# Project 3

*Zihuan Qiao*

*10/22/2016*

### 1.

#### (a) See the attached sheet

#### (b) See the attached sheet

#### (c)

```
library(magrittr)
#input X
X <- c(2, 0, 0, 1, 3, 0, 1, 6, 2, 0, 1, 0, 2, 0, 8, 0, 1, 3, 2, 0)
n <- length(X)

#Poisson pmf function
Poi <- function(x, lambda)
{
  result <- exp(-lambda) * (lambda^(x)) / factorial(x)
  return(result)
}

#E(Zi) function
EZi.t <- function(i, pi.t, lambdac.t, lambdad.t)
{
  result <- pi.t * Poi(X[i], lambdad.t) / (pi.t * Poi(X[i], lambdad.t) + (1 - pi.t) * Poi(X[i], lambdac
  return(result)
}

#estimate function
#set starting value for pi, lambdad, lambdac
pi.n <- 0.5
lambdac.n <- sum(X[X < mean(X)]) / length(X[X < mean(X)])
lambdad.n <- sum(X[X > mean(X)]) / length(X[X > mean(X)])
pi.l <- 100; lambdad.l <- 1000; lambdac.l <- 1000;
#find estimators
while ((abs(pi.n - pi.l) > 1e-8) | (abs(lambdac.n - lambdac.l) > 1e-8) | (abs(lambdad.n - lambdad.l) > 
  pi.l <- pi.n
  lambdac.l <- lambdac.n
  lambdad.l <- lambdad.n
  pi.n <- mean(1:20 %>% EZi.t(pi.l, lambdac.l, lambdad.l))
  lambdac.n <- sum((1-(1:20 %>% EZi.t(pi.l, lambdac.l, lambdad.l))) * X) / sum(1-(1:20 %>% EZi.t(pi.l, 
  lambdad.n <- sum((1:20 %>% EZi.t(pi.l, lambdac.l, lambdad.l)) * X) / sum(1:20 %>% EZi.t(pi.l, lambdac
}

pi.n
```

```
## [1] 0.1244548
```

```r
lambdac.n
```

```
## [1] 0.9581722
```

```r
lambdad.n
```

```
## [1] 6.115288
```

### (d)

```r
#the probability of the r_th intersection being dangerous given X[r]
dan <- function(r)
{
  (Poi(X[r],lambdad.n) * pi.n) / (Poi(X[r],lambdad.n) * pi.n + (Poi(X[r],lambdac.n) * (1-pi.n)))
}

for(i in 1:20)
{
  print(list(intersection = c(i), Pdan = c(dan(i))))
}
```

```
## $intersection
## [1] 1
##
## $Pdan
## [1] 0.03226482
##
## $intersection
## [1] 2
##
## $Pdan
## [1] 0.0008178443
##
## $intersection
## [1] 3
##
## $Pdan
## [1] 0.0008178443
##
## $intersection
## [1] 4
##
## $Pdan
## [1] 0.005196806
##
## $intersection
## [1] 5
##
## $Pdan
## [1] 0.1754532
##
```

```
## $intersection
## [1] 6
##
## $Pdan
## [1] 0.0008178443
##
## $intersection
## [1] 7
##
## $Pdan
## [1] 0.005196806
##
## $intersection
## [1] 8
##
## $Pdan
## [1] 0.9822437
##
## $intersection
## [1] 9
##
## $Pdan
## [1] 0.03226482
##
## $intersection
## [1] 10
##
## $Pdan
## [1] 0.0008178443
##
## $intersection
## [1] 11
##
## $Pdan
## [1] 0.005196806
##
## $intersection
## [1] 12
##
## $Pdan
## [1] 0.0008178443
##
## $intersection
## [1] 13
##
## $Pdan
## [1] 0.03226482
##
## $intersection
## [1] 14
##
## $Pdan
## [1] 0.0008178443
##
```

```
## $intersection
## [1] 15
##
## $Pdan
## [1] 0.9995564
##
## $intersection
## [1] 16
##
## $Pdan
## [1] 0.0008178443
##
## $intersection
## [1] 17
##
## $Pdan
## [1] 0.005196806
##
## $intersection
## [1] 18
##
## $Pdan
## [1] 0.1754532
##
## $intersection
## [1] 19
##
## $Pdan
## [1] 0.03226482
##
## $intersection
## [1] 20
##
## $Pdan
## [1] 0.0008178443
```

The probability of the first intersection being dangerous is 0.03226482. The probability of the fifth intersection being dangerous is 0.1754532.

