Paper 090-2007

Presentation Quality Forecast Visualization with SAS/GRAPH®

Samuel T. Croker, ALTEK Information Technology Inc., Frederick, MD

ABSTRACT

A statistical forecast is useless without sharp, attractive and informative graphics to present it. SAS/GRAPH® can produce superior results without resorting to exporting the data and drawing in Microsoft Excel. This case study demonstrates this simplicity using a few macros that make it possible to generate production quality forecast, confidence interval and other forecast-specific visualizations for appropriate SAS procedures. Methods for dealing with odd date intervals and usefully displaying difficult datetime axis values are explored. The ANNOTATE facility is used to help with displaying model and additional date information. A method for displaying confidence intervals as regions is shown as well as a way to produce separate regions for different types of data. Use of SYMBOL, AXIS and LEGEND statements is discussed along with details for using fonts, colors and lines to make the graphics look magazine worthy.

INTRODUCTION

There is a lot of information that should be contained in a forecast graph, and it can be difficult to represent this information in an un-cluttered and meaningful way. This case study covers the following forecast graphic elements:

- Drawing attention to different data situations such as one-step-ahead or fit regions, holdout data compared with forecasts and the actual forecast regions by using shaded or colored confidence regions
- Representing compound date-time axis tick values that include several orders of magnitude such as day-hour-minute increments
- Presenting the point forecast compared with known actual values in a concise fashion

FUNCTIONAL ELEMENTS

The following SAS/GRAPH statements are used extensively along with the GPLOT procedure:

- SYMBOL
- AXIS
- LEGEND
- PROC GPLOT options such as OVERLAY, HREF, VREF, GRID and many others

Other options and facilities used:

- The ANNOTATE facility to help display information that might be confusing if put directly on an axis
- Date, Time and Datetime formats and functions to alter the display of and process date values
- SAS graphic devices such as ACTIVEX

A TRANSFER FUNCTION FORECAST EXAMPLE

SETTING THE STAGE

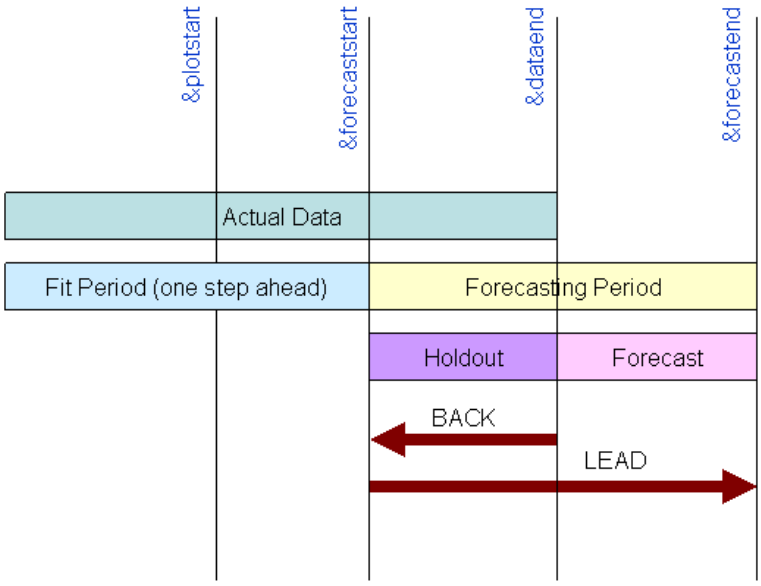
An adaptation of the well known Box and Jenkins Series J example is used to demonstrate a few techniques that are helpful. Series J does not have a datetime stamp, so an arbitrary one is added using the following data step statement:

```
timeid='01JAN2007/00:00:00'dt + (_n_*3600);
```

GATHERING INFORMATION ABOUT THE DATA AND FORECASTS

The forecasts are produced using SAS sample code (see References). Even though the ARIMA procedure is used for this example, the graphical method shown works equally well with any SAS module that produces forecasts and confidence intervals over time.

