Auf verschiedenen Wegen zur richtigen Graphik Tipps & Tricks für den leichteren Umgang mit der SAS Software

Ralf Minkenberg
Boehringer Ingelheim Pharma GmbH & Co. KG
Binger Str. 173
55216 Ingelheim
ralf.minkenberg@boehringer-ingelheim.com

Thomas Rüdiger
AXA Konzern AG
Colonia-Allee 19-21
51067 Köln
thomas.ruediger@axa.de

Zusammenfassung

Die Erstellung von Graphiken in SAS ist schon immer von Diskussionen begleitet, in wie weit diese Aufgabe effizient und in hoher Qualität überhaupt geleistet werden kann. Im Folgenden soll anhand von einigen Beispielen gezeigt werden, wie Graphiken in guter Qualität sowohl mit den „klassischen" SAS/GRAPH-Prozeduren als auch mit den „neueren" ODS-Prozeduren erstellt werden können. Es soll keinesfalls eine komplette Übersicht über das Thema gegeben werden, vielmehr werden Beispielgraphen erzeugt, an denen einige wesentliche Ideen zu Graphiken in SAS mit verschiedenen Prozeduren erläutert werden. Insgesamt sollen diese Beispiele ermutigen, in SAS Graphiken zu erstellen, die den heutigen Anforderungen für Veröffentlichungen definitiv genügen.

Schlüsselwörter: Graphik, SAS/GRAPH, ODS, Statistical Graphics

1 Einleitung

1.1 Verschiedene Graphik-Prozeduren in SAS

Zur Erstellung von Graphiken in SAS stehen schon seit einiger Zeit verschiedene Möglichkeiten zur Verfügung.
Traditionell können Graphiken mit Hilfe von Prozeduren aus dem SAS/GRAPH-Modul erstellt werden. Hier stehen eine Vielzahl an Prozeduren und Optionen zur Verfügung, um nahezu jede Art Graphik zu erzeugen. Jedoch ist gerade ohne Vorkenntnisse die Verwendung dieser Prozeduren häufig nicht intuitiv und einfach.Es sind eine Vielzahl an ergänzenden Optionen und Befehlen notwendig, um die Standardausgabe einer Graphikprozedur seinen Bedürfnissen anzupassen bzw. um eine qualitativ befriedigende Ausgabe zu erhalten. Trotzdem wird im Folgenden gezeigt, dass es auch ohne Benutzung von ANNOTATE-Datensätzen genügend Möglichkeiten gibt, eine Graphik qualitativ hochwertig zu erzeugen. Hierbei wird auch deutlich, dass nur eine überschaubare Anzahl an Optionen notwendig ist.
Seit einiger Zeit wird als Alternative zur Graphikerstellung auch die ODS-Philosophie angeboten, mit deren Hilfe unter Verwendung „neuer" sog. SG(„Statistical Graphics")-

Prozeduren sehr einfach Graphiken erzeugt werden können. Die Syntax ist hier in einer Weise gegeben, dass ein einfacher Einstieg möglich ist und auch mit sehr wenigen Optionen bereits beeindruckende Ergebnisse erzielt werden. Jedoch muss eine neue Syntax erlernt werden, die sich nahezu komplett von der Syntax der SAS/GRAPH-Prozeduren unterscheidet. Die Beispiele im Folgenden werden dies verdeutlichen.

1.2 Verwendete Daten

Als Datenbasis wurden Daten aus der letzten und aktuellen Fußball-Bundesliga-Saison verwendet (Quelle: www.kicker.de). In dem benutzten Datensatz für die „klassischen" Graphikprogramme finden sich die geschossenen und erhaltenen Tore neben der Tordifferenz der ersten 21 Spieltage der Saisons 2014/15 und 2015/16 für die Vereine FC Bayern München (Variablen BAYERN_*) sowie Borussia Dortmund (Variablen BVB_*). Die gleichen Daten werden für die ODS-Graphikprogramme verwendet, jedoch sind die Daten hier in einer vertikalen Struktur gespeichert. In den Abbildungen 1 und 2 ist jeweils ein Ausschnitt der beiden verwendeten Datensätze dargestellt.

	saison	spieltag	bayern_t	bayern_g	bvb_t	bvb_g	diff	verein	bayern_d	bvb_d
34	2014	17	.	.	.	.	-1	2	1	-1
35	2014	18	42	8	18	26	-3	1	-3	0
36	2014	18	.	.	.	.	0	2	-3	0
37	2014	19	43	9	18	27	0	1	0	-1
38	2014	19	.	.	.	.	-1	2	0	-1
39	2014	20	45	9	21	27	2	1	2	3
40	2014	20	.	.	.	.	3	2	2	3
41	2014	21	53	9	25	29	8	1	8	2
42	2014	21	.	.	.	.	2	2	8	2
43	2015	1	5	0	4	0	5	1	5	4
44	2015	1	.	.	.	.	4	2	5	4
45	2015	2	7	1	8	0	1	1	1	4
46	2015	2	.	.	.	.	4	2	1	4
47	2015	3	10	1	11	1	3	1	3	2
48	2015	3	.	.	.	.	2	2	3	2
49	2015	4	12	2	15	3	1	1	1	2

Abbildung 1: Ausschnitt der Daten für die SAS/GRAPH-Prozeduren

	Gruppe	Spieltag	Tore	Tordifferenz
153	Bayern Tore 2014	20	45	.
154	Bayern Gegentore 2014	20	9	36
155	Dortmund Tore 2014	20	21	.
156	Dortmund Gegentore 2014	20	27	-6
157	Bayern Tore 2015	20	50	.
158	Bayern Gegentore 2015	20	9	41
159	Dortmund Tore 2015	20	52	.
160	Dortmund Gegentore 2015	20	24	28
161	Bayern Tore 2014	21	53	.
162	Bayern Gegentore 2014	21	9	44
163	Dortmund Tore 2014	21	25	.
164	Dortmund Gegentore 2014	21	29	-4
165	Bayern Tore 2015	21	53	.
166	Bayern Gegentore 2015	21	10	43
167	Dortmund Tore 2015	21	53	.
168	Dortmund Gegentore 2015	21	24	29

Abbildung 2: Ausschnitt der Daten für die ODS-Prozeduren

2 SAS/GRAPH

2.1 PROC GBARLINE – Grundlagen

Zur Illustration der Möglichkeiten zur Erstellung publikationsreifer Graphiken mit SAS/GRAPH-Prozeduren wird die Prozedur GBARLINE gewählt. Mit dieser lassen sich Balkendiagramme und Linienplots kombinieren. Die finale Graphik ist in Abbildung 3 zu sehen.

