FISH621 - Homework 1

Marcel Gietzmann-Sanders

```
setwd("/workspaces/schooling/abundance/hwk_1/")
```

Problem 1

We'll start by loading in the data.

```
small_sample = read.table("data/Length Sample 50.csv", header=T, sep=",")['x'][,1]
large_sample = read.table("data/Length Sample 250.csv", header=T, sep=",")['x'][,1]
N = 1000
```

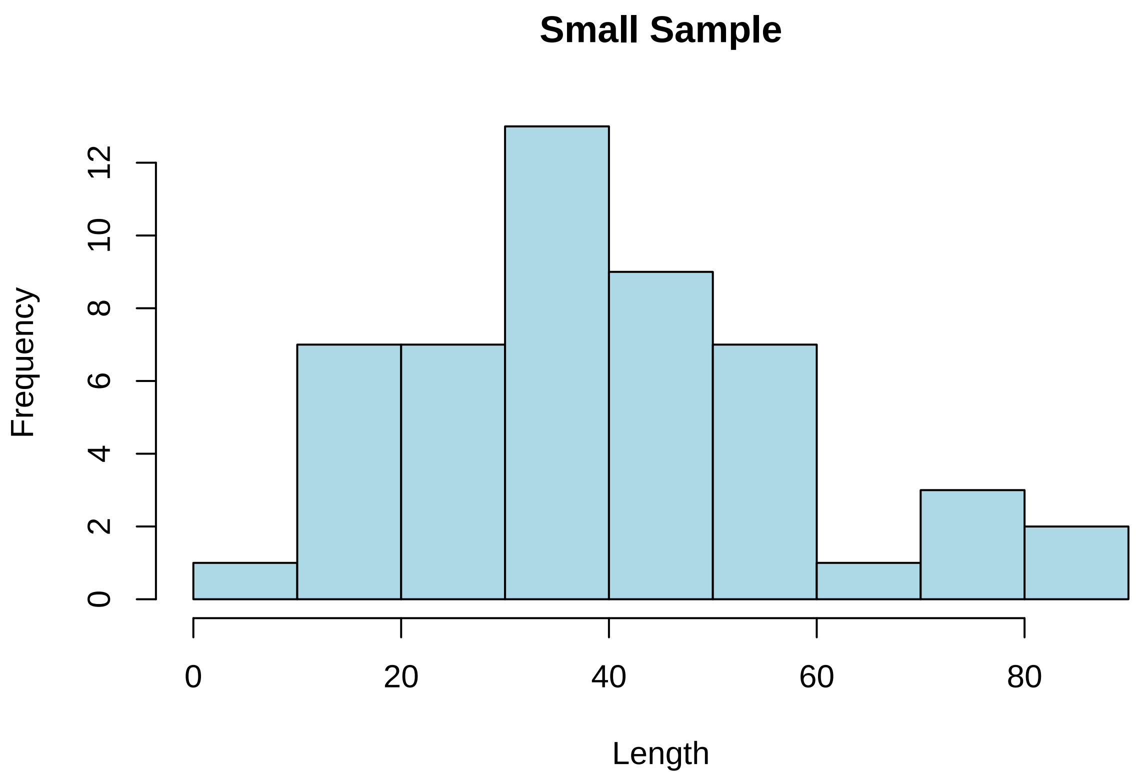
Then create the functions we'll use to answer the question for each sample.

```
variance_of_estimator = function(sample, N) {
    n = length(sample)
    return(
        (N - n) / N * var(sample) / n
    )
}
```

```
summarize <- function(sample, N) {
    print(paste("Sample Mean:", round(mean(sample), 2)))
    print(paste("Sample Variance:", round(var(sample), 2)))
    print(paste("Estimated Variance of Estimator:", round(variance_of_estimator(sample, N), 2)))
    print(paste("Estimated CV of Estimator:", round(variance_of_estimator(sample, N) / mean(sample), 2)
}
```

Small Sample

```
hist(small_sample, main="Small Sample", xlab="Length", ylab="Frequency", col="lightblue")
```



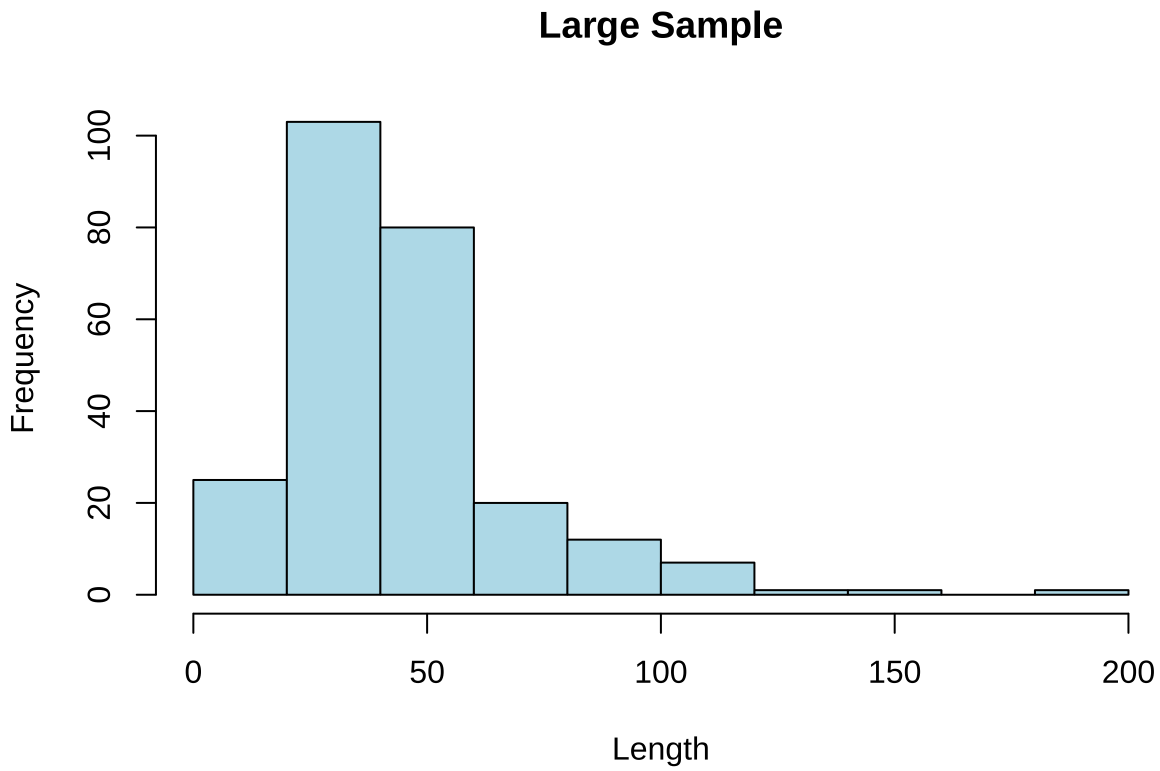
```
summarize(small_sample, N)
```

```
## [1] "Sample Mean: 39.96"
## [1] "Sample Variance: 353.14"
## [1] "Estimated Variance of Estimator: 6.71"
## [1] "Estimated CV of Estimator: 0.17"
```

