Applied Statistical Modeling & Inference: Homework #2

Due on February 27, 2015 at 5:00pm

Professor Ying Lu & Professor Daphna Harel

Sriniketh Vijayaraghavan

Problem 1

Show that the sample Variance $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \hat{X}^2)$ is an unbiased estimator of $V(X)$.

- Note: $V(X) = E(X^2) - (E(X))^2$, and $\frac{1}{n}\sum_{i=1}^{n} X_i^2$ is an unbiased estimator of $E(X^2)$.

Solution
If $E[u(X_1, ..., X_n)] = \theta$, then $u(X_1, ..., X_n)$ is an unbiased estimator of θ.

$$\begin{aligned}
&E(S^2) = \\
&\quad E(\frac{1}{n-1}\sum(X_i - \bar{X})^2) = E(\frac{1}{n-1}\sum(X_i^2 + \bar{X}^2 - X_i\bar{X})) \\
&\quad = \frac{1}{n-1}\sum(E(X_i^2) + E(\bar{X}^2) - 2\bar{X}E(X_i)) \\
&\quad = \frac{1}{n-1}\sum(E(X_i^2) + E(\bar{X}^2) - 2\bar{X}E(X_i)) \\
(n-1)E(S^2) &= E(\sum X_i^2) + E(\sum \bar{X}^2) - 2E(\sum \bar{X}^2) \\
(n-1)E(S^2) &= nE(X^2) - nE(\bar{X}^2) \\
\implies \frac{n-1}{n}E(S^2) &= E(X^2) - E(\bar{X}^2)
\end{aligned}$$

Let Y be a random variable with sample mean x

$$\begin{aligned}
&E(Y^2) \\
&\quad = E(\bar{X}^2) = V(Y) + (E(Y))^2 \\
&\quad = V[\frac{1}{n}\sum x_i] + \mu^2 = \frac{1}{n^2}V[\sum x_i] + \mu^2 \\
&\quad = \frac{1}{n^2}\sum V(x_i) + \mu^2 = \frac{1}{n^2}\sum \sigma^2 + \mu^2 = \frac{\sigma^2}{n} + \mu^2 \\
&\quad \implies \frac{n-1}{n}E(S^2) = (\sigma^2 + \mu^2) - \frac{\sigma^2}{n} - \mu^2 \\
&\quad \implies E(S^2) = \sigma^2
\end{aligned}$$

