

Zusammenstellung aller im Buch abgedruckten SPSS-Syntax-Files

Im Buchtext sind die kompletten Programmzeilen zahlreicher Syntaxfiles bzw. Steuerfiles abgedruckt, mit denen die im Buch beschriebenen regressionsanalytischen Auswertungen unter Verwendung des EDV-Statistik-Programmpakets „SPSS“ durchgeführt werden können.

Alle diese Syntaxfiles werden unten aufgelistet und können mittels der beiden WINDOWS-Anweisungen „Kopieren“ und „Einfügen“ in jeden SPSS-Syntax-Editor eingefügt werden. Allerdings sind diese Syntaxfiles in aller Regel in SPSS nicht direkt lauffähig, sondern sie setzen zumindest voraus, dass in SPSS zuvor ein SPSS-Datenfile hochgeladen wurde.

Zusätzlich gilt für alle hier aufgeführten SPSS-Programmzeilen:

- komplett großgeschriebene Textstrings dürfen in SPSS nicht verändert werden (z.B.: COMPUTE);
- kursiv und zumeist kleingeschriebene Textstrings sind anwenderdefinierte Textstrings (z.B.: *index*, *Index*);
 - (-) diese sind entweder vom Anwender frei zu bestimmen:
z.B.: COMPUTE *zahl*=1.
 - (-) oder diese ergeben sich aus dem benutzten Datensatz mit den darin festgelegten Variablennamen:
z.B.: RECODE *income* (MISSING=1).
 - (-) oder diese ergeben sich aus zuvor in SPSS vergebenen Variablen- oder Dateinamen, die wieder neu aufgegriffen werden:
z.B.: DATASET ACTIVATE *datei2*.

* SPSS-Syntax 1.1

```
COMPUTE index = einst1 + einst2 + einst3.
```

* SPSS-Syntax 1.2

```
IF MISSING(x1)=0 AND MISSING(x2)=0 AND MISSING(y)=0 vfilter=1.  
FILTER BY vfilter.  
FRE x1 x2 y / statistics =all .
```

*SPSS-Syntax 3.1

```
REGRESSION  
/MISSING LISTWISE  
/STATISTICS COEFF OUTS CI R ANOVA CHANGE ZPP  
/CRITERIA=PIN(.05) POUT(.10)  
/NOORIGIN  
/DEPENDENT Ausländerablehnung  
/METHOD=ENTER Autoritarismus  
/METHOD=ENTER Prestige.
```

* SPSS-Syntax 4.1

```
/SAVE RESID ZRESID SDRESID
```

* SPSS-Syntax 4.2

COMPUTE PARTRES_k = RES_1 + *Regressionskoeffizient* * X-Variable.

* SPSS-Syntax 4.3

GRAPH

/SCATTERPLOT(BIVAR)= *alter* WITH ZRE_1

/MISSING=LISTWISE.

* SPSS-Syntax 4.4

/SCATTERPLOT>(*ZRESID ,*ZPRED)

* bzw.:

/SCATTERPLOT>(*SDRESID ,*ZPRED)

* SPSS-Syntax 4.5

TSPLIT VARIABLES= ZRE_1.

* bzw.:

TSPLIT VARIABLES= SDR_1.

* SPSS-Syntax 4.6

/CASEWISE PLOT(ZRESID) OUTLIERS(2)

* SPSS-Syntax 4.7

/RESIDUALS = OUTLIER(MAHAL)

* SPSS-Syntax 4.8

/SAVE LEVER SDRESID SDBETA SDFIT

* SPSS-Syntax 4.9

FRE ZRE_1 /STATISTICS=SKEWNESS SESKEW KURTOSIS SEKURT.

* SPSS-Syntax 4.10

/RESIDUALS HIST(ZRESID)

* SPSS-Syntax 4.11

NPART TESTS /K-S(NORMAL)= ZRE_1 /MISSING ANALYSIS.

