MAST30025 Assingment 2

Kim Seang CHY

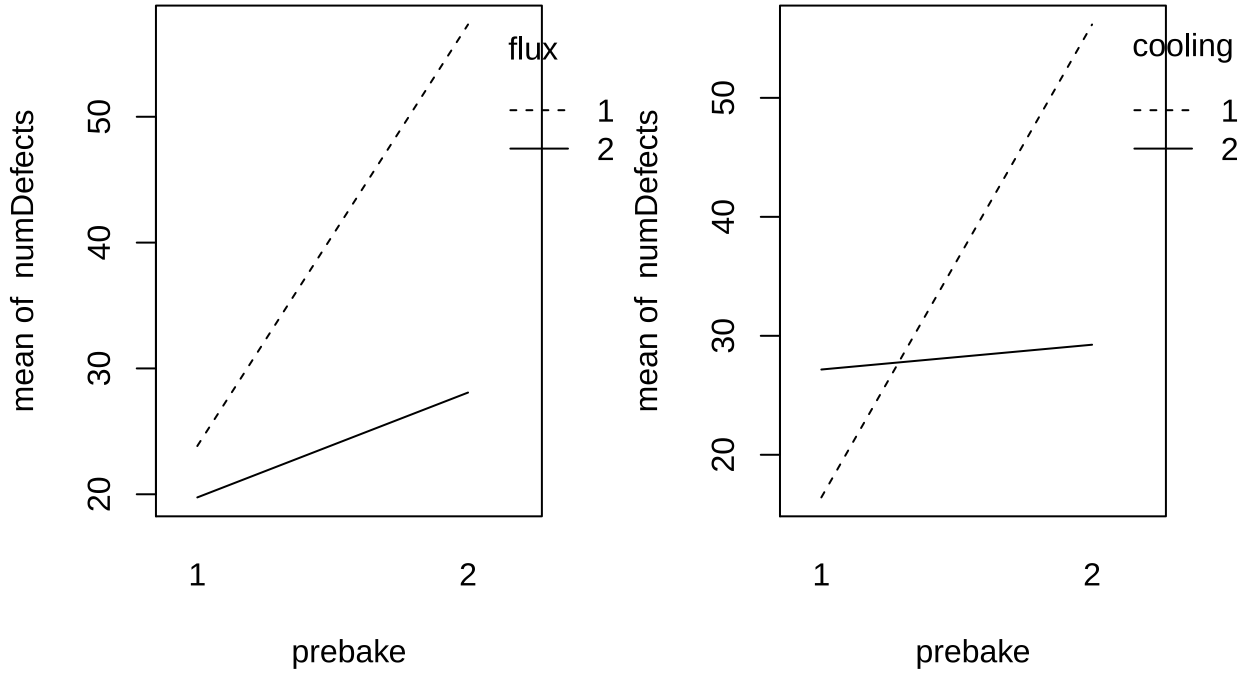
998008

Question 1:

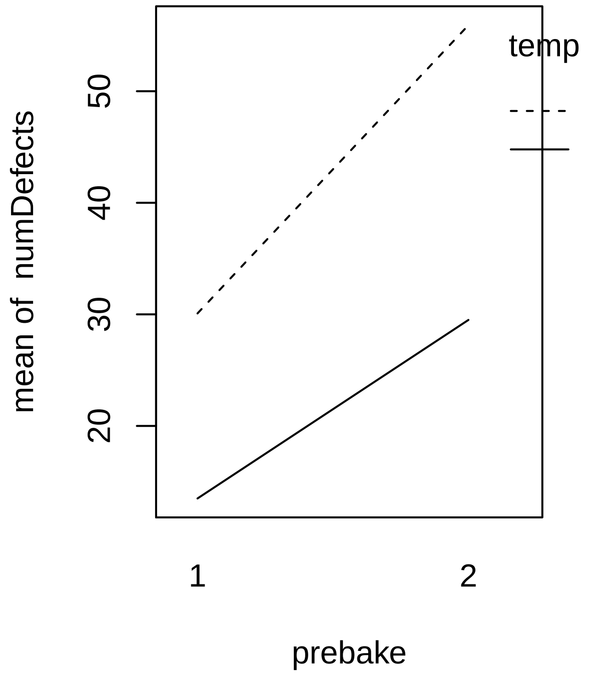
```
q1.data <- read.table(file ="assignment2_prob1_2021.txt", header=TRUE)
names(data)
```

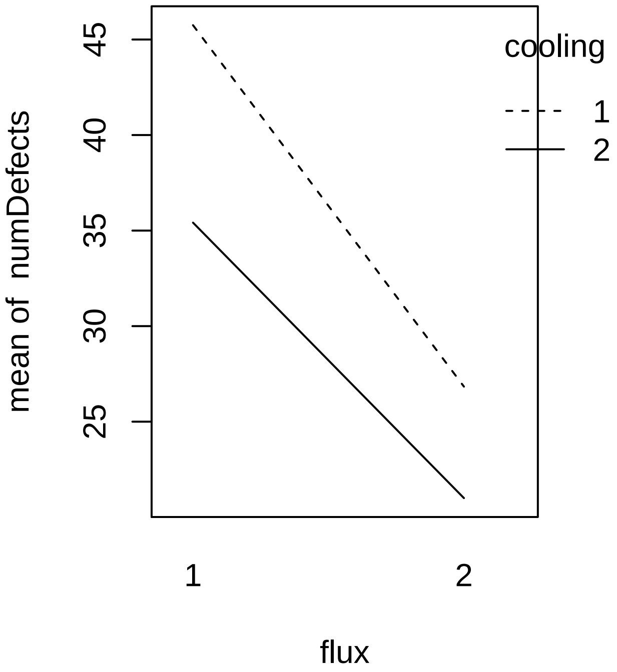
```
## NULL
```

```
q1.data$prebake <- factor(q1.data$prebake)
q1.data$flux <- factor(q1.data$flux)
q1.data$cooling <- factor(q1.data$cooling)
q1.data$temp <- factor(q1.data$temp)

#Plotting Interaction plot
par(mfrow=c(1,2))
with(data=q1.data, interaction.plot(prebake, flux, numDefects))
with(data=q1.data, interaction.plot(prebake, cooling, numDefects))
```

