

# GESStabs

---

Punktwertdarstellung mit  
Score Tab



Gesellschaft für Software  
in der Sozialforschung mbH

Waterloohain 6 - 8  
22769 Hamburg  
Tel.: 040 - 853 753 - 0  
Fax: 040 - 853 753 - 33  
[www.gessgroup.de](http://www.gessgroup.de)

---

# ScoreTab

## Ein spezialisierter Tabellentyp für Punktwertdarstellungen.

In einigen Bereichen der Marktforschung benötigt man spezielle Darstellungen, in der Punktwerte summiert und als Soll-Ist-Erfüllung dargestellt werden. Ein typisches Beispiel sind sogenannte „Mystery-shopping“-Messungen, in denen erwartete Aspekte z.B. des Verkäuferverhaltens oder der Geschäftsausstattung mit Hilfe eines standardisierten Erhebungsinstruments erfasst werden.

Für die gewünschte Darstellung sollen einzelne Aspekte mit Punktwerten (Scores) assoziiert werden, und die Summe dieser Punktwerte soll ermittelt und in Relation zur maximal erreichbaren Summe dargestellt werden.

Hierfür steht das GESS tabs Tabellenkommando Scoretab zur Verfügung.

Für ein Beispiel gehen wir von nur 4 Variablen bzw. Fragen aus: Die Platzierung von Werbung für Produkt xxx außen vor dem Verkaufsbüro (a1) und innen (a2); die aktive Signalisierung durch einen Verkäufer, dass der Kunde wahrgenommen wurde (a3), und die Wartezeit bis zum ersten Kontakt (a4).

Den Ausprägungen dieser Variablen wird nun jeweils ein Punktwert zugeordnet. Nehmen wir einmal an, die Produktwerbung außen oder innen kann nun gar nicht, sehr gut sichtbar oder eher versteckt sein.

Es gelte also:

```
labels a1 a2 =  
1 "gar nicht"  
2 "verdeckt"  
3 "gut sichtbar"  
;
```

Für gar nicht (Code 1) gibt es 0 Punkte, für verdeckte Darstellung (Code 2) gibt es 5, für gut sichtbare Darstellung gilt ein Score von 15.

Für a3 gebe es nur die Antworten Ja (Code 1) oder Nein (Code 2); ein Nein führt zu Null, ein Ja zu 10 Punkten.

Die Frage nach der Wartezeit hat die Ausprägungen:

```
labels a4 =  
1 "sofort"  
2 "unter einer Minute"  
3 "bis 3 Minuten"  
4 "länger"  
;
```

Die zu vergebenden Punktwerte sind: sofort: 15 Punkte, unter 1 Minute 10 Punkte, bis 3 Minuten 5 Punkte, bei „länger“ hat der Kandidat Null Punkte.

Erwartet wird also eine Tabelle in etwa der folgenden Form für einen Einzelfall:

### Mysteryshopping in der einfachsten Form

	Maximale Punktzahl	Erreichte Punktzahl	Zielerfüllung
<b>Geschäftsausstattung</b>			
Werbung für Produkt xxx außen sichtbar	15	5	33 %
Werbung für Produkt xxx innen sichtbar	15	15	100 %
<b>Bei Eintritt ins Geschäft</b>			
Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert	10	0	0 %
Schnelle Bedienung	15	10	67 %

GESS mbH

Wie sind wir jetzt zu dieser Tabelle gekommen. In der Folge die relevante Passage aus dem Script:

```

ScoreTab =
CONTROL fallnr
LABELS "Maximale Punktzahl" "Erreichte Punktzahl" "Ziel~erfüllung"
IDENT no
| SUBTITLE "Geschäftsausstattung"
  "Werbung für Produkt xxx außen sichtbar"
  VARIABLE a1 1:0 2:5 3:15 MAX 15
| "Werbung für Produkt xxx innen sichtbar"
  VARIABLE a2 1:0 2:5 3:15 MAX 15
| SUBTITLE "Bei Eintritt ins Geschäft"
  "Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert"
  VARIABLE a3 1:10 2:0 MAX 10
| "Schnelle Bedienung"
  variable a4 1:15 2:10 3:5 4:0 MAX 15
;

```

Gehen wir die einzelnen Aspekte durch:

#### CONTROL:

Die hier angegebene Variable dient der Differenzierung, welche Fälle in eine Tabelle zusammengefasst werden sollen. Wir haben hier die Fallnummer gewählt; die Darstellung bezieht sich hier also auf Einzelfälle. (Zusätzlich haben wir mit einem TABSELECT auch nur einen einzelnen Fall ausgewählt; aber das ist nicht zwingend, s.u.) Würde man hier z.B. die Betriebsnummer angeben, so würden alle Fälle im Datensatz mit derselben Betriebsnummer zu einer Tabelle zusammengefasst werden, und es würden so viele Tabellen produziert werden, wie es unterschiedliche Betriebsnummern gibt.

