MAST30027_Assignment2

Zi Ng (1085130)

Thursday 4.15 pm, Anubhav Kaphle

Question 1

Firstly we will read and inspect the data. There are 70 observations. We aim to determine which factors (duration, residence, education) and two-way interactions are related to the number of children per woman (fertility rate).

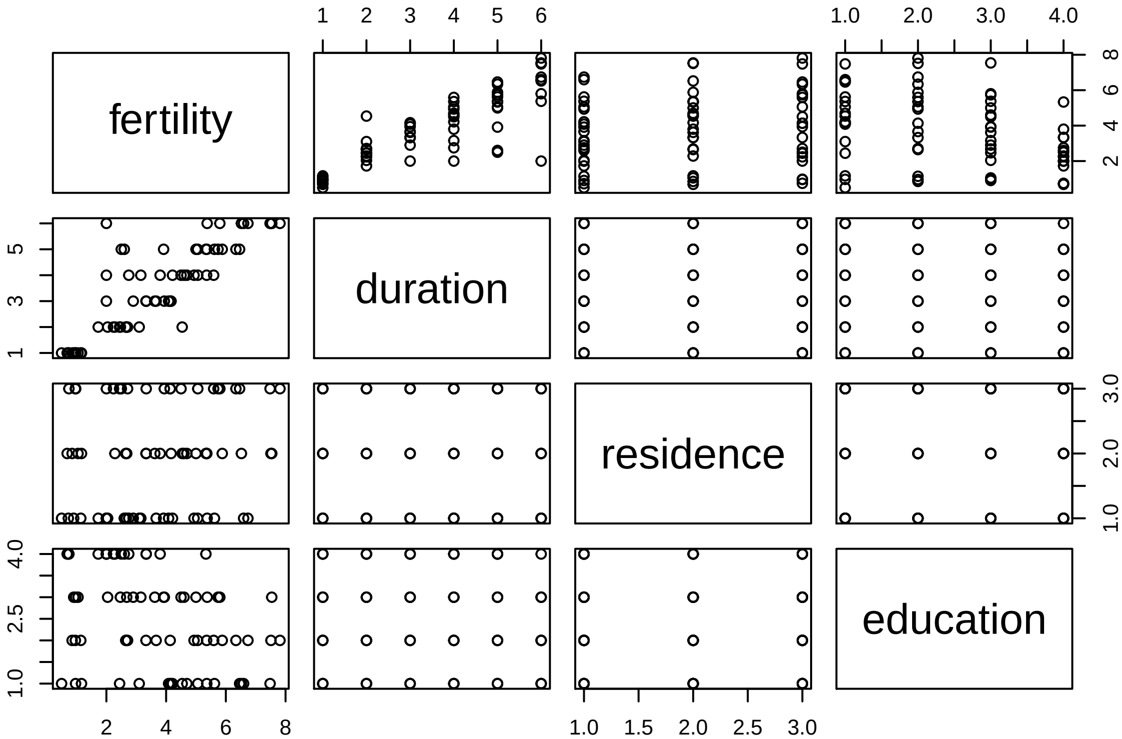
```
data <- read.table(file ="assignment2_prob1.txt", header=TRUE)
data$duration <- factor(data$duration,
                        levels=c("0-4","5-9","10-14","15-19","20-24","25-29"),
                        ordered=TRUE)
data$residence <- factor(data$residence, levels=c("Suva", "urban", "rural"))
data$education <- factor(data$education, levels=c("none", "lower", "upper", "sec+"))
data$fertility <- data$nChildren / data$nMother
str(data)
```

```
## 'data.frame':    70 obs. of  6 variables:
##  $ duration : Ord.factor w/ 6 levels "0-4"<"5-9"<"10-14"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ residence: Factor w/ 3 levels "Suva","urban",..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ education: Factor w/ 4 levels "none","lower",..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ nMother  : int  8 21 42 51 12 27 39 51 62 102 ...
##  $ nChildren: int  4 24 38 37 14 23 41 35 60 98 ...
##  $ fertility: num  0.5 1.143 0.905 0.725 1.167 ...
```

```
# ftable(xtabs(cbind(nChildren,nMother,fertility) ~
#                 duration + residence + education, data))
```

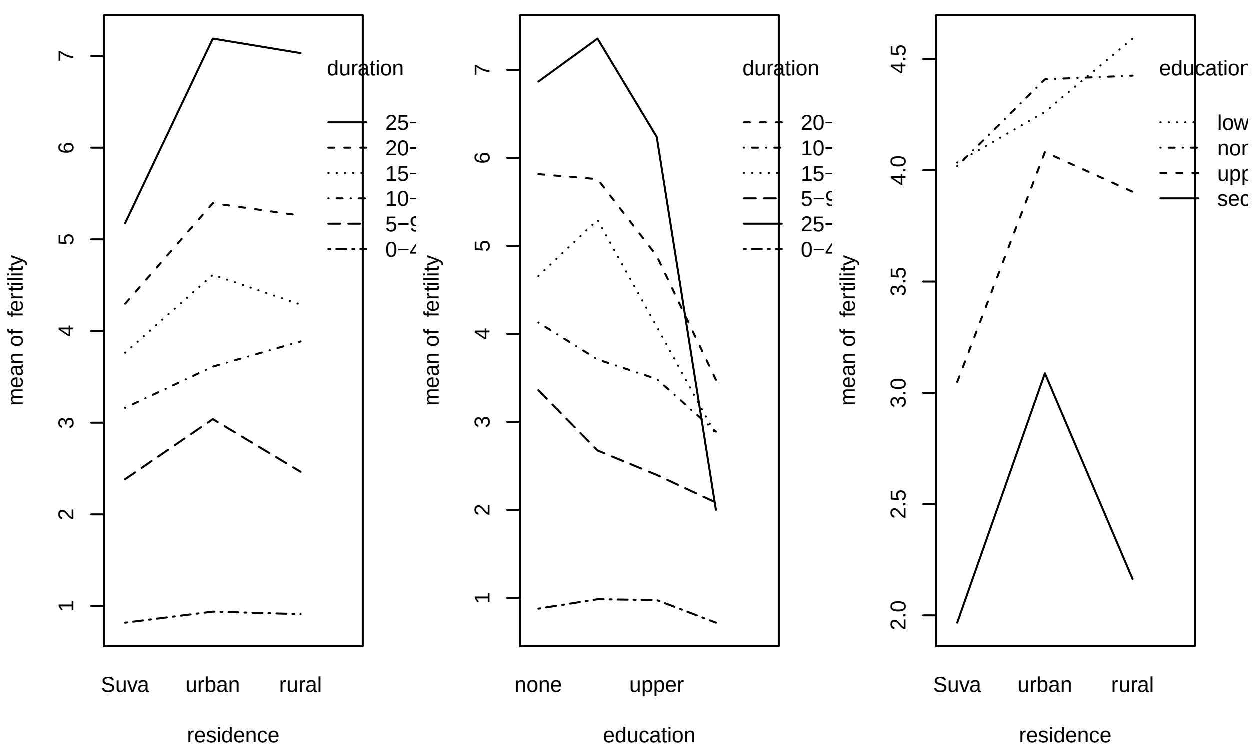
We can visualize the data with pair plots. Visually, we can roughly make out a relationships between fertility and duration, residence as well as education.

```
with(data, pairs(fertility ~ duration + residence + education))
```



