Bias Exploration

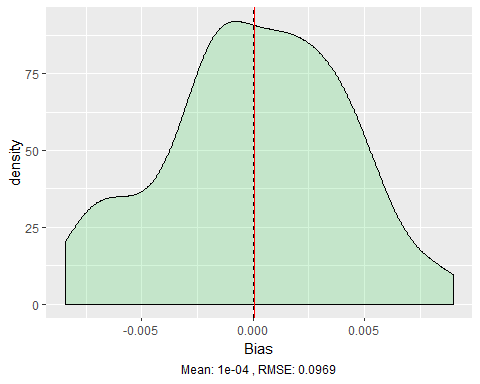
August 19, 2019

Here, I am exploring the distributions of the bias in our estimates to examine where bias enters the analysis.

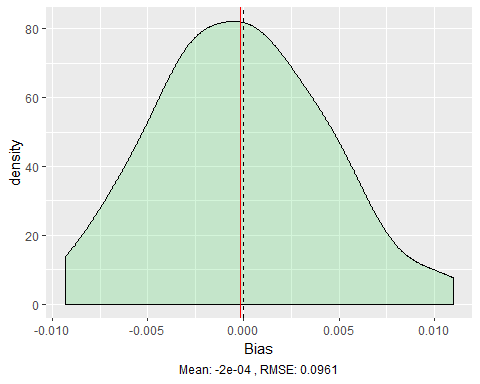
#calling the functions to be used  
source('defor\_DGP.R')  
source('quickmontey.R')  
source('quickmonteyit.R')  
source('DID\_bias.R')  
library(ggplot2)  
library(clubSandwich)  
library(matrixStats)  
library(ggplot2)  
library(plm)  
library(Metrics)

Starting as simple as possible: we only have one error term, an intercept, and a group effect. We also allow the outcome to vary between 0 and 1 across periods.

std\_a <- 0.1  
 std\_v <- 0.25  
 years <- 2  
 nobs <- 10000  
   
 b0 <- .3  
 b1 <- -.1  
   
 n <- 100  
 coeffmatrix <- matrix(nrow = n, ncol = 1)  
 for(i in 1:n){  
   
 ATT <- pnorm(b0+b1, 0, std\_v ) - pnorm(b0, 0, std\_v )  
   
 panels <- fabricate(  
 pixels = add\_level(N = nobs, a\_i = rnorm(N, 0, std\_a), treat = rbinom(N, 1, 0.5)),  
 year = add\_level(N = (years\*2), nest = FALSE),  
 obs = cross\_levels(  
 by = join(pixels, year),  
 post = ifelse(year > years, 1, 0),  
 v\_it = rnorm(N, 0, std\_v),  
 ystar = b0 + b1\*treat + v\_it,  
 y = ifelse(ystar > 0, 1, 0)  
 )  
 )  
   
 coeffmatrix[i,1] <- lm(y ~ treat,   
 data = panels  
 )$coefficients[2] - ATT  
   
 }  
   
 did\_coeff <- as.data.frame(coeffmatrix)  
 actual <- rep(ATT, times = n)  
   
 ggplot() +  
 geom\_density(data = did\_coeff , aes(x = V1), alpha = .2, fill="#29CD44")+  
 geom\_vline(data = did\_coeff, xintercept = mean(did\_coeff$V1), color = 'red')+  
 geom\_vline(data = did\_coeff, xintercept = 0, linetype = "dashed")+  
 theme(plot.caption = element\_text(hjust = 0.5))+  
 labs(x= "Bias", caption = paste("Mean:", round(mean(did\_coeff$V1), digits = 4),  
 ", RMSE:", round(rmse(actual, did\_coeff$V1), digits = 4))   
 )

 There is no bias with only the interrcept, group effect, and random error term. Next, we introduce the individual level error terms.

