Homework 8

Jing Leng (GSI: Jiahe)

November 17, 2014

1

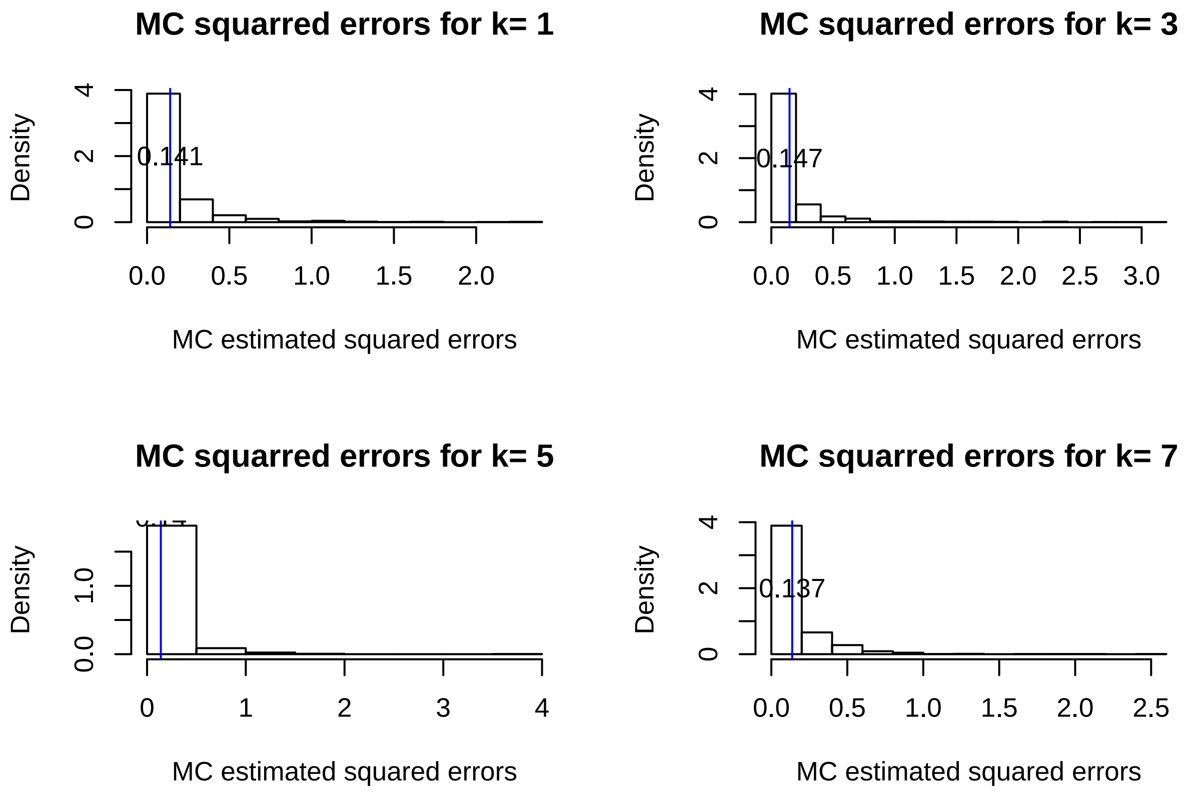
```
n = 20; mu = 0;
kseq = c(1,3,5,7)
nMC = 1000 #Monte Carlo sample size, i.e. number of trimmed means;
tmean = matrix(0, nMC, length(kseq))
#each column corresponds to a k, with 1000 Monte-Carlo trimmed means.
for (k in 1:length(kseq)){
  for (i in 1:nMC){
    sym = rcauchy(n)
    OStat = sort(sym)
    tmean[i,k] = median(OStat[((kseq[k]+1):(n-kseq[k]))])
  }
}

MSE_MC = apply(tmean, 2, function(v) mean((v - mu)^2))
MSE_MC
```

```
## [1] 0.1407 0.1474 0.1398 0.1374
```

```
squerr_MC = (tmean - mu)^2

par(mfrow = c(2,2))
for (k in 1:length(kseq)){
  hist(squerr_MC[,k], prob = TRUE, main = paste('MC squarred errors for k=',kseq[k]),xlab ="MC estimated
  abline(v = MSE_MC[k], col = 'blue')
  text(MSE_MC[k], 2, labels = round(MSE_MC[k],3))
}
```