Large Sample

I'm going to make the assumption that the first sample of 50 that we took was returned to the population before we ended up taking the second sample. If that is not the case one just need to replace $N = 1000$ with $N = 950$ in the following code.

```
hist(large_sample, main="Large Sample", xlab="Length", ylab="Frequency", col="lightblue")
```



```
summarize(large_sample, N)
```

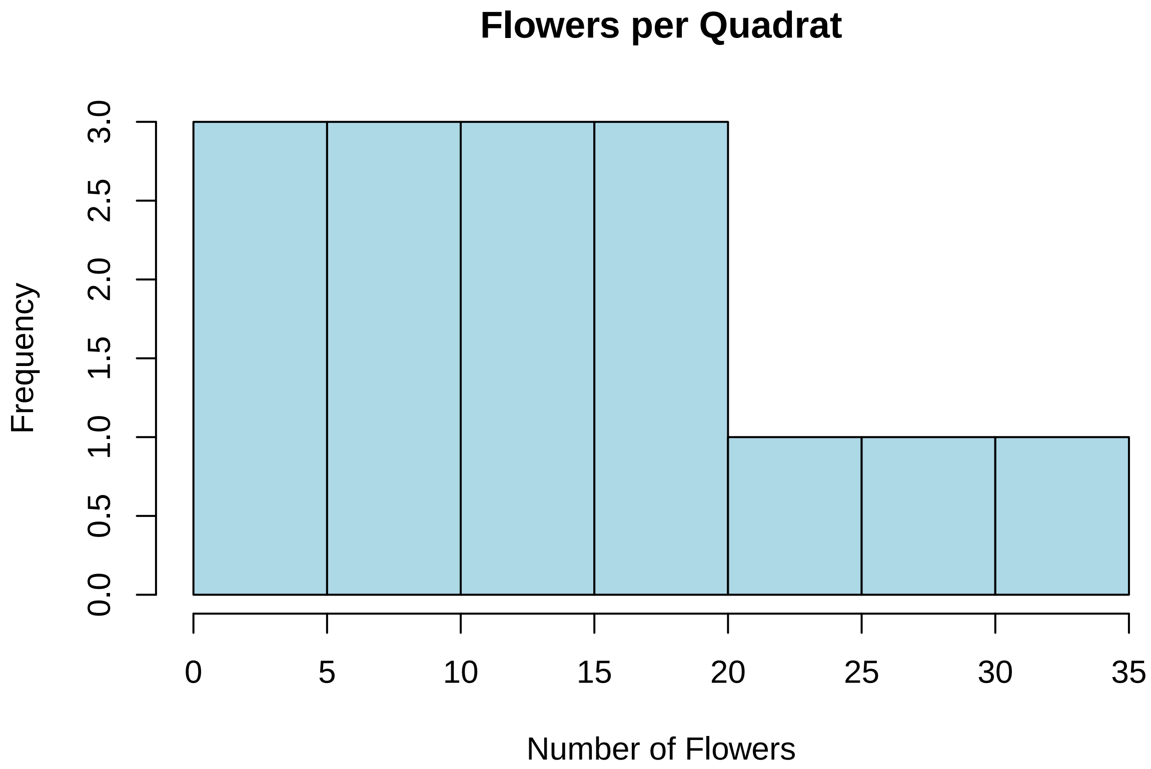
```
## [1] "Sample Mean: 44.46"
## [1] "Sample Variance: 638.16"
## [1] "Estimated Variance of Estimator: 1.91"
## [1] "Estimated CV of Estimator: 0.04"
```

Clearly we can see that in the larger sample, while we did end up with more sample variance (due to the extremely long tail), the variance in our estimator is much lower. This is due to our much larger sample size.

Problem 2

In this case our data is:

```
sample = c(10, 15, 1, 23, 18, 12, 19, 9, 5, 8, 3, 26, 20, 12, 31)
hist(sample, main="Flowers per Quadrat", xlab="Number of Flowers", ylab="Frequency", col="lightblue")
```



Given this is a mountain field and 15 square meters is not a lot of area we can assume that the finite population correction will be ≈ 1. We can acheive this by just setting N to a very large number (compared to our 15 segments).

From the second part of the question we know that our field is $50 \cdot 120 = 6000$ meters square. So we'll just use that as our large N value.

We therefore get the following statistics for the number of flowers per quadrat.

```
N = 6000
summarize(sample, N)
```

```
## [1] "Sample Mean: 14.13"
## [1] "Sample Variance: 74.84"
## [1] "Estimated Variance of Estimator: 4.98"
## [1] "Estimated CV of Estimator: 0.35"
```

To get the estimates of total population is now quite simple.

```
N_est = N * mean(sample)
N_est_var = N^2 * variance_of_estimator(sample, N)
N_est_cv = N_est_var / N_est
print(paste("Estimated Total Population:", round(N_est, 2)))
```

```
## [1] "Estimated Total Population: 84800"
```

```
print(paste("Estimated Variance of Estimator:", round(N_est_var, 2)))
```

```
## [1] "Estimated Variance of Estimator: 179162400"
```

```r
print(paste("Estimated CV of Estimator:", round(N_est_cv, 2)))
```

```
## [1] "Estimated CV of Estimator: 2112.76"
```

All in all we really know nothing about the number of flowers in this field... We'd need to take far more samples.

