1 Using R for Simulating Network-Dependent Observational Data

We start by defining the distribution of the observed network graph. In this example we choose the preferential attachment network model with power law node degree distribution, with the sampling function provided below.

```
require("igraph")
require("simcausal")
require("ggraph")
generate.igraph.prefattach <- function(n, power, zero.appeal, m, ...) {
  g <- sample_pa(n, power = power, zero.appeal = zero.appeal, m = m)
  g <- as.directed(as.undirected(g, mode = "collapse"), mode = "mutual")
  sparse_AdjMat <- simcausal::igraph.to.sparseAdjMat(g)
  NetInd_out <- simcausal::sparseAdjMat.to.NetInd(sparse_AdjMat)
  return(NetInd_out$NetInd_k)
}
```

The above network distribution is then added to a DAG object, which will define the observed data-generating distribution.

```
D <- DAG.empty()
Net.prefattach <- network("Net", netfun = "generate.igraph.prefattach", power = 0.5, zero.appeal = 5, m = 5)
```

Next, we define the distributions of the baseline covariates, as shown below. Note that we define the baseline indicator `HUB`, which indicates if a person has more or equal to 25 friends. We also define the baseline covariate `PA`, which indicates if a person is physically active at baseline and we define the network baseline summary `nF.PA`, which calculates the total number of friends of a person who are physically active.

```
D <- D + Net.prefattach +
  node("LatentW", distr = "rcat.b0", probs = c(0.0494, 0.1823, 0.2806, 0.2680,0.1651, 0.0546)) +
  node("HUB", distr = "rconst", const = ifelse(nF >= 25, 1, 0)) +
  node("W1", distr = "rcat.b1", probs = c(0.0494, 0.1823, 0.2806, 0.2680,0.1651, 0.0546)) +
  node("W2", distr = "rbern", prob = plogis(-0.2)) +
  node("WNoise", distr = "rbern", prob = plogis(-0.4)) +
  node("corrW.F1", distr = "rbern", prob = plogis(-8 + 2*LatentW + 2*LatentW)) +
  node("corrW.F2", distr = "rbern", prob = plogis(-6 + 1.5*LatentW + 1.5*LatentW)) +
  node("corrW.F3", distr = "rbern", prob = plogis(-6 + 1.5*LatentW + 1.5*LatentW)) +
  node("corrW.F4", distr = "rbern", prob = plogis(-4 + LatentW + LatentW)) +
  node("corrW.F5", distr = "rbern", prob = plogis(-4 + LatentW + LatentW)) +
# Status of being physically active at baseline:
  node("PA", distr = "rbern", prob = W2*0.05 + (1-W2)*0.15) +
# Total number of physically active friends:
```

```
  node("nF.PA", distr = "rconst", const = sum(PA[[1:Kmax]]), replaceNAw0 = TRUE)
```

As a next step we randomly assign the binary exposure, `A`, to 25% of the population. This exposure corresponds with an informational campaign about the benefits of physical exercise and is intended to promote and sustain attendance of the local gym by community members.

```
D <- D + node("A", distr = "rbern", prob = 0.25)
```

Next, we define some network summary measure, `sum.net.A3`, as shown below, which depends on the exposures of individuals' friends, as well as their friends' baseline covariate values.

```
D <- D + node("sum.net.A", distr = "rconst",
  const = (sum(A[[1:Kmax]])*(HUB==0) + sum((W1[[1:Kmax]] > 4)*A[[1:Kmax]])*(HUB==1)),
  replaceNAw0 = TRUE)
```

We define the binary outcome `Y` below, which is defined as an indicator of sustaining a membership in a local gym for the duration of 6 months following the intervention `A`. Note that we assumed that each `Y` depends on the individual exposure and baseline covariates. It also depends on the network summary `sum.net.A3` defined above, as well as `nF.PA`, which represents the total number of friends of the individual who were physically active `PA=1` at baseline.

```
D <- D +
  node("Y", distr = "rbern", prob = plogis(ifelse(PA == 1,
    +5 - 15*(nF.PA < 1),
    -8.0 + 0.25*A) +
    +0.5*sum.net.A + 0.25*nF.PA*sum.net.A + 0.5*nF.PA +
    +0.5*(W1-1) - 0.58*W2 +
    -0.5*(3.477-1) + 0.58*0.4496 +
    +4*corrW.F1 -2*corrW.F2 -2*corrW.F3 +2*corrW.F4 -2*corrW.F5 +
    -4*0.6841   +2*0.6727   +2*0.6724   -2*0.6513   +2*0.6538),
    replaceNAw0 = TRUE)
```

Finally, we define the data-generating distribution based on the preferential attachment network model, as shown below.

```
D.prefattach <- set.DAG(D, latent.v = c("LatentW"), n.test = 200)
```

We now call function `sim` to simulate a single network of 5,000 individuals using the above defined data-generating distribution, as shown below. We also look at the distribution of node connectivity in Figure **X.X** and we plot this network for a small sample of 50 observations in Figure **X.X**

```
datO_5K <- sim(D.prefattach, n = 5000)
```

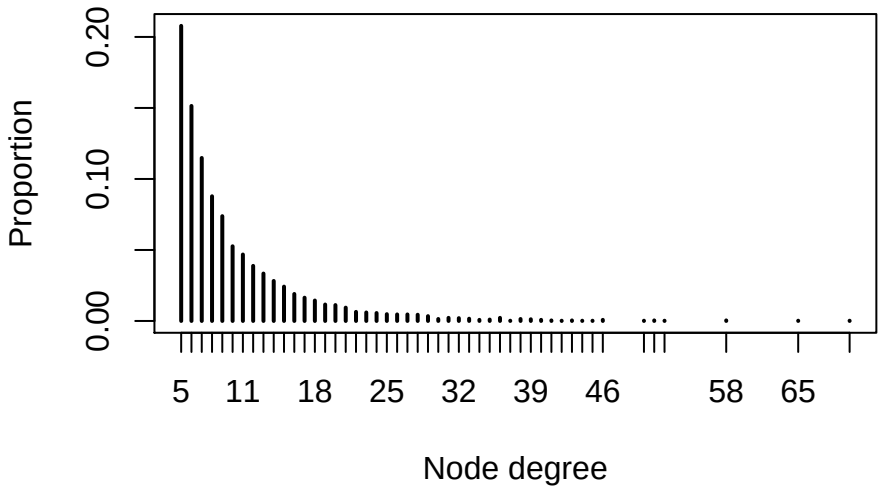

Figure 1: Degree distribution for a preferential attachment network with 5,000 observations.

```
## Warning:  package 'bindrcpp' was built under R version 3.3.2
```

1.1 Evaluating Target Causal Quantities from Simulated Counterfactual Data

Next we define several stochastic and dynamic interventions on the exposure `A`, as well as the total number of physically active friends `nF.PA`. We also calculate the corresponding causal effects of these interventions, using the preferential attachment network model. Note that one can easily evaluate the true values of the above causal parameters by simulating intervention-specific counterfactual data and then evaluating the estimated mean of the counterfactual outcomes, as shown in all of the following examples.

Mean causal outcome under 35% random coverage:

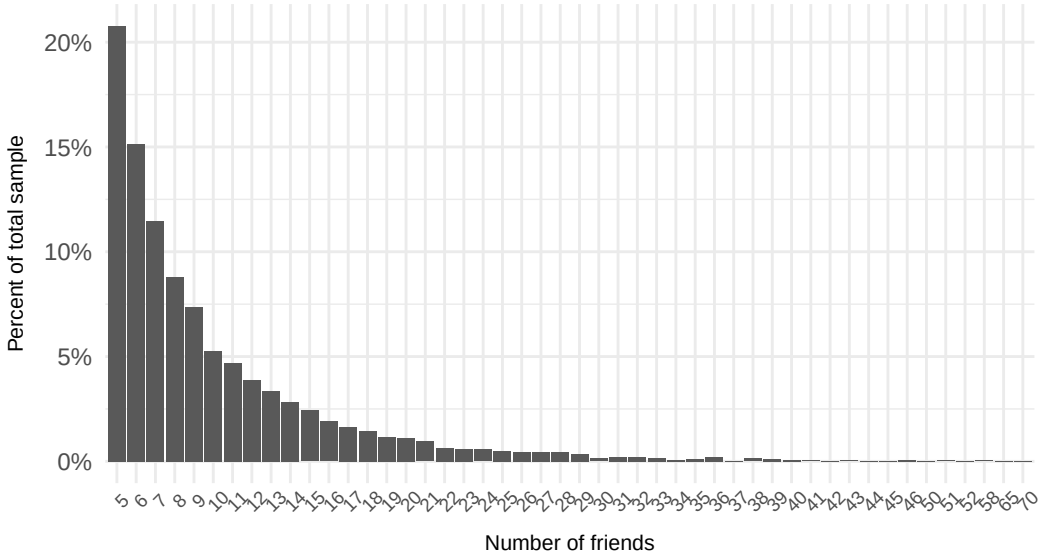

Figure 2: Degree distribution for a preferential attachment network with 5,000 observations.

```
D.prefattach <- D.prefattach +
  action("gstar", nodes = node("A", distr = "rbern", prob = aset), aset = 0.35)
datFull <- sim(D.prefattach, actions="gstar", n = 50000, rndseed = 54321)
print(psi0_a0.4 <- mean(datFull[["gstar"]]$Y))

## [1] 0.15186
```

Dynamic intervention that covers only around 10% of the population by intervening (stochastically) only on the most connected individuals:

```
D.prefattach <- D.prefattach +
action("gHubs",
  nodes = c(node("A", distr = "rbern", prob = ifelse(nF >= 20, 0.9, ifelse(nF >= 15, 0.40, 0)))))
datFull <- sim(D.prefattach, actions="gHubs", n = 50000, rndseed = 54321)
print(psi0_g.dynamic <- mean(datFull[["gHubs"]]$Y))

## [1] 0.1204
```

