Simulation Results

Results

Table 1: Results Table

k	m	True effect	Methods	Average Est	Average Est SD	Coverage %	Not Converge times
25	25	1.326	CRA	1.323	0.214	97.5	0
			IPW	1.383	0.327	90.6	112
			IPW_cluster	1.387	0.335	89.7	118
			IPW-GEE (CRTgeeDR)	1.323	0.169	93.2	0
			MMI	1.325	0.296	99.9	0
	50	1.319	CRA	1.317	0.174	98.1	0
			IPW	1.355	0.258	93.6	66
			IPW_cluster	1.359	0.262	93.5	70
			IPW-GEE (CRTgeeDR)	1.319	0.119	90.8	0
			MMI	1.327	0.232	99.8	0
50	25	1.320	CRA	1.319	0.153	98.1	0
			IPW	1.362	0.242	92.0	70
			IPW_cluster	1.364	0.248	91.4	74
			IPW-GEE (CRTgeeDR)	1.322	0.119	92.8	0
			MMI	1.321	0.211	99.9	0
	50	1.319	CRA	1.317	0.124	98.7	0
			IPW	1.343	0.190	94.5	52
			IPW_cluster	1.344	0.193	94.6	48
			IPW-GEE (CRTgeeDR)	1.320	0.084	92.6	0
			MMI	1.328	0.165	99.8	0

Table 1 presents the results of this simulation.

- k is the number of clusters in each intervention group. Here we consider k=25 or k=50
- m is the number of individuals in each cluster, m=25 or m=50.
- Under the assumption of CDM as a missing mechanism, we consider:
- 1. Complete Record Analysis (CRA), with covariants adjustment;
- 2. Inverse probability weighting (IPW) without cluster effects;
- 3. IPW, with cluster effects;
- 4. Multilevel Multiple Imputation (MMI)
- The true intervention effects are shown in each scenario.

Results Comparison

For each scenario, the missing percentage is 30%.

For average estimates, since we assume covariate dependent missingness (CDM), so CRA with adjusted for covariates gains unbiased effects. And in the results, the difference between the average estimate and true effects are quite small. The same thing happens to MMI, which can be considered as unbiased based on the results. However, IPW, no matter considering clusters or not, overestimates the true effects.

Compared to MMI, IPW cannot control uncertainty in missing values, and thus IPW has a larger standard deviation than MMI. Besides, IPW methods generate approximately 10% non-convergence, and it may return some warnings when fitting to the GEE models:

glm.fit: fitted probabilities numerically 0 or 1 occurred

which means that the model has over fit and made some dangerous extreme assumptions. Besides, the larger the cluster size is, the fewer warnings are generated. The package "CRTgeeDR" has a higher efficiency than using logistic regression + geepack. It can generate results quite similar with MMI and with fully converage (converaged every time in simulation)

In the missingness generation model, there are no cluster effects. This may be the reason that the IPW with or without cluster effects have similar results.

More information about the results are on the below:

Simulation

The simulation based on Hossain's paper *"Missing binary outcomes under covariate dependent missingness in cluster randomised trials"*. The goal of the simulation is to compare the effects of inverse probability weighting (IPW) and multilevel multiple inputation (MMI)

1. Data Generation

Assuming the true data generating model hs log link, suppose that each binary outcome Y_{ijl} is generated by:

$$\pi_{ijl} = exp(\beta_0 + \beta_1 i + f_i(x_{ijl}) + \delta_{ij})$$

ijl means the *ith* intervention group, the *jth* cluster, the *lth* individual. i=0 control group while i=1 intervention group.

- 1. β_0 is a constant, β_1 is the true intervention effect. $f_i(x_{ijl})$ is a function of baseline covaeriate X in the *ith* intervention group. We set $f_0(x_{ijl}) = f_1(x_{ijl}) = \beta_2 x_{ijl}$ Consistent with Hossain's paper, we set $\beta_0 = 1, \beta_1 = 1.36, \beta_2 = 1$
- 2. X_{ijl} is generated by using the methods:

  $$X_{ijl} = \alpha_{ij} + u_{ijl}$$

  where α_{ij} is the (ij)th cluster effect on X and u_{ijl} is the individual-level error on X. We assumed that $\alpha_{ij} \sim N(\mu_x, \sigma_\alpha^2)$, $u_{ijl} \sim N(0, \sigma_u^2)$, where σ_α^2 and σ_u^2 are the between-cluster and within-cluster variance of X, respectively. We set $\mu_x = 0, \sigma_\alpha^2 = 0.18, \sigma_u^2 = 3.37$
- 3. $\delta_{ij} \sim N(0, \sigma_b^2)$. We set $\sigma_b^2 = 0.2$
- 4. Y_{ijl} is generated as Bernoulli random varaible with parameter π_{ijl}

2. Missingness generation:

We assume the missing mechanism is covariate dependent missingness (CDM).

