GSimp simulation

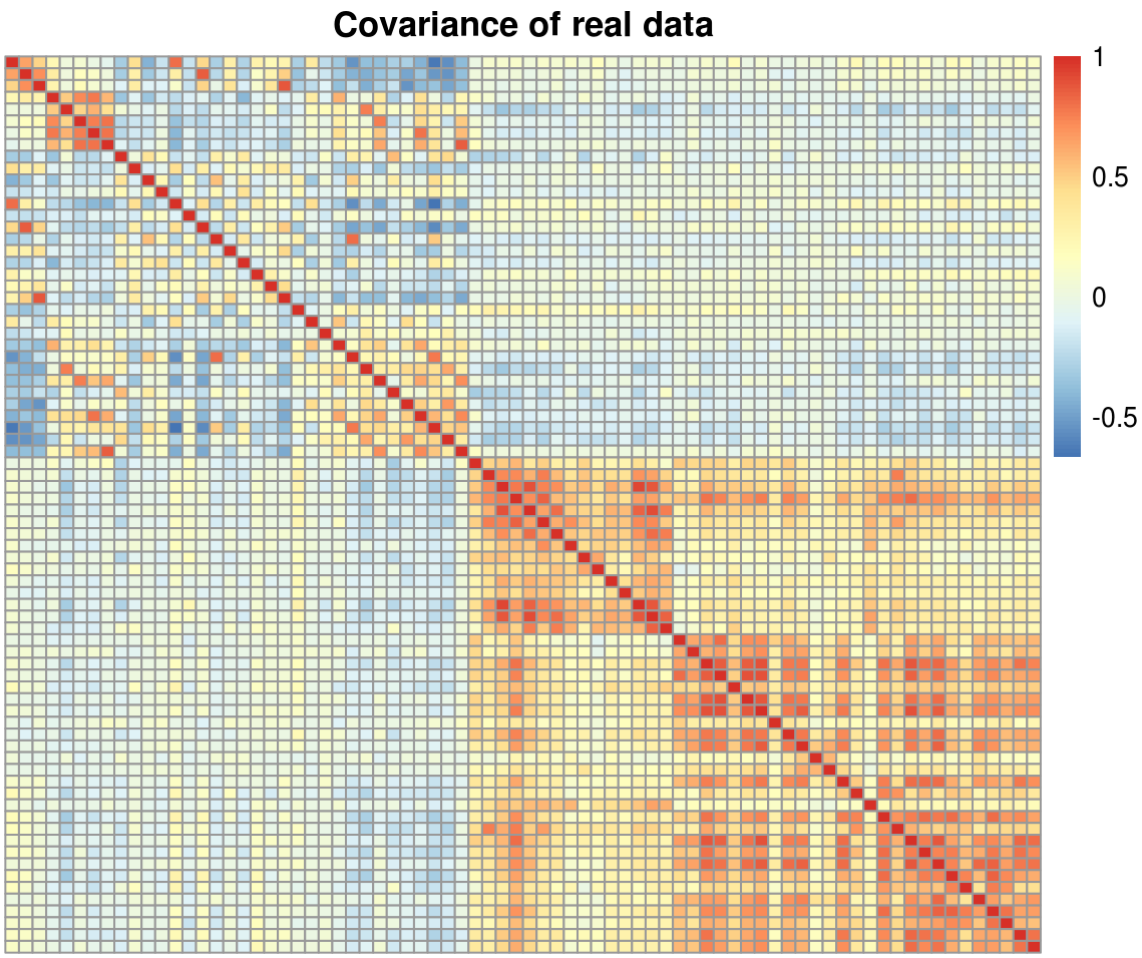
This is a vignette for GSimp on a simulation dataset

1.Preparation

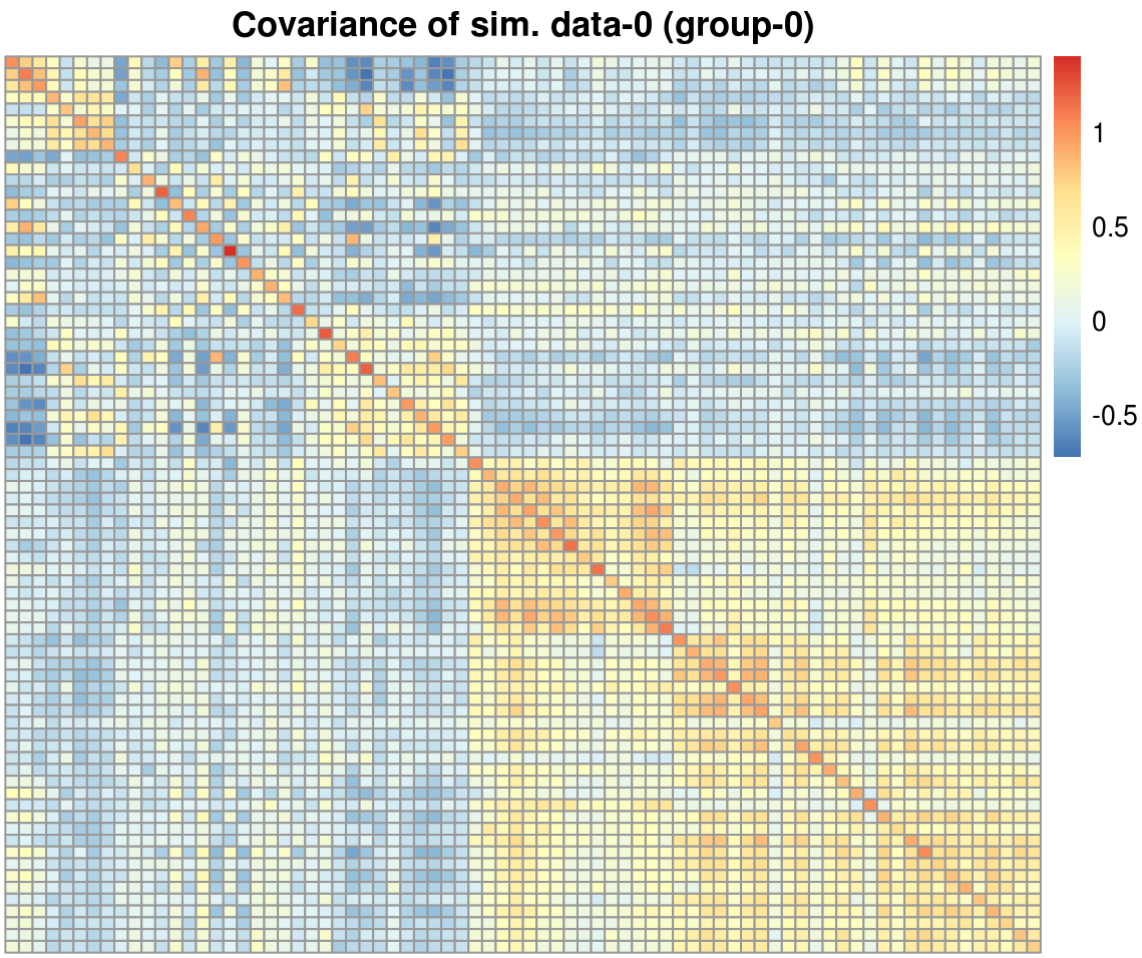
```
options(stringsAsFactors = F)
source('Trunc_KNN/Imput_funcs.r')
source('LfCs_eval.R')
source('LfCs_imp.R')
require(magrittr)
require(qvalue)
require(pheatmap)
```

