PCLE Replication

Emorie Beck

9/24/2017

Workspace

Packages

```r
library(survey)
library(mi)
library(psych)
library(MatchIt)
library(lavaan)
library(semTools)
library(rstan)
library(lme4)
library(brms)
library(rstanarm)
library(tidybayes)
library(MuMIn)
library(parallel)
library(gridExtra)
library(knitr)
library(kableExtra)
library(stargazer)
library(plyr)
library(stringr)
library(haven)
library(tidyverse)
library(ggridges)

data_path <- "~/Box/network/other projects/PCLE Replication"
model_path <- "~/Box/Models/PCLE Replication"
```

Data

Load Raw Data

```r
meta <- readxl::read_xlsx(sprintf("%s/data/Codebook.xlsx", data_path)) %>%
  mutate(Item = stringr::str_to_lower(Item))

all.old.cols <- (meta %>% filter(class == "proc" & Year == 0))$Item
all.new.cols <- (meta %>% filter(class == "proc" & Year == "0"))$new_name

# create short function to read in separate files for each wave
read_fun <- function(file, year){
  print(year)
  old.names <- (meta %>% filter(Year == year & class %in% c("group", "predictor", "proc")))$Item
  new.names <- (meta %>% filter(Year == year & class %in% c("group", "predictor", "proc")))$new_name
  z <- haven::read_sav(sprintf("%s/data/sav_files/%sp.sav", data_path, file)) %>%
    left_join(haven::read_sav(sprintf("%s/data/sav_files/%spkal.sav", data_path, file))) %>%
    left_join(haven::read_sav(sprintf("%s/data/sav_files/%sh.sav", data_path, file))) %>%
    select(one_of(all.old.cols), one_of(old.names)) %>%
    setNames(c(all.new.cols, new.names)) %>%
    mutate_all(funs(mapvalues(., seq(-1,-7,-1), c(NA, 0, rep(NA,5)), warn_missing = F))) %>%
    group_by(PROC_SID) %>%
    mutate(LE_ParDied = max(LE_MomDied,LE_DadDied, na.rm = T),
           LE_ParDied = ifelse(is.nan(LE_ParDied) == T, NA, LE_ParDied)) %>%
    gather(key = new_name, value = value, -PROC_SID, -PROC_household, -Dem_DOB, -Dem_Sex) %>%
    left_join(meta %>% filter(Year == year) %>% select(new_name, rev_code)) %>%
    mutate(value = ifelse(rev_code == 0 | is.na(rev_code), value,
                   reverse.code(keys=-1, items=value, mini=1, maxi=8))) %>%
    select(-rev_code)
}

dat <- tibble(
  Year = as.character(seq(2005, 2015,1)),
  file = c(letters[22:26], paste("b", letters[1:6], sep = ""))) %>%
  mutate(data = map2(file, Year, read_fun)) %>%
  unnest(data) %>%
  group_by(PROC_SID) %>%
  mutate(
    Dem_DOB = max(Dem_DOB, na.rm = T),
    Dem_DOB = ifelse(is.infinite(Dem_DOB) == T, NA, Dem_DOB),
    Dem_Sex = max(Dem_Sex, na.rm = T),
    Dem_Sex = ifelse(is.infinite(Dem_Sex) == T, NA, Dem_Sex)
  )
```

```
load(sprintf("%s/results/data.RData", data_path))
```

Clean BFI Data

```
bfi_dat <- dat %>% ungroup() %>%
  separate(new_name, c("type", "Item"), sep = "_") %>%
  filter(type == "BF") %>%
  mutate(wave = as.numeric(mapvalues(Year, seq(2005,2013,4), 1:3))) %>%
  group_by(PROC_SID) %>%
  mutate(fy = min(Year, na.rm = T),
         wave = ifelse(fy != 2005, wave-min(wave) + 1, wave)) %>%
  # mutate(wave = seq(1, n(), 1)) %>%
  # find people who didn't do any Big 5 responses
  group_by(PROC_SID, Item) %>%
  mutate(na = sum(!is.na(value))) %>%
  ungroup() %>%
  # recode gender & center age at first BFI wave (2005)
  mutate(sex12 = mapvalues(Dem_Sex, c(1,2), c(1,0), warn_missing = F),
         sex.c = as.numeric(scale(sex12, center = T, scale = F)),
         age = as.numeric(fy)-Dem_DOB,
         age.c = age - mean(age, na.rm = T),
         age.c2 = age.c^2, #agec2 = agec^2,
         age.c3 = age.c^3) #agec3 = agec^3,


bfi_wide <- bfi_dat %>%
  filter(na > 1) %>%
  separate(Item, c("Trait", "Item"), 1) %>%
  unite(Item, wave, Item, sep = "_") %>%
  mutate(Item = sprintf("T%s", Item)) %>%
  select(PROC_SID, Trait, Item, value, sex12:age.c3) %>%
  spread(key = Item, value = value)

bfi_match <- bfi_dat %>%
  separate(Item, c("Trait", "Item"), 1) %>%
  # filter(Year == 2005) %>%
  group_by(PROC_SID, Trait, wave) %>%
  summarize(mean = mean(value)) %>%
  unite(tmp, Trait, wave, sep = "_") %>%
  spread(key = tmp, value = mean)

bfi_match <- bfi_dat %>%
  filter(na > 1) %>%
  unite(Item, Item, wave, sep = "_") %>%
```

Table 1: Cronbach's Alpha Scale Reliabilities for the BFI-S Subscales

Trait	Wave 1	Wave 2	Wave 3
A	0.51	0.50	0.49
C	0.62	0.59	0.57
E	0.65	0.65	0.66
N	0.61	0.62	0.62
O	0.62	0.62	0.62

```
  select(PROC_SID, Item, value) %>%
  spread(key = Item, value = value)
```

BFI Scale Reliability

```
alpha_fun <- function(df){
  df <- df %>% select(-PROC_SID)
  psych::alpha(df)$total$raw_alpha
}

bfi_wide %>% select(PROC_SID, Trait, T1_1:T3_3) %>%
  gather(key = item, value = value, T1_1:T3_3) %>%
  separate(item, c("wave", "item"), sep = "_") %>%
  spread(key = item, value = value) %>%
  group_by(wave, Trait) %>%
  nest() %>%
  mutate(alpha = map(data, alpha_fun)) %>%
  unnest(alpha, .drop = T) %>%
  spread(key = wave, value = alpha) %>%
  kable(., "latex", digits = 2, escape = F, booktabs = T,
        col.names = c("Trait", "Wave 1", "Wave 2", "Wave 3"),
        caption = "Cronbach's Alpha Scale Reliabilities for the BFI-S Subscales") %>%
  kable_styling(full_width = F)
```

Clean Life Event Data

```
event_fun <- function(df, event){
  print(event)
  print(unique(df$Year))
  z <- df %>%
    select(-type, -file) %>%
    spread(key = Year, value = value) #%>%
  #   mutate(
  #     `2006` = ifelse(`2005` != 1 & `2006` == 1, 1, 0),
  #     `2007` = ifelse(`2006` != 1 & `2007` == 1, 1, 0),
  #     `2008` = ifelse(`2007` != 1 & `2008` == 1, 1, 0),
  #     `2009` = ifelse(`2008` != 1 & `2009` == 1, 1, 0),
  #     `2010` = ifelse(`2009` != 1 & `2010` == 1, 1, 0),
  #     `2011` = ifelse(`2010` != 1 & `2011` == 1, 1, 0),
  #     `2012` = ifelse(`2011` != 1 & `2012` == 1, 1, 0),
  #     `2013` = ifelse(`2012` != 1 & `2013` == 1, 1, 0))
  # if(event != "LeftPar"){z <- z %>%
  #   mutate(`2014` = ifelse(`2013` != 1 & `2014` == 1, 1, 0),
  #          `2015` = ifelse(`2014` != 1 & `2015` == 1, 1, 0))
  # }else{z <- z %>%
  #   mutate(`2015` = ifelse(`2013` != 1 & `2015` == 1, 1, 0))}
  z <- z %>% mutate(
      Event12 = ifelse(`2005` == 0 & rowSums(cbind(`2006`,`2007`,`2008`,`2009`),na.rm = T) > 0, 1, 0),
      Event23 = ifelse(Event12 == 0 & rowSums(cbind(`2010`,`2011`,`2012`,`2013`),na.rm = T) > 0, 1, 0))
  if(event != "LeftPar"){
    z <- z %>% mutate(
      Event3p = ifelse(rowSums(cbind(`2014`, `2015`), na.rm = T) > 0, 1, 0)) #%>%
      # gather(key = le.group, value = le.value, `2006`:Event3p)
  } else{
    z <- z %>% mutate(Event3p = ifelse(`2015` >= 1, 1, 0)) #%>%
      # gather(key = le.group, value = le.value, `2006`:Event3p)
  }
}

# missing moving out of parental home and how often a child was born

le_dat <- dat %>% ungroup() %>%
  separate(new_name, c("type", "Event"), sep = "_") %>%
  filter(type == "LE" & #PROC_SID %in% unique(bfi_wide$PROC_SID) &
           Event %in% c("Married", "Divorce", "MoveIn","SepPart", "PartDied",
                        "ChldMvOut", "ChldBrth", "MomDied", "DadDied", "ParDied",
                        "Unemploy", "Retire", "FrstJob", "LeftPar")) %>%
  mutate(value = ifelse(Event == "Retire" | Event == "Unemploy", mapvalues(value, c(2,1), c(0,1)),
                 ifelse(Event == "FrstJob", mapvalues(value, seq(1,6,1), c(1,rep(0,5))),value))) %>%
  group_by(Event) %>%
  nest() %>%
  mutate(event.dat = map2(data, Event, event_fun)) %>%
  unnest(event.dat, .drop = T)

le_dat <- le_dat %>% select(Event:Dem_Sex, Event12, Event23) %>%
  mutate(le.group = ifelse(Event12 == 1 | Event23 == 1, 1, 0),
         le.group = ifelse(is.na(Event12) == T & is.na(Event23) == T, NA_real_, le.group))
```

Clean Matching Data

```
all.old.cols <- (meta %>% filter(class %in% c("proc") & Year == "0"))$Item
all.new.cols <- (meta %>% filter(class %in% c("proc") & Year == "0"))$new_name

# create short function to read in separate files for each wave
read_fun <- function(file, year){
  print(year)
  old.names <- (meta %>% filter(Year == year & class %in% c("match", "proc") & Include == "Yes"))$Item
  new.names <- (meta %>% filter(Year == year & class %in% c("match", "proc") & Include == "Yes"))$new_name
  z <- haven::read_sav(sprintf("%s/data/sav_files/%sp.sav", data_path, file)) %>%
    left_join(haven::read_sav(sprintf("%s/data/sav_files/%spkal.sav", data_path, file))) %>%
    left_join(haven::read_sav(sprintf("%s/data/sav_files/%sh.sav", data_path, file))) %>%
    left_join(haven::read_sav(sprintf("%s/data/sav_files/%spequiv.sav", data_path, file))) %>%
    left_join(haven::read_sav(sprintf("%s/data/sav_files/%spgen.sav", data_path, file))) %>%
    left_join(haven::read_sav(sprintf("%s/data/sav_files/%shbrutto.sav", data_path, file))) %>%
    # left_join(haven::read_sav(sprintf("%s/data/sav_files/%shost.sav", data_path, file))) %>%
    # left_join(haven::read_sav(sprintf("%s/data/sav_files/%spost.sav", data_path, file))) %>%
    select(one_of(all.old.cols), one_of(old.names)) %>%
    setNames(c(all.new.cols, new.names)) %>%
    mutate_all(funs(mapvalues(., seq(-1,-7,-1), c(NA, 0, rep(NA,5)), warn_missing = F))) %>%
    group_by(PROC_SID) %>%
    gather(key = new_name, value = value, -PROC_SID, -PROC_household, -Dem_DOB, -Dem_Sex) %>%
    left_join(meta %>% filter(Year == year) %>% select(new_name, rev_code, mini, maxi, rule)) %>%
    mutate(value = ifelse(rev_code == 1, reverse.code(keys=-1, items=value, mini=mini, maxi=maxi), value)) %>%
    select(-rev_code, -mini, -maxi)
}

health.old.cols <- (meta %>% filter(dataset == "health"))$Item
health.new.cols <- (meta %>% filter(dataset == "health"))$new_name

health <- tbl_df(haven::read_sav(sprintf("%s/data/sav_files/health.sav", data_path))) %>%
  filter(valid == 1 & syear < 2006) %>%
  select(one_of(all.old.cols), one_of(health.old.cols)) %>%
  setNames(c(all.new.cols, health.new.cols)) %>%
  mutate_all(funs(mapvalues(., seq(-1,-7,-1), c(NA, 0, rep(NA,5)), warn_missing = F))) %>%
```

```r
  gather(key = new_name, value = value, -PROC_SID, -PROC_household, -Year) %>%
  left_join(meta %>% filter(dataset == "health") %>% select(new_name, rule))

match.dat <- tibble(
  Year = seq(1984, 2004,1),
  file = c(letters[1:21])) %>%
  mutate(data = map2(file, Year, read_fun)) %>%
  unnest(data) %>%
  full_join(health) %>%
  mutate(Dem_DOB = ifelse(Dem_DOB < 1850, NA_real_, Dem_DOB),
         Dem_DOB = recode(Dem_DOB, `0` = NA_real_),
         Dem_Sex = recode(Dem_Sex, `0` = NA_real_)) %>%
  group_by(PROC_SID) %>%
  mutate(
         Dem_DOB = max(Dem_DOB, na.rm = T),
         Dem_Sex = max(Dem_Sex, na.rm = T)) #%>%
```

```r
match.dat <- match.dat %>%
  mutate(value = ifelse(new_name == "Psych_OthWorr" & value >= 1, 1, value),
         value = ifelse(new_name %in% c("Bkgr_DadEdu", "Bkgr_MomEdu"),
                        mapvalues(value, c(0,6,7,1,2,9,3,4,5), rep(0:2, each = 3), warn_missing = F), value),
         value = ifelse(new_name == "Bkgr_Edu" & value > 0, 1, value),
         value = ifelse(new_name %in% c("Fnc_HouseAssist", "HH_Internet"),
                        recode(value, `2` = 0), value),
         value = ifelse(new_name == "HH_CndHouse", mapvalues(value, c(2,3,4,1), c(0,0,0,1), warn_missing = F), value),
         value = ifelse(new_name == "Bkgr_MarStat", mapvalues(value, c(2,6,7,1,3,4,5),
                        c(0,0,0,1,2,2,2), warn_missing = F), value),
         value = ifelse(new_name == "HH_ClnHlp", mapvalues(value, c(3,1,2), c(0,1,1), warn_missing = F), value),
         value = ifelse(new_name == "", mapvalues(value, c(98, 11,12,13,15,16, 21, 22, 31,32,33,34,
                 35,36,37,38, 14,41,44,42,43), c(0, rep(1, 15), rep(2,3), rep(3,2)), warn_missing = F), value),
         Dem_DOB = ifelse(is.infinite(Dem_DOB) == T | is.nan(Dem_DOB) == T, NA, Dem_DOB),
         Dem_Sex = ifelse(is.infinite(Dem_Sex) == T | is.nan(Dem_Sex) == T, NA, Dem_Sex))
```

```r
# create a small function for calculating the mode
Mode <- function(x) {
  ux <- unique(x)
  ux <- ux[!is.na(ux)]
  ux[which.max(tabulate(match(x, ux)))]
}
sum_fun <- function(df, Rule){
  fun_call <- function(x, rule){
    switch(rule,
           average = mean(x, na.rm = T),
           mode = Mode(x)[1],
           sum = sum(x, na.rm = T),
           select = unique(x)[1],
           max = max(x, na.rm = T))
  }
  df %>%
    group_by(PROC_SID, new_name, Dem_DOB, Dem_Sex, PROC_household) %>%
    summarize(value = fun_call(value, Rule)) %>%
    mutate(value = ifelse(is.nan(value) == T, NA,
                 ifelse(is.infinite(value) == T, NA, value))) %>%
    ungroup()
}

match.dat.wide <- match.dat %>%
  group_by(rule) %>%
  nest() %>%
  mutate(data = map2(data, rule, possibly(sum_fun, NA_real_))) %>%
  unnest(data, .drop = T) %>%
  select(-rule) %>% # get rid of the rule variables
  spread(key = new_name, value = value)  %>% # change to wide format
  left_join(bfi_match)
```

Match Subjects Across Data Categories

```
bfi_subs   <- unique(bfi_wide$PROC_SID)
match_subs <- unique(match.dat.wide$PROC_SID)
le_subs    <- unique(le_dat$PROC_SID)

subs <- bfi_subs[bfi_subs %in% match_subs]
subs <- subs[subs %in% le_subs]

match.dat.wide <- match.dat.wide %>% filter(PROC_SID %in% subs)
bfi_wide       <- bfi_wide       %>% filter(PROC_SID %in% subs)
bfi_match      <- bfi_match      %>% filter(PROC_SID %in% subs)
le_dat         <- le_dat         %>% filter(PROC_SID %in% subs)

save(match.dat, match.dat.wide, bfi_wide, le_dat, bfi_match,
     file = sprintf("%s/results/data.RData", data_path))
```

Multiple Imputation

Missing Data Frame

First, we check the missingness patterns of the match data by converting it to a missing data frame class object using the `missing_data.frame()` function in the `mi` package in `R`. We then use the `image()` function to graphically depict the missingness patterns.

```
# MI doesn't like tibbles, so we need to unclass and reclass the data
match.dat.wide <- data.frame(unclass(match.dat.wide)) # mi doesn't like tibbles

mdf <- missing_data.frame(match.dat.wide)

pdf(sprintf("%s/plots/%s.pdf", data_path, "mdf"), width = 9, height = 6.5)
  image(mdf)
dev.off()
```

Now, we want to ensure that `missing_data.frame()` has correctly detected the type of variable (nominal, ordinal, etc.), so that missing data will be imputed using the correct link function.

```
des <- describe(match.dat.wide,fast = T)
mdf <- change(data = mdf, y = rownames(des)[des$range >= 3],
                   what = "type", to = "continuous")
```

Multiple Imputation Procedure

Now, we use the `mi()` function in the `mi` package in to complete the multiple imputation procedure. We create 10 imputed data sets (by setting n.chains to 10) and use 20 iterations for each. We have a lot of variables and a lot of observations, so we use parallel processing to run the procedure.

```
mi.res <- mi(mdf, n.chains = 10, n.iter = 20, parallel = T)
```

Now we compare the missingness patterns before and after imputation and see that we no longer have missing data.

```
pdf(sprintf("%s/plots/%s.pdf", data_path, "mi"), width = 9, height = 6.5)
  image(mi.res)
dev.off()
```

And grab the imputed data sets from the MI procedure using the `complete()` function from the `mi` package, which saves them in a list.

```
complete_fun <- function(mi){
  clean_fun <- function(df){df %>% select(-contains("missing_"))}
  tibble(chain = 1:10,
         imp.data = mi::complete(mi)) %>%
    mutate(imp.data = map(imp.data, clean_fun)) %>%
    unnest(imp.data)
}
```

```r
imp.data <- complete_fun(mi.res)

bfi.imp <- unique(imp.data %>%
  select(chain, PROC_SID, A1_1:O3_3) %>%
  gather(key = item, value = value, A1_1:O3_3) %>%
  separate(item, c("item", "wave"), sep = "_") %>%
  separate(item, c("Trait", "item"),1) %>%
  unite(item, wave, item, sep = "_") %>%
  mutate(item = sprintf("T%s", item)) %>%
  spread(key = item, value = value) %>%
  left_join(bfi_wide %>% select(PROC_SID, sex12:age.c3)))

psw.imp.data <- imp.data %>%
  gather(key = item, value = value, A1_1:O3_3) %>%
  separate(item, c("item", "wave"), sep = "_") %>%
  separate(item, c("Trait", "item"), 1) %>%
  filter(wave == 1) %>%
  group_by(chain, Trait, PROC_SID, wave) %>%
  summarize(M = mean(value, na.rm = T)) %>%
  ungroup() %>%
  unite(Trait, Trait, wave, sep = "_") %>%
  spread(key = Trait, value = M) %>%
  full_join(imp.data %>% select(chain:Val_Chrch))

save(mdf, mi.res, file = paste(data_path, "results/mi_dat.RData", sep = "/"))
save(imp.data, bfi.imp, psw.imp.data, file = paste(data_path, "results/mi_dat_small.RData", sep = "/"))
rm("mi.res")
```

