Implementation Checks on T-FLASH Known Factors

David Gerard

2016-04-20

Abstract

This is just an implementation check to make sure I've coded up the known factors correctly.

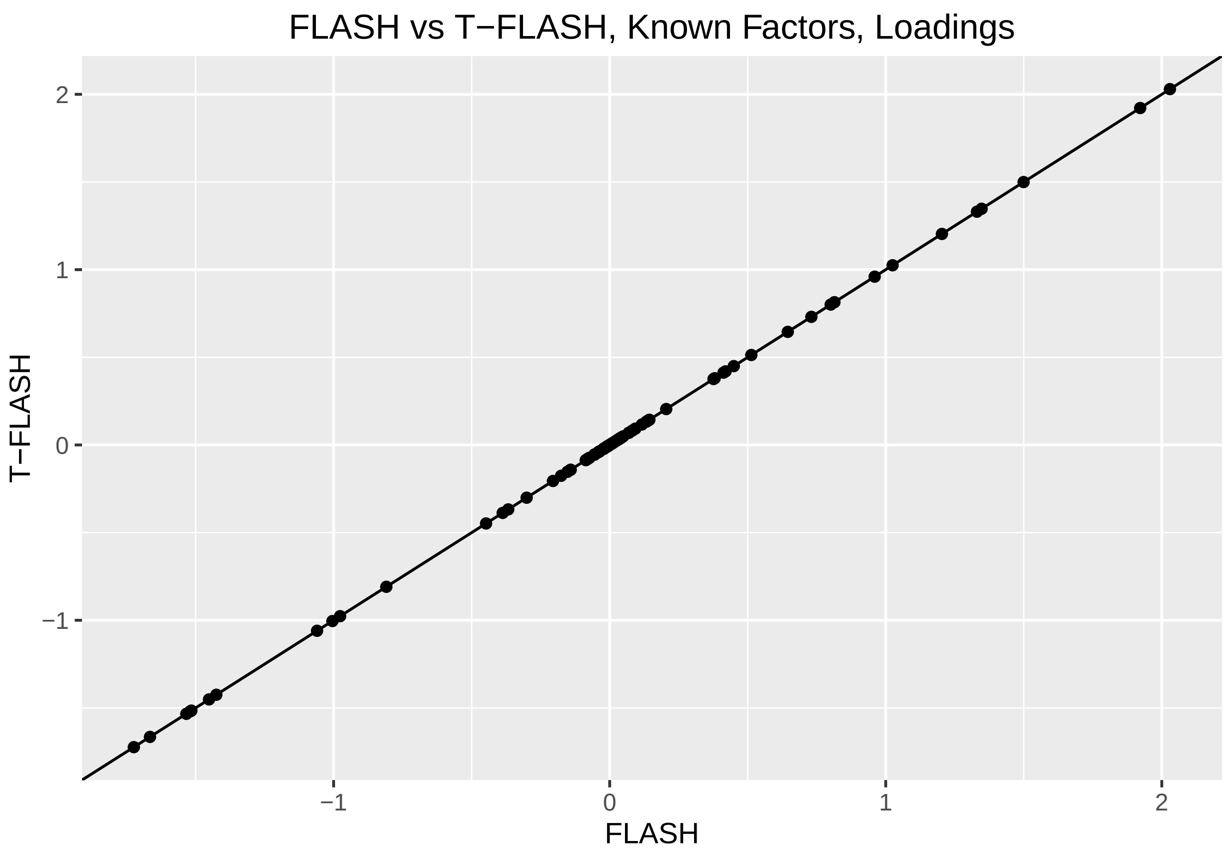
`tflash` and `flash` give equivalent results when the factor is known.

Simulate data and fit FLASH and T-FLASH.

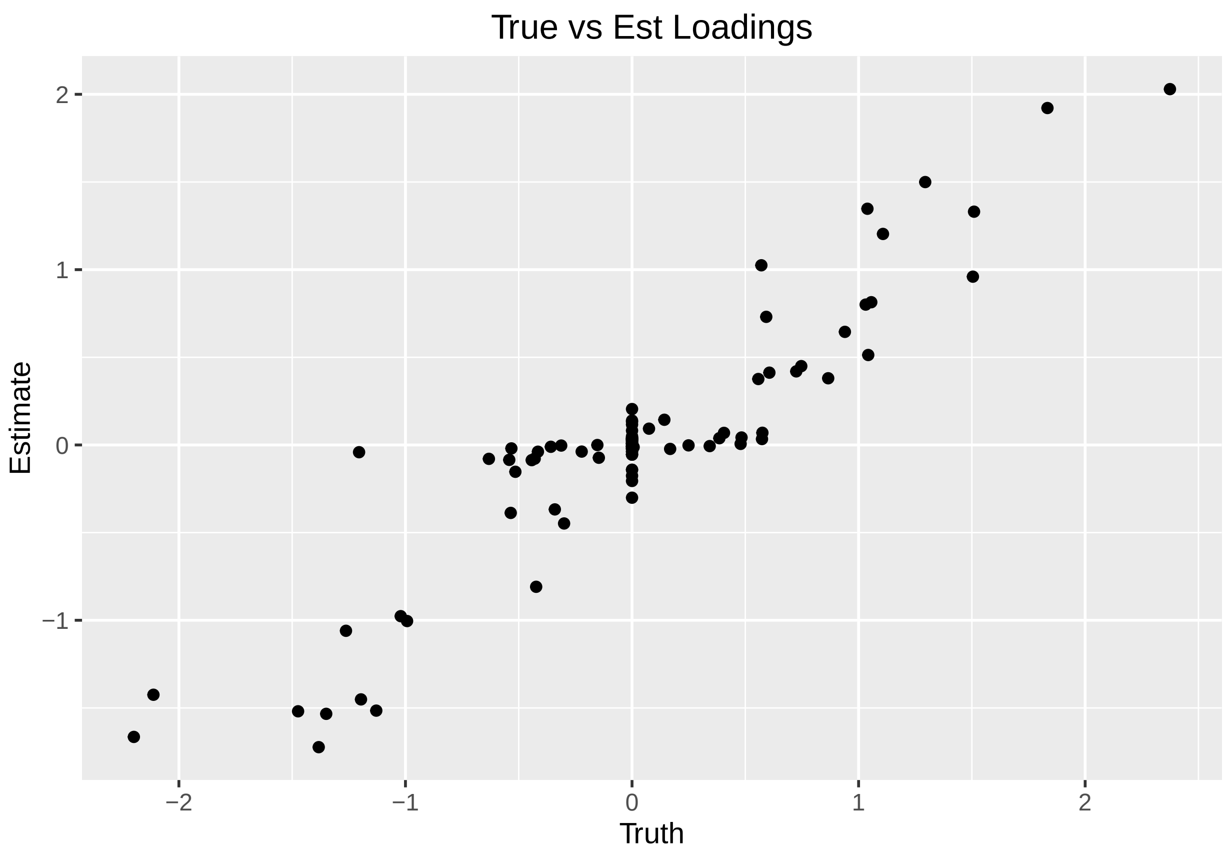
```
set.seed(211)
library(flashr)
library(ggplot2)
n <- 10
p <- 100
k <- 5
q <- 1

pi_vals <- c(0.5, 0.5)
tau_seq <- c(0, 1)

X <- matrix(rnorm(n * 1), nrow = n)
beta <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p)
E <- matrix(rnorm(n * p), nrow = n)
Y <- X %*% t(beta) + E


fout <- flash(Y = t(Y), factor_value = c(X), fix_factor = TRUE)
```

```
## [1] -1394
## [1] -1394
## [1] -1394
## [1] -1394
## [1] -1394
## [1] -1394
## [1] -1394
```

```
tout <- tflash(Y = Y, known_factors = list(X), known_modes = 1)

qplot(fout$l, tout$post_mean[[2]], main = "FLASH vs T-FLASH, Known Factors, Loadings") +
    xlab("FLASH") + ylab("T-FLASH") +
    geom_abline(slope = 1, intercept = 0)
```