2.Simulation dataset generation

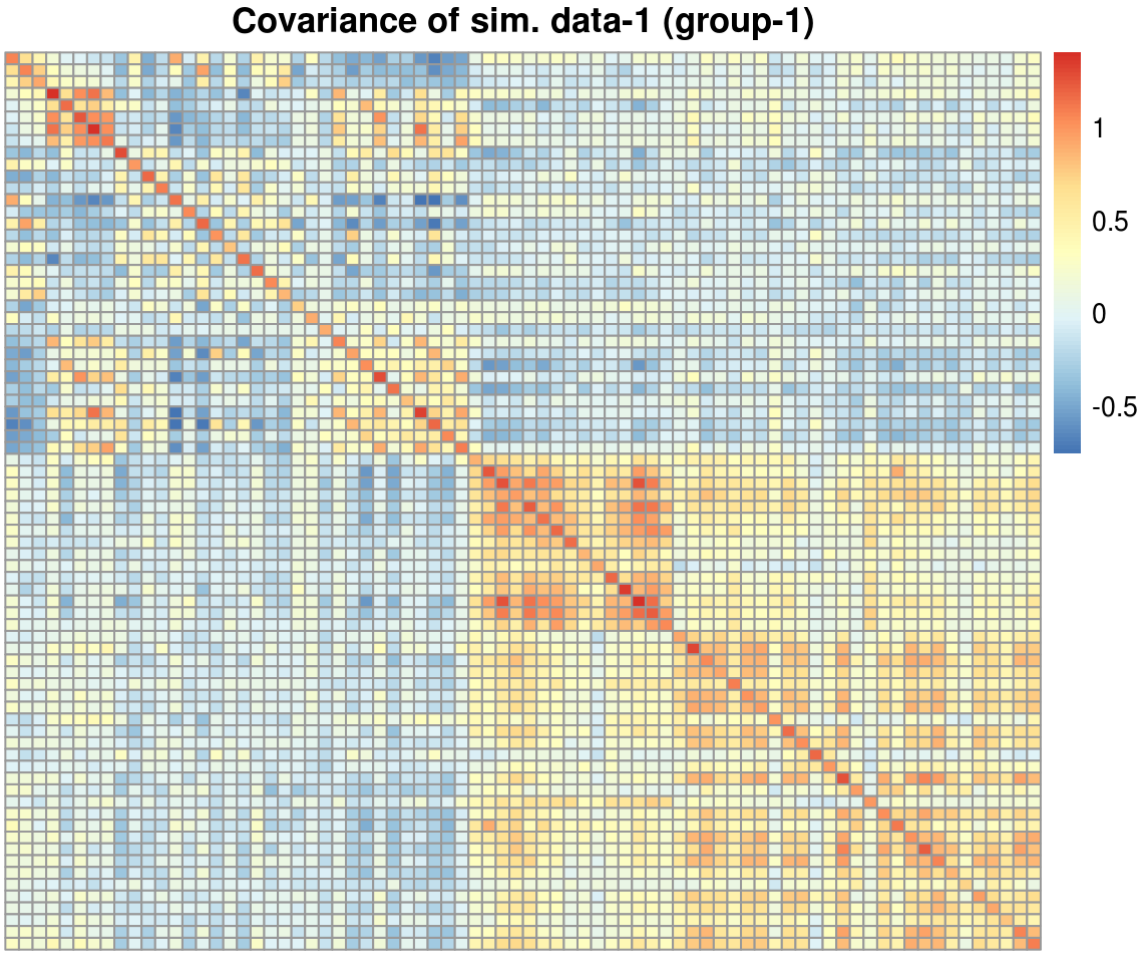
```
data_raw <- read.csv('real_data.csv', row.names=1)
data_lg_sc <- data_raw %>% log %>% scale
cov_mat <- cov(data_lg_sc)
pheatmap(cov_mat, cluster_rows=F, cluster_cols=F, show_rownames=F, show_colnames=F, m
ain='Covariance of real data')
```