A few macro variables are used to control the plotting and to break up the confidence intervals into regions. These parameters are easily changed to alter the display of the graph. These macro variables are related to the forecasting procedure through the BACK and LEAD settings of the ARIMA procedure's FORECAST statement for this case. Many SAS forecasting techniques use the BACK and LEAD options and the general concept is shown below:



BUILDING THE PLOT DATA

In this example, BACK was set to be prior to the end of the observed data in order to compare the forecasting with the actual data. Three confidence interval regions will be generated: one for the one step ahead forecasts (the fit region), one for the hold out data and one for the forecasts themselves. Earlier the BACK and LEAD options were set to control how the FORECAST statement of PROC ARIMA will generate forecasts. These values, along with the number of rows of the original data, were used to calculate the time id where hold out forecasting and complete forecasting begins.

```
data out(  drop=       sval0 sval1 sval2)
       low0( keep=date sval0 sval1 sval2)
       high0(keep=date sval0 sval1 sval2)
       low1( keep=date sval0 sval1 sval2)
       high1(keep=date sval0 sval1 sval2)
       low2( keep=date sval0 sval1 sval2)
       high2(keep=date sval0 sval1 sval2);
       set &ds;
       where date>=&plotstart;
       output out;
       if date <= &forecaststart then do;
              sval0=lower; output low0;
              sval0=upper; output high0;
       end;
       if date > &forecaststart and date <= &dataend then do;
              sval1=lower; output low1;
              sval1=upper; output high1;
       end;
       if date > &dataend then do;
              sval2=lower; output low2;
              sval2=upper; output high2;
       end;
run;
```

This data step results in seven data sets that are manipulated individually and then put back together to create the plot. Each pair of low and high datasets contains the same variable name for both lower and upper confidence band. The reason for this is that the INTERPOL=MS option of the SYMBOL statement will plot the values as a polygon. The lower bounds are sorted in descending order while the upper remain in the original order (should be ascending)

so that the upper and lower bounds can be drawn in the same way that you would draw any polygon without removing your pen from the paper. It is a “connect the dots” scenario and if they are not sorted in this way a mess is generated (Figure 2).

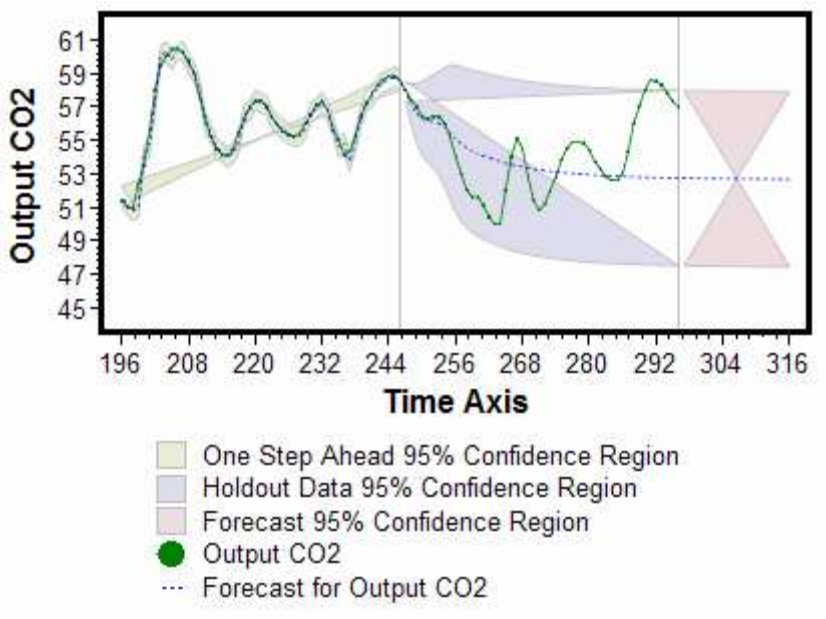
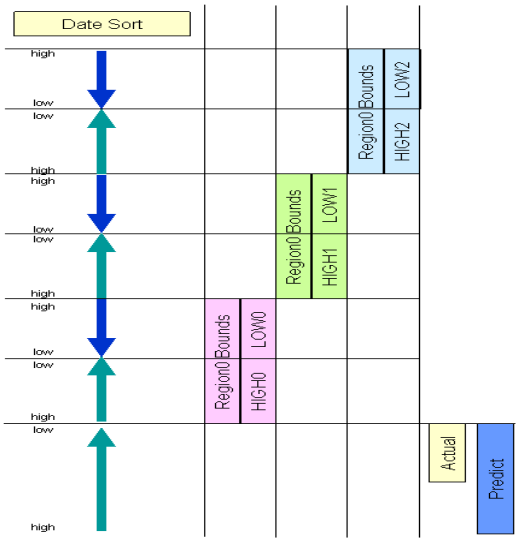


