Coding for Efficient Presentations: The Case of Trend Tests

Karen M. Schlangen, Hollis-Eden Pharmaceuticals, Inc., San Diego, CA

ABSTRACT
An efficient presentation of results tends to promote understanding of the subject matter and enhances readability. Presentations, and communications in general, attain their greatest value when they are friendly to many readers, not only to the ones who initiated the request. Producing a professional visual representation of extensive technical data is a challenge that almost always involves the programmer having to strike a balance among competing factors such as printing space, legibility, and level of detail.

In the pharmaceutical setting, dose-response trend plots represent a common analytical approach for pre-clinical and clinical trials data. Studies often involve the assessment of trends across multiple time points. Producing multiple plots per page is optimal for showing these plots over time.

The primary objective of this poster is to present SAS® code (as a macro routine) that produces compact and efficient graphical output, as well as tabular output with statistical trends tests, all in one document. The resulting output promotes better communication of technical data without compromising clarity or level of detail. Regarding the tabular output, PROC REG is used to calculate the slope and its significance, and PROC FREQ is used to show a nonparametric variant to assess trends: the Jonckheere-Terpstra (JT) test. Finally, an alternative approach will be presented to assess trends in the medians using the SAS (experimental) QUANTREG procedure.

To conclude, we offer some basic guidance to programmers on dose-response work, and suggest areas where further coding might perhaps be useful.

INTRODUCTION
The long process of discovering if a drug will ultimately be of clinical benefit will involve a myriad of data sets, all of which will hopefully reduce to a regulatory letter of approval. Clearly, efficient communication of quantitative information is one key to expedite the road to success (Friedman et al., 1998). We take the analysis of trends as a case in point. Trends in data surface and pervade many parts of drug development. The presence of a trend is the presence of a signal in favor of activity, or perhaps an indication of an emerging safety concern.

Not surprisingly, there are numerous ways to present and model these trends. This topic is clearly one involving many areas of expertise, and one that will be reviewed only briefly in this paper. However, the presentation of visual trends often falls in the realm of the programmer and things programmatic. By default, the programmer needs to deal with the issue of space and how to present a compact presentation. This issue is the central topic of this presentation. I will present macro SAS code to output 6 trend plots per page, and also present code to create a statistical summary table to support the plots, all in one document.

METHODS

CLINICAL TRIAL EXAMPLE
Consider an experimental drug thought to boost the immune system. Neutrophils, a type of white blood cell, play a major role in the body’s innate defense against bacteria, viruses, and fungi, and can be used to measure the drug activity against infections. Increased levels of neutrophil counts would indicate a better outcome. But would these counts show any movement in a healthy, uninfected person? An upward trend in neutrophil count as the dose level increases would be consistent with early activity in the healthy, and may indicate to regulatory authorities that, barring safety concerns, it is perhaps worth trying the drug in sick, more vulnerable individuals for whom the drug is intended. Neutrophils are often counted as number of neutrophils per nanoliter (nL), or cells/nL, a very small amount (1 nanoliter = 3.4 x 10^{-8} US fluid ounce), but a magnitude that can make the difference between life and death.

Twenty-seven healthy volunteers participated in a study designed to assess the effect of three dose levels. Subjects were randomized to groups to receive placebo (n=10), 100 milligrams (mg) (n=5), 200mg (n=6), or 400mg (n=6) of the drug compound. Each subject received a single, intramuscular injection of the assigned treatment on study day 1. Blood was
drawn to assess neutrophil count levels on study days 1 (pre-dose), 2, 5, 7, 14, 21, and 28. No trend is expected before treatment here, but in another study, a statistical check of baseline data may be of interest if baseline balance is suspect. Neutrophils were measured in cells per nL, platelet count was measured per nL, and hemoglobin was measured in grams (g) per deciliter (dL).

The data analyst’s task at hand is two fold. First, trends have to be detected and estimated analytically. Second, trends have to be shown to demonstrate the visual plausibility of the test claim. The first task is usually in the hands of the professional statisticians on the team. Here we focus on the second aspect of the task.