* SPSS-Syntax 4.12

/RESIDUALS NORM(ZRESID)

* SPSS-Syntax 4.13
COMPUTE *alter2* = *alter***alter*.

* SPSS-Syntax 4.14
/SAVE PRED

* SPSS-Syntax 4.15
CURVEFIT
/VARIABLES=*Einkommen* WITH *Alter* /CONSTANT
/MODEL=LINEAR QUADRATIC /PLOT FIT /SAVE=PRED.

* SPSS-Syntax 4.16
RECODE *au* (3=21) (4=20) (5=19) (6=18) (7=17) (8=16) (9=15) (10=14) (11=13)
(12=12) (13=11) (14=10) (15=9) (16=8) (17=7) (18=6) (19=5) (20=4) (21=3)
INTO *anti_au*.

* SPSS-Syntax 4.17
COMPUTE *anti_au* = 24 - *au*.

* SPSS-Syntax 4.18
COMPUTE *interaktion* = *prestige***anti_au*.

* SPSS-Syntax 4.19
GET FILE='Datei0.sav'.
DESCRIPTIVES VARIABLES=*aa anti_au prest*
/SAVE.
COMPUTE *Zinter*=*Zprest***Zanti_au*.
EXECUTE.
DATASET ACTIVATE DataSet1.
SAVE OUTFILE='Datei2.sav'.

* SPSS-Syntax 4.20
COMPUTE *nres*=ABS(*ZRE_1*).

* SPSS-Syntax 4.21
CORR *alter nres*.

* SPSS-Syntax 4.22
GET FILE='Datei0.sav'.
COMPUTE *C_anti_au* = *anti_au* - 14.60.
COMPUTE *C_prest* = *prest* - 94.00.

```
COMPUTE C_interaktion = C_anti_au * C_prest.  
EXECUTE.  
DATASET ACTIVATE DataSet1.  
SAVE OUTFILE='Datei2.sav'.
```

```
* SPSS-Syntax 4.23  
EXAMINE VARIABLES=ZRE_1 BY x-variable  
/PLOT=BOXPLOT  
/STATISTICS=DESCRIPTIVES  
/NOTOTAL.
```

```
* SPSS-Syntax 4.24  
COMPUTE group = -1.  
IF einst > median group = 0.  
IF einst <= median group = 1.  
MIS VAL group (-1).
```

```
* SPSS-Syntax 4.25  
ONEWAY  
ZRE_1 BY group  
/STATISTICS HOMOGENEITY  
/MISSING ANALYSIS.
```

```
* SPSS-Syntax 4.26  
/RESIDUALS DURBIN
```

```
* SPSS-Syntax 4.27  
COMPUTE nres=ABS(ZRE_1
```

```
* SPSS-Syntax 4.28  
REGRESSION  
/dep nres  
/method=enter x1 x2 x3 x4 x5.
```

```
* SPSS-Syntax 4.29  
WLS aa WITH au pr  
/SOURCE h  
/POWER -2 TO 2 BY 0.2  
/CONSTANT  
/SAVE WEIGHT  
/PRINT best.
```

```
* SPSS-Syntax 4.30
```

```
REGRESSION
/MISSING LISTWISE
/REGWGT=WGT_1
/STATISTICS COEFF OUTS R
/CRITERIA=PIN(.05) POUT(.10)
/NOORIGIN
/DEPENDENT aa
/METHOD=ENTER au pr
/SAVE PRED RESID.
```

```
* SPSS-Syntax 4.31
COMPUTE residuum = SQRT(WGT_1)*RES_1.
COMPUTE vorhergesagt = SQRT(WGT_1)*PRE_1.
```

```
* SPSS-Syntax 4.32
DESCRIPTIVES VARIABLES=residuum vorhergesagt /SAVE.
```

```
* SPSS-Syntax 4.33
GRAPH
/SCATTERPLOT(BIVAR)=zvorhergesagt WITH zresiduum.
```

```
* SPSS-Syntax 4.34
GENLIN y WITH x1 x2
/MODEL x1 x2
/CRITERIA COVB=ROBUST SCALE=MLE
/PRINT SOLUTION.
```

```
* SPSS-Syntax 4.35
/RESIDUALS DURBIN
```

```
* SPSS-Syntax 4.36
CREATE
/ZRE_1_1=LAG(ZRE_1 1).
```

```
* SPSS-Syntax 5.1
COMPUTE d1 = 9.
COMPUTE d2 = 9.
IF (einstell = 1) d1 = 1.
IF (einstell = 2) d2 = 1.
IF (einstell = 1) or (einstell = 3) d2 = 0.
IF (einstell = 2) or (einstell = 3) d1 = 0.
MIS VAL d1 d2 (9).
```

