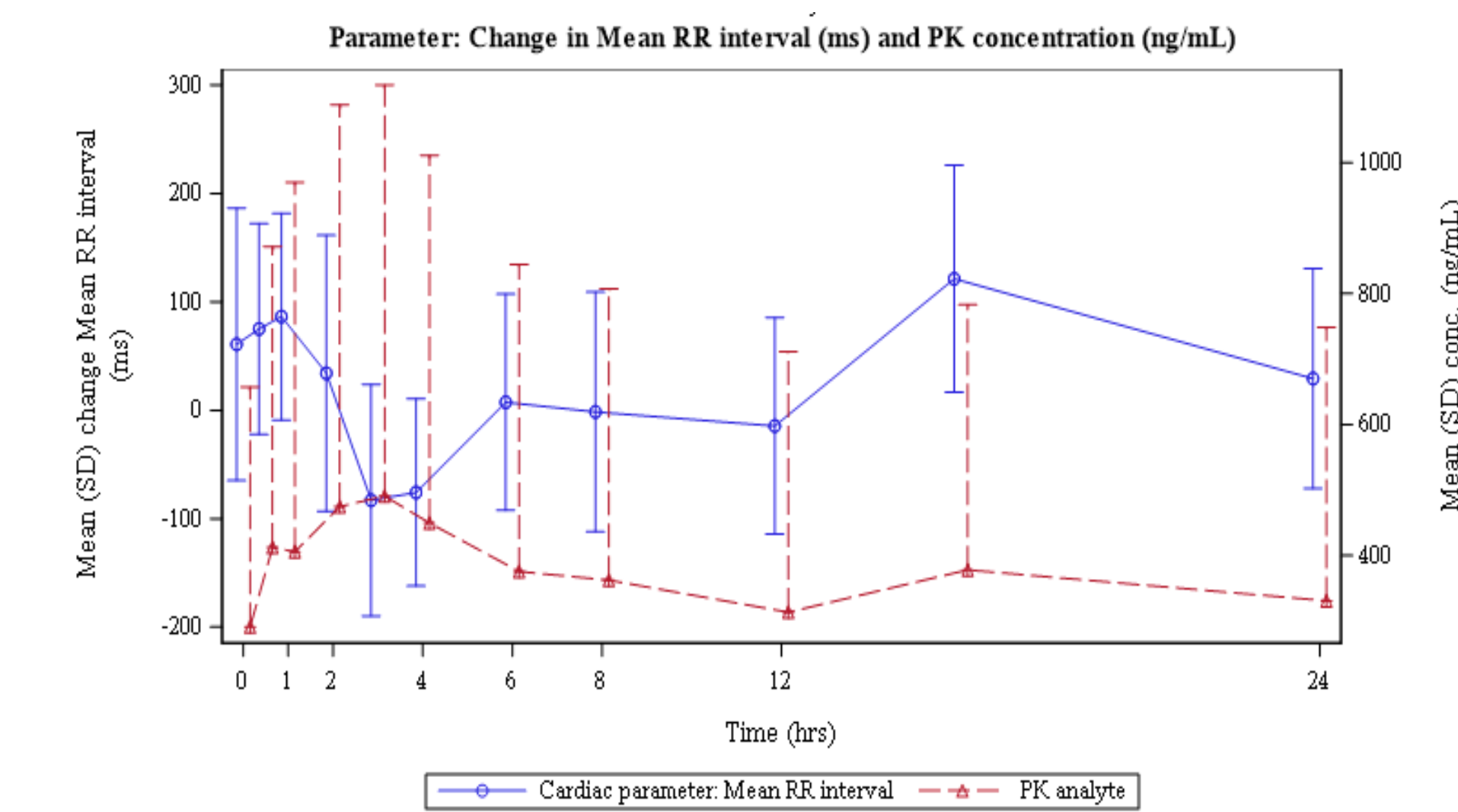


Abstract

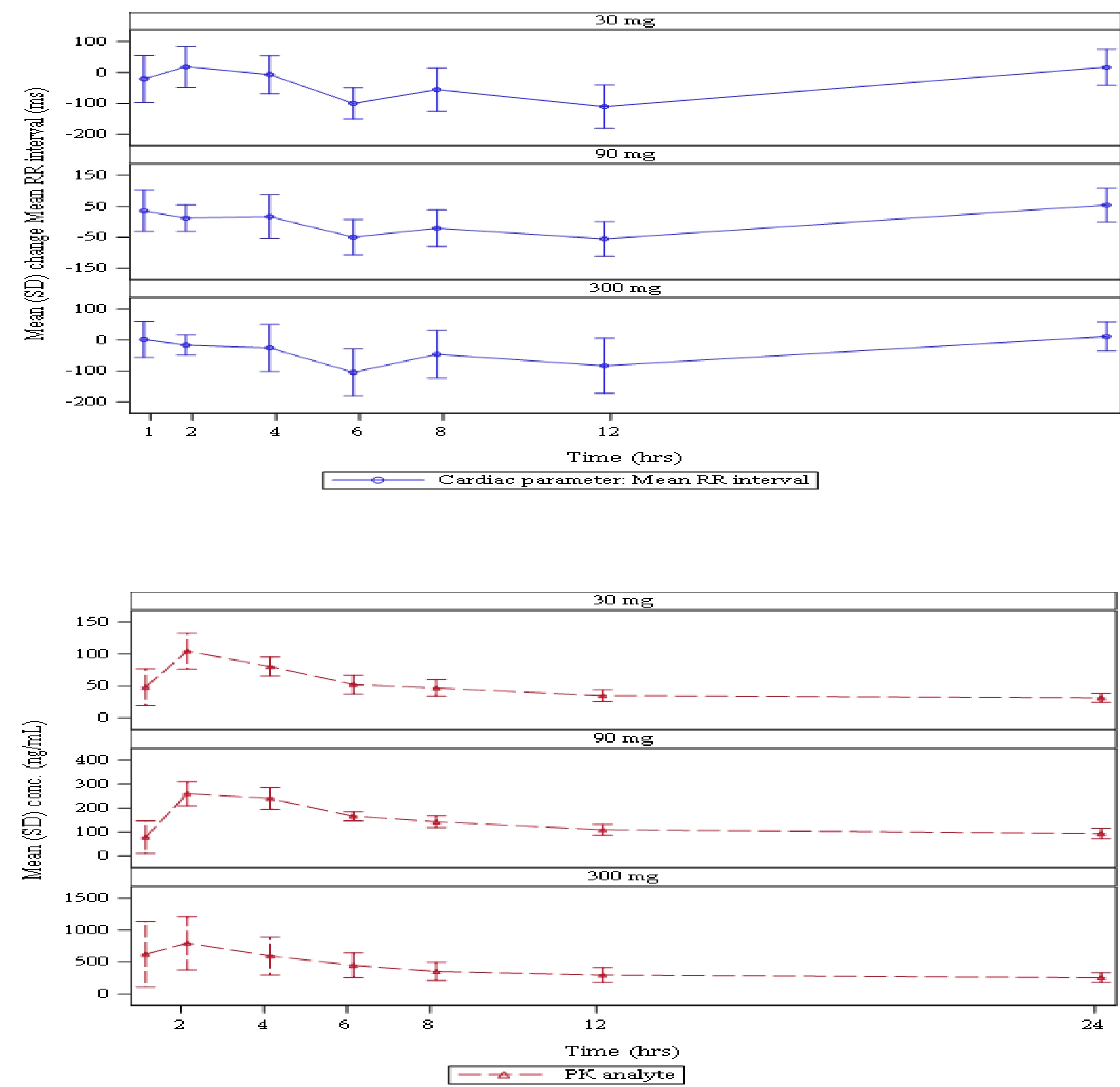
Graphs are a visual representation of the data. In a clinical trial setting, figures show the general trend and the relationships of variables collected during the conduct of a study. In the pharmacokinetic world, graphs play an important role in showing the drug’s concentration profiles. The line graph is a typical way of presenting drug’s concentration. However, depending on the data available and the analysis required, different visual presentations such as scatter plots, bar plots and many more, are needed to fully communicate the information at hand. Often times, panel by dose plots are requested to plot clinical data such as vital signs and electrocardiogram (ECG) data against pharmacokinetic concentration data over time to spot existing relationships between the two parameters. These panelled graphs can be achieved by using PROC SGPANEL and PROC SGRENDER (using GTL). This poster with the accompanying paper will explore both the procedures for producing the panel by dose plots along with the pros and cons of different procedures.



