Interactive Illustration of the Sampling Properties of Estimators

Markus Mößler

July 18, 2023

Abstract

This is the documentation of the implementation of a learning module for an interactive illustration of the sampling properties of estimators

1 Introduction

This repository contains the implementation of a learning module with interactive and animated illustrations of fundamental statistical concepts and properties.

2 Goal of the learning module (Why?)

Understanding the concept and properties of sampling distributions of estimators is one of the most important concepts of statistical inference, i.e, of learning from data about the underlying data generating process (DGP) formalized in probabilistic (population) model of interest, and thus, a fundamental part of empirical studies in social science in general and econometrics in particular. Practice, however, has shown that students often have difficulty understanding the concept of sampling distributions and their properties. One potential reason for this is that the sampling properties of estimators are often formulated and analyzed in an abstract way only. The goal of this learning module is to give a more intuitive understanding of the sampling properties of estimators using interactive and animated illustrations of simulation results.

One example is to understand the effect of increasing the sample size n on the sampling properties of the sample average $\overline{X}$ as an estimator for the mean μ of a random variable of interest as stated in the law of large numbers (LLN) and the central limit theorem (CLT). Using interactive illustrations of simulation results the students can increase the sample size and observe how the sample average gets closer to the mean (LLN) and how the sampling distribution of the standardized statistic of the sample average gets closer to the standard normal distribution (CLT).

The topics, i.e. the sample distribution of estimators, is part of every statistics and econometrics course and can be found in any introductory textbook for this field. However, by using interactive and animated illustrations the results we aim to provide a deeper and more intuitive understanding of these concepts. We believe that this kind of understanding is hard to achieve by just attending a *lecture*, reading a *textbook*, or, by chatting with *ChatGPT* about this topic. Certainly, another way to achieve this is to provide the implementation of the simulation studies or to let the students to implement the simulation studies themselves. However, this is often too big a hurdle, especially for undergraduate students.

3 Subject of the learning module (What?)

3.1 General

Subject of this learning module are the sampling properties of estimators for different data generating processes (DGPs). The DGPs are based on a statistical model with a particular parameter of interest, e.g., the mean of a Bernoulli random variable. The parameter(s) of interest of the DGPs are estimated using a particular estimator, e.g., the sample average. This learning module shows the effect of changes in the DGP, e.g., increasing the sample size n, on the sampling distribution of an estimator.

Note, the interactive illustration of the sampling properties of the sample average for increasing the sample size n, i.e., the large sample properties of the sample mean, is only one subject of interest. Other subjects are to understand the effect of omitted variable bias (OVB) and heteroskedasticity on the sampling distribution of the OLS estimator in a simple linear regression model.

3.2 Univariate random variables and sample average

3.2.1 Bernoulli distribution and sample average

This illustration shows the effect of changing the sample size n and the probability of success p of the Bernoulli distribution on the sampling distribution of the sample average as estimator for mean of the Bernoulli distribution.

3.2.2 Continuous uniform distribution and sample average

This illustration shows the effect of changing the sample size n and the lower bound a and the upper bound b of the continuous uniform distribution on the sampling distribution of the sample average as estimator for mean of the continuous uniform distribution.

3.3 Cross Section Linear Regression Model and Ordinary Least Squares

Illustration of the properties of the OLS estimator to estimate the slope coefficient β_1 of a linear regression model, i.e.,

$$Y_i = \beta_0 + \beta_1 X_i + u_i \tag{1}$$

3.3.1 Sampling distribution and sample size

This illustration shows the effect of increasing the sample size n on the sampling distribution of the OLS estimator for the slope coefficient of a simple linear regression model.

3.3.2 Sample Size and parameterization of the DGP

This illustration shows the effect of changing the parameters of the DFP, i.e.,

1. changing the sample size n,
2. changing the variance of u_i, i.e., $\sigma^2_{u_i}$, and,
3. changing the variance of X_i, i.e., σ^2_X,

on the sampling distribution of the OLS estimator for the slope coefficient of a simple linear regression model.

3.3.3 Effect of Heteroskedasticity

3.3.4 Effect of Omitted Variable Bias

4 Method/implementation of the learning module (How?)

4.1 General approach

For an interactive and immediate user experience it is useful to separate the simulation study and the interactive presentation of the results. This two-step procedure allows also a flexible implementation, i.e., the simulation studies can be conducted with any suitable software, e.g., *R*, *python*, etc. and the results can be presented interactively using basic web development languages, i.e., *html*, *css* and *javascript*.

Simulation study and illustration/reporting of the results:

- A realization of the DGP is simulated and the values of a given estimator is calculated.
- This procedure is repeated $10,000$ times.
- The simulation results, i.e., the sampling distribution of the estimator, are illustrated using barplots, scatterplots and/or histograms.
- Other simulation results can be reported using tables.
- The simulation study is performed for different specifications of the statistical model or for different estimators.
- For each specification or estimator the results, e.g., figures and/or tables are saved in different `.svg` and/or `html` files.
- The simulation study is performed using the programming language for statistical computing and graphics *R*.

