

# Beyond the Borders: Einbinden externer Shapefiles zur Erweiterung kartografischer Darstellungen

Kathrin Flunkert  
Institut für Biometrie  
Medizinische Hochschule Hannover  
Carl-Neuberg-Straße 1  
30625 Hannover  
Flunkert.Kathrin@mh-hannover.de

Stefanie Ernst  
Institut für Biometrie  
Medizinische Hochschule Hannover  
Carl-Neuberg-Straße 1  
30625 Hannover  
Ernst.Stefanie@mh-hannover.de

## Zusammenfassung

Zur Erstellung kartografischer Darstellungen werden von SAS Koordinatendateien vieler Staaten bereitgestellt. Allerdings sind die Möglichkeiten der Abbildung kleinerer Gebietsausschnitte und verschiedener Begrenzungen gering, da beispielsweise in Bezug auf Deutschland lediglich Kreis-, Länder- und Bundesgrenzen verfügbar sind. Zur Lösung dieses Problems kann die Prozedur PROC MAPIMPORT verwendet werden, die es ermöglicht, eigene ESRI-Shapefile-Dateien einzulesen und zu einem SAS/GRAPH-Map Data Set zu konvertieren. Dazu sind im Internet zahlreiche freie Geodaten verfügbar. Als Beispiel wird hier die Erstellung einer Deutschlandgrafik in SAS 9.3 demonstriert, die die Häufigkeitsverteilung von Patienten einer klinischen Prüfung innerhalb der Postleitzahlengebiete veranschaulicht. Weiterhin werden die Markierungen von Orten auf einer Stadtkarte anhand ihrer Koordinaten, die Verwendung des %MAPLABEL-Makros zur Gebietskennzeichnung sowie die Legendenerstellung und -ausrichtung aufgezeigt.

**Schlüsselwörter:** SAS/GRAPH, Choroplethenkarten, Makro, %MAPLABEL, PROC MAPIMPORT, PROC GREMOVE, PROC GMAP, ESRI-Shapefile

## 1 Einführung

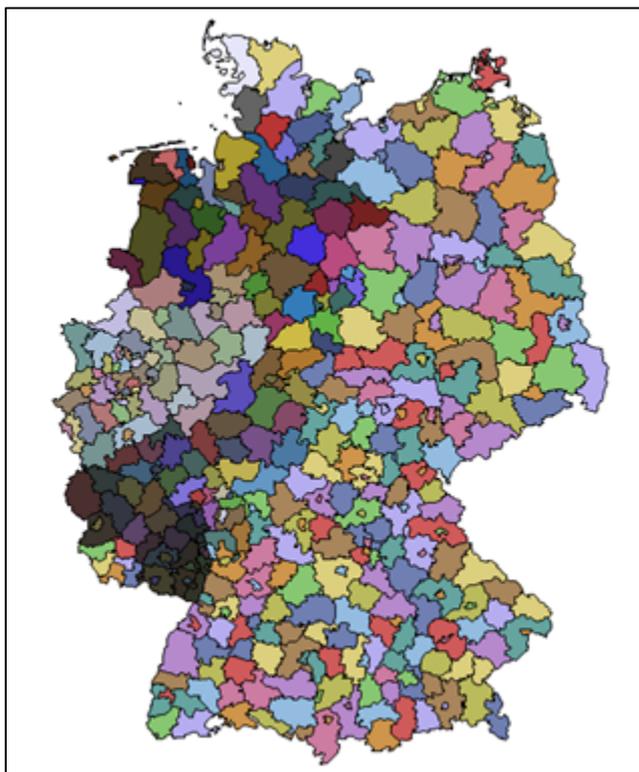
### 1.1 Verwendung der SAS Map Data Sets

Ein Bestandteil von SAS/GRAPH ist die Library Maps, welche zahlreiche Data Sets zur Verfügung stellt, auf deren Basis kartografische Darstellungen mit der Prozedur GMAP erstellt werden können. Die Data Sets liegen paarweise vor, so dass zu einem Map Data Set jeweils ein Response Data Set gehört. Das Map Data Set enthält X- und Y-Koordinaten, die zu Polygonen – in Bezug auf Deutschland zu Landkreisen, Bundesländern oder Gesamtdeutschland – verbunden werden können. Das Response Data Set, das über eine ID-Variable mit den Koordinatendaten verknüpft ist, liefert die auf der Karte dazustellenden Informationen der einzelnen Polygone. Es kann um eigene Variablen erweitert oder auch auf Basis eigener Daten gesondert angelegt werden.

Eine einfache Deutschlandkarte auf Landkreisebene mit den von SAS zur Verfügung gestellten Koordinaten wird durch die folgende Syntax erstellt. Das Ergebnis dieses Prozeduraufrufs ist in Abbildung 1 dargestellt.

