Handling Missing Data in Partially Clustered Randomized Controlled Trials

Supplemental Materials

Table 1

Relative Bias and MSE of Average Treatment Effect Estimates, When X1 Effect Was Random, $\gamma_{10} = 0.8$*, and* $p_{miss} = 30\%$*, Stratified by the Type of Incomplete Variables and* θ

Incomplete Variables	Method	$\theta = 0.3$		$\theta = 1$		$\theta = 3$	
		% Relative Bias	MSE	% Relative Bias	MSE	% Relative Bias	MSE
Y and X1	CD	0.1	0.02	-0.2	0.02	-1.4	0.02
	LD	**-11**	0.03	-3.4	0.03	**10.5**	0.04
	MI-JM-SIM	-7.9	0.02	-3.1	0.02	4.5	0.03
	MI-JM-AS	-0.6	0.02	-0.9	0.03	-2	0.03
	MI-SMC-AS	-0.2	0.02	-0.5	0.03	-1.8	0.03
	SFB-NON	-6.5	0.02	-1.9	0.02	6.3	0.03
	SFB-WEAK	-6.9	0.02	-2.5	0.02	5.7	0.03
Y and X2	CD	0.1	0.02	-0.7	0.02	-0.2	0.02
	LD	**-10.5**	0.04	-6	0.04	4.5	0.04
	MI-JM-SIM	**-10.2**	0.03	-6.8	0.03	1.9	0.03
	MI-JM-AS	-0.9	0.02	-1.5	0.03	-0.5	0.03
	MI-SMC-AS	-1	0.02	-1.6	0.03	-0.9	0.03
	SFB-NON	-6.9	0.02	-3	0.02	6.9	0.03
	SFB-WEAK	-7.3	0.02	-3.5	0.03	6.3	0.03

Note: CD = complete-data (pre-deletion) analysis; LD = listwise deletion; MI-JM-SIM = simultaneous multiple imputation using joint modeling; MI-JM-AS = arm-specific multiple imputation using joint modeling; MI-SMC-AS = arm-specific multiple imputation using substantive-model-compatible sequential modeling; SFB-NON = sequential fully Bayesian estimation using non-informative priors; SFB-WEAK = sequential fully Bayesian estimation using weakly-informative priors; θ = ratio of the person-level residual variance in the control arm to that in the treatment arm; Relative Bias = the ratio of absolute bias to the true value in percentage format (with relative bias of 10% or greater in boldface); MSE = mean square error.

Table 2

% Relative Bias for Estimating Three Variance Components, When X1 Effect Was Fixed, $\gamma_{10} = 0.8$*, and* $p_{miss} = 30\%$

Incomplete	Method	σ^2_{u1}			$\sigma^2_{e_1}$			$\sigma^2_{e_0}$		
Variables		$\theta = 0.3$	1	3	$\theta = 0.3$	1	3	$\theta = 0.3$	1	3
Y and X1	CD	0.9	1.5	1.5	-0.1	-0.7	-0.3	-0.1	-0.2	-0.3
	LD	1.1	2.1	2.7	-3.9	-4.7	-4.2	-1.9	-3.7	-5.3
	MI-JM-SIM	**11.1**	7.8	2.7	**-14.3**	1.9	**42.2**	**51.5**	2.7	**-10.8**
	MI-JM-AS	7.1	7.6	8.1	8.6	6.9	7.1	**15.8**	6.7	4.6
	MI-SMC-AS	**40**	**40.1**	**40.3**	**35.2**	**34.5**	**34.8**	3.2	2.5	2.7
	SFB-NON	**252.3**	**252.4**	**252.8**	4.5	4.1	4.8	**29.8**	3.7	-3.5
	SFB-WEAK	**63.3**	**64.7**	**64.2**	2.3	1.6	2.5	4.9	0.2	-2.4
Y and X2	CD	1.7	1.7	1.4	-0.2	-0.5	-0.5	0	-0.4	0.2
	LD	**-12.6**	**-11.9**	**-11.7**	-1.9	-2.4	-2.1	-1.2	-3	-3.6
	MI-JM-SIM	7.6	4.3	-0.5	**-11.2**	1.6	**38.2**	**42.9**	2.1	-9.6
	MI-JM-AS	7.6	9	8.1	6.7	5.9	5.5	**10.7**	5.6	4.7
	MI-SMC-AS	**47.2**	**50.1**	**45.7**	**30.5**	**29.6**	**30.3**	2.5	2.4	3.4
	SFB-NON	**249.9**	**251.4**	**250.8**	7.3	7.1	7.3	**27.8**	5.6	-0.8
	SFB-WEAK	**47.5**	**48.3**	**47.8**	5.3	5	5.2	5.7	2.4	0.4

Note: CD = complete-data (pre-deletion) analysis; LD = listwise deletion; MI-JM-SIM = simultaneous multiple imputation using joint modeling; MI-JM-AS = arm-specific multiple imputation using joint modeling; MI-SMC-AS = arm-specific multiple imputation using substantive-model-compatible sequential modeling; SFB-NON = sequential fully Bayesian estimation using non-informative priors; SFB-WEAK = sequential fully Bayesian estimation using weakly-informative priors; σ^2_{u1} = the between-cluster residual variance of treatment effect; $\sigma^2_{e_1}$ = the within-cluster person-level residual variance in the treatment arm; $\sigma^2_{e_0}$ = the person-level residual variance in the control arm; θ = ratio of the person-level residual variance in the control arm to that in the treatment arm; Relative Bias = the ratio of absolute bias to the true value in percentage format (with relative bias of 10% or greater in boldface).

