BCPP Preliminary Analysis

Melody Owen

2025-05-30

Contents

OVERVIEW

The Botswana Combination Prevention Project (BCPP)

Motivation

- Goal: The primary goal of BCPP was to determine whether implementation of combination prevention package (CP) can significantly reduce population-level, cumulative HIV incidence
- Population: Individuals in Botswana aged 16-64 years
- Timeline: Study length was approximately 3 years
- Design: 30 communities were selected and matched into pairs based on community characteristics thought to be associated with HIV incidence

Treatment in BCPP

The Combination Prevention (CP) prevention package included the following four components:

1. VMMC: Male circumcision (only for HIV-negative males)
2. HTC: HIV Testing and Counseling (only for HIV-negative individuals)
3. ART: Antiretroviral Therapy (only for HIV-positive individuals)
4. PMTCT: Prevention of mother-to-child transmission (only for pregnant HIV-positive females)

Clusters (30 communities) were randomized to either:

- Treatment: CP Package
- Control: Standard of Care

Our analysis examines the impact of CP for HIV-negative individuals, so we consider the "entire" package components 1 and 2 only.

Terminology

Individual Effects: Refers to an effect of one's own input on their own outcome

Spillover Effects: Refers to an effect of others' inputs on one's outcome

Direct Effects: Refers to an effect pathway that links directly from an input to an output with nothing else on the pathway

Indirect Effects: Refers to an effect pathway that links indirectly from an input to an output through a node (mediator) on the pathway

With these, we can define the following:

Individual Effects

- **Individual Direct Effect**: Path from one's own input to their own outcome with no other nodes on the pathway
- **Individual Indirect Effect**: Path from one's own input to their own outcome through a node (mediator) on the pathway
- **Total Individual Effect**: Total effect of one's input on their outcome through all pathways (Individual Direct + Individual Indirect)

Spillover Effects

- **Spillover Direct Effect**: Path from others' inputs to one's outcome with no other nodes on the pathway
- **Spillover Indirect Effect**: Path from others' inputs to one's outcome through a node (mediator) on the pathway
- **Total Spillover Effect**: Total effect of other's inputs on one's outcome through all pathways (Spillover Direct + Spillover Indirect)

Overall Effects

- **Overall Effect**: Total Individual Effect + Total Spillover Effect

Questions of Interest

1. What is the direct individual effect of the CP intervention on HIV incidence?
2. To what extent is the total individual effect mediated by Voluntary Male Medical Circumcision (VMMC)?
3. What is the direct spillover effect of the CP intervention on HIV incidence?
4. To what extent is the total spillover effect mediated by VMMC?
5. What is the overall effect of CP on HIV incidence? ("overall" includes both spillover and individual totals)
6. To what extent is the overall effect mediated by VMMC?

NOTATION

K is the total number of villages in the study, indexed as $k = 1, ..., K$

m_k is the total number of individuals in cluster k, indexed as $i = 1, ..., m_k$

- $m_k^{(\text{male})}$ are the total number of males in cluster k
- $m_k^{(\text{female})}$ are the total number of females in cluster k

Y_{ik} is the outcome of subject i in cluster k, and is binary

- In BCPP, $Y_{ik} = 1$ if a subject seroconverted by the end of the study, $Y_{ik} = 0$ otherwise

T_k is the cluster-level binary treatment assignment

- In BCPP, $T_k = 1$ if a cluster has been assigned to receive CP, and $T_k = 0$ otherwise

$X_{ik}^{(1)}$, $X_{ik}^{(2)}$ denotes each of the two components of the treatment, T_k.

- In BCPP, the Combination Prevention (CP) package included the following:
  1. MC: Male Circumcision (available only for HIV-negative males)
  2. HTC: HIV Testing and Counseling (available only for HIV-negative individuals)
  3. ART: Antiretroviral Therapy (available only for HIV-positive individuals)
  4. PMTCT: Prevention of Mother-to-Child Transmission (available only for HIV-positive females)
- We are only considering the first two components as the entire treatment package, since the last two apply to HIV-positive individuals only.
- $X_{ik}^{(1)}$ = "Yes" if individual i in cluster k was circumcised before or during the study, $X_{ik}^{(1)}$ = "No" if they are male and not circumcised, and $X_{ik}^{(1)}$ = "Female" if they are female (three levels are included as to not exclude females)
- $X_{ik}^{(2)} = 1$ if individual i in cluster k received HTC at enrollment or thereafter, and $X_{ik}^{(2)} = 0$ otherwise

$X_{ik}^{(12)}$ denotes whether individual i in cluster k received the entire treatment

- For males in BCPP, $X_{ik}^{(12)} = X_{ik}^{(1)} \times X_{ik}^{(2)} = 1$ if they received both MC and HTC, $X_{ik}^{(12)} = 0$ otherwise
- For females in BCPP, $X_{ik}^{(12)} = X_{ik}^{(2)} = 1$ if they received HTC, $X_{ik}^{(12)} = 0$ otherwise

$Z_k^{(1)}$, $Z_k^{(2)}$ is the proportion of indivividuals in village k who received the first component and second component of the treatment, respectively

- For males in BCPP, $Z_k^{(1)} = \sum_{i=1}^{m_k^{(\text{male})}} \frac{X_{ik}^{(1)}}{m_k^{(\text{male})}}$ is the proportion of males in village k who are circumcised before or during the study
- For all individuals in BCPP, $Z_k^{(2)} = \sum_{i=1}^{m_k} \frac{X_{ik}^{(2)}}{m_k}$ is the proportion of all individuals in village k who received HTC

$Z_{ik}^{(12)}$ is the proportion of individuals who received the full treatment

- For males in BCPP, $Z_{ik}^{(12)} = \sum_{i=1}^{m_k^{(\text{male})}} \frac{X_{ik}^{(1)} \times X_{ik}^{(2)}}{m_k^{(\text{male})}}$ is the proportion of males who are both circumcised and received HTC
- For females in BCPP, $Z_{ik}^{(12)} = Z_{ik}^{(2)} = \sum_{i=1}^{m_k^{(\text{female})}} \frac{X_{ik}^{(2)}}{m_k^{(\text{female})}}$ is the proprtion of females who received HTC

$\mathbf{C}_{ik} = (C_{1k}^{(1)}, ..., C_{m_k k}^{(1)}, C_{1k}^{(2)}, ..., C_{m_k k}^{(2)})$ are the individual level covariates

$\mathbf{V}_k = (V_k^{(1)}, ..., V_k^{(v)})$ are the cluster-level covariates

BASELINE CHARACTERISTICS

Characteristics Before Exclusions

The original dataset has 13131 total individuals in the study; 6591 in the treatment group, and 6540 in the control arm.

