

University of Edinburgh, School of Mathematics

Incomplete Data Analysis, 2020/2021

Workshop 4 – Solutions

Vanda Inácio

1. In this case we have $y_{\text{mis}} = \{y_1\}$ and $y_{\text{obs}} = \{y_2, y_3, y_4, y_5, y_6, y_7\}$. Let $\theta = (\lambda, \beta)$. The complete data likelihood is

$$\begin{aligned} L(\theta; y_{\text{obs}}, y_{\text{mis}}) &= \prod_{i=1}^5 \{\lambda e^{-\lambda y_i}\} \prod_{i=6}^7 \{\beta \lambda e^{-\beta \lambda y_i}\} \\ &= \lambda^7 \beta^2 e^{-\lambda(\sum_{i=1}^5 y_i + \beta \sum_{i=6}^7 y_i)}, \end{aligned}$$

and therefore the corresponding log likelihood is given by

$$\log L(\theta; y_{\text{obs}}, y_{\text{mis}}) = 7 \log \lambda + 2 \log \beta - \lambda \left(\sum_{i=1}^5 y_i + \beta \sum_{i=6}^7 y_i \right).$$

We proceed to the E step.

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= E_{Y_1} \left[\log L(\theta; y_{\text{obs}}, y_{\text{mis}}) \mid y_{\text{obs}}, \theta^{(t)} \right] \\ &= 7 \log \lambda + 2 \log \beta - \lambda \left(E \left[Y_1 \mid y_{\text{obs}}, \theta^{(t)} \right] + \sum_{i=2}^5 y_i + \beta \sum_{i=6}^7 y_i \right). \end{aligned}$$

We have to calculate $E(Y_1 \mid y_{\text{obs}}, \theta^{(t)})$. We know that if $Y \sim \text{Exp}(\lambda)$, then $E(Y) = \frac{1}{\lambda}$. Hence $E(Y_1 \mid y_{\text{obs}}, \theta^{(t)}) = E(Y_1 \mid \lambda^{(t)}) = \frac{1}{\lambda^{(t)}}$. We thus have

$$Q(\theta \mid \theta^{(t)}) = 7 \log \lambda + 2 \log \beta - \lambda \left(\frac{1}{\lambda^{(t)}} + \sum_{i=2}^5 y_i + \beta \sum_{i=6}^7 y_i \right).$$

We can now go to the M-step,

$$\frac{\partial}{\partial \lambda} Q(\theta \mid \theta^{(t)}) = \frac{7}{\lambda} - \left(\frac{1}{\lambda^{(t)}} + \sum_{i=2}^5 y_i + \beta \sum_{i=6}^7 y_i \right).$$

Then,

$$\frac{\partial}{\partial \lambda} Q(\theta \mid \theta^{(t)}) = 0 \Rightarrow \lambda^{(t+1)} = \frac{7\lambda^{(t)}}{1 + (\sum_{i=2}^5 y_i + \beta^{(t)} \sum_{i=6}^7 y_i)\lambda^{(t)}}.$$

Similarly,

$$\frac{\partial}{\partial \beta} Q(\theta \mid \theta^{(t)}) = \frac{2}{\beta} - \lambda(y_6 + y_7),$$

and

$$\frac{\partial}{\partial \beta} Q(\theta \mid \theta^{(t)}) = 0 \Rightarrow \beta^{(t+1)} = \frac{2}{\lambda^{(t+1)}(y_6 + y_7)}.$$

Since we know that $y_2 = 5$, $y_3 = 2$, $y_4 = 2.5$, $y_5 = 4$, $y_6 = 0.25$, and $y_7 = 0.75$, these two expressions simplify to

$$\lambda^{(t+1)} = \frac{7\lambda^{(t)}}{1 + (13.5 + \beta^{(t)})\lambda^{(t)}}, \quad \beta^{(t+1)} = \frac{2}{\lambda^{(t+1)}}.$$

We can code this procedure easily in R. Note that here the E step and M step do not need to be re-derived at each iteration: we just need to perform an iterative application of the updating formulas starting from some initial values.

```
expex <- function(theta0, eps){
  diff <- 1
  theta <- theta0
  lambda <- theta[1]
  beta <- theta[2]

  while(diff > eps){
    theta.old <- theta

    lambda <- 7*lambda/(1 + (13.5 + beta)*lambda)
    beta <- 2/lambda

    theta <- c(lambda, beta)
    diff <- sum(abs(theta-theta.old))
  }
  return(theta)
}
```

```
expex(theta0 = c(1, 1), 0.00001)
```

```
## [1] 0.2962966 6.7499930
```

