Simulation Experiments

Rafael Alcantara

2025-01-02

DGP

Let X denote the running variable and W denote a continuous feature. The data-generating processes investigated in this experiment can be summarized by the following general structure. First, the prognostic and treatment functions are constructed as:

$$\begin{aligned}
\mu_{0,x}(X) &= -0.03X^5 + K_1X^3 + 0.1X^2 - 0.1X + 1.72 \\
\mu_{0,w}(W) &= K_2\cos(W) \\
\tau_{0,x}(X) &= 1/(1-\exp(-4-X)) - 1.02 \\
\tau_{0,w}(W) &= \sin(W) \\
\mu(X,W) &= \mu_{0,x}(X) + \mu_{0,w}(W) \\
\tau(X,W) &= \tau_{0,x}(X) + \tau_{0,w}(W) + \bar{\tau} \\
K_1 &\in \{0.5, 2.5\} \\
K_2 &\in \{1.4, 4.2\}.
\end{aligned} \tag{1}$$

Features X, W, treatment variable Z and outcome Y are generated as:

$$\begin{aligned}
X &\sim \mathbb{N}(0,1) \\
Z &= \nVdash(X \geq 0) \\
W &\sim \mathbb{N}(\rho X, \sqrt{1-\rho^2}) \\
\rho &\in \{0.5, 0.9\} \\
Y &\sim \mathbb{N}(\mu(X,W) + \tau(X,W)Z, 1).
\end{aligned} \tag{2}$$

Each DGP is defined by a tuple K_1, K_2, ρ. Parameter K_1 controls how fast $\mu_{0,x}(X)$ moves away from $\mu_{0,x}(X=0)$. This is relevant particularly for S-BART and T-BART, since these estimators provide no control over how points far from the cutoff affect predictions at that point. Consider a split in W such that points with both X close to or far from $X = c$ are included in the same node. If $E[Y \mid X, W]$ differs substantially between these two regions of the support of X, such nodes will provide very poor approximations for $E[Y \mid X = c, W, Z = 1] - E[Y \mid X = c, W, Z = 0]$, compromising the performance of the unmodified BART estimators. BARDDT avoids this issue by significantly increasing the likelihood that these nodes will split in X, and imposing that they must be split until all nodes which contain the cutoff region feature only few points with X outside our proposed window. The polynomial estimator avoids this issue by discarding points outside their proposed window. Therefore, parameter K_1 allows us to investigate how much this should be an issue for S-BART and T-BART.

Parameter K_2 controls the variability due to W in μ compared to τ. Because RDD treatment effects are only identified at $X = c$, the variability in W in each function is what matters most to determine how hard it is to learn the treatment effect function. In particular, we expect larger values of K_2 to make the problem harder.

The specific grid for this parameter was chosen such that $sd(\mu(X = c, W))$ is, respectively, one and three times greater than $sd(\tau(X = c, W))$.

