QBS 120 - Problem Set 5

Rob Frost

Grading: problems 1 and 3, 6pts for problem 1 and 4 pts for problem 3

1. **Point and interval estimation for the first gene in the Golub data set. Assume the observed values of the first gene in the Golub data can be modeled as i.i.d $\mathcal{N}(\mu, \sigma^2)$.**

   Grading: 0.5 pts for parts a) - j), to get credit for these parts they just need to specify the correct answer and provide some type of justification if asked (exact wording is not important); 1 pt for part k), they get full credit if they get the right CI values and half credit if the approach looks valid but they made some error in the computations.

   (a) **What is the MLE of μ ($\hat{\mu}_{mle}$)?**
   For an iid sample $X_1, ..., X_n$, the MLE for $E[X] = \mu$ is the sample average:

   $$\hat{\mu}_{mle} = \bar{X}$$

   This is also the MOM estimator. For gene 1 in Golub, this is

   ```
   > library(multtest)
   > data(golub)
   > gene1.values = golub[1,]
   > (mu.mle = mean(gene1.values))
   [1] -1.129013
   ```

   (b) **Is your estimate from part a) unbiased? Justify.**

   Yes, this estimate is unbiased. Specifically, $E[\bar{X}] = \mu$.

   (c) **Is your estimate from part a) consistent? Justify.**

   Yes, this estimate is consistent. Per the LLN, we know that the sample average converges in probability to μ, which is the definition of a consistent estimator.

   (d) **If you cannot assume the data are normally distributed, is your estimate from a) for $E[X] = \mu$ still valid?**
   Yes, $\bar{X}$ is an unbiased and consistent estimator for the expected value of iid RVs regardless of the distribution.

   (e) **What is the MLE of σ^2 ($\hat{\sigma}^2_{mle}$)?**

We know that the MLE for σ^2 for iid normal RVs is:

$$\hat{\sigma}^2_{mle} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

This is also the MOM estimator. This is NOT the estimate returned by the R var() function (that is the unbiased estimator). So, need to use a different function to calculate for gene 1 in Golub. I'll implement one that can compute either the MLE or the unbiased estimator:

```
> varMLE = function(x, mle=T) {
+         x.bar = mean(x)
+         squared.diffs = (x-x.bar)^2
+         if (mle) {
+                 var.est = mean(squared.diffs)
+         } else {
+                 var.est = sum(squared.diffs)/(length(x)-1)
+         }
+         return (var.est)
+ }
> (var.mle = varMLE(gene1.values, mle=T))
[1] 0.3364397
> (var.unbiased = varMLE(gene1.values, mle=F))
[1] 0.3455326
```

Let's compare that with the standard R var():

```
> var(gene1.values)
[1] 0.3455326
```

(f) **Is your estimate from part e) unbiased? Justify.**

No, the MLE for σ^2 is not unbiased. The unbiased estimator uses $1/(n-1)$ rather than $1/n$. However, the MLE is asymptotically unbiased, i.e., as $n \to \infty, 1/(n-1) \to 1/n$. All MLEs are asymptotically unbiased.

(g) **Is your estimate from part e) consistent? Justify.**

Yes, it is consistent. This follows from being an MLE and MOM estimator.

(h) **If you cannot assume the data are normally distributed, is your estimate from e) for $Var(X) = \sigma^2$ still valid?**

Yes, it is still a valid (though biased) estimator of the variance of iid RVs. In the general case, it is the MOM estimator but not necessarily the MLE.

(i) **What is the distribution of $\hat{\mu}_{mle}$? If you cannot assume the data are normally distributed, does this sampling distribution still hold?**

For normal iid RVs, $\hat{\mu}_{mle} = \bar{X} \sim \mathcal{N}(\mu, \sigma^2/n)$. If the data is not normally distributed, $\bar{X}$ converges in distribution to $\mathcal{N}(\mu, \sigma^2/n)$.

(j) **What is the MSE of the $\hat{\mu}_{mle}$ estimate?**

Grading: get full credit as long as they recognize that the MSE in this case is just the variance of the estimate; they don't need to compute the approximate value.

