

## 12. ANHÄNGE:

<u>12.1. Versionen 2 und 3 des SD:</u> .....	65
<p>Die Reihenfolge der Items und ihre Links-Rechts-Orientierung wurde durch Zufall bestimmt. Dabei wurde folgende Prozedur zweimal durchgeführt: Aus einem Säckchen wurden Lottosteine mit den Nummern 1 bis 46 gezogen, um die Reihenfolge der Items festzulegen. Danach wurde pro Adjektivpaar einmal eine Münze geworfen, um die Links-Rechts-Orientierung zu bestimmen.</p>	
<u>12.2. kompletter Versuchsplan:</u> .....	69
<u>12.3. Testprotokolle:</u> .....	71
<u>12.4. Listing des PC-Programmes für die Datenerfassung und -auswertung:</u> .....	74
<u>12.5. Teil des Ausdruckes des PC-Programmes (Q-Werte, Standardabweichungen, Koordinaten des semantischen Raumes, D-Werte):</u> .....	115
<u>12.6. Ausdruck der BEDAG Faktoranalyse:</u> .....	118
<u>12.7. Ausdruck der BEDAG Varianzanalyse über die Standardabweichungen:</u> .....	123
<u>12.8. Ausdruck der BEDAG Varianzanalysen über die drei Faktoren und der dazugehörigen Einzelvergleiche:</u> .....	125
<u>12.9. restliche 8 Mittelwertsprofile der 9 Faktorstufenkombinationen:</u> .....	133
<u>12.10. restliche 5 Mittelwertsprofile der 6 Faktorstufen:</u> ..	150
<u>12.11. Einzelvergleiche der Varianzanalyse über die Standardabweichungen (zum Kap. 9.7.2.):</u> .....	161

## 12.1. Versionen 2 und 3 des SD:

Version 2, 1. Seite

geordnet	0	0	0	0	0	0	zufällig
leer	0	0	0	0	0	0	voll
aggressiv	0	0	0	0	0	0	friedlich
ernst	0	0	0	0	0	0	verspielt
gehemmt	0	0	0	0	0	0	schwungvoll
rauh	0	0	0	0	0	0	glatt
hässlich	0	0	0	0	0	0	schön
gedehnt	0	0	0	0	0	0	straff
klingend	0	0	0	0	0	0	dumpf
vorwärtsstrebend	0	0	0	0	0	0	kreisend
lebendig	0	0	0	0	0	0	tot
froh	0	0	0	0	0	0	traurig
dunkel	0	0	0	0	0	0	hell
symmetrisch	0	0	0	0	0	0	unsymmetrisch
blass	0	0	0	0	0	0	farbig
subjektiv	0	0	0	0	0	0	objektiv
munter	0	0	0	0	0	0	klagend
feierlich	0	0	0	0	0	0	keck
gedämpft	0	0	0	0	0	0	kräftig
aufregend	0	0	0	0	0	0	beruhigend
schwankend	0	0	0	0	0	0	stabil
drängend	0	0	0	0	0	0	behaglich
erregt	0	0	0	0	0	0	gemessen

glänzend	0	0	0	0	0	0	trübe
vertraut	0	0	0	0	0	0	fremd
verträumt	0	0	0	0	0	0	nüchtern
stockend	0	0	0	0	0	0	fliessend
klar	0	0	0	0	0	0	verschwommen
aufwärts	0	0	0	0	0	0	abwärts
fein	0	0	0	0	0	0	grob
passiv	0	0	0	0	0	0	aktiv
schnell	0	0	0	0	0	0	langsam
warm	0	0	0	0	0	0	kühl
bestimmt	0	0	0	0	0	0	undefinierbar
gefühlvoll	0	0	0	0	0	0	kühl
rund	0	0	0	0	0	0	eckig
straff	0	0	0	0	0	0	schleichend
hart	0	0	0	0	0	0	weich
zart	0	0	0	0	0	0	robust
statisch	0	0	0	0	0	0	dynamisch
regelmässig	0	0	0	0	0	0	unregelmässig
aufdringlich	0	0	0	0	0	0	zurückhaltend
angespannt	0	0	0	0	0	0	gelöst
angenehm	0	0	0	0	0	0	unangenehm
geschmeidig	0	0	0	0	0	0	starr
lebhaft	0	0	0	0	0	0	müde

hässlich	0	0	0	0	0	0	schön
kreisend	0	0	0	0	0	0	vorwärtsstrebend
keck	0	0	0	0	0	0	feierlich
aufdringlich	0	0	0	0	0	0	zurückhaltend
verschwommen	0	0	0	0	0	0	klar
fremd	0	0	0	0	0	0	vertraut
zufällig	0	0	0	0	0	0	geordnet
kühl	0	0	0	0	0	0	gefühlvoll
unangenehm	0	0	0	0	0	0	angenehm
lebhaft	0	0	0	0	0	0	müde
regelmässig	0	0	0	0	0	0	unregelmässig
aktiv	0	0	0	0	0	0	passiv
schleichend	0	0	0	0	0	0	straff
stockend	0	0	0	0	0	0	fliessend
abwärts	0	0	0	0	0	0	aufwärts
hell	0	0	0	0	0	0	dunkel
nüchtern	0	0	0	0	0	0	verträumt
glatt	0	0	0	0	0	0	rauh
leer	0	0	0	0	0	0	voll
grob	0	0	0	0	0	0	fein
gemessen	0	0	0	0	0	0	erregt
objektiv	0	0	0	0	0	0	subjektiv
verspielt	0	0	0	0	0	0	ernst

gelöst	0	0	0	0	0	0	angespannt
schnell	0	0	0	0	0	0	langsam
unsymmetrisch	0	0	0	0	0	0	symmetrisch
friedlich	0	0	0	0	0	0	aggressiv
eckig	0	0	0	0	0	0	rund
kühl	0	0	0	0	0	0	warm
hart	0	0	0	0	0	0	weich
gehemmt	0	0	0	0	0	0	schwungvoll
starr	0	0	0	0	0	0	geschmeidig
trübe	0	0	0	0	0	0	glänzend
undefinierbar	0	0	0	0	0	0	bestimmt
stabil	0	0	0	0	0	0	schwankend
zart	0	0	0	0	0	0	robust
dynamisch	0	0	0	0	0	0	statisch
kräftig	0	0	0	0	0	0	gedämpft
klagend	0	0	0	0	0	0	munter
bebaglich	0	0	0	0	0	0	drängend
traurig	0	0	0	0	0	0	froh
straff	0	0	0	0	0	0	gedehnt
dumpf	0	0	0	0	0	0	klingend
tot	0	0	0	0	0	0	lebendig
bläss	0	0	0	0	0	0	farbig
beruhigend	0	0	0	0	0	0	aufregend

12.2. kompletter Versuchsplan:

1	A1	M1/- : I	M2/V2: II	M3/ZV: III
2	A1	M1/- : II	M2/V2: III	M3/ZV: I
3	A2	M1/- : III	M3/ZV: I	M2/V2: II
4	A2	M1/- : I	M3/ZV: III	M2/V2: II
5	A3	M2/V2: III	M1/- : II	M3/ZV: I
6	A3	M2/V2: II	M1/- : I	M3/ZV: III
7	A4	M2/V2: II	M3/ZV: I	M1/- : III
8	A4	M2/V2: I	M3/ZV: III	M1/- : II
9	A5	M3/ZV: III	M1/- : II	M2/V2: I
10	A5	M3/ZV: II	M1/- : III	M2/V2: I
11	A6	M3/ZV: III	M2/V2: I	M1/- : II
12	A6	M3/ZV: I	M2/V2: II	M1/- : III
13	B1	M1/ZV: III	M2/- : I	M3/V3: II
14	B1	M1/ZV: I	M2/- : II	M3/V3: III
15	B2	M1/ZV: II	M3/V3: III	M2/- : I
16	B2	M1/ZV: III	M3/V3: II	M2/- : I
17	B3	M2/- : II	M1/ZV: I	M3/V3: III
18	B3	M2/- : I	M1/ZV: III	M3/V3: II
19	B4	M2/- : I	M3/V3: II	M1/ZV: III
20	B4	M2/- : II	M3/V3: III	M1/ZV: I
21	B5	M3/V3: III	M1/ZV: I	M2/- : II
22	B5	M3/V3: I	M1/ZV: III	M2/- : II
23	B6	M3/V3: III	M2/- : II	M1/ZV: I
24	B6	M3/V3: II	M2/- : I	M1/ZV: III
25	C1	M1/V1: II	M2/ZV: I	M3/- : III
26	C1	M1/V1: I	M2/ZV: III	M3/- : II
27	C2	M1/V1: III	M3/- : II	M2/ZV: I
28	C2	M1/V1: II	M3/- : III	M2/ZV: I
29	C3	M2/ZV: III	M1/V1: I	M3/- : II
30	C3	M2/ZV: I	M1/V1: II	M3/- : III
31	C4	M2/ZV: III	M3/- : I	M1/V1: II
32	C4	M2/ZV: I	M3/- : II	M1/V1: III
33	C5	M3/- : II	M1/V1: III	M2/ZV: I
34	C5	M3/- : III	M1/V1: II	M2/ZV: I
35	C6	M3/- : II	M2/ZV: I	M1/V1: III
36	C6	M3/- : I	M2/ZV: III	M1/V1: II

12.3. Testprotokolle:



Vpn-Nr.	Vorname	Vl	Datum	Startzeit	Dauer	Ort	Bemerkungen
1	Roland C.	St	2.3.88	11:19	28'	Lampersham	Fest-Nervensch!
2	Duke	St.	9.3.88	19:49	34'	Trimbach	2. konnte die Stücke zieren!
3	Nigge	St	9.3.88	18:30	ca 60'	Trimbach	} Festzeit mit Gespräch!
4	Elvira	St	9.3.88	18:30	"	"	
5	Tobias	St	20.3.88	15:35	40'	Dullheim	
6	Inna	St	20.3.88	15:35	40'	"	
7	Christe	St	18.4.88	13:15	32'	Lampersham	Rosend. Mieder. Öl
8	Silvane	St.	18.4.88	13:15	33"	Lampersham	Mylene Pianos gehört Stimm
9	Marianne	Ch.	15.3.88	19:30	~ 45'	Balsthal	
10	Gabi	Ch.	15.3.88	19:30	~ 45'	Balsthal	Jody "schon mal gehört"
11	Guido	Ch.	14.3.88	16:15	~ 45'	Balsthal	
12	Stephan	Ch.	14.3.88	16:15	~ 45'	Balsthal	
13	Mathieu	Ch.	27.3.88	14:00	~ 40'	Balsthal	
14	Freddy	Ch	27.3.88	14:00	~ 40'	Balsthal	Mylene schon gehört
15	Yvonne	Fr.	19.4.88	10:35	~ 40'	Gesellschaftstr.	Mylene schon gehört
16	Beatrice	Fr	20.4.88	15:20	~ 30'	"	
17	Helen M.	St.	9.3.88	20:30	?	Trimbach	
18	Johannes M.	St.	9.3.88	20:30	?	"	

Vpn-Nr.	Vorname	Vl	Datum	Startzeit	Dauer	Ort	Bemerkungen
19	Wolfgang	Fv.	21.4.88	11.25	35'	Gesellschaftsh.	Hand-Midring-Gel-Öl
20	Georg	Fv.	21.4.88	11.25	35'	Gesellschaftsh.	
21	Dennis	St.	1.4.88	16.45	35'	Trimbach	
22	Nora	St.	1.4.88	16.45	35'	Trimbach	
23	Alwin	St.	1.1.88	17.25	35'	Trimbach	Hand-Midring-Öl
24	Bodo	St.	1.1.88	17.25	35'	Trimbach	Hand-Midring-Öl
25	Brigitte	Ch.	14.3.88	15.35	~ 35'	Balsthal	
26	Herbert	Ch.	14.3.88	15.35	~ 35'	Balsthal	
27	Rolf	Ch.	15.3.88	11.10	~ 45'	Balsthal	
28	Dolf	Ch.	15.3.88	11.10	~ 45'	Balsthal	
29	Roger	Ch.	27.3.88	15.10	~ 50'	Balsthal	
30	Mario	Ch.	27.3.88	15.10	~ 50'	Balsthal	
31	Michi	Ch.	27.3.88	16.15	~ 60'	Balsthal	
32	Susi	Ch.	27.3.88	16.15	~ 60'	Balsthal	
33	Barbara	St.	4.4.88	13.35	~ 40'	Trimbach	
34	Christina	St.	5.4.88	21.15	~ 35'	Trimbach	
35	Markus	St.	18.3.88	21:12	41'	Bern, Hydagey	
36	Karin	St.	18.3.88	21:12	41'	Bern, Hydagey	

12.4. Listing des PC-Programmes für die Datenerfassung  
und -auswertung:

```

1  program Vordip (input,output,Datenfile,Druckfile,Asciifile,AscFile);
2
3  type  Daten = array[1..46,1..3,1..3,1..12] of integer;
4         (Item,Musik,Beding.,Person)
5
6  var  Datenfile, Druckfile      : file of Daten;
7       Asciifile, AscFile      : file of char;
8       i,m,b,p                 : integer;
9       Data, Druckdata         : Daten;
10      Datei, Druckdatei, AscDatei : packed array[1..20] of char;
11      Zahl                     : integer;
12      Taste, Entscheidung,
13      Auswahl, andereAuswahl   : char;
14      ok                       : boolean;
15      Musik, Bedingung, Version : char;
16      Mu, Bed, Ver             : integer;
17      Vpn                      : integer;
18      Zaehler                  : integer;
19      Matrix                   : array[1..46] of integer;
20      Differenz, Item, Wert    : integer;
21      Kolonne                  : integer;
22
23      ($Imittel_s.pas)
24
25      ($Iintro.pas)
26
27      ($Iladen.pas)
28
29      ($Ispeichern.pas)
30
31      ($Ieintippe.pas)
32
33      ($Ianzeigen.pas)
34
35      ($Iauswerte.pas)
36
37      ($Idrucken.pas)
38
39      ($Iascii.pas)
40
41      ($Iinitial.pas)
42
43      ($Iinit2.pas)
44
45      ($Ihaeufigkeiten.pas)
46
47      ($Imenu.pas)
48
49
50
51
52  begin
53      intro;
54      repeat
55          menu;
56          until (Auswahl = '0');
57
58  end.
59
60
61

```

```

1  procedure Mittelw_Standardabw;
2
3  var
4  {          MITTELWERTE:          }
5
6      Mittelwert_Musiken: array[1..46,1..3] of real;
7  {          Item-Mittelwert (Über alle Personen und Bedingungen):      }
8  {          für alle Item/Musik-Kombinationen                          }
9
10     Mittelwert_Bedingungen: array[1..46,1..3] of real;
11 {          Item-Mittelwert (Über alle Personen und Musiken):          }
12 {          für alle Item/Bedingungs-Kombinationen                      }
13 {-----}
14
15
16 {          VARIANZEN:          }
17
18     Varianz_Musiken   : array[1..46,1..3] of real;
19 {          Item-Varianz (Über alle Personen und Bedingungen):        }
20 {          für alle Item/Musik-Kombinationen                          }
21
22     Varianz_Bedingungen: array[1..46,1..3] of real;
23 {          Item-Varianz (Über alle Personen und Musiken):            }
24 {          für alle Item/Bedingungs-Kombinationen                    }
25 {-----}
26
27 {          STANDARDABWEICHUNGEN:          }
28
29     Standardabw_Musiken   : array[1..46,1..3] of real;
30 {          Item-Standardabw. (Über alle Personen und Bedingungen):    }
31 {          für alle Item/Musik-Kombinationen                          }
32
33     Standardabw_Bedingungen: array[1..46,1..3] of real;
34 {          Item-Standardabw. (Über alle Personen und Musiken):        }
35 {          für alle Item/Bedingungs-Kombinationen                    }
36 {-----}
37
38
39     LF, CR, Leer, Komma: char;
40     ExpDatei           : string[20];
41     Ziffer             : string[7];
42     Nummer             : string[1];
43
44
45 begin
46     clrscr;
47     gotoxy(1,8);
48     writeln('Hier werden die Mittelwerte, Varianzen und Standarabweichungen');
49     writeln('der Faktorstufen ausgerechnet. Anschliessend werden ASCII-Files');
50     writeln('gebildet und auf Diskette abgespeichert!');
51     gotoxy(1,23);
52     writeln('          Weiter mit irgendeiner Taste, Abbruch mit Ctrl-C!');
53     read(kbd,Taste);
54     clrscr;
55
56     for m := 1 to 3 do begin
57         writeln;
58         writeln('Mittelwerte der Musik ',m,':');
59         for i := 1 to 46 do begin
60             Mittelwert_Musiken[i,m] := 0;
61             for p := 1 to 12 do begin
62                 for b := 1 to 3 do begin

```

```

63         Mittelwert_Musiken[i,m] :=
64             Mittelwert_Musiken[i,m] + Data[i,m,b,p];
65     end; {for b}
66 end; {for p}
67 Mittelwert_Musiken[i,m] := Mittelwert_Musiken[i,m]/36;
68 {   writeln(' Item Nr. ',i,' : ',Mittelwert_Musiken[i,m]);   }
69     end; {for i}
70 end; {for m}
71
72
73
74 writeln;
75
76
77
78     for b := 1 to 3 do begin
79         writeln;
80         writeln('Mittelwerte der Bedingung ',b,':');
81         for i := 1 to 46 do begin
82             Mittelwert_Bedingungen[i,b] := 0;
83             for p := 1 to 12 do begin
84                 for m := 1 to 3 do begin
85                     Mittelwert_Bedingungen[i,b] :=
86                         Mittelwert_Bedingungen[i,b] + Data[i,m,b,p];
87                 end; {for m}
88             end; {for p}
89             Mittelwert_Bedingungen[i,b] := Mittelwert_Bedingungen[i,b]/36;
90             {   writeln(' Item Nr. ',i,' : ',Mittelwert_Bedingungen[i,b]);   }
91         end; {for i}
92     end; {for b}
93
94
95
96 writeln;
97
98
99
100     for m := 1 to 3 do begin
101         writeln;
102         writeln('Varianzen der Musik ',m,':');
103         for i := 1 to 46 do begin
104             Varianz_Musiken[i,m] := 0;
105             for p := 1 to 12 do begin
106                 for b := 1 to 3 do begin
107                     Varianz_Musiken[i,m] := Varianz_Musiken[i,m] +
108                         ((Data[i,m,b,p] - Mittelwert_Musiken[i,m]) *
109                         (Data[i,m,b,p] - Mittelwert_Musiken[i,m]));
110                 end; {for b}
111             end; {for p}
112             Varianz_Musiken[i,m] := Varianz_Musiken[i,m]/36;
113             {   writeln(' Item Nr. ',i,' : ',Varianz_Musiken[i,m]);   }
114         end; {for i}
115     end; {for m}
116
117
118
119 writeln;
120
121
122
123     for b := 1 to 3 do begin
124         writeln;

```

```

125     writeln('Varianzen der Bedingung ',b,':');
126     for i := 1 to 46 do begin
127         Varianz_Bedingungen[i,b] := 0;
128         for p := 1 to 12 do begin
129             for m := 1 to 3 do begin
130                 Varianz_Bedingungen[i,b] := Varianz_Bedingungen[i,b] +
131                     ((Data[i,m,b,p] - Mittelwert_Bedingungen[i,b]) *
132                     (Data[i,m,b,p] - Mittelwert_Bedingungen[i,b]));
133             end; {for m}
134         end; {for p}
135         Varianz_Bedingungen[i,b] := Varianz_Bedingungen[i,b]/36;
136     {     writeln(' Item Nr. ',i,': ',Varianz_Bedingungen[i,b]);
137     end; {for i}
138     end; {for b}
139
140
141     writeln;
142
143     for m := 1 to 3 do begin
144         writeln;
145         writeln('Standardabweichungen der Musik ',m,':');
146         for i := 1 to 46 do begin
147             Standardabw_Musiken[i,m] := SORT(Varianz_Musiken[i,m]);
148     {     writeln(' Item Nr. ',i,': ',Standardabw_Musiken[i,m]);
149     end; {for i}
150     end; {for m}
151
152     writeln;
153
154     for b := 1 to 3 do begin
155         writeln;
156         writeln('Standardabweichungen der Bedingung ',b,':');
157         for i := 1 to 46 do begin
158             Standardabw_Bedingungen[i,b] := SORT(Varianz_Bedingungen[i,b]);
159     {     writeln(' Item Nr. ',i,': ',Standardabw_Bedingungen[i,b]);
160     end; {for i}
161     end; {for b}
162
163
164     writeln;
165     writeln(' Weiter mit irgendeiner Taste...');
166     read(kbd,Taste);
167
168     {-----}
169
170     {ASCII-FILES HERSTELLEN...}
171
172
173     clrscr;
174     Leer := ' ';
175     CR := chr(13);
176     LF := chr(10);
177     Komma:= chr(44);
178     gotoxy(1,4);
179     writeln(' ASCII-DATEIEN BILDEN UND SPEICHERN');
180
181
182     {     zuerst die Musiken:
183

```

```

184 for m := 1 to 3 do begin
185
186     {Teile A: Items 1 bis 23}
187
188     Str(m,Nummer);
189     ExpDatei := 'Mu_' + Nummer + '_a.asc';
190     assign(AscFile,ExpDatei);
191     rewrite(AscFile);
192     for i := 1 to 23 do begin
193         Str(Mittelwert_Musiken[i,m]:6:5,Ziffer);
194         write(AscFile,Ziffer[1]);
195         write(AscFile,Komma);
196         for Zaehler := 3 to 7 do begin
197             write(AscFile,Ziffer[Zaehler]);
198         end; {for Zaehler}
199
200         write(AscFile,Leer);
201         Str(Standardabw_Musiken[i,m]:6:5,Ziffer);
202         write(AscFile,Ziffer[1]);
203         write(AscFile,Komma);
204         for Zaehler := 3 to 7 do begin
205             write(AscFile,Ziffer[Zaehler]);
206         end; {for Zaehler}
207         write(AscFile,CR);
208         write(AscFile,LF);
209     end; {for i}
210     close(AscFile);
211
212     {Teile B: Items 24 bis 46}
213     ExpDatei := 'Mu_' + Nummer + '_b.asc';
214     assign(AscFile,ExpDatei);
215     rewrite(AscFile);
216     for i := 24 to 46 do begin
217         Str(Mittelwert_Musiken[i,m]:6:5,Ziffer);
218         write(AscFile,Ziffer[1]);
219         write(AscFile,Komma);
220         for Zaehler := 3 to 7 do begin
221             write(AscFile,Ziffer[Zaehler]);
222         end; {for Zaehler}
223
224         write(AscFile,Leer);
225         Str(Standardabw_Musiken[i,m]:6:5,Ziffer);
226         write(AscFile,Ziffer[1]);
227         write(AscFile,Komma);
228         for Zaehler := 3 to 7 do begin
229             write(AscFile,Ziffer[Zaehler]);
230         end; {for Zaehler}
231         write(AscFile,CR);
232         write(AscFile,LF);
233     end; {for i}
234     close(AscFile);
235
236 end; {for m}
237
238 (           Und nun die Bedingungen:           )
239

```



```

240   for b := 1 to 3 do begin
241
242       (Teile A: Items 1 bis 23)
243
244       Str(b,Nummer);
245       ExpDatei := 'Be_' + Nummer + '_a.asc';
246       assign(AscFile,ExpDatei);
247       rewrite(AscFile);
248       for i := 1 to 23 do begin
249           Str(Mittelwert_Bedingungen[i,b]:6:5,Ziffer);
250           write(AscFile,Ziffer[1]);
251           write(AscFile,Komma);
252           for Zaehler := 3 to 7 do begin
253               write(AscFile,Ziffer[Zaehler]);
254           end; {for Zaehler}
255
256           write(AscFile,Leer);
257           Str(Standardabw_Bedingungen[i,b]:6:5,Ziffer);
258           write(AscFile,Ziffer[1]);
259           write(AscFile,Komma);
260           for Zaehler := 3 to 7 do begin
261               write(AscFile,Ziffer[Zaehler]);
262           end; {for Zaehler}
263           write(AscFile,CR);
264           write(AscFile,LF);
265       end; {for i}
266       close(AscFile);
267
268       (Teile B: Items 24 bis 46)
269       ExpDatei := 'Be_' + Nummer + '_b.asc';
270       assign(AscFile,ExpDatei);
271       rewrite(AscFile);
272       for i := 24 to 46 do begin
273           Str(Mittelwert_Bedingungen[i,b]:6:5,Ziffer);
274           write(AscFile,Ziffer[1]);
275           write(AscFile,Komma);
276           for Zaehler := 3 to 7 do begin
277               write(AscFile,Ziffer[Zaehler]);
278           end; {for Zaehler}
279
280           write(AscFile,Leer);
281           Str(Standardabw_Bedingungen[i,b]:6:5,Ziffer);
282           write(AscFile,Ziffer[1]);
283           write(AscFile,Komma);
284           for Zaehler := 3 to 7 do begin
285               write(AscFile,Ziffer[Zaehler]);
286           end; {for Zaehler}
287           write(AscFile,CR);
288           write(AscFile,LF);
289       end; {for i}
290       close(AscFile);
291
292   end; {for b}
293
294
295
296   clrscr;
297   gotoxy(1,12);
298   writeln('    O.k., nun sind die Daten auf Diskette gespeichert!');
299   gotoxy(1,23);
300   writeln('                Weiter mit irgendeiner Taste...');
301   read(kbd,Taste);

```

302  
303  
304  
305

end; (procedure Mittelw\_Standardabw)

```

1  procedure intro;
2
3  begin
4      clrscr;
5      writeln;
6      writeln;
7      writeln;
8      writeln('          PROGRAMM ZUR STATISTISCHEN DATENVERARBEITUNG DER');
9      writeln('          VORDIPLOMARBEIT VON STEFAN MARTI, FRAENZI JEKER UND');
10     writeln('          CHRISTOPH ARN');
11     writeln('          =====');
12     writeln;
13     writeln('          Geschrieben von Stefan Marti, im April 88');
14     writeln('          Mehrere Updates bis zum Juni 89');
15     gotoxy(1,23);
16     writeln('          Weiter mit irgendeiner Taste...');
17     read(kbd,Taste);
18     end; {procedure intro}
19

```

```

1  procedure Laden;
2
3  begin
4      clrscr;
5      gotoxy(1,4);
6      writeln('    LADEN DER DATEN VON DISKETTE');
7      writeln('    -----');
8      gotoxy(1,8);
9      writeln('    Wie lautet der Name der Datei, in der die Daten gespeichert sind?');
10     write('                ');
11     readln(Datei);
12     assign(Datenfile,Datei);
13     {$I-} reset(Datenfile) {$I+};
14     ok := (ioresult =0);
15     if not ok then begin
16         writeln;
17         writeln;
18         writeln('    Die Datei mit dem Namen ',Datei);
19         writeln('    wurde nicht gefunden!');
20         writeln('    Entweder existiert sie noch nicht oder der Name wurde falsch geschrieben!');
21         writeln;
22         writeln('                Wollen Sie nochmal versuchen? (j/n)');
23         repeat
24             read(kbd,Entscheidung);
25             until (Entscheidung in ['j','J','n','N']);
26             if (Entscheidung in ['j','J']) then Laden;
27         end (if)
28     else begin
29         reset(Datenfile);
30         read(Datenfile,Data);
31         close(Datenfile);
32         writeln;
33         writeln;
34         writeln;
35         writeln;
36         writeln('    O.k., die Daten sind geladen!');
37         gotoxy(1,23);
38         writeln('                Weiter mit irgendeiner Taste...');
39         read(kbd,Taste);
40     end; {else}
41 end; {procedure Laden}
42

```

```

1  procedure Speichern;
2
3  begin
4      clrscr;
5      gotoxy(1,4);
6      writeln('   SPEICHERN DER DATEN AUF DISKETTE');
7      writeln('   -----');
8      gotoxy(1,8);
9      writeln('   Wie lautet der Name der Datei, in der die Daten');
10     writeln('   abgespeichert werden sollen?');
11     writeln('   EMPFEHLUNG: Nehmen Sie einen Namen mit Extension ".DAT"!');
12     write('           ');
13     readln(Datei);
14     assign(Datenfile,Datei);
15     {$I-} reset(Datenfile) {$I+};
16     ok := (ioresult = 0);
17     if ok then begin
18         writeln;
19         writeln('   Vorsicht! Diese Datei existiert schon. ');
20         writeln('   Beim Abspeichern werden die alten Daten überschrieben. ');
21         writeln('   Ist dies in Ordnung? (j für ja, mit irgendeiner anderen ');
22         writeln('   Taste verlassen Sie dieses Untermenü. ');
23         read(kbd,Entscheidung);
24     end {if}
25     else begin
26         writeln;
27         writeln('   Diese Datei existiert noch nicht. ');
28         writeln('   Beim abspeichern wird sie neu eröffnet. ');
29         writeln('   Ist dies in Ordnung? (j für ja, mit irgendeiner anderen ');
30         writeln('   Taste verlassen Sie dieses Untermenü. ');
31         read(kbd,Entscheidung);
32     end; {else}
33     if (Entscheidung in ['j','J']) then begin
34         rewrite(Datenfile);
35         write(Datenfile,Data);
36         writeln;
37         writeln('   O.k., nun sind die Daten auf Diskette gespeichert, und zwar ');
38         writeln('   unter dem Namen ',Datei);
39         gotoxy(1,23);
40         writeln('           Weiter mit irgendeiner Taste... ');
41         read(kbd,Taste);
42     end; {if}
43     close(Datenfile);
44 end; {procedure Speichern}
45

```

```

1  procedure Eintippen;
2
3  begin
4      repeat
5          clrscr;
6          writeln('    NEUE DATEN EINTIPPEN');
7          writeln('    -----');
8          gotoxy(1,4);
9          writeln('    Zur Beachtung: wenn sie die neuen Daten in einer Datei');
10         writeln('    abspeichern wollen, in der Sie früher schon einmal Daten');
11         writeln('    eingetippt und abgespeichert haben (die sie behalten möchten!),');
12         writeln('    dann müssen Sie zuerst diese Datei laden, bevor sie in');
13         writeln('    diesem Menu weitermachen!');
14         writeln('    w für weiter; zum Verlassen dieses Untermenus');
15         writeln('    irgendeine andere Taste! ');
16         read(kbd,Taste);
17         if (Taste = 'w') then exit;
18         clrscr;
19         writeln('    NEUE DATEN EINTIPPEN');
20         writeln('    -----');
21         gotoxy(1,4);
22         writeln('    Zuerst muss eingegeben werden, von welchem Datenblatt');
23         writeln('    die einzugebenden Daten stammen:');
24         writeln;
25         writeln('    Welche der Musik: M1 => 1');
26         writeln('                    M2 => 2');
27         writeln('                    M3 => 3');
28         write('                    Bitte 1, 2 oder 3 eintippen: ');
29         repeat
30             read(kbd,Musik);
31             until (Musik in ['1','2','3']);
32             write(Musik);
33             writeln;writeln;
34             writeln('    Welche Bedingung: - (d.h. kein Video) => 1');
35             writeln('                    V',Musik,' (d.h. Originalvideo) => 2');
36             writeln('                    ZV (d.h. Zufallsvideo) => 3');
37             write('                    Bitte 1, 2 oder 3 eintippen: ');
38             repeat
39                 read(kbd,Bedingung);
40                 until (Bedingung in ['1','2','3']);
41                 write(Bedingung);
42                 writeln;writeln;
43                 writeln('    Welche Fragebogen-Version: I => 1');
44                 writeln('                    II => 2');
45                 writeln('                    III => 3');
46                 write('                    Bitte wiederum 1, 2 oder 3 eintippen: ');
47                 repeat
48                     read(kbd,Version);
49                     until (Version in ['1','2','3']);
50                     write(Version);
51                     writeln; writeln;
52                     writeln('    Wie ist die Nummer der Versuchsperson?');
53                     write('    Bitte eine Zahl zwischen 1 und 36 eintippen und RETURN-Taste: ');
54                     repeat
55                         gotoxy(67,23);
56                         write(' ');
57                         gotoxy(67,23);
58                         readln(Vpn);