The missingness is generated by the logistic regression model:

$$logit(R_{ijl} = 0|Y_{ij}, X_{ij}) = \psi_i + \phi_i X_{ijl}$$

For a simple example, we do not add group indicator in the model. We let: $\psi_0 = \psi_1 = -1.34$, $\phi_0 = \phi_1 = 1$

Parameter	value
β_0	0
β_1	1.36 (true effect)
β_2	1
α_{ij}	$N(\mu_x, \sigma_\alpha^2)$
u_{ijl}	$N(0, \sigma_u^2)$
$\psi_0 = \psi_1$	-1.34
$\phi_0 = \phi_1$	1
μ_x	0
σ_α^2	0.18
$\sigma_u^2 = 3.37$	3.37
δ_{ij}	$N(0, \sigma_b^2)$
σ_b^2	0.2

2. Missingness handling methods:

2.1 Complete Record Analysis (CRA)

For CRA, no imputation is performed, and only data from subjects with an observed outcome are considered for statistical analysis. Besides, we also adjusted covariates for CRA since we assume the missing mechanism is CDM. Therefore, our CRA here is adjusted CRA.

2.2 Inverse probability weigthing (IPW)

2.2.1 IPW without cluster effects

Suppose w_{ij} is the weight for y_{ij} and is defined as the inverse probability of observing y_{ij}. In other words, $w_{ij} = P(R_{ij} = 1|X_i, Y_i)^{-1}$. Suppose W_i is a $T * T$ diagonal. Consider a generalized estimating equation:

$$S(\beta) = \sum \frac{\partial \mu_i}{\partial \beta} V_i^{-1} W_i (Y - \mu_i(\beta)) = 0$$

The weights can be estimated for a logistic regrssion:

$$\hat{w}_{ijl} = expit(X_{ijl}\beta')$$

2.2.2 IPW with cluster effects (IPW_cluster)

Different with IPW without cluster effects, if we consider clusters, we need to change weights for each observed individuals and just modify the weigths equation:

$$\hat{(w_{ijl})} = expit(X_{ijl}\beta' + \delta_{ij})$$

where δ_{ij} is the cluster level variable.

2.3 Multilevel Multiple Inputation (MMI)

Since many researchers believe that MMI is the best MI methods that with consideration of cluster effects. Therefore, we use MMI as a representative of MI methods to compare with IPW.

The missing data are firstly imputed based on the random logistic regression model

$$logit(\pi_{ijl} = 0|Y_{ij}, X_{ij}) = \beta_0 + \beta_1 X_{ijl}$$

After the missing values are imputated, a full data is generated. Then GEE method can be used to analyze the full data. After several times of repeats of the previous procedures, the results can be pooled according to Rubin's rule and then one pooled estimate were generated.

Analysis model:

Generalized estimated equation (GEE) is used here to analyze the results.

And choose indenpent working covariation matrix

$$logit(\pi_{ijl} = 0) = \beta_0 + \beta_1 X_{ijl} + \beta_2 * group$$

Transform

Notice that, in our simulation, for data generation, we used generalized linear mixed model, while in analysis part we applied gee to analyze the generated data. Therefore, the data generation process gives us a conditional estimate while data analysis model provides us a marginal estimate. We have to make some transformations to make them both marginal or both conditional.

Hossain faced the same issue, and he got the true value of population averaged log(OR) for GEE by empirically estimation using full data.

Also, Zeger, et al. 1988 showed another transforamtion method, that for logistic regression:

$$\beta_M \simeq [(\frac{16\sqrt{(3)}}{15\pi})^2 V + 1]^{1/2} \beta_{RE}$$

Here, we used Hossain's method to make consistency.

