

Zusammenfassung

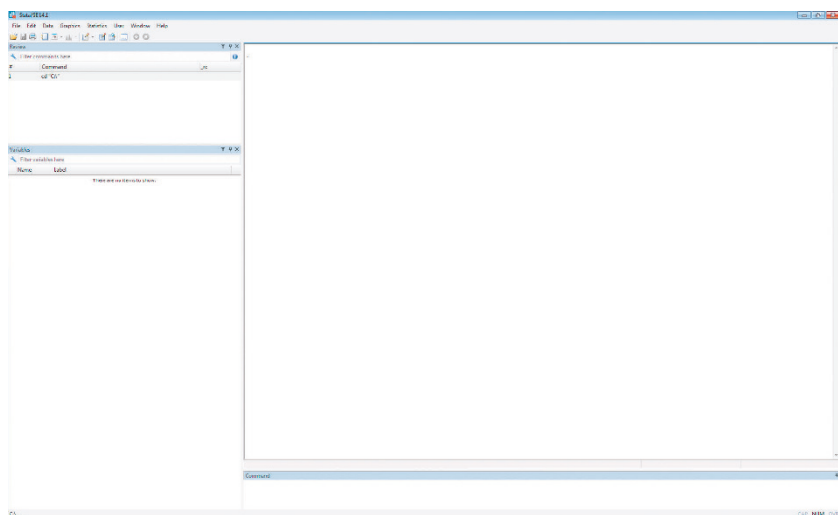
Dieses Einführungskapitel gibt eine knappe Übersicht über die Grundlagen der Kommandosprache in Stata als Basis für deren laufende Ergänzung im Rahmen der Analyse von SEM. Zusätzlich werden einige Anmerkungen und Tipps zur praktischen Anwendung von Stata generell gegeben. Ein Fokus richtet sich zuletzt auf Datenformate zur Analyse von SEM in Stata, nämlich die Möglichkeit der Analyse zusammengefasster Parameter (*summary statistics*).

Der folgende Abschnitt bietet einen sehr knappen Überblick über den Aufbau und wichtige Befehlselemente in **Stata**. Leserinnen und Leser, die mit der **Kommandosprache** und Datenanalyse in Stata bereits gut vertraut sind, können diesen Teil daher überspringen und zu Kapitel 3 übergehen.

Die Oberfläche von Stata besteht grundsätzlich aus verschiedenen nebeneinander stehenden Fenstern (s. Abbildung 4). Die Oberfläche setzt sich für gewöhnlich zusammen aus:

- Befehlsfenster (Eingabe von Kommandos): *Command*
- Review-Fenster (Rückblick auf Eingaben): *Review*
- Variablenliste (Variablennamen und deren Beschreibung): *Variables*
- Ergebnisfenster (Output der Analysen und Berechnungen)

Abbildung 4 Benutzeroberfläche in Stata



2.1 Die Kommandosprache: Stata-Syntax

Alle Befehlselemente der **Kommandosprache** in Stata (= Befehle) werden im Folgenden jeweils mittels der Schriftart *Courier* hervorgehoben. In diesem Skriptum wird, wie bereits erwähnt, deren Verwendung über das Befehlsfenster (*Command*) oder sogenannte do-Files in Stata nahe gelegt. Es sei jedoch darauf hingewiesen, dass sich alternativ ein Großteil der Operationen auch über das Menü und den eigens entwickelten „SEM Builder“ als grafische Eingabe in Stata durchführen ließe.

Generell bietet die Software Stata eine vergleichsweise einfache **Kommandosprache**. Alle Befehle haben die folgende allgemeine Form (= Syntaxdiagramm):

```
command [varlist] [if] [in] [weight] [, options]
```

Elemente ohne Klammern bzw. in runden Klammern bedeuten, dass diese erwähnt werden müssen. Elemente in eckigen Klammern sind erlaubt, können also, müssen aber nicht angegeben werden. Nicht erlaubte Elemente werden im Syntaxdiagramm nicht genannt. Darüber hinaus sind jeweils erlaubte **Abkürzungen des Befehls** oder seiner Optionen durch Unterstreichungen gekennzeichnet, wie z. B. help (Befehl zum Aufruf der Hilfe-Funktion).