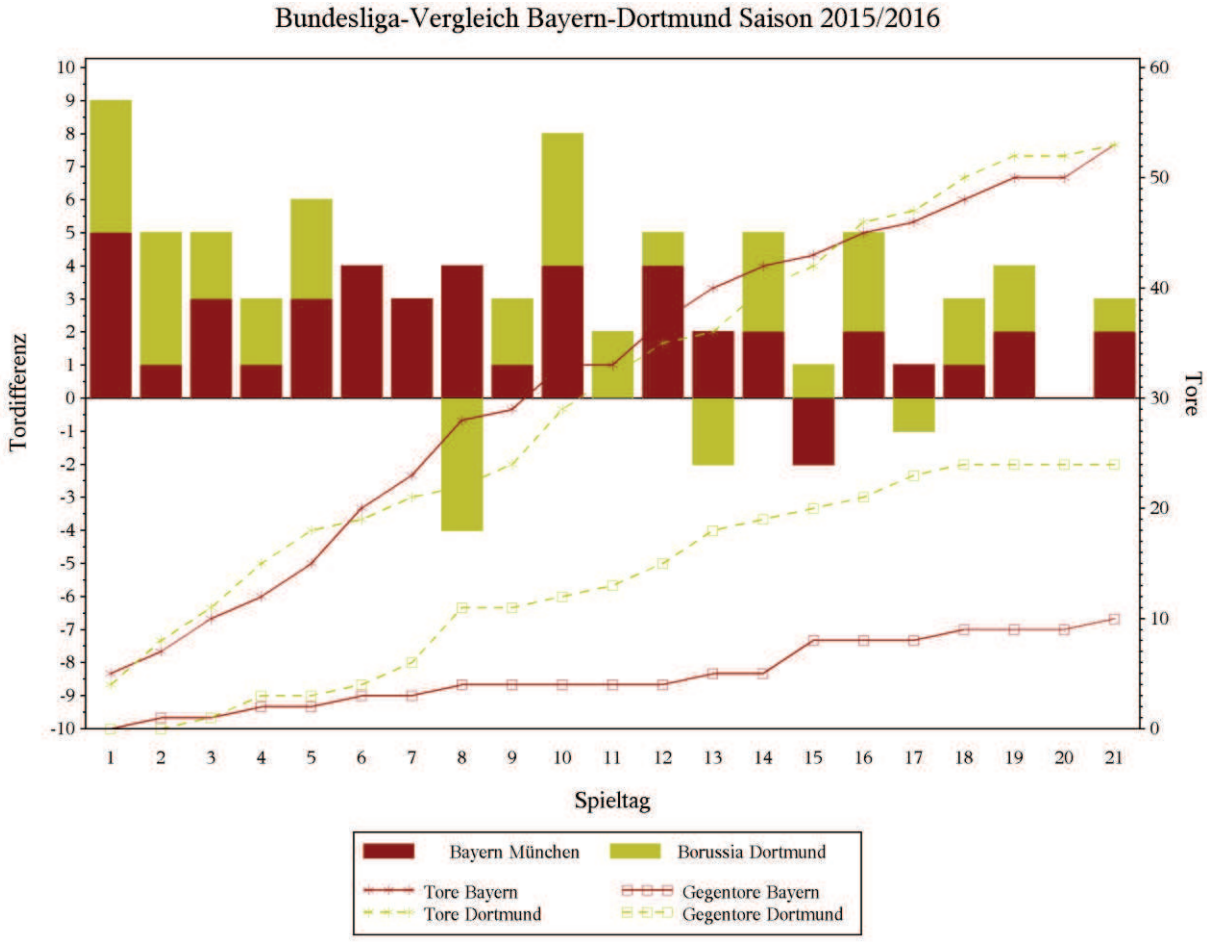

Abbildung 3: Darstellung der Torverläufe und Tordifferenz je Spieltag – finale Graphik

Die Syntax von PROC GBARLINE ist ähnlich der Prozeduren GCHART und GPLOT, dessen Ergebnisse auch in einer Graphik verbunden erstellt werden. Es gibt einen BAR-Befehl für ein Balkendiagramm; hier ist auch nur genau ein entsprechender Befehl erlaubt. Zusätzlich können mehrere PLOT-Befehle verwendet werden, mit denen Linienplots in die Graphik eingefügt werden. Die bekannten weiteren Befehle aus SAS/GRAPH-Prozeduren zur Formatierung der Achsen (AXIS), Symbole (SYMBOL) und Legenden (LEGEND) werden wie üblich angewandt.
Ohne Optionen zur Formatierung bzw. zum Layout kann eine Graphik mit folgendem Code erstellt werden:

```
proc gbarline data=mybuli(where=(saison=2015));
  title 'Bundesliga-Vergleich Bayern-Dortmund Saison 2015/2016';
  bar spieltag / discrete subgroup=verein type=sum sumvar=diff;
  plot / type=sum sumvar=bayern_t;
  plot / type=sum sumvar=bayern_g;
  plot / type=sum sumvar=bvb_t;
  plot / type=sum sumvar=bvb_g;
run;
quit;
```

Die so erzeugte Graphik (siehe Abbildung 4) erinnert schon an das gewünschte Ziel. Es sind im Wesentlichen noch Formatierungen notwendig, um tatsächlich das zu erzeugen, was hier gewünscht ist.