**Beispiel 1:**

PROC GMAP	
MAP = maps.germany	Map Data Set
DATA = maps.germany2;	Response Data Set
ID id;	Variable, die die Polygone identifiziert
CHORO id /	einzelne Einfärbung jeder ID-Fläche
discrete nolegend;	ohne Klassenbildung und Legende
RUN; QUIT;	



**Abbildung 1:** Deutschland auf Kreisebene

Mit der Prozedur GREMOVE können Polygone zu größeren Flächen zusammengefasst werden. Dazu muss das ursprüngliche Map Data Set eine weitere ID-Variable enthalten, die diese Gebiete kennzeichnet. In `maps.germany` ist neben der Variablen `id` (ID der Landkreise) auch die Variable `state` (ID der Bundesländer) enthalten. Die folgende Syntax legt ein neues Map Data Set namens `landesebene` an, in dem die Landkreise zu Bundesländern zusammengefasst werden. Wie bei der Verwendung eines BY-State-

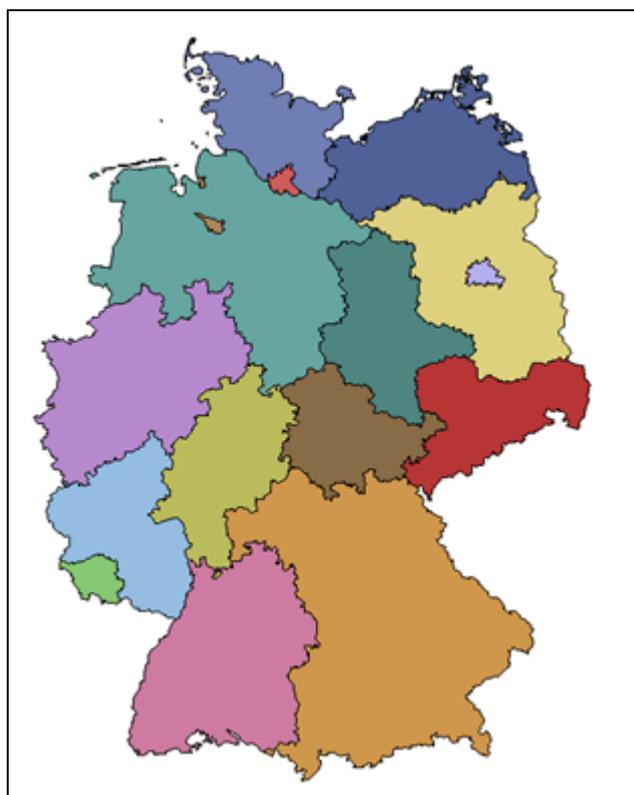
ments üblich, muss das unter dem Argument DATA angegebene Data Set nach dieser Variablen sortiert sein.

### Beispiel 2:

PROC GREMOVE	
DATA = maps.germany	ursprüngliches Map Data Set
OUT = landesebene;	Name unter dem SAS das zusammengefasste Map Data Set ablegen soll
ID id;	bisherige ID
BY state;	neue ID, die die zusammengefassten Polygone kennzeichnet
RUN;	

Wird nun erneut die Prozedur GMAP aufgerufen und das neue Map Data Set sowie die Variable state als ID übergeben, werden die Bundesländer farbig dargestellt.

```
PROC GMAP MAP = landesebene DATA = maps.germany2;
  ID state;
  CHORO state / discrete nolegend;
RUN; QUIT;
```



**Abbildung 2:** Deutschland auf Länderebene

## 1.2 Anlegen eigener Map Data Sets

Neben den von SAS zur Verfügung gestellten Map Data Sets können auch eigene Koordinatendateien erstellt werden. Dabei besteht keine Beschränkung auf reale geografische Koordinaten. Es können stattdessen auch geometrische Formen definiert werden, wozu die einzelnen Punkte einem Koordinatensystem entnommen werden müssen. Im folgenden Beispiel 3 soll das in Abbildung 3 gezeigte Data Set in Form eines in fünf Ebenen geteilten Dreiecks grafisch dargestellt werden. Abbildung 4 skizziert den gewünschten Aufbau des Endergebnisses und ermöglicht das Ablesen der Koordinatenpunkte. Diese müssen nun in SAS eingelesen werden, wobei jede der fünf Ebenen eine eigene ID erhält und durch Spezifizierung aller Umfangspunkte fünf eigenständige Polygone definiert werden. Zu beachten ist die Reihenfolge, in der die Koordinaten angegeben werden, da sie in dieser Abfolge zu Polygonen verbunden werden.