INCLUDE PRECISE EXPLANATION OF WHY WE SHOULD DEAL BETTER WITH THIS ISSUE THAN THE OTHERS

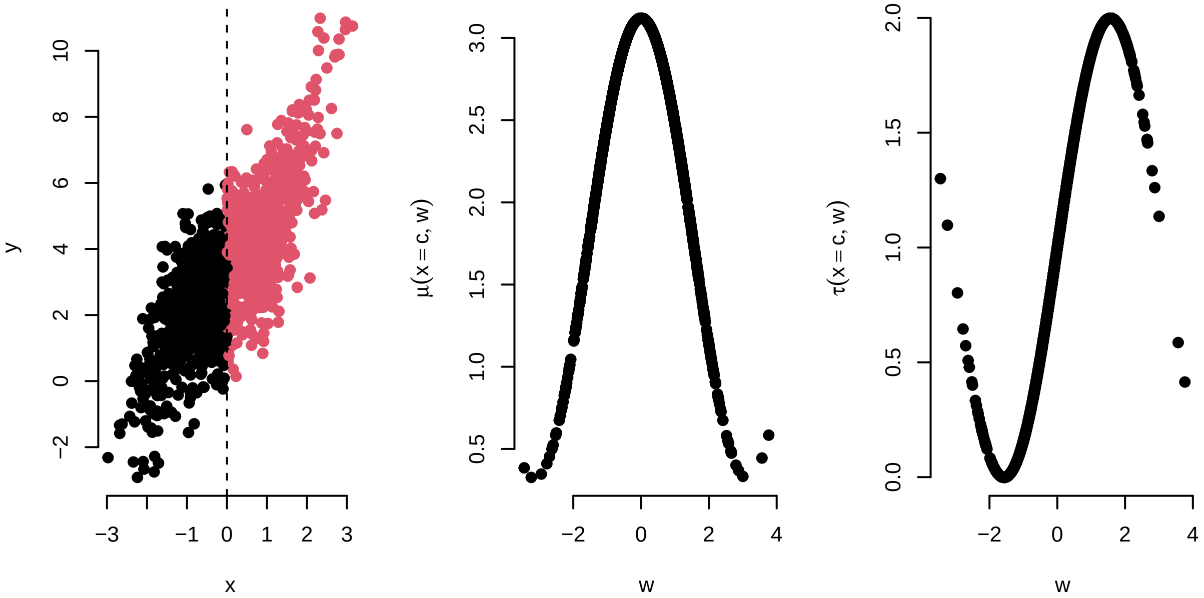
Finally, parameter ρ is the correlation between X and W. This is parameter can affect CATE estimation in the RDD significantly. If certain values of W are more likely to be observed on one side of $X = c$ than the other, the more likely it is that the value of W for points used to construct $E[Y \mid X = c, W, Z = 1]$ are very different than those for points used to construct $E[Y \mid X = c, W, Z = 0]$. This can lead to problems in extrapolating these functions to $X = c$ if they vary much in W (which is a reasonable scenario to consider in real applications if one is estimating CATE in the first place). BARDDT controls this by making sure that any node containing the point $X = c$ features a minimum number of points from both sides of the cutoff, which is not guaranteed to happen with any of the other estimators. Because we generate (X, W) as bivariate Gaussian, ρ allows us to control the distribution of $W \mid X \sim c$ more directly.

For each DGP, we generate 1000 samples of size $N \in \{500, 1000, 1500\}$. The figures below illustrate some features of each DGP for $N = 1500$.

