Time-varying treatment effect dgp

DGP

Data Size

```
n = 100
t1 = 40
t0 = 21
p = 1
```

Individual effect

```
# alpha_i ~ N(mu_a, sig_a)
mu_a = 0
sig_a = 1
alpha = rnorm(n, mu_a, sig_a)
```

Time fixed effect

```
# gamma_t ~ N(mu_g, sig_g)
mu_g = 1
sig_g = 1
gamma = rnorm(t1, mu_g, sig_g)
```

Time-varying (lagged) Treatment effect

```
x = as.matrix(rnorm(n))
tau <- function(x){a = (x + 1.5)^2; 5*sqrt(a) + sin(5*a) + 1}
tau_mat <- matrix(0, n, t1-t0+1)
tau_mat[,1] <- tau(x)
for (i in 2:(t1-t0+1)){
  tau_mat[,i] <- 0.9*tau_mat[,i-1]
}
```

Treatment

```
z = rbinom(n, 1, 0.5)
```

Error term

```
eps = matrix(rnorm(n*t1, 0, 0.2), nrow = n, ncol = t1)
```

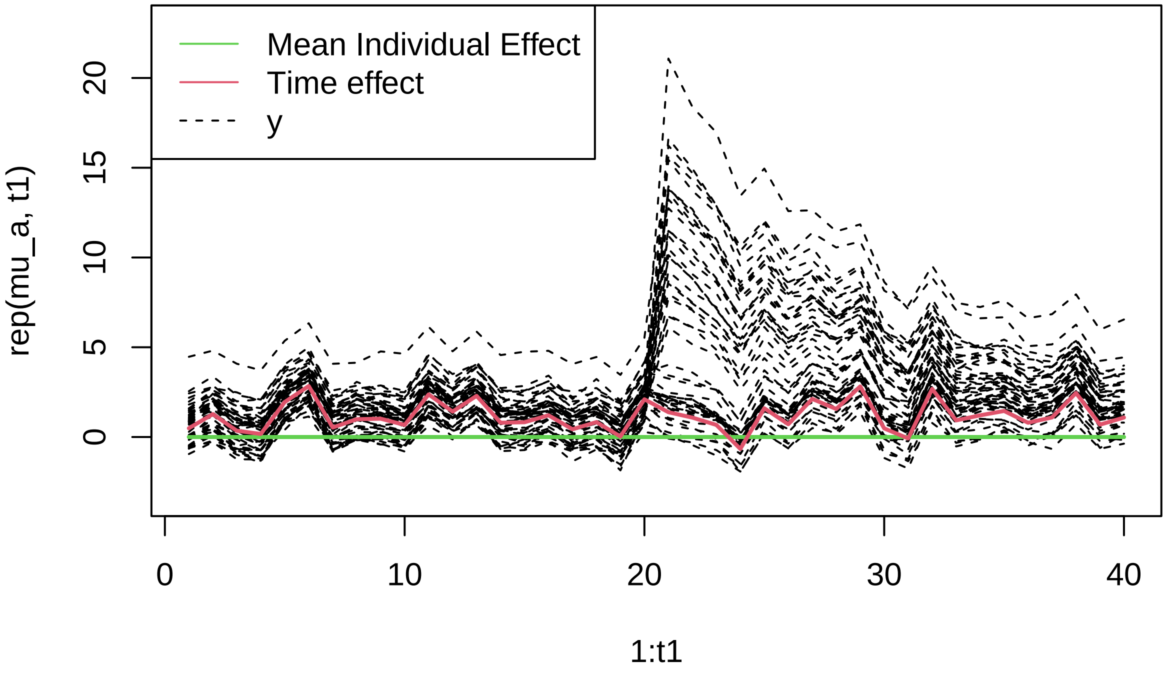
Generate observations

```
y0 = y1 = y = matrix(0, nrow = n, ncol = t1)
for (i in 1:n){
  y0[i,] = y0[i,] + alpha[i]
}
for (j in 1:t1){
  y0[,j] = y0[,j] + gamma[j]
}
y0 = y0 + eps
y1 = y0
```

```
y1[, t0:t1] = y0[, t0:t1] + tau_mat
z_mat = matrix(rep(z, t1), n, t1)
y = y0 * (1-z_mat) + y1 * z_mat
```