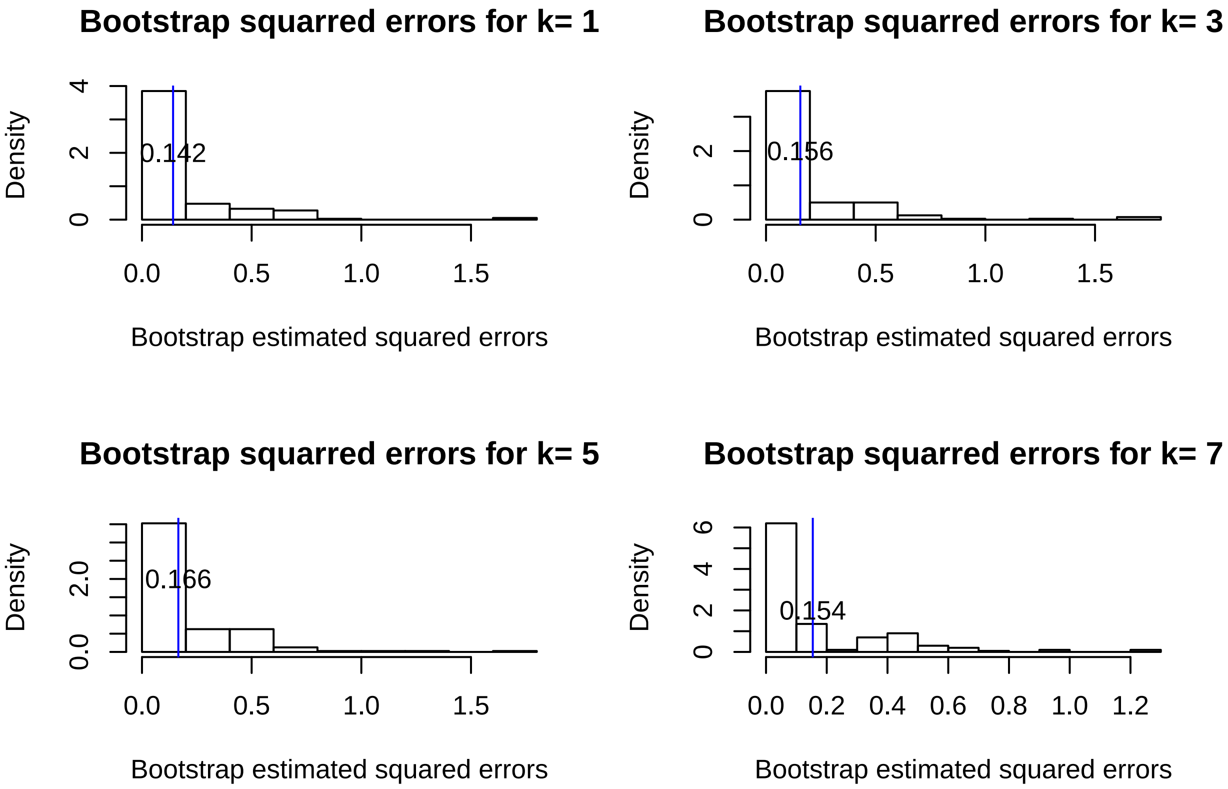
```
X = rcauchy(n)
B = 200
Vec = matrix(0, B, length(kseq))
OneSym = rcauchy(n)
TrueMu = median(OneSym)
for (k in 1:length(kseq)){
  for (i in 1:B){
    bsample = sample(OneSym, n, replace = TRUE)
    OStat = sort(bsample)
    Vec[i,k] = median(OStat[((kseq[k]+1):(n-kseq[k]))])
  }
}

MSE_Bt = sapply(1:length(kseq), function(k) mean((Vec[,k] - TrueMu)^2))
MSE_Bt
```

```
## [1] 0.1419 0.1565 0.1657 0.1541
```

```
squerr_Bt = sapply(1:length(kseq), function(k) (Vec[,k] - TrueMu)^2)

par(mfrow = c(2,2))
for (k in 1:length(kseq)){
  hist(squerr_Bt[,k], prob = TRUE, main = paste('Bootstrap squarred errors for k=',kseq[k]),xlab ="Boots
  abline(v = MSE_Bt[k], col = 'blue')
  text(MSE_Bt[k], 2, labels = round(MSE_Bt[k],3))
}
```



