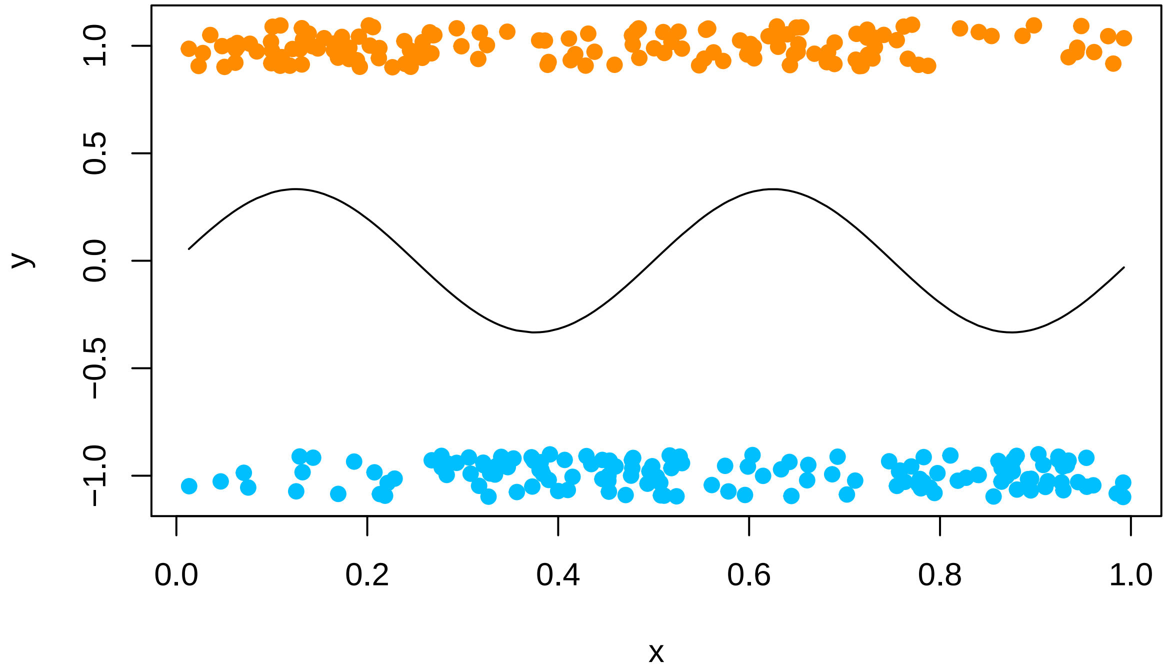
Author: Bin Dong

Email: bindong2@illinois.edu

Question 1

```
rm(list=ls())

set.seed(1)

n = 300
x = runif(n)
py <- function(x) sin(4*pi*x)/3 + 0.5
y = (rbinom(n, 1, py(x))-0.5)*2
plot(x, y + 0.1*runif(n, -1, 1), ylim = c(-1.1, 1.1), pch = 19,
col = ifelse(y == 1, "darkorange", "deepskyblue"), ylab = "y")
lines(sort(x), py(x)[order(x)] - 0.5)
```



```
data <- data.frame(x=x, y=y, weight=1/length(x))
```

```
stump <- function(data)
{
  n <- nrow(data)
  data_sorted <- data[order(data$x),]
  scores <- c(rep(0, n-2))
  criterias <- c(rep(0, n-2))
  for(i in 2:n-2)
  {
    criteria = data_sorted$x[i+1]
    left <- subset(data_sorted, x<=criteria)
    right <- subset(data_sorted, x>criteria)
    p_left <- sum((subset(left, y==1))$weight)/sum(left$weight)
    p_right <- sum((subset(right, y==1))$weight)/sum(right$weight)
    gini_left <- p_left * (1-p_left)
    gini_right <- p_right *(1-p_right)

    score <- -sum(left$weight)*gini_left - sum(right$weight)*gini_right

    scores[i-1] = score
    criterias[i-1] = criteria

  }
  index <- which.max(scores)
  return <- criterias[index]
}

criteria <- stump(data)
print(criteria)
```

