

Erste Schritte mit Stata

Fabian Pfeffer
Stephan Lindner
Bernd Weiß

2. September 2004

Zusammenfassung. Stata ist ein Programm zur statistischen Datenanalyse und Datenvisualisierung. In dieser Einführung wird die grundsätzliche Programmlogik vor dem Hintergrund sozialwissenschaftlicher Datenanalyse vorgestellt: Beginnend mit der Datenaufbereitung, deskriptiven und graphischen Verfahren bis hin zu multivariaten Ansätzen.

Inhaltsverzeichnis

1	Was spricht für Stata?	2
2	Einführung in das Programm	3
2.1	Grundlegender Aufbau des Programms	3
2.2	Dokumentation der Datenanalyse	4
2.3	Ergänzung: Ado-Files – the power of Stata	5
3	Das Hilfesystem	5
4	Allgemeine Befehlsstruktur	6
5	Einlesen der Daten und Datenaufbereitung	7
5.1	Einlesen von Daten	7
5.1.1	Einlesen von Daten im Stata-Format	7
5.1.2	Einlesen von Daten im freien Format	8
5.1.3	Direktes Eingeben von Daten	9
5.2	Aufbereitung von Daten	9
5.2.1	Erstellen und Verändern von Variablen	10
5.2.2	Die Optionen „by“, „_n“ und „_N“	10
5.2.3	Subskripte	11
5.2.4	Weitere Befehle und Missings	11
5.2.5	Beschriftung von Variablen	12

6	Deskriptive Analysen	13
6.1	Darstellung des Datensatzes	13
6.2	Einfache univariate Maßzahlen	14
6.3	Tabellen	15
6.4	Korrelationen	18
7	Graphiken	18
7.1	Histogramm	20
7.2	Box-and-Whisker-Plot	21
7.3	Scatterplot	21
8	Multivariate Verfahren: OLS-Regression	22
8.1	Stata-Ausgabe bei einer OLS-Regression	22
8.2	Erweiterungen zur OLS-Regression	24
8.3	Regression-Diagnostik	26
9	Die Variablen der Absolventenstudie	31

1 Was spricht für Stata?

Das Einarbeiten in ein fremdes Programm ist immer ein zeitaufwendiges Unterfangen. Insbesondere, wenn es darum geht, Alternativen zu bestehenden Lösungen zu erarbeiten. Da Zeit eine knappe Ressource ist, soll an dieser Stelle kurz darauf hingewiesen werden, weshalb Stata – vor allem vor dem Hintergrund der Dominanz von SPSS – eine gute Alternative darstellen kann.

- Zunächst lassen sich handfeste finanzielle Vorteile anführen. Während die Kosten für SPSS im vierstelligen Eurobereich liegen, kostet Stata weniger als 1000 Euro. Updates von SPSS sind immer kostenpflichtig. In Stata dagegen reicht – bei bestehender Verbindung zum Internet – ein `update all`, um eine bestehende Version auf den neusten Stand zu bringen.
- Hinzu kommt die Möglichkeit von Stata, von Benutzern zur Verfügung gestellte Funktionen zentral (vom Stata-Server) und konsistent zu installieren – und ebenfalls via `update all` auf dem neusten Stand zu halten. Damit lässt sich Stata einfach um Funktionen erweitern, die von Herstellerseite zunächst nicht zur Verfügung gestellt werden.
- Stata ist ungleich schneller als SPSS.
- Die Syntax von Stata ist unserer Ansicht nach logischer und systematischer aufgebaut als in SPSS – vielleicht mit Ausnahm des Graphik-Moduls.

- Ohne diesen Umstand schon ausreichend beurteilen können, so scheint es so zu sein, dass Stata statistisch das mächtigere Programm ist. So existieren etwa eine Reihe von Routinen aus dem Feld der Meta-Analyse. Weiterhin ist es mit Stata möglich, heteroskedastizitätskonsistente Regressionsmodelle zu schätzen.
- Mit Stata lassen sich komplexe und den jeweiligen Erfordernissen angepasste Graphiken erzeugen. Das bedeutet aber auch, dass die Bedienung alles andere als einfach (und logisch) ist.
- Das Hilfesystem von Stata ist sehr mächtig. Bei einem Anschluss an das Internet lassen sich auch Quellen und FAQs auf <http://www.stata.com> durchsuchen. Viele Hilfeseiten bieten ‚anklickbare‘ Beispiele an, mit denen anschaulich die Wirkungsweise der Befehle demonstriert werden können.
- In einigen Forscherkreisen scheint Stata bereits das Programm der Wahl zu sein.

Während der Beschäftigung mit dem Programm sind mindestens zwei Nachteile offenbar geworden. Zum einen kann Stata nur unzureichend Daten aus anderen Statistikpaketen importieren. Im Prinzip lassen sich nur ASCII-Files einlesen. StataCorp bietet zwar ein Programm namens Stat/Transfer an, mit dem sich eine Vielzahl verschiedener Datenformat importieren und exportieren lassen, doch dieses muss zusätzlich erworben werden.

Für SPSS-User wird vor allem der Umgang mit Missing Values ungewohnt sein – SPSS ist in dieser Hinsicht wesentlich eingängiger zu bedienen.

2 Einführung in das Programm

2.1 Grundlegender Aufbau des Programms

Stata wird entweder durch Aufrufen des Programms oder Öffnen eines Stata-Datensatzes mit der Endung `.dta` gestartet. Es öffnen sich folgende Fenster:

- Eingabefenster (Stata Command): Eingabezeile für Syntax-Befehle
- Ergebnisfenster (Stata Results): Fortlaufender Output der Analyse
- Protokollfenster (Review): Chronologische Liste aller abgeschickten Befehle (kann zur Wiederholung des Befehls angeklickt oder per PageUp/PageDown erreicht werden)
- Variablenfenster (Variables): Auflistung der Variablen des geöffneten Datensatzes

Außerdem steht für die Dateneingabe und Kontrolle über die Symbolleiste eigens ein Data-Editor und Data-Browser zur Verfügung (über Symbolleiste aufzurufen).

Im Folgenden behandeln wir ausschließlich die syntaxgesteuerte Datenanalyse und beziehen uns nicht auf die ebenfalls verfügbare Menüsteuerung. Die Eingabe von Befehlen geschieht in Stata meist erst einmal ‚interaktiv‘, sprich die Befehle werden im Eingabefenster eingetippt und direkt abgeschickt.

2.2 Dokumentation der Datenanalyse

Zur Dokumentation und Ermöglichung der Reproduktion unserer Ergebnisse stehen so-
dann folgende Möglichkeiten zur Verfügung:

- Log-Files zur simultanen Aufzeichnung der abgeschickten Befehle mit oder ohne deren Output.
- Do-Files zur Speicherung fester Befehlssequenzen von Stata-Befehlen (,Syntax-File‘).

Log-Files protokollieren parallel zur Eingabe entweder Befehle und Output (log) oder nur Befehle (cmdlog).

- Start des Protokolls: `log using protocol.log / cmdlog using protokoll.txt`
- Unterbrechen des Protokoll: `log off / cmdlog off`
- Wiederaufnahme des Protokolls: `log on / cmdlog on`
- Schließen des Protokolls: `log close / cmdlog close`

Log-Files (cmdlogs) sind ohne weiteres in Do-Files überführbar! Sie müssen lediglich mit dem Stata-Do-File-Editor (oder auch jeglichem anderen Editor¹) im Format `.do` abgespeichert und eventuell nachträglich bearbeitet werden²:

```
doedit protokoll.txt
```

Die Nachbearbeitung der Protokolldatei zur Erstellung eines sinnvollen Do-Files be-
schränkt sich auf Folgendes:

- Kommentierung: Ganzzeilige Kommentare werden mit `*` eingeleitet; Zitate inner-
halb einer Zeile werden von `/*` und `*/` eingerahmt.
- Mehrzeilige Befehle: Sind Befehle besonders lang, so werden sie in der log-Datei auf
mehrere Zeilen verteilt. Damit der Befehl im Do-File dennoch vollständig ausgeführt
werden kann, muss entweder der Zeilenumbruch gelöscht oder – falls die Zeilen
eben nicht ewig lang werden sollen – ,auskommentiert‘ werden (d.h. `/*` an das
Ende der ersten Zeile und `*/` an den Anfang der zweiten). Beispiele hierzu werden
im Laufe des Vortrags noch vorkommen. (Weitere hilfreiche Hinweise zu wichtigen
Bestandteilen eines Do-Files siehe Kohler/Kreuter 2001: 43-45)

Die Ausführung eines fertigen Do-Files geschieht über Eingabe von `do dateiname`.

¹Zum Beispiel stellt die Kombination von Emacs und dem ESS-Mode ein sehr mächtiges Gespann dar.

