

Einführung in die Prozedur FORMAT mit praktischen Anwendungen

Christoph Oeldorf, Carina Ortseifen
Rechenzentrum der
Ruprecht-Karls-Universität Heidelberg
Im Neuenheimer Feld 293
69120 Heidelberg
Carina.Ortseifen@urz.uni-heidelberg.de

Zusammenfassung

Formate und Informat sind im Vergleich zu anderen Statistikprogrammen ein herausragendes Alleinstellungsmerkmal der SAS Software. Sie dienen dazu, Werte beim Lesen aus bzw. beim Schreiben in SAS-Variablen umzuwandeln. Damit lassen sich zum einen Nicht-Standardwerte, wie Texte mit mehr als 8 Zeichen oder Datumswerte in SAS-Tabellen abspeichern, aber auch Werte in anderer Form anzeigen, z.B. Komma statt Punkt als Dezimaltrennzeichen, Messwerte mit Maßeinheit (1,25 m) bzw. einfache Gruppierungen durchgeführt werden.

In diesem Beitrag werden die grundlegenden Anweisungen vorgestellt und anhand kleiner Beispiele der Einsatz der Prozedur FORMAT gezeigt. Der Leser erfährt, wo Formate und Informat eingesetzt werden können, wie eigene Formate definiert, wo diese gespeichert werden und wie sich das ganze mit Hilfe von SAS-Tabellen automatisieren lässt.

Schlüsselwörter: PROC FORMAT, geschachteltes Format, VALUE-Anweisung, PICTURE-Anweisung, INFORMAT-Anweisung, Klassenbildung, SAS-Explorer, Properties-Fenster, Format, Informat

1 Einleitung

Formate und Informat sind im Vergleich zu anderen Statistikprogrammen ein herausragendes Alleinstellungsmerkmal der SAS Software¹. Ausgabeformate, oder kurz: Formate, dienen dazu, Werte beim Lesen aus SAS-Variablen umzuwandeln. Informat, auch Einleseformate, wandeln die Werte dagegen beim Schreiben in SAS-Variablen um. Anstelle 1,23456 wird beispielsweise mit einem passenden Format 1,24 angezeigt. Ein anderes Format könnte dazu dienen, die kodierten Werte „1“ und „2“ der Variable Geschlecht als „weiblich“ und „männlich“ darzustellen.

In folgenden Abschnitt wird die in den Beispielen verwendete SAS-Tabelle klasse vorgestellt. Anschließend, in Abschnitt 3, wird der Einsatz der Anweisung FORMAT auf-

¹ IBM SPSS kennt Wertelabels, die mit den SAS-Formaten dahingehend vergleichbar sind, dass die Variablenwerte in anderer als der gespeicherten Form dargestellt werden. Allerdings sind die Wertelabels nur mit jeweils einer Variablen verbunden und können nicht wie die SAS-Formate, für mehrere Variablen, auch in verschiedenen Tabellen und Bibliotheken eingesetzt werden.

gezeigt. In Abschnitt 4 folgen Syntax und Beispiele zum Aufruf der Prozedur FORMAT. Im letzten Abschnitt 5 werden weitere Optionen der Prozedur FORMAT vorgestellt, die das Abspeichern von Formaten erlauben. Abgerundet wird der Beitrag dann mit einer ausführlichen Literaturliste, die zum Weiterlesen anregen soll.

2 Der Beispieldatensatz: Klasse

Mit dem folgenden Datenschnitt erzeugen wir eine SAS-Tabelle namens *klasse*, die – der Einfachheit wegen – in der Bibliothek WORK abgelegt wird.

```
Data klasse;
  Informat Name Vorname $20. Gebdaten Ddmmyy10.;
  Input Name Vorname Geschlecht Groesse Note Gebdatum;
  Datalines;
Bauer Natalie 1 1.56 2.0 01.01.1960
Graf Jochen 2 1.67 1.3 06.09.1996
Metzger Jürgen 2 1.80 2.7 21.02.1995
Schäfer Inga 1 1.71 3.0 30.06.1996
Walter Maike 1 1.66 1.7 23.11.1995
Bletzer Jörg 2 1.87 2.3 30.03.1995
;
```

Bereits in diesem Datenschnitt werden Informate verwendet: \$20. und DDMMYY10.. \$20. für die Vor- und Nachnamen und DDMMYY10. für das Geburtsdatum der Schüler.