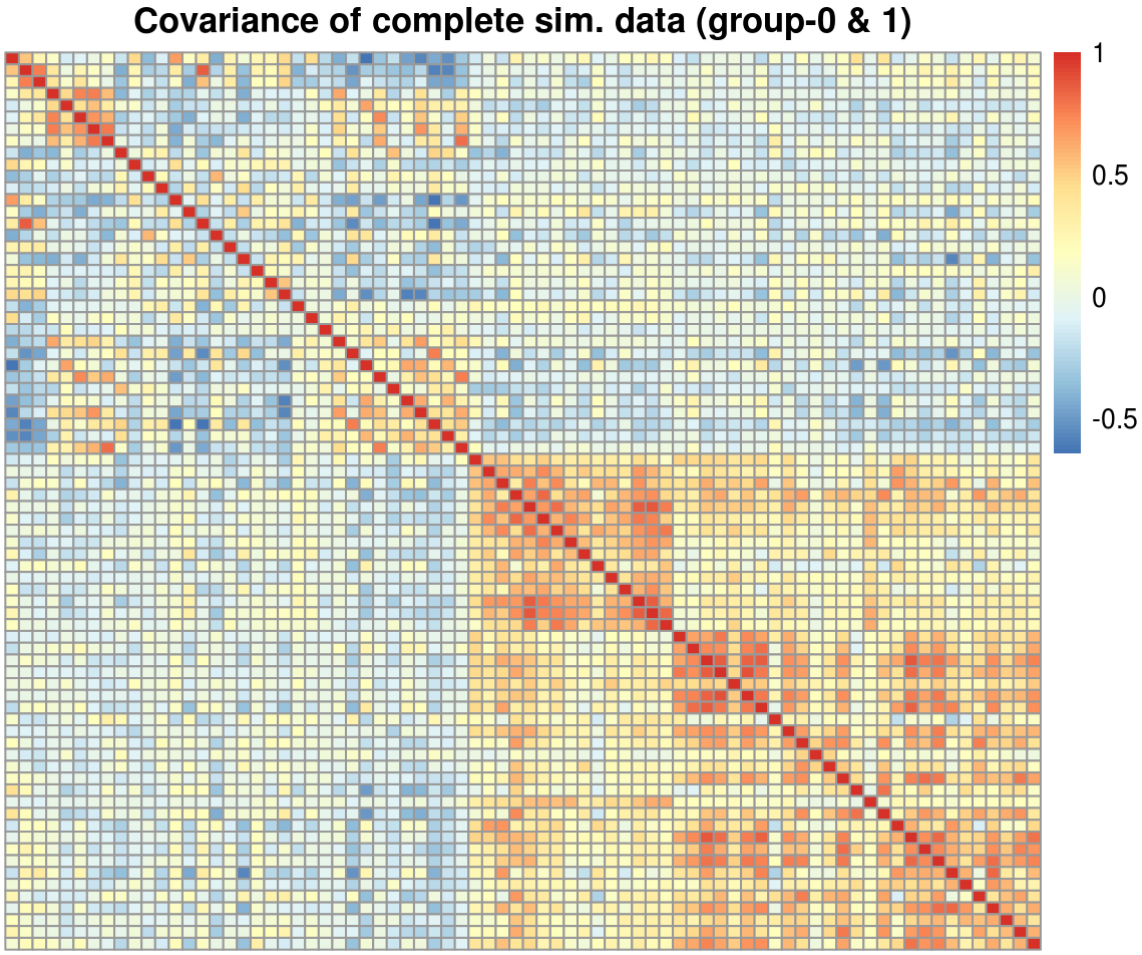
```
set.seed(123)
data_sim_0 <- rmvnorm(80, mean=rnorm(nrow(cov_mat), sd=.5), sigma=cov_mat, method='svd')
cov_mat_sim_0 <- cov(data_sim_0)
pheatmap(cov_mat_sim_0, cluster_rows=F, cluster_cols=F, show_rownames=F, show_colnames=F, main='Covariance of sim. data-0 (group-0)')
```




```
set.seed(321)
data_sim_1 <- rmvnorm(80, mean=rnorm(nrow(cov_mat), sd=.5), sigma=cov_mat, method='svd')
cov_mat_sim_1 <- cov(data_sim_1)
pheatmap(cov_mat_sim_1, cluster_rows=F, cluster_cols=F, show_rownames=F, show_colnames=F, main='Covariance of sim. data-1 (group-1)')
```




```
data_sim <- rbind(data_sim_0, data_sim_1)
data_sim_sc <- scale(data_sim)
cov_mat_sim <- cov(data_sim_sc)
pheatmap(cov_mat_sim, cluster_rows=F, cluster_cols=F, show_rownames=F,
show_colnames=F, main='Covariance of complete sim. data (group-0 & 1)')
```




```
group <- rep(c(0, 1), each=80) %>% as.factor
sim_pvals <- apply(data_sim_sc, 2, function(x) t.test(x ~ group)$p.value)
```

```
## P-values calculated on simulation dataset is:
```

```
## 7.746669e-11 1.963778e-01 8.522198e-06 9.885972e-02 1.165720e-01 5.295838e-06 5.51
0601e-01 3.338875e-06 1.934161e-04 1.385230e-01 3.116737e-03 5.767210e-03 3.043627e-0
1 3.155148e-13 5.439299e-02 3.157601e-11 7.767947e-01 1.358855e-14 5.886739e-01 2.089
223e-05
```

```
## ...
```