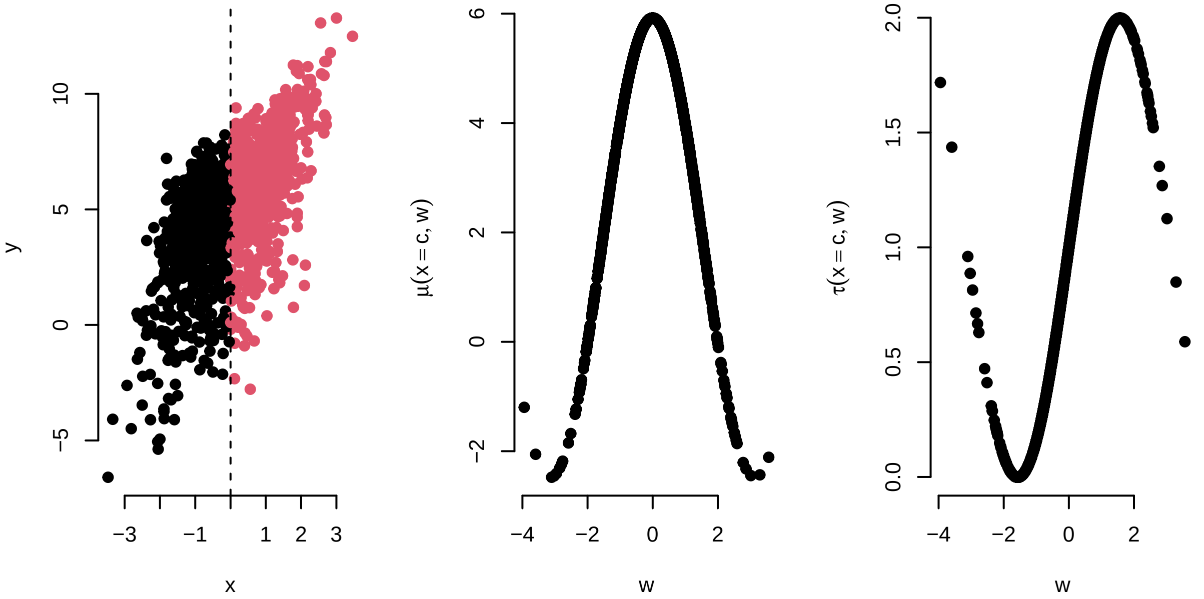
```
set.seed(7)
### Functions
mu0.x <- function(x,k) -0.03*x^5 + k*0.5*x^3 + 0.1*x^2 - 0.1*x + 1.72
mu0.w <- function(w,k) k*1.4*cos(w)
tau0.x <- function(x,c) 1/(1-exp(-4-x)) - 1.02
tau0.w <- function(w) sin(w)
mu <- function(x,w,k1,k2) mu0.x(x,k1) + mu0.w(w,k2)
tau <- function(x,c,w,ate) tau0.x(x,c) + tau0.w(w) + ate
h.grid <- function(x,c,grid)
{
  abs.x <- sort(abs(x-c))
  out <- rep(0,length(grid))
  names(out) <- grid
  x.right <- sum(c < x)
  x.left <- sum(x < c)
  x.tot <- length(x)
  for(total in grid)
  {
    i <- 1
    sum.right <- sum.left <- 0
    while(sum.right < total | sum.left < total)
    {
      sum.left <- sum(c-abs.x[i] <= x & x < c)
      sum.right <- sum(c < x & x <= c+abs.x[i])
      if (sum.left == sum(x<c) & sum.right == sum(c<x)) break
      i <- i+1
    }
    out[as.character(total)] <- abs.x[i]
  }
  return(out)
}
## Parameters
n <- 1500
rho <- c(0.5,0.9)
k1 <- c(1,5)
k2 <- c(1,3)
pts_in_window <- 75
```

```
sig_error <- 1
c <- 0
ate <- 1
for (Rho in rho)
{
  for (K1 in k1)
  {
    for (K2 in k2)
    {
      ## Generate data
      x <- rnorm(n)
      h <- h.grid(x,c,pts_in_window)
      test <- -h<x & x<h
      z <- as.numeric(x>=c)
      w <- rnorm(n,Rho*x,sqrt(1-Rho^2))
      cate <- tau(c,c,w,ate)
      prog <- mu(c,w,K1,K2)
      Ey <- mu(x,w,K1,K2) + tau(x,c,w,ate)*z
      y <- Ey + rnorm(n,0,sqrt(sig_error))
      title <- bquote(N==.(n)
                      ~";"~rho==.(Rho)
                      ~";"~K1==.(K1*0.5)
                      ~";"~K2==.(K2*1.4))
      ## Plot data
      par(bty="n",pch=19)
      layout(matrix(c(1,2,1,3,1,4),ncol=3),height=c(1,3))
      par(mar=c(2,2,1,1))
      plot.new()
      text(0.5,0.5,title,cex=2,font=2)
      ###
      par(mar=c(5,5,1,1))
      plot(x,y,col=z+1)
      abline(v=c,lty=2)
      ###
      plot(w,prog,ylab=bquote(mu(x==c,w)))
      ###
      plot(w,cate,ylab=bquote(tau(x==c,w)))
    }
  }
}
```

