kh694_Hw3

Kate Harline

2/16/2019

Quantitative Genomics and Genetics - Spring 2019
BTRY 4830/6830; PBSB 5201.01

Homework 3 (version 1)

Assigned February 12; Due 11:59PM February 16

Problem 1 (Easy)

a. Yes, this is a valid estimator. Any function that returns a value for a parameter is technically an estimator. However, creating an estimator that correctly represents the underlying probability distribution for the data is more difficult. $\hat{p} = 0.5$ would be correct if the coin is fair.

b. This would calculate the $\hat{p}$ based on the actual observations from the experiment, rather than relying on the underlying assumption that the experiment properly follows a Bernoulli distribution with $p = 0.5$, i.e. the coin is fair.

Problem 2 (Medium)

a.

```
library(ggplot2)
```

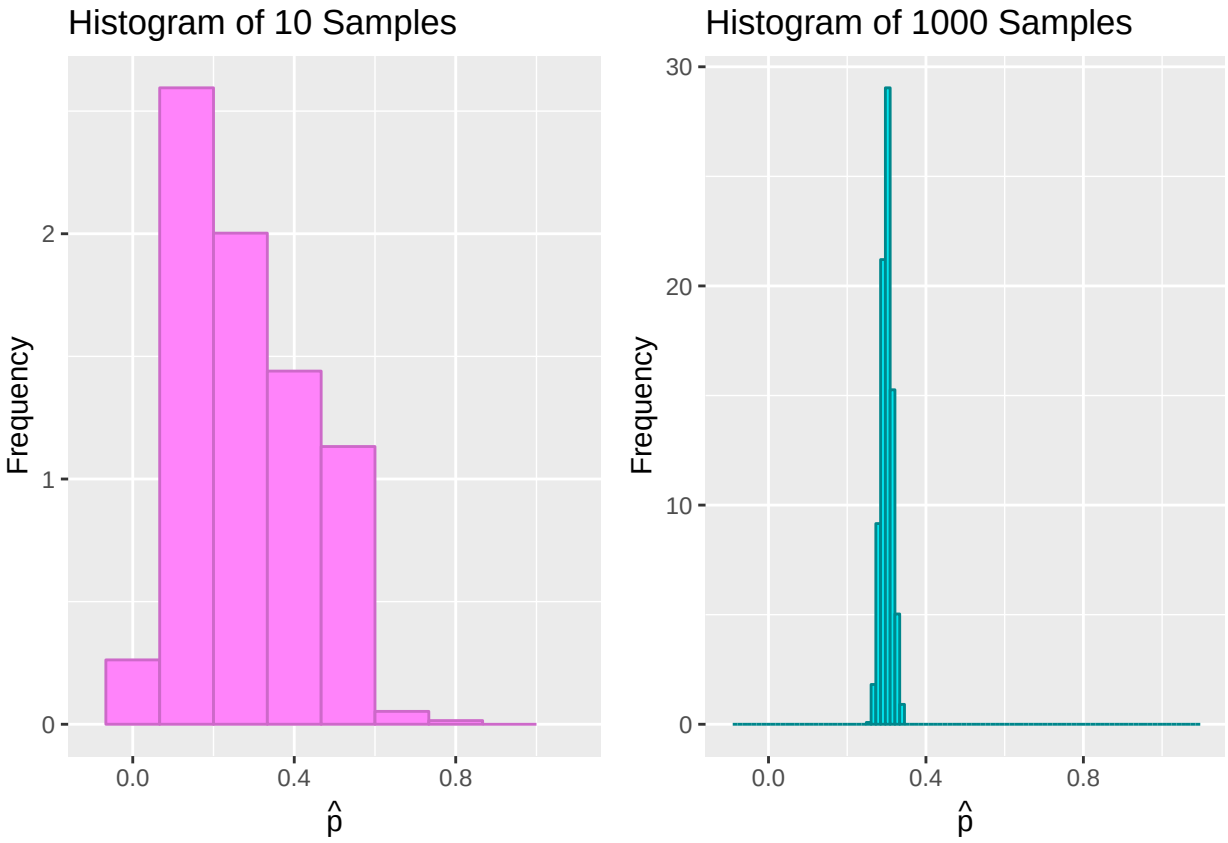