Network intervention that increases the number of physically active friends by 1:

```
D.prefattach <- D.prefattach +
  action("plus.nF.PA",
    nodes = node("nF.PA", distr = "rconst",
      const = ifelse(nF <= 15, sum(PA[[1:Kmax]]) + 1,
                     sum(PA[[1:Kmax]])),
```

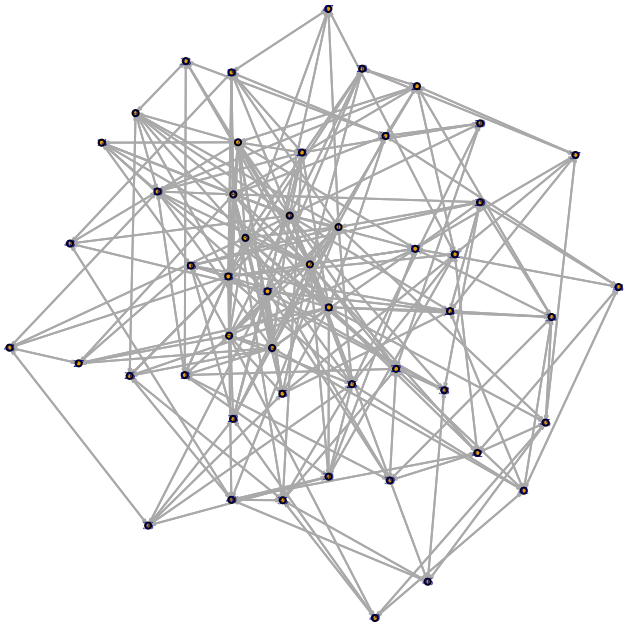

Figure 3: Example of a preferential attachment network for 50 observations.

```
      replaceNAw0 = TRUE))
datFull <- sim(D.prefattach, actions="plus.nF.PA", n = 50000, rndseed = 54321)
print(psi0_plusnF.PA <- mean(datFull[["plus.nF.PA"]]$Y))

## [1] 0.17822
```

2 Estimation of causal effects for network dependent data

Having defined the simulated network data, as well as the true value of the target causal quantity (the gold standard), we switch to the topic of using R for estimating such causal parameters.

Network-based summary measures Example below.

```
require("tmlenet")

sW <- def_sW(W1, W2, W3) +
  def_sW(W1.W2 = W1 * W2) +
  def_sW(mW1.W2 = (1 - W1) * (1 - W2)) +
  def_sW(W1.W3 = W1 * W3) +
```

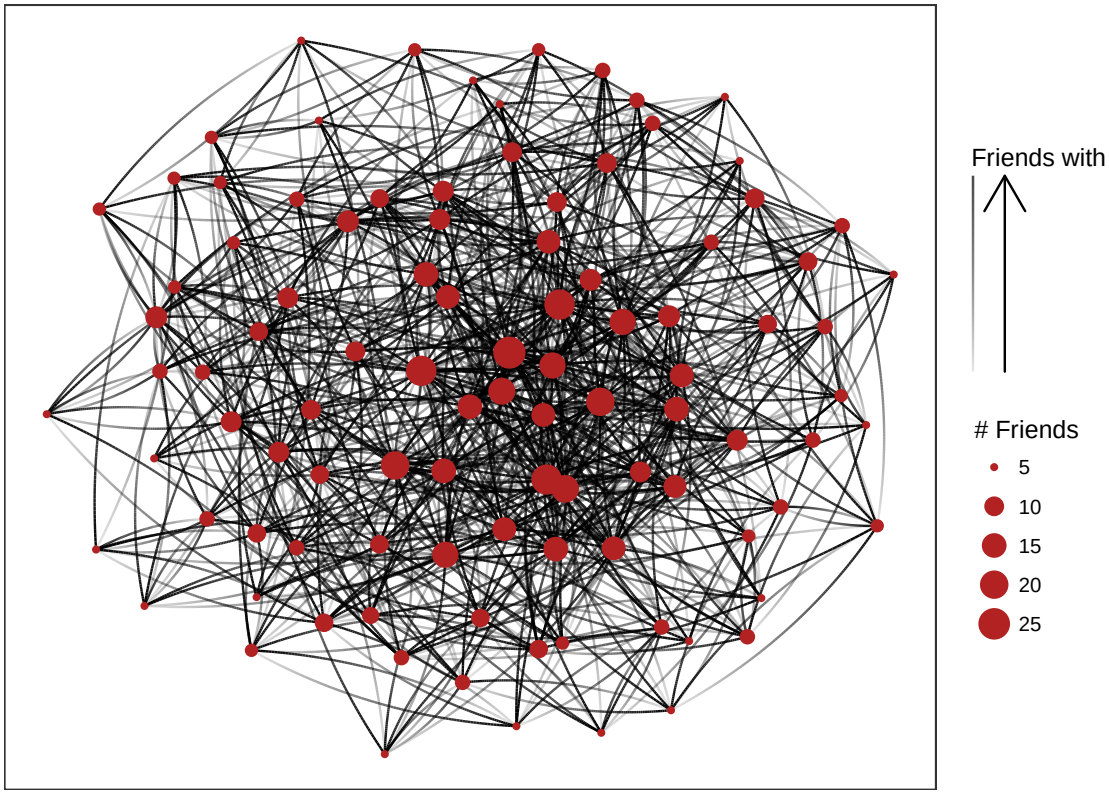

Figure 4: Alternative representation of the same preferential attachment network for 100 observations.

```
  def_sW(mW1.W3 = (1 - W1) * (1 - W3)) +
  def_sW(W2.W3 = W2 * W3) +
  def_sW(mW2.W3 = (1 - W2) * (1 - W3)) +
  def_sW(net.mean.W1 = ifelse(nF > 0, rowSums(W1[[1:Kmax]])/nF, 0), replaceNAw0 = TRUE)

sA <- def_sA(sA, net.mean.sA = sum(sA[[1:Kmax]])/nF, replaceNAw0 = TRUE)
```

Using R to define the summary measures with the **tmlenet** package.

```
sW <-  def_sW(W1, W2, WNoise, corrW.F1, corrW.F2, corrW.F3, corrW.F4, corrW.F5,
  HUB = ifelse(nF >= 25, 1, 0))

## Some summary measures were not named, automatic column name(s) will be generated during evaluation

sA <-  def_sA(A, nF.PA = sum(PA[[1:Kmax]]), replaceNAw0 = TRUE) +
 def_sA(A.PAeq0 = A * (PA == 0)) +
```

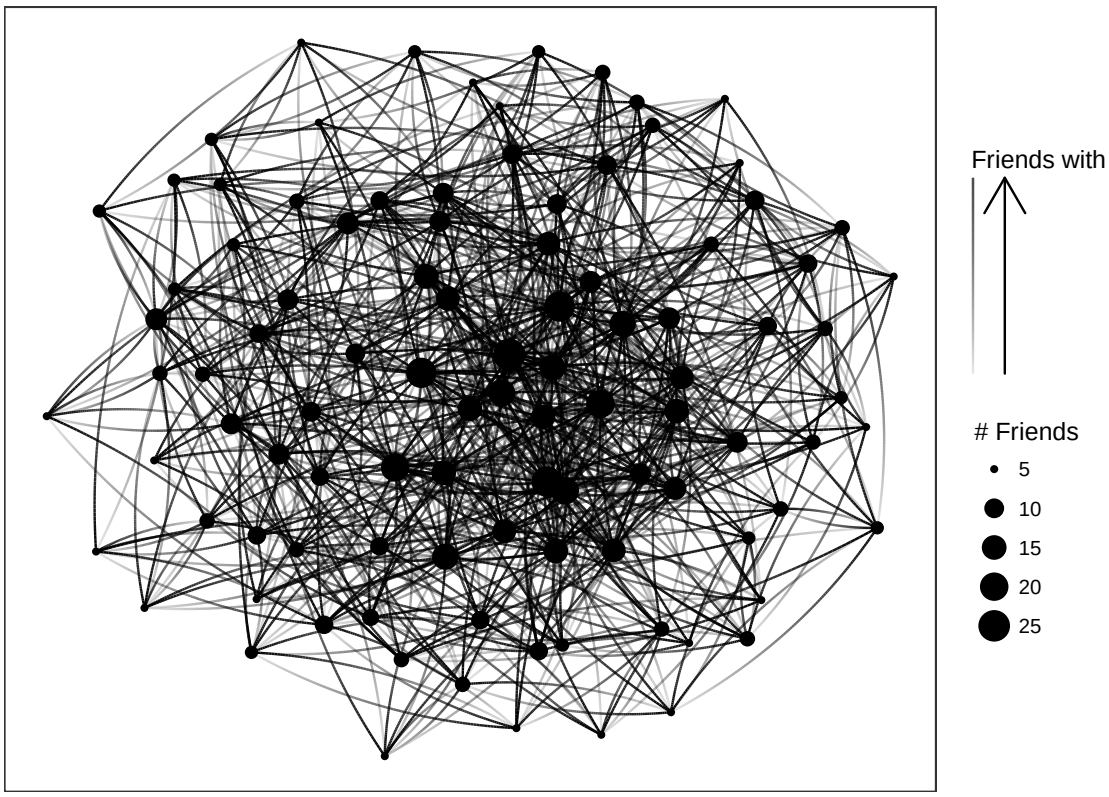

Figure 5: Alternative representation of the same preferential attachment network for 100 observations.

```
def_sA(nFPAeq0.PAeq1 = (nF.PA < 1) * (PA == 1)) +
def_sA(sum.net.A = (sum(A[[1:Kmax]])*(HUB==0) + sum((W1[[1:Kmax]] > 4)*A[[1:Kmax]])*(HUB==1)),
       sum.net.A.sum.netPA = sum.net.A*nF.PA,
       replaceNAw0 = TRUE)

## Some summary measures were not named, automatic column name(s) will be generated during evaluation
```

Regression models Examples of model specifications for the outcome and the *effective exposure* models.

```
Qforms <- "Y ~ nF.PA + A.PAeq0 + nFPAeq0.PAeq1 + sum.net.A + sum.net.A.sum.netPA + PA + W1 + W2 + corrW.F1 + corrW.F2 + cor
hform <- "A + sum.net.A ~ HUB + PA + nF.PA + nFPAeq0.PAeq1"
```

Interventions Examples of interventions on summary measures.

