LIDNS derived variable specification

Chapter 1: Background and Purpose

No derived variables archived for Chapter 1

Chapter 2: Methodology and response

AGEG (D) Age of respondent, grouped

1 2-10 years
2 11-18 years
3 19-34 years
4 35-49 years
5 50-64 years
6 65+ years

```
SPSS Syntax

recode agep (2 thru 10=1) (11 thru 18=2) (19 thru 34=3) (35 thru 49=4) (50 thru 64=5)
(65 thru high=6) into ageg .
variable label ageg  "(D) Age of respondent, grouped".
value labels ageg
1 '2-10 years'
2 '11-18 years'
3 '19-34 years'
4 '35-49 years'
5 '50-64 years'
6 '65+ years'.
```

AGEGR2 (D) Adult vs. child

1 Adult 19+ years
2 Child 2-18 years

```
SPSS Syntax

recode agep (19 thru Highest=1)  (2 thru 18=2) into agegr2 .
variable label agegr2 '(D) Adult vs. child'.
value label agegr2
1 'Adult 19+ years'
2 'Child 2-18 years'.
```

QUARTER4 (D) Quarter 1,2,3,4

1 Q1a+Q1b
2 Q2
3 Q3
4 Q4

```
SPSS Syntax

*GB.
do if country>=1 & country<=3.
RECODE issmon (1,2,3,13,14,15=1) (4,5,6=2) (7,8,9=3) (10,11,12=4) INTO Quarter4 .
end if.

*NI.
do if country=4.
recode niquart (2=2) (3=3) (4=4) (5=1) into Quarter4.
end if.

  VARIABLE LABELS Quarter4 '(D) Quarter 1,2,3,4'.
  EXECUTE .
  VALUE LABELS Quarter4
  1 'Q1a+Q1b'
  2 'Q2'
  3 'Q3'
  4 'Q4' .
```

```
*Note hhs containing preg women only (773) are treated as INELIGIBLE.
```

HHCOMP (D) Household composition

1 1 adult of working age
2 1 adult of retirement age
3 2+ adults, at least one of working age
4 2+ adults, all of retirement age
5 1 adult only, 1+ children
6 2+ adults, 1+ children
0 NO "ADUTLS": ALL AGED UNDER 19
11 Not known

```
SPSS Syntax

*** Definitions.
* Adult - 19+.
* Child - 0-18.

* Number of adults aged 19+.
if (gdvage>=19) tempad1=1.
if (gdvage2>=19) tempad2=1.
if (gdvage3>=19) tempad3=1.
if (gdvage4>=19) tempad4=1.
if (gdvage5>=19) tempad5=1.
if (gdvage6>=19) tempad6=1.
if (gdvage7>=19) tempad7=1.
if (gdvage8>=19) tempad8=1.
if (gdvage9>=19) tempad9=1.
if (gdvage10>=19) tempad10=1.

compute tadn=sum(tempad1, tempad2, tempad3, tempad4, tempad5, tempad6, tempad7,
tempad8, tempad9, tempad10).
recode tadn (sysmis=0) (else=copy).

* Number of children aged 0-18.
if (gdvage<19 & gdvage>=0) tempch1=1.
if (gdvage2<19 & gdvage2>=0) tempch2=1.
if (gdvage3<19 & gdvage3>=0) tempch3=1.
if (gdvage4<19 & gdvage4>=0) tempch4=1.
if (gdvage5<19 & gdvage5>=0) tempch5=1.
if (gdvage6<19 & gdvage6>=0) tempch6=1.
if (gdvage7<19 & gdvage7>=0) tempch7=1.
if (gdvage8<19 & gdvage8>=0) tempch8=1.
if (gdvage9<19 & gdvage9>=0) tempch9=1.
if (gdvage10<19 & gdvage10>=0) tempch10=1.

compute tchn=sum (tempch1, tempch2, tempch3, tempch4, tempch5, tempch6, tempch7,
tempch8, tempch9, tempch10).
recode tchn (sysmis=0) (else=copy).

* Number of adults of retirement age (60+ for women, 65+ for men).
* Women.
if (gdvage>=60 & gsex=2) tempp1=1.
if (gdvage2>=60 & gsex=2) tempp2=1.
if (gdvage3>=60 & gsex=2) tempp3=1.
if (gdvage4>=60 & gsex=2) tempp4=1.
if (gdvage5>=60 & gsex=2) tempp5=1.
if (gdvage6>=60 & gsex=2) tempp6=1.
if (gdvage7>=60 & gsex=2) tempp7=1.
if (gdvage8>=60 & gsex=2) tempp8=1.
if (gdvage9>=60 & gsex=2) tempp9=1.
if (gdvage10>=60 & gsex=2) tempp10=1.

* Men.
if (gdvage>=65 & gsex=1) tempp11=1.
if (gdvage2>=65 & gsex=1) tempp12=1.
if (gdvage3>=65 & gsex=1) tempp13=1.
if (gdvage4>=65 & gsex=1) tempp14=1.
if (gdvage5>=65 & gsex=1) tempp15=1.
if (gdvage6>=65 & gsex=1) tempp16=1.
```

```
if (gdvage7>=65 & gsex=1) tempp17=1.
if (gdvage8>=65 & gsex=1) tempp18=1.
if (gdvage9>=65 & gsex=1) tempp19=1.
if (gdvage10>=65 & gsex=1) tempp20=1.

compute tpn=sum (tempp1, tempp2, tempp3, tempp4, tempp5, tempp6, tempp7, tempp8,
tempp9, tempp10,
tempp11, tempp12, tempp13, tempp14, tempp15, tempp16, tempp17, tempp18, tempp19,
tempp20).
recode tpn (sysmis=0) (else=copy).

* Creating a variable for household composition.
compute hhcomp=0.

* Group 1: 1 adult of working age, no children.
if (tchn=0 & tadn=1 & tpn=0) hhcomp=1.

* Group 2: 1 adult of retirement age, no children.
if (tchn=0 & tadn=1 & tpn=1) hhcomp=2.

* Group 3: 2+ adults, at least one of whom is of working age, no children.
if (tchn=0 & tadn>=2 & tadn>tpn) hhcomp=3.

* Group 4: 2+ adults, all of whom are of retirement age, no children.
if (tchn=0 & tadn>=2 & tadn=tpn) hhcomp=4.

* Group 5: 1 adult only, 1+ children.
if (tadn=1 & tchn>0) hhcomp=5.

* Group 6: 2+ adults, 1+ children.
if (tadn>=2 & tchn>0) hhcomp=6.

variable label hhcomp "(D) Household composition".

value labels hhcomp
1 '1 adult of working age'
2 '1 adult of retirement age'
3 '2+ adults, at least one of working age'
4 '2+ adults, all of retirement age'
5 '1 adult only, 1+ children'
6 '2+ adults, 1+ children'
0 'NO "ADUTLS": ALL AGED UNDER 19'
 11 'Not known'.
```

REGION (D)Country/region

1 England: North
2 England: Central/Midlands
3 England: South (incl. London)
5 Wales
4 Scotland
6 Northern Ireland

```
SPSS Syntax

recode gor (1 thru 3=1) (4,5=2) (6 thru 9=3) (11=4) (10=5) (12=6) into region.

variable label region "(D)Country/region".
value label region
1 'England: North'
2 'England: Central/Midlands'
3 'England: South (incl. London)'
5 'Wales'
4 'Scotland'
6 'Northern Ireland'.
```

AGRNURSE (D) whether agreed nurse visit

1 agreed Nurse visit
2 f/p not agreed Nurse visit

```
SPSS Syntax

recode nurse (1=1) (else=2) into AgrNurse.
variable label AgrNurse '(D) whether agreed nurse visit'.
value label AgrNurse
1 'agreed Nurse visit'
2 'f/p not agreed Nurse visit'.
execute.
```

ETHGR5 (D) Ethnic group, 5 groups

1 White
2 Mixed ethnic group
3 Black or Black British
4 Asian or asian British
5 Any other group

```
SPSS Syntax

**Ethnic group - Ethgr5.
*GB.
do if country>=1 & country<=3.
recode ethgrp (1 thru 2=1) (3 thru 6 = 2) (7 thru 10 = 4) (11 thru 13 = 3) (14 thru 15
= 5) into ethgr5.
end if.
*NI.
do if country=4.
recode ethni  (1 thru 4 = 1) (5 thru 8 = 2) (13 thru 15 = 3) (9 thru 12 = 4) (16 thru
17 = 5)  into ethgr5.
end if.
Variable label ethgr5 "(D) Ethnic group, 5 groups".
value label ethgr5
1 'White'
2 'Mixed ethnic group'
3 'Black or Black British'
4 'Asian or asian British'
5 'Any other group'.
```

ETHGR2 (D) Ethnic group, 2 groups

1 White
2 Non-white

```
SPSS Syntax

recode ethgr5 (1=1) (2,3,4,5=2) into ethgr2 .
Variable label ethgr2 "(D) Ethnic group, 2 groups".
value label ethgr2
1 'White'
2 'Non-white' .
execute.
```

ETHGR4 (D) Ethnic group, 4 groups

1 White
2 Black
3 Asian
4 Other

```
SPSS Syntax

*GB.

do if country>=1 & country<=3.
```

```
recode ethgrp (1 thru 2=1) (3 thru 6, 14, 15=4) (7 thru 10=3) (11 thru 13=2) into
ethgr4.
end if.

* NIR.

do if country=4.
recode ethni (1 thru 4=1) (5 thru 8, 16, 17=4) (9 thru 12=3) (13 thru 15=2) into
ethgr4.
end if.
Variable label ethgr4 "(D) Ethnic group, 4 groups".
value label ethgr4
1 'White'
2 'Black'
3 'Asian'
4 'Other'.
```

WILLBS (D) Willing to have blood sample taken

1 Willing
2 f/p Not willing

```
SPSS Syntax

recode bswill (1=1) (else=2) into WillBS .
variable label WillBS '(D) Willing to have blood sample taken'.
value label WillBS
1 'Willing'
2 'f/p Not willing'.
execute.
```

BLOODOC1 (D) Blood outcome

1 Blood sample taken
2 No sample taken

```
SPSS Syntax

RECODE SampTak (1,2=1) (3=2) (ELSE=COPY) INTO BloodOC1.
VARIABLE LABELS BloodOC1 'Blood outcome'.
  EXECUTE .
  VALUE LABELS BloodOC1
  1 'Blood sample taken'
  2 'No sample taken'.
```

MEASWEIG (D) Weight measured

1 Weight measured
2 f/p weight not measured

```
SPSS Syntax

recode respwts (0,1=1) (else=2) into measweig .
variable label measweig '(D) weight measured'.
value label measweig
1 'Weight measured'
2 'f/p weight not measured'.
execute.
```

MEASHEIG (D) Height measured

1 Height measured
2 f/p height not measured

```
SPSS Syntax
```

```
recode resphts (0,1=1) (else=2) into measheig .
variable label measheig '(D) height measured'.
value label measheig
1 'Height measured'
2 'f/p height not measured'.
execute.
```

MEASWH (D) WH measured

1 WH measured
2 f/p WH not measured

```
SPSS Syntax

recode respwh (1,2=1) (else=2) into measwh .
variable label measWH '(D) WH measured'.
value label measWH
1 'WH measured'
2 'f/p WH not measured'.
execute.
```

MEASMUAC (D) MUAC measured

1 MUAC measured
2 f/p MUAC not measured

```
SPSS Syntax

recode muacint (1=1) (else=2) into measMUAC .
variable label measMUAC '(D) MUAC measured'.
value label measMUAC
1 'MUAC measured'
2 'f/p MUAC not measured'.
execute.
```

MEASBP (D) BP measured

1 BP measured
2 f/p BP not measured

```
SPSS Syntax

recode BPOut (1=1) (else=2) into measBP .
variable label measBP '(D) BP measured'.
value label measBP
1 'BP measured'
2 'f/p BP not measured'.
execute.
```

NVISIT (D) whether visited by AgrNurse

1 visited by AgrNurse
2 f/p not visited by AgrNurse

```
SPSS Syntax

recode nuroutc (810=1) (else=2) into Nvisit.
variable label Nvisit '(D) whether visited by AgrNurse'.
value label NVisit
1 'visited by AgrNurse'
2 'f/p not visited by AgrNurse'.
execute.
```

Chapter 3: Characteristics of the low income population in LIDNS

SBEN (D) Receive at least one state benefit

```
1 Receives at least one state benefit
2 No state benefits
```

```
SPSS Syntax

**Recode benefits into receives any state benefit SBEN.
recode srcinc01 (1 thru hi = 0) (-9 thru -1 = COPY) into sben.
if range (srcinc01, 1, 12) sben = 1.
if range (srcinc02, 1, 12) sben = 1.
if range (srcinc03, 1, 12) sben = 1.
if range (srcinc04, 1, 12) sben = 1.
if range (srcinc05, 1, 12) sben = 1.
if range (srcinc06, 1, 12) sben = 1.
recode sben (0=2) into sben.
VARIABLE LABELS sben "(D) Receive at least one state benefit".
VALUE LABELS sben
1 "Receives at least one state benefit"
2 "No state benefits".
Missing values sben (-9 thru -1).
```

EARN (D) Earnings per week

```
SPSS Syntax

* Earnings from jobs.
compute earn=0.
if frqearn=1 earn=earnings.
if frqearn=2 earn=earnings/2.
if frqearn=4 earn=earnings/4.
if frqearn=5 earn=earnings*12/52.
if frqearn=13 earn=earnings/13.
if frqearn=26 earn=earnings/26.
if frqearn=52 earn=earnings/52.

missing values earnings frqearn earn ().
if earnings=-8 earn=-8.
if earnings=-9 earn=-9.
if earnings=-1 earn=-1.
if frqearn=97 | frqearn=-8 earn=-8.
execute.
missing values earnings frqearn earn (lo thru -1).
variable label earn '(D) Earnings per week'.
```

CBEN (D) Child benefit per week

```
SPSS Syntax

* Child benefit.
compute cben=0.
if cbenpd=1 cben=cbenamt.
if cbenpd=2 cben=cbenamt/2.
if cbenpd=4 cben=cbenamt/4.
if cbenpd=5 cben=cbenamt*12/52.
if cbenpd=13 cben=cbenamt/13.
if cbenpd=26 cben=cbenamt/26.
if cbenpd=52 cben=cbenamt/52.

missing values cbenamt cbenpd cben ().
if cbenamt=-8 cben=-8.
if cbenamt=-9 cben=-9.
if cbenamt=-1 cben=-1.
if cbenpd=97 | cbenpd=-8 cben=-8.
```

```
execute.
missing values cbenamt cbenpd cben (lo thru -1).
var label cben "(D) Child benefit per week".
```

WTC (D) Working tax credit per week

```
SPSS Syntax

* Working tax credit.
compute wtc=0.
if wtcpd=1 wtc=wtcamt.
if wtcpd=2 wtc=wtcamt/2.
if wtcpd=4 wtc=wtcamt/4.
if wtcpd=5 wtc=wtcamt*12/52.
if wtcpd=13 wtc=wtcamt/13.
if wtcpd=26 wtc=wtcamt/26.
if wtcpd=52 wtc=wtcamt/52.

missing values wtcamt wtcpd wtc ().
if wtcamt=-8 wtc=-8.
if wtcamt=-9 wtc=-9.
if wtcamt=-1 wtc=-1.
if wtcpd=97 | wtcpd=-8 wtc=-8.
execute.
missing values wtcamt wtcpd wtc (lo thru -1).
var label wtc "(D) Working tax credit per week".
```

CTC (D) Child Tax Credit per week

```
SPSS Syntax

* Child Tax Credit.
compute ctc=0.
if ctcpd=1 ctc=ctcamt.
if ctcpd=2 ctc=ctcamt/2.
if ctcpd=4 ctc=ctcamt/4.
if ctcpd=5 ctc=ctcamt*12/52.
if ctcpd=13 ctc=ctcamt/13.
if ctcpd=26 ctc=ctcamt/26.
if ctcpd=52 ctc=ctcamt/52.

missing values ctcamt ctcpd ctc ().
if ctcamt=-8 ctc=-8.
if ctcamt=-9 ctc=-9.
if ctcamt=-1 ctc=-1.
if ctcpd=97 | ctcpd=-8 ctc=-8.
execute.
missing values ctcamt ctcpd ctc (lo thru -1).
var label ctc "(D) Child Tax Credit per week".
```

JSA (D) Job Seeker's Allowance per week

```
SPSS Syntax

* Jobseeker's Allowance.
compute jsa=0.
if jsapd=1 jsa=jsaamt.
if jsapd=2 jsa=jsaamt/2.
if jsapd=4 jsa=jsaamt/4.
if jsapd=5 jsa=jsaamt*12/52.
if jsapd=13 jsa=jsaamt/13.
if jsapd=26 jsa=jsaamt/26.
if jsapd=52 jsa=jsaamt/52.

missing values jsaamt jsapd jsa ().
if jsaamt=-8 jsa=-8.
```

```
if jsaamt=-9 jsa=-9.
if jsaamt=-1 jsa=-1.
if jsapd=97 | jsapd=-8 jsa=-8.
execute.
missing values jsaamt jsapd jsa (lo thru -1).
var label jsa "(D) Job Seeker's Allowance per week".
```

SPEN (D) State Retirement Pension per week

```
SPSS Syntax

* State Retirement Pension.
compute spen=0.
if spenpd=1 spen=spenamt.
if spenpd=2 spen=spenamt/2.
if spenpd=4 spen=spenamt/4.
if spenpd=5 spen=spenamt*12/52.
if spenpd=13 spen=spenamt/13.
if spenpd=26 spen=spenamt/26.
if spenpd=52 spen=spenamt/52.

missing values spenamt spenpd spen ().
if spenamt=-8 spen=-8.
if spenamt=-9 spen=-9.
if spenamt=-1 spen=-1.
if spenpd=97 | spenpd=-8 spen=-8.
execute.
missing values spenamt spenpd spen (lo thru -1).
var label spen "(D) State Retirement Pension per week".
```

EPEN (D) Employer's pension per week

```
SPSS Syntax

* Employer's pension.
compute epen=0.
if epenpd=1 epen=epenamt.
if epenpd=2 epen=epenamt/2.
if epenpd=4 epen=epenamt/4.
if epenpd=5 epen=epenamt*12/52.
if epenpd=13 epen=epenamt/13.
if epenpd=26 epen=epenamt/26.
if epenpd=52 epen=epenamt/52.

missing values epenamt epenpd epen ().
if epenamt=-8 epen=-8.
if epenamt=-9 epen=-9.
if epenamt=-1 epen=-1.
if epenpd=97 | epenpd=-8 epen=-8.
execute.
missing values epenamt epenpd epen (lo thru -1).
var label epen "(D) Employer's pension per week".
```

PPEN (D) Private Pension per week

```
SPSS Syntax

* Private pension.
compute ppen=0.
if ppenpd=1 ppen=ppenamt.
if ppenpd=2 ppen=ppenamt/2.
if ppenpd=4 ppen=ppenamt/4.
if ppenpd=5 ppen=ppenamt*12/52.
if ppenpd=13 ppen=ppenamt/13.
if ppenpd=26 ppen=ppenamt/26.
if ppenpd=52 ppen=ppenamt/52.
```

```
missing values ppenamt ppenpd ppen ().
if ppenamt=-8 ppen=-8.
if ppenamt=-9 ppen=-9.
if ppenamt=-1 ppen=-1.
if ppenpd=97 | ppenpd=-8 ppen=-8.
if ppenpd=-8 ppen=-8.
execute.
missing values ppenamt ppenpd ppen (lo thru -1).
var label ppen "(D) Private Pension per week".
```

INCS (D) Income support/pension credit per week

```
SPSS Syntax

* Income support/pension credit.
compute incs=0.
if ispcpd=1 incs=ispcamt.
if ispcpd=2 incs=ispcamt/2.
if ispcpd=4 incs=ispcamt/4.
if ispcpd=5 incs=ispcamt*12/52.
if ispcpd=13 incs=ispcamt/13.
if ispcpd=26 incs=ispcamt/26.
if ispcpd=52 incs=ispcamt/52.

missing values ispcamt ispcpd incs ().
if ispcamt=-8 incs=-8.
if ispcamt=-9 incs=-9.
if ispcamt=-1 incs=-1.
if ispcpd=97 | ispcpd=-8 incs=-8.
execute.
missing values ispcamt ispcpd incs (lo thru -1).
var label incs "(D) Income support/pension credit per week".
```

DLA (D) Disability Living Allowance per week

```
SPSS Syntax

* Disability Living Allowance.
compute dla=0.
if dlapd=1 dla=dlaamt.
if dlapd=2 dla=dlaamt/2.
if dlapd=4 dla=dlaamt/4.
if dlapd=5 dla=dlaamt*12/52.
if dlapd=13 dla=dlaamt/13.
if dlapd=26 dla=dlaamt/26.
if dlapd=52 dla=dlaamt/52.

missing values dlaamt dlapd dla ().
if dlaamt=-8 dla=-8.
if dlaamt=-9 dla=-9.
if dlaamt=-1 dla=-1.
if dlapd=97 | dlapd=-8 dla=-8.
execute.
missing values dlapd dlaamt dla (lo thru -1).
var label dla "(D) Disability Living Allowance per week".
```

INCAP (D) Incapacity Benefit, per week

```
SPSS Syntax

* Incapacity Benefit.
compute incap=0.
if incappd=1 incap=incapamt.
if incappd=2 incap=incapamt/2.
if incappd=4 incap=incapamt/4.
```

```
if incappd=5 incap=incapamt*12/52.
if incappd=13 incap=incapamt/13.
if incappd=26 incap=incapamt/26.
if incappd=52 incap=incapamt/52.

missing values incapamt incappd incap ().
if incapamt=-8 incap=-8.
if incapamt=-9 incap=-9.
if incapamt=-1 incap=-1.
if incappd=97 | incappd=-8 incap=-8.
execute.
missing values incapamt incappd incap (lo thru -1).
var label incap "(D) Incapacity Benefit, per week".
```

ATT (D) Attendance Allowance, per week

```
SPSS Syntax

* Attendance Allowance.
compute att=0.
if attpd=1 att=attamt.
if attpd=2 att=attamt/2.
if attpd=4 att=attamt/4.
if attpd=5 att=attamt*12/52.
if attpd=13 att=attamt/13.
if attpd=26 att=attamt/26.
if attpd=52 att=attamt/52.

missing values attamt attpd att ().
if attamt=-8 att=-8.
if attamt=-9 att=-9.
if attamt=-1 att=-1.
if attpd=97 | attpd=-8 att=-8.
execute.
missing values attamt attpd att (lo thru -1).
var label att "(D) Attendance Allowance, per week".
```

OTHB (D) Other benefits, per week

```
SPSS Syntax

* Other benefits.
compute othb=0.
if othbpd=1 othb=othbamt.
if othbpd=2 othb=othbamt/2.
if othbpd=4 othb=othbamt/4.
if othbpd=5 othb=othbamt*12/52.
if othbpd=13 othb=othbamt/13.
if othbpd=26 othb=othbamt/26.
if othbpd=52 othb=othbamt/52.

missing values othbamt othbpd othb ().
if othbamt=-8 othb=-8.
if othbamt=-9 othb=-9.
if othbamt=-1 othb=-1.
if othbpd=97 | othbpd=-8 othb=-8.
execute.
missing values othbamt othbpd othb (lo thru -1).
var label othb "(D) Other benefits, per week".
```

PARTN (D) Money from former partner, per week

```
SPSS Syntax

* MOney from former or absent spouse or partner.
compute partn=0.
```

```
if partnpd=1 partn=partnamt.
if partnpd=2 partn=partnamt/2.
if partnpd=4 partn=partnamt/4.
if partnpd=5 partn=partnamt*12/52.
if partnpd=13 partn=partnamt/13.
if partnpd=26 partn=partnamt/26.
if partnpd=52 partn=partnamt/52.

missing values partnamt partnpd partn ().
if partnamt=-8 partn=-8.
if partnamt=-9 partn=-9.
if partnamt=-1 partn=-1.
if partnpd=97 | partnpd=-8 partn=-8.
execute.
missing values partnamt partnpd partn (lo thru -1).
var label partn "(D) Money from former partner, per week".
```

GTS (D) Money from Government training schemes, per week

```
SPSS Syntax

* Money from governement training schemes.
compute gts=0.
if gtspd=1 gts=gtsamt.
if gtspd=2 gts=gtsamt/2.
if gtspd=4 gts=gtsamt/4.
if gtspd=5 gts=gtsamt*12/52.
if gtspd=13 gts=gtsamt/13.
if gtspd=26 gts=gtsamt/26.
if gtspd=52 gts=gtsamt/52.

missing values gtsamt gtspd gts ().
if gtsamt=-8 gts=-8.
if gtsamt=-9 gts=-9.
if gtsamt=-1 gts=-1.
if gtspd=97 | gtspd=-8 gts=-8.
execute.
missing values gtsamt gtspd gts (lo thru -1).
var label gts "(D) Money from Government training schemes, per week".
```

EDUCM (D) Educational grant, per week

```
SPSS Syntax

* Educational grant/bursary.
compute educm=0.
if educpd=1 educm=educamt.
if educpd=2 educm=educamt/2.
if educpd=4 educm=educamt/4.
if educpd=5 educm=educamt*12/52.
if educpd=13 educm=educamt/13.
if educpd=26 educm=educamt/26.
if educpd=52 educm=educamt/52.

missing values educamt educpd educm ().
if educamt=-8 educm=-8.
if educamt=-9 educm=-9.
if educamt=-1 educm=-1.
if educpd=97 | educpd=-8 educm=-8.
execute.
missing values educamt educpd educm (lo thru -1).
var label educm "(D) Educational grant, per week".
```

INT (D) Interest from savings, per week

```
SPSS Syntax

* Interest from savings.
compute int=0.
if intpd=1 int=intamt.
if intpd=2 int=intamt/2.
if intpd=4 int=intamt/4.
if intpd=5 int=intamt*12/52.
if intpd=13 int=intamt/13.
if intpd=26 int=intamt/26.
if intpd=52 int=intamt/52.

missing values intamt intpd int ().
if intamt=-8 int=-8.
if intamt=-9 int=-9.
if intamt=-1 int=-1.
if intpd=97 | intpd=-8 int=-8.
execute.
missing values intamt intpd int (lo thru -1).
var label int "(D) Interest from savings, per week".
```

RENT (D) Rent from property/subletting

```
SPSS Syntax

* Rent.
compute rent=0.
if rentpd=1 rent=rentamt.
if rentpd=2 rent=rentamt/2.
if rentpd=4 rent=rentamt/4.
if rentpd=5 rent=rentamt*12/52.
if rentpd=13 rent=rentamt/13.
if rentpd=26 rent=rentamt/26.
if rentpd=52 rent=rentamt/52.

missing values rentamt rentpd rent ().
if rentamt=-8 rent=-8.
if rentamt=-9 rent=-9.
if rentamt=-1 rent=-1.
if rentpd=97 | rentpd=-8 rent=-8.
execute.
missing values rentamt rentpd rent (lo thru -1).
var label rent "(D) Rent from property/subletting".
```

OREG (D) Other regular income

```
SPSS Syntax

* Other regular income.
compute oreg=0.
if oregpd=1 oreg=oregamt.
if oregpd=2 oreg=oregamt/2.
if oregpd=4 oreg=oregamt/4.
if oregpd=5 oreg=oregamt*12/52.
if oregpd=13 oreg=oregamt/13.
if oregpd=26 oreg=oregamt/26.
if oregpd=52 oreg=oregamt/52.

missing values oregamt oregpd oreg ().
if oregamt=-8 oreg=-8.
if oregamt=-9 oreg=-9.
if oregamt=-1 oreg=-1.
if oregpd=97 | oregpd=-8 oreg=-8.
execute.
missing values oregamt oregpd oreg (lo thru -1).
var label oreg "(D) Other regular income".
```

HB (D) Housing benefit per week

```
SPSS Syntax

* Housing benefit.
compute hb=0.
if hbpd=1 hb=hbamt.
if hbpd=2 hb=hbamt/2.
if hbpd=4 hb=hbamt/4.
if hbpd=5 hb=hbamt*12/52.
if hbpd=13 hb=hbamt/13.
if hbpd=26 hb=hbamt/26.
if hbpd=52 hb=hbamt/52.

missing values hbamt hbpd hb ().
if hbamt=-8 hb=-8.
if hbamt=-9 hb=-9.
if hbamt=-1 hb=-1.
if hbpd=97 | hbpd=-8 hb=-8.
execute.
missing values hbamt hbpd hb (lo thru -1).
var label hb "(D) Housing benefit per week".
```

NETINC1 (D) Hhold's net income from all sources pw DK for some sources=0

```
SPSS Syntax

* Calculating total household's net income per week.
* Will create several derived variables.

* NETINC1 - summarises all income, treats missing values as zeros.
missing values earn cben incs wtc ctc jsa spen epen ppen dla incap att othb partn gts
educm int rent oreg hb (-9 thru -1).
compute netinc1=sum(earn, cben, incs, wtc, ctc, jsa, spen, epen, ppen, dla, incap,
att, othb, partn, gts, educm, int, rent, oreg, hb).
recode netinc1 (0=-8) (sysmis=-8) (else=copy).
var label netinc1 "(D) Hhold's net income from all sources p.w., DK for some
sources=0".
value label netinc1
-8 'Dont know'.
execute.
missing value netinc1 (-8).
```

NETINC2 (D) Hhold's net income pw excl. DLA, Incap. ben. and AA, DK for some sources=0

```
SPSS Syntax

* NETINC2 - the same principle but DLA, Incap. ben and Attendance Allowance are not
included.
compute netinc2=sum(earn, cben, incs, wtc, ctc, jsa, spen, epen, ppen, othb, partn,
gts, educm, int, rent, oreg, hb).
recode netinc2 (0=-8) (sysmis=-8) (else=copy).
var label netinc2 "(D) Hhold's net income pw excl. DLA, Incap. ben. and AA, DK for
some sources=0".
value label netinc2
-8 'Dont know'.
execute.
missing values netinc2 (-8).

* if only income from DLA, Incap. ben. of AA, then netinc2=0.
missing values dla incap att netinc1 netinc2 ().
if (netinc2=-8 & (dla>0 | incap>0 | att>0)) netinc2=0.
execute.
missing values dla incap att netinc1 netinc2 (-8).
```

NETINC3 (D) Hhold's net income from all sources pw (valid values only)

```
SPSS Syntax

* NETINC3 - if any of the components are unknown (-8 or -9 but not -1), total income
is missing; income from all sources.
missing values earn cben incs wtc ctc jsa spen epen ppen dla incap att othb partn gts
educm int rent oreg hb netinc1 ().
compute netinc3=netinc1.
if any (-8, -9, earn, cben, incs, wtc, ctc, jsa, spen, epen, ppen, dla, incap, att,
othb, partn, gts, educm, int, rent, oreg, hb) netinc3=-8.
execute.
missing values earn cben incs wtc ctc jsa spen epen ppen dla incap att othb partn gts
educm int rent oreg hb netinc1 netinc3 (lo thru -1).
var label netinc3 "(D) Hhold's net income from all sources per week (valid values
only)".
value label netinc3
-8 'Dont know'.
```

NETINC4 (D) Hhold's net income pw exc. DLA, Incap. ben. & AA (valid values only)

```
SPSS Syntax

* NETINC4 - the same as netinc3 (i.e. includes valid values only) but does not include
DLA, Incap. ben. and AA.
missing values earn cben incs wtc ctc jsa spen epen ppen othb partn gts educm int rent
oreg hb netinc2 ().
compute netinc4=netinc2.
if any (-8, -9, earn, cben, incs, wtc, ctc, jsa, spen, epen, ppen, othb, partn, gts,
educm, int, rent, oreg, hb) netinc4=-8.
execute.
missing values earn cben incs wtc ctc jsa spen epen ppen othb partn gts educm int rent
oreg hb netinc2 netinc4 (lo thru -1).
var label netinc4 "(D) Hhold's net income pw excl. DLA, Incap. ben. and AA (valid
values only)".
value label netinc4
-8 'Dont know'.
```

MARRIED (D) Number of married/cohabiting people in Hhold

```
SPSS Syntax

*** Number of married people in the household.
*** We count those married, cohabiting and the same sex couples.
COUNT
  married = gdvmardf gdvmard2 gdvmard3 gdvmard4 gdvmard5 gdvmard6 gdvmard7
  gdvmard8 gdvmard9 gdvmar10  (1)  gdvmardf gdvmard2 gdvmard3 gdvmard4
  gdvmard5 gdvmard6 gdvmard7 gdvmard8 gdvmard9 gdvmar10  (2)  gdvmardf
  gdvmard2 gdvmard3 gdvmard4 gdvmard5 gdvmard6 gdvmard7 gdvmard8 gdvmard9
  gdvmar10  (7)  .
VARIABLE LABELS married '(D) Number of married/cohabiting people in Hhold' .
EXECUTE .

* 6 households include one married person only rather than a couple.
* Decided to treat them as unmarried.
if shserial=1037160 married=0.
if shserial=1011326 married=0.
if shserial=1135469 married=0.
if shserial=1128709 married=0.
if shserial=1075454 married=0.
if shserial=1070520 married=0.
execute.
```

MCCLEM (D) McClements equivalence score

```
SPSS Syntax

* The variables for everyone's age must be consecutive in the file (as a requirement
of the vector command).

*** Counting all ADULTS (i.e. 19+).
compute adults=0.
VECTOR adult=gdvage to gdvage10.
LOOP xxi=1 to 10.
if (range(adult(xxi), 19,150)) adults=adults+1.
end loop.

*** Set McClements score to 0.
compute mcclem=0.

*** Add scores for adults.
IF (adults=1) mcclem=mcclem+(61/100).
IF (adults=2) mcclem=mcclem+1.
IF (adults=3) mcclem=mcclem+(142/100).
IF (adults>=4) mcclem=mcclem+((142+(36*(adults-3)))/100).
IF (married=0&adults>1) mcclem=mcclem+(7/100).

*** Add scores for children (0-18).
*** The HSE distinguish between those aged 16-18 and in full-time education (i.e.
children) and others (i.e. adults).
*** ONS quidelines do not suggest such a distinction; so went with ONS rather than
HSE.

VECTOR child=gdvage to gdvage10.
LOOP xxj=1 to 10.
if (range(child(xxj),0,1)) mcclem=mcclem+0.09.
if (range(child(xxj),2,4)) mcclem=mcclem+0.18.
if (range(child(xxj),5,7)) mcclem=mcclem+0.21.
if (range(child(xxj),8,10)) mcclem=mcclem+0.23.
if (range(child(xxj),11,12)) mcclem=mcclem+0.25.
if (range(child(xxj),13,15)) mcclem=mcclem+0.27.
if (range(child(xxj),16,18)) mcclem=mcclem+0.36.
end loop.

variable label mcclem "(D McClements equivalence score".

* There are four hholds where mcclem=0.36.
* These are single-person households where this person is aged 18 years old.
* Recode McClements score to make it as if they were 19+.
recode mcclem (0.36=0.61) (else=copy).

* There are five hholds where mcclem=0.45.
* These are single-parent households with a parent aged 18 years old and an infant
under 2.
* Recode McClements score to make it as if the parent were 19+.
recode mcclem (0.45=0.70) (else=copy).
* it does not work for some reason.
if mcclem>0.44 & mcclem<0.46 mcclem=0.70.

* There are two hholds where mcclem=0.72.
* These are couples where both partners are aged 17-18 years old.
* Recode them to an adult couple.
recode mcclem (0.72=1) (else=copy).

*** Need to make changes to the following households - problems with household grid
missing some people out.
*** Need to change McClements' score.

* shserial=1126799.
* CAPI says: 42 and 13 year olds only.
* ARF says two adults and two children (second child is aged 2-17).
* If indeed two adults and two children, then mcclem could be (1+0.27+0.18)=1.45 or
(1+0.27+0.36)=1.63.
* Decided to take an average of these two values - 1.54.
if shserial=1126799 mcclem=1.54.
* shserial 1011134.
* a child aged 2-17 missing from the household grid.
* CAPI says there is one adult and one child aged 3 years old in the hhold.
```

```
* mcclem could be (0.61+0.18+0.36)=1.15 or (0.61+0.18+0.18)=0.97.
* Decided to take an average of these two values - 1.06.
if hserial=1011134 mcclem=1.06.
```

EQINC1 (D) Equiv. hhold net income pw, from all sources, DK=0
EQINC2 (D) Equiv. hhold net income pw exc. DLA, IB & AA, DK=0
EQINC3 (D) Equiv. hhold net income pw, from all sources, valid values only
EQINC4 (D) Equiv. hhold net income pw exc. DLA, IB & AA, valid values only

```
SPSS Syntax

*** Computing equivalised income.

compute eqinc1=netinc1/mcclem.
compute eqinc2=netinc2/mcclem.
compute eqinc3=netinc3/mcclem.
compute eqinc4=netinc4/mcclem.

variable label eqinc1 "(D) Equiv. hhold net income pw, from all sources, DK=0".
variable label eqinc2 "(D) Equiv. hhold net income pw excl. DLA, IB and AA, DK=0".
variable label eqinc3 "(D) Equiv. hhold net income pw, from all sources, valid values
only".
variable label eqinc4 "(D) Equiv. hhold net income pw excl. DLA, IB & AA, valid values
only".

recode eqinc1 (sysmis=-8) (else=copy).
recode eqinc2 (sysmis=-8) (else=copy).
recode eqinc3 (sysmis=-8) (else=copy).
recode eqinc4 (sysmis=-8) (else=copy).
execute.
missing values eqinc1 eqinc2 eqinc3 eqinc4 (-8).
value label eqinc1 eqinc2 eqinc3 eqinc4
-8 'Dont know'.
execute.
```

