

# Ausgewählte Beispiele zu komplexen Graphiken und Ihre Realisierung in SAS

Thomas Bruckner  
 Institut für Medizinische Biometrie  
 und Informatik,  
 Universität Heidelberg  
 INF 305  
 69120 Heidelberg  
 bruckner@imbi.uni-heidelberg.de

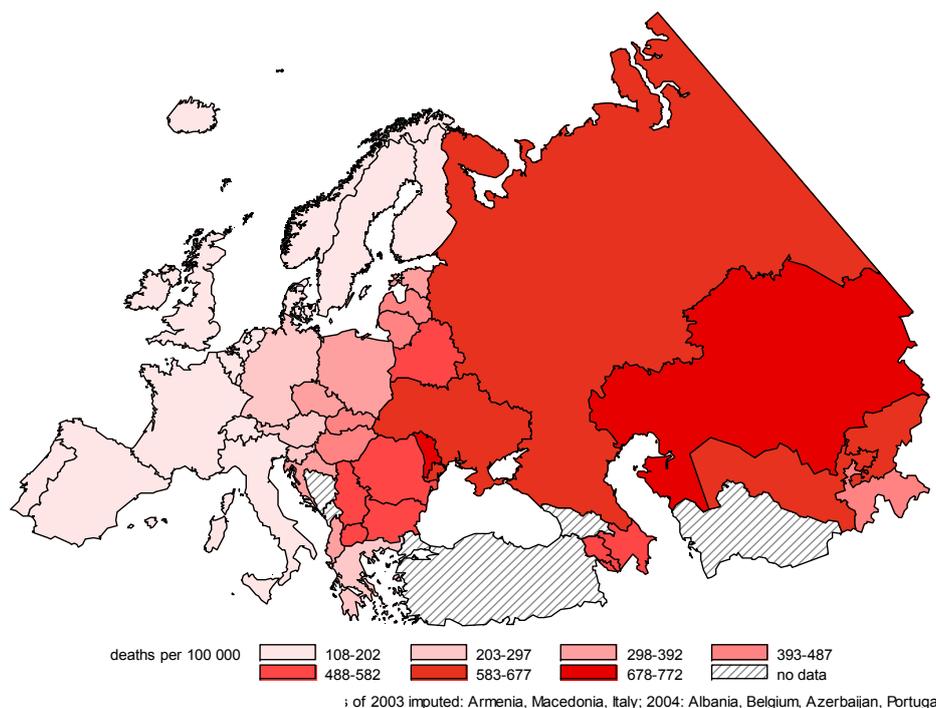
Andreas Deckert  
 Institut für Public Health,  
 Universität Heidelberg  
 INF 324  
 69120 Heidelberg  
 a.deckert@uni-heidelberg.de

## Zusammenfassung

Ausgehend von der Forderung eines Journals nach original in SAS erstellten komplexen Grafiken (ohne Nachbearbeitung durch andere Grafik-Software), sollen komplexe SAS-Grafiken vorgestellt werden, die über Standard-Ansätze hinausgehen. Beispiele sind die Verknüpfung von Mortalitätsdaten mit SAS-Maps, die Benutzung eines Annotate-Datasets zum Hinzufügen von Text und zusätzlichen Grafik-Elementen und die Überlagerung mehrerer Grafiken mit GREPLAY

**Schlüsselwörter:** Annotate-Datei, ANNOTATE=-Option, PROC GMAP, GMAP-Prozedur, PROC GREPLAY, GREPLAY-Prozedur

## 1 Beispiel: GMAP



**Abbildung 1:** Darstellung der Herz-Kreislauf-Mortalität bei Frauen in Europa als Beispiel für die Verwendung von PROC GMAP [1]

## 1.1 Problemstellung

Für eine Arbeit zur Herz-Kreislaufmortalität in Europa war eine kartografische Darstellung der Unterschiede in der Herz-Kreislaufmortalität zwischen den Ländern in Europa gefordert. Dazu wurden zunächst die standardisierten Herz-Kreislauf-Mortalitätsdaten (standardisiert nach der alten Europäischen Standardbevölkerung) in der letzten nahezu vollständigen Fassung der WHO von 2005 bezogen. Mit diesen Daten sollten nun die europäischen Länder entsprechend der Höhe ihrer Mortalitätsrate eingefärbt werden. Die Begrenzung Europas orientierte sich dabei an einer geografischen Definition und enthielt daher auch Länder wie z.B. Kasachstan.

