0.08975712 -0.08659858 -0.08438061 -0.08104467 -0.07940759
## [21] -0.05200260 -0.05200260 -0.05200260 -0.07764525 -0.10613977
## [26] -0.11107912 -0.10957407 -0.16248041 -0.18184974 -0.16248041
## [31] -0.07214125 -0.08327550 -0.16455958 -0.17592159 -0.17518705
## [36] -0.17223929 -0.18743327 -0.05930506 -0.09979052 -0.11193584
## [41] -0.11755523 -0.12226051 -0.12032741 -0.12153920 -0.12226051
## [46] -0.12032741 -0.12196048 -0.12244017 -0.12226051 -0.07788250
## [51] -0.12718159 -0.12977821 -0.05588679 -0.05588679 -0.05588679
## [56] -0.10663392 -0.10663392 -0.09926357 -0.09926357 -0.09926357
## [61] -0.08752386 -0.10292059 -0.10829997 -0.09061025 -0.09806689
## [66] -0.11100418 -0.12930841 -0.05605737 -0.11556522 -0.11556522
```

```
prosptsd_long <- prosptsd_long %>%
  mutate(
    #Model A: Knot at 1 month
    pre1m = ifelse(months_exact<=1, months_exact, 1),
    post1m= ifelse(months_exact>1, (months_exact-1), 0),
    #Model B: Knot at 2 months
    pre2m = ifelse(months_exact<=2, months_exact, 2),
    post2m= ifelse(months_exact>2, (months_exact-2), 0),
    #Model C: Knot at 3 months
    pre3m = ifelse(months_exact<=3, months_exact, 3),
    post3m= ifelse(months_exact>3, (months_exact-3), 0),
    #Model D: Knot at 4 months
    pre4m = ifelse(months_exact<=4, months_exact, 4),
    post4m= ifelse(months_exact>4, (months_exact-4), 0),
    #Model E: Knot at 5 months
    pre5m = ifelse(months_exact<=5, months_exact, 5),
    post5m= ifelse(months_exact>5, (months_exact-5), 0),
    #Model F: Knot at 6 months
    pre6m = ifelse(months_exact<=6, months_exact, 6),
    post6m= ifelse(months_exact>6, (months_exact-6), 0))

#Setting up dataframe for values to predict
months <- data.frame("months"=seq(0,12, by=1))
months$pre3m <- ifelse(months$months<=3, months$months, 3)
months$post3m<- ifelse(months$months>3, (months$months-3), 0)
```

```

### Recalculating effects under assumption 1: Attrition LESS likely with PTSD

# Running model
ptsdprop_as1 <- rma.mv(prop_pas_MIN, prop_pas_v_MIN,
                      mods = ~pre3m+post3m, random = ~1 | Effect, data = prosptsd_long)
summary(ptsdprop_as1)

##
## Multivariate Meta-Analysis Model (k = 70; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
## -3.7153   7.4306  15.4306  24.2493  16.0757
##
## Variance Components:
##
##           estim      sqrt  nlvls  fixed  factor
## sigma^2    0.0380  0.1948    20     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 67) = 1233.0284, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 851.1605, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      1.1488  0.0465  24.6963 <.0001   1.0576   1.2400 ***
## pre3m       -0.1451  0.0074 -19.5387 <.0001  -0.1596  -0.1305 ***
## post3m      -0.0355  0.0029 -12.2766 <.0001  -0.0412  -0.0298 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Generating predicted values and putting into dataframe
pprop_MIN<-as.data.frame(predict.rma
                          (ptsdprop_as1,
                           newmods = cbind(months$pre3m,months$post3m),
                           intercept=TRUE))

# Adding months to dataset
pprop_MIN$month<-seq(0,12, by=1)
#Back-transforming proportions
#from formula here: https://bit.ly/36J4S6a
#and multiplying by 100
pprop_MIN$pred<-((sin(pprop_MIN$pred))^2)*100
pprop_MIN$ci.lb<-((sin(pprop_MIN$ci.lb))^2)*100
pprop_MIN$ci.ub<-((sin(pprop_MIN$ci.ub))^2)*100
# Printing predicted values
pprop_MIN<-pprop_MIN[c("month","pred","ci.lb","ci.ub")]
# Output table for comparison to manuscript (Appendix G)
write.table(format(pprop_MIN, digits=4), row.names = F,sep = '\t')

## "month"  "pred"  "ci.lb"  "ci.ub"

```



```
## " 0" "83.23" "75.899" "89.45"
## " 1" "71.15" "62.886" "78.75"
## " 2" "57.30" "48.607" "65.77"
## " 3" "42.84" "34.214" "51.70"
## " 4" "39.35" "30.949" "48.09"
## " 5" "35.92" "27.744" "44.52"
## " 6" "32.55" "24.617" "41.02"
## " 7" "29.27" "21.590" "37.60"
## " 8" "26.09" "18.681" "34.27"
## " 9" "23.04" "15.911" "31.05"
## "10" "20.12" "13.301" "27.95"
## "11" "17.35" "10.873" "24.98"
## "12" "14.75" " 8.645" "22.15"
```

```
### Recalculating effects under assumption 2: Attrition MORE likely with PTSD
```

```
# Running model
```

```
ptsdprop_as2 <- rma.mv(prop_pas_MAX, prop_pas_v_MAX,
                      mods = ~pre3m+post3m, random = ~1 | Effect, data = prosptsd_long)
summary(ptsdprop_as2)
```

```
##
## Multivariate Meta-Analysis Model (k = 70; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
## 27.6494 -55.2988 -47.2988 -38.4800 -46.6536
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0244 0.1561    20    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 67) = 776.2138, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 79.5101, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      1.0935 0.0385 28.3930 <.0001  1.0180  1.1689 ***
## pre3m       -0.0645 0.0074 -8.7009 <.0001 -0.0791 -0.0500 ***
## post3m       0.0047 0.0029  1.6225 0.1047 -0.0010  0.0103
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Generating predicted values and putting into dataframe
```

```
pprop_MAX<-as.data.frame(predict.rma
                          (ptsdprop_as2,
                           newmods = cbind(months$pre3m,months$post3m),
                           intercept=TRUE))
```

```

#Adding months to dataset
pprop_MAX$month<-seq(0,12, by=1)
#Back-transforming proportions
#from formula here: https://bit.ly/36J4S6a
#and multiplying by 100
pprop_MAX$pred<-((sin(pprop_MAX$pred))^2)*100
pprop_MAX$ci.lb<-((sin(pprop_MAX$ci.lb))^2)*100
pprop_MAX$ci.ub<-((sin(pprop_MAX$ci.ub))^2)*100
#Printing predicted values
pprop_MAX<-pprop_MAX[c("month","pred","ci.lb","ci.ub")]
# Output table for comparison to manuscript (Appendix G)
write.table(format(pprop_MAX, digits=4), row.names = F, sep = '\t')

```

```

## "month" "pred" "ci.lb" "ci.ub"
## " 0" "78.89" "72.43" "84.70"
## " 1" "73.40" "66.85" "79.47"
## " 2" "67.52" "60.75" "73.94"
## " 3" "61.35" "54.18" "68.28"
## " 4" "61.80" "54.73" "68.64"
## " 5" "62.26" "55.23" "69.04"
## " 6" "62.71" "55.68" "69.48"
## " 7" "63.16" "56.09" "69.96"
## " 8" "63.62" "56.46" "70.48"
## " 9" "64.07" "56.79" "71.03"
## "10" "64.51" "57.07" "71.62"
## "11" "64.96" "57.32" "72.23"
## "12" "65.41" "57.54" "72.87"

```

Moderator Analyses (Table 3)