### Beispiel 3:

```
/* the map data set; */
DATA triangle_map;
INPUT id x y;
DATALINES;
1 2 4
1 2.5 5
1 3 4
2 1.5 3
2 2 4
2 3 4
2 3.5 3
3 1 2
3 1.5 3
3 3.5 3
3 4 2
4 0.5 1
4 1 2
4 4 2
4 4.5 1
5 0 0
5 0.5 1
5 4.5 1
5 5 0
; RUN;
```

	id	value
1	1	first
2	2	second
3	3	third
4	4	fourth
5	5	fifth

Abbildung 3: Data Set triangle

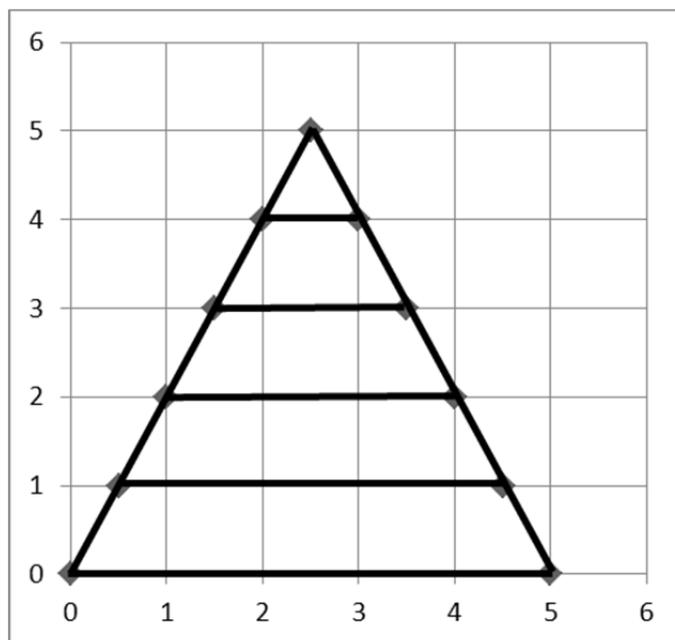


Abbildung 4: Koordinatenkreuz mit Dreieck

Um die einzelnen Ebenen des Dreiecks in der Grafik mit den Werten der Variablen `value` (siehe Abbildung 3) zu beschriften, kann das von SAS zur Verfügung gestellte `%MAPLABEL`-Makro verwendet werden. Dazu muss zunächst das `%ANNOMAC`-Makro aufgerufen werden, welches die Annotate-Makros kompiliert. Anschließend

steht das %MAPLABEL-Makro mit den im Folgenden beschriebenen Parametern zur Verfügung.

```
%ANNOMAC;
```

%MAPLABEL (	
triangle_map,	Name des Map Data Set
triangle,	Data Set, das die Beschriftungsvariable enthält (Response Data Set)
beschriftung,	Output Data Set, das durch das Makro erstellt wird
value,	Variable, die die Beschriftungen enthält
id,	ID-Variable, die in beiden Data Sets vorkommen muss
FONT = 'Arial/bold',	Schriftart
COLOR = black,	Schriftfarbe
SIZE = 4 );	Schriftgröße

Das %MAPLABEL-Makro legt nun das Data Set `beschriftung` in der WORK-Direktory an, dessen Aufbau in Abbildung 5 zu sehen ist. Annotate Data Sets können auch manuell per Data Step angelegt werden. Die Verwendung des %MAPLABEL-Makros erleichtert dies lediglich, da die Koordinaten der Gebietsmittelpunkte nicht selbst ermittelt werden müssen.

	position	xsys	ysys	when	hsys	text	function	style	color	size	x	y	id	value
1	5	2	2	a	3	first	label	'Arial/bold'	black	4	2.5	4.5	1	first
2	5	2	2	a	3	second	label	'Arial/bold'	black	4	2.5	3.5	2	second
3	5	2	2	a	3	third	label	'Arial/bold'	black	4	2.5	2.5	3	third
4	5	2	2	a	3	fourth	label	'Arial/bold'	black	4	2.5	1.5	4	fourth
5	5	2	2	a	3	fifth	label	'Arial/bold'	black	4	2.5	0.5	5	fifth

**Abbildung 5:** Annotate Data Set `beschriftung`

Um den fünf Ebenen jeweils eine eigene Füllfarbe zuzuweisen, sind entsprechende PATTERN zu definieren. Mit `v=s` (kurz für `value=solid`) wird das Versehen der Flächen mit Mustern verhindert und über das Argument `c` (color) Farbwerte übergeben. Eine anschauliche Übersicht aller SAS/Graph Farben liefert der SAS Online Support. [1]

```
PATTERN1 v=s c=BWH;
PATTERN2 v=s c=VLIGB;
PATTERN3 v=s c=LIGB;
PATTERN4 v=s c=MOGB;
PATTERN5 v=s c=STGB;
```

