

How can I get Monte Carlo standard errors for indirect effects in Stata?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I get Monte Carlo standard errors for indirect effects in Stata?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=163779>

Monte Carlo standard errors for indirect effects in Stata refer to a statistical method used to estimate the uncertainty in indirect effect measurements in Stata software. This method involves simulating multiple datasets based on the existing data and calculating the indirect effects in each simulated dataset. The resulting distribution of indirect effects can then be used to calculate standard errors, allowing for a more accurate estimation of the uncertainty in the indirect effect measurements. This approach is particularly useful in situations where the indirect effect is difficult to estimate using traditional methods, and can provide more reliable results in statistical analyses.

How can I get Monte Carlo standard errors for indirect effects? | Stata FAQ

This page shows how to obtain Monte Carlo standard errors and confidence intervals for indirect effects in a mediation analysis. We will illustrate the process using the hsbdemo dataset. For our example, science will act as the dependent variable, math as the independent variable and read as the mediator variable. We make no claims that this a good example of a mediation model. We only use it to illustrate the steps involved. We will begin by reading in the data and using the sureg command to generate the values we will use to compute the indirect effect as the product of coefficients.

use <https://stats.idre.ucla.edu/stat/data/hsbdemo>, clear

(highschool and beyond (200 cases))

sureg (read math)(science read math)

Seemingly unrelated regression

Equation Obs Parms RMSE "R-sq" chi2 P

read 200 1 7.662848 0.4386 156.26 0.0000

science 200 2 7.133989 0.4782 183.30 0.0000

| Coef. Std. Err. z P>|z|

-----+-----
read |

math | .724807 .0579824 12.50 0.000 .6111636 .8384504

_cons | 14.07254 3.100201 4.54 0.000 7.996255 20.14882
-----+-----

science |

read | .3654205 .0658305 5.55 0.000 .2363951 .4944459

math | .4017207 .0720457 5.58 0.000 .2605138 .5429276

_cons | 11.6155 3.031268 3.83 0.000 5.674324 17.55668

One method of computing the indirect effect, standard error and confidence interval is to use the nlcom command.

```
nlcom _b*_b
```

```
_nl_1: _b*_b
```

```
| Coef. Std. Err. z P>|z|
```

```
-----+-----
```

	Coef.	Std. Err.	z	P> z
_nl_1	.2648593	.0522072	5.07	0.000
	.1625351			.3671836

```
-----+-----
```

In the nlcom output, the Coef is the estimate of the indirect effect.

nlcom

uses the delta method to estimate the standard error.

The delta method is based on some

pretty strong normal theory assumptions. Additionally,

the confidence is also computed

with normality assumptions. Many researchers feel that

these normal theory assumption

are unlikely to hold for an estimate that is the product of two normally distributed coefficients. They might prefer a bootstrap approach such as shown on the [How can I perform Sobel-Goodman mediation tests?](#) page. Or, they might prefer a Monte Carlo approach, which we will cover on this page.

Basically, we will generate random normal values of our two coefficient with the parameter estimates as the mean and the standard error as the standard deviation. If the correlation of the two coefficients is zero (or close to zero), we can generate these value as two independent random normal distributions. If the correlation between the coefficient estimates is not zero we will have to generate the values as a random bivariate normal distribution.

So, let's check the correlation between the coefficients. The information we need can be found in the ereturn list matrix V which we will convert

to a correlation matrix.

```
mat V = e(V)
```

```
mat C = corr(V)
```

```
mat lis C
```

```
symmetric C
```

```
read: read: science: science: science:
```

```
math _cons read math _cons
```

```
read:math 1
```

```
read:_cons -.98460796 1
```

```
science:read -2.143e-31 2.177e-31 1
```

```
science:math 9.632e-16 -9.483e-16 -.66228013 1
```

```
science:_cons -1.205e-15 1.224e-15 -.3056154 -.50002438 1
```

The correlation coefficient for read predicted by math and the coefficient for science predicted by read is $-4.975e-30$. This value is close enough to zero that I will choose to ignore it.

Now we can go ahead and generate our random values. We begin by resetting the sample size to 50,000. Obviously if your sample size is already

50,000 or greater, you don't need to reset it. By the way, there is nothing magical about the 50,000 number. It just seemed like a nice large number. Next, we set the random number seed so that we can replicate the results.

