

Simulations for ‘Treatment effect estimation with missing attributes’

Imke Mayer*

Julie Josse†

March 2021

Abstract

This notebook allows to replicate the simulations for the paper under review *Treatment effect estimation with missing attributes*. With this notebook you can generate simulations and estimate treatment effects with incomplete covariates obtained with different missingness mechanisms and proportions of missing values.

Contents

1 Preliminaries	2
2 Note on distributional shift	4
3 Simulation setting	7
4 Classical ignorability + missingness assumptions	7
4.1 MCAR	8
4.1.1 On full data	8
4.1.2 Use only complete cases (for logistic+linear regressions)	9
4.1.3 Use EM to handle incomplete cases	10
4.1.4 Use MIA to handle incomplete cases	11
4.1.5 Use within-study multiple imputation	12
4.1.6 Use multilevel multiple imputation with micemd	14
4.2 MAR	15
4.2.1 On full data	15
4.2.2 Use only complete cases (for logistic+linear regressions)	17
4.2.3 Use EM to handle incomplete cases	18
4.2.4 Use MIA to handle incomplete cases	19
4.2.5 Use within-study multiple imputation	20
4.2.6 Use multilevel multiple imputation with micemd	22
4.3 MNAR (self-mask)	23
4.3.1 On full data	23
4.3.2 Use only complete cases (for logistic+linear regressions)	25
4.3.3 Use EM to handle incomplete cases	26
4.3.4 Use MIA to handle incomplete cases	27
4.3.5 Use within-study multiple imputation	28
4.3.6 Use multilevel multiple imputation with micemd	29
5 Conditional independence of selection (CIS)	30
5.1 MCAR	30
5.1.1 On full data	30

*EHESS, imke.mayer@ehess.fr

†Inria

5.1.2	Use only complete cases (for logistic+linear regressions)	32
5.1.3	Use EM to handle incomplete cases	33
5.1.4	Use MIA to handle incomplete cases	34
5.1.5	Use within-study multiple imputation	35
5.1.6	Use multilevel multiple imputation with micemd	36
5.2	MAR	38
5.2.1	On full data	38
5.2.2	Use only complete cases (for logistic+linear regressions)	40
5.2.3	Use EM to handle incomplete cases	41
5.2.4	Use MIA to handle incomplete cases	42
5.2.5	Use within-study multiple imputation	43
5.2.6	Use multilevel multiple imputation with micemd	45
5.3	MNAR (self-mask)	46
5.3.1	On full data	46
5.3.2	Use only complete cases (for logistic+linear regressions)	47
5.3.3	Use EM to handle incomplete cases	48
5.3.4	Use MIA to handle incomplete cases	49
5.3.5	Use within-study multiple imputation	50
5.3.6	Use multilevel multiple imputation with micemd	51
6	Other settings	53
6.1	Impact of number of multiple imputations	53
6.2	Impact of different proportions in different data	54
6.2.1	RCT: MCAR 0.1, RWE: MCAR 0.5	54
6.2.1.1	Use only complete cases (for logistic+linear regressions)	55
6.2.1.2	Use multilevel MI	56
6.2.1.3	Use MIA to handle incomplete cases	57
6.2.2	RCT: MCAR 0.05, RWE: MCAR 0.22	58
6.2.2.1	Use only complete cases (for logistic+linear regressions)	59
6.2.2.2	Use multilevel MI	60
6.2.2.3	Use MIA to handle incomplete cases	61
6.2.3	Equilibrated sample sizes - RCT: MCAR 0.1, RWE: MCAR 0.5	62
6.2.3.1	Use only complete cases (for logistic+linear regressions)	63
6.2.3.2	Use multilevel MI	64
6.2.3.3	Use MIA to handle incomplete cases	65
6.2.4	RCT: MAR 0.1, RWE: MAR 0.5	66
6.2.4.1	Use only complete cases (for logistic+linear regressions)	67
6.2.4.2	Use multilevel MI	68
6.2.4.3	Use MIA to handle incomplete cases	69
6.2.5	RCT: MAR 0.05, RWE: MAR 0.22	70
6.2.5.1	Use only complete cases (for logistic+linear regressions)	71
6.2.5.2	Use multilevel MI	72
6.2.5.3	Use MIA to handle incomplete cases	73
6.2.6	Equilibrated sample sizes - RCT: MAR 0.1, RWE: MAR 0.5	74
6.2.6.1	Use only complete cases (for logistic+linear regressions)	75
6.2.6.2	Use multilevel MI	76
6.2.6.3	Use MIA to handle incomplete cases	77

1 Preliminaries

```
knitr::opts_chunk$set(echo = TRUE, verbose = FALSE, warning = FALSE, message=FALSE, cache = FALSE)

# Clear any existing variables
```

```

rm(list = ls())

# Set seed for reproducibility
set.seed(1234)

# If not installed yet, you need to un-comment the following line once to install
# the genRCT package that allows to use the calibration weighting (CW) estimator
# install.packages("genRCT_0.1.0.tar.gz", repos = NULL)
access_genRCT <- require(genRCT) # calibration weighting estimator, implementation by Dong et al.

## Loading required package: genRCT
# Load implemented estimation functions from GitLab repository
if (access_genRCT) {
  source("estimators_and_simulations.R")
} else {
  source("estimators_and_simulations_wo_cw.R")
}

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##   select

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

## Loading required package: usethis
## Loading required package: mice

##
## Attaching package: 'mice'

## The following objects are masked from 'package:base':
##
##   cbind, rbind

##
## Attaching package: 'purrr'

## The following object is masked from 'package:pracma':
##
##   cross

## SHA-1 hash of file is a392b353c3ba88ecd276c2d94bd36009d5d40616

results_dir <- "./results/"
fig_dir <- "./figures/"

# Libraries
library(ggplot2)      # plots
library(dplyr)        # data frame tools

```

```

library(table1)      # table for baseline
library(wesanderson) # colors
library(naniar)      # visualize missing data

# number of repetitions in simulation
repetitions = 20
repetitions_long = 20 # to speed up the running time, reduce this number that sets the number of repetitions
fig_prefix <- paste0("rep", repetitions)
fig_prefix_long <- paste0("rep", repetitions_long)

n_range <- 1000*1:5

fig_prefix <- paste0("rep", repetitions, "_n", paste(n_range, sep="", collapse="_"))
fig_prefix_long <- paste0("rep", repetitions_long, "_n", paste(n_range, sep="", collapse="_"))

# For every possible choice of options (repetitions, repetitions_long, link, corX, rho, snr, etc.) the
# By default we assume the results have not yet been computed and saved
results_exist <- FALSE

```

Some of the options are kept fixed in these simulations but could be changed to alternative values.

```

# link function for outcome and selection score.
link <- links <- "linear" # alternatively, could be set to "non-linear"

# correlation coefficient for covariates
corX <- TRUE
rho <- 0.6
if (corX) {
  Sigma <- diag(1-rho, ncol = 4, nrow = 4) + matrix(rho, nrow = 4, ncol = 4)
} else {
  Sigma <- diag(4)
}

# off-set in selection score model
bs0 <- ifelse(identical(Sigma,diag(nrow(Sigma))), ifelse(link=="linear", -2.5, -0.92), ifelse(link=="li

# signal-to-noise ratio for selection score and outcome model
snr <- 5

```

2 Note on distributional shift

We first show a small example how the data looks like and visualize the distributional shift between the RCT and the target population.

```

one_simulation <- simulate_continuous(n = 1000, m = 10000, link=link, Sigma=Sigma, bs0=bs0,
                                     na_rct=list(mechanism="MNAR", prop_miss=0.2, idx_incomplete=rep(1,
                                     na_rwe=list(mechanism="MNAR_selfmask", prop_miss=c(0.5, 0.1, 0.1,

tau <- one_simulation$tau
one_simulation <- one_simulation$DF
sum(one_simulation$V==1)

## [1] 1091

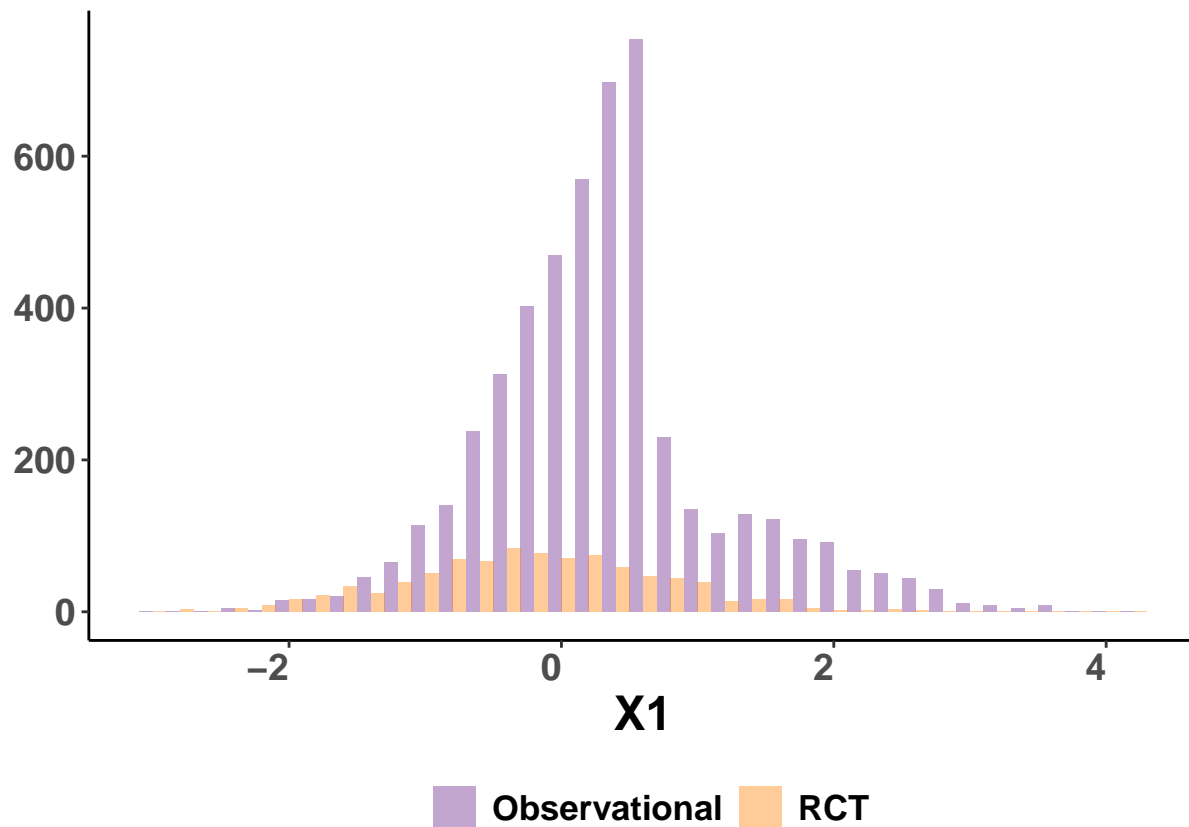
one_simulation$sample <- ifelse(one_simulation$V == 1, "RCT", "Observational")
baseline <- table1(~ X1 + X2 + X3 + X4 | sample, data = one_simulation, overall="Total")

```

baseline

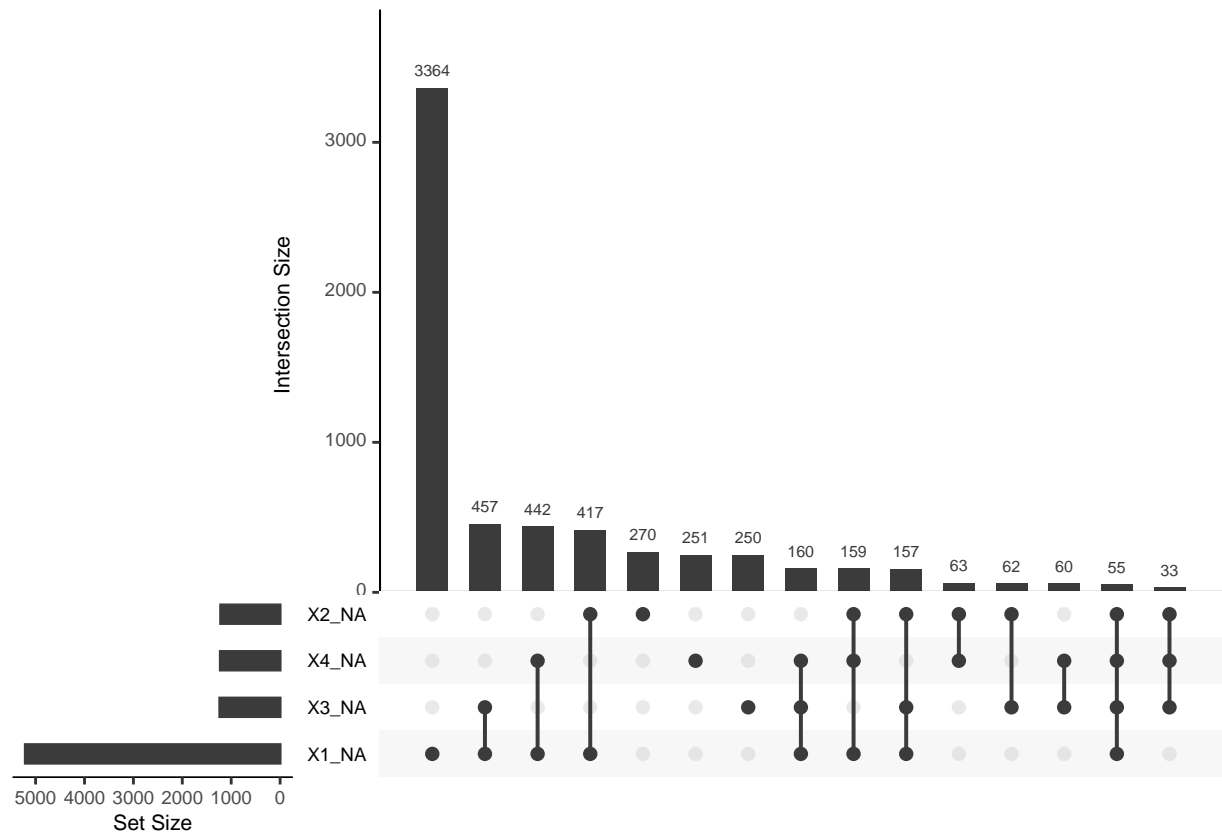
```
## [1] "<table class=\"Rtable1\">\n<thead>\n<tr>\n<th class='rowlabel firstrow lastrow'></th>\n<th class='collabel firstcol lastcol'></th>\n<tbody>\n<tr>\n<td>1</td>\n<td>2</td>\n<td>3</td>\n<td>4</td>\n<td>5</td>\n<td>6</td>\n<td>7</td>\n<td>8</td>\n<td>9</td>\n<td>10</td>\n<td>11</td>\n<td>12</td>\n<td>13</td>\n<td>14</td>\n<td>15</td>\n<td>16</td>\n<td>17</td>\n<td>18</td>\n<td>19</td>\n<td>20</td>\n<td>21</td>\n<td>22</td>\n<td>23</td>\n<td>24</td>\n<td>25</td>\n<td>26</td>\n<td>27</td>\n<td>28</td>\n<td>29</td>\n<td>30</td>\n<td>31</td>\n<td>32</td>\n<td>33</td>\n<td>34</td>\n<td>35</td>\n<td>36</td>\n<td>37</td>\n<td>38</td>\n<td>39</td>\n<td>40</td>\n<td>41</td>\n<td>42</td>\n<td>43</td>\n<td>44</td>\n<td>45</td>\n<td>46</td>\n<td>47</td>\n<td>48</td>\n<td>49</td>\n<td>50</td>\n<td>51</td>\n<td>52</td>\n<td>53</td>\n<td>54</td>\n<td>55</td>\n<td>56</td>\n<td>57</td>\n<td>58</td>\n<td>59</td>\n<td>60</td>\n<td>61</td>\n<td>62</td>\n<td>63</td>\n<td>64</td>\n<td>65</td>\n<td>66</td>\n<td>67</td>\n<td>68</td>\n<td>69</td>\n<td>70</td>\n<td>71</td>\n<td>72</td>\n<td>73</td>\n<td>74</td>\n<td>75</td>\n<td>76</td>\n<td>77</td>\n<td>78</td>\n<td>79</td>\n<td>80</td>\n<td>81</td>\n<td>82</td>\n<td>83</td>\n<td>84</td>\n<td>85</td>\n<td>86</td>\n<td>87</td>\n<td>88</td>\n<td>89</td>\n<td>90</td>\n<td>91</td>\n<td>92</td>\n<td>93</td>\n<td>94</td>\n<td>95</td>\n<td>96</td>\n<td>97</td>\n<td>98</td>\n<td>99</td>\n<td>100</td>\n</tr>\n</tbody>\n</table>\n"
```

```
ggplot(one_simulation, aes(x = X1, group = sample, fill = sample)) +  
  geom_histogram(binwidth = 0.2, alpha=0.4, position="dodge") +  
  scale_fill_manual(values=c("darkorchid4", "darkorange1")) +  
  theme_classic() +  
  theme(legend.title = element_blank(), legend.position = "bottom",  
        legend.box = "horizontal", legend.text = element_text(size=13,  
                                                                face="bold")) +  
  ylab("") + # no title in legend  
  theme(axis.text = element_text(vjust = 0.5, hjust=1, size=14, face="bold"),  
        axis.title.x = element_text(size=18, face="bold"))
```

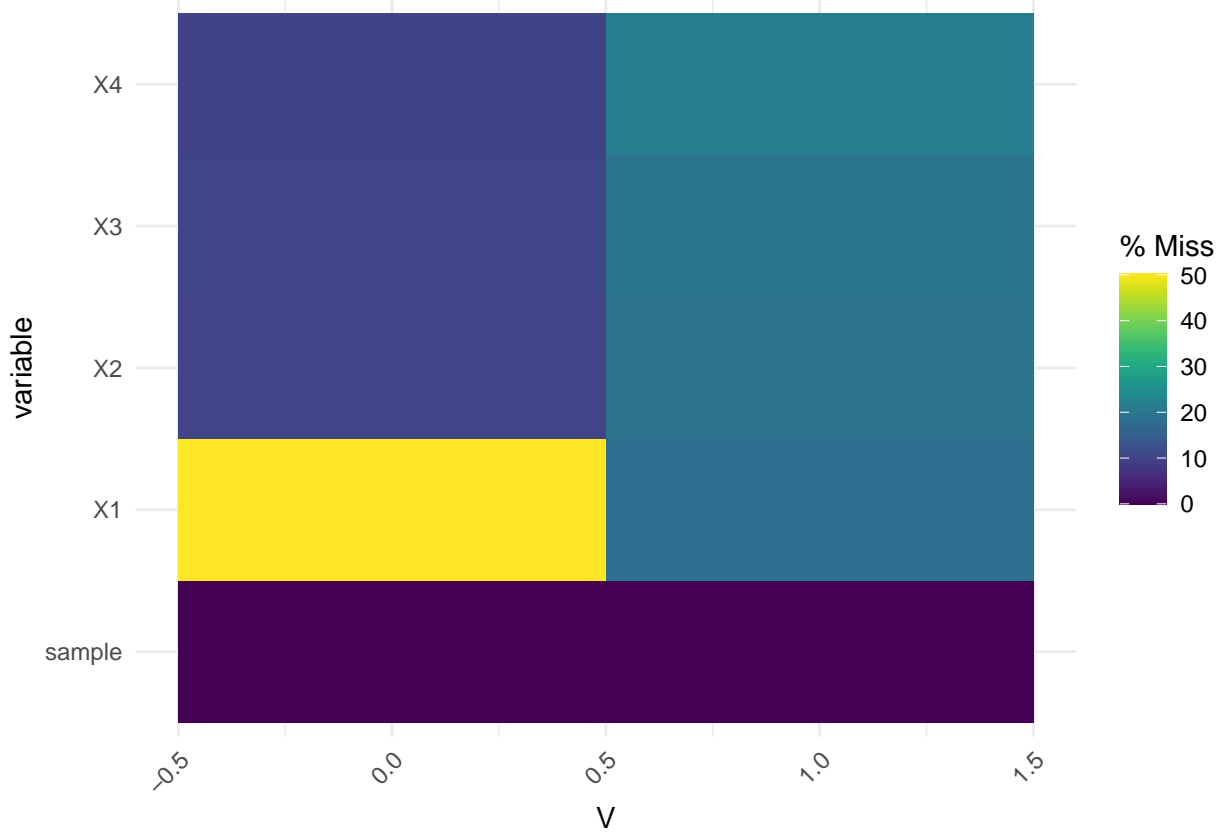


We can check that the missing values are differently distributed in the two studies:

```
gg_miss_upset(one_simulation[,!names(one_simulation) %in% c("A", "Y")])
```



```
gg_miss_fct(x = one_simulation[, !names(one_simulation) %in% c("A", "Y")], fct = V)
```



3 Simulation setting

We use the following simulation setting (see the submitted paper for more details):

$$\text{logit} \{ \pi_V(X) \} = -2.5 - 0.5X_1 - 0.3X_2 - 0.5X_3 - 0.4X_4,$$

where every X_j is drawn from a normal distribution $\mathcal{N}(1, 1)$. This model specifies the trial selection, V .

Unless specified differently, the outcome is generated as follows:

$$Y(a) = -100 + 27.4aX_1 + 13.7X_2 + 13.7X_3 + 13.7X_4 + \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(0, 1)$$

and the missing covariate values, indexed by the mask $M = 1 - R \in \{0, 1\}^{n \times p}$, are sampled using one of the three mechanisms below:

- MCAR such that $P(M_{ij} = 1) = 0.02$;
- MAR such that $P(M_{i,j} = 1 | X_{i,-j}) = 0.2$.
- MNAR such that $P(M_{i,j} = 1 | X_{i,j})$ depends on $X_{i,j}$ for each j .

We do not modify the treatment assignment mechanism since by assumption it is independent of everything and generally constant for all individuals ($e_1(x) = e_1 = 0.5$).

4 Classical ignorability + missingness assumptions

Below, we compare the methods in the setting where we assume the classical ignorability from the full data case and make assumptions about the missing values mechanism.

In practice that means that the missing values occur afterward after inclusion and randomization.

4.1 MCAR

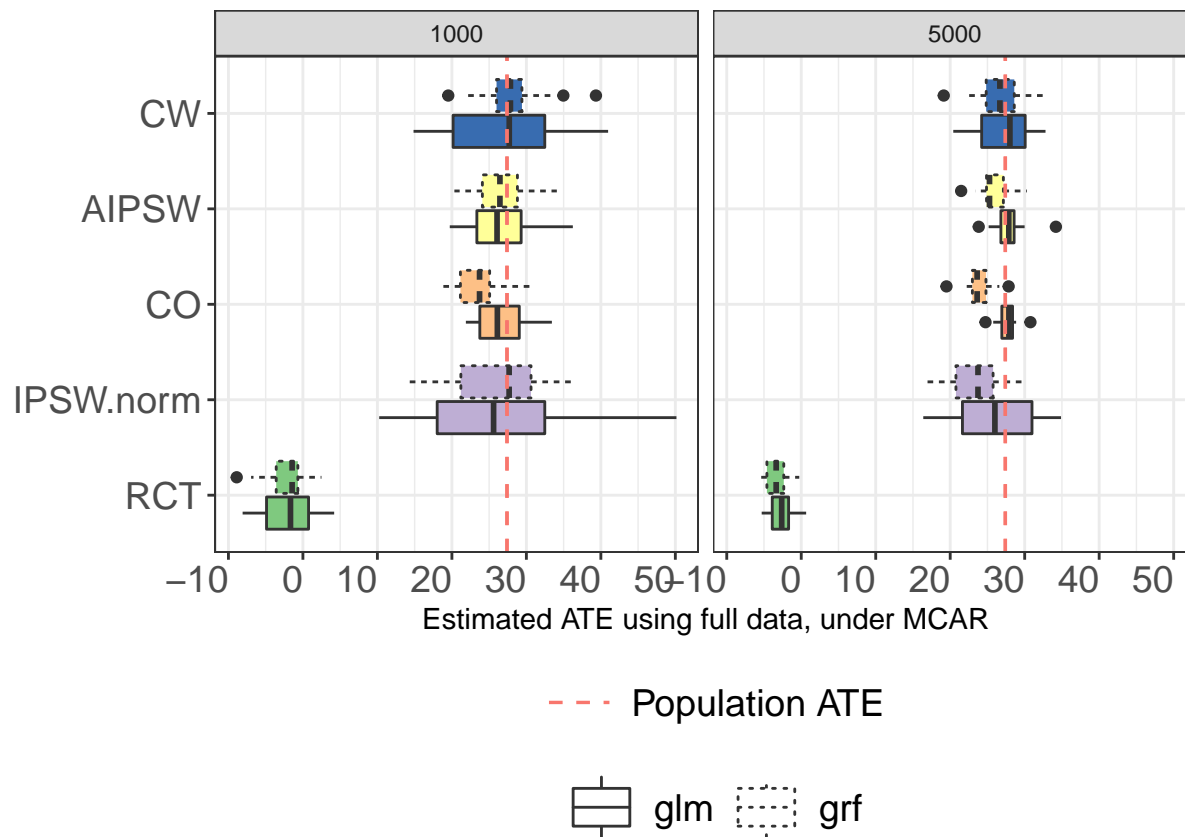
```
prop.miss <- 0.2
mechanism <- "MCAR"
```

4.1.1 On full data

```
methods <- c("glm", "grf")
if (!results_exist) {
  results_full <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomp=idx_incomp),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE, full_data=T,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_full <- rbind(results_full, tmp)
    }
  }
}
```

```
## # A tibble: 20 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -29.5   1000 glm  linear
## 2 RCT        -30.1   5000 glm  linear
## 3 RCT        -29.5   1000 grf  linear
## 4 RCT        -30.8   5000 grf  linear
## 5 IPSW.norm  -1.51   1000 glm  linear
## 6 IPSW.norm  -1.30   5000 glm  linear
## 7 IPSW.norm  -1.32   1000 grf  linear
## 8 IPSW.norm  -4.01   5000 grf  linear
## 9 CO         -0.872  1000 glm  linear
## 10 CO         0.263  5000 glm  linear
## 11 CO        -3.97   1000 grf  linear
## 12 CO        -3.58   5000 grf  linear
## 13 AIPSW     -0.625  1000 glm  linear
## 14 AIPSW      0.434  5000 glm  linear
## 15 AIPSW     -0.968  1000 grf  linear
## 16 AIPSW     -1.70   5000 grf  linear
## 17 CW        -0.552  1000 glm  linear
## 18 CW        -0.112  5000 glm  linear
## 19 CW         0.709  1000 grf  linear
## 20 CW        -0.799  5000 grf  linear
```

4.1.2 Use only complete cases (for logistic+linear regressions)

```

methods <- c("glm")

if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomp=idx_incomp),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T,verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_cc <- rbind(results_cc, tmp)
    }
  }
}

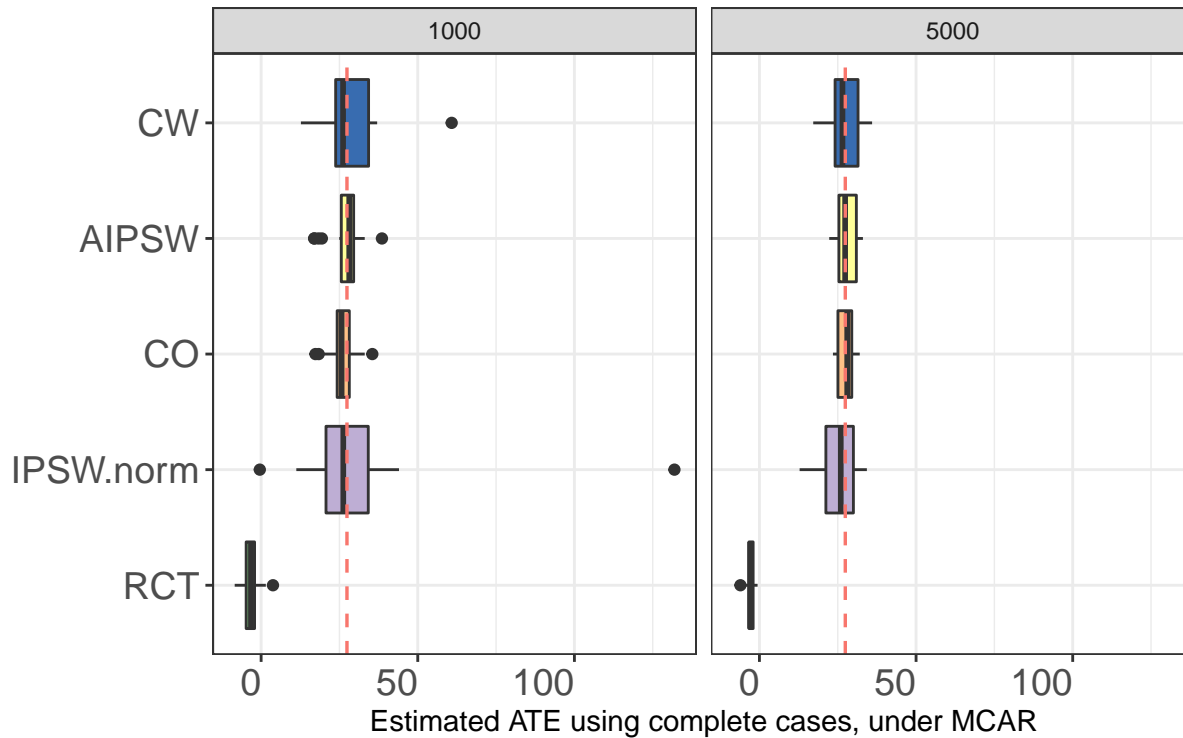
```

```

## # A tibble: 10 x 5
##   variable    bias    n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT       -30.8   1000 glm  linear
## 2 RCT       -30.3   5000 glm  linear
## 3 IPSW.norm   3.08   1000 glm  linear