²Natürlich kann alternativ zur Verwendung von Log-Files und deren Umwandlung in Do-Files auch die
aus SPSS bekannte Prozedur angewandt werden: Von Anfang könnten die Kommandos im Do-File
eingetragen und von dort ,geschickt‘ werden; damit ginge aber das für Stata kennzeichnende und oft
hilfreiche interaktive Arbeiten über das Eingabefenster verloren.

2.3 Ergänzung: Ado-Files – the power of Stata

Ein weiterer Schritt wäre die Umwandlung eines Do-Files in ein sogenanntes Ado-File. In einem solchen wird die Sequenz der einzelnen Befehle in einen kompilierten neuen Stata-Befehl überführt. Das heißt, das mühsam zusammengeschusterte Befehlswerk wird auf die Stufe eines Stata-Kommandos gehoben. Somit muss nur noch der Name des Ado-Files eingetippt werden und die entsprechenden Analyseschritte laufen im Hintergrund (silent) ab. Der wirklich Clue von Stata: Für alle möglichen Probleme und Problemchen sind im Internet – im open-source-Stil – zahlreiche Ados verfügbar! Eine sehr aktive Stata community stellt eine ganze Bandbreite von Neuerungen in Form solcher Ado-Files zur Verfügung, die nicht selten Eingang in neue Programmversionen finden.

3 Das Hilfesystem

Die Kenntnis des Hilfesystems und der angemessene Umgang damit sind essentiell für die Bedienung von Stata. Da der Aufbau der Hilfsfunktionen aber von der anderer MS-Windows Programmen abweicht (zum Beispiel der von SPSS), ist ein wenig Übung erforderlich.

Zwei Situationen lassen sich unterscheiden, in denen Stata-Anwender auf das Hilfesystem zurückgreifen:

1. Das exakte Stata-Kommando ist *nicht* bekannt, sondern gesucht werden Stata-Funktionen, die bestimmten statistischen Verfahren entsprechen. Für diesen Fall ist das `search` *Suchbegriff* Kommando hilfreich, das es erlaubt, eine Reihe von Hilfequellen nach Stichwörtern zu durchsuchen. Werden mit `search` die Ergebnisse in das „Stata Results“-Fenster ausgegeben, so öffnet der Befehl `findit` ein neues Fenster, das die Ergebnisse etwas übersichtlicher darstellt.
2. Anders sieht es aus, wenn bereits grundsätzliche Kenntnisse über einen Befehl vorhanden sind, aber beispielsweise nicht alle möglichen Parameter bekannt sind. Eine vollständige Beschreibung des Kommandos wird mit `help` *Schlüsselwort* aufgerufen. Die Hilfeseiten werden im „Stata Results“-Fenster dargestellt. Mit dem Befehl `whelp` *Schlüsselwort* wird ein neues Fenster geöffnet.

Der Aufbau dieser Hilfeseiten ist immer gleich: Nach einer Kurzbeschreibung des Kommandos³ sowie einer formalisierten Darstellung der Befehlsstruktur, folgt eine ausführliche Befehlsbeschreibung („Description“), eine ausführliche Erläuterung der zu übergebenden Befehlsparameter („Options“) und einige Beispiele („Examples“) für gültige Befehlsaufrufe. Mit dem Abschnitt „Also see“ wird auf verwandte Befehle oder andere Informationsquellen verwiesen. Einige Ausdrücke in diesen Hilfeseiten sind blau gefärbt (`help`) beziehungsweise unterstrichen (`whelp`). Diese lassen sich mit der Maus anklicken und verweisen auf die entsprechenden Einträge im Hilfesystem.

³Es ist möglich, dass hier auf mehr als ein Kommando verwiesen wird. Ein Beispiel dafür sind die beiden Befehle `do` und `run`. Innerhalb des Abschnitts „Description“ wird näher erklärt, worin sich beide Befehle unterscheiden.

4 Allgemeine Befehlsstruktur

Die Syntax von Stata-Kommandos folgt einer einheitlichen Logik. Am Beispiel des Befehls `summarize` soll dieses erläutert werden. Mit `summarize varlist` lassen sich einfache univariate Statistiken ausgeben, wie Anzahl der gültigen Fälle, Mittelwert, Standardabweichung, Minimum und Maximum; hier am Beispiel der Examensnote (`anote`):

```
. summarize anote
```

Variable	Obs	Mean	Std. Dev.	Min	Max
-----+-----					
anote	584	2.704795	.4881206	1.3	3.8

Die vollständige „Grammatik“ einer Stata-Kommandos ist komplexer. „Den Kern bildet der eigentliche *Befehl*. Ihm können, abgetrennt durch einen Doppelpunkt, *Befehls-Präfixe* vorangestellt werden. Direkt an den Befehl angeschlossen wird eine *Variablenliste*. Danach folgen in beliebiger Reihenfolge die Angaben über die etwaige Gewichtung des Datensatzes (die *Gewichtungsanweisungen*), die *If*-Bedingung und die *In*-Bedingung. Schließlich werden, abgetrennt durch ein Komma, die *Optionen* eingegeben.“ (Kohler / Kreuter 2001: 53). Formal lässt sich folgende Darstellung wählen (Juul 2004):

```
prefix: command varlist if exp in range weight , options
```

Vor allem in Kombination mit `by` (als sogenanntem Präfix) und `if` lassen sich wesentlich kompaktere Programmcodes als etwa in SPSS erzeugen, ohne das dies zu Lasten der Lesbarkeit geht. `by` entspricht in SPSS dem `split`-Befehl und produziert nach Gruppen getrennte Statistiken. Dazu muss allerdings der Datensatz zuvor mit `sort varlist` nach der Gruppierungsvariablen sortiert werden.⁴ Um beispielsweise zu vergleichen, wie sich die Abschlussnoten nach Geschlecht (`mann`) unterscheiden, lauten die beiden Befehle:

```
. sort mann
```

```
. by mann: summarize anote
```

```
-----
```

```
-> mann = Frau
```

Variable	Obs	Mean	Std. Dev.	Min	Max
-----+-----					
anote	153	2.664052	.4492119	1.4	3.7

```
-----
```

```
-> mann = Mann
```

⁴`sort` und `by` lassen sich auch durch `bysort` zusammenfassen.

Variable	Obs	Mean	Std. Dev.	Min	Max
anote	431	2.719258	.5009065	1.3	3.8

Deutlich ist zu erkennen, dass Frauen im Mittel einen etwas besseren Abschluss machen.

`if` ermöglicht es, (temporär) nur eine Subgruppe auszuwerten – ohne gleich den ganzen Datensatz zu reduzieren⁵. Um die Bezüge zu SPSS zu bemühen, entspräche `if` einem `temp`, gefolgt von `select if expression`. Die Berechnung des Mittelwertes nur für Frauen lässt sich wie folgt realisieren:

```
. summarize anote if mann==0
```

Variable	Obs	Mean	Std. Dev.	Min	Max
anote	153	2.664052	.4492119	1.4	3.7

Am Ende dieses Abschnitts soll der Hinweis stehen, dass Stata, sowohl was die Schreibweise der Befehle als auch der Variablen betrifft, zwischen Groß- und Kleinschreibung unterscheidet. Die beiden Ausdrücke `summarize Anote` und `summarize anote` sind nicht identisch.

5 Einlesen der Daten und Datenaufbereitung

Als erste Schritte einer statistischen Untersuchung mit Stata müssen die Daten eingelesen werden; anschließend können neue Variablen erstellt und bereits vorhandene Variablen verändert werden.

5.1 Einlesen von Daten

Beim Einlesen von Daten sind drei Vorgehensweisen zu unterscheiden: das Einlesen von Daten im Stata-Format, das Einlesen von Daten im freien Format (Importieren von Daten) sowie das direkte Eingeben von Daten in Stata.

5.1.1 Einlesen von Daten im Stata-Format

Stata-Files haben die Endung „.dta“. Die Dateien enthalten neben den eigentlichen Daten auch Informationen über Variablenbeschriftung etc.

Stata-Files können mit dem Befehl `use Dateiname` eingelesen werden. Dabei muss der komplette Pfad angegeben werden, wenn sich die Datei nicht im Homeverzeichnis befindet. Alternativ steht auch die Menübar zur Verfügung. Um die Absolventendaten aufzurufen, wird also folgender Befehl eingetippt:

```
use absolventen.dta
```

⁵Das dauerhafte Löschen von Fällen wird durch den Befehl `drop if expression` oder `keep if expression` erreicht.