We only need the columns that aren't meant to define missingness patterns, so we write a simple function to do that for each element in the list of imputed data sets. Then we put them into a dataframe in which each of the imputed data sets is saved in a cell of the data frame, which will make it much easier to use for propensity score weighting and growth curve modeling.

```r
load(paste(data_path, "results/mi_dat_small.RData", sep = "/"))
nested.psw <- crossing(
  Event = unique(le_dat$Event),
  match_set = c("selection", "socialization"),
  chain = 1:10
)
```

Propensity Score Matching

Then, we perform propensity score weighting using our imputed datasets using the `twang` package. We then add the weights to our matching data frame, along with our predictor variables. To test the effectiveness of the propensity score weighting procedure, we examine the average standardized effect size in the balance tables. minimal effect sizes are candidates for beng dropped from the propensity score weighting, and large effect sizes mean our weighting procedure wasn't effective. We can also examine these using balance plots.

```r
# this function actually runs the propensity score weighting procedure
psw_fun <- function(event, Chain, match_set){
  print(sprintf("%s Chain %s", event, Chain))
  Ratio <- ifelse(event %in% c("ChldMvOut", "ParDied", "Retire", "MoveIn"), 8, 4)
  Caliper <- ifelse(match_set == "socialization" &
      event %in% c("SepPart", "Unemploy", "Retire", "FrstJob"), .05,
      ifelse(match_set == "socialization" &
      event %in% c("MoveIn"), .01, .25))
  df <- psw.imp.data %>% filter(chain == Chain) %>%
    full_join(le_dat %>% filter(Event == event) %>%
                select(Event, PROC_SID, le.group)) %>%
    select(-chain, -Event)
  if(event == "Retire"){df <- df %>% filter(2005 - Dem_DOB > 40)}
  if(event == "FrstJob"){df <- df %>% filter(2005 - Dem_DOB < 40)}
  df <- df[complete.cases(df),]
  df <- data.frame(unclass(df))
  if(match_set == "socialization"){to.match <- colnames(df)[-which(colnames(df) %in% c("PROC_SID","le.group",
                    paste(rep(c("A", "C", "E", "N", "O"), each = 2), rep(2:3, times = 5), sep = "_")))]}
  else {to.match <- colnames(df)[-which(colnames(df) %in% c("PROC_SID","le.group",
                   paste(rep(c("A", "C", "E", "N", "O"), each = 3), rep(1:3, times = 5), sep = "_")))]}
  match.formula <- as.formula(paste("le.group ~ ", paste(to.match, collapse=" + "), sep = " "))
  y <- matchit(match.formula, data = df, method = "nearest", ratio = Ratio, caliper = Caliper)
}

# changing the data fed into psw into a data frame because it won't work with tibbles
psw_df <- function(psw){psw$data <- data.frame(psw$data); psw}

psw_df <- function(psw){
  data.frame(match.data(psw))
}
# this function creates the balance table of the psw weights and filters
# the results into variables the matching procedure did not fix and
# those that it did
unbalanced_fun <- function(x){
  y <- summary(x, standardize = T)
  raw <- y$sum.all %>%
    mutate(var = rownames(.)) %>%
    select(var, `Means Treated`, `Means Control`, `Std. Mean Diff.`)
  smalldiff.var <- raw %>% filter(abs(`Std. Mean Diff.`) <= .05)
  matched <- y$sum.matched %>%
    mutate(var = rownames(.)) %>%
    select(var, `Means Treated`, `Means Control`, `Std. Mean Diff.`)
  unbalanced.var <- matched %>% filter(abs(`Std. Mean Diff.`) >= .2)
  return(list(raw = raw, matched = matched,
              unbalanced = unbalanced.var,smalldiff = smalldiff.var))
}
```

```
nested.psw <- nested.psw %>%
  filter(chain == 1) %>%
  mutate(psw = pmap(list(Event, chain, match_set), psw_fun),
         psw.df = map(psw, possibly(psw_df, NA_real_)),
         bal.df = map(psw, unbalanced_fun),
         raw = map(bal.df, ~.$raw),
         matched = map(bal.df, ~.$matched),
         unbal.tab = map(bal.df, possibly(~.$unbalanced, NA_real_)),
         smalldiff.tab = map(bal.df, possibly(~.$smalldiff, NA_real_)))

nested.psw <- nested.psw %>% filter(!(match_set == "socialization" & chain == 1 &
      Event %in% c("FrstJob"))) %>% bind_rows(nested.psw2)

save(nested.psw, file = paste(data_path, "results/psw.RData", sep = "/"))
nested.psw <- nested.psw %>% select(-psw)
save(nested.psw, file = paste(data_path, "results/psw_small.RData", sep = "/"))
```

Balance Plots

In these plots, substantial reductions in effect sizes are observed for most variables (blue lines), with only one variable showing an increase in effect size (red lines), but only a seemingly trivial increase. Closed red circles indicate a statistically significant difference, many of which occur before weighting, none after.

```
plot_fun <- function(df, event, set){
  plot <- df %>%
    mutate(type = factor(type, level = c("raw", "matched"))) %>%
    ggplot(aes(x = type, y = `Std. Mean Diff.`)) +
    scale_y_continuous(limits = c(-5,5), breaks = seq(-5,5,2)) +
    geom_point() +
    geom_line(aes(group = var), size = .25, alpha = .8) +
    labs(x = NULL, y = "Standardized Mean Difference",
         title = sprintf("%s", event)) +
    facet_wrap(~chain, ncol = 2) +
    theme_classic()
  ggsave(sprintf("%s/plots/psw_bal_%s.png", data_path, event), width = 8, height = 10)
}
```

```
print_plot_fun <- function(p, Chain){
  p$main <- sprintf("Imputed dataset %s", gsub("chain.", "", Chain))
  p
}

nested.psw %>%
  unnest(raw) %>% mutate(type = "raw") %>%
  full_join(nested.psw %>% unnest(matched) %>% mutate(type = "matched"))%>%
  group_by(Event) %>%
  nest() %>%
  mutate(plot = map2(data, Event, plot_fun))

par(mfrow = c(2,5))
nested.psw <- nested.psw %>%
  mutate(plots = map(psw, possibly(~plot(., plots = "es"), NA_real_)),
         plots = map2(plots, chain, possibly(print_plot_fun, NA_real_)))
```

Balance Tables

Once propensity scores are estimated, bal.table() produces a table that shows how well the resulting weights succeed in manipulating the groups so that they match on pre-adolescent matching characteristics.

```
load(paste(data_path, "results/psw_small.RData", sep = "/"))
load(paste(data_path, "results/mi_dat_small.RData", sep = "/"))
# this table shows variables that are not matched after weighting
# unbal.tab <- nested.psw %>%
#   filter(match_set == "socialization") %>%
#   unnest(unbal.tab, .drop = T) %>%
#   group_by(Event, var) %>%
#   #summarize_at(vars(`Means Treated`:`Std. Mean Diff.`), funs(mean(., na.rm = T)))
#   summarize(mean = mean(`Std. Mean Diff.`, na.rm = T)) %>%
#   spread(key = Event, value = mean)
# kable(unbal.tab, "latex", longtable = T, booktabs = T, digits = 2,
#       caption = "Unbalanced Variables after Propensity Score Weighting") %>%
#   kable_styling(latex_options = c("striped","repeat_header"),full_width = F)

# this table shows variables that were already matched prior to weighting
smalldiff.tab <- nested.psw %>%
  unnest(smalldiff.tab, .drop = T) %>%
  group_by(Event, var) %>%
  #summarize_at(vars(`Means Treated`:`Std. Mean Diff.`), funs(mean(., na.rm = T)))
  summarize(mean = mean(`Std. Mean Diff.`, na.rm = T)) %>%
  spread(key = Event, value = mean)
kable(smalldiff.tab, "latex", longtable = T, booktabs = T, digits = 2,
      caption = "Balanced Variables after Propensity Score Weighting") %>%
  kable_styling(latex_options = c("striped","repeat_header"),full_width = F)
```

Table 2: Balanced Variables after Propensity Score Weighting

var	ChldBrth	ChldMvOut	DadDied	Divorce	FrstJob	LeftPar	Married	MomDied	MoveIn	ParDied	Pa
A_1	NA	0.01	NA	NA	0.01	NA	NA	0.02	-0.04	-0.04	
Act_Volunteer	NA	NA	NA	0.04	-0.04	0.02	NA	NA	NA	NA	
Bkgr_DadPres1	NA	NA	NA	NA	NA	-0.02	NA	0.02	NA	NA	
Bkgr_DisabStat1	NA	NA	NA	NA	-0.05	NA	NA	-0.03	NA	NA	
Bkgr_Edu1	NA	NA	NA	NA	NA	NA	NA	NA	0.04	NA	
Bkgr_MarStat.L	NA	NA	-0.04	NA	NA	NA	NA	NA	NA	NA	
Bkgr_MarStat.Q	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
Bkgr_PGovIncome	0.00	NA	NA	NA	NA	NA	NA	NA	NA	NA	
Bkgr_UrbOrRur2	-0.03	NA	NA	NA	-0.01	0.02	0.00	-0.01	0.04	-0.03	
C_1	NA	NA	0.00	0.00	NA	NA	NA	NA	NA	0.04	
Dem_DOB	NA	0.00	NA	NA	NA	NA	NA	NA	NA	NA	
Dem_Sex1	-0.02	NA	-0.01	NA	-0.01	0.04	0.03	0.04	-0.02	0.01	

Table 2: Balanced Variables after Propensity Score Weighting *(continued)*

var	ChldBrth	ChldMvOut	DadDied	Divorce	FrstJob	LeftPar	Married	MomDied	MoveIn	ParDied	Pa
Dem_Sex2	0.02	NA	0.01	NA	0.01	-0.04	-0.03	-0.04	0.02	-0.01	
E_1	NA	0.05	-0.01	NA	-0.01	NA	NA	0.00	NA	0.00	
Fnc_HouseAssist1	NA	NA	0.00	NA	NA	NA	NA	NA	NA	-0.03	
Fnc_StudGrnt1	NA	NA	-0.03	0.02	NA	NA	NA	NA	NA	NA	
Fnc_UnempBen	-0.04	0.01	NA	NA	NA	NA	NA	0.03	NA	-0.01	
HH_BrothPres1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
HH_ClnHlp1	NA	0.02	0.01	NA	NA	-0.05	NA	NA	NA	0.04	
HH_CndHouse1	NA	0.02	NA	NA	NA	NA	NA	-0.04	NA	NA	
HH_ColTV1	NA	0.01	0.04	-0.04	NA	-0.04	-0.02	NA	NA	NA	
HH_NumPer	NA	NA	NA	NA	NA	NA	0.01	NA	NA	NA	
HH_NumPer15to18	0.03	NA	0.05	0.00	NA	NA	NA	0.03	NA	0.04	
HH_NumPerBel14	-0.04	NA	NA	NA	NA	NA	NA	NA	NA	NA	
HH_SisPres1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	
Hlth_BMI	NA	NA	NA	NA	NA	NA	NA	NA	NA	0.01	
Hlth_BodPain	NA	0.02	NA	NA	NA	NA	NA	-0.05	NA	0.00	
Hlth_EmoRole	NA	-0.01	-0.04	NA	0.02	NA	NA	-0.05	0.04	-0.05	
Hlth_GenHlth	NA	0.01	NA	NA	NA	NA	NA	-0.03	NA	NA	
Hlth_HeightCM	NA	-0.03	NA	0.02	-0.02	NA	NA	0.01	NA	NA	
Hlth_HlthInsr	NA	NA	-0.03	0.01	NA	NA	NA	-0.04	NA	-0.03	
Hlth_MntlHlth	NA	NA	NA	NA	NA	-0.05	NA	NA	NA	NA	
Hlth_NumDrVisits	NA	NA	NA	0.04	NA	NA	NA	-0.03	NA	NA	
Hlth_PhysFunc	NA	NA	NA	NA	NA	NA	NA	-0.04	NA	NA	
Hlth_PhysHlth	NA	NA	NA	NA	NA	NA	NA	-0.02	NA	NA	
Hlth_PhysProb	NA	0.01	NA	NA	NA	NA	NA	NA	NA	-0.02	
Hlth_PhysRole	NA	0.03	NA	NA	NA	NA	NA	NA	NA	0.00	
Hlth_SocFunc	NA	0.00	NA	NA	NA	NA	0.04	NA	0.00	NA	
Hlth_Vitality	NA	-0.01	0.03	-0.01	NA	NA	NA	-0.02	NA	0.01	
Hlth_WeightKG	NA	NA	-0.02	NA	NA	NA	NA	NA	NA	0.05	
N_1	NA	-0.01	-0.02	NA	0.01	NA	0.04	-0.04	0.03	-0.03	
O_1	-0.01	0.03	0.03	NA	NA	NA	NA	0.00	NA	0.02	
PROC_household	NA	NA	0.03	-0.05	NA	NA	NA	NA	0.00	0.05	
Psych_LifeSat	NA	-0.05	-0.02	NA	NA	NA	-0.01	NA	-0.01	-0.04	
Psych_OthWorr1	NA	NA	0.05	0.02	NA	-0.05	NA	0.02	0.02	0.04	
Psych_SatFam	NA	0.04	0.04	0.04	NA	NA	NA	-0.02	0.02	0.00	
Psych_SatHealth	NA	0.01	NA	NA	NA	NA	NA	-0.04	NA	0.03	
Psych_SatIncome	NA	NA	-0.03	NA	NA	NA	NA	0.04	NA	0.01	
Psych_WorrCrm	NA	0.03	-0.04	-0.05	NA	NA	NA	NA	NA	0.01	
Rel_RelDad	NA	NA	NA	NA	NA	0.00	NA	0.02	NA	NA	
Rel_RelMom	NA	NA	NA	NA	NA	NA	NA	NA	0.04	NA	
Soc_SocGath	NA	NA	0.04	0.02	NA	NA	NA	NA	NA	NA	
Soc_VisFam	NA	NA	0.00	NA	NA	NA	0.03	0.05	NA	0.02	
Soc_VisNghbr	NA	NA	0.03	-0.04	NA	NA	NA	NA	NA	NA	
Val_Chrch	NA	NA	NA	NA	-0.02	NA	NA	0.03	NA	-0.03	

Plots

```
cohens_d <- function(x, y) {
    lx <- length(x)- 1; ly <- length(y)- 1
    md  <- mean(x, na.rm = T) - mean(y, na.rm = T)        ## mean difference (numerator)
    csd <- lx * var(x, na.rm = T) + ly * var(y, na.rm = T)
    csd <- csd/(lx + ly); csd <- sqrt(csd)                     ## common sd computation
    cd  <- md/csd                        ## cohen's d
    return(cd)
}

d_fun <- function(df, Var, chain){
  dat <- df %>% filter(chain == chain)
  d <- with(dat, cohens_d(value[le.group == 1], value[le.group == 0]))
}
levs <- c(colnames(psw.imp.data)[-c(1,2,10)], "Age")

diff <- psw.imp.data %>% filter(chain == 1) %>%
  mutate(match_set = "Unmatched") %>%
  full_join(le_dat %>% select(PROC_SID, Event, le.group))  %>%
      # filter(Event %in% c("FrstJob", "MoveIn", "PartDied"))) %>%
  filter(!is.na(Event) & !is.na(match_set)) %>%
  full_join(nested.psw %>%
      filter(match_set == "socialization" & chain == 1) %>%
      # filter(Event %in% c("FrstJob", "MoveIn", "PartDied")) %>%
      unnest(psw.df) %>%
      select(-weights)) %>%
  mutate(Dem_Sex = as.numeric(as.character(Dem_Sex)),
         Age = 2005-Dem_DOB) %>% select(-Dem_DOB) %>%
  rename(BFI_E.W1 = E_1, BFI_A.W1 = A_1, BFI_C.W1 = C_1, BFI_N.W1 = N_1, BFI_O.W1 = O_1) %>%
  mutate_if(is.factor, funs(as.numeric(as.character(.)))) %>%
  gather(key = var, value = value, -chain, -PROC_SID, -Event, -PROC_household, -match_set, -le.group) %>%
  separate(var, c("Category", "Item"), sep= "_", remove = F) %>%
  # mutate(var = factor(var, levels = rev(levs))) %>%
  group_by(Event, match_set, var, chain, Category, Item) %>%
  nest() %>%
  mutate(d = pmap(list(data, var, chain), d_fun))
save(diff, file = sprintf("%s/results/matching_diag.RData", data_path))
save(diff, file = sprintf("%s/results/mean_diff.RData", data_path))

events <- tibble(
  old = c("none", "Married", "MoveIn", "Divorce", "SepPart", "PartDied", "LeftPar",
          "ChldMvOut", "ChldBrth", "ParDied", "Unemploy", "Retire", "FrstJob"),
  new = c("Mean", "Marriage", "Moved in with Partner", "Divorce", "Separation from Partner",
          "Death of Partner/Spouse", "Leaving Parental Home", "Child Leaves Home",
          "Birth of Child", "Death of Parent",   "Unemployment", "Retirement", "First Job"),
  breaks = c("Mean", "Marriage", "Moved in\nwith Partner", "Divorce", "Separation\nfrom Partner",
          "Death of\nPartner/Spouse", "Leaving\nParental Home", "Child Leaves\nHome",
          "Birth of\nChild", "Death of\nParent",   "Unemployment", "Retirement", "First Job")
)

diff_fun <- function(event){
diff %>% unnest(d, .drop = T) %>% #filter(Category == "HH") %>%
  filter(var != "distance") %>%
  filter(Event == event) %>%
  group_by(Event, match_set, var) %>%
  summarize(d = mean(d)) %>% ungroup() %>%
  mutate(match_set = recode(match_set, `socialization` = "Matched"),
         Event = mapvalues(Event, events$old, events$breaks),
         Event = factor(Event, levels = events$breaks)) %>%
  ggplot(aes(x = var, y = d, shape = match_set)) +
  scale_shape_manual(values = c(19,1)) +
  scale_y_continuous(limits = c(-1.5, 1.5), breaks = seq(-1, 1, 1)) +
  geom_hline(aes(yintercept = 0), linetype = "dashed", size = .25) +
  geom_point(size = 1.5) +
  labs(y = "Cohen's d", x = NULL, shape = NULL, title = event) +
  # coord_flip() +
  facet_grid(.~Event) +
  theme_classic() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(face = "bold", size = rel(.7), angle = 45, hjust = 1),
        axis.text.y = element_text(face = "bold", size = rel(1.2)),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          strip.text = element_text(face = "bold"),
        plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5),
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)))
}
```

```
lapply(events$old[events$old != "none"], function(x){
  cat('####', x, '\n\n  ')
  diff_fun(x)
})
```