The mean squared error (MSE) of an estimator is defined as $MSE(\hat{\theta}) = Var(\hat{\theta}) + (\hat{\theta} - \theta)^2$ or the variance of the estimator plus the squared bias. In this case, the estimator is unbiased so the MSE of $\hat{\mu}_{mle}$ is just the variance or σ^2/n. For gene 1 in Golub, we don't know the true variance but can estimate the MSE using our estimate of σ^2:

```
> (n = length(gene1.values))
[1] 38
> (mse.mu.mle = var.mle/n)
[1] 0.008853675
```

(k) **Use the distribution of $\hat{\mu}_{mle}$ to compute a 95% CI for μ. Assume that the $\sigma^2 = \hat{\sigma}^2_{mle}$.**

To find the 95% CI for μ we need to first define values -z and z that define a region of a probability distribution with area 0.95. The MLE for μ , $\bar{X}$, has a normal distribution in this case and, after standardization, has a $\mathcal{N}(0, 1)$ distribution so we can use this as the density for determining z. Specifically, we want to find z such that:

$$P(-z \leq (\bar{X} - \mu)/(\sigma/\sqrt{n}) \leq z) = 0.95$$

Since we don't know the true value of σ, we need to plug in the MLE:

$$P(-z \leq (\bar{X} - \mu)/(\hat{\sigma}_{mle}/\sqrt{n}) \leq z) = 0.95$$

Normally the use of an unbiased estimate of σ (rather than the biased MLE) would imply that $(\bar{X} - \mu)/(\hat{\sigma}_{mle}/\sqrt{n})$ has a t distribution rather than a normal distribution, however, we are told that $\sigma^2 = \hat{\sigma}^2_{mle}$ so can still model using a standard normal. Specifically, $(\bar{X} - \mu)/(\hat{\sigma}_{mle}/\sqrt{n}) \sim \mathcal{N}(0, 1)$.

To find z, we can therefore use the CDF of the standard normal. Specifically:

$$\begin{aligned} \Phi(z) - \Phi(-z) &= 0.95 \\ \Phi(z) - (1 - \Phi(z)) &= 0.95 \\ 2\Phi(z) &= 1.95 \\ \Phi(z) &= 0.975 \\ z &= \Phi^{-1}(0.975) \end{aligned}$$

We can use qnorm() to get the value:

```
> (z = qnorm(0.975))
[1] 1.959964
```

Very close to the 1.96 rule-of-thumb value. So, we can now plug the value for z into the probability statement:

$$P(-\Phi^{-1}(0.975) \leq (\bar{X} - \mu)/(\hat{\sigma}_{mle}/\sqrt{n}) \leq \Phi^{-1}(0.975)) = 0.95$$

This defines a 95% CI for $(\bar{X} - \mu)/(\hat{\sigma}_{mle}/\sqrt{n})$ but we want a CI for μ. To find that, need to use simple algebra to isolate μ in the middle of the inequality:

$$P(-\Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n}) \leq \bar{X} - \mu \leq \Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n})) = 0.95$$
$$P(-\Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n}) - \bar{X} \leq -\mu \leq \Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n}) - \bar{X}) = 0.95$$
$$P(\Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n}) + \bar{X} \geq \mu \geq -\Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n}) + \bar{X}) = 0.95$$
$$P(\bar{X} - \Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n}) \leq \mu \leq \bar{X} + \Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n})) = 0.95$$

In other words, the CI is centered on $\bar{X}$, our MLE estimate for μ, and extends above and below by $\Phi^{-1}(0.975)(\hat{\sigma}_{mle}/\sqrt{n})$, which makes sense. We can compute the specific values for gene 1 in Golub using R:

```
> (lower.ci = mu.mle - qnorm(0.975)*sqrt(var.mle/n))
[1] -1.313434
> (upper.ci = mu.mle + qnorm(0.975)*sqrt(var.mle/n))
[1] -0.9445924
```

To check the validity of our parametric assumptions, can compare this to the bootstrap percentile CI (not required for the problem set but easy to do):

```
> # create 10,000 bootstrap resampled data sets and compute the mean for each
> x.bar.bs = rep(0, 10000)
> for (i in 1:10000) {
+         x.bar.bs[i] = mean(sample(gene1.values, n, replace=T))
+ }
> (lower.ci.bs = sort(x.bar.bs, decreasing=F)[250])
[1] -1.290872
> (upper.ci.bs = sort(x.bar.bs, decreasing=F)[9750])
[1] -0.9235847
```

The bootstrap percentile CI is close and a bit narrower (which is expected).

