



In this issue

- [Executive Update](#)
- [Tips & Tricks:](#)
 - [PROC SGPLOT](#)
 - [? and ?? Format Modifiers](#)
 - [Use IN: to Code Character Comparisons](#)
 - [TYPES Statement](#)

Who We Are

The Vancouver SAS Users Group (VanSUG) is an informal group of SAS users in the Vancouver area who meet and share their knowledge. There are no fees – we simply meet every spring and fall to have presentations, networking, and fun! Check us out at vansug.ca.

Newsletter Inputs

This newsletter is for and by the local Vancouver SAS community. If you'd like to contribute to a future newsletter, please email us at vansug@gmail.com!

Executive Team

President:

Colleen McGahan
(BC Cancer)

Vice President:

Suzanne Humphreys
(Rick Hansen Institute)

Other Members:

- Núria Chapinal
(BC Centre for Disease Control)
- Mei Chong
(BC Centre for Disease Control)
- Jing Dong
(BC Hydro)
- Nate Derby
(Stakana Analytics)
- Jeremy Hamm
(BC Cancer)

Executive Update

Welcome to another Vancouver SAS Users Group meeting! As always, we're eager to hear from any of you who would like to present or write a newsletter article for our next meeting. Even if you've never given a presentation or written an article before, please write to us at vansug@gmail.com. We would be happy to hear from you.

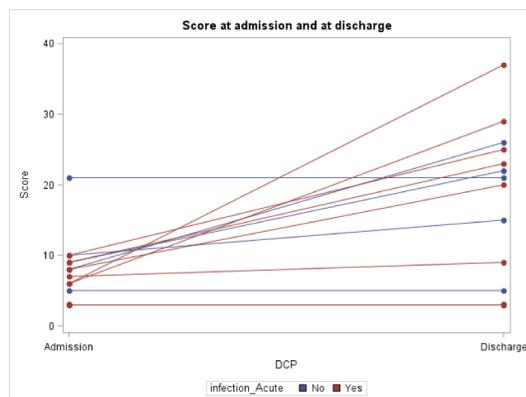
TIPS & TRICKS: PROC SGPLOT: Creating Spaghetti Plots by Another Covariate

Tian Shen: tshen@rickhanseninstitute.org

When plotting time series data, we usually plot individual patient's scores over time, but sometimes it is also interesting to stratify by another covariate, for example treatment or control group. The plotted graph will show different coloured lines to indicate which patients are in one group and which patients are in another.

In the example below, `GROUPLC` specifies that variable 'var2' determines the line colour for each patient. `GROUPMC` specifies the marker colours and `GROUPMS` specifies the marker symbols. By having these options, patients within the same group according to the 'var2' variable will have the same line colour, marker colour and marker symbol. The equally important statement option `NAME` assigns a name to the plot statement, which will be used in the `KEYLEGEND` statement later. If we do not assign a name, then the `SGPLOT` statement won't plot the legend for the variable 'var2'. The graph below plots each individual patient's score at admission to and discharge from hospital, and the colour of the line indicates whether or not the patient is in the infection='Yes' group or 'No' group.

```
PROC SGPLOT DATA=example2;
  STYLEATTRS DATASYMBOLS=(circlefilled circlefilled);
  SERIES X=DCP Y=score / GROUP=enrollmentidn
         GROUPLC=var2 GROUPMC=var2 GROUPMS=var2
         LINEATTRS=(pattern=solid) markers NAME='groups';
  YAXIS LABEL='Score';
  KEYLEGEND 'groups' /TYPE=linecolor TITLE='infection_Acute';
  TITLE 'Score at admission and at discharge';
RUN;
```



TIPS & TRICKS: ? and ?? Format Modifiers

Suzanne Humphreys: shumphreys@rickhanseninstitute.org

? and ?? can be used as format modifiers to suppress notes to the log for invalid values when using formats. This will be useful if you have some unknown values in a character date field but you don't need a message written to the log each time an invalid value is discovered. For example, a date is incorrectly entered into a character field as 20/20/2018. When we try to convert the character date into a numeric date using a SAS date format, an invalid argument note is written to the log.

```
DATA test1;
  cdate1 = '20/20/2018';
  ndate1 = INPUT(cdate1,ddmmyy10.);
  FORMAT ndate1 ddmmyy10.;
  PUT ndate1=;
RUN;
```

```
SAS Log:
note: Invalid argument to function input at line 896 column 9.
ndate1=
cdate1=20/20/2018 ndate1= _ERROR_=1 _N_=1
```

? can be used as a modifier to suppress the invalid argument message.

```
ndate1 = INPUT(cdate1,?ddmmyy10.);

SAS log:
ndate1=
cdate1=20/20/2018 ndate1= _ERROR_=1 _N_=1
```

?? will also prevent the automatic _ERROR_ message from being set to 1.

```
ndate1 = INPUT(cdate1,??ddmmyy10.);

SAS log:
ndate1=
```

TIPS & TRICKS: Using IN: to Code Character Comparisons with Criteria Having Different Lengths

Mei Chong: mei.chong@jcbrothers.com

When you evaluate a condition on a character variable based on different starting strings in length, instead of specifying each starting string using `substrn`, you can use `IN:`. For example, if we have to select the records for all the invoices which begin with 'A1' or 'B23' or 'C', we can use the `substrn` function to truncate the values of the variable:

```
DATA select;
  SET invoice_data;
  WHERE substrn(invoice,1,2) = 'A1'
  OR substrn(invoice,1,3) = 'B23'
  OR substrn(invoice,1,1) = 'C';
RUN;
```

Alternatively, we could improve this code by using `IN:`.

```
DATA select;
  SET invoice_data;
  WHERE invoice IN: ('A1', 'B23', 'C');
RUN;
```

TIPS & TRICKS: The TYPES Statement in PROC MEANS

Brendan Bakos:

Brendan.Bakos@bccancer.bc.ca

By default, `PROC MEANS` summarizes all combinations of variables specified in the `CLASS` statement. However, the `TYPES` statement can be used to specify the combination of the variables required rather than all combinations being output. For example, the program below will provide the sum of X for A alone and B alone, as well as A by C, A by B, and B by C.

```
PROC MEANS DATA=mydat SUM;
  CLASS A B C;
  VAR X;
  OUTPUT OUT=mydat_summed SUM=;
  TYPES A B A*(B C) B*C;
RUN;
```

Our next user group meeting will be in **November 7th, 2018**. Check out the VanSUG website at vansug.ca later for more information!

The archived presentations and newsletters, as well as a link to scheduled SAS training courses held in Vancouver, can all be found on the VanSUG website at vansug.ca!

The **SAS Global Forum 2019** will be held in Dallas, TX on April 28 - May 1, 2019. Information can be found at sasglobalforum.com.

The **2018 Western Users of SAS Software (WUSS) Conference** will be held in Sacramento, California on September 5-7, 2017. Information can be found at wuss.org.