

$$1. \quad Z_i \sim \text{categorical}(\pi_1, \pi_2, 1 - \pi_1 - \pi_2)$$

$$(X_i | Z_i = 1) \sim \text{Poisson}(\lambda_1)$$

$$(X_i | Z_i = 2) \sim \text{Poisson}(\lambda_2)$$

$$(X_i | Z_i = 3) \sim \text{Poisson}(\lambda_3)$$

$$f(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

$$a) \text{ Let } X = (X_1, \dots, X_{300}) \text{ and } Z = (Z_1, \dots, Z_{300})$$

$$\text{Derive } Q(\theta, \theta^0) = E_{Z|X, \theta^0} [\log(P(X, Z | \theta))]$$

$$\text{Let } n = 300$$

$$P(X, Z | \theta) = \prod_{i=1}^n P(X_i | Z_i, \theta) P(Z_i | \theta)$$

$$= \prod_{i=1}^n \prod_{k=1}^3 [P(X_i | Z_i = k, \theta) P(Z_i = k | \theta)]^{\mathbb{I}(Z_i = k)}$$

$$\log P(X, Z | \theta) = \sum_{i=1}^n \sum_{k=1}^3 \mathbb{I}(Z_i = k) [\log P(X_i | Z_i = k, \theta) + \log P(Z_i = k | \theta)]$$

$$Q(\theta, \theta^0) = \sum_{i=1}^n \sum_{k=1}^3 P(Z_i = k | X_i, \theta^0) [\log P(X_i | Z_i = k, \theta) + \log P(Z_i = k | \theta)]$$

$$= \sum_{i=1}^n \sum_{k=1}^3 P(Z_i = k | X_i, \theta^0) [X_i \log(\lambda_k) - \lambda_k - \log(X_i!) + \log \pi_k]$$

$$\text{where } \pi_3 = 1 - \pi_1 - \pi_2$$

$$b) \text{ Let } \theta^0 = (\pi_1^0, \pi_2^0, \lambda_1^0, \lambda_2^0, \lambda_3^0)$$

$$P(Z_i = k | X_i, \theta^0) = \frac{P(Z_i = k, X_i | \theta^0)}{P(X_i | \theta^0)}$$

$$= \frac{P(X_i | Z_i = k, \theta^0) P(Z_i = k | \theta^0)}{\sum_{k'=1}^3 P(X_i | Z_i = k', \theta^0) P(Z_i = k' | \theta^0)}$$

$$P(Z_i = 1 | X_i, \theta^0) = \frac{P(X_i | Z_i = 1, \theta^0) P(Z_i = 1 | \theta^0)}{P(X_i | Z_i = 1, \theta^0) P(Z_i = 1 | \theta^0) + P(X_i | Z_i = 2, \theta^0) P(Z_i = 2 | \theta^0) + P(X_i | Z_i = 3, \theta^0) P(Z_i = 3 | \theta^0)}$$

$$P(Z_i = 2 | X_i, \theta^0) = \frac{P(X_i | Z_i = 2, \theta^0) P(Z_i = 2 | \theta^0)}{P(X_i | Z_i = 1, \theta^0) P(Z_i = 1 | \theta^0) + P(X_i | Z_i = 2, \theta^0) P(Z_i = 2 | \theta^0) + P(X_i | Z_i = 3, \theta^0) P(Z_i = 3 | \theta^0)}$$

$$P(Z_i = 3 | X_i, \theta^0) = 1 - P(Z_i = 1 | X_i, \theta^0) - P(Z_i = 2 | X_i, \theta^0)$$

$$\text{where } P(X_i | Z_i = 1, \theta^0) = \frac{\lambda_1^{X_i} e^{-\lambda_1}}{X_i!}, \quad P(X_i | Z_i = 2, \theta^0) = \frac{\lambda_2^{X_i} e^{-\lambda_2}}{X_i!}, \quad P(Z_i = 1 | \theta^0) = \pi_1, \quad P(Z_i = 2 | \theta^0) = \pi_2$$

$$\begin{aligned}
 c) \text{ Let } \frac{\partial Q(\theta, \theta^0)}{\partial \pi_1} &= \sum_{i=1}^n \left[\frac{P(z_i=1|x_i, \theta^0)}{\pi_1} - \frac{P(z_i=3|x_i, \theta^0)}{1-\pi_1-\pi_2} \right] \\
 \Rightarrow \frac{(1-\pi_1-\pi_2) \left[\sum_{i=1}^n P(z_i=1|x_i, \theta^0) \right] - \pi_1 \left[\sum_{i=1}^n P(z_i=3|x_i, \theta^0) \right]}{\pi_1 (1-\pi_1-\pi_2)} &= 0 \quad (1)
 \end{aligned}$$

$$\begin{aligned}
 \frac{\partial Q(\theta, \theta^0)}{\partial \pi_2} &= \sum_{i=1}^n \left[\frac{P(z_i=2|x_i, \theta^0)}{\pi_2} - \frac{P(z_i=3|x_i, \theta^0)}{1-\pi_1-\pi_2} \right] \\
 \Rightarrow \frac{(1-\pi_1-\pi_2) \left[\sum_{i=1}^n P(z_i=2|x_i, \theta^0) \right] - \pi_2 \left[\sum_{i=1}^n P(z_i=3|x_i, \theta^0) \right]}{\pi_2 (1-\pi_1-\pi_2)} &= 0 \quad (2)
 \end{aligned}$$