Married

#### MoveIn

#### Divorce

#### SepPart

#### PartDied

#### LeftPar

#### ChldMvOut

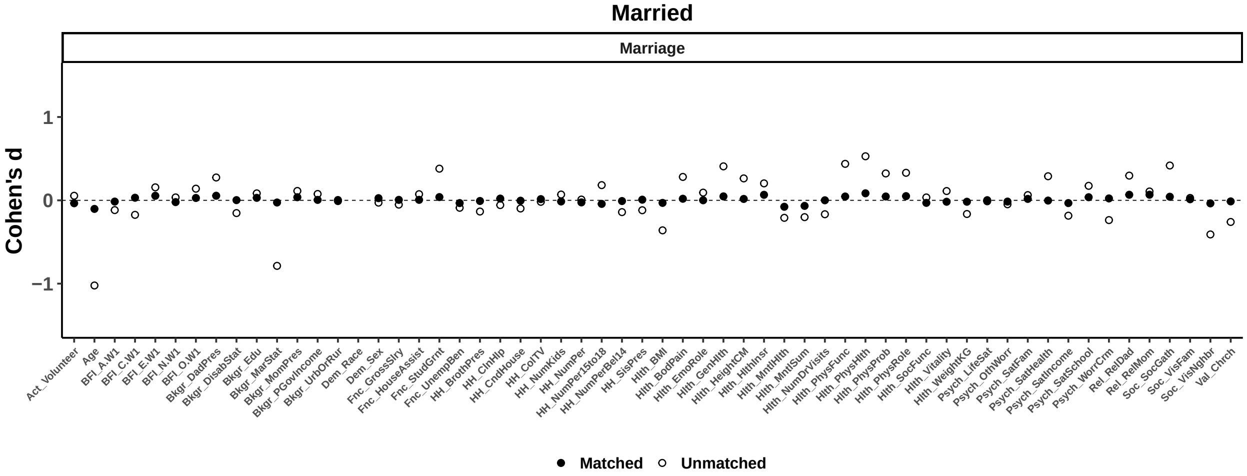
#### ChldBrth

#### ParDied

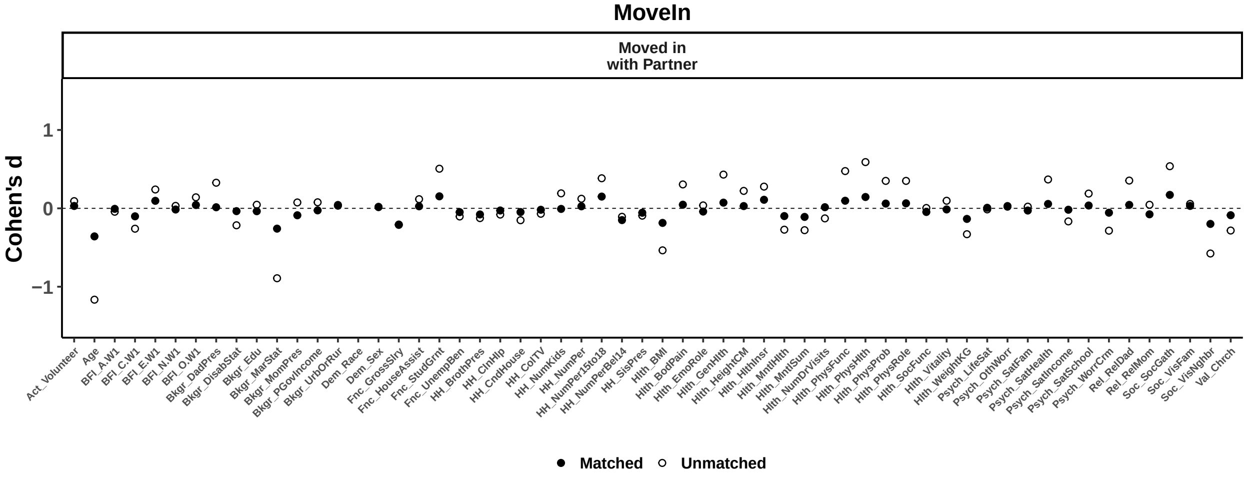
#### Unemploy

#### Retire

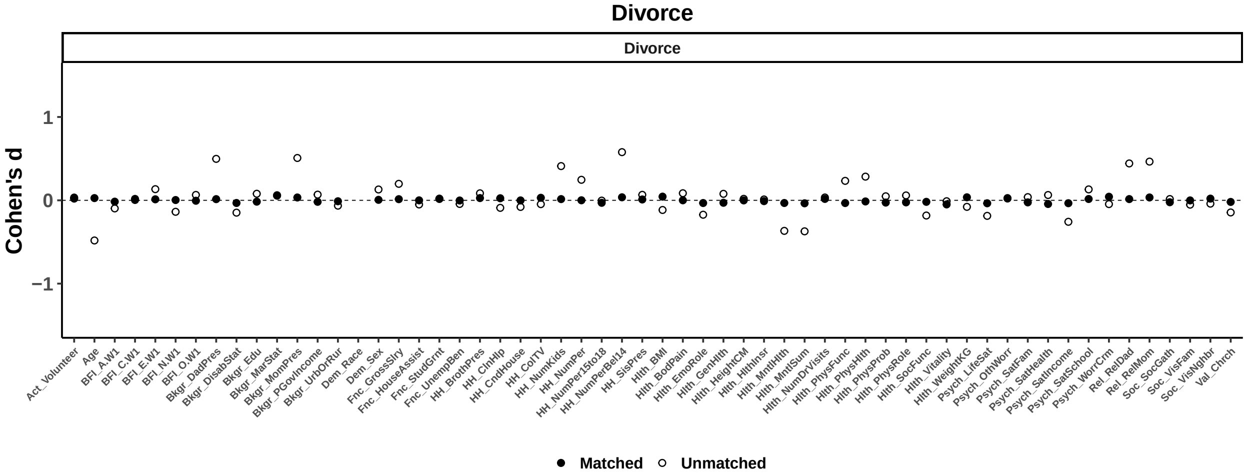
#### FrstJob



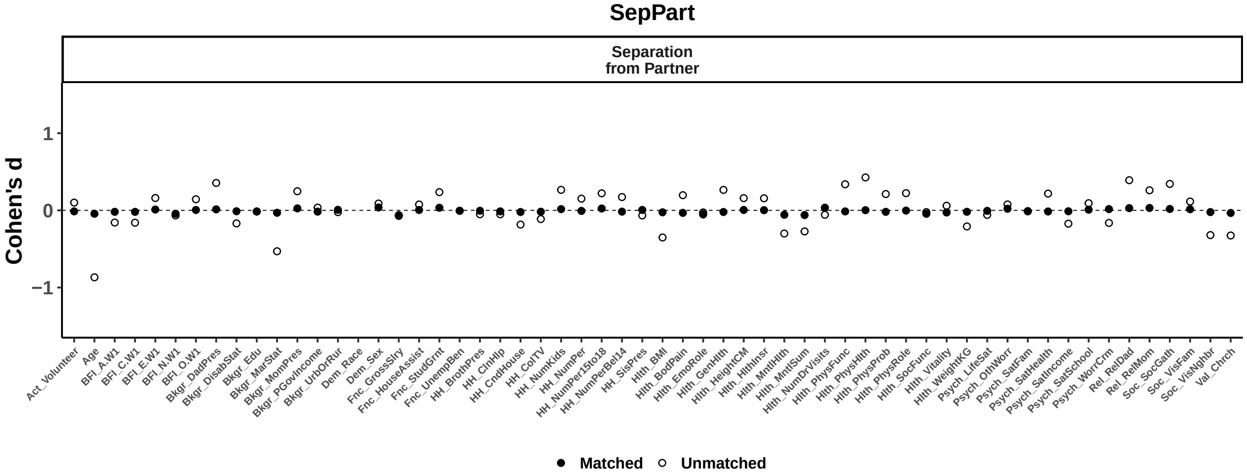
[[1]]



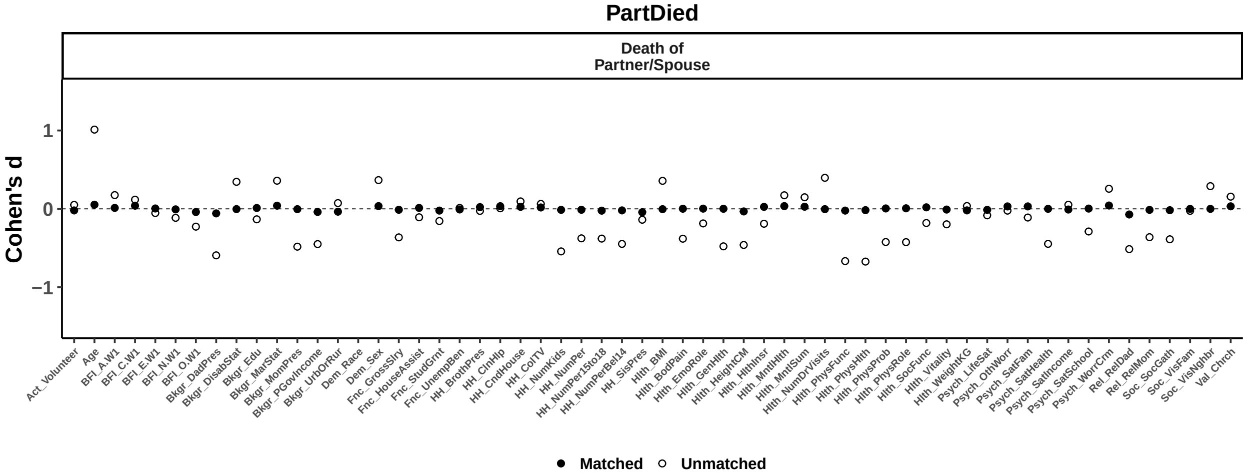
[[2]]


Divorce
Divorce
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeightKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

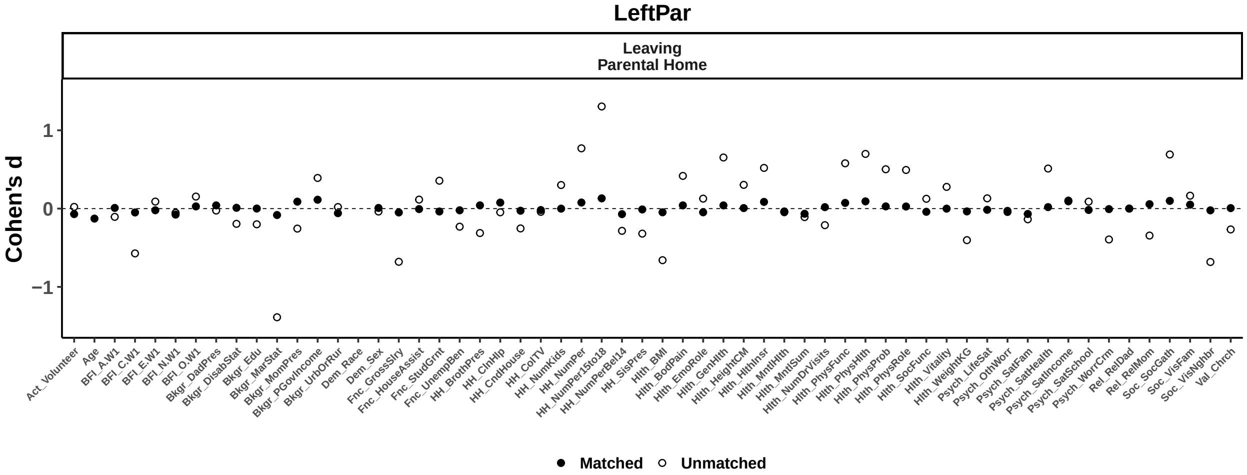
[[3]]


SepPart
Separation
from Partner
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeightKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

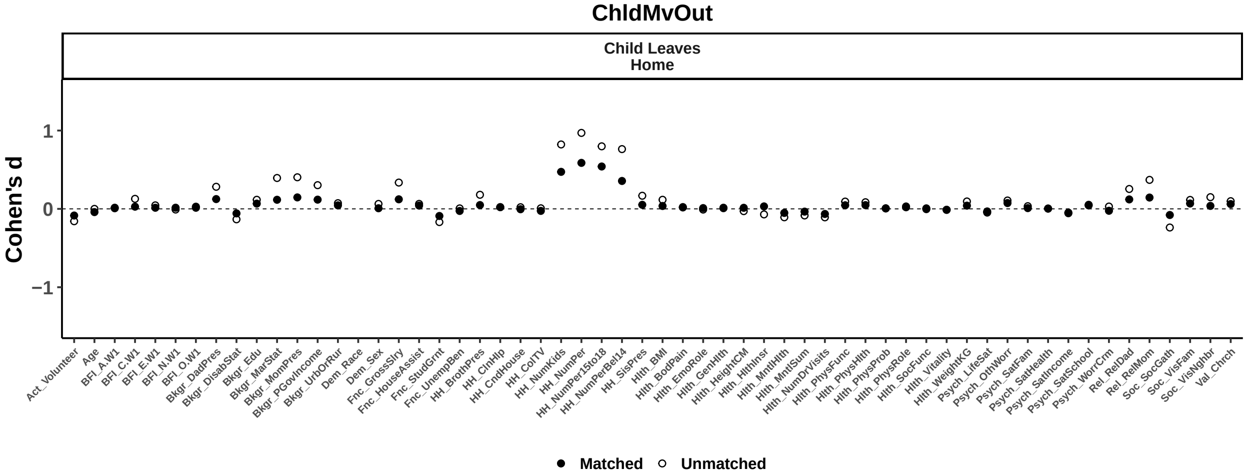
[[4]]


PartDied
Death of
Partner/Spouse
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeightKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

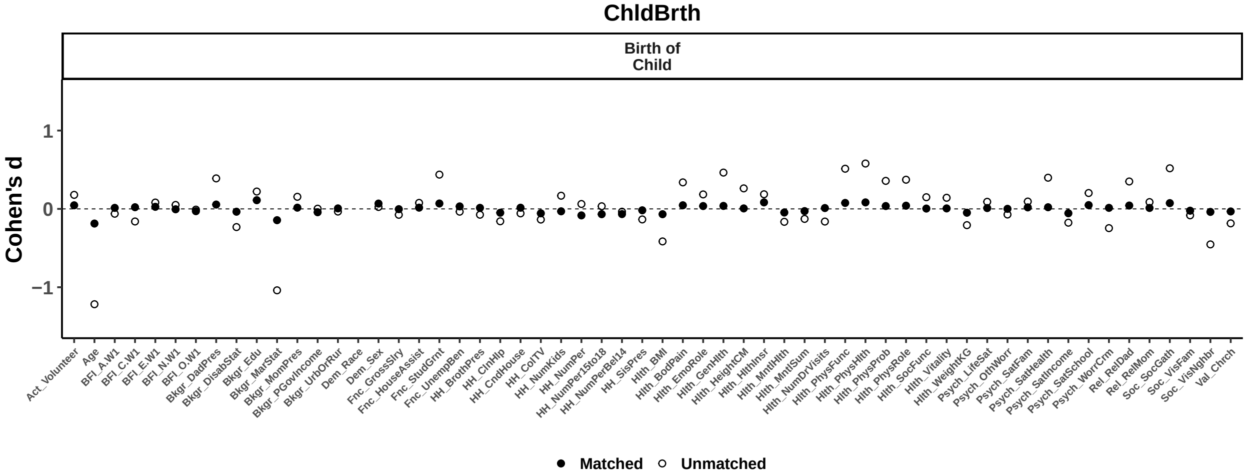
[[5]]


LeftPar
Leaving Parental Home
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeightKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

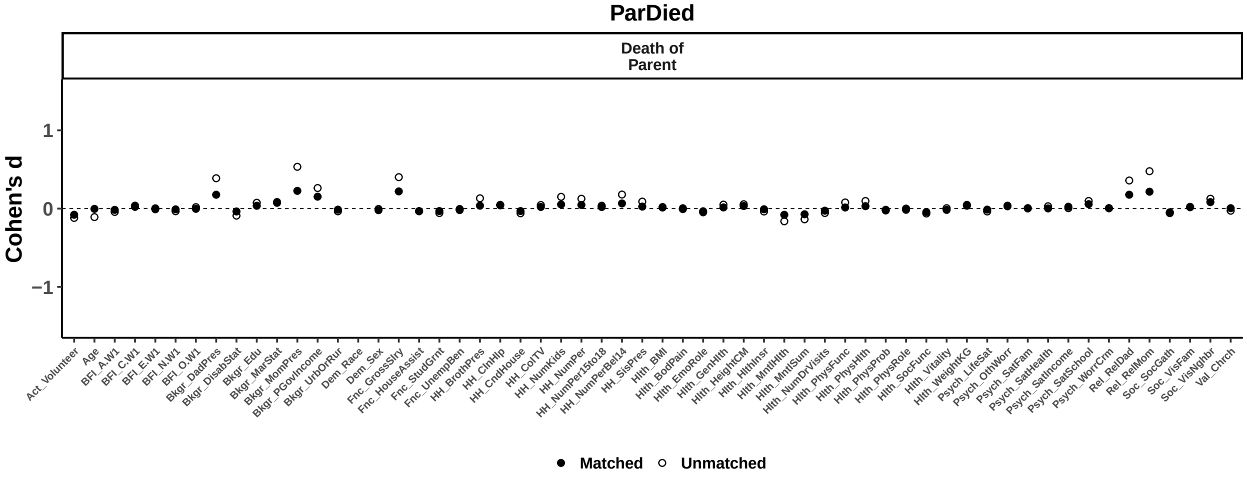
[[6]]


ChldMvOut
Child Leaves Home
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeightKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

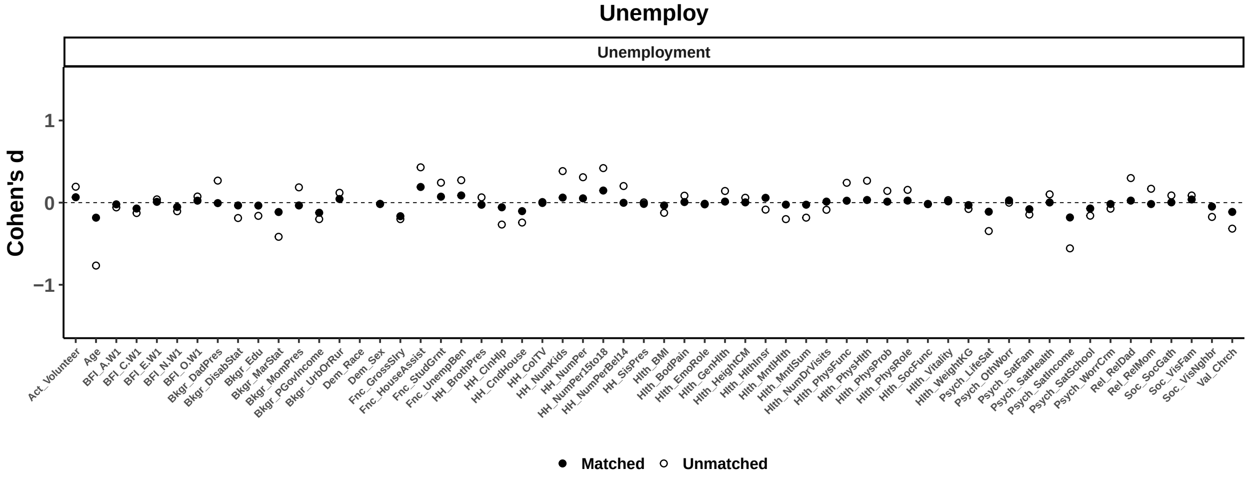
[[7]]


ChldBrth
Birth of Child
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeightKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

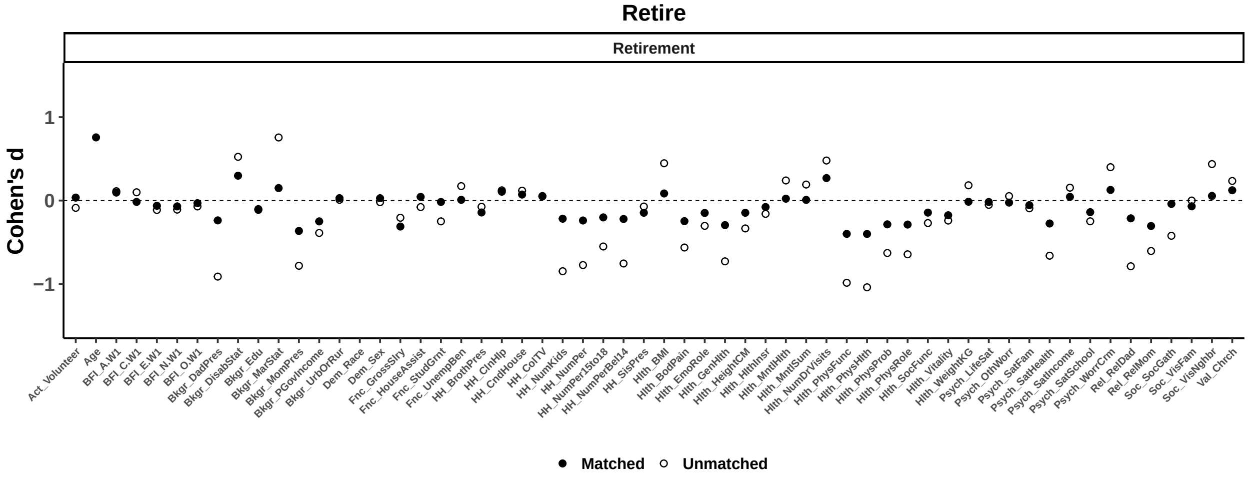
[[8]]


