How can I analyze a subpopulation of my survey data in Stata?

Authored by

stats writer

July 1, 2024

To analyze a subpopulation of survey data in Stata, first identify the specific subpopulation you want to examine. Then, use the "if" or "in" commands to specify the criteria for the subpopulation. Next, use the appropriate statistical commands to analyze the data for the subpopulation. This can include descriptive statistics, regression analysis, or other methods. Finally, interpret the results and draw conclusions specific to the subpopulation. It is important to document the steps taken in the analysis and any assumptions made.

How can I analyze a subpopulation of my survey data in Stata? | Stata FAQ

When analyzing survey data, it is common to want to look only a certain respondents, perhaps only women, or only respondents over age 50. When analyzing these subpopulations (AKA domains), you need to use the appropriate option. Stata has two subpopulation options that are very flexible and easy to use. Using the subpopulation option(s) is extremely important when analyzing survey data. If the data set is subset, meaning that observations not to be included in the subpopulation are deleted from the data set, the standard errors of the estimates cannot be calculated correctly. When the subpopulation option(s) is used, only the cases defined by the subpopulation are used in the calculation of the estimate, but all cases are used in the calculation of the standard errors. For more information on this issue, please see Sampling

Techniques, Third Edition by William G. Cochran (1977) and Small Area Estimation by J. N. K. Rao (2003).

For the sake of consistency, we will use the mean command for all of our examples. However, the subpop and over options work the same for all svy commands.

We will start by looking at the mean of our continuous variable, ell. Next, we will consider two variables to use with the subpop option, yr_rnd, which is coded 0/1, and both, which is coded 1/2. As you will see, the subpop option handles these two variables differently.

```
use
https://stats.idre.ucla.edu/stat/stata/seminars/svy_stata_intro/strsrs, clearsvy: mean ell
(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6194
Design df = 618

--------------------------------------------------------------
|  Linearized
```

| Mean Std. Err.

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

ell | 22.83578 .669696 21.52063 24.15094

Here we can see that yr_rnd is coded 0/1. (The missing option is used here to show that there are no missing values for this variable. We will want to know this later on.) Notice in the output of the svy: tab command that there are 789.6 cases coded 1. (It is not a whole number because we are estimating this value using the probability weights.) In the output of the svy: mean command, we also see that 789.552 cases are included in the subpopulation.

svy: tab yr_rnd, count nolabel missing
(running tabulate on estimation sample)

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6193.9997
Design df = 618

---

yr_rnd | count

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

0 | 5404
1 | 789.6
|
Total | 6194

Key: count = weighted counts

svy, subpop(yr_rnd): mean ell
(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6194
Subpop. no. obs = 79
Subpop. size = 789.552
Design df = 618

| Linearized
| Mean Std. Err.
---+---
ell | 43.50105 2.658549 38.28016 48.72193

Now let's try to use a variable coded 1/2 instead of 0/1. Here we can see that both is coded 1/2. (The missing option is used here to show that there are no missing values for this variable. We will want to know this later on.) Notice in the output of the svy: tab command that there are 1888 cases coded 1. However, in the output of the svy: mean command, we see that all of the observations, 6194 cases, are included in the subpopulation. This is because the subpop option must have a true/false variable. As stated in the Stata Survey manual, when the subpop option is used, the subpopulation is actually defined by the 0s (false), which indicate those cases to be excluded from the subpopulation. Non-0 values are included in the analysis, except for missing values, which are excluded from the analysis. Because we have no cases coded as 0, all of the cases are included in the subpopulation, as explained in the note in the output.

svy: tab both, count nolabel missing
(running tabulate on estimation sample)

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6193.9997

Design df = 618

---

both | count

---+---

1 | 1888

2 | 4306

|

Total | 6194

Key: count = weighted counts

svy, subpop(both): mean ell

(running mean on estimation sample)

Note: subpop() subpopulation is same as full population

subpop() = 1 indicates observation in subpopulation

subpop() = 0 indicates observation not in subpopulation

Survey: Mean estimation

Number of strata = 2 Number of obs = 620

Number of PSUs = 620 Population size = 6194

Subpop. no. obs = 620

```
Subpop. size = 6194
Design df = 618

-------------------------------------------------------------
| Linearized
| Mean Std. Err.
------------+------------------------------------------------
ell | 22.83578 .669696 21.52063 24.15094
-------------------------------------------------------------
```

