

Stat 3202 Lab 4

Jane Weissberg.11

2024-06-14

1a. The likelihood function of a normal distribution with the variance equal to 1 is $(2\pi)^{-\frac{n}{2}} e^{-\frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2}$.

1b.

```
L_Normal <- function(mu_domain, data) {  
  n <- length(data)  
  likelihoods <- numeric(length(mu_domain))  
  for (i in seq_along(mu_domain)) {  
    mu <- mu_domain[i]  
    L <- ((2*pi)^(-n/2)) * exp(-0.5 * sum((data - mu)^2))  
    likelihoods[i] <- L  
  }  
  return(likelihoods)  
}
```

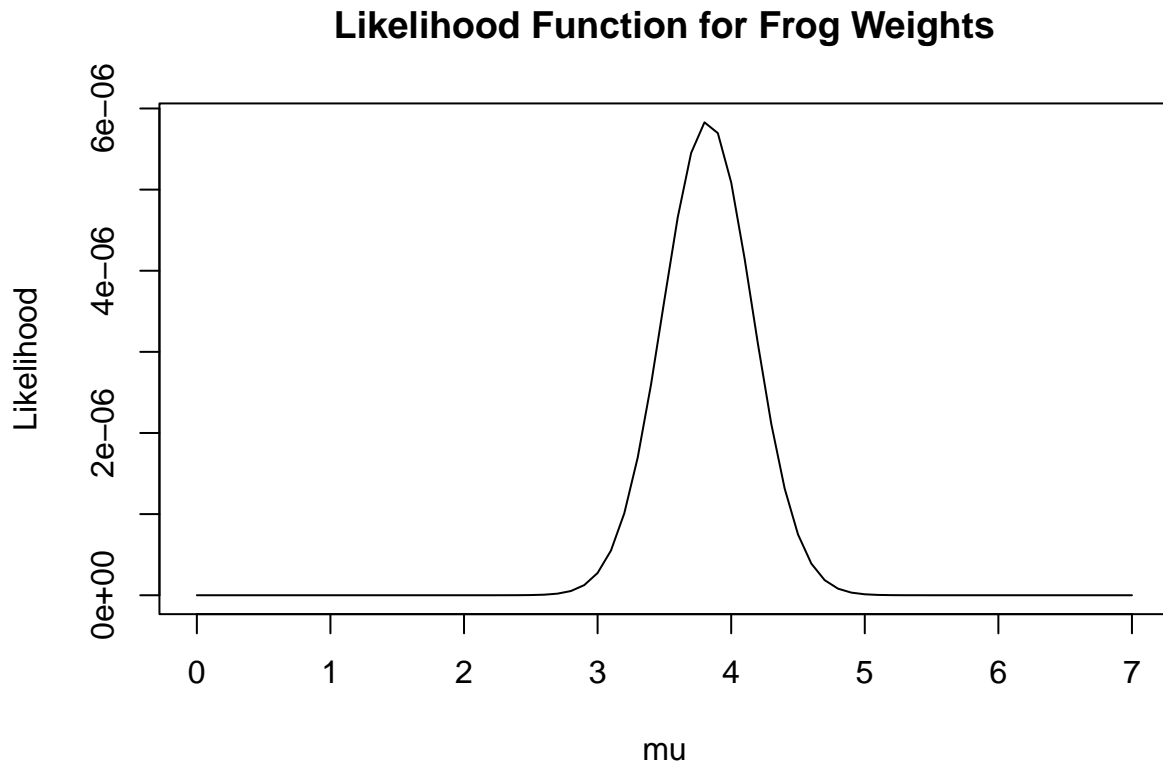
1c.

```
frogs <- c(4.57, 4.73, 2.24, 4.16, 4.53, 3.79, 2.48, 4.74, 3.18)  
mu_hat <- mean(frogs)  
print(mu_hat)
```

```
## [1] 3.824444
```

1d.

```
mu_domain <- seq(0, 7, by = 0.1)  
likelihoods <- L_Normal(mu_domain, frogs)  
plot(mu_domain, likelihoods, type = "l", xlab = "mu", ylab = "Likelihood", main = "Likelihood Function :")
```



This likelihood function has somewhat of a bell curve shape, with the maximum a little before $\mu=4$. I.e.

```
mle_result <- optimize(L_Normal, interval = c(0, 7), data = frogs, maximum = TRUE)
print(mle_result)
```

```
## $maximum
## [1] 3.824445
##
## $objective
## [1] 5.844802e-06
```

If. `$maximum` represents the `mu` value where the maximum of the function lies. `$objective` represents the likelihood at that `mu` value, which is the maximum likelihood.

