

Konvertierung von SPSS-Format Data Files in SAS Tabellen und umgekehrt

Dr. Carina Ortseifen
 Universität Heidelberg
 Im Neuenheimer Feld 293
 69120 Heidelberg
 Carina.Ortseifen@urz.uni-heidelberg.de

Wolf F. Lesener
 Humboldt-Universität zu Berlin
 Unter den Linden 6
 10099 Berlin
 wflesener@cms.hu-berlin.de

Zusammenfassung

Bei Datei-Konvertierungen geht es um eine möglichst verlustfreie Wandlung der in einem Dateiformat gespeicherten Informationen in ein anderes, verwandtes Dateiformat. In diesem Artikel wird die wechselseitige Konvertierung von SPSS-Format Data Files¹ in SAS Tabellen² (Datasets) betrachtet.

Es werden beide Richtungen der Konvertierung an je 3 Methoden verglichen und die damit beobachteten Probleme aufgezeigt:

- A) Was bietet das SAS System auf prozeduraler Basis?
- B) Was bietet SPSS auf Kommando-Ebene?
- C) Unsere Lösung: Ein SAS Makro.

Eine vollends befriedigende Lösung beider Konvertierungsrichtungen müsste nach der Konvertierung einer konvertierten Datei die ursprüngliche Quelldatei reproduzieren. Diese Forderung kann eine Dateikonvertierung zwischen SAS und SPSS nicht erfüllen, da es eine ganze Reihe inkompatibler Dateibestandteile bzw. unterschiedlicher Restriktionen gibt. Bleibt man im gemeinsamen Durchschnitt, so soll die Abweichung so minimal als möglich sein.

Schlüsselwörter: SAS Tabelle, SPSS Tabelle, Konvertierung, PROC EXPORT, EXPORT-Prozedur, PROC IMPORT, IMPORT-Prozedur, Get SAS-Befehl, Makro, %GetSAS, %GetSPSS

1 Die "Versuchsanordnung"

Bitte kopieren Sie alle Beispiele³ z.B. nach D:\Ulm (nachfolgend als *path* bezeichnet). Richten Sie anschließend zwei Desktop Icons (durch Kopieren) mit folgenden Eigenschaften ein:

beide Icons:	Ausführen in:	<i>path</i>
nur SAS ⁴	Ziel:	-AUTOEXEC " <i>path/Autoexec.sas</i> " -SASINITIALFOLDER " <i>path</i> "

¹ SPSS-Format Data Files werden im weiteren Artikel verkürzt als SPSS Dateien bezeichnet.

² Im Weiteren auch als SAS Datei bezeichnet.

³ Beispielpprogramme und Testdateien befinden sich auf der CD im Proceedingsband.

⁴ bisherigen Eintrag fortschreiben; vor jedem Minuszeichen ist ein signifikantes Leerzeichen erforderlich

Nur zur Erinnerung verweisen im Text angebrachte Icons auf die aktuelle Arbeitsumgebung:

SAS 9.1.3 SP4⁵: 

SPSS 14.0: 

Bevor auf die Makros `%GetSAS` und `%GetSPSS` eingegangen wird, werden die jeweiligen nativen Tools der Programmsysteme SPSS und SAS für die gegenseitige Dateikonvertierung in Beispielprogrammen aufgezeigt. Deren Resultate und eine Aufstellung beobachteter Probleme sind jeweils im Anhang aufgelistet.

Im Text sind die Namen der Beispielprogramme kursiv wiedergegeben.

Die Reihenfolge der Programme und den damit verbundenen Datenfluss zeigt Abbildung 1:

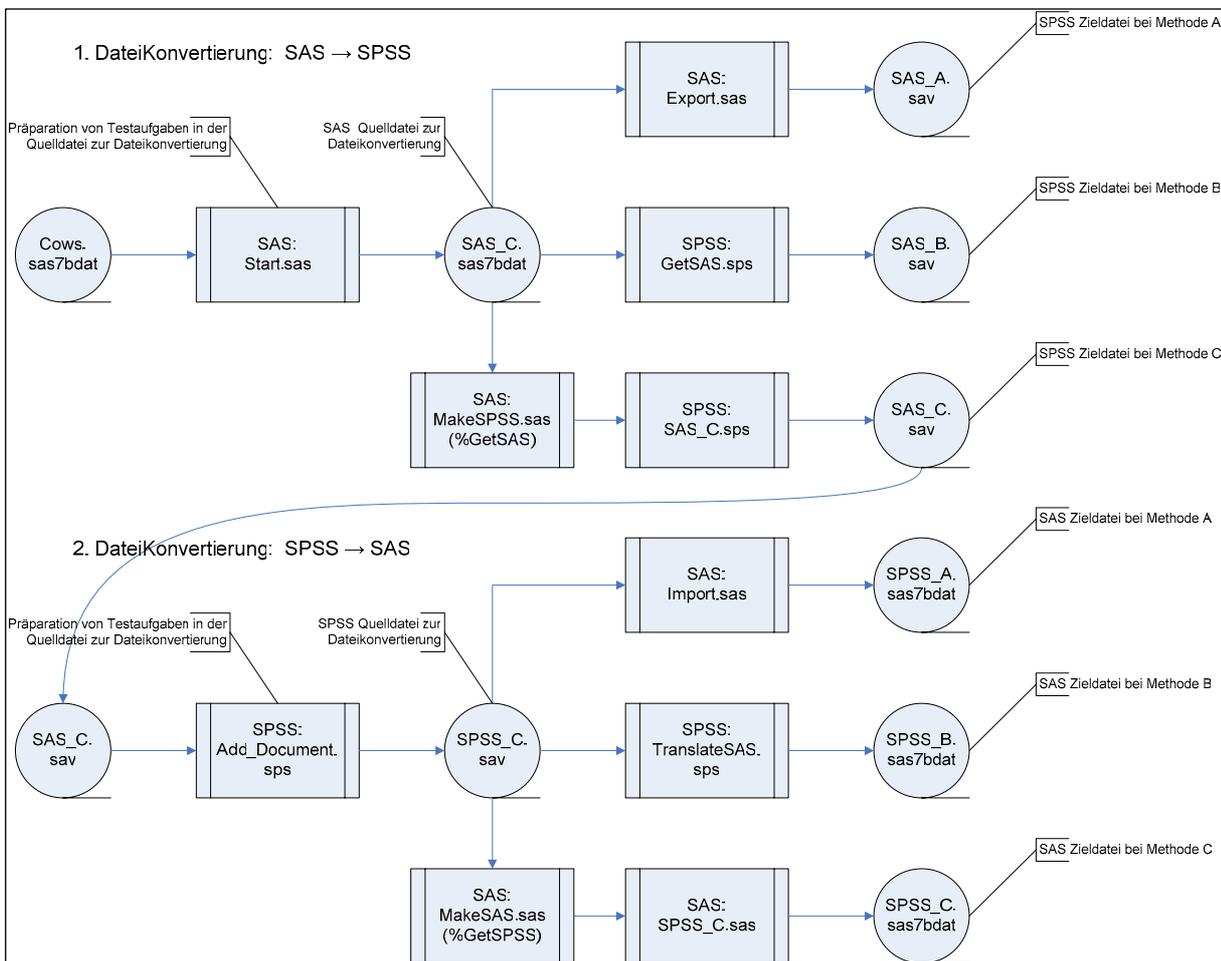


Abbildung 1: Betrachtete Methoden der Dateikonvertierung zwischen SAS und SPSS und die zugehörigen Beispielprogramme⁶ und beteiligten Dateien⁷

⁵ Download verfügbar unter http://ftp.sas.com/techsup/download/hotfix/e9_sbc_prod_list.html#e9h004

⁶ Oberhalb des Programmnamens wird jeweils die Arbeitsumgebung angegeben

⁷ Alle Dateien sollten trotz unterschiedlichen Formats weitestgehend dieselben Informationen enthalten.

Ausgangspunkt aller Konvertierungen ist die Testdatei *Cows.sas7bdat*⁸. Größtmögliche Übereinstimmung sollten letztlich die Dateien *SAS_C.sas7bdat* und die Datei *SPSS_C.sas7bdat* als rekonvertierte *SAS_C.sas7bdat* aufweisen.

Jede Richtung der Dateikonvertierung wird in den Beispielen durch ein vorgeschaltetes Programm eingeleitet, das im Wesentlichen dafür sorgt, dass die jeweilige Quelldatei Systemspezifika enthält, die auf ihre Beachtung und Umsetzung in die Zieldatei getestet wurden.

Der Schwerpunkt der Konvertierungen der proprietären Dateiformate per Makro liegt auf *SPSS nach SAS* in der Arbeitsumgebung SAS. Diese Konstellation ist besonders dann gefragt, wenn z.B. gar kein SPSS installiert ist. Das Makro für die Umkehroperation wurde mehr aus "Symmetriegründen" entwickelt und ist aus Sicht von SAS Anwendern von untergeordneter Bedeutung.