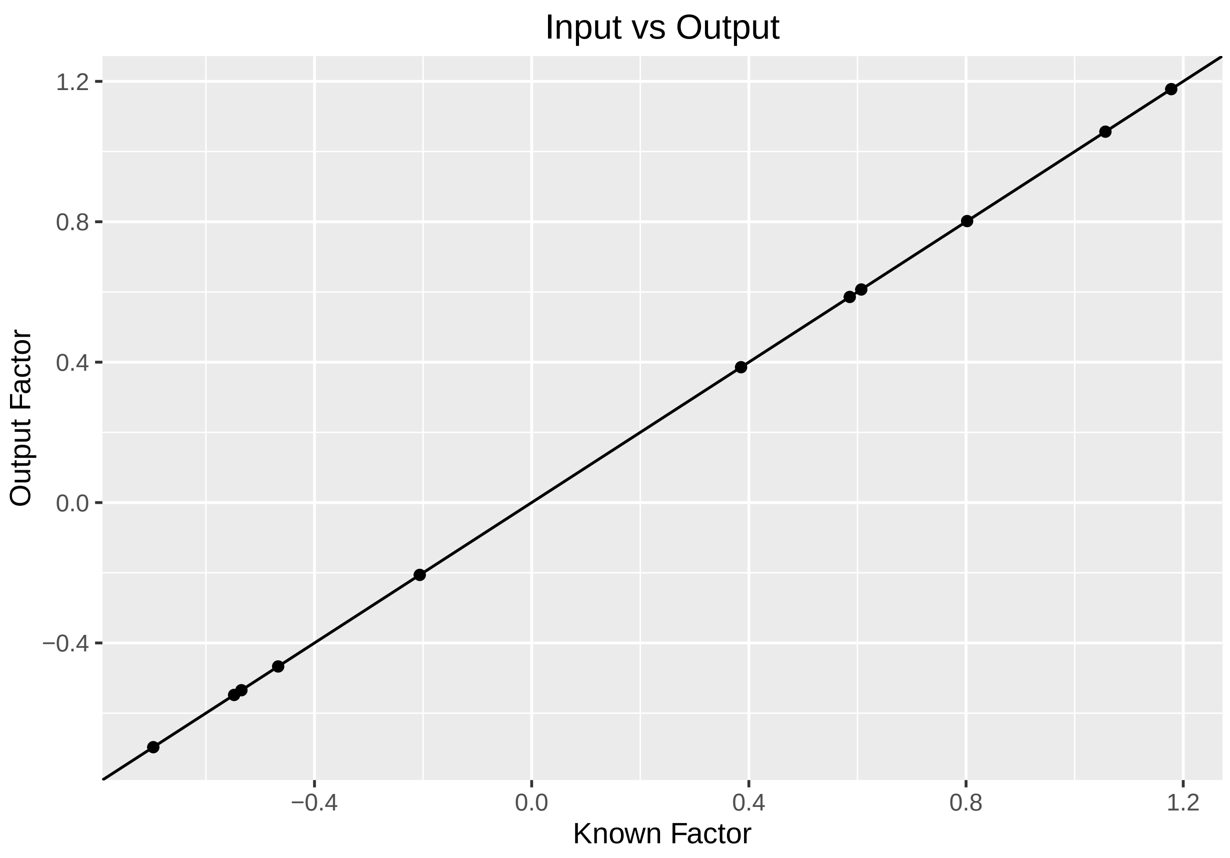
```
qplot(beta, tout$post_mean[[2]], xlab = "Truth", ylab = "Estimate", main = "True vs Est Loadings")
```



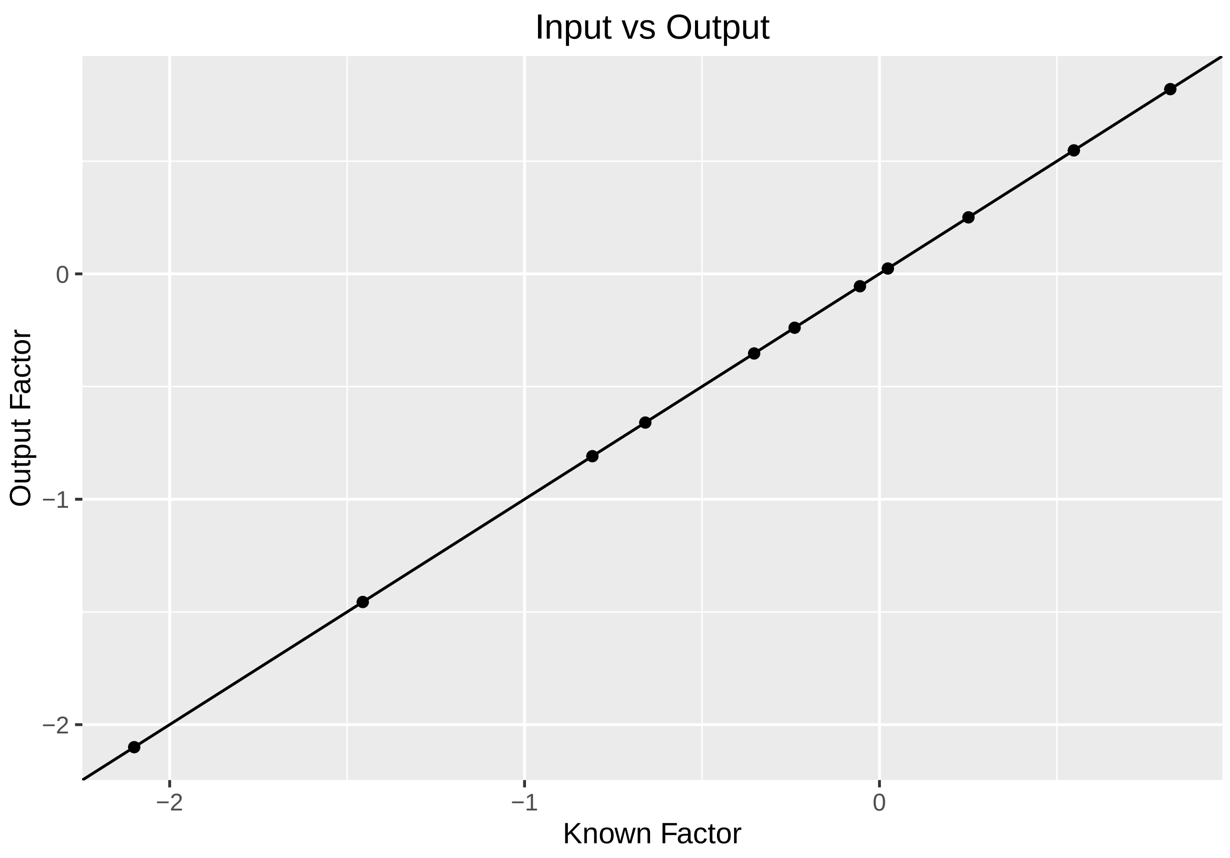
`tgreedy` returns known factors and results look reasonable.

First, I make sure that the input factors are correctly returned.

```r
set.seed(101)
rm(list = ls())
n <- 11
p <- 21

E <- matrix(rnorm(n * p), nrow = n)
X <- matrix(rnorm(n * 2), nrow = n)
beta <- matrix(rnorm(p * 2), ncol = 2)
Y <- X %*% t(beta) + E

tout <- tgreedy(Y = Y, known_factors = list(X), known_modes = 1)

qplot(X[, 1], tout$factor_list[[1]][, 1], xlab = "Known Factor", ylab = "Output Factor",
      main = "Input vs Output") +
    geom_abline(intercept = 0, slope = 1)
```

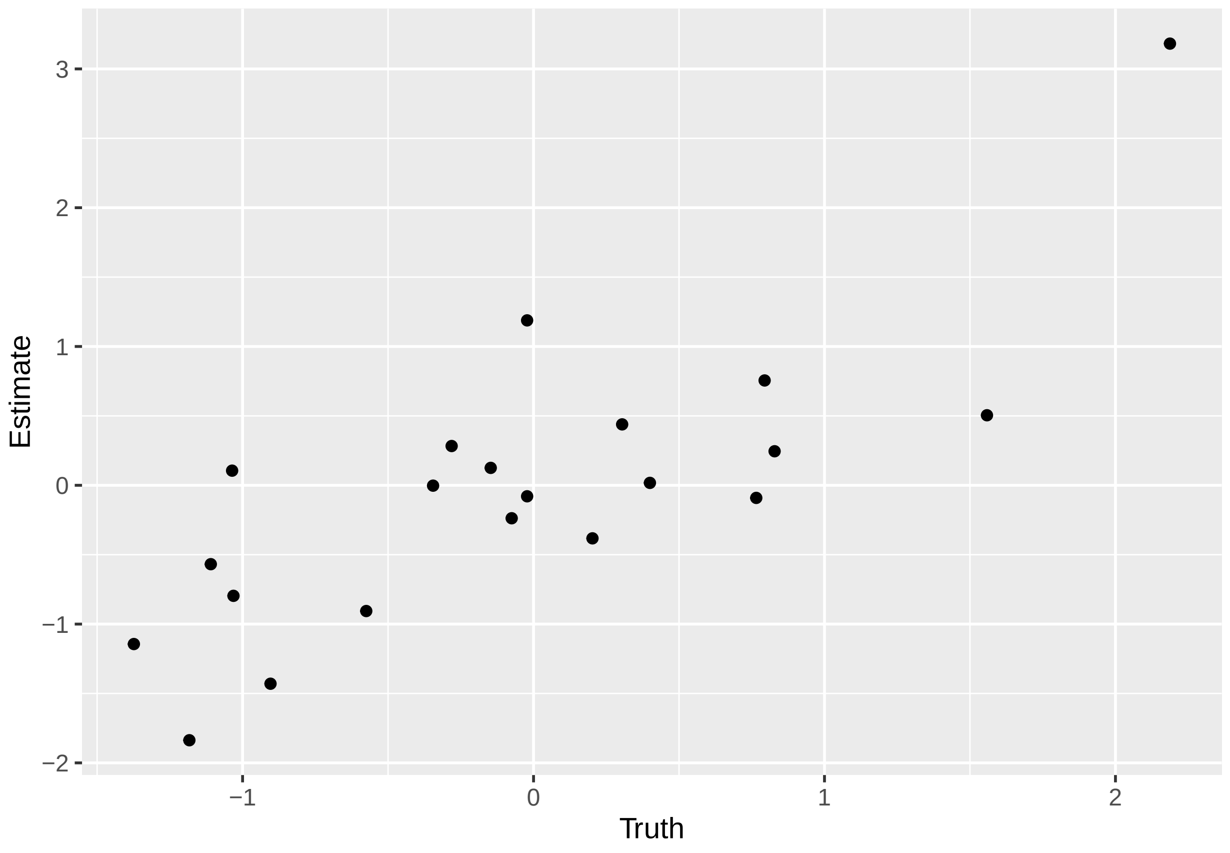


```r
qplot(X[, 2], tout$factor_list[[1]][, 2], xlab = "Known Factor", ylab = "Output Factor",
      main = "Input vs Output") +
    geom_abline(intercept = 0, slope = 1)
```