Part of the R code:

1. Data Generation

```
### function for generating one intervention group (intervention group or control gruop)
one_group=function(i,k,m,seed=123){ ## i=1, intervention; i=0, control
  set.seed(seed)
  x=matrix(NA,k,m)
  y=matrix(NA,k,m)
  pi=matrix(NA,k,m)
  for(k in 1:k){
    delta=rnorm(1,0,sigma_b) ## generate delta, cluster level
    alpha=rnorm(1,mu_x,sigma_alpha) ## generate alpha, cluster level
    for(j in 1:m){
      u=rnorm(1,0,sigma_u) ## generate u_ijl, individual level
      x[k,j]=u+alpha ## x_ijl
      pi[k,j]=expit(b0+b1*i+b2*x[k,j]+delta) ## pi_ijl
      y[k,j]=rbinom(1,1,pi[k,j]) ## y_ijl from a binormal distribution of p=pi_ijl
    }
  }
  return(list(x=x,y=y,pi=pi))
}
## expit function, to inverse logit link
expit=function(x){
  y=exp(x)/(1+exp(x))
  return(y)
}
## matrix tranformation function, for generating a dataset that easy to analyze
trans=function(X,name){
  x1=as.vector(X)
  x2=matrix(x1,m,k*2,byrow = TRUE)
  X=matrix(x2,2*k*m,1,byrow = TRUE)
  X=data.frame(X)
  colnames(X)=name
  X$group=c(rep(1,k*m),rep(0,k*m))
  X$cluster=rep(1:(2*k),each=m)
  return(X)
}
## DataGeneration function, combination all the functions above
data_generation=function(k,m,seed=123,print=0){
  set.seed(seed)
  trt=one_group(1,k,m,seed)  ## generate treatment group
  con=one_group(0,k,m,seed)  ## generate individual group
  X=rbind(trt$x,con$x)
  Y=rbind(trt$y,con$y)
  # generate missingness
  r=expit(psi+phi*X) ## missingness generation function
  Y_mis=Y
  R=matrix(NA,2*k,m)
  for(i in 1:(2*k)){
    for(j in 1:m){
      R[i,j]=rbinom(1,1,r[i,j])
      if(R[i,j]==1){Y_mis[i,j]=NA}
    }
  }
  # missing percentage:
  if(print==1){
    print(mean(r))
    print(mean(expit(psi+phi*trt$x)))
    print(mean(expit(psi+phi*con$x)))
    print(sum(is.na(Y_mis))/(k*m*2))
    mispre=sum(is.na(Y_mis))/(k*m*2)
  }
  X=trans(X,'X')
  Y=trans(Y,'Y')
  Y_mis=trans(Y_mis,'Y_mis')
  R=trans(R,'R')
  return(list(X=X,Y=Y,Y_mis=Y_mis,R=R,mispre=mispre))
}
```

2. Functions

```
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 3.4.1
```

```
## Loading required package: Matrix
```

```
library(geepack)
library(jomo)
```

```
## Warning: package 'jomo' was built under R version 3.4.1
```

```
### function for generate one

## Function for pooling results according to rubin's rule:
mypool=function(mean0,sd0,num=5,print='no'){
  m=mean(mean0)
  v=mean(sd0)
  B=sd(mean0)
  v_hat=v+(1+1/num)*B
  l=m-1.96*v_hat
  u=m+1.96*v_hat
  if(print=='no'){
    return(list(mean=m,std=v_hat))
  }
  if(print=='yes'){
    print('mean (95% CI)')
    print(paste(round(m,2)," (",round(l,2),',',round(u,2),')',sep=''))
    return(list(mean=m,std=v_hat))
  }
}

## function for analyzing MMI results
analysis_results_mmi=function(mmi){
  m=c()
  std=c()
  #icc=c()
  for(i in 1:5){
    # the imputated data generated in ith time of imputation
    temp=mmi[mmi$Imputation==i,]
    temp0=temp[,c('group','X','cluster','Y_mis')]
    # analysis: GEE
    formu=formula(Y_mis~X+group)
    temp0$Y_mis=as.numeric(temp0$Y_mis)
    temp0$Y_mis=temp0$Y_mis-1
    mmi2=geese(formu,data=temp0,id=cluster,
               family = binomial(link='logit'),
               corstr = 'independence')
    est=summary(mmi2)
    ## gee results
    est_trt=est$mean['group','estimate']
    sd_trt=est$mean['group','san.se']
    m=c(m,est_trt)
    std=c(std,sd_trt)
  }
  ## pool results
  return(mypool(m,std,print='yes'))
}
## function to track warnings
myTryCatch = function(expr) {
  warn <- err <- NULL
  value <- withCallingHandlers(
    tryCatch(expr, error=function(e) {
      err <<- e
      NULL
    }), warning=function(w) {
      warn <<- w
      invokeRestart("muffleWarning")
    })
  list(value=value, warning=warn, error=err)
}
```

