Monte Carlo program for the graphical causal model paper

Wesley Burnett

6/1/2021

Load Libraries

```
library(lavaan)
```

```
## This is lavaan 0.6-8
## lavaan is FREE software! Please report any bugs.
```

```
library(ggplot2)
library(ggthemes)
library(ggdag)
```

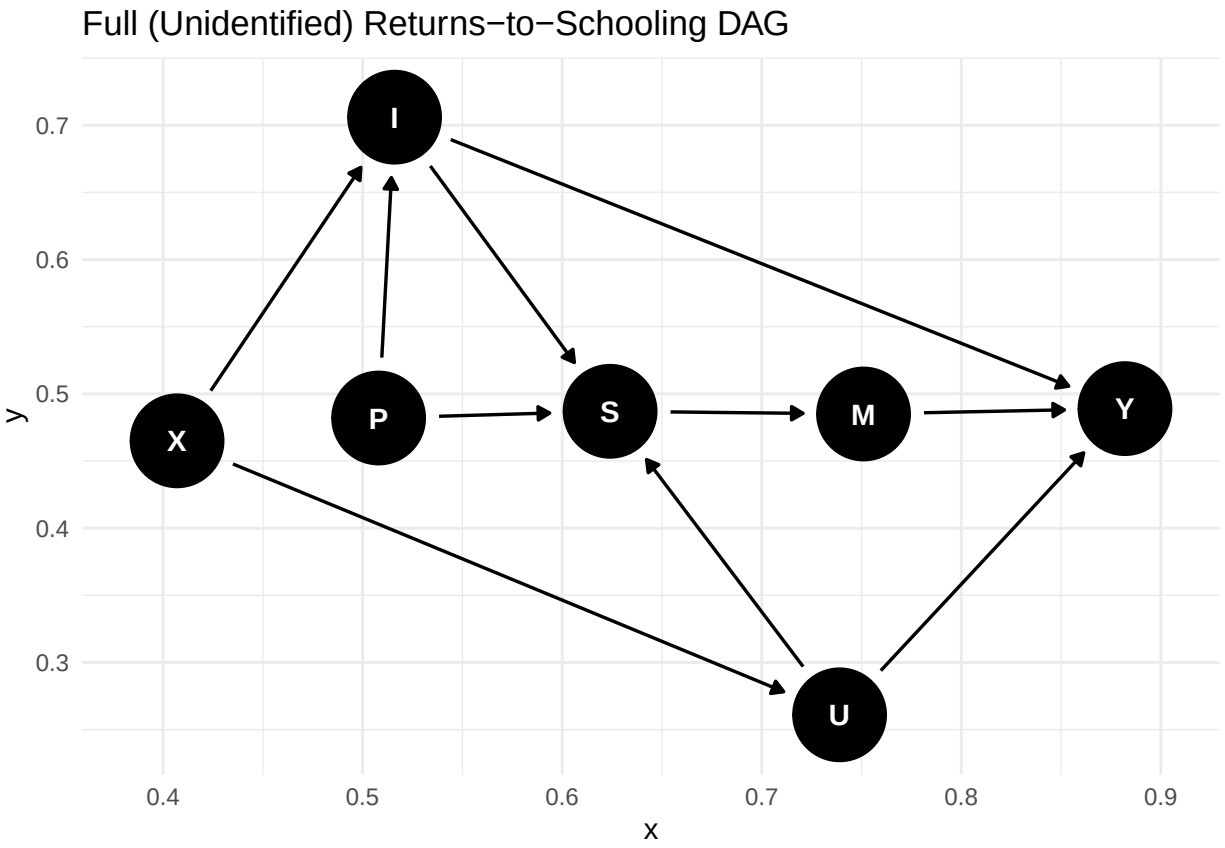
```
##
## Attaching package: 'ggdag'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(dagitty)
theme_set(theme_minimal())
```