Now let's create a copy of both and recode the 1s to 0s. We will also set some values to missing, to see what happens with missing values in the subpopulation variable. The output of the tab command shows us that the recoding went as planned. The output of the svy: mean command shows that the all of the cases not coded 0 or missing (the 424 cases coded as 2) are included in the subpopulation. Notice the note that Stata provides when the subpopulation variable is not coded 0/1.

```
gen both1 = bothrecode both1 (1=0)
(both1: 189 changes made)

replace both1 = . if _n < 11
```

(10 real changes made, 10 to missing)

tab both1, missing

both1 | Freq. Percent Cum.
------------+-----------------------------------
0 | 186 30.00 30.00
2 | 424 68.39 98.39
. | 10 1.61 100.00
------------+-----------------------------------
Total | 620 100.00

svy, subpop(both1): mean ell
(running mean on estimation sample)

Note: subpop() takes on values other than 0 and 1
subpop() != 0 indicates subpopulation

Survey: Mean estimation

Number of strata = 2 Number of obs = 610
Number of PSUs = 610 Population size = 6094.03
Subpop. no. obs = 424
Subpop. size = 4235.65
Design df = 608

| Linearized

| Mean Std. Err.

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

ell | 22.03727 .894207 20.28116 23.79338

You can also use if when defining your subpopulation. It should be stressed that this is VERY different from using if to remove cases from an analysis. Using if in the subpop option does not remove cases from the analysis. The cases excluded from the subpopulation by the if are still used in the calculation of the standard errors, as they should be.

svy, subpop(yr_rnd if mobility < 50): mean ell

(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620

Number of PSUs = 620 Population size = 6194

Subpop. no. obs = 78

Subpop. size = 779.555

Design df = 618

| Linearized
| Mean Std. Err.
---+---
ell | 43.86654 2.668957 38.62521 49.10786
---

svy, subpop(yr_rnd if mobility < 50 & hsg < 80): mean ell
(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6194
Subpop. no. obs = 78
Subpop. size = 779.555
Design df = 618

| Linearized
| Mean Std. Err.
---+---

ell | 43.86654 2.668957 38.62521 49.10786

You can use either subpop or over with multiple variables to create the subpopulation that you want. Let's see some examples using the over option. First, we will use yr_rnd, our 0/1 variable, then both, our 1/2 variable. Notice that the output is different from the output using the subpop option in that both categories of the variable are given, and there is no note when a 1/2 variable is used. Please note that the over option is only available for the survey commands mean, proportion, ratio and total.

svy: mean ell, over(yr_rnd)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6194
Design df = 618

0: yr_rnd = 0
No: yr_rnd = No

```
-------------------------------------------------------------
| Linearized
Over | Mean Std. Err.
------------+------------------------------------------------
ell |
0 | 19.81673 .6771138 18.48701 21.14646
No | 43.50105 2.658549 38.28016 48.72193
-------------------------------------------------------------

svy: mean ell, over(both)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6194
Design df = 618

No: both = No
Yes: both = Yes

-------------------------------------------------------------
| Linearized
Over | Mean Std. Err.
------------+------------------------------------------------
```

ell |
No | 24.64196 1.329677 22.03073 27.25319
Yes | 22.04363 .8854687 20.30473 23.78252

Now let's use both yr_rnd and both as the subpopulation variables. First we will use the svy: tab command to ensure that there are cases in all four categories. Then we use the svy: mean command with the over option.

svy: tab yr_rnd both, count
(running tabulate on estimation sample)

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6193.9997
Design df = 618

| met both targets
yr_rnd | No Yes Total
----------+----------
0 | 1659 3746 5404
No | 229.9 559.7 789.6
|

Total | 1888 4306 6194

Key: weighted counts

Pearson:

Uncorrected chi2(1) = 0.0807

Design-based F(1, 618) = 0.0896 P = 0.7647

svy: mean ell, over(yr_rnd both)

(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620

Number of PSUs = 620 Population size = 6194

Design df = 618

Over: yr_rnd both

_subpop_1: 0 No

_subpop_2: 0 Yes

_subpop_3: No No

_subpop_4: No Yes

| Linearized

Over | Mean Std. Err.

```
-------------+------------------------------------------------
ell |
_subpop_1 | 21.72287 1.246971 19.27405 24.17168
_subpop_2 | 18.9728 .8907884 17.22346 20.72213
_subpop_3 | 45.70399 4.841131 36.19692 55.21105
_subpop_4 | 42.59631 3.1987 36.31468 48.87795
-------------------------------------------------------------
```

