

# IBM SPSS Syntax

Eine anwendungsorientierte Einführung

Bearbeitet von

Von Prof. Dr. Marko Sarstedt, MBR, Prof. Dr. Tobias Schütz, MBR, und Sascha Raitchel, MBR

3. Auflage 2018. Buch. XIII, 228 S. Kartoniert

ISBN 978 3 8006 5761 2

Format (B x L): 16,0 x 24,0 cm

[Wirtschaft > Volkswirtschaft > Wirtschaftsstatistik, Demographie](#)

Zu [Inhalts-](#) und [Sachverzeichnis](#)

schnell und portofrei erhältlich bei

**beck-shop.de**  
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung [beck-shop.de](#) ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

Die Anweisung PRINT erlaubt es, die Matrix *A* im Ausgabefenster anzeigen zu lassen:

### Run MATRIX procedure :

```
A
  1  2
  3  4
  5  6

----- END MATRIX -----
```



Matrizenamen sollten in SPSS höchstens 8 Zeichen lang sein. Sind Namen für Matrizen länger, so gibt SPSS lediglich eine Warnung aus und kürzt den Namen automatisch auf 8 Zeichen. Das Programm wird trotzdem ausgeführt. Folglich kann es leicht zu Fehlern kommen, wenn beispielsweise mehrere Matrizen mit identischem Präfix (z.B. *modellmatrix\_1*, *modellmatrix\_2*, *modellmatrix\_3* etc.) in einem Syntaxprogramm spezifiziert werden.

### 7.1.2 Einlesen einer von SPSS erzeugten Matrix: MGET

Neben einer direkten Matrizeneingabe mit der COMPUTE-Anweisung ermöglicht SPSS das Einlesen von Matrizen, die zuvor im Rahmen einer anderen Analyse erstellt wurden. Hierzu verwenden wir die Anweisung MGET. Typische Anwendungen umfassen unter anderem das Einlesen einer Korrelations- oder Distanzmatrix, die zuvor beispielsweise mit Hilfe der CORRELATIONS- beziehungsweise CLUSTER-Prozedur erstellt wurden.

Im Folgenden wollen wir zunächst eine Korrelationsmatrix dreier Variablen unseres Beispieldatensatzes in ein Matrix-Programm einlesen. In einem zweiten Schritt wollen wir die sogenannten Eigenwerte der Korrelationsmatrix berechnen und ausgeben lassen. Die Eigenwerte haben beispielsweise bei einer Faktorenanalyse große Bedeutung (Sarstedt und Mooi 2019). Der Eigenwert eines Faktors gibt an, wie viel Varianz dieser Faktor aus den einzelnen Variablen bündelt. In unserem Beispiel können wir drei Faktoren bilden. Für diese drei Faktoren lassen wir uns wie folgt die drei Eigenwerte ausgeben:

```
CORRELATIONS
  /VARIABLES = customer_satisfaction_index_gesamt
              fahrzeugattraktivität_index
              haltungskosten_index
  /MATRIX=OUT(*).
MATRIX.
  MGET /FILE = *
      /TYPE = CORR.
  COMPUTE eigenw = EVAL(cr).
  PRINT eigenw.
END MATRIX.
```

Mit Hilfe der CORRELATIONS-Prozedur werden zunächst die Korrelationen zwischen den drei Variablen berechnet. Der Unterbefehl `VARIABLES` listet die drei Variablen, für die wir die Korrelationen berechnen wollen. Im SPSS-Ausgabefenster erhalten wir die Korrelationsmatrix. Durch die Angabe des Unterbefehls `MATRIX=OUT(*)` wird zusätzlich ein neues Datenfenster geöffnet. Der neue Datensatz beinhaltet (neben einigen deskriptiven Informationen) ebenfalls die resultierende (3x3)-Korrelationsmatrix (Abbildung 7.1).