* SPSS-Syntax 5.2
COMPUTE *ek1*=9.
COMPUTE *ek2*=9.
IF (*einstell* = 1) *ek1*=1.
IF (*einstell* = 1) *ek2*=0.
IF (*einstell* = 2) *ek2*=1.
IF (*einstell* = 2) *ek1*=0.
IF (*einstell* = 3) *ek1*=-1.
IF (*einstell* = 3) *ek2*=-1.
MIS VAL *ek1 ek2* (9).

*SPSS-Syntax 5.3
IF (*einstell* = 3) *ek1*=-557/114.
IF (*einstell* = 3) *ek2*=-328/114.

* SPSS-Syntax 6.1
SORT CASES BY *group*.
SPLIT FILE BY *group*.

* SPSS-Syntax 7.1
REGRESSION
/STATISTICS COEFF OUTS CI R ANOVA CHANGE
/DEPENDENT *prestige_sohn*
/METHOD=ENTER *prestige_vater bildung_vater*
/METHOD=ENTER *einkommen_sohn bildung_sohn*
/METHOD=ENTER *alter_sohn*.

* SPSS-Syntax 7.2
PRESERVE.
SET RNG=MT MTINDEX=2000000.
SHOW RNG.
BOOTSTRAP
/SAMPLING METHOD=SIMPLE
/VARIABLES TARGET= *y-variable* INPUT= *x-variablen*
/CRITERIA CILEVEL=95 CITYPE=BCA NSAMPLES=1000
/MISSING USERMISSING=EXCLUDE.
* *an dieser Stelle ist der lineare oder logistische SPSS-Regressionsbefehl*
* *einzufragen nach SPSS-Syntax 3.1 oder nach SPSS-Syntax 8.9*
RESTORE.

* SPSS-Syntax 8.1
GRAPH /SCATTERPLOT(BIVAR)= *lr* WITH PRE_1 BY *gew*.

* SPSS-Syntax 8.2
LOGISTIC REGRESSION VARIABLES *cdu*

/METHOD=ENTER Zgew Zlr.

* SPSS-Syntax 8.3

GRAPH
/SCATTERPLOT(BIVAR)= lr WITH me
/MISSING=LISTWISE.

* SPSS-Syntax 8.4

LOGISTIC REGRESSION VARIABLES Y-Variable
/METHOD=ENTER X1-Variable X2-Variable.
COMPUTE b_0 = Intercept des Regressionsmodells.
COMPUTE b_1 = b_1 -Koeffizient der X1-Variablen.
COMPUTE b_2 = b_2 -Koeffizient der X2-Variablen.
COMPUTE z_0 =(b_0) + ($b_1 * 0$) + ($b_2 * X_2$ -Variable).
COMPUTE exp_0 =exp(z_0).
COMPUTE p_0 =exp(z_0)/(1+exp(z_0)).
COMPUTE z_1 =(b_0) + ($b_1 * 1$) + ($b_2 * X_2$ -Variable).
COMPUTE exp_1 =exp(z_1).
COMPUTE p_1 =exp(z_1)/(1+exp(z_1)).
COMPUTE me =($p_1 - p_0$). *Bem.: Ermittlung der marginalen Effekte von X1
FRE me / STATISTICS=MEAN. *Bem.: der Mean entspricht dem AME von X1