Figure 2

To prevent this, the lower bounds are first sorted in descending order:

```
proc sort data=low0; by descending date;
run;
proc sort data=low1; by descending date;
run;
proc sort data=low2; by descending date;
run;

data forecast;
        set  low2 high2
        low1 high1
        low0 high0
        out;
        if date=. then delete;
run;
```



Then the data are put back together by stacking each dataset on top of each other. Missing values occur due to stacking these datasets, but this is actually what is needed to get to the final plot. Each low-high dataset pair constitutes a closed polygon. The chart to the right shows the way in which the high and low and the actual, datasets are sorted and how they are stacked to form the dataset that is plotted.

SETTING UP THE SAS/GRAPH OPTIONS

The SYMBOL statement's INTERPOL option is the mechanism for generating the confidence regions. The MS INTERPOL option generates solid areas. Controlling the axis is also necessary in most cases, and this is done using the AXIS statement. It is often necessary to specify the minimum, maximum and increment values for the date time axis using the ORDER option. The graphic options usually need to be reset every time any SAS/GRAPH procedure is used. The SAS documentation for GOPTIONS can be consulted for more selections and information.

```
goptions reset=all device=activex xpixels=800 ypixels=600  ;
symbol1 i=ms                                    c=bwh      co=libgr;
symbol2 i=ms                                    c=gwh      co=libgr;
symbol3 i=ms                                    c=pkwh     co=libgr;
symbol4 i=none     v=dot     l=1  h=1    w=1    c=blue;
symbol5 i=join     v=none    l=1  h=1    w=1    c=black;
legend1 across=3;
axis1 label=("X Axis Label" ) value=(f="SWISS" h=1 angle=90 rotate=0)  minor=(number=3)
        order=(&plotstart to &forecastend by dthour);
axis2 label=("Y Axis Label" angle=90 rotate=0) value=(h=1) major=(number=12);
```

PLOTTING THE FORECAST

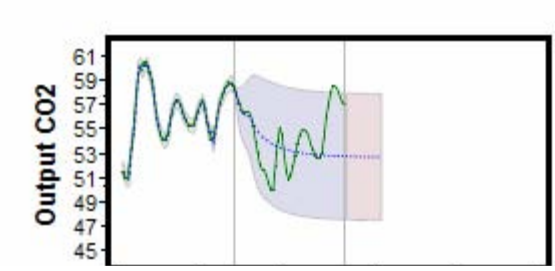
The incremental numbers after the equal sign correspond to the symbol.

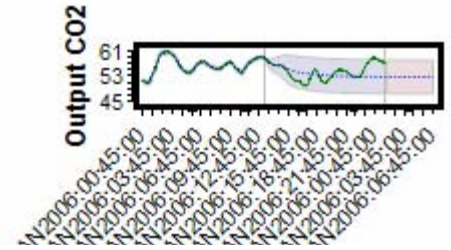
```
proc gplot data=forecast annotate=daylines;
  plot  sval0*date=1
        sval1*date=2
        sval2*date=3
        actual*date=4
        predict*date=5
       / grid haxis=axis1 vaxis=axis2 legend=legend1   overlay   chref=palg;
  format date TOD5.;
run; quit;
```

ADDITIONAL CONCERNS

DEALING WITH COMPLICATED DATETIME INTERVALS

SAS DATETIME formats are very difficult to display concisely on an axis. The length of the string representation is too long, and using major and minor tick marks makes the graph appear too cluttered, or presents too little information for use. The axis values can be angled, but the result is far from satisfactory.