$$\text{from (1)} \Rightarrow (1-\pi_1-\pi_2) \left[\sum_{i=1}^n P(z_i=1|x_i, \theta^0) \right] = \pi_1 \left[\sum_{i=1}^n P(z_i=3|x_i, \theta^0) \right] \quad (1)'$$

$$\text{from (2)} \Rightarrow (1-\pi_1-\pi_2) \left[\sum_{i=1}^n P(z_i=2|x_i, \theta^0) \right] = \pi_2 \left[\sum_{i=1}^n P(z_i=3|x_i, \theta^0) \right] \quad (2)'$$

$$\begin{aligned}
 (1)' + (2)' &\Rightarrow (1-\pi_1-\pi_2) \left[\sum_{i=1}^n (P(z_i=1|x_i, \theta^0) + P(z_i=2|x_i, \theta^0)) \right] \\
 &= (\pi_1 + \pi_2) \left[\sum_{i=1}^n P(z_i=3|x_i, \theta^0) \right]
 \end{aligned}$$

$$1 - \pi_1 - \pi_2 = \frac{\sum_{i=1}^n P(z_i=3|x_i, \theta^0)}{\sum_{i=1}^n (P(z_i=1|x_i, \theta^0) + P(z_i=2|x_i, \theta^0) + P(z_i=3|x_i, \theta^0))}$$

$$= \frac{\sum_{i=1}^n P(z_i=3|x_i, \theta^0)}{n}$$

$$\begin{aligned}
 \text{from (1)'} \Rightarrow \hat{\pi}_1 &= (1-\pi_1-\pi_2) \left[\sum_{i=1}^n P(z_i=1|x_i, \theta^0) \right] \\
 &= \frac{\sum_{i=1}^n P(z_i=3|x_i, \theta^0)}{n} \cdot \frac{\left[\sum_{i=1}^n P(z_i=1|x_i, \theta^0) \right]}{\left[\sum_{i=1}^n P(z_i=3|x_i, \theta^0) \right]} \\
 &= \frac{\sum_{i=1}^n P(z_i=1|x_i, \theta^0)}{n}
 \end{aligned}$$

$$\begin{aligned}
 \text{from (2)'} \Rightarrow \hat{\pi}_2 &= \frac{\sum_{i=1}^n P(z_i=2|x_i, \theta^0)}{n} & \hat{\pi}_3 &= 1 - \hat{\pi}_1 - \hat{\pi}_2
 \end{aligned}$$

$$\begin{aligned}
 \text{Let } \frac{\partial Q(\theta, \theta^0)}{\partial \lambda_k} &= \sum_{i=1}^n P(z_i=k|x_i, \theta^0) \left[-1 + \frac{x_i}{\lambda_k} \right] \\
 &= \frac{\sum_{i=1}^n P(z_i=k|x_i, \theta^0) x_i - \lambda_k \sum_{i=1}^n P(z_i=k|x_i, \theta^0)}{\lambda_k} = 0
 \end{aligned}$$

$$\Rightarrow \lambda_k \sum_{i=1}^n P(z_i=k|x_i, \theta^0) = \sum_{i=1}^n P(z_i=k|x_i, \theta^0) x_i$$

$$\Rightarrow \hat{\lambda}_k = \frac{\sum_{i=1}^n P(z_i=k|x_i, \theta^0) x_i}{\sum_{i=1}^n P(z_i=k|x_i, \theta^0)}$$

2. $Z_i \sim \text{categorical}(\pi_1, \pi_2, 1-\pi_1-\pi_2)$

$$(X_i | Z_i = 1) \sim \text{Poisson}(\lambda_1)$$

$$(X_i | Z_i = 2) \sim \text{Poisson}(\lambda_2) \quad \text{for } i=1, \dots, 300$$

$$(X_i | Z_i = 3) \sim \text{Poisson}(\lambda_3)$$

$$f(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

$$X_i \sim \text{Poisson}(\lambda_2) \quad \text{for } i=301, \dots, 400$$

a) Let $X = (X_1, \dots, X_{400})$ and $Z = (Z_1, \dots, Z_{300})$

$$\text{Derive } Q(\theta, \theta^0) = E_{Z|X, \theta^0} [\log(P(X, Z|\theta))]$$

Let $n=300$

$$P(X, Z|\theta) = \prod_{i=1}^n [P(X_i | Z_i, \theta) P(Z_i | \theta)] \prod_{i=301}^{400} P(X_i)$$

$$= \prod_{i=1}^n \prod_{k=1}^3 [P(X_i | Z_i = k, \theta) P(Z_i = k | \theta)]^{\mathbb{I}(Z_i = k)} \prod_{i=301}^{400} P(X_i)$$

$$\log P(X, Z|\theta) = \sum_{i=1}^n \sum_{k=1}^3 (\mathbb{I}(Z_i = k) [\log P(X_i | Z_i = k, \theta) + \log P(Z_i = k | \theta)]) + \sum_{i=301}^{400} \log P(X_i)$$

$$Q(\theta, \theta^0) = \sum_{i=1}^n \sum_{k=1}^3 P(Z_i = k | X_i, \theta^0) [\log P(X_i | Z_i = k, \theta) + \log P(Z_i = k | \theta)] + \sum_{i=301}^{400} \log P(X_i)$$