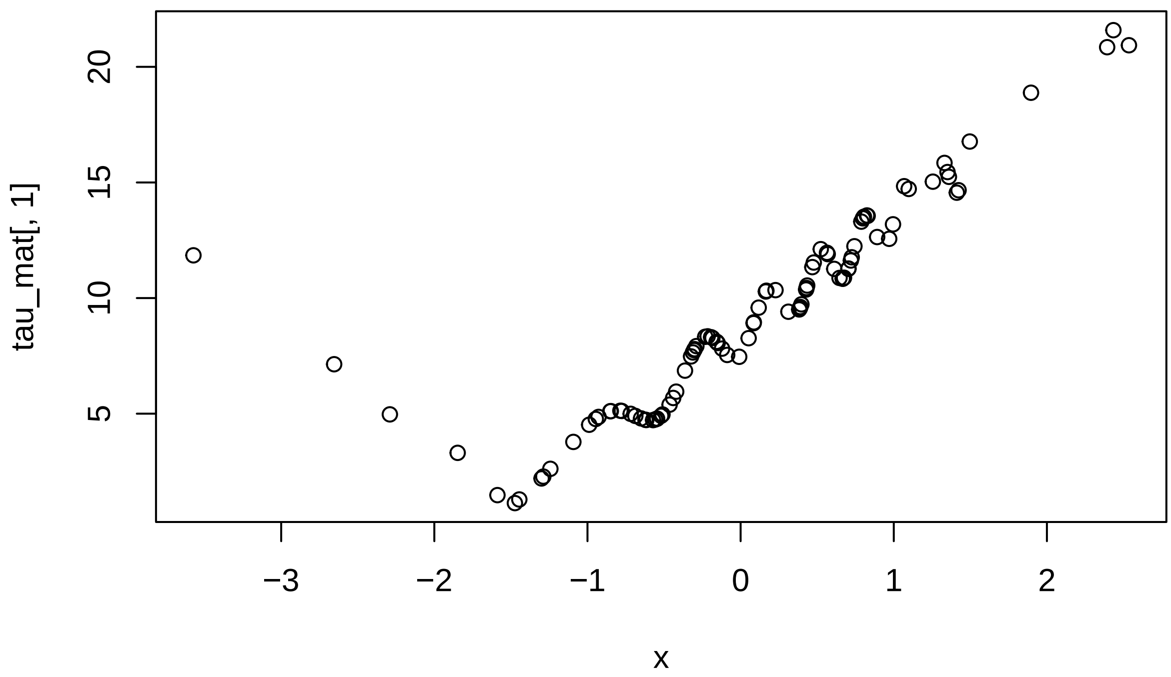
Visualize time series

```
plot(1:t1, rep(mu_a, t1), type = "l", col = 3, ylim = range(y), lwd = 2) # mean indiviudal effect
for (i in 1:50){
  lines(1:t1, y[i,], col = 1, lty = 2)
}
lines(1:t1, rep(mu_a, t1), col = 3, lwd = 2)
lines(1:t1, gamma, col = 2, lwd = 2) # time effect
legend("topleft", legend = c("Mean Individual Effect", "Time effect", "y"), col = c(3, 2, 1), lty = c(1
```