Variable	Level	Control	Treatment	Overall	Missing (Control)	Missing (Treatment)	Missing
Number of Individuals		6540	6591	13131			
Number of Clusters		15	15	30			
Mean Cluster Size		459	461	460			
Gender	Male	2378 (36%)	2413 (37%)	4791 (36%)			
	Female	4162 (64%)	4178 (63%)	8340 (64%)			
HIV Status at Start	HIV-uninfected	4487 (72%)	4487 (71%)	8974 (71%)	267 (4%)	254 (4%)	521 (4%)
	HIV-infected	1771 (28%)	1825 (29%)	3596 (29%)			
	Refused HIV testing	15 (0%)	25 (0%)	40 (0%)			
Treatment Component: MC	Yes	226 (4%)	335 (5%)	561 (5%)	437 (7%)	386 (6%)	823 (6%)
	No	1063 (17%)	915 (15%)	1978 (16%)			
	Began study circumcised	652 (11%)	777 (13%)	1429 (12%)			
	Female	4162 (68%)	4178 (67%)	8340 (68%)			
Treatment Component: HTC	Yes	2371 (38%)	2329 (37%)	4700 (37%)	267 (4%)	254 (4%)	521 (4%)
	No	3902 (62%)	4008 (63%)	7910 (63%)			
Treatment Component: Full	Yes	1963 (33%)	1966 (33%)	3929 (33%)	621 (9%)	581 (9%)	1202 (9%
	No	3956 (67%)	4044 (67%)	8000 (67%)			
Outcome: HIV Seroconversion (3-year period)	Yes	90 (1%)	57 (1%)	147 (1%)	477 (7%)	507 (8%)	984 (7%)
	No	4202 (69%)	4202 (69%)	8404 (69%)			
	Began study HIV-infected	1771 (29%)	1825 (30%)	3596 (30%)			

Table 1: Characteristics by treatment group before exclusions

Table below displays the mean proportion, per cluster, of various characteristics, including mean proportion of HIV infected individuals per cluster at baseline, etc. These are calculated before any exclusions. Note that for the proportion of males circumcised in a given cluster, this includes both circumcision that occurred during and before the study.

Variable	Control	Treatment
Proportion of HIV Infected in Cluster (Mean, SD)	0.28 (0.07)	0.28 (0.07)
Proportion of Males in Cluster (Mean, SD)	0.36 (0.02)	0.36 (0.03)
Proportion of Males Circumcised in Cluster (Mean, SD)	0.37 (0.06)	0.46 (0.07)
Proportion HTC in Cluster (Mean, SD)	0.37 (0.06)	0.36 (0.05)
Proportion Fully Treated in Cluster (Mean, SD)	0.3 (0.05)	0.3 (0.05)

Table 2: Cluster-level proportions by treatment group before exclusions

Characteristics After Exclusions

A total of 4580 individuals were excluded from the analysis dataset. This is because these individuals either began the study as HIV-positive (n = 3596), refused HIV testing (n = 40), or had a missing value (n = 521). Or, if they began the study HIV-negative, if they had a missing value for seroconversion (the outcome), they were excluded (n = 423).

Note that in our analyses, we evaluated whether the intervention reduced HIV incidence by modeling seroconversion among individuals who were HIV-negative at baseline (n = 8551). Although the analysis was restricted to this at-risk subset, all cluster-level characteristics (e.g., proportion HIV-positive at baseline, proportion of men circumcised, etc.) were calculated using the full study population. This approach ensures that the covariates reflect the overall context and implementation environment of each cluster, rather than being limited to the analytic subset.

The following table shows the baseline characteristics of the new dataset that excludes these individuals (n = 8551).

Variable	Level	Control	Treatment	Overall	Missing (Control)	Missing (Treatment)	Missin
Number of Individuals		4292	4259	8551			
Number of Clusters		15	15	30			
Mean Cluster Size		460	465	462.5			
Gender	Male	1679 (39%)	1673 (39%)	3352 (39%)			
	Female	2613 (61%)	2586 (61%)	5199 (61%)			
HIV Status at Start	HIV-uninfected	4292 (100%)	4259 (100%)	8551 (100%)			
Treatment Component: MC	Yes	149 (4%)	238 (6%)	387 (5%)	138 (2%)	86 (1%)	224 (2
	No	868 (21%)	723 (17%)	1591 (19%)			
	Began study circumcised	524 (13%)	626 (15%)	1150 (14%)			
	Female	2613 (63%)	2586 (62%)	5199 (62%)			
Treatment Component: HTC	Yes	844 (20%)	771 (18%)	1615 (19%)			
	No	3448 (80%)	3488 (82%)	6936 (81%)			
Treatment Component: Full	Yes	716 (17%)	690 (17%)	1406 (17%)	128 (2%)	81 (1%)	209 (2
	No	3448 (83%)	3488 (83%)	6936 (83%)			
Outcome: HIV Seroconversion (3-year period)	Yes	90 (2%)	57 (1%)	147 (2%)			
	No	4202 (98%)	4202 (99%)	8404 (98%)			

Table 3: Characteristics by treatment group after exclusions

MODELING RESULTS

Within-Village Spillover

Setup

- In this analysis, we include everyone in the study (who began the study HIV-negative) who DID NOT receive any part of the treatment.
- This setup will allow us to estimate
    a. Total Within-Cluster Spillover Effect of the Intervention
    b. Within-Cluster Spillover of the Intervention Effect Not through Male Circumcision
    c. Proportion of Within-Intervention Village Spillover Effect Mediated by Circumcision

Dataset

```
# Only include those in treatment group who DID NOT receive any part of the treatment
# Only include those in control group who DID NOT receive any part of treatment
modelDat_SpW <- modelDat %>%
  filter(X1_ik != "Yes", X2_ik == 0) # Exclude anyone who got any part of the treatment
```

The total sample size for this analysis is 5638, meaning that 2913 individuals are excluded.

Data Characteristics

Variable	Level	Control	Treatment	Overall
Number of Individuals		2872	2766	5638
Number of Clusters		15	15	30
Mean Cluster Size		458	464	461
Gender	Male	868 (30%)	723 (26%)	1591 (28%)
	Female	2004 (70%)	2043 (74%)	4047 (72%)
HIV Status at Start	HIV-uninfected	2872 (100%)	2766 (100%)	5638 (100%)
Treatment Component: MC	No	868 (30%)	723 (26%)	1591 (28%)
	Female	2004 (70%)	2043 (74%)	4047 (72%)
Treatment Component: HTC	No	2872 (100%)	2766 (100%)	5638 (100%)
Treatment Component: Full	No	2872 (100%)	2766 (100%)	5638 (100%)
Outcome: HIV Seroconversion (3-year period)	Yes	49 (2%)	33 (1%)	82 (1%)
	No	2823 (98%)	2733 (99%)	5556 (99%)

Table 4: Characteristics of Spillover Effects Analysis Data

A. Total Within-Cluster Spillover Effect of the Intervention

"SpW" denotes total spillover within intervention clusters. This compares participants in intervention villages who received neither relevant intervention component to people in the control villages (who also did not receive any part of the intervention component)

Then, under certain assumptions, the only way for an intervention village participant to have lower HIV risk is by association with others in the village with lower HIV risk because of their exposure to the intervention.