ParDied
Death of Parent
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeighKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

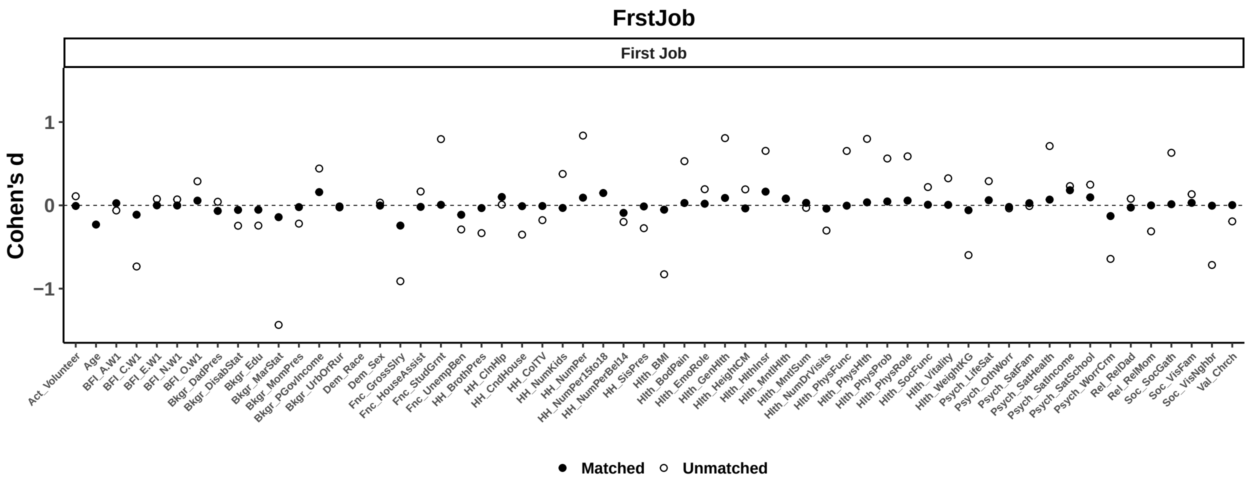
[[9]]


Unemploy
Unemployment
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeighKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

[[10]]


Retire
Retirement
Cohen's d
1
0
−1
Act_Volunteer
Age
BFI_A.W1
BFI_C.W1
BFI_E.W1
BFI_N.W1
BFI_O.W1
Bkgr_DadPres
Bkgr_DisabStat
Bkgr_Edu
Bkgr_MarStat
Bkgr_MomPres
Bkgr_PGovIncome
Bkgr_UrbOrRur
Dem_Race
Dem_Sex
Fnc_GrossSlry
Fnc_HouseAssist
Fnc_StudGrnt
Fnc_UnempBen
HH_BrothPres
HH_ClnHlp
HH_CndHouse
HH_ColTV
HH_NumKids
HH_NumPer
HH_NumPer15to18
HH_NumPerBel14
HH_SisPres
Hlth_BMI
Hlth_BodPain
Hlth_EmoRole
Hlth_GenHlth
Hlth_HeightCM
Hlth_HlthInsr
Hlth_MntlHlth
Hlth_MntlSum
Hlth_NumDrVisits
Hlth_PhysFunc
Hlth_PhysHlth
Hlth_PhysProb
Hlth_PhysRole
Hlth_SocFunc
Hlth_Vitality
Hlth_WeighKG
Psych_LifeSat
Psych_OthWorr
Psych_SatFam
Psych_SatHealth
Psych_SatIncome
Psych_SatSchool
Psych_WorrCrm
Rel_RelDad
Rel_RelMom
Soc_SocGath
Soc_VisFam
Soc_VisNghbr
Val_Chrch
● Matched ○ Unmatched

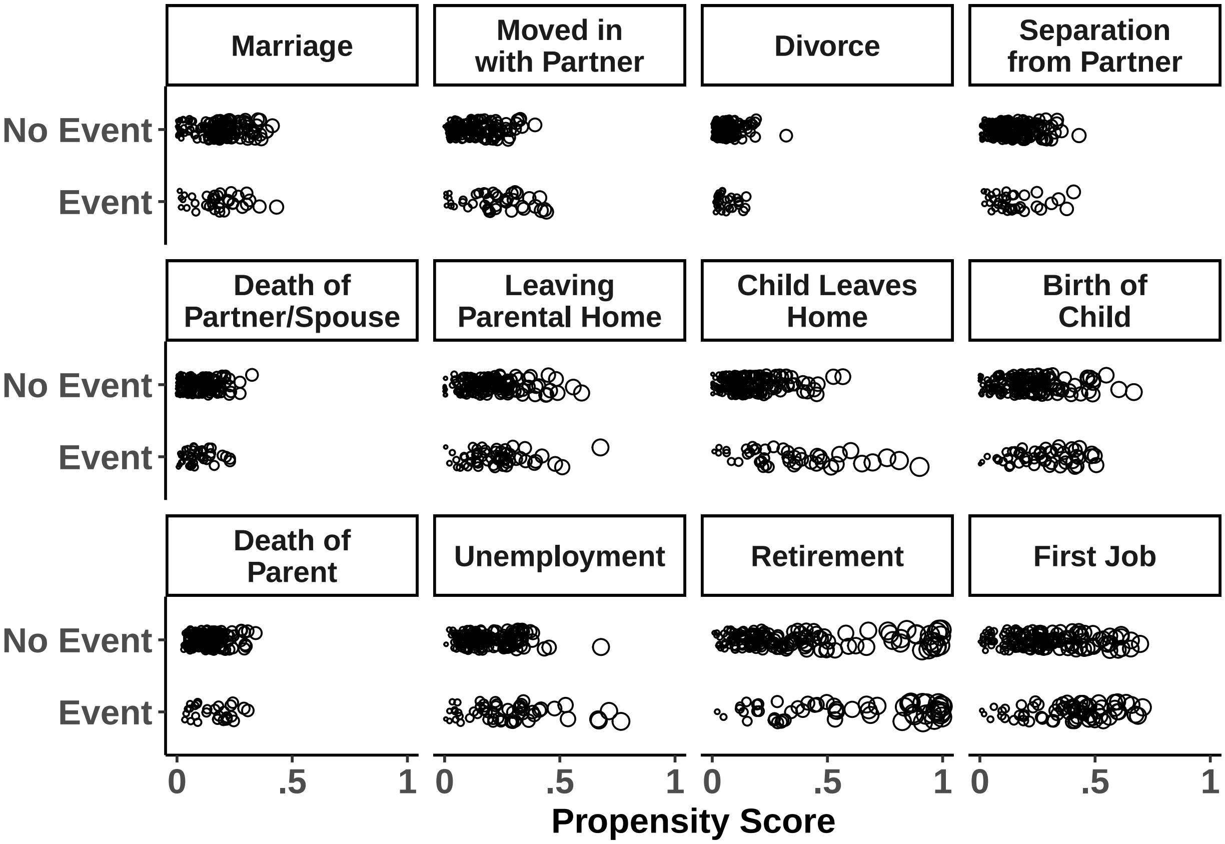
[[11]]



[[12]]

```
ggsave(sprintf("%s/results/plots/psm_match.png", data_path), width = 10, height = 4)
```

```
sample_fun <- function(df){
  df %>% filter(row_number() %in% sample(1:nrow(df), 200))
}

nested.psw %>%
  filter(!Event %in% c("MomDied", "DadDied")) %>%
  mutate(sample = map(psw.df, sample_fun)) %>%
  unnest(sample) %>%
  select(Event:PROC_SID, le.group, distance, weights) %>%
  mutate(le.group = mapvalues(le.group, c(0,1), c("No Event", "Event")),
         Event = mapvalues(Event, events$old, events$breaks),
         Event = factor(Event, levels = events$breaks)) %>%
  filter(chain == 1 & match_set == "socialization") %>%
  ggplot(aes(x = le.group, y = distance)) +
  scale_size_continuous(range = c(.1,3)) +
  scale_y_continuous(breaks = seq(0,1,.5), labels = c("0", ".5", "1")) +
  geom_jitter(aes(size = distance), shape = 1, width = .15) +
  labs(x = NULL, y = "Propensity Score") +
  coord_flip() +
  facet_wrap(~Event, nrow = 3) +
  theme_classic() +
  theme(legend.position = "none",
        strip.text = element_text(face = "bold", size = rel(1)),
        axis.text = element_text(face = "bold", size = rel(1.2)),
        axis.title = element_text(face = "bold", size = rel(1.2)))
```



```
ggsave(file = sprintf("%s/results/plots/ps_plot.png", data_path),
       width = 10, height = 6)
```

Group Differences

```
options(knitr.kable.NA = '')
le_dat %>% mutate(chain = 1, data = "raw", Dem_Sex = factor(Dem_Sex)) %>%
  full_join(nested.psw %>%
    filter(match_set == "socialization") %>%
    unnest(psw.df) %>%
    select(Event, chain, PROC_SID, Dem_DOB, Dem_Sex, le.group) %>%
    mutate(data = "matched")) %>%
  mutate(age = 2005-Dem_DOB) %>%
  #select(-le.value)) %>%
  filter(!is.na(le.group)) %>%
  group_by(Event, data, chain) %>%
  mutate(m.age = mean(age[le.group == 1], na.rm = T),
         sd.age = sd(age[le.group == 1], na.rm = T),
         perc_women = sum(Dem_Sex == 2 & le.group == 1) /
         sum(Dem_Sex %in% c(1,2) & le.group == 1)) %>%
  group_by(Event, chain, data, le.group, m.age, sd.age, perc_women) %>%
  dplyr::summarize(n = n()) %>%
  group_by(Event, data, le.group) %>%
  summarize_at(vars(m.age:n), funs(mean(., na.rm = T))) %>%
  spread(key = le.group, value = n) %>%
  mutate(Frequency = sprintf("%.0f (%.0f)", `1`, (`1` + `0`))) %>%
  mutate_at(vars(m.age:perc_women), funs(sprintf("%.2f", .))) %>%
  select(-`0`, -`1`) %>%
  gather(key = measure, value = value, m.age:Frequency) %>%
  unite(data, measure, data, sep = ".") %>%
  spread(key = data, value = value) %>%
  # select(Event, Frequency, everything()) %>%
  kable(., "latex", booktabs = T, escape = F,
        col.names = c("Life Event", rep(c("Matched", "Raw"), times = 4))) %>%
  kable_styling(latex_options = c("striped","repeat_header"),full_width = F) %>%
  #kable_styling(full_width = F) %>%
  column_spec(1, width = "4cm") %>%
  add_header_above(c(" " = 1, "Frequency" = 2, "$M$" = 2, "$SD$" = 2, "\\% women" = 2)) %>%
  add_header_above(c(" " = 3,  "Age in 2005" = 4, " " = 2))
```

			Age in 2005					
	Frequency		$M$		$SD$		\% women	
Life Event	Matched	Raw	Matched	Raw	Matched	Raw	Matched	Raw
ChldBrth	1116 (4352)	1154 (14132)	31.04	31.00	0.54	0.54	7.73	7.68
ChldMvOut	1814 (7757)	1991 (14124)	48.37	48.35	0.55	0.55	8.11	7.96
DadDied	865 (4320)	898 (14135)	43.47	43.51	0.53	0.53	9.85	9.84
Divorce	362 (1805)	378 (14133)	40.73	40.81	0.59	0.58	8.43	8.49
FrstJob	375 (1123)	449 (14134)	22.11	22.19	0.52	0.54	3.84	4.84
LeftPar	374 (1529)	396 (14133)	23.88	23.96	0.51	0.49	7.88	7.88
Married	965 (4453)	1002 (14132)	33.36	33.35	0.51	0.51	11.04	10.96
MomDied	906 (4497)	949 (14136)	50.11	50.13	0.51	0.51	9.87	9.88
MoveIn	885 (4898)	952 (14131)	31.55	31.38	0.53	0.53	10.76	10.82
ParDied	1633 (10614)	1710 (14131)	46.78	46.84	0.53	0.52	10.63	10.62
PartDied	390 (1926)	411 (14136)	64.08	64.20	0.70	0.71	11.74	11.62
Retire	1166 (3892)	4696 (14113)	60.55	64.90	0.54	0.52	10.79	8.98
SepPart	947 (4528)	1006 (14127)	35.62	35.69	0.57	0.56	11.36	11.45
Unemploy	1910 (7501)	2065 (14122)	38.45	38.05	0.52	0.52	12.68	12.76

Selection Effects: Logistic Regressions

Run Models

```
load(sprintf("%s/results/psw_small.RData", data_path))
load(paste(data_path, "results/mi_dat_small.RData", sep = "/"))

bfi_selection <- crossing(
  Event = c(unique(nested.psw$Event)),
  Trait = unique(bfi_wide$Trait),
  match = c("Matched", "Unmatched")
  )

sel_fun <- function (event, trait, match){
  k <- 1#ifelse(event == "ParDied" & trait == "A" & match == "Matched", 4, 1)
  lapply(k:10, function(x){
    print(paste(event, trait, match, x), sep = " ")
    if(match == "Matched"){
      subs <- unique((nested.psw %>%
          filter(match_set == "selection" & chain == 1 & Event == event) %>%
          unnest(psw.df))$PROC_SID)
      df <- bfi.imp %>% filter(chain == x & Trait == trait & PROC_SID %in% subs)
    } else {
      df <- bfi.imp %>% filter(chain == x & Trait == trait)
    }
    df <- df %>%
      mutate(value = rowMeans(cbind(T1_1, T1_2, T1_3), na.rm = T)) %>%
        select(PROC_SID, sex12:age.c3, value) %>%
        left_join(le_dat %>% filter(Event == event) %>% select(PROC_SID, le.group))
      mod <- rstanarm::stan_glm(
        le.group ~ value, family = binomial(link = "logit"), data = df
      )
      file <- sprintf("%s/results/selection/%s_%s_%s_chain%s.RData", data_path, trait, event, match, x)
      save(mod, file = file)
      rm(mod)
  return(T)
  })
}

start <- Sys.time()
bfi_selection <- bfi_selection %>%
  mutate(b.sel.mod = pmap(list(Event, Trait, match), sel_fun))
end <- Sys.time()
```

Pool Results

```
stantab_fun <- function(event, trait, match){
    models <- lapply(1:10, function(x){
      print(x)
      file <- sprintf("%s/results/selection/%s_%s_%s_chain%s.RData",
                      data_path, trait, event, match, x)
      load(file)
      return(mod)
  })
    sflist <- lapply(models, "[[", "stanfit")
    models[[1]]$stanfit <- rstan::sflist2stanfit(sflist)
    models[[1]]
}

tab_fun <- function(models, type){
  # model is a list of models
  # taret.var is a character vector of latent parameters
    UI_fun <- function(mod){
      UI <- rstanarm::posterior_interval(mod, prob = 0.95)
      names <- paste(rep(row.names(UI),2), rep(colnames(UI),each=2))
      UI <- as.vector(UI)
      names(UI) <- names
      return(UI)
    }
  ###
  ### Get FE's and RE's for pooling FE's across imputations ###
  ####
  nchains   <- length(models)
  coefs     <- sapply(models, get_parameters)
  variances <- sapply(models, FUN = function(x) diag(vcov(x)))
  CI        <- sapply(models, UI_fun)

  # average effects for each term
  coefs_mean <- apply(coefs,     1, mean)
  var_mean   <- apply(variances, 1, mean)
  CI_means   <- apply(CI,        1, mean)
  names(CI_means) <- paste(rep(names(coefs_mean),2),
                           rep(c("lower", "upper"),each=2), sep = "_")

  # variance of fixed effects
  coefs_var <- apply(coefs, 1, var)

  fixeff <- tibble(
    type = rep("fixeff",nrow(coefs)),
    term = rownames(coefs),
    Estimate = coefs_mean,
    # t = t.val,
    CI.lower = CI_means[grepl("lower", names(CI_means))],
    CI.upper = CI_means[grepl("upper", names(CI_means))]
  )
}
```

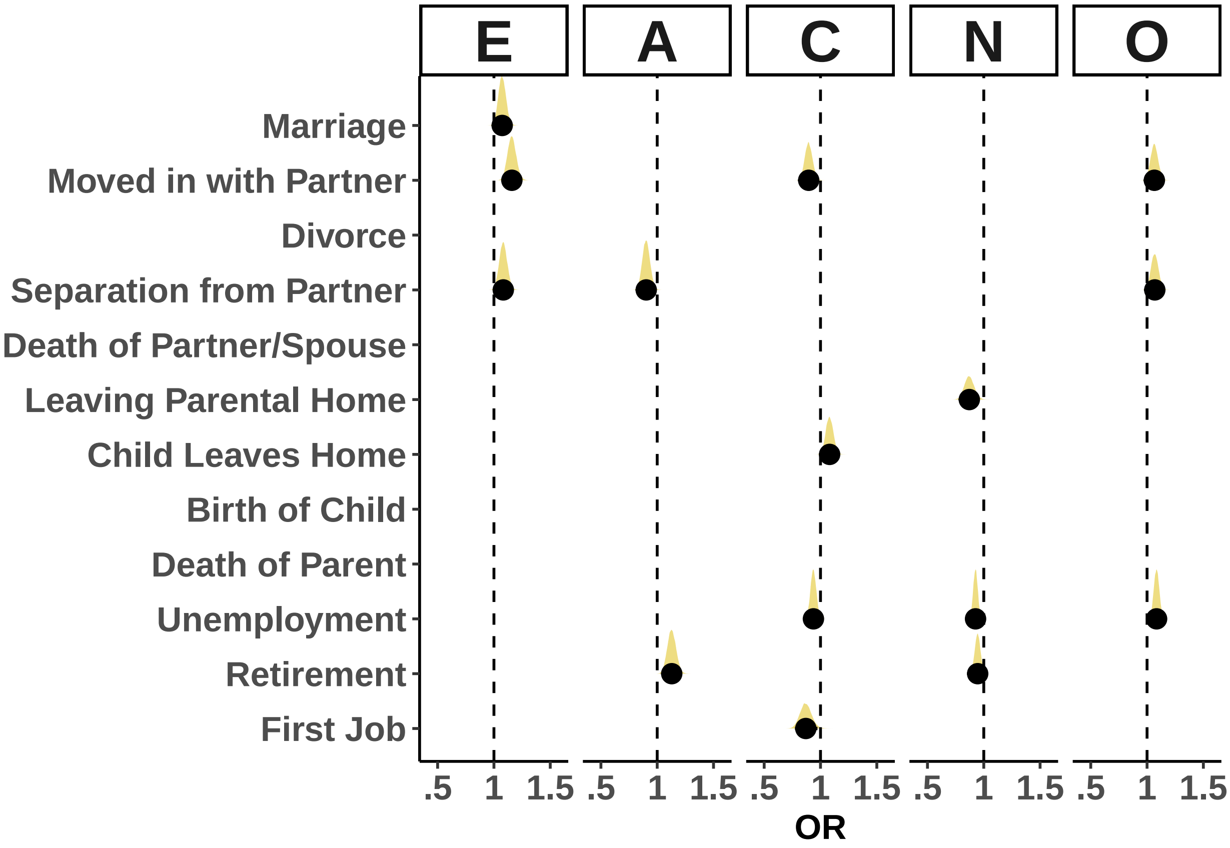
Table Results

```
bfi_selection <- crossing(
  Event = c(unique(le_dat$Event)),
  Trait = unique(bfi_wide$Trait),
  match = c("Matched", "Unmatched")
  ) %>%
  mutate(pool = pmap(list(Event, Trait, match), stantab_fun),
         samples = map(pool, ~gather_samples(., value)),
         qi = map(samples, mean_qi))
save(bfi_selection, file = sprintf("%s/results/selection.RData", data_path))
```

