SS3858 Tutorial Lecture 2

Yifan Li, yli2763@uwo.ca

2017-01-18

Introduction

Today's tutorial will:

- Discuss any possible questions you have met in lecture notes, R codes and so on.
- Revisit the identifiability of bivariate normal distribution mentioned last tutorial.
- Show how to simulate different estimators and compare their performance in R.

Two candidate estimators in Gamma distribution

Consider $X_i \overset{i.i.d.}{\sim} Gamma(2, \lambda)$ $i = 1, 2, \cdots, n$, then we have $\mathrm{E}[X_i] = \frac{2}{\lambda}$ and $\mathrm{Var}[X_i] = \frac{2}{\lambda^2}$. Therefore, we can construct two possible estimators for λ.

$$\hat{\lambda}_{n,1} = \frac{2}{\bar{X}_n}$$

$$\hat{\lambda}_{n,2} = \sqrt{\frac{2}{S_n^2}}$$

where $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2$.

Why do we do the simulation?

- Sometimes we not only want to know the expectation and variance of the estimators, but also their distribution to get more properties.
- In practice, when the estimators have complex form, we can rarely derive the exact distribution function. A feasible approach is to do the simulation to see the distribution.
- For example, The first one is a simple transformation and we can get its density function easily (by using the Jacobian determinant). However, the second one is not so easy to get its pdf.

Compute the estimators in Gamma distribution

```r
lam <- 4
n <- 50
y <- rgamma(n, 2, lam)
lamHat1 <- function(x)  2/mean(x)
lamHat2 <- function(x) sqrt(2/(var(x)))
EstGamma <- function(y) c(lamHat1(y), lamHat2(y))
#write a function to simplify your codes
EstGamma(y)
```

```
## [1] 4.263091 3.916611
```

Simulation for the estimators in Gamma distribution

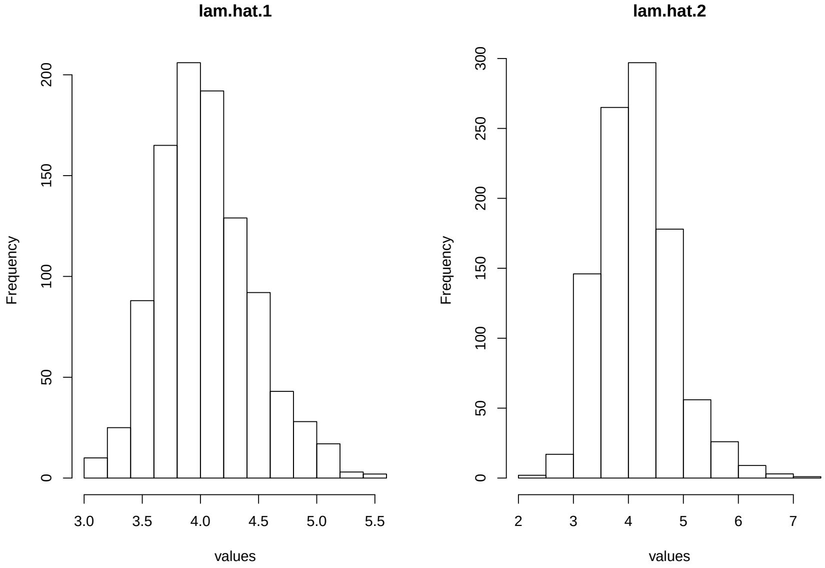
```
K <- 1000 #simulation times
EstSeq1 <- replicate(K, EstGamma(rgamma(n, 2, lam)))
EstSeq2 <- t(EstSeq1) #transpose
colnames(EstSeq2) <- c("lam.hat.1" , "lam.hat.2")
```

Simulation Results

```
head(EstSeq2)
```

```
##      lam.hat.1 lam.hat.2
## [1,]  3.968014  3.708758
## [2,]  3.852746  4.054655
## [3,]  4.261308  3.999977
## [4,]  4.434049  5.391828
## [5,]  3.700436  4.016281
## [6,]  4.901220  5.634008
```