Code 1 is provided below to recreate the fictitious `trends.sas7bdat` data set used in the macro. Having this data set available may aid in understanding the macro code in this particular case, so that it can be adapted to other data. Table 1 is provided to show the resulting structure of the data set for a selected number of observations.

**Code 1:**

```sas
data trends(drop=i nTotal);
  infile datalines dlm=' ';
  input group group2 dose $ nTotal;
  do i = 1 to nTotal;
    id = (group*10) + i;
    do studyday = 1, 2, 5, 7, 14, 21, 28;
      input neut plat hgb @;
      output;
    end;
  end;
datalines;
  1 0 placebo 10
  4.2 361 7.6 4 358 7.9 4.2 383 7.9 3.2 410 7.9 4 382 7.6 4.1 409 7.4 3.9 360 7.8
  3.5 256 9.2 4.2 258 9.4 4 244 9.4 4.3 232 9.3 4.1 252 9.5 4.2 274 9 3.8 233 8.5
  2.8 228 8.3 2.8 238 8 2.5 228 7.8 2.8 244 7.6 3.7 243 7.8 2.6 251 8.1 2.1 238 7.7
  2.6 210 9.8 3.2 226 9.3 2.9 241 9.3 2.8 242 9.6 3 234 9.4 2.8 227 9.7 3.1 239 9.8
  3.5 307 9.5 3.9 321 9.3 3.8 287 9.7 3.4 268 9.5 3.6 280 9.3 4.2 372 9.5 3.4 293 8.6
  2.3 196 9 2.3 201 8.8 2.1 204 9.4 2.4 193 8.7 2.3 190 8.8 2.6 206 8.6 2.3 194 8.9
  3 216 9.4 3 221 9.2 3 228 9 2.9 211 8.8 3.5 221 9.4 3.6 229 8.9 3 229 9.1
  3.9 363 9.4 3.3 365 9.5 3.3 398 9.4 2.9 373 9.1 3.6 240 9.5 4.2 419 9.5 3.1 398 9.7
  2.4 223 8.9 2.9 233 8.6 2.7 230 8.6 2.2 230 8.6 3.3 222 8.7 4.2 220 8.5 3.5 260 9.1
  2.5 204 9.3 2.8 196 9.2 2.6 177 9.1 2.5 205 9.6 2.9 195 9.3 2.4 213 9.2 2.7 218 9.4
  2 100 100mg 5
  4.2 292 7.8 5.3 302 7.8 6.3 288 7.9 7.1 321 7.9 7.2 323 8.1 4.4 371 7.9 5.4 321 7.8
  3.5 260 7.8 4.8 279 8.1 5.2 272 7.8 6.4 281 8.2 6.3 284 7.9 3.5 336 8.5 3.6 284 8.5
  2.1 217 8.1 2.4 238 8.4 2.5 220 8.3 2.2 244 8.1 2.5 232 8 2.2 248 7.7 2.2 219 7.9
  2.5 155 9.5 4 172 9.4 3.8 162 9.2 4 158 9.6 3.3 167 9.3 2.8 191 9.3 2.7 175 9.9
  6.2 277 9.3 6.3 263 9.5 5.9 251 9.2 5.6 257 9.1 6.7 260 9.2 5.9 313 9.2 4.5 292 9.3
  3 200 200mg 6
  3.5 244 8 4.4 229 8 5.4 226 7.9 5.3 218 8 7.1 224 7.7 2.9 311 7.9 3.8 242 7.9
  1.8 200 8.4 2 203 8.2 2 207 8 1.9 203 8.1 2.3 223 8.4 1.8 260 8 1.9 288 8.6
  2 218 8.6 3.2 223 7.9 3.3 200 7.8 3.6 213 8 3.5 220 8.1 3.4 325 8.5 2.6 216 7.9
  2.1 238 9.3 5.2 251 9.3 4.8 232 9.4 7.3 259 9.5 6 263 9.5 3.2 350 9.5 2.9 278 9.3
  2.1 237 9.1 3.5 221 8.8 2.7 240 9.2 4.6 237 8.9 5.1 249 9 2.3 318 8.8 1.8 226 9.2
  2.1 149 9 3 148 8.8 3.7 139 8.7 4.1 144 8.7 3.2 148 8.7 3.4 192 8.9 1.9 151 9
  4 400 400mg 6
  3.1 372 8.5 3.4 337 8.1 4.6 387 8.4 5 363 8.3 5.1 378 8.4 3.8 435 8.4 3.4 332 8.2
  3.8 329 7.4 5.3 369 7.8 5.7 381 7.3 6.4 357 6.9 5.5 365 7.1 4.5 420 6.8 5 353 7.4
  4.1 332 8.1 6.8 348 8.3 6 366 8.3 7.1 352 8.2 5.9 353 8.1 5.3 327 7.8 5.3 327 7.8
  1.9 277 10.1 5.2 268 9.9 6.2 288 10 8.7 256 9.5 7.2 270 9.5 2.7 327 9.5 2.2 243 9.7
  1.2 295 9.2 2 271 9.4 2.7 282 9.2 3.5 273 9.2 4.4 300 8.9 1.6 420 9.4 1.4 275 9.3
  3.5 255 8.4 7.1 262 8.2 11.2 268 8.2 13 254 7.9 10 277 7.6 5.9 546 8.3 6 267 8.4

run;
```

```
Table 1: Selected Observations from the Data Set “trends”

<table>
<thead>
<tr>
<th>group</th>
<th>group2</th>
<th>dose</th>
<th>id</th>
<th>studyday</th>
<th>neut</th>
<th>plat</th>
<th>hgb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>placebo</td>
<td>11</td>
<td>1</td>
<td>4.2</td>
<td>361</td>
<td>7.6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>placebo</td>
<td>11</td>
<td>2</td>
<td>4.0</td>
<td>358</td>
<td>7.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>100mg</td>
<td>21</td>
<td>1</td>
<td>4.2</td>
<td>292</td>
<td>7.8</td>
</tr>
</tbody>
</table>

group2: based on dose level; neut: neutrophil count; plat: platelet count; hgb: hemoglobin