std\_a <- 0.1  
std\_v <- 0.25  
 years <- 2  
 nobs <- 10000  
   
 b0 <- .3  
 b1 <- -.1  
   
 n <- 100  
 coeffmatrix <- matrix(nrow = n, ncol = 1)  
  
   
 for(i in 1:n){  
   
 ATT <- pnorm(b0+b1, 0, (std\_a^2+std\_v^2)^(1/2) ) - pnorm(b0, 0, (std\_a^2+std\_v^2)^(1/2) )  
   
 panels <- fabricate(  
 pixels = add\_level(N = nobs, a\_i = rnorm(N, 0, std\_a), treat = rbinom(N, 1, 0.5)),  
 year = add\_level(N = (years\*2), nest = FALSE),  
 obs = cross\_levels(  
 by = join(pixels, year),  
 post = ifelse(year > years, 1, 0),  
 v\_it = rnorm(N, 0, std\_v),  
 ystar = b0 + b1\*treat + a\_i + v\_it,  
 y = ifelse(ystar > 0, 1, 0)  
 )  
 )  
   
 coeffmatrix[i,1] <- lm(y ~ treat,   
 data = panels  
 )$coefficients[2] - ATT  
   
 }  
   
 did\_coeff <- as.data.frame(coeffmatrix)  
 actual <- rep(ATT, times = n)  
   
 ggplot() +  
 geom\_density(data = did\_coeff , aes(x = V1), alpha = .2, fill="#29CD44")+  
 geom\_vline(data = did\_coeff, xintercept = mean(did\_coeff$V1), color = 'red')+  
 geom\_vline(data = did\_coeff, xintercept = 0, linetype = "dashed")+  
 theme(plot.caption = element\_text(hjust = 0.5))+  
 labs(x= "Bias", caption = paste("Mean:", round(mean(did\_coeff$V1), digits = 4),  
 ", RMSE:", round(rmse(actual, did\_coeff$V1), digits = 4))   
 )

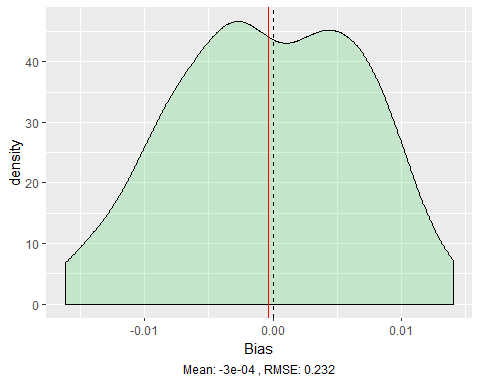


Introducing the individual errors to the simple model does not create any bias.

Now, I will look at estimates from the full DID model, but still allowing the outcome to vary between 0 and 1 across periods. I’ll start introducing the effects one at a time.

Here, we run the simulation without any group or time effects. There is only an intercept and treatment effect.

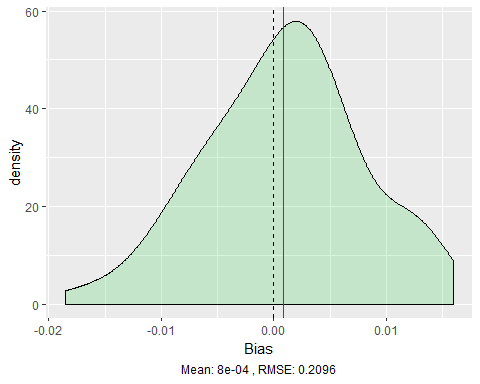
#distribution of bias in estimates using y as the outcome, allowed to switch between 1 and 0 across periods  
y\_nogrouptrend <- quickmontey(100, 10000, 3, .05, 0, 0, -.16)  
y\_nogrouptrend



When we run the simulation with only an intercept and treatment effect and allow the outcome to transition between periods, there is no bias.

Here, we add in a group effect to the above model.

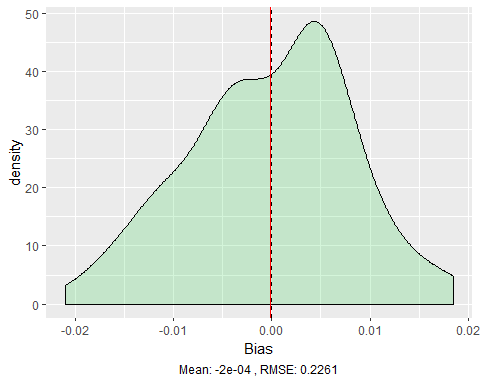
#let's add in a group effect to see if bias enters using y as the outcome  
y\_notrend <- quickmontey(100, 10000, 3, .05, -.1, 0, -.16)  
y\_notrend



Still no bias after adding in group effect with outcome switching across periods.

