

Problem Set 7 (answers)

S352

```
1. (a)    normal.nll <- function(pars, data){  
      mu <- pars[1]  
      sigma2 <- pars[2]  
      lik <- dnorm(data, mu, sqrt(sigma2))  
      return(-sum(log(lik)))  
    }  
  
    womensheights <- scan("womensheights.txt")  
  
    women.mle <- optim(par = c(160, 50), fn = normal.nll, data = womensheights,  
                        hessian = TRUE)  
  
    > women.mle$par  
    [1] 162.05788 52.91235
```

I got MLEs: $\hat{\mu} = 162.06$ cm and $\hat{\sigma}^2 = 52.91$ cm².

(b) Since we already ran `optim()`, we have the Hessian, and all we need to do is invert it:

```
> solve(women.mle$hessian)  
[,1] [,2]  
[1,] 1.405001e-02 4.533107e-05  
[2,] 4.533107e-05 1.487340e+00
```

(c) Take square root of the variance to get standard error:

```
se <- sqrt(diag(solve(women.mle$hessian)))
```

(d) # CI for the mean:

```
women.mle$par[1] - qnorm(.995) * se[1]  
women.mle$par[1] + qnorm(.995) * se[1]
```

A 99% Wald confidence interval for μ , the average of women's heights, is 161.75 cm to 162.36 cm.

(e) # CI for the variance:

```
women.mle$par[2] - qnorm(.995) * se[2]  
women.mle$par[2] + qnorm(.995) * se[2]
```

I get a 99% Wald CI for σ^2 , the variance of women's heights, 49.77 to 56.05.

(f) From

$$0.99 = P(L_{\sigma^2} < \sigma^2 < U_{\sigma^2}) = P(\sqrt{L_{\sigma^2}} < \sigma < \sqrt{U_{\sigma^2}}),$$

we can tell that the confidence bounds for σ are square roots of those for σ^2 . Since we got a 99% CI for the variance, 49.77 to 56.05. which equates to a 99% CI for the SD of 7.05 to 7.49 cm.

If you parameterized by the SD rather than the variance you'll get a little bit different but equally correct result. The difference is why it might be a bit better to do LR inference rather than Wald inference here.

(g)

$$H_0 : \mu = 161.3$$

We can get the Wald test statistic and then a P -value as follows:

```
w <- (women.mle$par[1] - 161.3)^2 / solve(women.mle$hessian)[1,1]
1 - pchisq(w, df = 1)
```

The Wald test P -value is about 1.62×10^{-10} . The evidence is strong for us to reject the null hypothesis that the true mean is 161.3 cm.

2. (a) Let's draw a picture first:

```
eruptions <- faithful$eruptions
plot(density(faithful$eruptions), main = "Eruptions density plot")
```

From the density plot, we confirm that a mixture of two Normals might be a good model. At a glance, it looks like maybe 40% of the area is under the left Normal and 60% is under the right Normal. The left Normal is centered near 2 with an SD of maybe 0.5; the right Normal is centered near 4 with an SD of maybe 0.5. We can use these as initial values in a call to `optim()`.

Adapting Millar's code:

```
normal.nll <- function(pars, data){
  p <- pars[1]
  mu <- pars[2]
  sigma <- pars[3]
  nu <- pars[4]
  tau <- pars[5]
  lik <- p * dnorm(data, mu, sigma) + (1 - p) * dnorm(data, nu, tau)
  return(-sum(log(lik)))
}

eruptions.mle <- optim(par = c(0.4, 2, 0.5, 4, 0.5), fn = normal.nll,
                       data = eruptions, hessian = TRUE)
```

