Web Appendix 3: Case Study — Inverse Probability Weights for Quasi-Continuous Ordinal Exposures with a Binary Outcome: Method Comparison and Case Study

Sack, Daniel E, Shepherd, Bryan E, Audet, Carolyn M, De Schact, Caroline, Samuels, Lauren R

Updated 2023-01-21 for American Journal of Epidemiology

Setup

```
# full code available at https://github.com/dannysack/gen_prop_wts

source('./orm.wt.R') # loads tidyverse, rms, and mice libraries
```

```
library(WeightIt)
library(MatchThem)
```

```
library(cobalt)
```

```
library(lme4)
```

```
library(kableExtra)
```

```
library(ggpubr)
library(parallel)
library(foreach)
```

```
library(doParallel)
```

```
library(extrafont)
```

```
library(ggdist)
library(table1)
```

```
library(gridExtra)
```

```
library(gtable)
library(grid)
library(broom.mixed)
```

```
library(forestplot)
```

```
library(DiagrammeR)
library(DiagrammeRsvg)
library(rsvg)
```

```
library(ggeffects)
```

```
library(future)
```

```
library(furrr)
loadfonts()
```

```
knitrSet(lang='markdown', fig.path='png2_pdf/', fig.align='center', w=9, h=8, cache=TRUE)
options(knitr.kable.NA = '')

# set seed across all all streams for parallel processing
set.seed(1111, kind = "L'Ecuyer-CMRG")

# set theme for plots
theme <- theme_pubr() +
  theme(legend.position = "bottom",
                legend.key = element_rect(fill = NA, color = NA),
                text = element_text(family = "Arial"),
                axis.line = element_line(),
                axis.text = element_text(family = "Arial", color = "black"),
                strip.background = element_rect(fill = "grey95"))
```

Load Data

```
# start by loading hops+ data
# load cleaned data
load("/Users/sackd/Library/CloudStorage/Box-Box/Vanderbilt University/PhD/Dissertation/Thesis/Analysis/

# start by removing variables not relevant to the imputations or analyses
data <- quant %>%
  select(district.female, age.female, day.female, group.female, clinic.female,
         rel_stat.female, edu_cat.female, job_cat.female,
         stigma_com.female:peer_prop.female, who_stage, who_stage.male,
         district.male, age.male, day.male, group.male, rel_stat.male, edu_cat.male,
         job_cat.male, stigma_com.male:peer_prop.male, fem_int_tot:m_prop_int,
         time_to_fp, FP_method_final, Y_lc:covid_sens) %>%
  mutate(clinic.female = as.numeric(clinic.female)) %>% # required for imputations
  arrange(clinic.female) %>%
  mutate(who_stage = factor(who_stage, levels = c("I", "II", "III", "IV")),
         who_stage.male = factor(who_stage.male, levels = c("I", "II", "III", "IV"))) %>%
  filter(group.female == "Intervention") %>% # only want folks in the intervention group
  filter(!is.na(Y_raw)) %>% # only want folks with complete outcome data
  select(district.female:phq9.female, skills_comp.female, peer_comp.female,
         who_stage:phq9.male, skills_comp.male, peer_comp.male,
         FP_method_final, Y_raw) %>% # remove excess columns
  mutate(exp.female = skills_comp.female + peer_comp.female,
         exp.male = skills_comp.male + peer_comp.male)
```

Describe Data

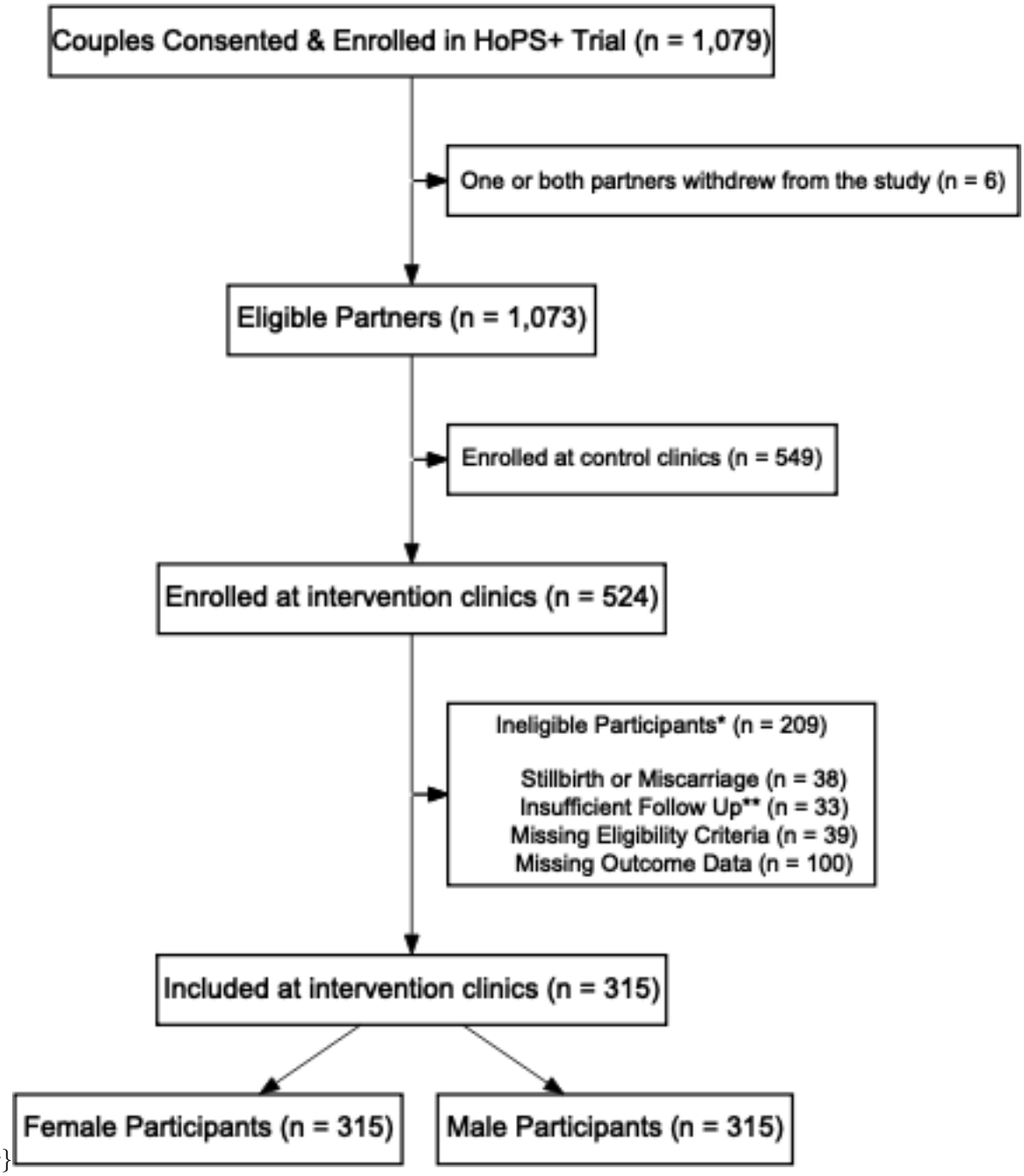
Given that the intervention consisted of six couples counseling and skills sessions focused on building communication skills, nine peer support sessions (with peer couples) focused on navigating pregnancy and

the postpartum period, and joint couples antenatal and postnatal HIV care, we were interested whether intervention adherence (among female and male participants separately) impacted a secondary outcome (postpartum contraceptive uptake) among female participants in the intervention arm (1).

