



# Übung zur Empirischen Wirtschaftsforschung

## II. Datenquellen und Datenaufbereitung

### 1. Vereinfachte Kategorisierung von Datenarten

**Zeitreihen** sind Daten, die über die Zeit wiederholt erhoben werden, i.d.R. wird hierbei eine regelmäßige Frequenz benutzt. Ferner werden die Daten nach Möglichkeit bei denselben Merkmalsträgern und mit gleichbleibender Methodik und Definition erhoben. Beispiele: gesamtwirtschaftliche Daten wie BIP, Zinssätze etc.

**Querschnittsdaten** beziehen sich auf zeitlich ungeordnete Beobachtungen für unterschiedliche Merkmalsträger, die häufig zu einem einheitlichen Zeitpunkt erhoben werden. Damit Querschnittsdaten einen empirischen Gehalt haben, müssen sie stärker disaggregiert werden. So hat das BIP eines Landes für eine Periode keinen Querschnittscharakter, wohl aber ein Datensatz mit Informationen über die Wertschöpfung in einzelnen Sektoren einer Volkswirtschaft. Beispiele: bei einzelnen Wirtschaftssubjekten erhobene Individualdaten.

**Paneldaten** sind eine Kombination aus Zeitreihen und Querschnittsdaten. Sie werden wie die Zeitreihendaten mit einer regelmäßigen Frequenz erhoben, möglichst bei denselben Merkmalsträgern. Wie die Querschnittsdaten werden sie für unterschiedliche Merkmalsträger zum selben Zeitpunkt ermittelt. Bei Paneldaten liegen somit für eine Vielzahl von Merkmalsträgern mehrere zeitlich geordnete Beobachtungen vor. Beispiele: Ifo-Konjunkturtest, die VGR der Länder.

## 2. Datenquellen

Ökonomisch relevante empirische/statistische Informationen können mittlerweile zu einem erheblichen Umfang aus dem Internet bezogen werden:

Die Volkswirtschaftlichen Gesamtrechnungen stellen das umfassendste statistische Instrumentarium der Wirtschaftsbeobachtung dar. Interessant zum Nachschlagen ist beispielsweise die vierteljährliche Volkswirtschaftliche Gesamtrechnung für die Bundesrepublik Deutschland bei der Deutschen Bundesbank <http://www.bundesbank.de>. Hier finden sich unter anderem auch monetäre Daten der Bundesrepublik Deutschland, die aktuellen Monatsberichte (als Download) und die Pressemitteilungen der Deutschen Bundesbank. Des Weiteren gibt es zahlreiche monetäre Statistiken und Links zu den anderen Europäischen Zentralbanken.

Um ein solches System auch für die einzelnen Bundesländer zu erstellen, wurde 1954 der Arbeitskreis Volkswirtschaftliche Gesamtrechnungen der Länder (VGR d L) gegründet. Das Datenangebot des Arbeitskreises deckt weitgehend die Entstehungs-, Verteilungs- und Verwendungsrechnung auf Bundesländerebene ab (<http://www.statistik-bw.de/VolkswPreise/Landesdaten/>).

Das ifo Institut für Wirtschaftsforschung in München verdankt seinen Bekanntheitsgrad unter anderem den monatlichen unternehmensbezogenen Umfragen über die deutsche Industrie. Besonders der Konjunkturtest mit der Ermittlung des Geschäftsklimas erlangt monatlich weitreichendes Interesse. Diese Ergebnisse werden regelmäßig auch im Internet veröffentlicht (<http://www.cesifo-group.de>).

Daten für weitere Industrieländer findet man unter anderem bei der OECD <http://www.oecd.org> und beim Internationalen Währungsfond <http://www.imf.org>.

Im Rahmen der Europäischen Vereinigung, wurde ein Europäisches Statistisches Amt etabliert. Unter <http://epp.eurostat.ec.europa.eu/> bekommt man für die europäischen Länder vergleichbare Daten. Diese Daten sind teilweise bis auf Landkreisebene aufgliedert. Des Weiteren besteht hier ein kostenloser Zugang zu einer umfangreichen Literaturdatenbank.

