

## **Die Makro-Sprache als Hilfsmittel bei komplexen Model-Statements am Beispiel von PROC NLIN**

Thomas Klein, Frank O. Martin

Institut für Soziologie  
Universität Heidelberg

### **Abstract**

Bei der Anwendung der Prozedur NLIN kann das MODEL-Statement mit einer steigenden Anzahl von Variablen und Parametern sehr unübersichtlich werden. Um Fehler und Schreibarbeit zu vermeiden kann man die Makrosprache verwenden, um die Generierung des MODEL-Statements zu automatisieren. In dem Beitrag soll - ausgehend von einem konkreten Beispiel - aufgezeigt werden, wie mit Hilfe der Makrosprache die genannten Probleme vermieden werden können.

### **Theoretische Vorbemerkungen zu Determinanten des Einkommens**

Die Humankapitaltheorie ist einer der wichtigsten theoretischen Ansätze, die es im Zusammenhang mit Einkommensdeterminanten gibt. Bereits Ende der 50er bzw. Anfang der 60er Jahre wurde dieser Ansatz von Jakob Mincer und Gary S. Becker entwickelt (Mincer 1974; Becker 1993). Auf diesem Ansatz basieren bis heute Analysen zu Einkommens-determinanten (vgl. z.B. Dworschak 1986; Szydlík 1993; Wunderlich 1997).

Im Rahmen der Humankapitaltheorie wird davon ausgegangen, daß das Einkommen im Lebensverlauf durch die Anhäufung von Humankapital maximiert wird. Die wichtigsten Faktoren sind dabei Schulbildung und Berufserfahrung. Investitionen, die getätigt werden, schlagen sich danach im späteren Zeitverlauf in einem höheren Einkommen nieder. Modelltechnisch läßt sich dieser Zusammenhang durch ein einfaches OLS-Regressionsmodell formulieren als:

$$\ln y = \beta_0 + \beta_1 s + \beta_2 x + \beta_3 x^2 + u$$

wobei  $y$  für das Einkommen,  $s$  für die Schuldauer in Jahren und  $x$  für die Berufserfahrung in Jahren steht. Durch Hinzunahme der quadrierten Berufserfahrung in Jahren wird dabei der abnehmende Grenzertrag der Berufserfahrung modelliert. Da die abhängige Variable logarithmiert ist, handelt es sich hierbei faktisch um ein multiplikatives Modell.

$$y = e^{(\beta_0 + \beta_1 s + \beta_2 x + \beta_3 x^2 + u)}$$

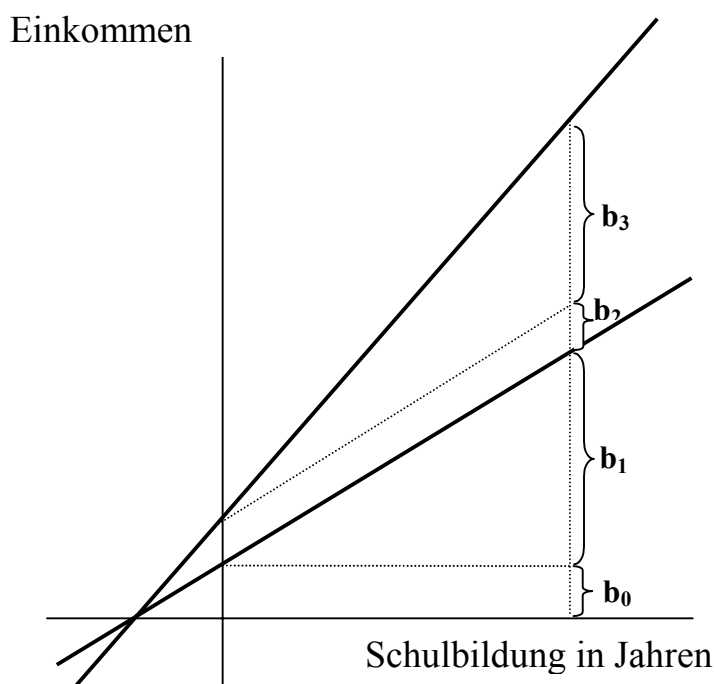
$$y = e^{\beta_0} * e^{\beta_1 s} * e^{\beta_2 x} * e^{\beta_3 x^2} * e^u$$