```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
library(gridExtra)

make_bernoullis <- function(M, n, p){
  # create vector to store means for each experiment
  means <- vector()
  for (i in 1:M){
    # make sample
    s <- rbinom(n, 1, p)
    # find mean
    sample_mean <- mean(s)
    # store sample mean
    means <- c(means, sample_mean)
  }
  return(means)
}
# M=1000,n=10,p=0.3
sim1 <- make_bernoullis(1000, 10, 0.3)
# M=1000,n=1000,p=0.3
sim2 <- make_bernoullis(1000, 1000, 0.3)
# plot hists
df_sim1 <- data.frame(sim1)
df_sim2 <- data.frame(sim2)
hist1 <- ggplot(df_sim1, aes(sim1)) +
  geom_histogram(aes(y=..density..),
```

```
                 bins = 10,
                 col = 'orchid3',
                 fill = 'orchid1') +
  labs(title = 'Histogram of 10 Samples', x = expression(hat(p)), y = 'Frequency') + xlim(-0.1, 1.1)
hist2 <- ggplot(df_sim2, aes(sim2)) +
  geom_histogram(aes(y=..density..),
                 bins = 100,
                 col = 'turquoise4',
                 fill = 'turquoise2') +
  labs(title = 'Histogram of 1000 Samples', x = expression(hat(p)), y = 'Frequency') + xlim(-0.1, 1.1)
grid.arrange(hist1, hist2, nrow = 1, ncol = 2)
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

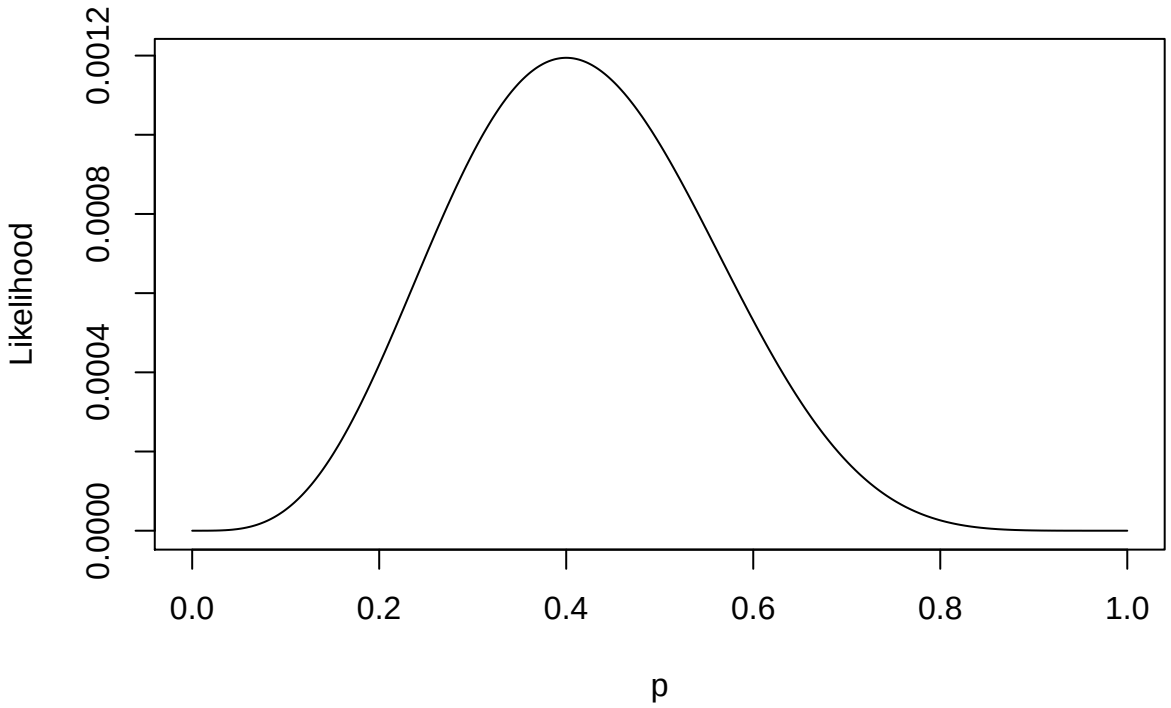


b.

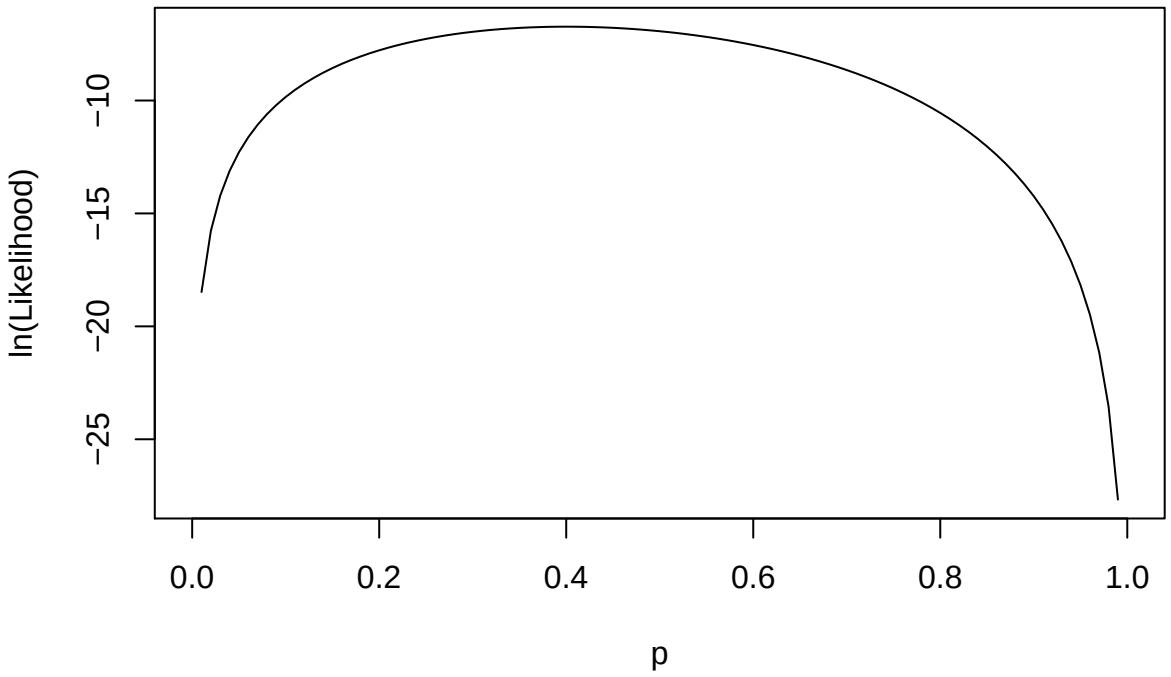
The spread of the histogram for the first simulation is much wider. More different values were obtained for $\hat{p}$ in the first simulation than the second. The second simulation with a larger sample size obtained the correct p more often and when the $\hat{p}$ was incorrect, it was closer to the correct p than in the first simulation.

c.

