SAS Format: Fallen und Tricks

Sascha Rampersad
inVentiv Health Germany GmbH –
a Syneos Health Company
Frankfurter Strasse 233, Triforum, Haus C1
63263 Neu-Isenburg
sascha.rampersad@syneoshealth.com

Zusammenfassung

Formate kann man als SAS Programmierer nicht wirklich umgehen. Formate liefern tolle Funktionalitäten um einfache Lösungen für komplexe Probleme zu erarbeiten. Dieses Papier zeigt zu dem Thema Gruppieren und Selektieren von Beobachtungen Anwendungsbeispiele und Fallen die zu unerwünschten Ergebnissen führen.

Schlüsselwörter: SAS Format, Input, Dezimalzahlen, Selektion, Gruppenbildung

1 SAS Format: Tricks und Fallen

1.1 Vorsicht! Format *w.d* im Input Statement

Input von Ganzzahlen gemischten mit Dezimalzahlen. Wer bei einem Input Statement das Format w.d benutzt kann eine böse Überraschung erleben. Ganzzahlen werden als Nachkommastellen eingelesen und führen zu falschen Ergebnissen.

```
data korb;
  input kdn_nr name $ richtig falsch 5.2;
  datalines;
10 pety 1.00 1.00
11 carl 0.25 0.25
12 hans    1    1
13 mike   41   41
14 jose  123  123
15 jimy   5    5
;
run;
```

Schauen wir uns den Output an, dann sehen wir falsche Werte bei der Variablen die das w.d Format im Input Statement benutzt.

```
proc print data=korb width=uniform noobs; run;
```

kdn_nr	name	richtig	falsch
10	pety	1.00	1.00
11	hans	1.00	0.01
12	mike	41.00	0.41
13	jose	123.00	1.23
14	carl	0.25	0.25
15	jim	5.00	0.05

1.2 Selektion mit einem Format

Kunden mit einem Format selektieren. Mit einem Format können sehr elegant alle Beobachtungen mit einer bestimmten Ausprägung selektiert werden. Wir selektieren mit der Variablen IDN aus folgender Datei nur Beobachtungen mit den Werten 2, 4 und 5.

NAME	SEX	AGE	HEIGHT	WEIGHT	VIEWKEY	ID	IDN
Alfred	M	14	69	112.5	N	1	1
Alice	F	13	56.5	84	Y	2	2
Barbara	F	13	65.3	98	N	3	3
Carol	F	14	62.8	102.5	Y	4	4
Henry	M	14	63.5	102.5	Y	5	5
James	M	12	57.3	83	N	6	6
Jane	F	12	59.8	84.5	N	7	7
Janet	F	15	62.5	112.5	N	8	8
Jeffrey	M	13	62.5	84	N	9	9
John	M	12	59	99.5	N	10	10
Joyce	F	11	51.3	50.5	N	11	11
Judy	F	14	64.3	90	N	12	12
Louise	F	12	56.3	77	N	13	13
Mary	F	15	66.5	112	N	14	14
Philip	M	16	72	150	N	15	15
Robert	M	12	64.8	128	N	16	16
Ronald	M	15	67	133	N	17	17
Thomas	M	11	57.5	85	N	18	18
William	M	15	66.5	112	N	19	19

Zuerst erstellen wir ein Format mit den Werten die selektiert werden sollen.

```
proc format;
    value keyn
        2,4,5 = 'Y'
        other = 'N' ;
run;
```

Dann benutzen wir das Format um die gewünschten Beobachtungen zu selektieren.

```
data subset;
    set large;
    where put(idn,keyn.) = 'Y';
run;
```

Output:

NAME	SEX	AGE	HEIGHT	WEIGHT	VIEWKEY	ID	IDN
Alice	F	13	56.5	84	Y	2	2
Carol	F	14	62.8	102.5	Y	4	4
Henry	M	14	63.5	102.5	Y	5	5

1.3 Mit Längenangaben im Format selektieren

Mit nur den ersten beiden Stellen wird eine Variablen selektiert. Zuerst erstellen wir das passende Format und dann wenden wir es im Where Statement an.

```
proc format;
 invalue $key
   'Al','Jo','Ju' = 'Y'
   other = 'N' ;
run;

data result;
 set sashelp.class;
 where input(name,$key2.) = 'Y';
run;
```