Problem 3

We'll start by assuming that each vole is independently distributed with an equal likelihood of appearing anywhere on the apartment floor. This would then mean that the probability of seeing a particular vole on the 15% of the floor that we can see is... well... 15%. Therefore, on average, we'd expect to see 15% of the voles.

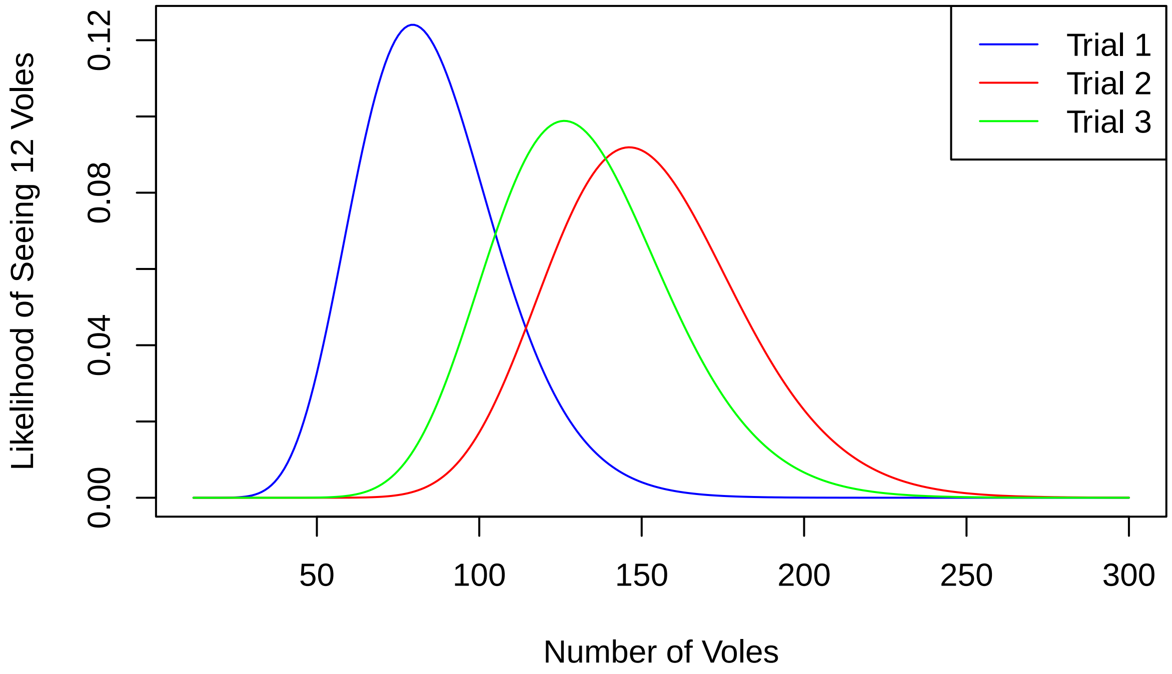
That means that we estimate respectively:

1. $12/0.15 = 80$
2. $22/0.15 \approx 147$
3. $19/0.15 \approx 127$

voles in each of our sampling events.

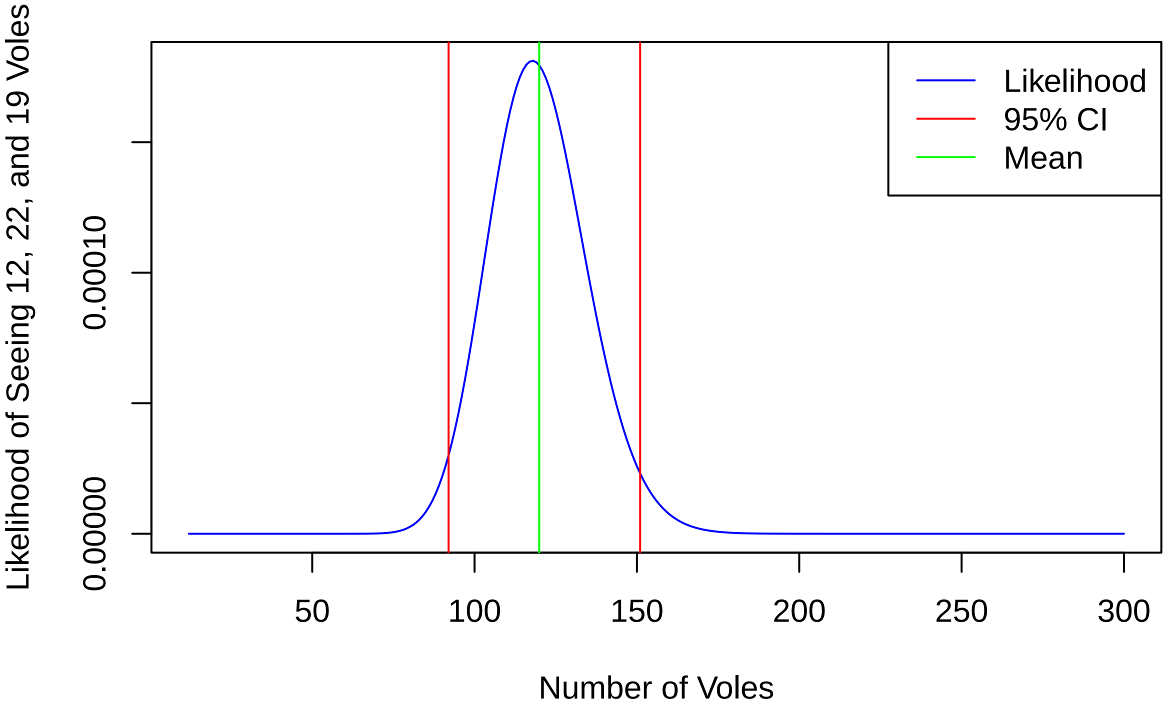
However we can model all of this using a binominal distribution where $p = 0.15$ and the number of trials is the number of voles overall N. Then n, the number of successes, is the number of voles we see. We can use this to estimate the likelihood of seeing n voles given N voles to get a sense of how certain our estimates are.

```r
N.possible = 12:300
likelihood.trial1 = dbinom(12, N.possible, 0.15)
likelihood.trial2 = dbinom(22, N.possible, 0.15)
likelihood.trial3 = dbinom(19, N.possible, 0.15)

plot(
    N.possible, likelihood.trial1,
    type="l", xlab="Number of Voles",
    ylab="Likelihood of Seeing 12 Voles",
    col="blue"
)
lines(
    N.possible, likelihood.trial2,
    type="l", col="red"
)
lines(
    N.possible, likelihood.trial3,
    type="l", col="green"
)
legend("topright", legend=c("Trial 1", "Trial 2", "Trial 3"), col=c("blue", "red", "green"), lty=1)
```