1. All intervention nodes must be named and must match some previously defined summary measure/node name (been previously defined in `def_sA`).

2. The interention nodes/summaries will replace the existing ones.

3. The summaries that were part of `def_sA` and were not re-defined in `def_sA.gstar` will be still re-evaluated on the data generated under `def_sA.gstar`.

4. Each intervention nodes/summary can reference the value of its own previously defined node, evaluated under observed data. For example, if we had an observed binary exposure data column A (registered with `def_sA(A)`), the intervention that reverses the value of A from 0 to 1 and from 1 to 0 could be simply defined as `def_sA.gstar(A = 1 - A)`.

5. All of the observed exposure summaries defined in obs.sW.sA are evaluated in EXACTLY the same order as they were defined. Hence all the intervention summaries preserve exactly the same order of evaluation as in `obs.sW.sA`.

```
# Example 1A: Increase the total number of phys-active friends by 1.
sA_star1a <- def_new_sA(nF.PA = (nF <= 10)*(sum(PA[[1:Kmax]])+1) + (nF > 10)*sum(PA[[1:Kmax]]), replaceNAw0 = TRUE)
# # Example 1B: Alternative way of defining exactly the same intervention (will over-ride the existing summary nF.PA define
sA_star1b <- def_new_sA(nF.PA = (nF <= 10)*(nF.PA+1) + (nF > 10)*nF.PA)
# Example 2: Sample A as a stochastic intervention:
sA_star2 <-  def_new_sA(A = rbinom(n = length(A), size = 1, prob = 0.10))
# Example 3A: Sample A as a stochastic intervention and don't intervene on the summaries of the HUBS:
sA_star3a <- def_new_sA(A = rbinom(n = length(A), size = 1, prob = 0.60)) +
  def_new_sA(sum.net.A = ifelse(HUB==1, sum.net.A, sum(A[[1:Kmax]])), replaceNAw0 = TRUE)
# Example 3B: Equivalent to 3A, but explicititely defining sum.net.A.sum.netPA as well:
sA_star3b <- def_new_sA(A = rbinom(n = length(A), size = 1, prob = 0.10)) +
  def_new_sA(sum.net.A = ifelse(HUB==1, sum.net.A, sum(A[[1:Kmax]])), replaceNAw0 = TRUE) +
  def_new_sA(sum.net.A.sum.netPA = sum.net.A*nF.PA)
```

Examples of interventions on the exposure variable. Stochastic intervention on A. Note that the the rest of the effective exposure summary measures are automatically evaluated based on the counterfactual values of the exposure variable specified below.

```
new.sA1.stoch.2 <-  def_new_sA(A = rbinom(n = length(A), size = 1, prob = 0.35))
```

Example of a dynamic intervention on A, conditional on the number of friends (`nF`). This intervention assigns the exposure to approximately top 10% of the most connected individuals in the observed network.

```
new.sA1.dyn.4 <- def_new_sA(A = rbinom(n = length(A), size = 1,
  prob = ifelse(nF >= 20, 0.9, ifelse(nF >= 15, 0.40, 0))))
```

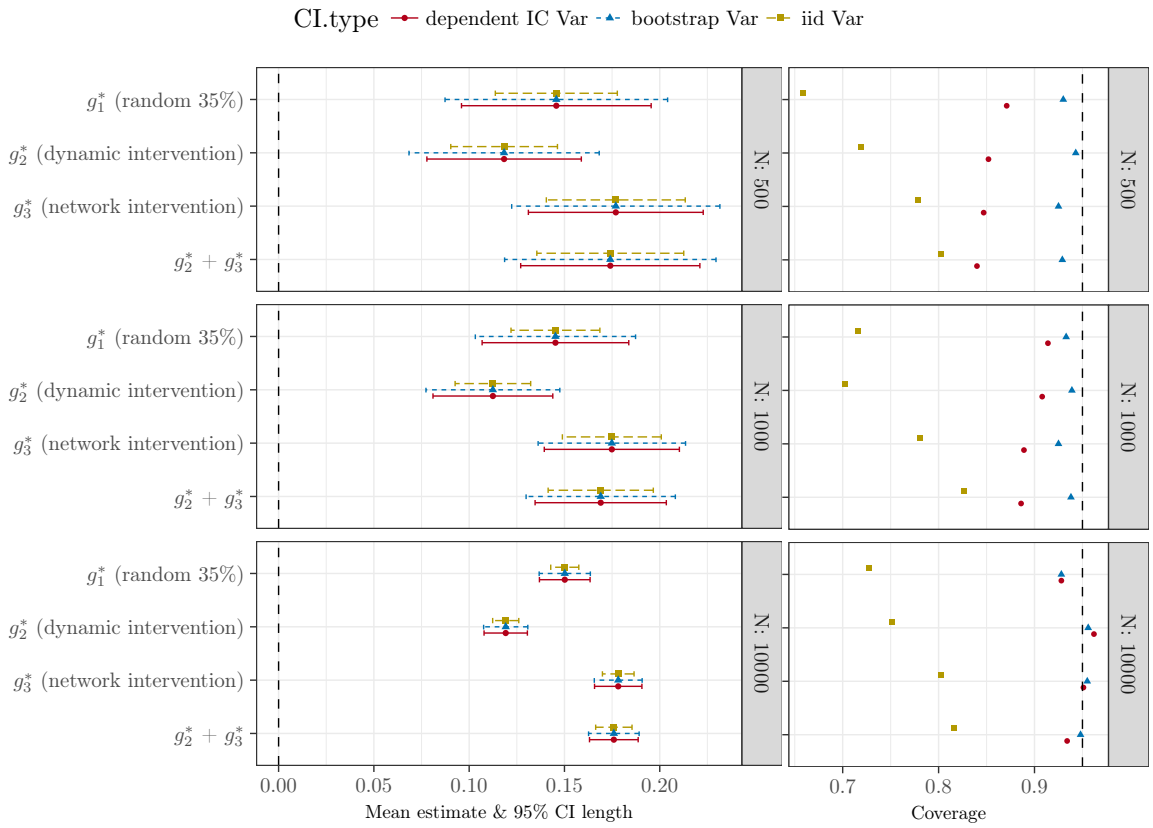

Figure 6: Mean 95% CI length (left panel) and coverage (right panel) for the preferential attachment network, by sample size, interevention and CI type. Results shown for average expected outcomes only.

Example of running the

```
res <- tmlenet(data = datO, sW = sW, sA = sA,
  Ynode = "Y", Kmax = K,
  NETIDmat = attributes(datO)$netind_cl$NetInd,
  intervene1.sA = new.sA1.stoch.2,
  Qform = Qform, hform.g0 = hform, hform.gstar = hform)
```

3 Main results

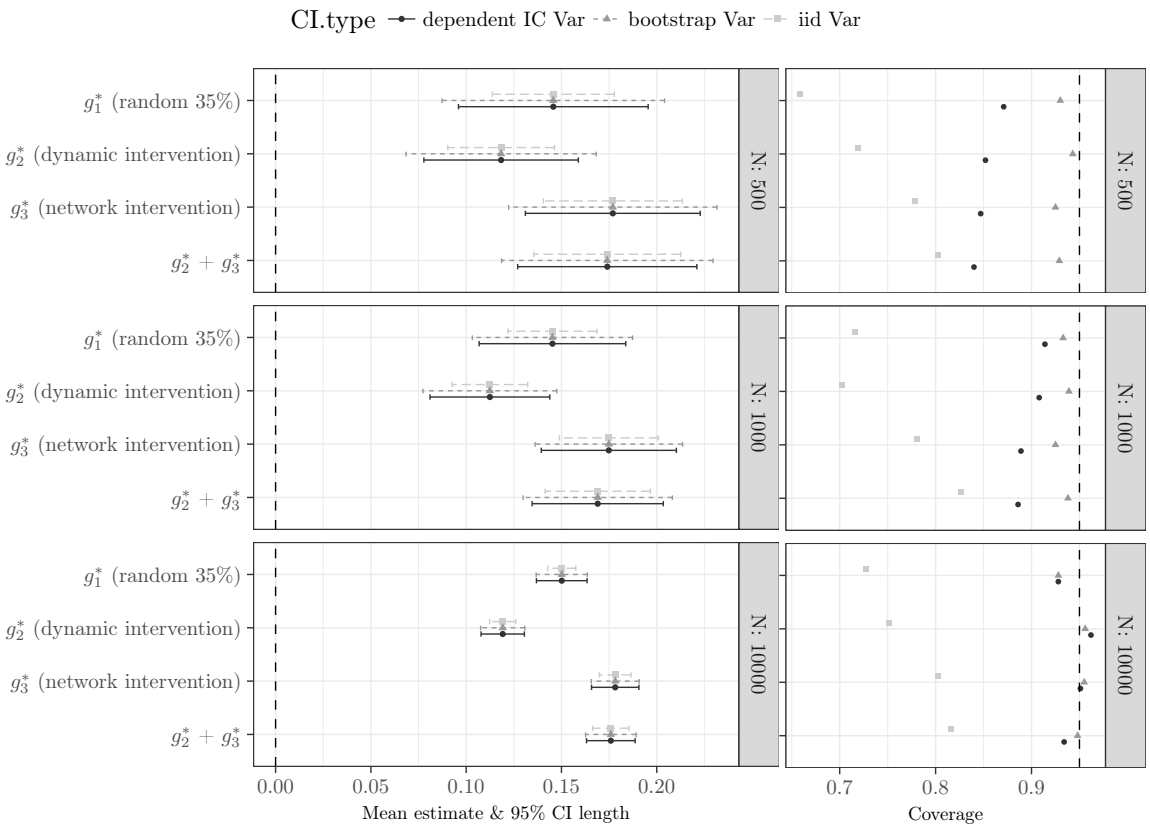

Figure 7: Mean 95% CI length (left panel) and coverage (right panel) for the preferential attachment network, by sample size, interevention and CI type. Results shown for average expected outcomes only.

