

```

*****
* Dieses Stata-Do-File generiert die Multidimensionalen Lagen *
* nach "Groh-Samberg, Buechler, Gerlitz: *
* Dokumentation zur Generierung Multidimensionaler Lagen *
* auf Basis des Sozio-Oekonomischen Panel. Bremen, 15.02.2021" *
* ***** *
* Datenbasis: SOEPv35 *
*****

*****
* Pfadangaben definieren *****
*****
global work "[Pfad]" /* Hier das Arbeitsverzeichnis einsetzen */
global soeplong "[Pfad]" /* Hier die Pfadangabe fuer SOEPv35-Daten (long) einsetzen.*/
global soepggk "[Pfad]" /* Hier die Pfadangabe fuer die SOEP-Gemeindegroessenklassen (ggkbou)
einsetzen.*/

* Clear
clear all
set more off
capture: log close _all
cd "$work"

*****
* Gemeindegroessenklassen *****
*****

use hid *ggk using "$soepggk/ggkbou", clear
forvalues i=1/35 {
local wave=char(`i'+96)
if `i'>26 {
local wave="b"+char(`i'+70)
}
rename `wave'ggk ggk`i'
}
reshape long ggk, i(hid) j(year)
keep if ggk>-2
gen syear=year+1983
keep hid syear ggk
save "$work/ggk", replace

*****
* Variable "In Ausbildung" erstellen *****
*****

use pid syear plg0012 plg0013_v1 plg0013_v3 plg0014_v? plg0292_h plg0293_h using "$soeplong/pl", clear

label define ausbild ///
0 "nicht in Ausbildung" ///
-1 "keine Angabe" ///
1 "Hauptschule" ///
2 "Realschule" ///
3 "Gymnasium" ///
4 "Gesamtschule" ///
5 "Abendschule, Gymnasium" ///
6 "Fachoberschule" ///
7 "Willk./Vorbereitungsklasse" ///
8 "Fachhochschule" ///
9 "Uni.sonst.Hochschule" ///
10 "Berufsgrundbildungsjahr,etc" ///
11 "Berufsschule ohne Lehre" ///
12 "Lehre" ///
13 "Berufsfachschule,etc." ///
14 "Schule des Gesundheitswesens" ///
15 "Fachschule" ///
16 "Beamtenausbildung" ///
17 "Sonstige berufl. Ausbildung" ///
18 "Berufl. Umschulung" ///
19 "Berufl. Fortbildung" ///
20 "Berufl. Rehabilitation" ///
21 "Allg.,polit.Weiterbildung" ///
22 "Sonstige Weiterbildung" ///
23 "Integrationskurs / Sprachkurs" ///
24 "Mehrfachnennung"
gen ausbildung=ausbild=0
gen plg0013=plg0013_v1
egen uni=anycount(plg0014_v1 plg0014_v2 plg0014_v4 plg0014_v5 plg0014_v6 plg0014_v7), values(2/5)
egen fh=anycount(plg0014_v1 plg0014_v2 plg0014_v3 plg0014_v5 plg0014_v6 plg0014_v7), values(1)
replace plg0013=plg0013_v3 if inrange(syear,2016,2017)
replace ausbildung=plg0013 if inrange(plg0013,1,7)
replace ausbildung=8 if fh==1
replace ausbildung=9 if uni==1
replace ausbildung=plg0293_h+9 if inrange(plg0293_h,1,8) & ~inlist(ausbildung,8,9)
replace ausbildung=17 if ausbildung==0 & plg0293_h==9

```

```

replace ausbildung=plg0292_h+17 if inrange(plg0292_h,1,6) & ~inlist(ausbildung,8,9)
replace ausbildung=22 if ausbildung==0 & plg0292_h==7
replace ausbildung=-1 if ausbildung==0 & inlist(plg0012,-1,1)
keep pid syear ausbildung
save "$work/ausbildung", replace

*****
* Daten einlesen *****
*****

use pid syear sex gebjahr migback psample netto hid sampreg pop phrf pbleib using "$soeplong/ppath1.dta",
clear
merge 1:1 pid syear using "$soeplong/pequiv", nogen keep(1 3) keepusing(i11102 i11104 i11105 i11110 y11101)
merge 1:1 pid syear using "$soeplong/pgen", nogen keep(1 3) keepusing(pglfs pgemplst pgjobch pgstib)
merge 1:1 pid syear using "$soeplong/pl", nogen keep(1 3) keepusing(plb0031_h plb0037_h)
merge 1:1 pid syear using "$work/ausbildung", nogen keep(1 3)
merge m:1 hid syear using "$soeplong/hgen", nogen keep(1 3) keepusing(hgsize hgroom hgowner)
merge m:1 hid syear using "$soeplong/hl", nogen keep(1 3) keepusing(hlc0093 hlc0096 hlc0104 hlc0105 hlc0106
hlc0107 hlc0108)
merge m:1 hid syear using "$soeplong/hpath1", nogen keep(1 3) keepusing(hnetto)
merge m:1 hid syear using "$work/ggk", nogen keep(1 3)
gen welle=syear-1983

*****
* Population Querschnitt (nur Privat-HH) *****
*****

keep if hnetto==1 & inrange(netto,10,39) & inlist(pop,1,2) & psample<16
drop if inlist(syear,1990,1991) & sampreg==2
* löschen von HH mit fehlenden Jahreseinkommen (50 Personen in 23 HH, alle Ost 1990)
drop if i11102==-2

*****
* Einkommen *****
*****

* Inflationsbereinigung
for var i111*: replace X=X/y11101*100
* oecd
gen alter=syear-gebjahr if gebjahr>0
recode alter (0/14=0.3) (15/200=0.5) (.=0.5), into(bedarf)
egen oecd=total(bedarf), by(hid welle)
replace oecd=oecd+0.5
* Einkommen
gen inc=i11102+i11105
gen incq=inc/oecd

* Medianeinkommen
gen phrfn=phrf if phrf>0 & phrf<.
