NaSTaB
National Survey of Tax and Benefit

재 정 패 널 조 사

NaSTaB [National Survey of Tax and Benefit]

데이터 가공법 소개

2016. 8.

KIPF 한국조세재정연구원

목 차

※본 코드 사용 시 D 드라이브에 데이터 저장 후 사용 권고

1. STATA

1.1. STATA 실행

```
set    memory 1200m                                  /* 메모리    설정*/
set    maxvar 32767
set    more off                                      /* 결과    설정*/

q                                                    /* 설정    확인*/

cd    "D:₩NaSTaB"                                    /* 디렉토리 확인*/
pwd                                                  /* 디렉토리 체크*/

use    "NaSTaB08H_s.dta", clear
       /*"D:₩NaSTaB₩NaSTaB08H_s.dta" 사용 및, 메모리    정리*/

save    "NaSTaB08H_s_test1.dta"                      /* NaSTaB08H_s_test1.dta로 저장*/
```

1.2. STATA의 기본 연산자

산술 연산자: +(add), -(sub), /(div), *(mul), ^(power)

조건 연산자: & (and), |(or), !(not)

관계 연산자: ==(equal), !=(not equal), < (less than), > (greater than)

<==(< or equql), >==(>or equal)

조건문: 실행 명령어 if + 조건, 실행 명령어 in + 범위

Help: 실행 명령어

Run: ctrl + d(전체 실행, 부분 실행)

Clear : 메모리 정리

1.3. 가구주정보추출 (gen, replace, label)

```
generate h08hgen=.                                   /* 변수 생성*/


replace h08hgen=w08gen01 if w08rel01==1              /* 변수 값 변환*/
replace h08hgen=w08gen02 if w08rel02==1
replace h08hgen=w08gen03 if w08rel03==1
replace h08hgen=w08gen04 if w08rel04==1
replace h08hgen=w08gen05 if w08rel05==1
replace h08hgen=w08gen06 if w08rel06==1
replace h08hgen=w08gen07 if w08rel07==1
replace h08hgen=w08gen08 if w08rel08==1
replace h08hgen=w08gen09 if w08rel09==1                                  (계속)
```

```
label variable h08hgen "[생성변수] 가구주    성별"    /* 변수 서식 설정*/


label define hgen 1 "남성" 2    "여성"                 /* 변수 값 서식 생성*/

label values h08hgen hgen                              /* 변수 값 서식 적용*/

tabulate h08hgen                                       /* one-way 기초 통계표 생성*/
tab h08hgen   hs08b10                                  /* two-way 통계표 생성*/
```

1.4. 합산변수생성(egen, sum, tabstat)

```
mvdecode _all, mv(-9)
                                          /* Numeric values 를 missing values 로 변환*/

egen exp_food=rowtotal(h08cc002 h08cc004)       /* 식료품 구입비+외식비*/

label var exp_food "[생성변수] 식비"

summarize exp_food, d                            /* exp_food 변수 합산*/

tabstat exp_food, stat(mean) by(h08hgen)
                                        /* Compact table of summary statistics*/
```

1.5. 변수 정리

```
keep hid08 h08hgen                               /* keep variables*/


drop if h08hgen==2                               /* Drop observations*/
       /*    keep if h08gen==1
rename h08hgen male                              /* 변수 이름 변경*/

format %10.0g hid08                              /* 변수들의 아웃풋 포맷 설정*/


/*destring : Convert string variables to numeric variables and    vice versa
  tostring : Convert numeric    variables to string variables*/
```

1.6. 기초노령연금수급자추출

```
use "D:₩NaSTaB₩NaSTaB08P_s.dta", clear

gen pba=.                                        /* pba 라는 변수 생성*/
keep pid08 p08ba03*   pba

foreach var1 in p08ba030 p08ba033 p08ba036{
       replace pba=1 if `var1'==11
       }
       keep if pba==1
```