## 1.2 Vorgehensweise

SAS stellt ein umfangreiches Sortiment an verschiedenen Karten bereit, neben Kartendaten zu einzelnen Ländern auch Daten zu Kontinenten und für die gesamte Welt. Jede Karte besteht in der Regel aus zwei Datensätzen, wobei eine davon geografische Namen (z.B. Ländernamen) enthält und die andere die exakten Koordinaten der geografischen Begrenzungen (z.B. Ländergrenzen) in verschiedenen Auflösungen.

Da hier eine erweiterte Definition von Europa verwendet wurde, war die in SAS enthaltene Europa-Karte alleine nicht ausreichend. Daher wurde hier die Weltkarte benutzt und dabei nicht benötigte Kontinente und Länder ausgeschlossen. Alternativ hätten auch die Karten von Europa und Asien kombiniert und dann nicht benötigte Länder in Asien ausgeschlossen werden können.

Die Karten-Daten stehen im Ordner „Maps“ in SAS bereit. Es empfiehlt sich allerdings, die aktuellen Kartendaten von SAS zu beziehen<sup>1</sup>. Die Vorgehensweise im Einzelnen:

1. Karten im Ordner „Maps“ in SAS benutzen oder aktuelle Karten von SAS downloaden (Für jede Karte sind zwei Dateien erforderlich, bei der Weltkarte sind das „Names“ mit den Ländernamen und „World“ mit den genauen geographischen Koordinaten)
2. Für das Einlesen der Dateien der Karten-Daten von der SAS-Homepage ist eine spezielle Prozedur erforderlich:

```
filename tranfile 'names.cpt';  
proc cimport library=work infile=tranfile;  
filename tranfile 'world.cpt';  
proc cimport library=work infile=tranfile;
```

---

<sup>1</sup> <http://support.sas.com/rnd/datavisualization/maponline/>  
<http://support.sas.com/rnd/datavisualization/maponline/html/v92maps.html>  
Für den Download der Karten-Dateien ist eine Registrierung erforderlich!

Die folgenden Tabellen zeigen Auszüge aus den Karten-Daten der Welt. Wichtig für uns als Nutzer sind hier zunächst die Variablen „continent“, „country“, „segment“ und „density“. Mit „continent“ und „country“ können nicht benötigte Kontinente und Länder ausgeschlossen werden. „Segment“ bezeichnet einzelne Teile eines Landes, z.B. Inseln, die damit ebenfalls ausgeschlossen werden können. „Density“ gibt die Stufe der Auflösung der Darstellung an. Für jedes Ländersegment sind mehrere Auflösungsstufen vorhanden. Eine spezielle Auflösung kann z.B. mit einer if-Anweisung gewählt werden (s. Code unten). „ISO-ID“ und „ISO-Name“ der zweiten Tabelle werden benutzt, um später die Verknüpfung mit den WHO-Daten vorzunehmen.

**Tabelle 1:** Inhalt der Tabelle „world“: mehrere Zeilen pro Land/Segment mit verschiedenen Auflösungen

Continent	Country	Segment	Density	X	Y	Lat	Long
93	115	2	3	0.3439	0.9973	1.114	-0.404
93	115	1	2	0.2556	0.9910	1.113	-0.450

**Tabelle 2:** Inhalt der Tabelle „names“: eine Zeile pro Land mit Anzahl der Einträge (Points) in „world“ und ID, Namen

Continent	Country	ISO-ID	ISO-Name	Start	End	Points	...
93	115	031	Azerbaijan	40782	40982	201	...

