Hvordan jobbe med sysselsettingsdata fra A-ordningen (jobbdata)

1) Konvertere fra jobbdata til persondata – numeriske opplysninger

a) Lag to datasett:
   - Datasett A: Persondatasett
   - Datasett B: Jobbdatasett bestående av personid (ARBEIDSFORHOLD_PERSON) og jobbopplysninger (data om arbeidsforhold)

b) Datasett B aggregeres til personnivå via personid. Siden jobbdataene har numerisk format, kan du bare bruke kommandoen collapse(sum) eller collapse(mean) etter behov.

c) Datasett B kobles mot A via personid

d) Flytt deg over til datasett A med kommandoen «use»

e) Datasett A inneholder nå også jobbopplysninger som kan brukes til diverse statistikk og analyser knyttet til sysselsettingsforhold. Dette kan kombineres med øvrige persondata (demografi etc).

2) Konvertere fra jobbdata til persondata – kategoriske opplysninger

a) Lag et persondatasett: Datasett A

b) Lag et jobbdatasett bestående av personid (ARBEIDSFORHOLD_PERSON) og jobbopplysninger: Datasett B

i. La oss si du ønsker å kategorisere jobber basert på om de er heltids- eller deltidsjobber. Da må jobbdatasettet inneholde en variabel med slik informasjon.

ii. Selekter personer som tilhører den ene gruppen, f.eks. heltid. Bruk kommandoen «keep if»

iii. Datasettet aggregeres til personnivå via personid. Bruk kommandoen collapse(count), collapse(sum), eller collapse(mean) etter behov. Man kan bruke ulike typer teknikker for å aggregere kategoriske variabler vha. kommandoen collapse():

- Alfanumeriske variabler kan først gjøres numeriske vha. kommandoen «destring». Dette gjør det mulig å bruke variantene collapse(sum) og collapse(mean) når man skal aggregere. En fordel med dette er at man kan inkludere også andre jobbopplysninger som skal aggregeres på samme måte, i collapse()-uttrykket, f.eks. arbeidstid.

- En enklere og vel så grei metode er å først lage en ny variabel som tar verdien 1 dersom den kategoriske variabelen er lik den aktuelle gruppeverdien (heltid evt. deltid). Da kan man også bruke collapse(sum) eller collapse(mean) og inkludere andre jobbopplysninger som kan tas med i collapse-uttrykket.

- Om du bare er ute etter å summere hvor mange personer som befinner seg i hver kategori av en kategorisk jobbvariabel, er det enkleste å bare bruke kommandoen collapse(count). Denne varianten teller opp antallet

observasjoner per aggregert enhet (= person), og stiller ingen krav til verdiformat. Alle jobber i det selekterte jobbdatasettet (selektert med «keep if») vil få verdien 1 siden hver jobb forekommer en gang for det gitte måletidspunktet. Etter at jobbene er aggregert til personnivå, vil hver person som kan knyttes til en jobb i den aktuelle kategorien få verdien 1 eller større, etter hvor mange jobber man har hatt i den aktuelle kategorien. Personer som ikke kan knyttes til denne kategorien vil ikke få påkoblet en verdi, og tilegnes en missingverdi.

iv. Lag nye navn til de aggregerte variablene (som passer med det aggregerte innholdet)
v. Datasettet kobles mot datasett A via personid

c) Gjenta trinnene i punkt b) og lag et tredje jobbdatasett (datasett C) som består av de samme variablene. Men nå selekterer du på jobber tilhørende den andre kategorien (deltid). Merk at ved jobbing i skript trenger du bare kopiere linjene som lager datasett B, og justere de aktuelle linjene slik at det passer med den nye kategorien.
    - Trenger du å gruppere i flere enn to jobbkategorier, gjentas trinnene i punkt b) til du har laget, aggregert og koblet på tilsvarende antall datasett.

d) Flytt deg over til datasett A med kommandoen «use»

e) Datasett A inneholder nå også jobbopplysninger som kan brukes til diverse statistikk og analyser knyttet til sysselsettingsforhold, inkludert opplysninger knyttet til kategorier av jobber. Dette kan kombineres med øvrige persondata (demografi etc).

3) Hvordan koble persondata på jobbdata

a) Lag to datasett:
    - Datasett A: Jobbdatasett bestående av personid (ARBEIDSFORHOLD_PERSON) og jobbopplysninger (yrke, arbeidstid etc)
    - Datasett B: Persondatasett med personopplysninger (kjønn etc)
b) Datasett B kobles mot A via personid
c) Flytt deg over til datasett A med kommandoen «use»
d) Datasett A inneholder nå persondata i tillegg til jobbdata. Jobber som er knyttet til samme person vil ha de samme personopplysningene.