$$logit(Y_{ik}) = \beta_0^{\text{SpW}} + \beta_1^{\text{SpW}}(T_k)$$

Then $\exp\left(\beta_1^{\text{SpW}}\right)$ is a within-village spillover OR, and estimates the causal effect of living in a CP village, despite receiving no components oneself, on the odds of seroconversion. This is total within-village spillover effect.

```r
# Total Within-Cluster Spillover Effect of the Intervention Model

# Model not accounting for clustering
model_SpW <- glm(Y_ik ~ T_k,
                 family = binomial(link = 'logit'),
                 data = modelDat_SpW) # Exclude those who received full trt

# Model accounting for clustering using GLMM
model_SpW_glmm <- glmer(Y_ik ~ T_k + (1|cluster_id), # Uses exchangeable structure
                        data = modelDat_SpW,
                        family = binomial(link = "logit"))

# Model accounting for clustering using GEE
model_SpW_gee <- geeglm(Y_ik ~ T_k,
                        family = binomial(link = "logit"), # logit link
                        id = cluster_id, # clustering variable
                        data = modelDat_SpW,
                        corstr  = "exchangeable") # working correlation
```

Model	Term	OR [95% CI]	p-value	ICC
GLM	T_k	0.696 [0.442, 1.08]	0.11	
GLMM	T_k	0.694 [0.434, 1.109]	0.13	0.01
GEE	T_k	0.694 [0.439, 1.096]	0.12	0.00

Table 5: Spillover Within Intervention Clusters Model Output

Thus, among people who received none of the intervention components, those living in CP villages had 30% lower odds of HIV seroconversion than otherwise comparable untreated people in control villages. Since every individual in this analytic set is personally untreated, any difference in their HIV risk can only arise from indirect protection, and thus, 0.7 is interpreted as the within-village spillover effect of CP.

B. Within-Cluster Spillover of the Intervention Effect Not through Male Circumcision

"SpWR" denotes all the remaining spillover that affects one's outcome that exists when we block the mediated spillover path that exists through male circumcision.

$$logit(Y_{ik}) = \beta_0^{\text{SpWR}} + \beta_1^{\text{SpWR}}(T_k) + \beta_2^{\text{SpWR}}(Z_k^{(1)})$$

Here, $\exp\left(\beta_1^{\text{SpWR}}\right)$ compares untreated individuals in CP villages with untreated individuals in control villages after we hold the village's male-circumcision coverage fixed at the same value for both groups. So, it's the OR for the remaining within-village spillover - whatever protection (or risk) is left once the male-circumcision pathway has been accounted for.

```
# Within-Cluster Spillover of the Intervention Not Through MC

# Model not accounting for clustering
model_SpWR <- glm(Y_ik ~ T_k + Z1_k,
                  family = binomial(link = 'logit'),
                  data = modelDat_SpW)

# Model accounting for clustering using GLMM
model_SpWR_glmm <- glmer(Y_ik ~ T_k + Z1_k + (1|cluster_id), # Uses exchangeable
                         data = modelDat_SpW,
                         family = binomial(link = "logit"))
```

```
## boundary (singular) fit: see help('isSingular')
```

```
# Model accounting for clustering using GEE
model_SpWR_gee <- geeglm(Y_ik ~ T_k + Z1_k,
                         family = binomial(link = "logit"),
                         id = cluster_id,
                         data = modelDat_SpW,
                         corstr = "exchangeable") # working correlation
```

Model	Term	OR [95% CI]	p-value	ICC
GLM	T_k	0.895 [0.524, 1.517]	0.68	
GLM	Z1_k	0.046 [0.001, 1.563]	0.10	
GLMM	T_k	0.895 [0.527, 1.521]	0.68	0.01
GLMM	Z1_k	0.046 [0.001, 1.686]	0.09	
GEE	T_k	0.896 [0.526, 1.524]	0.68	-0.00
GEE	Z1_k	0.045 [0.002, 1.176]	0.06	

Table 6: Spillover Not Due to Male Circumcision Model Output

After we hold village circumcision coverage fixed, untreated residence of CP villages will still have a 10% lower odds of seroconversion than untreated residence of control villages. This is spillover that operates through pathways other than male-circumcision coverage (e.g. HTC uptake, general behavior change, program outreach).

Then, moving from a 0% to 100% male circumcised coverage in a village multiplies an untreated person's odds of seroconversion by 0.05. This means 95% lower odds of HIV acquisition for an untreated person when their village goes from zero to complete male-circumcision coverage.

C. Proportion of Within-Intervention Village Spillover Effect Mediated by Circumcision

Then, the proportion of within-intervention village spillover effect mediated by circumcision is

$$\frac{\beta_1^{\mathrm{SpW}} - \beta_1^{\mathrm{SpWR}}}{\beta_1^{\mathrm{SpW}}}$$

Essentially, this is the total spillover minus the spillover that exists except through the circumcision component, divided by spillover total.

Model	Proportion of Spillover Mediated by MC
GLM	0.70
GLMM	0.70
GEE	0.70

Table 7: Proportion of Within-Cluster Spillover Due to Male Circumcision

Thus, about 70% of within-village spillover protection experienced by untreated people in CP villages is explained by the higher male-circumcision coverage in those villages. The remaining spillover benefit must come through other village-level channels (e.g. HTC uptake, community health behavior change, program outreach, etc.)

Individual Effects

Setup

- In this analysis, we include only males in the study.
- Here, we will estimate the effects of the intervention assignment on the outcome. In the mediation model, we will account for if they actually received the circumcision component or not.
- This setup will allow us to estimate

  d. Total Individual Effect of Treatment Assignment
  e. Individual Direct Effects of Treatment Assignment
  f. Indirect Individual Effect of Treatment Assignment
  g. Proportion of Individual Effect of Treatment Assignment Mediated by Circumcision

```
# Alternative to fix data availability
  # Include only those who were circumcised in the treatment
  # Include everyone in the control
modelDat_Ind <- modelDat %>%
  filter(C1_ik == 1)
```

The total sample size for this analysis is 3352, meaning that 5199 individuals are excluded.

Missing Data Imputation

There are 224 cases where a male has a missing value for his circumcision status ($X_{ik}^{(1)}$). Because this analysis is focused on calculating the mediated effect that circumcision has on HIV incidence, we will use missing data imputation methods for these cases.