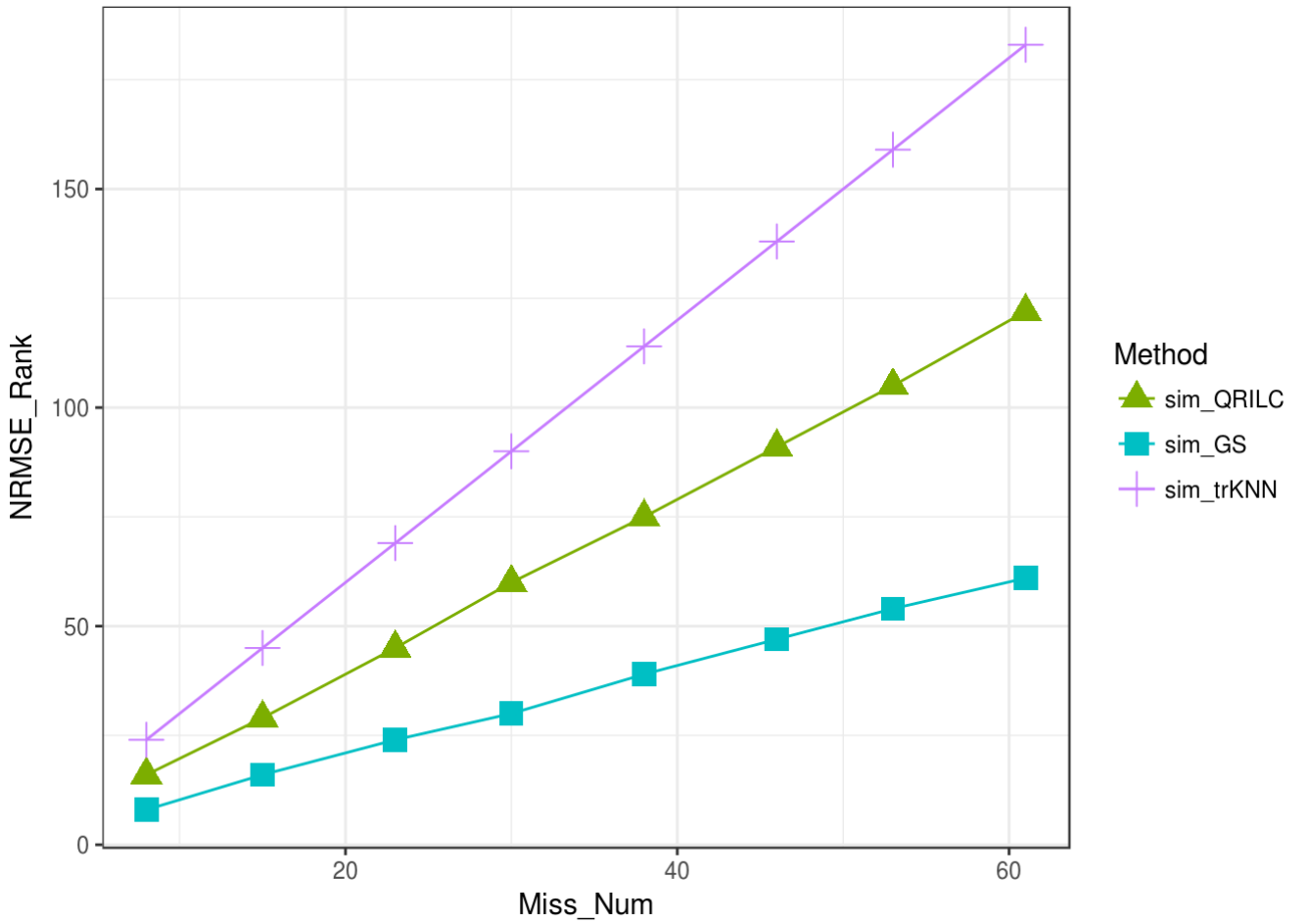
3.Imputation methods

```
sim_QRILC_wrapper <- function(data, ...) {
  result <- data %>%  impute.QRILC(., ...) %>% extract2(1)
  return(result)
}

sim_GS_wrapper <- function(data) {
  result <- data %>% multi_impute(., iters_each=50, iters_all=20, initial='qrilc',
lo=-Inf, hi='min',
                                  n_cores=1, imp_model='glmnet_pred')
  return(result$data_imp)
}

sim_trKNN_wrapper <- function(data) {
  result <- data %>% as.matrix %>% t %>% imputeKNN(., k=3, distance='truncation', per
c=0) %>% t
  return(result)
}
```

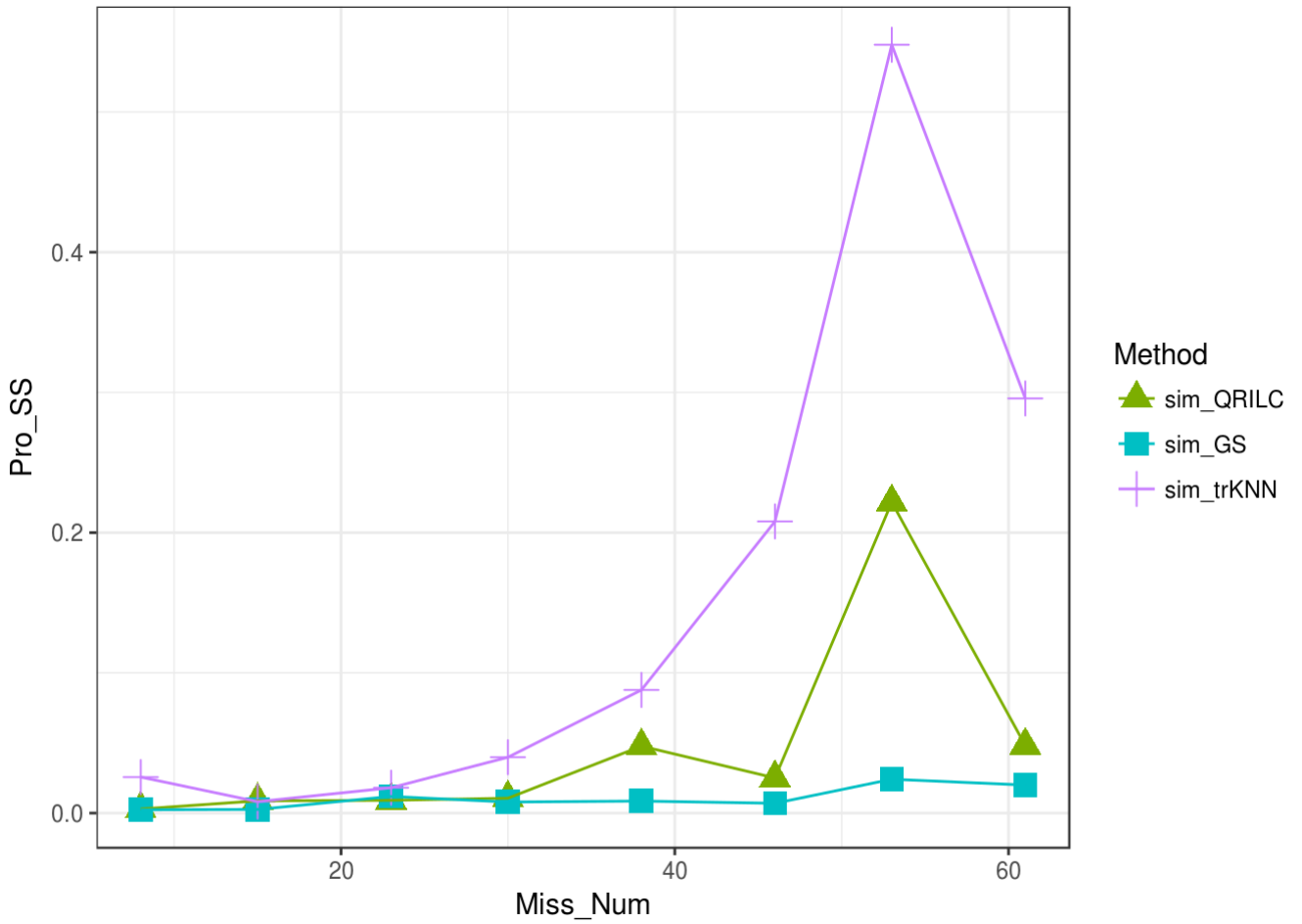
4.Imputation comparison

```
## a list of MNAR datasets generation and imputation with different imputation wrappe
r functions ##
sim_MNAR_list <- MNAR_gen_imp(data_c=data_sim_sc, mis_var_prop=seq(.1, .8, .1), var_m
is_prop=seq(.3, .6, .1),
                                impute_list=c('sim_QRILC_wrapper', 'sim_GS_wrapper',
'sim_trKNN_wrapper'), cores=10)
```