Here, instead of a group effect, we add in a time effect. So, there is a time effect, intercept, and treatment effect.

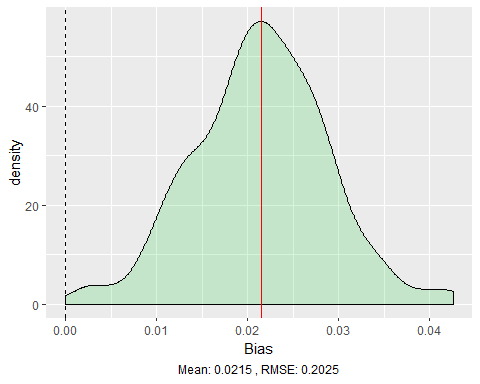
#using time effect to see if bias enters using y as the outcome  
y\_nogroup <- quickmontey(100, 10000, 3, .05, 0, 0.1, -.16)  
y\_nogroup



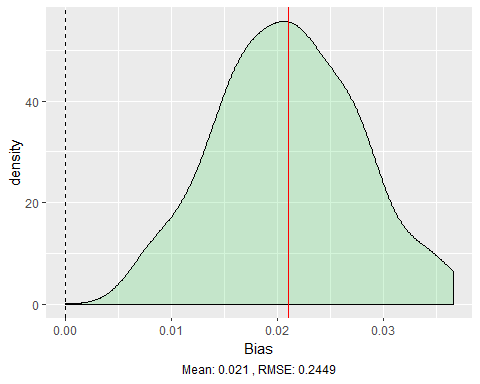
Still no bias with time effect but no group effect, still allowing outcome to switch across periods.

Now, we include an intercept, group effect, time effect, and treatment effect.

#let's add in a time and group effect to see if bias enters using y as the outcome  
ypos <- quickmontey(100, 10000, 3, .05, -0.25, 0.2, .16)  
yneg <- quickmontey(100, 10000, 3, .05, -0.25, 0.2, -.16)  
ypos



yneg

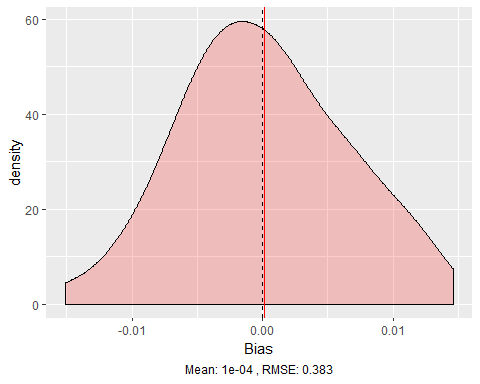


Bias seems to enter at this point when including both a group and time effect, still allowing the outcome to switch across periods. It is interesting to note that the bias does not depend on the sign of the treatment effect. It may depend on the interaction of the other effects, which will be explored later.

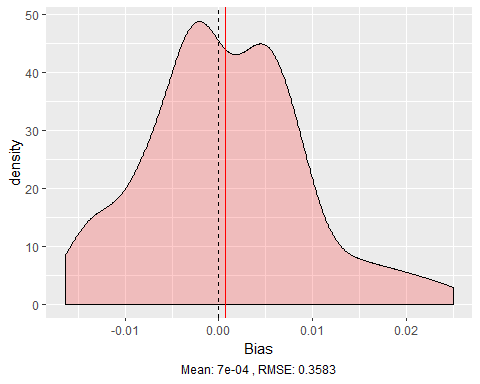
Now let’s transition to dropping deforested observations for the periods after they are first observed as deforested to see if the bias enters at the same point.