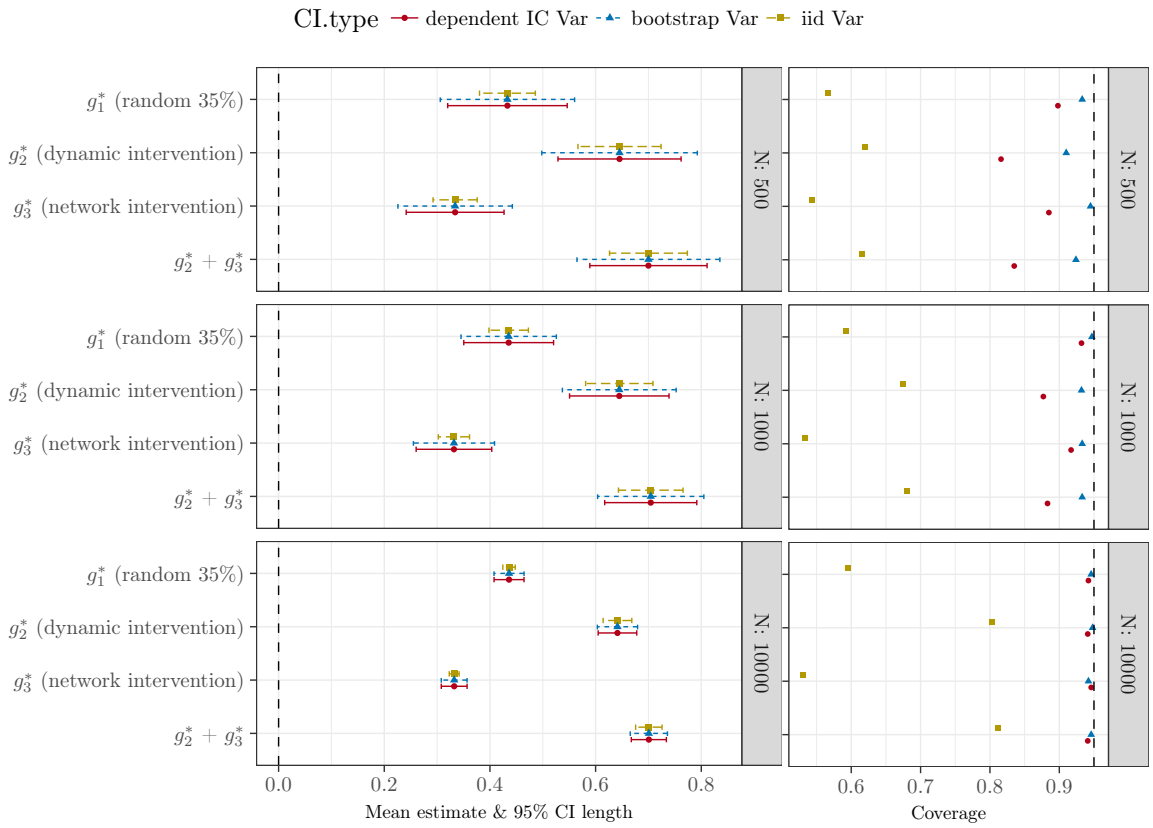

Figure 8: Mean 95% CI length (left panel) and coverage (right panel) for the small world network, by sample size, interevention and CI type. Results shown for average expected outcomes only.

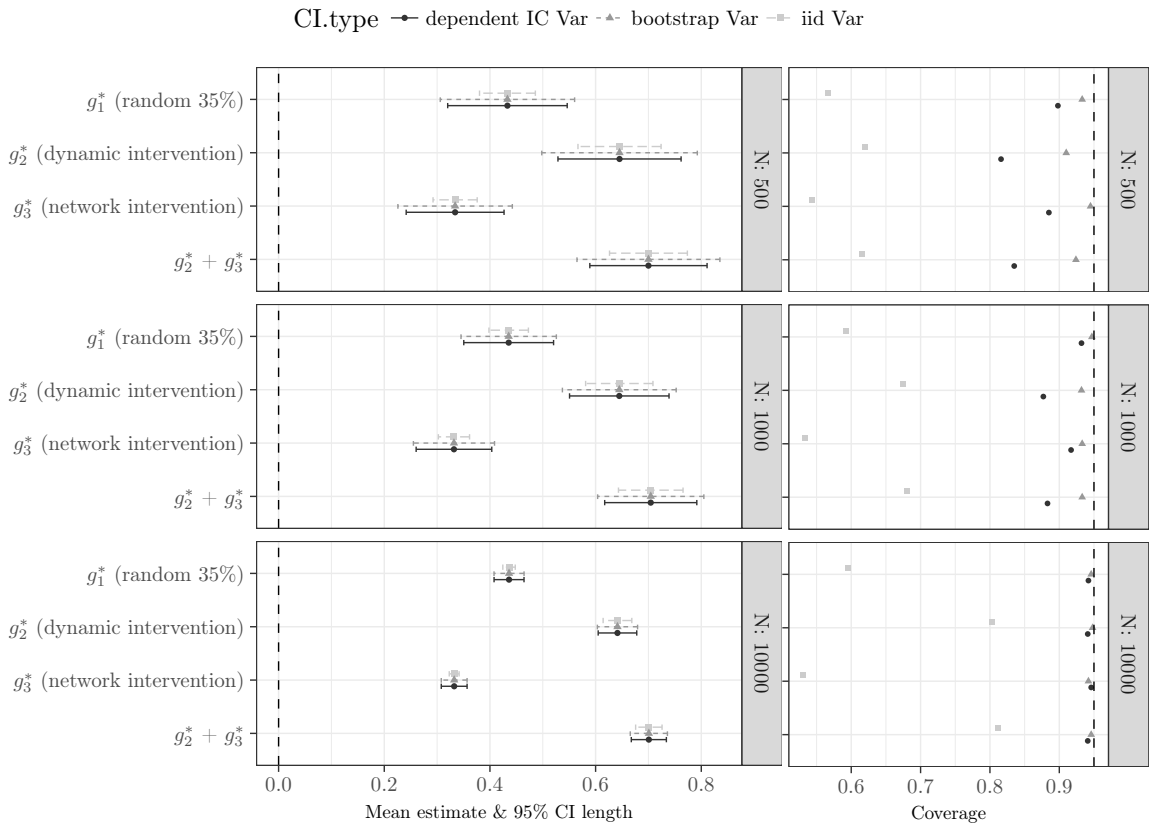

Figure 9: Mean 95% CI length (left panel) and coverage (right panel) for the small world network, by sample size, interevention and CI type. Results shown for average expected outcomes only.

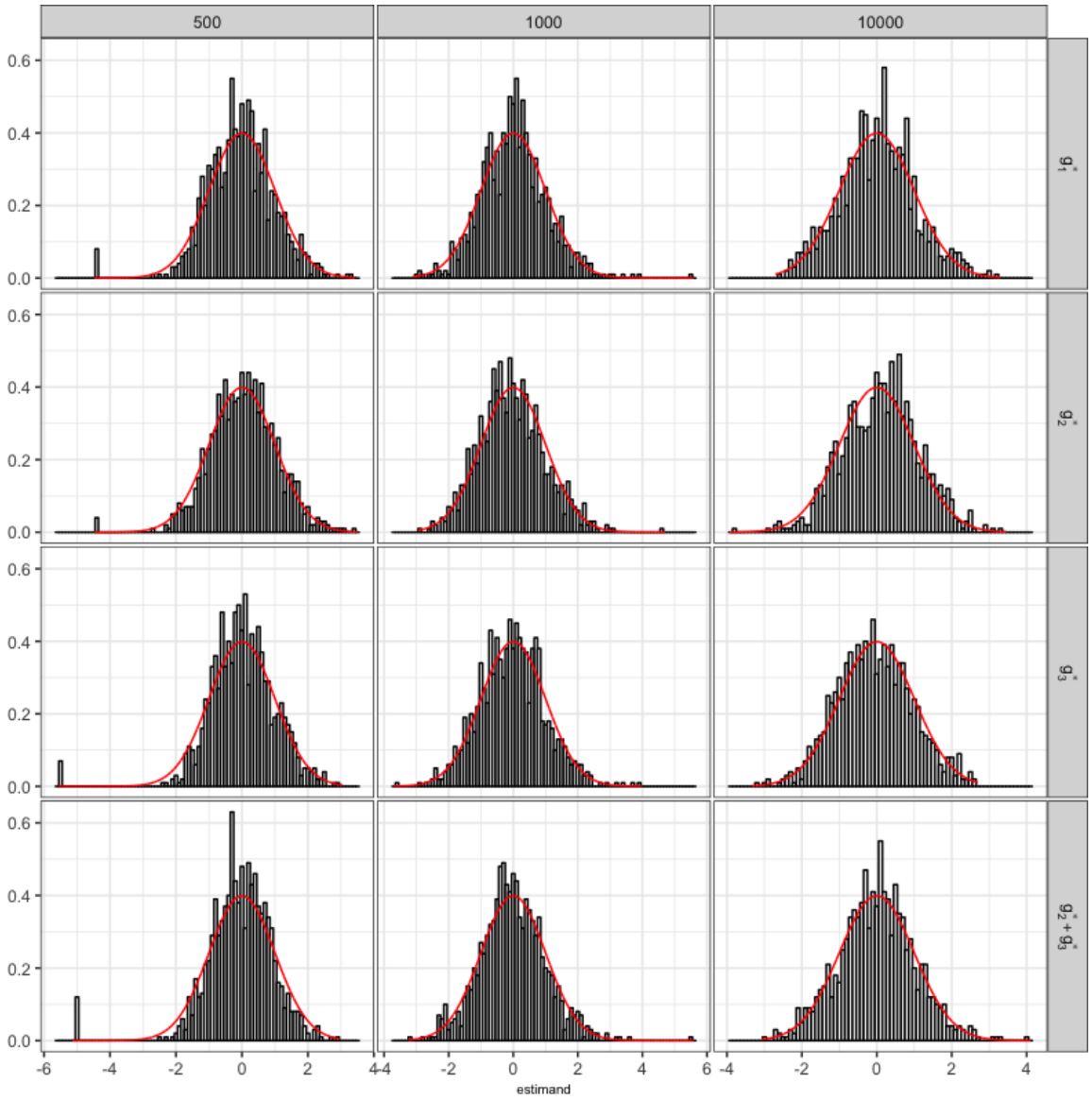

Figure 10: Distribution of the transformed TMLE (black) compared to the theoretical limiting distribution (red) by sample size (x-axis) and intervention type (y-axis). The estimates were centered at the truth and re-scaled by true SD. Results shown are for average expected outcomes in the preferential attachment network.