4) Hvordan sette sammen jobber over en lengre tidsperiode uten å miste jobber som kommer til underveis i måleperioden (bruk av «outer join»)

a) Opprett et datasett
b) Importer en jobbdata-variabel, f.eks. arbeidstid
c) Repeter trinn b) der du importerer samme opplysning for nyere måledatoer. Bruk nå import-opsjonen «outer_join». Dette sørger for at jobber som er nye i forhold til første måledato vil legges til datasettet ditt og at populasjonen din øker tilsvarende. Om du ikke bruker «outer_join», vil «left join» brukes som standard, og nye jobber til ikke legges til datasettet ditt. Da vil du bare få en populasjon bestående av jobber som var aktive ved første måledato.

Hvordan jobbe med data om pågående utdanning

1) Hvordan finne studenter ved et gitt tidspunkt

a) Lag to datasett:
   - Datasett A: Persondatasett med personopplysninger (kjønn etc)
   - Datasett B: Kursdatasett med personid (NUDB_KURS_FNR) og opplysning om kurstype (=studium) for et gitt tidspunkt (NUDB_KURS_NUS)

b) Datasett B aggregeres til personnivå via personid vha. kommandoen collapse(count)

c) Variabelen «kurstype» består nå av antall observasjoner for det aktuelle måletidspunkt. Bruk derfor kommandoen «rename» til å endre navn fra «kurstype» til «ant_kurs» evt. «ant_studier»

d) Datasett B kobles mot A via personid

e) Flytt deg over til datasett A med kommandoen «use»

f) Lag en dummyvariabel, «studerer», som er lik 1 dersom «ant_kurs» >= 1, og 0 ellers

g) Nå har du laget en dummyvariabel som kan brukes til å kategorisere personer som studerte ved et gitt tidspunkt

2) Hvordan finne personer som tok høyere studium i løpet av et lengre tidsspenn

a) Lag to datasett:
   - Datasett A: Persondatasett med personopplysninger (kjønn etc)
   - Datasett B: Kursdatasett på forløpsnivå (hendelsesnivå) som inneholder variabelen «kurstype» (NUDB_KURS_NUS). I stedet for kommandoen «import», brukes «import-event» til å importere alle observasjoner/hendelser for kurstype mellom to tidspunkter. Man angir da to tidspunkter med «to» mellom, i stedet for ett tidspunkt som man vanligvis gjør. F.eks. «import-event variabel1 2019-01-01 to 2019-12-31 as variabelforløp». Merk at det bare er tillatt å importere én «event»-variabel for hvert datasett.

b) Konverter «kurstype» til numerisk format vha. kommandoen «destring»

c) Selekter personer som tok høyere studium («kurstype» = 700000 -> 900000). Bruk kommandoen «keep if». Om du vil finne alle som studerte, kan trinn b) og c) droppes.

d) Lag et kursdatasett som inneholder personid (NUDB_KURS_FNR) (bruk vanlig import): Datasett C

e) Datasett C kobles mot datasett B. Datasett B inneholder nå en lenke til person (= personid)

f) Flytt deg over til datasett B med kommandoen «use»

g) Datasett B aggregeres til personnivå via personid vha. kommandoen collapse(count)

h) Som i 1) består nå variabelen «kurstype» av antall observasjoner for den aktuelle måleperioden. Bruk derfor kommandoen «rename» til å endre navn fra «kurstype» til «ant_kurs» evt. «ant_studier»

i) Datasett B kobles mot A via personid

j) Flytt deg over til datasett A med kommandoen «use»

k) Gjør som i 1) og lag en dummyvariabel, «studium_høy» (evt. «studerer» dersom du ikke har selektert på høyere studium i trinn b) og c)), som er lik 1 dersom «ant_kurs» >= 1, og 0 ellers.
l) Nå har du laget en dummyvariabel som kan brukes til å kategorisere personer som studerte (på et høyere studium) i løpet av en lengre tidsperiode

