VIEWS
International SAS Programmer Community

News, Hints, Tips and Information for SAS® Users

Issue 55, 4th Quarter 2011

It is now 2012, so this has changed from being the November issue to the December issue, and finally the January issue! It will still be the usual newsletter issue filled with SAS news, hints and tips that I hope you will find interesting. In this issue of *VIEWS News* I have articles from some of our regular contributors: LeRoy Bessler, David Shannon, Murphy Choy, Peter Crawford, Sanjay Matange and Sunil Gupta, with articles on charting long trends, solid-state disks and SAS, piecewise regression, calculating working days, a new blog about ODS Graphics and an introduction to SAS Savvy. Finally, my regular Formats, Options, and Functions section has some more new features in the new release, SAS 9.3.

If you would like to contribute an article, to re-visit and improve an existing article, or just discuss the possibility of doing so, please feel free to send an email to me at **newsletter@views-uk.org**. A list of subject suggestions for your articles can be found on the *VIEWS* web site, and anyone is very welcome to add to that list by sending emails to **newsletter@views-uk.org** with your own questions.

Philip R Holland (Newsletter Editor)

Membership of VIEWS is ***free*** *and you can register for notification of all future VIEWS events by emailing us at* ***membership@views-uk.org****, making certain you include your name and postal address. If you would like to receive email notification whenever a new issue of the free quarterly VIEWS News is published on the web site, please remember to also include your email address, and please inform us when you change it.*

Did You Know?

If you have a useful hint or tip, please send it to the Editor, and share it with the VIEWS membership.

Another Way To Chart a Very Long Trend

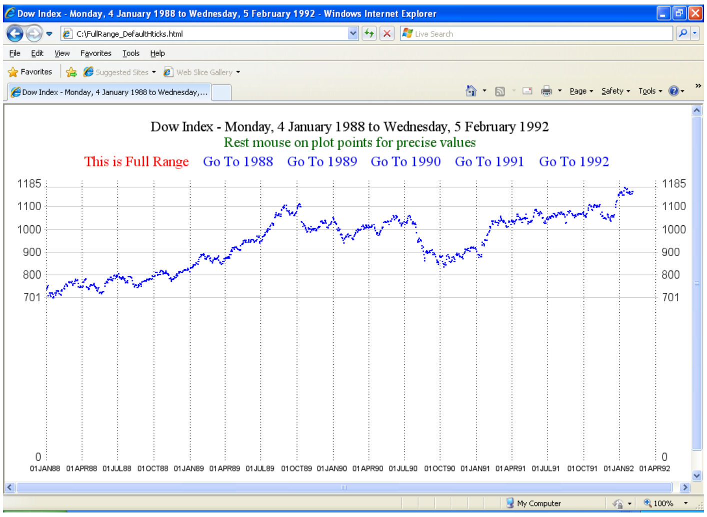
A very long trend plot has many data points. If you want horizontal tick values at reasonable intervals, readability is difficult to achieve. This article is a follow-up to "How To Chart a Very Long Trend" in *VIEWS News* 54. As emphasized there, any trend chart is best delivered with HTML since the viewer can then access pop-up text for precise knowledge of the y-value at any x-value, without needing to rely on estimates based on y-axis tick mark values.

I am using data set SASHELP.CITIDAY, so the reader can run the code as an exercise, display the results at monitor size, and then experiment further with the code if desired.

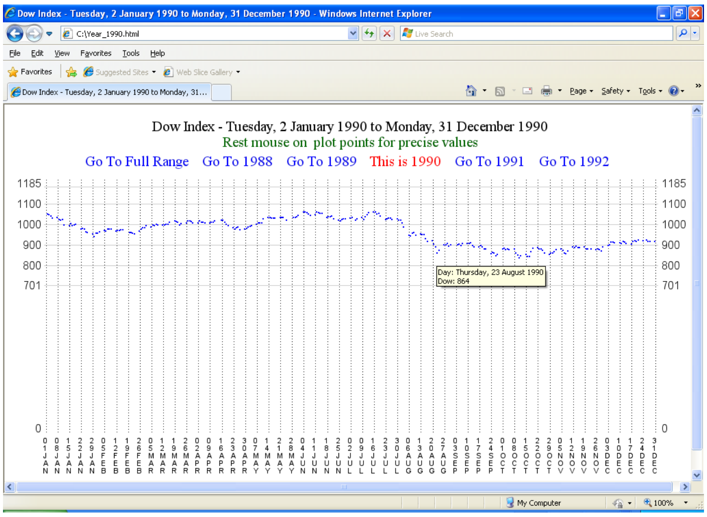
SASHELP.CITIDAY covers 1034 market trading days at the New York Stock Exchange, spanning 4+ years. Since providing an x-axis tick mark value for every trading day is infeasible, last time I focused on coming up with an optimal alternative way to present the full range with fewer, but readable, tick mark values.