```

```
## 4 IPSW.norm -2.20 5000 glm linear
## 5 CO -1.34 1000 glm linear
## 6 CO 0.0749 5000 glm linear
## 7 AIPSW -0.360 1000 glm linear
## 8 AIPSW 0.445 5000 glm linear
## 9 CW 1.19 1000 glm linear
## 10 CW -0.426 5000 glm linear
```



--- Population ATE

4.1.3 Use EM to handle incomplete cases

Now we do not throw away incomplete observations but rather adapt the estimation methods to take them into account. We start by using EM to handle (ignorable) missing values in linear and logistic regressions.

```
methods <- c("glm")

if (!results_exist) {
  results_em <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         method="glm", nb_strat=1,
                                         complete_cases=F,
                                         verbose=T,verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_em <- rbind(results_em, tmp)
    }
  }
}
```

```

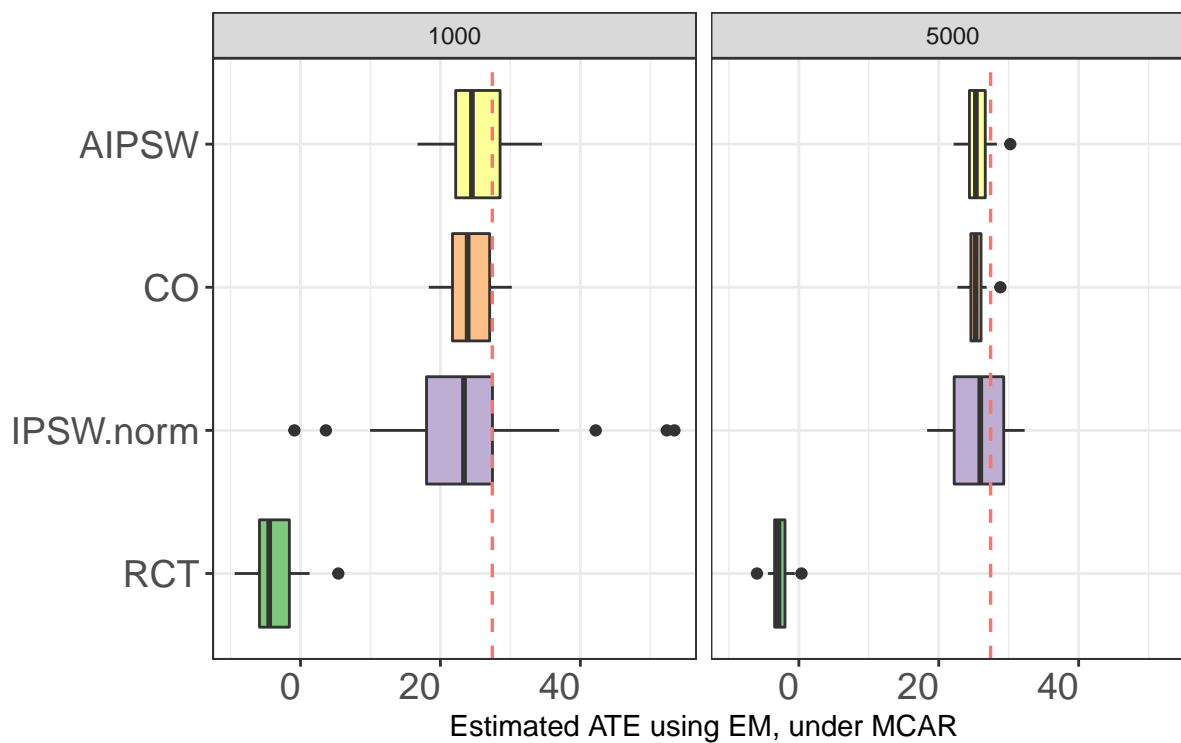
}
}
}

```

```

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -31.2  1000 glm   linear
## 2 RCT        -30.1  5000 glm   linear
## 3 IPSW.norm  -2.80  1000 glm   linear
## 4 IPSW.norm  -1.65  5000 glm   linear
## 5 CO         -3.39  1000 glm   linear
## 6 CO         -2.07  5000 glm   linear
## 7 AIPSW      -2.16  1000 glm   linear
## 8 AIPSW      -1.86  5000 glm   linear

```



--- Population ATE

4.1.4 Use MIA to handle incomplete cases

```

methods <- c("grf")

if (!results_exist) {
  results_grf <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom

```

```

method=methods, nb_strat=1,
complete_cases=FALSE,
verbose=T)

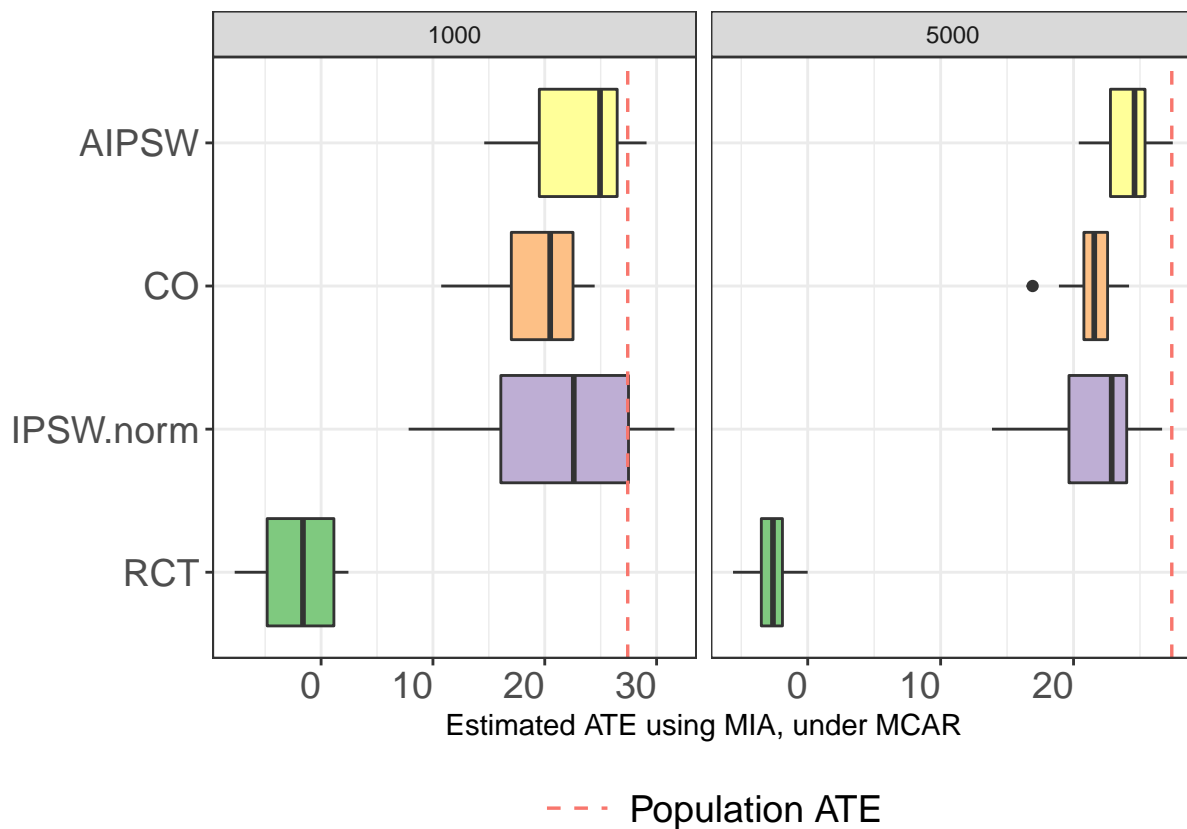
tmp$n <- n
tmp$link <- link
results_grf <- rbind(results_grf, tmp)
}
}
}

```

```

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -29.3  1000 grf    linear
## 2 RCT        -30.2  5000 grf    linear
## 3 IPSW.norm   -6.08  1000 grf    linear
## 4 IPSW.norm   -5.45  5000 grf    linear
## 5 CO          -7.96  1000 grf    linear
## 6 CO          -5.89  5000 grf    linear
## 7 AIPSW        -4.14  1000 grf    linear
## 8 AIPSW        -3.20  5000 grf    linear

```



4.1.5 Use within-study multiple imputation

```

methods <- c("glm")
nb_mi <- c(5, 10)

```

```

if (!results_exist) {
  results_mi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi,
                                         verbose=T,verbose_intern = F)

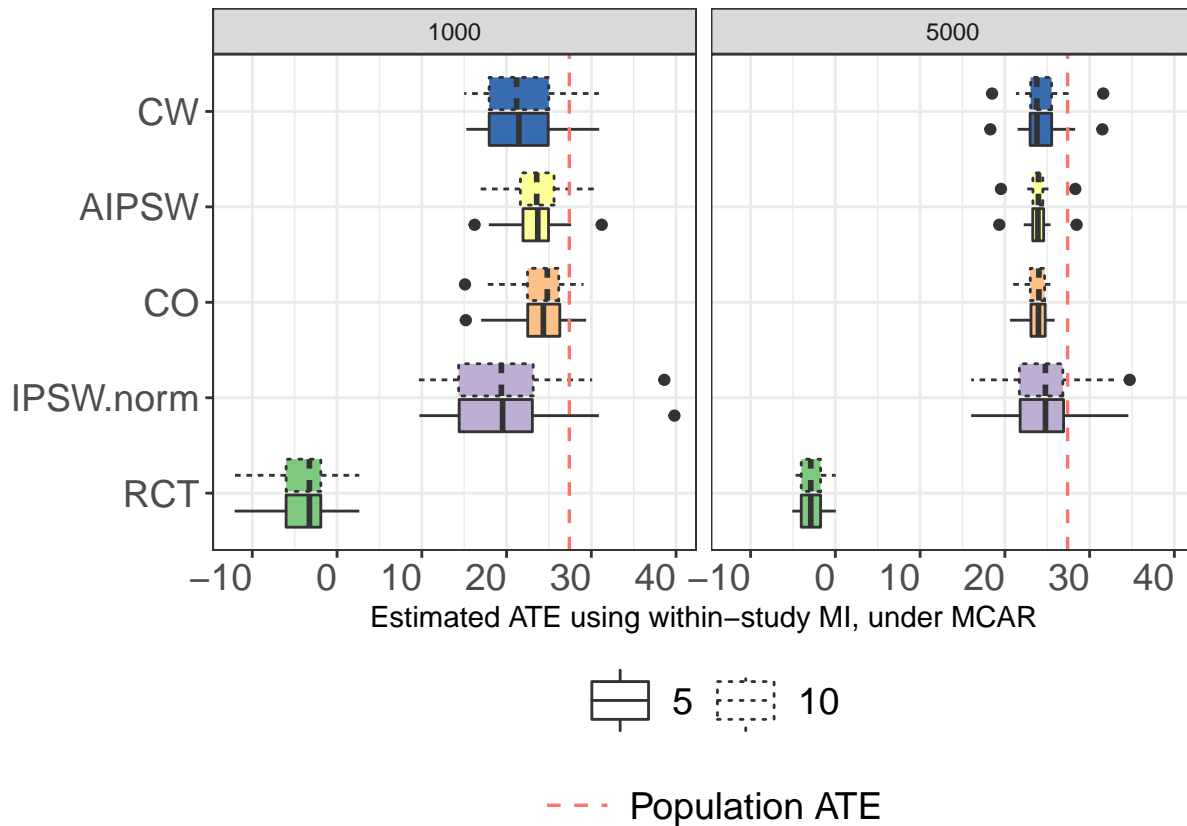
      tmp$n <- n
      tmp$link <- link
      results_mi <- rbind(results_mi, tmp)
    }
  }
}

```

```

## # A tibble: 10 x 6
##   variable    bias nb_mi      n method link
##   <fct>      <dbl> <dbl> <dbl> <chr>  <chr>
## 1 RCT        -31.1    10  1000 glm    linear
## 2 RCT        -30.3    10  5000 glm    linear
## 3 IPSW.norm  -7.40    10  1000 glm    linear
## 4 IPSW.norm  -2.73    10  5000 glm    linear
## 5 CO         -3.34    10  1000 glm    linear
## 6 CO         -3.63    10  5000 glm    linear
## 7 AIPSW      -3.93    10  1000 glm    linear
## 8 AIPSW      -3.50    10  5000 glm    linear
## 9 CW         -5.32    10  1000 glm    linear
## 10 CW        -3.16    10  5000 glm    linear

```



4.1.6 Use multilevel multiple imputation with micemd

```

methods <- c("glm")
nb_mi <- c(5, 10)

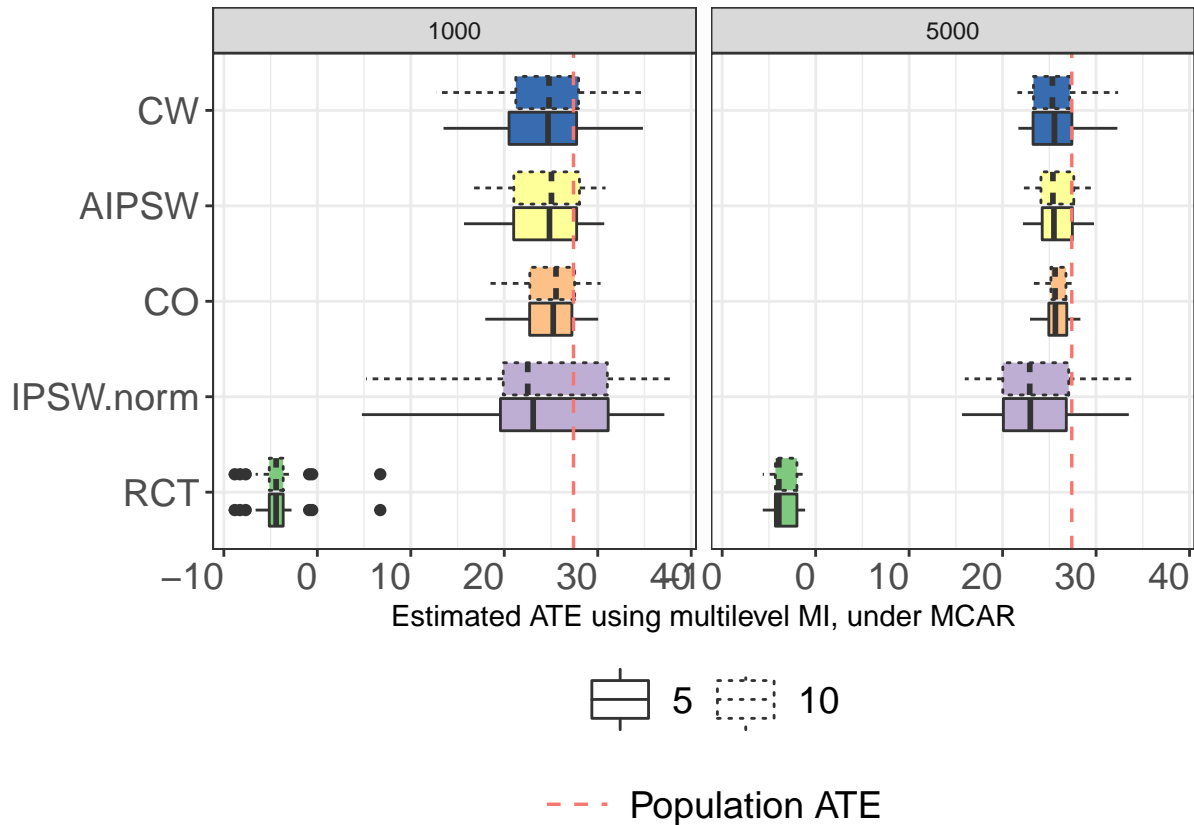
if (!results_exist) {
  results_mi_alt <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomp=idx_incomp),
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      tmp$strategy <- "multi-level-woY"
      results_mi_alt <- rbind(results_mi_alt, tmp)
    }
  }
}

## [1] "results_rep20_noCIS_linklinear_snr5_MCAR_propNA0.2_corTRUE_n1000_2000_3000_4000_5000_multilevel"
## # A tibble: 10 x 7
##   variable    bias nb_mi strategy          n method link
##   <fct>      <dbl> <dbl> <chr>          <dbl> <chr> <chr>

```

```
## 1 RCT      -31.5    10 multi-level-woY 1000 glm    linear
## 2 RCT      -30.9    10 multi-level-woY 5000 glm    linear
## 3 IPSW.norm -4.92   10 multi-level-woY 1000 glm    linear
## 4 IPSW.norm -3.76   10 multi-level-woY 5000 glm    linear
## 5 CO       -2.40    10 multi-level-woY 1000 glm    linear
## 6 CO       -1.51    10 multi-level-woY 5000 glm    linear
## 7 AIPSW    -2.98    10 multi-level-woY 1000 glm    linear
## 8 AIPSW    -1.60    10 multi-level-woY 5000 glm    linear
## 9 CW       -2.84    10 multi-level-woY 1000 glm    linear
## 10 CW      -1.79    10 multi-level-woY 5000 glm    linear
```



4.2 MAR

```
prop.miss <- 0.2
mechanism <- "MAR"
```

4.2.1 On full data

```
methods <- c("glm", "grf")

if (!results_exist) {
  results_full <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
        p = 4, Sigma = Sigma, snr=snr,
        na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomp=idx_incomp))
    }
  }
}
```

```

method=methods, nb_strat=1,
complete_cases=TRUE, full_data=T,
verbose=T, verbose_intern = F)

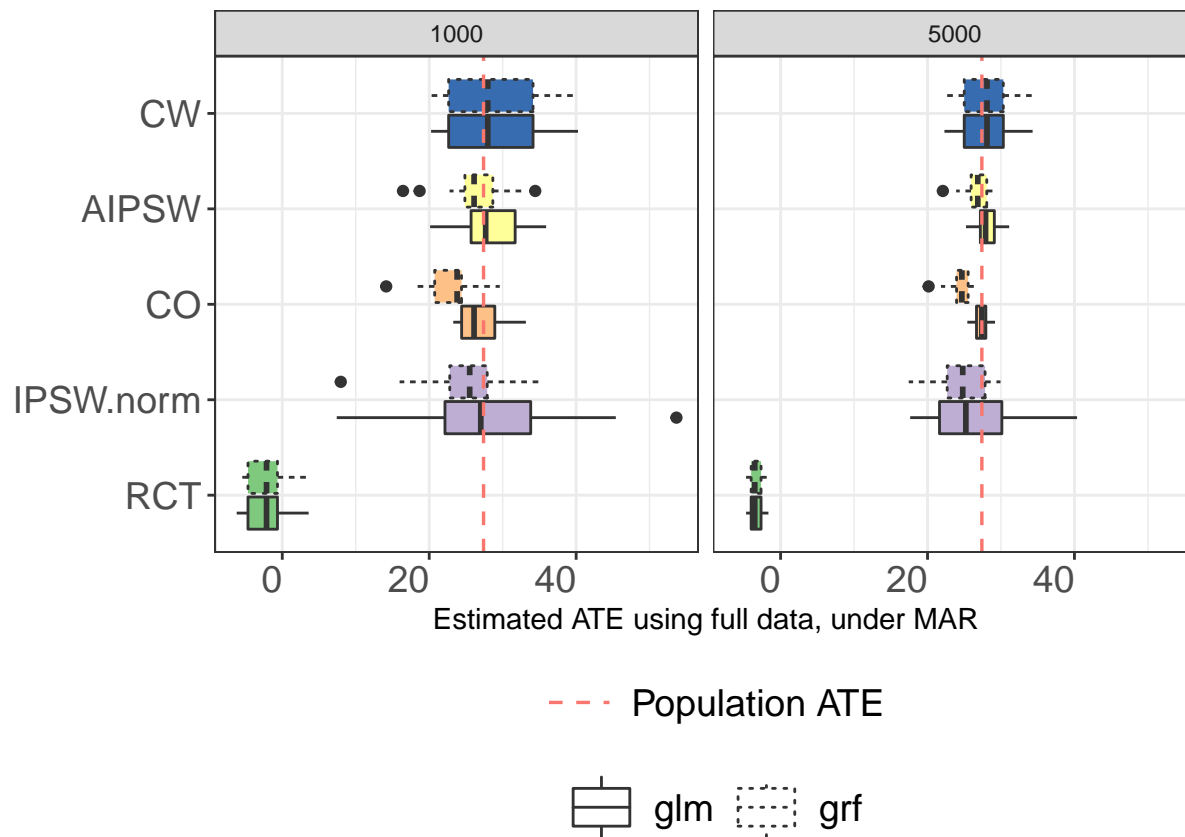
tmp$n <- n
tmp$link <- link
results_full <- rbind(results_full, tmp)
}
}
}

```

```

## # A tibble: 20 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -29.6   1000 glm   linear
## 2 RCT        -29.6   1000 grf    linear
## 3 RCT        -30.7   5000 glm   linear
## 4 RCT        -30.7   5000 grf    linear
## 5 IPSW.norm    1.47   1000 glm   linear
## 6 IPSW.norm   -2.77   1000 grf    linear
## 7 IPSW.norm   -1.33   5000 glm   linear
## 8 IPSW.norm   -2.43   5000 grf    linear
## 9 CO         -0.564   1000 glm   linear
## 10 CO         -4.60   1000 grf    linear
## 11 CO        -0.0844  5000 glm   linear
## 12 CO        -2.94   5000 grf    linear
## 13 AIPSW       1.02   1000 glm   linear
## 14 AIPSW      -0.923   1000 grf    linear
## 15 AIPSW       0.634   5000 glm   linear
## 16 AIPSW      -0.635   5000 grf    linear
## 17 CW         1.11   1000 glm   linear
## 18 CW         1.11   1000 grf    linear
## 19 CW         0.471   5000 glm   linear
## 20 CW         0.471   5000 grf    linear

```

4.2.2 Use only complete cases (for logistic+linear regressions)

```

methods <- c("glm")

if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_cc <- rbind(results_cc, tmp)
    }
  }
}

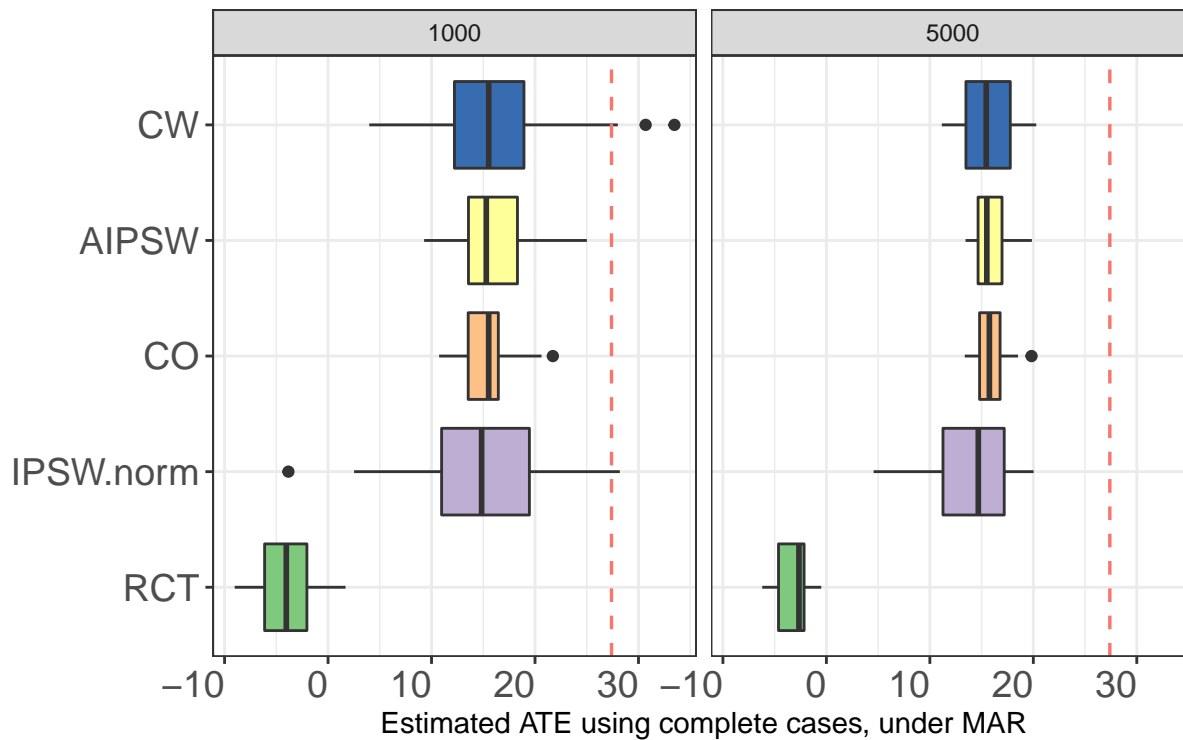
```

```

## # A tibble: 10 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -31.4  1000 glm   linear
## 2 RCT        -30.6  5000 glm   linear
## 3 IPSW.norm -12.8  1000 glm   linear

```

```
## 4 IPSW.norm -13.2 5000 glm linear
## 5 CO -11.9 1000 glm linear
## 6 CO -11.5 5000 glm linear
## 7 AIPSW -11.1 1000 glm linear
## 8 AIPSW -11.5 5000 glm linear
## 9 CW -10.5 1000 glm linear
## 10 CW -11.7 5000 glm linear
```



--- Population ATE

4.2.3 Use EM to handle incomplete cases

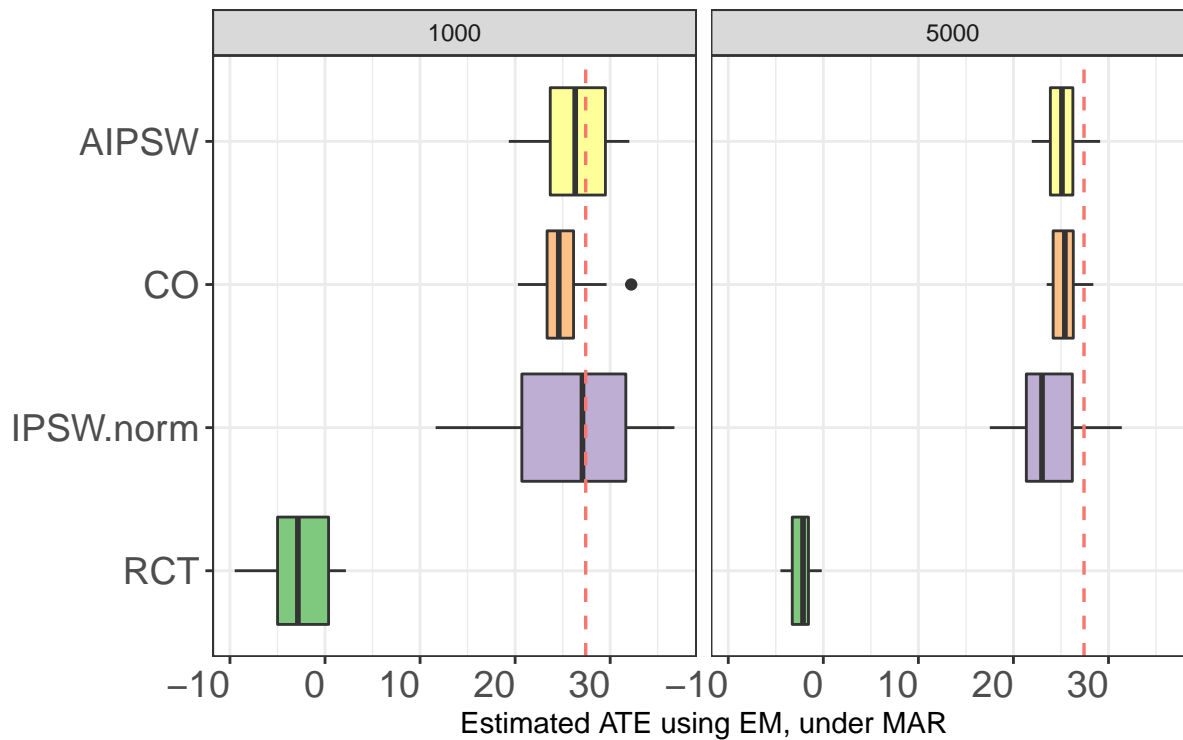
```
methods <- c("glm")

if (!results_exist) {
  results_em <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom=idx_incom),
                                         method="glm", nb_strat=1,
                                         complete_cases=F,
                                         verbose=T,verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_em <- rbind(results_em, tmp)
    }
  }
}
```

```
}
```

```
## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -29.9  1000 glm   linear
## 2 RCT        -29.7  5000 glm   linear
## 3 IPSW.norm   -1.31  1000 glm   linear
## 4 IPSW.norm   -3.50  5000 glm   linear
## 5 CO          -2.53  1000 glm   linear
## 6 CO          -1.95  5000 glm   linear
## 7 AIPSW        -1.04  1000 glm   linear
## 8 AIPSW        -2.24  5000 glm   linear
```



