Was Sie über Stata wissen sollten, wenn Sie einen Kurs der Reihe SOEPcampus besuchen wollen

- Unter der Kategorie **dringend erforderlich** listen wir Befehle auf, die mit denen Sie auf jeden Fall vertraut sein sollten, um im praktischen Teil der SOEPcampus Veranstaltung aktiv mitarbeiten zu können.
- Unter der Kategorie **erwünscht** finden Sie weitere Befehle, die Ihnen die Mitarbeit und das Verständnis erleichtern sollten, jedoch nicht dringend notwendig sind.
- Wenn Sie einzelne Befehle noch nicht kennen, machen Sie sich bitte im Vorfeld des Workshops mit den entsprechenden Befehlen vertraut.

Dringend erforderliche Stata Kenntnisse

Arbeit mit dem „Do-file Editor“

Sie sollten mit der Programmierung von do-Files so vertraut sein, dass Ihre komplette Datenanalyse sich durch die Ausführung („do“) von do-Files reproduzieren lässt.

- Wie starten Sie den Editor?
- Wie führen Sie das ganze Skript oder Teile davon aus?

Lokalisierung von Dateien auf dem Computer und im Netzwerk

Stata-dta-Dateien, die Sie im Windows Explorer finden, sollten Sie auch mit „use“ öffnen können.

```
use D:/temp/apausl, clear
use "\\hume\rdc-gen\apausl", clear
save D:/temp/apausl_test, replace
```

- Auf was müssen Sie achten, wenn Ordnernamen Leerzeichen oder Umlaute enthalten?
- Was ist der Unterschied zwischen / und \ im Pfadnamen?

Macros

Sie sollten mit der Verwendung von „locals“ und „globals“ vertraut sein.

```
global pfad "D:/temp/"
use "${pfad}apausl", clear

local uncpfad \\hume\rdc-gen\
use `uncpfad'apausl, clear
```

- Was ist der Unterschied von „locals“ und „globals“?
- Welche Tasten muss man drücken, um die Zeichen, von denen „locals“ umgeben werden, zu erzeugen?

If-conditions und logische operatoren in STATA

Sie sollten mit der Konditionierung von Befehlen über die if Erweiterung vertraut sein, und die logischen Operatoren in Stata kennen.

```
sum income if year == 2007 & age >= 45 & age <= 46 & health != .
tab xnetto xpop if (xnetto==10 | xnetto==12) & (xpop==1 | xpop==2)
```

- Wie kombiniert man AND und OR conditions logisch richtig?

list

```
list sex gebjahr todjahr if persnr==2102 | persnr==19202
```

- Was macht der list Befehl? Wann würden Sie den list Befehl benutzen?
- Mit welchen anderen Befehlen sollte der list Befehl stets benutzt werden?

keep/drop

Sie sollten wissen welchen Effekt die Befehle “keep” und “drop” haben.

```
keep if age >= 18 & age <=65
drop wp01742
```

rename

Sie sollten wissen wie man Variablen umbenennt.

```
rename yp10601 smoke2008
```

recode

Sie sollten wissen, wie man variablen umkodiert und Variablenlabels ändert.

```
recode age_3 (17/29=1) (30/64=2) (65/120=3)
label define age_3 1 "17-29" 2 "30-64" 3 "65+"
label values age_3 age_3
```

gen / egen / replace

Sie sollten neue Variablen mit hilfe von gen und egen generieren können

```
gen alt2003_15 = 1 if  2003-gebjahr == 15
```

- Was ist der Unterschied zwischen gen und egen? Wann wird welcher Befehl benutzt
- Was ist die Funktion des replace Befehls?

tab / tabstat / sum

Sie sollten die Befehle tab und tabstat kennen.

```
tabstat yp0101,  by(sex)
tab yp0101 sex
tab diff if year==2007
sum health if smokestop==1
```

- Was ist der Unterschied zwischen tab und tabstat und sum

merge

Sie sollten wissen, wie der merge Befehl funktioniert und wie die Syntax-Partikel 1:1 und m:1 zu verstehen sind.

```
merge 1:1 syear pid using "${MY_IN_PATH_long}\pgen.dta", keepus(pgemplst)
nogen
```

- Für welche Fälle enthält die beim mergen erzeugte Variable _merge eine 2?
- Was bedeutet die Optionen keep und nogen?
- Welchen Befehl sollte jemand sehr wahrscheinlich nutzen, wenn er merge m:n eingibt?

reshape

Sie sollten wissen was mit “long” und “wide” Format gemeint ist, und wie man Daten in das jeweils andere Dateiformat transferieren kann.

```
reshape long health smoke emplst hinc netto pop hhnrakt phrf pbleib ,
i(persnr) j(year)
```

- Welche Anforderungen an die Variablennamen haben die jeweiligen Transformationen?

Erwünschte Stata-Kenntnisse:

foreach/forvalues

Es ist hilfreich, wenn Sie bereits mit foreach / forvalues Loops gearbeitet haben.

```
local y=2006
    foreach w in w x y {
       rename `w'p0101 health`y'
       rename `w'phrf phrf`y'
       local y=`y'+1
      }
```

- Wann wird foreach benutzt, wann forvalues?
- Wie kann man das Loop-Interval bei forvalues bestimmen?

mvdecode

Es ist hilfreich wenn Sie diesen Befehl zur Umkodierung fehlender Werte kennen.

mvdecode `_all, mv(-1=. \ -2=.t \ -3=.x)`

Log-Files

Es ist hilfreich wenn Sie mit bereits mit log-files gearbeitet haben und die folgenden Befehle kennen.

```
capture log close
log using "${MY_LOG_FILE}", text replace
```

- Wo in ihrem Do-File sollte “capture log close” stehen.
- Wann würden Sie die Log-Files benutzen?