1.7. 사회보장급여액 추출

```
use "D:₩NaSTaB₩NaSTaB08P_s.dta", clear
gen pba=.

keep pid08 p08ba03*      pba

replace pba= p08ba032 if p08ba030==11
replace pba= p08ba035 if p08ba033==11
replace pba= p08ba038 if p08ba036==11

keep if pba>0 & pba!=.
```

1.8. 데이터구조변환- long to wide

```
use "NaSTaB08P_s.dta", clear

keep hid08 ps08a04  p08bb002                    /* 가구 id, 가구원 번호, 근로소득-*/

reshape wide p08bb002, i(hid08) j(ps08a04)      /* i(기준변수) j(구분자 변수)*/

egen p08pinc=rowtotal(p08bb0021 – p08bb0026)

label var p08pinc "[생성변수] 가구원 근로소득 합"

keep hid08 p08pinc

save "NaSTaB08P_s_test3.dta", replace
```

1.9. 데이터구조변환- wide to long

```
use "NaSTaB08H_s.dta", clear

keep hid08 hpid01-hpid09 w08byr0*           /* 가구 id, 가구원 id, 출생년도*/

reshape long hpid0 w08byr0, i(hid08) j(n)    /* i(기준변수) j(Number of variables)*/

rename hpid0 pid08
drop if mi(pid08)                           /* Missing values */

save "NaSTaB08H_s_test3.dta", replace
```

1.10. 데이터 구조 변환 응용 - 사회보장급여수급 정보 추출

```
use    "D:₩NaSTaB₩NaSTaB08P_s.dta", clear

keep pid08 p08ba03*

local k=11
```

(계속)

```
       foreach var in p08ba030 p08ba033 p08ba036  {
         rename    `var' ba`k'
         local    ++k
         }

local k=21
       foreach var in p08ba032 p08ba035 p08ba038  {
         rename    `var' ba`k'
         local    ++k
         }

local k=31
         foreach    var in p08ba031 p08ba034 p08ba037  {
           rename `var' ba`k'
           local ++k
           }

           reshape long ba1 ba2 ba3 ,i(pid08) j(n)

         label    var ba1 "정부지원 현금 종류"
         label    var ba3 "정부지원 현금 수급개월수"
         label    var ba2 "정부지원 현금 연간수급총액(만원)"

         keep    if ba1==11
```

1.11. 데이터 병합 - merge

```
use "NaSTaB08P_s.dta", clear

isid pid08                                          /* unique identifiers 여부 확인*/

merge 1:1 pid08 using    "D:₩NaSTaB₩NaSTaB08H_s_test3.dta"

                                                    /* 1:1, 1:m, m:1, m:m*/

tab _merge
```

1.12. 데이터 병합 - append

```
use "NaSTaB07H_s.dta", clear

keep hid07 h07ba001 h07ba002 h07ba003
rename hid07 hid
rename h07ba001 ba001
rename h07ba002 ba002
rename h07ba003 ba003

save    "D:₩NaSTaB₩NaSTaB07H_test4.dta", replace


use "NaSTaB08H_s.dta", clear

keep hid08 h08ba001 h08ba002 h08ba003
```

(계속)

```
rename hid08 hid
rename h08ba001 ba001
rename h08ba002 ba002
rename h08ba003 ba003

save "NaSTaB08H_test4.dta",    replace

use    "NaSTaB07H_test4.dta",clear
append using "NaSTaB08H_test4.dta",    gen (t)        /* 데이터 append */
tab t
recode t (0=2013) (1=2014)                            /* 분류변수 기록 */

save "NaSTaBH_panel.dta",    replace
```

1.13. 패널 데이터 setting

```
use "NaSTaBH_panel.dta",    clear

xtset hid t                              /* panel data 로 명명, xtset panelvar timevar*/

xtset, clear                             /* xt 셋팅 초기화*/
```