#### LABELS:

Hier werden die Beschriftungen der Spalten der Tabelle eingetragen. Dies müssen mindestens 3 Strings sein; dies sind die Texte für die Spalte mit den maximal erreichbaren Punktzahlen (1), den tatsächlich erreichten Punkten (2), und der prozentualen Zielerfüllung (3). Es dürfen bis zu 5 Strings stehen; dann werden weitere Spalten mit konstanten Texten angefügt, die der Benutzer eingeben kann. Hier haben wir nur von den minimal 3 Spalten Gebrauch gemacht.

#### IDENT:

Wie oben ausgeführt, entsteht je eine Tabelle für alle Werte, die die unter CONTROL angegebene Variable im Datensatz annimmt. Wenn man also mehrere solcher Tabellen bekommt, ist es sinnvoll, diese auch im Tabellenausdruck identifizieren zu können. Dazu dient die IDENT-Spezifikation. Hier: IDENT NO. Das bedeutet: keine Identifikation. Das Thema nehmen wir unten wieder auf.

Die Tabellenzeilen werden jeweils durch einen senkrechten Strich eingeleitet. Zwischen dem Senkrechten Strich und dem Zeilenlabel darf SUBTITLE <string> stehen. Dann wird vor dem eigentlichen Text noch ein Untertitel eingefügt. Dann folgt das Label für die Tabellenzeile. Danach dürfen dann alle Variablen stehen, die in die Punktsumme für die Zeile eingehen sollen, gefolgt von den Punktberechnungsvorschriften und dem maximalen Punktwert für diese Variable.

Das war's für diese einfachste Tabelle.

Als nächstes wollen wir die Identifikation des Falles in die Tabelle aufnehmen. Standardmäßig sind 3 mögliche Positionierungen vorgesehen: als TABLETITLE, als HEADER oder als FOOTER. Anschließend steht dann der Name der Variablen, die zur Identifizierung herangezogen werden soll. Wir wählen als Beispiel den TABLETITLEm und wollen über die Fallnummer identifizieren. Die IDENT-Zeile im Script oben ändert sich dann zu:

```
IDENT TABLETITLE fallnr
```

Die Tabelle oben haben wir mit TABSELECT auf die fallnr 1 beschränkt. Die Variable fallnr hat hierfür außerdem ein passendes Label bekommen:

```

labels fallnr =
1 "autohaus xyz in abc am 24.12.2007"
;

```

## autohaus xyz in abc am 24.12.2007

	Maximale Punktzahl	Erreichte Punktzahl	Zielerfüllung
<b>Geschäftsausstattung</b>			
Werbung für Produkt xxx außen sichtbar	15	5	33 %
Werbung für Produkt xxx innen sichtbar	15	15	100 %
<b>Bei Eintritt ins Geschäft</b>			
Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert	10	0	0 %
Schnelle Bedienung	15	10	67 %

GESS mbH

Vom Autohaus xyz sind nun mehrere Fälle im Datensatz vorhanden, insgesamt 7 Stück. Relativ einfach kann man nun eine Tabelle für alle diese Fälle bekommen; man setzt CONTROL und IDENT auf die Variable autohaus.

```
TabSelect autohaus eq 1;
labels autohaus =
1 "Autohaus xyz"
;
ScoreTab =
CONTROL autohaus
LABELS "Maximale Punktzahl" "Erreichte Punktzahl" "Zielerfüllung"
IDENT TABLETITLE autohaus
| SUBTITLE "Geschäftsausstattung"
  "Werbung für Produkt xxx außen sichtbar"
  VARIABLE a1 1:0 2:5 3:15 MAX 15
| "Werbung für Produkt xxx innen sichtbar"
  VARIABLE a2 1:0 2:5 3:15 MAX 15
| SUBTITLE "Bei Eintritt ins Geschäft"
  "Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert" //
MAX kopf4 "87,5%" // "1..."
  VARIABLE a3 1:10 2:0 MAX 10
| "Schnelle Bedienung"
  variable a4 1:15 2:10 3:5 4:0 MAX 15
;
```

Dabei kommt folgendes heraus:

### Autohaus xyz

	Maximale Punktzahl	Erreichte Punktzahl	Zielerfüllung
<b>Geschäftsausstattung</b>			
Werbung für Produkt xxx außen sichtbar	105	25	24 %
Werbung für Produkt xxx innen sichtbar	105	40	38 %
<b>Bei Eintritt ins Geschäft</b>			
Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert	70	0	0 %
Schnelle Bedienung	105	50	48 %

GESS mbH

Relativ oft wird zusätzlich eine Spalte gewünscht, in der die durchschnittlichen Resultate vorangegangener Studien sichtbar gemacht werden, damit der Leser den Prozentwert der Zielerfüllung einordnen kann; das sogenannte „Benchmarking“. Wie schon oben angedeutet, kann man eine zusätzliche Spalte ganz einfach dadurch anfordern, dass man einen zusätzlichen Spaltentitel vergibt. Dann muss man in jeder Zeile nach dem Zeilenlabel auch einen zusätzlich Text angeben.

