MA677_Final

Carolyn Wright

Assignment

Continue reading Introduction to Empirical Bayes, chapters 5, 6, 11, 12, and 13. Now, read Chapter 6 in Computer Age Statistical Inference which takes you to the next step in using empirical Bayes. Chapter 6 contains four examples – insurance claims, species discovery, Shakespeare's vocabulary, and lymph node counts. In each example, the result of the empirical Bayes analysis is given. Using R, reproduce the analyses in Chapter 6 describing how you are proceeding and providing commentary. Note that at the end of the chapter, the authors noted that empirical Bayes is often used to support analyses of false-discovery rates which is discussed in Chapter 15.

Insurance Claims Example

```
x <- 0:7
y <- c(7840,1317,239, 42,14,4,4,1)
F66 <- y/9461

#Use Robbins rule to calculate the distribution
Insurance_Claims <- data.frame(cbind(x,y,F66))
Insurance_Claims$Robbins <- round((x+1)*lead(F66)/F66,3)

#Estimate mean and variance using MLE
  X <- data.frame(c(rep(0,7840), rep(1,1317), rep(2,239), rep(3,42),
                    rep(4, 14), rep(5,4), rep(6, 4), rep(7,1)))
  names(X) <- "X"


  #Attempt 1;
  gmll <- function(theta,datta)
    {
      a <- theta[1];
      b <- theta[2]
      n <- length(datta);
      sumd <- sum(datta);
      sumlogd <- sum(log(datta))
     gmll <- n*a*log(b) + n*lgamma(a) + sumd/b - (a-1)*sumlogd
     gmll
       } # End function gmll

  momalpha <- mean(X$X)^2/var(X$X);

  mombeta <- var(X$X)/mean(X$X);

  gammasearch = nlm(gmll,c(momalpha,mombeta),hessian=T,datta=X$X);
```

```
## Warning in nlm(gmll, c(momalpha, mombeta), hessian = T, datta = X$X): NA/Inf
## replaced by maximum positive value

## Warning in nlm(gmll, c(momalpha, mombeta), hessian = T, datta = X$X): NA/Inf
## replaced by maximum positive value

## Warning in nlm(gmll, c(momalpha, mombeta), hessian = T, datta = X$X): NA/Inf
## replaced by maximum positive value

## Warning in nlm(gmll, c(momalpha, mombeta), hessian = T, datta = X$X): NA/Inf
## replaced by maximum positive value

## Warning in nlm(gmll, c(momalpha, mombeta), hessian = T, datta = X$X): NA/Inf
## replaced by maximum positive value

## Warning in nlm(gmll, c(momalpha, mombeta), hessian = T, datta = X$X): NA/Inf
## replaced by maximum positive value

## Warning in nlm(gmll, c(momalpha, mombeta), hessian = T, datta = X$X): NA/Inf
## replaced by maximum positive value

## Warning in nlm(gmll, c(momalpha, mombeta), hessian = T, datta = X$X): NA/Inf
## replaced by maximum positive value
```

```
#Attempt 2
NLL = function(pars, data) {
  # Extract parameters from the vector
  mu = pars[1]
  sigma = pars[2]
  # Calculate Negative Log-LIkelihood
  -sum(dnorm(x = data, mean = mu, sd = sigma, log = TRUE))
}

mle = optim(par = c(mu = 0.5, sigma = 1), fn = NLL, data = X$X,
            control = list(parscale = c(mu = 0.5, sigma = 1)))


#Plug in estimates in order to calculate the gamma distribution
sigma = 0.3055570451142745
nu = 0.701509650727234
lambda = sigma/(1+sigma)
Gamma_calc <- {}

for (i in  0:7){
  j = i+1

  f_x0<- ((lambda^(nu+i))*gamma(nu+i))/((sigma^nu)*gamma(nu)*factorial(i))
  f_x1 <- ((lambda^(nu+j))*gamma(nu+j))/((sigma^nu)*gamma(nu)*factorial(j))

  Gamma_calc <- rbind(Gamma_calc,j*f_x1/f_x0)

}

Insurance_Claims <- data.frame(cbind(Insurance_Claims,Gamma_calc))
```

