

Example 2: Analysis at household level

How many households are in low income?

```
/* Create a temporary dataset using the HBAI dataset which is restricted to only
those cases where benunit equals 1 and is therefore at a household level. */
rsubmit;
data hbai;
set hbai.hbai0708;
where benunit =1;
run;

/* Calculating the number of households in low income using the low income flag
on the HBAI dataset.
As dataset is at a household level use the household grossing factor. */
proc freq data=hbai;
table low60bhc;
weight gs_newhh;
run;
endrsubmit;
```

Output:

The FREQ Procedure

LOW60BHC	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	20994311	80.55	20994311	80.55
1	5070282	19.45	26064593	100.00

As LOW60BHC equals 1 are those households in low income the answer, before housing costs is 5,070,282 or 5.1 million using HBAI rounding conventions.

Example 3: What proportion of children live in couple families? What proportion of children in couple families are in low income households?

```
/* Create a temporary dataset using the HBAI dataset and then create a variable
(KIDFAM) to indicate whether children are in lone parent or couple family using
the HBAI variable KIDECOBU. */
rsubmit;
data hbai;
set hbai.hbai0708;
kidfam = 0;
if kidecobu in (1,2,3) then kidfam = 1;
if kidecobu in (4,5,6,7,8,9) then kidfam = 2;
run;

/* Calculating the number of children by the variable created above.
As dataset is at a benunit level use the children grossing factor. */
proc freq data=hbai;
table kidfam;
weight gs_newch;
run;

/* Calculating the number of children living in low income households by the
KIDFAM variable created above and using the low income flag on the HBAI dataset.
As dataset is at a benunit level use the children grossing factor. */
proc freq data=hbai;
table low60bhc*kidfam;
weight gs_newch;
run;
endrsubmit;
```

Output 1:

The FREQ Procedure

KIDFAM	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	3060195	23.82	3060195	23.82
2	9787606	76.18	12847801	100.00

This output shows that the proportion of children in couple families is 76.18, or 76 per cent, using HBAI rounding conventions.

Output 2:

The FREQ Procedure

Table of LOW60BHC by KIDFAM

LOW60BHC	KIDFAM		
Frequency			
Percent			
Row Pct			
Col Pct	1	2	Total
0	1955038	8001289	9956327
	15.22	62.28	77.49
	19.64	80.36	
	63.89	81.75	
1	1105157	1786317	2891474
	8.60	13.90	22.51
	38.22	61.78	
	36.11	18.25	
Total	3060195	9787606	1.285E7
	23.82	76.18	100.00

This output shows that the proportion of children in couple families living in low income households, before housing costs, is 18.25 or 18 per cent using HBAI rounding conventions.

Example 4: Merging FRS and HBAI data to calculate the number of working-age adults in poverty by the amount of savings and assets they have.

```
/* Create a temporary dataset using the FRS Benunit dataset and then create a
variable (CAPITAL) banding the amount of savings and assets held using the FRS
variable TOTCAPB2. */
rsubmit;
data save (keep=sernum benunit capital);
set frs0708.benunit (keep=sernum benunit totcapb2);

if totcapb2 le 0 then capital=1; /* No savings */
if totcapb2 gt 0 and totcapb2 lt 1500 then capital=2; /* £1 to £1499 */
if totcapb2 ge 1500 and totcapb2 lt 3000 then capital=3; /* £1500 to £2999 */
if totcapb2 ge 3000 and totcapb2 lt 8000 then capital=4; /* £3000 to £7999 */
if totcapb2 ge 8000 and totcapb2 lt 10000 then capital=5; /* £8000 to £9999 */
if totcapb2 ge 10000 and totcapb2 lt 16000 then capital=6; /* £10000 to £15999 */
if totcapb2 ge 16000 and totcapb2 lt 20000 then capital=7; /* £16000 to £19999 */
if totcapb2 ge 20000 then capital=8; /* £20000 or more */
run;

/* Merge the HBAI dataset onto the temporary dataset using the variables in the BY
statement and keeping only those records that are held in the HBAI dataset using
the (IN=A) and IF A statements. */
data save;
merge save
      hbai.hbai0708 (in=a);
by sernum benunit;
if a;
run;

/* Calculating the number of working-age adults living in low income households
by the amount of savings and assets held (CAPITAL) and using the low income flag
on the HBAI dataset.
As dataset is at a benunit level use the working-age adults grossing factor. */
proc freq data=save;
table low60bhc*capital;
weight gs_newwa;
run;
endrsubmit;
```