2. **(Based on Rice, Chapter 8, Problem 3) One of the earliest applications of the Poisson distribution was made by Student (1907) in studying errors made in counting yeast cells or blood corpuscles with a haemacytometer. In this study, yeast cells were killed and mixed with water and gelatin; the mixture was then spread on a glass and allowed to cool. Four different concentrations were used. Counts were made on 400 squares, and the data are summarized in the data.frame below. In this data.frame, each of the "concen.*" columns records the number of squares associated with that concentration for which the number of counted cells equals the value in the "cells" column.**

   (a) **Compute the MLE estimate of the parameter λ for each of the four sets of data.**

   First, need to create a R data.frame to old all of the yeast counts from the table in Rice:

```
> yeast.counts = data.frame(cells=0:12,
+          concen.1 = c(213,128,37,18,3,1,0,0,0,0,0,0,0),
+          concen.2 = c(103,143,98,42,8,4,2,0,0,0,0,0,0),
+          concen.3 = c(75,103,121,54,30,13,2,1,0,1,0,0,0),
+          concen.4 = c(0,20,43,53,86,70,54,37,18,10,5,2,2))
```

The MLE estimate of λ for a Poisson distribution is $\bar{X}$. For this data set, these MLE estimates are the following:

```
>         computeMLE = function(cell.counts, square.counts) {
+                 total.squares = sum(square.counts)
+                 total.yeast = sum(cell.counts*square.counts)
+                 mean.yeast.square = total.yeast/total.squares
+                 return (mean.yeast.square)
+         }
>         concen.1.mle = computeMLE(yeast.counts$cells, yeast.counts$concen.1)
>         concen.2.mle = computeMLE(yeast.counts$cells, yeast.counts$concen.2)
>         concen.3.mle = computeMLE(yeast.counts$cells, yeast.counts$concen.3)
>         concen.4.mle = computeMLE(yeast.counts$cells, yeast.counts$concen.4)
```

- $\hat{\lambda}_{MLE,concen1} = 0.682$
- $\hat{\lambda}_{MLE,concen2} = 1.32$
- $\hat{\lambda}_{MLE,concen3} = 1.8$
- $\hat{\lambda}_{MLE,concen4} = 4.68$

(b) **Approximate the theoretical standard error of the $\hat{\lambda}$ values computed for part a). Do not use simulation.**

Although we know that the theoretical standard error of $\bar{X}$ is $\sigma/\sqrt{n} = \sqrt{\lambda/n}.$, this can not be calculated without knowing the true population value of λ. However, we can estimate it by plugging in the estimated $\hat{\lambda}_{mle}$ values:

$$SE_{\hat{\lambda}_{mle}} \approx \sqrt{\hat{\lambda}_{mle}/n}$$

```
>         computeSE = function(lambda.hat, n) {
+                 return (sqrt(lambda.hat/n))
+         }
>         concen.1.se = computeSE(concen.1.mle, sum(yeast.counts$concen.1))
>         concen.2.se = computeSE(concen.2.mle, sum(yeast.counts$concen.2))
>         concen.3.se = computeSE(concen.3.mle, sum(yeast.counts$concen.3))
>         concen.4.se = computeSE(concen.4.mle, sum(yeast.counts$concen.4))
```

- $SE_{\hat{\lambda}_{mle},concen1} = 0.0413$
- $SE_{\hat{\lambda}_{mle},concen2} = 0.0575$
- $SE_{\hat{\lambda}_{mle},concen3} = 0.0671$
- $SE_{\hat{\lambda}_{mle},concen4} = 0.108$

(c) **For the $\hat{\lambda}$ values compute for part a), estimate the standard error using the parametric bootstrap. How do these values compare to the approximate theoretical values? Do these results match you expectations?**

