Mini Project #3
David McCormick (DTM190000)
I am in a solo group.

Section 1

1. Part 1
   a. We can calculate the mean squared errors (MSE) for the method of moments (MOM) estimator and the maximum likelihood estimator (MLE) estimator against a fixed parameter theta using Monte Carlo simulations. The theta value will be set, and then we will use sample values to calculate the MLE and MOM, and then calculate the difference between each and the actual value of theta.
   b. In order to calculate the MSE's for the MLE and MOM estimators for a sample size of 1 and theta of 1, we would create 1000 replications of estimating the MLE and MOM with the aforementioned parameters. Afterwards, we would calculate the difference between the theta and estimator, and square it for each simulation. For n = 1, theta = 1, we get:
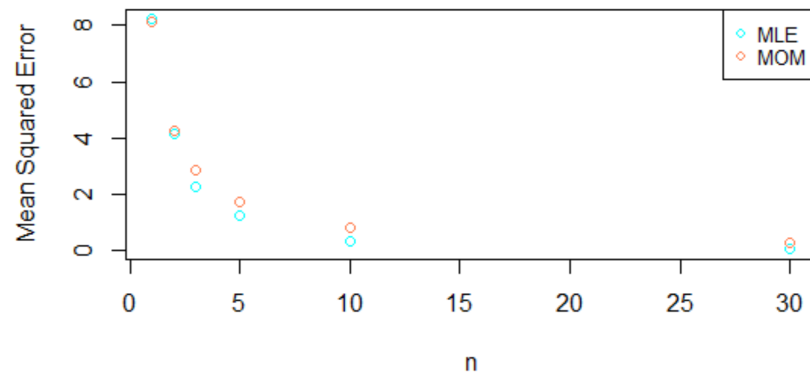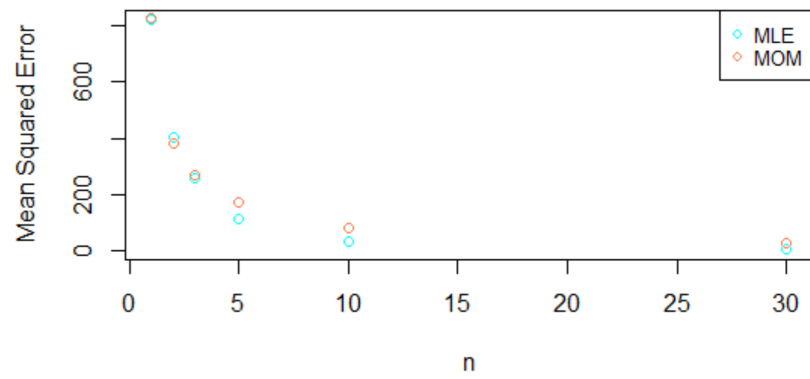
$$MSE_{MLE} = .320$$
$$MSE_{MOM} = .342$$

   c. The results for each combination of n and theta for n = [1,2,3,5,10,30] and theta = [1,5,50,100] are below.
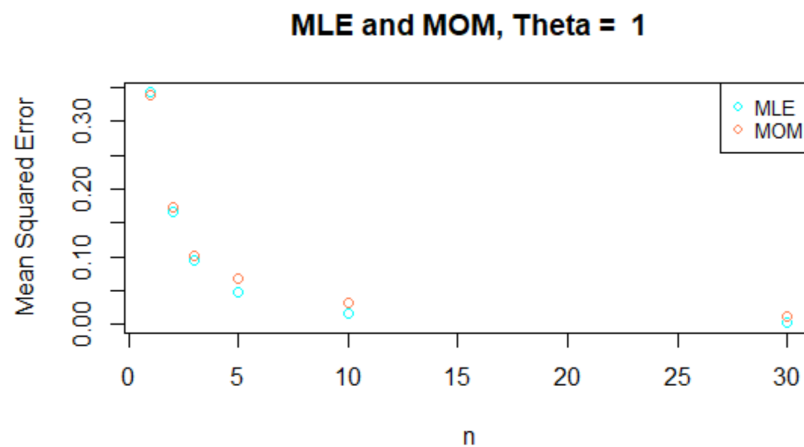
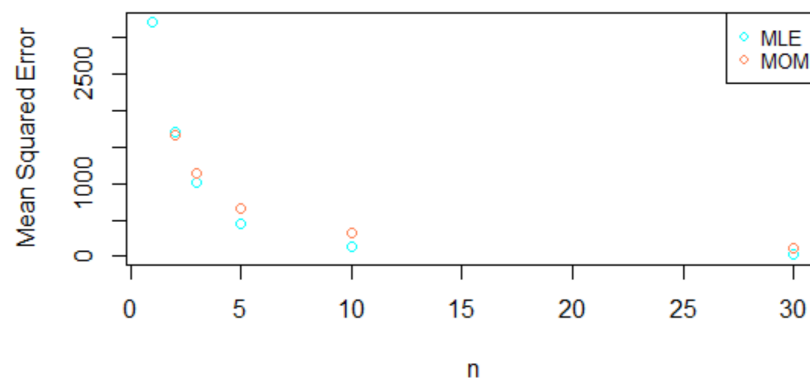# MLE and MOM, Theta = 5