Das Penn-World Table stellt 24 Basisdatenreihen für 188 Länder der Welt zur Verfügung (<http://pwt.econ.upenn.edu/>). Damit sind insbesondere auch Vergleiche mit Ländern ohne eigene Statistische Ämter möglich.

Die Fed St. Louis bietet mit ihrer Datenbank *FRED*<sup>®</sup> (*Federal Reserve Economic Data*) <http://research.stlouisfed.org/fred2/>, für die USA ein extrem umfangreiches Datenangebot an. Hier stehen über 3.000 ökonomische Zeitreihen zur Verfügung.

## 3. Daten in EViews einlesen

### 3.1 Daten im Fremdformat

Mit dem Programm EViews können Daten im Fremdformat eingelesen werden. Dies ist im einzelnen möglich bei:

- Excel (.xls)
- Text-Ascii (\*.\*)

#### 3.1.1 Excel-Tabelle einlesen

In den meisten Fällen liegen Daten in Form einer Excel-Tabelle vor. Um in EViews Daten aus einer Excel-Tabelle einzulesen, stehen dem Benutzer verschiedene Vorgehensweisen zur Auswahl:

Als erste Möglichkeit können die Daten per Mausklick durch die Menüleisten eingelesen werden. Diese Variante stellt sogleich die benutzerfreundlichste Form des Datenimports nach EViews dar. Durch Klicks mit der linken Maustaste auf *File*, *Open* und *Foreign Data as Workfile* öffnet sich ein Fenster, in welchem der Benutzer die Excel-Tabelle der einzulesenden Daten angeben kann. Ist der Benutzer mit allen von EViews vorgeschlagenen Grundeinstellungen einverstanden, so genügt zum Einlesen der Daten ein Klick auf das Feld *Fertig stellen*. EViews erstellt nun automatisch ein Workfile mit dem Namen der Excel-Datei und öffnet alle Datenreihen als Gruppe.

Bei der zweiten Variante können die Daten über das Eingabefenster eingelesen werden.

- Dazu muss zuerst ein Workfile mit dem Befehl *wfcreate* erstellt werden, welches hinsichtlich der Periodizität und des Zeitraumes mit den Daten des Excel-Sheets übereinstimmt (Genauere Informationen: siehe Programmkommentar in *einlesen.prg*). Z.B. *wfcreate Zinsen m 1960:1 2008:12*.
- Danach wird der Befehl *read* eingegeben und im folgenden Fenster die einzulesende Excel-Datei ausgewählt und geöffnet. Nun benötigt EViews weitere Informationen:
- Unter *Data Order* kann der Benutzer zwischen zwei Optionen wählen. Wird die Voreinstellung *By Observation - series in columns* beibehalten, so liest EViews die Daten spaltenweise ein. Wird hingegen die Option *By Series - series in rows* gewählt, dann liest das Programm zeilenweise Daten ein.
- Unter *Import sample* ist der Einlesezeitraum automatisch voreingestellt. Dieser kann aber bei Bedarf geändert werden.
- Im Feld *Upper-left data cell* wird das Feld eingetragen, in dem in der Excel-Tabelle die erste einzulesende Zahl steht.
- In den Eingabebereich unter *Excel 5+ sheet name* wird der Name des Excel-Sheets eingetragen, in welchem die entsprechenden Daten hinterlegt sind. Z.B. *Tabelle3*.
- Im Feld *Names for series or Number if named in File* wird die Anzahl der einzulesenden Datenreihen angegeben, falls diese in der Excel-Datei bereits benannt worden sind. Ansonsten werden in diesem Eingabefeld die Benennungen der Datenreihen eingetragen. Das Trennsymbol ist hierbei das Leerzeichen. Z.B. *gdppc\_aus gdppc\_aut gdppc\_bel* usw.

Die dritte Möglichkeit zum Einlesen von Daten basiert auf der Programmierung einer Einleseprozedur. Ein Beispiel ist in der Datei `einlesen.prg` zu finden.

**Beachte!** Bei allen drei oben genannten Einlesemöglichkeiten muss stets die zugrunde liegende Excel-Datei geschlossen sein.