Table 3

% Relative Bias for Estimating the Within-Cluster Person-Level Residual Variance in the Treatment Arm $\sigma^2_{e_1}$, When X1 Effect Was Fixed, $\gamma_{10} = 0.8$, and $p_{miss} = 30\%$, Stratified by c, m, and the Type of Incomplete Variables

Incomplete	Method	$c = 4$			$c = 8$			$c = 16$		
Variables		m=5	15	30	m=5	15	30	m=5	15	30
Y and X1	CD	-2.8	0	0.3	-0.6	0.3	-0.1	0	-0.1	-0.2
	LD	**-10.8**	-3.9	-2.7	-5.5	-2.8	-2.8	-3.6	-3.1	-3.2
	MI-JM-SIM	**16.6**	**10.6**	8.9	**12.8**	8.7	7.6	9.5	7.8	7.1
	MI-JM-AS	**28.2**	8	4	**14.2**	3.7	1.7	6	1.7	0.6
	MI-SMC-AS	**234.4**	**15.7**	7	**33.1**	6.2	3	**10.2**	2.7	1.4
	SFB-NON	**28.6**	6	0.8	**11.2**	0.4	-2.1	1.4	-2.4	-3.6
	SFB-WEAK	**15.2**	2.3	-1	8.9	-0.6	-2.8	3.7	-2.6	-3.9
Y and X2	CD	-2.2	-0.5	0.2	-0.2	-0.1	0	-0.6	-0.1	-0.1
	LD	-4.6	-2	-1.2	-3.1	-1.5	-1.2	-2.8	-1.4	-1.4
	MI-JM-SIM	**15.4**	**10.4**	8.8	**11.7**	8.2	7.6	8.9	7.5	7.1
	MI-JM-AS	**24.5**	5.9	3.3	**11.1**	2.4	1.4	4.2	1.1	0.5
	MI-SMC-AS	**204.1**	**13.8**	5.4	**30.8**	4.6	2.3	7.6	1.8	0.8
	SFB-NON	**31.8**	8.4	3.9	**13.4**	3	1	3.6	0.5	-0.4
	SFB-WEAK	**20.2**	4.9	2.2	**11.3**	2.1	0.4	5.9	0.3	-0.7

Note: CD = complete-data (pre-deletion) analysis; LD = listwise deletion; MI-JM-SIM = simultaneous multiple imputation using joint modeling; MI-JM-AS = arm-specific multiple imputation using joint modeling; MI-SMC-AS = arm-specific multiple imputation using substantive-model-compatible sequential modeling; SFB-NON = sequential fully Bayesian estimation using non-informative priors; SFB-WEAK = sequential fully Bayesian estimation using weakly-informative priors; c = number of clusters in the treatment arm; m = cluster size; Relative Bias = the ratio of absolute bias to the true value in percentage format (with relative bias of 10% or greater in bold).

Table 4

% Relative Bias for Estimating the Person-Level Residual Variance in the Unclustered Control Arm $\sigma^2_{e_0}$, When X1 Effect Was Fixed, $\gamma_{10} = 0.8$, and $p_{miss} = 30\%$, Stratified by c, m and the Type of Incomplete Variables

Incomplete	Method	$c = 4$			$c = 8$			$c = 16$		
Variables		m=5	15	30	m=5	15	30	m=5	15	30
Y and X1	CD	-1.6	0.2	-0.1	-0.1	0.3	0	-0.5	0	0.2
	LD	-4.3	-3.5	-3.7	-3.2	-3.8	-3.6	-4.3	-3.3	-3.1
	MI-JM-SIM	**27.4**	**14.9**	**12.5**	**16.8**	**12.5**	**11.5**	**11.9**	**11.4**	**11.3**
	MI-JM-AS	**37.2**	9.2	4.3	**14.9**	4.2	1.9	6	2.2	1.5
	MI-SMC-AS	9.6	3.3	1.5	5.1	1.5	0.6	1.7	0.9	0.8
	SFB-NON	**46.7**	**11.7**	3.4	**20.7**	3.4	-0.7	6.9	-0.3	-2.1
	SFB-WEAK	**15.2**	1.4	-1.7	5.1	-1.7	-3.2	-0.8	-2.9	-3.3
Y and X2	CD	-1.4	0.3	0.3	-0.5	0.2	0	0.4	0	0
	LD	-3.8	-1.8	-2.2	-3.3	-2.5	-2.4	-2.4	-2.5	-2.5
	MI-JM-SIM	**22.9**	**12.7**	**10.5**	**13.4**	9.8	9.3	**10**	8.9	8.7
	MI-JM-AS	**29.3**	7.7	3.5	**10.5**	3.2	1.7	5.2	1.6	0.7
	MI-SMC-AS	**10.4**	3.6	1.8	3.9	1.4	0.8	2.2	0.6	0.3
	SFB-NON	**45.5**	**13**	5	**19.5**	4.7	1.2	8.6	1.1	-0.7
	SFB-WEAK	**17.3**	4	0.5	5.7	0.3	-1	1.9	-1.1	-1.9

Note: CD = complete-data (pre-deletion) analysis; LD = listwise deletion; MI-JM-SIM = simultaneous multiple imputation using joint modeling; MI-JM-AS = arm-specific multiple imputation using joint modeling; MI-SMC-AS = arm-specific multiple imputation using substantive-model-compatible sequential modeling; SFB-NON = sequential fully Bayesian estimation using non-informative priors; SFB-WEAK = sequential fully Bayesian estimation using weakly-informative priors; c = number of clusters in the treatment arm; m = cluster size; Relative Bias = the ratio of absolute bias to the true value in percentage format (with relative bias of 10% or greater in bold).

Table 5

Type I Error Rate and Statistical Power for Examining the Average Treatment Effect When X1 Effect Was Fixed and $p_{miss} = 30\%$

Incomplete Variables	Method	$\theta = 0.3$		$\theta = 1$		$\theta = 3$	
		Type I Error	Power	Type I Error	Power	Type I Error	Power
Y and X1	CD	0.056	0.75	0.065	0.76	0.057	0.75
	LD	0.072	0.69	0.056	0.72	**0.086**	0.75
	MI-JM-SIM	**0.09**	0.83	0.073	0.82	0.071	0.80
	MI-JM-AS	0.071	0.83	0.071	0.81	0.058	0.74
	MI-SMC-AS	0.066	0.80	0.066	0.78	0.057	0.72
	SFB-NON	***0.006***	0.57	***0.007***	0.59	***0.019***	0.61
	SFB-WEAK	0.056	0.76	0.048	0.76	0.061	0.76
Y and X2	CD	0.052	0.75	0.058	0.76	0.059	0.74
	LD	**0.108**	0.61	**0.092**	0.65	**0.085**	0.68
	MI-JM-SIM	**0.083**	0.78	0.066	0.78	0.053	0.76
	MI-JM-AS	0.055	0.76	0.051	0.74	0.047	0.69
	MI-SMC-AS	0.067	0.75	0.065	0.73	0.059	0.68
	SFB-NON	***0.007***	0.52	***0.007***	0.54	***0.012***	0.56
	SFB-WEAK	0.053	0.70	0.045	0.71	0.056	0.72