NSSEC7 (D) Nssec grouped

```
1 Higher managerial and professional occupations
2 Lower managerial and professioanl occupations
3 Intermediate occupations
4 Small employers and own account workers
5 Lower supervisory and technical occupations
6 Semi-routine occupations
7 Routine occupations
```

```
SPSS Syntax

**Recode Nssec into nssec7 for tables 3 3 to 3 6.
Recode nssec (-9 thru -1 = copy) (1 thru 3.9 = 1) (4 thru 6.9 = 2) (7 thru 7.9 = 3) (8
thru 9.9 = 4) (10 thru 11.9 = 5) (12 thru 12.9 = 6) (13  thru 13.9 = 7) into nssec7.
VARIABLE LABELS nssec7 "(D) Nssec grouped".
VALUE LABELS nssec7
1 "Higher managerial and professional occupations"
2 "Lower managerial and professioanl occupations"
3 "Intermediate occupations"
4 "Small employers and own account workers"
5 "Lower supervisory and technical occupations"
6 "Semi-routine occupations"
7 "Routine occupations".
Missing values nssec7 (-9 thru -1).
```

QUAL7 (D) Qualifications gained, grouped

1 Degree or equivalent
2 Higher education, below degree level
3 GCE, A level or equivalent
4 GCSE grades A - C or equivalent
5 GCSE grades D-G/Commercial qualifications/apprenticeship
6 Foreign or other qualifications
7 No qualifications
8 Still in FT education

```
SPSS Syntax

**Recode qual into qual7.
RECODE qual (-9 thru -1 = COPY) (1 thru 2 = 1) (3 thru 5 = 2) (6 thru 8 = 3) (9 thru
17 = 4) (18 thru 24 = 5) (26 thru 31 = 6) into qual7.
RECODE qualch (2 = 7) into qual7.
do if agep>=16.
RECODE wrkstat (2=8) into qual7.
end if.
VARIABLE LABELS qual7 "(D) Qualifications gained, grouped".
VALUE LABELS qual7
1 "Degree or equivalent"
2 "Higher education, below degree level"
3 "GCE, A level or equivalent"
4 "GCSE grades A - C or equivalent"
5 "GCSE grades D-G/Commercial qualifications/apprenticeship"
6 "Foreign or other qualifications"
7 "No qualifications"
8 "Still in FT education".
missing values qual7 (-9 thru -1).
```

EDUC2GR (D) Age finished school grouped

1 15 or under
2 16
3 17
4 18 or over
5 still in FT education

```
SPSS Syntax

**Recode for tables 3 15 to 3 17.
***Recode educ2 into grouped ages educ2gr.
RECODE educ2 (-9 thru -1 =copy)(lo thru 15=1)(16 = 2)(17=3)(18 thru hi=4) into
educ2gr.
do if agep >=16.
recode wrkstat (2=5) into educ2gr.
end if.
VARIABLE LABELS educ2gr "(D) Age finished school grouped".
VALUE LABELS educ2gr
1 "15 or under"
2 "16"
3 "17"
4 "18 or over"
5 "still in FT education".
missing values educ2gr (-9 thru -1).
```

GCSEAC (D) Above or below A-C at GCSE

1 Above A-C GCSE
2 Below A-C GCSE
3 Still in FT Education

```
SPSS Syntax

**VARIABLE GCSE A-C AND ABOVE.
```

```
RECODE qual (1 thru 17, 26, 27 = 1) (18 thru 25 = 2) (28 thru hi = 2) (-9 thru -1 =
COPY) into gcseac.
RECODE qual7 (7 = 2) (8 = 3) into gcseac.
VARIABLE LABELS gcseac "(D) Above or below A-C at GCSE".
VALUE LABELS gcseac
1 "Above A-C GCSE"
2 "Below A-C GCSE"
3 "Still in FT Education".
missing values gcseac (-9 thru -1).
```

LIMITILL (D) Whether have a limiting longstanding illness

1 None
2 Limiting
3 Non limiting

```
SPSS Syntax

**Recode heal and limit into limiting illness (limitill) for tables 3 18 and 3 19 Only
includes 16+.
do if agep >=16.
recode limit (-9 thru -1 = COPY) (1 = 2) (2 = 3) into limitill.
recode heal (-9 thru -1 = COPY) ( 2 = 1) into limitill.
end if.
do if agep<16.
recode heal (-9 thru hi = -1) into limitill.
recode limit (-1 = COPY) into limitill.
end if.
VARIABLE LABELS limitill "(D) Whether have a limiting longstanding illness".
VALUE LABELS limitill
1 "None"
2 "Limiting"
3 "Non limiting".
missing values limitill (-9 thru -1).
```

EQ1DEC (D) Eqinc1 deciles

1 Lowest
2 2nd
3 3rd
4 4th
5 5th
6 6th
7 7th
8 8th
9 9th
10 Highest

```
SPSS Syntax

***Creating deciles of eqinc1_1.
**rename decile variable.
rename variable (neqinc1_  = eq1dec).
variable labels eq1dec "(D) Eqinc1 deciles".
Value labels eq1dec
1 "Lowest"
2 "2nd"
3 "3rd"
4 "4th"
5 "5th"
6 "6th"
7 "7th"
8 "8th"
9 "9th"
10 "Highest".
missing values eq1dec (-9 thru -1).
```

TENURE (D) Tenure

1 Own outright
2 Own with mortgage
3 Rent local authority
4 Rent housing association
5 Rent privately, furnished
6 Rent privately, unfurnished

```
SPSS Syntax

***Derive tenure from ten1 llord and furn.
Recode ten1 (-9 thru -1=copy) (0 thru hi=0) into tenure.
Recode llord (-9 thru -1=copy) into tenure.
if ten1 = 1 tenure =1.
if ten1 = 2 tenure =2.
if ten1 = 3 tenure =2.
if ten1 = 4 & llord = 1 tenure = 3.
if ten1 = 4 & llord = 2 tenure = 4.
if (any (ten1, 4, 5) & range (llord, 3, 7) & furn = 1) tenure = 5.
if (any (ten1, 4,5) & range (llord, 3, 7) & furn = 2) tenure = 6.
if (any (ten1, 4,5) & range (llord, 3, 7) & furn = 3) tenure = 6.
VARIABLE LABELS tenure "(D) Tenure".
VALUE LABELS tenure
1 "Own outright"
2 "Own with mortgage"
3 "Rent local authority"
4 "Rent housing association"
5 "Rent privately, furnished"
6 "Rent privately, unfurnished".
missing values tenure (-9 thru -1).
```

PPR (D) Persons per room

```
SPSS Syntax

recode totrooms (0.00 thru hi=0) (lo thru -1 = copy) into ppr.
If (totrooms>=0) ppr = dmhsize/totrooms.
VARIABLE LABEL ppr "(D) Persons per room".
```

PPRG (D) Persons per room grouped

1 under 0.5
2 0.5 to 0.65
3 0.66 to 0.99
4 1
5 over 1

```
SPSS Syntax

Recode ppr (1.01 thru hi=5) (1.00=4) (0.66 thru 0.99=3) (0.50 thru 0.65=2) (0.00 thru
0.49=1) (lo thru -1=copy) into pprg.
VARIABLE LABEL pprg "(D) Persons per room grouped".
VALUE LABELS pprg
1 "under 0.5"
2 "0.5 to 0.65"
3 "0.66 to 0.99"
4 "1"
5 "over 1"
missing values pprg (-9 thru -1).
```

Chapter 4: Foods consumed

SUPGP1 (D) Supplement 1 taken on any day

1 Cod liver oil and other fish based supplements

SUPGP2 'D) Supplement 2 taken on any day

1 Evening primrose oil type supplements

SUPGP3 (D) Supplement 3 taken on any day

1 Vitamin C only

SUPGP4 (D) Supplements 4,5,8 or 10 taken on any day

1 Other vitamins including multivitamins

SUPGP5 (D) Supplements 6 or 9 taken on any day

1 Vitamins with minerals, including iron

SUPGP6 (D) Supplements 7 or 11 taken on any day

1 Minerals only including iron

SUPGP7 (D) Supplement 12 taken on any day

1 Other

```
SPSS Syntax

GET
  FILE='C:\day level data.sav'.


DO IF (dayno = 1) .
RECODE
 supno (1=1) (MISSING=Copy) (else=sysmis)   INTO  supn1d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno (1=1) (MISSING=Copy) (else=sysmis)   INTO  supn1d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno (1=1) (MISSING=Copy) (else=sysmis)  INTO  supn1d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
  supno (1=1) (MISSING=Copy) (else=sysmis)  INTO  supn1d4.
END IF .
EXECUTE .


DO IF (dayno = 1) .
RECODE
  supno2 (2=2) (MISSING=Copy) (else=sysmis)  INTO  supn2d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno2 (2=2) (MISSING=Copy) (else=sysmis)  INTO  supn2d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno2 (2=2) (MISSING=Copy) (else=sysmis)  INTO  supn2d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
```

```
RECODE
  supno2 (2=2) (MISSING=Copy) (else=sysmis)  INTO  supn2d4.
END IF .
EXECUTE .


DO IF (dayno = 1) .
RECODE
  supno3 (3=3) (MISSING=Copy) (else=sysmis)  INTO  supn3d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno3 (3=3) (MISSING=Copy) (else=sysmis)  INTO  supn3d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno3 (3=3) (MISSING=Copy) (else=sysmis)  INTO  supn3d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
  supno3 (3=3) (MISSING=Copy) (else=sysmis)  INTO  supn3d4.
END IF .
EXECUTE .



DO IF (dayno = 1) .
RECODE
  supno4 (4=4) (MISSING=Copy) (else=sysmis)  INTO  supn4d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno4 (4=4) (MISSING=Copy) (else=sysmis)  INTO  supn4d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno4 (4=4) (MISSING=Copy) (else=sysmis)  INTO  supn4d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
  supno4 (4=4) (MISSING=Copy) (else=sysmis)  INTO  supn4d4.
END IF .
EXECUTE .


DO IF (dayno = 1) .
RECODE
  supno5 (5=5) (MISSING=Copy) (else=sysmis)  INTO  supn5d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno5 (5=5) (MISSING=Copy) (else=sysmis)  INTO  supn5d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno5 (5=5) (MISSING=Copy) (else=sysmis)  INTO  supn5d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
```

```
RECODE
  supno5 (5=5) (MISSING=Copy) (else=sysmis)  INTO  supn5d4.
END IF .
EXECUTE .


DO IF (dayno = 1) .
RECODE
  supno6 (6=6) (MISSING=Copy) (else=sysmis)  INTO  supn6d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno6 (6=6) (MISSING=Copy) (else=sysmis)  INTO  supn6d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno6 (6=6) (MISSING=Copy) (else=sysmis)  INTO  supn6d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
  supno6 (6=6) (MISSING=Copy) (else=sysmis)  INTO  supn6d4.
END IF .
EXECUTE .


DO IF (dayno = 1) .
RECODE
  supno7 (7=7) (MISSING=Copy) (else=sysmis)  INTO  supn7d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno7 (7=7) (MISSING=Copy) (else=sysmis)  INTO  supn7d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno7 (7=7) (MISSING=Copy) (else=sysmis)  INTO  supn7d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
  supno7 (7=7) (MISSING=Copy) (else=sysmis)  INTO  supn7d4.
END IF .
EXECUTE .


DO IF (dayno = 1) .
RECODE
  supno8 (8=8) (MISSING=Copy) (else=sysmis)  INTO  supn8d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno8 (8=8) (MISSING=Copy) (else=sysmis)  INTO  supn8d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno8 (8=8) (MISSING=Copy) (else=sysmis)  INTO  supn8d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
```

```
  supno8 (8=8) (MISSING=Copy) (else=sysmis)  INTO  supn8d4.
END IF .
EXECUTE .



DO IF (dayno = 1) .
RECODE
  supno9 (9=9) (MISSING=Copy) (else=sysmis)  INTO  supn9d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno9 (9=9) (MISSING=Copy) (else=sysmis)  INTO  supn9d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno9 (9=9) (MISSING=Copy) (else=sysmis)  INTO  supn9d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
  supno9 (9=9) (MISSING=Copy) (else=sysmis)  INTO  supn9d4.
END IF .
EXECUTE .




DO IF (dayno = 1) .
RECODE
  supno10 (10=10) (MISSING=Copy) (else=sysmis)  INTO  supn10d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno10 (10=10) (MISSING=Copy) (else=sysmis)  INTO  supn10d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno10 (10=10) (MISSING=Copy) (else=sysmis)  INTO  supn10d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
  supno10 (10=10) (MISSING=Copy) (else=sysmis)  INTO  supn10d4.
END IF .
EXECUTE .


DO IF (dayno = 1) .
RECODE
  supno11 (11=11) (MISSING=Copy) (else=sysmis)  INTO  supn11d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno11 (11=11) (MISSING=Copy) (else=sysmis)  INTO  supn11d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno11 (11=11) (MISSING=Copy) (else=sysmis)  INTO  supn11d3.
END IF .
EXECUTE .
```

```
DO IF (dayno = 4) .
RECODE
  supno11 (11=11) (MISSING=Copy) (else=sysmis)  INTO  supn11d4.
END IF .
EXECUTE .


DO IF (dayno = 1) .
RECODE
  supno12 (12=12) (MISSING=Copy) (else=sysmis)  INTO  supn12d1.
END IF .
EXECUTE .

DO IF (dayno = 2) .
RECODE
  supno12 (12=12) (MISSING=Copy) (else=sysmis)  INTO  supn12d2.
END IF .
EXECUTE .

DO IF (dayno = 3) .
RECODE
  supno12 (12=12) (MISSING=Copy) (else=sysmis)  INTO  supn12d3.
END IF .
EXECUTE .

DO IF (dayno = 4) .
RECODE
  supno12 (12=12) (MISSING=Copy) (else=sysmis)  INTO  supn12d4.
END IF .
EXECUTE .



** value and variable labels**.

Variable labels supn1d1 ‘Day 1’.
Variable labels supn1d2 'Day 2'.
Variable labels supn1d3 'Day 3'.
Variable labels supn1d4 'Day 4'.
Variable labels supn2d1 ‘Day 1’.
Variable labels supn2d2 'Day 2'.
Variable labels supn2d3 'Day 3'.
Variable labels supn2d4 'Day 4'.
Variable labels supn3d1 ‘Day 1’.
Variable labels supn3d2 'Day 2'.
Variable labels supn3d3 'Day 3'.
Variable labels supn3d4 'Day 4'.
Variable labels supn4d1 ‘Day 1’.
Variable labels supn4d2 'Day 2'.
Variable labels supn4d3 'Day 3'.
Variable labels supn4d4 'Day 4'.
Variable labels supn5d1 ‘Day 1’.
Variable labels supn5d2 'Day 2'.
Variable labels supn5d3 'Day 3'.
Variable labels supn5d4 'Day 4'.
Variable labels supn6d1 ‘Day 1’.
Variable labels supn6d2 'Day 2'.
Variable labels supn6d3 'Day 3'.
Variable labels supn6d4 'Day 4'.
Variable labels supn7d1 ‘Day 1’.
Variable labels supn7d2 'Day 2'.
Variable labels supn7d3 'Day 3'.
Variable labels supn7d4 'Day 4'.
Variable labels supn8d1 ‘Day 1’.
Variable labels supn8d2 'Day 2'.
Variable labels supn8d3 'Day 3'.
Variable labels supn8d4 'Day 4'.
Variable labels supn9d1 ‘Day 1’.
Variable labels supn9d2 'Day 2'.
Variable labels supn9d3 'Day 3'.
Variable labels supn9d4 'Day 4'.
Variable labels supn10d1 ‘Day 1’.
Variable labels supn10d2 'Day 2'.
Variable labels supn10d3 'Day 3'.
Variable labels supn10d4 'Day 4'.
Variable labels supn11d1 ‘Day 1’.
```

```
Variable labels supn11d2 'Day 2'.
Variable labels supn11d3 'Day 3'.
Variable labels supn11d4 'Day 4'.
Variable labels supn12d1 ‘Day 1’.
Variable labels supn12d2 'Day 2'.
Variable labels supn12d3 'Day 3'.
Variable labels supn12d4 'Day 4'.

Value LABEL supn1d1 supn1d2 supn1d3 supn1d4 1'Cod liver oil and other fish-based
supplements' .
Value LABEL supn2d1 supn2d2 supn2d3 supn2d4 2'Evening primrose oil type supplements'
.
Value LABEL supn3d1 supn3d2 supn3d3 supn3d4 3'Vitamin C only' .
Value LABEL supn4d1 supn4d2 supn4d3 supn4d4 4'Other single vitamins, not vitamin C' .
Value LABEL supn5d1 supn5d2 supn5d3 supn5d4 5'Vitamins A, C and D only' .
Value LABEL supn6d1 supn6d2 supn6d3 supn6d4 6'Vitamins with iron' .
Value LABEL supn7d1 supn7d2 supn7d3 supn7d4 7'Iron only'  .
Value LABEL supn8d1 supn8d2 supn8d3 supn8d4 8'Non-prescribed folic acid only' .
Value LABEL supn9d1 supn9d2 supn9d3 supn9d4 9'Multivitamins and multiminerals' .
Value LABEL supn10d1 supn10d2 supn10d3 supn10d4 10'Multivitamins, no minerals' .
Value LABEL supn11d1 supn11d2 supn11d3 supn11d4 11'Minerals only, not fluoride or iron
only' .
Value LABEL supn12d1 supn12d2 supn12d3 supn12d4 12'Other'.


SORT CASES BY
  shserial (A) serp (A)  .

AGGREGATE
  /OUTFILE='C:\aggr supplement numbers & types.sav'
  /BREAK=shserial serp
/supn1d_1 = MEAN(supn1d1)
/supn1d_2 = MEAN(supn1d2)
/supn1d_3 = MEAN(supn1d3)
/supn1d_4 = MEAN(supn1d4)
/supn2d_1 = MEAN(supn2d1)
/supn2d_2 = MEAN(supn2d2)
/supn2d_3 = MEAN(supn2d3)
/supn2d_4 = MEAN(supn2d4)
/supn3d_1 = MEAN(supn3d1)
/supn3d_2 = MEAN(supn3d2)
/supn3d_3 = MEAN(supn3d3)
/supn3d_4 = MEAN(supn3d4)
/supn4d_1 = MEAN(supn4d1)
/supn4d_2 = MEAN(supn4d2)
/supn4d_3 = MEAN(supn4d3)
/supn4d_4 = MEAN(supn4d4)
/supn5d_1 = MEAN(supn5d1)
/supn5d_2 = MEAN(supn5d2)
/supn5d_3 = MEAN(supn5d3)
/supn5d_4 = MEAN(supn5d4)
/supn6d_1 = MEAN(supn6d1)
/supn6d_2 = MEAN(supn6d2)
/supn6d_3 = MEAN(supn6d3)
/supn6d_4 = MEAN(supn6d4)
/supn7d_1 = MEAN(supn7d1)
/supn7d_2 = MEAN(supn7d2)
/supn7d_3 = MEAN(supn7d3)
/supn7d_4 = MEAN(supn7d4)
/supn8d_1 = MEAN(supn8d1)
/supn8d_2 = MEAN(supn8d2)
/supn8d_3 = MEAN(supn8d3)
/supn8d_4 = MEAN(supn8d4)
/supn9d_1 = MEAN(supn9d1)
/supn9d_2 = MEAN(supn9d2)
/supn9d_3 = MEAN(supn9d3)
/supn9d_4 = MEAN(supn9d4)
/supn10_1 = MEAN(supn10d1)
/supn10_2 = MEAN(supn10d2)
/supn10_3 = MEAN(supn10d3)
/supn10_4 = MEAN(supn10d4)
/supn11_1 = MEAN(supn11d1)
/supn11_2 = MEAN(supn11d2)
/supn11_3 = MEAN(supn11d3)
/supn11_4 = MEAN(supn11d4)
/supn12_1 = MEAN(supn12d1)
```

```
/supn12_2 = MEAN(supn12d2)
/supn12_3 = MEAN(supn12d3)
/supn12_4 = MEAN(supn12d4).


SAVE OUTFILE='C:\day level data.sav'
  /COMPRESSED.

GET
  FILE='C:\aggr supplement numbers & types.sav'

sort cases by shserial serp.

SAVE OUTFILE='C:\aggr supplement numbers & types.sav'
  /COMPRESSED.



GET
  FILE='C:\person level nutrient and food.sav'.

sort cases by shserial serp.

MATCH FILES /FILE=*
 /TABLE='C:\aggr supplement numbers & types.sav'
 /BY shserial serp.
EXECUTE.

Variable labels supn1d_1 'Day 1'.
Variable labels supn1d_2 'Day 2'.
Variable labels supn1d_3 'Day 3'.
Variable labels supn1d_4 'Day 4'.
Variable labels supn2d_1 'Day 1'.
Variable labels supn2d_2 'Day 2'.
Variable labels supn2d_3 'Day 3'.
Variable labels supn2d_4 'Day 4'.
Variable labels supn3d_1 'Day 1'.
Variable labels supn3d_2 'Day 2'.
Variable labels supn3d_3 'Day 3'.
Variable labels supn3d_4 'Day 4'.
Variable labels supn4d_1 'Day 1'.
Variable labels supn4d_2 'Day 2'.
Variable labels supn4d_3 'Day 3'.
Variable labels supn4d_4 'Day 4'.
Variable labels supn5d_1 'Day 1'.
Variable labels supn5d_2 'Day 2'.
Variable labels supn5d_3 'Day 3'.
Variable labels supn5d_4 'Day 4'.
Variable labels supn6d_1 'Day 1'.
Variable labels supn6d_2 'Day 2'.
Variable labels supn6d_3 'Day 3'.
Variable labels supn6d_4 'Day 4'.
Variable labels supn7d_1 'Day 1'.
Variable labels supn7d_2 'Day 2'.
Variable labels supn7d_3 'Day 3'.
Variable labels supn7d_4 'Day 4'.
Variable labels supn8d_1 'Day 1'.
Variable labels supn8d_2 'Day 2'.
Variable labels supn8d_3 'Day 3'.
Variable labels supn8d_4 'Day 4'.
Variable labels supn9d_1 'Day 1'.
Variable labels supn9d_2 'Day 2'.
Variable labels supn9d_3 'Day 3'.
Variable labels supn9d_4 'Day 4'.
Variable labels supn10_1 'Day 1'.
Variable labels supn10_2 'Day 2'.
Variable labels supn10_3 'Day 3'.
Variable labels supn10_4 'Day 4'.
Variable labels supn11_1 'Day 1'.
Variable labels supn11_2 'Day 2'.
Variable labels supn11_3 'Day 3'.
Variable labels supn11_4 'Day 4'.
Variable labels supn12_1 'Day 1'.
Variable labels supn12_2 'Day 2'.
Variable labels supn12_3 'Day 3'.
Variable labels supn12_4 'Day 4'.
```

```
Value LABEL supn1d_1 supn1d_2 supn1d_3 supn1d_4 1'Cod liver oil and other fish-based
supplements' .
Value LABEL supn2d_1 supn2d_2 supn2d_3 supn2d_4 2'Evening primrose oil type
supplements'  .
Value LABEL supn3d_1 supn3d_2 supn3d_3 supn3d_4 3'Vitamin C only' .
Value LABEL supn4d_1 supn4d_2 supn4d_3 supn4d_4 4'Other single vitamins, not vitamin
C' .
Value LABEL supn5d_1 supn5d_2 supn5d_3 supn5d_4 5'Vitamins A, C and D only' .
Value LABEL supn6d_1 supn6d_2 supn6d_3 supn6d_4 6'Vitamins with iron' .
Value LABEL supn7d_1 supn7d_2 supn7d_3 supn7d_4 7'Iron only'  .
Value LABEL supn8d_1 supn8d_2 supn8d_3 supn8d_4 8'Non-prescribed folic acid only' .
Value LABEL supn9d_1 supn9d_2 supn9d_3 supn9d_4 9'Multivitamins and multiminerals' .
Value LABEL supn10_1 supn10_2 supn10_3 supn10_4 10'Multivitamins, no minerals' .
Value LABEL supn11_1 supn11_2 supn11_3 supn11_4 11'Minerals only, not fluoride or iron
only' .
Value LABEL supn12_1 supn12_2 supn12_3 supn12_4 12'Other'.


compute sup1tot=0.
If (supn1d_1=1 or supn1d_2=1 or supn1d_3 =1 or supn1d_4 =1) sup1tot=1.
compute sup2tot=0.
If (supn2d_1=2 or supn2d_2=2 or supn2d_3 =2 or supn2d_4 =2) sup2tot=2.
compute sup3tot=0.
If (supn3d_1=3 or supn3d_2=3 or supn3d_3 =3 or supn3d_4 =3) sup3tot=3.
compute sup4tot=0.
If (supn4d_1=4 or supn4d_2=4 or supn4d_3 =4 or supn4d_4 =4) sup4tot=4.
compute sup5tot=0.
If (supn5d_1=5 or supn5d_2=5 or supn5d_3 =5 or supn5d_4 =5) sup5tot=5.
compute sup6tot=0.
If (supn6d_1=6 or supn6d_2=6 or supn6d_3 =6 or supn6d_4 =6) sup6tot=6.
compute sup7tot=0.
If (supn7d_1=7 or supn7d_2=7 or supn7d_3 =7 or supn7d_4 =7) sup7tot=7.
compute sup8tot=0.
If (supn8d_1=8 or supn8d_2=8 or supn8d_3 =8 or supn8d_4 =8) sup8tot=8.
compute sup9tot=0.
If (supn9d_1=9 or supn9d_2=9 or supn9d_3 =9 or supn9d_4 =9) sup9tot=9.
compute sup10tot=0.
If (supn10_1=10 or supn10_2=10 or supn10_3 =10 or supn10_4 =10) sup10tot=10.
compute sup11tot=0.
If (supn11_1=11 or supn11_2=11 or supn11_3 =11 or supn11_4 =11) sup11tot=11.
compute sup12tot=0.
If (supn12_1=12 or supn12_2=12 or supn12_3 =12 or supn12_4 =12) sup12tot=12.
execute.

Var label sup1tot 'Supplement 1 taken on any day'.
Var label sup2tot 'Supplement 2 taken on any day'.
Var label sup3tot 'Supplement 3 taken on any day'.
Var label sup4tot 'Supplement 4 taken on any day'.
Var label sup5tot 'Supplement 5 taken on any day'.
Var label sup6tot 'Supplement 6 taken on any day'.
Var label sup7tot 'Supplement 7 taken on any day'.
Var label sup8tot 'Supplement 8 taken on any day'.
Var label sup9tot 'Supplement 9 taken on any day'.
Var label sup10tot 'Supplement 10 taken on any day'.
Var label sup11tot 'Supplement 11 taken on any day'.
Var label sup12tot 'Supplement 12 taken on any day'.

value label sup1tot 1'Cod liver oil and other fish-based supplements'.
value label sup2tot 2'Evening primrose oil type supplements' .
value label sup3tot 3'Vitamin C only'.
value label sup4tot 4'Other single vitamins, not vitamin C' .
value label sup5tot 5'Vitamins A, C and D only'.
value label sup6tot 6'Vitamins with iron'.
value label sup7tot  7'Iron only'  .
value label sup8tot  8 'Non-prescribed folic acid only' .
value label sup9tot  9'Multivitamins and multiminerals' .
value label sup10tot 10'Multivitamins, no minerals' .
value label sup11tot 11'Minerals only, not fluoride or iron only' .
value label sup12tot 12'Other'.


compute suptot2=0.
If (sup1tot=1 or sup2tot=2 or sup3tot=3 or sup4tot =4 or sup5tot=5 or sup6tot=6 or
sup7tot=7 or sup8tot=8 or sup9tot=9
or sup10tot=10 or sup11tot=11 or sup12tot=12) suptot2=1.
execute.
```

```
Var label suptot2 'Supplement taker on any day'.
Value label suptot2 0'Not taking supplements' 1'Supplement taker on 1-4days'.


compute Supgp1=0.
IF (sup1tot = 1) Supgp1 = 1 .
VARIABLE LABELS Supgp1 '(D) Supplement 1 taken on any day' .
Value label Supgp1 1'Cod liver oil and other fish based supplements'.
EXECUTE .

compute Supgp2=0.
IF (sup2tot = 2) Supgp2 = 1 .
VARIABLE LABELS Supgp2 '(D) Supplement 2 taken on any day' .
Value label Supgp2 1'Evening primrose oil type supplements'.
EXECUTE .

compute Supgp3=0.
IF (sup3tot = 3) Supgp3 = 1 .
VARIABLE LABELS Supgp3 '(D) Supplement 3 taken on any day' .
Value label Supgp3 1'Vitamin C only'.
EXECUTE .

compute Supgp4=0.
IF (sup4tot = 4 or sup5tot = 5 or sup8tot = 8 or sup10tot =10) Supgp4 = 1 .
VARIABLE LABELS Supgp4 '(D) Supplements 4,5,8 or 10 taken on any day' .
Value label Supgp4 1'Other vitamins including multivitamins'.
EXECUTE .

compute Supgp5=0.
IF (sup6tot = 6 or sup9tot = 9) Supgp5 = 1 .
VARIABLE LABELS Supgp5 '(D) Supplements 6 or 9 taken on any day' .
Value label Supgp5 1'Vitamins with minerals, including iron'.
EXECUTE .

compute Supgp6=0.
IF (sup7tot = 7 or sup11tot = 11) Supgp6 = 1 .
VARIABLE LABELS Supgp6 '(D) Supplements 7 or 11 taken on any day' .
Value label Supgp6 1'Minerals only including iron'.
EXECUTE .

compute Supgp7=0.
IF (sup12tot = 12) Supgp7 = 1 .
VARIABLE LABELS Supgp7 '(D) Supplement 12 taken on any day' .
Value label Supgp7 1'Other'.
EXECUTE .
```