Lets start with a basic model to make sure the errors and coefficients are ok

std\_a <- 0.1  
 std\_v <- 0.25  
 years <- 2  
 nobs <- 10000  
   
 b0 <- 0.12  
 b1 <- -.25  
   
 n <- 100  
 coeffmatrix <- matrix(nrow = n, ncol = 1)  
  
   
 for(i in 1:n){  
   
 ATT <- pnorm(b0+b1, 0, std\_v ) - pnorm(b0, 0, std\_v )  
   
 panels <- fabricate(  
 pixels = add\_level(N = nobs, a\_i = rnorm(N, 0, std\_a), treat = rbinom(N, 1, 0.5)),  
 year = add\_level(N = (years\*2), nest = FALSE),  
 obs = cross\_levels(  
 by = join(pixels, year),  
 post = ifelse(year > years, 1, 0),  
 v\_it = rnorm(N, 0, std\_v),  
 ystar = b0 + b1\*treat + v\_it,  
 y = ifelse(ystar > 0, 1, 0)  
 )  
 )  
   
   
 #need to determine which year deforestation occurred  
 year\_df <- subset(panels, select = c(pixels, year, y))  
 #year\_df <- melt(year\_df, id.vars = c("pixels", "y\_it"), value.name = "year")  
 year\_df <- dcast(year\_df, pixels ~ year )  
 rownames(year\_df) <- year\_df$pixels  
 year\_df <- subset(year\_df, select = -c(pixels))  
   
 #creating variable for the year a pixel is deforested  
 not\_defor <- rowSums(year\_df)<1 \*1  
 defor\_year <- max.col(year\_df, ties.method = "first")   
 defor\_df <- transform(year\_df, defor\_year = ifelse(not\_defor==1, years\*2+1, defor\_year))  
 defor\_df <- tibble::rownames\_to\_column(defor\_df)  
 names(defor\_df)[1] <- paste("pixels")  
 defor\_df <- subset(defor\_df, select = c(pixels, defor\_year))  
 panels <- merge(defor\_df, panels, by = "pixels")  
   
   
 # creating three outcome variables for each possible situation  
 ### y: allows the outcome to switch between 0 and 1 across years  
 ### y\_it: outcome is dropped in years after pixel is first deforested  
 ### defor: outcome is set to 1 in each year after the pixel is deforested  
 panels$year <- as.numeric(panels$year)  
 panels$indic <- (panels$year - panels$defor\_year)  
 panels$y\_it <- ifelse(panels$indic > 0 , NA, panels$y)  
 panels$defor <- ifelse(panels$indic > 0 , 1, panels$y\_it)  
 panels <- subset(panels, select = -c(indic))  
   
 coeffmatrix[i,1] <- lm(y\_it ~ treat,   
 data = panels  
 )$coefficients[2] - ATT  
   
 }  
   
 did\_coeff <- as.data.frame(coeffmatrix)  
 actual <- rep(ATT, times = n)  
   
 ggplot() +  
 geom\_density(data = did\_coeff , aes(x = V1), alpha = .2, fill="red")+  
 geom\_vline(data = did\_coeff, xintercept = mean(did\_coeff$V1), color = 'red')+  
 geom\_vline(data = did\_coeff, xintercept = 0, linetype = "dashed")+  
 theme(plot.caption = element\_text(hjust = 0.5))+  
 labs(x= "Bias", caption = paste("Mean:", round(mean(did\_coeff$V1), digits = 4),  
 ", RMSE:", round(rmse(actual, did\_coeff$V1), digits = 4))   
 )