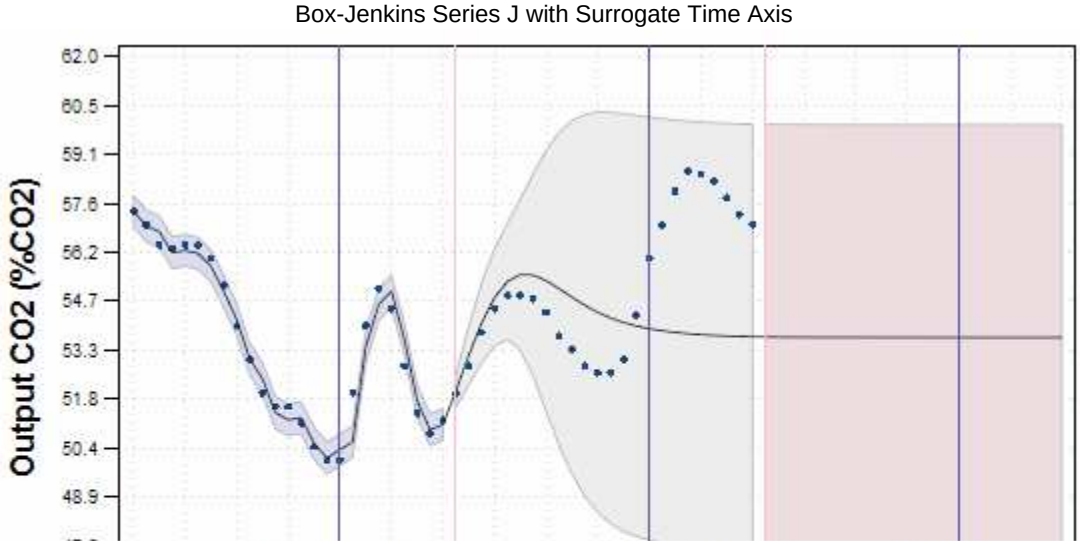




One effective solution is to present only the time values on the X-axis and annotate the graph to show the date values. This is done using the ANNOTATE facility. This process is begun by defining an ANNOTATE data set. This technique could be done using other sub-intervals such as hour-minute-second breaks. The same ANNOTATE data set can also be used to draw other data driven objects on the graph. In the code below black vertical lines are drawn to delimit the different days and pink vertical lines show when the regions change.

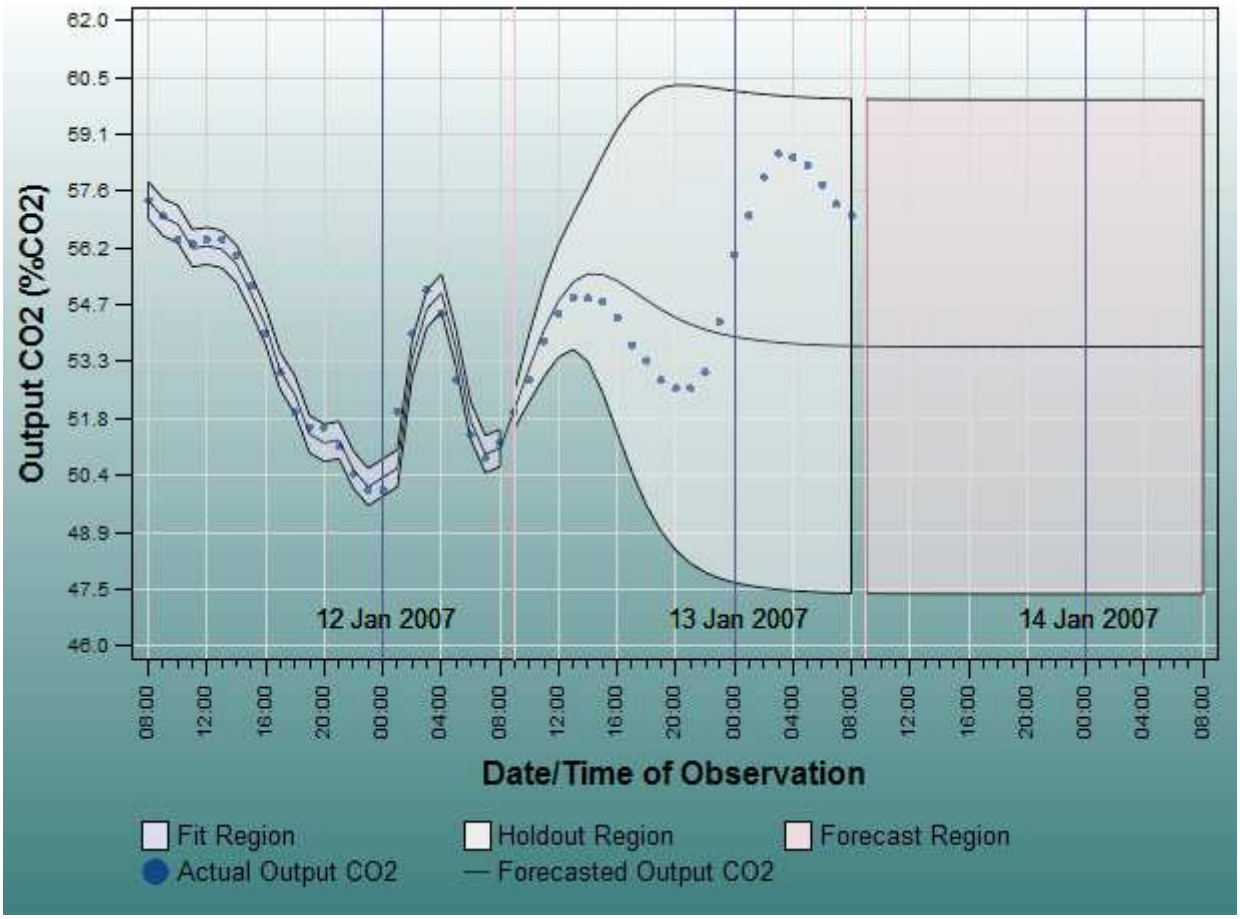
```
data DayLines; set forecast(keep=date );
        length color function $8 text $25;
        retain xsys '2' ysys '1' when 'a';
        if hour(date)=0 and minute(date)=0
                and date>=intnx('dthour',&plotstart,-1) then do;
                wdate=put(datepart(date),worddatx12.);
                function='move'; x=date; y=0; output;
                function='draw'; x=date; y=100; color='lib'; size=1; output;
                function='label';x=date; y=5; size=1; position='2';angle=90;color='black';
                text=wdate; output;
        end;
        if date=intnx('dthour',&forecaststart,1) or date=intnx('dthour',&dataend,1) then
        do;
                function='move';x=date; y=0;                          output;
                function='draw';x=date; y=100; color='pink'; size=1; output;
        end;
run;
```

