MaximumLikelihoodEstimator

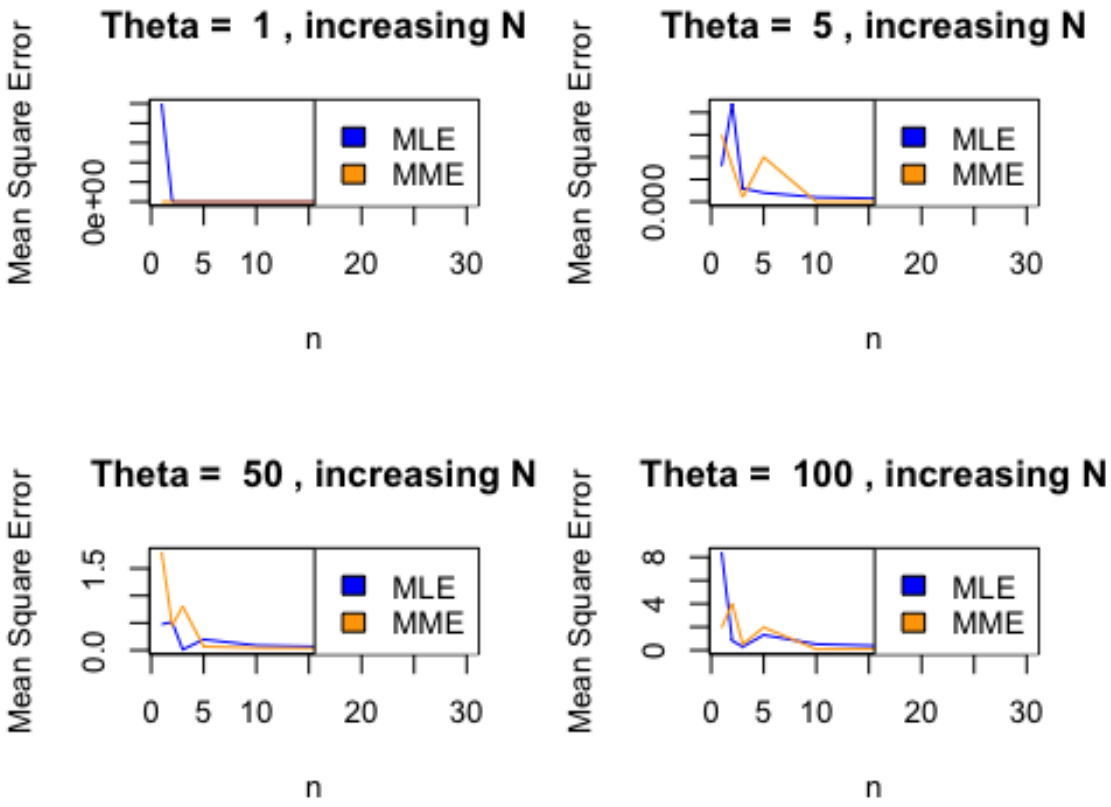
Part 1: Maximum likelihood estimator vs Monte Carlo.

```
set.seed(14)
n = c(1,2,3,5,10,30)
theta = c(1,5,50,100)

# (b) compute MSE using two methods
monte.mse <- function(n,theta){
  mse1 <- 0                          # for method 1
  mse2 <- 0                          # for method 2
  for (i in range(1000)){            # replicate 1000 simulations
    simX = runif(n,0,theta)
    theta1 = max(simX)               # method 1: max value
    theta2 = 2*mean(simX)            # method 2: 2*mean
    # sum up all MSE
    mse1 <- mse1 + (theta1-theta)^2
    mse2 <- mse2 + (theta2-theta)^2
  }
  result <- c(theta,n,mse1/1000,mse2/1000)
  names(result) <- c('Theta','N','MSE of MLE','MSE of MME')
  return(round(result,3))
}
# sample
monte.mse(8,10)
```

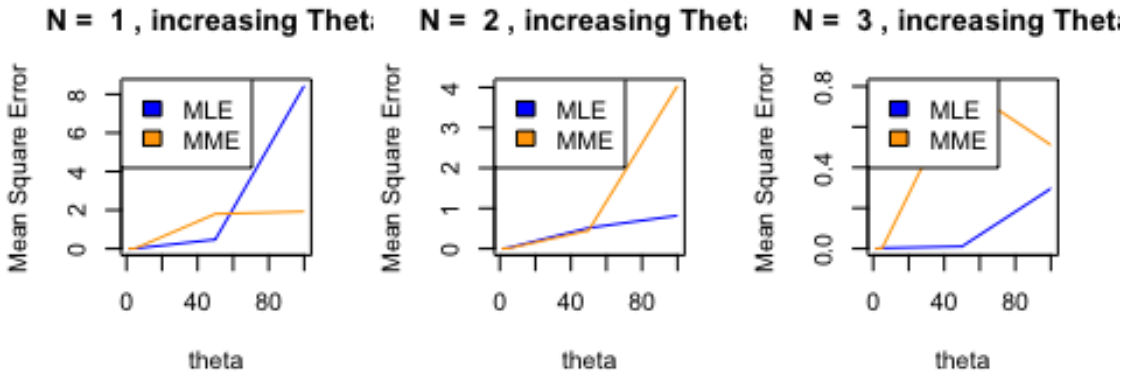
```
##      Theta          N MSE of MLE MSE of MME
##     10.000      8.000      0.001      0.014
```