Output:

The FREQ Procedure

Table of LOW60BHC by CAPITAL

LOW60BHC	CAPITAL								Total
Frequency									
Percent									
Row Pct									
Col Pct	1	2	3	4	5	6	7	8	Total
0	8624190	6871305	2283557	4402976	1033513	1974937	845043	4325714	3.036E7
	24.01	19.13	6.36	12.26	2.88	5.50	2.35	12.04	84.52
	28.41	22.63	7.52	14.50	3.40	6.50	2.78	14.25	
	73.11	86.86	90.61	91.63	89.88	92.45	91.58	92.42	
1	3172650	1039802	236579	402357	116383	161257	77650	354797	5561475
	8.83	2.89	0.66	1.12	0.32	0.45	0.22	0.99	15.48
	57.05	18.70	4.25	7.23	2.09	2.90	1.40	6.38	
	26.89	13.14	9.39	8.37	10.12	7.55	8.42	7.58	
Total	1.18E7	7911107	2520136	4805333	1149896	2136194	922693	4680511	3.592E7
	32.84	22.02	7.02	13.38	3.20	5.95	2.57	13.03	100.00

As LOW60BHC equals 1 are those working-age adults living in low income households the answer for each of the values of CAPITAL, before housing costs, using HBAI rounding conventions are as follows;

- Band 1 (No savings) is 3.2m
- Band 2 (£1 to £1,499) is 1.0m
- Band 3 (£1,500 to £2,999) is 0.2m
- Band 4 (£3,000 to £7,999) is 0.4m
- Band 5 (£8,000 to £9,999) is 0.1m
- Band 6 (£10,000 to £15,999) is 0.2m
- Band 7 (£16,000 to £19,999) is 0.1m
- Band 8 (£20,000 or more) is 0.4m

Example 5: Using a retain statement to calculate how many children under 10 by the number of children under 10 in the family are in poverty.

```
/* Create a temporary dataset using the FRS Child dataset and create
variables to count the number of children under 10 (UNDER10) and over 10
(OVER10) years of age. The BY, RETAIN, FIRST. and LAST. statements are used
to collapse a dataset from one level to another; in this case from an
Individual to a Benefit Unit level. */
rsubmit;
data child10;
set frs0708.child;
by sernum benunit;
retain under10 over10;
if first.benunit then do;
    under10=0; over10=0; end;
    if age le 10 then under10=under10+1;
    if age gt 10 then over10=over10+1;
if last.benunit;
run;

/* Merge the HBAI dataset onto the temporary dataset using the variables in the BY
statement and keeping only those records that are held in the HBAI dataset using
the (IN=A) and IF A statements. */
data hbaichild10;
merge child10
      hbai.hbai0708 (in=a);
by sernum benunit;
if a;
run;

/* Calculating the number of children living in low income households by the
UNDER10 variable created above and using the low income flag on the HBAI dataset.
As dataset is at a benunit level use the children grossing factor. */
proc freq data=hbaichild10;
table low60bhc*under10;
weight gs_newch;
run;
endrsubmit;
```

Output:

The FREQ Procedure

Table of LOW60BHC by UNDER10

LOW60BHC	UNDER10							Total
Frequency	0	1	2	3	4	5	6	
Percent								
Row Pct								
Col Pct	0	1	2	3	4	5	6	Total
0	2746216	3237432	3007072	743610	186515	27550	7932	9956327
	21.37	25.20	23.41	5.79	1.45	0.21	0.06	77.49
	27.58	32.52	30.20	7.47	1.87	0.28	0.08	
	80.34	78.02	79.88	64.74	66.13	50.62	26.09	
1	672196	911911	757545	404939	95536	26877	22470	2891474
	5.23	7.10	5.90	3.15	0.74	0.21	0.17	22.51
	23.25	31.54	26.20	14.00	3.30	0.93	0.78	
	19.66	21.98	20.12	35.26	33.87	49.38	73.91	
Total	3418412	4149343	3764617	1148549	282051	54427	30402	1.285E7
	26.61	32.30	29.30	8.94	2.20	0.42	0.24	100.00