```
# likelihood function
l = function(p){
  # sample
  s <- c(1, 0, 1, 0, 0, 0, 0, 1, 0, 1)
  likelihood <- 1
  for(x in s){
    likelihood <- likelihood*p^x*(1-p)^(1-x)
  }
  return(likelihood)
}

curve(l, from = 0, to = 1, n = 1000, xlab = 'p', ylab = 'Likelihood')
```




```
ll = function(p){
  return(log(l(p)))
}
curve(ll, from = 0, to = 1, xlab = 'p', ylab = 'ln(Likelihood)')
```




```
# find maximum value
optimize(ll, interval = c(0,1), maximum = TRUE)
```

```
## $maximum
## [1] 0.4000015
##
## $objective
## [1] -6.730117
```

```
# log likelihood of p = 0.3, actual parameter value
ll(0.3)
```

```
## [1] -6.955941
```

d.

The graphs both have maxima around $p = 0.4$, but the shapes of the curves are pretty different. While the likelihood slopes like a more typical parabola, with the edges asymptoting towards -∞ and ∞ in the x-axis, the log transformed version is more of a rounded dome shape with the asymptotes approaching -∞ in the y-axis. Also the scale of the y-axis is very different. The likelihood is all positive, whereas the log likelihood is negative.

e.

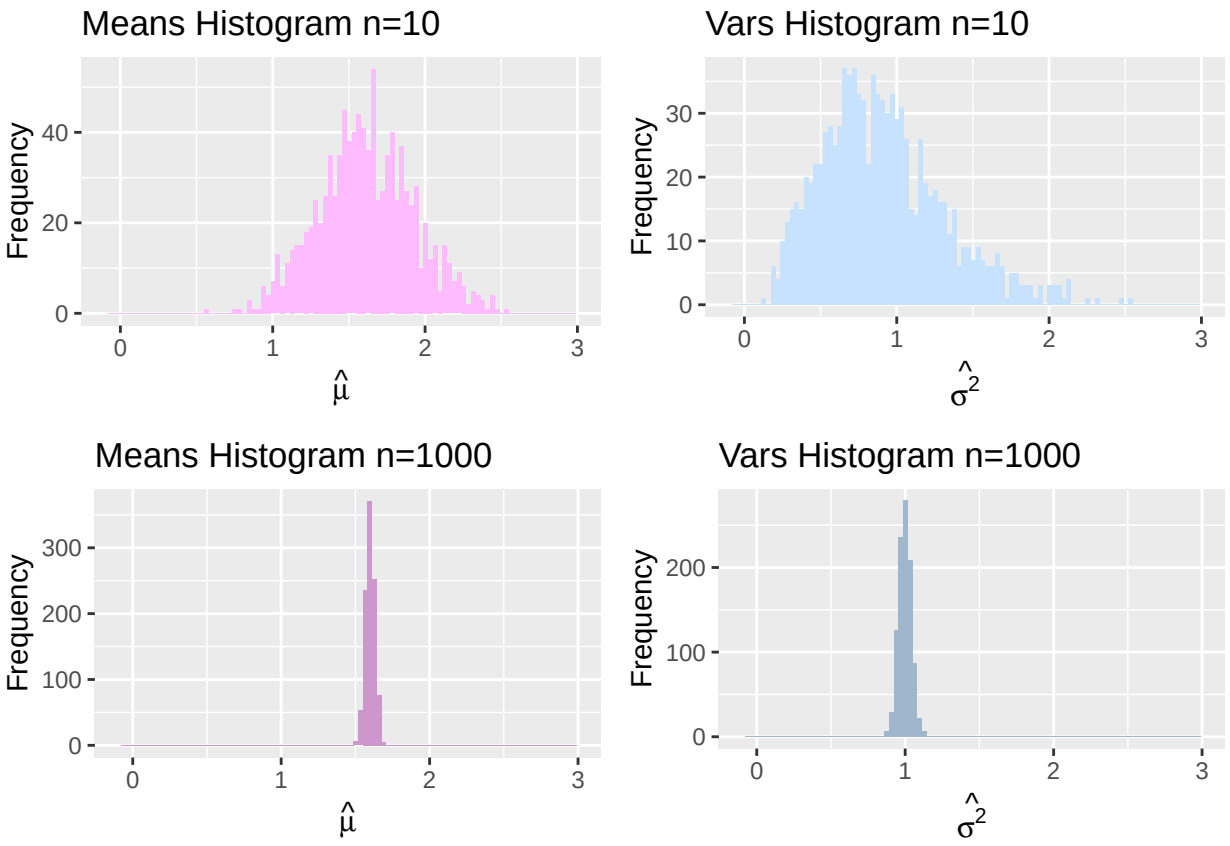
From a visual estimation from the graph, it seems $p = 0.4$ has the highest log likelihood (using optimize function gives $p = 0.4000015$, with log likelihood -6.730117). The log likelihood of $p = 0.3$ is -6.955941. No, the parameter value with the highest log likelihood given the sample is not the true parameter value. Again, $p = 0.4000015$, is the highest parameter value with log likelihood -6.730117. We expect that the parameter with the highest likelihood will not be the true parameter value–though it will be close to the true value– because we have only taken a sample of the entire space that is governed by the assumed correct

probability model. The natural randomness of this choice dictates that we will not get a sample that is perfectly representative of the underlying 'true' probability model.

f.