Beide Konvertierungen per Makro von SPSS nach SAS und umgekehrt, sind je als zweiteiliger Prozess realisiert:

1. Generierung eines bzgl. der Quelldatei sehr spezifischen Konverterprogramms jeweils durch Aufruf eines SAS Makros.
2. Ausführung des generierten Konverterprogramms, das die Zieldatei(en) speichert. Dies geschieht hier stets in der dem Ziel entsprechenden Arbeitsumgebung.

Die Zeitspanne zwischen Schritt 1 und Schritt 2 ist beliebig. Aber natürlich darf in der Zwischenzeit die Quelldatei, auf der das Konverterprogramm aufsetzt, nicht verändert werden.

2 Konvertierung: SAS Datei => SPSS Datei⁹

 Bitte vergessen Sie nicht, zunächst das Programm *Start.sas* abzuarbeiten, falls Sie die Beispiele auf Ihrem PC nachvollziehen wollen¹⁰.

2.1 PROC EXPORT¹¹

```

OPTIONS FMTSEARCH=(library.formats); /* 1 Export.sas */
PROC EXPORT DATA =library.SAS_C /* 2 */
          OUTFILE="SPSS_A.sav" /* 3 */
          DBMS =sav /* 4 */
          REPLACE; /* 5 */
RUN;

```

1. Die Bibliothek *library* enthält die zu konvertierende SAS Tabelle und einen SAS Katalog und wurde für alle Beispiele bereits mit Ausführung von *Autoexec.sas* verein-

⁸ Eine ausführliche Erläuterung enthält Anhang A.

⁹ Es gibt weitere Software als die hier betrachtete, die eine entsprechende Dateikonvertierung zwischen SAS und SPSS realisiert (vgl. [1])

¹⁰ Die z.T. notwendige Programmsequenz ist in Abbildung 1 zu sehen.

¹¹ Proc EXPORT mit Ziel SPSS wird nicht in der Online Dokumentation beschrieben (vgl. [2])

bart.

Um nutzerdefinierte Formate aufzufinden, müssen die Kataloge, die die Formatbeschreibungen enthalten, mit der Systemoption FMTSEARCH deklariert werden. Wurde ein Formatkatalog mit dem Defaultnamen *formats* angelegt (wie hier), wäre es hinreichend, allein die Bibliotheksreferenz in der Systemoption FMTSEARCH zu spezifizieren. Selbst dies kann entfallen, wenn der Bibliotheksname (hier *library*) bereits in der Voreinstellung von FMTSEARCH aufgelistet ist.

2. Die Konvertierung einer SAS Datei in eine SPSS Datei geschieht mit Hilfe von PROC EXPORT. Wie bei allen SAS Prozeduren üblich, benennt die DATA Option die Eingabedatei der Prozedur. Hier ist es die SAS Quelldatei, die in die SPSS Zieldatei zu konvertieren ist.
3. Mit der OUTFILE Option werden Name und Speicherort der Zieldatei definiert.
4. Der Wert, der der DBMS Option zugewiesen wird, bestimmt das Format, in dem die Zieldatei zu speichern ist. Im Beispiel steht *sav* für Konvertierung nach SPSS.
5. Die Option REPLACE erlaubt ein Überschreiben einer evtl. vorhandenen gleichnamigen Zieldatei und damit eine fehlerfreie Wiederholbarkeit des Prozedurschritts.

Das Resultat von *Export.sas* ist angesichts der Tatsache, dass keine VALUE LABEL aus den nutzerdefinierten SAS Formaten abgeleitet wurden, ggf. mit hohem Nachbesserungsaufwand verbunden. Etwas seltsam erscheint deshalb eine Ausschrift im LOG Fenster, die die SPSS Datei trotz Fehlermeldung als *successfully created* beschreibt. Eine Liste der beobachteten Probleme findet sich im Anhang B.

2.2 Aus dem SyntaxFenster¹²

```
GET                                /* 1   GetSAS.sps */
  SAS DATA='SAS_C.sas7bdat'      /* 2   */
  /FORMATS='Formats.sas7bcat'.    /* 3   */
```

1. Kommando für Dateneingabe aus einer Datei ...
2. vom Format SAS Tabelle¹³, mit Namen wie bei DATA angegeben und ...
3. ValueLabel sollen bitte aus den Formatbeschreibungen des benannten SAS Katalogs abgeleitet werden.

Das Resultat von *GetSAS.sps* kann man im SPSS DataView betrachten. Es ist besser als das von PROC EXPORT aus SAS, aber auch hier fehlen die VALUE LABEL, die

¹² Bei interaktiver Steuerung via Menü: File => Open => Data ... kann man zwar eine SAS Datei über Dateityp *SAS Long File Name (*.sas7bdat)* anwählen, aber man hat keine Möglichkeit, einen Formatkatalog zuzuordnen.

¹³ Erst ab SPSS 14.0 verarbeitet das GET SAS Kommando von SPSS auch komprimierte SAS Dateien korrekt. In SPSS 13.0 erhält man als Fehleranzeige: Error # 7202. Command name: GET SAS - SAS dictionary read error.

Stringvariablen mit Länge<9 auch im SPSS zugewiesen werden könnten. Eine Liste der beobachteten Probleme findet sich im Anhang C.

2.3 Konvertierung mit Makro %GetSAS

Die Reihenfolge der Arbeitsschritte und den damit verbundenen Datenfluss zeigt die folgende Abbildung:

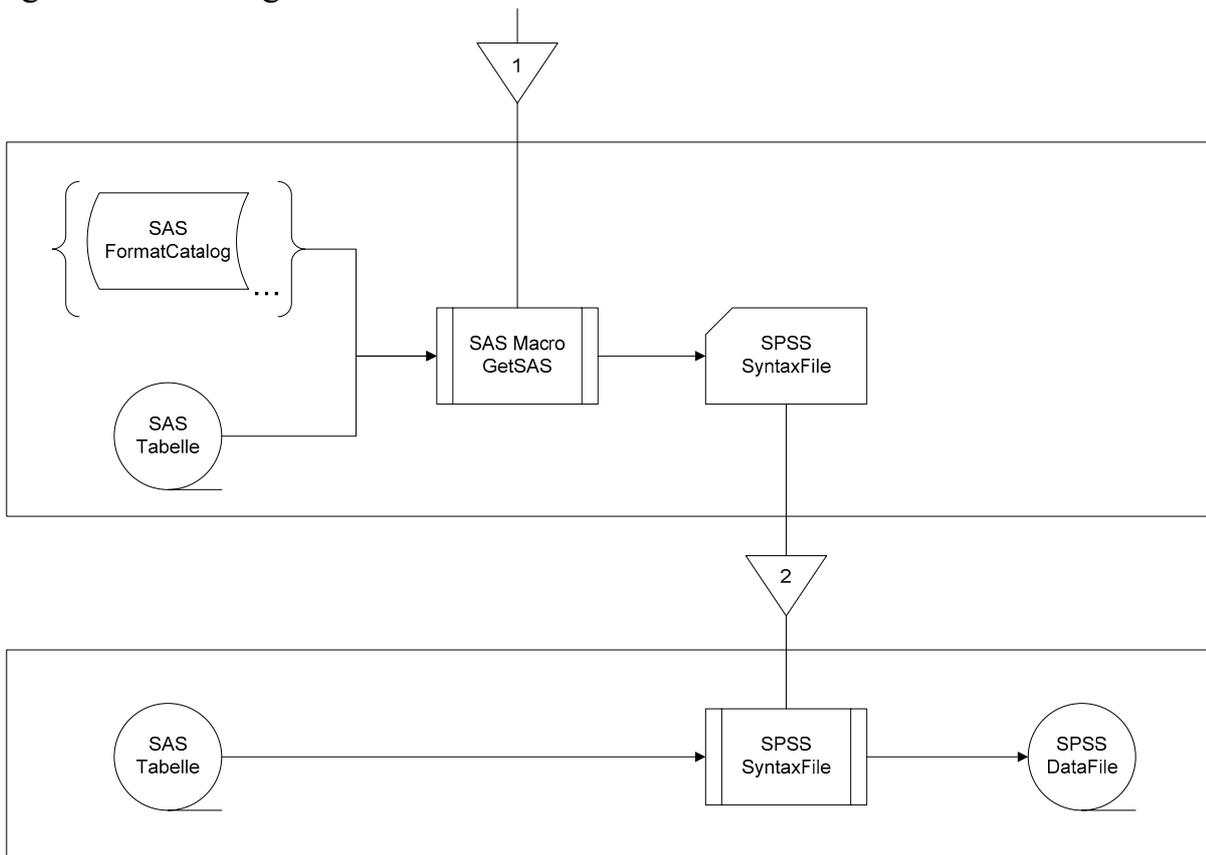


Abbildung 2: Datenfluss und Steuerung bei Konvertierung von SAS nach SPSS mit Makro %GetSAS

Schritt 1: Ausführung des SAS Makros %GetSAS zur Generierung eines spezifischen Konverterprogramms, das aus der zu konvertierenden SAS Datei abgeleitet und als SPSS Syntaxfile unter demselben Namen und Pfad wie die SAS Datei gespeichert wird. Namentlich ändert sich nur die Dateinamen-Extension von *sas7bdat* bei der SAS Datei zu *sps* beim SPSS Syntaxfile.