We can also use interaction plots to see if there are two-way relationships related to the fertility rate. Since these slopes are not quite parallel, it appears that the two-way interactions might impact fertility rate.

```
par(mfrow=c(1,3))
with(data, interaction.plot(residence, duration, fertility))
with(data, interaction.plot(education, duration, fertility))
with(data, interaction.plot(residence, education, fertility))
```



Since the number of children a woman has is count data, it makes sense to fit a Poisson model. Since the number of children depends on the number of women, we can model the rate per unit in the form of a Poisson glm with log link.

$$\log(\lambda_i/t_i) = x_i^T\beta$$
$$\log(\lambda_i) = \log(t_i) + x_i^T\beta$$

We can fit the rate model using the glm command with offset to constrain the coefficient of $\log(t_i)$ to 1.

```r
model = glm(nChildren ~ offset(log(nMother)) + duration + residence + education +
              duration*residence + duration*education + education*residence,
            family = poisson, data = data)
summary(model)
```

```
##
## Call:
## glm(formula = nChildren ~ offset(log(nMother)) + duration + residence +
##     education + duration * residence + duration * education +
##     education * residence, family = poisson, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7572  -0.3222   0.0414   0.3298   2.8134
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.262560   0.054120  23.329  < 2e-16 ***
## duration.L                     1.322461   0.109693  12.056  < 2e-16 ***
## duration.Q                    -0.475204   0.099868  -4.758 1.95e-06 ***
## duration.C                     0.310979   0.090042   3.454 0.000553 ***
## duration^4                    -0.123519   0.082325  -1.500 0.133514
## duration^5                     0.003130   0.077310   0.040 0.967704
## residenceurban                 0.004121   0.066846   0.062 0.950846
## residencerural                 0.048692   0.054980   0.886 0.375822
## educationlower                -0.015048   0.064926  -0.232 0.816718
## educationupper                -0.284101   0.081056  -3.505 0.000457 ***
## educationsec+                 -0.665426   0.152905  -4.352 1.35e-05 ***
## duration.L:residenceurban      0.147030   0.109403   1.344 0.178971
## duration.Q:residenceurban     -0.101429   0.096908  -1.047 0.295260
## duration.C:residenceurban      0.049790   0.090883   0.548 0.583798
## duration^4:residenceurban     -0.059840   0.086231  -0.694 0.487714
## duration^5:residenceurban      0.084682   0.082494   1.027 0.304646
## duration.L:residencerural      0.232160   0.094578   2.455 0.014100 *
## duration.Q:residencerural     -0.112487   0.084271  -1.335 0.181937
## duration.C:residencerural     -0.038218   0.078852  -0.485 0.627904
## duration^4:residencerural      0.020052   0.075060   0.267 0.789356
## duration^5:residencerural     -0.037891   0.072443  -0.523 0.600943
## duration.L:educationlower      0.063735   0.093908   0.679 0.497332
## duration.Q:educationlower      0.020680   0.087169   0.237 0.812474
## duration.C:educationlower     -0.048863   0.076118  -0.642 0.520921
## duration^4:educationlower      0.074274   0.065747   1.130 0.258605
## duration^5:educationlower      0.091940   0.057318   1.604 0.108704
## duration.L:educationupper     -0.066616   0.102487  -0.650 0.515696
## duration.Q:educationupper      0.103240   0.096634   1.068 0.285355
## duration.C:educationupper     -0.033646   0.086988  -0.387 0.698916
```

```
## duration^4:educationupper      0.080111   0.078622   1.019 0.308232
## duration^5:educationupper     -0.025175   0.073140  -0.344 0.730700
## duration.L:educationsec+      -0.481404   0.444798  -1.082 0.279120
## duration.Q:educationsec+      -0.310273   0.410113  -0.757 0.449317
## duration.C:educationsec+      -0.161468   0.299016  -0.540 0.589197
## duration^4:educationsec+      -0.042075   0.198420  -0.212 0.832068
## duration^5:educationsec+      -0.043235   0.157360  -0.275 0.783506
## residenceurban:educationlower  0.014568   0.078828   0.185 0.853377
## residencerural:educationlower  0.036396   0.066889   0.544 0.586350
## residenceurban:educationupper  0.258773   0.099801   2.593 0.009517 **
## residencerural:educationupper  0.201583   0.089264   2.258 0.023928 *
## residenceurban:educationsec+   0.318915   0.144496   2.207 0.027308 *
## residencerural:educationsec+   0.244863   0.147421   1.661 0.096717 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3731.852  on 69  degrees of freedom
## Residual deviance:   30.856  on 28  degrees of freedom
## AIC: 544.33
##
## Number of Fisher Scoring iterations: 4
```

For model selection, we will use the step function based on the AIC.

```
model.step = step(model, scope = ~.)
```

```
## Start:  AIC=544.33
## nChildren ~ offset(log(nMother)) + duration + residence + education +
##     duration * residence + duration * education + education *
##     residence
##
##                        Df Deviance    AIC
## - duration:education  15   44.311 527.79
## - duration:residence  10   44.523 538.00
## - residence:education  6   42.652 544.13
## <none>                     30.856 544.33
##
## Step:  AIC=527.79
## nChildren ~ duration + residence + education + duration:residence +
##     residence:education + offset(log(nMother))
##
##                        Df Deviance    AIC
## - duration:residence  10   59.921 523.40
## <none>                     44.311 527.79
## - residence:education  6   57.135 528.61
## + duration:education  15   30.856 544.33
##
## Step:  AIC=523.4
## nChildren ~ duration + residence + education + residence:education +
##     offset(log(nMother))
##
##                        Df Deviance     AIC
## - residence:education  6    70.67  522.14
```