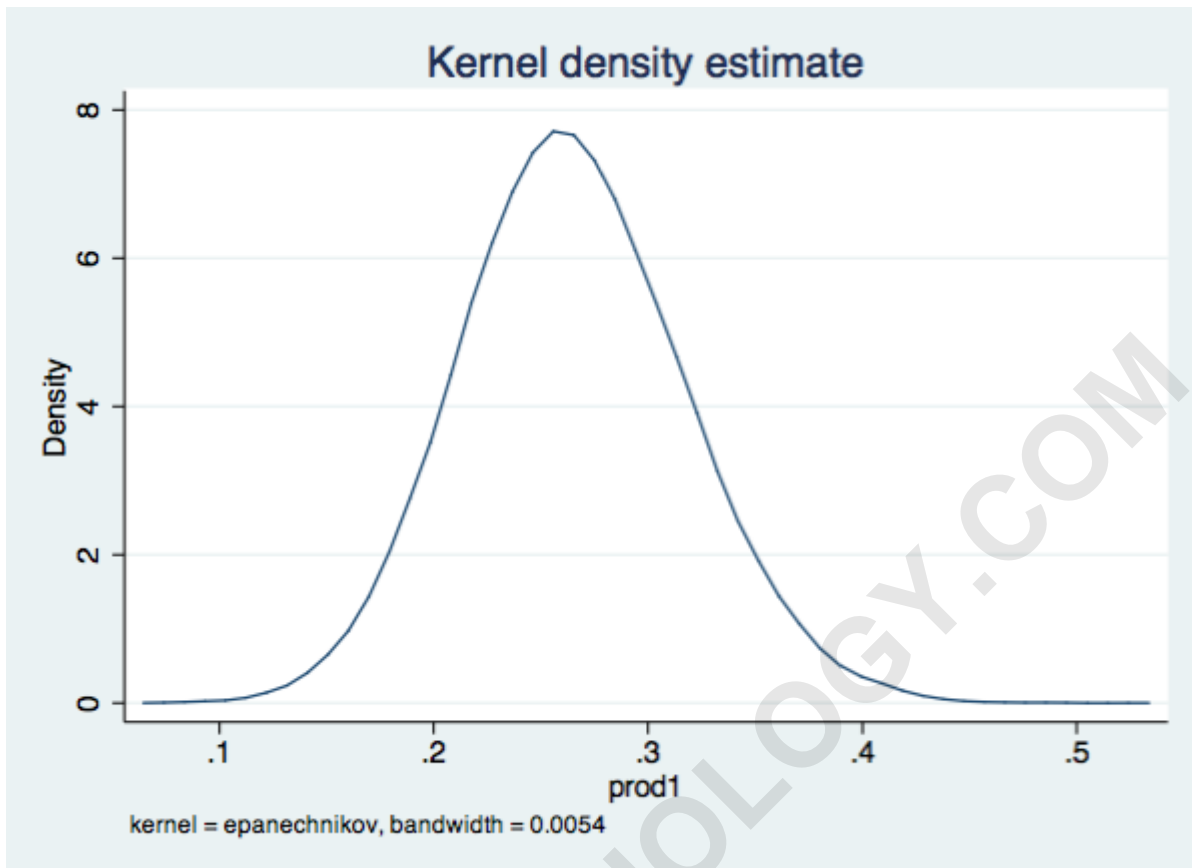
The next commands generate two random normal variates using the rnormal function with the original coefficient values as the mean and the standard error as the standard error. These commands are followed by generating the product of the random coefficients. This product is the random indirect effect, i.e., the mediation effect.

```
set obs 50000  
set seed 987654  
generate a1 = rnormal(_b,_se)  
generate b1 = rnormal(_b,_se)  
generate prod1 = a1*b1
```

Let's look at prod1 to see how things went.

kdensity

prod1



su

mmarize prod1

Variable | Obs Mean Std. Dev. Min Max

-----+-----

prod1 | 50,000 .2652127 .0520892 .0681946 .5289696

_pctile prod1, p(2.5 97.5)

return list

scalars:

r(r1) = .1667018160223961

r(r2) = .3709507137537003

The kernel density plot shows a normal distribution of the random products, the indirect effect.

The mean random indirect effect is .2652127 with a standard deviation of .0520892. The standard deviation of a sampling distribution is a standard error. So, .0520892 is the Monte Carlo standard error for our indirect effect. Finally, the Monte Carlo 95% confidence interval is (.167 .371).

But what if the coefficients in the two sureg equations are correlated?

When coefficients are correlated you will only need to change a few lines of code for the Monte Carlo analysis. The primary change is that the two rnormal function calls are replaced

with the drawnorm command. To use the drawnorm command we will create a vector of means (m), a vector of standard deviations (s), and a correlation matrix (c).