Einziger Parameter des Makros %GetSAS ist ein ein- oder zweistufiger SAS Dateiname. Referenziert die SAS Datei nutzerdefinierte Formate, so werden diese in den Quellen, die mit der Systemoption FMTSEARCH=(...) vereinbart wurden, gesucht und zur Generierung von Value Labels im SPSS Syntaxfiles herangezogen.

Schritt 2: Ausführung des generierten SPSS Syntaxfiles, das das spezifische Konverterprogramm enthält und die SAS Datei in eine SPSS Datei unter demselben Namen und

Pfad wie die SAS Datei konvertiert. Namentlich ändert sich nur die Extension des Dateinamens von *sas7bdat* bei der SAS Datei zu *sav* bei der SPSS Datei.

Das SPSS Syntaxfile ist in der SPSS Kommandosprache notiert und speichert die SPSS Datei inkl. des File Label, aller Variable Label und aller Value Label (und referenziert hier deshalb auch keinen SAS Formatkatalog mehr).

Eine Abkürzung (truncation) der ggf. längeren SAS Pendants bei den Label wird von SPSS automatisch vorgenommen.

Der Aufruf von %GetSAS

```
OPTIONS FMTSEARCH=(work library ...); /* 1 MakeSPSS.sas */
%GetSAS (library.SAS_C); /* 2 */
```

1. Die Bibliothek mit der Referenz *library* enthält im Beispiel den Katalog *formats* mit den nutzerdefinierten Formaten, die in der SAS Tabelle referenziert werden. Es ist nur nötig der Option FMTSEARCH Werte zuzuweisen, wenn überhaupt nutzerdefinierte Formate in der SAS Datei verwendet werden und deren Beschreibungen nicht in den voreingestellten Quellen zu finden sind. Es sind natürlich auch mehrere Bibliotheksreferenzen bzw. zweistufige Namen von SAS Katalogen als Quellen nutzerdefinierter Format-Beschreibungen zulässig.
2. Für die Dateikonvertierung von SAS nach SPSS ist das Makro %GetSAS aufzurufen. Einziger Parameter ist ein SAS Dateiname einer SAS Tabelle hier zweistufig: *library.SAS_C*.
Einstufige Dateinamen sind ebenso zulässig, referenzieren jedoch temporäre SAS Dateien in der Arbeitsbibliothek *work*, was wegen des gleichen Speicherortes für das dann ebenfalls nur temporär verfügbare Konverter-Programm von Bedeutung wäre. Aus dem Namen der SAS Datei leitet sich der Name des Konverter-Programms ab (im Beispiel: *SAS_C.sps*) Nur die Extension verweist auf ein SPSS Syntaxfile, das man als ASCII Textfile mit jedem Texteditor visualisieren könnte.

Mögliche Fehleranzeigen des Makros %GetSAS, deren Ursachen und Behebung:

Ursache	Behebung
File <i>SAS Datei.DATA</i> does not exist.	evtl. Schreibfehler im Dateinamen
Keine referenzierten nutzerdefinierten Formate auffindbar	OPTIONS FMTSEARCH=(); ergänzen

Tabelle 1: Mögliche Fehleranzeigen von %GetSAS im LOG Fenster

Das generierte Konverterprogramm *SAS_C.sps*:

```

GET SAS DATA="path\SAS_C.sas7bdat".      /* 1 SAS_C.sps */
FILE LABEL  "Leistungsdaten von Milchkühen" . /* 2 */
VALUE LABELS  Betrieb                      /* 3 */
              'K' 'Kaltenhausen'
              'S' 'Seegrehna'.

VALUE LABELS  Genotyp
              33 'Frisien'
              34 'Schwarz-Bunt'
              35 'Jersey'
              36 'Galloway'.

SAVE OUTFILE="path\SAS_C.sav".          /* 4 */

```

1. SPSS liest und konvertiert mit dem GET SAS Kommando direkt SAS Dateien. *path* repräsentiert den Speicherort (Verzeichnis).
2. Ein evtl. vorhandenes File Label der SAS Datei wird nicht automatisch durch das GET SAS Kommando übertragen, sondern ist erneut zur SPSS Datei hinzuzusetzen.
3. Aus den nutzerdefinierten Formaten, die in den SAS Dateien referenziert werden, sind VALUE LABELS Kommandos generiert worden, die wiederum den zugehörigen Variablen zugeordnet werden. Value Label sind grundsätzlich optional, aber nur für kategorielle Variablen sinnvoll. Verwendet man sie, sind sie stets paarweise als gespeicherter und gedruckter Wert anzugeben. Ist die kategorielle Variable eine String-Variable, können gespeicherten Werten nur bis zu einer Länge von 8 Zeichen Value Label zugeordnet werden.
Wenn das Kommando VALUE LABELS auskommentiert wurde, ist der Variablenname aus einem SAS Namens-Literal abgeleitet worden, das sich nicht direkt als SPSS Variablenname eignet. Hier muss der Anwender ggf. händisch editieren.
4. Die Ausführung des Programms im SPSS Syntaxfenster führt zur Bereitstellung der Daten im SPSS Dateneditor und der Speicherung der Zielfeile (SPSS Datei) im selben Verzeichnis wie die Quelldatei (SAS Datei).

Die Zielfeile *SAS_C.sav*

	Datum	Tag	Meßzeit	ME_Zeit	Betrieb
	①	②			
1	28-APR-1984	28-APR-1984	12:15:00	13:15:00	Kaltenhausen
2	26-AUG-1984	26-AUG-1984	12:15:00	13:15:00	Kaltenhausen
3	28-DEC-1984	28-DEC-1984	12:15:00	13:15:00	Kaltenhausen

	Genotyp	Tier	Jahr	Milch	Fett	Fett_A	VaterTier
						③	
1	Frisien	82451	1984	2753	118,00	,043	.
2	Frisien	83251	1984	2909	114,30	,039	.
3	Frisien	82319	1984	3036	112,00	,037	.

Der Auszug wurde dem SPSS Dateneditor¹⁴ im Data View in der Wertelabel-Ansicht entnommen und ist auf die ersten 3 Fälle (Beobachtungen) beschränkt.

Anmerkungen

Die Behandlung der nachfolgenden Probleme durch das Makro *%GetSAS* ist noch ausstehend:

1. Das SAS Format der Variablen *Datum* DDMMYYPw. wird vom GET SAS Kommando des SPSS nicht korrekt in das SPSS Format EDATE (europäisches Datum) überführt, sondern in das internationale Datumsformat DATEw.
2. Die Variable *Tag*, die im SAS als Wochentag (DOWNAME2.) formatiert ist, bleibt auch im SPSS ein Datumswert, dem das Format DATE11. für "Internationales Datum" zugewiesen wird.
Die Ableitung des Wochentages und Zuweisung des Formates WKDAY2 käme der Absicht des Anwenders sicher näher.
3. Die Variable *Fett%* wird im Zuge der Konvertierung zu Recht umbenannt in *Fett_A*, da das Sonderzeichen % in SPSS Variablenamen unzulässig ist. Aber die SAS Prozentwerte werden nicht mit 100 multipliziert und es wird in der SPSS Zielfeile auch nicht das SPSS Format PCT zugewiesen.

2.4 Die Methoden für SAS => SPSS im Vergleich

	 Proc Export	 GET SAS	 %GetSAS
komprimierte SPSS Zielfeile		+	+
Index zur Quellfeile			
Integrity Constraints			
FileCreationDate	- ¹⁵	+	+
Anzahl parallel verarbeiteter Formatkataloge	- ¹⁶	1	n
File Label			+
max. Länge der Variablenamen ¹⁷	8	32	32

¹⁴ LIST und PRINT Kommando zeigen im SPSS die gespeicherten und nicht die formatierten Werte.

¹⁵ Für das File Creation Date gilt in diesem Fall: dateintern ≠ Windows Explorer ≠ SYSFILE INFO

¹⁶ *Format ... was not found or could not be loaded*

¹⁷ Einkürzung auf angegebenen Wert; SPSS ließe ab Version 12 max. 64 Zeichen lange Variablen Namen zu

Namensliterale	(+) ¹⁸	+	+
Variablelabel	+	+	+
Valuelabel		(+) ¹⁹	(+) ¹⁹
Datumstransformation	+	+	+
Reskalierung der Wochentage			
%-Wert-Multiplikation * 100			
Datenmatrix	(+) ²⁰	+	+

Tabelle 2: Methodenvergleich der Datei-Konvertierung von SAS nach SPSS²¹

3 Konvertierung: SPSS Datei => SAS Datei

 Bitte vergessen Sie nicht, zunächst das Programm *Add Document.sps* abzuarbeiten, falls Sie die Beispiele auf Ihrem PC nachvollziehen wollen²².