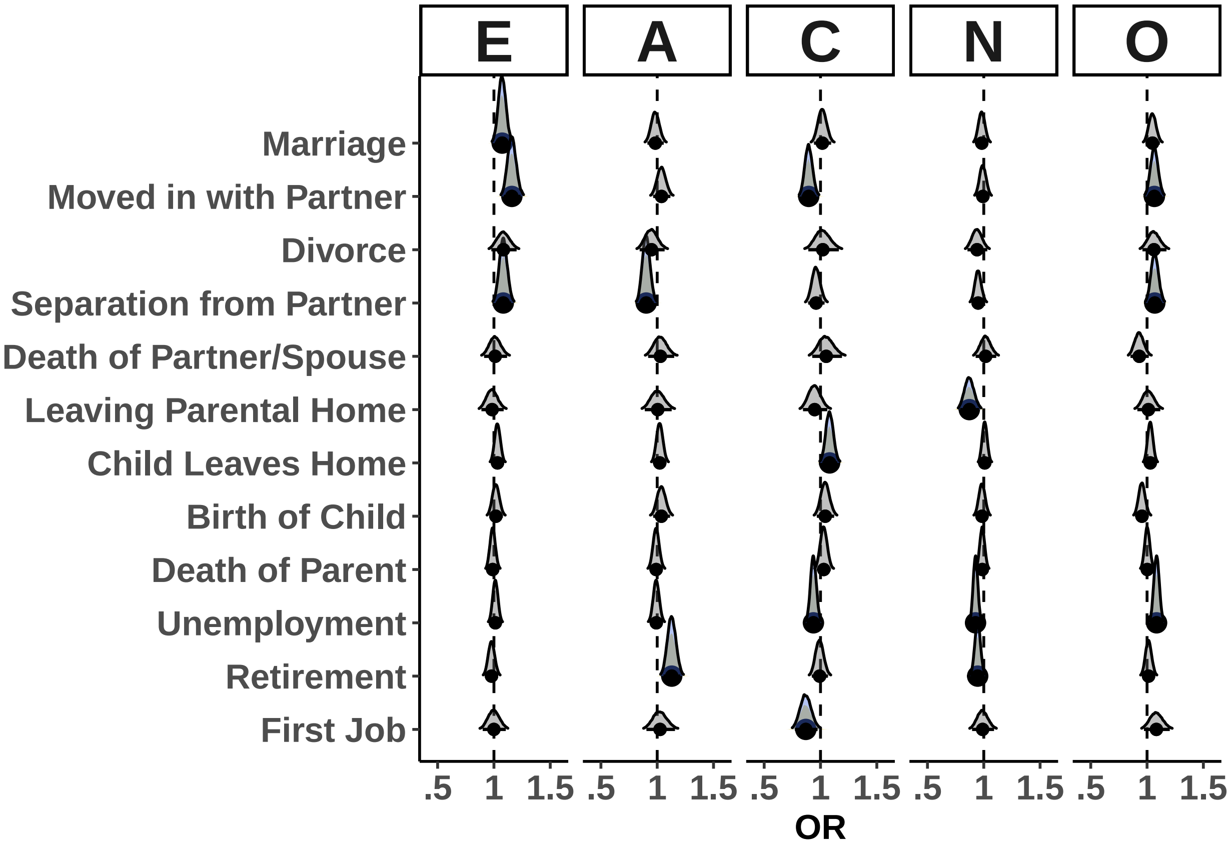
Plots

```
load(sprintf("%s/results/selection.RData", data_path))
bfi_samples <- bfi_selection %>% unnest(samples) %>%
  filter(term == "value") %>%
  mutate(estimate = exp(estimate))
save(bfi_samples, files = sprintf("%s/results/selection_samples.RData", data_path))

bfi_selection <- bfi_selection %>% select(-samples, -pool)

save(sel.tab, bfi_selection, file = sprintf("%s/results/sel.tab.RData", data_path))
```

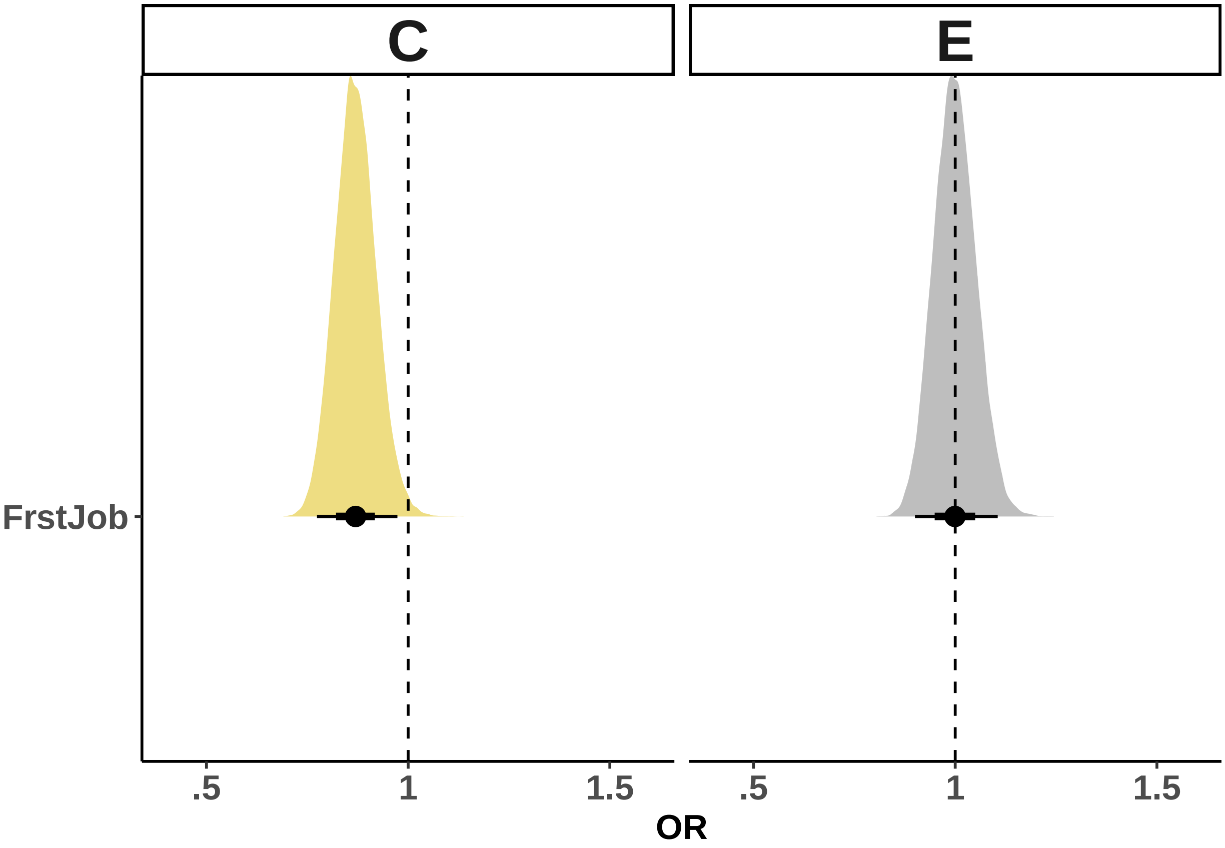
```
load(sprintf("%s/results/selection.RData", data_path))
load(sprintf("%s/results/selection_samples.RData", data_path))
load(sprintf("%s/results/sel.tab.RData", data_path))

bfi_qi <- bfi_selection %>% unnest(qi) %>%
  filter(term == "value") %>%
  mutate(sig = ifelse(sign(conf.low) == sign(conf.high), "sig", "ns")) %>%
  mutate_at(vars(estimate:conf.high), funs(exp))

(p <- bfi_samples %>% full_join(bfi_qi %>% select(Event:term, sig)) %>%
    filter(!(Event %in% c("MomDied", "DadDied"))) %>%
  mutate(Trait = factor(Trait, levels = c("E", "A", "C", "N", "O")),
         Event = mapvalues(Event, events$old, events$new),
         Event = factor(Event, rev(events$new))) %>%
  ggplot(aes(y = Event, x = estimate)) +
  scale_x_continuous(limits = c(.4, 1.6), breaks = seq(.5, 1.5, .5), labels = c(".5", "1", "1.5")) +
  # geom_density_ridges(data = . %>% filter(match == "Matched"), aes(fill = sig), rel_min_height = 0.025, scale = 4) +
    # geom_halfeyeh(data = . %>% filter(match == "Unmatched"), aes(fill = sig)) +
    geom_blank(data = bfi_samples %>% filter(!(Event %in% c("MomDied", "DadDied"))) %>%
                 mutate(Trait = factor(Trait, levels = c("E", "A", "C", "N", "O")),
                        Event = mapvalues(Event, events$old, events$new),
                        Event = factor(Event, rev(events$new)))) +
    geom_halfeyeh(data = . %>% filter(match == "Matched" & sig == "sig"), fill = "lightgoldenrod") +
  geom_vline(aes(xintercept = 1), linetype = "dashed") +
    scale_fill_manual(values = c("gray", "lightgoldenrod")) +
  labs(x = "OR", y = NULL, fill = NULL, color = NULL) +
  facet_wrap(~Trait, nrow = 1) +
  theme_classic() +
  theme(legend.position = "none",
        axis.text = element_text(face = "bold", size = rel(1.2)),
        strip.text = element_text(face = "bold", size = rel(2)),
        axis.title = element_text(face = "bold", size = rel(1.2)),
        legend.text = element_text(face = "bold", size = rel(1.2))))
```



```r
ggsave(p, file = sprintf("%s/results/plots/selection_matched_posterior_sig.png", data_path),
       width = 14, height = 6)

p +  geom_density_ridges(data = . %>% filter(match == "Matched" & sig == "sig"),
                         alpha= .4, scale = .8, rel_min_height = 0.025, fill = "royalblue") +
  geom_density_ridges(data = . %>% filter(match == "Matched" & sig == "ns"),
                      alpha= .4, scale = .8, rel_min_height = 0.025, fill = "gray40") +
  stat_pointintervalh(data = . %>% filter(match == "Matched" & sig == "sig"),
                      size = .8, .prob = c(.66, .95)) +
  stat_pointintervalh(data = . %>% filter(match == "Matched" & sig == "ns"),
                      size = .8, .prob = c(.66, .95))
```



```
ggsave(file = sprintf("%s/results/plots/selection_posterior_sig.png", data_path),
       width = 10, height = 6)


# example plot

bfi_samples %>% full_join(bfi_qi %>% select(Event:term, sig)) %>%
  filter(Trait %in% c("C", "E") & Event == "FrstJob") %>%
  ggplot(aes(y = Event, x = estimate)) +
    scale_x_continuous(limits = c(.4, 1.6), breaks = seq(.5, 1.5, .5),
                       labels = c(".5", "1", "1.5")) +
    geom_halfeyeh(data = . %>% filter(match == "Matched"), aes(fill = sig)) +
    geom_vline(aes(xintercept = 1), linetype = "dashed") +
    scale_fill_manual(values = c("gray", "lightgoldenrod")) +
    labs(x = "OR", y = NULL, fill = NULL, color = NULL) +
    facet_wrap(~Trait, nrow = 1) +
    theme_classic() +
    theme(legend.position = "none",
          axis.text = element_text(face = "bold", size = rel(1.2)),
          strip.text = element_text(face = "bold", size = rel(2)),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.text = element_text(face = "bold", size = rel(1.2)))
```



```r
ggsave(file = sprintf("%s/results/plots/selection_example_plot.png", data_path),
       width = 6, height = 2)
```

Socialization Effects: Growth Models

Define Functions

Standard Errors

```r
# function to get the standard errors
se_fun <- function(vcov_mat, x, m){
  print("se_fun")
  se_map <- function(vcov, parameter, w){
    z <- 1
    vcov[parameter, parameter] + 2 * z * vcov[parameter, w] + z^2 * vcov[w, w]
  }
  cols <- colnames(vcov_mat)
  x <- cols[grepl(x,cols) & !grepl(":", cols)]
  se <- expand.grid(
    parameter = c("Intercept", x),
    term = m,
    stringsAsFactors = F
  ) %>% tbl_df() %>%
    mutate(se = map2_dbl(parameter, term, ~se_map(vcov_mat, .x, .y)),
           # term = str_replace(term, "groups", ""),
           parameter = ifelse(parameter == "Intercept", parameter, "Slope"))
return(se)
}
```

Hypothesis Tests

```r
# function to test hypotheses for slopes and intercepts in each group
hyp_fun <- function(fit, event){
  print("hyp_fun")
  me <-
    fixef(fit) %>% data.frame %>% mutate(term = rownames(.)) %>%
    filter(term %in% c("Intercept", "new.wave")) %>%
    mutate(term.type = mapvalues(term, unique(term), c("Intercept", "Slope")),
           term = "No Event") %>%
    setNames(c("b", "SE", "lower", "upper", "term", "term.type"))
  if(event != "none"){
  h <- c(
    "new.wave + new.wave:le.group1 = 0",
    "Intercept + le.group1 = 0"
  )
  h <- hypothesis(fit, h, alpha = .05)
  me <- h$hypothesis %>% data.frame %>%
    mutate(term = "Event", term.type = c("Slope", "Intercept")) %>%
    dplyr::select(-Evid.Ratio, -Star) %>%
    setNames(c("b", "SE", "lower", "upper", "term", "term.type")) %>%
    full_join(me)
  }
  return(me)
}
```

Predicted Values

```r
# function to get predicted fixed and random effects for plotting
plot_fun <- function(fit, event){
  print("plot_fun")
    frame <- crossing(
      new.wave = seq(0,2, .01),
      le.group = 0:1
    ) %>%
      tbl_df()
  if (event == "none"){
    frame <- frame %>% dplyr::select(-le.group)
  }
  combine_fun <- function(x,y){
      cbind(x,y)
  }

   frame <- frame %>%
     nest() %>%
     mutate(pred = map(data, ~fitted(fit, newdata = ., re_formula = NA, probs = c(.025, .975))),
            data = map2(data, pred, combine_fun)) %>%
     unnest(data, .drop = T)

   ranef_frame <- fit$data %>%
     nest() %>%
     mutate(pred = map(data, ~predict(fit, probs = c(.025, .975))),
            data = map2(data, pred, combine_fun)) %>%
     unnest(data, .drop = T)
   return(list(fixed = frame, ranef = ranef_frame))
}
```

Pooling

```r
pool_fun <- function(event, trait){
  cat(event, trait, sep = " ")
  nchains = 10

eff_fun <- function(chain){
    file <- sprintf("%s/unpooled/%s_%s_chain%s_brm_cv.RData", model_path, trait, event, chain)
    load(file)
    fixeffs <- fixef(fit, probs = c(.025,.975))
    rownames(fixeffs) <- gsub("1", "", rownames(fixeffs)); colnames(fixeffs) <- gsub("1", "", colnames(fixeffs))
    vc <- vcov(fit, probs = c(.025,.975))
    rownames(vc) <- gsub("1", "", rownames(vc)); colnames(vc) <- gsub("1", "", colnames(vc))
    dvc <- diag(vc)

    raneffs <- broom::tidy(fit, probs = c(.025, .975))
    ind_ranef <- raneffs %>% tbl_df %>%
      filter(grepl("r_PROC_SID\\[", term)) %>%
      mutate(term = str_replace(term, "r_PROC_SID", ""),
             term = str_replace(term, "\\[*", ""),
             term = str_replace(term, "\\]$", "")) %>%
      separate(term, c("PROC_SID", "term"), sep = ",")
    eff <- fixef(fit, probs = c(.025, .975))[, 1]
    if(event != "none"){
      ind_coef <- le_dat %>% filter(Event == event & PROC_SID %in% unique(ind_ranef$PROC_SID)) %>%
        select(PROC_SID, le.group) %>%
        full_join(ind_ranef %>% mutate(PROC_SID = as.numeric(PROC_SID))) %>%
        mutate(term = str_replace(term, "le.group1", "le.group"),
               b = ifelse(term == "Slope", eff["new.wave"] + eff["new.wave:le.group"]*le.group + estimate,
               eff["Intercept"] + eff["le.group"]*le.group + estimate))
      } else{
        ind_coef <- ind_ranef %>% mutate(PROC_SID = as.numeric(PROC_SID)) %>%
          mutate(b = ifelse(term == "Slope", eff["new.wave"] + estimate, eff["Intercept"] + estimate),
                 le.group = 0)
      }

    L2 <- raneffs %>% tbl_df() %>%
      filter(grepl("sd_", term) | grepl("cor_", term) | term == "sigma") %>%
      mutate(type = "raneff") %>%
      mutate_at(vars(estimate:upper), funs(ifelse(grepl("cor_", term) == F, .^2,.))) %>%
      mutate(term = ifelse(term != "sigma", str_extract(term, "(?<=__).*$"), term),
             term = mapvalues(term, unique(term),
          c("\\tau_{00}", "\\tau_{01}", "\\tau_{11}", "\\sigma^2")))

    pred <- plot_fun(fit, event)
    fixef_pred <- pred$fixed; ranef_pred <- pred$ranef
    se <- if(event != "none"){se_fun(vc, "new.wave", "le.group")} else {NA_real_}
    h <- hyp_fun(fit, event)

    results <- list(fx = fixeffs, vc = vc, L2 = L2, re = ind_coef, hyp = h,
                    fixef_pred = fixef_pred, ranef_pred = ranef_pred, se = se)
    return(results)
  }

res <- tibble(chain = 1:nchains) %>%
    mutate(fx = map(chain, eff_fun),
           vc = map(fx, ~.$vc),
           L2 = map(fx, ~.$L2),
           re = map(fx, ~.$re),
           hyp = map(fx, ~.$hyp),
           fixef_pred = map(fx, ~.$fixef_pred),
           ranef_pred = map(fx, ~.$ranef_pred),
           se = map(fx, ~.$se),
           fx = map(fx, ~.$fx))

vcov_mean    <- apply(simplify2array(res$vc), 1:2, mean)
fixeffs_mean <- apply(simplify2array(res$fx), 1:2, mean)
fixeff_var  <- apply(simplify2array(res$fx), 1:2, var)
raneffs_mean <- diag(vcov_mean)

T <- raneffs_mean + (1 + nchains^(-1)) * fixeff_var[,"Estimate"] # ??
r <- (1 + nchains^(-1)) * fixeff_var/raneffs_mean# RIV value
df <- (nchains - 1) * (1 + r^(-1))^2
se <- sqrt(T)
```

```
t.val <- fixeffs_mean[,"Estimate"]/se
p <- 2 * (1 - pt(abs(t.val), df = df))
CI = qt(.975, df = df)*se

fixeff <- fixeffs_mean %>% data.frame %>% mutate(term = rownames(.)) %>%
  setNames(c("estimate", "std.error", "lower", "upper", "term")) %>%
  mutate(type = "fixeff",
         term = str_replace(term, "new.wave", "Slope"))

raneff <- res %>% unnest(L2) %>%
  group_by(term, type) %>%
  summarize_all(funs(mean)) %>%
    ungroup()

pooled_re <- res %>% unnest(re) %>%
  group_by(term, PROC_SID, le.group) %>%
  summarize_all(funs(mean)) %>%
  ungroup()

if(any(grepl("le.group", rownames(fixeffs_mean)))){
    re_sd <- (pooled_re %>% filter(term == "Intercept" & le.group == 0) %>%
      summarise(sd = sd(b)))$sd
    fixeff <- fixeff %>%
      mutate(d = ifelse(grepl("le.group", term) == F, NA, estimate/re_sd))
}

 # if (type != "bayesian"){
 #    sum_mod <- tribble(
 #      ~type, ~term, ~Estimate,
 #      "summary", "CFI", fitmeas_mean["cfi"],
 #      "summary", "RMSEA", fitmeas_mean["rmsea"],
 #      "summary", "$\\chi^2$", fitmeas_mean["chisq"],
 #      "summary", "df", fitmeas_mean["df"]
 #    )
 #  } # else {
  #   sum_mod <- tribble(
  #     ~type, ~term, ~Estimate,
  #     "summary", "bic", fitmeas_mean["bic"],
  #     "summary", "waic", fitmeas_mean["waic"],
  #     "summary", "margloglik", fitmeas_mean["margloglik"]
  #   )
  # }
  results <- fixeff %>% full_join(raneff) #%>% full_join(sum_mod)

  hyp_mean <- res %>% unnest(hyp) %>%
    group_by(term, term.type) %>%
    summarize_all(funs(mean)) %>%
    ungroup()

  fixef_pred <- res %>% unnest(fixef_pred) %>%
    group_by(new.wave, le.group) %>%
    summarize_all(funs(mean)) %>%
    ungroup()

  ranef_pred <- res %>% unnest(ranef_pred) %>%
    group_by(new.wave, le.group, PROC_SID) %>%
    summarize_all(funs(mean)) %>%
    ungroup()

  se <- res %>% unnest(se) %>%
    group_by(parameter,term) %>%
    summarise_all(funs(mean)) %>%
    ungroup()

  results <- list(fixef_pred = fixef_pred, ranef_pred = ranef_pred, table = results,
              ranef.tab = pooled_re, vcov = vcov_mean, se = se, hypoth = hyp_mean)
  save(results, file = sprintf("%s/results/model results/pooled/%s_%s.RData", data_path, trait, event))
  beepr::beep(sound = 2)
  return(TRUE)
}
```

