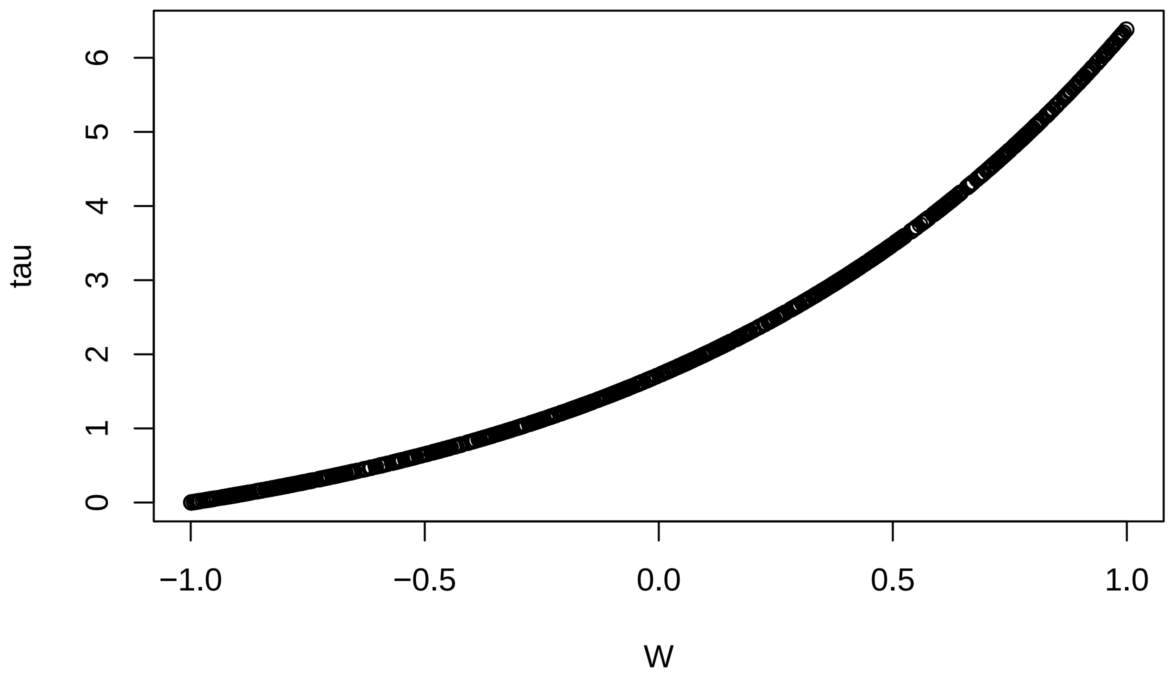
vignette

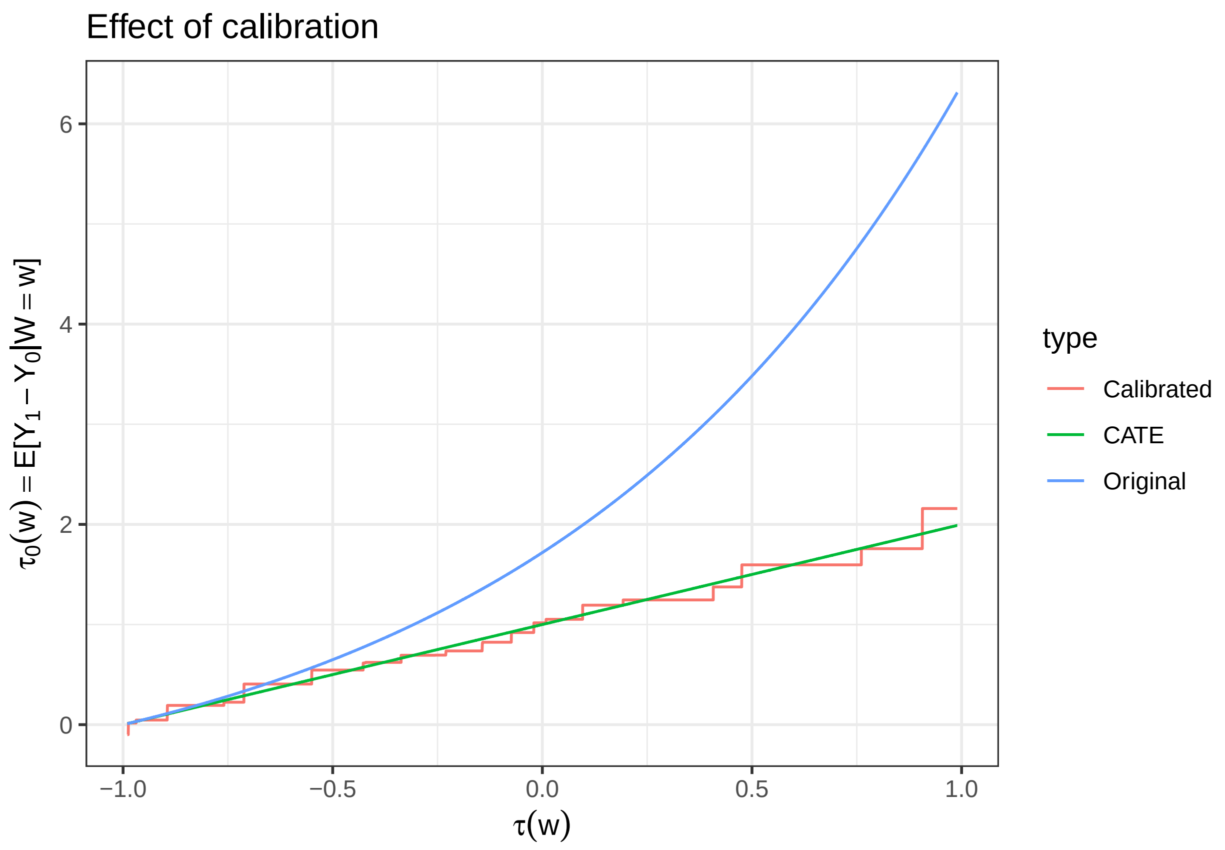
2022-12-20

Causal isotonic calibration

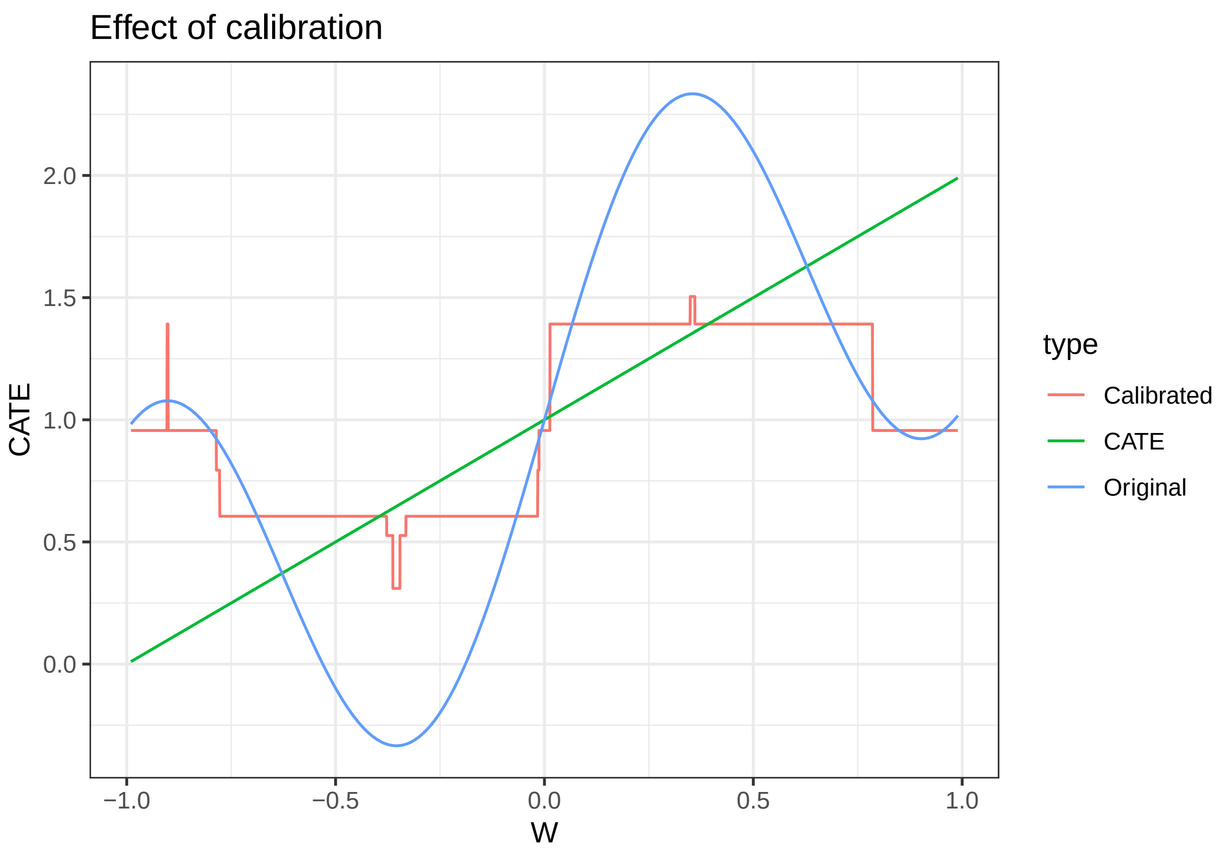
```r
library(causalCalibration)
# Generate dataset used for calibration
set.seed(123)
n <- 1000
W <- runif(n, -1 , 1)
pA1 <- plogis(2*W)
A <- rbinom(n, size = 1, pA1)
# True CATE(W) = 1 + W
CATE <- 1 + W
EY0 <- W
EY1 <- W + CATE
Y <- rnorm(n, W + A * CATE, 0.3)

# Initial uncalibrated predictor is a monotone transformation of the true CATE
# Thus it is highly correlated with true CATE but predictions are not uncalibrated for CATE values
tau <- exp(CATE) - 1

plot(W, tau)
```



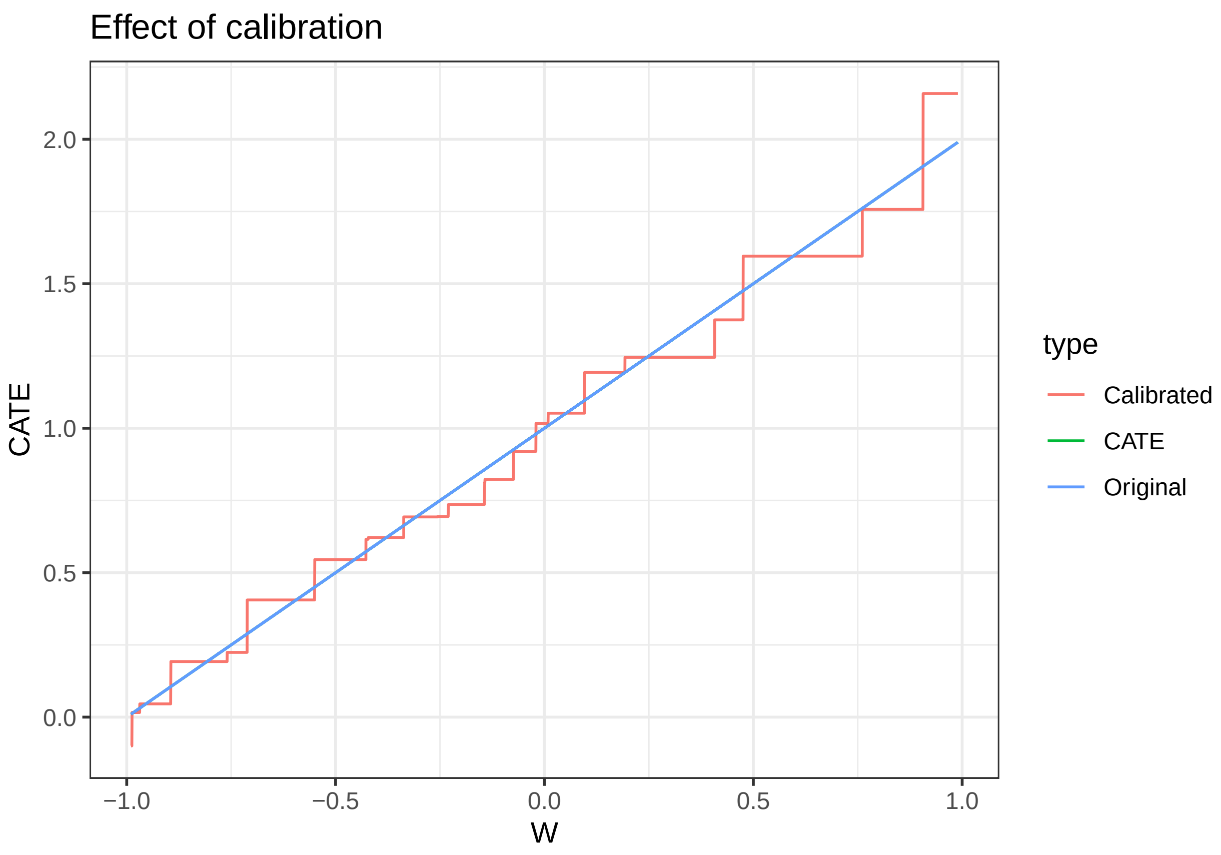
```
n <- 10000
Wnew <- runif(n, -1 , 1)
Wnew <- Wnew[abs(Wnew) <= 0.99]

n <- length(Wnew)
pA1new <- plogis(2*Wnew)
Anew <- rbinom(n, size = 1, pA1new)
# True CATE(W) = 1 + W
CATE <- 1 + Wnew
tau_new <- exp(CATE) - 1


calibrator <- causalCalibrate(tau, A, Y, EY1, EY0, pA1, tau_pred = tau_new)
library(ggplot2)
library(latex2exp)
tau_cal <- calibrator$tau_calibrated
data <- data.frame(W = Wnew, CATE = c(CATE, tau_new, tau_cal), type = rep(c("CATE", "Original", "Calibra

ggplot(data, aes(x = W, y = CATE, color = type)) + geom_line() + labs(x = TeX("$\\tau(w)$"), y =  TeX("$
```



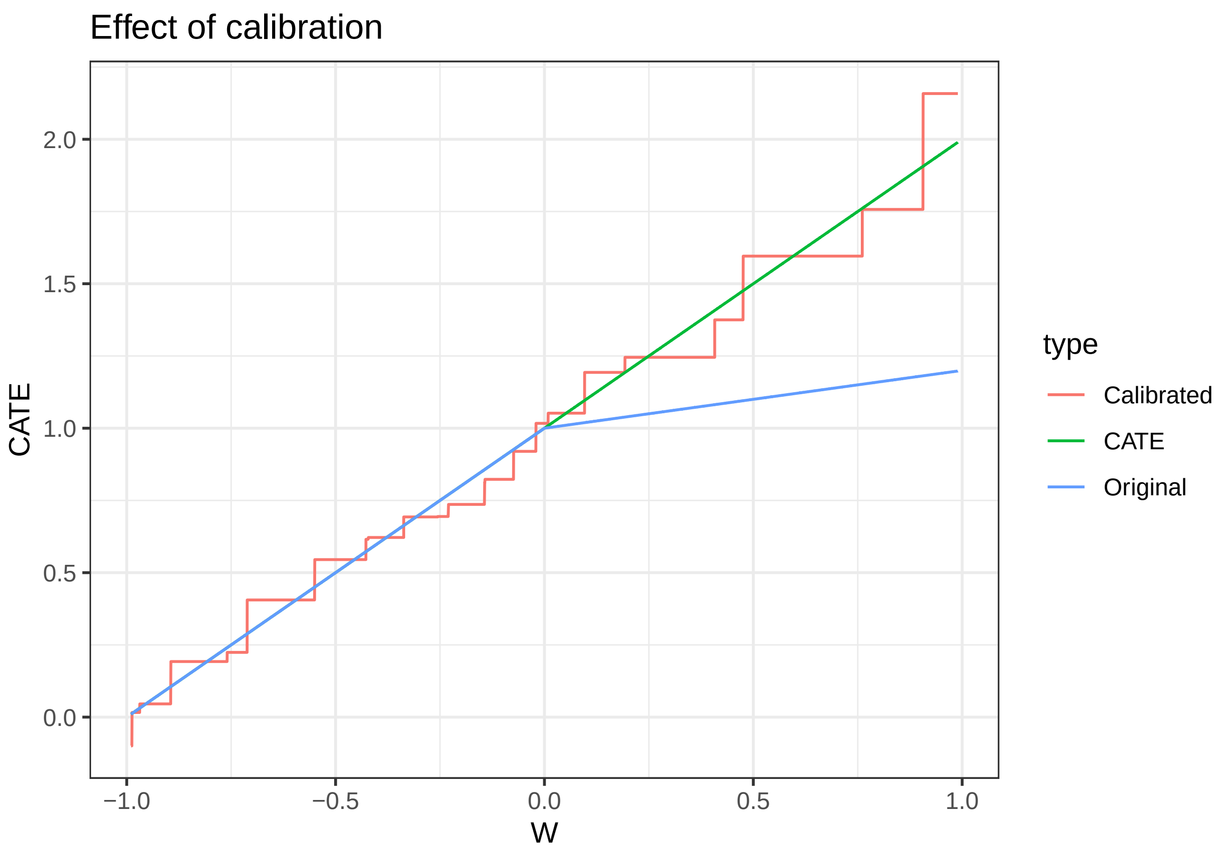
```
tau <-  1 + W +sin(5*W)
tau_new <- 1 + Wnew+ sin(5*Wnew)