STATISTICAL TOOLS TO ASSESS TRENDS

Trends can be modeled in many ways. Accordingly, several SAS procedures work well for trends, including the following procedures: REG, GLM, FREQ, MIXED, NLIN, NLMIXED, QUANTREG, GLIMMIX, GAM (experimental in v9.1), RELIABILITY (in SAS/QC), and MULTTEST.

At first sight, it might be thought that the simplest way to model a trend is by applying a simple regression through the means, for example via PROC REG. In fact, this simplicity may bring some inherent difficulties. The model is too structured, restricted to a linear function (trends come in many other shapes), and it contains implicit assumptions about independence of observations, the normality of the errors, and the constancy of dispersion. For small samples, some of these assumptions may be suspect. This is usually the case with any model: a tradeoff is often necessary, depending on the purpose of the application, a matter most likely to fall into the hands of the project statistician. But clearly, seeing the trend is a key to successful understanding of what is being modeled.

Other tests relax some of these assumptions and may be preferred. For example, a trend in the medians instead of the means may be more robust to the presence of extreme observations that made the original data look non-normal. This type of trend can be modeled as a quantile regression through the median, implemented in PROC QUANTREG. Likewise, a model that avoids issues about the concrete shape of the trend may be able to pick whether there is any trend at all. The nonparametric Jonckheere-Terpstra (JT) does just that. It is an exact test and can be implemented via PROC FREQ.

While these are involved technical matters, it should be clear to the programmer that at the end of the day, the reader needs to be convinced that a trend is visually present, that the model makes sense, and that tabulation of the data clearly supports these contentions. A good $p$-value will also help.

SAS MACRO CODE FOR GRAPHICAL OUTPUT

The programming code presented in this paper is based on version 9.1 of SAS. The SAS modules needed to run this code are BASE, STAT, and GRAPH. A macro routine and macro parameters were used in order to output the plots and the statistical summary table at the same time. In addition, it was designed as a macro to check for trends across multiple parameters (in this case, multiple laboratory parameters), which is common practice in the industry. The code in this example is written using only one laboratory parameter, neutrophils. However, macro invocation statements are also provided for two other laboratory parameters: platelet count and hemoglobin.

Because of the length of the SAS macro code, it has been placed in an Appendix. Comments within the code explain some features of the program, along with boxed numbers in red corresponding to each step below.

1. PROC FORMAT assigns red colored font to the significant $p$-value results (<0.05), and blue colored font to the trending results (0.05 - 0.10). Refer to the Statistical Summary Table output in the RESULTS section to see the effects of this format.

2. Since the JT test is a nonparametric test for a trend between an ordinal grouping variable and another variable that is also distributed on at least an ordinal trend, either the “group” or “group2” variable can be used in the “tables” statement (“group” values = 1, 2, 3, 4; “group2” values = 0, 100, 200, 400). In the JT test SAS code, “group2” was used, but the results using “group” are the same.