```

```

59     if      ((Musik = '1') AND (Bedingung = '1') OR
60             (Musik = '2') AND (Bedingung = '2') OR
61             (Musik = '3') AND (Bedingung = '3')) then Differenz := 0
62     else if ((Musik = '2') AND (Bedingung = '1') OR
63             (Musik = '3') AND (Bedingung = '2') OR
64             (Musik = '1') AND (Bedingung = '3')) then Differenz := 12
65             else                                     Differenz := 24;
66     if ((Vpn-Differenz < 1) OR (Vpn-Differenz > 12)) then begin
67         gotoxy(71,23);
68         write('UNMOEGlich');
69         sound(800);
70         delay(80);
71         nosound;
72         delay(1000);
73     end;
74     until ((Vpn <= 36) AND (Vpn-Differenz >= 1) AND (Vpn-Differenz <= 12));
75     clrscr;
76     writeln('    NEUE DATEN EINTIPPEN');
77     writeln('    -----');
78     gotoxy(1,8);
79     writeln('    Die Daten, die im folgenden eingetippt werden, stammen von');
80     writeln('    folgendem Fragebogen:');
81     writeln;
82     write('          M',Musik,' / ');
83     case Bedingung of
84         '1': write('-');
85         '2': write('V',Musik);
86         '3': write('ZV');
87     end; {case}
88     write(', Version ');
89     case Version of
90         '1': write('I');
91         '2': write('II');
92         '3': write('III');
93     end; {case}
94     write(', Versuchsperson Nr. ',Vpn);
95     gotoxy(1,15);
96     write('    Ist dies in Ordnung? (j/n) ');
97     repeat
98         read(kbd,Entscheidung);
99     until (Entscheidung in ['j','n','N','J']);
100    until (Entscheidung in ['j','J']);
101    repeat
102        clrscr;
103        writeln('    EINLESEN VON DATEN:');
104        writeln;
105        writeln('    Jetzt müssen die 46 Werte eingegeben werden. ');
106        writeln('    Immer eine Zahl zwischen 1 und 6 eingeben! ');
107        writeln('    Für vorzeitigen Abbruch der Eingabeserie "0" eintippen! ');
108        writeln('    Für einen Wert zurück: "9" eintippen! ');
109        writeln;
110        Zaehler := 0;
111        repeat
112            Zaehler := Zaehler+1;
113            if (Zaehler > 1) then begin
114                if (Matrix[Zaehler-1] = 9) then Zaehler := Zaehler -2;
115            end;

```

```

116     repeat
117         write(' ',Zaehler,'. Wert: BISHER:',Matrix[Zaehler],' ; NEU: ');
118         readln(Matrix[Zaehler]);
119         if (Matrix[Zaehler] > 6) AND (Matrix[Zaehler] <> 9) then begin
120             writeln('      EINGABE UNGÜLTIG. ZAHL ZWISCHEN 1 UND 6 EINTIPPEN!!');
121             sound(800);
122             delay(80);
123             nosound;
124             end; {if}
125         until (Matrix[Zaehler] (= 6) OR (Matrix[Zaehler] =9));
126     until (Zaehler >= 46) OR (Matrix[Zaehler] = 0);
127     if (Matrix[Zaehler] = 0) then begin
128         writeln;
129         writeln('      Sie haben die Eingabesequenz abgebrochen. ');
130         writeln('      Wollen Sie die Eingabe der Daten der Versuchs- ');
131         writeln('      person Nr. ',Vpn,' noch einmal von vorne beginnen? ');
132         writeln('      (j für noch einmal beginnen und irgendeine andere Taste ');
133         write('      zum Verlassen des Untermenü NEUE DATEN EINTIPPEN): ');
134         read(kbd,Entscheidung);
135     end {if}
136     else begin
137         Entscheidung := 'n';
138         writeln;
139         writeln;
140         writeln('      O.k.! Die 46 Daten sind eingegeben! ');
141         writeln;
142         writeln('      Wollen Sie die eingegeben Daten nochmals anschauen? ');
143         writeln('      (Ich empfehle Ihnen JA: vergleichen Sie nochmals mit ');
144         write('      den Original-Daten!) j/n: ');
145         read(Taste);
146         if (Taste = 'j') then begin
147             clrscr;
148             writeln;
149             write('      M',Musik,' / ');
150             case Bedingung of
151                 '1': write('-');
152                 '2': write('V',Musik);
153                 '3': write('ZV');
154             end; {case}
155             write(', Version ');
156             case Version of
157                 '1': write('I');
158                 '2': write('II');
159                 '3': write('III');
160             end; {case}
161             writeln(', Versuchsperson Nr. ',Vpn);
162             writeln('      =====');
163             for Zaehler := 1 to 46 do begin
164                 if (Zaehler (= 16) then Kolonne := 2
165                 else if (Zaehler (= 32) then Kolonne := 30
166                     else Kolonne := 62;
167                 gotoxy(Kolonne,((Zaehler-1) mod 16)+6);
168                 write('Item Nr. ',Zaehler);
169                 if (Zaehler (= 9) then write(' ');
170                 write(': ',Matrix[Zaehler]);
171             end; {for}
172         end; {if}
173         gotoxy(1,24);
174         writeln('      Weiter mit irgendeiner Taste... ');
175         read(kbd,Taste);
176     end; {else}
177 until (Entscheidung <> 'j');

```



```

178
179
180 if (Entscheidung = 'n') then begin
181     case Musik of (Datenkonversion)
182         '1': Mu := 1;
183         '2': Mu := 2;
184         '3': Mu := 3;
185     end;
186     case Bedingung of
187         '1': Bed := 1;
188         '2': Bed := 2;
189         '3': Bed := 3;
190     end;
191
192
193
194 for Zaehler := 1 to 46 do begin
195     if ((Mu = 1) AND (Bed = 1) OR
196         (Mu = 2) AND (Bed = 2) OR
197         (Mu = 3) AND (Bed = 3)) then Differenz := 0
198     else if ((Mu = 2) AND (Bed = 1) OR
199             (Mu = 3) AND (Bed = 2) OR
200             (Mu = 1) AND (Bed = 3)) then Differenz := 12
201     else Differenz := 24;
202
203
204
205 if (Version = '1') then
206     Data[Zaehler,Mu,Bed,Vpn-Differenz] := Matrix[Zaehler];
207 if (Version = '2') then begin
208     case Zaehler of
209         1: begin Item:= 5; Wert:= Matrix[1]; end;
210         2: begin Item:= 23; Wert:= 7-Matrix[2]; end;
211         3: begin Item:= 44; Wert:= Matrix[3]; end;
212         4: begin Item:= 7; Wert:= Matrix[4]; end;
213         5: begin Item:= 45; Wert:= 7-Matrix[5]; end;
214         6: begin Item:= 11; Wert:= Matrix[6]; end;
215         7: begin Item:= 34; Wert:= 7-Matrix[7]; end;
216         8: begin Item:= 30; Wert:= 7-Matrix[8]; end;
217         9: begin Item:= 35; Wert:= Matrix[9]; end;
218         10: begin Item:= 24; Wert:= Matrix[10]; end;
219         11: begin Item:= 17; Wert:= Matrix[11]; end;
220         12: begin Item:= 39; Wert:= 7-Matrix[12]; end;
221         13: begin Item:= 26; Wert:= Matrix[13]; end;
222         14: begin Item:= 42; Wert:= Matrix[14]; end;
223         15: begin Item:= 29; Wert:= 7-Matrix[15]; end;
224         16: begin Item:= 10; Wert:= 7-Matrix[16]; end;
225         17: begin Item:= 22; Wert:= Matrix[17]; end;
226         18: begin Item:= 6; Wert:= Matrix[18]; end;
227         19: begin Item:= 32; Wert:= 7-Matrix[19]; end;
228         20: begin Item:= 41; Wert:= 7-Matrix[20]; end;
229         21: begin Item:= 8; Wert:= Matrix[21]; end;
230         22: begin Item:= 2; Wert:= 7-Matrix[22]; end;
231         23: begin Item:= 21; Wert:= Matrix[23]; end;
232         24: begin Item:= 36; Wert:= 7-Matrix[24]; end;
233         25: begin Item:= 9; Wert:= Matrix[25]; end;
234         26: begin Item:= 18; Wert:= Matrix[26]; end;
235         27: begin Item:= 1; Wert:= 7-Matrix[27]; end;
236         28: begin Item:= 16; Wert:= Matrix[28]; end;
237         29: begin Item:= 46; Wert:= 7-Matrix[29]; end;
238         30: begin Item:= 14; Wert:= Matrix[30]; end;
239         31: begin Item:= 27; Wert:= 7-Matrix[31]; end;

```

```

240      32: begin Item:= 3; Wert:= 7-Matrix[32]; end;
241      33: begin Item:= 38; Wert:= Matrix[33]; end;
242      34: begin Item:= 4; Wert:= 7-Matrix[34]; end;
243      35: begin Item:= 15; Wert:= 7-Matrix[35]; end;
244      36: begin Item:= 20; Wert:= 7-Matrix[36]; end;
245      37: begin Item:= 13; Wert:= Matrix[37]; end;
246      38: begin Item:= 37; Wert:= 7-Matrix[38]; end;
247      39: begin Item:= 19; Wert:= 7-Matrix[39]; end;
248      40: begin Item:= 12; Wert:= Matrix[40]; end;
249      41: begin Item:= 31; Wert:= 7-Matrix[41]; end;
250      42: begin Item:= 33; Wert:= 7-Matrix[42]; end;
251      43: begin Item:= 25; Wert:= Matrix[43]; end;
252      44: begin Item:= 28; Wert:= Matrix[44]; end;
253      45: begin Item:= 43; Wert:= Matrix[45]; end;
254      46: begin Item:= 40; Wert:= 7-Matrix[46]; end;
255      end; {case Nr.2}
256      Data[Item,Mu,Bed,Vpn-Differenz] := Wert;
257  end; {if Nr.2}
258  if (Version = '3') then begin
259      case Zaehler of
260          1: begin Item:= 34; Wert:= 7-Matrix[1]; end;
261          2: begin Item:= 24; Wert:= 7-Matrix[2]; end;
262          3: begin Item:= 6; Wert:= 7-Matrix[3]; end;
263          4: begin Item:= 33; Wert:= 7-Matrix[4]; end;
264          5: begin Item:= 16; Wert:= 7-Matrix[5]; end;
265          6: begin Item:= 9; Wert:= 7-Matrix[6]; end;
266          7: begin Item:= 5; Wert:= 7-Matrix[7]; end;
267          8: begin Item:= 15; Wert:= Matrix[8]; end;
268          9: begin Item:= 28; Wert:= 7-Matrix[9]; end;
269          10: begin Item:= 40; Wert:= 7-Matrix[10]; end;
270          11: begin Item:= 31; Wert:= 7-Matrix[11]; end;
271          12: begin Item:= 27; Wert:= Matrix[12]; end;
272          13: begin Item:= 13; Wert:= 7-Matrix[13]; end;
273          14: begin Item:= 1; Wert:= 7-Matrix[14]; end;
274          15: begin Item:= 46; Wert:= Matrix[15]; end;
275          16: begin Item:= 26; Wert:= 7-Matrix[16]; end;
276          17: begin Item:= 18; Wert:= 7-Matrix[17]; end;
277          18: begin Item:= 11; Wert:= 7-Matrix[18]; end;
278          19: begin Item:= 23; Wert:= 7-Matrix[19]; end;
279          20: begin Item:= 14; Wert:= 7-Matrix[20]; end;
280          21: begin Item:= 21; Wert:= 7-Matrix[21]; end;
281          22: begin Item:= 10; Wert:= Matrix[22]; end;
282          23: begin Item:= 7; Wert:= 7-Matrix[23]; end;
283          24: begin Item:= 25; Wert:= 7-Matrix[24]; end;
284          25: begin Item:= 3; Wert:= 7-Matrix[25]; end;
285          26: begin Item:= 42; Wert:= 7-Matrix[26]; end;
286          27: begin Item:= 44; Wert:= 7-Matrix[27]; end;
287          28: begin Item:= 20; Wert:= Matrix[28]; end;
288          29: begin Item:= 38; Wert:= 7-Matrix[29]; end;
289          30: begin Item:= 37; Wert:= 7-Matrix[30]; end;
290          31: begin Item:= 45; Wert:= 7-Matrix[31]; end;
291          32: begin Item:= 43; Wert:= 7-Matrix[32]; end;
292          33: begin Item:= 36; Wert:= Matrix[33]; end;
293          34: begin Item:= 4; Wert:= Matrix[34]; end;
294          35: begin Item:= 8; Wert:= 7-Matrix[35]; end;
295          36: begin Item:= 19; Wert:= 7-Matrix[36]; end;
296          37: begin Item:= 12; Wert:= 7-Matrix[37]; end;
297          38: begin Item:= 32; Wert:= Matrix[38]; end;
298          39: begin Item:= 22; Wert:= 7-Matrix[39]; end;
299          40: begin Item:= 2; Wert:= Matrix[40]; end;
300          41: begin Item:= 39; Wert:= Matrix[41]; end;
301          42: begin Item:= 30; Wert:= Matrix[42]; end;

```

```
302         43: begin Item:= 35; Wert:= 7-Matrix[43]; end;
303         44: begin Item:= 17; Wert:= 7-Matrix[44]; end;
304         45: begin Item:= 29; Wert:= 7-Matrix[45]; end;
305         46: begin Item:= 41; Wert:= Matrix[46]; end;
306     end; {case Nr.2}
307     Data[Item,Mu,Bed,Vpn-Differenz] := Wert;
308 end; {if Nr.3}
309 end; {46-er for-Schlaufe}
310 end; {if Entscheidung = 'n'}
311 end; {procedure Eintippen}
312
```

```

1  procedure Anzeigen;
2
3  procedure It_Anz;
4  begin
5  end; {procedure It_Anz}
6
7  procedure Bed_Anz;
8  begin
9  end; {procedure Bed_Anz}
10
11 procedure Mus_Anz;
12 begin
13 end; {procedure Mus_Anz}
14
15 procedure Pers_Anz;
16 begin
17     clrscr;
18     writeln;
19     writeln('    Die Daten welcher Person sollen angezeigt werden?');
20     write('        (Zahl zwischen 1 und 36 tippen!): ');
21     repeat
22         gotoxy(35,5);
23         write(' ');
24         gotoxy(35,5);
25         read(Vpn);
26     until ((Vpn >= 1) AND (Vpn <= 36));
27     writeln;
28     writeln('        Achtung! Die Daten werden in der (Standard-)Form der');
29     writeln('        Fragebogen-Version I dargestellt!');
30     gotoxy(3,9);
31
32
33
34
35
36     read(kbd,Taste);
37 end; {procedure Pers_Anz}
38
39 procedure All_Anz;
40 begin
41     clrscr;
42     writeln('    Die Datei, die angezeigt wird, heisst: ',Datei);
43     writeln('    Um den Bildschirm anzuhalten, (CTRL + S) drücken!');
44     writeln('    Weiter mit irgendeiner Taste...');
45     read(kbd,Taste);
46     gotoxy(40,5); write('ITEM NUMMER:');
47     gotoxy(27,7); write('    1 1111111112 2222222223 3333333334 444444');
48     gotoxy(27,8); write('1234567890 1234567890 1234567890 1234567890 123456');
49     gotoxy(1,8); write('MUSIK BEDINGUNG VPN');
50     gotoxy(1,10);
51     for m := 1 to 3 do begin
52         for b := 1 to 3 do begin
53             for p := 1 to 12 do begin
54                 write(' ',m,' ',b,' ',p);
55                 if (p<10 ) then write(' ');
56                 write(' ');
57                 for i:= 1 to 46 do begin
58                     write(Data[i,m,b,p]);
59                     if (i=10) OR (i=20) OR (i=30) OR (i=40) then write(' ');
60                 end;
61                 writeln;
62             end; {for p}

```

```

63         end; {for b}
64     end; {for m}
65     writeln;
66     writeln('        Weiter mit irgendeiner Taste...');
67     read(kbd,Taste);
68     end; {procedure All_Anz}
69
70 begin
71     clrscr;
72     gotoxy(1,4);
73     writeln('  DATEN ANZEIGEN');
74     writeln('  -----');
75     gotoxy(1,8);
76     writeln('  1. Daten eines Items (1 bis 46)');
77     writeln('  2. Daten einer Bedingung (1 bis 3)');
78     writeln('  3. Daten einer Musik (1 bis 3)');
79     writeln('  4. Daten einer Person (1 bis 36)');
80     writeln('  5. alle Daten');
81     writeln;
82     writeln('  6. Abbruch');
83     gotoxy(1,20);
84     writeln('  Bitte tippen Sie die Nummer des Gewünschten (Zahl zwischen 1 und 6):');
85     write('                ');
86     repeat
87         read(kbd,Auswahl);
88     until (Auswahl in ['1','2','3','4','5','6']);
89     case Auswahl of
90         '1': It_Anz;
91         '2': Bed_Anz;
92         '3': Mus_Anz;
93         '4': Pers_Anz;
94         '5': All_Anz;
95         '6':;
96     end; {case}
97 end; {Anzeigen}
98

```

```

1  procedure Auswerten;
2
3  var
4      X_M : array[1..46,1..3,1..3] of real;
5      {
6          Item-Mittelwert (Über alle Personen):
7          }
8      {
9          für alle Item/Musik/Bedingungs-Kombinationen
10         }
11
12     X_SM : array[1..3,1..3] of real;
13     {
14         Skalen-Mittelwert (Über alle Personen und Items):
15         }
16     {
17         für alle Musik/Bedingungs-Kombinationen
18         }
19
20     X_I : array[1..46,1..3,1..3] of real;
21     {
22         transformierter Item-Mittelwert (Über alle Personen):
23         }
24     {
25         für alle Item/Musik/Bedingungs-Kombinationen
26         }
27     {
28         (gewonnen aus dem SKALEN-Mittelwert X_SM)
29         }
30
31     O_MBMB: array[1..3,1..3,1..3,1..3] of real;
32     {
33         O-Werte (Über alle Items und Personen):
34         }
35     {
36         für alle Musik/Bedingungs-Kombinations-PAARE
37         }
38
39     (-----)
40
41     X_BM : array[1..3] of real;
42     {
43         Bedingungs-Mittelwert (Über alle Personen, Items und Musiken):
44         }
45     {
46         für alle Bedingungen
47         }
48
49     X_MM : array[1..3] of real;
50     {
51         Musiken-Mittelwert (Über alle Personen, Items und Bedingungen):
52         }
53     {
54         für alle Musiken
55         }
56
57     X_I_B: array[1..46,1..3,1..3] of real;
58     {
59         transformierter Item-Mittelwert (Über alle Personen):
60         }
61     {
62         für alle Musik/Bedingungs/Item-Kombinationen
63         }
64     {
65         (gewonnen aus dem BEDINGUNGS-Mittelwert X_BM)
66         }
67
68     X_I_M: array[1..46,1..3,1..3] of real;
69     {
70         transformierter Item-Mittelwert (Über alle Personen):
71         }
72     {
73         für alle Musik/Bedingungs/Item-Kombinationen
74         }
75     {
76         (gewonnen aus dem MUSIKEN-Mittelwert X_MM)
77         }
78
79     O_BB : array[1..3,1..3] of real;
80     {
81         O-Werte (Über alle Items, Personen und Musiken):
82         }
83     {
84         für alle Bedingungs-PAARE
85         }
86
87     O_MM : array[1..3,1..3] of real;
88     {
89         O-Werte (Über alle Items, Personen und Bedingungen):
90         }
91     {
92         für alle Musiken-PAARE
93         }
94
95     (-----)
96
97     V_Item : array[1..46,1..3,1..3] of real;
98     {
99         Item-Varianz (Über alle Personen):
100    }
101    {
102        für alle Item/Bedingungs/Musik-Kombinationen
103    }
104
105    V_Skala : array[1..3,1..3] of real;
106    {
107        Varianz eines Fragebogens (Über alle Personen und Items):
108    }
109    {
110        für alle Musik/Bedingungs-Kombinationen
111    }
112
113    V_Bedingung : array[1..3] of real;
114    {
115        Varianz einer Bedingung (Über alle Personen, Items und Musiken)
116    }
117    {
118        für alle Bedingungen
119    }
120

```

```

63     V_Musiken : array[1..3] of real;
64     {
65         Varianz einer Musik (Über alle Personen, Items und Bedingungen)
66         für alle Musiken
67     }
68
69     STANDARDABWEICHUNGEN:
70
71     S_Skala : array[1..3,1..3] of real;
72     S_Bedingung: array[1..3] of real;
73     S_Musiken: array[1..3] of real;
74
75     {-----}
76     F1_MB : array[1..3,1..3] of real;
77     {
78         Faktorscore des Faktors AKTIVITAET eines Fragebogens
79         (Über alle Personen und die 6 spezifisch ladenden Items):
80         für alle Musik/Bedingungs-Kombinationen
81     }
82
83     F2_MB : array[1..3,1..3] of real;
84     {
85         Faktorscore des Faktors POTENZ eines Fragebogens
86         (Über alle Personen und die 6 spezifisch ladenden Items):
87         für alle Musik/Bedingungs-Kombinationen
88     }
89
90     F3_MB : array[1..3,1..3] of real;
91     {
92         Faktorscore des Faktors HARMONIE/ORDUNG eines Fragebogens
93         (Über alle Personen und die 6 spezifisch ladenden Items):
94         für alle Musik/Bedingungs-Kombinationen
95     }
96
97     D_MBBB: array[1..3,1..3,1..3,1..3] of real;
98     {
99         D-Werte (Über die drei mal 6 spezifisch ladenden Items und
100        alle Personen):
101        für alle Musik/Bedingungs-Kombinations-PAARE
102    }
103
104     {-----}
105     D_MM : array[1..3,1..3] of real;
106     {
107         D-Werte (Über die drei mal 6 spezifisch ladenden Items,
108         Personen und Bedingungen):
109         für alle Musiken-PAARE
110     }
111
112     D_BB : array[1..3,1..3] of real;
113     {
114         D-Werte (Über die drei mal 6 spezifisch ladenden Items,
115         Personen und Musiken):
116         für alle Bedingungs-PAARE
117     }
118
119     {-----}
120
121     HILFSVARIABLEN:
122
123     Feld : array[1..9] of record
124         Mus: integer;
125         Bed: integer;
126         end;
127     Zaehler1, Zaehler2, Zaehler3 : integer;
128     hilf1, hilf2, hilf3 : real;

```

```

124 procedure Q_Werte;
125
126 var Entsch1, Entsch2, Entsch3, Entsch4, Entsch5, Entsch6,
127     Entsch7, Entsch8, Entsch9, Entsch10, Entsch0 :char;
128
129 begin
130     clrscr;
131     writeln(' Was soll alles ausgedruckt werden? jeweils mit j/n antworten!');
132     writeln;
133     writeln('     Item-Mittelwert (Über alle Personen:');
134     write('         für alle Item/Musik/Bedingungs-Kombinationen: ');
135     readln(Entsch1);
136     writeln;
137     writeln('     Skalen-Mittelwert (Über alle Personen und Items:');
138     write('         für alle Musik/Bedingungs-Kombinationen: ');
139     readln(Entsch2);
140     writeln;
141     writeln('     transformierter Item-Mittelwert (Über alle Personen:');
142     write('         für alle Item/Musik/Bedingungs-Kombinationen: ');
143     readln(Entsch3);
144     writeln;
145     writeln('     Q-Werte (Über alle Items und Personen:');
146     write('         für alle Musik/Bedingungs-Kombinations-PAARE: ');
147     readln(Entsch4);
148     writeln;
149     writeln('     Bedingungs-Mittelwert (Über alle Personen, Items und Musiken:');
150     write('         für alle Bedingungen: ');
151     readln(Entsch5);
152     writeln;
153     writeln('     Musiken-Mittelwert (Über alle Personen, Items und Bedingungen:');
154     write('         für alle Musiken: ');
155     readln(Entsch6);
156     writeln;
157     writeln('     transformierter Item-Mittelwert (Über alle Personen:');
158     writeln('         für alle Musik/Bedingungs/Item-Kombinationen');
159     write('         (gewonnen aus dem BEDINGUNGS-Mittelwert X_BM): ');
160     readln(Entsch7);
161     writeln;
162     writeln('     transformierter Item-Mittelwert (Über alle Personen:');
163     writeln('         für alle Musik/Bedingungs/Item-Kombinationen');
164     write('         (gewonnen aus dem MUSIKEN-Mittelwert X_MM): ');
165     readln(Entsch8);
166     writeln;
167     writeln('     Q-Werte (Über alle Items, Personen und Musiken:');
168     write('         für alle Bedingungs-PAARE: ');
169     readln(Entsch9);
170     writeln;
171     writeln('     Q-Werte (Über alle Items, Personen und Bedingungen:');
172     write('         für alle Musiken-PAARE: ');
173     readln(Entsch10);
174
175     clrscr;
176     gotoxy(1,10);
177     writeln('     Soll das alles auch auf dem Printer gedruckt werden?');
178     write('         j/n: ');
179     readln(Entsch0);
180
181
182     clrscr;
183     gotoxy(10,12);
184     writeln('Bitte warten, ich rechne...');
185

```



```

186      writeln;
187      for i := 1 to 46 do begin
188          for m := 1 to 3 do begin
189              for b := 1 to 3 do begin
190                  X_M[i,m,b] := 0;
191                  for p := 1 to 12 do begin
192                      X_M[i,m,b] := X_M[i,m,b] + Data[i,m,b,p];
193                  end; {for p}
194                  X_M[i,m,b] := X_M[i,m,b]/12;
195                  if (Entsch1='j') then
196                      if (Entsch0 () 'j') then
197                          writeln('X_M (' ,i ,',',m ,',',b ,') : ',X_M[i,m,b])
198                      else writeln(1st,'X_M (' ,i ,',',m ,',',b ,') : ',X_M[i,m,b]);
199                  end; {for b}
200              end; {for m}
201          end; {for i}
202
203      writeln;
204
205      for m := 1 to 3 do begin
206          for b := 1 to 3 do begin
207              X_SM[m,b] := 0;
208              for i := 1 to 46 do begin
209                  for p := 1 to 12 do begin
210                      X_SM[m,b] := X_SM[m,b] + Data[i,m,b,p];
211                  end; {for p}
212              end; {for i}
213              X_SM[m,b] := X_SM[m,b]/(46*12);
214              if (Entsch2='j') then
215                  if (Entsch0 () 'j') then
216                      writeln('X_SM (' ,m ,',',b ,') : ',X_SM[m,b])
217                  else writeln(1st,'X_SM (' ,m ,',',b ,') : ',X_SM[m,b]);
218              end; {for b}
219          end; {for m}
220
221      writeln;
222
223      for i := 1 to 46 do begin
224          for m := 1 to 3 do begin
225              for b := 1 to 3 do begin
226                  X_T[i,m,b] := X_M[i,m,b] - X_SM[m,b];
227                  if (Entsch3='j') then
228                      if (Entsch0 () 'j') then
229                          writeln('X_T (' ,i ,',',m ,',',b ,') : ',X_T[i,m,b])
230                      else writeln(1st,'X_T (' ,i ,',',m ,',',b ,') : ',X_T[i,m,b]);
231                  end; {for b}
232              end; {for m}
233          end; {for i}
234
235      writeln;
236
237      Zaehler3 := 0;
238      for m := 1 to 3 do begin
239          for b := 1 to 3 do begin
240              Zaehler3 := Zaehler3 + 1;
241              Feld[Zaehler3].Mus := m;
242              Feld[Zaehler3].Bed := b;
243          end; {for b}
244      end; {for m}
245

```

```

246   for Zaehler1 := 2 to 9 do begin
247     for Zaehler2 := 1 to (Zaehler1 - 1) do begin
248       hilf1 := 0;
249       hilf2 := 0;
250       hilf3 := 0;
251       for i := 1 to 46 do begin
252         hilf1 := hilf1 + (X_T[i,Feld[Zaehler1].Mus,Feld[Zaehler1].Bed] *
253           X_T[i,Feld[Zaehler2].Mus,Feld[Zaehler2].Bed] );
254         hilf2 := hilf2 + (X_T[i,Feld[Zaehler1].Mus,Feld[Zaehler1].Bed] *
255           X_T[i,Feld[Zaehler1].Mus,Feld[Zaehler1].Bed] );
256         hilf3 := hilf3 + (X_T[i,Feld[Zaehler2].Mus,Feld[Zaehler2].Bed] *
257           X_T[i,Feld[Zaehler2].Mus,Feld[Zaehler2].Bed] );
258       end; {for i}
259
260       {writeln('Zaehler1: ',Zaehler1); }
261       {writeln('Zaehler2: ',Zaehler2); }
262       {writeln(' hilf1: ',hilf1); }
263       {writeln(' hilf2: ',hilf2); }
264       {writeln(' hilf3: ',hilf3); }
265       Q_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
266         Feld[Zaehler2].Mus,Feld[Zaehler2].Bed] := hilf1/(SQRT(hilf2 * hilf3));
267       if (Entsch4='j') then
268         if (Entsch0 () 'j') then begin
269           write('Q-Wert von Mus.',Feld[Zaehler1].Mus,'/ Bed.',Feld[Zaehler1].Bed);
270           write(' mit Mus.',Feld[Zaehler2].Mus,'/ Bed.',Feld[Zaehler2].Bed);
271           writeln(' : ',Q_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
272             Feld[Zaehler2].Mus,Feld[Zaehler2].Bed]);
273         end
274         else begin
275           write(1st,'Q-Wert von Mus.',Feld[Zaehler1].Mus,'/ Bed.',Feld[Zaehler1].Bed);
276           write(1st,' mit Mus.',Feld[Zaehler2].Mus,'/ Bed.',Feld[Zaehler2].Bed);
277           writeln(1st,' : ',Q_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
278             Feld[Zaehler2].Mus,Feld[Zaehler2].Bed]);
279         end;
280       end; {for Zaehler2}
281     end; {for Zaehler1}
282
283   writeln;
284
285   for b := 1 to 3 do begin
286     X_BM[b] := 0;
287     for i := 1 to 46 do begin
288       for m := 1 to 3 do begin
289         for p := 1 to 12 do begin
290           X_BM[b] := X_BM[b] + Data[i,m,b,p];
291         end; {for p}
292       end; {for m}
293     end; {for i}
294     X_BM[b] := X_BM[b]/(46*3*12);
295     if (Entsch5='j') then
296       if (Entsch0 () 'j') then
297         writeln('X_BM (' ,b,') : ',X_BM[b])
298       else writeln(1st,'X_BM (' ,b,') : ',X_BM[b]);
299     end; {for b}
300
301   writeln;
302
303   for m := 1 to 3 do begin
304     X_MM[m] := 0;
305     for i := 1 to 46 do begin
306       for b := 1 to 3 do begin
307         for p := 1 to 12 do begin