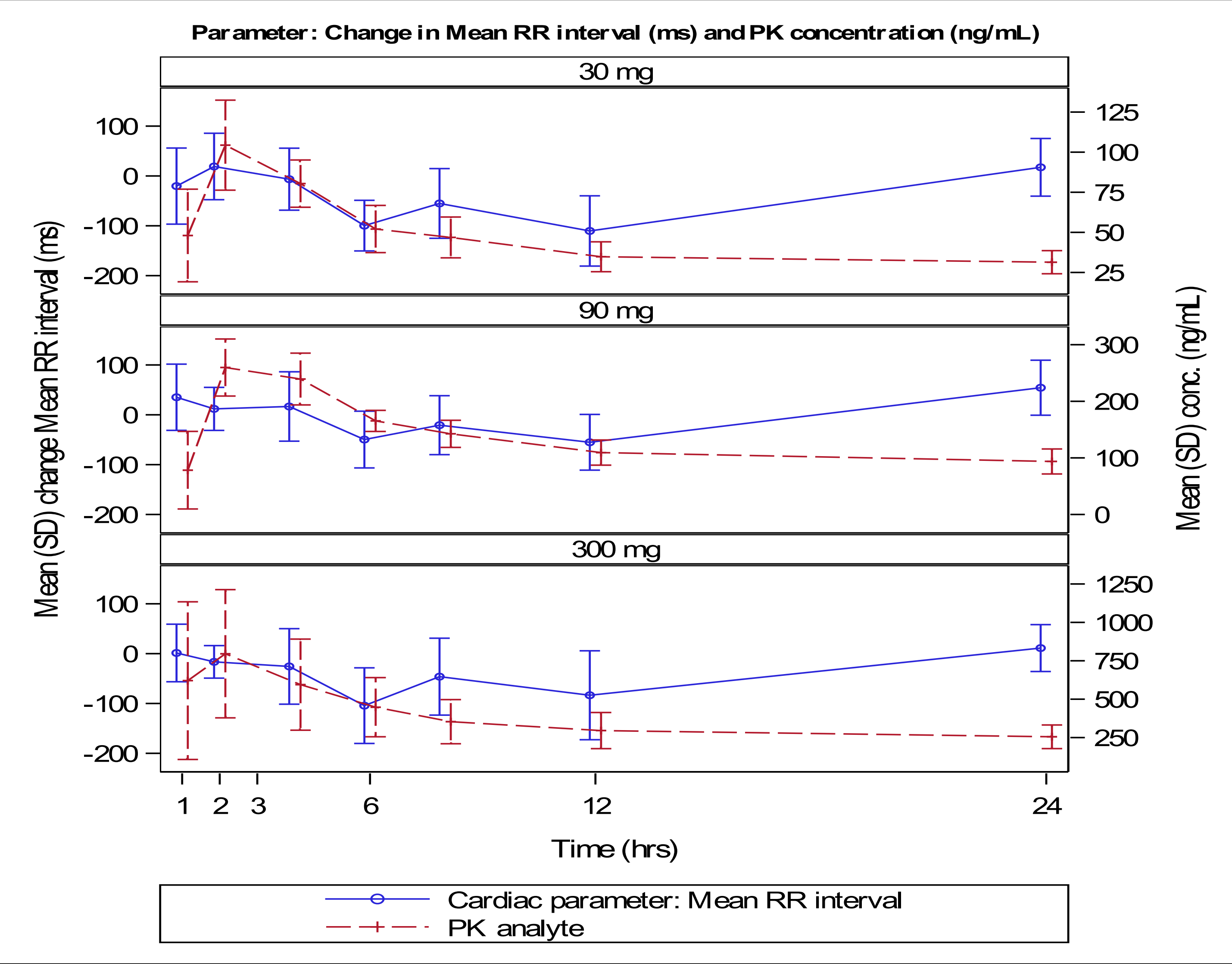
SGPANEL Procedure

SGRENDER Procedure

The SGPANEL procedure creates a panel of graph cells for the values of one or more classification variables defined in the PANELBY statement.



Y2AXIS option on the plot statement is not supported in SGPANEL. So, it gets difficult to plot both the clinical data and PK data on the same graph using SGPANEL procedure.



Discussion

The SGRENDER and the SGPANEL procedures provide alternative ways to produce modern statistical graphs. Both are powerful and flexible, but the GTL offers the greatest power, whereas the SGPANEL have a simpler syntax. Each approach has its own advantages and disadvantages, and which one to use depends on the situation and the kind of plot.

The SGRENDER procedure creates customized plots by rendering user-defined templates written in the GTL. Similar to the SGPANEL, which uses the PANEL layout, GTL also offers the DATAPANEL layout giving more freedom and flexibility to produce highly customizable graphs.

```
proc template;
  * define graph template;
  define statgraph panelbytemplate;
    * defines the values for dynamic variables used in sgrender procedure to specify required attributes;
    dynamic _byline _byval2 _byval3 _byval4 _byval5 _byval6 _byval7 _byval8 _byval9 _byval10;
    * defines macro variables;
    mvar rowrange;
    * graph definition;
    begingraph;
      * parameter title;