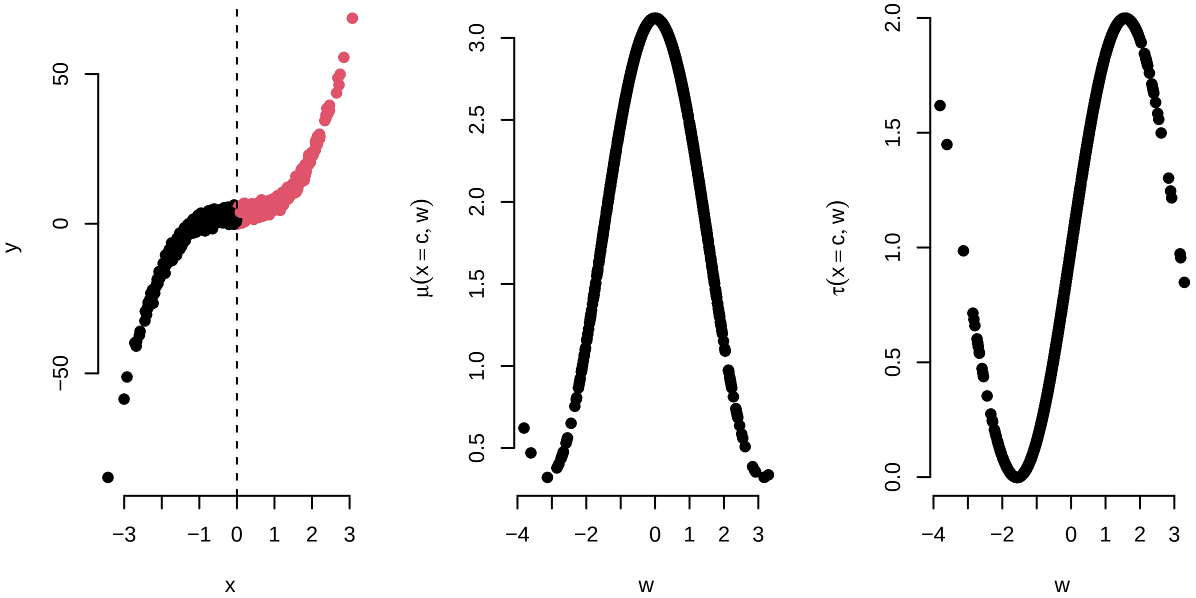
N = 1500 ; ρ = 0.5 ; K1 = 0.5 ; K2 = 1.4



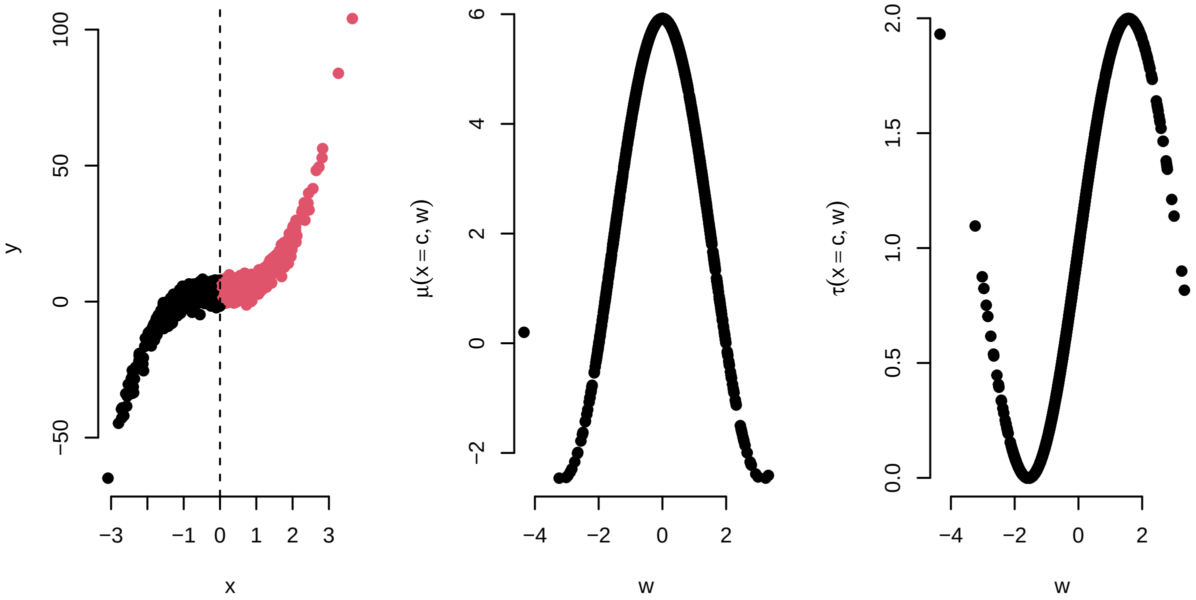
N = 1500 ; ρ = 0.5 ; K1 = 0.5 ; K2 = 4.2