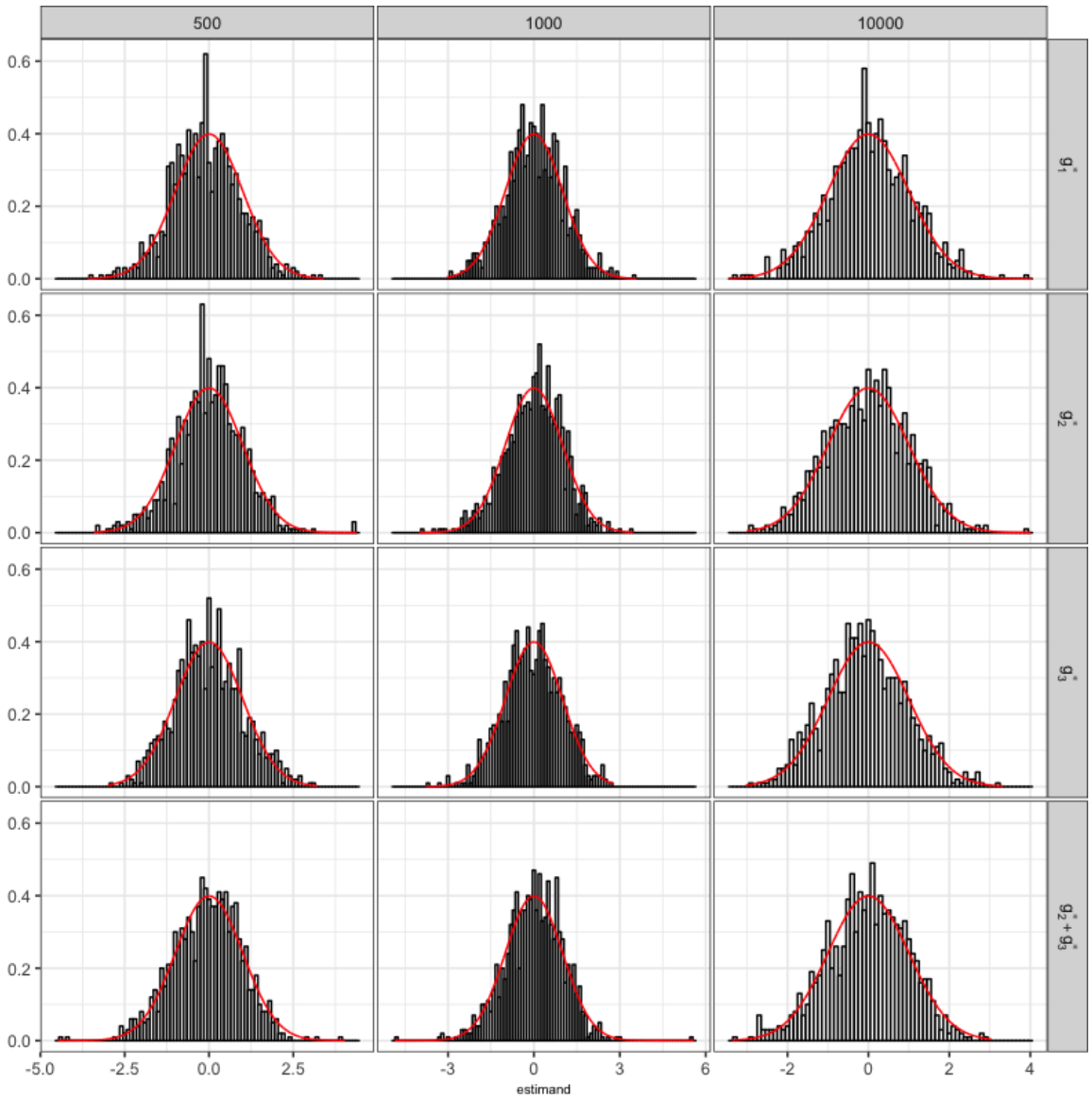

Figure 11: Distribution of the transformed TMLE (black) compared to the theoretical limiting distribution (red) by sample size (x-axis) and intervention type (y-axis). The estimates were centered at the truth and re-scaled by true SD. Results shown are for average expected outcomes in the small world network.

4 Supplementary results

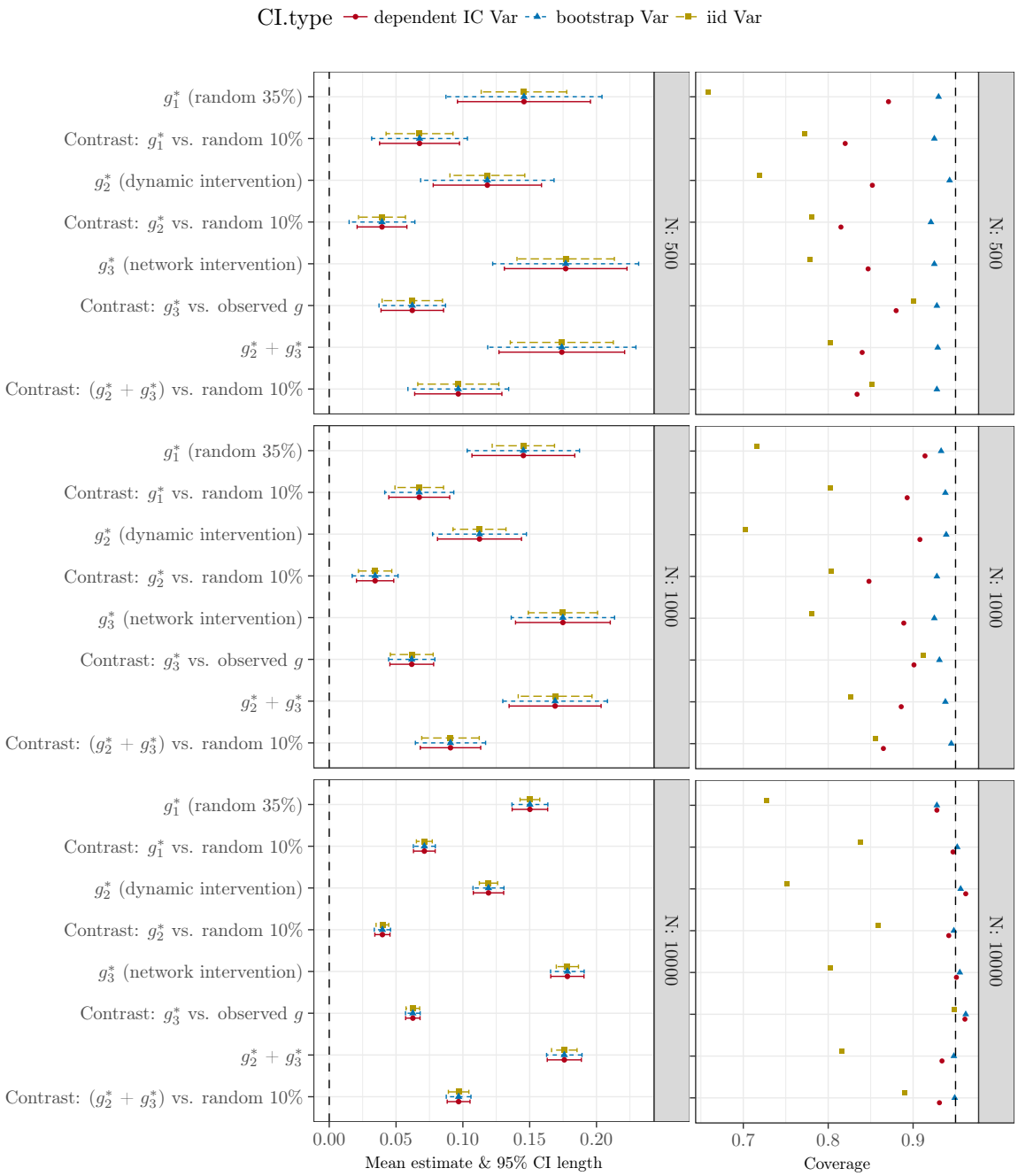

Figure 12: Mean 95% CI length (left panel) and coverage (right panel) for the preferential attachment network, by sample size, interevention and CI type. Results shown for all scenarios.

Figure 13: Mean 95% CI length (left panel) and coverage (right panel) for the small world network, by sample size, interevention and CI type. Results shown for all scenarios.

