

# *GTL (Graphics Template Language) in SAS 9.2*

Philip R Holland, Holland Numerics Ltd

---

---

# Agenda

- Introduction to GTL
  - Preparing the Graph Data
  - Simple Graphs using PROC SGPLOT
  - Creating ODS Graph Templates using PROC TEMPLATE
  - Using ODS Graph Templates for:
    - Line Plots
    - Forest Plots
  - Conclusions
- 
-

# *Introduction to GTL*

- GTL is a template language used to produce high quality graphs via PROC TEMPLATE.
  - The template language is similar in structure to other templates generated using PROC TEMPLATE.
  - The ODS Graph (STATGRAPH) templates created, stored in a SAS Item Store file, can be shared, controlled by parameters, and reused.
- 
-

# STATGRAPH Templates

Like all other templates generated by PROC TEMPLATE, an ODS Graph (STATGRAPH) Template is structured as a nested set of clauses, where the plotting code is located inside LAYOUT clauses that define the graph areas.

```
DEFINE STATGRAPH name; – create the template
  DYNAMIC name(s); – define any parameters (optional)
  COLUMN names(s); – define any SAS variable parameters by name (optional)
  BEGINGRAPH; – start the graph (new in SAS 9.2!)
  ENTRYTITLE title; – create a title (can be repeated for additional titles)
  LAYOUT – at least one layout statement is required
  LAYOUT – nested layout statements (optional)
  Any plot statements, including titles, graph areas, footnotes, etc.
  ENDLAYOUT;
  Any plot statements, including titles, graph areas, footnotes, etc.
  ENDLAYOUT;
  ENTRYFOOTNOTE footnote; – create a footnote (can be repeated for additional footnotes)
  ENDGRAPH;
END;
```

## Preparing the Graph Data

The following code generates simulated clinical data with visit numbers, products, an absolute value with a standard error (for the simple line plot), and a relative value (for the forest plot).

```
PROC SQL;
  CREATE TABLE plotdata AS
    SELECT INTCK('QTR', '01jan1994'd, monyr) AS visitnum
      ,product
      ,MEAN(predict) AS value1
      ,STDERR(predict) AS value1_se
      ,MEAN(predict) - 1200 AS value2
      ,COUNT(*) AS count
    FROM   sashelp.prdsal2
          (WHERE = (product IN ('BED', 'CHAIR', 'DESK')
                    AND predict > 400))
    GROUP BY visitnum, product;
QUIT;
```

