

Felix Brosius

Syntaxbeispiele  
zum Download



# SPSS- Programmierung

**Effizientes Datenmanagement  
und Automatisierung mit SPSS-Syntax**

Einführung in die Programmierung mit SPSS-Syntax, Dateimanagement und Datenaustausch

Berechnung, Aufbereitung  
und Analyse von Daten

Makro-Programmierung und  
Automatisierung von Programmabläufen

# Einleitung und Überblick

## 1.1 Arbeiten mit der SPSS-Befehlssprache

SPSS ist ein umfassendes Programm Paket für die statistische Datenanalyse, das den gesamten Analyseprozess von der Dateneingabe über die Datenaufbereitung bis hin zur Ausführung der statistischen Verfahren und dem Erstellen von Grafiken ermöglicht. Alle diese Schritte lassen sich bei SPSS wahlweise über die Dialogfelder der grafischen Benutzeroberfläche oder über Programmcode in einer SPSS-eigenen Befehlssprache ausführen. Beide Wege haben ihre spezifischen Vor- und Nachteile, und die effizienteste Arbeitsweise besteht zumeist darin, sie sinnvoll miteinander zu kombinieren.

Dies ist bei SPSS besonders einfach, weil man jederzeit zwischen der Verwendung von Dialogfeldern und dem Schreiben und Ausführen von Programmcode wechseln und damit sehr bequem einzelne Schritte über die grafische Oberfläche und dann wieder weitere Schritte über die Befehlssyntax ausführen kann. Ferner bietet SPSS die äußerst hilfreiche Möglichkeit, Befehle mithilfe der Dialogfelder zu spezifizieren und diese Einstellungen anschließend in Programmsyntax übersetzen zu lassen. Dies macht insbesondere den Einsteig in die Arbeit mit der Programmsyntax besonders einfach. So wird man auch ohne tiefe Kenntnis der SPSS-Befehlssprache und ohne einschlägige Erfahrungen mit anderen Programmiersprachen sehr schnell in die Lage versetzt, Befehlscode zu erstellen und die Vorteile des Programmierens zu nutzen.

Der typische Ablauf einer Datenanalyse mit SPSS umfasst drei Schritte, die jeder für sich mehr oder weniger umfangreich sein können: Im ersten Schritt müssen die benötigten Daten aus der oder den Datenquellen eingelesen, in das SPSS-Format überführt und als DatenSet für die laufende SPSS-Sitzung bereitgestellt werden. Im zweiten Schritt sind die Daten für die anstehende Analyse aufzubereiten, indem beispielsweise fehlende Werte speziell markiert, neue Variablen berechnet oder bestehende Variablen kategorisiert werden. Erst danach kann im dritten Schritt die eigentliche Datenanalyse erfolgen, indem auf Basis der Daten statistische Kennzahlen berechnet oder aufwendigere Analyseverfahren wie eine Regressions- oder Clusteranalyse durchgeführt werden. Dabei kann wohl jeder Analyst aus seiner praktischen Arbeit von der leidvollen Erfahrung berichten, dass gerade die ersten beiden Schritte einen hohen Arbeitsaufwand erfordern, bevor man mit

den eigentlich interessierenden Analyseverfahren beginnen kann. Deshalb ist es gerade für diese vorbereitenden Arbeitsschritte besonders hilfreich, wiederkehrende Abläufe in der Programmiersprache von SPSS aufzuschreiben, um sie fortan mit geringem Aufwand wiederholt ausführen zu können.

