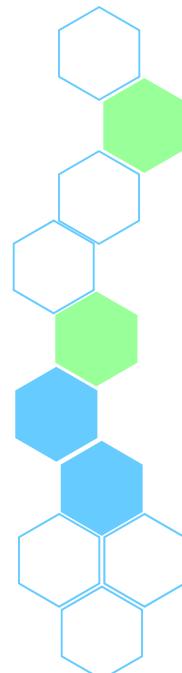
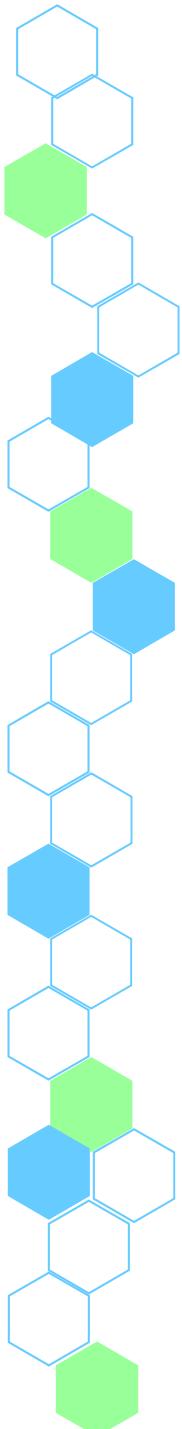


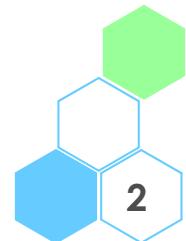
SAS sgplot 超入門

舟尾 暢男



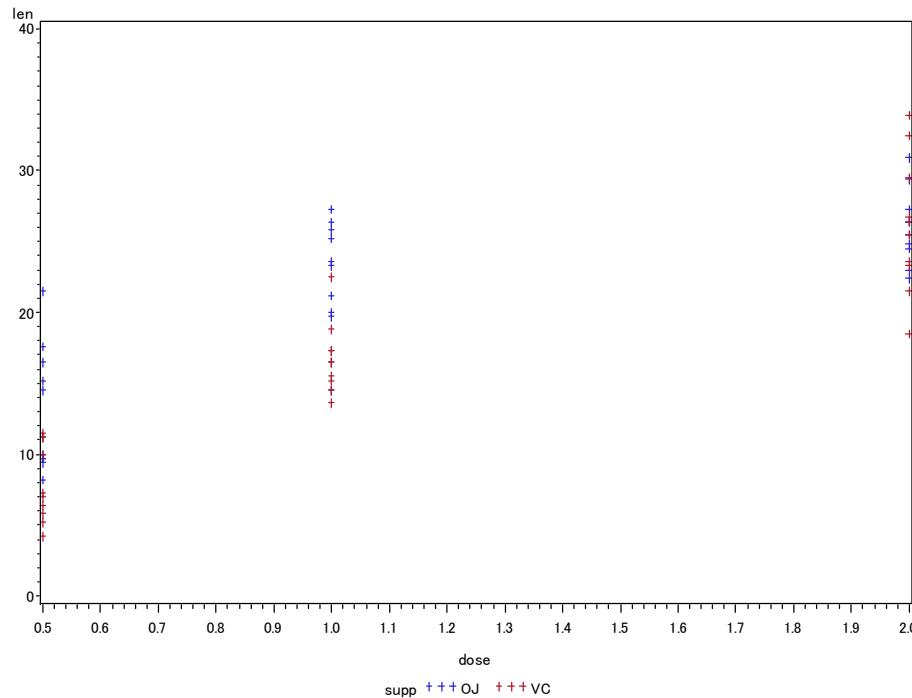
Menu

- 事始
- 種々のグラフ
- グラフのカスタマイズ
- 応用例: Forest Plot
- 応用例: Kaplan-Meier Plot

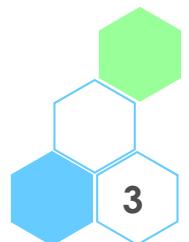
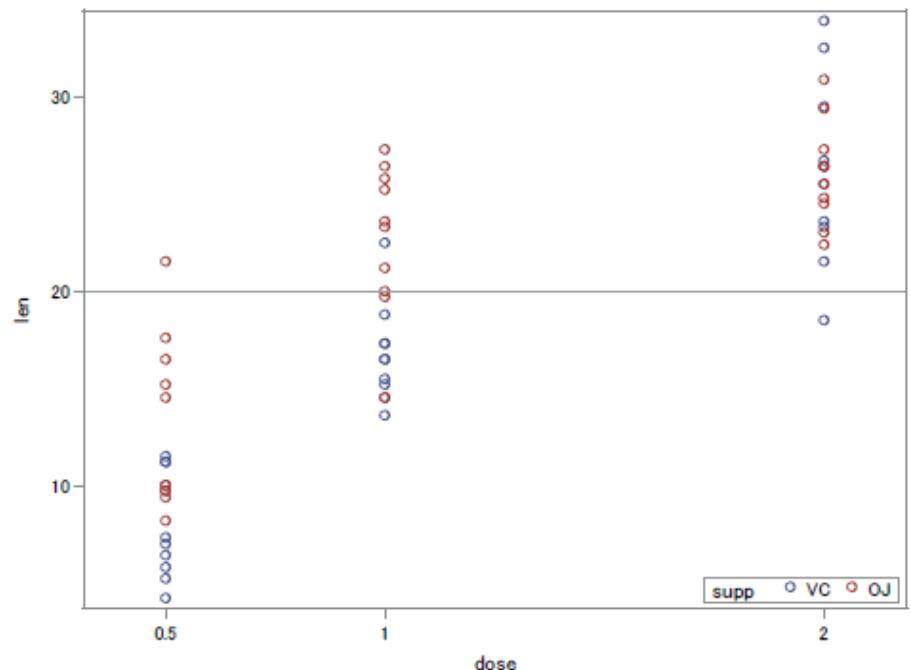


Traditional vs. sgplot

Traditional

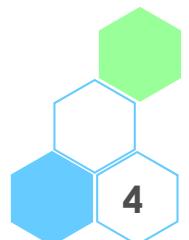


sgplot プロシジャ



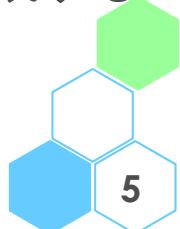
Traditional vs. sgplot

- Traditional Graphics プロシージャ
 - SAS 9.1 以前でも利用できるグラフ作成用のプロシージャ
 - gplot, gchart, boxplot, etc..
- Statistical Graphics(SG)プロシージャ
 - SAS 9.2 から利用できる新しいグラフ作成用のプロシージャ
(一部 9.1 より利用可能)
 - **sgplot**, **sgpanel**, **sgscatter**, etc..



Statistical Graphics(SG) プロシージャの種類

- sgplot プロシージャ ← 本日はこの解説
 - 様々なグラフを作成する(メインのプロシージャ)
- sgpanel プロシージャ
 - 層ごとに分割してグラフを作成(種類は sgplot プロシージャとほぼ同じ)
- sgpie プロシージャ
 - 凝った円グラフを描く
- sgscatter プロシージャ
 - 散布図行列を作成する
- sgrender プロシージャ
 - GTL(Graphical Template Language)でカスタマイズしたグラフを作成する
- sgdesign プロシージャ
 - ODS グラフエディタでカスタマイズしたグラフを作成する



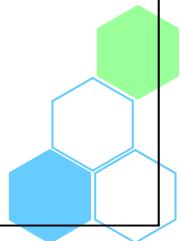
使用するデータ①: *ToothGrowth*

- モルモットにビタミン C 又はオレンジジュースを与えた時の歯の長さを調べたデータ
 - len**: 長さ(mm)
 - supp**: サプリの種類(VC(ビタミンC) 又は OJ(オレンジジュース))
 - dose**: 用量(0.5mg, 1.0mg, 2.0mg)

```
data ToothGrowth ;
input len supp$ dose ;
cards ;
  4.2  VC  0.5
  11.5 VC  0.5
  7.3   VC  0.5
  5.8   VC  0.5
  6.4   VC  0.5
  10.0  VC  0.5
  11.2  VC  0.5
  11.2  VC  0.5
  5.2   VC  0.5
  7.0   VC  0.5
  16.5  VC  1.0
  16.5  VC  1.0
  15.2  VC  1.0
  17.3  VC  1.0
  22.5  VC  1.0
  17.3  VC  1.0
  13.6  VC  1.0
  14.5  VC  1.0
  18.8  VC  1.0
```

15.5	VC	1.0
23.6	VC	2.0
18.5	VC	2.0
33.9	VC	2.0
25.5	VC	2.0
26.4	VC	2.0
32.5	VC	2.0
26.7	VC	2.0
21.5	VC	2.0
23.3	VC	2.0
29.5	VC	2.0
15.2	OJ	0.5
21.5	OJ	0.5
17.6	OJ	0.5
9.7	OJ	0.5
14.5	OJ	0.5
10.0	OJ	0.5
8.2	OJ	0.5
9.4	OJ	0.5
16.5	OJ	0.5
9.7	OJ	0.5
19.7	OJ	1.0

```
23.3  OJ  1.0
23.6  OJ  1.0
26.4  OJ  1.0
20.0  OJ  1.0
25.2  OJ  1.0
25.8  OJ  1.0
21.2  OJ  1.0
14.5  OJ  1.0
27.3  OJ  1.0
25.5  OJ  2.0
26.4  OJ  2.0
22.4  OJ  2.0
24.5  OJ  2.0
24.8  OJ  2.0
30.9  OJ  2.0
26.4  OJ  2.0
27.3  OJ  2.0
29.4  OJ  2.0
23.0  OJ  2.0
;
run ;
```



使用するデータ②: *sashelp.iris*

SepalLength	SepalWidth	PetalLength	PetalWidth	Species
5.1	3.5	1.4	0.2	Setosa
4.9	3.0	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
5.0	3.6	1.4	0.2	Setosa
5.4	3.9	1.7	0.4	Setosa
4.6	3.4	1.4	0.3	Setosa
...

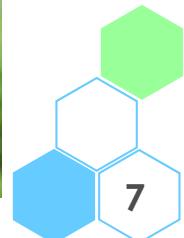
- フィッシャーが判別分析法を紹介するために利用したアヤメの品種分類（変数 Species : Setosa、 Versicolor、 Virginica）に関するデータ

⇒ 以下の 4 変数を説明変数としてアヤメの種類を判別しようとした

- SepalLength : アヤメのがくの長さ
- SepalWidth : アヤメのがくの幅
- PetalLength : アヤメの花弁の長さ
- PetalWidth : アヤメの花弁の幅



Graphic by (c)Tomo.Yun (<http://www.yunphoto.net>)



使用するデータ③: *sashelp.bmt*

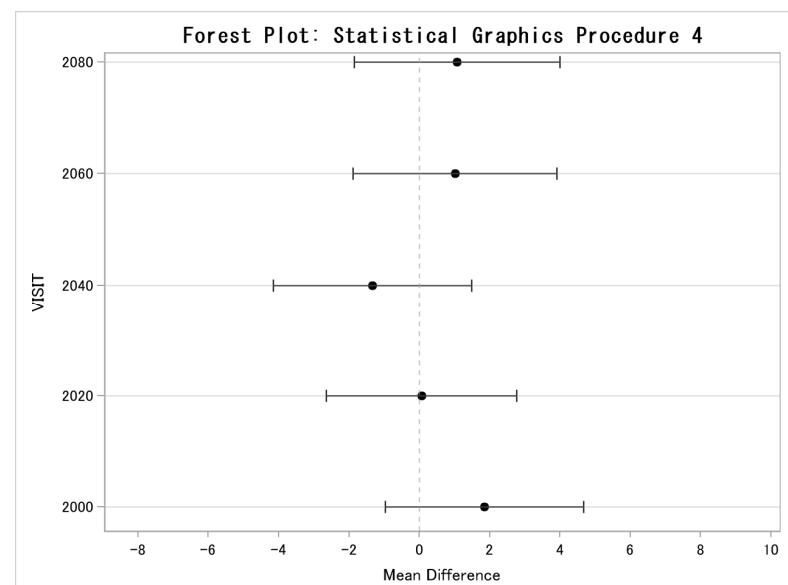
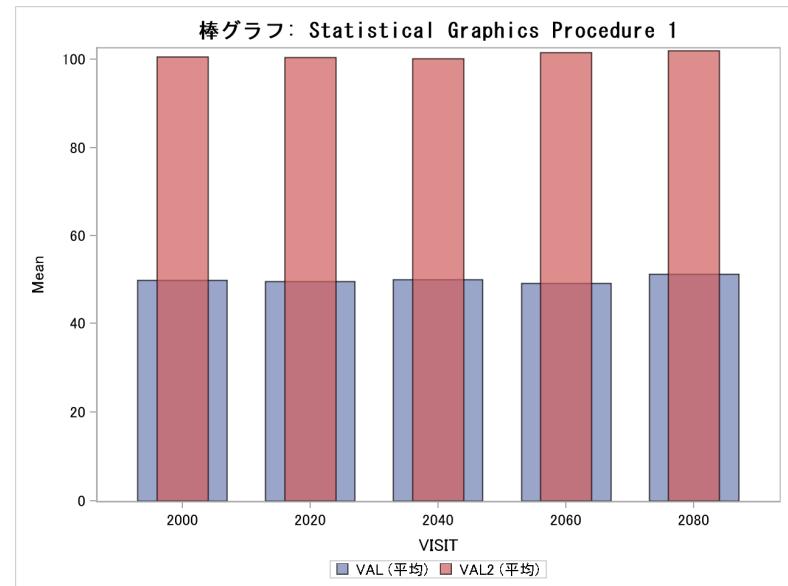
Group	T	Status
ALL	2081	0
ALL	1602	0
ALL	1496	0
ALL	1462	0
ALL	1433	0
ALL	1377	0
ALL	1330	0
...