X1_ik	Y_ik = Yes	Y_ik = No
X1_ik = Yes	9	1528
X1_ik = No	0	1591
X1_ik = Missing	19	205

Table 8: Counts of HIV Seroconversion (Y) by Male Circumcision (X1)

```
# THIS APPROACH DOES NOT WORK
# It just predicts everything to be Y_ik = 0 so it does not help with the problem that we have only 9 observat
# So I had to use Ridge regression to make the imputation model a bit more flexible
# 1. Prepare data ------------------------------------------------------------
modelDat_Ind_test <- modelDat_Ind %>%
  mutate(X1_ik = ifelse(X1_ik == "Yes", 1, ifelse(X1_ik == "No", 0, NA)))

# Define variables
imp_vars_test <- c("cluster_id", "Y_ik", "X1_ik", "X2_ik", "T_k", "Z1_k", "Z2_k")

# Separate data into complete cases and missing cases
dat_complete_test <- filter(modelDat_Ind_test, !is.na(X1_ik))[, imp_vars_test]
dat_missing_test <- filter(modelDat_Ind_test, is.na(X1_ik))[, imp_vars_test]

# 2. Fit logistic model for missing variable X1_ik ---------------------------
lower_form_test <- X1_ik ~ Y_ik * T_k

upper_form_test <- X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Z2_k +
  Y_ik:T_k + Y_ik:X2_ik + Y_ik:Z1_k + Y_ik:Z2_k +
  T_k:Z1_k + T_k:Z2_k + T_k:X2_ik +
  X2_ik:Z1_k + X2_ik:Z2_k

force_terms_test <- c("Y_ik", "T_k", "Y_ik:T_k")

# Start with the full model
glm_full_test <- glm(upper_form_test, data = dat_complete_test,
                     family = binomial(link = "logit"))
```

```r
glm_step_test <- stepAIC(glm_full_test,
                    scope = list(lower = lower_form_test, upper = upper_form_test),
                    direction = "both",
                    k = 4.6,
                    trace = FALSE)

summary(glm_step_test)
```

```
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial(link = "logit"),
##     data = dat_complete_test)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
```

```r
# Function to complete one imputation
impute_once_test <- function(seed = NULL) {
  if(!is.null(seed)) set.seed(seed)
  p_hat <- predict(glm_step_test, newdata = dat_missing_test, type = "response")
  x_imp <- rbinom(length(p_hat), 1, p_hat)
  dat_imp <- modelDat_Ind_test
  dat_imp$X1_ik[is.na(dat_imp$X1_ik)] <- x_imp
  dat_imp
}

# 3. Generate datasets ------------------------------------------------------
# Now generate m = 10 completed datasets
m <- 10
imp_list_test <- lapply(1:m, function(j) impute_once_test(seed = j))

imp_data_test <- data.table::rbindlist(imp_list_test, idcol = "m") %>%
  mutate(X1_ik = ifelse(X1_ik == 1, "Yes", ifelse(X1_ik == 0, "No", NA)),
         Y_ik = ifelse(Y_ik == 1, "Yes", ifelse(Y_ik == 0, "No", NA))) %>%
  dplyr::select(m, X1_ik, Y_ik) %>%
  mutate(Y_ik = paste0("Y_ik = ", Y_ik)) %>%
  mutate(X1_ik = paste0("X1_ik = ", X1_ik)) %>%
  group_by(m, X1_ik, Y_ik) %>%
  dplyr::summarize(n = n()) %>%
  ungroup() %>%
  pivot_wider(names_from = Y_ik, values_from = n) %>%
  dplyr::select(m, `X1_ik`, `Y_ik = Yes`, `Y_ik = No`) %>%
  arrange(m, desc(`X1_ik`)) %>%
  mutate_if(is.numeric, ~replace_na(., 0))
```

```
## `summarise()` has grouped output by 'm', 'X1_ik'. You can override using the
## `.groups` argument.
```

```
imp_data_test
```

```
## # A tibble: 20 x 4
##        m X1_ik       `Y_ik = Yes` `Y_ik = No`
##    <int> <chr>              <int>       <int>
##  1     1 X1_ik = Yes           28        1733
##  2     1 X1_ik = No             0        1591
##  3     2 X1_ik = Yes           28        1733
##  4     2 X1_ik = No             0        1591
##  5     3 X1_ik = Yes           28        1733
##  6     3 X1_ik = No             0        1591
##  7     4 X1_ik = Yes           28        1733
##  8     4 X1_ik = No             0        1591
##  9     5 X1_ik = Yes           28        1733
## 10     5 X1_ik = No             0        1591
## 11     6 X1_ik = Yes           28        1733
## 12     6 X1_ik = No             0        1591
## 13     7 X1_ik = Yes           28        1733
## 14     7 X1_ik = No             0        1591
## 15     8 X1_ik = Yes           28        1733
## 16     8 X1_ik = No             0        1591
## 17     9 X1_ik = Yes           28        1733
## 18     9 X1_ik = No             0        1591
## 19    10 X1_ik = Yes           28        1733
## 20    10 X1_ik = No             0        1591
```

```
# ------------------------------------------------------------------------------
# impute_once_ridge()
# ------------------------------------------------------------------------------
# Generates ONE completed data set by imputing missing X1_ik values.
# Workflow:
#     1)  complete-case logistic with stepwise (alpha = 0.10) to pick predictors
#     2)  ridge-penalised logistic on those predictors
#     3)  predict Pr(X1_ik = 1) for missing rows
#     4)  Bernoulli draw -> insert into a copy of the original data
#
# Arguments you might tweak:
#   seed          : reproducibility seed; NULL = no set.seed().
#   data          : data frame containing X1_ik and all predictors.
#   bootstrap     : TRUE = refit ridge on a bootstrap sample each call
#                   FALSE = use all complete rows (less between-imp variation).
#   lambda_sel    : "lambda.1se" or "lambda.min" (from cv.glmnet()).
#   lambda_factor : MULTIPLIER on the chosen lambda.
#                   • 1   = default ridge strength (baseline).
#                   • >1  = STRONGER penalty → coefficients shrink more →
#                           lower predicted p when Y=1 →
#                           MORE chance you'll impute X1_ik = 0
#                           (values 2-5 usually give the needed flexibility).
#                   • <1  = weaker penalty (approaches separation; rarely useful).
#
# ------------------------------------------------------------------------------
step_model_list <- list()
impute_once_ridge <- function(seed       = NULL,
                              myData,
                              #dat_missing,