Die Zuweisung der Informate zu den drei Variablen Name, Vorname und Gebdaten erfolgt mit der Anweisung INFORMAT. (Die übrigen Variablen werden mit dem Standard eingelesen und benötigen kein Extra-Informat.) Alternativ könnte man auch die Anweisung ATTRIB verwenden:

```
Data klasse;
  Attrib name vorname Informat=$20. gebdaten Informat=ddmmyy10.;
  Input name Vorname Geschlecht Groesse Note Gebdaten;
  ...
```

Die Anweisung ATTRIB ist dann von Vorteil, wenn nicht nur das Informat festgelegt werden soll, sondern auch Formate, Längen und andere Variablenmerkmale.

Es gibt noch eine dritte Variante, die hier auch erwähnt werden sollte, da sie in der Literatur sehr häufig zu finden ist: Angabe des Formats direkt in der Anweisung INPUT:

```
Data klasse;
  Input Name :$20. Vorname :$20. Geschlecht Groesse Note
  Gebdatum :ddmmyy10.;
```

Direkt im Anschluss an den Variablennamen werden die Informate übergeben. Vor der Informat-Angabe ist hier zusätzlich noch ein Doppelpunkt notwendig. Mit diesem Doppelpunkt wird die Länge des Informats als Maximalwert definiert. D.h. die tatsächliche Länge kann kürzer sein (z.B. fünf Zeichen bei „Bauer“).

Ganz gleich, welche der drei Varianten Sie vorziehen, Informate werden immer dann benötigt, wenn keine Standardwerte vorliegen. Standardtextwerte sind höchstens 8 Zeichen lang. Die Vor- und Nachnamen der Schüler können dagegen mehr Zeichen haben, daher sind hier die Informate notwendig. Für Datumsangaben benötigt man immer Informate (und Formate, vgl. dazu das folgende Beispiel und Abschnitt 3), denn als Zahlenwert enthält das Datum zu viele Punkte. (Würde man das Datum nur als Text abspeichern, würde der Informationsgehalt der Datumsangabe verloren gehen.)

Die Namen der Informate als auch die Namen der Formate beginnen stets mit einem \$-Zeichen, wenn sie für Werte von Textvariablen verwendet werden (\$20.). Formate und Informate für numerische Variablen haben kein \$-Zeichen (10.2, DDMMYY10.).

Der Zahlenwert 20 nach dem \$-Zeichen bzw. 10 nach DDMMYY bezeichnet die Länge des formatierten Wertes. Vor- und Nachname können 20 Zeichen lang sein, das Datum 10 Zeichen. (Die Punkte zur Abtrennung von Tag, Monat und Jahr zählen mit und das Jahr ist vierstellig.)

Im SAS-Code sind die Formatnamen durch den Punkt gekennzeichnet. Dieser steht meistens an dessen Ende (\$20., DDMMYY10.). Bei numerischen Formatbezeichnungen kann nach dem Punkt allerdings noch ein Zahlenwert folgen, der die Nachkommastellen bezeichnet, wie z.B. 10.2 für numerische Werte mit zwei Dezimalstellen.

Die Ausgabe der neuen SAS-Datei work.klasse mit der Prozedur PRINT sieht wie folgt aus:

Obs	Name	Vorname	Geschlecht	Groesse	Note	Gebdatum
1	Bauer	Natalie	1	1.56	2.0	0
2	Graf	Jochen	2	1.67	1.3	13398
3	Metzger	Jürgen	2	1.80	2.7	12835
4	Schäfer	Inga	1	1.71	3.0	13330
5	Walter	Maike	1	1.66	1.7	13110
6	Bletzer	Jörg	2	1.87	2.3	12872

Als Geburtsdatum gibt SAS jetzt nicht die ursprünglich eingetragenen Daten aus, sondern die intern gespeicherten, seit dem 1.1.1960 vergangenen Tage. (Daher auch die 0 in Zeile 1 bei Natalie Bauer.) Ein für uns lesbares Datum erscheint erst dann, wenn wir mittels der Anweisung FORMAT ein passendes Format verwenden.