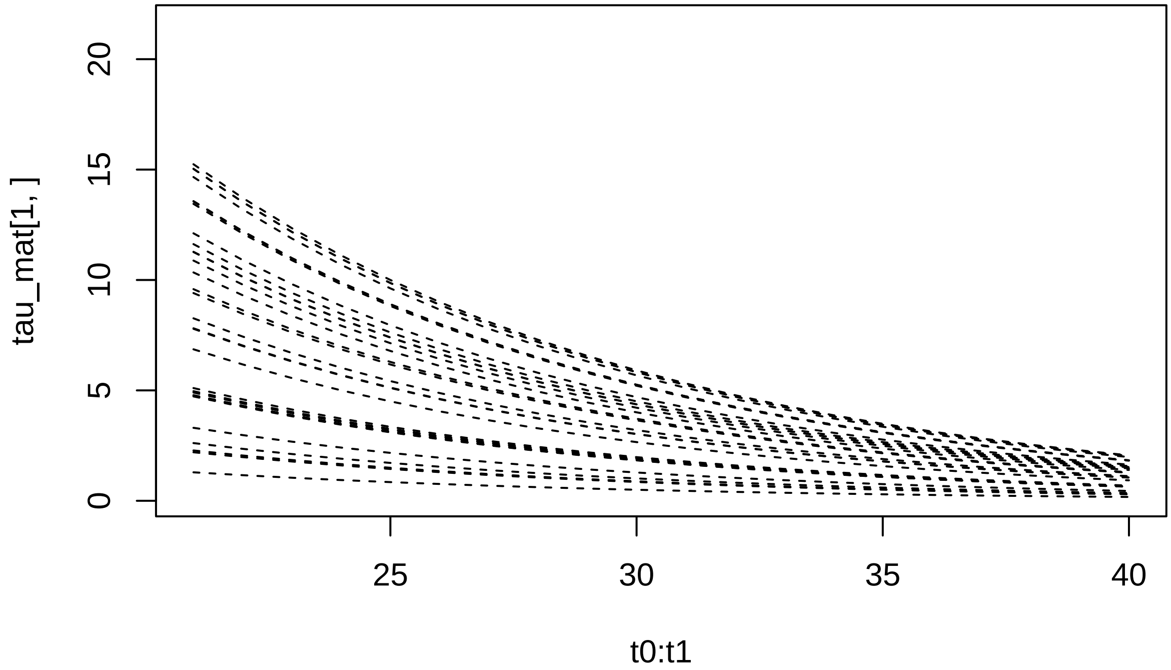


Visualize treatment effect

```
# treatment effect over x
plot(x, tau_mat[,1])
```



```
# treatment effect over time
plot(t0:t1, tau_mat[1,], type = "l", lty = 2, ylim = range(tau_mat))
for (i in seq(1, n, length.out = 30)){
  lines(t0:t1, tau_mat[i,], lty=2)
}
```