brm Models

```
growth_fun <- function(event, trait){
  no_cores <- detectCores()-1
  k <- ifelse(trait == "C", 10,1)
  lapply(k:10, function(x){
      rstan_options(auto_write = TRUE)
      options(mc.cores = parallel::detectCores()-1)
    print(paste(event, trait, x), sep = " ")
    if(event == "none"){
      subs <- unique((nested.psw %>% filter(match_set == "socialization" & chain == 1) %>%
                        unnest(psw.df))$PROC_SID)
      df <- bfi.imp %>% filter(chain == x & Trait == trait & PROC_SID %in% subs) %>%
        gather(key = item, value = value, T1_1:T3_3) %>%
        separate(item, c("wave", "item"), sep = "_") %>%
        mutate(new.wave = as.numeric(mapvalues(wave, c("T1", "T2", "T3"), 0:2))) %>%
        group_by(PROC_SID, new.wave, sex.c, age, age.c, age.c2, age.c3) %>%
        summarize(value = mean(value, na.rm = T))
      Prior <- get_prior(value ~ new.wave + (new.wave|PROC_SID), data = df)
      Prior <- set_prior("cauchy(0,10)", class = "sd")
      start.tmp <- Sys.time()
      fit <- brm(value ~ new.wave + (new.wave|PROC_SID),
                 data = df, control = list(adapt_delta = 0.99))
      # fit <- lmer(value ~ sex.c*new.wave + age.c*new.wave + age.c2*new.wave + (new.wave|PROC_SID), data = df)
      end.tmp <- Sys.time()
    } else{
      subs <- unique((nested.psw %>%
        filter(match_set == "socialization" & chain == 1 & Event == event) %>%
          unnest(psw.df))$PROC_SID)
      df <- bfi.imp %>% filter(Trait == trait & chain == x & PROC_SID %in% subs) %>%
        gather(key = item, value = value, T1_1:T3_3) %>%
        separate(item, c("wave", "item"), sep = "_") %>%
        mutate(new.wave = as.numeric(mapvalues(wave, c("T1", "T2", "T3"), 0:2))) %>%
        group_by(PROC_SID, new.wave, sex.c, age, age.c, age.c2, age.c3) %>%
        summarize(value = mean(value, na.rm = T)) %>%
        left_join(le_dat %>% select(Event, PROC_SID, le.group) %>% filter(Event == event)) %>%
        mutate(le.group = factor(le.group))
      if((trait == "A" & event %in% c("ChldMvOut", "MoveIn", "ChldBrth")) |
         (trait == "E" & event %in% c("ParDied")) |
         (trait == "C" & event %in% c("Retire")) |
         (trait == "O" & event %in% c("Unemploy")) |
         event %in% c("PartDied", "FrstJob", "Divorce", "LeftPar")){
          Iter <- 8000; Warmup <- 4000; treedepth <- 20
         } else {Iter <- 2000; Warmup <- 1000; treedepth <- 10}
          Prior <- get_prior(value ~ new.wave*le.group + (new.wave|PROC_SID), data = df)
          Prior <-  c(set_prior("cauchy(0,1)", class = "sd"), set_prior("cauchy(0,1)", class = "sigma"))
      start.tmp <- Sys.time()
      fit <- brm(value ~ new.wave*le.group + (new.wave|PROC_SID),  data = df,  prior = Prior,
                 control = list(adapt_delta = 0.99, max_treedepth = treedepth), iter = Iter, warmup = Warmup)
      end.tmp <- Sys.time()
      }
      print(end.tmp - start.tmp)
      file <- sprintf("%s/unpooled/%s_%s_chain%s_brm_cv.RData", model_path, trait, event, x)
      save(fit, file = file)
      rm(fit)
  return(T)
  })
}
```

Run Models

```r
load(sprintf("%s/results/psw_small.RData", data_path))
load(paste(data_path, "results/mi_dat_small.RData", sep = "/"))

bfi_growth <- crossing(
  Event = c(unique(nested.psw$Event)),#, "none"),
  Trait = unique(bfi_wide$Trait)
  ) %>%
  mutate(b.grp.mod = map2(Event, Trait, growth_fun))
```

Pool Results

```r
bfi_growth <- crossing(
  Event = c("none", unique(le_dat$Event)),
  Trait = unique(bfi_wide$Trait)
  ) %>%
  mutate(b.grp.mod = map2(Event, Trait, pool_fun))
save(bfi_growth, file = sprintf("%s/results/lav_growth.RData", data_path))
```

Load Results

The models take days to weeks to run, and pooling takes at least an additional day, so we're going to load in the results of the models, rather than the models themselves.

```r
model_path <- "~/Box/Models/PCLE Replication"
files <- list.files(sprintf("%s/results/model results/pooled", data_path))
load_fun <- function(event, trait){
  file <- sprintf("%s/results/model results/pooled/%s_%s.RData", data_path, trait, event)
  load(file)
  return(results)
}

bfi_growth <- crossing(
  Event = c(unique(le_dat$Event), "none"),
  Trait = unique(bfi_wide$Trait)
  ) %>% filter(!(Event %in% c("DadDied", "MomDied"))) %>%
  mutate(b.grp.pool = map2(Event, Trait, load_fun),
         b.fixef.tab = map(b.grp.pool, ~.$table),
         b.ranef.tab = map(b.grp.pool, ~.$ranef.tab),
         b.vcov.mat = map(b.grp.pool, ~.$vc),
         b.se = map(b.grp.pool, ~.$se),
         b.fixef.pred = map(b.grp.pool, ~.$fixef_pred),
         b.ranef.pred = map(b.grp.pool, ~.$ranef_pred),
         b.hyp = map(b.grp.pool, ~.$hypoth))
save(bfi_growth, file = sprintf("%s/results/lav_growth.RData", data_path))
```

Table Results

```r
load(sprintf("%s/results/lav_growth.RData", data_path))
events <- unique(bfi_growth$Event)
events <- events[events != "none"]

bfi_growth_tab <- bfi_growth %>% unnest(b.fixef.tab, .drop = T) %>%
  filter(!grepl("age.c", term) & !grepl("sex.c", term) & term != "df") %>%
  filter((Event == "none" & term %in% c("Intercept", "new.wave")) |
         (Event != "none" & !(term %in% c("Intercept", "new.wave"))) |
          type %in% c("summary", "raneff")) %>%
  filter(!(Event != "none" & type == "summary")) %>%
  mutate(term = ifelse(term == "le.group", "Intercept",
                ifelse(term == "new.wave:le.group", "Slope", term)),
         CI = sprintf("[%.2f, %.2f]", lower, upper),
         CI = ifelse(sign(lower) == sign(upper),
                     sprintf("\\textbf{%s}", CI), CI),
         estimate = round(estimate, 2),
         estimate = ifelse(type != "summary" & sign(lower) == sign(upper),
            sprintf("\\textbf{%s}", estimate), estimate)) %>%
  select(-lower, -upper, -d, -std.error) %>%
  gather(key = est, value = value, estimate, CI) %>%
  mutate(est = mapvalues(est, c("estimate", "CI"), c("CI", "b"))) %>%
  unite(comb, Trait, est, sep = ".") %>%
  spread(key = comb, value = value) %>%
  mutate(Event = factor(Event, levels = c("none", events))) %>%
  arrange(type, term, Event) %>%
  select(type, term, everything())
```

Extract Samples

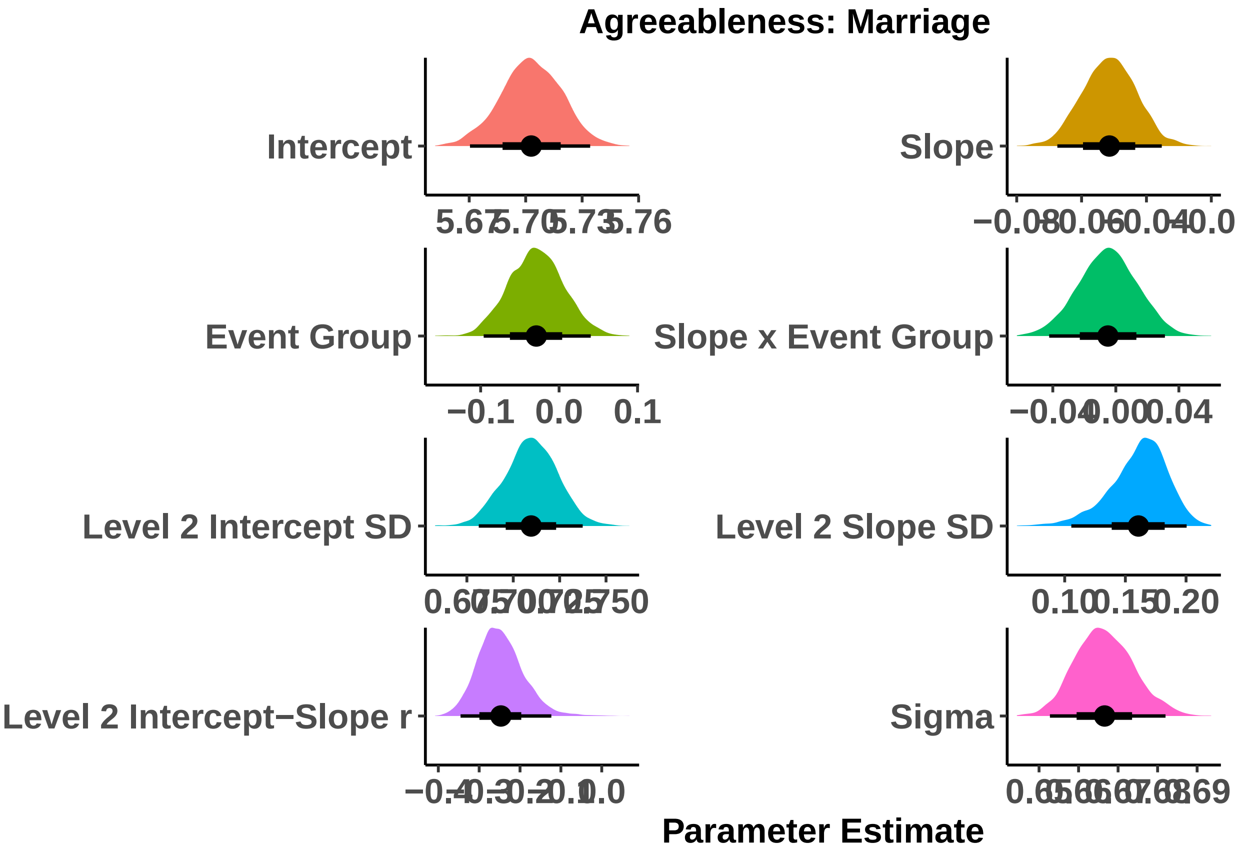
```r
sample_fun <- function(event, trait){
  load(sprintf("%s/unpooled/%s_%s_chain1_brm_cv.RData", model_path, trait, event))
  post <- posterior_samples(fit, add_chain = T, pars = c("b_Intercept", "b_new.wave", "b_le.group",
        "cor_PROC_SID__Intercept__new.wave", "sd_PROC_SID__Intercept", "sd_PROC_SID__new.wave", "sigma")) %>%
    tbl_df %>%
    setNames(c("Intercept", "Slope", "Event Group", "Slope x Event Group", "Level 2 Intercept SD",
             "Level 2 Slope SD", "Level 2 Intercept-Slope r", "Sigma", "Chain", "Iter")) %>%
    gather(key = term, value = estimate, Intercept:Sigma)
  # save(post, file = sprintf("%s/results/samples/%s_%s.RData", data_path, trait, event))
  return(post)
}

bfi_growth <- crossing(
  Event = c(unique(le_dat$Event)),#, "none"),
  Trait = unique(bfi_wide$Trait)
  ) %>%
  mutate(samples = map2(Event, Trait, sample_fun))
```

```r
load_fun <- function(event, trait)  {
  load(sprintf("%s/results/samples/%s_%s.RData", data_path, trait, event))
  return(post)
}

bfi_growth <- bfi_growth %>%
  mutate(samples = map2(Event, Trait, load_fun))

growth_samples <- bfi_growth %>% unnest(samples) %>% tbl_df

save(growth_samples, file = sprintf("%s/results/growth_samples.RData", data_path))
```

Plot Results

Sample Posterior Distributions

```r
load(sprintf("%s/results/growth_samples.RData", data_path))

levs <- c("Intercept", "Slope", "Event Group", "Slope x Event Group", "Level 2 Intercept SD",
             "Level 2 Slope SD", "Level 2 Intercept-Slope r", "Sigma")

growth_samples %>% filter(Trait == "A" & Event == "Married") %>%
  mutate(term = factor(term, levels = levs)) %>%
  ggplot(aes(y = term, x = estimate, fill = term)) +
  geom_halfeyeh() +
  facet_wrap(~term, scales = "free", ncol = 2, strip.position = "right") +
  labs(x = "Parameter Estimate", y = NULL, title = "Agreeableness: Marriage") +
  theme_classic() +
  theme(strip.text = element_blank(),
        legend.position = "none",
        axis.text = element_text(face = "bold", size = rel(1.2)),
        axis.title = element_text(face = "bold", size = rel(1.2)),
        plot.title = element_text(face = "bold", size = rel(1.2), hjust = .4))
```

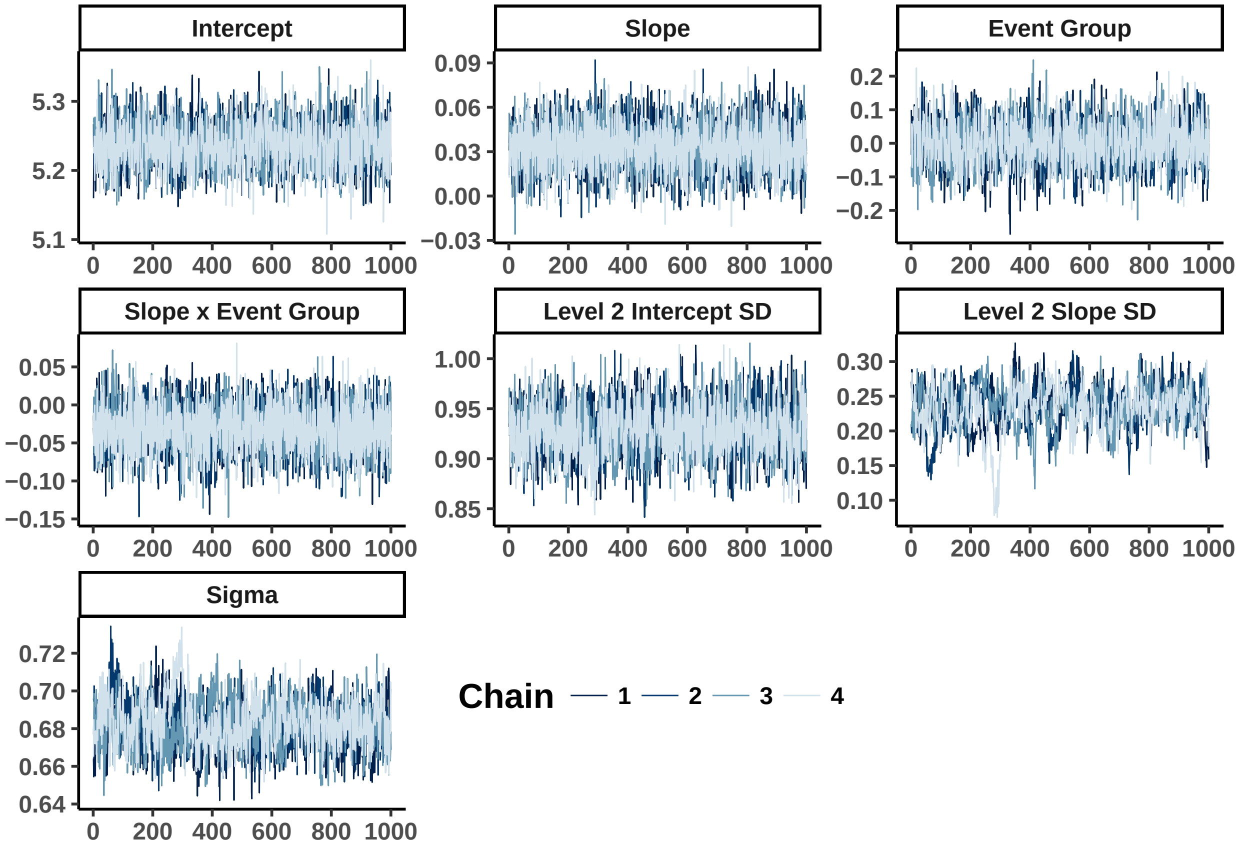


```
ggsave(sprintf("%s/results/plots/A_marriage_posterior_soc.png", data_path), width = 14, height = 8)
```

Sample Trace Plot

```
load("~/Box/Models/PCLE Replication/unpooled/E_LeftPar_chain1_brm_cv.RData")
post <- posterior_samples(fit, add_chain = T)
post.small <- post %>% select(-contains(",Intercept"), -contains(",new.wave")) %>% tbl_df
post.small <- post.small %>% select(-iter, -`lp__`, -contains("sex"), -contains("age"), -contains("cor")) %>%
  setNames(c("Intercept", "Slope", "Event Group", "Slope x Event Group", "Level 2 Intercept SD",
             "Level 2 Slope SD", "Sigma", "Chain"))

save(post.small, file = sprintf("%s/results/diagnostics.RData", data_path))
```

```
load(sprintf("%s/results/diagnostics.RData", data_path))
library(bayesplot)
mcmc_trace(post.small,
           size = .25) +
  theme_classic() +
  theme(legend.position = c(.5, .15),
        legend.direction = "horizontal") +
    theme(axis.text = element_text(face = "bold"),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          strip.text = element_text(face = "bold", size = rel(.8)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



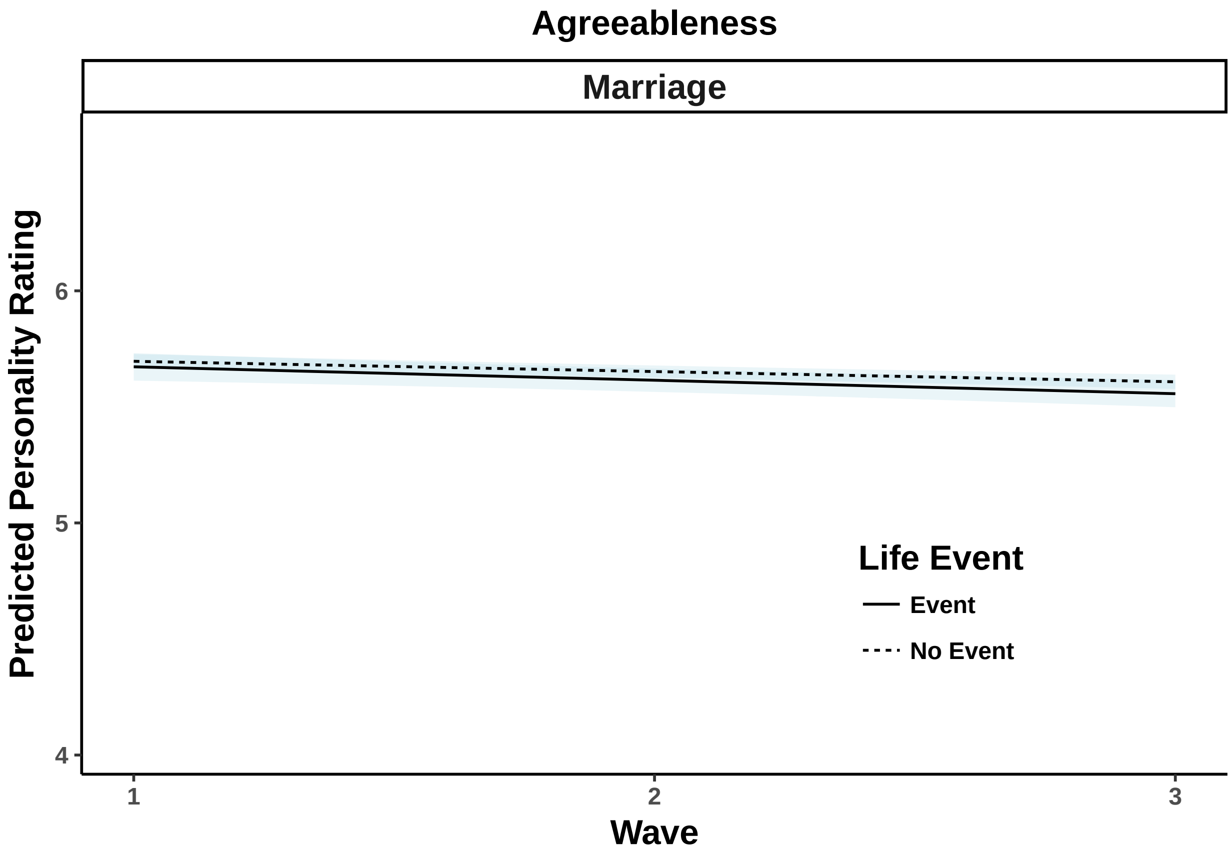
Mean Level Trajectories