Note: CD = complete-data (pre-deletion) analysis; LD = listwise deletion; MI-JM-SIM = simultaneous multiple imputation using joint modeling; MI-JM-AS = arm-specific multiple imputation using joint modeling; MI-SMC-AS = arm-specific multiple imputation using substantive-model-compatible sequential modeling; SFB-NON = sequential fully Bayesian estimation using non-informative priors; SFB-WEAK = sequential fully Bayesian estimation using weakly-informative priors; θ = ratio of the person-level residual variance in the control arm to that in the treatment arm; Type I error rates of 0.075 or higher are presented in boldface; Type I error rates of 0.025 or lower are presented in italic and boldface.

Table 6

Type I Error Rate and Statistical Power for Examining the Average Treatment Effect When X1 Effect Was Random and $p_{miss} = 30\%$

Incomplete Variables	Method	$\theta = 0.3$		$\theta = 1$		$\theta = 3$	
		Type I Error	Power	Type I Error	Power	Type I Error	Power
Y and X1	CD	0.049	1.00	0.051	1.00	0.051	0.99
	LD	**0.078**	0.99	0.052	0.99	**0.081**	0.98
	MI-JM-SIM	0.065	1.00	0.057	0.99	0.06	0.99
	MI-JM-AS	0.056	1.00	0.059	0.99	0.057	0.95
	MI-SMC-AS	0.054	1.00	0.052	0.99	0.054	0.96
	SFB-NON	0.026	1.00	0.03	0.99	0.04	0.98
	SFB-WEAK	0.067	1.00	0.058	0.99	0.062	0.99
Y and X2	CD	0.046	1.00	0.051	1.00	0.047	0.99
	LD	0.072	0.93	0.053	0.94	0.053	0.95
	MI-JM-SIM	**0.08**	0.99	0.064	0.98	0.051	0.98
	MI-JM-AS	0.051	0.99	0.053	0.98	0.052	0.96
	MI-SMC-AS	0.049	0.99	0.05	0.98	0.043	0.96
	SFB-NON	0.031	0.99	***0.022***	0.98	0.03	0.98
	SFB-WEAK	0.069	0.99	0.057	0.99	0.06	0.98

Note: CD = complete-data (pre-deletion) analysis; LD = listwise deletion; MI-JM-SIM = simultaneous multiple imputation using joint modeling; MI-JM-AS = arm-specific multiple imputation using joint modeling; MI-SMC-AS = arm-specific multiple imputation using substantive-model-compatible sequential modeling; SFB-NON = sequential fully Bayesian estimation using non-informative priors; SFB-WEAK = sequential fully Bayesian estimation using weakly-informative priors; θ = ratio of the person-level residual variance in the control arm to that in the treatment arm; Type I error rates of 0.075 or higher are presented in boldface; Type I error rates of 0.025 or lower are presented in italic and boldface.

Table 7

% Relative Bias for Estimating Parameters Using Weakly Informative Priors Based on Inverse Gamma and Inverse Wishart Distributions (SFB-WEAK2), When Y and $X1$ Were Incomplete, $\gamma_{10} = 0.8$, and $p_{miss} = 30\%$

Parameter	θ	c = 4, fixed			c = 8, fixed			c = 16, fixed			c = 16, random		
		$m = 5$	15	30	$m = 5$	15	30	$m = 5$	15	30	$m = 5$	15	30
γ_{10}	0.30	-7.4	-4.9	-5.7	-6.7	-6.4	-4.6	-5.6	-5.3	-5.8	-8.3	-7.3	-6.3
	1.00	-3.5	0.8	-1.8	-0.8	1.4	1	0.3	-0.1	0.3	-3.4	-1.9	-1.9
	3.00	**10.4**	9.8	9.1	9.8	**13.1**	**12.1**	**11**	**11.2**	**12.3**	7.4	8.3	8.3
σ^2_{u1}	0.30	**204.1**	**181.2**	**172.5**	**119.3**	**98.9**	**91**	**71.5**	**50**	**44.2**	**15.4**	**11.9**	**10.2**
	1.00	**204.2**	**181.1**	**172.4**	**120.1**	**95.3**	**87.5**	**70.7**	**51.3**	**46.3**	**16**	**13.5**	**10.6**
	3.00	**203.1**	**182.6**	**171.5**	**121.9**	**95.8**	**93**	**69.2**	**49.2**	**46.8**	**15.8**	**11.3**	**10.8**
$\sigma^2_{e_1}$	0.30	9.9	1.1	-2.4	0.4	-2.7	-3	-3.5	-3.7	-3.7	-1.7	-3.8	-3.7
	1.00	7.8	0.9	-2.1	1.2	-2.3	-3.8	-3.1	-3	-3.6	-1.9	-2.7	-3.6
	3.00	**11.3**	0.8	-1.9	2	-2.4	-2.9	-3.8	-3.1	-4	-0.7	-2.9	-3.7
$\sigma^2_{e_0}$	0.30	**81.2**	**27.2**	**11.5**	**39.6**	**11.5**	4.8	**19.4**	4.2	1.4	**27.2**	8	2.9
	1.00	7.2	-0.8	-2.8	0.2	-1.3	-3.5	-1.7	-3.2	-4	0.3	-2.7	-3.7
	3.00	**-13.6**	-7	-6	-7.9	-7	-6.5	-6.9	-6	-6.2	-6.5	-5.6	-6
σ^2_{u2}	0.30	—	—	—	—	—	—	—	—	—	**79.1**	**47.9**	**39.6**
	1.00	—	—	—	—	—	—	—	—	—	**79.6**	**50.1**	**39.6**
	3.00	—	—	—	—	—	—	—	—	—	**82.4**	**50.3**	**39.5**
σ_{u12}	0.30	—	—	—	—	—	—	—	—	—	**-55.8**	**-35.8**	**-29.1**
	1.00	—	—	—	—	—	—	—	—	—	**-54.8**	**-31.3**	**-27.7**
	3.00	—	—	—	—	—	—	—	—	—	**-52.1**	**-36.8**	**-29.3**