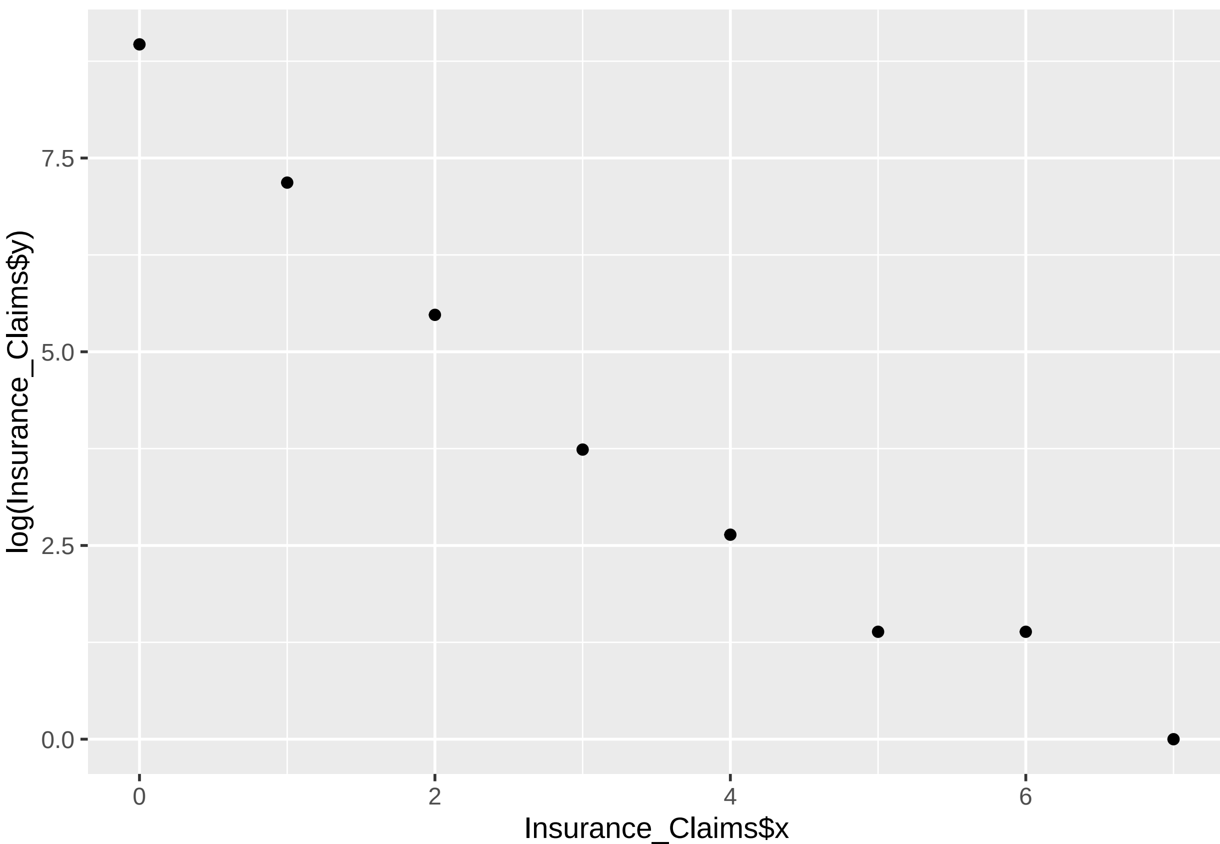
Table 6.1

```
Insurance_Claims[c("x","y","Robbins","Gamma_calc")]
```

```
##   x    y Robbins Gamma_calc
## 1 0 7840   0.168  0.1641837
## 2 1 1317   0.363  0.3982271
## 3 2  239   0.527  0.6322706
## 4 3   42   1.333  0.8663140
## 5 4   14   1.429  1.1003574
## 6 5    4   6.000  1.3344009
## 7 6    4   1.750  1.5684443
## 8 7    1      NA  1.8024877
```