---

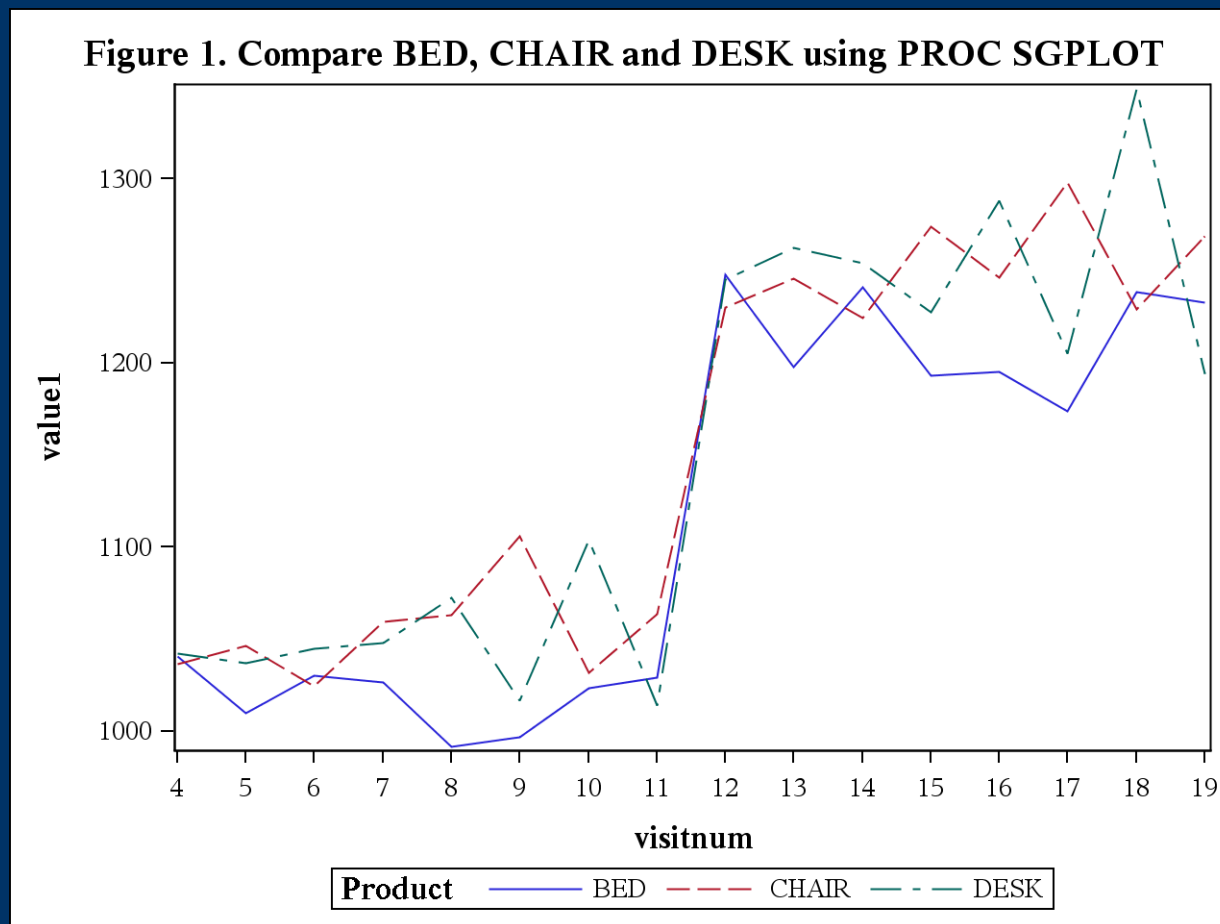
---

# Preparing the Graph Data

The graph data is shown below:

	visitnum	PRODUCT	value1	value1_se	value2	count
1	4	BED	1040.6677551	40.011098087	-159.3322449	196
2	4	CHAIR	1036.2542408	33.209771073	-163.7457592	191
3	4	DESK	1041.9857592	39.327194523	-158.0142408	191
4	5	BED	1010.1102703	37.880361433	-189.8897297	185
5	5	CHAIR	1046.4845596	39.065874299	-153.5154404	193
6	5	DESK	1037.0896842	37.467233628	-162.9103158	190
7	6	BED	1030.3346535	36.002625028	-169.6653465	202
8	6	CHAIR	1024.7254167	37.877005153	-175.2745833	192
9	6	DESK	1044.6475393	38.222565292	-155.3524607	191
10	7	BED	1026.5428571	39.581535969	-173.4571429	182
11	7	CHAIR	1059.6993814	39.215069736	-140.3006186	194
12	7	DESK	1047.9932275	40.410979863	-152.0067725	189
13	8	BED	991.50772277	33.451793946	-208.4922772	202
14	8	CHAIR	1062.9608649	38.346487743	-137.0391351	185
15	8	DESK	1072.4597895	36.865002105	-127.5402105	190
16	9	BED	996.72460733	35.528605628	-203.2753927	191
17	9	CHAIR	1105.9675556	41.999681776	-94.03244444	180
18	9	DESK	1016.6338776	35.642286028	-183.3661224	196
19	10	BED	1023.2299487	37.901280136	-176.7700513	195
20	10	CHAIR	1031.7762376	35.069337737	-168.2237624	202
21	10	DESK	1104.0844444	41.598427649	-95.91555556	198

# Simple Graphs using PROC SGPLOT (1)



# Simple Graphs using PROC SGPLOT (1)

The following basic code can be used to produce the previous graph using the new PROC SGPLOT, which is one of the new “SG” procedures introduced in SAS 9.2.

```
PROC SGPLOT DATA = plotdata;  
  TITLE  
    'Figure 1. Compare BED, CHAIR and DESK using PROC SGPLOT';  
  XAXIS TYPE = DISCRETE;  
  SERIES X = visitnum Y = value1 / GROUP = product;  
RUN;
```