### 3.1.2 Text-Ascii einlesen

Das Einlesen einer Ascii-Datei funktioniert analog zum Einlesen einer Excel-Tabelle. Hier beispielhaft anhand der zweiten Einlesemöglichkeit erklärt:

- Im ersten Schritt wird ein Workfile erstellt. Z.B. `create q 1960:3 2004:2`.
- Danach wird der Befehl `read` eingegeben.
- Unter `Dateityp` wird Text-Ascii eingestellt und die entsprechende Datei geöffnet.
- Die Voreinstellung `Data order` wird gelassen, falls die einzulesenden Variablen in den Spalten stehen.
- Unter `Import sample` ist der Einlesezeitraum voreingestellt.
- Im Feld `Names for series or Number if named in File` wird die Anzahl der einzulesenden Datenreihen angegeben, falls diese in der Ascii-Datei bereits benannt worden sind. Ansonsten werden in diesem Eingabefeld die Benennungen der Datenreihen eingetragen.
- Im Punkt `Series headers` wird angegeben, ob in der Zeile über den Daten der Name der Datenreihe zu finden ist.
- Bei `Rectangular file layout` können Anfangsspalten bzw. -zeilen beim Einlesen gelöscht werden.

Im folgenden Beispiel wird im ersten Schritt ein Workfile mit Monatsdaten gebildet:  
`wfcreate test m 1990:1 1999:12`

Danach werden die Daten aus der Ascii-Textdatei eingelesen:

```
read(t=txt, skipcol=2) d:\test.txt 1
```

Aus der Text-Datei `test.txt` auf der Festplatte `d` wird 1 Reihe in EViews eingelesen, nachdem zwei Spalten gelöscht wurden.

Beispiel:

```
datum § PR1
01.90 § 11
02.90 § 17
...
...
...
10.99 § 0
11.99 § 1
12.99 § -14
```

## 3.2 EViews Help

EViews bietet dem Benutzer eine sehr ausführliche Hilfefunktion. Neben der *Help Reference*, in der die wichtigsten Befehle beschrieben sind, stehen mit *EViews 6.1 Update (pdf)*, *Users Guide (pdf)* und *Command Reference (pdf)* drei vollständige Handbücher als PDF innerhalb der Hilfe zur Verfügung.

## 3.3 Programmieren mit EViews

Zur Dokumentation und um häufig wiederkehrende Arbeitsschritte deutlich schneller zu erledigen, bietet EViews die Möglichkeit Programme zu erstellen.

### 3.3.1 Kommentare

Als Kommentarzeichen wird ' verwendet. Mit Einschränkungen kann hierfür aber auch @ verwendet werden. Dieses hat den Vorteil, dass Kommentare besser hervorgehoben werden und vor allem in längeren Programmen Kommentare stärker ins Auge stechen und gesuchte Stellen schneller gefunden werden können. Der Kommentar geht bei beiden Zeichen bis zum Ende der Zeile und muss nicht durch eine Wiederholung des Zeichens beendet werden.

Beispiele:

' Kommentar

@ Kommentar

```
@ ----- @  
@ --- sehr hervorgehobener Kommentar --- @  
@ ----- @
```

### 3.3.2 Optionen

Bei Verwendung von Programmen ist es meist nicht erwünscht, dass sich nach Eingabe einer Programmzeile ein Menüfenster öffnet, um die entsprechende Zeile genauer zu spezifizieren. Dies kann man umgehen, indem man die einzugebende Zeile im Eingabefenster solange verbessert, bis sich kein Fenster mehr öffnet und auch das gewünschte Ergebnis erreicht ist. Jetzt kann diese Zeile mit *Copy Paste* in das Programm übernommen werden, ohne dass es zu diesem Effekt kommt. In der *Command Reference Chapter 4*. sind die entsprechenden Optionen zu finden.

### 3.3.3 Verwendung von Variablen

Manchmal ist es sinnvoll, Platzhalter für die Variablen zu verwenden. Hierfür werden ! und % als Marker verwendet. Variablen, die mit % beginnen, können Stringtexte enthalten. Schreibt man mehrere solcher Variablen direkt hintereinander, werden die Referenzwerte zu einem String zusammengefasst. EViews unterscheidet hierbei, ob der Variablenname aus **einer** Zahl (0 - 9), **einem** Buchstaben (a - z) oder einem Namen (max. 16 Zeichen) besteht. Bei Namen sollte der Aufruf der Variable in { } gesetzt werden.