FRTPORT (D) Fruit and fruit juice portions, juice counts once

```
GET
  FILE='C:\Food level data.sav’.


if (foodcode = 935 & foodgp ="23R") fg4 = 44.
if (foodcode = 936 & foodgp ="23R") fg4 = 44.
if (foodcode = 938 & foodgp ="23R") fg4 = 44.
if (foodcode = 939 & foodgp ="23R") fg4 = 44.
if (foodcode = 941 & foodgp ="23R") fg4 = 44.
if (foodcode = 942 & foodgp ="23R") fg4 = 44.
if (foodcode = 943 & foodgp ="23R") fg4 = 44.
if (foodcode = 944 & foodgp ="23R") fg4 = 44.
if (foodcode = 945 & foodgp ="23R") fg4 = 45.
if (foodcode = 946 & foodgp ="23R") fg4 = 45.
if (foodcode = 947 & foodgp ="23R") fg4 = 45.
if (foodcode = 948 & foodgp ="23R") fg4 = 45.
if (foodcode = 950 & foodgp ="23R") fg4 = 44.
if (foodcode = 951 & foodgp ="23R") fg4 = 44.
if (foodcode = 952 & foodgp ="23R") fg4 = 44.
if (foodcode = 953 & foodgp ="23R") fg4 = 44.
if (foodcode = 955 & foodgp ="23R") fg4 = 44.
if (foodcode = 957 & foodgp ="23R") fg4 = 44.
if (foodcode = 958 & foodgp ="23R") fg4 = 44.
if (foodcode = 960 & foodgp ="23R") fg4 = 44.
```

```
if (foodcode = 961 & foodgp ="23R") fg4 = 45.
if (foodcode = 962 & foodgp ="23R") fg4 = 45.
if (foodcode = 963 & foodgp ="23R") fg4 = 45.
if (foodcode = 967 & foodgp ="23R") fg4 = 44.
if (foodcode = 969 & foodgp ="23R") fg4 = 44.
if (foodcode = 970 & foodgp ="23R") fg4 = 44.
if (foodcode = 971 & foodgp ="23R") fg4 = 44.
if (foodcode = 1051 & foodgp ="23R") fg4 = 44.
if (foodcode = 1231 & foodgp ="23R") fg4 = 44.
if (foodcode = 1233 & foodgp ="23R") fg4 = 44.
if (foodcode = 1243 & foodgp ="23R") fg4 = 44.
if (foodcode = 1244 & foodgp ="23R") fg4 = 44.
if (foodcode = 1246 & foodgp ="23R") fg4 = 44.
if (foodcode = 1317 & foodgp ="23R") fg4 = 45.
if (foodcode = 1318 & foodgp ="23R") fg4 = 44.
if (foodcode = 1319 & foodgp ="23R") fg4 = 45.
if (foodcode = 1320 & foodgp ="23R") fg4 = 45.
if (foodcode = 1321 & foodgp ="23R") fg4 = 44.
if (foodcode = 1323 & foodgp ="23R") fg4 = 45.
if (foodcode = 1324 & foodgp ="23R") fg4 = 44.
if (foodcode = 1325 & foodgp ="23R") fg4 = 45.
if (foodcode = 1329 & foodgp ="23R") fg4 = 45.
if (foodcode = 1347 & foodgp ="23R") fg4 = 45.
if (foodcode = 1348 & foodgp ="23R") fg4 = 45.
if (foodcode = 1356 & foodgp ="23R") fg4 = 44.
if (foodcode = 1357 & foodgp ="23R") fg4 = 45.
if (foodcode = 1365 & foodgp ="23R") fg4 = 45.
if (foodcode = 2663 & foodgp ="23R") fg4 = 44.
if (foodcode = 2716 & foodgp ="23R") fg4 = 44.
if (foodcode = 2719 & foodgp ="23R") fg4 = 45.
if (foodcode = 2759 & foodgp ="23R") fg4 = 45.
if (foodcode = 2767 & foodgp ="23R") fg4 = 45.
if (foodcode = 2770 & foodgp ="23R") fg4 = 45.
if (foodcode = 2791 & foodgp ="23R") fg4 = 45.
if (foodcode = 2792 & foodgp ="23R") fg4 = 45.
if (foodcode = 2810 & foodgp ="23R") fg4 = 45.
if (foodcode = 2857 & foodgp ="23R") fg4 = 45.
if (foodcode = 2861 & foodgp ="23R") fg4 = 45.
if (foodcode = 2866 & foodgp ="23R") fg4 = 45.
if (foodcode = 2881 & foodgp ="23R") fg4 = 45.
if (foodcode = 2884 & foodgp ="23R") fg4 = 45.
if (foodcode = 2908 & foodgp ="23R") fg4 = 45.
if (foodcode = 2939 & foodgp ="23R") fg4 = 45.
if (foodcode = 2989 & foodgp ="23R") fg4 = 44.
if (foodcode = 3005 & foodgp ="23R") fg4 = 45.
if (foodcode = 3011 & foodgp ="23R") fg4 = 44.
if (foodcode = 3029 & foodgp ="23R") fg4 = 44.
if (foodcode = 3087 & foodgp ="23R") fg4 = 45.
if (foodcode = 3142 & foodgp ="23R") fg4 = 45.
if (foodcode = 3187 & foodgp ="23R") fg4 = 45.
if (foodcode = 3278 & foodgp ="23R") fg4 = 44.
if (foodcode = 3281 & foodgp ="23R") fg4 = 45.
if (foodcode = 3296 & foodgp ="23R") fg4 = 45.
if (foodcode = 3297 & foodgp ="23R") fg4 = 45.
if (foodcode = 3298 & foodgp ="23R") fg4 = 45.
if (foodcode = 3302 & foodgp ="23R") fg4 = 45.
if (foodcode = 3309 & foodgp ="23R") fg4 = 45.
if (foodcode = 3386 & foodgp ="23R") fg4 = 45.
if (foodcode = 3388 & foodgp ="23R") fg4 = 45.
if (foodcode = 3397 & foodgp ="23R") fg4 = 45.
if (foodcode = 3479 & foodgp ="23R") fg4 = 45.
if (foodcode = 3494 & foodgp ="23R") fg4 = 45.
if (foodcode = 3608 & foodgp ="23R") fg4 = 45.
if (foodcode = 3698 & foodgp ="23R") fg4 = 45.
if (foodcode = 3702 & foodgp ="23R") fg4 = 45.
if (foodcode = 3703 & foodgp ="23R") fg4 = 45.
if (foodcode = 3756 & foodgp ="23R") fg4 = 44.
if (foodcode = 3814 & foodgp ="23R") fg4 = 45.
if (foodcode = 3816 & foodgp ="23R") fg4 = 45.
if (foodcode = 3821 & foodgp ="23R") fg4 = 44.
if (foodcode = 3827 & foodgp ="23R") fg4 = 45.
if (foodcode = 3829 & foodgp ="23R") fg4 = 44.
if (foodcode = 3835 & foodgp ="23R") fg4 = 44.
if (foodcode = 3845 & foodgp ="23R") fg4 = 45.
if (foodcode = 3864 & foodgp ="23R") fg4 = 45.
if (foodcode = 3906 & foodgp ="23R") fg4 = 45.
```

```
if (foodcode = 3909 & foodgp ="23R") fg4 = 45.
if (foodcode = 3913 & foodgp ="23R") fg4 = 45.
if (foodcode = 3916 & foodgp ="23R") fg4 = 44.
if (foodcode = 3926 & foodgp ="23R") fg4 = 45.
if (foodcode = 3936 & foodgp ="23R") fg4 = 45.
if (foodcode = 3947 & foodgp ="23R") fg4 = 45.
if (foodcode = 4031 & foodgp ="23R") fg4 = 45.
if (foodcode = 4047 & foodgp ="23R") fg4 = 45.
if (foodcode = 4054 & foodgp ="23R") fg4 = 44.
if (foodcode = 4062 & foodgp ="23R") fg4 = 45.
if (foodcode = 4075 & foodgp ="23R") fg4 = 45.
if (foodcode = 4077 & foodgp ="23R") fg4 = 45.
if (foodcode = 4080 & foodgp ="23R") fg4 = 45.
if (foodcode = 4095 & foodgp ="23R") fg4 = 45.
if (foodcode = 4096 & foodgp ="23R") fg4 = 45.
if (foodcode = 4099 & foodgp ="23R") fg4 = 45.
if (foodcode = 4102 & foodgp ="23R") fg4 = 45.
if (foodcode = 4116 & foodgp ="23R") fg4 = 45.
if (foodcode = 4124 & foodgp ="23R") fg4 = 45.
if (foodcode = 4125 & foodgp ="23R") fg4 = 45.
if (foodcode = 4129 & foodgp ="23R") fg4 = 45.
if (foodcode = 4130 & foodgp ="23R") fg4 = 44.
if (foodcode = 4132 & foodgp ="23R") fg4 = 45.
if (foodcode = 4137 & foodgp ="23R") fg4 = 45.
if (foodcode = 4139 & foodgp ="23R") fg4 = 44.
if (foodcode = 4144 & foodgp ="23R") fg4 = 45.
if (foodcode = 4149 & foodgp ="23R") fg4 = 45.
if (foodcode = 4161 & foodgp ="23R") fg4 = 45.
if (foodcode = 4165 & foodgp ="23R") fg4 = 45.
if (foodcode = 4167 & foodgp ="23R") fg4 = 45.
if (foodcode = 4169 & foodgp ="23R") fg4 = 45.
if (foodcode = 4172 & foodgp ="23R") fg4 = 45.
if (foodcode = 4181 & foodgp ="23R") fg4 = 45.
if (foodcode = 4187 & foodgp ="23R") fg4 = 45.
if (foodcode = 4191 & foodgp ="23R") fg4 = 45.
if (foodcode = 4210 & foodgp ="23R") fg4 = 44.
if (foodcode = 4228 & foodgp ="23R") fg4 = 45.
if (foodcode = 4229 & foodgp ="23R") fg4 = 44.
if (foodcode = 4231 & foodgp ="23R") fg4 = 45.
if (foodcode = 4236 & foodgp ="23R") fg4 = 45.
if (foodcode = 4245 & foodgp ="23R") fg4 = 45.
if (foodcode = 4246 & foodgp ="23R") fg4 = 45.
if (foodcode = 4251 & foodgp ="23R") fg4 = 45.
if (foodcode = 4262 & foodgp ="23R") fg4 = 45.
if (foodcode = 4264 & foodgp ="23R") fg4 = 45.
if (foodcode = 4265 & foodgp ="23R") fg4 = 45.
if (foodcode = 4267 & foodgp ="23R") fg4 = 45.
if (foodcode = 4269 & foodgp ="23R") fg4 = 44.
if (foodcode = 4271 & foodgp ="23R") fg4 = 45.
if (foodcode = 4277 & foodgp ="23R") fg4 = 45.
if (foodcode = 4280 & foodgp ="23R") fg4 = 45.
if (foodcode = 4283 & foodgp ="23R") fg4 = 45.
if (foodcode = 4284 & foodgp ="23R") fg4 = 45.
if (foodcode = 4285 & foodgp ="23R") fg4 = 45.
if (foodcode = 4286 & foodgp ="23R") fg4 = 45.
if (foodcode = 4287 & foodgp ="23R") fg4 = 45.
if (foodcode = 4288 & foodgp ="23R") fg4 = 45.
if (foodcode = 4300 & foodgp ="23R") fg4 = 45.
if (foodcode = 4301 & foodgp ="23R") fg4 = 44.
if (foodcode = 4302 & foodgp ="23R") fg4 = 45.
if (foodcode = 4303 & foodgp ="23R") fg4 = 45.
if (foodcode = 4307 & foodgp ="23R") fg4 = 45.
if (foodcode = 4318 & foodgp ="23R") fg4 = 45.
if (foodcode = 4321 & foodgp ="23R") fg4 = 45.
if (foodcode = 4323 & foodgp ="23R") fg4 = 45.
if (foodcode = 4325 & foodgp ="23R") fg4 = 45.
if (foodcode = 4327 & foodgp ="23R") fg4 = 45.
if (foodcode = 4330 & foodgp ="23R") fg4 = 45.
if (foodcode = 4334 & foodgp ="23R") fg4 = 45.
if (foodcode = 4339 & foodgp ="23R") fg4 = 45.
if (foodcode = 4343 & foodgp ="23R") fg4 = 45.
if (foodcode = 4346 & foodgp ="23R") fg4 = 45.
if (foodcode = 4349 & foodgp ="23R") fg4 = 45.
if (foodcode = 4367 & foodgp ="23R") fg4 = 45.
if (foodcode = 4368 & foodgp ="23R") fg4 = 45.
if (foodcode = 4372 & foodgp ="23R") fg4 = 45.
```

```
if (foodcode = 4373 & foodgp ="23R") fg4 = 45.
if (foodcode = 4376 & foodgp ="23R") fg4 = 45.
if (foodcode = 4379 & foodgp ="23R") fg4 = 45.
if (foodcode = 4380 & foodgp ="23R") fg4 = 45.
if (foodcode = 4390 & foodgp ="23R") fg4 = 45.
if (foodcode = 4394 & foodgp ="23R") fg4 = 45.
if (foodcode = 4396 & foodgp ="23R") fg4 = 45.
if (foodcode = 4400 & foodgp ="23R") fg4 = 45.
if (foodcode = 4401 & foodgp ="23R") fg4 = 44.
if (foodcode = 4403 & foodgp ="23R") fg4 = 45.
if (foodcode = 4410 & foodgp ="23R") fg4 = 45.
if (foodcode = 4412 & foodgp ="23R") fg4 = 45.
if (foodcode = 4415 & foodgp ="23R") fg4 = 45.
if (foodcode = 4417 & foodgp ="23R") fg4 = 45.
if (foodcode = 4418 & foodgp ="23R") fg4 = 45.
if (foodcode = 4419 & foodgp ="23R") fg4 = 45.
if (foodcode = 4428 & foodgp ="23R") fg4 = 45.
if (foodcode = 5012 & foodgp ="23R") fg4 = 45.
if (foodcode = 5013 & foodgp ="23R") fg4 = 45.
if (foodcode = 5016 & foodgp ="23R") fg4 = 45.
if (foodcode = 5027 & foodgp ="23R") fg4 = 45.
if (foodcode = 5036 & foodgp ="23R") fg4 = 45.
if (foodcode = 5038 & foodgp ="23R") fg4 = 45.
if (foodcode = 5046 & foodgp ="23R") fg4 = 45.
if (foodcode = 5052 & foodgp ="23R") fg4 = 45.
if (foodcode = 5053 & foodgp ="23R") fg4 = 45.
if (foodcode = 5054 & foodgp ="23R") fg4 = 45.
if (foodcode = 5056 & foodgp ="23R") fg4 = 45.
if (foodcode = 5058 & foodgp ="23R") fg4 = 45.
if (foodcode = 5063 & foodgp ="23R") fg4 = 45.
if (foodcode = 5064 & foodgp ="23R") fg4 = 45.
if (foodcode = 5069 & foodgp ="23R") fg4 = 45.
if (foodcode = 5072 & foodgp ="23R") fg4 = 45.
if (foodcode = 5074 & foodgp ="23R") fg4 = 44.
if (foodcode = 5076 & foodgp ="23R") fg4 = 45.
if (foodcode = 5078 & foodgp ="23R") fg4 = 45.
if (foodcode = 5080 & foodgp ="23R") fg4 = 44.
if (foodcode = 5081 & foodgp ="23R") fg4 = 45.
if (foodcode = 5082 & foodgp ="23R") fg4 = 45.
if (foodcode = 5089 & foodgp ="23R") fg4 = 45.
if (foodcode = 5093 & foodgp ="23R") fg4 = 45.
if (foodcode = 5107 & foodgp ="23R") fg4 = 45.
if (foodcode = 5108 & foodgp ="23R") fg4 = 45.
if (foodcode = 5131 & foodgp ="23R") fg4 = 44.
if (foodcode = 5132 & foodgp ="23R") fg4 = 45.
if (foodcode = 5135 & foodgp ="23R") fg4 = 45.
if (foodcode = 5198 & foodgp ="23R") fg4 = 45.
if (foodcode = 5245 & foodgp ="23R") fg4 = 44.
if (foodcode = 5309 & foodgp ="23R") fg4 = 44.
if (foodcode = 5311 & foodgp ="23R") fg4 = 45.
if (foodcode = 5312 & foodgp ="23R") fg4 = 44.
if (foodcode = 5313 & foodgp ="23R") fg4 = 44.
if (foodcode = 5314 & foodgp ="23R") fg4 = 44.
if (foodcode = 5315 & foodgp ="23R") fg4 = 44.
if (foodcode = 5368 & foodgp ="23R") fg4 = 44.
if (foodcode = 5374 & foodgp ="23R") fg4 = 45.
if (foodcode = 5385 & foodgp ="23R") fg4 = 45.
if (foodcode = 5391 & foodgp ="23R") fg4 = 45.
if (foodcode = 5397 & foodgp ="23R") fg4 = 45.
if (foodcode = 5402 & foodgp ="23R") fg4 = 44.
if (foodcode = 5451 & foodgp ="23R") fg4 = 45.
if (foodcode = 5461 & foodgp ="23R") fg4 = 45.
if (foodcode = 5511 & foodgp ="23R") fg4 = 45.
if (foodcode = 5517 & foodgp ="23R") fg4 = 45.
if (foodcode = 5521 & foodgp ="23R") fg4 = 45.
if (foodcode = 5522 & foodgp ="23R") fg4 = 45.
if (foodcode = 5530 & foodgp ="23R") fg4 = 45.
if (foodcode = 5533 & foodgp ="23R") fg4 = 45.
if (foodcode = 5549 & foodgp ="23R") fg4 = 45.
if (foodcode = 5554 & foodgp ="23R") fg4 = 45.
if (foodcode = 5572 & foodgp ="23R") fg4 = 45.
if (foodcode = 5663 & foodgp ="23R") fg4 = 45.
if (foodcode = 5664 & foodgp ="23R") fg4 = 45.
if (foodcode = 5678 & foodgp ="23R") fg4 = 45.
if (foodcode = 5697 & foodgp ="23R") fg4 = 45.
if (foodcode = 5706 & foodgp ="23R") fg4 = 45.
```

```
if (foodcode = 5716 & foodgp ="23R") fg4 = 45.
if (foodcode = 5819 & foodgp ="23R") fg4 = 45.
if (foodcode = 5865 & foodgp ="23R") fg4 = 45.
if (foodcode = 5872 & foodgp ="23R") fg4 = 45.
if (foodcode = 5891 & foodgp ="23R") fg4 = 45.
if (foodcode = 5914 & foodgp ="23R") fg4 = 45.
if (foodcode = 5963 & foodgp ="23R") fg4 = 45.
if (foodcode = 6004 & foodgp ="23R") fg4 = 45.
if (foodcode = 6030 & foodgp ="23R") fg4 = 45.
if (foodcode = 6052 & foodgp ="23R") fg4 = 45.
if (foodcode = 6102 & foodgp ="23R") fg4 = 45.
if (foodcode = 6119 & foodgp ="23R") fg4 = 45.
if (foodcode = 6124 & foodgp ="23R") fg4 = 45.
if (foodcode = 6131 & foodgp ="23R") fg4 = 45.
if (foodcode = 6200 & foodgp ="23R") fg4 = 45.
if (foodcode = 6207 & foodgp ="23R") fg4 = 45.
if (foodcode = 6236 & foodgp ="23R") fg4 = 44.
if (foodcode = 6247 & foodgp ="23R") fg4 = 45.
if (foodcode = 6256 & foodgp ="23R") fg4 = 45.
if (foodcode = 6260 & foodgp ="23R") fg4 = 45.
if (foodcode = 6263 & foodgp ="23R") fg4 = 45.
if (foodcode = 6278 & foodgp ="23R") fg4 = 45.
if (foodcode = 6298 & foodgp ="23R") fg4 = 45.
if (foodcode = 6325 & foodgp ="23R") fg4 = 44.
if (foodcode = 6350 & foodgp ="23R") fg4 = 44.
if (foodcode = 6401 & foodgp ="23R") fg4 = 45.
if (foodcode = 6416 & foodgp ="23R") fg4 = 45.
if (foodcode = 6427 & foodgp ="23R") fg4 = 45.
if (foodcode = 6439 & foodgp ="23R") fg4 = 45.
if (foodcode = 6440 & foodgp ="23R") fg4 = 45.
if (foodcode = 6441 & foodgp ="23R") fg4 = 45.
if (foodcode = 6456 & foodgp ="23R") fg4 = 45.
if (foodcode = 6476 & foodgp ="23R") fg4 = 45.
if (foodcode = 6494 & foodgp ="23R") fg4 = 45.
if (foodcode = 6496 & foodgp ="23R") fg4 = 45.
if (foodcode = 6510 & foodgp ="23R") fg4 = 45.
if (foodcode = 6556 & foodgp ="23R") fg4 = 45.
if (foodcode = 6559 & foodgp ="23R") fg4 = 45.
if (foodcode = 6563 & foodgp ="23R") fg4 = 45.
if (foodcode = 6637 & foodgp ="23R") fg4 = 45.
if (foodcode = 6641 & foodgp ="23R") fg4 = 44.
if (foodcode = 6642 & foodgp ="23R") fg4 = 44.
if (foodcode = 6660 & foodgp ="23R") fg4 = 45.
if (foodcode = 6692 & foodgp ="23R") fg4 = 45.
if (foodcode = 6695 & foodgp ="23R") fg4 = 45.
if (foodcode = 6720 & foodgp ="23R") fg4 = 45.
if (foodcode = 6747 & foodgp ="23R") fg4 = 45.
if (foodcode = 6764 & foodgp ="23R") fg4 = 45.
if (foodcode = 6769 & foodgp ="23R") fg4 = 45.
if (foodcode = 7003 & foodgp ="23R") fg4 = 45.
if (foodcode = 7010 & foodgp ="23R") fg4 = 45.
if (foodcode = 7011 & foodgp ="23R") fg4 = 45.
if (foodcode = 7012 & foodgp ="23R") fg4 = 45.
if (foodcode = 7013 & foodgp ="23R") fg4 = 44.
if (foodcode = 7017 & foodgp ="23R") fg4 = 45.
if (foodcode = 7020 & foodgp ="23R") fg4 = 45.
if (foodcode = 7029 & foodgp ="23R") fg4 = 45.
if (foodcode = 7030 & foodgp ="23R") fg4 = 45.
if (foodcode = 7033 & foodgp ="23R") fg4 = 45.
if (foodcode = 7034 & foodgp ="23R") fg4 = 45.
if (foodcode = 7040 & foodgp ="23R") fg4 = 45.
if (foodcode = 7042 & foodgp ="23R") fg4 = 45.
if (foodcode = 7044 & foodgp ="23R") fg4 = 45.
if (foodcode = 7109 & foodgp ="23R") fg4 = 45.
if (foodcode = 7122 & foodgp ="23R") fg4 = 45.
if (foodcode = 7124 & foodgp ="23R") fg4 = 45.
if (foodcode = 7130 & foodgp ="23R") fg4 = 45.
if (foodcode = 7133 & foodgp ="23R") fg4 = 44.
if (foodcode = 7141 & foodgp ="23R") fg4 = 45.
if (foodcode = 7153 & foodgp ="23R") fg4 = 45.
if (foodcode = 7154 & foodgp ="23R") fg4 = 45.
if (foodcode = 7156 & foodgp ="23R") fg4 = 45.
if (foodcode = 7159 & foodgp ="23R") fg4 = 45.
if (foodcode = 7160 & foodgp ="23R") fg4 = 45.
if (foodcode = 7161 & foodgp ="23R") fg4 = 45.
if (foodcode = 7162 & foodgp ="23R") fg4 = 45.
```

```
if (foodcode = 7211 & foodgp ="23R") fg4 = 45.
if (foodcode = 7220 & foodgp ="23R") fg4 = 45.
if (foodcode = 7221 & foodgp ="23R") fg4 = 45.
if (foodcode = 7779 & foodgp ="23R") fg4 = 44.
if (foodcode = 7780 & foodgp ="23R") fg4 = 44.
if (foodcode = 7782 & foodgp ="23R") fg4 = 44.
if (foodcode = 8596 & foodgp ="23R") fg4 = 44.
if (foodcode = 8644 & foodgp ="23R") fg4 = 44.
if (foodcode = 8686 & foodgp ="23R") fg4 = 44.
if (foodcode = 8710 & foodgp ="23R") fg4 = 45.
if (foodcode = 8743 & foodgp ="23R") fg4 = 45.
if (foodcode = 8765 & foodgp ="23R") fg4 = 45.
if (foodcode = 8795 & foodgp ="23R") fg4 = 45.
if (foodcode = 8829 & foodgp ="23R") fg4 = 45.
if (foodcode = 8881 & foodgp ="23R") fg4 = 45.
if (foodcode = 8886 & foodgp ="23R") fg4 = 45.
if (foodcode = 8905 & foodgp ="23R") fg4 = 45.
if (foodcode = 8982 & foodgp ="23R") fg4 = 45.
if (foodcode = 9027 & foodgp ="23R") fg4 = 44.
if (foodcode = 9045 & foodgp ="23R") fg4 = 44.
if (foodcode = 9051 & foodgp ="23R") fg4 = 44.
if (foodcode = 9052 & foodgp ="23R") fg4 = 45.
if (foodcode = 9058 & foodgp ="23R") fg4 = 45.
if (foodcode = 9104 & foodgp ="23R") fg4 = 45.
if (foodcode = 9155 & foodgp ="23R") fg4 = 44.
if (foodcode = 9244 & foodgp ="23R") fg4 = 44.
if (foodcode = 9245 & foodgp ="23R") fg4 = 44.
if (foodcode = 9300 & foodgp ="23R") fg4 = 44.
if (foodcode = 9318 & foodgp ="23R") fg4 = 44.
if (foodcode = 9347 & foodgp ="23R") fg4 = 45.
if (foodcode = 9359 & foodgp ="23R") fg4 = 44.
if (foodcode = 9415 & foodgp ="23R") fg4 = 44.
if (foodcode = 9416 & foodgp ="23R") fg4 = 44.
if (foodcode = 9417 & foodgp ="23R") fg4 = 44.
if (foodcode = 9418 & foodgp ="23R") fg4 = 44.
if (foodcode = 9419 & foodgp ="23R") fg4 = 44.
if (foodcode = 9420 & foodgp ="23R") fg4 = 44.
if (foodcode = 9422 & foodgp ="23R") fg4 = 44.
if (foodcode = 9423 & foodgp ="23R") fg4 = 44.
if (foodcode = 9424 & foodgp ="23R") fg4 = 44.
if (foodcode = 9425 & foodgp ="23R") fg4 = 44.
if (foodcode = 9426 & foodgp ="23R") fg4 = 44.
if (foodcode = 9427 & foodgp ="23R") fg4 = 44.
if (foodcode = 9465 & foodgp ="23R") fg4 = 44.
if (foodcode = 9566 & foodgp ="23R") fg4 = 45.
if (foodcode = 9810 & foodgp ="23R") fg4 = 45.
if (foodcode = 1065 & foodgp ="27R") fg4 = 51.
if (foodcode = 1066 & foodgp ="27R") fg4 = 51.
if (foodcode = 1067 & foodgp ="27R") fg4 = 51.
if (foodcode = 1068 & foodgp ="27R") fg4 = 51.
if (foodcode = 1069 & foodgp ="27R") fg4 = 51.
if (foodcode = 1070 & foodgp ="27R") fg4 = 51.
if (foodcode = 1074 & foodgp ="27R") fg4 = 51.
if (foodcode = 1076 & foodgp ="27R") fg4 = 51.
if (foodcode = 1088 & foodgp ="27R") fg4 = 51.
if (foodcode = 1089 & foodgp ="27R") fg4 = 51.
if (foodcode = 1090 & foodgp ="27R") fg4 = 51.
if (foodcode = 1091 & foodgp ="27R") fg4 = 51.
if (foodcode = 1092 & foodgp ="27R") fg4 = 51.
if (foodcode = 1093 & foodgp ="27R") fg4 = 51.
if (foodcode = 1094 & foodgp ="27R") fg4 = 51.
if (foodcode = 1095 & foodgp ="27R") fg4 = 51.
if (foodcode = 1098 & foodgp ="27R") fg4 = 51.
if (foodcode = 1099 & foodgp ="27R") fg4 = 52.
if (foodcode = 1101 & foodgp ="27R") fg4 = 51.
if (foodcode = 1102 & foodgp ="27R") fg4 = 51.
if (foodcode = 1103 & foodgp ="27R") fg4 = 51.
if (foodcode = 1104 & foodgp ="27R") fg4 = 52.
if (foodcode = 1106 & foodgp ="27R") fg4 = 51.
if (foodcode = 1107 & foodgp ="27R") fg4 = 51.
if (foodcode = 1108 & foodgp ="27R") fg4 = 51.
if (foodcode = 1113 & foodgp ="27R") fg4 = 52.
if (foodcode = 1116 & foodgp ="27R") fg4 = 52.
if (foodcode = 1117 & foodgp ="27R") fg4 = 51.
if (foodcode = 1122 & foodgp ="27R") fg4 = 51.
if (foodcode = 1123 & foodgp ="27R") fg4 = 51.
```

```
if (foodcode = 1125 & foodgp ="27R") fg4 = 51.
if (foodcode = 1126 & foodgp ="27R") fg4 = 51.
if (foodcode = 1127 & foodgp ="27R") fg4 = 51.
if (foodcode = 1146 & foodgp ="27R") fg4 = 51.
if (foodcode = 1148 & foodgp ="27R") fg4 = 51.
if (foodcode = 1149 & foodgp ="27R") fg4 = 51.
if (foodcode = 1168 & foodgp ="27R") fg4 = 51.
if (foodcode = 1363 & foodgp ="27R") fg4 = 51.
if (foodcode = 1366 & foodgp ="27R") fg4 = 51.
if (foodcode = 1367 & foodgp ="27R") fg4 = 51.
if (foodcode = 1372 & foodgp ="27R") fg4 = 51.
if (foodcode = 1373 & foodgp ="27R") fg4 = 51.
if (foodcode = 1374 & foodgp ="27R") fg4 = 51.
if (foodcode = 1375 & foodgp ="27R") fg4 = 51.
if (foodcode = 2661 & foodgp ="27R") fg4 = 51.
if (foodcode = 2664 & foodgp ="27R") fg4 = 51.
if (foodcode = 2705 & foodgp ="27R") fg4 = 51.
if (foodcode = 2724 & foodgp ="27R") fg4 = 51.
if (foodcode = 2734 & foodgp ="27R") fg4 = 51.
if (foodcode = 2752 & foodgp ="27R") fg4 = 52.
if (foodcode = 2786 & foodgp ="27R") fg4 = 52.
if (foodcode = 2826 & foodgp ="27R") fg4 = 51.
if (foodcode = 2854 & foodgp ="27R") fg4 = 52.
if (foodcode = 2874 & foodgp ="27R") fg4 = 52.
if (foodcode = 2883 & foodgp ="27R") fg4 = 51.
if (foodcode = 2957 & foodgp ="27R") fg4 = 52.
if (foodcode = 3066 & foodgp ="27R") fg4 = 51.
if (foodcode = 3086 & foodgp ="27R") fg4 = 52.
if (foodcode = 3096 & foodgp ="27R") fg4 = 52.
if (foodcode = 3157 & foodgp ="27R") fg4 = 51.
if (foodcode = 3185 & foodgp ="27R") fg4 = 51.
if (foodcode = 3221 & foodgp ="27R") fg4 = 52.
if (foodcode = 3253 & foodgp ="27R") fg4 = 51.
if (foodcode = 3270 & foodgp ="27R") fg4 = 52.
if (foodcode = 3325 & foodgp ="27R") fg4 = 52.
if (foodcode = 3340 & foodgp ="27R") fg4 = 52.
if (foodcode = 3352 & foodgp ="27R") fg4 = 51.
if (foodcode = 3380 & foodgp ="27R") fg4 = 51.
if (foodcode = 3391 & foodgp ="27R") fg4 = 52.
if (foodcode = 3416 & foodgp ="27R") fg4 = 52.
if (foodcode = 3490 & foodgp ="27R") fg4 = 52.
if (foodcode = 3527 & foodgp ="27R") fg4 = 51.
if (foodcode = 3555 & foodgp ="27R") fg4 = 51.
if (foodcode = 3612 & foodgp ="27R") fg4 = 52.
if (foodcode = 3627 & foodgp ="27R") fg4 = 52.
if (foodcode = 3645 & foodgp ="27R") fg4 = 52.
if (foodcode = 3660 & foodgp ="27R") fg4 = 51.
if (foodcode = 3788 & foodgp ="27R") fg4 = 51.
if (foodcode = 3836 & foodgp ="27R") fg4 = 52.
if (foodcode = 3851 & foodgp ="27R") fg4 = 51.
if (foodcode = 3862 & foodgp ="27R") fg4 = 52.
if (foodcode = 3865 & foodgp ="27R") fg4 = 52.
if (foodcode = 3912 & foodgp ="27R") fg4 = 52.
if (foodcode = 3917 & foodgp ="27R") fg4 = 51.
if (foodcode = 3927 & foodgp ="27R") fg4 = 52.
if (foodcode = 4012 & foodgp ="27R") fg4 = 52.
if (foodcode = 4013 & foodgp ="27R") fg4 = 51.
if (foodcode = 4018 & foodgp ="27R") fg4 = 52.
if (foodcode = 4019 & foodgp ="27R") fg4 = 51.
if (foodcode = 4025 & foodgp ="27R") fg4 = 52.
if (foodcode = 4035 & foodgp ="27R") fg4 = 51.
if (foodcode = 4052 & foodgp ="27R") fg4 = 51.
if (foodcode = 4060 & foodgp ="27R") fg4 = 52.
if (foodcode = 4072 & foodgp ="27R") fg4 = 52.
if (foodcode = 4094 & foodgp ="27R") fg4 = 52.
if (foodcode = 4106 & foodgp ="27R") fg4 = 52.
if (foodcode = 4123 & foodgp ="27R") fg4 = 51.
if (foodcode = 4128 & foodgp ="27R") fg4 = 52.
if (foodcode = 4131 & foodgp ="27R") fg4 = 51.
if (foodcode = 4135 & foodgp ="27R") fg4 = 52.
if (foodcode = 4151 & foodgp ="27R") fg4 = 52.
if (foodcode = 4158 & foodgp ="27R") fg4 = 52.
if (foodcode = 4159 & foodgp ="27R") fg4 = 52.
if (foodcode = 4160 & foodgp ="27R") fg4 = 52.
if (foodcode = 4170 & foodgp ="27R") fg4 = 52.
if (foodcode = 4173 & foodgp ="27R") fg4 = 51.
```

```
if (foodcode = 4174 & foodgp ="27R") fg4 = 52.
if (foodcode = 4183 & foodgp ="27R") fg4 = 52.
if (foodcode = 4186 & foodgp ="27R") fg4 = 51.
if (foodcode = 4190 & foodgp ="27R") fg4 = 52.
if (foodcode = 4198 & foodgp ="27R") fg4 = 52.
if (foodcode = 4199 & foodgp ="27R") fg4 = 52.
if (foodcode = 4215 & foodgp ="27R") fg4 = 52.
if (foodcode = 4219 & foodgp ="27R") fg4 = 52.
if (foodcode = 4226 & foodgp ="27R") fg4 = 51.
if (foodcode = 4232 & foodgp ="27R") fg4 = 52.
if (foodcode = 4248 & foodgp ="27R") fg4 = 52.
if (foodcode = 4281 & foodgp ="27R") fg4 = 52.
if (foodcode = 4309 & foodgp ="27R") fg4 = 52.
if (foodcode = 4313 & foodgp ="27R") fg4 = 51.
if (foodcode = 4342 & foodgp ="27R") fg4 = 52.
if (foodcode = 4344 & foodgp ="27R") fg4 = 52.
if (foodcode = 4345 & foodgp ="27R") fg4 = 52.
if (foodcode = 4358 & foodgp ="27R") fg4 = 52.
if (foodcode = 4361 & foodgp ="27R") fg4 = 52.
if (foodcode = 4365 & foodgp ="27R") fg4 = 52.
if (foodcode = 4377 & foodgp ="27R") fg4 = 52.
if (foodcode = 4382 & foodgp ="27R") fg4 = 51.
if (foodcode = 4384 & foodgp ="27R") fg4 = 52.
if (foodcode = 4387 & foodgp ="27R") fg4 = 52.
if (foodcode = 4389 & foodgp ="27R") fg4 = 51.
if (foodcode = 4393 & foodgp ="27R") fg4 = 51.
if (foodcode = 4404 & foodgp ="27R") fg4 = 52.
if (foodcode = 4405 & foodgp ="27R") fg4 = 52.
if (foodcode = 4407 & foodgp ="27R") fg4 = 52.
if (foodcode = 4424 & foodgp ="27R") fg4 = 52.
if (foodcode = 4427 & foodgp ="27R") fg4 = 52.
if (foodcode = 5011 & foodgp ="27R") fg4 = 52.
if (foodcode = 5014 & foodgp ="27R") fg4 = 51.
if (foodcode = 5020 & foodgp ="27R") fg4 = 52.
if (foodcode = 5021 & foodgp ="27R") fg4 = 52.
if (foodcode = 5022 & foodgp ="27R") fg4 = 52.
if (foodcode = 5025 & foodgp ="27R") fg4 = 52.
if (foodcode = 5028 & foodgp ="27R") fg4 = 51.
if (foodcode = 5029 & foodgp ="27R") fg4 = 52.
if (foodcode = 5049 & foodgp ="27R") fg4 = 52.
if (foodcode = 5055 & foodgp ="27R") fg4 = 52.
if (foodcode = 5066 & foodgp ="27R") fg4 = 52.
if (foodcode = 5075 & foodgp ="27R") fg4 = 52.
if (foodcode = 5091 & foodgp ="27R") fg4 = 52.
if (foodcode = 5092 & foodgp ="27R") fg4 = 52.
if (foodcode = 5097 & foodgp ="27R") fg4 = 52.
if (foodcode = 5099 & foodgp ="27R") fg4 = 51.
if (foodcode = 5100 & foodgp ="27R") fg4 = 52.
if (foodcode = 5127 & foodgp ="27R") fg4 = 51.
if (foodcode = 5128 & foodgp ="27R") fg4 = 51.
if (foodcode = 5157 & foodgp ="27R") fg4 = 52.
if (foodcode = 5169 & foodgp ="27R") fg4 = 51.
if (foodcode = 5171 & foodgp ="27R") fg4 = 51.
if (foodcode = 5265 & foodgp ="27R") fg4 = 51.
if (foodcode = 5266 & foodgp ="27R") fg4 = 51.
if (foodcode = 5267 & foodgp ="27R") fg4 = 51.
if (foodcode = 5268 & foodgp ="27R") fg4 = 51.
if (foodcode = 5269 & foodgp ="27R") fg4 = 51.
if (foodcode = 5270 & foodgp ="27R") fg4 = 51.
if (foodcode = 5271 & foodgp ="27R") fg4 = 51.
if (foodcode = 5272 & foodgp ="27R") fg4 = 51.
if (foodcode = 5273 & foodgp ="27R") fg4 = 51.
if (foodcode = 5275 & foodgp ="27R") fg4 = 51.
if (foodcode = 5276 & foodgp ="27R") fg4 = 51.
if (foodcode = 5277 & foodgp ="27R") fg4 = 51.
if (foodcode = 5279 & foodgp ="27R") fg4 = 51.
if (foodcode = 5280 & foodgp ="27R") fg4 = 52.
if (foodcode = 5281 & foodgp ="27R") fg4 = 51.
if (foodcode = 5282 & foodgp ="27R") fg4 = 51.
if (foodcode = 5284 & foodgp ="27R") fg4 = 51.
if (foodcode = 5285 & foodgp ="27R") fg4 = 51.
if (foodcode = 5289 & foodgp ="27R") fg4 = 52.
if (foodcode = 5290 & foodgp ="27R") fg4 = 51.
if (foodcode = 5292 & foodgp ="27R") fg4 = 51.
if (foodcode = 5295 & foodgp ="27R") fg4 = 51.
if (foodcode = 5296 & foodgp ="27R") fg4 = 51.
```

```
if (foodcode = 5297 & foodgp ="27R") fg4 = 51.
if (foodcode = 5298 & foodgp ="27R") fg4 = 51.
if (foodcode = 5303 & foodgp ="27R") fg4 = 51.
if (foodcode = 5348 & foodgp ="27R") fg4 = 52.
if (foodcode = 5354 & foodgp ="27R") fg4 = 51.
if (foodcode = 5371 & foodgp ="27R") fg4 = 52.
if (foodcode = 5382 & foodgp ="27R") fg4 = 51.
if (foodcode = 5417 & foodgp ="27R") fg4 = 52.
if (foodcode = 5450 & foodgp ="27R") fg4 = 51.
if (foodcode = 5453 & foodgp ="27R") fg4 = 51.
if (foodcode = 5466 & foodgp ="27R") fg4 = 51.
if (foodcode = 5494 & foodgp ="27R") fg4 = 51.
if (foodcode = 5584 & foodgp ="27R") fg4 = 52.
if (foodcode = 5585 & foodgp ="27R") fg4 = 52.
if (foodcode = 5586 & foodgp ="27R") fg4 = 52.
if (foodcode = 5624 & foodgp ="27R") fg4 = 51.
if (foodcode = 5627 & foodgp ="27R") fg4 = 51.
if (foodcode = 5635 & foodgp ="27R") fg4 = 52.
if (foodcode = 5638 & foodgp ="27R") fg4 = 51.
if (foodcode = 5674 & foodgp ="27R") fg4 = 52.
if (foodcode = 5685 & foodgp ="27R") fg4 = 52.
if (foodcode = 5741 & foodgp ="27R") fg4 = 51.
if (foodcode = 5774 & foodgp ="27R") fg4 = 52.
if (foodcode = 5815 & foodgp ="27R") fg4 = 51.
if (foodcode = 5832 & foodgp ="27R") fg4 = 51.
if (foodcode = 5880 & foodgp ="27R") fg4 = 52.
if (foodcode = 5883 & foodgp ="27R") fg4 = 51.
if (foodcode = 5916 & foodgp ="27R") fg4 = 52.
if (foodcode = 5920 & foodgp ="27R") fg4 = 52.
if (foodcode = 5932 & foodgp ="27R") fg4 = 52.
if (foodcode = 5940 & foodgp ="27R") fg4 = 52.
if (foodcode = 5972 & foodgp ="27R") fg4 = 52.
if (foodcode = 5973 & foodgp ="27R") fg4 = 52.
if (foodcode = 6000 & foodgp ="27R") fg4 = 51.
if (foodcode = 6014 & foodgp ="27R") fg4 = 51.
if (foodcode = 6067 & foodgp ="27R") fg4 = 51.
if (foodcode = 6079 & foodgp ="27R") fg4 = 52.
if (foodcode = 6080 & foodgp ="27R") fg4 = 51.
if (foodcode = 6210 & foodgp ="27R") fg4 = 52.
if (foodcode = 6296 & foodgp ="27R") fg4 = 51.
if (foodcode = 6320 & foodgp ="27R") fg4 = 52.
if (foodcode = 6421 & foodgp ="27R") fg4 = 52.
if (foodcode = 6423 & foodgp ="27R") fg4 = 51.
if (foodcode = 6469 & foodgp ="27R") fg4 = 52.
if (foodcode = 6482 & foodgp ="27R") fg4 = 52.
if (foodcode = 6502 & foodgp ="27R") fg4 = 52.
if (foodcode = 6662 & foodgp ="27R") fg4 = 52.
if (foodcode = 6725 & foodgp ="27R") fg4 = 51.
if (foodcode = 6729 & foodgp ="27R") fg4 = 52.
if (foodcode = 6789 & foodgp ="27R") fg4 = 52.
if (foodcode = 6865 & foodgp ="27R") fg4 = 52.
if (foodcode = 6869 & foodgp ="27R") fg4 = 52.
if (foodcode = 6870 & foodgp ="27R") fg4 = 52.
if (foodcode = 6954 & foodgp ="27R") fg4 = 52.
if (foodcode = 6988 & foodgp ="27R") fg4 = 51.
if (foodcode = 6989 & foodgp ="27R") fg4 = 51.
if (foodcode = 6991 & foodgp ="27R") fg4 = 51.
if (foodcode = 6993 & foodgp ="27R") fg4 = 51.
if (foodcode = 7004 & foodgp ="27R") fg4 = 51.
if (foodcode = 7018 & foodgp ="27R") fg4 = 52.
if (foodcode = 7031 & foodgp ="27R") fg4 = 52.
if (foodcode = 7041 & foodgp ="27R") fg4 = 51.
if (foodcode = 7110 & foodgp ="27R") fg4 = 52.
if (foodcode = 7123 & foodgp ="27R") fg4 = 52.
if (foodcode = 7125 & foodgp ="27R") fg4 = 52.
if (foodcode = 7135 & foodgp ="27R") fg4 = 51.
if (foodcode = 7138 & foodgp ="27R") fg4 = 52.
if (foodcode = 7140 & foodgp ="27R") fg4 = 52.
if (foodcode = 7144 & foodgp ="27R") fg4 = 52.
if (foodcode = 7147 & foodgp ="27R") fg4 = 52.
if (foodcode = 7148 & foodgp ="27R") fg4 = 52.
if (foodcode = 7204 & foodgp ="27R") fg4 = 52.
if (foodcode = 7210 & foodgp ="27R") fg4 = 51.
if (foodcode = 7217 & foodgp ="27R") fg4 = 51.
if (foodcode = 7218 & foodgp ="27R") fg4 = 52.
if (foodcode = 7219 & foodgp ="27R") fg4 = 52.
```

```
if (foodcode = 7777 & foodgp ="27R") fg4 = 51.
if (foodcode = 8260 & foodgp ="27R") fg4 = 51.
if (foodcode = 8261 & foodgp ="27R") fg4 = 51.
if (foodcode = 8262 & foodgp ="27R") fg4 = 51.
if (foodcode = 8514 & foodgp ="27R") fg4 = 51.
if (foodcode = 8768 & foodgp ="27R") fg4 = 51.
if (foodcode = 8835 & foodgp ="27R") fg4 = 52.
if (foodcode = 8892 & foodgp ="27R") fg4 = 52.
if (foodcode = 8903 & foodgp ="27R") fg4 = 51.
if (foodcode = 8933 & foodgp ="27R") fg4 = 51.
if (foodcode = 9094 & foodgp ="27R") fg4 = 51.
if (foodcode = 9126 & foodgp ="27R") fg4 = 51.
if (foodcode = 9176 & foodgp ="27R") fg4 = 52.
if (foodcode = 9358 & foodgp ="27R") fg4 = 52.
if (foodcode = 9386 & foodgp ="27R") fg4 = 51.
if (foodcode = 9387 & foodgp ="27R") fg4 = 51.
if (foodcode = 9531 & foodgp ="27R") fg4 = 52.
if (foodcode = 9534 & foodgp ="27R") fg4 = 51.
if (foodcode = 9537 & foodgp ="27R") fg4 = 52.
if (foodcode = 9565 & foodgp ="27R") fg4 = 52.
if (foodcode = 9634 & foodgp ="27R") fg4 = 52.
if (foodcode = 9653 & foodgp ="27R") fg4 = 51.
if (foodcode = 9731 & foodgp ="27R") fg4 = 51.
if (foodcode = 9769 & foodgp ="27R") fg4 = 52.
if (foodcode = 9826 & foodgp ="27R") fg4 = 52.
if (foodcode = 977 & foodgp ="24R") fg4 = 46.
if (foodcode = 979 & foodgp ="24R") fg4 = 47.
if (foodcode = 980 & foodgp ="24R") fg4 = 46.
if (foodcode = 981 & foodgp ="24R") fg4 = 46.
if (foodcode = 982 & foodgp ="24R") fg4 = 46.
if (foodcode = 983 & foodgp ="24R") fg4 = 46.
if (foodcode = 984 & foodgp ="24R") fg4 = 46.
if (foodcode = 985 & foodgp ="24R") fg4 = 46.
if (foodcode = 986 & foodgp ="24R") fg4 = 46.
if (foodcode = 987 & foodgp ="24R") fg4 = 46.
if (foodcode = 988 & foodgp ="24R") fg4 = 46.
if (foodcode = 989 & foodgp ="24R") fg4 = 46.
if (foodcode = 990 & foodgp ="24R") fg4 = 46.
if (foodcode = 991 & foodgp ="24R") fg4 = 46.
if (foodcode = 994 & foodgp ="24R") fg4 = 46.
if (foodcode = 995 & foodgp ="24R") fg4 = 46.
if (foodcode = 996 & foodgp ="24R") fg4 = 46.
if (foodcode = 997 & foodgp ="24R") fg4 = 46.
if (foodcode = 999 & foodgp ="24R") fg4 = 46.
if (foodcode = 1000 & foodgp ="24R") fg4 = 46.
if (foodcode = 1002 & foodgp ="24R") fg4 = 46.
if (foodcode = 1003 & foodgp ="24R") fg4 = 46.
if (foodcode = 1005 & foodgp ="24R") fg4 = 47.
if (foodcode = 1006 & foodgp ="24R") fg4 = 47.
if (foodcode = 1008 & foodgp ="24R") fg4 = 46.
if (foodcode = 1009 & foodgp ="24R") fg4 = 46.
if (foodcode = 1055 & foodgp ="24R") fg4 = 46.
if (foodcode = 1238 & foodgp ="24R") fg4 = 46.
if (foodcode = 2763 & foodgp ="24R") fg4 = 47.
if (foodcode = 2774 & foodgp ="24R") fg4 = 47.
if (foodcode = 2806 & foodgp ="24R") fg4 = 47.
if (foodcode = 2926 & foodgp ="24R") fg4 = 47.
if (foodcode = 3248 & foodgp ="24R") fg4 = 47.
if (foodcode = 3646 & foodgp ="24R") fg4 = 47.
if (foodcode = 3813 & foodgp ="24R") fg4 = 47.
if (foodcode = 3825 & foodgp ="24R") fg4 = 47.
if (foodcode = 3868 & foodgp ="24R") fg4 = 47.
if (foodcode = 3871 & foodgp ="24R") fg4 = 47.
if (foodcode = 3878 & foodgp ="24R") fg4 = 47.
if (foodcode = 3919 & foodgp ="24R") fg4 = 47.
if (foodcode = 3946 & foodgp ="24R") fg4 = 47.
if (foodcode = 3966 & foodgp ="24R") fg4 = 47.
if (foodcode = 3967 & foodgp ="24R") fg4 = 46.
if (foodcode = 3970 & foodgp ="24R") fg4 = 47.
if (foodcode = 3972 & foodgp ="24R") fg4 = 47.
if (foodcode = 4058 & foodgp ="24R") fg4 = 47.
if (foodcode = 4133 & foodgp ="24R") fg4 = 47.
if (foodcode = 4138 & foodgp ="24R") fg4 = 47.
if (foodcode = 4145 & foodgp ="24R") fg4 = 47.
if (foodcode = 4155 & foodgp ="24R") fg4 = 47.
if (foodcode = 4163 & foodgp ="24R") fg4 = 47.
```

```
if (foodcode = 4177 & foodgp ="24R") fg4 = 47.
if (foodcode = 4192 & foodgp ="24R") fg4 = 47.
if (foodcode = 4194 & foodgp ="24R") fg4 = 47.
if (foodcode = 4222 & foodgp ="24R") fg4 = 47.
if (foodcode = 4233 & foodgp ="24R") fg4 = 47.
if (foodcode = 4234 & foodgp ="24R") fg4 = 47.
if (foodcode = 4254 & foodgp ="24R") fg4 = 47.
if (foodcode = 4259 & foodgp ="24R") fg4 = 47.
if (foodcode = 4260 & foodgp ="24R") fg4 = 47.
if (foodcode = 4276 & foodgp ="24R") fg4 = 47.
if (foodcode = 4278 & foodgp ="24R") fg4 = 47.
if (foodcode = 4292 & foodgp ="24R") fg4 = 47.
if (foodcode = 4308 & foodgp ="24R") fg4 = 47.
if (foodcode = 4315 & foodgp ="24R") fg4 = 47.
if (foodcode = 4316 & foodgp ="24R") fg4 = 47.
if (foodcode = 4322 & foodgp ="24R") fg4 = 47.
if (foodcode = 4341 & foodgp ="24R") fg4 = 47.
if (foodcode = 4362 & foodgp ="24R") fg4 = 47.
if (foodcode = 4363 & foodgp ="24R") fg4 = 47.
if (foodcode = 4420 & foodgp ="24R") fg4 = 46.
if (foodcode = 4421 & foodgp ="24R") fg4 = 47.
if (foodcode = 4423 & foodgp ="24R") fg4 = 47.
if (foodcode = 4425 & foodgp ="24R") fg4 = 47.
if (foodcode = 5044 & foodgp ="24R") fg4 = 46.
if (foodcode = 5068 & foodgp ="24R") fg4 = 46.
if (foodcode = 5088 & foodgp ="24R") fg4 = 47.
if (foodcode = 5090 & foodgp ="24R") fg4 = 47.
if (foodcode = 5217 & foodgp ="24R") fg4 = 46.
if (foodcode = 5316 & foodgp ="24R") fg4 = 46.
if (foodcode = 5317 & foodgp ="24R") fg4 = 47.
if (foodcode = 5318 & foodgp ="24R") fg4 = 47.
if (foodcode = 5319 & foodgp ="24R") fg4 = 46.
if (foodcode = 5321 & foodgp ="24R") fg4 = 46.
if (foodcode = 5322 & foodgp ="24R") fg4 = 46.
if (foodcode = 5443 & foodgp ="24R") fg4 = 47.
if (foodcode = 5454 & foodgp ="24R") fg4 = 47.
if (foodcode = 5524 & foodgp ="24R") fg4 = 47.
if (foodcode = 5540 & foodgp ="24R") fg4 = 47.
if (foodcode = 5592 & foodgp ="24R") fg4 = 47.
if (foodcode = 5748 & foodgp ="24R") fg4 = 47.
if (foodcode = 5756 & foodgp ="24R") fg4 = 47.
if (foodcode = 5762 & foodgp ="24R") fg4 = 47.
if (foodcode = 5800 & foodgp ="24R") fg4 = 47.
if (foodcode = 5844 & foodgp ="24R") fg4 = 47.
if (foodcode = 5917 & foodgp ="24R") fg4 = 47.
if (foodcode = 5930 & foodgp ="24R") fg4 = 46.
if (foodcode = 5937 & foodgp ="24R") fg4 = 47.
if (foodcode = 5939 & foodgp ="24R") fg4 = 47.
if (foodcode = 5942 & foodgp ="24R") fg4 = 47.
if (foodcode = 6025 & foodgp ="24R") fg4 = 47.
if (foodcode = 6088 & foodgp ="24R") fg4 = 46.
if (foodcode = 6425 & foodgp ="24R") fg4 = 47.
if (foodcode = 6667 & foodgp ="24R") fg4 = 47.
if (foodcode = 6685 & foodgp ="24R") fg4 = 47.
if (foodcode = 6726 & foodgp ="24R") fg4 = 47.
if (foodcode = 6990 & foodgp ="24R") fg4 = 46.
if (foodcode = 7001 & foodgp ="24R") fg4 = 47.
if (foodcode = 7005 & foodgp ="24R") fg4 = 47.
if (foodcode = 7007 & foodgp ="24R") fg4 = 47.
if (foodcode = 7015 & foodgp ="24R") fg4 = 47.
if (foodcode = 7120 & foodgp ="24R") fg4 = 47.
if (foodcode = 7126 & foodgp ="24R") fg4 = 47.
if (foodcode = 7146 & foodgp ="24R") fg4 = 47.
if (foodcode = 7157 & foodgp ="24R") fg4 = 47.
if (foodcode = 7301 & foodgp ="24R") fg4 = 47.
if (foodcode = 8248 & foodgp ="24R") fg4 = 46.
if (foodcode = 8642 & foodgp ="24R") fg4 = 46.
if (foodcode = 8674 & foodgp ="24R") fg4 = 47.
if (foodcode = 8861 & foodgp ="24R") fg4 = 46.
if (foodcode = 8999 & foodgp ="24R") fg4 = 47.
if (foodcode = 9429 & foodgp ="24R") fg4 = 46.
if (foodcode = 9430 & foodgp ="24R") fg4 = 46.
if (foodcode = 9431 & foodgp ="24R") fg4 = 46.
if (foodcode = 9432 & foodgp ="24R") fg4 = 46.
if (foodcode = 9433 & foodgp ="24R") fg4 = 46.
if (foodcode = 9434 & foodgp ="24R") fg4 = 46.
```

```
if (foodcode = 9435 & foodgp ="24R") fg4 = 46.
if (foodcode = 9437 & foodgp ="24R") fg4 = 46.
if (foodcode = 9439 & foodgp ="24R") fg4 = 46.
if (foodcode = 9440 & foodgp ="24R") fg4 = 46.
if (foodcode = 9441 & foodgp ="24R") fg4 = 46.
if (foodcode = 9547 & foodgp ="24R") fg4 = 46.
if (foodcode = 9630 & foodgp ="24R") fg4 = 47.
if (foodcode = 1020 & foodgp ="25R") fg4 = 48.
if (foodcode = 1022 & foodgp ="25R") fg4 = 48.
if (foodcode = 1023 & foodgp ="25R") fg4 = 48.
if (foodcode = 1024 & foodgp ="25R") fg4 = 48.
if (foodcode = 1025 & foodgp ="25R") fg4 = 48.
if (foodcode = 1026 & foodgp ="25R") fg4 = 48.
if (foodcode = 1027 & foodgp ="25R") fg4 = 48.
if (foodcode = 1028 & foodgp ="25R") fg4 = 48.
if (foodcode = 1030 & foodgp ="25R") fg4 = 48.
if (foodcode = 1032 & foodgp ="25R") fg4 = 48.
if (foodcode = 1033 & foodgp ="25R") fg4 = 48.
if (foodcode = 1041 & foodgp ="25R") fg4 = 49.
if (foodcode = 1331 & foodgp ="25R") fg4 = 48.
if (foodcode = 1352 & foodgp ="25R") fg4 = 48.
if (foodcode = 1353 & foodgp ="25R") fg4 = 48.
if (foodcode = 1355 & foodgp ="25R") fg4 = 48.
if (foodcode = 3447 & foodgp ="25R") fg4 = 48.
if (foodcode = 3808 & foodgp ="25R") fg4 = 48.
if (foodcode = 3885 & foodgp ="25R") fg4 = 49.
if (foodcode = 3920 & foodgp ="25R") fg4 = 49.
if (foodcode = 4069 & foodgp ="25R") fg4 = 49.
if (foodcode = 4071 & foodgp ="25R") fg4 = 48.
if (foodcode = 4097 & foodgp ="25R") fg4 = 48.
if (foodcode = 4205 & foodgp ="25R") fg4 = 49.
if (foodcode = 4214 & foodgp ="25R") fg4 = 49.
if (foodcode = 4266 & foodgp ="25R") fg4 = 48.
if (foodcode = 4353 & foodgp ="25R") fg4 = 48.
if (foodcode = 4357 & foodgp ="25R") fg4 = 49.
if (foodcode = 4391 & foodgp ="25R") fg4 = 49.
if (foodcode = 5023 & foodgp ="25R") fg4 = 49.
if (foodcode = 5060 & foodgp ="25R") fg4 = 48.
if (foodcode = 5323 & foodgp ="25R") fg4 = 48.
if (foodcode = 5422 & foodgp ="25R") fg4 = 48.
if (foodcode = 5578 & foodgp ="25R") fg4 = 49.
if (foodcode = 5589 & foodgp ="25R") fg4 = 49.
if (foodcode = 5601 & foodgp ="25R") fg4 = 49.
if (foodcode = 5616 & foodgp ="25R") fg4 = 49.
if (foodcode = 5695 & foodgp ="25R") fg4 = 49.
if (foodcode = 5735 & foodgp ="25R") fg4 = 48.
if (foodcode = 6009 & foodgp ="25R") fg4 = 48.
if (foodcode = 6168 & foodgp ="25R") fg4 = 48.
if (foodcode = 6235 & foodgp ="25R") fg4 = 48.
if (foodcode = 6300 & foodgp ="25R") fg4 = 49.
if (foodcode = 6519 & foodgp ="25R") fg4 = 49.
if (foodcode = 6707 & foodgp ="25R") fg4 = 48.
if (foodcode = 6908 & foodgp ="25R") fg4 = 48.
if (foodcode = 8249 & foodgp ="25R") fg4 = 48.
if (foodcode = 9402 & foodgp ="25R") fg4 = 48.
if (foodcode = 9443 & foodgp ="25R") fg4 = 48.
if (foodcode = 9444 & foodgp ="25R") fg4 = 48.
if (foodcode = 9448 & foodgp ="25R") fg4 = 48.
if (foodcode = 9449 & foodgp ="25R") fg4 = 48.
if (foodcode = 9450 & foodgp ="25R") fg4 = 48.
if (foodcode = 9451 & foodgp ="25R") fg4 = 48.
if (foodcode = 9452 & foodgp ="25R") fg4 = 48.
if (foodcode = 9454 & foodgp ="25R") fg4 = 48.
if (foodcode = 9456 & foodgp ="25R") fg4 = 48.
if (foodcode = 9458 & foodgp ="25R") fg4 = 48.
if (foodcode = 9462 & foodgp ="25R") fg4 = 48.
if (foodcode = 9463 & foodgp ="25R") fg4 = 48.
if (foodcode = 9476 & foodgp ="25R") fg4 = 48.
if (foodcode = 9763 & foodgp ="25R") fg4 = 48.
execute.

sort cases by shserial(a) respno(a).

save
outfile ='C:\food level fg4.sav'
 /COMPRESSED.
```

```
Get file ='C:\food level fg4.sav'.



AGGREGATE
  /OUTFILE='C:\day-food level fg4.sav'
  /BREAK=shserial  respno dayno fg4
  /wghttot4 'weight total by fg4' = sum(wtfood1).



GET FILE='C:\day-food level fg4.sav'.

IF(fg4=1) wg4fg1 = wghttot4.
IF(fg4=2) wg4fg2 = wghttot4.
IF(fg4=3) wg4fg3 = wghttot4.
IF(fg4=4) wg4fg4 = wghttot4.
IF(fg4=5) wg4fg5 = wghttot4.
IF(fg4=6) wg4fg6 = wghttot4.
IF(fg4=7) wg4fg7 = wghttot4.
IF(fg4=8) wg4fg8 = wghttot4.
IF(fg4=9) wg4fg9 = wghttot4.
IF(fg4=10) wg4fg10 = wghttot4.
IF(fg4=11) wg4fg11 = wghttot4.
IF(fg4=12) wg4fg12 = wghttot4.
IF(fg4=13) wg4fg13 = wghttot4.
IF(fg4=14) wg4fg14 = wghttot4.
IF(fg4=15) wg4fg15 = wghttot4.
IF(fg4=16) wg4fg16 = wghttot4.
IF(fg4=17) wg4fg17 = wghttot4.
IF(fg4=18) wg4fg18 = wghttot4.
IF(fg4=19) wg4fg19 = wghttot4.
IF(fg4=20) wg4fg20 = wghttot4.
IF(fg4=21) wg4fg21 = wghttot4.
IF(fg4=22) wg4fg22 = wghttot4.
IF(fg4=23) wg4fg23 = wghttot4.
IF(fg4=24) wg4fg24 = wghttot4.
IF(fg4=25) wg4fg25 = wghttot4.
IF(fg4=26) wg4fg26 = wghttot4.
IF(fg4=27) wg4fg27 = wghttot4.
IF(fg4=28) wg4fg28 = wghttot4.
IF(fg4=29) wg4fg29 = wghttot4.
IF(fg4=30) wg4fg30 = wghttot4.
IF(fg4=31) wg4fg31 = wghttot4.
IF(fg4=32) wg4fg32 = wghttot4.
IF(fg4=33) wg4fg33 = wghttot4.
IF(fg4=34) wg4fg34 = wghttot4.
IF(fg4=35) wg4fg35 = wghttot4.
IF(fg4=36) wg4fg36 = wghttot4.
IF(fg4=37) wg4fg37 = wghttot4.
IF(fg4=38) wg4fg38 = wghttot4.
IF(fg4=39) wg4fg39 = wghttot4.
IF(fg4=40) wg4fg40 = wghttot4.
IF(fg4=41) wg4fg41 = wghttot4.
IF(fg4=42) wg4fg42 = wghttot4.
IF(fg4=43) wg4fg43 = wghttot4.
IF(fg4=44) wg4fg44 = wghttot4.
IF(fg4=45) wg4fg45 = wghttot4.
IF(fg4=46) wg4fg46 = wghttot4.
IF(fg4=47) wg4fg47 = wghttot4.
IF(fg4=48) wg4fg48 = wghttot4.
IF(fg4=49) wg4fg49 = wghttot4.
IF(fg4=50) wg4fg50 = wghttot4.
IF(fg4=51) wg4fg51 = wghttot4.
IF(fg4=52) wg4fg52 = wghttot4.
IF(fg4=53) wg4fg53 = wghttot4.
IF(fg4=54) wg4fg54 = wghttot4.
IF(fg4=55) wg4fg55 = wghttot4.
IF(fg4=56) wg4fg56 = wghttot4.
IF(fg4=57) wg4fg57 = wghttot4.
IF(fg4=58) wg4fg58 = wghttot4.
IF(fg4=59) wg4fg59 = wghttot4.
IF(fg4=60) wg4fg60 = wghttot4.
IF(fg4=61) wg4fg61 = wghttot4.
IF(fg4=62) wg4fg62 = wghttot4.
```

```
IF(fg4=63) wg4fg63 = wghttot4.
IF(fg4=64) wg4fg64 = wghttot4.
IF(fg4=65) wg4fg65 = wghttot4.
IF(fg4=66) wg4fg66 = wghttot4.
IF(fg4=67) wg4fg67 = wghttot4.
IF(fg4=68) wg4fg68 = wghttot4.
IF(fg4=69) wg4fg69 = wghttot4.
IF(fg4=70) wg4fg70 = wghttot4.
IF(fg4=71) wg4fg71 = wghttot4.
IF(fg4=72) wg4fg72 = wghttot4.
IF(fg4=73) wg4fg73 = wghttot4.
IF(fg4=74) wg4fg74 = wghttot4.
IF(fg4=75) wg4fg75 = wghttot4.
IF(fg4=76) wg4fg76 = wghttot4.
IF(fg4=77) wg4fg77 = wghttot4.
IF(fg4=78) wg4fg78 = wghttot4.
IF(fg4=79) wg4fg79 = wghttot4.
IF(fg4=80) wg4fg80 = wghttot4.
IF(fg4=81) wg4fg81 = wghttot4.
IF(fg4=82) wg4fg82 = wghttot4.
IF(fg4=83) wg4fg83 = wghttot4.
IF(fg4=84) wg4fg84 = wghttot4.
IF(fg4=85) wg4fg85 = wghttot4.
IF(fg4=86) wg4fg86 = wghttot4.
IF(fg4=87) wg4fg87 = wghttot4.
IF(fg4=88) wg4fg88 = wghttot4.
IF(fg4=89) wg4fg89 = wghttot4.
IF(fg4=90) wg4fg90 = wghttot4.
IF(fg4=91) wg4fg91 = wghttot4.
IF(fg4=92) wg4fg92 = wghttot4.
IF(fg4=93) wg4fg93 = wghttot4.
IF(fg4=94) wg4fg94 = wghttot4.
IF(fg4=95) wg4fg95 = wghttot4.
IF(fg4=96) wg4fg96 = wghttot4.
IF(fg4=97) wg4fg97 = wghttot4.
IF(fg4=98) wg4fg98 = wghttot4.
IF(fg4=99) wg4fg99 = wghttot4.
IF(fg4=100) wg4fg100 = wghttot4.
IF(fg4=101) wg4fg101 = wghttot4.
IF(fg4=102) wg4fg102 = wghttot4.
IF(fg4=103) wg4fg103 = wghttot4.
IF(fg4=104) wg4fg104 = wghttot4.
IF(fg4=105) wg4fg105 = wghttot4.
IF(fg4=106) wg4fg106 = wghttot4.
IF(fg4=107) wg4fg107 = wghttot4.
IF(fg4=108) wg4fg108 = wghttot4.
IF(fg4=109) wg4fg109 = wghttot4.
IF(fg4=110) wg4fg110 = wghttot4.
IF(fg4=111) wg4fg111 = wghttot4.
IF(fg4=112) wg4fg112 = wghttot4.
IF(fg4=113) wg4fg113 = wghttot4.
IF(fg4=114) wg4fg114 = wghttot4.
IF(fg4=115) wg4fg115 = wghttot4.
IF(fg4=116) wg4fg116 = wghttot4.
IF(fg4=117) wg4fg117 = wghttot4.
IF(fg4=118) wg4fg118 = wghttot4.
IF(fg4=119) wg4fg119 = wghttot4.
IF(fg4=120) wg4fg120 = wghttot4.
IF(fg4=121) wg4fg121 = wghttot4.
IF(fg4=122) wg4fg122 = wghttot4.
IF(fg4=123) wg4fg123 = wghttot4.
IF(fg4=124) wg4fg124 = wghttot4.
EXECUTE .


IF(fg4~=1) wg4fg1=0.
IF(fg4~=2) wg4fg2=0.
IF(fg4~=3) wg4fg3=0.
IF(fg4~=4) wg4fg4=0.
IF(fg4~=5) wg4fg5=0.
IF(fg4~=6) wg4fg6=0.
IF(fg4~=7) wg4fg7=0.
IF(fg4~=8) wg4fg8=0.
IF(fg4~=9) wg4fg9=0.
IF(fg4~=10) wg4fg10=0.
IF(fg4~=11) wg4fg11=0.
```

```
IF(fg4~=12) wg4fg12=0.
IF(fg4~=13) wg4fg13=0.
IF(fg4~=14) wg4fg14=0.
IF(fg4~=15) wg4fg15=0.
IF(fg4~=16) wg4fg16=0.
IF(fg4~=17) wg4fg17=0.
IF(fg4~=18) wg4fg18=0.
IF(fg4~=19) wg4fg19=0.
IF(fg4~=20) wg4fg20=0.
IF(fg4~=21) wg4fg21=0.
IF(fg4~=22) wg4fg22=0.
IF(fg4~=23) wg4fg23=0.
IF(fg4~=24) wg4fg24=0.
IF(fg4~=25) wg4fg25=0.
IF(fg4~=26) wg4fg26=0.
IF(fg4~=27) wg4fg27=0.
IF(fg4~=28) wg4fg28=0.
IF(fg4~=29) wg4fg29=0.
IF(fg4~=30) wg4fg30=0.
IF(fg4~=31) wg4fg31=0.
IF(fg4~=32) wg4fg32=0.
IF(fg4~=33) wg4fg33=0.
IF(fg4~=34) wg4fg34=0.
IF(fg4~=35) wg4fg35=0.
IF(fg4~=36) wg4fg36=0.
IF(fg4~=37) wg4fg37=0.
IF(fg4~=38) wg4fg38=0.
IF(fg4~=39) wg4fg39=0.
IF(fg4~=40) wg4fg40=0.
IF(fg4~=41) wg4fg41=0.
IF(fg4~=42) wg4fg42=0.
IF(fg4~=43) wg4fg43=0.
IF(fg4~=44) wg4fg44=0.
IF(fg4~=45) wg4fg45=0.
IF(fg4~=46) wg4fg46=0.
IF(fg4~=47) wg4fg47=0.
IF(fg4~=48) wg4fg48=0.
IF(fg4~=49) wg4fg49=0.
IF(fg4~=50) wg4fg50=0.
IF(fg4~=51) wg4fg51=0.
IF(fg4~=52) wg4fg52=0.
IF(fg4~=53) wg4fg53=0.
IF(fg4~=54) wg4fg54=0.
IF(fg4~=55) wg4fg55=0.
IF(fg4~=56) wg4fg56=0.
IF(fg4~=57) wg4fg57=0.
IF(fg4~=58) wg4fg58=0.
IF(fg4~=59) wg4fg59=0.
IF(fg4~=60) wg4fg60=0.
IF(fg4~=61) wg4fg61=0.
IF(fg4~=62) wg4fg62=0.
IF(fg4~=63) wg4fg63=0.
IF(fg4~=64) wg4fg64=0.
IF(fg4~=65) wg4fg65=0.
IF(fg4~=66) wg4fg66=0.
IF(fg4~=67) wg4fg67=0.
IF(fg4~=68) wg4fg68=0.
IF(fg4~=69) wg4fg69=0.
IF(fg4~=70) wg4fg70=0.
IF(fg4~=71) wg4fg71=0.
IF(fg4~=72) wg4fg72=0.
IF(fg4~=73) wg4fg73=0.
IF(fg4~=74) wg4fg74=0.
IF(fg4~=75) wg4fg75=0.
IF(fg4~=76) wg4fg76=0.
IF(fg4~=77) wg4fg77=0.
IF(fg4~=78) wg4fg78=0.
IF(fg4~=79) wg4fg79=0.
IF(fg4~=80) wg4fg80=0.
IF(fg4~=81) wg4fg81=0.
IF(fg4~=82) wg4fg82=0.
IF(fg4~=83) wg4fg83=0.
IF(fg4~=84) wg4fg84=0.
IF(fg4~=85) wg4fg85=0.
IF(fg4~=86) wg4fg86=0.
IF(fg4~=87) wg4fg87=0.
```

```
IF(fg4~=88) wg4fg88=0.
IF(fg4~=89) wg4fg89=0.
IF(fg4~=90) wg4fg90=0.
IF(fg4~=91) wg4fg91=0.
IF(fg4~=92) wg4fg92=0.
IF(fg4~=93) wg4fg93=0.
IF(fg4~=94) wg4fg94=0.
IF(fg4~=95) wg4fg95=0.
IF(fg4~=96) wg4fg96=0.
IF(fg4~=97) wg4fg97=0.
IF(fg4~=98) wg4fg98=0.
IF(fg4~=99) wg4fg99=0.
IF(fg4~=100) wg4fg100=0.
IF(fg4~=101) wg4fg101=0.
IF(fg4~=102) wg4fg102=0.
IF(fg4~=103) wg4fg103=0.
IF(fg4~=104) wg4fg104=0.
IF(fg4~=105) wg4fg105=0.
IF(fg4~=106) wg4fg106=0.
IF(fg4~=107) wg4fg107=0.
IF(fg4~=108) wg4fg108=0.
IF(fg4~=109) wg4fg109=0.
IF(fg4~=110) wg4fg110=0.
IF(fg4~=111) wg4fg111=0.
IF(fg4~=112) wg4fg112=0.
IF(fg4~=113) wg4fg113=0.
IF(fg4~=114) wg4fg114=0.
IF(fg4~=115) wg4fg115=0.
IF(fg4~=116) wg4fg116=0.
IF(fg4~=117) wg4fg117=0.
IF(fg4~=118) wg4fg118=0.
IF(fg4~=119) wg4fg119=0.
IF(fg4~=120) wg4fg120=0.
IF(fg4~=121) wg4fg121=0.
IF(fg4~=122) wg4fg122=0.
IF(fg4~=123) wg4fg123=0.
IF(fg4~=124) wg4fg124=0.
execute.

sort cases by shserial(a) respno(a).

save
outfile ='C:\day-food level fg4.sav'.

AGGREGATE
  /OUTFILE='C:\day-food level fg4_1.sav'
  /BREAK=shserial  respno dayno
/wd4fg1= sum(wg4fg1)
/wd4fg2= sum(wg4fg2)
/wd4fg3= sum(wg4fg3)
/wd4fg4= sum(wg4fg4)
/wd4fg5= sum(wg4fg5)
/wd4fg6= sum(wg4fg6)
/wd4fg7= sum(wg4fg7)
/wd4fg8= sum(wg4fg8)
/wd4fg9= sum(wg4fg9)
/wd4fg10= sum(wg4fg10)
/wd4fg11= sum(wg4fg11)
/wd4fg12= sum(wg4fg12)
/wd4fg13= sum(wg4fg13)
/wd4fg14= sum(wg4fg14)
/wd4fg15= sum(wg4fg15)
/wd4fg16= sum(wg4fg16)
/wd4fg17= sum(wg4fg17)
/wd4fg18= sum(wg4fg18)
/wd4fg19= sum(wg4fg19)
/wd4fg20= sum(wg4fg20)
/wd4fg21= sum(wg4fg21)
/wd4fg22= sum(wg4fg22)
/wd4fg23= sum(wg4fg23)
/wd4fg24= sum(wg4fg24)
/wd4fg25= sum(wg4fg25)
/wd4fg26= sum(wg4fg26)
/wd4fg27= sum(wg4fg27)
/wd4fg28= sum(wg4fg28)
/wd4fg29= sum(wg4fg29)
```

```
/wd4fg30= sum(wg4fg30)
/wd4fg31= sum(wg4fg31)
/wd4fg32= sum(wg4fg32)
/wd4fg33= sum(wg4fg33)
/wd4fg34= sum(wg4fg34)
/wd4fg35= sum(wg4fg35)
/wd4fg36= sum(wg4fg36)
/wd4fg37= sum(wg4fg37)
/wd4fg38= sum(wg4fg38)
/wd4fg39= sum(wg4fg39)
/wd4fg40= sum(wg4fg40)
/wd4fg41= sum(wg4fg41)
/wd4fg42= sum(wg4fg42)
/wd4fg43= sum(wg4fg43)
/wd4fg44= sum(wg4fg44)
/wd4fg45= sum(wg4fg45)
/wd4fg46= sum(wg4fg46)
/wd4fg47= sum(wg4fg47)
/wd4fg48= sum(wg4fg48)
/wd4fg49= sum(wg4fg49)
/wd4fg50= sum(wg4fg50)
/wd4fg51= sum(wg4fg51)
/wd4fg52= sum(wg4fg52)
/wd4fg53= sum(wg4fg53)
/wd4fg54= sum(wg4fg54)
/wd4fg55= sum(wg4fg55)
/wd4fg56= sum(wg4fg56)
/wd4fg57= sum(wg4fg57)
/wd4fg58= sum(wg4fg58)
/wd4fg59= sum(wg4fg59)
/wd4fg60= sum(wg4fg60)
/wd4fg61= sum(wg4fg61)
/wd4fg62= sum(wg4fg62)
/wd4fg63= sum(wg4fg63)
/wd4fg64= sum(wg4fg64)
/wd4fg65= sum(wg4fg65)
/wd4fg66= sum(wg4fg66)
/wd4fg67= sum(wg4fg67)
/wd4fg68= sum(wg4fg68)
/wd4fg69= sum(wg4fg69)
/wd4fg70= sum(wg4fg70)
/wd4fg71= sum(wg4fg71)
/wd4fg72= sum(wg4fg72)
/wd4fg73= sum(wg4fg73)
/wd4fg74= sum(wg4fg74)
/wd4fg75= sum(wg4fg75)
/wd4fg76= sum(wg4fg76)
/wd4fg77= sum(wg4fg77)
/wd4fg78= sum(wg4fg78)
/wd4fg79= sum(wg4fg79)
/wd4fg80= sum(wg4fg80)
/wd4fg81= sum(wg4fg81)
/wd4fg82= sum(wg4fg82)
/wd4fg83= sum(wg4fg83)
/wd4fg84= sum(wg4fg84)
/wd4fg85= sum(wg4fg85)
/wd4fg86= sum(wg4fg86)
/wd4fg87= sum(wg4fg87)
/wd4fg88= sum(wg4fg88)
/wd4fg89= sum(wg4fg89)
/wd4fg90= sum(wg4fg90)
/wd4fg91= sum(wg4fg91)
/wd4fg92= sum(wg4fg92)
/wd4fg93= sum(wg4fg93)
/wd4fg94= sum(wg4fg94)
/wd4fg95= sum(wg4fg95)
/wd4fg96= sum(wg4fg96)
/wd4fg97= sum(wg4fg97)
/wd4fg98= sum(wg4fg98)
/wd4fg99= sum(wg4fg99)
/wd4fg100= sum(wg4fg100)
/wd4fg101= sum(wg4fg101)
/wd4fg102= sum(wg4fg102)
/wd4fg103= sum(wg4fg103)
/wd4fg104= sum(wg4fg104)
/wd4fg105= sum(wg4fg105)
```

```
/wd4fg106= sum(wg4fg106)
/wd4fg107= sum(wg4fg107)
/wd4fg108= sum(wg4fg108)
/wd4fg109= sum(wg4fg109)
/wd4fg110= sum(wg4fg110)
/wd4fg111= sum(wg4fg111)
/wd4fg112= sum(wg4fg112)
/wd4fg113= sum(wg4fg113)
/wd4fg114= sum(wg4fg114)
/wd4fg115= sum(wg4fg115)
/wd4fg116= sum(wg4fg116)
/wd4fg117= sum(wg4fg117)
/wd4fg118= sum(wg4fg118)
/wd4fg119= sum(wg4fg119)
/wd4fg120= sum(wg4fg120)
/wd4fg121= sum(wg4fg121)
/wd4fg122= sum(wg4fg122)
/wd4fg123= sum(wg4fg123)
/wd4fg124= sum(wg4fg124).

get
file ='C:\day-food level fg4_1.sav'.

AGGREGATE
  /OUTFILE='C:\day-food level fg4_2.sav'
  /BREAK=shserial  respno
/w4fg1= sum(wd4fg1)
/w4fg2= sum(wd4fg2)
/w4fg3= sum(wd4fg3)
/w4fg4= sum(wd4fg4)
/w4fg5= sum(wd4fg5)
/w4fg6= sum(wd4fg6)
/w4fg7= sum(wd4fg7)
/w4fg8= sum(wd4fg8)
/w4fg9= sum(wd4fg9)
/w4fg10= sum(wd4fg10)
/w4fg11= sum(wd4fg11)
/w4fg12= sum(wd4fg12)
/w4fg13= sum(wd4fg13)
/w4fg14= sum(wd4fg14)
/w4fg15= sum(wd4fg15)
/w4fg16= sum(wd4fg16)
/w4fg17= sum(wd4fg17)
/w4fg18= sum(wd4fg18)
/w4fg19= sum(wd4fg19)
/w4fg20= sum(wd4fg20)
/w4fg21= sum(wd4fg21)
/w4fg22= sum(wd4fg22)
/w4fg23= sum(wd4fg23)
/w4fg24= sum(wd4fg24)
/w4fg25= sum(wd4fg25)
/w4fg26= sum(wd4fg26)
/w4fg27= sum(wd4fg27)
/w4fg28= sum(wd4fg28)
/w4fg29= sum(wd4fg29)
/w4fg30= sum(wd4fg30)
/w4fg31= sum(wd4fg31)
/w4fg32= sum(wd4fg32)
/w4fg33= sum(wd4fg33)
/w4fg34= sum(wd4fg34)
/w4fg35= sum(wd4fg35)
/w4fg36= sum(wd4fg36)
/w4fg37= sum(wd4fg37)
/w4fg38= sum(wd4fg38)
/w4fg39= sum(wd4fg39)
/w4fg40= sum(wd4fg40)
/w4fg41= sum(wd4fg41)
/w4fg42= sum(wd4fg42)
/w4fg43= sum(wd4fg43)
/w4fg44= sum(wd4fg44)
/w4fg45= sum(wd4fg45)
/w4fg46= sum(wd4fg46)
/w4fg47= sum(wd4fg47)
/w4fg48= sum(wd4fg48)
/w4fg49= sum(wd4fg49)
/w4fg50= sum(wd4fg50)
```

```
/w4fg51= sum(wd4fg51)
/w4fg52= sum(wd4fg52)
/w4fg53= sum(wd4fg53)
/w4fg54= sum(wd4fg54)
/w4fg55= sum(wd4fg55)
/w4fg56= sum(wd4fg56)
/w4fg57= sum(wd4fg57)
/w4fg58= sum(wd4fg58)
/w4fg59= sum(wd4fg59)
/w4fg60= sum(wd4fg60)
/w4fg61= sum(wd4fg61)
/w4fg62= sum(wd4fg62)
/w4fg63= sum(wd4fg63)
/w4fg64= sum(wd4fg64)
/w4fg65= sum(wd4fg65)
/w4fg66= sum(wd4fg66)
/w4fg67= sum(wd4fg67)
/w4fg68= sum(wd4fg68)
/w4fg69= sum(wd4fg69)
/w4fg70= sum(wd4fg70)
/w4fg71= sum(wd4fg71)
/w4fg72= sum(wd4fg72)
/w4fg73= sum(wd4fg73)
/w4fg74= sum(wd4fg74)
/w4fg75= sum(wd4fg75)
/w4fg76= sum(wd4fg76)
/w4fg77= sum(wd4fg77)
/w4fg78= sum(wd4fg78)
/w4fg79= sum(wd4fg79)
/w4fg80= sum(wd4fg80)
/w4fg81= sum(wd4fg81)
/w4fg82= sum(wd4fg82)
/w4fg83= sum(wd4fg83)
/w4fg84= sum(wd4fg84)
/w4fg85= sum(wd4fg85)
/w4fg86= sum(wd4fg86)
/w4fg87= sum(wd4fg87)
/w4fg88= sum(wd4fg88)
/w4fg89= sum(wd4fg89)
/w4fg90= sum(wd4fg90)
/w4fg91= sum(wd4fg91)
/w4fg92= sum(wd4fg92)
/w4fg93= sum(wd4fg93)
/w4fg94= sum(wd4fg94)
/w4fg95= sum(wd4fg95)
/w4fg96= sum(wd4fg96)
/w4fg97= sum(wd4fg97)
/w4fg98= sum(wd4fg98)
/w4fg99= sum(wd4fg99)
/w4fg100= sum(wd4fg100)
/w4fg101= sum(wd4fg101)
/w4fg102= sum(wd4fg102)
/w4fg103= sum(wd4fg103)
/w4fg104= sum(wd4fg104)
/w4fg105= sum(wd4fg105)
/w4fg106= sum(wd4fg106)
/w4fg107= sum(wd4fg107)
/w4fg108= sum(wd4fg108)
/w4fg109= sum(wd4fg109)
/w4fg110= sum(wd4fg110)
/w4fg111= sum(wd4fg111)
/w4fg112= sum(wd4fg112)
/w4fg113= sum(wd4fg113)
/w4fg114= sum(wd4fg114)
/w4fg115= sum(wd4fg115)
/w4fg116= sum(wd4fg116)
/w4fg117= sum(wd4fg117)
/w4fg118= sum(wd4fg118)
/w4fg119= sum(wd4fg119)
/w4fg120= sum(wd4fg120)
/w4fg121= sum(wd4fg121)
/w4fg122= sum(wd4fg122)
/w4fg123= sum(wd4fg123)
/w4fg124= sum(wd4fg124)
/day = nu(dayno).
```

```
get
file = 'C:\day-food level fg4_2.sav'
compute wa4fg1=w4fg1/day.
compute wa4fg2=w4fg2/day.
compute wa4fg3=w4fg3/day.
compute wa4fg4=w4fg4/day.
compute wa4fg5=w4fg5/day.
compute wa4fg6=w4fg6/day.
compute wa4fg7=w4fg7/day.
compute wa4fg8=w4fg8/day.
compute wa4fg9=w4fg9/day.
compute wa4fg10=w4fg10/day.
compute wa4fg11=w4fg11/day.
compute wa4fg12=w4fg12/day.
compute wa4fg13=w4fg13/day.
compute wa4fg14=w4fg14/day.
compute wa4fg15=w4fg15/day.
compute wa4fg16=w4fg16/day.
compute wa4fg17=w4fg17/day.
compute wa4fg18=w4fg18/day.
compute wa4fg19=w4fg19/day.
compute wa4fg20=w4fg20/day.
compute wa4fg21=w4fg21/day.
compute wa4fg22=w4fg22/day.
compute wa4fg23=w4fg23/day.
compute wa4fg24=w4fg24/day.
compute wa4fg25=w4fg25/day.
compute wa4fg26=w4fg26/day.
compute wa4fg27=w4fg27/day.
compute wa4fg28=w4fg28/day.
compute wa4fg29=w4fg29/day.
compute wa4fg30=w4fg30/day.
compute wa4fg31=w4fg31/day.
compute wa4fg32=w4fg32/day.
compute wa4fg33=w4fg33/day.
compute wa4fg34=w4fg34/day.
compute wa4fg35=w4fg35/day.
compute wa4fg36=w4fg36/day.
compute wa4fg37=w4fg37/day.
compute wa4fg38=w4fg38/day.
compute wa4fg39=w4fg39/day.
compute wa4fg40=w4fg40/day.
compute wa4fg41=w4fg41/day.
compute wa4fg42=w4fg42/day.
compute wa4fg43=w4fg43/day.
compute wa4fg44=w4fg44/day.
compute wa4fg45=w4fg45/day.
compute wa4fg46=w4fg46/day.
compute wa4fg47=w4fg47/day.
compute wa4fg48=w4fg48/day.
compute wa4fg49=w4fg49/day.
compute wa4fg50=w4fg50/day.
compute wa4fg51=w4fg51/day.
compute wa4fg52=w4fg52/day.
compute wa4fg53=w4fg53/day.
compute wa4fg54=w4fg54/day.
compute wa4fg55=w4fg55/day.
compute wa4fg56=w4fg56/day.
compute wa4fg57=w4fg57/day.
compute wa4fg58=w4fg58/day.
compute wa4fg59=w4fg59/day.
compute wa4fg60=w4fg60/day.
compute wa4fg61=w4fg61/day.
compute wa4fg62=w4fg62/day.
compute wa4fg63=w4fg63/day.
compute wa4fg64=w4fg64/day.
compute wa4fg65=w4fg65/day.
compute wa4fg66=w4fg66/day.
compute wa4fg67=w4fg67/day.
compute wa4fg68=w4fg68/day.
compute wa4fg69=w4fg69/day.
compute wa4fg70=w4fg70/day.
compute wa4fg71=w4fg71/day.
compute wa4fg72=w4fg72/day.
compute wa4fg73=w4fg73/day.
compute wa4fg74=w4fg74/day.
```

```
compute wa4fg75=w4fg75/day.
compute wa4fg76=w4fg76/day.
compute wa4fg77=w4fg77/day.
compute wa4fg78=w4fg78/day.
compute wa4fg79=w4fg79/day.
compute wa4fg80=w4fg80/day.
compute wa4fg81=w4fg81/day.
compute wa4fg82=w4fg82/day.
compute wa4fg83=w4fg83/day.
compute wa4fg84=w4fg84/day.
compute wa4fg85=w4fg85/day.
compute wa4fg86=w4fg86/day.
compute wa4fg87=w4fg87/day.
compute wa4fg88=w4fg88/day.
compute wa4fg89=w4fg89/day.
compute wa4fg90=w4fg90/day.
compute wa4fg91=w4fg91/day.
compute wa4fg92=w4fg92/day.
compute wa4fg93=w4fg93/day.
compute wa4fg94=w4fg94/day.
compute wa4fg95=w4fg95/day.
compute wa4fg96=w4fg96/day.
compute wa4fg97=w4fg97/day.
compute wa4fg98=w4fg98/day.
compute wa4fg99=w4fg99/day.
compute wa4fg100=w4fg100/day.
compute wa4fg101=w4fg101/day.
compute wa4fg102=w4fg102/day.
compute wa4fg103=w4fg103/day.
compute wa4fg104=w4fg104/day.
compute wa4fg105=w4fg105/day.
compute wa4fg106=w4fg106/day.
compute wa4fg107=w4fg107/day.
compute wa4fg108=w4fg108/day.
compute wa4fg109=w4fg109/day.
compute wa4fg110=w4fg110/day.
compute wa4fg111=w4fg111/day.
compute wa4fg112=w4fg112/day.
compute wa4fg113=w4fg113/day.
compute wa4fg114=w4fg114/day.
compute wa4fg115=w4fg115/day.
compute wa4fg116=w4fg116/day.
compute wa4fg117=w4fg117/day.
compute wa4fg118=w4fg118/day.
compute wa4fg119=w4fg119/day.
compute wa4fg120=w4fg120/day.
compute wa4fg121=w4fg121/day.
compute wa4fg122=w4fg122/day.
compute wa4fg123=w4fg123/day.
compute wa4fg124=w4fg124/day.
execute.

sort cases by shserial(a) respno(a).

SAVE OUTFILE='C:\aggr person food level fg4.sav'
 /COMPRESSED.

GET FILE='C:\aggr person food level fg4.sav'.

compute frcomp=wa4fg13*45/100+wa4fg17*61/100.
execute.
var labels
frcomp 'Fruit in composite dishes, grams'.

compute fruit=wa4fg81+wa4fg82+wa4fg83+wa4fg84*66/100+wa4fg85*65/100+wa4fg86+ frcomp.
execute.
var labels
fruit 'Total fruit consumed inc comp dishes, excl fr juice, grams'.

compute frport=fruit/80.
execute.
var labels
frport 'Fruit portions excluding fruit juice'.

compute frjport=wa4fg84*34/100+wa4fg93.
execute.
```

```
compute frjport=frjport/157.
execute.
var labels
frjport 'Fruit juice portions'.

RECODE
  frjport
  (Lowest thru 1=Copy)  (ELSE=1)  INTO  frprmax1 .
VARIABLE LABELS frprmax1 'Fruit portions maximum 1'.
EXECUTE .

compute frtport=frport+frprmax1.
execute.
var labels
frtport '(D) Fruit and fruit juice portions, juice counts once'.
```