5.1.2 Einlesen von Daten im freien Format

Für das Einlesen von Daten aus anderen Formaten sind spezielle Programme erforderlich, d.h. Stata kann andere Systemfiles nicht direkt importieren. Ein von Kohler/Kreuter empfohlenes Programm ist *Stat/Transfer*. Alternativ lassen sich Daten auch zuerst in ein *ASCII-Format* umwandeln und anschließend von Stata einlesen. Dies kann durch drei verschiedene Befehle geschehen: `infile`, `insheet` und `infix` (siehe das Hilfemenü für Details). Das grundsätzliche Kommando lautet (am Beispiel des ersten Befehls): `infile using Dateiname, Optionen`, d.h. nach `using` wird die ASCII-Datei angegeben, die eingelesen werden soll. Meistens sind weitere Optionen vonnöten, um den Datensatz korrekt einzulesen.

Die hier verwendeten Absolventen-Datei war zunächst im „sav-Format“ für SPSS gespeichert. Um sie für Stata nutzen zu können, wurden sie zunächst als ASCII-Code in eine „dat“-Datei exportiert, welche ausschnittsweise folgendermaßen aussieht:

RESPNUM\\$	V3	V11	V12	V13	BERZUF	ANOTE
9	2,1	1	1	8	2,7	4
10	2,2	0	1	7	3	0
11	2,3	0	1	10	3,3	6

Wie man erkennt, sind die Variablen durch Tabulatoren voneinander getrennt. Dies muss Stata durch die Option `tab` mitgeteilt werden. Zusätzlich befinden sich die Variablenamen in der ersten Zeile, was mit Hilfe der Option `names` berücksichtigt werden kann. Der vollständige Befehl zum Einlesen der Daten lautet demnach:

```
insheet using "absolventen.dat", tab names clear
```

Die Option `clear` bewirkt, dass die neuen Daten eventuell im Speicher vorhandene Daten ersetzt. Wenn man nun die eingelesenen Daten betrachtet, fällt folgendes auf:

```
describe
```

```
Contains data
```

```
obs:          584
vars:          24
size:         22,192 (97.9% of memory free)
```

```
-----
      storage  display      value
variable name  type   format      label      variable label
-----
respnum        int   %8.0g
v3              str3  %9s
v11            str1  %9s
v12            str1  %9s
v13            str1  %9s
-----
```

berzufr	str2	%9s	BERZUFRR
anote	str3	%9s	ANOTE
ersteink	byte	%8.0g	ERSTEINK
	...		

Betrachtet man die Spalte „storage type“, so fällt auf, dass einige Variablen als *alpha-numerische Strings* gespeichert wurden. Mit ihnen könnten so keine rechentechnischen Operationen durchgeführt werden. Um diese Variablen in ein *numerisches Format* zu überführen (byte, int, float etc.)⁶, muss der Storage-Typ verändert werden. Dies geschieht am einfachsten durch den Befehl

```
. destring, replace

respnum already numeric; no replace
v3 contains non-numeric characters; no replace
v11 has all characters numeric; replaced as byte
(1 missing value generated)
v12 has all characters numeric; replaced as byte
(1 missing value generated)
...
```

Der Stata-Output zeigt an, dass die meisten bisher als strings gespeicherten Variablen als numerische Variablen erkannt werden und dementsprechend in dieses Format umgewandelt werden. Probleme bereitet nur die Variable *v3*, die die Schulnoten darstellt. Der Grund liegt darin, dass die Schulnoten mit Kommata angegeben werden, was Stata nicht als Zahl erkennt (der Trennungsoperator ist bei Stata ein Punkt). Um dies zu beheben, muss der Trennungsoperator geändert werden, was am einfachsten direkt in der ASCII-Datei vorgenommen werden kann.

5.1.3 Direktes Eingeben von Daten

Daten können auch durch den Befehl `edit` eingegeben werden. Dies öffnet den *Dateneditor* (siehe auch Abschnitt 2). Alternativ kann auch der Befehl `input` verwendet werden. Nach dem Einlesen von Daten können diese mit dem Befehl `save Dateiname` (im Homeverzeichnis) gespeichert werden. Nun kann die Aufbereitung der Daten beginnen.

5.2 Aufbereitung von Daten

Dieser Abschnitt soll einen Überblick über die Möglichkeiten der Datenaufbereitung liefern. Dabei wird vor allem auf die grundsätzlichen Befehle des Erstellens und Veränderns von Variablen eingegangen, die zusammen mit einigen weiteren Optionen für praktisch alle möglichen Rekodierungsarbeiten ausreichen. Abschließend wird auf weitere Befehle und Missings kurz eingegangen.

⁶Zu einem Überblick der Formattypen vgl. Kohler/Kreuter (2001, S.97f.)

5.2.1 Erstellen und Verändern von Variablen

Neue Variablen können durch den Befehl `generate var=exp` erstellt, bereits vorhandene Variablen durch den Befehl `replace var=exp` verändert werden. Dabei steht *exp* für eine beliebige Funktion, und lässt sowohl algebraische (Addition etc.) wie auch relationale (wahr/falsch) Operatoren zu.

Beispiel: Erstellen eines additiven Indexes aus dem ersten und höchsten Einkommen:

```
generate dureink = ersteink + maxeink
(44 missing values generated)
```

Die neue Datei hat theoretisch einen Wertebereich bis 20:

```
. summarize dureink
```

Variable	Obs	Mean	Std. Dev.	Min	Max
dureink	540	7.683333	3.369857	2	20

Beispiel: Replikation der Variablen *einsatz*:

```
. generate einsatz2=0

. replace einsatz2 = (v11==1)+(v12==1)+(v13==1)+(kursbes==1)
(551 real changes made)

. corr einsatz2 einsatz
(obs=581)
```

	einsatz2	einsatz
einsatz2	1.0000	
einsatz	1.0000	1.0000

Der Befehl `corr` zeigt die Korrelation zwischen den beiden Variablen an, die erwartungsgemäß gleich 1 ist.

5.2.2 Die Optionen „by“, „_n“ und „_N“

Diese Befehlszusätze sind etwas gewöhnungsbedürftig, aber ungemein hilfreich. `_n` gibt die Position einer Beobachtung im Datensatz an, `_N` ist der höchste Wert von `_n` und `by` erlaubt die Unterteilung in Subgruppen (siehe Abschnitt 4), die dann *jeweils* durchgezählt werden. Um die `by`-Option zu verwenden, müssen die Daten zunächst mit dem Befehl `sort Variablenname` entsprechend sortiert werden.

Beispiel: Es soll eine Variable erstellt werden, die die höchste Zahl an BWLern allen BWL-Studenten zuteilt.

```
. sort bwl
. quietly by bwl: generate bwlnum = _N
. replace bwlnum = . if bwl==0
```

Die Option *quietly* kann verwendet werden, um den Stata-Output zu unterdrücken. Der letzte Befehl teilt allen Nicht-BWLern ein Missing zu (siehe 5.2.4).

5.2.3 Subskripte

Subskripte können durch eine eckige Klammer direkt nach der Variablen benannt werden.

Beispiel: Berechnung der (negativen) Median-Studiendauer.

```
. sort studdau
. display 0.5 * studdau[_N/2]+studdau[_N/2+1]
-16.5
```

5.2.4 Weitere Befehle und Missings

Weitere Rekodierungsbefehle sind **recode** und **egen**, auf die hier nicht näher eingegangen werden soll, da mit den Befehlen **generate** und **replace** an sich alle Variablentransformationen durchgeführt werden können.

Missings werden bei Stata durch ein **.** gekennzeichnet. Damit lassen sich beliebig Missings erstellen bzw. eliminieren.

Beispiel: Die bei *einsatz2* mit einer 0 gekennzeichneten Beobachtungen sollen als Missings deklariert werden.

```
. replace einatz2 = . if einatz2 == 0
```

Achtung: Missings werden von Stata mit dem Wert „*plus Unendlich*“ versehen, deshalb immer auf Missings achten!

Beispiel: Auflistung aller Personen, die länger als 50 Monate auf Arbeitssuche waren:

```
. list arbsuch if arbsuch>50
```

```
+-----+
| arbsuch |
|-----|
8. |      . |
47. |      . |
51. |      . |
67. |      . |
```

```

120. |      . |
      |-----|
150. |      . |
226. |      . |
242. |      . |
256. |      . |
277. |      . |
      |-----|
295. |      . |
336. |      . |
372. |     145 |
391. |      . |
409. |      . |
      |-----|
420. |      . |
504. |      . |
527. |      . |
565. |      . |
583. |      . |
      +-----+

```

Wie man erkennt, hat nur eine Person tatsächlich länger als 50 Monate nach einer Arbeit gesucht – alle anderen Personen haben keine Monatszahl angegeben. Diese Handhabung der Missings durch Stata ist nicht unproblematisch, da bspw. bei einer neuen Unterteilung einer Variablen die Missings automatisch der Kategorie mit den höchsten Werten zugeschlagen werden. Dies kann allerdings durch entsprechend vorsichtige Variablentransformation umgangen werden.⁷

5.2.5 Beschriftung von Variablen

Allgemein kann durch den Befehl `label` eine Variable beschriftet werden. *Variablenlabels* können durch den Befehl `label variable` erstellt werden:

```
. label variable fachint ''Fachinteresse als Grund für Studienwahl''
```

Das Label erscheint dann rechts im Variablen-Fenster.

