Cyngor Cyllido Addysg
Uwch Cymru
Higher Education Funding
Council for Wales

Algorithm to estimate non-completed credit values for modules with unknown outcome from the HESA student record (amended October 2009) – version 2

Throughout, fields on the HESA student record are referenced by the field name. A list of which entity each field belongs to is at Annex A.

This method (Method 2 in the original algorithm of 17 September 2009) uses data about the modules with outcomes returned in the reporting year to estimate proportions in each module outcome category for those modules with an unknown outcome. Proportions are calculated for each module subject by institution, mode, level, ASC, year of course and module status (with unspecified year of course and years of course greater than or equal to 6 combined into one category). The proportions for modules continuing from the previous year are used to estimate the proportions for those continuing into the next year.

Step 1: Carry out steps 1, 2 and 3 of the extraction algorithm. Additionally include data where MODSTAT=1 (module continuing from previous reporting year).

Step 2: Create dataset to calculate proportions in each module outcome category

****
Retain data for all module outcomes except those with unknown outcome (those with MODSTAT=1, 2 will be in the resulting dataset). Make adjustment for year of course greater than 6.
****

```
if modout = '6' then delete

if yearprg ≥ 6 then yearprg = 6
```

Step 3: Summarise the number of observations in each module outcome category

****
Create variables to store the number of credits in each module outcome category.
****

```
modout1=0
modout2=0
modout3=0
modout4=0
```

```
if modout='1' then modout1=crdtpts_count
else if modout='2' then modout2=crdtpts_count
else if modout='3' then modout3=crdtpts_count
else if modout='4' then modout4=crdtpts_count
```

****

The results for MODSTAT=1 (continuing from previous year) are to be applied to cases where MODSTAT=3 (continuing into next year), all of which have unknown module outcomes. Therefore, recode MODSTAT=1 to MODSTAT=3 so that results can be merged back onto extracted data to the rows of data the proportions are to be applied to.

****

```
if modstat=’1’ then modstat=’3’
```

****

Sum each variable over institution, mode, level, ASC, module subject (2-character JACS code), year of course and module status. A new variable FREQ containing the number of observations in each institution, mode, level, ASC, module subject, year of course and module status combination is also calculated in this step.

****

```
summary by instid hesmode level asc modsbj2 yearprg modstat
variables modout1 modout2 modout3 modout4 crdtpts_count
output dataset=outcred1
frequency count=FREQ
```

****

Do the same by institution, mode, level, ASC, year of course and module status; by institution, mode, level, ASC and module status; by institution, mode, level and module status; and by institution, mode and level.

****

```
summary by instid hesmode level asc yearprg modstat
variables modout1 modout2 modout3 modout4 crdtpts_count
output dataset=outcred2
frequency count=FREQ

summary by instid hesmode level asc modstat
variables modout1 modout2 modout3 modout4 crdtpts_count
output dataset=outcred3
frequency count=FREQ

summary by instid hesmode level modstat
variables modout1 modout2 modout3 modout4 crdtpts_count
output dataset=outcred4
frequency count=FREQ

summary by instid hesmode level
variables modout1 modout2 modout3 modout4 crdtpts_count
output dataset=outcred5
frequency count=FREQ
```

Step 4: Calculate proportions in each module outcome category, rounded to 2 decimal places, for each of the five summary datasets.

****

The proportions by institution, mode, level, ASC, year of course and module status; institution, mode, level, ASC and module status; institution, mode, level and module status; or institution, mode and level will be used to fill in the gaps where proportions are missing. This will be where a particular module subject, year of course, ASC or module status has no data to use in calculating proportions or there are less than N data rows available to use in calculating proportions. (N = 10, 20 used to generate output). (For the final two sets of calculations, there are either no rows of data in those combinations (leading to missing values of the proportions) or more than the threshold and so no threshold is required.)

****

```
data outcred1

if FREQ≥N then do
      p_modout1=round(modout1/crdtpts_count,0.01)
      p_modout2=round(modout2/crdtpts_count,0.01)
      p_modout3=round(modout3/crdtpts_count,0.01)
      p_modout4=round(modout4/crdtpts_count,0.01)
end


data outcred2

if FREQ≥N then do
      p_modout1_yp=round(modout1/crdtpts_count,0.01)
      p_modout2_yp=round(modout2/crdtpts_count,0.01)
      p_modout3_yp=round(modout3/crdtpts_count,0.01)
      p_modout4_yp=round(modout4/crdtpts_count,0.01)
end


data outcred3

if FREQ≥N then do
      p_modout1_asc=round(modout1/crdtpts_count,0.01)
      p_modout2_asc=round(modout2/crdtpts_count,0.01)
      p_modout3_asc=round(modout3/crdtpts_count,0.01)
      p_modout4_asc=round(modout4/crdtpts_count,0.01)
end


data outcred4
      p_modout1_modstat=round(modout1/crdtpts_count,0.01)
      p_modout2_modstat=round(modout2/crdtpts_count,0.01)
      p_modout3_modstat=round(modout3/crdtpts_count,0.01)
      p_modout4_modstat=round(modout4/crdtpts_count,0.01)
end


data outcred5
      p_modout1_modlev=round(modout1/crdtpts_count,0.01)
      p_modout2_modlev=round(modout2/crdtpts_count,0.01)
      p_modout3_modlev=round(modout3/crdtpts_count,0.01)
      p_modout4_modlev=round(modout4/crdtpts_count,0.01)
end
```

****

Merge these five datasets containing proportions together and where the proportions by module subject are missing, replace with the proportions by mode, level, ASC, year of course and module status.

****

```
merge outcred1 outcred2 outcred3 outcred4 outcred5

if p_modout1=. then do
      p_modout1=p_modout1_yp
      p_modout2=p_modout2_yp
      p_modout3=p_modout3_yp
      p_modout4=p_modout4_yp
end
```

****

Where the proportions by mode, level, ASC, year of course and module status are also missing, replace with the proportion by mode, level, ASC and module status.

****

```
if p_modout1=. then do
      p_modout1=p_modout1_asc
      p_modout2=p_modout2_asc
      p_modout3=p_modout3_asc
      p_modout4=p_modout4_asc
end
```

****

Where the proportions by mode, level, ASC and module status are also missing, replace with the proportion by mode, level and module status.

****

```
if p_modout1=. then do
      p_modout1=p_modout1_modstat
      p_modout2=p_modout2_modstat
      p_modout3=p_modout3_modstat
      p_modout4=p_modout4_modstat
end
```

****

Where the proportions by mode, level and module status are also missing, replace with the proportion by mode and level.

****

```
if p_modout1=. then do
      p_modout1=p_modout1_modlev
      p_modout2=p_modout2_modlev
      p_modout3=p_modout3_modlev
      p_modout4=p_modout4_modlev
end
```

Step 5: These proportions are merged back onto the dataset in Step 6 of the extraction algorithm and used to calculate the estimated number of credits in each module outcome category for rows of data with MODOUT=6