This shows that S^2 is an unbiased estimator of $V(X)$

Problem 2

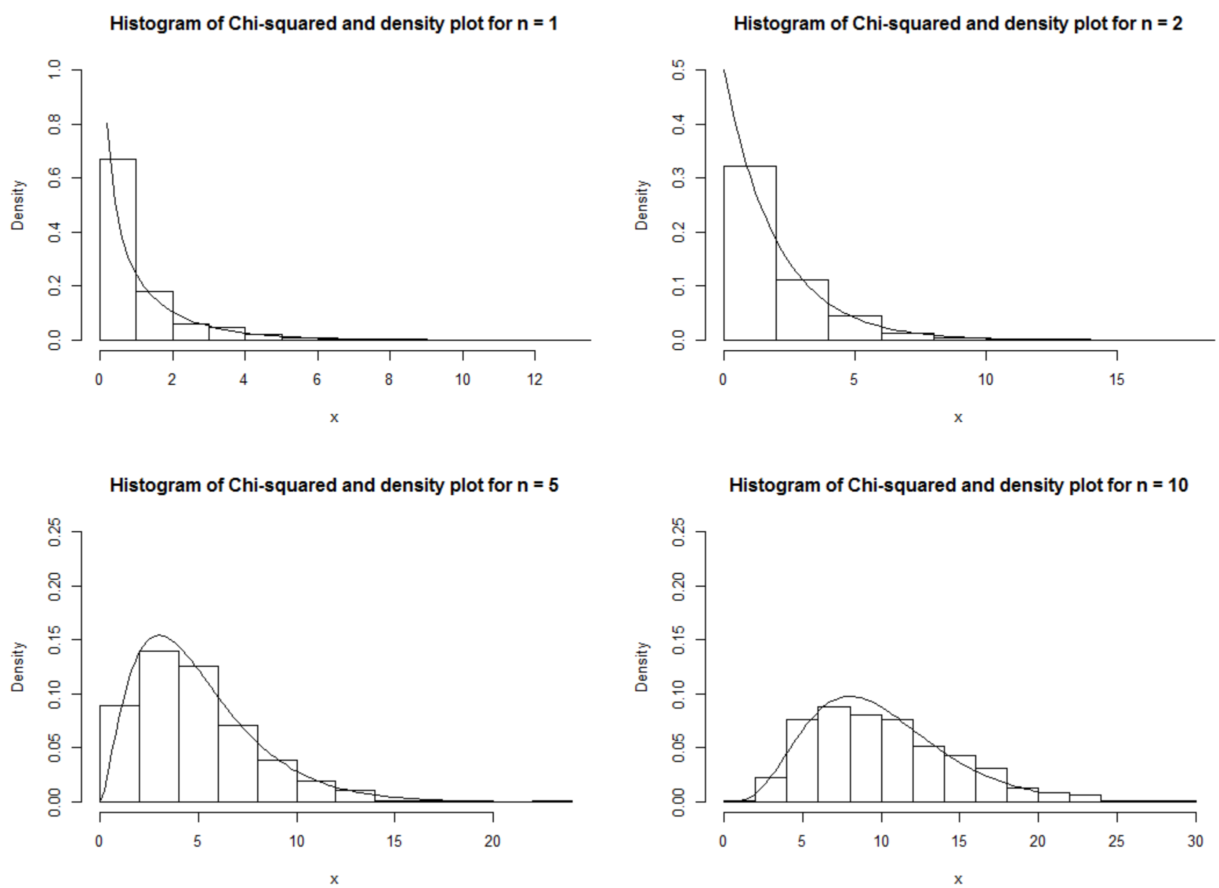
Using the tool of Monte Carlo simulation, illustrate the following result: if $X_1, ..., X_n \overset{iid}{\sim} N(0,1)$ then $\sum_{i=1}^{n} X_i^2 \sim \chi_n^2$.

- Note: Use different values of n, $n = 1, 2, 5, 10, ...$ then generate random samples of size of n from $N(0,1)$, compute the statistic. Repeat for a large number of times, say 1000, and plot the histogram of these sample statistics. To compare with χ_n^2, use the built-in density function in R **dchisq** and overlay the density curve on the histogram, use **lines** to add lines to existing graphs.

Solution
We can illustrate the following result through an R code shown below:

$$X_1, ..., X_n \overset{iid}{\sim} N(0,1)$$

$$\implies \sum_{i=1}^{n} X_i^2 \sim \chi_n^2$$



Problem 2 continued on next page. . .

```
#Q2)
par(mfrow=c(2,2))

#Used n = 1

n = 1000
x = matrix(0,1,n)

for(i in 1:n){
  #rdm<-sample(1:10,1)
  x[i] = sum(rnorm(1,0,1)^2)
}
hist(x, prob = TRUE,ylim =range(c(0,1)), main="Histogram of Chi-squared and density plot for n = 1")
a = seq(from = 0, to = 20,length.out = 100)
lines(a,dchisq(a,1), type = 'l')


#Used n = 2

n = 1000
x = matrix(0,1,n)

for(i in 1:n){
  #rdm<-sample(1:10,1)
  x[i] = sum(rnorm(2,0,1)^2)
}
hist(x, prob = TRUE,ylim =range(c(0,0.5)), main="Histogram of Chi-squared and density plot for n = 2")
a = seq(from = 0, to = 20,length.out = 100)
lines(a,dchisq(a,2), type = 'l')

#Used n = 5
n = 1000
x = matrix(0,1,n)

for(i in 1:n){
  #rdm<-sample(1:10,1)
  x[i] = sum(rnorm(5,0,1)^2)
}
hist(x, prob = TRUE,ylim =range(c(0,0.25)), main="Histogram of Chi-squared and density plot for n = 5")
a = seq(from = 0, to = 20,length.out = 100)
lines(a,dchisq(a,5), type = 'l')

#used n = 10
n = 1000
x = matrix(0,1,n)

for(i in 1:n){
  #rdm<-sample(1:10,1)
  x[i] = sum(rnorm(10,0,1)^2)
}
hist(x, prob = TRUE,ylim =range(c(0,0.25)), main="Histogram of Chi-squared and density plot for n = 10")
a = seq(from = 0, to = 20,length.out = 100)
lines(a,dchisq(a,10), type = 'l')
```