2. SPSS

2.1. SPSS 실행

```
get file    "D:₩NaSTaB₩NaSTaB08H_s.sav".          /*데이터 불러오기.
dataset name test1 window = front.
save outfile    "d:₩NaSTaB₩NaSTaB08H_s_test1.sav".
```

2.2. SPSS 기본 연산자

```
산술 연산자:    +(add), -(sub), /(div), *(mul), **(exponentiation)
조건 연산자:    and, or, not
관계    연산자: =(equal), NE (not equal), LT (less than), GT (greater    than)
                LE(<=, < or equal), GE(>=, >    or equal)
Missing    : sysmis(var), missing(var)
```

2.3. 가구주정보추출 (compute, if, label)

```
compute h08hgen=0. /* Create of    variable.

if (w08rel01=1) h08hgen=w08gen01.          /* if (조건식) newvar=값.
if (w08rel02=1) h08hgen=w08gen02.
if (w08rel03=1) h08hgen=w08gen03.
if (w08rel04=1) h08hgen=w08gen04.
if (w08rel05=1) h08hgen=w08gen05.
if (w08rel06=1) h08hgen=w08gen06.
if (w08rel07=1) h08hgen=w08gen07.
if (w08rel08=1) h08hgen=w08gen08.
if (w08rel09=1) h08hgen=w08gen09.

var lab h08hgen [생성변수] 가구주성별.      /* variable label.

value lables h08hgen 1 '남성' 2 '여성'.

                                          /* Value label 정의, value label 을 할당.

freq h08hgen.                             /* one-way tables of summary statistics.

crosstabs h08hgen by hs08b10.             /*t wo-way tables(시도코드).
```

2.4. 합산변수 생성 (egen, sum, tabstat)

```
missing values hid08 to H08LWT (-9).

/* Change numeric values to missing  values. missing values all (-9).

compute exp_food=sum(h08cc002,    h08cc004). /*식료품 구입비+외식비.

variable labels exp_food [생성변수] 식비.

descriptives exp_food.                          /* des exp_food.

means exp_food by h08hgen
/cells mean.                                    /* mean count    stddev.

save outfile    "d:₩NaSTaB₩NaSTaB08H_s_test1.sav".
```

2.5. 변수 정리

```
del var hid08b to version exp_food.             /* Drop variables.

display name.
select if h08hgen=1.                            /* Drop observations.

rename var (h08hgen=male).                      /* Rename variable.

formats hid08(f10.0).                           /* Set variables' output format.

/* autorecode version/ into v.
/* alter type v(f1.0).
```

2.6. 기초노령연금 수급자 추출

```
get file    "D:₩NaSTaB₩NaSTaB08P_s.sav"
/keep pid08 p08ba030 to p08ba038.           /* p08ba030 부터 p08ba038 변수 선택.

dataset name test2_4 window = front.

do repeat
var1= p08ba030 p08ba033 p08ba036.           /* repeat var1.
if (var1=11) pba=1.
end repeat.

select if pba=1.

fre pba.

save outfile    "d:₩NaSTaB₩NaSTaB08H_s_test2_4.sav".
```

2.7. 사회보장급여수급액 추출

```
get file    "D:₩NaSTaB₩NaSTaB08P_s.sav"
/keep pid08 p08ba030 to p08ba038.
dataset name test2_5 window = front.

if (p08ba030=11) pba= p08ba032.
if (p08ba033=11) pba= p08ba035.
if (p08ba036=11) pba= p08ba038.

select if not sysmis(pba).                                  /* select if pba gt 0 .

save outfile    "d:₩NaSTaB₩NaSTaB08H_s_test2_5.sav".

dataset close test1.
dataset close test2_4.
```

