Modelos Computacionais em Economia

Marcleiton Morais

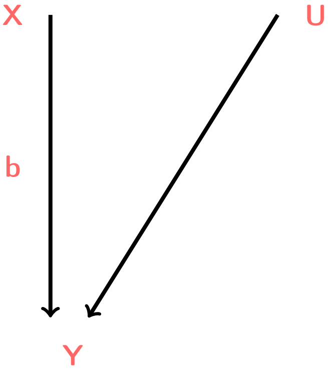
Universidade Federal do Tocantins (UFT)

15 de julho de 2021

Ordinary least squares (OLS) [Adams, 2020]

OLS is the work-horse model of microeconometrics. It is quite simple to estimate. It is straightforward to understand. It presents reasonable results in a wide variety of circumstances.

Estimating the Causal Effect (year of schooling (X) and person's income (Y))



Estimating the Causal Effect

A Linear Causal Model:

Individual i earns income y_i determined by their education level x_i and unobserved characteristics v_i.

$$y_i = a + bx_i + v_i$$

where a and b are the parameters that determine how much icome individual i earns and how much of that is determined by their leval of education.

Our goal is to estimate these parameters from the data we have.

Estimating the Causal Effect

Simulation of the Causal Effect

Sumulated data

- Linear relationship between x and y with an intercept of 2 and a slope of 3.
- Unobserved characteristics is ditributed **standard normal** **(**$v_i \sim \mathcal{N}(0,1)$ **)**.

We want to estimate the value of b, which has a true value of 3.

```
# Create a simulated data set
set.seed(123456789)
# use to get the exact same answer each time the code is run.
# you need to set the seed each time you want to get the
# same answer.
N <- 100
# Set N to 100, to represent the number of observations.
a <- 2
b <- 3 # model parameters of interest
# Note the use of <- to mean "assign".
x <- runif(N)
```

Estimating the Causal Effect

Simulation of the Causal Effect

Sumulated data

- Linear relationship between x and y with an intercept of 2 and a slope of 3.
- Unobserved characteristics is ditributed **standard normal** **(**$v_i \sim \mathcal{N}(0, 1)$ **)**.

We want to estimate the value of b, which has a true value of 3.

```
# create a vector where the observed characteristic, x,
# is drawn from a uniform distribution.
u <- rnorm(N)
# create a vector where the unobserved characteristic,
# v is drawn from a standard normal distribution.
y <- a + b*x + v # create a vector y
# * allows a single number to be multiplied through
# the whole vector
# + allows a single number to be added to the whole vector
# or for two vectors of the same length to be added together.
```

Estimating the Causal Effect

Averaging to Estimate the Causal Effect

Plot of x and y with the true relationship represented by the line.

```
mean(y[x > 0.95]) - mean(y[x < 0.05])
plot(x, y) # creates a simple plot
abline(a = 2, b = 3) # adds a linear function to the plot.
# a - intercept, b - slope.

#mean takes an average
#the logical expression inside the square brackets
#creates an index for the elements of y where the logical
#expression in x holds.
```

By taking the difference in the average of Y calculated at two different values of X, we can determine how X affects the average value of Y . In essence, this is what OLS does.

Estimating the Causal Effect

Assumptions of the OLS Model

Unobserved characteristics enter independently and additively:

- **Independence:** states that conditional on observed characteristics (the $X's$), the unobserved characteristic (the U) has independent effects on the outcome of interest (Y).
  Our estimated model does not allow students from wealthy families to be more likely to go to college and get a good job due to their family background.
- **Additive:** states that unobserved characteristics enter the model additively.
  Attending college increases everyone's income by the same amount

Matrix Algebra of the OLS Model

Standard Algebra of the OLS Model

Consider

$$y_i = a + bx_i + v_i$$

and let $a = 2$. So

$$b = \frac{y_i - 2 - v_i}{x_i} \tag{1}$$

This highlights two problems:

- **First:** the observed terms $(\{y_i, x_i\})$ are different for each person i, but Equation 1 states that b is exactly the same for each person.
- **Second:** second problem is that the unobserved term (v_i) is unobserved.

"kill two birds with one stone"

We can determine b by averaging:

$$\frac{1}{N}\sum^{N} y_i = \frac{1}{N}\sum^{N}(2 + bx_i + v_i)$$

Matrix Algebra of the OLS Model

Standard Algebra of the OLS Model

$$\frac{1}{N}\sum_{i=1}^{N} y_i = 2 + b\frac{1}{N}\sum_{i=1}^{N} x_i + \frac{1}{N}\sum_{i=1}^{N} v_i)$$

or

$$\overline{y} = 2 + b\overline{x} + \overline{v}$$

Dividing by $\overline{x}$

$$b = \frac{\overline{y} - 2 - \overline{v}}{\overline{x}}$$

We still cannot observe the unobserved terms, the $v_i^{\prime}s$. However, we can use

$$\hat{b} = \frac{\overline{y} - 2}{\overline{x}}$$

How close is our estimate to the true value of interest? How close is $\hat{b}$ to b?