Note: fixed = X1 effect was fixed; random = X1 effect was random; c = number of clusters in the treatment arm; m = cluster size; γ_{10} = average treatment effect; σ^2_{u1} = the between-cluster residual variance of treatment effect; σ^2_{u2} = the between-cluster variance of X1 effect in the treatment arm; σ_{u12} = the covariance between random treatment effect and random X1 effect in the treatment arm; $\sigma^2_{e_1}$ = the within-cluster person-level residual variance in the treatment arm; $\sigma^2_{e_0}$ = the person-level residual variance in the control arm; θ = ratio of the person-level residual variance in the control arm to that in the treatment arm; Relative Bias = the ratio of absolute bias to the true value in percentage format (with relative bias of 10% or greater in boldface).

Table 8

% Relative Bias for Estimating Parameters Using Weakly Informative Priors Based on Inverse Gamma and Inverse Wishart Distributions (SFB-WEAK2), When Y and $X2$ Were Incomplete, $\gamma_{10} = 0.8$, and $p_{miss} = 30\%$

Parameter	θ	c = 4, fixed			c = 8, fixed			c = 16, fixed			c = 16, random		
		$m = 5$	15	30	$m = 5$	15	30	$m = 5$	15	30	$m = 5$	15	30
γ_{10}	0.30	**-10.4**	-9.1	**-10.8**	-8	-5.8	-6.6	-5.3	-6.2	-6.5	-6.8	-7.4	-7.2
	1.00	-1.2	-4.5	-3.6	-1.4	-0.9	-1.7	0.2	-0.2	-0.4	-2.6	-1.7	-3.1
	3.00	5.6	8.3	5.1	9.1	9.6	**11.9**	**10.8**	**10.3**	**10.6**	7.2	6.7	7.4
σ^2_{u1}	0.30	**195.3**	**173.7**	**171.1**	**119.1**	**97.3**	**89.4**	**67.3**	**48.9**	**43.5**	**14**	**10.9**	**10.2**
	1.00	**196**	**175.1**	**170.4**	**119.1**	**94.8**	**91.8**	**67.5**	**50.5**	**44.1**	**16.3**	**11**	9.3
	3.00	**196.4**	**172.9**	**170.9**	**115.2**	**98.3**	**89.8**	**69.5**	**48.4**	**44.7**	**17.9**	**10.2**	**10.7**
$\sigma^2_{e_1}$	0.30	**10.5**	3.8	-0.4	3.5	-0.2	-0.3	-0.1	-0.5	-1	2.3	-0.5	-1
	1.00	**12.3**	3.6	1.1	4.1	-0.3	-0.5	1.4	-0.5	-1.3	2.8	-0.3	-1
	3.00	**11.1**	3.3	0.8	4.3	1.3	-0.5	-0.7	-0.3	-0.8	0.7	0.2	-1.2
$\sigma^2_{e_0}$	0.30	**76.3**	**23.7**	**10.4**	**37.8**	**11.5**	4.6	**18.3**	4.7	1.9	**24.9**	7.9	3.5
	1.00	**10.9**	2.6	0	4.5	-0.4	-1.4	1.5	-1.1	-1.8	3	-0.8	-1.2
	3.00	-7.4	-5.2	-4.2	-7.9	-4.6	-4.6	-4.6	-4.2	-3.8	-4.2	-3.5	-4
σ^2_{u2}	0.30	—	—	—	—	—	—	—	—	—	**82.9**	**51.8**	**42.9**
	1.00	—	—	—	—	—	—	—	—	—	**82.3**	**52.3**	**41.6**
	3.00	—	—	—	—	—	—	—	—	—	**84.2**	**51.7**	**44**
σ_{u12}	0.30	—	—	—	—	—	—	—	—	—	**-45.7**	**-28.6**	**-25.1**
	1.00	—	—	—	—	—	—	—	—	—	**-46.2**	**-30.4**	**-26.6**
	3.00	—	—	—	—	—	—	—	—	—	**-44.6**	**-29.9**	**-24.5**

Note: fixed = X1 effect was fixed; random = X1 effect was random; c = number of clusters in the treatment arm; m = cluster size; γ_{10} = average treatment effect; σ^2_{u1} = the between-cluster residual variance of treatment effect; σ^2_{u2} = the between-cluster variance of X1 effect in the treatment arm; σ_{u12} = the covariance between random treatment effect and random X1 effect in the treatment arm; $\sigma^2_{e_1}$ = the within-cluster person-level residual variance in the treatment arm; $\sigma^2_{e_0}$ = the person-level residual variance in the control arm; θ = ratio of the person-level residual variance in the control arm to that in the treatment arm; Relative Bias = the ratio of absolute bias to the true value in percentage format (with relative bias of 10% or greater in boldface).