Die Syntax der abschließenden Grafikerstellung unterscheidet sich nicht von den in Beispiel 1 und 2 erzeugten Karten. Es wird nun lediglich noch die Annotate-Option spezifiziert, um das mit dem %MAPLABEL-Makro angelegte Beschriftungs-Data Set zu verwenden.

```
PROC GMAP DATA = triangle MAP = triangle_map;
  ID id;
  CHORO id / discrete nolegend annotate=beschriftung;
RUN; QUIT;
```

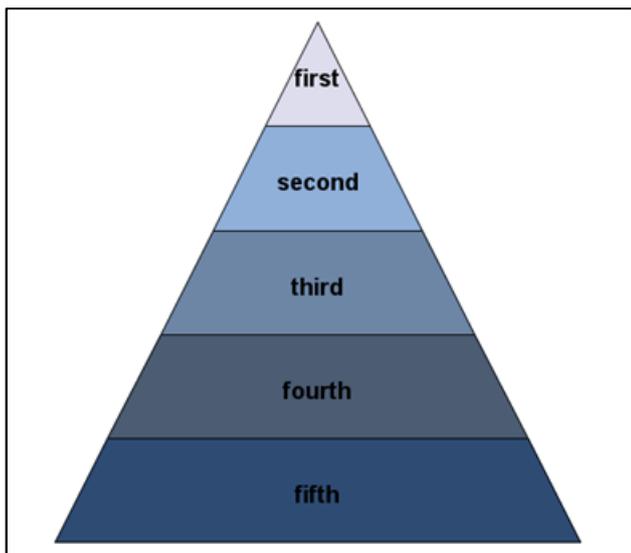


Abbildung 6: mit PROC GMAP erstellte Grafik eines Dreiecks

## 2 Problemstellung

Wie in den beiden einleitenden Beispielen demonstriert, können mit den von SAS zur Verfügung gestellten Koordinatendateien innerhalb Deutschlands lediglich Kreis- und Länderflächen angesprochen und dargestellt werden. Was aber, wenn davon abweichende Grenzverläufe gewünscht sind oder individuelle Teilausschnitte abgebildet werden sollen? Diese Polygone per Hand in einem Data Set zu definieren wäre mit einem viel zu großen Aufwand verbunden.

Praktisch stellte sich diese Frage, als im Rahmen einer Auswertung von Registerdaten die Anzahl eingeschlossener Patienten grafisch auf Postleitzonenebene (erste Stelle der Postleitzahl) dargestellt werden sollte (siehe Abbildung 7). Der Versuch, die 544 Landkreise, deren Umrisse von

	1. Stelle der PLZ	Anzahl Patienten
1	0	714
2	1	448
3	2	1238
4	3	5979
5	4	3657
6	5	6367
7	6	3265
8	7	1322
9	8	2582
10	9	1050

Abbildung 7: Data Set Freq\_data

SAS bereitgestellt werden, den jeweiligen Postleitzonen zuzuweisen, führte zu einem inakzeptablen Ergebnis, da sich die Postleitzonen nicht aus Landkreisen zusammensetzen, sondern davon abweichende Grenzlinien besitzen. Im Folgenden wird daher die Möglichkeit beschrieben, externe Shapefiles mit der MAPIMPORT Prozedur in SAS einzulesen.

### 3 Einlesen externer Shapefiles

Um individuellere kartografische Darstellungen mit SAS/GRAPH zu erstellen, besteht die Möglichkeit, mit der MAPIMPORT Prozedur eigene Koordinatendateien einzulesen. Die Daten müssen dafür in Form von ESRI-Shapefiles, einem speziellem Dateiformat für Geodaten, vorliegen. Im Internet sind solche Koordinatendateien in großer Zahl zu finden und können meist kostenlos heruntergeladen werden. Grenzkordinaten aller deutschen Postleitzahlgebiete werden beispielsweise auf <http://arnulf.us/PLZ> zum Download angeboten. Die Daten dürfen frei verwendet werden, und es besteht keinerlei Einschränkung in Bezug auf die Nutzungsrechte [2]. Weiterhin bieten zahlreiche kommerzielle Anbieter Shapefiles zum Kauf an.

Ein ESRI-Shapefile besteht immer aus mindestens drei Dateien:

- einer Hauptdatei (.shp),
- einer Indexdatei (.shx)
- und einer dBASE Tabelle (.dbf) [3].