	ROWTYPE_	VARNAME_	customer_satisfaction_index_gesamt
1	MEAN		798,3801468
2	STDDEV		107,9729435
3	N	customer_satisfaction_index_gesamt	17612,0000000
4	N	fahrzeugattraktivität_index	17612,0000000
5	N	haltungskosten_index	17612,0000000
6	CORR	customer_satisfaction_index_gesamt	1,0000000
7	CORR	fahrzeugattraktivität_index	,8336517
8	CORR	haltungskosten_index	,7606015

Abbildung 7.1: Korrelationsmatrix

Das Symbol \* im Unterbefehl `MATRIX=OUT(*)` stellt sicher, dass obige Datendatei als neue (temporäre) Arbeitsdatei definiert wird. Alternativ können wir statt \* einen Laufwerkpfad angeben, unter dem die Matrix gespeichert wird. Durch \* beziehen sich die folgenden Befehle auf die temporäre Arbeitsdatei. Zunächst wird mit Hilfe von `Matrix` der Matrix-Processor aktiviert. Innerhalb des Matrix-Programms wird dann die Korrelationsmatrix mit der `MGET`-Anweisung eingelesen.

Die Syntax der `MGET`-Anweisung lautet in allgemeiner Form:

```
MGET [/FILE=file-reference]
[/TYPE={ COV   }]
      { CORR  }
      { MEAN  }
      { STDDEV}
      { N     }
      { COUNT }
```

`MGET /FILE = *` liest zunächst die aktive Arbeitsdatei ein. Im Unterbefehl `FILE` können wir alternativ einen Laufwerkpfad angeben. Dort ist die einzulesende Matrix abgelegt. Wichtig ist hierbei, dass die Datei im `.sav`-Format vorliegt und die Daten in Matrixform eingegeben sind. Soll die Matrix des momentan aktiven Datensatzes angesprochen werden, so geling dies, wie bereits im obigen Beispiel behandelt, durch die Eingabe \*.

Der Unterbefehl `TYPE = CORR` sagt SPSS, dass wir eine Korrelationsmatrix einlesen wollen. SPSS steuert in der aktiven Arbeitsdatei somit die Zeilen `ROWTYPE_ = CORR` an (siehe obige Abbildung 7.1). Da es sich um eine Korrelationsmatrix handelt, vergibt SPSS automatisch den Namen `cr` für die einzulesende Matrix. Den Namen `cr` können wir in den folgenden Befehlen nutzen und diese Matrix gezielt ansteuern.

Der Unterbefehl `TYPE` ermöglicht uns neben Korrelationsmatrizen aber auch andere Matrixentypen einzulesen. Tabelle 7.1 gibt einen Überblick über die unterstützten Typen.

Schlüsselwort in <code>TYPE</code>	Resultierende Matrixbezeichnung	Bedeutung
<code>COV</code>	<i>cv</i>	Kovarianzmatrix
<code>CORR</code>	<i>cr</i>	Korrelationsmatrix
<code>MEAN</code>	<i>mn</i>	Mittelwertvektor
<code>STDDEV</code>	<i>sd</i>	Vektor mit Standardabweichungen
<code>N</code>	<i>nc</i>	Vektor mit Fallzahlen
<code>COUNT</code>	<i>cn</i>	Vektor mit Häufigkeiten

Tabelle 7.1: Matrixentypen

Nach dem Einlesen der Korrelationsmatrix wollen wir die Eigenwerte berechnen. Dies gelingt mit der `COMPUTE`-Anweisung in Verbindung mit der Matrix-Funktion `EVAL` (vgl. Kapitel 7.4). Die Matrix-Funktion `EVAL` berechnet für symmetrische Matrizen wie eine Korrelationsmatrix deren sogenannte Eigenwerte. Nachfolgend werden mit dem `PRINT`-Befehl die Eigenwerte im Ausgabefenster angezeigt:

**Run MATRIX procedure:**

**MGET created matrix CR.**

**The matrix has 3 rows and 3 columns.**

**The matrix was read from the record(s) of row type CORR.**

**EIGENW**

**2,437730912**

**,451411589**

**,110857498**

**----- END MATRIX -----**

SPSS protokolliert zunächst die Matrix-Prozedur, die wir initiiert hatten. Der `PRINT`-Befehl listet anschließend die drei Eigenwerte. Diese betragen 2,44, 0,45 und 0,11.

