Cook's Distance for Observation Selection-MC experiments

Weihao Li

2020-10-10

Correct DGP

Assume We have such a data generating process:

$$Y_t = \beta_0 + \varepsilon_t, \varepsilon_t \sim N(0, 10^2), t = 1, 2, .., T$$

And then use the `lm` function to fit a model with only a constant term.

$$\hat{y}_t = \hat{\beta}_0$$

The MLE or the OLS solution for this model is $\hat{\beta}_0 = \frac{1}{T}\sum_{t=1}^{T} y_t$

After that, we use the Cook's distance to drop observation with value greater than $\frac{4}{T}$. Refit the model, and record the coefficient as $\hat{\beta}_{0,Cook}$.

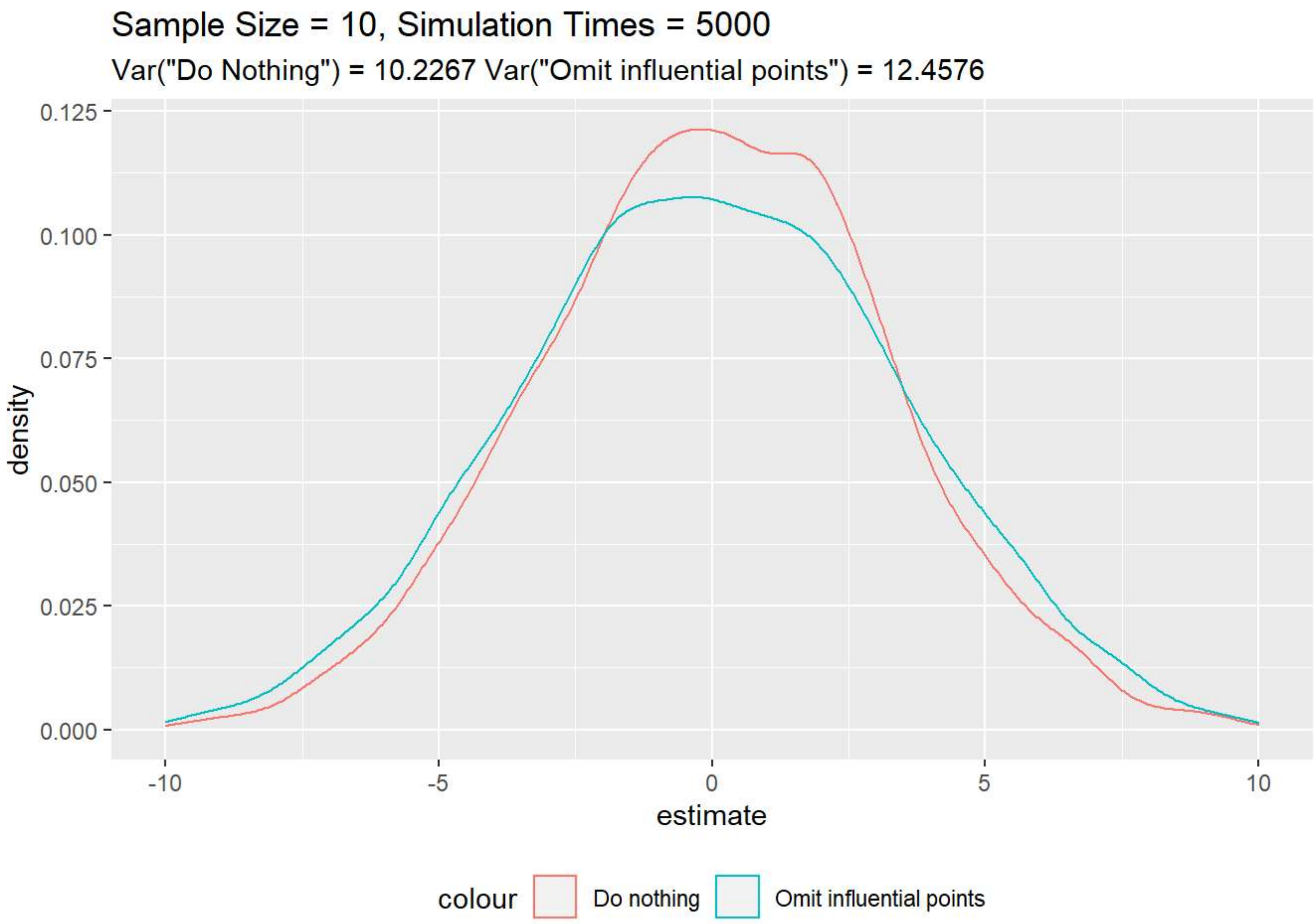
The aim of this experiment is to understand the behavior of $\hat{\beta}_0$ and $\hat{\beta}_{0,Cook}$.

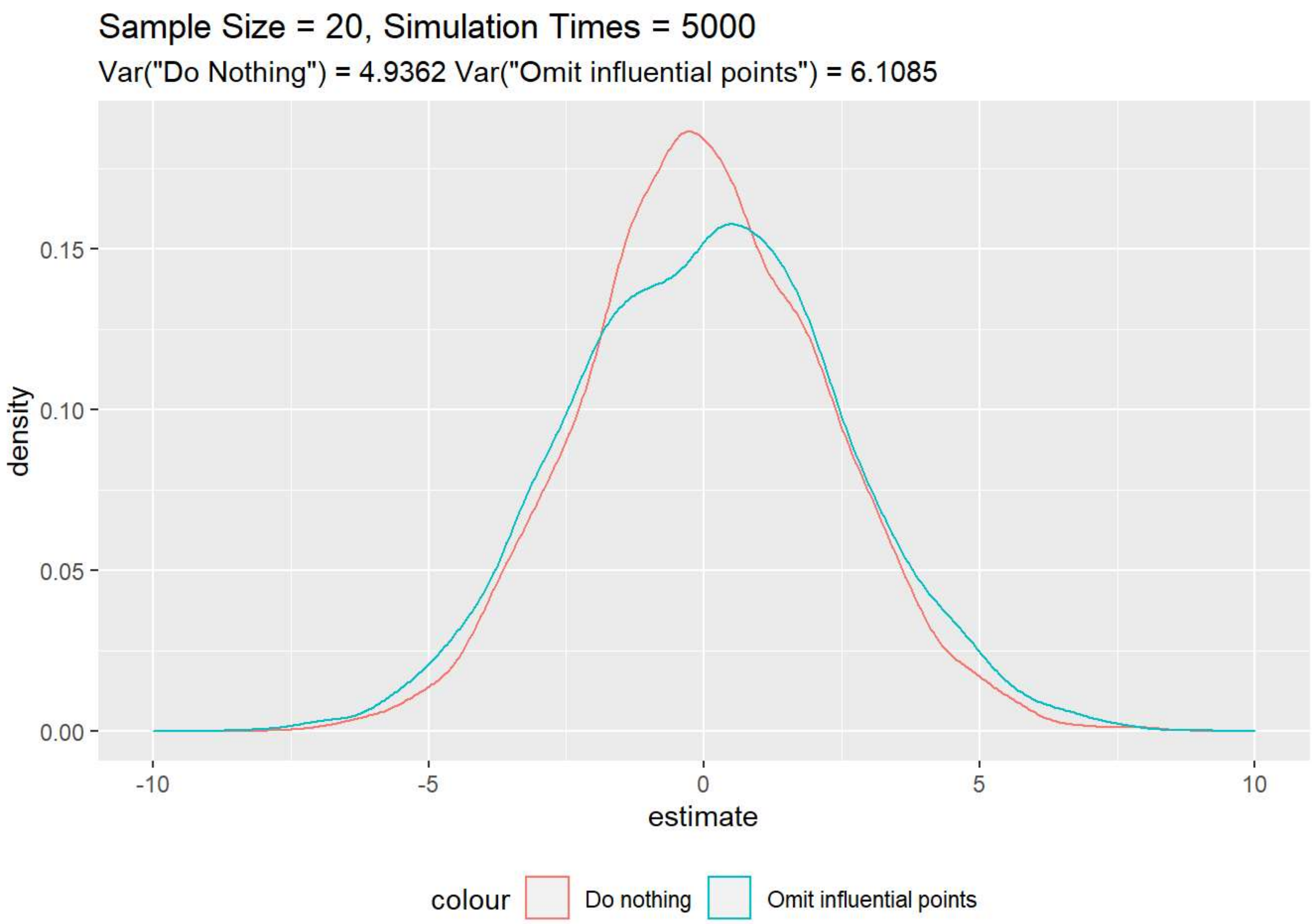
Theoretically, with correctly specified model, MLE is asymptotically efficient, which reach the Cramer Rao lower bound.

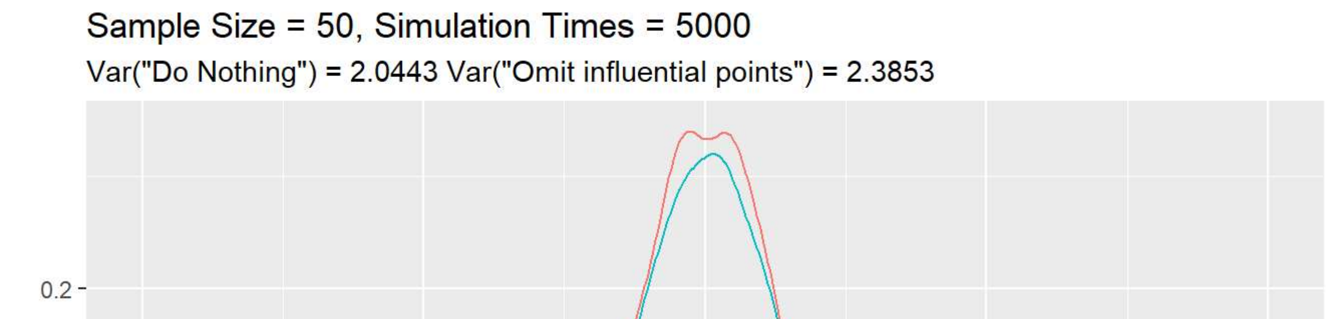
Thus, if we believe our model is correct, don't use Cook's Distance to drop any observation.

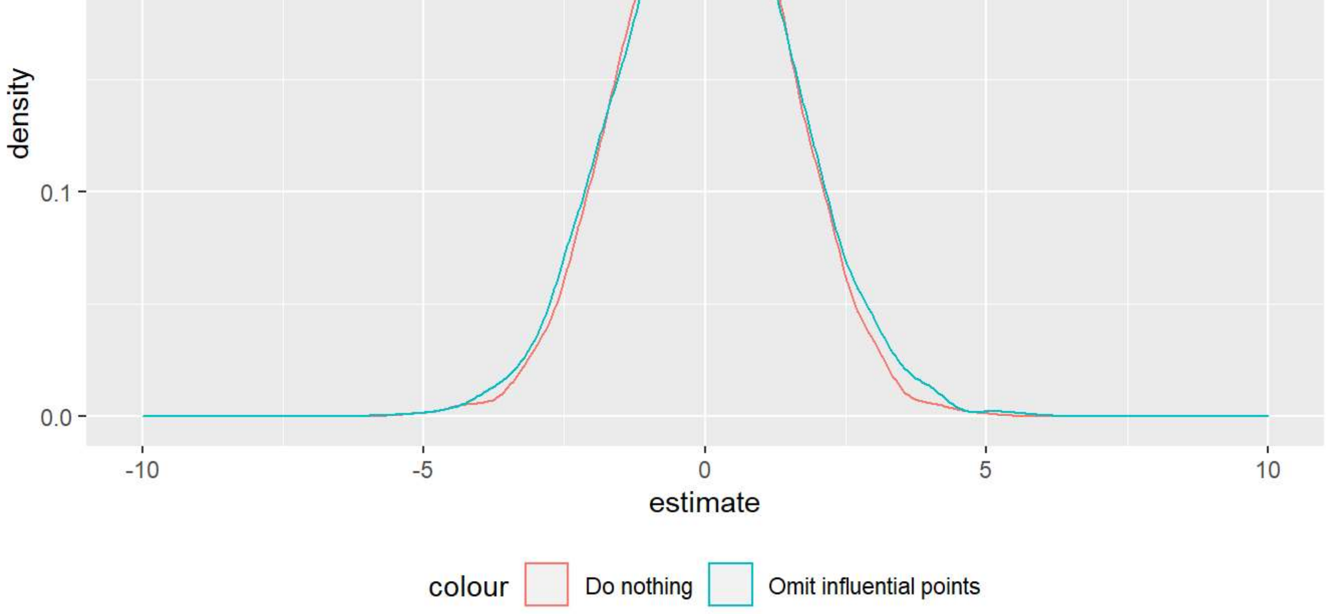
As you can see from the sampling distribution, MLE always has a smaller variance, no matter what the sample size is.

```r
mc <- function(T = 100, N = 100, cutoff = TRUE){

  estimate <- rep(NA, N)

  for (i in 1:N){

    y <- rnorm(T, sd = 10)
    mod <- lm(y~1)

    if (cutoff){
      y <- y[!cooks.distance(mod) > 4/T]
      mod <- lm(y~1)
    }

    estimate[i] <- as.numeric(mod$coefficients)
  }

  return(estimate)

}

set.seed(100)
N <- 5000

for (T in c(10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000)){
  estimate_d <- mc(T = T, N = N, cutoff = TRUE)
  estimate <- mc(T = T, N = N, cutoff = FALSE)

  vard <- var(estimate_d)
  varn <- var(estimate)

  p <- ggplot() +
    geom_density(aes(estimate, col = "Do nothing")) +
    geom_density(aes(estimate_d, col = "Omit influential points")) +
    ggtitle(paste0("Sample Size = ", T, ", Simulation Times = ", N)) +
    xlim(c(-10, 10)) +
    theme(legend.position = "bottom") +
    labs(subtitle = paste0('Var("Do Nothing") = ',
                           round(varn, 4),
                           ' Var("Omit influential points") = ',
                           round(vard, 4)))

  print(p)
  #ggsave(paste0("plots/", T, ".jpeg"), plot = p)
}
```


Sample Size = 10, Simulation Times = 5000
Var("Do Nothing") = 10.2267 Var("Omit influential points") = 12.4576
0.125
0.100
0.075
0.050
0.025
0.000
density
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points