```

### Unconditional model

# Create datasets for models
prosptsd_long <- prosptsd_long %>%
  group_by(Effect) %>%
  mutate(us = ifelse(country=="US", 1, 0))
propdata<-prosptsd_long[!is.na(prosptsd_long$prop_pas_v),]
meansdata<-prosptsd_long[!is.na(prosptsd_long$v_rescaled01),]

# Frequencies
nrow(propdata) #number of effects

```

```
## [1] 70
```

```

prop_agg <- propdata %>%
  group_by(SampleID) %>%
  mutate(countpersample = seq(n())) %>%
#exclude ID23 as redundant with 18 and 22 for descriptives
  filter(countpersample == 1 & SampleID != 23)
nrow(prop_agg)

```

```
## [1] 18
```

```
nrow(meansdata) #number of effects
```

```
## [1] 30
```

```
mean_agg <- meansdata %>%  
  group_by(SampleID) %>%  
  mutate(countpersample = seq(n())) %>%  
  #exclude ID23 as redundant with 18 and 22 for descriptives  
  filter(countpersample == 1 & SampleID != 23)  
nrow(mean_agg)
```

```
## [1] 11
```

```
# Run models  
ptsdprop <- rma.mv(prop_pas, prop_pas_v,  
  mods = ~pre3m+post3m,  
  random = ~1 | Effect, data = prosptsd_long)  
summary(ptsdprop)
```

```
##  
## Multivariate Meta-Analysis Model (k = 70; method: REML)  
##  
##   logLik  Deviance      AIC      BIC     AICc  
## 19.4678 -38.9355 -30.9355 -22.1167 -30.2903  
##  
## Variance Components:  
##  
##           estim  sqrt  nlvls  fixed  factor  
## sigma^2    0.0347 0.1864    20     no  Effect  
##  
## Test for Residual Heterogeneity:  
## QE(df = 67) = 866.0971, p-val < .0001  
##  
## Test of Moderators (coefficients 2:3):  
## QM(df = 2) = 253.1018, p-val < .0001  
##  
## Model Results:  
##  
##           estimate      se      zval    pval    ci.lb    ci.ub  
## intrcpt    1.1574 0.0452  25.6145 <.0001  1.0689  1.2460 ***  
## pre3m     -0.1117 0.0084 -13.2627 <.0001 -0.1282 -0.0952 ***  
## post3m    -0.0143 0.0040  -3.6048 0.0003 -0.0221 -0.0065 ***  
##  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ptsdmean <- rma.mv(m_rescaled01, v_rescaled01,  
  mods = ~pre3m+post3m,  
  random = ~1 | Effect, data = prosptsd_long)  
summary(ptsdmean)
```

```
##
## Multivariate Meta-Analysis Model (k = 30; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
## 16.7189 -33.4379 -25.4379 -20.2545 -23.6197
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0125 0.1118    11    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 27) = 705.5635, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 242.8736, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      0.5558  0.0355  15.6663 <.0001    0.4862    0.6253 ***
## pre3m       -0.0602  0.0061  -9.8751 <.0001   -0.0721   -0.0482 ***
## post3m      -0.0167  0.0035  -4.7602 <.0001   -0.0236   -0.0098 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Quality

```
# Create datasets for models, dropping effects with no relevant moderator data
```

```
prosptsd_long_qual_prop <- prosptsd_long[ !(is.na(prosptsd_long$quality))&!(is.na(prosptsd_long$prop_pa
prosptsd_long_qual_mean <- prosptsd_long[ !(is.na(prosptsd_long$quality))&!(is.na(prosptsd_long$v_resca
```