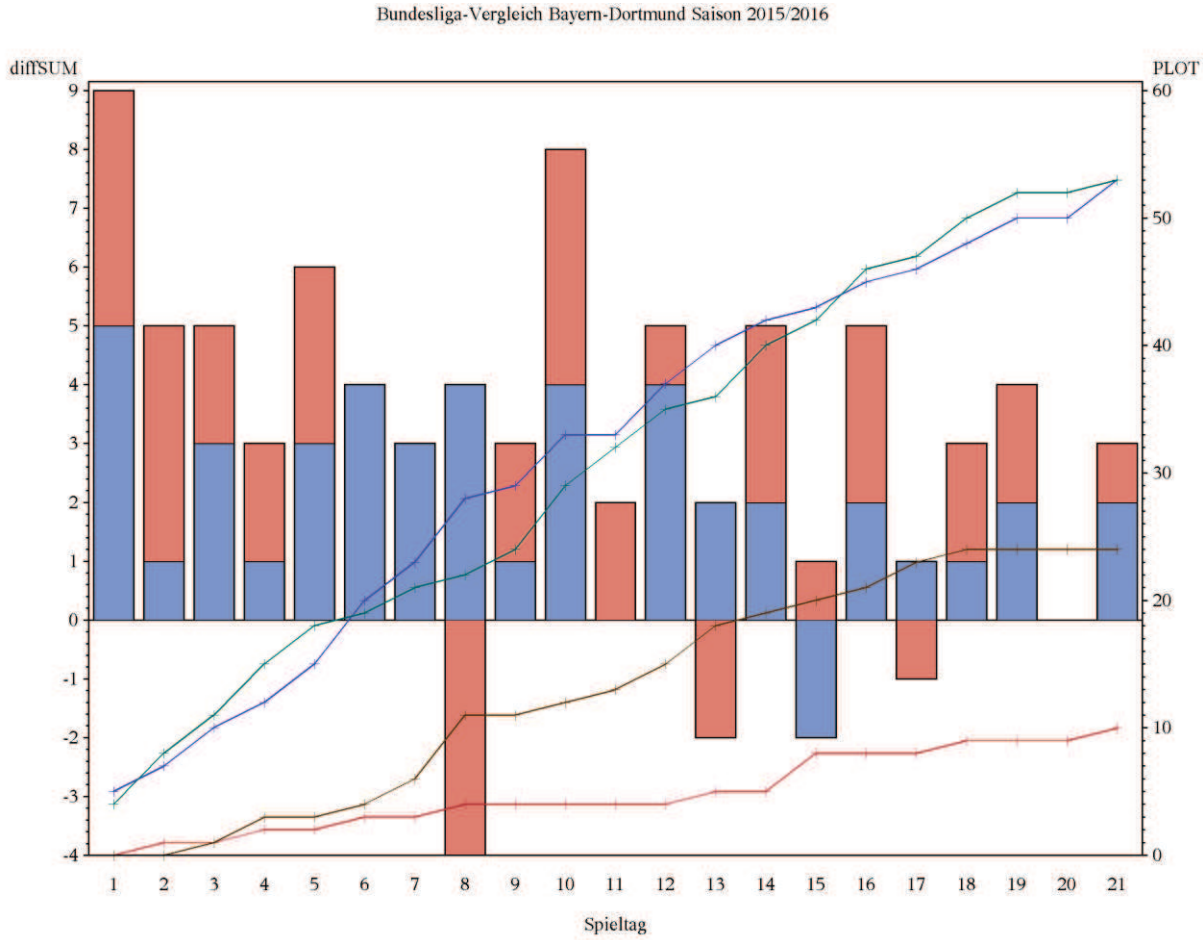

Abbildung 4: Graphik mit GBARLINE ohne zusätzliche Formatierungen

2.2 Möglichkeiten zur Verbesserung der Graphik

Es gibt eine Vielzahl Möglichkeiten, die bisher erzeugte Graphik zu verbessern. Die Formatierung der x- bzw. y-Achsen ist ebenso denkbar wie eine angepasste Auswahl der Farben und Füllmuster oder eine benutzerdefinierte Legende. Auch der Hintergrund kann angepasst oder es können Annotierungen eingefügt werden. Der folgende finale Code wird danach im Detail besprochen, um zu zeigen, welche Befehle welchen Effekt in der Graphik bewirken.

```
filename grafout 'c:\temp\ksfe1.png';
goptions reset=all device=png300 target=png300 rotate=landscape
         gsfname=grafout gsfmode=replace ctext=black ftext="Times"
         htext=2 gunit=pct lfactor=2;

proc gbarline data=mybuli(where=(saison=2015));
  title h=3 j=c 'Bundesliga-Vergleich Bayern-Dortmund Saison 15/16';
  bar spieltag / discrete subgroup=verein type=sum sumvar=diff
                 maxis=axis1 raxis=axis2
                 legend=legend1 coutline=same;

  plot / type=sum sumvar=bayern_t raxis=axis3 legend=legend2;
  plot / type=sum sumvar=bayern_g raxis=axis3;
  plot / type=sum sumvar=bvb_t raxis=axis3;
  plot / type=sum sumvar=bvb_g raxis=axis3;

  axis1 label=(h=2.5 'Spieltag') order=(1 to 21 by 1);
  axis2 label=(a=90 h=2.5 'Tordifferenz') order=(-10 to 10 by 1);
  axis3 label=(a=270 h=2.5 'Tore') order=(0 to 60 by 10);

  pattern1 c=darkred v=solid;
  pattern2 c=darkyellow v=solid;

  symbol1 c=darkred v=star i=join l=1 h=2;
  symbol2 c=darkred v=square i=join l=1 h=2;
  symbol3 c=darkyellow v=star i=join l=2 h=2;
  symbol4 c=darkyellow v=square i=join l=2 h=2;

  legend1 label=none value=('Bayern München' 'Borussia Dortmund')
          position=(outside bottom center) down=1 across=2 frame;
  legend2 label=none value=('Tore Bayern' 'Gegentore Bayern'
          'Tore Dortmund, 'Gegentore Dortmund') position=(outside
          bottom center) down=2 across=2 frame;
run;
quit;
```

2.2.1 Globale Graphikoptionen

Mit dem GOPTIONS-Befehl werden verschiedene globale Optionen festgelegt, die das Aussehen aller folgenden Graphiken bestimmt. Mit den DEVICE= und TARGET= Optionen wird der Graphiktreiber, d.h. der Dateityp für die Ausgabe der Graphik festgelegt. PNG300 ist das Kürzel für png-Dateien mit einer Auflösung von 300 dpi, was für die meisten Zwecke eine genügend gute Darstellung erwarten lässt.
Für die ROTATE= Option sind die Werte PORTRAIT bzw. LANDSCAPE erlaubt, mit denen die Ausrichtung der Graphik bestimmt ist.
Die Optionen GSFNAME= und GSFMODE= beziehen sich auf den Namen und Modus (REPLACE bzw. APPEND) der Ausgabedatei.
Mit den Optionen CTEXT=, FTEXT= und HTEXT= definiert man die Farbe, den Zeichensatz und die Schriftgröße für Text in den folgenden Graphiken. Die hierbei verwendete Einheit für die Schriftgröße (und auch für weitere Größenangaben in der Folge)

wird mit GUNIT= festgelegt. PCT bedeutet dabei Prozent bezogen auf die gesamte Graphikfläche. Weitere Werte sind z.B. CELLS (Standardwert), CM, IN (Inch) und PT (Punkte).
Mit LFACTOR=2 wird eine Liniendicke von doppelter Stärke wie vordefiniert verlangt.

2.2.2 Erstellung der Balken und Linien

Die Befehle für das Balkendiagramm in GBARLINE lautet BAR, wobei die Syntax sehr ähnlich zu der von den HBAR- bzw. VBAR-Befehlen in der Prozedur GCHART ist. Neben den notwendigen Befehlen (siehe 2.1) werden Referenzen zu den Achsendefinitionen (MAXIS=, RAXIS=) und der Legende (LEGEND=) angegeben. Mit COUTLINE=SAME wird definiert, dass die die Balken begrenzenden Linien die gleiche Farbe haben sollen wir die Füllung des Balkens. Hier kann auch eine bestimmte Farbe eingegeben werden.
In GBARLINE können durch eine beliebige Anzahl an PLOT-Befehlen zusätzliche Plots definiert werden, die mit dem Balkendiagramm zusammen in einer Graphik dargestellt werden. Bei den PLOT-Befehlen werden weitere Achsen- (RAXIS=) und Legenden-Definitionen (LEGEND=) ergänzt.

2.2.3 Achsendefinitionen

Alle drei in der Graphik dargestellten Achsen (linke und rechte y-Achse, x-Achse) werden über AXISn-Befehle modifiziert. Mit dem LABEL= Befehl wird der Text (und die Formatierung) der Achsenbeschriftungen angegeben (H= definiert die Schriftgröße und A= einen Winkel, um den die gesamte Beschriftung gedreht werden soll). ORDER= legt die darzustellenden Werte der Achse (Minimum, Maximum und Schrittweite) fest.