$$= \sum_{i=1}^n \sum_{k=1}^3 P(Z_i = k | X_i, \theta^0) [X_i \log(\lambda_k) - \lambda_k - \log(X_i!) + \log \pi_k] + \sum_{i=301}^{400} [X_i \log(\lambda_2) - \lambda_2 - \log(X_i!)]$$

where $\pi_3 = 1 - \pi_1 - \pi_2$

b) E step:

$$\text{Let } \theta^0 = (\pi_1^0, \pi_2^0, \lambda_1^0, \lambda_2^0, \lambda_3^0)$$

$$P(Z_i = k | X_i, \theta^0) = \frac{P(Z_i = k, X_i | \theta^0)}{P(X_i | \theta^0)}$$

$$= \frac{P(X_i | Z_i = k, \theta^0) P(Z_i = k | \theta^0)}{\sum_{k'=1}^3 P(X_i | Z_i = k', \theta^0) P(Z_i = k' | \theta^0)}$$

$$P(Z_i = 1 | X_i, \theta^0) = \frac{P(X_i | Z_i = 1, \theta^0) P(Z_i = 1 | \theta^0)}{P(X_i | Z_i = 1, \theta^0) P(Z_i = 1 | \theta^0) + P(X_i | Z_i = 2, \theta^0) P(Z_i = 2 | \theta^0) + P(X_i | Z_i = 3, \theta^0) P(Z_i = 3 | \theta^0)}$$

$$P(Z_i = 2 | X_i, \theta^0) = \frac{P(X_i | Z_i = 2, \theta^0) P(Z_i = 2 | \theta^0)}{P(X_i | Z_i = 1, \theta^0) P(Z_i = 1 | \theta^0) + P(X_i | Z_i = 2, \theta^0) P(Z_i = 2 | \theta^0) + P(X_i | Z_i = 3, \theta^0) P(Z_i = 3 | \theta^0)}$$

$$P(Z_i = 3 | X_i, \theta^0) = 1 - P(Z_i = 1 | X_i, \theta^0) - P(Z_i = 2 | X_i, \theta^0)$$

$$\text{where } P(X_i | Z_i = 1, \theta^0) = \frac{\lambda_1^{X_i} e^{-\lambda_1}}{X_i!}, P(X_i | Z_i = 2, \theta^0) = \frac{\lambda_2^{X_i} e^{-\lambda_2}}{X_i!}, P(Z_i = 1 | \theta^0) = \pi_1, P(Z_i = 2 | \theta^0) = \pi_2$$

M step:

The new summation is removed from all partial derivatives except for $\frac{\partial Q(\theta, \theta^0)}{\partial \lambda_2}$ so we copy our answers from 1c

$$\Rightarrow \hat{\pi}_1 = \frac{\sum_{i=1}^n P(Z_i = 1 | X_i, \theta^0)}{n}$$

$$\Rightarrow \hat{\pi}_2 = \frac{\sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0)}{n} \quad \hat{\pi}_3 = 1 - \hat{\pi}_1 - \hat{\pi}_2$$

$$\text{Let } \frac{\partial Q(\theta, \theta^0)}{\partial \lambda_k} = \sum_{i=1}^n P(Z_i = k | X_i, \theta^0) \left[-1 + \frac{X_i}{\lambda_k} \right] \quad \text{for } k \neq 2$$

$$= \frac{\sum_{i=1}^n P(Z_i = k | X_i, \theta^0) X_i - \lambda_k \sum_{i=1}^n P(Z_i = k | X_i, \theta^0)}{\lambda_k}$$

$$\lambda_k \sum_{i=1}^n P(Z_i = k | X_i, \theta^0) = \sum_{i=1}^n P(Z_i = k | X_i, \theta^0) X_i$$

$$\Rightarrow \hat{\lambda}_k = \frac{\sum_{i=1}^n P(Z_i = k | X_i, \theta^0) X_i}{\sum_{i=1}^n P(Z_i = k | X_i, \theta^0)} \quad \text{for } k \neq 2$$

$$\frac{\partial Q(\theta, \theta^0)}{\partial \lambda_2} = \sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0) \left[-1 + \frac{X_i}{\lambda_2} \right] + \sum_{i=301}^{400} \left[-1 + \frac{X_i}{\lambda_2} \right]$$

$$= \frac{\sum_{i=1}^n [P(Z_i = 2 | X_i, \theta^0) X_i - \lambda_2 \sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0)] + \sum_{i=301}^{400} [X_i - \lambda_2]}{\lambda_2} = 0$$

$$= \frac{\sum_{i=1}^n [P(Z_i = 2 | X_i, \theta^0) X_i - \lambda_2 \sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0)] + \sum_{i=301}^{400} X_i - 100 \lambda_2}{\lambda_2} = 0$$

$$\Rightarrow \lambda_2 \sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0) + 100 \lambda_2 = \sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0) X_i + \sum_{i=301}^{400} X_i$$

$$\Rightarrow \lambda_2 \left(\sum_{i=1}^n [P(Z_i = 2 | X_i, \theta^0)] + 100 \right) = \sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0) X_i + \sum_{i=301}^{400} X_i$$

$$\Rightarrow \hat{\lambda}_2 = \frac{\sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0) X_i + \sum_{i=301}^{400} X_i}{\sum_{i=1}^n P(Z_i = 2 | X_i, \theta^0) + 100}$$

