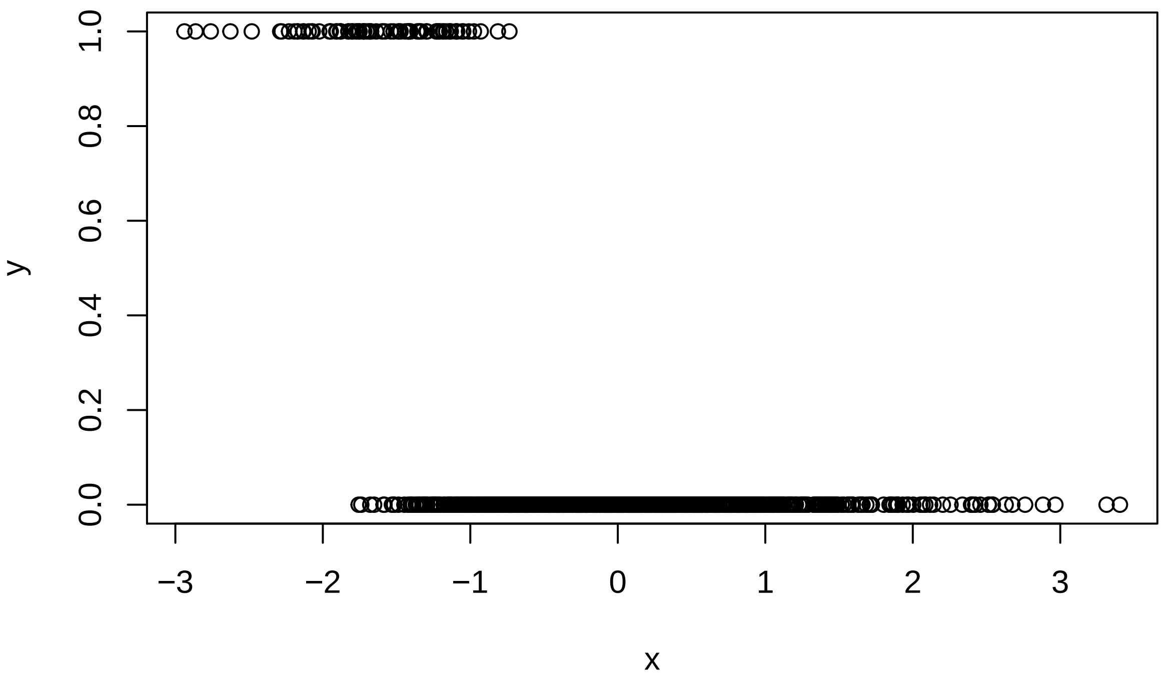
R Notebook

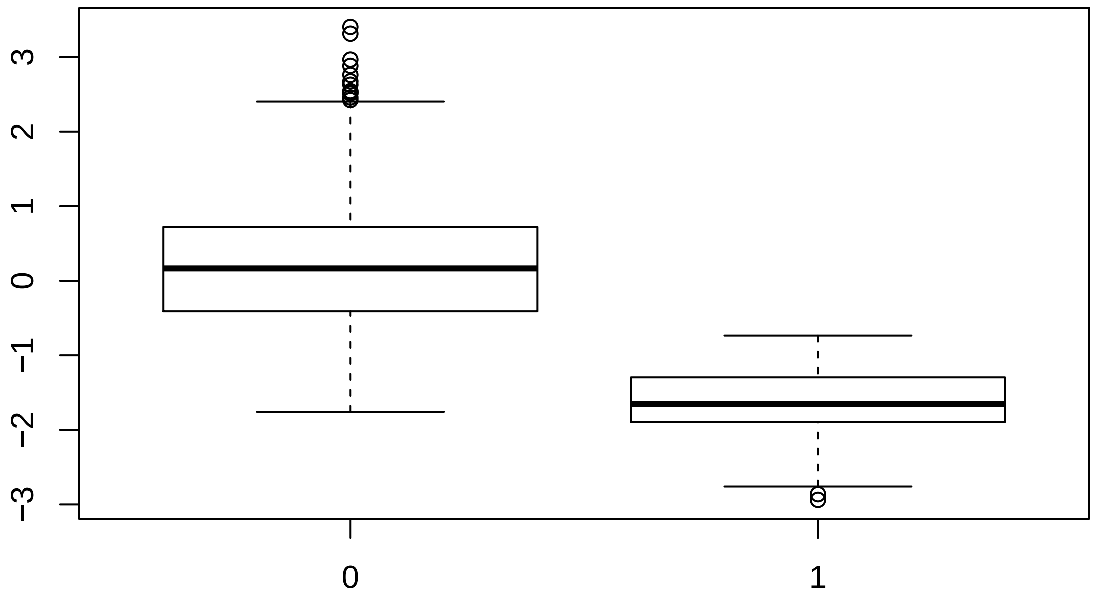
```r
options(scipen=999)
set.seed(1234)
LR2 <- read.table(file="./LR2.csv", header = TRUE, sep = ",")
names(LR2)
```

```
## [1] "y" "x"
```

```r
attach(LR2)
plot(x,y)
```




```r
boxplot(x~y)
```