This shows how Chi-Squared perfectly models the histogram of the distribution above confirming the relation.

Problem 3

Let $X_1, ..., X_n \overset{iid}{\sim} Poisson(\lambda)$. For Poisson distribution, $E_\lambda(X) = \lambda$ and $V_\lambda(X) = \lambda$, therefore $\hat{X}$ and S^2 are both unbiased and consistent estimators of λ.

a. Consider a new estimator $\hat{g}(\bar{X}, S^2) = (\bar{X}S^2)^{1/2}$, use the Continuous Mapping Theorem, argue that $\hat{g}(\bar{X}, S^2)$ is a consistent estimator of λ.

b. Conduct Monte Carlo Simulation in R to generate sampling distribution of $\hat{X}, S^2$ and $\hat{g}(\bar{X}, S^2)$. Set the sample size n to be 100 and the parameter λ to be 1. Show the sampling distribution using histograms.[hint: the R function for random draws from poisson distribution is rpois].

c. Calculate bias, variance and Mean Square Error(MSE) for each of the three estimators and compare the results.

d. Vary the sample size to demonstrate the consistency of the third estimator $\hat{g}(\bar{X}, S^2)$.

Solution

a. Given: $plim_{n->\infty}\bar{X} = \lambda$ and $plim_{n->\infty}S^2 = \lambda$ for a Poisson Distribution charectarized by $X_1, ..., X_n \overset{iid}{\sim} Poisson(\lambda)$

   From the Continuous Mapping Theorem, we know that,

   $$X_n \overset{p}{->} a \,\&\, Y_n \overset{p}{->} b \implies X_nY_n \overset{p}{->} ab$$