Input Datensatz

NAME	SEX	AGE	HEIGHT	WEIGHT
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83
Jane	F	12	59.8	84.5
Janet	F	15	62.5	112.5
Jeffrey	M	13	62.5	84
John	M	12	59	99.5
Joyce	F	11	51.3	50.5
Judy	F	14	64.3	90
Louise	F	12	56.3	77
Mary	F	15	66.5	112
Philip	M	16	72	150
Robert	M	12	64.8	128
Ronald	M	15	67	133
Thomas	M	11	57.5	85
William	M	15	66.5	112

Output:

```
proc print width=uniform noobs;
run;
```

NAME	SEX	AGE	HEIGHT	WEIGHT
Alfred	M	14	69.0	112.5
Alice	F	13	56.5	84.0
John	M	12	59.0	99.5
Joyce	F	11	51.3	50.5
Judy	F	14	64.3	90.0

1.4 Format im Format

Um ein Format zu erweitern ohne es zu ändern integrieren wir ein Format in ein anderes Format. Hier wird das Format „sex“ um den Wert „Missing“ erweitert.

```
proc format;
    value sex
        1='Men'
        2='Woman'
        ;
    value msex
        .='Missing'
        other = [sex.]
        ;
run;
```

Output mit sex Format

ID	NAME	SEXN	SEX
1	hans	1	Men
2	klaus	1	Men
3	ute	2	Woman
4	sven		
5	gaby	1	Men

Output mit msex Format

ID	NAME	SEXN	SEX
1	hans	1	Men
2	klaus	1	Men
3	ute	2	Woman
4	sven		Missing
5	gaby	1	Men

1.5 Gruppenbildung mit einem Format

Mit den Schlüsselwörtern *low* und *high* können alle Werte zugeordnet werden die kleiner oder grösser sind als eine bestimme Zahl.

```
proc format
  value grage
    low-13  = 'jung'
    14      = 'normal'
    15-high = 'alt';
run;

data view(keep=name age grage);
  set sashelp.class;
  format grage grage.;
  grage=age;
run;
```

Output:

```
proc print width=uniform noobs;
run;
```

Alfred	14	normal
Alice	13	jung
Barbara	13	jung
Carol	14	normal
Henry	14	normal
James	12	jung
Jane	12	jung
Janet	15	alt
Jeffrey	13	jung
John	12	jung
Joyce	11	jung
Judy	14	normal
Louise	12	jung
Mary	15	alt
Philip	16	alt
Robert	12	jung
Ronald	15	alt
Thomas	11	jung
William	15	alt

1.6 Vorsicht! Berechnungen mit Dezimalzahlen

Computer können nicht rechnen? Für SAS ergibt 15,6 - 14,4 nicht 1,2 sondern 1,200000000000001. Berechnung mit Dezimalzahlen können von Computern nicht korrekt verarbeitet werden. Ein Ergebnis muss gerundet werden, damit es weiterverarbeitet werden kann. Problem ist die begrenzte Möglichkeit Zahlen mit mehreren Stellen zu speichern. Ein Format verändert nur die Ausgabe nicht den Wert einer Variablen.

Input Datensatz:

```
data korb;
  input zahl1 zahl2 subtraktion addition;
  datalines;
14.7 11.9 2.8 26.6
15.4 14.4 1.0 29.8
;
run;
```

Kann SAS wirklich nicht rechnen? Schauen wir uns das Ergebnis genauer an und dann wird klar, dass es bei Berechnungen mit Dezimalzahlen zu unerwarteten Ergebnissen kommen kann.

```
data frucht;
  set korb;
  format ergebnis best32. gleich nj.;
  ergebnis = zahl1 - zahl2;
  gleich = (ergebnis = subtraktion);
  output;
run;
```

Output:

```
zahl1    zahl2    subtraktion          ergebnis    gleich
 14.7     11.9        2.8      2.79999999999999      nein
 15.4     14.4        1.0                     1        ja
```