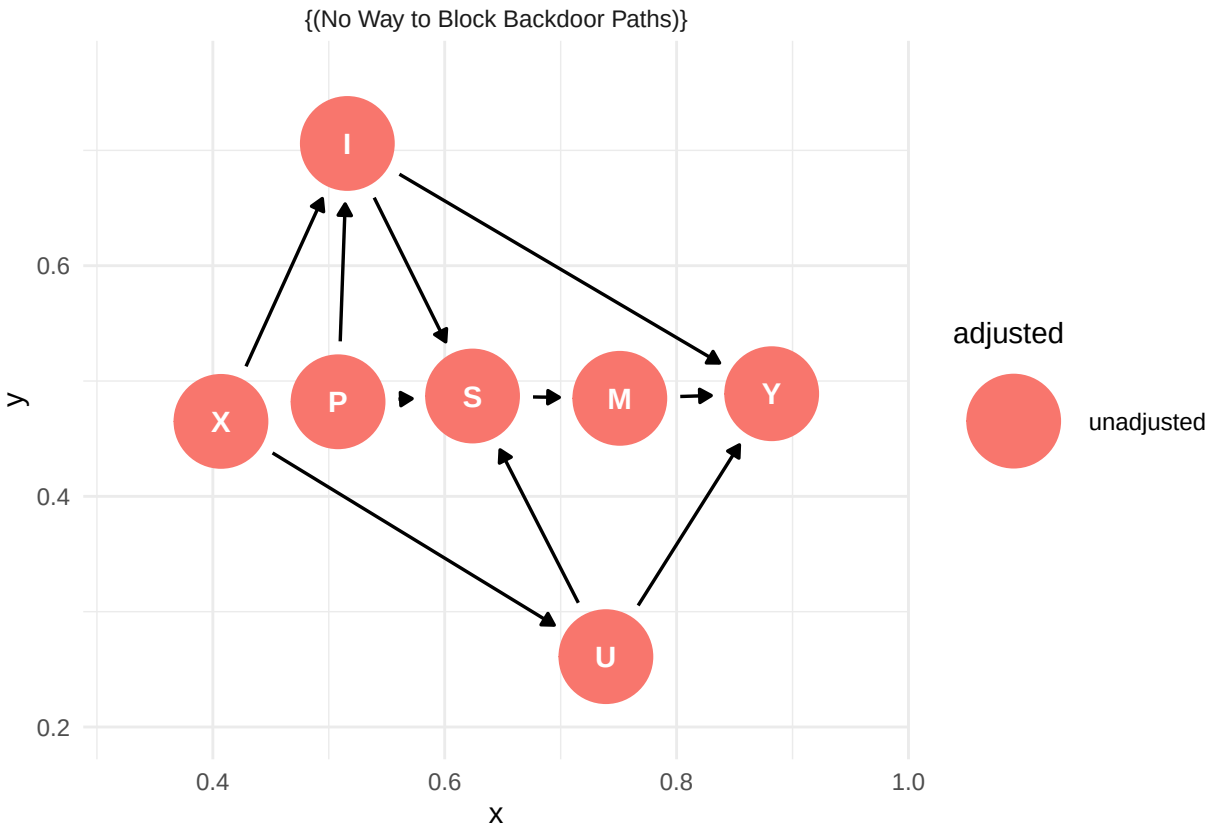
Illustrate the DAG analysis from dagitty

```
# We created the full returns-to-schooling DAG and pre-loaded it to dagitty.net
dag_full <- downloadGraph("dagitty.net/mcq2YLa")
ggdag(dag_full) +
  labs(title = "Full (Unidentified) Returns-to-Schooling DAG")
```

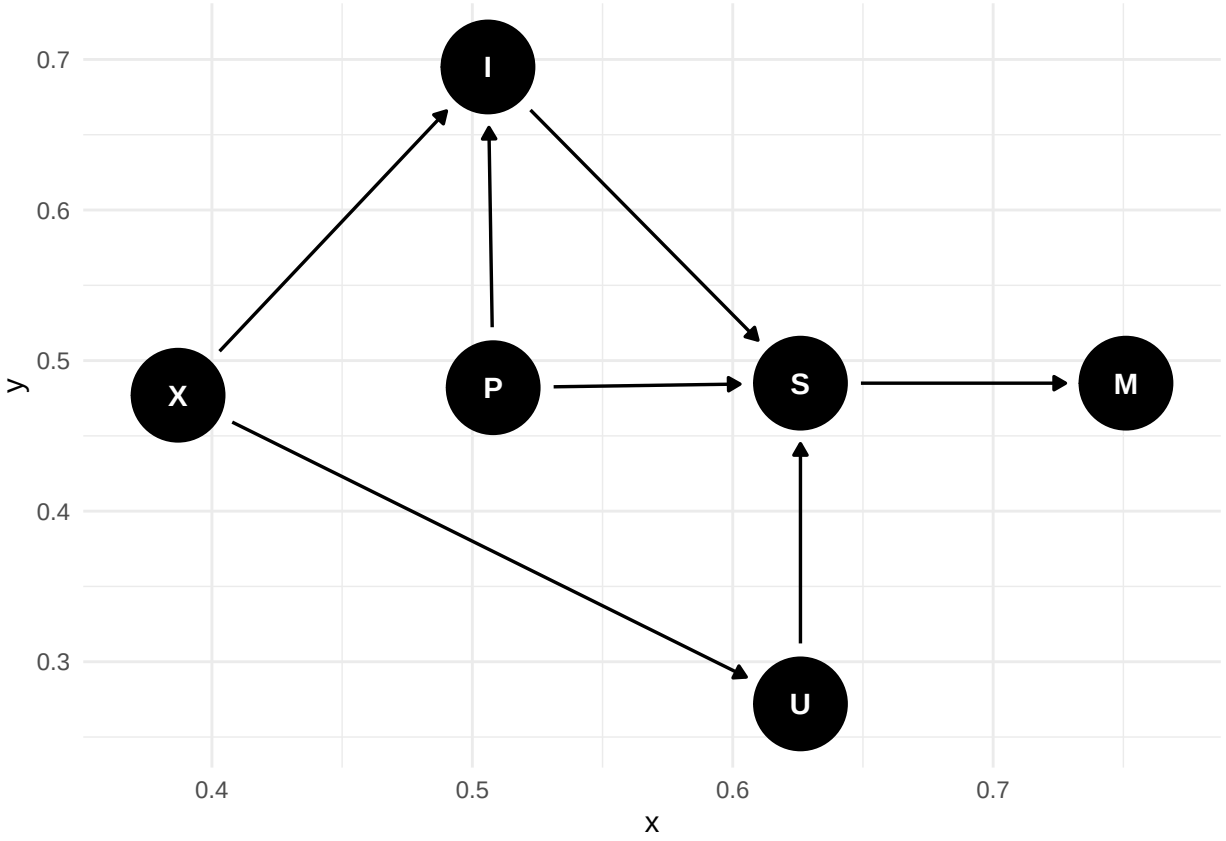



```
# This checks to see if the DAG can be identified based on the graphical identification criteria
ggdag_adjustment_set(dag_full, outcome = "Y", exposure = "S")
```

```
## Warning in dag_adjustment_sets(., exposure = exposure, outcome = outcome, : Failed to close backdoor
##             * graph is not acyclic
##             * backdoor paths are not closeable with given set of variables
##             * necessary variables are unmeasured (latent)
```