We have obtained that $\hat{\lambda} = 0.296$ and $\hat{\beta} = 6.750$. We should try different initial values and check whether we are obtaining the same results (although it still does not 100% rule out the possibility of reaching a local (rather than the global) maximum).

```
expex(theta0 = c(10, 10), 0.00001)
```

```
## [1] 0.2962964 6.7499965
```

```
expex(theta0 = c(20, 50), 0.00001)
```

```
## [1] 0.296296 6.750006
```

```
expex(theta0 = c(50, 20), 0.00001)
```

```
## [1] 0.2962961 6.7500037
```

```
expex(theta0 = c(100, 100), 0.00001)
```

```
## [1] 0.296296 6.750006
```

Aside: As an (simpler!) alternative to the EM, we could maximize directly the log likelihood of the observed data (both approaches need ignorability to provide valid estimates). The log likelihood of the observed data is given by

$$\log L(\theta; y_{\text{obs}}) = 6 \log \lambda + 2 \log \beta - \lambda \left(\sum_{i=2}^5 y_i + \beta \sum_{i=6}^7 y_i \right).$$

```
log_likelihood <- function(param){
  lambda <- param[1]
  beta <- param[2]
  6*log(lambda) + 2*log(beta) - lambda*(13.5 + beta)
}
res <- optim(c(2,2), log_likelihood, control = list(fnscale = -1))
```

```
## Warning in log(lambda): NaNs produced
```

```
## Warning in log(lambda): NaNs produced
```

```
## Warning in log(lambda): NaNs produced
```

```
res$par
```

```
## [1] 0.2962551 6.7517747
```

Although we have obtained estimates that are pretty much similar to those obtained with the EM algorithm, there were some warnings. These are due to the fact that both λ and β need to be greater than zero and so basically we have a constrained optimization problem. ‘Exponentiating’ both parameters is an easy way to overcome such problem.

```
log_likelihood_exp <- function(param){
  lambda <- exp(param[1])
  beta <- exp(param[2])
  6*log(lambda) + 2*log(beta) - lambda*(13.5 + beta)
}

res <- optim(c(2,2), log_likelihood_exp, control = list(fnscale = -1))
exp(res$par)
```

```
## [1] 0.2960743 6.7561905
```

No more warnings!!

2. (a) Let Y_i be the number of deaths on the i th day, $i = 1, \dots, 1096$, where 1096 is the sum of the observed frequencies (total number of days). We are assuming $Y_i \stackrel{\text{iid}}{\sim} \text{Poisson}(\lambda)$, i.e.,

$$\Pr(Y = y) = e^{-\lambda} \frac{\lambda^y}{y!}, \quad y = 0, 1, \dots, 9.$$

The likelihood is

$$L(\lambda) = \prod_{i=1}^n \left\{ e^{-\lambda} \frac{\lambda^{y_i}}{y_i!} \right\} = e^{-n\lambda} \frac{\lambda^{\sum_{i=1}^n y_i}}{\prod_{i=1}^n y_i!},$$

and the corresponding log likelihood is

$$\log L(\lambda) = -n\lambda + \log \lambda \sum_{i=1}^n y_i - \sum_{i=1}^n \log y_i!.$$