Matrix Algebra of the OLS Model

Standard Algebra of the OLS Model

We need to assume that $\mathrm{E}(v_i) = 0$.

- **Law of Large Numbers**: if N is large, $\frac{1}{N}\sum_{i=1}^{N} v_i = \overline{v} = 0$

Algebraic OLS Estimator in R

```
b_hat <- (mean(y) - 2)/mean(x)
b_hat
```

```
[1] 3.459925
```

Matrix Algebra of the OLS Model

Using Matrice

In general, we do not know a and so we need to solve for both a and b.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 2 + 3x_1 + v_1 \\ 2 + 3x_2 + v_2 \\ 2 + 3x_3 + v_3 \\ \vdots \end{bmatrix}$$

or

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \end{bmatrix}$$

Matrix Algebra of the OLS Model

Multiplying Matrices in R

```
x1 = x[1:5] # only include elements 1 to 5.
X1 = cbind(1,x1) # create a matrix with a columns of 1
# cbind means column bind -
# it joins columns of the same length together.
# It returns a matrix-like object.
# Predict value of y using the model
X1%*%c(2,3)

# See how we can add and multiply vectors and numbers
# In R %*% represents standard matrix multiplication.
# Note that R automatically assumes c(2,3) is a column
# Compare to the true values
y[1:5]
```

Why aren't the predicted values equal to the true values?

Matrix Algebra of the OLS Model

Matrix Estimator of OLS

$$y = X\beta + v \tag{2}$$

y is a 100×1 column vector and X is a 100×2 rectangular matrix. We can use the same "division" idea, but we need a **full-rank square matrice**. They are invertible.

- **Square** by pre-multiplying it by its transpose: $X'y = X'X\beta + X'v$ $X'X$ is a 22 matrix as it is a 2100 matrix multiplied by a 1002 matrix.
- The **inverse**:

$$(X'X)^{-1}X'y = (X'X)^{-1}X'X\beta + (X'X)^{-1}X'v$$

$$\beta = (X'X)^{-1}X'y - (X'X)^{-1}X'v$$

$$\hat{\beta} = (X'X)^{-1}X'y$$

$(X'X)^{-1}X'v$ will generally be close to zero.

Matrix Algebra of the OLS Model

Matrix Estimator of OLS in R

A first column of 1's:

```
X <- cbind(1,x) # remember the column of 1's

A <- matrix(c(1:6),nrow=3)
# creates a 3 x 2 matrix.
A

# See how R numbers elements of the matrix.
t(A) # transpose of matrix A

t(A)%*%A
# matrix multiplication of the transpose by itself
```

Matrix Algebra of the OLS Model

Matrix Estimator of OLS in R

In our problem

```
t(X)%*%X

beta_hat <- solve(t(X)%*%X)%*%t(X)%*%y
beta_hat
```

Try running the simulation again, but changing N to 1,000. Are the new estimates closer to the their true values? Why?

We averaged over the unobserved term

```
solve(t(X)%*%X)%*%t(X)%*%u
```

Least Squares Method for OLS

Moment Estimation

Moment? A moment refers to the expectation of a random variable taken to some power. We say that the first moment of the unobserved characteristic is 0.

$$\mathrm{E}(v_i) = 0$$

From $y_i = a + bx_i + v_i$,

$$\mathrm{E}(y_i - a - bx_i) = 0$$

or, the **analog estimation**

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - a - bx_i)$$

We can make this number as close to zero as possible by minimizing the sum of squares.

Least Squares Method for OLS

Algebra of Least Squares

Again, we want to find the b, or better $\hat{b}$.

$$\min_{\hat{b}} \quad \frac{1}{N}\sum_{i=1}^{N}(y_i - a - \hat{b}x_i)^2$$

The first order condition is

$$\frac{1}{N}\sum_{i=1}^{N} -2x_i(y_i - a - \hat{b}x_i)^2 = 0$$

Divide both sides by -2 and rearranging

$$\hat{b} = \frac{\frac{1}{N}\sum_{i=1}^{N} x_i y_i - 2\frac{1}{N}\sum_{i=1}^{N} x_i}{\frac{1}{N}\sum_{i=1}^{N} x_i x_i}$$

Least Squares Method for OLS

Estimating Least Squares in R

Using the optimation problem:

```
optimize(function(b)
sum((y - 2 - b*x)^2), c(-10,10))$minimum
# optimize() is used when there is one variable.
#the function can be defined on the fly
# $minimum one of the outcomes from optimize()
```

Why do you think this is so far from the true value of 3?