      entrytitle textattrs=(*options*) "Parameter: Change in " _byval2 _byval3 " and PK concentration (" _byval5 _byval10);
      * graph layout: panel of multiple graphs;
      * datalattice layout creates a grid of plots based on one or more classification variables and a graphical prototype.
      This requires a row classifier, a column classifier, or both;
      layout datalattice rowvar=(*classification variable*) / columnweight=Proportional
        rows=(*reference*) skipemptycells=true
        headerlabeldisplay=value headerlabellocation=inside
        columnaxisopts=(label="Time (hrs)" type=linear linearopts=(tickvaluelist=(0 1 2 3 6 12 24)))
        rowdatarange=rowrange rowaxisopts=(label= _byval1)
        row2datarange=union row2axisopts=(label="Mean (SD) conc. (ng/mL)");
    * graph prototype for each cell in the panel;
    layout prototype;
      * PD;
      scatterplot x=stm_pd y=mn_pdc / yaxis=y YErrorUpper=upp_pdc YErrorLower=low_pdc markerattrs=graphdata1 errorbarattrs=graphdata1;
      seriesplot x=stm_pd y=mn_pdc / yaxis=y name="pd" display=(markers) lineattrs=graphdata1 markerattrs=graphdata1;
      * PK;
      scatterplot x=stm_pk y=mn_pk / yaxis=y2 YErrorUpper=upp_pk YErrorLower=low_pk markerattrs=graphdata2 errorbarattrs=graphdata2;
      seriesplot x=stm_pk y=mn_pk / yaxis=y2 name="pk" display=(markers) lineattrs=graphdata2 markerattrs=graphdata2;
    endlayout;
  * legend;
  sidebar;
    discretelegend 'pd' 'pk' / across=1 down=2 displayclipped=true;
  endsidebar;
endlayout;
endgraph;
end;
```