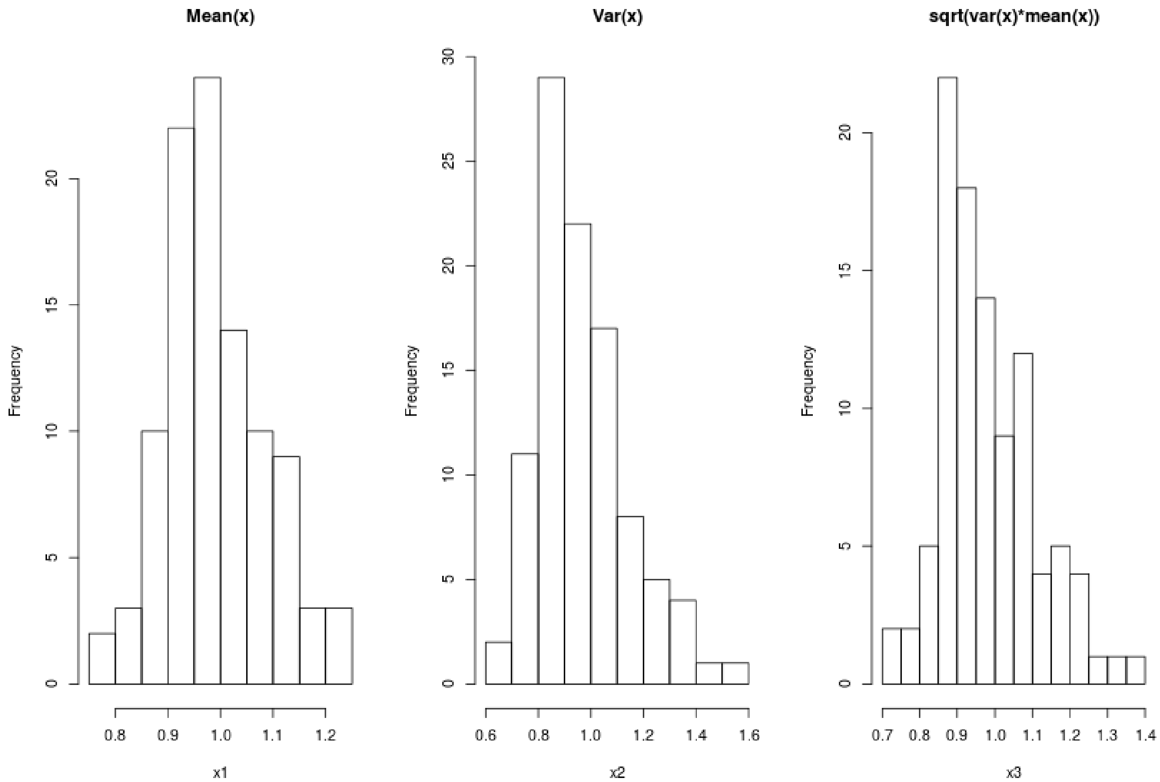
   Thus, from the Continuous Mapping Theorem, we can see that if $plim_{n->\infty}\bar{X} = \lambda$ and $plim_{n->\infty}S^2 = \lambda$, then $\bar{X}S^2 \overset{p}{->} \lambda^2$.

   We also know that $\hat{g}(\bar{X}, S^2) = (\bar{X}S^2)^{1/2}$. So, we can say that $\bar{X}S^2 = \hat{g}^2$. Also, since we know that $\bar{X}S^2 \overset{p}{->} \lambda^2$, we get that $\hat{g}(\bar{X}, S^2) = \lambda$.

   We also know that if $plim_{n->\infty}T_n = \theta$, then T_n is a consistent estimator of θ.

   Thus, from all of the above information, we can say that since $plim_{n->\infty}\hat{g}(\bar{X}, S^2) = \lambda$, $\hat{g}$ is a consistent estimator of λ

b. The sampling distributions can be represented by using histograms as follows. The R code for performing the Monte Carlo simulation is also given.



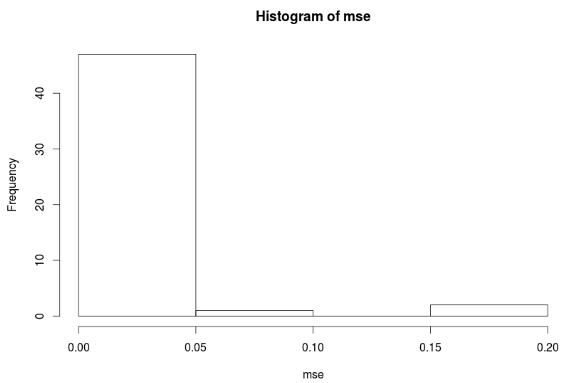

```
#Q3 Part b)
ghat<- function(x){
  a <- mean(x)*var(x)
  a<- sqrt(a)
  return(a)
}

x1 = matrix(0,1,100)
x2 = matrix(0,1,100)
x3 = matrix(0,1,100)

for(i in 1:100){
  y = rpois(100,lambda = 1)
  x1[i] = mean(y)
  x2[i] = var(y)
  x3[i] = ghat(y)
}
par(mfrow=c(1,3))
hist(x1, breaks = 10, main="Mean(x)")
hist(x2, breaks = 10, main="Var(x)")
hist(x3, breaks = 10, main="sqrt(var(x)*mean(x))")
```

c. Here we compute the **bias, variance and Mean Squared Error(MSE)** for each of the 3 distributions x1,x2 & x3 obtained from the Monte Carlo Simulation above.