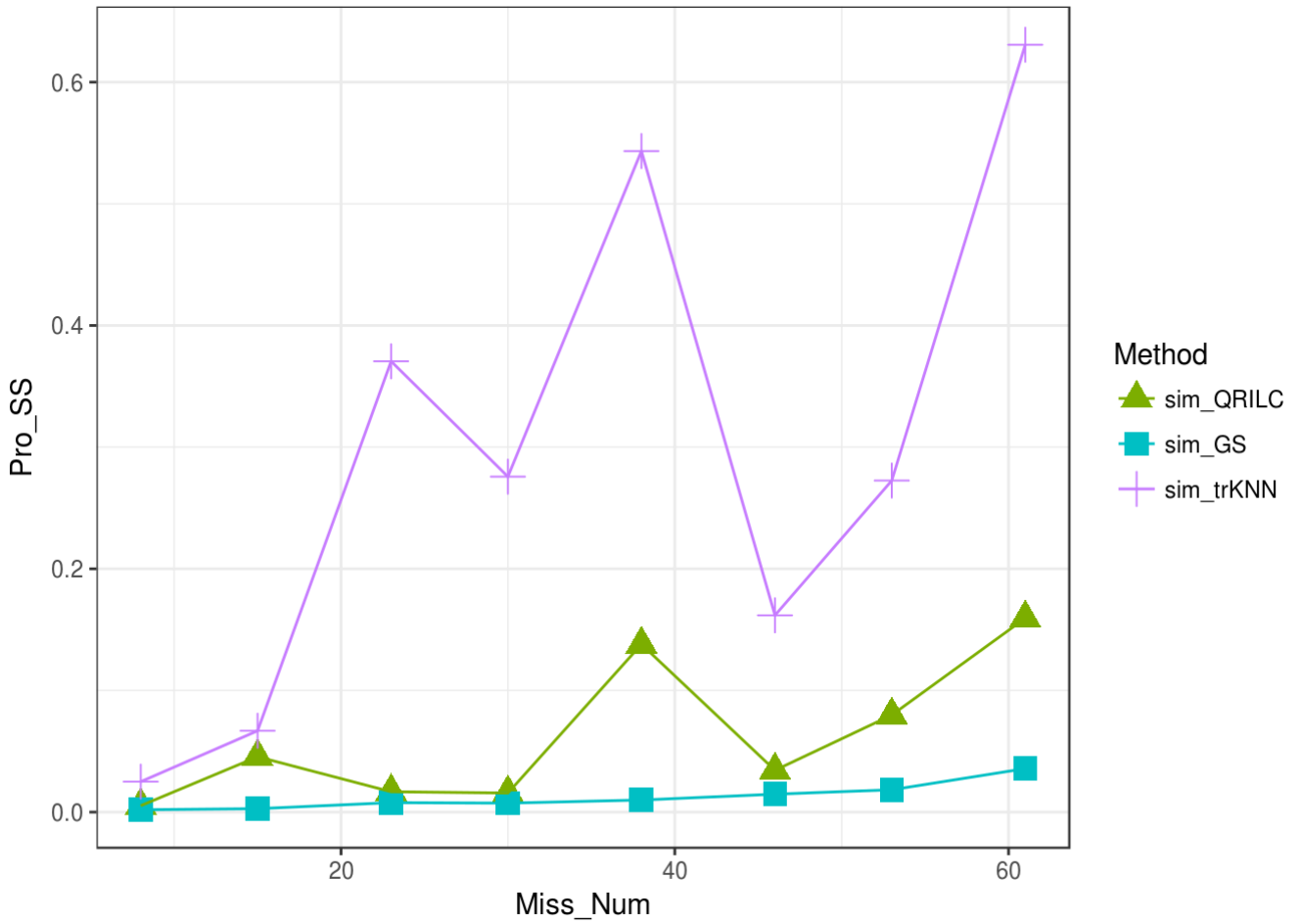
```
## SOR calculation and plot ##
sim_MNAR_NRMSE_rank_list <- NRMSE_rank_cal_plot(sim_MNAR_list, plot=T, x='Miss_Num',
                                                colors=c('#7CAE00', '#00BFC4', '#C77C
FF'), shapes=c(17, 15, 3))
```



