ISSN: 2641-6336
DOI: 10.33552/ABBA.2020.04.000585
Annals of
Biostatistics & Biometric Applications
Iris Publishers

Research Article

Appendix 1

The Gauss-Markov Theorem

Suppose that the assumptions made in hold *and* that the errors are *homoscedastic*. The OLS estimator is the best (in the sense of smallest variance) linear conditionally unbiased estimator (BLUE) in this setting.

Let us have a closer look at what this means:

- Estimators of β1β1 that are linear functions of the Y1,...,Yn and that are unbiased conditionally on the regressor X1,...,Xn can be written as~β1=n∑i=1aiYiβ~Yiwhere the ai are weights that are allowed to depend on the Xi but *not* on the Yi.
- We already know that ~β1β~1 has a sampling distribution: ~β1. β~1 is a linear function of the Yi which are random variables. IfnowE(~β1|X1...,Xn)=β1, ,~β1β~1 is a linear unbiased estimator of β1, conditionally on the X1,...,Xn.
- We may ask if ~β1β~1 is also the *best* estimator in this class, i.e., the most efficient one of all linear conditionally unbiased estimators where "most efficient" means smallest variance. The weights ai play an important role here and it turns out that OLS uses just the right weights to have the BLUE property.

Consider the case of a regression of Yi..., Yn only on a constant. Here, the Yi are assumed to be a random sample from a population with mean μ and variance σ2. The OLS estimator in this model is simply the sample mean^β1=n∑i=11n=aiYi(5.3)β^1=∑i=1n⏟= ai Yi Clearly, each observation is weighted by

ai=1n. We also know that Var(^β1) =σ2n. We now can use R to conduct a simulation study that demonstrates what happens to the variance of (5.3) if different weightswi=1±ϵnwi=1±ϵn are assigned to either half of the sample Y1,...,YnY1,...,Yn instead of using 1n1n, the OLS weights.

```
# set sample size and number of repetitions

n <- 100

reps <- 1e5

# choose epsilon and create a vector of weights as defined above

epsilon <- 0.8

w <- c (rep ((1 + epsilon) / n, n / 2),

   rep ((1 - epsilon) / n, n / 2) )

# draw a random sample y_1...,y_n from the standard normal distribution,

# use both estimators 1e5 times and store the result in the vectors 'ols' and

# 'weightedestimator'

ols <- rep (NA, reps)

weighted estimator <- rep (NA, reps)

for (i in 1: reps) { y <- rnorm(n)  ols[i] <- mean(y) weighted estimator[i] <- cross prod (w, y) }
```

```
# plot kernel density estimates of the estimators' distributions:
# OLS
plot(density(ols),
    col = "purple",
    lwd = 3,
    main = "Density of OLS and Weighted Estimator",
    xlab = "Estimates")
# weighted
lines(density(weightedestimator),
     col = "steelblue",
     lwd = 3)
# add a dashed line at 0 and add a legend to the plot
Alline (v = 0, lty = 2)
legend ('top right',
      c ("OLS", "Weighted"),
      col = c ("purple", "steel blue"),
      lwd = 3)
```