4.2.4 Use MIA to handle incomplete cases

```
methods <- c("grf")

if (!results_exist) {
  results_grf <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom=idx_incom),
                                         method=methods, nb_strat=1,
                                         complete_cases=FALSE,
```

```

                                verbose=T)

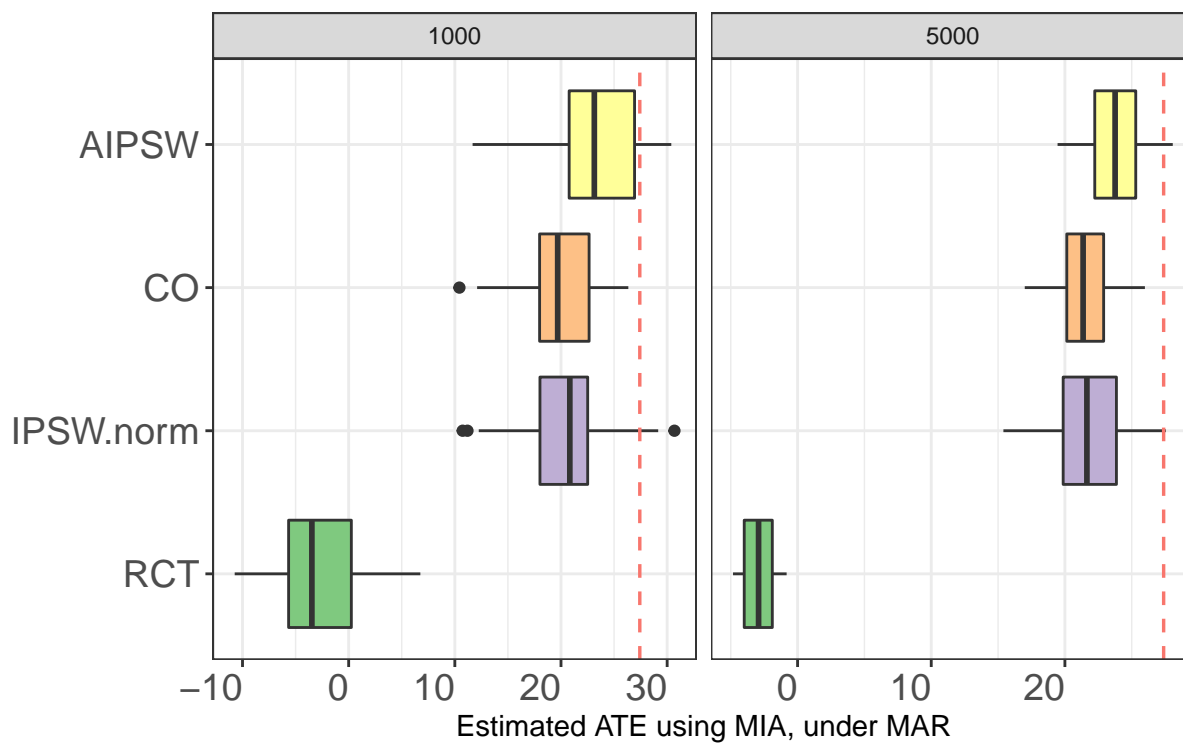
  tmp$n <- n
  tmp$link <- link
  results_grf <- rbind(results_grf, tmp)
}
}
}

```

```
## [1] "results_rep20_noCIS_noCIO_linklinear_snr5_MAR_propNA0.2_corTRUE_n1000_2000_3000_4000_5000_mia_g"
```

```
## # A tibble: 8 x 5
```

```
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -29.8  1000 grf    linear
## 2 RCT        -30.3  5000 grf    linear
## 3 IPSW.norm   -6.86  1000 grf    linear
## 4 IPSW.norm   -5.69  5000 grf    linear
## 5 CO          -7.65  1000 grf    linear
## 6 CO          -5.94  5000 grf    linear
## 7 AIPSW        -4.15  1000 grf    linear
## 8 AIPSW        -3.82  5000 grf    linear
```



--- Population ATE

4.2.5 Use within-study multiple imputation

```

methods <- c("glm")
nb_mi <- c(5, 10)

```

```

if (!results_exist) {
  results_mi <- c()
  for (link in links){
    bs0 <- ifelse(identical(Sigma,diag(nrow(Sigma))), ifelse(link=="linear", -2.5, -0.92), ifelse(link=="
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom
                                         method=methods, nb_strat=1, bin="quantile",
                                         do_mi=T, nb_mi=nb_mi,
                                         verbose=T,verbose_intern = F)

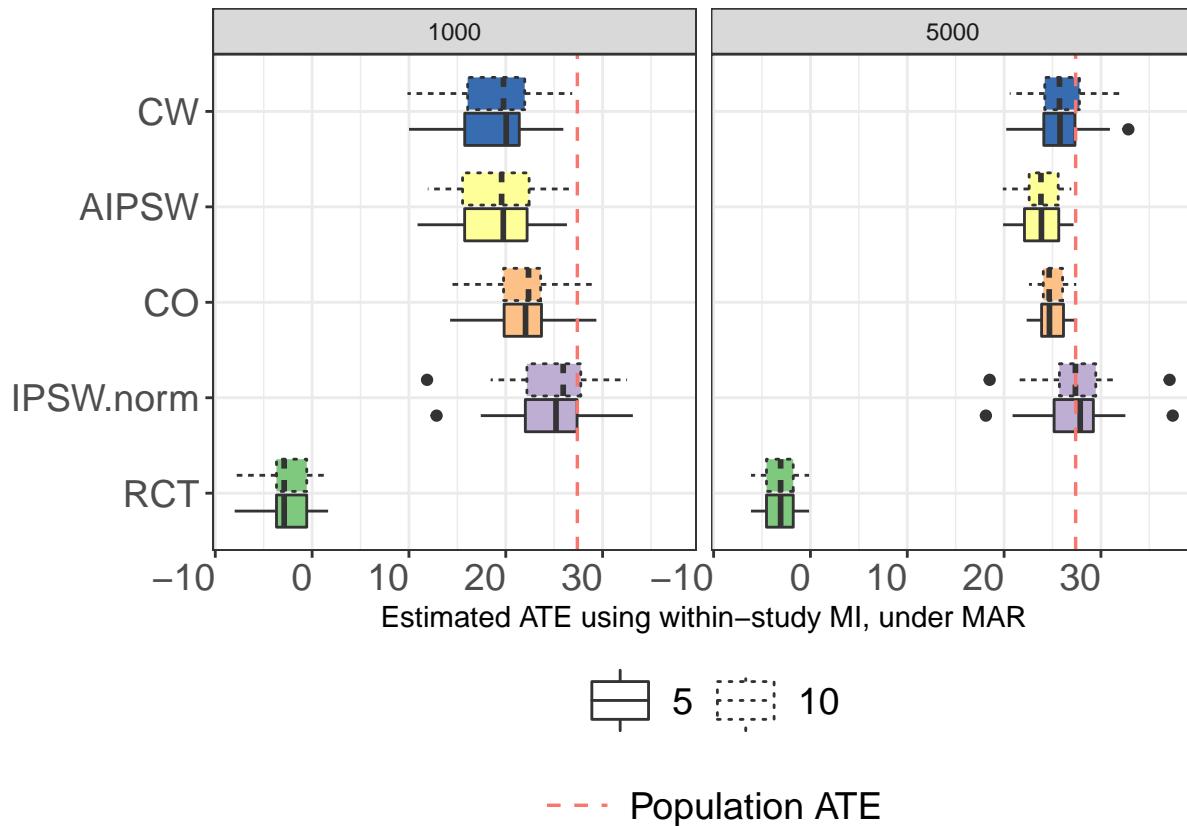
      tmp$n <- n
      tmp$link <- link
      results_mi <- rbind(results_mi, tmp)
    }
  }
}

```

```

## # A tibble: 10 x 6
##   variable      bias nb_mi      n method link
##   <fct>      <dbl> <dbl> <dbl> <chr>  <chr>
## 1 RCT        -30.0     10  1000 glm    linear
## 2 RCT        -30.4     10  5000 glm    linear
## 3 IPSW.norm  -2.81      10  1000 glm    linear
## 4 IPSW.norm  -0.0718    10  5000 glm    linear
## 5 CO         -5.76      10  1000 glm    linear
## 6 CO         -2.53      10  5000 glm    linear
## 7 AIPSW      -8.37      10  1000 glm    linear
## 8 AIPSW      -3.62      10  5000 glm    linear
## 9 CW         -8.40      10  1000 glm    linear
## 10 CW        -1.54      10  5000 glm    linear

```



4.2.6 Use multilevel multiple imputation with micemd

```

methods <- c("glm")
nb_mi <- c(5, 10)

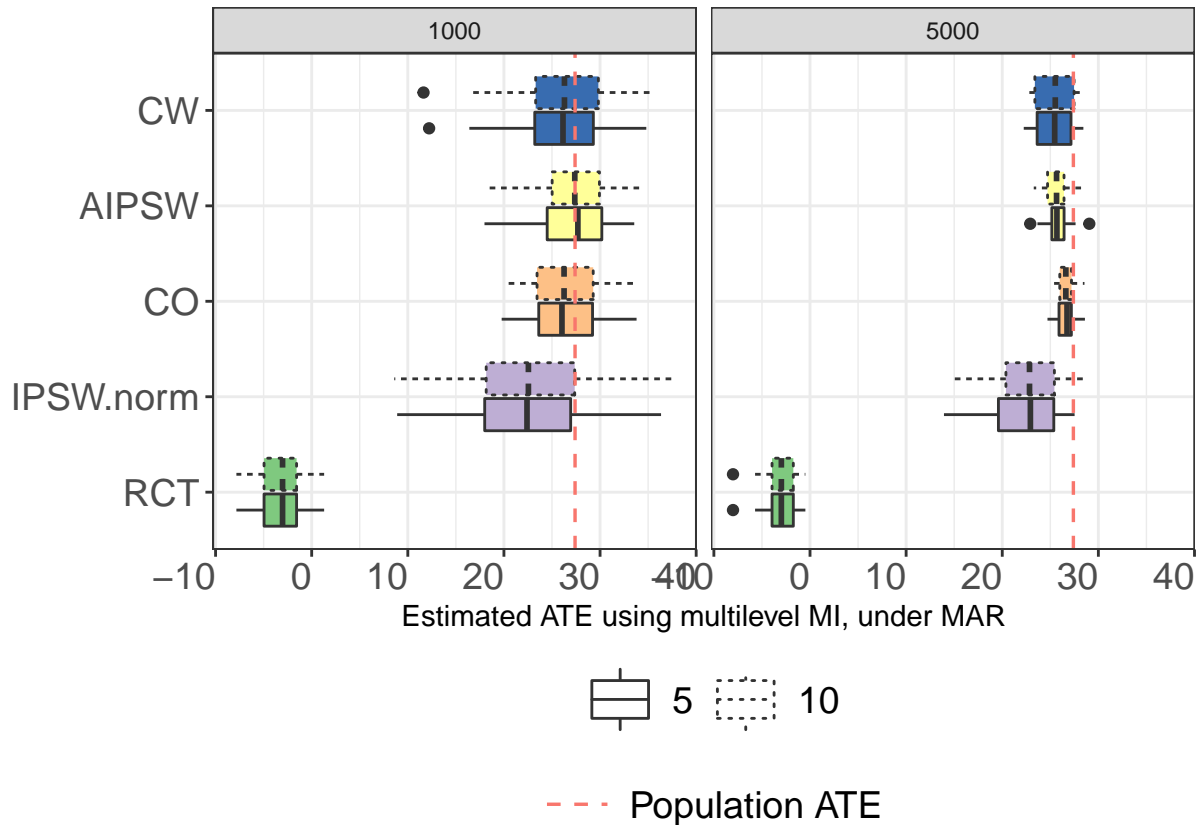
if (!results_exist) {
  results_mi_alt <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      tmp$strategy <- "multi-level-woY"
      results_mi_alt <- rbind(results_mi_alt, tmp)
    }
  }
}

## [1] "results_rep20_noCIS_noCIO_linklinear_snr5_MAR_propNA0.2_corTRUE_n1000_2000_3000_4000_5000_multi..."
## # A tibble: 10 x 7
##   variable      bias nb_mi strategy          n method link
##   <fct>        <dbl> <dbl> <chr>        <dbl> <chr> <chr>

```

##	1	RCT	-30.5	10	multi-level-woY	1000	glm	linear
##	2	RCT	-30.5	10	multi-level-woY	5000	glm	linear
##	3	IPSW.norm	-4.70	10	multi-level-woY	1000	glm	linear
##	4	IPSW.norm	-4.87	10	multi-level-woY	5000	glm	linear
##	5	CO	-0.853	10	multi-level-woY	1000	glm	linear
##	6	CO	-0.773	10	multi-level-woY	5000	glm	linear
##	7	AIPSW	-0.638	10	multi-level-woY	1000	glm	linear
##	8	AIPSW	-1.64	10	multi-level-woY	5000	glm	linear
##	9	CW	-1.29	10	multi-level-woY	1000	glm	linear
##	10	CW	-1.92	10	multi-level-woY	5000	glm	linear



4.3 MNAR (self-mask)

```
prop.miss <- 0.2
mechanism <- "MNAR_selfmask"
```

4.3.1 On full data

```
methods <- c("glm", "grf")

if (!results_exist) {
  results_full <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
        p = 4, Sigma = Sigma, snr=snr,
        na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor))
    }
  }
}
```

```

method=methods, nb_strat=1,
complete_cases=TRUE, full_data=T,
verbose=T, verbose_intern = F)

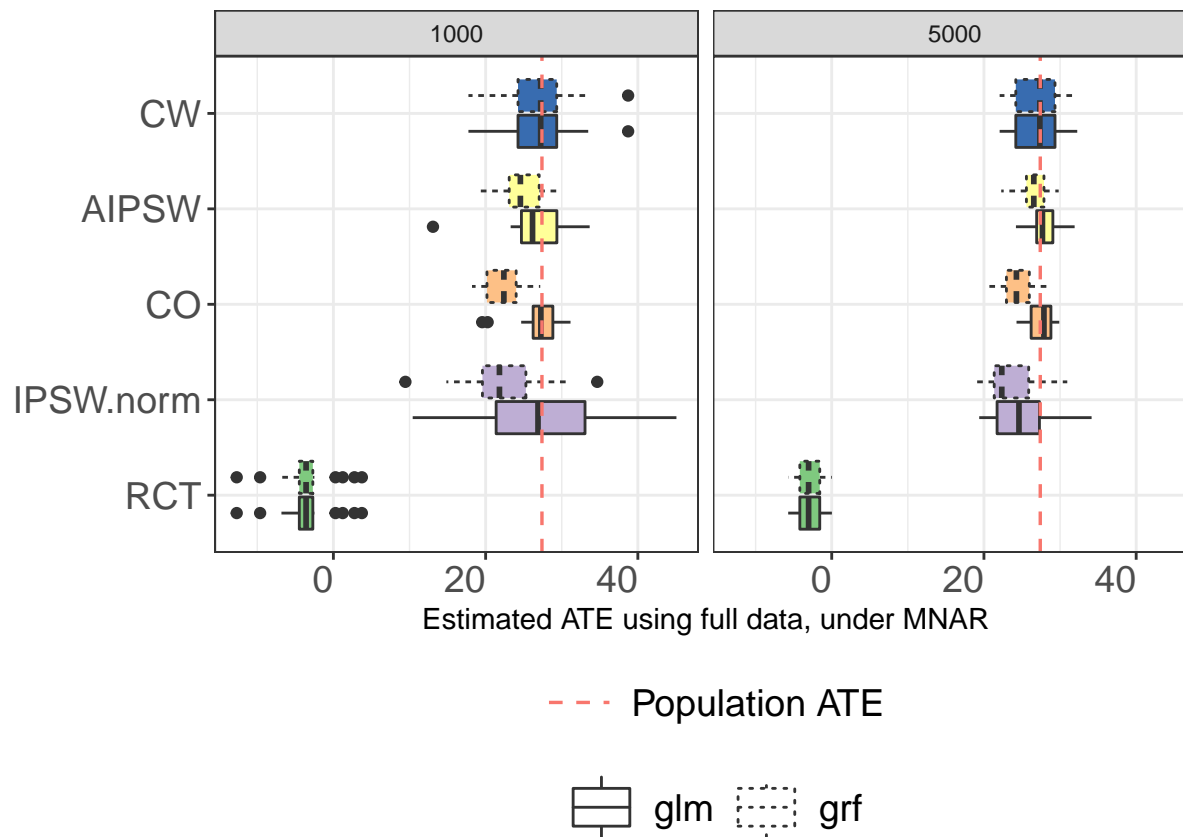
tmp$n <- n
tmp$link <- link
results_full <- rbind(results_full, tmp)
}
}
}

```

```

## # A tibble: 20 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -30.8   1000 glm   linear
## 2 RCT        -30.8   1000 grf   linear
## 3 RCT        -30.4   5000 glm   linear
## 4 RCT        -30.4   5000 grf   linear
## 5 IPSW.norm  -0.948   1000 glm   linear
## 6 IPSW.norm  -5.17    1000 grf   linear
## 7 IPSW.norm  -2.22    5000 glm   linear
## 8 IPSW.norm  -3.66    5000 grf   linear
## 9 CO         -0.472   1000 glm   linear
## 10 CO        -5.11    1000 grf   linear
## 11 CO        -0.0101  5000 glm   linear
## 12 CO        -3.05    5000 grf   linear
## 13 AIPSW     -0.790   1000 glm   linear
## 14 AIPSW     -2.68    1000 grf   linear
## 15 AIPSW      0.473   5000 glm   linear
## 16 AIPSW     -0.864   5000 grf   linear
## 17 CW        -0.227   1000 glm   linear
## 18 CW        -0.227   1000 grf   linear
## 19 CW        -0.390   5000 glm   linear
## 20 CW        -0.390   5000 grf   linear

```

4.3.2 Use only complete cases (for logistic+linear regressions)

```

methods <- c("glm")

if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T,verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_cc <- rbind(results_cc, tmp)
    }
  }
}

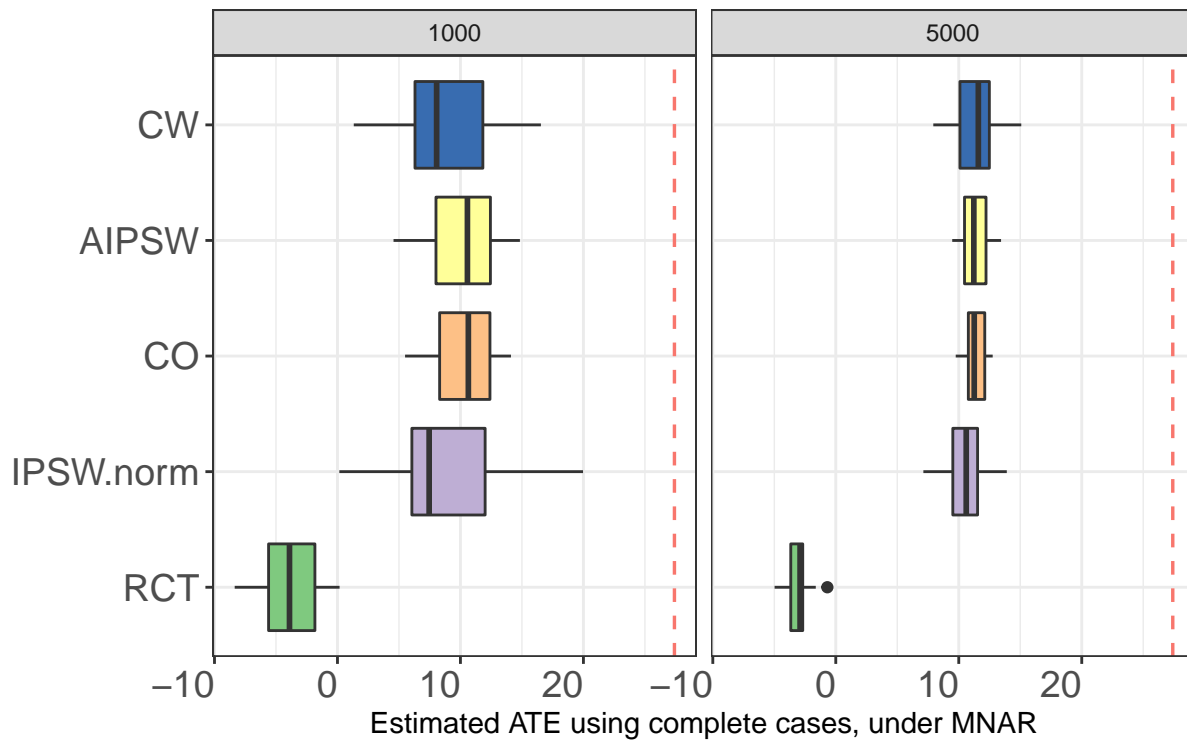
```

```

## # A tibble: 10 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -31.1  1000 glm   linear
## 2 RCT        -30.5  5000 glm   linear
## 3 IPSW.norm -18.4  1000 glm   linear

```

```
## 4 IPSW.norm -16.9 5000 glm linear
## 5 CO        -17.0 1000 glm linear
## 6 CO        -16.0 5000 glm linear
## 7 AIPSW     -17.1 1000 glm linear
## 8 AIPSW     -16.1 5000 glm linear
## 9 CW        -18.6 1000 glm linear
## 10 CW       -16.0 5000 glm linear
```



--- Population ATE

4.3.3 Use EM to handle incomplete cases

```
methods <- c("glm")

if (!results_exist) {
  results_em <- c()
  for (link in links){
    for (n in n_range){
      tmp <- NULL
      try(tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                             p = 4, Sigma = Sigma, snr = snr,
                                             na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                             method="glm", nb_strat=1,
                                             complete_cases=F,
                                             verbose=T,verbose_intern = F))

      if (!is.null(tmp)){
        tmp$n <- n
        tmp$link <- link
        results_em <- rbind(results_em, tmp)
      }
    }
  }
}
```

```

    }
  }
}

```

4.3.4 Use MIA to handle incomplete cases

```

methods <- c("grf")

if (!results_exist) {
  results_grf <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom=idx_incom),
                                         method=methods, nb_strat=1,
                                         complete_cases=FALSE,
                                         verbose=T)

      tmp$n <- n
      tmp$link <- link
      results_grf <- rbind(results_grf, tmp)
    }
  }
}

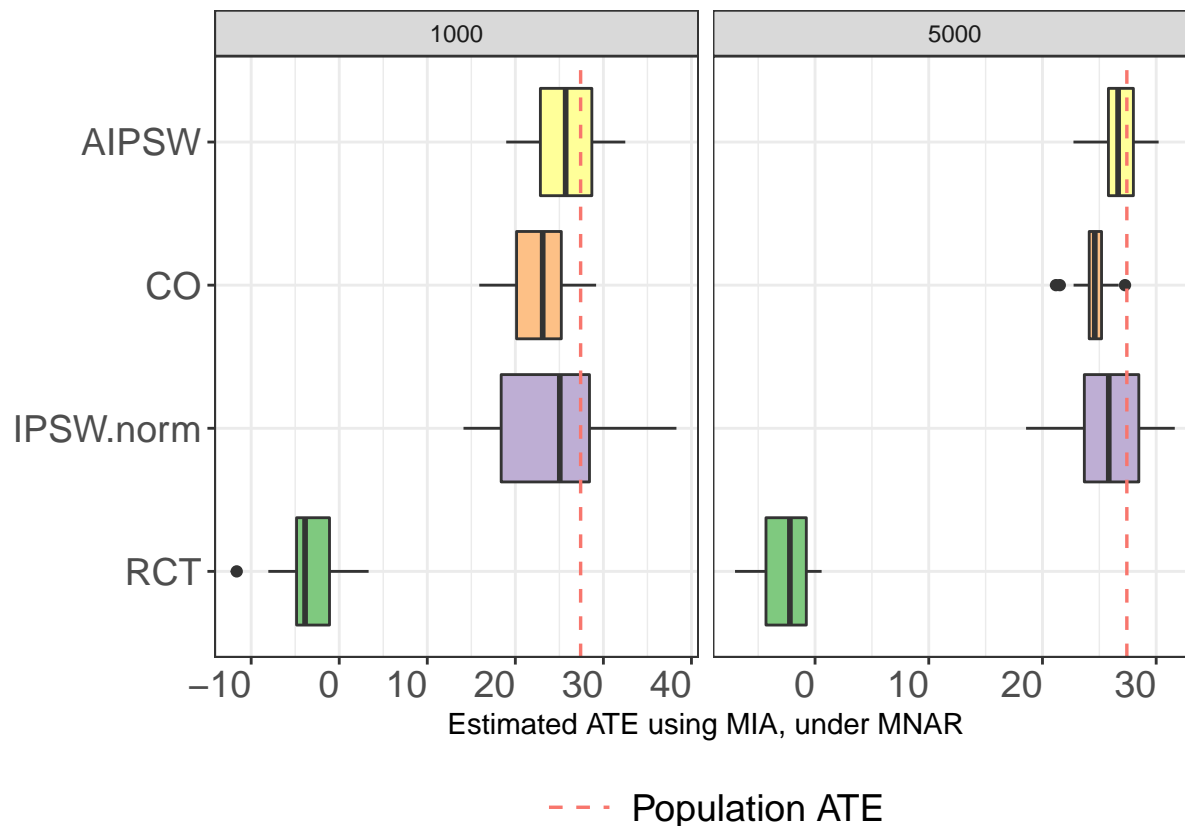
```

```

## [1] "results_rep20_noCIS_noCIO_linklinear_snr5_MNAR_selfmask_propNA0.2_corTRUE_n1000_2000_3000_4000_5000"

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -30.7   1000 grf    linear
## 2 RCT        -30.0   5000 grf    linear
## 3 IPSW.norm  -2.75   1000 grf    linear
## 4 IPSW.norm  -1.75   5000 grf    linear
## 5 CO         -4.79   1000 grf    linear
## 6 CO         -2.88   5000 grf    linear
## 7 AIPSW      -1.79   1000 grf    linear
## 8 AIPSW      -0.684  5000 grf    linear

```



4.3.5 Use within-study multiple imputation

```

methods <- c("glm")
nb_mi <- c(5, 10)

if (!results_exist) {
  results_mi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- NULL
      try(tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                             p = 4, Sigma = Sigma, snr = snr,
                                             na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                             method=methods, nb_strat=1,
                                             do_mi=T, nb_mi=nb_mi,
                                             verbose=T, verbose_intern = F))

      if (!is.null(tmp)){
        tmp$n <- n
        tmp$link <- link
        results_mi <- rbind(results_mi, tmp)
      }
    }
  }
}

```

```

## # A tibble: 10 x 6
##   variable    bias nb_mi    n method link

```

```
##      <fct>          <dbl> <dbl> <dbl> <chr>  <chr>
##  1 RCT            -29.2      10  1000 glm    linear
##  2 RCT            -30.8      10  5000 glm    linear
##  3 IPSW.norm      -3.35      10  1000 glm    linear
##  4 IPSW.norm      -0.934     10  5000 glm    linear
##  5 CO             -3.96      10  1000 glm    linear
##  6 CO             -4.44      10  5000 glm    linear
##  7 AIPSW          -4.59      10  1000 glm    linear
##  8 AIPSW          -4.86      10  5000 glm    linear
##  9 CW             -5.38      10  1000 glm    linear
## 10 CW             -2.94      10  5000 glm    linear
```

4.3.6 Use multilevel multiple imputation with micemd

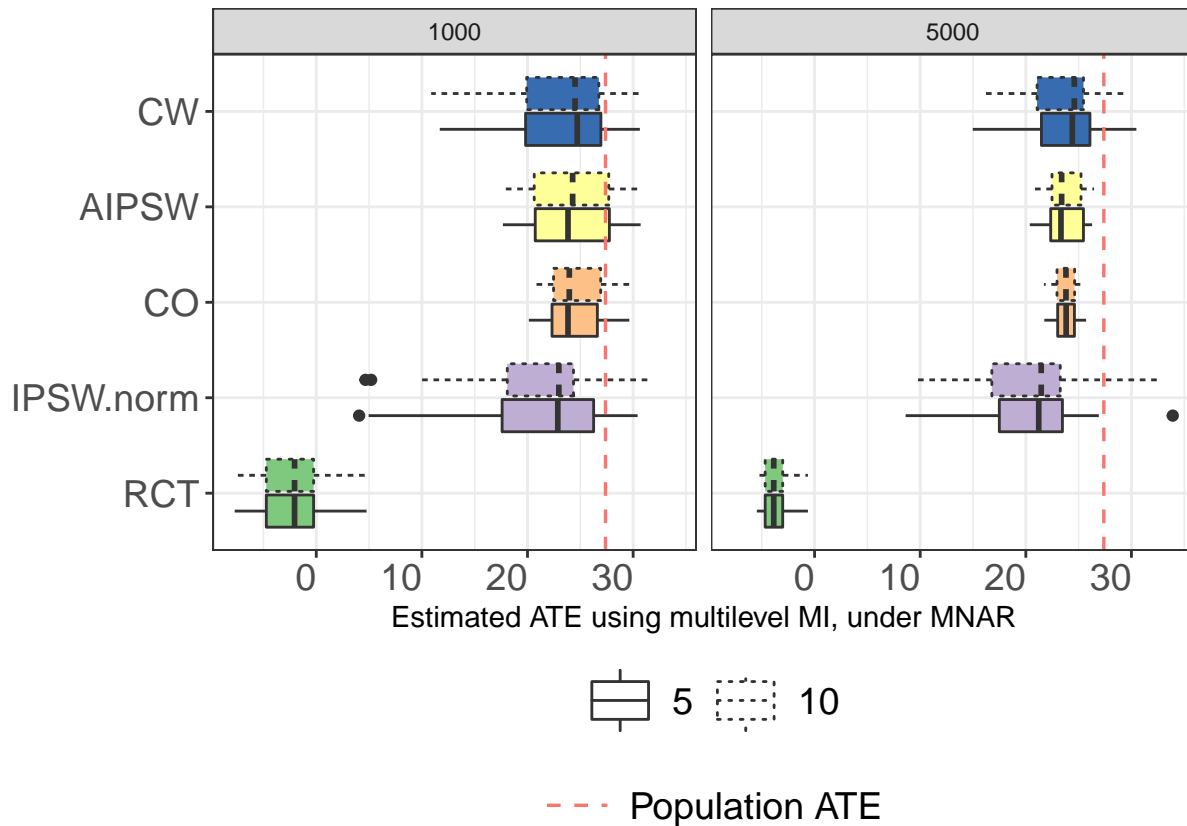
```
methods <- c("glm")
nb_mi <- c(5, 10)

if (!results_exist) {
  results_mi_alt <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      tmp$strategy <- "multi-level-woY"
      results_mi_alt <- rbind(results_mi_alt, tmp)
    }
  }
}
```

```
## [1] "results_rep20_noCIS_noCIO_linklinear_snr5_MNAR_selfmask_propNA0.2_corTRUE_n1000_2000_3000_4000_5000"

## # A tibble: 10 x 7
##   variable    bias nb_mi strategy          n method link
##   <fct>      <dbl> <dbl> <chr>          <dbl> <chr> <chr>
##  1 RCT      -29.6     10 multi-level-woY  1000 glm    linear
##  2 RCT      -31.1     10 multi-level-woY  5000 glm    linear
##  3 IPSW.norm -6.86     10 multi-level-woY  1000 glm    linear
##  4 IPSW.norm -6.72     10 multi-level-woY  5000 glm    linear
##  5 CO        -2.80     10 multi-level-woY  1000 glm    linear
##  6 CO        -3.68     10 multi-level-woY  5000 glm    linear
##  7 AIPSW     -3.13     10 multi-level-woY  1000 glm    linear
##  8 AIPSW     -3.75     10 multi-level-woY  5000 glm    linear
##  9 CW        -4.25     10 multi-level-woY  1000 glm    linear
## 10 CW        -3.76     10 multi-level-woY  5000 glm    linear
```