bysort welle (incq): gen popcum=sum(phrfn)
bysort welle (incq): gen medinc=incq if popcum>=(popcum[_N]/2) & popcum[_n-1]<(popcum[_N]/2)
egen mediinc=total(medinc), by(welle)
drop medinc popcum

* Log Grenzwerte
log using "$work/medianinc.log", replace
table syear, c(min mediinc) format(%4.0f)
log close

* Einkommensgruppen
gen posinc=incq/mediinc
recode posinc (0/0.6=1) (0.6/0.8=2) (0.8/1.2=3) (1.2/2=4) (2/max=5), into(poskatinc)

*****
* Dimension Erwerbsarbeit *****
*****
*** ökonomische Unabhängigkeit:
*** Individuallohn < Armutsgrenze
*** für alle Erwerbspersonen im HH: 18-65, nicht in Ausbildung & nicht in Rente

**** FIX BEFRISTUNG ****
gen frist:plb0037_h=plb0037_h
bysort pid (syear): replace frist=frist[_n-1] if frist==-2 & pgjobch==2 & inlist(pgemplst,1,2,3,4,6) &
inrange(frist[_n-1],1,4) ///
& pid=pid[_n-1] & syear==syear[_n-1]+1 & inrange(syear,1986,1994)
bysort pid (syear): replace frist=frist[_n-1] if ~inrange(frist,1,4) & pgjobch==2 &
inlist(pgemplst,1,2,3,4,6) & inrange(frist[_n-1],1,4) ///
& pid=pid[_n-1] & syear==syear[_n-1]+1
replace frist=1 if frist==-2 & sampreg==2 & inrange(syear,1992,1994) & inlist(pgemplst,1,2,3,4,6)

gen indep=i11110>=mediinc*0.6 if i11110<.
gen ewper=inrange(alter,18,65) & inlist(pglfs,1,6,8,9,10,11,12) & ~inrange(ausbildung,1,20)
replace ewper=1 if indep=1
gen sicher=indep==1 & frist~2

```

```

gen unsicher=indep==1 & frist==2
gen noinc=ewper==1 & indep==0
gen zeroinc=ewper==1 & ill110==0
label define ewind 0 "reg. NEW" 1 "NEW Ewp" 2 "lowinc" 3 "unsicher" 4 "sicher"
gen ewind:ewind=ewper*2-zeroinc+indep+sicher

egen ewpers=total(ewper), by(hid welle)
egen indeps=total(indep), by(hid welle)
egen sichers=total(sicher), by(hid welle)
egen unsichers=total(unsicher), by(hid welle)
egen noincs=total(noinc), by(hid welle)
egen zeroincs=total(zeroinc), by(hid welle)

label define ewarbeit 1 "erwerbsarm" 2"prekaer" 3 "gemischt" 4 "gesichert" 5"erwerbsreich" 6 "nicht-EW-HH"
gen ewarbeit:ewarbeit=2
replace ewarbeit=1 if indeps==0 & zeroincs>=1
replace ewarbeit=3 if sichers>=1 & noincs>=1
replace ewarbeit=4 if sichers>=1 & indeps==ewpers & ewpers>0
replace ewarbeit=5 if sichers==ewpers & ewpers>1
replace ewarbeit=6 if ewpers==0

*****
* Dimension Wohnen *****
*****

* Personen>1 Jahr im HH
gen over1=alter>1
egen pers=total(over1), by(hid syear)
gen pers3=pers
replace pers3=3 if pers>3
gen persi=pers-2
replace persi=0 if persi<0
* aphrf
replace phrf=. if phrf==0
egen phrfmean=mean(phrf), by(syear)
gen aphrf=phrf/phrfmean
replace phrf=0 if phrf==.
drop phrfmean
* welle
gen welle2=welle*welle
gen wepe=welle*persi
* ggk2
replace ggk=3 if ggk== -1 | ggk==.
