# Statistical Computing - CWB - 2019

*Sharan Maiya (S1608480)*

```
# Sources, libraries, seed
source("CWB2019code.R")
library(tidyverse);library(xtable)
set.seed(10)
```

## Question 1

### Task 1

The function `negloglike` shown below takes as input the values $N$, $\theta$, $y_1$ and $y_2$ and outputs $l(N, \theta)$.

```
negloglike <- function(param, Y) {
  if (param[1] < max(Y)) { # If N >= max(y1, y2) then return +Infinity
    return(+Inf)
  } else { # Otherwise we calculate the negated log-likelihood
    return(sum(
      lgamma(Y + 1),
      lgamma(param[1] - Y + 1),
      -2 * lgamma(param[1] + 1),
      2 * param[1] * log(1 + exp(param[2])),
      -param[2] * sum(Y)
    ))
  }
}
```

### Task 2

We seek to use the `optim` function with `negloglike` to find a maximum likelihood estimate of $N$ and $\theta$ (and in turn $\phi$). Since `optim` is a numerical optimiser it is only guaranteed to find a local minima. It therefore makes sense to try `optim` at different sensible starting values to try and find the best MLE we can in a grid search. The following parameter values were tried as starting points:

- $N$: We know we must have $N > max(y_1, y_2)$ so we try both $N = max(y_1, y_2) + 1$ and $N = 2max(y_1, y_2)$.
- $\theta$: Since this is derived from the actual probability $\phi$ we choose sensible values of $\phi$ and convert them to a value of $\theta$ using the `logit` function provided. Since $\phi$ is a probability is makes sense to try starting at the values of 0.01, 0.5 and 0.99.

```
Y <- c(256, 237) # Given data
bestopt <- list(value = +Inf) # Initialise our optimisation
```

```r
# Perform the grid search
for (N_start in list(max(Y) + 1, 2 * max(Y))) {
  for (theta_start in lapply(list(0.01, 0.5, 0.99), logit)) {
    # Use optim with the current starting values
    opt <- optim(par = c(N_start, theta_start), fn = negloglike, Y = Y)
    if (opt$value < bestopt$value) { # Update if we found a better minima
      bestopt <- opt
    }
  }
}

# Record MLEs of N and theta
N_hat <- bestopt$par[1]
theta_hat <- bestopt$par[2]
# Obtain MLE of phi
phi_hat <- ilogit(theta_hat)
```

Table 1: MLEs for $\hat{N}$, $\hat{\theta}$ and $\hat{\phi}$

| $\hat{N}$ | $\hat{\theta}$ | $\hat{\phi}$ |
|---|---|---|
| 388.13 | 0.55 | 0.64 |

In Table 1 we see the maximum likelihood estimates for $N$, $\theta$ and consequently $\phi$. This means that in order to maximise the likelihood $p(y|N, \phi)$ we would require there to be around 388 people buried at the site, with a probability 0.64 of finding a femur.

### Task 3

We now want to take our values for $\hat{N}$ and $\hat{\theta}$ from Table 1 and use `optimHess` to determine a Hessian $\mathbf{H}$. The inverse $\mathbf{H}^{-1}$ will be a joint covariance matrix we can use to compute a 95% confidence interval for $N$. Here we use Normal approximation.

Table 2: 95% confidence interval for $N$

| 2.5% | 97.5% |
|---|---|
| -50.58 | 826.85 |

Table 2 shows a 95% confidence interval for $N$: we are 95% certain the true value of $N$ lies in this range. It is clear that this interval is not very helpful; the lower bound of -50.28 is well below the bound we had already deduced ($N > max(y_1, y_2)$). In fact, values of $N$ below zero are simply nonsensical as we cannot have a negative number of burials. Our upper bound of 826.85 is also considerably high: consider that even if every bone found belonged to a separate person the excavation would still only have found ~60% of the total number of burials should the true $N$ be near this figure. This seems very unlikely.