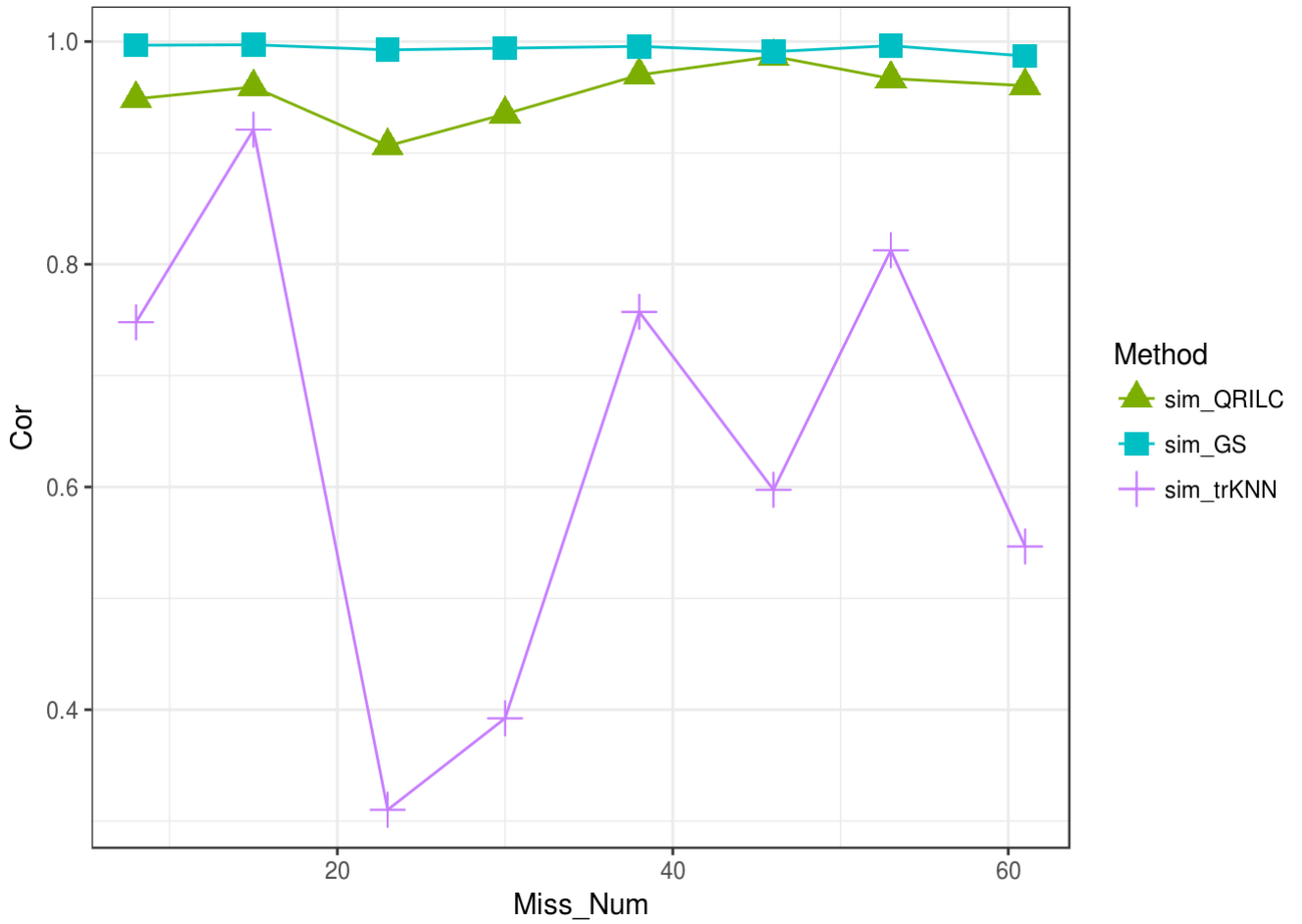

```
## PCA-Procrustes and plot ##
sim_MNAR_PCA_ProSS_list <- Procrustes_cal_plot(sim_MNAR_list, DR='PCA', nPCs=2, x='Mi
ss_Num', plot=T,
                                               colors=c('#7CAE00', '#00BFC4', '#C77CF
F'), shapes=c(17, 15, 3))
```



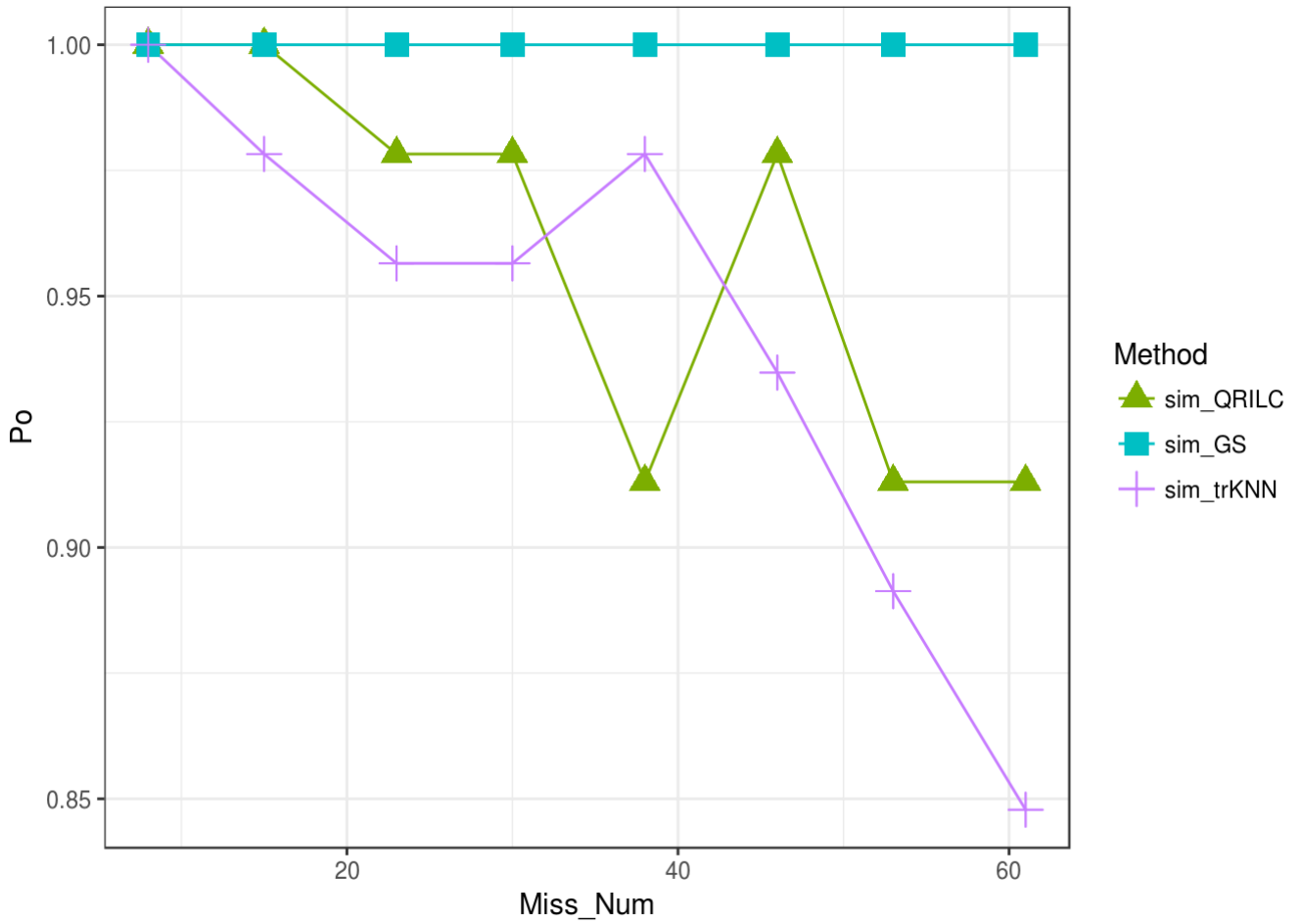

```
## PLS-Procrustes and plot ##
sim_MNAR_PLS_ProSS_list <- Procrustes_cal_plot(sim_MNAR_list, DR='PLS', nPCs=2, outco
me=group, x='Miss_Num', plot=T,
                                                colors=c('#7CAE00', '#00BFC4', '#C77CF
F'), shapes=c(17, 15, 3))
```