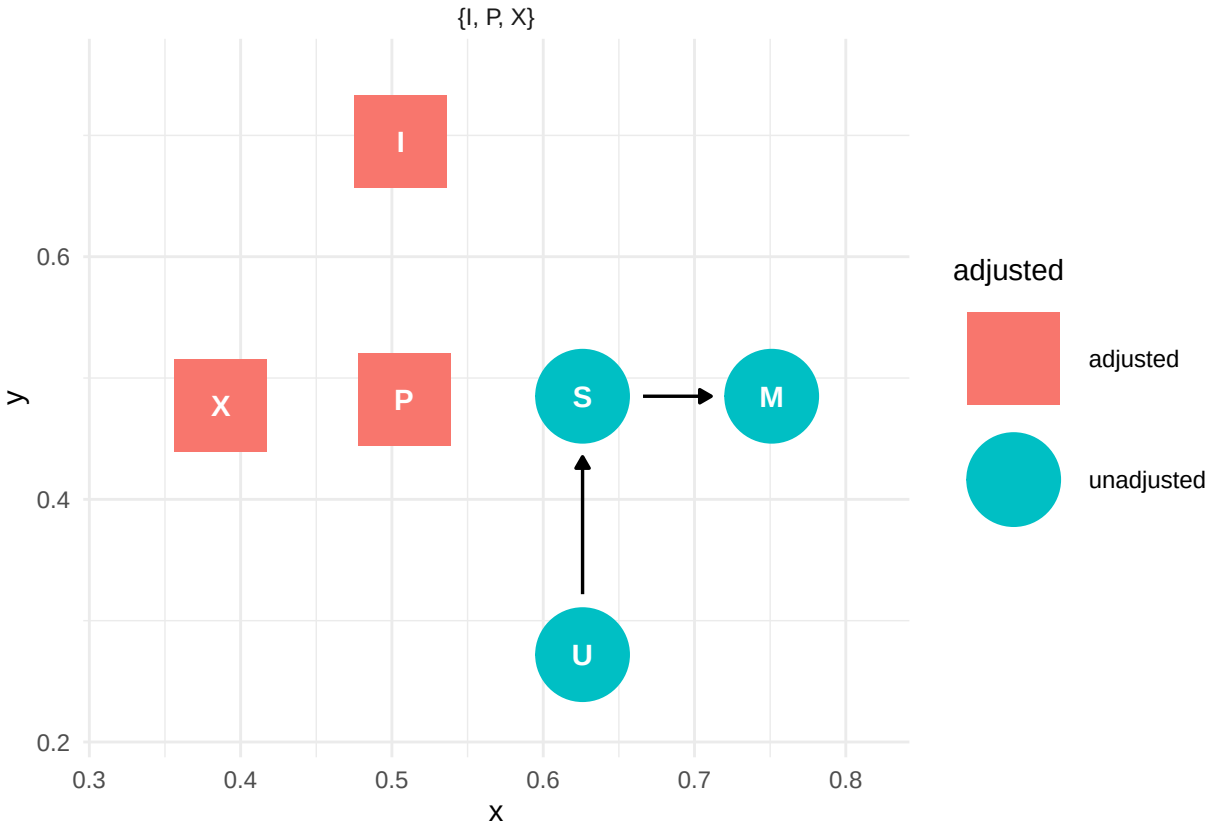

```
# This is a pre-created DAG showing the first-step of the front-door criterion estimation approach.
# Note that the FD criterion is just two back-door criterion for the full model
dag_first <- downloadGraph("dagitty.net/mSId9w-")
ggdag(dag_first)
```