2.2.4 Füllmuster und Symbole

Mit PATTERNn werden Farbe (C=) und Muster (V=) der Balkenfüllung definiert. Beim Muster sind die Werte EMPTY und SOLID sowie für linierte Füllmuster Werte in der Form STYLE<DENSITY> möglich. Für STYLE kann L, R oder X verwendet werden und DENSITY erlaubt ganze Zahlen von 1 bis 5.
Symbole werden entsprechend durch SYMBOLn-Befehle eingestellt. In unserem Programm werden die Farbe (C=), Symbolart (V=), Interpolationsart (I=), Linientyp (L=) und Symbolgröße (H=) angegeben. Es ist empfehlenswert, bei SYMBOL auf jeden Fall immer Farbe und Symbolart anzugeben. Für die möglichen Werte der einzelnen Befehle kann das SAS-Handbuch benutzt werden.

2.2.5 Legenden

Legenden werden mit LEGENDn-Befehlen definiert. Mit LABEL= wird die Legendenbeschriftung, mit VALUE= die Legendenwerte festgelegt. Die Platzierung der Legende geschieht mit dem POSITION= Befehl, der drei Werte für die verschiedenen Möglichkeiten, eine Legende zu setzen, erlaubt. Dies sind OUTSIDE bzw. INSIDE in Bezug auf

die Graphikfläche, BOTTOM bzw. TOP und LEFT, CENTER bzw. RIGHT für die Lokalisation der Legende. Mit DOWN= bzw. ACROSS= wird die Anzahl der Zeilen bzw. Spalten, die der Legende zur Verfügung stehen, angegeben. FRAME zeichnet schließlich einen Rahmen um die Legende.

3 SG-Prozeduren

3.1 Verwendete Daten

Die Datenbasis entspricht inhaltlich der für die Prozedur gbarline beschriebenen Datenbasis, jedoch –wie bereits beschrieben- in einer vertikalen Struktur, s. Abb. 2.

Die Datentabelle besteht aus den 4 numerischen Feldern:

- Gruppe (User-SAS-Format gruppe)
- Spieltag
- Tore entsprechend Gruppe
- Tordifferenz pro Spieltag und Mannschaft

Das User-SAS-Format `gruppe` ist ein numerisches SAS-Format mit 8 in der Datenansicht sichtbaren textuellen Ausprägungen, was über die Systemoption `fmtsearch` im `WORK.Formats`–Catalog hinterlegt ist.

```
options fmtsearch=(WORK.Formats) locale=german_germany;
proc format lib=WORK.Formats;
     value gruppe
             1='Bayern Tore 2014'
               2='Bayern Gegentore 2014'
              3='Dortmund Tore 2014'
               4='Dortmund Gegentore 2014'
              5='Bayern Tore 2015'
               6='Bayern Gegentore 2015'
              7='Dortmund Tore 2015'
               8='Dortmund Gegentore 2015';
      value gruptordif
               2='Bayern Tordifferenz 2014'
               4='Dortmund Tordifferenz 2014'
               6='Bayern Tordifferenz 2015'
               8='Dortmund Tordifferenz 2015';
run;
```

3.2 Statistical Graphics-Prozeduren im Überblick

Außerhalb des lizenzpflichtigen SAS/GRAPH-Moduls stehen dem SAS-Anwender 5 graphische Prozeduren (kurz „SG-Prozeduren“) in SAS/BASE zur Verfügung.

Tabelle 1: "Give me 5" SG-Prozeduren in SAS/BASE

Prozedur	Einsatzbereich
SGPANEL	Balkendiagramme, Box-Plots, Histogramme, Loess-Kurven
SGPLOT	Box-Plots, Histogramme, Regressionsplots, Scatterplots, Linienplots, transparente Balkendiagramme, Landkarten (Alternative zu PROC GMAP)
SGSCATTER	Scatterplots, Panel-Scatterplots, Verteilungskurven
SGRENDER	Template-Wiedergabe aus SG-Prozeduren anhand Input-Daten
SGDESIGN	verarbeitet Daten mit ODS Graph Designer Datei (SGD)

Für das graphische Layout kommt insbesondere die Prozedur PROC TEMPLATE zum Einsatz, zu der im Enterprise Guide über den ODS GRAPHICS DESIGNER eine Syntaxhilfe angeboten wird, was der Diagrammpalette in Excel ähnelt und im Nachgang erklärt wird.

Beispiel-Galerien mit SAS-Codes befinden sich auf der SAS-Support-Seite im Internet (https://support.sas.com/sassamples/graphgallery/PROC_SGPLOT.html). Auf der angegebenen Seite zur Prozedur SGPLOT wird auch die Erstellung von Landkarten analog der SAS/GRAPH-Prozedur GMAP beschrieben.

Die Definition von Graphik-Titeln und –Fußnoten erfolgt außerhalb und jeweils vor der jeweiligen SG-Prozedur. Im Unterschied zu den SAS/GRAPH-Prozeduren ist zum Beenden der Prozedur nur ein „run;“, aber kein „quit;“ notwendig.
Achsendefinitionen wiederum finden innerhalb der SG-Prozeduren –also zwischen „proc“ und „run“- statt, je nach SG-Prozedur mit unterschiedlicher Nomenklatur, s. nachfolgende Programmbeispiele.

Für die Beschriftung der `xaxis` (x-Achse) werden die `values` (Spieltage) über eine Macrovariable `Spieltage` für die korrekte Reihenfolge übergeben.

```
proc sql noprint;
     select distinct Spieltag into :Spieltage
      separated by "' '"
     from Daten.Buli(keep=Spieltag);
      select distinct min(Tordifferenz), max(Tordifferenz)
      into :Tordifferenz1,:Tordifferenz2
     from Daten.Buli(keep=Tordifferenz where=(Tordifferenz is not
null));
quit;

data _null_;
     call symput ('Spieltage',"'"!!"&Spieltage."!!"'");
      length Tordifferenz $32767;
      retain Schritt 5;
      do _I_=Schritt*floor(&Tordifferenz1./Schritt) to
Schritt*ceil(&Tordifferenz2./Schritt) by Schritt;
```

```
Tordifferenz=left(trim(Tordifferenz)!!byte(32)!!"'"!!compress
(put(_I_,best.))!!"'");
      end;
      call symput ('Tordifferenz',trim(Tordifferenz));
run;
```

Mit Ausnahme der Prozedur sgscatter werden die SG-Prozeduren zur Darstellung der Fußballdaten im Folgenden beschrieben.