Figure 6.1

```
ggplot() + geom_point(aes(x = Insurance_Claims$x, y = log(Insurance_Claims$y)))
```



Species Example

```
x <- 1:24
y <- c(118,74,44,24,29,22,20,19,20,15,12,14,6,12,6,9,9,6,10,10,11,5,3,3)

Species <- data.frame(cbind(x,y))

#Formula 6.19
```

```
f619 <- function(time){
  z = {}
  for (i in 1:24){
  z[i] <- ((-1)^(x[i]-1)) * y[i] * time**x[i]
  }

  sum(z)
}

E_t <- cbind(f619(0),f619(.1),f619(.2),f619(.3),f619(.4),f619(.5),f619(.6),
             f619(.7),f619(.8),f619(.9),f619(1))


#Formula 6.21 -- this does not match what is in the book unless the 2*x[i] is
#changed to just 2
f621 <- function(time){
  v = {}
  for (i in 1:24){
    v[i] <- y[i]*time^(2)
  }

  sqrt(sum(v))
}

sd_t <- cbind(f621(0),f621(.1),f621(.2),f621(.3),f621(.4),f621(.5),f621(.6),
              f621(.7),f621(.8),f621(.9),f621(1))

lb <- E_t - sd_t
ub <- E_t + sd_t
```

Table 6.3

```
print("Table 6.3:")
```

```
## [1] "Table 6.3:"
```

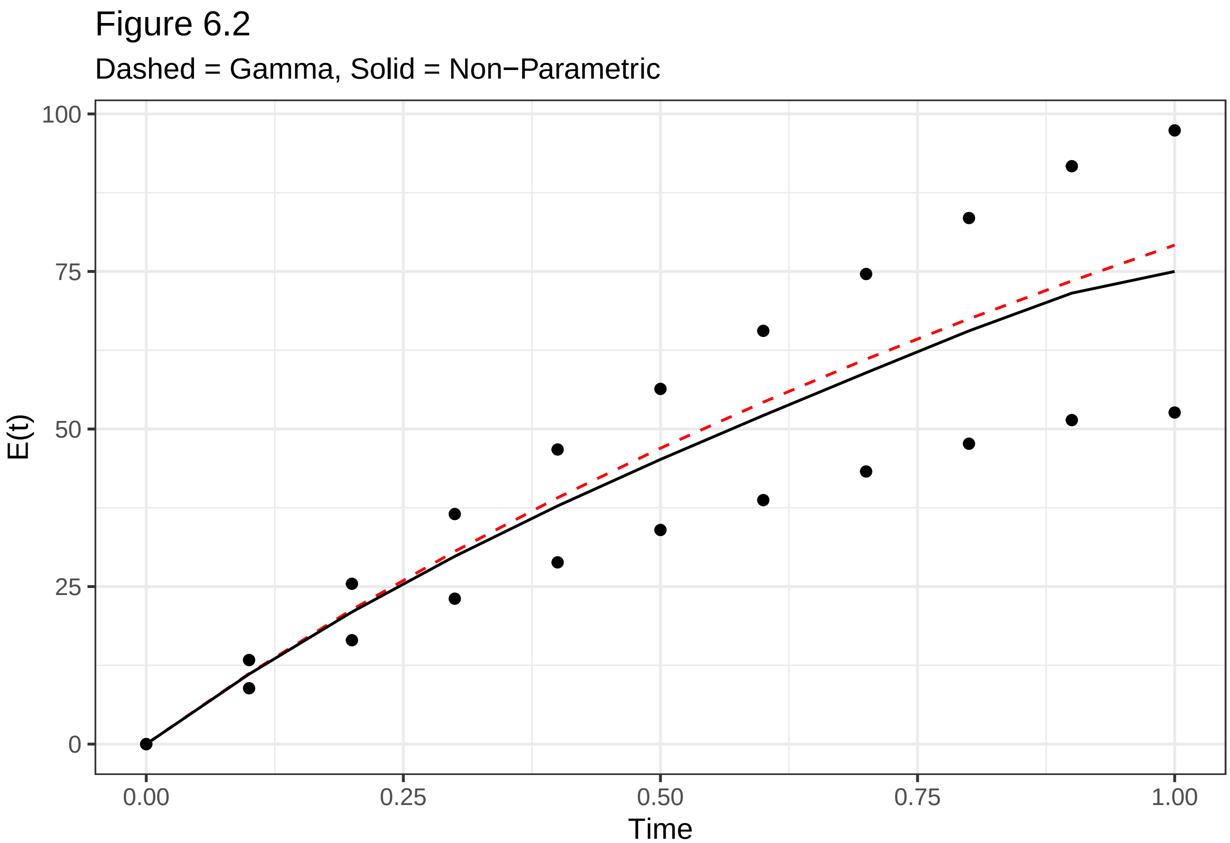
```
(Table_63 <- rbind(E_t, sd_t))
```

```
##      [,1]      [,2]      [,3]      [,4]      [,5]     [,6]     [,7]     [,8]
## [1,]    0 11.101870 20.961688 29.791479 37.792715 45.17149 52.14693 58.92833
## [2,]    0  2.238303  4.476606  6.714909  8.953212 11.19151 13.42982 15.66812
##          [,9]    [,10]    [,11]
## [1,] 65.57362 71.55992 75.00000
## [2,] 17.90642 20.14473 22.38303
```