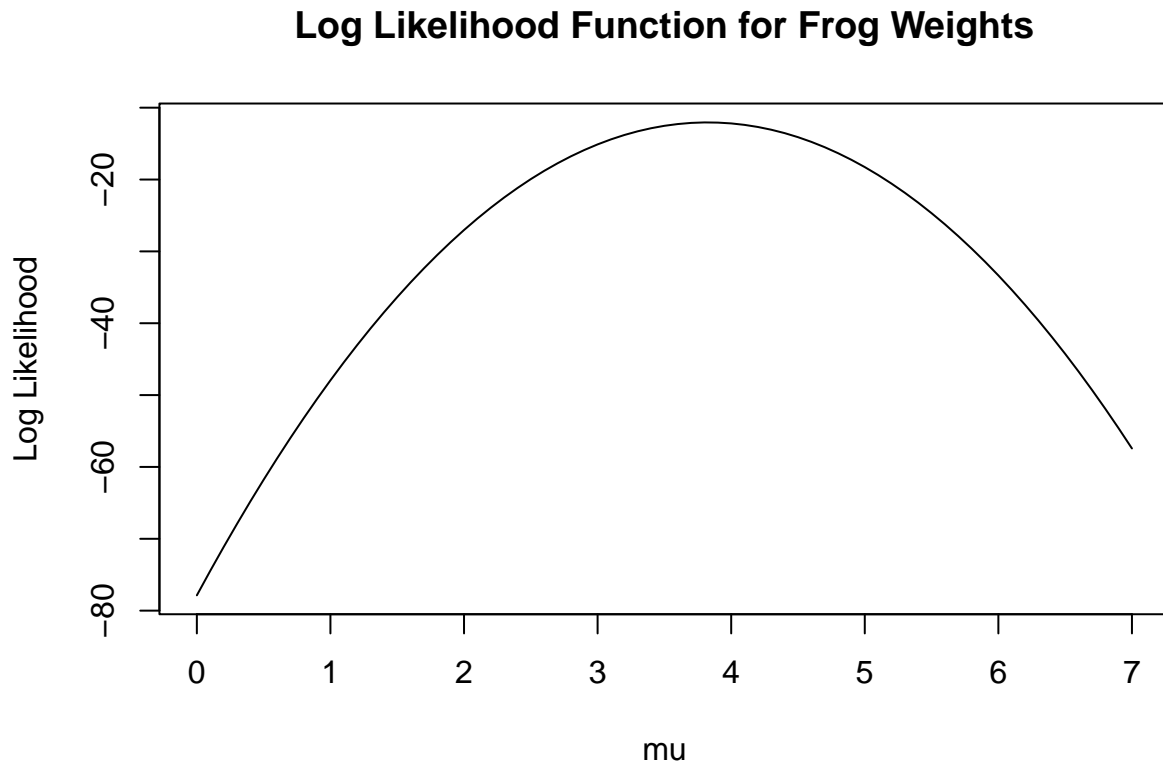
I.g.

```
log_L_Normal <- function(mu_domain, data) {
  n <- length(data)
  likelihoods <- numeric(length(mu_domain))
  for (i in seq_along(mu_domain)) {
    mu <- mu_domain[i]
    L <- ((-n/2)*log(2*pi) - 0.5*sum((data-mu)^2))
    likelihoods[i] <- L
  }
  return(likelihoods)
}
```

```

likelihoods <- log_L_Normal(mu_domain, frogs)
plot(mu_domain, likelihoods, type = "l", xlab = "mu", ylab = "Log Likelihood", main = "Log Likelihood F

```



1h.

```

mle_result <- optimize(log_L_Normal, interval = c(0, 7), data = frogs, maximum = TRUE)
print(mle_result)

```

```

## $maximum
## [1] 3.824444
##
## $objective
## [1] -12.04996

```

1i. The shapes of the likelihood function and the log-likelihood function are different because the logs produce different numbers and change the shape of the graph. However, the maximum is in the same spot because although the likelihood values are changing, they are still in the same order, taking the same values of μ . This means that the maximum will still fall at the same μ value, even though it is a different likelihood.