3.3 PROC SGPLOT – Darstellung der Fußballdaten

Über das `group`-Statement lassen sich bei der Prozedur sgplot mehrere Kurven mit direkt beigefügten Kurvenbeschriftungen darstellen.

```
*Kurvendiagramm;

libname Daten "<Datenpfad mit Fußballdaten>";

title1 j=center
"Bundesliga Vergleich Bayern-Dortmund Saison 2014/2015 und
2015/2016";
title2 j=center "sgplot-series (Saison 2015/2016)";

proc sgplot data=Daten.Buli(where=(Gruppe>=5))
autolegend noborder
tmplout="%sysfunc(pathname(Daten))\templates_buli_sgplot_series.sas"
;

   series x=Spieltag y=Tore/
          group=Gruppe
            curvelabel
            curvelabelpos=max
            lineattrs=(pattern=1);

   xaxis
     label="Spieltag"
     labelattrs=(size=8 weight=bold)
     type=discrete
      values=(&Spieltage.)
     valueattrs=(size=6) ;

   yaxis
    label="Tore"
    labelattrs=(size=8 weight=bold)
     type=discrete
    valueattrs=(size=8);

run;
```

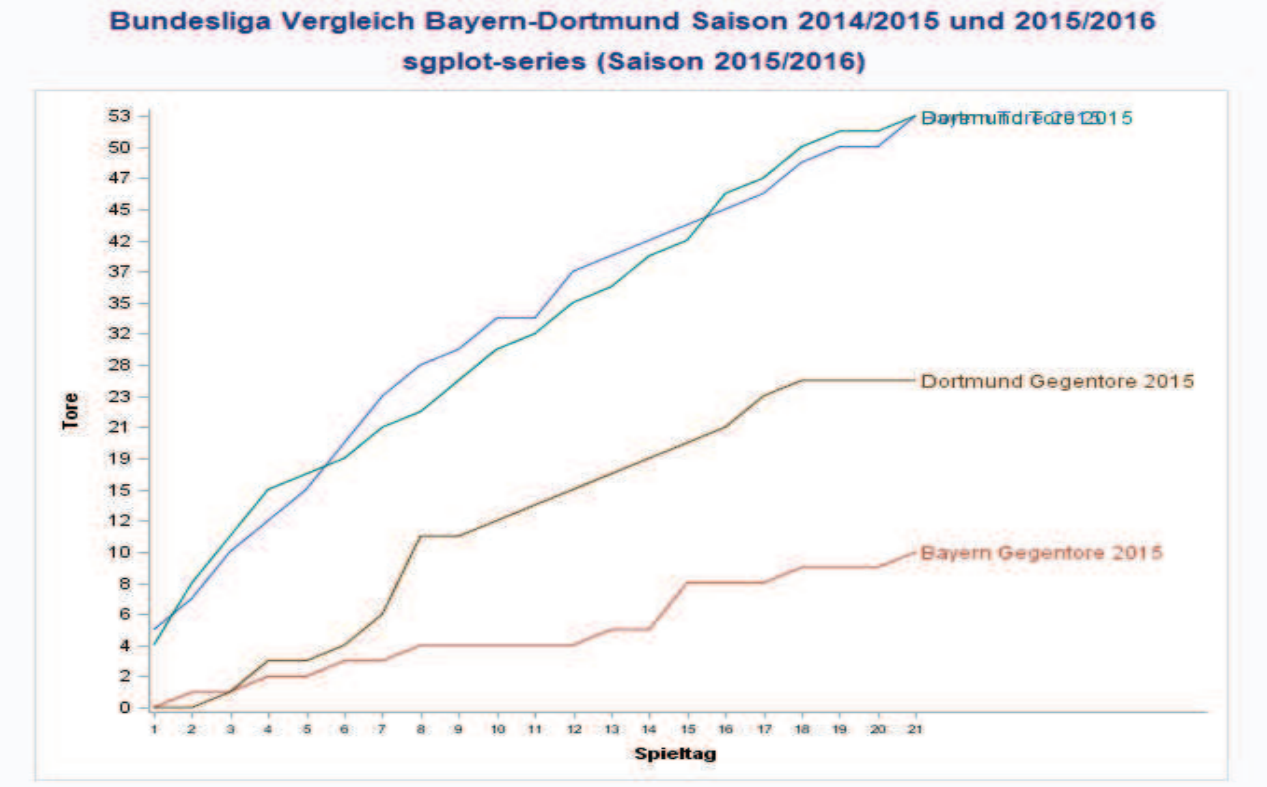

Abbildung 5: Graphik mit SGPLOT

Aus der Abbildung 5 ist erkennbar, dass es bei den Kurvenbeschreibungen zu Textkollisionen kommen kann, die sich nur durch Datenaufbereitung oder eine andere Achsenskalierung vermeiden lassen. Neben diesem Nachteil gibt es aber auch Vorteile wie die Template-Generierung zur weiteren Verwendung oder die Möglichkeit diverser Style-Optionen (Farben, Schriftgröße, Font, etc.).

3.4 PROC SGRENDER – Darstellung der Fußballdaten

Die proc sgrender stellt unter den SG-Prozeduren das Hauptanwendungsgebiet der vorab definierten proc template-Prozedur dar. Zuvor definierte Titel und Fußnoten sollten vor der proc template-Definition entfernt oder über dynamische Werte definiert werden, da bei gleichem Template Titel und Fußnoten aus der Template-Datei übernommen werden.

```
proc template;
define statgraph sgplot;
begingraph / subpixel=on;

EntryTitle halign=center "Bundesliga Vergleich Bayern-Dortmund
Saison 2014/2015 und 2015/2016" / halignCenter=Graph;
EntryTitle halign=center "sgplot-series (Saison 2015/2016)" /
textattrs=(size=GraphLabelText:fontsize) halignCenter=Graph;

layout overlay / walldisplay=(fill)
x2axisopts=(labelFitPolicy=Split) xaxisopts=( Label="Spieltag"
labelFitPolicy=Split LabelAttrs=( Size=8 Weight=bold)
```

```
TickValueAttrs=( Size=6) type=discrete discreteopts=(
tickvaluelist=(  "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"
"13" "14" "15" "16" "17" "18" "19" "20" "21")
TickValueFitPolicy=SplitRotate ) ) yaxisopts=( Label="Tore"
LabelAttrs=( Size=8 Weight=bold) TickValueAttrs=( Size=8)
type=discrete ) x2axisopts=(labelFitPolicy=Split);

   SeriesPlot X='Spieltag'n Y='Tore'n / primary=true Group='Gruppe'n
Lineattrs=( Pattern=1) CurveLabel=Gruppe CurveLabelLocation=Inside
CurveLabelPosition=Max LegendLabel="Tore";
endlayout;
endgraph;

end;
run;

proc sgrender data=daten.buli
         (where=(gruppe lt 5)) template=sgplot;
run;
```