As LOW60BHC equals 1 are those children living in low income households the answer for the number of children under 10 by the number of children under 10 in poverty, before housing costs, using HBAI rounding conventions are as follows;

- No children under 10 is 0.7m
- One child under 10 is 0.9m
- Two children under 10 is 0.8m
- Three children under 10 is 0.4m
- Four children under 10 is 0.1m
- Five children under 10 is 0.0m
- Six children under 10 is 0.0m

Example 6: Checking sample sizes for the above example.

```
/* Create a temporary dataset using the FRS Child dataset and create
variables to count the number of children under (UNDER10) and over 10
(OVER10) years of age. The BY, RETAIN, FIRST. and LAST. statements are used
to collapse a dataset from one level to another; in this case from an
Individual to a Benefit Unit level. */
rsubmit;
data child10;
set frs0708.child;
by sernum benunit;
retain under10 over10;
if first.benunit then do;
    under10=0; over10=0; end;
    if age le 10 then under10=under10+1;
    if age gt 10 then over10=over10+1;
if last.benunit;
run;

/* Merge the HBAI dataset onto the temporary dataset using the variables in the BY
statement and keeping only those records that are held in the HBAI dataset using
the (IN=A) and IF A statements. */
data hbaichild10;
merge child10
        hbai.hbai0708 (in=a);
by sernum benunit;
if a;
run;

/* As the dataset is at a benunit level we need to create a variable to give the
actual numbers of children on the dataset. E.g if for a specific record
GS_NEWCH=2158 and GS_NEWBU=1079 then GS_SAMP=2 and means that there are 2 children
in this family.
Note: If your dataset is at an individual level then this step is not required as
you can just use the PROC FREQ as below but without the WEIGHT statement. */
data hbaichild10;
set hbaichild10;
gs_samp=gs_newch/gs_newbu;
run;

/* Calculating the sample sizes of children living in low income households by the
UNDER10 variable created above and using the low income flag on the HBAI dataset.
As it is the sample sizes that are of interest here use the sample grossing factor
GS SAMP created above. */
proc freq data=hbaichild10;
table low60bhc*under10;
weight gs_samp;
run;
endrsubmit;
```

Output:

The FREQ Procedure

Table of LOW60BHC by UNDER10

LOW60BHC	UNDER10							Total
Frequency	0	1	2	3	4	5	6	
Percent								
Row Pct								
Col Pct	0	1	2	3	4	5	6	Total
0	2711	3249	3193	871	247	36	12	10319
	20.06	24.05	23.63	6.45	1.83	0.27	0.09	76.37
	26.27	31.49	30.94	8.44	2.39	0.35	0.12	
	78.85	76.83	79.13	65.49	67.30	53.73	26.09	
1	727	980	842	459	120	31	34	3193
	5.38	7.25	6.23	3.40	0.89	0.23	0.25	23.63
	22.77	30.69	26.37	14.38	3.76	0.97	1.06	
	21.15	23.17	20.87	34.51	32.70	46.27	73.91	
Total	3438	4229	4035	1330	367	67	46	13512
	25.44	31.30	29.86	9.84	2.72	0.50	0.34	100.00

This output is the same as that for example 5 but this time shows the number of children on the HBAI dataset. Given the very low number of those in categories 5 and 6 it would be advisable to combine these with category 4 and display this as four or more children.