- Klein and Moeschberger (1997) が生存時間解析の手法の紹介の際に用いた造血幹細胞移植 (Bone Marrow Transplant) に関するデータ
 - Group : 群 (ALL、 AML-Low Risk、 AML-High Risk)
 - T : 生存時間 (日)
 - Status : イベントに関する変数 (1 : イベント、 0 : 打ち切り)

sgplot プロジェクション事始

```
*** 主なステートメント ;
proc sgplot ;
  <graphs> ;
  xaxis ;
  yaxis ;
  refline ;
  keylegend ;
quit ;
```

- <graphs>: グラフの種類や各種定義
- xaxis: 横軸の定義
- yaxis: 縦軸の定義
- refline: 参照線の定義
- keylegend: 凡例の定義

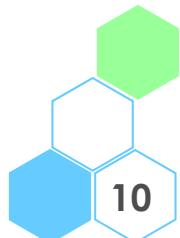


sgplot プロジェクション事始

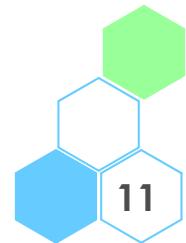
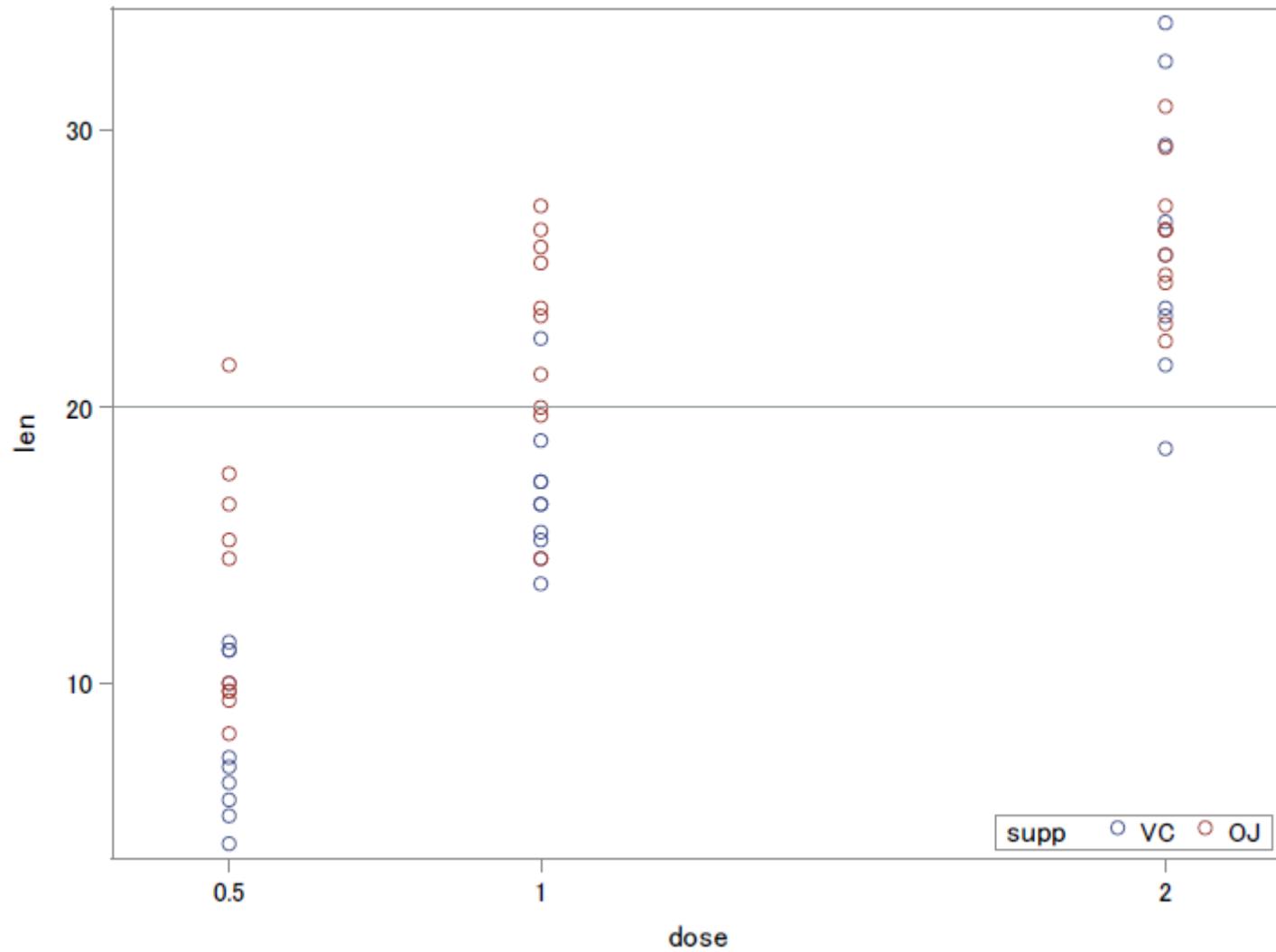
```
*==== 初期設定 ;
ods listing image_dpi=300 ;
ods graphics / reset height=450px width=450px ;

proc sgplot data=ToothGrowth ;
  scatter    x=dose y=len / group=supp ;
  xaxis      values=(0.5 1 2) min=0.4 max=2.1
              offsetmin=0.1 offsetmax=0.1 ;
  refline   20 / axis=y lineattrs=(color=black pattern=1) ;
  keylegend / location=inside position=bottomright ;
run ;
quit ;
```

- グラフ:**scatter**(散布図)、x 軸は dose、y 軸は len、群は supp
- **xaxis**(横軸):見やすくなるように設定
- **refline**(参照線):y=20 の黒色直線を追記
- **keylegend**(凡例):右下に表示

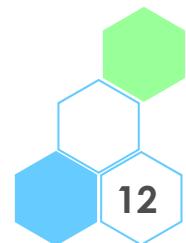


散布図が完成



Menu

- 事始
- 種々のグラフ
- グラフのカスタマイズ
- 応用例: Forest Plot
- 応用例: Kaplan-Meier Plot



sgplot プロシージャで描けるグラフ①

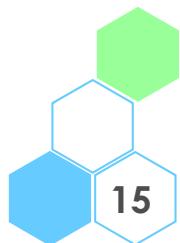
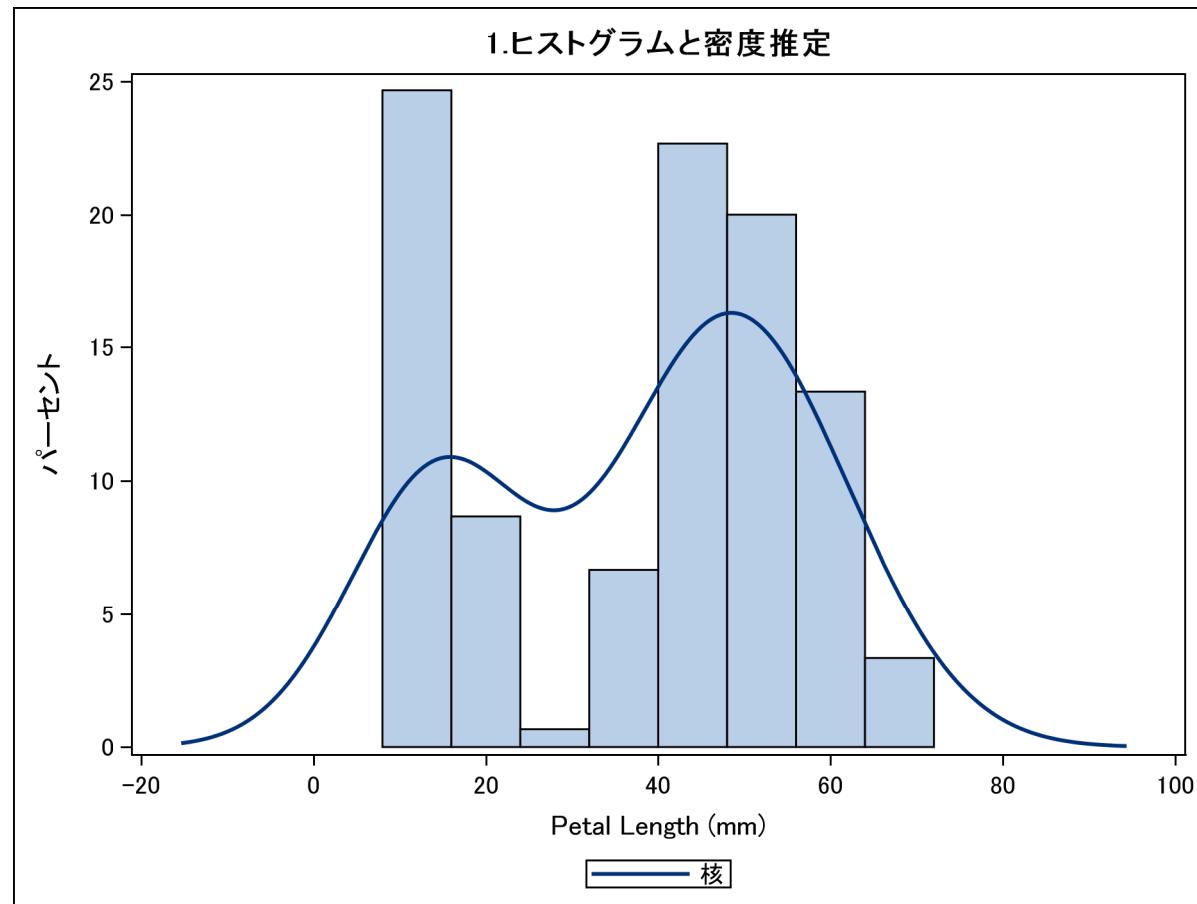
グラフの種類	ステートメント
バンド幅	<code>band</code> x=variable y=variable upper= numeric-value numeric-variable lower= numeric-value numeric-variable ... ;
Block Plot	<code>block</code> x=category-variable block=block-variable ... ;
バブルプロット	<code>bubble</code> x=variable y=variable size=numeric-variable ... ;
密度推定曲線	<code>density</code> response-variable ... ;
ドットプロット	<code>dot</code> category-variable ... ;
Drop Line	<code>dropline</code> x=variable x-axis-value y=variable y-axis-value ... ;
橢円曲線	<code>ellipse</code> x=numeric-variable y=numeric-variable ... ; <code>ellipseparm</code> semimajor=numeric-variable positive-number semiminor=numeric-variable positive-number ... ;
Fringe Plot	<code>fringe</code> numeric-variable ... ;
Gradientな凡例	<code>gradlegend</code> <"name"> ... ;
棒グラフ(横)	<code>hbar</code> category-variable ... ; <code>hbarbasic</code> category-variable ... ; <code>hbarparm</code> category=category-variable response=numeric-variable ... ;
箱ひげ図(横)	<code>hbox</code> numeric-analysis-variable ... ;
ヒートマップ	<code>heatmap</code> x=variable y=variable ... ; <code>heatmapparm</code> x=variable y=variable colorgroup=variable <code>heatmapparm</code> x=variable y=variable colorresponse=numeric-variable ... ;
High-Low Plot	<code>highlow</code> x=variable y=variable high=numeric-variable low=numeric-variable ... ;
ヒストグラム	<code>histogram</code> response-variable ... ;
折れ線グラフ	<code>hline</code> category-variable ... ;
グラフ中に文字 凡例	<code>inset</code> "text-string" < ..."text-string-n"> (label-list) ... ; <code>keylegend</code> <"name-1" ... "name-n"> ... ;
複数の図の凡例	<code>legenditem</code> type=type name="text-string" ... ;
loess 曲線	<code>loess</code> x=numeric-variable y=numeric-variable ... ;

sgplot プロシージャで描けるグラフ②

グラフの種類	ステートメント
直線	<code>lineparm</code> x=numeric-value numeric-variable y=numeric-value numeric-variable slope=numeric-value numeric-variable... ;
Needle Plot	<code>needle</code> x=variable y=numeric-variable ... ;
罰則付B-spline 曲線	<code>pbspline</code> x=numeric-variable y=numeric-variable ... ;
ポリゴン	<code>polygon</code> x=x-variable y=y-variable id=id-variable ... ;
参照線	<code>refline</code> variable value(s) ... ;
回帰直線	<code>reg</code> x=numeric-variable y=numeric-variable ... ;
散布図	<code>scatter</code> x=variable y=variable ... ;
時系列プロット	<code>series</code> x=variable y=variable ... ;
スプライン曲線	<code>spline</code> x=variable y=variable ... ;
階段関数のプロット	<code>step</code> x=variable y=variable ... ;
文字のプロット	<code>symbolchar</code> name=identifier char="hex-string" keyword ... ;
図のプロット	<code>symbolimage</code> name=identifier image="image-file-specification" ... ;
図の中に文字入力	<code>text</code> x=variable y=variable text=variable ... ;
棒グラフ(縦)	<code>vbar</code> category-variable ... ; <code>vbarbasic</code> category-variable ... ; <code>vbarparm</code> category=category-variable response=numeric-variable ... ;
箱ひげ図(縦)	<code>vbox</code> numeric-analysis-variable ... ;
Vector Plot	<code>vector</code> x=numeric-variable y=numeric-variable ... ;
折れ線グラフ	<code>vline</code> category-variable ... ;
Waterfall プロット	<code>waterfall</code> category=variable response=numeric-variable ... ;
表の挿入	<code>xaxistable</code> variable ... ; <code>yaxistable</code> variable ... ;