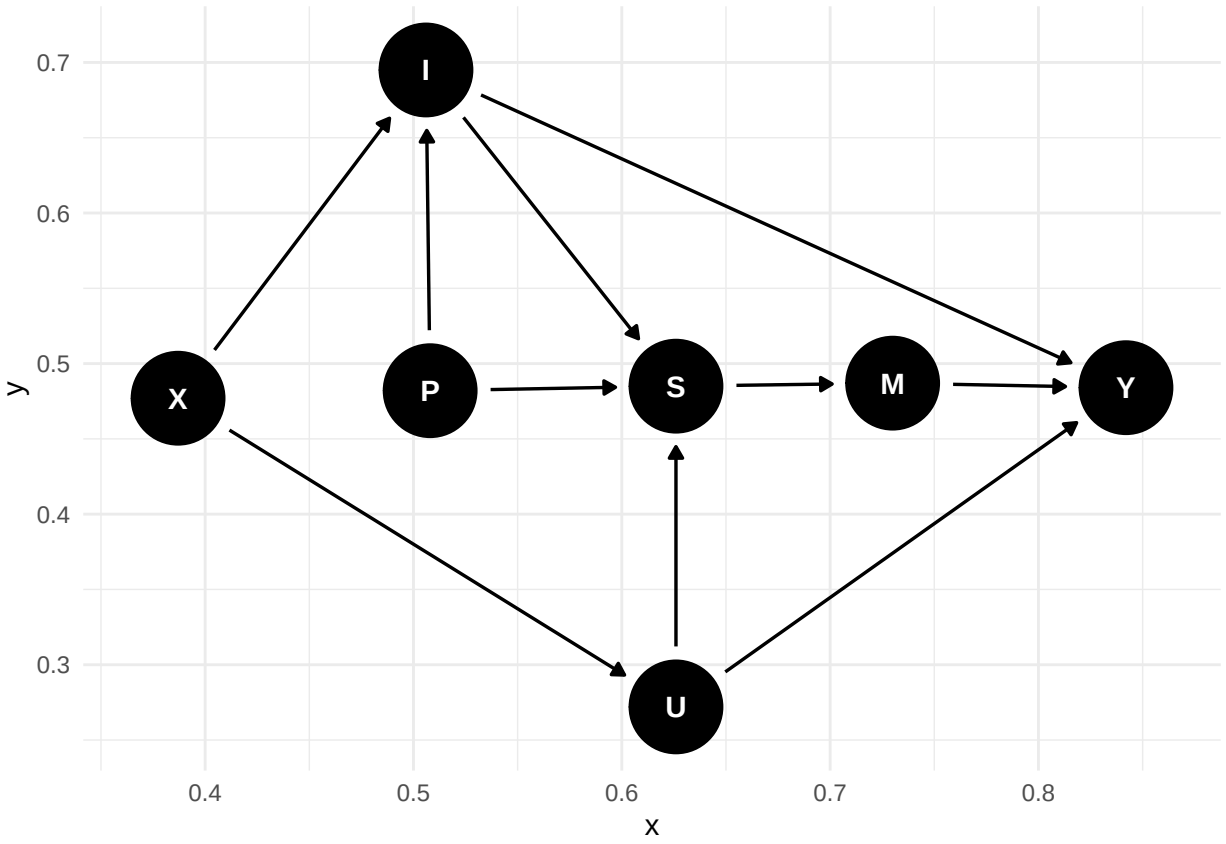

```
  labs(title = "First Step of Front-Door Criterion Estimation")
```

```
## $title
## [1] "First Step of Front-Door Criterion Estimation"
##
## attr(,"class")
## [1] "labels"
```

```
ggdag_adjustment_set(dag_first, outcome = "M", exposure = "S")
```