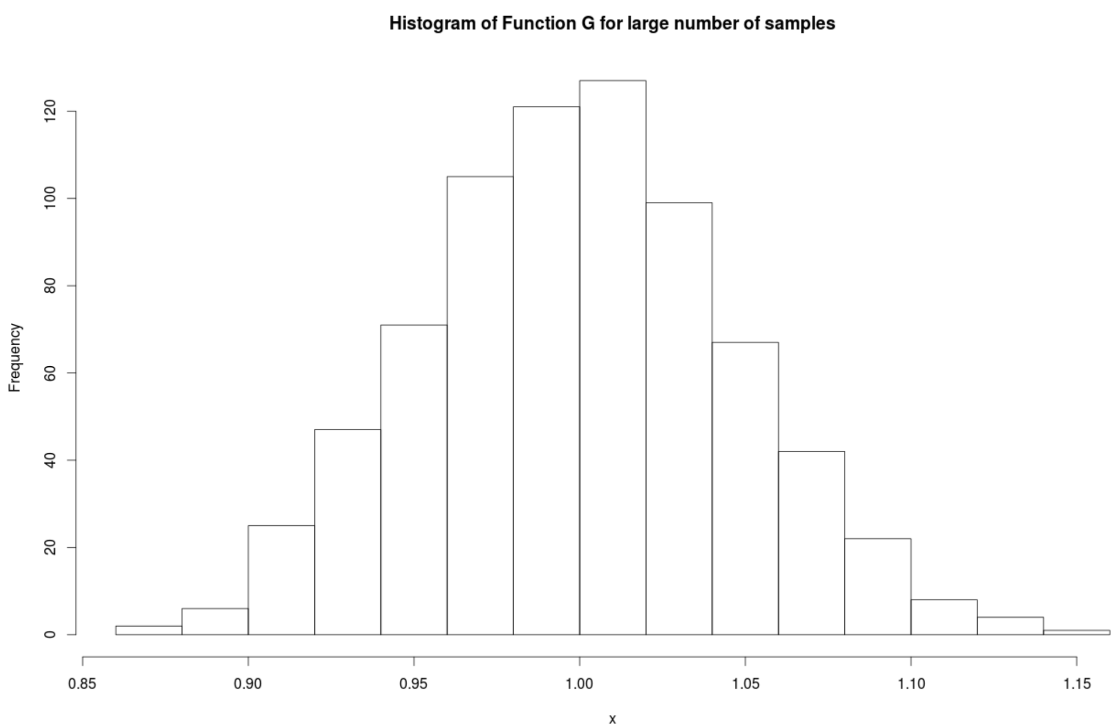
```
> ##Q3 Part c)
> ##bias for x1
> mean(x1)-1
[1] -0.0038
> ##variance for x1
> var(as.numeric(x1))
[1] 0.009355111
> ##MSE for x1
> (mean(x1)-1)^2+var(as.numeric(x1))
[1] 0.009369551
>
> ##bias for x2
> mean(x2)-1
[1] -0.0344202
> ##variance for x2
> var(as.numeric(x2))
[1] 0.03116609
> ##MSE for x2
> (mean(x2)-1)^2+var(as.numeric(x2))
[1] 0.03235084
>
> ##bias for x3
> mean(x3)-1
[1] -0.02150436
> ##variance for x3
> var(as.numeric(x3))
[1] 0.01568117
> ##MSE for x3
> (mean(x3)-1)^2+var(as.numeric(x3))
[1] 0.01614361
>
```