Another solution is to break up the full period into segments, in this case separate years (the last one being incomplete). For ways to present multiple years, my articles in *VIEWS News* 52 and 53 (Issue 53 co-authored with Alexandra Riley) show several ways to do that, either with overlays or with an array of panels. There the data was for a year of months, but a collection of years can be handled analogously.

This time I instead want to present an interlinked set of single-year web pages, which are also linked both ways with this picture of the full range:


Dow Index - Monday, 4 January 1988 to Wednesday, 5 February 1992 - Windows Internet Explorer
C:\FullRange_DefaultHticks.html
Live Search
File Edit View Favorites Tools Help
Favorites
Suggested Sites
Web Slice Gallery
Dow Index - Monday, 4 January 1988 to Wednesday,...
Page Safety Tools
Dow Index - Monday, 4 January 1988 to Wednesday, 5 February 1992
Rest mouse on plot points for precise values
This is Full Range
Go To 1988
Go To 1989
Go To 1990
Go To 1991
Go To 1992
1185
1100
1000
900
800
701
0
01JAN88 01APR88 01JUL88 01OCT88 01JAN89 01APR89 01JUL89 01OCT89 01JAN90 01APR90 01JUL90 01OCT90 01JAN91 01APR91 01JUL91 01OCT91 01JAN92 01APR92
My Computer
100%

After selecting "Go To Year 1990", and resting the mouse on a plot point marker:


Dow Index - Tuesday, 2 January 1990 to Monday, 31 December 1990 - Windows Internet Explorer
C:\Year_1990.html
Live Search
File Edit View Favorites Tools Help
Favorites
Suggested Sites
Web Slice Gallery
Dow Index - Tuesday, 2 January 1990 to Monday, 31...
Page Safety Tools
Dow Index - Tuesday, 2 January 1990 to Monday, 31 December 1990
Rest mouse on plot points for precise values
Go To Full Range
Go To 1988
Go To 1989
This is 1990
Go To 1991
Go To 1992
1185
1100
1000
900
800
701
0
Day: Thursday, 23 August 1990
Dow: 864
01JAN 08JAN 15JAN 22JAN 29JAN 05FEB 12FEB 19FEB 26FEB 05MAR 12MAR 19MAR 26MAR 02APR 09APR 16APR 23APR 30APR 07MAY 14MAY 21MAY 28MAY 04JUN 11JUN 18JUN 25JUN 02JUL 09JUL 16JUL 23JUL 30JUL 06AUG 13AUG 20AUG 27AUG 03SEP 10SEP 17SEP 24SEP 01OCT 08OCT 15OCT 22OCT 29OCT 05NOV 12NOV 19NOV 26NOV 03DEC 10DEC 17DEC 24DEC 31DEC
My Computer
100%