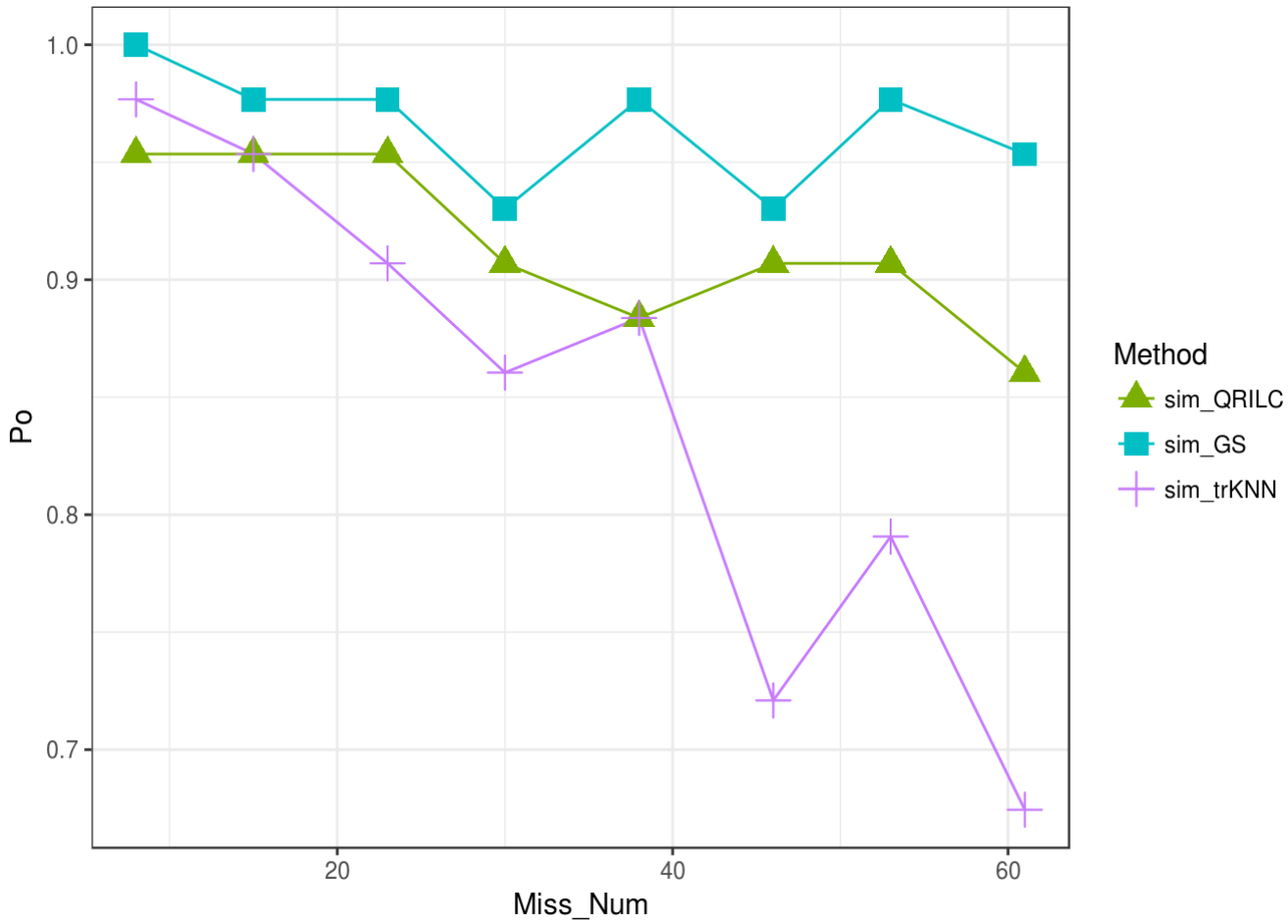

```
## Ttest correlation and plot ##
sim_MNAR_Ttest_Cor_list <- Ttest_cor_cal_plot(sim_MNAR_list, group=group, plot=T,
x='Miss_Num', cor='P',
                                              colors=c('#7CAE00', '#00BFC4', '#C77CF
F'), shapes=c(17, 15, 3))
```




```
## Ttest Po calculation and plot ##
sim_MNAR_Ttest_Power_list_2 <- Ttest_P_cal_plot(impute_results=sim_MNAR_list, group=g
roup, plot=T, p_cut=.05, x = 'Miss_Num',
                                                colors=c('#7CAE00', '#00BFC4', '#C77C
FF'), shapes=c(17, 15, 3))
```




```
sim_MNAR_Ttest_Power_list_3 <- Ttest_P_cal_plot(impute_results=sim_MNAR_list, group=g
roup, plot=T, p_cut=.01, x = 'Miss_Num',
                                                colors=c('#7CAE00', '#00BFC4', '#C77C
FF'), shapes=c(17, 15, 3))
```