3. Zunächst werden nun die eingelesenen Karten-Daten mit den geographischen Koordinaten („World“) so manipuliert, dass die gewünschte Begrenzung Europas in der gewünschten Auflösung darstellbar ist:

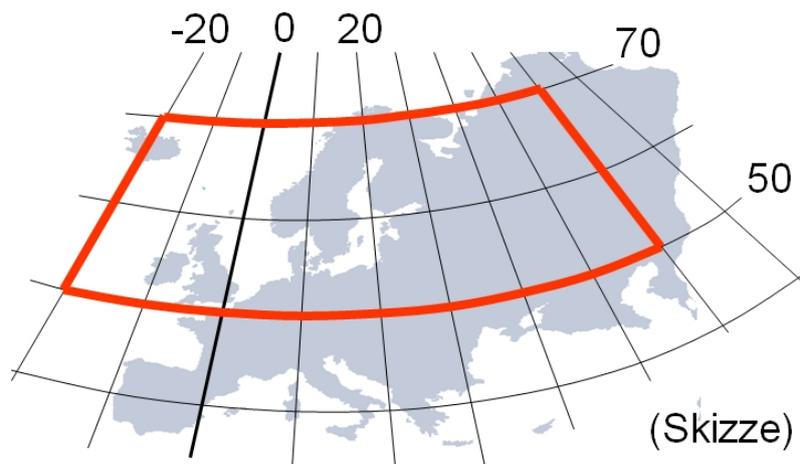
```
data Europe_extended; set world;
  if (density>2) then delete; /*Auflösung bzw. Detailgenauigkeit*/
  /*Begrenzung auf Europa und Asien*/
  if ((cont ne 93) && (cont ne 95)) then delete;
  /*Ausschließen von einzelnen Ländern*/
  if (cont = 95 && id ne 825) ... then delete;
  /*Ausschließen einzelner Ländersegmente (z.B. Inseln)*/
  if (id = 735 && segment>1) then delete;
...

```

4. Die Länderdaten („Names“) werden mit den WHO-Daten verbunden. Dazu wird ein Merge über die in beiden Datensätzen vorhandene ISO-ID oder den Ländernamen (ISO-standardisiert) durchgeführt. Der daraus resultierende Datensatz sei „datafinal“.
5. Nun muss noch eine Projektionsart für die Kartendarstellung sowie der Kartenausschnitt festgelegt werden. Dazu wird wieder die Datei mit den inzwischen begrenzten geographischen Daten („extended\_Europe“) verwendet. Die Wahl der Projektion erfolgt mit proc gproject. Hier wurde die so genannte Lambert-Projektion gewählt

und für den Kartenausschnitt Längengrade zwischen 20 und 60 sowie Breitengrade zwischen 50 und 70. Zu beachten ist, dass als Option „east“ angegeben werden sollte, da sonst die Darstellung aus Sicht Amerikas in für uns spiegelverkehrter Form stattfindet.

```
Proc gproject data=Europe_extended out=Europe_extended
  project = lambert /*Art der Projektion*/
  east /*Spiegelung der Karte (Standard: US-Ansicht)*/
  longmin=-20 longmax=60 latmin=50 latmax=70;
  /*Kartenausschnitt: Längen- und Breitengrade*/
  id id;
```