```
big5 <- tibble(
  old = c("E", "A", "C", "N", "O"),
  new = c("Extraversion", "Agreeableness", "Conscientiousness", "Neuroticism", "Openness"),
  colors = c("royalblue", "orange", "lightgoldenrod", "springgreen3", "purple")
)
events <- tibble(
  old = c("none", "Married", "MoveIn", "Divorce", "SepPart", "PartDied", "LeftPar",
          "ChldMvOut", "ChldBrth", "ParDied", "Unemploy", "Retire", "FrstJob"),
  new = c("Mean", "Marriage", "Moved in with Partner", "Divorce", "Separation from Partner",
          "Death of Partner/Spouse", "Leaving Parental Home", "Child Leaves Home",
          "Birth of Child", "Death of Parent",   "Unemployment", "Retirement", "First Job"),
  breaks = c("Mean", "Marriage", "Moved in\nwith Partner", "Divorce", "Separation\nfrom Partner",
          "Death of\nPartner/Spouse", "Leaving\nParental Home", "Child Leaves\nHome",
          "Birth of\nChild", "Death of\nParent",   "Unemployment", "Retirement", "First Job")
)

# unnest and refactor fixed effect predictions
growth_pred <- bfi_growth %>%
  unnest(b.fixef.pred, .drop = T) %>%
  filter(!(Event %in% c("DadDied", "MomDied", "none"))) %>%
  mutate(le_value = mapvalues(le.group, unique(le.group), c("No Event", "Event")),
         shrt_Event = Event, shrt_Trait = Trait,
         Event = mapvalues(Event, events$old, events$breaks),
         Event = factor(Event, levels = events$breaks),
         Trait = mapvalues(Trait, big5$old, big5$new),
         Trait = factor(Trait, levels = big5$new))

# get "significance"
sig <- bfi_growth %>%
  unnest(b.fixef.tab) %>%
  filter(!(Event %in% c("DadDied", "MomDied", "none"))) %>%
  filter(term == "Slope:le.group") %>%
  mutate(sig = ifelse(sign(lower) == sign(upper), "sig", "ns")) %>%
  select(Event, Trait, sig, d)

# create a data frame that will help control faceted axes
range_act <- growth_pred %>%
  group_by(Trait, Event, shrt_Event, shrt_Trait) %>%
  summarize(min = min(Estimate) - .5, max = max(Estimate) + .5) %>%
  gather(key = est, value = Estimate, min, max) %>%
  full_join(crossing(new.wave = 1:3, shrt_Trait = c("E", "A", "C", "N", "O"))) %>%
  full_join(sig %>% select(Event, Trait, sig) %>%
    rename(shrt_Trait = Trait, shrt_Event = Event)) %>%
  select(-est) %>%
  unite(comb, Trait, Event, sep = ".", remove = F)

# save for later use in the app
save(growth_pred, range_act, file = sprintf("%s/results/growth_pred.RData", data_path))

traj_fun <- function(trait){
  color <- (big5 %>% filter(new == trait))$colors
  # base of plot
  p <- growth_pred %>% filter(Trait == trait) %>%
  ggplot(aes(x = new.wave + 1, y = Estimate)) +
    scale_x_continuous(limits = c(1,3), breaks = seq(1,3,1)) +
    scale_color_manual(values = c(color, "black"))
  # background highlights
  if(any((range_act %>% filter(Trait == trait))$sig == "sig")){
    p <- p +
    geom_rect(data = range_act %>% filter(Trait == trait & sig == "sig"), fill = "khaki1",
              alpha = .5, aes(xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = Inf))
  }
  # the rest of the plot
  p +
    geom_ribbon(aes(ymin = `2.5%ile`, ymax = `97.5%ile`, group = le_value), fill = "lightblue", alpha = .25) +
    geom_line(aes(color = factor(le_value), linetype = factor(le_value)), size = .5) +
    geom_blank(data = range_act %>% filter(Trait == trait)) +
    labs(x = "Wave", y = "Predicted Personality Rating",
         color = "Life Event", title = trait, linetype = "Life Event") +
    facet_wrap(~ Event, nrow = 3) +
    theme_classic() +
    theme(axis.text = element_text(face = "bold"),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.position = "bottom",
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          strip.text = element_text(face = "bold", size = rel(.7)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
  ggsave(sprintf("%s/results/plots/%s_trajectories.png", data_path, trait), width = 6, height = 5)
}

# refactor sig df
pl.sig <- sig %>%
  mutate(shrt_Event = Event, shrt_Trait = Trait,
         Event = mapvalues(Event, events$old, events$breaks),
         Event = factor(Event, levels = events$breaks),
         Trait = mapvalues(Trait, big5$old, big5$new),
         Trait = factor(Trait, levels = big5$new)) %>%
  unite(comb, Trait, Event, sep = ".", remove = F) %>%
  filter(sig == "sig")
```

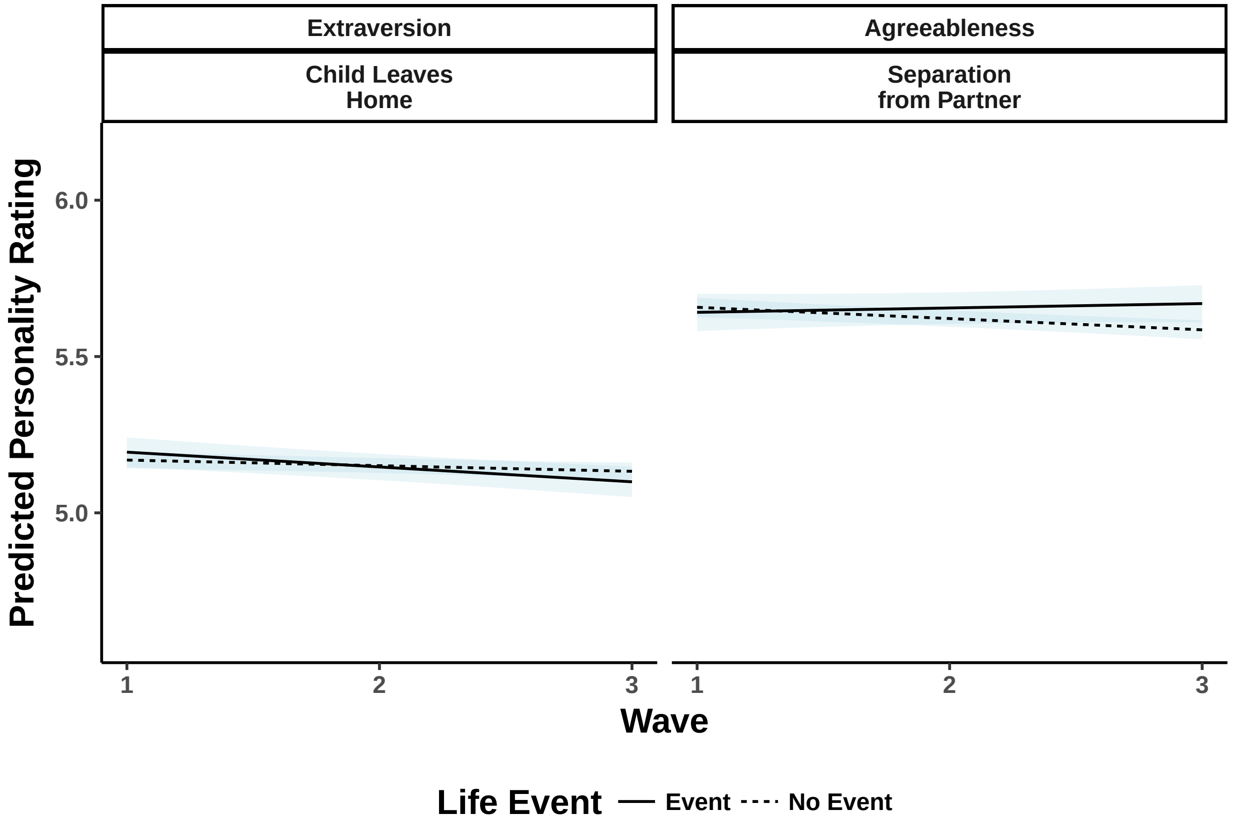
Sample Plots

```
# plot sample trajectory for Agreeabelness + Marriage
growth_pred %>%
  filter(shrt_Trait == "A" & shrt_Event == "Married") %>%
  ggplot(aes(x = new.wave + 1, y = Estimate)) +
```

```r
    scale_x_continuous(limits = c(1,3), breaks = seq(1,3,1)) +
    scale_color_manual(values = c("royalblue", "black")) +
    geom_ribbon(aes(ymin = `2.5%ile`, ymax = `97.5%ile`, group = le_value), fill = "lightblue", alpha = .25) +
    geom_line(aes(linetype = factor(le_value)), size = .5) +
    geom_blank(data = range_act %>% filter(shrt_Event == "Married") ) +
    labs(x = "Wave", y = "Predicted Personality Rating",
         color = "Life Event", linetype = "Life Event", title = "Agreeableness") +
    facet_wrap(~  Event) +
    theme_classic() +
    theme(axis.text = element_text(face = "bold"),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.position = c(.75,.25),
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          strip.text = element_text(face = "bold", size = rel(1.2)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



```r
ggsave(sprintf("%s/results/plots/A_Marriage_traj.png",data_path), width = 4, height = 4)


# plot only the "significant" trajectories
growth_pred %>% unite(comb, Trait, Event, sep = ".", remove = F) %>%
  filter(comb %in% pl.sig$comb) %>%
  mutate(Trait = factor(Trait, levels = as.character(unique(pl.sig$Trait))),
         Event = factor(Event, levels = as.character(unique(pl.sig$Event)))) %>%
  ggplot(aes(x = new.wave + 1, y = Estimate)) +
    scale_x_continuous(limits = c(1,3), breaks = seq(1,3,1)) +
    scale_color_manual(values = c("royalblue", "black")) +
    geom_ribbon(aes(ymin = `2.5%ile`, ymax = `97.5%ile`, group = le_value), fill = "lightblue", alpha = .25) +
    geom_line(aes(linetype = factor(le_value)), size = .5) +
    geom_blank(data = range_act %>% filter(comb %in% pl.sig$comb) ) +
    labs(x = "Wave", y = "Predicted Personality Rating",
         color = "Life Event", linetype = "Life Event") +
    facet_wrap(~ Trait + Event) +
    theme_classic() +
    theme(axis.text = element_text(face = "bold"),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.position = "bottom",
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          strip.text = element_text(face = "bold", size = rel(.8)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



```
ggsave(sprintf("%s/results/plots/sig_traj.png",data_path), width = 6, height = 5)
```

```
lapply(big5$new, function(x){
  cat('####', x, '\n\n  ')
  traj_fun(x)
})
```

Extraversion

#### Agreeableness

#### Conscientiousness

#### Neuroticism

#### Openness

[[1]] NULL

[[2]] NULL

[[3]] NULL

[[4]] NULL

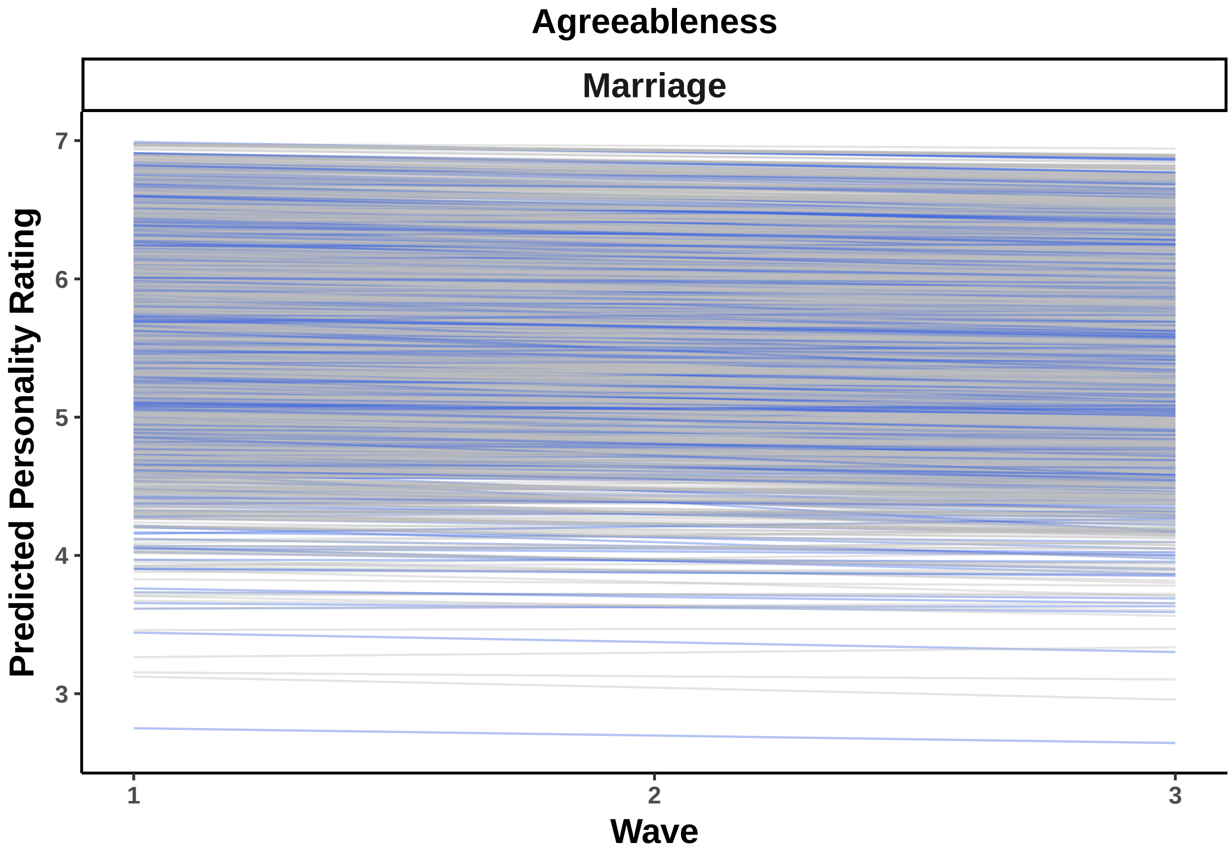
[[5]] NULL

Individual Level Trajectories

```r
ind_plots <- bfi_growth %>%
  unnest(b.ranef.pred) %>%
  filter(!(Event %in% c("DadDied", "MomDied", "none"))) %>%
  mutate(le.group = as.numeric(as.character(le.group)),
    le.group = mapvalues(le.group, unique(le.group), c("No Event", "Event")),
    Trait = mapvalues(Trait, big5$old, big5$new),
    Trait = factor(Trait, levels = big5$new),
    Event = mapvalues(Event, events$old, events$breaks),
    Event = factor(Event, levels = events$breaks))

ind_plot_fun <- function(trait){
  sample_fun <- function(df){
  subs1 <- sample(unique((df %>% filter(le.group == "No Event"))$PROC_SID), 50)
  subs2 <- sample(unique((df %>% filter(le.group == "Event"))$PROC_SID), 50)
  df <- df %>% filter(PROC_SID %in% c(subs1, subs2))
  }
  color <- (big5 %>% filter(new == trait))$colors
  df <- ind_plots %>% filter(Trait == trait) %>%
    group_by(Event) %>% nest() %>%
    mutate(data = map(data, sample_fun)) %>%
    unnest(data) %>%
    full_join(range_act %>% select(Trait, Event, sig) %>% filter(Trait == trait))
  p <- df %>% ggplot(aes(x = new.wave+1, y = Estimate, color = le.group)) +
    scale_x_continuous(limits = c(1,3), breaks = seq(1,3,1)) +
    scale_y_continuous(limits = c(1,7), breaks = seq(1,7,3)) +
    scale_color_manual(values = c(color, "gray80"))
  if(any((range_act %>% filter(Trait == trait))$sig == "sig")){
    p <- p +
    geom_rect(data = . %>% filter(sig == "sig"), fill = "khaki1", alpha = .5,
    aes(xmin = -Inf, xmax = Inf, ymin = -Inf, ymax = Inf), color = "khaki1")
  }
  p +
    geom_line(aes(group = PROC_SID), size = .25) +
    facet_wrap(~ Event, nrow = 3) +
    labs(x = "Wave", y = "Predicted Personality Rating",
         color = "Life Event", title = trait) +
    theme_classic() +
    theme(axis.text = element_text(face = "bold"),
          strip.text = element_text(face = "bold", size = rel(.7)),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.position = "bottom",
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
   ggsave(sprintf("%s/results/plots/%s_ranef_trajectories.png", data_path, trait), width = 6, height = 5)
}
```

Sample Plot

```r
# color <- (big5 %>% filter(new == "Agreeableness"))$colors
  ind_plots %>% filter(Trait == "Agreeableness" & Event == "Marriage") %>%
  ggplot(aes(x = new.wave+1, y = Estimate, color = le.group)) +
    scale_x_continuous(limits = c(1,3), breaks = seq(1,3,1)) +
    scale_color_manual(values = c("royalblue", "gray")) +
    geom_line(aes(group = PROC_SID), size = .4, alpha = .4) +
    geom_line(data = growth_pred %>% filter(Trait == "Agreeableness" & Event == "None"),
              aes(group = le.group), color = "royalblue", size = 2)+
    facet_wrap(~ Event, nrow = 2) +
    labs(x = "Wave", y = "Predicted Personality Rating",
         color = "Health Event", title = "Agreeableness") +
    theme_classic() +
    theme(axis.text = element_text(face = "bold"),
          strip.text = element_text(face = "bold", size = rel(1.2)),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.position = "none",
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



```
  ggsave(sprintf("%s/results/plots/Agreeableness_Marriage.png", data_path), width = 4, height = 4)
```

```
lapply(big5$new, function(x){
  cat('####', x, '\n\n  ')
  ind_plot_fun(x)
})
```