Vedlegg A: Skriptsyntax for jobbdata

```
textblock
Temakurs: Hvordan jobbe med jobbdata
-----------------------------------
endblock

require no.ssb.fdb:15 as db

textblock
1) Hvordan konvertere data fra jobb- til personnivå - numeriske opplysninger
-------------------------------------------------------------------------
endblock

//Lager datasett for sysselsatte (persondata)
create-dataset sysselsatte
import db/ARBLONN_PERS_KJOENN 2021-07-16 as kjønn
import db/ARBLONN_PERS_ALDER 2021-07-16 as alder

//Lager datasett for arbeidsforhold (jobbdata)
create-dataset arbeidsforhold
import db/ARBLONN_ARB_ARBEIDSTID 2021-07-16 as arbtid
import db/ARBLONN_ARB_STILLINGSPST 2021-07-16 as stillingspst
import db/ARBEIDSFORHOLD_PERSON as personid

//Aggregerer fra jobbdata- til persondatanivå ved å summere arbeidstid og stillingsprosent per person. Kobler deretter jobbopplysningene på persondatasettet sysselsatte
collapse (sum) arbtid stillingspst, by(personid)
merge arbtid stillingspst into sysselsatte

//Lager jobbstatistikk
use sysselsatte
```

```
summarize arbtid stillingspst
tabulate kjønn, summarize(arbtid)
tabulate kjønn, summarize(stillingspst)

generate aldersgr = 1
replace aldersgr = 2 if alder > 25
replace aldersgr = 3 if alder > 40
replace aldersgr = 4 if alder > 60
define-labels alderslabel 1 '0-25' 2 '26-40' 3 '41-60' 4 '61->'
assign-labels aldersgr alderslabel
tabulate aldersgr kjønn, summarize(arbtid)
tabulate aldersgr kjønn, summarize(stillingspst)



textblock
2) Hvordan konvertere data fra jobb- til personnivå - kategoriske opplysninger
--------------------------------------------------------------------------------
endblock

create-dataset vestland
import db/ARBLONN_PERS_KOMMNR 2021-07-31 as bosted
keep if substr(bosted,1,2) == '46'
import db/ARBLONN_PERS_KJOENN 2021-07-16 as kjønn

create-dataset heltid
import db/ARBLONN_ARB_H3LDELTID 2021-07-16 as heldel
import db/ARBLONN_ARB_ARBEIDSTID 2021-07-16 as arbtid
import db/ARBEIDSFORHOLD_PERSON as personid
tabulate heldel
keep if heldel == '1'
destring heldel
collapse(sum) heldel arbtid, by(personid)
rename heldel hel
```

```
rename arbtid arbtid_hel
merge hel arbtid_hel into vestland

create-dataset deltid
import db/ARBLONN_ARB_H3LDELTID 2021-07-16 as heldel
import db/ARBLONN_ARB_ARBEIDSTID 2021-07-16 as arbtid
import db/ARBEIDSFORHOLD_PERSON as personid
keep if heldel == '2'
destring heldel
replace heldel = heldel/2
collapse(sum) heldel arbtid, by(personid)
rename heldel del
rename arbtid arbtid_del
merge del arbtid_del into vestland

use vestland
tabulate hel
tabulate del
tabulate del kjønn, rowpct freq
tabulate del, summarize(arbtid_del)
tabulate kjønn, summarize(arbtid_del)
tabulate kjønn, summarize(arbtid_hel)


textblock
3) Case om turnusordninger
-------------------------------------------------
endblock

//Lager datasett for sysselsatte (persondata)
create-dataset sysselsatte2
import db/ARBLONN_PERS_KJOENN 2021-07-16 as kjønn
import db/ARBLONN_PERS_ALDER 2021-07-16 as alder
```

```
//Lager datasett for arbeidsforhold (jobbdata)

create-dataset arbeidsforhold2

import db/ARBLONN_ARB_TID_ORDNING 2021-07-16 as arb_ordning

import db/ARBLONN_ARB_ARBEIDSTID 2021-07-16 as arbtid

import db/ARBEIDSFORHOLD_PERSON as personid


tabulate arb_ordning


//Aggregerer fra jobbdata- til persondatanivå ved å summere arbeidstid og stillingsprosent per person. Kobler deretter jobbopplysningene på persondatasettet sysselsatte

keep if arb_ordning == 'dogn355'

generate ant_turnus = 1

collapse (sum) ant_turnus arbtid, by(personid)

merge ant_turnus arbtid into sysselsatte2


//Lager turnus-statistikk

use sysselsatte2

tabulate ant_turnus, summarize(arbtid) mean freq


generate turnus = 0

replace turnus = 1 if ant_turnus >= 1


summarize arbtid if turnus


tabulate turnus, cellpct freq

tabulate kjønn turnus, rowpct freq

tabulate kjønn if turnus, summarize(arbtid)


generate aldersgr = 1

replace aldersgr = 2 if alder > 25

replace aldersgr = 3 if alder > 40

replace aldersgr = 4 if alder > 60
```

```
assign-labels aldersgr alderslabel

tabulate aldersgr turnus, rowpct freq
tabulate aldersgr if turnus, summarize(arbtid)


textblock
4) Data på jobbnivå: Hvordan koble sammen persondata med jobbdata
----------------------------------------------------------------------
endblock

create-dataset jobber
import db/ARBLONN_ARB_YRKE_STYRK08 2021-07-16 as yrke
import db/ARBLONN_ARB_ARBEIDSTID 2021-07-16 as arbeidstid
import db/ARBLONN_LONN_EKV_IALT 2021-06-30 as mndlønn_heltidsekv
import db/ARBEIDSFORHOLD_PERSON as personid

create-dataset personer
import db/ARBLONN_PERS_KJOENN 2021-07-16 as kjønn
merge kjønn into jobber on personid

use jobber
tabulate yrke kjønn, summarize(mndlønn_heltidsekv)
tabulate yrke kjønn, summarize(arbeidstid)

textblock
5) Case med outer join for å få med personer som kommer i jobb utover observasjonsperioden

---------------------------------------------------------------------------------------
endblock

create-dataset jobber_alle
import db/ARBLONN_ARB_ARBEIDSTID 2021-01-16 as arbtid2101
```

```
import db/ARBLONN_ARB_ARBEIDSTID 2021-02-16 as arbtid2102, outer_join
import db/ARBLONN_ARB_ARBEIDSTID 2021-03-16 as arbtid2103, outer_join

summarize
```