```

```

308         X_MM[m] := X_MM[m] + Data[i,m,b,p];
309     end; {for p}
310 end; {for b}
311 end; {for i}
312 X_MM[m] := X_MM[m]/(46*3*12);
313 if (Entsch6='j') then
314     if (Entsch0 (>'j')) then
315         writeln('X_MM (' ,m,') : ',X_MM[m])
316     else writeln(1st,'X_MM (' ,m,') : ',X_MM[m]);
317 end; {for m}
318
319 writeln;
320
321 for i := 1 to 46 do begin
322     for m := 1 to 3 do begin
323         for b := 1 to 3 do begin
324             X_I_B[i,m,b] := X_M[i,m,b] - X_BM[b];
325             if (Entsch7='j') then
326                 if (Entsch0 (>'j')) then
327                     writeln('X_I_B (' ,i,',' ,m,',' ,b,') : ',X_I_B[i,m,b])
328                 else writeln(1st,'X_I_B (' ,i,',' ,m,',' ,b,') : ',X_I_B[i,m,b]);
329             end; {for b}
330         end; {for m}
331     end; {for i}
332
333     writeln;
334
335     for i := 1 to 46 do begin
336         for m := 1 to 3 do begin
337             for b := 1 to 3 do begin
338                 X_I_M[i,m,b] := X_M[i,m,b] - X_MM[m];
339                 if (Entsch8='j') then
340                     if (Entsch0 (>'j')) then
341                         writeln('X_I_M (' ,i,',' ,m,',' ,b,') : ',X_I_M[i,m,b])
342                     else writeln(1st,'X_I_M (' ,i,',' ,m,',' ,b,') : ',X_I_M[i,m,b]);
343                 end; {for b}
344             end; {for m}
345         end; {for i}
346
347         writeln;
348
349         for Zaehler1 := 2 to 3 do begin
350             for Zaehler2 := 1 to (Zaehler1-1) do begin
351                 hilf1 := 0;
352                 hilf2 := 0;
353                 hilf3 := 0;
354                 for m := 1 to 3 do begin
355                     for i := 1 to 46 do begin
356                         hilf1 := hilf1 + (X_I_B[i,m,Zaehler1] * X_I_B[i,m,Zaehler2]);
357                         hilf2 := hilf2 + (X_I_B[i,m,Zaehler1] * X_I_B[i,m,Zaehler1]);
358                         hilf3 := hilf3 + (X_I_B[i,m,Zaehler2] * X_I_B[i,m,Zaehler2]);
359                     end; {for i}
360                 end; {for m}
361                 Q_BB[Zaehler1,Zaehler2] := hilf1/(SQRT(hilf2 * hilf3));
362                 if (Entsch9='j') then
363                     if (Entsch0 (>'j')) then begin
364                         write('Q-Wert von Bedingung ',Zaehler1,' mit Bedingung ',Zaehler2);
365                         writeln(' : ',Q_BB[Zaehler1,Zaehler2]);
366                     end
367                     else begin
368                         write(1st,'Q-Wert von Bedingung ',Zaehler1,' mit Bedingung ',Zaehler2);
369                         writeln(1st,' : ',Q_BB[Zaehler1,Zaehler2]);

```

```

370         end;
371     end; {for Zaehler2}
372 end; {for Zaehler1}
373
374 writeln;
375
376     for Zaehler1 := 2 to 3 do begin
377         for Zaehler2 := 1 to (Zaehler1-1) do begin
378             hilf1 := 0;
379             hilf2 := 0;
380             hilf3 := 0;
381             for b := 1 to 3 do begin
382                 for i := 1 to 46 do begin
383                     hilf1 := hilf1 + (X_I_M[i,Zaehler1,b] * X_I_M[i,Zaehler2,b]);
384                     hilf2 := hilf2 + (X_I_M[i,Zaehler1,b] * X_I_M[i,Zaehler1,b]);
385                     hilf3 := hilf3 + (X_I_M[i,Zaehler2,b] * X_I_M[i,Zaehler2,b]);
386                 end; {for i}
387             end; {for m}
388             Q_MM[Zaehler1,Zaehler2] := hilf1/(SQRT(hilf2 * hilf3));
389             if (Entsch10='j') then
390                 if (Entsch0 () 'j') then begin
391                     write('Q-Wert von Musik ',Zaehler1,' mit Musik ',Zaehler2);
392                     writeln(' : ',Q_MM[Zaehler1,Zaehler2]);
393                 end
394             else begin
395                 write(1st,'Q-Wert von Musik ',Zaehler1,' mit Musik ',Zaehler2);
396                 writeln(1st,' : ',Q_MM[Zaehler1,Zaehler2]);
397             end;
398         end; {for Zaehler2}
399     end; {for Zaehler1}
400
401     writeln;
402     writeln(' Weiter mit irgendeiner Taste...');
403     read(kbd,Taste);
404
405 end; {procedure Q_Wert}
406
407
408
409 procedure Standardabweichungen;
410
411 var Entsch1, Entsch2, Entsch3, Entsch4, Entsch5, Entsch6,
412     Entsch7, Entsch0 :char;
413
414 begin
415     clrscr;
416     writeln(' Was soll alles ausgedruckt werden? jeweils mit j/n antworten!');
417     writeln;
418     writeln(' Item-Varianz (Über alle Personen):');
419     write(' für alle Item/Bedingungs/Musik-Kombinationen: ');
420     readln(Entsch1);
421     writeln;
422     writeln(' Varianz eines Fragebogens (Über alle Personen und Items):');
423     write(' für alle Musik/Bedingungs-Kombinationen: ');
424     readln(Entsch2);
425     writeln;
426     writeln(' Varianz einer Bedingung (Über alle Personen, Items und Musiken)');
427     write(' für alle Bedingungen: ');
428     readln(Entsch3);
429     writeln;
430     writeln(' Varianz einer Musik (Über alle Personen, Items und Bedingungen)');
431     write(' für alle Musiken: ');

```

```

432     readln(Entsch4);
433     writeln;
434     writeln('      Standardabweichung eines Fragebogens (Über alle Personen und Items:');
435     write('      für alle Musik/Bedingungs-Kombinationen: ');
436     readln(Entsch5);
437     writeln;
438     writeln('      Standardabweichung einer Bedingung (Über alle Personen, Items');
439     write('      und Musiken): für alle Bedingungen: ');
440     readln(Entsch6);
441     writeln;
442     writeln('      Standardabweichung einer Musik (Über alle Personen, Items');
443     write('      und Bedingungen): für alle Musiken: ');
444     readln(Entsch7);
445
446     clrscr;
447     gotoxy(1,10);
448     writeln('      Soll das alles auch auf dem Printer gedruckt werden?');
449     write('      j/n: ');
450     readln(Entsch0);
451
452     clrscr;
453     gotoxy(10,12);
454     writeln('Bitte warten, ich rechne...');
455     writeln;
456     for i := 1 to 46 do begin
457         for m := 1 to 3 do begin
458             for b := 1 to 3 do begin
459                 V_Item[i,m,b] := 0;
460                 for p := 1 to 12 do begin
461                     V_Item[i,m,b] := V_Item[i,m,b] +
462                         ((Data[i,m,b,p] - X_M[i,m,b]) *
463                         (Data[i,m,b,p] - X_M[i,m,b]));
464                 end; {for p}
465                 V_Item[i,m,b] := V_Item[i,m,b]/12;
466                 if (Entsch1='j') then
467                     if (Entsch0 () 'j') then
468                         writeln('V_Item (' ,i ,',', ,m ,',', ,b ,') : ',V_Item[i,m,b])
469                     else writeln(1st,'V_Item (' ,i ,',', ,m ,',', ,b ,') : ',V_Item[i,m,b]);
470                 end; {for b}
471             end; {for m}
472         end; {for i}
473
474     writeln;
475
476     for m := 1 to 3 do begin
477         for b := 1 to 3 do begin
478             V_Skala[m,b] := 0;
479             for i := 1 to 46 do begin
480                 for p := 1 to 12 do begin
481                     V_Skala[m,b] := V_Skala[m,b] +
482                         ((Data[i,m,b,p] - X_M[i,m,b]) *
483                         (Data[i,m,b,p] - X_M[i,m,b]));
484                 end; {for p}
485             end; {for i}
486             V_Skala[m,b] := V_Skala[m,b]/(12 * 46);
487             if (Entsch2='j') then
488                 if (Entsch0 () 'j') then
489                     writeln('V_Skala (' ,m ,',', ,b ,') : ',V_Skala[m,b])
490                 else writeln(1st,'V_Skala (' ,m ,',', ,b ,') : ',V_Skala[m,b]);
491             S_Skala[m,b] := SQRT(V_Skala[m,b]);
492             if (Entsch5='j') then
493                 if (Entsch0 () 'j') then

```

```

494         writeln('Standardabweichung von Musik ',m,' / Bed.',b,' : ',S_Skala[m,b])
495     else writeln(1st,'Standardabweichung von Musik ',m,' / Bed.',b,' : ',S_Skala[m,b]);
496     end; {for b}
497 end; {for m}
498
499 writeln;
500
501     for b := 1 to 3 do begin
502         V_Bedingung[b] := 0;
503         for i := 1 to 46 do begin
504             for m := 1 to 3 do begin
505                 for p := 1 to 12 do begin
506                     V_Bedingung[b] := V_Bedingung[b] +
507                         ((Data[i,m,b,p] - X_M[i,m,b]) *
508                         (Data[i,m,b,p] - X_M[i,m,b]));
509                 end; {for p}
510             end; {for m}
511         end; {for i}
512         V_Bedingung[b] := V_Bedingung[b]/(12 * 46 * 3);
513         if (Entsch3='j') then
514             if (Entsch0 () 'j') then
515                 writeln('V_Bedingung (' ,b,') : ',V_Bedingung[b])
516             else writeln(1st,'V_Bedingung (' ,b,') : ',V_Bedingung[b]);
517         S_Bedingung[b] := Sqrt(V_Bedingung[b]);
518         if (Entsch6='j') then
519             if (Entsch0 () 'j') then
520                 writeln('Standardabweichung von Bedingung ',b,' : ',S_Bedingung[b])
521             else writeln(1st,'Standardabweichung von Bedingung ',b,' : ',S_Bedingung[b]);
522         end; {for b}
523
524     writeln;
525
526     for m := 1 to 3 do begin
527         V_Musiken[m] := 0;
528         for i := 1 to 46 do begin
529             for b := 1 to 3 do begin
530                 for p := 1 to 12 do begin
531                     V_Musiken[m] := V_Musiken[m] +
532                         ((Data[i,m,b,p] - X_M[i,m,b]) *
533                         (Data[i,m,b,p] - X_M[i,m,b]));
534                 end; {for p}
535             end; {for b}
536         end; {for i}
537         V_Musiken[m] := V_Musiken[m]/(12 * 46 * 3);
538         if (Entsch4='j') then
539             if (Entsch0 () 'j') then
540                 writeln('V_Musiken (' ,m,') : ',V_Musiken[m])
541             else writeln(1st,'V_Musiken (' ,m,') : ',V_Musiken[m]);
542         S_Musiken[m] := Sqrt(V_Musiken[m]);
543         if (Entsch7='j') then
544             if (Entsch0 () 'j') then
545                 writeln('Standardabweichung von Musik ',m,' : ',S_Musiken[m])
546             else writeln(1st,'Standardabweichung von Musik ',m,' : ',S_Musiken[m]);
547         end; {for m}
548
549     writeln;
550     writeln(' Weiter mit irgendeiner Taste...');
551     read(kbd,Taste);
552
553 end; {procedure Standardabweichungen}
554
555

```

```

556
557
558 procedure D_Werte;
559
560 var Entsch1, Entsch2, Entsch3, Entsch4, Entsch0 :char;
561
562 begin
563   clrscr;
564   writeln(' Was soll alles ausgedruckt werden? jeweils mit j/n antworten!');
565   writeln;
566   writeln('      Faktorscore der drei Faktoren AKTIVITAET, POTENZ und');
567   writeln('      HARMONIE/ORDNUG eines Fragebogens');
568   writeln('      (Über alle Personen und die je 6 spezifisch ladenden Items):');
569   write('          für alle Musik/Bedingungs-Kombinationen: ');
570   readln(Entsch1);
571   writeln;
572   writeln('      D-Werte (Über die drei mal 6 spezifisch ladenden Items');
573   writeln('      und alle Personen):');
574   write('          für alle Musik/Bedingungs-Kombinations-PAARE: ');
575   readln(Entsch2);
576   writeln;
577   writeln('      D-Werte (Über die drei mal 6 spezifisch ladenden Items,');
578   writeln('      alle Personen und Bedingungen):');
579   write('          für alle Musiken-PAARE: ');
580   readln(Entsch3);
581   writeln;
582   writeln('      D-Werte (Über die drei mal 6 spezifisch ladenden Items,');
583   writeln('      alle Personen und Musiken):');
584   write('          für alle Bedingungs-PAARE: ');
585   readln(Entsch4);
586
587   clrscr;
588   gotoxy(1,10);
589   writeln('      Soll das alles auch auf dem Printer gedruckt werden?');
590   write('          j/n: ');
591   readln(Entsch0);
592
593
594   clrscr;
595   for m := 1 to 3 do begin
596     for b := 1 to 3 do begin
597       F1_MB[m,b] := 7-X_M[40,m,b] +
598                 X_M[27,m,b] +
599                 7-X_M[46,m,b] +
600                 X_M[45,m,b] +
601                 7-X_M[12,m,b] +
602                 X_M[22,m,b];
603       if (Entsch1='j') then
604         if (Entsch0 (>)'j') then
605           writeln('F1_MB['m',' ',',',b,']: ',F1_MB[m,b])
606         else writeln(1st,'F1_MB('m',' ',',',b,'): ',F1_MB[m,b]);
607
608       F2_MB[m,b] := 7-X_M[37,m,b] +
609                 7-X_M[14,m,b] +
610                 X_M[19,m,b] +
611                 X_M[44,m,b] +
612                 X_M[11,m,b] +
613                 X_M[20,m,b];
614       if (Entsch1='j') then
615         if (Entsch0 (>)'j') then
616           writeln('F2_MB['m',' ',',',b,']: ',F2_MB[m,b])
617         else writeln(1st,'F2_MB('m',' ',',',b,'): ',F2_MB[m,b]);

```

```

618
619     F3_MB[m,b] := 7-X_M[4,m,b] +
620                 X_M[9,m,b] +
621                 X_M[42,m,b] +
622                 X_M[5,m,b] +
623                 7-X_M[8,m,b] +
624                 7-X_M[31,m,b];
625     if (Entsch1='j') then
626         if (Entsch0 (<) 'j') then
627             writeln('F3_MB[' ,m ,',',b ,']: ',F3_MB[m,b])
628         else writeln(1st,'F3_MB(' ,m ,',',b ,'): ',F3_MB[m,b]);
629     writeln;
630
631     end; {for b}
632 end; {for m}
633
634 Zaehler3 := 0;
635 for m := 1 to 3 do begin
636     for b := 1 to 3 do begin
637         Zaehler3 := Zaehler3 + 1;
638         Feld[Zaehler3].Mus := m;
639         Feld[Zaehler3].Bed := b;
640     end; {for b}
641 end; {for m}
642
643 writeln;
644
645 for Zaehler1 := 2 to 9 do begin
646     for Zaehler2 := 1 to (Zaehler1 - 1) do begin
647         hilf1 := F1_MB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed] -
648                 F1_MB[Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
649         hilf2 := F2_MB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed] -
650                 F2_MB[Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
651         hilf3 := F3_MB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed] -
652                 F3_MB[Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
653
654         D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
655                 Feld[Zaehler2].Mus,Feld[Zaehler2].Bed] :=
656                 SQR(hilf1*hilf1 + hilf2*hilf2 + hilf3*hilf3);
657
658         if (Entsch2='j') then
659             if (Entsch0 (<) 'j') then begin
660                 write('D-Wert von Mus.',Feld[Zaehler1].Mus,'/ Bed.',Feld[Zaehler1].Bed);
661                 write(' mit Mus.',Feld[Zaehler2].Mus,'/ Bed.',Feld[Zaehler2].Bed);
662                 writeln(' : ',D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
663                                     Feld[Zaehler2].Mus,Feld[Zaehler2].Bed])
664             end
665         else begin
666             write(1st,'D-Wert von Mus.',Feld[Zaehler1].Mus,'/ Bed.',Feld[Zaehler1].Bed);
667             write(1st,' mit Mus.',Feld[Zaehler2].Mus,'/ Bed.',Feld[Zaehler2].Bed);
668             writeln(1st,' : ',D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
669                                     Feld[Zaehler2].Mus,Feld[Zaehler2].Bed]);
670         end;
671     end; {for Zaehler2}
672 end; {for Zaehler1}
673
674 writeln;
675
676 for Zaehler1 := 2 to 3 do begin
677     for Zaehler2 := 1 to (Zaehler1-1) do begin
678         D_MM[Zaehler1,Zaehler2] := 0;
679         D_BB[Zaehler1,Zaehler2] := 0;

```



```

680     end; {for Zaehler2}
681 end; {for Zaehler1}
682
683
684 for Zaehler1 := 2 to 9 do begin
685     for Zaehler2 := 1 to (Zaehler1 - 1) do begin
686
687         if (((Feld[Zaehler1].Mus = 2) AND (Feld[Zaehler2].Mus = 1)) OR
688             ((Feld[Zaehler1].Mus = 1) AND (Feld[Zaehler2].Mus = 2)))
689         then begin
690             D_MM[2,1] := D_MM[2,1] +
691                 D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
692                     Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
693         end; {2/1 Musiken}
694
695         if (((Feld[Zaehler1].Mus = 3) AND (Feld[Zaehler2].Mus = 1)) OR
696             ((Feld[Zaehler1].Mus = 1) AND (Feld[Zaehler2].Mus = 3)))
697         then begin
698             D_MM[3,1] := D_MM[3,1] +
699                 D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
700                     Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
701         end; {3/1 von Musiken}
702
703         if (((Feld[Zaehler1].Mus = 3) AND (Feld[Zaehler2].Mus = 2)) OR
704             ((Feld[Zaehler1].Mus = 2) AND (Feld[Zaehler2].Mus = 3)))
705         then begin
706             D_MM[3,2] := D_MM[3,2] +
707                 D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
708                     Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
709         end; {3/2 von Musiken}
710
711
712         if (((Feld[Zaehler1].Bed = 2) AND (Feld[Zaehler2].Bed = 1)) OR
713             ((Feld[Zaehler1].Bed = 1) AND (Feld[Zaehler2].Bed = 2)))
714         then begin
715             D_BB[2,1] := D_BB[2,1] +
716                 D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
717                     Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
718         end; {2/1 von Bedingungen}
719
720         if (((Feld[Zaehler1].Bed = 3) AND (Feld[Zaehler2].Bed = 1)) OR
721             ((Feld[Zaehler1].Bed = 1) AND (Feld[Zaehler2].Bed = 3)))
722         then begin
723             D_BB[3,1] := D_BB[3,1] +
724                 D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
725                     Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
726         end; {3/1 von Bedingungen}
727
728         if (((Feld[Zaehler1].Bed = 3) AND (Feld[Zaehler2].Bed = 2)) OR
729             ((Feld[Zaehler1].Bed = 2) AND (Feld[Zaehler2].Bed = 3)))
730         then begin
731             D_BB[3,2] := D_BB[3,2] +
732                 D_MBMB[Feld[Zaehler1].Mus,Feld[Zaehler1].Bed,
733                     Feld[Zaehler2].Mus,Feld[Zaehler2].Bed];
734         end; {3/2 von Bedingungen}
735
736     end; {for Zaehler2}
737 end; {for Zaehler1}
738
739
740
741

```

```

742   if (Entsch3='j') then
743     if (Entsch0 (<) 'j') then begin
744       for Zaehler1 := 2 to 3 do begin
745         for Zaehler2 := 1 to (Zaehler1-1) do begin
746           write('D-Wert von Musik ',Zaehler1,' mit Musik ',Zaehler2);
747           writeln(' : ',D_MM[Zaehler1,Zaehler2]);
748         end; {for Zaehler2}
749       end; {for Zaehler1}
750     end
751     else begin
752       for Zaehler1 := 2 to 3 do begin
753         for Zaehler2 := 1 to (Zaehler1-1) do begin
754           write(1st,'D-Wert von Musik ',Zaehler1,' mit Musik ',Zaehler2);
755           writeln(1st,' : ',D_MM[Zaehler1,Zaehler2]);
756         end; {for Zaehler2}
757       end; {for Zaehler1}
758     end;
759
760   writeln;
761
762   if (Entsch4='j') then
763     if (Entsch0 (<) 'j') then begin
764       for Zaehler1 := 2 to 3 do begin
765         for Zaehler2 := 1 to (Zaehler1-1) do begin
766           write('D-Wert von Bedingung ',Zaehler1,' mit Bedingung ',Zaehler2);
767           writeln(' : ',D_BB[Zaehler1,Zaehler2]);
768         end; {for Zaehler2}
769       end; {for Zaehler1}
770     end
771     else begin
772       for Zaehler1 := 2 to 3 do begin
773         for Zaehler2 := 1 to (Zaehler1-1) do begin
774           write(1st,'D-Wert von Bedingung ',Zaehler1,' mit Bedingung ',Zaehler2);
775           writeln(1st,' : ',D_BB[Zaehler1,Zaehler2]);
776         end; {for Zaehler2}
777       end; {for Zaehler1}
778     end;
779
780
781   writeln;
782
783
784
785   writeln('      Weiter mit irgendeiner Taste....');
786   read(kbd,Taste);
787 end; {procedure D-Werte}
788
789
790
791 begin {procedure Auswerten}
792
793 repeat
794   clrscr;
795   writeln;
796   writeln;

```

```

797     writeln;
798     writeln('-----');
799     writeln('      AUSWERTEN:');
800     writeln('-----');
801     writeln;
802     writeln('1. Q-Werte berechnen (Produkt-Moment-Korrelation)');
803     writeln('2. Standardabweichungen berechnen');
804     writeln('   VORAUSSETZUNG: Vorhergehender Aufruf von 1. ');
805     writeln('3. D-Werte berechnen (Entfernungen im semantischen Raum)');
806     writeln;
807     writeln('0. AUSWERTEN verlassen ');
808     writeln;
809     writeln;
810     writeln;
811     writeln('  Bitte geben Sie die Nummer (0 bis 3) der gewünschten Bearbeitung ein!');
812     writeln('      ');
813     repeat
814         read(kbd, andereAuswahl);
815     until (andereAuswahl in ['0', '1', '2', '3']);
816     case andereAuswahl of
817         '1': Q_Werte;
818         '2': Standardabweichungen;
819         '3': D_Werte;
820         '0':;
821     end; {case}
822 until (andereAuswahl = '0');
823
824 end; {procedure Auswerten}
825

```

```

1  procedure Drucken;
2
3  begin
4      clrscr;
5      gotoxy(1,4);
6      writeln('      ROHDATEN AUSDRUCKEN');
7      writeln('      -----');
8      gotoxy(1,8);
9      writeln('      Welche Datei soll ausgedruckt werden?');
10     writeln('      Gerade die aktuell geladene: ',Datei);
11     write('      (j/n) ');
12     read(Taste);
13     if (Taste = 'j') then begin
14         repeat
15             writeln;
16             write('      Wie ist also der Name der Datei? ');
17             readln(Druckdatei);
18             assign(Druckfile,Druckdatei);
19             {$I-} reset(Druckfile) {$I+};
20             ok := (ioresult = 0);
21             if not ok then begin
22                 writeln;
23                 writeln;
24                 writeln('      Die Datei mit dem Namen ',Druckdatei);
25                 writeln('      wurde nicht gefunden!');
26                 writeln('      Entweder existiert sie noch nicht oder der Name wurde falsch geschrieben!');
27                 writeln;
28                 writeln('      Wollen Sie nochmal versuchen? (j/n)');
29                 repeat
30                     read(kbd,Entscheidung);
31                     until (Entscheidung in ['j','n']);
32                 end {if}
33                 else begin
34                     Entscheidung := 'w';      ( "w" wie "Weiter" )
35                     reset(Druckfile);
36                     read(Druckfile,DruckData);
37                     close(Druckfile);
38                 end; {else}
39                 until (Entscheidung in ['n','w']);
40             end {if nicht aktuelle Datei drucken}
41         else begin
42             Druckdata := Data;
43             Entscheidung := 'w';
44         end; {else}
45         if (Entscheidung = 'n') then begin
46             clrscr;
47             gotoxy(1,5);
48             writeln('      Anweisung zum Ausdrucken:');
49             writeln('      -----');
50             writeln;
51             writeln('      Bitte zuerst den Drucker einschalten! O.k.?');
52             writeln;
53             writeln('      Weiter:');
54             writeln('      Der Ausdruck geht leider nicht vollautomatisch!');
55             writeln('      Der Bildschirm wird nun SCROLLEN. Immer wenn er');
56             writeln('      stoppt, bitte folgendes tun:');
57             writeln('      1. (SHIFT) + (PRINT SCR) drücken');
58             writeln('      2. Warten, bis der Drucker den Bildschirminhalt');
59             writeln('      gedruckt hat');
60             writeln('      3. (w) drücken (für "weiter")');
61             writeln;

```

```

62     write('      O.k.? Zum Start irgendeine Taste drücken! ');
63     read(kbd,Taste);
64
65     clrscr;
66     Zaehler := 0;
67     gotoxy(40,1); write('ITEM NUMMER:');
68     gotoxy(27,3); write('      1 111111112 222222223 333333334 444444');
69     gotoxy(27,4); write('1234567890 1234567890 1234567890 1234567890 123456');
70     gotoxy(1,4); write('MUSIK BEDINGUNG VPN');
71     gotoxy(1,6);
72     for m := 1 to 3 do begin
73         for b := 1 to 3 do begin
74             for p := 1 to 12 do begin
75                 write(' ',m,' ',b,' ',p);
76                 if (p<10 ) then write(' ');
77                 write(' ');
78                 for i:= 1 to 46 do begin
79                     write(DruckData[i,m,b,p]);
80                     if (i=10) OR (i=20) OR (i=30) OR (i=40) then write(' ');
81                 end;
82                 Zaehler := Zaehler + 1;
83                 If (Zaehler = 20) OR
84                    (Zaehler = 45) OR
85                    (Zaehler = 70) OR
86                    (Zaehler = 95)
87                 then begin
88                     repeat
89                         read(kbd,Taste);
90                         until (Taste = 'w');
91                     end; {if}
92                     writeln;
93                 end; {for p}
94             end; {for b}
95         end; {for m}
96         for Zaehler := 1 to 11 do writeln;
97         repeat
98             read(kbd,Taste)
99             until (Taste = 'w');
100        end; {if Entscheidung () nein}
101        end; {procedure Drucken}
102 /

```

```

1  procedure ASCII;
2
3  var  Aus1, Aus2, Leer, Ziffer : char;
4         LF, CR : char;
5         s : integer;
6
7  procedure Konvert (Ein: integer; var Aus1, Aus2: char);
8
9  begin
10     case Ein of
11         1,2,3,4,5,6,7,8,9: Aus1 := '0';
12         10,11,12       : Aus1 := '1';
13     end; {case}
14
15     case Ein of
16         10: Aus2 := '0';
17         1,11: Aus2 := '1';
18         2,12: Aus2 := '2';
19         3   : Aus2 := '3';
20         4   : Aus2 := '4';
21         5   : Aus2 := '5';
22         6   : Aus2 := '6';
23         7   : Aus2 := '7';
24         8   : Aus2 := '8';
25         9   : Aus2 := '9';
26     end; {case}
27     end; {procedure Konvert}
28
29 begin
30     Leer := ' ';
31     CR  := chr(13);
32     LF  := chr(10);
33     clrscr;
34     gotoxy(1,4);
35     writeln(' ASCII-DATEI BILDEN UND SPEICHERN');
36     writeln(' -----');
37     gotoxy(1,8);
38     writeln(' Wie soll der Name der Ascii-Datei lauten, in der die Daten');
39     writeln('   abgespeichert werden sollen?');
40     writeln(' EMPFEHLUNG: Nehmen Sie einen Namen mit Extension ".ASC!');
41     write(' ');
42     readln(AscDatei);
43     assign(Ascifile,AscDatei);
44     {$I-} reset(Ascifile) {$I+};
45     ok := (ioresult = 0);
46     if ok then begin
47         writeln;
48         writeln(' Vorsicht! Diese Datei existiert schon. ');
49         writeln(' Beim Abspeichern werden die alten Daten überschrieben! ');
50         writeln(' Ist dies in Ordnung? (j für ja, mit irgendeiner anderen ');
51         writeln(' Taste verlassen Sie dieses Untermenu) ');
52         read(kbd,Entscheidung);
53     end {if}
54     else begin
55         writeln;
56         writeln(' Diese Datei existiert noch nicht. ');
57         writeln(' Beim abspeichern wird sie neu eröffnet. ');
58         writeln(' Ist dies in Ordnung? (j für ja, mit irgendeiner anderen ');
59         writeln(' Taste verlassen Sie dieses Untermenu) ');
60         read(kbd,Entscheidung);
61     end; {else}

```

```

62     if (Entscheidung in ['j','J']) then begin
63         clrscr;
64         gotoxy(1,8);
65         writeln('    Bitte einen Moment warten...');
66         rewrite(Asciifile);
67         for m:= 1 to 3 do begin
68             for b:= 1 to 3 do begin
69                 for p:= 1 to 12 do begin
70                     Konvert(m,Aus1,Aus2);
71                     write(Asciifile,Aus1);write(Asciifile,Aus2);write(Asciifile,Leer);
72                     Konvert(b,Aus1,Aus2);
73                     write(Asciifile,Aus1);write(Asciifile,Aus2);write(Asciifile,Leer);
74                     Konvert(p,Aus1,Aus2);
75                     write(Asciifile,Aus1);write(Asciifile,Aus2);write(Asciifile,Leer);
76                 for i := 1 to 46 do begin
77                     case Data[i,m,b,p] of
78                         1: Ziffer := '1';
79                         2: Ziffer := '2';
80                         3: Ziffer := '3';
81                         4: Ziffer := '4';
82                         5: Ziffer := '5';
83                         6: Ziffer := '6';
84                     end;
85                     if (Data[i,m,b,p] < 1) OR (Data[i,b,m,p] >6) then Ziffer := '0';
86                     write(Asciifile,Ziffer);
87                 end; {for i}
88                 write(Asciifile,CR);
89                 write(Asciifile,LF);
90             end; {for p}
91         end; {for b}
92     end; {for m}
93     gotoxy(1,12);
94     writeln('    O.k., nun sind die Daten auf Diskette gespeichert, und zwar');
95     writeln('    unter dem Namen ',AscDatei);
96     gotoxy(1,23);
97     writeln('                Weiter mit irgendeiner Taste...');
98     read(kbd,Taste);
99     end; {if}
100    close(Asciifile);
101    end; {procedure ASCII}
102