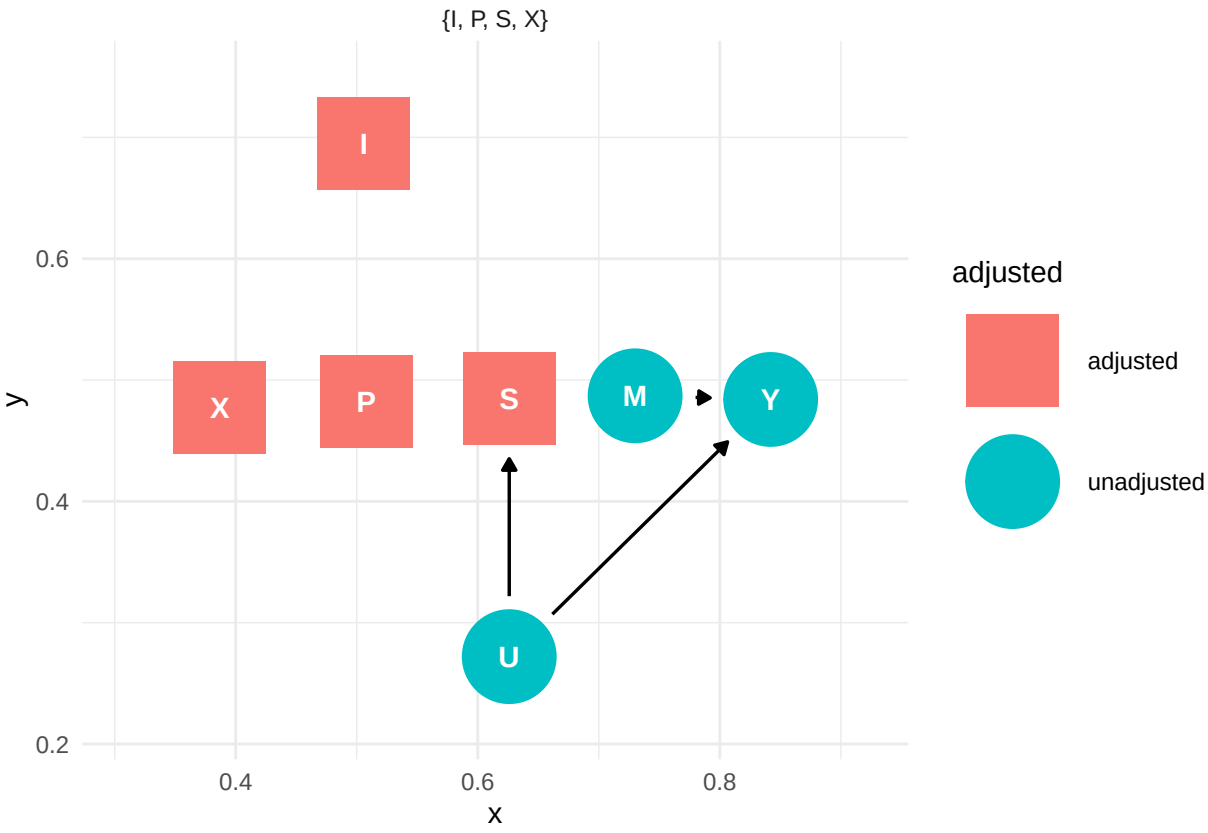

```
# This is a pre-created DAG showing the second-step of the front-door criterion estimation approach.
# Note that the FD criterion is just two back-door criterion for the full model
dag_second <- downloadGraph("dagitty.net/m9viMnf")
ggdag(dag_second)
```




```
  labs(title = "Second Step of Front-Door Criterion Estimation")
```

```
## $title
## [1] "Second Step of Front-Door Criterion Estimation"
##
## attr(,"class")
## [1] "labels"
```

```
ggdag_adjustment_set(dag_second, outcome = "M", exposure = "S")
```



Define the function to conduct the Monte Carlo (MC) Experiment

```
fd <- function(n, a, b, c, d, g, h, j, k, l, u1, u2) {
  U <- rnorm(n,10,1)                          # `U' is the unobserved confounder
  P <- rnorm(n, 12, sd = 1)                   # `P' is parents' education; on average, twelve years
  X <- rnorm(n, 5, sd = 1)                    # `X' is social origin
  I <- rnorm(n, 25, sd = 1)                   # `I' is parents' income; defined as dollars per hr
  e_m <- rnorm(n,0,1)                         # `e_m' is the noise term on the mediator equation
  e_y <- rnorm(n,0,1)                         # `e_y' is the noise term on the outcome equation
  e_s <- rnorm(n,0,1)                         # `e_s' is the noise term on the schooling equation
  S <- 14 + u1*U + e_s                        # `S' is the years of schooling; on average, fourteen years
  M <- a*S + b*I + c*P + d*X + e_m            # `M' is the mediator; e.g., training program
  Y <- g*M + j*I + k*P + l*X + u2*U + e_y     # `Y' is current earnings; defined as dollars per hr

  # Notice that `S' is missing from this `Y' equation.
  # This is done intentionally so that the true and biased causal models
  # do not double count years of schooling; otherwise, the estimated parameter
  # is twice as large as it should be. For the front-door (FD) criterion estimator,
  # we bring the 'S' variable back into the equation below in the `fdmodel'.

  df1 <- data.frame(X, I, P, S, M, Y, U)     # This collects all of the data into a data frame

  # The `fdmodel' specification is necessary for the MC simulations (lavaan pkg).
  # `ATE' is the FD average treatment effect estimate.
```

```
  fdmodel <- "M ~ a*S + b*I + c*P + d*X
              Y ~ g*M + h*S + j*I + k*P + l*X
              ATE := a*g"

  # The `fd' variable is the ATE estimate.
  # The `unadj' variable is the biased (naive) estimate; i.e., not accounting for `U'.
  # The `adj' variable is the true causal estimate; i.e., accounting for `U'.

  fd <- sem(fdmodel,df1)@ParTable$est[22]
  unadj <- coef(lm(Y~S))[2]
  adj <- coef(lm(Y~S+U))[2]
  return(c(fd,unadj,adj))
}
```