**Abbildung 2:** Darstellung des Kartenausschnittes mit Projektion LAMBERT

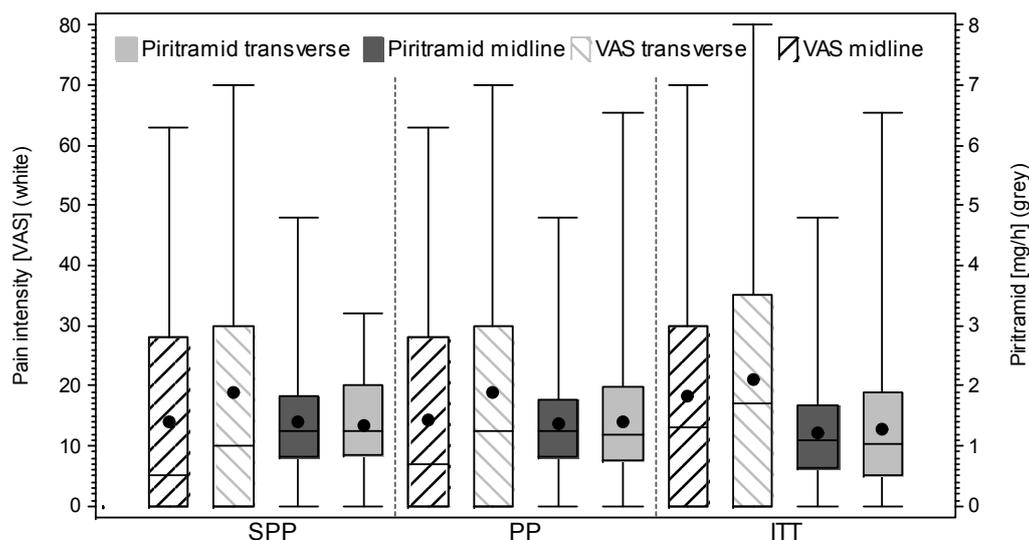
6. Abschließend kann nun die Karte mit proc gmap erstellt werden. Dabei wird „map“ die Tabelle mit den finalen geografischen Daten und „data“ die Tabelle mit den verknüpften WHO-Daten übergeben. „Choro“ gibt den Illustrationstyp einer zweidimensionalen Kartendarstellung an. „Mortality“ ist die WHO-Variable, die die Mortalitätsrate pro Land enthält. Mit „level“ kann die Anzahl der Farbabstufungen gewählt werden. Für die fehlenden Werte wird hier „cdefault=gray“ festgelegt. Dadurch werden Länder für die keine Mortalitätsdaten vorliegen, in durchgehendem Grau eingefärbt (für Schraffur siehe Hinweis unten).

```
Proc gmap map=Europe_extended data=datafinal;
  id id;
  choro mortality/ /*Illustrationstyp, Variable*/
  levels=8 /*Anzahl der Kategorien*/
  cdefault=GRAY /*Fehlende Werte*/
  legend=legend1 /*Zuweisen der Legende*/
  stretch; /*Darstellungs-Optimierung*/
run;
```

Um Länder mit fehlenden Werten zu schraffieren anstatt mit einem durchgehenden Grau einzufärben, können z.B. im Datensatz die fehlenden Werte mit 99999 kodiert werden. Für die einzelnen Wertebereiche können dann eigene Muster bzw. Farben (für reguläre Werte) definiert werden. Hier ein Beispiel-pattern für eine Schraffur:

```
pattern9 value=M3N45 color=CX8C8C8C;
```

## 2 Beispiel: Annotate (Grafik)



**Abbildung 1:** Gleichzeitige Darstellung zweier unterschiedlicher Variablen als Boxplot, getrennt nach Intervention und zusätzlich gruppiert (hier nach Analyse-Sets) und unterschiedlich schraffiert [2]

### 2.1 Problemstellung

In einer klinischen Studie zur Untersuchung einer Äquivalenz-Fragestellung wurden zwei Interventionsgruppen verglichen. Die Zielvariable bestand aus einem kombinierten Endpunkt mit den Variablen subjektives Schmerzempfinden (VAS) und gemessenem Schmerzmittelverbrauch. Da es im Verlauf der Studie aufgrund von vorher nicht bekannten Faktoren zu einer Änderung des Messprotokolls kam, wurde für die Auswertung neben den üblichen intention-to-treat und per-protocol-Studienpopulationen eine zusätzliche per-protocol-Gruppe definiert. Bei der Einreichung des Artikels bei einem Journal wurde dann zum einen gefordert, dass nur Original Grafiken der Auswertungssoftware verwendet werden dürfen und zum anderen war die Anzahl an Grafiken auf fünf begrenzt. Da noch 4 weitere wichtige Grafiken in diesen Artikel integriert waren, musste die Darstellung des kombinierten primären Endpunktes in einer Grafik erfolgen. Die Darstellung sollte also Boxplots zweier unterschiedlicher Variablen, getrennt nach Intervention und zusätzlich gruppiert (hier nach Analyse-Sets) und unterschiedlich schraffiert enthalten.

