STAT432 Assignment 2

Rory Sarten 301005654

21 August 2020

Question 1

a)

These are independent binomial processes (despite happening in sequence), so the three trapping occasions can be denoted:

$n_1 \sim Binomial(N, \theta) = \binom{N}{n_1}\theta^{n_1}(1-\theta)^{N-n_1}$

$n_2 \sim Binomial(N-n_1, \theta) = \binom{N-n_1}{n_2}\theta^{n_2}(1-\theta)^{N-n_1-n_2}$

$n_3 \sim Binomial(N-n_1-n_2, \theta) = \binom{N-n_1-n_2}{n_3}\theta^{n_3}(1-\theta)^{N-n_1-n_2-n_3}$

So the probability of catching $n_1+n_2+n_3$ rats is given by

$Binomial(N, \theta) \times Binomial(N-n_1, \theta) \times Binomial(N-n_1-N-2, \theta)$

b)

$$\begin{aligned}
\ell(N,\theta) =& \ln[\frac{N!}{n_1(N-n_1)!}\theta^{n_1}(1-\theta)^{N-n_1} \\
&\times \frac{(N-n_1)!}{n_2(N-n_1-n_2)!}\theta^{n_2}(1-\theta)^{N-n_1-n_2} \\
&\times \frac{(N-n_1-n_2)!}{n_3(N-n_1-n_2-n_3)!}\theta^{n_3}(1-\theta)^{N-n_1-n_2-n_3}]
\end{aligned}$$

$$\begin{aligned}
\ell(N,\theta) =& \ln N! - [\ln n_1 + \ln(N-n_1)] + n_1\ln\theta + (N-n_1)\ln(1-\theta) + \\
& \ln(N-n_1)! - [\ln n_2 + \ln(N-n_1-n_2)] + n_2\ln\theta + (N-n_1-n_2)\ln(1-\theta) + \\
& \ln(N-n_1-n_2)! - [\ln n_3 + \ln(N-n_1-n_2-n_3)] + n_3\ln\theta + (N-n_1-n_2-n_3)\ln(1-\theta)
\end{aligned}$$

Gives

$$\ln N! - \ln(N-n_1-n_2-n_3)! + (n_1+n_2+n_3)\ln\theta + (3N-3n_1-2n_2-n_3)\ln(1-\theta)$$

c)

```r
llfunc <- function(par, n1, n2, n3) {
  N <- par[1]
  theta <- par[2]
  lfactorial(N) - lfactorial(N - n1 - n2 - n3) +
    sum(n1, n2, n3)*log(theta) +
    (3*N - 3*n1 - 2*n2 - n3)*log(1 - theta)
}

n1 <- 82L
n2 <- 54L
n3 <- 23L
## since N >= n1 + n2 + n3
N_0 <- n1 + n2 + n3
par_start <- c(N_0, 0.5)
```

```r
optim_fit <- optim(par = par_start,
                   fn = llfunc,
                   n1 = n1, n2 = n2, n3 = n3,
                   lower = c(N_0, 1e-4),
                   upper = c(Inf, 1),
                   method = "L-BFGS-B",
                   control = list(fnscale = -1),
                   hessian = TRUE)

MLE <- optim_fit$par
SE <- sqrt(diag(solve(-optim_fit$hessian)))
LowerBound <- MLE - qnorm(0.975) * SE
UpperBound <- MLE + qnorm(0.975) * SE
results <- cbind(MLE, SE, LowerBound, UpperBound) %>%
  round(2)
rownames(results) <- c("N", "theta")
results
```