For the parametric boostrap, we generate many samples using the estimated distribution parameters:

```
>         bootstrapSE = function(lambda.hat, n, b=10000) {
+                 sim.vals = matrix(rpois(n*b, lambda=lambda.hat) , nrow=b, ncol=n)
+                 sim.lambda.hats = apply(sim.vals, 1, mean)
+                 sim.lambda.SE =sd(sim.lambda.hats)
+                 return (sim.lambda.SE)
+         }
>         concen.1.boot.se = bootstrapSE(concen.1.mle, n=sum(yeast.counts$concen.1))
>         concen.2.boot.se = bootstrapSE(concen.2.mle, n=sum(yeast.counts$concen.2))
>         concen.3.boot.se = bootstrapSE(concen.3.mle, n=sum(yeast.counts$concen.3))
>         concen.4.boot.se = bootstrapSE(concen.4.mle, n=sum(yeast.counts$concen.4))
```

- $SE^b_{\hat{\lambda}_{mle},concen1} = 0.0414$
- $SE^b_{\hat{\lambda}_{mle},concen2} = 0.0575$
- $SE^b_{\hat{\lambda}_{mle},concen3} = 0.0667$
- $SE^b_{\hat{\lambda}_{mle},concen4} = 0.108$

The bootstrap SEs are very close to the approximated theoretical SEs, which matches expectations.

(d) **Find an approximate** 95% **confidence interval for each estimate.**
The CI could be computed using either the asymptotic normal or parametric bootstrap approximation. I will use the asymptotic normal distribution:

$$\hat{\lambda} \rightarrow_d \mathcal{N}(\lambda, 1/nI(\lambda))$$
$$\hat{\lambda} \rightarrow_d \mathcal{N}(\lambda, \lambda/n)$$

Since λ is unknown, can substitute the MLE estimate:

$$\hat{\lambda} \rightarrow_d \mathcal{N}(\bar{X}, \bar{X}/n)$$

The 95% CI is: $\bar{X} + Z(.025)\sqrt{\bar{X}/n}$ to $\bar{X} + Z(.975)\sqrt{\bar{X}/n}$

```
>         n = 400
>         concen.1.CI.low = concen.1.mle + qnorm(.025)*sqrt(concen.1.mle/n)
>         concen.1.CI.high = concen.1.mle + qnorm(.975)*sqrt(concen.1.mle/n)
>         concen.2.CI.low = concen.2.mle + qnorm(.025)*sqrt(concen.2.mle/n)
>         concen.2.CI.high = concen.2.mle + qnorm(.975)*sqrt(concen.2.mle/n)
>         concen.3.CI.low = concen.3.mle + qnorm(.025)*sqrt(concen.3.mle/n)
>         concen.3.CI.high = concen.3.mle + qnorm(.975)*sqrt(concen.3.mle/n)
>         concen.4.CI.low = concen.4.mle + qnorm(.025)*sqrt(concen.4.mle/n)
>         concen.4.CI.high = concen.4.mle + qnorm(.975)*sqrt(concen.4.mle/n)
>
```

- 95% CI for $\lambda_{MLE,concen1} = 0.602$-$0.763$

- 95% CI for $\lambda_{MLE,concen2}$ = 1.21-1.44
- 95% CI for $\lambda_{MLE,concen3}$ = 1.67-1.93
- 95% CI for $\lambda_{MLE,concen4}$ = 4.47-4.89

(e) **Compare observed and expected counts.**

The match is good for all of the data sets.

- Concentration 1:

```
>         obs.vs.exp = data.frame(observed=yeast.counts$concen.1,
+               expected=round(400*dpois(yeast.counts$cells, concen.1.mle)))
>         print(t(obs.vs.exp))
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
observed  213  128   37   18    3    1    0    0    0     0     0     0     0
expected  202  138   47   11    2    0    0    0    0     0     0     0     0
```

- Concentration 2:

```
>         obs.vs.exp = data.frame(observed=yeast.counts$concen.2,
+               expected=round(400*dpois(yeast.counts$cells, concen.2.mle)))
>         print(t(obs.vs.exp))
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
observed  103  143   98   42    8    4    2    0    0     0     0     0     0
expected  107  141   93   41   14    4    1    0    0     0     0     0     0
```

- Concentration 3:

```
>         obs.vs.exp = data.frame(observed=yeast.counts$concen.3,
+              expected=round(400*dpois(yeast.counts$cells, concen.3.mle)))
>         print(t(obs.vs.exp))
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
observed   75  103  121   54   30   13    2    1    0     1     0     0     0
expected   66  119  107   64   29   10    3    1    0     0     0     0     0
```