We have

$$\frac{d}{d\lambda} \log L(\lambda) = -n + \frac{\sum_{i=1}^n y_i}{\lambda},$$

which implies that

$$\hat{\lambda} = \frac{1}{n} \sum_{i=1}^n Y_i.$$

Based on the data, we obtain $\hat{\lambda} = 2.157$. The plot of the observed counts against the expected counts under a Poisson model with parameter 2.157 is presented below.

```

y <- rep(0:9, times = c(162, 267, 271, 185, 111, 61, 27, 8, 3, 1))
yagg <- table(y)
n <- length(y)

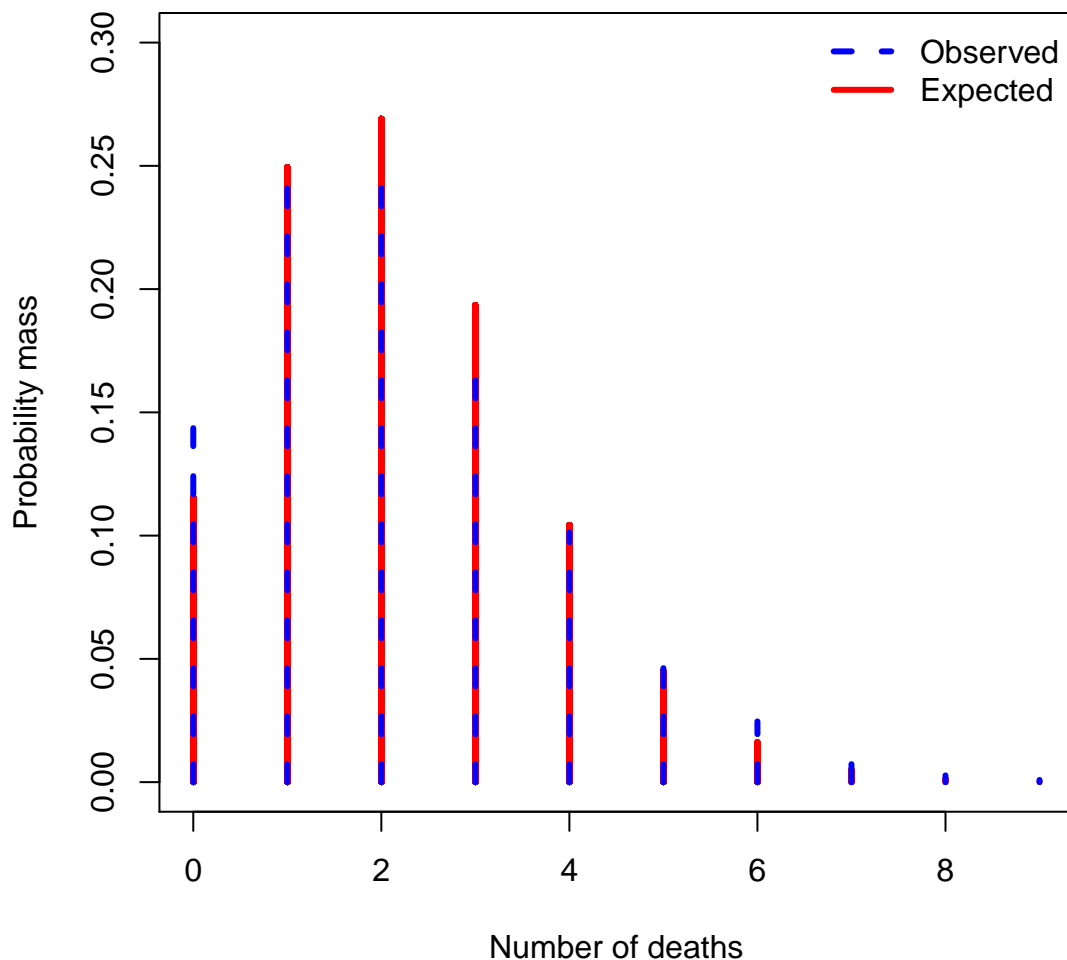
#mle estimate
lambdamle <- sum(y)/n
lambdamle

## [1] 2.156934

fit <- dpois(y, lambda = lambdamle)

plot(y, fit, type = "h", lwd = 3, lty = 1, col = "red", ylim = c(0, 0.3),
      xlab = "Number of deaths", ylab = "Probability mass")
points(0:9, yagg/n, lty = 2, lwd = 3, type = "h", col = "blue")
legend("topright", legend = c("Observed", "Expected"),
      lty = c(2, 1), lwd = c(3, 3), col = c("blue", "red"), bty = "n")

```



The fit although not terrible, does not seem very good either. In fact, and out of curiosity, if we conduct a chi-square test, we reject the null hypothesis that the data follow a Poisson distribution with parameter $\hat{\lambda} = 2.157$. For simplicity, I will use the function `chisq.test` and will set the option of computing the p-value by simulation to `TRUE`.

```

probs <- dpois(0:9, lambda = lambdamle)
comp <- 1 - sum(probs)
chisq.test(x = c(162, 267, 271, 185, 111, 61, 27, 8, 3, 1, 0),
           p = c(probs, comp), simulate.p.value = TRUE, B = 5000)