```

```

1  procedure Initialisieren;
2
3  begin
4      clrscr;
5      gotoxy(1,8);
6      write('      Was soll geschrieben werden? (eine Zahl):  ');
7      readln(Zahl);
8      for i:= 1 to 46 do begin
9          for m := 1 to 3 do begin
10             for b := 1 to 3 do begin
11                 for p := 1 to 12 do begin
12                     Data[i,m,b,p] := Zahl;
13                 end;
14             end;
15         end;
16     end;
17     writeln;
18     writeln('      Wie lautet der Name für die Speicherdatei?');
19     writeln('      VORSICHT! Eventuell schon existierende Dateien mit diesem');
20     writeln('      Namen werden gelöscht!');
21     write('      ');
22     readln(Datei);
23     assign(Datenfile,Datei);
24     rewrite(Datenfile);
25     write(Datenfile,Data);
26     close(Datenfile);
27     writeln('ok, geschrieben');
28 end; {procedure Initialisieren}
29

```



```
1 procedure Init2;
2
3 begin
4     clrscr;
5     gotoxy(10,20);
6     writeln('Ich bin am Initialisieren. Bitte (ab-)warten...');
7     for i:= 1 to 46 do begin
8         for m := 1 to 3 do begin
9             for b := 1 to 3 do begin
10                for p := 1 to 12 do begin
11                    Data[i,m,b,p] := TRUNC(RANDOM(6)) + 1;
12                end; {for p}
13            end; {for b}
14        end; {for m}
15    end; {for i}
16 end; {procedure Init2}
17
```

```

1  procedure Haeufigkeiten;
2
3  var i,j,k,l : integer;
4      eins, zwei, drei, vier, fuenf, sechs : integer;
5
6  begin
7      clrscr;
8      gotoxy(5,10);
9      writeln('Moment, ich rechne die Häufigkeiten aus...');
10     eins := 0;
11     zwei := 0;
12     drei := 0;
13     vier := 0;
14     fuenf := 0;
15     sechs := 0;
16     for i := 1 to 46 do begin
17         for j := 1 to 3 do begin
18             for k := 1 to 3 do begin
19                 for l := 1 to 12 do begin
20                     if (data[i,j,k,l] = 1) then eins := eins + 1;
21                     if (data[i,j,k,l] = 2) then zwei := zwei + 1;
22                     if (data[i,j,k,l] = 3) then drei := drei + 1;
23                     if (data[i,j,k,l] = 4) then vier := vier + 1;
24                     if (data[i,j,k,l] = 5) then fuenf := fuenf + 1;
25                     if (data[i,j,k,l] = 6) then sechs := sechs + 1;
26                 end;
27             end;
28         end;
29     end;
30     clrscr;
31     gotoxy(1,10);
32     writeln(' Eins: ',eins);
33     writeln(' Zwei: ',zwei);
34     writeln(' Drei: ',drei);
35     writeln(' Vier: ',vier);
36     writeln(' Fünf: ',fuenf);
37     writeln(' Sechs: ',sechs);
38     gotoxy(5,22);
39     writeln('Weiter mit irgendeiner Taste...');
40     read(kbd,Taste);
41 end; {procedure Haeufigkeiten}
42
43

```

```

1  procedure menu;
2
3  begin
4      clrscr;
5      writeln('          -----');
6      writeln('          HAUPTMENU:');
7      writeln('          -----');
8      writeln;
9      writeln('      1. schon früher eingegebene Daten von Diskette LADEN');
10     writeln('      2. neu eingegebene Daten auf Diskette SPEICHERN');
11     writeln('      3. neue Daten EINTIPPEN');
12     writeln('      4. Daten ANZEIGEN');
13     writeln('      5. Daten statistisch AUSWERTEN');
14     writeln('      6. Rohdaten AUSDRUCKEN');
15     writeln('      7. ASCII-File herstellen');
16     writeln('      8. INITIALISIEREN des Programmes');
17     writeln;
18     writeln('      9. special 1 (Zufälliges Initialisieren der Daten-');
19     writeln('         strukturen');
20     writeln('      A. special 2 (Häufigkeiten der Rohdatenwerte)');
21     writeln('      B. special 3 (Mittelwerte/Varianzen für die MW-Pro-');
22     writeln('         file der Faktorstufen)');
23     writeln;
24     writeln('      0. Programm VERLASSEN');
25     writeln;
26     writeln;
27     writeln('      Bitte geben Sie die Nummer (0 bis 9) der gewünschten Bearbeitung ein!');
28     writeln('         ');
29     repeat
30         read(kbd,Auswahl);
31     until (Auswahl in ['0','1','2','3','4','5','6','7','8','9','A','B']);
32     case Auswahl of
33         '1': Laden;
34         '2': Speichern;
35         '3': Eintippen;
36         '4': Anzeigen;
37         '5': Auswerten;
38         '6': Drucken;
39         '7': ASCII;
40         '8': Initialisieren;
41         '9': Init2;
42         'A': Haeufigkeiten;
43         'B': Mittelw_Standardabw;
44         '0':;
45     end; {case}
46 end; {procedure menu}
47

```

12.5. Teil des Ausdruckes des PC-Programmes (Q-Werte,  
Standardabweichungen, Koordinaten des semantischen  
Raumes, D-Werte):

Q-Wert von Mus.1/ Bed.2 mit Mus.1/ Bed.1 : 6.0366937965E-01  
Q-Wert von Mus.1/ Bed.3 mit Mus.1/ Bed.1 : 3.8968755173E-01  
Q-Wert von Mus.1/ Bed.3 mit Mus.1/ Bed.2 : 7.1359929029E-01  
Q-Wert von Mus.2/ Bed.1 mit Mus.1/ Bed.1 : 1.6214843529E-01  
Q-Wert von Mus.2/ Bed.1 mit Mus.1/ Bed.2 : 3.8163771063E-01  
Q-Wert von Mus.2/ Bed.1 mit Mus.1/ Bed.3 : 5.9658323832E-01  
Q-Wert von Mus.2/ Bed.2 mit Mus.1/ Bed.1 : 8.2488865325E-02  
Q-Wert von Mus.2/ Bed.2 mit Mus.1/ Bed.2 : 4.1026550240E-01  
Q-Wert von Mus.2/ Bed.2 mit Mus.1/ Bed.3 : 6.2024991248E-01  
Q-Wert von Mus.2/ Bed.2 mit Mus.2/ Bed.1 : 7.4895751053E-01  
Q-Wert von Mus.2/ Bed.3 mit Mus.1/ Bed.1 : -9.5805542381E-02  
Q-Wert von Mus.2/ Bed.3 mit Mus.1/ Bed.2 : 1.8054642737E-01  
Q-Wert von Mus.2/ Bed.3 mit Mus.1/ Bed.3 : 3.8822191995E-01  
Q-Wert von Mus.2/ Bed.3 mit Mus.2/ Bed.1 : 6.8022122281E-01  
Q-Wert von Mus.2/ Bed.3 mit Mus.2/ Bed.2 : 7.6528588410E-01  
Q-Wert von Mus.3/ Bed.1 mit Mus.1/ Bed.1 : 8.0807714597E-02  
Q-Wert von Mus.3/ Bed.1 mit Mus.1/ Bed.2 : 3.0085470297E-01  
Q-Wert von Mus.3/ Bed.1 mit Mus.1/ Bed.3 : 4.9075896628E-01  
Q-Wert von Mus.3/ Bed.1 mit Mus.2/ Bed.1 : 3.3801437085E-01  
Q-Wert von Mus.3/ Bed.1 mit Mus.2/ Bed.2 : 6.0508410259E-01  
Q-Wert von Mus.3/ Bed.1 mit Mus.2/ Bed.3 : 6.8809887104E-01  
Q-Wert von Mus.3/ Bed.2 mit Mus.1/ Bed.1 : 4.1818416864E-01  
Q-Wert von Mus.3/ Bed.2 mit Mus.1/ Bed.2 : 6.9600418991E-01  
Q-Wert von Mus.3/ Bed.2 mit Mus.1/ Bed.3 : 9.0351060212E-01  
Q-Wert von Mus.3/ Bed.2 mit Mus.2/ Bed.1 : 6.6866847397E-01  
Q-Wert von Mus.3/ Bed.2 mit Mus.2/ Bed.2 : 6.2234750847E-01  
Q-Wert von Mus.3/ Bed.2 mit Mus.2/ Bed.3 : 3.8355913188E-01  
Q-Wert von Mus.3/ Bed.2 mit Mus.3/ Bed.1 : 4.5559879949E-01  
Q-Wert von Mus.3/ Bed.3 mit Mus.1/ Bed.1 : 1.2234806145E-01  
Q-Wert von Mus.3/ Bed.3 mit Mus.1/ Bed.2 : 2.6439221546E-01  
Q-Wert von Mus.3/ Bed.3 mit Mus.1/ Bed.3 : 4.6097350191E-01  
Q-Wert von Mus.3/ Bed.3 mit Mus.2/ Bed.1 : 4.0345061212E-01  
Q-Wert von Mus.3/ Bed.3 mit Mus.2/ Bed.2 : 5.5523403849E-01  
Q-Wert von Mus.3/ Bed.3 mit Mus.2/ Bed.3 : 7.1940412372E-01  
Q-Wert von Mus.3/ Bed.3 mit Mus.3/ Bed.1 : 8.2561489912E-01  
Q-Wert von Mus.3/ Bed.3 mit Mus.3/ Bed.2 : 4.3687982453E-01  
Q-Wert von Bedingung 2 mit Bedingung 1 : 5.8792633792E-01  
Q-Wert von Bedingung 3 mit Bedingung 1 : 6.0562412271E-01  
Q-Wert von Bedingung 3 mit Bedingung 2 : 6.2170163834E-01  
Q-Wert von Musik 2 mit Musik 1 : 3.3025178014E-01  
Q-Wert von Musik 3 mit Musik 1 : 4.2800402645E-01  
Q-Wert von Musik 3 mit Musik 2 : 5.4803006366E-01  
Standardabweichung von Musik 1/ Bed.1 : 1.2597446255E+00  
Standardabweichung von Musik 1/ Bed.2 : 1.5572314556E+00  
Standardabweichung von Musik 1/ Bed.3 : 1.4154406459E+00  
Standardabweichung von Musik 2/ Bed.1 : 1.2495772231E+00  
Standardabweichung von Musik 2/ Bed.2 : 1.1258383885E+00  
Standardabweichung von Musik 2/ Bed.3 : 1.3869430814E+00  
Standardabweichung von Musik 3/ Bed.1 : 1.5316212085E+00  
Standardabweichung von Musik 3/ Bed.2 : 1.1086778912E+00  
Standardabweichung von Musik 3/ Bed.3 : 1.3384751400E+00  
Standardabweichung von Bedingung 1 : 1.3533000755E+00  
Standardabweichung von Bedingung 2 : 1.2808393796E+00  
Standardabweichung von Bedingung 3 : 1.3806519030E+00  
Standardabweichung von Musik 1 : 1.4160271359E+00  
Standardabweichung von Musik 2 : 1.2586456244E+00  
Standardabweichung von Musik 3 : 1.3374784598E+00  
F1\_MB(1,1): 2.0583333333E+01  
F2\_MB(1,1): 2.5083333333E+01  
F3\_MB(1,1): 1.7750000000E+01  
F1\_MB(1,2): 1.8500000000E+01  
F2\_MB(1,2): 2.5083333333E+01  
F3\_MB(1,2): 1.7166666667E+01

F1\_MB(1,3): 1.5750000000E+01  
F2\_MB(1,3): 2.2500000000E+01  
F3\_MB(1,3): 1.6000000000E+01  
F1\_MB(2,1): 1.8000000000E+01  
F2\_MB(2,1): 1.8333333333E+01  
F3\_MB(2,1): 1.4833333333E+01  
F1\_MB(2,2): 1.6250000000E+01  
F2\_MB(2,2): 1.6166666667E+01  
F3\_MB(2,2): 1.7333333333E+01  
F1\_MB(2,3): 1.8500000000E+01  
F2\_MB(2,3): 1.4583333333E+01  
F3\_MB(2,3): 1.6583333333E+01  
F1\_MB(3,1): 1.7083333333E+01  
F2\_MB(3,1): 1.7833333333E+01  
F3\_MB(3,1): 2.1083333333E+01  
F1\_MB(3,2): 1.4250000000E+01  
F2\_MB(3,2): 2.2333333333E+01  
F3\_MB(3,2): 1.3666666667E+01  
F1\_MB(3,3): 1.9583333333E+01  
F2\_MB(3,3): 1.8833333333E+01  
F3\_MB(3,3): 2.0333333333E+01  
D-Wert von Mus.1/ Bed.2 mit Mus.1/ Bed.1 : 2.1634591643E+00  
D-Wert von Mus.1/ Bed.3 mit Mus.1/ Bed.1 : 5.7530185313E+00  
D-Wert von Mus.1/ Bed.3 mit Mus.1/ Bed.2 : 3.9493318703E+00  
D-Wert von Mus.2/ Bed.1 mit Mus.1/ Bed.1 : 7.7937831350E+00  
D-Wert von Mus.2/ Bed.1 mit Mus.1/ Bed.2 : 7.1593955362E+00  
D-Wert von Mus.2/ Bed.1 mit Mus.1/ Bed.3 : 4.8769582961E+00  
D-Wert von Mus.2/ Bed.2 mit Mus.1/ Bed.1 : 9.9226172622E+00  
D-Wert von Mus.2/ Bed.2 mit Mus.1/ Bed.2 : 9.1976748269E+00  
D-Wert von Mus.2/ Bed.2 mit Mus.1/ Bed.3 : 6.4914473647E+00  
D-Wert von Mus.2/ Bed.2 mit Mus.2/ Bed.1 : 3.7425852621E+00  
D-Wert von Mus.2/ Bed.3 mit Mus.1/ Bed.1 : 1.0768072664E+01  
D-Wert von Mus.2/ Bed.3 mit Mus.1/ Bed.2 : 1.0516191220E+01  
D-Wert von Mus.2/ Bed.3 mit Mus.1/ Bed.3 : 8.4009754724E+00  
D-Wert von Mus.2/ Bed.3 mit Mus.2/ Bed.1 : 4.1683330001E+00  
D-Wert von Mus.2/ Bed.3 mit Mus.2/ Bed.2 : 2.8516564387E+00  
D-Wert von Mus.3/ Bed.1 mit Mus.1/ Bed.1 : 8.7134155823E+00  
D-Wert von Mus.3/ Bed.1 mit Mus.1/ Bed.2 : 8.3612033955E+00  
D-Wert von Mus.3/ Bed.1 mit Mus.1/ Bed.3 : 7.0282169384E+00  
D-Wert von Mus.3/ Bed.1 mit Mus.2/ Bed.1 : 6.3366219532E+00  
D-Wert von Mus.3/ Bed.1 mit Mus.2/ Bed.2 : 4.1874481755E+00  
D-Wert von Mus.3/ Bed.1 mit Mus.2/ Bed.3 : 5.7288257475E+00  
D-Wert von Mus.3/ Bed.2 mit Mus.1/ Bed.1 : 8.0216720340E+00  
D-Wert von Mus.3/ Bed.2 mit Mus.1/ Bed.2 : 6.1542668125E+00  
D-Wert von Mus.3/ Bed.2 mit Mus.1/ Bed.3 : 2.7788886667E+00  
D-Wert von Mus.3/ Bed.2 mit Mus.2/ Bed.1 : 5.6056766863E+00  
D-Wert von Mus.3/ Bed.2 mit Mus.2/ Bed.2 : 7.4479676571E+00  
D-Wert von Mus.3/ Bed.2 mit Mus.2/ Bed.3 : 9.3076282932E+00  
D-Wert von Mus.3/ Bed.2 mit Mus.3/ Bed.1 : 9.1260463631E+00  
D-Wert von Mus.3/ Bed.3 mit Mus.1/ Bed.1 : 6.8363814340E+00  
D-Wert von Mus.3/ Bed.3 mit Mus.1/ Bed.2 : 7.0897030184E+00  
D-Wert von Mus.3/ Bed.3 mit Mus.1/ Bed.3 : 6.8495741960E+00  
D-Wert von Mus.3/ Bed.3 mit Mus.2/ Bed.1 : 5.7451670510E+00  
D-Wert von Mus.3/ Bed.3 mit Mus.2/ Bed.2 : 5.2174919475E+00  
D-Wert von Mus.3/ Bed.3 mit Mus.2/ Bed.3 : 5.7704948758E+00  
D-Wert von Mus.3/ Bed.3 mit Mus.3/ Bed.1 : 2.7950849719E+00  
D-Wert von Mus.3/ Bed.3 mit Mus.3/ Bed.2 : 9.2270736904E+00  
D-Wert von Musik 2 mit Musik 1 : 7.5127115777E+01  
D-Wert von Musik 3 mit Musik 1 : 6.1833322078E+01  
D-Wert von Musik 3 mit Musik 2 : 5.5347322387E+01  
D-Wert von Bedingung 2 mit Bedingung 1 : 5.8290103879E+01  
D-Wert von Bedingung 3 mit Bedingung 1 : 5.3700058634E+01  
D-Wert von Bedingung 3 mit Bedingung 2 : 5.7429412510E+01

## 12.6. Ausdruck der BEDAG Faktoranalyse:

10 MAY 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 17:41:40 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

For MVS/SP 2.1.7 BERNISCHE DATENVERARBEITUNG AG License Number 1526

Use INFO OVERVIEW for more information on:

\* INCLUDE - To bring in command files \* Improvements in:  
 \* RENAME VARS - To rename variables \* MANOVA  
 \* AUTORECODE - To recode strings as numbers \* TABLES  
 \* DROP DOCUMENTS \*

1 0 UNNUMBERED  
 2 0 SET LENGTH=NONE  
 3 0 DATA LIST FILE=IN /1 M 1-2 B 4-5 V 7-8 I1 TO I46 10-55

THE ABOVE DATA LIST STATEMENT WILL READ 1 RECORDS FROM FILE IN

VARIABLE	REC	START	END	FORMAT	WIDTH	DEC
M	1	1	2	F	2	0
B	1	4	5	F	2	0
V	1	7	8	F	2	0
I1	1	10	10	F	1	0
I2	1	11	11	F	1	0
I3	1	12	12	F	1	0
I4	1	13	13	F	1	0
I5	1	14	14	F	1	0
I6	1	15	15	F	1	0
I7	1	16	16	F	1	0
I8	1	17	17	F	1	0
I9	1	18	18	F	1	0
I10	1	19	19	F	1	0
I11	1	20	20	F	1	0
I12	1	21	21	F	1	0
I13	1	22	22	F	1	0
I14	1	23	23	F	1	0
I15	1	24	24	F	1	0
I16	1	25	25	F	1	0
I17	1	26	26	F	1	0
I18	1	27	27	F	1	0
I19	1	28	28	F	1	0
I20	1	29	29	F	1	0
I21	1	30	30	F	1	0
I22	1	31	31	F	1	0
I23	1	32	32	F	1	0
I24	1	33	33	F	1	0
I25	1	34	34	F	1	0
I26	1	35	35	F	1	0
I27	1	36	36	F	1	0
I28	1	37	37	F	1	0
I29	1	38	38	F	1	0
I30	1	39	39	F	1	0
I31	1	40	40	F	1	0
I32	1	41	41	F	1	0
I33	1	42	42	F	1	0
I34	1	43	43	F	1	0
I35	1	44	44	F	1	0
I36	1	45	45	F	1	0
I37	1	46	46	F	1	0
I38	1	47	47	F	1	0
I39	1	48	48	F	1	0
I40	1	49	49	F	1	0
I41	1	50	50	F	1	0
I42	1	51	51	F	1	0
I43	1	52	52	F	1	0
I44	1	53	53	F	1	0
I45	1	54	54	F	1	0
I46	1	55	55	F	1	0

END OF DATALIST TABLE.

4 0 FACTOR VARIABLES=I1 TO I46/FORMAT=SORT BLANK(.3)/CRITERIA=FACTORS(3)

THERE ARE 492536 BYTES OF MEMORY AVAILABLE.  
THE LARGEST CONTIGUOUS AREA HAS 492536 BYTES.

>NOTE 11284

>Since the ANALYSIS subcommand is not used, all variables on the VARIABLES  
>subcommand will be used for the first analysis.

THIS FACTOR ANALYSIS REQUIRES 240240 ( 234.6K) BYTES OF MEMORY.

10 MAY 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
17:41:42 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

----- FACTOR ANALYSIS -----

ANALYSIS NUMBER 1 LISTWISE DELETION OF CASES WITH MISSING VALUES

EXTRACTION 1 FOR ANALYSIS 1, PRINCIPAL-COMPONENTS ANALYSIS (PC)

INITIAL STATISTICS:

VARIABLE	COMMUNALITY	*	FACTOR	EIGENVALUE	PCT OF VAR	CUM PCT
I1	1.00000	*	1	11.56926	25.2	25.2
I2	1.00000	*	2	8.24285	17.9	43.1
I3	1.00000	*	3	3.46681	7.5	50.6
I4	1.00000	*	4	2.28317	5.0	55.6
I5	1.00000	*	5	1.72876	3.8	59.3
I6	1.00000	*	6	1.28556	2.8	62.1
I7	1.00000	*	7	1.21632	2.6	64.8
I8	1.00000	*	8	1.16478	2.5	67.3
I9	1.00000	*	9	1.06605	2.3	69.6
I10	1.00000	*	10	.99144	2.2	71.8
I11	1.00000	*	11	.91696	2.0	73.8
I12	1.00000	*	12	.88938	1.9	75.7
I13	1.00000	*	13	.81181	1.8	77.5
I14	1.00000	*	14	.74756	1.6	79.1
I15	1.00000	*	15	.73123	1.6	80.7
I16	1.00000	*	16	.66912	1.5	82.1
I17	1.00000	*	17	.64071	1.4	83.5
I18	1.00000	*	18	.63574	1.4	84.9
I19	1.00000	*	19	.52792	1.1	86.1
I20	1.00000	*	20	.50801	1.1	87.2
I21	1.00000	*	21	.49234	1.1	88.2
I22	1.00000	*	22	.46097	1.0	89.2
I23	1.00000	*	23	.43083	.9	90.2
I24	1.00000	*	24	.41575	.9	91.1
I25	1.00000	*	25	.36959	.8	91.9
I26	1.00000	*	26	.32600	.7	92.6
I27	1.00000	*	27	.32119	.7	93.3
I28	1.00000	*	28	.29657	.6	93.9
I29	1.00000	*	29	.29142	.6	94.6
I30	1.00000	*	30	.27686	.6	95.2
I31	1.00000	*	31	.24940	.5	95.7
I32	1.00000	*	32	.22860	.5	96.2
I33	1.00000	*	33	.21063	.5	96.7
I34	1.00000	*	34	.19045	.4	97.1
I35	1.00000	*	35	.17601	.4	97.5
I36	1.00000	*	36	.16749	.4	97.8



I37	1.00000	*	37	.15000	.3	98.1
I38	1.00000	*	38	.13613	.3	98.4
I39	1.00000	*	39	.13239	.3	98.7
I40	1.00000	*	40	.11965	.3	99.0
I41	1.00000	*	41	.09906	.2	99.2
I42	1.00000	*	42	.09127	.2	99.4
I43	1.00000	*	43	.08550	.2	99.6
I44	1.00000	*	44	.07603	.2	99.8
I45	1.00000	*	45	.05937	.1	99.9
I46	1.00000	*	46	.05308	.1	100.0

PC EXTRACTED 3 FACTORS.

FACTOR MATRIX:

	FACTOR 1	FACTOR 2	FACTOR 3
I15	-.78514		
I38	.74483		
I34	.74059		.30131
I43	.72888		
I35	.72846		
I46	-.71945		
I36	-.67618		
I28	.67550		.38269
I1	.67232		
I29	.66876	.35416	
I17	.66624	.44074	
I23	.64224	.33981	
I20	-.63020	.42272	
I9	.61520		.38888
I45	.61014	.36638	
I25	-.58380	.43360	
I12	-.57441	-.41397	
I5	.50835		.40451
I4	-.50453		-.48041
I26	-.50365		
I11	-.50334	.49976	
I18	.50289	-.40304	-.31241
I16	.44134		.41210
I32		.79892	
I37	.36193	-.72733	
I13		.67850	
I41		-.66970	
I44	-.45981	.66045	
I21		.65717	
I3		-.64447	
I40	-.53558	-.63781	
I19	-.37130	.63385	.31408
I14	.48012	-.57395	
I27	.48193	.57338	
I30		.57149	
I33		-.56032	
I2	.30609	-.49407	
I24	.32513	.41465	
I6		-.37726	
I10			
I7	-.34098		.63986
I42	.38633		.51027
I8			-.50779
I31			-.47992
I39	-.37265		.46179
I22	.39777	.32707	-.44245

FINAL STATISTICS:

VARIABLE	COMMUNALITY	✕	FACTOR	EIGENVALUE	PCT OF VAR	CUM PCT
I1	.48416	✕	1	11.56926	25.2	25.2
I2	.35479	✕	2	8.24285	17.9	43.1
I3	.46357	✕	3	3.46681	7.5	50.6
I4	.53472	✕				
I5	.44174	✕				
I6	.19831	✕				
I7	.52626	✕				
I8	.37066	✕				
I9	.53282	✕				
I10	.04374	✕				
I11	.55320	✕				
I12	.50164	✕				
I13	.50731	✕				
I14	.63216	✕				
I15	.63349	✕				
I16	.44455	✕				
I17	.64353	✕				
I18	.51294	✕				
I19	.63827	✕				
I20	.59084	✕				
I21	.45374	✕				
I22	.46097	✕				
I23	.52881	✕				
I24	.28376	✕				
I25	.54345	✕				
I26	.33730	✕				
I27	.56174	✕				
I28	.63720	✕				
I29	.57585	✕				
I30	.35415	✕				
I31	.35455	✕				
I32	.66536	✕				
I33	.46926	✕				
I34	.64531	✕				
I35	.53714	✕				
I36	.52910	✕				
I37	.74049	✕				
I38	.59567	✕				
I39	.39067	✕				
I40	.72622	✕				
I41	.47423	✕				
I42	.44133	✕				
I43	.56173	✕				
I44	.66037	✕				
I45	.51524	✕				
I46	.62658	✕				

VARIMAX ROTATION 1 FOR EXTRACTION 1 IN ANALYSIS 1 - KAISER NORMALIZATION.

VARIMAX CONVERGED IN 6 ITERATIONS.

ROTATED FACTOR MATRIX:

	FACTOR 1	FACTOR 2	FACTOR 3
I40	-.83962		
I27	.70421		
I46	-.69549		
I17	.68398		.41761
I29	.67337		.31460
I45	.66060		
I12	-.64429		
I22	.63616		
I23	.61499		.37145
I32	.59893	.53864	
I3	-.58405	-.32154	

I36	-.57864	.40516	
I39	-.53621		
I21	.53319	.39041	
I13	.51469	.45552	
I35	.49933	-.36329	.39476
I30	.46505	.35052	
I7	-.46254	.44387	.33955
I24	.45849		
I6	-.43163		
I26	-.41349	.40559	
I10			
I37		-.83974	
I14		-.79206	
I19		.79148	
I44		.77685	
I11		.73396	
I20		.70931	
I18		-.70209	
I25		.68998	
I38	.30075	-.56057	.43701
I43	.33735	-.55326	.37661
I15	-.40062	.55171	-.41062
I41	-.44676	-.49127	
I2		-.47003	
I33	-.32152	-.44979	.40445
I28		-.35648	.70189
I4			-.67189
I9			.66843
I34		-.35098	.66698
I42			.64529
I5			.61425
I8			-.58420
I16	.31009		.57772
I31			-.51828
I1		-.37551	.51097

FACTOR TRANSFORMATION MATRIX:

	FACTOR 1	FACTOR 2	FACTOR 3
FACTOR 1	.60211	-.56913	.55996
FACTOR 2	.70448	.70875	-.03716
FACTOR 3	-.37573	.41686	.82768

10 MAY 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 17:41:43 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

PRECEDING TASK REQUIRED 0.76 SECONDS CPU TIME; 2.43 SECONDS ELAPSED.

5 0 FINISH

5 COMMAND LINES READ.  
 0 ERRORS DETECTED.  
 0 WARNINGS ISSUED.  
 1 SECONDS CPU TIME.  
 3 SECONDS ELAPSED TIME.  
 END OF JOB.

12.7. Ausdruck der BEDAG Varianzanalyse über die Standardabweichungen:

```

- 01 JUN 88  SPSS-X RELEASE 2.2  FOR IBM/MVS
3 16:57:57  BERNISCHE DATENVERARBEITUNG AG IBM 3090-180  MVS/SP 2.1.7
0      *** ANALYSIS OF VARIANCE ***
      BY      STD
            M
            B
- SOURCE OF VARIATION      SUM OF      MEAN      SIGNIF
      SQUARES      DF      SQUARE      F      OF F
O MAIN EFFECTS      3.035      4      0.759      13.793      0.000
  M      1.954      2      0.977      17.758      0.000
  B      1.081      2      0.541      9.828      0.000
O2-WAY INTERACTIONS      7.567      4      1.892      34.390      0.000
  M      B      7.567      4      1.892      34.390      0.000
OEXPLAINED      10.602      8      1.325      24.091      0.000
ORESIDUAL      22.278      405      0.055
OTOTAL      32.880      413      0.080
- 414 CASES WERE PROCESSED.
  0 CASES ( 0.0 PCT) WERE MISSING.