Kann ein Format das Problem beheben, nein. Ein Format verändert nur die Ausgabe nicht den Wert einer Variablen. Das Ergebnis bleibt das gleiche, 2.8 ist nicht gleich 2.8!

```
data frucht;
  set korb;
  format ergebnis best32. ertrag best12. gleich nj.;
  ergebnis = zahl1 - zahl2;
  ertrag = ergebnis;
  gleich = (ertrag = subtraktion);
 output;
run;
```

Output:

```
zahl1    zahl2    subtraktion          ergebnis    ertrag   gleich
 14.7     11.9        2.8      2.79999999999999       2.8     nein
 15.4     14.4        1.0                     1         1       ja
```

Wie kann ich das Ergebnis einer Berechnung mit Dezimalzahlen ohne Probleme vergleichen oder weiterverarbeiten? Der Wert muss gerundet werden.

```
data frucht;
  set korb;
  format ergebnis best32. ertrag best12. gleich nj.;
  ergebnis = zahl1 - zahl2;
  ertrag = round(ergebnis,0.1);
  gleich = (ertrag = subtraktion);
 output;
run;
```

Output:

zahl1	zahl2	subtraktion	ergebnis	ertrag	gleich
14.7	11.9	2.8	2.79999999999999	2.8	ja
15.4	14.4	1.0	1	1	ja

1.7 Format mit Informat aus einer Datei erstellen

Die Reihenfolge mit einem Format ändern. Formate können mehr als nur die Reihenfolge einer Variablen ändern. Ich möchte aber auch dazu einen kleinen Code-Auszug zu diesem Thema vorstellen. Die Variable Car soll nach Marke, PS und dann Model sortiert werden.

Sortieren die Variable Car alphabetisch

Car	Make	Horsepower	Model
Porsche 911 Carrera 4S coupe 2dr (convert)	Porsche	315	911 Carrera 4S coupe 2dr (convert)
Porsche 911 Carrera convertible 2dr (coupe)	Porsche	315	911 Carrera convertible 2dr (coupe)
Porsche 911 GT2 2dr	Porsche	477	911 GT2 2dr
Porsche 911 Targa coupe 2dr	Porsche	315	911 Targa coupe 2dr
Porsche Boxster S convertible 2dr	Porsche	258	Boxster S convertible 2dr
Porsche Boxster convertible 2dr	Porsche	228	Boxster convertible 2dr
Porsche Cayenne S	Porsche	340	Cayenne S
Volkswagen GTI 1.8T 2dr hatch	Volkswagen	180	GTI 1.8T 2dr hatch
Volkswagen Golf GLS 4dr	Volkswagen	115	Golf GLS 4dr
Volkswagen Jetta GL	Volkswagen	115	Jetta GL
Volkswagen Jetta GLI VR6 4dr	Volkswagen	200	Jetta GLI VR6 4dr
Volkswagen Jetta GLS TDI 4dr	Volkswagen	100	Jetta GLS TDI 4dr
Volkswagen New Beetle GLS 1.8T 2dr	Volkswagen	150	New Beetle GLS 1.8T 2dr
Volkswagen New Beetle GLS convertible 2dr	Volkswagen	115	New Beetle GLS convertible 2dr
Volkswagen Passat GLS 1.8T	Volkswagen	170	Passat GLS 1.8T
Volkswagen Passat GLS 4dr	Volkswagen	170	Passat GLS 4dr

Wenn wir alles richtig machen dann werden wir später dieses Ergebnis sehen. Die Variable Car sortiert nach Marke, PS und Model!