Das Script wird also folgendermaßen erweitert:

```

tabletitle = "Mysteryshopping mit Benchmarking";
ScoreTab =
CONTROL fallnr
LABELS "Maximale Punktzahl" "Erreichte Punktzahl" "Ziel~erfüllung" "Bench~mark"
IDENT no // TABLETITLE fallnr
| SUBTITLE "Geschäftsausstattung"
  "Werbung für Produkt xxx außen sichtbar" "53,9%"
  VARIABLE a1 1:0 2:5 3:15 MAX 15
| "Werbung für Produkt xxx innen sichtbar" "67,2%"
  VARIABLE a2 1:0 2:5 3:15 MAX 15
| SUBTITLE "Bei Eintritt ins Geschäft"
  "Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert"
"83,6%"
  VARIABLE a3 1:10 2:0 MAX 10
| "Schnelle Bedienung" "78,7%"
  variable a4 1:15 2:10 3:5 4:0 MAX 15
;

```

In der zusätzlichen Spalte stehen jetzt als konstante Strings die hinzugefügten Vergleichswerte.

### Mysteryshopping mit Benchmarking

	Maximale Punktzahl	Erreichte Punktzahl	Zielerfüllung	Benchmark
<b>Geschäftsausstattung</b>				
Werbung für Produkt xxx außen sichtbar	15	5	33 %	53,9%
Werbung für Produkt xxx innen sichtbar	15	15	100 %	67,2%
<b>Bei Eintritt ins Geschäft</b>				
Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert	10	0	0 %	83,6%
Schnelle Bedienung	15	10	67 %	78,7%

---

GESS mbH

Der Font für diese erste zusätzliche Spalte wird übrigens über die Anweisung

```
USEFONT COMMENT = ...,
```

gesteuert. Dieses Spiel kann man mit einer weiteren Spalte weiter spielen; man gibt einen weiteren Spaltentitel unter LABELS an und einen zusätzlichen Text in jeder Zeile. Wir benutzen diese Spalte, um dem Leser einen Verweis auf die im Fragebogen zugrunde liegenden Fragen zu geben. (Der Font dieser Spalte wird über USEFONT DESCRIPTION angesteuert.)

```

tabletitle = "Mysteryshopping mit Benchmarking und Verweis";
ScoreTab =
CONTROL fallnr
LABELS "Maximale Punktzahl" "Erreichte Punktzahl" "Ziel~erfüllung" "Bench~mark"
"siehe Fragen"
IDENT no // TABLETITLE fallnr
| SUBTITLE "Geschäftsausstattung"
  "Werbung für Produkt xxx außen sichtbar" "53,9%" A1
  VARIABLE a1 1:0 2:5 3:15 MAX 15
| "Werbung für Produkt xxx innen sichtbar" "67,2%" A2
  VARIABLE a2 1:0 2:5 3:15 MAX 15
| SUBTITLE "Bei Eintritt ins Geschäft"
  "Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert"
"83,6%" A3
  VARIABLE a3 1:10 2:0 MAX 10
| "Schnelle Bedienung" "78,7%" A4
  variable a4 1:15 2:10 3:5 4:0 MAX 15
;

```

## Mysteryshopping mit Benchmarking und Verweis

	Maximale Punktzahl	Erreichte Punktzahl	Zielerfüllung	Benchmark	siehe Fragen
<b>Geschäftsausstattung</b>					
Werbung für Produkt xxx außen sichtbar	15	5	33 %	53,9%	A1
Werbung für Produkt xxx innen sichtbar	15	15	100 %	67,2%	A2
<b>Bei Eintritt ins Geschäft</b>					
Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert	10	0	0 %	83,6%	A3
Schnelle Bedienung	15	10	67 %	78,7%	A4

GESS mbH

Natürlich kann man auch Score-Zeilen bilden, die auf mehrere Variablen zurückgehen. Als Beispiel fügen wir eine zusätzliche Zeile ein, die sowohl die Außen- als auch die Innenwerbung umfasst.

```

tabselect fallnr eq 1;
tabletitle = "Mysteryshopping mit mehreren Variablen in einer Zeile";
ScoreTab =
CONTROL fallnr
LABELS "Maximale Punktzahl" "Erreichte Punktzahl" "Ziel~erfüllung"
IDENT no
| SUBTITLE "Geschäftsausstattung"
  "Werbung für Produkt xxx außen sichtbar"
  VARIABLE a1 1:0 2:5 3:15 MAX 15
| "Werbung für Produkt xxx innen sichtbar"
  VARIABLE a2 1:0 2:5 3:15 MAX 15
| "Werbung für Produkt xxx innen und außen"
  VARIABLE a1 1:0 2:5 3:15 MAX 15
  VARIABLE a2 1:0 2:5 3:15 MAX 15
| SUBTITLE "Bei Eintritt ins Geschäft"
  "Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert"
  VARIABLE a3 1:10 2:0 MAX 10
| "Schnelle Bedienung"
  variable a4 1:15 2:10 3:5 4:0 MAX 15
;