SPSS ermöglicht es dabei nicht nur, umfassende Analyseprozesse vom Einlesen der Daten über die Datenaufbereitung bis zur Anwendung der statistischen Verfahren in der Befehlssprache zu formulieren und als Programm abzuspeichern, so dass der gesamte Prozess jederzeit in identischer Form wiederholt werden kann. Mit verschiedenen Automatisierungstechniken trägt SPSS darüber hinaus dem Umstand Rechnung, dass sich bestimmte Aufgaben zwar regelmäßig wiederholen, dabei aber immer wieder leichte Modifikationen auftreten, so dass beispielsweise verschiedene Datenquellen, abweichende Filterkriterien oder unterschiedliche Stichprobengrößen zur Anwendung kommen sollen. Derartige Variationen lassen sich in dem Automatisierungsprozess berücksichtigen, indem zu Beginn des Programms entsprechende Parameter wie die Datenquelle oder die gewünschte Stichprobengröße abgefragt und anschließend in dem Programm verarbeitet werden.

Die möglichen Automatisierungen verschiedener Art kann man so weit optimieren, dass die selbst erstellten Programme auch von Anwendern ausgeführt werden können, die keinerlei Erfahrung im Umgang mit SPSS haben. So lassen sich derartige Programme beispielsweise mit einem Symbol auf dem Desktop von Windows verknüpfen und entsprechend durch Doppelklick auf das Symbol starten, ohne dass zuvor SPSS gestartet wurde. Das Programm fragt dann automatisch zu Beginn die notwendigen Parameter ab, führt im Hintergrund die entsprechenden Befehle aus und schreibt die Ergebnisse in eine zuvor festgelegte Datei, wobei SPSS während des gesamten Ablaufs praktisch nicht in Erscheinung tritt.

## 1.2 Zum Aufbau dieses Buches

In den folgenden Kapiteln dieses Buches wird die Verwendung der Befehlssprache für ein effizientes Datenmanagement mit SPSS erläutert. Der Befehlsumfang der Programmiersprache von SPSS hat sich in der Vergangenheit von einer SPSS-Version zur nächsten nur in sehr geringem Maße verändert. Die Darstellungen basieren auf der Version von SPSS 16, gelten aber in weiten Teilen ebenso für ältere und mutmaßlich auch für spätere Programmversionen. Die folgende Übersicht skizziert den Aufbau dieses Buches und damit zugleich das Leistungsspektrum der Befehlssprache von SPSS. Für einen schnellen Einstieg in die Nutzung der Befehlssprache finden Sie im nächsten Abschnitt 1.3 ab Seite 20 ein einführendes Beispiel, das Sie ohne Vorkenntnisse über die SPSS-Syntax direkt am PC nacharbeiten können.

## Beispiele in diesem Buch

Auch in den weiteren Kapiteln werden zahlreiche Beispiele zur Erläuterung der Befehlssprache verwendet. Jedes dieser Beispiele finden Sie als Syntaxdatei auf der beiliegenden CD-ROM in dem Verzeichnis *Listings*, so dass Sie den Programmcode aus dem Buch nicht abtippen müssen, wenn Sie ein Beispiel ausprobieren, übernehmen oder abwandeln möchten. Ferner finden Sie in dem Verzeichnis *Daten* sämtliche Datendateien, die in den folgenden Beispielen verwendet werden. In jenen Beispielen, die explizit auf eine bestimmte Datendatei zugreifen, wird zumeist unterstellt, diese befände sich im Verzeichnis C:\Daten; wenn Sie ein solches Beispiel auf Ihrem Rechner ausführen möchten, sollten Sie daher entweder dieses Verzeichnis einrichten und dort die betreffenden Datendateien ablegen oder den Dateizugang im Programmcode entsprechend anpassen.

## Grundlagen für die Arbeit mit SPSS

In Kapitel 2 werden die Grundlagen zur Bedienung von SPSS und für die Arbeit mit der Befehlssprache dargestellt. Dies umfasst die Verwaltung von Syntaxdateien und die Arbeit mit dem Syntaxeditor, der nicht nur zum Schreiben, sondern auch zum Ausführen des Befehlscode dient. Ferner lernen Sie hier die allgemeinen Syntaxregeln der SPSS-Programmiersprache und erfahren, welche Hilfestellungen SPSS beim Erstellen von Syntaxprogrammen bietet. Jene Anwender, die im Umgang mit SPSS bereits geübt sind und über erste Erfahrungen mit der Befehlssprache verfügen, können dieses Kapitel überspringen.