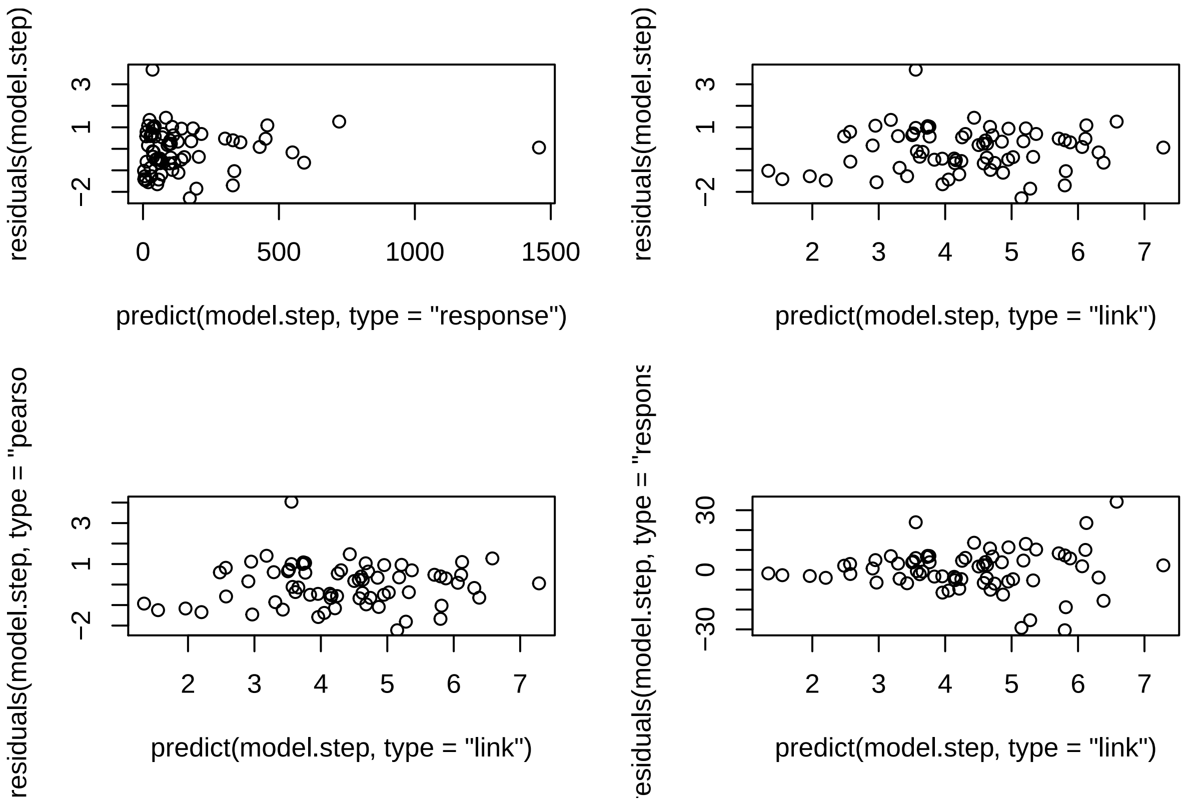
```
## <none>                        59.92  523.40
## + duration:residence  10    44.31  527.79
## + duration:education  15    44.52  538.00
## - duration             5  2625.89 3079.36
##
## Step:  AIC=522.14
## nChildren ~ duration + residence + education + offset(log(nMother))
##
##                       Df Deviance     AIC
## <none>                        70.67  522.14
## + residence:education  6    59.92  523.40
## + duration:residence  10    57.13  528.61
## + duration:education  15    54.80  536.28
## - residence            2   100.19  547.67
## - education            3   120.68  566.16
## - duration             5  2646.49 3087.97
```

```r
summary(model.step)
```

```
##
## Call:
## glm(formula = nChildren ~ duration + residence + education +
##     offset(log(nMother)), family = poisson, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2960  -0.6641   0.0725   0.6336   3.6782
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.17314    0.03054  38.415  < 2e-16 ***
## duration.L      1.49288    0.03387  44.082  < 2e-16 ***
## duration.Q     -0.52726    0.03026 -17.424  < 2e-16 ***
## duration.C      0.25258    0.02776   9.098  < 2e-16 ***
## duration^4     -0.07613    0.02570  -2.962 0.003059 **
## duration^5      0.03025    0.02402   1.259 0.207880
## residenceurban  0.11242    0.03250   3.459 0.000541 ***
## residencerural  0.15166    0.02833   5.353 8.63e-08 ***
## educationlower  0.02297    0.02266   1.014 0.310597
## educationupper -0.10127    0.03099  -3.268 0.001082 **
## educationsec+  -0.31015    0.05521  -5.618 1.94e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3731.852  on 69  degrees of freedom
## Residual deviance:   70.665  on 59  degrees of freedom
## AIC: 522.14
##
## Number of Fisher Scoring iterations: 4
```

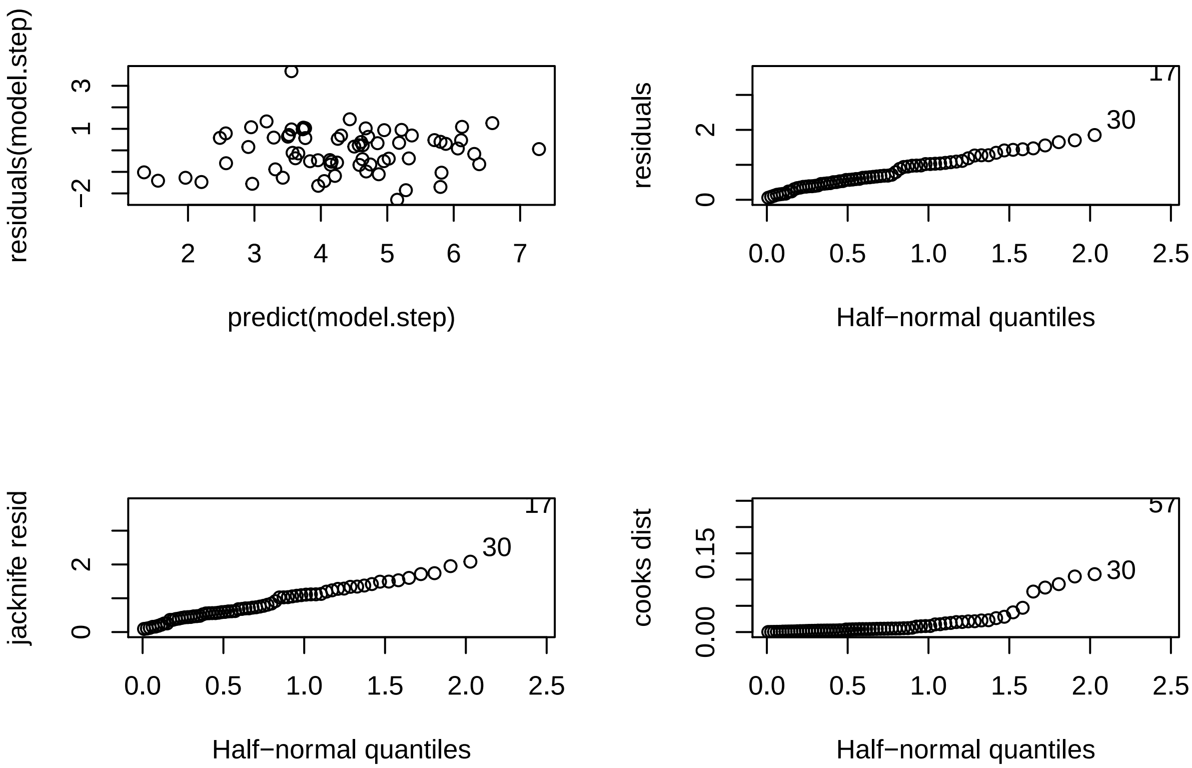
Next, we will perform diagnostics.

```r
par(mfrow=c(2,2))
# data points look cramped
plot(residuals(model.step) ~ predict(model.step,type="response"))
# data points look OK
plot(residuals(model.step) ~ predict(model.step, type="link"))
# data points look OK
plot(residuals(model.step, type="pearson") ~ predict(model.step, type="link"))
# appear to be heteroskedastic
plot(residuals(model.step, type="response") ~ predict(model.step, type="link"))
```