MAST30027 Modern Applied Statistics Assignment 3

Tutorial: Wed 1-2PM, Yidi Deng

James La Fontaine

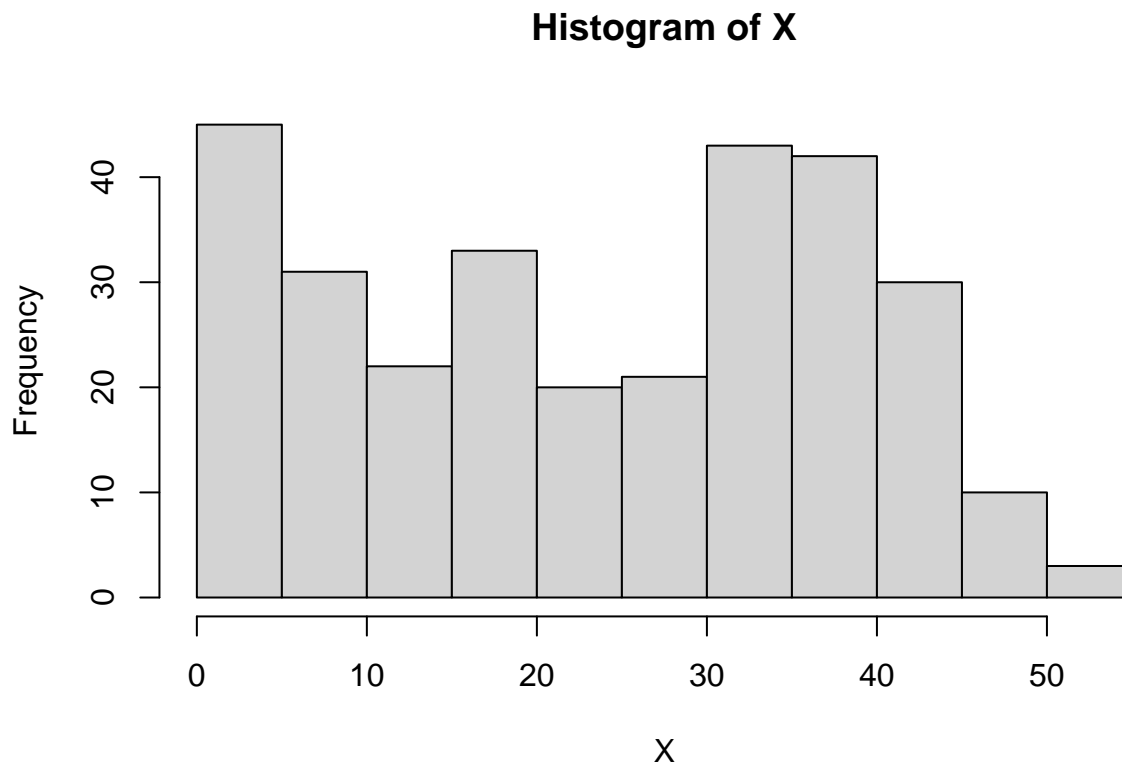
2023-09-24

Question 1d

```
X = scan(file="assignment3_prob1_2023.txt", what=double())  
length(X)
```

```
## [1] 300
```

```
hist(X)
```



Implementation of the EM algorithm

We will assume that the observed data follows a mixture of three Poisson distributions. Specifically, for $i = 1 \dots, n$,

$$Z_i \sim \text{categorical}(\pi_1, \pi_2, 1 - \pi_1 - \pi_2),$$

$$X_i | Z_i = 1 \sim \text{Poisson}(\lambda_1),$$

$$X_i | Z_i = 2 \sim \text{Poisson}(\lambda_2),$$

$$X_i | Z_i = 3 \sim \text{Poisson}(\lambda_3).$$

We aim to obtain MLE of parameters $\theta = (\pi_1, \pi_2, \lambda_1, \lambda_2, \lambda_3)$ using the EM algorithm.