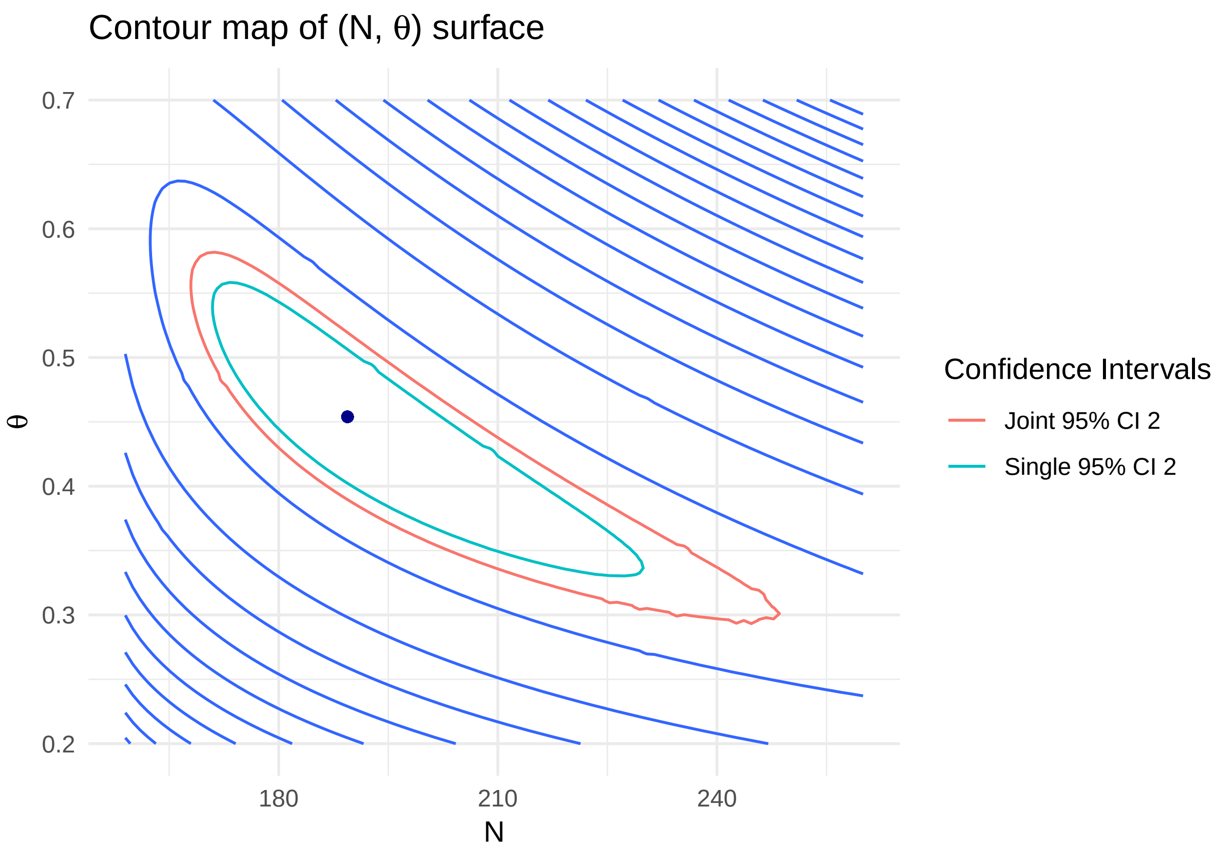
```
##          MLE    SE LowerBound UpperBound
## N     189.43 13.23     163.49     215.36
## theta   0.45  0.06       0.34       0.57
```

The maximum likelihood point $\hat{N} = 189.43$ with confidence interval $(163.49, 215.36)$

The maximum likelihood point $\hat{\theta} = 0.45$ with confidence interval $(0.34, 0.57)$

d)

```r
n_vals <- seq(from = n1+n2+n3, to = 260, length = 100)
theta_vals <- seq(from = 0.2, to = 0.7, length = 100)

combos <- expand.grid(n_vals, theta_vals)
surface <- apply(expand.grid(n_vals, theta_vals), 1, llfunc, n1 = n1, n2 = n2, n3 = n3) %>%
  round(2)
outcome <- cbind(combos, surface)
names(outcome) <- c("N", "theta", "value")

ggplot(data = outcome, mapping = aes(N, theta)) +
  geom_contour(aes(z = value),
               breaks = seq(from = min(outcome$value),
                            to = max(outcome$value), by = 12)) +
  geom_contour(aes(z = value,
                   colour = factor(..level.. == max(surface) - 1.92,
                                   levels = c(F, T),
                                   labels = c("Single 95% CI "))),
               breaks = max(surface) - 1.92) +
  geom_contour(aes(z = value,
                   colour = factor(..level.. == max(surface) - 3,
                                   levels = c(F, T),
                                   labels = c("Joint 95% CI "))),
               breaks = max(surface) - 3) +
  geom_point(mapping = aes(x = MLE[1], y = MLE[2]), colour = "darkblue") +
  labs(colour = "Confidence Intervals") +
  ggtitle(expression(paste("Contour map of (N, ", theta, ") surface"))) +
  ylab(expression(theta)) +
  theme_minimal()
```



e)

$$\frac{\partial \ell}{\partial \theta} = \frac{n_1 + n_2 + n_3}{\theta} - \frac{3N - 3n_1 - 2n_2 - n_3}{1 - \theta}$$

$$= \frac{(n_1 + n_2 + n_3) - (n_1 + n_2 + n_3)\theta - (3N - 3n_1 - 2n_2 - n_3)\theta}{\theta(1 - \theta)}$$