3 Die Anweisung FORMAT

Formate werden mit Hilfe der Anweisung FORMAT in einem Daten- oder einem Prozedurschritt einer Variablen zugewiesen. Die ursprünglichen Werte der Variablen werden dadurch nicht verändert, aber die Erscheinungsform der Werte an das Format angepasst.

Die allgemeine Syntax dieser lokalen Anweisung FORMAT lautet:

```
Format variablenname formatname.;
```

Für unsere Beispieldatei können wir mit der Prozedur PRINT eine besser lesbare Liste erzeugen, wenn wir ein Datumsformat verwenden, hier DDMMYYYP10., „P“ für den Punkt zur Abtrennung, im Unterschied zum Standardtrennzeichen „/“. Außerdem wenden wir gleich noch zwei weitere Formate an: \$5. für die Variable Vorname und COMMAX10.2 für die Größe.

```
Proc Print Data=work.klasse;  
  Var name vorname groesse gebdatum;  
  Format vorname $5. groesse COMMAX10.2 gebdatum DDMMYYYP10. ;  
Run;
```

Im Ausgabefenster sieht das Ergebnis dann wie folgt aus:

Obs	Name	Vorname	Groesse	Gebdatum
1	Bauer	Natal	1,56	01.01.1960
2	Graf	Joche	1,67	06.09.1996
3	Metzger	Jürge	1,80	21.02.1995
4	Schäfer	Inga	1,71	30.06.1996
5	Walter	Maike	1,66	23.11.1995
6	Bletzer	Jörg	1,87	30.03.1995

In der Online-Hilfe finden Sie über das Menü *Help > SAS Help and Documentation* unter *SAS Products > Base SAS > SAS 9.2 Language Reference: Dictionary ... > Dictionary of Language Elements > Formats* bzw. *Informats* die Liste aller zur Verfügung stehenden Formate und Informate.

3.1 Formate im Daten- oder Prozedurschritt zuweisen?

Die Zuweisung von Informaten kann nur im Datenschnitt erfolgen, wenn Daten in SAS-Tabellen geschrieben werden. Daher können sowohl die ATTRIB- als auch die INFORMAT-Anweisung nur im Datenschnitt stehen, genau wie INPUT. Die FORMAT-Anweisung dagegen kann sowohl im Daten- als auch im Prozedurschritt ausgeführt werden.

In obigem Beispiel hatten wir FORMAT im PROC PRINT-Schritt eingesetzt. Die Variablen Vorname, Groesse und Gebdatum wurden in der angegebenen Form ausgegeben. Würden wir den Prozedurschritt ohne die Anweisung FORMAT wiederholen, kämen wieder die alten, unformatierten Werte zum Vorschein. Denn die Formatierung wurde nur im ersten, nicht aber im zweiten Prozedurschritt durchgeführt.