recode ggk (1 2=1) (3=2) (4 5=3) (6 7=4), into(ggk2)

* Median-Regressionen separat für ggk2 und pers3
gen predsize=.
forvalues g=1/4 {
forvalues p=1/3 {
qreg hgsz welle welle2 persi wepe if ggk2==`g' & pers3==`p' [pw=aphrf]
predict predtemp
replace predsize=predtemp if ggk2==`g' & pers3==`p'
drop predtemp
}
}

* Log Grenzwerte
log using "$work/wohnflaeche.log", replace
table syear ggk2 pers if pers<7, c(min predsize) format(%5.1f)
log close

* Relative Wohnfläche
gen wgrosa=hgsz/predsize if hgsz>=0
recode wgrosa (0/0.66=1) (0.66/0.8=2) (0.8/1.2=3) (1.2/1.66=4) (1.66/max=5), into(wgrosakat)
* Anzahl Wohnräume-Anzahl HH-Personen
gen wgrosb=hgroom-pers
recode wgrosb (min/-1=1) (0=2) (1=3) (2=4) (3/max=5), into(wgrosbkat)
* Gesamtindikator Wohnen
label define wohn 1 "sehr beengt" 2 "beengt" 3 "durchschnittlich" 4 "geraeumig" 5 "sehr geraeumig"
gen wohnen:wohn=wgrosakat
* Korrektur um Anzahl Wohnräume
replace wohnen=2 if wgrosakat==1 & wgrosbkat==4
replace wohnen=3 if wgrosakat==1 & wgrosbkat==5
replace wohnen=3 if wgrosakat==2 & wgrosbkat==5
replace wohnen=4 if wgrosakat==5 & wgrosbkat==2
replace wohnen=3 if wgrosakat==5 & wgrosbkat==1
replace wohnen=3 if wgrosakat==4 & wgrosbkat==1
replace wohnen=wgrosbkat-1 if wohnen==. & wgrosbkat>2 /* 4 Personen in 3 HH */

*****
* Dimension Vermoegen *****
*****

```

```

for var hlc0093 hlc0096 hlc0104 hlc0105 hlc0106 hlc0107 \ new wkeine wka wbetrieb wbau wleben wwert: rename X
Y
* festverzinsliche Wertpapiere mit sonstigen zusammenfassen
replace wwert=1 if hlc0108==1
drop hlc0108

* Äquivalenzgewichtung Vermögen
for var i11104 i11105: gen Xeq=round(X/oeed)
* Gesamtbetrag (i11104+i11105, gewichtet)
egen i11145=rowtotal(i11104eq i11105eq)
recode i11145 (0=2) (1/1000=3) (1000/5000=4) (5000/max=5), into(vermoegen_kat)
* Vermoegensportfolio
recode wbetrieb wbau wleben wwert (-2=0) (-5=.)
egen vermn=rowtotal(wbetrieb wbau wleben wwert)
* Gesamtindikator Vermoegen
gen vermoegen=vermoegen_kat
replace vermoegen=vermoegen-1 if vermn==0
* erste Welle ersetzen durch zweite bei identischen Portfolio (Datenprobleme in der ersten Welle)
bysort pid (welle): replace vermoegen=vermoegen[_n+1] if pid==pid[_n+1] & welle==1 & welle[_n+1]==2 &
vermn==vermn[_n+1]

*****
*** kombinierte Lebenslagen: Vermögen, Wohnen, Arbeit
*****
* Scoring -2 -1 0 1 2 für Wohnen, Vermögen und Erwerbsarbeit
recode ewarbeit (6=3), into(lewarbeit)
gen llagen=(vermoegen-3)+(wohnen-3)+(lewarbeit-3)

* 5-stufige Lebenslagen
recode llagen (-6/-4=1) (-3/-2=2) (-1/1=3) (2/3=4) (4/6=5), into(llkat)

* Inkonsistenzen
gen llaginkon=(max(vermoegen,wohnen,lewarbeit)-min(vermoegen,wohnen,lewarbeit)>2)
drop lewarbeit

*****
*** Kombination Einkommens-Lebenslagen *****
*****

label define mlage 1 "Armut" 2 "Prekaritaet" 3 "untere Mitte" 4 "Mitte" 5 "Wohlstand" 6 "Wohlhabenheit" 7
"Inkonsistenz"
gen mlage:mlage=poskatinc*10+llkat
recode mlage (11 12 21=1) (13 22 31=2) (23 32=3) (24 33 42 34 43=4) (35 44 53=5) (45 54 55=6) (14 15 25 41 51
52=7)

* Inkonsistenzen
gen inkon=(poskatinc-llkat)

*****
*** Trends Querschnitte *****
*****

set line 255
log using "$work/Indikatoren.log", replace
foreach vari in poskatinc ewarbeit wohnen vermoegen llagen mlage {
tab syear `vari' [aw=phrf], nof row
}
log close

*****
***** Speichern Querschnittsdaten *****
*****

save "$work/Indikatoren", replace

```