We will examine the data points to find outliers and points with significant impact. Based on the graphs, it looks like observations 17 and 57 are potential outliers. Observation 17 looks like it might still belong on the smooth curve, however observation 57 clearly deviates from the curve. We will refit a model that excludes observation 57 to see if it changes the model.

```r
par(mfrow=c(2,2))
plot(predict(model.step), residuals(model.step))
halfnorm(residuals(model.step), ylab="residuals")
halfnorm(rstudent(model.step), ylab="jacknife resid")
halfnorm(cooks.distance(model.step), ylab="cooks dist")
```




```r
model.subset = glm(nChildren ~ offset(log(nMother)) + duration + residence + education +
                     duration*residence + duration*education + education*residence,
                   family = poisson, data = data, subset = c(-57))
model.subset.step = step(model.subset, scope=~., trace=0)
summary(model.subset.step)
```

```
##
## Call:
## glm(formula = nChildren ~ duration + residence + education +
##     offset(log(nMother)), family = poisson, data = data, subset = c(-57))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3884  -0.6241   0.0929   0.6219   3.7210
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.16426    0.03101  37.540  < 2e-16 ***
## duration.L      1.48577    0.03415  43.502  < 2e-16 ***
## duration.Q     -0.52279    0.03037 -17.213  < 2e-16 ***
## duration.C      0.26981    0.02953   9.138  < 2e-16 ***
## duration^4     -0.05706    0.02800  -2.037  0.04160 *
## duration^5      0.04095    0.02481   1.650  0.09885 .
## residenceurban  0.11111    0.03250   3.418  0.00063 ***
## residencerural  0.14451    0.02864   5.046 4.52e-07 ***
## educationlower  0.03591    0.02388   1.504  0.13255
## educationupper -0.08855    0.03188  -2.778  0.00547 **
## educationsec+  -0.29975    0.05558  -5.394 6.91e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3568.432  on 68  degrees of freedom
## Residual deviance:   67.621  on 58  degrees of freedom
## AIC: 510.63
##
## Number of Fisher Scoring iterations: 4
```

Apparently, removing observation 57 does not have a large impact on the model. Hence we will revert to the original model without subsetting the data.

Checking the scaled deviance, we see verify that the model is appropriate.

```r
anova(model.step)
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: nChildren
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                         69     3731.9
## duration   5   3565.8        64      166.1
## residence  2     45.4        62      120.7
## education  3     50.0        59       70.7
```

```r
pchisq(deviance(model.step), 59, lower.tail = FALSE)
```

```
## [1] 0.1421387
```

Finally, we can check for overdispersion by estimating phi. Since $\hat{\phi} \approx 1$, there is no overdispersion.

```r
(phihat <- sum(residuals(model.step, type="pearson")^2) / 59)
```

```
## [1] 1.212432
```

Hence, we can model the number children born to married women of the Indian race using a Poisson model with variables including the marriage duration of mothers, the residence of families in each group and the education of the mothers in each group.

Question 2

2a Expectation of the complete log-likelihood.

The likelihood of observing $X = (X_1, ..., X_{300})$ and $Z = (Z_1, ..., Z_{300})$ given $\theta = (\pi_1, \pi_2, p_1, p_2, p_3)$ is calculated, noting that $\pi_3 = 1 - \pi_1 - \pi_2$.

$$\begin{aligned}
&\Pr(X_1, ..., X_{300}, Z_1, ..., Z_{300}|\theta) \\
=&\Pi_{i=1}^{300} \Pr(X_i|Z_i, \theta) \cdot \Pr(Z_i|\theta) \\
=&\Pi_{i=1}^{300} \Pi_{j=1}^{3} [\Pr(X_i|Z_i = j, \theta) \cdot \Pr(Z_i = j|\theta)]^{I(Z_i=j)}
\end{aligned}$$

The log-likelihood is then calculated.

$$\begin{aligned}
&\log \Pr(X_1, ..., X_{300}, Z_1, ..., Z_{300}|\theta) \\
=&\sum_{i=1}^{300} \sum_{j=1}^{3} I(Z_i = j)(\log(\Pr(X_i|Z_i = j, \theta)) + \log(\Pr(Z_i = j|\theta))) \\
=&\sum_{i=1}^{300} \sum_{j=1}^{3} I(Z_i = j)(\log \binom{20}{x_i} \cdot p_j^{x_i} \cdot (1 - p_j)^{20 - x_i} + \log \pi_j) \\
=&\sum_{i=1}^{300} \sum_{j=1}^{3} I(Z_i = j)(x_i \cdot \log p_j + (20 - x_i) \log(1 - p_j) + \log \binom{20}{x_i} + \log \pi_j)
\end{aligned}$$

Finally, we can take the expectation to derive the complete log-likelihood.

$$\begin{aligned}
&Q(\theta, \theta^0) \\
=&E_{Z|X,\theta^0}[\log(\Pr(X, Z|\theta))] \\
=&\sum_{i=1}^{300} [\sum_{j=1}^{3} \Pr(Z_i = j|X, \theta^0)(x_i \log p_j + (20 - x_i) \log(1 - p_j) + \log \binom{20}{x_i} + \log \pi_j)]
\end{aligned}$$

2b The E-step of the EM algorithm.

Using $\theta^0 = (\pi_1^0, \pi_2^0, p_1^0, p_2^0, p_3^0)$, we can derive the posterior distribution of the latent variables, where $\pi_3 = 1 - \pi_1 - \pi_2$.

$$
\begin{aligned}
&\Pr(Z_i = j | X, \theta^0) \\
=&\frac{\Pr(Z_i = j, X_i | \theta^0)}{\Pr(X_i | \theta^0)} \\
=&\frac{\Pr(X_i | Z_i = j, \theta^0) \Pr(Z_i = j | \theta^0)}{\sum_{k=1}^3 \Pr(X_i | Z_i = k, \theta^0) \Pr(Z_i = k | \theta^0)} \\
=&\frac{\binom{20}{x_i} p_j^{x_i} (1 - p_j)^{20 - x_i} \pi_j}{\sum_{k=1}^3 \binom{20}{x_i} p_k^{x_i} (1 - p_k)^{20 - x_i} \pi_k}
\end{aligned}
$$

2c The M-step of the EM algorithm.

Firstly, derive the new estimate of π_j, for $j = 1, 2$. Based on the working shown during the lectures, the estimate of pi_j can be derived.

$$\frac{\partial Q(\theta, \theta^0)}{\partial \pi_j} = 0$$

$$\sum_{i=1}^{300}[\frac{\Pr(Z_i = j|X, \theta^0)}{\pi_j} - \frac{\Pr(Z_i = k|Z, \theta^0)}{1 - \pi_1 - \pi_2}] = 0$$

$$\hat{\pi_j} = \frac{1}{300}P(Z_i = j|X, \theta^0)$$

Secondly, derive the new estimate of p_j, for $j = 1, 2, 3$.