calibrator <- causalCalibrate(tau, A, Y, EY1, EY0, pA1, tau_pred = tau_new)
library(ggplot2)
library(latex2exp)
tau_cal <- calibrator$tau_calibrated
data <- data.frame(W = Wnew, CATE = c(CATE, tau_new, tau_cal), type = rep(c("CATE", "Original", "Calibra

ggplot(data, aes(x = W, y = CATE, color = type)) + geom_line()  + ggtitle("Effect of calibration")   + s
```



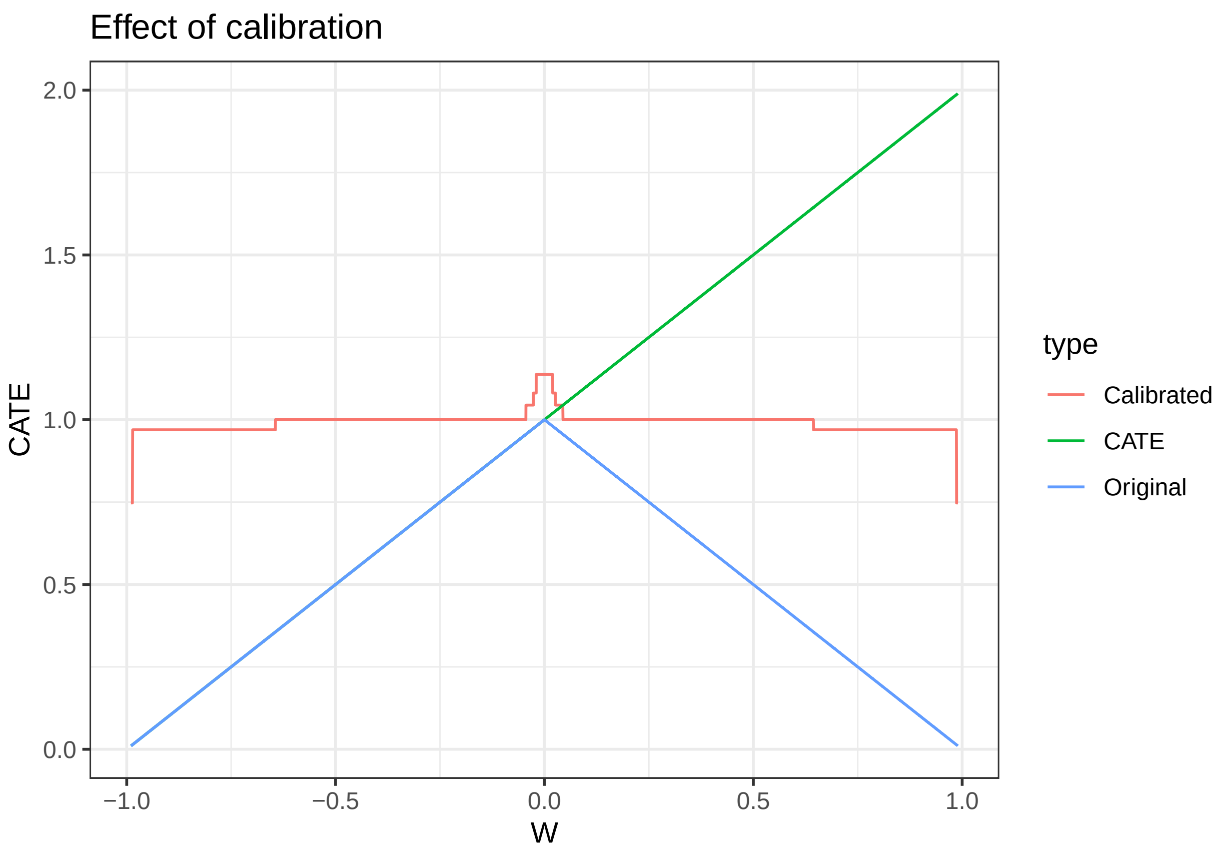
```
tau <-  1 + W
tau_new <- 1 + Wnew


calibrator <- causalCalibrate(tau, A, Y, EY1, EY0, pA1, tau_pred = tau_new)
library(ggplot2)
library(latex2exp)
tau_cal <- calibrator$tau_calibrated
data <- data.frame(W = Wnew, CATE = c(CATE, tau_new, tau_cal), type = rep(c("CATE", "Original", "Calibra

ggplot(data, aes(x = W, y = CATE, color = type)) + geom_line()  + ggtitle("Effect of calibration")   + s
```



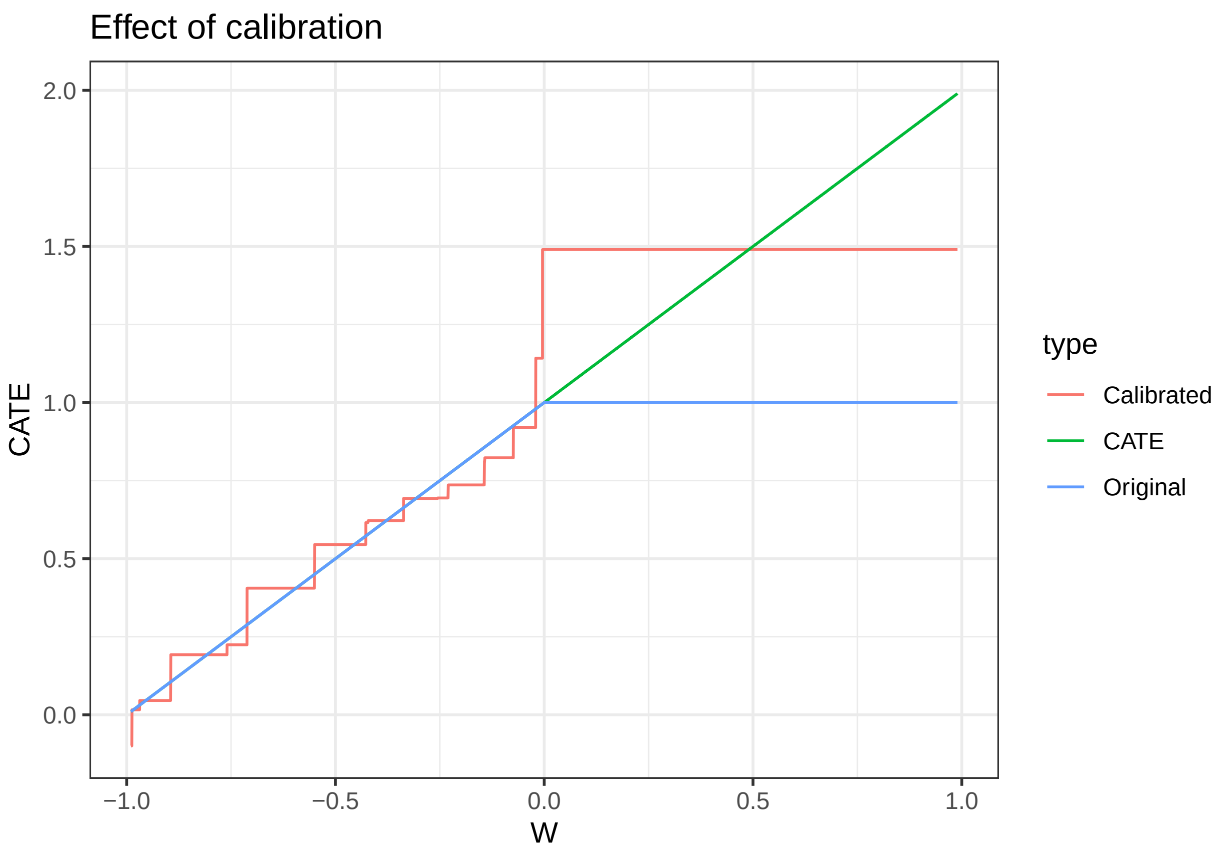
```
        tau <-  1 + W + (W >= 0) * ( -0.8*   W)
tau_new <- 1 + Wnew + (Wnew >= 0) * ( -0.8* Wnew)


calibrator <- causalCalibrate(tau, A, Y, EY1, EY0, pA1, tau_pred = tau_new)
library(ggplot2)
library(latex2exp)
tau_cal <- calibrator$tau_calibrated
data <- data.frame(W = Wnew, CATE = c(CATE, tau_new, tau_cal), type = rep(c("CATE", "Original", "Calibra

ggplot(data, aes(x = W, y = CATE, color = type)) + geom_line()  + ggtitle("Effect of calibration")   + s
```



```
  tau <-  1 + W + (W >= 0) * ( -2*   W)