Using the first order condition:

```
(mean(x*y) - 2*mean(x))/mean(x*x)
```

Least Squares Method for OLS

The *lm()* Function

The standard method for estimating OLS in R is to use the *lm()* function.

```
data1 <- as.data.frame(cbind(y,x))
# creates a data.frame() object which will
# be used in the next section.
lm1 <- lm(y ~ x) # lm creates a linear model object
# reports the number of elements of the list object
length(lm1)
# reports the names of the elements
length(lm1)
# reports the coefficient estimates
lm1$coefficients
# results from the matrix algebra.
t(beta_hat)
```

Measuring Uncertainty

Data Simulations

The simulation is run 1,000 times. In each case a sample of 100 is drawn using our parameters.

```
set.seed(123456789)
K <- 1000
sim_res <- matrix(NA,K,2)
# creates a 1000 x 2 matrix filled with NAs.
for (k in 1:K) {
        x <- runif(N)
        u <- rnorm(N)
        y <- a + b*x + u
        sim_res[k,] <- lm(y ~ x)$coefficients
        # print(k)
        # remove the hash to keep track of the loop
}
```

Measuring Uncertainty

Data Simulations

```
# xtable package produces fairly nice latex tables
colnames(sim_res) <- c("Est. of a", "Est. of b")
# install.packages("xtable")
require(xtable)
# summary produces a standard summary of the matrix.
sum_tab <- summary(sim_res)
rownames(sum_tab) <- NULL # no row names.
# NULL creates an empty object in R.
print(xtable(sum_tab), floating=FALSE)
```

What happens when the sample size is decreased or increased? Try $N = 10$ or $N = 5,000$.

Measuring Uncertainty

Introduction to the Bootstrap

- The idea is to **repeatedly draw pseudo-samples from the actual sample**, randomly and with replacement, and then for each pseudo-sample re-estimate the model.
- The **distribution of pseudo-estimates provides us with information on how uncertain our original estimate is.**

Measuring Uncertainty

Bootstrap in R

Fistor: we create a simulated sample data set.

```
set.seed(123456789)
K <- 1000
bs_mat <- matrix(NA,K,2)
for (k in 1:K) {
        index_k <- round(runif(N, min=1, max=N))
        # creates a pseudo-random sample.
        # draws N elements uniformly between 1 and N.
        data_k <- data1[index_k,]
        bs_mat[k,] <- lm(y ~ x,data=data_k)$coefficien
        # print(k)
}
```

Measuring Uncertainty

Bootstrap in R

```
tab_res <- matrix(NA,2,4)
tab_res[,1] <- colMeans(bs_mat)
# calculates the mean for each column of the matrix.
# inputs into the first column of the results matrix.
tab_res[,2] <- apply(bs_mat, 2, sd)
# a method for having the function sd() to act on each
# column of the matrix. Dimension 2 is the columns.
tab_res[,3] <- quantile(bs_mat[,1],c(0.025,0.975))
# calculates quantiles of the column at 2.5% and 97.5%
tab_res[,4] <- quantile(bs_mat[,2],c(0.025,0.975))
colnames(tab_res) <- c("Mean", "SD", "2.5%", "97.5%")
rownames(tab_res) <- c("Est. of a","Est. of b")
tab_res
```

Measuring Uncertainty

Bootstrap in R

```
#Standard Errors
print(xtable(summary(lm1)), floating=FALSE)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.9447	0.0643	30.26	0.0000
x	3.0633	0.1106	27.71	0.0000

The true values do lie in the 95% range.

Returns to Schooling

Do policies that encourage people to get more education, improve their economic outcomes?

A Linear Model of Returns to Schooling [Card, 1993]:

$$Income_i = \alpha + \beta Education_i + Unobserved_i$$

National Longitudinal Survey of Older and Younger Men (NLSM):

```
x <- read.csv("nls.csv",as.is=TRUE)
# It is important to add "as.is = TRUE",
# otherwise R may change your variables into "factors"
x$wage76 <- as.numeric(x$wage76)
x$lwage76 <- as.numeric(x$lwage76)
# "el wage 76" where "el" is for "log"
# Logging helps make OLS work better. Wages
# have a skewed distribution, and log of wages do not.
# creates a new data set
x1 <- x[is.na(x$lwage76)==0,]
```

Returns to Schooling

Ploting

```
lm1 <- lm(lwage76 ~ ed76, data=x1)
plot(x1$ed76,x1$lwage76, xlab="Years of Education",
ylab="Log Wages (1976)")
# plot allows us to label the charts
abline(a=lm1$coefficients[1],b=lm1$coefficients[2],lwd
```

Estimating Returns to Schooling

```
lm1
```

References

Adams, C. P. (2020).
Learning Microeconometrics with R.
Chapman and Hall/CRC.

Card, D. (1993).
Using geographic variation in college proximity to estimate the return to schooling.
NBER working paper, (w4483).

Obrigado!