```
# function to use the mle form, not the unbiased var
mle_var <- function(x){
  return( sum((x - mean(x))^2)/ length(x))
}
# function to create simulation
sim_norm <- function(M, n, sigma_sq, mu){
  # store means and vars
  means <- vector()
  vs <- vector()
  for(i in 1:M){
    # create random normal sample
    s <- rnorm(n = n, mean = mu, sd = sqrt(sigma_sq))
    means <- c(means, mean(s))
    vs <- c(vs, mle_var(s))
  }
  # make dataframe for return
  return_df = as.data.frame(means, col.names = 'mean')
  return_df['var'] = vs
  return(return_df)
}
# M=1000,n=10,\mu=1.6, \sigma^2 = 1
norm_sim1 <- sim_norm(1000, 10, 1, 1.6)
# M=1000,n=1000,\mu=1.6, \sigma^2 = 1
norm_sim2 <- sim_norm(1000, 1000, 1, 1.6)

# plot histograms
hist_means1 <- ggplot(data.frame(norm_sim1$means), aes(norm_sim1$means)) +
  geom_histogram(bins = 100, fill='plum1') + labs(title = 'Means Histogram n=10', x=expression(hat(mu))
hist_vars1 <- ggplot(data.frame(norm_sim1$var), aes(norm_sim1$var)) +
  geom_histogram(bins = 100, fill='slategray1') + labs(title = 'Vars Histogram n=10', x=expression(hat(s
hist_means2 <- ggplot(data.frame(norm_sim2$means), aes(norm_sim2$means)) +
  geom_histogram(bins = 100, fill='plum3') + labs(title = 'Means Histogram n=1000', x=expression(hat(mu
hist_vars2 <- ggplot(data.frame(norm_sim2$var), aes(norm_sim2$var)) +
  geom_histogram(bins = 100, fill='slategray3') + labs(title = 'Vars Histogram n=1000', x=expression(ha
grid.arrange(hist_means1, hist_vars1, hist_means2, hist_vars2, nrow=2, ncol=2)
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).