tau_new <- 1 + Wnew + (Wnew >= 0) * ( -2* Wnew)


calibrator <- causalCalibrate(tau, A, Y, EY1, EY0, pA1, tau_pred = tau_new)
library(ggplot2)
library(latex2exp)
tau_cal <- calibrator$tau_calibrated
data <- data.frame(W = Wnew, CATE = c(CATE, tau_new, tau_cal), type = rep(c("CATE", "Original", "Calibra

ggplot(data, aes(x = W, y = CATE, color = type)) + geom_line()  + ggtitle("Effect of calibration")   + s
```



```
    tau <-  1 + W + (W >= 0) * ( -1*   W)
tau_new <- 1 + Wnew + (Wnew >= 0) * ( -1* Wnew)


calibrator <- causalCalibrate(tau, A, Y, EY1, EY0, pA1, tau_pred = tau_new)
library(ggplot2)
library(latex2exp)
tau_cal <- calibrator$tau_calibrated
data <- data.frame(W = Wnew, CATE = c(CATE, tau_new, tau_cal), type = rep(c("CATE", "Original", "Calibra

ggplot(data, aes(x = W, y = CATE, color = type)) + geom_line()  + ggtitle("Effect of calibration")   + 
```



cross-calibration

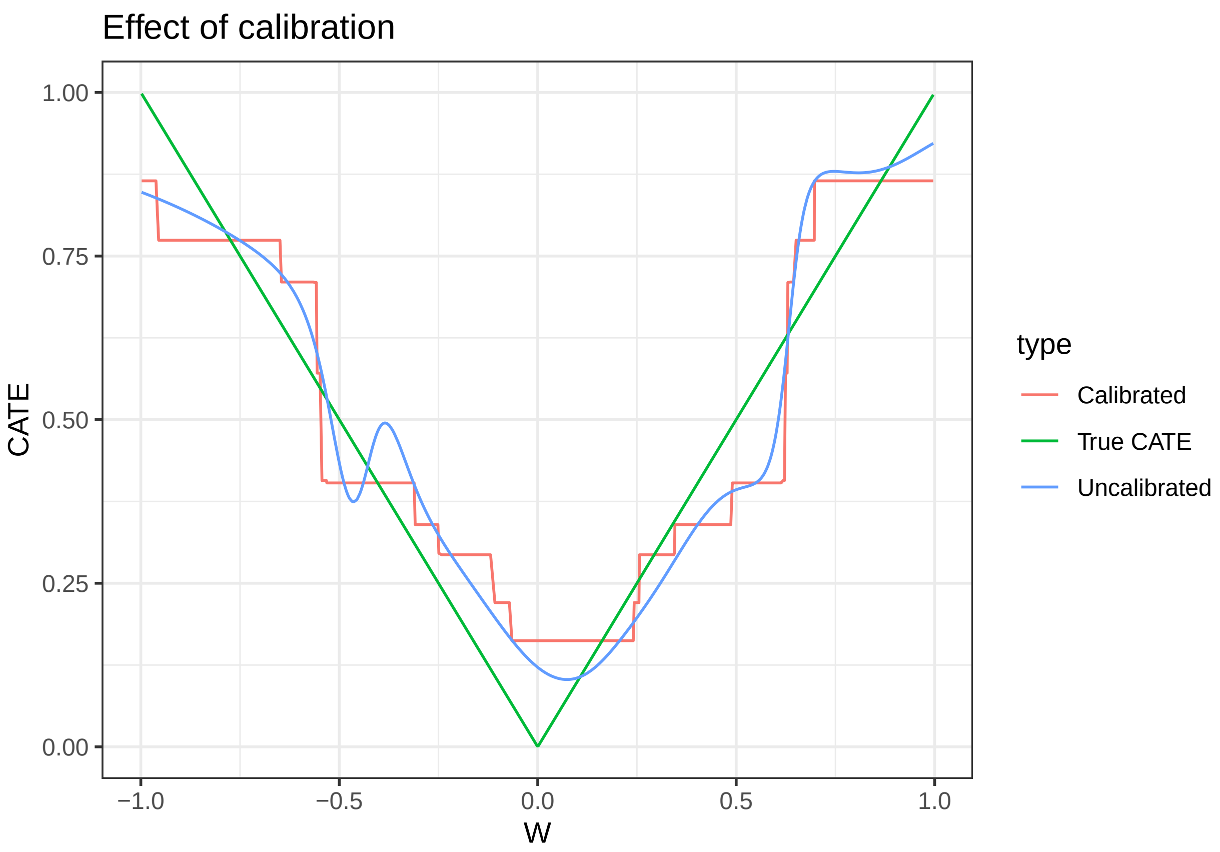
cross-calibration with nnet

```
library(sl3) #tlverse/sl3 on github. Used for cross fitting
library(data.table)

n <- 1000
W <- runif(n, -1 , 1)
pA1 <- plogis(1.5*W)
A <- rbinom(n, size = 1, pA1)
# True CATE(W) = 1 + W
CATE <- abs(W)
EY0 <- W
EY1 <- W + CATE
EY <- ifelse(A==1, EY1, EY0)
Y <- rnorm(n, W + A * CATE, 0.3)
zeta <- EY1 - EY0 + (A - pA1) /( (1-pA1)*pA1) *(Y - EY)
data <- data.table(W, zeta = zeta)

task <- sl3_Task$new(data, covariates = c("W"), outcome = "zeta", folds = 10)