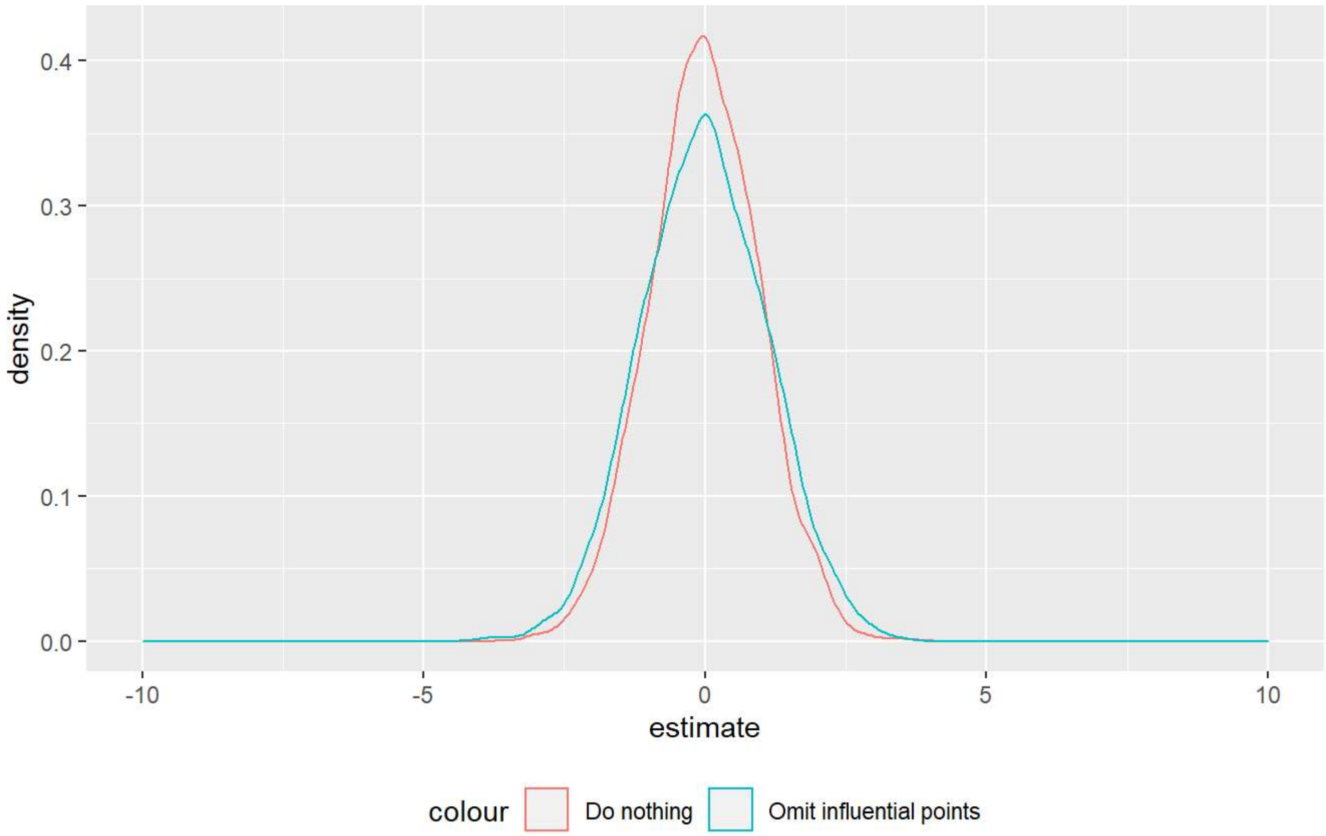

Sample Size = 20, Simulation Times = 5000
Var("Do Nothing") = 4.9362 Var("Omit influential points") = 6.1085
0.15
0.10
0.05
0.00
density
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points

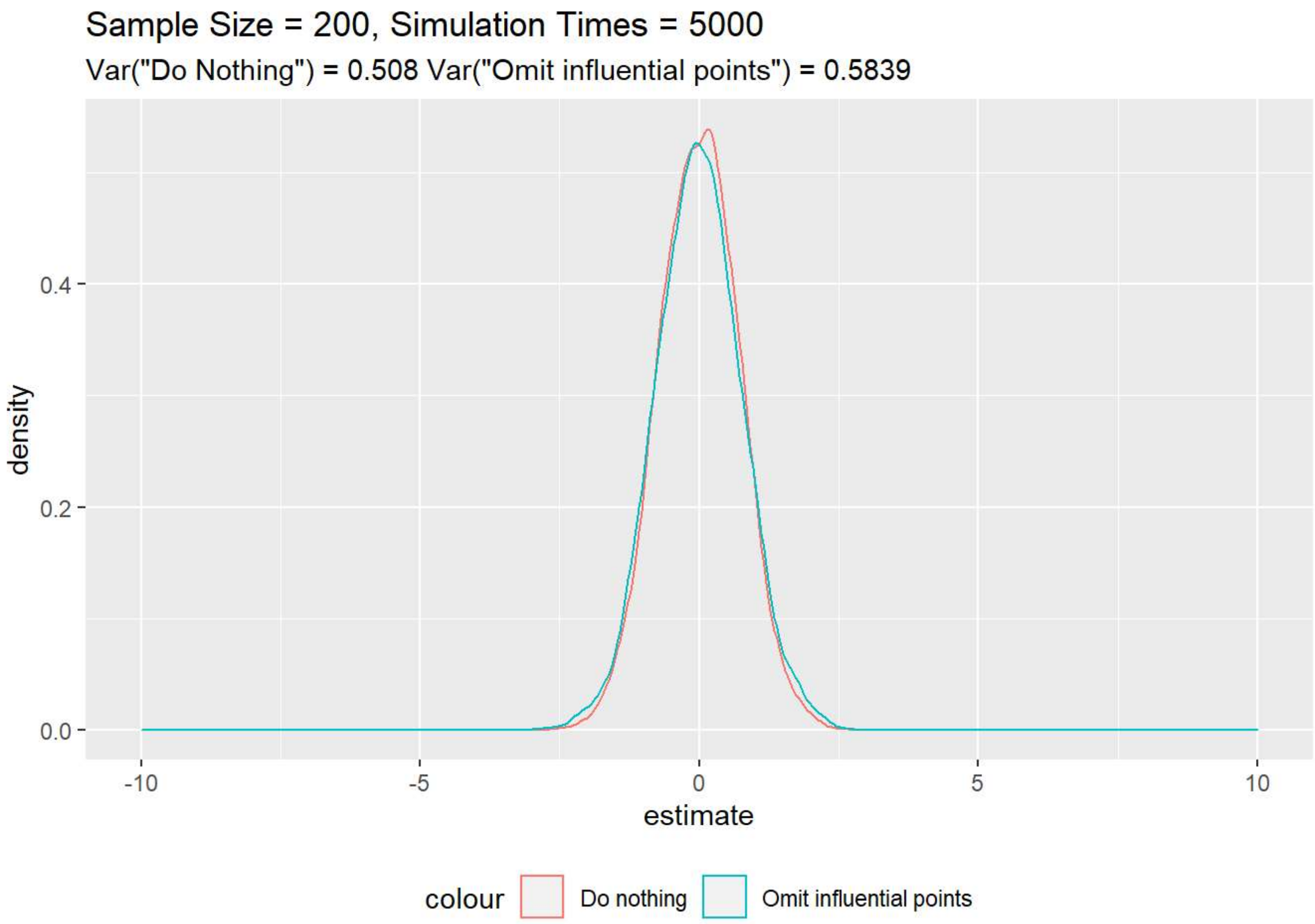

Sample Size = 50, Simulation Times = 5000
Var("Do Nothing") = 2.0443 Var("Omit influential points") = 2.3853
0.2



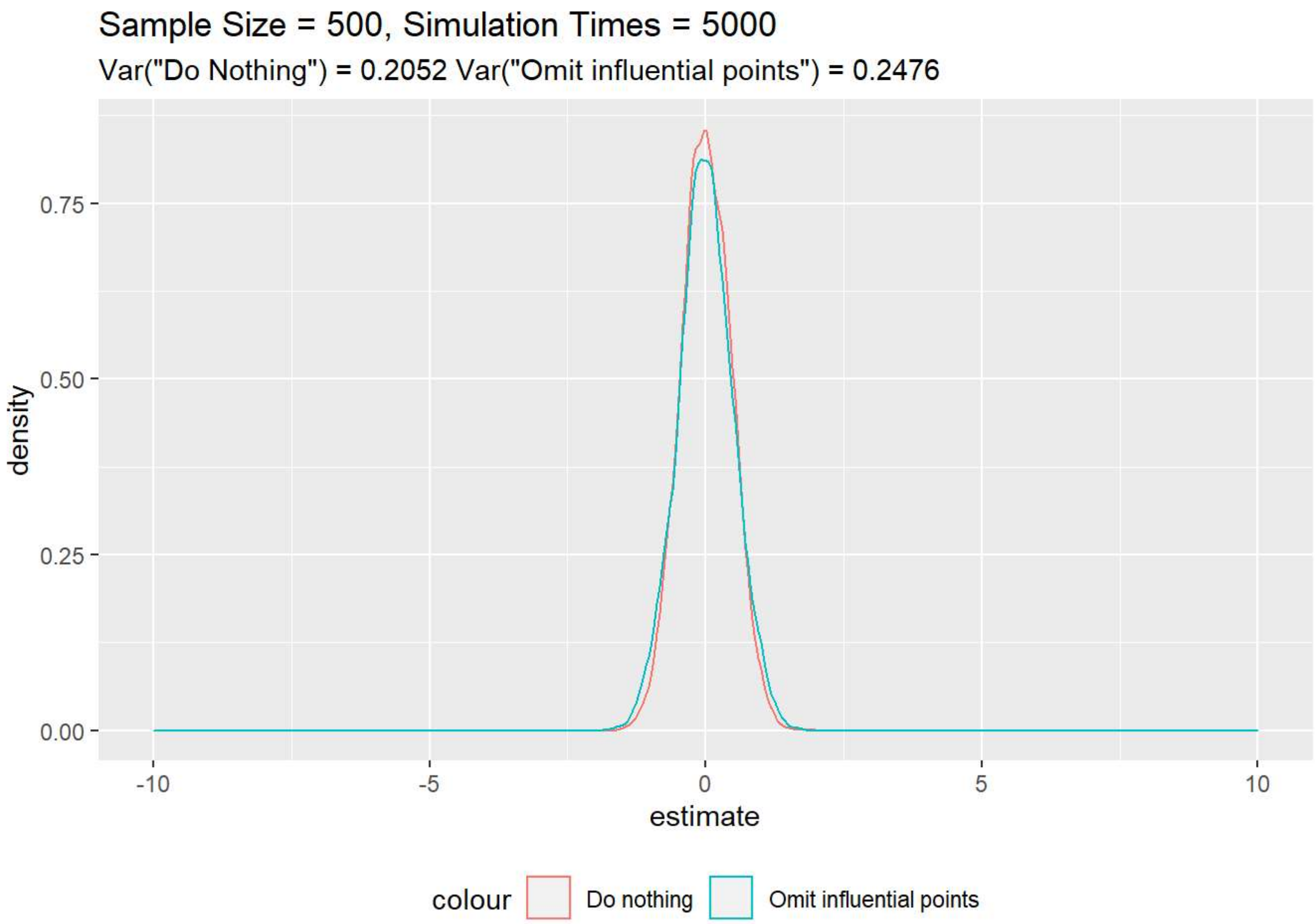
Sample Size = 100, Simulation Times = 5000