---

---

# Creating ODS Graph Templates using PROC TEMPLATE

The following code demonstrates the basic structure:

```
PROC TEMPLATE;  
  DEFINE STATGRAPH Graphics.LinePlot;  
    DYNAMIC _yvar1 _yupper1 _ylower1 _nvar1 _group  
      _xvar _xlabel _ylabel _yintercepta _yinterceptb  
      _title _title2 ..... _footnote _footnote2 .....;  
    BEGINGRAPH;  
      ENTRYTITLE _title;  
      ENTRYTITLE .....;  
      LAYOUT LATTICE / COLUMNS = 1 COLUMNDATARANGE = UNIONALL  
        ROWS = 2 ROWWEIGHTS = (.85 .15);  
      .... the plotting code goes in here ....  
    ENDLAYOUT;  
    ENTRYFOOTNOTE HALIGN = LEFT _footnote;  
    ENTRYFOOTNOTE .....;  
  ENDGRAPH;  
END;  
RUN;
```

---

---

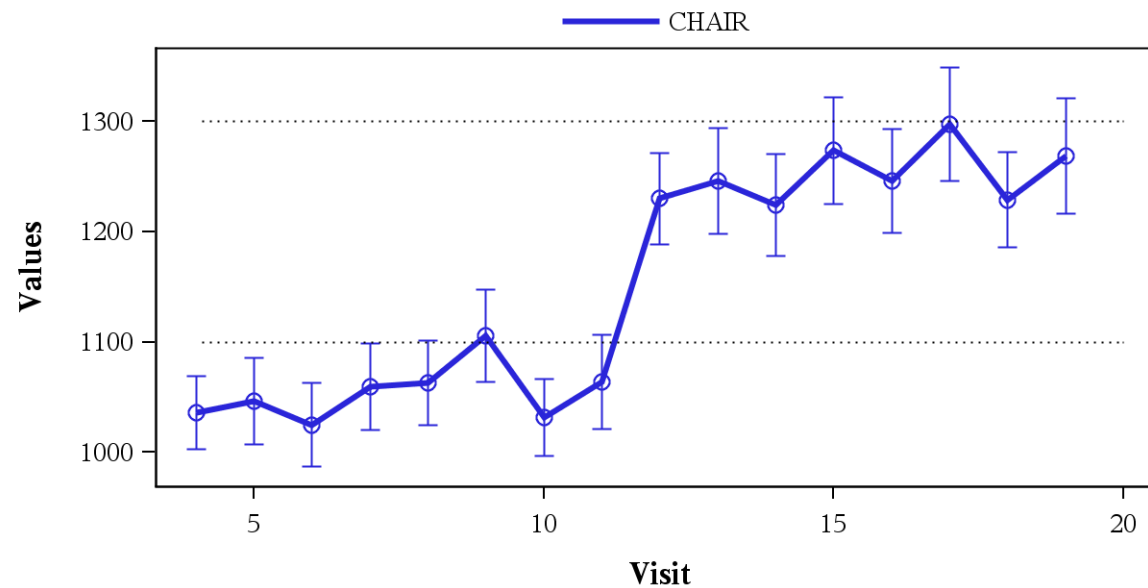
# Using ODS Templates for: Line Plots (2, 3)

Some plotting code for Line Plots:

```
LAYOUT OVERLAY / XAXISOPTS = (LABEL = _xlabel)
  YAXISOPTS = (LABEL = _ylabel) OPAQUE = FALSE
  PAD = (TOP = 2% BOTTOM = 2% LEFT = 2% RIGHT = 2%);
SERIESPLOT X = _xvar Y = _yvar1 / GROUP = _group
  NAME = 'series' MARKERATTRS = (SIZE = 10PX)
  LINEATTRS = (THICKNESS = 3PX);
SCATTERPLOT X = _xvar Y = _yvar1 / GROUP = _group
  YERRORUPPER = _yupper1 YERRORLOWER = _ylower1
  MARKERATTRS = (SIZE = 10PX);
DISCRETELEGEND 'series' / ACROSS = 4 BORDER = FALSE
  VALIGN = TOP;
ENDLAYOUT;
LAYOUT OVERLAY / PAD = (BOTTOM = 2% LEFT = 2% RIGHT = 2%)
  XAXISOPTS = (DISPLAY = NONE) X2AXISOPTS = (DISPLAY = NONE)
  Y2AXISOPTS = (DISPLAY = NONE) BORDER = FALSE
  YAXISOPTS = (DISPLAY = (TICKVALUES)) WALLDISPLAY = NONE;
SCATTERPLOT X = _xvar Y = _group / MARKERCHARACTER = _nvar1
  MARKERCHARACTERATTRS = (COLOR = BLACK);
ENDLAYOUT;
```