```

```

##
## Chi-squared test for given probabilities with simulated p-value (based
## on 5000 replicates)
##
## data: c(162, 267, 271, 185, 111, 61, 27, 8, 3, 1, 0)
## X-squared = 27.069, df = NA, p-value = 0.014

```

2. (b) We now consider a mixture of two Poisson distributions

$$\Pr(Y = y) = pe^{-\lambda_1} \frac{\lambda_1^y}{y!} + (1-p)e^{-\lambda_2} \frac{\lambda_2^y}{y!}, \quad 0 \leq p \leq 1, \lambda_1 > 0, \lambda_2 > 0, \quad \theta = (p, \lambda_1, \lambda_2).$$

Let z_i be binary latent variables indicating component membership, i.e.,

$$z_i = \begin{cases} 1 & \text{if } y_i \text{ belongs to the first component,} \\ 0 & \text{if } y_i \text{ belongs to the second component.} \end{cases}$$

The observed data in this context is $\mathbf{y} = (y_1, \dots, y_n)$ and the missing data is $\mathbf{z} = (z_1, \dots, z_n)$. The likelihood of the complete data (\mathbf{y}, \mathbf{z}) is

$$L(\theta; \mathbf{y}, \mathbf{z}) = \prod_{i=1}^n \left\{ \left[pe^{-\lambda_1} \frac{\lambda_1^{y_i}}{y_i!} \right]^{z_i} \left[(1-p)e^{-\lambda_2} \frac{\lambda_2^{y_i}}{y_i!} \right]^{1-z_i} \right\},$$

with the corresponding log likelihood being

$$\log L(\theta; \mathbf{y}, \mathbf{z}) = \sum_{i=1}^n z_i \{\log p - \lambda_1 + y_i \log \lambda_1 - \log(y_i!)\} + \sum_{i=1}^n (1-z_i) \{\log(1-p) - \lambda_2 + y_i \log \lambda_2 - \log(y_i!)\}.$$

For the E-step we need to calculate

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= E_Z \left[\log L(\theta; \mathbf{y}, \mathbf{z}) \mid \mathbf{y}, \theta^{(t)} \right] \\ &= \sum_{i=1}^n E[Z_i \mid y_i, \theta^{(t)}] \{\log p - \lambda_1 + y_i \log \lambda_1 - \log(y_i!)\} \\ &\quad + \sum_{i=1}^n (1 - E[Z_i \mid y_i, \theta^{(t)}]) \{\log(1-p) - \lambda_2 + y_i \log \lambda_2 - \log(y_i!)\}. \end{aligned}$$

We know that $E[Z_i \mid y_i, \theta^{(t)}] = \Pr(Z_i = 1 \mid y_i, \theta^{(t)})$, and applying Bayes theorem and the law of total probability, we obtain

$$E[Z_i \mid y_i, \theta^{(t)}] = \Pr(Z_i = 1 \mid y_i, \theta^{(t)}) = \frac{p^{(t)} e^{-\lambda_1^{(t)}} \frac{(\lambda_1^{(t)})^{y_i}}{y_i!}}{p^{(t)} e^{-\lambda_1^{(t)}} \frac{(\lambda_1^{(t)})^{y_i}}{y_i!} + (1-p^{(t)}) e^{-\lambda_2^{(t)}} \frac{(\lambda_2^{(t)})^{y_i}}{y_i!}} = \tilde{p}_i^{(t)}, \quad i = 1, \dots, n.$$

Therefore,

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= \sum_{i=1}^n \tilde{p}_i^{(t)} \{\log p - \lambda_1 + y_i \log \lambda_1 - \log(y_i!)\} \\ &\quad + \sum_{i=1}^n (1 - \tilde{p}_i^{(t)}) \{\log(1-p) - \lambda_2 + y_i \log \lambda_2 - \log(y_i!)\}. \end{aligned}$$