Kapitel 3 geht näher darauf ein, wie SPSS bei der Ausführung einzelner Befehle und längerer Befehlsfolgen verfährt. Diese Kenntnisse sind notwendig, um die Befehlssprache effizient verwenden zu können und unerwartete Fehler beim Ausführen von Syntaxbefehlen zu vermeiden. Anschließend werden weitere Techniken für effizientes Programmieren wie die Verwendung temporärer Variablen und temporärer Transformationen sowie typische Fehlerquellen bei der Arbeit mit der SPSS-Syntax vorgestellt. Für ein effizientes Arbeiten mit der Befehlssprache ist die Kenntnis der in diesem Kapitel zusammengestellten Techniken unverzichtbar; wenn Sie jedoch gerade erst mit dem Erlernen der Programmiertechniken bei SPSS beginnen, sollten Sie sich hier zunächst nur einen Überblick über die Möglichkeiten und damit ein Problembewusstsein verschaffen, um später bei Bedarf wieder gezielt nachschlagen zu können.

## Daten einlesen und Datendateien verwalten

Alle Daten, die mit SPSS analysiert werden sollen, müssen zunächst in ein fest vor-gegebenes Schema überführt und als aktives DataSet in der jeweiligen SPSS-Sitzung bereitgestellt werden. Hierzu gibt es bei SPSS ein eigenes Dateiformat für Datendateien, in denen gemeinsam mit den reinen Daten auch Meta-Informatio-

nen wie Variablen- und Wertelabels oder die Definition fehlender Werte gespeichert werden. Kapitel 4 beschreibt die Handhabung dieser Datendateien und zeigt, wie sie mit der Befehlssprache von SPSS geöffnet, gespeichert und gelöscht werden können. Hier werden auch die Techniken und Befehle zur Handhabung mehrerer gleichzeitig geöffneter Datendateien vorgestellt. In Kapitel 5 wird beschrieben, wie sich neue Datendateien mithilfe der Befehlssprache erstellen und Daten über die Syntax »eingeben« oder berechnen lassen.

Befinden sich die zu analysierenden Daten noch nicht in einer SPSS-Datendatei, sondern in einer externen Datenquelle, können sie nach SPSS importiert werden. Dabei unterstützt SPSS nahezu alle wichtigen Dateiformate wie Excel-, SAS- und einfache Textdateien. Das Vorgehen zum Einlesen von Daten aus solchen Fremddateien wird in Kapitel 6 erläutert. Ferner können Daten aus jeder ODBC-fähigen Datenquelle eingelesen werden; die Vorgehensweise hierzu finden Sie in Kapitel 7 dargestellt.

### **Variablen definieren**

Die Datendateien bei SPSS folgen stets einer klar vorgegebenen Struktur, deren zentrales Merkmal ein festes Raster aus Variablen und Fällen (Datensätzen) ist. Diese Struktur und die sinnvolle Anordnung der Daten in einer Datendatei werden in Kapitel 8 erläutert. Anschließend werden hier die Techniken zur Handhabung der Variablen in einer Datendatei beschrieben. Dies umfasst das Erstellen, Löschen, Kopieren und Sortieren von Variablen sowie das Ändern der Variableneigenschaften wie die Definitionen über Werte- und Variablenlabels, Zahlenformate und fehlende Werte.