3. The dependent variables used in regression should have meaningful numeric values. In other words, these values should be ordinal, represent an interval, and should have a ratio scale. For example, “group” is ordinal but it is not interval, nor does it have a ratio scale. The variable “group2”, on the other hand, preserves the dose interval accurately and has a ratio scale. For example, in “group2” the difference between the 1st group and the 2nd group, 0mg and 100mg, is correctly represented by an interval of 100. Using the values in “group” (“1”, “2”, “3”, and “4”) would not have preserved these intervals or ratios.

4. The GOPTIONS statement “device=” specifies the device driver to which SAS/GRAPH sends the procedure output.
The device driver controls both the form and destination output. The device driver “saswmf” is the Windows Metafile Format, and it is just one of many different device drivers available in SAS.

5. The statement “value=” is a text option within the axis statement. It modifies the text labels for the major tick marks on the axis. The “tick=n” suboption designates the tick mark value to alter. The present experiment has 3 dose levels and a placebo group. The “tick=4” suboption suppresses the 4th tick mark value so that the x-axis displays only the dose levels from the experiment.

6. The symbol statement “i=boxjtf” (INTERPOL=BOX<option(s)>) produces box and whisker plots. As stated in the SAS Online Documentation, the bottom and top edges of the box are located at the sample 25th and 75th percentiles. The center horizontal line is drawn at the 50th percentile (median). The default is “i=box”, in which case the vertical lines, or whiskers, are drawn from the box to the most extreme point within 1.5 inter-quartile ranges. An inter-quartile range is the distance between the 25th and the 75th sample percentiles. Any value more extreme than this is marked with a plot symbol. In the SAS example code in the Appendix, a dot is used as the plot symbol. Values for options in the symbol statement are as follows: “f” fills the box with the color specified by “cv=”, and outlines the box with the color specified by “co=”; “j” joins the median points of the boxes with a line; “t” draws the tops and bottoms on the whiskers. The option “mode=include” specifies that interpolation calculations include data values that are outside the range of plot axes. “Mode=exclude” is the default.

7. The PROC GPlot statement uses a “by studyday” line that creates plots for the six study days in the data set. Because the data set is arranged this way (multiple rows per ID), a BY statement can be used. Also of note is the use of horizontal and vertical axes options (autohref, cautohref, autovref, cautovref). These options add the light gray horizontal and vertical lines to the plots.

8. The statement “title1” is re-invoked just before the PROC GREPLAY to produce a global title for the top of the page. The first “title1” statement creates a title for each individual plot, but this individual plot title is not desirable for a global page title. Also note that because “nogtitle” and “nogfootnote” are used at the end of the “ods rtf file=” line, the titles and footnotes will print in the header and footer of the page, not within the graphic itself. Regarding the footnotes, in your SAS session you will want to put the text of each footnote on one coding line in the Enhanced Editor. This will allow for a cleaner output of your footnotes. I was limited by available space in this paper format, and continued the footnote text over multiple coding lines.