# choose between base_learners

base_learner <- Lrnr_nnet$new(size = 25) # single layer neural net
```

```
lrnr_tau <- Lrnr_cv$new(base_learner, full_fit = TRUE)
#train predictor
lrnr_tau <- lrnr_tau$train(task) # does both cross fitting and full sample fitting
```

```
## # weights:  76
## initial  value 518.501726
## iter  10 value 468.633767
## iter  20 value 444.039940
## iter  30 value 443.818666
## iter  40 value 443.337846
## iter  50 value 442.321231
## iter  60 value 442.256032
## iter  70 value 442.220098
## iter  80 value 442.120112
## iter  90 value 441.621280
## iter 100 value 440.594066
## final  value 440.594066
## stopped after 100 iterations
## # weights:  76
## initial  value 505.846295
## iter  10 value 426.143603
## iter  20 value 413.695303
## iter  30 value 412.835717
## iter  40 value 412.707629
## iter  50 value 412.538402
## iter  60 value 412.052000
## iter  70 value 410.912299
## iter  80 value 410.065315
## iter  90 value 408.438045
## iter 100 value 406.485747
## final  value 406.485747
## stopped after 100 iterations
## # weights:  76
## initial  value 445.626837
## iter  10 value 383.049392
## iter  20 value 380.042644
## iter  30 value 379.714645
## iter  40 value 379.509674
## iter  50 value 379.201075
## iter  60 value 379.055335
## iter  70 value 378.940975
## iter  80 value 378.831538
## iter  90 value 