For the M-step, we only need to compute the partial derivatives

$$\begin{aligned}\frac{\partial}{\partial p} Q(\theta \mid \theta^{(t)}) = 0 &\Rightarrow p^{(t+1)} = \frac{1}{n} \sum_{i=1}^n \tilde{p}_i^{(t)}, \\ \frac{\partial}{\partial \lambda_1} Q(\theta \mid \theta^{(t)}) = 0 &\Rightarrow \lambda_1^{(t+1)} = \frac{\sum_{i=1}^n \tilde{p}_i^{(t)} y_i}{\sum_{i=1}^n \tilde{p}_i^{(t)}}, \\ \frac{\partial}{\partial \lambda_2} Q(\theta \mid \theta^{(t)}) = 0 &\Rightarrow \lambda_2^{(t+1)} = \frac{\sum_{i=1}^n (1 - \tilde{p}_i^{(t)}) y_i}{\sum_{i=1}^n (1 - \tilde{p}_i^{(t)})}.\end{aligned}$$

- (c) Using the R code provided below, we obtain $\hat{p} = 0.361$, $\hat{\lambda}_1 = 1.257$, and $\hat{\lambda}_2 = 2.664$. The plot of the observed counts against the expected counts under this mixture model is shown below. Compared with the fitted distribution in (a), it seems the mixture model describes the data considerably better. Note that in the code below we have used for the stopping criterion

$$|p^{(t+1)} - p^{(t)}| + |\lambda_1^{(t+1)} - \lambda_1^{(t)}| + |\lambda_2^{(t+1)} - \lambda_2^{(t)}| < \varepsilon,$$

with $\varepsilon = 0.00001$. For the starting values we have considered $p^{(0)} = 0.8$, $\lambda_1^{(0)} = 0.5$, $\lambda_2^{(0)} = 5$.

```
em.mixture.two.poisson <- function(y, theta0, eps){
  n <- length(y)
  theta <- theta0
  p <- theta[1]; lambda1 <- theta[2]; lambda2 <- theta[3]
  diff <- 1
  while(diff > eps){
    theta.old <- theta

    #E-step
    ptilde1 <- p*dpois(y, lambda = lambda1)
    ptilde2 <- (1-p)*dpois(y, lambda = lambda2)
    ptilde <- ptilde1/(ptilde1 + ptilde2)

    #M-step
    p <- mean(ptilde)
    lambda1 <- sum(y*ptilde)/sum(ptilde)
    lambda2 <- sum(y*(1-ptilde))/sum(1-ptilde)
    theta <- c(p, lambda1, lambda2)
    diff <- sum(abs(theta-theta.old))
  }
  return(theta)
}

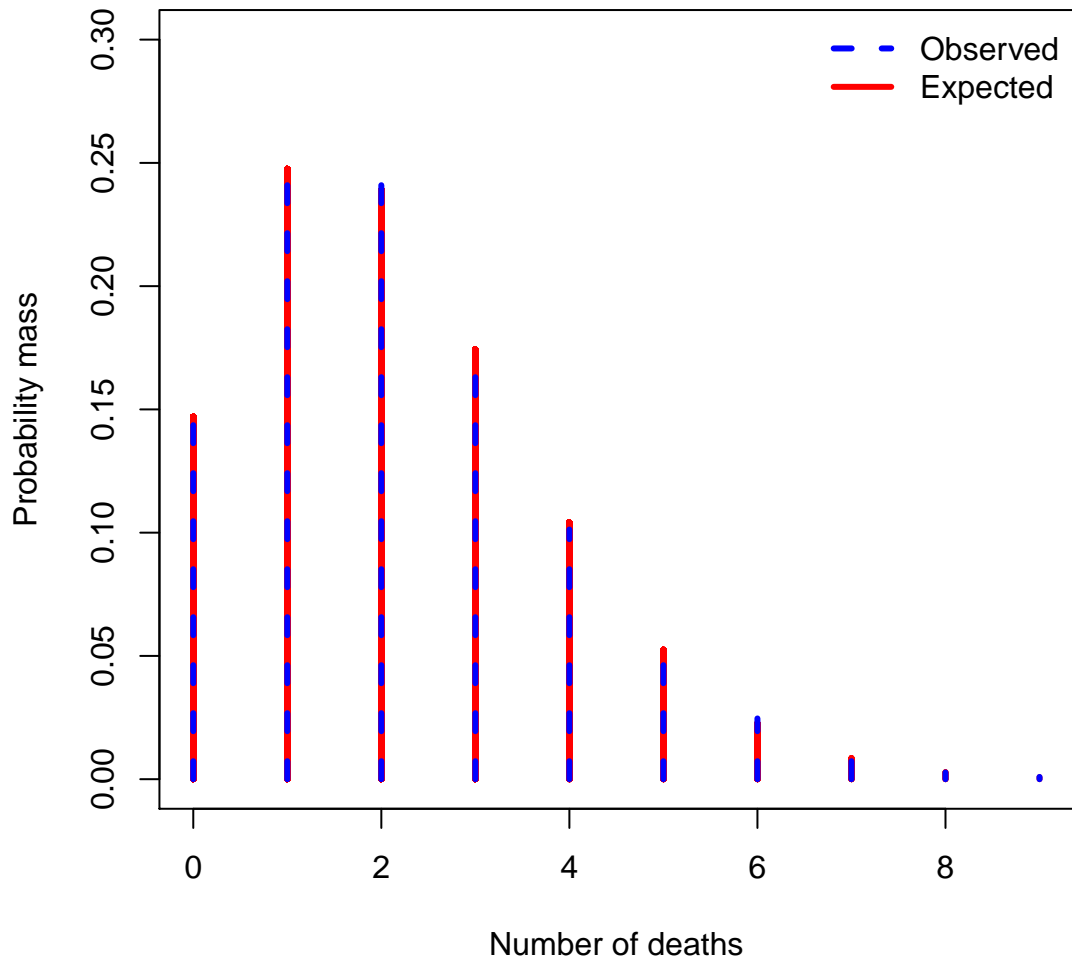
res <- em.mixture.two.poisson(y = y, theta0 = c(0.8, 1, 3), eps = 0.00001)
pest <- res[1]; lambda1est <- res[2]; lambda2est <- res[3]
pest; lambda1est; lambda2est

## [1] 0.3604639
## [1] 1.2571
## [1] 2.664111

fitmix <- pest*dpois(y, lambda = lambda1est) + (1-pest)*dpois(y, lambda = lambda2est)

plot(y, fitmix, type = "h", lwd = 3, lty = 1, col = "red", ylim = c(0, 0.3),
      xlab = "Number of deaths", ylab = "Probability mass")
```

```
points(0:9, yagg/n, lty = 2, lwd = 3, type = "h", col = "blue")
legend("topright", legend = c("Observed", "Expected"),
      lty = c(2, 1), lwd = c(3, 3), col = c("blue", "red"), bty = "n")
```