5 Conditional independence of selection (CIS)

5.1 MCAR

```
prop.miss <- 0.2
mechanism <- "MCAR"
```

5.1.1 On full data

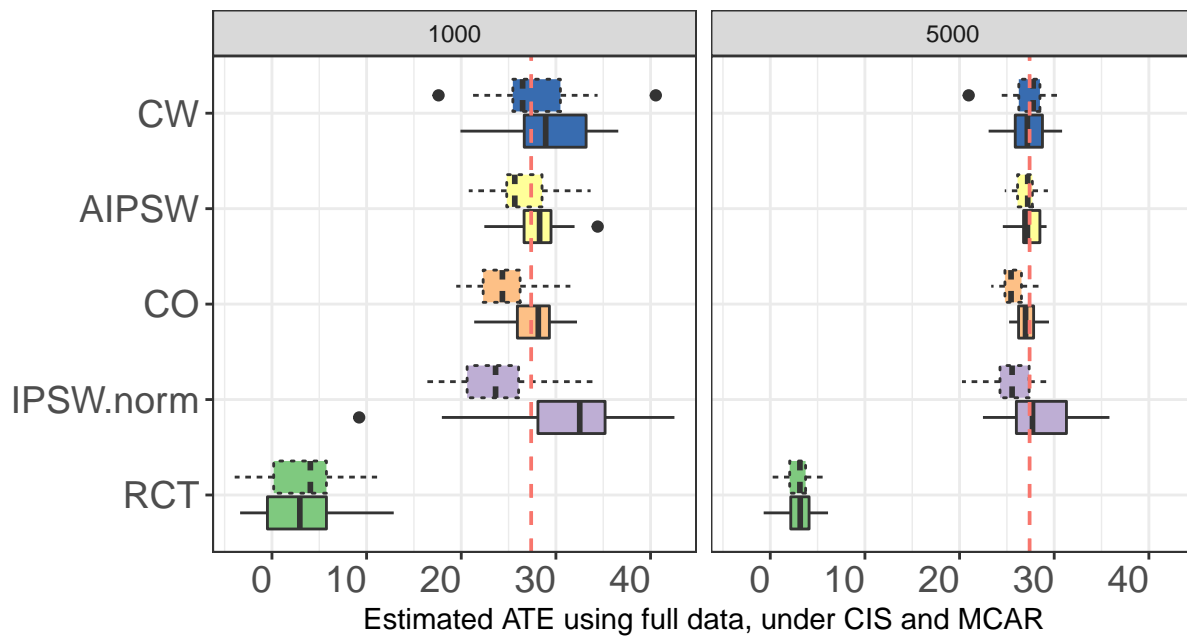
```
methods <- c("glm")

if (!results_exist) {
  results_full <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomp=idx_incomp),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomp=idx_incomp),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE, full_data=T,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_full <- rbind(results_full, tmp)
    }
  }
}
```

```
}
}
```

```
## # A tibble: 20 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -24.3   1000 glm   linear
## 2 RCT        -24.4   5000 glm   linear
## 3 RCT        -24.4   1000 grf    linear
## 4 RCT        -24.5   5000 grf    linear
## 5 IPSW.norm    3.22   1000 glm   linear
## 6 IPSW.norm    1.17   5000 glm   linear
## 7 IPSW.norm   -3.48   1000 grf    linear
## 8 IPSW.norm   -1.97   5000 grf    linear
## 9 CO           0.206  1000 glm   linear
## 10 CO        -0.144  5000 glm   linear
## 11 CO        -2.67   1000 grf    linear
## 12 CO        -1.73   5000 grf    linear
## 13 AIPSW        0.649  1000 glm   linear
## 14 AIPSW       -0.127  5000 glm   linear
## 15 AIPSW       -0.600  1000 grf    linear
## 16 AIPSW       -0.362  5000 grf    linear
## 17 CW          1.74   1000 glm   linear
## 18 CW        -0.0821  5000 glm   linear
## 19 CW          0.0581  1000 grf    linear
## 20 CW        -0.0207  5000 grf    linear
```



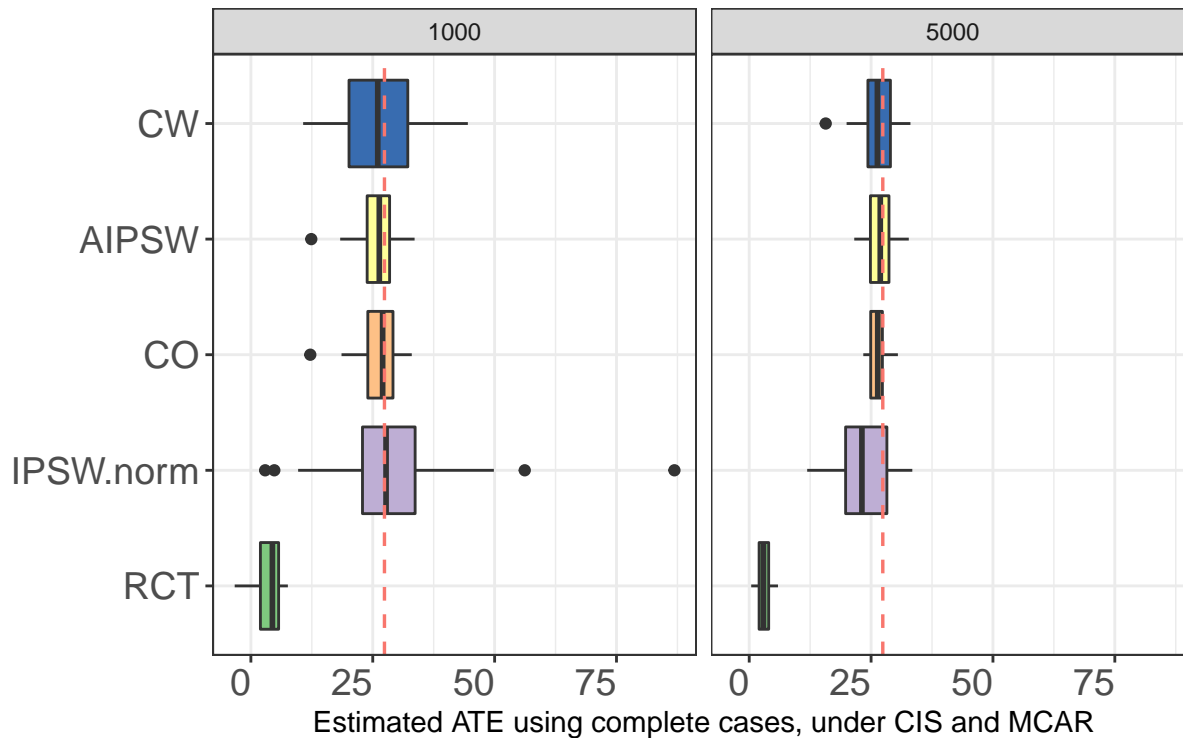
5.1.2 Use only complete cases (for logistic+linear regressions)

```
methods <- c("glm")

if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_cc <- rbind(results_cc, tmp)
    }
  }
}
```

```
## # A tibble: 10 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -24.2   1000 glm  linear
## 2 RCT        -24.3   5000 glm  linear
## 3 IPSW.norm    2.99   1000 glm  linear
## 4 IPSW.norm   -4.18   5000 glm  linear
## 5 CO         -1.22   1000 glm  linear
## 6 CO         -1.15   5000 glm  linear
## 7 AIPSW       -1.60   1000 glm  linear
## 8 AIPSW       -0.610  5000 glm  linear
## 9 CW         -1.26   1000 glm  linear
## 10 CW        -1.24   5000 glm  linear
```

--- Population ATE

5.1.3 Use EM to handle incomplete cases

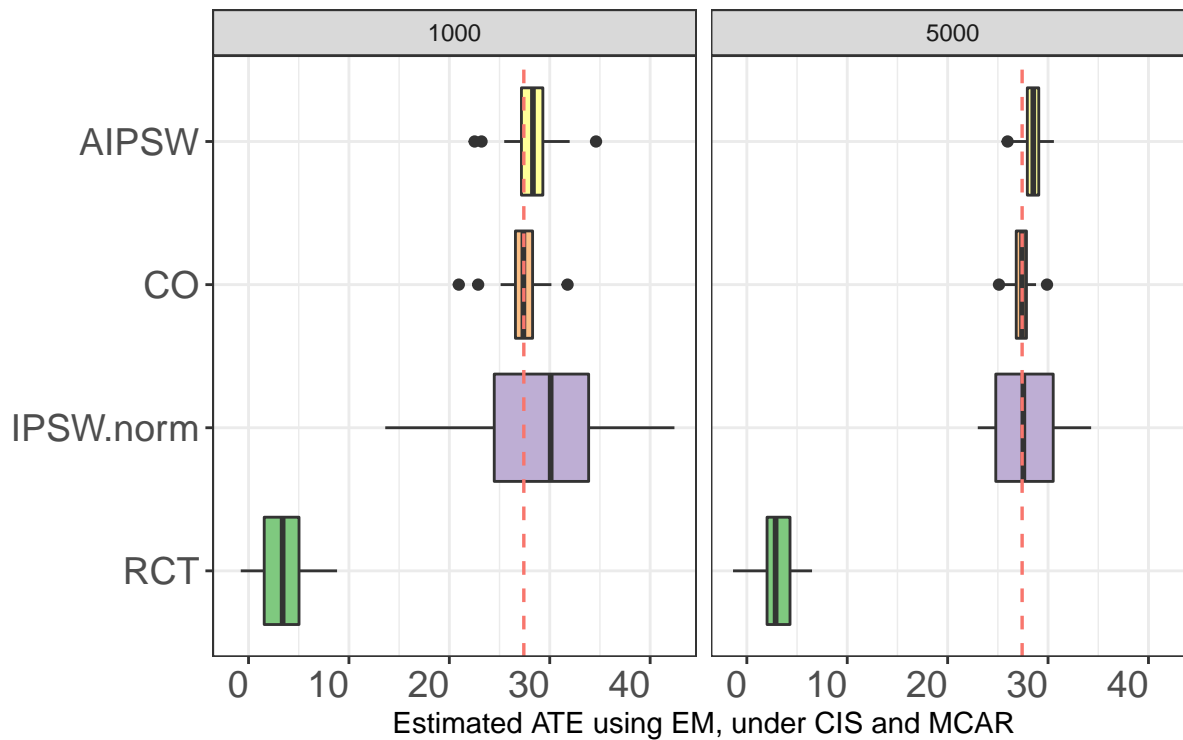
```
methods <- c("glm")

if (!results_exist) {
  results_em <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         method="glm", nb_strat=1,
                                         complete_cases=F,
                                         verbose=T,verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_em <- rbind(results_em, tmp)
    }
  }
}
```

```
## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT      -23.9   1000 glm  linear
## 2 RCT      -24.4   5000 glm  linear
```

```
## 3 IPSW.norm    1.74   1000 glm    linear
## 4 IPSW.norm    0.395  5000 glm    linear
## 5 CO           -0.241  1000 glm    linear
## 6 CO           -0.103  5000 glm    linear
## 7 AIPSW        0.734   1000 glm    linear
## 8 AIPSW        0.961   5000 glm    linear
```



--- Population ATE

5.1.4 Use MIA to handle incomplete cases

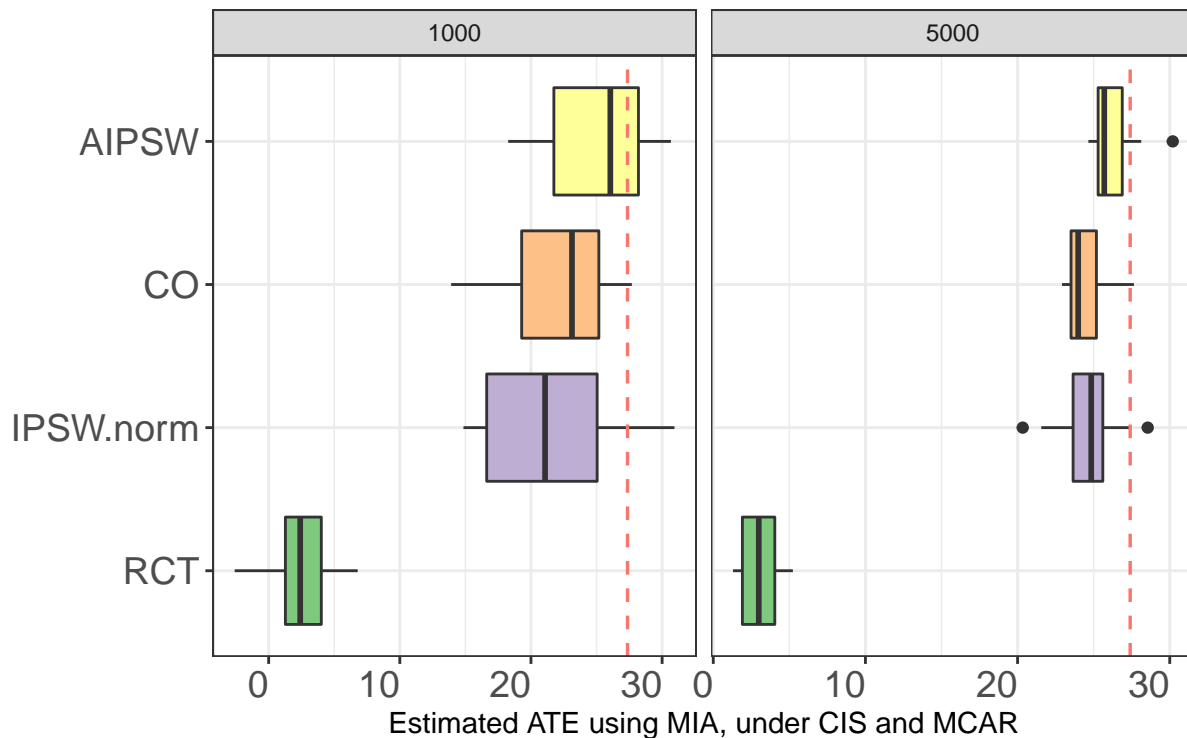
```
methods <- c("grf")

if (!results_exist) {
  results_grf <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom=idx_incom),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom=idx_incom),
                                         method=methods, nb_strat=1,
                                         complete_cases=FALSE,
                                         verbose=T)

      tmp$n <- n
      tmp$link <- link
      results_grf <- rbind(results_grf, tmp)
    }
  }
}
```

```
}
```

```
## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr>  <chr>
## 1 RCT        -25.0  1000 grf    linear
## 2 RCT        -24.3  5000 grf    linear
## 3 IPSW.norm   -6.33  1000 grf    linear
## 4 IPSW.norm   -2.83  5000 grf    linear
## 5 CO          -4.96  1000 grf    linear
## 6 CO          -3.01  5000 grf    linear
## 7 AIPSW       -2.21  1000 grf    linear
## 8 AIPSW       -1.15  5000 grf    linear
```



--- Population ATE

5.1.5 Use within-study multiple imputation

```
methods <- c("glm")

if (!results_exist) {
  results_mi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_rct=list(mechanism=mechanism, prop_miss=prop.miss, idx_inc=idx_inc),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_inc=idx_inc),
                                         method=methods, nb_strat=1,
```

```

do_mi=T, nb_mi=nb_mi,
verbose=T, verbose_intern = F)

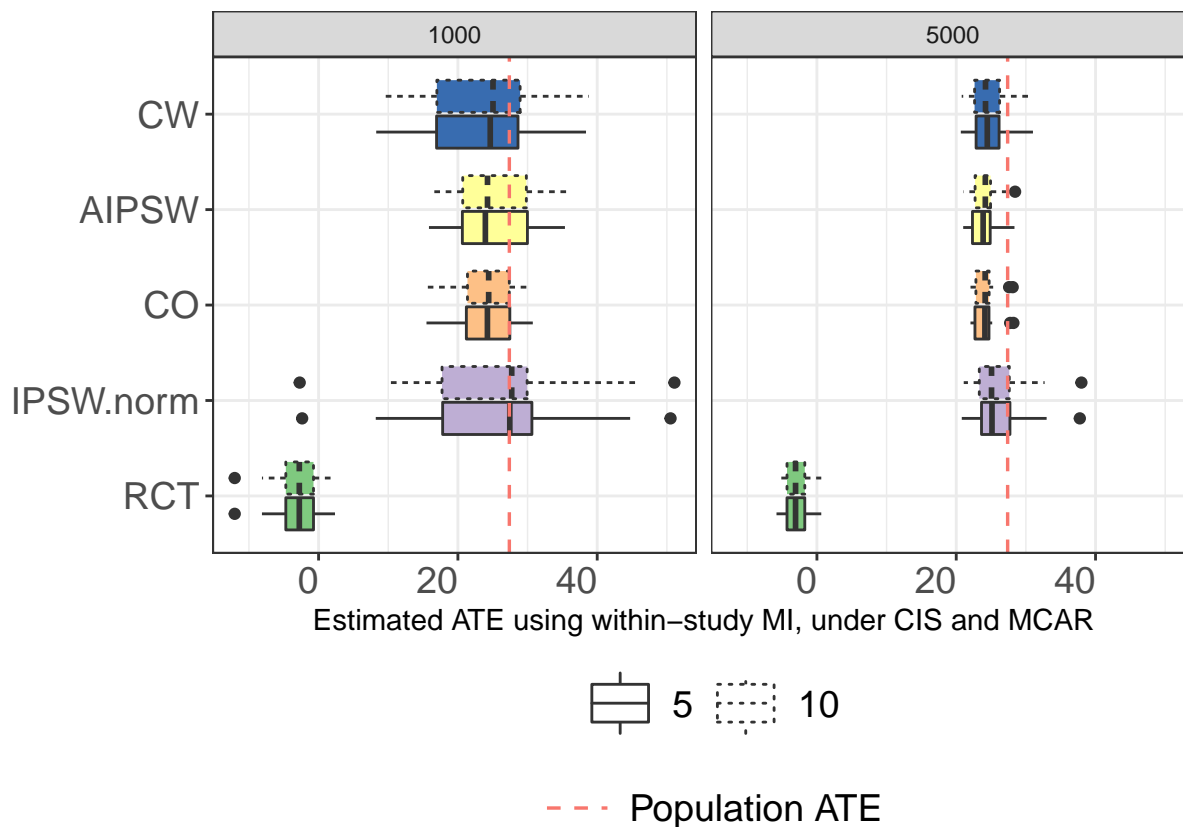
tmp$n <- n
tmp$link <- link
results_mi <- rbind(results_mi, tmp)
}
}
}

```

```

## # A tibble: 10 x 6
##   variable    bias      n method link  nb_mi
##   <fct>      <dbl> <dbl> <chr> <chr> <dbl>
## 1 RCT        -30.6  1000 glm   linear  10
## 2 RCT        -30.3  5000 glm   linear  10
## 3 IPSW.norm  -2.31  1000 glm   linear  10
## 4 IPSW.norm  -1.37  5000 glm   linear  10
## 5 CO         -3.35  1000 glm   linear  10
## 6 CO         -3.28  5000 glm   linear  10
## 7 AIPSW      -2.81  1000 glm   linear  10
## 8 AIPSW      -3.33  5000 glm   linear  10
## 9 CW         -3.39  1000 glm   linear  10
## 10 CW        -2.72  5000 glm   linear  10

```



5.1.6 Use multilevel multiple imputation with micemd

```

methods <- c("glm")
nb_mi <- c(5, 10)

```

```

if (!results_exist) {
  results_mi_alt <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism, prop_miss=prop.miss, idx_inc=idx_inc),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_inc=idx_inc),
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

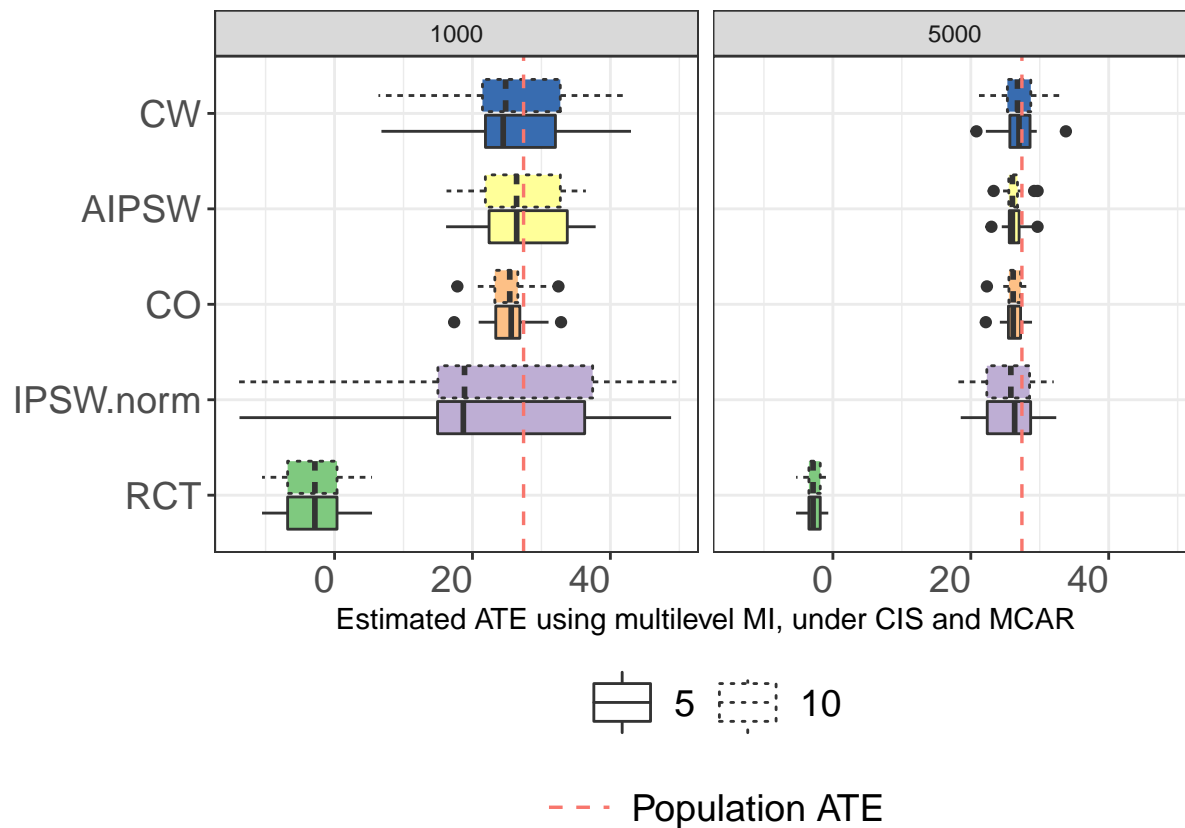
      tmp$n <- n
      tmp$link <- link
      tmp$strategy <- "multi-level-woY"
      results_mi_alt <- rbind(results_mi_alt, tmp)
    }
  }
}

```

```
## [1] "results_rep20_CIS_noCIO_linklinear_snr5_MCAR_propNA0.2_corTRUE_n1000_2000_3000_4000_5000_multil"
```

```
## # A tibble: 10 x 7
```

##	variable	bias	nb_mi	strategy	n	method	link
##	<fct>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<chr>
## 1	RCT	-30.6	10	multi-level-woY	1000	glm	linear
## 2	RCT	-30.2	10	multi-level-woY	5000	glm	linear
## 3	IPSW.norm	-3.20	10	multi-level-woY	1000	glm	linear
## 4	IPSW.norm	-1.87	10	multi-level-woY	5000	glm	linear
## 5	CO	-1.98	10	multi-level-woY	1000	glm	linear
## 6	CO	-1.22	10	multi-level-woY	5000	glm	linear
## 7	AIPSW	-0.854	10	multi-level-woY	1000	glm	linear
## 8	AIPSW	-1.03	10	multi-level-woY	5000	glm	linear
## 9	CW	-0.809	10	multi-level-woY	1000	glm	linear
## 10	CW	-0.683	10	multi-level-woY	5000	glm	linear



5.2 MAR

```
prop.miss <- 0.2
mechanism <- "MAR"
```

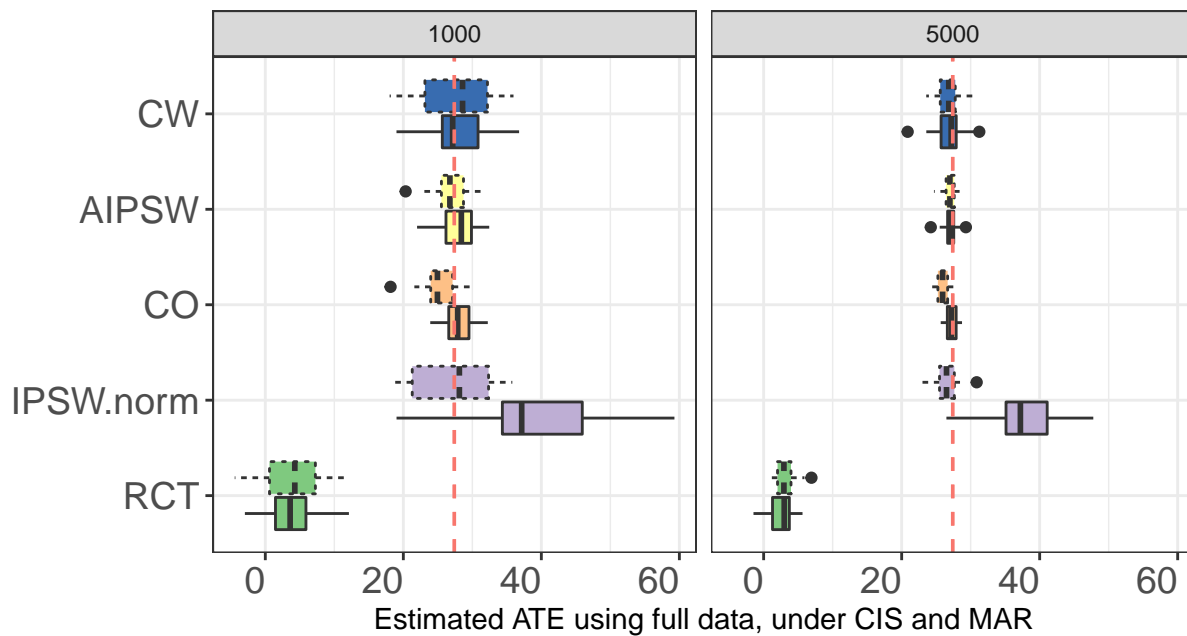
5.2.1 On full data

```
methods <- c("glm", "grf")

if (!results_exist) {
  results_full <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomp=idx_incomp),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomp=idx_incomp),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE, full_data=T,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_full <- rbind(results_full, tmp)
    }
  }
}
```

```
## # A tibble: 20 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -23.6   1000 glm   linear
## 2 RCT        -24.7   5000 glm   linear
## 3 RCT        -23.3   1000 grf   linear
## 4 RCT        -24.2   5000 grf   linear
## 5 IPSW.norm    11.2   1000 glm   linear
## 6 IPSW.norm    10.4   5000 glm   linear
## 7 IPSW.norm     0.144 1000 grf   linear
## 8 IPSW.norm   -0.941 5000 grf   linear
## 9 CO           0.379   1000 glm   linear
## 10 CO        -0.172   5000 glm   linear
## 11 CO        -1.94   1000 grf   linear
## 12 CO        -1.45   5000 grf   linear
## 13 AIPSW        0.822   1000 glm   linear
## 14 AIPSW       -0.262   5000 glm   linear
## 15 AIPSW       -0.372   1000 grf   linear
## 16 AIPSW       -0.406   5000 grf   linear
## 17 CW          0.0589 1000 glm   linear
## 18 CW        -0.500   5000 glm   linear
## 19 CW          0.769   1000 grf   linear
## 20 CW        -0.624   5000 grf   linear
```