378.645762
## iter 100 value 378.608403
## final  value 378.608403
## stopped after 100 iterations
## # weights:  76
## initial  value 496.426464
## iter  10 value 416.312793
## iter  20 value 414.684136
## iter  30 value 414.615697
## iter  40 value 414.345428
## iter  50 value 413.500406
## iter  60 value 413.222365
## iter  70 value 413.068851
## iter  80 value 412.974226
## iter  90 value 412.922173
## iter 100 value 412.580365
## final  value 412.580365
## stopped after 100 iterations
## # weights:  76
## initial  value 470.106282
## iter  10 value 396.700352
## iter  20 value 396.408951
## iter  30 value 396.356093
## iter  40 value 396.283115
## iter  50 value 395.830913
## iter  60 value 394.959228
## iter  70 value 394.510892
## iter  80 value 394.357024
## iter  90 value 394.209165
## iter 100 value 394.156998
## final  value 394.156998
## stopped after 100 iterations
## # weights:  76
## initial  value 461.542562
## iter  10 value 418.254781
## iter  20 value 396.593819
## iter  30 value 396.387349
## iter  40 value 396.096452
## iter  50 value 394.492864
## iter  60 value 394.252163
## iter  70 value 394.144664
## iter  80 value 394.017489
## iter  90 value 393.773345
## iter 100 value 393.611289
## final  value 393.611289
## stopped after 100 iterations
## # weights:  76
## initial  value 546.395285
## iter  10 value 409.107891
## iter  20 value 402.186711