$$\frac{\partial Q(\theta, \theta^0)}{\partial p_j} = 0$$

$$\sum_{i=1}^{300} \Pr(Z_i = j|X, \theta^0)(\frac{x_i}{p_j} - \frac{20 - x_i}{1 - p_j}) = 0$$

$$\sum_{i=1}^{300} \Pr(Z_i = j|X, \theta^0)(x_i(1 - p_j) - (20 - x_i)p_j) = 0$$

$$\sum_{i=1}^{300} \Pr(Z_i = j|Z, \theta^0)x_i - 20 \cdot p_j \sum_{i=1}^{300} \Pr(Z_i = j|Z, \theta^0) = 0$$

$$\hat{p_j} = \frac{\sum_{i=1}^{300} \Pr(Z_i = j|X, \theta^0)x_i}{20 \sum_{i=1}^{300} \Pr(Z_i = j|X, \theta)}$$

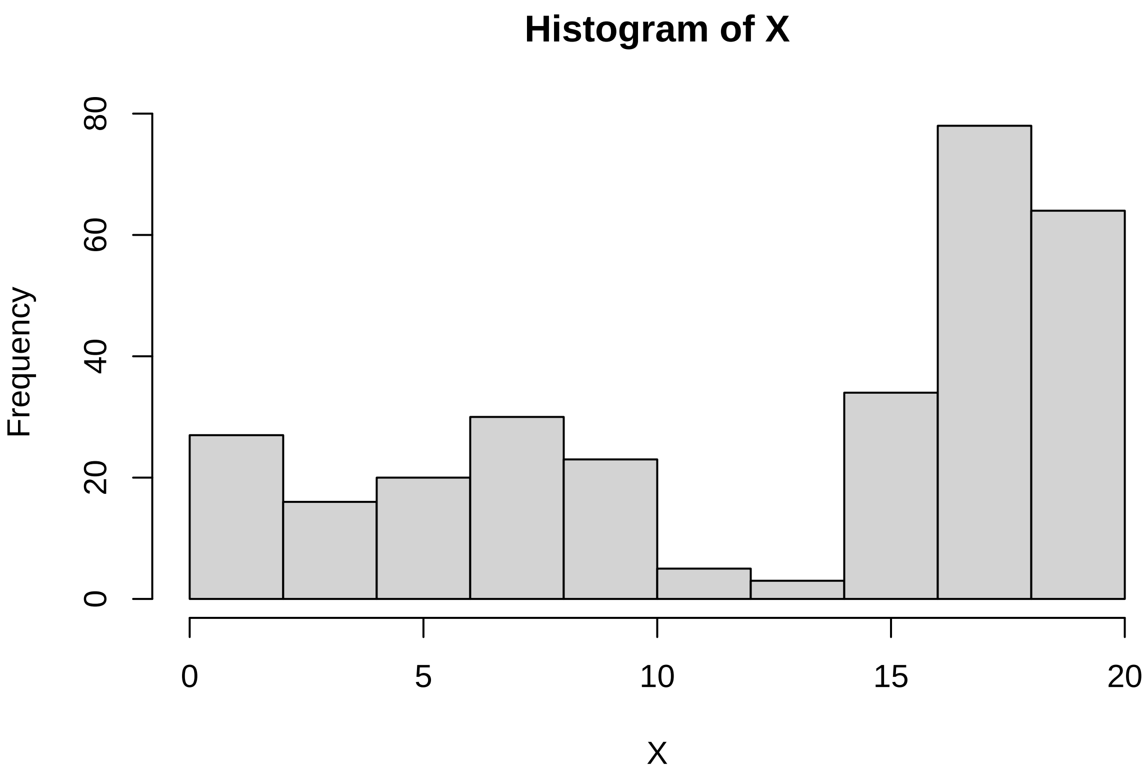
2d Implement the EM algorithm and obtain MLE of the paramenters.

Read the data

```
X = scan(file="assignment2_prob2.txt", what=double())
length(X)
```

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

```
hist(X)
```



Implementation of the EM algorithm

```
mixture.EM = function(X, w.init, p.init, epsilon=1e-5, max.iter=100) {

  # initialize current parameter values
  w.curr = w.init
  p.curr = p.init

  # compute incomplete log=likelihoods using intial value of parameters.
  log_liks = c()
  log_liks = c(log_liks, compute.log.lik(X, w.curr, p.curr)$ill)

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

  # number of iterations
  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 the EM step
    EM.out = EM.iter(X, w.curr, p.curr)

    # replace the current parameter estimates
    w.curr = EM.out$w.new
    p.curr = EM.out$p.new

    # compute the change in incomplete log-likelihood
    log_liks = c(log_liks, compute.log.lik(X, w.curr, p.curr)$ill)
    delta.ll = log_liks[length(log_liks)] - log_liks[length(log_liks) - 1]

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

# EM-iteration
EM.iter = function(X, w.curr, p.curr) {

  # E-step
  prob.x.z = compute.prob.x.z(X, w.curr, p.curr)$prob.x.z
  P_ik = prob.x.z / rowSums(prob.x.z)

  # M-step
  w.new = colSums(P_ik) / sum(P_ik)
  p.new = colSums(P_ik * X) / colSums(P_ik) / 20

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

# Incomplete log-likelihoods
compute.log.lik = function(X, w.curr, p.curr) {

  # compute probabilities
  prob.x.z = compute.prob.x.z(X, w.curr, p.curr)$prob.x.z

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

  return(list(ill=ill))
}

# Compute probabilities
compute.prob.x.z = function(X, w.curr, p.curr) {

  L = matrix(NA, nrow=length(X), ncol=length(w.curr))
  for (k in seq_len(ncol(L))) {
    L[,k] = dbinom(X, size=20, prob=p.curr[k]) * w.curr[k]
  }
  return(list(prob.x.z=L))
}
```

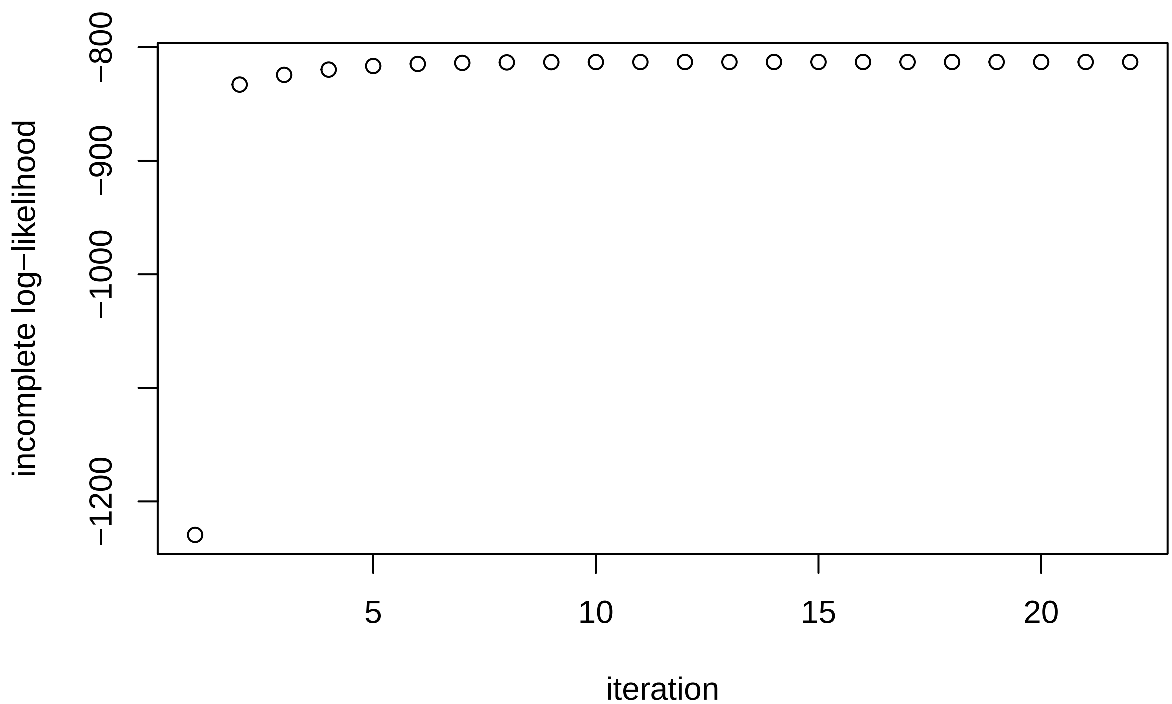
Apply the EM algorithm

```
EM1 = mixture.EM(X, w.init=c(0.3,0.3,0.4), p.init=c(0.2, 0.5, 0.7))
EM2 = mixture.EM(X, w.init=c(0.1,0.2,0.7), p.init=c(0.1, 0.3, 0.7))
```