Var("Do Nothing") = 0.9375 Var("Omit influential points") = 1.2406

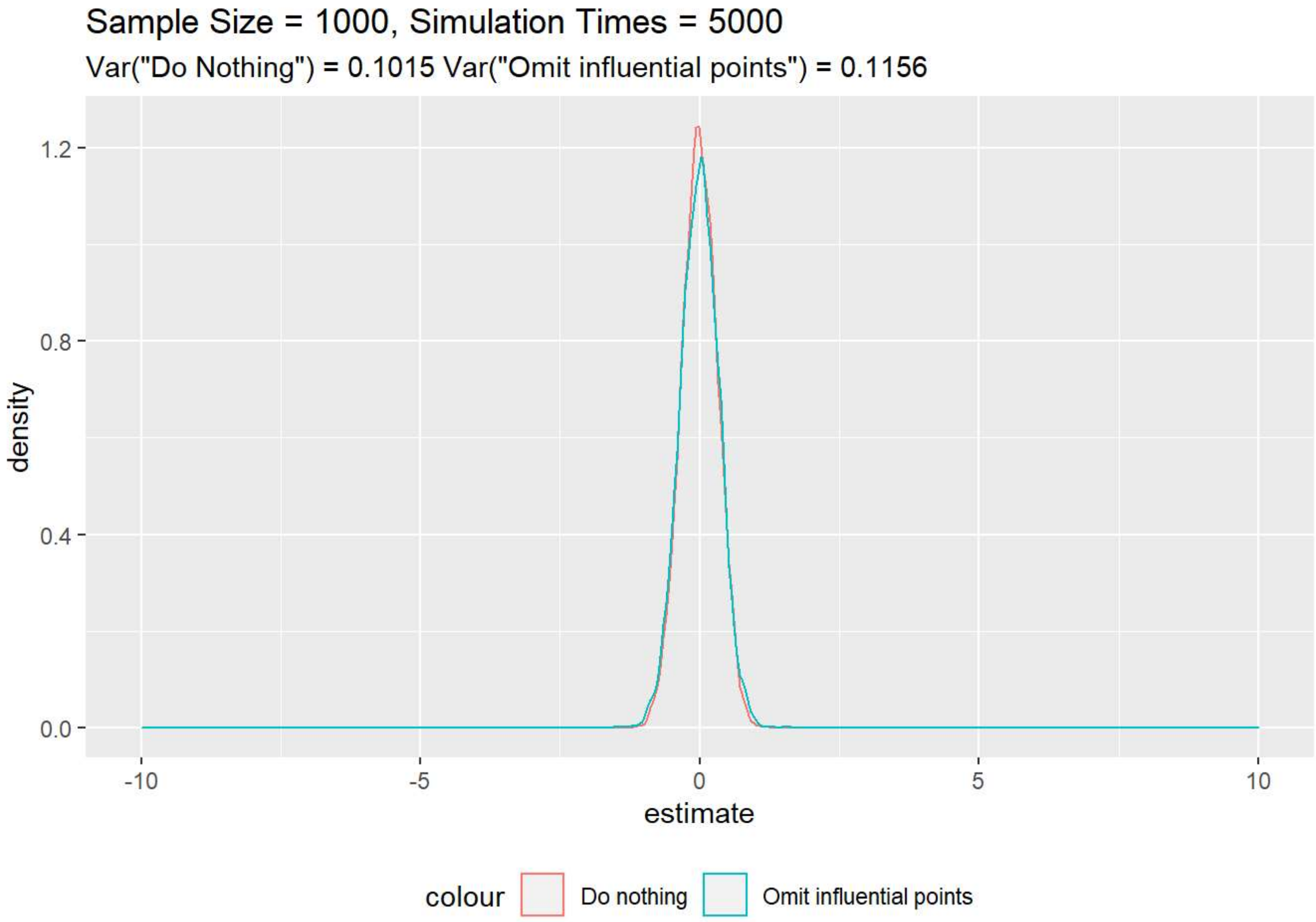


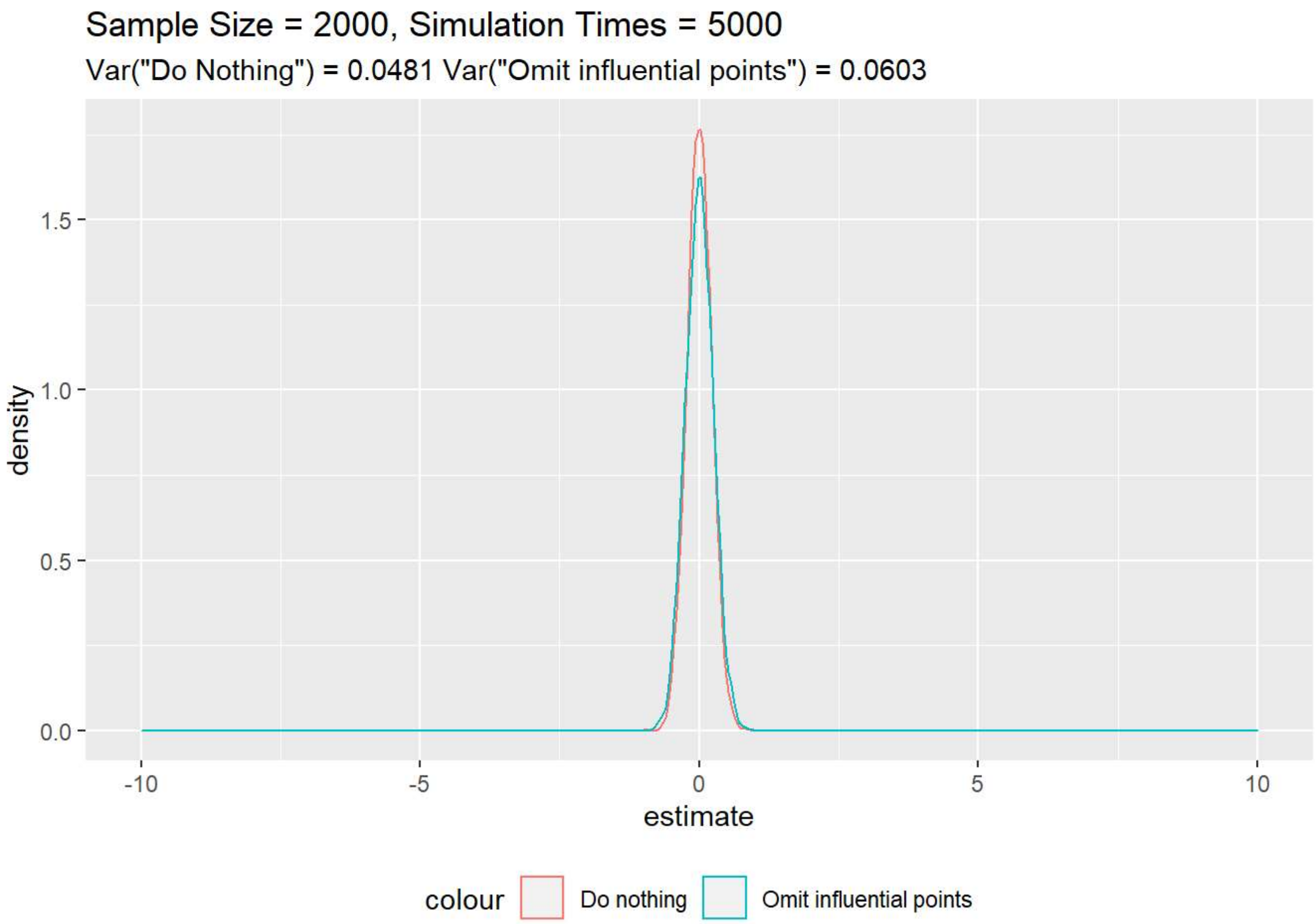


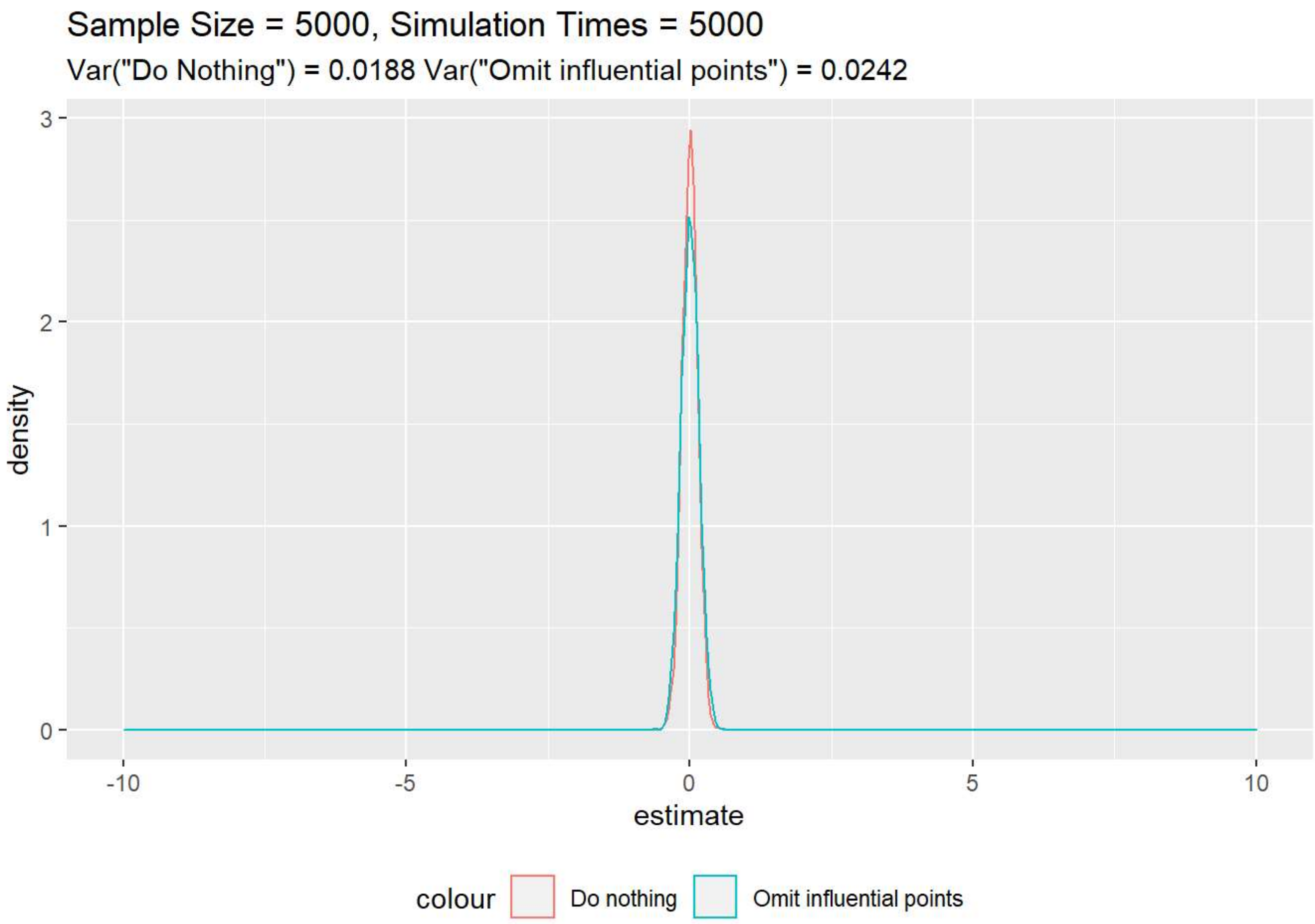
Sample Size = 200, Simulation Times = 5000
Var("Do Nothing") = 0.508 Var("Omit influential points") = 0.5839
0.4
0.2
0.0
density
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points

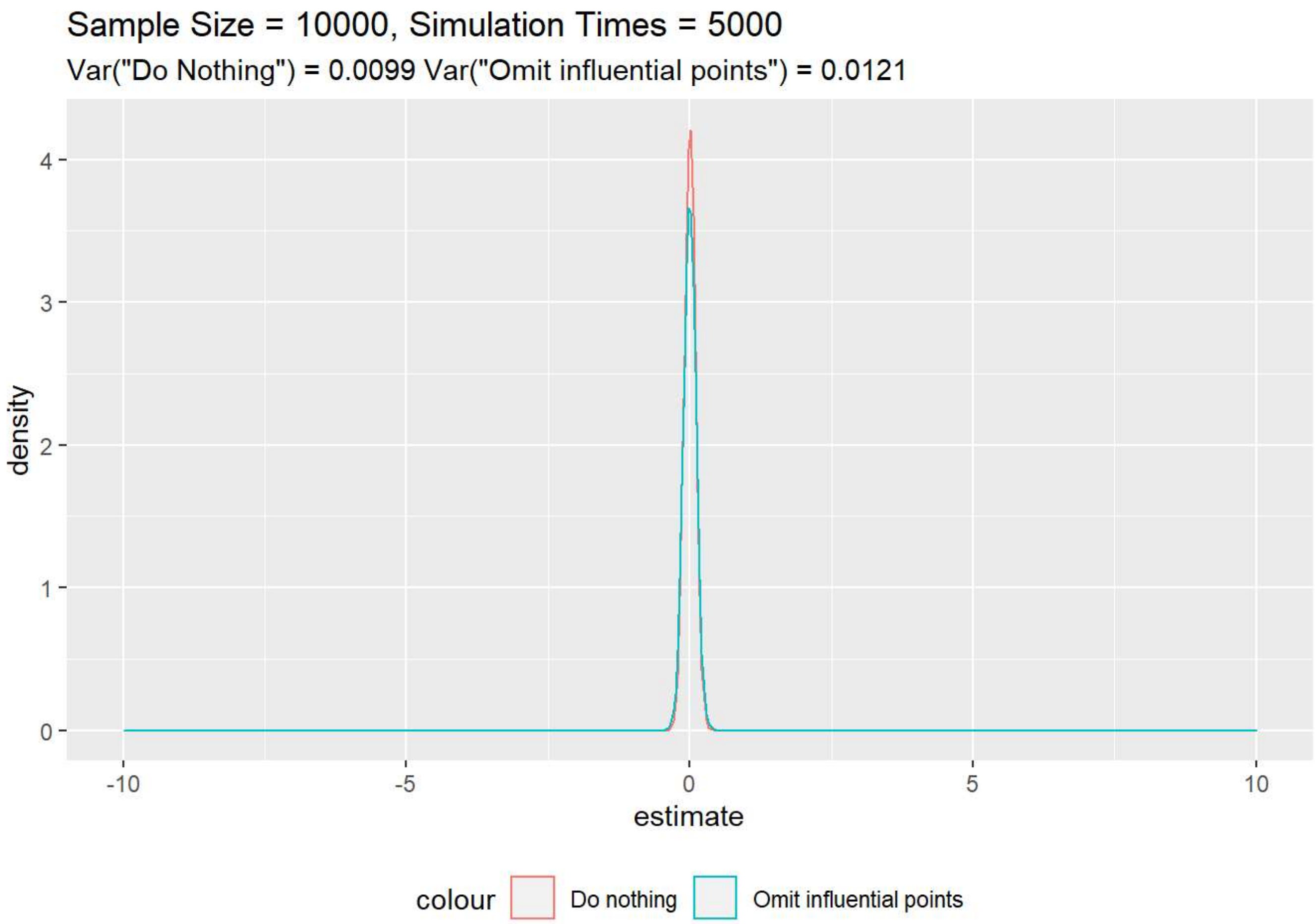


Sample Size = 500, Simulation Times = 5000
Var("Do Nothing") = 0.2052 Var("Omit influential points") = 0.2476
0.75
0.50
0.25
0.00
density
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points


Sample Size = 1000, Simulation Times = 5000
Var("Do Nothing") = 0.1015 Var("Omit influential points") = 0.1156
1.2
0.8
0.4
0.0
density
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points


Sample Size = 2000, Simulation Times = 5000
Var("Do Nothing") = 0.0481 Var("Omit influential points") = 0.0603
1.5
1.0
0.5
0.0
density
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points





Heteroscedasticity

What if we encounter heteroscedasticity?

Assume we have such a dataset

95% of them have this DGP:

$$Y_t = \beta_0 + \varepsilon_t, \varepsilon_t \sim N(0, 10^2), t = 1, 2, .., T$$

And the remaining 5% have another DGP:

$$Y_t = \beta_0 + \varepsilon_t, \varepsilon_t \sim N(0, 100^2), t = 1, 2, .., T$$

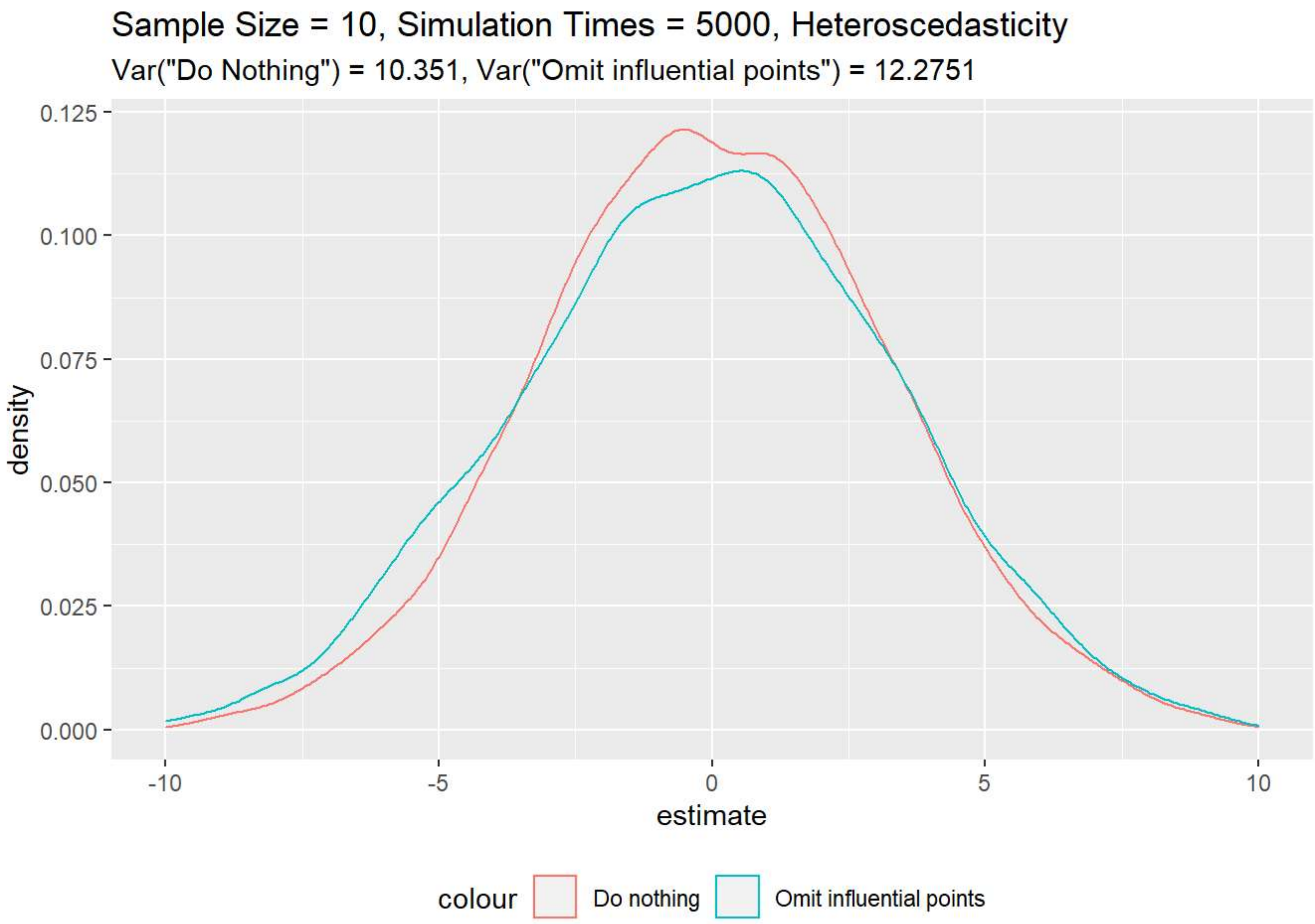
You can consider this 5% of data to be outliers.