* SPSS-Syntax 8.5

LOGISTIC REGRESSION VARIABLES Y-Variable
/METHOD=ENTER X1-Variable X2-Variable.
COMPUTE b_0 = Intercept des Regressionsmodells.
COMPUTE b_1 = b_1 -Koeffizient der X1-Variablen.
COMPUTE b_2 = b_2 -Koeffizient der X2-Variablen.
FRE X2-Variable /STATISTICS=STDDEV. *Bem.: Ermittlung der StAbw. von X2
COMPUTE δ = Standardabweichung von X2 /1000.
COMPUTE z_0 =(b_0) + ($b_1 * X_1$ -Variable) + ($b_2 * X_2$ -Variable).
COMPUTE exp_0 =exp(z_0).
COMPUTE p_0 =exp(z_0)/(1+exp(z_0)).
COMPUTE z_1 =(b_0) + ($b_1 * X_1$ -Variable) + ($b_2 * (X_2$ -Variable + δ)).
COMPUTE exp_1 =exp(z_1).
COMPUTE p_1 =exp(z_1)/(1+exp(z_1)).
COMPUTE me =($p_1 - p_0$)/(δ). *Bem.: Ermittlung der marginalen Effekte von X2
FRE me / STATISTICS=MEAN. *Bem.: der Mean entspricht dem AME von X2

* SPSS-Syntax 8.6

CROSSTABS /TABLES=x1 BY x2 /CELLS=COUNT EXPECTED.

* SPSS-Syntax 8.7

COMPUTE x_ln = ln(x).

* SPSS-Syntax 8.8
METHOD = ENTER $x \cdot x^*x_{ln}$

* SPSS-Syntax 8.9
GET FILE = "xxx.SAV".
LOGISTIC REGRESSION VARIABLES *cdu*
/METHOD = ENTER *gew lr*
/PRINT= ITER(1)
/SAVE = PRED DFBETA LEVER RESID SRESID ZRESID LRESID.
SAVE OUTFILE = "xxx.SAV" / KEEP = ALL.

* SPSS-Syntax 8.10
GET FILE='xxx.SAV'.
REGRESSION
/DEPENDENT *lr*
/METHOD=ENTER *gew*
/SAVE RESID.
SAVE OUTFILE='xxx2.SAV' /KEEP=ALL.
GET FILE='xxx2.SAV'.
LOGISTIC REGRESSION VARIABLES *cdu*
/METHOD = ENTER *gew RES_1*.

* SPSS-Syntax 9.1
COUNT *missk* = $y \cdot x_1 \cdot x_2$ (MISSING).
IF (*missk*>0) *missk*=1.
FREQ *missk*.

* SPSS-Syntax 9.2
COMPUTE *AS_Index* = MEAN.4(*item1*, *item2*, *item3*, *item4*).

* SPSS-Syntax 9.3
COUNT *missk*= *AS_Index*, *Bildung*, *Alter* (MISSING).
IF (*missk* >0) *missk* =1.
FREQ *missk*.

* SPSS-Syntax 9.4
RECODE *AS_Index Bildung Alter* (MISSING=1) (ELSE=0) INTO *miss1 miss2 miss3*.
EXECUTE.
FREQ VARIABLES=*miss1 miss2 miss3*.

* SPSS-Syntax 9.5
MVA VARIABLES = *AS_Index, Bildung, Alter* /EM.

* SPSS-Syntax 9.6
RECODE AS_Index (MISSING=1) (ELSE=0) INTO AS_miss.
EXECUTE.
FREQ VARIABLES=AS_miss.

*SPSS-Syntax 9.7
LOGISTIC REGRESSION VARIABLES AS_miss
/METHOD=ENTER *Bildung Alter*.

* SPSS-Syntax 9.8
REGRESSION
/MISSING LISTWISE
/DEPENDENT AS_Index
/METHOD=ENTER *Alter Bildung*.