VEGBBPO (D) Total portions of veg, beans and pulses portion counts once only
FRVEGPO (D) Total fruit and veg portions

```
SPSS Syntax

GET FILE='C:\aggr person food level fg4.sav'.

compute vegcomp=wa4fg45*19/100+wa4fg47*29/100+wa4fg49*29/100+wa4fg52*30/100.
execute.

var labels
vegcomp 'Veg in composite meat dishes'.

compute vegdis=wa4fg72*40/100.
execute.

var labels
vegdis 'Veg in vegetable dishes'.

compute vegtot=vegcomp+vegdis +
wa4fg63+wa4fg64+wa4fg65+wa4fg66+wa4fg67+wa4fg69+wa4fg70+wa4fg71+wa4fg75.
execute.
var labels
vegtot 'Veg consumption inc veg in dishes excluding beans and pulses, grams'.

compute vegpor=vegtot/80.
execute.
var labels
vegpor 'Veg portions excluding beans and pulses'.

compute bbpul=wa4fg68+wa4fg73.
compute bbpulpo=bbpul/80.
execute.
var labels
bbpulpo 'Portions of beans and pulses consumed'.


RECODE
  bbpulpo
  (Lowest thru 1=Copy)  (ELSE=1)  INTO  bpulpo1 .
VARIABLE LABELS bpulpo1 'Bean and pulses portion, max one'.
EXECUTE .

compute vegbbpo=vegpor+bpulpo1.
execute.
var labels
vegbbpo '(D) Total portions of veg, beans and pulses portion counts once only'.

compute frvegpo=vegbbpo+frtport.
execute.
var labels
frvegpo '(D) Total fruit and veg portions'

*Merge FRTPORT, VEGBBPO, FRVEGPO variables into person level nutrient and food.sav'.

GET
```

```
  FILE='C:\aggr person food level fg4.sav'.

SAVE OUTFILE='C:\portions.sav'
/ KEEP = shserial serp frtport vegbpo frvegpo.

GET
  FILE='C:\portions.sav’.

sort cases by shserial (A) serp (A).

**save file**.

GET
  FILE='C:\person level nutrient and food.sav’.

sort cases by shserial (A) serp (A).

MATCH FILES /FILE=*
 /TABLE='C:\portions.sav'
 /BY shserial serp.
EXECUTE.
```