Visualization of the distributions



Comparison from various aspects

```r
summary(EstSeq2)
```

```
##    lam.hat.1       lam.hat.2
##  Min.   :3.012   Min.   :2.048
##  1st Qu.:3.752   1st Qu.:3.684
##  Median :4.008   Median :4.092
##  Mean   :4.045   Mean   :4.158
##  3rd Qu.:4.290   3rd Qu.:4.565
##  Max.   :5.440   Max.   :7.054
```

Comparison from various aspects

```r
apply(EstSeq2, 2, var)
```

```
## lam.hat.1 lam.hat.2
## 0.1662991 0.4498651
```

```r
mse <- function(x) mean((x-lam)^2)
apply(EstSeq2, 2, mse)
```

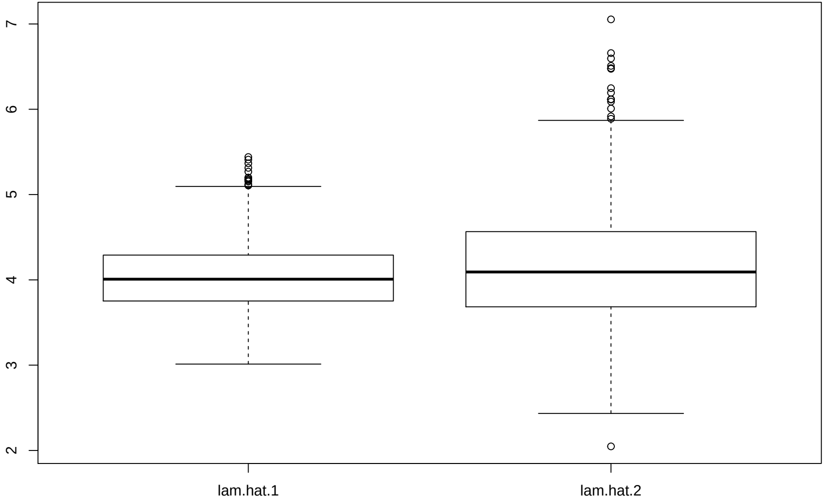
```
## lam.hat.1 lam.hat.2
## 0.1681539 0.4743503
```

Use a table to summarize the results

	lam.hat.1	lam.hat.2
true value	4.000	4.000
# sims	1000.000	1000.000
MC mean	4.045	4.158
MC bias	0.045	0.158
MC relative bias	0.011	0.039
MC standard deviation	0.408	0.671
MC MSE	0.168	0.474
MC relative efficiency	1.000	0.354

Visualization of the comparison

```
boxplot(EstSeq2)
```



Other ways to compare . . .

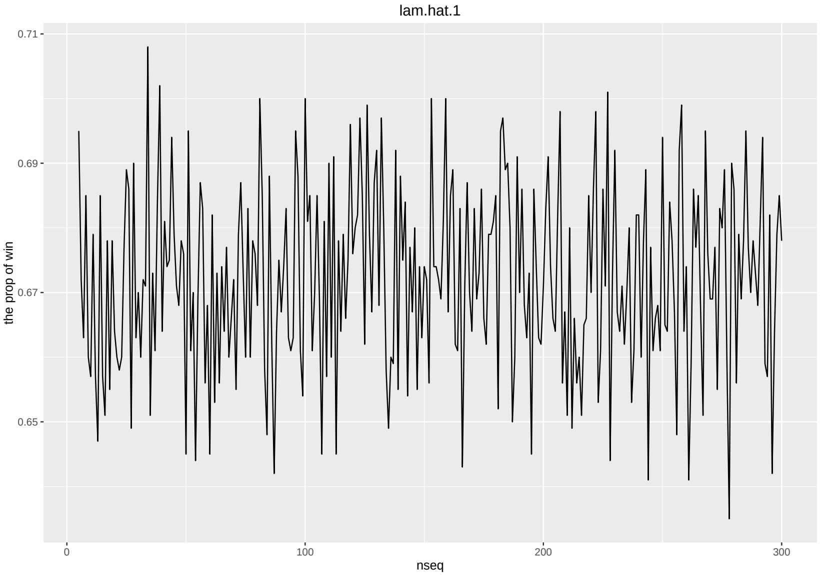
Which estimator is closer to the true value?

```r
# codes used in class
f <- function(y, lambda = 4){
  ifelse( abs(y[1] - lambda) <=   abs(y[2] - lambda) ,  1
}
prop1 <- apply(EstSeq2, 1, f); table(prop1)/K
```

```
## prop1
##     1     2
## 0.676 0.324
```

According to the frequency, the first one is closer to the true value.