## 2.2 Vorgehensweise

Für die Darstellung zweier Outcome-Variablen mit verschiedenen y-Achsen (links und rechts) sind proc boxplot und andere Verfahren ungeeignet bzw. lässt z.B. proc boxplot eine Darstellung mit zwei unterschiedlichen Achsen nicht zu. Eine gleichzeitige Darstellung zweier Boxplot-Grafiken ist jedoch mit plot2 in proc gplot und der Option i=boxt im symbol-Statement möglich. Allerdings ist hierbei die Verwendung eines pattern-Statement für das Schraffieren der Boxen wirkungslos. Zudem stellt die Option i=boxt nur die Mediane im Boxplot dar, die Mittelwerte müssen zusätzlich in die Grafik eingebaut werden. Die Schraffur und die Darstellung der Mittelwerte werden hier über einen annotate-Datensatz realisiert. Mit einem annotate-Datensatz stehen spezielle Grafikbefehle zur Verfügung, die das Einbinden von Grafik-Elementen und Text in von normalen Grafik-Prozeduren erstellte Darstellungen erlauben.

1. Der eine Teil des primären Endpunktes (outcome1) soll später mit plot und der andere (outcome2) mit plot2 innerhalb derselben gplot Prozedur mit zwei unterschiedlichen Achsen dargestellt werden. Der Datensatz enthält die beiden Outcomes sowie eine Variable „set“ die die beiden Outcomes den unterschiedlichen Analyse-Sets zuweist (6 Stufen). Damit die Darstellungen der beiden Variablen abwechselnd auf der x-Achse erfolgen, wird der Datensatz zunächst verdoppelt und mit „set“ und „var“ 12 „Fälle“ eingeführt. Diese zwölf Fälle werden dann abwechselnd den beiden Outcomes zugewiesen. Falls „outcome1“ gesetzt ist, wird „outcome2“ dabei auf Missing gesetzt und umgekehrt.

```
data double_data; set single_data;
    var=1; output; var=2; output;
data final; set double_data ;
    if (set=1 && var=1) then _var=1;
...
if (set=6 && var=2) then _var=12;
if _var in (1,2,5,6,9,10) then outcome1=.;
if _var in (3,4,7,8,11,12) then outcome2=.;
```

2. In einem nächsten Schritt werden die Mittelwerte/Quartile für beide Endpunkte in die Tabelle „means“ ausgegeben.

```
proc means data=final mean q1 q3;var outcome1;by group _var;
    output out=means mean=m_end1 q1=_q1 q3=_q3;
```

(für outcome2 entsprechend)

3. Dann wird ein Annotate-Datensatz für die Füllmuster und die Mittelwerte (und Legende) erzeugt. Mit „xsys='2'“ und „ysys='2'“ wird ein Koordinatensystem gewählt, mit dem direkt auf die Darstellung der 12 Fälle zugegriffen werden kann, ohne deren genaue x- und y-Koordinaten im Grafen zu kennen. So kann z.B. der linke untere Punkt der Schraffurbox dadurch bestimmt werden, dass von „var“ (die

12 Fälle) einfach die Hälfte der Boxplot-Dicke für den x-Wert abgezogen wird und der y-Wert auf q1 gesetzt wird. Mit „Move“ erreicht man, dass dieser Punkt in der Grafik zunächst angefahren wird. Nachdem dann die rechte obere Ecke der Schraffurbox in derselben Weise bestimmt wurde, wird mit „Bar“ ein Balken (hier in Form einer Schraffur) erstellt. Die Parameter, die vor diesem Befehl angegeben werden, definieren dabei das Aussehen der Schraffur.