R and SAS code for MI-JM-SIM, MI-JM-AS, and MI-SMC-AS

Step 1: R code for the *Imputation Phase*

```
##############################################################################
# The csv data file "mydata.csv" includes the following variables:
# id:          the id number for each individual
# group_id: the id number for each cluster (i.e., intervention group), all the individuals in the control arm are
#              considered as one cluster with the same group_id
# TREAT:   the binary indicator of treatment assignment (1=treatment arm, 0=control arm)
# y:           the outcome variable that is subject to missingness
# x1:          a level-1 covariate that is subject to missingness
# x2:          a level-2 covariate that is subject to missingness
# a1:          a level-1 auxiliary variable that determines the probability of missingness for x1 and y
# a2:          a level-2 auxiliary variable that determines the probability of missingness for x2 and y
# for example, in our empirical dataset, y is the Multidimensional Health Locus of Control (MHLC) Internal
# subscale score at the 15 months follow-up (INT15), x1 is the MHLC Internal subscale score at baseline
# (INT0), x2 is the years of intervention experience that the co-facilitator in a PTSC group had (FACEXP),
# a1 is the stress symptoms score at baseline (STS0), and there was not an available a2.
##############################################################################

#=========================================================#
#     Simultaneous Multiple Imputation Using Joint Modeling (MI-JM-SIM)
#=========================================================#
library(jomo)  # package needed for joint modeling MI
indata <- read.csv("mydata.csv")  # read in the original data with missing values
indata$cons <- 1    # create a column of 1s for the intercept, which is needed in MI

#--- 1. specify variables/parameters for multiple imputation ---#
# specify level-1 response variables in the joint imputation model
# include auxiliary variables, outcome, and covariates that are subject to missingness
mi_y  <- indata[, c("x1 ","a1","y ")]
# specify level-2 response variables in the joint imputation model
# include auxiliary variables and covariates that are subject to missingness
mi_y2 <- indata[, c("x2 ","a2")]
# specify fully observed predictors in the imputation model
# if an intercept term is wanted in the imputation model, a column of 1 is needed
mi_x <- indata[, c("cons", "TREAT")]
# specify fully observed predictors that are associated to random effects in the joint imputation model
# for partially clustered designs, only random treatment effect is included in the model
mi_z <- indata[, c("TREAT "), drop=FALSE]
# specify the cluster indicator
mi_clus <- indata[, c("group_id"), drop=FALSE]
# number of imputed datasets
mi_nimp <- 20
```

```
#--- 2. create imputed datasets ---#
mi_impdata0 <- jomo2com(Y.con = mi_y, Y2.con = mi_y2, X = mi_x, Z=mi_z, clus=mi_clus, nburn = 1000,
nbetween = 1000, nimp = mi_nimp)
mi_impdata <- subset(mi_impdata0, select = -TREAT) # remove a redundant column of TREAT
mi_impdata$group_id <- mi_impdata$clus
# keep the 20 sets of imputed data and remove the original data with missing values
mi_impdata_out <- subset(mi_impdata, select=c(id,group_id,TREAT,x1,a1,x2,a2,y,Imputation), Imputation !=
0)

#--- 3. save imputed datasets in a csv file ---#
write.csv(mi_impdata_out, file="mydata_MIJMSIM.csv ")

#=========================================================#
#     Arm-Specific Multiple Imputation Using Joint Modeling (MI-JM-AS)
#=========================================================#
library(jomo)  # package needed for joint modeling MI
indata <- read.csv("mydata.csv")  # read in the original data with missing values
indata$cons <- 1    # create a column of 1s for the intercept, which is needed in MI

#=== 1. divide original data into two subsets for the two arms ===#
#--- 1a. specify variables/parameters for multiply imputing treatment arm data ---#
# specify level-1 response variables in the joint imputation model
# include auxiliary variables, outcome, and covariates that are subject to missingness
mi_y_t    <- indata[indata$TREAT ==1, c("x1","a1","y")]
# specify level-2 response variables in the joint imputation model
# include auxiliary variables and covariates that are subject to missingness
mi_y2_t   <- indata[indata$TREAT ==1, c("x2 ","a2")]
# specify fully observed predictors in the imputation model
# for the treatment arm, only intercept is included in the model
mi_x_t    <- indata[indata$TREAT ==1, c("cons"), drop=FALSE]
# specify full observed predictors associated to random effects in the joint imputation model
# for the treatment arm, only random intercept is included in the model
mi_z_t    <- indata[indata$TREAT==1, c("cons"), drop=FALSE]
# specify the cluster indicator
mi_clus_t <- indata[indata$TREAT==1, c("group_id"), drop=FALSE]

#--- 1b. specify variables/parameters for multiply imputing control arm data ---#
# specify response variables in the joint imputation model
# include auxiliary variables, outcome, and covariates that are subject to missingness
mi_y_c    <- indata[indata$TREAT==0, c("x1","a1","x2","a2","y")]
# specify fully observed predictors in the imputation model
# for the control arm, only intercept is included in the model
mi_x_c    <- indata[indata$TREAT==0, c("cons"), drop=FALSE]

mi_nimp <- 20  # number of imputed datasets
```

```
#=== 2. create imputed datasets (for treatment arm and control arm separately) ===#
mi_impdata_t <- jomo2com(Y.con = mi_y_t, Y2.con = mi_y2_t, X = mi_x_t, Z=mi_z_t, clus=mi_clus_t, nburn
= 1000, nbetween = 1000, nimp = mi_nimp)
mi_impdata_c <- jomo1con(Y = mi_y_c, X = mi_x_c, nburn = 1000, nbetween = 1000, nimp = mi_nimp)

#=== 3. combine imputed datasets from treatment arm and control arm ===#
#--- 3a. clean imputed treatment arm data ---#
mi_impdata_t$TREAT <- 1
mi_impdata_t <- subset(mi_impdata_t, select = -cons) # remove a redundant column of “cons”
mi_impdata_t <- mi_impdata_t[,c("x1","a1","x2","a2","y","cons","TREAT","clus","id","Imputation")]
# the “id” variable is automatically generated when imputing data, to differentiate id for the treatment and
control arm, recode id in the treatment arm by adding each id by 20000
mi_impdata_t$id <- mi_impdata_t$id + 20000

#--- 3b. clean imputed control arm data ---#
mi_impdata_c$TREAT <- 0
mi_impdata_c$clus <- as.factor(0) # assign a cluster id of 0 to every individual in the control arm
mi_impdata_c <- mi_impdata_c[,c("x1","a1","x2","a2","y","cons","TREAT","clus","id","Imputation")]

#--- 3c. combine imputed treatment arm and control arm data ---#
mi_impdata <- rbind(mi_impdata_t, mi_impdata_c)
mi_impdata$group_id <- as.factor(mi_impdata$clus)
# keep the 20 sets of imputed data and remove the original data with missing values
mi_impdata_out <- subset(mi_impdata, select=c(id,group_id,TREAT,x1,a1,x2,a2,y,Imputation), Imputation !=
0)

#=== 4. save imputed datasets in a csv file ===#
write.csv(mi_impdata_out, file="mydata_MIJMAS.csv ")

#=========================================================#
#    Arm-Specific Multiple Imputation Using Substantive-Model-Compatible
#    Sequential Modeling (MI-SMC-AS)
#=========================================================#
library(mdmb)  # package needed for SMC sequential modeling MI
indata <- read.csv("mydata.csv")  # read in the original data with missing values
# divide original data into two subsets for the two arms
indata_t <- subset(indata, select=c(id,group_id,TREAT,x1,a1,x2,a2,y), TREAT==1)
indata_c <- subset(indata, select=c(id,group_id,TREAT,x1,a1,x2,a2,y), TREAT==0)

#=== 1. conduct multiple imputation for treatment arm ===#
#--- 1a. define outcome model ---#
# select to use one of the following two lines depending on the effect of x1 on y
y_t.fml <- y ~ 1 + x1 + x2 + a1 + a2 + (1 + x1|group_id)  # use this line if x1 effect is random
y_t.fml <- y ~ 1 + x1 + x2 + a1 + a2 + (1|group_id)       # use this line if x1 effect is fixed
```

```
#--- 1b. define covariate models ---#
# here auxiliary variables are assumed to be fully observed, but code could be modified to define models
# for partially missing auxiliary variables
x1_t.fml <- x1 ~ 1 + x2 + a1 + a2
x2_t.fml <- x2 ~ 1 + a1 + a2

#--- 1c. define types of outcome and covariate models ---#
# here x1 is assumed to have variation only at level 1, so a simple linear regression model “linreg” is specified.
# if x1 has variations at multiple levels, a multilevel model “mlreg” needs to be used
y_t.mod  <- list(model="mlreg",  outcome="normal", formula=y_t.fml, sampling_level="group_id")
x1_t.mod <- list(model="linreg", outcome="normal", formula=x1_t.fml)
x2_t.mod <- list(model="linreg", outcome="normal", formula=x2_t.fml, variable_level="group_id ")

#--- 1d. combine all covariate models into a single predictor model ---#
# select to use one of the following two lines
pred_t.mod <- list(x1 = x1_t.mod)                       # use this line if x2 is not missing
pred_t.mod <- list(x1 = x1_t.mod, x2 = x2_t.mod)  # use this line if x2 is partially missing

#--- 1e. create imputed datasets ---#
imp_t.mdmb <- frm_fb(dat=indata_t, dep=y_t.mod, ind=pred_t.mod, burnin = 1000, iter = 1000+5000, Nimp =
20, print_iter=250, aggregation = TRUE)
mi_impdatalist_t <- frm2datlist(imp_t.mdmb)
mi_impdata_t <- do.call("bind_rows", c(mi_impdatalist_t, list(.id="Imputation")))

#=== 2. conduct multiple imputation for control arm ===#
#--- 2a. define outcome model ---#
y_c.fml <- y ~ 1 + x1 + a1

#--- 2b. define covariate models ---#
# here auxiliary variables are assumed to be fully observed, but code could be modified to define models
# for partially missing auxiliary variables
x1_c.fml <- x1 ~ 1 + a1

#--- 2c. define types of outcome and covariate models ---#
y_c.mod  <- list(model="linreg", outcome="normal", formula=y_c.fml)
x1_c.mod <- list(model="linreg", outcome="normal", formula=x1_c.fml)

#--- 2d. combine all covariate models into a single predictor model ---#
pred_c.mod <- list(x1 = x1_c.mod)

#--- 2e. create imputed datasets ---#
imp_c.mdmb <- frm_fb(dat=indata_c, dep=y_c.mod, ind=pred_c.mod, burnin = 1000, iter = 1000+5000, Nimp
= 20, print_iter=250, aggregation = TRUE)
mi_impdatalist_c <- frm2datlist(imp_c.mdmb)
mi_impdata_c <- do.call("bind_rows", c(mi_impdatalist_c, list(.id="Imputation")))
```

```
#=== 3. combine imputed datasets from treatment arm and control arm ===#
mi_impdata <- rbind(mi_impdata_t, mi_impdata_c)
mi_impdata_out <- subset(mi_impdata, select=c(id,group_id,TREAT,x1,a1,x2,a2,y,Imputation))

#=== 4. save imputed datasets in a csv file ===#
write.csv(mi_impdata_out, file="mydata_MISMCAS.csv ")
```