Sortieren die Variable Car nach Make, Horsepower und Model

Car	Make	Horsepower	Model
Porsche Boxster convertible 2dr	Porsche	228	Boxster convertible 2dr
Porsche Boxster S convertible 2dr	Porsche	258	Boxster S convertible 2dr
Porsche 911 Carrera 4S coupe 2dr (convert)	Porsche	315	911 Carrera 4S coupe 2dr (convert)
Porsche 911 Carrera convertible 2dr (coupe)	Porsche	315	911 Carrera convertible 2dr (coupe)
Porsche 911 Targa coupe 2dr	Porsche	315	911 Targa coupe 2dr
Porsche Cayenne S	Porsche	340	Cayenne S
Porsche 911 GT2 2dr	Porsche	477	911 GT2 2dr
Volkswagen Jetta GLS TDI 4dr	Volkswagen	100	Jetta GLS TDI 4dr
Volkswagen Golf GLS 4dr	Volkswagen	115	Golf GLS 4dr
Volkswagen Jetta GL	Volkswagen	115	Jetta GL
Volkswagen New Beetle GLS convertible 2dr	Volkswagen	115	New Beetle GLS convertible 2dr
Volkswagen New Beetle GLS 1.8T 2dr	Volkswagen	150	New Beetle GLS 1.8T 2dr
Volkswagen Passat GLS 1.8T	Volkswagen	170	Passat GLS 1.8T
Volkswagen Passat GLS 4dr	Volkswagen	170	Passat GLS 4dr
Volkswagen GTI 1.8T 2dr hatch	Volkswagen	180	GTI 1.8T 2dr hatch
Volkswagen Passat GLX V6 4MOTION 4dr	Volkswagen	190	Passat GLX V6 4MOTION 4dr

Um eine Variable nach verschieden Kriterien zu sortieren, kann es hilfreich sein, ein Format mit passendem Informat zu erstellen. Wir definieren die Länge der Variablen und wollen den Formatname vorne als erster Variable stehen haben, dann folgt ein Keep Statement.

```
data fmtout;
   length fmtname $8 start label help $200 type $1;
  retain fmtname 'carfmt';
  keep fmtname type start label startn;
  set cars;
```

Für die richtige Sortierung definieren wir die Variable “startn”.

```
startn = _n_; /*fuer die sortierung*/
```

Jetzt erstelle wir das Format und Informat und definieren die dafür nötigen Variablen Type (numerisch oder character), Start und Label. Automatisch wird dann auch die Variable End erzeugt.

```
/*format*/
type = 'N';
start = strip(put(_n_, best.));
label = strip(car);
output;

/*informat*/
if type eq 'N' then type='I';
else if type eq 'C' then type='J';
else abort;
help=strip(start);
start=strip(label);
label=strip(help);
output;
```

Zum Schluss müssen wir die Formate noch sortieren und als Datei speichern.

```
/*formate sortieren*/
proc sort data=fmtout nodupkey;
  by fmtname type startn label;
run;
/*formate in katalog schreiben*/
proc format cntlin=fmtout;
run;
```

Die Variable CARN erstellen wir mit dem Informat und legen dann das Format auf die Variable.

```
format Carn carfmt.;
Carn=input(car, carfmt.);
```

Ein Blick auf das Format zeigt uns die Zuordnungen von Start und Label. Die Variable CARN ist jetzt nach der gewünschten Reihenfolge sortiert und kann verwendet werden.

Format carfmt

START	LABEL
1	Porsche Boxster convertible 2dr
2	Porsche Boxster S convertible 2dr
3	Porsche 911 Carrera 4S coupe 2dr (convert)
4	Porsche 911 Carrera convertible 2dr (coupe)
5	Porsche 911 Targa coupe 2dr
6	Porsche Cayenne S
7	Porsche 911 GT2 2dr
8	Volkswagen Jetta GLS TDI 4dr
9	Volkswagen Golf GLS 4dr
10	Volkswagen Jetta GL
11	Volkswagen New Beetle GLS convertible 2dr
12	Volkswagen New Beetle GLS 1.8T 2dr
13	Volkswagen Passat GLS 1.8T
14	Volkswagen Passat GLS 4dr

Sortierte Variable Carn!

CARN
Porsche Boxster convertible 2dr
Porsche Boxster S convertible 2dr
Porsche 911 Carrera 4S coupe 2dr (convert)
Porsche 911 Carrera convertible 2dr (coupe)
Porsche 911 Targa coupe 2dr
Porsche Cayenne S
Porsche 911 GT2 2dr
Volkswagen Jetta GLS TDI 4dr
Volkswagen Golf GLS 4dr
Volkswagen Jetta GL
Volkswagen New Beetle GLS convertible 2dr
Volkswagen New Beetle GLS 1.8T 2dr
Volkswagen Passat GLS 1.8T
Volkswagen Passat GLS 4dr