                              #dat_complete,
                              bootstrap  = TRUE,
                              lambda_sel = c("lambda.1se", "lambda.min"),
```

```
                            lambda_factor = 1,
                            verbose    = FALSE) {

dat_missing <<- dplyr::filter(myData, is.na(X1_ik))
dat_complete <<- dplyr::filter(myData, !is.na(X1_ik))

if (!is.null(seed)) set.seed(seed)
lambda_sel <- match.arg(lambda_sel)

## --- 0. split ---------------------------------------------------
idx_miss     <- which(is.na(myData$X1_ik))
if (length(idx_miss) == 0) {stop("No missing X1_ik values left!")}

## --- 1. stepwise ------------------------------------------------
step_model <- suppressWarnings(
  stepAIC(glm(upper_form, family = binomial, data = dat_complete),
          scope = list(lower = lower_form, upper = upper_form),
          k = 4.6, trace = FALSE)
)
form_sel <- formula(step_model)

print(summary(step_model))

# --- 2. design for complete rows ------------------------------
X_full <- model.matrix(form_sel, dat_complete)   # include intercept
y_full <- dat_complete$X1_ik

## --- 3. bootstrap sample ----------------------------------------
if (bootstrap) {
  idx_b   <- sample(seq_len(nrow(dat_complete)), replace = TRUE)
  X_train <- X_full[idx_b, , drop = FALSE]
  y_train <- y_full[idx_b]
} else {
  X_train <- X_full
  y_train <- y_full
}
#
## --- 4. ridge fit -----------------------------------------------
cv_fit <- glmnet::cv.glmnet(X_train, y_train,
                            family = "binomial",
                            alpha  = 0,          # ridge
                            nfolds = 10,
                            type.measure = "deviance")

## pick lambda and allow user multiplier
lambda_pick <- cv_fit[[lambda_sel]] * lambda_factor

## --- 5. model matrix for missing rows ---------------------------
terms_mis <- delete.response(stats::terms(step_model))
attr(terms_mis, "na.action") <- NULL            # keep rows!
X_mis <- model.matrix(terms_mis, dat_missing)

need <- setdiff(colnames(X_train), colnames(X_mis))
if (length(need) > 0) {
  X_mis <- cbind(
    X_mis,
    matrix(0, nrow = nrow(X_mis), ncol = length(need),
           dimnames = list(NULL, need))
  )
```

```
  }
  X_mis <- X_mis[ , colnames(X_train), drop = FALSE]

  if (verbose) {
    message("[INFO] rows in X_mis: ", nrow(X_mis),
            " | cols: ", ncol(X_mis),
            " | lambda used: ", signif(lambda_pick, 4))
  }

  ## --- 6. predicted probabilities ------------------------------
  p_hat <- drop(predict(cv_fit,                  # <- generic predict()
                        newx = X_mis,
                        s    = lambda_pick,
                        type = "response"))

  ## --- 7. Bernoulli draw & insert ------------------------------
  x_imp <- rbinom(length(p_hat), 1, p_hat)
  data_out <- myData
  data_out$X1_ik[idx_miss] <- x_imp
  cat(paste0("Final predictive model used is: ", form_sel, "\n"))
  return(data_out)
}
```

```
# Model forms, least variables and most variables
lower_form <- X1_ik ~ Y_ik * T_k
upper_form <- X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Z2_k +
  Y_ik:T_k + Y_ik:X2_ik + Y_ik:Z1_k + Y_ik:Z2_k +
  T_k:Z1_k + T_k:Z2_k + T_k:X2_ik +
  X2_ik:Z1_k + X2_ik:Z2_k

# Variables needed in the imputation dataset
imp_vars <- c("cluster_id", "Y_ik", "X1_ik", "X2_ik", "T_k", "Z1_k", "Z2_k")

# Dataset to use
modelDat_Ind_impute <- modelDat_Ind[, imp_vars] %>%
  mutate(X1_ik = ifelse(X1_ik == "Yes", 1, ifelse(X1_ik == "No", 0, NA)))

#modelDat_Ind_missing <- filter(modelDat_Ind_impute, is.na(X1_ik))
#modelDat_Ind_complete <- filter(modelDat_Ind_impute, !is.na(X1_ik))

# Create 10 imputed datasets
imp_list <- lapply(1:10, function(j)
  impute_once_ridge(seed = 700 + j,
                    myData = modelDat_Ind_impute,
                    #dat_missing = modelDat_Ind_missing,
                    #dat_complete = modelDat_Ind_complete,
                    bootstrap = TRUE,
                    lambda_sel = "lambda.1se",
                    lambda_factor = 4))   # try 2-5 for more rare cases
```

```
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
```

```
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
```

```
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
##
## Call:
## glm(formula = X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k, family = binomial,
##     data = dat_complete)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.79280    0.22517  -7.962 1.69e-15 ***
## Y_ik          17.78071 1363.70105   0.013    0.990
## T_k            0.11102    0.09126   1.217    0.224
## X2_ik         17.71917  241.77981   0.073    0.942
## Z1_k           3.63299    0.58421   6.219 5.02e-10 ***
## Y_ik:T_k      -1.27629 2506.89924  -0.001    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4335.4  on 3127  degrees of freedom
## Residual deviance: 3868.5  on 3122  degrees of freedom
## AIC: 3880.5
##
## Number of Fisher Scoring iterations: 16
##
## Final predictive model used is: X1_ik ~ Y_ik + T_k + X2_ik + Z1_k + Y_ik:T_k
```

```r
# Ridge Regression Imputation Summary (10 datasets)
imp_data <- data.table::rbindlist(imp_list, idcol = "m") %>%
  mutate(X1_ik = ifelse(X1_ik == 1, "Yes", ifelse(X1_ik == 0, "No", NA)),
         Y_ik = ifelse(Y_ik == 1, "Yes", ifelse(Y_ik == 0, "No", NA))) %>%
  dplyr::select(m, X1_ik, Y_ik) %>%
  mutate(Y_ik = paste0("Y_ik = ", Y_ik)) %>%
  mutate(X1_ik = paste0("X1_ik = ", X1_ik)) %>%
  group_by(m, X1_ik, Y_ik) %>%
  dplyr::summarize(n = n()) %>%
  ungroup() %>%
  pivot_wider(names_from = Y_ik, values_from = n) %>%
  dplyr::select(m, `X1_ik`, `Y_ik = Yes`, `Y_ik = No`) %>%
  arrange(m, desc(`X1_ik`)) %>%
  mutate_if(is.numeric, ~replace_na(., 0))
```

```
## `summarise()` has grouped output by 'm', 'X1_ik'. You can override using the
## `.groups` argument.
```