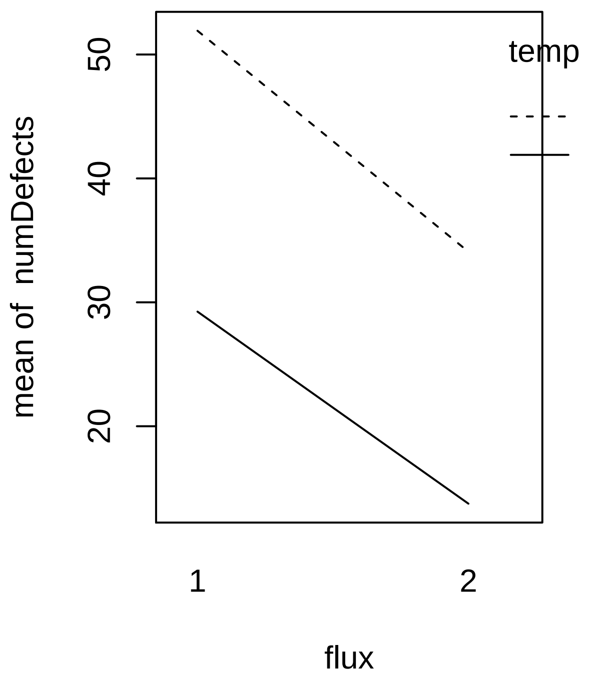


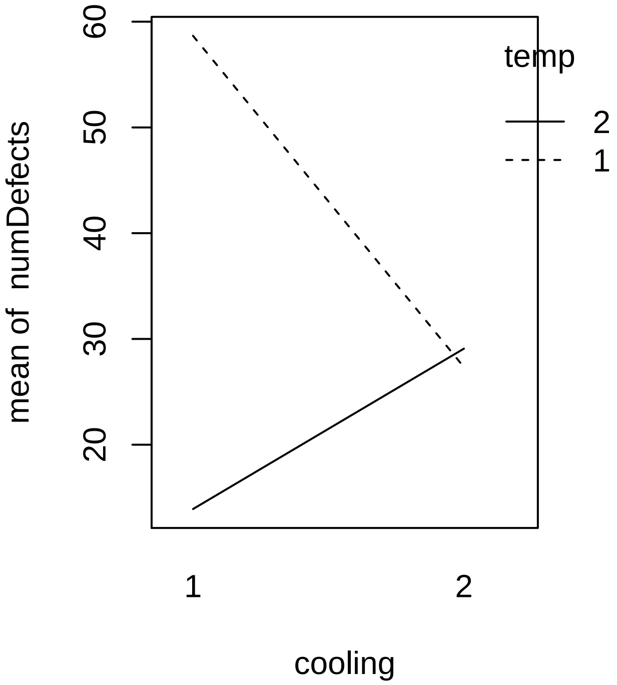
```
with(data=q1.data, interaction.plot(prebake, temp, numDefects))
with(data=q1.data, interaction.plot(flux, cooling, numDefects))
```





```
with(data=q1.data, interaction.plot(flux, temp, numDefects))
with(data=q1.data, interaction.plot(cooling, temp, numDefects))
```





I plot the interaction plot to see all the possible interaction.Looking at all combination of interactions beside the two above the rest the line of the interaction plot are very close to parallel thus indincating there are no interaction. While, prebake and cooling has a clear indication of interaaction, cooling and temp does not and will require further invetigation

here seem to also be interaction between Prebake condition and cooling time. Thus suggesting the additive model is inadequate

Looking at fundamental of data set, we are testing various factors on number of defects thus it is more an independence bernulli variable with different probability of defects, thus we will model it using a poisson.

```
Amodel <- glm(numDefects~., family = "poisson", data = q1.data)
summary(Amodel)
```

```
##
## Call:
## glm(formula = numDefects ~ ., family = "poisson", data = q1.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -11.196   -2.961   -1.116    1.730   17.544
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.71712    0.05584  66.571  < 2e-16 ***
## prebake2     0.67287    0.05374  12.522  < 2e-16 ***
## flux2       -0.52878    0.05262 -10.049  < 2e-16 ***
## cooling2    -0.25197    0.05124  -4.918 8.75e-07 ***
## temp2       -0.69315    0.05392 -12.856  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1450.52  on 47  degrees of freedom
## Residual deviance:  980.49  on 43  degrees of freedom
## AIC: 1221.2
##
## Number of Fisher Scoring iterations: 5
```

The inadequacy of the additive was further supported here as the model has deviance of 980.49 relative the 43 degree of freedoms.

```
#Fullmodel
Fullmodel <- glm(numDefects~ (prebake+flux+cooling+temp)^2, family = "poisson", data=q1.data)
summary(Fullmodel)
```

```
##
## Call:
## glm(formula = numDefects ~ (prebake + flux + cooling + temp)^2,
##     family = "poisson", data = q1.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.8230  -2.6586  -0.3118   1.4014  13.4735
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        3.40257    0.09099  37.395  < 2e-16 ***
## prebake2           1.33522    0.09850  13.555  < 2e-16 ***
## flux2             -0.07552    0.11253  -0.671   0.5022
## cooling2           0.08918    0.11330   0.787   0.4312
## temp2             -1.87164    0.14963 -12.508  < 2e-16 ***
## prebake2:flux2    -0.52572    0.11601  -4.532 5.85e-06 ***
## prebake2:cooling2 -1.40598    0.12406 -11.333  < 2e-16 ***
## prebake2:temp2     0.66473    0.13189   5.040 4.66e-07 ***
## flux2:cooling2    -0.02873    0.11955  -0.240   0.8101
## flux2:temp2       -0.30910    0.12274  -2.518   0.0118 *
## cooling2:temp2     1.70585    0.12465  13.685  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1450.52  on 47  degrees of freedom
## Residual deviance:  626.93  on 37  degrees of freedom
## AIC: 879.67
##
## Number of Fisher Scoring iterations: 5
```

```
#Reduced
model1 <- glm(numDefects~prebake*flux+prebake*cooling
              +prebake*temp+flux*temp+cooling*temp,
              family = "poisson", data=q1.data)
summary(model1)
```

```
##
## Call:
## glm(formula = numDefects ~ prebake * flux + prebake * cooling +
##     prebake * temp + flux * temp + cooling * temp, family = "poisson",
##     data = q1.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.7921  -2.6541  -0.2946   1.3936  13.5042
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        3.40957    0.08598  39.653  < 2e-16 ***
## prebake2           1.33040    0.09625  13.823  < 2e-16 ***
## flux2             -0.09016    0.09461  -0.953   0.3406
## cooling2           0.07550    0.09794   0.771   0.4408
## temp2             -1.86838    0.14883 -12.554  < 2e-16 ***
## prebake2:flux2    -0.51723    0.11047  -4.682 2.84e-06 ***
## prebake2:cooling2 -1.40241    0.12314 -11.389  < 2e-16 ***
## prebake2:temp2     0.66403    0.13185   5.036 4.75e-07 ***
## flux2:temp2       -0.31964    0.11465  -2.788   0.0053 **
## cooling2:temp2     1.70784    0.12439  13.730  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1450.52  on 47  degrees of freedom
## Residual deviance:  626.99  on 38  degrees of freedom
## AIC: 877.73
##
## Number of Fisher Scoring iterations: 5
```

```
anova(model1,Fullmodel, test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: numDefects ~ prebake * flux + prebake * cooling + prebake * temp +
##     flux * temp + cooling * temp
## Model 2: numDefects ~ (prebake + flux + cooling + temp)^2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        38     626.99
## 2        37     626.93  1 0.057773   0.8101
```

The model only need the interaction between prebake and flux, prebake and cooling time, prebake and temp, flux and temp and cooling time and temp as remove anymore variable from model would make Full model become significant under LRT test. .

```
phihat <- sum(residuals(model1,type="pearson")^2)/(38)
phihat
```

```
## [1] 19.65839
```