Da proc gplot wie oben schon erwähnt keine Mittelwerte im Box-Plot darstellt, müssen diese ebenfalls im Annotate-Datensatz angegeben werden. Dies geschieht in ähnlicher Weise wie bei den Schraffuren. Zunächst wird mit „Move“ und den Koordinaten (die wieder direkt aus den 12 Fällen und dem berechneten Mittelwert abgeleitet werden) der Mittelwert angefahren. Dann erfolgt die Darstellung des Mittelwerts über den Befehl „Pie“. Die Mittelwerte von „outcome2“ werden getrennt in derselben Weise in einem zweiten Annotate-Datensatz definiert. Diese Vorgehensweise ist für die getrennte Übergabe an plot und plot2 in proc gplot notwendig.

```
data _annotate; set means end=eof;
  length function style $8;
  xsys='2'; ysys='2'; /*Auswahl des Koordinatensystems*/
/*Eckpunkt für Füllmuster outcome1 links unten festlegen:*/
  if _var in (1 2 5 6 9 10) then do;
    x=_var-.3; y=q1; end;
  else do; x=0; y=0; end; /*Kein Füllmuster für outcome2*/
  function='MOVE'; output;
/*Eckpunkt für Füllmuster outcome1 rechts oben festlegen:*/
  y=q3;
  if _var in (1 5 9) then do;
    x=_var+.3; style='R10'; color='BLACK';
  end; else if _var in (2 6 10) then do;
    x=_var+.3; style='L10'; color='LIGR'; end;
  else do; x=0; y=0; end; /*Kein Füllmuster für outcome2*/
  function='BAR'; output;
/*Mittelwerte outcome1*/2
  if _var in (1 2 5 6 9 10) then do;
  y=m_end1; x=_var; function='MOVE'; output;
  y=m_end1; x=_var; color='BLACK';
  style='solid'; size=0.2;
  Angle=0; Rotate=360; function='PIE'; output;
  end;
/*Erstellen der Legende mit ähnlichen Befehlen! (s. auch Bsp. 3)*/
...
```

Tabelle 1 zeigt den Inhalt des Annotate-Datensatzes. Während des Plots wird diese Tabelle nun Zeile für Zeile abgearbeitet und dabei die Befehle schrittweise ausgeführt.

<sup>2</sup> Die Mittelwerte für den zweiten Endpunkt werden separat in einen zweiten annotate-Datensatz (\_annotate2) geschrieben

**Tabelle 1:** Inhalt der Tabelle „\_annotate“: Darstellung der Befehlskette für die Grafikerstellung

group	_var	m_end1	_q1	_q3	function	style	X	Y	color	rot	...
1	1	14.0	0.1	28	MOVE		0.7	0.1			...
1	1	14.0	0.1	28	BAR	R10	1.3	28	black		...
1	1	14.0	0.1	28	MOVE	R10	0.7	0.1	black		...
⋮	1	14.0	0.1	28	PIE	solid	1	14	black	360	...
⋮	1	.	.	.	MOVE		0	0			...
⋮	2	19.0	1.2	30	MOVE		1.7	1.2			...

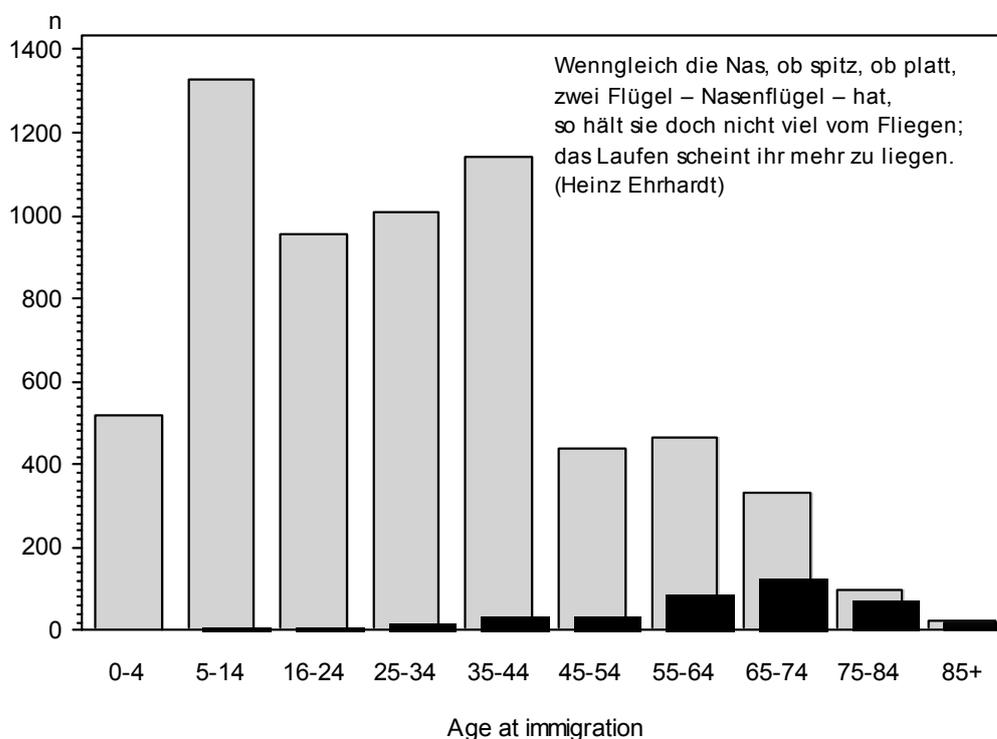
4. Nun kann die kombinierte Grafik mit proc gplot erstellt werden. Dazu müssen hier neben der x-Achse zwei y-Achsen definiert werden. In plot und plot2 werden dann mit „annotate=“ die Annotate-Datensätze zugewiesen.

```

symbol1 i=boxt00 co=black bwidth=5; /*Dasselbe für symbol2*/
symbol3 i=boxft00 co=black cv=dagr bwidth=5; /* >> symbol4*/
/*3 Achsen definieren!*/ ...
axis3 label=(h=0.35 a=90 r=0 'Piritramid [mg/h] (grey)')
order=(0 to 8 by 1);
proc gplot data=final;
plot outcome1*_var=group/haxis=axis1
vaxis=axis2 nolegend annotate=_annotate;
plot2 outcome2*_var=group/vaxis=axis3
nolegend annotate=_annotate2;
run; quit;