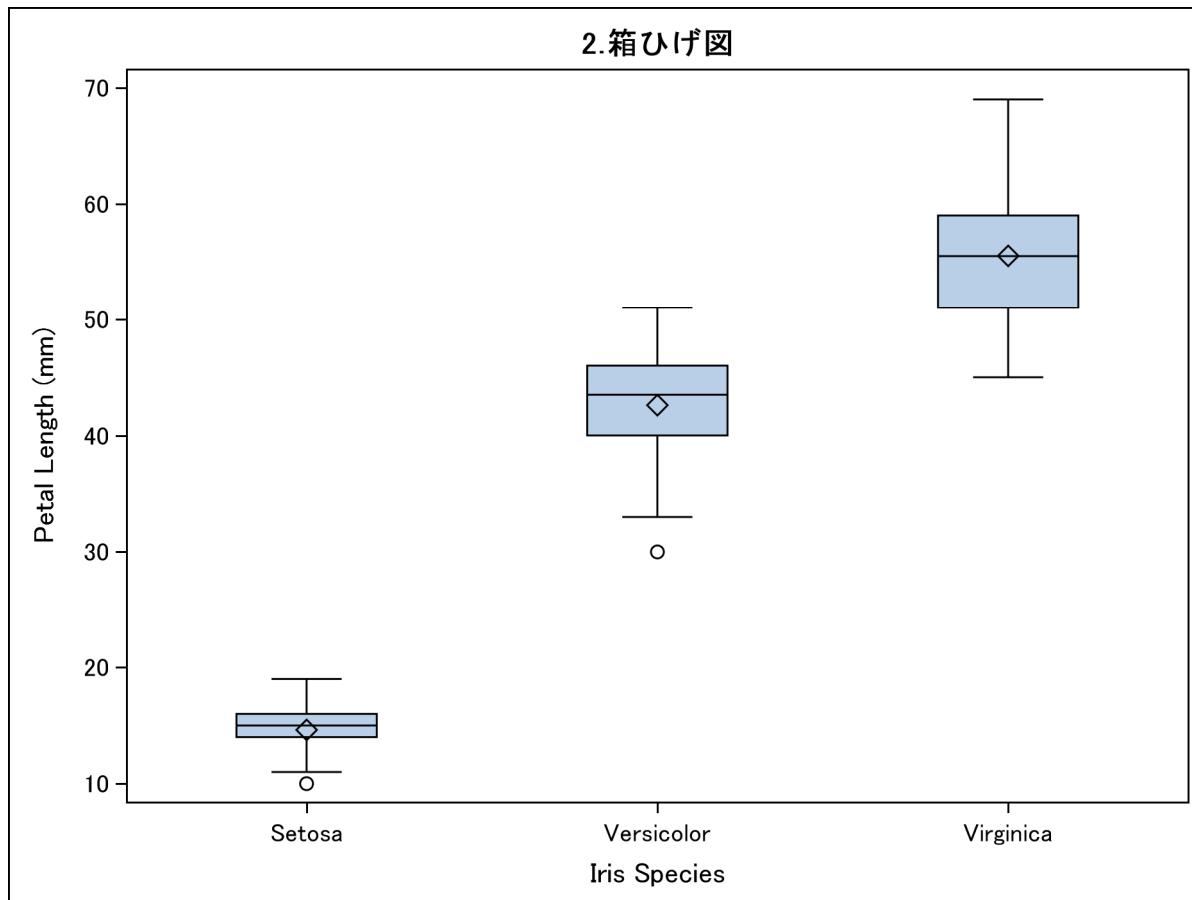
例1: ヒストグラムと密度推定

```
title "1.ヒストグラムと密度推定" ;  
proc sgplot data=sashelp.iris ;  
    histogram PetalLength ;  
    density PetalLength / type=kernel ;  
run ;
```



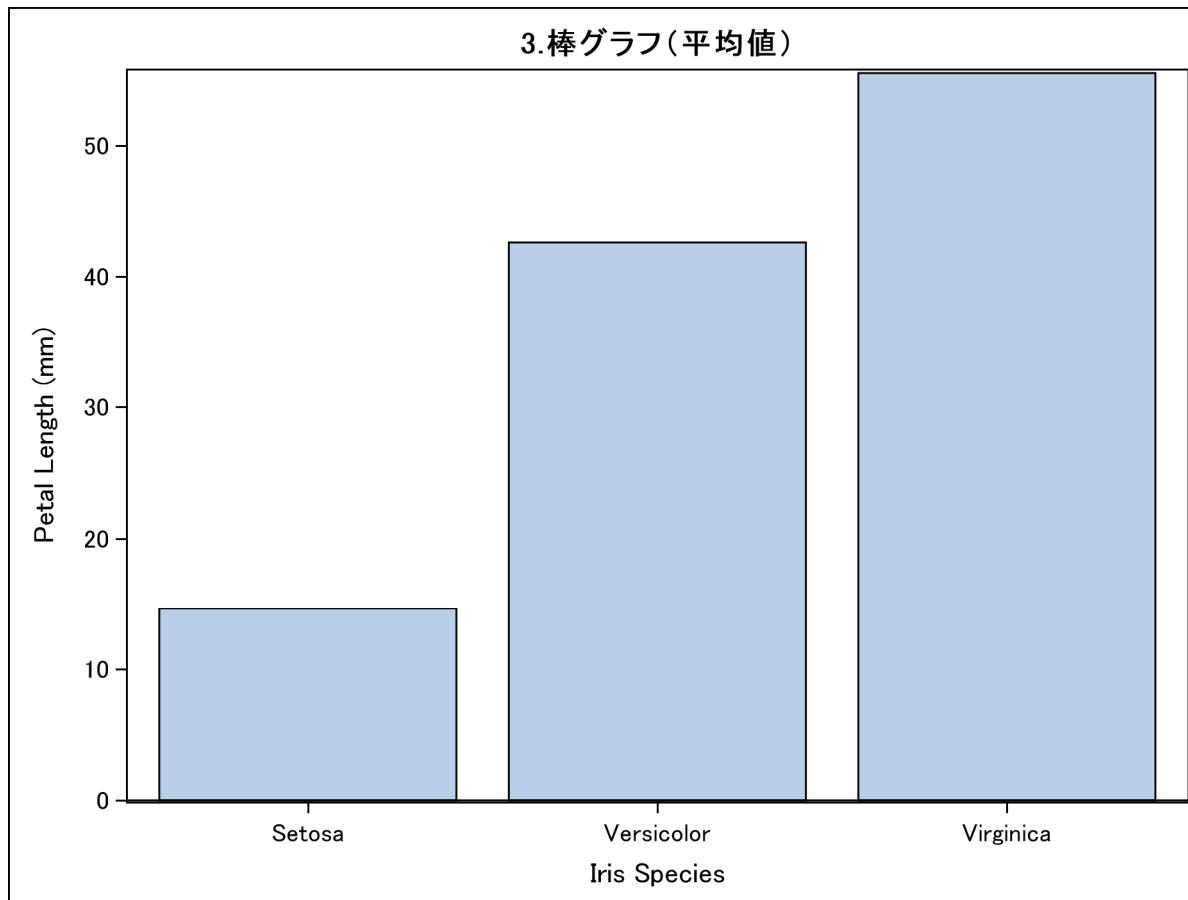
例2: 箱ひげ図

```
title "2.箱ひげ図" ;  
proc sgplot data=sashelp.iris ;  
    vbox PetalLength / category=Species ;  
    *--- vbox を hbox にすると横向きになる ;  
run ;
```



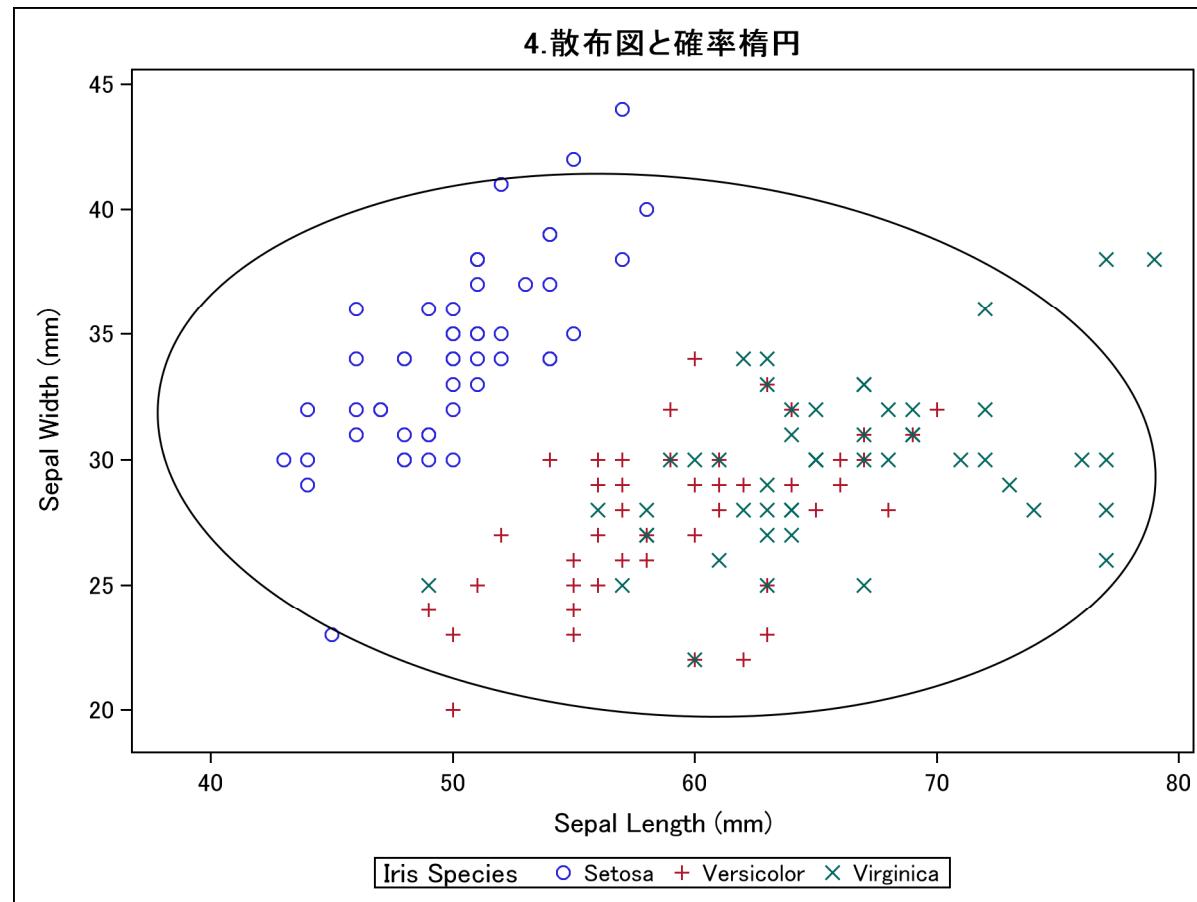
例3: 棒グラフ (平均値)

```
title "3.棒グラフ(平均値)" ;  
proc sgplot data=sashelp.iris ;  
    vbar Species / response=PetalLength stat=mean ;  
    *--- vxxx を hxxx にすると横向きになる;  
run ;
```



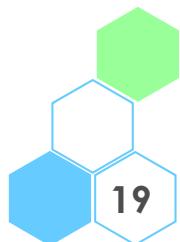
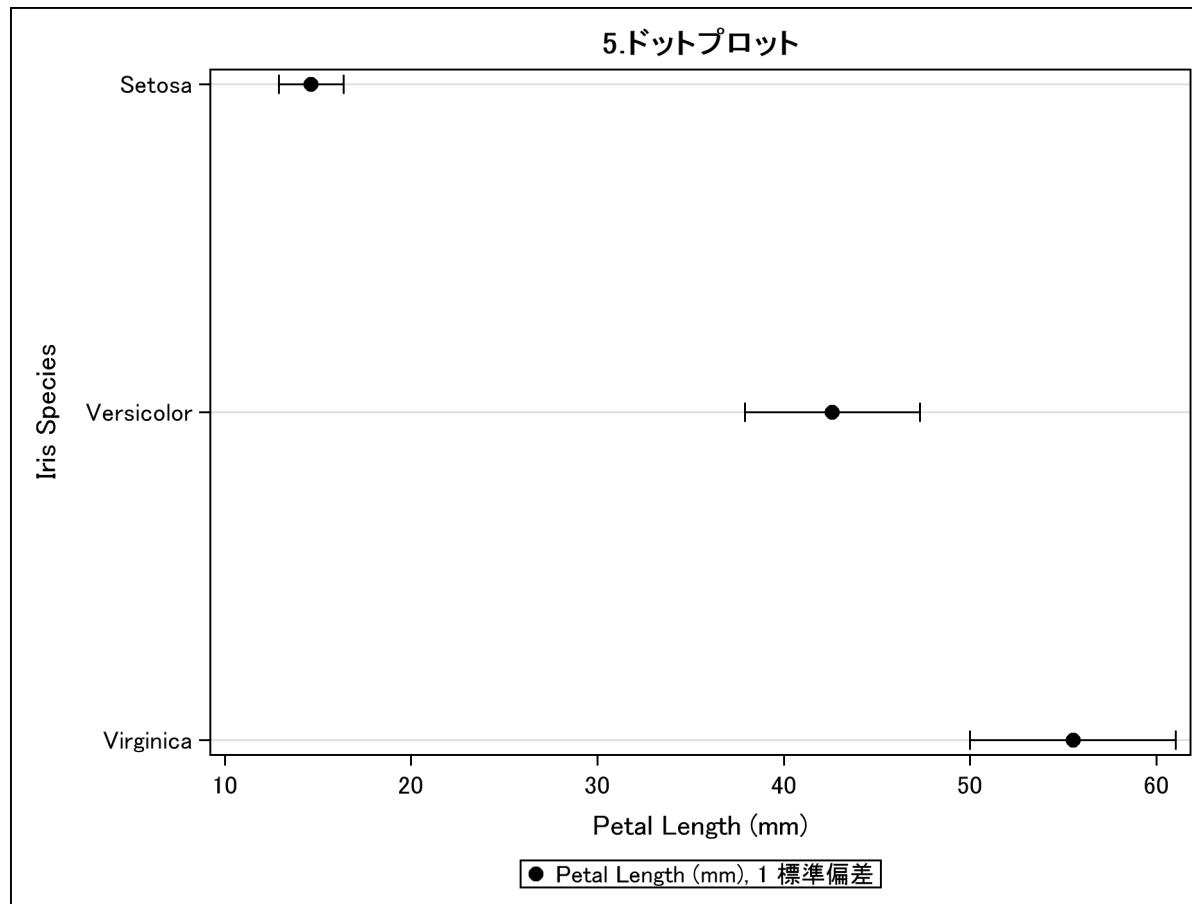
例4: 散布図と確率楕円

```
title "4.散布図と確率楕円" ;  
proc sgplot data=sashelp.iris ;  
    scatter x=SepalLength y=SepalWidth / group=Species ;  
    ellipse x=SepalLength y=SepalWidth ;  
run ;
```



例5: ドットプロット

```
title "5. ドットプロット" ;
proc sgplot data=sashelp.iris ;
    dot Species / response=PetalLength
        stat=mean limitstat=stddev ;
run ;
```

