The SAS INT2WAY Macro

Ellen Hertzmark and Donna Spiegelman

October 4, 2010

Abstract

The %INT2WAY macro is a SAS macro that constructs all the 2-way interactions among a set of variables. It also makes a global macro variable that lists the new variables.

Keywords: SAS, macro, 2-way interactions

Contents

1 Description **2**

2 Invocation and Details **2**

3 Examples **3**

3.1 Example 1. Using %INT2WAY on indicators with individual names 3

3.2 Example 2. Making interactions of indicators made from categorical variables by %INDIC3 4

3.3 Example 3. Interactions of a categorical variable with a set of indicators 5

4 WARNINGS **6**

5 Credits **7**

1 Description

%INT2WAY is a SAS macro that takes a list of variable names and makes all the 2-way interactions among them, as well as a global macro variable listing all the interaction variables, so the user is not burdened with excessive typing (and chances to make errors). It runs inside a DATA step. %INT2WAY is primarily intended to create all the 2-way interactions of a large list of variables, as well as the macro variable listing all the interaction variables, to be used in an automatic selection procedure. This version has two new parameters that allow multiple uses of %INT2WAY in a program without overwriting any ordinary or macro variables.

2 Invocation and Details

To call %INT2WAY, your program must know where to look for it. The most efficient way is to include

```
options mautosource sasautos='/usr/local/channing/sasautos';
```

at the beginning of your program.

%INT2WAY has one positional parameter (which must come first) and two keyword parameters (which both have defaults, so can be omitted if you wish).

Inside a data step, and after all the variables (main effects) have been created, type

```
%int2way( v1 v2 v3 v4 v5........vn  (the list of variables for which you
          want interactions),
  vblpre= int (The name of the prefix you want for the interaction variables),
  mvblname = _iv_ (the prefix for the name of the macro variable that holds
                   the list of interaction variable names)
);
```

The list `v1.....vn` is the list of `n` variables for which you want all the 2-way interactions. It must be written out fully so the macro can count the number of items. The macro variables created by %INDIC3 can be used, but notation such as `x1-x6` cannot. The variable names for the interaction variables will be of the form *vblpre2_1 vblpre3_1.....vblpre(n)_(n-1)*, where *vblpre* is the parameter giving the prefix for the names of the interaction variables. It will be `int` if you do not use the parameter *VBLPRE* in your macro call. Note that the numbering is *vblpre(higher)_(lower)*. Each interaction variable will be labelled as the product of the 2 original variables.

3 Examples

In the examples, we will use 3 categorical variables `bmiq`, `protq`, `vitaq`, representing quintiles of bmi, protein intake, and vitamin A intake, respectively. We used %INDIC3 on these variables to make sets of indicators. The %INDIC3 call for `bmiq` had USEMISS=1, and the other two had USEMISS=0. The reference levels were 3, 1, and 5, respectively.

3.1 Example 1. Using %INT2WAY on indicators with individual names

```
data one;  set one ;
%int2way(bmiq1 bmiq5 protq2 protq3 protq4);
run;
```

In this example, we just gave the macro the list of variables. The interaction variables have prefix `'int'` and the macro variable listing all the 2-way interactions is called `IVLIST`.
Note that we needed to write out the variable names for the macro, and we could not use the shorthand `protq2-protq4`. The global macro variable &_IVLIST_ is

```
int2_1 int3_1 int3_2 int4_1 int4_2 int4_3 int5_1 int5_2 int5_3 int5_4
```

The part of the `PROC CONTENTS` showing the interaction variables is

```
20    int2_1      Num       8    bmiq5 * bmiq1
21    int3_1      Num       8    protq2 * bmiq1
22    int3_2      Num       8    protq2 * bmiq5
23    int4_1      Num       8    protq3 * bmiq1
24    int4_2      Num       8    protq3 * bmiq5
25    int4_3      Num       8    protq3 * protq2
26    int5_1      Num       8    protq4 * bmiq1
27    int5_2      Num       8    protq4 * bmiq5
28    int5_3      Num       8    protq4 * protq2
29    int5_4      Num       8    protq4 * protq3
```

Note that the interactions `int2_1`, `int4_3`, `int5_3`, `int5_4` are all identically zero, since the product of 2 different indicators made from the same categorical variable is always zero. The macro labelled `int2_1` as `bmiq5*bmiq1`.

3.2 Example 2. Making interactions of indicators made from categorical variables by %INDIC3

We made the 2-way interactions using %INT2WAY using the optional keyword parameters.

```
%int2way(&bmiq_ &protq_ &vitaq_, vblpre=all, mvblname=all);
```

The macro, like other procedures and macros, treats the global macro variables made by %INDIC3 as if they were written out completely.