2a. $\lambda^n e^{-\lambda \sum_{i=1}^n x_i}$

2b.

```
L_Exp <- function(lambda_domain, data) {
  n <- length(data)
  likelihoods <- numeric(length(lambda_domain))
  for (i in seq_along(lambda_domain)) {
    lambda <- lambda_domain[i]
    L <- (lambda^n) * exp(-lambda * sum(data))
    likelihoods[i] <- L
  }
  return(likelihoods)
}
```

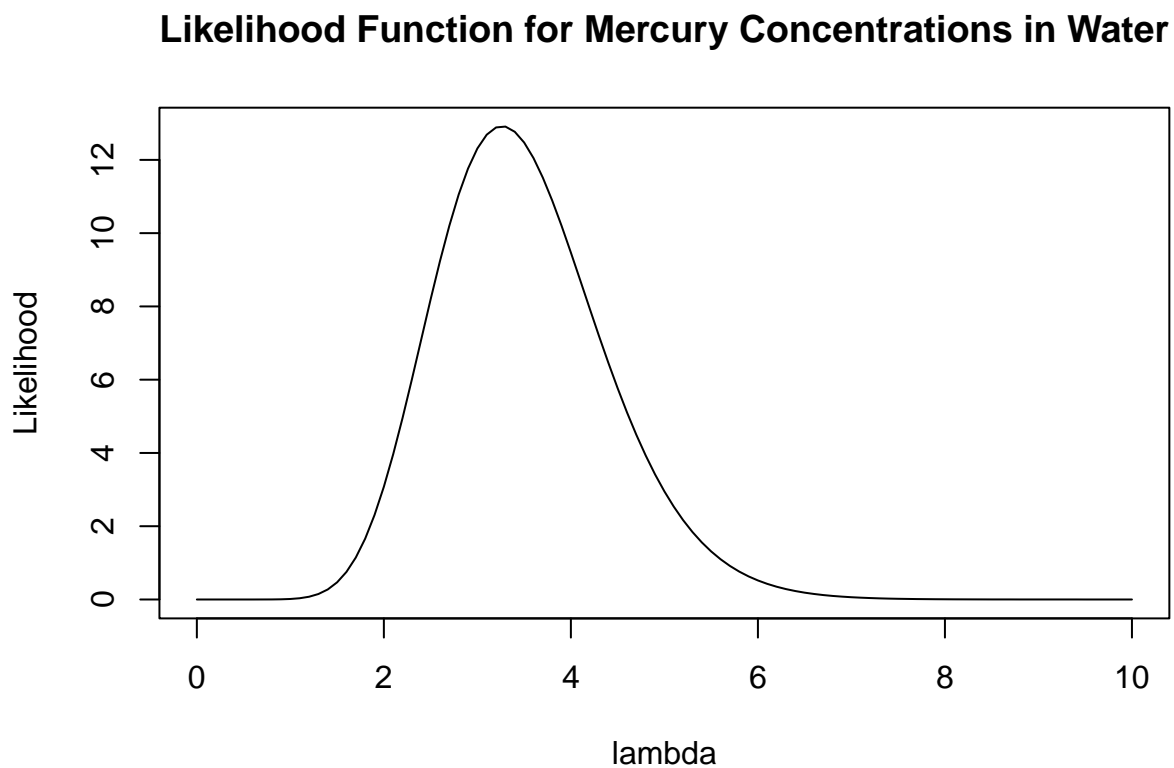
2c.

```
mercury <- c(0.51, 0.02, 0.15, 0.46, 0.11, 0.04, 0.39, 0.52, 0.2, 0.17, 0.01, 0.02, 0.32, 1.37)
xbar <- mean(mercury)
print(xbar)
```

```
## [1] 0.3064286
```

2d.

```
lambda_domain <- seq(0, 10, by = 0.1)
likelihoods <- L_Exp(lambda_domain, mercury)
plot(lambda_domain, likelihoods, type = "l", xlab = "lambda", ylab = "Likelihood", main = "Likelihood F
```



The likelihood function is somewhat of a bell curve shape, but slightly skewed right. The maximum seems to be between $\lambda=2$ and $\lambda=4$, closer to $\lambda=4$.