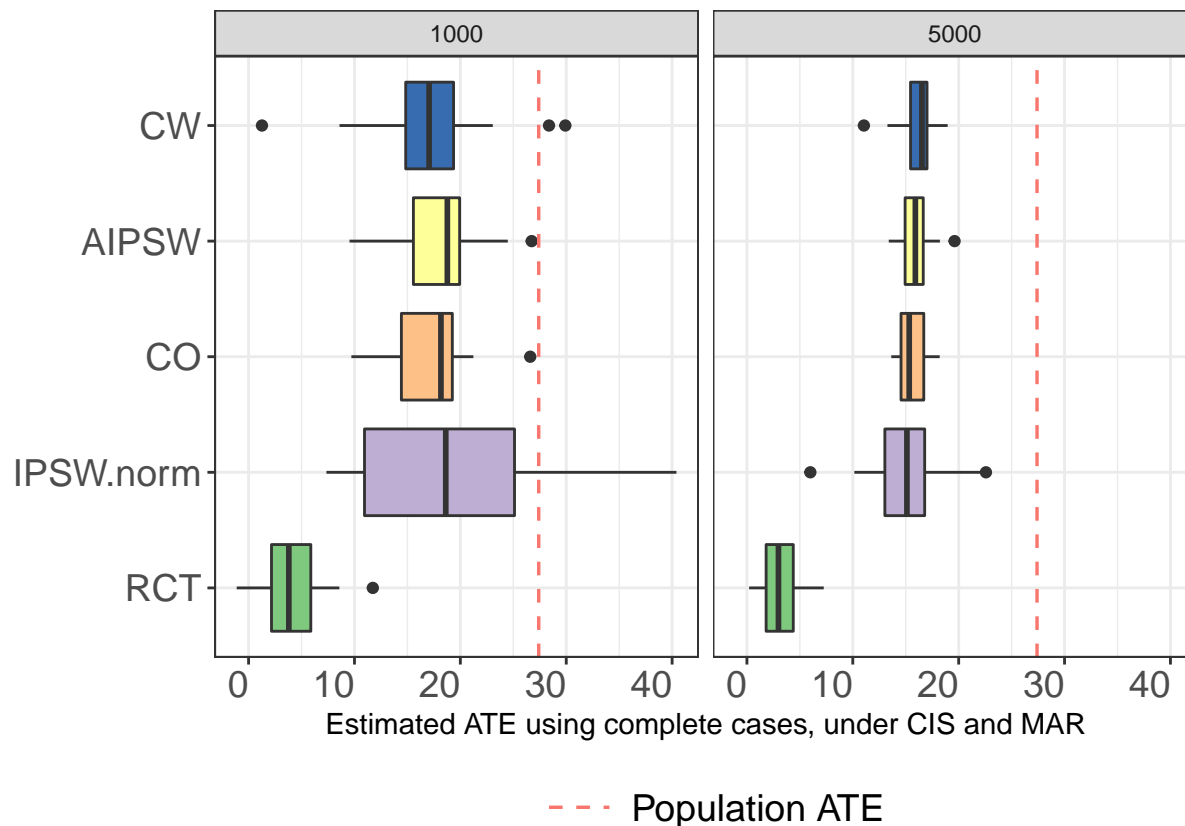
5.2.2 Use only complete cases (for logistic+linear regressions)

```
methods <- c("glm")

if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_cc <- rbind(results_cc, tmp)
    }
  }
}
```

```
## # A tibble: 10 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -23.4  1000 glm  linear
## 2 RCT        -24.2  5000 glm  linear
## 3 IPSW.norm  -8.46  1000 glm  linear
## 4 IPSW.norm -12.1   5000 glm  linear
## 5 CO         -10.1  1000 glm  linear
## 6 CO         -11.8  5000 glm  linear
## 7 AIPSW       -9.52  1000 glm  linear
## 8 AIPSW      -11.5  5000 glm  linear
## 9 CW         -10.3  1000 glm  linear
## 10 CW        -11.4  5000 glm  linear
```

5.2.3 Use EM to handle incomplete cases

```

methods <- c("glm")

if (!results_exist) {
  results_em <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         method="glm", nb_strat=1,
                                         complete_cases=F,
                                         verbose=T,verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_em <- rbind(results_em, tmp)
    }
  }
}

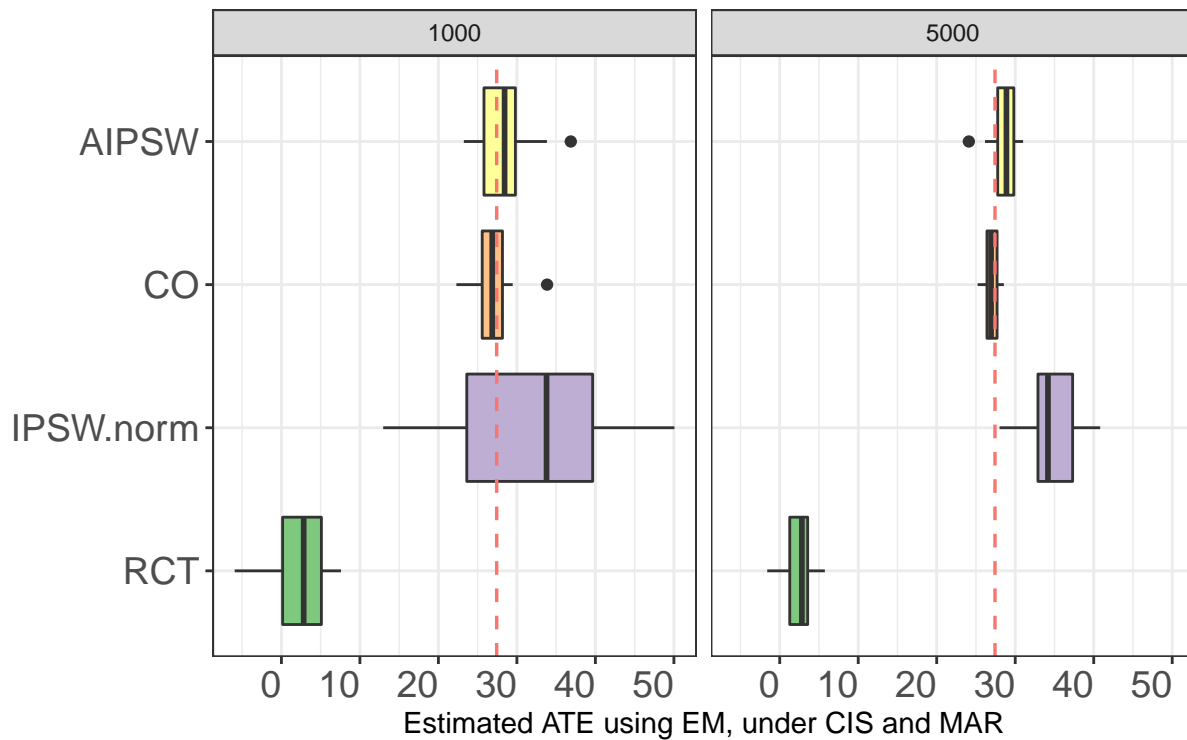
```

```

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT       -24.9   1000 glm  linear
## 2 RCT       -24.9   5000 glm  linear

```

```
## 3 IPSW.norm    4.06   1000 glm   linear
## 4 IPSW.norm    7.01   5000 glm   linear
## 5 CO           -0.485 1000 glm   linear
## 6 CO           -0.472 5000 glm   linear
## 7 AIPSW        0.745  1000 glm   linear
## 8 AIPSW        1.21   5000 glm   linear
```



--- Population ATE

5.2.4 Use MIA to handle incomplete cases

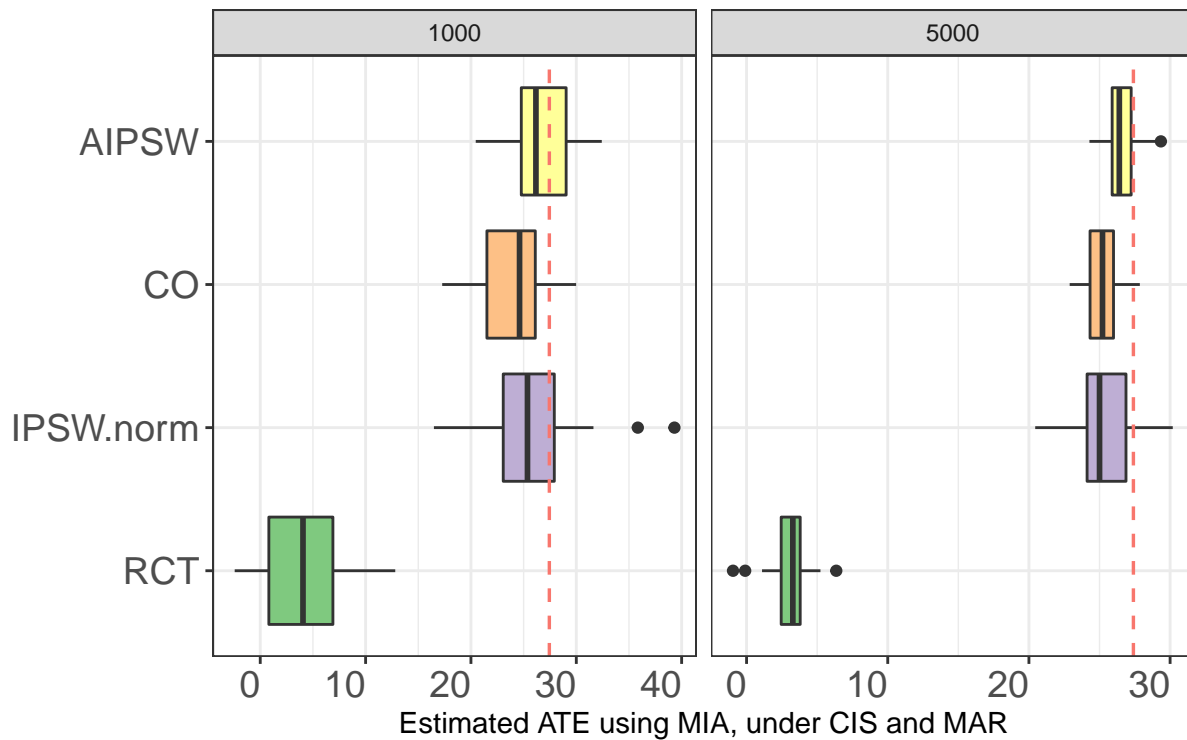
```
methods <- c("grf")

if (!results_exist) {
  results_grf <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         method=methods, nb_strat=1,
                                         complete_cases=FALSE,
                                         verbose=T)

      tmp$n <- n
      tmp$link <- link
      results_grf <- rbind(results_grf, tmp)
    }
  }
}
```

```
}
```

```
## # A tibble: 8 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -23.5   1000 grf    linear
## 2 RCT        -24.4   5000 grf    linear
## 3 IPSW.norm   -1.72   1000 grf    linear
## 4 IPSW.norm   -1.96   5000 grf    linear
## 5 CO          -3.53   1000 grf    linear
## 6 CO          -2.22   5000 grf    linear
## 7 AIPSW        -1.12   1000 grf    linear
## 8 AIPSW        -0.796  5000 grf    linear
```



--- Population ATE

5.2.5 Use within-study multiple imputation

```
methods <- c("glm")
nb_mi <- c(5, 10)

if (!results_exist) {
  results_mi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_rct=list(mechanism=mechanism, prop_miss=prop.miss, idx_inc=idx_inc),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_inc=idx_inc))
    }
  }
}
```

```

method=methods, nb_strat=1,
do_mi=T, nb_mi=nb_mi,
verbose=T,verbose_intern = F)

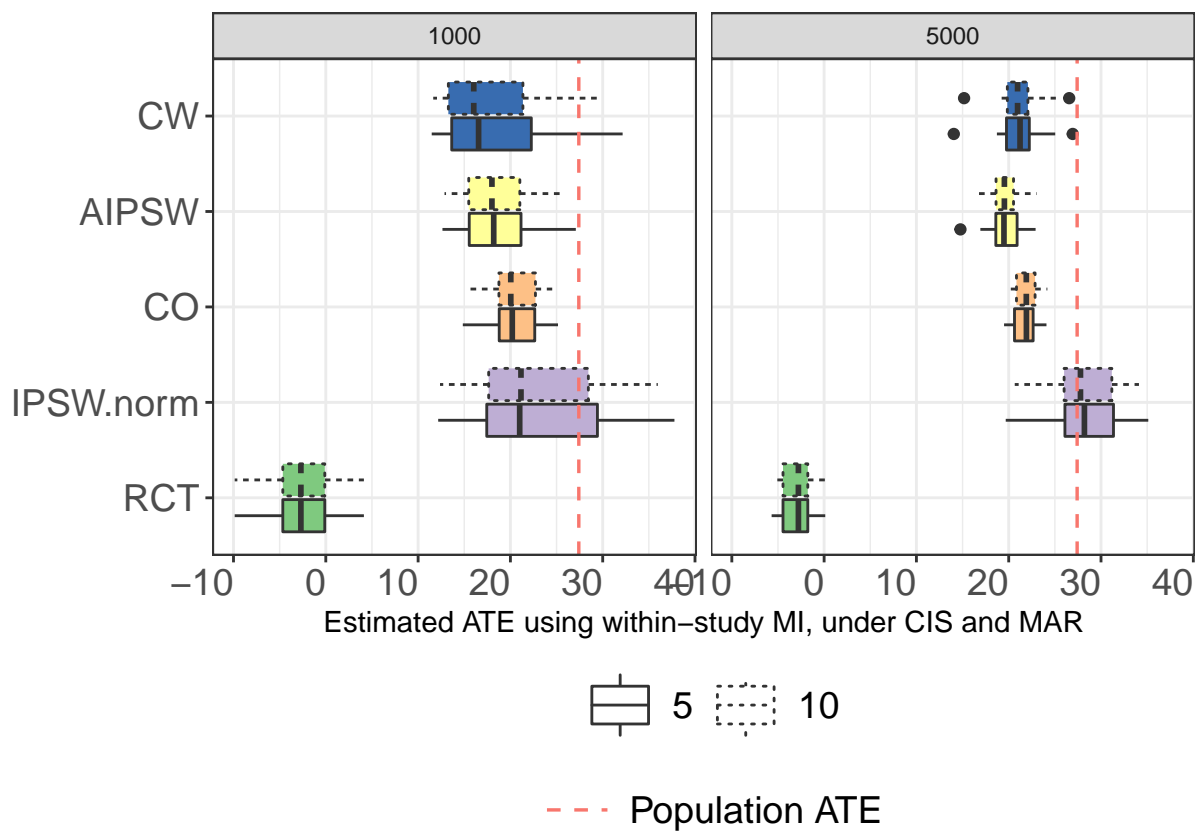
tmp$n <- n
tmp$link <- link
results_mi <- rbind(results_mi, tmp)
}
}
}

```

```

## # A tibble: 10 x 6
##   variable      bias nb_mi      n method link
##   <fct>      <dbl> <dbl> <dbl> <chr> <chr>
## 1 RCT        -29.9    10  1000 glm  linear
## 2 RCT        -30.4    10  5000 glm  linear
## 3 IPSW.norm   -4.39    10  1000 glm  linear
## 4 IPSW.norm    0.463    10  5000 glm  linear
## 5 CO         -7.26    10  1000 glm  linear
## 6 CO         -5.52    10  5000 glm  linear
## 7 AIPSW       -9.02    10  1000 glm  linear
## 8 AIPSW       -7.85    10  5000 glm  linear
## 9 CW         -9.57    10  1000 glm  linear
## 10 CW        -6.35    10  5000 glm  linear

```



5.2.6 Use multilevel multiple imputation with micemd

```

methods <- c("glm")
nb_mi <- c(5, 10)

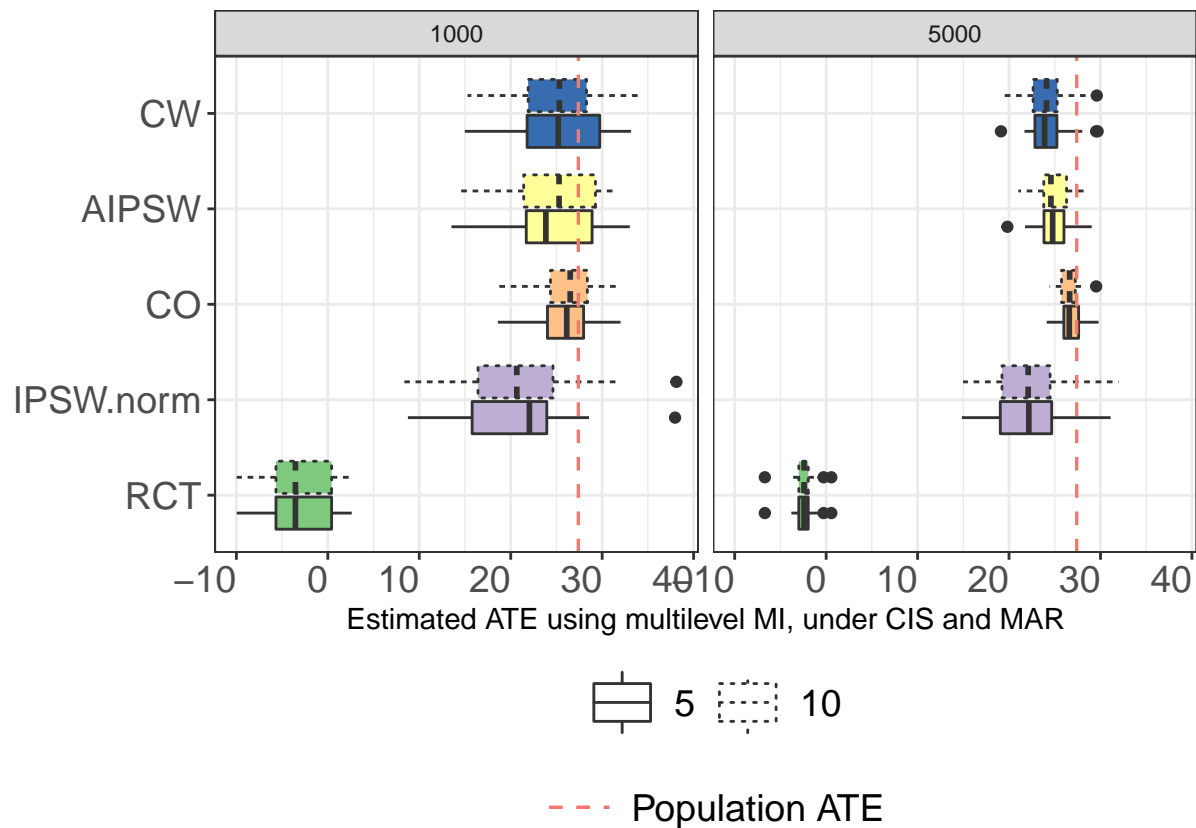
if (!results_exist) {
  results_mi_alt <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism, prop_miss=prop.miss, idx_inco
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_inco
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      tmp$strategy <- "multi-level-woY"
      results_mi_alt <- rbind(results_mi_alt, tmp)
    }
  }
}

```

```
## [1] "results_rep20_CIS_linklinear_snr5_MAR_propNA0.2_corTRUE_n1000_2000_3000_4000_5000_multilevelmi_r
```

```
## # A tibble: 10 x 7
##   variable      bias nb_mi strategy      n method link
##   <fct>      <dbl> <dbl> <chr>    <dbl> <chr> <chr>
## 1 RCT        -30.2     10 multi-level-woY 1000 glm linear
## 2 RCT        -29.9     10 multi-level-woY 5000 glm linear
## 3 IPSW.norm  -6.17     10 multi-level-woY 1000 glm linear
## 4 IPSW.norm  -5.15     10 multi-level-woY 5000 glm linear
## 5 CO         -1.17     10 multi-level-woY 1000 glm linear
## 6 CO         -0.814     10 multi-level-woY 5000 glm linear
## 7 AIPSW      -2.53     10 multi-level-woY 1000 glm linear
## 8 AIPSW      -2.71     10 multi-level-woY 5000 glm linear
## 9 CW         -2.37     10 multi-level-woY 1000 glm linear
## 10 CW        -3.15     10 multi-level-woY 5000 glm linear
```



5.3 MNAR (self-mask)

```
prop.miss <- 0.2
mechanism <- "MNAR_selfmask"
```

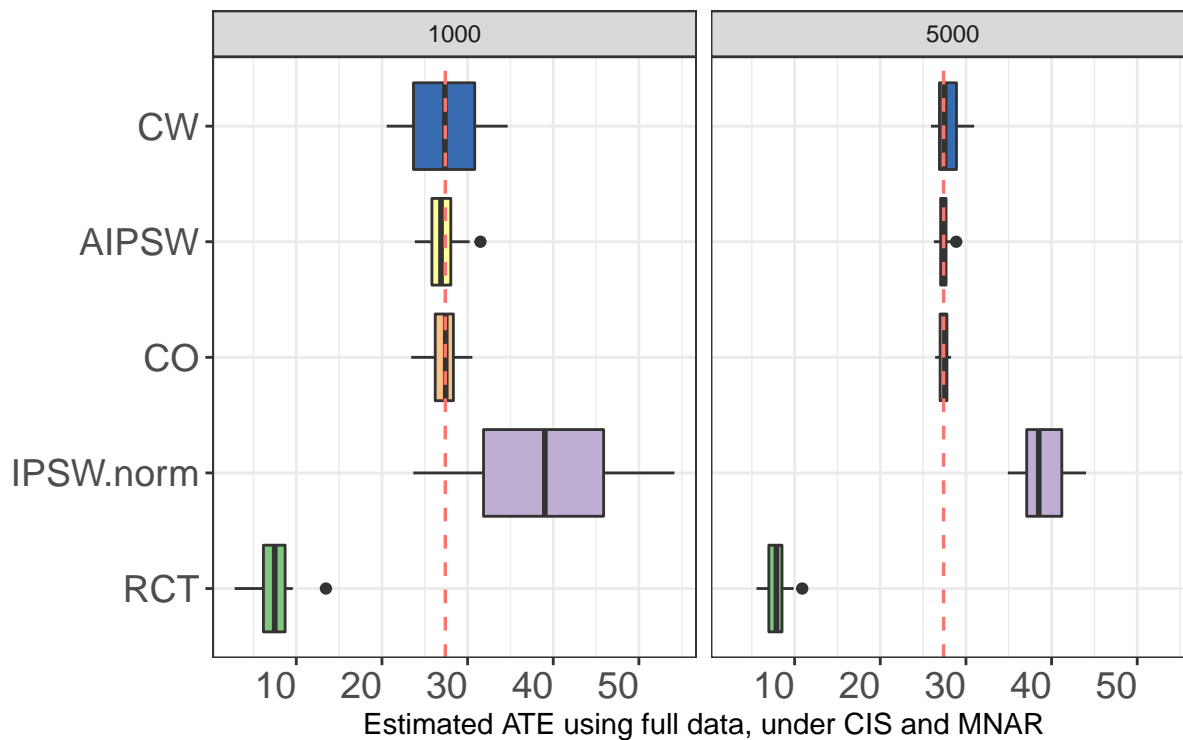
5.3.1 On full data

```
methods <- c("glm", "grf")

if (!results_exist) {
  results_full <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE, full_data=T,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_full <- rbind(results_full, tmp)
    }
  }
}
```

```
## # A tibble: 10 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -20.1   1000 glm   linear
## 2 RCT        -19.5   5000 glm   linear
## 3 IPSW.norm   11.4   1000 glm   linear
## 4 IPSW.norm   11.6   5000 glm   linear
## 5 CO         -0.117  1000 glm   linear
## 6 CO          0.0451 5000 glm   linear
## 7 AIPSW       -0.457  1000 glm   linear
## 8 AIPSW       -0.0341 5000 glm   linear
## 9 CW         -0.0313 1000 glm   linear
## 10 CW         0.577   5000 glm   linear
```



--- Population ATE

5.3.2 Use only complete cases (for logistic+linear regressions)

```
methods <- c("glm")

if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                         method=methods, nb_strat=1,
```

```

complete_cases=TRUE,
verbose=T,verbose_intern = F)

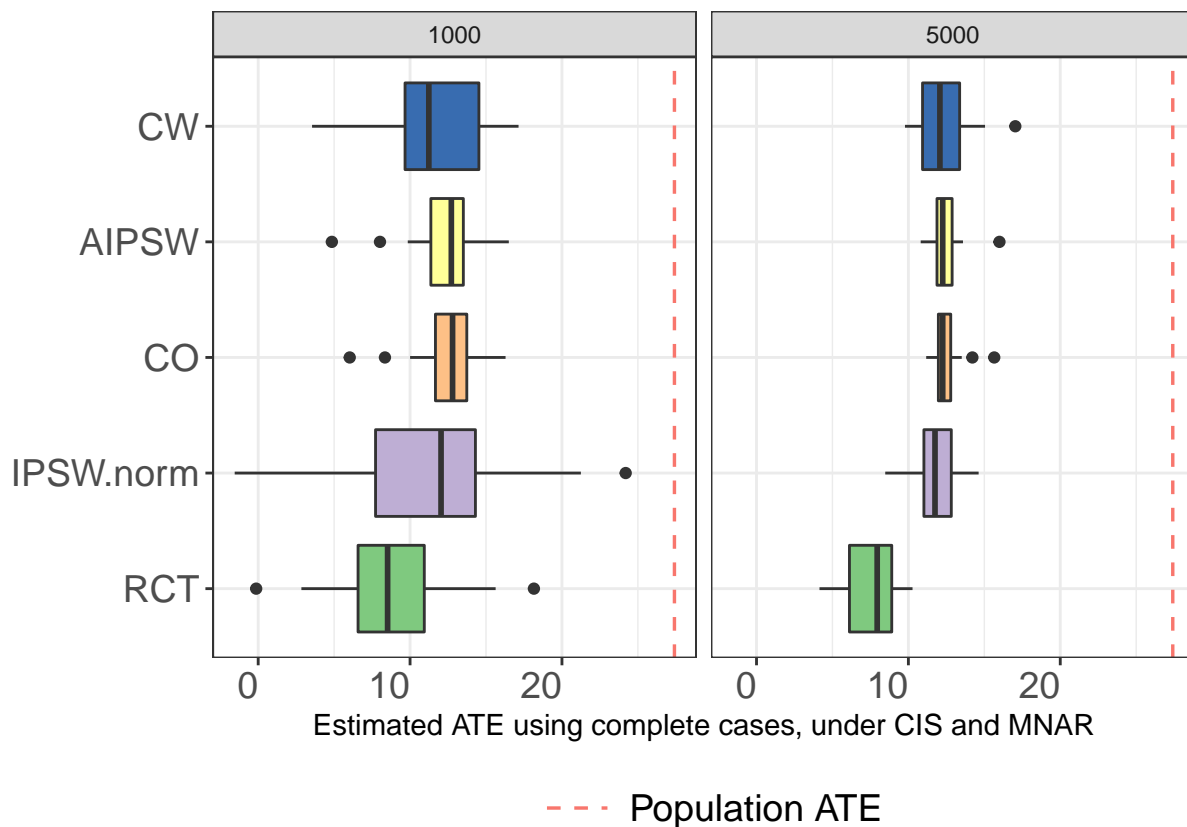
tmp$n <- n
tmp$link <- link
results_cc <- rbind(results_cc, tmp)
}
}
}

```

```

## # A tibble: 10 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -18.7  1000  glm   linear
## 2 RCT        -19.8  5000  glm   linear
## 3 IPSW.norm  -15.7  1000  glm   linear
## 4 IPSW.norm  -15.6  5000  glm   linear
## 5 CO         -14.9  1000  glm   linear
## 6 CO         -14.9  5000  glm   linear
## 7 AIPSW      -15.0  1000  glm   linear
## 8 AIPSW      -14.9  5000  glm   linear
## 9 CW         -15.7  1000  glm   linear
## 10 CW        -15.0  5000  glm   linear

```



5.3.3 Use EM to handle incomplete cases

```

methods <- c("glm")

```



```

if (!results_exist) {
  results_em <- c()
  for (link in links){
    for (n in n_range){
      tmp <- NULL
      try(tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                             p = 4, Sigma = Sigma, snr = snr,
                                             na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                             na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                             method="glm", nb_strat=1,
                                             complete_cases=F,
                                             verbose=T,verbose_intern = F))

      if (!is.null(tmp)){
        tmp$n <- n
        tmp$link <- link
        results_em <- rbind(results_em, tmp)
      }
    }
  }
}