Setting this to 0

$$(n_1 + n_2 + n_3) - [(n_1 + n_2 + n_3) + (3N - 3n_1 - 2n_2 - n_3)]\theta = 0$$

$$[(n_1 + n_2 + n_3) + (3N - 3n_1 - 2n_2 - n_3)]\theta = (n_1 + n_2 + n_3)$$

$$\theta = \frac{(n_1 + n_2 + n_3)}{(n_1 + n_2 + n_3) + (3N - 3n_1 - 2n_2 - n_3)} = \frac{(n_1 + n_2 + n_3)}{3N - 3n_1 + n_1 - 2n_2 + n_2 - n_3 + n_3}$$

$$\hat{\theta} = \frac{n_1 + n_2 + n_3}{3N - 2n_1 - n_2}$$

f)

We have

$$\hat{\theta} = \frac{n_1 + n_2 + n_3}{3N - 2n_1 - n_2}$$

and

$$1 - \hat{\theta} = \frac{3N - 2n_1 - n_2}{3N - 2n_1 - n_2} - \frac{n_1 + n_2 + n_3}{3N - 2n_1 - n_2} = \frac{3N - 3n_1 - 2n_2 - n_3}{3N - 2n_1 - n_2}$$

So we can rewrite the original kernel of the log likelihood in terms of N as:

$$\ell(N) = \ln N! - \ln(N - n_1 - n_2 - n_3)! + (n_1 + n_2 + n_3)\ln\left(\frac{n_1 + n_2 + n_3}{3N - 2n_1 - n_2}\right) + (3N - 3n_1 - 2n_2 - n_3)\ln\left(\frac{3N - 3n_1 - 2n_2 - n_3}{3N - 2n_1 - n_2}\right)$$

$$\begin{aligned}\ell(N) = &\ln N! - \ln N! - \ln(N - n_1 - n_2 - n_3)! + \\ &(n_1 + n_2 + n_3)\ln(n_1 + n_2 + n_3) - (n_1 + n_2 + n_3)\ln(3N - 2n_1 - n_2) + \\ &(3N - 3n_1 - 2n_2 - n_3)\ln(3N - 3n_1 - 2n_2 - n_3) - (3N - 3n_1 - 2n_2 - n_3)\ln(3N - 2n_1 - n_2)\end{aligned}$$

$$\begin{aligned}\ell(N) = &\ln N! - \ln N! - \ln(N - n_1 - n_2 - n_3)! + \\ &(3N - 3n_1 - 2n_2 - n_3)\ln(3N - 3n_1 - 2n_2 - n_3) - \\ &(n_1 + n_2 + n_3 + 3N - 3n_1 - 2n_2 - n_3)\ln(3N - 2n_1 - n_2)\end{aligned}$$

Therefore

$$\begin{aligned}\ell(N) = &\ln N! - \ln N! - \ln(N - n_1 - n_2 - n_3)! + \\ &(3N - 3n_1 - 2n_2 - n_3)\ln(3N - 3n_1 - 2n_2 - n_3) - \\ &(3N - 2n_1 - n_2)\ln(3N - 2n_1 - n_2)\end{aligned}$$

g)

```r
llfunc_N <- function(N, n1, n2, n3) {
  lfactorial(N) - lfactorial(N - n1 - n2 - n3) +
    (3*N - 3*n1 - 2*n2 - n3)*log(3*N - 3*n1 - 2*n2 - n3) -
    (3*N - 2*n1 - n2)*log(3*N - 2*n1 - n2)
}

n1 <- 82L
n2 <- 54L
n3 <- 23L
## since N >= n1 + n2 + n3
N_0 <- n1 + n2 + n3
N_start <- c(N_0)

optim_fit <- optim(par = N_start,
                   fn = llfunc_N,
                   n1 = n1, n2 = n2, n3 = n3,
                   lower = c(N_0, 1e-4),
                   upper = c(Inf, 1),
                   method = "L-BFGS-B",
                   control = list(fnscale = -1),
```

```
                   hessian = TRUE)

N_hat <- optim_fit$par
log_max <- optim_fit$value

## calculate 95% CI
crit_point <- log_max - 0.5 * qchisq(0.95, df = 1)

ci_func <- function(critical_value, ...) {
  llfunc_N(...) - critical_value
}

lower_ci <- uniroot(ci_func, interval = c(N_0, N_hat),
                    n1 = n1, n2 = n2, n3 = n3,
                    critical_value = crit_point)$root
upper_ci <- uniroot(ci_func, interval = c(N_hat, 300),
                    n1 = n1, n2 = n2, n3 = n3,
                    critical_value = crit_point)$root

cbind(N_hat, lower_ci, upper_ci)
```