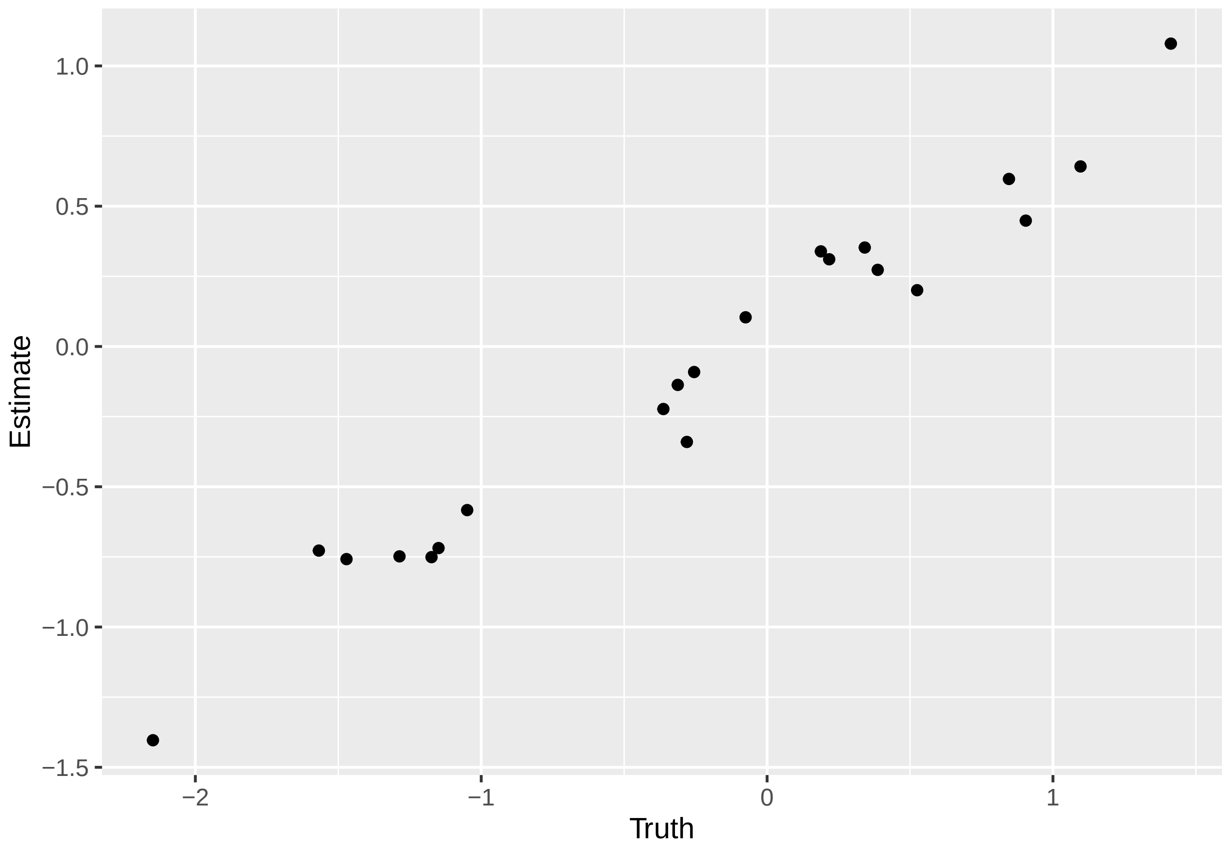


Now I make sure the estimates are reasonable

```
qplot(beta[, 1], tout$factor_list[[2]][, 1], xlab = "Truth", ylab = "Estimate")
```



```
qplot(beta[, 2], tout$factor_list[[2]][, 2], xlab = "Truth", ylab = "Estimate")
```



Tensor Case

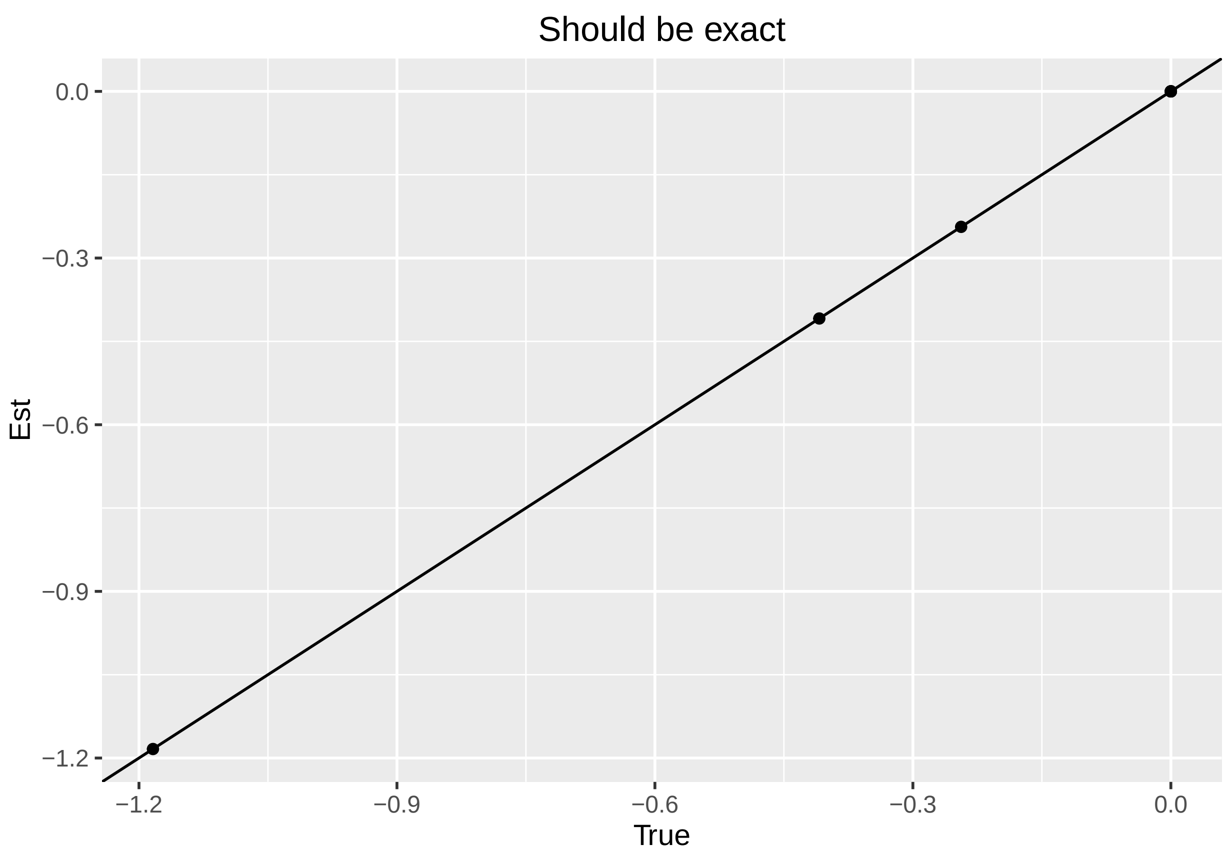
See if tensor estimates are reasonable when given two of the first mode's, one of the second, and none of the third. So three of these should be exact.

```r
rm(list = ls())
set.seed(349)
p <- c(11, 13, 17)
u <- list()
pi_vals <- c(0.5, 0.3, 0.1)
tau_seq <- c(0, 1, 2)
u[[1]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[1])
u[[2]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[2])
u[[3]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[3])
v <- list()
v[[1]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[1])
v[[2]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[2])
v[[3]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[3])
w <- list()
w[[1]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[1])
w[[2]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[2])
w[[3]] <- succotashr::draw_beta(pi_vals = pi_vals, tau_seq = tau_seq, p = p[3])


Theta <- form_outer(u) + form_outer(v) + form_outer(w)
E <- array(rnorm(prod(p)), dim = p)
```