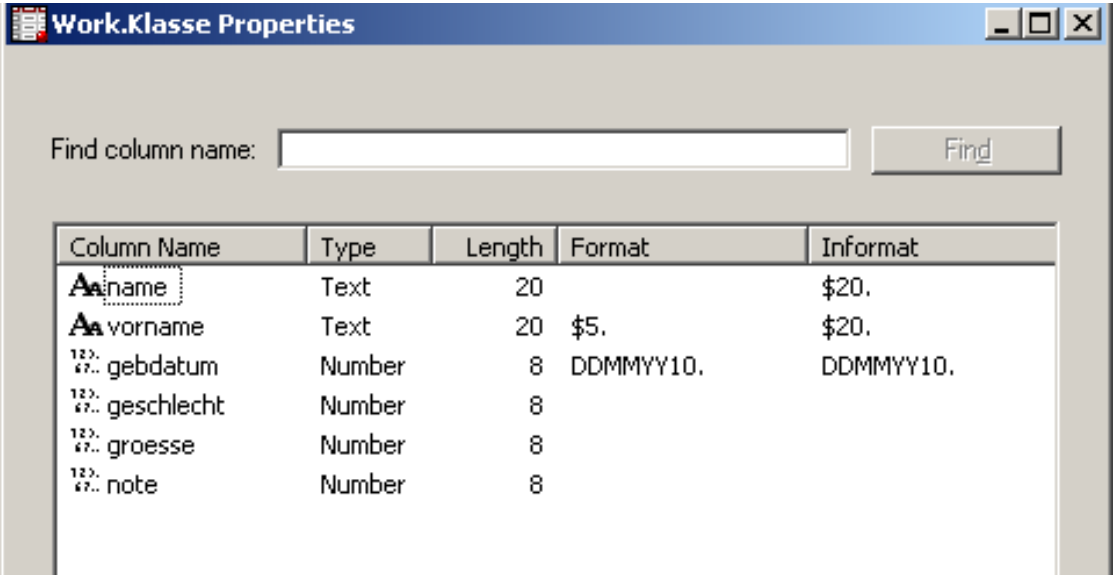
Möchte man die Formatierung permanent setzen, muss die Zuweisung im Datenschnitt erfolgen. Entweder gleich im Datenschnitt, mit dem die Tabelle klasse angelegt wird:

```
Data klasse;
  Informat Name Vorname $20. Gebdaten Ddmmyy10.;
  Format vorname $5. Gebdaten Ddmmyy10.;
  Input Name Vorname Geschlecht Groesse Note Gebdatum; ...
```

Oder in einem separaten Datenschnitt:

```
Data klasse;
  Set klasse;
  Format vorname $5. Gebdaten Ddmmyy10.;
Run;
```

Damit wird das Format fest in der SAS Tabelle abgelegt und ist für jeden anschließenden Daten- und Prozedurschritt verfügbar.



Column Name	Type	Length	Format	Informat
Aname	Text	20		\$20.
A vorname	Text	20	\$5.	\$20.
122. 67.. gebdatum	Number	8	DDMMYY10.	DDMMYY10.
122. 67.. geschlecht	Number	8		
122. 67.. groesse	Number	8		
122. 67.. note	Number	8		

Abbildung 1: Anzeige der Variablen-Eigenschaften im Properties-Fenster des SAS Explorers

Kontrollieren kann man die Eintragung im Properties-Fenster des SAS-Explorers. (Dazu im SAS-Explorer die Bibliothek Work öffnen, mit der rechten Maustaste auf das Symbol der Tabelle klasse klicken und View Columns auswählen.)

3.2 Klassenbildung mit Formaten

Die Formate können wir aber nicht nur für Listenberichte einsetzen, sondern auch zur Bildung von Klassen. Für die Vorbereitung des KSFE-Tagungsbüros sollten die Buchstaben festgelegt werden, nach denen die drei Schalter bestimmt werden, möglichst gleichmäßig. Es sollte also eine Klassierung der angemeldeten Konferenzteilnehmer nach dem Anfangsbuchstaben deren Nachnamen vorgenommen werden. Auszählen, auch kumulativ, erledigt die Prozedur FREQ, klassieren können wir mit dem passenden Format, und alles in einem Prozedurschritt:

```
Proc Freq Data=work.klasse;  
  Tables name;  
  Format name $1.;  
Run;
```

Im Ausgabefenster erhalten wir:

Name	Frequency	Percent	Cumulative Frequency	Cumulative Percent
B	2	33.33	2	33.33
G	1	16.67	3	50.00
M	1	16.67	4	66.67
S	1	16.67	5	83.33
W	1	16.67	6	100.00

Weitere Anwendungen können im deutschsprachigen SAS-Wiki nachgelesen werden unter de.saswiki.org, Suche nach Format.

Die bislang verwendeten Formate sind alles SAS-eigene Formate. Für die Kodierung der Variable Geschlecht (1 für weiblich, 2 für männlich) liegt leider kein passendes Format vor. Dieses muss erst mit der Prozedur FORMAT definiert werden.