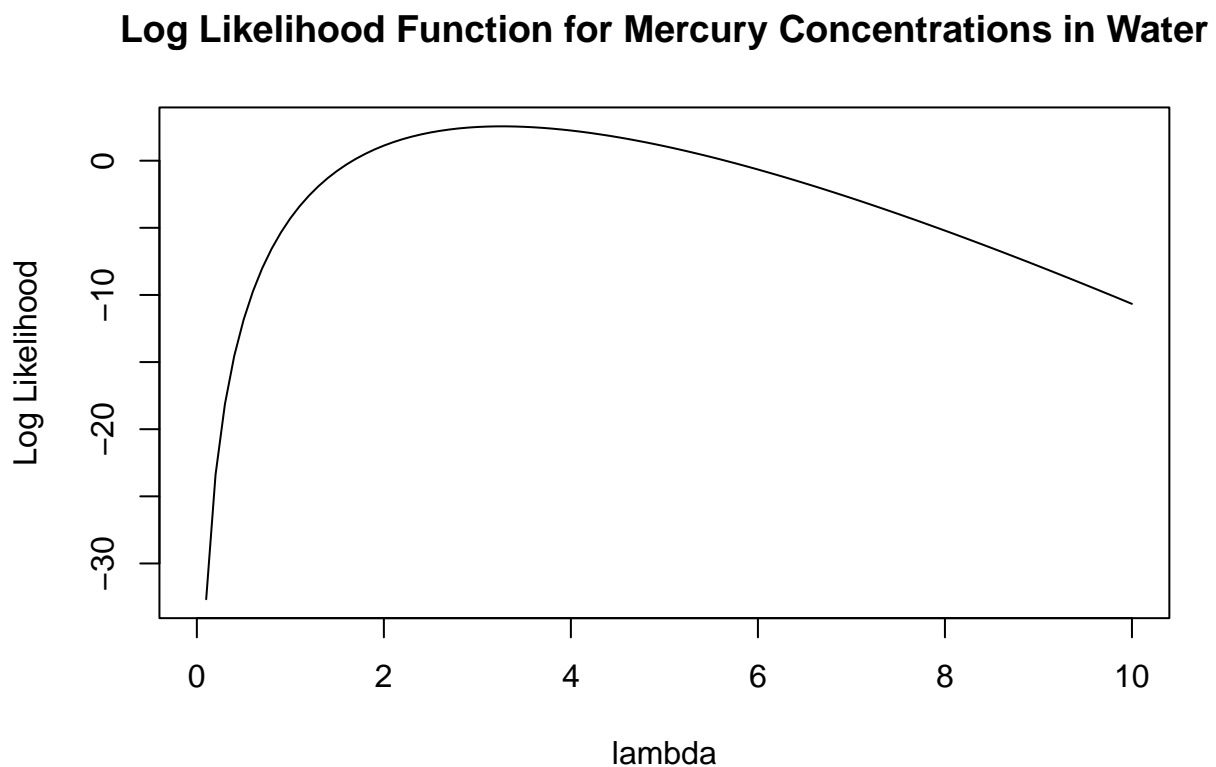
1e.

```
mle_result <- optimize(L_Exp, interval = c(0, 10), data = mercury, maximum = TRUE)
print(mle_result)
```

```
## $maximum
## [1] 3.263401
##
## $objective
## [1] 12.92015
```

1f.

```
log_L_Exp <- function(lambda_domain, data) {
  n <- length(data)
  likelihoods <- numeric(length(lambda_domain))
  for (i in seq_along(lambda_domain)) {
    lambda <- lambda_domain[i]
    L <- (n*log(lambda) - (lambda * sum(data)))
    likelihoods[i] <- L
  }
  return(likelihoods)
}
lambda_domain <- seq(0, 10, by = 0.1)
likelihoods <- log_L_Exp(lambda_domain, mercury)
plot(lambda_domain, likelihoods, type = "l", xlab = "lambda", ylab = "Log Likelihood", main = "Log Likelihood Function for Mercury Concentrations in Water")
```



```
mle_result <- optimize(log_L_Exp, interval = c(0, 10), data = mercury, maximum = TRUE)
print(mle_result)
```

```
## $maximum
## [1] 3.263402
##
## $objective
## [1] 2.558788
```

The location of the maximum of the log likelihood function is at $\lambda=3.26$ and $\log\text{-likelihood}=2.56$.