Nach Speichern und Entpacken der heruntergeladenen Dateien können diese mit PROC MAPIMPORT in SAS eingelesen werden. Hierzu müssen alle Shapefile-Dateien in einem gemeinsamen Ordner liegen, als Argument ist SAS allerdings lediglich die Hauptdatei zu übergeben. Unter OUT muss der Name des Map Data Sets angegeben werden, in das die Koordinaten importiert werden sollen.

```
PROC MAPIMPORT DATAFILE="C:\Downloads\PLZ\post_pl.shp" OUT=plz_map;
RUN;
```

Das in dieser Prozedur erstellte Data Set `plz_map` enthält über 1 Mio. Datensätze, die jeweils eine X- und Y-Koordinate beschreiben (siehe Abbildung 8). Werden diese Punkte je Ausprägung der ID-Variablen `PLZ99` verbunden, entstehen Polygone, die die Flächen der Postleitzahlgebiete darstellen.

	X	Y	SEGMENT	PLZ99	PLZ99_N	PLZORT99
1	13.7189358	51.07600017	1	01067	1067	Dresden
2	13.72121532	51.07498173	1	01067	1067	Dresden
3	13.72244724	51.07323078	1	01067	1067	Dresden
4	13.7243034	51.07141494	1	01067	1067	Dresden
5	13.72696308	51.06893157	1	01067	1067	Dresden
6	13.727583	51.06863403	1	01067	1067	Dresden
7	13.72820184	51.06833649	1	01067	1067	Dresden

Abbildung 8: Auszug des Data Sets `plz_map`

Die MAPIMPORT Prozedur ermöglicht es, beliebige Shapefiles in SAS einzulesen. Mit den Statements RENAME, EXCLUDE, SELECT und ID können beim Importieren Spalten des Map Data Set umbenannt, Variablen aus- bzw. eingeschlossen sowie die Datensätze nach bestimmten IDs geordnet werden. Diese Anpassungen sind allerdings auch im Anschluss an das Importieren durch Anweisungen in weiteren DATA Steps möglich. Zur weiteren Verwendung importierter Koordinatendaten werden im Folgenden zwei Beispielkarten erstellt und detailliert beschrieben.

## 4 Beispiele erweiterter kartografischer Darstellungen mit importierten Shapefiles

### 4.1 Deutschlands Postleitzonen

Um auf Grundlage aller deutschen Postleitzahlen, die im vorherigen Abschnitt als `plz_map` importiert wurden, eine Grafik der Postleitzonen zu erstellen, müssen die jeweiligen Gebiete zusammengefasst werden. Hierzu wird die Gruppierungsvariable und neue ID-Variable `plz_eins`, die die erste Ziffer der jeweiligen Postleitzahl enthält, im Map Data Set angelegt. Mit PROC GREMOVE wird anschließend das Map Data Set `map_zonen` erstellt, das die Polygone der Postleitzahlen (identifiziert durch die Variable `PLZ99`) zu Postleitzonen (identifiziert durch die Variable `plz_eins`) zusammenfasst. Das Data Set `Freq_data` (siehe Abbildung 7) bildet nun das Response Data Set zu `map_zonen`, da beide die ID-Variable `plz_eins` beinhalten und ihre Inhalte darüber verknüpft werden können.

#### Beispiel 4:

```
DATA plz;
  SET plz_map;
  plz_eins = SUBSTR(PLZ99,1,1);
RUN;

PROC GREMOVE DATA = plz OUT = map_zonen;
  ID PLZ99;
  BY plz_eins;
RUN;
```

Die Einfärbung der Postleitzahlzonen soll anhand der Anzahl der rekrutierten Patienten erfolgen. Daher wird ein Format angelegt, welches drei Klassen definiert sowie drei entsprechende PATTERN vorbereitet.

```
PROC FORMAT;
  VALUE Freq_fmt
  low -< 1000 = '< 1000'
  1000 - 3000 = '1000 bis 3000'
  3001 - high = '>= 3000'; RUN;
```

```
PATTERN1 v=s c=Lavender;
PATTERN2 v=s c=VLIGB;
PATTERN3 v=s c=VLIB;
```

Die Legende wird in diesem Beispiel beim späteren Aufruf der Prozedur GMAP nicht mit der NOLEGEND-Option unterdrückt. Stattdessen werden folgende Formatierungs- und Ausrichtungsvorbereitungen getroffen.

