Hw2

Quetsion 2 Part 2

```r
load("dataex2.Rdata")
X = dataex2$X
R = dataex2$R
N = length(X)

print(paste0("The average of X is: ", sum(X)/ N))
```

```
## [1] "The average of X is: 5.63826788567866"
```

```r
loglik = function(param){
  mu = param
  ll = 0

  for(i in 1:N)
  {

    phi = dnorm(X[i], mu, 1.5)
    Phi = pnorm(X[i], mu, 1.5)

    ll = ll + R[i]*log(phi) + (1 - R[i])*log(Phi)
  }
  ll
}

init = 1
mu = maxLik(loglik, start=init)
print(paste0("The maximum likelihood value of mu is: ", mu[2]))
```

```
## [1] "The maximum likelihood value of mu is: 5.53280299650365"
```

Question 4

```r
df = load('dataex4.Rdata')
miss = which(is.na(dataex4$Y))
data_miss = dataex4[miss,]
data_obs = dataex4[-miss,]

y_obs = data_obs$Y
x = dataex4$X
x_obs = data_obs$X
x_mis = data_miss$X

func = function(beta, xi){
  b0 = beta[1]
  b1 = beta[2]

f = exp(b0+xi*b1)/(1+exp(b0+xi*b1))
return(f)
}

Param_estimate = function(beta0, e){
  bt = beta0
  y = function(beta){
    b0 = beta[1]
    b1 = beta[2]

    b0_0 = bt[1]
    b0_1 = bt[2]

    res = sum(y_obs*(b0 + b1*x_obs)) + sum(func(bt, x_mis)*(b0 + b1*x_mis)) - sum(log(1 + exp(b0 + b1*x
    return(res)}

  bt2 = maxLik(logLik = y, start = c(beta = c(1,1)))$estimate

  while(sum(abs(bt2 - bt) > e)){

    bt = bt2
    bt2 = maxLik(logLik = y, start = c(beta = c(1,1)))$estimate
  }
  return (bt2)
}
Param_estimate(c(1,1), 1e-7)
```

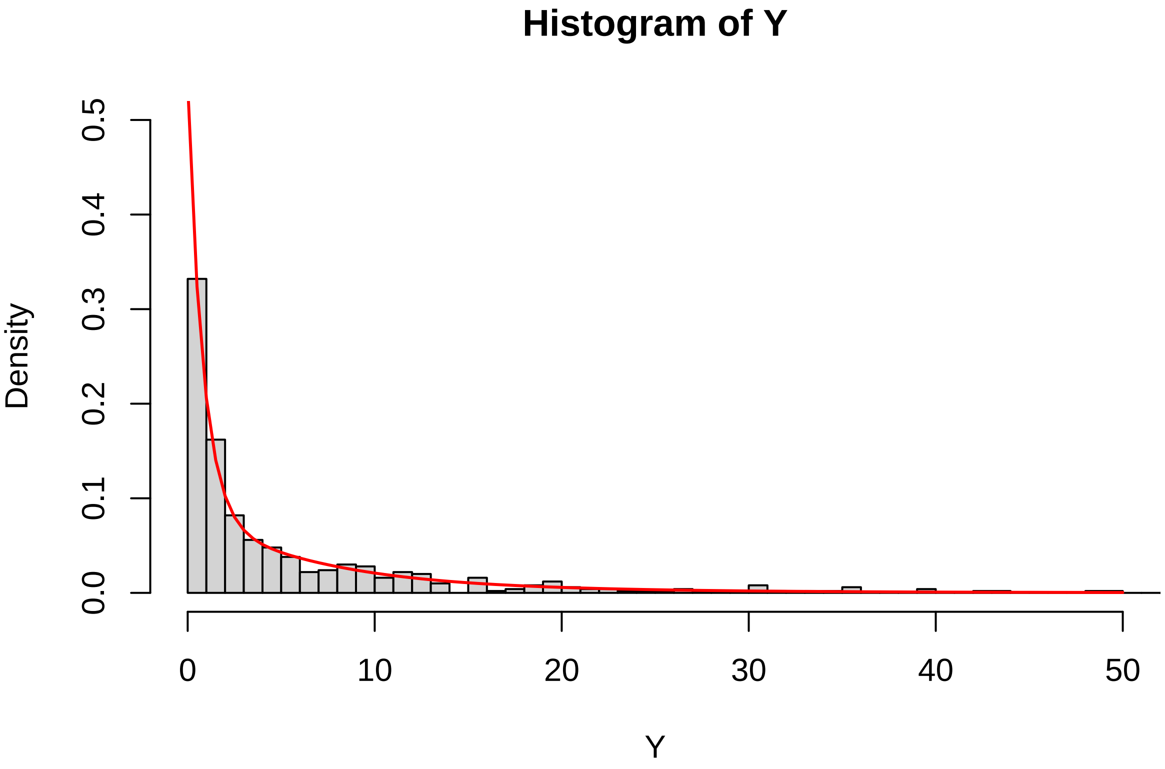
```
##      beta1      beta2
##  0.9755261 -2.4803840
```