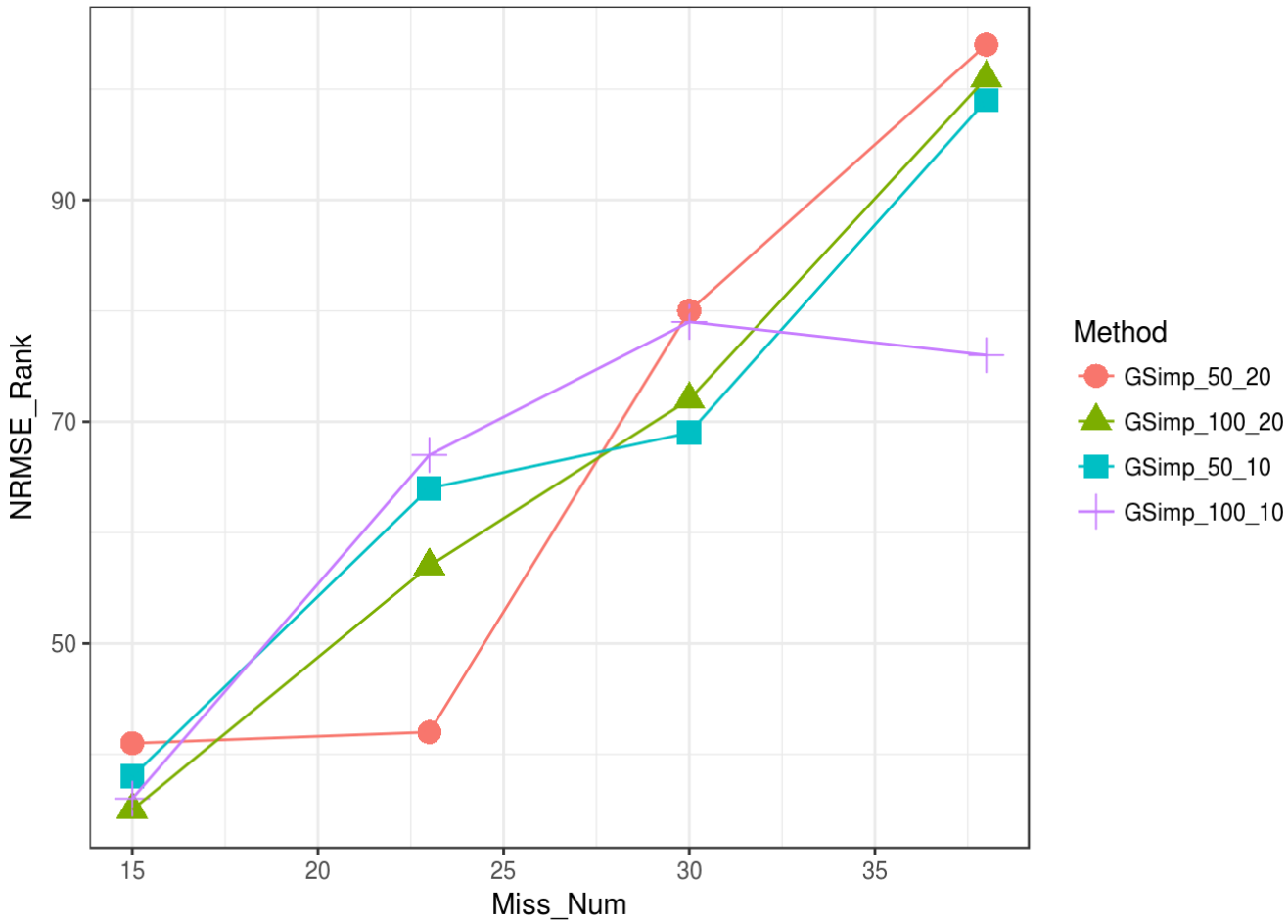


5.GSimp with different iterations

```
GSimp_50_20_wrapper <- function(data) {
  result <- data %>% multi_impute(., iters_each=50, iters_all=20, initial='qrilc',
lo=-Inf, hi='min',
                                  n_cores=1, imp_model='glmnet_pred')
  return(result$data_imp)
}

GSimp_100_20_wrapper <- function(data) {
  result <- data %>% multi_impute(., iters_each=100, iters_all=20, initial='qrilc', l
o=-Inf, hi='min',
                                  n_cores=1, imp_model='glmnet_pred')
  return(result$data_imp)
}

GSimp_50_10_wrapper <- function(data) {
  result <- data %>% multi_impute(., iters_each=50, iters_all=10, initial='qrilc',
lo=-Inf, hi='min',
                                  n_cores=1, imp_model='glmnet_pred')
  return(result$data_imp)
}

GSimp_100_10_wrapper <- function(data) {
  result <- data %>% multi_impute(., iters_each=100, iters_all=10, initial='qrilc', l
o=-Inf, hi='min',
                                  n_cores=1, imp_model='glmnet_pred')
  return(result$data_imp)
}
```

```
GSimp_iters_MNAR_list <- MNAR_gen_imp(data_c=data_sim_sc, mis_var_prop=seq(.2, .5,
.1), var_mis_prop=seq(.1, .5, .1),
                              impute_list=c('GSimp_50_20_wrapper', 'GSimp_100_20_wrap
per', 'GSimp_50_10_wrapper', 'GSimp_100_10_wrapper'),
                              cores=15)
```

```
GSimp_iters_MNAR_NRMSE_rank_list <- NRMSE_rank_cal_plot(GSimp_iters_MNAR_list,
plot=T, x='Miss_Num')
```


NRMSE_Rank
90
70
50
15
20
25
30
35
Miss_Num
Method
GSimp_50_20
GSimp_100_20
GSimp_50_10
GSimp_100_10