

# DATA Step

## Can I count instances of values stored in a group of variables in an observation?

As with most things in SAS, the simple answer is "Yes!".

A combination of functions allow you to create a concatenated string from all of the responses, then count instances of the desired value(s).

Use the following code to create some sample data:

```
data responses ;
  input (q1-q5) ($1.)
  ;
datalines ;
yYyNN
NNNNN
YYNYn
;
run ;
```

The results dataset contains character strings: y | Y | n | N and we wish to count instances of 'y' and 'n' discounting case differences. In the following code the CATS function concatenates and strips off any leading and trailing blanks. Use the OF keyword to specify a range of variables and the COLON MODIFIER to identify a name-range. The COUNT function then reads along the concatenated string and counts instances of the desired string; the 'i' modifier causes the system to ignore case.

```
data results ;
  set responses ;
  count_true  = count(cats(of q:), 'y', 'i') ;
  count_false = count(cats(of q:), 'n', 'i') ;
run ;
```

Output:

q1	q2	q3	q4	q5	count_true	count_false
y	Y	y	N	N	3	2
N	N	N	N	N	0	5
Y	Y	N	Y	n	3	2

**Mixed character / numeric responses**

# DATA Step

If however the data is in mixed forms - containing both character and numeric responses:

```
*** Mixed character / numeric responses *** ;
data responses1 ;
    input (q1-q5) ($1.)
           (q6-q10) ( 1.)
    ;
datalines ;
yYyNN00110
NNNNN00000
YYNYn11010
;
run ;
```

then it is still possible to use the same syntax to build a concatenated string; the CATS function simply converts the numeric variables to character strings, while suppressing the normal 'conversion' NOTE in the LOG.

Because the data string now contains multiple possible outcomes: 1 | y | Y | 0 | n | N then the COUNT function cannot be used as it searches for a character 'string'. Use the COUNTC function however, and we can now search for any item on a character 'list':

```
data results1 ;
    set responses1 ;
    count_true  = countc(cats(of q:), 'y1', 'i') ;
    count_false = countc(cats(of q:), 'n0', 'i') ;
run ;
```

Output:

q1	q2	q3	q4	q5	q6	q7	q8	q9	q10	count_true	count_false
y	Y	y	N	N		0	0	1	1	0	5
N	N	N	N	N		0	0	0	0	0	10
Y	Y	N	Y	n		1	1	0	1	0	4

Unique solution ID: #1031

Author: Alan D Rudland

Last update: 2017-08-31 15:30