We can even multiply the likelihoods and see how estimates of N explain our overall data.

```
likelihood = likelihood.trial1 * likelihood.trial2 * likelihood.trial3
cdf = cumsum(likelihood) / sum(likelihood)
lower = N.possible[cdf > 0.025][1]
upper = N.possible[cdf > 0.975][1]

plot(
    N.possible, likelihood,
    type="l", xlab="Number of Voles",
    ylab="Likelihood of Seeing 12, 22, and 19 Voles",
    col="blue"
)
abline(v=lower, col="red")
abline(v=upper, col="red")
abline(v=sum(likelihood * N.possible) / sum(likelihood), col="green")

legend("topright", legend=c("Likelihood", "95% CI", "Mean"), col=c("blue", "red", "green"), lty=1)
```



```
print(paste("95% CI:", lower, upper))
```

```
## [1] "95% CI: 92 151"
```

```
print(paste("Mean:", sum(likelihood * N.possible) / sum(likelihood)))
```

```
## [1] "Mean: 119.908855550388"
```

According to this we could have anywhere from 92 to 151 voles. That's a lot of voles…

However we are assuming that the voles are uniformly and independently distributed across the floor, which, if we knew more about voles, may not be a good assumption.

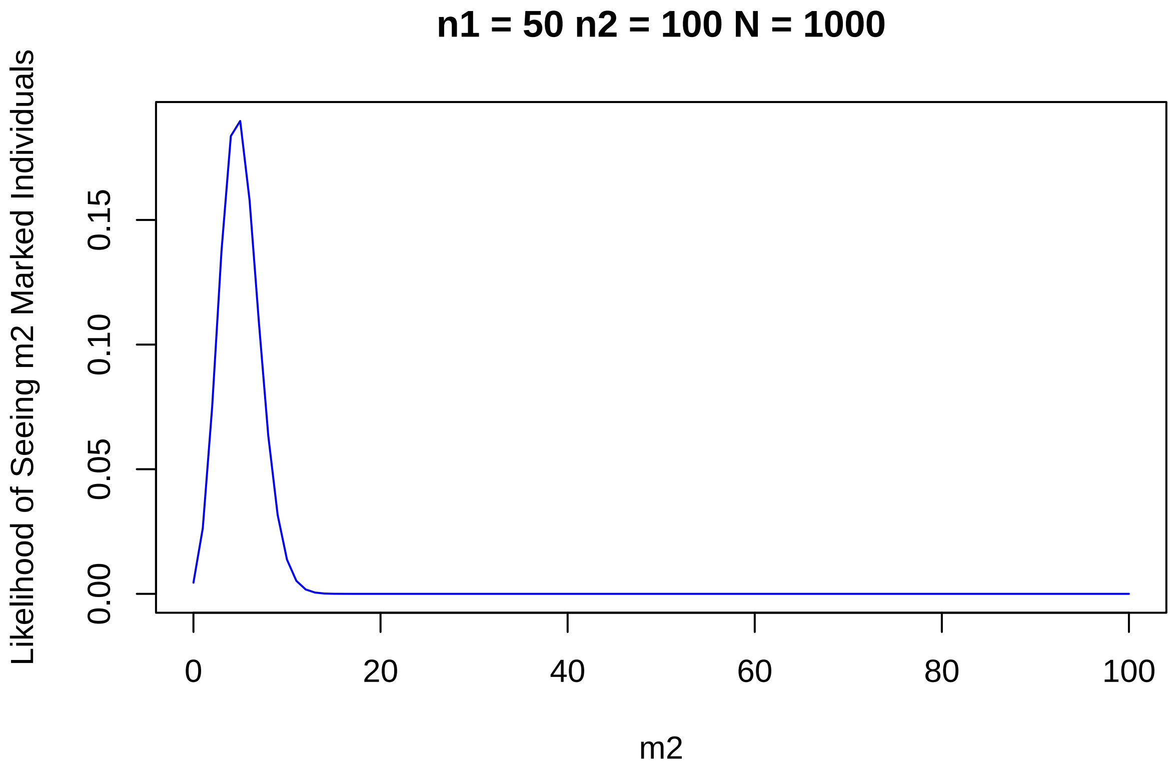
Problem 4

Given we marked 50 individuals out of 1000 our mark fraction $p_1 = 50/1000 = 0.05$ So our expected number of marked individuals in the second sample is $m_2 = 0.05 \cdot 100 = 5$.

However let's play around with the Hypergeometric distribution to convince ourselves of this.

```
summarize = function(n1, n2, N) {
    m2 = 0:n2
    likelihood = dhyper(m2, n1, N-n1, n2)
    plot(
        m2, likelihood,
        type="l", xlab="m2",
        ylab="Likelihood of Seeing m2 Marked Individuals",
        col="blue",
        main=paste("n1 =", n1, "n2 =", n2, "N =", N)
    )
    print(paste("Mean:", sum(likelihood * m2) / sum(likelihood)))
}
```