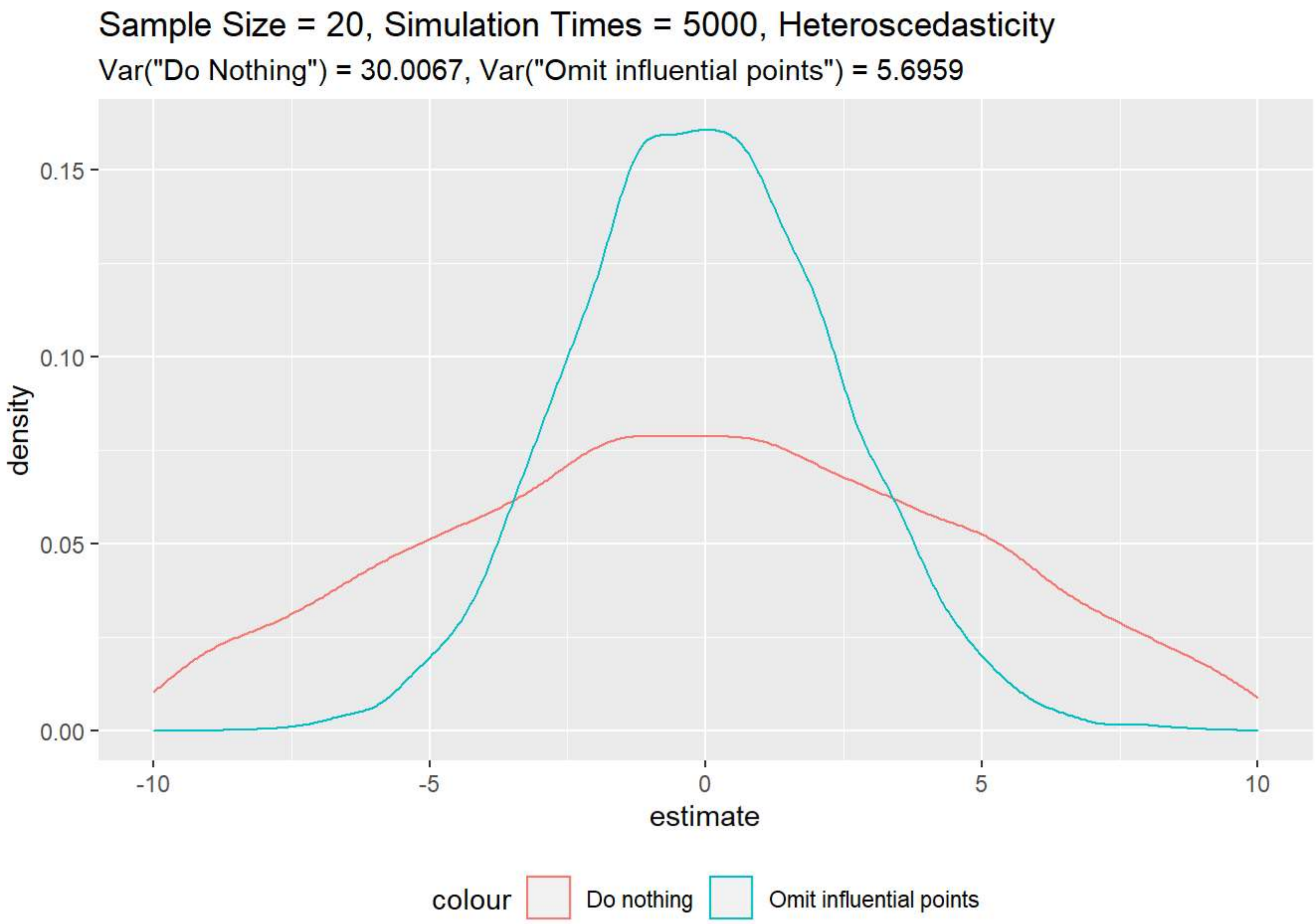
Using this dataset, we redo the experiment. This time, we can clearly see the benefit of dropping influential points. The sampling distribution of $\hat{\beta}_{0,Cook}$ is better.

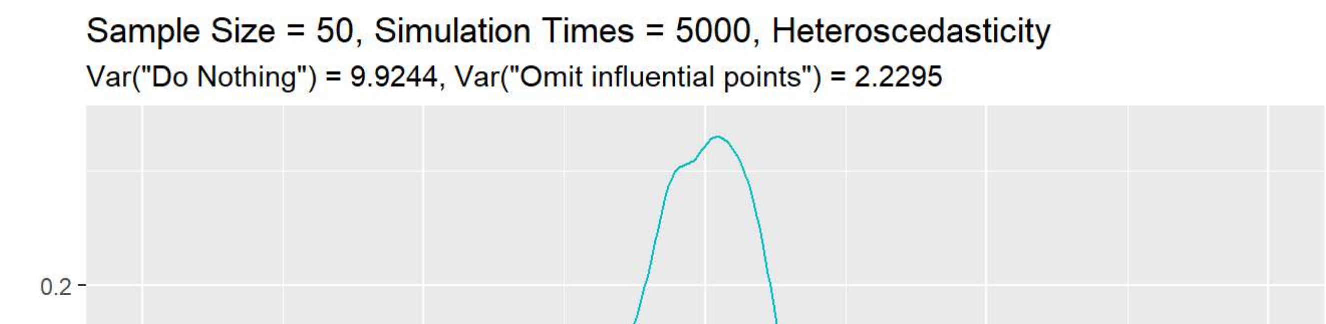
In both methods, the estimator is consistent. You can prove it by using simple algebra.

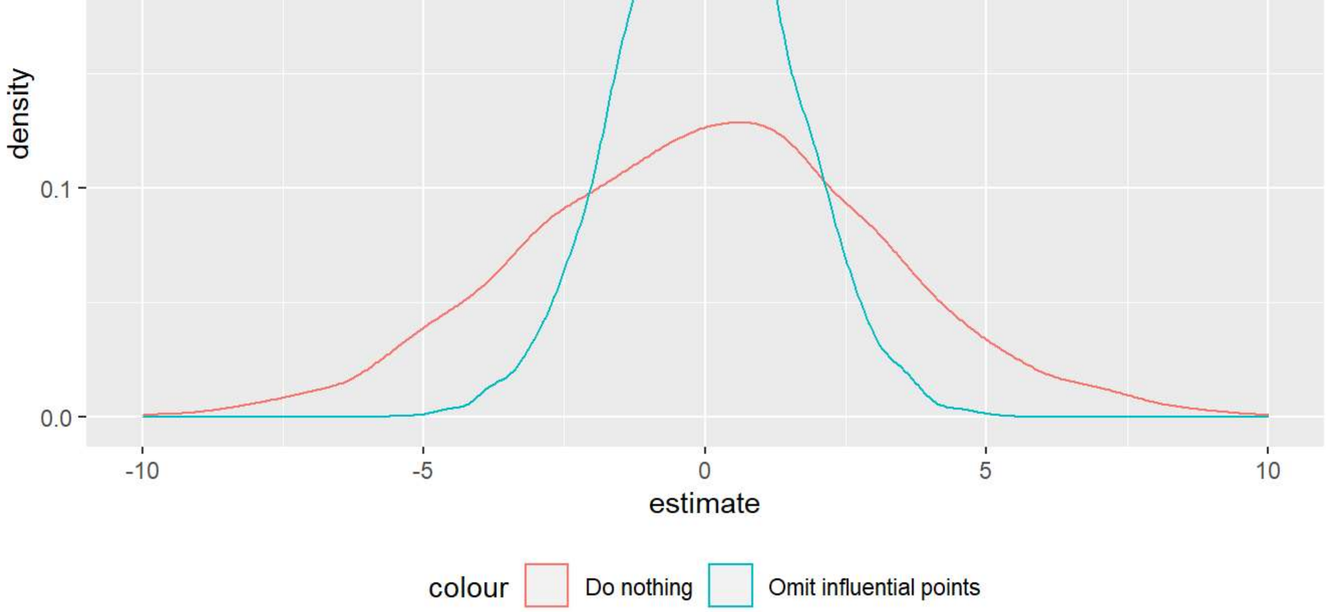
Therefore, when you feel there is a heteroscedasticity problem, you may try to drop influential points to boost your efficiency.

```r
mc2 <- function(T = 100, N = 100, cutoff = TRUE){

  estimate <- rep(NA, N)

  for (i in 1:N){

    y <- rnorm(round(T*0.95), sd = 10)
    y <- c(y, rnorm(T - round(T*0.95), sd = 100))
    mod <- lm(y~1)

    if (cutoff){
      y <- y[!cooks.distance(mod) > 4/T]
      mod <- lm(y~1)
    }

    estimate[i] <- as.numeric(mod$coefficients)
  }

  return(estimate)

}

set.seed(200)
N <- 5000

for (T in c(10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000)){
  estimate_d <- mc2(T = T, N = N, cutoff = TRUE)
  estimate <- mc2(T = T, N = N, cutoff = FALSE)

  vard <- var(estimate_d)
  varn <- var(estimate)

  p <- ggplot() +
    geom_density(aes(estimate, col = "Do nothing")) +
    geom_density(aes(estimate_d, col = "Omit influential points")) +
    ggtitle(paste0("Sample Size = ", T, ", Simulation Times = ", N, ", Heteroscedasticity")) +
    xlim(c(-10, 10)) +
    theme(legend.position = "bottom") +
    labs(subtitle = paste0('Var("Do Nothing") = ',
                           round(varn, 4),
                           ', Var("Omit influential points") = ',
                           round(vard, 4)))

  print(p)
  #ggsave(paste0("plots/H-", T, ".jpeg"), plot = p)
}
```


Sample Size = 10, Simulation Times = 5000, Heteroscedasticity
Var("Do Nothing") = 10.351, Var("Omit influential points") = 12.2751
density
0.125
0.100
0.075
0.050
0.025
0.000
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points

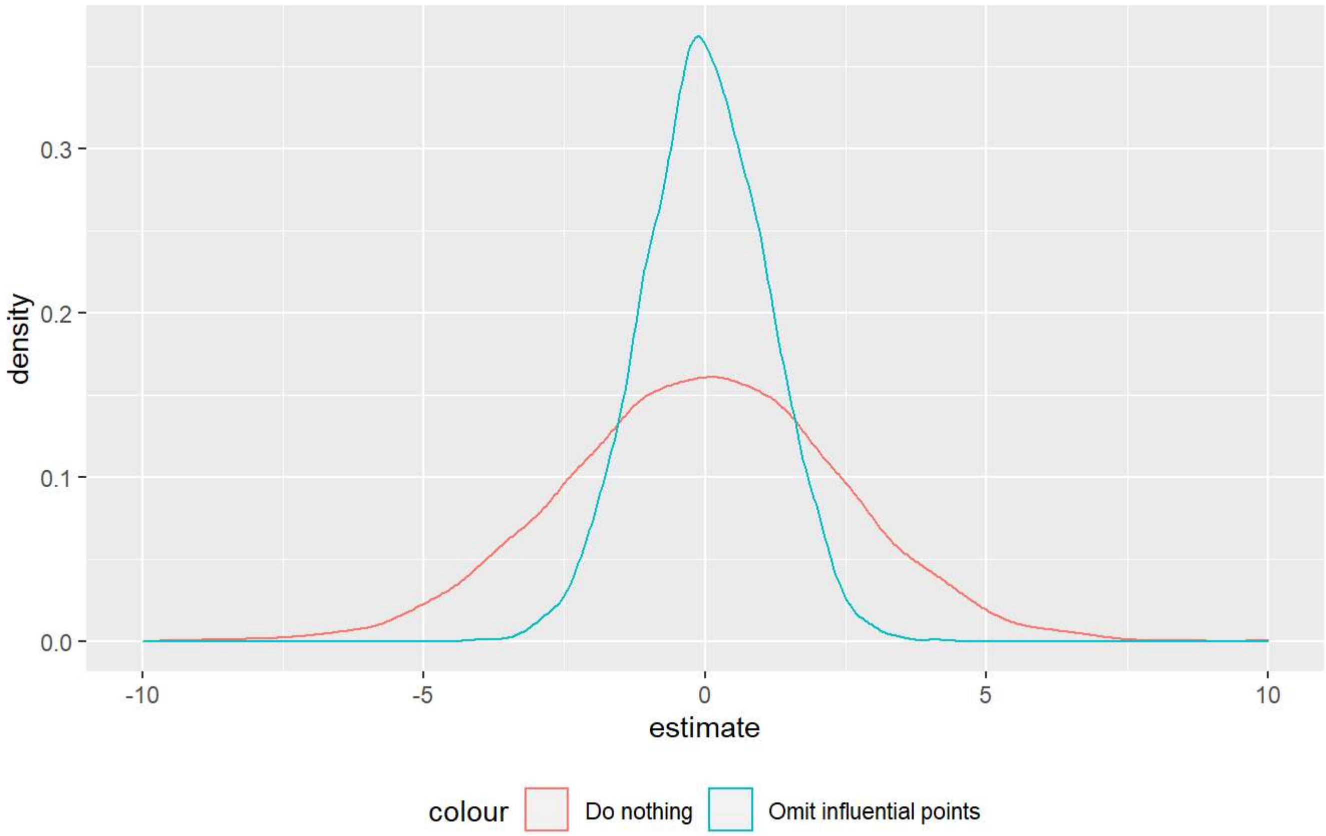

Sample Size = 20, Simulation Times = 5000, Heteroscedasticity
Var("Do Nothing") = 30.0067, Var("Omit influential points") = 5.6959
density
0.15
0.10
0.05
0.00
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points


Sample Size = 50, Simulation Times = 5000, Heteroscedasticity
Var("Do Nothing") = 9.9244, Var("Omit influential points") = 2.2295
0.2