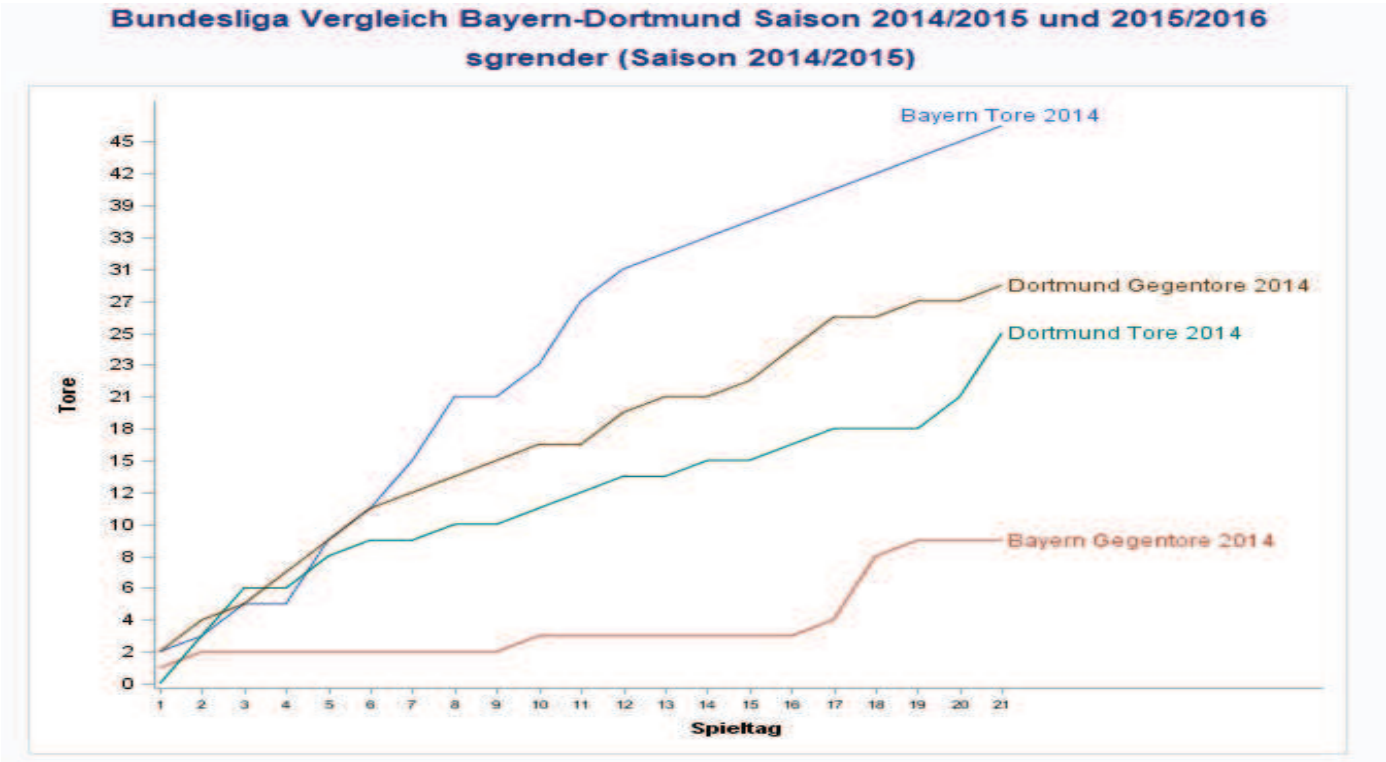

Abbildung 6: Graphik mit SGRENDER

3.5 PROC SGPANEL – Darstellung der Fußballdaten

Über proc sgpanel lassen sich bequem Balkendiagramme darstellen. Zu den Stärken von proc sgpanel zählt die Nebeneinander-Anordnung der Balken. Die Farbauswahl der Balken lässt sich bequem über die Optionen `datacolors` und `datacontrastcolors` (Beispiel: `datacolors=(cxcc0000 blue) datacontrastcolors=(cxcc0000 blue)`) einstellen.

Im Unterschied zu gängigen SAS-Prozeduren ist eine Reihenfolge der zur Prozedur gehörenden Statements einzuhalten. Dies gilt für das `panelby` -Statement, wenn man nicht `styleattrs` verwendet. Steht das `panelby` –Statement nicht als Erstes hinter dem proc-Statement, kommt es in SAS 9.4 zu einer Fehlermeldung und die Grafik wird nicht ausgegeben: „ ERROR: The PANELBY statement must be the first statement.“

```
proc sgpanel data=daten.buli
     (where=(tordifferenz ne . and gruppe le 5));

  panelby spieltag / layout=columnlattice
          onepanel colheaderpos=bottom rows=1
          novarname noborder;
  label gruppe = "Vergleich";

  styleattrs datacolors=(cxcc0000 blue)
             datacontrastcolors=(cxcc0000 blue);

  vbar gruppe / group=gruppe response=tordifferenz
                barwidth=1 stat=sum nostatlabel;
  colaxis display=none;
  rowaxis;
run;
```