We can see that the MSE is the least for estimator $x1$ which is sampling the mean of the poisson distribution. With higher values of 'n', we should get the value closer to 0 in all 3 of the estimators since they are all consistent.

d. For this question, I iterate through and generate random integers between 1 and 1000 50 times to get a sample size and each time evaluating $\hat{g}(\bar{X}, S^2)$ and its Mean Squared Error and determining its value. If it is a consistent estimator, we should get the values of MSE to be close to 0 on average. The Histogram of MSE for multiple n is shown as follows:




```
> ghat<- function(x){
+   a <- mean(x)*var(x)
+   a<- sqrt(a)
+   return(a)
+ }
> n = sample.int(1000,50, replace = T)
> mse = numeric(50)
> for(j in 1:50){
+   x = numeric(n[j])
+   for(i in 1:n[j]){
+     y = rpois(n[j],lambda = 1)
+     x[i] = ghat(y)
+   }
+   ##MSE for x given lambda = 1
+   mse[j] = (mean(x)-1)^2+var(as.numeric(x))
+ }
> hist(mse)
> mean(mse)
[1] 0.01574822
>
```



We get to see that most of the values of the MSE lie on 0 except for a few outliers. We also notice how most of the values of the function $\hat{g}(\bar{X}, S^2)$ centers around the value 1, which is the value of the parameter λ. Thus, we can show that this estimator is consistent.

Problem 4

The dataset hwk2_data.txt contains an i.i.d. sample from an exponential distribution $f_\theta(x) = \theta^{-1}e^{-x/\theta}$, with unknown parameter θ. The goal is to find the MLE of θ.

a. Write down the Likelihood function $L(\theta)$ and the log-Likelihood function $l(\theta)$ for this problem.

b. Derive the score function $l'(\theta)$, and compute $\hat{\theta}_{MLE}$. You can use R to compute any statistics that are involved in this calculation.

c. Derive $l''(\theta)$

d. Write an R function that runs the Newton's algrithm and use it to solve $\hat{\theta}_{MLE}$.

e. Use R built-in function **nlm** to solve the problem.

f. Compare your results of b), d) and e).

g. Report an estimate of the information $I(\theta)$ and using **nlm** and estimate the standard error of $\hat{\theta}_{MLE}$.

Solution

a. For an i.i.d. with $f(x) = \theta^{-1}e^{-x/\theta}$, with unknown parameter θ, the likelihood function

$$L(X,\theta) = \prod_{i=1}^{n} f(x_i,\theta) = \prod_{i=1}^{n} \theta^{-1}e^{-x/\theta}$$
$$= \theta^{-\sum_{i=1}^{n} 1} e^{-1/\theta \sum x}$$
$$= \theta^{-n} e^{-\frac{\sum x}{\theta}}$$

$\implies$ log-likelihood function $l(\theta) = log(L(X,\theta)) = -nlog(\theta) - \frac{\sum x}{\theta}$

b. $l'(\theta) = \frac{\partial}{\partial\theta}(l(\theta))$

$$\frac{\partial}{\partial\theta}(log(L(X,\theta))) = \frac{\partial}{\partial\theta}(-nlog(\theta) - \theta^{-1}\sum x) = \frac{-n}{\theta} + \theta^{-2}\sum x$$

Thus,

$$l'(\theta) = \frac{-n}{\theta} + \frac{\sum x}{\theta^2} = u(\theta)$$

To compute $\hat{\theta}_{MLE}$, we set $l'(\theta) = 0$ (to maximize $L(X,\theta)$)

$$\frac{-n}{\theta} + \frac{\sum x}{\theta^2} = 0$$
$$\implies \hat{\theta} = \frac{\sum x}{n}$$

Using this formula in R using the data from hwk2_data.txt, we get the value of $\hat{\theta}$ as 0.623.

```
> x0 = sum(hwk2_data)/length(hwk2_data$V1)
> x0
[1] 0.623
>
```

c.