3.1 PROC IMPORT

```
PROC IMPORT DATAFILE="SPSS_C.sav"          /* 1 Import.sas */
            OUT      =library.SPSS_A        /* 2 */
            DBMS     =sav                   /* 3 */
            REPLACE;                          /* 4 */
RUN;
```

1. Die Konvertierung einer SPSS Datei in eine SAS Datei geschieht mit Hilfe von PROC IMPORT. Die DATAFILE Option benennt Name und Speicherort der Quelldatei.
2. Mit der OUT Option wird der Name der Zieldatei, d.h. einer SAS Tabelle definiert. Die Bibliothek *library* wird das Resultat der Konvertierung als SAS Tabelle aufnehmen.
Existieren in der SPSS Quelldatei Valuelabel, so werden diese in SAS Formate transformiert, deren Beschreibungen im SAS Katalog *work.formats*²³ gespeichert werden.
3. Der Wert, der der DBMS Option zugewiesen wird, bestimmt das Format, in dem die Quelldatei vorliegt. Im Beispiel steht *sav* für SPSS-Format Datafile.

¹⁸ Es werden mehr Zeichen zur Bildung von SPSS Variablenamen ersetzt als nötig

¹⁹ Nur numerische nutzerdefinierte Formate werden in VALUE LABEL konvertiert

²⁰ Nicht erkannte SAS Datumsformate verhindern die Transformation des Variablenwertes in ein SPSS Datum

²¹ Berücksichtigung: leer => keine; - => fehlerhaft; (+) => ordentlich mit Einschränkung; + => ordentlich

²² Die z.T. notwendige Programmsequenz ist in Abbildung 1 zu sehen.

²³ Die mit der Datei assoziierten Formate stehen damit ohne Eingriff des Anwenders in einer neuen SAS Session nicht mehr zur Verfügung!

- Die Option REPLACE erlaubt ein Überschreiben einer evtl. vorhandenen gleichnamigen Zieldatei und damit eine fehlerfreie Wiederholbarkeit des Prozedurschritts.

Das LOG Protokoll spricht von *successfully created*. Die beobachteten Probleme bei Ausführung von *Import.sas* sind im Anhang D aufgeführt.

3.2 Aus dem Syntax-Fenster

```
GET FILE='SPSS_C.sav'.                               /* TranslateSAS.sps */
SAVE TRANSLATE                                     /* 1 */
    OUTFILE='SPSS_B.sas7bdat'                         /* 2 */
    /TYPE=SAS /VERSION=7 /PLATFORM=WINDOWS           /* 3 */
    /VALFILE='SPSS_B.sas'                             /* 4 */
    /REPLACE                                           /* 5 */
    /MAP.                                              /* 6 */
DISPLAY DOCUMENTS.                                   /* 7 */

Data written to path\SPSS_B.sas7bdat.                /* 8 */
12 variables and 1053 cases written.

Value labels written to path\SPSS_B.sas .            /* 9 */
```

- Kommando für Datenausgabe einer SPSS Datei in ein Datenformat eines anderen Programmsystems als SPSS daselbst.
- Name der Zieldatei: *SPSS_B.sas7bdat*
- Zielformat sei eine SAS Tabelle (ab Version 7), speziell für das Betriebssystem MS Windows.
- Aus den Valuelabel wird ein SAS Programm abgeleitet, das durch seine Ausführung SAS Formatbeschreibungen, die in der SAS Zieldatei referenziert werden, in den SAS Katalog *library.formats* generiert, der im selben Verzeichnis wie die SPSS Quelldatei angelegt wird.
Das SAS Programm zur Generierung der Formatbeschreibungen und die SAS Zieldatei sind gleichnamig, nur die Extensionen der Namen unterscheiden sich: Tabelle => *SPSS_B.sas7bdat*, Programm => *SPSS_B.sas*.
- Die Option REPLACE erlaubt ein Überschreiben einer evtl. vorhandenen gleichnamigen Zieldatei und damit eine fehlerfreie Wiederholbarkeit des Prozedurschritts.
- Liefert zusätzliche Informationen im SPSS Viewer über die generierten SAS Dateien, wie z.B. Anzahl der Variablen und Beobachtungen, sowie Pfade und Dateinamen.
- Der mit dem DOCUMENT Kommando der SPSS Quelldatei hinzugefügte Text, bleibt vom Kommando SAVE TRANSLATE unberücksichtigt und unerwähnt. Da-

her sollte zumindest, wie im Beispiel, allein auf Verdacht, ein DISPLAY DOCUMENTS Kommando im Programm enthalten sein.

8. Die SAS Tabelle wurde als Datei *SPSS_B.sas7bdat* aufgezeichnet.
9. Das SAS Programm zur Generierung der SAS Formatbeschreibungen wurde als Datei *SPSS_B.sas* gespeichert. Erst die Ausführung dieses Programms wird permanente Formatbeschreibungen im Katalog *library.formats* speichern.

In SPSS 14.0 hätte das Resultat von *TranslateSAS.sps* bei der Konvertierung einer SPSS Datei in eine SAS Datei mit dem SPSS Kommando SAVE TRANSLATE fast überzeugt²⁴. Eine Liste mit auszugsweisen Resultaten findet man im Anhang E.

3.3 Konvertierung mit Makro %GetSPSS

In Abbildung 3 sind Datenfluss und Steuerung bei der zweiteiligen Konvertierung eines SPSS-Format Datafiles in eine SAS Tabelle mit Hilfe des Makros %GetSPSS dargestellt. Bitte beachten Sie, dass beide Arbeitsschritte "Generieren des Konverterprogramms" und "Exekution des Konverterprogramms" in derselben Umgebung SAS ablaufen. SPSS ist hier nicht mehr erforderlich.

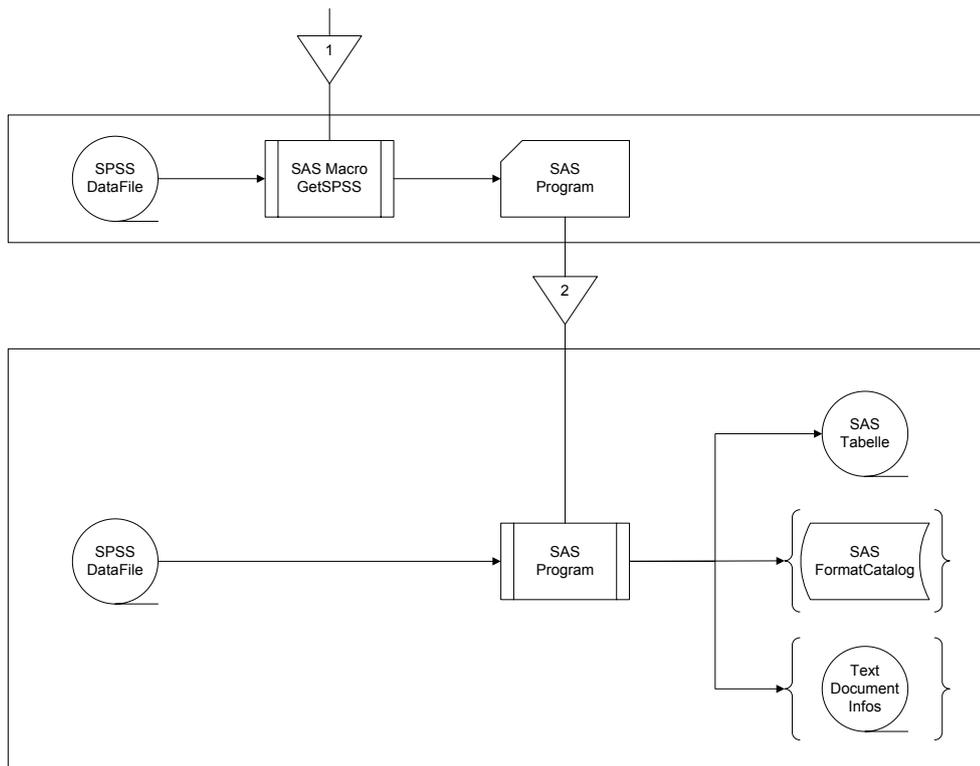


Abbildung 3: Datenfluss und Steuerung bei Konvertierung von SPSS nach SAS mit Makro %GetSPSS

²⁴ SPSS 13.0 lieferte hier noch ein syntaktisch mehrfach fehlerhaftes SAS Programm zur Generierung der nutzerdefinierten Formate.

vgl. auch SAS usage notes: <http://support.sas.com/techsup/unotes/SN/013/013209.html>

SPSS 14.0 bemerkt nicht, wenn Modifikationen an Variablenamen zu mehrfach gleichen Variablenamen in der SAS Zielfeile (was im SAS unerwünschte Nebeneffekte hat) führen.

Schritt 1: Ausführen des SAS Makros `%GetSPSS` zur Generierung eines spezifischen Konverterprogramms, das aus der zu konvertierenden SPSS Datei abgeleitet und als SAS Programm unter demselben Namen und demselben Pfad wie die SPSS Datei gespeichert wird. Namentlich ändert sich nur die Dateinamens-Extension von `sav` bei der SPSS Datei zu `sas` beim SAS Programm.

Einziger Parameter des Makros `%GetSPSS` ist der Name einer SPSS Datei. Dieser ist in der Host-Notation anzugeben und kann in den meisten Fällen einen Pfad einschließen.

