

2. Dateneingabe und -transformation

2.1 Eingabe über die Tastatur

2.2 Transformationen

2.3 Eingabe eines externen ASCII-Files

2.4 Varianten der INPUT-Anweisung, Formatierungselemente

2.5 Ein- und Ausgabe von SAS-Systemfiles

2.6 Zusammenfügen von Files

2.7 OUTPUT-Anweisung

2.8 DO-Schleifen im DATA-Step

Allgemeine Syntax

```
DATA <dateiname <(dateioptionen)>>;
```

...

```
RUN;
```

<... > kennzeichnet optionale Parameter

- INFILE ' ... ' ; - externes File
INPUT ... ;
- INPUT ... ; - Tastatur
CARDS;
Daten
;
- SET dateiname; - SAS-System-File

+ zusätzliche Anweisungen

Programmbeispiele: Eingabe... .sas

2.1. Eingabe über die Tastatur

```
DATA Eingabe1;  
INPUT a $ x y z;  
s = x + y + z;  
CARDS;  
b 1 2 3  
c 4 5 6  
d 7 8 9  
RUN;  
/* Erläuterung dazu: siehe Datei Eingabe.sas. */  
PROC PRINT; RUN;
```

Mit PROC PRINT; wird die gesamte erzeugte Datei ausgedruckt ins Output-Fenster.

Aktivierung des Programms geschieht durch

- klicken auf MännchenLogoGrafik oder
- klicken auf 'run' → 'submit'
- F3-Taste

Bem: Die Datei Eingabe1 hat

3 Beobachtungen (Individuen, Wertesätze)

5 Merkmale (Variablen)

a, x, y, z und s.

Bemerkung: Mit
Solutions
Analysis
Interactive Data Analysis

kann die erzeugte Datei auch angesehen werden, je nach DATA-Kommando:

	Bibliothek	Dateiname
DATA Eingabe1;	WORK	Eingabe1
DATA sasuser.Eing1;	SASUSER	Eing1
DATA;	WORK	DATA1 DATA2 ...

Bemerkung: Dateien, die sich im Arbeitsverzeichnis WORK befinden, werden am Ende der Sitzung gelöscht.

Bemerkung: Die Variante “DATA sasuser.Eing1;” nicht verwenden.

Bem.: Es werden automatisch die Variablen

N

ERROR

angelegt. (vgl. Log-Fenster)

Beispiel: Nichtzulässige mathematische Operationen führen

zu *_ERROR_* = 1 und das Ergebnis wird auf
“.” (missing value) gesetzt. (vgl. Beispiel Ein-
gabe2)

Bemerkung: Ausfälle (missing values) werden durch einen
Punkt (.) kodiert.

2.2. Transformationen

- immer nach der INPUT-Anweisung angeben!

a) IF THEN ELSE und logische Operationen

vgl. Programm Eingabe2

b) Funktionen

vgl. Programm Eingabe3

c) Arithmetische Operationen

** Potenzierung

* Multiplikation

/,+,-

zu **IF THEN ELSE**

wenn jeweils nur eine Anweisung ausgeführt werden soll:

```
IF (log. Ausdruck) THEN Anweisung;  
ELSE Anweisung;
```

wenn mehrere Anweisungen ausgeführt werden sollen

```
IF (log. Ausdruck) THEN Anweisung;  
ELSE DO  
Anweisung1; Anweisung2; ...  
END;
```

```
IF (log. Ausdruck) THEN DO
```

```
Anweisung1; ...  
END;  
ELSE DO  
Anweisung1; Anweisung2; ...  
END;
```

2.3. Eingabe durch externes File (ASCII)

```
DATA Eingabe4;  
INFILE 'Pfadname';  
INPUT Variablen;  
evtl. Transformationen;  
RUN;
```

Bemerkungen:

- Diese Eingabe ist formatfrei, d.h. die Variablen sind im Rohdatenfile durch Leerzeichen getrennt.
- Sind die Eingabedaten durch ein anderes Zeichen, z.B. ';', getrennt, dann ist in der INFILE-Anweisung die Option
DELIMITER=';' (oder DLM=';')
anzugeben.

Tabulatorzeichen: DLM='09'X;

- Bedingungen:
fehlende Werte: . (Punkt)
alphanumerische Zeichenketten dürfen keine Leerzeichen enthalten.
- Die INPUT-Anweisung kann auch abgekürzt werden,
z.B.
INPUT V1-V7;

2.3. Eingabe durch externes File (EXCEL)

```
PROC IMPORT datafile="... .xls"  
out Dateiname; /*SAS-Datei*/  
getnames=no; *Var.namen werden nicht übernommen  
sheet=spreadsheetname;  
RUN;
```

2.4 Wichtige Varianten der INPUT-Anweisung

- bisher: formatfrei

```
INPUT a $ b $ c d;
```

- formatiert-spaltenorientiert

```
INPUT a $ 1-10 b $ 11 c 13-14 .1;
```

- formatiert-über die Zeichenlänge

```
INPUT a $10. b $ 1. c 2. d 5.1;
```

Eingabeformate

w. 2. standard numerisch

w.d 2.1 standard numerisch mit Dezimalstelle

\$w. \$10 Zeichenlänge

Beispiele:

Eingabe5.

Eingabe6 (komplexere Formate)

Weitere Formatierungselemente

- Spaltenzeiger

@ n: Zeige auf Spalte n (z.B. @12)

+n: Setze den Zeiger n Positionen weiter

+Zeigervariable Setze den Zeiger entsprechend dem Wert der Zeigervariablen weiter

@ Zeigervariable Setze den Zeiger auf die Spalte mit dem Wert der Zeigervariablen

Bsp.: A = 15; INPUT @ A name \$;

- Zeilenzeiger

n: Zeige auf Spalte 1 der n-ten Zeile

variable: Zeige auf Spalte 1 der durch variable bestimmten Zeilennummer

/: Zeige auf Spalte 1 der nächsten Zeile

- Zeilenhalter

@ (nachgestellt) Datenzeile wird von mehreren INPUT-Anweisungen gelesen

@ @ (nachgestellt) Aus einer Eingabezeile werden mehrere Beobachtungen gelesen

2.5 Ein- u. Ausgabe von SAS-System Files

a) Abspeichern einer permanenten SAS-Datei

```
DATA sasuser.banknote;  
    /* Eine Datei mit dem Namen 'banknote'  
       wird im SAS-internen Verzeichnis  
       'sasuser' gespeichert */  
    <INFILE ' Pfadname der einzulesenden  
           Datei;>  
    INPUT Formatangaben;  
    <CARDS;  
        Daten (zeilenweise)  
    ; >  
RUN;
```

b) Einlesen einer SAS-Datei

```
DATA banknote1;  
    SET sasuser.banknote < (Optionen)>;  
RUN;
```

```
DATA;  
SET sasuser.banknote <(Optionen)>;  
RUN;
```

Optionen

DROP = Varname(n); Weglassen von Variablen

KEEP = Varname(n); nur diese Variablen
werden verwendet

FIRSTOBS=integer; 1. zu verarbeitender
Wertesatz

OBS = integer; letzter zu verarbeitender
Wertesatz

RENAME = (alter Varname = neuer Varname);