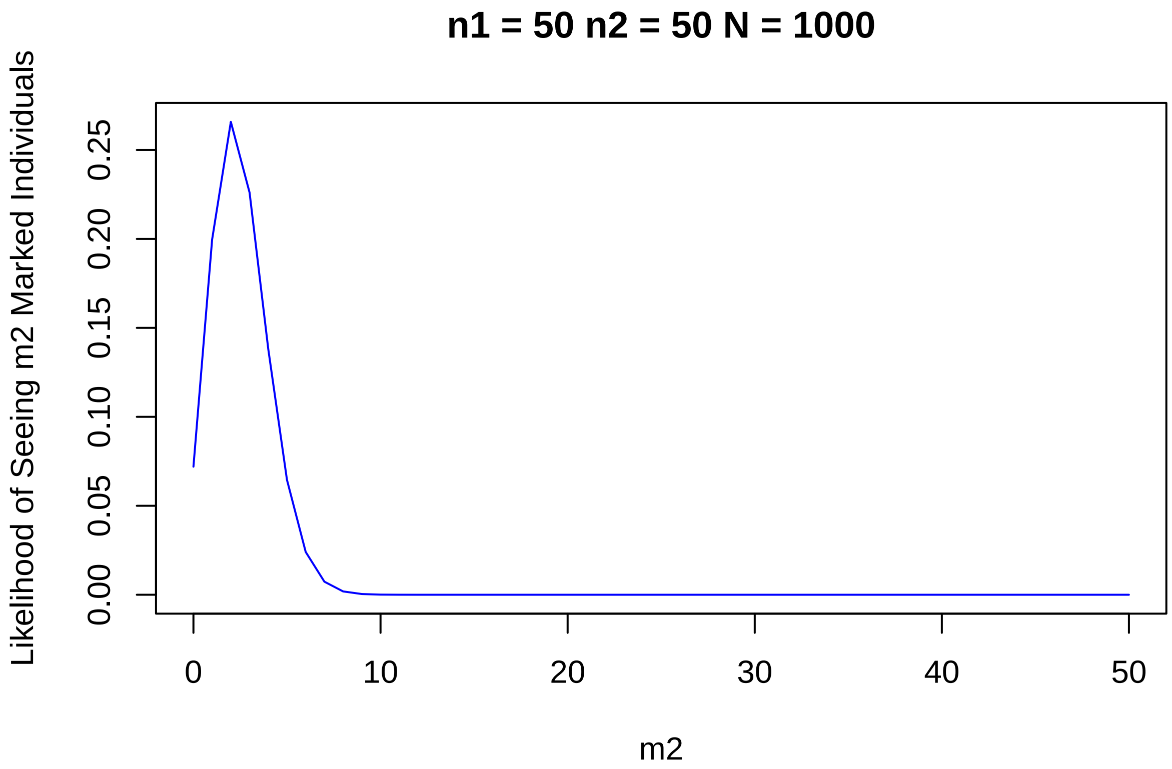
```
n1 = 50
n2 = 100
N = 1000
summarize(n1, n2, N)
```



```
## [1] "Mean: 5"
```

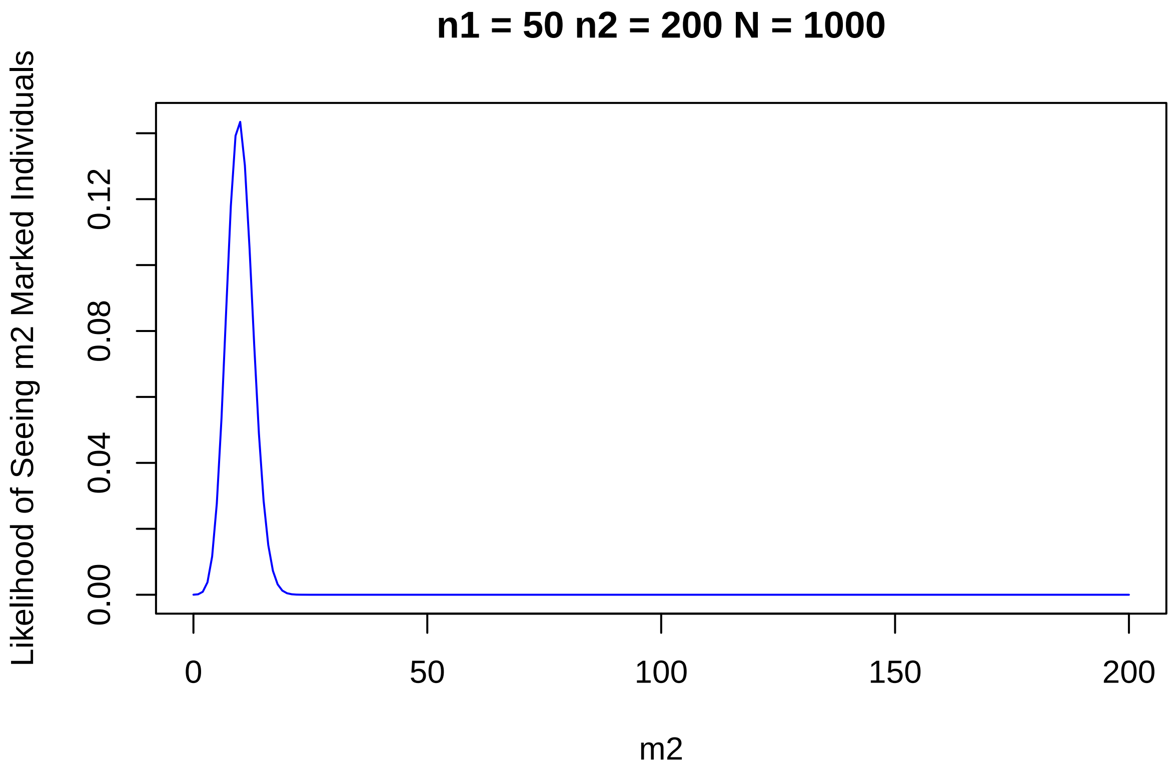
Case 1: $n1 = 50$ **vs** $n1 = 200$

```
summarize(n1, 50, N)
```



```
## [1] "Mean: 2.5"
```

```
summarize(n1, 200, N)
```