```

```
## [1] "EM did not converge in any case of this scenario."
```

5.3.4 Use MIA to handle incomplete cases

```

methods <- c("grf")

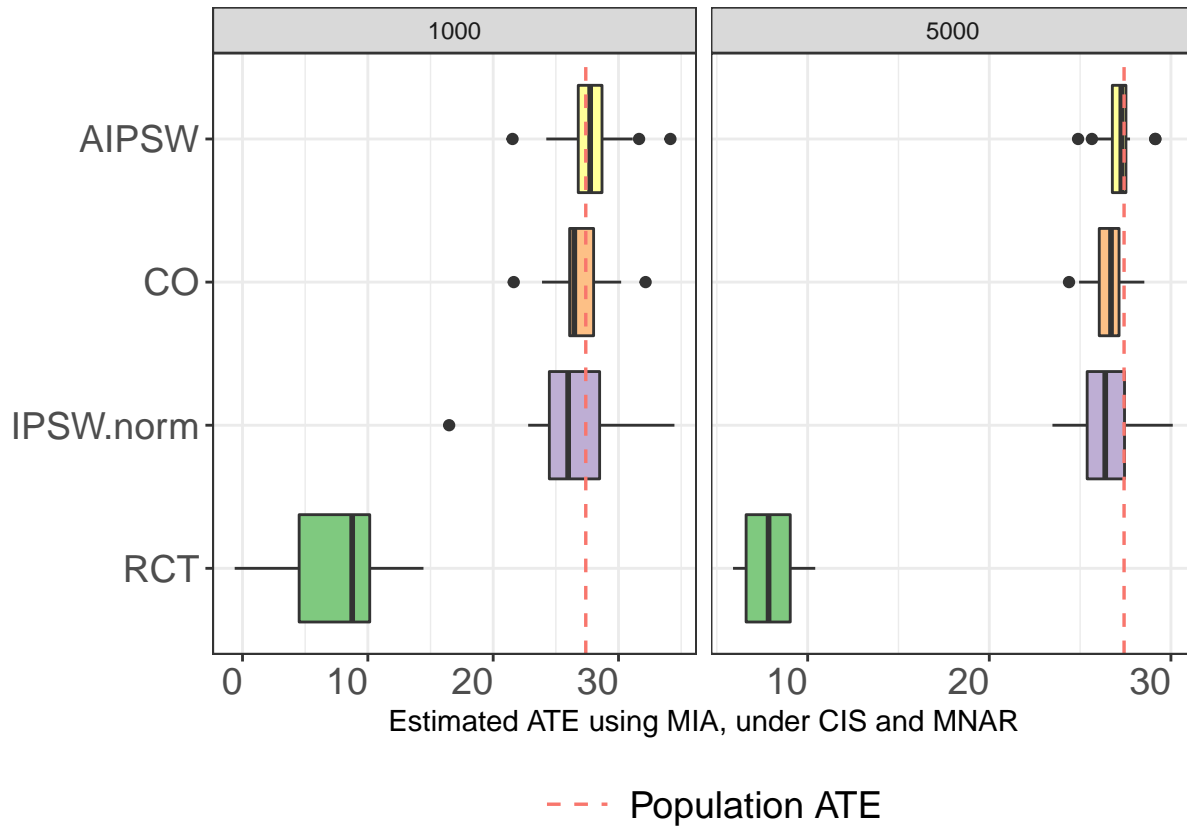
if (!results_exist) {
  results_grf <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                             p = 4, Sigma = Sigma, snr = snr,
                                             na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                             na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incomplete=idx_incomplete),
                                             method=methods, nb_strat=1,
                                             complete_cases=FALSE,
                                             verbose=T)

      tmp$n <- n
      tmp$link <- link
      results_grf <- rbind(results_grf, tmp)
    }
  }
}

```

```
## # A tibble: 8 x 5
##   variable      bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -19.8   1000 grf    linear
## 2 RCT        -19.6   5000 grf    linear
## 3 IPSW.norm  -1.21   1000 grf    linear
## 4 IPSW.norm  -0.899  5000 grf    linear
## 5 CO         -0.375  1000 grf    linear
## 6 CO         -0.873  5000 grf    linear
```

```
## 7 AIPSW      0.467 1000 grf linear
## 8 AIPSW     -0.294 5000 grf linear
```



5.3.5 Use within-study multiple imputation

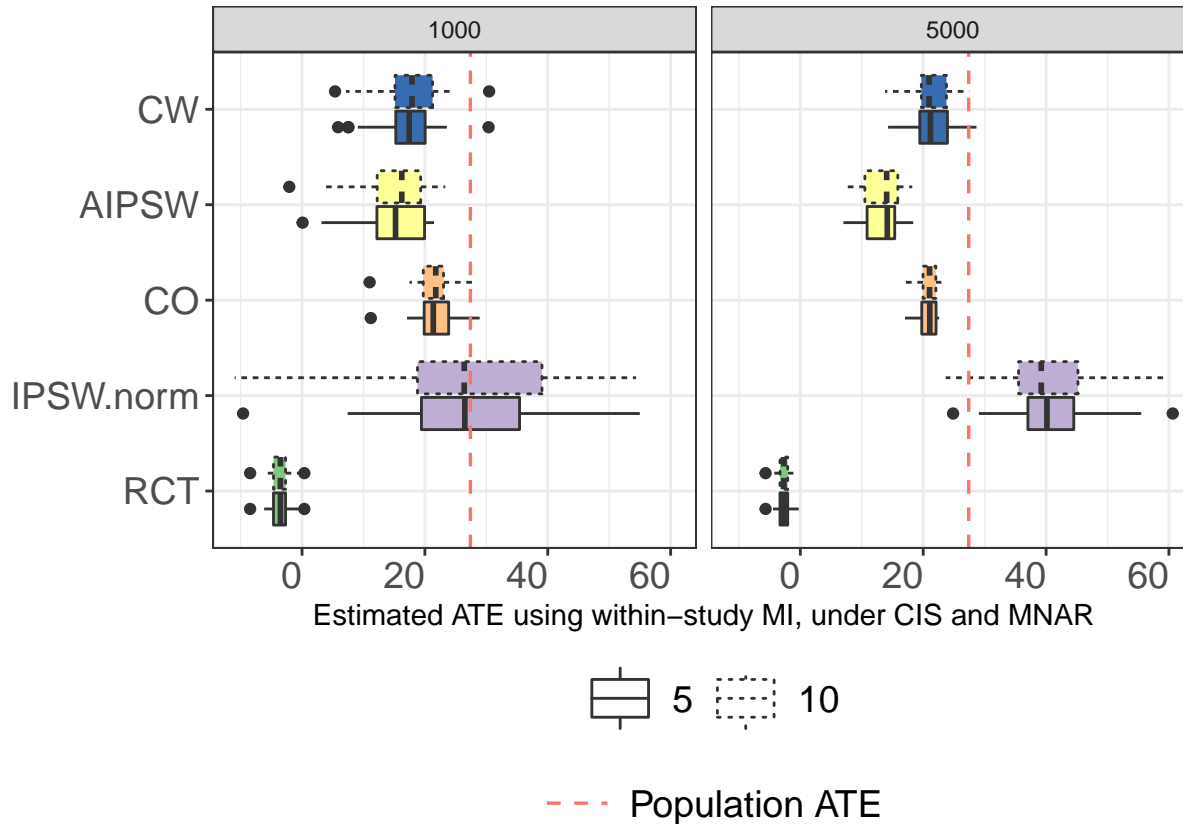
```
methods <- c("glm")
nb_mi <- c(5, 10)

if (!results_exist) {
  results_mi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_mi <- rbind(results_mi, tmp)
    }
  }
}
```

```
## # A tibble: 10 x 6
##   variable    bias nb_mi    n method link
```

##	<fct>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
## 1	RCT	-30.9	10	1000	glm	linear
## 2	RCT	-30.2	10	5000	glm	linear
## 3	IPSW.norm	-0.204	10	1000	glm	linear
## 4	IPSW.norm	13.0	10	5000	glm	linear
## 5	CO	-6.00	10	1000	glm	linear
## 6	CO	-6.64	10	5000	glm	linear
## 7	AIPSW	-13.0	10	1000	glm	linear
## 8	AIPSW	-14.1	10	5000	glm	linear
## 9	CW	-10.0	10	1000	glm	linear
## 10	CW	-6.13	10	5000	glm	linear



5.3.6 Use multilevel multiple imputation with micemd

```

methods <- c("glm")
nb_mi <- c(5, 10)

if (!results_exist) {
  results_mi_alt <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                         na_rwe=list(mechanism=mechanism, prop_miss=prop.miss, idx_incor=idx_incor),
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
  
```

```

                                verbose=T, verbose_intern = F)

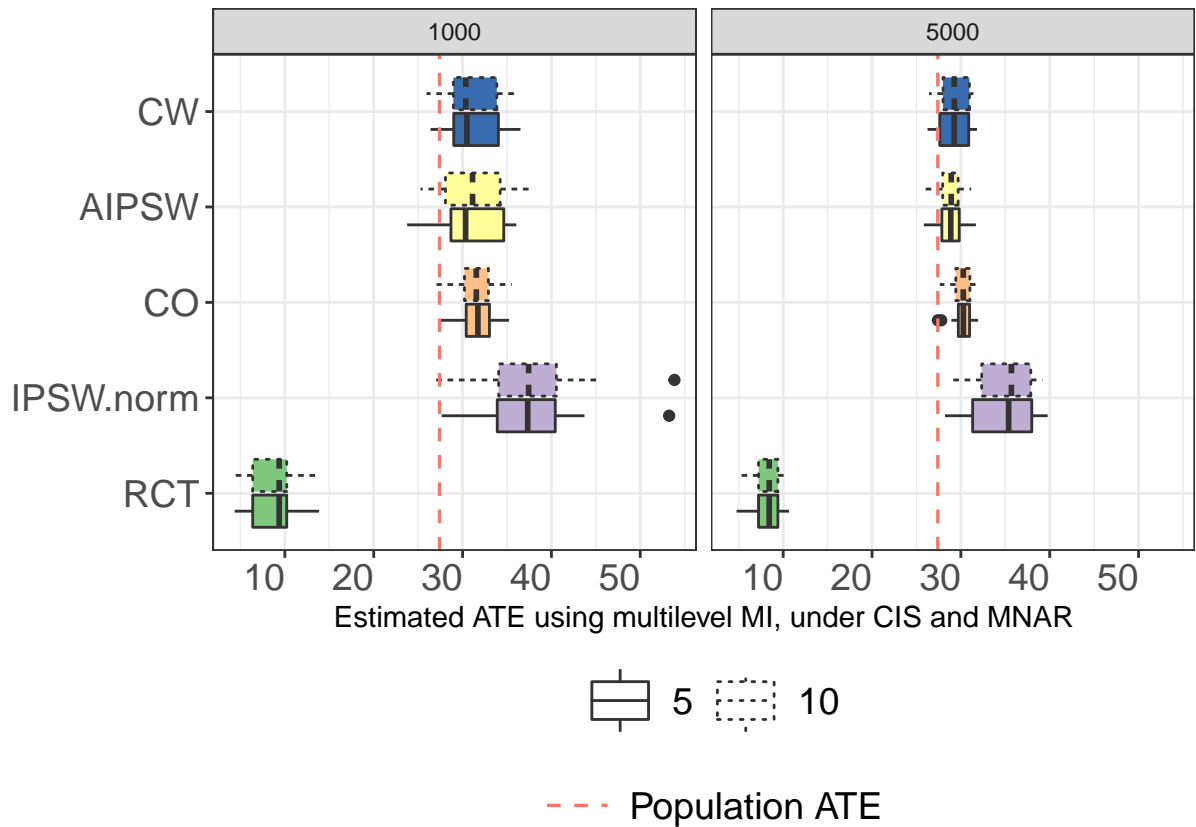
    tmp$n <- n
    tmp$link <- link
    tmp$strategy <- "multi-level-woY"
    results_mi_alt <- rbind(results_mi_alt, tmp)
  }
}
}

```

```
## [1] "results_rep20_CIS_linklinear_snr5_MNAR_selfmask_propNA0.2_corTRUE_n1000_2000_3000_4000_5000_mul"
```

```
## # A tibble: 10 x 7
```

	variable	bias	nb_mi	strategy	n	method	link
	<fct>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<chr>
## 1	RCT	-18.7	10	multi-level-woY	1000	glm	linear
## 2	RCT	-19.2	10	multi-level-woY	5000	glm	linear
## 3	IPSW.norm	10.6	10	multi-level-woY	1000	glm	linear
## 4	IPSW.norm	7.27	10	multi-level-woY	5000	glm	linear
## 5	CO	4.12	10	multi-level-woY	1000	glm	linear
## 6	CO	2.75	10	multi-level-woY	5000	glm	linear
## 7	AIPSW	3.62	10	multi-level-woY	1000	glm	linear
## 8	AIPSW	1.47	10	multi-level-woY	5000	glm	linear
## 9	CW	3.92	10	multi-level-woY	1000	glm	linear
## 10	CW	1.78	10	multi-level-woY	5000	glm	linear



6 Other settings

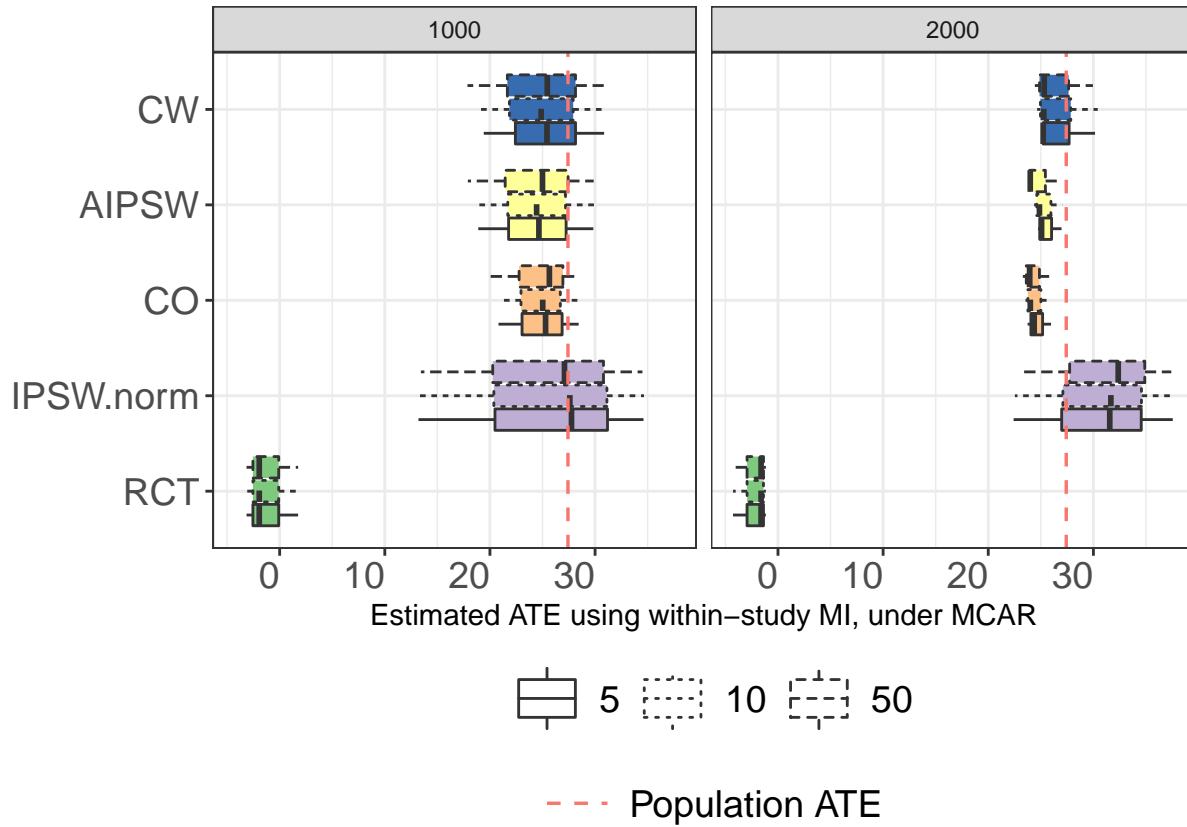
6.1 Impact of number of multiple imputations

```
mechanism <- "MCAR"
methods <- c("glm")
nb_mi <- c(10, 50)

if (!results_exist) {
  results_mi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = 3, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr = snr,
                                         na_df=list(mechanism=mechanism, prop_miss=prop.miss, idx_incom
                                         method=methods, nb_strat=1,
                                         do_mi=T, nb_mi=nb_mi,
                                         verbose=T,verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_mi <- rbind(results_mi, tmp)
    }
  }
}
```

```
## # A tibble: 10 x 6
##   variable      bias nb_mi      n method link
##   <fct>      <dbl> <dbl> <dbl> <chr>  <chr>
## 1 RCT        -28.5    50  1000 glm    linear
## 2 RCT        -29.8    50  2000 glm    linear
## 3 IPSW.norm  -2.41    50  1000 glm    linear
## 4 IPSW.norm   3.54    50  2000 glm    linear
## 5 CO         -2.83    50  1000 glm    linear
## 6 CO         -3.08    50  2000 glm    linear
## 7 AIPSW      -3.16    50  1000 glm    linear
## 8 AIPSW      -2.51    50  2000 glm    linear
## 9 CW         -2.71    50  1000 glm    linear
## 10 CW        -0.849    50  2000 glm    linear
```



6.2 Impact of different proportions in different data

6.2.1 RCT: MCAR 0.1, RWE: MCAR 0.5

```
mechanism <- c("MCAR", "MCAR")
prop.miss <- c(0.1, 0.5)
```

```
methods <- c("glm")

if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T, verbose_intern = F)

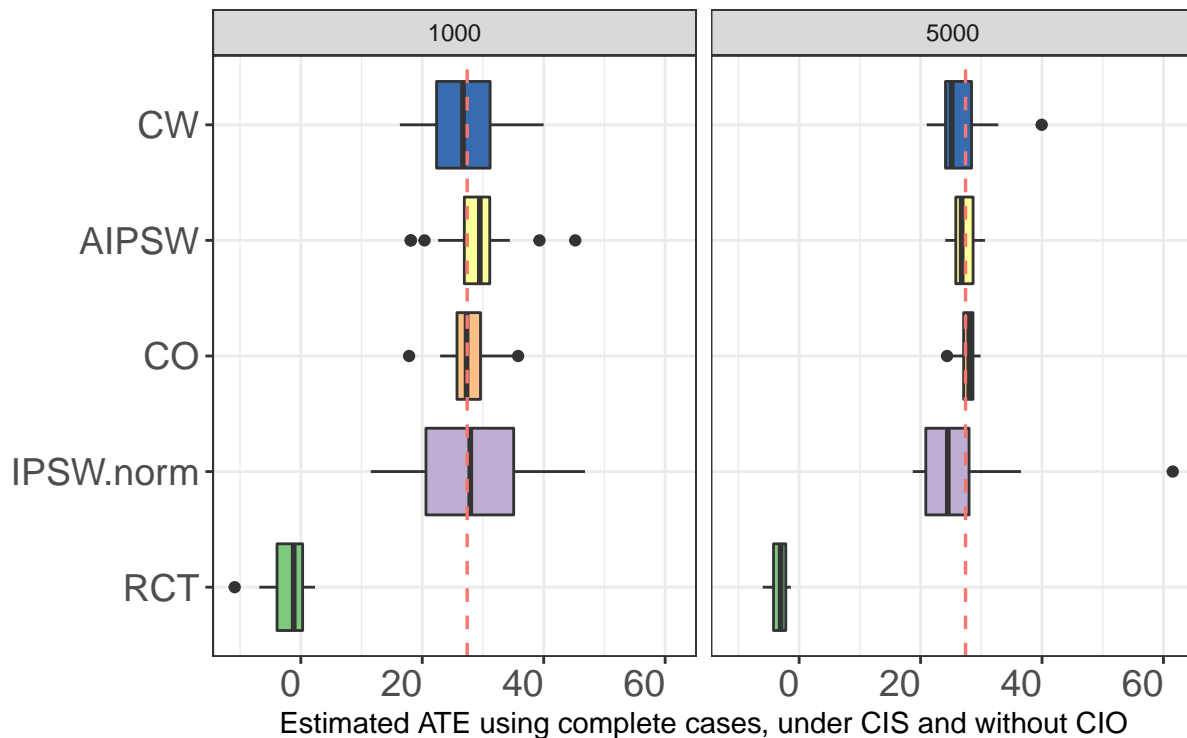
      tmp$n <- n
      tmp$link <- link
      results_cc <- rbind(results_cc, tmp)
    }
  }
}
```

6.2.1.1 Use only complete cases (for logistic+linear regressions)

```
## [1] "results_rep20_diffpropNA_noCIS_noCIO_linklinear_snr5_MCAR_MCAR_propNA0.1_0.5_corTRUE_n1000_2000"
```

```
## # A tibble: 10 x 5
```

	variable	bias	n	method	link
	<fct>	<dbl>	<dbl>	<chr>	<chr>
## 1	RCT	-29.6	1000	glm	linear
## 2	RCT	-30.7	5000	glm	linear
## 3	IPSW.norm	0.505	1000	glm	linear
## 4	IPSW.norm	-1.08	5000	glm	linear
## 5	CO	0.349	1000	glm	linear
## 6	CO	0.261	5000	glm	linear
## 7	AIPSW	1.71	1000	glm	linear
## 8	AIPSW	-0.161	5000	glm	linear
## 9	CW	0.126	1000	glm	linear
## 10	CW	-0.710	5000	glm	linear



-- Population ATE

```
methods <- c("glm")
nb_mi <- c(5,10)

if (!results_exist) {
  results_mlmi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                          p = 4, Sigma = Sigma, snr=snr,
```

```

na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i

method=methods, nb_strat=1,

do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
verbose=T, verbose_intern = F)

tmp$strategy <- "multi-level-woY"
tmp$n <- n
tmp$link <- link
results_mlmi <- rbind(results_mlmi, tmp)
}
}
}

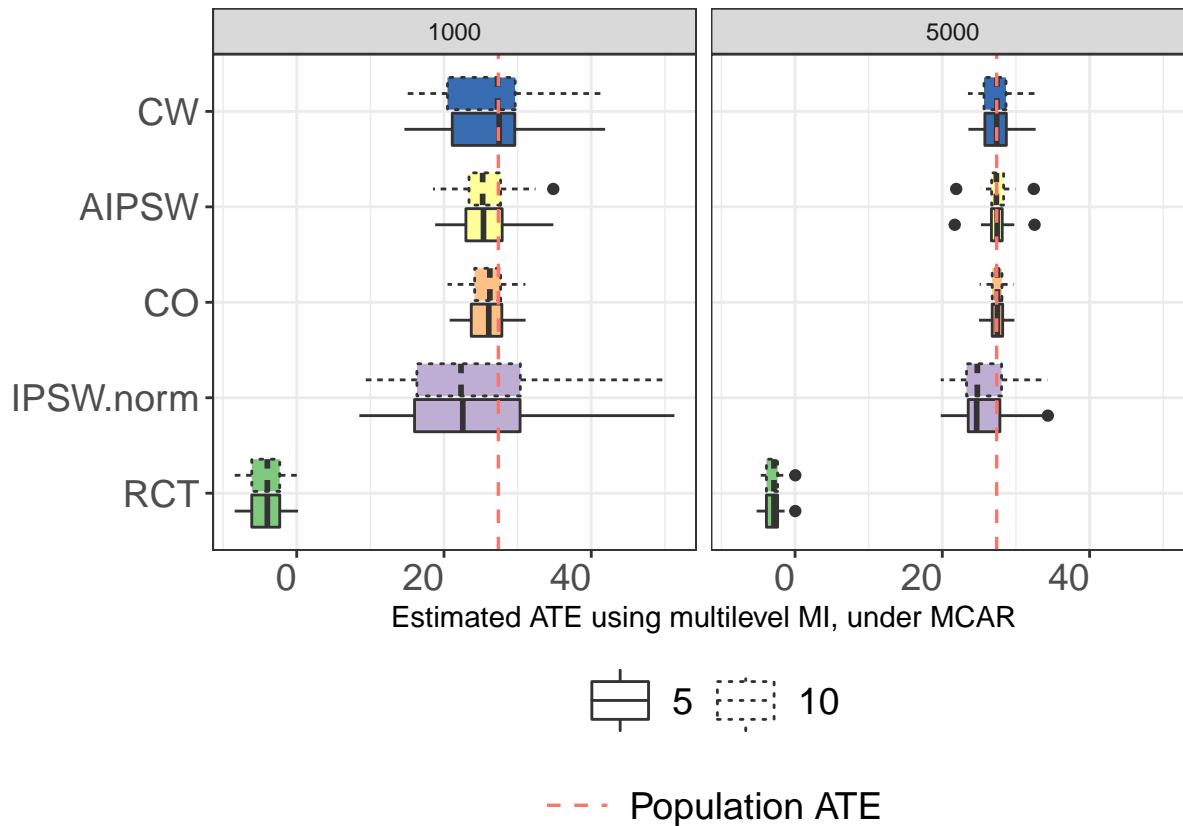
```

6.2.1.2 Use multilevel MI

```

## [1] "results_rep20_diffpropNA_noCIS_linklinear_snr5_MCAR_MCAR_propNA0.1_0.5_corTRUE_n1000_2000_3000_4
## # A tibble: 10 x 7
##   variable      bias nb_mi strategy      n method link
##   <fct>      <dbl> <dbl> <chr>    <dbl> <chr> <chr>
## 1 RCT        -31.3     10 multi-level-woY 1000 glm linear
## 2 RCT        -30.4     10 multi-level-woY 5000 glm linear
## 3 IPSW.norm  -2.53     10 multi-level-woY 1000 glm linear
## 4 IPSW.norm  -1.29     10 multi-level-woY 5000 glm linear
## 5 CO         -1.50     10 multi-level-woY 1000 glm linear
## 6 CO          0.00550  10 multi-level-woY 5000 glm linear
## 7 AIPSW      -1.59     10 multi-level-woY 1000 glm linear
## 8 AIPSW       0.110     10 multi-level-woY 5000 glm linear
## 9 CW        -0.909     10 multi-level-woY 1000 glm linear
## 10 CW         0.0636     10 multi-level-woY 5000 glm linear

```

```

methods <- c("grf")

if (!results_exist) {
  results_mia <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_mia <- rbind(results_mia, tmp)
    }
  }
}

```

6.2.1.3 Use MIA to handle incomplete cases

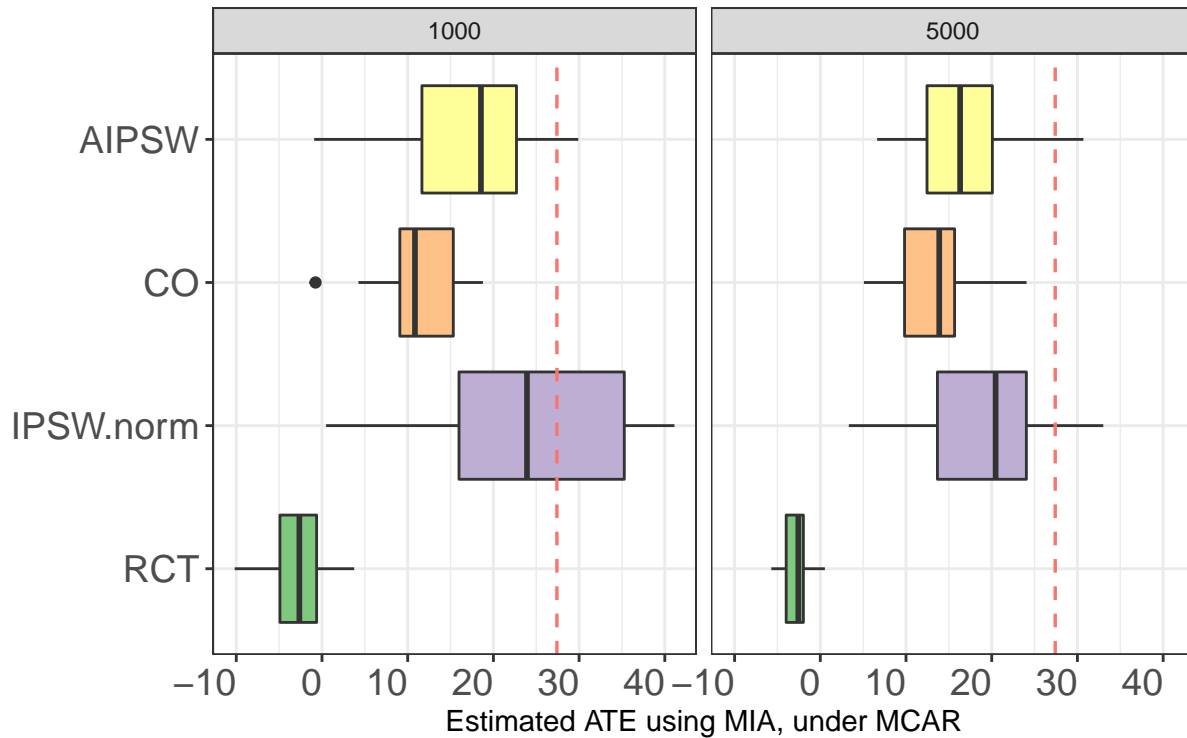
```

## [1] "results_rep20_diffpropNA_noCIS_noCIO_linklinear_snr5_MCAR_MCAR_propNA0.1_0.5_corTRUE_n1000_2000"

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -30.2   1000 grf    linear

```

```
## 2 RCT      -30.2  5000 grf    linear
## 3 IPSW.norm -3.35 1000 grf    linear
## 4 IPSW.norm -8.50 5000 grf    linear
## 5 CO       -15.9 1000 grf    linear
## 6 CO       -14.0 5000 grf    linear
## 7 AIPSW    -10.1 1000 grf    linear
## 8 AIPSW    -10.9 5000 grf    linear
```



