

## DATA Step

### Can I count instances of values with mixed response types stored in a group of variables in an observation?

Counting simple boolean Y | N or 1 | 0 responses is [fairly straightforward in SAS](#), however where the responses are more complex some additional logic is required. Consider the data generated by the following DATALINES:

```
datalines ;
yYyNN00110True FALSE False
NNNNN00000false false flase
YYNYN11010False TRUE true
;
```

In addition to the boolean outcomes, there are also text-string responses, with mixed case values. It is possible to read these values from a raw data file by generating a bespoke informat:

```
proc format ;
    invalue $bool 'true'   = '1'
                  'false'  = '0'
                  other    = ' '
;
run ;
```

This however requires an exact character match against the raw data. Rather than endeavour to match all possible outcomes, it is simpler to perform some data cleansing on the INPUT BUFFER. We can do this by applying a character transformation to the automatic `_infile_` variable (which contains the INPUT BUFFER). The INPUT BUFFER (and the `_infile_` variable) is only populated when the system encounters an INPUT statement - use the single trailing `@` to load a line of raw data before performing the transformation:

```
input @ ;
_infile_ = lowercase(_infile_) ;
```

The informat can then be used to read in the cleansed raw data in a second INPUT statement:

```
input (q1-q5)          ($1.)
      (a1-a5)          ( 1.)
      (resp1-resp3)    (: $bool5.)
;
```

# DATA Step

The stored values of resp1-resp3 are now 1 | 0 and if desired a user-defined format can be applied to these data values to make them render as true | false.

When acted upon by the CATS function, it is the 'underlying' values which will be concatenated. In this example the variables have different names and a name range cannot be specified, instead the `_ALL_` keyword has been used.

```
proc format ;
  invalue $bool 'true'  = '1'
                'false' = '0'
                other   = ' '
;
run ;

proc format ;
  value $bool '1'    = 'true'
              '0'    = 'false'
              other = ' '
;
run ;

data responses2 ;
  input @ ;
  _infile_ = lowercase(_infile_) ;
  input (q1-q5) ($1.)
        (a1-a5) ( 1.)
  (resp1-resp3) (: $bool5.)
;
  format resp1-resp3 $bool. ;
datalines ;
yYyNN00110True FALSE False
NNNNN00000false false flase
YYNYN11010False TRUE true
;
run ;

data results2 ;
  set responses2 ;
  count_true  = countc(cats(of _all_), 'y1', 'i') ;
  count_false = countc(cats(of _all_), 'n0', 'i') ;
run ;
```

Unique solution ID: #1032

Author: Alan D Rudland

Last update: 2017-08-31 16:28