WTA6FG1 (D) Pasta, rice, pizza and other cereals
WTA6FG2 (D) White and other bread
WTA6FG3 (D) Wholemeal bread
WTA6FG4 (D) Wholegrain and high fibre cereals
WTA6FG5 (D) Other breakfast cereals
WTA6FG6 (D) Biscuits, fruit pies, buns, cakes and pastries
WTA6FG7 (D) Puddings including ice cream
WTA6FG8 (D) Milk and cream
WTA6FG9 (D) Cheese
WTA6FG10 (D) Yoghurt, fromage frais and dairy desserts
WTA6FG11 (D) Eggs and egg dishes
WTA6FG12 (D) Fats and oils
WTA6FG13 (D) Meats and meat dishes, excluding processed meat
WTA6FG14 (D) Processed meat including sausages, burgers, coated chicken
WTA6FG15 (D) Fish and fish dishes, excluding oily fish
WTA6FG16 (D) Oily fish and dishes
WTA6FG17 (D) Vegetables excluding potatoes and baked bean
WTA6FG18 (D) Baked beans
WTA6FG19 (D) Chips, fried and roast potatoes and fried potato products
WTA6FG20 (D) Other potatoes, potato salads and dishes, potato products cooked without fat
WTA6FG21 (D) Crisps and savory snacks
WTA6FG22 (D) Fruit, excluding fruit juice
WTA6FG23 (D) Nuts and seeds
WTA6FG24 (D) Sugar, preserves and confectionery
WTA6FG25 (D) Fruit juice
WTA6FG26 (D) Soft drinks, not diet
WTA6FG27 (D) Soft drinks, diet
WTA6FG28 (D) Alcoholic drinks, including low alcohol
WTA6FG29 (D) Tea, coffee and water
WTA6FG30 (D) Beverages (dry wt), soups and sauces

```
SPSS Syntax

GET
  FILE='C:\person level nutrient and food.sav'.


compute wta6fg1=wta5fg1+wta5fg2+wta5fg3+wta5fg4.
compute wta6fg2=wta5fg5+wta5fg7.
compute wta6fg3=wta5fg6.
compute wta6fg4=wta5fg8.
compute wta6fg5=wta5fg9.
compute wta6fg6=wta5fg10+wta5fg11+wta5fg12.
compute wta6fg7=wta5fg13+wta5fg20.
compute wta6fg8=wta5fg14+wta5fg15+wta5fg16+wta5fg17.
compute wta6fg9=wta5fg18.
compute wta6fg10=wta5fg19.
compute wta6fg11=wta5fg21.
```

```
compute
wta6fg12=wta5fg22+wta5fg23+wta5fg24+wta5fg25+wta5fg26+wta5fg27+wta5fg28+wta5fg29+wta5f
g30.
compute wta6fg13=wta5fg31+wta5fg32+wta5fg34+wta5fg39.
compute wta6fg14=wta5fg33+wta5fg35+wta5fg36+wta5fg37+wta5fg38.
compute wta6fg15=wta5fg40+wta5fg41+wta5fg42+wta5fg43.
compute wta6fg16=wta5fg44.
compute
wta6fg17=wta5fg45+wta5fg46+wta5fg47+wta5fg48+wta5fg50+wta5fg51+wta5fg52+wta5fg53.
compute wta6fg18=wta5fg49.
compute wta6fg19=wta5fg54.
compute wta6fg20=wta5fg55.
compute wta6fg21=wta5fg56.
compute wta6fg22=wta5fg57+wta5fg58+wta5fg59+wta5fg60.
compute wta6fg23=wta5fg61.
compute wta6fg24=wta5fg62+wta5fg63+wta5fg64+wta5fg65.
compute wta6fg25=wta5fg66.
compute wta6fg26=wta5fg67+wta5fg68.
compute wta6fg27=wta5fg69+wta5fg70.
compute wta6fg28=wta5fg71+wta5fg72+wta5fg73+wta5fg74+wta5fg75.
compute wta6fg29=wta5fg76+wta5fg77+wta5fg78+wta5fg79.
compute wta6fg30=wta5fg80+wta5fg81+wta5fg82.

var labels
wta6fg1 '(D) Pasta, rice, pizza and other cereals'
wta6fg2 '(D) White and other bread'
wta6fg3 '(D) Wholemeal bread'
wta6fg4 '(D) Wholegrain and high fibre cereals'
wta6fg5 '(D) Other breakfast cereals'
wta6fg6 '(D) Biscuits, fruit pies, buns, cakes and pastries'
wta6fg7 '(D) Puddings including ice cream'
wta6fg8 '(D) Milk and cream'
wta6fg9 '(D) Cheese'
wta6fg10 '(D) Yoghurt, fromage frais and dairy desserts'
wta6fg11 '(D) Eggs and egg dishes'
wta6fg12 '(D) Fats and oils'
wta6fg13 '(D) Meats and meat dishes, excluding processed meat'
wta6fg14 '(D) Processed meat including sausages, burgers, coated chicken'
wta6fg15 '(D) Fish and fish dishes, excluding oily fish'
wta6fg16 '(D) Oily fish and dishes'
wta6fg17 '(D) Vegetables excluding potatoes and baked bean'
wta6fg18 '(D) Baked beans'
wta6fg19 '(D) Chips, fried and roast potatoes and fried potato products'
wta6fg20 '(D) Other potatoes, potato salads and dishes, potato products cooked without
fat'
wta6fg21 '(D) Crisps and savory snacks'
wta6fg22 '(D) Fruit, excluding fruit juice'
wta6fg23 '(D) Nuts and seeds'
wta6fg24 '(D) Sugar, preserves and confectionery'
wta6fg25 '(D) Fruit juice'
wta6fg26 '(D) Soft drinks, not diet'
wta6fg27 '(D) Soft drinks, diet'
wta6fg28 '(D) Alcoholic drinks, including low alcohol'
wta6fg29 '(D) Tea, coffee and water'
wta6fg30 '(D) Beverages (dry wt), soups and sauces'.
```

Chapter 5: Energy Intake

See Chapter 4 for percentage contribution.

ALCMJA (D) Average Energy content of alcohol consumed in MJ from food sources only

ALCKCALA (D) Average Energy content of alcohol consumed in kcal from food sources only

NF8BXALA (D) Average Energy (MJ excluding alcohol g) from food sources only

NF7XALA (D) Average Energy (kcal excluding alcohol g) from food sources only

```
GET FILE='C:\day level data.sav'.

compute alcmj=nf9*0.029288.
compute alckcal=nf9*7.
compute nf8bxalc=nf8b-alcmj.
compute nf7xalc=nf7-alckcal.

VARIABLE LABELS alcmj '(I) Energy content of alcohol consumed in MJ/day from food
sources only'.
VARIABLE LABELS alckcal '(I) Energy content of alchol consumed in kcal/day from food
sources only'.
VARIABLE LABELS nf8bxalc '(I) Energy (MJ/day excluding alcohol g) from food sources
only'.
VARIABLE LABELS nf7xalc '(I) Energy (kcal/day excluding alcohol g) from food sources
only'.

sort cases by shserial serp.

AGGREGATE
  /OUTFILE='C:\aggr person energy_alcohol 04.07.06.sav'
  /BREAK=shserial serp
  /alcmja = MEAN(alcmj) /alckcala = MEAN(alckcal) /nf8bxala = MEAN(nf8bxalc)
  /nf7xala = MEAN(nf7xalc).

SAVE OUTFILE='C:\day level data.sav'.
  /COMPRESSED.

GET
  FILE='C:\aggr person energy_alcohol 04.07.06.sav'.


sort cases by shserial serp.

SAVE OUTFILE='C:\aggr person energy_alcohol 04.07.06.sav'
  /COMPRESSED.


GET
  FILE='C:\person level nutrient and food.sav.

sort cases by shserial serp.

SAVE OUTFILE='C:\person level nutrient and food.sav
  /COMPRESSED.

MATCH FILES /FILE=*
 /TABLE='C:\aggr person energy_alcohol 04.07.06.sav'
 /BY shserial serp.
EXECUTE.

VARIABLE LABELS alcmja '(D) Average Energy content of alcohol consumed in MJ from food
sources only'.
VARIABLE LABELS alckcala '(D) Average Energy content of alcohol consumed in kcal from
food sources only'.
VARIABLE LABELS nf8bxala '(D) Average Energy (MJ excluding alcohol g) from food
sources only'.
```

```
VARIABLE LABELS nf7xala '(D) Average Energy (kcal excluding alcohol g) from food
sources only'.
```

Percentage contribution

The syntax provided calculates the percent contribution of food groups to energy. In order to compute the percent contribution for other nutrients, several variables in the example syntax need to be replaced. Each nutrient in the food level file is identifiable by a number e.g. protein = 4, vitamin A = 33. The table (below) lists the variables in the example syntax which need to be replaced: asterix denotes the number of the nutrient for which percent contribution is being calculated.

The syntax provided for energy is an example: no derived variables have been archived in the dataset.

Variables	Example for Energy kcal/nutrient 7
Food group 1	*Food group 1*
np*_sum	np7_sum
np*	np7
n*	n7
n*g1_1 to n*g1_13	n7g1_1 to n7g1_13
n*f1_1 to n*f1_13	n7f1_1 to n7f1_13
n*f1p1 to n*f1p13	n7f1p1 to n7f1p13
nfa*	nfa7
n*f1pt	n7f1pt
Food group 5	*Food group 5*
n*	n7
np*tot	np7tot
np*	np7
n*g5_1 to n*g5_84	n7g5_1 to n7g5_84
n*f5_1 to n*f5_84	n7f5_1 to n7f5_84
n*f5_t	n7f5_t
n*f5p1 to n*f5p84	n7f5p1 to n7f5p84
nfa*	nfa7
n*f5pt	n7f5pt

Percent contribution from food group 1a (13 groups)
N7F1P1 (D) Energy : % contribution from Cereals and cereal products
N7F1P2 (D) Energy : % contribution from Milk and milk products
N7F1P3 (D) Energy : % contribution from Eggs and Egg dishes
N7F1P4 (D) Energy : % contribution from Fat spreads
N7F1P5 (D) Energy : % contribution from Meat and meat products
N7F1P6 (D) Energy : % contribution from Fish and fish dishes
N7F1P7 (D) Energy : % contribution from Vegetables (not potatoes & savoury snacks)
N7F1P8 (D) Energy : % contribution from Potatoes & savoury snacks
N7F1P9 (D) Energy : % contribution from Fruit and nuts
N7F1P10 (D) Energy : % contribution from Sugar, preserves and confectionery
N7F1P11 (D) Energy : % contribution from Total beverages
N7F1P12 (D) Energy : % contribution from Miscellaneous
N7F1P13 (D) Energy : % contribution from Supplements
N7F1PT (D) Energy : % contribution from total fg1 (=100)

Percent contribution from food group 5 (84 groups)
N7F5P1 (D) Energy : % contribution from Pasta
N7F5P2 (D) Energy : % contribution from Rice
N7F5P3 (D) Energy : % contribution from Pizza
N7F5P4 (D) Energy : % contribution from Other cereals
N7F5P5 (D) Energy : % contribution from White bread
N7F5P6 (D) Energy : % contribution from Wholemeal bread
N7F5P7 (D) Energy : % contribution from Other breads
N7F5P8 (D) Energy : % contribution from Wholegrain and high fibre breakfast cereals
N7F5P9 (D) Energy : % contribution from Other breakfast cereals
N7F5P10 (D) Energy : % contribution from Biscuits
N7F5P11 (D) Energy : % contribution from Fruit pies
N7F5P12 (D) Energy : % contribution from Buns, cakes and pastries
N7F5P13 (D) Energy : % contribution from Cereal based milk puddings, sponge & other puddings
N7F5P14 (D) Energy : % contribution from Whole milk
N7F5P15 (D) Energy : % contribution from Semi-skimmed milk
N7F5P16 (D) Energy : % contribution from Skimmed milk
N7F5P17 (D) Energy : % contribution from Other milk and cream
N7F5P18 (D) Energy : % contribution from Cheese including cottage cheese

N7F5P19 (D) Energy : % contribution from Yoghurt, fromage frais and other dairy desserts
N7F5P20 (D) Energy : % contribution from Icecream
N7F5P21 (D) Energy : % contribution from Eggs and egg dishes
N7F5P22 (D) Energy : % contribution from Butter
N7F5P23 (D) Energy : % contribution from Block margarine
N7F5P24 (D) Energy : % contribution from Soft margarine not polyunsaturated
N7F5P25 (D) Energy : % contribution from Polyunsaturated margarine
N7F5P26 (D) Energy : % contribution from Other cooking fats and oils, not polyunsaturated
N7F5P27 (D) Energy : % contribution from Reduced fat spread, not polyunsaturated
N7F5P28 (D) Energy : % contribution from Reduced fat spread, polyunsaturated
N7F5P29 (D) Energy : % contribution from Low fat spread, not polyunsaturated
N7F5P30 (D) Energy : % contribution from Low fat spread polyunsaturated
N7F5P31 (D) Energy : % contribution from Bacon and ham
N7F5P32 (D) Energy : % contribution from Beef, veal, lamb and pork dishes
N7F5P33 (D) Energy : % contribution from Coated chicken and turkey
N7F5P34 (D) Energy : % contribution from Chicken and turkey dishes
N7F5P35 (D) Energy : % contribution from Burgers and kebabs
N7F5P36 (D) Energy : % contribution from Sausages
N7F5P37 (D) Energy : % contribution from Meat pies and pastries
N7F5P38 (D) Energy : % contribution from Other meat and meat products
N7F5P39 (D) Energy : % contribution from Liver, products and dishes
N7F5P40 (D) Energy : % contribution from White fish coated or fried including fish fingers
N7F5P41 (D) Energy : % contribution from Other white fish and fish dishes
N7F5P42 (D) Energy : % contribution from Shellfish and dishes
N7F5P43 (D) Energy : % contribution from Tuna (canned) and dishes (canned)
N7F5P44 (D) Energy : % contribution from Oily fish and dishes
N7F5P45 (D) Energy : % contribution from Carrots (raw)
N7F5P46 (D) Energy : % contribution from Salad and other vegetables (raw)
N7F5P47 (D) Energy : % contribution from Tomatoes (raw)
N7F5P48 (D) Energy : % contribution from Peas and green beans (not raw)
N7F5P49 (D) Energy : % contribution from Baked beans
N7F5P50 (D) Energy : % contribution from Leafy green vegetables (not raw)
N7F5P51 (D) Energy : % contribution from Carrots (not raw)
N7F5P52 (D) Energy : % contribution from Tomatoes (not raw)

N7F5P53 (D) Energy : % contribution from Other vegetables and vegetable dishes (not raw)
N7F5P54 (D) Energy : % contribution from Chips, fried & roast potatoes,fried potato products
N7F5P55 (D) Energy : % contribution from Other potatoes & dishes, pot prods ckd without fat
N7F5P56 (D) Energy : % contribution from Crisps and savoury snacks
N7F5P57 (D) Energy : % contribution from Apples and pears not canned
N7F5P58 (D) Energy : % contribution from Citrus fruit not canned
N7F5P59 (D) Energy : % contribution from Bananas
N7F5P60 (D) Energy : % contribution from Other fruit including canned
N7F5P61 (D) Energy : % contribution from Nuts and seeds
N7F5P62 (D) Energy : % contribution from Sugar
N7F5P63 (D) Energy : % contribution from Preserves, sweet spreads, fillings and icings
N7F5P64 (D) Energy : % contribution from Sugar confectionery
N7F5P65 (D) Energy : % contribution from Chocolate confectionery
N7F5P66 (D) Energy : % contribution from Fruit juice
N7F5P67 (D) Energy : % contribution from Conc soft dk as eaten & rtd soft dk, not diet
N7F5P68 (D) Energy : % contribution from Carbonated soft drinks not diet
N7F5P69 (D) Energy : % contribution from Conc soft dk as eaten & rtd soft dk, diet
N7F5P70 (D) Energy : % contribution from Carbonated soft drinks, diet
N7F5P71 (D) Energy : % contribution from Spirits and liquers
N7F5P72 (D) Energy : % contribution from Wine including fortified, low alcohol, alcohol free
N7f5p73 (D) Energy : % contribution from Beers and lager including low alcohol, alcohol free
N7f5p74 (D) Energy : % contribution from Cider and perry
N7F5P75 (D) Energy : % contribution from Alcoholic soft drinks
N7F5P76 (D) Energy : % contribution from Coffee (made up)
N7F5P77 (D) Energy : % contribution from Tea (made up)
N7F5P78 (D) Energy : % contribution from Herbal tea (made up)
N7F5P79 (D) Energy : % contribution from Tap and bottled water
N7F5P80 (D) Energy : % contribution from Beverages (dry weight) eg drinking choc, cocoa, etc
N7F5P81 (D) Energy : % contribution from Soups
N7F5P82 (D) Energy : % contribution from Savoury sauces, pickles, gravies, condiments
N7F5P83 (D) Energy : % contribution from Supplements and toddler food codes not reassigned
N7F5P84 (D) Energy : % contribution from Artificial sweeteners