We implement the E and M step in the `EM.iter` function below. The `compute.log.lik` function below computes the incomplete log-likelihood, assuming the parameters are known. We will check that the incomplete log-likelihoods increases at each step by plotting them. The `mixture.EM` function is the main function which runs multiple EM steps and checks for convergence by computing the incomplete log-likelihoods at each step.

```
# w.init : initial value for pi
# lambda.init : initial value for lambda
# epsilon : If the incomplete log-likelihood has changed by less than epsilon,
# EM will stop.
# max.iter : maximum number of EM-iterations
mixture.EM <- function(X, w.init, lambda.init, epsilon=1e-5, max.iter=100) {

  w.curr = w.init
  lambda.curr = lambda.init

  # store incomplete log-likelihoods for each iteration
  log_lik = c()

  # compute incomplete log-likelihoods using initial values of parameters.
  log_lik = c(log_lik, compute.log.lik(X, w.curr, lambda.curr)$ll)

  # set the change in incomplete log-likelihood with 1
  delta.ll = 1

  # number of iteration
  n.iter = 1

  # If the log-likelihood has changed by less than epsilon, EM will stop.
  while((delta.ll > epsilon) & (n.iter <= max.iter)){

    # run EM step
    EM.out = EM.iter(X, w.curr, lambda.curr)

    # replace the current value with the new parameter estimate
    w.curr = EM.out$w.new
    lambda.curr = EM.out$lambda.new

    # incomplete log-likelihoods with new parameter estimate
    log_lik = c(log_lik, compute.log.lik(X, w.curr, lambda.curr)$ll)

    # compute the change in incomplete log-likelihood
    delta.ll = log_lik[length(log_lik)] - log_lik[length(log_lik)-1]

    # increase the number of iteration
    n.iter = n.iter + 1
  }
  return(list(w.curr=w.curr, lambda.curr=lambda.curr, log_lik=log_lik))
}
```

```
EM.iter <- function(X, w.curr, lambda.curr) {

  # E-step: compute  $E_{Z|X, \theta_0}[I(Z_i = k)]$ 
```



```

# for each sample $X_i$, compute $P(X_i, Z_i=k)$
prob.x.z = compute.prob.x.z(X, w.curr, lambda.curr)$prob.x.z

# compute $P(Z_i=k | X_i)$
P_ik = prob.x.z / rowSums(prob.x.z)

# M-step
w.new = colSums(P_ik)/sum(P_ik) # sum(P_ik) is equivalent to sample size
lambda.new = colSums(P_ik*X)/colSums(P_ik)

return(list(w.new=w.new, lambda.new=lambda.new))
}

```

Now we write a function to compute the incomplete log-likelihood, assuming the parameters are known.

$$\ell(\theta) = \sum_{i=1}^n \log \left(\sum_{k=1}^3 \pi_k f(x_i; \lambda_k) \right)$$

```

# Compute incomplete log-likelihoods
compute.log.lik <- function(X, w.curr, lambda.curr) {

  # for each sample $X_i$, compute $P(X_i, Z_i=k)$
  prob.x.z = compute.prob.x.z(X, w.curr, lambda.curr)$prob.x.z

  # incomplete log-likelihoods
  ill = sum(log(rowSums(prob.x.z)))

  return(list(ill=ill))
}

# for each sample $X_i$, compute $P(X_i, Z_i=k)$
compute.prob.x.z <- function(X, w.curr, lambda.curr) {

  # for each sample $X_i$, compute $P(X_i, Z_i=k)$. Store these values in the columns of L:
  L = matrix(NA, nrow=length(X), ncol=length(w.curr))
  for(k in seq_len(ncol(L))) {
    L[, k] = dpois(X, lambda=lambda.curr[k])*w.curr[k]
  }

  return(list(prob.x.z=L))
}

```

Apply the EM algorithm

Run EM algorithm with different initial values and check that the incomplete log-likelihoods increases at each step by plotting them.

```

EM1 <- mixture.EM(X, w.init=c(0.3,0.3, 0.4), lambda.init=c(3, 20, 35), epsilon=1e-5, max.iter=100)
ee = EM1
print(paste("Estimate pi = (", round(ee$w.curr[1],2), ",",
  round(ee$w.curr[2],2), ",",
  round(ee$w.curr[3],2), ")", sep=""))

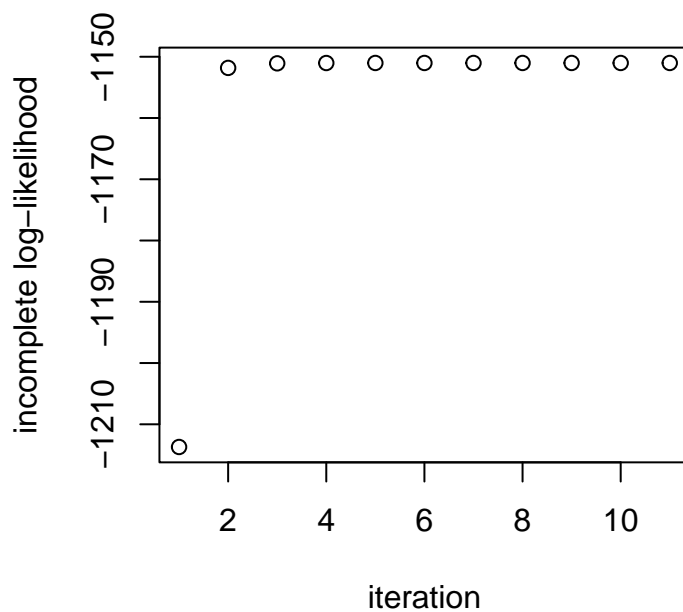
```

```
## [1] "Estimate pi = (0.25,0.25,0.5)"
```

```
print(paste("Estimate lambda = (", round(ee$lambda.curr[1],2), ",",  
          round(ee$lambda.curr[2],2), ",",  
          round(ee$lambda.curr[3],2), ")", sep=""))
```

```
## [1] "Estimate lambda = (5.17,18.09,36.94)"
```

```
plot(ee$log_lik, ylab='incomplete log-likelihood', xlab='iteration')
```



```
EM2 <- mixture.EM(X, w.init=c(0.1,0.2, 0.7), lambda.init=c(5, 25, 40), epsilon=1e-5, max.iter=100)  
ee = EM2
```