```

```

- 01 JUN 88  SPSS-X RELEASE 2.2  FOR IBM/MVS
4 16:57:57  BERNISCHE DATENVERARBEITUNG AG IBM 3090-180  MVS/SP 2.1.7
0      *** MULTIPLE CLASSIFICATION ANALYSIS ***
      BY      STD
            M
            B
-GRAND MEAN =      1.37
      UNADJUSTED      ADJUSTED FOR      ADJUSTED FOR
      DEV'N      ETA      INDEPENDENTS      + COVARIATES
      DEV'N      BETA      DEV'N      BETA
O VARIABLE + CATEGORY      N
O M      1      138      0.09      0.09
      2      138      -0.08      -0.08
      3      138      -0.01      -0.01
      0.24      0.24
O B      1      138      0.02      0.02
      2      138      -0.07      -0.07
      3      138      0.05      0.05
      0.18      0.18
OMULTIPLE R SQUARED      0.092
MULTIPLE R      0.304

```

```

- 01 JUN 88  SPSS-X RELEASE 2.2  FOR IBM/MVS
5 16:57:57  BERNISCHE DATENVERARBEITUNG AG IBM 3090-180  MVS/SP 2.1.7
O PRECEDING TASK REQUIRED      0.16 SECONDS CPU TIME;      0.75 SECONDS ELAPSED.
6 0 FINISH
0 6 COMMAND LINES READ.
0 ERRORS DETECTED.
0 WARNINGS ISSUED.
0 SECONDS CPU TIME.
2 SECONDS ELAPSED TIME.
  END OF JOB.
END OF DATA

```

101 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS

16:57:56 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

0For MVS/SP 2.1.7 BERNISCHE DATENVERARBEITUNG AG License Number 1526

Use INFO OVERVIEW for more information on:

- 0\* INCLUDE - To bring in command files
  - \* RENAME VARS - To rename variables
  - \* AUTORECODE - To recode strings as numbers
  - \* DROP DOCUMENTS
- \* Improvements in:
  - \* MANOVA
  - \* TABLES
  - \*

1 0 UNNUMBERED  
 2 0 SET LENGTH=NONE  
 3 0 DATA LIST FILE=IN NOTABLE /1 M 1 B 2 N 3-4 STD 5-12(5)  
 4 0 ANOVA STD BY M(1,3) B(1,3)  
 5 0 STATISTICS 1,3  
 0'ANOVA' PROBLEM REQUIRES 1120 BYTES OF MEMORY.

01 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS

2 16:57:57 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

0 \*\*\* CELL MEANS \*\*\*

0 STD  
BY M  
B

-TOTAL POPULATION

0 1.37  
( 414)

0M

	1	2	3
0	1.46	1.29	1.36
	( 138)	( 138)	( 138)

0B

	1	2	3
0	1.39	1.30	1.42
	( 138)	( 138)	( 138)

0

B

		1	2	3
M	1	1.30	1.61	1.45
		( 46)	( 46)	( 46)
	2	1.28	1.15	1.43
		( 46)	( 46)	( 46)
	3	1.58	1.13	1.38
		( 46)	( 46)	( 46)

12.8. Ausdruck der BEDAG Varianzanalysen über die drei Faktoren und der dazugehörigen Einzelvergleiche:

123 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 1  
 11:16:20 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7  
 0For MVS/SP 2.1.7 BERNISCHE DATENVERARBEITUNG AG License Number 1526

Use INFO OVERVIEW for more information on:  
 0\* INCLUDE - To bring in command files \* Improvements in:  
 \* RENAME VARS - To rename variables \* MANOVA  
 \* AUTORECODE - To recode strings as numbers \* TABLES  
 \* DROP DOCUMENTS \*

1 0 UNNUMBERED  
 2 0 SET LENGTH=NONE  
 3 0 DATA LIST FILE=IN NOTABLE /1 M 1 B 2 V 3-4 F1,F2,F3 5-28(5)  
 4 0 ANOVA F1,F2,F3 BY M(1,3) B(1,3)  
 5 0 STATISTICS 1,3  
 0'ANOVA' PROBLEM REQUIRES 1498 BYTES OF MEMORY.

23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 2  
 11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

0 \*\*\* CELL MEANS \*\*\*

0 F1  
 0 BY M  
 B  
 -TOTAL POPULATION  
 0 2.94  
 ( 108)

0M  
 0 1 2 3  
 3.05 2.93 2.83  
 ( 36) ( 36) ( 36)

0B  
 0 1 2 3  
 3.09 2.72 2.99  
 ( 36) ( 36) ( 36)

0 B 1 2 3  
 M  
 1 3.43 3.08 2.62  
 ( 12) ( 12) ( 12)  
 2 3.00 2.71 3.08  
 ( 12) ( 12) ( 12)  
 3 2.85 2.38 3.26  
 ( 12) ( 12) ( 12)

23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS

3

11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

0 \*\*\* CELL MEANS \*\*\*

0 F2  
BY M  
B  
-TOTAL POPULATION

0 3.35  
( 108)

OM

0 1 2 3  
( 4.04 ( 2.73 ( 3.28  
( 36) ( 36) ( 36)

OB

0 1 2 3  
( 3.40 ( 3.53 ( 3.11  
( 36) ( 36) ( 36)

0 B 1 2 3  
M 1 4.18 4.18 3.75  
( 12) ( 12) ( 12)  
2 3.06 2.69 2.43  
( 12) ( 12) ( 12)  
3 2.97 3.72 3.14  
( 12) ( 12) ( 12)

23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS

4

11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

0 \*\*\* CELL MEANS \*\*\*

0 F3  
BY M  
B  
-TOTAL POPULATION

0 2.87  
( 108)

OM

0 1 2 3  
( 2.83 ( 2.71 ( 3.06  
( 36) ( 36) ( 36)

OB

0 1 2 3  
( 2.98 ( 2.68 ( 2.94  
( 36) ( 36) ( 36)

0 B 1 2 3  
M 1 2.96 2.86 2.67  
( 12) ( 12) ( 12)  
2 2.47 2.89 2.76  
( 12) ( 12) ( 12)  
3 3.51 2.28 3.39  
( 12) ( 12) ( 12)

23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 5 11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

0 \*\*\* ANALYSIS OF VARIANCE \*\*\*

BY		F1				
		M				
		B				
SOURCE OF VARIATION			SUM OF SQUARES	DF	MEAN SQUARE	SIGNIF OF F
OMAIN EFFECTS			3.489	4	0.872	0.731 0.573
M			0.853	2	0.427	0.358 0.700
B			2.636	2	1.318	1.104 0.336
02-WAY INTERACTIONS			6.960	4	1.740	1.458 0.221
M B			6.960	4	1.740	1.458 0.221
OEXPLAINED			10.449	8	1.306	1.094 0.374
ORESIDUAL			118.153	99	1.193	
OTOTAL			128.602	107	1.202	

- 108 CASES WERE PROCESSED.  
 0 CASES ( 0.0 PCT) WERE MISSING.

23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 6 11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

0 \*\*\* MULTIPLE CLASSIFICATION ANALYSIS \*\*\*

BY		F1				
		M				
		B				
-GRAND MEAN =		2.94				
VARIABLE + CATEGORY	N	UNADJUSTED DEV'N	ETA	ADJUSTED FOR INDEPENDENTS DEV'N	BETA	ADJUSTED FOR INDEPENDENTS + COVARIATES DEV'N BETA
OM						
1	36	0.11		0.11		
2	36	0.00		0.00		
3	36	-0.11		-0.11		
			0.08		0.08	
OB						
1	36	0.16		0.16		
2	36	-0.21		-0.21		
3	36	0.06		0.06		
			0.14		0.14	
OMULTIPLE R SQUARED					0.027	
MULTIPLE R					0.165	

23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 7 11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

0 \*\*\* ANALYSIS OF VARIANCE \*\*\*

BY		F2				
		M				
		B				
SOURCE OF VARIATION			SUM OF SQUARES	DF	MEAN SQUARE	SIGNIF OF F
OMAIN EFFECTS			34.591	4	8.648	7.513 0.000
M			31.159	2	15.579	13.536 0.000
B			3.432	2	1.716	1.491 0.230
02-WAY INTERACTIONS			4.136	4	1.034	0.898 0.468
M B			4.136	4	1.034	0.898 0.468
OEXPLAINED			38.727	8	4.841	4.206 0.000
ORESIDUAL			113.947	99	1.151	
OTOTAL			152.674	107	1.427	

- 108 CASES WERE PROCESSED.  
 0 CASES ( 0.0 PCT) WERE MISSING.



23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

\*\*\* MULTIPLE CLASSIFICATION ANALYSIS \*\*\*

		BY F2 M B			
-GRAND MEAN =		3.35			
VARIABLE + CATEGORY	N	UNADJUSTED DEV'N	ETA	ADJUSTED FOR INDEPENDENTS DEV'N	BETA
OM					
1	36	0.69		0.69	
2	36	-0.62		-0.62	
3	36	-0.07		-0.07	
			0.45		0.45
OB					
1	36	0.06		0.06	
2	36	0.19		0.19	
3	36	-0.24		-0.24	
			0.15		0.15
MULTIPLE R SQUARED				0.227	
MULTIPLE R				0.476	

23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

\*\*\* ANALYSIS OF VARIANCE \*\*\*

		BY F3 M B			
SOURCE OF VARIATION	SUM OF SQUARES	DF	MEAN SQUARE	F	SIGNIF OF F
MAIN EFFECTS	4.279	4	1.070	1.060	0.381
M	2.302	2	1.151	1.141	0.324
B	1.977	2	0.988	0.979	0.379
02-WAY INTERACTIONS	10.762	4	2.691	2.666	0.037
M B	10.762	4	2.691	2.666	0.037
UNEXPLAINED	15.042	8	1.880	1.863	0.074
RESIDUAL	99.928	99	1.009		
TOTAL	114.970	107	1.074		

108 CASES WERE PROCESSED.  
 0 CASES ( 0.0 PCT) WERE MISSING.

23 JUN 88 SPSS-X RELEASE 2.2 FOR IBM/MVS  
 11:16:21 BERNISCHE DATENVERARBEITUNG AG IBM 3090-180 MVS/SP 2.1.7

\*\*\* MULTIPLE CLASSIFICATION ANALYSIS \*\*\*

		BY F3 M B			
-GRAND MEAN =		2.87			
VARIABLE + CATEGORY	N	UNADJUSTED DEV'N	ETA	ADJUSTED FOR INDEPENDENTS DEV'N	BETA
OM					
1	36	-0.04		-0.04	
2	36	-0.16		-0.16	
3	36	0.19		0.19	
			0.14		0.14
OB					
1	36	0.12		0.12	
2	36	-0.19		-0.19	
3	36	0.07		0.07	
			0.13		0.13
MULTIPLE R SQUARED				0.037	
MULTIPLE R				0.193	

1

OPRECEDING TASK REQUIRED 0.15 SECONDS CPU TIME; 0.85 SECONDS ELAPSED.

6 0 FINISH
6 COMMAND LINES READ.
0 ERRORS DETECTED.
0 WARNINGS ISSUED.
0 SECONDS CPU TIME.
2 SECONDS ELAPSED TIME.
END OF JOB.

END OF DATA

L

11 1 2.66667 4.83333 3.16667
11 2 3.66667 4.33333 2.66667
11 3 4.16667 4.83333 3.16667
11 4 3.83333 5.66667 1.33333
11 5 3.83333 4.50000 2.50000
11 6 4.66667 3.66667 2.66667
11 7 3.00000 4.50000 2.83333
11 8 2.16667 4.50000 3.16667
11 9 4.50000 1.66667 4.66667
1110 4.16667 5.16667 2.33333
1111 1.33333 3.16667 2.83333
1112 3.16667 3.33333 4.16667
12 1 2.00000 3.16667 1.50000
12 2 3.16667 3.16667 2.16667
12 3 4.83333 5.33333 2.00000
12 4 2.50000 4.33333 1.66667
12 5 1.50000 5.16667 2.33333
12 6 3.83333 5.50000 4.33333
12 7 4.16667 3.66667 4.83333
12 8 2.50000 2.33333 3.16667
12 9 5.16667 4.16667 5.16667
1210 2.16667 5.66667 1.66667
1211 1.66667 4.50000 2.33333
1212 3.50000 3.16667 3.16667
13 1 1.50000 4.00000 3.33333
13 2 3.00000 4.16667 2.83333
13 3 3.33333 2.00000 2.00000
13 4 3.66667 2.66667 3.00000
13 5 1.16667 5.83333 1.33333
13 6 2.33333 4.00000 2.83333
13 7 1.83333 1.83333 1.50000
13 8 4.66667 3.83333 2.50000
13 9 1.50000 2.50000 5.83333
1310 2.50000 5.16667 2.16667
1311 4.00000 4.50000 2.33333
1312 2.00000 4.50000 2.33333
21 1 2.16667 2.66667 2.00000
21 2 4.66667 1.66667 3.50000
21 3 3.50000 3.50000 2.83333
21 4 3.16667 4.00000 2.83333
21 5 1.00000 4.00000 1.66667
21 6 2.66667 2.16667 2.16667
21 7 2.50000 2.00000 1.66667
21 8 3.66667 2.16667 2.66667
21 9 3.50000 3.66667 3.16667
2110 3.66667 2.33333 2.66667
2111 3.50000 4.33333 2.16667
2112 2.00000 4.16667 2.33333
22 1 3.50000 2.00000 3.83333
22 2 2.66667 1.66667 3.50000
22 3 1.83333 3.83333 1.66667
22 4 1.50000 2.66667 2.33333
22 5 2.83333 3.00000 3.83333
22 6 2.66667 2.50000 2.16667
22 7 1.16667 2.16667 1.33333
22 8 2.83333 2.16667 4.00000
22 9 3.00000 2.83333 3.00000
2210 4.16667 2.66667 2.33333
2211 3.33333 3.16667 3.33333
2212 3.00000 3.66667 3.33333

23 1 3.66667 1.33333 4.50000
23 2 2.83333 2.00000 2.00000
23 3 3.66667 2.50000 2.16667
23 4 2.00000 4.16667 2.83333
23 5 2.00000 2.33333 2.16667
23 6 3.33333 3.83333 3.16667
23 7 1.83333 1.66667 4.66667
23 8 3.50000 1.16667 3.83333
23 9 4.66667 2.66667 2.66667
2310 1.33333 4.00000 1.16667
2311 5.00000 1.83333 1.33333
2312 3.16667 1.66667 2.66667
31 1 2.33333 1.66667 4.83333
31 2 2.66667 2.00000 2.16667
31 3 2.66667 5.00000 3.50000
31 4 2.33333 4.00000 2.33333
31 5 2.00000 2.66667 3.33333
31 6 1.66667 5.83333 2.00000
31 7 2.83333 2.16667 3.33333
31 8 5.33333 1.50000 1.66667
31 9 5.16667 3.66667 5.50000
3110 1.50000 2.16667 5.16667
3111 3.83333 2.83333 4.50000
3112 1.83333 2.16667 3.83333
32 1 4.16667 4.83333 3.00000
32 2 2.16667 4.33333 2.83333
32 3 2.83333 3.16667 2.16667
32 4 2.16667 2.83333 2.00000
32 5 1.16667 5.50000 1.00000
32 6 2.50000 4.33333 2.83333
32 7 2.00000 2.50000 1.83333
32 8 4.16667 3.16667 3.16667
32 9 1.00000 3.66667 1.83333
3210 1.50000 3.16667 1.50000
3211 2.00000 3.50000 2.00000
3212 2.83333 3.66667 3.16667
33 1 3.16667 2.33333 4.50000
33 2 2.66667 3.50000 3.83333
33 3 3.66667 1.33333 3.00000
33 4 5.33333 5.00000 3.33333
33 5 2.83333 4.00000 2.66667
33 6 4.50000 4.16667 2.50000
33 7 3.16667 4.16667 3.66667
33 8 2.33333 2.83333 2.33333
33 9 5.16667 2.50000 3.33333
3310 2.33333 1.33333 4.83333
3311 2.00000 2.83333 4.50000
3312 2.00000 3.66667 2.16667
END OF DATA

SCHEFFE-TESTS DER ANOVA ÜBER DEN FAKTOR "AKTIVITÄT"

a) Faktorstufenkombinationen:

Mittelwertsdifferenzen der Faktorstufenkombinationen:

	AUDIO:		A/V MATCH:			A/V MISMATCH:		
	Musik 2	3	1	2	3	1	2	3
Musik 1	0.43	0.58	0.35	0.72	1.05	0.81	0.35	0.17
AUDIO: Musik 2	0.15		-0.08	0.29	0.62	0.38	-0.08	-0.26
Musik 3			-0.23	0.14	0.47	0.23	-0.23	-0.41
A/V MATCH :			Musik 1	0.37	0.70	0.46	0.00	-0.18
			Musik 2	0.33	0.09	-0.37	-0.55	
			Musik 3			-0.24	-0.70	-0.88
A/V MISMATCH						Musik 1	-0.46	-0.64
						Musik 2		-0.18

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 1.258  
 $\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 1.396      -> \*  
 $\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 1.664      -> \*\*

b) Faktor "Musik":

Mittelwertsdifferenzen des Faktors "Musik":

	Musik 2	Musik 3
Musik 1	0.12	0.22
Musik 2		0.10

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 0.558  
 $\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 0.638      -> \*  
 $\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 0.797      -> \*\*

c) Faktor "Präsentationsbedingungen":

Mittelwertsdifferenzen des Faktors "Präsentationsbedingungen":

	Audio	A/V match
A/V mismatch	-0.10	0.27
Audio		0.37

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 0.558  
 $\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 0.638      -> \*  
 $\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 0.797      -> \*\*

SCHEFFE-TESTS DER ANOVA ÜBER DEN FAKTOR "MACHO"

a) Faktorstufenkombinationen:

Mittelwertsdifferenzen der Faktorstufenkombinationen:

	AUDIO:		A/V MATCH:			A/V MISMATCH:		
	Musik 2	Musik 3	Musik 1	Musik 2	Musik 3	Musik 1	Musik 2	Musik 3
Musik 1	1.12	1.21	0.00	1.49*	0.46	0.43	1.75**	1.04
AUDIO: Musik 2	0.09		-1.12	0.37	-0.66	-0.69	0.63	-0.08
Musik 3			-1.21	0.28	-0.75	-0.78	0.54	-0.17
A/V MATCH :			Musik 1	1.49*	0.46	0.43	1.75**	1.04
			Musik 2		-1.03	-1.06	0.26	-0.45
			Musik 3			-0.03	1.29	0.58
A/V MISMATCH						Musik 1	1.32	0.61
						Musik 2		-0.71

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 1.235

$\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 1.371 -> \*

$\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 1.624 -> \*\*

b) Faktor "Musik":

Mittelwertsdifferenzen des Faktors "Musik":

	Musik 2	Musik 3
Musik 1	1.31**	0.76*
Musik 2		-0.55

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 0.548

$\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 0.627 -> \*

$\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 0.783 -> \*\*

c) Faktor "Präsentationsbedingungen":

Mittelwertsdifferenzen des Faktors "Präsentationsbedingungen":

	Audio	A/V match
A/V mismatch	-0.29	-0.42
Audio		-0.13

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 0.558

$\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 0.638 -> \*

$\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 0.797 -> \*\*

SCHEFFE-TESTS DER ANOVA ÜBER DEN FAKTOR "HARMONIE"

a) Faktorstufenkombinationen:

Mittelwertsdifferenzen der Faktorstufenkombinationen:

	AUDIO:		A/V MATCH:			A/V MISMATCH:		
	Musik 2	Musik 3	Musik 1	Musik 2	Musik 3	Musik 1	Musik 2	Musik 3
Musik 1	0.49	-0.55	0.10	0.07	0.68	0.29	0.20	-0.43
AUDIO: Musik 2	-1.04		-0.39	-0.42	0.19	-0.20	-0.29	-0.92
Musik 3			0.65	0.62	1.23	0.84	0.75	0.12
A/V MATCH : Musik 1			-0.03	0.58		0.19	0.10	-0.53
Musik 2				0.61		0.22	0.13	-0.50
Musik 3						-0.39	-0.48	-1.11
A/V MISMATCH Musik 1							-0.09	-0.72
Musik 2								-0.63

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 1.157  
 $\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 1.284      -> \*  
 $\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 1.530      -> \*\*

b) Faktor "Musik":

Mittelwertsdifferenzen des Faktors "Musik":

	Musik 2	Musik 3
Musik 1	0.12	-0.23
Musik 2		-0.35

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 0.513  
 $\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 0.587      -> \*  
 $\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 0.733      -> \*\*

c) Faktor "Präsentationsbedingungen":

Mittelwertsdifferenzen des Faktors "Präsentationsbedingungen":

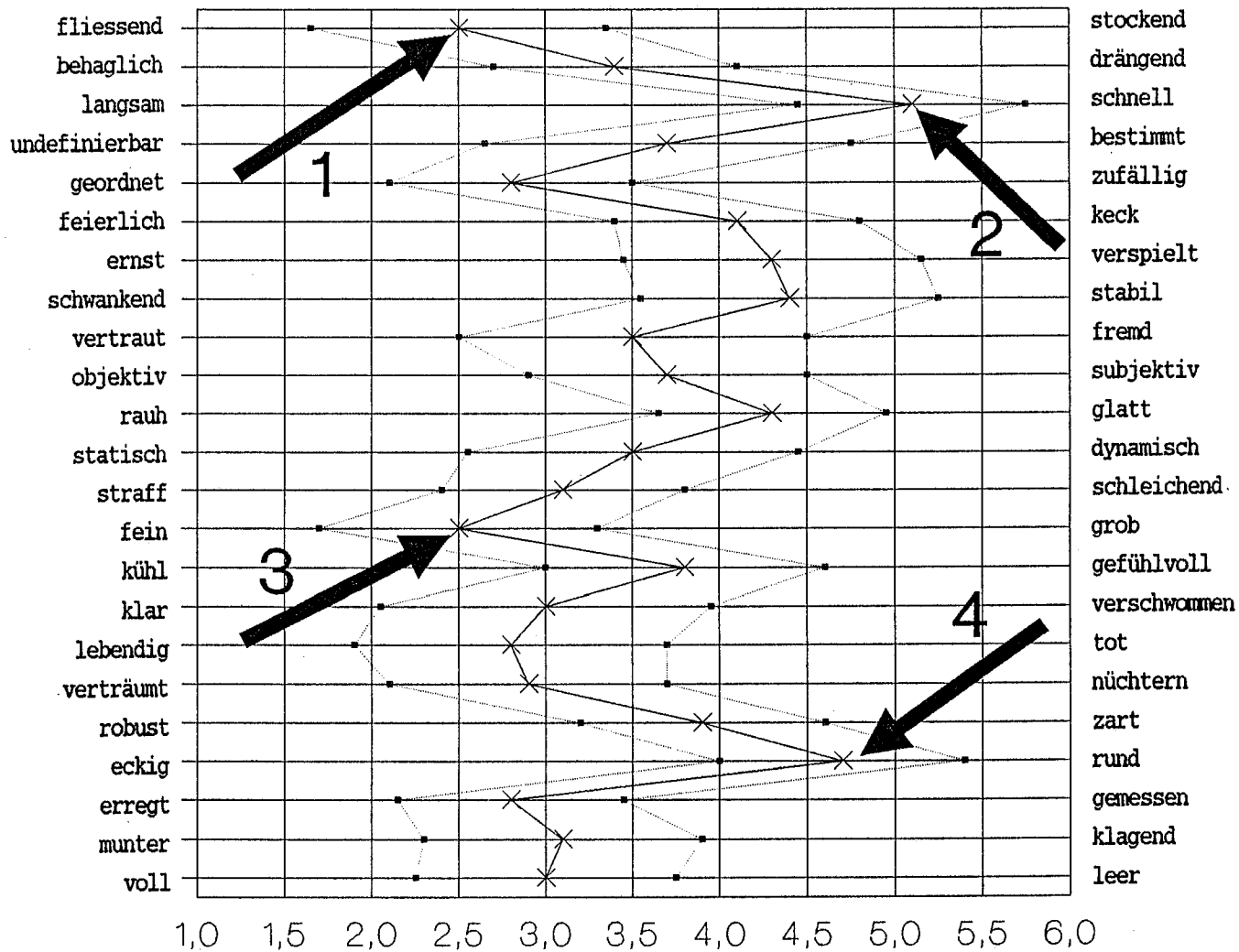
	Audio	A/V match
A/V mismatch	-0.04	0.26
Audio		0.30

$\alpha$ -Niveau 10%: Diff<sub>crit</sub> = 0.513  
 $\alpha$ -Niveau 5%: Diff<sub>crit</sub> = 0.587      -> \*  
 $\alpha$ -Niveau 1%: Diff<sub>crit</sub> = 0.733      -> \*\*

12.9. restliche 8 Mittelwertsprofile der 9 Faktorstufen-  
kombinationen:

# Mittelwertsprofil Nr. 2: Musik Nr. 1, mit passendem Video

Items

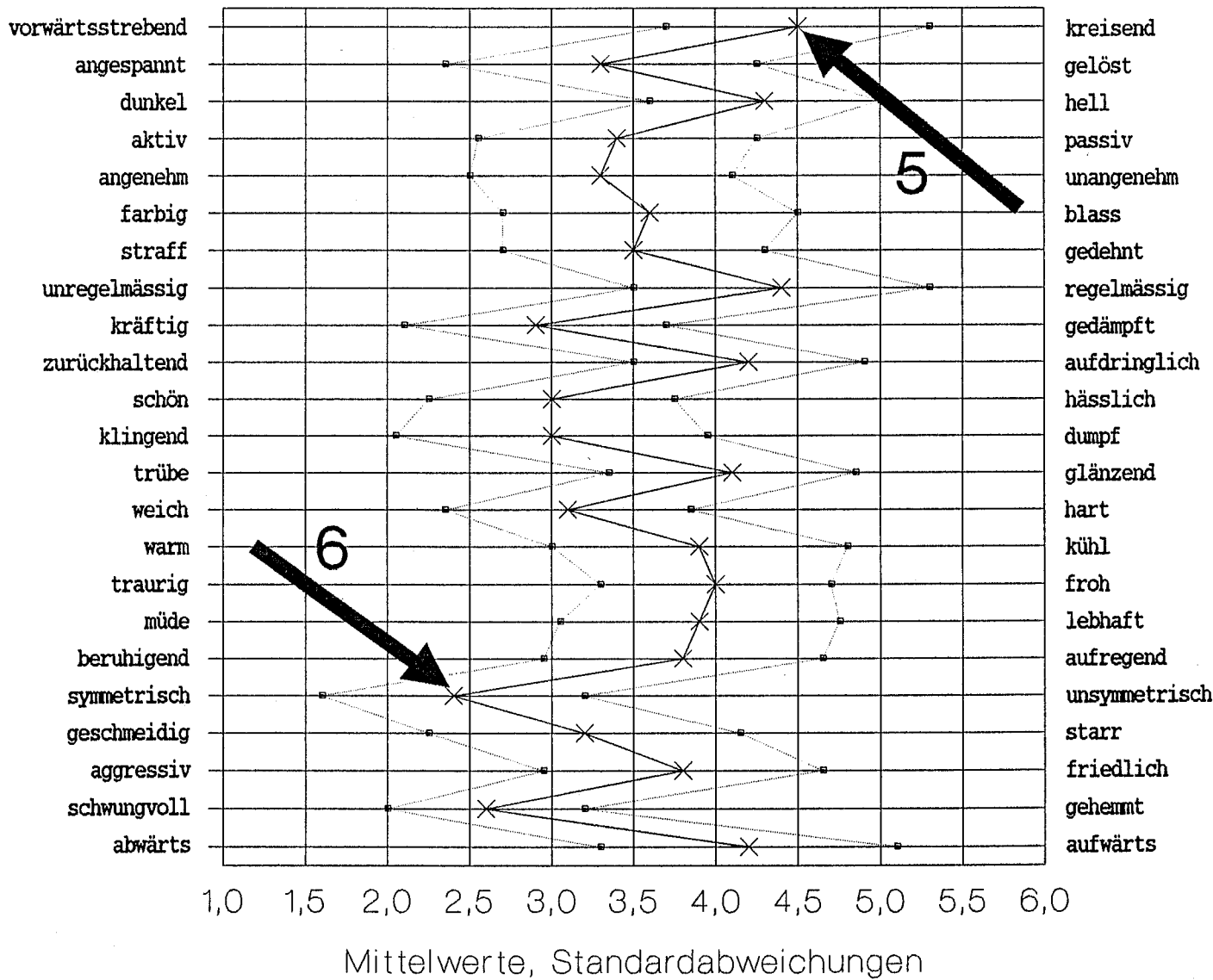


Mittelwerte, Standardabweichungen  
**fliegend (1), schnell (2), fein (3),  
 rund (4)**

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 2: Musik Nr. 1, mit passendem Video

Items



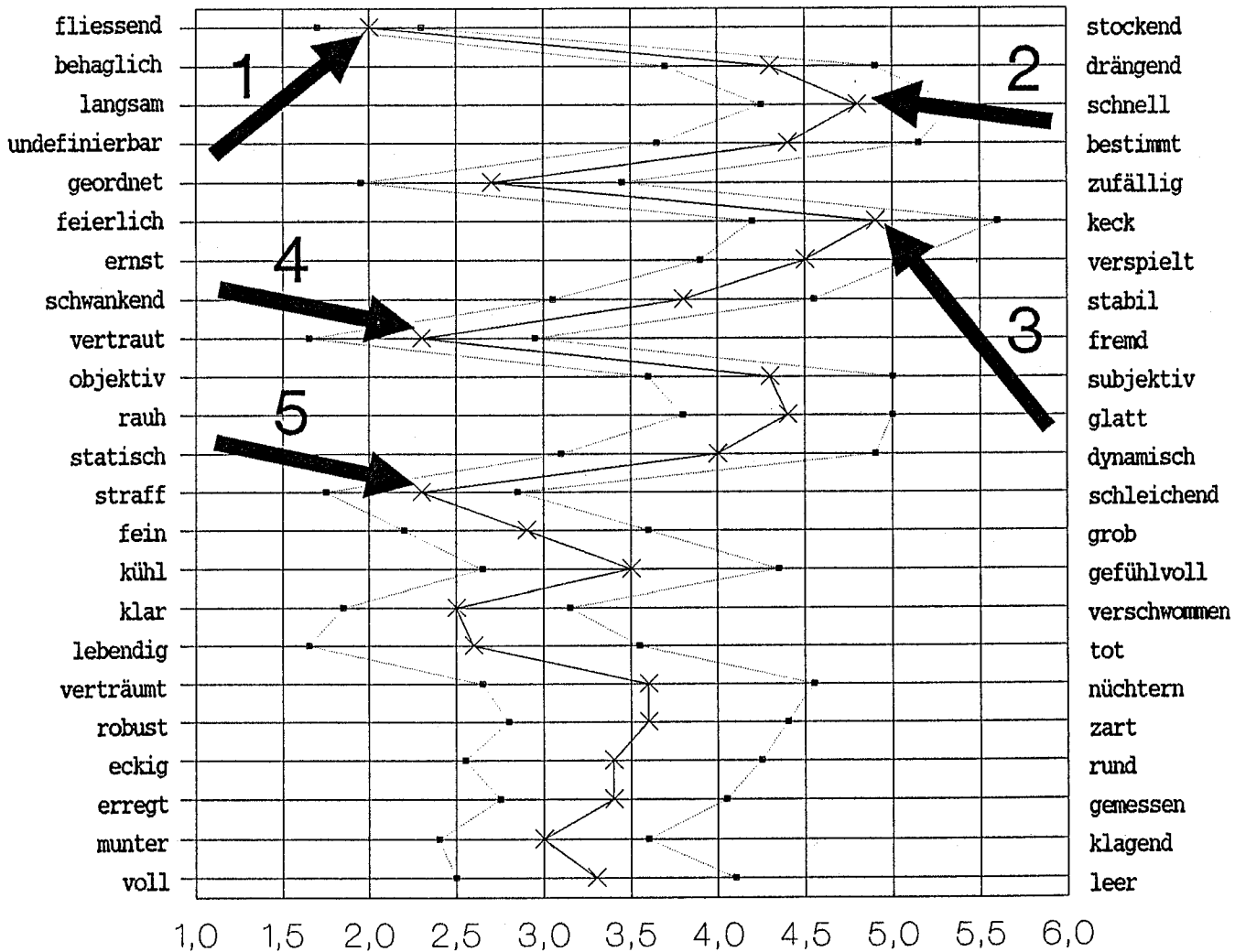
**kreisend (5), symmetrisch (6)**

Seite 2 (Items 24 bis 46)