Wertelabels können ebenfalls durch den Befehl `label` erstellt werden, allerdings muss dazu zunächst mit dem Befehl `label define` ein „Behälter“ oder Objekt der Werte definiert werden. Anschließend wird dieser Behälter durch den Befehl `label value` auf die Variable angewandt.⁸

Beispiel: Bezeichnung der Variablen *mann* durch „Mann“ bzw. „Frau“.

⁷In dem hier betrachteten Beispiel kann man die Missings relativ einfach ausschließen: `list arbsuch if arbsuch>50 & arbsuch =.` liefert nur die eine Person, die 145 Monate lang eine Arbeit gesucht hat.

⁸Auf die hier kurz genannte Möglichkeit, mit Objekten zu arbeiten, wird im Abschnitt 8.3 näher eingegangen.

```
. label define mf 0 "Frau" 1 "Mann"
. label value mann mf
. tabulate mann
```

Geschlecht	Freq.	Percent	Cum.
Frau	153	26.20	26.20
Mann	431	73.80	100.00
Total	584	100.00	

Damit wurden einige der wichtigsten Rekodierungsbefehle vorgestellt. Im anschließenden Abschnitt wird nun auf deskriptive Statistiken eingegangen.

6 Deskriptive Analysen

6.1 Darstellung des Datensatzes

Noch vor der Betrachtung statistischer Kennzahlen für die Variablen des Datensatzes, kann ein Blick auf den Aufbau des gesamten oder Teilen des Datensatzes geworfen werden. Eine erste Beschreibung des gesamten Datensatzes erfolgt über den Befehl `describe`.

```
.describe
```

```
Contains data from C:\absolventen.dta
```

```
obs:          584
vars:          24          6 Jul 2004 04:18
size:         23,360 (97.8% of memory free)
```

variable name	storage type	display format	value label	variable label
respnum	int	%8.0g		RESPNUM\$
v3	double	%10.0g		V3
v11	byte	%10.0g		V11

Um die Ausprägung aller oder bestimmter Variablen für alle oder bestimmte Fälle zu betrachten, kann entweder in gewohnter Weise der Data-Browser benutzt werden, oder aber – und hier ergeben sich weitere Funktionalitäten – der Befehl

```
list varlist
```

Wenn zuvor der Datensatz nach einem bestimmten Merkmal sortiert wurde, kann entsprechend auch eine sinnvolle Auswahl der Fälle vorgenommen werden. Beispiel:

```
. sort anote
. list maxeink berzufr anote in 1/5
```

```
+-----+
| maxeink  berzufr  anote |
+-----+
1. |      1      7    1.3 |
2. |      3      6    1.3 |
3. |      3      7    1.4 |
4. |      .      9    1.4 |
5. |      9      8    1.4 |
+-----+
```

6.2 Einfache univariate Maßzahlen

Abschnitt 4 hat bereits kurz in die Darstellung einfacher univariater Maßzahlen eingeführt. Der zentrale Befehl lautet

```
summarize varlist
```

und führt zur Ausgabe von Zahl der gültigen Fälle, arithmetischem Mittel, Standardabweichung und Minimum und Maximum:

Variable	Obs	Mean	Std. Dev.	Min	Max
maxeink	540	4.568519	2.124364	1	10
berzufr	582	7.628866	6.811936	0	99
anote	584	2.704795	.4881206	1.3	3.8

Weitere Statistiken sind über die Option `details` verfügbar. Hier werden Quantile, Varianz, Schiefe und Kurtosis angezeigt. Beispiel:

```
summarize maxeink, detail
```

```

                                MAXEINK
-----
Percentiles      Smallest
1%                1          1
5%                2          1
10%               3          1      Obs          540
25%               3          1      Sum of Wgt.  540

50%               4          Mean          4.568519
```

		Largest	Std. Dev.	2.124364
75%	5	10		
90%	8	10	Variance	4.512922
95%	10	10	Skewness	1.028748
99%	10	10	Kurtosis	3.558173

Der vorzügliche Einsatz von `if` und `by` für das `summary` Kommando wurde bereits in Abschnitt 4 beschrieben.

6.3 Tabellen

Der Befehl `tabulate` ist die Grundlage für die Erstellung von Häufigkeitstabellen. Mit nur einer Variable, also

```
tabulate variable
```

führt er zu einer einfachen eindimensionalen Häufigkeitstabelle.

MAXEINK	Freq.	Percent	Cum.
-----+-----			
1	15	2.78	2.78
2	33	6.11	8.89
3	146	27.04	35.93
4	139	25.74	61.67
5	74	13.70	75.37
6	44	8.15	83.52
7	24	4.44	87.96
8	24	4.44	92.41
9	12	2.22	94.63
10	29	5.37	100.00
-----+-----			
Total	540	100.00	

Die Berücksichtigung von Missing values erfolgt über die Option `missing`:

```
. tab maxeink, missing
```

MAXEINK	Freq.	Percent	Cum.
-----+-----			
1	15	2.57	2.57
2	33	5.65	8.22
3	146	25.00	33.22
4	139	23.80	57.02
5	74	12.67	69.69
6	44	7.53	77.23

7	24	4.11	81.34
8	24	4.11	85.45
9	12	2.05	87.50
10	29	4.97	92.47
.	44	7.53	100.00

Total	584	100.00
-------	-----	--------

Der Einbezug von zwei Variablen, also

`tabulate variable1 variable2`

führt zu einer zweidimensionalen Kreuztabelle, wobei *variable1* die Zeilen bildet und *variable2* die Spalten. Außerdem können hier durch die Optionen *row* und *column* leicht Zeilen- und Spaltenprozentage mit ausgegeben werden. Beispiel: `tabulate einsatz fachint, row column`

EINSATZ	FACHINT						Total
	0	1	2	3	4	5	
0	0	1	0	14	9	8	32
	0.00	3.13	0.00	43.75	28.13	25.00	100.00
	0.00	33.33	0.00	10.07	3.38	5.48	5.51
1	0	2	6	26	65	19	118
	0.00	1.69	5.08	22.03	55.08	16.10	100.00
	0.00	66.67	23.08	18.71	24.44	13.01	20.31
2	1	0	11	52	97	54	215
	0.47	0.00	5.12	24.19	45.12	25.12	100.00
	100.00	0.00	42.31	37.41	36.47	36.99	37.01
3	0	0	9	40	77	57	183
	0.00	0.00	4.92	21.86	42.08	31.15	100.00
	0.00	0.00	34.62	28.78	28.95	39.04	31.50
4	0	0	0	7	18	8	33
	0.00	0.00	0.00	21.21	54.55	24.24	100.00
	0.00	0.00	0.00	5.04	6.77	5.48	5.68
Total	1	3	26	139	266	146	581
	0.17	0.52	4.48	23.92	45.78	25.13	100.00
	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Mit der zusätzlichen Option `nofreq` könnte die Tabelle außerdem von den absoluten Häufigkeiten befreit werden, so dass nur Zeilen- und/oder Spaltenprozentage übrig blieben.

Ein besonderer Typus von Kreuztabellen ist durch die Kombination des `tabulate-` mit dem `summarize-` Befehl produzierbar: Mittelwerte und Standardabweichungen der einen Variable sollen gegen die Ausprägung der anderen Variable abgetragen werden. Am Beispiel:

```
tabulate mann, summarize(maxeink)b
```

Summary of MAXEINK			
MANN	Mean	Std. Dev.	Freq.
0	3.993007	1.8134727	143
1	4.7758186	2.1910331	397
Total	4.5685185	2.1243638	540

Dieser Typus ist auch übertragbar auf drei interessierende Merkmale. Der Übersichtlichkeit halber verzichten wird hier durch die Optionen *nostandard* und *nofreq* auf die Darstellung von Standardfehlern und Häufigkeit:

```
tabulate fachint mann, summarize(maxeink) nostandard nofreq
```

Means of MAXEINK			
FACHINT	MANN		Total
	0	1	
0	.	4	4
1	.	3	3
2	3.4	4	3.7391304
3	4.0652174	4.6117647	4.4198473
4	3.8032787	4.7150259	4.496063
5	4.5384615	5.1442308	5.0230769
Total	3.993007	4.7758186	4.5685185

Außerdem bietet Stata auch die Option, mehrgliedrige Tabellen selbst zu erstellen (d.h. Tabellen mit bis zu vier Dimensionen, also zwei Spaltenvariablen und zwei Zeilenvariablen). Mit dem etwas komplizierteren `table` Befehl werden die Kategorievariablen selbst ausgewählt und sodann der Inhalte der Tabelle festgelegt. Hier stehen erfreulicherweise alle denkbaren statistischen Maßzahlen zur Verfügung, so dass eine ganze Reihe an verschiedenen Tabellentypen über diesen Befehl generiert werden kann (zum genauen Vorgehen, siehe Kohler/Kreuter, 2001: S.155-157).