Here is the code used to create the web pages:

```
%let Path = C:\;

DATA WORK.DowByDay;
  LENGTH HTMLvar $512 Year $4;
  SET sashelp.citiday
    (KEEP=date snydjcm WHERE=(snydjcm NE .));
  snydjcm = ROUND(snydjcm,1);
  HTMLvar = "alt='"
      || "Day: "
      || TRIM(LEFT(PUT(Date, WEEKDATX32.)))
      || '0D'X
      || "Dow: "
      || TRIM(LEFT(PUT(snydjcm, 4.)))
      || "'";
  Year = YEAR(date);
RUN;

PROC MEANS DATA=WORK.DowByDay MIN MAX NOPRINT;
  VAR snydjcm date;
  OUTPUT OUT=WORK.MinMax
         MIN=MinY StartDate MAX=MaxY EndDate;
RUN;

DATA _NULL_;
  SET MinMax;
  CALL SYMPUT('MinY', TRIM(LEFT(MinY)));
  CALL SYMPUT('MaxY', TRIM(LEFT(MaxY)));
  CALL SYMPUT('DateRange',
    TRIM(LEFT(PUT(StartDate, WEEKDATX32.)))
    || ' to ' ||
    TRIM(LEFT(PUT(EndDate, WEEKDATX32.))));
RUN;

PROC SUMMARY DATA=WORK.DowByDay NWAY;
  CLASS Year;
  VAR date;
  OUTPUT OUT=WORK.EachYearWithStartDateEndDate
         MIN=StartDate MAX=EndDate;
RUN;

DATA _NULL_;
  SET WORK.EachYearWithStartDateEndDate
      END=LastYear;
  CALL SYMPUT('DateRange' || TRIM(LEFT(_N_)),
    TRIM(LEFT(PUT(StartDate, WEEKDATX32.)))
    || ' to ' ||
    TRIM(LEFT(PUT(EndDate, WEEKDATX32.))));
  CALL SYMPUT('Year' || TRIM(LEFT(_N_)),
    TRIM(LEFT(Year)));
  IF LastYear;
  CALL SYMPUT('NumberOfYears',TRIM(LEFT(_N_)));
RUN;

DATA _NULL_;
  SET WORK.EachYearWithStartDateEndDate
      END=LastYear;
  * Want to start on a Monday
    not later than the first day of data *;
  StartDate = MDY(1, 1, Year);
  DayOfWeekStart = WEEKDAY(StartDate);
  IF DayOfWeekStart GT 2
    THEN StartDate =
           StartDate - (DayOfWeekStart - 2);
    ELSE IF DayOfWeekStart EQ 1
      THEN StartDate = StartDate - 6;
  CALL SYMPUT('StartDateTick'
              || TRIM(LEFT(_N_)), StartDate);
  * Want to end on a Monday
    not earlier than the last day of data *;
  EndDate = MDY(12,31,Year);
  DayOfWeekEnd = WEEKDAY(EndDate);
```

```
  IF DayOfWeekEnd GT 2
    THEN EndDate = EndDate + (9-DayOfWeekEnd);
    ELSE IF DayOfWeekEnd EQ 1
      THEN  EndDate = EndDate + 1;
  CALL SYMPUT('EndDateTick' || TRIM(LEFT(_N_)),
    EndDate);
RUN;

%MACRO CreateYearLinks;
  %DO j = 1 %TO &NumberOfYears. %BY 1;
    COLOR=BLUE LINK="Year_&&Year&j...html"
    "  Go To &&Year&j.  "
  %END;
%MEND CreateYearLinks;

PROC CATALOG CAT=WORK.gseg KILL;
RUN;
QUIT;

ODS NORESULTS; /* do not open in SAS session */
ODS LISTING CLOSE;

  /* START SetUp that applies to all graphs */
GOPTIONS RESET=ALL;
GOPTIONS FTEXT="Arial"; /* for axis values */
GOPTIONS XPIXELS=993 YPIXELS=528;

SYMBOL1 COLOR=BLUE HEIGHT=0.5 FONT=ZAPF
  VALUE='E2'X;
SYMBOL2 COLOR=BLUE HEIGHT=0.5 FONT=ZAPF
  VALUE='E2'X;

TITLE1 HEIGHT=1 PCT ' ';

AXIS1 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0
  ORDER = 0 TO 1200 BY 100
  VALUE=(HEIGHT=3PCT '0' '' '' '' '' '' ''
    "&MinY." '800' '900' '1000' '1100'
    "&MaxY.");
AXIS2 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0
  ORDER = 0 TO 1200 BY 100
  VALUE=(JUSTIFY=LEFT
         HEIGHT=3PCT '0' '' '' '' '' '' ''
    "&MinY." '800' '900' '1000' '1100'
    "&MaxY.");

  /* END SetUp that applies to all graphs */
AXIS3 LABEL=NONE MAJOR=NONE MINOR=NONE STYLE=0
  VALUE=(HEIGHT=2PCT);

TITLE2 FONT='Times New Roman' HEIGHT=4PCT
  COLOR=BLACK "Dow Index - &DateRange."
  JUSTIFY=CENTER COLOR=CX006600
  "Rest mouse on plot points for precise
   values";
TITLE3 HEIGHT=1PCT ' ';
TITLE4 HEIGHT=4PCT FONT='Times New Roman'
  COLOR=RED LINK="FullRange_DefaultHticks.html"
  "This is Full Range  "
  %CreateYearLinks
  ;

ODS HTML PATH="&Path." (URL=NONE)
  STYLE=Minimal GTITLE GFOOTNOTE
  BODY="FullRange_DefaultHticks.html"
  (TITLE="Dow Index - &DateRange.");

PROC GPLOT DATA=WORK.DowByDay;
  PLOT snydjcm * date /
    HTML=HTMLvar NAME="atZERO" DESCRIPTION=' '
    VAXIS=AXIS1 CVREF=CXCCCCCC LVREF=1
    VREF = (&MinY. 800 900 1000 1100 &MaxY.)
    HAXIS=AXIS3 AUTOHREF CHREF=BLACK LHREF=33;
  PLOT2 snydjcm*date / VAXIS=AXIS2;
  FORMAT date DATE7.;
RUN;
QUIT;

ODS HTML CLOSE;

%MACRO LinkedYearlyTrendCharts;

  %DO i = 1 %TO &NumberOfYears. %BY 1;

    AXIS3 LABEL=NONE MAJOR=NONE MINOR=NONE
      STYLE=0 VALUE=(HEIGHT=2PCT)
      ORDER =
   &&StartDateTick&i. to &&EndDateTick&i. by 7;

    TITLE2 FONT='Times New Roman' HEIGHT=4PCT
      COLOR=BLACK "Dow Index - &&DateRange&i."
      JUSTIFY=CENTER COLOR=CX006600
      "Rest mouse on plot points for precise
       values";
    TITLE3 HEIGHT=1PCT ' ';
    TITLE4 HEIGHT=4PCT FONT='Times New Roman'
       COLOR=BLUE
       LINK="FullRange_DefaultHticks.html"
       "Go To Full Range  "
    %DO j = 1 %TO &NumberOfYears. %BY 1;
      %IF &&Year&j. NE &&Year&i. %THEN %DO;
       COLOR=BLUE LINK="Year_&&Year&j...html"
       "  Go To &&Year&j.  "
      %END;
      %ELSE %DO;
       COLOR=RED LINK="Year_&&Year&j...html"
       "  This is &&Year&j.  "
      %END;
    %END;
       ;

    ODS HTML PATH="&Path." (URL=NONE)
      STYLE=Minimal GTITLE GFOOTNOTE
      BODY="Year_&&Year&i...html"
      (TITLE="Dow Index - &&DateRange&i.");

    PROC GPLOT DATA=WORK.DowByDay
     (WHERE=(Year EQ "&&Year&i."));
      PLOT snydjcm * date /
        HTML=HTMLvar NAME="atZERO&i."
        DESCRIPTION=' '
        VAXIS=AXIS1 CVREF=CXCCCCCC LVREF=1
        VREF = (&MinY. 800 900 1000 1100
                &MaxY.)
        HAXIS=AXIS3 AUTOHREF CHREF=BLACK
        LHREF=33;
      PLOT2 snydjcm * date / VAXIS=AXIS2;
      FORMAT date DATE5.;
    RUN;
    QUIT;

    ODS HTML CLOSE;

  %END;

%MEND LinkedYearlyTrendCharts;

%LinkedYearlyTrendCharts;

ODS LISTING;
```