Below we create a new variable from emer with four categories. Then we will use this variable with yr_rnd and both; all combinations of the variables are shown in the output. This is often very useful and saves you from having to create a new subpopulation variable. However, if each of your variables have many categories, the output can become long and cumbersome, especially if you are only interested in a few combinations of categories.

```
egen emergrp = cut(emer), group(5)
svy: mean ell, over(emergrp)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620
```

Number of PSUs = 620 Population size = 6194
Design df = 618

1: emergrp = 1
2: emergrp = 2
3: emergrp = 3
4: emergrp = 4

| Linearized
Over | Mean Std. Err.
----------+---
ell |
1 | 14.53282 .9707122 12.62652 16.43911
2 | 19.21555 1.594388 16.08447 22.34662
3 | 26.41136 1.815472 22.84612 29.97661
4 | 38.60681 1.926595 34.82334 42.39028
---

svy: mean ell, over(emergrp yr_rnd both)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620

Number of PSUs = 620 Population size = 6194

Design df = 618

Over: emergrp yr_rnd both

_subpop_1: 1 0 No

_subpop_2: 1 0 Yes

_subpop_3: 1 No No

_subpop_4: 1 No Yes

_subpop_5: 2 0 No

_subpop_6: 2 0 Yes

_subpop_7: 2 No No

_subpop_8: 2 No Yes

_subpop_9: 3 0 No

_subpop_10: 3 0 Yes

_subpop_11: 3 No No

_subpop_12: 3 No Yes

_subpop_13: 4 0 No

_subpop_14: 4 0 Yes

_subpop_15: 4 No No

_subpop_16: 4 No Yes

| Linearized

Over | Mean Std. Err.

```
-------------+----------------------------------------------
ell |
_subpop_1 | 19.04537 2.13396 14.85468 23.23606
_subpop_2 | 12.37844 1.113872 10.19101 14.56588
_subpop_3 | 17.33333 5.380975 6.766121 27.90055
_subpop_4 | 25.9189 6.110832 13.91839 37.91941
_subpop_5 | 18.32239 2.38222 13.64416 23.00061
_subpop_6 | 18.38206 2.244956 13.97339 22.79072
_subpop_7 | 26.01227 12.7449 .9837157 51.04082
_subpop_8 | 27.36803 5.615893 16.33949 38.39658
_subpop_9 | 22.41529 2.814223 16.8887 27.94189
_subpop_10 | 24.47567 2.371769 19.81797 29.13337
_subpop_11 | 50.29112 6.681067 37.17077 63.41146
_subpop_12 | 39.33854 7.992979 23.64185 55.03524
_subpop_13 | 29.11945 2.818771 23.58392 34.65498
_subpop_14 | 32.95607 2.611229 27.82811 38.08403
_subpop_15 | 54.09091 6.870972 40.59763 67.58419
_subpop_16 | 57.8 3.730597 50.47382 65.12618
------------------------------------------------------------
```

The subpop option can be combined with the over option. This is handy because if cannot be used with the over option. By combining the options, you can have "the best of both worlds."

svy, subpop(yr_rnd if mobility < 50 & hsg < 80): mean ell, over(emergrp both)
(running mean on estimation sample)

Survey: Mean estimation

Number of strata = 2 Number of obs = 620
Number of PSUs = 620 Population size = 6194
Subpop. no. obs = 78
Subpop. size = 779.555
Design df = 618

Over: emergrp both
_subpop_1: 1 No
_subpop_2: 1 Yes
_subpop_3: 2 No
_subpop_4: 2 Yes
_subpop_5: 3 No
_subpop_6: 3 Yes
_subpop_7: 4 No
_subpop_8: 4 Yes

| Linearized
Over | Mean Std. Err.

```
-------------+----------------------------------------------
ell |
_subpop_1 | 17.33333 5.380975 6.766121 27.90055
_subpop_2 | 25.9189 6.110832 13.91839 37.91941
_subpop_3 | 26.01227 12.7449 .9837157 51.04082
_subpop_4 | 27.36803 5.615893 16.33949 38.39658
_subpop_5 | 50.29112 6.681067 37.17077 63.41146
_subpop_6 | 42.38135 8.451664 25.78389 58.97882
_subpop_7 | 54.09091 6.870972 40.59763 67.58419
_subpop_8 | 57.8 3.730597 50.47382 65.12618
------------------------------------------------------------
```