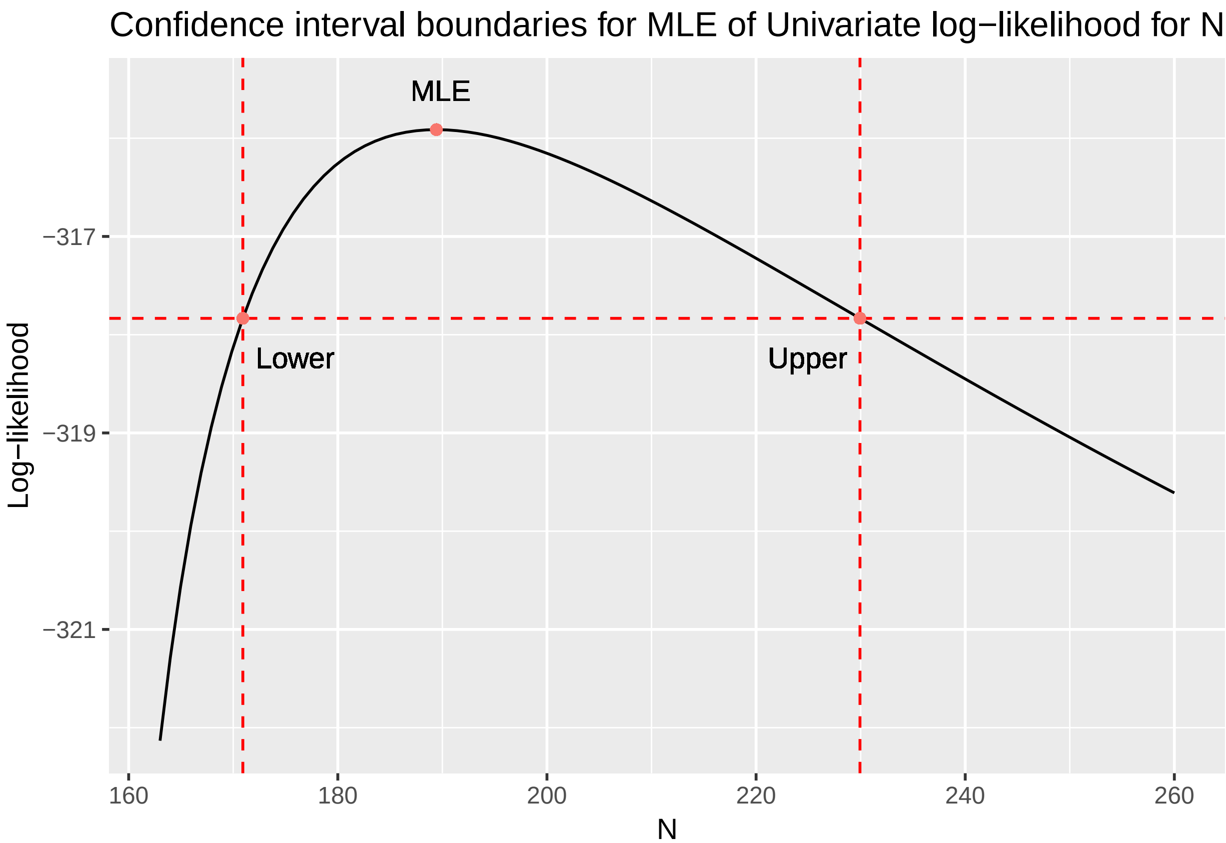
```
##         N_hat lower_ci upper_ci
## [1,] 189.4255  170.918 229.9337
```

The univariate log-likelihood estimate for $\hat{N} = 189.4254905$

with 95% confidence interval (170.92, 229.93).

Compared with the confidence interval in (c), this interval is similar in size, but is no longer symmetric around the MLE.

```
n_vals <- seq(from = 163, to = 260, length = 100)
data <- data.frame(x = n_vals,
                   y = llfunc_N(n_vals, n1 = n1, n2 = n2, n3 = n3))

ggplot(data = data, mapping = aes(x = x, y = y)) +
  geom_line() +
  geom_point(aes(x = N_hat, y = log_max, color = "red")) +
  geom_text(aes(x = N_hat + 0.4, y = log_max + 0.4, label = "MLE")) +
  geom_hline(yintercept=crit_point, linetype="dashed", color = "red") +
  geom_vline(xintercept=lower_ci, linetype="dashed", color = "red") +
  geom_vline(xintercept=upper_ci, linetype="dashed", color = "red") +
  geom_point(aes(x = lower_ci, y = crit_point, color = "red")) +
  geom_point(aes(x = upper_ci, y = crit_point, color = "red")) +
  geom_text(aes(x = lower_ci + 5, y = crit_point - 0.4, label = "Lower")) +
  geom_text(aes(x = upper_ci - 5, y = crit_point - 0.4, label = "Upper")) +
  ggtitle("Confidence interval boundaries for MLE of Univariate log-likelihood for N") +
  xlab("N") + ylab("Log-likelihood") +
  theme(legend.position="none")
```



h)

Given the 95% profile liklihood-ratio for N calculated above, and that we know a total of 159 rats were caught over the three occasions, we can our confidence interval for R to be $(lower - 159, upper - 159) = (171 - 159, 230 - 159) = (12, 71)$ (with rounding to the nearest whole rat).