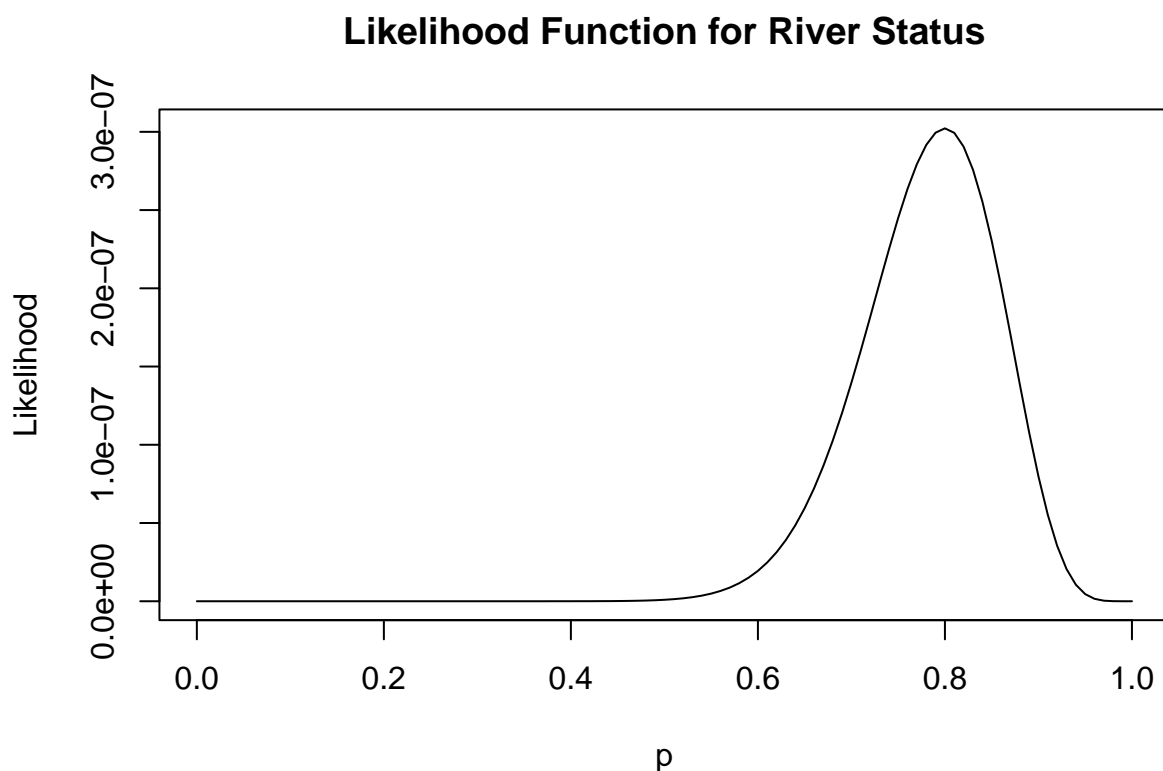
3.

```
L_Bern <- function(p, data) {
  n <- length(data)
  L <- (p^(sum(data=="Safe"))) * (1-p)^(sum(data=="Flood"))
  return(L)
}

p_domain <- seq(0, 1, by = 0.01)

river <- c("Safe", "Safe", "Safe", "Safe", "Safe",
"Safe", "Safe", "Safe", "Safe", "Flood",
"Safe", "Safe", "Safe", "Flood", "Safe",
"Safe", "Flood", "Flood", "Safe", "Safe",
"Flood", "Safe", "Safe", "Safe", "Safe",
"Safe", "Safe", "Safe", "Flood", "Safe")

likelihoods <- L_Bern(p_domain, river)
plot(p_domain, likelihoods, type = "l", xlab = "p", ylab = "Likelihood", main = "Likelihood Function for Bernoulli")
```



```
mle_result <- optimize(L_Bern, interval = c(0, 1), data = river, maximum = TRUE)
print(mle_result)
```

```
## $maximum
## [1] 0.7999833
##
## $objective
## [1] 3.022314e-07
```

