

## Per Capita – extraction of data from HESA Record

```
*****
* Algorithm for institutions to use with INSTANCE record merged with STUDENT *
* ON MODULE and MODULE record *
* Example for use with 2008/09 HESA data for 2010/11 funding *
*****

*****
* Please also refer to guidance given in circular Higher Education Data Requirements *
* 2008/09 (W09/12HE) and further guidance in supplement found on HEFCW website *
* http://www.hefcw.ac.uk/Finance\_and\_Assurance/calculations.htm *
*****

*****
* 1. Extract eligible students *
* variables taken from HESA record are in upper case of the form ENTITY.FIELDNAME) *
* *
*****

*****
* Notes *
* ne means not equals *
* proc means procedure *
* =: means begins with *
*****

data popn;
set hesa;

where INSTANCE.FUNDCODE = '1' and
       INSTANCE.FESTUMK ne '3' and
       INSTANCE.EXCHANGE not in ('1','2','3','4','6') and
       INSTANCE.MODE in ('01','02','23','24','25','31','52','53','73','74') and
       COURSE.COURSEAIM in (all C codes, all E codes, all H codes, all I codes, all J codes,
                           all M codes);

*****
* 2. Only include students active between 1 August 2008 and 31 July 2009 *
* and not in the final academic year of a non-standard academic year course *
*****

if INSTANCE.TYPEYR = '1' and
   INSTANCE.COMDATE <='31Jul2009'd and
   (INSTANCE.ENDDATE >='1Aug2008'd or INSTANCE.ENDDATE = ' ') then count=1;

if INSTANCE.TYPEYR='2' and
   INSTANCE.ENDDATE <='31Jul2009'd and
   INSTANCE.ENDDATE ne ' ' and
   INSTANCE.ENDDATE > (AVDATE+14) then count=1;

*****
*AVDATE is anniversary of INSTANCE.COMDATE in 2008/09*
*****

if INSTANCE.TYPEYR='2' and
```

```

INSTANCE.ENDDATE <= '31Jul2009'd and
INSTANCE.ENDDATE ne ' ' and
INSTANCE.ENDDATE <= (AVDATE+14) then do;
    if INSTANCE.UNITLGTH = '3' and INSTANCE.SPLENGTH in ('01','02') or
       INSTANCE.UNITLGTH = '4' and INSTANCE.SPLENGTH in ('01' to '14') or
       INSTANCE.UNITLGTH = '5' and INSTANCE.SPLENGTH in ('01' to '42') then
        count=1;

if     INSTANCE.TYPEYR='2' and
      INSTANCE.COMDATE <='31Jul2009'd and
      INSTANCE.ENDDATE = ' ' then count=1;

```

if count ne 1 then delete;

```

*****
* 3. Set number of credit points coded as '999' or missing to 0 *
*****

```

if MODULE.CRDTPTS in ('999',.) then MODULE.CRDTPTS=0;

```

*****
* 4. Delete duplicate modules on courses by student *
*     nodupkey means delete duplicates with same values of institution INSTANCE.HUSID *
*     and MODULE.MODID *
*****

```

proc sort nodupkey;
by institution INSTANCE.HUSID MODULE.MODID;

```

*****
* 5. Sum credits by institution and MODULE.HUSID *
* Dataset 'outcred' is output and contains the total number of credits per student (totcred). *
*****

```

proc summary;
by institution INSTANCE.HUSID;
var MODULE.CRDTPTS;
output out=outcred sum=totcred;

```

*****
* 6. Merge total credit dataset back onto individual instance dataset *
*****

```

data merged;
merge popn outcred;
by institution INSTANCE.HUSID;

```

*****
* 7. Delete duplicate students, keeping 1 entry with highest mode of study *
* Mod values of 'FT', 'PT' derived using INSTANCE.MODE. FT being highest mode. *
*****

```

```

*****
* Input data will be in the following form: *
* *
* Institution      HUSID      mod      CRDTPTS      totcred      *
* 1                1          FT        60           140          *

```

```

*      1          1          FT      40          140          *
*      1          1          FT      20          140          *
*      1          1          PT      20          140          *
*      1          2          PT      10          60           *
*      1          2          PT      50          60           *
*

```

\* Output will be in the following form:

```

*      Institution      HUSID      mod      totcred          *
*      1                1          FT      140           *
*      1                2          PT      60            *
*****

```

```

proc sort;
by institution INSTANCE.HUSID mod;

```

```

if first.INSTANCE.HUSID=1 then keep;

```

```

*****
* 8. Delete students studying less than 10 credit values. *
*****

```

```

if totcred<10 then delete;

```

```

*****
* 9. Count students that are eligible. Dataset 'outtot' contains counts by *
* mode of study and institution. *
*****

```

```

proc summary;
by institution mod;
output out=outtot (keep=institution mod frequency);

```