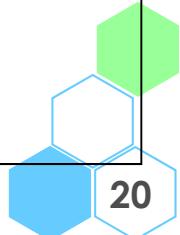


例6: Waterfall Plot (抗がん剤の試験用)

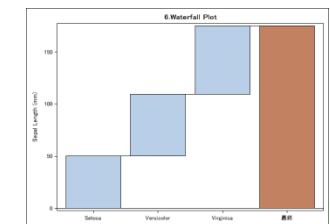
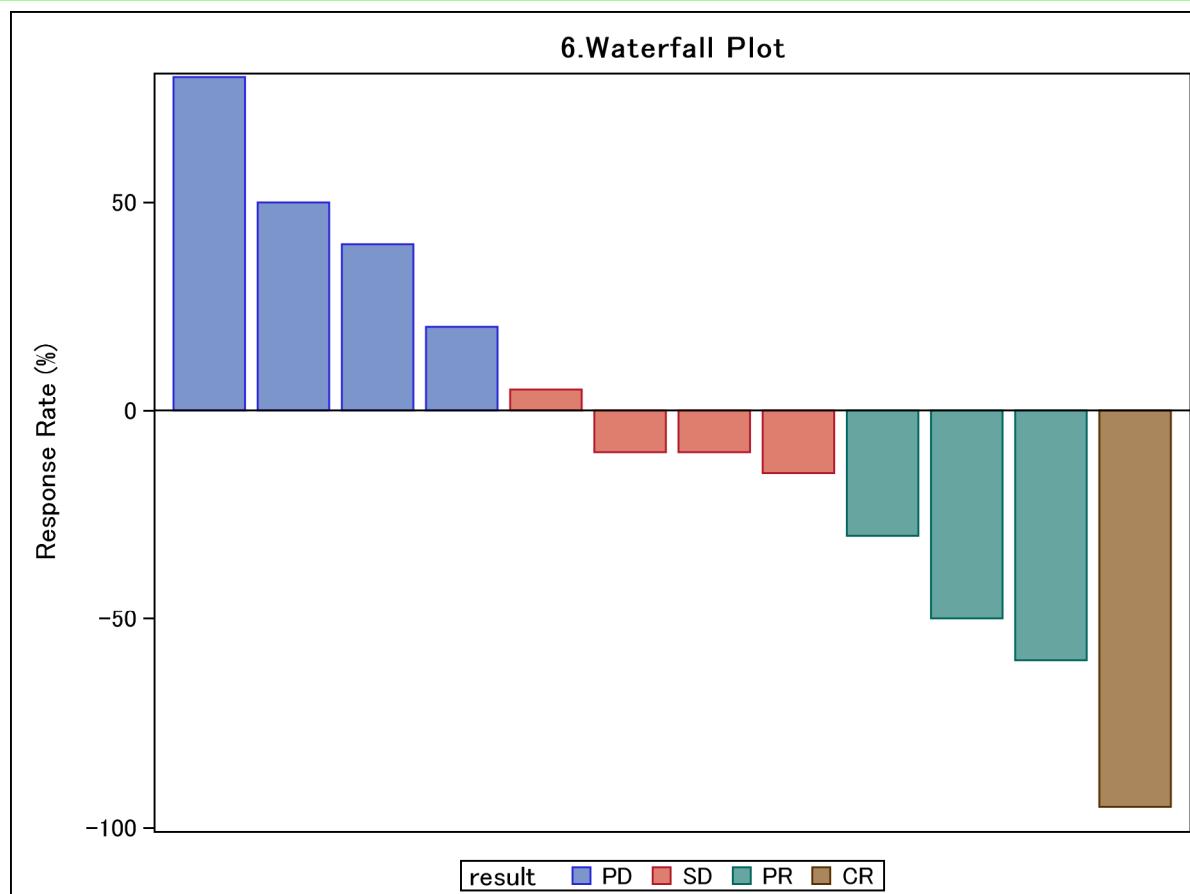
```
title "6.Waterfall Plot" ;
data ResponseRate ;
  input id rrate result$ @@ ;
  cards ;
    1  80 PD    2   50 PD    3   40 PD    4   20 PD    5    5 SD    6  -10 SD
    7  -10 SD   8  -15 SD   9  -30 PR   10  -50 PR   11  -60 PR   12  -95 CR
  ;
run ;

proc sgplot data=ResponseRate ;
  vbar id / response=rrate group=result ;
  xaxis display=none ;
  yaxis label='Response Rate (%)' ;
run ;

/* こういう Waterfall Plot は想定しておりません。
proc sgplot data=sashelp.iris ; * Waterfall Chart for Business
  Setting ;
  waterfall category=Species response=SepalLength / stat=mean ;
run;
*/
```



例6: Waterfall Plot (抗がん剤の試験用)



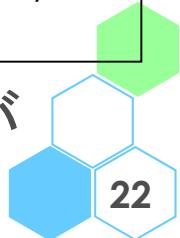
こういうWaterfall Plot
は意図しておりません。

例7: 数学関数のプロット

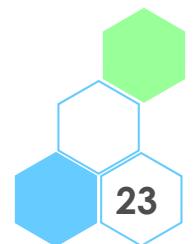
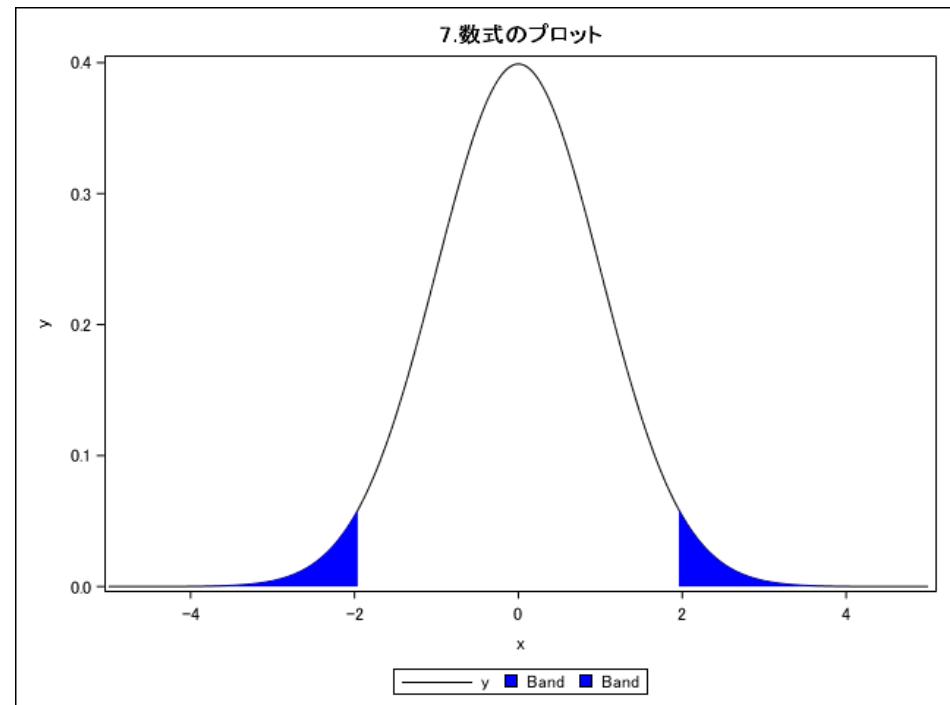
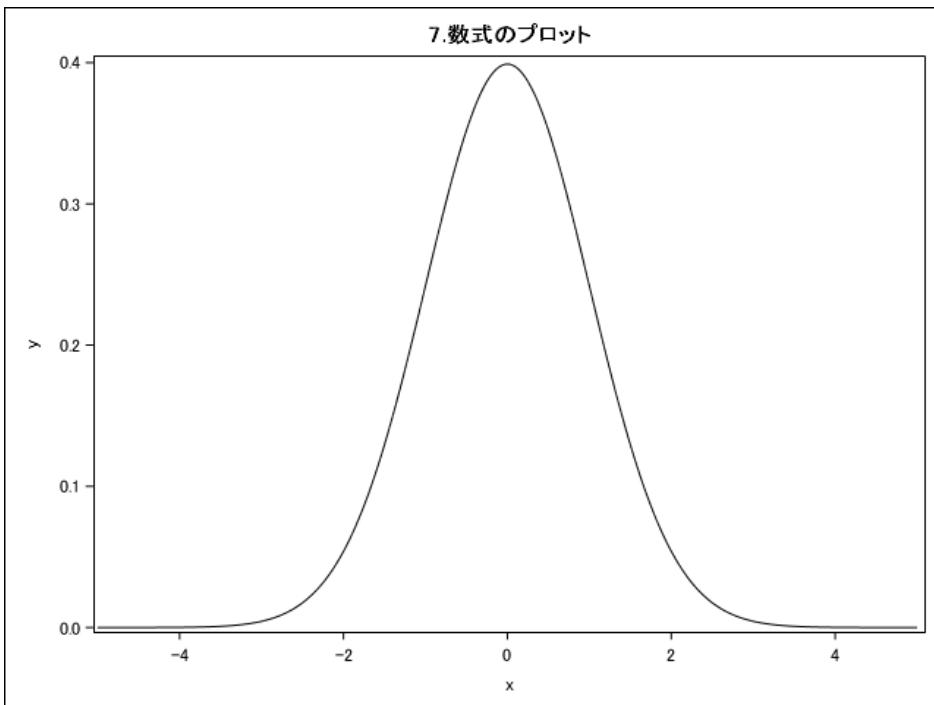
```
data test ;
  do x=-5 to 5 by 0.01 ; y=pdf("normal",x,0,1) ; output ; end ;
run ;
proc sgplot data=test ;
  series x=x y=y ;
run ;

data test ;
  low  = quantile("normal",0.025) ; *--- 2.5%点 ;
  high = quantile("normal",0.975) ; *--- 97.5%点 ;
  do x=-5 to 5 by 0.01 ;
    y=pdf("normal",x,0,1) ;           * 標準正規分布の確率密度関数 ;
    if x <= low then a1=y ; else a1=. ; * 2.5%点以下のみ描画 ;
    if x >= high then a2=y ; else a2=. ; * 97.5%点以上のみ描画 ;
    output ;
  end ;
run ;
proc sgplot data=test ;
  series x=x y=y ;           *--- 標準正規分布の確率密度関数描画 ;
  band x=x upper=a1 lower=0 / fillattrs=(color=blue) ; *--- 2.5%点以下を描画 ;
  band x=x upper=a2 lower=0 / fillattrs=(color=blue) ; *--- 97.5%点以上を描画 ;
run ;
```

- 関数作成用のデータを細かい刻み幅ごとに生成する必要があるので若干大変



例7: 数学関数のプロット



Menu

- 事始
- 種々のグラフ
- グラフのカスタマイズ
- 応用例: Forest Plot
- 応用例: Kaplan–Meier Plot

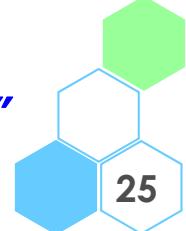


平均値の推移図

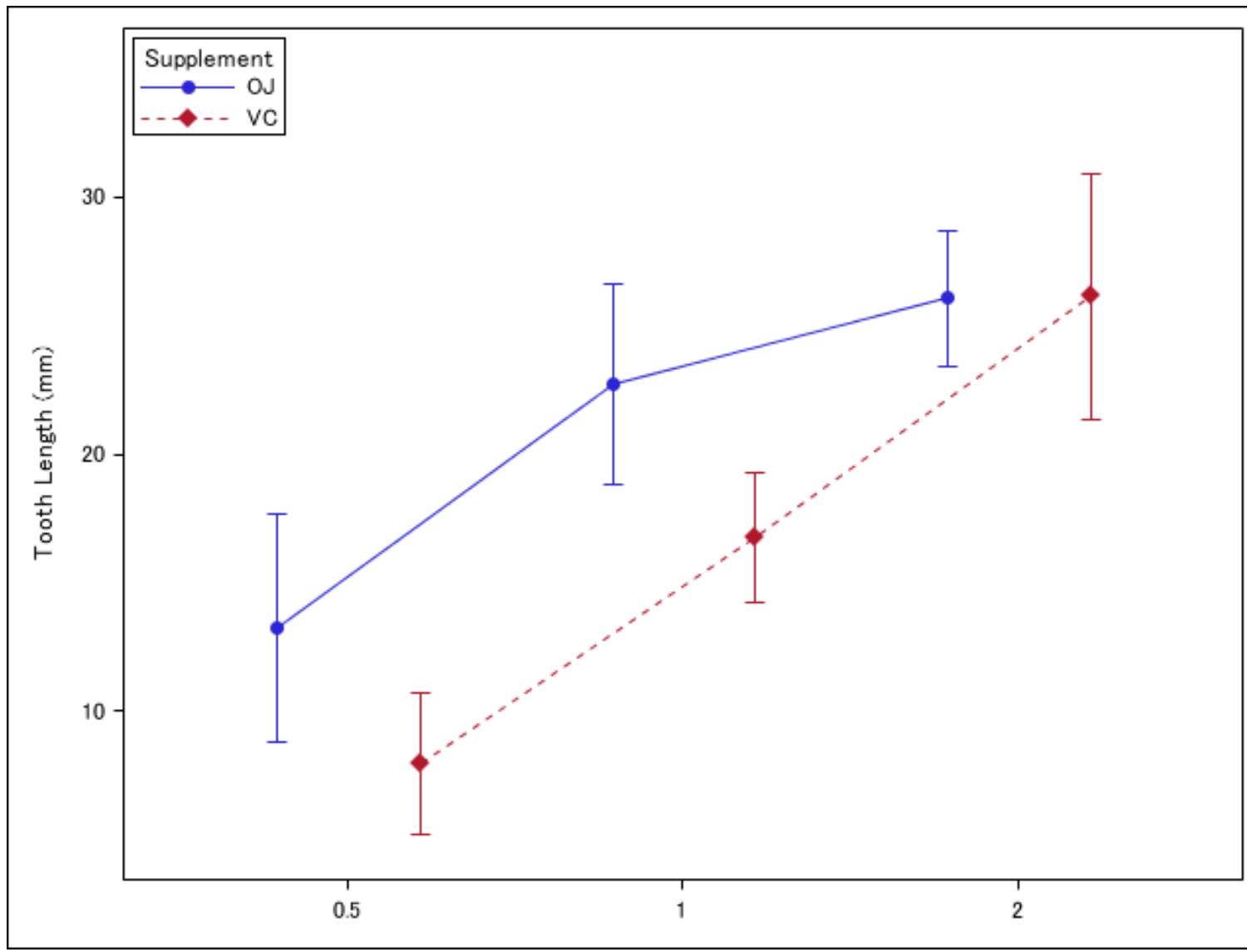
```
proc format ;
  value dosef 0.5="0.5 mg" 1="1.0 mg" 2="2.0 mg" ;
run ;

proc sgplot data=ToothGrowth ;
  styleattrs
    datacolors      =(blue red)
    datalinepatterns=(1 2)
    datasymbols     =(circlefilled diamondfilled) ;
  vline dose / response   =len      group =supp
          groupdisplay=cluster  stat  =mean
          limitstat   =stddev   limits=both
          markers ;
  xaxis      type=discrete offsetmin=0.2 offsetmax=0.2
             display=(nolabel) tickvalueformat=dosef. ;
  yaxis      values=(10 20 30) min=0 max=40 offsetmin=0.2 offsetmax=0.2
             label="Tooth Length (mm)" ;
  keylegend / down=2 location=inside position=topleft title="Supplement" ;
run ;
```