Schritt 2: Ausführen des generierten SAS Programms, das das spezifische Konverterprogramm darstellt und die SPSS Quelldatei in eine SAS Zieldatei konvertiert. Wenn die SPSS Datei Vereinbarungen über die Zuweisung von Valuelabels oder spezifische SPSS Formate, wie z.B. `WKDAY` enthält, so werden diese in nutzerdefinierte SAS Formate übertragen und deren Beschreibungen in einem SAS Katalog gespeichert. Sollten der SPSS Datei Dokument-Informationen hinzugefügt worden sein, werden diese in ein ASCII Textfile abgefiltert.

Für alle drei möglichen Ausgabedateien des Konverterprogramms gilt:

- Speicherung unter demselben Namen²⁵ und Pfad wie die SPSS Quelldatei.
- Nur die Extensionen der Dateinamen variieren.
- Die SAS Tabelle erhält als Extension `sas7bdat`, der Formatkatalog `sas7bcat` und das Textfile `txt`.

Das Konverterprogramm ist in der SAS Base Language codiert. Es

- verarbeitet komprimierte und nicht komprimierte SPSS Dateien,
- erzeugt und assoziiert alle nutzerdefinierten Formate,
- transformiert die Werte aller Datumsvariablen durch Umrechnung des Bezugspunkts vom 15.10.1582 im SPSS auf den 1.1.1960 im SAS,
- reskaliert die Datumswerte von Wochentagen so, dass Variablen, denen in der SPSS Quelldatei das Format `WKDAY` zugeordnet ist, im SAS denselben Tag wiedergeben,
- dividiert explizit ausgewiesene Prozentwerte (im SPSS: `PCT` Format) durch 100 und weist als Format `PERCENT6.2` im SAS zu,
- speichert die SAS Tabelle inkl. des Filelabel und aller Variablenlabel.



Der Aufruf von `%GetSPSS`

```
%GetSPSS (path\SPSS_C.sav); /* 1 MakeSAS.sas */
```

²⁵ Nur für die Namen der SAS Tabelle und des SAS Formatkatalogs, die intern im SAS System einer definierten Namensbildung und -kontrolle unterliegen gilt:

Sie dürfen nicht länger als 32 Zeichen sein, keine Sonder- und Leerzeichen enthalten und auch nicht mit einer Ziffer starten. Bei Längenüberschreitung wird der Name abgeschnitten und bei Auftreten unzulässiger Zeichen, werden diese durch Underlines ersetzt.

Das Problem ist auch nicht durch Verwendung sogenannter Namensliterals im SAS Programm zu beheben. (vgl. usage note SN-005156: <http://support.sas.com/techsup/unotes/SN/005/005156.html>)

- | | | |
|----|--|--|
| 1. | Der Parameter benennt die SPSS Quelldatei
Der Pfad bildet zugleich den Speicherort für
das generierte Konverterprogramm
und nach dessen Exekution für
die SAS Tabelle
den Formatkatalog ²⁶
das Textfile ²⁶ | <i>SPSS_C.sav.</i>

<i>SPSS_C.sas</i>

<i>SPSS_C.sas7bdat,</i>
<i>SPSS_C.sas7bcat</i> und
<i>SPSS_C.txt.</i> |
|----|--|--|
- Alle vorhandenen gleichnamigen Zieldateien werden überschrieben!

Mögliche Fehleranzeigen des Makros *%GetSPSS*, deren Ursachen und Behebung:

Ursache	Behebung
<i>Datei</i> ist kein SPSS DATA FILE für MS Windows	wird nicht unterstützt
<i>Datei</i> ist kein SPSS DATA FILE für MS Windows ab Release 10.0	wird nicht unterstützt
Asynchrone Interpretation <i>%GetSPSS</i> erkennt die SPSS Filestruktur nicht	bitte kontaktieren Sie: wflesener@cms.hu-berlin.de
Programm gehört nicht zur <i>Datei</i> (Ausschrift des Konverterprogramms)	die SPSS Quelldatei wurde nach der Generierung des Konverterprogramms modifiziert

Tabelle 3: Mögliche Fehleranzeigen von *%GetSPSS* im LOG Fenster

Das vom Beispiel *MakeSAS.sas* generierte Konverterprogramm kann wegen des Umfangs hier nicht abgedruckt und erläutert werden. Dies würde auch kaum Sinn machen, da das Programm stets dateispezifisch generiert wird. Es gliedert sich in für Anwender leicht erkennbare Segmente:

1. Protokoll von ausgelesenen Metadaten
2. Globale Vereinbarungen
3. Abfiltern der DOCUMENT Textinformationen aus der SPSS Datei²⁷
4. Generieren der user defined formats inkl. String-Formate²⁷
5. Vereinbarungen zur Erstellung der SAS Tabelle inkl. Einsatz von Namensliteralen
6. Auslesen der Daten aus der SPSS Quelldatei



Das Resultat: *library.SPSS_C* (Auszug der ersten 3 Beobachtungen):

Import SPSS into SAS mit %GetSPSS Macro					
Obs	Datum	Tag	Meßzeit	ME_Zeit	Betrieb
1	28.04.84	Sa	12:15:00	13:15:00	Kaltenhausen
2	26.08.84	Su	12:15:00	13:15:00	Kaltenhausen
3	28.12.84	Fr	12:15:00	13:15:00	Kaltenhausen

²⁶ Wird nur bei Bedarf angelegt

²⁷ nur bei Bedarf

Obs	Genotyp	Tier	Jahr	Milch	Fett	Fett_A	VaterTier
1	Frisien	82451	1984	2753	118.00	4.29%	.
2	Frisien	83251	1984	2909	114.30	3.93%	.
3	Frisien	82319	1984	3036	112.00	3.69%	.

3.4 Die Methoden für SPSS => SAS im Vergleich

	 Proc Import	 SAVE	 %GetSPSS
komprimierte SPSS Quelldatei	+	+	+
komprimierte SAS Zieldatei			+
Formatkataloge	- ²⁸	+	+
Filelabel		- ²⁹	+
Länge der max. Variablennamen ³⁰	8	32	32
Namensliterale		- ³¹	+
Variable-Label	+	+	+
Value-Label	+	+	+
Datumstransformation	+	(+) ³²	+
Reskalierung der Wochentage	- ³³	+	+
%-Wert-Multiplikation * 100			+
Datenmatrix	- ³³	+	+

Tabelle 4: Methodenvergleich der Konvertierung von SPSS nach SAS³⁴

4 Fazit

Die Kritik an den nativen Konvertierungstools, die beide Programmsysteme vorhalten, fiel noch 2006 für die alten Versionen³⁵ teilweise berechtigt harsch aus und bewog uns, eigene Programme bereitzustellen. Wir hofften auf deutliche Nachbesserung innerhalb der Programmsysteme SPSS und SAS. Beide Systeme haben sich in nur einem Jahr

²⁸ Nur numerische nutzerdefinierte Formate werden aus VALUE LABEL gebildet

²⁹ Das vorhandene File Label wird ignoriert und statt dessen ein standardisierter Text "*Written by SAS*" erzeugt

³⁰ Einkürzung auf angegebenen Wert; SPSS ließe ab Version 12 max. 64 Zeichen lange Variablennamen zu

³¹ Enthält ein SPSS Variablenname Zeichen, die in SAS Variablennamen unzulässig sind, werden diese durch UNDERLINE ersetzt. Entstehende gleichnamige Variablen bleiben dem SPSS verborgen.

³² Das Europäische Datenformat EDATE wird in Amerikanisches Datumsformat MMDDYY10. konvertiert

³³ Das WKDAY Format in der SPSS Quelldatei bewirkt eine fehlerhafte Datenkonvertierung der Variablen

³⁴ Berücksichtigung: leer => keine; - => fehlerhaft; (+) => ordentlich mit Einschränkung; + => ordentlich

³⁵ SPSS 13.0 und SAS 9.1.3 SP3

bzgl. des gegenseitigen Datenaustausches mit den Versionen SPSS 14.0 und SAS 9.1.3 SP4 deutlich und spürbar entwickelt. Bei wem die Konvertierung mit "Ungemach" verbunden ist, der sollte zunächst dringend seine Softwareversionen kontrollieren und ggf. aktualisieren.

Wie man aber sehen kann, gibt es dennoch Wünsche und Handlungsbedarf.

Unsere Programme sind als eine Hilfe auf Zeit gedacht, denn natürlich können wir nichts direkt an einer Firmen-Software ändern. Solange unsere Programme benötigt werden, werden wir auf jede berechtigte Kritik konstruktiv reagieren. Haftung und Regressansprüche aus Fehlern unserer Programme sind ausgeschlossen.