Question 2

a)

$$\begin{aligned}
\ell(\lambda, x) &= \log(\prod_{i=1}^{n}\left(\frac{e^{-\lambda}\lambda^{x_i}}{x_i!}\right)) \\
&= \sum_{i=1}^{n}(-\lambda + x_i \ln\lambda - \ln(x_i!)) \\
&= -n\lambda + \left(\sum_{i=1}^{n} x_i\right)\ln\lambda - \sum_{i=1}^{n}\ln(x_i!)
\end{aligned}$$

The last term can be dropped, giving the kernel of the log likelihood

$$-n\lambda + \left(\sum_{i=1}^{n} x_i\right)\ln\lambda$$

Differentiating gives

$$\frac{d\ell}{d\lambda} = -n + \frac{\sum_{i=1}^{n} x_i}{\lambda}$$

Setting this to 0 gives

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

We can calculte the standard error by taking the second derivative of the log-likelihood at the MLE and calculating the negative inverse

$$\frac{d^2\ell}{d\lambda^2} = -\frac{\sum_{i=1}^{n} x_i}{\lambda^2}$$

$$\left.\frac{d^2\ell}{d\lambda^2}\right|_{\hat{\lambda}} = -\left[\frac{n}{\sum_{i=1}^{n} x_i}\right]^2 \times \sum_{i=1}^{n} x_i = -\frac{n^2}{\sum_{i=1}^{n} x_i}$$

The negative inverse is

$$\frac{\sum_{i=1}^{n} x_i}{n^2}$$

So we can calculate the standard error as

$$SE[\hat{\lambda}] = \frac{\sqrt{\sum_{i=1}^{n} x_i}}{n}$$

b)

i)

```
counts <- readr::read_rds("counts.rds")
MLE <- sum(counts)/length(counts) # or mean(counts)
SE <- (sqrt(sum(counts))/length(counts)) %>% round(digits = 2)
LOWER <- (MLE - SE * qnorm(0.975)) %>% round(digits = 2)
UPPER <- (MLE + SE * qnorm(0.975)) %>% round(digits = 2)
cbind(MLE, SE, LOWER, UPPER)
```

```
##        MLE   SE LOWER UPPER
## [1,] 4.66 0.31  4.05  5.27
```