Dieses Grundmodell wurde im Zeitverlauf auch um Variablen erweitert, die nicht einzelnen Personen und deren Humankapital zuzurechnen sind, sondern vielmehr auf Seiten der Arbeitsnachfrage zu verorten sind, wie z.B. Betriebsgröße, Wirtschaftszweig und Region. Randy Hodson hat in diesem Zusammenhang u.E. mit Recht darauf hingewiesen, daß: "Many, if not most, characteristics of social structure (...) cannot, in any sense, be interpreted in terms of investment decisions" (Hodson 1985: 377). Ein Ansatz, der diesem Einwand Rechnung tragen soll, ist die gleichzeitige Modellierung multiplikativer und additiver Effekte in einem Regressionsmodell. Im folgenden soll dieser Ansatz kurz skizziert werden, bevor dann die technische Umsetzung mit Hilfe der Makro-Sprache ausgeführt wird.

## Die Spezifikation von additiven und multiplikativen Effekten in einem Regressionsmodell

Der hier skizzierte Ansatz basiert auf einem additiven Regressionsmodell mit Interaktionseffekten, bei dem jedoch die Parameter der Interaktionseffekte restringiert sind. Bei einem Modell mit zwei unabhängigen Variablen wird dabei der Interaktionseffekt dergestalt restringiert, daß sich der unrestringierte Parameter in einen Prozenteffekt umrechnen läßt.

**Abbildung 1:** Regressionsmodell mit restringiertem Interaktionseffekt



Die Gleichung für ein solches Modell würde dabei z.B. folgendermaßen lauten:  
 $y = \beta_0 + \beta_1 \text{ bild} + \beta_2 \text{ geschlecht} + \beta_3 \text{ bild geschlecht}$

wobei durch die Restriktion

$$\frac{\beta_2}{\beta_0} = \frac{\beta_3}{\beta_1}$$

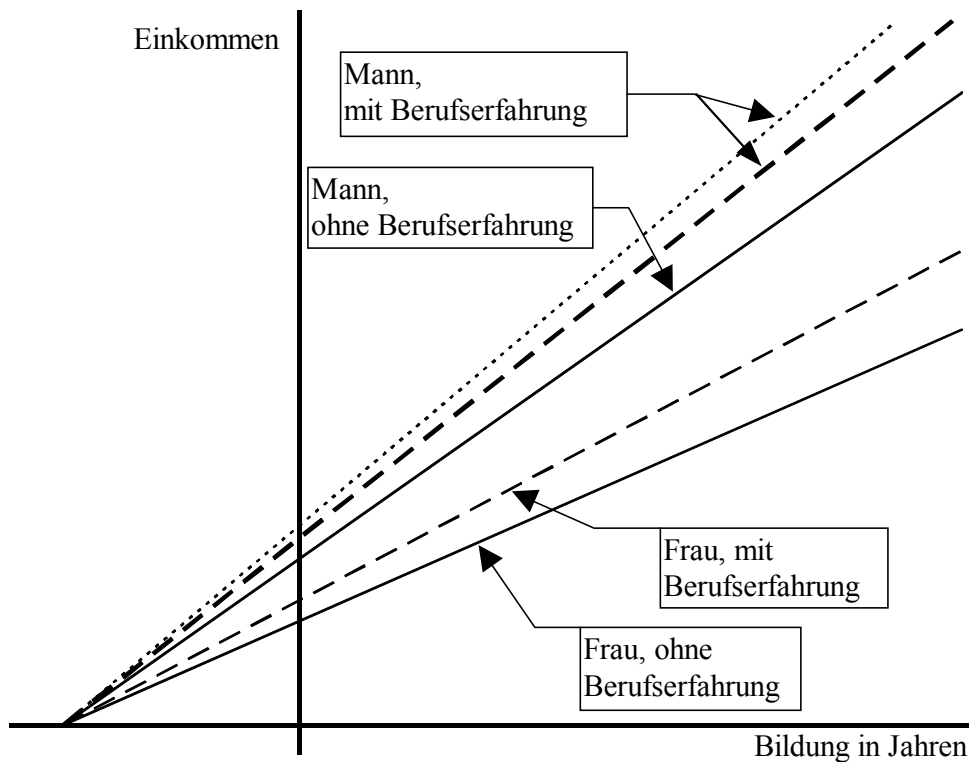
folgt, daß

$$\beta_2 = \frac{\beta_0 \beta_3}{\beta_1}$$

Der Quotient der geschätzten Parameter für  $\beta_3$  und  $\beta_1$  wäre dabei dann der Prozenteffekt für das Merkmal Geschlecht.