Literatur

- [1] T. Lochmann, C. Ortseifen "Von SAS nach SPSS und umgekehrt - Umwandlung von Dateien inklusive Formaten bzw. Wertelabels", Proceedings 6. Konferenz der SAS Anwender in Forschung und Entwicklung (KSFE) Dortmund 2002, S. 161-175
<http://www.urz.uni-heidelberg.de/statistik/sas-ah/01.01.04/sas-spss.html>
- [2] SAS Institute Inc. 2005 "Excerpt from Base SAS® 9.1.3 Procedures Guide", Cary, NC: SAS Institute Inc.
<http://support.sas.com/techsup/technote/ts745.pdf>
- [3] SAS Institute Inc. 2004 "SAS® 9.1.3 Language Reference: Dictionary, Volumes 1 and 2", Cary, NC: SAS Institute Inc.
- [4]  Help => SAS Help and Documentation => SAS Products => Base SAS => SAS Language Dictionary => Dictionary of Language Elements => Formats

Anhang A: Die Demonstrationsdaten ³⁶

PROC CONTENTS: *library.SAS_C* (1053 Beobachtungen á 12 Variable):

Variables in Creation Order /* <i>Start.sas</i> */					
#	Variable	Type	Len	Format	Label
1	Datum	Num	8	DDMMYYP10.	Meßtag
2	Tag	Num	8	DOWNAME2.	Wochentag der Messung
3	Meßzeit	Num	8	TIME5.	VarNameTest: Namensliteral
4	ME_Zeit	Num	8	TIME5.	MittelEuropäischeZeit
5	Betrieb	Char	1	\$HOF.	Betrieb
6	Genotyp	Num	4	RASSE.	Genotyp
7	Tier	Num	4	5.	Tiernr.
8	Jahr	Num	4	4.	Jahr
9	Milch	Num	4	6.1	Milchleistung in kg
10	Fett	Num	4	6.1	Fettausbeute in kg
11	Fett%	Num	4	PERCENT6.2	Fettanteil an der Milch in %
12	Vater	Num	8		LabelTest:interne Apostroph's

Variable	Inhalt und Zweck
<i>Datum</i>	über dem Jahr gleichverteilte Zufallszahl, die in ein Datum umgerechnet wurde Test der korrekten Datumstranformation ³⁷ zwischen SAS und SPSS als externe Darstellung mit Europäischem Format <i>dd.mm.yyyy</i>
<i>Tag</i>	= <i>Datum</i> aber mit Format DOWNAME2. Test der korrekten Datumstranformation zwischen SAS und SPSS als externe Darstellung mit 2-buchst. Abkürzung für den Wochentag
<i>Meßzeit</i>	Konstante (= 12:15 Uhr) Test des Namenshandlings bei Auftreten von Ligatur(, Umlaut, Sonderzeichen)
<i>ME_Zeit</i>	Konstante (= <i>Meßzeit</i> +1h) Test auf korrekte Generierung originärer Namen bei Namensanpassung infolge der Verwendung von SAS Namensliteralen
<i>Betrieb</i>	Zeichenkette; nur 'K' oder 'S' (Kaltenhausen, Seegrehna) als Werte <i>Betrieb</i> ist permanent das String-Format \$HOF. zugeordnet <i>Betrieb</i> ist ein Integrity Constraint (check) zugeordnet <i>Betrieb</i> ist Komponente des Composed Index mit Namen <i>Laktatio</i> Test auf Generierung von Valuelabel aus nutzerdefiniertem String-Format

³⁶ Nutzung der Daten für Lehr- und Ausbildungszwecke mit freundlicher Genehmigung von Frau Dr. Ulrike Funke, Landwirtschaftlich-Gärtnerische Fakultät der Humboldt-Universität zu Berlin

³⁷ SPSS Datumsangaben sind Sekunden seit dem 15.10.1582; SAS Datumsangaben sind relative Angaben in Tagen bzgl. dem 1.1.1960

	Test der Umsetzung von Integrity Constraints und Composed Indexes
<i>Genotyp</i>	kategorielle Variable; Werte aus (33, 34, 35) <i>Genotyp</i> ist permanent das Format RASSE. zugeordnet <i>Genotyp</i> ist ein Integrity Constraint (check) zugeordnet Test auf Generierung von Valuelabel aus nutzerdefiniertem Format
<i>Tier</i>	nominelle numerische Variable TierNr. <i>Tier</i> ist Komponente des Composed Index <i>Laktatio</i>
<i>Jahr</i>	numerische Variable Jahr der Erhebung <i>Jahr</i> ist Komponente des Composed Index <i>Laktatio</i>
<i>Milch</i>	metrische Variable; jährliche Milchleistung in kg <i>Milch</i> ist ein Integrity Constraint (not null) zugeordnet
<i>Fett</i>	metrische Variable; jährliche Fettausbeute in kg Wert der zweiten Beobachtung wurde um 0.3 kg erhöht, damit eine reelle Dezimalstelle existiert und bzgl. der Dateikonvertierung getestet wird <i>Fett</i> ist ein Integrity Constraint (not null) zugeordnet
<i>Fett%</i>	Quotient Fett/Milch (metrische Variable) Test der korrekten Behandlung des Namensliterals mit Sonderzeichen (%) und Test der Behandlung von Variablenwerten, denen im SAS ein PERCENTw.d Format zugeordnet ist, weil diese Werte im SAS mit 100 multipliziert ausgegeben werden und erst dadurch eine korrekte Prozentangabe wiedergeben.
<i>VaterTier</i>	numerische Konstante; stets Missing Value; Test auf korrekte Umsetzung eingebetteter Apostroph in Variablelabel und Variablenname mit mehr als 8 Zeichen

Es sind zwei nutzerdefinierte Value-Formate *\$HOF*. mit der String-Variablen *Betrieb* und *RASSE*. mit der numerischen Variablen *Genotyp* assoziiert:

```
proc format library=library;                                /* Start.sas */
  value $hof 'K' = 'Kaltenhausen' 'S' = 'Seegreha';
  value rasse 33 = 'Frisien'      34 = 'Schwarzbunt'
              35 = 'Jersey'      36 = 'Galloway';
run;
```

Der PROC FORMAT Prozedurschritt wurde bereits vor Erzeugung der zu konvertierenden SAS Datei ausgeführt, damit die nutzerdefinierten Formate bei Speicherung permanent zugewiesen werden konnten. Die Formatbeschreibungen befinden sich hier im SAS Katalog *formats.sas7bcat*, der im selben Verzeichnis wie die SAS Tabellen gespeichert ist³⁸.

³⁸ Die Speicherung von SAS Tabellen und der Beschreibung der in ihnen ggf. referenzierten nutzerdefinierten Formate im selben Verzeichnis ist bestimmt eine probate Vorgehensweise.

Das Format RASSE. enthält testweise eine überzählige Kategorie (36), die im Datenmaterial nicht vorkommt.

Anhang B: Resultate bei PROC EXPORT

Das LOG Protokoll

```

2  OPTIONS FMTSEARCH=(library.formats); /* Export.sas */
3  PROC EXPORT DATA   =library.SAS_C
4      OUTFILE="SAS_A.sav"
5      DBMS   =sav
6      REPLACE;
ERROR: The format DDMMYYP was not found or could not be loaded.①
ERROR: The format RASSE was not found or could not be loaded.②
NOTE: "path\SAS_A.sav" was successfully created.③
    
```

Das Resultat SAS A.sav

```

/* Check: Resultat von Export.sas          CheckExport.sps */
SYSFILE INFO FILE='SAS_A.sav'.           ④
GET FILE='SAS_A.sav'.
    
```

SPSS Data View³⁹ (auf 3 cases beschränkt):

	datum ①⑤	tag ⑥	me_zeit ⑦	me_zeit1 ⑦	betrieb ⑧
1	8884	28-APR-1984	12:15:00	13:15:00	K
2	9004	26-AUG-1984	12:15:00	13:15:00	K
3	9128	28-DEC-1984	12:15:00	13:15:00	K

	genotyp ⑧	tier	jahr	milch	fett	fett_ ⑨	vatertie ⑩
1	33	82451	1984	2753	118	0	.
2	33	83251	1984	2909	114⑨	0	.
3	33	82319	1984	3036	112	0	.

Selbstverständlich können der Katalog umbenannt und auch die Formate über mehrere Kataloge gestreut gespeichert werden.

³⁹ Wenn die Datenmatrix spaltenweise, d.h. für alle Werte von Variablen den Wert "0" annimmt, verweist dies darauf, dass die SAS Datei numerische Variable mit Längenattributen, die vom Systemstandard LENGTH=8 abweichen, enthält und zugleich mit einer SAS Version älter SAS 9.1.3 SP4 gearbeitet wurde.