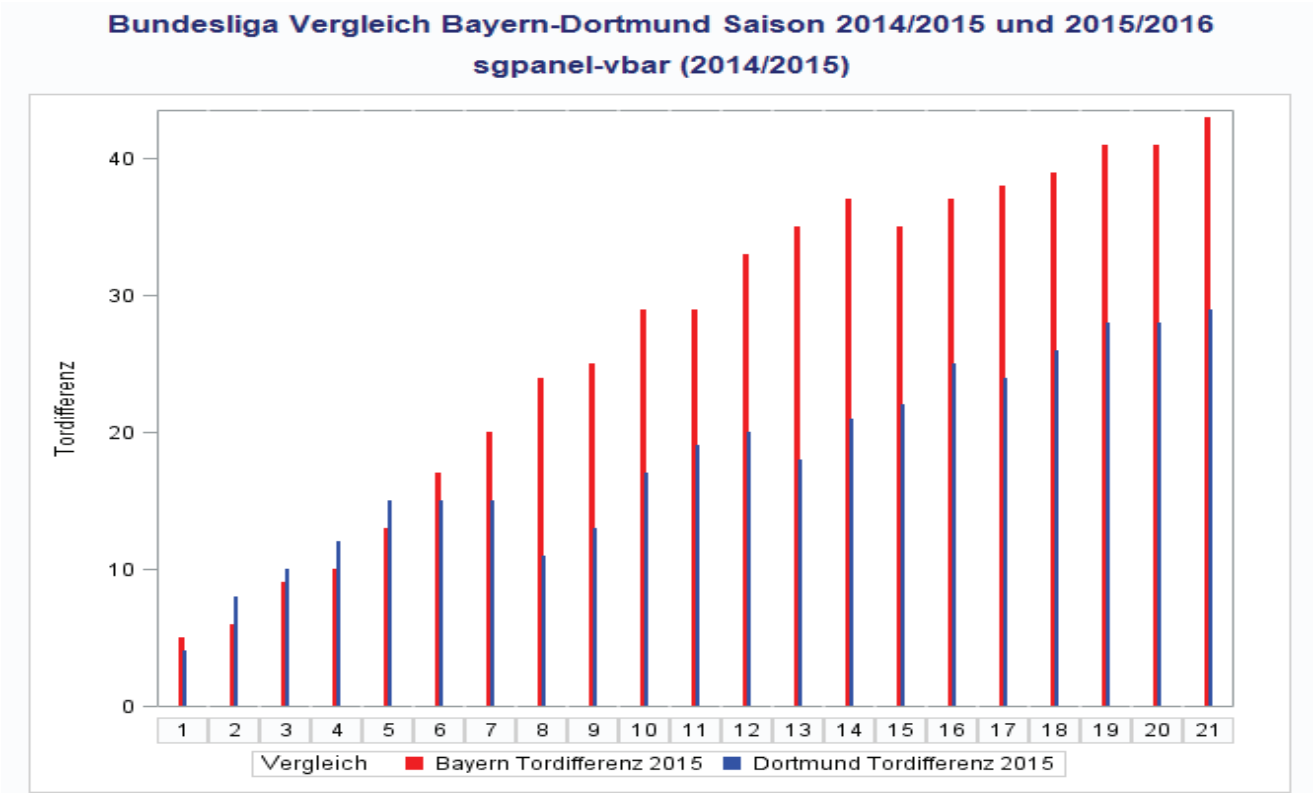

Abbildung 7: Graphik mit SGPANEL

3.6 PROC SGDESIGN – Darstellung der Fußballdaten

Für proc sgdesign muss eine Textdatei (Datei-Endung sgd) erstellt werden, in der das Design festgelegt wird. Dies geschieht über das im Enterprise Guide hinterlegte Feature „ODS Graphics Designer“, was der Diagrammerstellung in Excel in der Vorgehensweise in großen Teilen entspricht.

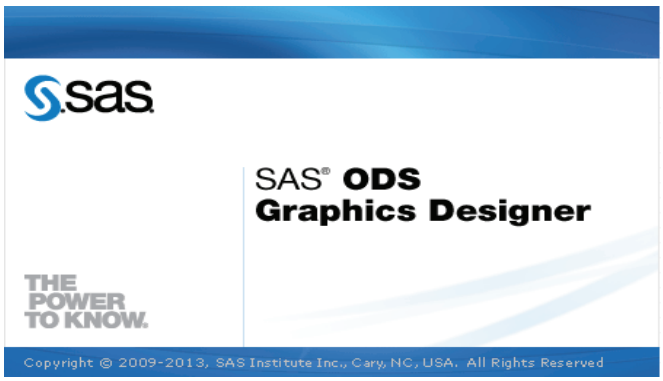

Abbildung 8: ODS Graphics Designer

Der „ODS Graphics Designer“ befindet sich als Menüpunkt im Enterprise Guide in der vorliegenden Version unter „Anwendungsroutinen“/“Grafiken“/“ODS Graphics Designer öffnen …“

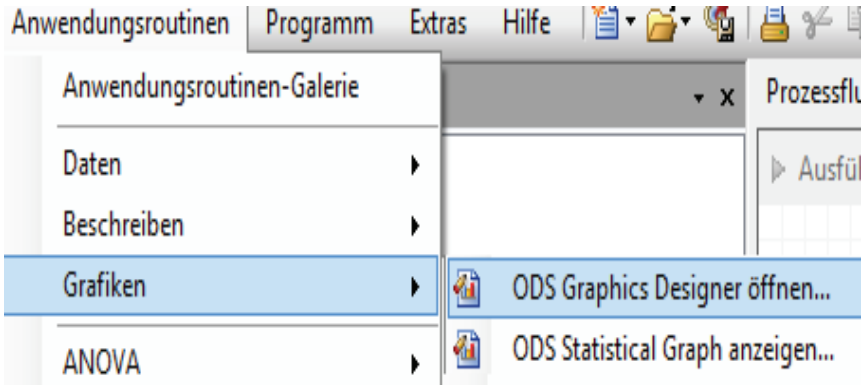

Abbildung 9: Menüpunkt „ODS Graphics Designer“

Der „ODS Graphics Designer“ bietet eine große Auswahl von Graphik-Designs, u.a. Histogramme, Zeitreihen, Balken- und Streuungsdiagramme. Allgemeine Parameter wie Titel oder Achsenbeschriftungen lassen sich mit dyn(Parameter) „dynamisieren“, sprich ähnlich zu SAS-Makrovariablen an die Prozedur sgrender weitergeben und dynamisieren.

Nicht dynamisieren lässt sich der Dateiname, der erst bei der Graphik auslösenden proc sgdesign verwendet wird: Als „Hilfstabellen“ für den „ODS Graphics Designer“ bieten sich dennoch zahlreiche Tabellen aus der SAS-Bibliothek SASHELP an, über die eine Graphik-Vorschau „ODS Graphics Designer“ möglich ist (in der nachfolgenden Abbildung die Tabelle SASHELP.CARS). Dynamische Werte inkl. Tabellenname und Feldnamen lassen sich später mit dem dynamic-Statement in den Prozeduren sgdesign und sgrender auf die Zieltabelle ändern.

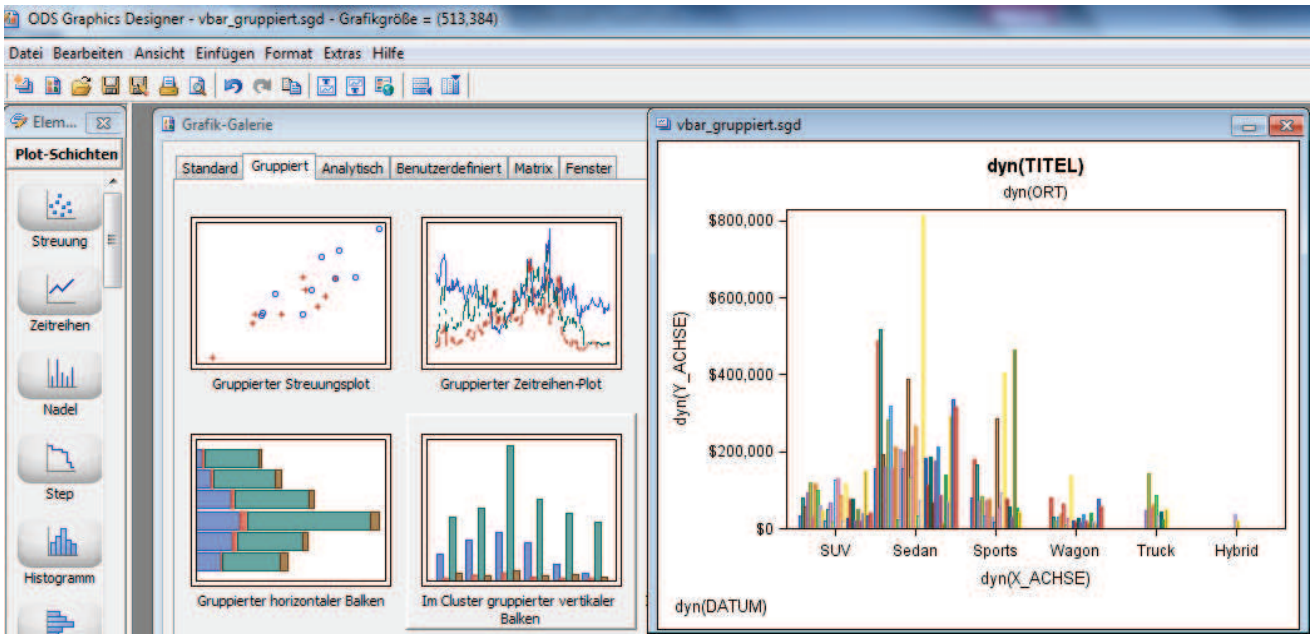

Abbildung 10: Dynamisieren mit dem „ODS Graphics Designer“

Nach der Design-Definition lässt sich die Graphik-Vorlage als sgd-Datei speichern und zu einem späteren Zeitpunkt wiederöffnen.