Schließlich kann Stata auch Variablen mit so vielen Ausprägungen, dass sie in kreuztabellarischer Form nicht sinnvoll darstellbar wären, gut handhaben. Die Gruppierung der Variablen entlang ihrer Quantile durch die Befehle `pctile` und `xtile` oder die Einteilung in gleichgroße Klassengruppen durch den Befehl `autocode()` sind relativ unkompliziert (genaues Vorgehen auch hier siehe Kohler/Kreuter, 2001: 148-151).

6.4 Korrelationen

Korrelationskoeffizienten zur Darstellung bivariater Zusammenhänge können von STATA ebenfalls recht intuitiv berechnet und gut dargestellt werden. Der Befehl

```
correlate varlist
```

führt zu einer Korrelationsmatrix, die immerhin schon so weit aufbereitet ist, dass die (redundante) Hälfte oberhalb der Hauptdiagonalen der Korrelationsmatrix gelöscht wurde:

```
(obs=540)

          |  maxeink  berzufr   anote
-----+-----
maxeink |   1.0000
berzufr |  -0.0053   1.0000
anote   |   0.0215   0.0213   1.0000
```

Außerdem stehen folgende weitere Optionen zur Verfügung: `correlate varlist, covariance` gibt statt der Korrelationsmatrix die Kovarianzmatrix aus `correlate varlist, _coef` berechnet die Korrelation zwischen den Koeffizienten des letzten geschätzten Modells.

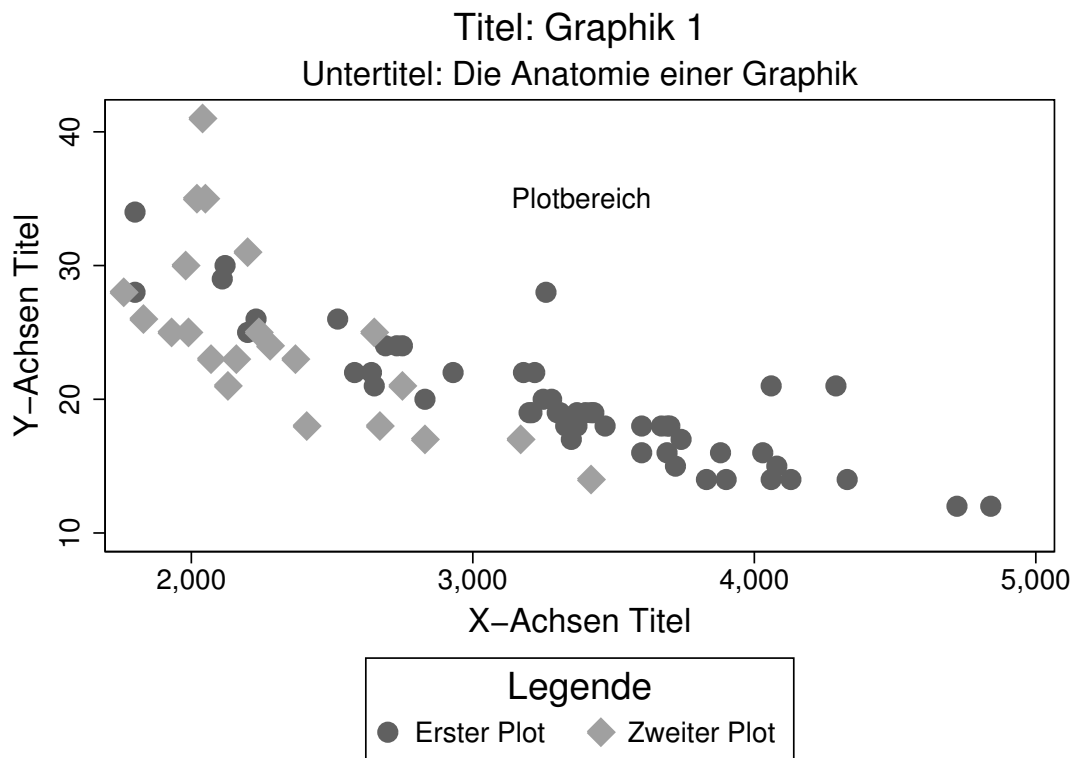
7 Graphiken

Stata stellt vielfältige Möglichkeiten bereit, um Graphiken zu erstellen. Es braucht allerdings eine gewisse Einarbeitung, um mit der Vielfalt zurecht zu kommen. Abbildung 1 veranschaulicht die wesentlichen Bereiche einer Statagraphik.

Listing 1 enthält die entsprechenden Befehle, um Abbildung 1 zu erhalten (Juul 2004).

Listing 1: Ein Beispiel

```
1 sysuse auto // open auto.dta accompanying Stata
2 set scheme s1manual
3 twoway (scatter mpg weight if foreign==0) ///
4   (scatter mpg weight if foreign==1) ///
5   , ///
6   title("Titel: Graphik 1") ///
7   subtitle("Untertitel: Die Anatomie einer Graphik") ///
8   ytitle("Y-Achsen Titel") xtitle("X-Achsen Titel") ///
9   note("Das ist der äußere Bereich der Graphik") ///
10  legend(title("Legende")
11         lab(1 "Erster Plot")
12         lab(2 "Zweiter Plot"))
13  text(35 3400 "Plotbereich ")
```



Das ist der äußere Bereich der Graphik

Abbildung 1: Elemente einer Graphik in Stata

Der grundsätzliche Aufbau eines Graphikbefehls ist:

```
graph-command (plot-command, options), options
```

Im Prinzip wird jeder Graphikbefehl mit dem Schlüsselwort `graph` eingeleitet. Danach gilt es zu entscheiden, ob es sich um eine Graphik mit zwei numerischen Achsen handelt, wie etwa ein Streudiagramm, oder, ob die Graphik vom Typ `bar`, `dot`, `box` oder `pie` ist. Schließlich gibt es noch (xxx den Befehl xxx) `matrix`, mit dem sich ganze Matrizen von Streudiagrammen erstellen lassen.

Daneben gibt es aber noch eine ganze Klasse weiterer Graphiken, die nicht nach diesem Schema konstruiert werden. Eine Liste der verfügbaren Typen erhält man über die Hilfsfunktion.

7.1 Histogramm

Mit `histogram varlist` (die Kurzform von `graph twoway (histogram varlist)`) lässt sich ein Histogramm erstellen. Neben der Spezifizierung der üblichen Parametern wie `if` oder `in` lassen sich nach dem Komma weitere Befehlszusätze übergeben, die danach unterschieden werden, ob das Merkmal als diskret oder kontinuierlich erachtet wird. Schließlich gibt es noch eine Liste von Optionen, die unabhängig von der Merkmalskalierung gesetzt werden können.

Der Befehl `graph twoway (histogram anote, percent)` erzeugt die nachfolgende Graphik:

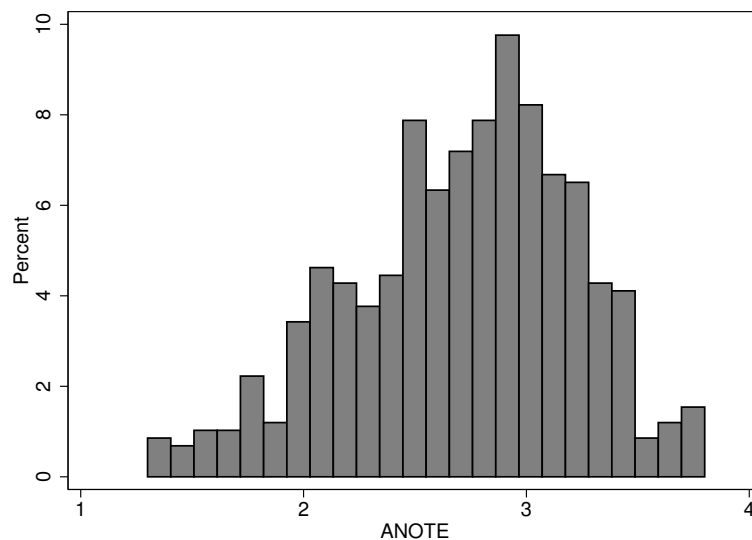


Abbildung 2: Ein einfaches Histogramm der Abschlussnote

7.2 Box-and-Whisker-Plot

Auch Boxplots lassen sich leicht konstruieren. An dieser Stelle soll noch auf die Option `by` hingewiesen werden, mit der sich leicht konditionale Graphiken konstruieren lassen; hier mit `by(mann)` zwei Boxplots der Abschlussnote getrennt für Frauen und Männer. Der vollständige Befehl lautet: `graph box anote, ytitle(Abschlussnote) by(mann)`.