## iter  30 value 402.049518
## iter  40 value 402.031613
## iter  50 value 401.899781
## iter  60 value 401.357854
## iter  70 value 400.612818
## iter  80 value 400.219777
## iter  90 value 399.736495
## iter 100 value 397.714227
## final  value 397.714227
## stopped after 100 iterations
## # weights:  76
## initial  value 542.268700
## iter  10 value 392.630702
## iter  20 value 383.034267
## iter  30 value 382.663268
## iter  40 value 382.236619
## iter  50 value 381.898344
## iter  60 value 381.537709
## iter  70 value 381.270969
## iter  80 value 381.061264
## iter  90 value 380.785642
## iter 100 value 380.489676
## final  value 380.489676
## stopped after 100 iterations
## # weights:  76
## initial  value 489.237734
## iter  10 value 420.517828
## iter  20 value 406.837446
## iter  30 value 406.379995
## iter  40 value 405.503571
## iter  50 value 403.901741
## iter  60 value 402.563705
## iter  70 value 401.411274
## iter  80 value 399.755684
## iter  90 value 399.097412
## iter 100 value 398.832809
## final  value 398.832809
## stopped after 100 iterations
## # weights:  76
## initial  value 589.588123
## iter  10 value 418.178488
## iter  20 value 405.200999
## iter  30 value 404.906084
## iter  40 value 404.310263
## iter  50 value 403.621169
## iter  60 value 403.508321
## iter  70 value 403.438642
## iter  80 value 403.315428
## iter  90 value 403.246354