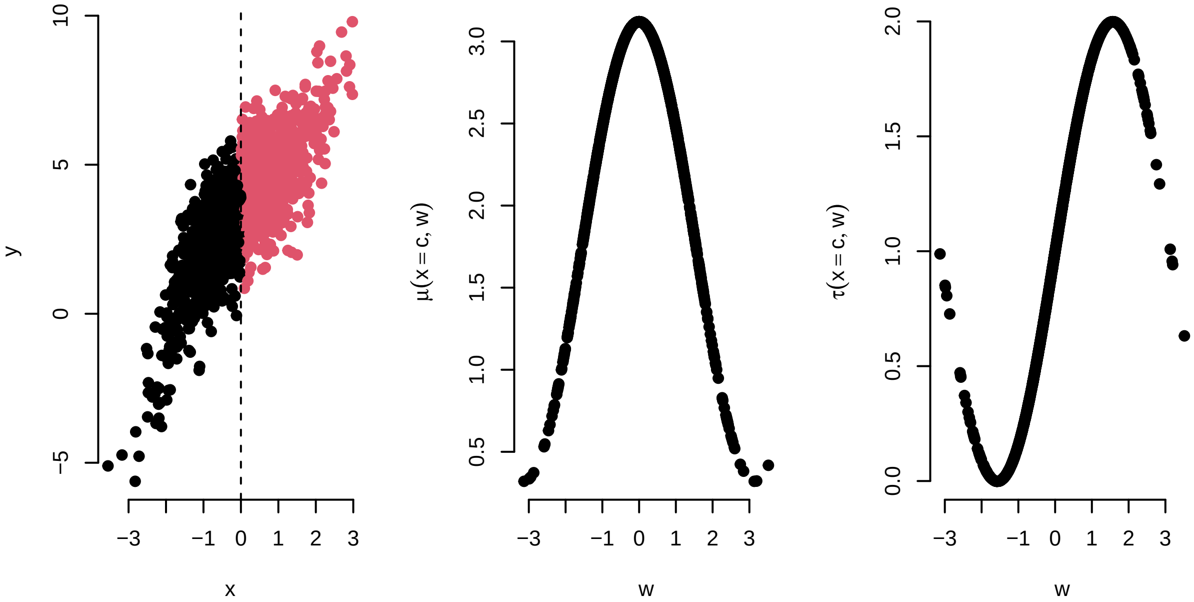
$N = 1500 \; ; \; \rho = 0.5 \; ; \; K1 = 2.5 \; ; \; K2 = 1.4$