2

estimate for mu is: $\hat{\mu} = \frac{\sum_k \ln x_k}{n}$

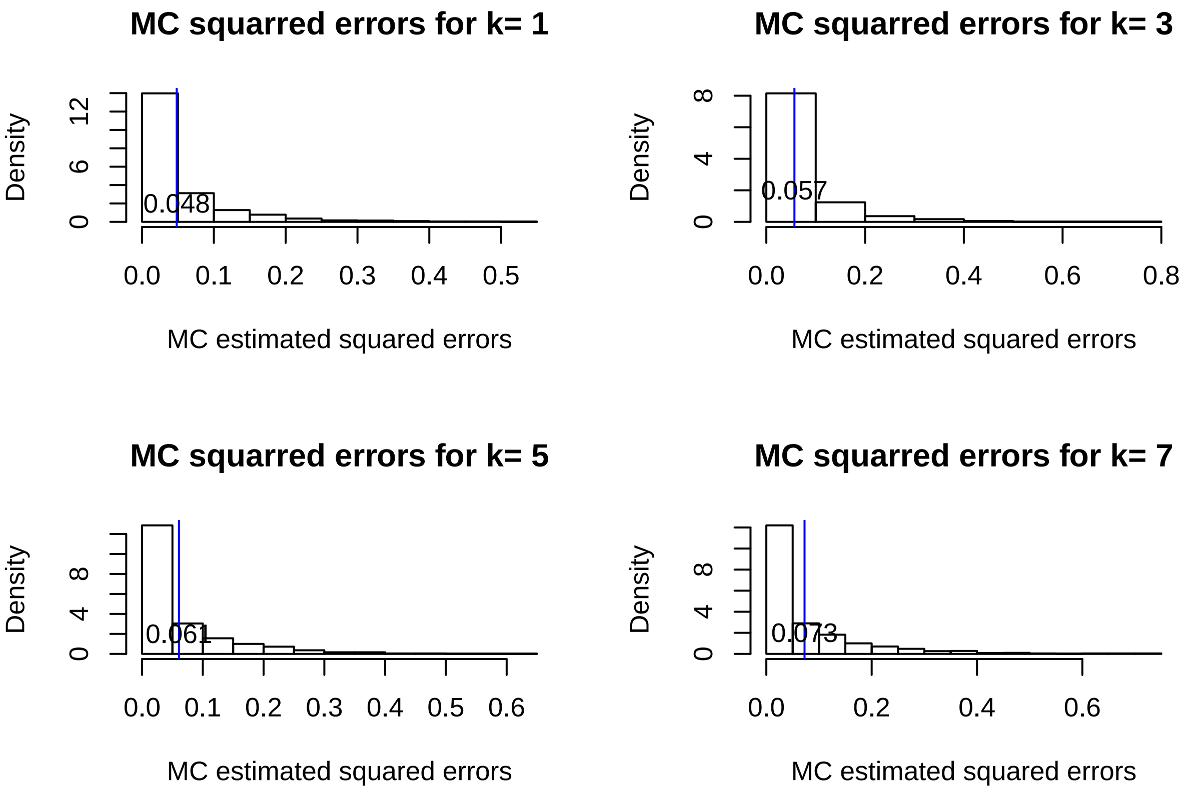
```
n = 20; mu = 0;
kseq = c(1,3,5,7)
nMC = 1000 #Monte Carlo sample size, i.e. number of trimmed means;
tmean = matrix(0, nMC, length(kseq))
#each column corresponds to a k, with 1000 Monte-Carlo trimmed means.
for (k in 1:length(kseq)){
  for (i in 1:nMC){
    sym = rlnorm(n)
    OStat = sort(sym)
    tmean[i,k] = mean(log(OStat[((kseq[k]+1):(n-kseq[k]))]))
  }
}

MSE_MC = apply(tmean, 2, function(v) mean((v - mu)^2))
MSE_MC
```

```
## [1] 0.04824 0.05693 0.06079 0.07257
```

```
squerr_MC = (tmean - mu)^2

par(mfrow = c(2,2))
for (k in 1:length(kseq)){
  hist(squerr_MC[,k], prob = TRUE, main = paste('MC squarred errors for k=',kseq[k]),xlab ="MC estimate
  abline(v = MSE_MC[k], col = 'blue')
  text(MSE_MC[k], 2, labels = round(MSE_MC[k],3))
}
```



