



In this issue

- [Executive Update](#)
- [Tips & Tricks:](#)
 - [Smart DOS Batch Files](#)
 - [Modifying and Removing All Labels with PROC DATASET](#)
 - [The PVALUEX. Format](#)
 - [Tricky Operator <>](#)

Who We Are

We, the Vancouver SAS Users Group (VanSUG), are an informal group of SAS users in the Vancouver area who wish to periodically meet and share their knowledge. There are no fees to join – we simply meet twice annually (spring and fall) to have a half day of presentations, networking, and fun! Keep informed of our activities at www.sas.com/vancouver.

Newsletter Inputs

This newsletter gives information for/by the local Vancouver SAS community. If you would like to provide input to a future newsletter, please email us at vansug@gmail.com. We would love to hear from you!

Executive Team

President-elect:

Mei Chong
(BC Centre for Disease Control)

Vice President-elect:

Colleen McGahan
(BC Cancer Agency)

Other Members:

Nate Derby (Stakana Analytics)
Peter Hruby (HSBC)

Executive Update

Colleen McGahan: vansug@gmail.com

Blah blah blah

TIPS & TRICKS: Smart DOS Batch Files

Matthew Yip: myip@intrawest.com

You can do more with X than just run DOS commands. DOS .bat files can be built to accept SAS macro variables. Variables in DOS take the form of %[number], where number corresponds to the order of the variable being passed. For example, this is the base code used to automate zipping and password protecting a file being sent out:

```
%LET zdrive = H;  
%LET zpath = \Zip_Files;  
%LET zfile = em_list;  
%LET zsource = H:\Source_File\em_list.csv;  
%LET zpwd = password;
```

```
X "c:\batch\zip.bat &zdrive. &zpath. &zfile. &zsource. &zipwd.";
```

This is the contents of the zip.bat file:

```
::Zip  
  
%1:  
cd %2  
  
"C:\Program Files\WinZip\WZZIP.exe" %3.zip %4 -s"%5"  
EXIT
```

This resolves in DOS to:

```
H:  
cd \Zip_Files  
"C:\Program Files\WinZip\WZZIP.exe" em_list.zip  
H:\Source_File\em_list.csv -s"password"
```

TIPS & TRICKS: Modifying and Removing All Labels with PROC DATASET

Christina Chu: cchu@bccancer.bc.ca

Sometimes after merging multiple datasets, some variables will have identical label, and this may lead to confusion. The following code will let you change the labels and/or remove all labels on the variables in the dataset.

```
* Modify labels ;
PROC DATASETS LIB=<sas_library> MEMTYPE=data;
  MODIFY <sas_dataset>;
  LABEL <variable_name1> ='<new_label1>'
        <variable_name2> ='<new_label2>';
RUN;

* Remove all labels ;
PROC DATASETS LIB=<sas_library> MEMTYPE=data;
  MODIFY <sas_dataset>;
  ATTRIB _all_ label=' ';
RUN;
```

TIPS & TRICKS: The PVALUEX. Format

Ryan Woods: rwoods@bccancer.bc.ca

In programming table output, numeric formats on *p*-values that are very small (e.g. 0.0000125) result in the problem that the displayed *p*-value rounds to 0; for example, for the data set:

variable	group_a	group_b	p_value
x	15.7	16.4	0.0000125
y	9.2	9.6	0.2517214

applying a format of 6.4 to the *p*-value column to round *p*-values to 4 decimals yields the following:

variable	group_a	group_b	p_value
x	15.7	16.4	0.0000
y	9.2	9.6	0.2517

A similar result occurs if you ask SAS to create a new variable which is the *p*-value rounded to the 4th decimal using the ROUND() function. A better approach is to make use of the PVALUEX. formats SAS has available. For example, applying the format PVALUE6. to the *p*-value column would yield the following table:

variable	group_a	group_b	p_value
x	15.7	16.4	<.0001
y	9.2	9.6	0.2517

Looking at last meeting's evaluation forms, 35% of you did not know about the VanSUG website. It provides information on

- The next meeting, including a link to register for it.
- Highlights from the last meeting
- An archive of all the presentations we have had since 2004
- An archive of the newsletter since its onset
- A link to SAS training courses scheduled in Vancouver
- Information about the Executive Team and how to get more involved

Check it out at sas.com/vancouver!

TIPS & TRICKS: Tricky Operator <>

Weiwei Du, wdu@shaw.ca

<> is a tricky operator in SAS. Depending on where you use it, there are two entirely different interpretations. In PROC SQL, it means "not equal to":

```
PROC SQL;
  SELECT *
  FROM [TABLE]
  WHERE X <> Y;
QUIT;
```

The above query returns rows for which the variables X and Y are not equal. However, in DATA steps, X<>Y means MAX(X, Y), which returns the larger number of the two!

```
IF X<>Y THEN DELETE;
```

You may think the above statement means "if X is not equal to Y, delete the row." Wrong. It's actually interpreted as "If the larger of X and Y is non-zero, then delete the row." If X=3 and Y=3, then X<>Y returns 3, which is non-zero, so this row will be deleted.

Our next meeting will be on **November 17, 2010**. Check out the VanSUG website at sas.com/vancouver later for more information!

The **2010 Joint Statistical Meetings (JSM)** conference will be held right here in Vancouver on July 31 - August 5. This is the largest statistical conference in North America. Information is at www.amstat.org/meetings/jsm/2010.

SAS is holding its **M2010 Data Mining Conference** on October 25-26 in Las Vegas, Nevada. Information can be found at sas.com/m2010.

The **2010 Western Users of SAS Software (WUSS) Conference** will be held in San Diego on November 3-5. Come on down! Information can be found at wuss.org.