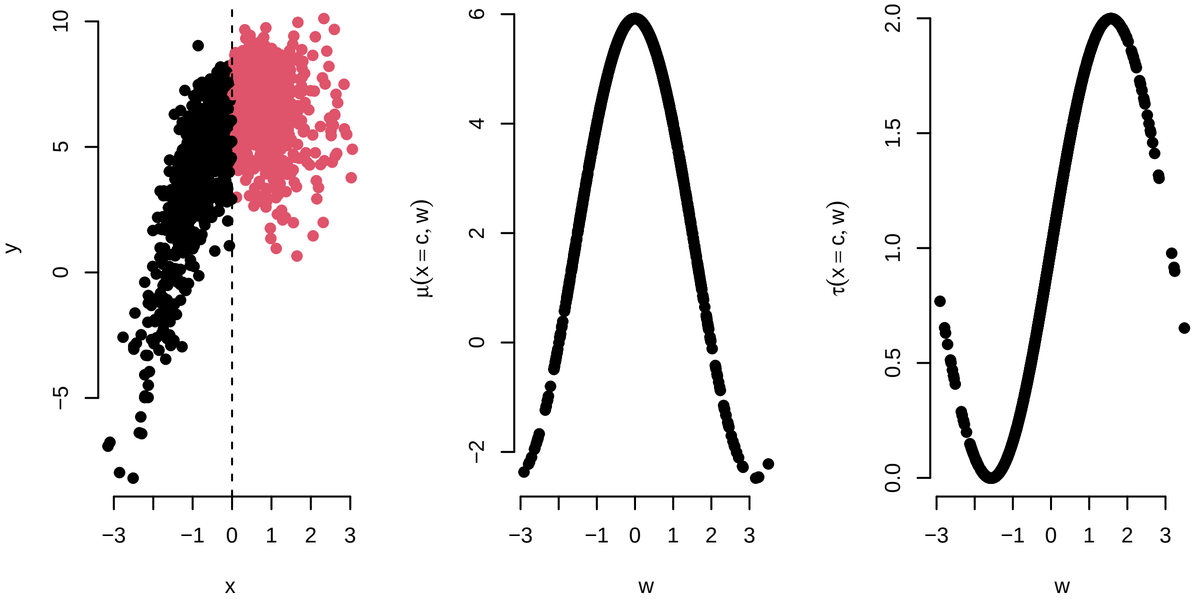
$N = 1500 \; ; \; \rho = 0.5 \; ; \; K1 = 2.5 \; ; \; K2 = 4.2$



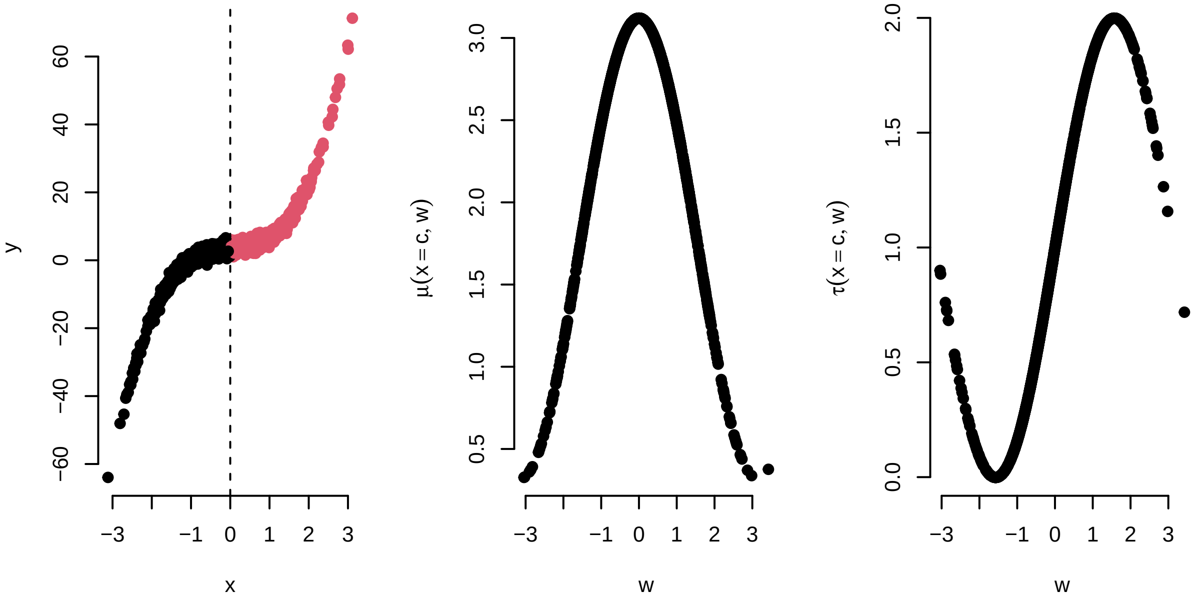
$N = 1500 ; \rho = 0.9 ; K1 = 0.5 ; K2 = 1.4$