NOTE: If curious, please see Issue 54 for why `FONT=ZAPF VALUE='E2'X` is recommended for the SYMBOL statements.

LeRoy Bessler

Are you on Solids yet?

Before the start of this year I could count on one hand the number of our customers who used solid state disks (SSD) in their SAS® environment. This has started to change and it is changing rapidly.

Usually we store our SAS installation, the operating system, the temporary data created during processing (WORK library), permanent data and other types of SAS files on *mechanical* disks. Although there are various methods of interfacing with mechanical disks such as SATA, SAS (not to be confused with the software), generally the faster they spin the faster they serve up data to your computer or servers' CPU and memory for processing.

SAS Software is highly I/O dependent, i.e. it reads-In and writes-Out lots of data. That strategy for processing data is very clever (ultimately it allows SAS to process an almost infinite amount of data). It just takes a while sometimes. In the last decade advances in CPU and memory technology have pretty much kept pace with the growing volumes of data we deal with. Disk technology has not, basically because there is a limit to how fast you can physically spin a disk. Until recently solid state disks were not able to read and write with equal speed, usually reading very quickly but writing relatively slowly. The devices themselves were expensive too. However in the last year or so solid state drives have come onto the market with equal read and write performance. This suits SAS and by using such drives, for instance, as WORK libraries we could have a profound benefit on long-running SAS jobs.

Amadeus regularly receives requests for help speeding up SAS processing. A few weeks ago, along with one of our customers, we set about putting some new solid state disks through their paces. A large variety of tests were performed on two types of SSD along with a traditional mechanical disk as the control. The bottom line? One type of SSD saw an average 60% reduction in the time required for SAS jobs on the Windows 2008 R2 x64 operating system.

We work with a range of technologies and appliances for large volume data processing, from SPD Server to in-database processing, but we were amazed with these results. Better still with costs for high volume, high specification SSDs plunging, you can expect the return on investment in SSDs to be recognised in a short space of time.

Andrew Williams, Consultant at Amadeus Software, published his findings on SAS processing with solid state technologies. You can read more and download the paper from **www.amadeus.co.uk/ssd**. Before you know it, you'll be weaned onto solids too.