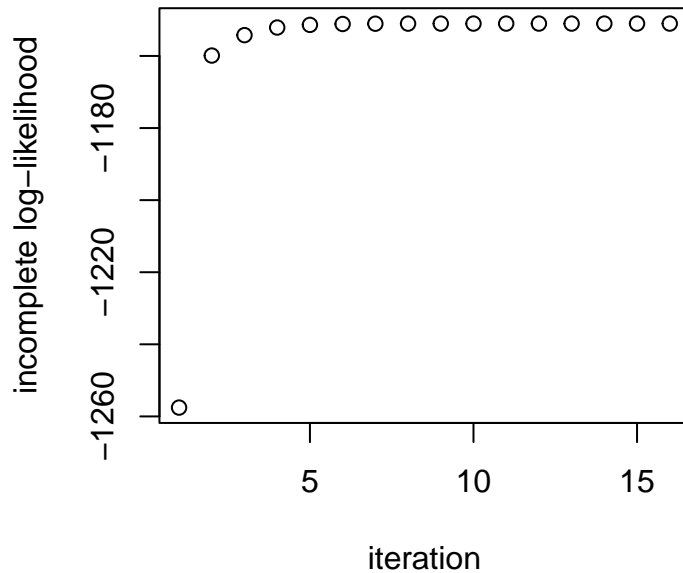
```
print(paste("Estimate pi = (", round(ee$w.curr[1],2), ",",  
          round(ee$w.curr[2],2), ",",  
          round(ee$w.curr[3],2), ")", sep=""))
```

```
## [1] "Estimate pi = (0.25,0.25,0.5)"
```

```
print(paste("Estimate lambda = (", round(ee$lambda.curr[1],2), ",",  
          round(ee$lambda.curr[2],2), ",",  
          round(ee$lambda.curr[3],2), ")", sep=""))
```

```
## [1] "Estimate lambda = (5.17,18.09,36.94)"
```

```
plot(ee$log_lik, ylab='incomplete log-likelihood', xlab='iteration')
```



Check which estimators have the highest incomplete log-likelihood.

```
EM1$log_lik[length(EM1$log_lik)]
```

```
## [1] -1151.015
```

```
EM2$log_lik[length(EM2$log_lik)]
```

```
## [1] -1151.015
```

Estimators from the two EM runs have (equally) highest incomplete log-likelihoods. You can see that the estimators from the EM runs are the same, so it doesn't matter which estimators we choose. Lets choose the estimators from the first EM run - $\hat{\pi}_1 = 0.25$, $\hat{\pi}_2 = 0.25$, $\hat{\lambda}_1 = 5.17$, $\hat{\lambda}_2 = 18.09$, $\hat{\lambda}_3 = 36.94$.

Question 2d

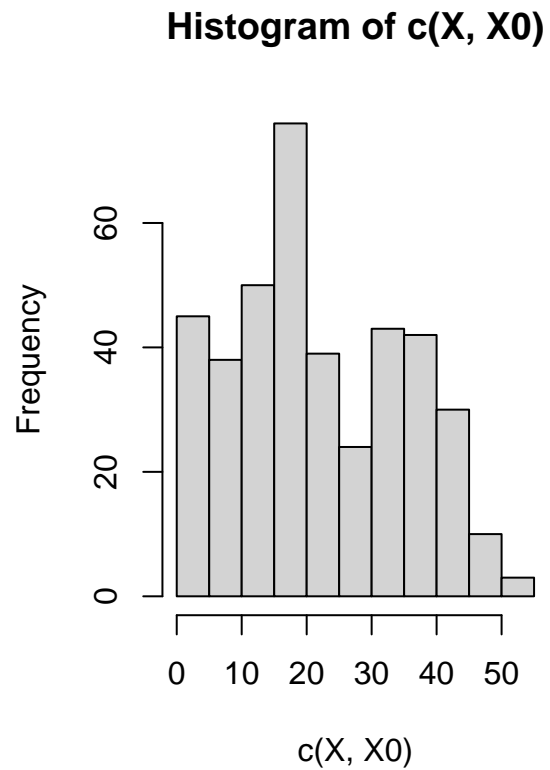
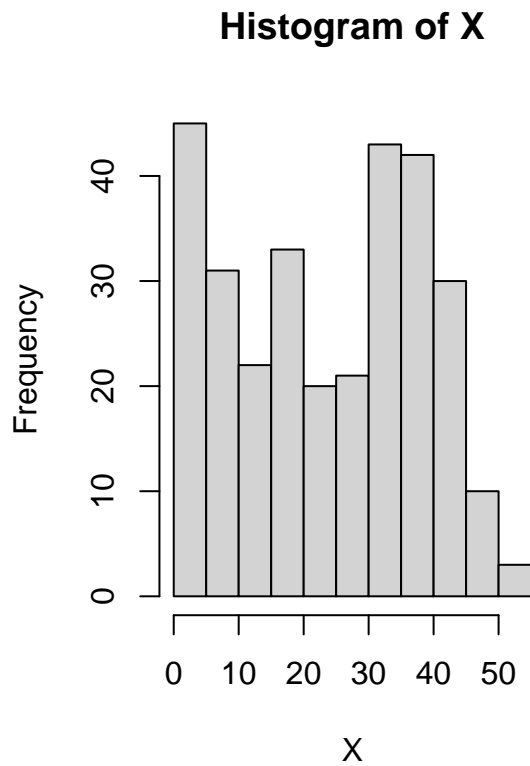
```
X = scan(file="assignment3_prob1_2023.txt", what=double())  
  
X0 = scan(file="assignment3_prob2_2023.txt", what=double())  
  
length(X)
```

```
## [1] 300
```

```
length(X0)
```

```
## [1] 100
```

```
par(mfrow=c(1,2))  
  
hist(X)  
  
hist(c(X,X0))
```