m	X1_ik	Y_ik = Yes	Y_ik = No
1	X1_ik = Yes	23	1693
1	X1_ik = No	5	1631
2	X1_ik = Yes	21	1694
2	X1_ik = No	7	1630
3	X1_ik = Yes	25	1684
3	X1_ik = No	3	1640
4	X1_ik = Yes	24	1682
4	X1_ik = No	4	1642
5	X1_ik = Yes	28	1701
5	X1_ik = No	0	1623
6	X1_ik = Yes	25	1683
6	X1_ik = No	3	1641
7	X1_ik = Yes	24	1703
7	X1_ik = No	4	1621
8	X1_ik = Yes	25	1689
8	X1_ik = No	3	1635
9	X1_ik = Yes	25	1686
9	X1_ik = No	3	1638
10	X1_ik = Yes	24	1688
10	X1_ik = No	4	1636

Table 9: Counts of HIV Seroconversion (Y) by Male Circumcision (X1) for 10 Imputed Datasets

Data Characteristics

Variable	Level	Control	Treatment	Overall	Missing (Control)	Missing (Treatment)	Missin
Number of Individuals		1679	1673	3352			
Number of Clusters		15	15	30			
Mean Cluster Size		462	465	463.5			
Gender	Male	1679 (100%)	1673 (100%)	3352 (100%)			
HIV Status at Start	HIV-uninfected	1679 (100%)	1673 (100%)	3352 (100%)			
Treatment Component: MC	Yes	149 (10%)	238 (15%)	387 (12%)	138 (2%)	86 (1%)	224 (2
	No	868 (56%)	723 (46%)	1591 (51%)			
	Began study circumcised	524 (34%)	626 (39%)	1150 (37%)			
Treatment Component: HTC	Yes	235 (14%)	228 (14%)	463 (14%)			
	No	1444 (86%)	1445 (86%)	2889 (86%)			
Treatment Component: Full	Yes	107 (7%)	147 (9%)	254 (8%)	128 (2%)	81 (1%)	209 (2
	No	1444 (93%)	1445 (91%)	2889 (92%)			
Outcome: HIV Seroconversion (3-year period)	Yes	20 (1%)	8 (0%)	28 (1%)			
	No	1659 (99%)	1665 (100%)	3324 (99%)			

Table 10: Characteristics of Individual Effects Analysis Data

D. Mediator Model for Individual Effect of Treatment Assignment

"IndM" denotes individual effects, i.e. effects of a male's own treatment assignment on their own outcome. Here, we use the product method to calculate the direct and indirect effects, with the outcome being Y_{ik}, treatment being T_k, and mediator being $X_{ik}^{(1)}$, whether or not a male was circumcised..

The mediator model regresses the mediator on the exposure and confounders. Here, we block the spillover that exists through the proportion circumcised and proportion who received HTC in the cluster by controlling for it in the model (inclusion of $Z_k^{(1)}$ and $Z_k^{(2)}$ in the model). The abbreviation "IndM" refers to the individual effect mediator model, shown below.

$$logit(X_{ik}^{(1)}) = \beta_0^{\text{IndM}} + \beta_1^{\text{IndM}}(T_k) + \beta_2^{\text{IndM}}(Z_k^{(1)}) + \beta_3^{\text{IndM}}(Z_k^{(2)})$$

```
# Mediator Model for Individual Effect of Treatment Assignment

# Model not accounting for clustering

fits_IndM <- lapply(imp_list, function(d)
  glm(X1_ik ~ T_k + Z1_k + Z2_k,
                  family = binomial(link = 'logit'),
                  data = d))
# model_IndM <- glm(X1_ik ~ T_k + Z1_k + Z2_k,
#                   family = binomial(link = 'logit'),
#                   data = modelDat_Ind)

# Model accounting for clustering using GLMM
fits_IndM_glmm <- lapply(imp_list, function(d)
  glmer(X1_ik ~ T_k + Z1_k + Z2_k + (1|cluster_id), # Uses exchangeable
                         data = d,
                         family = binomial(link = "logit")))
# model_IndM_glmm <- glmer(X1_ik ~ T_k + Z1_k + Z2_k + (1|cluster_id), # Uses exchangeable
#                          data = modelDat_Ind,
#                          family = binomial(link = "logit"))

# Model accounting for clustering using GEE
fits_IndM_gee <- lapply(imp_list, function(d)
  geeglm(X1_ik ~ T_k + Z1_k + Z2_k,
                         family = binomial(link = "logit"),
                         id = cluster_id,
                         data = d,
                         corstr = "exchangeable"))
# model_IndM_gee <- geeglm(X1_ik ~ T_k + Z1_k + Z2_k,
#                          family = binomial(link = "logit"),
#                          id = cluster_id,
#                          data = modelDat_Ind,
#                          corstr = "exchangeable") # working correlation
```

```
# Pool the estimates by Rubin's rule
pool_rubins <- function(fit_list,
                        coef_fun = coef,          # how to grab betas
                        vcov_fun = vcov,          # how to grab Var(betas)
                        conf.level = 0.95) {
  m <- length(fit_list)
  if (m < 2) stop("Need at least 2 imputations to pool.")

  # 1. Stack coefficient vectors (m × p)
  beta_mat <- do.call(rbind, lapply(fit_list, coef_fun))

  # 2. Within-imputation variances
  U_list   <- lapply(fit_list, vcov_fun)

  # --- Rubin's rules ------------------------------------------------
```

```
  Q_bar <- colMeans(beta_mat)                    # pooled beta
  U_bar <- Reduce(`+`, U_list) / m               # pooled within-var
  centered <- sweep(beta_mat, 2, Q_bar)          # beta_j - Q
  B      <- t(centered) %*% centered / (m - 1)   # between-var
  T_mat  <- U_bar + (1 + 1/m) * B                # total var
  se     <- sqrt(diag(T_mat))

  z  <- Q_bar / se
  p  <- 2 * pnorm(abs(z), lower.tail = FALSE)
  alpha <- 1 - conf.level
  zcrit <- qnorm(1 - alpha/2)
  ci_l  <- Q_bar - zcrit * se
  ci_u  <- Q_bar + zcrit * se

  tibble(Term     = names(Q_bar),
             Estimate = Q_bar,
             SE       = se,
             `z Value`  = z,
             `p Value`  = p,
             `CI Lower`   = ci_l,
             `CI Upper`  = ci_u,
             row.names = NULL)
}

# GLM Results
pooled_results_IndM <- pool_rubins(fits_IndM) %>%
  mutate(Model = "GLM") %>%
  relocate(Model) %>%
  filter(Term != "(Intercept)") %>%
  mutate(`Mean ICC` = c(NA, NA, NA))

# GLMM Results
icc_IndM_glmm <- sapply(fits_IndM_glmm, function(f){
  out <- performance::icc(f, tolerance = 1e-10000)$ICC_adjusted[[1]]
})
icc_IndM_glmm_mean <- mean(icc_IndM_glmm)
icc_IndM_glmm_range <- range(icc_IndM_glmm)
pooled_results_IndM_glmm <- pool_rubins(fits_IndM_glmm,
                                        coef_fun = lme4::fixef) %>%
  mutate(Model = "GLMM") %>%
  relocate(Model) %>%
  filter(Term != "(Intercept)") %>%
  mutate(`Mean ICC` = c(icc_IndM_glmm_mean, NA, NA))

# GEE Results
alpha_IndM_gee <- sapply(fits_IndM_gee, function(f) f$geese$alpha[[1]])
alpha_mean_IndM_gee <- mean(alpha_IndM_gee)
alpha_range_IndM_gee <- range(alpha_IndM_gee)
pooled_results_IndM_gee <- pool_rubins(fits_IndM_gee,
                                       coef_fun = coef,
                                       vcov_fun = function(x) vcov(x,
                                                                   type = "robust")) %>%
  mutate(Model = "GEE") %>%
  relocate(Model) %>%
  filter(Term != "(Intercept)") %>%
  mutate(`Mean ICC` = c(alpha_mean_IndM_gee, NA, NA))

# Final Results Table
final_IndM_results_pooled <- rbind(pooled_results_IndM,
```

```
                                  pooled_results_IndM_glmm) %>%
  rbind(pooled_results_IndM_gee) %>%
  mutate(`OR [95% CI]` = paste0(round(exp(Estimate), 3),
                                " [", round(exp(`CI Lower`), 3), ", ",
                                round(exp(`CI Upper`), 3), "]")) %>%
  dplyr::select(Model, Term, `OR [95% CI]`, `p-value` = `p Value`, `Mean ICC`, Estimate)
```