Just as an example, I illustrate how to use the (nonparametric) bootstrap to quantify uncertainty about p , λ_1 , and λ_2 .

```
B <- 1000
pb <- lambda1b <- lambda2b <- numeric(B)

set.seed(1)
for(l in 1:B){
  yb <- sample(y, size = n, replace = TRUE)
  fitb <- em.mixture.two.poisson(y = yb, theta0 = c(pest, lambda1est, lambda2est),
                                eps = 0.00001)

  pb[l] <- fitb[1]
  lambda1b[l] <- fitb[2]
  lambda2b[l] <- fitb[3]
}

ql <- function(x){quantile(x,0.025)}
qh <- function(x){quantile(x,0.975)}
```

```

pl <- ql(pb); ph <- qh(pb)
lambda1l <- ql(lambda1b); lambda1h <- qh(lambda1b)
lambda2l <- ql(lambda2b); lambda2h <- qh(lambda2b)

df <- data.frame("Estimate" = c(pest, lambda1est, lambda2est),
                 "lq" = c(pl, lambda1l, lambda2l),
                 "uq" = c(ph, lambda1h, lambda2h))

rownames(df) <- c("$p$", "$\\lambda_1$", "$\\lambda_2$")
colnames(df) <- c("Estimate", "2.5% quantile", "97.5% quantile")
knitr::kable(df, escape = FALSE, digits = 4)

```

	Estimate	2.5% quantile	97.5% quantile
p	0.3605	0.0865	0.7290
λ_1	1.2571	0.2972	1.7582
λ_2	2.6641	2.3139	3.3300