4 Die Prozedur Format

Die Prozedur Format bietet dem Anwender die Möglichkeit, neue/eigene Formate zu definieren, beispielsweise für die Variablen Geschlecht und Note. Im Prozedur FORMAT-Schritt setzt man dazu die Anweisung VALUE ein. Mit der Anweisung PICTURE werden ebenfalls Formate definiert, allerdings da, wo es darum geht, Zahlen in bestimmter Form, wie z.B. Telefonnummern oder Längen mit Maßeinheit auszugeben.

Die allgemeine Syntax der Prozedur FORMAT lautet:

```
Proc Format <Optionen>;  
  Value ...;  
  Picture ...;  
Run;
```

4.1 Die Anweisung VALUE

Für die SAS Tabelle klasse werden wir jetzt zwei Formate definieren: gender, um die Werte 1 und 2 der Variable Geschlecht zu erklären, und note, um die Noten im Klartext auszugeben.

Das Format gender weist jedem Wert einen neuen Wert zu: 1 wird zu weiblich, 2 zu männlich. Bei dem Format note dagegen wird einem Wertebereich ein neuer Wert zugewiesen, beispielsweise Werte zwischen 0.7 und 1.3 werden zu „sehr gut“.

Dem Schlüsselwort VALUE folgt zunächst der Formatname, bei Textformaten muss dieser mit \$ beginnen. Anschließend wird der alte Wert bzw. der alte Bereich angegeben und gefolgt von einem Gleichheitszeichen der neue Wert.

Unser Beispiel lautet dann:

```
Proc Format;
  Value gender 1 = "weiblich"
             2 = "männlich";
  Value note   Low-<1.5='sehr gut'
             1.5-<2.5='gut'
             2.5-<3.5='befriedigend'
             3.5-<4.5='ausreichend'
             4.5-<5.5='mangelhaft'
             5.7-High='ungenügend';
```

```
Run;
```

LOW und HIGH in der Definition des Formats Note bezeichnen den kleinsten bzw. größtmöglichen Wert.

Wir nutzen die neu definierten Formate nun in einem Listenbericht:

```
Proc Print Data=klasse;
  Var name vorname geschlecht note;
  Format geschlecht gender. note note.;
Run;
```

Und erhalten folgende Darstellung:

Obs	Name	Vorname	Geschlecht	Note
1	Bauer	Natalie	weiblich	gut
2	Graf	Jochen	männlich	sehr gut
3	Metzger	Jürgen	männlich	befriedigend
4	Schäfer	Inga	weiblich	befriedigend
5	Walter	Maike	weiblich	gut
6	Bletzer	Jörg	männlich	gut

4.2 Die Anweisung PICTURE

Während man mit der Anweisung VALUE Wertebereichen oder Einzelwerten neue Werte zuweist, bietet die Anweisung PICTURE die Möglichkeit, die Form von Zahlenwerten festzulegen. So kann beispielsweise eine Größeneinheit wie „m“ (für Meter) an die Körpergröße angefügt werden, das \$- oder €-Zeichen kann vor einem Geldbetrag angezeigt werden, führende Nullen lassen sich ausgegeben, z.B. bei numerisch kodierten Postleitzahlen, oder Telefonnummern lesbar formatieren (siehe auch [5]).

Wir werden für unsere Beispieldatei die Variable `groesse` mit einem entsprechenden Format belegen. Dazu formulieren wir folgende Picture-Anweisung in einem PROC FORMAT-Schritt:

```
Proc Format;  
    Picture groesse Low-High='009.99 m';  
Run;
```

Der Formatname lautet `groesse`. Die Definition gilt für den gesamten Wertebereich, also: „Low-High“. Nach dem Gleichheitszeichen erfolgt in Anführungszeichen eine Kombination von Nullen und Neunen. Diese geben die Stellen an, die ausgegeben werden. „0“ wird nur ausgegeben, wenn vorhanden, „9“ wird immer ausgegeben, d.h. die Werte werden immer mit zwei Nachkommastellen dargestellt. Das Leerzeichen und „m“ geben die Maßeinheit an, hier: Meter.