9. PROC GREPLAY is used to redisplay graphs that have been stored in temporary or permanent catalogs. The GREPLAY procedure uses three kinds of catalog entries: graphics output, templates, and color maps. The graphics output is of type GRSEG (graphics segment), and is referred to as the input-catalog and output-catalog. In the macro code, the input-catalog is used (igout) because it contains the graphics output that you want to replay.

10. The PROC CATALOG statement cleans out the graphics output catalog for the next time the macro is executed. The graphics output catalog is specified by the “entrytype=grseg” statement.

11. To create the Statistical Summary Table output in PROC REPORT, style overrides are used to format both the slope p-value cell and the JT p-value cell. Also, the “pvalue<3.” format is used to output 3 decimal places, and to format “<.001” where appropriate.

12. The “%JTplot6” statement invokes the macro and runs it based on five macro parameters: the data set name (trends), the variable name for the hematometry parameter of interest (neut), the parameter name to use in the titles (Neutrophils), the order statement for the y-axis (0 to 14 by 2), and the units (cells/mL). Regarding the macro parameter “order”, a thorough examination of your data is needed to decide on the optimal plotting range for the y-axis. One suggestion would be to run descriptive statistics on the variables you want to plot, to get the best values for “order”. Example macro invocation statements are also provided in a comment section for platelet count and hemoglobin. If desired, these three macro statements could be run simultaneously, thereby reinforcing the appeal of the macro.

RESULTS

EXAMPLE OF OUTPUT

Below is output from the PROC GPlot program when it is set up for one plot per page. It is presented to stress that clarity is retained on the 6-per-page output, which can be found in the following pages. Trends are still clear, legends are visible, and tick marks are interpretable. Figures are used for illustrative purposes to show the magnitude of a change (here, the trend). Both the 1-per-page and 6-per-page formats are effective in illustrating the magnitude of change. In both formats, the axes are clear, the titles are clear, and horizontal and vertical comparisons can be made.
Study Day 5

The bottom and top edges of the box are located at the sample 25th and 75th percentiles. The line is drawn at the 50th percentile (median). The vertical lines, or whiskers, are drawn from the box to the most extreme point within 1.5 inter-quartile ranges. Any value more extreme than this is marked with a dot plot symbol.

Below are examples of the 6-plot-per-page output and the Statistical Summary Table output, both produced by the macro program.
The bottom and top edges of the box are located at the sample 25th and 75th percentiles. The line is drawn at the 50th percentile (median). The vertical lines, or whiskers, are drawn from the box to the most extreme point within 1.5 inter-quartile ranges. Any value more extreme than this is marked with a dot plot symbol.
As discussed earlier, an alternative way to assess trends, one that uses the more robust medians as opposed to the means, is the QUANTREG procedure. Code 2 is an example of SAS code using the QUANTREG procedure, and the accompanying p-value output in tabular form. The data set is sorted for later BY variable processing by PROC QUANTREG. The statement "quantile=0.5" calls on the 50th percentile, or the median, to be used in the quantile regression.

Code 2:
```sas
proc sort data=trends out=quantreg;
  by studyday;
run;

