The VanSUG Newsletter

In this issue

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 Agency)

Vice President:
  Suzanne Humphreys
    (Rick Hansen Institute)

Other Members:
  Núria Chapinal
    (Perinatal Services BC)
  Mei Chong
    (BC Centre for Disease Control)
  Jing Dong
    (BC Hydro)
  Nate Derby
    (Stakana Analytics)
  Jeremy Hamm
    (BC Cancer Agency)

Executive Update

10 Year Celebration!

It has been 10 years since users took over the running of the Vancouver SAS user group, now known as VanSUG. Over the years VanSUG has grown considerably from less than 30 attendees to now sometimes over 100 attendees, we saw the introduction of the newsletter, which I'm pleased to say is still going strong, and for a few years we had the additional Business Analytics session. I have been on the Executive for those 10 years and I personally want to thank you all for your contributions over the years. Many of you have contributed in making it a success, whether by presenting, submitting a newsletter article, being on the Executive or just by attending our sessions and supporting the user group community. Without you, we would not have a meeting! I look forward to many more smooth running and fun meetings ahead of us.

Colleen, President of VanSUG

TIPS & TRICKS: Working with Large Datasets

Colleen McGahan: cmcgahan@bccancer.bc.ca

Below are just a few of many simple steps you can take to make working with large datasets easier and more efficient:

- Use the subsetting `IF` statement to only keep observations of interest.

- Some variables have a greater length to them than what is really required. For example, an indicator variable of just 0 and 1 does not require the SAS default length of 8. You can reduce the length of variables using the `LENGTH` statement. It can be surprising the significant effect this can have.

- Only keep variables that are required. You can do this using the `KEEP=` or `DROP=` option.

- Having lots of extremely large temporary datasets in the `WORK` directory can take up a lot of space and can hinder running programs, particularly if you need the space for `PROC SORT`. Try to limit the number of `DATA` steps when programming and consolidate `DATA` steps, making them more efficient.

- Try to avoid sorting these large datasets - as a rough guide; SAS requires up to 4 times the size of the dataset you are sorting. Consider using `PROC SQL` for merging instead or turn a coded key variable into a format.

- Develop and test your programs using `OBS=0` - this can be used to check your program for syntax error. Once you have verified your syntax, use a smaller random sample of the dataset using the `RANUNI` function to check your code.

Tips & Tricks: Custom Boxplot using `SGPLOT`

Lovedeep Gondara: Lovedeep.Gondara@bccancer.bc.ca

Customizing whiskers on grouped boxplot using `PROC SGPLOT` with SAS 9.3 or earlier, `PROC SGPLOT` offers a `VBOX` statement for creating boxplots. However, prior to release of SAS 9.4, user cannot specify the length of whiskers as per requirement. Following syntax can be used to overlay different plots with results same as `VBOX` and user defined whisker length.

1. Create summary statistics needed using `PROC UNIVARIATE`. i.e. below we are creating a dataset with 10th, 25th, 50th, 75th and 90th percentile that will be used to define our boxplot in next step.

```
proc univariate data = datasetname noprint;
    class    varname1;
    by       varname2;
    var      varname3;
    output   out = outstats
             pctlpts = 10 25 50 75 90
             pctlpre = diff_
             pctlname = p10 p25 p50 p75 p90;
run;
```

2. Use `PROC SGPLOT` to overlay different plots. Specifically, we are overlaying three highlow plots, first one is just a line from 10th to 90th percentile, second and third are upper and lower parts of our box respectively.

```
proc sgplot data = outstats nocycleattrs;
    highlow   x = varname2 high = diff_p90 low = diff_p10 /
              group = varname1 groupdisplay=cluster
              grouporder = ascending clusterwidth = 0.7;
    highlow   x = varname2 high = diff_p75 low = diff_p50 /
              group = varname1 type = bar groupdisplay = cluster
              grouporder = ascending clusterwidth = 0.7
              barwidth = 0.7 name = 'Legend';
    highlow   x = varname2 high = diff_p50 low = diff_p25 /
              group = varname1 type = bar groupdisplay = cluster
              grouporder = ascending clusterwidth = 0.7
              barwidth = 0.7;
    keylegend 'Legend';
    yaxis     grid;
    xaxis     display=(nolabel);
run;
```

Obtaining VIF, AIC and BIC in `PROC REG`

Dilinuer Kuerban: dkuerban@rickhanseninstitute.org

While both `PROC GLM` and `PROC REG` can be used for linear regression modeling purposes, using `PROC REG` can be advantageous in certain cases. In particular, when we want to obtain variance inflation factor (VIF) for multicollinearity check and AIC/BIC for model selection, `PROC REG` seems to be the only option. We can easily specify VIF, AIC and BIC options in the model statement to get these three quantities in `PROC REG`. A further note is that AIC and BIC need to be output using the `OUTEST=` option as they will not be printed directly in the results window as VIF does. A simple example follows:

```
proc reg data = data1 outest = out1;
    model y = x1 x2 x3 x4 x5 / vif aic bic;
run;

proc print data = out1;
    var _aic_ _bic_;
run;
```

Tips & Tricks: `PROC FORMAT`

Jing Dong: jing.dong@bchydro.com

The `PROC FORMAT` enables you to define your own formats for variables. For example, if you want to define your phone number format to be (xxx) xxx-xxx.

```
data phones1;
    input pn;
    datalines;
    6041234567
    7781234567
;
run;

proc format;
    picture phone_f
     (default = 16)
     low-high = '999) 999-9999'
     (prefix = '(');
run;

data phones2;
    set phones1;
    pn_f = put(pn, phone_f.);
run;
```

Obs	pn	pn_f
1	6041234567	(604) 123-4567
2	7781234567	(778) 123-4567

Our next user group meeting will be on **November 30th, 2016**. 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 2017** will be held in Orlando, FL on April 2-5, 2017. Information can be found at sasglobalforum.com.

The **Western Users of SAS Software (WUSS) Conference 2016** will be held in San Francisco, California on September 7-9, 2016. Information can be found at wuss.org.