In Form eines Listenberichts

```
Proc Print Data=klasse;  
    Var name vorname groesse;  
    Format groesse groesse.;  
Run;
```

zeigt sich das dann wie folgt:

Obs	Name	Vorname	Groesse
1	Bauer	Natalie	1.56 m
2	Graf	Jochen	1.67 m
3	Metzger	Jürgen	1.80 m
4	Schäfer	Inga	1.71 m
5	Walter	Maike	1.66 m
6	Bletzer	Jörg	1.87 m

Der durch das Picture-Format definierte Wert kann eine Länge von bis zu 40 Zeichen annehmen.

Die Option `PREFIX=` (die Option muss hierbei in Klammern gesetzt werden) gestattet die Einfügung von Zeichen vor dem eigentlichen Wert. Auch hierzu ein Beispiel:

```
Proc Format;  
    Picture groesse Low-High='009.99' (Prefix="(in Metern) ");  
Run;
```


Liefert im Listenbericht (gleicher Prozedurschritt wie im vorherigen Beispiel):

Obs	Name	Vorname	Groesse
1	Bauer	Natalie	(in Metern) 1.56
2	Graf	Jochen	(in Metern) 1.67
3	Metzger	Jürgen	(in Metern) 1.80
4	Schäfer	Inga	(in Metern) 1.71
5	Walter	Maike	(in Metern) 1.66
6	Bletzer	Jörg	(in Metern) 1.87

Weitere hilfreiche Optionen der PICTURE Anweisung sind ROUND und MULT=. ROUND rundet den Wert auf den nächsten ganzzahligen Wert (ab 0,5 wird aufgerundet, ansonsten abgerundet), MULT=n oder MULTIPLIER=n bezeichnet den Faktor, mit dem der Wert multipliziert wird. Also n=3 bedeutet, dass der Wert mit 10³ multipliziert wird. ROUND und MULT= werden vor dem Formatieren angewandt, d.h. erst wird gerundet bzw. multipliziert, dann die mittels Picture definierte Form angewendet. Für Beispiele siehe [8].

4.3 Geschachtelte Formate

Bei der Definition neuer Formate können im Label-Teil der Anweisung VALUE anstelle einzelner Werte mit Hilfe von eckigen Klammern bereits vorhandene Formate benannt werden.

Im folgenden Beispiel wird das Format gebgruppen definiert.

```
Proc Format;
  Value gebgruppen
    . = "k.A."
    Low-"31dec1990"d = [Year4.]
    "01jan1991"d-High = [Monyy7.];
Run;
```

Fehlende Werte werden damit als „k.A.“ (für keine Angabe) dargestellt, Datumsangaben bis zum 31. Dezember 1990 mit der Jahreszahl (ausgedrückt mit Hilfe des Formats Year4.), und Datumsangaben ab dem 1. Januar 1991 mit Monat und Jahresangabe (Format Monyy7.). Für unsere Beispieldatei sieht der Listenbericht mit

```
Proc Print Data=klasse;
  Format gebdatum gebgruppen.;
  Var vorname name gebdatum;
Run;
```

dann wie folgt aus:

Obs	Vorname	Name	Gebdatum
1	Natalie	Bauer	1960
2	Jochen	Graf	SEP1996
3	Jürgen	Metzger	FEB1995
4	Inga	Schäfer	JUN1996
5	Maike	Walter	NOV1995
6	Jörg	Bletzer	MAR1995

5 Weitere Optionen der Prozedur Format

In den folgenden drei Abschnitten werden noch weitere Optionen der Anweisung PROC FORMAT vorgestellt, die das Arbeiten mit Formaten erleichtern: LIB= zum permanenten Abspeichern von eigenen Formaten und CNTLIN= und CNTLOUT=, um Formate in Tabellen abzulegen und wieder einzulesen, eine bequeme Möglichkeit, um Metadaten bzw. sich dynamisch verändernde Datenwerte in Formate zu verwandeln.

5.1 Formate permanent abspeichern mit LIB=

Die mit unseren bisherigen Beispielprogrammen definierten Formate werden im Katalog Formats in der Bibliothek Work abgelegt. Damit sind diese Formate nur temporär verfügbar, da mit Beendigung der SAS-Sitzung der gesamte Work-Bereich gelöscht wird und damit auch die Formate. Möchte man diese in der nächsten Sitzung wieder verwenden, muss der Proc Format-Schritt erneut ausgeführt werden.