Since $\hat{\phi}$=19.65 and is significant greater than 1, hence we can conclude that hour model has a dispersion problem. Thus we will now model our model using quasipoisson.

```
Fullmodel <- glm(numDefects~ (prebake+flux+cooling+temp)^2, family = "quasipoisson", data=q1.data)
summary(Fullmodel)
```

```
##
## Call:
## glm(formula = numDefects ~ (prebake + flux + cooling + temp)^2,
##     family = "quasipoisson", data = q1.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.8230  -2.6586  -0.3118   1.4014  13.4735
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.40257    0.40862   8.327 5.25e-10 ***
## prebake2             1.33522    0.44236   3.018  0.00458 **
## flux2               -0.07552    0.50536  -0.149  0.88203
## cooling2             0.08918    0.50882   0.175  0.86182
## temp2               -1.87164    0.67198  -2.785  0.00838 **
## prebake2:flux2      -0.52572    0.52097  -1.009  0.31948
## prebake2:cooling2 -1.40598    0.55715  -2.524  0.01604 *
## prebake2:temp2       0.66473    0.59232   1.122  0.26898
## flux2:cooling2      -0.02873    0.53690  -0.054  0.95761
## flux2:temp2         -0.30910    0.55119  -0.561  0.57833
## cooling2:temp2       1.70585    0.55977   3.047  0.00424 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 20.16775)
##
##     Null deviance: 1450.52  on 47  degrees of freedom
## Residual deviance:  626.93  on 37  degrees of freedom
## AIC: NA
```

```
##
## Number of Fisher Scoring iterations: 5
```

```
model1 <- glm(numDefects~prebake+flux+cooling*temp,
              family = "quasipoisson", data=q1.data)
summary(model1)
```

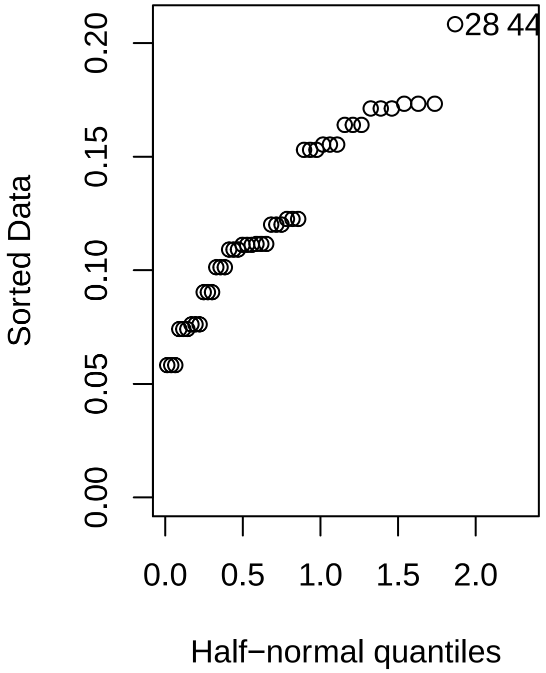
```
##
## Call:
## glm(formula = numDefects ~ prebake + flux + cooling * temp, family = "quasipoisson",
##     data = q1.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -9.5447  -3.0950  -0.7472   2.1584  13.8271
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.9097     0.2550  15.331  < 2e-16 ***
## prebake2          0.6729     0.2474   2.719 0.009472 **
## flux2            -0.5288     0.2423  -2.182 0.034727 *
## cooling2         -0.7638     0.3078  -2.481 0.017186 *
## temp2            -1.4388     0.3963  -3.630 0.000763 ***
## cooling2:temp2    1.5008     0.5315   2.824 0.007226 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 21.20337)
##
##     Null deviance: 1450.52  on 47  degrees of freedom
## Residual deviance:  799.07  on 42  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

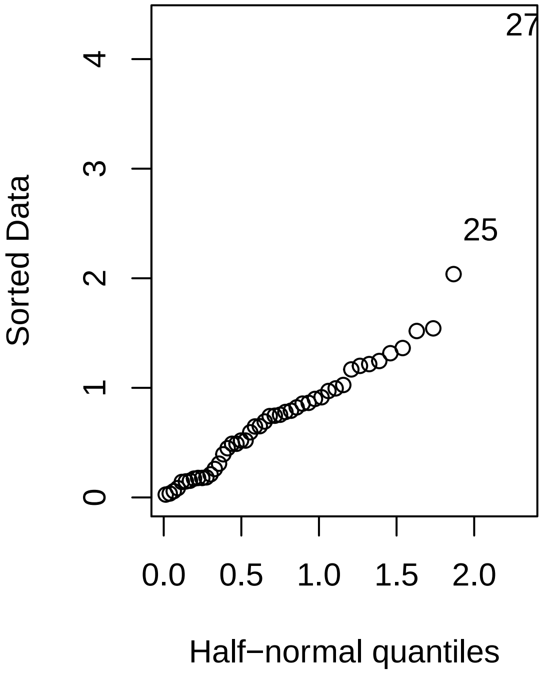
```
anova(model1,Fullmodel, test="LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: numDefects ~ prebake + flux + cooling * temp
## Model 2: numDefects ~ (prebake + flux + cooling + temp)^2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        42     799.07
## 2        37     626.93  5   172.14   0.1291
```

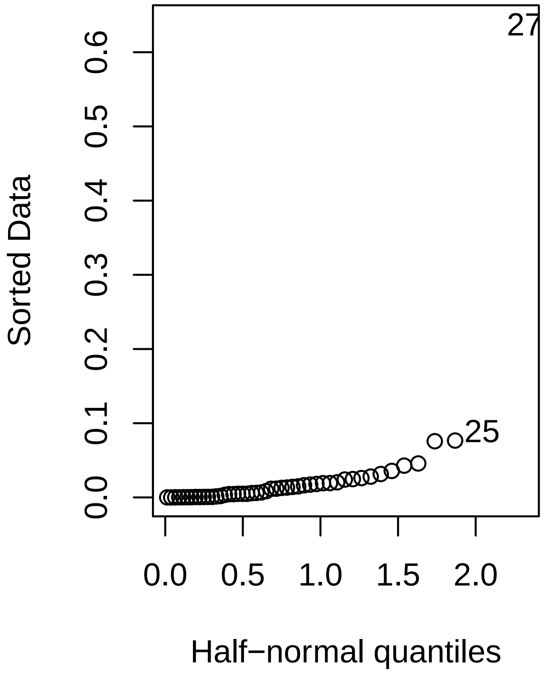
After taking into consideration for over dispersion and removing one variable at a time from the Fullmodel then do LRT test we get the final model that has significantly less and differences if we were not to take dispersion into consideration as the new model contain prebake, flux that was not in the orginal relevant model.