- Concentration 4:

```
>         obs.vs.exp = data.frame(observed=yeast.counts$concen.4,
+             expected=round(400*dpois(yeast.counts$cells, concen.4.mle)))
>         print(t(obs.vs.exp))
          [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
observed    0   20   43   53   86   70   54   37   18    10     5     2     2
expected    4   17   41   63   74   69   54   36   21    11     5     2     1
```

3. **(Based on Rice, Chapter 8, Problem 9) How would you respond to the following argument? This talk of sampling distributions is ridiculus! Consider Example A of Section 8.4. The experimenter found the mean of the number of fibers to be 24.9. How can this be a "random variable" with an associated "probability distribution" when it's just a number? The author of this book is guilty of deliberate mystification!**

Grading: 4pts; just need to provide a reasonable answer, exact wording is not important; everyone should get full credit for this one!

A random variable is a function mapping events in the sample space, Ω, to the real numbers, $\mathbb{R}$. The specific value of the method of moments estimate $\hat{\lambda}$ provided in Example A of section

8.4 represents just a single realization of this function (i.e., the mapping of one event in Ω to $\mathbb{R}$). A set of randomly distributed values (distributed according to the MOM distribution) would be generated if multiple experiments were performed and the MOM estimate were computed for each experiment.

Another way of viewing $\hat{\lambda}$ is as a function of n iid Poisson RVs and a function of RVs is a RV. In this case, the function is being computed for just a single realization of each of the underlying Poisson RVs.

4. **(Based on Rice, Chapter 8, Problem 13) In Example D of Section 8.4, the MOM estimate was found to be $\hat{\alpha} = 3\bar{X}$. In this problem, you will consider the sampling distribution of $\hat{\alpha}$.**

   (a) **Show that $E[\hat{\alpha}] = \alpha$, i.e., the estimate is unbiased.**
   Note that Example D found that $\mu = \alpha/3$ so we can equivalently show that $E[\hat{\alpha}] = 3\mu$.

$$\begin{aligned} E[\hat{\alpha}] &= E[3\bar{X}] && \text{given} \\ &= 3E[\bar{X}] && E[bX] = bE[X] \\ &= 3\mu && E[\bar{X}] = \mu \\ &= \alpha \end{aligned}$$

   (b) **Show that $Var(\hat{\alpha}) = (3 - \alpha^2)/n$. Hint: What is $Var(\bar{X})$?**

$$\begin{aligned} Var(\hat{\alpha}) &= Var(3\bar{X}) && \text{given} \\ &= 9Var(\bar{X}) && Var(bX) = b^2Var(X) \\ &= 9/n^2 Var(\sum_{i=1}^{n} X_i) && Var(bX) = b^2Var(X) \\ &= 9/n^2 \sum_{i=1}^{n} Var(X_i)) && \text{Var of sum of indep RVs} \\ &= 9/nVar(X) && X_i \text{ are iid} \end{aligned}$$

   Now need to find Var(X):

$$
\begin{aligned}
Var(X) &= E[X^2] - E[X]^2 && \text{by defn} \\
&= E[X^2] - \mu^2 && E[X] = \mu \\
&= \int_{-1}^{1} x^2 \frac{1+\alpha x}{2} dx - \mu^2 && \text{def of E} \\
&= \int_{-1}^{1} \frac{x^2 + \alpha x^3}{2} dx - \mu^2 \\
&= \left[x^3/6 + \alpha x^4/8 \right]_{-1}^{1} - \mu^2 \\
&= 1/6 + \alpha/8 + 1/6 - \alpha/8 - \mu^2 \\
&= 1/3 - \mu^2 \\
&= 1/3 - \alpha^2/9
\end{aligned}
$$

Combine everything:

$$
\begin{aligned}
Var(\hat{\alpha}) &= 9/n Var(X) \\
&= 9/n(1/3 - \alpha^2/9) \\
&= \frac{3 - \alpha^2}{n}
\end{aligned}
$$