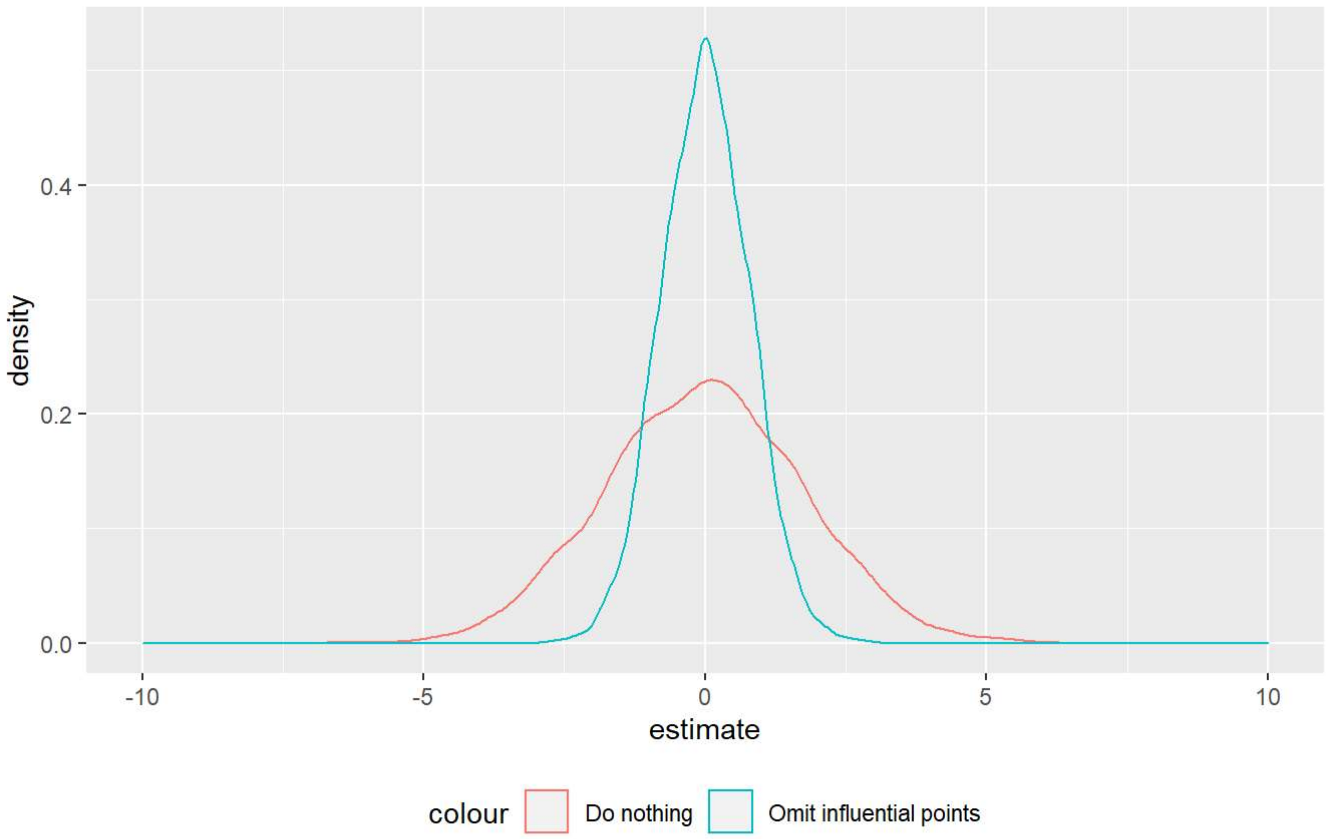
Sample Size = 100, Simulation Times = 5000, Heteroscedasticity

Var("Do Nothing") = 6.0209, Var("Omit influential points") = 1.2196



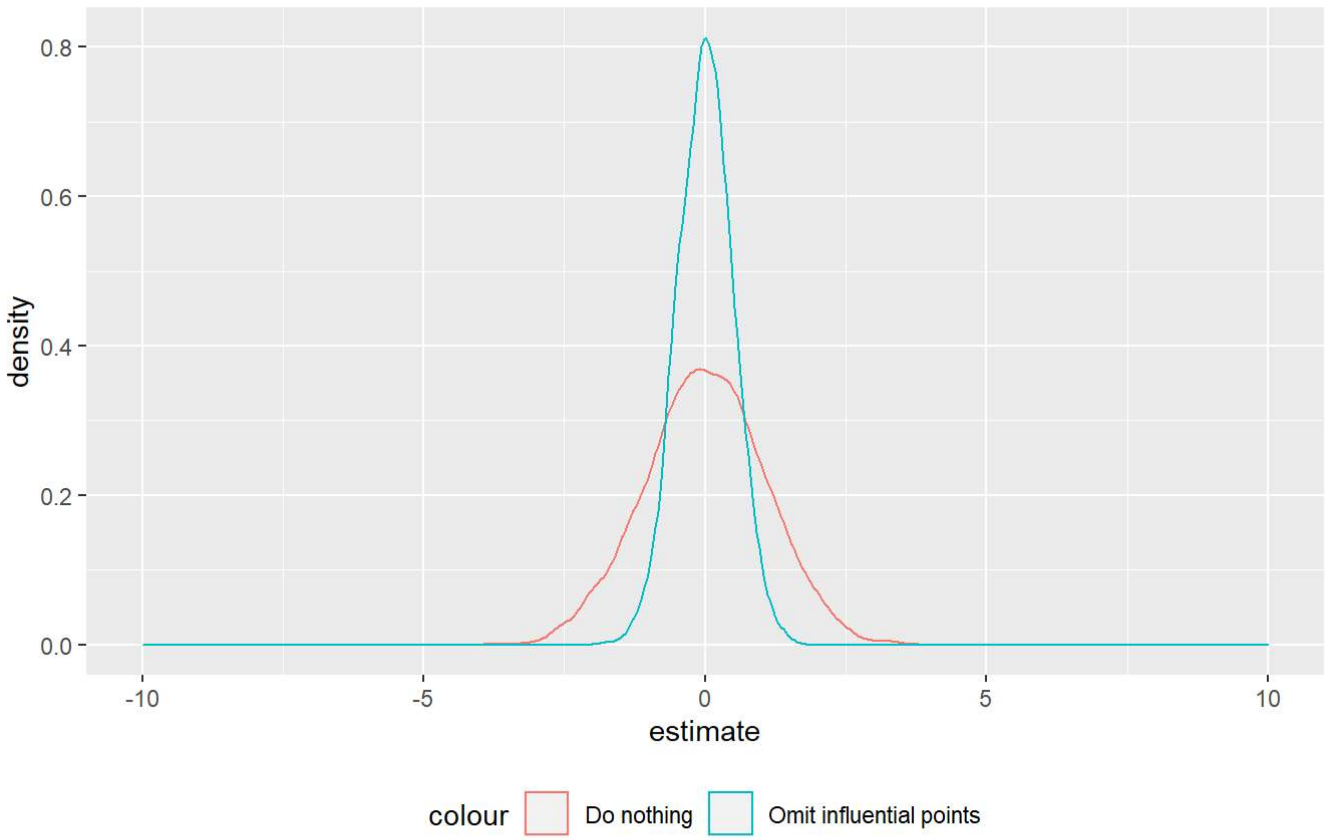
Sample Size = 200, Simulation Times = 5000, Heteroscedasticity

Var("Do Nothing") = 3.0372, Var("Omit influential points") = 0.6014



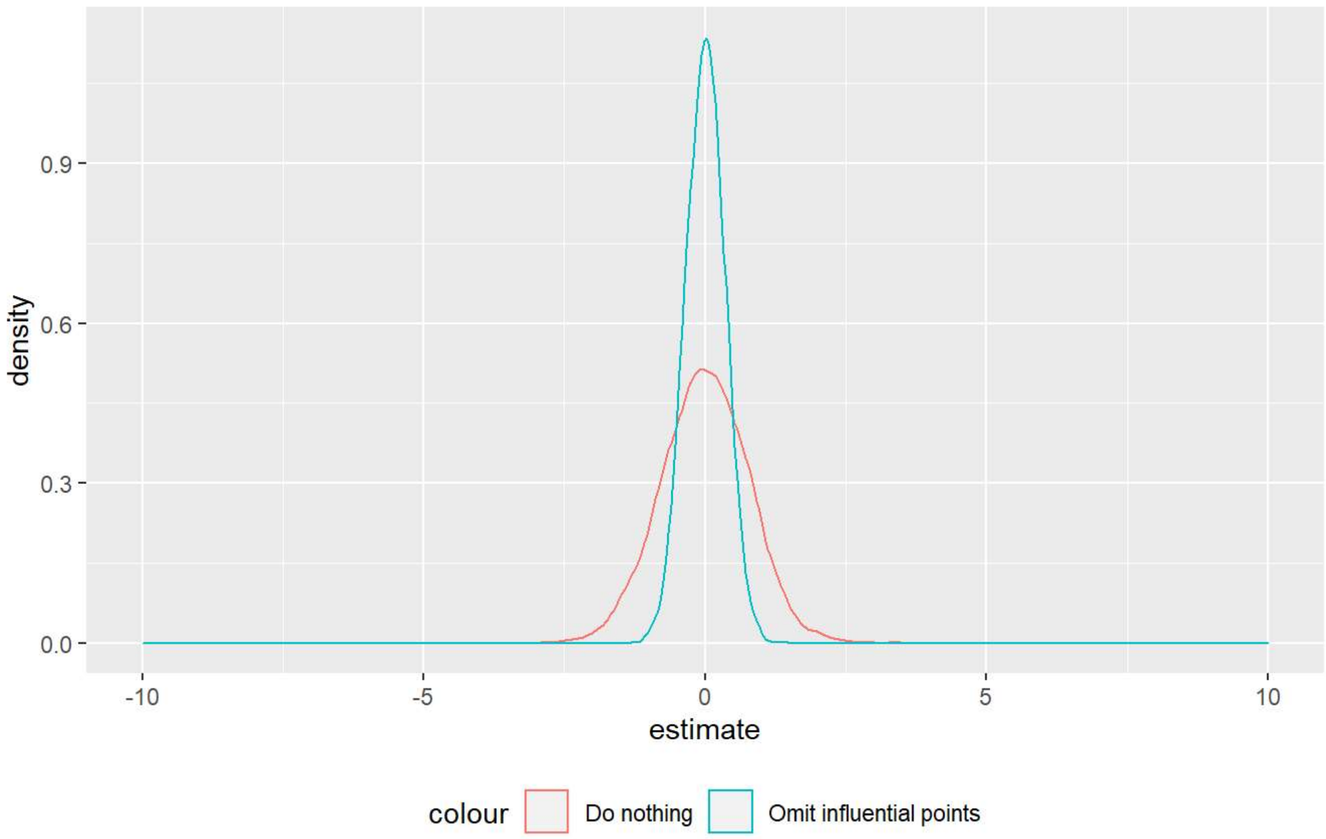
Sample Size = 500, Simulation Times = 5000, Heteroscedasticity

Var("Do Nothing") = 1.1224, Var("Omit influential points") = 0.2453



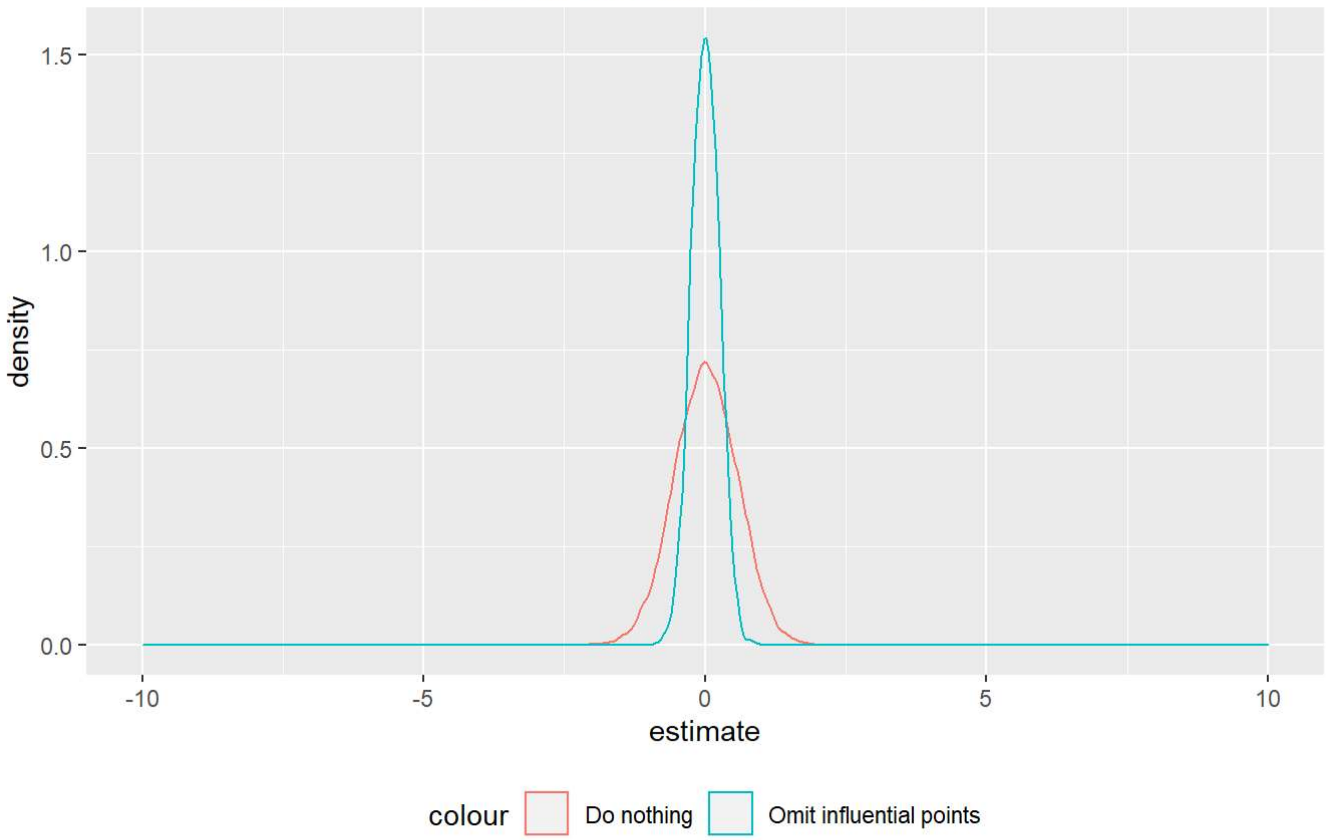
Sample Size = 1000, Simulation Times = 5000, Heteroscedasticity