```
library(faraway)
par(mfrow=c(1,2))
halfnorm(influence(model1)$hat)
halfnorm(rstudent(model1))
```





```
halfnorm(cooks.distance(model1))
```



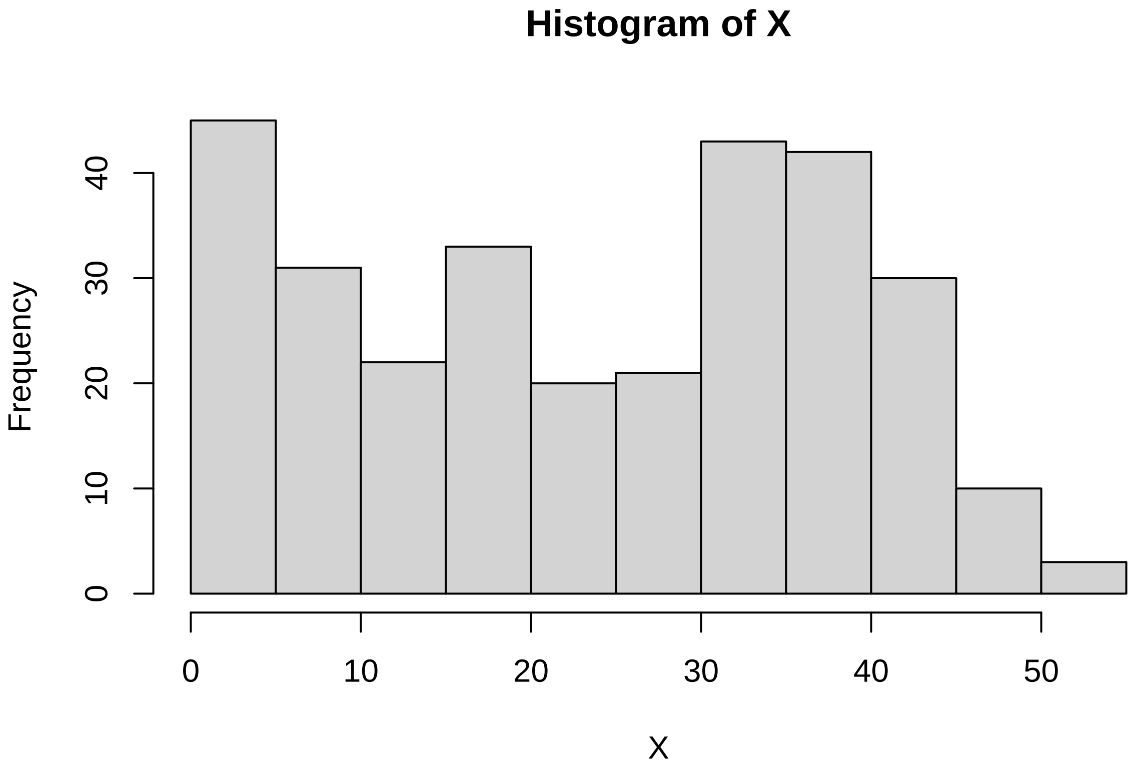
Looking at the leverage, we can see that 28 and 24 has moderately high leverage indicating that it is a points that has moderately potential impact on our data. However, Jackknife's residual and Cook's distance indicate that 25 particularly 27 are a points of outlier and highly influential. Thus, it is worth considering running the data again following the same procedure. Beside that our model is fine.

Question 2:

```
#Pre-question
X = scan(file="assignment2_prob2_2021.txt",what=double())
length(X)
```

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

```
hist(X)
```



```
str(X)
```

```
##  num [1:300] 6 6 34 18 35 45 15 4 4 22 ...
```

a. Derive the expectation of the complete log-likelihood

Let $n = 300$, $\theta = (\pi_1, \pi_2, \lambda_1, \lambda_2, \lambda_3)$, $I(Z_i = k_i)$ be the indicator function

$$\begin{aligned} P(X, Z|\theta) &= P(X_1, \ldots, X_n, Z_1, \ldots, Z_n|\theta) \\ &= \prod_{i=1}^{n} P(X_i|Z_i, \theta) P(Z_i|\theta) \\ &= \prod_{i=1}^{n} \prod_{k=1}^{3} [P(X_i|Z_i = k, \theta) P(Z_i = k|\theta)]^{I(Z_i = k)} \end{aligned}$$

$$\log(P(X_1, \ldots, X_n, Z_1, \ldots, Z_n|\theta)) = \sum_{i=1}^{n} \sum_{k=1}^{3} [I(Z_i = k)(\log(P(X_i|Z_i = k, \theta)) + \log(P(Z_i = k|\theta)))]$$

Since $X_i|Z_i \sim \text{Poisson}(\lambda_i) \implies P(X_i = x_i|Z_i = k, \theta) = \frac{\lambda_k^{x_i} e^{-\lambda_k}}{x_i!} \implies \log(P(X_i = x_i|Z_i = k, \theta)) = X_i \log(\lambda_k) - \lambda_k - \log(X_i!)$

$$\begin{aligned} Q(\theta, \theta^0) &= E_{Z|X, \theta^0}[P(X, Z|\theta)] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{3} P(Z_i = k|X_i, \theta^0) \left[\log(P(X_i|Z_i = k, \theta)) + \log(P(Z_i = k|\theta))\right] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{3} P(Z_i = k|X_i, \theta^0) \left[X_i \log(\lambda_k) - \lambda_k - \log(X_i!) + \log(\pi_k)\right] \end{aligned}$$

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

(b). Derive E-Step of the EM algorithm

Let the initial value $\theta^0 = (\pi_1^0, \pi_2^0, \lambda_1^0, \lambda_2^0, \lambda_3^0)$.

$$\begin{aligned} P(Z_i = j|X_i, \theta^0) &= \frac{P(Z_i = j, X_i = \theta^0)}{P(X_i|\theta^0)} \\ &= \frac{P(X_i|Z_i = j, \theta^0) P(Z_i = j|\theta^0)}{\sum_{k=1}^{3} [P(X_i|Z_i = k, \theta^0) P(Z_i = k|\theta^0)]} \\ &= \frac{\left(\frac{(\lambda_j^0)^{X_i} e^{-\lambda_j^0}}{x_i!}\right) \pi_j^0}{\sum_{k=1}^{3} \left(\frac{(\lambda_k^0)^{X_i} e^{-\lambda_k^0}}{x_i!}\right) \pi_k^0} \\ &= \frac{(\lambda_j^0)^{X_i} e^{-\lambda_j^0} \pi_j^0}{\sum_{k=1}^{3} \left((\lambda_k^0)^{X_i} e^{\lambda_k^0}\right) \pi_k^0} \end{aligned}$$

Where $j \in \{1, 2, 3\}$.

(c). Derive M-Step of the EM algorithm

Since

$$\begin{aligned}\frac{dQ(\theta,\theta^0)}{d\pi_1} &= \sum_{i=1}^{n}\left(\frac{P(Z_i=1|X_i,\theta^0)}{\pi_1} - \frac{P(Z_i=3|X_i,\theta^0)}{1-\pi_2-\pi_1}\right) = 0\\ \implies \pi_3\sum_{i=1}^{n}P(Z_i=1|X_i,\theta^0) &= \pi_1\sum_{i=1}^{n}P(Z_i=3|X_i,\theta^0) \qquad (1)\end{aligned}$$

$$\begin{aligned}\frac{dQ(\theta,\theta^0)}{d\pi_2} &= \sum_{i=1}^{n}\left(\frac{P(Z_i=2|X_i,\theta^0)}{\pi_2} - \frac{P(Z_i=3|X_i,\theta^0)}{1-\pi_2-\pi_1}\right) = 0\\ \implies \pi_3\sum_{i=1}^{n}P(Z_i=2|X_i,\theta^0) &= \pi_2\sum_{i=1}^{n}P(Z_i=3|X_i,\theta^0) \qquad (2)\end{aligned}$$

Sub (1) and (2), we get:

$$\begin{aligned}\hat{\pi}_3\sum_{i=1}^{n}\sum_{k=1}^{2}P(Z_i=k|X_i,\theta^0) &= (1-\pi_3)\sum_{i=1}^{n}P(Z_i=3|X_i,\theta^0)\\ \implies \pi_3\sum_{i=1}^{n}\sum_{k=1}^{3}P(Z_i=k|X_i,\theta^0) &= \sum_{i=1}^{n}P(Z_i=3|X_i,\theta^0)\\ \implies \pi_3 = \frac{1}{n}\sum_{i=1}^{n}P(Z_i=3|X_i,\theta^0) \qquad (3)\end{aligned}$$

Sub (3) into (1) and (2) we get:

$$\begin{aligned}\hat{\pi}_1 &= \frac{1}{n}\sum_{i=1}^{n}P(Z_i=1|X_i,\theta^0)\\ \hat{\pi}_2 &= \frac{1}{n}\sum_{i=1}^{n}P(Z_i=2|X_i,\theta^0)\end{aligned}$$

For $\hat{\lambda}_k$, since

$$\begin{aligned}\frac{dQ(\theta,\theta^0)}{d\lambda_k} &= \sum_{i=1}^{n}P(Z_i=k|X_i,\theta^0)\left(\frac{X_i}{\lambda_k}-1\right) = 0\\ \implies \sum_{i=1}^{n}P(Z_i=k|X_i,\theta^0)\frac{X_i}{\lambda_k} &= \sum_{i=1}^{n}P(Z_i=k|X_i,\theta^0)\\ \implies \hat{\lambda}_k &= \frac{\sum_{i=1}^{n}\left[P(Z_i=k|X_i,\theta^0)X_i\right]}{\sum_{i=1}^{n}\left[P(Z_i=k|X_i,\theta^0)\right]}\end{aligned}$$

(d). Implementing EM Algorithm