David Shannon, Amadeus Software

Ask The Consultant

This part of VIEWS News is where you can get your technical questions answered. Send your questions to the Editor.

Calculating working days

Q: How do you calculate working days?

A:**[SAS 9.2]** Working days = Weekday interval - holidays

Now available in SAS9.2, the option INTERVALDS allows you to define your own intervals and the online-doc shows a weekday demonstration among the INTNX examples at **is.gd/1JN5aX.**

The code below was proposed in the SAS Discussion Forums at **is.gd/sNUFcv**:

```
LIBNAME my (WORK);

%LET tod = 31-Jan-2012;
%LET frd =
  %SYSFUNC(INTNX(WEEKDAY,"&tod."D,-1000),
           DATE9);

DATA my.hols;
  FORMAT date WEEKDATE.;
  INPUT date DATE9.;
  LIST;
  CARDS;
27Jan11
26jan11
2jan12
1jan1960
;
RUN

DATA span;
  FORMAT begin end WEEKDATE.;
  KEEP begin end;
  * build the diary: holidays first;
  ARRAY dates(%SYSEVALF("&frd."D):
              %SYSEVALF("&tod."D))
        $1 _TEMPORARY_;
  DO WHILE(NOT eof1);
    SET my.hols END=eof1;
    WHERE date BETWEEN "&frd."D AND "&tod."D;
    dates(date) ='H';
  END;
  * now mark weekends;
  * the first Saturday;
  s1 = "&frd."D + 7 - WEEKDAY("&frd."D);
  DO date = s1 TO "&tod."D BY 7;
    * Saturday to Saturday;
    dates(date) = 'W';
    * Sunday;
    IF "&tod."D > date THEN dates(date+1)='W';
  END;
  * now build custom intervals;
  DO date = "&frd."D TO "&tod."D;
    IF dates(date) = ' ' THEN DO;
      IF NOT MISSING(oldDay) THEN DO;
        begin = oldDay;
        end = date - 1;
        OUTPUT;
        oldDay = date;
      END;
      ELSE oldDay = COALESCE(oldDay, date-1);
    END;
  END;
  * check if my diary finished on non-working
    day;
  IF oldDay NE . THEN DO;
    begin = oldDay;
    end = "&tod."D;
    OUTPUT;
  END;
  STOP;
RUN;
```

```
OPTION INTERVALDS = (myWorkingDay = span);

%MACRO test_my_working_days(date=&SYSDATE.
                           ,for=2
                           );
  %LOCAL D1 last_wd ;
  %LET D1=%SYSFUNC(PUTN("&date."d,WEEKDATE));

  %* .. .. not making the next macro var LOCAL
     allows it to update this macro variable
     in the calling environment;

  %LET last_wd_ex_hols =
     %SYSFUNC(INTNX(myWorkingDay, "&date."D,
                    -&for.), WEEKDATE);
  %LET last_wd =
     %SYSFUNC(INTNX(WEEKDAY, "&date."D,
                    -&for.), WEEKDATE);
  %PUT 2 WEEKdays before &D1. will be &last_wd.
       but excluding also dates in my holiday
       calendar the date would be
       &last_wd_ex_hols.;
%MEND  test_my_working_days;

%SYMDEL last_wd_ex_hols;

%test_my_working_days(date=31-Jan-2011);

%test_my_working_days(date=3-Jan-2012);

* show the macro var has not been created
  in the calling environmenmt ;
%put &last_wd_ex_hols.;

* now create that name in the calling env;
%LET last_wd_ex_hols = &SYSDATE.;

%test_my_working_days(date=29-Dec-2011);

* and show value updated by macro;
%put &last_wd_ex_hols.;
```

***Peter Crawford**, Crawford Software Consultancy*

Piecewise Regression using SAS

Q: Is there a way to perform linear regression on data with a change point?

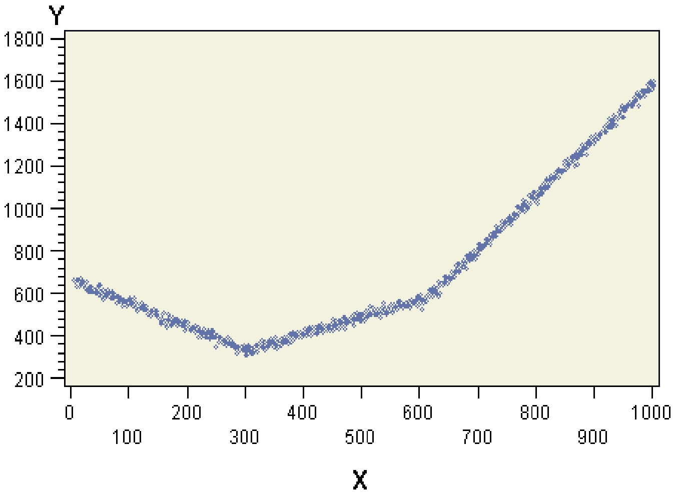
A: Piecewise regression, also known as segmented regression, is an approach that is adopted in situations where the data seems to have a change point. As this is rarely seen, the approach of defining the variables is not commonly known to modellers. In this paper, we will keep the focus on the creation of dummy variables and how they can be used to create piecewise regression