## iter 100 value 403.131705
## final  value 403.131705
## stopped after 100 iterations
## # weights:  76
## initial  value 483.699286
## iter  10 value 402.074035
## iter  20 value 393.018130
## iter  30 value 392.882001
## iter  40 value 392.634871
## iter  50 value 391.769264
## iter  60 value 391.519052
## iter  70 value 391.437239
## iter  80 value 391.397336
## iter  90 value 391.356254
## iter 100 value 391.338767
## final  value 391.338767
## stopped after 100 iterations
```

```r
tau_no_crossfit <-  lrnr_tau$predict_fold(task, "full") # predictions using full predictor fit on all d
tau_out_of_fold <- lrnr_tau$predict_fold(task, "validation") # pooled out of fold predictions from cros
tau_all <- do.call(cbind, lapply(1:10, function(k) {
  lrnr_tau$predict_fold(task, k) # fold-specific predictions
})) # Get full sample predictions from each fold-specific predictor.

# Compute calibrator from data
calibrator <- causalCalibrate(tau = tau_out_of_fold, A = A, Y = Y, EY1 = EY1, EY0 = EY0, pA1 = pA1)


# tau_all should be the stacked cross-fitted prediction matrix for observations at which we would like
tau_cal <- cross_calibrate(calibrator, tau_all)


data <- data.frame(W = W, CATE = c(CATE, tau_no_crossfit, tau_cal), type = rep(c("True CATE", "Uncalibra

ggplot(data, aes(x = W, y = CATE, color = type)) + geom_line()  + ggtitle("Effect of calibration")   + 
```