#Frequencies

```
nrow(prosptsd_long_qual_prop) #number of effects
```

```
## [1] 67
```

```
qual_prop_agg <- prosptsd_long_qual_prop %>%
  select(SampleID, quality) %>%
  #exclude ID23 as redundant with 18 and 22 for descriptives
  filter(SampleID != 23) %>%
  group_by(SampleID) %>%
  summarise(mean_qual = mean(quality))
```

```
## Adding missing grouping variables: `Effect`
```

```
nrow(qual_prop_agg)
```

```
## [1] 18
```

```
nrow(prosptsd_long_qual_mean) #number of effects
```

```
## [1] 30
```

```
qual_mean_agg <- prosptsd_long_qual_mean %>%  
  select(SampleID, quality) %>%  
  #exclude ID23 as redundant with 18 and 22 for descriptives  
  filter(SampleID != 23) %>%  
  group_by(SampleID) %>%  
  summarise(mean_qual = mean(quality))
```

```
## Adding missing grouping variables: `Effect`
```

```
nrow(qual_mean_agg)
```

```
## [1] 11
```

```
# Mean center variable for use in moderation analyses  
prosptsd_long_qual_prop <- prosptsd_long_qual_prop %>%  
  mutate(qualMC = quality - mean(qual_prop_agg$mean_qual, na.rm=TRUE))  
prosptsd_long_qual_mean <- prosptsd_long_qual_mean %>%  
  mutate(qualMC = quality - mean(qual_mean_agg$mean_qual, na.rm=TRUE))  
  
# Running models  
ptsdpropquality <- rma.mv(prop_pas, prop_pas_v,  
  mods = ~pre3m*qualMC+post3m*qualMC,  
  random = ~1 | Effect, data = prosptsd_long_qual_prop)  
summary(ptsdpropquality)
```

```
##  
## Multivariate Meta-Analysis Model (k = 67; method: REML)  
##  
##   logLik  Deviance      AIC      BIC      AICc  
## 17.1042 -34.2083 -20.2083  -5.4322 -18.0951  
##  
## Variance Components:  
##  
##           estim  sqrt  nlvls  fixed  factor  
## sigma^2    0.0322  0.1793    19     no  Effect  
##  
## Test for Residual Heterogeneity:  
## QE(df = 61) = 673.4385, p-val < .0001  
##  
## Test of Moderators (coefficients 2:6):  
## QM(df = 5) = 249.7075, p-val < .0001  
##  
## Model Results:  
##  
##           estimate      se      zval  pval  ci.lb  ci.ub  
## intrcpt           1.1914  0.0456  26.1486 <.0001  1.1021  1.2807 ***
```

```
## pre3m          -0.1185  0.0093 -12.7580 <.0001 -0.1367 -0.1003 ***
## qualMC         0.0004  0.0028  0.1571  0.8752 -0.0050  0.0059
## post3m        -0.0168  0.0045 -3.7078  0.0002 -0.0257 -0.0079 ***
## pre3m:qualMC  0.0002  0.0005  0.3380  0.7353 -0.0009  0.0013
## qualMC:post3m -0.0001  0.0002 -0.3784  0.7052 -0.0006  0.0004
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ptsdmeanquality <- rma.mv(m_rescaled01, v_rescaled01,
  mods = ~pre3m*qualMC+post3m*qualMC,
  random = ~1 | Effect, data = prosptsd_long_qual_mean)
summary(ptsdmeanquality)
```

```
##
## Multivariate Meta-Analysis Model (k = 30; method: REML)
##
##   logLik Deviance      AIC      BIC      AICc
## 13.3434 -26.6868 -12.6868 -4.4404 -5.6868
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0127  0.1127    11    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 24) = 514.6145, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 247.0594, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt          0.5502  0.0359  15.3318 <.0001  0.4798  0.6205 ***
## pre3m            -0.0544  0.0071  -7.7015 <.0001 -0.0682 -0.0405 ***
## qualMC           -0.0028  0.0022  -1.2680  0.2048 -0.0071  0.0015
## post3m           -0.0206  0.0044  -4.6935 <.0001 -0.0292 -0.0120 ***
## pre3m:qualMC     0.0007  0.0004   1.9018  0.0572 -0.0000  0.0015 .
## qualMC:post3m   -0.0003  0.0002  -1.2365  0.2163 -0.0007  0.0002
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### Interview vs. self-report
```

```
# Frequencies
```

```
prop_intsr <- propdata[!(is.na(propdata$interview_or_self_report)),]
nrow(prop_intsr)
```

```
## [1] 70
```

```
summary(prop_intsr$interview_or_self_report)
```

```
## Interview Self-report  
##           42           28
```

```
summary(prop_agg$interview_or_self_report)
```

```
## Interview Self-report  
##           11           7
```

```
mean_intsr <- meansdata[!(is.na(meansdata$interview_or_self_report)),]  
nrow(mean_intsr)
```

```
## [1] 30
```

```
summary(mean_intsr$interview_or_self_report)
```

```
## Interview Self-report  
##           17           13
```

```
summary(mean_agg$interview_or_self_report)
```

```
## Interview Self-report  
##           7           4
```

```
# Run models
```

```
ptsdpropintsr <- rma.mv(prop_pas, prop_pas_v,  
  mods = ~pre3m*interview_or_self_report+post3m*  
  interview_or_self_report,  
  random = ~1 | Effect, data = prosptsd_long)  
summary(ptsdpropintsr)
```

```
##  
## Multivariate Meta-Analysis Model (k = 70; method: REML)  
##  
## logLik Deviance AIC BIC AICc  
## 17.3562 -34.7124 -20.7124 -5.6003 -18.7124  
##  
## Variance Components:  
##  
## estim sqrt nlvls fixed factor  
## sigma^2 0.0381 0.1952 20 no Effect  
##  
## Test for Residual Heterogeneity:  
## QE(df = 64) = 834.8897, p-val < .0001  
##  
## Test of Moderators (coefficients 2:6):  
## QM(df = 5) = 256.0815, p-val < .0001  
##
```

```
## Model Results:
##
##
##           estimate      se      zval
## intrcpt           1.1873  0.0611  19.4435
## pre3m            -0.1241  0.0119 -10.4354
## interview_or_self_reportSelf-report -0.0706  0.0958  -0.7376
## post3m           -0.0129  0.0069  -1.8654
## pre3m:interview_or_self_reportSelf-report 0.0277  0.0173   1.6029
## interview_or_self_reportSelf-report:post3m -0.0016  0.0085  -0.1865
##
##           pval      ci.lb      ci.ub
## intrcpt      <.0001  1.0676   1.3069  ***
## pre3m        <.0001 -0.1474  -0.1008  ***
## interview_or_self_reportSelf-report 0.4607 -0.2584  0.1171
## post3m       0.0621 -0.0265  0.0007
## pre3m:interview_or_self_reportSelf-report 0.1090 -0.0062  0.0615
## interview_or_self_reportSelf-report:post3m 0.8521 -0.0182  0.0150
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ptsdmeanintsr <- rma.mv(m_rescaled01, v_rescaled01,
  mods = ~pre3m*interview_or_self_report+post3m*
  interview_or_self_report,
  random = ~1 | Effect, data = prosptsd_long)
summary(ptsdmeanintsr)
```

```
##
## Multivariate Meta-Analysis Model (k = 30; method: REML)
##
##   logLik Deviance      AIC      BIC      AICc
## 12.6719 -25.3439 -11.3439  -3.0975  -4.3439
##
## Variance Components:
##
##           estim      sqrt  nlvls  fixed  factor
## sigma^2    0.0111  0.1056     11     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 24) = 523.3709, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 246.4526, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval
## intrcpt           0.5144  0.0427  12.0505
## pre3m            -0.0553  0.0088  -6.2738
## interview_or_self_reportSelf-report 0.1081  0.0698   1.5497
## post3m           -0.0192  0.0045  -4.2649
## pre3m:interview_or_self_reportSelf-report -0.0089  0.0122  -0.7232
## interview_or_self_reportSelf-report:post3m 0.0069  0.0078   0.8842
##
##           pval      ci.lb      ci.ub
## intrcpt      <.0001  0.4307   0.5981  ***
```



```
## pre3m <.0001 -0.0726 -0.0380 ***
## interview_or_self_reportSelf-report 0.1212 -0.0286 0.2449
## post3m <.0001 -0.0281 -0.0104 ***
## pre3m:interview_or_self_reportSelf-report 0.4695 -0.0328 0.0151
## interview_or_self_reportSelf-report:post3m 0.3766 -0.0084 0.0221
##
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# PTSD diagnosis determination
```

```
# Frequencies
```

```
prop_cut <- propdata[!(is.na(propdata$prop_cutscore)),]
nrow(prop_cut)
```

```
## [1] 70
```

```
summary(prop_cut$prop_cutscore)
```

```
## N Y
## 60 10
```

```
summary(prop_agg$prop_cutscore)
```

```
## N Y
## 15 3
```

```
# Run models
```

```
ptsdpropcut <- rma.mv(prop_pas, prop_pas_v,
  mods = ~pre3m*prop_cutscore+post3m*
  prop_cutscore, random = ~1 | Effect,
  data = prosptsd_long)
summary(ptsdpropcut)
```

```
##
## Multivariate Meta-Analysis Model (k = 70; method: REML)
##
## logLik Deviance AIC BIC AICc
## 16.5759 -33.1518 -19.1518 -4.0396 -17.1518
##
## Variance Components:
##
## estim sqrt nlvls fixed factor
## sigma^2 0.0345 0.1858 20 no Effect
##
## Test for Residual Heterogeneity:
## QE(df = 64) = 791.5288, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 254.4265, p-val < .0001
##
```

