Example 7 — Table of regression results using survey data

Description   Remarks and examples   Reference   Also see

Description

In this example, we demonstrate how to use `collect` to create a table of regression results when working with complex survey data.

Remarks and examples

Remarks are presented under the following headings:

> *Introduction*
> *Table of regression results with complex survey data*

Introduction

Throughout this manual, we have used data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981), ignoring the fact that it was derived from a complex survey design. Below, we report the current survey settings:

```
. use https://www.stata-press.com/data/r19/nhanes2l
(Second National Health and Nutrition Examination Survey)

. svyset
Sampling weights: finalwgt
             VCE: linearized
     Single unit: missing
        Strata 1: strata
 Sampling unit 1: psu
           FPC 1: <zero>
```

Because the survey design has already been declared for these data, we can perform our analyses accounting for the complex survey design by prefixing our estimation commands with the `svy` prefix. We will use this prefix when fitting a regression model below.

Table of regression results with complex survey data

We would like to fit the model we fit in [TABLES] **Example 5** but using the `svy` prefix to account for the complex survey design of our data. Also, we would like to include information about the survey design in our table, specifically the number of strata and primary sampling units. Below, we create a new collection named `ex7`. Then, we fit the model and use the `collect` prefix to collect all the results from `e()`. We also use the `quietly` prefix to suppress the output:

```
. collect create ex7
(current collection is ex7)

. quietly: collect: svy: regress bpsystol bmi i.agegrp i.sex i.race
```

Example 7 — Table of regression results using survey data 2

Then, we arrange the items in our collection with `collect layout`. We specify that the coefficients (`_r_b`) for each covariate (`colname`) be placed on the rows, along with the number of strata (`N_strata`) and the primary sampling units (`N_psu`). All collected results are stored in the dimension `result`, and we use brackets (`[]`) to refer to the levels of a dimension. On the columns, we place the results for each command (`cmdset`). In this case, we collected results from a single command, but including this dimension will allow us to add a label at the top.

```
. collect layout (colname#result[_r_b] result[N_strata N_psu]) (cmdset)
Collection: ex7
      Rows: colname#result[_r_b] result[N_strata N_psu]
   Columns: cmdset
   Table 1: 28 x 1
```

	1
Body mass index (BMI)	
Coefficient	1.314165
20-29	
Coefficient	0
30-39	
Coefficient	.9554468
40-49	
Coefficient	6.40242
50-59	
Coefficient	14.87049
60-69	
Coefficient	21.61949
70+	
Coefficient	27.90405
Male	
Coefficient	0
Female	
Coefficient	-5.616304
White	
Coefficient	0
Black	
Coefficient	1.397293
Other	
Coefficient	-.3119783
Intercept	
Coefficient	88.59293
Number of strata	31
Number of sampled PSU	62

We can polish our table with a few modifications. First, we add a label to the level of `cmdset`, which will indicate the outcome variable. Then, because we are reporting only coefficients for each variable, we can hide the label `Coefficient`. This label is attached to the coefficients (`_r_b`), which are a level of the dimension `result`. Below, we hide the label for this level of the dimension with `collect style header`.

```
. collect label levels cmdset 1 "Model for systolic BP"
. collect style header result[_r_b], level(hide)
```

If you are wondering why we do not just hide the labels for all levels of `result`, recall that our survey design statistics are also part of this dimension.

Example 7 — Table of regression results using survey data 3

Next, we omit the base levels for the factor variables with `collect style showbase`. We would like to include the labels for the factor variables and the labels for their levels in a single column. So we stack the row headers with `collect style row`, adding a blank line between the stacked row dimensions. By default, when we stack the row headers, the factor variables will be bound to their levels by an equal sign. We specify the `nobinder` option so that the levels will be stacked under the factor-variable labels, instead of bound by an equal sign.

```
. collect style showbase off
. collect style row stack, nobinder spacer
```

Finally, we format the regression results to display only two digits to the right of the decimal. We do not need to apply this format to the survey design information, so we specify `colname#result`, which includes only the regression results. Additionally, we remove the border that is displayed on the right by default by setting the border pattern to `nil`. Then, we preview our finalized table:

```
. collect style cell colname#result, nformat(%5.2f)
. collect style cell border_block, border(right, pattern(nil))
. collect preview
```

	Model for systolic BP
Body mass index (BMI)	1.31
Age group	
30–39	0.96
40–49	6.40
50–59	14.87
60–69	21.62
70+	27.90
Sex	
Female	-5.62
Race	
Black	1.40
Other	-0.31
Intercept	88.59
Number of strata	31
Number of sampled PSU	62

Reference

McDowell, A., A. Engel, J. T. Massey, and K. Maurer. 1981. "Plan and operation of the Second National Health and Nutrition Examination Survey, 1976–1980". In *Vital and Health Statistics*, ser. 1, no. 15. Hyattsville, MD: National Center for Health Statistics.

Also see

[TABLES] **collect style header** — Collection styles for hiding and showing header components

[TABLES] **collect style row** — Collection styles for row headers