Mini Project 3

CS6313.001

Rutvij Shah (rds190000)

15 October 2021

Question 1

(a) MSE w/ Monte Carlo ~ Steps

1. Generate a random sample with the target parameter θ, in our case θ being the upper bound of a uniform distribution.
2. Apply the method of moments to calculate the parameter estimate $\hat{\theta_2}$ and maximum likelihood estimator to estimate the parameter $\hat{\theta_1}$
3. Calculate MSE by taking the mean of the squared error in estimation of the parameter θ over N simulations.

(b) Parameter estimation w/ Monte Carlo Given a sample size of $n = 10$ and param $\theta = 5$.

```r
mom_estimator <- function(sample) {
    sample_max = max(sample)
    return(sample_max)
}

mle_estimator <- function(sample) {
    sample_mean = mean(sample)
    return(2 * sample_mean)
}

sample_estimates <- function(sample_1) {
    theta_hat_1 = mle_estimator(sample_1)
    theta_hat_2 = mom_estimator(sample_1)
    return(c(theta_hat_1, theta_hat_2))
}

squared_error <- function(actual, predicted) {
    return((actual - predicted)^2)
}

mse_monte_carlo <- function(sample_size, theta, replications = 1000) {
    sum.sq1 = 0
    sum.sq2 = 0
    for (i in 1:replications) {
        sample = runif(sample_size, 0, theta)
        x = sample_estimates(sample)
        sq1 = squared_error(theta, x[1])
        sq2 = squared_error(theta, x[2])
        sum.sq1 = sum.sq1 + sq1
```

```r
        sum.sq2 = sum.sq2 + sq2
    }
    return(c(sum.sq1/replications, sq2/replications))
}

sample_size = 1
theta = 1
mean.squared.errors = mse_monte_carlo(sample_size, theta)
theta_hat_1.mse = mean.squared.errors[1]
theta_hat_2.mse = mean.squared.errors[2]
line1 <- paste("theta = ", theta, ", sample size = ", sample_size, sep = "")
line2 <- paste("Mean Squared Error for Maximum Likelihood Estimator (Sample Max): ",
    round(theta_hat_1.mse, 5), sep = "")
line3 <- paste("Mean Squared Eror for Method of Moments (twice the sample mean): ",
    round(theta_hat_2.mse, 5), sep = "")

cat(paste(line1, line2, line3, sep = "\n\n"))
```

```
theta = 1, sample size = 1

Mean Squared Error for Maximum Likelihood Estimator (Sample Max): 0.32901

Mean Squared Eror for Method of Moments (twice the sample mean): 0.00044
```

(c) Parameter estimation w/ Monte Carlo at different parameter values Graphical comparison of the effect of variation of θ and n on the MSE for Maximum Likelihood Estimator and Method of Moments.

```r
sample_sizes = c(1, 2, 3, 5, 10, 30)
paramater_values = c(1, 5, 50, 100)
df1 = data.frame(Sample_Size = numeric(), Parameter = numeric(), MSE_MLE = numeric(),
    MSE_MOM = numeric(), stringsAsFactors = F)
i = 1
for (n in sample_sizes) {
    for (theta in paramater_values) {
        mean.squared.errors = mse_monte_carlo(n, theta)
        theta_hat_1.mse = mean.squared.errors[1]
        theta_hat_2.mse = mean.squared.errors[2]
        df1[i, ] <- c(n, theta, theta_hat_1.mse, theta_hat_2.mse)
        i = i + 1
    }
}
library(data.table)
t1 = data.table(df1)
data.table::dcast(t1, Sample_Size ~ Parameter, value.var = "MSE_MLE")
```

```
   Sample_Size          1         5        50       100
1:           1 0.33920663 8.2387210 854.50016 3421.7600
2:           2 0.16512905 4.2324245 418.06450 1647.9110
3:           3 0.10945103 2.8822529 301.24973 1159.1055
4:           5 0.07296686 1.7508300 158.55524  679.0875
5:          10 0.03377168 0.8286621  84.57766  332.1229
6:          30 0.01117882 0.2910442  27.42631  105.7211
```

```r
data.table::dcast(t1, Sample_Size ~ Parameter, value.var = "MSE_MOM")
```

	Sample_Size	1	5	50	100
1:	1	2.389454e-04	1.154459e-02	0.126121839	7.326488e+00
2:	2	2.681405e-04	3.176109e-04	0.027583479	1.222032e-04
3:	3	1.202216e-04	4.567115e-05	0.268493591	2.558123e+00
4:	5	3.275429e-05	8.953546e-04	0.014838411	2.206607e-05
5:	10	3.743950e-06	9.308037e-05	0.113873645	5.184418e-03
6:	30	6.574598e-07	1.125842e-05	0.004016415	7.819573e-03

(d) Number patterns formatted correctly speak louder than lines & squiggles ("visualizations are for corporate drones") (Caveat: unless well planned and carefully constructed, the viz can be wiz) The beauty of statistics is within the numbers themselves, and when placed within a pivot table, we can see the patterns of change for our parameter estimates wrt to the Sample Size in the column header (y-axis) and the Parameter values in the row-header (x-axis). With their cross-section giving us the value for the parameter estimation mean squared error.