Interactive presentation of the simulation results:

- The results for different specifications, e.g., figures and/or tables, are interactively embedded in a `.html` file and linked to a slider input.
- The effect of different specifications can be studied by changing the slider, i.e., the specification, and thus, the embedded results.
- The interactive integration of the illustrations into the `.html` file is based on javascript.

Additional material, e.g., for explanation purposes:

- Verbal text or mathematical formula inserted in the `.html` file directly or using “collapsibles”.
- Interactive animation of changes in the slider inputs with audio explanations.
  - For each figure one explanation for the figure is added
  - For each slider and figure one animated explanation can be added
- Any additional material, e.g., video explanations, can be added into the `html` file interactively or non-interactively.

4.2 Flexible presentation using *html*

The content of each learning module is presented using *html*, the most popular language to build web pages.[1] This allows a very flexible presentation of the results. The styling of the learning module is based on *css*, a popular language for styling web pages, and on *bootstrap*, a popular *css* framework. The interactive and animated feature of the learning module is based on *javascript*, a popular language for programming web pages. The *bootstrap* framework with its grid system allows the development of responsive, mobile-first websites with an appealing design. This enables an appealing experience of the learning also on mobile devices.

4.3 Interactive user experience using *javascript*

The following explanations are based on the illustration of the properties of the sample average as estimator for the mean of a Bernoulli random variable, i.e., based on module `ber-dis-sam-ave`. The values of the parameters can be changed using *bootstrap-slider* inputs[2]. For example for the input of the sample size the following *html* code snippet is used.

Changing the slider input values invokes *javascript* function.

The interactive experience of the illustrations is based on three *javascript* function which are linked to the sliders of the parameters. There is a

[1] An introduction to the most important web page development languages can be found here: https://www.w3schools.com/

[2] See: https://github.com/seiyria/bootstrap-slider

Code Snippet 1: *Html* code snippet for *bootstrap-slider* input

```
<p style='font-size: 12pt; text-align: center;'>Sample size \(n\)</p><p style='font-size: 12pt; text-align: center' id="sliderValue01Id"></p>
<input id="slider01Id" type="text" data-slider-min="0" data-slider-max="4" data-slider-step="1" data-slider-value="2"/>
```

Code Snippet 2: *Javascript* code snippet for interactive sliders (no 1): Initialize slider values

```
var nVec = [5, 10, 25, 50, 100]

// set initial value for slider 1
document.getElementById("sliderValue01Id").innerHTML = "\\(n = " + nVec[2] + "\\)";
var element = document.getElementById("sliderValue01Id");
MathJax.typeset([element]);
```

Code Snippet 3: *Javascript* code snippet for interactive sliders (no 2): Update slider values

```
var slider01 = new Slider("#slider01Id", {
        tooltip: "never",
        formatter: function(value) {
                return nVec[value]
        }
});
```

Code Snippet 4: *Javascript* code snippet for interactive sliders (no 3): Change illustration

```
slider01.on("slide", function() {

        var sliderValue01 = slider01.getValue() + 1;
        var sliderValue02 = slider02.getValue() + 1;

        document.getElementById("imageL1N1Id").setAttribute("src", "./figures/figure_01_" + sliderValue01 + "_" + sliderValue02 + ".svg");
        document.getElementById("imageL1N2Id").setAttribute("src", "./figures/figure_02_" + sliderValue01 + "_" + sliderValue02 + ".svg");
        document.getElementById("imageL1N3Id").setAttribute("src", "./figures/figure_03_" + sliderValue01 + "_" + sliderValue02 + ".svg");

        document.getElementById("sliderValue01Id").innerHTML = "\\(n = " + nVec[slider01.getValue()] + "\\)";
        var element = document.getElementById("sliderValue01Id");
        MathJax.typeset([element]);

})
```

4.4 Explained and animated user experience using *javascript*

The following explanations are based on the illustration of the properties of the sample average as estimator for the mean of a Bernoulli random variable, i.e., based on module `ber-dis-sam-ave`. The explanation of the figure is based on a simple *javascript* function `explainButto01Click`.

Code Snippet 5: *Javascript* snippet for explanation

```
explainButto01Click = function(audioId) {
        var audio = document.getElementById(audioId);
        audio.play();
}
```

The explained animation of the parameters is based on the *javascipt* function `sliderLoop()` and the functions `animateButto01Click()` and `animateButto02Click()`. Note, for each parameter a function `animateButtoXXClick()` is used, i.e., for the module `ber-dis-sam-ave` there are two functions, `animateButto01Click` and `animateButto02Click`. Furthermore, the body of both functions, `sliderLoop()` and `animateButtoXXClick()` depends on the number of figures (for `ber-dis-sam-ave` equals three) and the number of parameters (for `ber-dis-sam-ave` equals two) used in the module.