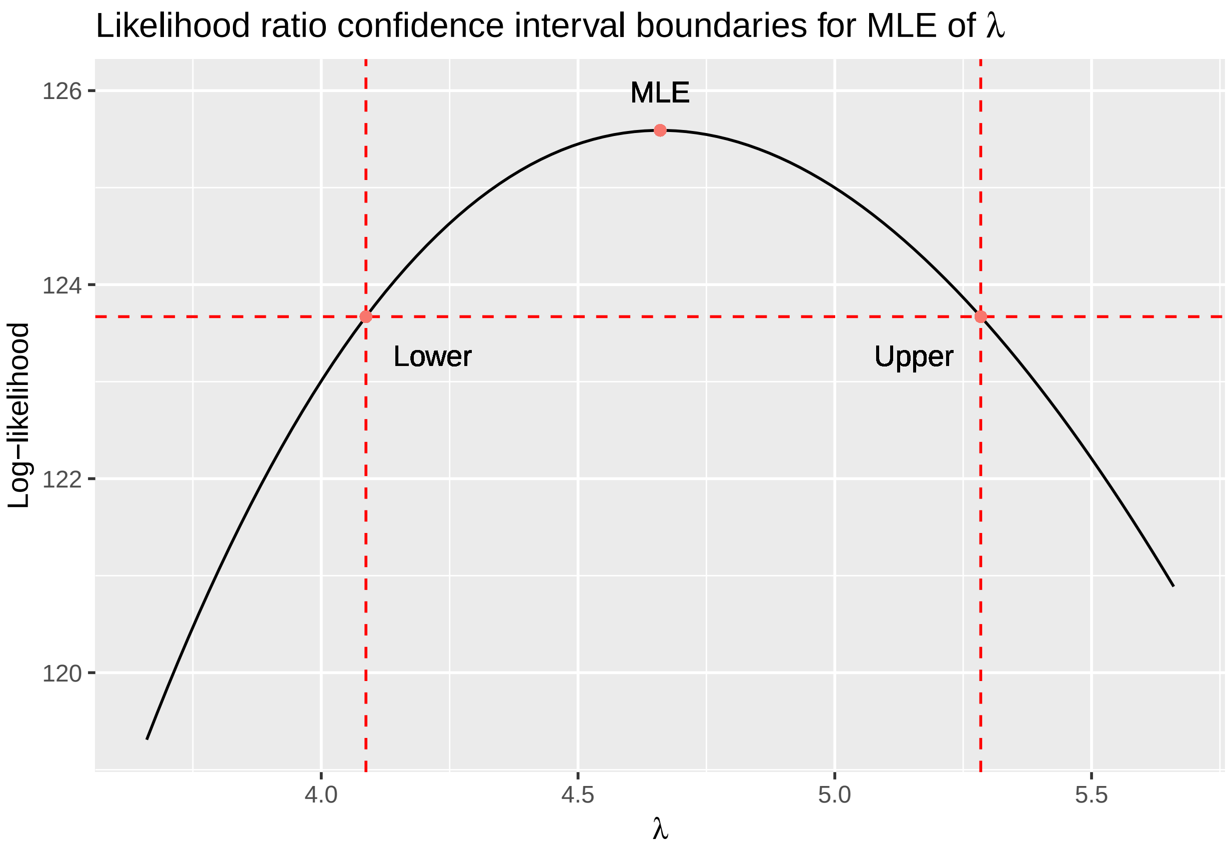
ii)

```
poisson_llfunc <- function(lambda, counts) {
  -(length(counts) * lambda) + sum(counts) * log(lambda)
}

optim_fit <- optim(par = 1,
                   fn = poisson_llfunc,
                   counts = counts,
                   lower = 1e-3,
                   upper = Inf,
                   method = "L-BFGS-B",
                   control = list(fnscale = -1),
                   hessian = TRUE)
```

```r
MLE <- optim_fit$par

n_vals <- seq(from = MLE - 1, to = MLE + 1, length = 100)
data <- data.frame(x = n_vals,
                   y = poisson_llfunc(n_vals, counts))

log_max <- optim_fit$value
crit_point <- optim_fit$value - 0.5 * qchisq(0.95, df = 1)
ci_func <- function(critical_point, ...) poisson_llfunc(...) - critical_point

lower_ci <- uniroot(ci_func,
                    interval = c(0, MLE),
                    counts = counts,
                    critical_point = crit_point)$root
upper_ci <- uniroot(ci_func,
                    interval = c(MLE, MLE*5),
                    counts = counts,
                    critical_point = crit_point)$root
cbind(MLE, lower_ci, upper_ci)
```

```
##       MLE lower_ci upper_ci
## [1,] 4.66 4.086982 5.284223
```

```r
ggplot(data = data, mapping = aes(x = x, y = y)) +
  geom_line() +
  geom_point(aes(x = MLE, y = log_max, color = "red")) +
  geom_text(aes(x = MLE, y = log_max + 0.4, label = "MLE")) +
  geom_hline(yintercept=crit_point, linetype="dashed", color = "red") +
  geom_vline(xintercept=lower_ci, linetype="dashed", color = "red") +
  geom_vline(xintercept=upper_ci, linetype="dashed", color = "red") +
  geom_point(aes(x = lower_ci, y = crit_point, color = "red")) +
  geom_point(aes(x = upper_ci, y = crit_point, color = "red")) +
  geom_text(aes(x = lower_ci + 0.13, y = crit_point - 0.4, label = "Lower")) +
  geom_text(aes(x = upper_ci - 0.13, y = crit_point - 0.4, label = "Upper")) +
  ggtitle(expression(paste(
    "Likelihood ratio confidence interval boundaries for MLE of ", lambda))) +
  xlab(expression(lambda)) + ylab("Log-likelihood") +
  theme(legend.position="none")
```



c)

d) this is i)

$$\ell(\boldsymbol{\lambda}, \boldsymbol{\pi}|\boldsymbol{x}) = \sum_{i=1}^{n} \ln \left(\sum_{g=1}^{G} \pi_g \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \right)$$

$$\ell_C(\boldsymbol{\lambda}, \boldsymbol{\pi}|\boldsymbol{x}, Z) = \sum_{i=1}^{n} \ln \left(\sum_{g=1}^{G} z_{ig}\pi_g \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \right) \text{ where } z_{ig} \text{ has value 0 or 1}$$

ii)

If we are looking at each $g = 1, ..., G$ then we can ignore the π_g term, as we are not concerned with the proportion in the group g. We can also exploit that z_{ig} is 0 or 1, by treating it as an exponent.