Web Figure 7 - Enrollment Flow Chart

```
# create study flow chart

# set font
par(family = "Arial")

flow <- grViz("digraph flowchart {
      # main nodes
      node [fontname = Arial, shape = rectangle, penwidth = 1.5, fontsize = 14]
      consented [label = 'Couples Consented & Enrolled in HoPS+ Trial (n = 1,079)']
      enrolled [label = 'Eligible Partners (n = 1,073)']
      intclin [label = 'Enrolled at intervention clinics (n = 524)']
      finint [label = 'Included at intervention clinics (n = 315)']
      females [label= 'Female Participants (n = 315)']
      males [label = 'Male Participants (n = 315)']

      # exclusion nodes (smaller fontsize)
      node [fontname = Arial, shape = rectangle, penwidth = 1.5, fontsize = 12]
      excall [label = 'One or both partners withdrew from the study (n = 6)']
      exccont [label = 'Enrolled at control clinics (n = 549)']
      excint [label = 'Ineligible Participants* (n = 209)\\n
        Stillbirth or Miscarriage (n = 38)
        Insufficient Follow Up** (n = 33)
        Missing Eligibility Criteria (n = 39)
        Missing Outcome Data (n = 100)']

      # dummy nodes
      node [shape=none, width=0, height=0, label='']
      n1
      n2
      n3

      # edge definitions with the node IDs
      n1 -> excall
      n2 -> exccont
      n1 -> enrolled
      n2 -> intclin
      n3 -> excint
      n3 -> finint
      finint -> females
      finint -> males
      {rank = same; n1 -> excall}
      {rank = same; n2 -> exccont}
      {rank = same; n3 -> excint}
      {rank = same; females; males}

      edge[dir = none]
      consented -> n1
```

```
      enrolled -> n2
      intclin -> n3
      }
      ")

flow
```



\begin{center}

```
# export
# flow %>%
#   export_svg() %>%
#   charToRaw() %>%
#   rsvg_pdf("./png2_pdf/supfig3_1.pdf")
```

```
# # embed fonts
# embedFonts("./png2_pdf/supfig3_1.pdf", fontpaths = "/Library/Fonts/Microsoft/Arial.ttf")

# * indicates that individuals could share more than one reason
# **Delivery less than 1 year before Nov. 30, 2021
```

*indicates that individuals could share more than one reason

**Delivery less than 1 year before Nov. 30, 2021

Web Figure 8 - Directed Acyclic Graph

To calculate each type of sIPW, we adjusted for the available minimally sufficient set of available confounding variables. These included: participant depression (continuous, measured via the Patient Health Questionaire-9)(2), female and male education (no education, some primary, completed primary, some secondary, completed secondary, college/higher education), participant WHO clinical stage (ordinal), participant social support (continuous, measured via adapted Berlin Social Support Scales)(3), partner HoPS+ engagement (continuous), participant age (continuous), and participant enrollment date (continuous). As this is merely a proof-of-concept analysis, we recognize that we are likely missing other potentially important confounders such as participant fertility intentions (which is part of the minimally sufficient set of theorized covariates), which were not available at the time of this analysis. The directed acyclic graph is shown below.

```
knitr::include_graphics("dag.pdf")
```