Das Element `[varlist]` steht für einen oder mehrere **Variablen**namen. Für beliebige Zufallsvariablen wird auch im Folgenden die Beschreibung *varname* oder *varlist* verwendet. Mehrere Variablen werden durch Leerzeichen getrennt oder über Symbole angegeben: von bis „-“ sowie eine beliebige Erweiterung „?“ oder mehrere beliebige Erweiterungen des Variablennamens „*“. Abkürzungen der Variablennamen sind ebenfalls erlaubt, solange diese eindeutig zuordenbar sind.

Das Element `[if]` beschreibt eine **Bedingung** („...wenn zutrifft, dass...“), die den Befehl auf bestimmte Beobachtungen einschränkt und somit eine Art Filter ist. Ausdrücke in der if-Bedingung können mit Hilfe von Operatoren und Funktionen definiert werden (s. `help operators`).

Das Element `[in]` beschreibt ebenfalls eine Bedingung, die den Befehl jedoch auf bestimmte Beobachtungen laut der aktuellen **Sortierung** im Datensatz beschränkt. Die in-Bedingung ist somit immer nur in Verbindung mit einer zuvor definierten Sortierung der Daten, z. B. mittels `sort`, sinnvoll.

Das Element `[weight]` beschreibt grundsätzlich vier mögliche **Gewichtungstypen** (s. `help weights`), wie etwa Wahrscheinlichkeitsgewichte `[pweight]` zur Korrektur ungleicher Auswahlwahrscheinlichkeiten. Je nach Befehl sind unterschiedliche Gewichtungstypen erlaubt. Die Verwendung von Gewichten für den `sem` Befehl ist bspw. in Kombination mit `[pweight]` sowie der Definition eines konkreten Stichprobendesigns (s. `help svyset`) und dem Präfix `svy:sem` möglich.

Das Element `[, options]` beschreibt alle zusätzlichen **Optionen** zum Befehl und oftmals wichtige Erweiterungen in der Analyse. Diese werden jedenfalls immer nach einem Komma angegeben. Im entsprechenden Hilfenü des Hauptbefehls werden die Optionen, deren Funktion im Detail sowie mögliche Abkürzungen der Befehle erläutert.

2.2 Praktische Anmerkungen und Tipps

Die praktische Verwendung der Kommandosprache in Stata mit Fokus SEM wird in den folgenden Kapiteln laufend eingeführt und ergänzt. Dennoch seien hier einige hilfreiche Anmerkungen und Tipps allgemeiner Natur erwähnt:

- Der Befehl `help` führt immer in das **Hilfefenster** und kann mit spezifischen Kommandos verbunden werden, z. B. `help regress`.
- Stata unterscheidet zwischen **Groß- und Kleinschreibung** (*case sensitivity*), d. h. „sem“ und „SEM“ ist nicht ident. Dies ist insofern von Bedeutung als der `sem` Befehl defaultmäßig Variablen, die mit **Großbuchstaben** beginnen, als latente und daher nicht gemessene Variablen erkennt. Diese Grundeinstellung

kann im Zuge des `sem` Befehls mit der Option `sem paths ...`, `nocapslatent` aufgehoben werden.