Piecewise regression is a regression approach which allows users to create linear regression models for several segments in the data. Typically, most modellers will use the BY statement in conjunction with class variables to achieve segmented regression. However, this approach typically produces multiple regression equations which will then be applied on different segments depending on the break point. In many situations, this is undesirable and users may request for just a single equation. In this case, piecewise regression using a dummy-variable approach would be the most suitable way to solve the problem.

To better understand the background of the data, let us first use a simple step which generates 3 segments of data. Here is the code for this step.

```
/*Piecewise data creation*/
DATA lineardata;
  /*Loop to create data*/
  DO i = 1 TO 1000;
    /*First equation*/
    IF i LE 300 THEN DO;
      x = i;
      y = (-1.1 * x + 670) +
          (SQRT(200) * RANNOR(12345));
    END;
    /*Second equation*/
    ELSE IF i LE 600 THEN DO;
      x = i;
      y = (0.8 * x + 100) +
          (SQRT(200) * RANNOR(12345));
    END;
    /*Second equation*/
    ELSE DO;
      x = i;
      y = (2.6 * x - 1000) +
          (SQRT(200) * RANNOR(12345));
    END;
    /*Output the information*/
    OUTPUT;
  END;
  DROP i;
RUN;
```

As you can observe from the above, the breakpoints occur at X = 300 and X = 600. There are 3 equations that are present in the data above. Thus, we will expect to see 3 different lines in the scatter plot of Y against X below.



From the graph, we can see that the lines are connected in such a way that might otherwise indicate a quadratic trend. To use the segment-regression approach, we have to first create dummy variables which possess the right qualities for creating the model.

```
/*Pre-processing the data*/
DATA lineardata;
  SET lineardata;
  IF x LE 300 THEN x1 = 0;
             ELSE x1 = 1;
  IF x LE 600 THEN x2 = 0;
             ELSE x2 = 1;
  s1 = (x - 300) * x1;
  s2 = (x - 600) * x2;
RUN;
```

The dummy variables are created to increase the influence of the variables that denote where the change points occur. To achieve this effect, we first create the binary variables X1 and X2 to signify the change points. Once created, we create 2 additional variables to signify the effect added onto the original variable. This particular approach will allow us to merge the effect into a single regression equation. One important point to note is that this method of creating dummy variables is just one example of a myriad of methods.

Once the modelling data has been created, we can then use it to model the data to produce a piecewise regression. Below is the program to run the piecewise regression.

```
PROC REG DATA = lineardata;
  MODEL y = x s1 s2;
RUN;
QUIT;
```

From the regression output, the table below shows the parameter estimates of the regression model. The variable X indicates the original equation slope of -1.1. The additional variables of S1 and S2 adds to the base value of X to form the equation slope of 0.8 and 2.6. Although the intercept is only 667, the additional values from S1 and S2 will add to the effect, thus increasing the values.

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	667.46504	1.61833	412.44	<.0001
X	1	-1.08277	0.00775	-139.71	<.0001
S1	1	1.83351	0.01244	147.41	<.0001
S2	1	1.81906	0.01016	179.05	<.0001

Piecewise regression is an excellent modelling approach should the data have several segments which are in need of differential modelling. It is both simple to implement and to interpret.

Murphy Choy, University College Dublin

Formats, Options, and Functions

This section is devoted to the description of useful, or unusual, SAS Formats, Options, and Functions.

[SAS 9.3] The **GETOPTION()** function now allows the **STARTUPVALUE** option, which returns the system option value that was used to start SAS, either on the command line, or in a configuration file.

[SAS 9.3] The default ODS destination for interactive SAS in Windows or UNIX is now **HTML**, while in batch SAS, and in z/OS, it remains as **LISTING**. **ODS GRAPHICS** is switched on by default in interactive SAS.

[SAS 9.3] The **B8601DJ*w*.** informat reads a Java date and time value that is in the form “yyyymmddhhmmssffffff”.