## Warning: Removed 2 rows containing missing values (geom_bar).

## Warning: Removed 2 rows containing missing values (geom_bar).

## Warning: Removed 2 rows containing missing values (geom_bar).
```



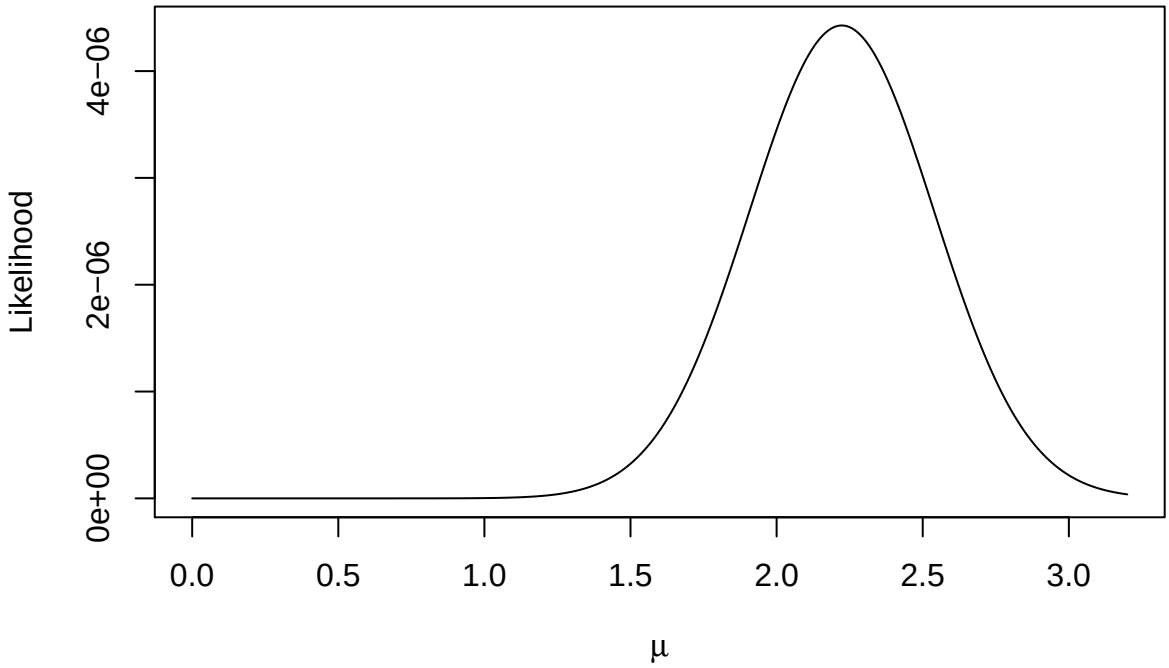
g.

The hitograms with $n = 1000$ were much more tightly centered around the correct values for the estimators. The sample with $n = 1000$ gave the correct answer for μ and even when the estimator was off, it was closer to the correct value of μ much more often than the sample with $n = 10$. The same pattern is seen for the σ^2

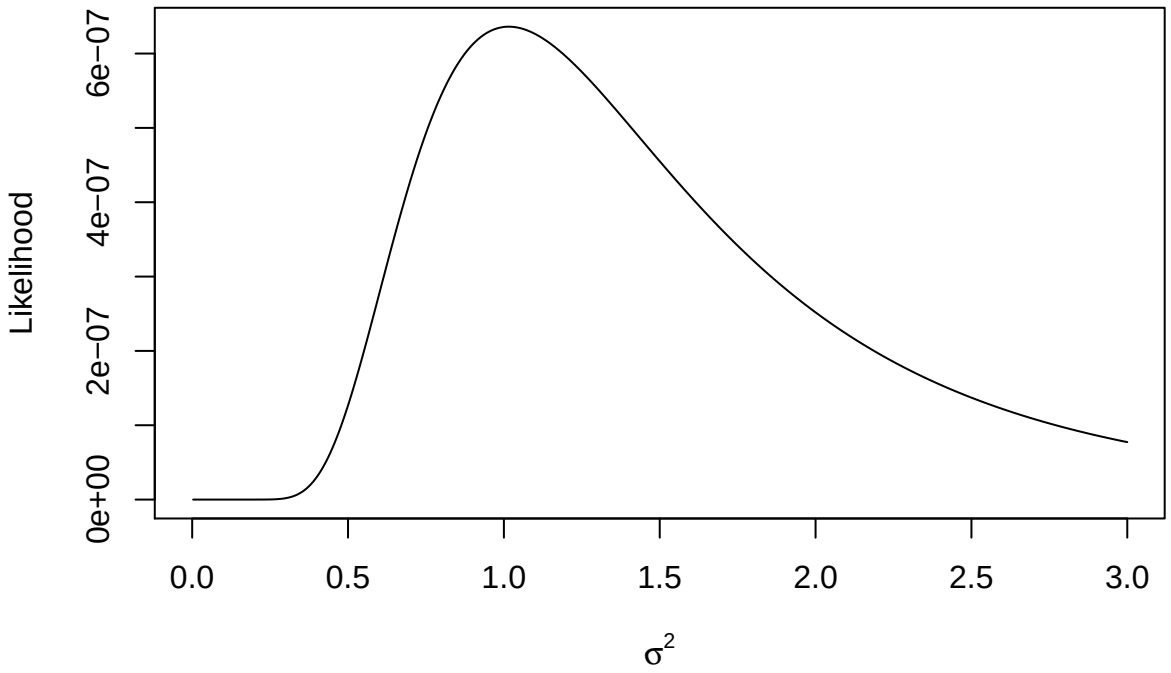
h.

```
# likelihood function mu
l_mu = function(mu){
  # sample
  sigma_sq <- 1
  s <- c(2.22, 0.98, 2.63, 3.33, 1.86, 3.25, 2.25, 2.92, 1.78, 1.01)
  likelihood <- 1
  for(x in s){
    likelihood <- likelihood*1/(sqrt(2*pi*sigma_sq))*exp(-(x-mu)^2/(2*sigma_sq))
  }
  return(likelihood)
}