Variablen, die mit ! beginnen, enthalten Zahl Werte. Mit ihnen kann auch gerechnet werden.

EViews ersetzt alle Variablen durch ihren Referenzwert und führt den dadurch entstehenden regulären Ausdruck

aus. Hierdurch ist es insbesondere nicht direkt möglich, Leerzeichen in einer Variable zu verwenden. Insbesondere bei Pfadnamen ist dies zu beachten, da u.a. `C:\Dokumente und Einstellungen` nur dann in einer Variable verwendet werden kann, wenn der Aufruf der Variable in " " gesetzt wird und Leerzeichen im entsprechenden Ausdruck überhaupt zulässig sind.

Beispiel:

```
%Folder      = "C:\"
%Asciinput   = "Daten\Asciiformat"
%Excelinput  = "Daten\Excelformat"
!EViewsVersion = 6.1
!StartDate   = 1991
!EndDate     = 2009

create q {!StartDate}.1 {!EndDate}.4  <- keine Leerzeichen möglich
cd {%Folder}\{%Excelinput}            <- keine Leerzeichen möglich
cd "{%Folder}\{%Excelinput}"          <- Leerzeichen möglich
```

### 3.3.4 Schleifen

Die einfachste Schleife, die EViews zur Verfügung stellt, ist die *FOR NEXT* Schleife. *FOR* bezeichnet den Schleifenkopf, die Argumente stehen in der gleichen Zeile dahinter, nach dem ersten Zeilenumbruch beginnt die Schleife. *NEXT* beschreibt das Ende der Schleife und beginnt einen neuen Schleifendurchlauf. Die *FOR NEXT* Schleife kann in EViews noch zwei Ausprägungen annehmen.

**FOR Schleife mit String Variablen.** Hierbei wird eine Liste mit Strings direkt in den Schleifenkopf geschrieben. Die Variable nimmt nun einen String nach dem anderen an. Es können auch mehrere Variablen gleichzeitig verändert werden. EViews arbeitet die Strings dann in Gruppen ab.

Beispiel:

```
for %1 z3 con yv
  plot %1
next
```

das entspricht

```
plot z3
plot con
plot yv
```

**FOR Schleife mit Zahl Variablen.** Hierbei handelt es sich um die klassische For Schleife aus vielen anderen Programmiersprachen mit Counter, Start- und Endwert, sowie optionaler Schrittgröße.

Beispiel:

```
For !i = 1 to 3

  Zinsdifferenz(!i) = z3(!i) - zw(!i)

next
```

das entspricht

$$\text{Zinsdifferenz}(1) = z3(1) - \text{zwp}(1)$$

$$\text{Zinsdifferenz}(2) = z3(2) - \text{zwp}(2)$$

$$\text{Zinsdifferenz}(3) = z3(3) - \text{zwp}(3)$$

### 3.3.5 Sonstige Tipps und Kleinigkeiten

EViews kann wie die meiste englischsprachige Software nicht mit Umlauten umgehen. Mit etwas Mühe ist es dennoch möglich, Umlaute zu verwenden. Hierzu muss der Umlaut von Hand, oder mit einem Programm, jedesmal wieder gesetzt werden. Umlaute werden beim Verlassen von EViews nicht abgespeichert und führen zu einem Zeilenumbruch an der entsprechenden Stelle beim nächsten Öffnen des Workfiles.

Ähnliche Sorgfalt ist mit dem Umgang von Leerzeichen geboten. Diese müssen meist (aber nicht immer) in " " gesetzt werden.

Die Verwendung von Groß- und Kleinschreibung ist völlig irrelevant, da EViews alles in Kleinbuchstaben konvertiert. Einzige Ausnahme ist die Variablenbeschreibung und der Displayname. Die Verwendung von Groß- und Kleinschreibung dient also ausschließlich der Lesbarkeit der Programme.

Schreibt man längere EViews Programme, kann es unter Umständen sinnvoll sein, das Programm nicht immer komplett ablaufen zu lassen. Mit *STOP* kann an jeder beliebigen Stelle im Programm ausgestiegen werden. Ein Einstieg in das Programm an einer bestimmten Stelle ist nicht möglich.