Linear Regression Assumptions, Properties, and Goodness-of-fit

Babak Rezaee Daryakenari
The Institute of Political Science
Leiden University
s.rezaeedaryakenari@fsw.leidenuniv.nl
For more research methods handouts: https://babakrezaee.github.io

Spring 2020

What is an unbiased estimator?

Last session, we discussed that we would like to use a linear form to estimate the association between the outcome variable, y, and its covariates,$X = (x_1, x_2, ..., x_k)$:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + ... + \beta_k x_k + \epsilon \tag{1}$$

Also, we learned that one of the popular methods of estimating the parameters of the model, mostly known as the coefficients of regression model, is *Ordinary Least Square* (*OLS*). It is mathematically proven that under some assumptions, known as classic OLS assumptions, OLS estimated parameters are unbiased. Before discussing what assumptions make OLS an unbiased estimator, let's briefly talk about what is an estimator and when it is (ub)biased.

Any empirical analysis should deal with some sorts of uncertainties. For example, one of the main challenges in an empirical analysis is that we do not have access to the population. We, therefore, use a sample that spatially or temporally is limited. For example, when we want to study electoral behavior of citizens in European countries, we cannot conduct a survey that include everyone. Instead, we use a sample of citizens that gives us the best representation according to the research or policy questions that we are interested in exploring.

This means the association that we are trying to establish/measure/study to test our theoretical argument/hypothesis is to some degrees uncertain. Statistics allows us to measure and sometime address these uncertainties.

Given:

- A sample of observations: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$
- A statistical model with a set of *assumptions* where $\theta = (\beta_0, \beta_1)$ is unknown: $y = \beta_0 + \beta_1 x$
- An estimator of θ is an educated guess about its value using the observations: $g((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N))$

In Figure 1, the estimated $\hat{\beta}$ are distributed around the true value of β, and this means the estimator *in average* is doing well in telling us what is the true value of β. This is known as an unbiased estimator/estimation and mathematically formalized as:

$$E(\hat{\beta}) = \beta \tag{2}$$

Figure 2 and Figure 3 show two estimators that are not estimating the true value of β.