Consider various sample sizes



Consider various sample sizes

```
summary(prop1sum[,1])
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.6350   0.6610   0.6720   0.6721   0.6830   0.7080
```

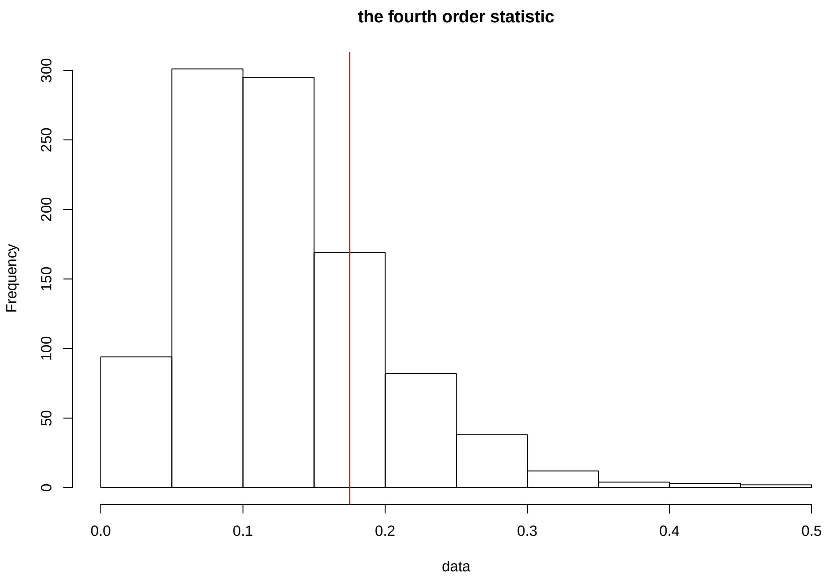
In total, lam.hat.1 perform better for different sample size n.

Simulate the order statistic

Simulation can be used to describe the distribution in other situations.

```
lam2 <- 10
xseq <- rexp(5, lam2)
x.four <- sort(xseq)[4]
order.stat <- function(order, xseq) sort(xseq)[order]
K <- 1000
```

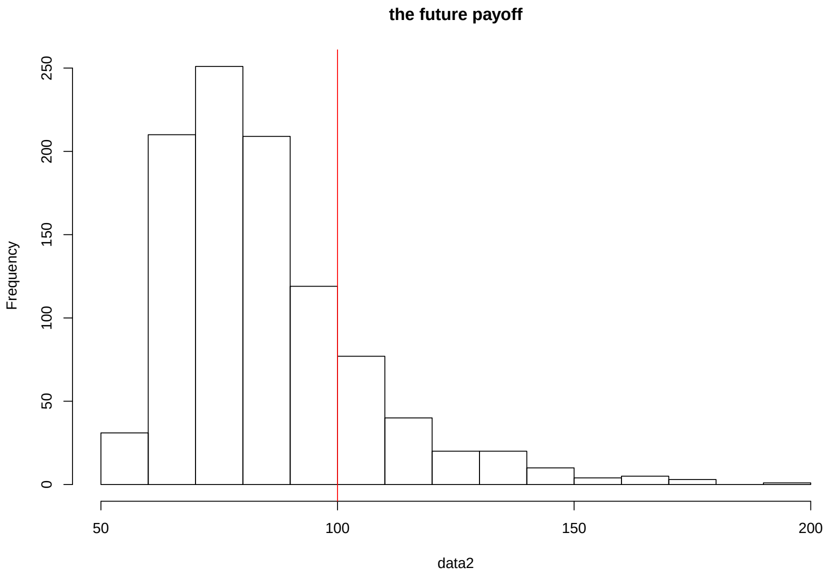
Results


the fourth order statistic
Frequency
0
50
100
150
200
250
300
0.0
0.1
0.2
0.3
0.4
0.5
data

Simulate the stock price

```
#the stock price today is 50 dollars.
S.today <- 50; xseq <- rexp(5, lam2)
#the rate of return
payoff <- function(xseq) S.today * exp(sum(xseq))
```

Results



the future payoff
Frequency
0
50
100
150
200
250
50
100
150
200
data2