```

## Mysteryshopping mit mehreren Variablen in einer Zeile

	Maximale Punktzahl	Erreichte Punktzahl	Zielerfüllung
<b>Geschäftsausstattung</b>			
Werbung für Produkt xxx außen sichtbar	15	5	33 %
Werbung für Produkt xxx innen sichtbar	15	15	100 %
Werbung für Produkt xxx innen und außen	30	20	67 %
<b>Bei Eintritt ins Geschäft</b>			
Wahrnehmung des Kunden wird durch das Verkaufspersonal aktiv signalisiert	10	0	0 %
Schnelle Bedienung	15	10	67 %

GESS mbH

## Hier noch ein einfaches Syntaxschema mit ein paar zusätzlichen Erläuterungen:

```
SCORETAB =  
CONTROL <controlvar>  
LABELS "label for max" "label for score" "label for percent" [ "label for comment" [ "label for description" ] ]  
IDENT <ident>  
| [ SUBTITLE <subtitle> ]  
  <text for label> [ MAX <max> ] [ <comment> ] [ <description> ]  
  { VARIABLE <variable> <scorecodes> [ MAX <max> ] } *n  
| ....
```

```
<ident> ::= [ <destination> <variable> | NO ]  
<destination> ::= [ FOOTER | HEADER | TABLETITLE ]
```

```
<max> ::= [ <number> | variable ]
```

Als Maximum kann man sowohl eine Konstante als auch eine Variable angeben.

```
<scorecodes> ::= [ <number> : <number> ] * n
```

Beschreibt die Zuordnung von Score zu den einzelnen Codes der in die Tabellierung einbezogenen Variablen. z.B.: 1:0 2:10 d.h.: code 1 ergibt einen score von 0, code 2 einen von 10 etc.

Als Variablen haben wir in unserem Beispiel nur SINGLEQ verwendet; das ganze funktioniert aber auch mit Mehrfachnennungsvariablen.

Syntaktisch **muss** man die Maximalwerte nicht zwingend angeben; als Default wird immer die Summe aller Scores gebildet. Automatisch ist hier aber eigentlich nicht sinnvoll, denn es ist oft nicht ersichtlich, welche Wertekombinationen auftreten können und welche nicht. Die Software geht davon aus, dass der Benutzer sinnvolle Maximalwerte einträgt.

Wie man in der Syntax sieht, kann man neben dem Maximum für die Variablen auch ein Maximum für die Zeile angeben. Es gibt Konstellationen, wo man aus logischen Gründen das Maximum in der einen Variablen nicht erreichen kann, wenn man in einer anderen Variablen das Maximum erreicht hat, dann ist das Zeilenmaximum nicht identisch mit der Summe der Variablenmaxima. Dem kann man dann durch eine Maximum-Angabe auf der Zeilenebene entsprechend Rechnung tragen.

Als Maximalwerte kann man neben numerischen Konstanten auch den Namen einer SINGLEQ Variablen angeben; dies ist z.B. dann von Belang wenn bei bestimmten Fallkonstellationen andere Maxima gelten als bei anderen.

Ein Tabellierlauf kann, wie oben angedeutet, mehrere Tabellenseiten je Scoretab-Tabellenstatement erzeugen. Dann kann es praktisch sein, dass diese Tabellen nicht in einer gemeinsamen Ausgabedatei zusammen vorliegen. Hierfür gibt es einen globalen Schalter:

```
SingleScoreFiles = YES;
```

Wenn man diese Option benutzt, wird für jede Tabelle eine einzelne Postscript-Datei erzeugt. Diese Dateien tragen systematische Namen, die aus dem Wert der Variablen <CONTROL> und der Tabellenummer gebildet werden. Die Dateien heißen

```
Score_<x>_<y>.ps
```

<x> steht dabei für den numerischen Wert der CONTROL-Variablen ; <y> für die interne Tabellenummer. Das erste ScoreTab-Statement im Script hat die Nummer 1, das zweite die Nummer 2 etc.

Wenn man SingleScoreFiles benutzt, sollten in diesem Tabellierlauf keine anderen Tabellen (TABLE, XTAB, CODEBOOK etc) erzeugt werden.