--- Population ATE

6.2.2 RCT: MCAR 0.05, RWE: MCAR 0.22

```
mechanism <- c("MCAR", "MCAR")
prop.miss <- c(0.05, 0.22)
```

```
methods <- c("glm")
```

```
if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T, verbose_intern = F)
```

```

    tmp$n <- n
    tmp$link <- link
    results_cc <- rbind(results_cc, tmp)
  }
}
}

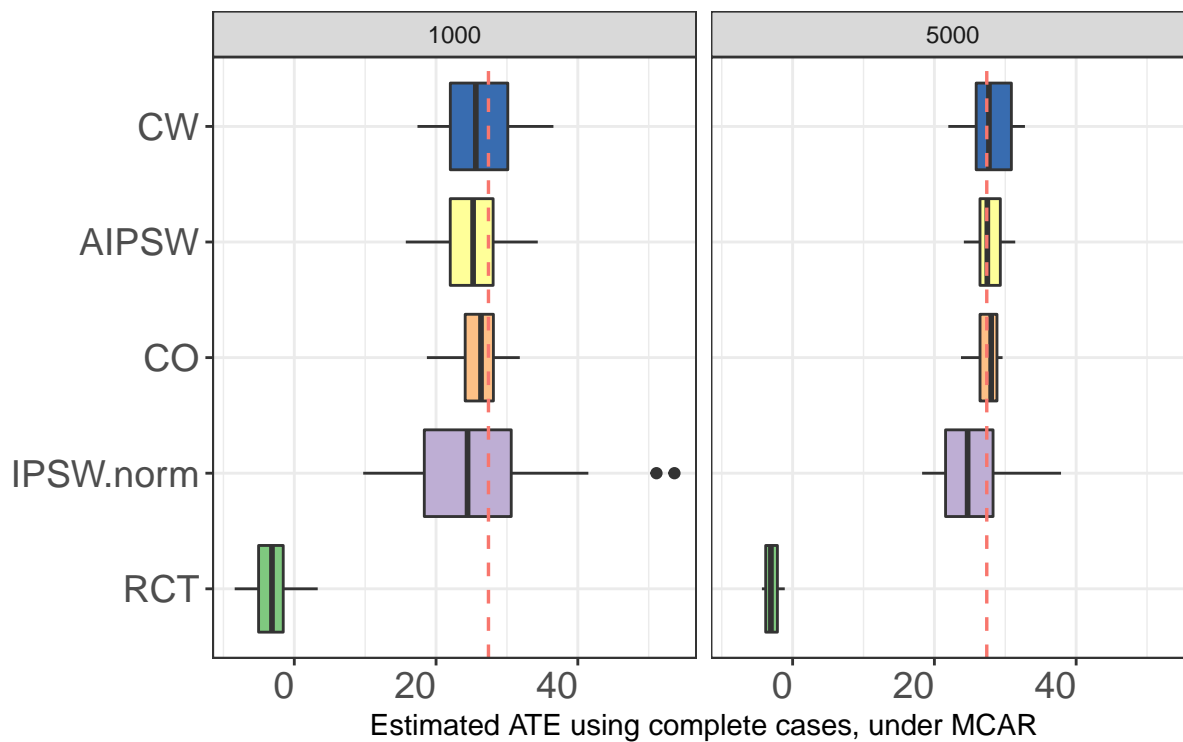
```

6.2.2.1 Use only complete cases (for logistic+linear regressions)

```
## [1] "results_rep20_diffpropNA_noCIS_noCIO_linklinear_snr5_MCAR_MCAR_propNA0.05_0.22_corTRUE_n1000_20"
```

```
## # A tibble: 10 x 5
```

##	variable	bias	n	method	link
##	<fct>	<dbl>	<dbl>	<chr>	<chr>
## 1	RCT	-30.7	1000	glm	linear
## 2	RCT	-30.3	5000	glm	linear
## 3	IPSW.norm	-0.781	1000	glm	linear
## 4	IPSW.norm	-1.99	5000	glm	linear
## 5	CO	-1.22	1000	glm	linear
## 6	CO	0.204	5000	glm	linear
## 7	AIPSW	-1.70	1000	glm	linear
## 8	AIPSW	0.333	5000	glm	linear
## 9	CW	-1.21	1000	glm	linear
## 10	CW	0.507	5000	glm	linear



--- Population ATE

```

methods <- c("glm")
nb_mi <- c(5,10)

```

```

if (!results_exist) {
  results_mlmi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i

      method=methods, nb_strat=1,

                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$strategy <- "multi-level-woY"
      tmp$n <- n
      tmp$link <- link
      results_mlmi <- rbind(results_mlmi, tmp)
    }
  }
}

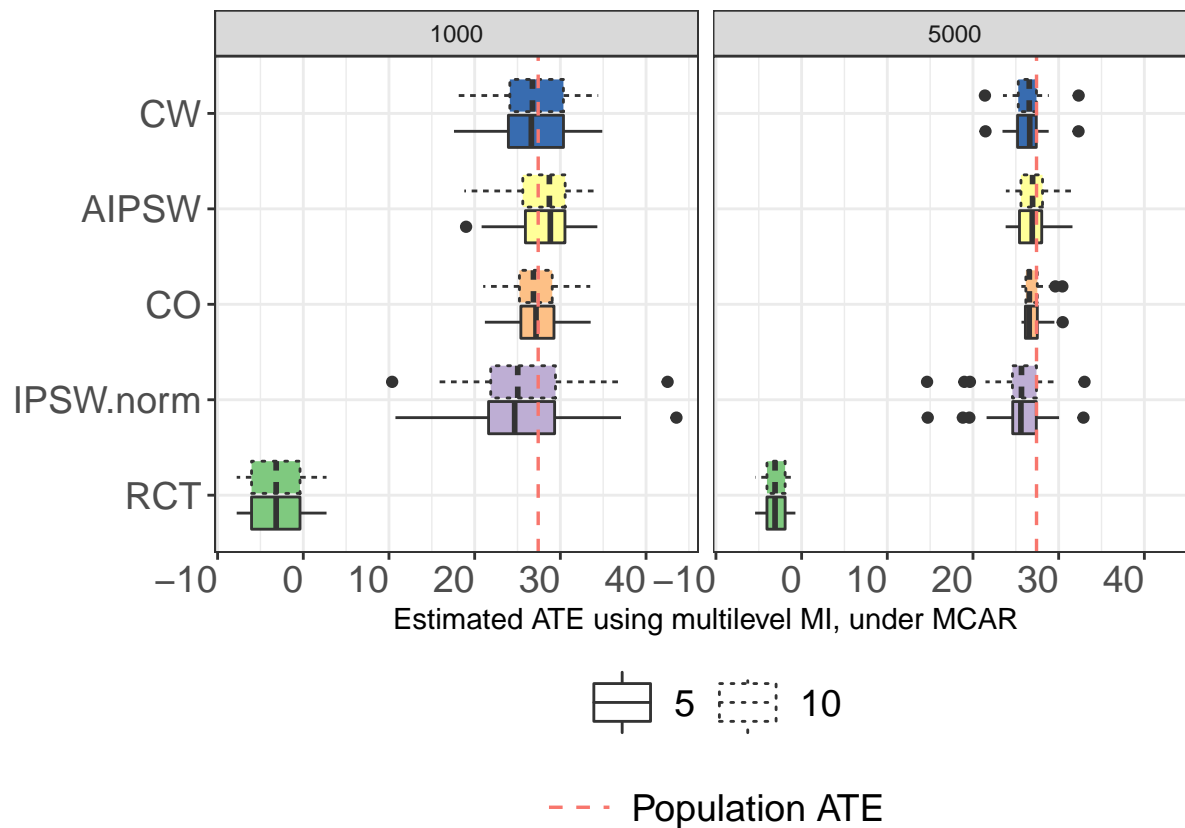
```

6.2.2.2 Use multilevel MI

```
## [1] "results_rep20_diffpropNA_noCIS_linklinear_snr5_MCAR_MCAR_propNA0.05_0.22_corTRUE_n1000_2000_3000"
```

```
## # A tibble: 10 x 7
```

	variable	bias	nb_mi	strategy	n	method	link
	<fct>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<chr>
## 1	RCT	-30.4	10	multi-level-woY	1000	glm	linear
## 2	RCT	-30.5	10	multi-level-woY	5000	glm	linear
## 3	IPSW.norm	-2.04	10	multi-level-woY	1000	glm	linear
## 4	IPSW.norm	-2.20	10	multi-level-woY	5000	glm	linear
## 5	CO	-0.00417	10	multi-level-woY	1000	glm	linear
## 6	CO	-0.317	10	multi-level-woY	5000	glm	linear
## 7	AIPSW	0.417	10	multi-level-woY	1000	glm	linear
## 8	AIPSW	-0.364	10	multi-level-woY	5000	glm	linear
## 9	CW	-0.444	10	multi-level-woY	1000	glm	linear
## 10	CW	-1.03	10	multi-level-woY	5000	glm	linear



```

methods <- c("grf")

if (!results_exist) {
  results_mia <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_mia <- rbind(results_mia, tmp)
    }
  }
}

```

6.2.2.3 Use MIA to handle incomplete cases

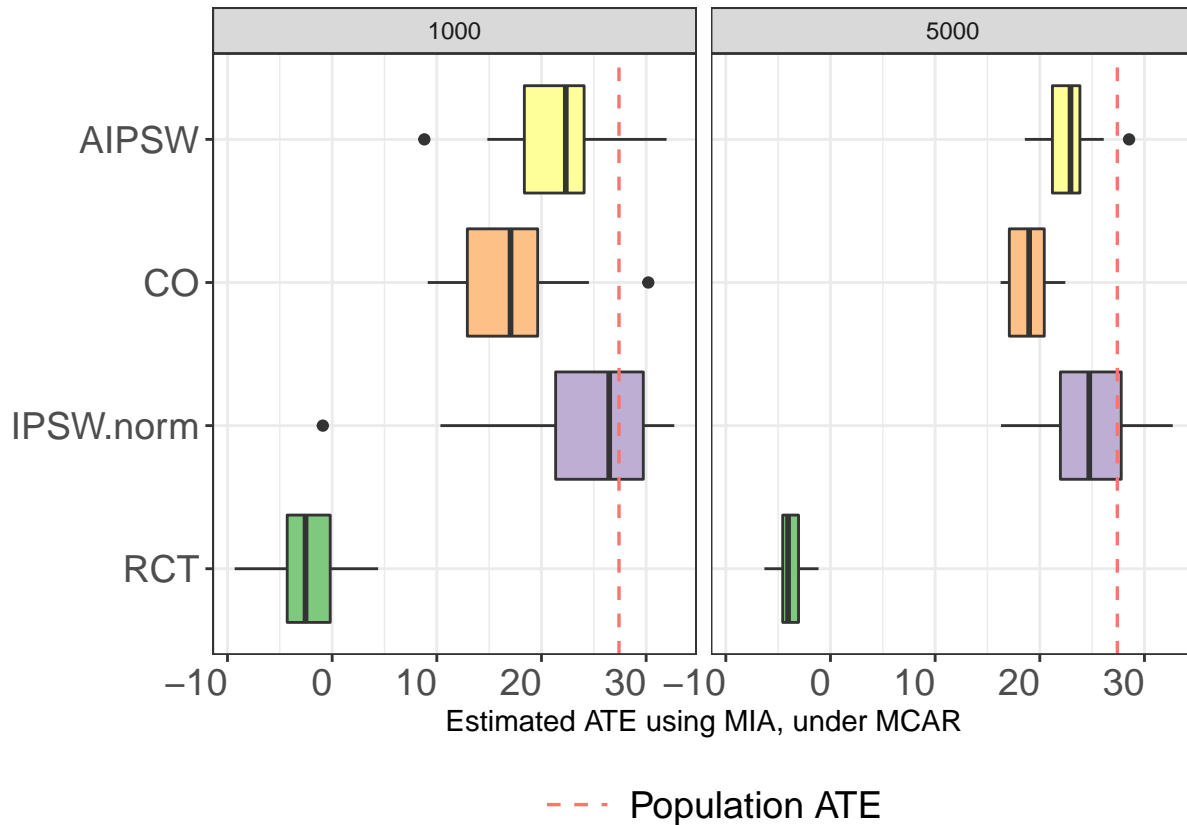
```

## [1] "results_rep20_diffpropNA_noCIS_linklinear_snr5_MCAR_MCAR_propNA0.05_0.22_corTRUE_n1000_2000_3000"

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -29.6   1000 grf    linear

```

```
## 2 RCT      -31.2  5000 grf    linear
## 3 IPSW.norm -3.24 1000 grf    linear
## 4 IPSW.norm -2.74 5000 grf    linear
## 5 CO       -10.1  1000 grf    linear
## 6 CO       -8.40 5000 grf    linear
## 7 AIPSW    -5.96 1000 grf    linear
## 8 AIPSW    -4.57 5000 grf    linear
```



6.2.3 Equilibrated sample sizes - RCT: MCAR 0.1, RWE: MCAR 0.5

```
mechanism <- c("MCAR", "MCAR")
prop.miss <- c(0.1, 0.5)
```

```
methods <- c("glm")
```

```
if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T, verbose_intern = F)
```

```

    tmp$n <- n
    tmp$link <- link
    results_cc <- rbind(results_cc, tmp)
  }
}
}

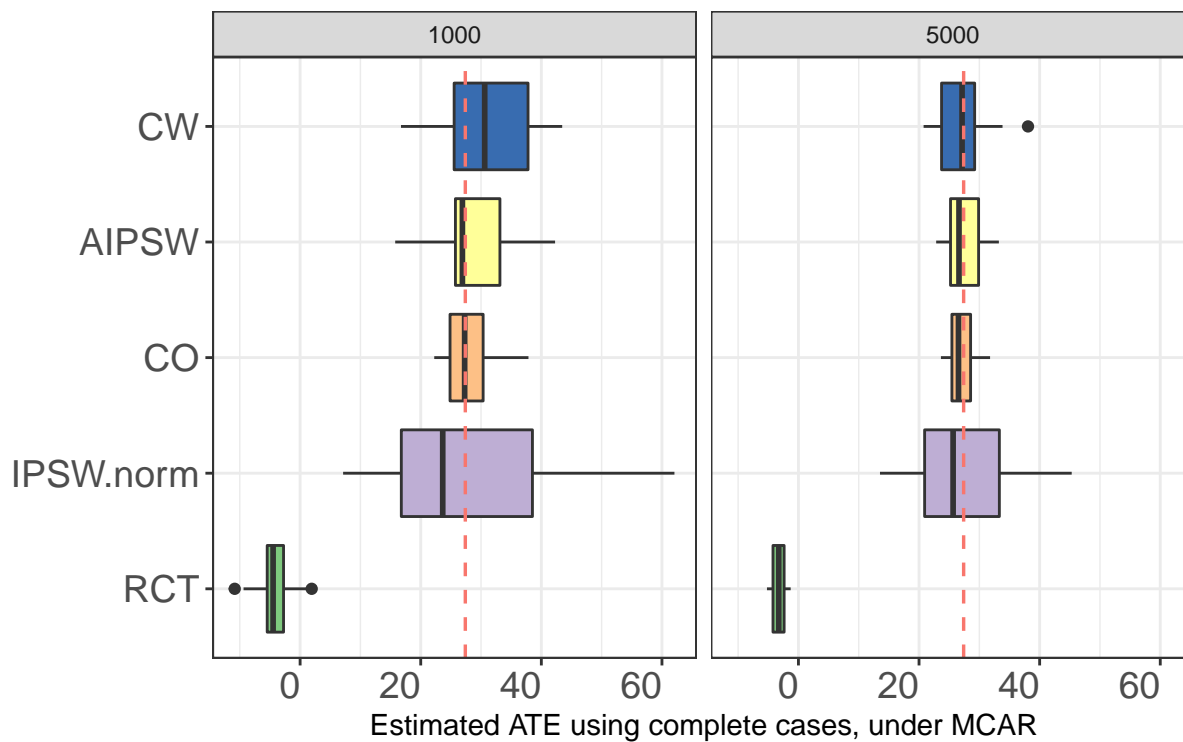
```

6.2.3.1 Use only complete cases (for logistic+linear regressions)

```
## [1] "results_rep20_diffpropNA_samesize_noCIS_noCIO_linklinear_snr5_MCAR_MCAR_propNA0.1_0.5_corTRUE_n"
```

```
## # A tibble: 10 x 5
```

##	variable	bias	n	method	link
##	<fct>	<dbl>	<dbl>	<chr>	<chr>
## 1	RCT	-31.5	1000	glm	linear
## 2	RCT	-30.7	5000	glm	linear
## 3	IPSW.norm	1.98	1000	glm	linear
## 4	IPSW.norm	-0.633	5000	glm	linear
## 5	CO	0.639	1000	glm	linear
## 6	CO	-0.187	5000	glm	linear
## 7	AIPSW	1.28	1000	glm	linear
## 8	AIPSW	0.249	5000	glm	linear
## 9	CW	3.57	1000	glm	linear
## 10	CW	-0.208	5000	glm	linear



--- Population ATE

```

methods <- c("glm")
nb_mi <- c(5,10)

```

```

if (!results_exist) {
  results_mlmi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i

      method=methods, nb_strat=1,

                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$strategy <- "multi-level-woY"
      tmp$n <- n
      tmp$link <- link
      results_mlmi <- rbind(results_mlmi, tmp)
    }
  }
}

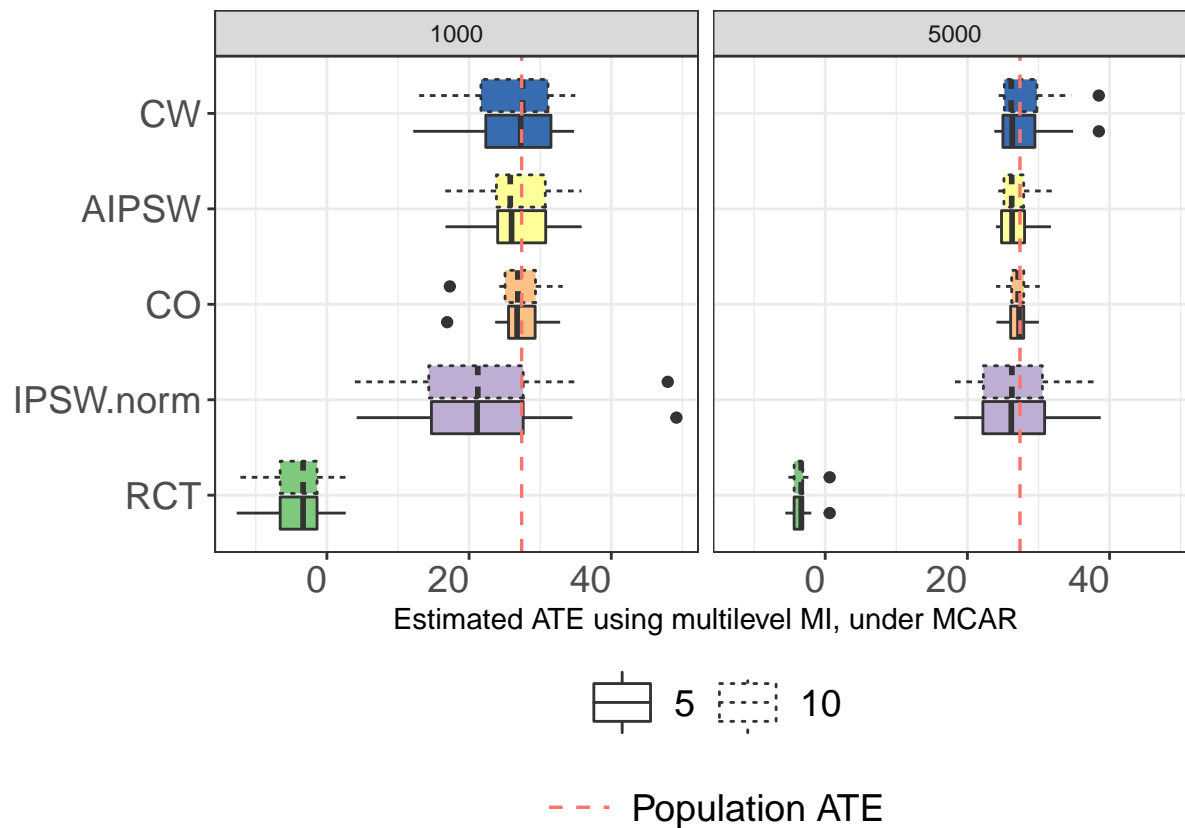
```

6.2.3.2 Use multilevel MI

```
## [1] "results_rep20_diffpropNA_samesize_noCIS_linklinear_snr5_MCAR_MCAR_propNA0.1_0.5_corTRUE_n1000_2"
```

```
## # A tibble: 10 x 7
```

	variable	bias	nb_mi	strategy	n	method	link
	<fct>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<chr>
## 1	RCT	-31.3	10	multi-level-woY	1000	glm	linear
## 2	RCT	-30.9	10	multi-level-woY	5000	glm	linear
## 3	IPSW.norm	-5.94	10	multi-level-woY	1000	glm	linear
## 4	IPSW.norm	-0.439	10	multi-level-woY	5000	glm	linear
## 5	CO	-0.471	10	multi-level-woY	1000	glm	linear
## 6	CO	-0.278	10	multi-level-woY	5000	glm	linear
## 7	AIPSW	-0.568	10	multi-level-woY	1000	glm	linear
## 8	AIPSW	-0.671	10	multi-level-woY	5000	glm	linear
## 9	CW	-1.63	10	multi-level-woY	1000	glm	linear
## 10	CW	0.313	10	multi-level-woY	5000	glm	linear



```

methods <- c("grf")

if (!results_exist) {
  results_mia <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_mia <- rbind(results_mia, tmp)
    }
  }
}

```

6.2.3.3 Use MIA to handle incomplete cases

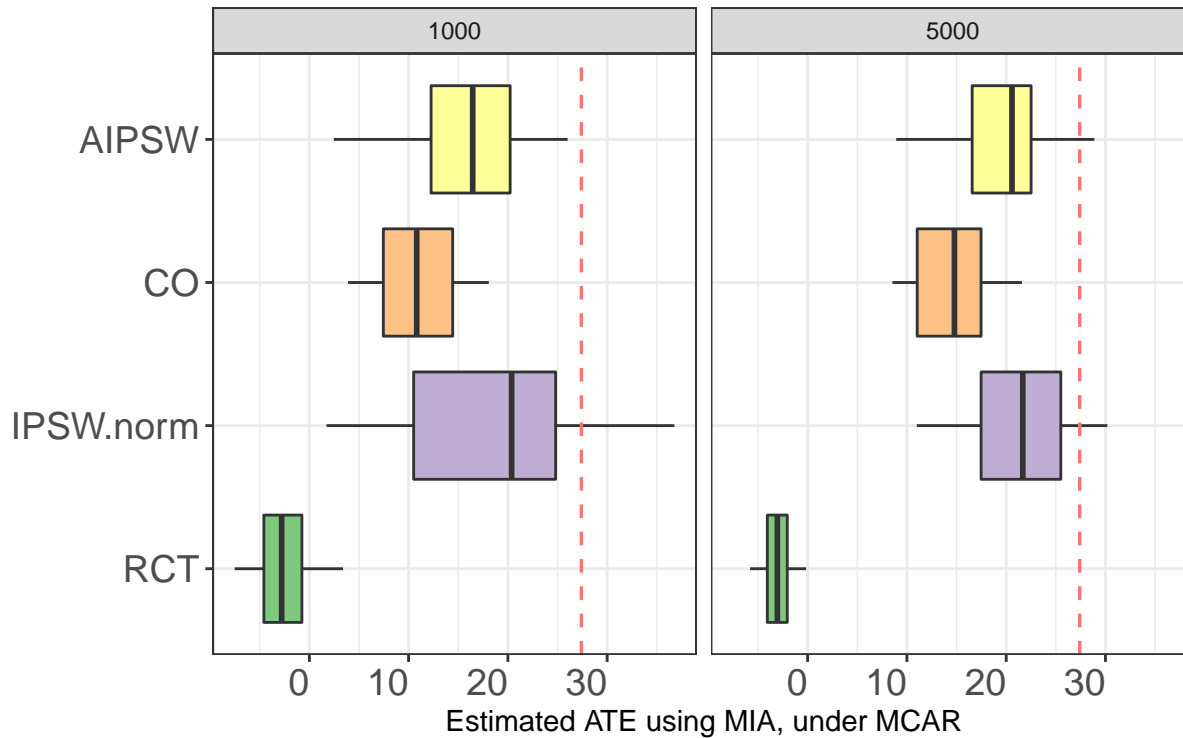
```

## [1] "results_rep20_diffpropNA_samesize_noCIS_linklinear_snr5_MCAR_MCAR_propNA0.1_0.5_corTRUE_n1000_20"

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -30.1  1000 grf   linear

```

```
## 2 RCT      -30.4  5000 grf    linear
## 3 IPSW.norm -8.83 1000 grf    linear
## 4 IPSW.norm -6.73 5000 grf    linear
## 5 CO       -16.2 1000 grf    linear
## 6 CO       -12.8 5000 grf    linear
## 7 AIPSW    -11.6 1000 grf    linear
## 8 AIPSW    -7.98 5000 grf    linear
```



--- Population ATE

6.2.4 RCT: MAR 0.1, RWE: MAR 0.5

```
mechanism <- c("MAR", "MAR")
prop.miss <- c(0.1, 0.5)
```

```
methods <- c("glm")
```

```
if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         complete_cases=TRUE,
                                         verbose=T, verbose_intern = F)
```

```

    tmp$n <- n
    tmp$link <- link
    results_cc <- rbind(results_cc, tmp)
  }
}
}

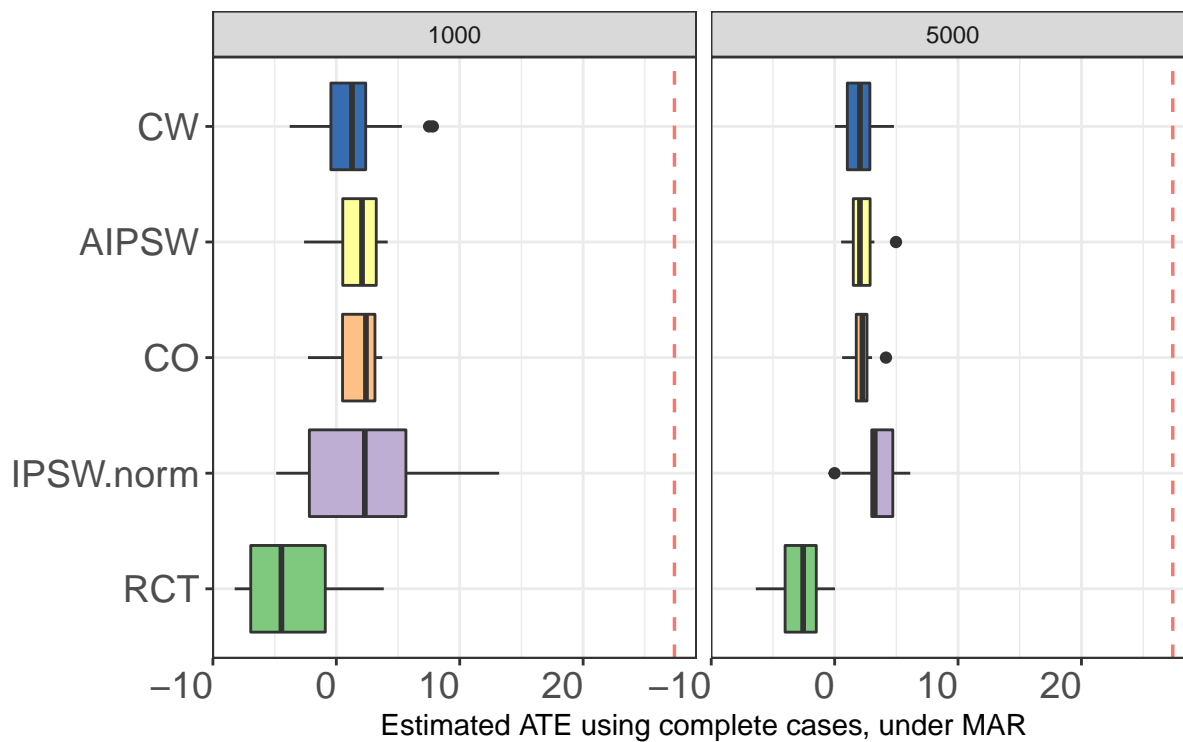
```

6.2.4.1 Use only complete cases (for logistic+linear regressions)

```
## [1] "results_rep20_diffpropNA_noCIS_linklinear_snr5_MAR_MAR_propNA0.1_0.5_corTRUE_n1000_2000_3000_4000"
```

```
## # A tibble: 10 x 5
```

```
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -31.1  1000  glm   linear
## 2 RCT        -30.2  5000  glm   linear
## 3 IPSW.norm  -25.0  1000  glm   linear
## 4 IPSW.norm  -23.9  5000  glm   linear
## 5 CO         -25.7  1000  glm   linear
## 6 CO         -25.2  5000  glm   linear
## 7 AIPSW      -25.8  1000  glm   linear
## 8 AIPSW      -25.2  5000  glm   linear
## 9 CW         -26.0  1000  glm   linear
## 10 CW        -25.4  5000  glm   linear
```



```

methods <- c("glm")
nb_mi <- c(5,10)