LEGEND1	
LABEL=( position=top h=13pt)	Ausrichtung der Überschrift oberhalb der Legende und Formatierung der Schriftgröße
VALUE = (h=13pt c=black tick=3 f=symbol "B3"x f=Arial " 3000")	Formatierung der Schriftgröße und -farbe; Überschreiben des dritten Legendeneintrags, um das „≥“-Zeichen korrekt darstellen zu können.
ACROSS = 1	Legendeneinträge in einer Spalte
DOWN = 3	Anordnung der Legendeneinträge in drei Zeilen untereinander
POSITION=(right bottom )	Ausrichtung unten rechts innerhalb der Grafik
OFFSET = (-2, 2.5) cm	Verschieben der Legende um 2 cm nach links und 2.5 cm nach oben
MODE = reserve;	Verhindert das Überschneiden von Grafik und Legende (Platz für die Legende wird explizit reserviert)

Wie schon in Beispiel 3 wird auch diese Grafik mit dem %MAPLABEL-Makro annotiert. Die absoluten Häufigkeiten der Variablen `Freq_Pat` werden als Beschriftungen innerhalb der Grafik gewählt.

```
%ANNOMAC;

%MAPLABEL( map_zonen, /* Map Data Set */
           Freq_data, /* Response Data Set */
           maplabel, /* Output Data Set */
           Freq_Pat, /* Variable, die Beschriftung enthält */
           plz_eins, /* ID-Variable */
           FONT='Arial/bold', COLOR=black, SIZE=2.4);
```

Globale Formatierungen der Höhe und Breite der Gesamtgrafik sowie die Zuweisung einer Schriftart werden abschließend mittels GOPTIONS realisiert.

```
GOPTIONS vsize=13cm hsize=14cm ftext='Arial';
```

In der Prozedur GMAP werden dann alle in den vorherigen Schritten getroffenen Vorbereitungen vereint. Das Response Data Set `Freq_data`, welches die Patientenzahlen

enthält, und das Map Data Set `map_zonen`, das die X- und Y-Koordinaten liefert, werden an die Prozedur übergeben und die verknüpfende Variable `plz_eins` benannt. Um die Einfärbung der Gebiete anhand der drei angelegten Klassen vorzunehmen, wird der Variablen `Freq_Pat` das entsprechende Format zugewiesen. Die Beschriftung der Klassen wird in der zuvor definierten Legende angezeigt. Da die Grafik verzerrt ausgegeben werden kann, wenn SAS versucht, den zur Verfügung stehenden Platz gänzlich auszunutzen, werden außerdem explizite Größendimensionen spezifiziert, die korrekte Seitenverhältnisse garantieren.

```
TITLE1 height=14pt "Anzahl Patienten je Postleitzone";  
PROC GMAP DATA = Freq_data MAP = map_zonen;  
  ID plz_eins;  
  FORMAT Freq_Pat Freq_fmt.;  
  CHORO Freq_Pat / discrete  
          ANNOTATE = maplabel  
          COUTLINE = DAGRG /* graue Umrandung der Gebiete*/  
          LEGEND = legend1  
          XSIZE = 7.4 cm  
          YSIZE = 10 cm;  
RUN; QUIT;
```