N7F5PT (D) Energy : % contribution from total fg5 (=100) .

```
GET
  FILE='C:\Food level data.sav'.

USE ALL.
COMPUTE filter_$=(fg2<>54).
VARIABLE LABEL filter_$ 'fg2<>54 (FILTER)'.
VALUE LABELS filter_$  0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .

AGGREGATE
  /OUTFILE='C:\aggr n7 fg1a personlev_1.sav'
  /BREAK=shserial respno fg1a
  /np7_sum = SUM(np7).

GET
  FILE='C:\aggr n7 fg1a personlev_1.sav'.

if (fg1a=1)n7g1_1=np7_sum.
if (fg1a=2)n7g1_2=np7_sum.
if (fg1a=3)n7g1_3=np7_sum.
if (fg1a=4)n7g1_4=np7_sum.
if (fg1a=5)n7g1_5=np7_sum.
if (fg1a=6)n7g1_6=np7_sum.
if (fg1a=7)n7g1_7=np7_sum.
if (fg1a=8)n7g1_8=np7_sum.
if (fg1a=9)n7g1_9=np7_sum.
if (fg1a=10)n7g1_10=np7_sum.
if (fg1a=11)n7g1_11=np7_sum.
if (fg1a=12)n7g1_12=np7_sum.
if (fg1a=13)n7g1_13=np7_sum.
execute.

AGGREGATE
  /OUTFILE='C:\aggr n7 fg1a personlev_2.sav'
  /BREAK=shserial respno
  /n7f1_1 = SUM(n7g1_1)
 /n7f1_2 = SUM(n7g1_2)
 /n7f1_3 = SUM(n7g1_3)
 /n7f1_4 = SUM(n7g1_4)
 /n7f1_5 =  SUM(n7g1_5)
 /n7f1_6 = SUM(n7g1_6)
 /n7f1_7 = SUM(n7g1_7)
 /n7f1_8 = SUM(n7g1_8)
 /n7f1_9 = SUM(n7g1_9)
 /n7f1_10 = SUM(n7g1_10)
 /n7f1_11 = SUM(n7g1_11)
 /n7f1_12 = SUM(n7g1_12)
 /n7f1_13 = SUM(n7g1_13).

GET
  FILE='C:\aggr n7 fg1a personlev_2.sav'.

recode n7f1_1 to n7f1_13 (sysmis=0) (else=copy).
execute.


rename variables (respno=serp).

sort cases by shserial serp.

SAVE OUTFILE='C:\aggr n7 fg1a personlev_2.sav'
 /COMPRESSED.

GET
  FILE='C:\Person level nutrient and food.sav'

sort cases by shserial serp.

MATCH FILES /FILE=*
 /TABLE='C:\aggr n7 fg1a personlev_2.sav'
```

```
 /BY shserial serp.
EXECUTE.

GET
  FILE='C:\day level data.sav'.


freq dayno.

RECODE
  dayno  (1=1)  (2=1)  (3=1)  (4=1)  .
EXECUTE .

RECODE
  dayno
  (1 thru Highest=1)  INTO  dayno2 .
VARIABLE LABELS dayno2 'All days numbered as 1'.
EXECUTE .

freq dayno2.

AGGREGATE
  /OUTFILE='C:\aggr dayno2.sav'
  /BREAK=shserial serp
  /dayno2 = SUM(dayno2).

GET FILE ='C:\aggr dayno2.sav'.

sort cases by shserial serp.

GET
  FILE='C:\person level nutrient and food.sav'

sort cases by shserial serp.

MATCH FILES /FILE=*
 /TABLE='C:\aggr dayno2.sav'
 /BY shserial serp.
EXECUTE.


compute n7f1p1=n7f1_1/nfa7*100/dayno2.
compute n7f1p2=n7f1_2/nfa7*100/dayno2.
compute n7f1p3=n7f1_3/nfa7*100/dayno2.
compute n7f1p4=n7f1_4/nfa7*100/dayno2.
compute n7f1p5=n7f1_5/nfa7*100/dayno2.
compute n7f1p6=n7f1_6/nfa7*100/dayno2.
compute n7f1p7=n7f1_7/nfa7*100/dayno2.
compute n7f1p8=n7f1_8/nfa7*100/dayno2.
compute n7f1p9=n7f1_9/nfa7*100/dayno2.
compute n7f1p10=n7f1_10/nfa7*100/dayno2.
compute n7f1p11=n7f1_11/nfa7*100/dayno2.
compute n7f1p12=n7f1_12/nfa7*100/dayno2.
compute n7f1p13=n7f1_13/nfa7*100/dayno2.

compute n7f1pt=sum(n7f1p1 to n7f1p13).

var labels
n7f1p1 (D) Energy :  % contribution from Cereals and cereal products
n7f1p2 (D) Energy :  % contribution from Milk and milk products
n7f1p3 (D) Energy :  % contribution from Eggs and Egg dishes
n7f1p4 (D) Energy :  % contribution from Fat spreads
n7f1p5 (D) Energy :  % contribution from Meat and meat products
n7f1p6 (D) Energy :  % contribution from Fish and fish dishes
n7f1p7 (D) Energy :  % contribution from Vegetables (not potatoes & savoury snacks)

n7f1p8 (D) Energy :  % contribution from Potatoes & savoury snacks
n7f1p9 (D) Energy :  % contribution from Fruit and nuts
n7f1p10 (D) Energy :  % contribution from Sugar, preserves and confectionery
n7f1p11 (D) Energy :  % contribution from Total beverages
n7f1p12 (D) Energy :  % contribution from Miscellaneous
n7f1p13 (D) Energy :  % contribution from Supplements
n7f1pt (D) Energy :  % contribution from total fg1 (=100)   .
```

```
**Note: if nfa variable = 0 then we would have missing value for p1-p13 so replace
with zero

recode n7f1p1 to n7f1p13 (sysmis=0) (else=copy).

sort cases by shserial serp.


SAVE OUTFILE='C:\person level nutrient and food.sav'
 /COMPRESSED.

*********                              Food Group 5                  **********.


GET
  FILE='C:\Food level data.sav'.

USE ALL.
COMPUTE filter_$=(fg2<>54).
VARIABLE LABEL filter_$ 'fg2<>54 (FILTER)'.
VALUE LABELS filter_$  0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .

AGGREGATE
  /OUTFILE='C:\aggr n7 fg5 personlev_1.sav'
  /BREAK=shserial serp fg5
  /np7tot = SUM(np7).


GET
  FILE='C:\aggr n7 fg5 personlev_1.sav'.

if (fg5=1)n7g5_1=np7tot.
if (fg5=2)n7g5_2=np7tot.
if (fg5=3)n7g5_3=np7tot.
if (fg5=4)n7g5_4=np7tot.
if (fg5=5)n7g5_5=np7tot.
if (fg5=6)n7g5_6=np7tot.
if (fg5=7)n7g5_7=np7tot.
if (fg5=8)n7g5_8=np7tot.
if (fg5=9)n7g5_9=np7tot.
if (fg5=10)n7g5_10=np7tot.
if (fg5=11)n7g5_11=np7tot.
if (fg5=12)n7g5_12=np7tot.
if (fg5=13)n7g5_13=np7tot.
if (fg5=14)n7g5_14=np7tot.
if (fg5=15)n7g5_15=np7tot.
if (fg5=16)n7g5_16=np7tot.
if (fg5=17)n7g5_17=np7tot.
if (fg5=18)n7g5_18=np7tot.
if (fg5=19)n7g5_19=np7tot.
if (fg5=20)n7g5_20=np7tot.
if (fg5=21)n7g5_21=np7tot.
if (fg5=22)n7g5_22=np7tot.
if (fg5=23)n7g5_23=np7tot.
if (fg5=24)n7g5_24=np7tot.
if (fg5=25)n7g5_25=np7tot.
if (fg5=26)n7g5_26=np7tot.
if (fg5=27)n7g5_27=np7tot.
if (fg5=28)n7g5_28=np7tot.
if (fg5=29)n7g5_29=np7tot.
if (fg5=30)n7g5_30=np7tot.
if (fg5=31)n7g5_31=np7tot.
if (fg5=32)n7g5_32=np7tot.
if (fg5=33)n7g5_33=np7tot.
if (fg5=34)n7g5_34=np7tot.
if (fg5=35)n7g5_35=np7tot.
if (fg5=36)n7g5_36=np7tot.
if (fg5=37)n7g5_37=np7tot.
if (fg5=38)n7g5_38=np7tot.
if (fg5=39)n7g5_39=np7tot.
if (fg5=40)n7g5_40=np7tot.
if (fg5=41)n7g5_41=np7tot.
if (fg5=42)n7g5_42=np7tot.
```

```
if (fg5=43)n7g5_43=np7tot.
if (fg5=44)n7g5_44=np7tot.
if (fg5=45)n7g5_45=np7tot.
if (fg5=46)n7g5_46=np7tot.
if (fg5=47)n7g5_47=np7tot.
if (fg5=48)n7g5_48=np7tot.
if (fg5=49)n7g5_49=np7tot.
if (fg5=50)n7g5_50=np7tot.
if (fg5=51)n7g5_51=np7tot.
if (fg5=52)n7g5_52=np7tot.
if (fg5=53)n7g5_53=np7tot.
if (fg5=54)n7g5_54=np7tot.
if (fg5=55)n7g5_55=np7tot.
if (fg5=56)n7g5_56=np7tot.
if (fg5=57)n7g5_57=np7tot.
if (fg5=58)n7g5_58=np7tot.
if (fg5=59)n7g5_59=np7tot.
if (fg5=60)n7g5_60=np7tot.
if (fg5=61)n7g5_61=np7tot.
if (fg5=62)n7g5_62=np7tot.
if (fg5=63)n7g5_63=np7tot.
if (fg5=64)n7g5_64=np7tot.
if (fg5=65)n7g5_65=np7tot.
if (fg5=66)n7g5_66=np7tot.
if (fg5=67)n7g5_67=np7tot.
if (fg5=68)n7g5_68=np7tot.
if (fg5=69)n7g5_69=np7tot.
if (fg5=70)n7g5_70=np7tot.
if (fg5=71)n7g5_71=np7tot.
if (fg5=72)n7g5_72=np7tot.
if (fg5=73)n7g5_73=np7tot.
if (fg5=74)n7g5_74=np7tot.
if (fg5=75)n7g5_75=np7tot.
if (fg5=76)n7g5_76=np7tot.
if (fg5=77)n7g5_77=np7tot.
if (fg5=78)n7g5_78=np7tot.
if (fg5=79)n7g5_79=np7tot.
if (fg5=80)n7g5_80=np7tot.
if (fg5=81)n7g5_81=np7tot.
if (fg5=82)n7g5_82=np7tot.
if (fg5=83)n7g5_83=np7tot.
if (fg5=84)n7g5_84=np7tot.

execute.

AGGREGATE
  /OUTFILE='C:\aggr n7 fg5 personlev_2.sav'
  /BREAK=shserial serp
/n7f5_1=sum (n7g5_1)
/n7f5_2=sum (n7g5_2)
/n7f5_3=sum (n7g5_3)
/n7f5_4=sum (n7g5_4)
/n7f5_5=sum (n7g5_5)
/n7f5_6=sum (n7g5_6)
/n7f5_7=sum (n7g5_7)
/n7f5_8=sum (n7g5_8)
/n7f5_9=sum (n7g5_9)
/n7f5_10=sum (n7g5_10)
/n7f5_11=sum (n7g5_11)
/n7f5_12=sum (n7g5_12)
/n7f5_13=sum (n7g5_13)
/n7f5_14=sum (n7g5_14)
/n7f5_15=sum (n7g5_15)
/n7f5_16=sum (n7g5_16)
/n7f5_17=sum (n7g5_17)
/n7f5_18=sum (n7g5_18)
/n7f5_19=sum (n7g5_19)
/n7f5_20=sum (n7g5_20)
/n7f5_21=sum (n7g5_21)
/n7f5_22=sum (n7g5_22)
/n7f5_23=sum (n7g5_23)
/n7f5_24=sum (n7g5_24)
/n7f5_25=sum (n7g5_25)
/n7f5_26=sum (n7g5_26)
/n7f5_27=sum (n7g5_27)
/n7f5_28=sum (n7g5_28)
```

```
/n7f5_29=sum (n7g5_29)
/n7f5_30=sum (n7g5_30)
/n7f5_31=sum (n7g5_31)
/n7f5_32=sum (n7g5_32)
/n7f5_33=sum (n7g5_33)
/n7f5_34=sum (n7g5_34)
/n7f5_35=sum (n7g5_35)
/n7f5_36=sum (n7g5_36)
/n7f5_37=sum (n7g5_37)
/n7f5_38=sum (n7g5_38)
/n7f5_39=sum (n7g5_39)
/n7f5_40=sum (n7g5_40)
/n7f5_41=sum (n7g5_41)
/n7f5_42=sum (n7g5_42)
/n7f5_43=sum (n7g5_43)
/n7f5_44=sum (n7g5_44)
/n7f5_45=sum (n7g5_45)
/n7f5_46=sum (n7g5_46)
/n7f5_47=sum (n7g5_47)
/n7f5_48=sum (n7g5_48)
/n7f5_49=sum (n7g5_49)
/n7f5_50=sum (n7g5_50)
/n7f5_51=sum (n7g5_51)
/n7f5_52=sum (n7g5_52)
/n7f5_53=sum (n7g5_53)
/n7f5_54=sum (n7g5_54)
/n7f5_55=sum (n7g5_55)
/n7f5_56=sum (n7g5_56)
/n7f5_57=sum (n7g5_57)
/n7f5_58=sum (n7g5_58)
/n7f5_59=sum (n7g5_59)
/n7f5_60=sum (n7g5_60)
/n7f5_61=sum (n7g5_61)
/n7f5_62=sum (n7g5_62)
/n7f5_63=sum (n7g5_63)
/n7f5_64=sum (n7g5_64)
/n7f5_65=sum (n7g5_65)
/n7f5_66=sum (n7g5_66)
/n7f5_67=sum (n7g5_67)
/n7f5_68=sum (n7g5_68)
/n7f5_69=sum (n7g5_69)
/n7f5_70=sum (n7g5_70)
/n7f5_71=sum (n7g5_71)
/n7f5_72=sum (n7g5_72)
/n7f5_73=sum (n7g5_73)
/n7f5_74=sum (n7g5_74)
/n7f5_75=sum (n7g5_75)
/n7f5_76=sum (n7g5_76)
/n7f5_77=sum (n7g5_77)
/n7f5_78=sum (n7g5_78)
/n7f5_79=sum (n7g5_79)
/n7f5_80=sum (n7g5_80)
/n7f5_81=sum (n7g5_81)
/n7f5_82=sum (n7g5_82)
/n7f5_83=sum (n7g5_83)
/n7f5_84=sum (n7g5_84) .
execute.

GET
  FILE='C:\aggr n7 fg5 personlev_2.sav'.

***recode sysmis to zero before merging data into individual level file.

recode n7f5_1 to n7f5_84 (sysmis=0) (else=copy).
execute.

** *find sum of Energy (n7) content of daily intake from all food groups.

compute n7f5_t=sum(n7f5_1 to n7f5_84).


sort cases by shserial serp.

SAVE OUTFILE='C:\aggr n7 fg5 personlev_2.sav'
 /COMPRESSED.
```

```
GET
  FILE='C:\person level nutrient and food.sav'.

sort cases by shserial serp.

MATCH FILES /FILE=*
 /TABLE='C:\aggr n7 fg5 personlev_2.sav'
 /BY shserial serp.
EXECUTE.


*DERIVED VARIABLE to create percent contribution of food group to total intake by
respondent*.

compute n7f5p1=n7f5_1/nfa7*100/dayno2.
compute n7f5p2=n7f5_2/nfa7*100/dayno2.
compute n7f5p3=n7f5_3/nfa7*100/dayno2.
compute n7f5p4=n7f5_4/nfa7*100/dayno2.
compute n7f5p5=n7f5_5/nfa7*100/dayno2.
compute n7f5p6=n7f5_6/nfa7*100/dayno2.
compute n7f5p7=n7f5_7/nfa7*100/dayno2.
compute n7f5p8=n7f5_8/nfa7*100/dayno2.
compute n7f5p9=n7f5_9/nfa7*100/dayno2.
compute n7f5p10=n7f5_10/nfa7*100/dayno2.
compute n7f5p11=n7f5_11/nfa7*100/dayno2.
compute n7f5p12=n7f5_12/nfa7*100/dayno2.
compute n7f5p13=n7f5_13/nfa7*100/dayno2.
compute n7f5p14=n7f5_14/nfa7*100/dayno2.
compute n7f5p15=n7f5_15/nfa7*100/dayno2.
compute n7f5p16=n7f5_16/nfa7*100/dayno2.
compute n7f5p17=n7f5_17/nfa7*100/dayno2.
compute n7f5p18=n7f5_18/nfa7*100/dayno2.
compute n7f5p19=n7f5_19/nfa7*100/dayno2.
compute n7f5p20=n7f5_20/nfa7*100/dayno2.
compute n7f5p21=n7f5_21/nfa7*100/dayno2.
compute n7f5p22=n7f5_22/nfa7*100/dayno2.
compute n7f5p23=n7f5_23/nfa7*100/dayno2.
compute n7f5p24=n7f5_24/nfa7*100/dayno2.
compute n7f5p25=n7f5_25/nfa7*100/dayno2.
compute n7f5p26=n7f5_26/nfa7*100/dayno2.
compute n7f5p27=n7f5_27/nfa7*100/dayno2.
compute n7f5p28=n7f5_28/nfa7*100/dayno2.
compute n7f5p29=n7f5_29/nfa7*100/dayno2.
compute n7f5p30=n7f5_30/nfa7*100/dayno2.
compute n7f5p31=n7f5_31/nfa7*100/dayno2.
compute n7f5p32=n7f5_32/nfa7*100/dayno2.
compute n7f5p33=n7f5_33/nfa7*100/dayno2.
compute n7f5p34=n7f5_34/nfa7*100/dayno2.
compute n7f5p35=n7f5_35/nfa7*100/dayno2.
compute n7f5p36=n7f5_36/nfa7*100/dayno2.
compute n7f5p37=n7f5_37/nfa7*100/dayno2.
compute n7f5p38=n7f5_38/nfa7*100/dayno2.
compute n7f5p39=n7f5_39/nfa7*100/dayno2.
compute n7f5p40=n7f5_40/nfa7*100/dayno2.
compute n7f5p41=n7f5_41/nfa7*100/dayno2.
compute n7f5p42=n7f5_42/nfa7*100/dayno2.
compute n7f5p43=n7f5_43/nfa7*100/dayno2.
compute n7f5p44=n7f5_44/nfa7*100/dayno2.
compute n7f5p45=n7f5_45/nfa7*100/dayno2.
compute n7f5p46=n7f5_46/nfa7*100/dayno2.
compute n7f5p47=n7f5_47/nfa7*100/dayno2.
compute n7f5p48=n7f5_48/nfa7*100/dayno2.
compute n7f5p49=n7f5_49/nfa7*100/dayno2.
compute n7f5p50=n7f5_50/nfa7*100/dayno2.
compute n7f5p51=n7f5_51/nfa7*100/dayno2.
compute n7f5p52=n7f5_52/nfa7*100/dayno2.
compute n7f5p53=n7f5_53/nfa7*100/dayno2.
compute n7f5p54=n7f5_54/nfa7*100/dayno2.
compute n7f5p55=n7f5_55/nfa7*100/dayno2.
compute n7f5p56=n7f5_56/nfa7*100/dayno2.
compute n7f5p57=n7f5_57/nfa7*100/dayno2.
compute n7f5p58=n7f5_58/nfa7*100/dayno2.
compute n7f5p59=n7f5_59/nfa7*100/dayno2.
compute n7f5p60=n7f5_60/nfa7*100/dayno2.
compute n7f5p61=n7f5_61/nfa7*100/dayno2.
compute n7f5p62=n7f5_62/nfa7*100/dayno2.
```

```
compute n7f5p63=n7f5_63/nfa7*100/dayno2.
compute n7f5p64=n7f5_64/nfa7*100/dayno2.
compute n7f5p65=n7f5_65/nfa7*100/dayno2.
compute n7f5p66=n7f5_66/nfa7*100/dayno2.
compute n7f5p67=n7f5_67/nfa7*100/dayno2.
compute n7f5p68=n7f5_68/nfa7*100/dayno2.
compute n7f5p69=n7f5_69/nfa7*100/dayno2.
compute n7f5p70=n7f5_70/nfa7*100/dayno2.
compute n7f5p71=n7f5_71/nfa7*100/dayno2.
compute n7f5p72=n7f5_72/nfa7*100/dayno2.
compute n7f5p73=n7f5_73/nfa7*100/dayno2.
compute n7f5p74=n7f5_74/nfa7*100/dayno2.
compute n7f5p75=n7f5_75/nfa7*100/dayno2.
compute n7f5p76=n7f5_76/nfa7*100/dayno2.
compute n7f5p77=n7f5_77/nfa7*100/dayno2.
compute n7f5p78=n7f5_78/nfa7*100/dayno2.
compute n7f5p79=n7f5_79/nfa7*100/dayno2.
compute n7f5p80=n7f5_80/nfa7*100/dayno2.
compute n7f5p81=n7f5_81/nfa7*100/dayno2.
compute n7f5p82=n7f5_82/nfa7*100/dayno2.
compute n7f5p83=n7f5_83/nfa7*100/dayno2.
compute n7f5p84=n7f5_84/nfa7*100/dayno2.

compute n7f5pt=sum(n7f5p1 to n7f5p84).

var labels
n7f5p1 (D) Energy :  % contribution from Pasta
n7f5p2 (D) Energy :  % contribution from Rice
n7f5p3 (D) Energy :  % contribution from Pizza
n7f5p4 (D) Energy :  % contribution from Other cereals
n7f5p5 (D) Energy :  % contribution from White bread
n7f5p6 (D) Energy :  % contribution from Wholemeal bread
n7f5p7 (D) Energy :  % contribution from Other breads
n7f5p8 (D) Energy :  % contribution from Wholegrain and high fibre breakfast cereals

n7f5p9 (D) Energy :  % contribution from Other breakfast cereals
n7f5p10 (D) Energy :  % contribution from Biscuits
n7f5p11 (D) Energy :  % contribution from Fruit pies
n7f5p12 (D) Energy :  % contribution from Buns, cakes and pastries
n7f5p13 (D) Energy :  % contribution from Cereal based milk puddings, sponge & other
puddings
n7f5p14 (D) Energy :  % contribution from Whole milk
n7f5p15 (D) Energy :  % contribution from Semi-skimmed milk
n7f5p16 (D) Energy :  % contribution from Skimmed milk
n7f5p17 (D) Energy :  % contribution from Other milk and cream
n7f5p18 (D) Energy :  % contribution from Cheese including cottage cheese
n7f5p19 (D) Energy :  % contribution from Yoghurt, fromage frais and other dairy
desserts
n7f5p20 (D) Energy :  % contribution from Icecream
n7f5p21 (D) Energy :  % contribution from Eggs and egg dishes
n7f5p22 (D) Energy :  % contribution from Butter
n7f5p23 (D) Energy :  % contribution from Block margarine
n7f5p24 (D) Energy :  % contribution from Soft margarine not polyunsaturated
n7f5p25 (D) Energy :  % contribution from Polyunsaturated margarine
n7f5p26 (D) Energy :  % contribution from Other cooking fats and oils, not
polyunsaturated
n7f5p27 (D) Energy :  % contribution from Reduced fat spread, not polyunsaturated
n7f5p28 (D) Energy :  % contribution from Reduced fat spread, polyunsaturated
n7f5p29 (D) Energy :  % contribution from Low fat spread, not polyunsaturated
n7f5p30 (D) Energy :  % contribution from Low fat spread polyunsaturated
n7f5p31 (D) Energy :  % contribution from Bacon and ham
n7f5p32 (D) Energy :  % contribution from Beef, veal, lamb and pork dishes
n7f5p33 (D) Energy :  % contribution from Coated chicken and turkey
n7f5p34 (D) Energy :  % contribution from Chicken and turkey dishes
n7f5p35 (D) Energy :  % contribution from Burgers and kebabs
n7f5p36 (D) Energy :  % contribution from Sausages
n7f5p37 (D) Energy :  % contribution from Meat pies and pastries
n7f5p38 (D) Energy :  % contribution from Other meat and meat products
n7f5p39 (D) Energy :  % contribution from Liver, products and dishes
n7f5p40 (D) Energy :  % contribution from White fish coated or fried including fish
fingers
n7f5p41 (D) Energy :  % contribution from Other white fish and fish dishes
n7f5p42 (D) Energy :  % contribution from Shellfish and dishes
n7f5p43 (D) Energy :  % contribution from Tuna (canned) and dishes (canned)
n7f5p44 (D) Energy :  % contribution from Oily fish and dishes
n7f5p45 (D) Energy :  % contribution from Carrots (raw)
```

```
n7f5p46 (D) Energy :  % contribution from Salad and other vegetables (raw)
n7f5p47 (D) Energy :  % contribution from Tomatoes (raw)
n7f5p48 (D) Energy :  % contribution from Peas and green beans (not raw)
n7f5p49 (D) Energy :  % contribution from Baked beans
n7f5p50 (D) Energy :  % contribution from Leafy green vegetables (not raw)
n7f5p51 (D) Energy :  % contribution from Carrots (not raw)
n7f5p52 (D) Energy :  % contribution from Tomatoes (not raw)
n7f5p53 (D) Energy :  % contribution from Other vegetables and vegetable dishes (not
raw)
n7f5p54 (D) Energy :  % contribution from Chips, fried & roast potatoes,fried potato
products
n7f5p55 (D) Energy :  % contribution from Other potatoes & dishes, pot prods ckd
without fat
n7f5p56 (D) Energy :  % contribution from Crisps and savoury snacks
n7f5p57 (D) Energy :  % contribution from Apples and pears not canned
n7f5p58 (D) Energy :  % contribution from Citrus fruit not canned
n7f5p59 (D) Energy :  % contribution from Bananas
n7f5p60 (D) Energy :  % contribution from Other fruit including canned
n7f5p61 (D) Energy :  % contribution from Nuts and seeds
n7f5p62 (D) Energy :  % contribution from Sugar
n7f5p63 (D) Energy :  % contribution from Preserves, sweet spreads, fillings and
icings
n7f5p64 (D) Energy :  % contribution from Sugar confectionery
n7f5p65 (D) Energy :  % contribution from Chocolate confectionery
n7f5p66 (D) Energy :  % contribution from Fruit juice
n7f5p67 (D) Energy :  % contribution from Conc soft dk as eaten & rtd soft dk, not
diet
n7f5p68 (D) Energy :  % contribution from Carbonated soft drinks not diet
n7f5p69 (D) Energy :  % contribution from Conc soft dk as eaten & rtd soft dk, diet

n7f5p70 (D) Energy :  % contribution from Carbonated soft drinks, diet
n7f5p71 (D) Energy :  % contribution from Spirits and liquers
n7f5p72 (D) Energy :  % contribution from Wine including fortified, low alcohol,
alcohol free
n7f5p73 (D) Energy :  % contribution from Beers and lager including low alcohol,
alcohol free
n7f5p74 (D) Energy :  % contribution from Cider and perry
n7f5p75 (D) Energy :  % contribution from Alcoholic soft drinks
n7f5p76 (D) Energy :  % contribution from Coffee (made up)
n7f5p77 (D) Energy :  % contribution from Tea (made up)
n7f5p78 (D) Energy :  % contribution from Herbal tea (made up)
n7f5p79 (D) Energy :  % contribution from Tap and bottled water
n7f5p80 (D) Energy :  % contribution from Beverages (dry weight) eg drinking choc,
cocoa, etc
n7f5p81 (D) Energy :  % contribution from Soups
n7f5p82 (D) Energy :  % contribution from Savoury sauces, pickles, gravies, condiments

n7f5p83 (D) Energy :  % contribution from Supplements and toddler food codes not
reassigned
n7f5p84 (D) Energy :  % contribution from Artificial sweeteners
n7f5pt (D) Energy :  % contribution from total fg5 (=100)   .


***If nfa = 0, then our percent contribution variable would be missing value,, so
replace with ZERO.

recode n7f5p1 to n7f5p84 (sysmis=0) (else=copy).
```

Chapter 6: Protein, Carbohydrate, Non-starch

N.B. Previously derived variable 'nf7xala' created in Chapter 5 is required to derive variables in this chapter.

For percentage contribution syntax, see Chapter 5

ENFA4 (D) % food energy from protein
ENFA6 (D) % food energy from total carbohydrate
ENFA12 (D) % food energy from starch
ENFA13 (D) % food energy from total sugars
ENFA14 (D) % food energy from non-milk extrinsic sugars
ENFA15 (D) % food energy from intrinsic and milk sugars
ENFA1215 (D) % food energy from intrinsic sugars & milk sugars + starch

```
GET
  FILE='C:\person level nutrient and food.sav'.

COMPUTE enfa4=((nfa4*4)/nf7xala)*100.
VAR LAB enfa4 ' (D) % food energy from protein'.

COMPUTE enfa6=((nfa6*3.75)/nf7xala)*100.
VAR LAB enfa6 ' (D) % food energy from total carbohydrate'.

COMPUTE enfa12=((nfa12*3.75)/nf7xala)*100.
VAR LAB enfa12 ' (D) % food energy from starch'.

COMPUTE enfa13=((nfa13*3.75)/nf7xala)*100.
VAR LAB enfa13 '(D) % food energy from total sugars'.

COMPUTE enfa14=((nfa14*3.75)/nf7xala)*100.
VAR LAB enfa14 '(D) % food energy from non-milk extrinsic sugars'.

COMPUTE enfa15=((nfa15*3.75)/nf7xala)*100.
VAR LAB enfa15 '(D) % food energy from intrinsic and milk sugars'.

COMPUTE enfa1215=(((nfa12+nfa15)*3.75)/nf7xala)*100.
VAR LAB enfa1215 '(D)% food energy from intrinsic sugars & milk sugars + starch'.
```

NFA10G (D) Grouped Englyst fibre - grams

```
 1 <6
 2 <8
 3 <10
 4 <12
 5 <14
 6 <16
 7 <18
 8 <20
 9 <22
 10 <24
 11 24 or more
```

```
GET
  FILE='C:\person level nutrient and food.sav'.

RECODE nfa10 (lo thru 5.9999=1) (6 thru 7.9999=2) (8 thru 9.9999=3) (10 thru
11.9999=4) (12 thru 13.9999=5) (14 thru 15.9999=6) (16 thru 17.9999=7)
  (18 thru 19.9999=8) (20 thru 21.9999=9) (22 thru 23.9999=10) (24 thru hi=11) INTO
nfa10g.
VAR LAB nfa10g '(D) Grouped Englyst fibre - grams'.
VAL LAB nfa10g
```

```
1'<6'
2'<8'
3'<10'
4'<12'
5'<14'
6'<16'
7'<18'
8'<20'
9'<22'
10'<24'
11'24 or more'.
```

ENTA9A (D) % total energy from alcohol

```
GET
  FILE='C:\person level nutrient and food.sav'.

COMPUTE enta9a=((nta9*7)/nta7)*100.
VAR LAB enta9A '(D) % total energy from alcohol'.
```

NFA9G (D) Grouped Alcohol consumption in grams

```
0 0
1 less than 1.0
2 less than 2.0
3 less than 5.0
4 less than 10.0
5 less than 15.0
6 less than 20.0
7 less than 30.0
8 less than 40.0
9 less than 50.0
10 less than 60.0
11 60+
```

OHFLAG (D) Flag for alcohol consumption

```
0 non consumer
1 drinks alcohol
```

```
GET FILE='C:\person level nutrient and food.sav'.

RECODE nfa9 (0=0) (0 thru 0.9999=1) (1 thru 1.9999=2) (2 thru 4.9999=3) (5 thru
9.9999=4) (10 thru 14.9999=5) (15 thru 19.9999=6)
  (20 thru 29.9999=7) (30 thru 39.9999=8) (40 thru 49.9999=9) (50 thru 59.9999=10) (60
thru hi=11) INTO nfa9g.
VAR LAB nfa9g '(D) Grouped Alcohol consumption in grams'.
VAL LAB nfa9g
0'0'
1'less than 1.0'
2'less than 2.0'
3'less than 5.0'
4'less than 10.0'
5'less than 15.0'
6'less than 20.0'
7'less than 30.0'
8'less than 40.0'
9'less than 50.0'
10'less than 60.0'
11'60+'.


DO IF nfa9g=0 & agep>=19 .
COMPUTE ohflag=0.
ELSE IF nfa9g>0 & agep>=19.
COMPUTE ohflag=1.
END IF.
VAR LAB ohflag '(D) Flag for alcohol consumption'.
VAL LAB ohflag 0'non consumer' 1'drinks alcohol'.
```

Chapter 7: Fat and Fatty Acids

N.B. Previously derived variable ‘nf7xala’ created in Chapter 5 is required to derive variables in this chapter.

For percentage contribution syntax, see Chapter 5

DF5 (D) % food energy from fat
DF22 (D) % food energy from Satd FA
DF23 (D) % food energy from Cis-Mon FA
DF24 (D) % food energy from Cis-n3 FA
DF25 (D) % food energy from Cis-n6 FA
DF26 (D) % food energy from Trans FA
DF27 (D) % food energy from cis PUFA
DF5DRVP (D) Energy from total fat (kcal/day food only) as % of DRV (35E%)
DF22DRVP (D) Energy from SFA (kcal/day) as % of DRV (11E%)
DF23DRVP (D) Energy from MUFA (kcal/day) as % of DRV (13E%)
DF27DRVP (D) Energy from PUFA (kcal/day) as % of DRV (6.5E%)
DF26DRVP (D) Energy from TRANS FA (kcal/day) as % of DRV (2E%)

```
GET
  FILE='C:\person level nutrient and food.sav'.

COMPUTE df5 = (( nfa5* 9) / nf7xala) * 100 .
VARIABLE LABELS df5 '(D) % food energy from fat' .
EXECUTE .

COMPUTE df22 = (( nfa22* 9) / nf7xala) * 100 .
VARIABLE LABELS df22 '(D) % food energy from Satd FA' .
EXECUTE .

COMPUTE df23 = (( nfa23* 9) / nf7xala) * 100 .
VARIABLE LABELS df23 '(D) % food energy from Cis-Mon FA' .
EXECUTE .

COMPUTE df24 = (( nfa24* 9) / nf7xala) * 100 .
VARIABLE LABELS df24 '(D) % food energy from Cis-n3 FA' .
EXECUTE .

COMPUTE df25 = (( nfa25* 9) / nf7xala) * 100 .
VARIABLE LABELS df25 '(D) % food energy from Cis-n6 FA' .
EXECUTE .

COMPUTE df26 = (( nfa26* 9) / nf7xala) * 100 .
VARIABLE LABELS df26 '(D) % food energy from Trans FA' .
EXECUTE .

COMPUTE df27 = (( (nfa24+nfa25)* 9) / nf7xala) * 100 .
VARIABLE LABELS df27 '(D) % food energy from cis PUFA' .
EXECUTE .

COMPUTE df5drvp = (df5 / 35*100).
VARIABLE LABELS df5drvp '(D) Energy from total fat (kcal/day food only) as % of DRV
(35E%)' .
EXECUTE .
COMPUTE df22drvp = (df22 / 11*100) .
VARIABLE LABELS df22drvp '(D) Energy from SFA (kcal/day) as % of DRV (11E%)' .
EXECUTE .
COMPUTE df23drvp = (df23 / 13*100) .
```

```
VARIABLE LABELS df23drvp '(D) Energy from MUFA (kcal/day) as % of DRV (13E%)' .
EXECUTE .
COMPUTE df27drvp = (df27 / 6.5*100) .
VARIABLE LABELS df27drvp '(D) Energy from PUFA (kcal/day) as % of DRV (6.5E%)' .
EXECUTE .
COMPUTE df26drvp = (df26 / 2*100) .
VARIABLE LABELS df26drvp '(D) Energy from TRANS FA (kcal/day) as % of DRV (2E%)' .
EXECUTE .
```

Chapter 8: Vitamins

N.B. For percentage contribution syntax, see Chapter 5

SUPNTA28 (D) Retinol intake, all sources, sup taker
SUPNTA33 (D) Vitamin A intake, all sources, sup taker
SUPNTA35 (D) Thiamin intake, all sources, sup taker
SUPNTA36 (D) Riboflavin intake, all sources, sup taker
SUPNTA39 (D) Niacin equi intake, all sources, sup taker
SUPNTA42 (D) B6 intake, all sources, sup taker
SUPNTA43 (D) B12 intake, all sources, sup taker
SUPNTA44 (D) Folate intake, all sources, sup taker
SUPNTA45 (D) Pantothenic acid intake, all sources, sup taker
SUPNTA46 (D) Biotin intake, all sources, sup taker
SUPNTA40 (D) Vitamin C intake, all sources, sup taker
SUPNTA34 (D) Vitamin D intake, all sources, sup taker
SUPNTA41 (D) Vitamin E intake, all sources, sup taker

```
SPSS Syntax

GET
  FILE='C:\person level nutrient and food.sav'.

compute supp28=nta28-nfa28.
compute supp33=nta33-nfa33.
compute supp35=nta35-nfa35.
compute supp36=nta36-nfa36.
compute supp39=nta39-nfa39.
compute supp42=nta42-nfa42.
compute supp43=nta43-nfa43.
compute supp44=nta44-nfa44.
compute supp45=nta45-nfa45.
compute supp46=nta46-nfa46.
compute supp40=nta40-nfa40.
compute supp34=nta34-nfa34.
compute supp41=nta41-nfa41.

var labels
supp28 'Supplement intake of retinol'
supp33 'Supplement intake of vitamin A'
supp35 'Supplement intake of thiamin'
supp36 'Supplement intake of riboflavin'
supp39 'Supplement intake of niacin equivalents'
supp42 'Supplement intake of B6'
supp43 'Supplement intake of B12'
supp44 'Supplement intake of folate'
supp45 'Supplement intake of pantothenic acid'
supp46 'Supplement intake of biotin'
supp40 'Supplement intake of vitamin C'
supp34 'Supplement intake of vitamin D'
supp41 'Supplement intake of vitamin E'.


RECODE
  supp28 supp33 supp35 supp36 supp39 supp42 supp43 supp44 supp45 supp46
  supp40 supp34 supp41
  (0=Copy)  (MISSING=SYSMIS)  (ELSE=1)  INTO  cons28  cons33  cons35  cons36
  cons39  cons42  cons43  cons44  cons45  cons46  cons40  cons34  cons41 .
EXECUTE .

var labels
```

```
cons28 'Retinol supplement taker'
cons33 'Vitamin A supplement taker'
cons35 'Thiamin supplement taker'
cons36 'Riboflavin supplement taker'
cons39 'Niacin equiv supplement taker'
cons42 'B6 supplement taker'
cons43 'B12 supplement taker'
cons44 'Folate supplement taker'
cons45 'Pantothenic acid supplement taker'
cons46 'Biotin supplement taker'
cons40 'Vitamin C supplement taker'
cons34 'Vitamin D supplement taker'
cons41 'Vitamin E supplement taker'.

value label cons28 cons33 cons35 cons36 cons39 cons42 cons43 cons44 cons45 cons46
cons40 cons34 cons41
1 'Supplement taker' 0 'Not a supplement taker'.


IF (cons28 = 1) supnta28 = nta28 .
EXECUTE .
IF (cons33 = 1) supnta33 = nta33 .
EXECUTE .
IF (cons35 = 1) supnta35 = nta35 .
EXECUTE .
IF (cons36 = 1) supnta36 = nta36 .
EXECUTE .
IF (cons39 = 1) supnta39 = nta39 .
EXECUTE .
IF (cons42 = 1) supnta42 = nta42 .
EXECUTE .
IF (cons43 = 1) supnta43 = nta43 .
EXECUTE .
IF (cons44 = 1) supnta44 = nta44 .
EXECUTE .
IF (cons45 = 1) supnta45 = nta45 .
EXECUTE .
IF (cons46 = 1) supnta46 = nta46 .
EXECUTE .
IF (cons40 = 1) supnta40 = nta40 .
EXECUTE .
IF (cons34 = 1) supnta34 = nta34 .
EXECUTE .
IF (cons41 = 1) supnta41 = nta41 .
EXECUTE .

var labels
supnta28 '(D) Retinol intake, all sources, sup taker'
supnta33 '(D) Vitamin A intake, all sources, sup taker'
supnta35 '(D) Thiamin intake, all sources, sup taker'
supnta36 '(D) Riboflavin intake, all sources, sup taker'
supnta39 '(D) Niacin equi intake, all sources, sup taker'
supnta42 '(D) B6 intake, all sources, sup taker'
supnta43 '(D) B12 intake, all sources, sup taker'
supnta44 '(D) Folate intake, all sources, sup taker'
supnta45 '(D) Pantothenic acid intake, all sources, sup taker'
supnta46 '(D) Biotin intake, all sources, sup taker'
supnta40 '(D) Vitamin C intake, all sources, sup taker'
supnta34 '(D) Vitamin D intake, all sources, sup taker'
supnta41 '(D) Vitamin E intake, all sources, sup taker'.
```

SUPNFA28 (D) Retinol intake, food sources, sup taker
SUPNFA33 (D) Vitamin A intake, food sources, sup taker
SUPNFA35 (D) Thiamin intake, food sources, sup taker
SUPNFA36 (D) Riboflavin intake, food sources, sup taker
SUPNFA39 (D) Niacin equi intake, food sources, sup taker
SUPNFA42 (D) B6 intake, food sources, sup taker
SUPNFA43 (D) B12 intake, food sources, sup taker
SUPNFA44 (D) Folate intake, food sources, sup taker
SUPNFA45 (D) Pantothenic acid intake, food sources, sup taker
SUPNFA46 (D) Biotin intake, food sources, sup taker
SUPNFA40 (D) Vitamin C intake, food sources, sup taker
SUPNFA34 (D) Vitamin D intake, food sources, sup taker
SUPNFA41 (D) Vitamin E intake, food sources, sup taker

```
SPSS Syntax

GET
  FILE='C:\person level nutrient and food.sav'.

IF (cons28 = 1) supnfa28 = nfa28 .
EXECUTE .
IF (cons33 = 1) supnfa33 = nfa33 .
EXECUTE .
IF (cons35 = 1) supnfa35 = nfa35 .
EXECUTE .
IF (cons36 = 1) supnfa36 = nfa36 .
EXECUTE .
IF (cons39 = 1) supnfa39 = nfa39 .
EXECUTE .
IF (cons42 = 1) supnfa42 = nfa42 .
EXECUTE .
IF (cons43 = 1) supnfa43 = nfa43 .
EXECUTE .
IF (cons44 = 1) supnfa44 = nfa44 .
EXECUTE .
IF (cons45 = 1) supnfa45 = nfa45 .
EXECUTE .
IF (cons46 = 1) supnfa46 = nfa46 .
EXECUTE .
IF (cons40 = 1) supnfa40 = nfa40 .
EXECUTE .
IF (cons34 = 1) supnfa34 = nfa34 .
EXECUTE .
IF (cons41 = 1) supnfa41 = nfa41 .
EXECUTE .

var labels
supnfa28 '(D) Retinol intake, food sources, sup taker'
supnfa33 '(D) Vitamin A intake, food sources, sup taker'
supnfa35 '(D) Thiamin intake, food sources, sup taker'
supnfa36 '(D) Riboflavin intake, food sources, sup taker'
supnfa39 '(D) Niacin equi intake, food sources, sup taker'
supnfa42 '(D) B6 intake, food sources, sup taker'
supnfa43 '(D) B12 intake, food sources, sup taker'
supnfa44 '(D) Folate intake, food sources, sup taker'
supnfa45 '(D) Pantothenic acid intake, food sources, sup taker'
supnfa46 '(D) Biotin intake, food sources, sup taker'
supnfa40 '(D) Vitamin C intake, food sources, sup taker'
supnfa34 '(D) Vitamin D intake, food sources, sup taker'
supnfa41 '(D) Vitamin E intake, food sources, sup taker'.
```