$$l''(\theta) = \frac{\partial}{\partial\theta}[\frac{\partial}{\partial\theta}L(X,\theta)]$$

$$= \frac{\partial}{\partial\theta}[\frac{-n}{\theta} + \frac{\sum x}{\theta^2}] = [\frac{n}{\theta^2} - \frac{-2\sum x}{\theta^3}]$$

$$\implies l''(\theta) = [\frac{n}{\theta^2} - \frac{-2\sum x}{\theta^3}]$$

d. The function I wrote to compute Newton's algorithm in R was as follows:

$$\hat{\theta}^{k+1} = \hat{\theta}^k + I(\hat{\theta}^k|X)^-1S(\hat{\theta}^k|X)$$

where

$$I(\theta) = -\frac{\partial^2}{\partial\theta^2}l(X,\theta) = \frac{2\sum x}{\theta^3} - \frac{n}{\theta^2}$$

$$s(\hat{\theta}, X) = l'(\hat{\theta}) = -\frac{n}{\theta} + \frac{1}{\theta^2}\sum x$$

```
> start<-median(hwk2_data$V1)
> newton<-function(start){
+   N = 20
+   x = hwk2_data$V1
+   for(i in 1:N){
+     x1<-start + ((sum(x)/(start^2))-N/start)/(2*sum(x)/(start^3)-N/(start^2))
+     if(abs(x1-start)<0.0001) break
+     start<-x1
+   }
+   return(x1)
+ }
> result<-newton(start)
> result
[1] 0.623
>
```

e. Using the nlm function in R we get

```
> N = length(hwk2_data$V1)
> f<-function(mu,x){
+   N*log(mu) + sum(x)/mu
+ }
> start<-median(hwk2_data$V1)
> out<-nlm(f, start, x = hwk2_data)
> mu_hat<-out$estimate
> mu_hat
[1] 0.6229996
>
```

f. We can see that using all of the 3 methods the values of $\hat{\theta}_{MLE}$ is the same. So, for the function $f(X, \theta)$ we can see that $\hat{\theta}_{MLE}$ is essentially the mean of the distribution. So, the maximum likelihood estimator of this data is the mean.

g. So, the information matrix $I(\theta)$ and the standard error of the $\hat{\theta}_{MLE}$ is found in the R code shown below.

```
> f<-function(mu,x){
+   20*log(mu) + sum(x)/mu
+ }
> start<-median(hwk2_data$V1)
> out<-nlm(f, start, x = hwk2_data, hessian = TRUE)
>
> var<-solve(out$hessian)
> #Standard Error of Theta_MLE
> std_error<-sqrt(diag(var))
> std_error
[1] 0.1393516
> #Information Matrix
> I<- -out$hessian
> I
          [,1]
[1,] -51.49632
>
```

Problem 5

The dataset **hwk2_data2.txt** contains an i.i.d. sample from a Gamma distribution with parameter $\theta = (\alpha, \beta)$, $f_\theta(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} X^{\alpha-1} e^{/\beta}$, where $\Gamma(\alpha)$ is the Gamma function (gamma in R)

a. Find the MLE of (α, β) using nlm, and estimate the standard errors using the Hessian matrix.

b. Find the MLE of (α, β) using optim, and estimate the standard errors using the Hessian matrix.

c. The true values of α and β that are used to generate this data are 5 and 1. Construct a 95% confidence interval for the true values based on the results in (b) and comment on it.

d. Comment on how the choice of initial values are important for this exercise.