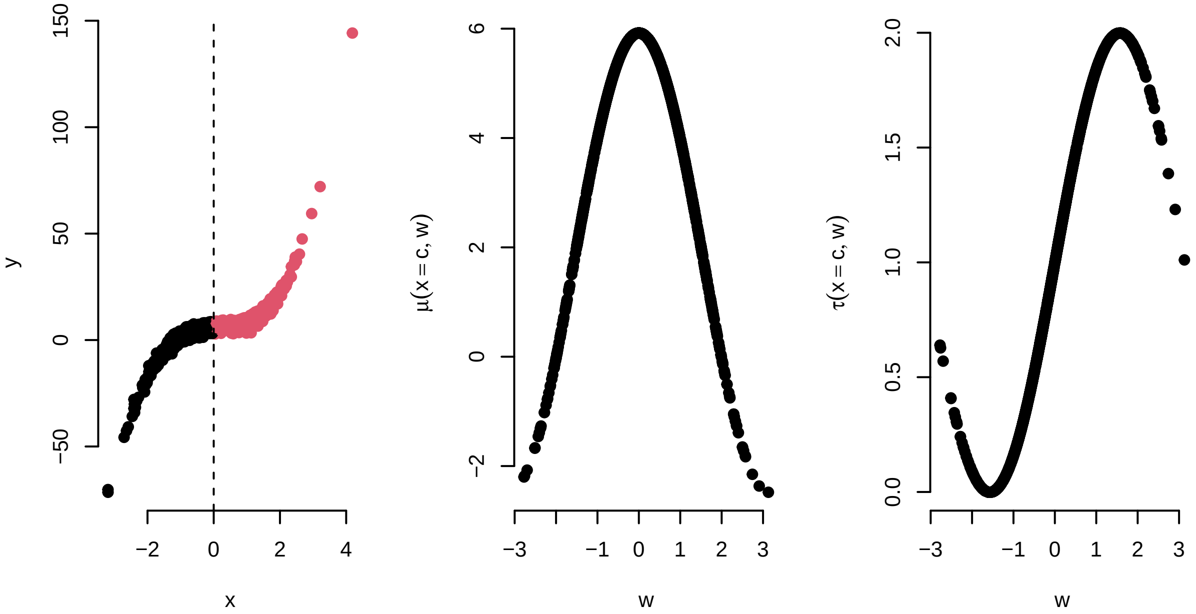
$N = 1500 ; \rho = 0.9 ; K1 = 0.5 ; K2 = 4.2$



N = 1500 ; ρ = 0.9 ; K1 = 2.5 ; K2 = 1.4



N = 1500 ; ρ = 0.9 ; K1 = 2.5 ; K2 = 4.2


y
150
100
50
0
−50
−2
0
2
4
x
μ(x = c, w)
6
4
2
0
−2
−3
−1
0
1
2
3
w
τ(x = c, w)
2.0
1.5
1.0
0.5
0.0
−3
−1
0
1
2
3
w