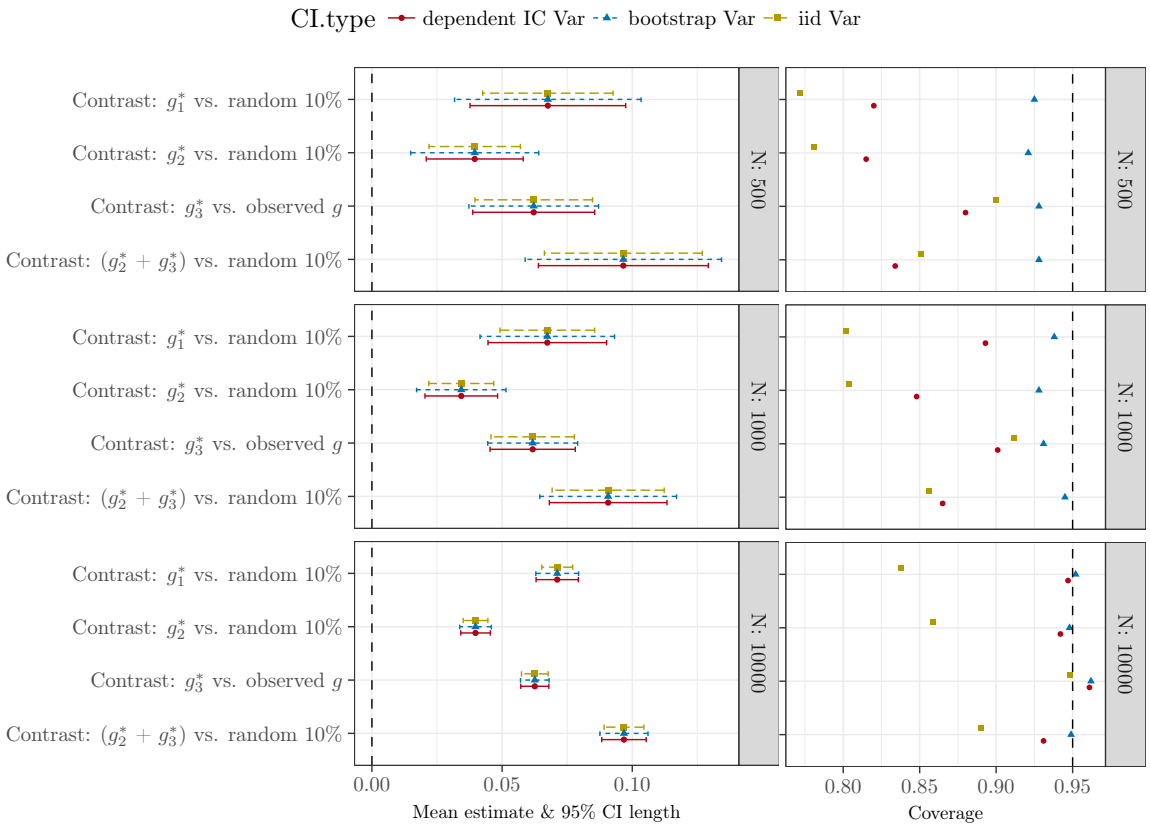

Figure 14: Mean 95% CI length (left panel) and coverage (right panel) for the preferential attachment network, by sample size, interevention and CI type. Results shown for contrasts only.

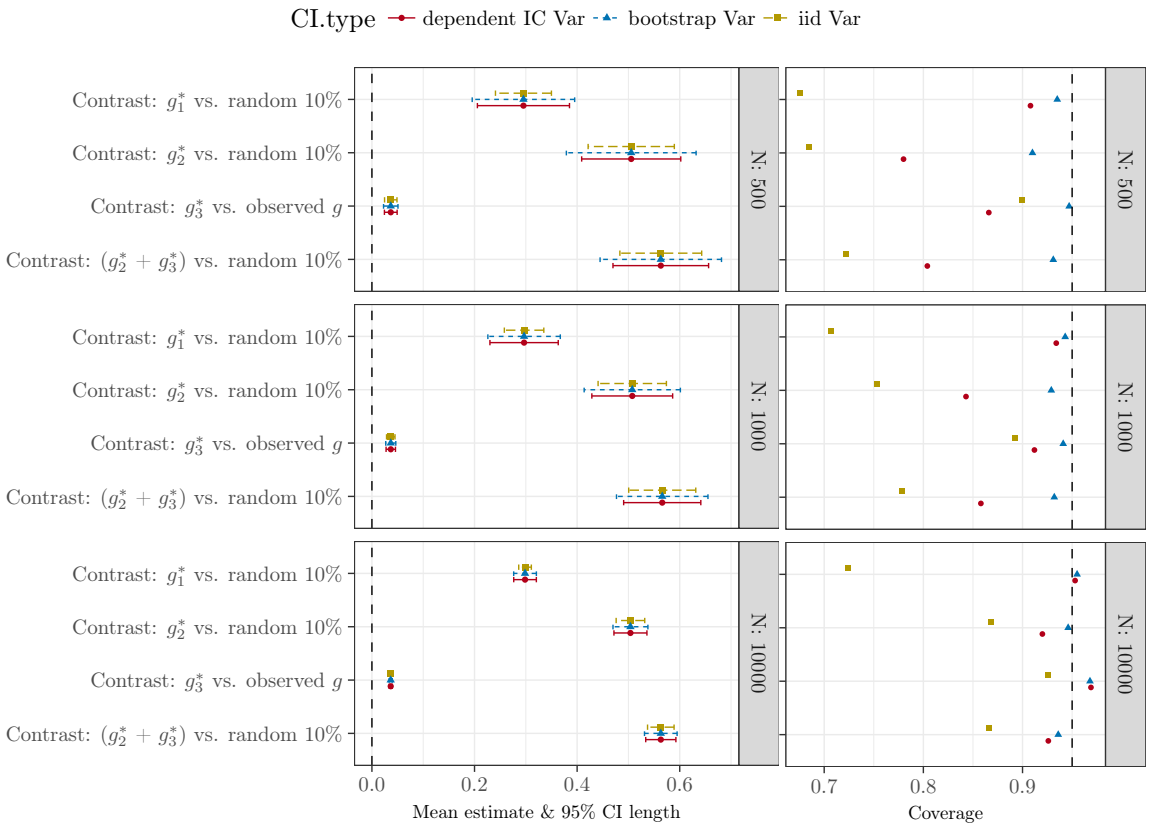

Figure 15: Mean 95% CI length (left panel) and coverage (right panel) for the small world network, by sample size, interevention and CI type. Results shown for contrasts only.

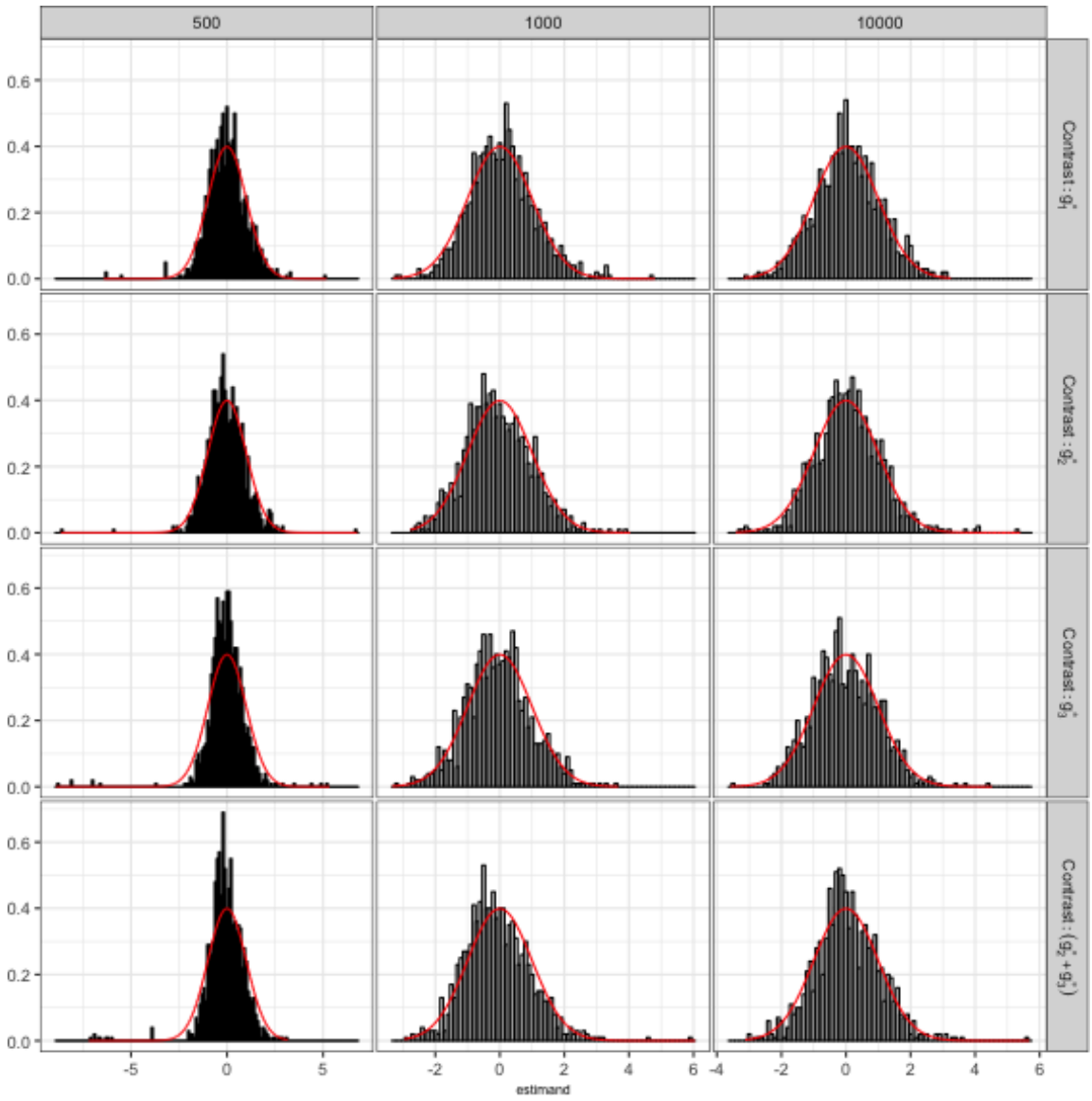

Figure 16: Distribution of the transformed TMLE (black) compared to the theoretical limiting distribution (red) by sample size (x-axis) and intervention type (y-axis). The estimates were centered at the truth and re-scaled by true SD. Results shown are for contrasts in preferential attachment network.