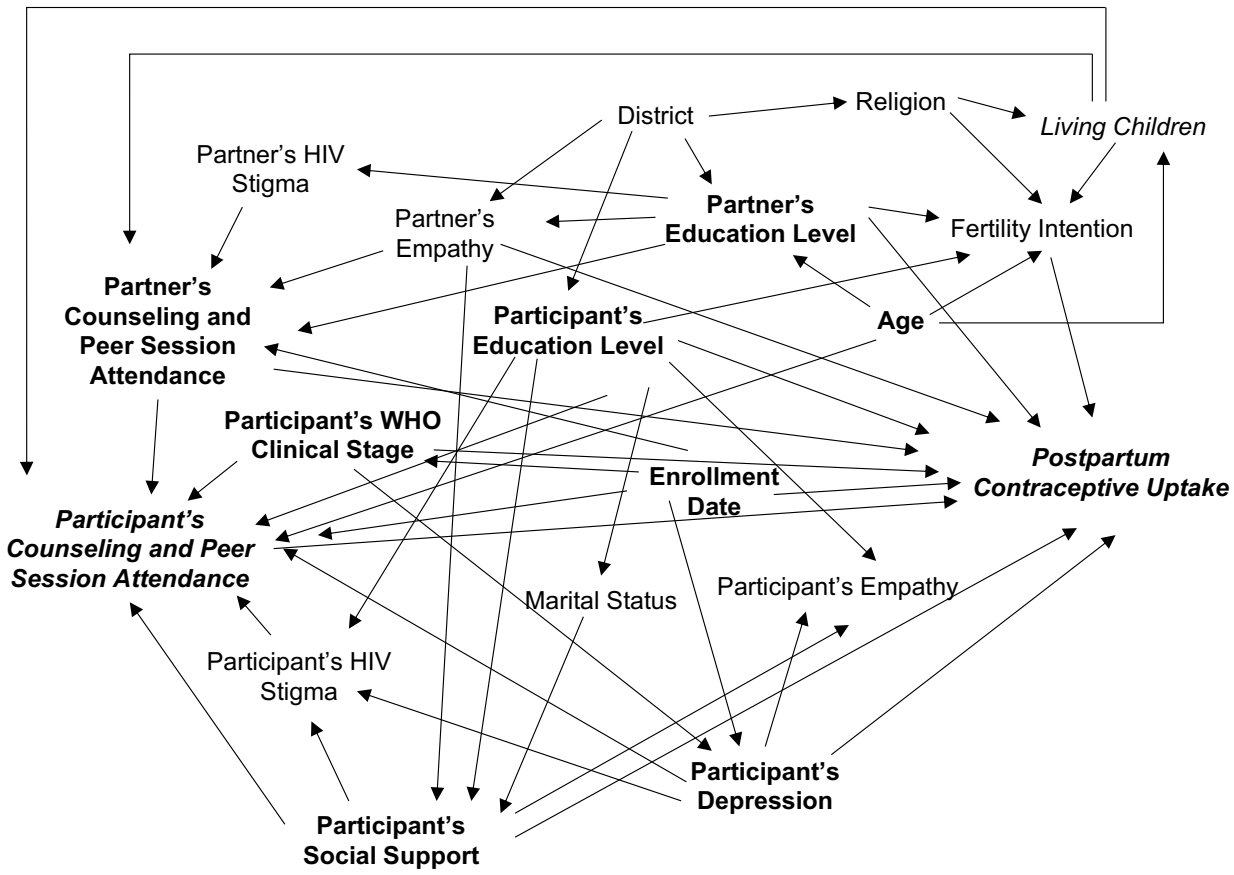


Table 4 - HoPS+ Descriptive Data

```
# code to generate table 1

# first split and combine female datasets
fem <- data %>% select(ends_with(".female"), who_stage, FP_method_final, Y_raw) %>%
  mutate(sex = "Female")
male <- data %>% select(ends_with(".male")) %>% mutate(sex = "Male")

# remove suffix
names(fem) <- gsub(".female", "", names(fem))
names(male) <- gsub(".male", "", names(male))

# now rowbind
tab1 <- bind_rows(fem, male) %>%
  mutate(sex = factor(sex, levels = c("Female", "Male")),
         skills_fac = factor(skills_comp),
         peer_fac = factor(peer_comp),
         exp_fac = factor(exp),
         fp_use = factor(Y_raw, labels = c("No", "Yes")))

# add some labels
label(tab1$exp) <- "Total Sessions Attended"
```

```
label(tab1$district) <- "District"
label(tab1$rel_stat) <- "Relationship Status"
label(tab1$edu_cat) <- "Education"
label(tab1$job_cat) <- "Occupation"
label(tab1$who_stage) <- "World Health Organization HIV Stage"
label(tab1$skills_fac) <- ""
label(tab1$peer_fac) <- ""
label(tab1$exp_fac) <- ""
label(tab1$fp_use) <- "Postpartum Contraceptive Use"
label(tab1$FP_method_final) <- "Method Type"

# table 1
table1(~ age + district + rel_stat +
         edu_cat + job_cat +
         soc_sup_ps + soc_sup_ns +
         phq9 + who_stage +
         skills_comp + skills_fac +
         peer_comp +peer_fac +
         exp + exp_fac +  fp_use +
         FP_method_final | sex, data = tab1,
       render.continuous = c(.="Median [Q1, Q3]"), overall = FALSE)

# final data (do not need method type in imputation since all missing for males)
data <- data %>% select(-FP_method_final)
```

Table 1 describes baseline covariates among female and male HoPS+ participants. Female participants were younger (median age 23 years, interquartile range [IQR] 19.5, 27 years) than males (median age 27 years, IQR 24, 32 years) and less likely to have received any education (15.9% versus 6.0%). Female participants were far more likely to work as a domestic worker (50.2% versus 16.2% of male participants) and less likely to work as a trader or fisherman (0.6% and 0% versus 13.7% and 19% among males respectively). Psychometric scale scores were similar in female and male participants.

Figure 3 - HoPS+ Exposure Distribution

```
# code to generate figure 4

fem_exp <- ggplot(data, aes(x = exp.female)) +
  geom_histogram(aes(y = ..count..),binwidth = 1, alpha = 0.5, color = "grey50") +
  # geom_density(adjust = 2) +
  scale_y_continuous(name = "Participants", breaks = c(0, 25, 50, 75)) +
  scale_x_continuous(name = "Female Session Attendance", limits = c(-1.25, 15)) +
  coord_cartesian(clip = "off", xlim = c(0, 15), ylim = c(0, 90)) +
  annotate(geom = "text", x = -1.25, y = 99, label = "A)", size = 10 / .pt) +
  theme +
  theme(text = element_text(size = 10),
        plot.margin = unit(c(15, 6, 6, 6), "pt"))

m_exp <- ggplot(data, aes(x = exp.male)) +
  geom_histogram(aes(y = ..count..), binwidth = 1, alpha = 0.5, color = "grey50") +
  # geom_density(adjust = 2) +
  scale_y_continuous(name = "Participants", breaks = c(0, 25, 50, 75)) +
  scale_x_continuous(name = "Male Session Attendance", limits = c(-1.25, 15)) +
```

```
  coord_cartesian(clip = "off", xlim = c(0, 15), ylim = c(0, 90)) +
  annotate(geom = "text", x = -1.25, y = 99, label = "B)", size = 10 / .pt) +
  theme +
  theme(text = element_text(size = 10),
        plot.margin = unit(c(15, 6, 6, 6), "pt"))

ggarrange(fem_exp, m_exp)

# save plot
ggsave("fin_figs/fig3.pdf", width = 5, height = 3)
# embed the font
embed_fonts("fin_figs/fig3.pdf")
```

Figure 3 Panels

```
# Panel A
fem_exp
ggsave("fin_figs/fig3a.pdf", width = 2.5, height = 3)
embed_fonts("fin_figs/fig3a.pdf")

# Panel A
m_exp
ggsave("fin_figs/fig3b.pdf", width = 2.5, height = 3)
embed_fonts("fin_figs/fig3b.pdf")
```

Imputations

```
# create predictor matrix
predmat <- make.predictorMatrix(data)

# now edit predictor matrix, from "mice" documentation
# "Each row corresponds to a variable block, i.e., a set of variables to be imputed.
# A value of 1 means that the column variable is used as a predictor for
  # the target block (in the rows)."
# my understanding is that the column then denotes whether that variable
  #will be used to predict the row variable

# first make clinic the cluster variable for all predictions (except it's own)
# only need to do it in the column because it is not missing in the rows
predmat[, "clinic.female"] <- -2
predmat["clinic.female", "clinic.female"] <- 0

# 25 imputations
# update default for binary variables (outcome) to account for clustering in "defaultMethod"
imp <- mice(data, m = 25,
                predictorMatrix = predmat,
                defaultMethod = c("pmm", "2l.bin", "polyreg", "polr"),
                seed = 1111)

# now save for future analyses
```

```
Save(imp)

# now check some diagnostics
# stripplots to check distributions of imputed data
stripplot(imp)

# bwplots to re-check distributions of imputed data
bwplot(imp)
```