```r
# We improve mean-squared error by calibrating.
mean((tau_no_crossfit - CATE)^2)
```

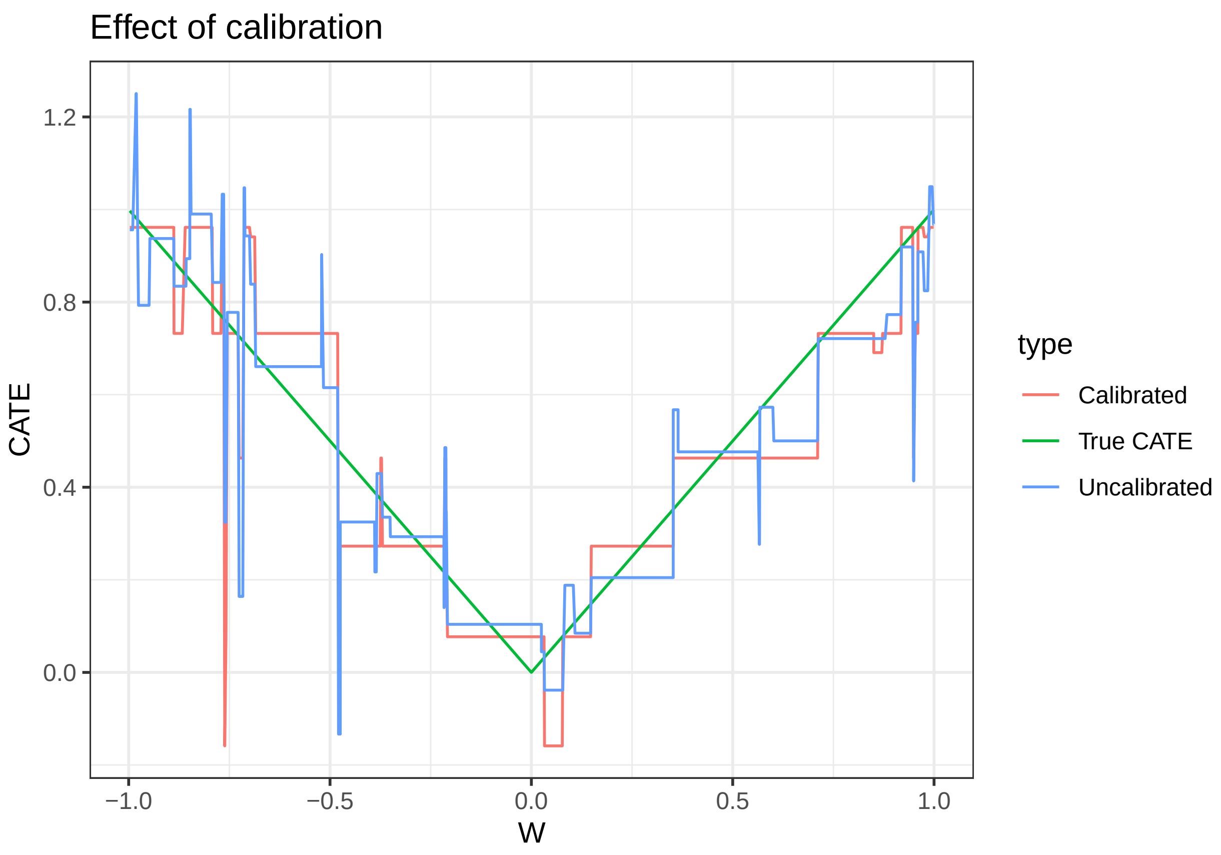
```
## [1] 0.00684028
```

```r
mean((tau_cal - CATE)^2)
```

```
## [1] 0.009296488
```

cross-calibration with xgboost

```r
library(sl3) #tlverse/sl3 on github. Used for cross fitting
library(data.table)

n <- 1000
W <- runif(n, -1 , 1)
pA1 <- plogis(1.5*W)
A <- rbinom(n, size = 1, pA1)
# True CATE(W) = 1 + W
CATE <- abs(W)
EY0 <- W
EY1 <- W + CATE
EY <- ifelse(A==1, EY1, EY0)
Y <- rnorm(n, W + A * CATE, 0.3)
zeta <- EY1 - EY0 + (A - pA1) /( (1-pA1)*pA1) *(Y - EY)
data <- data.table(W, zeta = zeta)

task <- sl3_Task$new(data, covariates = c("W"), outcome = "zeta", folds = 10)
# choose between base_learners
base_learner <- Lrnr_xgboost$new(max_depth = 3, nrounds = 10) # xgboost




lrnr_tau <- Lrnr_cv$new(base_learner, full_fit = TRUE)
#train predictor
lrnr_tau <- lrnr_tau$train(task) # does both cross fitting and full sample fitting

tau_no_crossfit <-  lrnr_tau$predict_fold(task, "full") # predictions using full predictor fit on all d
tau_out_of_fold <- lrnr_tau$predict_fold(task, "validation") # pooled out of fold predictions from cros
tau_all <- do.call(cbind, lapply(1:10, function(k) {
  lrnr_tau$predict_fold(task, k) # fold-specific predictions
})) # Get full sample predictions from each fold-specific predictor.

# Compute calibrator from data
calibrator <- causalCalibrate(tau = tau_out_of_fold, A = A, Y = Y, EY1 = EY1, EY0 = EY0, pA1 = pA1)


# tau_all should be the stacked cross-fitted prediction matrix for observations at which we would like
tau_cal <- cross_calibrate(calibrator, tau_all)


data <- data.frame(W = W, CATE = c(CATE, tau_no_crossfit, tau_cal), type = rep(c("True CATE", "Uncalibra

ggplot(data, aes(x = W, y = CATE, color = type)) + geom_line()  + ggtitle("Effect of calibration")   + 
```



```r
# We improve mean-squared error by calibrating.
mean((tau_no_crossfit - CATE)^2)
```

```
## [1] 0.01335983
```

```r
mean((tau_cal - CATE)^2)
```

```
## [1] 0.0154369
```