Calibration and replication of the the 'fd' function

```
# `res' is a data frame with the results. In this case, the number of replications is 5000.
# The sample size, for each replication, is defined as 1000 persons.
# Otherwise, the parameter callibrations in `res' correspond to the `fd' function above.
# Note that `u1', the parameter on `U' in the schooling equation, is set to 0.8, which indicates
# fairly strong correlation between `S' and `U'.

res <- data.frame(t(replicate(5000,fd(1000, 0.875, 0.5, 0.5, 0.5, 0.875, 0.5, 0.5, 0.5, 0.5, 0.8, 0.1 ))
names(res) <- c("fd","unadj","adj")

# Average, FD estimate
mean(res$fd)
```

```
## [1] 0.7656678
```

```
# Average, true causal estimate
mean(res$adj)
```

```
## [1] 0.7657582
```

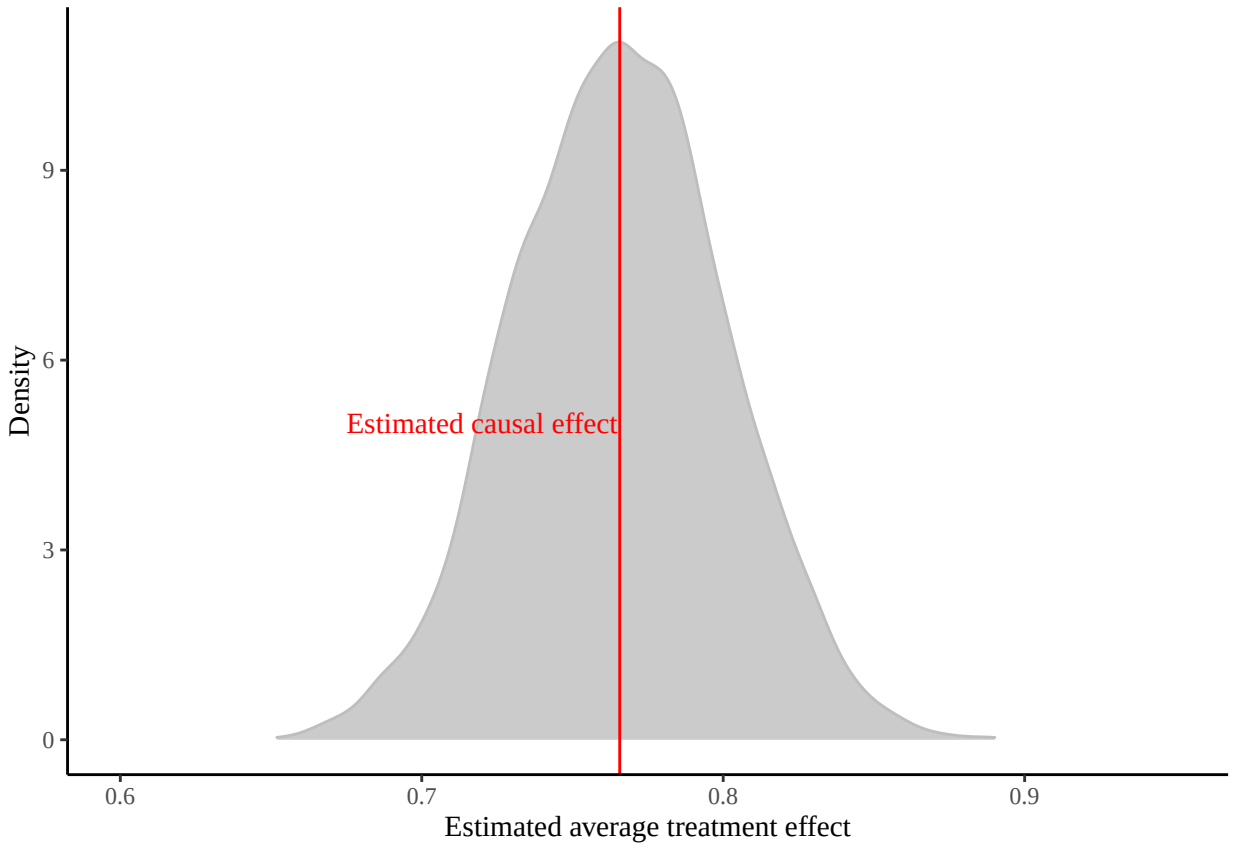
```
# Average, biased (naive) estimate
mean(res$unadj)
```

```
## [1] 0.8150807
```