# Line Plots (2)

Figure 2. Plot CHAIR as line plot with reference lines  
Overall



CHAIR 191 193 192 194 185 180 202 188 207 208 206 206 204 198 218 202

Program: v92\_ods\_lineplot.sas

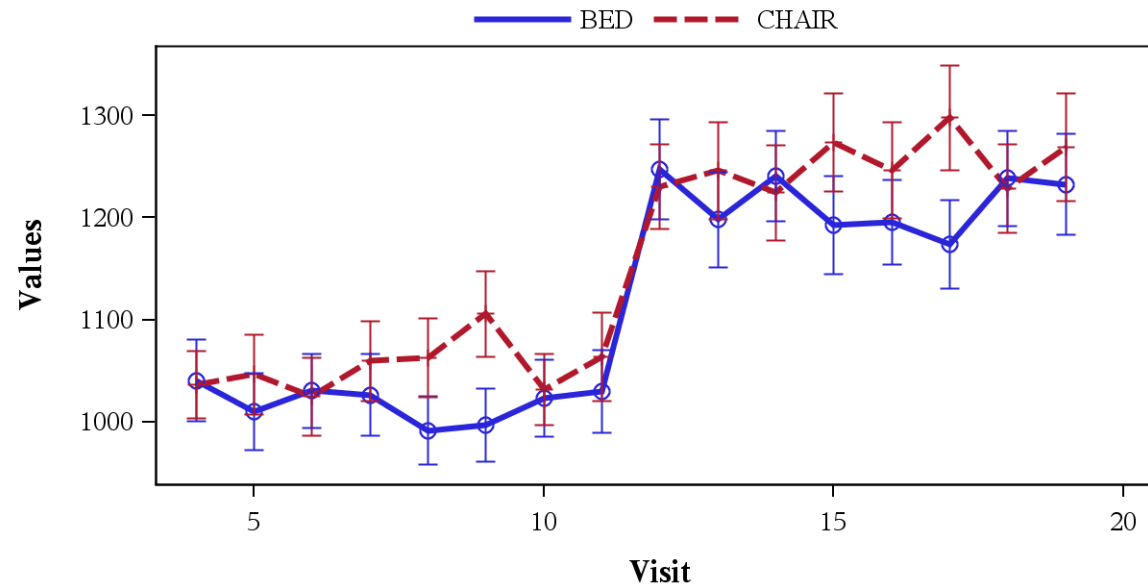
# Line Plots: DATA Step Code (2)

You only need a DATA Step to generate the plot:

```
ODS GRAPHICS ON;
DATA _NULL_;
  LENGTH ccount $4;
  SET plotdata (WHERE = (product = 'CHAIR'));
  value1_upper = value1 + value1_se;
  value1_lower = value1 - value1_se;
  ccount = STRIP(PUT(count, 4.));
  FILE PRINT ODS = (TEMPLATE = 'Graphics.LinePlot'
    DYNAMIC = (_title = "Figure 2. Plot CHAIR as line plot
                with reference lines"
              _title2 = "Overall" _footnote = "Program: &pgm..sas"
              _xvar = "visitnum"   _xlabel = "Visit"
              _yvar1 = "value1"    _ylabel = "Values"
              _yintercepta = 1300 _yinterceptb = 1100
              _yupper1 = "value1_upper" _ylower1 = "value1_lower"
              _nvar1 = "ccount"    _group = "product"));
  PUT _ODS_;
RUN;
ODS GRAPHICS OFF;
```