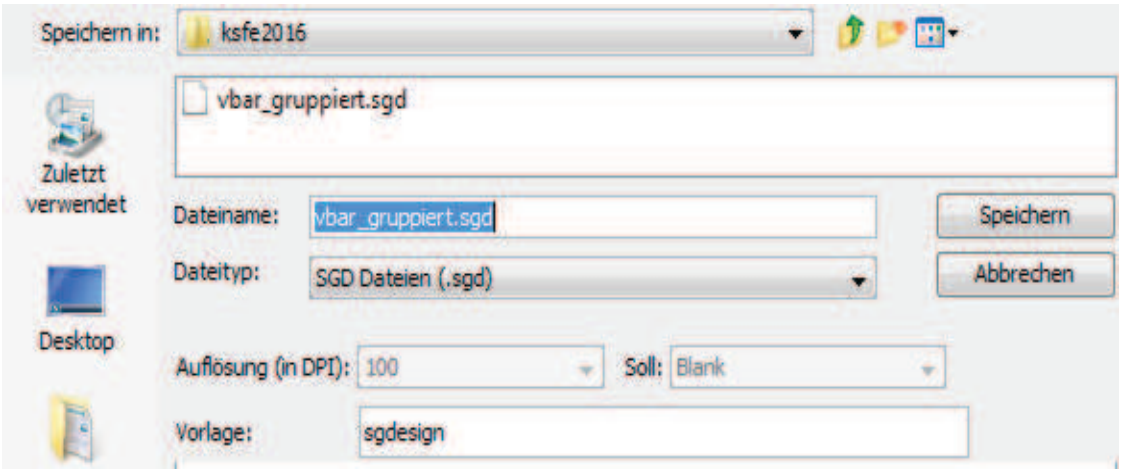

Abbildung 11: Erstellung der sgd-Datei

Die gewählten Graphik-Optionen, die dynamischen Parameter und der Template-Name (im Beispiel `sgdesign`) sind in der sgd-Datei in einer xml-Struktur abgelegt, nachfolgend der Datei-Header der sgd-Datei.

```
<Meta>
        <Template>sgdesign</Template>
        <Datasets>
            <Dataset>SASHELP.CARS</Dataset>
        </Datasets>
        <DynVars>
            <DynVar name="_TYPE"
type="any">SASHELP.CARS.TYPE</DynVar>
            <DynVar name="_INVOICE"
type="any">SASHELP.CARS.INVOICE</DynVar>
            <DynVar name="_MAKE"
type="any">SASHELP.CARS.MAKE</DynVar>
```

```
        </DynVars>
        <DynScalars>
            <DynScalar name="DATUM"/>
            <DynScalar name="ORT"/>
            <DynScalar name="TITEL"/>
            <DynScalar name="Y_ACHSE"/>
            <DynScalar name="X_ACHSE"/>
        </DynScalars>        <Style>LISTING</Style>
        <TemplateCode><![CDATA[proc template;define statgraph
sgdesign;dynamic _TYPE _INVOICE _MAKE;dynamic DATUM ORT TITEL
Y_ACHSE X_ACHSE;
…
```

Die gewählten Graphik-Optionen sind in der sgd-Datei in einer xml-Struktur abgelegt, nachfolgend der Datei-Header der sgd-Datei. Die sgd-Datei wird über die Option `sgd="<Name der sgd-Datei>"` in der proc sgdesign angesprochen.

```
proc sgdesign data=Daten.Buli
sgd="%sysfunc(pathname(Daten))\vbar_gruppiert.sgd";
run;
```

Änderungen der dynamischen Parameter sind über die proc sgrender möglich:

```
*Aufruf Bundesliga-Daten mit sgrender;
proc sgrender data=Daten.Buli template=sgdesign;
   dynamic DATUM="%sysfunc(date(),deudfdd10.)"
      TITEL="KSFE %eval(%sysfunc(date(),year4.)+2)"
      ORT="Hawai"
     Y_ACHSE="Tore" _INVOICE="Tore"
     X_ACHSE="Spieltag" _TYPE="Spieltag"
     _MAKE="Gruppe";
run;
```

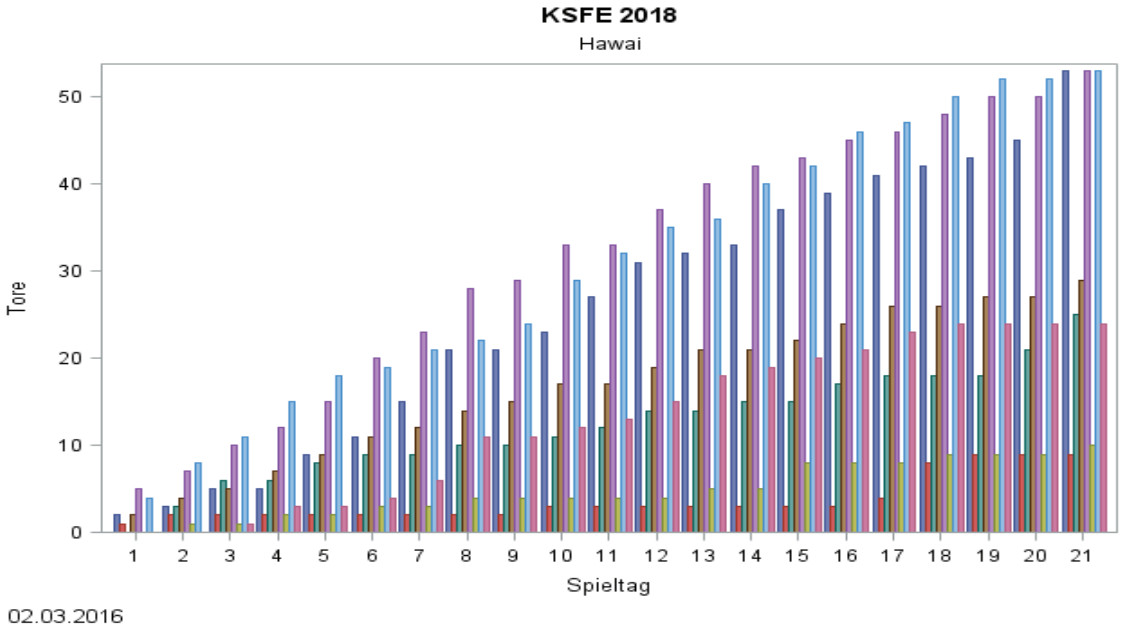

Abbildung 12: Graphik-Ausgabe mit sgd-Datei