### 7.1.3 Definition von Matrizen: `GET`

Neben der direkten Eingabe und dem Einlesen bereits existierender Matrizen können wir auch ein Bündel von Variablen als Satz von Vektoren in einer Matrix zusammenfassen. Dies gelingt uns mit der `GET`-Anweisung:

```

GET variable-reference
  [/FILE=file]
  [/VARIABLES=variable-list]
  [/MISSING={ACCEPT}]
                {OMIT  }
                {value }
  [/SYSMIS= {OMIT }]
                {value}

```

variable-reference	Name der neu zu bildenden Matrix.
FILE	Angabe des Laufwerkpfades der einzulesenden Datendatei. Durch die Eingabe eines Sterns (*) oder das Auslassen dieses Unterbefehls wird der aktive Datensatz angesprochen.
VARIABLES	Spezifikation einer Variablenliste ( <i>variable-list</i> ), die in die Matrix eingelesen werden soll. Variablennamen müssen durch Kommas getrennt werden.
MISSING	Festlegung wie benutzerdefinierte fehlende Werte innerhalb der Matrix behandelt werden sollen. Weist eine der Variablen einen benutzer- oder systemdefiniert fehlenden Wert auf, so muss MISSING angegeben werden. Ist dies nicht der Fall, so gibt SPSS eine Fehlermeldung aus. ACCEPT: Benutzerdefinierte fehlende Werte werden übernommen. OMIT: Beobachtungen mit benutzerdefinierten fehlenden Werten werden verworfen. value: Benutzerdefinierte fehlende Werte werden in den angegebenen (numerischen) Wert umcodiert.
SYSMIS	Festlegung wie systemdefinierte fehlende Werte innerhalb der Matrix behandelt werden sollen. Der Unterbefehl kommt nur im Zusammenhang mit MISSING = ACCEPT zum Tragen. OMIT: Beobachtungen mit systemdefinierten fehlenden Werten werden verworfen. value: Systemdefinierte fehlende Werte werden in den angegebenen (numerischen) Wert umcodiert.

Innerhalb des folgenden Matrix-Programms werden aus unserem Beispieldatensatz die sechs aufeinander folgenden Variablen *fahrverhalten\_bremseleistung* bis *fahrverhalten\_gesamteindruck* in die Matrix *fahrverh* eingelesen. Hierbei werden benutzerdefinierte fehlende Werte übernommen und Beobachtungen, die systemdefinierte fehlende Werte aufweisen, verworfen. Mit der PRINT-Anweisung erfolgt eine Ausgabe der Matrix im Ausgabefenster. Damit im Ausgabefenster nicht alle 17.612 Beobachtungen angezeigt werden, reduzieren wir mit der SAMPLE-Anweisung die Anzahl der Beobachtungen zuvor auf fünf. Der Befehl SAMPLE zieht aus den 17.612 Beobachtungen eine Zufallsstichprobe von fünf Beobachtungen. Da wir mit SAMPLE eine Zufallsstichprobe ziehen, sieht die Matrix *fahrverh* nach jedem Durchlauf des Syntax-Programmes höchstwahrscheinlich anders aus.

```

GET FILE= '/Users/SPSS/JDPA Germany CSI.sav'.
SAMPLE 5 FROM 17612.
MATRIX.
  GET fahrverh
    /FILE = *
    /VARIABLES = fahrverhalten_bremsleistung TO
                fahrverhalten_gesamteindruck
    /MISSING = ACCEPT
    /SYSMIS = OMIT.
  PRINT fahrverh.
END MATRIX.

```

In unserem Fall wird im Ausgabefenster folgende Matrix ausgegeben:

### Run MATRIX procedure:

#### FAHRVERH

```

  8  8  9  9  9  9
  9  8 10  8  5  9
10 10 10 10 10 10
10  9  9  8  8  9
  9  9  9  9  9  9

```

----- END MATRIX -----

## 7.2 Matrizenausgabe: PRINT

In den letzten Beispielen haben wir bereits die PRINT-Anweisung aufgegriffen, die es uns ermöglicht Matrizen oder Matrixausdrücke im Ausgabefenster abzubilden.

```

PRINT [(matrix-expression)]
  [/FORMAT = format descriptor]
  [/TITLE = 'title']
  [{/RLABELS = list of quoted names}]
  [{/CLABELS = list of quoted names}]

```

Der Matrixausdruck (*matrix-expression*) bezeichnet die Matrix, welche im Ausgabefenster abgebildet werden soll. Anstatt eines Matrixnamens können wir auch einen komplexen Matrixausdruck in Form einer Funktion (vgl. Kapitel 7.4) angeben.