Vedlegg B: Skriptsyntax for data om pågående utdanning

```
textblock
Temakurs: Hvordan jobbe med data om pågående utdanning (hvordan finne studenter)
--------------------------------------------------------------------------------
endblock

require no.ssb.fdb:15 as db

textblock
1) Finne personer som studerte på et gitt tidspunkt
--------------------------------------------------
endblock

//Oppretter persondatasett for bosatte per 2019-01-01
create-dataset bosatte
import db/BEFOLKNING_KJOENN as kjønn
import db/BEFOLKNING_STATUSKODE 2019-01-01 as regstatus
keep if regstatus == '1'

//Henter personer som tar utdanning per 1. november 2019, og kobler dette på persondatasettet. Siden kursdata kan ha flere observasjoner per individ, må kommandoen collapse brukes til å aggregere opp til personnivå. Vi bruker count som aggregeringsverdi (antall records)
create-dataset kursdata
import db/NUDB_KURS_NUS 2019-11-01 as kurstype
import db/NUDB_KURS_FNR as fnr
collapse (count) kurstype, by(fnr)
rename kurstype ant_kurs
merge ant_kurs into bosatte

//Lager tabell over antall personer som studerer per 1. november 2019
use bosatte
```

```
generate studerer = 0

replace studerer = 1 if ant_kurs >= 1

tabulate studerer kjønn



textblock

2) Finne personer som tok høyere studium i løpet av en lengre tidsperiode (et år)

------------------------------------------------------------------------

endblock


//Oppretter persondatasett for bosatte per 2019-01-01

create-dataset bosatte2

import db/BEFOLKNING_KJOENN as kjønn

import db/BEFOLKNING_STATUSKODE 2019-01-01 as regstatus

keep if regstatus == '1'


//Henter personer som tar høyere utdanning i løpet av 2019

create-dataset kursdata2

import-event db/NUDB_KURS_NUS 2019-01-01 to 2019-12-31 as kurstype

destring kurstype

keep if kurstype >= 700000 & kurstype < 900000


//Kobler på lenke mellom kurs-id og fødselsnumre

create-dataset lenke_kurs_person

import db/NUDB_KURS_FNR as fnr

merge fnr into kursdata2


//Lager statistikk (collapser) over antall hendelser med høy utdanning per individ, og kobler dette på persondatasettet

use kursdata2

collapse (count) kurstype, by(fnr)
```

```
rename kurstype ant_kurs

merge ant_kurs into bosatte2


//Lager tabell over antall personer som tok høy utdanning i løpet av 2019

use bosatte2

generate utdanning_høy = 0

replace utdanning_høy = 1 if ant_kurs >= 1

tabulate utdanning_høy kjønn
```