Implementation of the EM algorithm

We will assume that the observed data follows a mixture of three Poisson distributions. Additionally, for samples 301 to 400, we assume that they follow a known Poisson distribution. Specifically, for $i = 1 \dots, n$,

$$Z_i \sim \text{categorical}(\pi_1, \pi_2, 1 - \pi_1 - \pi_2),$$

$$X_i|Z_i = 1 \sim \text{Poisson}(\lambda_1),$$

$$X_i|Z_i = 2 \sim \text{Poisson}(\lambda_2),$$

$$X_i|Z_i = 3 \sim \text{Poisson}(\lambda_3),$$

and for $i=301, \dots, 400$,

$$X_i \sim \text{Poisson}(\lambda_2).$$

We aim to obtain MLE of parameters $\theta = (\pi_1, \pi_2, \lambda_1, \lambda_2, \lambda_3)$ using the EM algorithm.

We implement the E and M step in the `EM.iter` function below. The `compute.log.lik` function below computes the incomplete log-likelihood, assuming the parameters are known. We will check that the incomplete log-likelihoods increases at each step by plotting them. The `mixture.EM` function is the main function which runs multiple EM steps and checks for convergence by computing the incomplete log-likelihoods at each step.

```
# w.init : initial value for pi
# lambda.init : initial value for lambda
# epsilon : If the incomplete log-likelihood has changed by less than epsilon,
# EM will stop.
# max.iter : maximum number of EM-iterations
mixture.EM <- function(X, w.init, lambda.init, epsilon=1e-5, max.iter=100) {

  w.curr = w.init
  lambda.curr = lambda.init

  # store incomplete log-likelihoods for each iteration
  log_lik = c()

  # compute incomplete log-likelihoods using initial values of parameters.
  log_lik = c(log_lik, compute.log.lik(X, w.curr, lambda.curr)$ll)

  # set the change in incomplete log-likelihood with 1
  delta.ll = 1

  # number of iteration
  n.iter = 1

  # If the log-likelihood has changed by less than epsilon, EM will stop.
  while((delta.ll > epsilon) & (n.iter <= max.iter)){

    # run EM step
    EM.out = EM.iter(X, w.curr, lambda.curr)

    # replace the current value with the new parameter estimate
    w.curr = EM.out$w.new
    lambda.curr = EM.out$lambda.new

    # incomplete log-likelihoods with new parameter estimate
```

```

log_liks = c(log_liks, compute.log.lik(X, w.curr, lambda.curr)$ill)

# compute the change in incomplete log-likelihood
delta.ll = log_liks[length(log_liks)] - log_liks[length(log_liks)-1]

# increase the number of iteration
n.iter = n.iter + 1
}
return(list(w.curr=w.curr, lambda.curr=lambda.curr, log_liks=log_liks))
}

```

```

EM.iter <- function(X, w.curr, lambda.curr) {

# E-step: compute  $E_{\{Z|X, \theta_0\}}[I(Z_i = k)]$ 

# for each sample  $X_i$ , compute  $P(X_i, Z_i=k)$ 
prob.x.z = compute.prob.x.z(X, w.curr, lambda.curr)$prob.x.z

# compute  $P(Z_i=k | X_i)$ 
P_ik = prob.x.z / rowSums(prob.x.z)

# M-step
w.new = colSums(P_ik)/sum(P_ik) # sum(P_ik) is equivalent to sample size
lambda.new = colSums(P_ik*X)/colSums(P_ik)
lambda.new[2] = (colSums(P_ik*X)[2] + sum(X0))/(colSums(P_ik)[2] + 100)

#print(paste("lambda.new = (", lambda.new, ")", sep=""))

return(list(w.new=w.new, lambda.new=lambda.new))
}

```

Now we write a function to compute the incomplete log-likelihood, assuming the parameters are known.

$$\ell(\theta) = \sum_{i=1}^n \log \left(\sum_{k=1}^3 \pi_k f(x_i; \lambda_k) \right)$$

```

# Compute incomplete log-likelihoods
compute.log.lik <- function(X, w.curr, lambda.curr) {

# for each sample  $X_i$ , compute  $P(X_i, Z_i=k)$ 
prob.x.z = compute.prob.x.z(X, w.curr, lambda.curr)$prob.x.z

# incomplete log-likelihoods
ill = sum(log(rowSums(prob.x.z)))

return(list(ill=ill))
}

# for each sample  $X_i$ , compute  $P(X_i, Z_i=k)$ 
compute.prob.x.z <- function(X, w.curr, lambda.curr) {

# for each sample  $X_i$ , compute  $P(X_i, Z_i=k)$ . Store these values in the columns of L:

```

```

L = matrix(NA, nrow=length(X), ncol= length(w.curr))
for(k in seq_len(ncol(L))) {
  L[, k] = dpois(X, lambda=lambda.curr[k])*w.curr[k]
}

return(list(prob.x.z=L))
}