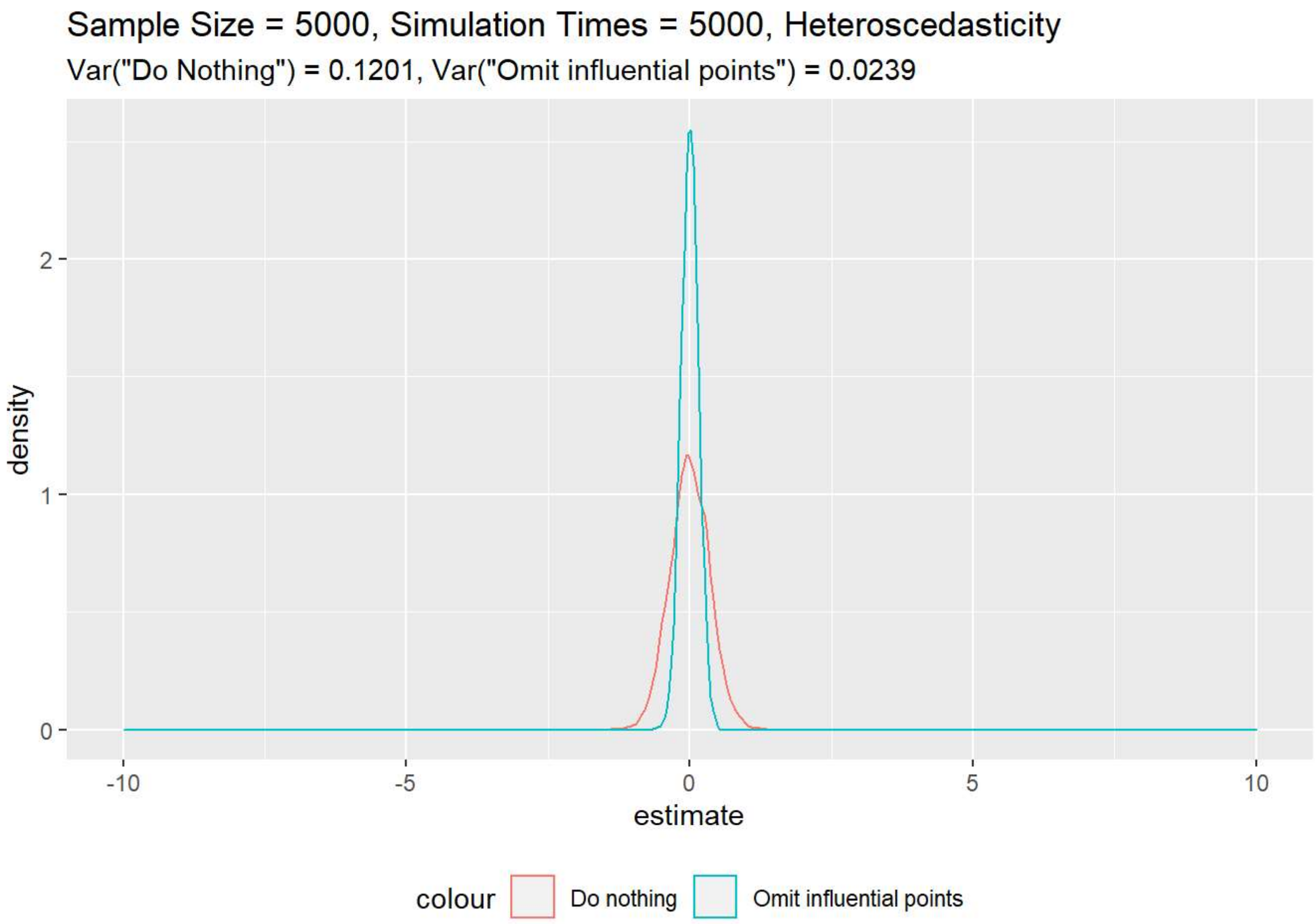
Var("Do Nothing") = 0.5901, Var("Omit influential points") = 0.1197

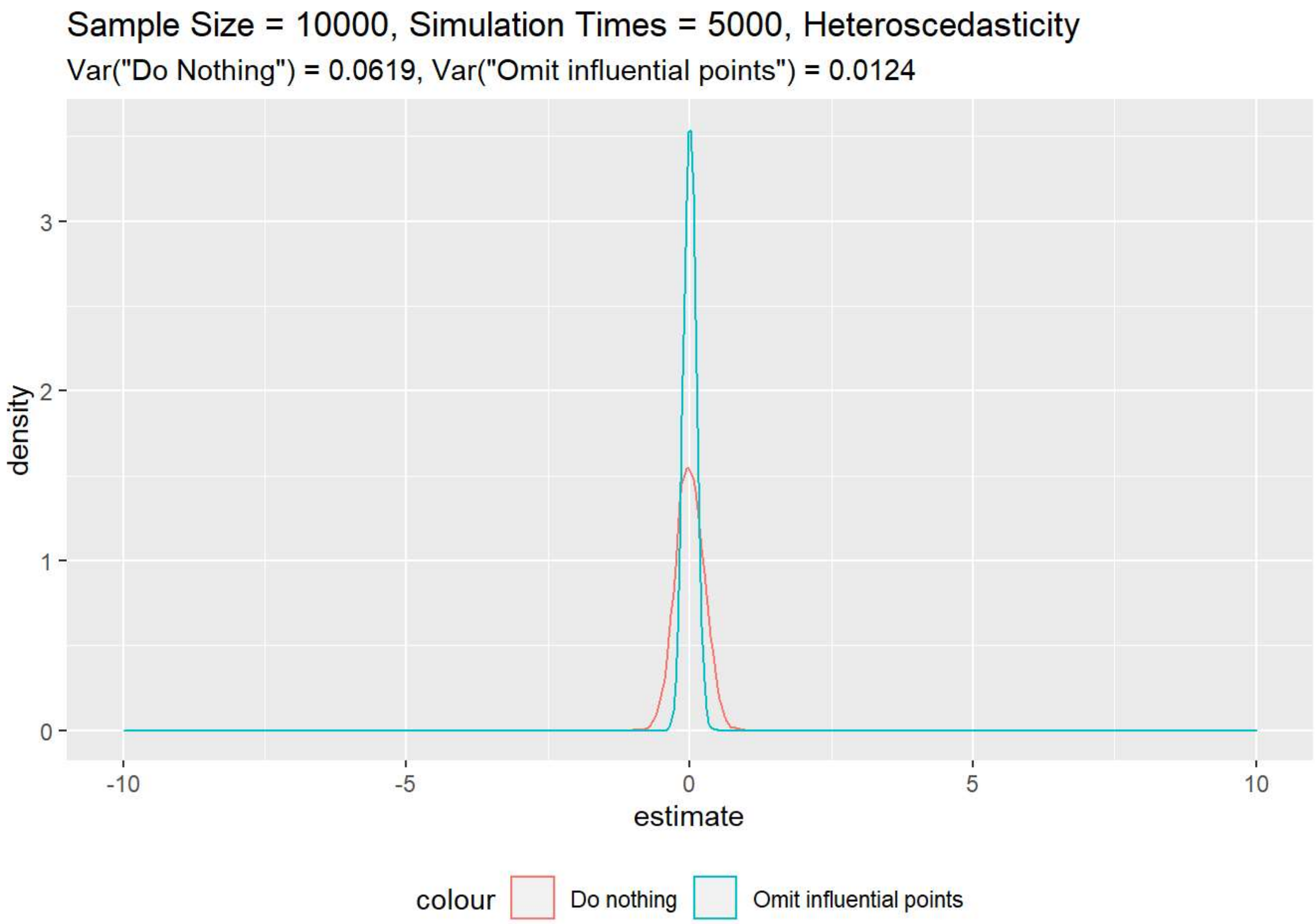


Sample Size = 2000, Simulation Times = 5000, Heteroscedasticity

Var("Do Nothing") = 0.305, Var("Omit influential points") = 0.0604







Two different DGPs

What if the mean is different?

Consider such a dataset consists of two groups of data,

the first group (95%) have such a DGP:

$$Y_t = \beta_0 + \varepsilon_t, \varepsilon_t \sim N(0, 10^2), t = 1, 2, .., T$$

the second group (5%) have such a DGP:

$$Y_t = \beta_0 + \varepsilon_t, \varepsilon_t \sim N(20, 10^2), t = 1, 2, .., T$$

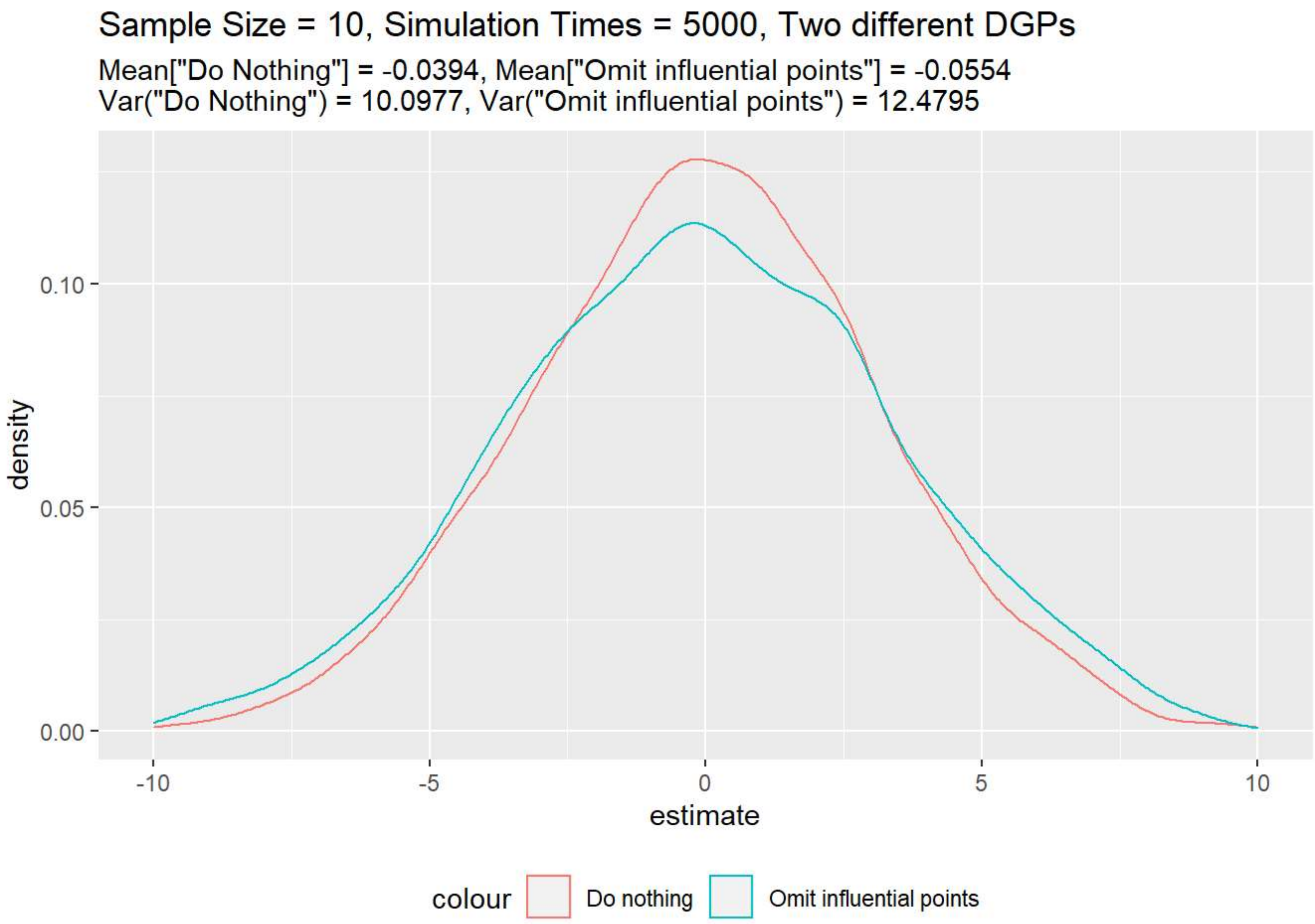
We could immediately know that by using the `lm` function, we will get the $\hat{\beta}_0$ around $20 \times 5\% + 0 \times 95\% = 1$. But what if we just want to estimate the β_0 for the first group, which is 0?

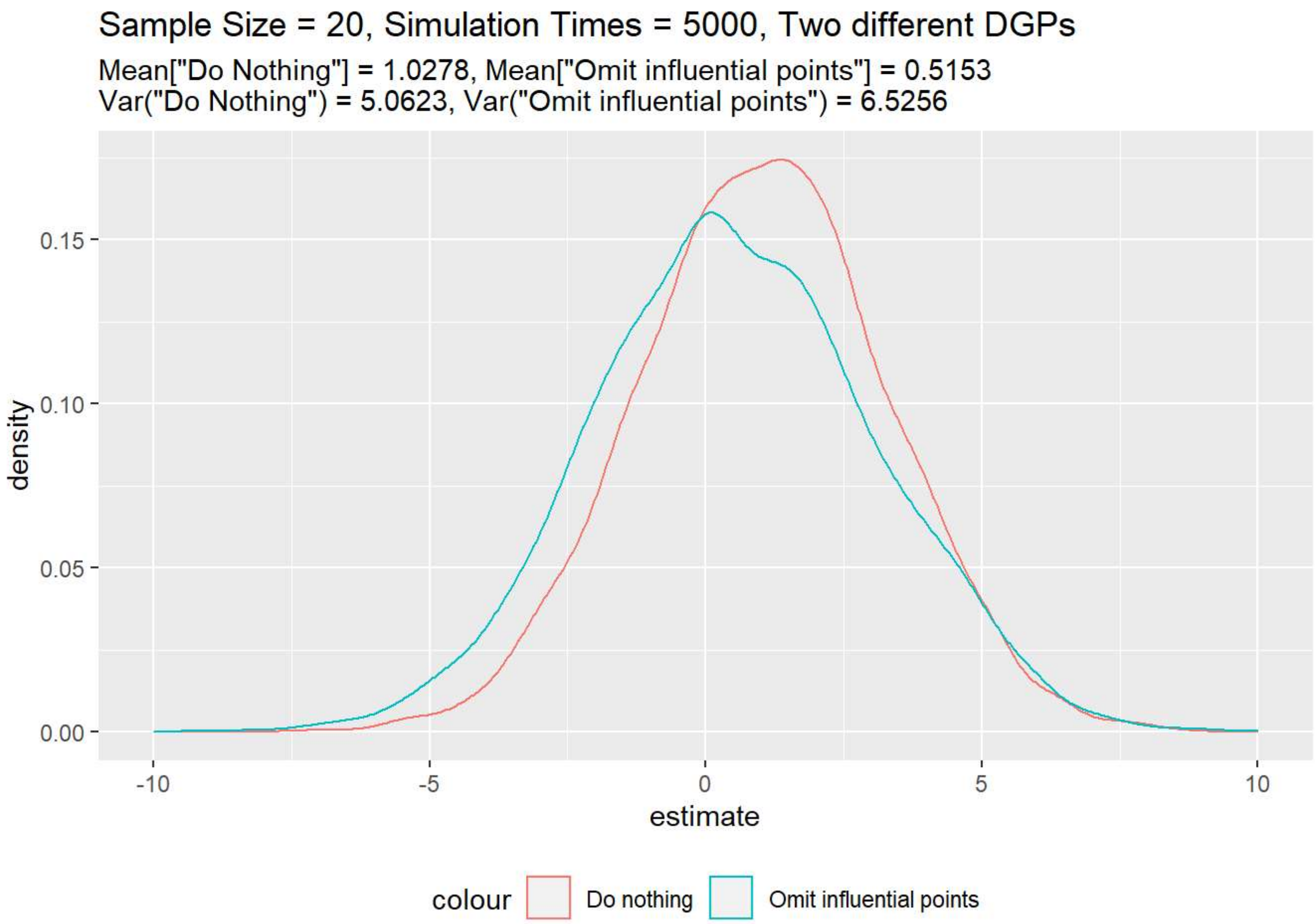
Let's see what will happen.