Figure 6.2

```
#Figure 6.2

nu = 0.104
sigma = 89.79
gamma = sigma / (1 + sigma)

#Formula 6.23
```

```
f623 <- function(time){
  t = {}
  y[1]* (1 - (1 + gamma * time)**(-nu)) / (gamma * nu)
}

t <- c(0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1)
E_t_1 <- cbind(f623(0),f623(.1),f623(.2),f623(.3),f623(.4),f623(.5),f623(.6),
               f623(.7),f623(.8),f623(.9),f623(1))

ggplot() + geom_line(aes(x = t, y = E_t_1), linetype ='dashed', color = "red") +
  ylab("E(t)") +
  geom_line(aes(x = t, y = E_t) ) + ylab("E(t)")+ xlab("Time") +
  geom_point(aes(x=t, y=ub)) +
  geom_point(aes(x=t, y=lb)) +
  ggtitle("Figure 6.2", subtitle = "Dashed = Gamma, Solid = Non-Parametric") +
  theme_bw()
```



Shakespeare Example

```
x<- 1:100
y <- c(14376, 4343,2292,1463,1043,837,638,519,430,364,305,259,242,223,187,
181,179,130,127,128,104,105,99,112,93,74,83,76,72,63,73,47,56,59,53,45,
34,49,45,52,49,41,30,35,37,21,41,30,28,19,25,19,28,27,31,19,19,22,23,14,30,
19,21,18,15,10,15,14,11,16,13,12,10,16,18,11,8,15,12,7,13,12,11,8,10,11,7,
12, 9,8,4,7,6,7,10,10,15,7,7,5)
```

```r
f619 <- function(time){
  z = {}
  for (i in 1:100){
    z[i] <- ((-1)^(x[i]-1)) * y[i] * time**x[i]
  }

  sum(z)
}

f621 <- function(time){
  v = {}
  for (i in 1:100){
    v[i] <- y[i]*time^(2)
  }

  sqrt(sum(v))
}
```

Value 6.25

```r
#6.25
paste0("Table 6.25: ",f619(1)," +/- ",round(f621(1),2))
```

```
## [1] "Table 6.25: 11486 +/- 175.18"
```

Value 6.32

```r
#6.32
paste0("Table 6.32: ",round(f619(429/884647),2))
```

```
## [1] "Table 6.32: 6.97"
```