Anmerkungen

Über die eigenmächtige Veränderung der Schreibweise der Variablennamen in der Zieldatei (komplett kleinbuchstabig) könnte man hinwegsehen. Aber:

1. *DDMMYY*. ist ein internes SAS Format. Es sollte wohl immer zu finden und zu laden sein!
2. *RASSE*. ist eines von zwei nutzerdefinierten Formaten. Wieso soll eines gefunden worden sein und das andere nicht, wenn beide im selben zugewiesenen Formatkatalog *library.formats* gespeichert sind?
3. Trotz der Fehler soll nun aber alles erfolgreich verlaufen sein?
4. Das Filelabel geht bei der Konvertierung verloren.
Integrity Constraints und Indizierung aus der SAS Datei finden keine Entsprechungen in der SPSS Datei und sind damit nicht konvertierbar.
PROC EXPORT speichert die SPSS Zieldatei immer UNCOMPRESSED.
5. Folge des Fehlers (vgl. 1.) ist: Die Variable *Datum* wird unformatiert und ohne Transformation⁴⁰ als ganzzahliger numerischer Wert gespeichert.
6. Die Variable *Tag*, die im SAS als Wochentag (*DOWNAME2*.) formatiert ist, bleibt auch im SPSS ein Datumswert, dem das Format *DATE12*. für "*Internationales Datum*" zugewiesen wird.
Eine Ableitung des Wochentages und die Zuweisung des Formates *WKDAY2* käme der Absicht des Anwenders sicher näher.
7. Substitutionen von Umlauten, Ligaturen und Sonderzeichen in SAS Namensliteralen zu einem gültigen SPSS Variablennamen werden häufiger als nötig ausgeführt.⁴¹
8. Nutzerdefinierte Formate bleiben unberücksichtigt. Valuelabel werden nicht gebildet.
9. Die Variable *Fett_* hat scheinbar den konstanten Wert 0. Dies liegt an der Unterdrückung der Dezimalstellen bei der Anzeige im Data View (auch bei *Fett* im Fall 2). Die Werte sind aber korrekt aus SAS nach SPSS übernommen und gespeichert.
Die SAS-Prozentwerte werden bei der Konvertierung nach SPSS aber nicht mit 100 multipliziert und es wird auch nicht das Format *PCT* zugewiesen, das sicher dem SAS Format *PERCENT*. näher käme.
10. Variablennamen in der SPSS Zieldatei werden unnötig kurz auf maximal 8 Zeichen eingekürzt.

Alle Probleme lassen sich jedoch - immer vorausgesetzt, man gewahrt sie - mit unterschiedlichem Aufwand im SPSS nacharbeiten.

⁴⁰ SPSS Datumsangaben sind Sekunden seit dem 15.10.1582; SAS Datumsangaben sind relative Angaben in Tagen bzgl. dem 1.1.1960.

⁴¹ In neueren SPSS Versionen können Umlaute und Ligaturen Bestandteil von Variablennamen sein. Das Sonderzeichen % aus *Fett%* ist aber in SPSS Variablennamen unzulässig.

Anhang C: Resultate mit dem GET SAS Command

 Das Resultat kann man als *SAS B.sav* im SPSS DataView⁴² betrachten:

	Datum ②	Tag ③	Meßzeit	ME_Zeit	Betrieb ④
1	28-APR-1984	28-APR-1984	12:15:00	13:15:00	K
2	26-AUG-1984	26-AUG-1984	12:15:00	13:15:00	K
3	28-DEC-1984	28-DEC-1984	12:15:00	13:15:00	K

	Genotyp	Tier	Jahr	Milch	Fett	Fett_A ⑤	VaterTier
1	Frisien	82451	1984	2753	118,00	,043	.
2	Frisien	83251	1984	2909	114,30	,039	.
3	Frisien	82319	1984	3036	112,00	,037	.

Anmerkungen

Leider ist auch das Ergebnis, das mit SPSS bei der Konvertierung von SAS Dateien in SPSS Dateien erzielt wird, nicht so restlos befriedigend:

1. Das Filelabel⁴³ wurde nicht von SAS nach SPSS übertragen.
2. Das SAS Format der Variablen *Datum* DDMMYYP. wird vom GET SAS Kommando des SPSS nicht in das SPSS Format EDATE (*europäisches Datum*) überführt, sondern in das internationale Datumsformat DATE.
3. Die Variable *Tag*, die im SAS als Wochentag (DOWNAME2.) formatiert ist, bleibt auch im SPSS ein Datumswert, dem das Format DATE11. für "*Internationales Datum*" zugewiesen wird.
Eine Ableitung des Wochentages und die Zuweisung des Formates WKDAY2 käme der Absicht des Anwenders sicher näher.
4. Valuelabel werden nur aus numerischen nutzerdefinierten Formaten abgeleitet.
Das nutzerdefinierte String-Format \$HOF. wurde ignoriert, obwohl das nominale Merkmal *Betrieb* lediglich zwei Kategorien aufweist, deren Codierung mit einem Zeichen deutlich unter dem Limit von 8 Zeichen, die das SPSS als maximale Länge von gespeicherten Werten bei String-Variablen vorsieht, wenn man ihnen ein Value Label zuordnen möchte.
5. Die Variable *Fett%* wird im Zuge der Konvertierung zu recht umbenannt, da das Sonderzeichen % in SPSS Variablenamen unzulässig ist. Aber die SAS-Prozentwerte werden nicht mit 100 multipliziert und es wird auch nicht das SPSS Format PCT zugewiesen.

Auch hier ließen sich alle Probleme - immer vorausgesetzt man gewahrt sie - relativ leicht (bei 4. könnte größerer Aufwand entstehen) im SPSS nacharbeiten.

⁴² In der Wertelabel-Ansicht

⁴³ Kann man leicht mit dem SYSFILE INFO FILE="..." Command überprüfen.

Anhang D: Resultate bei PROC IMPORT

Das LOG Protokoll

```

1  PROC IMPORT  DATAFILE="SPSS_C.sav"      /* Import.sas */
2                OUT      =library.SPSS_A
3                DBMS     =sav
4                REPLACE;
NOTE:  Variable Name Change.  MEßZEIT -> ME_ZEIT      ①
NOTE:  Variable Name Change.  ME_ZEIT -> ME_ZEIT_1    ①
NOTE:  LIBRARY.SPSS A was successfully created          ②

```

Das Resultat mit PROC CONTENTS zu *library.SPSS_A*

Variables in Creation Order						③
#	Variable	Type	Len	Format	Label	
1	DATUM	Num	8	DDMMYY10.	Meßtag	④
2	TAG	Num	8	DATE2.	Wochentag der Messung	②⑤
3	ME_ZEIT	Num	8	TIME8.	VarNameTest:Namensliteral	①
4	ME_ZEIT_1	Num	8	TIME8.	MittelEuropäischeZeit	①
5	BETRIEB	Char	1	\$ETRIEB.	Betrieb	②⑥
6	GENOTYP	Num	8	GENOTYP.	Genotyp	⑦
7	TIER	Num	8	F5.	Tiernr.	⑧
8	JAHR	Num	8	F4.	Jahr	⑧
9	MILCH	Num	8	F4.	Milchleistung in kg	⑧
10	FETT	Num	8	F6.2	Fettausbeute in kg	⑧
11	FETT A	Num	8		Fettanteil in der Milch in %	⑨
12	VATERTIE	⑩Num	8	F1.	LabelTest:interne Apostroph's	

LOG Protokoll zu PROC PRINT: *library.SPSS_A*

```

1  PROC PRINT data=library.SPSS_A;
2  RUN;
ERROR: Format $ETRIEB not found or couldn't be loaded for variable
BETRIEB. ②⑥
NOTE: The SAS System stopped processing this step because of
errors. ②⑥

```

PROC PRINT: *library.SPSS_A* (ohne Formatuweisung zu *Betrieb*)

Obs	DATUM	TAG	ME_ZEIT	ME_ZEIT_1	BETRIEB
	④	②⑤	①	①	⑥
1	28/04/1984	14	12:15:00	13:15:00	K
2	26/08/1984	14	12:15:00	13:15:00	K
3	28/12/1984	14	12:15:00	13:15:00	K

Obs	GENOTYP	TIER	JAHR	MILCH	FETT	FETT_A	VATERTIE
	⑦	⑧	⑧	⑧	⑧	⑨	⑧⑩
1	Frisien	82451	1984	2753	118.00	4.28623	.
2	Frisien	83251	1984	2909	114.30	3.92918	.
3	Frisien	82319	1984	3036	112.00	3.68906	.



Anmerkungen

Über die eigenmächtige Veränderung der Schreibweise der Variablennamen in der Zieldatei (komplett großbuchstabig) könnte man hinwegsehen. Aber:

- Die Ligatur β im Variablennamen *Meßzeit* führt nicht zur Umsetzung des SPSS Variablennamens in ein SAS Namensliteral. Der Variablenname wird in der SAS Zieldatei gegenüber der SPSS Quelldatei modifiziert.
Die Namensveränderung führt zur Namensgleichheit mit einer weiteren Variablen, so dass dann auch noch Folgemodifikationen weiterer Variablennamen nötig werden können.
- Die Aussage *successfully created* ist mit Vorsicht zu betrachten (vgl. 6.+7.)
Den mit dem DOCUMENT Kommando der SPSS Quelldatei hinzugefügten Text, lässt die SAS Prozedur PROC IMPORT völlig unerwähnt.
- PROC CONTENTS zeigt, daß ein Filelabel bei der Konvertierung verloren geht. Die Variablelabel werden korrekt von SAS nach SPSS übertragen.
- Datum* (im SPSS: Format EDATE für *Europäisches Datum*) wird abgesehen vom eigentlich erwünschten Separator-Punkt korrekt transformiert.
- Für die Variable *Tag* ist in der SPSS Quelldatei ein Format WKDAY2 vereinbart. Wieso generiert PROC IMPORT in der SAS Zieldatei ein DATE2. anstelle von korrekt DOWNAME2. bei gleichzeitiger Transformation des Wertevorrates. Das vorliegende Ergebnis ist sonderbar:
Alle Werte von *Tag* werden fehlerhaft in eine Konstante (-137775) konvertiert, die durch DATE2. als 14 wiedergegeben wird - damit ist die Datenmatrix der Zieldatei fehlerhaft!
- Der Versuch, die Datei im View Table Fenster zu betrachten, bewirkt im LOG Fenster:
ERROR: Format \$ETRIEB not found or couldn't be loaded for variable BETRIEB.
und man sieht nichts außer der Fehleranzeige im LOG-Fenster.
Das der Variablen *Betrieb* zugeordnete nutzerdefinierte Format \$ETRIEB. ist nicht generiert worden, existiert also nicht und folglich führt seine gar noch permanente Zuweisung zu Fehlern.
CheckImport.sas zeigt, dass im SAS Katalog *work.formats* keine Format-Beschreibung zu \$ETRIEB eingetragen wurde. Es gibt den Eintrag BETRIEB. Der ist aber inhaltlich leer und hätte mit \$ETRIEB wohl auch wenig zu tun.
- Die generierten Formate zu numerischen Variablen (hier *Genotyp*) müssten aus dem temporären Katalog *work.formats* in einen permanenten Katalog kopiert werden, um sie in späteren SAS Sessions mit der System-Option FMTSEARCH=(*katalog* ...) zzgl. zu bereits vorhandenen Einträgen zu vereinbaren