Solution

a. We find the MLE using nlm in this case. The results are shown below:

```
>
> xvec<-hwk2_data2$V1
> l<-length(xvec)
>
> fn<-function(theta){
+   sum( theta[1]*log(theta[2]) + log(gamma(theta[1])) + (1-theta[1])*log(xvec) + xvec/theta[2] )
+ }
> alpha<-mean(xvec)^2/var(xvec)
> beta<-mean(xvec)/var(xvec)
> theta<-c(alpha,beta)
> out<-nlm(fn, theta, hessian=TRUE)
Warning messages:
1: In log(theta[2]) : NaNs produced
2: In nlm(fn, theta, hessian = TRUE) :
  NA/Inf replaced by maximum positive value
3: In log(theta[2]) : NaNs produced
4: In nlm(fn, theta, hessian = TRUE) :
  NA/Inf replaced by maximum positive value
> # MLE of alpha and beta
> out$estimate
[1] 5.7732726 0.9583415
> var1<-solve(out$hessian)
> #Standard Error using the hessian matrix
> std_error<-sqrt(diag(var1))
> std_error
[1] 1.1255272 0.1952307
>
```

We see that the estimated values of α and β are 5.77 and 0.95 respectively, using nlm.

b. Using optim, we get the following result:

```
>
> ##Q5 Part b)
> xvec<-hwk2_data2$V1
>
> fn<-function(theta){
+   sum( theta[1]*log(theta[2]) + log(gamma(theta[1])) + (1-theta[1])*log(xvec) + xvec/theta[2] )
+ }
> alpha<-mean(xvec)^2/var(xvec)
> beta<-mean(xvec)/var(xvec)
> theta<-c(alpha,beta)
> out1<-optim(theta, fn, hessian = TRUE)
> #MLE of alpha and beta using optim
> out1$par
[1] 5.7730255 0.9582715
> var2<-solve(out$hessian)
> #Standard Error using the hessian matrix
> std_error<-sqrt(diag(var2))
> std_error
[1] 1.1255272 0.1952307
>
```

The value of α and β are the same in both the cases.

c. Given that the true values of *alpha* and *beta* are 5 and 1, we construct the confidence interval as shown below:

```
> ##Q5 Part c)
> xvec<-hwk2_data2$V1
>
> fn<-function(theta){
+   sum( theta[1]*log(theta[2]) + log(gamma(theta[1])) + (1-theta[1])*log(xvec) + xvec/theta[2] )
+ }
> alpha<-mean(xvec)^2/var(xvec)
> beta<-mean(xvec)/var(xvec)
> theta<-c(alpha,beta)
> out1<-optim(theta, fn, hessian = TRUE)
> #MLE of alpha and beta using optim
> theta<-out1$par
> var2<-solve(out1$hessian)
> #Standard Error using the hessian matrix
> std_error<-sqrt(diag(var2))
>
>
> conf_interval<-0.95
> theta1<-c(5,1)
> critical_value<-qnorm((1+conf_interval)/2)
> for(i in 1:2){
+   print(theta1[i] + c(-1,1)*critical_value*std_error[i])
+ }
[1] 2.802161 7.197839
[1] 0.6188842 1.3811158
>
```

We see that the value of 5.77 for α and 0.95 for β fall well within the range of the 95% confidence interval. Thus we see that the values of α and β that we get from nlm and optim are very close to the true values and can be used as god estimators.

d. Since we are using an optimization algorithm, the simplest estimators are the method of moments estimators. So, with the 2 parameters being α and β, the obvious choices are $E_{\alpha,\beta}(x) = \frac{\alpha}{\beta}$ and $Var_{\alpha,\beta}(x) = \frac{\alpha}{\beta^2}$.

   Solving the simultaneous equations, we get the values of α and β as:

$$\alpha = \frac{E_{\alpha,\beta}(x)}{Var_{\alpha,\beta}} \quad \& \quad \beta = \frac{E_{\alpha,\beta}(x)}{Var_{\alpha,\beta}(x)^2}$$

   The method of moments isn't always applicable, and it doesn't necessarily produce good estimators. Maximum likelihood estimators are asymptotically efficient and Method of moment estimators generally aren't. But they provide good enough starting points for maximum likelihood estimation.

   The choice of the initial values are important because we want to reduce the number of steps in the iteration process. The closer the initial values are the actual estimate, the faster we can get the results.