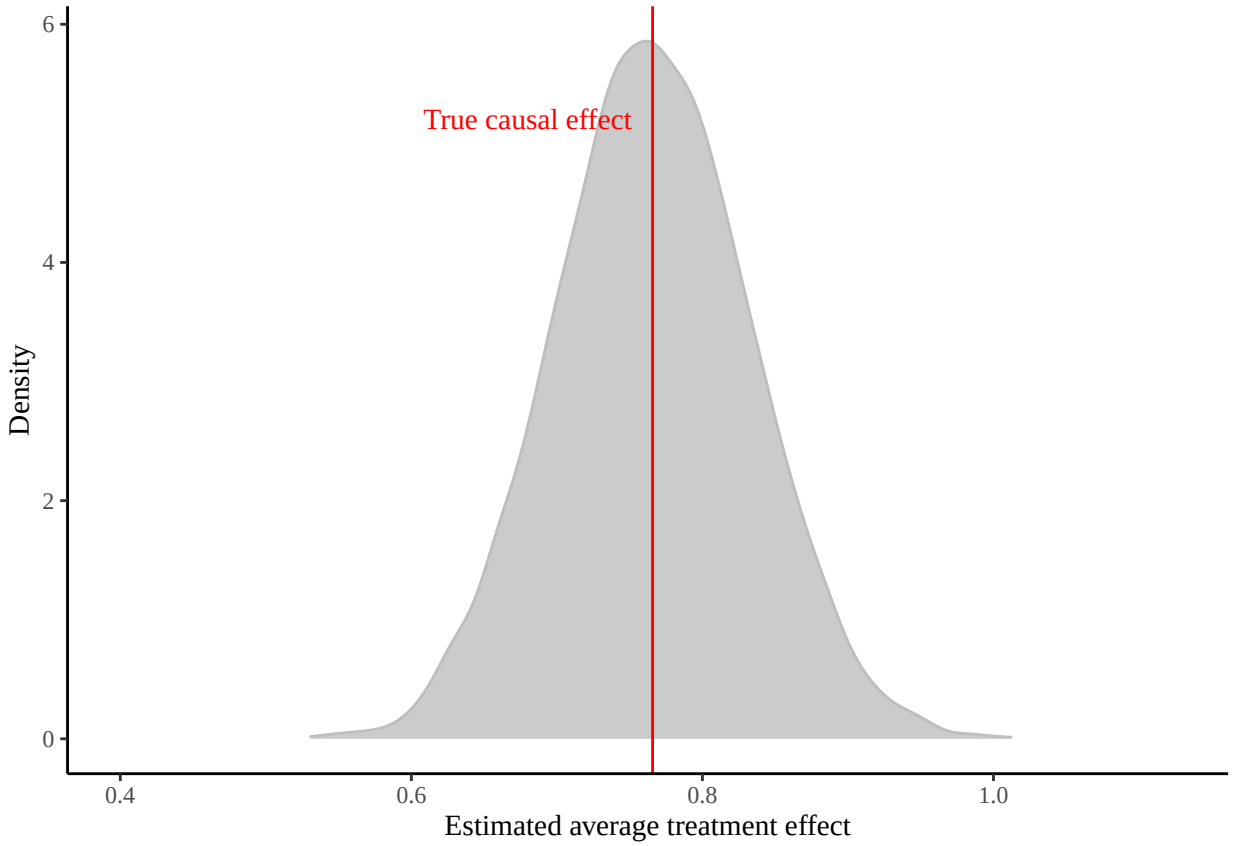
Graph the distributions of the estimated parameters based on the MC simulations

```
# Graph for the distribution of the FD estimate
ggplot(res,aes(x=fd)) +
  geom_density(alpha =0.8, fill = "gray", color = "gray") +
  theme_classic() +
  geom_vline(xintercept = mean(res$fd), linetype = 1, color = "red") +
  annotate("text", x = 0.72, y = 5, family = 'serif', label = "Estimated causal effect", color = "red")
```