Assignment 2

Exercise 1

$$\Pr(Y=1|X=x)=\Phi(\beta_0+\beta_1x)$$
$$\Phi(x)=\int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}\exp^{-\frac{1}{2}t^2}\,dt$$

```
phi_pdf <- function(x) {
  cons <- sqrt(2*pi)^-1

  # ab <- seq(-10,x,by = 0.001)
  cons*exp(-(x^2/2))
}
#PDF
phi_pdf(0)  #0.3989423
```

```
## [1] 0.3989423
```

```
dnorm(0,0,1)#0.3989423
```

```
## [1] 0.3989423
```

$$\Phi(z)=\mathrm{P}(Z\leq z),Z\sim\mathcal{N}(0,1)$$

Thus $\Phi(\beta_0+\beta_1x)=\mathrm{P}(Z\leq z)$

Write an `R` function that computes the maximum likelihood estimate, `\mathscr{L} \left( \beta_0, \beta_1 \right)`, along with bootstrapped errors.

```
# objective function
probit_mle_b <- function(x,y,...) {

  opts <- list(...)
```

```r
# probit link
#
probit <- function(b,x,y) {
  n <- length(y)
  ll <- 0
  for(i in 1:n) {
    z <- b[1]+b[2]*x[i]
    z <- pnorm(z, mean=0, sd=1, log.p = FALSE)
    ll <- ll + log(z)*(y[i]==1) + log(1-z)*(y[i]==0)
  }

  return(-ll)
}

# mle
#
obj = optim(c(0,0), probit, x=x, y=y)

coef1 <- obj$par[1]
coef2 <- obj$par[2]

return_list <- list(
  model = obj,
  fitted = pnorm(coef1+coef2*x,0,1),
  coefficients = c(coef1,coef2)
)

## Bootstrap
##

B <- 100

b_boot = matrix(rep(0,2*B),B,2)
n <- length(y) #n=1000
for (i in 1:B) {
  # indices for the i-th bootstrap subsample
  ind_ = sample(n,n,replace=TRUE)
  # input vector in the subsample
  xb = x[ind_]
  # output vector in the subsample
  yb = y[ind_]

  # compute the maximum likelihood estimates
  obj = optim(c(0,0), probit, x=xb, y=yb)

  b_boot[i,1] = obj$par[1]
  b_boot[i,2] = obj$par[2]
}

return_list$standard_errors <- c(sd(b_boot[,1]),sd(b_boot[,2]))
return_list$boots <- b_boot

if(!is.null(opts)) {
```

```r
    opts <- unlist(opts)
    return_list$response = ifelse(pnorm(coef1+coef2*opts,0,1)>1/2,1,0)
  }

  return(
    return_list
  )
}
```

```r
# Apply the probit estimator to LR2
#
est <- probit_mle_b(LR2$x,LR2$y,x)
est$coefficients
```

```
## [1] -4.242606 -3.086030
```

```r
#-4.242606 -3.086030

sum(diag(prop.table(table(est$response,y))))
```

```
## [1] 0.957
```

```r
#0.957

train <- LR2[1:800,]
test <- LR2[801:1000,]
est.2 <- probit_mle_b(train$x,train$y,test$x)
est.2$coefficients
```

```
## [1] -4.557228 -3.213275
```

```r
#-4.557228 -3.213275

sum(diag(prop.table(table(est.2$response,test$y))))
```