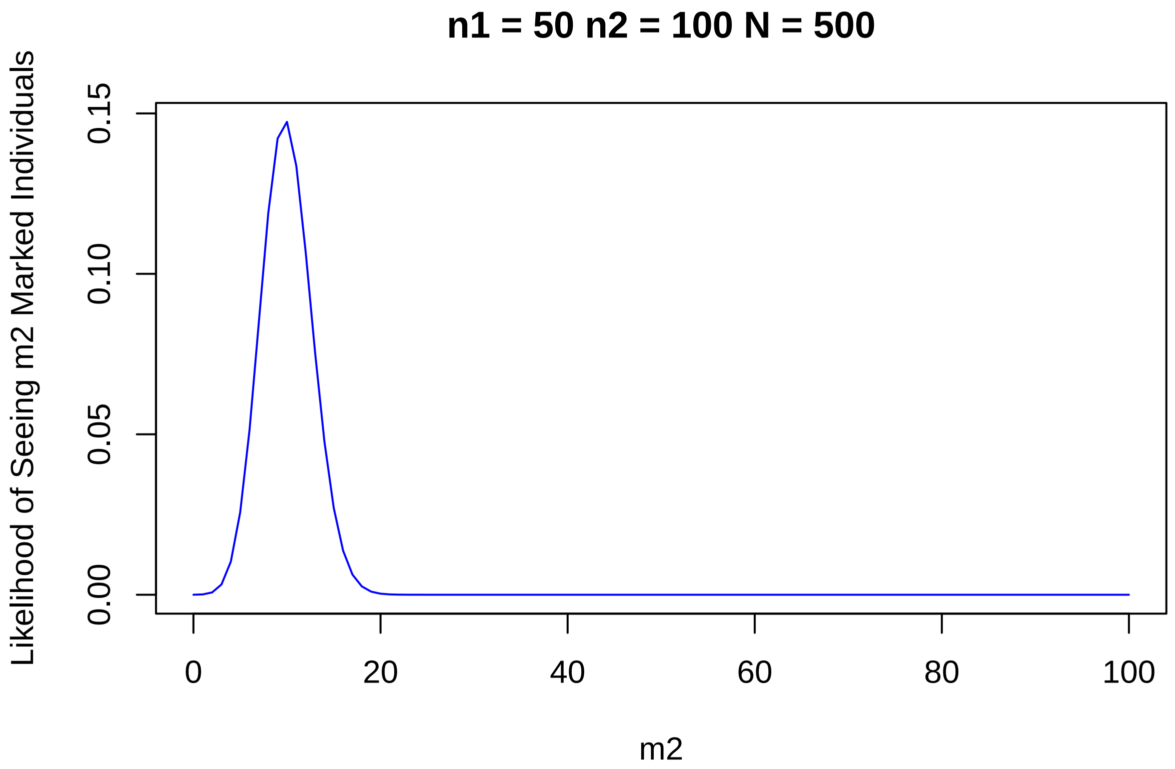


```
## [1] "Mean: 10"
```

If we decrease or increase the number of whales in the second sample we will expect to see proportionally fewer or more marked whales.

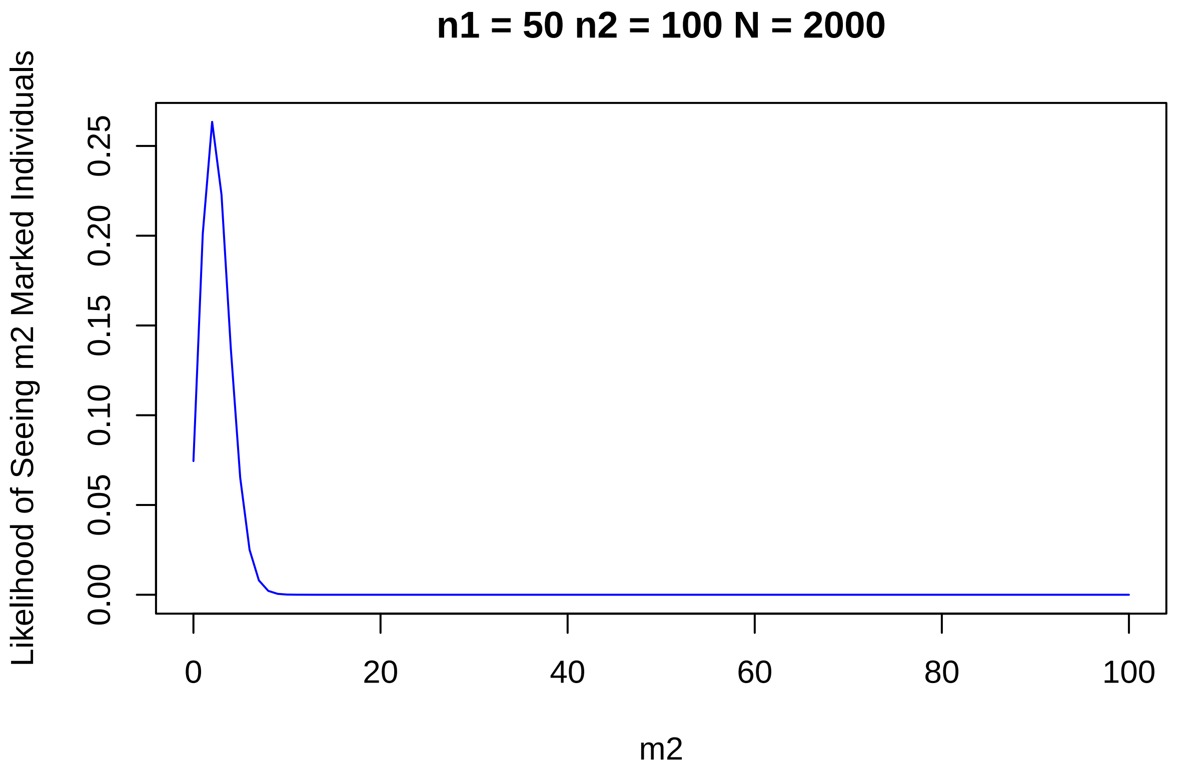
Case 2: $N = 500$ vs $N = 2000$

```
summarize(n1, n2, 500)
```



```
## [1] "Mean: 10"
```

```r
summarize(n1, n2, 2000)
```



```
## [1] "Mean: 2.5"
```

If N increases our $p1$ will decrease and therefore we will expect to see fewer marked whales. If N decreases the opposite will be true.