Web Figure 9 - Covariate Balance

```
# pull in imputations
Load(imp)

# first need to make clinic.female a factor in all iterations
long <- complete(imp, action = "long", include = TRUE)
long$clinic.female <- factor(long$clinic.female)
# now make it back into a mids object
imp <- as.mids(long)
```

```
# create model formula to assess female and male participation per DAG
f_form <- formula(exp.female ~ skills_comp.male + peer_comp.male +
                    rcs(age.female, 4) + who_stage + edu_cat.female +
                    edu_cat.male + rcs(day.female, 4) + rcs(phq9.female, 4) +
                    clinic.female + rcs(soc_sup_ps.female, 4) + rcs(soc_sup_ns.female, 4))
m_form <- formula(exp.male ~ skills_comp.female + peer_comp.female +
                    rcs(age.male, 4) + who_stage.male + edu_cat.female +
                    edu_cat.male + rcs(day.male, 4) + rcs(phq9.male, 4) +
                    clinic.female +  rcs(soc_sup_ps.male, 4) + rcs(soc_sup_ns.male, 4))


# Now assess weights across imputations
f_w_ols <- weightthem(f_form, datasets = imp, approach = "within", method = "ps")
```

```
f_w_cbps <- weightthem(f_form, datasets = imp, approach = "within", method = "cbps",
                       over = FALSE)
```

```
# f_w_npcbps <- weightthem(f_form, datasets = imp, approach = "within", method = "npcbps",
#                          over = FALSE)
m_w_ols <- weightthem(m_form, datasets = imp, approach = "within", method = "ps")
```

```
m_w_cbps <- weightthem(m_form, datasets = imp, approach = "within", method = "cbps",
                       over = FALSE)
```

```
# m_w_npcbps <- weightthem(m_form, datasets = imp, approach = "within", method = "npcbps",
#                          over = FALSE)

# create olr weights from my own function
f_w_olr <- orm.wt(object = imp,
                  exposure = "exp.female",
                  cov_form = "~ skills_comp.male + peer_comp.male + rcs(age.female, 4) +
                  who_stage + edu_cat.female + edu_cat.male + rcs(day.female, 4) +
                  rcs(phq9.female, 4) + clinic.female +
```

```
                  rcs(soc_sup_ps.female, 4) + rcs(soc_sup_ns.female, 4)")
```

```
m_w_olr <- orm.wt(object = imp,
                  exposure = "exp.male",
                  cov_form = "~ skills_comp.female + peer_comp.female + rcs(age.male, 4) +
                  who_stage.male + edu_cat.female + edu_cat.male + rcs(day.male, 4) +
                  rcs(phq9.male, 4) + clinic.female +
                  rcs(soc_sup_ps.male, 4) + rcs(soc_sup_ns.male, 4)")
```