```
## Model Results:
##
##           estimate      se      zval      pval      ci.lb
## intrcpt           1.1404  0.0491  23.2454 <.0001  1.0442
## pre3m             -0.1132  0.0094 -12.0467 <.0001 -0.1316
## prop_cutscoreY    0.1151  0.1245   0.9242  0.3554 -0.1290
## post3m            -0.0139  0.0042  -3.3261  0.0009 -0.0221
## pre3m:prop_cutscoreY 0.0082  0.0216   0.3802  0.7038 -0.0341
## prop_cutscoreY:post3m -0.0039  0.0135  -0.2911  0.7710 -0.0303
##           ci.lb
## intrcpt           1.2365 ***
## pre3m             -0.0948 ***
## prop_cutscoreY    0.3592
## post3m            -0.0057 ***
## pre3m:prop_cutscoreY 0.0505
## prop_cutscoreY:post3m 0.0225
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### DSM version
```

```
# Frequencies
```

```
prop_dsm <- propdata[ !(is.na(propdata$DSM_version)),]
nrow(prop_dsm)
```

```
## [1] 70
```

```
summary(prop_dsm$DSM_version)
```

```
##   III III-R   IV
##   11    21   38
```

```
summary(prop_agg$DSM_version)
```

```
##   III III-R   IV
##    2    3   13
```

```
mean_dsm <- meansdata[ !(is.na(meansdata$DSM_version)),]
nrow(mean_dsm)
```

```
## [1] 30
```

```
summary(mean_dsm$DSM_version)
```

```
##   III III-R   IV
##    4    3   23
```

```
summary(mean_agg$DSM_version)
```

```
##   III III-R   IV  
##   1     1     9
```

```
# Run models
```

```
ptsdpropdsm <- rma.mv(prop_pas, prop_pas_v,  
                      mods = ~pre3m*DSM_version+post3m*DSM_version,  
                      random = ~1 | Effect, data = prosptsd_long)  
summary(ptsdpropdsm)
```

```
##  
## Multivariate Meta-Analysis Model (k = 70; method: REML)  
##  
##   logLik Deviance      AIC      BIC      AICc  
## 19.9373 -39.8745 -19.8745  1.2342 -15.4745  
##  
## Variance Components:  
##  
##           estim  sqrt  nlvls  fixed  factor  
## sigma^2    0.0368 0.1918    20     no  Effect  
##  
## Test for Residual Heterogeneity:  
## QE(df = 61) = 742.5098, p-val < .0001  
##  
## Test of Moderators (coefficients 2:9):  
## QM(df = 8) = 267.2750, p-val < .0001  
##  
## Model Results:  
##  
##           estimate      se      zval      pval      ci.lb  
## intrcpt           1.0660 0.1168  9.1264 <.0001  0.8370  
## pre3m             -0.0658 0.0209 -3.1469 0.0017 -0.1068  
## DSM_versionIII-R  0.0951 0.1639  0.5804 0.5616 -0.2261  
## DSM_versionIV    0.1170 0.1299  0.9014 0.3674 -0.1375  
## post3m            -0.0203 0.0082 -2.4801 0.0131 -0.0363  
## pre3m:DSM_versionIII-R -0.0728 0.0268 -2.7130 0.0067 -0.1254  
## pre3m:DSM_versionIV  -0.0520 0.0237 -2.1933 0.0283 -0.0986  
## DSM_versionIII-R:post3m 0.0385 0.0159  2.4277 0.0152  0.0074  
## DSM_versionIV:post3m  0.0001 0.0096  0.0149 0.9881 -0.0186  
##           ci.ub  
## intrcpt           1.2949 ***  
## pre3m             -0.0248 **  
## DSM_versionIII-R  0.4164  
## DSM_versionIV    0.3716  
## post3m            -0.0042 *  
## pre3m:DSM_versionIII-R -0.0202 **  
## pre3m:DSM_versionIV  -0.0055 *  
## DSM_versionIII-R:post3m 0.0696 *  
## DSM_versionIV:post3m  0.0189  
##  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ptsdmeandsm <- rma.mv(m_rescaled01, v_rescaled01,
                    mods = ~pre3m*DSM_version+post3m*DSM_version,
                    random = ~1 | Effect, data = prosptsd_long)
summary(ptsdmeandsm)
```

```
##
## Multivariate Meta-Analysis Model (k = 30; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
##   8.7597 -17.5194   2.4806  12.9258  24.4806
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0139 0.1179    11    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 21) = 624.4000, p-val < .0001
##
## Test of Moderators (coefficients 2:9):
## QM(df = 8) = 249.0880, p-val < .0001
##
## Model Results:
##
##              estimate      se      zval      pval      ci.lb
## intrcpt              0.6153 0.1234   4.9870 <.0001   0.3735
## pre3m                -0.0620 0.0168  -3.6920 0.0002  -0.0949
## DSM_versionIII-R    -0.0341 0.1875  -0.1818 0.8558  -0.4016
## DSM_versionIV       -0.0630 0.1301  -0.4847 0.6279  -0.3180
## post3m              -0.0111 0.0068  -1.6290 0.1033  -0.0244
## pre3m:DSM_versionIII-R -0.1068 0.0767  -1.3926 0.1637  -0.2570
## pre3m:DSM_versionIV   0.0087 0.0183   0.4756 0.6344  -0.0272
## DSM_versionIII-R:post3m 1.7287 1.5167   1.1397 0.2544  -1.2441
## DSM_versionIV:post3m  -0.0097 0.0080  -1.2075 0.2272  -0.0254
##
##              ci.ub
## intrcpt              0.8572 ***
## pre3m                -0.0291 ***
## DSM_versionIII-R     0.3334
## DSM_versionIV        0.1919
## post3m               0.0023
## pre3m:DSM_versionIII-R 0.0435
## pre3m:DSM_versionIV   0.0446
## DSM_versionIII-R:post3m 4.7014
## DSM_versionIV:post3m  0.0060
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### Country
```

```
# Frequencies
```

```
prop_us <- propdata[!(is.na(propdata$us)),]
prop_us$us <-as.factor(prop_us$us)
```

```
nrow(prop_us)
```

```
## [1] 70
```

```
summary(prop_us$us)
```

```
## 0 1  
## 16 54
```

```
prop_agg$us <-as.factor(prop_agg$us)  
summary(prop_agg$us)
```

```
## 0 1  
## 6 12
```

```
mean_us <- meansdata[!(is.na(meansdata$us)),]  
mean_us$us <-as.factor(mean_us$us)  
nrow(mean_us)
```

```
## [1] 30
```

```
summary(mean_us$us)
```

```
## 0 1  
## 4 26
```

```
mean_agg$us <-as.factor(mean_agg$us)  
summary(mean_agg$us)
```

```
## 0 1  
## 2 9
```

```
# Run models
```

```
ptsdpropcountry <- rma.mv(prop_pas, prop_pas_v,  
                        mods = ~pre3m*us+post3m*us,  
                        random = ~1 | Effect, data = prosptsd_long)  
summary(ptsdpropcountry)
```

```
##  
## Multivariate Meta-Analysis Model (k = 70; method: REML)  
##  
##   logLik Deviance      AIC      BIC      AICc  
## 17.5458 -35.0916 -21.0916  -5.9794 -19.0916  
##  
## Variance Components:  
##  
##      estim  sqrt  nlvls  fixed  factor  
## sigma^2  0.0369 0.1920   20    no  Effect
```

```
##
## Test for Residual Heterogeneity:
## QE(df = 64) = 820.7063, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 256.6851, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      1.1025  0.0888  12.4121 <.0001   0.9284   1.2766 ***
## pre3m       -0.0807  0.0187  -4.3175 <.0001  -0.1173  -0.0441 ***
## us           0.0654  0.1043   0.6273  0.5305  -0.1390   0.2699
## post3m      -0.0165  0.0050  -3.3125 0.0009  -0.0262  -0.0067 ***
## pre3m:us    -0.0391  0.0211  -1.8521 0.0640  -0.0804   0.0023 .
## us:post3m   0.0036  0.0084   0.4350 0.6636  -0.0128   0.0201
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ptsdmeancountry <- rma.mv(m_rescaled01, v_rescaled01,
                          mods = ~pre3m*us+post3m*us,
                          random = ~1 | Effect, data = prosptsd_long)
summary(ptsdmeancountry)
```

```
##
## Multivariate Meta-Analysis Model (k = 30; method: REML)
##
##      logLik Deviance      AIC      BIC      AICc
## 18.4349 -36.8697 -24.8697 -17.5564 -20.2030
##
## Variance Components:
##
##      estim      sqrt  nlvls  fixed  factor
## sigma^2  0.0067  0.0821    11    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 25) = 266.7808, p-val < .0001
##
## Test of Moderators (coefficients 2:5):
## QM(df = 4) = 260.4218, p-val < .0001
##
## Model Results:
##
##      estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt      0.3643  0.0616   5.9141 <.0001   0.2436   0.4850 ***
## pre3m       -0.0455  0.0094  -4.8326 <.0001  -0.0640  -0.0271 ***
## us           0.2416  0.0686   3.5232 0.0004   0.1072   0.3761 ***
## post3m      -0.0173  0.0035  -4.8887 <.0001  -0.0242  -0.0104 ***
## pre3m:us    -0.0224  0.0107  -2.0940 0.0363  -0.0433  -0.0014 *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

### Age

# Create datasets for models, dropping effects with no relevant moderator data
prosptsd_long_age_prop <- prosptsd_long[ !(is.na(prosptsd_long$mean_age)) &! (is.na(prosptsd_long$prop_pa
prosptsd_long_age_mean <- prosptsd_long[ !(is.na(prosptsd_long$mean_age)) &! (is.na(prosptsd_long$v_resca

# Frequencies
nrow(prosptsd_long_age_prop) #number of effects