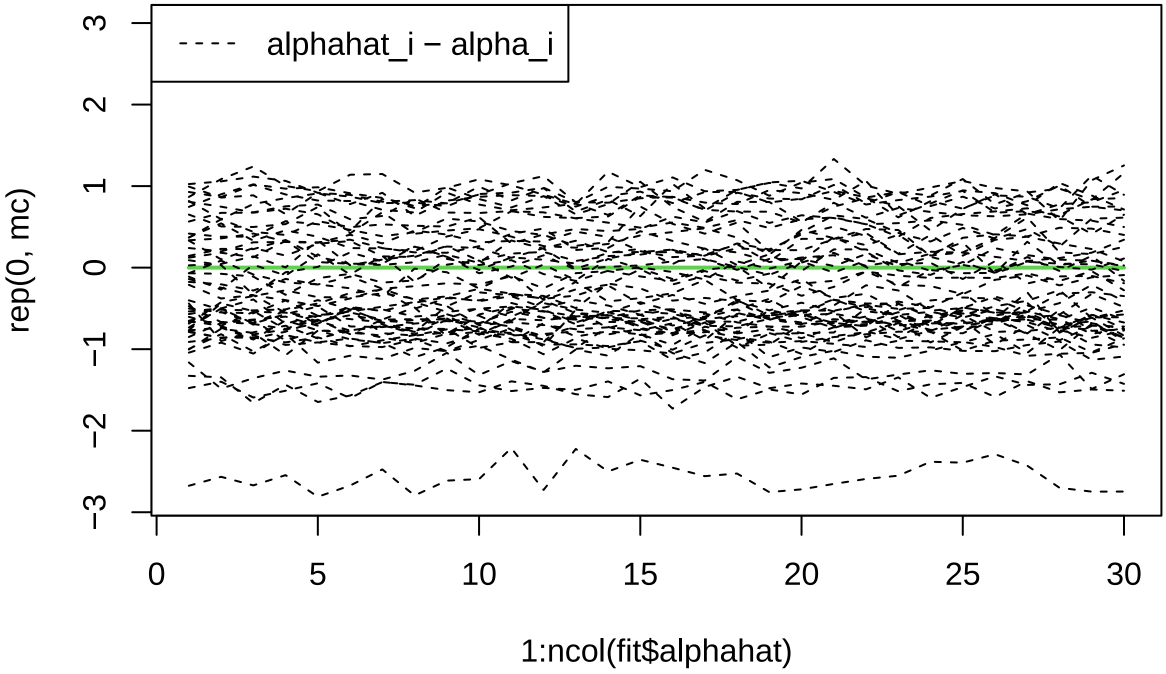
Demo model

Model

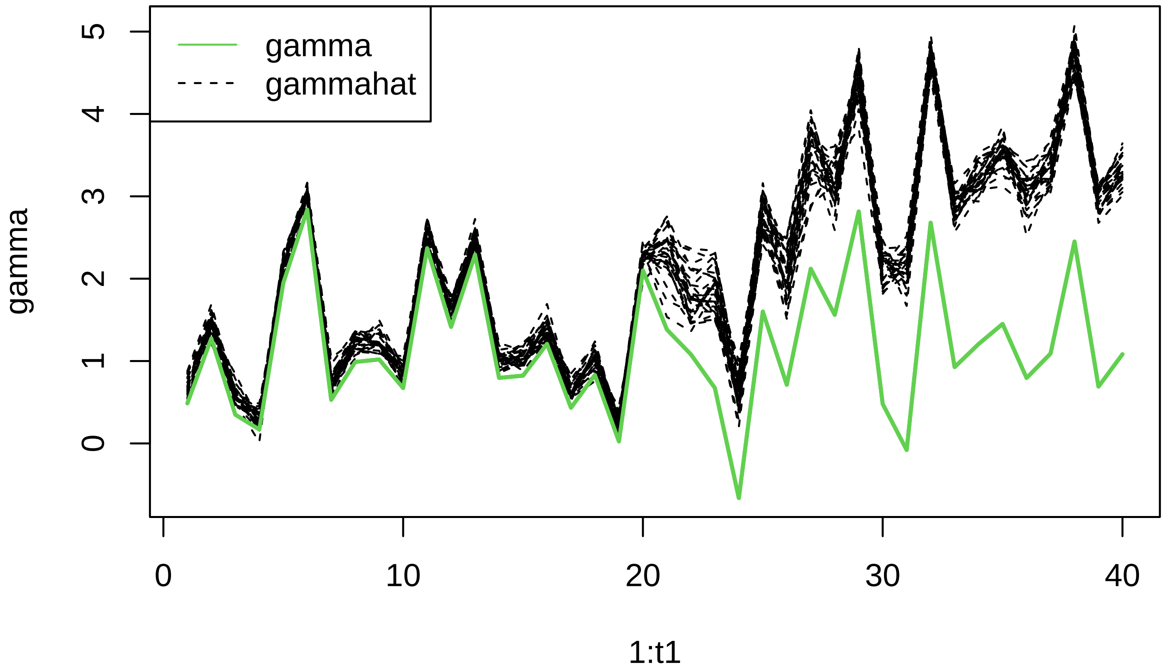
```
source('longBet_re.R')
library(XBCF)
```

```
fit <- longBet_re(y, x, z, t0, 50, 20)
```