Model	Term	OR [95% CI]	p-value	Mean ICC
GLM	T_k	1.07 [0.901, 1.271]	0.44	
GLM	Z1_k	36.061 [12.032, 108.074]	0.00	
GLM	Z2_k	5.095 [1.417, 18.317]	0.01	
GLMM	T_k	1.07 [0.901, 1.271]	0.44	0.00
GLMM	Z1_k	36.061 [12.043, 107.977]	0.00	
GLMM	Z2_k	5.095 [1.419, 18.298]	0.01	
GEE	T_k	1.042 [0.925, 1.175]	0.50	-0.00
GEE	Z1_k	39.438 [15.886, 97.902]	0.00	
GEE	Z2_k	5.741 [2.604, 12.661]	0.00	

Table 11: Mediator Models for Individual Effect of Treatment Assignment

E. Outcome Model for Individual Effect of Treatment Assignment

Now, we regress the outcome, Y_{ik} on the treatment assignment T_k and MC mediator $X_{ik}^{(1)}$. Here, we block the spillover that exists through the proportion circumcised and proportion who received HTC in the cluster by controlling for it in the model (inclusion of $Z_k^{(1)}$ and $Z_k^{(2)}$ in the model). The abbreviation "IndO" refers to the individual effect outcome model, shown below.

$$logit(Y_{ik}) = \beta_0^{\text{IndO}} + \beta_1^{\text{IndO}}(T_k) + \beta_2^{\text{IndO}}(X_{ik}^{(1)}) + \beta_3^{\text{IndO}}(Z_k^{(1)}) + \beta_4^{\text{IndO}}(Z_k^{(2)})$$

```
# Outcome Model for Individual Effect of Treatment Assignment

# Model not accounting for clustering
fits_IndO <- lapply(imp_list, function(d)
  glm(Y_ik ~ T_k + X1_ik + Z1_k + Z2_k,
                  family = binomial(link = 'logit'),
                  data = d))
# model_IndO <- glm(Y_ik ~ T_k + X1_ik + Z1_k + Z2_k,
#                   family = binomial(link = 'logit'),
#                   data = modelDat_Ind)

# Model accounting for clustering using GLMM
fits_IndO_glmm <- lapply(imp_list, function(d)
  glmer(Y_ik ~ T_k + X1_ik + Z1_k + Z2_k + (1|cluster_id), # Uses exchangeable
                         data = d,
                         family = binomial(link = "logit")))
# model_IndO_glmm <- glmer(Y_ik ~ T_k + X1_ik + Z1_k + Z2_k + (1|cluster_id), # Uses exchangeable
#                          data = modelDat_Ind,
#                          family = binomial(link = "logit"))

# Model accounting for clustering using GEE
fits_IndO_gee <- lapply(imp_list, function(d)
  geeglm(Y_ik ~ T_k + X1_ik + Z1_k + Z2_k,
                         family = binomial(link = "logit"),
                         id = cluster_id,
                         data = d,
                         corstr = "exchangeable"))
# model_IndO_gee <- geeglm(Y_ik ~ T_k + X1_ik + Z1_k + Z2_k,
#                          family = binomial(link = "logit"),
#                          id = cluster_id,
```

```r
#                           data = dplyr::select(modelDat_Ind, cluster_id, Y_ik,
#                                                T_k, X1_ik, Z1_k, Z2_k) %>% drop_na(),
#                           corstr = "exchangeable") # working correlation
```

```r
# GLM Results
pooled_results_Ind0 <- pool_rubins(fits_Ind0) %>%
  mutate(Model = "GLM") %>%
  relocate(Model) %>%
  filter(Term != "(Intercept)") %>%
  mutate(`Mean ICC` = c(NA, NA, NA, NA))

# GLMM Results
icc_Ind0_glmm <- sapply(fits_Ind0_glmm, function(f){
  out <- performance::icc(f, tolerance = 1e-10000)$ICC_adjusted[[1]]
})
icc_Ind0_glmm_mean <- mean(icc_Ind0_glmm)
icc_Ind0_glmm_range <- range(icc_Ind0_glmm)
pooled_results_Ind0_glmm <- pool_rubins(fits_Ind0_glmm,
                                        coef_fun = lme4::fixef) %>%
  mutate(Model = "GLMM") %>%
  relocate(Model) %>%
  filter(Term != "(Intercept)") %>%
  mutate(`Mean ICC` = c(icc_Ind0_glmm_mean, NA, NA, NA))

# GEE Results
alpha_Ind0_gee <- sapply(fits_Ind0_gee, function(f) f$geese$alpha[[1]])
alpha_mean_Ind0_gee <- mean(alpha_Ind0_gee)
alpha_range_Ind0_gee <- range(alpha_Ind0_gee)
pooled_results_Ind0_gee <- pool_rubins(fits_Ind0_gee,
                                       coef_fun = coef,
                                       vcov_fun = function(x) vcov(x,
                                                                   type = "robust")) %>%
  mutate(Model = "GEE") %>%
  relocate(Model) %>%
  filter(Term != "(Intercept)") %>%
  mutate(`Mean ICC` = c(alpha_mean_Ind0_gee, NA, NA, NA))

# Final Results Table
final_Ind0_results_pooled <- rbind(pooled_results_Ind0,
                                   pooled_results_Ind0_glmm) %>%
  rbind(pooled_results_Ind0_gee) %>%
  mutate(`OR [95% CI]` = paste0(round(exp(Estimate), 3),
                                " [", round(exp(`CI Lower`), 3), ", ",
                                round(exp(`CI Upper`), 3), "]")) %>%
  dplyr::select(Model, Term, `OR [95% CI]`, `p-value` = `p Value`, `Mean ICC`, Estimate)
```

F. Direct and Indirect Individual Effects of Treatment Assignment

The direct individual effect of treatment assignment on HIV seroconversion is β_1^{IndO}. The indirect individual effect of treatment assignment on HIV seroconversion is $\beta_1^{\text{IndM}} \times \beta_2^{\text{IndO}}$. Then, the total effect is the sum of these, namely $\beta_1^{\text{IndO}} + \beta_1^{\text{IndM}} \times \beta_2^{\text{IndO}}$.

G. Proportion of Individual Effect of Treatment Assignment Mediated by Circumcision

The proportion of the total individual effect that is mediated by circumcision can be calculated as:

$$\frac{\beta_1^{\text{IndM}} \times \beta_2^{\text{IndO}}}{\beta_1^{\text{IndO}} + \beta_1^{\text{IndM}} \times \beta_2^{\text{IndO}}}$$

This is the indirect effect (i.e. effect of treatment assignment through the MC mediator on the outcome) divided by the total effect of the treatment assignment on the outcome (both through and not through the mediator).