NSNFA28 (D) Retinol intake, food sources, non sup taker
NSNFA33 (D) Vitamin A intake, food sources, non sup taker
NSNFA35 (D) Thiamin intake, food sources, non sup taker
NSNFA36 (D) Riboflavin intake, food sources, non sup taker
NSNFA39 (D) Niacin equi intake, food sources, non sup taker
NSNFA42 (D) B6 intake, food sources, non sup taker
NSNFA43 (D) B12 intake, food sources, non sup taker
NSNFA44 (D) Folate intake, food sources, non sup taker
NSNFA45 (D) Pantothenic acid intake, food sources, non sup taker
NSNFA46 (D) Biotin intake, food sources, non sup taker
NSNFA40 (D) Vitamin C intake, food sources, non sup taker
NSNFA34 (D) Vitamin D intake, food sources, non sup taker
NSNFA41 (D) Vitamin E intake, food sources, non sup taker

```
SPSS Syntax

GET
  FILE='C:\person level nutrient and food.sav'.

IF (cons28 = 0) NSnfa28 = nfa28 .
EXECUTE .
IF (cons33 = 0) NSnfa33 = nfa33 .
EXECUTE .
IF (cons35 = 0) NSnfa35 = nfa35 .
EXECUTE .
IF (cons36 = 0) NSnfa36 = nfa36 .
EXECUTE .
IF (cons39 = 0) NSnfa39 = nfa39 .
EXECUTE .
IF (cons42 = 0) NSnfa42 = nfa42 .
EXECUTE .
IF (cons43 = 0) NSnfa43 = nfa43 .
EXECUTE .
IF (cons44 = 0) NSnfa44 = nfa44 .
EXECUTE .
IF (cons45 = 0) NSnfa45 = nfa45 .
EXECUTE .
IF (cons46 = 0) NSnfa46 = nfa46 .
EXECUTE .
IF (cons40 = 0) NSnfa40 = nfa40 .
EXECUTE .
IF (cons34 = 0) NSnfa34 = nfa34 .
EXECUTE .
IF (cons41 = 0) NSnfa41 = nfa41 .
EXECUTE .

var labels
NSnfa28 '(D) Retinol intake, food sources, non sup taker'
NSnfa33 '(D) Vitamin A intake, food sources, non sup taker'
NSnfa35 '(D) Thiamin intake, food sources, non sup taker'
NSnfa36 '(D) Riboflavin intake, food sources, non sup taker'
NSnfa39 '(D) Niacin equi intake, food sources, non sup taker'
NSnfa42 '(D) B6 intake, food sources, non sup taker'
NSnfa43 '(D) B12 intake, food sources, non sup taker'
NSnfa44 '(D) Folate intake, food sources, non sup taker'
NSnfa45 '(D) Pantothenic acid intake, food sources, non sup taker'
NSnfa46 '(D) Biotin intake, food sources, non sup taker'
NSnfa40 '(D) Vitamin C intake, food sources, non sup taker'
NSnfa34 '(D) Vitamin D intake, food sources, non sup taker'
NSnfa41 '(D) Vitamin E intake, food sources, non sup taker'.
```

DFLRNI33 (D) <LRNI for vitamin A food only
DFLRNI35 (D) <LRNI for thiamin food only
DFLRNI36 (D) <LRNI for riboflavin food only
DFLRNI39 (D) <LRNI for niacin eq food only
DFLRNI42 (D) <LRNI for vitamin B6 food only
DFLRNI43 (D) <LRNI for vitamin B12 food only
DFLRNI44 (D) <LRNI for folate food only
DFLRNI40 (D) <LRNI for vitamin C food only

```
SPSS Syntax

GET
  FILE='C:\person level nutrient and food.sav'.

RECODE
  dfloa33 dfloa35 dfloa36 dfloa39 dfloa42 dfloa43 dfloa44 dfloa40
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dflrni33  dflrni35
  dflrni36  dflrni39  dflrni42  dflrni43  dflrni44  dflrni40 .
VARIABLE LABELS
dflrni33 '(D) <LRNI for vitamin A food only'
/dflrni35 '(D) <LRNI for thiamin food only'
/dflrni36 '(D) <LRNI for riboflavin food only'
/dflrni39 '(D) <LRNI for niacin eq food only'
/dflrni42 '(D) <LRNI for vitamin B6 food only'
/dflrni43 '(D) <LRNI for vitamin B12 food only'
/dflrni44 '(D) <LRNI for folate food only'
/dflrni40 '(D) <LRNI for vitamin C food only'.
EXECUTE.
```

DFRNIH45 (D) %>=7mg for pantothenic acid
DFRNIL45 (D) %<3mg for pantothenic acid
DFRNIH46 (D) %>=200ug for biotin
DFRNIL46 (D) %<10ug for biotin
DFRNI41 (D) %<SI for vitamin E

```
SPSS Syntax

GET
  FILE='C:\person level nutrient and food.sav'.

RECODE
  dfsiha45
  (SYSMIS=SYSMIS)  (100 thru Highest=1)  (ELSE=0)  INTO  dfrnih45 .
VARIABLE LABELS dfrnih45 '(D) %>=7mg for pantothenic acid'.
EXECUTE .

RECODE
  dfsila45
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dfrnil45 .
VARIABLE LABELS dfrnil45 '(D) %<3mg for pantothenic acid'.
EXECUTE .

RECODE
  dfsiha46
  (SYSMIS=SYSMIS)  (100 thru Highest=1)  (ELSE=0)  INTO  dfrnih46 .
VARIABLE LABELS dfrnih46 '(D) %>=200ug for biotin'.
EXECUTE .

RECODE
  dfsila46
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dfrnil46 .
VARIABLE LABELS dfrnil46 '(D) %<10ug for biotin'.
EXECUTE .
```

```
RECODE
  dfsia41
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dfrni41 .
VARIABLE LABELS dfrni41 '(D) %<SI for vitamin E'.
EXECUTE .
```

DFLRT33 (D) <LRNI for vitamin A inc supplements
DFLRT36 (D) <LRNI for riboflavin inc supplements
DFLRT44 (D) <LRNI for folate inc supplements

1 Less than 100%
0 More than 100%

```
GET
  FILE='C:\person level nutrient and food.sav’.

RECODE
  dtloa33 dtloa36 dtloa44
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dflrt33 dflrt36 dflrt44 .
VARIABLE LABELS dflrt33 '(D) <LRNI for vitamin A inc supplements'  /dflrt36 '(D) <LRNI
for riboflavin inc supplements' /dflrt44 '(D) <LRNI for folate inc supplements'.
EXECUTE.

value labels dflrt33 dflrt36 dflrt44
1'Less than 100%' 0'More than 100%'.
```

UNDER5 (D) aged 2, 3 or 4

1 under 5
2 over 5

```
GET
  FILE='C:\person level nutrient and food.sav’.

weight off.
split file off.

recode agep (lo thru 4 =1)(5 thru hi=2) into under5.
var label under5 '(D) aged 2, 3 or 4'.
value label under5 1 'under 5' 2 'over 5'.

EXECUTE.
```

Chapter 9: Minerals

NB To derive some variables in this chapter, previously derived variables are used from CHAPTER 4.
For percentage contribution syntax, see Chapter 5

SUPNTA52 (D) Iron intake, all sources, sup taker
SUPNTA49 (D) Calcium intake, all sources, sup taker
SUPNTA50 (D) Magnesium intake, all sources, sup taker
SUPNTA56 (D) Zinc intake, all sources, sup taker
SUPNTA55 (D) Copper intake, all sources, sup taker
SUPNTA58 (D) Iodine intake, all sources, sup taker
SUPNTA59 (D) Manganese intake, all sources, sup taker

```
GET
  FILE='C:\person level nutrient and food.sav'.

compute supp52=nta52-nfa52.
compute supp49=nta49-nfa49.
compute supp50=nta50-nfa50.
compute supp56=nta56-nfa56.
compute supp55=nta55-nfa55.
compute supp58=nta58-nfa58.
compute supp59=nta59-nfa59.

var labels
supp52 'Supplement intake of iron'
supp49 'Supplement intake of calcium'
supp50 'Supplement intake of magnesium'
supp56 'Supplement intake of zinc'
supp55 'Supplement intake of copper'
supp58 'Supplement intake of iodine'
supp59 'Supplement intake of manganese'.


RECODE
  supp52 supp49 supp50 supp56 supp55 supp58 supp59
  (0=Copy)  (MISSING=SYSMIS)  (ELSE=1)  INTO  cons52  cons49  cons50  cons56
  cons55  cons58  cons59 .
EXECUTE .

var labels
cons52 'Iron supplement taker'
cons49 'Calcium supplement taker'
cons50 'Magnesium supplement taker'
cons56 'Zinc supplement taker'
cons55 'Copper supplement taker'
cons58 'Iodine supplement taker'
cons59 'Manganese supplement taker'

value label cons52 cons49 cons50 cons56 cons55 cons58 cons59
1 'Supplement taker' 0 'Not a supplement taker'.

IF (cons52 = 1) supnta52 = nta52 .
EXECUTE .
IF (cons49 = 1) supnta49 = nta49 .
EXECUTE .
IF (cons50 = 1) supnta50 = nta50 .
EXECUTE .
IF (cons56 = 1) supnta56 = nta56 .
EXECUTE .
IF (cons55 = 1) supnta55 = nta55 .
EXECUTE .
IF (cons58 = 1) supnta58 = nta58 .
EXECUTE .
IF (cons59 = 1) supnta59 = nta59 .
EXECUTE .
```

```
var labels
supnta52 '(D) Iron intake, all sources, sup taker'
supnta49 '(D) Calcium intake, all sources, sup taker'
supnta50 '(D) Magnesium intake, all sources, sup taker'
supnta56 '(D) Zinc intake, all sources, sup taker'
supnta55 '(D) Copper intake, all sources, sup taker'
supnta58 '(D) Iodine intake, all sources, sup taker'
supnta59 '(D) Manganese intake, all sources, sup taker'.
```

SUPNFA52 (D) Iron intake, food only, sup taker
SUPNFA49 (D) Calcium intake, food only, sup taker
SUPNFA50 (D) Magnesium intake, food only, sup taker
SUPNFA56 (D) Zinc intake, food only, sup taker
SUPNFA55 (D) Copper intake, food only, sup taker
SUPNFA58 (D) Iodine intake, food only, sup taker
SUPNFA59 (D) Manganese intake, food only, sup taker

```
GET
  FILE='C:\person level nutrient and food.sav'.


IF (cons52 = 1) supnfa52 = nfa52 .
EXECUTE .
IF (cons49 = 1) supnfa49 = nfa49 .
EXECUTE .
IF (cons50 = 1) supnfa50 = nfa50 .
EXECUTE .
IF (cons56 = 1) supnfa56 = nfa56 .
EXECUTE .
IF (cons55 = 1) supnfa55 = nfa55 .
EXECUTE .
IF (cons58 = 1) supnfa58 = nfa58 .
EXECUTE .
IF (cons59 = 1) supnfa59 = nfa59 .
EXECUTE .

filter off.

var labels
supnfa52 '(D) Iron intake, food only, sup taker'
supnfa49 '(D) Calcium intake, food only, sup taker'
supnfa50 '(D) Magnesium intake, food only, sup taker'
supnfa56 '(D) Zinc intake, food only, sup taker'
supnfa55 '(D) Copper intake, food only, sup taker'
supnfa58 '(D) Iodine intake, food only, sup taker'
supnfa59 '(D) Manganese intake, food only, sup taker'.
```

NSNFA52 (D) Iron intake, non sup taker
NSNFA49 (D) Calcium intake, non sup taker
NSNFA50 (D) Magnesium intake, non sup taker
NSNFA 56 (D) Zinc intake, non sup taker
NSNFA55 (D) Copper intake, non sup taker
NSNFA58 (D) Iodine intake, non sup taker
NSNFA59 (D) Manganese intake, non sup taker

```
GET
  FILE='C:\person level nutrient and food.sav'.

IF (cons52 = 0) NSnfa52 = nfa52 .
EXECUTE .
IF (cons49 = 0) NSnfa49 = nfa49 .
EXECUTE .
```

```
IF (cons50 = 0) NSnfa50 = nfa50 .
EXECUTE .
IF (cons56 = 0) NSnfa56 = nfa56 .
EXECUTE .
IF (cons55 = 0) NSnfa55 = nfa55 .
EXECUTE .
IF (cons58 = 0) NSnfa58 = nfa58 .
EXECUTE .
IF (cons59 = 0) NSnfa59 = nfa59 .
EXECUTE .


var labels
NSnfa52 '(D) Iron intake, non sup taker'
NSnfa49 '(D) Calcium intake, non sup taker'
NSnfa50 '(D) Magnesium intake, non sup taker'
NSnfa56 '(D) Zinc intake, non sup taker'
NSnfa55 '(D) Copper intake, non sup taker'
NSnfa58 '(D) Iodine intake, non sup taker'
NSnfa59 '(D) Manganese intake, non sup taker'.
```

DFLRNI52 (D) <LRNI for total iron food only
DFLRNI49 (D) <LRNI for calcium food only
DFLRNI47 (D) <LRNI for sodium food only
DFLRNI48 (D) <LRNI for potassium food only
DFLRNI56 (D) <LRNI for zinc food only
DFLRNI58 (D) <LRNI for iodine food only
DFLRNI50 (D) <LRNI for magnesium food only
DFLRNI51 (D) <LRNI for phosphorus food only
DFLRNI57 (D) <LRNI for chloride food only

```
GET
  FILE='C:\person level nutrient and food.sav’.

RECODE
  dfloa52 dfloa49 dfloa47 dfloa48 dfloa56 dfloa58 dfloa50
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dflrni52  dflrni49
  dflrni47  dflrni48  dflrni56  dflrni58  dflrni50 .
VARIABLE LABELS dflrni52 '(D) <LRNI for total iron food only' /dflrni49 '(D) <LRNI for
calcium food only'
 /dflrni47 '(D) <LRNI for sodium food only' /dflrni48 '(D) <LRNI for potassium food
only' /dflrni56 '(D) <LRNI for'+
 ' zinc food only' /dflrni58 '(D) <LRNI for iodine food only' /dflrni50 '(D) <LRNI for
magnesium food only'.
EXECUTE .
RECODE
  dfloa51 dfloa57
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dflrni51  dflrni57 .
VARIABLE LABELS dflrni51 '(D) <LRNI for phosphorus food only' /dflrni57 '(D) <LRNI for
chloride food only'.
EXECUTE .
```

DFRNIL59 (D) %<SI manganese

```
GET
  FILE='C:\person level nutrient and food.sav’.

RECODE
  dfsia59
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dfrnil59 .
VARIABLE LABELS dfrnil59 '(D) %<SI manganese'.
EXECUTE .
```

IRONSUP (D) Takes iron containing supplement

```
 0 Does not take iron supplements
 1 Takes vits with iron only
 2 Takes iron only
 3 Takes vits with iron and iron only
```

```
GET
  FILE='C:\person level nutrient and food.sav'.

compute ironsup=0.
if (sup6tot=1 & sup7tot=0) ironsup=1.
if (sup6tot=0 & sup7tot=1) ironsup=2.
if (sup6tot=1 & sup7tot=1) ironsup=3.
var label ironsup '(D) Takes iron containing supplement'.
value label ironsup 0 'Does not take iron supplements ' 1 'Takes vits with iron only'
2 'Takes iron only' 3 'Takes vits with iron and iron only'.
execute.
```

DFLRT52 (D) <LRNI for total iron, inc supplements
DFLRT49 (D) <LRNI for calcium inc supplements
DFLRT48 (D) <LRNI for potassium inc supplements
DFLRT56 (D) <LRNI for zinc inc supplements
DFLRT58 (D) <LRNI for iodine inc supplements
DFLRT50 (D) <LRNI for magnesium inc supplements

```
 1 Less than 100%
 0 More than 100%
```

```
GET
  FILE='C:\person level nutrient and food.sav'.

RECODE
  dtloa52 dtloa49 dtloa48 dtloa56 dtloa58 dtloa50
  (SYSMIS=SYSMIS)  (Lowest thru 99.99=1)  (ELSE=0)  INTO  dflrt52  dflrt49
  dflrt48  dflrt56  dflrt58  dflrt50 .
VARIABLE LABELS dflrt52 '(D) <LRNI for total iron, inc supplements' /dflrt49 '(D)
<LRNI for calcium inc supplements'
 /dflrt48 '(D) <LRNI for potassium inc supplements' /dflrt56 '(D) <LRNI for'+
 ' zinc inc supplements' /dflrt58 '(D) <LRNI for iodine inc supplements' /dflrt50 '(D)
<LRNI for magnesium inc supplements'.
EXECUTE .

value labels dflrt48 dflrt49 dflrt50  dflrt52 dflrt56 dflrt58
1'Less than 100%' 0'More than 100%'.
```

Chapter 10: Comparisons with NDNS

AGEGP3 (D) NDNS age groups

```
 1 4 -18 years
 2 19 - 64 years
 3 65+ years
```

```
GET
  FILE='C:\person level nutrient and food.sav'.

recode agep
(1 thru 3=sysmiss)(4 thru 18=1)(19 thru 64=2)(65 thru hi=3) into agegp3.

var labels
agegp3 '(D) NDNS age groups'.

value labels agegp3
1 '4 -18 years'
2 '19 - 64 years'
3 '65+ years'.
execute.
```

FRTPORTN (D) Fruit and fruit juice portions, juice counts once, NDNS method

VEGBBPON (D) Total portions of veg, beans and pulses portion counts once only, NDNS method

FRVEGPON (D) Total fruit and veg portions, NDNS method

NB To create these derived variables it is necessary to run syntax for variables FRTPORT, VEGBBPO, FRVEGPO from Chapter 4.

```
GET
  FILE='C:\person level nutrient and food.sav'.

compute frcompN=wa4fg13*45/100.
execute.
var labels
frcompN 'fruit in fruit pies, grams, NDNS method'.

compute fruitN=wa4fg81+wa4fg82+wa4fg83+wa4fg84+wa4fg85+wa4fg86+ frcompN.
execute.
var labels
fruitN 'total fruit consumed inc comp dishes, excl fr juice, grams, NDNS method'.

compute frportN=fruitN/80.
execute.
var labels
frportN 'fruit portions excluding fruit juice, NDNS method'.

compute frjportN=wa4fg93.
execute.
compute frjportN=frjportN/80.
execute.
var labels
frjportN 'fruit juice portions, NDNS method'.

RECODE
  frjportN
  (Lowest thru 0.9999999=0)  (ELSE=1)  INTO  frpmax1N .
VARIABLE LABELS frpmax1N 'fruit juice portions maximum 1, NDNS method'.
EXECUTE .

compute frtportN=frportN+frpmax1N.
execute.
```

```
var labels
frtportN '(D) Fruit and fruit juice portions, juice counts once,NDNS method'.
compute vegdisN=wa4fg72*40/100.
execute.

var labels
vegdisN 'veg in vegetable dishes, NDNS method'.


compute vegtotN=vegdisN +
wa4fg63+wa4fg64+wa4fg65+wa4fg66+wa4fg67+wa4fg69+wa4fg70+wa4fg71+wa4fg75.
execute.
var labels
vegtotN 'veg consumption inc veg in dishes excluding beans and pulses, grams, NDNS
method'.

compute vegporN=vegtotN/80.
execute.
var labels
vegporN 'veg portions excluding beans and pulses, NDNS method'.

compute bbpulN=wa4fg68+wa4fg73.
execute.

compute bpulpoN=bbpulN/80.
var labels
bpulpoN 'portions of baked beans,beans and pulses consumed, NDNS method'.


RECODE
  bpulpoN
  (Lowest thru 0.99999999=0)  (ELSE=1)  INTO  bpulpo1N .
VARIABLE LABELS bpulpo1N 'bean and pulses portion, max one, NDNS method'.
EXECUTE .

compute vegbbpoN=vegporN+bpulpo1N.
execute.
var labels
vegbbpoN '(D) Total portions of veg, beans and pulses portion counts once only, NDNS
method'.

compute frvegpoN=vegbbpoN+frtportN.
execute.
var labels
frvegpoN '(D) Total fruit and veg portions, NDNS method'.
```

DTRNI34 (D) >=RNI for vitamin D all sources

1 More than 100%
0 Less than 100%

DFRNI34 (D) >=RNI for vitamin D food only

1 More than 100%
0 Less than 100%

```
GET
  FILE='C:\person level nutrient and food.sav’.


****DERIVED proportion of respondents aged 65 and over meeting the RNI for vit D*****

RECODE
  dta34 dfa34
  (SYSMIS=SYSMIS)  (100 thru Highest=1)  (ELSE=0)  INTO  dtrni34 dfrni34.
VARIABLE LABELS dtrni34 '(D) >=RNI for vitamin D all sources' /dfrni34 '(D) >=RNI for
vitamin D food only'.
EXECUTE .

value labels dtrni34 dfrni34
1'More than 100%' 0'Less than 100%'.
```

Chapter 11: Interpretation

HTOK (D) valid height
```
 1 valid
```
WTOK (D) valid weight
```
 1 valid
```
BMIOK (D) valid BMI
```
 1 valid
```
BMICUT (D) UK BMI national classification standards (85th/95th centile)
```
 1 normal-weight
 2 over-weight
 3 obese
```
BMICUT2 (D) UK BMI status (ovrght inc. obese)
```
 1 Neither overweight nor obese
 2 Overweight incl. obese
```
BMICUT3 (D) UK BMI status (non-obese vs obese)
```
 1 Non-obese
 2 Obese
```
BMIVG5 (D) Valid BMI (grouped:<18.5,18.5-25,25-30,30-40 40+)
```
 1 Under 18.5
 2 Over 18.5-25
 3 Over 25-30
 4 Over 30-40
 5 Over 40
```
TEEOK (D) valid TEE
```
 1 valid TEE
 2 no TEE
```

LG2 (D) Type of reporter, confidence interval-based method, 3 groups
```
 1 CI difference, low-energy reporter
 2 CI difference, good reporter
 3 CI difference, high-energy reporter
```