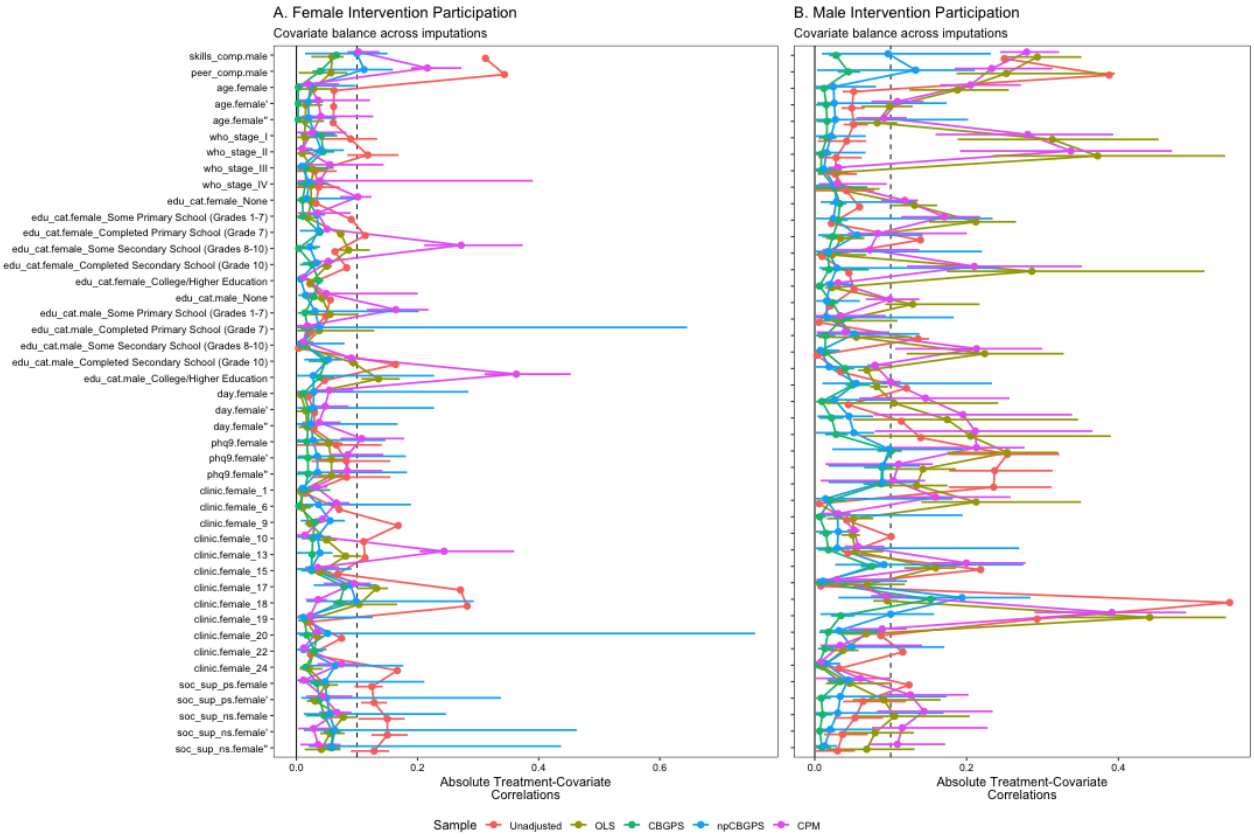
```
# run these seperately because they are slow
f_w_npcbps <- weightthem(f_form, datasets = imp, approach = "within", method = "npcbps",
                         over = FALSE)
Save(f_w_npcbps)
m_w_npcbps <- weightthem(m_form, datasets = imp, approach = "within", method = "npcbps",
                         over = FALSE)
Save(m_w_npcbps)
```

```
# load female and male npcbps
Load(f_w_npcbps)
Load(m_w_npcbps)
# Make female and male plot side by side
# make theme for tables
tt_theme <- ttheme_minimal(base_family = "Arial",
                           base_size = 10,
                           core = list(fg_params = list(hjust = 0, x = 0.1)),
                           colhead = list(fg_params = list(hjust = 0, x = 0.1)))

# female
# first make table
fem_bal_tab <- bal.tab(f_form, data = imp,
        weights = list(OLS = get.w(f_w_ols),
                       CBGPS = get.w(f_w_cbps),
                       npCBGPS = get.w(f_w_npcbps),
                       OLR = unname(unlist(f_w_olr))),
        stats = c("c"),
        un = TRUE, thresholds = c(cor = .1))

# get effective sample size
bal_fem_obs <- tibble(All = rep(NA, 25),
                      OLS = rep(NA, 25),
                      CBGPS = rep(NA, 25),
                      npCBGPS = rep(NA, 25),
                      OLR = rep(NA, 25))
for(i in 1:25) {
    bal_fem_obs[i, ] <- t(fem_bal_tab$Imputation.Balance[[i]]$Observations)
}

# get list of mean number of balanced covariates
bal_fem_cor <- tibble(OLS = rep(NA, 25),
                      CBGPS = rep(NA, 25),
                      npCBGPS = rep(NA, 25),
                      OLR = rep(NA, 25))
for(i in 1:25) {
    bal_fem_cor[i, ] <- fem_bal_tab$Imputation.Balance[[i]]$Balanced.correlations[2, ]
```

```
}

# function to get mean of mean weights from list of weights
mean_wts <- function(object){
  object <- object
  # create list of weights
  wts <- c(rep(NA, length(object$models)))
  # add in mean weights
  for(i in 1:length(wts)){
    wts[i] <- mean(object$models[[i]]$weights)
  }
  wts
}

# create bal_fem table for plot
bal_fem <- tibble(Sample = c("Unadjusted",
                             "OLS",
                             "CBGPS",
                             "npCBGPS",
                             "CPM"),
                  `Correlation > 0.1` = c("",
                                          paste0(mean(bal_fem_cor$OLS), " (",
                                                 min(bal_fem_cor$OLS), ", ",
                                                 max(bal_fem_cor$OLS), ")"),
                                          paste0(mean(bal_fem_cor$CBGPS), " (",
                                                 min(bal_fem_cor$CBGPS), ", ",
                                                 max(bal_fem_cor$CBGPS), ")"),
                                          paste0(mean(bal_fem_cor$npCBGPS), " (",
                                                 min(bal_fem_cor$npCBGPS), ", ",
                                                 max(bal_fem_cor$npCBGPS), ")"),
                                          paste0(mean(bal_fem_cor$OLR), " (",
                                                 min(bal_fem_cor$OLR), ", ",
                                                 max(bal_fem_cor$OLR), ")")),
                  `sIPW Distribution` = c("",
                                          paste0(round(mean(mean_wts(f_w_ols)), 2), " (",
                                                 round(min(mean_wts(f_w_ols)), 2), ", ",
                                                 round(max(mean_wts(f_w_ols)), 2), ")"),
                                          paste0(round(mean(mean_wts(f_w_cbps)), 2), " (",
                                                 round(min(mean_wts(f_w_cbps)), 2), ", ",
                                                 round(max(mean_wts(f_w_cbps)), 2), ")"),
                                          paste0(round(mean(mean_wts(f_w_npcbps)), 2), " (",
                                                 round(min(mean_wts(f_w_npcbps)), 2), ", ",
                                                 round(max(mean_wts(f_w_npcbps)), 2), ")"),
                                          paste0(round(mean(map_dbl(f_w_olr, ~mean(.x))), 2), " (",
                                                 round(min(map_dbl(f_w_olr, ~mean(.x))), 2), ", ",
                                                 round(max(map_dbl(f_w_olr, ~mean(.x))), 2), ")")))

# make into grob
fem_tab <- tableGrob(bal_fem, theme = tt_theme, rows = NULL)
fem_tab <- gtable_add_grob(fem_tab,
                           grobs = rectGrob(gp = gpar(fill= NA, lwd = 2)),
                           t = 1, b = nrow(fem_tab), l = 1, r = ncol(fem_tab))
```

```r
# now make plot
fem_love <- love.plot(f_form, data = imp,
        weights = list(OLS = get.w(f_w_ols),
                       CBGPS = get.w(f_w_cbps),
                       npCBGPS = get.w(f_w_npcbps),
                       CPM = unname(unlist(f_w_olr))),
          stats = c("c"), thresholds = c(cor = .1),
          abs = TRUE, line = TRUE) +
  # annotation_custom(fem_tab,
  #                   xmin = 0.18, ymin = 2, ymax = 8) +
  labs(title = "A. Female Intervention Participation",
       subtitle = "Covariate balance across imputations") +
  theme(text = element_text(family = "Arial"),
        plot.title = element_text(hjust = 0),
        plot.subtitle = element_text(hjust = 0))

# male
# first make table
m_bal_tab <- bal.tab(m_form, data = imp,
        weights = list(OLS = get.w(m_w_ols),
                       CBGPS = get.w(m_w_cbps),
                       npCBGPS = get.w(m_w_npcbps),
                       OLR = unname(unlist(m_w_olr))),
        stats = c("c"),
        un = TRUE, thresholds = c(cor = .1))

# get effective sample size
bal_m_obs <- tibble(All = rep(NA, 25),
                    OLS = rep(NA, 25),
                    CBGPS = rep(NA, 25),
                    npCBGPS = rep(NA, 25),
                    OLR = rep(NA, 25))
for(i in 1:25) {
    bal_m_obs[i, ] <- t(m_bal_tab$Imputation.Balance[[i]]$Observations)
}

# get list of mean number of balanced covariates
bal_m_cor <- tibble(OLS = rep(NA, 25),
                    CBGPS = rep(NA, 25),
                    npCBGPS = rep(NA, 25),
                    OLR = rep(NA, 25))
for(i in 1:25) {
    bal_m_cor[i, ] <- m_bal_tab$Imputation.Balance[[i]]$Balanced.correlations[2, ]
}

# create bal_fem table for plot
bal_male <- tibble(Sample = c("Unadjusted",
                              "OLS",
                              "CBGPS",
                              "npCBGPS",
                              "CPM"),
                   `Correlation > 0.1` = c("",
                                           paste0(mean(bal_m_cor$OLS), " (",
```

```
                                              min(bal_m_cor$OLS), ", ",
                                              max(bal_m_cor$OLS), ")"),
                                       paste0(mean(bal_m_cor$CBGPS), " (",
                                              min(bal_m_cor$CBGPS), ", ",
                                              max(bal_m_cor$CBGPS), ")"),
                                       paste0(mean(bal_m_cor$npCBGPS), " (",
                                              min(bal_m_cor$npCBGPS), ", ",
                                              max(bal_m_cor$npCBGPS), ")"),
                                       paste0(mean(bal_m_cor$OLR), " (",
                                              min(bal_m_cor$OLR), ", ",
                                              max(bal_m_cor$OLR), ")")),
                `sIPW Distribution` = c("",
                                       paste0(round(mean(mean_wts(m_w_ols)), 2), " (",
                                              round(min(mean_wts(m_w_ols)), 2), ", ",
                                              round(max(mean_wts(m_w_ols)), 2), ")"),
                                       paste0(round(mean(mean_wts(m_w_cbps)), 2), " (",
                                              round(min(mean_wts(m_w_cbps)), 2), ", ",
                                              round(max(mean_wts(m_w_cbps)), 2), ")"),
                                       paste0(round(mean(mean_wts(m_w_npcbps)), 2), " (",
                                              round(min(mean_wts(m_w_npcbps)), 2), ", ",
                                              round(max(mean_wts(m_w_npcbps)), 2), ")"),
                                       paste0(round(mean(map_dbl(m_w_olr, ~mean(.x))), 2), " (",
                                              round(min(map_dbl(m_w_olr, ~mean(.x))), 2), ", ",
                                              round(max(map_dbl(m_w_olr, ~mean(.x))), 2), ")")))

# make into grob
male_tab <- tableGrob(bal_male, theme = tt_theme, rows = NULL)
male_tab <- gtable_add_grob(male_tab,
                            grobs = rectGrob(gp = gpar(fill= NA, lwd = 2)),
                            t = 1, b = nrow(male_tab), l = 1, r = ncol(male_tab))

# now make plot
male_love <- love.plot(m_form, data = imp,
        weights = list(OLS = get.w(m_w_ols),
                       CBGPS = get.w(m_w_cbps),
                       npCBGPS = get.w(m_w_npcbps),
                       CPM = unname(unlist(m_w_olr))),
          stats = c("c"), thresholds = c(cor = .1),
          abs = TRUE, line = TRUE) +
  # annotation_custom(male_tab,
  #                   xmin = 0.15, ymin = 2, ymax = 8) +
  labs(title = "B. Male Intervention Participation",
       subtitle = "Covariate balance across imputations") +
  theme(text = element_text(family = "Arial"),
        axis.text.y = element_blank(),
        plot.title = element_text(hjust = 0),
        plot.subtitle = element_text(hjust = 0))
# combine
ggarrange(fem_love, male_love, nrow = 1,
          widths = c(1, 0.6),
          common.legend = TRUE,
          legend = "bottom")
```