proc quantreg data=quantreg algorithm=simplex ci=sparsity(hs)/IID;
  by studyday;
  model neut=group2/diagnostics cutoff=3 quantile=0.5;
  ods output parameterEstimates=quant2(where=(parameter='group2'));
run;
```

The SAS Institute regards QUANTREG as experimental (at the time this paper was written); see the REFERENCES section of this paper for more information on the procedure.

### DISCUSSION OF TRENDS

A simple visual inspection of the plots may lead the reader to suspect that there are indeed upward trends for some days (7 and 14), but perhaps not across every study day. The situation for days 21 and 28 looks rather flat, perhaps because the effect of a single injection does not reach that far into the study. The visual case for the presence of trends on days 2 and 5 is not easily settled. This is where a statistical analysis might be most useful. The Statistical Summary Table confirms a trend on days 7 and 14 by both the PROC REG and JT test results. Results in the table also show that there is no hope of any trend on days 21 or 28, day 2 remains unsettled, and a statistical trend on day 5 is significant after all. This might be an encouraging indication that a quick drug response is achievable. The more robust QUANTREG procedure adds valuable insight into the matter. A regression through the medians, impervious to isolated outliers, assigns a significant trend to day 2. It also appears that outliers at day 14, perhaps those at the top of the 75th percentile for the 400mg group, make a trend
unwarranted and the procedure denies it of statistical significance, which reverses the result found in the previous Statistical Summary Table. Perhaps day 14 belongs with days 21 and 28. Of course other interpretations are also possible, but we won’t pursue this matter any further.

CONCLUSION
The code advanced in this paper may assist the programmer to produce a professional presentation of technical data. This macro code produces both plots and statistical output together in one document, and can be run across multiple analysis parameters. Having both the consolidated plots and supporting statistical output together to present to the investigator, aides in better and more thorough communication of the technical data at hand.

In terms of the statistics, output from both PROC REG and the JT test provide a formal assessment of the evidence for the significance of trends. PROC QUANTREG further refines this picture by using a quantile regression analysis (through the medians in our example) instead of the means, which in turn offers a more sober, or robust, way to model the data. QUANTREG could be extended to overlay various quantiles such as the 25th, the median, and the 75th percentile, resulting in a more thorough description of the distributions involved.

It should be clear from the discussion of the trends in the example above, that no amount of coding can bypass issues of clarity or ambivalence of results. The same basic tenet applies to the presentation of data for visual purposes. Ultimately, the programmer will have to judge the adequacy of any compression of visual outputs such as tables and plots. This is an issue that involves the science of technical communication, the psychology of how people read data, and the subject matter that originated the plots. In reality, it is the programmer that often has to deal with these topics.

The code presented here can be extended in many directions to meet other programming needs. For example, the design of the graphics output could incorporate the $p$-values directly on the plots themselves, thus eliminating the need for a second page. In this case, you could use a NOTE statement to place the desired $p$-value(s) directly on each plot. There are countless ways to present output. The macro code presented here was not intended to work for all examples of trends, but hopefully it’s a good starting point to incorporate into your specific needs.

REFERENCES

ACKNOWLEDGMENTS
Many thanks are extended to Daniel Destiche who provided exceptional program code and coding guidance, to Armando Garsd for his critical reading of this paper and assistance with the statistical matters, and to Rosemay Remigio-Baker and Richard Weng for their assistance with this presentation.

RECOMMENDED READING
In the SAS Online Documentation: PROC GREPLAY; NOTE statement; #BYVAL title option

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:
Karen M. Schlangen
Statistical Programmer/Analyst
Hollis-Eden Pharmaceuticals, Inc.
4435 Eastgate Mall, Suite 400
San Diego, CA 92121
Phone: 858-587-9333
FAX: 858-320-2593
Email: kschlangen@holliseden.com
Web: www.holliseden.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.
Other brand and product names are trademarks of their respective companies.
