Suggested Solutions/Code for Stata Exercises
AERC Technical Workshop
Nairobi, 21-26 May 2001

In the following you will find a set of Stata commands that complete the exercises proposed in the Introduction to Stata sessions. Note that these are just suggested commands since there is no one way to do these exercises:

Introduction to Stata:

p. 19: Using the VNLSS93 household roster located in

```
c:\aerc\stata_training\origdata\sect01a
```

create a household-level dataset with the following information

1. Number of adult household members (15+ years)
2. Average age of household members
3. Gender of the household head

```
use c:\aerc\stata_training\origdata\sect01a;
 gen int adult = (agey>=15);
 gen headsex = sex if rel == 1;
collapse (sum) adult (mean) agey (min) headsex, by(hid);
```

p. 32: Using the VNLSS93 household roster located in

```
c:\aerc\stata_training\origdata\sect01a
```

create a variable for the most senior male in the household, where senior male is defined as

1) HH head if HH head is male;
2) spouse of HH head if HH head is female;
3) oldest male if neither 1) or 2).

```
use c:\aerc\stata_training\origdata\sect01a;
 gen int  headml   = pid if rel == 1 & sex == 1;
 gen int  spouseml = pid if rel == 2 & sex == 1;
 gen int  headfm   = pid if rel == 1 & sex == 2;
 gen int  spousefm = pid if rel == 2 & sex == 2;
 gsort hid  sex -agey;
  quietly by hid: gen int oldestml = pid[1];
 gsort hid -sex -agey;
  quietly by hid: gen int oldestfm = pid[1];
collapse (min) headml-oldestfm, by(hid sex);
  replace oldestml = . if sex == 2;
  replace oldestfm = . if sex == 1;
collapse (min) headml-oldestfm, by(hid);

for any ml fm: gen int senidX = headX
             \ replace senidX = spouseX if senidX == .
             \ replace senidX = oldestX if senidX == .;

keep hid senidml senidfm;
sort hid;
save c:\temp\temp, replace;

use c:\aerc\stata_training\origdata\sect01a;
sort  hid;
merge hid using c:\temp\temp;
tab _m;
for any ml fm: gen int seniorX = (senidX==pid);
```

This actually creates variables that identify the most senior male and female. Note that `pid[1]` refers to the individual id of the first observation within the household (`hid`). This is the way I would have done it, but since we haven't seen this syntax you obviously couldn't be expected to do it this way. Let's redo

```
gsort hid sex -agey;
quietly by hid: gen int oldestml = pid[1];
```

using syntax that we already know…

```
gsort hid sex -agey;
quietly by hid: gen int morder = _n;
gen int oldestml = pid if morder == 1;
```

Introduction to Stata II:

p. 8: Using the following data

```
c:\aerc\stata_training\origdata\anthmodel.dta
```

create a series of interactions between the urban dummy in the dataset and the following variables: `age`, `sex`, `bord`, `hhmemb`, and your region dummies.

```
use c:\aerc\stata_training\origdata\anthmodel.dta;
tab region, gen(reg);
for var age sex bord hhmemb reg1-reg7: gen uX = X * urban;
```

p. 8: Using the following two datasets

```
c:\aerc\stata_training\origdata\anthmodel.dta
c:\aerc\stata_training\origdata\sect01a.dta
```

create a variable that indicates if the father or the mother of the child is the household head.

```
use c:\aerc\stata_training\origdata\sect01a;
 keep hid pid rel;
 sort hid pid;
save c:\temp\temp, replace;

use c:\aerc\stata_training\origdata\anthmodel.dta;
 sort  hid pid;
 merge hid pid using c:\temp\temp;
 tab _m;
 drop if _m==2;
 drop _m;
 gen int parenthd = (rel==3);
```

So we merged the two datasets and identified those kids who are children of the household head (i.e. the mother or the father is the head).

Suppose you want to be more specific and create two variables that indicate if the mother is the head, or if the father is the head…

```
use c:\aerc\stata_training\origdata\anthmodel.dta;
 rename pid chid;
 rename mid pid;
 sort  hid pid;
 merge hid pid using c:\temp\temp;
 tab _m;
 drop if _m==2;
 gen int motherhd = (rel==1);
 drop rel _m;

 rename pid mid;
 rename fid pid;
 sort  hid pid;
```

```
merge hid pid using c:\temp\temp;
tab _m;
drop if _m==2;
gen int fatherhd = (rel==1);
drop rel _m;
gen int parenthd = fatherhd + motherhd;
rename pid  fid;
rename chid pid;
```

The `parenthd` variable created here should be exactly the same as in the previous set of commands. Let's check…

```
.  merge hid pid using c:\temp\temp;

.  tab _m;

     _merge |      Freq.     Percent        Cum.
------------+-----------------------------------
          2 |      21545       89.50       89.50
          3 |       2527       10.50      100.00
------------+-----------------------------------
      Total |      24072      100.00

.  drop if _m==2;
(21545 observations deleted)

.  drop _m;

.  gen int parenth2 = (rel==3);

.  list hid pid mid fid rel if parenthd~=parenth2;

           hid       pid         mid         fid         rel
 221.     2014         9          10           1           4
1546.    16908        10           2           1           4
```

It's not quite the same, but the problem wasn't in our program. It's a data problem! For these two kids, the father is the household head (as defined by matching the father's id with the id's in the household roster), and yet the child is recorded to be the grandchild of the household head.