```r
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-likehoods for each iteration
  log_liks = c()

  # compute incomplete log-likehoods using initial values of parameters.
  log_liks = c(log_liks, compute.log.lik(X, w.curr, lambda.curr)$ill)

  # 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-likehoods 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))
}
```

```r
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))
}
```

```
# Compute incomplete log-likehoods
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-likehoods
  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.curr[k])*w.curr[k]
  }

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

Running the EM algorithm with the two different intial value

	π_1	π_2	λ_1	λ_2	λ_3
1st Initial Values	0.3	0.3	3	20	30
2nd Initial Values	0.1	0.2	5	25	40

Running EM-alogrithm for 1st initial value

```
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)
print(paste("Estimate pi_i = (", round(EM1$w.curr[1],2), ",",
            round(EM1$w.curr[2],2), ",",
            round(EM1$w.curr[3],2), ")", sep=""))
```

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

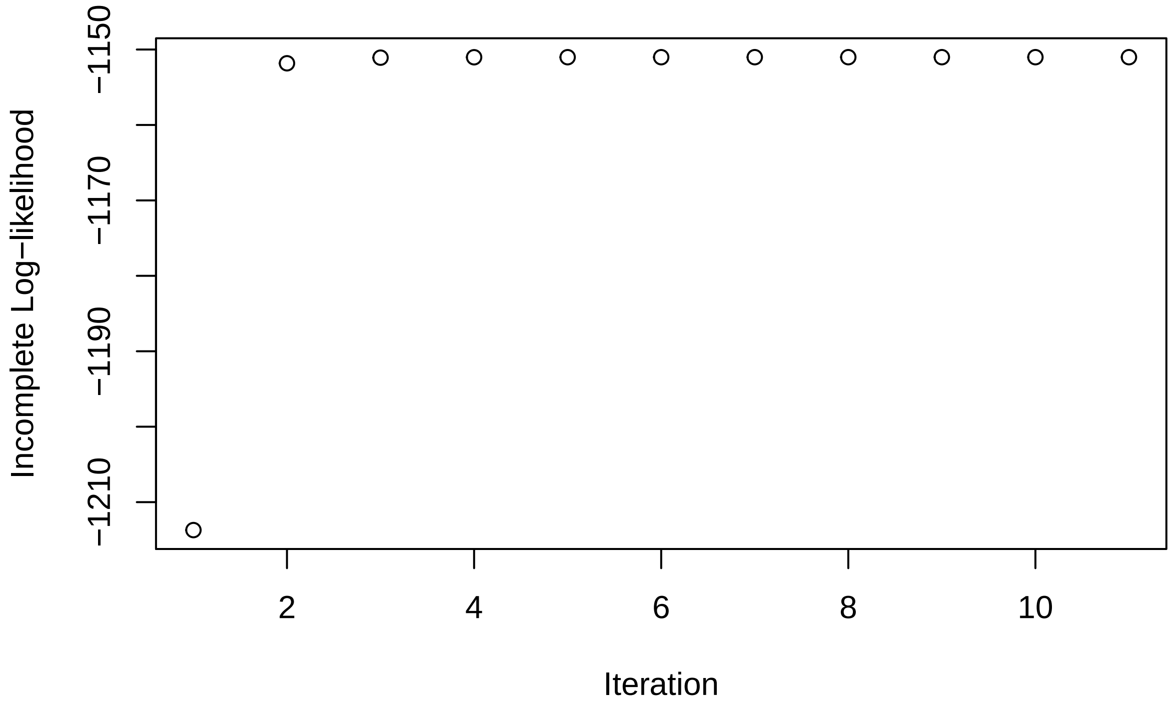
```
print(paste("Estimate lambda_i = (", round(EM1$lambda.curr[1],2), ",",
            round(EM1$lambda.curr[2],2), ",",
            round(EM1$lambda.curr[3],2), ")", sep=""))
```

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

```
print(paste("Log likelihood value = ",round(EM1$log_liks[length(EM1$log_liks)],2)))
```

```
## [1] "Log likelihood value =  -1151.01"
```

```
plot(EM1$log_liks, ylab = "Incomplete Log-likelihood", xlab = "Iteration")
```



Running EM-alogrithm for 2st initial value

```
EM2 <- mixture.EM(X, w.init=c(0.1,0.2,0.8), lambda.init = c(5,25,40) ,epsilon = 1e-5, max.iter = 100)
print(paste("Estimate pi_i = (", round(EM1$w.curr[1],2), ",",
            round(EM1$w.curr[2],2), ",",
            round(EM1$w.curr[3],2), ")", sep=""))
```