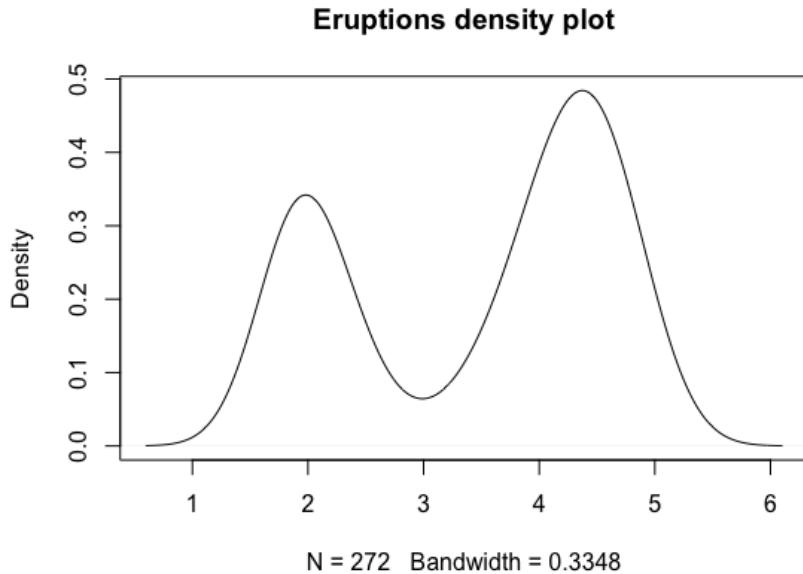


Figure 1: Density plot of Old Faithful eruption times, question 2(a).

```

estimates <- eruptions.mle$par
var.matrix <- solve(eruptions.mle$hessian)
std.errors <- sqrt(diag(var.matrix))

lower.CI <- estimates - qnorm(.975) * std.errors
upper.CI <- estimates + qnorm(.975) * std.errors

results <- data.frame(estimates, std.errors, lower.CI, upper.CI)
row.names(results) <- c("p", "mu", "sigma", "nu", "tau")

```

Print out the results. The five rows are for probability of first Normal, mean of first Normal, SD of first Normal, mean of second Normal, and SD of second Normal.

```

> round(results, 3)
      estimates std.errors lower.CI upper.CI
p          0.348     0.029    0.291    0.406
mu        2.019     0.026    1.967    2.070
sigma     0.236     0.023    0.190    0.281
nu        4.273     0.034    4.207    4.340
tau       0.437     0.027    0.384    0.490

```

These estimates aren't very stable. Different initial values may lead to different results, but they should be in this ballpark.

```
(b)      w <- (estimates[1] - 0.5)^2 / var.matrix[1, 1]
        1 - pchisq(w, df = 1)
```

I got a Wald statistic of 27.0 and a P -value of 2×10^{-7} . Again, these aren't very stable numbers, but it's clear that the two Normals aren't equally likely (the right one is more likely than the left one.)

3. (a) Poisson model, $\text{Poi}(\lambda)$

```
hurricanes <- read.csv("hurricanes.csv")
deaths <- hurricanes$alldeaths
lambda.hat <- mean(deaths)
lambda.hat
```

The MLE is $\hat{\lambda} = 20.65$.

- (b) Negative binomial, $\text{NB}(r, p)$

```
nbinom.nll <- function(params, data){
  size <- params[1]
  prob <- params[2]
  lik <- dnbinom(data, size, prob)
  loglik <- log(lik)
  nll <- -sum(loglik)
  return(nll)
}

x.bar <- mean(deaths)
sigma2.hat <- mean(deaths^2) - x.bar^2
p.mom <- x.bar / sigma2.hat
r.mom <- p.mom / (1 - p.mom) * x.bar

deaths.nb.mle <- optim(par = c(r.mom, p.mom), fn = nbinom.nll, data = deaths,
                        hessian = TRUE)
deaths.nb.mle$par
```

The MLEs are $\hat{r} = 0.446$ and $\hat{p} = 0.0211$.

- (c) Zero-inflated Poisson

```
pois0.nll <- function(pars, data){
  p <- pars[1]
  lambda <- pars[2]
  pois.lik <- dpois(data, lambda)
  real.lik <- rep(NA, length(data))
  which0 <- (data == 0)
  real.lik[which0 == FALSE] = (1 - p) * pois.lik[which0 == FALSE]
  real.lik[which0 == TRUE] = p + (1 - p) * pois.lik[which0 == TRUE]
```

```

        loglik <- log(real.liik)
        return(-sum(loglik))
    }

deaths.zip.mle <- optim(par = c(mean(deaths == 0), mean(deaths)), fn = pois0.nll,
                        data = deaths)
deaths.zip.mle$par

```

The MLEs are $\hat{p}_0 = 0.109$ and $\lambda = 23.2$.

(d) Zero-inflated negative binomial

```

nb0.nll <- function(pars, data){
    p <- pars[1]
    size <- pars[2]
    prob <- pars[3]
    nb.liik <- dnbinom(data, size, prob)
    real.liik <- rep(NA, length(data))
    which0 <- (data == 0)
    real.liik[which0 == FALSE] = (1 - p) * nb.liik[which0 == FALSE]
    real.liik[which0 == TRUE] = p + (1 - p) * nb.liik[which0 == TRUE]
    loglik <- log(real.liik)
    return(-sum(loglik))
}

deaths.nb0.mle <- optim(par = c(deaths.zip.mle$par[1], deaths.nb.mle$par[1],
                                  deaths.nb.mle$par[2]), fn = nb0.nll, data = deaths)

deaths.nb0.mle$par

```

The MLE for the probability of an automatic zero is -1.39 , which is not a probability. In fact, the probability of a zero is higher than necessary in the negative binomial model. So the probability of an automatic zero should be set to 0, and the model reduces to the ordinary negative binomial model.