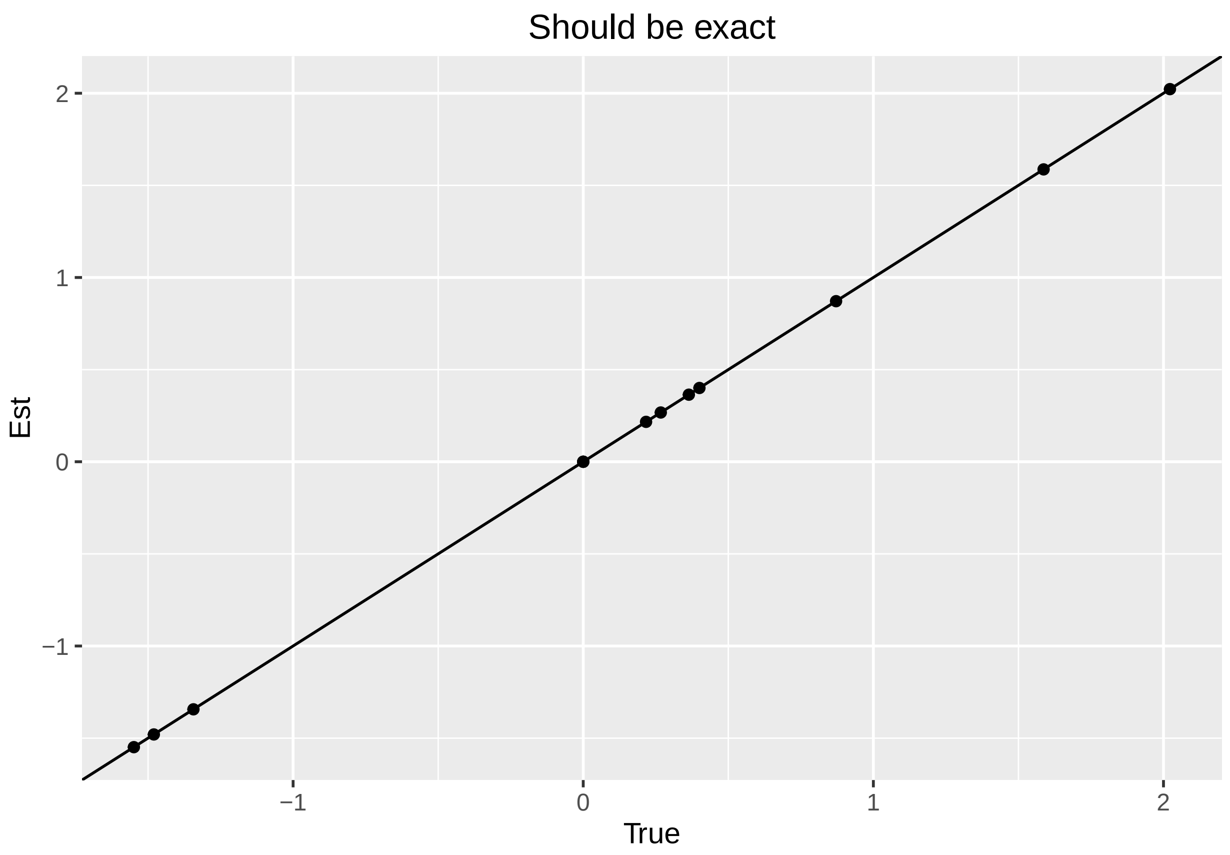
```r
Y <- Theta + E

tout <- tgreedy(Y = Y, known_factors = list(cbind(u[[1]], v[[1]]), u[[2]]), known_modes = c(1, 2))
tback <- tbackfitting(Y = Y, factor_list = tout$factor_list, sigma_est = tout$sigma_est,
                 known_factors = list(cbind(u[[1]], v[[1]]), matrix(u[[2]], ncol = 1)),
                 known_modes = c(1, 2))
```

```
## Diff: 0.628763494541832

## Diff: 0.0138689173614632

## Diff: 6.49123918473604e-05

## Diff: 3.58247623388763e-06

## Diff: 2.4845656554362e-07
```

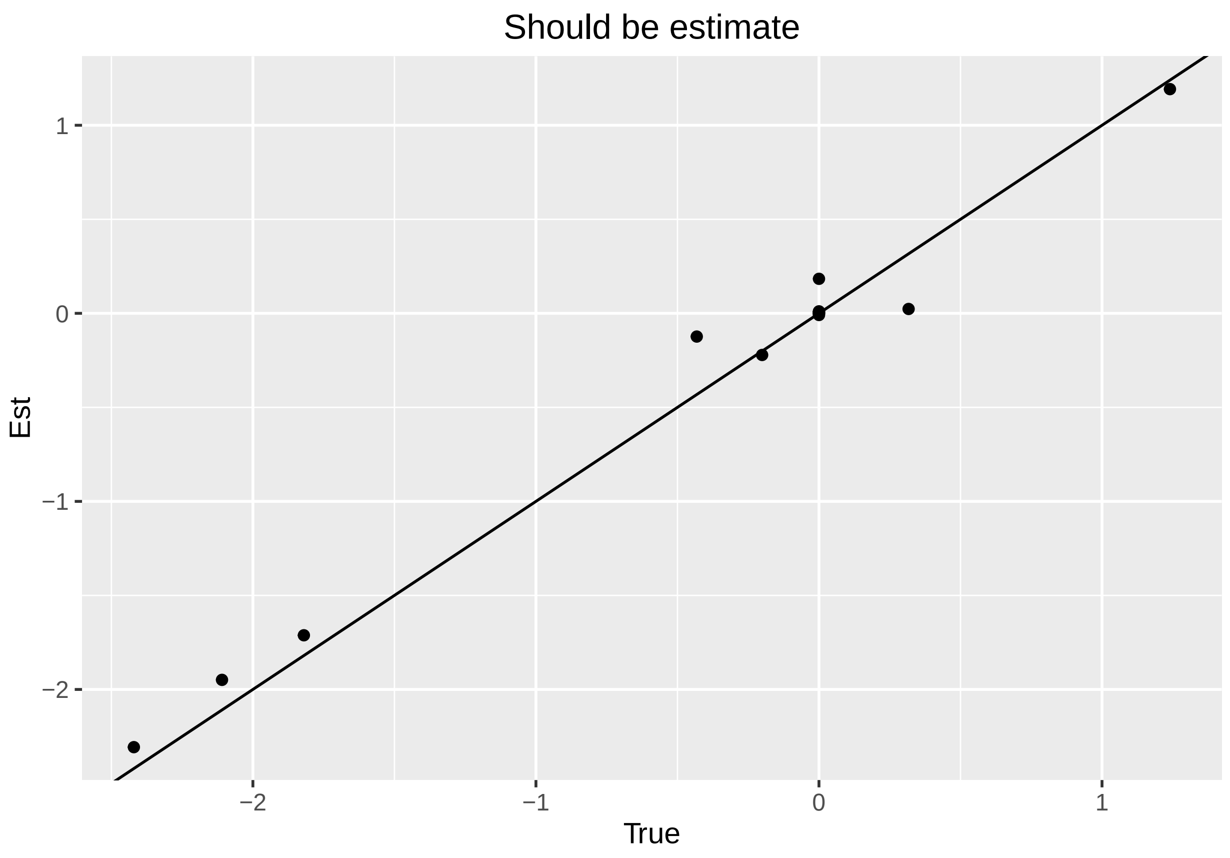
```r
qplot(u[[1]], tback$factor_list[[1]][, 1], main = "Should be exact",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```



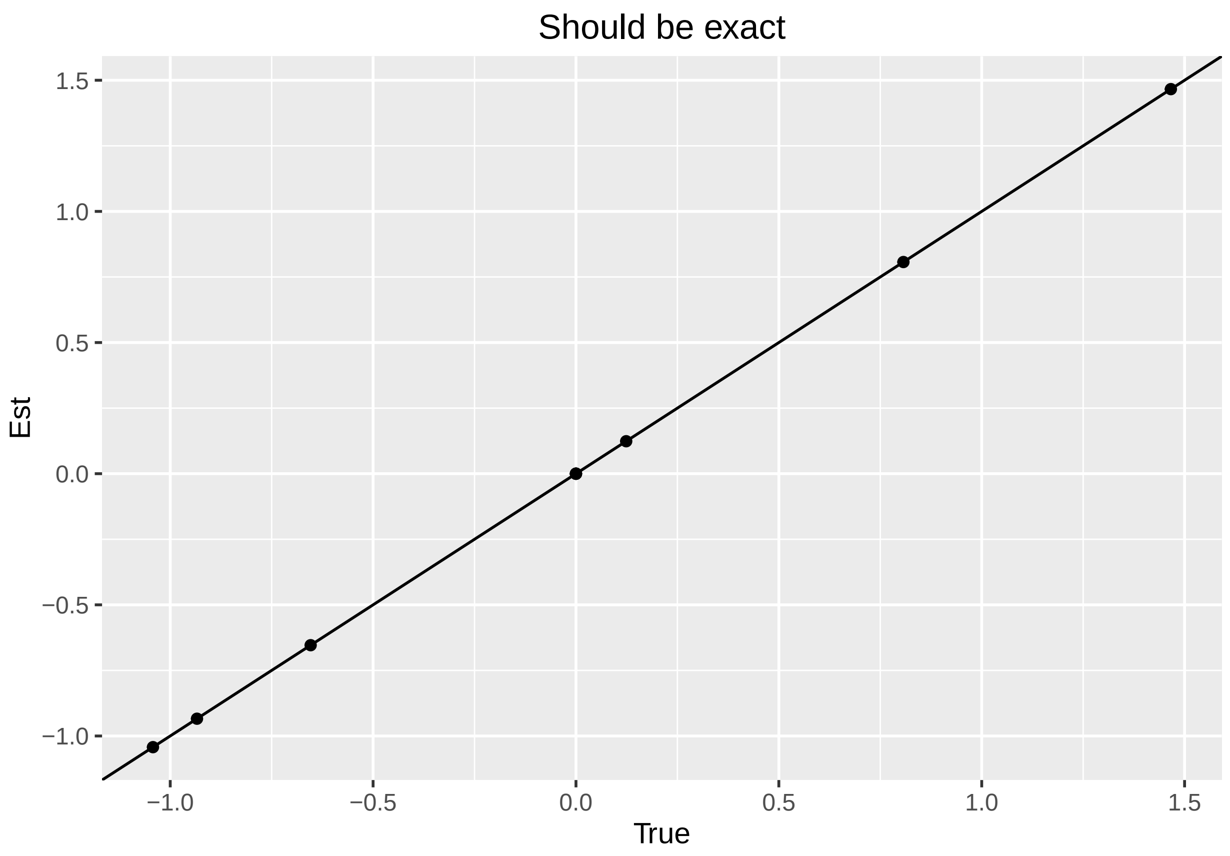
```r
qplot(u[[2]], tback$factor_list[[2]][, 1], main = "Should be exact",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```