Code Snippet 6: *Javascript* snippet for explanation and animation (part I)

```
function sliderLoop(loopSliderValue01, loopSliderValue02) {

        var sliderValue01 = loopSliderValue01 + 1;
        var sliderValue02 = loopSliderValue02 + 1;

        document.getElementById("imageL1N1Id").setAttribute("src", "./figures/figure_01_" + sliderVal
        document.getElementById("imageL1N2Id").setAttribute("src", "./figures/figure_02_" + sliderVal
        document.getElementById("imageL1N3Id").setAttribute("src", "./figures/figure_03_" + sliderVal

        document.getElementById("sliderValue01Id").innerHTML = "\\(n = " + nVec[slider01.getValue()]
        var element = document.getElementById("sliderValue01Id");
        MathJax.typeset([element]);

        document.getElementById("sliderValue02Id").innerHTML = "\\(p = " + pVec[slider02.getValue()]
        var element = document.getElementById("sliderValue02Id");
        MathJax.typeset([element]);

        var slider = document.getElementById("slider01Id");
        slider.value = sliderValue01;

        var slider = document.getElementById("slider02Id");
        slider.value = sliderValue02;

};
```

4.5 Integration in the lecture

- The material of this learning module can be provided on a gradual basis using links to the specific illustrations/sub modules or as a complete course/module with a starting page and links to the sub modules.

- The material can be hosted on *GitHub* or on a learning platform such as *ILIAS*. The easiest way to host the material on a learning platform such as *ILIAS* is using a import interface for `.html` structures. In the case of the learning platform *ILIAS* this procedure is quite easy and flexible.

Code Snippet 7: *Javascript* snippet for explanation and animation (part II)

```
animateButto01Click = function(org, start, stop, audioId) {
        var audio = document.getElementById(audioId);
        audio.play();
        var ind = start;
        var loopSliderValue02 = slider02.getValue();
        var outerInterval = setInterval(function() {
                var loopSliderValue01 = ind;
                slider01.setValue(ind);
                sliderLoop(loopSliderValue01, loopSliderValue02);
                ind++;
                if (ind > stop) {
                        var innerInterval = setInterval(function() {
                                slider01.setValue(ind);
                                slider01.setValue(org);
                                sliderLoop(org, loopSliderValue02);
                                clearInterval(innerInterval);
                        }, 1000);
                        clearInterval(outerInterval);
                }
        }, 1000);
}
```

Code Snippet 8: *Javascript* snippet for explanation and animation (part III)

```
animateButto02Click = function(org, start, stop, audioId) {
        var audio = document.getElementById(audioId);
        audio.play();
        var ind = start;
        var loopSliderValue01 = slider01.getValue();
        var outerInterval = setInterval(function() {
                var loopSliderValue02 = ind;
                slider02.setValue(ind);
                sliderLoop(loopSliderValue01, loopSliderValue02);
                ind++;
                if (ind > stop) {
                        var innerInterval = setInterval(function() {
                                slider02.setValue(ind);
                                slider02.setValue(org);
                                sliderLoop(loopSliderValue01, org);
                                clearInterval(innerInterval);
                        }, 1000);
                        clearInterval(outerInterval);
                }
        }, 1000);
}
```

4.6 Structure of the learning module

- The learning module contains different sub modules where each sub module has a specific learning goal, e.g., "understand the effect of the sample size on the sample properties of the sample average to estimate the mean of a Bernoulli distribution".
- The material of a sub module is collected in a sub folder, e.g., `ber-dis-sam-ave`.
- The sub folder contains:
  - `.R` file, e.g., `ber_dis_sam_ave.R`, with the simulation study and the results stored in the `figures` and/or `tables` subdirectory
  - `figures` subdirectory with the illustrations of the simulation results
  - `audios` subdirectory with the audio text and files for the explanations using interactive animations
  - `tables` subdirectory with the reports of the simulation results (optional)
  - `.html` file, e.g., `ber_dis_sam_ave.html`, with the interactive representation of the illustrations
  - `myScript.js` with the javascript for the interactive illustrations
  - `myStyle.css` with the css styles for the interactive illustrations
  - Additional assets, e.g.,:
    - `.png` file with a logo for the header of the `.html` file
    - ...

4.7 Structure of the `.html` file

The content is divided across four bootstrap containers:

- 1st Container: Header of the module
- 2nd Container with three sections
  - Topic of the module
  - Data generating process (DGP)
  - Estimator and parameter of interest
- 3rd Container:
  - Parameter panel with the slider to change the parameter of interest
  - Illustration panel with four tabs for the illustration of
    - Sample draw: Particular outcome
    - Residual: Particular diagnostic
    - Estimates: Histogram of parameter estimates
    - Std. Estimates: Histogram of standardized parameter etimates
- 4th Container: More Details
  - Simulation Exercise:

References

[SW19] James H. Stock and Mark W. Watson. *Introduction to Econometrics*. Pearson, 2019.