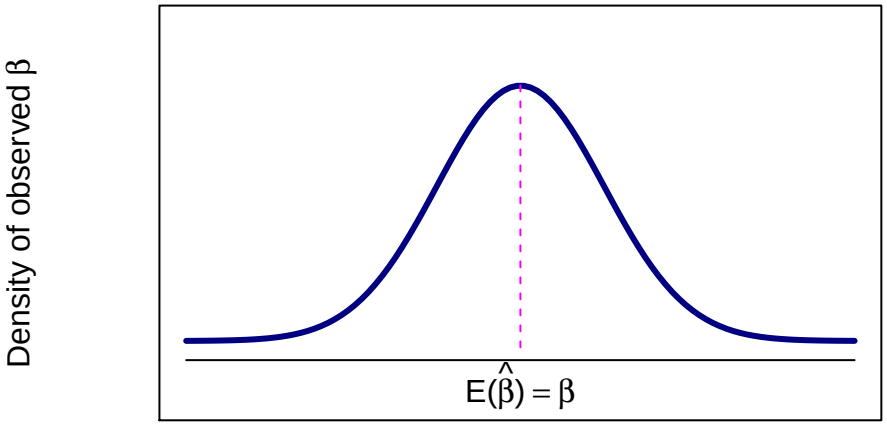

Figure 1: Estimations of $\beta : E(\hat{\beta}) = \beta$

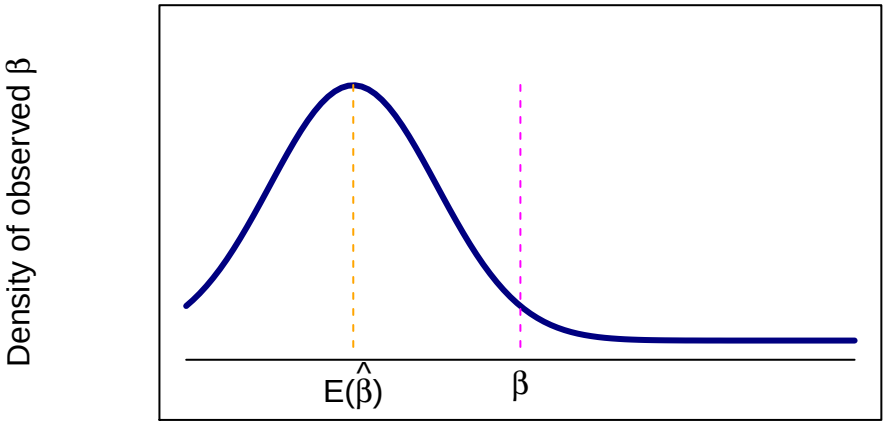

Figure 2: Estimations of $\beta : E(\hat{\beta}) < \beta$

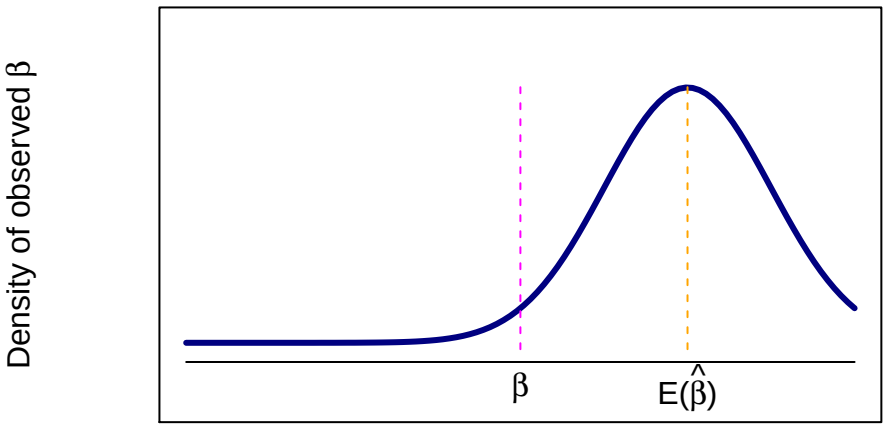

Figure 3: Estimations of $\beta : E(\hat{\beta}) > \beta$

The Gauss-Markov Assumptions

1. Linearity assumption: $y = \beta_0 + \beta_1 x + \epsilon$

2. X **is a full rank matrix.**

This assumption states that there is no perfect multicollinearity. In other words, the columns of X are linearly independent. This assumption is known as the identification condition.

3. $E(\epsilon|X) = 0$

This assumption - the zero conditional mean assumption - states that the disturbances average out to 0 for any value of X. Put differently, no observations of the independent variables convey any information about the expected value of the disturbance. The assumption implies that $E(y) = X\beta$. This is important since it essentially says that we get the mean function right.

4. $E(\epsilon\epsilon'|X) = \sigma^2 I$

This captures the familiar assumption of homoskedasticity and no auto-correlation.

5. X **and** ϵ **are orthogonal** $X \perp \epsilon$

X may be fixed or random, but must be generated by a mechanism that is unrelated to ϵ:

6. $\epsilon|X \sim N(0, \sigma^2 I)$

Th assumption 6 is not actually required for the Gauss-Markov Theorem. However, we often assume it to make hypothesis testing easier. The Central Limit Theorem is typically evoked to justify this assumption.

Theorem: The *Gauss-Markov Theorem* states that, conditional on assumptions 1-5, OLS estimator is the Best Linear, Unbiased and Efficient estimator (BLUE):

1. $\hat{\beta}$ is an unbiased estimator of β.
2. $\hat{\beta}$ is a linear estimator of β.
3. $\hat{\beta}$ has minimal variance among all linear and unbiased estimators.

Now that we know why we are fitting a function to our data, the function is linear, and OLS is the best available method to estimate the parameters of a linear model, the next step is learning how to estimate a model using a statistical software, in our case $\mathcal{R}$:

```
rawData=read.csv("https://raw.githubusercontent.com/babakrezaee/MethodsCourses/master/LeidenUniv_MAQM201
```

```
#Drop the missing values
rawData=na.omit(rawData, cols=c("avexpr", "logpgp95"))
```

```
# Generate first order linear model
lin.mod <- lm(logpgp95~avexpr, data=rawData)

lin.mod
```

```
##
## Call:
## lm(formula = logpgp95 ~ avexpr, data = rawData)
##
## Coefficients:
## (Intercept)       avexpr
##      4.8934       0.4868
```

```
summary(lin.mod)
```

```
##
## Call:
## lm(formula = logpgp95 ~ avexpr, data = rawData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8704 -0.4120  0.1456  0.4469  1.1309
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  4.89344    0.38256  12.791 < 0.0000000000000002 ***
## avexpr       0.48683    0.05725   8.504      0.0000000000133 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6486 on 55 degrees of freedom
## Multiple R-squared:  0.568,  Adjusted R-squared:  0.5602
## F-statistic: 72.32 on 1 and 55 DF,  p-value: 0.00000000001325
```

Although regression tables are popular, they are not the best way of presenting your estimated results. It is better while reporting the regression tables, we use a graph to present either all or the important variable:

```
library(arm)

coefplot(lin.mod)

coefplot(lin.mod, col.pts="blue")
```

```r
coefplot(lin.mod, intercept=TRUE)

labels_list <- c("Intercept","Rule of law")

coefplot(lin.mod, labels_list ,intercept=TRUE, frame.plot=TRUE, zeroColor='red')

coefplot(lin.mod, vertical=FALSE, var.las=1, frame.plot=TRUE)
```