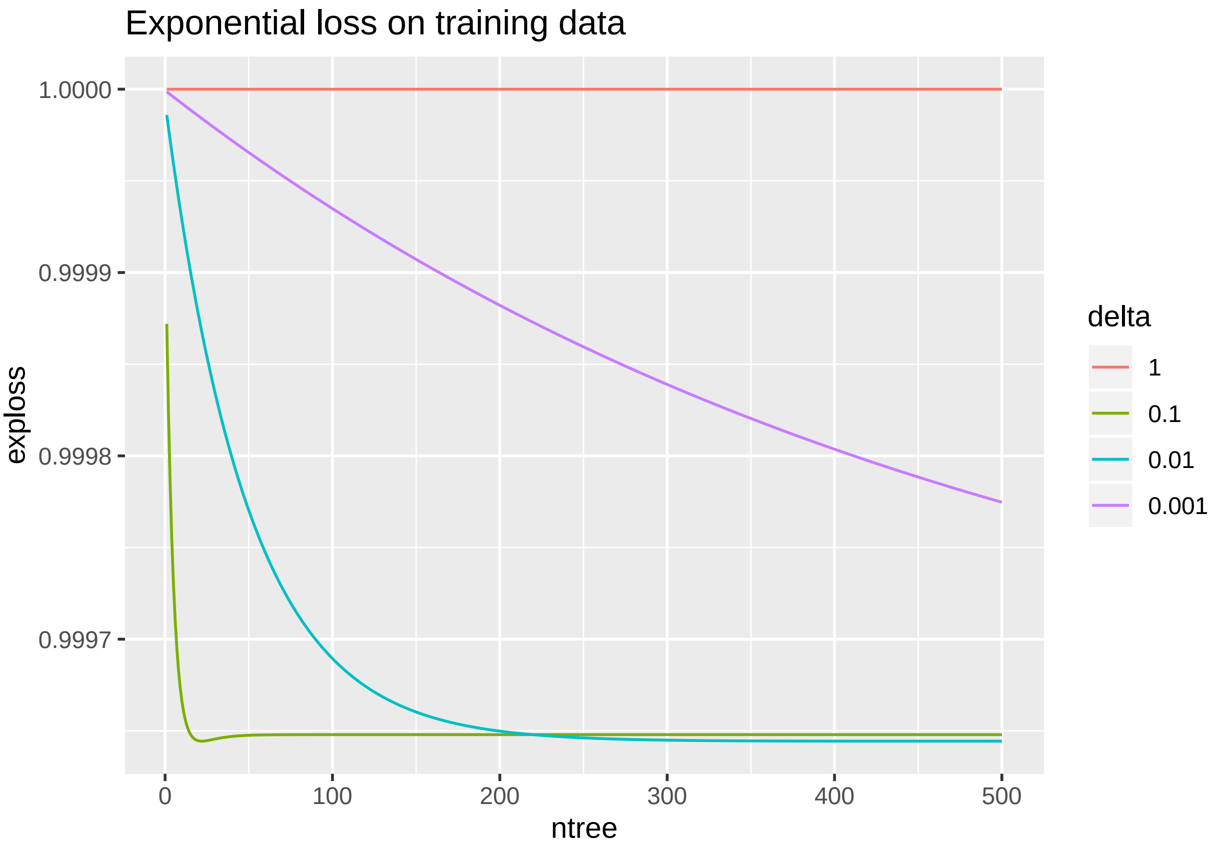
```
## [1] 0.2702601
```

Adaboost

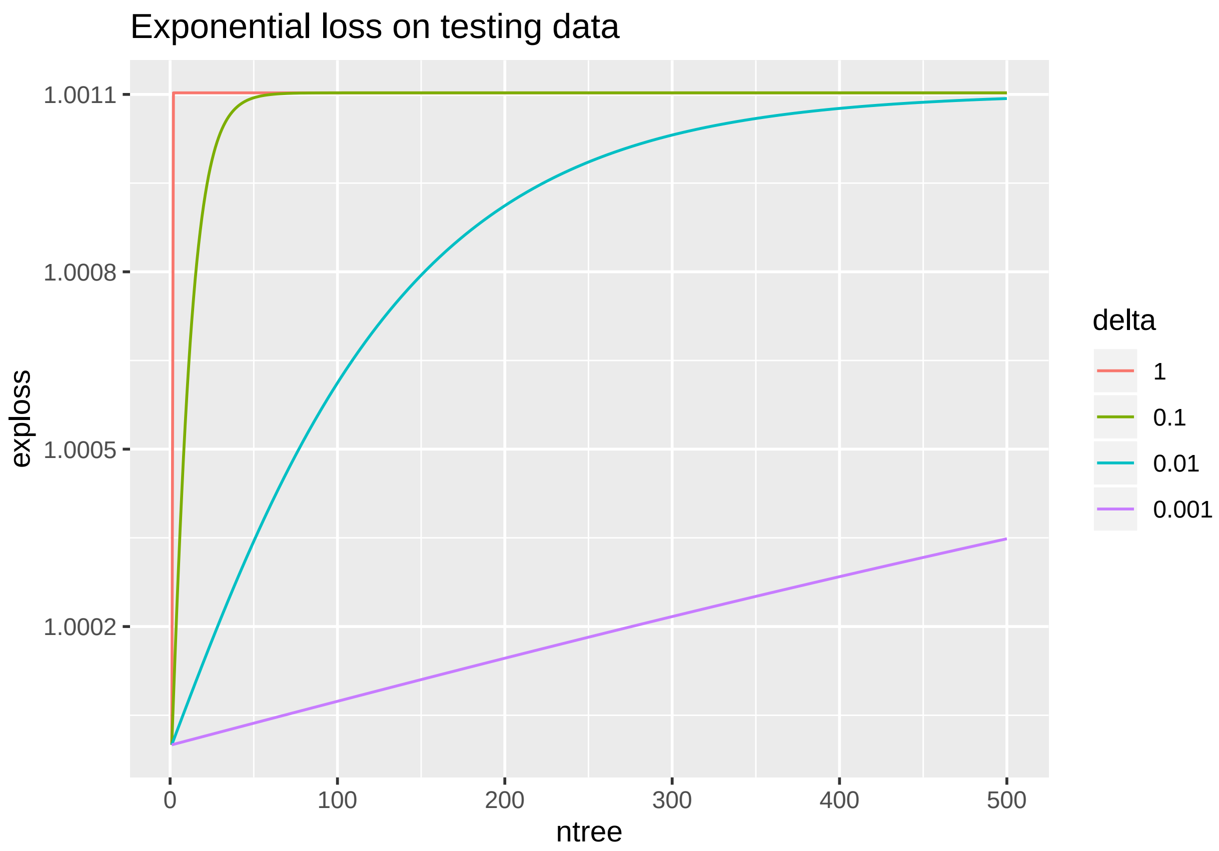
```
adaboost <- function(P, data, delta)
{
  n = nrow(data)
  c = stump(data)

  left <- subset(data, x<=criteria)
  right <- subset(data, x>criteria)

  left_majority <- ifelse(sum(left$weight)>0, 1, -1)
  right_majority <- ifelse(sum(right$weight)>0, 1, -1)

  prediction <- ifelse(x<=c, left_majority, right_majority)

  eta <- sum((1-data$y*prediction)*data$weight/2)

  alpha <- 0.5*delta*log((1-eta)/eta)

  P <- P + alpha*prediction

  weight_updated <- data$weight*exp(-alpha*y*prediction)
  data$weight <- weight_updated/sum(weight_updated)

  return <- list(P=P, data=data, prediction=sign(alpha)*prediction, eta=min(eta, 1-eta), alpha=alpha, c
}
```

```
# Test data
testx = seq(0, 1, length.out = 1000)
testy = (rbinom(1000, 1, py(testx))-0.5)*2


ntree=500;
n = length(x);
deltas = c(1, 0.1, 0.01, 0.001)
exp_loss.matrix <- matrix(ncol=ntree, nrow=length(deltas))

n_test = length(testx);
exp_loss.predict.matrix <- matrix(ncol=ntree, nrow=length(deltas))

get_loss <- function(y, P, n)
{
  return <- sum(exp(-y*P))/n
}


for(j in 1:length(deltas))
{
  delta <- deltas[j]
  exp_loss <- rep(0,ntree);