Print results

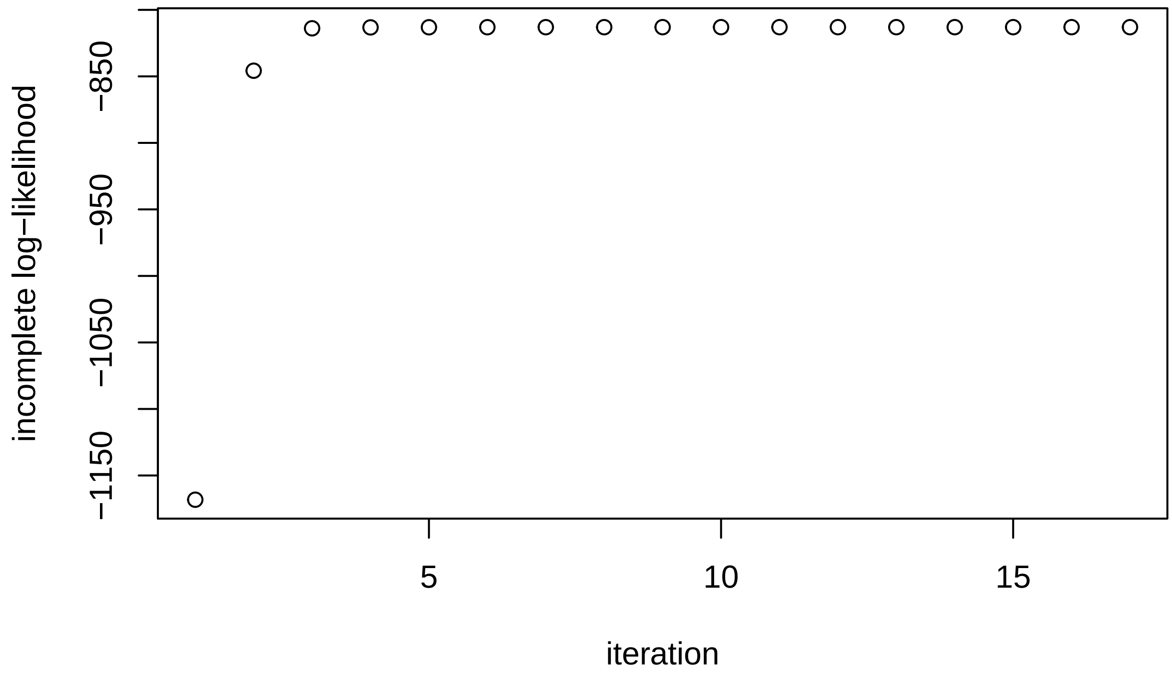
```
print.results <- function(EM) {
  print(paste("Estimate pi = (", round(EM$w.curr[1],2), ",",
              round(EM$w.curr[2],2), ",",
              round(EM$w.curr[3],2), ")", sep=""))
  print(paste("Estimate p = (", round(EM$p.curr[1],2), ",",
                round(EM$p.curr[2],2), ",",
                round(EM$p.curr[3],2), ")", sep=""))
  plot(EM$log_liks, ylab="incomplete log-likelihood", xlab="iteration")
}
print.results(EM1)
```

```
## [1] "Estimate pi = (0.12,0.28,0.6)"
## [1] "Estimate p = (0.09,0.38,0.89)"
```




```
print.results(EM2)
```

```
## [1] "Estimate pi = (0.12,0.28,0.6)"
## [1] "Estimate p = (0.09,0.38,0.89)"
```


incomplete log−likelihood
−850
−950
−1050
−1150
5
10
15
iteration

Question 3

3a The expectation of the complete log-likelihood.

The likelihood of observing $X = (X_1, ..., X_{300}, X_{301}, ..., X_{400})$ and $Z = (Z_1, ..., Z_{300})$ given $\theta = (\pi_1, \pi_2, p_1, p_2, p_3)$ is calculated, noting that $\pi_3 = 1 - \pi_1 - \pi_2$.

$$\begin{aligned}
&\Pr(X_1, ..., X_{300}, X_{301}, ..., X_{400}, Z_1, ..., Z_{300}|\theta) \\
=&\Pi_{i=1}^{300}[\Pr(X_i|Z_i, \theta) \cdot \Pr(Z_i|\theta)]\Pi_{i=301}^{400} \Pr(X_j|\theta) \\
=&\Pi_{i=1}^{300}\Pi_{j=1}^{3}[\Pr(X_i|Z_i = j, \theta) \cdot \Pr(Z_i = j|\theta)]^{I(Z_i=j)} \cdot \Pi_{i=301}^{400} \Pr(X_j|\theta)
\end{aligned}$$

The log-likelihood is then calculated.

$$\begin{aligned}
&\log \Pr(X_1, ..., X_{300}, Z_1, ..., Z_{300}|\theta) \\
=&\sum_{i=1}^{300}[\sum_{j=1}^{3} I(Z_i = j)(\log(\Pr(X_i|Z_i = j, \theta)) + \log(\Pr(Z_i = j|\theta)))] + \sum_{k=301}^{400} \log(\Pr(X_k|\theta)) \\
=&\sum_{i=1}^{300}[\sum_{j=1}^{3} I(Z_i = j)(\log \binom{20}{x_i} \cdot p_j^{x_i} \cdot (1 - p_j)^{20-x_i} + \log \pi_j)] + \sum_{k=301}^{400} [\log(\binom{20}{x_k} p_1^{x_k} (1 - p_1)^{20-x_k})] \\
=&\sum_{i=1}^{300}\sum_{j=1}^{3}[I(Z_i = j)(x_i \cdot \log p_j + (20 - x_i) \log(1 - p_j) + \log \binom{20}{x_i} + \log \pi_j)]+ \\
&\sum_{k=301}^{400} [log \binom{20}{x_k} + x_k \log p_1 + (20 - x_k) \log(1 - p_1)]
\end{aligned}$$

Finally, we can take the expectation to derive the complete log-likelihood.

$$\begin{aligned}
&Q(\theta, \theta^0) \\
=&E_{Z|X,\theta^0}[\log(\Pr(X, Z|\theta))] \\
=&\sum_{i=1}^{300}[\sum_{j=1}^{3} \Pr(Z_i = j|X, \theta^0)(x_i \log p_j + (20 - x_i) \log(1 - p_j) + \log \binom{20}{x_i} + \log \pi_j)]+ \\
&\sum_{k=301}^{400} [\log \binom{20}{x_k} + x_k \log p_1 + (20 - x_k) \log(1 - p_1)]
\end{aligned}$$

3b Derive E-step and M-step of the EM algorithm.