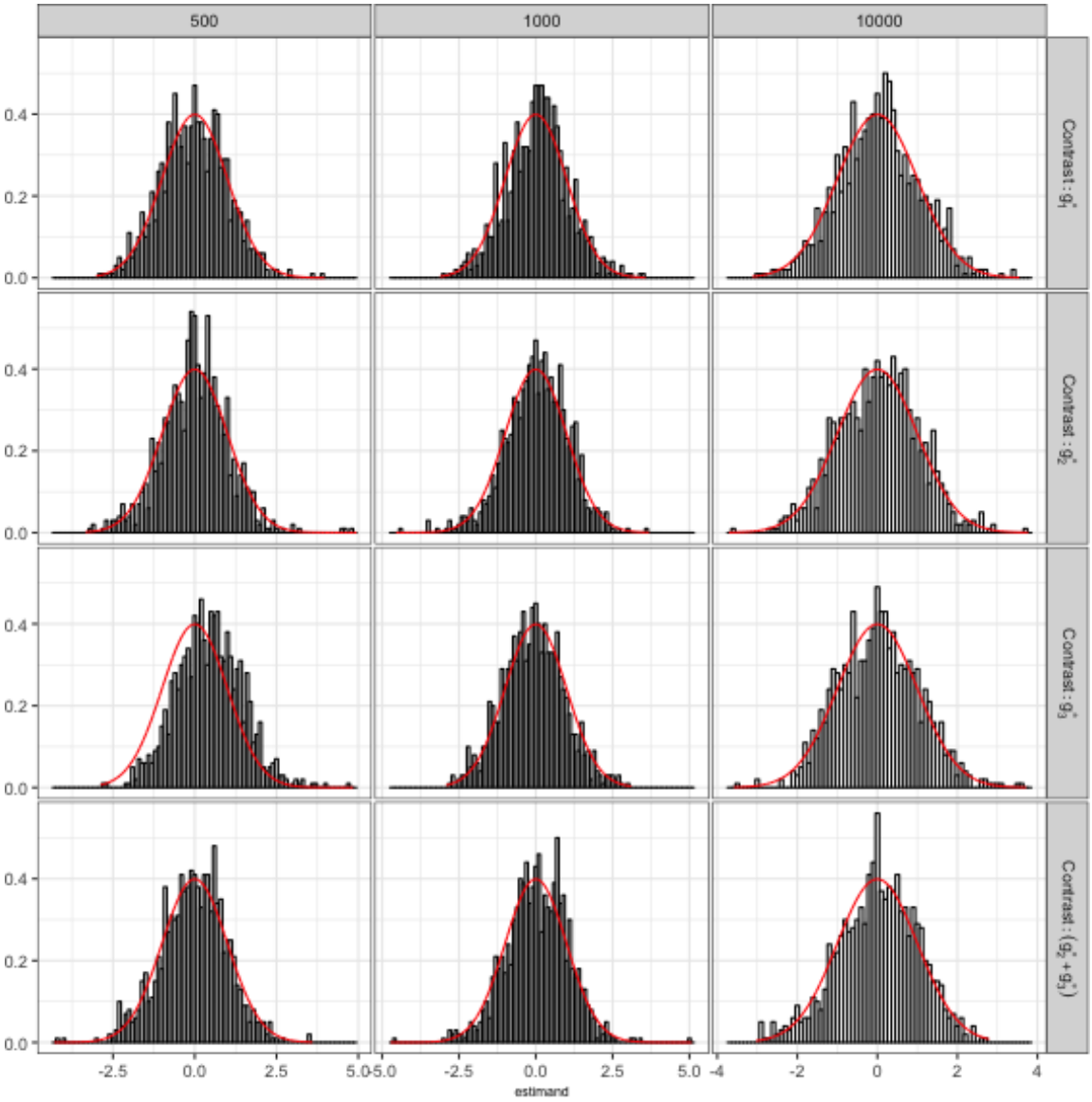

Figure 17: Distribution of the transformed TMLE (black) compared to the theoretical limiting distribution (red) by sample size (x-axis) and intervention type (y-axis). The estimates were centered at the truth and re-scaled by true SD. Results shown are for contrasts in small world network.

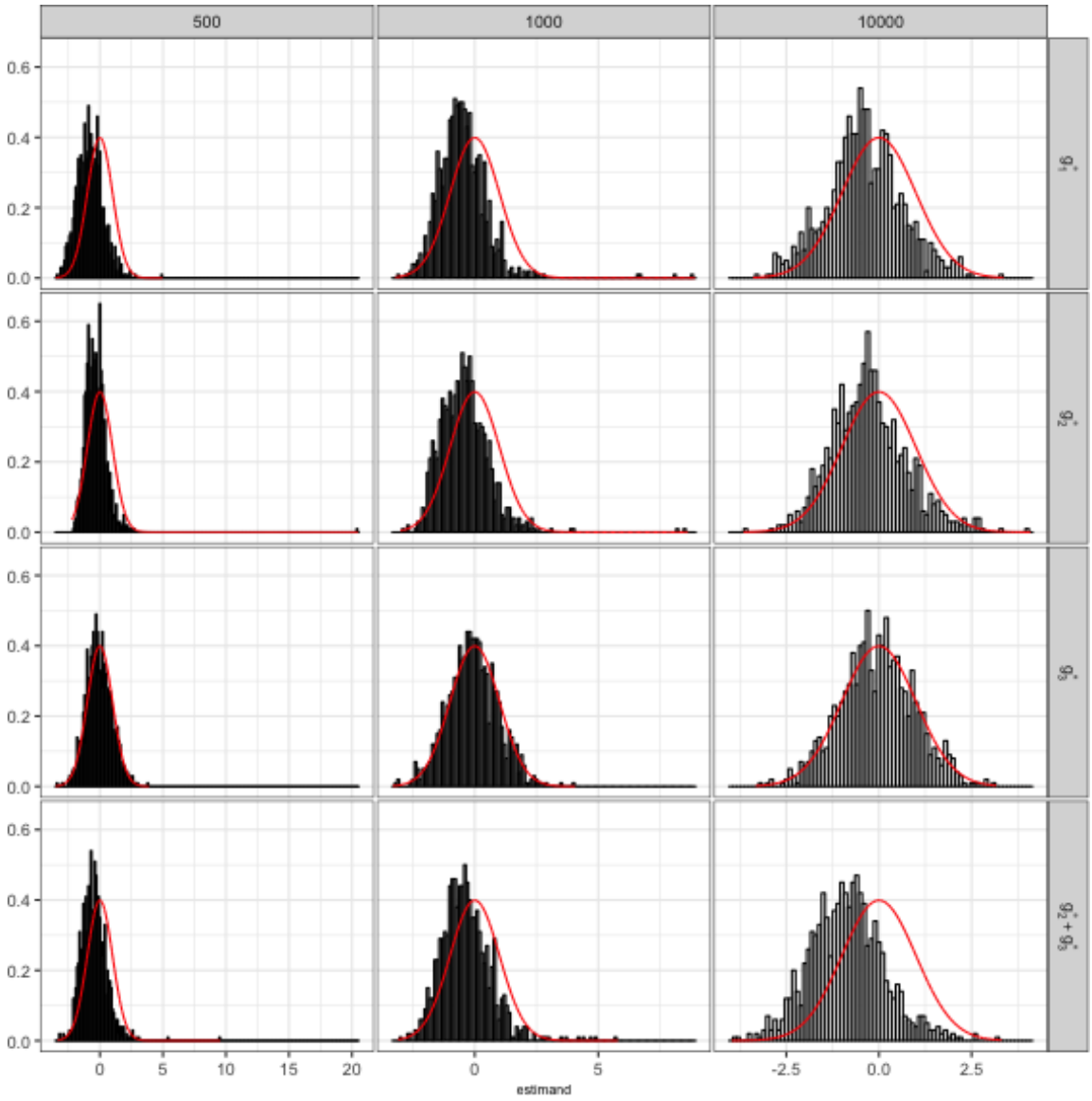

Figure 18: Distribution of the transformed IPTW (black) compared to the theoretical limiting distribution (red) by sample size (x-axis) and intervention type (y-axis). The estimates were centered at the truth and re-scaled by true SD. Results shown are for average expected outcomes in the preferential attachment network.