Step 2: SAS code for the *Analysis and Pooling Phase*

```
/*************************************************************************************/
/*  Conduct inference analysis and pooling using imputed datasets generated in R
/*  1. "mydata_MIJMSIM.csv " includes 20 imputed datasets using MI-JM-SIM
/*  2. "mydata_MIJMAS.csv " includes 20 imputed datasets using MI-JM-AS
/*  3. "mydata_MISMCAS.csv " includes 20 imputed datasets using MI-SMC-AS
/*  The analysis and pooling process is the same for the three multiple imputation methods (MI-JM-SIM, MI-
/*  JM-AS, MI-SMC-AS)
/*************************************************************************************/
*--- read in imputed datasets based on a particular multiple imputation method ---;
*run the following line if using MI-JM-SIM;
proc import datafile="C:\ mydata_MIJMSIM.csv" out= indata_mi dbms=CSV replace; run;
*run the following line if using MI-JM-AS;
proc import datafile="C:\ mydata_MIJMAS.csv" out= indata_mi dbms=CSV replace; run;
*run the following line if using MI-SMC-AS;
proc import datafile="C:\ mydata_MISMCAS.csv" out= indata_mi dbms=CSV replace; run;

/*--- 1. clean and sort imputed datasets ---*/
*SAS requires a variable "_Imputation_" that indicates the id of imputed dataset;
data indata_mi_v0; set indata_mi;_Imputation_ = Imputation; run;
proc sort data=indata_mi_v0 out=indata_mi_sort; by _Imputation_ TREAT group_id; run;

/*--- 2. analyze each imputed dataset ---*/
# select to use one of the following two lines depending on the effect of x1 on y
%let ranstat=TREAT;      *use this line if x1 effect is fixed;
%let ranstat=TREAT x1; *use this line if x1 effect is random;

proc mixed data=indata_mi_sort method=reml;
        class group_id;
        model y = TREAT x1 TREAT*x2 /s ddfm=kr cl;
        random &ranstat /subject=group_id type=un;
        repeated /group=TREAT;
        by _Imputation_;
        ods output SolutionF=mixparms_mi CovParms=varparms_mi;
run;

/*--- 3. obtain overall estimates of the variance components ---*/
/*       by averaging across estimates from 20 imputed datasets */
*between-cluster variance of treatment effect in treatment arm (sigma2_u1);
proc sql; select mean(Estimate) into: sigma2_u1 from varparms_mi
            where compress(CovParm) = "UN(1,1)"; quit;
*between-cluster variance of x1 effect in treatment arm (sigma2_u2);
*use the following proc sql code only if the effect of x1 on y is random;
```

```
proc sql; select mean(Estimate) into: sigma2_u2 from varparms_mi
          where compress(CovParm) = "UN(2,2)"; quit;
*covariance between two random effects (i.e., treatment effect and x1 effect) in treatment arm (sigma_u12);
*use the following proc sql code only if the effect of x1 on y is random;
proc sql; select mean(Estimate) into: sigma_u12 from varparms_mi
          where compress(CovParm) = "UN(2,1)"; quit;

*within-cluster residual variance in treatment arm (sigma2_e1);
proc sql; select mean(Estimate) into: sigma2_e1 from varparms_mi
          where compress(Group)  = "Group2"; quit;
*residual variance in control arm (sigma2_e0);
proc sql; select mean(Estimate) into: sigma2_e0 from varparms_mi
          where compress(Group)  = "Group1"; quit;

/*--- 4. obtain overall estimates of the treatment effect ---*/
/*       by pooling over 20 imputed datasets                 */
proc mianalyze parms=mixparms_mi;
        modeleffects TREAT;
        ods output ParameterEstimates=mi_results1;
run;
```