  exp_loss.predict <- rep(0,ntree);
  P <- rep(0,n);
  P.pred=rep(0,n_test)

  data$weight <- 1/n
  result <- adaboost(P, data, delta);


  for (i in 1:ntree)
  {
    result=adaboost(result$P, data, delta);

    exp_loss[i] <- get_loss(y, result$P, n)
    exp_loss.predict[i] <- get_loss(testy, P.pred, n_test)

    predictions <- ifelse(testx <= result$criteria, result$left, result$right)

    P.pred <- P.pred + result$alpha*predictions
    data <- result$data

  }

  exp_loss.matrix[j,] <- exp_loss
  exp_loss.predict.matrix[j,] <- exp_loss.predict
}
```

```r
library(ggplot2)

exp_loss.frame <- data.frame(t(exp_loss.matrix))
colnames(exp_loss.frame) <- deltas
exp_loss.frame$ntree <- c(1:ntree)
exp_loss.melted <- reshape2::melt(exp_loss.frame, id.var='ntree')
colnames(exp_loss.melted) <- c("ntree", "delta","exploss")

# testing data loss
exp_loss.pred.frame <- data.frame(t(exp_loss.predict.matrix))
colnames(exp_loss.pred.frame) <- deltas
exp_loss.pred.frame$ntree <- c(1:ntree)
exp_loss.pred.melted <- reshape2::melt(exp_loss.pred.frame,id.var='ntree')
colnames(exp_loss.pred.melted) <- c("ntree", "delta","exploss")


ggplot(exp_loss.melted, aes(x=ntree, y=exploss, col=delta)) + geom_line() + ggtitle("Exponential loss o
```