Model	Term	OR [95% CI]	p-value	Mean ICC
GLM	T_k	0.454 [0.155, 1.334]	0.15	
GLM	X1_ik	32.239 [0, 1.57533931522552e+192]	0.99	
GLM	Z1_k	0.142 [0, 119.408]	0.57	
GLM	Z2_k	275.908 [0.405, 187773.188]	0.09	
GLMM	T_k	0.457 [0.145, 1.436]	0.18	0.04
GLMM	X1_ik	39.143 [0, 2.850071972498e+30]	0.91	
GLMM	Z1_k	0.132 [0, 196.783]	0.59	
GLMM	Z2_k	359.68 [0.221, 584549.569]	0.12	
GEE	T_k	0.452 [0.181, 1.125]	0.09	0.00
GEE	X1_ik	29.759 [0.001, 1365442.318]	0.54	
GEE	Z1_k	0.145 [0.001, 41.082]	0.50	
GEE	Z2_k	272.667 [1.791, 41518.904]	0.03	

Table 12: Mediator Models for Individual Effect of Treatment Assignment

Model	Direct Effect	Indirect Effect	Total Effect
GLM	0.45	1.26	0.57
GLMM	0.46	1.28	0.59
GEE	0.45	1.15	0.52

Table 13: Direct, Indirect, and Total Individual Effects of Treatment Assignment (Shown as ORs)

The issue with the data still arises when using the product method. When using the product method, we still need to regress $Y_{ik} \sim X_{ik}^{(1)}$. It seems as if we either need to use a penalty model (like Firth's method that I did before) or potentially impute data to account for the fact that 0 participants have $X_{ik}^{(1)} = 0$ and $Y_{ik} = 1$ (see table below).

Model	Proportion Mediated
GLM	-0.42
GLMM	-0.46
GEE	-0.22

Table 14: Proportion of Individual Effect of Treatment Assignment Mediated by Circumcision

Overall Effects

H. Overall Intervention Village Effect

The overall effect of being in an intervention village can be calculated by just fitting the following model on the overall dataset of HIV-negative individuals (at the start of the study, n = 8551), without controlling for any other causal pathways.

$$logit(Y_{ik}) = \beta_0^{\text{Overall}} + \beta_1^{\text{Overall}}(T_k)$$

```
# Overall Effects of T_k on Y_ik

# Model not accounting for clustering
model_overall <- glm(Y_ik ~ T_k,
                     family = binomial(link = 'logit'),
                     data = modelDat) # Everyone

# Model accounting for clustering using GLMM
model_overall_glmm <- glmer(Y_ik ~ T_k + (1|cluster_id),# Uses exchangeable
                            data = modelDat_Ind,
                            family = binomial(link = "logit"))

# Model accounting for clustering using GEE
model_overall_gee <- geeglm(Y_ik ~ T_k,
                            family = binomial(link = "logit"),
                            id = cluster_id,
                            data = modelDat_Ind,
                            corstr = "exchangeable") # working correlation
```

Model	Term	OR [95% CI]	p-value	ICC
GLM	T_k	0.633 [0.451, 0.882]	0.01	
GLMM	T_k	0.396 [0.159, 0.985]	0.05	0.07
GEE	T_k	0.394 [0.182, 0.854]	0.02	0.00

Table 15: Overall Effect of Treatment Assignment on HIV Model Output

Thus, the OR is 0.63, meaning that for HIV-negative individuals at baseline, living in an intervention village is associated with a 37% reduction in the odds of seroconversion during follow-up compared with living in a control village. This single odds ratio blends every causal pathway, thus resulting in an overall effect. It is the total impact of the intervention environment on an average resident.

I. Proportion of total effect mediated by male circumcision The proportion of total effect mediated by male circumcision is

(Total Individual - Direct Individual) + (Total Spillover - Spillover Not Mediated by MC) / Overall (Or Total Individual + Total Spillover)

$$\frac{[(\beta_1^{\text{IndM}} \times \beta_2^{\text{IndO}}) + (\beta_1^{\text{SpW}} - \beta_1^{\text{SpWR}})]}{(\beta_1^{\text{Overall}})}$$

Model	Proportion of Overall Effect Mediated by Circumcision
GLM	0.04
GLMM	0.01
GEE	0.12

Table 16: Proportion of Overall Effect of Treatment Assignment Mediated by Circumcision

CURRENT ISSUES AND NOTES

Ashley's email:

1. It would be helpful to write out the estimands for each effect. Are these the same as Tyler's paper? Does it matter that your mediator is part of the intervention package, while Tyler's is another covariate? https://pmc.ncbi.nlm.nih.gov/articles/PMC3753117/
2. Are the meditators at the individual level, group level or both? If group level, are you using an exposure mapping function?
3. Can the spillover effect itself be mediated? In the case of just two people in a cluster, I believe the spillover problem can be exactly described as meditation.
4. How are the other package components handled? Could treat as confounders, averaging over them.
5. For the OR =22, I would check the table of village assignment, VMMC exposure and outcome, probably a zero or small cell. Ke, any other ideas here?
6. Are you concerned about mediator confounding in this case? How is the correlation within cluster modeled?
7. I think the positivity assumption is OK - as you have a three level exposure (women, male cirm, male unicorn) and fairly sure there are no gender restrictions on the covariates., but the "woman" status is not an intervention like VMMC is.

Notes from Laura

1. what we are doing is first estimating the total effect (which is a sum of the direct and indirect effects) by regressing on T.
    a. Laura's response: If regressing on T, this is the overall (total) effect, that is the overall effect of being in an intervention cluster compared to control. It's total because it's regardless of mediation.
2. then we are adjusting for x1 (vmmc) to get the direct effect of CP (assuming perfect compliance) and we can get the indirect effect through vmmc through the usual method we have in the slides.
    a. This is the overall indirect effect, that is the effect of cluster assignment through individual VMMC.

the new suggestion allows for imperfect compliance and this uses X12 in place of T.

`a. What is the new suggestion?`

Other Notes

1. Circumcision that is done locally, not for medical purposes but for cultural purposes, is incomplete and may not be effective for preventing HIV. So at some point later in the analysis, it would be of interest to assess the effects of circumcision before the study started, with circumcision after.
2. Positivity assumption was mentioned as a possible issue because women can't have VMMC
3. Add a dag for mediation, and a dag for spillover - has anybody ever done this?
4. Do we have data on death? We can combine HIV and death as another outcome to have more events
5. There is a variable called hiv_status_time that gives HIV status by each visit, with status already positive, new positive, negative. Do you see this? And then there is a variable hiv_results_days that is days from enrollment to that test. That could give us 1 year incidence when/if we want that (for survival data analysis for example). These should add up to the number of cases in total.
6. If we want to create a combined variable, death or seroconversion, which could give us more cases and more power, if all or most of the deaths are due to HIV.

7. Time from enrollment to death is death_days. There is also a variable death_cause (can we take a look at this? It might be possible to delete deaths that are obviously not HIV related, such as accidents). And then there is another variable death_primary, which is the primary cause of death.