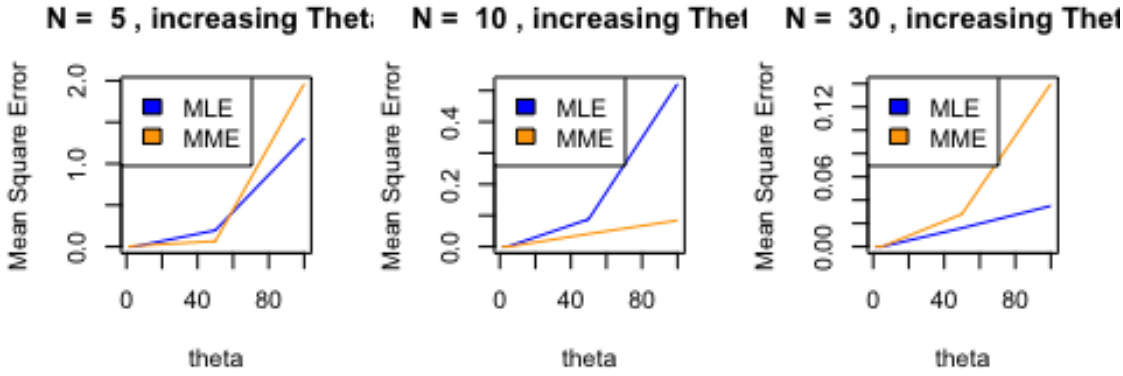
```
# (c) and (d)
# create matrix to store results
mleResult <- matrix(nrow=length(theta), ncol=length(n))
mmeResult <- matrix(nrow=length(theta), ncol=length(n))
for(i in 1:length(theta)){
  t <- theta[i]
  for(j in 1:length(n)){
    c <- n[j]
    temp <- monte.mse(c,t)
    mleResult[i,j] <- temp[3]
    mmeResult[i,j] <- temp[4]
  }
}

# plot results
par(mfrow=c(2,2))
for(i in 1:length(theta)){
  plot(n, mleResult[i,], main=paste("Theta = ", theta[i], ", increasing N"),
```

```
       ylab="Mean Square Error", type="l",col="blue",
       ylim=c(0,max(mleResult[i,],mmeResult[i,])))
  lines(n,mmeResult[i,], col="orange")
  legend("topright", c("MLE","MME"), fill=c("blue","orange"))
}
```




```
par(mfrow=c(2,3))
for(j in 1:length(n)){
  plot(theta, mleResult[,j], main=paste("N = ", n[j], ", increasing Theta"),
       ylab="Mean Square Error", type="l",col="blue",
       ylim=c(0,max(mleResult[,j],mmeResult[,j])))
  lines(theta,mmeResult[,j], col="orange")
  legend("topleft", c("MLE","MME"), fill=c("blue","orange"))
}
```






```
print(mleResult)
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
## [1,] 0.001 0.000 0.000 0.000 0.000 0.000
## [2,] 0.008 0.022 0.003 0.002 0.001 0.000
## [3,] 0.477 0.515 0.011 0.197 0.088 0.016
## [4,] 8.441 0.820 0.296 1.309 0.522 0.035
```

```
print(mmeResult)
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
## [1,] 0.000 0.000 0.000 0.000 0.000 0.000
## [2,] 0.015 0.008 0.001 0.010 0.000 0.000
## [3,] 1.801 0.452 0.803 0.066 0.041 0.028
## [4,] 1.925 4.041 0.511 1.963 0.084 0.140
```

Part 2: Log-likelihood function

```
sam = c(21.72, 14.65, 50.42, 28.78, 11.23)

# (b) estimate theta using equation from (a)
theta2 = length(sam)/sum(log(sam))

# (c) create log-likelihood function
```

```r
neg.loglh <- function(par,x){
  result <- sum(log(par/x^(par+1)))
  return(-result)
}
# calculate theta using OPTIM
est.theta <- optim(par=theta2,fn=neg.loglh,lower=0,
                   method="L-BFGS-B",hessian=TRUE,x=sam)
est.theta$par
```

```
## [1] 0.3233874
```

```r
# (d) get standard error from (c) and calculate 95% confidence interval
se <- sqrt(diag(solve(est.theta$hessian)))
se
```

```
## [1] 0.1446219
```

```r
CI = est.theta$par + c(-1,1)*1.96*se
CI
```

```
## [1] 0.03992856 0.60684630
```