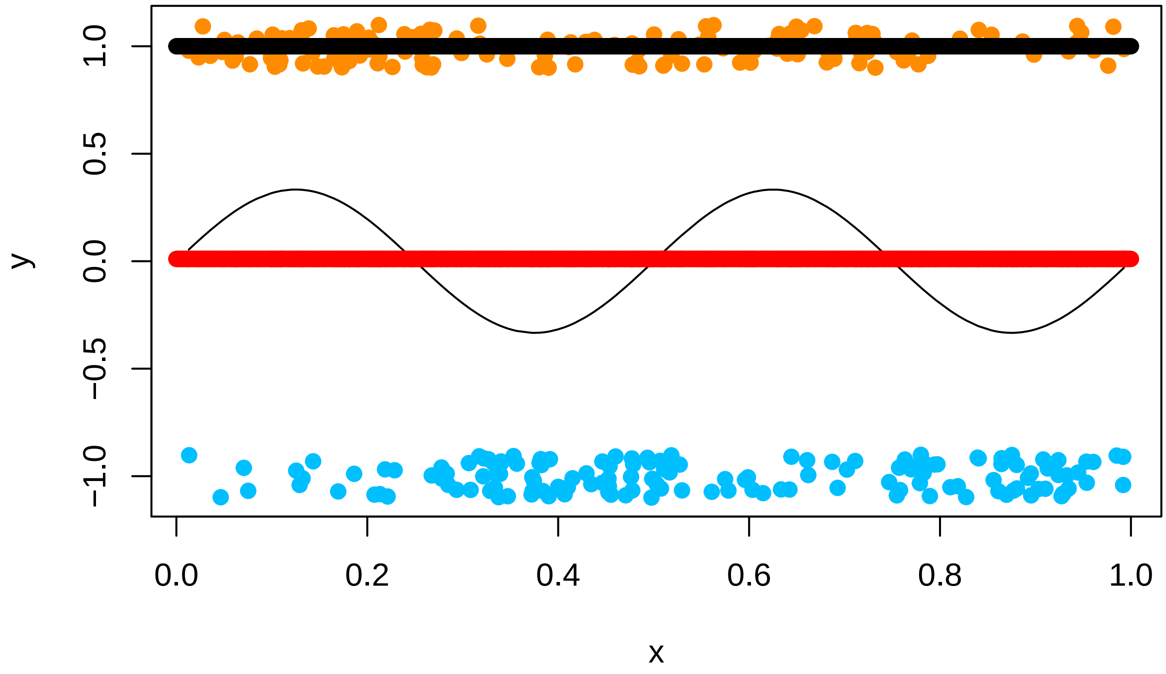


```r
ggplot(exp_loss.pred.melted, aes(x=ntree, y=exploss, col=delta)) + geom_line() + ggtitle("Exponential l
```



the best shrinkage factor is 0.01

```
data <- data.frame(x=x, y=y, weight=1/length(x))

P <- rep(0,n);
P.pred=rep(0,n_test)

data$weight <- 1/n
result <- adaboost(P, data, delta);

for (i in 1:500)
{
  result=adaboost(result$P, data, delta);

  predictions <- ifelse(testx <= result$criteria, result$left, result$right)

  P.pred <- P.pred + result$alpha*predictions
  data <- result$data

}

plot(x, y + 0.1*runif(n, -1, 1), ylim = c(-1.1, 1.1), pch = 19,
col = ifelse(y == 1, "darkorange", "deepskyblue"), ylab = "y")
lines(sort(x), py(x)[order(x)] - 0.5)

points(x, sign(result$P), col="black")
points(x, result$P, col="red")

points(testx, sign(P.pred), col="black")
points(testx, P.pred, col="red")
```


1.0
0.5
0.0
−0.5
−1.0
y
0.0
0.2
0.4
0.6
0.8
1.0
x