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

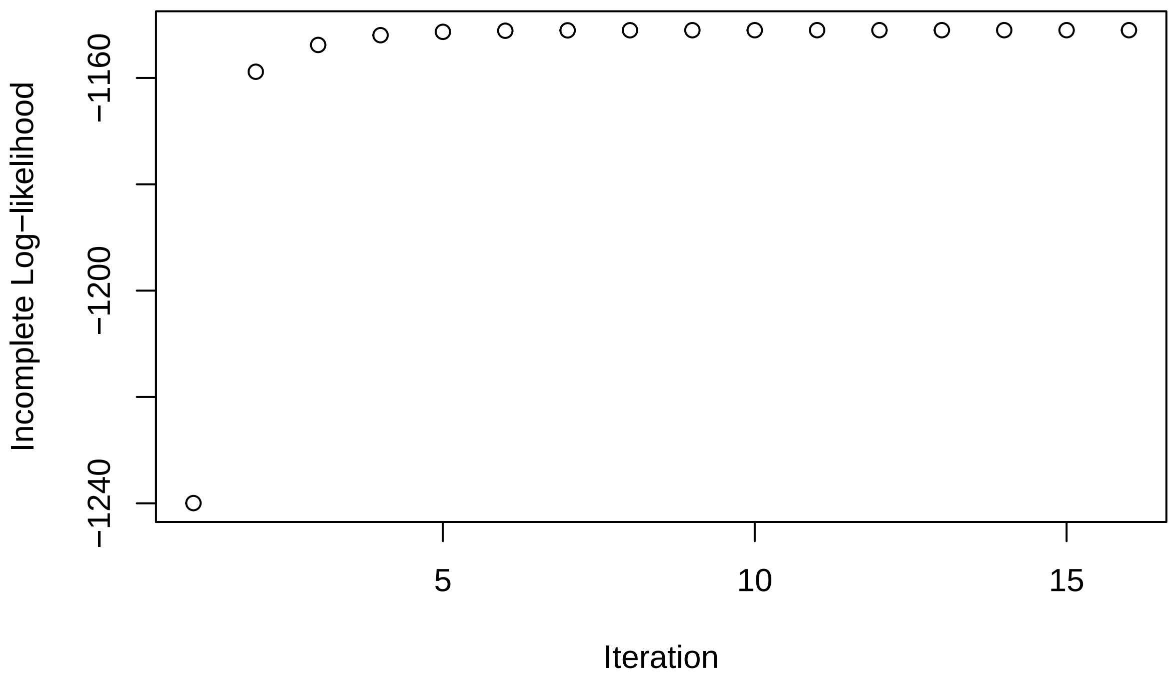
```
print(paste("Estimate lambda_i = (", round(EM1$lambda.curr[1],2), ",",
            round(EM2$lambda.curr[2],2), ",",
            round(EM2$lambda.curr[3],2), ")", sep=""))
```

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

```
print(paste("Log likelihood value = ",round(EM1$log_liks[length(EM1$log_liks)],2)))
```

```
## [1] "Log likelihood value =  -1151.01"
```

```
plot(EM2$log_liks, ylab = "Incomplete Log-likelihood", xlab = "Iteration")
```


Incomplete Log−likelihood
−1160
−1200
−1240
5
10
15
Iteration

Question 3:

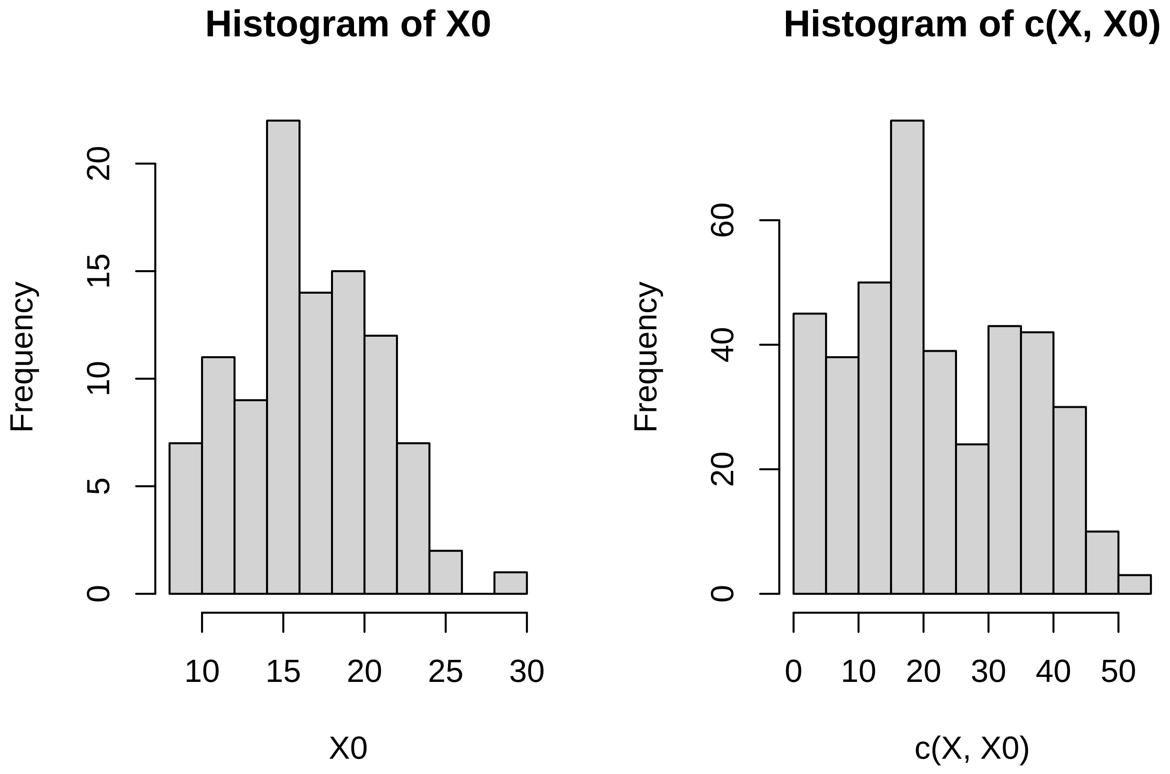
```r
#Pre Question

X0 = scan(file="assignment2_prob3_2021.txt", what=double())
length(X)
```

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

```r
length(X0)
```

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

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



```r
X.Both = c(X,X0)
```

a. Derive the expectation of the complete log-likelihood

$$P(X_1,\dots,X_{300},Z_1,\dots,Z_3|\theta)+P(X_{301},\dots,X_{400}) = \prod_{i=1}^{300} P(X_i|Z_i,\theta)P(Z_i|\theta) + \prod_{i=301}^{400} P(X_i)$$

$$\log(P(X_1,\dots,X_{300},Z_1,\dots,Z_3|\theta)+P(X_{301},\dots,X_{400})) =$$
$$\sum_{i=1}^{300}\sum_{k=1}^{3}[I(Z_i=k)(\log(P(X_i|Z_i=k,\theta))+\log(P(Z_i=k|\theta)))] + \sum_{i=301}^{400} P(X_i)$$

$$
\begin{aligned}
Q(\theta,\theta^0) &= E_{Z|X,\theta^0}[\log(P(X_1,\dots,X_{300},Z_1,\dots,Z_3|\theta)+P(X_{301},\dots,X_{400}))] \\
&= \sum_{i=1}^{300}\sum_{k=1}^{3} P(Z_i=k|X_i,\theta^0)\left[\log(P(X_i|Z_i=k,\theta))+\log(P(Z_i=k|\theta))\right]+\sum_{i=301}^{400} P(X_i) \\
&= \sum_{i=1}^{300}\sum_{k=1}^{3} P(Z_i=k|X_i,\theta^0)\left[X_i\log(\lambda_k)-\lambda_k-\log(X_i!)+\log(\pi_k)\right]+\sum_{i=301}^{400}\left[X_i\log(\lambda_2)-\lambda_2-\log(X_i!)\right]
\end{aligned}
$$