PhUSE News

2011 PhUSE Conference in Brighton

PhUSE hosted their 7th conference at the Hilton Metropole in Brighton in October, which was attended by over 375 people. The conference was fortuitous to have two amazing keynote speakers: Chuck Cooper from the FDA who spoke about “Modernisation of the Product Review Process in CDER: Current and Future Efforts”, and Professor Lord Winston who spoke about “Improvements in Medical Care: Have We Lost Our Way?” At this successful conference, chaired by Yvonne Moores, there were over 100 presentations, several discussion clubs and 23 exhibitors. You can read more about the conference on the PhUSE website **www.phuse.eu** and PhUSE members will be able to view papers and presentations. Next year's conference will be held in Hungary in Budapest on 14-17 October 2012.

Registration has opened for the FDA/PhUSE Annual Computational Science Symposium which will be held on 19-20 March 2012 at Silver Spring Civic Center, Silver Spring, Maryland, USA. The conference theme is "Update on Standards, Tools, and Process Initiatives Across Regulatory Review: Collaboration with Key Working Groups to Improve the Product Lifecycle". For further details please see the PhUSE website, **www.phuse.com/css.aspx**.

PhUSE Committee

News

A Shameless Plug for Graphically Speaking

ODS Graphics is mainstream. With the release of ODS Graphics with SAS 9.2, creating analytical graphs in SAS is easy. With SAS 9.3, ODS Graphics is included in Base and does not require a separate SAS/GRAPH license.

ODS Graphics means different things to different people and provides different sets of tools appropriate for different audiences.

For the Analyst:

You can get graphs from the SAS procedures automatically by switching on the ODS Graphics feature. Just include the statement “**ODS GRAPHICS ON;**” in the code prior to submitting the procedure statements. Graphs are automatically created by the procedures and included in your output along with the data tables in the right order. You can edit these graphs by using the ODS Graphics Editor. No additional knowledge of graph coding is required, freeing you to focus on the analysis.

For the Graph User:

You can create analytical graphs without knowing a shred of graph coding by using the ODS Graphics Designer. Designer is an interactive application with a point-and-click approach to creating graphs. Designer generates the GTL code needed to create the graphs which you can view as you go. Submit the code “**%SGDESIGN();**” in the program editor to start Designer. Graphs created using Designer can also be run in batch with different data.

For the Graph Programmer:

You can create custom analytical graphs from the raw data or from the results of your analysis by using the Statistical Graphics (SG) procedures. These procedures provide an easy and concise syntax to create the commonly used graphs, and generate GTL syntax under the covers.

For the Advanced Graph Programmer:

You can use the Graph Template Language (GTL) directly to create your custom graphs. This is the same code used by SAS procedure writers to create the automatic graphs from the procedures.

As users migrate to SAS 9.2 and 9.3, there is an increased need to provide more information about these features. To support this, Graphically Speaking, a new SAS blog on Data Visualization with a focus on ODS Graphics has been started. Our goal is to write articles on different features of the software and to engage with the audience in a lively conversation.

We plan to write in these categories:

- **Feature du jour** – In depth discussion of specific features in GTL, SG procedures.
- **Clinical Graphs** – Full examples for commonly used clinical graphs.
- **Tips and Tricks** – Ways to use GTL and SG procedures that are not obvious.

Please visit our blog and provide your feedback. I appreciate this opportunity to publicize this blog through *VIEWS*.

The Graphically Speaking blog can be found at **blogs.sas.com/content/graphicallyspeaking**.

Sanjay Matange, SAS Institute

SAS Savvy: 4 Clicks away from SAS Answers

Are you ready for the next generation of focused online SAS resource and training with over 600 links to the best SAS answers from two of the most popular SAS search engines, **support.sas.com** and **www.lexjansen.com** as well as others? Each dedicated page on a SAS Procedure or SAS Topic not only contains links to beginner, advanced and HOW SAS papers, but also tips, diagrams and summary guides to dramatically increase your productivity. The Common SAS Programming FAQ page is vital to prevent 're-inventing the wheel'. Sunil also shares his great collection of over 20 invited presentations and workshops. Finally, those in the pharmaceutical industry can greatly benefit from Sunil's in-depth knowledge and experience.

I hope you will take a few minutes to explore how SAS Savvy at **www.sassavvy.com** can be used as an online resource to directly sharpen your SAS skills and enhance your career. For my daily SAS consulting projects, I actively search SAS Savvy to find answers when I need them in an organized and central location. You too can save hours of SAS programming time and have this same level of access to SAS Savvy.

For a limited time and while supplies last, I am happy to offer *VIEWS* readers a free one-month trial subscription.