- **vline** ステートメントで平均値の推移図が描ける
- 指定すべきオプションは “**SAS 9.4 ODS Graphics: Procedures Guide**” を参照のこと



平均値の推移図



平均値の推移図

```
proc format ;
  value dosef 0.5="0.5 mg" 1="1.0 mg" 2="2.0 mg" ;
run ;

proc sgplot data=ToothGrowth ;
  styleattrs
    datacolors      = (blue red)
    datalinepatterns=(1 2)
    datasymbols     = (circlefilled diamondfilled) ;
  vline dose / response      =len      group =supp
          groupdisplay=cluster   stat   =mean
          limitstat    =stddev   limits=both
          markers ;
  xaxis      type=discrete offsetmin=0.2 offsetmax=0.2
            display=(nolabel) tickvalueformat=dosef. ;
  yaxis      values=(10 20 30) min=0 max=40 offsetmin=0.2 offsetmax=0.2
            label="Tooth Length (mm)" ;
  keylegend / down=2 location=inside position=topleft title="Supplement" ;
run ;
```

- これまで、群ごとにプロットの見た目を変更するのは大変だったが
SAS 9.4 より **styleattrs** ステートメントが追加された



styleattrs ステートメント

オプション	機能
<code>axisbreak=bracket notch spark z slantedleft slantedright squiggle </code>	軸の省略記号の種類
<code>axisextent=full data</code>	x 軸と y 軸を結合する／しない
<code>backcolor=色</code>	グラフの背景色を指定
<code>datacolors=(色1 色2 ...)</code>	群ごとにグラフの要素（点や線）の色を指定
<code>datacontrastcolors=(色1 色2 ...)</code>	datacolors と同様だが色が鮮やか？
<code>datafillpatterns=(ラベル1 ラベル2)</code>	棒などの塗りつぶしの模様を指定
<code>datalinepatterns=(線種1 線種2 ...)</code>	群ごとに線の種類を指定
<code>datasymbols=(点種1 点種2 ...)</code>	群ごとに点の種類を指定
<code>wallcolor=色</code>	グラフの壁紙部分の色を指定

- 色は英語名を指定
 - BLACK、WHITE、RED、GREEN、BLUE、PURPLE、VIOLET、ORANGE、YELLOW、PINK、CYAN、MAGENTA、BROWN、GOLD、LIME、GRAY



点の種類

点	ラベル	点	ラベル	点	ラベル	点	ラベル
↓	ArrowDown	I	Ibeam	<	TriangleLeft	▼	HomeDownFilled
*	Asterisk	+	Plus	>	TriangleRight	■	SquareFilled
○	Circle	□	Square	∪	Union	★	StarFilled
◇	Diamond	☆	Star	×	X	▲	TriangleFilled
>	GreaterThan	T	Tack	Y	Y	▽	TriangleDownFilled
<	LessThan	~	Tilde	Z	Z	◀	TriangleLeftFilled
#	Hash	△	Triangle	●	CircleFilled	▶	TriangleRightFilled
盾	HomeDown	▽	TriangleDown	◆	DiamondFilled		



模様(*datafillpatterns*)の種類

点	模様	点	模様	点	模様
L1		R1		X1	
L2		R2		X2	
L3		R3		X3	
L4		R4		X4	
L5		R5		X5	

線の種類

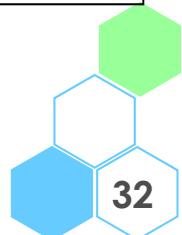
番号	キーワード	線
1	Solid	——
2	ShortDash	- - - - -
4	MediumDash	- - - - - - - - - -
5	LongDash	- - - - - - - - - - - -
8	MediumDashShortDash	- - - - - - - - - - - - -
14	DashDashDot	- - - - - - - - - - - - - -
15	DashDotDot	- - - - - - - - - - - - - -
20	Dash	- - - - - - - - - - - - - -
26	LongDashShortDash	- - - - - - - - - - - - - - -
34	Dot
35	ThinDot
41	ShortDashDot	- - - - - - - - - - - - - -
42	MediumDashDotDot	- - - - - - - - - - - - - -

平均値の推移図

```
proc format ;
  value dosef 0.5="0.5 mg" 1="1.0 mg" 2="2.0 mg" ;
run ;

proc sgplot data=ToothGrowth ;
  styleattrs
    datacolors      = (blue red)
    datalinepatterns=(1 2)
    datasymbols     = (circlefilled diamondfilled) ;
  vline dose / response   =len      group =supp
          groupdisplay=cluster  stat   =mean
          limitstat   =stddev   limits=both
          markers ;
  xaxis      type=discrete offsetmin=0.2 offsetmax=0.2
             display=(nolabel) tickvalueformat=dosef. ;
  yaxis      values=(10 20 30) min=0 max=40 offsetmin=0.2 offsetmax=0.2
             label="Tooth Length (mm)" ;
  keylegend / down=2 location=inside position=topleft title="Supplement" ;
run ;
```

- x 軸、y 軸、凡例については、それぞれ `xaxis`、`yaxis`、`keylegend` ステートメントで調整出来る



xaxis と *yaxis* のオプション①

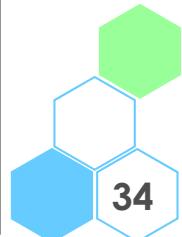
オプション	機能
<code>colorbands=none even odd</code> <code>colorbandsattr=(...)</code>	色のついた帯を追記 (Forest Plot 作成時などに使用)
<code>display=all, none,</code> <code>(nolabel noline noticks novalues)</code>	軸を全て表示する/表示しない 軸ラベル, 軸, 目盛, 値を表示しない
<code>grid</code> <code>gridatrrs=(...)</code>	グリッド線を追記
<code>logbase=2, 10, e</code> <code>logstyle=linear, logexpand, logexponent</code>	対数表示をする際、対数の底を指定 対数表示の間隔の尺度を指定
<code>min=0, max=40</code>	表示する下限値と上限値を指定
<code>offsetmin=0.2, offsetmax=0.2</code>	下限側と上限側のオフセットを指定
<code>ranges=(開始1-終了1 開始2-終了2 ...)</code>	軸の途中で省略する際の範囲を指定
<code>reverse</code>	座標の向きを逆にする
<code>tickstyle=outside inside across inbetween</code>	軸のメモリの場所を指定
<code>type=discrete, linear, log, time</code>	データの型を指定
<code>interval=auto second minute hour</code>	時間データの場合の単位を指定

xaxis と *yaxis* のオプション②

オプション	機能
label =“ラベル” labelattrs =(...)	軸ラベルを指定
labelpos =center datacenter left right	軸ラベルの位置を指定
values =(10 20 30) valueattrs =(...)	軸に表示したい値を指定
values =("1" "2") valuesdisplay =("one" "two")	軸のラベルをデータの値ごとに変更 (valuesdisplay でのラベルを表示)
valuesrotate =diagonal diagonal2 vertical	軸ラベルを回転する
valueformat =dosef. tickvalueformat =dosef.	軸にフォーマットを当てる

- 2 軸のグラフ作成時に使用する **x2axis** / **y2axis** ステートメントも同様のオプションが適用出来る

```
proc sgplot data=sashelp.iris ;  
  scatter x=SepalLength y=SepalWidth / group=Species ;  
  ellipse x=SepalLength y=SepalWidth / x2axis y2axis ;  
  y2axis min=0 max=60 ;  
run ;
```



keylegend のオプション

オプション	機能
<code>across=m down=n</code>	凡例の列数(m列)と行数(n行)を指定
<code>border noborder</code>	凡例の外枠を描く／描かない
<code>location=outside inside</code>	凡例を図の外側/内側に描く
<code>opaque noopaque</code>	凡例を不透明に／透明にする
<code>outerpad=(left=9pt top=8pt right=7pt bottom=)</code>	凡例枠の内側余白を指定
<code>position=bottom bottomleft bottomright left right top topleft topright</code>	凡例を描く場所を指定
<code>sortorder=ascending descending ...</code>	凡例のラベルの順序を指定
<code>type=fill fillcolor line linecolor linepattern marker markercolor markersymbol</code>	凡例にどの記号を表示するか指定
<code>title="ラベル"</code> <code>titleattrs=(...)</code>	凡例のタイトルを指定
<code>valueattrs=(color:red family=Arial size=9 style=Italic weight=Bold ...)</code>	凡例の書式を指定



xaxistable / *yaxistable* で表をグラフへ追記

オプション	機能
<code>class</code> =カテゴリ変数 <code>classdisplay</code> =stack cluster <code>classorder</code> =data reversedata ascending descending	カテゴリ変数ごとに表を作成
<code>nomissingclass</code>	カテゴリ内に値が無い場合は表示しない
<code>indent</code> =単位 <code>indentweight</code> =幅に関する変数	ラベルを表示する際にインデントを付ける
<code>label</code> <code>nolabel</code> <code>label</code> =“text-string” <code>labelattrs</code> =(...) <code>labelpos</code> =left right	表のラベルの表示の有無とその書式を指定
<code>location</code> =outside inside	表をグラフの外／中に表示する
<code>pad</code> =(left=9pt top=8pt right=7pt bottom=6pt)	表内の余白を指定
<code>position</code> =bottom top	表の位置を指定
<code>separator</code>	罫線を表示する
<code>title</code> =“ラベル” <code>titleattrs</code> =(...)	表のタイトルを指定
<code>valueattrs</code> =(color:red size=9 ...)	表の書式を指定
<code>x</code> =変数 <code>y</code> =変数	グラフと紐付ける x 軸／y 軸変数を指定
<code>x2axis</code> <code>y2axis</code>	2 軸目の方に表を追記

※ 適用例は「応用例：Forest Plot」を参照

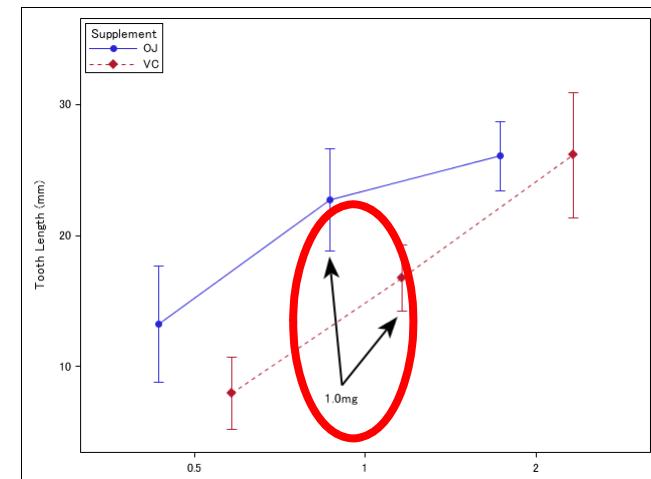


図形や文字をグラフへ追記

```
data Line ;
  input function $ x1 y1 x2 y2 shape $ direction $ label $ ;
  cards ;
    arrow 50 20 48 48 barbed out .
    arrow 50 20 59 35 barbed out .
    text 50 17 . . . . . 1.0mg
  ;
  proc sgplot data=ToothGrowth sganno=Line ;
    ... ...
  ;

```

- 完成したグラフへ、さらに図形や文字を追記することも出来る
- ただ、追記する図形の位置情報や文字情報をデータセットで準備する必要があるため若干面倒

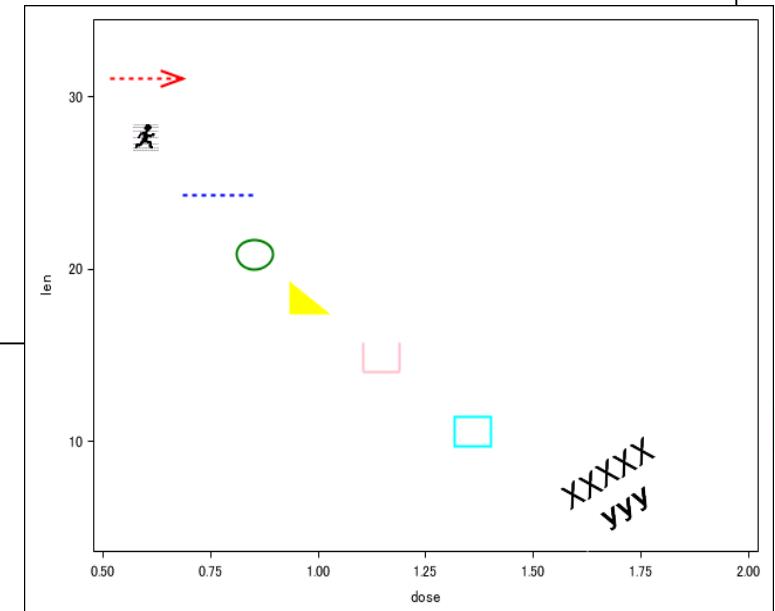


図形や文字をグラフへ追記

```
%sganno ; * 種々のマクロを呼び出し ;
data myAnno ;
  %sgarrow(x1=10, x2=20, y1=90, y2=90, linecolor="red", linepattern=2) ;
  %sgimage(x1=15, y1=80, height=5, image="C:\temp\icon.gif") ;
  %sgline (x1=20, x2=30, y1=70, y2=70, linecolor="blue", linepattern=2) ;
  %sgoval (x1=30, y1=60, height=5, width=5, linecolor="green", linepattern=1) ;
  %sgpolygon (x1=35, y1=55, fillcolor="yellow", linecolor="yellow", display="all") ;
  %sgpolycont(x1=35, y1=50) ; * (35,55)→(35,50)→(40,50) の順で多角形を描写 ;
  %sgpolycont(x1=40, y1=50) ; * 次の %sgpolyline も同様の手順でポリゴン線を描く ;
  %sgpolyline(x1=45, y1=45, linecolor="pink") ;
  %sgpolycont(x1=45, y1=40) ; %sgpolycont(x1=50, y1=40) ; %sgpolycont(x1=50, y1=45) ;
  %sgrectangle(x1=60, y1=30, height=5, width=5, linecolor="cyan", fillcolor="white") ;
  %sgtext(x1=80, y1=20, label="XXXXX", textcolor="black", textsize=20, width=15,
         rotate=30, justify="center") ;
  %sgtextcont(label="yyy", textweight="bold") ;
run ;

proc sgplot data=ToothGrowth sganno=myAnno ;
  scatter x=dose y=len / markerattrs=(color=white) ;
run ;
```