Example 7: Removing an item from income and recalculating the poverty line.

```
/* Create a temporary dataset using the FRS Benefits dataset keeping only the
details of the benefits required, in this case 1 (DLA - self care),
2 (DLA - mobility) and 12 (Attendance Allowance). */
rsubmit;
data disben;
set frs0708.benefits (keep=sernum benunit person benefit benamt);
where benefit in (1,2,12);
run;

/* Collapsing the temporary dataset to a household level to be compatible with
the HBAI dataset. */
data disben (keep=sernum benam);
set disben;
by sernum;
retain benam;
if first.sernum then benam=0;
benam=benam+benamt;
if last.sernum;
run;

/* Merge the HBAI dataset onto the temporary dataset using the variable in the BY
statement and keeping only those records that are held in the HBAI dataset using
the (IN=B) and IF B statements. Also using the IF NOT A statement to enter a zero
value for BENAM if the HBAI record does not have a benefit amount. */
data hbai;
merge hbai.hbai0708 (in=b)
      disben (in=a);
by sernum;
if b;
if not a then benam=0;

/* Creating a new deflated equivalised household income (S_OE_BHC2) by deducting
the total of the benefits from the existing deflated household income.
Note: Before any amounts derived from the FRS can be added/deducted from the
HBAI income they must be deflated (e.g. BHCDEF) and equivalised (e.g EQOBHCHH). */
s_oe_bhc2=s_oe_bhc-((benam*bhcdef)/eqobhchh);
s_oe_ahc2=s_oe_ahc-((benam*ahcdef)/eqoahchh);
run;

/* Calculating medians for the new variables created above and saving these in
a temporary dataset called STATO.
As dataset is at a benunit level and we want the median for the whole population
use the population grossing factor. */
proc univariate data=hbai noprint;
var s_oe_ahc2 s_oe_bhc2;
output out=stato median=mdoeahc2 mdoebhc2;
freq gs_newpp;
run;

/* The new medians need to be added to the dataset. The IF _N_ statement tells
SAS to put the medians against every record in the dataset. */
data hbai;
if _n_=1 then set stato;
set hbai;
run;

/* Create a second temporary dataset using the first, then construct a new
poverty line using the new income (S_OE_BHC2) and median (MDOEBHC2) variables
to indicate if the benunit unit is in low income (=1) or not (=0). */
```

```

data newhbai;
set hbai;

/* Recalculated poverty line */
low60bhc2=0;
low60ahc2=0;
if s_oe_bhc2 lt 0.6*mdoebhc2 then low60bhc2=1;
if s_oe_ahc2 lt 0.6*mdoeahc2 then low60ahc2=1;
run;

/* Calculating the number of individuals living in low income households. First
on the original basis using the low income flag on the HBAI dataset and then on
the recalculated basis using the new poverty line created above.
As dataset is at a benunit level use the population grossing factor. */
proc freq data=newhbai;
table low60bhc2;
weight gs_newpp;
run;

proc freq data=newhbai;
table low60ahc2;
weight gs_newpp;
run;
endrsubmit;

```

Output using original poverty line:

The FREQ Procedure

LOW60BHC	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	48891213	81.67	48891213	81.67
1	10970168	18.33	59861381	100.00

This output shows that the number of individuals living in low income households, before housing costs, using HBAI rounding conventions is 11.0 million or 18 per cent.

Output using new poverty line:

The FREQ Procedure

LOW60BHC2	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	48348590	80.77	48348590	80.77
1	11512791	19.23	59861381	100.00

This output shows that the number of individuals living in low income households, before housing costs, using the new poverty line and HBAI rounding conventions is 11.5 million or 19 per cent.

In this example we can see that the effect of removing the disability benefits from the household income results in an increase in the number of individuals living in low income households.