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

(b). Derive E-Step of the EM algorithm

E-Step

Let the initial value $\theta^0 = (\pi_1^0,\pi_2^0,\lambda_1^0,\lambda_2^0,\lambda_3^0)$.

$$
\begin{aligned}
P(Z_i=j|X_i,\theta^0) &= \frac{P(Z_i=j,X_i=\theta^0)}{P(X_i|\theta^0)} \\
&= \frac{P(X_i|Z_i=j,\theta^0)P(Z_i=j|\theta^0)}{\sum_{k=1}^{3}\left[P(X_i|Z_i=k,\theta^0)P(Z_i=k|\theta^0)\right]} \\
&= \frac{\left(\frac{(\lambda_j^0)^{X_i}e^{-\lambda_j^0}}{x_i!}\right)\pi_j^0}{\sum_{k=1}^{3}\left(\frac{(\lambda_k^0)^{X_i}e^{-\lambda_k^0}}{x_i!}\right)\pi_k^0} \\
&= \frac{(\lambda_j^0)^{X_i}e^{-\lambda_j^0}\pi_j^0}{\sum_{k=1}^{3}\left((\lambda_k^0)^{X_i}e^{\lambda_k^0}\right)\pi_k^0}
\end{aligned}
$$

Where $j\in\{1,2,3\}$.

M-Step

Since

$$
\frac{dQ(\theta,\theta^0)}{d\pi_1} = \sum_{i=1}^{300}\left(\frac{P(Z_i=1|X_i,\theta^0)}{\pi_1}-\frac{P(Z_i=3|X_i,\theta^0)}{1-\pi_2-\pi_1}\right)=0
$$

$$
\implies \pi_3\sum_{i=1}^{300}P(Z_i=1|X_i,\theta^0)=\pi_1\sum_{i=1}^{300}P(Z_i=3|X_i,\theta^0) \tag{1}
$$

$$
\frac{dQ(\theta,\theta^0)}{d\pi_2} = \sum_{i=1}^{300}\left(\frac{P(Z_i=2|X_i,\theta^0)}{\pi_2}-\frac{P(Z_i=3|X_i,\theta^0)}{1-\pi_2-\pi_1}\right)=0
$$

$$
\implies \pi_3\sum_{i=1}^{300}P(Z_i=2|X_i,\theta^0)=\pi_2\sum_{i=1}^{n}P(Z_i=3|X_i,\theta^0) \tag{2}
$$

Sub (1) and (2), we get:

$$\hat{\pi}_3 \sum_{i=1}^{300}\sum_{k=1}^{2} P(Z_i = k|X_i, \theta^0) = (1-\pi_3)\sum_{i=1}^{300} P(Z_i = 3|X_i, \theta^0)$$

$$\implies \pi_3 \sum_{i=1}^{300}\sum_{k=1}^{3} P(Z_i = k|X_i, \theta^0) = \sum_{i=1}^{300} P(Z_i = 3|X_i, \theta^0)$$

$$\implies \pi_3 = \frac{1}{300}\sum_{i=1}^{300} P(Z_i = 3|X_i, \theta^0) \tag{3}$$

Sub (3) into (1) and (2) we get:

$$\hat{\pi}_1 = \frac{1}{300}\sum_{i=1}^{300} P(Z_i = 1|X_i, \theta^0)$$

$$\hat{\pi}_2 = \frac{1}{300}\sum_{i=1}^{300} P(Z_i = 2|X_i, \theta^0)$$

For $\hat{\lambda}_k$, where $k \in \{1, 3\}$since

$$\frac{dQ(\theta, \theta^0)}{d\lambda_k} = \sum_{i=1}^{300} P(Z_i = k|X_i, \theta^0)\left(\frac{X_i}{\lambda_k} - 1\right) = 0$$

$$\implies \sum_{i=1}^{300} P(Z_i = k|X_i, \theta^0)\frac{X_i}{\lambda_k} = \sum_{i=1}^{300} P(Z_i = k|X_i, \theta^0)$$

$$\implies \hat{\lambda}_k = \frac{\sum_{i=1}^{300}\left[P(Z_i = k|X_i, \theta^0)X_i\right]}{\sum_{i=1}^{300}\left[P(Z_i = k|X_i, \theta^0)\right]}$$

For $\hat{\lambda}_2$

$$\frac{dQ(\theta, \theta^0)}{d\lambda_2} = \sum_{i=1}^{300} P(Z_i = k|X_i, \theta^0)\left(\frac{X_i}{\lambda_2} - 1\right) + \sum_{i=301}^{400}\left[\frac{X_i}{\lambda_2} - 1\right] = 0$$

$$\sum_{i=1}^{300}\left[P(Z_i = k|X_i, \theta^0)\frac{X_i}{\lambda_2}\right] + \sum_{i=301}^{400}\frac{X_i}{\lambda_2} = \sum_{i=1}^{300} P(Z_i = k|X_i, \theta^0) + 100$$

$$\implies \hat{\lambda}_2 = \frac{\sum_{i=1}^{300} P(Z_i = k|X_i, \theta^0)X_i + \sum_{i=301}^{400} X_i}{\sum_{i=1}^{300} P(Z_i = k|X_i, \theta^0) + 100}$$

(c). Implementing EM Algorithm

```r
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-likehoods for each iteration
  log_liks = c()

  # compute incomplete log-likehoods using initial values of parameters.
  log_liks = c(log_liks, compute.log.lik(X, w.curr, lambda.curr)$ill)

  # 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-likehoods 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
  lambda.new = c(0,0,0)
  # 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[1]  = sum(P_ik[1:300,1]*X[1:300])/sum(P_ik[1:300,1])
  lambda.new[3]  = sum(P_ik[1:300,3]*X[1:300])/sum(P_ik[1:300,3])

  lambda.new[2]= (sum(P_ik[1:300,2]*X[1:300])+sum(X[301:400]))/(sum(P_ik[1:300,2])+100)

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

```r
test=compute.log.lik(X.Both,w.curr = c(0.1,0.3,0.6),lambda.curr=c(5,10,10))

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

  # for each sample $X_i$, compute $P(X_i, Z_i=k) and P(X_i)$ for i from 1 to 300
  prob.x.z = compute.prob.x.z(X, w.curr, lambda.curr)$prob.x.z

  #Compute X_i for i from 301 to 400
  px = dpois(X[301:400], lambda.curr[2])
  # incomplete log-likehoods
  ill = sum(log(rowSums(prob.x.z)))+ sum(log(px))

  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=300, ncol= length(w.curr))
  for(k in seq_len(ncol(L))) {
    L[, k] = dpois(X[1:300],lambda.curr[k])*w.curr[k]
  }
  return(list(prob.x.z=L))
}
```

Running the EM algorithm with the two different initial value

	π_1	π_2	λ_1	λ_2	λ_3
1st Initial Values	0.3	0.3	3	20	30
2nd Initial Values	0.1	0.2	5	25	40

Running EM algorithm with 1st set of initial values

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