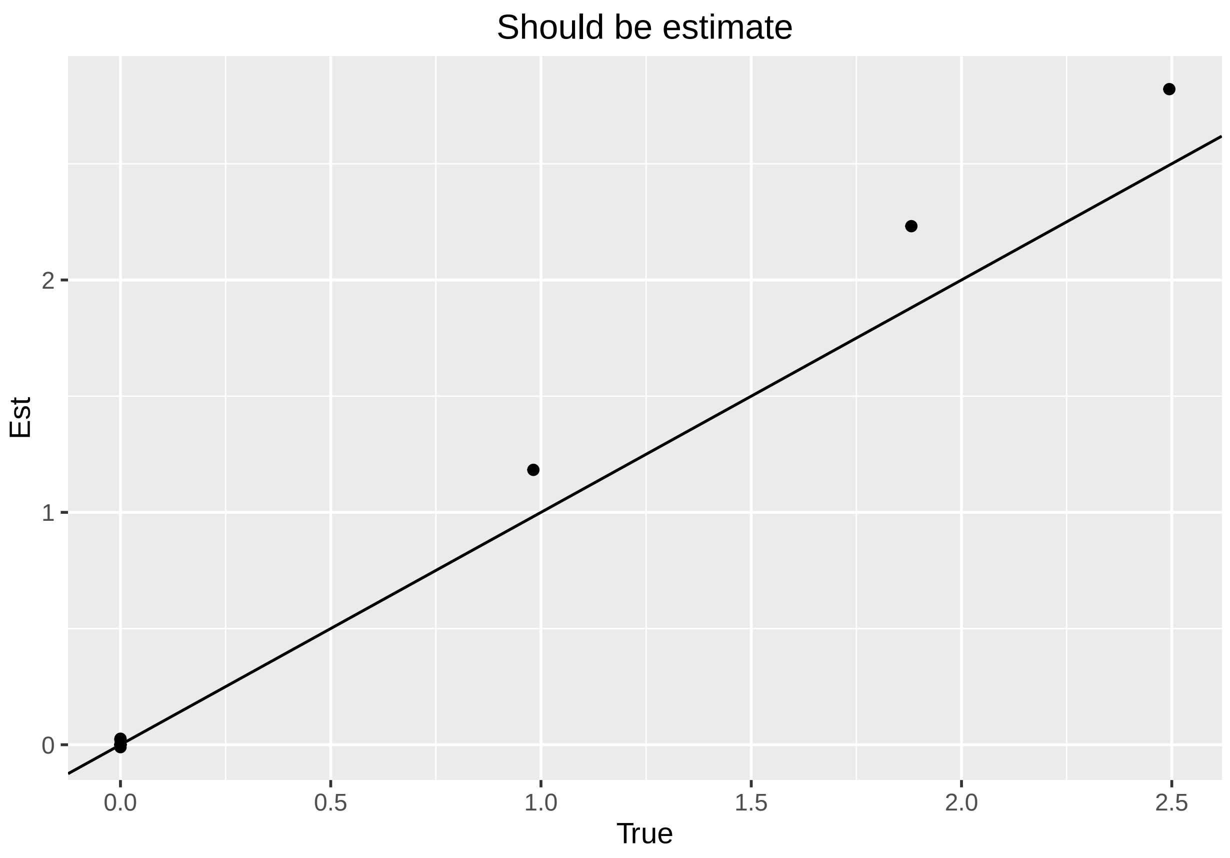
```
qplot(u[[3]], tback$factor_list[[3]][, 1], main = "Should be estimate",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```



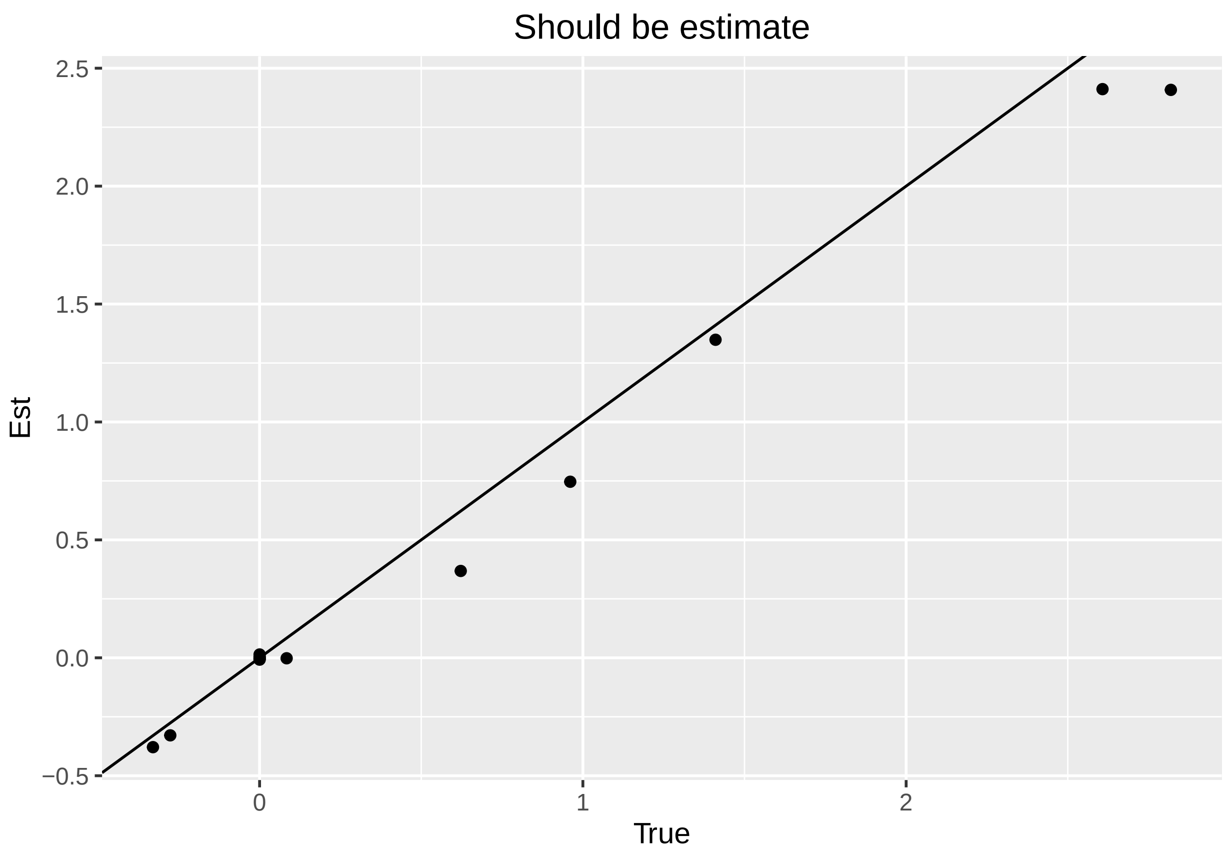
```
qplot(v[[1]], tback$factor_list[[1]][, 2], main = "Should be exact",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```