Problem 5

```
data = rbind(
    c("May", 65, 120, 19),
    c("June", 25, 150, 12),
    c("July", 40, 110, 15),
    c("August", 45, 80, 35),
    c("September", 60, 70, 42)
)
colnames(data) = c("Month", "n1", "n2", "m2")
data = as.data.frame(data)
data$n1 = as.numeric(data$n1)
data$n2 = as.numeric(data$n2)
data$m2 = as.numeric(data$m2)
data
```

```
##       Month n1  n2 m2
## 1       May 65 120 19
## 2      June 25 150 12
## 3      July 40 110 15
## 4    August 45  80 35
## 5 September 60  70 42
```

Sampling Without Replacement

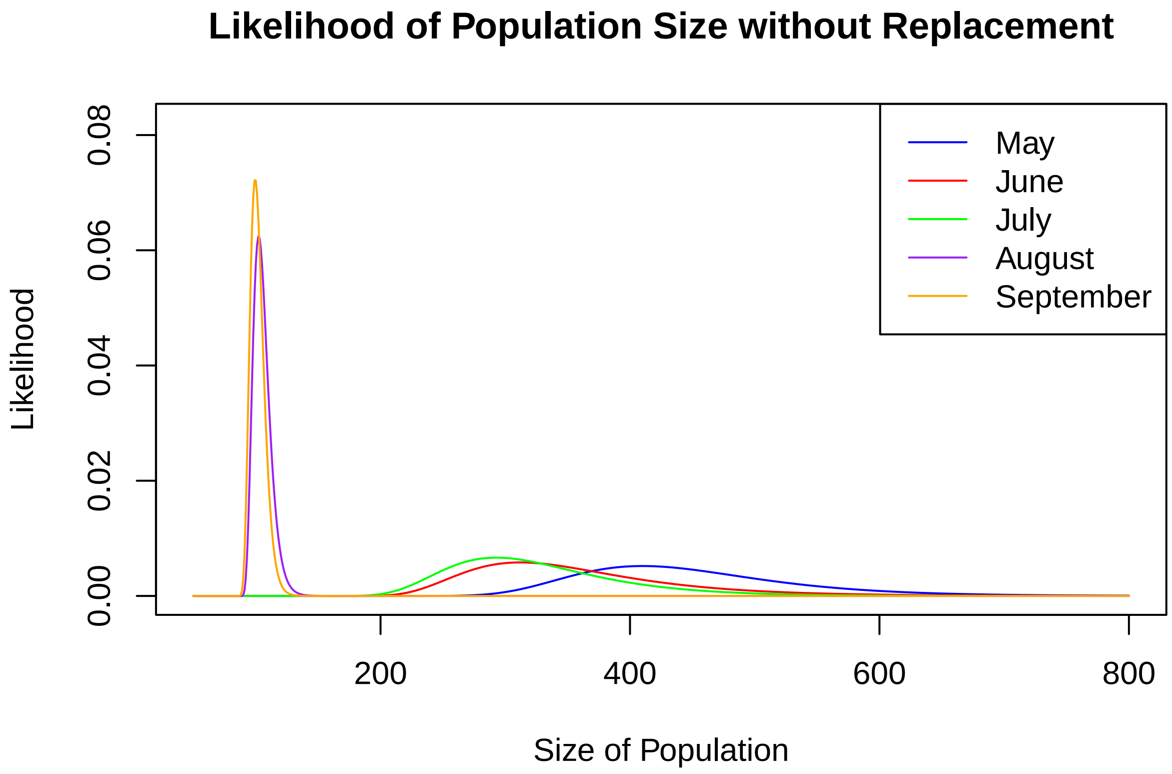
```
data$N.chap = (data$n1 + 1) * (data$n2 + 1) / (data$m2 + 1) - 1
data$var.N.chap = (
    (data$n1 + 1) * (data$n2 + 1) * (data$n1 - data$m2) * (data$n2 - data$m2)
    / (data$m2 + 1)^2 / (data$m2 + 2)
)
data$cv.N.chap = data$var.N.chap / data$N.chap
data
```

```
##       Month n1  n2 m2    N.chap var.N.chap  cv.N.chap
## 1       May 65 120 19 398.30000 4417.01857 11.0896776
## 2      June 25 150 12 301.00000 2976.85714  9.8898908
## 3      July 40 110 15 283.43750 2483.59949  8.7624238
## 4    August 45  80 35 102.50000   34.96622  0.3411338
## 5 September 60  70 42  99.72093   26.83052  0.2690561
```

```
N.trial = 50:800
months = c("May", "June", "July", "August", "September")
likelihoods = c()
for (i in 1:nrow(data)) {
    likelihood = dhyper(data$m2[i], data$n1[i], N.trial - data$n1[i], data$n2[i])
    likelihood[is.na(likelihood)] = 0
    likelihood = likelihood / sum(likelihood)
    likelihoods = rbind(likelihoods, likelihood)
}
```

```
## Warning in dhyper(data$m2[i], data$n1[i], N.trial - data$n1[i], data$n2[i]):
## NaNs produced

## Warning in dhyper(data$m2[i], data$n1[i], N.trial - data$n1[i], data$n2[i]):
## NaNs produced

## Warning in dhyper(data$m2[i], data$n1[i], N.trial - data$n1[i], data$n2[i]):
## NaNs produced

## Warning in dhyper(data$m2[i], data$n1[i], N.trial - data$n1[i], data$n2[i]):
## NaNs produced

## Warning in dhyper(data$m2[i], data$n1[i], N.trial - data$n1[i], data$n2[i]):
## NaNs produced
```

```
y_max = max(likelihoods) + 0.01
plot(
    N.trial, likelihoods[1,],
    type="l", xlab="Size of Population",
    ylab="Likelihood",
    col="blue",
    main="Likelihood of Population Size without Replacement",
    ylim=c(0, y_max)
)
lines(
    N.trial, likelihoods[2,],
    type="l", col="red"
)
lines(
    N.trial, likelihoods[3,],
    type="l", col="green"
)
lines(
    N.trial, likelihoods[4,],
    type="l", col="purple"
)
lines(
    N.trial, likelihoods[5,],
    type="l", col="orange"
)
legend("topright", legend=months, col=c("blue", "red", "green", "purple", "orange"), lty=1)
```



Sampling With Replacement

```
data$N.bailey = data$n1 * (data$n2 + 1) / (data$m2 + 1)
data$var.N.bailey = (
    data$n1 * data$n1 * (data$n2 + 1) * (data$n2 - data$m2)
    / (data$m2 + 1)^2 / (data$m2 + 2)
)
data$cv.N.bailey = data$var.N.bailey / data$N.bailey
data
```