Die weiteren Unteranweisungen sind wie folgt zu interpretieren:

**FORMAT** Formatbeschreibung zur Ausgabe der Matrixdaten. Bezüglich der Formatbeschreibung gilt die in Kapitel 2.5.5 dargestellte Syntax.

**TITLE** Angabe eines Titels für die auszugebende Matrix. Der Titel muss in Anführungszeichen oder Apostrophe gesetzt werden.

- RLABELS** Angabe von Zeilenetiketten für die Matrix. Die Etiketten müssen in Anführungszeichen oder Apostrophe gesetzt werden. Sollen Etiketten für mehrere Zeilen angegeben werden, so sind diese durch Kommas zu trennen.
- CLABELS** Mit Hilfe von **CLABELS** können Spaltenetiketten vergeben werden. Die Unteranweisung funktioniert analog zu **RLABELS**.

Unter Bezugnahme auf das vorherige Beispiel gibt beispielsweise die **NCOL**-Funktion die Spaltenzahl (**N**umber of **C**olumns) der zuvor definierten Matrix *fahrverh* in Höhe von sechs aus. Wir nutzen die folgende Syntax:

```
GET FILE= '/Users/SPSS/JDPA Germany CSI.sav'.
SAMPLE 5 FROM 17612.
MATRIX.
  GET fahrverh
    /FILE = *
    /VARIABLES = fahrverhalten_bremsleistung TO
                fahrverhalten_gesamteindruck
    /MISSING = ACCEPT
    /SYSMIS = OMIT.
  PRINT NCOL(fahrverh)
    /TITLE = 'Anzahl Spalten'.
END MATRIX.
```

Der **PRINT**-Unterbefehl **TITLE = 'Anzahl Spalten'** vergibt eine individuelle Überschrift für die auszugebende Matrix (beziehungsweise in diesem Fall Zahl) und wir erhalten folgendes Ergebnis im Ausgabefenster:

**Run MATRIX procedure:**

**Anzahl Spalten**

**6**

----- **END MATRIX** -----

### 7.3 Arithmetische Operatoren

Neben der Ausführung einfacher **arithmetischer Operationen** (Addition, Subtraktion, Multiplikation und Division von Matrizen) ermöglicht SPSS auch die Ausführung elementweiser Multiplikationen, Divisionen oder Potenzierungen.

Bei Angabe der arithmetischen Operatoren zur **Addition (+)** oder **Subtraktion (-)** werden die entsprechenden Matrixelemente miteinander addiert bzw. voneinander subtrahiert. Entscheidend hierbei ist, dass die angegebenen Matrizen dieselbe Ordnung (d.h. dieselbe Zeilen- und Spaltenzahl) aufweisen. Gleichfalls kann, wie im nachfolgenden Beispiel, auch ein Skalar addiert oder subtrahiert werden.

Im folgenden Beispiel wird jedes Element der Matrix *x* um den Betrag fünf erhöht. Das Ergebnis dieser Operation wird der Matrix *summe* zugewiesen.

```

MATRIX.
COMPUTE x = {3,15,12;5,2,9;10,7,1}.
COMPUTE summe = x + 5.
PRINT summe.
END MATRIX.

```

Entsprechend resultiert im Ausgabefenster:

### Run MATRIX procedure :

```

SUMME
  8  20  17
 10   7  14
 15  12   6

----- END MATRIX -----

```

Soll eine **Multiplikation** (\*) zweier Matrizen gemäß den Regeln der Matrizenrechnung vorgenommen werden, so muss die Anzahl der Spalten der ersten Matrix genauso groß sein wie die Anzahl der Zeilen der zweiten Matrix. Ist eine der beiden Matrizen ein Skalar, so wird jedes Element der Matrix mit dem Skalar multipliziert (Skalarprodukt).