```

### 3 Beispiel: GREPLAY, Annotate (Text)



**Abbildung 1:** Überlagerung von Grafiken durch leichtes Verrücken der zweiten Grafik (in ähnlicher Weise können auch Grafiken aneinander gehängt oder ineinander verschachtelt werden)

#### 3.1 Problemstellung

Die Darstellung der Anzahl von Todesfällen pro Altersgruppe und der Altersverteilung sollte in einer einzigen Grafik als Balkendiagramme erfolgen. Dabei sollte die Darstellung der Todesfälle leicht nach rechts verschoben sein (in ähnlicher Weise können auch Grafiken aneinander gehängt oder ineinander verschachtelt werden). Zusätzlich war in der Grafik eine Legende am oberen rechten Eck mit zusätzlichen Grafik-Elementen eingefügt, dies wurde hier der Einfachheit halber durch ein Textfeld ersetzt.

#### 3.2 Vorgehensweise

1. Um mehrere Grafiken übereinander zu legen oder aneinander anzuhängen, kann man diese zunächst in einem SAS-Grafikcontainer intern ablegen und später dann in einer vordefinierten Schablone zusammensetzen. Die Grafiken werden dazu zunächst mit den üblichen Prozeduren erstellt, wobei dann mit `gout` in der Prozedur die Ablage im Grafikcontainer (hier Grafikablage genannt) erfolgt. Der Grafikcontainer wird im Work-Ordner angelegt. Die Namen, unter denen die Grafiken im

Grafikcontainer abgelegt werden, werden von SAS selbst festgelegt (hier bei der ersten Grafik GCHART und bei der zweiten GCHART1).

```
pattern1 c=lightgrey;
proc gchart data=data1 gout=Grafikablage; → GCHART
vbar var1/midpoints=0 to 9 by 1
maxis=axis1 axis=axis2 space=1.5;
run; quit;
pattern1 c=black;
proc gchart data=data2 gout=Grafikablage; → GCHART1
vbar var2/midpoints=0 to 9 by 1
noaxis noframe nolegend space=1.5;
run; quit;
```