Question 5 Part 2

```
df = load("dataex5.Rdata")
Y = dataex5
N = length(Y)
plog.value = function(params)
{
  mu = params[1]
  sigma2 = params[2]
  y = params[3]
  #print(c(sigma2))
  f.lognormal = 1/(y*sqrt(2*pi*sigma2)) * exp(-(log(y) - mu)^2/(2*sigma2))
  #print(f.lognormal)
  return(f.lognormal)
}

pexp.value = function(params)
{
  lambda = params[1]
  y = params[2]
  f.exp = lambda * exp(-lambda*y)
  #print(f.exp)
  return(f.exp)
}

M.update = function(params){
  p.t = params[1]
  mu = params[2]
  sigma = params[3]
  lambda = params[4]

  #print(params)
  sum.p.bar = 0
  for (i in 1:N) {
    p_bar = p.t*plog.value(c(mu,sigma,Y[i])) / ((1-p.t) * pexp.value(c(lambda, Y[i])) +  p.t*plog.value
    sum.p.bar = sum.p.bar + p_bar
  }
  p.t1 = sum.p.bar / N


  sum.mu = 0
  for (i in 1:N) {
    p_bar = p.t*plog.value(c(mu,sigma,Y[i])) / ((1-p.t) * pexp.value(c(lambda, Y[i])) +  p.t*plog.value
    sum.mu = sum.mu + p_bar*log(Y[i])
    }
  mu.t1 = sum.mu / sum.p.bar


  sum.sigma = 0
  for (i in 1:N) {
    p_bar = p.t*plog.value(c(mu,sigma,Y[i])) / ((1-p.t) * pexp.value(c(lambda, Y[i])) +  p.t*plog.value

    sum.sigma = sum.sigma + p_bar*(log(Y[i]) - mu.t1)*(log(Y[i]) - mu.t1)
  }

  sigma.t1 = sum.sigma / sum.p.bar

  sum.lambda = 0
  for (i in 1:N) {
    p_bar = p.t*plog.value(c(mu,sigma,Y[i])) / ((1-p.t) * pexp.value(c(lambda, Y[i])) +  p.t*plog.value
    sum.lambda = sum.lambda + (1-p_bar)*Y[i]
  }
  lambda.t1 = (N - sum.p.bar) / sum.lambda

  return(c(p.t1, mu.t1, sigma.t1, lambda.t1))
}

init = c(0.1, 1, 0.25, 3)
for(i in 1: 200)
{
  #print(init)
  res = M.update(init)
  init = res
}

p.v = res[1]
mu = res[2]
sigma = res[3]
lambda = res[4]
print(c(p.v,mu,sigma,lambda))
```

```
## [1] 0.4801995 2.0114695 0.8654688 1.0348996
```

```
t = seq(1,100,1)
curve.val = function(x){
  f = p.v*plog.value(c(mu,sigma,x)) + (1-p.v)*pexp.value(c(lambda,x))
  return(f)
}


hist(Y,ylim = c(0,0.5), xlim = c(0,50),breaks = 100, probability = TRUE)
curve((p.v*dlnorm(x, mu, sigma) +
(1 - p.v)*dexp(x, lambda)), add = TRUE, lwd = 1.5, col = "red")
```


Histogram of Y
Density
Y