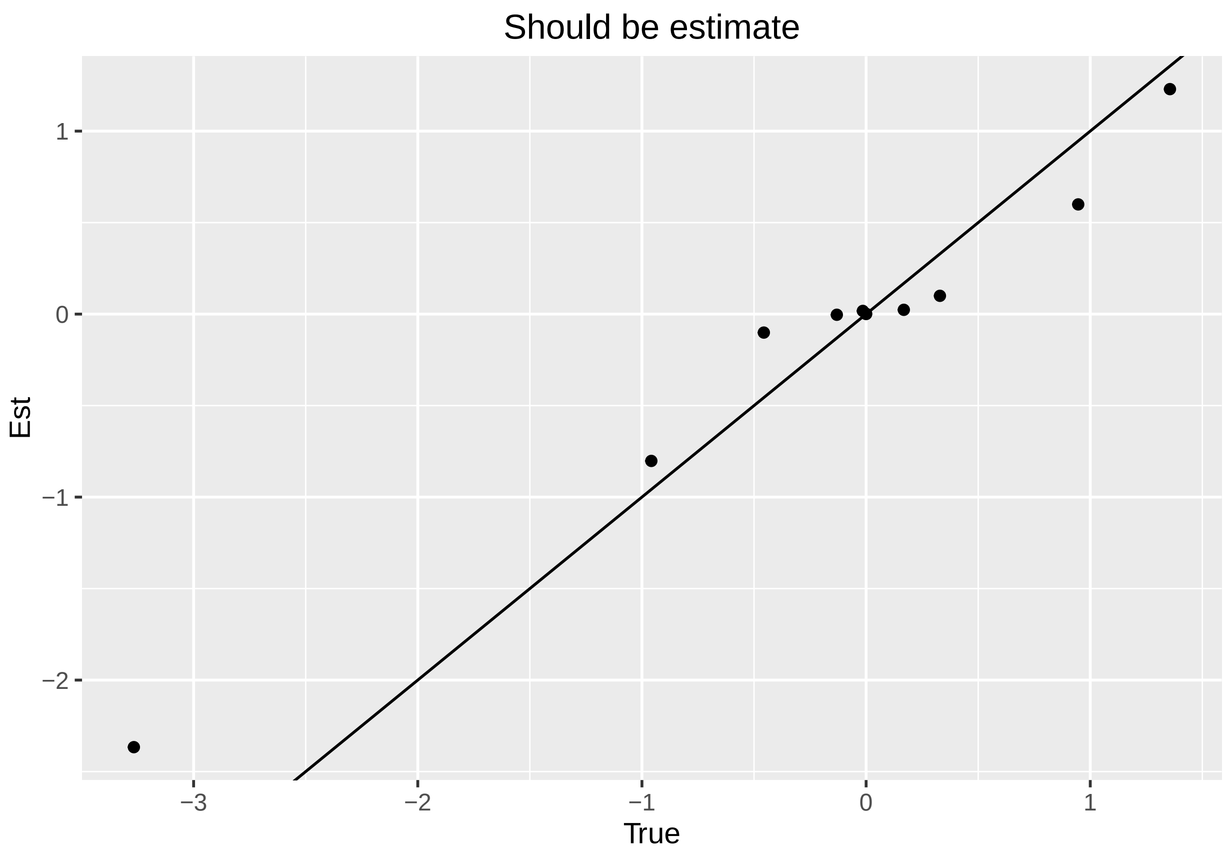
```
qplot(v[[2]], tback$factor_list[[2]][, 2] * -1, main = "Should be estimate",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```



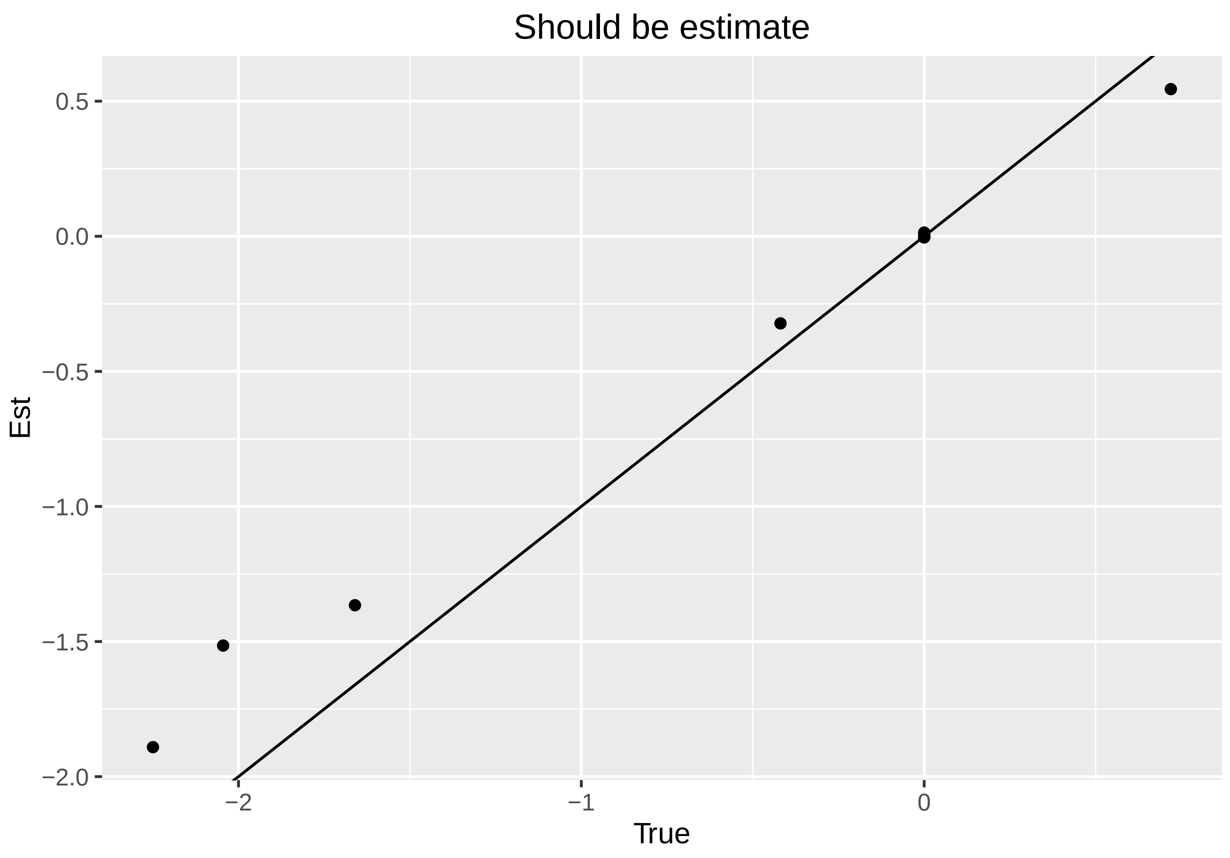
```
qplot(v[[3]], tback$factor_list[[3]][, 2] * -1, main = "Should be estimate",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```



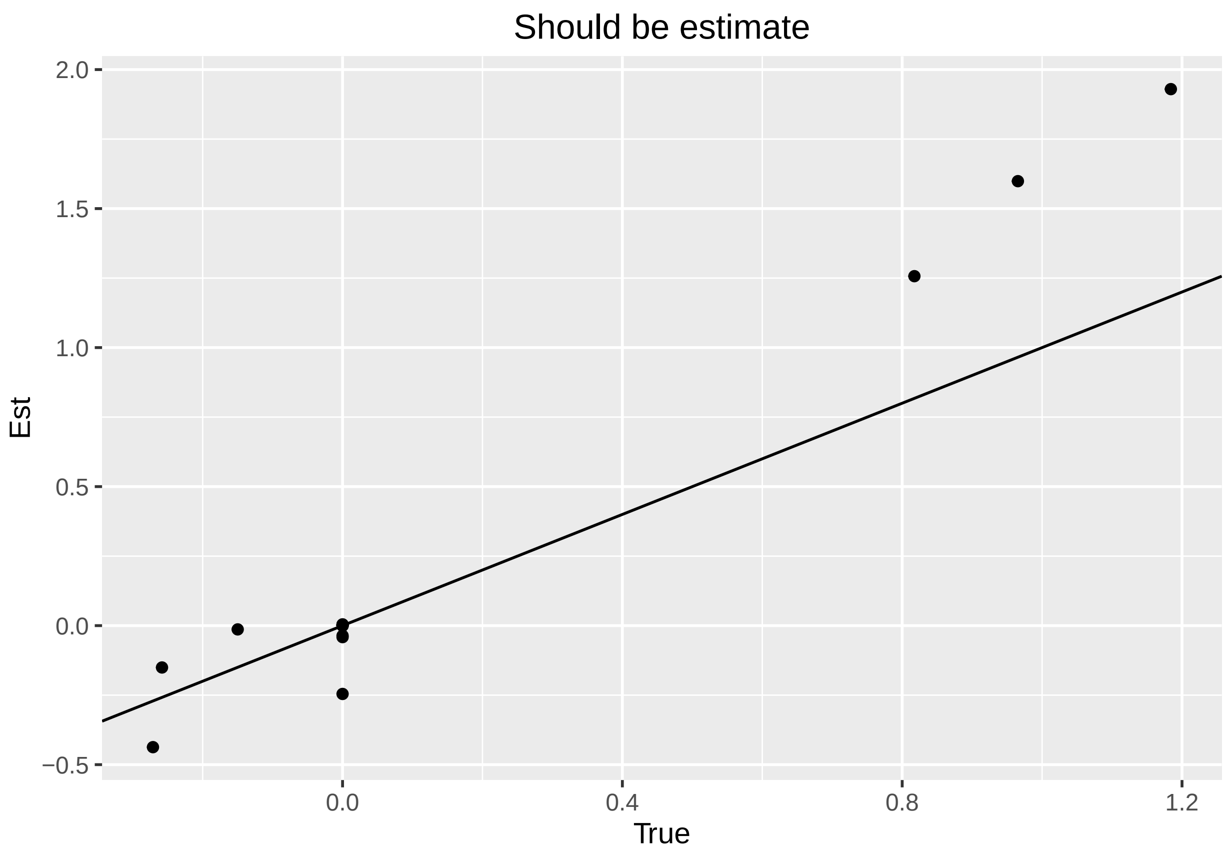
```
qplot(w[[1]], tback$factor_list[[1]][, 3], main = "Should be estimate",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```