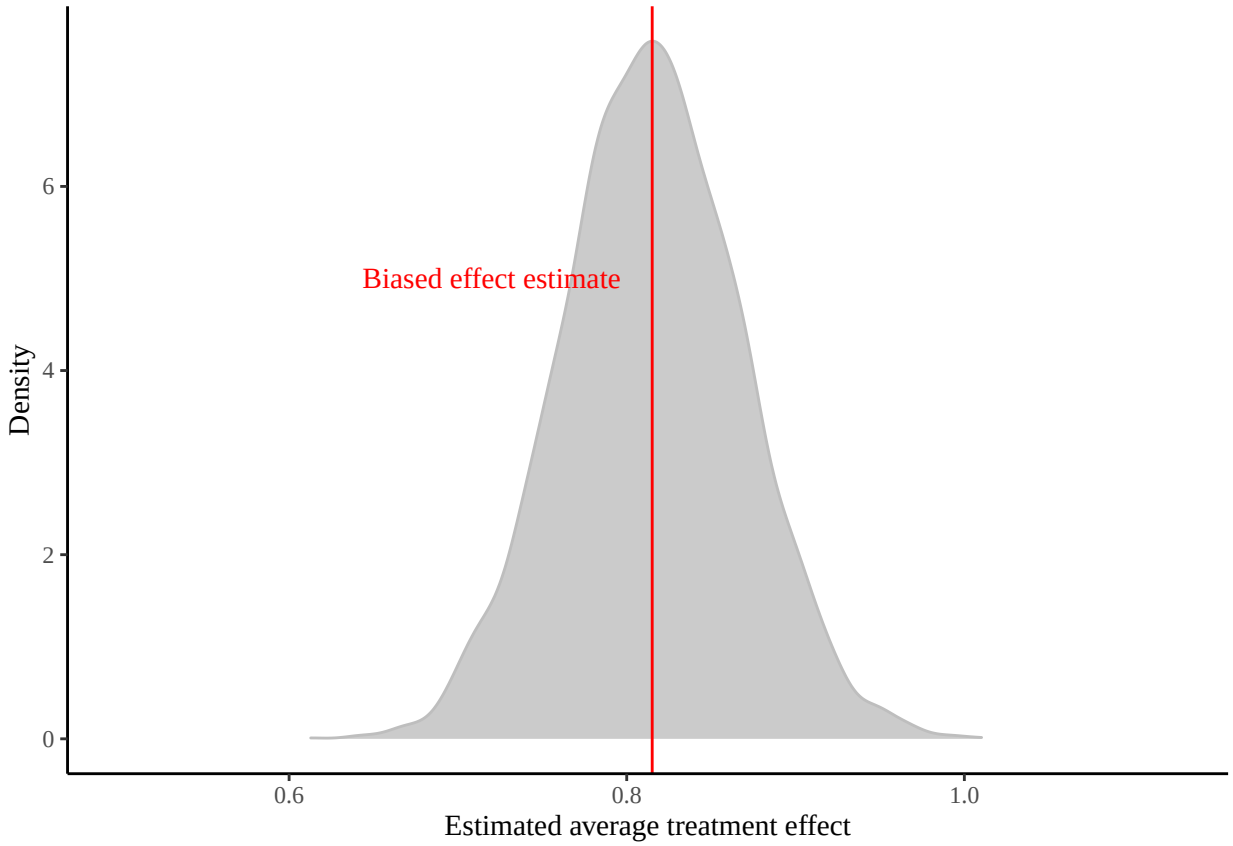
```
  xlab("Estimated average treatment effect") +
  ylab("Density") +
  theme(text=element_text(family = 'serif', color = 'black')) +
  coord_cartesian(xlim=c(0.6, 0.95))
```




```
# Graph for the distribution of the true causal estimate
ggplot(res,aes(x=adj)) +
  geom_density(alpha = 0.8, fill = "gray", color = "gray") +
  theme_classic() +
  geom_vline(xintercept = mean(res$adj), linetype = 1, color = "red") +
  annotate("text", x = 0.68, y = 5.2, family = 'serif', label = "True causal effect", color = "red") +
  xlab("Estimated average treatment effect") +
  ylab("Density") +
  theme(text=element_text(family = 'serif', color = 'black')) +
  coord_cartesian(xlim=c(0.4, 1.125))
```




```
# Graph for the distribution of the biased (naive) estimate
ggplot(res,aes(x=unadj)) +
  geom_density(alpha = 0.8, fill = "gray", color = "gray") +
  theme_classic() +
  geom_vline(xintercept = mean(res$unadj), linetype = 1, color = "red") +
  annotate("text", x = 0.72, y = 5, family = 'serif', label = "Biased effect estimate", color = "red") +
  xlab("Estimated average treatment effect") +
  ylab("Density") +
  theme(text=element_text(family = 'serif', color = 'black')) +
  coord_cartesian(xlim=c(0.5, 1.125))
```


Biased effect estimate
Density
6
4
2
0
0.6
0.8
1.0
Estimated average treatment effect