MLE and MOM, Theta = 50

MLE and MOM, Theta = 100

d. From the results, we can determine that using the MLE method gives better results as sample size n increases. The MLE was consistently lower than the MOM estimator: for instance, for n = 30 and theta = 100 the MSE for MLE was 20.090 whereas the MSE for MOM was 114.620. The answer does not depend on theta because all of the graphs are similar for each different theta.

2. Part 2
   a. The maximum likelihood estimator for theta given the function is

   $$log(L(\theta)) = log(\Pi_{i=1}^{n}(\theta/x_i^{\theta+1}))$$

   $$log(L(\theta)) = log(\theta^n * \Pi_{i=1}^{n}(1/x_i^{\theta+1}))$$

   $$log(L(\theta)) = nlog(\theta) + \Pi_{i=1}^{n}log(x_i^{-\theta-1})$$

   $$log(L(\theta)) = nlog(\theta) - \theta * \Pi_{i=1}^{n}log(x_i) - \Pi_{i=1}^{n}log(x_i)$$

   $$log(L(\theta)) = n/\theta - \sum_{i=1}^{n}log(x_i)$$

   $$\theta = n/\sum_{i=1}^{n}log(x_i)$$

   b. Given the sample values, the formula would be (log = log base e):

   $$\theta = 5/(log(21.72) + log(14.65) + log(50.42) + log(28.78) + log(11.23)$$
   $$\theta = .323387$$

   c. By maximizing the log-likelihood function using the optim function in R, we get a value of .323 which agrees with using the formula.
   d. Our confidence interval would be:

   $$.323 \pm .145 * 1.96 = [.0399, .6068]$$

   This approximation would be useful because the true theta will lie in the interval 95 times out of 100 trials.

Section 2

```r
# Gets MLE and MOM from the same data
f1 <- function(n, theta) {
  x = runif(n, min = 0, max = theta)
  mle = max(x)
  mom = 2 * mean(x)
  return(c(mle, mom))
}

# Gets the MSE of MLE and MOM from N replications, estimating theta with n trials
f2 <- function(N, n, theta) {
  estimates = replicate(N, f1(n, theta))
  t1 = (estimates-theta)^2
  mle = mean(t1[1,])
  mom = mean(t1[2,])
  mle = round(mle, digits = 3)
  mom = round(mom, digits = 3)
  avgs = c(mle, mom)
  t2 = c(n, theta, avgs)
  return (t2)
}

n_arr = c(1,2,3,5,10,30)
theta_arr = c(1,5,50,100)

res = c()

for (theta in theta_arr){ # Performs all of the trials by iterating through arrays
  for (n in n_arr){
    res <- rbind(res, f2(1000, n, theta))
  }
}
colnames(res) = c("n","Theta","MLE Error","MOM Error")


# Graphs all of the information
# MLE and MOM for the same theta/n pairs
# MLE is in cyan
# MOM is in coral
for (iter in seq(1,length(theta_arr))) {
  plot(res[(6 * iter-5):(6 * iter), 1], res[(6 * iter-5):(6 * iter), 3], xlab="n", ylab="Mean Squared Error",
main=paste("MLE and MOM, Theta = ", res[(6 * iter), 2]), col="cyan")
  points(res[(6 * iter-5):(6 * iter), 1], res[(6 * iter-5):(6 * iter), 4], xlab="", ylab="", col="coral")
  legend("topright", legend=c("MLE", "MOM"), col=c("cyan", "coral"), pch=1:1,cex=0.8)

}



####################

f3 <- function(theta, dat) {

  result = length(dat) * log(theta) - (theta + 1) * sum(log(dat))

  return(-result)

}
```

```r
# Data
dat = c(21.72, 14.65, 50.42, 28.78, 11.23)


# Optimization function
res = optim(par=1, fn=f3, method="L-BFGS-B", hessian=TRUE, lower=.01, dat=dat)


# Getting standard error from hessian matrix
se = sqrt(1/res$hessian)


# Setting up confidence interval for 95% confidence
ci = c(res$par + 1 * se * qnorm(.975), res$par + -1 * se * qnorm(.975))
```