$$\begin{aligned}
\ell_C(\boldsymbol{\lambda}, \boldsymbol{\pi}|\boldsymbol{x}, Z) &= \sum_{i=1}^{n} \ln \left(z_i \frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \right) \\
&= \sum_{i=1}^{n} \ln \left(\left[\frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \right]^{z_i} \right) \\
&= \sum_{i=1}^{n} z_i \ln \left(\frac{e^{-\lambda}\lambda^{x_i}}{x_i!} \right) \\
&= \sum_{i=1}^{n} z_i \left(-\lambda + x_i \ln \lambda - \ln x_i! \right) \\
&= -\lambda \sum_{i=1}^{n} z_i + \sum_{i=1}^{n} z_i x_i \ln \lambda - \sum_{i=1}^{n} z_i \ln x_i!
\end{aligned}$$

$$\frac{\partial \ell_C}{\partial \lambda} = -\sum_{i=1}^{n} z_i + \frac{\sum_{i=1}^{n} z_i x_i}{\lambda}$$

Setting equal to 0 and solving

$$\frac{\sum_{i=1}^{n} z_i x_i}{\lambda} = \sum_{i=1}^{n} z_i$$

therefore for each group g

$$\hat{\lambda_g} = \frac{\sum_{i=1}^{n} z_{ig} x_i}{\sum_{i=1}^{n} z_{ig}}$$

iv)

```
parameters <- list(g1 = c(lambda_est = 3, pi_est = 0.5),
                   g2 = c(lambda_est = 1, pi_est = 0.5))

## start with high max distance from parameters
previous_parameters <- list(g1 = c(lambda_est = 30, pi_est = 0.5),
                            g2 = c(lambda_est = 1, pi_est = 0.5))


####################################################
## Supporting functions
z_col <- function(parameters, x)
  parameters["pi_est"] * exp(-parameters["lambda_est"]) * parameters["lambda_est"]^x

z_matrix <- function(parameters, x) {
  Z <- lapply(parameters, z_col, x) %>%
    data.frame() %>%
    as.matrix()
  Z/apply(Z, 1, sum)
}

calc_lambda <- function(Z, x)
  as.data.frame(Z) %>% `*`(x) %>% apply(2, sum) %>% `/`(apply(Z, 2, sum))

not_converged <- function(current_para, previous_para, threshold = 1e-4)
  any(max(abs(unlist(current_para) - unlist(previous_para))) > threshold)
####################################################

while (not_converged(parameters, previous_parameters)) {
  previous_parameters <- parameters

  ## E-step - estimate Z from current parameters
  Z <- z_matrix(parameters, counts)

  ## M-step - update estimates for parameters
  ##pi_est
  parameters <- purrr::map2(parameters, apply(Z, 2, mean),
                            function(x, y) {x["pi_est"] <- y; x})
  ## lambda_est
  parameters <- purrr::map2(parameters, calc_lambda(Z, counts),
                            function(x, y) {x["lambda_est"] <- y; x})
}
```

```
parameters
```

```
## $g1
## lambda_est     pi_est
##   7.162798   0.454524
##
## $g2
## lambda_est     pi_est
##   2.574516   0.545476
```

Question 3

a)

$$
\begin{aligned}
\ell(\boldsymbol{p}|\boldsymbol{y}) &= \ln\left(\frac{n!}{\prod_{j=1}^m y_j!}\prod_{j=1}^m p_j^{y_j}\right) \\
&= \ln\left(\frac{n!}{\prod_{j=1}^m y_j!}\right) + \ln\left(\prod_{j=1}^m p_j^{y_j}\right) \\
&= c + \sum_{j=1}^m y_i \ln p_j \qquad \text{drop constant} \\
&= \sum_{j=1}^m y_i \ln p_j
\end{aligned}
$$

b)

From the result above we have the case where

$$
\begin{aligned}
\ell(\theta|\boldsymbol{y}) &= y_1 \ln p_1 + y_2 \ln p_2 + y_3 \ln p_3 \\
&= y_1 \ln\left(\frac{1}{2}\right) + y_2 \ln\left(\frac{1}{4}+\theta\right) + y_3 \ln\left(\frac{1}{4}-\theta\right) \\
\frac{d\ell}{d\theta} &= \frac{y_2}{\frac{1}{4}+\theta} - \frac{y_3}{\frac{1}{4}-\theta} \\
&\text{set to } 0 \\
0 &= \frac{y_2}{\frac{1}{4}+\theta} - \frac{y_3}{\frac{1}{4}-\theta} \\
y_2\left(\frac{1}{4}-\theta\right) &= y_3\left(\frac{1}{4}+\theta\right) \\
\theta(y_2+y_3) &= \frac{1}{4}(y_2-y_3) \\
\hat{\theta} &= \frac{1}{4}\frac{y_2-y_3}{y_2+y_3}
\end{aligned}
$$