2. Für den Text, der zusätzlich innerhalb der Grafik erscheinen soll, wird zunächst über ein Annotate-Dataset eine Tabelle mit entsprechenden Befehlen angelegt. Der entsprechende Befehl für die Textausgabe lautet "label". Mit "position" wird die Position des Textes relativ zu dem genau angegebenen Punkt durch die x- und y-Koordinaten beschrieben. Es kann z.B. genau, neben, oberhalb, unterhalb usw. positioniert werden. Mit "When=A" wird gewährleistet, dass der Text später nicht durch die Grafiken überdeckt wird. Da hier die Textzeilen untereinander stehen, wird x nur einmal gesetzt und dann für jede Zeile die jeweilige Y-Koordinate. Mit "text" wird dann der auszugebende Text selbst angegeben.

```
data _add_text;
length function style color $ 8 text $ 80;
retain xsys ysys "2";
function="label"; position="6"; size=0.75; x=55;
style="arial"; color="black";
When='A'; /*Text im Vordergrund der Grafik*/
y=93; text="Wenngleich die Nas, ob ..."; output;
y=89; text="zwei Flügel - Nasenflügel ..."; output;
...
```

Die daraus entstandene Tabelle enthält pro Zeile die jeweiligen Befehle zur Erstellung des Textes.

**Tabelle 1:** Inhalt der Tabelle „\_add\_text“: Darstellung der Befehlskette für den Text in der Grafik

<u>function</u>	style	color	when	pos	size	X	Y	text	...
label	arial	black	A	6	0.75	55	93	Wenngleich die ...	...
label	arial	black	A	6	0.75	55	89	zwei Flügel ...	...

3. Diese Tabelle wird nun mit `proc gslide` und Angabe desselben Grafikcontainers wie oben abgelegt und erhält von SAS den Namen `add_text`.

```
proc gslide gout=Grafikablage → ADD_TEXT
name="add_text"
annotate=_add_text;
run; quit;
```

4. In einem nächsten Schritt wird nun mit `proc greplay` die Grafikschaablone definiert. Hier werden drei gleichgroße Grafikrahmen übereinander gelegt, wobei der Rahmen der zweiten Grafik leicht nach rechts verschoben wird.

```
proc greplay nofs
tc=work.gseg; /*Festlegen des Rahmen-Containers*/
tdef newtemp /*Erstellen einer neuen Schablone*/
1/ llx=0 lly=0 /*Rahmen 1, Punkt links unten*/
lrx=100 lry=0 /*Rahmen 1, Punkt rechts unten*/
ulx=0 uly=100 /*Rahmen 1, Punkt links oben*/
urx=100 ury=100 /*Rahmen 1, Punkt rechts oben*/
2/ llx=1.5 lly=0 /*Der Rahmen der zweiten Grafik*/
lrx=101.5 lry=0 /*wird geringfügig nach rechts*/
ulx=1.5 uly=100 /*verschoben (gleiche Größe)*/
urx=101.5 ury=100
3/ lly=0 uly=100 llx=0 lrx=100
lry=0 ury=100 ulx=0 urx=100;
run; quit;
```

5. Ebenfalls mit `proc greplay` erfolgt dann die finale Zusammenführung der Grafiken und Texte. Dazu wird mit `"tc="` die Grafikschaablone, die oben festgelegt wurde, geladen und mit `"igout="` die Inhalte des Grafikcontainers. `"treplay"` fügt dann die einzelnen Elemente zusammen und verwendet dazu in der Reihenfolge die oben festgelegten Rahmen.

```
proc greplay nofs
tc=work.gseg /*Rahmen-Cont.*/
template=newtemp /*Schablone*/
gout=Grafikablage /*Output*/
igout=Grafikablage; /*Input*/
treplay
1: gchart
2: gchart1
3: add_text;
device emf;
run;
quit;
```

HINWEIS: Der Grafikcontainer „Grafikablage“ im Arbeitsverzeichnis von SAS muss vor jedem neuen Durchlauf geleert werden. SAS überschreibt vorhandene Grafikobjekte

nicht, sondern legt neue mit Namenszusätzen an. Diese Namen stimmen dann nicht mehr mit denen in proc greplay überein.

## **Literatur**

- [1] Deckert et.al.: Time trends in cardiovascular disease mortality in Russia and Germany from 1980 to 2007 - are there migration effects? BMC Public Health. 2010 Aug 17;10:488
- [2] Seiler CM et.al.: POVATI: Midline versus transverse incision in major abdominal surgery: a randomized, double-blind equivalence trial (POVATI: ISRCTN 60734227). Ann Surg. 2009 Jun;249(6):913-20.