Alternatively, you can use a covariance matrix instead of the standard deviation vector and correlation matrix. Starting from the set seed

command

here is what the code looks like.

```
set seed 6574839
```

```
mat m = (_b, _b)
```

```
mat sd = (_se, _se)
```

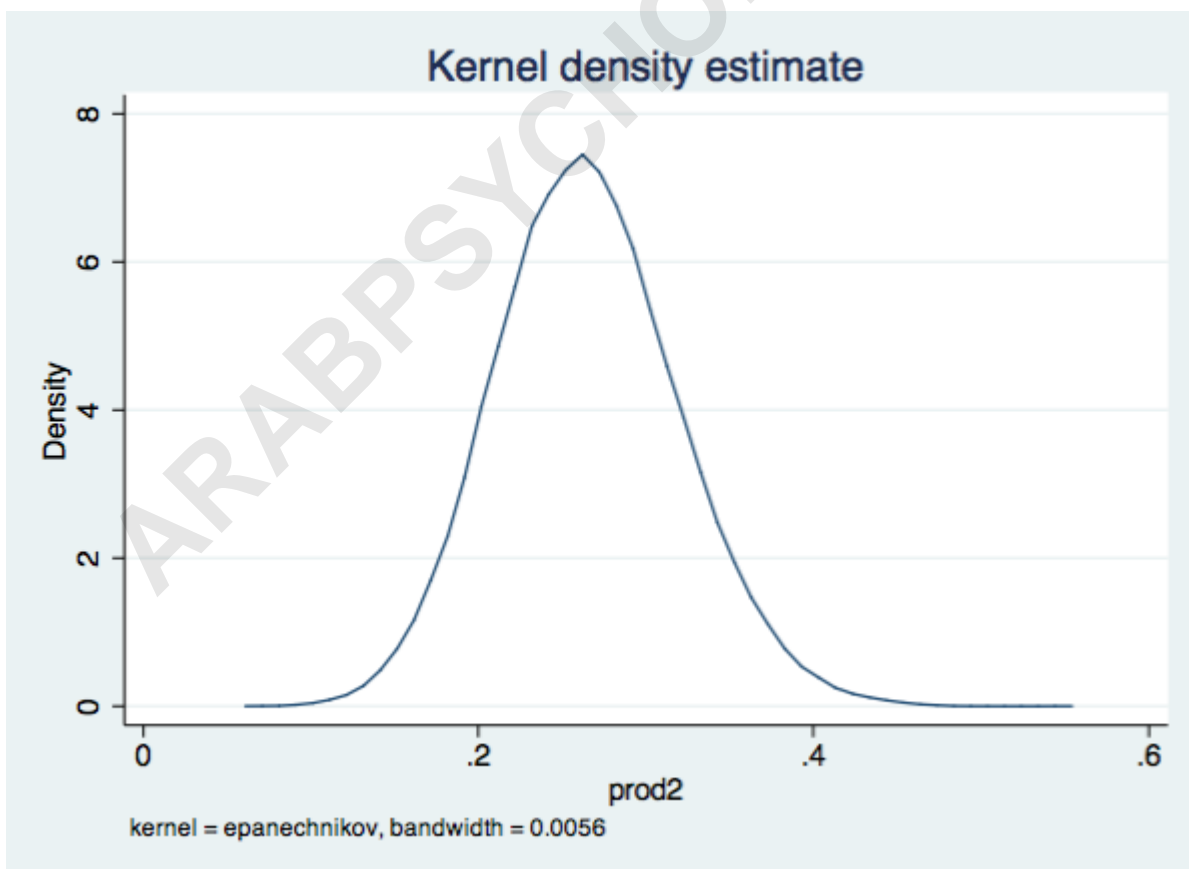
```
mat c = (1, .1 .1, 1)
```

```
drawnorm a2 b2, means(m) sds(sd) corr(c)
```

```
generate prod2 = a2*b2
```

kdensity

prod2



SU

mmarize prod2

Variable | Obs Mean Std. Dev. Min Max

```
-----+-----
prod2 | 50,000 .2649602 .0541438 .0734409 .5078928
```

_pctile prod2, p(2.5 97.5)

return list

scalars:

r(r1) = .1641761288046837

r(r2) = .37570521235466

Once again the kernel density plot appears to be normal.

This time the mean random indirect effect is .2649602 with a standard deviation of .0541438. The standard deviation of a sampling distribution is a standard error. So, .0541438 is the Monte Carlo standard error for our indirect effect. Finally, the Monte Carlo 95% confidence interval is (.164 .376).

Here is a comparison of the standard errors using the

three methods from this page.

Delta method using nlcom .0522072

Monte Carlo, coefficients uncorrelated .0520892

Monte Carlo, coefficients correlated .1 .0541438

As you can see, all of these standard errors fairly close to one another.

ARABPSYCHOLOGY.COM