3. Analysis

codes for missingness analysis methods:

```
#for example,k=25,m=25
times=1 ## times is the seed
data_sim=data_generation(k,m,times+130,1)  ## Generate a data set
```

```
## [1] 0.2937879
## [1] 0.2937879
## [1] 0.2937879
## [1] 0.3064
```

```
org_data=data_sim$Y_mis
print(head(org_data))
```

```
##   Y_mis group cluster
## 1    NA     1       1
## 2     0     1       1
## 3     0     1       1
## 4     1     1       1
## 5     0     1       1
## 6     1     1       1
```

```
org_data$X=data_sim$X$X
cca=na.omit(org_data)                          ## Generate complete data set
dim(cca)
```

```
## [1] 867   4
```

```
print(head(cca))
```

```
##   Y_mis group cluster          X
## 2     0     1       1 -1.5400348
## 3     0     1       1 -0.6189987
## 4     1     1       1 -1.1096953
## 5     0     1       1 -3.4790138
## 6     1     1       1 -0.3588921
## 7     1     1       1  0.3562128
```

```
### analyze the complete data set with GEE
### myTryCatch is the function to track warnings
formu=formula(Y_mis~X+group)
cca_adj=myTryCatch(
  geese(formu,data=cca,id=cluster,
              family = binomial(link='logit'),
              corstr = 'independence'))

### Keep results
error_cca=c(error_cca,cca_adj$error)
warning_cca=c(warning_cca,cca_adj$warning)
if(is.null(cca_adj$error)==0){errors=errors+1}
if(is.null(cca_adj$warning)==0){warn=warn+1}

cca_adj=cca_adj$value
est0=summary(cca_adj)

est_trt0=est0$mean['group','estimate']
sd_trt0=est0$mean['group','san.se']
cca_mean=c(cca_mean,est_trt0)
cca_std=c(cca_std,sd_trt0)

print(est_trt0)
```

```
## [1] 1.247007
```

```
print(sd_trt0)
```

```
## [1] 0.2178779
```

```
### 2. IPW, without clusters
## first calculate the missing weight through logistic regression.
mis=data_sim$R
mis$X=data_sim$X$X
head(mis)
```

```
##   R group cluster          X
## 1 1     1       1  1.6433708
## 2 0     1       1 -1.5400348
## 3 0     1       1 -0.6189987
## 4 0     1       1 -1.1096953
## 5 0     1       1 -3.4790138
## 6 0     1       1 -0.3588921
```

```
logs=glm(R ~ X , data = mis,
         family = binomial(link='logit'))
logsum=summary(logs)
weight=expit(logsum$coefficients[,'Estimate'][1]+
               logsum$coefficients[,'Estimate'][2]*mis$X)
weight=matrix(weight,((2*k)*m),1)
## GEE analysis, added weigth
ipw=myTryCatch(
geese(formu,data=cca,id=cluster,
           family = binomial(link='logit'),
           weights = cca$weight,
           corstr = 'independence'))

error_ipw1=c(error_ipw1,ipw$error)
warning_ipw1=c(warning_ipw1,ipw$warning)
if(is.null(ipw$error)==0){errors_ipw1=errors_ipw1+1}
if(is.null(ipw$warning)==0){
  warn_ipw1=warn_ipw1+1
  warn_time1=c(warn_time1,times)}
## get the results
ipw=ipw$value
est=summary(ipw)
est_trt=est$mean['group','estimate']
sd_trt=est$mean['group','san.se']
ipw_mean=c(ipw_mean,est_trt)
ipw_std=c(ipw_std,sd_trt)

print(est_trt)
```

```
## [1] 1.247007
```

```
print(sd_trt)
```

```
## [1] 0.2178779
```

```
## ipw-gee with cluster effects
mis2=data_sim$R
mis2$X=data_sim$X$X
head(mis2)
```

```
##   R group cluster          X
## 1 1     1       1  1.6433708
## 2 0     1       1 -1.5400348
## 3 0     1       1 -0.6189987
## 4 0     1       1 -1.1096953
## 5 0     1       1 -3.4790138
## 6 0     1       1 -0.3588921
```