(c) **Use the CLT to deduce that a normal approximation to the sampling distribution of $\hat{\alpha}$. According to this approximation, if $n = 20$ and $\alpha = 1$, what is the $P(\hat{\alpha} > 0.5)$? Define in terms of $\Phi()$, the CDF for the standard normal.**
Because $\hat{\alpha} = 3\bar{X}$, it is a function of the average of independent so, by the CLT, converges in distribution to a normal. Using the expectation and variance we found above, we can state the approximate distribution of $\hat{\alpha}$:

$$\hat{\alpha} \sim \mathcal{N}(\alpha, \frac{3 - \alpha^2}{n})$$

If n=20 and $\alpha = 1$, this becomes:

$$\hat{\alpha} \sim \mathcal{N}(1, 0.1)$$

Standardizing we get:

$$\sqrt{10}(\hat{\alpha} - 1) \sim \mathcal{N}(0, 1)$$

We are asked to find $P(\alpha > 0.5)$:

$$
\begin{aligned}
P(\alpha > 0.5) &= 1 - P(\alpha < 0.5) \\
&= 1 - P(\sqrt{10}(\alpha - 1) < \sqrt{10}(0.5 - 1)) \\
&= 1 - P(\sqrt{10}(\alpha - 1) < -0.5\sqrt{10}) \\
&= 1 - \Phi(-0.5\sqrt{10})
\end{aligned}
$$

We can solve this using pnorm():

```
> 1-pnorm(-0.5*sqrt(10))
[1] 0.9430769
```

5. **(Based on Rice, Chapter 8, Problem 58) For a population in Hardy-Weinberg equilibrium, alleles occur with the following frequencies:**

   - AA: $(1-\theta)^2$
   - Aa: $2\theta(1-\theta)$
   - aa: θ^2

   For a specific sample of 190 people, the haptoglobin types occurred as follows:

$$X_1 : Hp1-1 : 10$$
$$X_2 : Hp1-2 : 68$$
$$X_3 : Hp2-2 : 112$$

   Assuming the haptoglobin genotype for this population is in Hardy-Weinberg equilibrium.

   (a) **Find the mle of θ.**
   Log-likelihood of θ:

$$L(\theta) = \frac{n!}{\prod_{i=1}^{3} x_i!} \prod_{i=1}^{3} p_i^{x_i}$$
$$log(L(\theta)) = log(\frac{n!}{\prod_{i=1}^{3} x_i!} \prod_{i=1}^{3} p_i^{x_i})$$
$$l(\theta) = log(n!) - \sum_{i=1}^{3} log(x_i!) + \sum_{i=1}^{3} x_i log(p_i)$$
$$l(\theta) = log(190!) - (log(10!) + log(68!) + log(112!)) + 20log(1-\theta) + 68log(2\theta(1-\theta)) + 224log(\theta)$$
$$l(\theta) = log(190!) - (log(10!) + log(68!) + log(112!)) + 20log(1-\theta) + 68log(2) + 68log(\theta) + 68log(1-\theta) + 224log(\theta)$$
$$l(\theta) = constant + 88log(1-\theta) + 292log(\theta)$$

   MLE of θ:

$$\frac{\delta}{\delta\sigma} l(\sigma) = \frac{\delta}{\delta\sigma}(constant + 88log(1-\theta) + 292log(\theta))$$
$$\frac{\delta}{\delta\sigma} l(\sigma) = -88/(1-\theta) + 292/\theta$$
$$0 = -88/(1-\theta) + 292/\theta$$
$$0 = -88\theta + 292(1-\theta)$$
$$380\theta = 292$$
$$\hat{\theta} = .768$$

(b) **Find the asymptotic variance of the mle.**
For random multinomial counts, $Var(\hat{\theta}) \approx -\frac{1}{E[l''(\theta)]}$.