options orientation=portrait ls=90 ps=49;
%m macro JTplot6(dataset, parameter, title, order, units);

options nobyline;

/* Set up a color format for the p-value. Used in PROC REPORT output. */
proc format;
  value color
    0 - 0.0499 = 'RED'
    0.05 - 0.1 = 'BLUE';
run;

/* Sort data for later BY-variable-processing use in PROC FREQ and PROC REG. */
/* Neutrophil values at study day 1 are baseline pre-dose values, and not of interest
   in the trend plot that has a post-dose focus. */
proc sort data=&dataset;
  by studyday;
  where studyday>1;
run;

/* Jonckheere-Terpstra (JT) test. Two-sided p-value (variable name=p2 jt) is kept. */
proc freq data=&dataset;
  by studyday;
  tables group2* &parameter/noprint jt;
  output out=JT_pvals(keep=studyday p2 jt) ;
  /*exact jt;*/      /* Warning: Exact version of JT test takes over an hour to run. */
  run;                 /* Removed here, but it is available. */

/* PROC REG code for the slope, and the p-value for the slope. */
ods output parameterEstimates=parms_&dataset
  (keep=studyday dependent variable estimate probt
   where=(variable='group2'));
proc reg data=&dataset;
  by studyday;
  model &parameter=group2;
run;
quit;
ods output close;

/* Clean data by renaming variables and keeping only those variables that are needed, and
   sort the data to merge the p-value output together for the PROC REPORT. */
proc sort data=parms_&dataset (rename=(estimate=slope probt=p_slope));
  by studyday;
run;

data &dataset._table(keep=studyday slope p_slope p2 jt);
  merge JT_pvals parms_&dataset;
  by studyday;
run;

/* Direct ODS output to c:\temp. Make sure folder is available. */
ods escapechar='~';
ods decimal_align;
ods rtf file="c:\temp\JTplot &title PharmaSUG2007.rtf" nogtitle nogfootnote;

***** Plots *****
options nobyline;
goptions reset=all gunit=pct
ftext="Arial" htext=5
colors=(black blue green red)
ymax=10in xmax=7.5in
vorigin=0.0in horigin=0.0in
vsize=9in hsize=6.5in
display
device=saswmf
goutmode=append;

axis1
order=(&order)
label=none
offset=(1.0, 1.0)cm
minor=(n=1);

axis2
order=(0 to 400 by 100) value=(tick=4 ' ')
label="Dose (mg)"
offset=(1.0, 1.0)cm
minor=none;

symbol1 i=boxjtf v=dot cv=green h=4 l=1 w=1.1 mode=include co=green;
goptions nodisplay;
title1 f='Arial' j=c h=5.3 "Study Day #byval(studyday)"
proc gplot data=&dataset gout=trends2;
by studyday;
plot &parameter*group2 / vaxis=axis1 haxis=axis2 autohref autovref
cautohref=ltgray cautovref=ltgray;
/* Note statement used to position y-axis label. */
note m=(4, 25) a=90 "&title (&units)"
run;
quit;
goptions display;
title1 f='Arial' j=c h=5.3 "Trend Plots: &title"
footnote1 f='Arial' j=c h=2 "The bottom and top edges of the box are located at
the sample 25th and 75th percentiles. The line is drawn at the 50th";
footnote2 f='Arial' j=c h=2 "percentile (median). The vertical lines, or
whiskers, are drawn from the box to the most extreme point within 1.5
inter-quartile ranges."
footnote3 f='Arial' j=c h=2 "Any value more extreme than this is marked with a
dot plot symbol."
proc greplay igout = trends2
   tc = tempcat
   nofs;
list igout;
device = saswmf;
tdef sixpage
1/llx=0 lly=68
ulx=0 uly=100
urx=48 ury=100
1rx=48 lry=68

2/llx=52 lly=68
ulx=52 uly=100
urx=100 ury=100
lrx=100  lry=68
3/lrx=0 lly=33
ulx=0 uly=66
urx=48 ury=66
lrx=48  lry=33
4/lrx=52 lly=33
ulx=52 uly=66
urx=100 ury=66
lrx=100  lry=33
5/lrx=0 lly=0
ulx=0 uly=32
urx=48 ury=32
lrx=48  lry=0
6/lrx=52 lly=0
ulx=52 uly=32
urx=100 ury=32
lrx=100  lry=0
;
template sixpage;
treplay 1:gplot2:gplot1 3:gplot2 4:gplot3 5:gplot4 6:gplot5;
run;
quit;
proc catalog cat=work.trends2;
delete gplot gplot1 gplot2 gplot3 gplot4 gplot5 / entrytype=grseg;
run;
quit;
title1 f='Arial' j=c h=5.3 "Statistical Summary Table: &title";
footnote1 f='Arial' j=c h=2 "Slope: regression coefficient for dose.  p-slope: significance of the slope in a linear regression model.";
footnote2 f='Arial' j=c h=2 "p-JT: significance of the nonparametric Jonckheere – Terpstra test for the presence of any upward or downward trend.";
footnote3 f='Arial' j=c h=2 "All tests are two-sided.";
proc report data=&dataset._table nowindows headline headskip missing split='@';
columns studyday slope p_slope p2_jt;
declare studyday   / width=8 center order        'Study Day';
declare slope      / width=8 center format=8.4 'Slope';
declare p_slope    / width=8 center style(column)={foreground=color.} format=pvalue5.3 'p-slope';
declare p2_jt      / width=8 center style(column)={foreground=color.} format=pvalue5.3 'p-JT';
run;
ods rtf close;
%mend JTplot6;

%JTplot6(trends, neut, Neutrophils, 0 to 14 by 2, cells/nL);
/* Macro invocation statements for two more example laboratory parameters.
%JTplot6(trends, plat, Platelets, 100 to 600 by 100, /nL);
%JTplot6(trends, hgb, Hemoglobin, 6 to 11 by 1, g/dL); */