All tables are mean (min, max). The points represent the mean absolute treatment-covariate correlation across the 25 imputed datasets and the error bars show the range of correlations across datasets.

Model Fitting

```
# create pool function that bootstraps standard errors for each imputed dataset

# start with function that does resampling at cluster level instead of at individual level
# keep all participants within each cluster
# relabel clusters so there are the same number of clusters at the orginal dataset
resample_cluster <- function(df, clus_id){
  # this is specific how how the clustering is set up (as a factor of numbers)
  # pull out all levels
  levels <- unique(df[clus_id]) %>% unlist() %>% as.character() %>% as.numeric()

  # resample levels with replacement
  resamp_levels <- sample(levels, length(levels), replace = TRUE)

  # new cluster id
  cluster <- c(1:length(levels))

  # create dataset with new cluster id
  resamp_dat <- map2_df(resamp_levels,
                        cluster,
                       ~ df[df[clus_id] == .x, ] %>%
```

```
                          mutate(cluster = .y))
  # make it a factor
  resamp_dat <- resamp_dat %>%
    mutate(cluster = factor(cluster))

  # replace clus_id so that it will work in the formula
  resamp_dat[clus_id] <- resamp_dat$cluster

  # output
  resamp_dat
}

# create function for pooling mm function

# first need to create tidy dataframe with bootstrapped version of glmer
# rows is the number of rows in the combined dataset
fitlist.boot.glmer <- function(object, B) {
  # first get summary
  sum <- summary(object)
  terms <- rownames(sum$coefficients)
  # now make bootstrapped version

  # initialize dataframe
  boot.est <- tibble(B = rep(NA, B))
  for(i in 1:length(terms)){
    boot.est[terms[i]] <- NA
  }
  # fill dataframe
  for(b in 1:B) {
    b_tib <- resample_cluster(df = object@frame, clus_id = names(object@flist))
    # allows for skipping of glmer functions that don't run on particular bootstrapped data
    b_mod <- suppressWarnings(tryCatch(glmer(object@call$formula, data = b_tib,
                                             family = binomial, weights = `(weights)`),
                                       error = function(e) e))
    # if error, retun NA for beta coefficients
    ifelse(any(class(b_mod) == "error"),
           boot.est[b, ] <- t(c(b, NA, NA)),
           boot.est[b, ] <- t(c(b, b_mod@beta)))
  }

  # get standard errors for each term
  ses <- boot.est %>%
    select(-1) %>%
    map_dbl(sd, na.rm = TRUE)
  # now make tidy version
  sum_tidy <- tibble(nobs = nrow(object@frame),
                     term = terms,
                     estimate = object@beta,
                     std.error = ses)
  sum_tidy
}

# barnard rubin function pulled directly from mice code https://rdrr.io/cran/mice/src/R/barnard.rubin.R
```

```r
barnard.rubin <- function(m, b, t, dfcom = 999999) {
  lambda <- (1 + 1 / m) * b / t
  lambda[lambda < 1e-04] <- 1e-04
  dfold <- (m - 1) / lambda^2
  dfobs <- (dfcom + 1) / (dfcom + 3) * dfcom * (1 - lambda)
  dfold * dfobs / (dfold + dfobs)
}

pool.boot.glmer <- function(object, B) {
  call <- paste0("pool(object = ")
  # based on pool.fitlist from https://rdrr.io/cran/mice/src/R/pool.R
  p <- future_map_dfr(object, ~ fitlist.boot.glmer(.x, B))

  # we don't want group_by to change the order of the terms
  p <- p %>% mutate(term = factor(term, levels = unique(term)))

  pooled <- p %>%
    group_by(term) %>%
    summarise(
      m = n(),
      qbar = mean(estimate),
      ubar = mean(std.error^2),
      b = var(estimate),
      t = ubar + (1 + 1 / m) * b,
      dfcom = nrow(object[[1]]@frame),
      df = barnard.rubin(m, b, t, dfcom),
      riv = (1 + 1 / m) * b / ubar,
      lambda = (1 + 1 / m) * b / t,
      fmi = (riv + 2 / (df + 3)) / (riv + 1)
    )
  pooled <- data.frame(pooled)
  names(pooled)[names(pooled) == "qbar"] <- "estimate"

  # finally need make mipo object to work with mice summary function
  mipo.boot.glmer <- list(m = pooled$m[1],
                  pooled = pooled)
  class(mipo.boot.glmer) <- c("mipo", "data.frame")
  mipo.boot.glmer
}
```

```r
# combine imp and weights
f_imp_wts <- complete(imp, action = "long", include = TRUE)
f_imp_wts$ols_wt <- c(rep(NA, 315), get.w(f_w_ols)) # since first 315 are the originals
f_imp_wts$cbgps_wt <- c(rep(NA, 315), get.w(f_w_cbps)) # since first 315 are the originals
f_imp_wts$npcbgps_wt <- c(rep(NA, 315), get.w(f_w_npcbps)) # since first 315 are the originals
f_imp_wts$olr_wt <- c(rep(NA, 315), unlist(f_w_olr)) # since first 315 are the originals
imp_wt_f <- as.mids(f_imp_wts)

# run in parallel with furrr
plan(multisession, workers = 7)

# ols
f_ols_mod_boot <- complete(imp_wt_f, "all") %>%
  map(~ glmer(Y_raw ~ exp.female + (1 | clinic.female),
```

```
              data = .x, weights = ols_wt, family = binomial)) %>%
  pool.boot.glmer(., B = 1000) %>%
  summary()

# cbgps
f_cbgps_mod_boot <- complete(imp_wt_f, "all") %>%
  map(~ glmer(Y_raw ~ exp.female + (1 | clinic.female),
              data = .x, weights = cbgps_wt, family = binomial)) %>%
  pool.boot.glmer(., B = 1000) %>%
  summary()

# npcbgps
f_npcbgps_mod_boot <- complete(imp_wt_f, "all") %>%
  map(~ glmer(Y_raw ~ exp.female + (1 | clinic.female),
              data = .x, weights = npcbgps_wt, family = binomial)) %>%
  pool.boot.glmer(., B = 1000) %>%
  summary()

# olr
f_olr_mod_boot <- complete(imp_wt_f, "all") %>%
  map(~ glmer(Y_raw ~ exp.female + (1 | clinic.female),
              data = .x, weights = olr_wt, family = binomial)) %>%
  pool.boot.glmer(., B = 1000) %>%
  summary()

# save as list
f_mods_boot <- list(OLS = f_ols_mod_boot,
                    CBGPS = f_cbgps_mod_boot,
                    npCBGPS = f_npcbgps_mod_boot,
                    OLR = f_olr_mod_boot)
Save(f_mods_boot)
```

```
# combine imp and weights (TBD)
m_imp_wts <- complete(imp, action = "long", include = TRUE)
m_imp_wts$ols_wt <- c(rep(NA, 315), get.w(m_w_ols)) # since mirst 315 are the originals
m_imp_wts$cbgps_wt <- c(rep(NA, 315), get.w(m_w_cbps)) # since mirst 315 are the originals
m_imp_wts$npcbgps_wt <- c(rep(NA, 315), get.w(m_w_npcbps)) # since mirst 315 are the originals
m_imp_wts$olr_wt <- c(rep(NA, 315), unlist(m_w_olr)) # since mirst 315 are the originals
imp_wt_m <- as.mids(m_imp_wts)

# run in parallel with murrr
plan(multisession, workers = 7)

# ols
m_ols_mod_boot <- complete(imp_wt_m, "all") %>%
  map(~ glmer(Y_raw ~ exp.male + (1 | clinic.female),
              data = .x, weights = ols_wt, family = binomial)) %>%
  pool.boot.glmer(., B = 1000) %>%
  summary()

# cbgps
m_cbgps_mod_boot <- complete(imp_wt_m, "all") %>%
  map(~ glmer(Y_raw ~ exp.male + (1 | clinic.female),
              data = .x, weights = cbgps_wt, family = binomial)) %>%
```

```
  pool.boot.glmer(., B = 1000) %>%
  summary()

# npcbgps
m_npcbgps_mod_boot <- complete(imp_wt_m, "all") %>%
  map(~ glmer(Y_raw ~ exp.male + (1 | clinic.female),
              data = .x, weights = npcbgps_wt, family = binomial)) %>%
  pool.boot.glmer(., B = 1000) %>%
  summary()

# olr
m_olr_mod_boot <- complete(imp_wt_m, "all") %>%
  map(~ glmer(Y_raw ~ exp.male + (1 | clinic.female),
              data = .x, weights = olr_wt, family = binomial)) %>%
  pool.boot.glmer(., B = 1000) %>%
  summary()

# save as list
m_mods_boot <- list(OLS = m_ols_mod_boot,
                    CBGPS = m_cbgps_mod_boot,
                    npCBGPS = m_npcbgps_mod_boot,
                    OLR = m_olr_mod_boot)
Save(m_mods_boot)
```