```

```
## [1] 64
```

```

age_prop_agg <- prosptsd_long_age_prop %>%
  select(SampleID, mean_age) %>%
  #exclude ID23 as redundant with 18 and 22 for descriptives
  filter(SampleID != 23) %>%
  group_by(SampleID) %>%
  summarise(mean_age = mean(mean_age))

```

```
## Adding missing grouping variables: `Effect`
```

```
nrow(age_prop_agg)
```

```
## [1] 16
```

```
nrow(prosptsd_long_age_mean) #number of effects
```

```
## [1] 24
```

```

age_mean_agg <- prosptsd_long_age_mean %>%
  select(SampleID, mean_age) %>%
  #exclude ID23 as redundant with 18 and 22 for descriptives
  filter(SampleID != 23) %>%
  group_by(SampleID) %>%
  summarise(mean_age = mean(mean_age))

```

```
## Adding missing grouping variables: `Effect`
```

```
nrow(age_mean_agg)
```

```
## [1] 9
```

```

# Mean center variable for use in moderation analyses
prosptsd_long_age_prop <- prosptsd_long_age_prop %>%
  mutate(ageMC = mean_age - mean(age_prop_agg$mean_age, na.rm=TRUE))

prosptsd_long_age_mean <- prosptsd_long_age_mean %>%
  mutate(ageMC = mean_age - mean(age_mean_agg$mean_age, na.rm=TRUE))

# Run models

```

```
ptsdpropage <- rma.mv(prop_pas, prop_pas_v,
  mods = ~ageMC*pre3m+ageMC*post3m,
  random = ~1 | Effect, data = prosptsd_long_age_prop)
summary(ptsdpropage)
```

```
##
## Multivariate Meta-Analysis Model (k = 64; method: REML)
##
##   logLik Deviance      AIC      BIC      AICc
## 17.0670 -34.1340 -20.1340  -5.7109 -17.8940
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0374 0.1935    18    no Effect
##
## Test for Residual Heterogeneity:
## QE(df = 58) = 713.7797, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 242.9021, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt           1.1471 0.0495  23.1581 <.0001  1.0500  1.2441 ***
## ageMC              0.0042 0.0089   0.4678 0.6399 -0.0133  0.0217
## pre3m             -0.1047 0.0091 -11.4912 <.0001 -0.1225 -0.0868 ***
## post3m            -0.0168 0.0054  -3.0990 0.0019 -0.0274 -0.0062 **
## ageMC:pre3m      -0.0056 0.0018  -3.1493 0.0016 -0.0090 -0.0021 **
## ageMC:post3m    -0.0006 0.0015  -0.4203 0.6742 -0.0035  0.0023
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ptsdmeanage <- rma.mv(m_rescaled01, v_rescaled01,
  mods = ~ageMC*pre3m+ageMC*post3m,
  random = ~1 | Effect, data = prosptsd_long_age_mean)
summary(ptsdmeanage)
```

```
##
## Multivariate Meta-Analysis Model (k = 24; method: REML)
##
##   logLik Deviance      AIC      BIC      AICc
##  6.5952 -13.1905   0.8095   7.0421  12.0095
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0063 0.0795     9    no Effect
##
## Test for Residual Heterogeneity:
```



```

## QE(df = 18) = 199.0932, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 176.2297, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt          0.5822  0.0297  19.5806 <.0001   0.5239   0.6404 ***
## ageMC            0.0144  0.0086   1.6615  0.0966  -0.0026   0.0313 .
## pre3m           -0.0622  0.0077  -8.0774 <.0001  -0.0773  -0.0471 ***
## post3m          -0.0201  0.0042  -4.7460 <.0001  -0.0284  -0.0118 ***
## ageMC:pre3m     -0.0030  0.0025  -1.1934  0.2327  -0.0079   0.0019
## ageMC:post3m    0.0002  0.0026   0.0768  0.9388  -0.0048   0.0052
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
## AGE- Plotting significant interaction for means models (Appendix H)
```

```
#Create objects with mean, +/- 1SD - MEANS
```

```

meanage <- mean(prosptsd_long_age_mean$ageMC, na.rm=TRUE)
neg1sd <- mean(prosptsd_long_age_mean$ageMC,
               na.rm=TRUE)-sd(prosptsd_long_age_mean$ageMC)
pos1sd <- mean(prosptsd_long_age_mean$ageMC,
               na.rm=TRUE)+sd(prosptsd_long_age_mean$ageMC)

```

```
#per https://stat.ethz.ch/pipermail/r-help/2011-September/289312.html
```

```
## Specify each point of interest - MEANS
```

```

p1 <- as.data.frame(predict(ptsdmeanage, newmods=c(meanage, 0,0,0*meanage,0), addx=T))
p2 <- as.data.frame(predict(ptsdmeanage, newmods=c(meanage, 1,0,1*meanage,0), addx=T))
p3 <- as.data.frame(predict(ptsdmeanage, newmods=c(meanage, 3,0,3*meanage,0), addx=T))
p4 <- as.data.frame(predict(ptsdmeanage, newmods=c(meanage, 3,3,3*meanage,3*meanage), addx=T))
p5 <- as.data.frame(predict(ptsdmeanage, newmods=c(meanage, 3,6,3*meanage,6*meanage), addx=T))
p6 <- as.data.frame(predict(ptsdmeanage, newmods=c(meanage, 3,9,3*meanage,9*meanage), addx=T))

p7 <- as.data.frame(predict(ptsdmeanage, newmods=c(neg1sd, 0,0,0*neg1sd,0), addx=T))
p8 <- as.data.frame(predict(ptsdmeanage, newmods=c(neg1sd, 1,0,1*neg1sd,0), addx=T))
p9 <- as.data.frame(predict(ptsdmeanage, newmods=c(neg1sd, 3,0,3*neg1sd,0), addx=T))
p10<- as.data.frame(predict(ptsdmeanage, newmods=c(neg1sd, 3,3,3*neg1sd,3*neg1sd), addx=T))
p11<- as.data.frame(predict(ptsdmeanage, newmods=c(neg1sd, 3,6,3*neg1sd,6*neg1sd), addx=T))
p12<- as.data.frame(predict(ptsdmeanage, newmods=c(neg1sd, 3,9,3*neg1sd,9*neg1sd), addx=T))

p13<- as.data.frame(predict(ptsdmeanage, newmods=c(pos1sd, 0,0,0*pos1sd,0), addx=T))
p14<- as.data.frame(predict(ptsdmeanage, newmods=c(pos1sd, 1,0,1*pos1sd,0), addx=T))
p15<- as.data.frame(predict(ptsdmeanage, newmods=c(pos1sd, 3,0,3*pos1sd,0), addx=T))
p16<- as.data.frame(predict(ptsdmeanage, newmods=c(pos1sd, 3,3,3*pos1sd,3*pos1sd), addx=T))
p17<- as.data.frame(predict(ptsdmeanage, newmods=c(pos1sd, 3,6,3*pos1sd,6*pos1sd), addx=T))
p18<- as.data.frame(predict(ptsdmeanage, newmods=c(pos1sd, 3,9,3*pos1sd,9*pos1sd), addx=T))

```

```
# Create combined dataset - MEANS
```

```

predicted_vals_mean <-
  rbind(p1, p2, p3, p4, p5, p6, p7, p8, p9, p10,
        p11, p12, p13, p14, p15, p16, p17, p18) %>%