# Line Plots (3)

Figure 3. Compare BED with CHAIR as line plot  
Overall



CHAIR	191	193	192	194	185	180	202	188	207	208	206	206	204	198	218	202
BED	196	185	202	182	202	191	195	187	209	201	214	204	214	210	205	200

Program: v92\_ods\_lineplot.sas

## Line Plots: DATA Step Code (3)

Small amendments can make a big difference (with a different WHERE clause and no intercepts):

```
ODS GRAPHICS ON;
DATA _NULL_;
  LENGTH ccount $4;
  SET plotdata (WHERE = (product IN ('BED', 'CHAIR')));
  value1_upper = value1 + value1_se;
  value1_lower = value1 - value1_se;
  ccount = STRIP(PUT(count, 4.));
  FILE PRINT ODS = (TEMPLATE = 'Graphics.LinePlot'
    DYNAMIC = (_title = "Figure 3. Compare BED with CHAIR
      as line plot"
      _title2 = "Overall" _footnote = "Program: &pgm..sas"
      _xvar = "visitnum" _xlabel = "Visit"
      _yvar1 = "value1" _ylabel = "Values"
      _yupper1 = "value1_upper" _ylower1 = "value1_lower"
      _nvar1 = "ccount" _group = "product"));
  PUT _ODS_;
RUN;
ODS GRAPHICS OFF;
```