Table 5 - Method Characteristics and Likelihood of Female Postpartum Contraceptive Uptake

```
Load(f_mods_boot)
Load(m_mods_boot)

# female participants
fem_or <- tibble(Analysis = c("OLS",
                              "CBGPS",
                              "npCBGPS",
                              "CPM"),
                 coef = c(f_mods_boot$OLS$estimate[2],
                          f_mods_boot$CBGPS$estimate[2],
                          f_mods_boot$npCBGPS$estimate[2],
                          f_mods_boot$OLR$estimate[2]),
                    se = c(f_mods_boot$OLS$std.error[2],
                          f_mods_boot$CBGPS$std.error[2],
                          f_mods_boot$npCBGPS$std.error[2],
                          f_mods_boot$OLR$std.error[2])) %>%
  mutate(lb = coef - (1.96*se),
         ub = coef + (1.96*se),
         or = exp(coef),
         or_lb = exp(lb),
         or_ub = exp(ub),
         `Odds Ratio (95% Confidence Interval)` = paste0(round(or, 2), " (", round(or_lb, 3), ", ", rou

# male participants
```

```r
m_or <- tibble(Analysis = c("OLS",
                            "CBGPS",
                            "npCBGPS",
                            "CPM"),
               coef = c(m_mods_boot$OLS$estimate[2],
                        m_mods_boot$CBGPS$estimate[2],
                        m_mods_boot$npCBGPS$estimate[2],
                        m_mods_boot$OLR$estimate[2]),
                  se = c(m_mods_boot$OLS$std.error[2],
                        m_mods_boot$CBGPS$std.error[2],
                        m_mods_boot$npCBGPS$std.error[2],
                        m_mods_boot$OLR$std.error[2])) %>%
  mutate(lb = coef - (1.96*se),
         ub = coef + (1.96*se),
         or = exp(coef),
         or_lb = exp(lb),
         or_ub = exp(ub),
         `Odds Ratio (95% Confidence Interval)` = paste0(round(or, 2), " (", round(or_lb, 3), ", ", rou

# now make a forestplot
tab4 <- bind_cols(c("", "Female Participants", "Mean sIPW (min, max)",
                          "Mean correlations > 0.1 (min, max)",
                          "+1 session [OR (95% CI)]",
                          "Male Participants", "Mean sIPW (min, max)",
                          "Mean correlations > 0.1 (min, max)",
                          "+1 session [OR (95% CI)]"),
                    c("OLS", "",
                      bal_fem$`sIPW Distribution`[2],
                      bal_fem$`Correlation > 0.1`[2],
                      fem_or$`Odds Ratio (95% Confidence Interval)`[1],
                      "",
                      bal_male$`sIPW Distribution`[2],
                      bal_male$`Correlation > 0.1`[2],
                      m_or$`Odds Ratio (95% Confidence Interval)`[1]),
                    c("CBGPS", "",
                      bal_fem$`sIPW Distribution`[3],
                      bal_fem$`Correlation > 0.1`[3],
                      fem_or$`Odds Ratio (95% Confidence Interval)`[2],
                      "",
                      bal_male$`sIPW Distribution`[3],
                      bal_male$`Correlation > 0.1`[3],
                      m_or$`Odds Ratio (95% Confidence Interval)`[2]),
                    c("npCBGPS", "",
                      bal_fem$`sIPW Distribution`[4],
                      bal_fem$`Correlation > 0.1`[4],
                      fem_or$`Odds Ratio (95% Confidence Interval)`[3],
                      "",
                      bal_male$`sIPW Distribution`[4],
                      bal_male$`Correlation > 0.1`[4],
                      m_or$`Odds Ratio (95% Confidence Interval)`[3]),
                    c("CPM", "",
                      bal_fem$`sIPW Distribution`[5],
                      bal_fem$`Correlation > 0.1`[5],
```

```
                    fem_or$`Odds Ratio (95% Confidence Interval)`[4],
                    "",
                    bal_male$`sIPW Distribution`[5],
                    bal_male$`Correlation > 0.1`[5],
                    m_or$`Odds Ratio (95% Confidence Interval)`[4]))
kable(tab4)
```