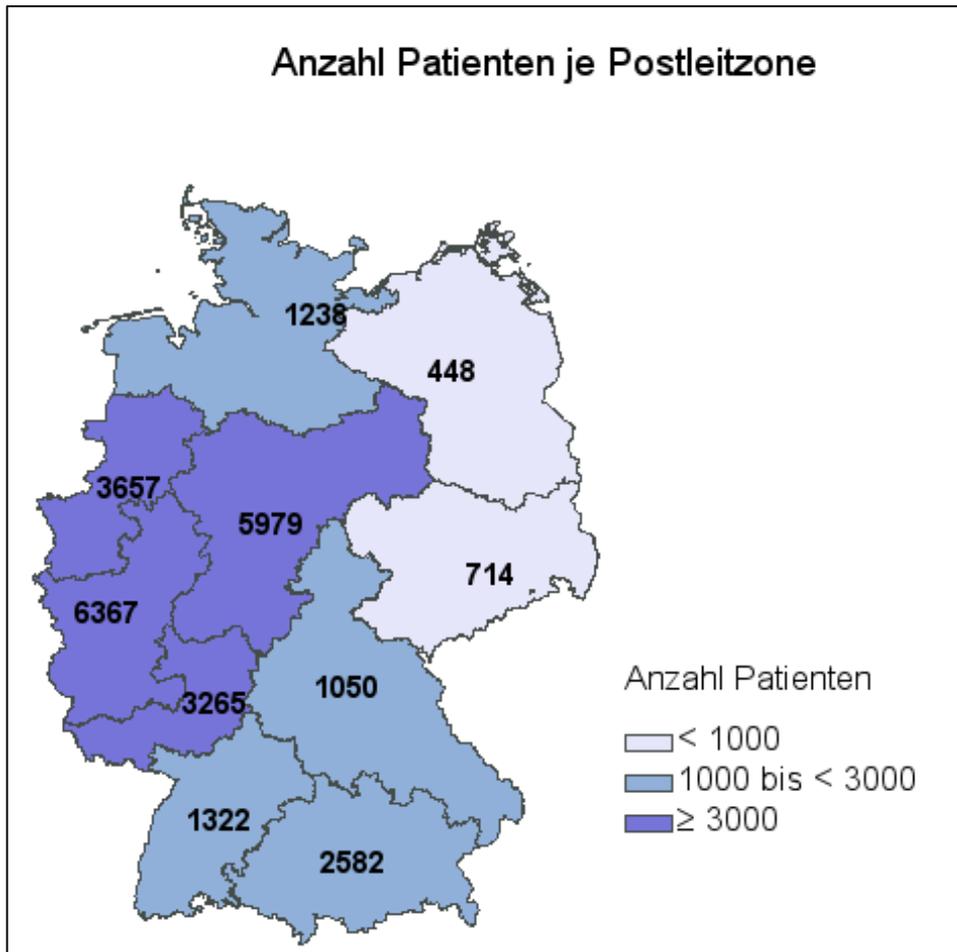


Abbildung 9: Anzahl Patienten je Postleitzone

## 4.2 Ulmer Stadtteile

Nachdem das vorherige Beispiel die Erstellung einer Grafik der gesamten Bundesrepublik mit von den SAS Maps abweichenden inneren Grenzen demonstrierte, soll nun abschließend auf die Möglichkeit der Verwendung externer Shapefiles zur Darstellung kleinerer Gebietsausschnitte hingewiesen werden. Im Folgenden entsteht eine Karte von Ulm gegliedert in Stadtteile, auf der über das %MAPLABEL-Makro hinausgehende Beschriftungen vorgenommen werden.

Unter <http://www.ulmapi.de/> können Gauss-Krueger-Shapefiles mit den Koordinaten der Stadtteile Ulms heruntergeladen werden. Es handelt sich um Open Data, deren Verwendung in Drittanwendungen explizit erlaubt ist. [4] Das Einlesen der Datei erfolgt erneut unter Angabe des Pfades zur .shp-Datei und eines Namen für das Output Data Set mittels MAPIMPORT. Ein Ausschnitt des angelegten Data Sets ist in Abbildung 10 erkennbar. Da dieses zunächst nicht nach der ID ST\_NR sortiert ist, erfolgt anschließend die Anwendung der SORT Prozedur.

### Beispiel 5:

```
PROC MAPIMPORT DATAFILE="C:\Downloads\Stadtteile_Gesamt_270209.shp"
  OUT=ulm;
RUN;
```

	X	Y	SEGMENT	ST_NAME	ST_NR
1	3573843.8582	5364380.4372	1	Mitte	ST 11
2	3573831.2569	5364453.1506	1	Mitte	ST 11
3	3573820.6828	5364503.0939	1	Mitte	ST 11
4	3573804.3421	5364545.3532	1	Mitte	ST 11
5	3573782.4742	5364596.7376	1	Mitte	ST 11
6	3573761.8064	5364631.3137	1	Mitte	ST 11
7	3573745.4641	5364660.1269	1	Mitte	ST 11

Abbildung 10: Auszug des Data Sets ulm

```
PROC SORT DATA=ulm;
  BY ST_NR;
RUN;
```

In Beispiel 4 lag zuerst das Response Data Set vor, für das anschließend das Map Data Set mit der Verknüpfungsvariable erstellt wurde. Im aktuellen Fall ist der Weg andersherum, da auf Basis des importierten Map Data Sets nun eine Response-Datei angelegt werden muss. Hierfür werden mit der SQL Prozedur die ST\_NR und die Stadtteilnamen in ein eigenes Data Set extrahiert. Die Variable SEGMENT dient der späteren Einfärbung aller Teilgebiete in der Prozedur GMAP.

```
PROC SQL;
  CREATE TABLE ulm_label as
  SELECT DISTINCT ST_NR, ST_NAME, SEGMENT FROM ulm ORDER BY ST_NR;
QUIT;
```

Anschließend kann wie gehabt das %MAPLABEL-Makro aufgerufen werden. Dieses legt das Annotate Data Set `beschriftung` auf Grundlage des Map Data Sets `ulm` und des soeben erstellten Response Data Sets `ulm_label` an.

```
%ANNOMAC;  
  
%MAPLABEL( ulm,          /* Map Data Set */  
           ulm_label,    /* Response Data Set */  
           beschriftung, /* Output Data Set */  
           ST_NAME,      /* Variable, die Beschriftung enthält */  
           ST_NR,        /* ID-Variable */  
           FONT='Arial/bold', COLOR=OLGR, SIZE=2.4);
```