2.8. 데이터 구조변환 - cases to vars (wide)

```
get file    "d:₩NaSTaB₩NaSTaB08P_s.sav"
/keep hid08 ps08a04 p08bb002.                    /* 가구 id, 가구원 번호, 근로소득.
dataset name test3_1 window = front.

casestovars
     /id=hid08                                   /* 기준    변수.
     /index=ps08a04.                             /* 구분자 변수.   "/count=num."

compute p08pinc=sum(p08bb002.1 to    p08bb002.6).

var lab p08pinc 가구원 총 소득 합.

des p08pinc.

dele var p08bb002.1 to p08bb002.6.

save outfile      "d:₩NaSTaB₩NaSTaB08P_s_test3.sav".
```

2.9. 데이터 구조변환 - vars to cases (long)

```
get file  "d:₩NaSTaB₩NaSTaB08H_s.sav".
dataset name test3_2 window = front.

varstocases
/make w08byr from w08byr01    w08byr02        w08byr03        w08byr04
w08byr05        w08byr06        w08byr07        w08byr08        w08byr09
/* 출생년도

/make pid08 from hpid01        hpid02  hpid03  hpid04  hpid05  hpid06  hpid07
hpid08  hpid09
/* 가구원 id
```

(계속)

```
/keep hid08
/* 가구 id

/null keep.

select if pid08>0. /* missing    values.

sort cases by pid08.

save outfile      "d:₩NaSTaB₩NaSTaB08H_s_test3.sav".
```

2.10. 데이터 구조 변환 응용 - vars to cases (long)

```
get file    "d:₩NaSTaB₩NaSTaB08P_s.sav".
dataset name test3_3 window = front.

varstocases
/make ba1 from p08ba030 p08ba033    p08ba036
/make ba2 from p08ba032 p08ba035    p08ba038
/make ba3 from p08ba031 p08ba034    p08ba037
/keep pid08
/null keep.

select if ba1=11.

des ba1 to ba3.

save outfile      "d:₩NaSTaB₩NaSTaB08P_s_test3_3.sav".
```

2.11. 데이터 병합 - merge

```
get file    "d:₩NaSTaB₩NaSTaB08P_s.sav".
dataset name test4_1 window = front.

sort cases by pid08.

* Check for unique identifiers.
MATCH FILES
     /FILE=*
     /BY pid08
     /FIRST=PrimaryFirst
     /LAST=PrimaryLast.
DO IF (PrimaryFirst).
COMPUTE      MatchSequence=1-PrimaryLast.
ELSE.
COMPUTE      MatchSequence=MatchSequence+1.
END IF.
LEAVE      MatchSequence.
FORMATS      MatchSequence (f7).
```

(계속)

```
COMPUTE      InDupGrp=MatchSequence>0.
SORT CASES InDupGrp(D).
MATCH FILES
     /FILE=*
     /DROP=PrimaryFirst InDupGrp MatchSequence.
VARIABLE LABELS   PrimaryLast '마지막 일치하는 각 케이스를 기본으로 나타내는 표시자'.
VALUE LABELS  PrimaryLast 0 '중복 케이스' 1 '기본 케이스'.
VARIABLE LEVEL  PrimaryLast (ORDINAL).
FREQUENCIES VARIABLES=PrimaryLast.
EXECUTE.

dataset activate test4_1.
match files /table=* /* 활성화 파일에 기준표 있음(table 파일은 중복없어야함).
     /file=test3_2
     /by pid08. /*기준변수.

des w08byr.
```

2.12. 데이터 병합 - append

```
get    file="d:₩NaSTaB₩NaSTaB07H_s.sav"
/keep=hid07 h07ba001
/rename = (hid07 h07ba001 = hid ba001    ).
dataset name test4_2_1  window = front.

compute time=2013.

save outfile    ="D:₩NaSTaB₩NaSTaB07H_test4.sav".

get    file="d:₩NaSTaB₩NaSTaB08H_s.sav"
/keep=hid08 h08ba001
/rename= (hid08 h08ba001 = hid ba001).
dataset name test4_2_2  window = front.

compute time=2014.

save outfile =    "D:₩NaSTaB₩NaSTaB08H_test4.sav".

dataset active test4_2_1.

add files /file=*
                /file=test4_2_2.
fre time.

save outfile    ="d:₩NaSTaB₩NaSTaBH_panel.sav".
```