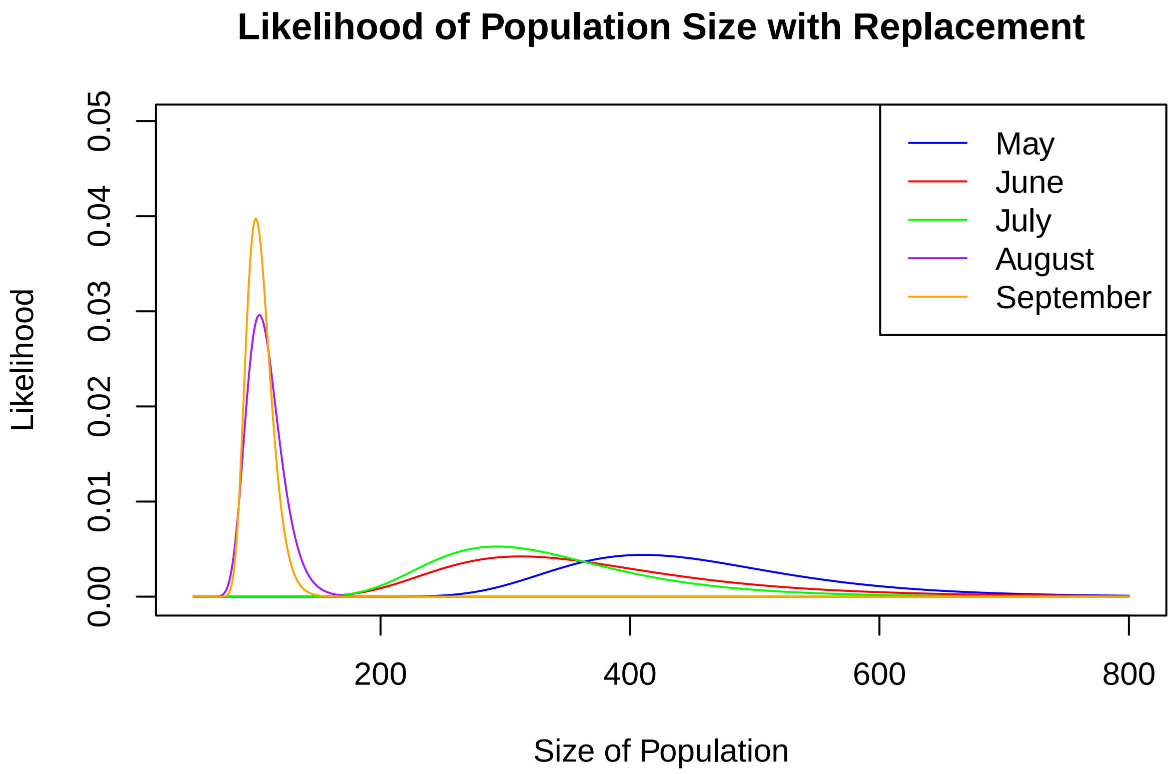
```
##       Month n1  n2 m2    N.chap var.N.chap  cv.N.chap  N.bailey var.N.bailey
## 1       May 65 120 19 398.30000 4417.01857 11.0896776 393.25000   6146.87202
## 2      June 25 150 12 301.00000 2976.85714  9.8898908 290.38462   5504.54353
## 3      July 40 110 15 283.43750 2483.59949  8.7624238 277.50000   3876.83824
## 4    August 45  80 35 102.50000   34.96622  0.3411338 101.25000    153.92736
## 5 September 60  70 42  99.72093   26.83052  0.2690561  99.06977     87.96893
##   cv.N.bailey
## 1  15.6309524
## 2  18.9560440
## 3  13.9705882
## 4   1.5202703
## 5   0.8879493
```

```
N.trial = 50:800
months = c("May", "June", "July", "August", "September")
likelihoods = c()
for (i in 1:nrow(data)) {
    likelihood = dbinom(x=data$m2[i], size=data$n2[i], prob=data$n1[i] / N.trial)
    likelihood[is.na(likelihood)] = 0
    likelihood = likelihood / sum(likelihood)
    likelihoods = rbind(likelihoods, likelihood)
}
```

```
## Warning in dbinom(x = data$m2[i], size = data$n2[i], prob =
## data$n1[i]/N.trial): NaNs produced

## Warning in dbinom(x = data$m2[i], size = data$n2[i], prob =
## data$n1[i]/N.trial): NaNs produced
```

```
y_max = max(likelihoods) + 0.01

plot(
    N.trial, likelihoods[1,],
    type="l", xlab="Size of Population",
    ylab="Likelihood",
    col="blue",
    main="Likelihood of Population Size with Replacement",
    ylim=c(0, y_max)
)
lines(
    N.trial, likelihoods[2,],
    type="l", col="red"
)
lines(
    N.trial, likelihoods[3,],
    type="l", col="green"
)
lines(
    N.trial, likelihoods[4,],
    type="l", col="purple"
)
lines(
    N.trial, likelihoods[5,],
    type="l", col="orange"
)
```

```
legend("topright", legend=months, col=c("blue", "red", "green", "purple", "orange"), lty=1)
```



Assuming this is a simple mark recapture experiment then there is clearly a loss in fish over time, specifically between the first three months and the last two. This inference is coming from the fact that our marked fraction in the second sample is a much higher proportion in the last two months.

However, there is a key problem here. We are assuming that tags are unique to the month which means that all tags would need to be lost or all tagged fish would need to die each month. This is probably unlikely (although I admit I don't know a lot about tag loss rates). What this means is that it is more likely that the fish we are catching at each time point have tags from earlier time steps.

That being said in this problem it's stated that I am marking all the fish I catch each time which seems to imply that in this scenario we are assuming tags are lost each month. That however brings in the question of how many tags I'd lose in the time between sample one and sample two.

Also simple mark recapture assumes there's no ingress or egress. Given we're looking at a stream that connects to a lake this is probably not a terribly good assumption either.

All in all then I think there's a reasonable bit of uncertainty in the validity of these results.

Time Allocation

I spent around 3 hours on this assignment.