$$E[l''(\theta)] = E[\frac{\delta^2}{\delta^2\theta}(log(n!) - \sum_{i=1}^{3} log(x_i!) + \sum_{i=1}^{3} x_i log(p_i))]$$

$$E[l''(\theta)] = E[\frac{\delta^2}{\delta^2\theta}((2X_1 + X_2)log(1-\theta) + (2X_3 + X_2)log(\theta))]$, *drop non-theta terms and expand*$$

$$E[l''(\theta)] = E[\frac{\delta}{\delta\theta}(-(2X_1 + X_2)/(1-\theta) + (2X_3 + X_2)/\theta)]$$

$$E[l''(\theta)] = E[-(2X_1 + X_2)/(1-\theta)^2 - (2X_3 + X_2)/\theta^2]$$

$$E[l''(\theta)] = \frac{-2n}{\theta(1-\theta)}$, *binomial expectations and simplification*$$

$$E[l''(\theta)] = \frac{-2n}{\hat{\theta}(1-\hat{\theta})}$, *sub in mle*$$

```
>          (var.theta.hat = (.768*(1-.768))/(2*190))
[1] 0.0004688842
```

(c) **Find an approximate 99% confidence interval for θ.**
Given the asymptotic normal distribution of the MLE, the 99% CI for θ is:

$$Z(.005) \leq \sqrt{Var(\hat{\theta})}(\theta - \hat{\theta}) \leq Z(.995)$$
$$\frac{Z(.005)}{\sqrt{Var(\hat{\theta})}} \leq (\theta - \hat{\theta}) \leq \frac{Z(.995)}{\sqrt{Var(\hat{\theta})}}$$
$$\frac{Z(.005)}{\sqrt{Var(\hat{\theta})}} + \hat{\theta} \leq \theta \leq \frac{Z(.995)}{\sqrt{Var(\hat{\theta})}} + \hat{\theta}$$

```
> (lower.CI = .768 + qnorm(.005)*sqrt(var.theta.hat))
[1] 0.7122237
> (upper.CI = .768 + qnorm(.995)*sqrt(var.theta.hat))
[1] 0.8237763
```

(d) **Use the parametric bootstrap to estimate the sampling distribution of the MLE of θ. Plot this distribution along with the asymptotic distribution. How does the shape of the bootstrap sampling distribution compare to the asymptotic distribution?**
This is very similar to the example from the lecture. For the parametric bootstrap, we will use the MLE estimate of θ to generate multiple data sets each containing 190 members and recompute the MLE for each simulated sample. Using $\hat{\theta}_{mle} = 0.768$, we get the following probabilities for the different multinomial categories:

- Hp1-1: $(1-\theta)^2 = 0.054$
- Hp1-2: $2\theta(1-\theta) = 0.356$
- Hp2-2: $\theta^2 = 0.589$

In order to compute the MLE for each of the generated samples, we need an expression for the MLE in terms of the counts of the three groups. From lecture, we know this is:

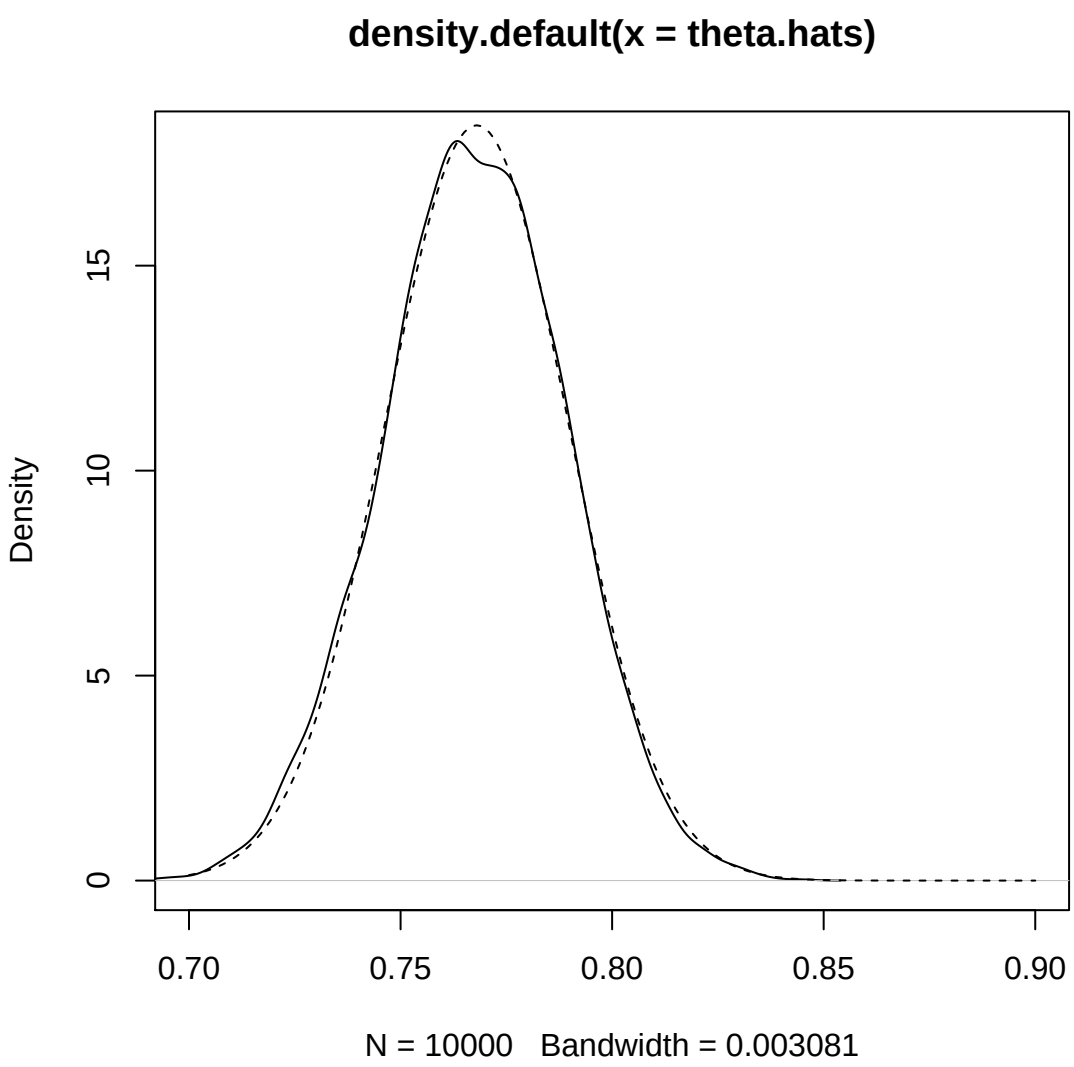
$$\hat{\theta}_{mle} = (2 * n3 + n2)/(2 * (n1 + n2 + n3)))$$

```
> sim.samples = rmultinom(10000, size=190,
+         prob=c(0.054, 0.356, 0.589))
> sim.samples[,1:10]
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]   10    8   11    5   10   10   11   12   15     7
[2,]   73   64   73   72   79   81   76   68   70    63
[3,]  107  118  106  113  101   99  103  110  105   120
```

```
> thetaMLE = function(n1, n2, n3) {
+         return ((2*n3 + n2)/(2*(n1+n2+n3)))
+ }
> theta.hats = apply(sim.samples, 2, function(x) {
+         return (thetaMLE(x[1], x[2], x[3]))
+ })
```

```
> plot(density(theta.hats), xlim=c(0.7, 0.9))
> x.vals = seq(from=0.7, to=0.9, by=0.001)
> asym.density = dnorm(x.vals, mean=0.768, sd=sqrt(var.theta.hat))
> points(x.vals, asym.density, type="l", lty="dashed")
```



As expected, the asymptotic and bootstrap sampling distributions are quite close!

(e) **Use the bootstrap sampling distribution to estimate the variance of the MLE of θ. How does the bootstrap variance compare with the asymptotic variance?**

```
> var.theta.hat
[1] 0.0004688842
> (var.theta.hat.bootstrap = var(theta.hats))
[1] 0.0004667424
```

As expected, they are quite close.

(f) **Compute the 99% CI for θ using the bootstrap percentile approach. How does this compare with the CI computed in part c)?**

This is very similar to the bootstrap CI example from the lecture. Specifically, we want to find the 0.005 and 0.995 quantiles:

```
> (theta.005 = sort(theta.hats, decreasing=F)[50])
```

```
[1] 0.7105263
> (theta.995 = sort(theta.hats, decreasing=F)[9950])
[1] 0.8236842
```

The asymptotic 99% CI was:

```
> (lower.CI = .768 + qnorm(.005)*sqrt(var.theta.hat))
[1] 0.7122237
> (upper.CI = .768 + qnorm(.995)*sqrt(var.theta.hat))
[1] 0.8237763
```

As expected, these are similar.