# likelihood function sigma_sq
l_ss = function(sigma_sq){
  # sample
```

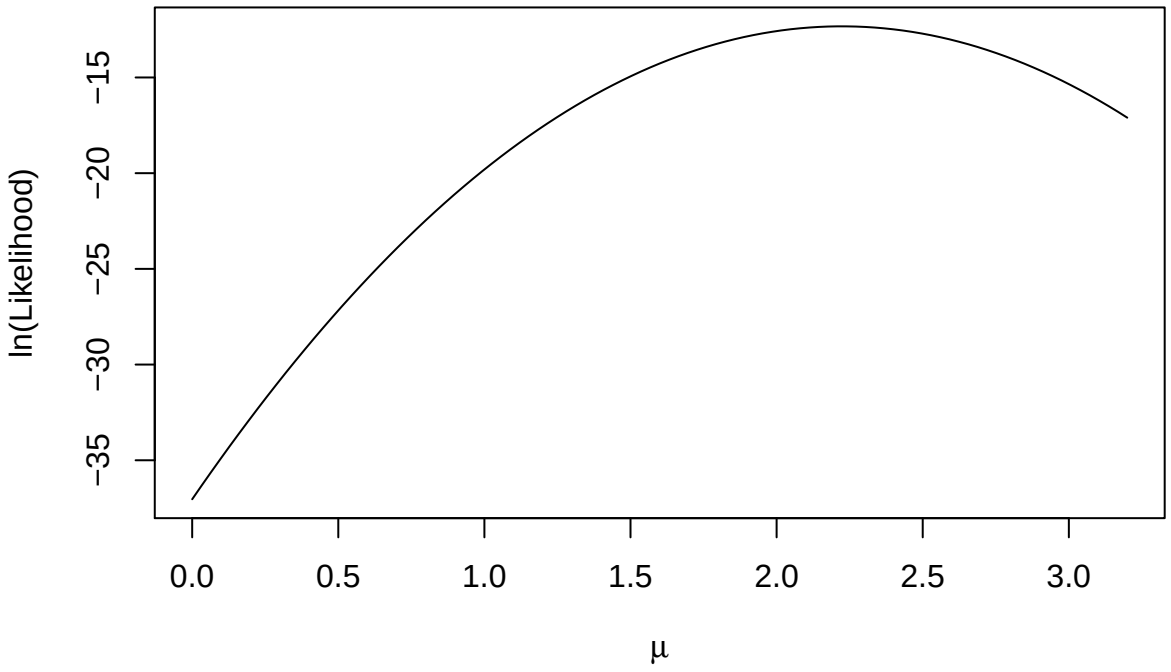
```
  mu <- 1.6
  s <- c(2.22, 0.98, 2.63, 3.33, 1.86, 3.25, 2.25, 2.92, 1.78, 1.01)
  likelihood <- 1
  for(x in s){
    likelihood <- likelihood*1/(sqrt(2*pi*sigma_sq))*exp(-(x-mu)^2/(2*sigma_sq))

  }
  return(likelihood)
}
# graph likelihoods
curve(l_mu, from = 0, to = 3.2, n = 1000, xlab = expression(mu), ylab = 'Likelihood')
```



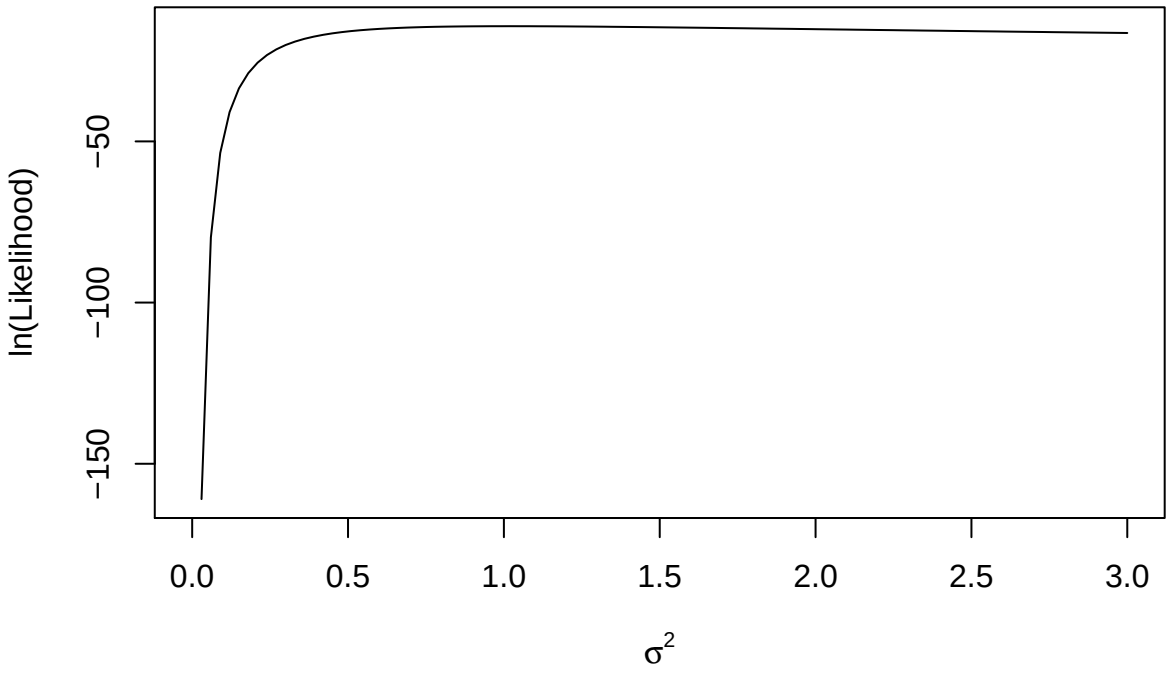

