CSSS/POLS 510 - Maximum Likelihood Estimation

Lab 2: Probability Distributions, Statistical Inference, and Ordinay Least Squares

Minji Jeong

Preview

- Statistical Inference
- Probability Distributions
- Least Squares Estimation

Statistical inference: estimation

- In statistical inference, we are concerned with making **predictions** (inferences) about a **DGP** or *population* based on information obtained from a *sample*.
- This involves the following key concepts:
  - **Estimand**: The **quantity of interest** from the data-generating process that we aim to estimate or infer.
  - **Estimator**: A statistical **method** or **formula** used to estimate the estimand based on sample data.
  - **Estimate**: it is the calculated value that serves as the **best guess** or approximation of the estimand based on the available information from the sample.

Estimand, estimator, and esitmate

- Statistical inference involves using **estimators** to obtain **estimates** of **estimands** from sample data to make predictions about the population.

- Analogy (from Spain): have you ever heard about the *ecce homo*?
    - explanation: 1 min. YouTube video.

Statistical inference: estimation

Estimand

Statistical inference: estimation

Estimand

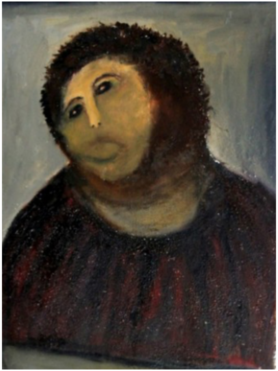
Estimator

Statistical inference: estimation

Estimand

Estimator

Estimate



Statistical inference: Plug-in moment estimator.

Population mean μ and plug-in estimator with sample mean $\hat{\mu}$.

Estimand	**Estimator**	**Estimate**
μ	$\hat{\mu} = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$	$\hat{\mu} = 3.6$

Statistical inference: Ordinay Least Squares

We could be interested in the linear relationship (β) between two treatment varibales $\mathbf{X}$ and an outcome $\mathbf{y}$.

Estimand

$$\beta$$

Estimator

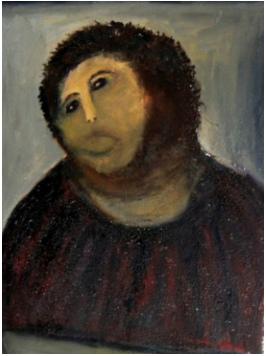
$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Estimate

$$\hat{\beta} = \begin{bmatrix} 3.4 \\ -0.20 \end{bmatrix}$$

Estimand, estimator, and esitmate

- **Estimates** are *best guesses*, but they never return you the *"true"*.



Maximum Likelihood Inference

- **1. Choose a Probability Model**: Select a **pdf** (e.g., normal, binomial) that best describes the DGP.

Maximum Likelihood Inference

- **1. Choose a Probability Model**: Select a **pdf** (e.g., normal, binomial) that best describes the DGP.
- **2. Maximize the Likelihood**: Estimate the **parameters** by maximizing the **likelihood** or **log-likelihood** function using observed data.

Maximum Likelihood Inference

- **1. Choose a Probability Model**: Select a **pdf** (e.g., normal, binomial) that best describes the DGP.
- **2. Maximize the Likelihood**: Estimate the **parameters** by maximizing the **likelihood** or **log-likelihood** function using observed data.
- **3. Evaluate the Fit**: Use the **estimated parameters** to compute probabilities, perform hypothesis tests, and make inferences about the population.

How do we evaluate our estimators

- **Bias**: Measures the **difference between the expected value** of the estimator and the true parameter. An **unbiased estimator** has $E[\hat{\theta}] = \theta$.
- **Efficiency**: Compares **variance** among unbiased estimators; the **most efficient estimator** has the **smallest variance** and thus provides **more precise estimates**.
- **Consistency**: Ensures that the estimator **converges to the true parameter** as the sample size increases ($\hat{\theta} \to \theta$ as $n \to \infty$), indicating **reliability** with large samples.

Chris will cover all these in detail next week in class!

`sample()` Function in R:

- Generates a **random sample** from a specified vector or a set of integers.
- **Syntax**: `sample(x, size, replace = FALSE, prob = NULL)`
  - `x`: The vector to sample from (e.g., `1:10`).
  - `size`: Number of elements to sample.
  - `replace`: If `TRUE`, allows **sampling with replacement**.
  - `prob`: A vector of **probabilities** for sampling each element.
- **Example**: `sample(x=1:5, size=3, replace=TRUE)` might return `4, 1, 2`.

`for` Loops in R:

- Executes a block of code **repeatedly** for each element in a sequence.
- **Basic Syntax**:

```
for (index in sequence) {
  # Code to execute
}
```

- **`index`**: Loop or control variable that takes values from the sequence.
- **`sequence`**: A vector or list of elements to iterate over.

`for` Loops in R:

- **Example**:

```r
# Use a for loop to iterate over numbers 1 to 5
for (i in 1:5) {
  print(paste("Index:", i, "Value:", i))
}
```

```
## [1] "Index: 1 Value: 1"
## [1] "Index: 2 Value: 2"
## [1] "Index: 3 Value: 3"
## [1] "Index: 4 Value: 4"
## [1] "Index: 5 Value: 5"
```

`function()` in R:

- Creates a **custom function** to encapsulate code and perform specific tasks.
- **Basic Syntax**:

```r
function_name <- function(arg1, arg2, ...) {
    # Code to execute
    return(result)
  }
```

- `arg1, arg2, ...`: Function arguments.
- `return()`: Specifies the output.

function() in R:

- **Example**:

```r
add_numbers <- function(x, y) {
    sum <- x + y
    return(sum)
}

# use the function

add_numbers(3, 4)
```

```
## [1] 7
```

Open lab 2

- Open the `lab2.Rmd` file and follow the explanations and code.
- We will then work on the `Lab2_practice.Rmd` file.
  - If we don't finish all the exercises today, we'll continue from where we left off next week.

FIN