und verfügbar zu machen. Anderenfalls sind die generierten nutzerspezifischen Formate verloren und ihre Referenzierung in der SAS Datei blockiert zumindest im ersten Zupacken die Arbeit mit der SAS Zielfdatei. Es kann einen erheblichen Arbeitsaufwand verursachen, wenn man die Formate im SAS wieder hinzufügen muss.

8. Ein Format *Fw.[d]*, wie von PROC IMPORT generiert, wird nirgendwo in einer SAS Dokumentation [3] beschrieben - arbeitet aber korrekt.
9. Der Variablen *Fett_A* wird kein SAS Format zugeordnet, was angesichts der Vereinbarung in der SPSS Quelldatei (PCT6.2) zwar nicht falsch wird. Aber man fragt sich natürlich, warum das vorhandene SAS interne Format PERCENT., das der SPSS Vereinbarung und damit der Intension des Anwenders am nächsten käme dennoch nicht genutzt wird.
Fett_A enthält die korrekten Werte. Sie werden aber infolge der ausgebliebenen Formatzuordnung mit dem Standardformat BEST12.5 mit einer für Prozentangaben unüblich hohen Anzahl von Dezimalstellen visualisiert.
10. Variablenamen in der SAS Zielfdatei werden unnötig kurz auf maximal 8 Zeichen eingekürzt.

Anhang E: Resultate mit SAVE TRANSLATE Command



Resultate (SAS-Programm zur Erzeugung nutzerdefinierter Formate)

Das vom SPSS generierte SAS Programm *SPSS_B.sas* zur Ableitung nutzerdefinierter SAS Formate aus den Angaben der SPSS Datei *SPSS_C.sav*:

```
libname library 'path' ;                               /* SPSS_B.sas */
proc format library = library ;
  value TAG /* Wochentag der Messung */
    1 = 'Sunday'                                       ②
    2 = 'Monday'                                       ②
    3 = 'Tuesday'                                       ②
    4 = 'Wednesday'                                    ②
    5 = 'Thursday'                                     ②
    6 = 'Friday'                                       ②
    7 = 'Saturday' ;                                  ②
  value $BETRIEB /* Betrieb */
    'K' = 'Kaltenhausen'
    'S' = 'Seegrehna' ;
  value GENOTYP /* Genotyp */
    33 = 'Frisien'
    34 = 'Schwarzbunt'
    35 = 'Jersey'
    36 = 'Galloway' ;
proc datasets library = library ;
```

```

modify SPSS_B;
  format      Tag TAG.;
  format      Betrieb $BETRIEB.;
  format      Genotyp GENOTYP.;
quit;
    
```



Anmerkung

Das vom SPSS erzeugte Programm für die nutzerdefinierten Formate ist korrekt und wirklich gut. Einzige Frage: Warum erzeugt eine deutschsprachige SPSS-Installation englische Wochentagsnamen (②)?



Resultate (die Datenmatrix *library.SPSS_B* z.B. mit PROC PRINT)

Das Programm *SPSS_B.sas* ist zuvor einmalig auszuführen!

Obs	Datum ①	Tag ②	Me_zeit ③	ME_Zeit ③	Betrieb
1	04/28/1984	Saturday	12:15:00	13:15:00	Kaltenhausen
2	08/26/1984	Sunday	12:15:00	13:15:00	Kaltenhausen
3	12/28/1984	Friday	12:15:00	13:15:00	Kaltenhausen

Obs	Genotyp	Tier ④	Jahr ④	Milch ④	Fett ④	Fett_A ④⑤	VaterTier ④
1	Frisien	82451	1984	2753	118.000	4.29623	.
2	Frisien	83251	1984	2909	114.300	3.93918	.
3	Frisien	82319	1984	3036	112.000	3.69909	.

Um die Wiedergabe der Daten zu verstehen, verschafft man sich einen Überblick zu den den Variablen aktuell zugewiesenen Formaten mit PROC CONTENTS:

library.SPSS_B

Variables in Creation Order					⑥	
#	Variable	Type	Len	Format	Label	
1	Datum	Num	8	MMDDYY10.	Meßtag	①
2	Tag	Num	8	TAG.	Wochentag der Messung	②
3	Me_zeit	Num	8	TIME18.	VarNameTest: Namensliteral	③
4	ME_Zeit	Num	8	TIME18.	MittelEuropäischeZeit	③
5	Betrieb	Char	1	\$BETRIEB.	Betrieb	
6	Genotyp	Num	8	GENOTYP.	Genotyp	
7	Tier	Num	8		Tiernr.	④
8	Jahr	Num	8		Jahr	④
9	Milch	Num	8		Milchleistung in kg	④
10	Fett	Num	8		Fettausbeute in kg	④
11	Fett_A	Num	8		Fettanteil an der Milch in %	④
12	VaterTier	Num	8		LabelTest: interne Apostroph's	④

Anmerkungen

1. *Datum* (im SPSS: Format EDATE für *Europäisches Datum*) wird zwar korrekt transformiert, aber in der amerikanischen Schreibweise wiedergegeben, weil das Datumsformat MMDDYY10. permanent zugewiesen ist. Dies kann sehr leicht zu Fehldeutungen führen.
2. *Tag* wird korrekt konvertiert, aber man könnte sich vorstellen, dass SPSS entsprechend der installierten Sprachversion das Format TAG. generiert.
3. Die Ligatur β im Variablennamen *Meßzeit* führt nicht zur Umsetzung des SPSS Variablennamens in ein SAS Namensliteral, sondern der Variablenname wird in der SAS Zielfeile gegenüber der SPSS Quelldatei modifiziert.
Die Namensveränderung zu *Me_zeit* führt, da SAS nicht case sensitive arbeitet, zur Namensgleichheit mit einer der Variablen *ME_Zeit*, so dass nun auch noch Folgemodifikationen weiterer Variablennamen nötig würden. Diese Notwendigkeit erkennt SPSS 14.0⁴⁴ aber nicht, was die Verwendbarkeit der SAS Datei *SPSS_B.sas7bdat* tangiert. Weitere Variablen gleichen Namens sind in einer SAS Tabelle unzulässig und wären z.B. mit keinem VAR Statement adressierbar.
Bei beiden Uhrzeiten (*Me_zeit* und *ME_Zeit*) wird ein TIME w . Format mit Feldweite $w=18$ zugewiesen, obwohl Feldweite $w=8$ ausreichte.
4. Für allen diese Variablen wurde kein Format vereinbart, so dass sie mit dem Standardformat BEST12.5 mit einer z.B. für Prozentangaben unüblich hohen Anzahl von Dezimalstellen visualisiert werden.

⁴⁴ Namensgleiche Variablen in der SAS Zielfeile hatte SPSS 13.0 mit dem SAVE TRANSLATE Command nicht erzeugt. Dafür war jedoch das Programm, das die nutzerdefinierten SAS Formate bildete syntaktisch mehrfach fehlerhaft.

5. Der Variablen *Fett_A* ist im SPSS Quelldatei ein Format PCT6.2 zugewiesen. Bei der Dateikonvertierung in die SAS Zieldatei wird der Wert der Variablen zwar korrekt in die Datenmatrix übernommen, aber es wird nicht das entsprechende SAS interne Format PERCENT $w.d$ zugewiesen. Würde es zugewiesen, dann erforderte es jedoch eine Anpassung aller Werte von *Fett_A* durch eine Division durch 100.
6. Das Filelabel geht bei der Konvertierung verloren und wird durch einen nichts sagenden Standard ersetzt: "*Written by SAS*" - warum so etwas?

Verwendet man bereits bei der Namensgebung für Variablen in der SPSS Quelldatei keine Umlaute, Ligaturen oder gar Sonderzeichen, so sollte sich der ohnehin wohl seltene Fall, dass gleiche Variablennamen in der SAS Zieldatei entstehen, immer ausschließen lassen und damit wäre das SAVE TRANSLATE Command des SPSS dann wirklich eine gute Wahl für die Dateikonvertierung einer SPSS Datei in eine SAS Datei.