- 追記用のデータセット作成の手間をある程度減らすため、マクロが用意されている
- 詳細は %sganno_help(all); を実行すると…



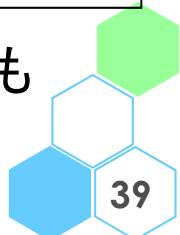
見た目の変更

```
proc template ;
  define style styles.XXX ;
    parent=styles.journal ;

/** その他、いろいろスタイルを変更することが出来る
class GraphdataDefault
  / contrastcolor=black linestyle=1 linethickness=1px ;
class Graphdata1
  / contrastcolor=blue markersymbol="trianglefilled" ;
class Graphdata2
  / contrastcolor=red foreground=black color=red ;
*/
end ;
run ;

ods listing style=styles.XXX ;
proc sgplot data=ToothGrowth;
  ...
;
```

- template プロシージャにてグラフのスタイルを変更し、見た目を変えることも出来る(詳細は高浪(2015)を参照)



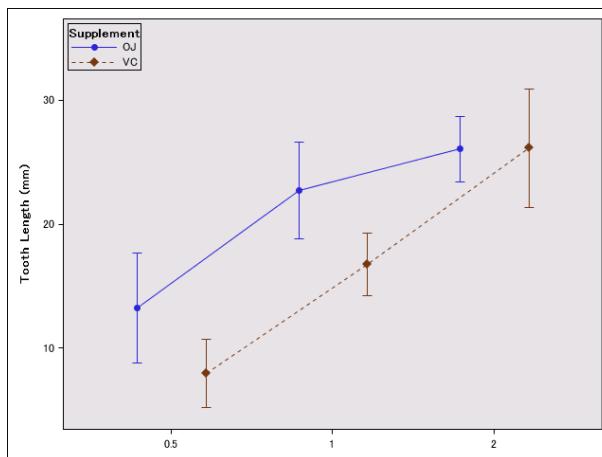
グラフのスタイル一覧

Analysis
BarrettsBlue
BlockPrint
DTree
Daisy
Default
Dove
EGDefault
FancyPrinter
Festival
FestivalPrinter
Gantt
GrayscalePrinter
HTMLBlue

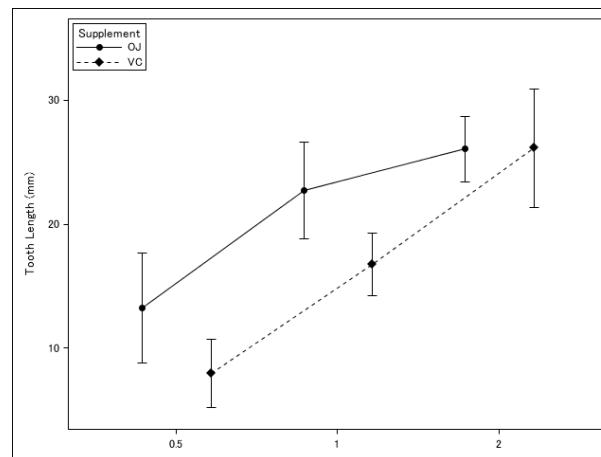
Harvest
HighContrast
HighContrastLarge
Journal
Journal1a
Journal2
Journal2a
Journal3
Journal3a
Listing
Meadow
MeadowPrinter
Minimal
MonochromePrinter

Monospace
Moonflower
Netdraw
NoFontDefault
Normal
NormalPrinter
Ocean
Pearl
PearlJ
Plateau
PowerPointDark
PowerPointLight
Printer
Raven

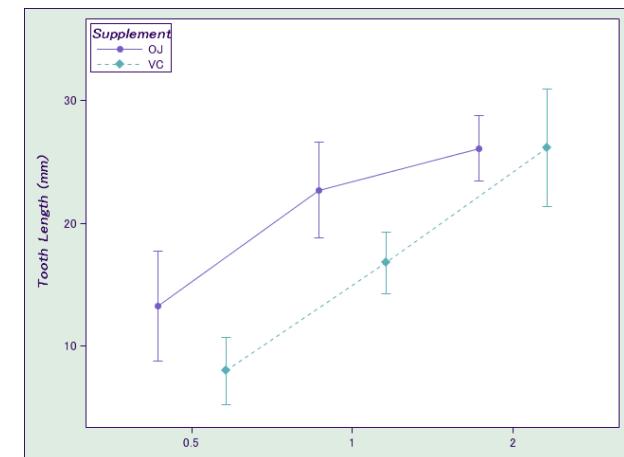
Rtf
Sapphire
SasDocPrinter
SasWeb
Seaside
SeasidePrinter
StatDoc
Statistical
vaDark
vaHighContrast
vaLight



BarrettsBlue



Journal



Ocean

Menu

- 事始
- 種々のグラフ
- グラフのカスタマイズ
- **応用例: Forest Plot**
- 応用例: Kaplan–Meier Plot

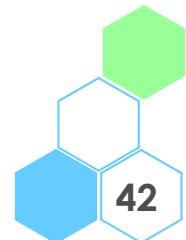


iris データからグラフ用データを生成

```
ods listing close ;
proc means data=sashelp.iris; by Species; var SepalLength; output out=TMP1 mean=M P1=L P99=U ;
proc means data=sashelp.iris; by Species; var SepalWidth ; output out=TMP2 mean=M P1=L P99=U ;
proc means data=sashelp.iris; by Species; var PetalLength; output out=TMP3 mean=M P1=L P99=U ;
proc means data=sashelp.iris; by Species; var PetalWidth ; output out=TMP4 mean=M P1=L P99=U ;
run ; ods listing ;

data MYDATA(drop=_TYPE_ Species rename=(_FREQ_=N)) ;
  set TMP1 TMP2 TMP3 TMP4 TMP1 TMP2 TMP3 TMP4 ;
  ID=_N_ ; Group="Group "||strip(_N_) ; M=round(M,0.1) ;
  if (Species="Setosa") then do ; R=Group ; W=0 ; end ;
  else W=1 ;
  label M="Mean" L="Lower" U="Upper" ;
run ;
```

ID	Group	N	M	L	U	R	W
1	Group 1	50	50.1	43	58	Group 1	0
2	Group 2	50	59.4	49	70		1
3	Group 3	50	65.9	49	79		1
4	Group 4	50	34.3	23	44	Group 4	0
5	Group 5	50	27.7	20	34		1
6	Group 6	50	29.7	22	38		1
7	Group 7	50	14.6	10	19	Group 7	0
8	Group 8	50	42.6	30	51		1
9	Group 9	50	55.5	45	69		1
10	Group 10	50	2.5	1	6	Group 10	0
11	Group 11	50	13.3	10	18		1
12	Group 12	50	20.3	14	25		1



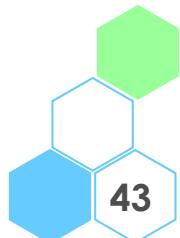
例1: Forest Plot

```
ods listing gpath="c:/temp" image_dpi=300 ;
ods graphics / reset height=450px width=600px ;

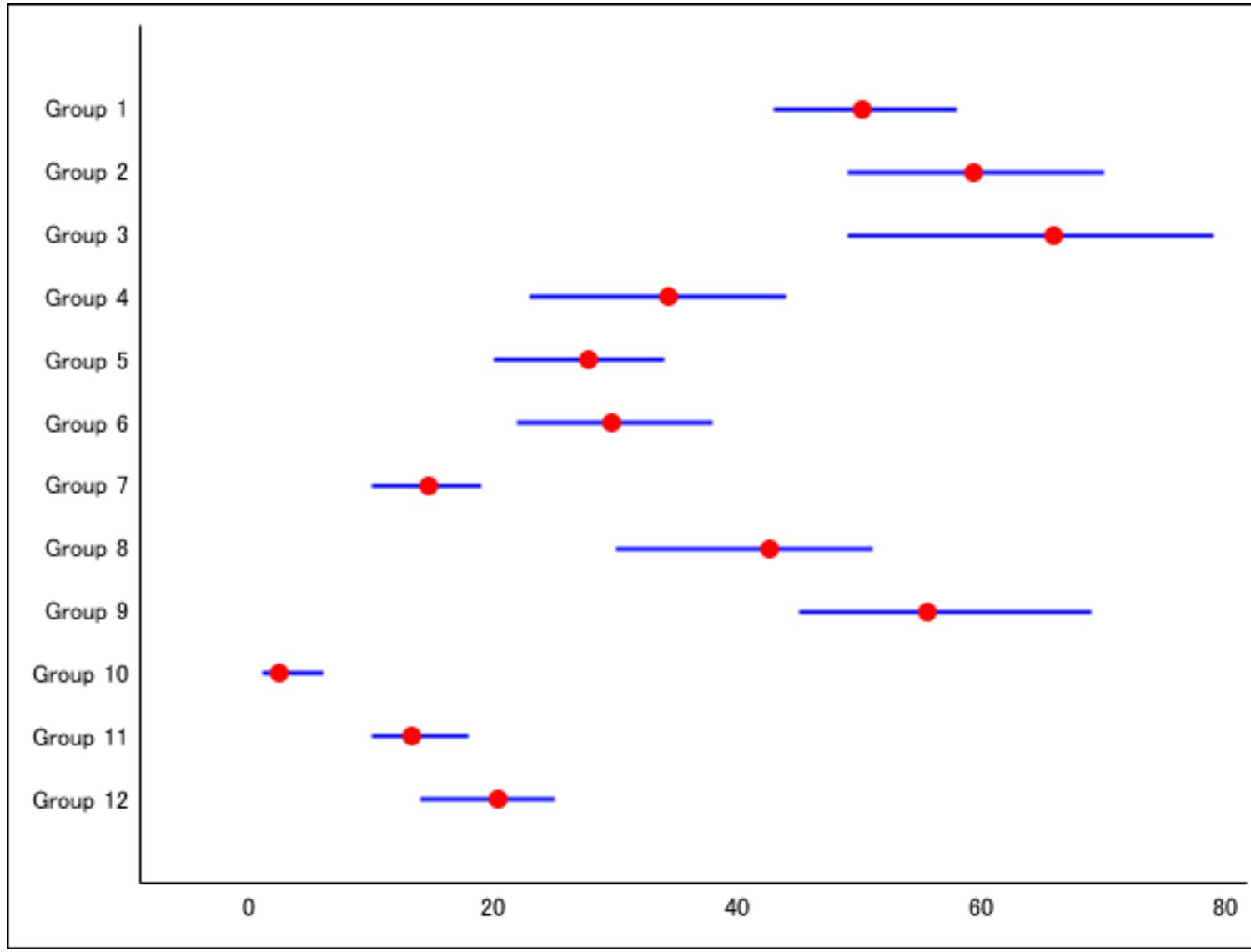
proc sgplot data=MYDATA nowall noborder noautolegend ;
    highlow y=GROUP low=L high=U
        / lineattrs=(pattern=1 color=blue thickness=2) ;
    scatter y=Group x=M
        / markerattrs=(symbol=circlefilled size=10 color=red) ;

    yaxis display=(noticks nolabel)
        reverse offsetmin=0.1 offsetmax=0.1 ;
    xaxis display=(noticks nolabel)
        values=(0 to 80 by 20) offsetmin=0.1 ;
run ;
```

- 線種:highlow ステートメントの lineattrs にて
- 点種:scatter ステートメントの markerattrs にて
- scatter ステートメントの xerrorupper, xerrorlower, errorbarattrs
はエラーバーの属性制御が困難な場合があるので使用せず