Medical Example

Figure 6.3

```r
nodes <- read.table("nodes.txt", header = TRUE)

nodes$prob <- as.numeric(nodes$x)/as.numeric(nodes$n)

ggplot() + geom_histogram(aes(x = nodes$prob), fill = "green", color = "black") +
  ylab("Frequency") + xlab("p=x/n") + ggtitle("Figure 6.3")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

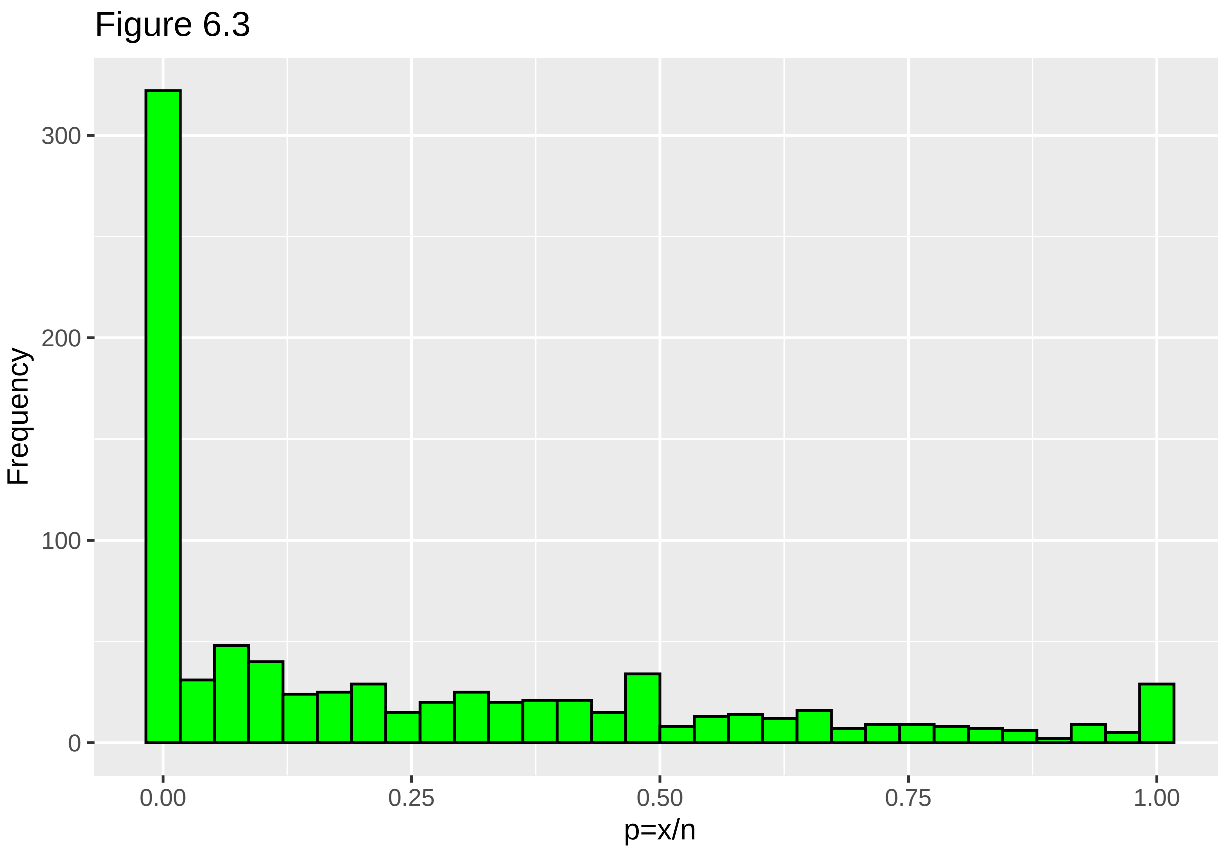


Figure 6.4

```
theta <- seq(from = 0.01, to = 0.99, by = 0.01)


output <- deconv(tau = theta, X = nodes, family = "Binomial")
nodes2 <- data.frame(output$stats)
indices <- seq(5, 99, 3)
error <- theta[indices]