$$
\begin{aligned}
\frac{d^2\ell}{d\theta^2} &= -\frac{y_2}{(\frac{1}{4}+\theta)^2} - \frac{y_3}{(\frac{1}{4}-\theta)^2} \\
\left.\frac{d^2\ell}{d\theta^2}\right|_{\hat{\theta}} &= -\frac{y_2}{(\frac{1}{4}+\frac{1}{4}\frac{y_2-y_3}{y_2+y_3})^2} - \frac{y_3}{(\frac{1}{4}-\frac{1}{4}\frac{y_2-y_3}{y_2+y_3})^2} \\
&= -\frac{y_2}{(\frac{1}{4}\frac{2y_2}{y_2+y_3})^2} - \frac{y_3}{(\frac{1}{4}\frac{2y_3}{y_2+y_3})^2} \\
&= -\frac{y_2}{(\frac{1}{2}\frac{y_2}{y_2+y_3})^2} - \frac{y_3}{(\frac{1}{2}\frac{y_3}{y_2+y_3})^2} \\
&= -\frac{y_2[4(y_2+y_3)^2]}{y_2^2} - \frac{y_3[4(y_2+y_3)^2]}{y_3^2} \\
&= -4(y_2+y_3)^2\left(\frac{1}{y_2}+\frac{1}{y_3}\right) \\
&= -4\frac{(y_2+y_3)^2}{y_2y_3} \qquad \text{then find the negative inverse} \\
&= \frac{1}{4}\frac{y_2y_3}{(y_2+y_3)^2} \qquad \text{and square to give} \\
SE[\hat{\theta}] &= \frac{1}{2}\sqrt{\frac{y_2y_3}{(y_2+y_3)^2}}
\end{aligned}
$$

c)

$$q1 = \frac{\Pr(y_1)}{\Pr(y_{12})} = \frac{p_1}{p_1 \cap p_2} = \frac{\frac{1}{2}}{\frac{1}{2}\frac{1}{4}+\theta} = \frac{\frac{1}{2}}{\frac{3}{4}+\theta} = \frac{2}{3+4\theta}$$

As $(y_1, y_2) \sim Bin(y_{12}, \boldsymbol{q})$ where $\boldsymbol{q} = (q_1, 1-q_2)$

$$q_2 = 1 - q_1 = \frac{3+4\theta}{3+4\theta} - \frac{2}{3+4\theta} = \frac{1+4\theta}{3+4\theta}$$

Given these results it is straightforward to posit

$$\hat{y}_1 = y_{12}q_1 = \frac{2y_{12}}{3+4\theta}$$

$$\hat{y}_2 = y_{12}q_2 = \frac{(1+4\theta)y_{12}}{3+4\theta}$$

d)

```
y12 <- 592
parameters <- list(y1 = y12/2, y2 = y12/2, y3 = 183, y12 = 592, theta = 0.2)
previous_parameters <- list(y1 = 592/2, y2 = y12/2, y3 = 183, y12 = 592, theta = 1)

####################################################
## Supporting functions

recalc_y <- function(p) {
  p[["y1"]] <- (2*p[["y12"]])/(3 + 4*p[["theta"]])
  p[["y2"]] <- ((1 + 4*p[["theta"]])*p[["y12"]])/(3 + 4*p[["theta"]])
  p
}

recalc_theta <- function(p) {
  p[["theta"]] <- (1/4)* (p[["y2"]] - p[["y3"]])/(p[["y2"]] + p[["y3"]])
  p
}
```

```r
not_converged <- function(current_para, previous_para, para = "theta", threshold = 1e-6)
  abs(current_para[[para]] - previous_para[[para]]) > threshold
####################################################

while (not_converged(parameters, previous_parameters)) {
  previous_parameters <- parameters

  ## E-step - estimate new values for y1 and y2 using current theta
  parameters <- recalc_y(parameters)

  ## M-step - update estimate for theta
  parameters <- recalc_theta(parameters)
}

parameters
```

```
## $y1
## [1] 387.4996
##
## $y2
## [1] 204.5004
##
## $y3
## [1] 183
##
## $y12
## [1] 592
##
## $theta
## [1] 0.01387124
```

```r
p <- parameters
se_theta <- (1/2)*sqrt((p[["y2"]]*p[["y3"]])/(p[["y2"]] + p[["y3"]])^3)
se_theta
```

```
## [1] 0.01268044
```

The EM estimate $\hat{\theta}$ converged at 0.0139 with a standard error of 0.0127