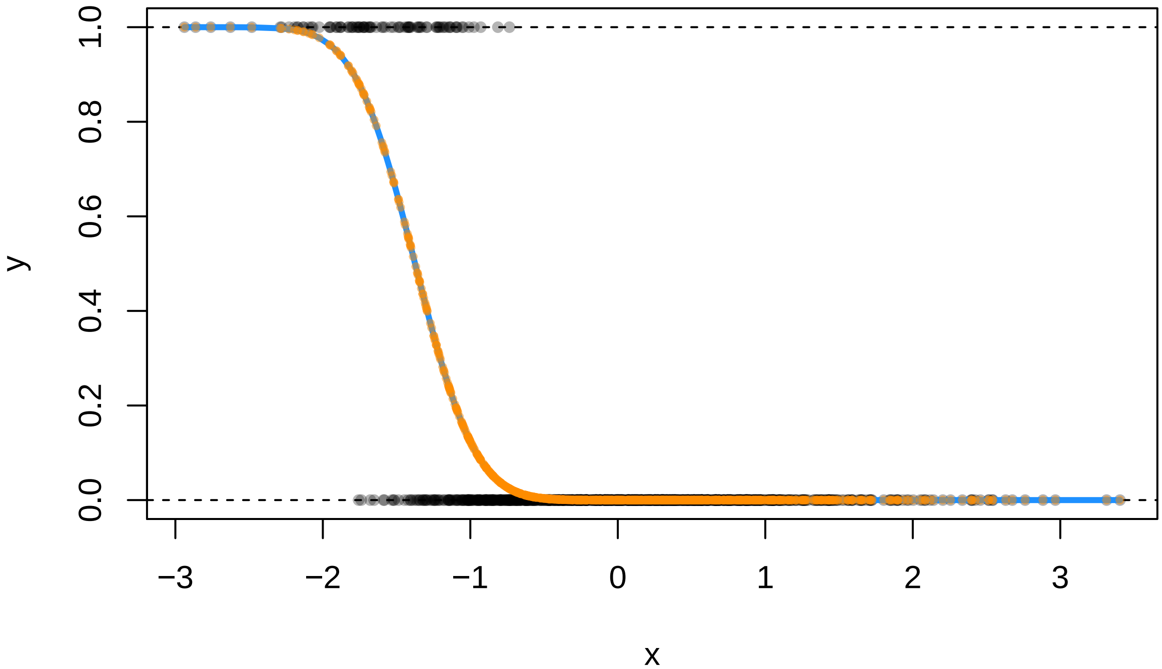
```
## [1] 0.94
```

```r
#0.94
```

```r
glm.est <- glm(y~x,family=binomial(link = "probit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
plot(x,y, pch=20, col=scales::alpha("black",alpha = 0.3))
abline(h=1, lty=2)
abline(h=0, lty=2)
y0 <- sort(predict(glm.est,list(x),type="response"), decreasing = TRUE)
y1 <- sort(est$fitted,TRUE)
lines(sort(x),y0,lwd=3,col="dodgerblue")
points(sort(x),y1,pch=20,cex=0.7,col=scales::alpha("darkorange",0.5))
```



Exercise 2

Consider the model. Let X and U be two independent uniformly distributed random variables and let Y be given by the equation

$$Y = I\left(U \leq \frac{1}{1+\exp(-\beta_0 - \beta_1 X - \beta_2 X^3 - \beta_3 \log(X))}\right)$$