Table 1 is for MLE based parameter estimates, and we see that estimates generally improve with sample size irrespective of the parameter value. We also observe that estimates have a smaller mean squared errors for for smaller values of the parameter than for larger ones.

Table 2 is MOM based parameter estimates, and we can see the same general patterns of MSE wrt size & parameter values. But it can also be observed that the MOM estimates are consistently more accurate than those by MLE, with MSE being within 1000ths of the actual parameter value.

MOM outperforms MLE in this situation by a large margin. The dependence on n is larger than that on θ simply because for any parameter value, a sufficiently large sample of the population will always gives us a good estimate.

Question 2

(a) Expression for MLE of Parameter

1. Likelihood Function

$$L(\theta) = \prod_{i=1}^{n} \frac{\theta}{x_i^{\theta+1}}$$

2. Log of the Likelihood Function

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

$$log(L(\theta)) = log(\theta) - \sum_{i=1}^{n}[(\theta+1) \cdot log(x_i)]$$

3. Setting the partial derivative of the Log Likelihood to 0

$$\frac{\partial log(L(\theta))}{\partial \theta} = 0$$

$$\frac{1}{\theta} - \sum_{i=1}^{n} log(x_i) = 0$$

4. MLE Esitmator for θ

$$\hat{\theta}_{MLE} = \frac{1}{\sum_{i=1}^{n} log(x_i)}$$

(b) ML Estimation using Expression $n = 5$ and $x_1 = 21.72,\ x_2 = 14.65,\ x_3 = 50.42,\ x_4 = 28.78,\ x_5 = 11.23$

To Find: ML estimate of θ

```
ml_estimator <- function(sample) {
    log.sample = log(sample)
    sum.log.sample = sum(log.sample)
    return(1/sum.log.sample)
}

life <- c(21.72, 14.65, 50.42, 28.78, 11.23)
theta_hat_mle = ml_estimator(life)
cat(paste("The Maximum Likelihood Esitmate of theta is: ", round(theta_hat_mle, 5)))
```

```
The Maximum Likelihood Esitmate of theta is:  0.06468
```

```

neg.loglik.fun <- function(par, dat) {
    ll.x_i = log(par) - (par + 1) * sum(log(dat))
    result <- ll.x_i
    return(-result)
}
ml.est <- optim(par = 0.5, fn = neg.loglik.fun, method = "L-BFGS-B", lower = 1e-06,
    hessian = TRUE, dat = life)

cat(paste("The Numerical maximum likelihood Esitmate of theta is: ", ml.est$par))
```

(c) ML Estimation numerically using `optim()`

```
The Numerical maximum likelihood Esitmate of theta is:  0.0646826372694763
```

The answers match upto 4 decimal places and are in the ratio 1:1.00008 !!!

(d) Standard Error & CI The approximations seem to be fair since the value of theta seems to lie within the middle third of our 95% confidence interval and the interval is relatively small.

```
# The standard error for theta calc
se = sqrt(diag(solve(ml.est$hessian)))
line1.2 = paste("The approximate standard error for ", "maximum likelihood estimation of theta is: \n")
cat(paste(line1.2, round(se, 5)))
```

```
The approximate standard error for  maximum likelihood estimation of theta is:
 0.06467
```

```
# Assuming that theta follows a normal distribution

conf.int <- function(theta_hat, standard.error, alpha) {
    ci <- theta_hat + c(-1, 1) * qnorm(1 - (alpha/2)) * standard.error
    return(ci)
}

alpha = 0.05
ci = conf.int(ml.est$par, se, alpha)
cat(paste("The 95% confidence interval for theta is: ", " [", ci[1], ", ", ci[2],
    "]", sep = ""))
```

```
The 95% confidence interval for theta is:  [-0.0620626925119832, 0.191427967050936]
```

```r
############################################## Experimental Rough Work
############################################## #############
pdf <- function(theta, x) {
    if (x >= 1) {
        x.power.theta.plus.1 = x^(theta + 1)
        return(theta/x.power.theta.plus.1)
    } else {
        return(0)
    }
}

cdf <- function(theta, x) {
    pdf.at.theta <- function(y) {
        return(pdf(theta, y))
    }
    pdf.at.theta <- Vectorize(pdf.at.theta)
    return(integrate(pdf.at.theta, 1, x)$value)
}

library(GoFKernel)

cdf.at.theta = function(x) {
    return(cdf(0.064, x))
}

quantile.function = inverse(cdf.at.theta, lower = 1, upper = Inf)

##############################################
```