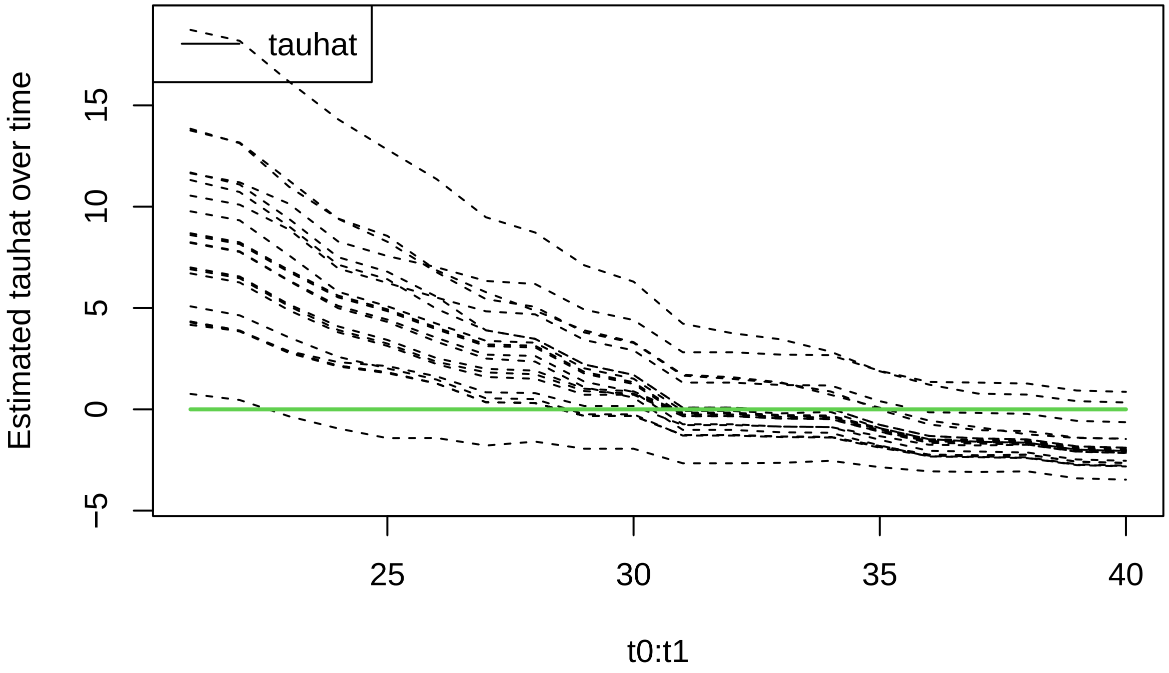
```
mc = ncol(fit$alphahat)
plot(1:ncol(fit$alphahat), rep(0, mc), type = "l", col = 3, ylim = range(fit$alphahat - alpha), lwd = 2
for (i in 1:50){
  lines(1:ncol(fit$alphahat), fit$alphahat[i,]-alpha[i], col = 1, lty = 2)
}
legend("topleft", legend = c("alphahat_i - alpha_i"), col = c(1), lty = c(2))
```



```
mc = ncol(fit$gammahat)
plot(1:t1, gamma, type = "l", col = 3, ylim = range(fit$gammahat, gamma), lwd = 2)
for (i in 1:mc){
  lines(1:t1, fit$gammahat[,i], col = 1, lty = 2)
}
lines(1:t1, gamma, col = 3, lwd = 2)
legend("topleft", legend = c("gamma", "gammahat"), col = c(3,1), lty = c(1,2))
```

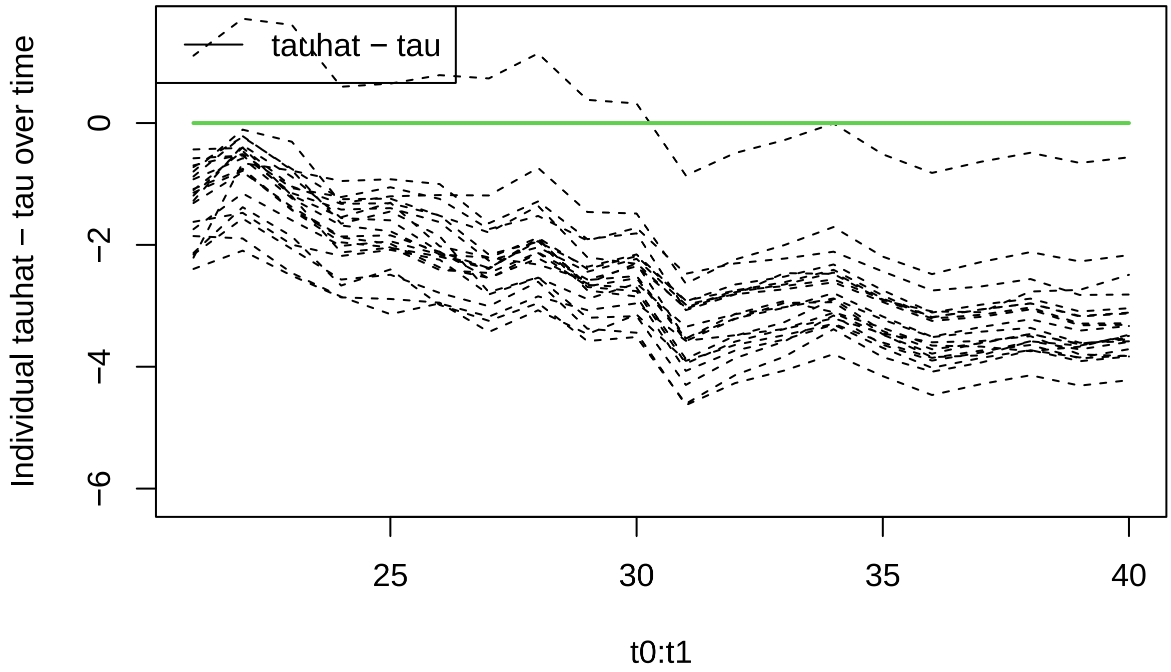


```
plot(t0:t1, colMeans(fit$tauhat[,1,]) , type = "l", col = 1, ylim = range(colMeans(fit$tauhat)),lty = 2
for (i in seq(1, n, length.out = 20)){
  lines(t0:t1, colMeans(fit$tauhat[1:10,i,]), col = 1, lty = 2)
}
lines(t0:t1, rep(0, t1-t0+1), col = 3, lty = 1, lwd = 2)
legend("topleft", legend = c("tauhat"), col = c(1), lty = c(1,2))
```