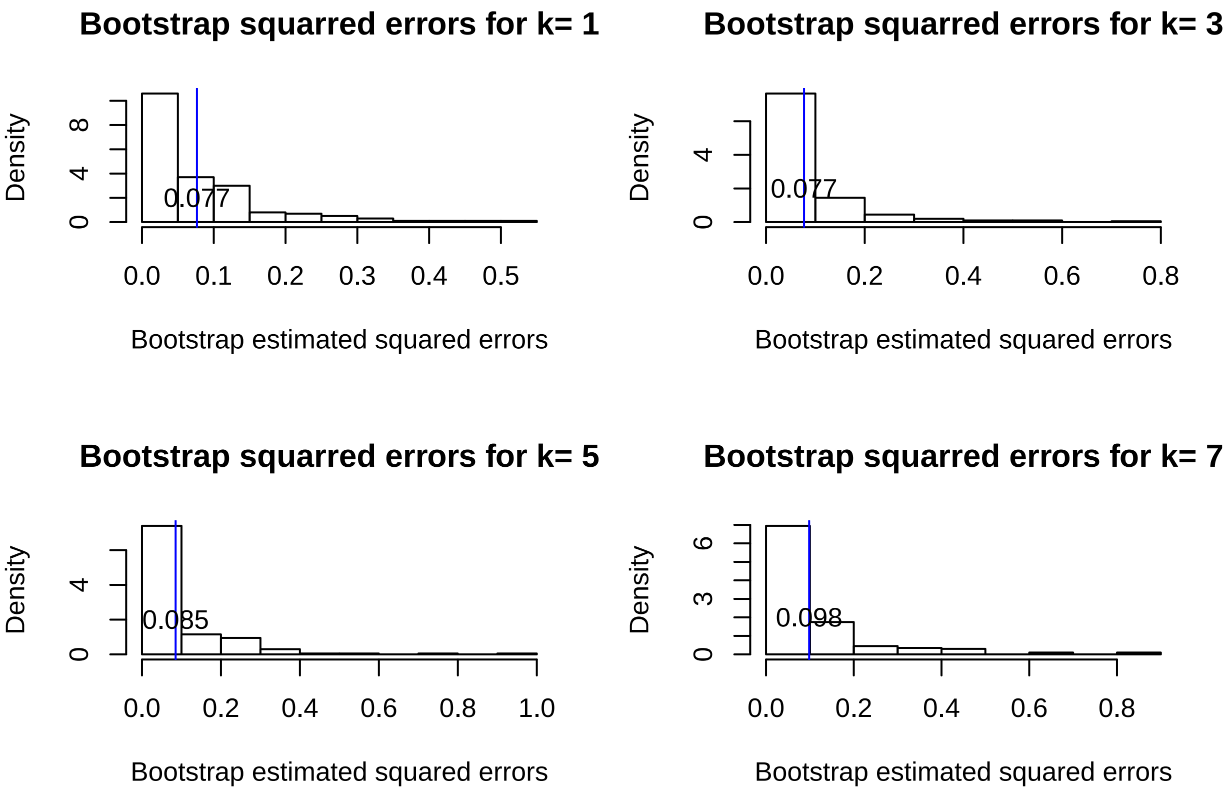
```
B = 200
Vec = matrix(0, B, length(kseq))
OneSym = rlnorm(n)
TrueMu = mean(log(OneSym))
for (k in 1:length(kseq)){
  for (i in 1:B){
    bsample = sample(OneSym, n, replace = TRUE)
    OStat = sort(bsample)
    Vec[i,k] = mean(log(OStat[((kseq[k]+1):(n-kseq[k]))]))
  }
}

MSE_Bt = sapply(1:length(kseq), function(k) mean((Vec[,k] - TrueMu)^2))
MSE_Bt
```

```
## [1] 0.07659 0.07700 0.08517 0.09830
```

```
squerr_Bt = sapply(1:length(kseq), function(k) (Vec[,k] - TrueMu)^2)

par(mfrow = c(2,2))
for (k in 1:length(kseq)){
  hist(squerr_Bt[,k], prob = TRUE, main = paste('Bootstrap squarred errors for k=',kseq[k]),xlab ="Boots
  abline(v = MSE_Bt[k], col = 'blue')
  text(MSE_Bt[k], 2, labels = round(MSE_Bt[k],3))
}
```


Bootstrap squarred errors for k= 1
Density
Bootstrap estimated squared errors
Bootstrap squarred errors for k= 3
Density
Bootstrap estimated squared errors
Bootstrap squarred errors for k= 5
Density
Bootstrap estimated squared errors
Bootstrap squarred errors for k= 7
Density
Bootstrap estimated squared errors