* SPSS-Syntax 9.9
MVA VARIABLES = AS_Index *Alter Bildung Linksrechts*
/REGRESSION AS_Index *WITH Alter Bildung Linksrechts*
(ADDTYPE=NORMAL OUTFILE='AS2.sav').

* SPSS-Syntax 9.10
REGRESSION
/DEPENDENT AS_Index
/METHOD=ENTER *Alter Bildung*.

* SPSS-Syntax 9.11
DATASET DECLARE AS_imputiert WINDOW=HIDDEN.
MULTIPLE IMPUTATION AS_Index *Alter Bildung Linksrechts Geschl*
/IMPUTE METHOD=FCS NIMPUTATIONS=20
/IMPUTATIONSUMMARIES MODELS
/OUTFILE IMPUTATIONS=AS_imputiert.
DATASET ACTIVATE AS_imputiert WINDOW=ASIS.
REGRESSION
/STATISTICS COEFF OUTS CI(95) R ANOVA
/DEPENDENT AS_Index
/METHOD=ENTER *Alter Bildung*.

* SPSS-Syntax 9.12
DATASET DECLARE AS_imputiert WINDOW=HIDDEN.
MULTIPLE IMPUTATION AS_Index *Alter Bildung Linksrechts Geschl*
/IMPUTE METHOD=FCS NIMPUTATIONS=20
/CONSTRAINTS *Alter* (ROLE=IND)
/CONSTRAINTS *Bildung* (ROLE=IND)
/CONSTRAINTS *Linksrechts* (ROLE=IND)
/CONSTRAINTS *Geschl* (ROLE=IND)

```
/IMPUTATIONSUMMARIES MODELS
/OUTFILE IMPUTATIONS=AS_imputiert.
DATASET ACTIVATE AS_imputiert WINDOW=ASIS.
REGRESSION
/STATISTICS COEFF OUTS CI(95) R ANOVA
/DEPENDENT AS_Index
/METHOD=ENTER Alter Bildung.
```

```
* SPSS-Syntax 9.13
DATASET DECLARE AS_imputiert WINDOW=HIDDEN.
MULTIPLE IMPUTATION AS_Index Alter Bildung Linksrechts Geschl
/IMPUTE METHOD=FCS NIMPUTATIONS=20
/IMPUTATIONSUMMARIES MODELS
/OUTFILE IMPUTATIONS=AS_imputiert.
DATASET ACTIVATE AS_imputiert WINDOW=ASIS.
REGRESSION
/STATISTICS COEFF OUTS CI(95) R ANOVA
/DEPENDENT AS_Index
/METHOD=ENTER Alter Bildung.
DESCRIPTIVES AS_Index Alter Bildung
/SAVE.
REGRESSION
/DEPENDENT ZAS_Index
/METHOD=ENTER ZAlter ZBildung.
```

```
* SPSS-Syntax 9.14
/CONSTRAINTS Bildung (MIN=1 MAX=5 RND=1)
```

```
* SPSS-Syntax 9.15
VARIABLE LEVEL Praeferenz(NOMINAL).
EXECUTE.
```

```
* SPSS-Syntax 9.16
WEIGHT BY v870.
```

```
* SPSS-Syntax 9.17
* (in kompletter Version als SPSS-Syntax 9.13 in Unterkapitel 9.3.3 zu finden)
DATASET DECLARE AS_imputiert WINDOW=HIDDEN.
MULTIPLE IMPUTATION AS_Index Alter Bildung Linksrechts Geschl
* usw. entsprechend SPSS-Syntax 9.13
/OUTFILE IMPUTATIONS=AS_imputiert.
DATASET ACTIVATE AS_imputiert WINDOW=ASIS.
REGRESSION
* usw. entsprechend SPSS-Syntax 9.13
DESCRIPTIVES AS_Index Alter Bildung
SAVE.
```

* usw. entsprechend SPSS-Syntax 9.13

* SPSS-Syntax 9.18
DATASET ACTIVATE *AS_imputiert*.
WEIGHT BY *g1*.