Durch Angabe von / (alternativ: &/) führt SPSS eine **elementweise Division** der Matrizen durch, d.h. jedes Element der ersten Matrix wird durch das entsprechende Element der zweiten Matrix dividiert. Hier müssen die Matrizen von gleicher Dimension sein, beziehungsweise eine Matrix muss ein Skalar darstellen.

Durch Angabe zweier Sterne (\*\*) können wir die **Exponentiation** einer Matrix vornehmen. Hierbei müssen wir beachten, dass Matrizen immer nur ganzzahlig potenziert werden können. Ist der Exponent negativ, so wird die Ergebnismatrix invertiert.

Analog zur elementweisen Division kann auch eine elementweise Multiplikation oder Exponentiation durchgeführt werden. In beiden Fällen müssen die Matrizen wiederum von gleicher Dimension sein, beziehungsweise eine Matrix muss ein Skalar darstellen. Die entsprechenden arithmetischen Operatoren lauten &\* (elementweise Multiplikation) und &\*\* (elementweise Exponentiation).

Im folgenden Beispiel wird eine elementweise Multiplikation der beiden Matrizen  $x$  und  $y$  durchgeführt.

```

MATRIX.
  COMPUTE x = {3,15,12;5,2,9;10,7,1}.
  COMPUTE y = {3,1,2;5,2,9;11,7,16}.
  COMPUTE mult = x &* y.
  PRINT mult.
END MATRIX.

```



Das Syntax-Programm liefert die folgende Ausgabe:

**Run MATRIX procedure:**

```
MULT
  9   15  24
 25   4   81
110  49  16

----- END MATRIX -----
```

Die Prioritätenregel für arithmetische Operatoren findet in Matrix-Programmen analoge Anwendung (vgl. Kapitel 3.2). Eine elementweise Multiplikation weist dieselbe Priorität auf wie eine einfache Multiplikation.

Häufig soll bei einfachen arithmetischen Operationen nicht die gesamte Matrix sondern lediglich einzelne Vektoren oder gar Elemente modifiziert werden. Natürlich wäre es denkbar, einen entsprechenden Befehl mit `COMPUTE` zu formulieren. Wir können diese Eingabe jedoch vereinfachen, indem wir einen Zeilen- oder Spaltenvektor einer Matrix unmittelbar ansprechen. Hierfür setzen wir direkt hinter den Matrizennamen einen in Klammern gesetzten Index, in diesem Fall einen Doppelpunkt.

Allgemein bezeichnet  $A(i, :)$  die  $i$ -te Zeile der Matrix  $A$  und  $A(:, j)$  die  $j$ -te Spalte der Matrix  $A$ . Des Weiteren können wir auch ein einzelnes Element einer Matrix oder eines Vektors ansprechen. Hierfür benutzen wir als Index konkrete Zeilen-/ Spaltenkombinationen.  $A(i, j)$  bezeichnet allgemein das Matrixelement der  $i$ -ten Zeile und  $j$ -ten Spalte der Matrix  $A$ . Diese Notation ist natürlich auch auf Vektoren anwendbar:  $V(i)$  bezeichnet das  $i$ -te Element des Vektors  $V$ .

Neben den hier aufgeführten arithmetischen Operatoren ermöglicht SPSS den elementweisen Vergleich zweier Matrizen durch relationale und logische Operatoren. Die Matrizen müssen die gleiche Dimension aufweisen oder eine Matrix muss ein Skalar sein. Das Ergebnis ist eine Matrix von der gleichen Dimension wie die Operanden, welche die Werte 1 (wahr) oder 0 (falsch) enthält. In der folgenden Syntax bewirkt die Anweisung `COMPUTE err = x = 5`, dass jedes Element der Matrix  $x$  mit dem Wert 5 verglichen wird. Ein Element der Matrix  $err$  nimmt genau dann den Wert 1 an, wenn das entsprechende Element der Matrix  $x$  die Ausprägung 5 annimmt, sonst 0.

```
MATRIX.
  COMPUTE x = {3,15,12;5,2,9;10,7,1}.
  COMPUTE err = x = 5.
  PRINT err.
END MATRIX.
```