```r
qplot(w[[2]], tback$factor_list[[2]][, 3], main = "Should be estimate",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```

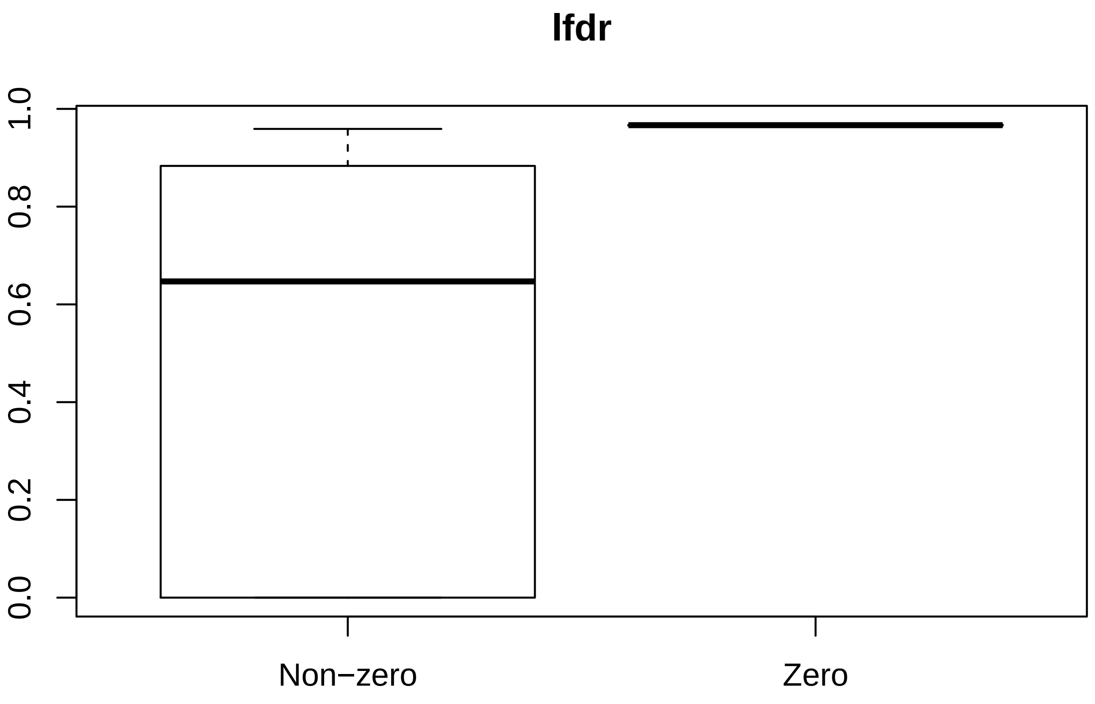


```
qplot(w[[3]], tback$factor_list[[3]][, 3], main = "Should be estimate",
      xlab = "True", ylab = "Est") + geom_abline(intercept = 0, slope = 1)
```

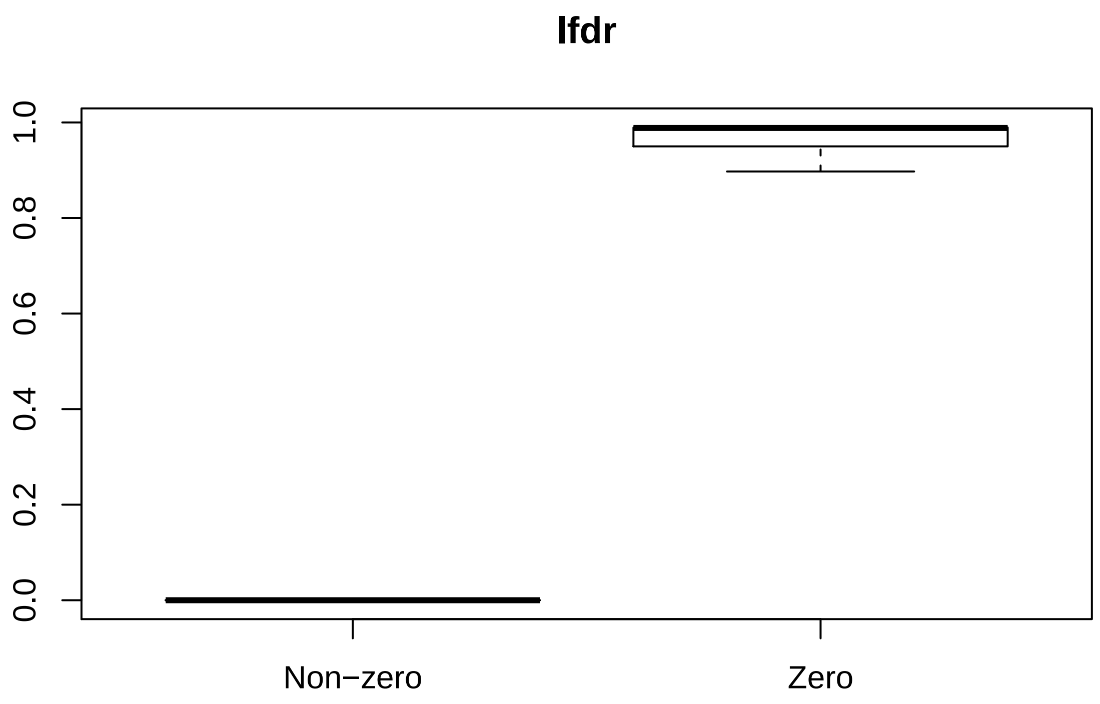


The lfdrs look really good:

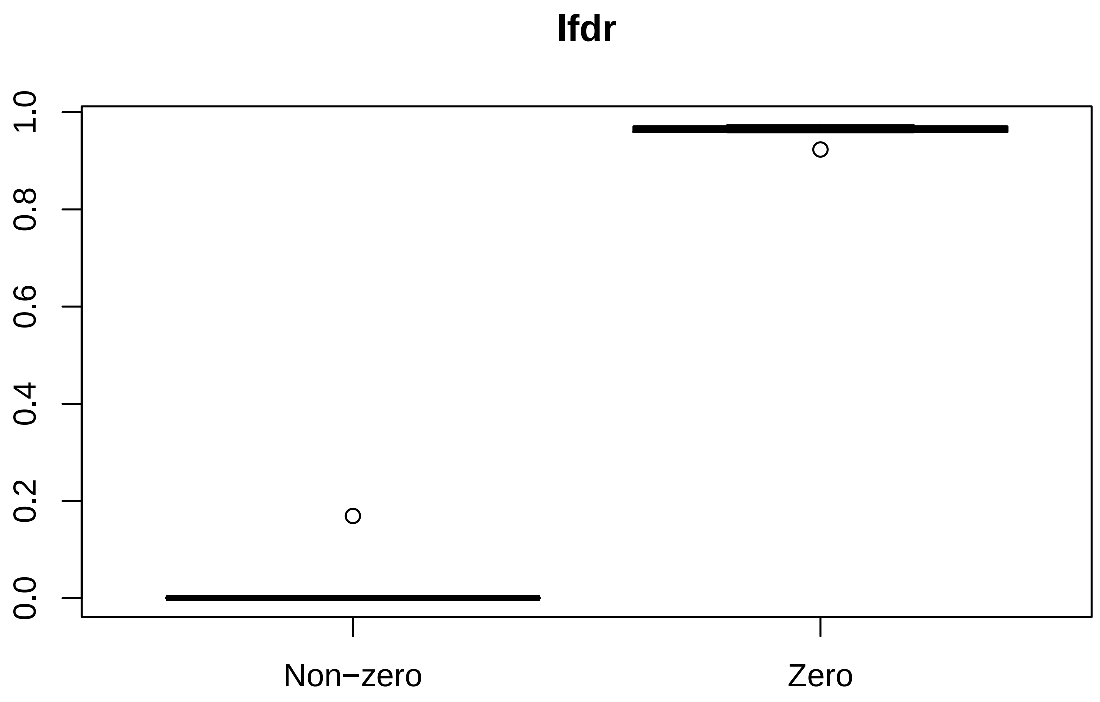
```
boxplot(tback$prob_zero[[1]][, 3] ~ (abs(w[[1]]) < 10 ^ -6), names = c("Non-zero", "Zero"),
        main = "lfdr")
```