This ANNOTATE data set, specified in the PROC GPLOT statement, displays vertical black lines at midnight labeled with the date. This takes away the necessity to display the date on the X axis, and a smaller format can be used. The TOD5. format is a good choice to generate HH:MM X axis values.



ADDITIONAL REFINEMENTS

Using special output devices such as ACTIVEX, a good deal of customization is possible. Right clicking on the ACTIVEX graph in a web browser brings up a control panel. All SAS Styles are available via the drop down box, or the user can create a custom display fairly easily and without coding.



CONCLUSION

Effective forecast visualization is a difficult due to the amount of information that must be displayed and complicated by many factors. It is very important for any visualization to be very accessible to a non-technical audience so it is necessary to keep things as simple as possible. SAS/GRAPH has a fairly comprehensive set of tools that allow the automation of the production of forecast graphs with no need to export the forecast data to another package.

REFERENCES

SAS Institute Inc., *Sample 1151: Filling the area between plot lines using SYMBOL statement*. http://support.sas.com/ctx/samples/index.jsp?sid=1151

SAS Institute Inc., *ARIMA: Models for Series J from Box-Jenkins* http://ftp.sas.com/techsup/download/sample/samp_lib/etssampArima_Models_for_Series_J_from_B.html

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994), *Time Series Analysis: Forecasting and Control*, Englewood Cliffs, NJ: Prentice Hall, Inc.

Brocklebank, John C., and Dickey, David A. (1994), *SAS for Forecasting Time Series*, Cary, NC: SAS Institute/John Wiley and Sons;

ACKNOWLEDGMENTS

A big thanks to ALTEK Information Technology Inc. for sponsoring this paper. Thanks to Shane Hornibrook and Tomonori Ishikawa for helping me to document this process.

RECOMMENDED READING

Using SAS Annotate Data Sets http://support.sas.com/onlinedoc/913/getDoc/en/graphref.hlp/annodata-ov.htm
SAS/Graph Colors http://www.devenezia.com/docs/SAS/sas-colors.html
SAS Sample http://support.sas.com/ctx/samples/index.jsp?sid=1151&tab=about
Author's Webpage http://www.scoyote.net/forecasting

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Feel free to contact the author for more information, source code or comments at:

Samuel T. Croker
ALTEK Information Technology
241 East 4th Street, Suite 102
Frederick, MD 21701
Work Phone: 803-240-2805
Email: scroker@al-tekinc.com or scoyote@scoyote.net
Web: www.scoyote.net/forecasting