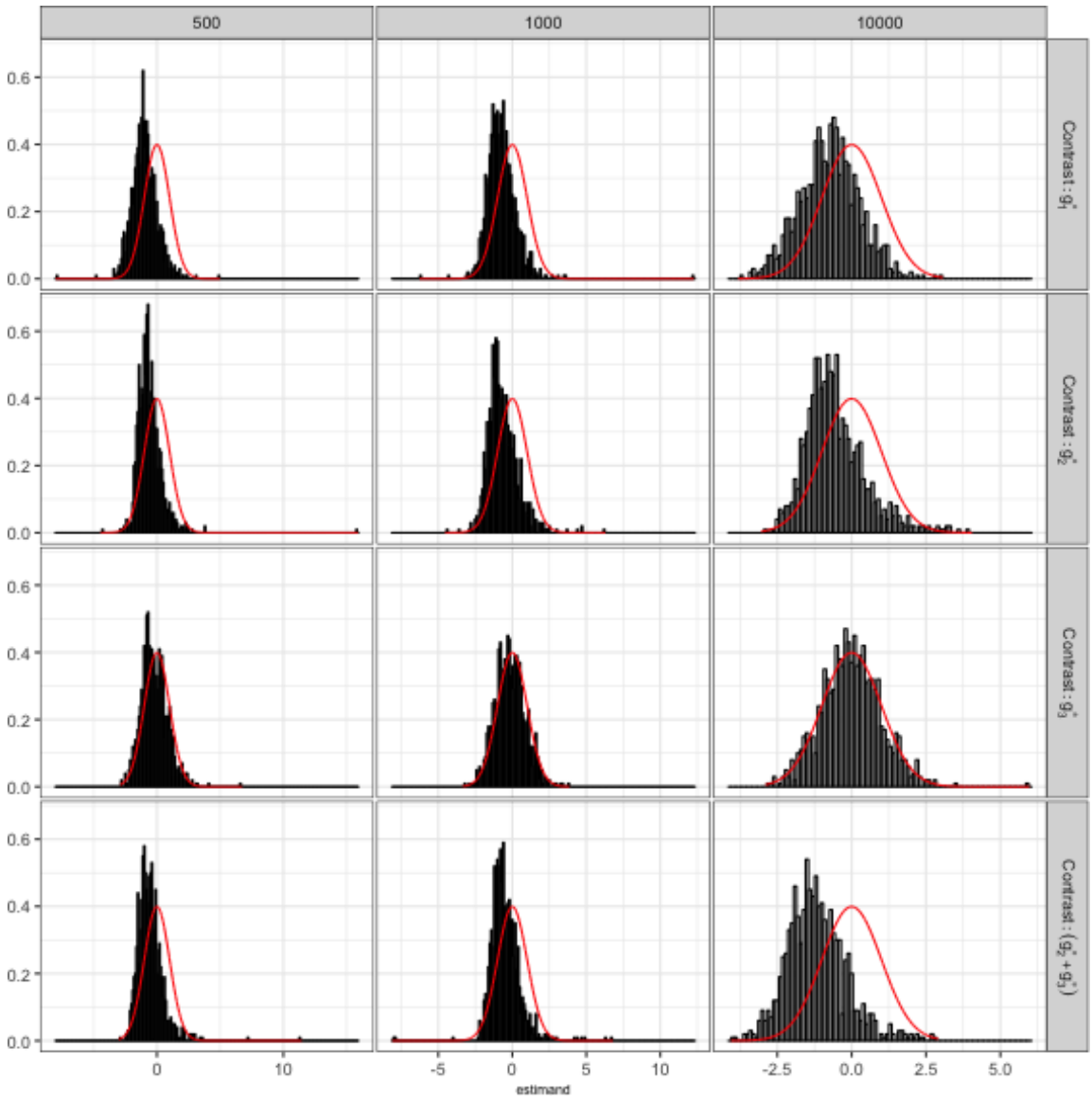

Figure 19: Distribution of the transformed IPTW (black) compared to the theoretical limiting distribution (red) by sample size (x-axis) and intervention type (y-axis). The estimates were centered at the truth and re-scaled by true SD. Results shown are for contrasts in the preferential attachment network.

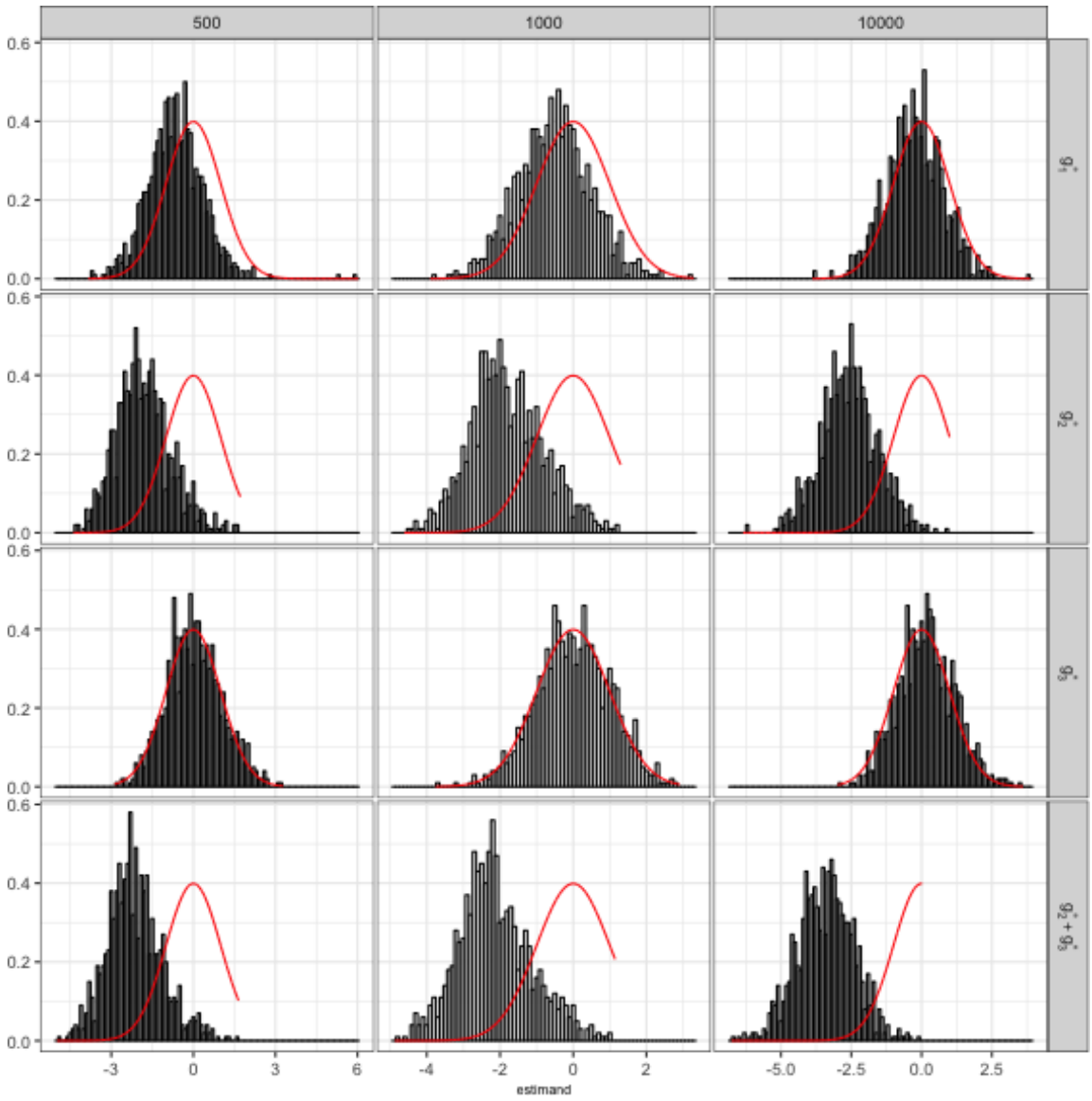

Figure 20: Distribution of the transformed IPTW (black) compared to the theoretical limiting distribution (red) by sample size (x-axis) and intervention type (y-axis). The estimates were centered at the truth and re-scaled by true SD. Results shown are for average expected outcomes in the small world network.

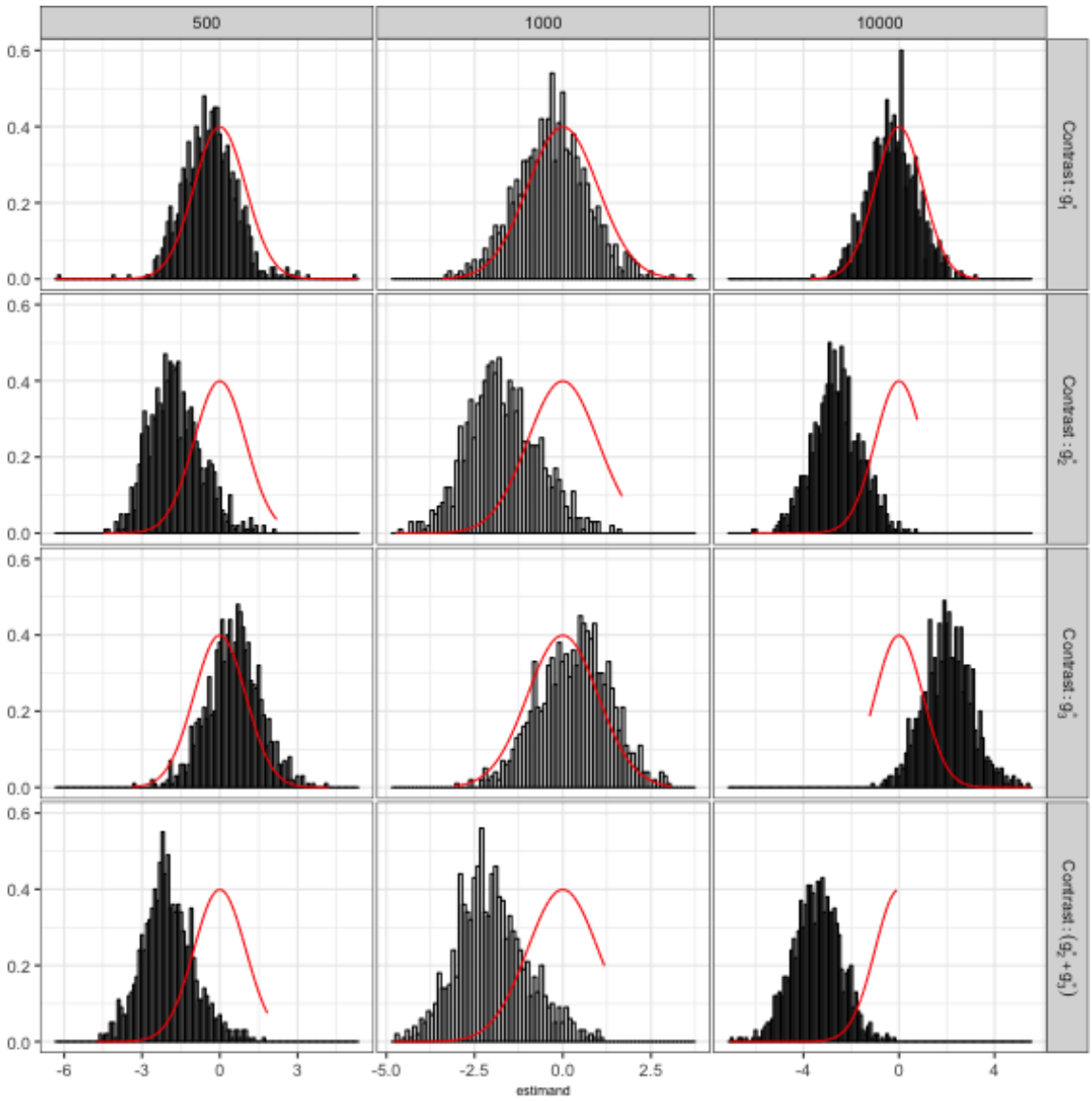

Figure 21: Distribution of the transformed IPTW (black) compared to the theoretical limiting distribution (red) by sample size (x-axis) and intervention type (y-axis). The estimates were centered at the truth and re-scaled by true SD. Results shown are for contrasts in the small world network.