例1: Forest Plot

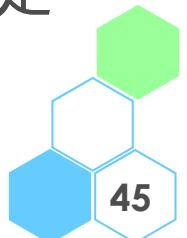


例2: Forest Plot の点を装飾

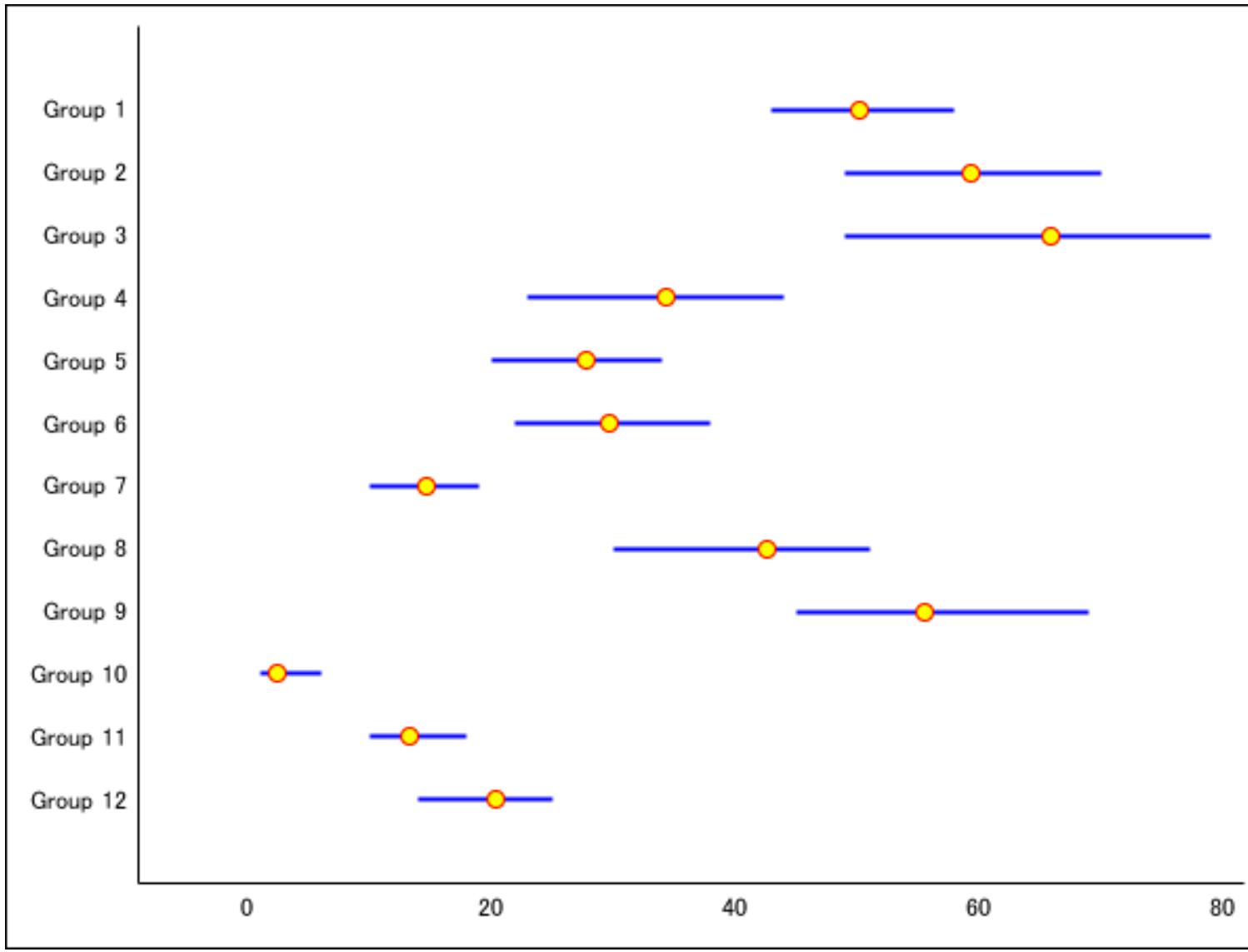
```
proc sgplot data=MYDATA nowall noborder noautolegend ;
    highlow y=GROUP low=L high=U
        / lineattrs=(pattern=1 color=blue thickness=2) ;
    scatter y=Group x=M
        / filledoutlinedmarkers
            markeroutlineattrs=(color=red thickness=1)
            markerfillattrs=(color=yellow)
            markerattrs=(symbol=circlefilled size=10) ;

    yaxis display=(noticks nolabel)
        reverse offsetmin=0.1 offsetmax=0.1 ;
    xaxis display=(noticks nolabel)
        values=(0 to 80 by 20) offsetmin=0.1 ;
run ;
```

- プロット点について「線」と「塗りつぶし」の色等を別々に指定したい場合は、filledoutlinedmarkers 付きで属性制御
- 何気に yaxis ステートメントの reverse が便利



例2: Forest Plot の点を装飾



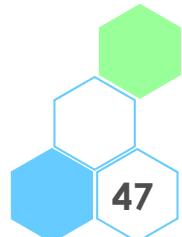
例3: Forest Plot に各種情報をテキスト表示

```
proc sgplot data=MYDATA nowall noborder nocycleattrs noautolegend ;
    highlow    y=GROUP low=L high=U / lineattrs=(pattern=1 color=blue thickness=2) ;
    scatter     y=Group x=M / markerattrs=(symbol=circlefilled size=10 color=red) ;
    yaxistable Group N M L U / y=Group location=inside position=left ;

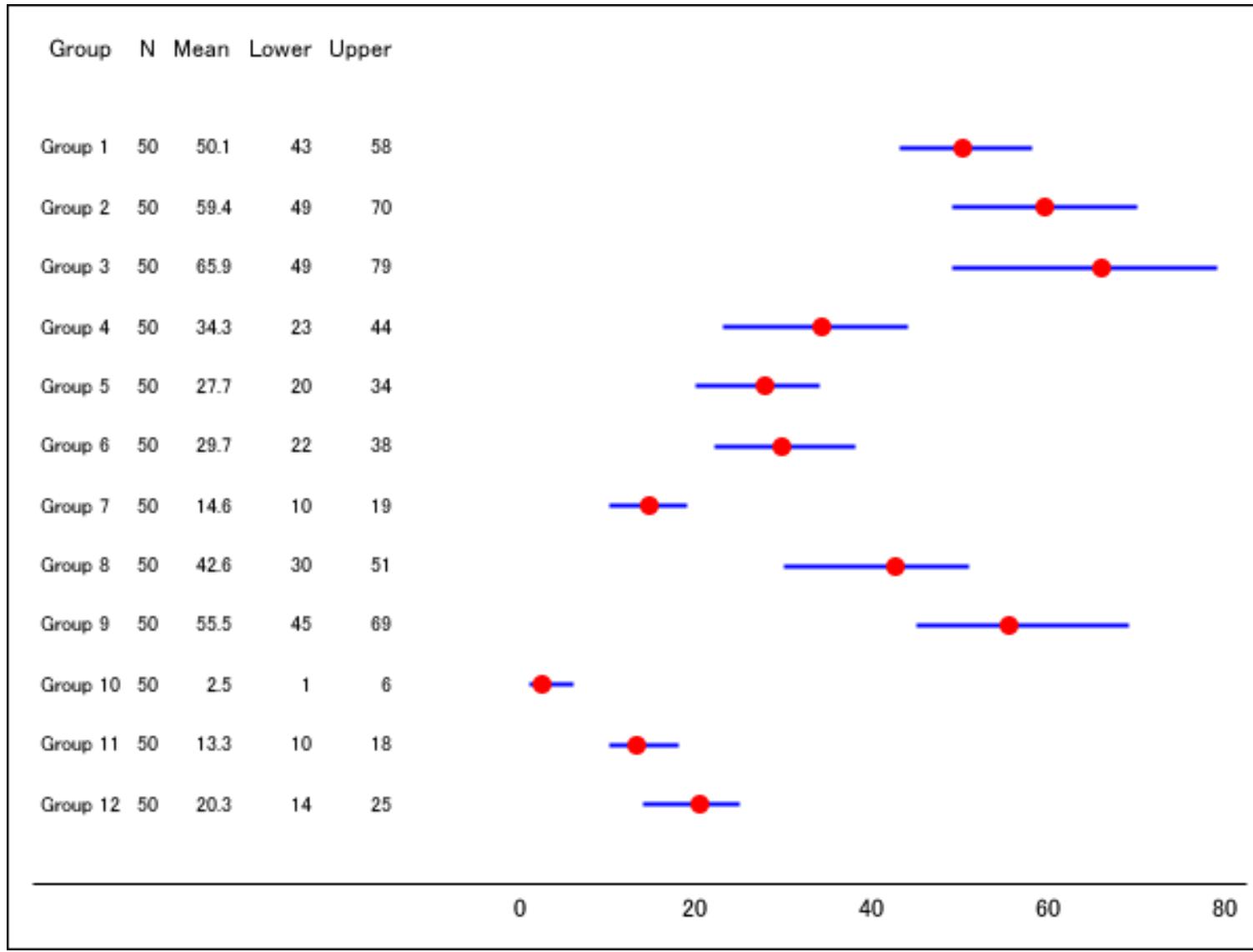
    yaxis display=none reverse offsetmin=0.1 offsetmax=0.1 ;
    xaxis display=(noticks nolabel) values=(0 to 80 by 20) offsetmin=0.1 ;
run ;
```

- 例 1 を改変
- yaxistable で各種情報をテキストで出力
 - 群情報も yaxistable で出力
 - よって yaxis にて display=none を設定

※ 巷では scatter ステートメントでハザード比等の情報を出力している資料もあるが、時代遅れ



例3: Forest Plot に各種情報をテキスト表示



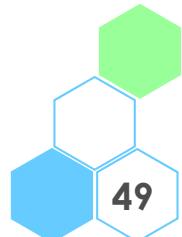
例4: Forest Plot に参照線や注釈を追記

```
proc sgplot data=MYDATA nowall noborder nocycleattrs noautolegend ;
    highlow    y=GROUP low=L high=U / lineattrs=(pattern=1 color=blue thickness=2) ;
    scatter     y=Group x=M / markerattrs=(symbol=circlefilled size=10 color=red) ;
    yaxistable Group N / y=Group location=inside position=left ;
    yaxistable M L U   / y=Group location=inside position=right ;

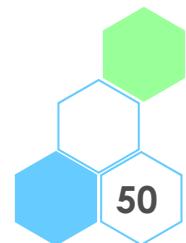
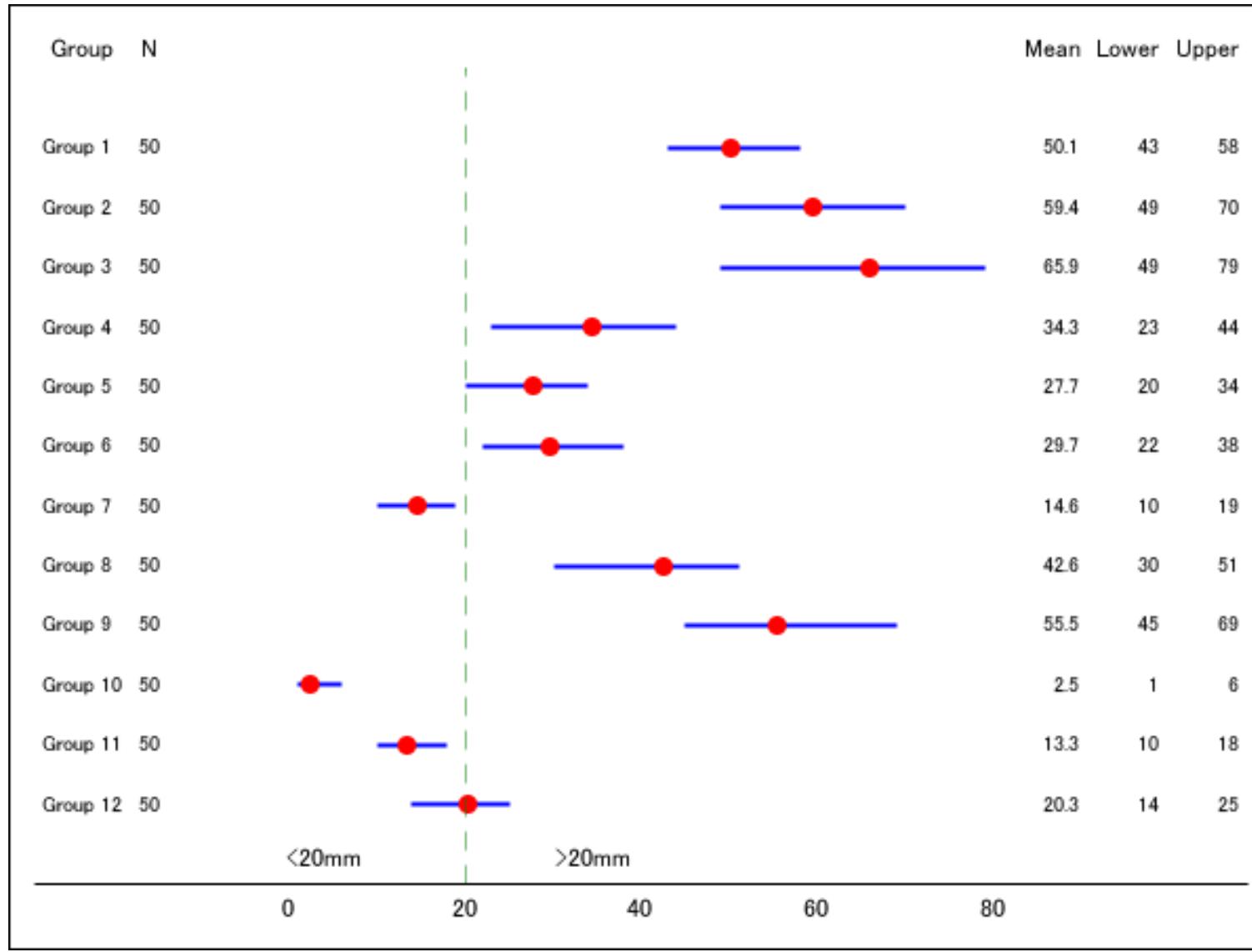
    refline 20 / axis=x lineattrs=(pattern=dash color=green) transparency=0.5 ;
    inset '      <20mm' / position=bottomleft ;
    inset '>20mm'       / position=bottom ;

    yaxis display=none reverse offsetmin=0.1 offsetmax=0.1 ;
    xaxis display=(noticks nolabel) values=(0 to 80 by 20) offsetmin=0.1 ;
run ;
```

