Asymptotic Sampling Distributions of MLEs

Cheng Peng

West Chester University

Contents

1 Introduction

Maximum Likelihood Estimation (MLE) is one of the most widely used methods for parameter estimation in statistical inference. Given a statistical model with probability density (or mass) function $f(x|\theta)$ and a random sample $\mathbf{X} = (X_1, X_2, \ldots, X_n)$, the likelihood function is defined as:

$$L(\theta|\mathbf{x}) = \prod_{i=1}^{n} f(x_i|\theta)$$

The maximum likelihood estimator $\hat{\theta}_{MLE}$ is the value that maximizes this likelihood function. Note that MLE is only a point estimate, the next step is to characterize the (asymptotic) sampling distribution of the MLE. Before discussing the sampling distribution of MLE, we review the basics of multivariate normal distribution with a primary focus on bivariate normal distribution.

2 Some Foundamental Concepts

To discuss the asymptotic sampling distribution of the MLE of parameters, we need to understand a few important concepts. We will not dive into deep weed of derivation. Instead, we will only explain these concept using minimum mathematics.

2.1 Illustration of Covariance Matrix

A **covariance matrix** provides a complete picture of the pairwise linear relationships between variables. Its **diagonal** represents the individual variance of each variable, while the **off-diagonal** entries indicate how pairs of variables move together. A positive value means the two variables **tend to increase together** (i.e., are positively correlated), while a negative value means **one tends to increase as the other decreases** (i.e., they are negatively correlated). The magnitude of each entry (relative to the variances) indicates the strength of the linear relationship. In short, a covariance matrix captures the **linear correlations** among the variables that define it.

All covariance matrices are symmetric - this because correlation between X and Y is the as between Y and X.

To have better understanding , we use an artificial example with three variables: **Height (H)**, **Weight (W)**, and **Age (A)**. Assume that their covariance matrix is:

$$\mathbf{\Sigma} = \begin{bmatrix} 25 & 12 & 8 \\ 12 & 36 & 6 \\ 8 & 6 & 16 \end{bmatrix}$$

Diagonals (in blue) Are Variances

Diagonal elements Σ_{ii} represent variances of individual variables:

- $\Sigma_{11} = 25$: Variance of **Height**

$$\sigma_H = \sqrt{25} = 5\text{cm}$$

- $\Sigma_{22} = 36$: Variance of **Weight**

$$\sigma_W = \sqrt{36} = 6\text{kg}$$

- $\Sigma_{33} = 16$: Variance of **Age**

$$\sigma_A = \sqrt{16} = 4\text{yr}$$

Off-diagonal Elements (Covariances in Green, Purple, Orange)

- Height-Weight Covariance: $\Sigma_{12} = \Sigma_{21} = 12$
  - **Sign**: Positive →, Taller people tend to weigh more
  - **Correlation coefficient**: $\rho_{HW} = \frac{12}{\sqrt{25 \cdot 36}} = \frac{12}{5 \times 6} = 0.4$, moderate positive correlation.
- Height-Age Covariance: $\Sigma_{13} = \Sigma_{31} = 8$
  - **sign**: Positive → Older people tend to be taller
  - **Correlation coefficient**: $\rho_{HA} = \frac{8}{\sqrt{25 \cdot 16}} = \frac{8}{5 \times 4} = 0.4$, moderate positive correlation
- Weight-Age Covariance: $\Sigma_{23} = \Sigma_{32} = 6$
  - **sign**: Positive but smallest magnitude.
  - **Correlation coefficient**: $\rho_{WA} = \frac{6}{\sqrt{36 \cdot 16}} = \frac{6}{6 \times 4} = 0.25$, weak positive correlation.

Some Extreme Cases

- **Diagonal Matrix (Uncorrelated Variables)**: All covariances = 0 → variables are uncorrelated

$$\mathbf{\Sigma}_{\text{diag}} = \begin{bmatrix} 25 & 0 & 0 \\ 0 & 36 & 0 \\ 0 & 0 & 16 \end{bmatrix}$$

- **High Correlation Matrix**: Large off-diagonals → stronger relationships

$$\mathbf{\Sigma}_{\text{high}} = \begin{bmatrix} 25 & 20 & 15 \\ 20 & 36 & 18 \\ 15 & 18 & 16 \end{bmatrix}$$

- **Mixed Sign**: Height-Weight negative correlation

$$\mathbf{\Sigma}_{\text{mixed}} = \begin{bmatrix} 25 & -12 & 8 \\ -12 & 36 & -6 \\ 8 & -6 & 16 \end{bmatrix}$$

2.2 Coveriance Matrix of MLE

As we know, maximum likelihood estimators (MLEs) of population parameters are derived from random samples. For instance, let $\hat{\alpha}$ and $\hat{\beta}$ denote the MLEs of α and β, respectively. Since $\hat{\alpha}$ and $\hat{\beta}$ are estimated from the same random sample, they are:

1. **Random variables**
2. **Generally correlated**

To fully characterize the correlation between $\hat{\alpha}$ and $\hat{\beta}$, we require their covariance matrix. Direct derivation of this covariance matrix is often analytically demanding. As an alternative approach, we introduce two key matrices derived from the log-likelihood function: the Hessian matrix and the Fisher information matrix.

2.2.1 Observed Hessian and Fisher Information Matrices

Let $\{x_1, x_2, \cdots, x_n\}$ be an i.i.d. random sample from a population with density function $f(x : \alpha, \beta)$. The likelihood function of α and β is

$$L(\alpha, \beta) = \prod_{i=1}^{n} f(x_i : \alpha, \beta)$$

The log-likelihood function is

$$l(\alpha, \beta) = \sum_{i=1}^{n} \log f(x_i : \alpha, \beta)$$

**

The Hessian matrix of α and β is defined to be

$$\mathcal{H}(\alpha, \beta) = \begin{bmatrix} \frac{\partial l(\alpha,\beta)}{\partial \alpha^2} & \frac{\partial l(\alpha,\beta)}{\partial \alpha \partial \beta} \\ \frac{\partial l(\alpha,\beta)}{\partial \beta \partial \alpha} & \frac{\partial l(\alpha,\beta)}{\partial \beta^2} \end{bmatrix}.$$

plugging the MLE to the Hessian matrix, we have **observed Hessian** matrix

$$\mathcal{H}(\hat{\alpha}, \hat{\beta}) = \left. \begin{bmatrix} \frac{\partial l(\alpha,\beta)}{\partial \alpha^2} & \frac{\partial l(\alpha,\beta)}{\partial \alpha \partial \beta} \\ \frac{\partial l(\alpha,\beta)}{\partial \beta \partial \alpha} & \frac{\partial l(\alpha,\beta)}{\partial \beta^2} \end{bmatrix} \right|_{\alpha=\hat{\alpha}\beta=\hat{\beta}}.$$

This means that for a given random sample from a population, we can find the **observed Hessian matrix** in two Steps:

- Finding the MLE of the population parameters.
- Substituting the MLE estimates into the Hessian matrix.

The **observed Fisher Information Matrix** based on a random sample with size n is equal to the **negative** of the **observed Hessian Matrix**. That is,

$$\mathcal{J}_n(\hat{\alpha}, \hat{\beta}) = -\mathcal{H}(\hat{\alpha}, \hat{\beta})$$

Note that $\mathcal{J}_n(\hat{\alpha}, \hat{\beta})$ is completely dependent on the random sample. The theoretical Fisher information matrix is denoted by

$$\mathbb{I}_n(\alpha, \beta) = \mathbb{E}[\mathcal{H}(\alpha, \beta)].$$

This theoretical Fisher information matrix will be used later in hypothesis testing.

2.2.2 Covariance Metrix of MLE

After some algebra (integral and derivative), the variance-covariance matrix of the MLE of α and β can be expressed in the following

$$\text{cov}(\hat{\alpha}, \hat{\beta}) = \mathbb{I}_n^{-1}(\alpha, \beta).$$

This means the covariance matrix of the MLE depends on the unknown parameters α and β. We therefore estimate it by applying the MLE plug-in principle:

The estimator $\mathbb{I}_n^{-1}(\hat{\alpha}, \hat{\beta})$ is used to estimate $\mathbb{I}_n^{-1}(\alpha, \beta)$, with $\hat{\alpha}$ and $\hat{\beta}$ being the MLEs of α and β.

Following the MLE plug-in principle, the estimator of the covariance matrix $\text{cov}(\hat{\alpha}, \hat{\beta})$ takes the following form:

$$\widehat{\text{cov}(\hat{\alpha}, \hat{\beta})} = \mathbb{I}_n^{-1}(\hat{\alpha}, \hat{\beta}) \approx \mathcal{J}_n^{-1}(\hat{\alpha}, \hat{\beta})$$

This means that the covariance matrix of the maximum likelihood estimator is approximated by the **inverse** of the **observed Hessian matrix**.

2.2.3 A Numerical Example

We consider an example based on the two-parameter Weibull distribution. The following are kep steps for finding the covariance matrix of the MLE.

1. Model and Data

Recall that the 2-parameter Weibull density function is given by

$$f(x; \alpha, \beta) = \frac{\beta}{\alpha} \left(\frac{x}{\alpha}\right)^{\beta-1} e^{-(x/\alpha)^{\beta}}, \; x > 0.$$

Assume the following small sample (for illustration) is from a Weibull population

$$\{x_1, \ldots, x_5\} = \{2.3,\ 1.4,\ 2.6,\ 3.1,\ 1.8\}.$$

2. Log-Likelihood

$$\ell(\alpha, \beta) = n \ln \beta - n\beta \ln \alpha + (\beta - 1) \sum_{i=1}^{n} \ln x_i - \sum_{i=1}^{n} \left(\frac{x_i}{\alpha}\right)^{\beta}.$$

The score equations (i.e., gradient functions)

$$\frac{\partial \ell(\alpha, \beta)}{\partial \alpha} = \frac{n}{\beta} + \sum_{i=1}^{n} \ln\left(\frac{x_i}{\alpha}\right) - \sum_{i=1}^{n} \left(\frac{x_i}{\alpha}\right)^{\beta} \ln\left(\frac{x_i}{\alpha}\right)$$

$$\frac{\partial \ell(\alpha, \beta)}{\partial \beta} = \frac{\beta}{\alpha}\left[\sum_{i=1}^{n} \left(\frac{x_i}{\alpha}\right)^{\beta} - n\right]$$

Setting the above score functions to zero, we have

$$\frac{\partial \ell}{\partial \alpha} = 0 \quad \Rightarrow \quad \hat{\alpha}(\beta) = \left[\frac{1}{n} \sum_{i=1}^{n} x_i^{\beta}\right]^{1/\beta}.$$

$$\frac{\partial \ell}{\partial \beta} = 0 \quad \Rightarrow \quad \frac{n}{\beta} - n \ln \hat{\alpha} + \sum \ln x_i - \sum \left(\frac{x_i}{\hat{\alpha}}\right)^{\beta} \ln\left(\frac{x_i}{\hat{\alpha}}\right) = 0.$$

3. Numerical MLE

Solving the profile likelihood yields:

$$\hat{\beta} \approx 4.0, \qquad \hat{\alpha} \approx 2.449.$$

To check the solution,

we would expect to check whether $\frac{\partial \ell(\alpha,\beta)}{\partial \alpha} \approx 0$ and $\frac{\partial \ell(\alpha,\beta)}{\partial \beta} \approx 0$ at $\hat{\alpha} \approx 2.449$ and $\hat{\beta} \approx 4.0$

We manually check

$$\frac{\partial \ell(\alpha, \beta)}{\partial \beta} \approx 0 \quad \text{implies} \quad \frac{\beta}{\alpha}\left[\sum_{i=1}^{n} \left(\frac{x_i}{\alpha}\right)^{\beta} - n\right] \approx 0,$$

which is equivalent to

$$\sum_{i=1}^{n} \left(\frac{x_i}{\alpha}\right)^{\beta} - n \approx 0$$

To check the above approximation, let $y_i = (x_i/\hat{\alpha})^{\hat{\beta}}$. We only need to check: $\sum y_i - n \approx 0$:

$$y_i \approx 0.777,\ 0.1068,\ 1.271,\ 2.566,\ 0.2918, \quad \sum y_i \approx 5.0126 \approx 5.$$

We can similarly check

$$\frac{\partial \ell(\alpha, \beta)}{\partial \alpha} \approx 0.$$

4. Observed Fisher Information Matrix

Define

$$t_i = \ln\left(\frac{x_i}{\hat{\alpha}}\right) \quad \text{and} \quad y_i = \left(\frac{x_i}{\hat{\alpha}}\right)^{\hat{\beta}}.$$

The Hessian $H(\hat{\alpha}, \hat{\beta})$ of the log-likelihood is:

$$\begin{aligned}
\frac{\partial^2 \ell}{\partial \alpha^2} &= -\frac{\hat{\beta}^2 n}{\hat{\alpha}^2}, \\
\frac{\partial^2 \ell}{\partial \beta^2} &= -\frac{n}{\hat{\beta}^2} - \sum_{i=1}^{n} y_i t_i^2, \\
\frac{\partial^2 \ell}{\partial \alpha \partial \beta} &= \frac{\hat{\beta}}{\hat{\alpha}} \sum_{i=1}^{n} y_i t_i.
\end{aligned}$$

Numerical values of MLE: ($\hat{\alpha} = 2.449$, $\hat{\beta} = 4.0$)

- $\sum y_i t_i^2 \approx 0.210971$.
- $\sum y_i t_i \approx 0.4820$.

Thus,

$$\begin{aligned}
H_{11} &= -\frac{4^2 \cdot 5}{2.449^2} \approx -13.338, \\
H_{22} &= -\frac{5}{16} - 0.210971 \approx -0.52347, \\
H_{12} = H_{21} &= \frac{4}{2.449} \times 0.4820 \approx 0.7875.
\end{aligned}$$

The **observed** Hessian matrix is

$$\mathcal{H}(\hat{\alpha} = 2.449, \hat{\beta} = 4.0) = \begin{pmatrix} -13.338 & 0.7875 \\ 0.7875 & -0.52347 \end{pmatrix}.$$

Therefore, the **Observed Fisher Information Matrix** is

$$\mathcal{J}(\hat{\alpha} = 2.449, \hat{\beta} = 4.0) = -\mathcal{H}(\hat{\alpha} = 2.449, \hat{\beta} = 4.0) = \begin{pmatrix} 13.338 & -0.7875 \\ -0.7875 & 0.52347 \end{pmatrix}.$$

5. Estimated Covariance Matrix of MLE

$$\widehat{\mathrm{Cov}}(\hat{\alpha}, \hat{\beta}) = \mathcal{J}^{-1}(\hat{\alpha}, \hat{\beta}).$$

$$\det(\mathcal{J}) = (13.338)(0.52347) - (0.7875)^2 \approx 6.980 - 0.620 = 6.360.$$

Therefore,

$$\mathcal{J}^{-1}(\hat{\alpha}, \hat{\beta}) = \frac{1}{6.360}\begin{pmatrix} 0.52347 & 0.7875 \\ 0.7875 & 13.338 \end{pmatrix}.$$

$$\widehat{\text{Var}}(\hat{\alpha}) \approx 0.0823, \quad \widehat{\text{Var}}(\hat{\beta}) \approx 2.097, \quad \text{and} \quad \widehat{\text{Cov}}(\hat{\alpha}, \hat{\beta}) \approx 0.1238.$$

Finally, the covariance matrix id given by

$$\boxed{\widehat{\text{Cov}} = \begin{pmatrix} 0.0823 & 0.1238 \\ 0.1238 & 2.097 \end{pmatrix}}.$$

3 Bivariate Normal Distribution of MLE

Recall that the sample mean ($\bar{x}$), as an estimate of the population mean, serves as the maximum likelihood estimator (MLE) for distributions including the normal and Poisson. By applying the Central Limit Theorem (CLT), we have the following **asymptotic sampling distribution**

$$\bar{x} \to N(\mu, \sigma^2)$$

Where μ and σ^2 are the population mean and variance, respectively. How would we characterize the **joint asymptotic sampling distribution** of MLE $(\bar{x}, s^2)$ for the parameters (μ, σ^2)? We need multivariate normal distribution to characterize the sampling distribution of the MLE of population parameters.

Since the density function of a joint distribution with more than two variables becomes very complicated, we use the vector representation of the multivariate normal distribution. For convenience, we will use the **trivariate normal distribution** as an example. Recall our artificial example with three variables: **Height (H)**, **Weight (W)**, and **Age (A)**. Assume their covariance matrix is:

$$\mathbf{\Sigma} = \begin{bmatrix} 25 & 12 & 8 \\ 12 & 36 & 6 \\ 8 & 6 & 16 \end{bmatrix}$$

Furthermore, assume $E[H] = \mu_h, E[W] = \mu_w$ and $E[A] = \mu_a$. Denote

$$\mathbf{X} = \begin{bmatrix} H \\ W \\ A \end{bmatrix} \quad \text{and} \quad E[\mathbf{X}] = \begin{bmatrix} \mu_h \\ \mu_w \\ \mu_a \end{bmatrix}.$$

The joint density function of trivariate normal distribution is given in the following vector form

$$f(h, w, a) = f(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}\,|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

Using the above density function, we have complete information about the three random variables **height**, **weight**, and **age** and their relationships. While the mathematical expression may appear complex, we will not work with it directly for computation or derivation, but only for illustration.

4 Asymptotic Sampling Distribution

Following the above technical review of related mathematical and statistical concepts, we now state the fundamental property of MLE. To make this presentation self-contained, we will repeat the basic notations and concepts introduced in earlier sections.

To explain the basic idea, we consider estimating a population with two parameters. Assume an i.i.d random sample $\{x_1, x_2, \cdots, x_n\} \to f(x : \alpha, \beta)$, where α and β are unknown. Let $\hat{\alpha}$ and $\hat{\beta}$ be the MLE of α and β obtained from the maximum likelihood procedure introduced in the previous module.

Let $\mathcal{J}_{\backslash}$ be the **observed Fisher Information Matrix** and

$$\mathbf{\Sigma}(\alpha, \beta) = {\mathcal{I}_{\backslash}}^{-1}(\alpha, \beta) \equiv \begin{bmatrix} \sigma_\alpha^2 & \sigma_{\alpha\beta} \\ \sigma_{\alpha\beta} & \sigma_\beta^2 \end{bmatrix}$$

Using the MLE plug-in principle

$$\widehat{\mathbf{\Sigma}}(\hat{\alpha}, \hat{\beta}) = {\mathcal{I}_{\backslash}}^{-1}(\hat{\alpha}, \hat{\beta}) \equiv \begin{bmatrix} \hat{\sigma}_\alpha^2 & \hat{\sigma}_{\alpha\beta} \\ \hat{\sigma}_{\alpha\beta} & \hat{\sigma}_\beta^2 \end{bmatrix} \approx \mathcal{J}_n^{-1}(\hat{\alpha}, \hat{\beta})$$

be the covariance of $(\hat{\alpha}, \hat{\beta})$.

If the sample size is large, we have the following asymptotic sampling distribution.

$$\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} \to \mathcal{N}_2 \left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \begin{bmatrix} \sigma_\alpha^2 & \sigma_{\alpha\beta} \\ \sigma_{\alpha\beta} & \sigma_\beta^2 \end{bmatrix} \right)$$

This is a special version of bivariate central limit theorem! With the above bivariate normal distribution, we can find the asymptotic sampling distributions of $\hat{\alpha}$ and $\hat{\beta}$ in the following

$$\hat{\alpha} \to N(\alpha, \hat{\sigma}_\alpha^2) \quad \text{and} \quad \hat{\beta} \to N(\beta, \hat{\sigma}_\beta^2).$$

They are the marginal distribution of $(\hat{\alpha}, \hat{\beta})$.

5 A Numerical Example Using R

To conclude this module, we present a numerical example following the same methodology used in Section 2.2.3, where a small toy dataset was employed. Each step will be documented in the accompanying code.

Data Set: Let's use wind speed data (in m/s) from a wind energy application. The Weibull distribution is commonly used to model wind speeds for wind power generation analysis.

12.4 2.7 4.3 4.3 11.1 2.5 5.3 8.7 11.4 4.2 7.5 6.5 12.1 3.2 2.5 8.5 4.2 7.2 16.0 7.0 9.5 6.3 7.9 2.7 10.9 1.1 4.6 10.4 4.2 0.6 4.8 4.8 3.7 15.6 13.3 0.4 12.1 3.2 15.2 1.5 2.5 6.2 5.4 3.4 2.8 0.8 4.0 0.5 5.4 6.5

5.1 Implementation of `optim()`

`optim()` is a powerful but **black-box** optimization routine. Blindly accepting its output can lead to wrong estimates, misleading uncertainty quantification, and incorrect inferences. When use it to find the MLE, please pay attention to the following key information.

Key Cautionary Points

- **Local vs. global optima**: `optim()` finds a local optimum, not necessarily the global MLE. This is especially problematic for multi-modal likelihoods.
- **Convergence codes are not guarantees**: A convergence code of 0 **does not** guarantee you've found the true MLE, only that optim()'s internal stopping conditions were met. **However, if it is not 0, the reported parameter value is not the true MLE.**
- **Sensitivity to initial values**: Different starting points can lead to different results. **Try several set of slightly different initial values to test the resulting parameter values are stable**.
- **Incorrect Hessian approximation**: If you use `hessian = TRUE`, `optim()` returns a finite-difference approximation, which can be inaccurate (especially if parameters are at very different scales).
- **Boundary issues**: `optim()` can converge to values at the boundary of plausible parameter space, which may not satisfy theoretical regularity conditions for MLE asymptotics.
- **Poorly behaved likelihood**: Flat regions, discontinuities, or numerical instability can cause `optim()` to fail silently.

KEY IMPLEMENTATION STEPS

Step 1: Data Loading

```
wspeed <- c(12.4, 2.7, 4.3, 4.3, 11.1, 2.5, 5.3, 8.7, 11.4, 4.2, 7.5, 6.5, 12.1, 3.2,
            2.5, 8.5, 4.2, 7.2, 16.0, 7.0, 9.5, 6.3, 7.9, 2.7, 10.9, 1.1, 4.6, 10.4,
            4.2, 0.6, 4.8, 4.8, 3.7, 15.6, 13.3, 0.4, 12.1, 3.2, 15.2, 1.5, 2.5, 6.2,
            5.4, 3.4, 2.8, 0.8, 4.0, 0.5, 5.4, 6.5)
```

Step 2: Likelihood and Gradient Functions

We translate the log-likelihood function and score (gradient) functions in section 2.2.3 into R code directly in the following.

```
###
# **Negative** log-likelihood function for Weibull distribution
neg_log_likelihood <- function(params, data) {
  k <- params[1]       # shape parameter
  lambda <- params[2]  # scale parameter

  # Ensure parameters are positive
  if (k <= 0 || lambda <= 0) {
    return(1e10)  # Return large value for invalid parameters
  }

  n <- length(data)
  log_lik <- n * log(k) - n * k * log(lambda) +
    (k - 1) * sum(log(data)) - sum((data / lambda)^k)

  return(-log_lik)  # Return negative log-likelihood
}
###
# Gradient of **negative** log-likelihood function
neg_log_likelihood_gradient <- function(params, data) {
  k <- params[1]       # shape parameter
```

```r
  lambda <- params[2]  # scale parameter

  n <- length(data)

  # Partial derivatives
  dk <- -n/k + n * log(lambda) - sum(log(data)) +
        sum(log(data/lambda) * (data/lambda)^k)

  dlambda <- n * k/lambda - k/lambda * sum((data/lambda)^k)

  return(c(-dk, -dlambda))  # Gradient of negative log-likelihood
}
```

Step 3: Calling `optim()`

```r
# Finding Initial Values
## It is critical to select appropriate initial values to
## ensure fast convergence. In general, we can use whatever
## methods (such as MME) that are available to get appropriate
## initial values
# Initial parameter estimates (method of moments)

initial_k <- (sd(wspeed)/mean(wspeed))^(-1.086)
initial_lambda <- mean(wspeed)/gamma(1 + 1/initial_k)

####
# MLE using optim() with gradient
mle_result <- optim(
  #par = c(initial_k,initial_lambda),
  par = c(1,7),
  fn = neg_log_likelihood,
  gr = neg_log_likelihood_gradient,
  data = wspeed,
  #method = "BFGS",         # BFGS can use gradient information
  hessian = TRUE,          # this is the observed Hessian matrix
  control = list(maxit = 1000, trace = 0)
)
##
mle_result
```

```
$par
[1] 1.504525 6.903388

$value
[1] 136.0852

$counts
function gradient
      43       NA

$convergence
[1] 0

$message
NULL
```

```
$hessian
           [,1]      [,2]
[1,] -38.530012  2.964567
[2,]   2.964567 -2.374943
```

CAUTION: Before reporting any estimates, validate the optimization output with the following steps:

- **Check the convergence code**. A code of 0 indicates the algorithm completed its process, but this is only an internal check—it does not confirm the parameters are the true MLE.
- **Test different starting values**. The final parameter estimates should not be identical to the initial guesses. Running the optimization from multiple starting points helps verify that the solution is robust and not an artifact of the initial conditions.
- **Verify the gradient is near zero**. Compute the gradient at the reported solution. A near-zero gradient confirms a critical point has been reached.
- **Inspect the inverse Hessian**. Calculate the inverse of the Hessian matrix. The diagonal elements of this matrix (the estimated variances) must be positive.
- If `optim()` fails to work properly, consider using alternative optimization functions like `optimx()` (**optimx** package), `nlminb()`/`nlm()` (**stats** package), or `DEoptim()` (**DEoptim** package) for global optimization.

Step 4: Checking Score and Observed Fisher Information Matrix

The score equations in the following code.

```
neg_log_likelihood_gradient(mle_result$par, wspeed)
```

```
[1] -0.0071454743  0.0001360629
```

Both score functions are close to zero as expected. The **estimated covariance matrix**, the inverse of the negative Hessian, can be extracted from the `optim()` in the following

```
Hess <- mle_result$hessian
covar<- solve(-Hess)  # solve() finds the inverse of a square matrix
covar
```

```
           [,1]       [,2]
[1,] 0.02871135 0.03583949
[2,] 0.03583949 0.46580012
```

Step 5: Final Sampling Distribution of MLE"

$$\begin{bmatrix} \hat{\alpha} - \alpha \\ \hat{\beta} - \beta \end{bmatrix} \to \mathcal{N}_2 \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.02871135 & 0.03583949 \\ 0.03583949 & 0.46580012 \end{bmatrix} \right)$$

The two marginal distributions are explicitly given below

$$\hat{\alpha} \to N(\alpha, \sigma_\alpha^2 = 0.0287) \quad \text{and} \quad \hat{\beta} \to N(\beta, \sigma_\beta^2 = 0.4658)$$

The asymptotic univariate distributions discussed above will form the basis for subsequent inference procedures in the following modules.

Concluding Remarks: We have examined several optimization functions available in R packages. After performing appropriate checks to ensure the parameter estimates constitute maximum likelihood estimates,

the Hessian matrix should be extracted and inverted to obtain the covariance matrix for subsequent inference. When utilizing functions other than `optim()`, you should consult the corresponding documentation or AI tools to understand the expected inputs and the nature of the outputs. Before applying optimization methods to your analytical tasks, it is helpful to practice with numerical examples from the documentation or those generated by AI tools.