R and JAGS code for SFB

Step 1: JAGS script for SFB analysis

```
################################################################################
# Create a text file "jags_model.txt" with the following script.
# Since non-informative priors are not recommended for analyzing partially clustered data, the example script
# specifies weakly-informative priors for all parameters and uses a separation strategy when specifying
# priors for variance components
################################################################################

model {
        #--- 1. Specify the inferential analysis model separately for treatment and control arms ---#
        # The variables y_t, y_c, x1_t, x1_c, a1_t, a1_gm_t, a1_c, x2_t, x2_c are specified in R code in Step 2 #
        for (i in 1:nsubj_t) {
            # select to use one of the following two lines depending on x1 effect
            mu.y_t[i] <- mean_yc + u[grp_t[i],1] + beta_x1*x1_t[i]          # use this line if x1 effect is fixed
            mu.y_t[i] <- mean_yc + u[grp_t[i],1] + u[grp_t[i],2]*x1_t[i]   # use this line if x1 effect is random
            y_t[i] ~ dnorm(mu.y_t[i],tau.y[2])
            }
        for (i in 1:nsubj_c) {
            mu.y_c[i] <- mean_yc + beta_x1*x1_c[i]
            y_c[i] ~ dnorm(mu.y_c[i],tau.y[1])
            }
        for (j in 1:ngrp_t) {
            # select to use one of the following two lines depending on x1 effect
            u[j,1] ~ dnorm(mu.u[j,],tau.u)           # use this line if x1 effect is fixed
            u[j,1:2] ~ dmnorm(mu.u[j,],tau.u)     # use this line if x1 effect is random
            mu.u[j,1] <- beta_y_tr + beta_x2*x2_t[j]
```

```
    mu.u[j,2] <- beta_x1                    # add this line if x1 effect is random
    }

#--- 2. Specify the prior distributions of parameters in the inferential analysis model ---#
mean_yc  ~ dnorm(0, 0.04)
beta_y_tr ~ dnorm(0, 0.04)
beta_x1   ~ dnorm(0, 0.04)
beta_x2   ~ dnorm(0, 0.04)

sd_u_trt ~ dt(0, 4, 1)I(0,)
var_u_trt <- pow(sd_u_trt,2)
sd_u_x1t ~ dt(0, 4, 1)I(0,)                  # add this line if x1 effect is random
var_u_x1t <- pow(sd_u_x1t,2)                 # add this line if x1 effect is random
rho.u ~ dunif(-1,1)                          # add this line if x1 effect is random
cov_u_trx1t <- rho.u * sd_u_trt * sd_u_x1t   # add this line if x1 effect is random
Sigma.u[1, 1] <- var_u_trt                   # add this line if x1 effect is random
Sigma.u[2, 2] <- var_u_x1t                   # add this line if x1 effect is random
Sigma.u[2, 1] <- cov_u_trx1t                 # add this line if x1 effect is random
Sigma.u[1, 2] <- cov_u_trx1t                 # add this line if x1 effect is random
# select to use one of the following two lines depending on x1 effect
tau.u    <- 1/var_u_trt                      # use this line if x1 effect is fixed
tau.u[1:2, 1:2] <- inverse(Sigma.u[1:2, 1:2])   # use this line if x1 effect is random

sd_e_yc  ~ dt(0, 1, 1)I(0,)
sd_e_yt  ~ dt(0, 1, 1)I(0,)
var_e_yc  <- pow(sd_e_yc,2)
var_e_yt  <- pow(sd_e_yt,2)
tau.y[1] <- 1/var_e_yc
tau.y[2] <- 1/var_e_yt

#--- 3. Specify covariate models ---#
# here auxiliary variables are assumed to be fully observed, but script could be modified to define
# models for partially missing auxiliary variables
for (i in 1:nsubj_t) {
        mu.x1_t[i] <- alpha.x1[1] + alpha.x1[2]*x2_t[grp_t[i]] + alpha.x1[3]*a1_t[i] +
                        alpha.x1[4]*a2_t[grp_t[i]]
        x1_t[i] ~ dnorm(mu.x1_t[i],tau.x1_t)
    }
 for (i in 1:nsubj_c) {
        mu.x1_c[i] <- alpha.x1[5] + alpha.x1[6]*a1_c[i]
        x1_c[i] ~ dnorm(mu.x1_c[i],tau.x1_c)
    }
  for (j in 1:ngrp_t) {
        mu.x2_t[j] <- alpha.x2[1] + alpha.x2[2]*a1_gm_t[j] + alpha.x2[3]*a2_t[j]
        x2_t[j] ~ dnorm(mu.x2_t[j],tau.x2)
    }
```

```
#--- 4. Specify the prior distributions of parameters in the covariate models ---#
alpha.x1[1] ~ dnorm(0, 0.04)
alpha.x1[2] ~ dnorm(0, 0.04)
alpha.x1[3] ~ dnorm(0, 0.04)
alpha.x1[4] ~ dnorm(0, 0.04)
alpha.x1[5] ~ dnorm(0, 0.04)
alpha.x1[6] ~ dnorm(0, 0.04)
alpha.x2[1] ~ dnorm(0, 0.04)
alpha.x2[2] ~ dnorm(0, 0.04)
alpha.x2[3] ~ dnorm(0, 0.04)
sd_x1_t ~ dt(0, 1, 1)I(0,)
sd_x1_c ~ dt(0, 1, 1)I(0,)
sd_x2   ~ dt(0, 1, 1)I(0,)
tau.x1_t <- 1/pow(sd_x1_t,2)
tau.x1_c <- 1/pow(sd_x1_c,2)
tau.x2 <- 1/pow(sd_x2,2)

}
```