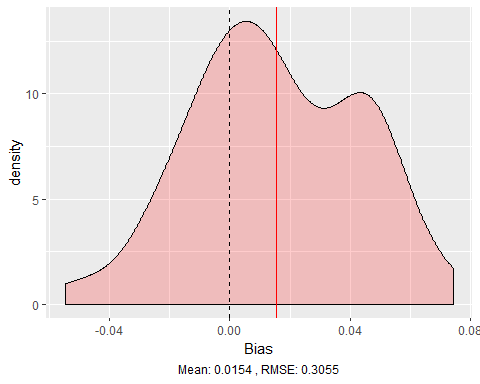


std\_a <- 0.1  
 std\_v <- 0.25  
 years <- 2  
 nobs <- 10000  
   
 b0 <- 0.12  
 b1 <- -.25  
   
 n <- 100  
 coeffmatrix <- matrix(nrow = n, ncol = 1)  
  
   
 for(i in 1:n){  
   
 ATT <- pnorm(b0+b1, 0, (std\_a^2+std\_v^2)^(1/2) ) - pnorm(b0, 0, (std\_a^2+std\_v^2)^(1/2) )  
   
 panels <- fabricate(  
 pixels = add\_level(N = nobs, a\_i = rnorm(N, 0, std\_a), treat = rbinom(N, 1, 0.5)),  
 year = add\_level(N = (years\*2), nest = FALSE),  
 obs = cross\_levels(  
 by = join(pixels, year),  
 post = ifelse(year > years, 1, 0),  
 v\_it = rnorm(N, 0, std\_v),  
 ystar = b0 + b1\*treat + a\_i + v\_it,  
 y = ifelse(ystar > 0, 1, 0)  
 )  
 )  
   
   
 #need to determine which year deforestation occurred  
 year\_df <- subset(panels, select = c(pixels, year, y))  
 #year\_df <- melt(year\_df, id.vars = c("pixels", "y\_it"), value.name = "year")  
 year\_df <- dcast(year\_df, pixels ~ year )  
 rownames(year\_df) <- year\_df$pixels  
 year\_df <- subset(year\_df, select = -c(pixels))  
   
 #creating variable for the year a pixel is deforested  
 not\_defor <- rowSums(year\_df)<1 \*1  
 defor\_year <- max.col(year\_df, ties.method = "first")   
 defor\_df <- transform(year\_df, defor\_year = ifelse(not\_defor==1, years\*2+1, defor\_year))  
 defor\_df <- tibble::rownames\_to\_column(defor\_df)  
 names(defor\_df)[1] <- paste("pixels")  
 defor\_df <- subset(defor\_df, select = c(pixels, defor\_year))  
 panels <- merge(defor\_df, panels, by = "pixels")  
   
   
 # creating three outcome variables for each possible situation  
 ### y: allows the outcome to switch between 0 and 1 across years  
 ### y\_it: outcome is dropped in years after pixel is first deforested  
 ### defor: outcome is set to 1 in each year after the pixel is deforested  
 panels$year <- as.numeric(panels$year)  
 panels$indic <- (panels$year - panels$defor\_year)  
 panels$y\_it <- ifelse(panels$indic > 0 , NA, panels$y)  
 panels$defor <- ifelse(panels$indic > 0 , 1, panels$y\_it)  
 panels <- subset(panels, select = -c(indic))  
   
 coeffmatrix[i,1] <- lm(y\_it ~ treat,   
 data = panels  
 )$coefficients[2] - ATT  
   
 }  
   
 did\_coeff <- as.data.frame(coeffmatrix)  
 actual <- rep(ATT, times = n)  
   
 ggplot() +  
 geom\_density(data = did\_coeff , aes(x = V1), alpha = .2, fill="red")+  
 geom\_vline(data = did\_coeff, xintercept = mean(did\_coeff$V1), color = 'red')+  
 geom\_vline(data = did\_coeff, xintercept = 0, linetype = "dashed")+  
 theme(plot.caption = element\_text(hjust = 0.5))+  
 labs(x= "Bias", caption = paste("Mean:", round(mean(did\_coeff$V1), digits = 4),  
 ", RMSE:", round(rmse(actual, did\_coeff$V1), digits = 4))   
 )



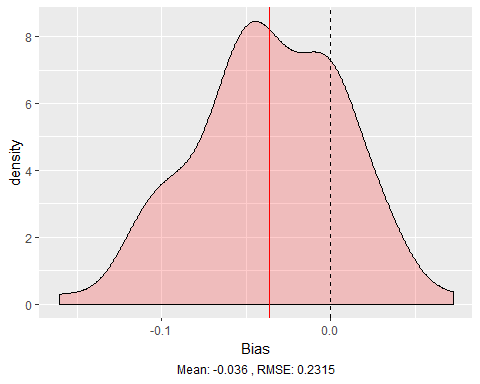
Using the full model, we start without a group or time effect. There is only an intercept and treatment effect to begin.