Ein Modell mit mehreren unabhängigen Variablen ist in Abbildung 2 schematisch dargestellt. Anhand dieser Abbildung läßt sich auch gut der Unterschied zwischen nebeneinander stehenden und aufsattelnden Effekten zeigen. Denn in Relation zu den durchgezogenen Linien (Geschlecht) sind auch gestrichelte Linien eingezeichnet, die das höhere Einkommen bei größerer Berufserfahrung darstellen sollen. Die gepunktete Linie steht dabei für das Einkommen, wenn die Berufserfahrung auf das Geschlecht aufsattelt.

Abbildung 2



Eine solche Modellierung kann man durch zusätzliche Restriktionen der Parameter erreichen. Bei der Modellierung

$$\begin{aligned}
 y &= \beta_0 + \beta_1 \text{ bild} \\
 &+ \beta_2 G + \beta_3 \text{ bild } G \\
 &+ \beta_4 E + \beta_5 \text{ bild } E \\
 &+ \beta_6 G E + \beta_7 \text{ bild } G E,
 \end{aligned}$$

lauten die entsprechenden Parameterrestriktionen

$$\beta_2 = \frac{\beta_0 \beta_3}{\beta_1}, \quad \beta_4 = \frac{\beta_0 \beta_5}{\beta_1}, \quad \beta_6 = \frac{\beta_2 \beta_4}{\beta_0} \quad \text{und} \quad \beta_7 = \frac{\beta_3 \beta_5}{\beta_1}.$$

Die Prozenteffekte lassen sich dann als  $\beta_3/\beta_1$  sowie  $\beta_5/\beta_1$  berechnen.

## Die Umsetzung mit PROC NLIN (exemplarisch)

Im nachfolgenden Programmtext ist die Umsetzung einer Regression mit einem restringierten Parameter beispielhaft mit Hilfe von PROC NLIN ausgeführt.

```
proc nlin data=beispiel method=dud outest=out;

parameters b0=2500 _bild=300 _mann=100;

model y = b0 * 1
      + _bild * bild
      + (b0 * _mann / _bild) * mann
      + _mann * (bild * mann)
      ;
run;
```

Nach dem Aufruf der Prozedur und der Benennung der zu schätzenden Parameter (mit Startwerten) folgt das eigentliche Model-Statement. Dabei ist der Haupteffekt für die Dummy-Variable mann (0/1 Codierung) restringiert. Aus den geschätzten Parametern `_bild` und `_mann` läßt sich dabei der Prozentsatz errechnen, um den die geschätzte Gerade für Männer (mann=1) im Vergleich zu Frauen (mann=0) höher bzw. niedriger verläuft (vgl. auch Abbildung 1).

Im folgenden Programmtext ist ein weiteres Merkmal (Berufserfahrung) in der Modellierung berücksichtigt. Dabei ist modelltechnisch eine multiplikative Verknüpfung umgesetzt, d.h. der Effekt für Berufserfahrung sattet auf dem Effekt für das Geschlecht auf. Die Zahl der Restriktionen, die in dem Modell berücksichtigt werden müssen, ist entsprechend größer (vgl. auch Abbildung 2).

```
proc nlin data=beispiel method=dud outest=out;

parameters b0=2500 _bild=300 _mann=100 _berf=3;

model y = b0 * 1
      + _bild * bild
      + (b0 * _mann / _bild) * mann
      + _mann * (mann * bild)
      + (b0 * _berf / _bild) * berf
      + _berf * (berf * bild)
      + (b0 * _berf / _bild) * (b0 * _mann / _bild) / b0
      * (berf * mann)
      + (_berf * _mann / _bild) * (berf * mann * bild)
      ;
run;
```

## Automation mit der Makro-Sprache

Wie an den exemplarischen Programmtexten gut ersichtlich ist, wird das Model-Statement bereits bei einer geringen Anzahl von Variablen unübersichtlich. Für Analysen mit explorativem Charakter, bei denen eine größere Anzahl von Merkmalen im Model-Statement variiert werden soll, erscheint es daher notwendig, die Generierung der Restriktionen im Model-Statement mit Hilfe der Makro-Sprache zu automatisieren.