```
# estimate the weights with consideration of clusters
logs2 = myTryCatch(
  glmer(R ~ X+(1|cluster) , data = mis2,
              family = binomial(link='logit')))
error_ipw2_glmer=c(error_ipw2_glmer,logs2$error)
warning_ipw2_glmer=c(warning_ipw2_glmer,logs2$warning)

logs2=logs2$value
logsum2=summary(logs2)

## counts random effects in the glmer model
ran=as.data.frame(VarCorr(logs2))['sdcor']
ran=as.numeric(ran)
rans=c()

for(i in 1:(2*k)){
  rans=c(rans,rnorm(1,0,sqrt(ran)))
}
```

```
rans2=rep(rans,each=m)
mis2$rans=rans2
head(mis2)
```

```
##   R group cluster          X          rans
## 1 1     1       1  1.6433708 -1.732342e-05
## 2 0     1       1 -1.5400348 -1.732342e-05
## 3 0     1       1 -0.6189987 -1.732342e-05
## 4 0     1       1 -1.1096953 -1.732342e-05
## 5 0     1       1 -3.4790138 -1.732342e-05
## 6 0     1       1 -0.3588921 -1.732342e-05
```

```
weight=expit(logsum$coefficients[,'Estimate'][1]+
               logsum$coefficients[,'Estimate'][2]*mis2$X+
               mis2$rans)

org_data2=data_sim$Y_mis
org_data2$X=data_sim$X$X
org_data2$weight=round(1/weight)
head(org_data2)
```

```
##   Y_mis group cluster          X weight
## 1    NA     1       1  1.6433708      2
## 2     0     1       1 -1.5400348     23
## 3     0     1       1 -0.6189987      9
## 4     1     1       1 -1.1096953     14
## 5     0     1       1 -3.4790138    194
## 6     1     1       1 -0.3588921      7
```

```
cca2=na.omit(org_data2)
head(cca2)
```

```
##   Y_mis group cluster          X weight
## 2     0     1       1 -1.5400348     23
## 3     0     1       1 -0.6189987      9
## 4     1     1       1 -1.1096953     14
## 5     0     1       1 -3.4790138    194
## 6     1     1       1 -0.3588921      7
## 7     1     1       1  0.3562128      4
```

```
## gee analysis, with cluster effects in weigths
formu=formula(Y_mis~X+group)
ipw2=myTryCatch(
  geese(formu,data=cca2,id=cluster,
            family = binomial(link='logit'),
            #family=quasibinomial(),
            weights = cca2$weight,
            corstr = 'independence'))
error_ipw2_gee=c(error_ipw2_gee,ipw2$error)
warning_ipw2_gee=c(warning_ipw2_gee,ipw2$warning)
if(is.null(ipw2$error)==0){errors_ipw2=errors_ipw2+1}
if(is.null(ipw2$warning)==0){
  warn_ipw2=warn_ipw2+1
  warn_time2=c(warn_time2,times)}
## get results
```

```
ipw2=ipw2$value
est2=summary(ipw2)
est_trt2=est2$mean['group','estimate']
sd_trt2=est2$mean['group','san.se']

ipw_mean2=c(ipw_mean2,est_trt2)
ipw_std2=c(ipw_std2,sd_trt2)
print(est_trt2)
```

```
## [1] 1.253452
```

```
print(sd_trt2)
```

```
## [1] 0.3411204
```

```
## MMI-GEE
  org_data=data_sim$Y_mis
  org_data$X=data_sim$X$X
  org_data$R=data_sim$R$R
  mmidata=org_data[,1:4]
head(mmidata)
```

```
##   Y_mis group cluster          X
## 1    NA     1       1  1.6433708
## 2     0     1       1 -1.5400348
## 3     0     1       1 -0.6189987
## 4     1     1       1 -1.1096953
## 5     0     1       1 -3.4790138
## 6     1     1       1 -0.3588921
```

```
cluster=mmidata$cluster
mmidata$Y_mis=as.factor(mmidata$Y_mis)
## use jomo to deal with random effects, add cluster effects
mmi=myTryCatch(
  jomo(mmidata,nburn=nburn,clus=cluster,
         nbetween=nbetween,nimp=nimp,output = 0))
```

```