Session Info

```
sessionInfo()
```

```
R version 4.1.0 (2021-05-18)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS 13.0.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRblas.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRlapack.dylib

Random number generation:
 RNG:     L'Ecuyer-CMRG
 Normal:  Inversion
 Sample:  Rejection

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] grid      parallel  stats     graphics  grDevices utils     datasets
[8] methods   base

other attached packages:
 [1] furrr_0.2.3        future_1.23.0      ggeffects_1.1.4    rsvg_2.1.2
 [5] DiagrammeRsvg_0.1  DiagrammeR_1.0.6.1 forestplot_2.0.1   checkmate_2.0.0
 [9] magrittr_2.0.2     broom.mixed_0.2.7  gtable_0.3.0       gridExtra_2.3
[13] table1_1.4.2       ggdist_3.0.0       extrafont_0.17     doParallel_1.0.16
[17] iterators_1.0.14   foreach_1.5.2      ggpubr_0.4.0       kableExtra_1.3.4
[21] lme4_1.1-27.1      Matrix_1.3-4       cobalt_4.3.1       MatchThem_1.0.1
[25] WeightIt_0.12.0    mice_3.13.0        rms_6.2-0          SparseM_1.81
[29] Hmisc_4.6-0        Formula_1.2-4      survival_3.2-13    lattice_0.20-45
[33] forcats_0.5.1      stringr_1.4.0      dplyr_1.0.8        purrr_0.3.4
[37] readr_2.0.2        tidyr_1.2.0        tibble_3.1.6       ggplot2_3.3.5
[41] tidyverse_1.3.1

loaded via a namespace (and not attached):
  [1] utf8_1.2.2           tidyselect_1.1.2     htmlwidgets_1.5.4
  [4] pROC_1.18.0          munsell_0.5.0        codetools_0.2-18
  [7] MatchIt_4.3.0        withr_2.5.0          colorspace_2.0-3
 [10] knitr_1.37           rstudioapi_0.13      stats4_4.1.0
 [13] ggsignif_0.6.3       Rttf2pt1_1.3.9       listenv_0.8.0
 [16] labeling_0.4.2       farver_2.1.0         parallelly_1.28.1
 [19] vctrs_0.3.8          CBPS_0.22            generics_0.1.2
```

```
 [22] TH.data_1.1-0        ipred_0.9-12         xfun_0.30
 [25] R6_2.5.1             assertthat_0.2.1     scales_1.1.1
 [28] multcomp_1.4-17      nnet_7.3-16          globals_0.14.0
 [31] conquer_1.2.1        processx_3.5.2       sandwich_3.0-1
 [34] timeDate_3043.102    rlang_1.0.2          MatrixModels_0.5-0
 [37] systemfonts_1.0.3    splines_4.1.0        rstatix_0.7.0
 [40] extrafontdb_1.0      ModelMetrics_1.2.2.2 broom_0.7.10
 [43] yaml_2.3.5           reshape2_1.4.4       abind_1.4-5
 [46] modelr_0.1.8         backports_1.3.0      caret_6.0-90
 [49] tools_4.1.0          lava_1.6.10          ellipsis_0.3.2
 [52] RColorBrewer_1.1-2   Rcpp_1.0.8.3         plyr_1.8.6
 [55] base64enc_0.1-3      visNetwork_2.1.0     ps_1.6.0
 [58] rpart_4.1-15         cowplot_1.1.1        zoo_1.8-9
 [61] haven_2.4.3          cluster_2.1.2        fs_1.5.2
 [64] survey_4.1-1         data.table_1.14.2    magick_2.7.3
 [67] reprex_2.0.1         mvtnorm_1.1-3        matrixStats_0.61.0
 [70] hms_1.1.1            evaluate_0.15        jpeg_0.1-9
 [73] readxl_1.3.1         shape_1.4.6          compiler_4.1.0
 [76] V8_3.6.0             crayon_1.5.0         minqa_1.2.4
 [79] htmltools_0.5.2      tzdb_0.2.0           lubridate_1.8.0
 [82] DBI_1.1.1            dbplyr_2.1.1         MASS_7.3-54
 [85] boot_1.3-28          car_3.0-12           cli_3.2.0
 [88] mitools_2.4          gower_0.2.2          pkgconfig_2.0.3
 [91] numDeriv_2016.8-1.1  foreign_0.8-81       recipes_0.1.17
 [94] xml2_1.3.3           svglite_2.0.0        webshot_0.5.2
 [97] prodlim_2019.11.13   rvest_1.0.2          distributional_0.2.2
[100] callr_3.7.0          digest_0.6.29        rmarkdown_2.13
[103] cellranger_1.1.0     htmlTable_2.3.0      curl_4.3.2
[106] quantreg_5.86        nloptr_1.2.2.3       lifecycle_1.0.1
[109] nlme_3.1-153         jsonlite_1.8.0       carData_3.0-4
[112] viridisLite_0.4.0    fansi_1.0.2          pillar_1.7.0
[115] fastmap_1.1.0        httr_1.4.2           glue_1.6.2
[118] png_0.1-7            glmnet_4.1-2         class_7.3-19
[121] stringi_1.7.6        polspline_1.1.19     latticeExtra_0.6-29
[124] future.apply_1.8.1
```

References

1. Audet CM, Graves E, Barreto E, et al. Partners-based HIV treatment for seroconcordant couples attending antenatal and postnatal care in rural Mozambique: A cluster randomized trial protocol. *Contemp Clin Trials*. 2018;71:63–69.
2. Kroenke K, Spitzer RL, Williams JB. The PHQ-9: Validity of a brief depression severity measure. *J Gen Intern Med* [electronic article]. 2001;16(9):606–13. (https://www.ncbi.nlm.nih.gov/pubmed/11556941)
3. Schulz U, Schwarzer R. Social support in coping with illness: The Berlin Social Support Scales (BSSS). *Diagnostica* [electronic article]. 2003;49(2):73–82. (://WOS:000182834400003)