yit\_nogrouptrend <- quickmonteyit(100, 10000, 3, .12, 0, 0, -.2)  
yit\_nogrouptrend



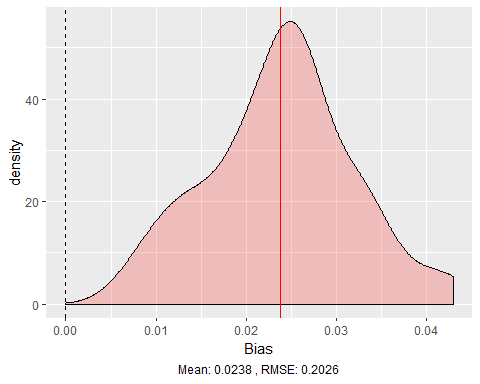
There is maybe a bit of bias with only the intercept and treatment effect in the model. Next, we introduce the group effect.

yit\_nogroup <- quickmonteyit(100, 10000, 3, .12, 0.1, 0, -.2)  
yit\_nogroup



There is bias with only the group effect. Now, instead of the group effect, we introduce the time effect.

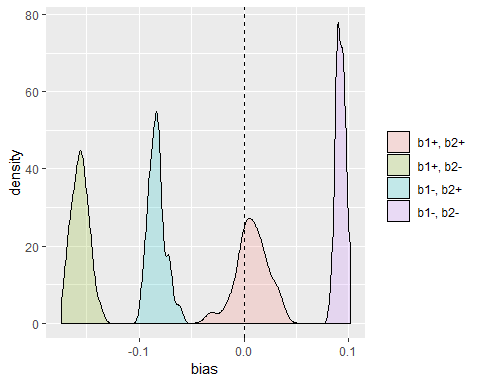
yit\_notrend <- quickmonteyit(50, 10000, 3, -.2, 0, .1, -.15)  
yit\_notrend



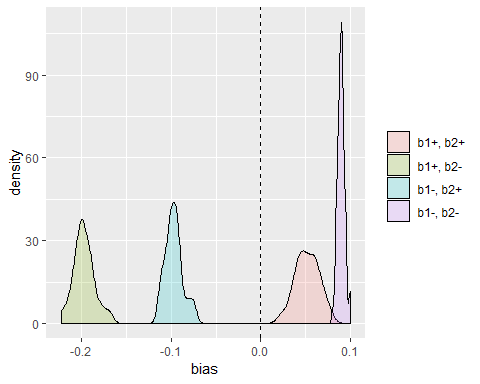
Here, there is bias with a time effect introduced but no group effect.

Let’s introduce both the time and group effects. This is where the bias entered in the case where the outcome could switch across periods. Here, the graphs show how the bias depends on the sign of the group and time effects. The first graph represents a treatment effect of 0. The second has a positive treatment effect, and the thirds has a negative treatment effect.

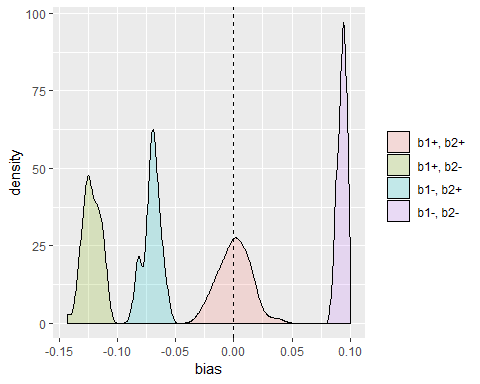
yit\_noeffect <- DID\_bias(50, 10000, 3, -.25, 0.2, 0.2, 0)  
yit\_postreat <- DID\_bias(50, 10000, 3, -.25, 0.2, 0.2, .16)  
yit\_negtreat <- DID\_bias(50, 10000, 3, -.25, 0.2, 0.2, -.16)  
  
yit\_noeffect



yit\_postreat



yit\_negtreat



It seems that the bias is positive when the time and group effects have the same sign and negative when the signs differ. This is regardless of the sign of the treatment effect.

