How can I create dummy variables in Stata?

Authored by

stats writer

July 1, 2024

To create dummy variables in Stata, one can use the "encode" or "tabulate" command. The "encode" command assigns a numeric value to each category of a variable, while the "tabulate" command creates a separate dummy variable for each category. Both methods are useful for creating dummy variables to represent categorical data in statistical analyses. It is important to properly label and define the dummy variables to accurately interpret the results. Additionally, Stata offers various options and functions for manipulating and managing dummy variables, providing flexibility in creating and using them in data analysis.

How can I create dummy variables in Stata? | Stata FAQ

There are two easy ways to create dummy variables in Stata. Let's begin with a simple dataset that has three levels of the variable `group`:

```
input group
1
1
2
3
2
2
1
3
3
end
```

We can create dummy variables using the `tabulate` command and the `generate( )` option, as shown below.

```
tabulate group, generate(dum)
```

group | Freq. Percent Cum.

----------+----------

1 | 3 33.33 33.33

2 | 3 33.33 66.67

3 | 3 33.33 100.00

----------+----------

Total | 9 100.00

```
list
```

group dum1 dum2 dum3

1. 1 1 0 0

2. 1 1 0 0

3. 2 0 1 0

4. 3 0 0 1

5. 2 0 1 0

6. 2 0 1 0

7. 1 1 0 0

8. 3 0 0 1

9. 3 0 0 1

The `tabulate` command with the `generate` option created three dummy variables called `dum1`, `dum2` and `dum3`.

An Example Using the High School and Beyond Dataset

Using High School and Beyond dataset we wish to account for variability in the writing test scores using information on reading, math and the program type the student is in. The categorical variable `prog` has three levels: 1) general program, 2) academic program, and 3) vocational program. First, we will load the dataset from the Internet, then we will create dummy variables for `prog` using the `tabulate` command.

```
use https://stats.idre.ucla.edu/stat/stata/notes/hsb2, clear
```

tabulate prog, generate(prog)

type of |
program | Freq. Percent Cum.
----------+-----------------------------------
general | 45 22.50 22.50
academic | 105 52.50 75.00
vocation | 50 25.00 100.00
----------+-----------------------------------
Total | 200 100.00

The `tabulate` command with the `generate` option created the following variables: `prog1`, `prog2`, and `prog3`. In a regression analysis we can only use two of the three dummy variables. Since `prog` has three levels it uses two degrees of freedom. Here is the regression analysis.

```
regress write read math prog2 prog3
```

```
Source | SS df MS Number of obs = 200
-------------+------------------------------ F( 4, 195) = 41.03
Model | 8170.58624 4 2042.64656 Prob > F = 0.0000
Residual | 9708.28876 195 49.7860962 R-squared = 0.4570
-------------+------------------------------ Adj R-squared = 0.4459
Total | 17878.875 199 89.843593 Root MSE = 7.0559

------------------------------------------------------------------------------
write | Coef. Std. Err. t P>|t|
-------------+----------------------------------------------------------------
read | .289028 .0659478 4.38 0.000 .1589656 .4190905
math | .3587215 .0745443 4.81 0.000 .2117048 .5057381
prog2 | .6647754 1.32845 0.50 0.617 -1.955198 3.284749
prog3 | -2.253484 1.468445 -1.53 0.127 -5.149556
```

.6425886

_cons | 19.00854 3.40933 5.58 0.000 12.28465 25.73243

In the analysis all of the variables were statistically significant except for `prog2` and `prog3`. However, it is necessary to remember that it is the combination of `prog2` and `prog3` that makes up the variable program type.

Let's test `prog2` and `prog3` together.

```
test prog2 prog3
```

(1) prog2 = 0.0

(2) prog3 = 0.0

F(2, 195) = 2.32

Prob > F = 0.1015

As it turns out, by testing `prog2` and `prog3` together, we find that the variable program type is not statistically significant.

We can also do this in one step using the `i.` or factor variable notation, as shown below. Factor variables

create indicator variables from categorical variables and are allowed with most estimation and postestimation commands Note how the results below match those above exactly.

```
regress write read math i.prog
```

Source | SS df MS Number of obs = 200
----------+---------- F(4, 195) = 41.03
Model | 8170.58624 4 2042.64656 Prob > F = 0.0000
Residual | 9708.28876 195 49.7860962 R-squared = 0.4570
----------+---------- Adj R-squared = 0.4459
Total | 17878.875 199 89.843593 Root MSE = 7.0559

--
write | Coef. Std. Err. t P>|t|
-----------+--
read | .289028 .0659478 4.38 0.000 .1589656 .4190905
math | .3587215 .0745443 4.81 0.000 .2117048 .5057381
|
prog |
academic | .6647754 1.32845 0.50 0.617 -1.955198 3.284749

vocation | -2.253484 1.468445 -1.53 0.127 -5.149556 .6425886

|

_cons | 19.00854 3.40933 5.58 0.000 12.28464 25.73243

As we did in the prior example, we can test the overall effect of program

type with the `test` command as shown below.

```
test 2.prog 3.prog
```

(1) 2.prog = 0

(2) 3.prog = 0

F(2, 195) = 2.32

Prob > F = 0.1015

The `contrast` command can be used to get the multi-degree-of-freedom test of the categorical variable.

```
contrast prog
```

Contrasts of marginal linear predictions

Margins : asbalanced

```
-----------------------------------------------
| df F P>F
-------------+---------------------------------
prog | 2 2.32 0.1015
|
Denominator | 195
-----------------------------------------------
```

For more information

See the Stata manual on tabulate and factor variables.