NB The derived variable ‘METAC’ is created in chapter 15.

```
GET
  FILE='C:\Individual.sav’.

COMPUTE apal = metac/24.
execute.

*adults.
if shserial =1156861 & serp=1 apal=-9.
if shserial =1086748 & serp=1 apal=-9.
if shserial =1030547 & serp=2 apal=-9.
if shserial =1093354 & serp=1 apal=-9.
if shserial =1053782 & serp=2 apal=-9.
if shserial =1155770 & serp=1 apal=-9.
if shserial =1125574 & serp=1 apal=-9.
if shserial =1148361 & serp=1 apal=-9.
if shserial =1115956 & serp=1 apal=-9.
if shserial =1117278 & serp=1 apal=-9.

*children.
if shserial =1122254 & serp=2 apal=-9.
if shserial =1091184 & serp=2 apal=-9.
if shserial =1131182 & serp=2 apal=-9.
if shserial =1057133 & serp=2 apal=-9.
if shserial =1085799 & serp=2 apal=-9.
variable labels apal '(I) aPAL - activity physical activity level'.
value labels apal -9 'Excluded as >168hours of activity recorded'.
```

```
MIS val apal (-9).


RECODE
  apal
  (MISSING=Copy)  (Lowest thru .9999999999999999999=-7)  (2.5 thru Highest=-8)
  (1 thru 1.399999999999999999=1)  (1.4 thru 1.599999999999999999=2)  (1.6
  thru 1.899999999999999999=3)  (1.9 thru 2.499999999999999999=4)  INTO  PAL.
VARIABLE LABELS PAL '(I) Physical activity level for equations'.
EXECUTE .
value labels pal -9 'Excluded as >168hours of activity recorded' -7 'aPAL value of <1'
-8 'aPAL value of >=2.5' 1'Sedentary' 2'Low active' 3 'Active' 4'Very active'.
MIS val pal (-9 -7 -8).

*convert height into metres from cm*.
compute heightm = (height/100).
execute.
variable label heightm '(I) Height in metres'.


**************                ADULTS         *****************.
*Normal-weight, Overweight, and Obese Men Ages 19 Years and Older**.


DO IF (agep >= 19 & sexp = 1) .
recode pal (1=1.00) (2=1.12) (3=1.27) (4=1.54) into PA.
VARIABLE LABELS PA '(I) Physical activity coefficient for equations'.
end if.

DO IF (agep >= 19 & sexp = 1) .
compute tee = 864 - (9.72*agep) + PA * (14.2 * weight + 503 * heightm).
VARIABLE LABELS tee '(I) TEE or EER (kcal)'.
end if.
execute.

*Normal-weight, Overweight, and Obese Women Ages 19 Years and Older*.

DO IF (agep >= 19 & sexp = 2) .
recode pal (1=1.00) (2=1.14) (3=1.27) (4=1.45) into PA.
end if.

DO IF (agep >= 19 & sexp = 2) .
compute tee = 387 - (7.31*agep) + PA * (10.9 * weight + 660.7 * heightm).
end if.
execute.


*create BMI, bmivg2,bmivg3,bmivg5 variables from ch12 syntax.
*run bmi centiles syntax from ch12 syntax /*need bmicut variable.

MIS VAL weight height (-1).

COMPUTE htok=2.
IF height>0 htok=1.
VAR LAB htok '(D) valid height'.
VAL LAB htok 1'valid' .


COMPUTE wtok=2.
IF weight>0 wtok=1.
VAR LAB wtok '(D) valid weight'.
VAL LAB  wtok 1'valid'.


compute heightm = (height/100).
execute.
variable label heightm '(I) Height in metres'.


COMPUTE bmiok=2.
IF bmi>0 bmiok=1.
VAR LAB bmiok '(D) valid BMI'.
VAL LAB  bmiok 1'valid'.


USE ALL.
```

```
COMPUTE filter_$=(agep >= 2 & agep < 19 & bmiok=1).
VARIABLE LABEL filter_$ 'agep >= 2 & agep < 19 (FILTER)'.
VALUE LABELS filter_$  0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .

compute bmicut=1.
IF sexp=1 AND RANGE(intexage, 2, 2.49) AND RANGE(bmi,18.12,19.10) bmicut=2.
IF sexp=2 AND RANGE(intexage, 2, 2.49) AND RANGE(bmi,17.83, 18.84) bmicut=2.
IF sexp=1 AND RANGE(intexage, 2.5, 2.99) AND RANGE(bmi,17.80,18.77) bmicut=2.
IF sexp=2 AND RANGE(intexage, 2.5, 2.99) AND RANGE(bmi,17.55,18.56) bmicut=2.


IF sexp=1 AND RANGE(intexage, 3, 3.49) AND RANGE(bmi,17.55,18.51) bmicut=2.
IF sexp=2 AND RANGE(intexage, 3, 3.49) AND RANGE(bmi,17.39,18.42) bmicut=2.
IF sexp=1 AND RANGE(intexage, 3.5, 3.99) AND RANGE(bmi,17.32,18.27 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 3.5, 3.99) AND RANGE(bmi,17.29,18.35) bmicut=2.


IF sexp=1 AND RANGE(intexage, 4, 4.49) AND RANGE(bmi,17.13,18.08) bmicut=2.
IF sexp=2 AND RANGE(intexage, 4, 4.49) AND RANGE(bmi,17.23,18.32) bmicut=2.
IF sexp=1 AND RANGE(intexage, 4.5, 4.99) AND RANGE(bmi,17.01,17.97) bmicut=2.
IF sexp=2 AND RANGE(intexage, 4.5, 4.99) AND RANGE(bmi,17.17,18.31) bmicut=2.


IF sexp=1 AND RANGE(intexage, 5, 5.49) AND RANGE(bmi,16.96,17.95 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 5, 5.49) AND RANGE(bmi,17.16 ,18.35) bmicut=2.
IF sexp=1 AND RANGE(intexage, 5.5, 5.99) AND RANGE(bmi,16.96,17.99) bmicut=2.
IF sexp=2 AND RANGE(intexage, 5.5, 5.99) AND RANGE(bmi,17.21 ,18.46) bmicut=2.


IF sexp=1 AND RANGE(intexage, 6, 6.49) AND RANGE(bmi,17.01 ,18.10) bmicut=2.
IF sexp=2 AND RANGE(intexage, 6, 6.49) AND RANGE(bmi,17.32 ,18.65) bmicut=2.
IF sexp=1 AND RANGE(intexage, 6.5, 6.99) AND RANGE(bmi,17.10, 18.26) bmicut=2.
IF sexp=2 AND RANGE(intexage, 6.5, 6.99) AND RANGE(bmi,17.49 ,18.91) bmicut=2.


IF sexp=1 AND RANGE(intexage, 7, 7.49) AND RANGE(bmi,17.24 ,18.48) bmicut=2.
IF sexp=2 AND RANGE(intexage, 7, 7.49) AND RANGE(bmi,17.71 ,19.22) bmicut=2.
IF sexp=1 AND RANGE(intexage, 7.5, 7.99) AND RANGE(bmi,17.41 ,18.74) bmicut=2.
IF sexp=2 AND RANGE(intexage, 7.5, 7.99) AND RANGE(bmi,17.96 ,19.56) bmicut=2.


IF sexp=1 AND RANGE(intexage, 8, 8.49) AND RANGE(bmi,17.61 ,19.04 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 8, 8.49) AND RANGE(bmi,18.23 ,19.93) bmicut=2.
IF sexp=1 AND RANGE(intexage, 8.5, 8.99) AND RANGE(bmi,17.83 ,19.36) bmicut=2.
IF sexp=2 AND RANGE(intexage, 8.5, 8.99) AND RANGE(bmi, 18.52,20.30 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 9, 9.49) AND RANGE(bmi,18.08 ,19.70 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 9, 9.49) AND RANGE(bmi,18.82 ,20.70) bmicut=2.
IF sexp=1 AND RANGE(intexage, 9.5, 9.99) AND RANGE(bmi, 18.35, 20.05) bmicut=2.
IF sexp=2 AND RANGE(intexage, 9.5, 9.99) AND RANGE(bmi,19.15 ,21.10) bmicut=2.


IF sexp=1 AND RANGE(intexage, 10,10.49) AND RANGE(bmi,18.64 ,20.42 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 10,10.49) AND RANGE(bmi,19.49 ,21.52) bmicut=2.
IF sexp=1 AND RANGE(intexage, 10.5,10.99) AND RANGE(bmi,18.94 ,20.79) bmicut=2.
IF sexp=2 AND RANGE(intexage, 10.5,10.99) AND RANGE(bmi,19.85, 21.94) bmicut=2.


IF sexp=1 AND RANGE(intexage, 11,11.49) AND RANGE(bmi,19.26 ,21.18 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 11,11.49) AND RANGE(bmi,20.22 ,22.36) bmicut=2.
IF sexp=1 AND RANGE(intexage, 11.5,11.99) AND RANGE(bmi,19.59 ,21.57) bmicut=2.
IF sexp=2 AND RANGE(intexage, 11.5,11.99) AND RANGE(bmi,20.60,22.80 ) bmicut=2.

IF sexp=1 AND RANGE(intexage, 12,12.49) AND RANGE(bmi,19.93,21.96 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 12,12.49) AND RANGE(bmi,20.98,23.22) bmicut=2.
IF sexp=1 AND RANGE(intexage, 12.5,12.99) AND RANGE(bmi,20.29,22.36) bmicut=2.
IF sexp=2 AND RANGE(intexage, 12.5,12.99) AND RANGE(bmi,21.37,23.65 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 13,13.49) AND RANGE(bmi,20.65,22.77 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 13,13.49) AND RANGE(bmi,21.74 ,24.06) bmicut=2.
IF sexp=1 AND RANGE(intexage, 13.5,13.99) AND RANGE(bmi,21.02 ,23.17) bmicut=2.
```

```
IF sexp=2 AND RANGE(intexage, 13.5,13.99) AND RANGE(bmi,22.10,24.45 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 14,14.49) AND RANGE(bmi,21.39,23.58) bmicut=2.
IF sexp=2 AND RANGE(intexage, 14,14.49) AND RANGE(bmi,22.45,24.82) bmicut=2.
IF sexp=1 AND RANGE(intexage, 14.5,14.99) AND RANGE(bmi,21.76 ,23.97) bmicut=2.
IF sexp=2 AND RANGE(intexage, 14.5,14.99) AND RANGE(bmi,22.77, 25.16) bmicut=2.


IF sexp=1 AND RANGE(intexage, 15,15.49) AND RANGE(bmi,22.12,24.36) bmicut=2.
IF sexp=2 AND RANGE(intexage, 15,15.49) AND RANGE(bmi,23.08,25.49) bmicut=2.
IF sexp=1 AND RANGE(intexage, 15.5,15.99) AND RANGE(bmi,22.48,24.74) bmicut=2.
IF sexp=2 AND RANGE(intexage, 15.5,15.99) AND RANGE(bmi,23.35,25.78 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 16,16.49) AND RANGE(bmi,22.82,25.09) bmicut=2.
IF sexp=2 AND RANGE(intexage, 16,16.49) AND RANGE(bmi,23.61,26.05) bmicut=2.
IF sexp=1 AND RANGE(intexage, 16.5,16.99) AND RANGE(bmi,23.15,25.44) bmicut=2.
IF sexp=2 AND RANGE(intexage, 16.5,16.99) AND RANGE(bmi,23.84,26.29 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 17,17.49) AND RANGE(bmi,23.46,25.77) bmicut=2.
IF sexp=2 AND RANGE(intexage, 17,17.49) AND RANGE(bmi,24.06,26.52) bmicut=2.
IF sexp=1 AND RANGE(intexage, 17.5,17.99) AND RANGE(bmi,23.76,26.08) bmicut=2.
IF sexp=2 AND RANGE(intexage, 17.5,17.99) AND RANGE(bmi,24.25,26.72 ) bmicut=2.

IF sexp=1 AND RANGE(intexage, 18,18.49) AND RANGE(bmi,24.05,26.37) bmicut=2.
IF sexp=2 AND RANGE(intexage, 18,18.49) AND RANGE(bmi,24.43,26.91) bmicut=2.
IF sexp=1 AND RANGE(intexage, 18.5,18.99) AND RANGE(bmi,24.32,26.65) bmicut=2.
IF sexp=2 AND RANGE(intexage, 18.5,18.99) AND RANGE(bmi,24.60,27.08 ) bmicut=2.
EXECUTE.


IF sexp=1 AND RANGE(intexage, 2, 2.49) AND (bmi ge 19.1001) bmicut=3.
IF sexp=2 AND RANGE(intexage, 2, 2.49) AND (bmi ge 18.8401) bmicut=3.
IF sexp=1 AND RANGE(intexage, 2.5, 2.99) AND (bmi ge 18.7701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 2.5, 2.99) AND (bmi ge 18.5601) bmicut=3.

IF sexp=1 AND RANGE(intexage, 3, 3.49) AND (bmi ge 18.5101) bmicut=3.
IF sexp=2 AND RANGE(intexage, 3, 3.49) AND (bmi ge 18.4201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 3.5, 3.99) AND (bmi ge 18.2701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 3.5, 3.99) AND (bmi ge 18.3501 ) bmicut=3.

IF sexp=1 AND RANGE(intexage, 4, 4.49) AND (bmi ge 18.0801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 4, 4.49) AND (bmi ge 18.3201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 4.5, 4.99) AND (bmi ge 17.9701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 4.5, 4.99) AND (bmi ge 18.3101) bmicut=3.

IF sexp=1 AND RANGE(intexage, 5, 5.49) AND (bmi ge 17.9501) bmicut=3.
IF sexp=2 AND RANGE(intexage, 5, 5.49) AND (bmi ge 18.3501) bmicut=3.
IF sexp=1 AND RANGE(intexage, 5.5, 5.99) AND (bmi ge 17.9901) bmicut=3.
IF sexp=2 AND RANGE(intexage, 5.5, 5.99) AND (bmi ge 18.4601) bmicut=3.

IF sexp=1 AND RANGE(intexage, 6, 6.49) AND (bmi ge 18.1001) bmicut=3.
IF sexp=2 AND RANGE(intexage, 6, 6.49) AND (bmi ge 18.6501) bmicut=3.
IF sexp=1 AND RANGE(intexage, 6.5, 6.99) AND (bmi ge 18.2601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 6.5, 6.99) AND (bmi ge 18.9101) bmicut=3.

IF sexp=1 AND RANGE(intexage, 7, 7.49) AND (bmi ge 18.4801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 7, 7.49) AND (bmi ge 19.2201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 7.5, 7.99) AND (bmi ge 18.7401) bmicut=3.
IF sexp=2 AND RANGE(intexage, 7.5, 7.99) AND (bmi ge 19.5601) bmicut=3.

IF sexp=1 AND RANGE(intexage, 8, 8.49) AND (bmi ge 19.0401) bmicut=3.
IF sexp=2 AND RANGE(intexage, 8, 8.49) AND (bmi ge 19.9301) bmicut=3.
IF sexp=1 AND RANGE(intexage, 8.5, 8.99) AND (bmi ge 19.3601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 8.5, 8.99) AND (bmi ge 20.3001) bmicut=3.

IF sexp=1 AND RANGE(intexage, 9, 9.49) AND (bmi ge 19.7001) bmicut=3.
IF sexp=2 AND RANGE(intexage, 9, 9.49) AND (bmi ge 20.7001) bmicut=3.
IF sexp=1 AND RANGE(intexage, 9.5, 9.99) AND (bmi ge 20.0501) bmicut=3.
IF sexp=2 AND RANGE(intexage, 9.5, 9.99) AND (bmi ge 21.1001) bmicut=3.

IF sexp=1 AND RANGE(intexage, 10,10.49) AND (bmi ge 20.4201) bmicut=3.
IF sexp=2 AND RANGE(intexage, 10,10.49) AND (bmi ge 21.5201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 10.5,10.99) AND (bmi ge 20.7901) bmicut=3.
```

```
IF sexp=2 AND RANGE(intexage, 10.5,10.99) AND (bmi ge 21.9401) bmicut=3.

IF sexp=1 AND RANGE(intexage, 11,11.49) AND (bmi ge 20.1801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 11,11.49) AND (bmi ge 22.3601) bmicut=3.
IF sexp=1 AND RANGE(intexage, 11.5,11.99) AND (bmi ge 21.5701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 11.5,11.99) AND (bmi ge 22.8001) bmicut=3.

IF sexp=1 AND RANGE(intexage, 12,12.49) AND (bmi ge 21.9601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 12,12.49) AND (bmi ge 23.2201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 12.5,12.99) AND (bmi ge 22.3601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 12.5,12.99) AND (bmi ge 23.6501) bmicut=3.

IF sexp=1 AND RANGE(intexage, 13,13.49) AND (bmi ge 22.7701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 13,13.49) AND (bmi ge 24.0601) bmicut=3.
IF sexp=1 AND RANGE(intexage, 13.5,13.99) AND (bmi ge 23.1701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 13.5,13.99) AND (bmi ge 24.4501) bmicut=3.

IF sexp=1 AND RANGE(intexage, 14,14.49) AND (bmi ge 23.5801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 14,14.49) AND (bmi ge 24.8201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 14.5,14.99) AND (bmi ge 23.9701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 14.5,14.99) AND (bmi ge 25.1601) bmicut=3.

IF sexp=1 AND RANGE(intexage, 15,15.49) AND (bmi ge 24.3601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 15,15.49) AND (bmi ge 25.4901) bmicut=3.
IF sexp=1 AND RANGE(intexage, 15.5,15.99) AND (bmi ge 24.7401) bmicut=3.
IF sexp=2 AND RANGE(intexage, 15.5,15.99) AND (bmi ge 25.7801) bmicut=3.


IF sexp=1 AND RANGE(intexage, 16,16.49) AND (bmi ge 25.0901) bmicut=3.
IF sexp=2 AND RANGE(intexage, 16,16.49) AND (bmi ge 26.0501) bmicut=3.
IF sexp=1 AND RANGE(intexage, 16.5,16.99) AND (bmi ge 25.4401) bmicut=3.
IF sexp=2 AND RANGE(intexage, 16.5,16.99) AND (bmi ge 26.2901) bmicut=3.

IF sexp=1 AND RANGE(intexage, 17,17.49) AND (bmi ge 25.7701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 17,17.49) AND (bmi ge 26.5201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 17.5,17.99) AND (bmi ge 26.0801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 17.5,17.99) AND (bmi ge 26.7201) bmicut=3.

IF sexp=1 AND RANGE(intexage, 18,18.49) AND (bmi ge 26.3701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 18,18.49) AND (bmi ge 26.9101) bmicut=3.
IF sexp=1 AND RANGE(intexage, 18.5,18.99) AND (bmi ge 26.6501) bmicut=3.
IF sexp=2 AND RANGE(intexage, 18.5,18.99) AND (bmi ge 27.0801) bmicut=3.
EXECUTE.

VAR LAB bmicut '(D) UK BMI national classification standards (85th/95th centile)'.
value labels bmicut
1 'normal-weight'
2 'over-weight'
3 'obese'.


RECODE bmicut (1=1) (2 thru 3=2) INTO bmicut2.
VAR LAB bmicut2 '(D) UK BMI status (ovrght inc. obese)'.
VAL LAB bmicut2 1 'Neither overweight nor obese'  2 'Overweight incl. obese'.

RECODE bmicut (1 thru 2=1) (3=2) INTO bmicut3.
VAR LAB bmicut3 '(D) UK BMI status (non-obese vs obese)'.
VAL LAB bmicut3 1 'Non-obese'  2 'Obese'.

filter off.
USE ALL.


******* compute BMI 5 groups*********.

DO IF (agep >= 19 & bmiok=1) .
RECODE bmi (0 thru 18.5=1)(18.5 thru 25=2)(25 thru 30=3) (30 thru 40=4)
  (40 thru hi=5) (lo thru -1=COPY) INTO bmivg5.
end if.

VARIABLE LABELS bmivg5 "(D) Valid BMI (grouped:<18.5,18.5-25,25-30,30-40 40+)".
VALUE LABELS bmivg5
  1 "Under 18.5"
  2 "Over 18.5-25"
  3 "Over 25-30"
  4 "Over 30-40"
```

```
  5 "Over 40".

******* compute BMI 3 groups*********.

DO IF (agep >= 19 & bmiok=1) .
RECODE bmivg5 (1 thru 2=1) (3=2) (4 thru hi=3) into bmivg3.
VAR LAB bmivg3 'Normal, overweight, obese'.
VAL LAB bmivg3 1'normal weight' 2'overweight' 3'obese'.
end if.


******* compute BMI 2 groups*********.

DO IF (agep >= 19 & bmiok=1) .
RECODE bmivg5 (1 thru 2=1) (3 thru hi=2) into bmivg2.
VAR LAB bmivg2 'Overweight + obese'.
VAL LAB bmivg2 1 'normal weight' 2'overweight + obese'.
end if.


* Exclude underweight*.
do if bmivg5=1 & agep >= 19 .
recode tee (else= sysmis).
end if.


**************                CHILDREN              *****************.
**** ALL CHILDREN <3YEARS   ******.
DO IF (agep <3 ) .
compute tee = (89*weight - 100) +20 .
end if.
execute.


***NORMAL WEIGHT      3-<19 years***.

*** EER for Boys 3 Through 8 years ****.
DO IF ((agep >=3) & (agep <9) & (sexp = 1) & (bmicut =1)) .
recode pal (1=1.00) (2=1.13) (3=1.26) (4=1.42) into pa.
end if.

DO IF ((agep >=3) & (agep <9) & (sexp = 1) & (bmicut =1)) .
compute tee = 88.5 - (61.9*agep) + pa * (26.7 * weight + 903 * heightm) +20.
end if.
execute.

***  EER for Girls 3 Through 8 Years****.
DO IF ((agep >=3) & (agep <9) & (sexp = 2) & (bmicut =1)) .
recode pal (1=1.00) (2=1.16) (3=1.31) (4=1.56) into pa.
end if.
DO IF ((agep >=3) & (agep <9) & (sexp = 2) & (bmicut =1)) .
compute tee = 135.3 - (30.8*agep) + pa * (10.0 * weight + 934 * heightm) + 20.
end if.
execute.

*Ages 9 Through 18 Years*.
*EER for Boys 9 Through 18 Years*.
DO IF ((agep >=9) & (agep <19) & (sexp = 1) & (bmicut =1)) .
recode pal (1=1.00) (2=1.13) (3=1.26) (4=1.42) into pa.
end if.
DO IF ((agep >=9) & (agep <19) & (sexp = 1) & (bmicut =1)) .
compute tee = 88.5 - (61.9*agep) + pa * (26.7 * weight + 903 * heightm) +25.
end if.
execute.

*EER for Girls 9 Through 18 Years*.
DO IF ((agep >=9) & (agep <19) & (sexp = 2) & (bmicut =1)) .
recode pal (1=1.00) (2=1.16) (3=1.31) (4=1.56) into pa.
end if.
DO IF ((agep >=9) & (agep <19) & (sexp = 2) & (bmicut =1)) .
compute tee = 135.3 - (30.8*agep) + pa * (10.0 * weight + 934 * heightm) + 25.
end if.
execute.


***OVERWEIGHT  3-<19 years***.
```

```
*Weight Maintenance TEE in Overweight Boys Ages 3 Through 18 Years*.
DO IF ((agep >=3) & (agep <19) & (sexp = 1) & (bmicut >1)) .
recode pal (1=1.00) (2=1.12) (3=1.24) (4=1.45) into pa.
end if.
DO IF ((agep >=3) & (agep <19) & (sexp = 1) & (bmicut >1)) .
compute tee = 114 - (50.9*agep) + pa * (19.5 * weight + 1161.4 * heightm).
end if.
execute.

*Weight Maintenance TEE in Overweight Girls Ages 3 Through 18 Years.
DO IF ((agep >=3) & (agep <19) & (sexp = 2) & (bmicut >1)) .
recode pal (1=1.00) (2=1.18) (3=1.35) (4=1.60) into pa.
end if.
DO IF ((agep >=3) & (agep <19) & (sexp = 2) & (bmicut >1)) .
compute tee = 389 - (41.2*agep) + pa * (15.0 * weight + 701.6 * heightm).
end if.
execute.



sort cases by shserial serp.
SAVE OUTFILE='C:\Individual.sav'
 /COMPRESSED.

GET
  FILE='C:\person level nutrient and food.sav'.

sort cases by shserial serp.

SAVE OUTFILE='C:\person level nutrient and food.sav'
 /COMPRESSED.



MATCH FILES /FILE=*
 /TABLE='C:\Individual.sav'
 /RENAME (ageg agegr2 AgeP area country cu_wt ethgr4 hhcomp indiv_wt nurse_wt region
SexP strata urban = d0 d1 d2 d3 d4 d5 d6
  d7 d8 d9 d10 d11 d12 d13)
 /BY shserial serp
 /DROP= d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13.
EXECUTE.

SAVE OUTFILE='C:\person level nutrient and food DATE TODAY.sav'
 /COMPRESSED.


******         COMPUTING THE CVt             *************.

GET
  FILE='C:\day level data.sav'.

AGGREGATE
  /OUTFILE='C:\Aggr mean and sd of'+
 ' EI.sav'
  /BREAK=shserial serp
  /nt7_1 = MEAN(nt7)
  /nt7_2 = SD(nt7)
  /N_BREAK=N.

GET
  FILE='C:\Aggr mean and sd of EI.sav'.

rename variables  (nt7_1=nt7mean) (nt7_2=nt7sd) .
VAR LAB nt7mean '(I) Mean energy intake for calculations (kcal)'.
VAR LAB nt7sd '(I) Standard deviation of mean energy intake (kcal)'.

compute wscvei7 = (nt7sd/nt7mean)*100.
VAR LAB wscvei7 '(I) Within subject coefficient of variation for energy intake
(kcal)'.
execute.

sort cases by shserial serp.

SAVE OUTFILE='C:\Aggr mean and sd of EI.sav'
```

```
 /COMPRESSED.


GET
  FILE='C:\person level nutrient and food DATE TODAY.sav'.

sort cases by shserial serp.


SAVE OUTFILE='C:\person level nutrient and food DATE TODAY.sav'
 /COMPRESSED.

MATCH FILES /FILE=*
 /TABLE='C:\Aggr mean and sd of EI.sav'
 /BY shserial serp.
EXECUTE.

compute cvei7sq = wscvei7**2.
VAR LAB cvei7sq '(I) Squared - Within subject coefficient of variation for energy
intake (kcal)'.
execute.


*MEN*.
do if agegp6=1 & sexp=1.
compute cvt = 14.99.
end if.
do if agegp6=2 & sexp=1.
compute cvt = 17.59.
end if.
do if agegp6=3 & sexp=1.
compute cvt = 18.49.
end if.
do if agegp6=4 & sexp=1.
compute cvt = 17.88.
end if.
do if agegp6=5 & sexp=1.
compute cvt = 15.59.
end if.
do if agegp6=6 & sexp=1.
compute cvt = 14.42.
end if.

*WOMEN*.
do if agegp6=1 & sexp=2.
compute cvt = 15.26.
end if.
do if agegp6=2 & sexp=2.
compute cvt = 17.91.
end if.
do if agegp6=3 & sexp=2.
compute cvt = 17.72.
end if.
do if agegp6=4 & sexp=2.
compute cvt = 17.53.
end if.
do if agegp6=5 & sexp=2.
compute cvt = 15.64.
end if.
do if agegp6=6 & sexp=2.
compute cvt = 13.94.
end if.
execute.

VAR LAB cvt '(I) Coefficient of variation for both EE and EI'.

**      COMPUTE VALID TEE             **.

compute teeok = 2.
IF tee>0 teeok=1.
VAR LAB teeok '(D) valid TEE'.
VAL LAB  teeok 1'valid TEE'   2'no TEE'.

************************************************.

* find classification based on log % difference.
```

```
do if teeok = 1.
compute teeilpd=100*(ln(tee)-ln(nta7)).
end if.
var label
teeilpd '(I) Log percent difference of TEE and EI'.
execute.

* classify resp based on computation of individual confidence interval around tee and
ei.
* compute negative (lower) value for cvt.
do if teeok = 1.
compute cvtlow2=cvt*(-3.182/2).
compute cvthi2=cvt*3.182/2.
end if.

var label
cvtlow2 '(I) Lower limit of confidence interval-based difference between EE and EI'
cvthi2 '(I) Upper limit of confidence interval-based difference between EE and EI'.


if (teeilpd < cvtlow2) lg2=3.
if (teeilpd >= cvtlow2 and teeilpd <= cvthi2 ) lg2=2.
if (teeilpd > cvthi2) lg2=1.

var label
lg2 '(D) Type of reporter, confidence interval-based method, 3 groups'.
value label
lg2 1 'CI difference, low-energy reporter' 2 'CI difference, good reporter' 3 'CI
difference, high-energy reporter'.

SAVE OUTFILE='C:\person level nutrient and food DATE TODAY.sav'
 /COMPRESSED.
```

EATBALGP (D) Overall intake level - grouped

-1 Ate less than usual on 2 or more days
1 Ate more than usual on 2 or more days
0 Ate usual amount or less/more on 1day

```
***    Eating more or less than usual       *****.
GET
  FILE='C:\day level data.sav'.

* identify those whose eating is affected because ate more than usual or less than
usual for any reason (including being unwell) on 2 or more days.
if (foodam=1) eatusual=1.
if (foodam=2) eatless=1.
if (foodam=3) eatmore=1.

SAVE OUTFILE='C:\day level data.sav'
 /COMPRESSED.

* create aggregate file with sum of days per person according to eating usual, more or
less than usual.
AGGREGATE
  /OUTFILE='C:\Aggr eat usual, less or more.sav'
  /BREAK=shserial serp
  /eatusual = SUM(eatusual) /eatless = SUM(eatless) /eatmore =
  SUM(eatmore).


GET
  FILE='C:\Aggr eat usual, less or more.sav'.

SORT CASES BY
  shserial (A) serp (A) .

* balance days eaten more or less and compute score.

compute eatl= eatless*(-1).
execute.

variable label eatusual '(I) Number of days of eating usual'.
variable label eatless '(I) Number of days of eating less than usual'.
```

```
variable label eatmore '(I) Number of days of eating more than usual'.
variable label eatl '(I) Number of days of eating less than usual - negative value'.

compute eatbal=sum(eatmore, eatl).
variable label eatbal '(I) Overall intake level'.

recode eatbal (-4, -3,-2 =-1) (4,3,2=1) (-1,1,0=0) (else=copy) into eatbalgp.
execute.
value label eatbalgp -1 'Ate less than usual on 2 or more days' 1 'Ate more than usual
on 2 or more days' 0 'Ate usual amount or less/more on 1day'.
variable label eatbalgp '(D) Overall intake level - grouped'.

SORT CASES BY
  shserial (A) serp (A) .

SAVE OUTFILE='C:\Aggr eat usual, less or more.sav'
 /COMPRESSED.

GET
  FILE='C:\person level nutrient and food DATE TODAY.sav'.

SORT CASES BY
  shserial (A) serp (A) .

MATCH FILES /FILE=*
 /TABLE='C:\Aggr eat usual, less or more.sav'
 /BY shserial serp.
EXECUTE.
```

DIETER (D) Weight reducing diet

1 Dieter
2 Non dieter or missing

```
***     Dieting -weight reducing              *****.

* identify dieters .

GET
  FILE='C:\Individual.sav'.

recode diett1 (2=1) into dieter.
recode diett2 (2=1) into dieter.
recode dieter (1=1) (else=2).
execute.
value label dieter 1 'Dieter'  2'Non dieter or missing' .
variable label dieter '(D) Weight reducing diet'.

SAVE OUTFILE='C:\Aggr dieters.sav'.

*Kept only dieters and sherial and serp variables*.

SORT CASES BY
  shserial (A) serp (A) .

SAVE OUTFILE='C:\Aggr dieters.sav'
 /COMPRESSED.

GET
  FILE='C:\person level nutrient and food DATE TODAY.sav.'

SORT CASES BY
  shserial (A) serp (A) .


MATCH FILES /FILE=*
 /TABLE='C:\Aggr dieters.sav'
 /BY shserial serp.
EXECUTE.

SAVE OUTFILE='C:\ person level nutrient and food DATE TODAY.sav'
 /COMPRESSED.
```

TEEMJ (D) TEE or EER (mj)
EIVSTEE (D) Reported energy intake - (minus) TEE or EER (kcal)
EIVSTEE2 (D) Reported energy intake - (minus) TEE or EER (MJ)
TEEDIFP (D) Difference as a percent of energy needs

```
GET FILE='C:\person level nutrient and food DATE TODAY.sav'

**      COMPUTE TEE IN MJ             **.

do if teeok =1.
compute teemj = (tee * 4.184)/1000.
end if.
execute.
VAR LAB teemj '(D) TEE or EER (mj)'.

**      COMPUTE EI - (MINUS) TEE IN KCAL & MJ  **.

*compute reported energy intake (EI) - (minus) TEE/EER in mj/day.

do if teeok =1.
compute eivstee = nta7 - tee .
end if.
execute.
VAR LAB eivstee '(D) Reported energy intake - (minus) TEE or EER (kcal)'.



do if teeok =1.
compute eivstee2 = (eivstee*4.184)/1000.
end if.
execute.
VAR LAB eivstee2 '(D) Reported energy intake - (minus) TEE or EER (MJ)'.

**      COMPUTE EI - (MINUS) TEE AS A % OF ENERGY NEEDS**.

*Compute difference as a % of energy needs.

do if teeok =1.
compute teedifp = (eivstee2/teemj)*100.
end if.
execute.
VAR LAB teedifp '(D) Difference as a percent of energy needs'.
```

EATBALOK (D) valid eatbalgp
1 valid eatbalgp
2 no eatbalgp

REPORTER (D) Type of reporter - 4 groups, no missing eatbalgp
1 Low energy reporter
2 Low consumer
3 Good reporter
4 High energy reporter
-8 Excluded - dieting, no valid eatbalgp or no valid tee

```
*NEW FINAL REPORTER VARIABLE*.

* separate the lg2 variable into 'low energy reporters, low consumers & high energy
reporters, high consumers.
* Exclude dieters from the variable.
* only 7 high consumers so leave them in the 'high energy reporter' group / not worth
separating.
* Exclude if no valid eatbalgp.
*1624 both usual questions and type of reporter valid.
* remove those with no reponses/ incomplete responses to usual / less than usual from
type of reporter.
```

```
GET FILE='C:\ person level nutrient and food DATE TODAY.sav'

* COMPUTE VALID EATBALGP VARIABLE *.
compute eatbalok = 2.
IF eatbalgp = -1 eatbalok=1.
IF eatbalgp = 0 eatbalok=1.
IF eatbalgp = 1 eatbalok=1.
VAR LAB eatbalok '(D) valid eatbalgp'.
VAL LAB  eatbalok 1'valid eatbalgp'   2'no eatbalgp'.

RECODE
  lg2 (1=1) (2=3) (3=4) (else=-8)
  INTO  Reporter .
EXECUTE .

DO IF (eatbalgp =  - 1 & lg2 = 1) .
RECODE
  reporter  (1=2).
END IF .
EXECUTE .

DO IF (eatbalok = 2) .
RECODE
  reporter  (ELSE=-8)  .
END IF .
EXECUTE .

DO IF (dieter = 1) .
RECODE
  reporter  (ELSE=-8)  .
END IF .
EXECUTE .

Mis val reporter (-8).

Value labels Reporter 1'Low energy reporter' 2'Low consumer' 3'Good reporter' 4'High
energy reporter' -8 'Excluded - dieting, no valid eatbalgp or no valid tee' .
VAR LAB reporter '(D) Type of reporter - 4 groups, no missing eatbalgp'.
EXECUTE .
```

WTA6FG31 (D) Supplements

```
*Compare foods and nuts intake by type of reporter*.

**MERGE IN THE 30 FOOD GROUPS CREATED IN CH 4.

GET FILE='C:\person level nutrient and food.sav'.

sort cases by shserial serp.

*save file*.

GET
  FILE='C:\person level nutrient and food DATE TODAY.sav'.

sort cases by shserial serp.

SAVE OUTFILE='C:\30 FGs.sav'
/ KEEP= Shserial serp
 WTA6FG1
WTA6FG2
WTA6FG3
WTA6FG4
WTA6FG5
WTA6FG6
WTA6FG7
WTA6FG8
WTA6FG9
WTA6FG10
WTA6FG11
WTA6FG12
WTA6FG13
WTA6FG14
```

```
WTA6FG15
WTA6FG16
WTA6FG17
WTA6FG18
WTA6FG19
WTA6FG20
WTA6FG21
WTA6FG22
WTA6FG23
WTA6FG24
WTA6FG25
WTA6FG26
WTA6FG27
WTA6FG28
WTA6FG29
WTA6FG30
 /COMPRESSED.

GET FILE ='C:\30 FGs.sav'.

sort cases by shserial serp.

*save file*.

GET
  FILE='C:\ person level nutrient and food DATE TODAY.sav’.

sort cases by shserial serp.

MATCH FILES /FILE=*
 /TABLE='C:\Chapter 04\30 FGs.sav'
 /BY shserial serp.
EXECUTE.

**create supplement food group for wta6.**.
compute wta6fg31=wta5fg83.
var label wta6fg31 '(D) Supplements'.
```

Chapter 12: Anthropometry

WSTOKB (D) Whether waist measurements are valid

```
 1 Usable 1st & 2nd measurements
 2 Usable 1st & 3rd measurements
 3 Usable 2nd & 3rd measurements
 4 Usable 1st & 2nd & 3rd measurements
 5 Not useable: unreliable
 6 Not useable: difference > 3cm'
 7 Not useable: only one measurement
 8 Refused
 9 Not attempted
 10 late refusal
 -1 no nurse visit or less than 11 years old
 -9 No answer/refusal
```

HIPOKB (D) Whether hip measurements are valid

```
 1 Usable 1st & 2nd measurements
 2 Usable 1st & 3rd measurements
 3 Usable 2nd & 3rd measurements
 4 Usable 1st & 2nd & 3rd measurements
 5 Not useable: unreliable
 6 Not useable: difference > 3cm
 7 Not useable:only 1 measurement
 8 Refused
 9 Not attempted
 10 late refusal
 -9 No answer/refusal
 -1 less than 11 yrs old or no measurement
```

WHOKB (D) Whether waist/hip measure is valid

```
 1 Valid
 2 Waist/Hip not usable
 3 Waist/Hip partial response
 4 Waist/Hip refused
 5 Waist/Hip not attempted
 -9 No answer/refusal
 -1 less than 11 yrs old or no measurement
```

WSTVAL (D) Valid mean waist (cm)
HIPVAL (D) Valid mean hip (cm)
WHVAL (D) Valid mean waist/hip ratio

```
GET
  FILE='C:\Individual.sav'.

MIS VAL whintro waist waist2 waist3 wjrel pregnt (-9, -1).

if shserial=1020594 and serp=2 waist=-9.
if shserial=1020594 and serp=2 waist2=-9.
if shserial=1020594 and serp=2 respwh=2.
if shserial=1020594 and serp=2 ynowh=2.
if shserial=1020594 and serp=2 whpnabm1=5.
if shserial=1020594 and serp=2 wjrel=-1.
execute.

if shserial=1118874 and serp=1 whintro=2.
execute.
```

```
RECODE whintro (1=10)(2=8)(3=9)(ELSE=COPY) INTO wstokb.
COMPUTE xxwst12=abs(waist-waist2).
COMPUTE xxwst13=abs(waist-waist3).
COMPUTE xxwst23=abs(waist2-waist3).
IF whintro=1 & xxwst12<=3 & any(wjrel,1,2,3) wstokb=1.
DO IF whintro=1 & xxwst12>3.
COMPUTE wstokb=6.
IF xxwst13<=3 wstokb=2.
IF xxwst23<=3 wstokb=3.
IF xxwst13<=3 & xxwst23<=3 wstokb=4.
END IF.
DO IF wstokb=10.
IF waist>0 wstokb=7.
END IF.
IF ANY(wjrel,4,-9) wstokb=5.
IF agep<11 wstokb=-1.
VARIABLE LABELS wstokb "(D) Whether waist measurements are valid".
VALUE LABELS wstokb
 1 'Usable 1st & 2nd measurements'
 2 'Usable 1st & 3rd measurements'
 3 'Usable 2nd & 3rd measurements'
 4 'Usable 1st & 2nd & 3rd measurements'
 5 'Not useable: unreliable'
 6 'Not useable: difference > 3cm'
 7 'Not useable: only one measurement'
 8 'Refused'
 9 'Not attempted'
 10'late refusal'
 -1 'no nurse visit or less than 11 years old'
-9 'No answer/refusal'.
MIS val wstokb (-90, -9, -1).


MIS VAL whintro hip hip2 hip3 hjrel (-9, -1).


RECODE whintro (1=10)(2=8)(3=9)(-9,-6,-2,-1=COPY) INTO hipokb.
COMPUTE xxhip12=abs(hip-hip2).
COMPUTE xxhip13=abs(hip-hip3).
COMPUTE xxhip23=abs(hip2-hip3).
IF whintro=1 & xxhip12<=3 & any(hjrel,1,2,3) hipokb=1.
DO IF whintro=1 & xxhip12>3.
COMPUTE hipokb=6.
IF xxhip13<=3 hipokb=2.
IF xxhip23<=3 hipokb=3.
IF xxhip13<=3 & xxhip23<=3 hipokb=4.
END IF.
DO IF hipokb=10.
IF hip>0 hipokb=7.
END IF.
IF ANY(hjrel,4,-9) hipokb=5.
IF agep<11 hipokb=-1.
VARIABLE LABELS hipokb "(D) Whether hip measurements are valid".
VALUE LABELS hipokb
 1 'Usable 1st & 2nd measurements'
 2 'Usable 1st & 3rd measurements'
 3 'Usable 2nd & 3rd measurements'
 4 'Usable 1st & 2nd & 3rd measurements'
 5 'Not useable: unreliable'
 6 'Not useable: difference > 3cm'
 7 'Not useable:only 1 measurement'
 8 'Refused'
 9 'Not attempted'
10'late refusal'
-9 'No answer/refusal'
 -1 'less than 11 yrs old or no measurement'.
MIS VAL hipokb (-90,-9,-1).

RECODE wstokb(-9 thru -1=COPY) into whokb.
IF RANGE(wstokb,1,4) & RANGE(hipokb,1,4) whokb=1.
IF ANY(5,wstokb,hipokb) | ANY(6,wstokb,hipokb) whokb=2.
IF ANY(7,wstokb,hipokb) whokb=3.
IF ANY(8,wstokb,hipokb) whokb=4.
IF ANY(9,wstokb,hipokb) whokb=5.
IF ANY (10,wstokb,hipokb) whokb=4.
```

```
IF agep<11 whokb=-1.
VARIABLE LABELS whokb "(D) Whether waist/hip measure is valid".
VALUE LABELS whokb
  1 "Valid"
  2 "Waist/Hip not usable"
  3 "Waist/Hip partial response"
  4 "Waist/Hip refused"
  5 "Waist/Hip not attempted"
-9 'No answer/refusal'
 -1 'less than 11 yrs old or no measurement'.
MIS VAL whokb (-90, -9, -1).

COMPUTE wstval=-1.
IF wstokb=1 wstval=(waist+waist2)/2.
IF wstokb=2 wstval=(waist+waist3)/2.
IF wstokb=3 wstval=(waist2+waist3)/2.
IF wstokb=4 wstval=(waist+waist2+waist3)/3.
VARIABLE LABEL wstval "(D) Valid Mean Waist (cm)".
MIS VAL wstval (-1).

COMPUTE hipval=-1.
IF hipokb=1 hipval=(hip+hip2)/2.
IF hipokb=2 hipval=(hip+hip3)/2.
IF hipokb=3 hipval=(hip2+hip3)/2.
IF hipokb=4 hipval=(hip+hip2+hip3)/3.
VARIABLE LABEL hipval "(D) Valid Mean Hip (cm)".
MIS VAL hipval (-1).

COMPUTE whval=-1.
IF whokb=1 whval=wstval/hipval.
VARIABLE LABEL whval "(D) Valid Mean Waist/Hip ratio".
MIS VAL whval (-1).
```

MENWHGP (D) Male waist/hip ratio groups

```
 1 Less than 0.80
 2 0.80, less than 0.85
 3 0.85, less than 0.90
 4 0.90, less than 0.95
 5 0.95, less than 1.00
 6 1.00 or more
```

MENWHHI (D) Male high waist/hip ratio

```
 1 Less than 0.95
 2 0.95 or more
```

WOMWHGP (D) Female waist/hip ratio groups

```
 1 Less than 0.70
 2 0.70, less than 0.75
 3 0.75, less than 0.80
 4 0.80, less than 0.85
 5 0.85, less than 0.90
 6 0.90 or more
 -90 Pregnant
```

WOMWHHI (D) Female high waist/hip ratio

```
 1 Less than 0.85
 2 0.85 or more
 -90 Pregnant
```

MWSTHI (D) Male high waist circumference

```
 1 Less than 102cm
 2 102cm or more
```

FWSTHI (D) Female high waist circumference

```
 1 Less than 88cm
 2 88cm or more
```

```
GET
  FILE='C:\Individual.sav'.

do if sexp=1 .
RECODE whokb (-99 thru -1=COPY)(2 thru 5=-1) into menwhgp.
RECODE whval (1.00 thru hi=6)(0.95 thru 1.00=5)(0.90 thru 0.95=4)(0.85 thru 0.90=3)
(0.80 thru 0.85=2)(0.01 thru 0.80=1) into menwhgp.
VAR LAB menwhgp '(D) Male waist hip ratio groups'.
VAL LAB menwhgp
    1 'Less than 0.80'
    2 '0.80, less than 0.85'
    3 '0.85, less than 0.90'
    4 '0.90, less than 0.95'
    5 '0.95, less than 1.00'
    6 '1.00 or more'.
recode menwhgp (1 thru 4=1)(5 thru 6=2)(-99 thru -1=copy) into menwhhi.
VAR LAB menwhhi '(D) Male high waist hip ratio'.
VAL LAB menwhhi
  1 'Less than 0.95'
  2 '0.95 or more'.
end if.
if sexp=2 menwhgp=-1.
if sexp=2 menwhhi=-1.
MIS VAL menwhgp menwhhi (-1, -2).

do if sexp=2.
RECODE whokb (-99 thru -1=COPY)(2 thru 5=-1) into womwhgp.
RECODE whval (0.90 thru hi=6)(0.85 thru 0.90=5)(0.80 thru 0.85=4)(0.75 thru 0.80=3)
  (0.70 thru 0.75=2)(0.01 thru 0.70=1) into womwhgp.
recode womwhgp (1 thru 4=1)(5 thru 6=2)(-99 thru -1=copy) into womwhhi.
VAR LAB womwhgp '(D) Female waist hip ratio groups'.
VAL LAB womwhgp
    1 'Less than 0.70'
    2 '0.70, less than 0.75'
    3 '0.75, less than 0.80'
    4 '0.80, less than 0.85'
    5 '0.85, less than 0.90'
    6 '0.90 or more'
    -90 'Pregnant'.
VAR LAB womwhhi '(D) Female high waist hip ratio'.
VAL LAB womwhhi
  1 'Less than 0.85'
  2 '0.85 or more'
-90 'Pregnant'.
end if.
if sexp=1 womwhgp=-1.
if sexp=1 womwhhi=-1.
MIS VAL womwhgp womwhhi (-90 thru -1).

do if sexp=1 .
RECODE
  wstval
  (50 thru 101.99999=1)  (102 thru Highest=2) (else=copy) INTO  mwsthi .
VARIABLE LABEL mwsthi '(D) Male high waist circumference'.
VALUE LABELS mwsthi 1 'Less than 102cm' 2 '102cm or more'.
end if.
if sexp=2 mwsthi=-1.
MIS VAL mwsthi (-1,-2).
execute.

do if sexp=2.
RECODE
  wstval
  (50 thru 87.99999=1)  (88 thru Highest=2)  (else=copy) INTO  fwsthi .
VARIABLE LABEL fwsthi '(D) Female high waist circumference'.
VALUE LABELS fwsthi 1 'Less than 88cm' 2 '88cm or more'.
end if.
if sexp=1 fwsthi=-1.
MIS VAL fwsthi (-1,-2).
execute.
```

ARMOK (D) Whether arm circumference measurements are valid

1 Usable 1st & 2nd measurements

```
 2 Usable 1st & 3rd measurements
 3 Usable 2nd & 3rd measurements
 4 Usable 1st & 2nd & 3rd measurements
 5 Not useable: unreliable
 6 Not useable: difference > 1.5cm
 7 Refused
 8 Attempted but not obtained
```

ARMVAL (D) Valid mean MUAC (cm)

```
GET
  FILE='C:\Individual.sav'.

MIS VAL cuprel cuprel2 cuprel3 (-9, -8, -1).

RECODE cuprel (1=1) (2=0) (-9, -8, -1=COPY) INTO cuprela.
RECODE cuprel2 (1=1) (2=0) (-9, -8, -1=COPY) INTO cuprel2a.
RECODE cuprel3 (1=1) (2=0) (-9, -8, -1=COPY) INTO cuprel3a.
VAL LAB cuprela cuprel2a cuprel3a 1'YES' 0'NO' -1'Not applicable'.
MIS VAL cuprela cuprel2a cuprel3a (-9, -8, -1).

COMPUTE armrel12 = (cuprela + cuprel2a).
COMPUTE armrel13 = (cuprela + cuprel3a).
COMPUTE armrel23 = (cuprel2a + cuprel3a).
VAL LAB armrel12 armrel13 armrel23
  1'only one valid measurement'
  2 'two valid measurements'.

MIS VAL muacint (-1).

RECODE muacint (1=9)(2=7)(3=8)(ELSE=COPY) INTO armok.
COMPUTE xxac12=abs(cuparm-cuparm2).
COMPUTE xxac13=abs(cuparm-cuparm3).
COMPUTE xxac23=abs(cuparm2-cuparm3).
IF armok=9 & xxac12<=1.5 & armrel12=2 armok=1.
DO IF armok=9 & xxac12>1.5.
IF xxac13<=1.5 & armrel13=2 armok=2.
IF xxac23<=1.5 & armrel23=2 armok=3.
IF xxac13<=1.5 & xxac23<=1.5 & armrel13=2 & armrel23=2 armok=4.
END IF.

RECODE armok (9=5) (else=copy).


VARIABLE LABELS armok "(D) Whether ht measurements are valid".
VALUE LABELS armok
 1 'Usable 1st & 2nd measurements'
 2 'Usable 1st & 3rd measurements'
 3 'Usable 2nd & 3rd measurements'
 4 'Usable 1st & 2nd & 3rd measurements'
 5 'Not useable: unreliable'
 6 'Not useable: difference > 1.5cm'
 7 'Refused'
 8 'Attempted but not obtained'.
MIS VAL armok (-1).

COMPUTE armval=-1.
IF armok=1 armval=(cuparm+cuparm2)/2.
IF armok=2 armval=(cuparm+cuparm3)/2.
IF armok=3 armval=(cuparm2+cuparm3)/2.
IF armok=4 armval=(cuparm+cuparm2+cuparm3)/3.
VARIABLE LABEL armval "(D) Valid Mean MUAC (cm)".
execute.
MIS VAL armval (-1).
```

SPANOK (D) Whether demi span measurements are valid

```
 1 Usable 1st & 2nd measurements
 2 Not useable: unreliable
 3 Not useable: difference > 3.0cm
```

```
 4 Refused
 5 Unable to measure ds for other reason than refused
 6 only one measurement taken
```

SPANOK1 (D) