Week 6: Maximum Likelihood Estimation

ResEcon 703: Topics in Advanced Econometrics

Matt Woerman
University of Massachusetts Amherst

Agenda

Last two weeks

- Logit model

Today

- Maximum likelihood overview
- Maximum likelihood estimator
- Maximum likelihood examples
- Properties of the maximum likelihood estimator
- MLE variance estimator
- Model fit and tests
- Numerical optimization
- Maximum likelihood estimation R example

This week's reading

- Maximum likelihood estimation supplement
- Train textbook, chapter 8

Maximum Likelihood Overview

Recap and Looking Ahead

Last three weeks

- Discrete choice framework
- Random utility model
- Logit model

But we still do not know how to estimate the logit model!

Next two weeks

- Maximum likelihood estimation
- Numerical optimization
- Estimating the logit model

Maximum Likelihood Estimation

Maximum likelihood (ML) estimation is one of the most common estimation methods in structural econometrics

- ML is more flexible than OLS regression
  - ML can accommodate nonlinear models
  - OLS is a special case of ML
- ML requires stronger distributional assumptions than OLS
  - When these assumptions hold, the maximum likelihood estimator (MLE) is consistent and efficient
  - But if these assumptions are invalid, the interpretation is less clear

Overview of maximum likelihood estimation

- ML requires distributional assumptions about the data-generating process you observe
- MLE are the parameters that make it most likely to generate the observed data

Maximum Likelihood Intuition

Suppose we have five random draws from a normal distribution, but we do not know which normal distribution, $\mathcal{N}(\mu, \sigma^2)$

$$\boldsymbol{y} = \{48.7, 50.9, 48.8, 50.6, 48.8\}$$

Consider two candidate distributions

$$\mathcal{N}(0, 1) \quad \text{or} \quad \mathcal{N}(50, 1)$$

- What is the likelihood of generating $\boldsymbol{y}$ from $\mathcal{N}(0, 1)$?
  - Practically zero
- What is the likelihood of generating $\boldsymbol{y}$ from $\mathcal{N}(50, 1)$?
  - Much greater!
- Given these data, $\boldsymbol{y}$, $\mu = 50$ has a greater likelihood than $\mu = 0$

This is a simple example of the intuition of maximum likelihood estimation

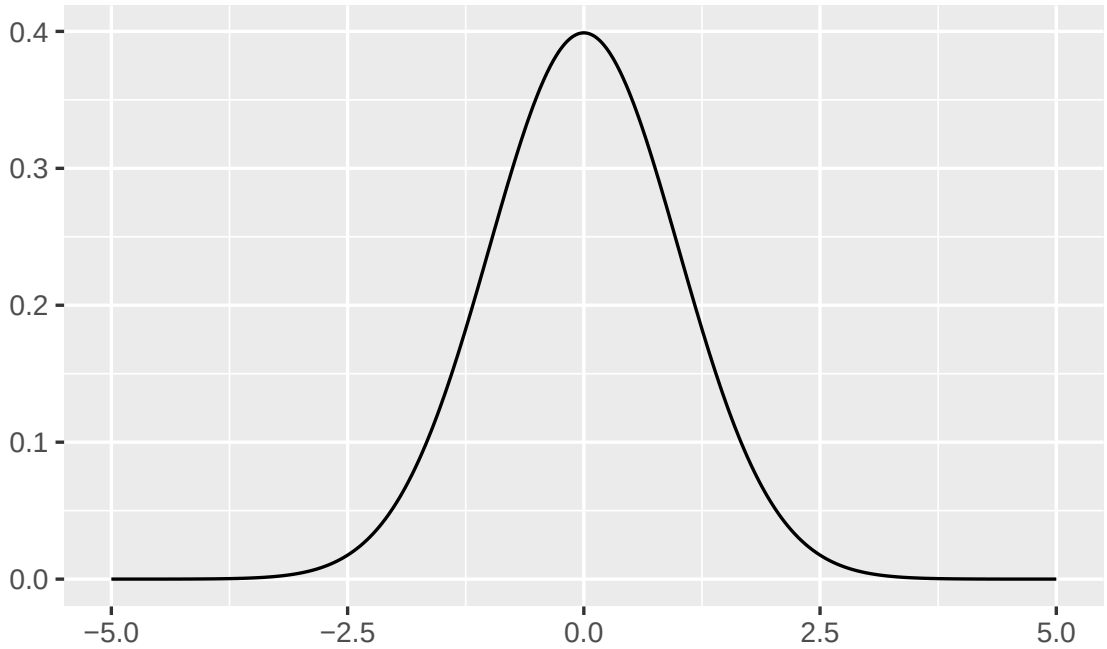
- Find the parameters that maximize the likelihood of generating the data you observe

Probability Density Function

The probability density function, $f(y \mid \boldsymbol{\theta})$, gives us the relative likelihood that a random variable would take a particular value

The probability density function of the normal distribution is

$$f(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{2\sigma^2}}$$



Likelihood

But what if the opposite is true—we know the outcome of the random variable draw, but we do not know the parameters that generated it?

- We can use the same mathematical expression to give us the likelihood that a particular set of parameter values would have generated that sample
- We call this the likelihood function and denote it as $L(\boldsymbol{\theta} \mid \boldsymbol{y})$

The likelihood function for a single known random draw, y, from the normal distribution is

$$L(\mu, \sigma^2 \mid y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{2\sigma^2}}$$

Maximum Likelihood Estimator

Maximum Likelihood Estimation Assumption

The probability density function for a random variable, y, conditioned on a set of parameters, $\boldsymbol{\theta}$, is

$$f(y \mid \boldsymbol{\theta})$$

- This function identifies the data-generating process that underlies an observed sample of data and provides a mathematical description of the data that the process will produce
- We are making an assumption about the density of y, not just its expectation and variance

We could generalize to a random vector, $\boldsymbol{y}$, with joint density $f(\boldsymbol{y} \mid \boldsymbol{\theta})$

- But the random variable assumption will be sufficient for this course

Likelihood Function

The joint density of n independent and identically distributed (i.i.d.) random variables, each with density $f(y \mid \boldsymbol{\theta})$, is

$$f(y_1, \ldots, y_n \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\theta})$$

This representation suggests that the parameters are known and the data are unknown, but usually the opposite is true

- We have data and want to know the parameters of the data-generating process

We simply switch the conditioning and define the likelihood function as a function of the unknown parameters, $\boldsymbol{\theta}$, conditioned on the data we observe, $\boldsymbol{y}$

$$L(\boldsymbol{\theta} \mid \boldsymbol{y}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\theta})$$

Log-Likelihood Function

The likelihood function of unknown parameters $\boldsymbol{\theta}$ conditioned on the data $\boldsymbol{y}$ is

$$L(\boldsymbol{\theta} \mid \boldsymbol{y}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{\theta})$$

It is usually easier to work with the log of this likelihood function, or the log-likelihood function, so we have a sum instead of a product on the right-hand side

$$\ln L(\boldsymbol{\theta} \mid \boldsymbol{y}) = \sum_{i=1}^{n} \ln f(y_i \mid \boldsymbol{\theta})$$

Log is a monotonic transformation, so finding the greatest log-likelihood will get us the same result as finding the greatest likelihood

Maximum Likelihood Estimator

The maximum likelihood estimator, $\widehat{\boldsymbol{\theta}}$, is the set of parameters that maximizes the likelihood function and log-likelihood function

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, L(\boldsymbol{\theta} \mid \boldsymbol{y})$$

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \ln L(\boldsymbol{\theta} \mid \boldsymbol{y})$$

A necessary condition for maximizing $\ln L(\boldsymbol{\theta} \mid \boldsymbol{y})$ is

$$\frac{\partial \ln L(\boldsymbol{\theta} \mid \boldsymbol{y})}{\partial \boldsymbol{\theta}} = \mathbf{0}$$

The maximum likelihood estimator gives the parameter values that maximize the likelihood of having generated the data that we observe

Conditional Likelihood

So far, we have assumed our data, $\boldsymbol{y}$, are conditional on only parameters

- But we usually model our outcome data, $\boldsymbol{y}$, as a function of both parameters, $\boldsymbol{\theta}$, and other data, $\boldsymbol{X}$

When y is also a function of $\boldsymbol{x}$, we need to define its conditional probability density function, $f(y \mid \boldsymbol{x}, \boldsymbol{\theta})$

In almost all cases, we can simply use $f(y \mid \boldsymbol{x}, \boldsymbol{\theta})$ in place of $f(y \mid \boldsymbol{\theta})$ in the definition of the likelihood function and log-likelihood function

$$L(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{X}) = \prod_{i=1}^{n} f(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta})$$

$$\ln L(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{X}) = \sum_{i=1}^{n} \ln f(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta})$$

- This is technically a conditional likelihood function, but we often drop the "conditional" for convenience

Maximum Likelihood Examples

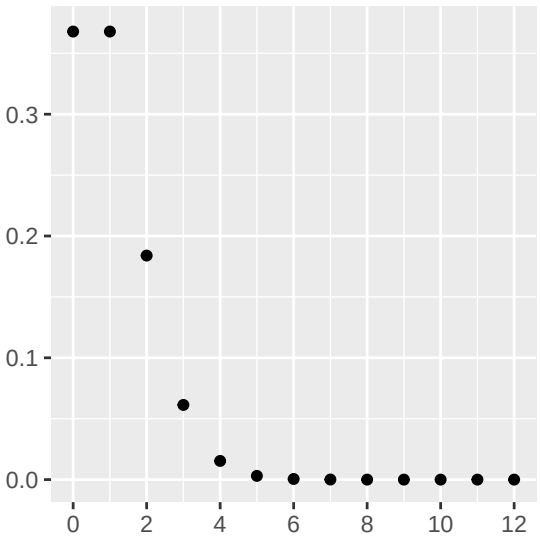
Maximum Likelihood Poisson Example

We have ten data points from a Poisson distribution, but what is the λ parameter of the distribution?

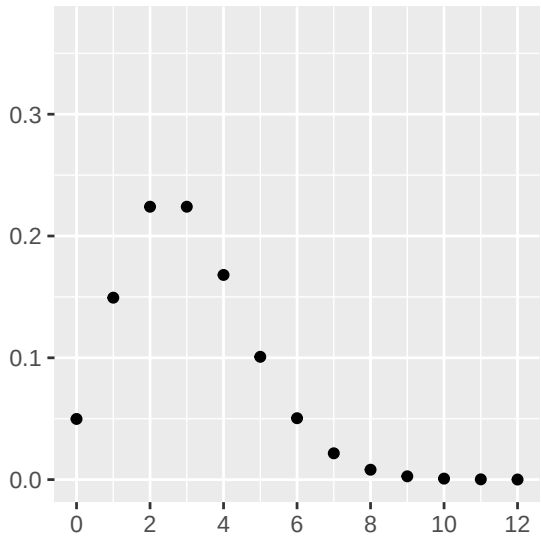
$$\boldsymbol{y} = \{2, 0, 1, 2, 2, 2, 0, 2, 1, 1\}$$

$$f(y \mid \lambda) = \frac{e^{-\lambda}\lambda^{y}}{y!}$$

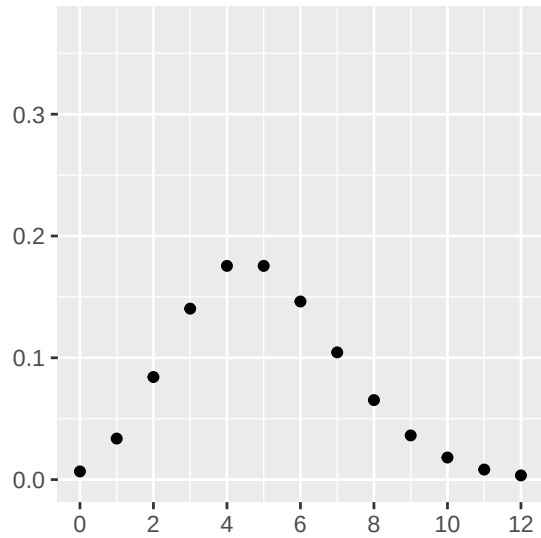
$f(y \mid \lambda = 1)$



$f(y \mid \lambda = 3)$



$f(y \mid \lambda = 5)$



Maximum Likelihood Poisson Example

We have ten data points from a Poisson distribution, but what is the λ parameter of the distribution?

$$\boldsymbol{y} = \{2, 0, 1, 2, 2, 2, 0, 2, 1, 1\}$$

$$L(\lambda \mid \boldsymbol{y}) = \prod_{i=1}^{n} \frac{e^{-\lambda} \lambda^{y_i}}{y_i!} = \frac{e^{-n\lambda} \lambda^{\sum_{i=1}^{n} y_i}}{\prod_{i=1}^{n} y_i!}$$

$$\ln L(\lambda \mid \boldsymbol{y}) = -n\lambda + \ln \lambda \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \ln(y_i!)$$

$$\frac{\partial \ln L(\lambda \mid \boldsymbol{y})}{\partial \lambda} = -n + \frac{1}{\lambda} \sum_{i=1}^{n} y_i$$

$$\frac{\partial \ln L(\lambda \mid \boldsymbol{y})}{\partial \lambda} = 0 \quad \Rightarrow \quad \widehat{\lambda} = \frac{1}{n} \sum_{i=1}^{n} y_i = 1.3$$

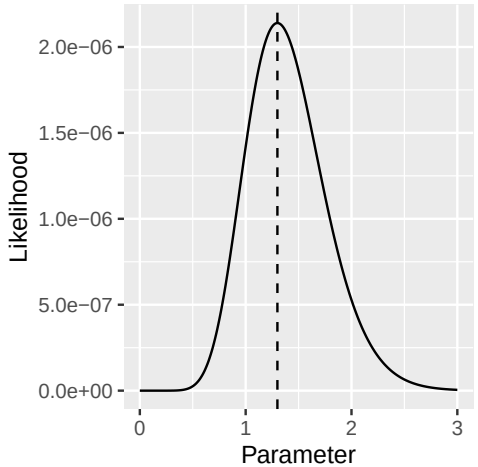
Maximum Likelihood Poisson Example

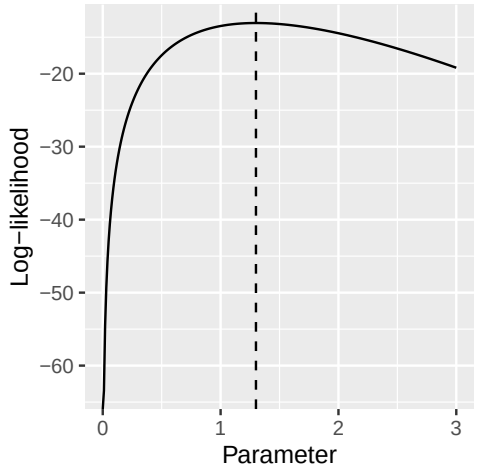
We have ten data points from a Poisson distribution, but what is the λ parameter of the distribution?

$$\boldsymbol{y} = \{2, 0, 1, 2, 2, 2, 0, 2, 1, 1\}$$

$$L(\lambda \mid \boldsymbol{y}) = \prod_{i=1}^{n} \frac{e^{-\lambda}\lambda^{y_i}}{y_i!} = \frac{e^{-n\lambda}\lambda^{\sum_{i=1}^{n} y_i}}{\prod_{i=1}^{n} y_i!} = \frac{e^{-10\lambda}\lambda^{\sum_{i=1}^{n} 13}}{32}$$

$$\ln L(\lambda \mid \boldsymbol{y}) = -n\lambda + \ln \lambda \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \ln(y_i!) = -10\lambda + 13 \ln \lambda - 3.47$$





Maximum Likelihood Normal Example

We have five data points from a normal distribution, but what are the μ and σ^2 parameters of the distribution?

$$\boldsymbol{y} = \{6.08, 5.29, 2.52, 2.94, 5.36\}$$

$$f(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{2\sigma^2}}$$

$$L(\mu, \sigma^2 \mid \boldsymbol{y}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2}$$

$$\ln L(\mu, \sigma^2 \mid \boldsymbol{y}) = -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln \sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2$$

$$\frac{\partial \ln L(\mu, \sigma^2 \mid \boldsymbol{y})}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - \mu)$$

$$\frac{\partial \ln L(\mu, \sigma^2 \mid \boldsymbol{y})}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{n}(y_i - \mu)^2$$

Maximum Likelihood Normal Example

We have five data points from a normal distribution, but what are the μ and σ^2 parameters of the distribution?

$$\boldsymbol{y} = \{6.08, 5.29, 2.52, 2.94, 5.36\}$$

$$\frac{\partial \ln L(\mu, \sigma^2 \mid \boldsymbol{y})}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \mu)$$

$$\frac{\partial \ln L(\mu, \sigma^2 \mid \boldsymbol{y})}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^{n} (y_i - \mu)^2$$

$$\frac{\partial \ln L(\mu, \sigma^2 \mid \boldsymbol{y})}{\partial \mu} = 0 \quad \Rightarrow \quad \widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} y_i = 4.44$$

$$\frac{\partial \ln L(\mu, \sigma^2 \mid \boldsymbol{y})}{\partial \sigma^2} = 0 \quad \Rightarrow \quad \widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{\mu})^2 = 2.04$$

Maximum Likelihood OLS Regression Example

In the previous two examples, we have estimated the parameters that maximize the likelihood of generating the data that we observe

But in most econometric applications, we really want to estimate the parameters that maximize the likelihood of generating our outcome data (or dependent variable) conditional on the other data (or independent variables) that we observe

- A basic example is a simple OLS regression

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

If we make a distributional assumption about ε_i, then we can estimate the parameters of this model using maximum likelihood

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Maximum Likelihood OLS Regression Example

Combining our simple OLS regression equation

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

with the distributional assumption about the error term

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

gives us a conditional distribution of y_i

$$y_i \mid x_i \sim \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma^2)$$

Then the conditional probability density function of y_i is

$$f(y_i \mid x_i, \beta_0, \beta_1, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_i - \beta_0 - \beta_1 x_i)^2}{2\sigma^2}}$$

Maximum Likelihood OLS Regression Example

The (conditional) log-likelihood function is

$$\ln L(\beta_0, \beta_1, \sigma^2 \mid \boldsymbol{y}, \boldsymbol{x}) = -\frac{n}{2} \ln 2\pi - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Taking the derivative with respect to each parameter—β_0, β_1, and σ^2—and setting them equal to zero yields the OLS estimators

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 \bar{x}$$

$$\widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_i)^2$$

Properties of the Maximum Likelihood Estimator

Asymptotic Properties of MLE

Under certain regularity conditions, the maximum likelihood estimator (MLE) has these properties

1. Consistency
2. Asymptotic normality
3. Asymptotic efficiency
4. Invariance

Regularity Conditions for MLE

The following properties are true only if these regularity conditions are met

1. The first three derivatives of $\ln f(y_i \mid \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ are continuous and finite for almost all y_i and for all $\boldsymbol{\theta}$
2. The conditions necessary to obtain the expectations of the first and second derivatives of $\ln f(y_i \mid \boldsymbol{\theta})$ are met
3. For all values of $\boldsymbol{\theta}$, $\left| \frac{\partial^3 \ln f(y_i|\boldsymbol{\theta})}{\partial\theta_j \partial\theta_k \partial\theta_l} \right|$ is less than a function that has a finite expectation

Consistency of MLE

The MLE, $\widehat{\boldsymbol{\theta}}$, converges in probability to the true parameter value(s), $\boldsymbol{\theta}_0$

$$\widehat{\boldsymbol{\theta}} \xrightarrow{p} \boldsymbol{\theta}_0$$

- As our sample size grows (to infinity), the MLE becomes vanishingly close to the true parameter value(s)

Asymptotic Normality of MLE

The asymptotic distribution of the MLE, $\widehat{\boldsymbol{\theta}}$, is normal with mean at the true parameter value(s), $\boldsymbol{\theta}_0$, and known variance

$$\widehat{\boldsymbol{\theta}} \overset{a}{\sim} \mathcal{N}\left(\boldsymbol{\theta}_0, I(\boldsymbol{\theta}_0)^{-1}\right)$$

where

$$I(\boldsymbol{\theta}_0) = -E_0\left[\frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0 \partial \boldsymbol{\theta}_0'}\right]$$

- The asymptotic variance-covariance matrix of the MLE is

$$Var(\widehat{\boldsymbol{\theta}}) = \left\{-E_0\left[\frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0 \partial \boldsymbol{\theta}_0'}\right]\right\}^{-1}$$

- We are more certain of the MLE when the likelihood function has more curvature

Asymptotic Efficiency of MLE

The MLE, $\widehat{\boldsymbol{\theta}}$, is asymptotically efficient and achieves the Cramér-Rao lower bound

$$Var(\widehat{\boldsymbol{\theta}}) = I(\boldsymbol{\theta}_0)^{-1}$$

- No consistent estimator has lower asymptotic variance than the MLE

Invariance of MLE

The MLE of $\gamma_0 = \boldsymbol{c}(\boldsymbol{\theta}_0)$ is $\boldsymbol{c}(\widehat{\boldsymbol{\theta}})$ if $\boldsymbol{c}(\boldsymbol{\theta}_0)$ is continuous and continuously differentiable, where $\widehat{\boldsymbol{\theta}}$ is the MLE of true parameter(s) $\boldsymbol{\theta}_0$

- The MLE of a function of some parameter(s) is the function applied to the MLE of the parameter(s)

MLE Variance Estimator

Variance of the Maximum Likelihood Estimator

From the asymptotic normality of MLE, the variance-covariance matrix of the MLE is

$$Var(\widehat{\boldsymbol{\theta}}) = \left\{ -E_0 \left[\frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0 \partial \boldsymbol{\theta}_0'} \right] \right\}^{-1}$$

- The inner-most term (inside the [ ]) is the Hessian of the log-likelihood function with respect to the parameters
- The term that is inverted (the $-E_0[\,]$ term) is equivalent to the Fisher information matrix
- The variance of the MLE is evaluated at $\boldsymbol{\theta}_0$, the true parameter value(s), and requires taking an expectation

Hessian of the Log-Likelihood Function

The Hessian of the log-likelihood function with respect to the parameters is the square matrix that contains the second derivative of the log-likelihood function with respect to all pairwise combinations of parameters

$$\frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0 \partial \boldsymbol{\theta}_0'} = \begin{pmatrix} \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial^2 \theta_1} & \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \theta_1 \partial \theta_k} \\ \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \theta_1 \partial \theta_2} & \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial^2 \theta_2} & \cdots & \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \theta_2 \partial \theta_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \theta_1 \partial \theta_k} & \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \theta_2 \partial \theta_k} & \cdots & \frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial^2 \theta_k} \end{pmatrix}$$

This matrix describes the local curvature of the log-likelihood function around the true parameter values, $\boldsymbol{\theta}_0$

Information Matrix Equality

The Fisher information matrix measures the amount of information that our data, $\boldsymbol{y}$ and $\boldsymbol{X}$, contains about the unknown parameters, $\boldsymbol{\theta}$

$$I(\boldsymbol{\theta}_0) = E_0\left[\frac{\partial \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}\frac{\partial \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0'}\right]$$

The information matrix equality gives that the Fisher information matrix equals the negative of the expectation of the Hessian of the log-likelihood function

$$E_0\left[\frac{\partial \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}\frac{\partial \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0'}\right] = -E_0\left[\frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0 \partial \boldsymbol{\theta}_0'}\right]$$

MLE Variance Estimator

The true variance-covariance matrix of the MLE is evaluated at the true parameter values, $\boldsymbol{\theta}$, and requires taking an expectation

$$Var(\widehat{\boldsymbol{\theta}}) = \left\{ -E_0 \left[\frac{\partial^2 \ln L(\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0 \partial \boldsymbol{\theta}_0'} \right] \right\}^{-1}$$

We can estimate this variance by evaluating the actual Hessian (not its expectation) at the MLE, $\widehat{\boldsymbol{\theta}}$

The estimator of the MLE variance-covariance matrix is

$$\widehat{Var}(\widehat{\boldsymbol{\theta}}) = \left\{ -\frac{\partial^2 \ln L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \bigg|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}} \right\}^{-1}$$

More robust variance estimators exist, but we will use this most basic one

Model Fit and Tests

Likelihood Ratio Index

One measure of how well a MLE fits the data is the likelihood ratio index

$$\rho = 1 - \frac{\ln L(\widehat{\boldsymbol{\theta}})}{\ln L(\mathbf{0})}$$

where $L(\mathbf{0})$ measures the fit of a model with only a constant term (all other parameters equal to 0)

- This index looks like R^2 and is sometimes called a "pseudo R^2"
- But this name is misleading because this metric is nothing like R^2 other than having the same range of $[0, 1]$

Larger values of ρ imply a better model fit, but this is no different from saying larger values of the likelihood and log-likelihood functions are better

Hypothesis Tests

Suppose we want to test hypotheses about the parameters of our model

$$H_0 : \boldsymbol{h}(\boldsymbol{\theta}_0) = \mathbf{0}$$

where $\boldsymbol{h}(\boldsymbol{\theta}_0)$ is any set of J parameter restrictions

This specification of hypotheses is fully general

- We can test if parameters are equal to zero

$$\boldsymbol{h}(\boldsymbol{\theta}_0) = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

- We can test if parameters are equal to each other

$$\boldsymbol{h}(\boldsymbol{\theta}_0) = \begin{pmatrix} \theta_1 - \theta_3 \\ \theta_2 - \theta_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Likelihood Ratio Test

The likelihood ratio test compares the log-likelihood values of the unrestricted model and the restricted model

- If the hypotheses are true, then these values should be close

The likelihood ratio test statistic is

$$-2 \ln \lambda = 2 \left(\ln L(\widehat{\boldsymbol{\theta}}_U) - \ln L(\widehat{\boldsymbol{\theta}}_R) \right)$$

- $\widehat{\boldsymbol{\theta}}_U$ is the MLE of the unrestricted model
- $\widehat{\boldsymbol{\theta}}_R$ is the MLE of the restricted model
- λ is the likelihood ratio, $\lambda = L(\widehat{\boldsymbol{\theta}}_R)/L(\widehat{\boldsymbol{\theta}}_U)$

This test statistic is distributed χ^2 with degrees of freedom equal to the number of model restrictions

$$-2 \ln \lambda \sim \chi^2(J)$$

Wald and Lagrange Multiplier Tests

Two other common tests for MLE parameters are the Wald test and the Lagrange multiplier test

Wald test

- If the hypotheses are true, then $\boldsymbol{h}(\widehat{\boldsymbol{\theta}}_U) \approx \mathbf{0}$
- Test if $\boldsymbol{h}(\widehat{\boldsymbol{\theta}}_U)$ is sufficiently close to $\mathbf{0}$

Lagrange multiplier test

- If the hypotheses are true, then $\partial \ln L(\widehat{\boldsymbol{\theta}}_R)/\partial \boldsymbol{\theta} \approx \mathbf{0}$
- Test if $\partial \ln L(\widehat{\boldsymbol{\theta}}_R)/\partial \boldsymbol{\theta}$ is sufficiently close to $\mathbf{0}$

The Wald and Lagrange multiplier test statistics tend to be more complicated than the likelihood ratio test statistic

- We will use the likelihood ratio test with MLE in this course

Numerical Optimization

Numerical Optimization

Most structural estimation requires maximizing or minimizing an objective function

- For ML, we want to maximize the log-likelihood function

In theory, this is a relatively simple proposition

- Some optimization problems have a closed-form expression
- For only one or two parameters, a grid search may suffice

In practice, finding the correct parameters in an efficient way can be challenging

- Especially when you are optimizing over a vector of many parameters and using a complex objective function
- Numerical optimization algorithms can solve this problem

Numerical Optimization Steps

We want to find the set of K parameters, $\widehat{\boldsymbol{\theta}}$, that maximize the objective function, $\ell(\boldsymbol{\theta})$

1. Begin with some initial parameter values, $\boldsymbol{\theta}^0$
2. Check if you can "walk up" to a higher value
3. If so, take a step in the right direction to $\boldsymbol{\theta}^1$
4. Repeat steps (2) and (3), stepping from $\boldsymbol{\theta}^s$ to $\boldsymbol{\theta}^{s+1}$ until you reach the maximum

But which direction should you step and how big of a step should you take from $\boldsymbol{\theta}^s$ to $\boldsymbol{\theta}^{s+1}$?

- If your steps are too small, optimization can take too long
- If your steps are too big, you may never converge to a solution

Gradient and Hessian

The gradient tells us which direction to step

$$\boldsymbol{g}^s = \left.\frac{\partial \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^s}$$

- The gradient is a vector of K elements that tells us which direction to move each parameter to increase the objective function

The Hessian tells us how far to step

$$\boldsymbol{H}^s = \left.\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^s}$$

- The Hessian is a $K \times K$ matrix that gives us information about the curvature of the objective function in all dimensions

Numerical Optimization Algorithms

There are many numerical optimization algorithms that use the gradient and Hessian (and sometimes other statistics or constraints) in different ways to maximize the objective function

- Newton-Raphson
- BHHH (Berndt-Hall-Hall-Hausman)
- BHHH-2
- Steepest ascent
- DFP (Davidson-Fletcher-Powell)
- BFGS (Broyden-Fletcher-Goldfarb-Shanno)
- Nelder-Mead
- Conjugate gradients
- Limited-memory BFGS
- Simulated annealing

Convergence Criterion

How do we know the model has converged and we can stop taking steps?

- In theory, the gradient vector equals zero
- In practice, you will never hit the precise vector of parameters (down to the 15th decimal point) that yields a gradient of zero
- So we stop taking steps when we get "close enough"

How do we know when we are "close enough?"

- Calculate a statistic, m^s, at every optimization step

$$m^s = (\boldsymbol{g}^s)'(-\boldsymbol{H}^s)^{-1}(\boldsymbol{g}^s)$$

- Stop iterating when this statistic is less than a predetermined tolerance level, $\breve{m}$

$$m^s < \breve{m}$$

Global or Local Maximum

Global maximum

- The largest value of the objective function over all possible sets of parameter values
- This is the maximum you want to converge to
- When the objective function is globally concave (as in the logit model with linear utility), you will always hit the global maximum

Local maximum

- The largest value of the objective function within a range of parameter values, but not the global maximum
- Optimization algorithms will sometimes converge to a local maximum instead of the global maximum
- More complex objective functions have local maxima

Try different starting values and algorithms to ensure you have converged to the global maximum, not a local maximum

Maximum Likelihood Estimation R Example

Maximum Likelihood Example of OLS Regression

Using the `mtcars` dataset, regress `mpg` on `hp`

$$\texttt{mpg}_i = \beta_0 + \beta_1 \texttt{hp}_i + \varepsilon_i$$

Instead of using the "canned" `lm()` function or a "hand-coded" OLS estimator—both of which we did in week 2—we will estimate the parameters of this model using

- Maximum likelihood estimation
- Numerical optimization

Reminder: OLS is a special case of maximum likelihood, so we should estimate the same parameters as in week 2, but in a very different way

Look at the mtcars Dataset

You should always double-check the structure of your dataset

```
## Load tidyverse
library(tidyverse)
## Look at the mtcars data
tibble(mtcars)
## # A tibble: 32 x 11
##      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  21       6  160    110  3.9   2.62  16.5     0     1     4     4
##  2  21       6  160    110  3.9   2.88  17.0     0     1     4     4
##  3  22.8     4  108     93  3.85  2.32  18.6     1     1     4     1
##  4  21.4     6  258    110  3.08  3.22  19.4     1     0     3     1
##  5  18.7     8  360    175  3.15  3.44  17.0     0     0     3     2
##  6  18.1     6  225    105  2.76  3.46  20.2     1     0     3     1
##  7  14.3     8  360    245  3.21  3.57  15.8     0     0     3     4
##  8  24.4     4  147.    62  3.69  3.19  20       1     0     4     2
##  9  22.8     4  141.    95  3.92  3.15  22.9     1     0     4     2
## 10  19.2     6  168.   123  3.92  3.44  18.3     1     0     4     4
## # ... with 22 more rows
```

Summarize the mtcars Dataset

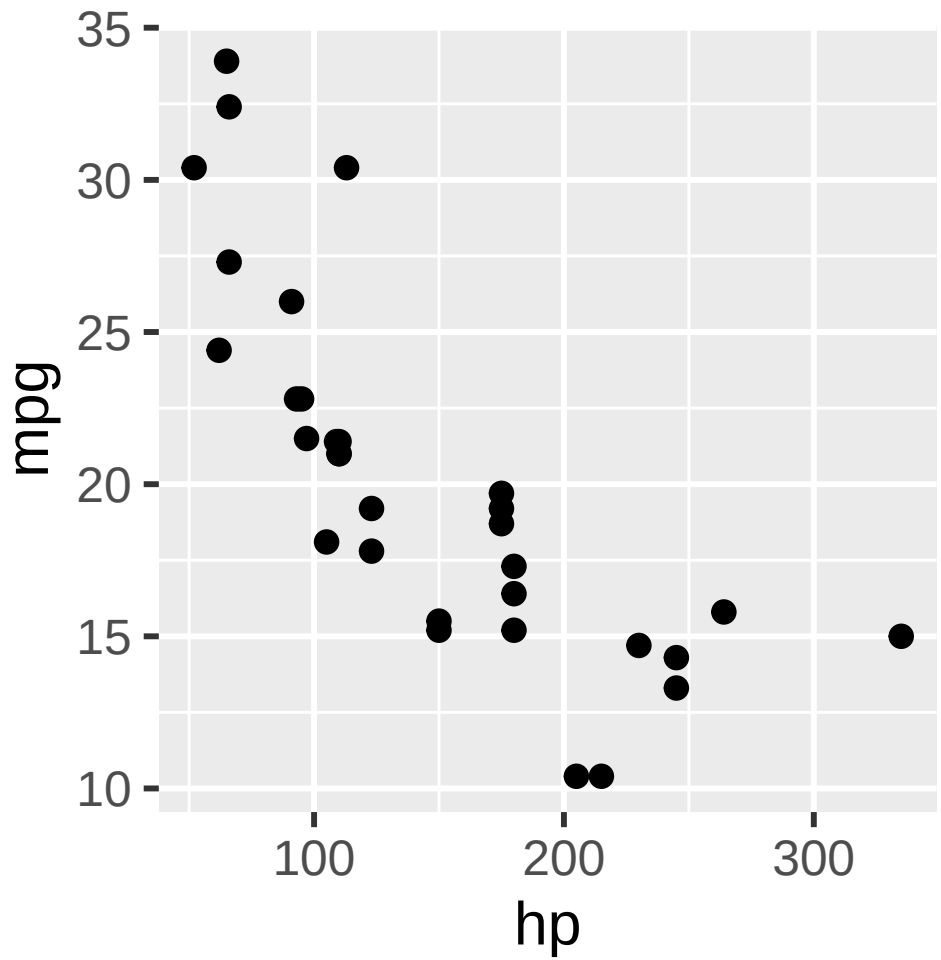
It can be helpful to generate basic summary statistics for your dataset to get a sense for the scale and variation of each variable

```
## Summarize the mtcars dataset
mtcars %>%
  select(mpg, disp, hp, wt, qsec) %>%
  summary()
##       mpg             disp             hp              wt
##  Min.   :10.40   Min.   : 71.1   Min.   : 52.0   Min.   :1.513
##  1st Qu.:15.43   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:2.581
##  Median :19.20   Median :196.3   Median :123.0   Median :3.325
##  Mean   :20.09   Mean   :230.7   Mean   :146.7   Mean   :3.217
##  3rd Qu.:22.80   3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.610
##  Max.   :33.90   Max.   :472.0   Max.   :335.0   Max.   :5.424
##       qsec
##  Min.   :14.50
##  1st Qu.:16.89
##  Median :17.71
##  Mean   :17.85
##  3rd Qu.:18.90
##  Max.   :22.90
```

Plot the `mtcars` Dataset

Plotting the data can give an idea of what to expect from your regression

```
## Plot the mtcars dataset
ggplot(data = mtcars, mapping = aes(x = hp, y = mpg)) +
  geom_point()
```



Maximum Likelihood Estimation of OLS Regression

$$\texttt{mpg}_i = \beta_0 + \beta_1 \texttt{hp}_i + \varepsilon_i$$

How do we estimate the parameters of this model using ML?

For the general regression equation

$$y_i = \boldsymbol{\beta}' \boldsymbol{x_i} + \varepsilon_i$$

with a distributional assumption about the error term

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

we have a conditional distribution of y_i

$$y_i \mid \boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\beta}' \boldsymbol{x_i}, \sigma^2)$$

Log-Likelihood Function of OLS Regression

If each y_i has a conditional distribution of

$$y_i \mid \boldsymbol{x}_i \sim \mathcal{N}(\beta' \boldsymbol{x}_i, \sigma^2)$$

then the (conditional) log-likelihood function is

$$\ln L(\beta, \sigma^2 \mid \boldsymbol{y}, \boldsymbol{X}) = \sum_{i=1}^{n} \ln f(y_i \mid \boldsymbol{x}_i, \beta, \sigma^2)$$

For our example, we have three parameters to estimate

$$\boldsymbol{\theta} = \left(\beta_0, \beta_1, \sigma^2\right)$$

We could take the derivative of $\ln L(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{X})$ with respect to each parameter and solve the first-order conditions

- Or we could maximize $\ln L(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{X})$ by numerical optimization!

Numerical Optimization for MLE

We want to find the set of K parameters, $\widehat{\boldsymbol{\theta}}$, that maximize the log-likelihood function, $\ln L(\boldsymbol{\theta})$

1. Begin with some initial parameter values, $\boldsymbol{\theta}^0$
2. Check if you can "walk up" to a higher value
3. If so, take a step in the right direction to $\boldsymbol{\theta}^1$
4. Repeat steps (2) and (3), stepping from $\boldsymbol{\theta}^s$ to $\boldsymbol{\theta}^{s+1}$ until you reach the maximum

The `optim()` function in R will perform this numerical optimization for us

- We just have to give the `optim()` function two things:
  - Some initial parameter values, $\boldsymbol{\theta}^0$
  - A function that will take those parameters as an argument and calculate the log-likelihood, $\ln L(\boldsymbol{\theta})$
  - (And sometimes additional information to fine-tune the optimization procedure and output)

Optimization in R

```
## Help file for the optimization function, optim
?optim
## Arguments for optim function
optim(par, fn, gr, ..., method, lower, upper, control, hessian)
```

`optim()` requires that you create a function, `fn`, that

1. Takes a set of parameters and other arguments as inputs
2. Calculates your objective function given those parameters
3. Returns this value of the objective function

You also have to give `optim()` arguments for

- `par`: starting parameter values
- `...`: dataset and other things needed by your function
- `method`: optimization algorithm
  - I recommend `method = 'BFGS'` for our estimation

`optim()` will find the parameters that minimize the objective function

- To maximize, minimize the negative of the objective function

Steps to Calculate the OLS Log-Likelihood

The OLS log-likelihood function is

$$\ln L(\beta, \sigma^2 \mid \boldsymbol{y}, \boldsymbol{X}) = \sum_{i=1}^{n} \ln f(y_i \mid \boldsymbol{x_i}, \beta, \sigma^2)$$

where the conditional distribution of each y_i is

$$y_i \mid \boldsymbol{x}_i \sim \mathcal{N}(\beta' \boldsymbol{x_i}, \sigma^2)$$

Steps to calculate the OLS log-likelihood conditional on $\boldsymbol{\theta}$

1. Construct matrices $\boldsymbol{X}$ and $\boldsymbol{y}$
2. Calculate fitted values of $\boldsymbol{y}$, $\widehat{\boldsymbol{y}} = \beta' \boldsymbol{x}_i$, which is the mean of each y_i
3. Calculate the density for each y_i, $f(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta})$
4. Calculate the log-likelihood, $\ln L(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{X}) = \sum_{i=1}^{n} \ln f(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta})$

Function to Calculate OLS Log-likelihood

```r
## Create function to calculate OLS log-likelihood
ll_ols <- function(params, data, y_var, x_vars) {
  ## Add column of ones for the constant term
  reg_data <- data %>%
    mutate(constant = 1)
  ## Select data for X and convert to a matrix
  X <- reg_data %>%
    select(all_of(c('constant', x_vars))) %>%
    as.matrix()
  ## Select data for y and convert to a matrix
  y <- reg_data %>%
    select(all_of(y_var)) %>%
    as.matrix()
  ## Select coefficient parameters
  beta_hat <- params[-length(params)]
  ## Select error variance parameters
  sigma2_hat <- params[length(params)]
  ## Calculate fitted y values
  y_hat <- X %*% beta_hat
  ## Calculate the pdf values of each outcome
  y_pdf <- dnorm(y, mean = y_hat, sd = sqrt(sigma2_hat))
  ## Calculate the log-likelihood
  ll <- sum(log(y_pdf))
  ## Return the negative of log-likelihood for minimization
  return(-ll)
}
```

`optim()` will minimize our objective function

- We will have `optim()` minimize $-\ln L(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{X})$

Maximize OLS Log-Likelihood

```
## Maximize the OLS log-likelihood function
mle_ols_1 <- optim(par = c(0, 0, 1), fn = ll_ols,
                   data = mtcars, y_var = 'mpg', x_vars = 'hp',
                   method = 'BFGS', hessian = TRUE)
```

Optimization Results

```
## Show optimization results
mle_ols_1
## $par
## [1] 30.09908613 -0.06822967 13.99015277
##
## $value
## [1] 87.61931
##
## $counts
## function gradient
##       84       28
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##                [,1]          [,2]           [,3]
## [1,]  2.287323e+00 3.355217e+02 -3.520739e-06
## [2,]  3.355217e+02 5.963323e+04  5.199112e-04
## [3,] -3.520739e-06 5.199112e-04  8.174375e-02
```

Maximum Likelihood Estimator and Standard Errors

```
## Show parameter estimates
mle_ols_1$par
## [1] 30.09908613 -0.06822967 13.99015277

## Calculate MLE standard errors
mle_ols_1$hessian %>%
  solve() %>%
  diag() %>%
  sqrt()
## [1] 1.58205585 0.00979809 3.49762080
```

MLE of Another OLS Regression

Now use the same optimization function for a different regression

- Regress `mpg` on `hp`, `disp`, `wt`, `qsec` from the `mtcars` dataset

```
## Maximize the OLS log-=likelihood function
mle_ols_2 <- optim(par = c(rep(0, 5), 1), fn = ll_ols,
                   data = mtcars, y_var = 'mpg',
                   x_vars = c('hp', 'disp', 'wt', 'qsec'),
                   method = 'BFGS', hessian = TRUE)
## Show parameter estimates
mle_ols_2$par
## [1] 29.171504479 -0.021155823  0.002340875 -4.508394892  0.447863784
## [6]  5.901025047

## Calculate MLE standard errors
mle_ols_2$hessian %>%
  solve() %>%
  diag() %>%
  sqrt()
## [1] 8.017208039 0.014478340 0.009948106 1.173009722 0.432869462
## [6] 1.500889360
```

MLE of Another OLS Regression

Try a different dataset in our optimization function

- Regress `Petal.Length` on `Petal.Width`, `Sepal.Length`, and `Sepal.Width` from the `iris` dataset

```
## Maximize the OLS log-=likelihood function
mle_ols_3 <- optim(par = c(rep(0, 4), 1), fn = ll_ols,
                   data = iris, y_var = 'Petal.Length',
                   x_vars = c('Petal.Width', 'Sepal.Length',
                              'Sepal.Width'),
                   method = 'BFGS', hessian = TRUE)
## Show parameter estimates
mle_ols_3$par
## [1] -0.26270817  1.44679345  0.72913805 -0.64601245  0.09902641

## Calculate MLE standard errors
mle_ols_3$hessian %>%
  solve() %>%
  diag() %>%
  sqrt()
## [1] 0.29342388 0.06670595 0.05753860 0.06758029 0.01143187
```