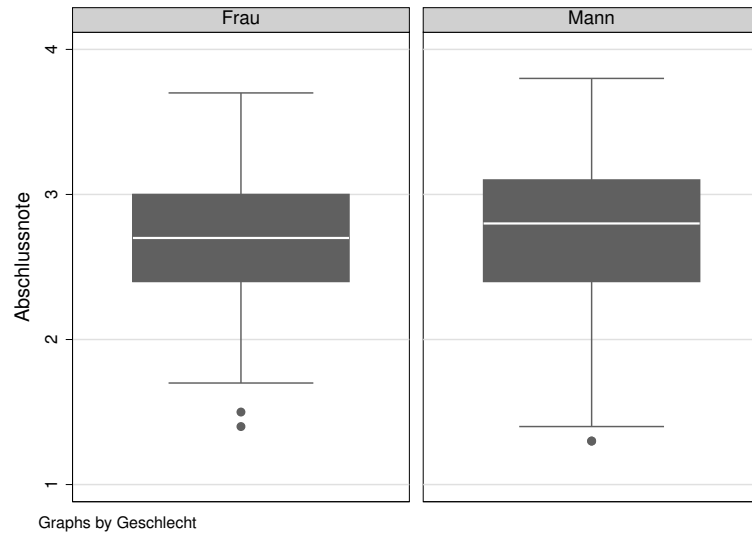


Abbildung 3: Boxplot der Abschlussnote nach Geschlecht

7.3 Scatterplot

Die nachfolgende Abbildung 4 erfordert geringfügig mehr Aufwand wie Listing 2 zeigt:

Listing 2: Abschlussnote nach Schulnote und Auslandsstudium

```
1 replace v3 =. if v3>6;  
2 bysort v11:reg anote v3;  
3 predict yhat;  
4 graph twoway (scatter yhat anote v3, connect(1) by(v11));  
5 graph export abb/scatter.eps, as(eps) replace;
```

In der ersten Zeile werden Werte größer als 6 als missing values deklariert. Danach wird eine bivariate lineare Regression getrennt für Studenten mit und ohne Auslandserfahrung durchgeführt und mit `predict yhat` die geschätzten Werte in der Variablen *yhat* abgespeichert.

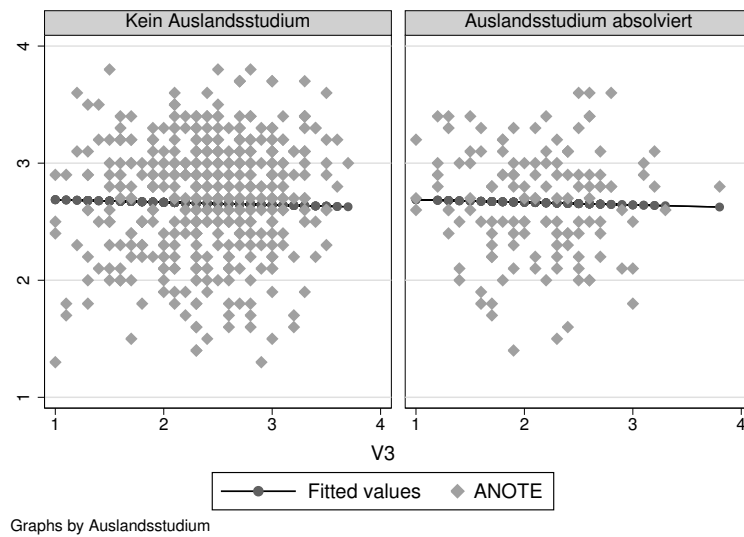


Abbildung 4: Abschlussnote nach Schulnote und Auslandsstudium

8 Multivariate Verfahren: OLS-Regression

Dieser Abschnitt soll einen Einblick in multivariate Analysen mit Stata am Beispiel einer OLS-Regression geben. Zunächst wird der Output der OLS-Regression anhand des Absolventen-Datensatzes erläutert. Anschließend werden einige weiterführende Befehle vorgestellt und ein Einblick in die Regressions-Diagnostik mit Stata gegeben.

8.1 Stata-Ausgabe bei einer OLS-Regression

Der grundsätzliche Befehl für eine OLS-Regression lautet: `regress abhängige Variable unabhängige Variablen, Optionen`. Dabei kann der Befehl mit `reg` abgekürzt werden. Als ein einfaches Beispiel lässt sich untersuchen, welchen Einfluss unabhängige Variablen wie Studienengagement, Schul- und Examensnote oder Geschlecht auf die abhängige Variable Ersteinkommen haben. Es ergibt sich folgender Output:

```
. reg ersteink anote fachint studdau einsatz bwl arbsuch mann v3
```

Source	SS	df	MS	
Model	46.0193356	8	5.75241694	Number of obs = 559
Residual	1525.91626	550	2.77439321	F(8, 550) = 2.07
Total	1571.9356	558	2.81708889	Prob > F = 0.0366
				R-squared = 0.0293
				Adj R-squared = 0.0152
				Root MSE = 1.6657

ersteink	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
anote	.0092893	.145837	0.06	0.949	-.2771764	.295755
fachint	.0377579	.0850054	0.44	0.657	-.1292171	.2047329
studdau	.0212631	.0242938	0.88	0.382	-.026457	.0689831
einsatz	-.0047703	.0757077	-0.06	0.950	-.153482	.1439414
bwl	.2917585	.1544836	1.89	0.059	-.0116916	.5952087
arbsuch	-.0115739	.0098192	-1.18	0.239	-.0308616	.0077138
mann	.4055152	.1631252	2.49	0.013	.0850905	.7259399
v3	-.1962839	.1314263	-1.49	0.136	-.4544428	.061875
_cons	3.057537	.6986942	4.38	0.000	1.685102	4.429973

Der Stata-Ausdruck lässt sich in drei Blöcke unterteilen (im Uhrzeigersinn von links oben):

1. *Der Anova-Block* zeigt die Zerlegung der gesamten Quadratsumme ("SS" steht für "Sum of Squares") in die durch das Modell erklärte Quadratsumme und die Quadratsumme der Residuen. Die Abkürzung "df" steht für "Degree of Freedoms", gibt also die Freiheitsgrade an. Die rechte Spalte zeigt die mittlere Quadratsumme bezogen auf die jeweiligen Freiheitsgrade ($MS = \frac{SS}{df}$).
2. *Der Modellfit-Block* gibt neben der Anzahl an untersuchten Einheiten die F-Statistik und den (korrigierten) R^2 -Wert an (der in dem hier betrachteten Fall äußerst mager ausfällt). "Root MSE" entspricht der Wurzel der durchschnittlichen Residuen des Modelles.⁹
3. *Der Koeffizientenblock* listet die unabhängigen Variablen zusammen mit der Regressionskonstanten (`_const`, dem ermittelten Koeffizientenwert, Standardfehler, t- und P-Wert sowie dem Konfidenzintervalle auf. Es lässt sich bspw. ablesen, dass die Examensnote (`anote`) keinen statistisch signifikanten Einfluss auf das Ersteinkommen hat (genauso wie die meisten anderen Variablen – außer den Dummy-Variablen `bwl` und `mann`...). Bei den Koeffizienten handelt es sich um die unstandardisierten Werte. Die (häufig als beta-Koeffizienten bezeichneten) standardisierten Koeffizienten erhält man mit der Option `beta`. Die abhängige Variable `ersteink` befindet sich in der Tabelle oberhalb der unabhängigen Variablen.

Angesichts der desaströsen Ergebnisse dieser Regression stellt sich die Frage, ob Variablen fehlen und ob die Annahmen einer OLS-Regression überhaupt erfüllt sind. Im folgenden Abschnitt werden deswegen zunächst weitere Stata-Kommandos rund um die Regression vorgestellt, anschließend erfolgt eine kurze Regressions-Diagnostik.

⁹Die Formel lautet: $RootMS = \sqrt{\frac{RSS}{n-k}}$, mit RSS für die Quadratsumme der Residuen, n für die Anzahl der Beobachtungen und k für die Anzahl unabhängiger Variablen.