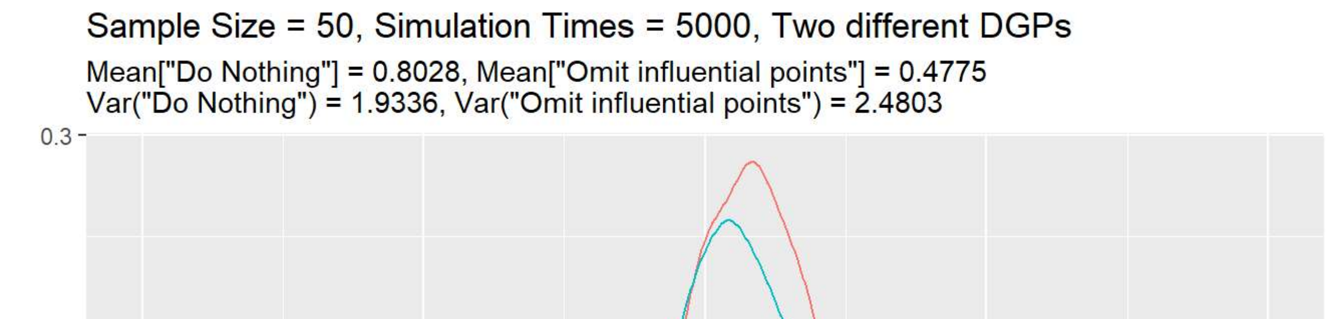
With no surprise, $\hat{\beta}_0$ will converge to 1. However, if we drop the influential points, we will end up getting the estimate $\hat{\beta}_{0,Cook}$ around 0.6, which is closer to 0.

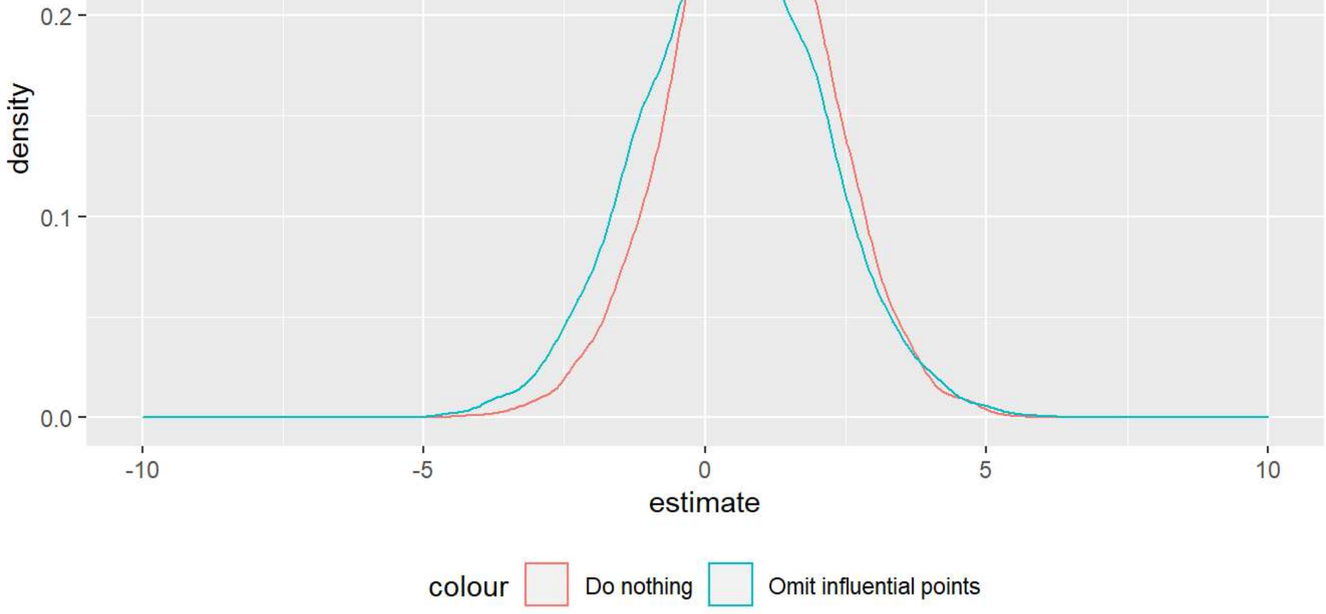
Therefore, if you believe there are a small proportion of noisy points that do not belong to your interesting group, try to drop influential points to reduce bias.

```r
mc3 <- function(T = 100, N = 100, cutoff = TRUE){

  estimate <- rep(NA, N)

  for (i in 1:N){

    y <- rnorm(round(T*0.95), sd = 10)
    y <- c(y, rnorm(T - round(T*0.95), mean = 20, sd = 10))
    mod <- lm(y~1)

    if (cutoff){
      y <- y[!cooks.distance(mod) > 4/T]
      mod <- lm(y~1)
    }

    estimate[i] <- as.numeric(mod$coefficients)
  }

  return(estimate)

}

set.seed(300)
N <- 5000

for (T in c(10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000)){
  estimate_d <- mc3(T = T, N = N, cutoff = TRUE)
  estimate <- mc3(T = T, N = N, cutoff = FALSE)

  vard <- var(estimate_d)
  varn <- var(estimate)
  ed <- mean(estimate_d)
  en <- mean(estimate)

  p <- ggplot() +
    geom_density(aes(estimate, col = "Do nothing")) +
    geom_density(aes(estimate_d, col = "Omit influential points")) +
    ggtitle(paste0("Sample Size = ", T, ", Simulation Times = ", N, ", Two different DGPs")) +
    xlim(c(-10, 10)) +
    theme(legend.position = "bottom") +
    labs(subtitle = paste0('Mean["Do Nothing"] = ',
                           round(en, 4),
                           ', ',
                           'Mean["Omit influential points"] = ',
                           round(ed, 4),
                           '\n',
                           'Var("Do Nothing") = ',
                           round(varn, 4),
                           ', Var("Omit influential points") = ',
                           round(vard, 4)))

  print(p)
  #ggsave(paste0("plots/HM-", T, ".jpeg"), plot = p)
}
```


Sample Size = 10, Simulation Times = 5000, Two different DGPs
Mean["Do Nothing"] = -0.0394, Mean["Omit influential points"] = -0.0554
Var("Do Nothing") = 10.0977, Var("Omit influential points") = 12.4795
density
0.10
0.05
0.00
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points

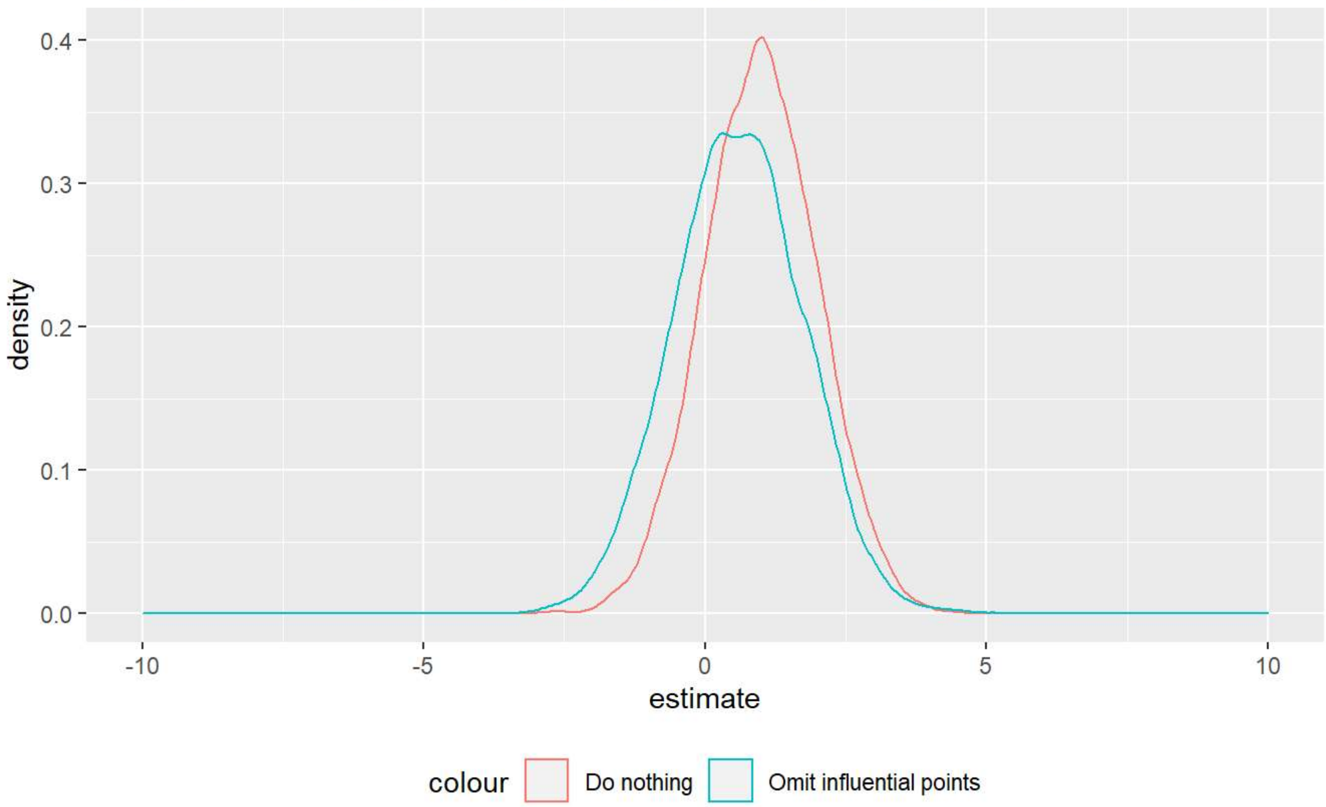

Sample Size = 20, Simulation Times = 5000, Two different DGPs
Mean["Do Nothing"] = 1.0278, Mean["Omit influential points"] = 0.5153
Var("Do Nothing") = 5.0623, Var("Omit influential points") = 6.5256
density
0.15
0.10
0.05
0.00
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points

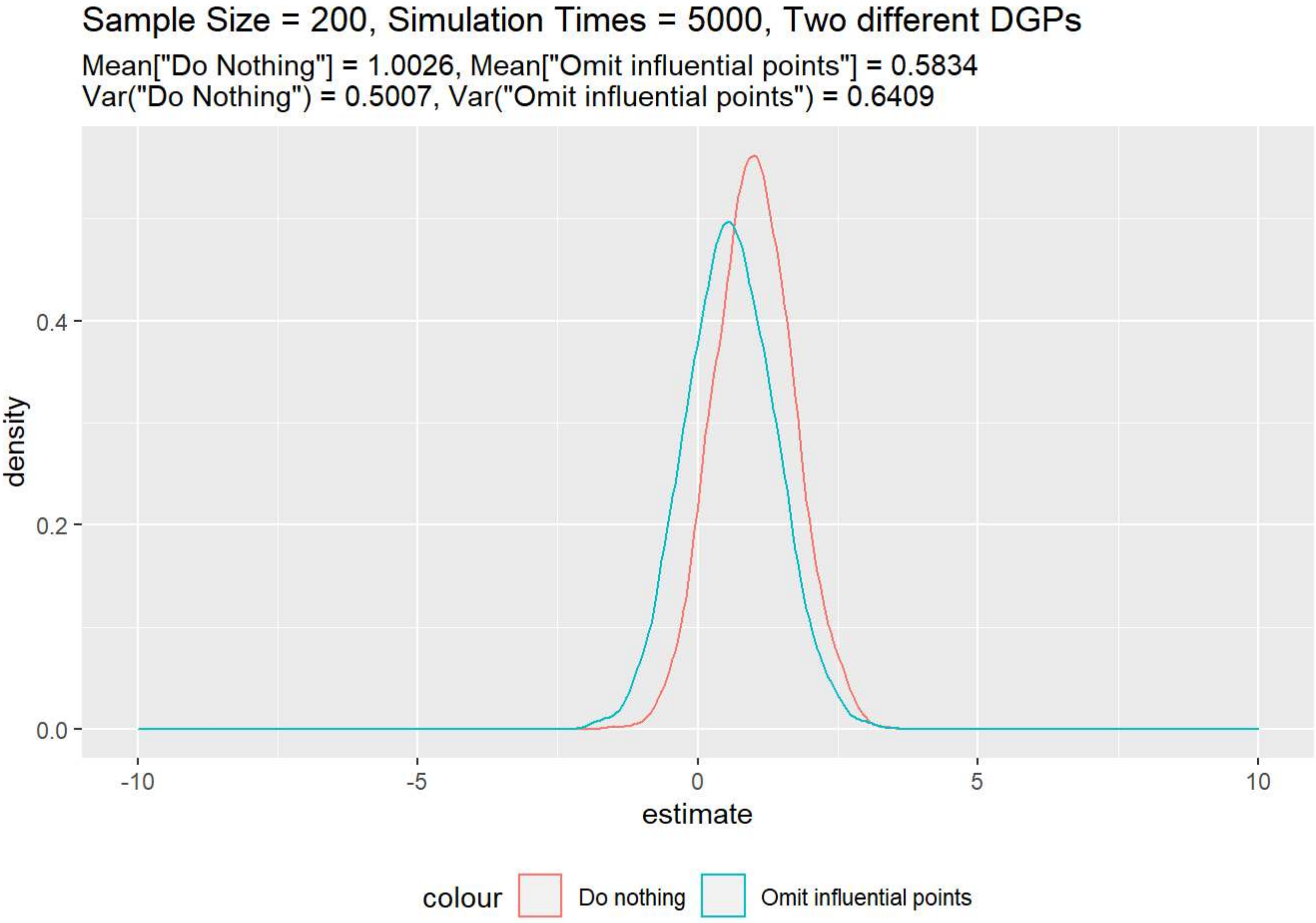

Sample Size = 50, Simulation Times = 5000, Two different DGPs
Mean["Do Nothing"] = 0.8028, Mean["Omit influential points"] = 0.4775
Var("Do Nothing") = 1.9336, Var("Omit influential points") = 2.4803
0.3