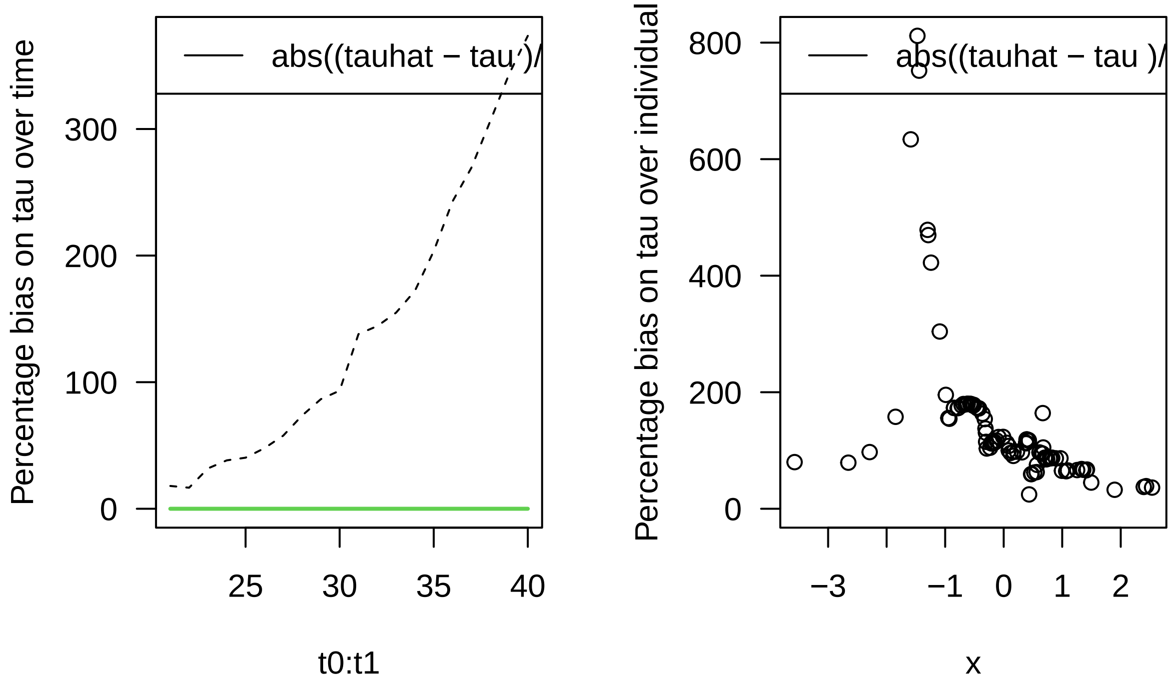


```
plot(t0:t1, colMeans(fit$tauhat[,1,]) - tau_mat[1,], type = "l", col = 1, ylim = range(colMeans(fit$tau
for (i in seq(1, n, length.out = 20)){
  lines(t0:t1, colMeans(fit$tauhat[1:10,i,]) - tau_mat[i,], col = 1, lty = 2)
}
```

```
lines(t0:t1, rep(0, t1-t0+1), col = 3, lty = 1, lwd = 2)
legend("topleft", legend = c("tauhat - tau"), col = c(1), lty = c(1,2))
```