Problematisch kann dies werden, wenn die Formate permanent mit einer SAS-Tabelle abgespeichert werden und dann beim Öffnen nicht vorhanden sind. Ein Doppelklick aus dem SAS Explorer heraus endet dann mit einer wüsten Fehlermeldung, Prozedurschritte werden nicht ausgeführt.

Abhilfe kann dann eines der folgenden Verfahren schaffen:

1. Alle Formate aus der Tabelle entfernen mit der Anweisung

```
Format _All_ ;
```

Nachteil: Damit verliert man auch die Zuweisungen SAS-eigener Formate, z.B. bei Datumsvariablen.

2. Mit der Systemeinstellung NOFMterr die Ausgabe der Fehlermeldungen und den Abbruch des Daten- oder Prozedurschritts bei fehlenden Formaten verhindern. Diese Systemeinstellung kann mit der Option OPTIONS innerhalb der SAS Sitzung aktiviert werden:

```
Options Nofmterr;
```

Damit bleiben die SAS-eigenen Formate, z.B. für Datumsvariablen, erhalten und der mit dem ersten Vorschlag benannte Nachteil tritt nicht ein.

3. Den Prozedur Format-Schritt erneut ausführen, wenn dieser zur Verfügung steht. (Bei Tabellen von Dritten ist dies oft nicht der Fall.)
4. Die Formate permanent abspeichern und beim Weitergeben diesen Katalog mitgeben, was im Folgenden genauer erläutert wird.

Die permanente Speicherung erfolgt durch Angabe der Option LIB= in der PROC FORMAT-Anweisung. Dabei kann auch der Katalog spezifiziert werden:

```
Proc Format Lib=bibliothek;
oder
Proc Format Lib=bibliothek.katalog;
```

Im Beispiel:

```
Libname ksfeFmt "c:/temp";

Proc Format Lib=ksfeFmt;
  Value noten
    1 = "exzellent"
    2 = "annehmbar"
    other = "inakzeptabel"
;
Run;
```

Der Datenschnitt

```
Data _null_;
  x = 1;
  Put x= x=noten.;
Run;
```

produziert aber lediglich eine Fehlermeldung, denn SAS kann das neue Format noten (noch) nicht finden.

```
77  Data _null_;
78      x = 1;
79      Put x= x=noten.;
          -----
          48
ERROR 48-59: The format NOTEN was not found or could not be loaded.

80  Run;
```

NOTE: The SAS System stopped processing this step because of errors.

Die Frage, die sich jetzt sofort stellt, ist: Wo sucht SAS nach Formaten bzw. wie kann ich dem System mitteilen, wo meine Formate stehen?

Standardmäßig durchsucht das SAS System folgende Orte:

1. Work.Formats
2. Library.Formats
3. SAS-eigene Formate

Um Punkt 2 zu testen, legen wir eine weitere Bibliothek namens LIBRARY an:

```
Libname library (KsfeFmt);
```

Wenn wir jetzt den gleichen Datenschnitt wiederholen, wird das Format `noten` gefunden.

```
...  
85 Run;  
  
x=1 x=exzellent  
NOTE: DATA statement used (Total process time):
```

Alternativ können wir aber auch den Suchpfad, der standardmäßig nur Work und Library beinhaltet, um unsere Bibliothek mit der Systemoption `FMTSEARCH=` und der globalen Anweisung `Options` erweitern:

```
Options Fmtsearch=(Work Library ksfeFmt);
```

5.2 Formate in Tabellen schreiben mit `CNTLOUT=`

Die Option `CNTLOUT=` erlaubt es dem Anwender, Formate aus dem Formatekatalog heraus in eine SAS-Tabelle zu übertragen. `CNTLOUT=` benennt dabei die Ausgabetable, in der Online-Hilfe als Kontrolltabelle (Output Control Data Set) bezeichnet. Die Anweisung `Select` gestattet die Auswahl der zu übertragenden Formate.

```
Proc Format Lib=ksfeFmt Cntlout = Work.test;  
  Select noten;  
Run;
```

Die Tabelle `work.test` erhält als Listenbericht folgende Form:

```
Proc Print Data = Work.test;  
Run;
```