```
curve(l_ss, from = 0, to = 3, n = 1000, xlab = expression(sigma^2), ylab = 'Likelihood')
```




```r
# calculate log likelihoods
ll_mu = function(p){
  return(log(l_mu(p)))
}
ll_ss = function(p){
  return(log(l_ss(p)))
}
# graph log likelihoods
curve(ll_mu, from = 0, to = 3.2, xlab = expression(mu), ylab = 'ln(Likelihood)')
```



```r
curve(ll_ss, from = 0, to = 3, xlab = expression(sigma^2), ylab = 'ln(Likelihood)')
```



i.

```r
# find maximum value
optimize(ll_mu, interval = c(0,3.2), maximum = TRUE)
```

```
## $maximum
## [1] 2.223
##
## $objective
## [1] -12.32779
```

```r
optimize(ll_ss, interval = c(0,3), maximum = TRUE)
```

```
## $maximum
## [1] 1.015799
##
## $objective
## [1] -14.26782
```

$\mu = 2.223$ has the highest log likelihood of -12.32779 and $\sigma^2 = 1.015799$ has the highest log likelihood of -14.26782. The σ^2 is pretty close to the correct parameter value of 1, whereas the μ is further off. This makes sense because quickly looking at the data, they are spread about evenly, but the values are generally higher to much higher than the correct μ. We can expect the calculated parameters to be off from the correct values because we have only tested a sample of the entire space. The underlying probability model of this situation therefore dictates the chances we obtain certain values, but until $n = \infty$ we cannot be sure that we will obtain the correct value. The likelihood is a function of the data that was observed so the obtained distribution of the data, not the underlying assumed probability distribution governs the parameter estimates.

j.

It's impossible to know which calculation returns closer to the correct value for a given sample in a real instance because we cannot know what the true value is. Although, there is some comfort in that

$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - mean(\mathbf{x}))^2$

has been proven to return an unbiased estimator, such that the difference between the expected value of the estimator is no different from the underlying parameter. With increasing n, we assume the estimators will approach the same value, however with smaller n, the unbiased estimator is preferred because it is less likely to systematically over or under estimate the true value. On the other hand,

$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - mean(\mathbf{x}))^2$

is a consistent estimator. Which is a nice property for larger n as we can expect this estimator to converge to a single value.

Problem 3 (Difficult)

Considering the Bernoulli case

$$
\begin{aligned}
L(p|x) &= \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} \\
l(p|x) &= ln(p\sum_{i}^{n} x_i) + ln(1-p)\sum_{i}^{n} 1 - x_i \\
\frac{\partial l}{\partial p} &= \frac{\sum_i^n x_i}{p} - \frac{\sum_i^n 1 - x_i}{1-p} = 0 \\
\frac{\sum_i^n x_i}{p} &= \frac{\sum_i^n 1 - x_i}{1-p} \\
\frac{1-p}{p} &= \frac{\sum_i^n 1 - \sum_i^n x_i}{\sum_i^n x_i} \\
\frac{1}{p} - 1 &= \frac{n}{\sum_i^n x_i} - \frac{\sum_i^n x_i}{\sum_i^n x_i} \\
\hat{p} &= \frac{\sum_i^n x_i}{n}
\end{aligned}
$$

Gives the same $MLE(\hat{p})$ as solving the binomial case

$$
\begin{aligned}
L(p|n,x) &= \binom{n}{x} p^y (1-p)^{n-y} \\
l(p|n,x) &= ln(\binom{n}{x}) + xln(p) + (n-x)ln(1-p) \\
\frac{\partial l}{\partial p} &= \frac{x}{p} - \frac{n-x}{1-p} = 0 \\
\frac{x}{p} &= \frac{n-x}{1-p} \\
\frac{1-p}{p} &= \frac{n-x}{x} \\
\frac{1}{p} - 1 &= \frac{n}{x} - 1 \\
p &= \frac{x}{n}
\end{aligned}
$$