### **Variablen berechnen**

Ein großes Thema bei der Arbeit mit SPSS ist die Berechnung von Variablen. Dies beschränkt sich nicht nur auf numerische Variablen, mit denen tatsächlich »gerechnet« werden kann, sondern gilt auch für Textvariablen, die Sie ebenfalls den unterschiedlichsten Transformationen unterwerfen können. So lassen sich über die »Berechnung von Variablen« bestehende Variablen umkodieren, die Werte aus verschiedenen Variablen in unterschiedlicher Weise zusammenfassen, Textwerte aufbereiten, verknüpfen oder zerlegen, Datumswerte verarbeiten oder Zufallszahlen generieren. Hier bietet SPSS ein sehr breites Spektrum an möglichen Transformationen, für die zahlreiche Funktionen zur Verfügung stehen. So gibt es einen Befehl, `compute`, mit dem sich nahezu alle denkbaren Berechnungen und Transformationen durchführen lassen; diesem Befehl ist das gesamte Kapitel 9 gewidmet. Das nachfolgende Kapitel 10 beschreibt dann einige spezifischere Befehle, mit denen Sie bedingte Berechnungen für ausgewählte Datensätze durchführen und andere spezielle Variablentransformationen veranlassen können.

## Fälle aufbereiten

Kapitel 11 erläutert die Techniken zum Sortieren, Filtern und Gewichten der Fälle in einer Datendatei. Ferner werden hier verschiedene Wege zur Identifizierung von Dubletten unter den Fällen der Datendatei beschrieben. In Kapitel 12 finden Sie Techniken zum Gruppieren und Aggregieren eines DatenSets.

## Datendateien zusammenführen und umstrukturieren

SPSS stellt strenge Anforderungen an die Anordnung der Daten innerhalb einer Datendatei. Zu diesen Anforderungen gehört unter anderem, dass eine Variable jeweils einer Spalte entspricht, während jede Zeile einen Fall repräsentiert. Ferner müssen sämtliche Daten, die gemeinsam analysiert werden sollen, auch gemeinsam in einem DataSet vorliegen; sind die Daten bisher in verschiedenen Dateien gespeichert, müssen sie daher vor der Analyse in einem DataSet zusammengeführt werden. Die Techniken hierzu sowie zum Transponieren und Umstrukturieren von Datendateien werden in Kapitel 13 vorgestellt.

## Bedingungen, Wiederholungen und Schleifen

Zu den fortgeschritteneren Programmiertechniken bei SPSS gehört die Formulierung von Bedingungen, Wiederholungen und Schleifen. Diese Techniken werden in Kapitel 14 behandelt und ermöglichen es, Transformationen auf bestimmte Fälle der Datendatei zu beschränken, iterative Berechnungen durchzuführen und gleichartige Berechnungen in effizienter Weise auf mehrere Variablen anzuwenden.

## Daten exportieren

Die Daten aus einem geöffneten DataSet können nicht nur jederzeit als SPSS-Datendatei gespeichert, sondern auch in Dateien mit Fremdformat exportiert werden. So lassen sich die Daten als Excel- oder SAS-Datei oder auch in einfachem Textformat speichern oder via ODBC in eine Datenbank schreiben. Die Techniken hierzu werden in Kapitel 15 erläutert.

## Statistische Prozeduren

Auch wenn die zentrale Leistung von SPSS in der Ausführung statistischer Verfahren wie der Berechnung von Chi-Quadrat- und T-Tests oder der Durchführung von Regressions- und Clusteranalysen besteht, nehmen die statistischen Prozeduren in diesem Buch nur einen sehr geringen Platz ein. In Kapitel 16 wird die Vorgehensweise zur Durchführung statistischer und grafischer Prozeduren an wenigen Beispielen grob skizziert, ohne dass das breite Spektrum der zur Verfügung stehenden Verfahren auch nur annähernd abgedeckt wird. Durch die Möglichkeit, statistische Prozeduren mithilfe der Dialogfelder zu spezifizieren und in Syntax übersetzen zu lassen, ist es aber sehr einfach, die Programmiertechniken auf alle weiteren statistischen Verfahren zu übertragen.