Step 2: R code for preparing data and calling JAGS for SFB analysis

```
##############################################################################################
# The csv data file "mydata.csv" includes the following variables:
# id:          the id number for each individual
# group_id: the id number for each cluster (i.e., intervention group), all the individuals in the control arm are
#              considered as one cluster with the same group_id
# grp:         the id number for each cluster (i.e., intervention group), same as group_id
# TREAT:   the binary indicator of treatment assignment (1=treatment arm, 0=control arm)
# y:           the outcome variable that is subject to missingness
# x1:          a level-1 covariate that is subject to missingness
# x2:          a level-2 covariate that is subject to missingness
# a1:          a level-1 auxiliary variable that determines the probability of missingness for x1 and y
# a1_gm:     the mean of a1 within each cluster (i.e., intervention group)
# a2:          a level-2 auxiliary variable that determines the probability of missingness for x2 and y
# ngrp:        the total number of clusters within an arm
# nsubj:       the total number of individuals within an arm
# for example, in our empirical dataset, y is the Multidimensional Health Locus of Control (MHLC) Internal
# subscale score at the 15 months follow-up (INT15), x1 is the MHLC Internal subscale score at baseline
# (INT0), x2 is the years of intervention experience that the co-facilitator in a PTSC group had (FACEXP),
# a1 is the stress symptoms score at baseline (STS0), and there was not an available a2.
##############################################################################################

library(rjags)   # package needed for conducting SFB analysis via JAGS
indata <- read.csv("mydata.csv")  # read in the original data with missing values

#=== 1. specify data/variables for Bayesian estimation ===#
# "_t" represents variables in the treatment arm and "_c" represents variables in the control arm
dataList = list(
            y_t  = indata$y[indata$TREAT==1],
            x1_t = indata$x1[indata$TREAT==1],
```

```
            a1_t = indata$a1[indata$TREAT==1],
            a1_gm_t = aggregate(indata$a1[indata$TREAT==1], list(indata$group_id[indata$TREAT==1]),
                        FUN=mean)$x,
            x2_t = indata$x2[!duplicated(indata$group_id) & indata$TREAT==1],
            a2_t = indata$a2[!duplicated(indata$group_id) & indata$TREAT==1],
            grp_t = indata$group_id[indata$TREAT==1],
            ngrp_t = length(unique(indata$group_id[indata$TREAT==1])),
            nsubj_t = nrow(indata[indata$TREAT==1,]),
            y_c  = indata$y[indata$TREAT==0],
            x1_c = indata$x1[indata$TREAT==0],
            a1_c = indata$a1[indata$TREAT==0],
            nsubj_c = nrow(indata[indata$TREAT==0,])
           )

#=== 2. specify the MCMC chains ===#
adaptSteps = 500               # Number of steps to "tune" the samplers.
burnInSteps = 1000             # Number of steps to "burn-in" the samplers.
nChains = 4                    # Number of chains to run.
numSavedSteps=40000            # Total number of steps in chains to save.
thinSteps=1                    # Number of steps to "thin" (1=keep every step).
nIter = ceiling( ( numSavedSteps * thinSteps ) / nChains )        # Steps per chain.

#=== 3. read in the JAGS script created in Step 1 ===#
jagsModel = jags.model("jags_model.txt", data=dataList, n.chains=nChains, n.adapt=adaptSteps)

#=== 4. sample from the posterior distributions ===#
# Burn-in:
cat("Burning in the MCMC chain...\n")
update(jagsModel, n.iter=burnInSteps)

# The saved MCMC chain:
cat("Sampling final MCMC chain...\n")
codaSamples = coda.samples(jagsModel, variable.names=parameters, n.iter=nIter, thin=thinSteps)

#=== 5. summarize estimation results  ===#
mysummary <- summary(codaSamples)
gamma_10 <- mysummary$statistics["beta_y_tr","Mean"]         # average treatment effect (ATE)
se_gamma10  <- mysummary$statistics["beta_y_tr","SD"]        # standard error of ATE
cil_gamma10 <- mysummary$quantiles["beta_y_tr","2.5%"]       # lower bound of the 95% CI for ATE
ciu_gamma10 <- mysummary$quantiles["beta_y_tr","97.5%"]      # upper bound of the 95% CI for ATE

# between-cluster variance of treatment effect in treatment arm
sigma2_u1 <- mysummary$statistics["var_u_trt","Mean"]
# between-cluster variance of x1 effect in treatment arm
# run the following line only if x1 effect is random
sigma2_u2 <- mysummary$statistics["var_u_x1t","Mean"]
```

```
# covariance between the two random effects (i.e., treatment effect and x1 effect) in treatment arm,
# run the following line only if x1 effect is random
sigma_u12 <- mysummary$statistics["cov_u_trx1t","Mean"]

sigma2_e1 <- mysummary$statistics["var_e_yt","Mean"] # within-cluster residual variance in treatment arm
sigma2_e0 <- mysummary$statistics["var_e_yc","Mean"] # residual variance in control arm
```