※ SAS 코드 사용시 STATA 데이터 파일 사용 권고

3. SAS

3.1. SAS 실행

```
Libname nastab "D:₩nastab";                          /*라이브러리 setting*/

Proc import OUT=    NASTAB.NaSTaB08H_s               /*STATA 데이터 파일 불러오기*/
     DATAFILE= "D:₩NaSTaB₩NaSTaB08H_s.dta"
     DBMS=STATA REPLACE;
     run;
```

3.2. SAS의 기본 연산자

- 산술 연산자: +(add), -(sub), /(div), *(mul), ^(power)

- 조건 연산자: & (and), !(not), ~(not)

- 관계 연산자: =(equal), ^=(not equal), < (less than), > (greater than)

  <=(< or equql), >=(>or equal)

- 조건문: if 조건 + 실행 명령어

3.3. 가구주 정보 추출 (Data, if, Proc format)

```
Data nastab..NaSTaB08H_s_test1;
  set nastab.NaSTaB08H_s;
  h08hgen=. ;                                        /*'h08hgen' 라는 새로운 변수 생성*/

  if  w08rel01=1 then   h08hgen=w08gen01;
  if  w08rel02=1 then h08hgen=w08gen02;
  if  w08rel03=1 then h08hgen=w08gen03;
  if  w08rel04=1 then h08hgen=w08gen04;
  if  w08rel05=1 then h08hgen=w08gen05;
  if  w08rel06=1 then h08hgen=w08gen06;
  if  w08rel07=1 then h08hgen=w08gen07;
  if  w08rel08=1 then h08hgen=w08gen08;
  if  w08rel09=1 then h08hgen=w08gen09;

  label h08hgen='[generated variable] householders sex';
  run;

Proc format;
  value h08hgen 1="men"   2="women";
  run;

Data nastab.NaSTaB08H_s_test1;
  set nastab.NaSTaB08H_s_test1;
  format h08hgen h08hgen.;
  run;
```

(계속)

```
Proc freq data=nastab.NaSTaB08H_s_test1;    /*one-way tables of summary statistics*/
  tables h08hgen;
  run;

Proc freq data=nastab.NaSTaB08H_s_test1;    /*two-way tables of summary statistics*/
  tables h08hgen hs08b10;
  run;
```

3.4. 합산 변수 생성 (label, Proc means, Proc sort)

```
Data nastab.NaSTaB08H_s_test1;
  set nastab.NaSTaB08H_s_test1;
  exp_food=sum(of h08cc002 h08cc004);    /*식료품비+ 외식비*/

  label exp_food="[generated variable]food expenses";
  run;

Proc means;
  var    exp_food;
  run;

Proc sort;                                    /*Compact    table of summary statistics*/
  by   h08hgen;
  run;

Proc means mean;                              /*mean 만 포함*/
  var exp_food;
  by h08hgen;
  run;
```

3.5. 변수 정리 - keep, drop

```
Data nastab.NaSTaB08H_s_test1;
 set nastab.NaSTaB08H_s_test1;

 keep hid08 h08hgen;                          /*keep variables*/
 if h08hgen=2 then delete;                    /*drop observatio and keep if h08gen=1*/

 rename h08hgen=male;                         /*Rename variable*/

 format hid08 best10.;                        /*Set variables' output format*/
 run;
```

3.6. 기초노령연금 수급자 추출

```
Proc import OUT=    NASTAB.NaSTaB08P_s        /*STATA 데이터 불러오기*/
  DATAFILE= "D:₩NaSTaB₩NaSTaB08P_s.dta"
  DBMS=STATA REPLACE;
  run;

Data nastab.NaStaB08P_s_test1;                /*NaStaB08P_s_test1 로 저장*/
  set nastab.NaStaB08P_s;
```