ggplot() +
  geom_line(data = nodes2, aes(x = theta, y = g)) +
  geom_errorbar(data = nodes2[indices, ],aes(x = theta, ymin = g - SE.g,
                                             ymax = g + SE.g), width = .01,
                color = "red")
```

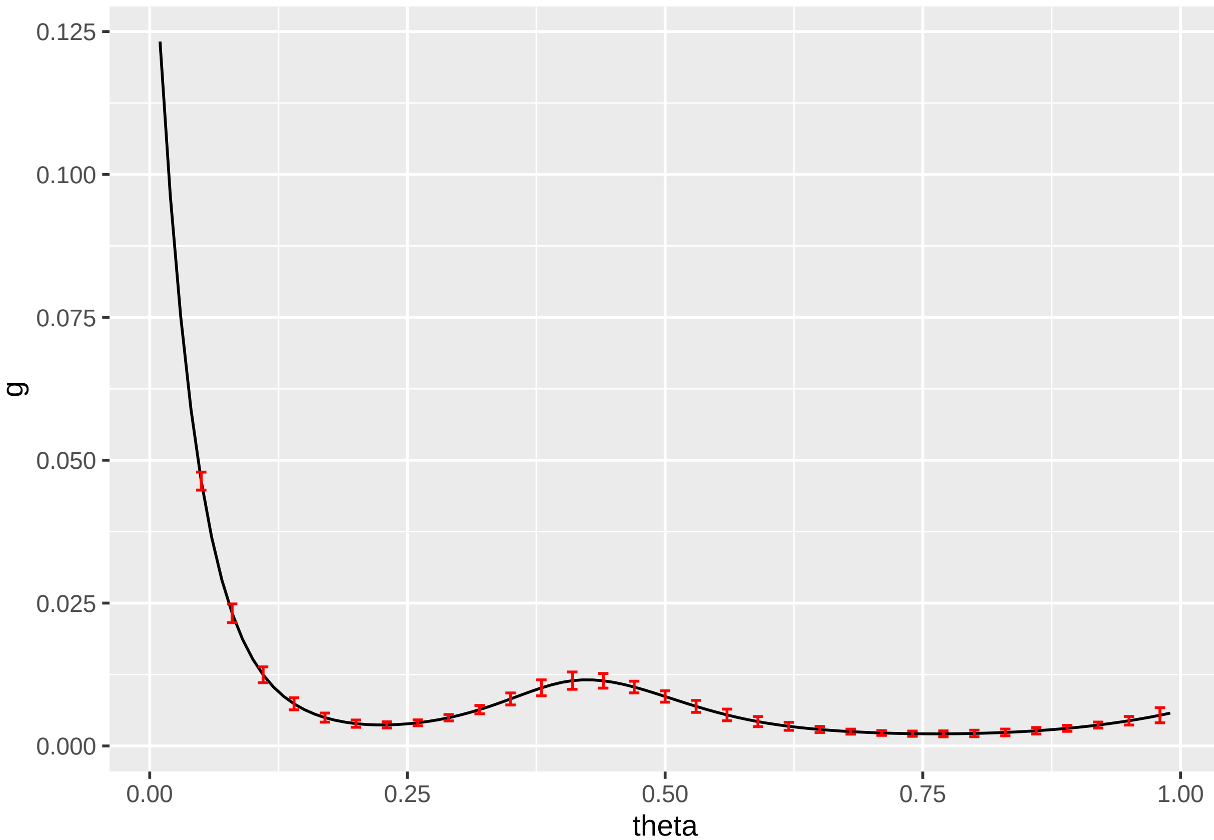


Figure 6.5

```
theta <- output$stats[, 'theta']
gTheta<- output$stats[, 'g']

denom <- function(n_k, x_k) {
    sum(dbinom(x = x_k, size = n_k, prob = theta) * gTheta) * .01
}

#Formula 6.43
f643 <- function(n_k, x_k) {
    gTheta * dbinom(x = x_k, size = n_k, prob = theta) / denom(n_k, x_k)
}

g1 <- f643(x_k =7, n_k=  32)
g2 <- f643(x_k =3, n_k=   6)
g3 <- f643(x_k =17,n_k=   18)
ggplot() + geom_line(mapping = aes(x = theta, y = g1), color = "blue",linetype = "dashed") +
  ylim(0,10)+
    geom_line(mapping =aes(x = theta, y = g2), color = "red") +
    geom_line(mapping =aes(x = theta, y = g3), color = "black", linetype ="dotted") +
  ggtitle("Figure 6.5")
```

Figure 6.5