Extraversion

#### Agreeableness

#### Conscientiousness

#### Neuroticism

#### Openness

[[1]] NULL

[[2]] NULL

[[3]] NULL

[[4]] NULL

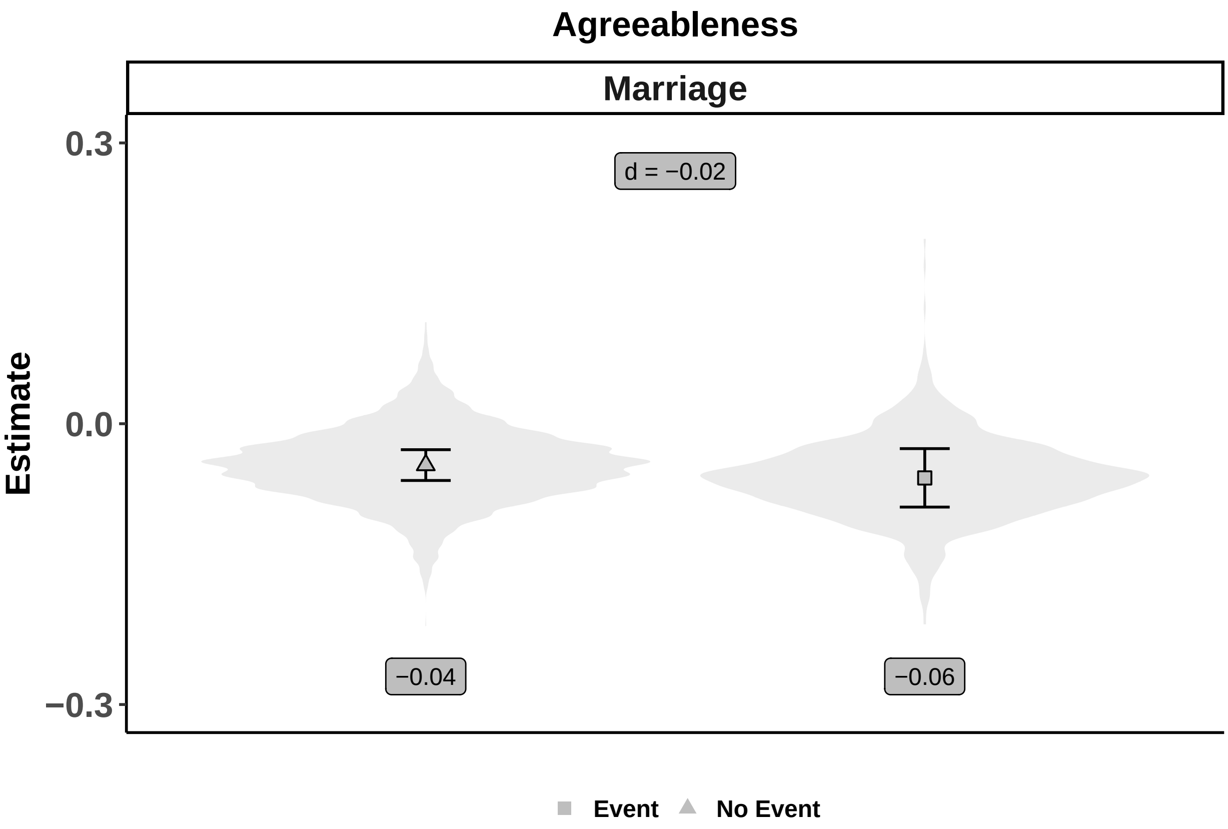
[[5]] NULL

Group Slopes

```r
slopes <- bfi_growth %>%
  unnest(b.hyp) %>% filter(term.type == "Slope") %>%
  full_join(sig) %>%
  full_join(bfi_growth %>% unnest(b.fixef.tab) %>% filter(term == "Slope:le.group") %>%
              select(Event, Trait, d)) %>%
  filter(!(Event %in% c("DadDied", "MomDied", "none"))) %>%
  mutate(shrt_Event = Event, shrt_Trait = Trait,
         Trait = mapvalues(Trait, big5$old, big5$new),
        Trait = factor(Trait, levels = big5$new),
        Event = mapvalues(Event, events$old, events$breaks),
        Event = factor(Event, levels = events$breaks),
        sig = ifelse(is.na(sig) == T, "sig", sig),
        term2 = as.numeric(mapvalues(term, unique(term), c(1,0))))

ranef_slopes <- bfi_growth %>% unnest(b.ranef.tab) %>%
  filter(term != "Intercept") %>%
  mutate(term = ifelse(term == "new.wave", "Slope", term)) %>%
  select(Event:Trait, PROC_SID, estimate) %>%
  filter(!(Event %in% c("DadDied", "MomDied"))) %>%
  # mutate(PROC_SID = as.numeric(PROC_SID)) %>%
  left_join(le_dat %>% select(Event, PROC_SID, le.group)) %>%
  full_join(sig) %>%
  dplyr::rename(term = le.group) %>%
  mutate(term = mapvalues(term, c(0,1), c("No Event", "Event"))) %>%
  mutate(shrt_Event = Event, shrt_Trait = Trait,
         Trait = mapvalues(Trait, big5$old, big5$new),
        Trait = factor(Trait, levels = big5$new),
        Event = mapvalues(Event, events$old, events$breaks),
        Event = factor(Event, levels = events$breaks))  %>%
  left_join(slopes %>% select(Event, Trait, term, b)) %>%
  mutate(b = b + estimate) %>%
  select(Event, shrt_Event, Trait, shrt_Trait, b, PROC_SID, term, sig) %>%
  filter(Event != "Mean") %>%
  mutate(term2 = as.numeric(mapvalues(term, unique(term), c(1,0))))

rects <- expand.grid(
  term = c("Event", "No Event"),
  xstart = 0.5,
  xend = 1.5,
  b = NA_real_,
  Event = unique(ranef_slopes$shrt_Event),
  Estimate = 1, stringsAsFactors = F) %>% tbl_df %>%
  full_join(sig) %>%
  mutate(term2 = as.numeric(mapvalues(term, unique(term), c(1,0))),
         shrt_Event = Event, shrt_Trait = Trait,
         Trait = mapvalues(Trait, big5$old, big5$new),
        Trait = factor(Trait, levels = big5$new),
        Event = mapvalues(Event, events$old, events$breaks),
        Event = factor(Event, levels = events$breaks))

save(pl.sig, growth_pred, range_act, slopes, ranef_slopes, rects,
     file = sprintf("%s/results/plot_files.RData", data_path))

slope_plot_fun <- function(trait){
  color = "gray"
  p <- slopes %>% filter(Trait == trait) %>% filter(!is.na(Event)) %>%
  ggplot(aes(x = term2, y = b)) +
    scale_y_continuous(limits = c(-.3,.3), breaks = seq(-.3,3,.3)) +
    geom_violin(data = ranef_slopes %>% filter(Trait == trait & sig == "ns" & term == "No Event"),
                aes(x = 0), color = NA, fill = color, alpha = .3) +
    geom_violin(data = ranef_slopes %>% filter(Trait == trait & sig == "ns" & term == "Event"),
                aes(x = 1), color = NA, fill = color, alpha = .3) +
    scale_shape_manual(values = c(15, 17)) +
    geom_errorbar(data = . %>% filter(sig == "ns"), aes(ymin = lower, ymax = upper), width = .1) +
    geom_point(data = slopes %>% filter(Trait == trait & sig == "ns" ),
               aes(shape = term), color = color, size = 2) + #shape = 15,
    geom_point(data = slopes %>% filter(Trait == trait & sig == "ns" & term == "No Event"),
               aes(shape = term), color = "black", size = 2, shape = 2) +
    geom_point(data = slopes %>% filter(Trait == trait & sig == "ns" & term == "Event"),
               aes(shape = term), color = "black", size = 2, shape = 0) +
    geom_label(data = slopes %>% filter(Trait == trait & sig == "ns"),
               aes(y = -.27, label = ifelse(abs(b) < .001, round(b, 4),
               ifelse(abs(b) < .01, round(b,3), round(b,2)))),
               fill = color, color = "black", size = 3) +
    geom_label(data = slopes %>% filter(Trait == trait & sig == "ns" & term == "Event"),
               aes(y = .27, label = paste("d =", ifelse(abs(d) < .001, round(d, 4),
               ifelse(abs(d) < .01, round(d,3), round(d,2))), sep = " ")),
               fill = color, color = "black", size = 3, nudge_x = -.5) +
    labs(x = NULL, y = "Estimate", title = trait, shape = NULL) +
    facet_wrap(~Event, nrow = 2) +
    theme_classic() +
    theme(legend.position = "bottom",
          axis.text = element_text(face = "bold", size = rel(1.2)),
          axis.text.x = element_blank(),#element_text(face = "bold", size = rel(1.2), angle = 45, hjust = 1),
          axis.ticks.x = element_blank(),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          strip.text = element_text(face = "bold", size = rel(.8)),
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
  if(any((slopes %>% filter(Trait == trait))$sig == "sig")){
    p+
    geom_rect(data = rects %>% filter(sig == "sig" & Trait == trait), fill = "khaki1", alpha = .5,
              aes(xmin = -.5, xmax = 1.5, ymin = -Inf, ymax = Inf)) +
      geom_violin(data = ranef_slopes %>% filter(Trait == trait & sig == "sig" & term == "No Event"), aes(x = 0),
                fill = "royalblue", color = NA, alpha = .3) +
      geom_violin(data = ranef_slopes %>% filter(Trait == trait & sig == "sig" & term == "Event"), aes(x = 1),
                fill = "royalblue", color = NA, alpha = .3) +
    geom_errorbar(data = . %>% filter(sig == "sig"), aes(ymin = lower, ymax = upper), width = .1) +
    geom_point(data = slopes %>% filter(Trait == trait & sig == "sig"),
               aes(shape = term), color = "royalblue", size = 2) + #, shape = 15
    geom_point(data = slopes %>% filter(Trait == trait & sig == "sig" & term == "No Event"),
               aes(shape = term), color = "black", size = 2, shape = 2) +
    geom_point(data = slopes %>% filter(Trait == trait & sig == "sig" & term == "Event"),
               aes(shape = term), color = "black", size = 2, shape = 0) +
    geom_label(data = slopes %>% filter(Trait == trait & sig == "sig"),
               aes(y = -.27, label = ifelse(abs(b) < .001, round(b, 4),
               ifelse(abs(b) < .01, round(b,3), round(b,2)))),
               fill = "royalblue", color = "white", size = 3) +
    geom_label(data = slopes %>% filter(Trait == trait & sig == "sig" & term == "Event"),
               aes(y = .27, label = paste("d =", ifelse(abs(b) < .001, round(b, 4),
               ifelse(abs(b) < .01, round(b,3), round(b,2))), sep = " ")),
               fill = "royalblue", color = "white", size = 3, nudge_x = -.5)
  }
  ggsave(sprintf("%s/results/plots/%s_slopes.png", data_path, trait), width = 8, height = 6)
}
```

Sample Plots

```
trait = "Agreeableness"
event = "Marriage"
color = "gray"
slopes %>% filter(shrt_Trait == "A" & shrt_Event == "Married") %>%
  ggplot(aes(x = term2, y = b)) +
  scale_y_continuous(limits = c(-.3,.3), breaks = seq(-.3,3,.3)) +
  scale_x_continuous(limits = c(-.5,1.5)) +
    geom_violin(data = ranef_slopes %>% filter(Trait == trait & sig == "ns" & term == "No Event" & Event == event),
                aes(x = 0), color = NA, fill = color, alpha = .3) +
    geom_violin(data = ranef_slopes %>% filter(Trait == trait & sig == "ns" & term == "Event" & Event == event),
    aes(x = 1), color = NA, fill = color, alpha = .3) +
    scale_shape_manual(values = c(15, 17)) +
    geom_errorbar(data = . %>% filter(sig == "ns" & Event == event), aes(ymin = lower, ymax = upper), width = .1) +
    geom_point(data = slopes %>% filter(Trait == trait & sig == "ns"  & Event == event),
               aes(shape = term), color = color, size = 2) + #shape = 15,
    geom_point(data = slopes %>% filter(Trait == trait & sig == "ns" & term == "No Event" & Event == event),
               aes(shape = term), color = "black", size = 2, shape = 2) +
    geom_point(data = slopes %>% filter(Trait == trait & sig == "ns" & term == "Event" & Event == event),
               aes(shape = term), color = "black", size = 2, shape = 0) +
    geom_label(data = slopes %>% filter(Trait == trait & sig == "ns" & Event == event),
               aes(y = -.27, label = ifelse(abs(b) < .001, round(b, 4),
               ifelse(abs(b) < .01, round(b,3), round(b,2)))),
               fill = color, color = "black", size = 3) +
    geom_label(data = slopes %>% filter(Trait == trait & sig == "ns" & term == "Event" & Event == event),
               aes(y = .27, label = paste("d =", ifelse(abs(d) < .001, round(d, 4),
               ifelse(abs(d) < .01, round(d,3), round(d,2))), sep = " ")),
               fill = color, color = "black", size = 3, nudge_x = -.5) +
    labs(x = NULL, y = "Estimate", title = trait, shape = NULL) +
    facet_wrap(~Event, nrow = 2) +
    theme_classic() +
    theme(legend.position = "bottom",
          axis.text = element_text(face = "bold", size = rel(1.2)),
          axis.text.x = element_blank(),#element_text(face = "bold", size = rel(1.2), angle = 45, hjust = 1),
          axis.ticks.x = element_blank(),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          strip.text = element_text(face = "bold", size = rel(1.2)),
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold", size = rel(1.2)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))
```



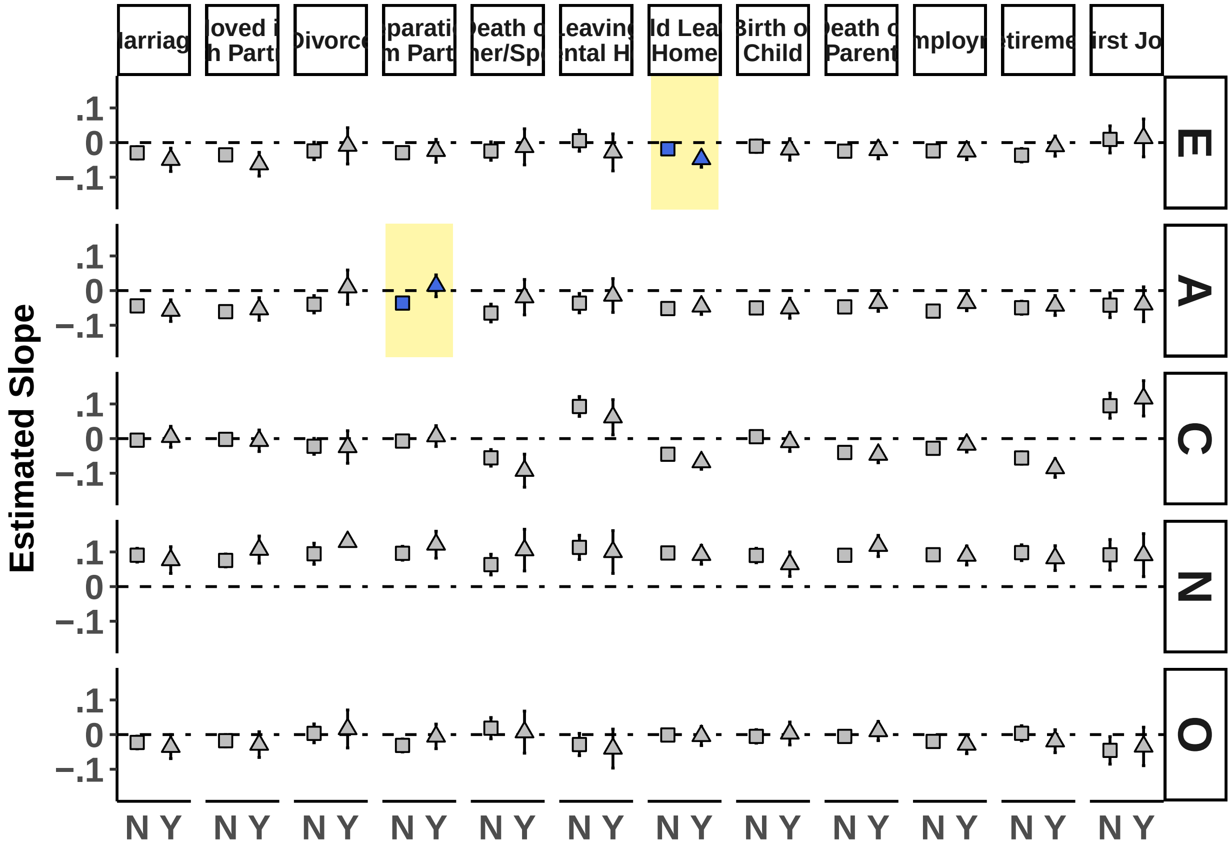
```
  ggsave(sprintf("%s/results/plots/Agreeableness_Marriage_slopes.png", data_path), width = 4, height = 4)
```

```
lapply(big5$new, function(x){
  cat('####', x, '\n\n  ')
  slope_plot_fun(x)
})
```

Extraversion

```
#### Agreeableness

#### Conscientiousness

#### Neuroticism

#### Openness

[[1]] NULL

[[2]] NULL

[[3]] NULL

[[4]] NULL

[[5]] NULL
```

```r
sig <- bfi_growth %>% unnest(b.fixef.tab) %>%
  filter(!(Event %in% c("MomDied", "DadDied"))) %>%
  filter(term == "Slope:le.group") %>%
  group_by(Event, Trait) %>%
  summarize(sig = ifelse(sign(lower) == sign(upper), "sig", "ns")) %>%
  ungroup()

d <- bfi_growth %>% unnest(b.fixef.tab) %>% filter(term == "Slope:le.group") %>% select(Event, Trait, d)

soc.tab <- bfi_growth %>%
  filter(!(Event %in% c("MomDied", "DadDied"))) %>%
  unnest(b.hyp, .drop = T) %>%
  full_join(sig) %>%
  full_join(d) %>%
  filter(term.type == "Slope") %>%
  mutate(term2 = as.numeric(mapvalues(term, unique(term), c(1,0))),
            sig = ifelse(is.na(sig) == T, "ns", sig),
         # Trait = mapvalues(Trait, big5$old, big5$new),
        Trait = factor(Trait, levels = big5$old),
        Event = mapvalues(Event, events$old, events$breaks),
        Event = factor(Event, levels = events$breaks)) %>%
  arrange(Event)

rects <- expand.grid(
  Trait = unique(bfi_growth$Trait),
  term = c("Event", "No Event"),
  xstart = 0.5,
  xend = 1.5,
  b = NA_real_,
  Event = unique(bfi_growth$Event),
  Estimate = 1, stringsAsFactors = F) %>% tbl_df %>%
  full_join(sig)  %>%
  filter(!(Event %in% c("MomDied", "DadDied", "none"))) %>%
  mutate(term2 = as.numeric(mapvalues(term, unique(term), c(1,0))),
        sig = ifelse(is.na(sig) == T, "ns", sig),
        Trait = factor(Trait, levels = big5$old),
        Event = mapvalues(Event, events$old, events$breaks),
        Event = factor(Event, levels = events$breaks))

soc.tab %>% filter(Event != "Mean") %>%
  ggplot(aes(x = term2, y = b)) +
    scale_color_manual(values = c("grey", "royalblue")) +
    scale_fill_manual(values = c("white", "khaki1")) +
    scale_x_continuous(limits = c(-.5,1.5), breaks = c(0,1), labels = c("N", "Y")) +
    scale_y_continuous(limits = c(-.175,.175), breaks = c(-.1,0,.1), labels = c("-.1", "0", ".1")) +
    geom_rect(data = rects,
      aes(xmin = -.5, xmax = 1.5, ymin = -Inf, ymax = Inf, fill = sig),
      alpha = .5) +
    geom_hline(aes(yintercept = 0), linetype = "dashed") +
    geom_errorbar(aes(ymin = lower, ymax = upper), width = .1) +
    geom_point(data = . %>% filter(term2 == 0), aes(color = sig),
               shape = 15, size = 2) +
    geom_point(data = . %>% filter(term2 == 0), color = "black",
               shape = 0, size = 2) +
    geom_point(data = . %>% filter(term2 == 1), aes(color = sig),
               shape = 17, size = 2) +
    geom_point(data = . %>% filter(term2 == 1), color = "black",
               shape = 2, size = 2) +
  labs(x = NULL, y = "Estimated Slope") +
    facet_grid(Trait~Event) +
    theme_classic() +
    theme(#axis.text.x = element_blank(),
          axis.ticks.x = element_blank(),
          strip.text.y = element_text(face = "bold", size = rel(2)),
          strip.text.x = element_text(face = "bold"),
          axis.text = element_text(face = "bold", size = rel(1.2)),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.position = "none")
```



```
ggsave(sprintf("%s/results/plots/slopes_all.png", data_path), width = 14, height = 7)
```