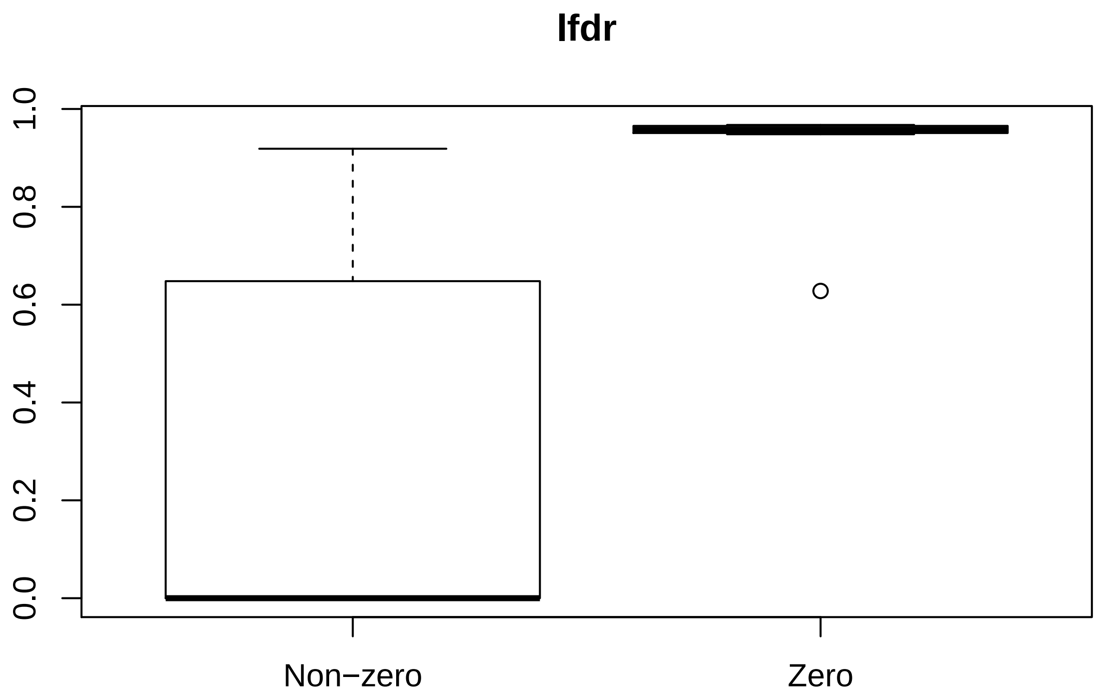
```r
boxplot(tback$prob_zero[[2]][, 2] ~ (abs(v[[2]]) < 10 ^ -6), names = c("Non-zero", "Zero"),
        main = "lfdr")
```



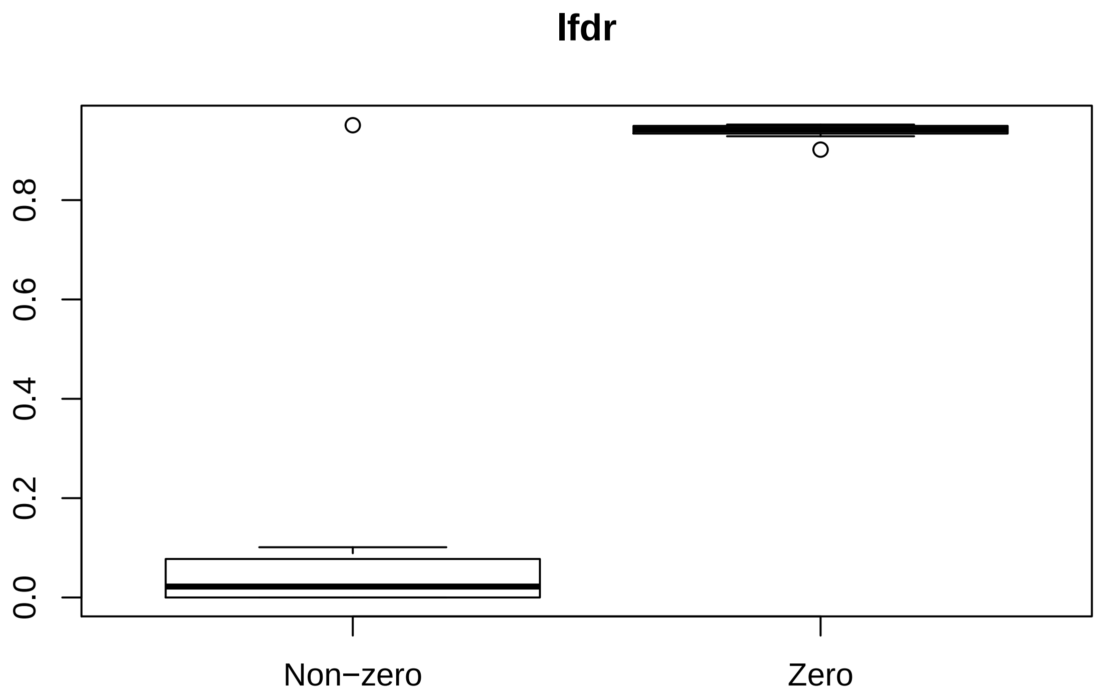
```r
boxplot(tback$prob_zero[[2]][, 3] ~ (abs(w[[2]]) < 10 ^ -6), names = c("Non-zero", "Zero"),
        main = "lfdr")
```



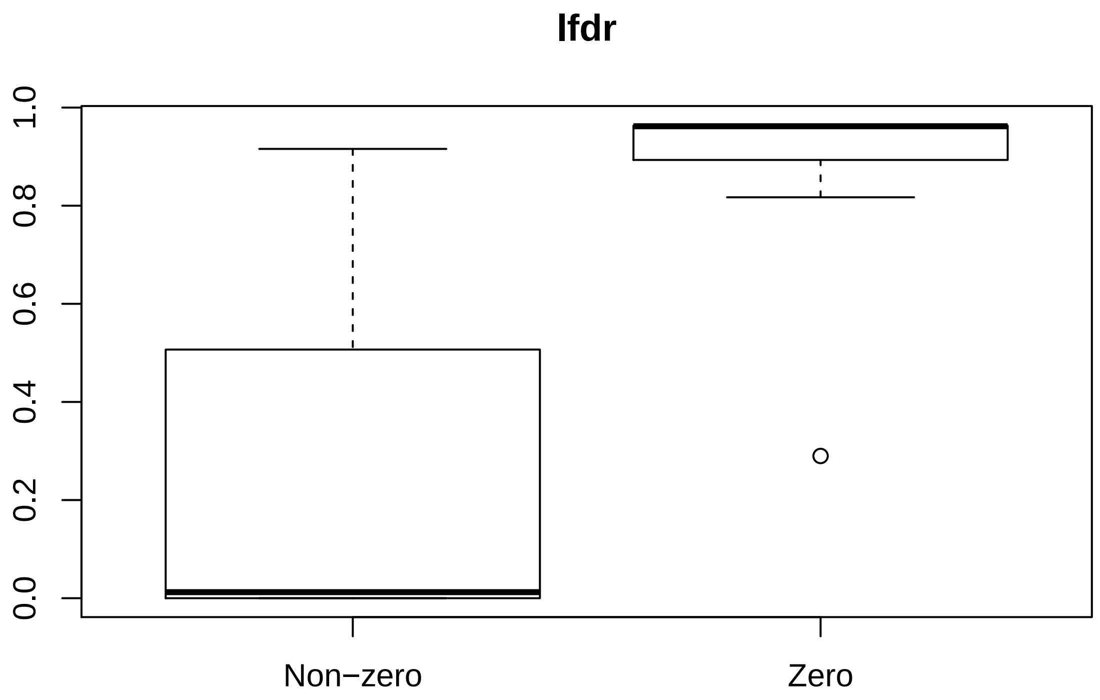
```r
boxplot(tback$prob_zero[[3]][, 1] ~ (abs(u[[3]]) < 10 ^ -6), names = c("Non-zero", "Zero"),
        main = "lfdr")
```



```
boxplot(tback$prob_zero[[3]][, 2] ~ (abs(v[[3]]) < 10 ^ -6), names = c("Non-zero", "Zero"),
        main = "lfdr")
```



```
boxplot(tback$prob_zero[[3]][, 3] ~ (abs(w[[3]]) < 10 ^ -6), names = c("Non-zero", "Zero"),
        main = "lfdr")
```



But the π_0's don't look so good for some reason. These should all be 0.5.

```
knitr::kable(sapply(tback$pi0_list, c))
```

NA	NA	0.7874
NA	0.8492	0.7150
0.7489	0.7664	0.7722