# Mittelwertsprofil Nr. 3: Musik Nr. 1, mit Zufallsvideo

Items

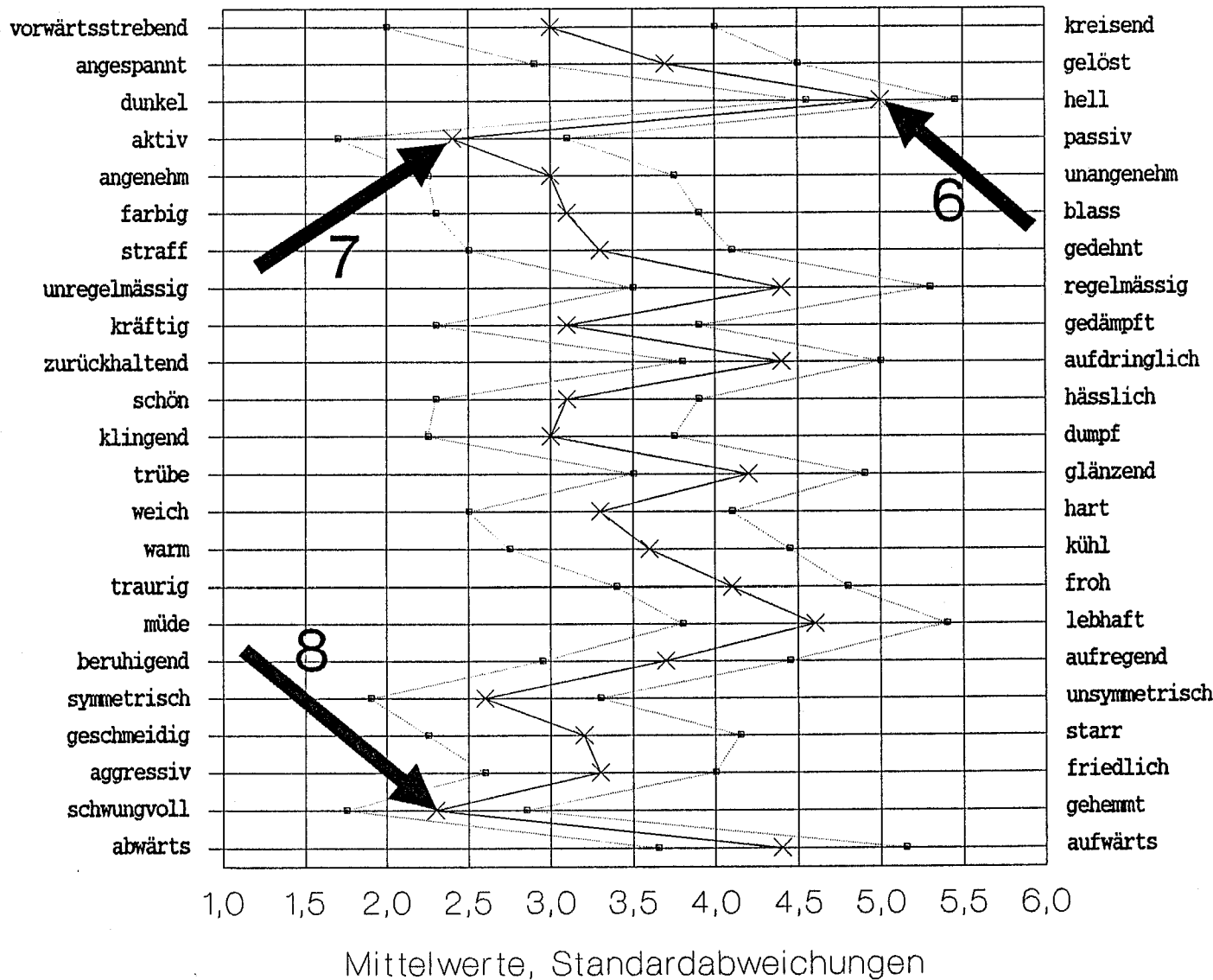


Mittelwerte, Standardabweichungen  
**fließend (1), schnell (2), keck (3),  
 vertraut (4), straff (5)**

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 3: Musik Nr. 1, mit Zufallsvideo

Items

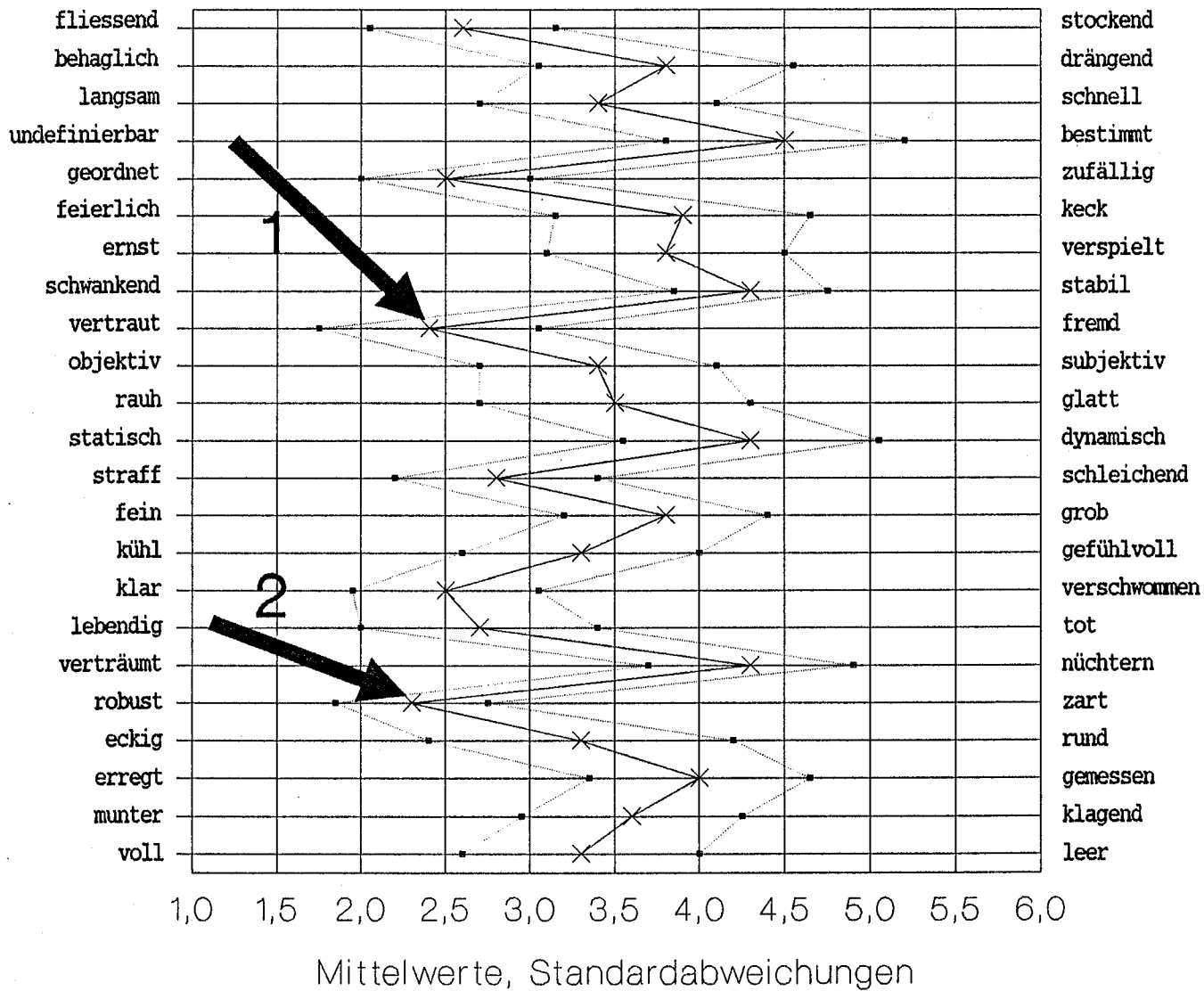


hell (6), aktiv (7), schwungvoll (8)

Seite 2 (Items 24 bis 46)

# Mittelwertsprofil Nr. 4: Musik Nr. 2, nur Audio

Items

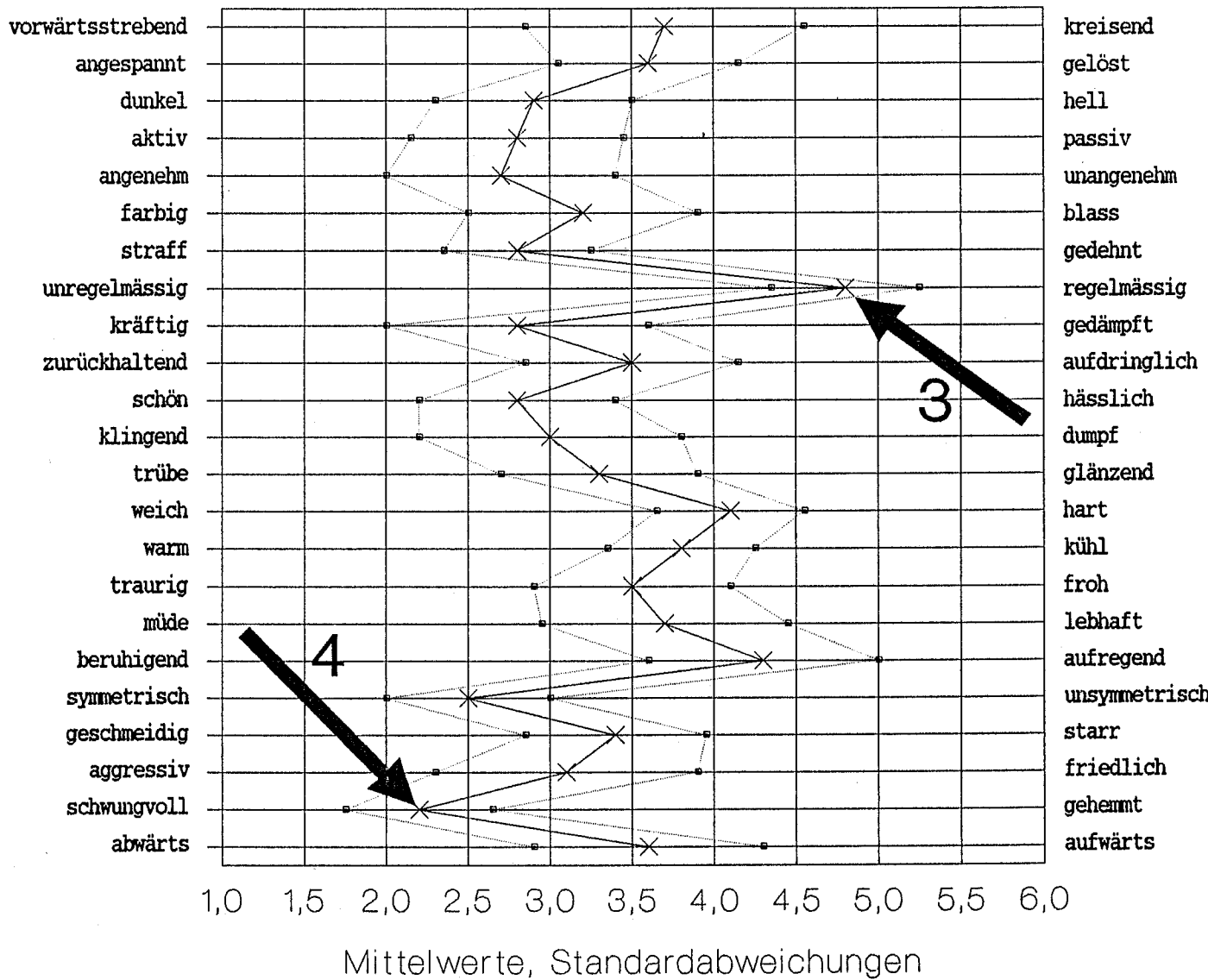


vertraut (1), robust (2)

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 4: Musik Nr. 2, nur Audio

Items

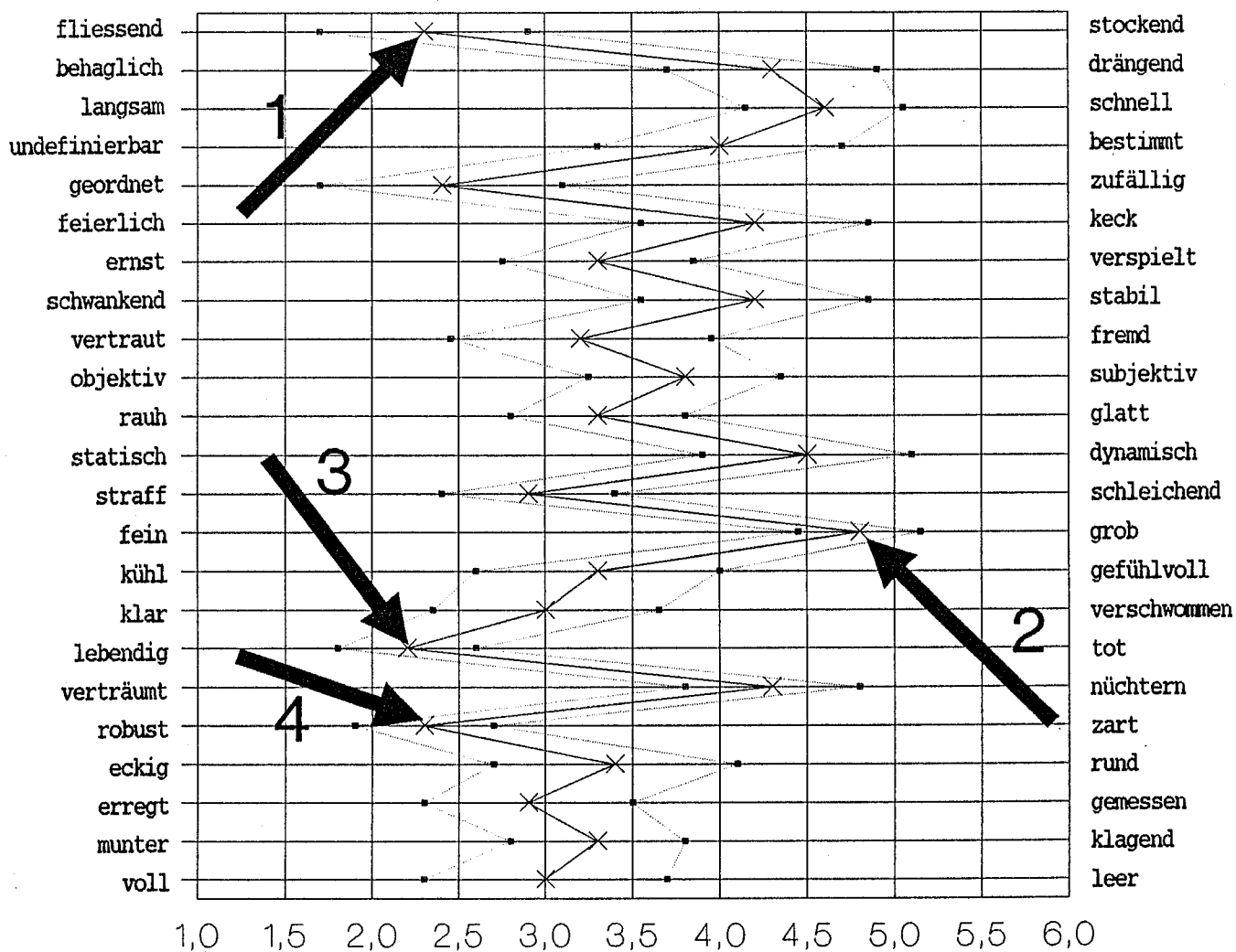


**regelmässig (3), schwungvoll (4)**

Seite 2 (Items 24 bis 46)

# Mittelwertsprofil Nr. 5: Musik Nr. 2, mit passendem Video

Items

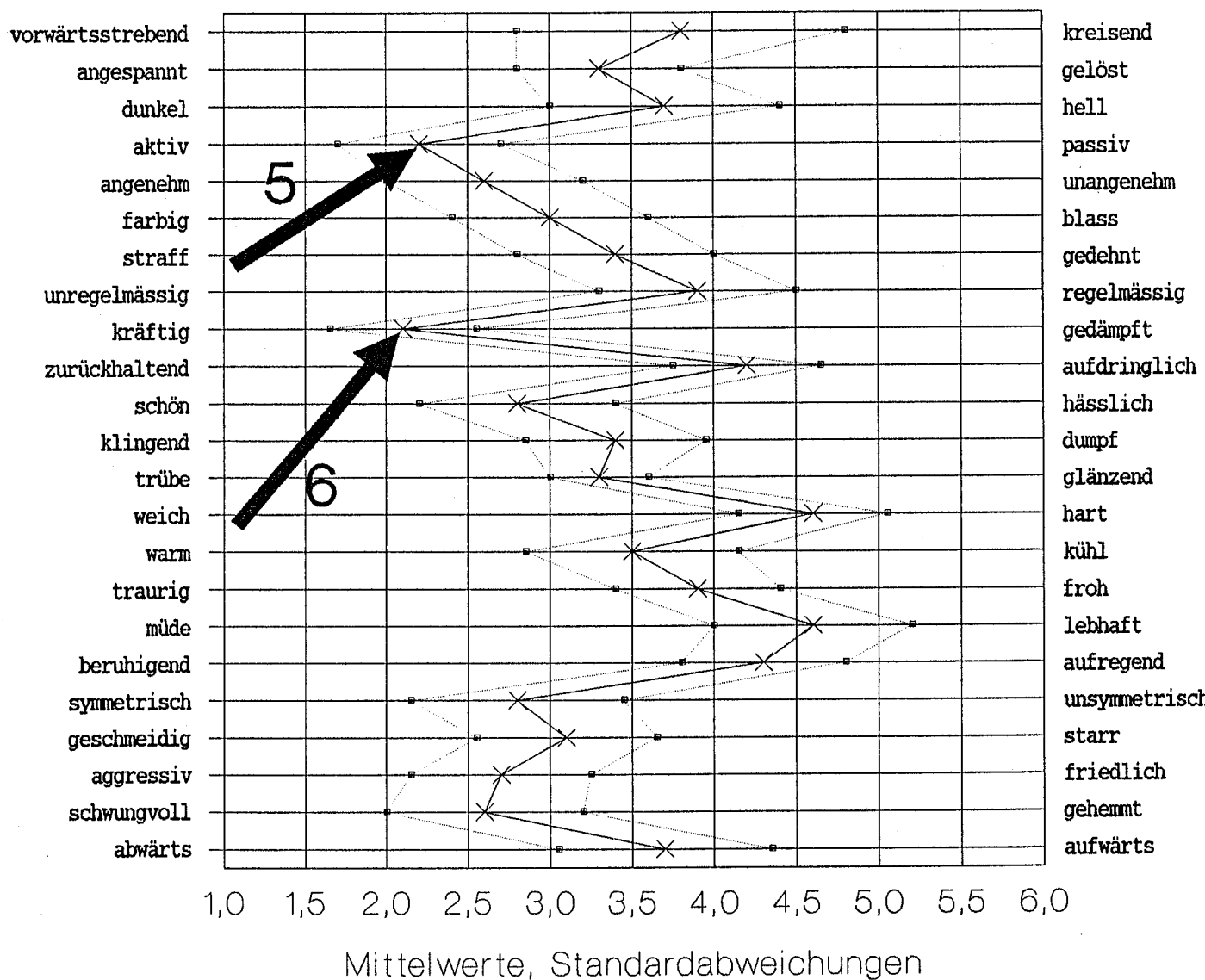


Mittelwerte, Standardabweichungen  
**fliegend (1), grob (2), lebendig (3),  
 robust (4)**

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 5: Musik Nr. 2, mit passendem Video

Items

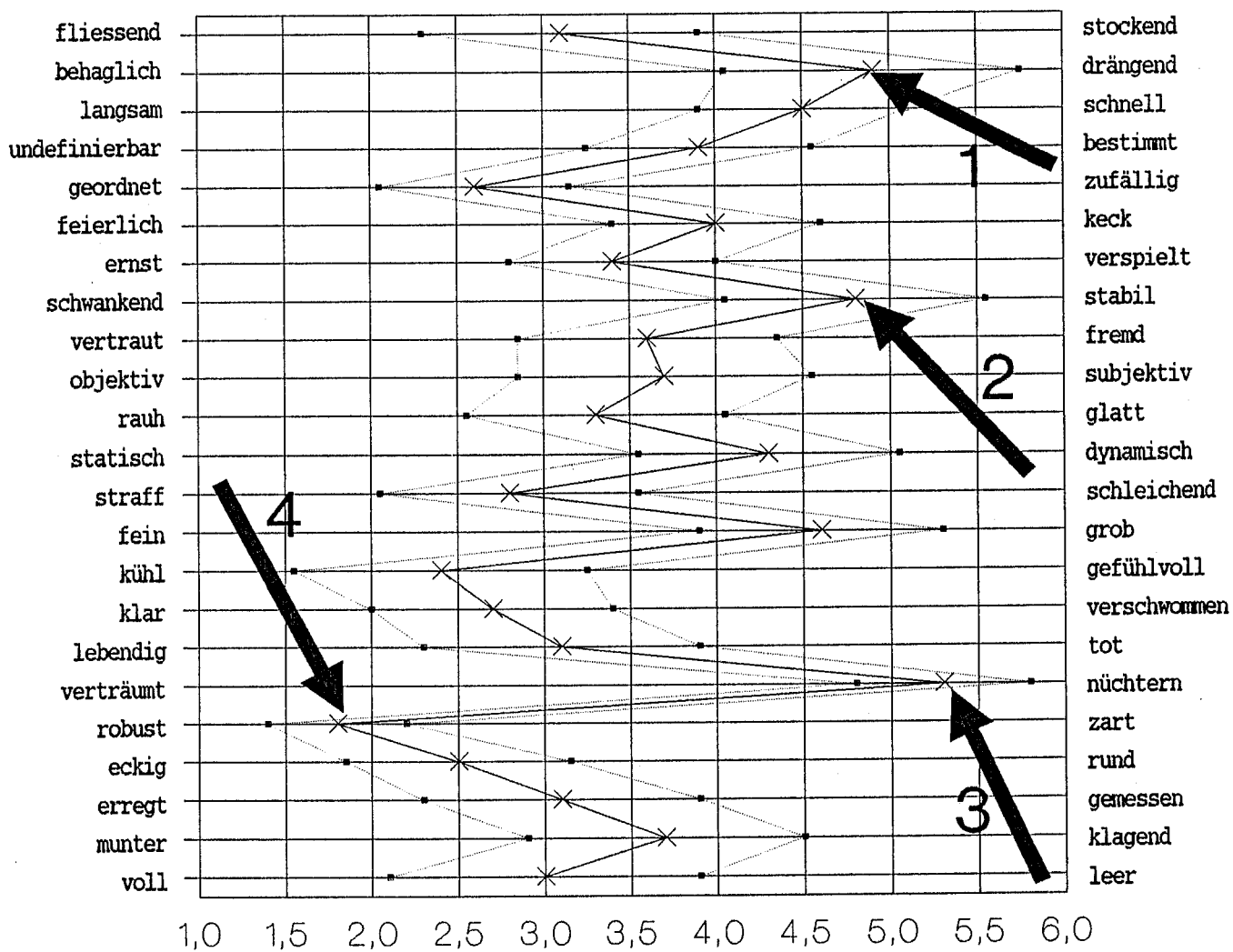


**aktiv (5), kräftig (6)**

Seite 2 (Items 24 bis 46)

# Mittelwertsprofil Nr. 6: Musik Nr. 2, mit Zufallsvideo

Items



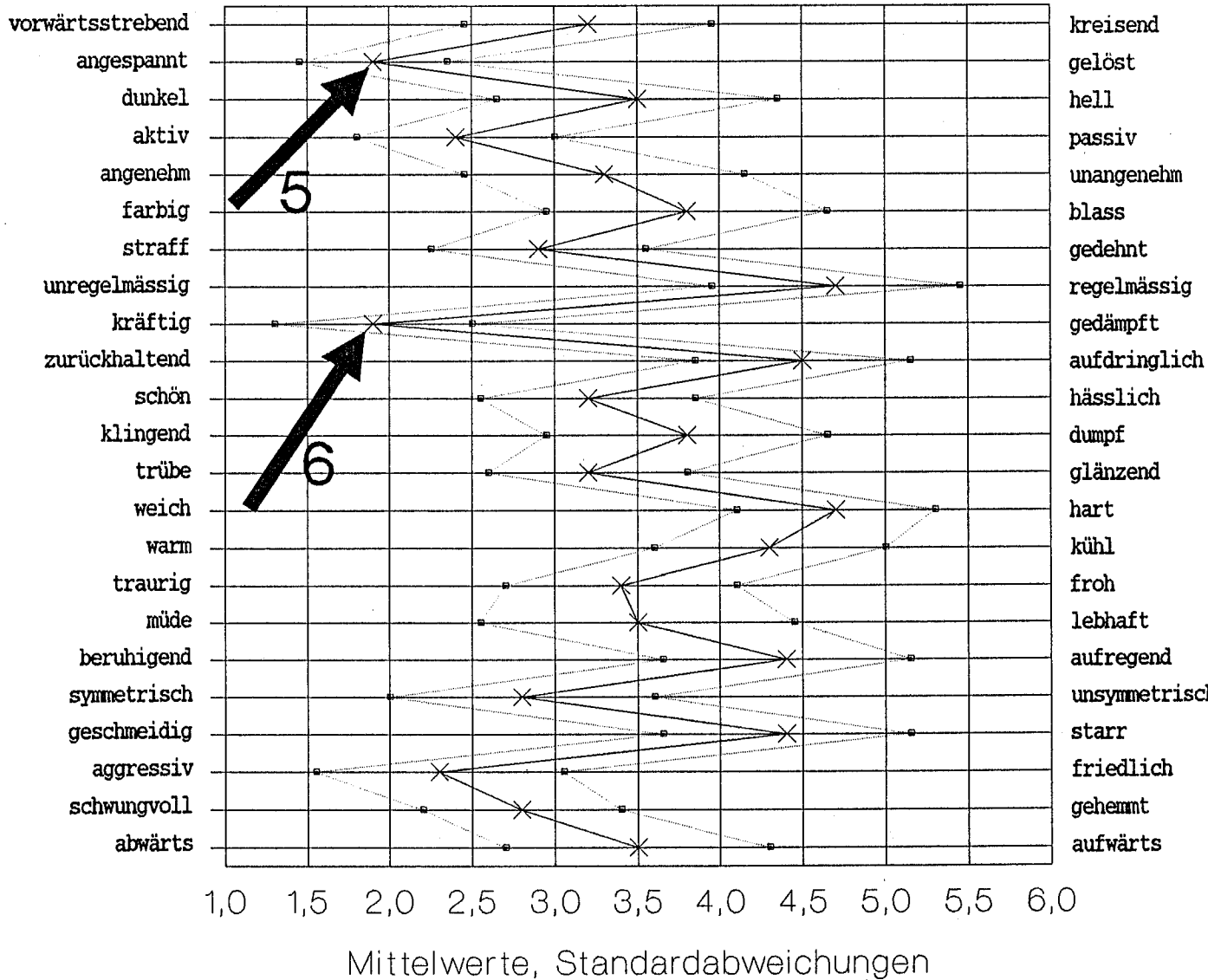
Mittelwerte, Standardabweichungen

**drängend (1), stabil (2),  
nüchtern (3), robust (4)**

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 6: Musik Nr. 2, mit Zufallsvideo

Items

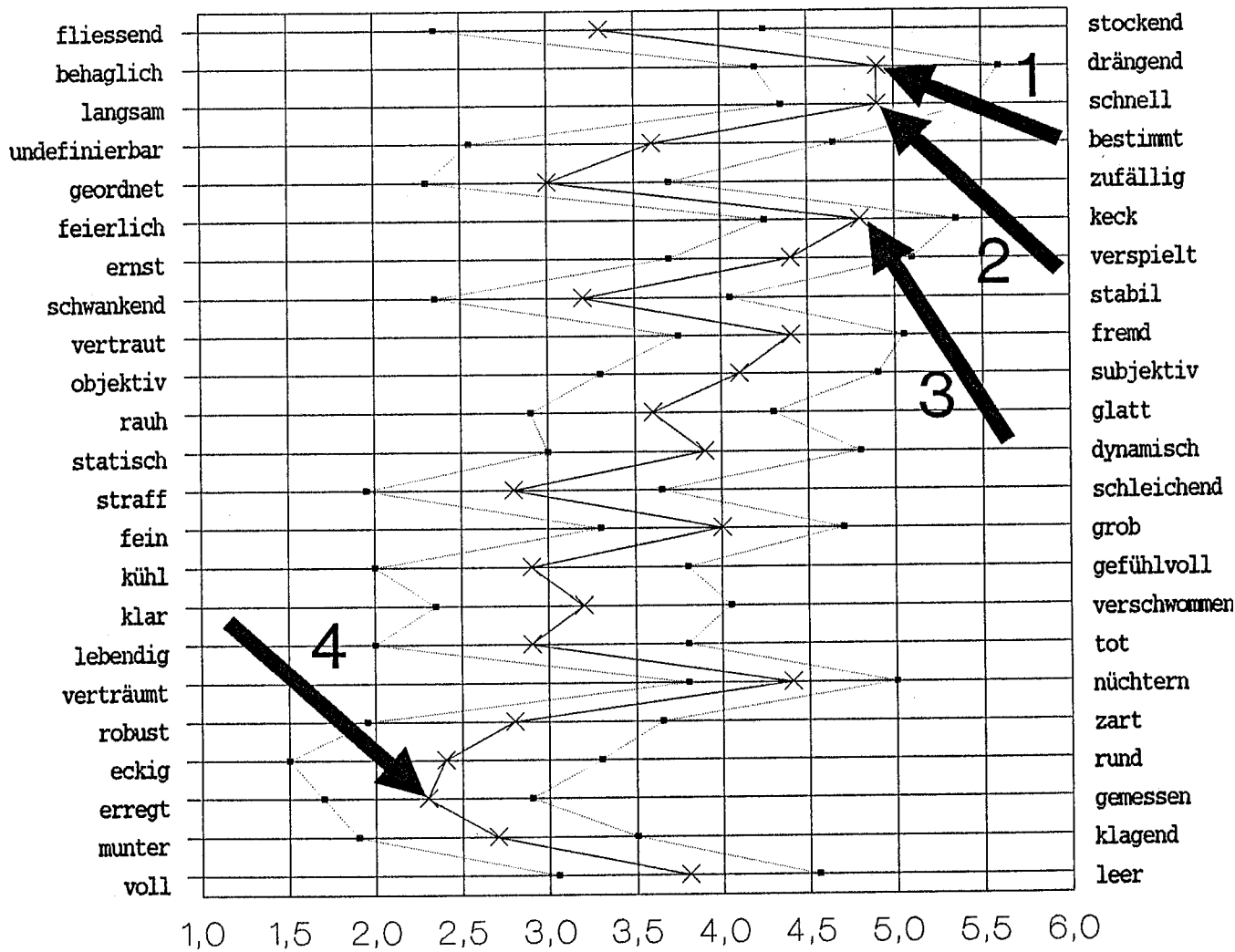


angespannt (5), kräftig (6)