The global macro variable &_ALLLIST_ has 78 variable names.

```
all2_1 all3_1 all3_2 all4_1 all4_2 all4_3 all5_1 all5_2 all5_3 all5_4 all6_1
all6_2 all6_3 all6_4 all6_5 all7_1 all7_2 all7_3 all7_4 all7_5 all7_6 all8_1
all8_2 all8_3 all8_4 all8_5 all8_6 all8_7 all9_1 all9_2 all9_3 all9_4 all9_5
all9_6 all9_7 all9_8 all10_1 all10_2 all10_3 all10_4 all10_5 all10_6 all10_7
all10_8 all10_9 all11_1 all11_2 all11_3 all11_4 all11_5 all11_6 all11_7
all11_8 all11_9 all11_10 all12_1 all12_2 all12_3 all12_4 all12_5 all12_6
all12_7 all12_8 all12_9 all12_10 all12_11 all13_1 all13_2 all13_3 all13_4
all13_5 all13_6 all13_7 all13_8 all13_9 all13_10 all13_11 all13_12
```

The indices go up to 13 because there are 5 variables in `&bmiq_` (because USEMISS=1) and 4 each in `&vitaq_` and `&protq_` (because USEMISS=0).

In this example `int2_1` means `bmiq2*bmiq1`.

A partial PROC CONTENTS for the dataset is

```
86    all13_1     Num       8    vitaq4 * bmiq1
87    all13_2     Num       8    vitaq4 * bmiq2
88    all13_3     Num       8    vitaq4 * bmiq4
89    all13_4     Num       8    vitaq4 * bmiq5
90    all13_5     Num       8    vitaq4 * bmiqm
91    all13_6     Num       8    vitaq4 * protq2
92    all13_7     Num       8    vitaq4 * protq3
93    all13_8     Num       8    vitaq4 * protq4
94    all13_9     Num       8    vitaq4 * protq5
95    all13_10    Num       8    vitaq4 * vitaq1
96    all13_11    Num       8    vitaq4 * vitaq2
97    all13_12    Num       8    vitaq4 * vitaq3
20    all2_1      Num       8    bmiq2 * bmiq1
21    all3_1      Num       8    bmiq4 * bmiq1
22    all3_2      Num       8    bmiq4 * bmiq2
23    all4_1      Num       8    bmiq5 * bmiq1
24    all4_2      Num       8    bmiq5 * bmiq2
25    all4_3      Num       8    bmiq5 * bmiq4
26    all5_1      Num       8    bmiqm * bmiq1
27    all5_2      Num       8    bmiqm * bmiq2
28    all5_3      Num       8    bmiqm * bmiq4
29    all5_4      Num       8    bmiqm * bmiq5
```

As pointed out before, many of the interaction variables will be identically zero, because they are interactions of 2 indicators made from the same categorical variable. For the primary intended use of %INT2WAY, this is not a problem, since these variables will not be selected by any procedure.

3.3 Example 3. Interactions of a categorical variable with a set of indicators

```
%int2way(bmiq &protq_);
```

The global macro variable &_IVLIST_ is

```
INT2_1 INT3_1 INT3_2 INT4_1 INT4_2 INT4_3 INT5_1 INT5_2 INT5_3 INT5_4
```

The part of the `PROC CONTENTS` containing the interaction variables is

```
20    int2_1      Num       8    protq2 * bmiq
21    int3_1      Num       8    protq3 * bmiq
22    int3_2      Num       8    protq3 * protq2
23    int4_1      Num       8    protq4 * bmiq
24    int4_2      Num       8    protq4 * protq2
25    int4_3      Num       8    protq4 * protq3
26    int5_1      Num       8    protq5 * bmiq
27    int5_2      Num       8    protq5 * protq2
28    int5_3      Num       8    protq5 * protq3
29    int5_4      Num       8    protq5 * protq4
```

In Example 1, `int2_1` means `bmiq5*bmiq1`. In Example 3, `int2_1` means `protq1*bmiq`. The meaning of a variable name `int(n)_(m)` made by the macro is completely dependent on the macro call, as is meaning of the global macro variable &_IVLIST_. That is why it is advisable to use the keyword parameters *VBLPRE* and *MVBLNAME* when you have more than one call to %INT2WAY in a program.

4 WARNINGS

1. The meaning of a variable name of the form `int(n)_(m)` varies depending on the dataset and the order the variables were listed in the call to %INT2WAY.

Similarly, the global macro variable &_IVLIST_ always refers to the results of the latest call to %INT2WAY.

When making multiple calls to %INT2WAY in a program, it is advisable to use the optional parameters *VBLPRE* and *MVBLNAME*.

2. Using %INT2WAY on sets of indicators made from categorical variables leads to a large number of interaction variables that are identically zero.

While these may slow down an automatic selection procedure, they do no harm to the ultimate analytic result.

5 Credits

Written by Ellen Hertzmark and Donna Spiegelman for the Channing Laboratory. Questions can be directed to Ellen Hertzmark,
`stleh@channing.harvard.edu`, (617) 432-4597.