Simulation Study to Understand Sampling Distribution

Problem 2

Sulagna Barat | MDS202244

Problem assignee: Sulagna Baratl

Reviewer(s): Rohan Karthikeyan and Adarsha Mondal

Problem 2 : Simulation Study to Understand Sampling Distribution

Part A Suppose $X_1, X_2, \cdots, X_n \overset{iid}{\sim} Gamma(\alpha, \sigma)$, with pdf as

$$f(x|\alpha,\sigma) = \frac{1}{\sigma^\alpha \Gamma(\alpha)} e^{-x/\sigma} x^{\alpha-1}, \quad 0 < x < \infty,$$

The mean and variance are $E(X) = \alpha\sigma$ and $Var(X) = \alpha\sigma^2$. Note that `shape = ` α and `scale = ` σ.

1. Write a `function` in `R` which will compute the MLE of $\theta = \log(\alpha)$ using `optim` function in `R`. You can name it `MyMLE`
2. Choose `n=20`, and `alpha=1.5` and `sigma=2.2`
   (i) Simulate $\{X_1, X_2, \cdots, X_n\}$ from `rgamma(n=20,shape=1.5,scale=2.2)`
   (ii) Apply the `MyMLE` to estimate θ and append the value in a vector
   (iii) Repeat the step (i) and (ii) 1000 times
   (iv) Draw histogram of the estimated MLEs of θ.
   (v) Draw a vertical line using `abline` function at the true value of θ.
   (vi) Use `quantile` function on estimated θ's to find the 2.5 and 97.5-percentile points.
3. Choose `n=40`, and `alpha=1.5` and repeat the (2).
4. Choose `n=100`, and `alpha=1.5` and repeat the (2).
5. Check if the gap between 2.5 and 97.5-percentile points are shrinking as sample size `n` is increasing?

Hint: Perhaps you should think of writing a single `function` where you will provide the values of `n`, `sim_size`, `alpha` and `sigma`; and it will return the desired output.

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

1

.

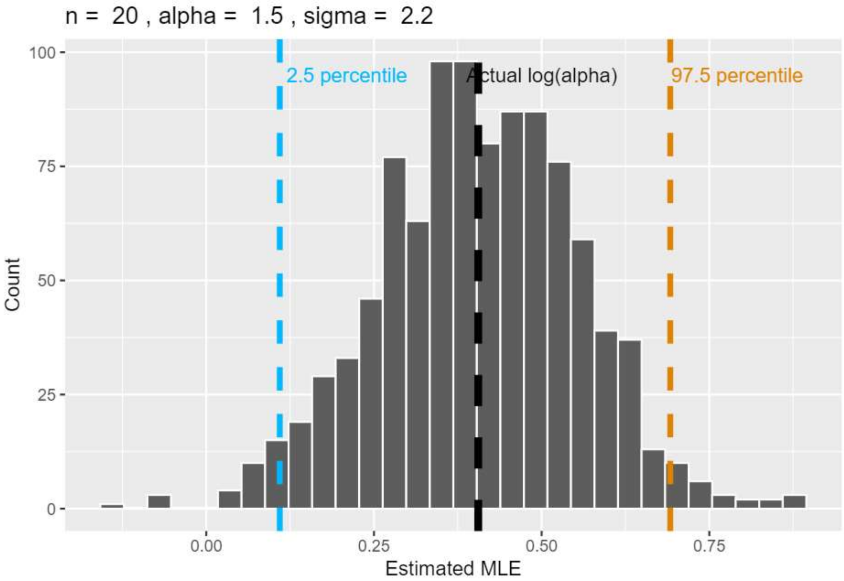
```
mle <- function(log_alpha, data, sigma) {
    l = sum(log(dgamma(data, shape = exp(log_alpha), scale = sigma)))
    # print(paste("l is ", l))
    return(-l)
}
MyMLE <- function(data, sigma) {
    log_alpha_initial <- log(mean(data)^2/var(data))
    # print(paste("log alpha initial is ", log_alpha_initial))
    estimator <- optim(log_alpha_initial,
                    mle,
                    data = data,
                    sigma = sigma)
    log_alpha_hat <- estimator$par
    return(log_alpha_hat)
}
```

```
get_estimates <- function(n, alpha, sigma) {
    estimates <- c()
    for (i in 1:1000) {
        samples <- rgamma(n, shape = alpha, scale = sigma)
        # print(paste("some of the samples are ", samples[1:5]))
        estimates <- append(estimates, MyMLE(data = samples, sigma = sigma))
    }
    return(estimates)
}
```

2.

```
n = 20
alpha = 1.5
sigma = 2.2
estimated_mle <- tibble(get_estimates(n = n, alpha = alpha, sigma = sigma))
colnames(estimated_mle) <- c("estimate")
perc_2.5 <- quantile(estimated_mle$estimate, probs = 0.025, names = FALSE)
perc_97.5 <- quantile(estimated_mle$estimate, probs = 0.975, names = FALSE)
estimated_mle %>%
    ggplot(aes(estimate)) +
    geom_histogram(color = "white", fill = "#5D5D5D") +
    geom_vline(xintercept = log(alpha),
               size = 2,
               linetype = "dashed") +
    annotate("text", label = "Actual log(alpha)", x = 0.5, y = 95, color = "black") +
    geom_vline(xintercept = perc_2.5,
               color = "#00B9FF", size = 1.5, linetype = "dashed") +
    annotate("text", label = "2.5 percentile", x = perc_2.5 + 0.1, y = 95, color = "#00B9FF") +
    geom_vline(xintercept = perc_97.5,
               color = "#E08304", size = 1.5, linetype = "dashed") +
    annotate("text", label = "97.5 percentile", x = perc_97.5 + 0.1, y = 95, color = "#E08304") +
    labs(title = paste("n = ", n, ", alpha = ", alpha, ", sigma = ", sigma),
         x = "Estimated MLE",
         y = "Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

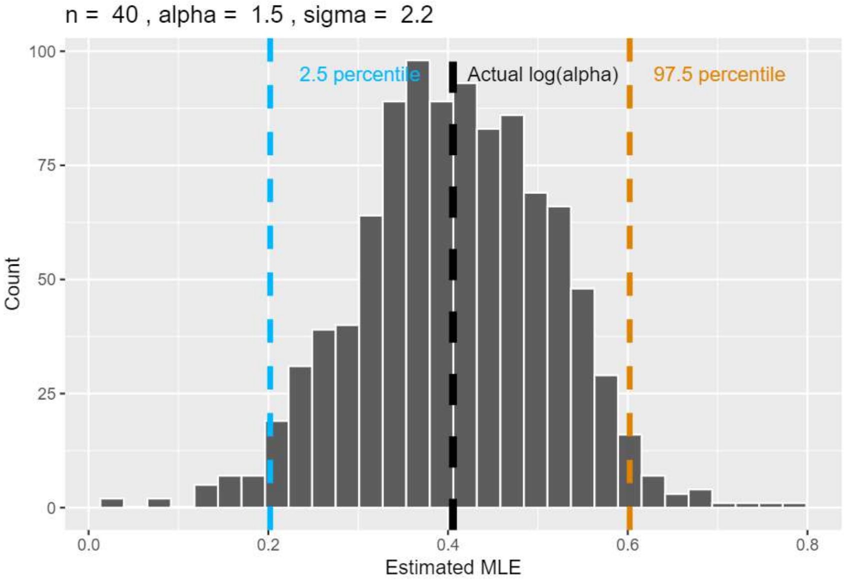


```
diff_20 <- perc_97.5 - perc_2.5
```

3.

```
n = 40
alpha = 1.5
sigma = 2.2
estimated_mle <- tibble(get_estimates(n = n, alpha = alpha, sigma = sigma))
colnames(estimated_mle) <- c("estimate")
perc_2.5 <- quantile(estimated_mle$estimate, probs = 0.025, names = FALSE)
perc_97.5 <- quantile(estimated_mle$estimate, probs = 0.975, names = FALSE)
estimated_mle %>%
    ggplot(aes(estimate)) +
    geom_histogram(color = "white", fill = "#5D5D5D") +
    geom_vline(xintercept = log(alpha),
               size = 2,
               linetype = "dashed") +
    annotate("text", label = "Actual log(alpha)", x = log(alpha) + 0.1, y = 95, color = "black") +
    geom_vline(xintercept = perc_2.5,
               color = "#00B9FF", size = 1.5, linetype = "dashed") +
    annotate("text", label = "2.5 percentile", x = perc_2.5 + 0.1, y = 95, color = "#00B9FF") +
    geom_vline(xintercept = perc_97.5,
               color = "#E08304", size = 1.5, linetype = "dashed") +
    annotate("text", label = "97.5 percentile", x = perc_97.5 + 0.1, y = 95, color = "#E08304") +
    labs(title = paste("n = ", n, ", alpha = ", alpha, ", sigma = ", sigma),
         x = "Estimated MLE",
         y = "Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

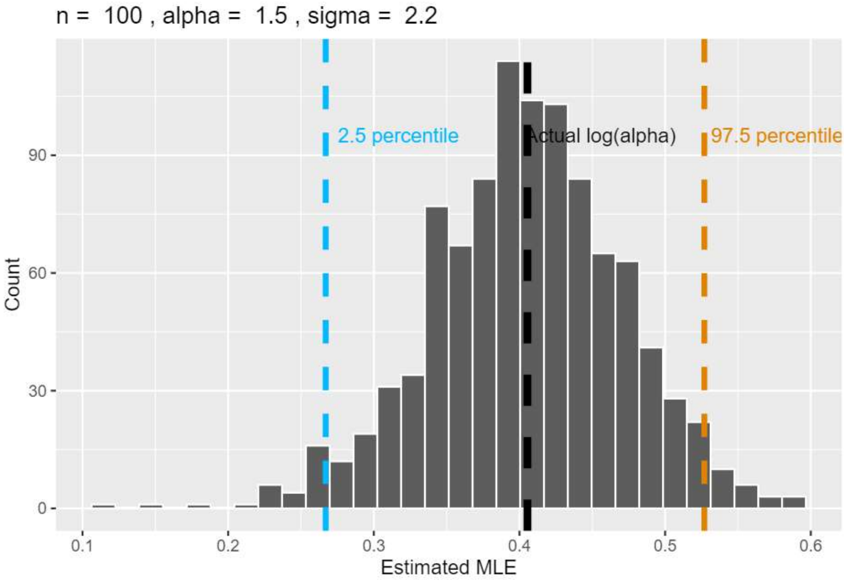


```
diff_40 <- perc_97.5 - perc_2.5
```

4.

```
n = 100
alpha = 1.5
sigma = 2.2
estimated_mle <- tibble(get_estimates(n = n, alpha = alpha, sigma = sigma))
colnames(estimated_mle) <- c("estimate")
perc_2.5 <- quantile(estimated_mle$estimate, probs = 0.025, names = FALSE)
perc_97.5 <- quantile(estimated_mle$estimate, probs = 0.975, names = FALSE)
estimated_mle %>%
    ggplot(aes(estimate)) +
    geom_histogram(color = "white", fill = "#5D5D5D") +
    geom_vline(xintercept = log(alpha),
               size = 2,
               linetype = "dashed") +
    annotate("text", label = "Actual log(alpha)", x = log(alpha) + 0.05, y = 95, color = "black") +
    geom_vline(xintercept = perc_2.5,
               color = "#00B9FF", size = 1.5, linetype = "dashed") +
    annotate("text", label = "2.5 percentile", x = perc_2.5 + 0.05, y = 95, color = "#00B9FF") +
    geom_vline(xintercept = perc_97.5,
               color = "#E08304", size = 1.5, linetype = "dashed") +
    annotate("text", label = "97.5 percentile", x = perc_97.5 + 0.05, y = 95, color = "#E08304") +
    labs(title = paste("n = ", n, ", alpha = ", alpha, ", sigma = ", sigma),
         x = "Estimated MLE",
         y = "Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
diff_100 <- perc_97.5 - perc_2.5
```

5.

```
diff_20
```

```
## [1] 0.5821454
```

```
diff_40
```

```
## [1] 0.4001029
```

```
diff_100
```

```
## [1] 0.2599853
```

We can see that the gap between the percentile points is decreasing as the sample size increases.