# Mittelwertsprofil Nr. 7: Musik Nr. 3, nur Audio

Items



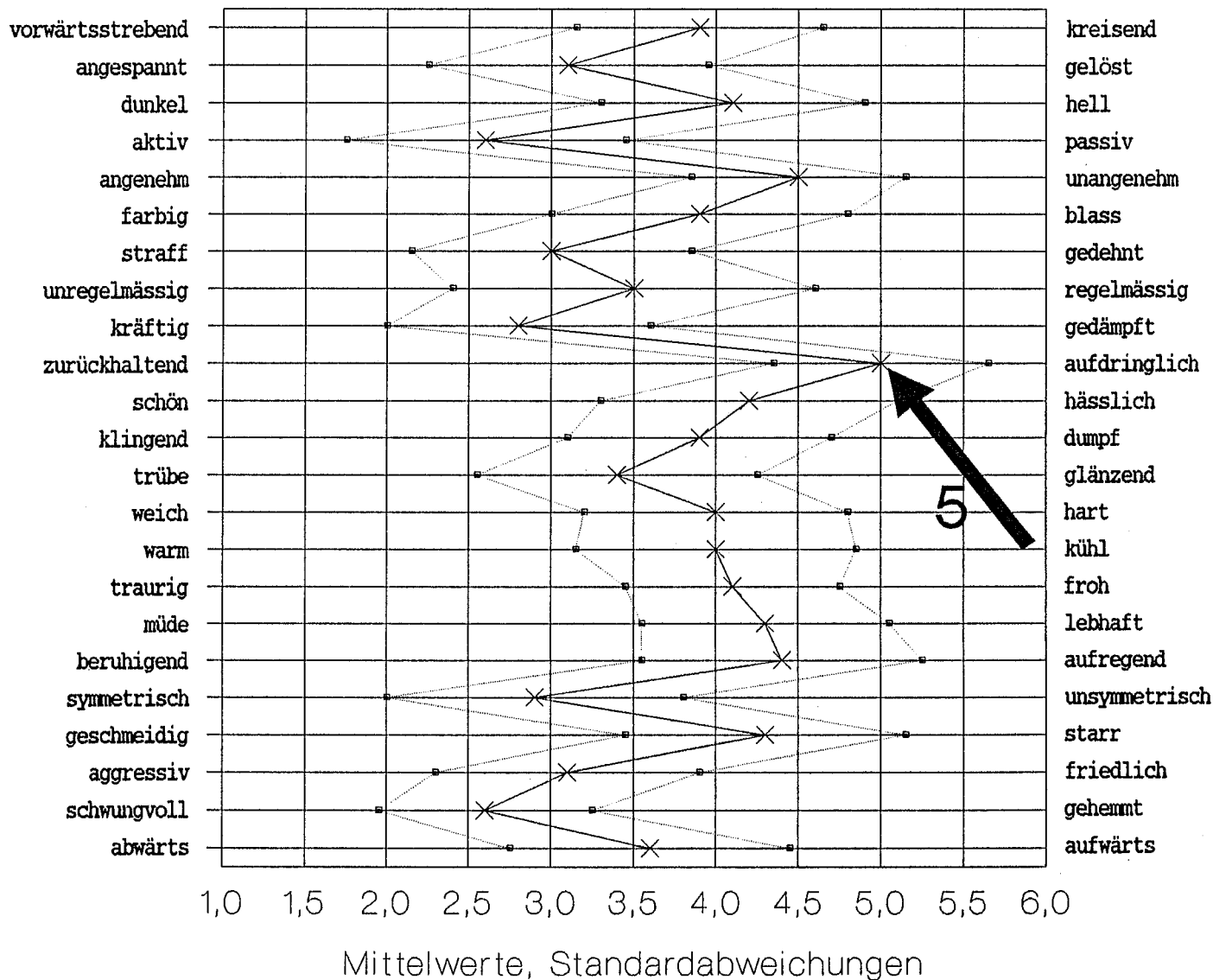
Mittelwerte, Standardabweichungen

**drängend (1), schnell (2),  
keck (3), erregt (4)**

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 7: Musik Nr. 3, nur Audio

Items

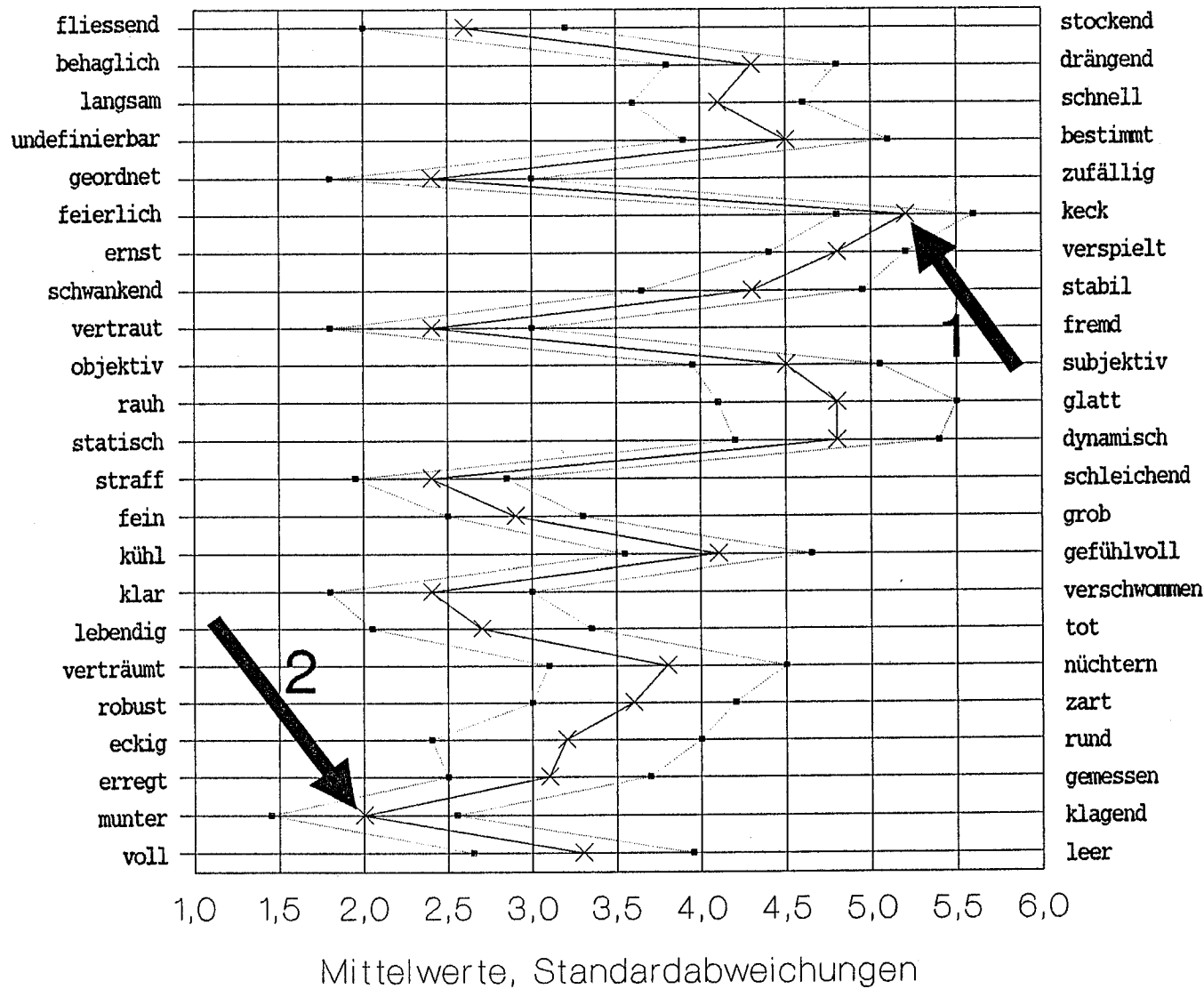


**aufdringlich (5)**

Seite 2 (Items 24 bis 46)

# Mittelwertsprofil Nr. 8: Musik Nr. 3, mit passendem Video

Items

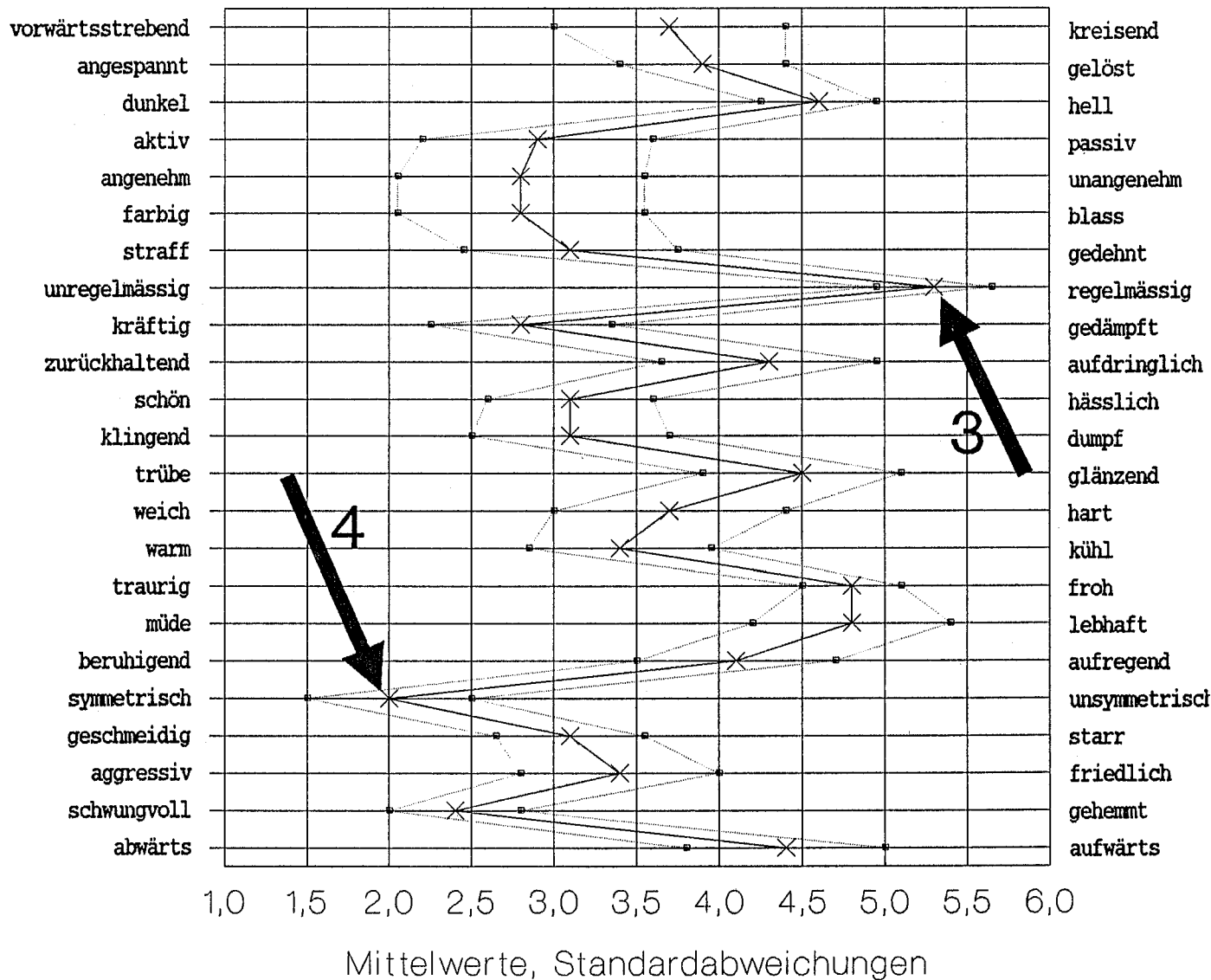


keck (1), munter (2)

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 8: Musik Nr. 3, mit passendem Video

Items

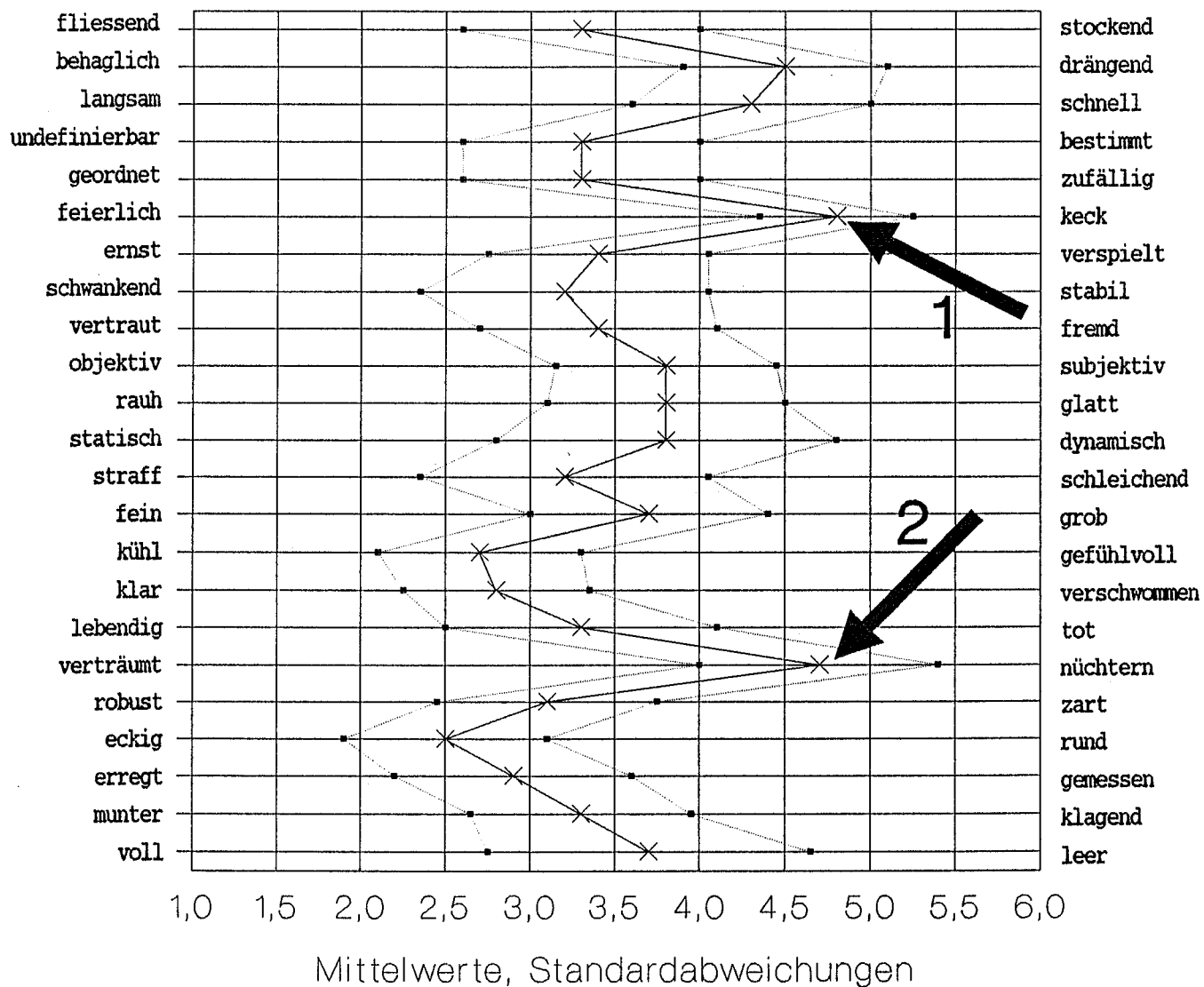


regelmässig (3), symmetrisch (4)

Seite 2 (Items 24 bis 46)

# Mittelwertsprofil Nr. 9: Musik Nr. 3, mit Zufallsvideo

Items

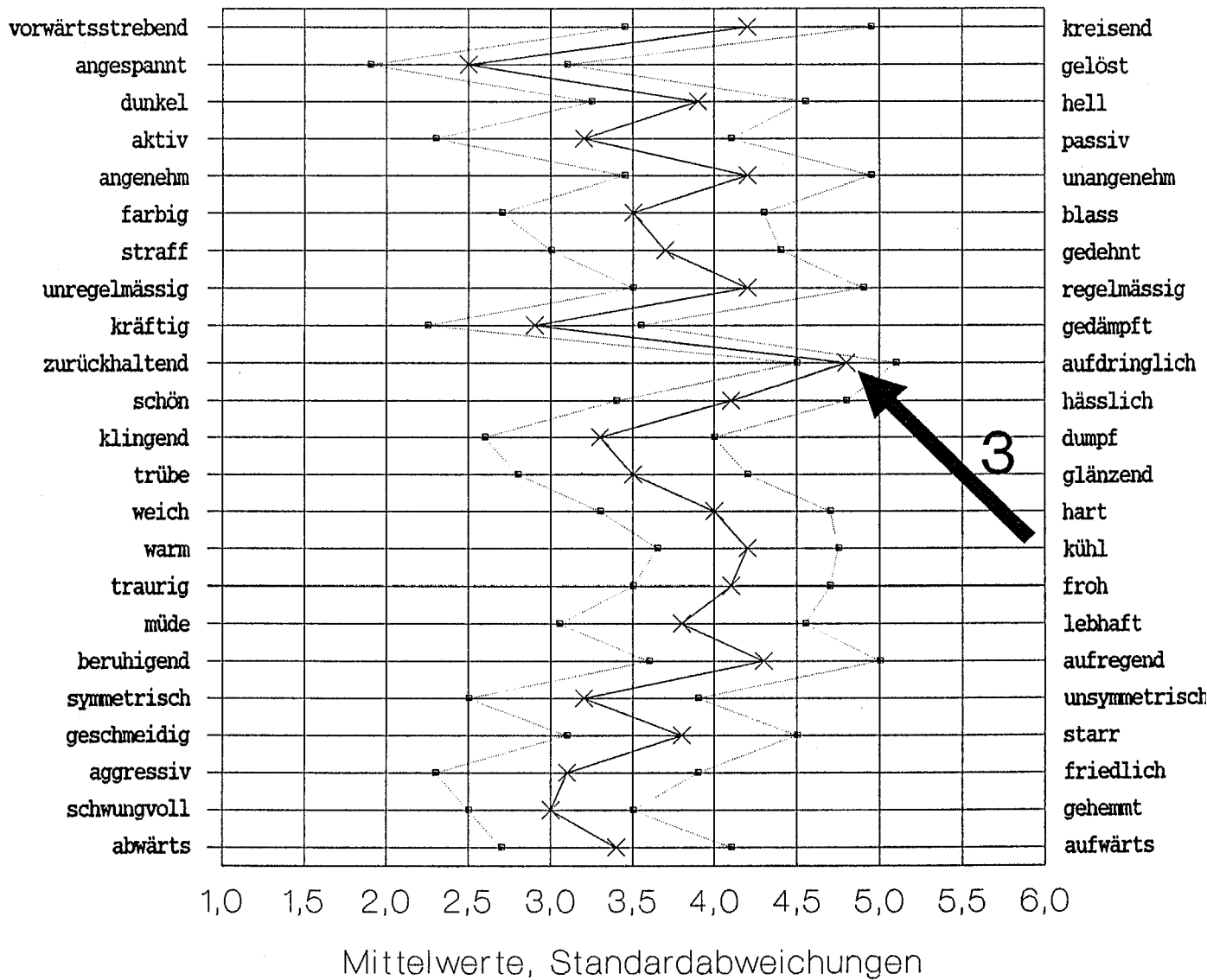


keck (1), nüchtern (2)

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 9: Musik Nr. 3, mit Zufallsvideo

Items

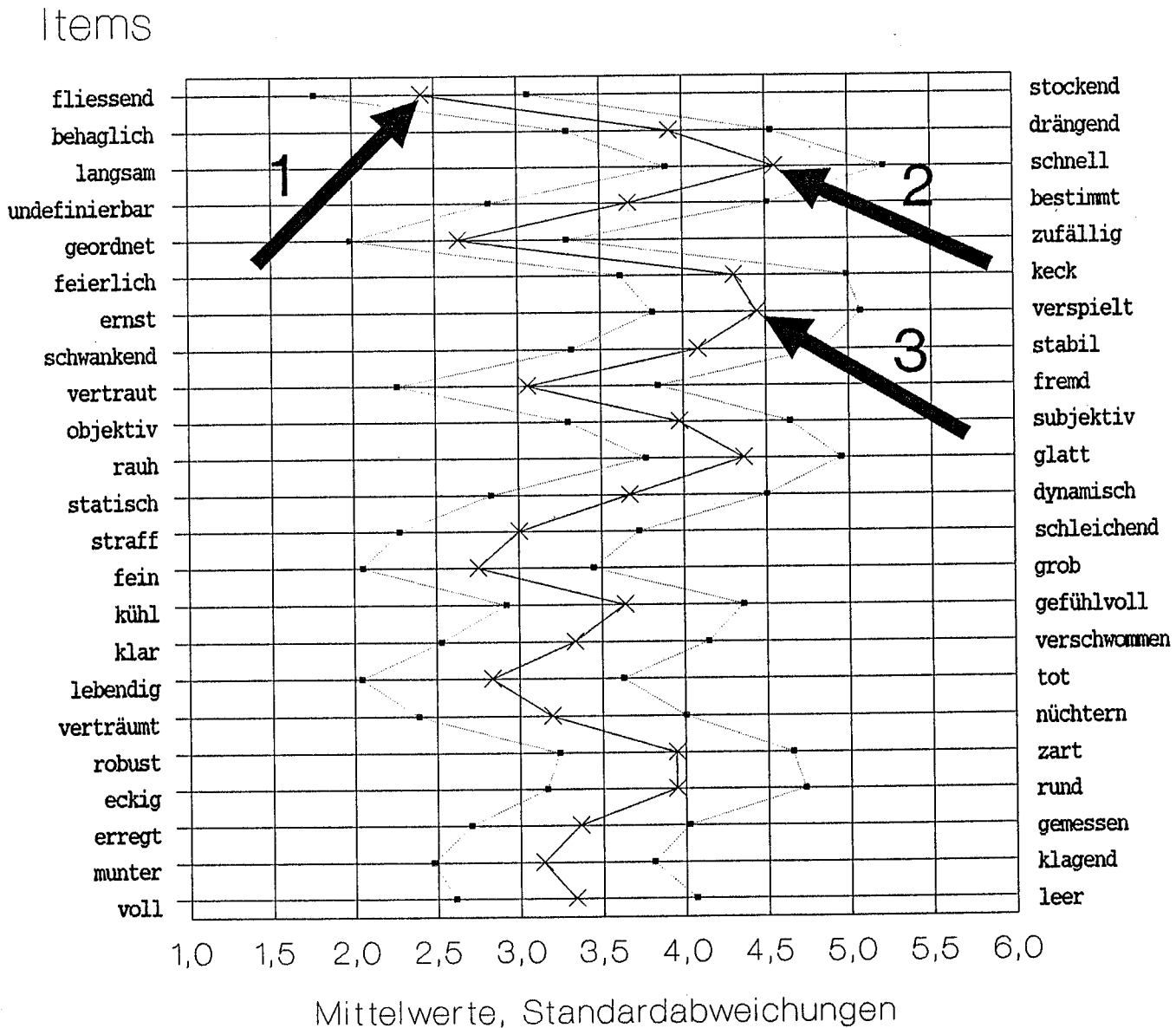


**aufdringlich (3)**

Seite 2 (Items 24 bis 46)

12.10. restliche 5 Mittelwertsprofile der 6 Faktorstufen:

# Mittelwertsprofil Nr. 10: Mylène Farmer (Musik Nr. 1)

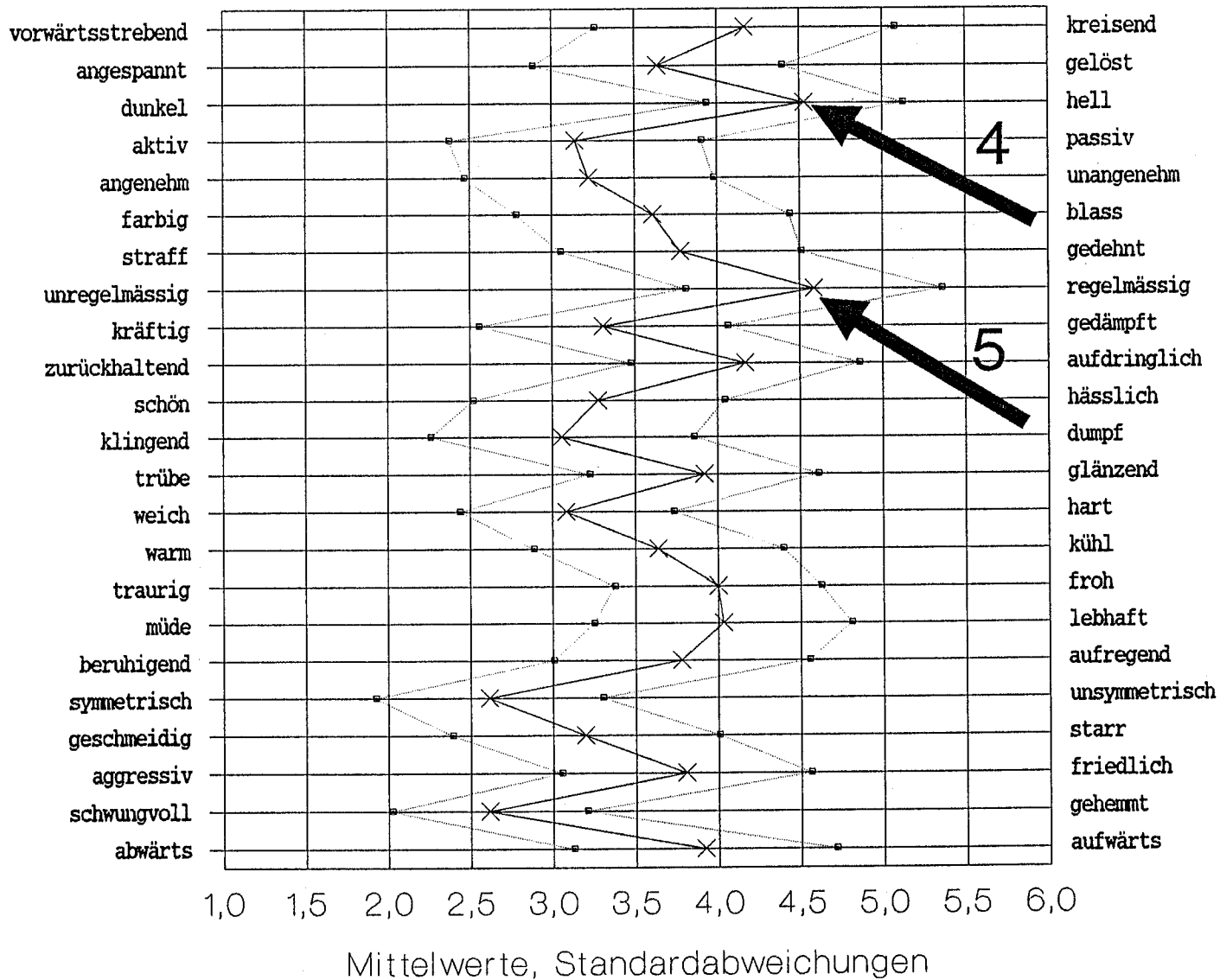


fließend (1), schnell (2), verspielt (3)



# Forts. Mittelwertsprofil Nr. 10: Mylène Farmer (Musik Nr. 1)

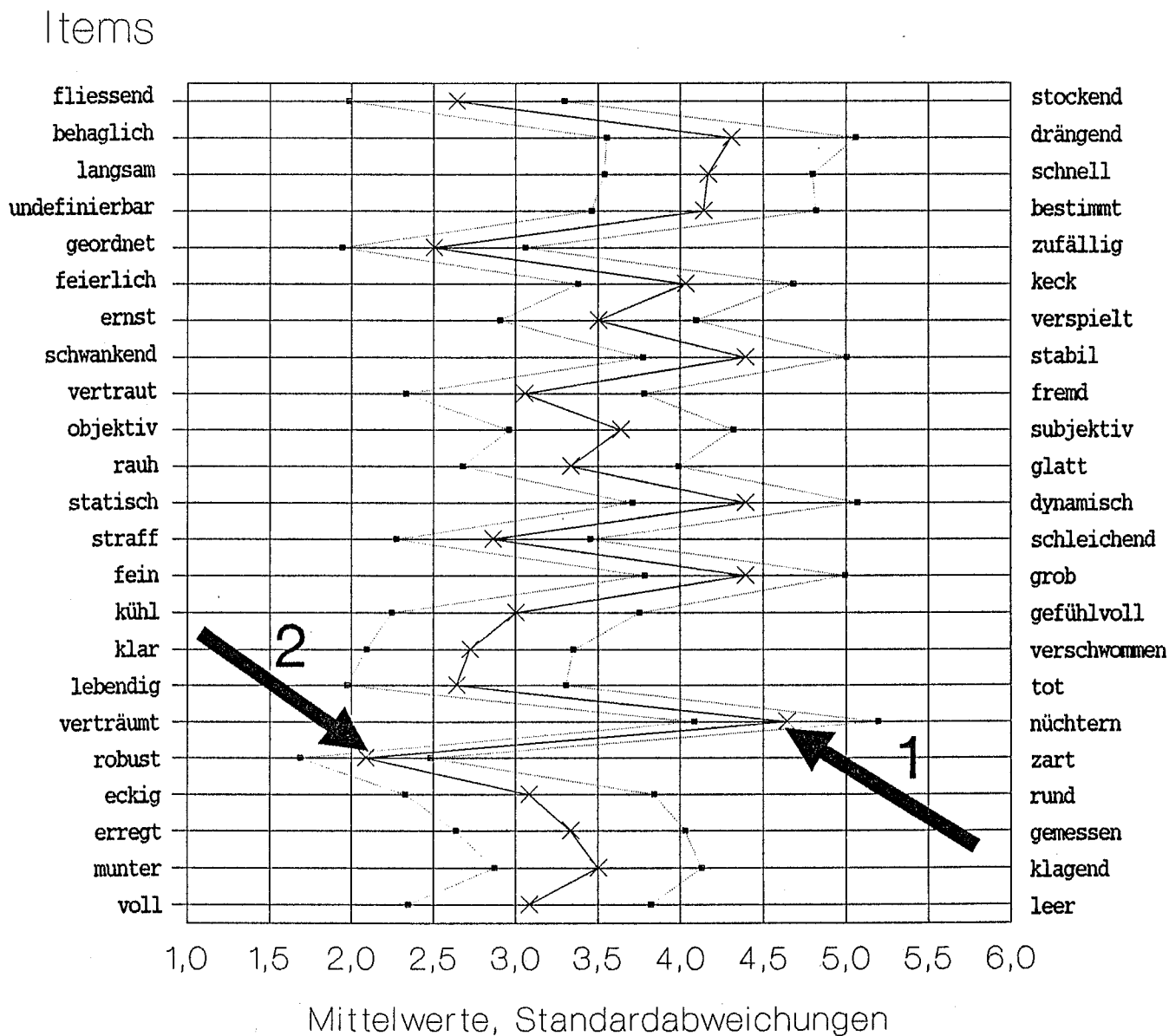
Items



hell (4), regelmässig (5)

