

# 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

<b>1</b>	<b>Description</b>	<b>2</b>
<b>2</b>	<b>Invocation and Details</b>	<b>2</b>
<b>3</b>	<b>Examples</b>	<b>3</b>
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
<b>4</b>	<b>WARNINGS</b>	<b>6</b>

## 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.

```
all12_1 all13_1 all13_2 all14_1 all14_2 all14_3 all15_1 all15_2 all15_3 all15_4 all16_1
all16_2 all16_3 all16_4 all16_5 all17_1 all17_2 all17_3 all17_4 all17_5 all17_6 all18_1
all18_2 all18_3 all18_4 all18_5 all18_6 all18_7 all19_1 all19_2 all19_3 all19_4 all19_5
all19_6 all19_7 all19_8 all110_1 all110_2 all110_3 all110_4 all110_5 all110_6 all110_7
all110_8 all110_9 all111_1 all111_2 all111_3 all111_4 all111_5 all111_6 all111_7
all111_8 all111_9 all111_10 all112_1 all112_2 all112_3 all112_4 all112_5 all112_6
all112_7 all112_8 all112_9 all112_10 all112_11 all113_1 all113_2 all113_3 all113_4
all113_5 all113_6 all113_7 all113_8 all113_9 all113_10 all113_11 all113_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	<code>all13_1</code>	Num	8	<code>vitaq4 * bmiq1</code>
87	<code>all13_2</code>	Num	8	<code>vitaq4 * bmiq2</code>
88	<code>all13_3</code>	Num	8	<code>vitaq4 * bmiq4</code>
89	<code>all13_4</code>	Num	8	<code>vitaq4 * bmiq5</code>
90	<code>all13_5</code>	Num	8	<code>vitaq4 * bmiqm</code>
91	<code>all13_6</code>	Num	8	<code>vitaq4 * protq2</code>
92	<code>all13_7</code>	Num	8	<code>vitaq4 * protq3</code>
93	<code>all13_8</code>	Num	8	<code>vitaq4 * protq4</code>
94	<code>all13_9</code>	Num	8	<code>vitaq4 * protq5</code>
95	<code>all13_10</code>	Num	8	<code>vitaq4 * vitaq1</code>
96	<code>all13_11</code>	Num	8	<code>vitaq4 * vitaq2</code>
97	<code>all13_12</code>	Num	8	<code>vitaq4 * vitaq3</code>
20	<code>all2_1</code>	Num	8	<code>bmiq2 * bmiq1</code>
21	<code>all3_1</code>	Num	8	<code>bmiq4 * bmiq1</code>
22	<code>all3_2</code>	Num	8	<code>bmiq4 * bmiq2</code>
23	<code>all4_1</code>	Num	8	<code>bmiq5 * bmiq1</code>
24	<code>all4_2</code>	Num	8	<code>bmiq5 * bmiq2</code>
25	<code>all4_3</code>	Num	8	<code>bmiq5 * bmiq4</code>
26	<code>all5_1</code>	Num	8	<code>bmiqm * bmiq1</code>
27	<code>all5_2</code>	Num	8	<code>bmiqm * bmiq2</code>
28	<code>all5_3</code>	Num	8	<code>bmiqm * bmiq4</code>
29	<code>all5_4</code>	Num	8	<code>bmiqm * bmiq5</code>

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](mailto:stleh@channing.harvard.edu), (617) 432-4597.