```
## [1] "Estimate pi_i = (0.25,0.24,0.51)"
```

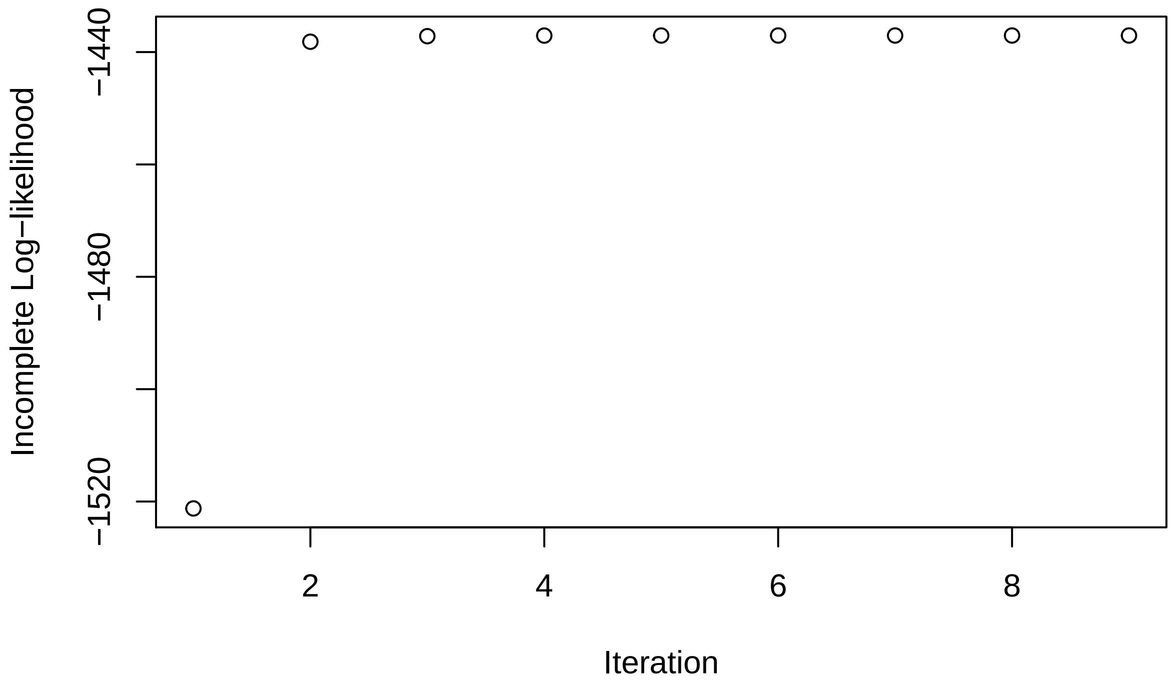
```r
print(paste("Estimate lambda_i = (", round(EM1$lambda.curr[1],2), ",",
            round(EM1$lambda.curr[2],2), ",",
            round(EM1$lambda.curr[3],2), ")", sep=""))
```

```
## [1] "Estimate lambda_i = (5.12,17.36,36.77)"
```

```r
print(paste("Log likelihood value = ",round(EM1$log_liks[length(EM1$log_liks)],2)))
```

```
## [1] "Log likelihood value =  -1437.05"
```

```r
plot(EM1$log_liks, ylab = "Incomplete Log-likelihood", xlab = "Iteration")
```



Running EM algorithm with 2nd set of initial values

```
EM2 <- mixture.EM(X.Both, w.init=c(0.1,0.2,0.8), lambda.init = c(5,25,40) ,epsilon = 1e-5, max.iter = 1
print(paste("Estimate pi_i = (", round(EM1$w.curr[1],2), ",",
            round(EM1$w.curr[2],2), ",",
            round(EM1$w.curr[3],2), ")", sep=""))
```

```
## [1] "Estimate pi_i = (0.25,0.24,0.51)"
```

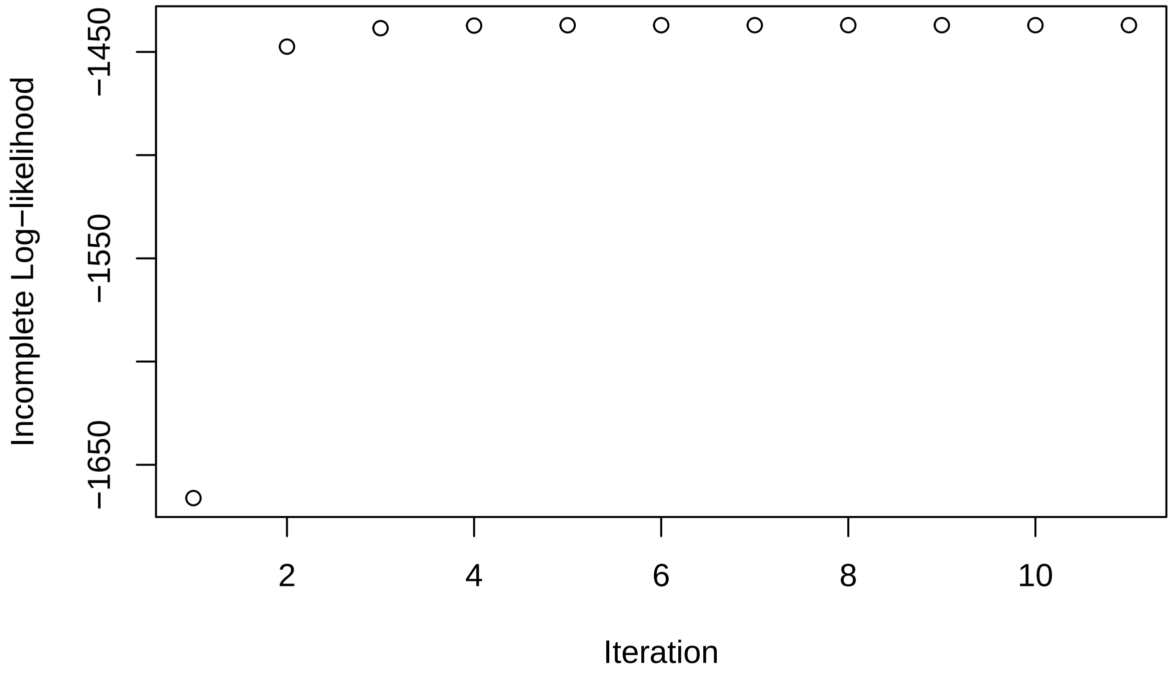
```
print(paste("Estimate lambda_i = (", round(EM1$lambda.curr[1],2), ",",
            round(EM2$lambda.curr[2],2), ",",
            round(EM2$lambda.curr[3],2), ")", sep=""))
```

```
## [1] "Estimate lambda_i = (5.12,17.36,36.76)"
```

```
print(paste("Log likelihood value = ",round(EM1$log_liks[length(EM1$log_liks)],2)))
```

```
## [1] "Log likelihood value =  -1437.05"
```

```
plot(EM2$log_liks, ylab = "Incomplete Log-likelihood", xlab = "Iteration")
```


−1450
−1550
−1650
Incomplete Log−likelihood
2
4
6
8
10
Iteration