---

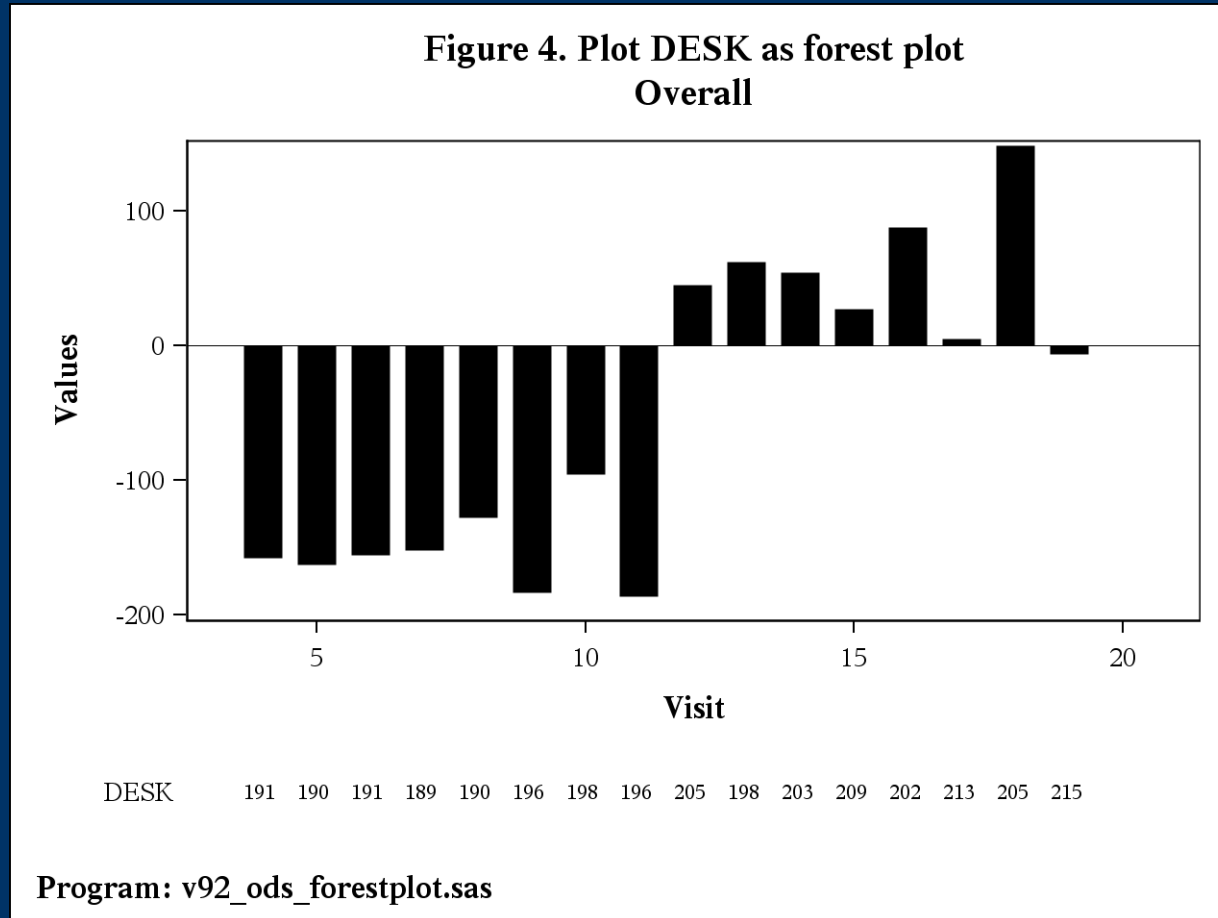
---

# Using ODS Templates for: Forest Plots (4)

Some small alterations to template plotting code can have dramatic effects too:

```
LAYOUT OVERLAY / XAXISOPTS = (LABEL = _xlabel)
  YAXISOPTS = (LABEL = _ylabel) OPAQUE = FALSE
  PAD = (TOP = 2% BOTTOM = 2% LEFT = 2% RIGHT = 2%);
  NEEDLEPLOT X = _xvar Y = _yvar1 / NAME = 'needle'
  DISPLAY = STANDARD LINEATTRS = (THICKNESS = 20PX);
ENDLAYOUT;
LAYOUT OVERLAY / PAD = (BOTTOM = 2% LEFT = 2% RIGHT = 2%)
  XAXISOPTS = (DISPLAY = NONE) X2AXISOPTS = (DISPLAY = NONE)
  Y2AXISOPTS = (DISPLAY = NONE) BORDER = FALSE
  YAXISOPTS = (DISPLAY = (TICKVALUES)) WALLDISPLAY = NONE;
  SCATTERPLOT X = _xvar Y = _group / MARKERCHARACTER = _nvar1
  MARKERCHARACTERATTRS = (COLOR = BLACK);
ENDLAYOUT;
```

# Forest Plots (4)



# Forest Plots: DATA Step Code (4)

The DATA Step still looks familiar:

```
ODS GRAPHICS ON;
DATA _NULL_;
  LENGTH ccount $4;
  SET plotdata (WHERE = (product = 'DESK'));
  ccount = STRIP(PUT(count, 4.));
  FILE PRINT ODS = (TEMPLATE = 'Graphics.ForestPlot'
    DYNAMIC = (_title = "Figure 4. Plot DESK as forest plot"
      _title2 = "Overall" _footnote = "Program: &pgm..sas"
      _xvar = "visitnum" _xlabel = "Visit"
      _yvar1 = "value2" _ylabel = "Values"
      _nvar1 = "ccount" _group = "product"));
  PUT _ODS_;
RUN;
ODS GRAPHICS OFF;
```

# Conclusions: Pros and Cons

- Disadvantages:

- Yet another "SAS" language to learn!
- Template code can be difficult to debug.
- Templates are not necessary for simple graphs, as the 'SG' procedures can be used instead.

- Advantages:

- High quality graphs can be created using templates.
  - Templates are reusable and produce the same output on all supported platforms.
  - Templates can be used to reproduce complex layouts.
- 
-

## *Conclusions: Summary*

- The time taken to learn GTL is likely to be repaid by the time saved in reusing the templates to create standard graphs.
  - The ability to reproduce graphs in Windows or UNIX will save time and effort currently required when developing graphs on multiple platforms.
  - The high quality of the graphs is a major improvement over that created by existing SAS/GRAPH procedures.
- 
-

## *Contact details*

Philip R Holland, Holland Numerics Ltd  
94 Green Drift, Royston, Herts SG8 5BT, UK  
[phil@hollandnumerics.com](mailto:phil@hollandnumerics.com)  
[www.hollandnumerics.com](http://www.hollandnumerics.com)

This paper, presentation and the SAS code used  
can all be downloaded from:  
[www.hollandnumerics.com/SASPAPER.HTM](http://www.hollandnumerics.com/SASPAPER.HTM)

---

---