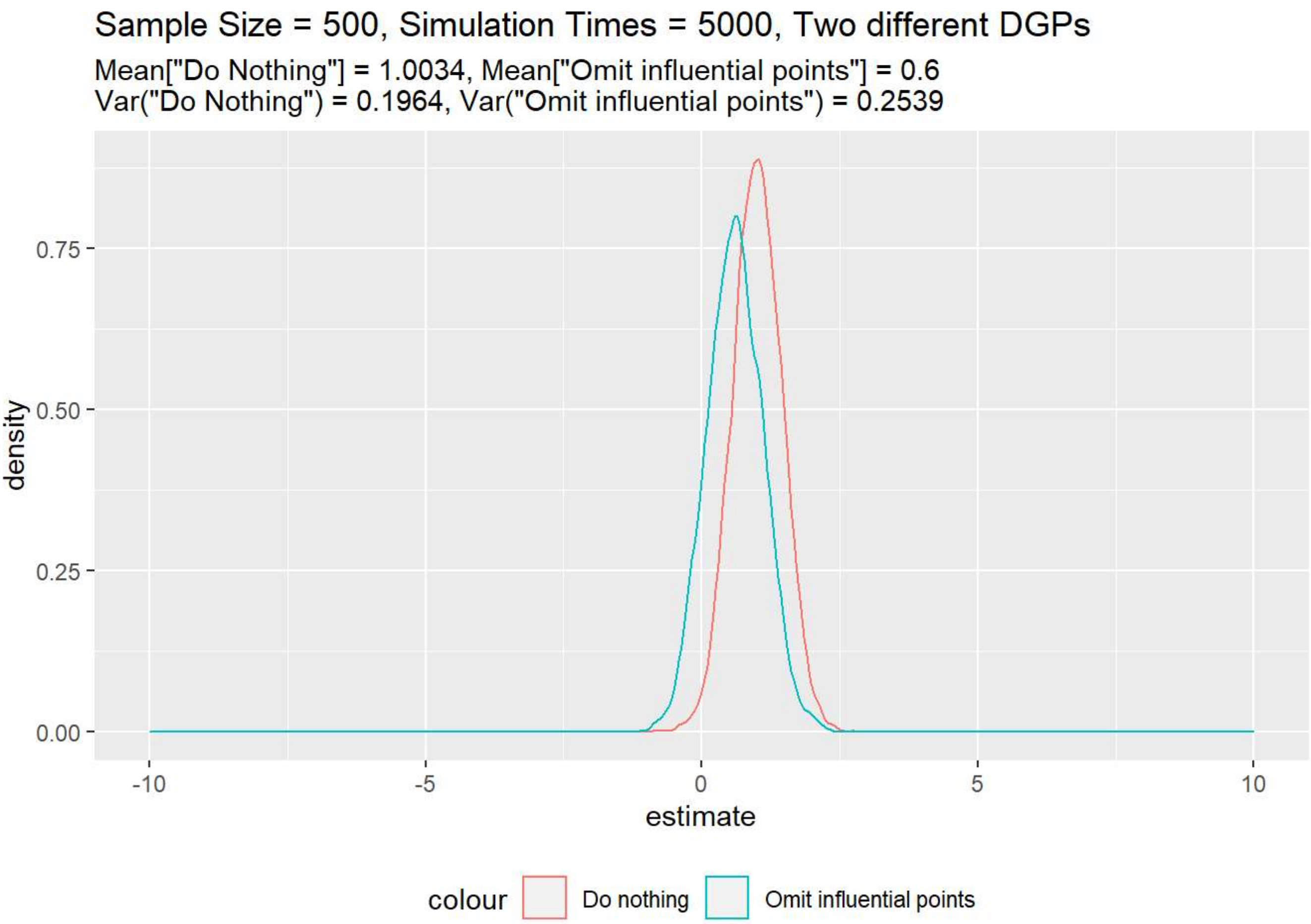


Sample Size = 100, Simulation Times = 5000, Two different DGPs

Mean["Do Nothing"] = 0.9938, Mean["Omit influential points"] = 0.587
Var("Do Nothing") = 1.0015, Var("Omit influential points") = 1.2918

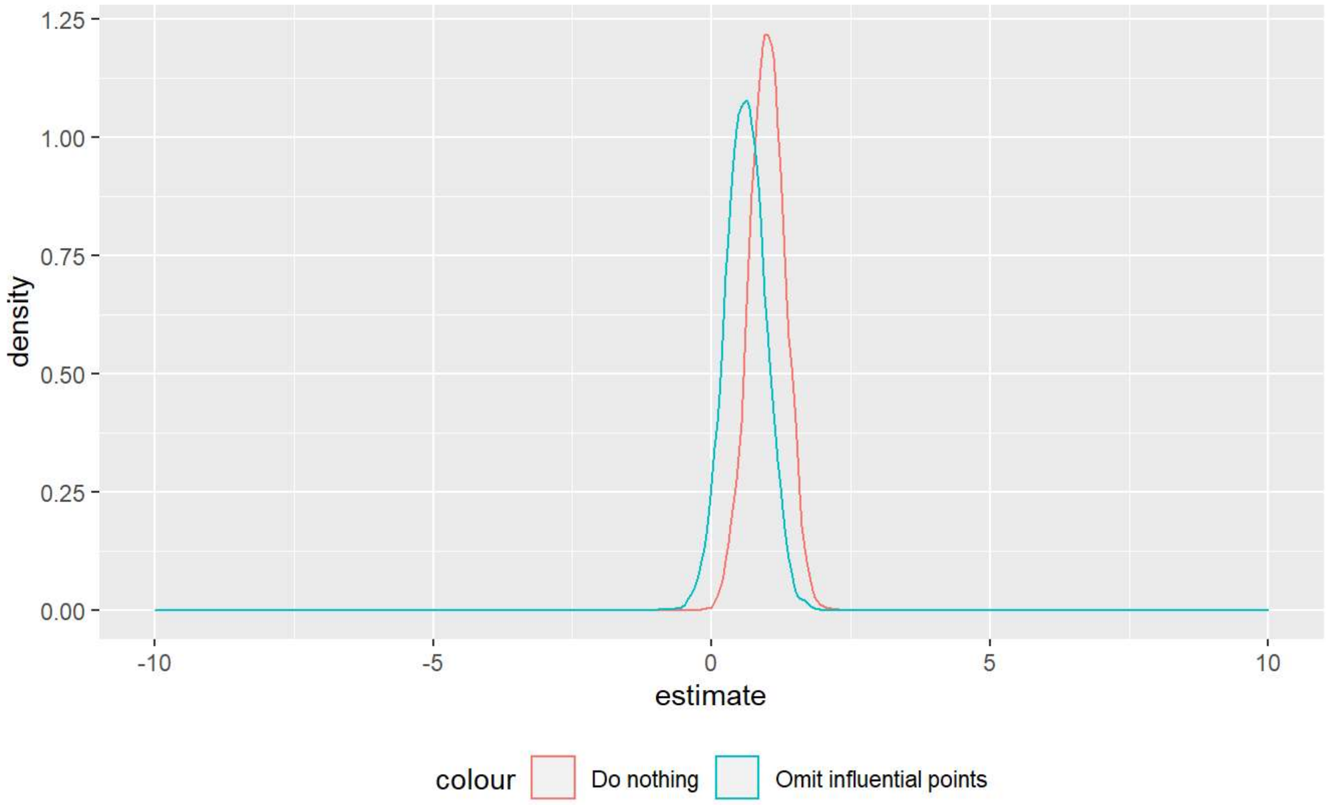



Sample Size = 200, Simulation Times = 5000, Two different DGPs
Mean["Do Nothing"] = 1.0026, Mean["Omit influential points"] = 0.5834
Var("Do Nothing") = 0.5007, Var("Omit influential points") = 0.6409
density
0.4
0.2
0.0
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points


Sample Size = 500, Simulation Times = 5000, Two different DGPs
Mean["Do Nothing"] = 1.0034, Mean["Omit influential points"] = 0.6
Var("Do Nothing") = 0.1964, Var("Omit influential points") = 0.2539
density
0.75
0.50
0.25
0.00
-10
-5
0
5
10
estimate
colour
Do nothing
Omit influential points

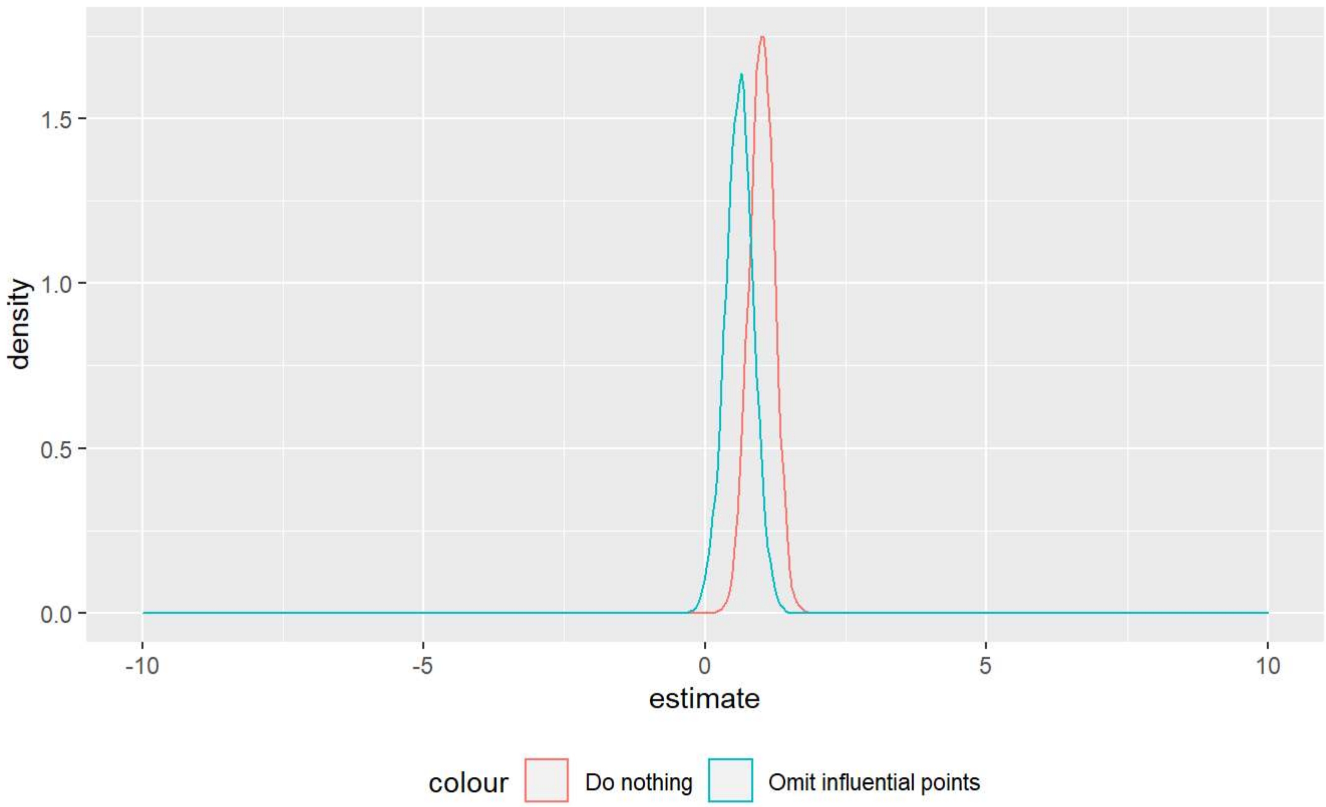
Sample Size = 1000, Simulation Times = 5000, Two different DGPs

Mean["Do Nothing"] = 0.9957, Mean["Omit influential points"] = 0.6061
Var("Do Nothing") = 0.1023, Var("Omit influential points") = 0.1303

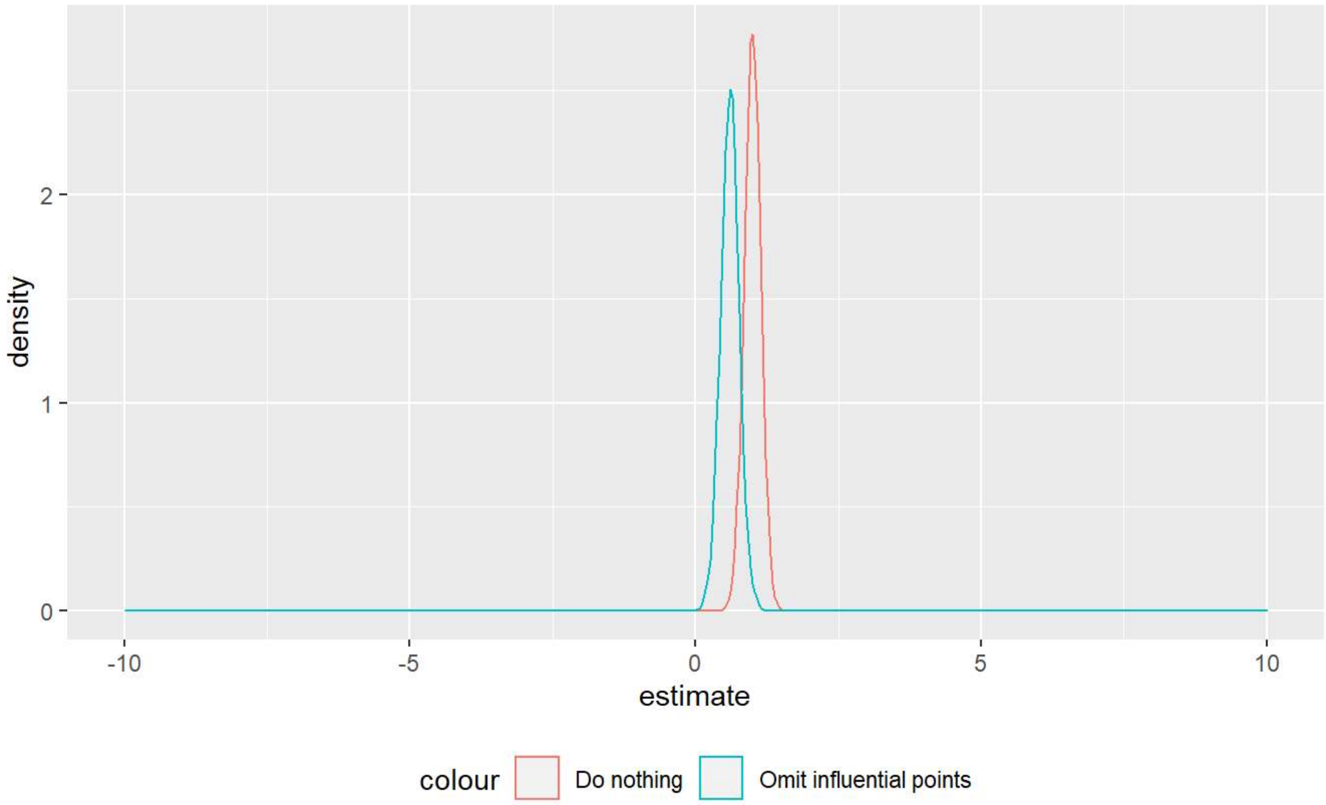


Sample Size = 2000, Simulation Times = 5000, Two different DGPs

Mean["Do Nothing"] = 1.0033, Mean["Omit influential points"] = 0.6051
Var("Do Nothing") = 0.0495, Var("Omit influential points") = 0.0628



Sample Size = 5000, Simulation Times = 5000, Two different DGPs
Mean["Do Nothing"] = 0.9966, Mean["Omit influential points"] = 0.5991
Var("Do Nothing") = 0.0204, Var("Omit influential points") = 0.0255



Sample Size = 10000, Simulation Times = 5000, Two different DGPs
Mean["Do Nothing"] = 0.9994, Mean["Omit influential points"] = 0.5979
Var("Do Nothing") = 0.0099, Var("Omit influential points") = 0.0121