Seite 2 (Items 24 bis 46)

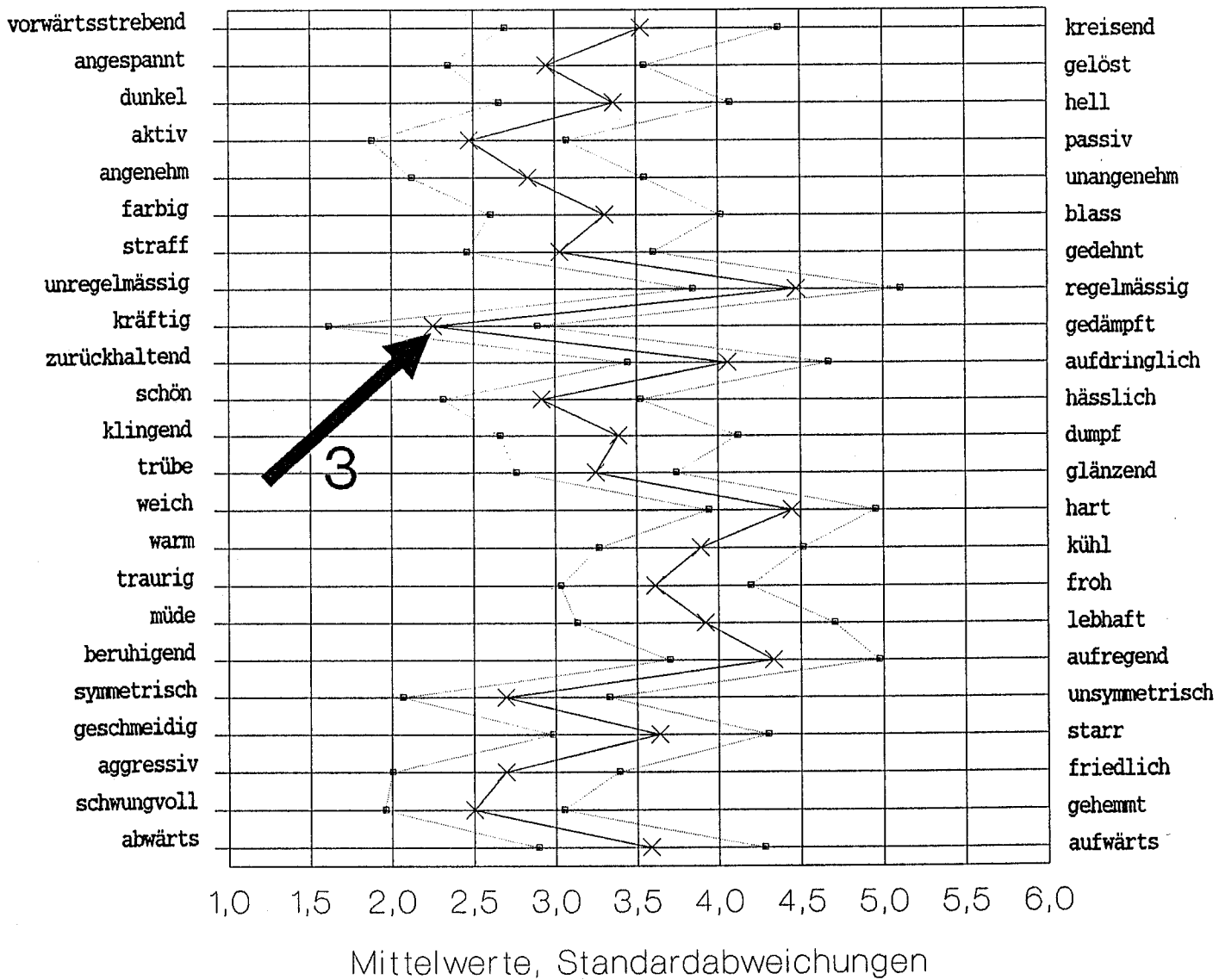
# Mittelwertsprofil Nr. 11: Midnight Oil (Musik Nr. 2)



nüchtern (1), robust (2)

# Forts. Mittelwertsprofil Nr. 11: Midnight Oil (Musik Nr. 2)

Items

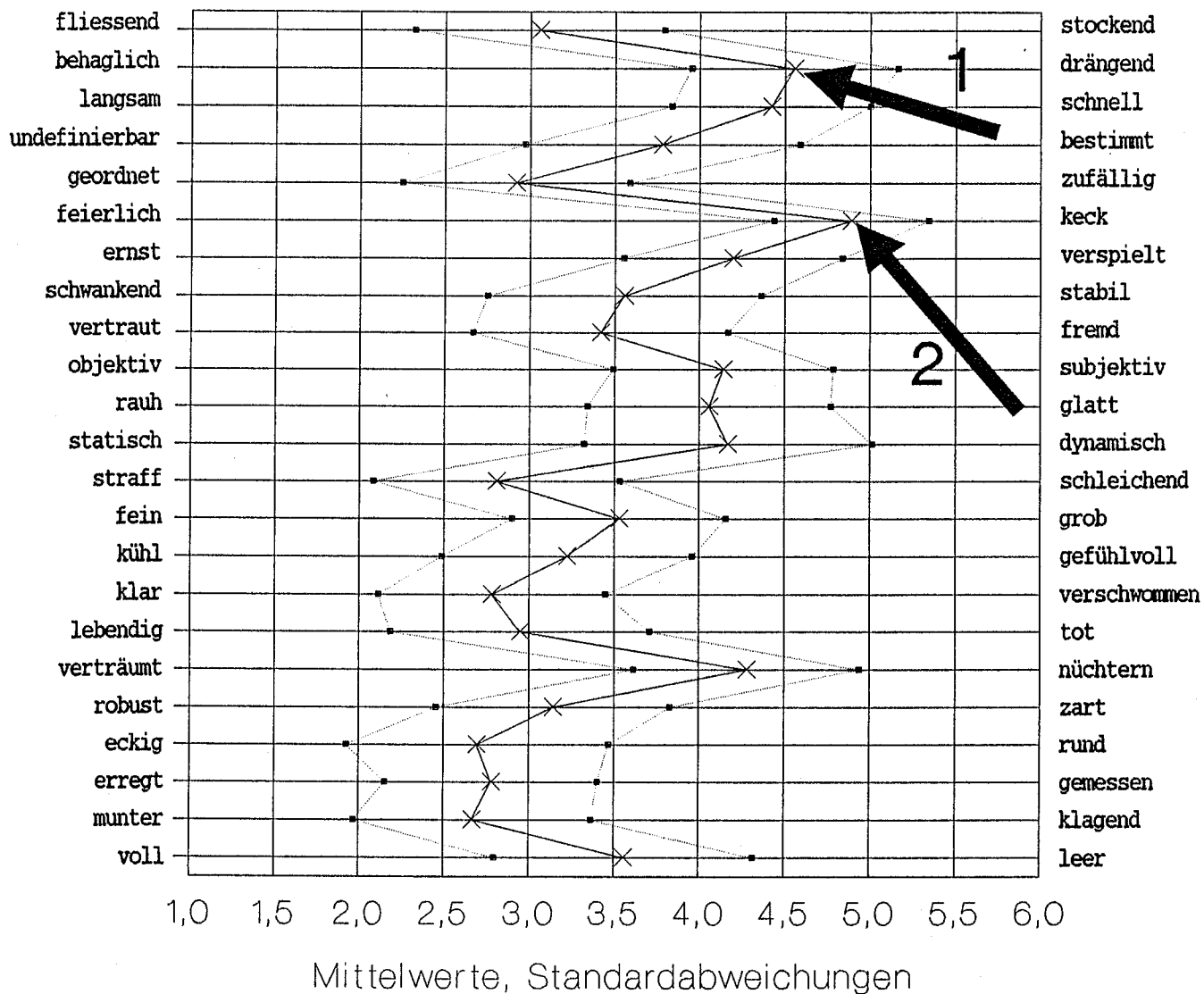


**kräftig (3)**

Seite 2 (Items 24 bis 46)

# Mittelwertsprofil Nr. 12: Jody Watley (Musik Nr. 3)

Items

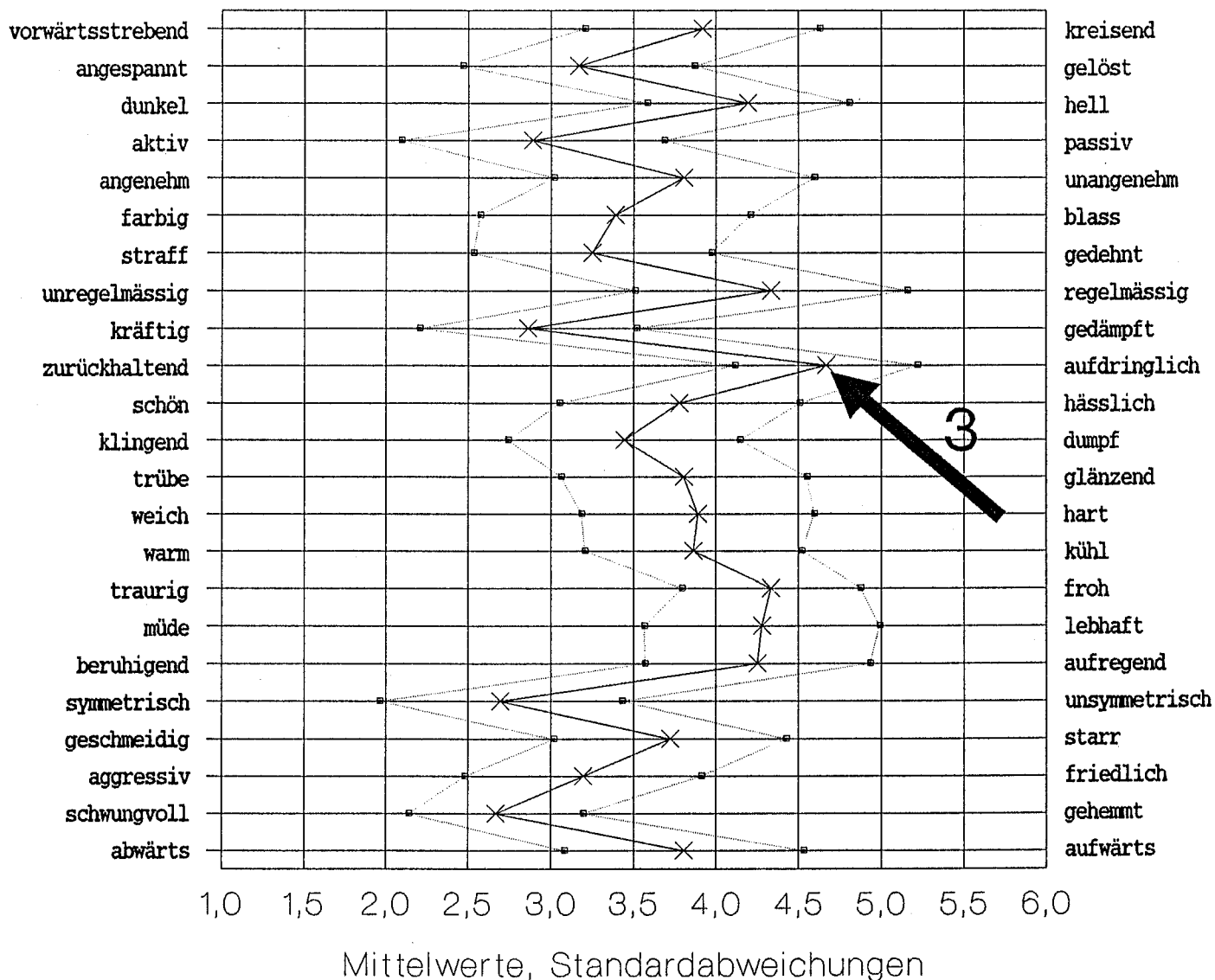


**drängend (1), keck (2)**

Seite 1 (Items 1 bis 23)

# Forts. Mittelwertsprofil Nr. 12: Jody Watley (Musik Nr. 3)

Items

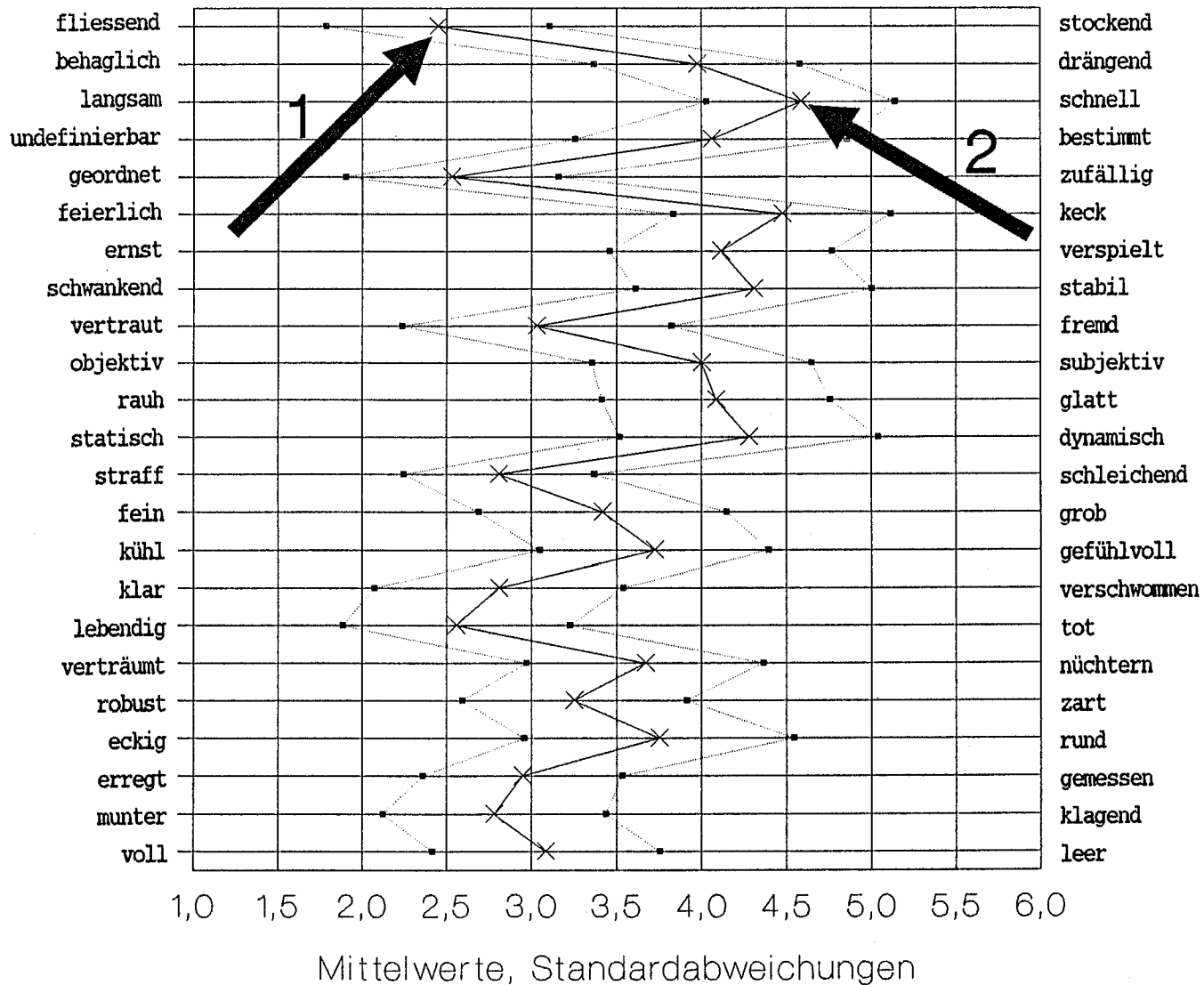


**aufdringlich (3)**

Seite 2 (Items 24 bis 46)

# Mittelwertsprofil Nr. 14: passendes Video (Bed. Nr. 2)

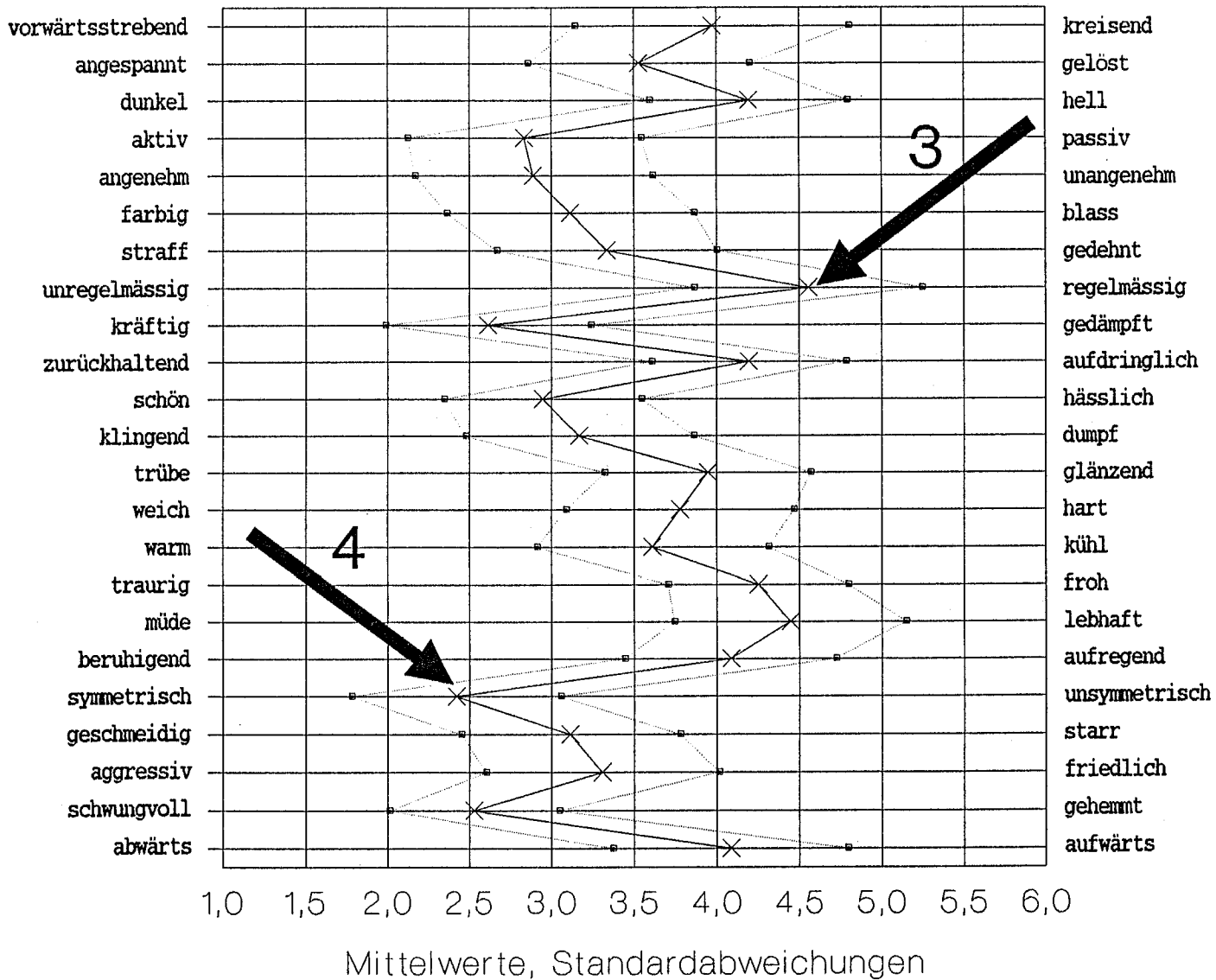
Items



**fliessend (1), schnell (2)**

# Forts. Mittelwertsprofil Nr. 14: passendes Video (Bed. Nr. 2)

Items

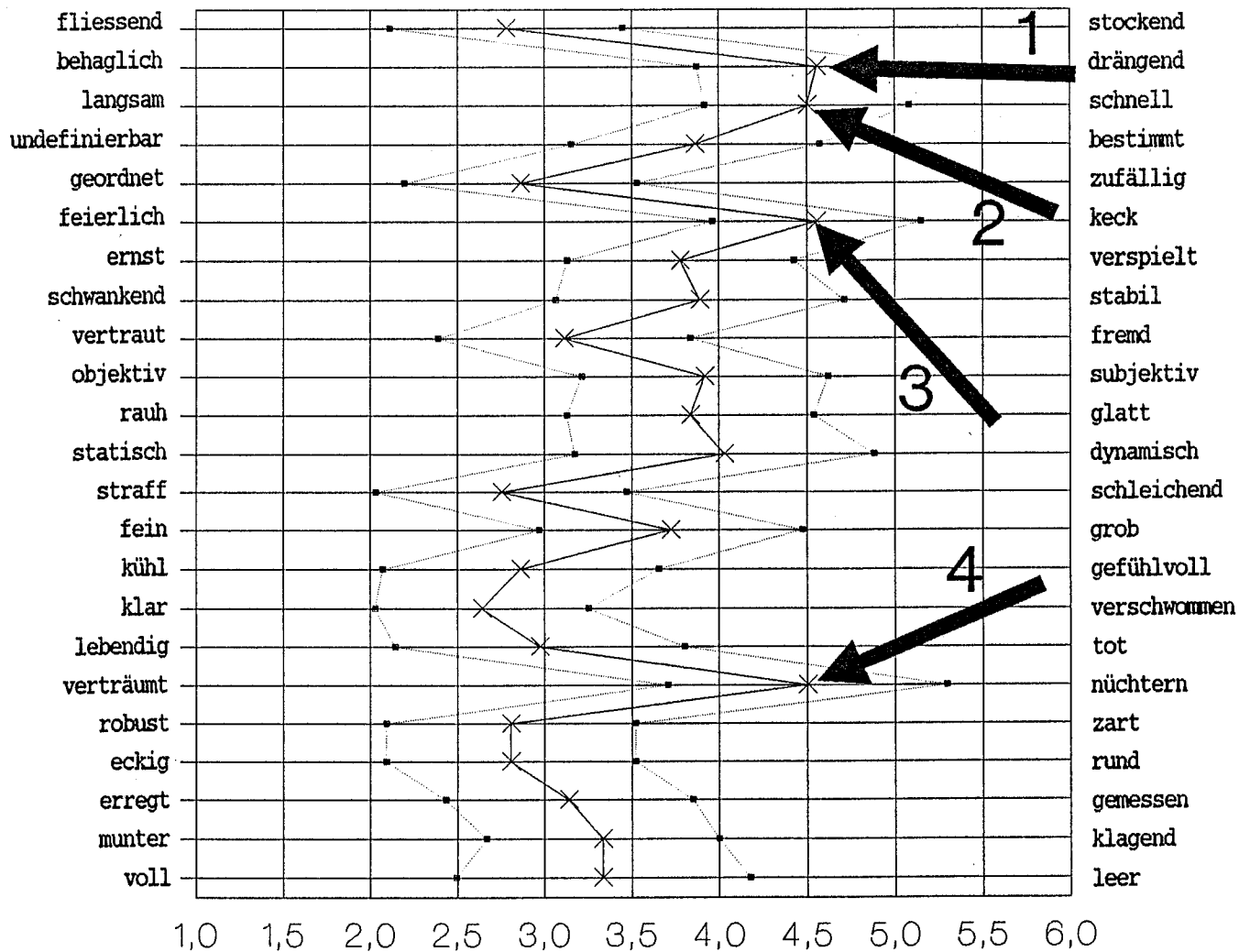


regelmässig (3), symmetrisch (4)

Seite 2 (Items 24 bis 46)

# Mittelwertsprofil Nr. 15: Zufallsvideo (Bedingung Nr. 3)

Items



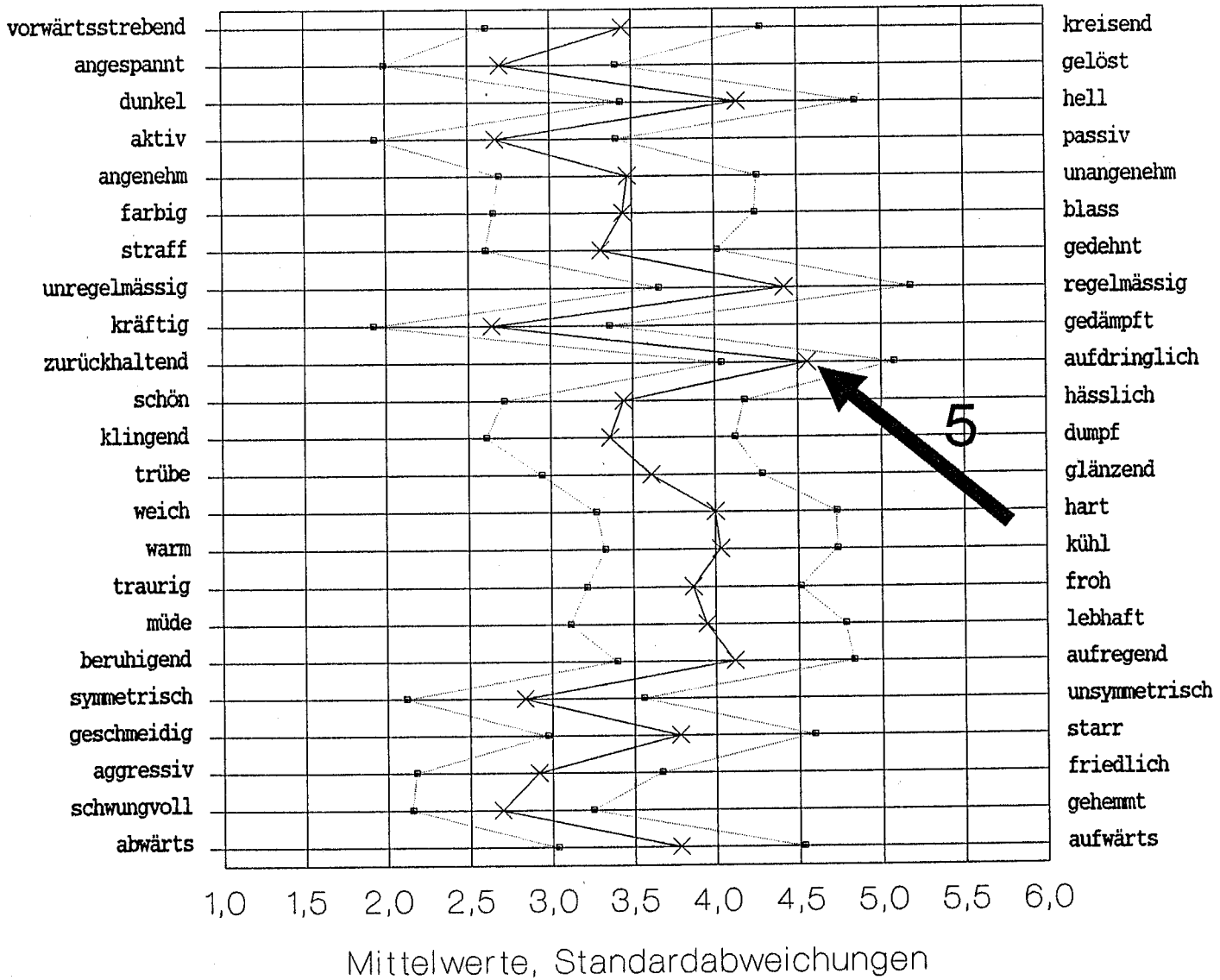
Mittelwerte, Standardabweichungen  
**drängend (1), schnell (2),  
 keck (3), nüchtern (4)**

Seite 1 (Items 1 bis 23)



# Forts. Mittelwertsprofil Nr. 15: Zufallsvideo (Bedingung Nr. 3)

Items



**aufdringlich (5)**

Seite 2 (Items 24 bis 46)

12.11. Einzelvergleiche der Varianzanalyse über die  
Standardabweichungen (zum Kap. 9.7.2.):

SCHEFFE-TESTS DER ANOVA ÜBER DIE VARIANZEN

a) Faktorstufenkombinationen:

Mittelwertsdifferenzen der Faktorstufenkombinationen:

	AUDIO:		A/V MATCH:			A/V MISMATCH:		
	Musik 2	Musik 3	Musik 1	Musik 2	Musik 3	Musik 1	Musik 2	Musik 3
Musik 1	0.02	-.28**	-.31**	0.15*	0.17*	-.15*	-.13	-.08
AUDIO: Musik 2		-.30**	-.33**	0.13	0.15*	-.17*	-.15*	-.10
	Musik 3		-.03	0.43**	0.45**	0.13	0.15*	0.20**
		Musik 1		0.46**	0.48**	0.16*	0.18**	0.23**
A/V MATCH :			Musik 2	0.02		-.30**	-.28**	-.23**
				Musik 3		-.32**	-.30**	-.25**
					Musik 1	0.02		0.07
A/V MISMATCH						Musik 2		0.05

$\alpha$ -Niveau 10%: Diffcrit = 0.136  
 $\alpha$ -Niveau 5%: Diffcrit = 0.151 -> \*  
 $\alpha$ -Niveau 1%: Diffcrit = 0.178 -> \*\*

b) Faktor "Musik":

Mittelwertsdifferenzen des Faktors "Musik":

	Musik 2	Musik 3
Musik 1	0.17**	0.10**
Musik 2		-0.07*

$\alpha$ -Niveau 10%: Diffcrit = 0.061  
 $\alpha$ -Niveau 5%: Diffcrit = 0.069 -> \*  
 $\alpha$ -Niveau 1%: Diffcrit = 0.086 -> \*\*

c) Faktor "Präsentationsbedingungen":

Mittelwertsdifferenzen des Faktors "Präsentationsbedingungen":

	Audio	A/V match
A/V mismatch	0.03	0.12**
Audio		0.09**

$\alpha$ -Niveau 10%: Diffcrit = 0.061  
 $\alpha$ -Niveau 5%: Diffcrit = 0.069 -> \*  
 $\alpha$ -Niveau 1%: Diffcrit = 0.086 -> \*\*