3. Our observed data are

$$y = (y_1, y_2, y_3, y_4),$$

whereas the complete data are

$$(y, z) = (z_1, y_1 - z_1, y_2, y_3, z_2, y_4 - z_2),$$

with z_1 and z_2 unobserved. The data are clearly distributed according to a multinomial distribution and therefore the likelihood of the complete data is proportional to

$$\begin{aligned}
L(\theta \mid y, z) &\propto \left(\frac{1}{4}\right)^{z_1} \left(\frac{\theta^2}{2}\right)^{y_1 - z_1} \left(\frac{1}{2}\theta(1 - \theta)\right)^{y_2 + y_3} \left(\frac{1}{4}\right)^{z_2} \left(\frac{1}{2}(1 - \theta)^2\right)^{y_4 - z_2} \\
&\propto \theta^{2y_1 + y_2 + y_3 - 2z_1} (1 - \theta)^{y_2 + y_3 + 2y_4 - 2z_2},
\end{aligned}$$

while the loglikelihood is

$$\log L(\theta \mid y, z) \propto (2y_1 + y_2 + y_3 - 2z_1) \log \theta + (y_2 + y_3 + 2y_4 - 2z_2) \log(1 - \theta).$$

We now proceed to the E-step

$$\begin{aligned}
Q(\theta \mid \theta^{(t)}) &= E_Z \left[\log L(\theta \mid y, z) \mid y, \theta^{(t)} \right] \\
&= \left(2y_1 + y_2 + y_3 - 2E[Z_1 \mid y_1, \theta^{(t)}] \right) \log \theta + \left(y_2 + y_3 + 2y_4 - 2E[Z_2 \mid y_4, \theta^{(t)}] \right) \log(1 - \theta).
\end{aligned}$$

Note that

$$\begin{aligned}
Z_1 \mid Y_1 = y_1 &\sim \text{Binomial}(y_1, p_1), & p_1 &= \frac{1/4}{1/4 + \theta^2/2} = \frac{1}{1 + 2\theta^2}, \\
Z_2 \mid Y_4 = y_4 &\sim \text{Binomial}(y_4, p_2), & p_2 &= \frac{1/4}{1/4 + (1 - \theta)^2/2} = \frac{1}{1 + 2(1 - \theta)^2}.
\end{aligned}$$

Hence,

$$\begin{aligned}
E[Z_1 \mid y_1, \theta^{(t)}] &= \frac{y_1}{1 + 2(\theta^{(t)})^2} = z_1^{(t)}, \\
E[Z_2 \mid y_4, \theta^{(t)}] &= \frac{y_4}{1 + 2(1 - \theta^{(t)})^2} = z_2^{(t)}.
\end{aligned}$$

We can now complete the E-step

$$Q(\theta \mid \theta^{(t)}) = \left(2y_1 + y_2 + y_3 - 2z_1^{(t)}\right) \log \theta + \left(y_2 + y_3 + 2y_4 - 2z_2^{(t)}\right) \log(1 - \theta).$$

For the M-step, we just need to take the derivative of $Q(\theta \mid \theta^{(t)})$

$$\frac{d}{d\theta} Q(\theta \mid \theta^{(t)}) = \frac{1}{\theta} (2y_1 + y_2 + y_3 - 2z_1^{(t)}) - \frac{1}{1-\theta} (y_2 + y_3 + 2y_4 - 2z_2^{(t)}).$$

Finally,

$$\frac{d}{d\theta} Q(\theta \mid \theta^{(t)}) = 0 \Rightarrow \theta^{(t+1)} = \frac{2y_1 + y_2 + y_3 - 2z_1^{(t)}}{2n - 2z_1^{(t)} - 2z_2^{(t)}},$$

with $n = y_1 + y_2 + y_3 + y_4$. The implementation in R is as follows.

```
multi <- function(y, theta0, eps){

  n <- sum(y)
  diff <- 1
  theta <- theta0

  while(diff > eps){
    theta.old <- theta

    #E step
    z1t <- y[1]/(1 + 2*theta^2)
    z2t <- y[4]/(1 + 2*(1-theta)^2)

    #M step
    theta <- (2*y[1] + y[2] + y[3] - 2*z1t)/(2*n - 2*z1t - 2*z2t)
    diff <- abs(theta - theta.old)
  }
  return(theta)
}

y <- c(69, 29, 25, 104)
thetamle <- multi(y = y, 0.1, 0.00001)
thetamle
```

```
## [1] 0.3554559
```

As can be seen, $\hat{\theta} = 0.355$.