```

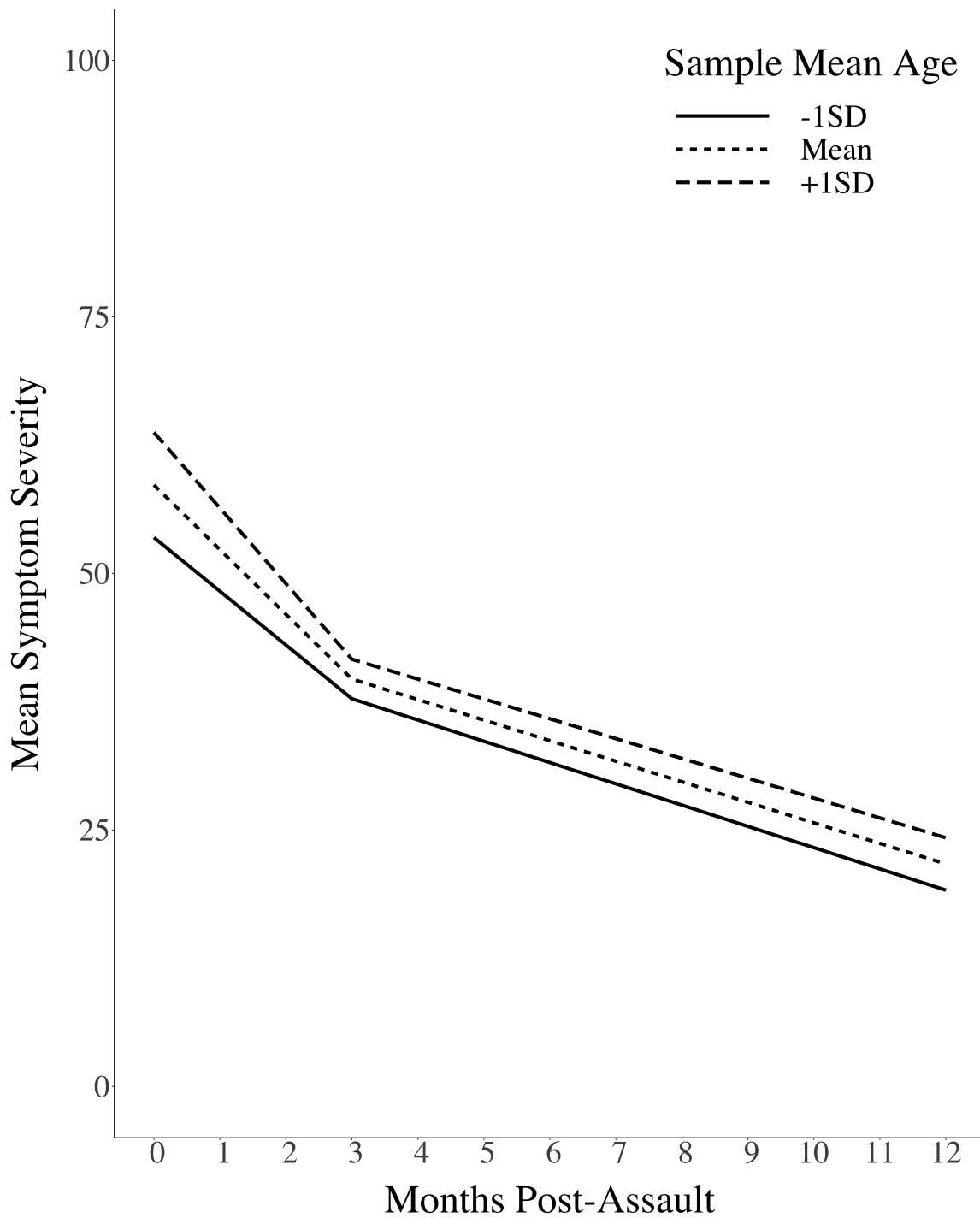
```

mutate(agetype = case_when(X.ageMC==neg1sd ~ "-1SD",
                           X.ageMC==meanage ~ "Mean",
                           X.ageMC==pos1sd ~ "+1SD"),
       months = X.pre3m + X.post3m)

predicted_vals_mean$agetype <- factor(predicted_vals_mean$agetype,
                                     levels=c("-1SD", "Mean", "+1SD"))

## Plot - MEANS
meanagemod <- ggplot() +
  geom_line(data=predicted_vals_mean, aes(x=months,
                                          y=pred*100, linetype=agetype), size=2)+
  labs(x= "Months Post-Assault", y="Mean Symptom Severity")+
  scale_linetype_discrete(name = "Sample Mean Age") +
  theme_classic() +
  theme(text = element_text(family = "Times", size=48)) +
  theme(axis.title.y = element_text(margin =
                                    margin(t = 0, r = 20, b = 0, l = 0))) +
  theme(axis.title.x = element_text(margin =
                                    margin(t = 20, r = 0, b = 0, l = 0))) +
  theme(legend.key.height=unit(2.8, "line"))+
  theme(legend.key.width = unit(5, "cm"))+
  theme(plot.margin=unit(c(2,2,2,2), "cm"))+
  theme(legend.box.margin=unit(c(2,2,2,2), "cm"))+
  theme(legend.position=c(0.8,0.9))+
  ylim(0,100)+
  coord_cartesian(clip = "off")+
  scale_x_discrete(limits=c(0,1,2,3,4,5,6,7,8,9,10,11,12))
ggsave(meanagemod, file='agemeanmod.png', width = 18, height = 22, dpi=300)
include_graphics('agemeanmod.png')