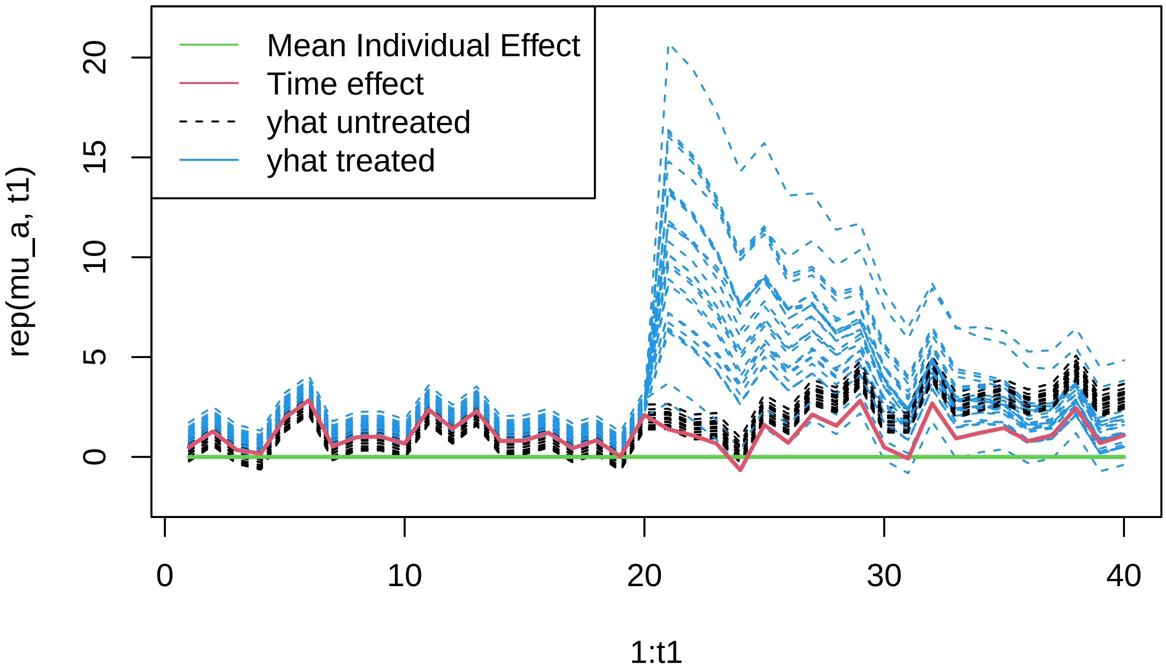


```
pct_bias <- abs((colMeans(fit$tauhat) - tau_mat) / tau_mat)
par(mfrow=c(1,2))
plot(t0:t1, colMeans(pct_bias), type = "l", col = 1, ylim = range(0, colMeans(pct_bias)),lty = 2, ylab =
lines(t0:t1, rep(0, t1-t0+1), col = 3, lty = 1, lwd = 2)
axis(2, at=pretty(colMeans(pct_bias)), lab=pretty(colMeans(pct_bias)) * 100, las=TRUE)
legend("topleft", legend = c("abs((tauhat - tau )/tau)"), col = c(1), lty = c(1,2))

plot(x, rowMeans(pct_bias), col = 1, ylim = range(0, rowMeans(pct_bias)), ylab = 'Percentage bias on ta
axis(2, at=pretty(rowMeans(pct_bias)), lab=pretty(rowMeans(pct_bias)) * 100, las=TRUE)
legend("topleft", legend = c("abs((tauhat - tau )/tau)"), col = c(1), lty = c(1,2))
```



Plot yhat

```
yhat <- matrix(rep(rowMeans(fit$alphahat), t1), n, t1)
yhat <- yhat +  t(matrix(rep(rowMeans(fit$gammahat), n), t1, n))
yhat[, t0:t1] <- yhat[,t0:t1] + colMeans(fit$muhat)
yhat[,t0:t1] <- yhat[,t0:t1] + colMeans(fit$tauhat) * matrix(rep(z, t1-t0+1), n, t1-t0+1)


plot(1:t1, rep(mu_a, t1), type = "l", col = 3, ylim = range(yhat), lwd = 2) # mean indiviudal effect
for (i in 1:50){
  lines(1:t1, yhat[i,], col = 1 + 3*z[i], lty = 2)
}
lines(1:t1, rep(mu_a, t1), col = 3, lwd = 2)
lines(1:t1, gamma, col = 2, lwd = 2) # time effect
legend("topleft", legend = c("Mean Individual Effect", "Time effect", "yhat untreated", "yhat treated")
       col = c(3, 2, 1, 4), lty = c(1, 1, 2))
```



Note that when the treatment effect goes to zero for all covariates, $\tau(x, t) = \tau(t)$ and it's not distinguishable from γ_t (not identifiable and can cause severe bias). The prognostic trees in XBCF can not work with γ_t when there's no prognostic effect due to the same reason.