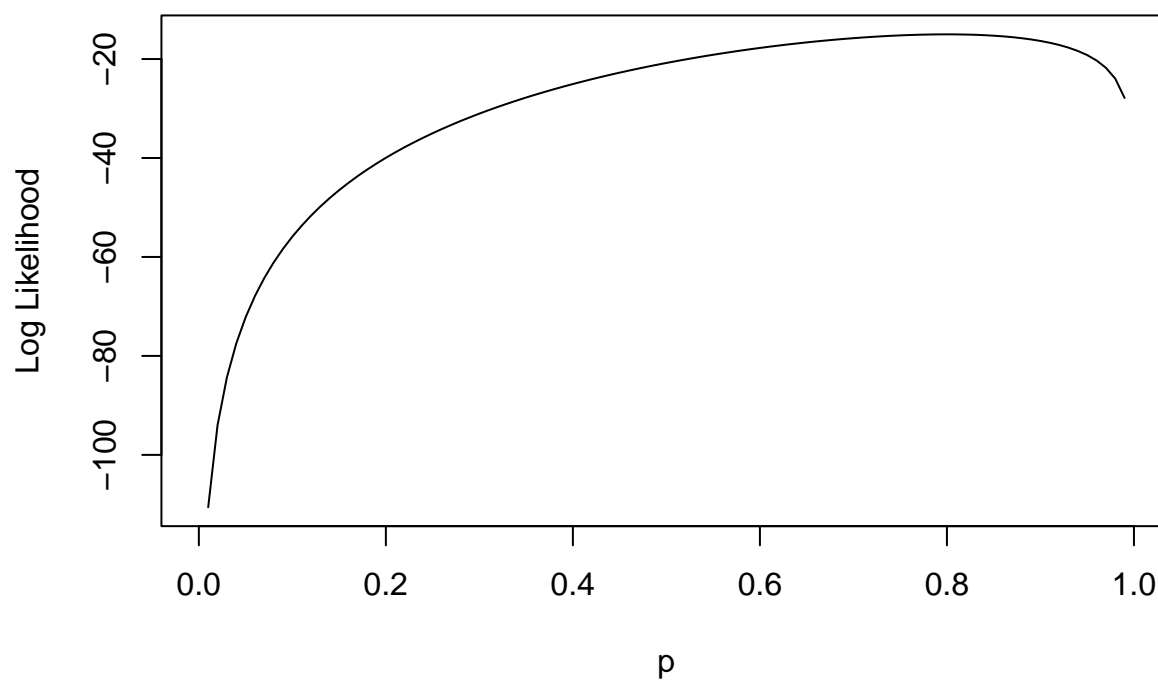
```
cat("This answer seems reasonable because there were 24 times that it was reported as safe out of 30, whi
```

```
## This answer seems reasonable because there were 24 times that it was reported as safe out of 30, whi
```

```
log_L_Bern <- function(p, data) {
  n <- length(data)
  L <- ((sum(data=="Safe")*log(p)) + (sum(data=="Flood")*log(1-p)))
  return(L)
}

likelihoods <- log_L_Bern(p_domain, river)
plot(p_domain, likelihoods, type = "l", xlab = "p", ylab = "Log Likelihood", main = "Log Likelihood Fun
```

Log Likelihood Function for River Status



```
mle_result <- optimize(log_L_Bern, interval = c(0, 1), data = river, maximum = TRUE)
print(mle_result)
```

```
## $maximum
## [1] 0.7999832
##
## $objective
## [1] -15.01207
```

```
cat("The maximum is located at p=0.7999 and log-likelihood=-15.012. This is at the same p value as the original estimate.")
```

```
## The maximum is located at p=0.7999 and log-likelihood=-15.012. This is at the same p value as the original estimate.
```