Firstly, we compute the E-step.

$$\begin{aligned}
&\Pr(Z_i = j|X, \theta^0)\\
=&\frac{\Pr(Z_i = j, X_i|\theta^0)}{\Pr(X_i|\theta^0)}\\
=&\frac{\Pr(X_i|Z_i = j, \theta^0)\Pr(Z_i = j|\theta^0)}{\sum_{k=1}^{3}\Pr(X_i|Z_i = k, \theta^0)\Pr(Z_i = k|\theta^0)}\\
=&\frac{\binom{20}{x_i}p_j^{x_i}(1-p_j)^{20-x_i}\pi_j}{\sum_{k=1}^{3}\binom{20}{x_i}p_k^{x_i}(1-p_k)^{20-x_i}\pi_k}
\end{aligned}$$

Secondly, we compute the M-step. The proportion estimates are similar to the previous derivation.

$$\frac{\partial Q(\theta, \theta^0)}{\partial \pi_j} = 0$$

$$\sum_{i=1}^{300}[\frac{\Pr(Z_i = 1|X, \theta^0)}{\pi_1} - \frac{\Pr(Z_i = 3|Z, \theta^0)}{1-\pi_1-\pi_2}] = 0$$

$$\hat{\pi_j} = \frac{1}{300}P(Z_i = j|X, \theta^0)$$

We differentiate w.r.t. p_1 to obtain the new parameter estimates for p_1.

$$\frac{\partial Q(\theta, \theta^0)}{\partial p_1} = 0$$

$$\sum_{i=1}^{300}[\Pr(Z_i = 1|X, \theta)(x_i(1-p_1) - (20-x_i)p_1)] + \sum_{i=301}^{400}[x_k(1-p_1) - (20-x_k)p_1] = 0$$

$$\sum_{i=1}^{300}\Pr(Z_i = 1|X_i, \theta)x_i - 20\sum_{i=1}^{300}\Pr(Z_i = 1|X_i, \theta)p_1 + \sum_{i=301}^{400}x_k - 20\sum_{i=301}^{400}p_1 = 0$$

$$\hat{p_1} = \frac{\sum_{i=1}^{300}\Pr(Z_i = 1|X_i, \theta)x_i + \sum_{k=301}^{400}x_k}{20(\sum_{i=1}^{300}\Pr(Z_i = 1|X_i, \theta) + 100)}$$

Finally, we compute the new parameter estimates for p_2 and p_3.

$$\frac{\partial Q(\theta, \theta^0)}{\partial p_2} = 0$$

$$\sum_{i=1}^{300}\Pr(Z_i = 2|X, \theta^0)(\frac{x_i}{p_2} - \frac{20-x_i}{1-p_2}) = 0$$

$$\sum_{i=1}^{300}\Pr(Z_i = 2|X, \theta^0)(x_i(1-p_2) - (20-x_i)p_2) = 0$$

$$\sum_{i=1}^{300}\Pr(Z_i = 2|Z, \theta^0)x_i - 20\cdot p_2\sum_{i=1}^{300}\Pr(Z_i = 2|Z, \theta^0) = 0$$

$$\hat{p_2} = \frac{\sum_{i=1}^{300}\Pr(Z_i = 2|X, \theta^0)x_i}{20\sum_{i=1}^{300}\Pr(Z_i = 2|X, \theta^0)}$$

$$\hat{p_3} = \frac{\sum_{i=1}^{300}\Pr(Z_i = 3|X, \theta^0)x_i}{20\sum_{i=1}^{300}\Pr(Z_i = 3|X, \theta^0)}$$

3c Implement and run the EM algorithm.

Read the data

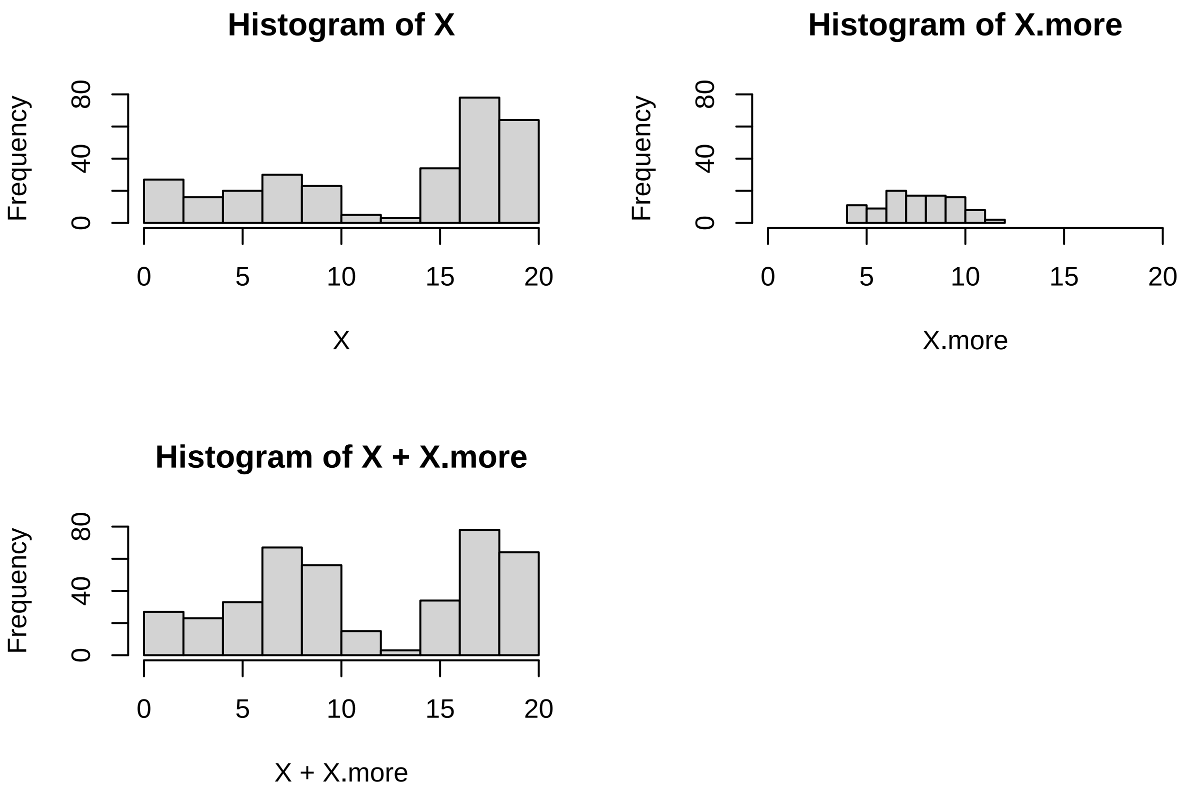
```
X = scan(file="assignment2_prob2.txt", what=double())
X.more = scan(file="assignment2_prob3.txt", what=double())
length(X)
```