## **Makros**

Um die Möglichkeiten der SPSS-Syntax wirklich ausnutzen zu können, ist der Einsatz von Makros unverzichtbar. Ein Makro ist so etwas wie eine benutzerdefinierte Funktion, die als »intelligenter Textbaustein« fungiert. Mithilfe von Makros können Routineaufgaben als Baustein gespeichert und in einfacher Weise in eine Befehlsfolge eingebunden werden. Ferner ermöglichen es Makros, die Ausführung einzelner Programmschritte von vorgegebenen Bedingungen abhängig zu machen und ausgewählte Befehlsfolgen in einer Schleife wiederholt ausführen zu lassen. Die Erstellung und Verwendung von Makros wird in Kapitel 17 erläutert. Eine Reihe von praktischen Anwendungsbeispielen für Makros finden Sie in Kapitel 18.

## **Automatisierungstechniken**

Kapitel 19 stellt verschiedene Automatisierungstechniken vor. Damit können Sie feste Programmbausteine in externe Dateien auslagern und in eine Befehlsfolge einbinden. Mithilfe des »Produktionsmodus« von SPSS können Programme automatisiert ablaufen, ohne dass SPSS explizit gestartet wird. Ferner ist es möglich, Syntaxprogramme in SPSS mit einer Schaltfläche zu verknüpfen und damit den Aufruf häufig benötigter Programme zu vereinfachen. Die Ergebnisse statistischer Prozeduren können nicht nur als Tabellen in eine Ausgabedatei geschrieben, sondern auch in eine Datendatei umgelenkt werden. Dadurch lassen sich die Ergebnisse auf einfache Weise für weitere Berechnungen verwenden.

## **Grundeinstellungen**

Zahlreiche Grundeinstellungen von SPSS steuern dessen Verhaltensweise bei der Ausführung von Syntaxbefehlen. Diese Grundeinstellungen lassen sich sowohl über den Menübefehl *Bearbeiten / Optionen* als auch über Syntaxbefehle ändern. Die Vorgehensweise zum Ändern dieser Einstellungen über die Syntax ist für die wichtigsten Grundeinstellungen in Kapitel 20 beschrieben.

## **1.3 Zum schnellen Einstieg: Ein erstes Programm schreiben und ausführen**

Im Folgenden werden an einem einfachen Beispiel die Schritte aufgezeigt, die notwendig sind, um ein Syntaxprogramm zu schreiben und auszuführen. Hierzu soll die Befehlsfolge aus Listing 1.1 ausgeführt werden. Da die Befehlsnamen in SPSS zum überwiegenden Teil sprechend sind, lässt sich auch ohne Kenntnis der Syntaxsprache erkennen, dass das Programm aus Listing 1.1 einen vollständigen einfachen Analyseprozess abbildet – beginnend mit dem Einlesen der Daten über einfache Schritte zur Datenaufbereitung bis zur Erstellung einer ersten Statistik, in diesem Fall einer Häufigkeitstabelle.

```

GET FILE ='C:\Daten\alter.sav' .
DATASET NAME altersdaten .

RECODE alter
  (18 THRU 35=1) (36 THRU 50=2) (50 THRU HIGHEST=3)
  (MISSING = -1) (ELSE =0)
INTO alterkat .

VARIABLE LABELS alterkat 'Alterskategorie' .

VALUE LABELS alterkat -1  'Unbekannt'
  0  'Kids'
  1  'Jung'
  2  'Mittel'
  3  'Alt' .

FREQUENCIES
  VARIABLES=alterkat .

```

**Listing 1.1:** Befehlsfolge zum Einlesen, Aufbereiten und Analysieren von Daten

Um das Programm aus Listing 1.1 zu erstellen und auszuführen, gehen Sie folgendermaßen vor:

- *Vorbereitungen.* Starten Sie SPSS und stellen Sie sicher, dass der Dateneditor im aktiven DataSet keine Daten enthält, die Sie noch benötigen und die noch nicht gespeichert sind. Stellen Sie ferner sicher, dass die Datendatei *alter.sav* von der beiliegenden CD-ROM entweder auf der Festplatte oder im CD-Laufwerk verfügbar ist, so dass sie im Folgenden in SPSS eingelesen werden kann.
- *Syntaxdatei öffnen.* Öffnen Sie eine leere Syntaxdatei, zum Beispiel mit dem Befehl *Datei / Neu / Syntax*.
- *Programm schreiben.* Geben Sie in die Syntaxdatei die Befehlsfolge aus Listing 1.1 ein. Wenn Sie die Befehle nicht abtippen möchten, können Sie auch die Datei *MeinErstesProgramm.sps* von der beiliegenden CD-ROM öffnen (Befehl *Datei / Öffnen / Syntax*).
- *Programm anpassen.* Ersetzen Sie in dem Programmcode in der ersten Zeile die Pfadangabe *C:\Daten\alter.sav* durch den Speicherort, an dem sich die Datei *alter.sav* auf Ihrem Rechner befindet.

- **Programm starten.** Markieren Sie wie in Abbildung 1.1 die gesamte Befehlsfolge in der Syntaxdatei und wählen Sie anschließend den Befehl *Ausführen / Alles* oder klicken Sie in der Symbolleiste auf die Schaltfläche mit dem Pfeil. Dadurch werden die insgesamt fünf Befehle aus Listing 1.1 ausgeführt.

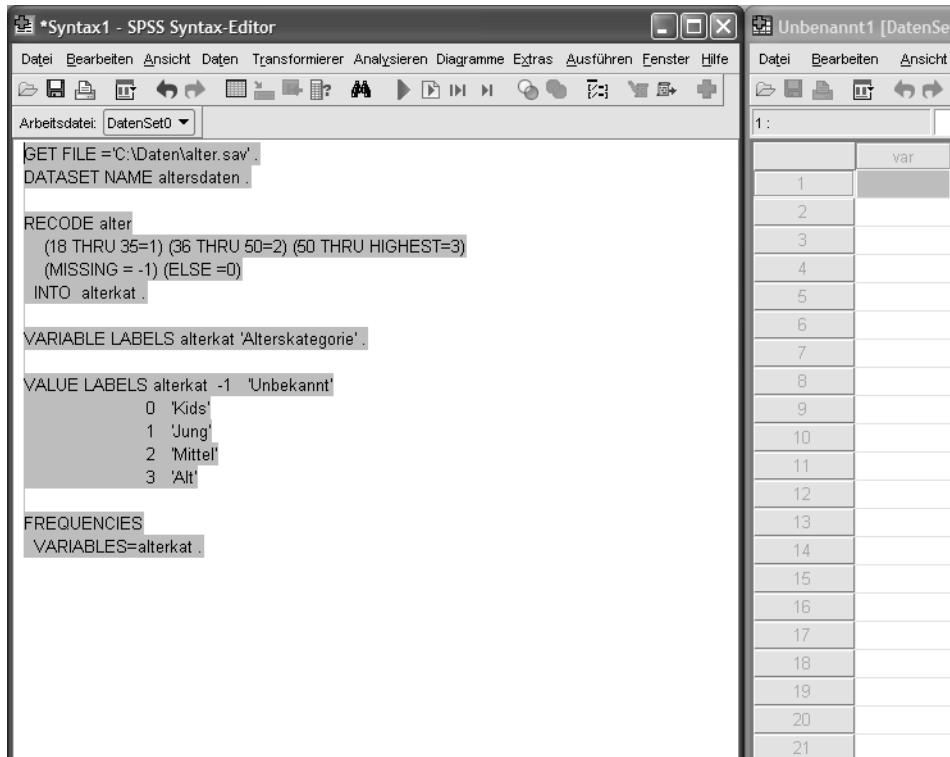


Abb. 1.1: Grafische Oberfläche von SPSS mit Syntaxdatei und leerem Dateneditor

Wenn Sie das Programm wie oben beschrieben starten, werden die einzelnen Befehle von SPSS abgearbeitet und damit folgende Analyseschritte ausgeführt:

- Der Befehl `get file` liest die SPSS-Datendatei `alter.sav` in den Dateneditor ein, so dass sie anschließend als aktives `DataSet` zur Verfügung steht. Diesem `DataSet` wird für die laufende SPSS-Sitzung mit dem Befehl `dataset name` der Name `altersdaten` zugewiesen; unter diesem Namen kann das `DataSet` nun jederzeit angesprochen werden, auch wenn weitere `DataSet`s geöffnet sind und `altersdaten` nicht mehr das aktive `DataSet` bildet.
- Der Befehl `recode` kodiert die Werte der Variablen `alter` um und schreibt die resultierenden Werte in die Variable `alterkat`. Diese Variable wird hierbei automatisch neu erstellt; sollte jedoch eine gleichnamige Variable in der Datendatei bereits vorhanden sein, würde diese ohne einen Hinweis überschrieben

werden. Beim Umkodieren werden fehlende Werte in den Wert -1 überführt, alle Werte zwischen 18 und 35 in 1 etc.

- Durch den Befehl **variable labels** erhält die Variable **alterkat** das Variablenlabel **Alterskategorie**.
- Entsprechend werden mit dem Befehl **value labels** die fünf angeführten Wertelabels für die Variable **alterkat** definiert.
- Der Befehl **frequencies** erstellt schließlich eine Häufigkeitstabelle für die Variable **alterkat**. Die Tabelle wird in die aktuelle Ausgabedatei geschrieben, vgl. Abbildung 1.2. Wenn noch keine Ausgabedatei geöffnet ist, wird von SPSS automatisch eine neue Ausgabedatei angelegt.

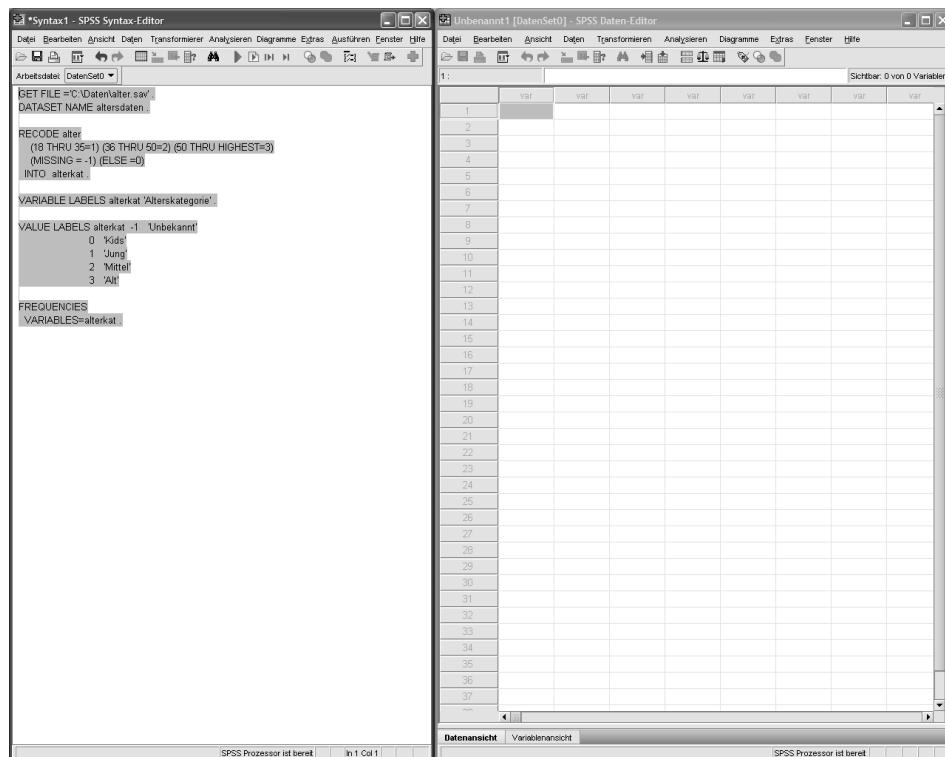


Abb. 1.2: Ergebnis des **frequencies**-Befehls aus Listing 1.1 in der Ausgabedatei