Obs	FMTNAME	START	END	LABEL	MIN	MAX	DEFAULT	LENGTH	FUZZ
1	NOTEN	1	1	exzellent	1	40	12	12	1E-12
2	NOTEN	2	2	annehmbar	1	40	12	12	1E-12
3	NOTEN	**OTHER**	**OTHER**	inakzeptabel	1	40	12	12	1E-12

Obs	PREFIX	MULT	FILL	NOEDIT	TYPE	SEXCL	EEXCL	HLO	DECSEP	DIG3SEP
1		0		0	N	N	N			

DATATYPE LANGUAGE

2	0	0	N	N	N	
3	0	0	N	N	N	0

Wichtig sind dabei die ersten vier Variablen, Fmtname, Start, End und Label.

5.3 Formate aus Tabellen lesen mit CNTLIN=

Die gerade eben erzeugte Kontrolltabelle könnte jetzt wieder eingelesen werden mit der Option CNTLIN=. Für das folgende Beispiel erzeugen wir aber aus einer anderen, bekannten, SAS-Tabelle, nämlich sashelp.class, eine Kontrolltabelle, die anschließend mit der Prozedur Format eingelesen wird:

```
Data work.kontrollinput;
  Set sashelp.class;
  fmtname="name2sex";
  type="C";
  start=name;
  end=name;
  label=sex;
  keep fmtname--label;
run;
```

Oder alternativ für SQL-Kenner:

```
Proc Sql;
  Create View Work.Tmp As
  Select "name2sex" As fmtName, "C" As type,
         name As start, name As end, sex As label
  From Sashelp.Class;
Quit;
```

Die für den anschließenden Prozedur Format-Schritt benötigten Variablen sind dabei:

- FMTNAME Name des Formats
- TYPE Typ des Formats [C,N]
- START Anfang des Wertebereiches
- END Ende des Wertebereiches
- LABEL Anzeigetext für diesen Bereich

Diese Kontrolldatei wird nun eingelesen

```
Proc Format Cntlin = work.kontrollinput;
Run;
```

und in einem Datenschnitt angewendet:

```
Data _Null_;
  Set sashelp.Class(keep = name);
  Put name = @20 name $name2sex.;
Run;
```

Das Ergebnis – im Log-Fenster – sieht wie folgt aus:

```
102 Data _Null_;  
103     Set sashelp.Class(keep = name);  
104     Put name = @20 name $name2sex.;  
105 Run;
```

```
Name=Alfred           M  
Name=Alice            F  
Name=Barbara          F  
Name=Carol            F  
Name=Henry            M  
Name=James            M  
Name=Jane             F  
Name=Janet            F  
Name=Jeffrey          M  
...
```

Literatur

- [1] Cody, Ron (2004): SAS Functions by Examples, SAS Institute Inc., Cary, North Carolina
- [2] Cody, Ron (2007): Learning SAS by Example – A Programmer’s Guide, SAS Institute Inc., Cary, North Carolina
- [3] Delwiche, Lora D.; Slaughter Susan J. (2008): The Little SAS Book: A Primer, 4. Edition, SAS Institute Inc., Cary, North Carolina
- [4] Deutschsprachiges SAS-Wiki: de.saswiki.org
Suche: Formate automatisieren (Tipps & Tricks-Beitrag der KSFE 2008)
- [5] Karp, Andrew H.: Getting in to the Picture (Format). SUGI 31, Paper 243-31
<http://www2.sas.com/proceedings/sugi31/243-31.pdf>
- [6] Ortseifen, Carina (1997): Der SAS-Kurs – Eine leicht verständliche Einführung, International Thomson Publishing, Bonn u.a.
- [7] SAS Education Guide: Unser Experten-Tipp im September 2010
<http://www.sas.com/offices/europe/germany/email/archiv/aktuell.html>
- [8] SAS Online Help zur Prozedur FORMAT
<http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/viewer.htm#a000063536.htm>
- [9] Shoemaker, Jack N.: PROC FORMAT in Action. SUGI 27, Paper 56-27
<http://www2.sas.com/proceedings/sugi27/p056-27.pdf>