## Question 2

### Task 1

We have that the negated log-likelihood $l(N, \theta)$ is given by:

$l(N,\theta) = \log\Gamma(y_1+1) + \log\Gamma(y_2+1) + \log\Gamma(N-y_1+1) + \log\Gamma(N-y_2+1) - 2\log\Gamma(N+1) + 2N\log\left(1+e^{\theta}\right) - (y_1+y_2)\theta.$

We begin with the first partial derivatives:

$$\frac{\partial l(N,\theta)}{\partial N} = \Psi(N-y_1+1) + \Psi(N-y_2+1) - 2\Psi(N+1) + 2\log(1+e^{\theta}),$$

and

$$\frac{\partial l(N,\theta)}{\partial \theta} = \frac{2Ne^{\theta}}{1+e^{\theta}} - (y_1+y_2).$$

Now we derive expressions for the second order partial derivatives:

$$\begin{aligned}
\frac{\partial^2 l(N,\theta)}{\partial N^2} &= \Psi'(N-y_1+1) + \Psi'(N-y_2+1) - 2\Psi'(N+1), \\
\frac{\partial^2 l(N,\theta)}{\partial \theta^2} &= \frac{2Ne^{\theta}}{(1+e^{\theta})^2}, \text{ and} \\
\frac{\partial^2 l(N,\theta)}{\partial N \partial \theta} &= \frac{2e^{\theta}}{1+e^{\theta}}.
\end{aligned}$$

### Task 2

The function `myhessian` will construcct a 2x2 Hessian Matrix for $l(N, \theta)$ using the expressions derived for its second order partial derivatives above. The Hessian matrix will be given by:

$$\begin{bmatrix} \frac{\partial^2 l(N,\theta)}{\partial N^2} & \frac{\partial^2 l(N,\theta)}{\partial N \partial \theta} \\ \frac{\partial^2 l(N,\theta)}{\partial \theta \partial N} & \frac{\partial^2 l(N,\theta)}{\partial \theta^2} \end{bmatrix}$$

Below is the implementation of `myhessian`:

```
myhessian <- function(param, Y) {
  # Extract parameters
  N <- param[1]
  theta <- param[2]

  # Compute second order partial derivatives
  thetatwo <- 2 * N * exp(theta) / (1 + exp(theta))^2
  theta_n <- 2 * exp(theta) / (1 + exp(theta))
  ntwo <- psigamma(N - Y[1] + 1, 1) + psigamma(N - Y[2] + 1, 1) - 2 * psigamma(N + 1, 1)

  # Return Hessian
  return(matrix(c(ntwo, theta_n, theta_n, thetatwo), nrow = 2, ncol = 2))
}
```

Let us now use our MLEs $\hat{N}$ and $\hat{\theta}$ to compare the output of `myhessian` and `optimHess`.

```
# Find hessian using myhessian
myhess <- myhessian(bestopt$par, Y=Y)
```

The Hessian matrix $\mathbf{H}$ determined by `optimHess` is:

$$\begin{bmatrix} 0.008988309 & 1.270193 \\ 1.270192513 & 179.898086 \end{bmatrix}$$

The Hessian matrix $\mathbf{H}'$ determied by `myhessian` is:

$$\begin{bmatrix} 0.008988314 & 1.270193 \\ 1.270192559 & 179.898109 \end{bmatrix}$$

The matrix of relative differences between $\mathbf{H}$ and $\mathbf{H}'$ is:

$$\begin{bmatrix} 5.500008 \times 10^{-9} & 4.574974 \times 10^{-8} \\ 4.574974 \times 10^{-8} & 2.351907 \times 10^{-5} \end{bmatrix}$$

We see from the matrix of relative differences that our two computed Hessian matrices are almost identical. Indeed the largest differnt between two computed values is $2.351907 \times 10^{-5}$ in the value of $\frac{\partial^2 l(N,\theta)}{\partial \theta^2}$ and even this is extremely close to zero. The reason the two matrices are not exact is likely due to the fact that in `myhessian` we calculated each value directly from its expression whereas in `optimHess` these are estimated numerically.

**Task 3**

**Task 4**

## Question 3

**Task 1**

**Task 2**

**Task 3**

## Question 4

**Task 1**

**Task 2**

**Task 3**

**Task 4**

## Appendix