Im Kern basiert der hier skizzierte Weg der Automatisierung darauf, einer Makro-Variable eine Liste von Variablen zuzuweisen und danach diese Liste strukturiert abzuarbeiten. Das nachfolgende Makro erledigt diese Aufgabe, wobei durch die Schleifen-Konstruktion mit **%do %until**<sup>1</sup> ermöglicht wird, daß die Zahl der Variablen nicht extra spezifiziert werden muß.

```
%macro listloop;

%let zaehler = 1;
%do %until ( &element = %str() );
  %let element = %scan( &liste. , &zaehler. );
  %if ( &element. > %str() ) %then %do;

    %* BEGINN MACRO-OUTPUT;

    + (b0 * _&element. / _bild) *          &element.
    + _&element.                * (bild * &element.)

    %* ENDE MACRO-OUTPUT;

  %end;
  %let zaehler = %eval( &zaehler. + 1 );
%end;

%mend listloop;
```

Wenn mehrere Listen abgearbeitet werden sollen, die verschiedene Programmtexte generieren (wie im Fall der aufsattelnden Effekte), kann man die Listen und den jeweiligen Programmtext auch in Form von Argumenten übergeben. Das Makro muß dazu einerseits um Argumente erweitert werden, andererseits muß beim Aufruf des Makros verhindert werden, daß Makro-Variablen, die im Programmtext vorkommen, aufgelöst werden. Letzteres kann mit der Funktion **%nrstr** bei der Übergabe des Arguments bewerkstelligt werden, wobei innerhalb des Makros die Funktion **%unquote** dafür sorgt, daß die Makro-Variablen im Programmtext korrekt aufgelöst werden (vgl. Statements auf folgender Seite).

---

<sup>1</sup> im Gegensatz zu **%do %while** wird bei **%do %until** die Statement-Gruppe bis zum **%end** beim ersten Durchlauf in jedem Fall abgearbeitet. Die Bedingung kann damit auch erst innerhalb der Schleife initialisiert werden!

```
%macro listloop(liste, statemnt);  
  ...  
  %unquote(statemnt);  
  ...  
%mend listloop;  
  
/* Beispielaufruf mit Argumenten */  
  
%listloop( var1 var2 var3,  
          %nrstr(  
            + (b0 * &_element. / _bild) *      &element.)  
            + &_element.                      * (bild * &element.)  
          )  
        );
```

## Literaturverzeichnis

- Becker, Gary S.*, 1993: Human Capital. A Theoretical and Empirical Analysis with Special Reference to Education. Chicago, London: University of Chicago Press.
- Dworschak, Franz*, 1986: Struktur von Arbeitseinkommen. Methodische und inhaltliche Probleme von Regressionsanalysen mit Querschnittsdaten für Mikrosimulationsmodelle. Idstein: Schulz-Kirchner.
- Hodson, Randy*, 1985: Some Considerations Concerning the Functional Form of Earnings, Social Science Research 14: 374-394.
- Mincer, Jacob*, 1974: Schooling, Experience, and Earnings. New York, London: Columbia University Press.
- Szydlik, Marc*, 1993: Arbeitseinkommen und Arbeitsstrukturen. Eine Analyse für die Bundesrepublik Deutschland und die Deutsche Demokratische Republik. Berlin: Ed. Sigma.
- Wunderlich, Gaby*, 1997: Einkommensungleichheit. Ein Vergleich der alten und neuen Bundesländer. 45-77 in: *Walter Müller* (Hg.): Soziale Ungleichheit. Opladen: Leske + Budrich.