APPENDIX A: GENERALIZED SOURCE CODE

More source code and explanations on how to use it are listed on the author's website. To run this code, the following must be added to the SAS sample for Box Jenkins Series J:

To PROC ARIMA (at the end): `forecast out=fcp back=24 lead=48 id=timeid interval=hour;`
To the data step: `timeid='01JAN2007/00:00:00'dt + (_n_*3600);`

```
/*************************************************************************
*       Program Name    :       $RCSfile: threeregionforecast.sas,v $
*       REV/REV AUTH    :       $Revision: 1.1 $ $Author: scoyote $
*       REV DATE        :       $Date: 2007/03/05 18:54:16 $
*      Autonr           :      SAMUEL T. CROKER
*************************************************************************/

%macro threeregionforecast(
        /* positional parameters (required) */
         ds                    /* dataset containing forecasts */
        ,dataend               /* ending date of the time series - not the forecast series */
        ,datastart             /* starting date of the time series */
        ,plotback              /* values to go back from the end of the series in the plot */
        ,predictlag            /* how many date values to predict ahead */
        /* optional parameters - not an exhaustive list */
        ,order=                /* date axis ordering (should be order=(<<data>>) )*/
        ,lciname=L95           /* variable name of the lower confidence bound */
        ,uciname=U95           /* variable name of the upper confidence bound */
        ,rsname=residual       /* variable name of the residual values */
        ,fcname=forecast       /* variable name of the forecast values */
        ,varname=actual        /* variable name of the actual values */
        ,varlab=               /* label for the actual values */
        ,fclab=                /* label for the predict values */
        ,dtm=date              /* variable name of the date axis variable */
        ,dtdisplay=datetime28. /* default format for the datetime axis */
        ,dtformat=mmddyy8.     /* format for time axis values */
        ,gname=FCST            /* name for the SAS graph output */
        ,gcat=work.GSEG        /* ouptut catalog */
```

```
,gdesc=Forecast Plot   /* description and title of the graph */
,fontname=SWISS        /* fonts */
,htitle=1              /* title height */
,cback=white           /* background color */
,grtitle=              /* main title of the graph */
,vaxisvalh=1           /* vertical axis value height */
,haxisvalh=1           /* horizontal axis value height */
,xinterval=hour.       /* order interval for the x axis */
,xminorticks=none
,ymajnum=10            /* number of y tick marks */
,hatitle=              /* horizontal axis title */
,vatitle=              /* vertical axis title */
,cicol1=bwh            /* first confidence region color */
,cicol2=gwh            /* second confidence region color */
,cicol3=pkwh           /* third confidence region color */
/* the following apply to actual symbol values */
,actcol=vigb           /* color */
,acth=1                /* height */
,actw=1                /* width */
,actv=dot              /* value type */
,actl=1                /* line type */
,acti=none             /* interpolation type */
/* the following apply to forecast symbol values */
,fcstcol=degb          /* color */
,fcsth=1               /* height */
,fcstw=1               /* width */
,fcstl=1               /* line type */
,fcstv=none            /* value type */
,fcsti=j               /* interpolation type */
);
data _null_;
        format forecaststart forecastend 20.;
        forecaststart=intnx('dthour',&dataend,-&plotback);
        forecastend=intnx('dthour',forecaststart,&predictlag);
        plotstart=intnx('dthour',forecaststart,-&plotback);
        call symput ('forecaststart',forecaststart);
        call symput ('forecastend',forecastend);
        call symput ('plotstart',plotstart);
run;
/* rebuild the output data so that the cis plot as polygons */
data out(  drop=     sval0 sval1 sval2)
                low0( keep=&dtm sval0 sval1 sval2)
                high0(keep=&dtm sval0 sval1 sval2)
                low1( keep=&dtm sval0 sval1 sval2)
                high1(keep=&dtm sval0 sval1 sval2)
                low2( keep=&dtm sval0 sval1 sval2)
                high2(keep=&dtm sval0 sval1 sval2);
        set &ds;
        where &dtm>=&plotstart;
        output out;
        if &dtm <= &forecaststart then do;
                sval0=&lciname; output low0;
                sval0=&uciname; output high0;
        end;
        if &dtm > &forecaststart and &dtm <= &dataend then do;
                sval1=&lciname; output low1;
                sval1=&uciname; output high1;
```

```
                end;
                if &dtm > &dataend then do;
                        sval2=&lciname; output low2;