## Clustered data, using functions for two-level imputation.
## Found  3 continuous outcomes and  1 categorical. Using function jomo1ranmix.
```

```
error_mmi=c(error_mmi,mmi$error)
warning_mmi=c(warning_mmi,mmi$warning)

mmi=mmi$value
m0=c()
std0=c()
#icc=c()
## the results function
result_mmi=function(mmi){
  for(i in 1:5){
    temp=mmi[mmi$Imputation==i,]
    #temp=temp[-1,]
    temp0=temp[,c('group','X','cluster','Y_mis')]
    formu=formula(Y_mis~X+group)
    temp0$Y_mis=as.numeric(temp0$Y_mis)
    temp0$Y_mis=temp0$Y_mis-1
    mmi2=geese(formu,data=temp0,id=cluster,
               family = binomial(link='logit'),
               corstr = 'independence')
    est=summary(mmi2)
    est_trt=est$mean['group','estimate']
    sd_trt=est$mean['group','san.se']
    m0=c(m0,est_trt)
    std0=c(std0,sd_trt)
  }
  return(list(m0,std0))
}
## track whether results have warnings
rmmi=myTryCatch(
  result_mmi(mmi))

error_mmi_result=c(error_mmi_result,rmmi$error)
warning_mmi_result=c(warning_mmi_result,rmmi$warning)
if(is.null(rmmi$error)==0){errors_jomo=errors_jomo+1}
if(is.null(rmmi$warning)==0){warn_jomo=warn_jomo+1}
## get results
rmmi=rmmi$value
m0=rmmi[[1]]
std0=rmmi[[2]]
mmi_result=mypool(m0,std0,print='yes')
```

```
## [1] "mean (95% CI)"
## [1] "1.33 (0.77,1.88)"
```

```
mmi_m=mmi_result$mean
mmi_s=mmi_result$std
jomo_mean=c(jomo_mean,mmi_m)
jomo_std=c(jomo_std,mmi_s)
print(mmi_m)
```

```
## [1] 1.325957
```

```
print(mmi_s)
```

```
## [1] 0.2834537
```

```
## IPW-GEE using CRTgeeDR package
library('CRTgeeDR')
```

```
## Warning: package 'CRTgeeDR' was built under R version 3.4.2
```

```
## Loading required package: MASS
```

```
## Loading required package: ggplot2
```

```
  org_data=data_sim$Y_mis
  org_data$X=data_sim$X$X
  org_data$R=data_sim$R$R
  names(org_data)=c('OUTCOME','TRT','CLUSTER','AGE','MISSING')
  head(org_data)
```

```
##   OUTCOME TRT CLUSTER        AGE MISSING
## 1      NA   1       1  1.6433708       1
## 2       0   1       1 -1.5400348       0
## 3       0   1       1 -0.6189987       0
## 4       1   1       1 -1.1096953       0
## 5       0   1       1 -3.4790138       0
## 6       1   1       1 -0.3588921       0
```

```
ipwresults<-myTryCatch(geeDREstimation(formula=OUTCOME~TRT+AGE,
                          id="CLUSTER" , data = org_data,
                          family = "binomial", corstr = "independence",
                          model.weights=I(MISSING==0)~AGE))
#print(ipwresults)
ipwresults=ipwresults$value
warnss=c(warnss,ipwresults$warning)
if(is.null(ipwresults$warning)==0){warn_n=warn_n+1}
res_m=summary(ipwresults)
print(res_m)
```

```
##             Estimates Model SE Robust SE   wald p
## (Intercept)    1.0970  0.11060   0.18180  6.037 0
## TRT            1.2920  0.16860   0.21790  5.928 0
## AGE            0.9887  0.06429   0.08402 11.770 0
##
##  Est. Correlation:  0
##  Correlation Structure:  independence
##  Est. Scale Parameter:  0.9867
##
##  Number of GEE iterations: 2
##  Number of Clusters:  50    Maximum Cluster Size:  25
##  Number of observations with nonzero weight:  867
```

```
ipw_mm=c(ipw_mm,res_m$beta[2])
ipw_mm_sd=c(ipw_mm_sd,res_m$se.model[2])
print(ipw_mm)
```

```
## [1] 1.291603
```

```
print(ipw_mm_sd)
```

```
## [1] 0.1685572
```