```

```

if (!results_exist) {
  results_mlmi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i

      method=methods, nb_strat=1,

                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$strategy <- "multi-level-woY"
      tmp$n <- n
      tmp$link <- link
      results_mlmi <- rbind(results_mlmi, tmp)
    }
  }
}

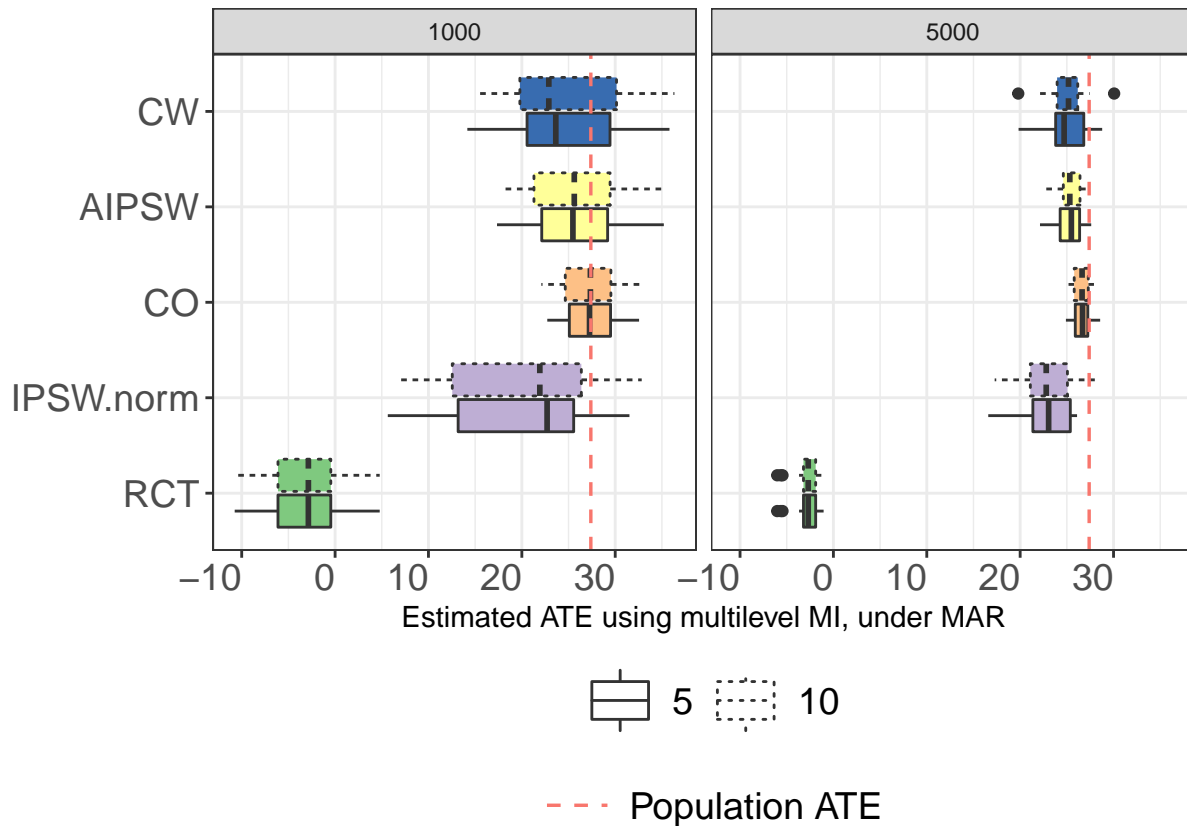
```

6.2.4.2 Use multilevel MI

```
## [1] "results_rep20_diffpropNA_noCIS_noCIO_linklinear_snr5_MAR_MAR_propNA0.1_0.5_corTRUE_n1000_2000_3"
```

```
## # A tibble: 10 x 7
```

	variable	bias	nb_mi	strategy	n	method	link
	<fct>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<chr>
## 1	RCT	-30.4	10	multi-level-woY	1000	glm	linear
## 2	RCT	-30.3	10	multi-level-woY	5000	glm	linear
## 3	IPSW.norm	-7.06	10	multi-level-woY	1000	glm	linear
## 4	IPSW.norm	-4.62	10	multi-level-woY	5000	glm	linear
## 5	CO	-0.307	10	multi-level-woY	1000	glm	linear
## 6	CO	-0.879	10	multi-level-woY	5000	glm	linear
## 7	AIPSW	-1.69	10	multi-level-woY	1000	glm	linear
## 8	AIPSW	-2.02	10	multi-level-woY	5000	glm	linear
## 9	CW	-2.80	10	multi-level-woY	1000	glm	linear
## 10	CW	-2.36	10	multi-level-woY	5000	glm	linear



```

methods <- c("grf")

if (!results_exist) {
  results_mia <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_mia <- rbind(results_mia, tmp)
    }
  }
}

```

6.2.4.3 Use MIA to handle incomplete cases

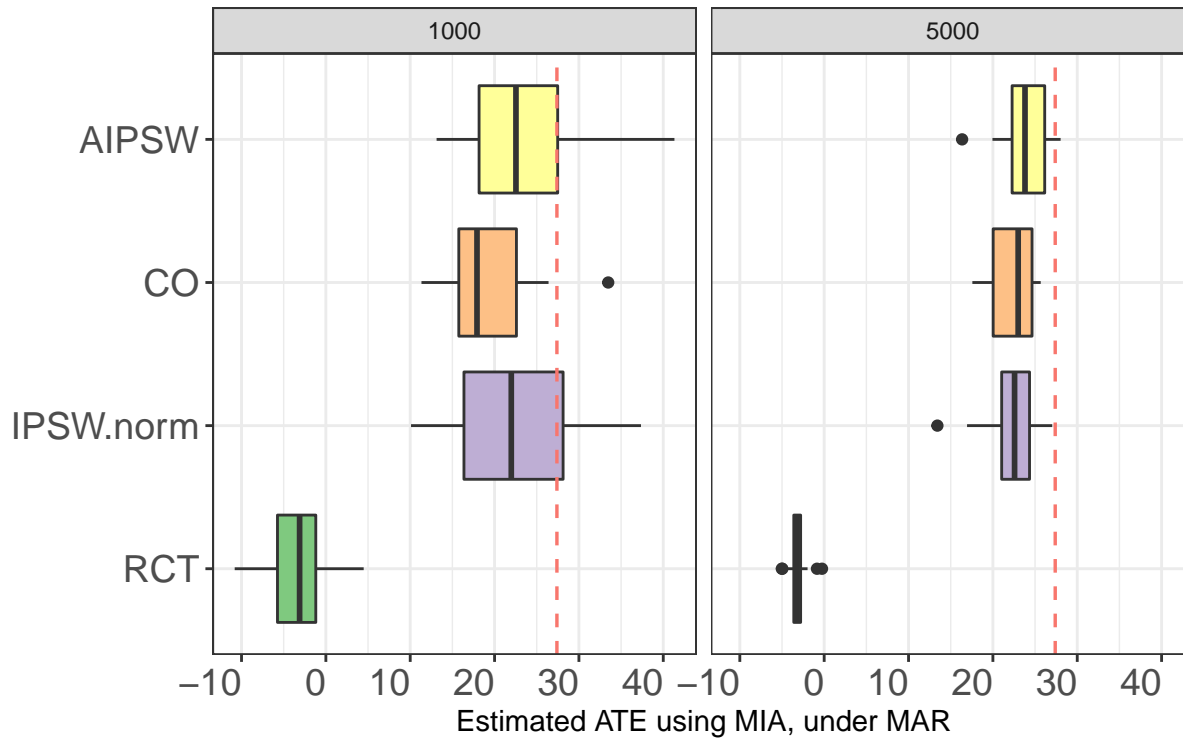
```

## [1] "results_rep20_diffpropNA_noCIS_linklinear_snr5_MAR_MAR_propNA0.1_0.5_corTRUE_n1000_2000_3000_4000"

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -30.5   1000 grf    linear

```

```
## 2 RCT      -30.5  5000 grf    linear
## 3 IPSW.norm -5.04 1000 grf    linear
## 4 IPSW.norm -5.18 5000 grf    linear
## 5 CO       -7.94 1000 grf    linear
## 6 CO       -5.05 5000 grf    linear
## 7 AIPSW    -4.07 1000 grf    linear
## 8 AIPSW    -3.70 5000 grf    linear
```



--- Population ATE

6.2.5 RCT: MAR 0.05, RWE: MAR 0.22

```
mechanism <- c("MAR", "MAR")
prop.miss <- c(0.05, 0.22)
```

```
methods <- c("glm")
```

```
if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
        p = 4, Sigma = Sigma, snr=snr,
        na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
        na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
        method=methods, nb_strat=1,
        complete_cases=TRUE,
        verbose=T, verbose_intern = F)
```

```

    tmp$n <- n
    tmp$link <- link
    results_cc <- rbind(results_cc, tmp)
  }
}
}

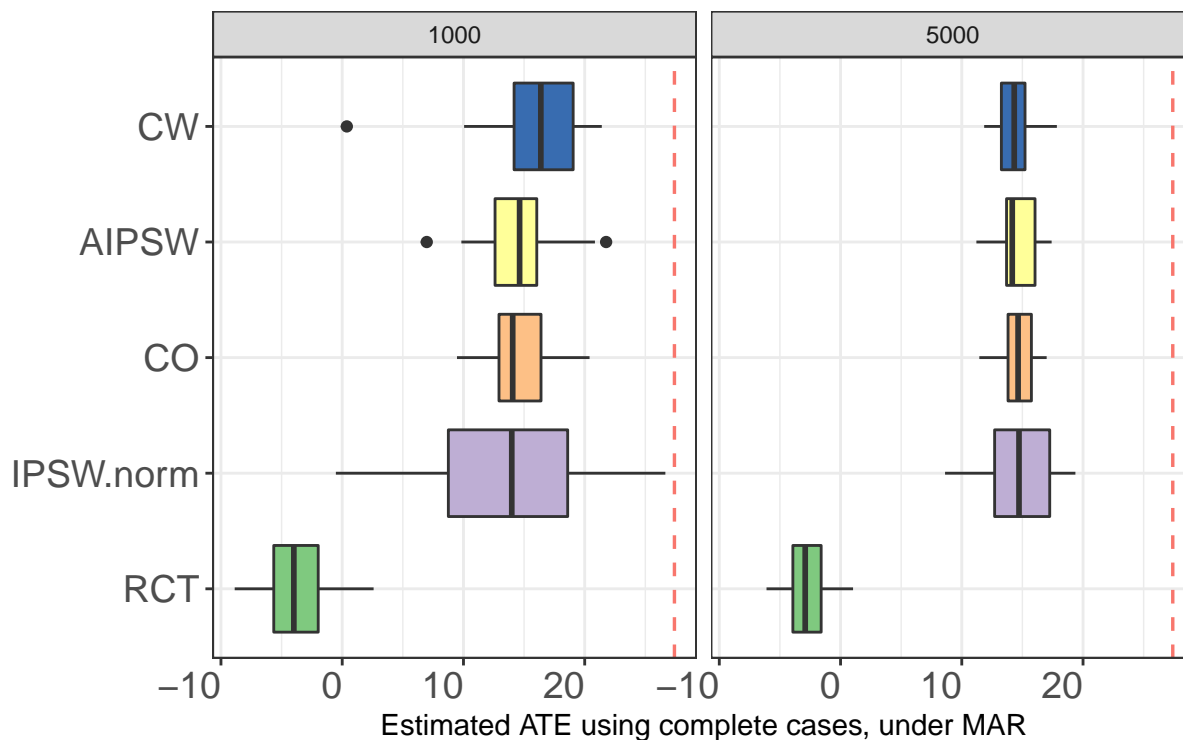
```

6.2.5.1 Use only complete cases (for logistic+linear regressions)

```
## [1] "results_rep20_diffpropNA_noCIS_noCIO_linklinear_snr5_MAR_MAR_propNA0.05_0.22_corTRUE_n1000_2000"
```

```
## # A tibble: 10 x 5
```

```
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -31.3  1000  glm   linear
## 2 RCT        -30.3  5000  glm   linear
## 3 IPSW.norm  -13.4  1000  glm   linear
## 4 IPSW.norm  -12.9  5000  glm   linear
## 5 CO         -12.9  1000  glm   linear
## 6 CO         -12.9  5000  glm   linear
## 7 AIPSW      -12.8  1000  glm   linear
## 8 AIPSW      -12.9  5000  glm   linear
## 9 CW         -11.8  1000  glm   linear
## 10 CW        -12.9  5000  glm   linear
```



--- Population ATE

```

methods <- c("glm")
nb_mi <- c(5,10)

```

```

if (!results_exist) {
  results_mlmi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i

method=methods, nb_strat=1,

                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$strategy <- "multi-level-woY"
      tmp$n <- n
      tmp$link <- link
      results_mlmi <- rbind(results_mlmi, tmp)
    }
  }
}

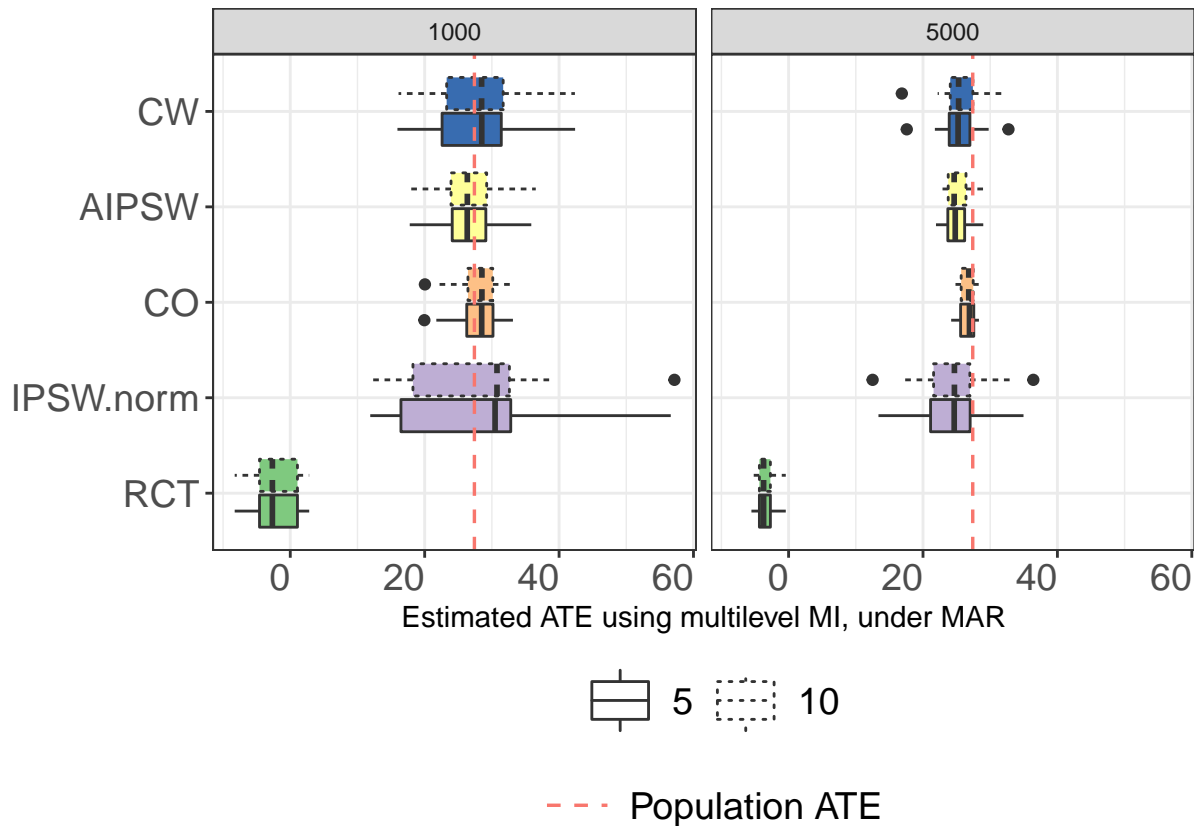
```

6.2.5.2 Use multilevel MI

```
## [1] "results_rep20_diffpropNA_noCIS_linklinear_snr5_MAR_MAR_propNA0.05_0.22_corTRUE_n1000_2000_3000_4"
```

```
## # A tibble: 10 x 7
```

	variable	bias	nb_mi	strategy	n	method	link
	<fct>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<chr>
## 1	RCT	-29.8	10	multi-level-woY	1000	glm	linear
## 2	RCT	-30.9	10	multi-level-woY	5000	glm	linear
## 3	IPSW.norm	0.108	10	multi-level-woY	1000	glm	linear
## 4	IPSW.norm	-2.79	10	multi-level-woY	5000	glm	linear
## 5	CO	0.347	10	multi-level-woY	1000	glm	linear
## 6	CO	-0.835	10	multi-level-woY	5000	glm	linear
## 7	AIPSW	-0.986	10	multi-level-woY	1000	glm	linear
## 8	AIPSW	-2.26	10	multi-level-woY	5000	glm	linear
## 9	CW	0.185	10	multi-level-woY	1000	glm	linear
## 10	CW	-1.75	10	multi-level-woY	5000	glm	linear



```

methods <- c("grf")

if (!results_exist) {
  results_mia <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = 10*n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_mia <- rbind(results_mia, tmp)
    }
  }
}

```

6.2.5.3 Use MIA to handle incomplete cases

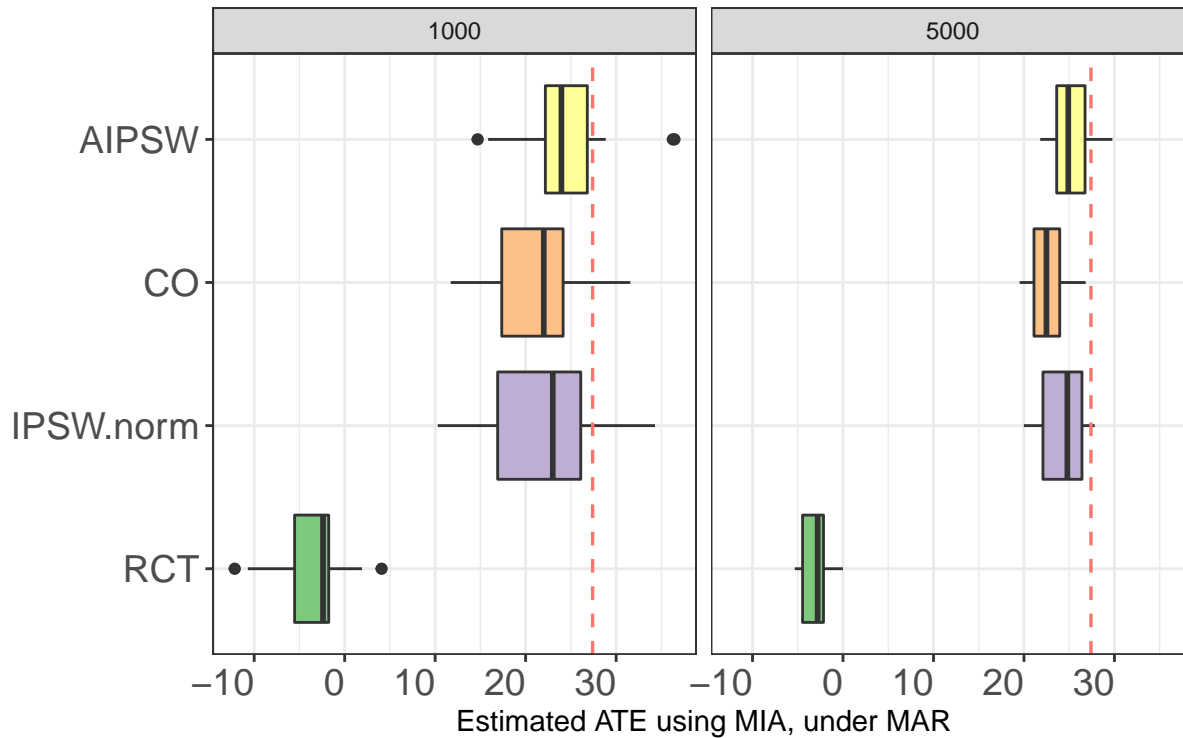
```

## [1] "results_rep20_diffpropNA_noCIS_linklinear_snr5_MAR_MAR_propNA0.05_0.22_corTRUE_n1000_2000_3000_4000_5000_6000_7000_8000_9000_10000"

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -30.6   1000 grf    linear

```

```
## 2 RCT      -30.5  5000 grf    linear
## 3 IPSW.norm -5.19 1000 grf    linear
## 4 IPSW.norm -3.24 5000 grf    linear
## 5 CO       -6.26 1000 grf    linear
## 6 CO       -4.75 5000 grf    linear
## 7 AIPSW    -3.05 1000 grf    linear
## 8 AIPSW    -2.34 5000 grf    linear
```



--- Population ATE

6.2.6 Equilibrated sample sizes - RCT: MAR 0.1, RWE: MAR 0.5

```
mechanism <- c("MAR", "MAR")
prop.miss <- c(0.1, 0.5)
```

```
methods <- c("glm")
```

```
if (!results_exist) {
  results_cc <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = n, link=link, bs0=bs0,
        p = 4, Sigma = Sigma, snr=snr,
        na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
        na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
        method=methods, nb_strat=1,
        complete_cases=TRUE,
        verbose=T, verbose_intern = F)
```

```

    tmp$n <- n
    tmp$link <- link
    results_cc <- rbind(results_cc, tmp)
  }
}
}

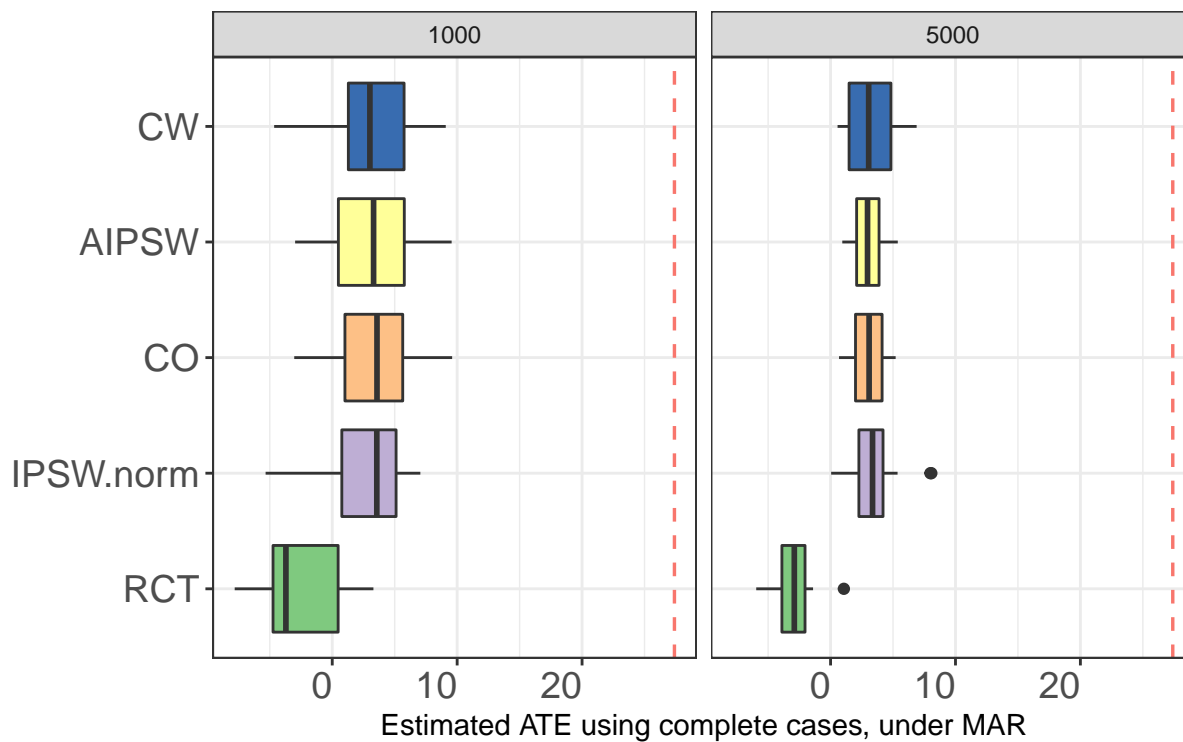
```

6.2.6.1 Use only complete cases (for logistic+linear regressions)

```
## [1] "results_rep20_diffpropNA_samesize_noCIS_linklinear_snr5_MAR_MAR_propNA0.1_0.5_corTRUE_n1000_2000"
```

```
## # A tibble: 10 x 5
```

```
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -29.9  1000 glm   linear
## 2 RCT        -30.4  5000 glm   linear
## 3 IPSW.norm  -24.8  1000 glm   linear
## 4 IPSW.norm  -24.0  5000 glm   linear
## 5 CO         -24.3  1000 glm   linear
## 6 CO         -24.4  5000 glm   linear
## 7 AIPSW      -24.4  1000 glm   linear
## 8 AIPSW      -24.4  5000 glm   linear
## 9 CW         -24.5  1000 glm   linear
## 10 CW        -24.1  5000 glm   linear
```



--- Population ATE

```

methods <- c("glm")
nb_mi <- c(5,10)

```

```

if (!results_exist) {
  results_mlmi <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions, n = n, m = n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i

      method=methods, nb_strat=1,

                                         do_mi=T, nb_mi=nb_mi, strategy="multi-level-woY",
                                         verbose=T, verbose_intern = F)

      tmp$strategy <- "multi-level-woY"
      tmp$n <- n
      tmp$link <- link
      results_mlmi <- rbind(results_mlmi, tmp)
    }
  }
}

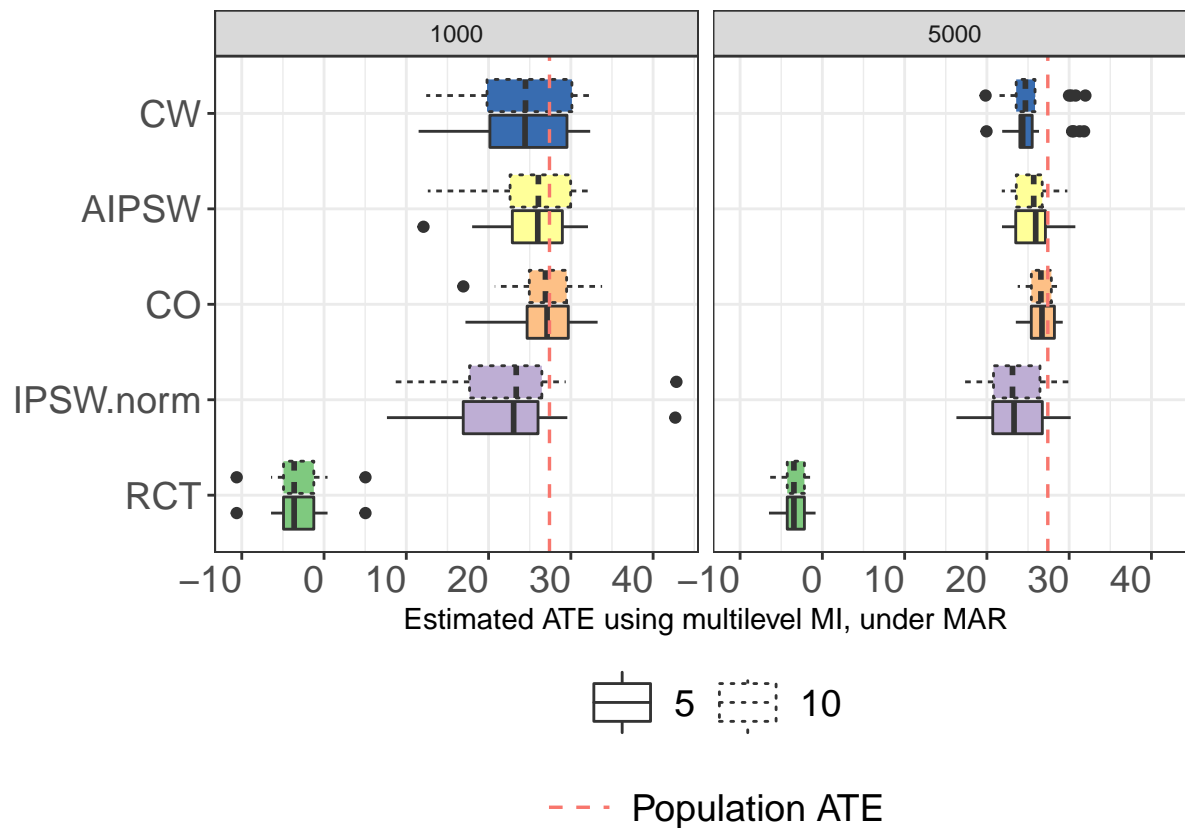
```

6.2.6.2 Use multilevel MI

```
## [1] "results_rep20_diffpropNA_samesize_noCIS_linklinear_snr5_MAR_MAR_propNA0.1_0.5_corTRUE_n1000_2000"
```

```
## # A tibble: 10 x 7
```

	variable	bias	nb_mi	strategy	n	method	link
	<fct>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<chr>
## 1	RCT	-30.5	10	multi-level-woY	1000	glm	linear
## 2	RCT	-30.7	10	multi-level-woY	5000	glm	linear
## 3	IPSW.norm	-4.77	10	multi-level-woY	1000	glm	linear
## 4	IPSW.norm	-3.66	10	multi-level-woY	5000	glm	linear
## 5	CO	-0.596	10	multi-level-woY	1000	glm	linear
## 6	CO	-0.834	10	multi-level-woY	5000	glm	linear
## 7	AIPSW	-1.83	10	multi-level-woY	1000	glm	linear
## 8	AIPSW	-2.07	10	multi-level-woY	5000	glm	linear
## 9	CW	-2.70	10	multi-level-woY	1000	glm	linear
## 10	CW	-2.10	10	multi-level-woY	5000	glm	linear



```

methods <- c("grf")

if (!results_exist) {
  results_mia <- c()
  for (link in links){
    for (n in n_range){
      tmp <- compute_estimators_and_store(rep = repetitions_long, n = n, m = n, link=link, bs0=bs0,
                                         p = 4, Sigma = Sigma, snr=snr,
                                         na_rct=list(mechanism=mechanism[1], prop_miss=prop.miss[1], i
                                         na_rwe=list(mechanism=mechanism[2], prop_miss=prop.miss[2], i
                                         method=methods, nb_strat=1,
                                         verbose=T, verbose_intern = F)

      tmp$n <- n
      tmp$link <- link
      results_mia <- rbind(results_mia, tmp)
    }
  }
}

```

6.2.6.3 Use MIA to handle incomplete cases

```

## [1] "results_rep20_diffpropNA_samesize_noCIS_linklinear_snr5_MAR_MAR_propNA0.1_0.5_corTRUE_n1000_2000"

## # A tibble: 8 x 5
##   variable    bias      n method link
##   <fct>      <dbl> <dbl> <chr> <chr>
## 1 RCT        -30.3   1000 grf    linear

```

##	2	RCT	-30.4	5000	grf	linear
##	3	IPSW.norm	-5.79	1000	grf	linear
##	4	IPSW.norm	-2.98	5000	grf	linear
##	5	CO	-5.41	1000	grf	linear
##	6	CO	-3.96	5000	grf	linear
##	7	AIPSW	-3.31	1000	grf	linear
##	8	AIPSW	-2.20	5000	grf	linear