- 例 3 を改変
- yaxistable を分割
- 参照線 x=20 を追記
- inset ステートメントで注釈を追記



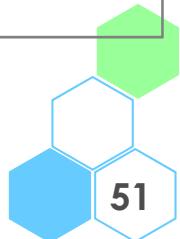
例4: Forest Plot に参照線や注釈を追記



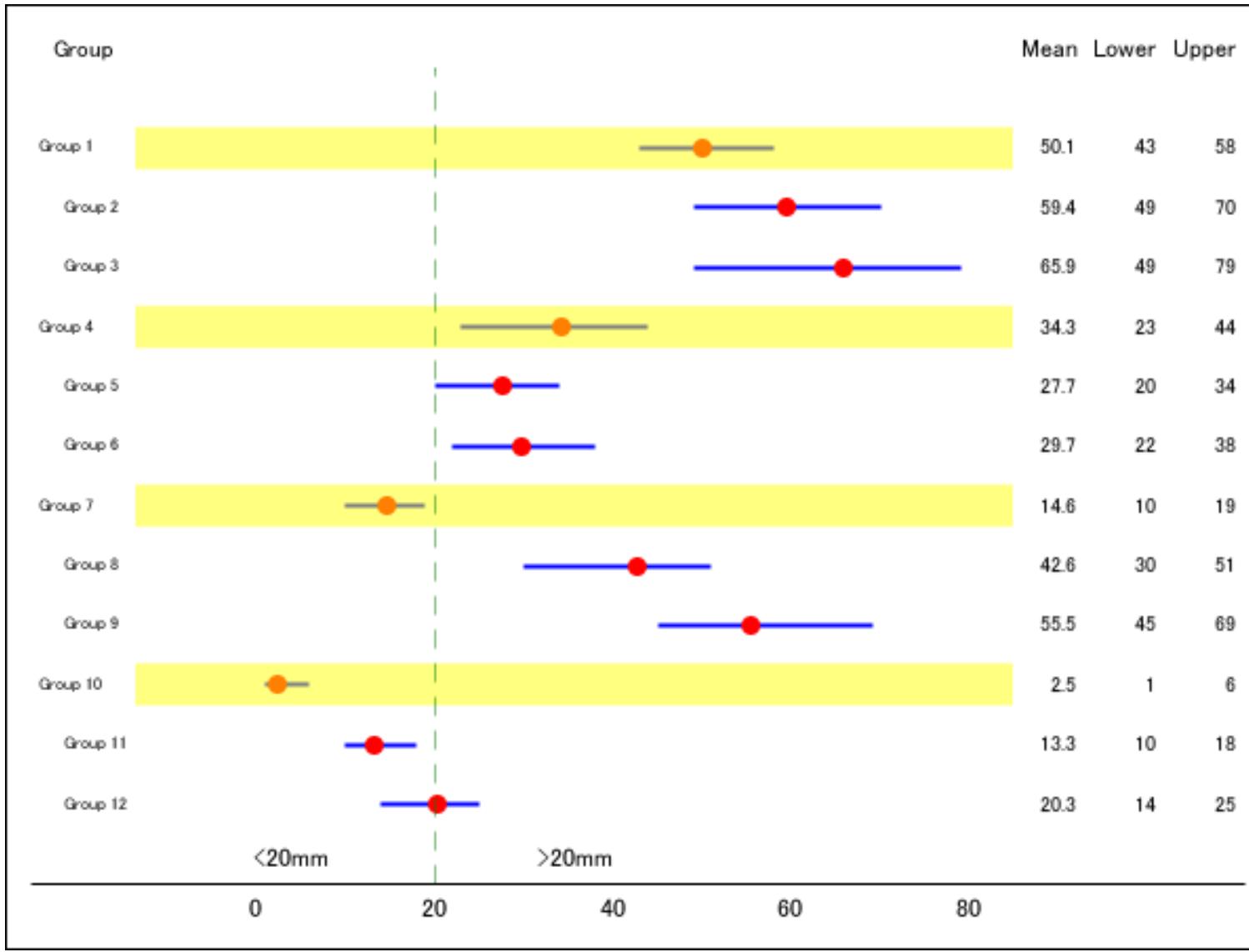
例5: Forest Plot を完成させる

```
proc sgplot data=MYDATA nowall noborder nocycleattrs noautolegend ;  
    highlow    y=GROUP low=L high=U / lineattrs=(pattern=1 color=blue thickness=2) ;  
    scatter     y=Group x=M / markerattrs=(symbol=circlefilled size=10 color=red) ;  
    yaxistable Group / y=Group location=inside position=left labelattrs=(size=8)  
                      valueattrs=(size=6) indentweight=W ;  
    yaxistable M L U / y=Group location=inside position=right ;  
  
    refline R / axis=y lineattrs=(thickness=20 color=yellow transparency=0.5) ;  
    refline 20 / axis=x lineattrs=(pattern=dash color=green transparency=0.5) ;  
    inset '          <20mm' / position=bottomleft ;  
    inset '>20mm'           / position=bottom ;  
  
    yaxis display=none reverse offsetmin=0.1 offsetmax=0.1 ;  
    xaxis display=(noticks nolabel) values=(0 to 80 by 20) offsetmin=0.1 ;  
run ;
```

- Group の種類によってインデントを追加(変数 W)
- Group の種類によってグラフに色を塗る(変数 R)

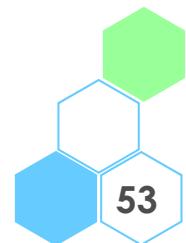


例5: Forest Plot を完成させる



Menu

- 事始
- 種々のグラフ
- グラフのカスタマイズ
- 応用例: Forest Plot
- 応用例: Kaplan-Meier Plot



カプランマイヤー・プロット(Kaplan Meier Plot)

```
proc format ;
  value groupf 1="ALL" 2="AML-Low Risk" 3="AML-High Risk" ;
run ;

data bmt ;
  set sashelp.bmt ;
  if Group="ALL"          then Groupn=1 ;
  if Group="AML-Low Risk" then Groupn=2 ;
  if Group="AML-High Risk" then Groupn=3 ;
run ;

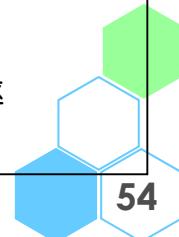
ods output Survivalplot=KM ; ←
proc lifetest data=bmt plots=survival ;
  by Groupn ;
  time T * Status(0) ;
run ;
ods output close ;

proc sgplot data=KM ;
  step x=Time y=Survival / group=Groupn ;
  scatter x=Time y=Censored / group=Groupn ;
  format Groupn groupf. ;
run ;
```

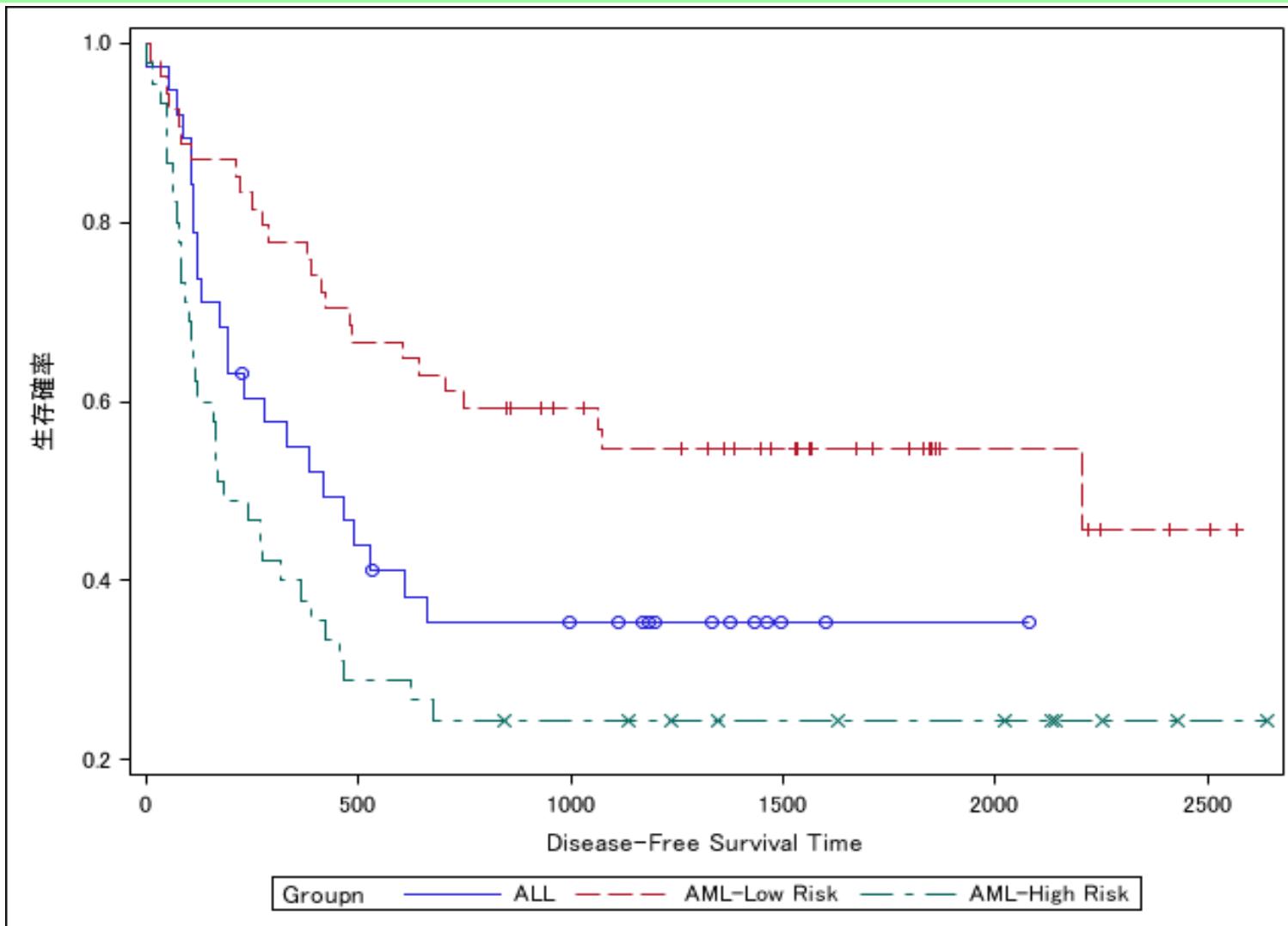
Groupn	Time	Survival	At Risk	Event	Censored	tAtRisk
1	0	1	38	0	.	.
1	0	.	38	.	.	0
1	1	0.97368	38	1	.	.
1	20	.	37	.	.	20
1	40	.	37	.	.	40
1	55	0.94737	37	1	.	.
1	60	.	36	.	.	60
1	74	0.92105	36	1	.	.
:	:	:	:	:	:	:

プロット用のデータを生成

- **Group** : 群 (元データの群変数)
- **Time**
- **Survival** : 生存率
- **AtRisk** : リスク集合
- **Event** : イベント数
- **Censored** : 打ち切り時の生存率
- **tAtRisk** : 表のための時間変数



カプランマイヤー・プロット(Kaplan Meier Plot)



※ 資料 “Customizing the Kaplan-Meier Survival Plot” が良くまとめているので、本資料では深追いしない
<https://support.sas.com/documentation/onlinedoc/stat/141/kaplan.pdf>
<https://support.sas.com/documentation/onlinedoc/stat/151/kaplan.pdf>

カプランマイヤー・プロット(Kaplan Meier Plot)

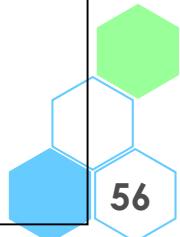
- 次に、累積発生率のグラフを作成する

```
proc format ;
  value groupf 1="ALL" 2="AML-Low Risk" 3="AML-High Risk" ;
run ;

data bmt ;
  set sashelp.bmt ;
  if Group="ALL"           then Groupn=1 ;
  if Group="AML-Low Risk"  then Groupn=2 ;
  if Group="AML-High Risk" then Groupn=3 ;
run ;

ods output Survivalplot=KM_TMP ;
proc lifetest data=bmt plots=survival(atrisk=0 to 160 by 20) ;
  by Groupn ;
  time T * Status(0) ;
run ;
ods output close ;

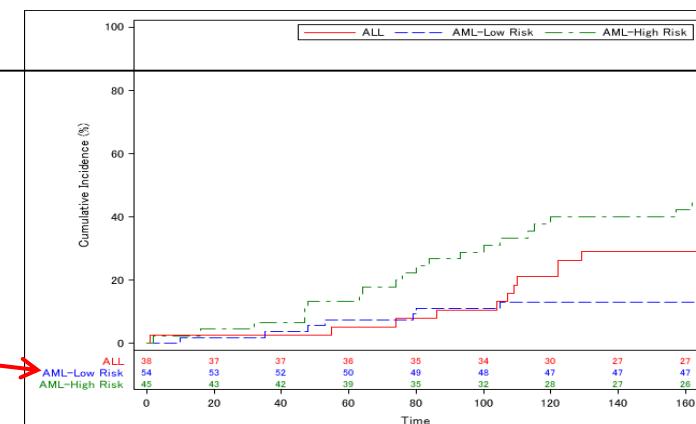
data KM ;
  set KM_TMP ;
  Failure  =100*(1-Survival) ;
  Censored2=100*(1-Censored) ;
run ;
```



カプランマイヤー・プロット(Kaplan Meier Plot)

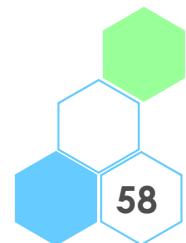
```
proc sgplot data=KM ;
  styleattrs
    datacontrastcolors=(red blue green)
    datasymbols      =(plus plus plus) ;
  step      x=Time y=Failure   / group=Groupn name="x" ;
  scatter   x=Time y=Censored2 / group=Groupn ;
  xaxis     values=(0 to 160 by 20) label="Time";
  yaxis     values=(0 to 100 by 20) label="Cumulative Incidence (%)";
  format   Groupn groupf. ;
  xaxistable Atrisk / x=tAtrisk class=Groupn
    location=inside colorgroup=Groupn separator ;
  *xaxistable Atrisk / x=tAtrisk class=Groupn
    location=outside colorgroup=Groupn separator ;
  keylegend "x" / location=inside position=topright;
run ;
```

- 打ち切り点を+、変数 `tAtrisk` の情報を元にリスク集合を表示等々、いくつか修飾を施す
- ちなみに、上記の `location` を `outside` にすることも可能



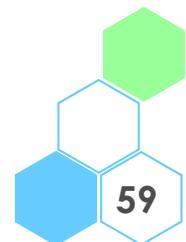
Menu

- 事始
- 種々のグラフ
- グラフのカスタマイズ
- 応用例: Forest Plot
- 応用例: Kaplan-Meier Plot



参考文献

- SAS "SAS 9.4 ODS Graphics: Procedures Guide", Sixth Edition.
<https://documentation.sas.com/api/docsets/grstatproc/9.4/content/grstatproc.pdf>
- SAS/STAT User's Guide "Customizing the Kaplan–Meier Survival Plot".
<https://support.sas.com/documentation/onlinedoc/stat/141/kaplan.pdf>
<https://support.sas.com/documentation/onlinedoc/stat/151/kaplan.pdf>
- 高浪 洋平 他(2015) 「統計解析ソフト SAS(カットシステム)」



END OF CONTENTS