8.2 Erweiterungen zur OLS-Regression

Kategorial unabhängige Variablen mit mehreren Ausprägungen können ohne Probleme in die Regression einbezogen werden. Für den verwendeten Datensatz könnte man bspw. die Variablen, die den Kontakt mit ehemaligen Kommilitonen misst, in die Untersuchung integrieren; sofern man aus Versehen alle gebildeten Dummies verwendet und damit Multikollinearität erzeugt, entfernt Stata automatisch eine der Dummy-Variablen (welche somit als Referenzkategorie verwendet wird).

Zum Erzeugen von Dummy-Variablen aus kategorialen Variablen kann der Befehl `tab` *Variablenname*, `gen(Neue Variable)` benutzt werden: Aus der mit *Variablenname* bezeichneten kategorialen Variable werden so viele mit *Neue Variable K* bezeichneten Dummy-Variablen gebildet, wie die Variable Kategorien besitzt.

Beispiel: Bildung von Dummy-Variablen aus der kategorialen Variablen *ersteink*.¹⁰

```
. quietly tab ersteink, gen(eink)
. describe eink*
```

variable name	storage type	display format	value label	variable label
eink1	byte	%8.0g	ersteink==	0.0000
eink2	byte	%8.0g	ersteink==	1.0000
eink3	byte	%8.0g	ersteink==	2.0000
eink4	byte	%8.0g	ersteink==	3.0000
eink5	byte	%8.0g	ersteink==	4.0000
eink6	byte	%8.0g	ersteink==	5.0000
eink7	byte	%8.0g	ersteink==	6.0000
eink8	byte	%8.0g	ersteink==	7.0000
eink9	byte	%8.0g	ersteink==	8.0000
eink10	byte	%8.0g	ersteink==	9.0000
eink11	byte	%8.0g	ersteink==	10.0000

Es wurden insgesamt 11 Dummy-Variablen gebildet, die mit *eink1* bis *eink11* bezeichnet werden.

Interaktionseffekte treten auf, wenn die Effekte einer Variablen auf die abhängige Variable mit den Werten einer dritten Variablen variieren. In Stata können Interaktionseffekte durch Multiplikation der Variablen gebildet und dann in das Modell einbezogen werden.

Beispiel: Zusammenhang zwischen Examensnote und Ersteinkommen in Abhängigkeit vom Geschlecht.

Prinzipiell ließe sich ein solcher Interaktionseffekt direkt in das obige Modell einbauen. Zur Vereinfachung wird nur der Effekt der Examensnote und des Geschlechts mit Interaktionseffekt auf das Ersteinkommen geschätzt.

¹⁰Der Befehl `quietly` unterdrückt den Stata-Output der Regression.

```
. reg ersteink anote mann iakt
```

Source	SS	df	MS	Number of obs =	584
Model	22.1943584	3	7.39811947	F(3, 580) =	2.48
Residual	1727.41523	580	2.97830212	Prob > F =	0.0599
Total	1749.60959	583	3.00104561	R-squared =	0.0127
				Adj R-squared =	0.0076
				Root MSE =	1.7258

ersteink	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
anote	.0278934	.3116099	0.09	0.929	-.5841279	.6399148
mann	.6643664	.9589777	0.69	0.489	-1.219126	2.547858
iakt	-.0831485	.3531369	-0.24	0.814	-.7767315	.6104345
_cons	2.481246	.841788	2.95	0.003	.8279219	4.13457

Die Regressionsschätzung ergibt, dass für Männer erwartungsgemäß das Ersteinkommen mit der Höhe der Examensnote sinkt, während es bei den Frauen tendenziell steigt; allerdings sind die Koeffizientenwerte allesamt nicht signifikant.¹¹

Schließlich können auch *nichtlineare Zusammenhänge* durch entsprechende Variablen-generierung und deren Einbau in das Modell leicht geschätzt werden.

Beispiel: Es wird angenommen, dass die Examensnote nichtlinear auf das Ersteinkommen wirkt, was durch einen quadratischen Term berücksichtigt werden soll.

```
. gen anote2=anote*anote
. label var anote2 "Examensnote quadriert"
. reg ersteink anote anote2 fachint studdau einsatz bwl arbsuch mann v3
```

Source	SS	df	MS	Number of obs =	559
Model	61.0244487	9	6.7804943	F(9, 549) =	2.46
Residual	1510.91115	549	2.75211503	Prob > F =	0.0093
Total	1571.9356	558	2.81708889	R-squared =	0.0388
				Adj R-squared =	0.0231
				Root MSE =	1.6589

ersteink	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
----------	-------	-----------	---	------	----------------------	--

¹¹Der Interaktionseffekt zwischen Examensnote und Geschlecht hat auch in dem ersten geschätzten Modell bei Abschnitt 8.1 keinen signifikanten Einfluss, weswegen die Ergebnisse hier nicht aufgeführt werden sollen und der Interaktionseffekt im Folgenden nicht weiter berücksichtigt wird.

anote		2.826166	1.215085	2.33	0.020	.4393805	5.212951
anote2		-.5394593	.2310321	-2.33	0.020	-.9932744	-.0856442
			...				

Der Ausschnitt aus dem Stata-Output zeigt, dass durch das Einfügen des quadratischen Terms der Einfluss der Examensnote auf das Ersteinkommen signifikant geworden ist. Es wird ein zunächst positiver (kontraintuitiver) Zusammenhang geschätzt, der ab einer Examensnote von ca. 2,5 negativ wird.

Dieser Zusammenhang kann mit Hilfe der Regressionkoeffizienten auch graphisch veranschaulicht werden:

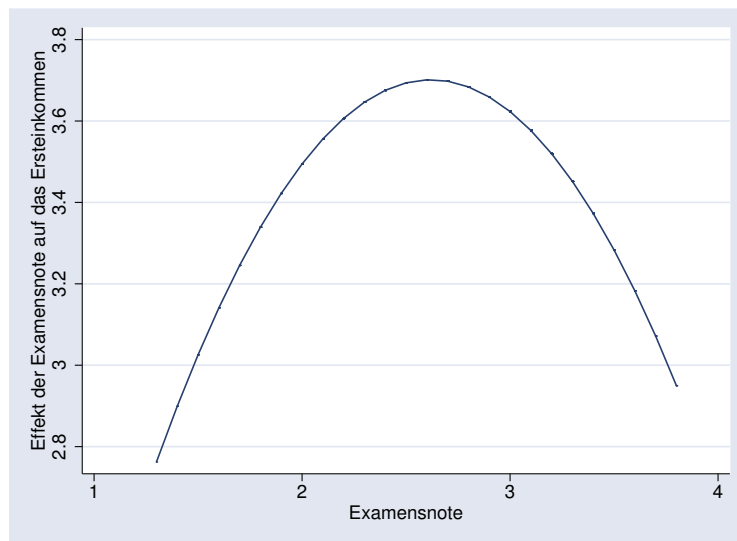


Abbildung 5: Der geschätzte Effekt der Examensnote auf das Ersteinkommen

Der dazugehörige Befehl lautet:

```
. gen anotehat=_b[anote]*anote+_b[anote2]*anote*anote
. scatter anotehat anote, c(1) s(i) sort
```

Dabei wird auf die „Behälter“ zurückgegriffen, die Stata nach einer Regression erzeugt; mehr davon im nächsten Abschnitt.

8.3 Regression-Diagnostik

Nach einer Regression können die der OLS-Regression zugrunde liegenden Annahmen überprüft werden. Dazu ist es hilfreich, zunächst auf die Stata-internen „Behälter“ kurz einzugehen.

Wie bereits erwähnt wurde, kennt Stata sogenannte „Behälter“ bzw. Objekte, in die Daten eingespeichert werden können. Bei Berechnungen jeglicher Art speichert Stata die

Resultate in interne Objekte. So werden bei der Regression u.a. die Koeffizientenwerte, Quadratsummen etc. abgespeichert. Ein Überblick über diese Behälter nach einem Schätzverfahren erhält man durch den Befehl `ereturn list`.¹² Es ergibt sich folgender Output:

scalars:

```
e(N) = 559
e(df_m) = 8
e(df_r) = 550
e(F) = 2.073396420824325
e(r2) = .0292755858318973
e(rmse) = 1.665650985883118
e(mss) = 46.01933555906635
e(rss) = 1525.916263725371
e(r2_a) = .0151559579894522
e(ll) = -1073.860781586727
e(ll_0) = -1082.165472148798
```

macros:

```
e(depvar) : "ersteink"
e(cmd) : "regress"
e(predict) : "regres_p"
e(model) : "ols"
```

matrices:

```
e(b) : 1 x 9
e(V) : 9 x 9
```

functions:

```
e(sample)
```

Die Bezeichnung `e()` steht für "estimates", d.h. es handelt sich hierbei um Objekte, die nach einem Schätzverfahren abgespeichert wurden. Bei `e(N)` hat Stata bspw. die Anzahl der Beobachtungen abgespeichert; `e(b)` ist eine Matrix bzw. Vektor mit den neun geschätzten Koeffizientenwerten.