- Im Befehlsfenster können über die Tastatur mit Bild↑ und Bild↓ zur **Wiederholung** alle eingegebenen Befehle wieder aufgerufen werden. Auch kann hierzu das Review-Fenster herangezogen werden.
- **Befehle** (Kommandos) können in Stata über das Befehlsfenster eingegeben werden oder über sogenannte do-Files (einfache Textfiles), die über `doedit` aufgerufen werden. Hier gilt schlichtweg der Vorteil der Nachvollziehbarkeit. Das gesamte do-File oder Zeilen daraus können schließlich mit dem Kürzel `Strg+D` ausgeführt werden.
- Bei langen Befehlen in do-Files eignen sich **Zeilenumbrüche** mittels der folgenden Zeichenabfolge: `///`
- Eine einfache **Suche** nach einem Variablen-Label, d.h. der Bezeichnung einer Variablen, ermöglicht bspw. der Befehl `lookfor string`.
- Zum Anzeigen längerer **Outputs**, anstatt einer schrittweisen Ausgabe mit Stopps (`-more-` im Ergebnisfenster), kann man `set more off`, `permanently` einstellen.
- Nachdem ein Befehl ausgeführt wurde, werden **Ergebnisse** (Koeffizienten, Modellparameter, etc.) immer kurzfristig gespeichert und können mittels `return list` (Koeffizienten) oder `ereturn list` (Modellparameter, Schätzergebnisse) aufgelistet werden.
- Koeffizienten aus gespeicherten **Modellergebnissen** sind mittels `display` (für Skalare) sowie `matrix list` (für Matrizen) einsehbar und können weiter verwendet werden.
- Von Nutzerinnen oder Nutzern geschriebene zusätzliche bzw. spezialisierte Befehle (**Zusatzpakete**) in Stata (Ado-Files) können über den Befehl `net search word` im **Internet** gesucht und danach selbst installiert werden.

2.3 Datenformate: Rohdaten und zusammengefasste Daten

Stata kann, so wie andere Software-Pakete auch, verschiedenste Formate von **Rohdaten** einlesen (s. dazu ausführlicher z.B. Kohler & Kreuter, 2012, Kap. 10). Diese mit Stata kompatiblen Datenfiles werden schließlich als `*.dta`-Files gespeichert und können – optional unter Angabe des Speicherortes – aufgerufen werden mit:

```
use filename [, clear nolabel]
```

Eine Besonderheit im Rahmen der Analyse mit dem `sem` Befehl ist, dass auch **zusammengefasste Parameter** aus einem Datensatz (*summary statistics*), wie Kovarianzen, Korrelationen und Mittelwerte sowie die Zahl der Fälle (Beobachtungseinheiten), eingelesen und analysiert werden können. Der Grund ist, dass solche Parameter oftmals mit publiziert werden, um Analysen replizieren zu können (McDonald & Ho, 2002). Hierzu dient der Hauptbefehl `ssd` und wird mit `ssd init varlist` (Liste der Variablen) eingeleitet. Ein Beispiel soll mit fiktiven Daten für drei Variablen (hier: `y`, `x1` und `x2`) die Verwendung veranschaulichen (s. Beispiel 1). Ein weiteres Beispiel zur Erstellung künstlicher Rohdaten aus zusammengefassten Parametern wird in Beispiel 2 (auf S. 30) angeführt.

Wichtig ist, sich zu vergegenwärtigen, dass der beschriebene Datensatz nicht wirklich Informationen über 500 Fälle enthält (Rohdaten), sondern lediglich zusammengefasste Parameter. Sonst übliche Analyseschritte wären somit irreführend und falsch. Da in SEM generell Varianz-Kovarianz-Strukturen und Mittelwertstrukturen analysiert werden, erlaubt der `sem` Befehl jedoch dies zu berücksichtigen und liefert eine korrekte Abbildung der Datenstruktur aus den zuvor beschriebenen Variablen. Eine Angabe, dass es sich um zusammengefasste Parameter handelt, ist im Rahmen des `sem` Befehls daher nicht weiter notwendig.

Beispiel 1 Erstellung fiktiver Daten für Analysebeispiele – Variante 1

```
clear
ssd init y x1 x2
ssd set observations 500
ssd set means 2 3 4
ssd set sd 5 6 7

ssd set correlations 1 \ .4 1 \ .3 .25 1
. ssd list

Observations = 500

Means:
  y  x1  x2
  2   3   4

Standard deviations:
  y  x1  x2
  5   6   7

Correlations:
  y  x1  x2
  1
  .4   1
  .3 .25   1
```

Einführung in lineare Strukturgleichungsmodelle mit
Stata

Aichholzer, J.

2017, XIII, 185 S. 23 Abb., 18 Abb. in Farbe., Softcover

ISBN: 978-3-658-16669-4