```



```
## AGE- Plotting significant interaction for point prevalence models (Appendix H)
# Create objects with mean, +/- 1SD - PREVALENCE
meanage <- mean(prosptsd_long_age_prop$ageMC, na.rm=TRUE)
neg1sd <- mean(prosptsd_long_age_prop$ageMC,
```

```

      na.rm=TRUE)-sd(prosptsd_long_age_prop$ageMC)
pos1sd <- mean(prosptsd_long_age_prop$ageMC,
      na.rm=TRUE)+sd(prosptsd_long_age_prop$ageMC)

#per https://stat.ethz.ch/pipermail/r-help/2011-September/289312.html
# Specify each point of interest - PREVALENCE
p1 <- as.data.frame(predict(ptsdpropage, newmods=c(meanage, 0,0,0*meanage,0), addx=T))
p2 <- as.data.frame(predict(ptsdpropage, newmods=c(meanage, 1,0,1*meanage,0), addx=T))
p3 <- as.data.frame(predict(ptsdpropage, newmods=c(meanage, 3,0,3*meanage,0), addx=T))
p4 <- as.data.frame(predict(ptsdpropage, newmods=c(meanage, 3,3,3*meanage,3*meanage), addx=T))
p5 <- as.data.frame(predict(ptsdpropage, newmods=c(meanage, 3,6,3*meanage,6*meanage), addx=T))
p6 <- as.data.frame(predict(ptsdpropage, newmods=c(meanage, 3,9,3*meanage,9*meanage), addx=T))

p7 <- as.data.frame(predict(ptsdpropage, newmods=c(neg1sd, 0,0,0*neg1sd,0), addx=T))
p8 <- as.data.frame(predict(ptsdpropage, newmods=c(neg1sd, 1,0,1*neg1sd,0), addx=T))
p9 <- as.data.frame(predict(ptsdpropage, newmods=c(neg1sd, 3,0,3*neg1sd,0), addx=T))
p10<- as.data.frame(predict(ptsdpropage, newmods=c(neg1sd, 3,3,3*neg1sd,3*neg1sd), addx=T))
p11<- as.data.frame(predict(ptsdpropage, newmods=c(neg1sd, 3,6,3*neg1sd,6*neg1sd), addx=T))
p12<- as.data.frame(predict(ptsdpropage, newmods=c(neg1sd, 3,9,3*neg1sd,9*neg1sd), addx=T))

p13<- as.data.frame(predict(ptsdpropage, newmods=c(pos1sd, 0,0,0*pos1sd,0), addx=T))
p14<- as.data.frame(predict(ptsdpropage, newmods=c(pos1sd, 1,0,1*pos1sd,0), addx=T))
p15<- as.data.frame(predict(ptsdpropage, newmods=c(pos1sd, 3,0,3*pos1sd,0), addx=T))
p16<- as.data.frame(predict(ptsdpropage, newmods=c(pos1sd, 3,3,3*pos1sd,3*pos1sd), addx=T))
p17<- as.data.frame(predict(ptsdpropage, newmods=c(pos1sd, 3,6,3*pos1sd,6*pos1sd), addx=T))
p18<- as.data.frame(predict(ptsdpropage, newmods=c(pos1sd, 3,9,3*pos1sd,9*pos1sd), addx=T))

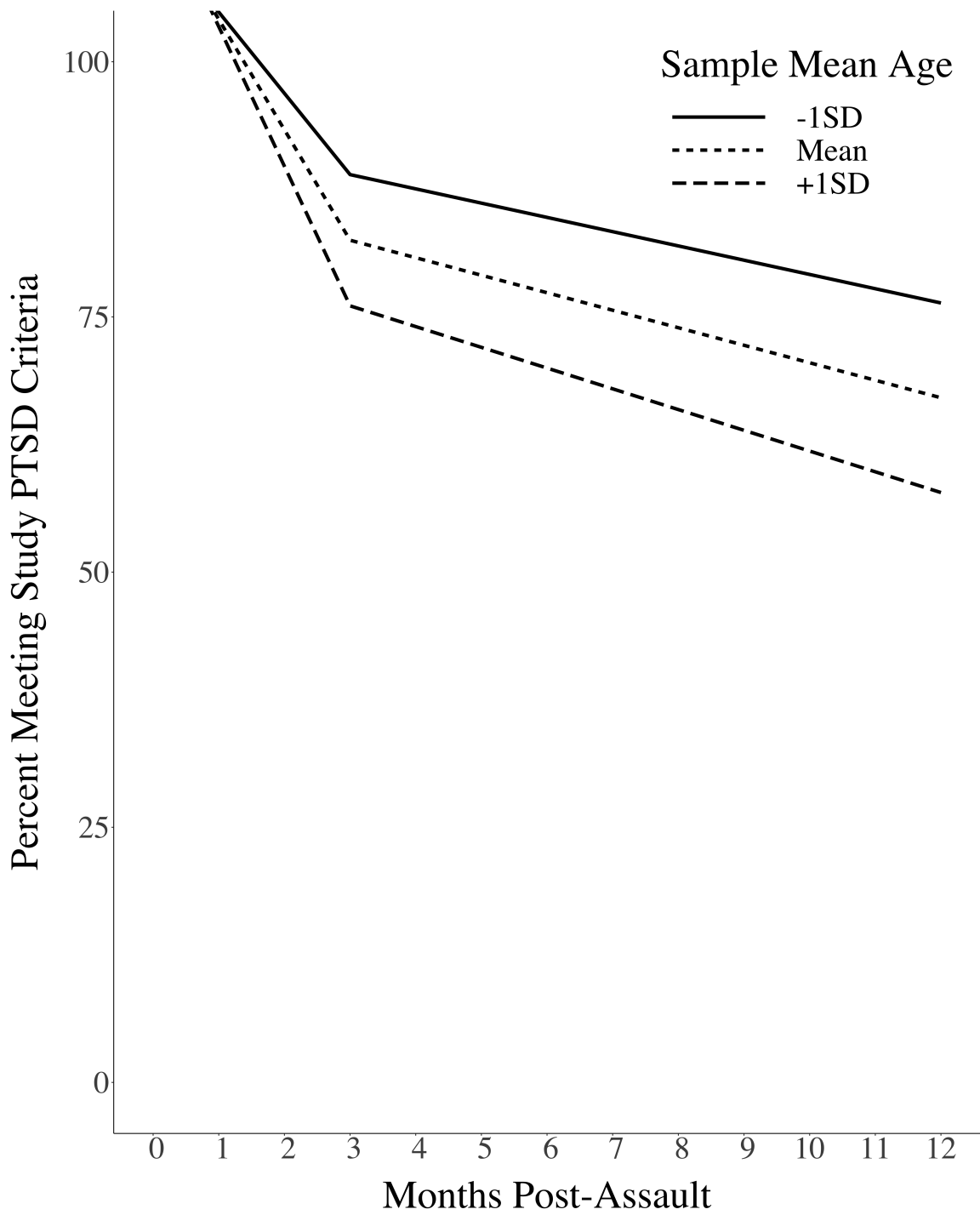
# Create combined dataset - PREVALENCE
predicted_vals_prop <-
  rbind(p1, p2, p3, p4, p5, p6, p7, p8, p9,
        p10, p11, p12, p13, p14, p15, p16, p17, p18) %>%
  mutate(agetype = case_when(X.ageMC==neg1sd ~ "-1SD",
                             X.ageMC==meanage ~ "Mean",
                             X.ageMC==pos1sd ~ "+1SD"),
         months = X.pre3m + X.post3m)

predicted_vals_prop$agetype <- factor(predicted_vals_prop$agetype,
                                     levels=c("-1SD", "Mean", "+1SD"))

# Plot - PREVALENCE
propagemod <- ggplot() +
  geom_line(data=predicted_vals_prop, aes(x=months,
                                          y=pred*100, linetype=agetype), size=2)+
  labs(x= "Months Post-Assault", y="Percent Meeting Study PTSD Criteria") +
  scale_linetype_discrete(name = "Sample Mean Age") +
  coord_cartesian(ylim=c(0,100), clip = "on")+
  theme_classic() +
  theme(text = element_text(family = "Times", size=48)) +
  theme(axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) +
  theme(axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) +
  theme(legend.key.height=unit(2.8, "line"))+

```

```
theme(legend.key.width = unit(5,"cm"))+
theme(plot.margin=unit(c(2,2,2,2),"cm"))+
theme(legend.box.margin=unit(c(2,2,2,2),"cm"))+
theme(legend.position=c(0.8,0.9))+
# ylim(0,100)+
scale_x_discrete(limits=c(0,1,2,3,4,5,6,7,8,9,10,11,12))
ggsave(propagemod, file='propagemod.png', width = 18, height = 22, dpi=300)
include_graphics('propagemod.png')
```



```
### Gender
```

```
# Create datasets for models, dropping effects with no relevant moderator data
prosptsd_long_women_prop <- prosptsd_long[ !(
  is.na(prosptsd_long$percent_women))&!(
```

```

  is.na(prosptsd_long$prop_pas_v)),]
prosptsd_long_women_mean <- prosptsd_long[ !(
  is.na(prosptsd_long$percent_women))&!(
  is.na(prosptsd_long$v_rescaled01)),]

# Frequencies
nrow(prosptsd_long_women_prop) #number of effects

```

```
## [1] 64
```

```

women_prop_agg <- prosptsd_long_women_prop %>%
  select(SampleID, percent_women) %>%
  #exclude ID23 as redundant with 18 and 22 for descriptives
  filter(SampleID != 23) %>%
  group_by(SampleID) %>%
  summarise(mean_women = mean(percent_women))

```

```
## Adding missing grouping variables: `Effect`
```

```
nrow(women_prop_agg)
```

```
## [1] 17
```

```
nrow(prosptsd_long_women_mean) #number of effects
```

```
## [1] 30
```

```

women_mean_agg <- prosptsd_long_women_mean %>%
  select(SampleID, percent_women) %>%
  #exclude ID23 as redundant with 18 and 22 for descriptives
  filter(SampleID != 23) %>%
  group_by(SampleID) %>%
  summarise(mean_women = mean(percent_women))