Auf diese Behälter kann man bei Regressions-Diagnostiken zurückgreifen. Dies wird deutlich, wenn man als erstes die *Residuen* der Regression betrachten will. Prinzipiell lässt sich jedes Residuum als Differenz zwischen dem tatsächlichen und dem durch das Modell geschätzten Wert bestimmen. Bei Stata wird dies durch den Befehl `predict varname, resid` automatisch berechnet. Genauso lassen sich durch `predict varname` die geschätzten Werte für die abhängige Variable in einer Variablen speichern. Mit diesen beiden neuen Variablen kann nun bspw. ein *Residual-versus-fitted-Plot* erzeugt wer-

¹²Bei früheren Stata-Versionen lautet der Befehl: `estimates list`.

den, bei dem die Residuen für die jeweiligen geschätzten Werte dargestellt werden. Dies geschieht durch die folgenden Befehle:

```
. reg ersteink anote anote2 fachint studdau einsatz bwl arbsuch mann v3  
. predict yhat  
. predict resid, resid  
. scatter resid yhat, s(o)
```

Alternativ kann die Graphik auch direkt durch den Befehl `rvfplot` nach der Regression erzeugt werden.

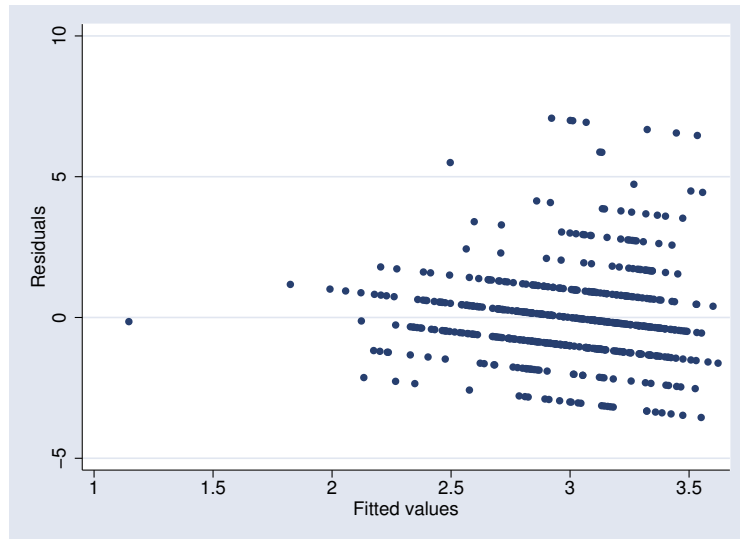


Abbildung 6: Residuals-versus-fitted Plot

Die Graphik macht deutlich, dass die Residuen für höhere geschätzte Werte eher positiv sind. Demnach liegt der Schluss nahe, dass eine Verletzung der Annahme $E(e|\mathbf{X}) = 0$ zugrunde liegt, d.h. der Erwartungswert der Fehlerterme (in Abhängigkeit von den Werten der unabhängigen Variablen \mathbf{X}) ist nicht Null. In einem solchen Fall ist keine effiziente Schätzung der Koeffizienten möglich.¹³

Es gibt eine Reihe graphischer Analysemöglichkeiten, um die Annahmen der OLS-Regression zu überprüfen und Ausreißer zu identifizieren. Auf sie soll hier nicht weiter eingegangen werden; bei Kohler/Kreuter (Kapitel 8.3) werden einige Verfahren ausführlich beschrieben. Stattdessen werden abschließend noch zwei Tests auf *Heteroskedastizität* vorgestellt.

Bei Heteroskedastizität ist die Annahme verletzt, dass die Varianz der Fehlerterme der OLS-Regression (in Abhängigkeit von den Werten der unabhängigen Variablen \mathbf{X}) konstant ist. Als Tests auf Heteroskedastizität bieten sich der White-Test sowie der Breusch-Pagan-Test an.

¹³Für eine ausführliche Darstellung der Annahmen einer OLS-Regression und der Konsequenzen bei Verletzung dieser siehe etwa Wooldridge (2003: Kapitel 3).

Für den *White-Test* genügt es, nach der Regression den Befehl `imtest, white` aufzurufen. Das Stata-Output sieht in dem hier betrachteten Beispiel wie folgt aus:

```
. quietly reg ersteink anote anote2 fachint studdau einsatz bwl arbsuch mann v3
. imtest, white
```

```
White's test for Ho: homoskedasticity
    against Ha: unrestricted heteroskedasticity
```

```
chi2(51)      =    49.17
Prob > chi2   =    0.5464
```

Cameron & Trivedi's decomposition of IM-test

Source	chi2	df	p
Heteroskedasticity	49.17	51	0.5464
Skewness	21.83	9	0.0094
Kurtosis	12.98	1	0.0003
Total	83.99	61	0.0271

Wie sich erkennen lässt, wird die Alternativhypothese zurückgewiesen, d.h. dieser Test zeigt keine Heteroskedastizität an.

Der Befehl für den *Breusch-Pagan-Test* lautet: `hettest`, und führt zu folgendem Stata-Output:

```
Breusch-Pagan / Cook-Weisberg test for heteroskedasticity
```

```
Ho: Constant variance
Variables: fitted values of ersteink
```

```
chi2(1)      =    31.89
Prob > chi2   =    0.0000
```

Der Breusch-Pagan-Test weist die H_0 zurück, wonach Homoskedastizität bestehe und deutet somit auf das Vorliegen von Heteroskedastizität hin. Die unterschiedlichen Testresultate sind auf die verschiedenen Testschätzungen zurückzuführen.¹⁴

Abschließend kann noch angemerkt werden, dass die hier beschriebenen Verfahren auf andere multivariate Modelle übertragbar sind. Das Logit-Modell wird bspw. mit dem Befehl `logit abhängige Variable unabhängige Variable, Optionen` aufgerufen, das Probit-Modell mit dem Befehl `probit abhängige Variable unabhängige Variable, Optionen`.

¹⁴Für eine Darstellung der beiden Tests vgl. Wooldrige (2003:S. 260ff.). Es ist dabei zu beachten, dass beim von Stata durchgeführten Breusch-Pagan-Test keine linear-additive Funktion für die quadrierten Residuen angenommen wird, sondern eine log-lineare Funktion.

9 Die Variablen der Absolventenstudie

Var.name	Beschreibung	Codierung
respsnum\$	ID	
fachint	Fachinteresse als Grund für Studienwahl	1 (nicht wichtig) - 5 (wichtig)
einsatz	Einsatz/Engagement in der Studienzeit	0-4 (generiert aus v11, v12, v13 und kursbesuch durch Addition)
anote	Examensnote	1,00 - 4,00
studdau	Studiendauer (*-1)	Achtung: Umgedreht (*-1), damit Interpretation einfacher (mehr Semester = schlechter)
ersteink	Erstes Einkommen	1-10 (Einkommensgruppen)
maxeink	Höchstes Einkommen	1-10 (Einkommensgruppen)
berzuf	Zufriedenheit mit dem Beruf	0-10
v11	Auslandsstudium	0/1
v12	Praktika	0/1
v13	Zusätzliche Kurse	0/1
kursbes	Pflichtveranstaltungen regelmäßig besucht?	0/1
v3	Schulnote	1,0 - 4,0
aalter	Alter des Befragten	in Jahren
mann	Geschlecht	Dummy
bwl	BWL-Student	Dummy
arbsuch	Dauer der Arbeitssuche nach dem Examen	in Monaten
varbeit*	Berufliche Stellung des Vaters: Arbeiter oder Nicht-Berufstätig	Dummy
vangest*	Berufliche Stellung des Vaters: Angestellter	Dummy
vbeamt*	Berufliche Stellung des Vaters: Beamter	Dummy
vselbst*	Berufliche Stellung des Vaters: Selbständiger	Dummy
konkein**	Kein Kontakt zu ehemaligen Kommilitonen	Dummy
konpriv**	Privater Kontakt zu ehemaligen Kommilitonen	Dummy
konber**	Beruflicher Kontakt zu ehemaligen Kommilitonen	Dummy
konmix**	Privater und beruflicher Kontakt zu ehemaligen Kommilitonen	Dummy

* / ** Komplementäre Dummies, eine davon muss ‚gedroppt‘ werden

10 Literatur

Wooldridge, Jeffrey, U. (2002): *Introductory Economics. A Modern Approach*. Thomson South-western