Example 8: Producing analysis from an individual level dataset by disability status

```
/* Create a temporary dataset using the FRS Adult dataset and create a variable
(DDA) to indicate whether a disability exists. Then create a further one (CLIENT)
to indicate whether a Working-age adult (=1) or a Pensioner (=2). */
rsubmit;
data adultdis;
set frs0708.adult(drop=adDDA);
  if disdif1 in (.B,.C) or
     disdif2 in (.B,.C) or
     disdif3 in (.B,.C) or
     disdif4 in (.B,.C) or
     disdif5 in (.B,.C) or
     disdif6 in (.B,.C) or
     disdif7 in (.B,.C) or
     disdif8 in (.B,.C) or
     disdif9 in (.B,.C)
  then DDA=.D;
else
  if disdif1=1 or
     disdif2=1 or
     disdif3=1 or
     disdif4=1 or
     disdif5=1 or
     disdif6=1 or
     disdif7=1 or
     disdif8=1 or
     disdif9=1
  then DDA=1;
else DDA=0;
  if DDA not in (0,1) then DDA=.D;

client=1;
if age ge 65 or (age ge 60 and sex=2) then client=2;
run;

/* Create a temporary dataset using the FRS Child dataset and create a variable
(DDA) to indicate whether a disability exists. Then create a further one (CLIENT)
to indicate the individual is a child (=0). */
data kiddis;
set frs0708.child(drop=chDDA);
  if cdisdif1 in (.B,.C) or
     cdisdif2 in (.B,.C) or
     cdisdif3 in (.B,.C) or
     cdisdif4 in (.B,.C) or
     cdisdif5 in (.B,.C) or
     cdisdif6 in (.B,.C) or
     cdisdif7 in (.B,.C) or
     cdisdif8 in (.B,.C) or
     cdisdif9 in (.B,.C)
  then DDA=.D;
else
  if cdisdif1=1 or
     cdisdif2=1 or
     cdisdif3=1 or
     cdisdif4=1 or
     cdisdif5=1 or
     cdisdif6=1 or
     cdisdif7=1 or
     cdisdif8=1 or
     cdisdif9=1
  then DDA=1;
else DDA=0;
  if DDA not in (0,1) then DDA=.D;
```

```

        cdisdif9=1
    then DDA=1;
else DDA=0;
    if DDA not in (0,1) then DDA=.D;
client=0;
run;

/* Merge the two temporary datasets created above using the variables in the BY
statement. */
data inddis;
merge adultdis
    kiddis;
by sernum benunit person;

/* Using the DDA and CLIENT variables created above create another variable
(CLIENTDDA) which combines these and makes the final output easier to read. */
if dda=1 then do;
    if client=0 then clientdda=1;
    if client=1 then clientdda=2;
    if client=2 then clientdda=3;
end;

if dda=0 then do;
    if client=0 then clientdda=4;
    if client=1 then clientdda=5;
    if client=2 then clientdda=6;
end;
run;

/* Merge the temporary dataset created above with the HBAI dataset using the
variables in the BY statement. This will keep the dataset at the individual
level. */
data inddis;
merge inddis
    hbai.hbai0708;
by sernum benunit;
run;

/* Calculating the number of disabled/non-disabled individuals by client group
(CLIENTDDA) living in low income households using the low income flag on the HBAI
dataset.
As dataset is at an individual level use the benunit level grossing factor. */
proc freq data=inddis;
table low60bhc*clientdda;
weight gs_newbu;
run;
endrsubmit;

```

Output:

The FREQ Procedure

Table of LOW60BHC by CLIENTDDA

LOW60BHC	CLIENTDDA						Total
Frequency							
Percent							
Row Pct							
Col Pct	1	2	3	4	5	6	Total
0	568056	3582301	3956149	9388271	2.678E7	4617502	4.889E7
	0.95	5.98	6.61	15.68	44.73	7.71	81.67
	1.16	7.33	8.09	19.20	54.77	9.44	
	73.82	72.07	78.10	77.73	86.52	76.63	
1	201451	1388161	1109387	2690023	4173314	1407832	1.097E7
	0.34	2.32	1.85	4.49	6.97	2.35	18.33
	1.84	12.65	10.11	24.52	38.04	12.83	
	26.18	27.93	21.90	22.27	13.48	23.37	
Total	769507	4970462	5065536	1.208E7	3.095E7	6025334	5.986E7
	1.29	8.30	8.46	20.18	51.71	10.07	100.00

This output shows that the number and proportion of those individuals living in low income households, before housing costs, by disability and client group using HBAI rounding conventions is as follows;

- Disabled Children is 0.2 million, 26 per cent.
- Disabled Working-age adults is 1.4 million, 28 per cent.
- Disabled Pensioners is 1.1 million, 22 per cent.
- Non-disabled Children is 2.7 million, 22 per cent.
- Non-disabled Working-age adults is 4.2 million, 13 per cent.
- Non-disabled Pensioners is 1.4 million, 23 per cent.