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

```
length(X.more)
```

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

```
par(mfrow=c(2,2))
hist(X, xlim=c(0,20), ylim=c(0,80))
hist(X.more, xlim=c(0,20), ylim=c(0,80))
hist(c(X,X.more), xlim=c(0,20), ylim=c(0,80), xlab="X + X.more",
     main = "Histogram of X + X.more")
```



Implementation

```
mixture.EM = function(X, X.more, w.init, p.init, epsilon=1e-5, max.iter=100) {

  # initialize current parameter values
  w.curr = w.init
  p.curr = p.init

  # compute incomplete log=likelihoods using intial value of parameters.
  log_liks = c()
  log_liks = c(log_liks, compute.log.lik(X, X.more, w.curr, p.curr)$ill)
```

```r
  # change in incomplete log-likelihood
  delta.ll = 1

  # number of iterations
  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 the EM step
    EM.out = EM.iter(X, X.more, w.curr, p.curr)

    # replace the current parameter estimates
    w.curr = EM.out$w.new
    p.curr = EM.out$p.new

    # compute the change in incomplete log-likelihood
    log_liks = c(log_liks, compute.log.lik(X, X.more, w.curr, p.curr)$ill)
    delta.ll = log_liks[length(log_liks)] - log_liks[length(log_liks) - 1]

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

# EM-iteration
EM.iter = function(X, X.more, w.curr, p.curr) {

  # E-step
  prob.x.z = compute.prob.x.z(X, X.more, w.curr, p.curr)$prob.x.z
  P_ik = (prob.x.z / rowSums(prob.x.z))[1:300,]

  # M-step
  w.new = colSums(P_ik[1:300,]) / sum(P_ik[1:300,])
  p.new = colSums((P_ik * X)[1:300,]) / colSums(P_ik[1:300,]) / 20
  p1.new = (colSums((P_ik * X)[1:300,])[1] + sum(X.more)) /
    (20 * (colSums(P_ik[seq(1,300),])[1] + 100))

  return(list(w.new=w.new, p.new=c(p1.new, p.new[2], p.new[3])))
}

# Compute Incomplete Log-likelihood
compute.log.lik = function(X, X.more, w.curr, p.curr) {

  # compute probabilities
  prob.x.z = compute.prob.x.z(X, X.more, w.curr, p.curr)$prob.x.z

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

  return(list(ill=ill))
}
```

```r
# Compute probabilities
compute.prob.x.z = function(X, X.more, w.curr, p.curr) {

  L = matrix(0, nrow=(length(X) + length(X.more)), ncol=length(w.curr))
  for (i in 1:length(X)) {
    for (k in 1:ncol(L)) {
      L[i,k] = dbinom(X[i], size=20, prob=p.curr[k]) * w.curr[k]
    }
  }
  for (i in 1:length(X.more)) {
    L[i+length(X),1] = dbinom(X.more[i], size=20, prob=p.curr[1])
  }
  return(list(prob.x.z=L))
}
```

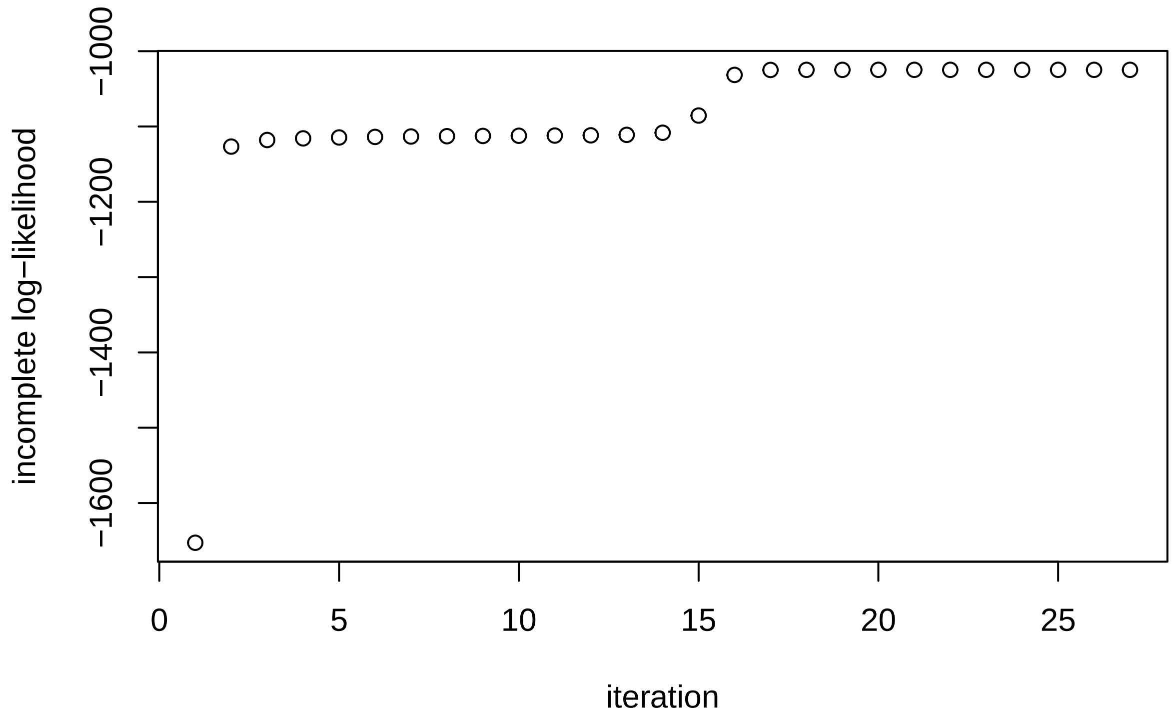
Apply the EM algorithm

```r
EM1 = mixture.EM(X, X.more, w.init=c(0.3,0.3,0.4), p.init=c(0.2, 0.5, 0.7))
EM2 = mixture.EM(X, X.more, w.init=c(0.1,0.2,0.7), p.init=c(0.1, 0.3, 0.7))
```

Print results

```r
print.results <- function(EM) {
  print(paste("Estimate pi = (", round(EM$w.curr[1],2), ",",
              round(EM$w.curr[2],2), ",",
              round(EM$w.curr[3],2), ")", sep=""))
  print(paste("Estimate p = (", round(EM$p.curr[1],2), ",",
                round(EM$p.curr[2],2), ",",
                round(EM$p.curr[3],2), ")", sep=""))
  plot(EM$log_liks, ylab="incomplete log-likelihood", xlab="iteration")
}
print.results(EM1)
```

```
## [1] "Estimate pi = (0.28,0.13,0.6)"
## [1] "Estimate p = (0.39,0.1,0.89)"
```




```
print.results(EM2)
```

```
## [1] "Estimate pi = (0.28,0.13,0.6)"
## [1] "Estimate p = (0.39,0.1,0.89)"
```