Das durch das Makro angelegte Annotate Data Set kann anschließend noch verändert und/oder erweitert werden. Sollen zusätzliche Punkte auf der Karte markiert werden, besteht die Möglichkeit, das Data Set um entsprechende Datensätze zu erweitern. Zu beachten ist lediglich, dass eventuell einige Variablenlängen erweitert werden müssen, damit Inhalte nicht abgeschnitten werden. Im Folgenden wird der Speicherplatz der Variablen `function` und `color` heraufgesetzt, um anschließend weitere Einträge in das Annotate Data Set `beschriftung` einzufügen. Über die Namen der Stadtteile hinaus, sollen die Universität – der Tagungsort der KSFE 2013 – sowie der Hauptbahnhof mit einem Stern markiert (`function='symbol' text='V'`) und die Orte entsprechend beschriftet werden. Die X- und Y-Werte der Orte sind einem Online Tool entnommen. Unter [http://www.deine-berge.de/umrechner\\_koordinaten.php](http://www.deine-berge.de/umrechner_koordinaten.php) kann eine Adresse spezifiziert werden, zu der das System die Koordinaten ermittelt. Zur Bedeutung aller Annotate-Variablen sei aufgrund der zahlreichen Möglichkeiten individueller Formatierungen auf die SAS Hilfe verwiesen.

```
DATA beschriftung;  
  length function $6. color $15.;  
  set beschriftung;  
RUN;  
  
PROC SQL;  
  INSERT INTO beschriftung (function, color, position, xsys, ysys,  
                           when, hsys, text, style, size, x, y)  
  VALUES('symbol', 'GOLD', '5', '2', '2', 'a', '3',  
         'V', 'marker', 6, 3570087.015, 5365660.451)  
  VALUES('label', 'SteelBlue', '5', '2', '2', 'a', '3',  
         'KSFE 2013', 'Arial/bold', 7, 3569800, 5365900)  
  VALUES('symbol', 'red', '5', '2', '2', 'a', '3',  
         'V', 'marker', 4, 3570087.015, 5365660.451)  
  VALUES('label', 'black', '3', '2', '2', 'a', '3',  
         'Universität', 'Arial/bold', 3, 3569800, 5365900)  
  VALUES('symbol', 'red', '5', '2', '2', 'a', '3',  
         'V', 'marker', 4, 3572734.677, 5362856.109)
```

```
VALUES('label', 'black', '3', '2', '2', 'a', '3', 'Hauptbahnhof',
      'Arial/bold', 3, 3572734.677, 5362856.109);
QUIT;
```

Abschließend erfolgt die Definition eines einzelnen PATTERN in hellgrau. Durch die Verwendung der Variablen `SEGMENT` im `CHORO`-Statement der `GMAP` Prozedur wird die gesamte Grafik dementsprechend eingefärbt, da alle Datensätze den Segmentwert 1 haben. Die Beschriftungen innerhalb der Abbildung stammen wiederum aus dem eben erweiterten Annotate Data Set.

```
PATTERN1 v=s color=lightgrey;

PROC GMAP MAP = ulm DATA = ulm_label;
  ID st_nr;
  CHORO segment / discrete
    ANNOTATE = beschriftung
    COUTLINE = black
    nolegend ;

RUN; QUIT;
```



Abbildung 11: Die Stadt Ulm

## **5 Fazit**

SAS/GRAPH liefert mit der Library Maps bereits zahlreiche Koordinatendateien, auf deren Basis kartografische Darstellungen erstellt werden können. Allerdings sind mit diesem Material die Grenzen des Möglichen noch nicht erreicht. Die Prozedur MAPIMPORT liest ESRI-Shapefiles aus externen Quellen in SAS ein und ermöglicht die Abbildung verschiedenster Gebietsverläufe und detailreicher Kartenausschnitte. In Verbindung mit den zahlreichen Optionen der Prozedur GMAP und der Annotate Facility entstehen anschauliche und individuell gestaltbare kartografische Darstellungen.

### **Literatur**

- [1] SAS Support Website: TS-688: Defining Colors using Hex Value [Online] Available at: <http://support.sas.com/techsup/technote/ts688/ts688.html> [Accessed 07.02.2013].
- [2] A. Christl: PLZ [Online] (Updated 29 May 2012) Available at: <http://arnulf.us/PLZ> [Accessed 08.09.2012].
- [3] Environmental Systems Research Institute, Inc.: ESRI Shapefile Technical Description: An ESRI White Paper, 1998 [Online] Available at: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf> [Accessed 08.02.2013].
- [4] Die Ulmer Open Data Initiative: UlmAPI.de – Open Data in Ulm [Online] Available at: <http://www.ulmapi.de/> [Accessed 13.09.2012].