                        sval2=&uciname; output high2;
                end;
        run;
        /* sort the lower bound datasets so that the polygons will be drawn correctly */
        proc sort data=low0; by descending &dtm; run;
        proc sort data=low1; by descending &dtm; run;
        proc sort data=low2; by descending &dtm; run;

        /* stack the low and high datasets in this way so that the graphs will be drawn correctly
*/
        data forecast;
                set
                        low2 high2
                        low1 high1
                        low0 high0
                        out;
                if &dtm=. then delete;
        run;

        /* generate vertical lines to denote the date, and highlight the start of the different
regions */
        data DayLines; set forecast(keep=&dtm );
                length color function $8 text $25;
                retain xsys '2' ysys '1' when 'a';
                if hour(&dtm)=0 and minute(&dtm)=0
                        and &dtm>=intnx('dthour',&plotstart,-1) then do;
                        wdate=put(datepart(&dtm),worddatx12.);
                        function='move'; x=&dtm; y=0;
                                output;
                        function='draw'; x=&dtm;
                                y=100; color='lib'; size=1; output;
                        function='label';x=&dtm;
                                y=5; size=1; position='2';
                                angle=90;color='black'; text=wdate; output;
                end;
                if &dtm=intnx('dthour',&forecaststart,1)
                        or &dtm=intnx('dthour',&dataend,1) then do;
                        function='move';x=&dtm; y=0; output;
                        function='draw';x=&dtm; y=100; color='pink'; size=1; output;
                end;
        run;


/* draw the graph */
        goptions reset=all
                        device=activex
                        xpixels=800
                        ypixels=600
                        ftext="&fontname"
                        htitle=&htitle
                        cback=&cback  ;
        title &grtitle;
        symbol1 i=ms                                    c=&cicol1  co=libgr;
        symbol2 i=ms                                    c=&cicol2  co=libgr;
```

```
        symbol3 i=ms                                    c=&cicol3  co=libgr;
        symbol4 i=&acti  v=&actv    l=&actl  h=&acth    w=&actw    c=&actcol;
        symbol5 i=&fcsti v=&fcstv   l=&fcstl h=&fcsth   w=&fcstw   c=&fcstcol;
        legend1 across=3;
        title &grtitle;
        axis1 label=(&hatitle )
                value=(f="&fontname" h=&haxisvalh angle=90 rotate=0)
                minor=(number=&xminorticks)
                order=(&plotstart to &forecastend by &xinterval);
        axis2 label=(&vatitle angle=90 rotate=0)
                value=(h=&vaxisvalh)
                major=(number=&ymajnum);
        proc gplot data=forecast gout=work.gseg annotate=daylines;
                label sval0='Fit Region';
                label sval1='Holdout Region';
                label sval2='Forecast Region';
                label &varname=&varlab;
                label &fcname=&fclab;
                plot  sval0*&dtm=1
                sval1*&dtm=2
                sval2*&dtm=3
                &varname*&dtm=4
                &fcname*&dtm=5
                        /       name="&gname" des="&gdesc "
                                grid
                                haxis=axis1
                                vaxis=axis2
                                legend=legend1
                                overlay
                                chref=palg;
                format &dtm &dtdisplay;
        run; quit;
%mend threeregionforecast;

/* Sample macro call */
%threeregionforecast(
                fcp                          /* dataset containing forecasts */
                ,"13JAN2007:08:00:00"dt     /* ending date of the time series */
                ,"01JAN2007:01:00:00"dt     /* starting date of the time series */
                ,24                          /* Corresponds to BACK in PROC ARIMA */
                ,48                          /* Corresponds to LEAD in PROC ARIMA */
                ,varname=y                   /* variable name of the actual values */
                ,dtm=timeid                  /* variable name of the date axis variable */
                ,dtformat=dthour.
                ,xinterval=hour4.
                ,xminorticks=3
                ,ymajnum=12
                ,dtdisplay=tod5.
                ,acth=.4
                ,grtitle="Box Jenkins Series J"
                ,hatitle="Date/Time of Observation"
                ,vatitle='Output CO2 (%CO2)'
                ,fclab='Forecasted Output CO2'
                ,varlab='Actual Output CO2'
                );
/* END OF PROGRAM */
```