(계속)

```
    pba=.;
    keep pid08 p08ba030-p08ba038 pba;

      array var{3} p08ba030 p08ba033 p08ba036;
      do i=1 to 3;
      if var[i]=11 then pba=1;
      end;

     if pba=1;
     run;
```

3.7. 사회보장급여액 추출

```
Data nastab.NaStaB08P_s_test2;                    /*NaStaB08P_s_test2 로 데이터 저장*/
    set nastab.NaStaB08P_s;
    keep pid08 p08ba030-p08ba038 pba;
    if p08ba030=11 then pba=p08ba032;
    if p08ba033=11 then pba=p08ba035;
    if p08ba036=11 then pba=p08ba038;

    if pba>0 & pba^=0;
    run;
```

3.8. 데이터 구조 변환 - wide transpose

```
Data nastab.NaSTaB08P_s_test3;
    set nastab.NaSTaB08P_s;
    keep hid08 ps08a04   p08bb002;
    run;


Proc transpose data=nastab.NaSTaB08P_s_test3
    prefix=ps08a04 out=nastab.NaSTaB08P_s_test3;   /*데이터 생성 –NaSTaB08P_s_test3*/
    by hid08;
    var p08bb002;
    id ps08a04 ;
    run;

Data nastab.NaSTaB08P_s_test3;
   set nastab.NaSTaB08P_s_test3;
   p08pinc=sum( of ps08a040001 - ps08a040006);
   label p08pinc="[generated variable] sum of family members'     income";
   keep hid08 p08pinc;
   run;
```

3.9. 데이터 구조 변환 - long transpose

```
Data nastab.NaSTaB08H_s_test3;
    set nastab.NaSTaB08H_s;
    keep hid08 hpid01-hpid09 w08byr01-w08byr09;
    run;

Proc transpose data=nastab.NaSTaB08H_s_test3
    out=nastab.NaSTaB08H_s_longh prefix=hpid;
                                        /*variables (hpid01-hpid09) were transposed*/
    by hid08;
    var hpid01-hpid09;
    run;

Proc transpose data=nastab.NaSTaB08H_s_test3
    out=nastab.NaSTaB08H_s_longw    prefix=w08byr;
    by hid08;
    var w08byr01-w08byr09;
    run;

Data nastab.NaSTaB08H_s_test3;
    merge nastab.NaSTaB08H_s_longh (rename=(hpid1=pid08) drop=_name_)
nastab.NaSTaB08H_s_longw (rename=(w08byr1=w08byr));
    by hid08;
    hmember_number=input(substr(_name_, 7), 5.);
    label hmember_number="family member number";

    drop _name_;

    if pid08=. then delete;

    run;
```

3.10. 데이터 병합 - merge

```
Data nastab.merged;
  merge nastab.NaSTaB08P_s nastab.NaSTaB08H_s_test3;
  by hid08;
  run;
```

3.11. 데이터 병합 - append

```
Proc import OUT= nastab.NaSTaB07H_s
  DATAFILE="D:₩NaSTaB₩NaSTaB07H_s.dta"
  DBMS=STATA REPLACE;
  run;

Data nastab.NaSTaB07H_test4;
  set nastab.NaSTaB07H_s;
  keep hid07 h07ba001 h07ba002 h07ba003 t;
  t=2013;
```

(계속)

```
  rename hid07=hid;
  rename h07ba001=ba001;
  rename h07ba002=ba002;
  rename h07ba003=ba003;
  run;

Data nastab.NaSTaB08H_test4;
  set nastab.NaSTaB08H_s;
  t=2014;
  keep hid08 h08ba001 h08ba002 h08ba003 t;

  rename hid08=hid;
  rename h08ba001=ba001;
  rename h08ba002=ba002;
  rename h08ba003=ba003;
  run;

Data nastab.panel;
  set nastab.NaSTaB07H_test4 nastab.NaSTaB08H_test4;
  run;
```