I would flag the eighth intersection and the fifteenth intersection as black spots. Because they have very high probability of being dangerous, 0.9822437 and 0.9995564 which are higher than 0.9 and approache 1.

### (e)

```r
library(magrittr)
#input X
X <- c(2, 0, 0, 1, 3, 0, 1, 6, 2, 0, 1, 0, 2, 0, 8, 0, 1, 3, 2, 0)
n <- length(X)

#Poisson pmf function
Poi <- function(x, lambda)
{
```

```r
  result <- exp(-lambda) * (lambda^(x)) / factorial(x)
  return(result)
}

#E(Zi) function
EZi.t <- function(i, pi.t, lambdac.t, lambdad.t)
{
  result <- pi.t * Poi(X[i], lambdad.t) / (pi.t * Poi(X[i], lambdad.t) + (1 - pi.t) * Poi(X[i], lambdac
  return(result)
}

#estimate function
#set starting value for pi, lambdad, lambdac
pi.nn <- 0.5
lambdac.nn <- sum(X[X > mean(X)]) / length(X[X > mean(X)])
lambdad.nn <- sum(X[X < mean(X)]) / length(X[X < mean(X)])
pi.l <- 100; lambdad.l <- 1000; lambdac.l <- 1000;
#find estimators
while ((abs(pi.nn - pi.l) > 1e-8) | (abs(lambdac.nn - lambdac.l) > 1e-8) | (abs(lambdad.nn - lambdad.l)
  pi.l <- pi.nn
  lambdac.l <- lambdac.nn
  lambdad.l <- lambdad.nn
  pi.nn <- mean(1:20 %>% EZi.t(pi.l, lambdac.l, lambdad.l))
  lambdac.nn <- sum((1-(1:20 %>% EZi.t(pi.l, lambdac.l, lambdad.l))) * X) / sum(1-(1:20 %>% EZi.t(pi.l,
  lambdad.nn <- sum((1:20 %>% EZi.t(pi.l, lambdac.l, lambdad.l)) * X) / sum(1:20 %>% EZi.t(pi.l, lambdac
}

pi.nn
```

```
## [1] 0.8755452
```

```r
lambdac.nn
```

```
## [1] 6.115288
```

```r
lambdad.nn
```

```
## [1] 0.9581722
```

Two estimates for $\pi$ from (c) and (e) sum up to 1. And estimates for $\lambda_c$ and $\lambda_d$ are the same number but swapped.

For explanation and related derivation, please see the attachment.

## 2.

### (c)

```r
#write a function to calculate alphaj.t, returns a vector that contains value of alpha0.t-alpha3.t
alpha.t <- function(lambda.t)
{
  sapply(0:3,function(j){
  if(j >= 0 & j <= 2)
  {
    1 / lambda.t + (j * exp(-lambda.t * j)- (j + 1) * exp(-lambda.t * (j + 1))) / (exp(-lambda.t * j) -
  }
  else
  {
    1 / lambda.t + 3
  }
  })
}

#initialize nj
nj <- c(40, 29, 19 ,12)

#find estimator lambda
estimation <- function(lam0)
{
  l.n <- lam0; l.l <- 100 #l.n is the new estimation for lambda. l.l is the last estimation for lambda
  while(abs(l.n-l.l) >= 10^-8)
  {
    l.l <- l.n
    l.n <- sum(nj) / sum(nj * alpha.t(l.l))
  }
  l.n
}


#set starting value lambda0 for lambda to be . For the reason of choosing this starting value, please s
lambda0 <- 0.9708738
estimation(lambda0)
```

```
## [1] 0.6175444
```