Sunil Gupta, Gupta Programming

In Brief

- Back issues of *VIEWS News* are available from the *VIEWS* web site, and also from the *VIEWS News* page at **www.sasCommunity.org/wiki/VIEWS_News**

- Andrew Ratcliffe has a blog at **notecolon.info**.
- More SAS hints and tips can be found on the SAS FAQ pages **www.hollandnumerics.com/sasfaq**, on the Amadeus Software Tips pages at **is.gd/0VMRHy** and in their free podcasts at **is.gd/xDNTHG**.
- There is a social networking group for SAS users on Ning called **SAS Professionals**, originally intended for SAS users in the UK, at **www.sasprofessionals.net**, but currently totalling over 6,200 members worldwide.
- SAS users who use Twitter can now join a "twibe" at **twibes.com/SAS**, which filters your tweets down to those containing "SAS", "EnterpriseGuide" or "PROC".
- Phil Holland has started a group page on Squidoo for other pages related to SAS, and would like to make it a central resource for SAS hints, tips, links and books at **www.squidoo.com/groups/SAS_users**
- Do you have an Android or webOS phone or tablet? There are now some SAS-related ebook apps available for them, with more details at **www.squidoo.com/android-apps-list** and **www.squidoo.com/howsas-webos-support.**
- Phil Holland has also created the SAS Contractors group on LinkedIn specifically for independent SAS consultants working in the UK and Europe. Requests to join the group should be made at:
  www.linkedin.com/groups?gid=2481512
  and the associated SAS Contractor Resources page, which is open to all, can be found at:
  www.squidoo.com/sas_contractors
  He also manages the SAS Professional Forum group on LinkedIn, which currently has over 13,500 members. Everyone is welcome to join, but discussions are restricted to SAS-related topics:
  www.linkedin.com/groups?gid=70702

Diary

Are you organising an event that would be of interest to the VIEWS readership? Let us know as we are interested in all non-profit making events related to SAS.

January 2012

25 **Using SAS to Create Microsoft Excel Reports**, webinar

See the Amadeus web site **www.amadeus.co.uk/webinars** for further details.

March 2012

22 **Integrate and Cleanse your Data with SAS Enterprise Data Integration**, webinar

See the Amadeus web site **www.amadeus.co.uk/webinars** for further details.

April 2012

22-25 **SAS Global Forum**, Orlando, FL, USA

The SAS Global Forum (SGF, formerly called SUGI) is held each year and focuses on the technical aspects of SAS software. See **www.sasglobalforum.org** for more details.

30 **Descriptive Statistics with Base SAS**, webinar

See the Amadeus web site **www.amadeus.co.uk/webinars** for further details.

May 2012

24 **Using SAS to Create Microsoft Excel Reports**, webinar

See the Amadeus web site **www.amadeus.co.uk/webinars** for further details.

30 **Business Intelligence in your Pocket with SAS Mobile BI**, webinar

See the Amadeus web site **www.amadeus.co.uk/webinars** for further details.

October 2012

14-17 **PhUSE Conference**, Budapest, Hungary

The Pharmaceutical Users Software Exchange (PhUSE) conference is the premier programming event for the pharmaceutical industry in Europe. See **www.phuse.eu** for more details.

All VIEWS event information will be posted and/or emailed to registered members of VIEWS, and will include an event application form. Please send your queries about any VIEWS events to **event@views-uk.org***, and don't forget to look at the web site for the latest news.*

Contacts

LeRoy Bessler	Email: **le_roy_bessler@wi.rr.com**
Murphy Choy University College Dublin	Email: **goladin@gmail.com**
Peter Crawford Crawford Software Consultancy	Email: **peter.crawford@blueyonder.co.uk**
Sunil Gupta Gupta Programming	Email: **sunil@guptaprogramming.com** Web: **www.sassavvy.com**
Sanjay Matange SAS Institute	Email: **sanjay.matange@sas.com**
David Shannon Amadeus Software	Email: **david.shannon@amadeus.co.uk** Web: **www.amadeus.co.uk**
PhUSE	Email: **office@phuse.eu** Web: **www.phuse.eu**
SAS UK	Tel: +44-(0)1628-486933 Web: **www.sas.com/uk**

VIEWS

Web Site	Web: **www.views-uk.org**
Chairman	Email: **chair@views-uk.org**
Newsletter Editor	Email: **newsletter@views-uk.org**
Membership secretary	Email: **membership@views-uk.org**

VIEWS News *is published quarterly by VIEWS International SAS Programmer Community. VIEWS is a not-for-profit organisation. Our Mission Statement and Rules can be viewed on our web site together with a list of current committee members.*