```

```
## Adding missing grouping variables: `Effect`
```

```
nrow(women_mean_agg)
```

```
## [1] 11
```

```

# Mean center variable for use in moderation analyses
prosptsd_long_women_prop <- prosptsd_long_women_prop %>%
  mutate(womenMC = percent_women -
    mean(women_prop_agg$mean_women, na.rm=TRUE))
prosptsd_long_women_mean <- prosptsd_long_women_mean %>%
  mutate(womenMC = percent_women -
    mean(women_mean_agg$mean_women, na.rm=TRUE))

```

```
#Running models
```

```
ptsdpropwomen <- rma.mv(prop_pas, prop_pas_v,
  mods = ~pre3m*womenMC+post3m*womenMC,
  random = ~1 | Effect, data = prosptsd_long_women_prop)
summary(ptsdpropwomen)
```

```
##
## Multivariate Meta-Analysis Model (k = 64; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
## 17.7202 -35.4404 -21.4404  -7.0173 -19.2004
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0360 0.1897    19    no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 58) = 755.8216, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 266.6897, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt           1.1689 0.0473  24.7248 <.0001  1.0763  1.2616 ***
## pre3m             -0.1178 0.0087 -13.4677 <.0001 -0.1350 -0.1007 ***
## womenMC           -0.0100 0.0139  -0.7194 0.4719 -0.0371  0.0172
## post3m            -0.0163 0.0046  -3.5502 0.0004 -0.0253 -0.0073 ***
## pre3m:womenMC    -0.0031 0.0026  -1.1805 0.2378 -0.0083  0.0021
## womenMC:post3m   0.0001 0.0008   0.0752 0.9401 -0.0016  0.0017
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ptsdmeanwomen <- rma.mv(m_rescaled01, v_rescaled01,
  mods = ~pre3m*womenMC+post3m*womenMC,
  random = ~1 | Effect, data = prosptsd_long_women_mean)
summary(ptsdmeanwomen)
```

```
##
## Multivariate Meta-Analysis Model (k = 30; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
## 16.7189 -33.4379 -25.4379 -20.2545 -23.6197
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0125 0.1118    11    no  Effect
##
## Test for Residual Heterogeneity:
```



```

## QE(df = 27) = 705.5635, p-val < .0001
##
## Test of Moderators (coefficients 2:3):
## QM(df = 2) = 242.8736, p-val < .0001
##
## Model Results:
##
##          estimate      se      zval   pval   ci.lb   ci.ub
## intrcpt    0.5558  0.0355  15.6663 <.0001  0.4862  0.6253 ***
## pre3m     -0.0602  0.0061  -9.8751 <.0001 -0.0721 -0.0482 ***
## post3m    -0.0167  0.0035  -4.7602 <.0001 -0.0236 -0.0098 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
### % Stranger perpetrators
```

```

# Create datasets for models, dropping effects with no relevant moderator data
prosptsd_long_strang_prop <- prosptsd_long[
  !(is.na(prosptsd_long$percent_stranger)) &
  !(is.na(prosptsd_long$prop_pas_v)),]
prosptsd_long_strang_mean <- prosptsd_long[
  !(is.na(prosptsd_long$percent_stranger)) &
  !(is.na(prosptsd_long$v_rescaled01)),]

```

```
# Frequencies
```

```
nrow(prosptsd_long_strang_prop) #number of effects
```

```
## [1] 38
```

```

strang_prop_agg <- prosptsd_long_strang_prop %>%
  select(SampleID, percent_stranger) %>%
  #exclude ID23 as redundant with 18 and 22 for descriptives
  filter(SampleID != 23) %>%
  group_by(SampleID) %>%
  summarise(mean_strang = mean(percent_stranger))

```

```
## Adding missing grouping variables: `Effect`
```

```
nrow(strang_prop_agg)
```

```
## [1] 10
```

```
nrow(prosptsd_long_strang_mean) #number of effects
```

```
## [1] 13
```

```

strang_mean_agg <- prosptsd_long_strang_mean %>%
  select(SampleID, percent_stranger) %>%
  #exclude ID23 as redundant with 18 and 22 for descriptives
  filter(SampleID != 23) %>%
  group_by(SampleID) %>%
  summarise(mean_strang = mean(percent_stranger))

```

```
## Adding missing grouping variables: `Effect`
```

```
nrow(strang_mean_agg)
```

```
## [1] 5
```

```
# Mean center variable for use in moderation analyses
prosptsd_long_strang_prop <- prosptsd_long_strang_prop %>%
  mutate(strangMC = percent_stranger -
         mean(strang_prop_agg$mean_strang, na.rm=TRUE))
prosptsd_long_strang_mean <- prosptsd_long_strang_mean %>%
  mutate(strangMC = percent_stranger -
         mean(strang_mean_agg$mean_strang, na.rm=TRUE))

# Run models
ptsdpropstranger <- rma.mv(prop_pas, prop_pas_v,
  mods = ~pre3m*strangMC+post3m*strangMC,
  random = ~1 | Effect, data = prosptsd_long_strang_prop)
summary(ptsdpropstranger)
```

```
##
## Multivariate Meta-Analysis Model (k = 38; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
##   1.9252  -3.8504   10.1496   20.4098   14.8163
##
## Variance Components:
##
##           estim      sqrt  nlvls  fixed  factor
## sigma^2    0.0566  0.2380     12     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 32) = 648.0057, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 153.0627, p-val < .0001
##
## Model Results:
##
##           estimate      se      zval      pval      ci.lb      ci.ub
## intrcpt           1.1404  0.0716  15.9293 <.0001    1.0001    1.2808 ***
## pre3m             -0.1040  0.0116  -8.9402 <.0001   -0.1268   -0.0812 ***
## strangMC          -0.0038  0.0041  -0.9407  0.3469   -0.0118    0.0041
## post3m            -0.0144  0.0067  -2.1505  0.0315   -0.0276   -0.0013 *
## pre3m:strangMC     0.0009  0.0007   1.2514  0.2108   -0.0005    0.0022
## strangMC:post3m    0.0006  0.0006   0.9510  0.3416   -0.0006    0.0019
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ptsdmeanstranger <- rma.mv(m_rescaled01, v_rescaled01,
  mods = ~pre3m*strangMC+post3m*strangMC,
```

```
random = ~1 | Effect, data = prosptsd_long_strang_mean)
summary(ptsdmeanstranger)
```

```
##
## Multivariate Meta-Analysis Model (k = 13; method: REML)
##
##   logLik  Deviance      AIC      BIC      AICc
## -8.1276  16.2552   30.2552  29.8766  142.2552
##
## Variance Components:
##
##           estim  sqrt  nlvls  fixed  factor
## sigma^2    0.0150 0.1223     5     no  Effect
##
## Test for Residual Heterogeneity:
## QE(df = 7) = 154.1935, p-val < .0001
##
## Test of Moderators (coefficients 2:6):
## QM(df = 5) = 179.1027, p-val < .0001
##
## Model Results:
##
##           estimate      se    zval    pval    ci.lb    ci.ub
## intrcpt          0.6087 0.0622   9.7918 <.0001   0.4869   0.7305 ***
## pre3m           -0.1438 0.0424  -3.3913 0.0007  -0.2269  -0.0607 ***
## strangMC        -0.0037 0.0026  -1.3969 0.1625  -0.0088   0.0015
## post3m           0.0189 0.0208   0.9077 0.3640  -0.0219   0.0597
## pre3m:strangMC  -0.0038 0.0021  -1.8709 0.0614  -0.0079   0.0002 .
## strangMC:post3m  0.0045 0.0021   2.1050 0.0353   0.0003   0.0087 *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```