I reduce the sd of the individual error term to see if the bias leaves with only a group or time effect.

std\_a <- 0.01  
 std\_v <- 0.25  
 years <- 2  
 nobs <- 10000  
   
 b0 <- 0.12  
 b1 <- -.25  
 b2 <- 0  
 b3 <- .1  
  
   
 n <- 100  
 coeffmatrix <- matrix(nrow = n, ncol = 1)  
  
   
 for(i in 1:n){  
   
 ATT <- pnorm(b0+b1+b2+b3, 0, (std\_a^2+std\_v^2)^(1/2) ) - pnorm(b0+b1+b2, 0, (std\_a^2+std\_v^2)^(1/2) )  
   
 panels <- fabricate(  
 pixels = add\_level(N = nobs, a\_i = rnorm(N, 0, std\_a), treat = rbinom(N, 1, 0.5)),  
 year = add\_level(N = (years\*2), nest = FALSE),  
 obs = cross\_levels(  
 by = join(pixels, year),  
 post = ifelse(year > years, 1, 0),  
 v\_it = rnorm(N, 0, std\_v),  
 ystar = b0 + b1\*treat + b2\*post +b3\*treat\*post + a\_i + v\_it,  
 y = ifelse(ystar > 0, 1, 0)  
 )  
 )  
   
   
 #need to determine which year deforestation occurred  
 year\_df <- subset(panels, select = c(pixels, year, y))  
 #year\_df <- melt(year\_df, id.vars = c("pixels", "y\_it"), value.name = "year")  
 year\_df <- dcast(year\_df, pixels ~ year )  
 rownames(year\_df) <- year\_df$pixels  
 year\_df <- subset(year\_df, select = -c(pixels))  
   
 #creating variable for the year a pixel is deforested  
 not\_defor <- rowSums(year\_df)<1 \*1  
 defor\_year <- max.col(year\_df, ties.method = "first")   
 defor\_df <- transform(year\_df, defor\_year = ifelse(not\_defor==1, years\*2+1, defor\_year))  
 defor\_df <- tibble::rownames\_to\_column(defor\_df)  
 names(defor\_df)[1] <- paste("pixels")  
 defor\_df <- subset(defor\_df, select = c(pixels, defor\_year))  
 panels <- merge(defor\_df, panels, by = "pixels")  
   
   
 # creating three outcome variables for each possible situation  
 ### y: allows the outcome to switch between 0 and 1 across years  
 ### y\_it: outcome is dropped in years after pixel is first deforested  
 ### defor: outcome is set to 1 in each year after the pixel is deforested  
 panels$year <- as.numeric(panels$year)  
 panels$indic <- (panels$year - panels$defor\_year)  
 panels$y\_it <- ifelse(panels$indic > 0 , NA, panels$y)  
 panels$defor <- ifelse(panels$indic > 0 , 1, panels$y\_it)  
 panels <- subset(panels, select = -c(indic))  
   
 coeffmatrix[i,1] <- lm(y\_it ~ treat\*post,   
 data = panels  
 )$coefficients[4] - ATT  
   
 }  
   
 did\_coeff <- as.data.frame(coeffmatrix)  
 actual <- rep(ATT, times = n)  
   
 ggplot() +  
 geom\_density(data = did\_coeff , aes(x = V1), alpha = .2, fill="red")+  
 geom\_vline(data = did\_coeff, xintercept = mean(did\_coeff$V1), color = 'red')+  
 geom\_vline(data = did\_coeff, xintercept = 0, linetype = "dashed")+  
 theme(plot.caption = element\_text(hjust = 0.5))+  
 labs(x= "Bias", caption = paste("Mean:", round(mean(did\_coeff$V1), digits = 4),  
 ", RMSE:", round(rmse(actual, did\_coeff$V1), digits = 4))   
 )