```

Apply the EM algorithm

Run EM algorithm with different initial values and check that the incomplete log-likelihoods increases at each step by plotting them.

```

EM1 <- mixture.EM(X, w.init=c(0.3,0.3, 0.4), lambda.init=c(3, 20, 35), epsilon=1e-5, max.iter=100)
ee = EM1
print(paste("Estimate pi = (", round(ee$w.curr[1],2), ",",
        round(ee$w.curr[2],2), ",",
        round(ee$w.curr[3],2), ")", sep=""))

```

```
## [1] "Estimate pi = (0.25,0.25,0.51)"
```

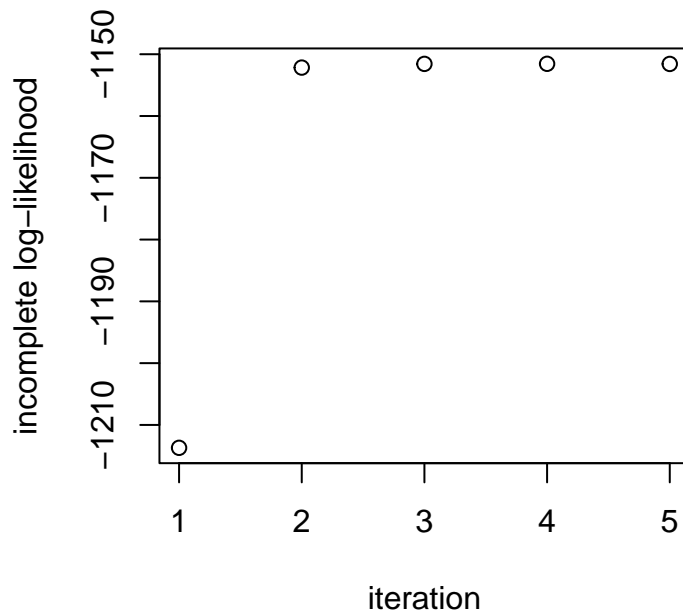
```

print(paste("Estimate lambda = (", round(ee$lambda.curr[1],2), ",",
        round(ee$lambda.curr[2],2), ",",
        round(ee$lambda.curr[3],2), ")", sep=""))

```

```
## [1] "Estimate lambda = (5.12,17.38,36.79)"
```

```
plot(ee$log_lik, ylab='incomplete log-likelihood', xlab='iteration')
```



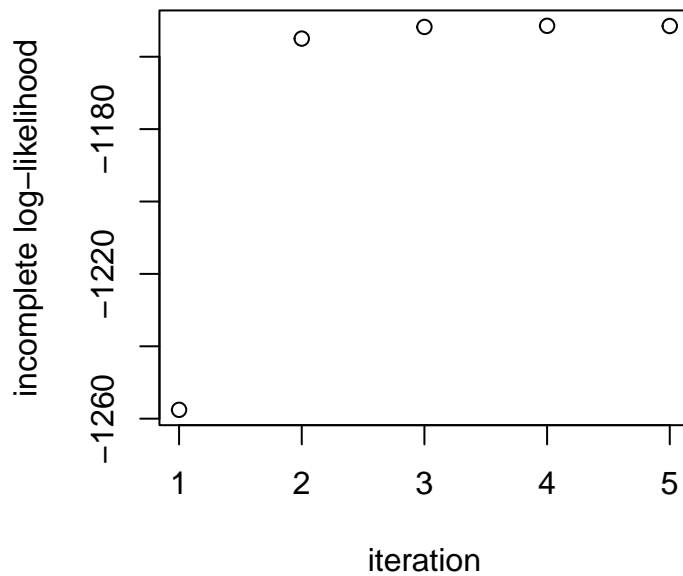
```
EM2 <- mixture.EM(X, w.init=c(0.1,0.2, 0.7), lambda.init=c(5, 25, 40), epsilon=1e-5, max.iter=100)
ee = EM2
print(paste("Estimate pi = (", round(ee$w.curr[1],2), ",",
          round(ee$w.curr[2],2), ",",
          round(ee$w.curr[3],2), ")", sep=""))
```

```
## [1] "Estimate pi = (0.25,0.25,0.5)"
```

```
print(paste("Estimate lambda = (", round(ee$lambda.curr[1],2), ",",
          round(ee$lambda.curr[2],2), ",",
          round(ee$lambda.curr[3],2), ")", sep=""))
```

```
## [1] "Estimate lambda = (5.13,17.43,36.86)"
```

```
plot(ee$log_lik, ylab='incomplete log-likelihood', xlab='iteration')
```

Check which estimators have the highest incomplete log-likelihood.

```
EM1$log_lik[length(EM1$log_lik)]
```

```
## [1] -1151.57
```

```
EM2$log_lik[length(EM2$log_lik)]
```

```
## [1] -1151.507
```

Estimators from the two EM runs nearly have equally highest incomplete log-likelihoods. You can see that the incomplete log-likelihood is slightly higher for the estimators from the second EM run and so we will choose the estimators from the second EM run - $\hat{\pi}_1 = 0.25$, $\hat{\pi}_2 = 0.25$, $\hat{\lambda}_1 = 5.13$, $\hat{\lambda}_2 = 17.43$, $\hat{\lambda}_3 = 36.86$.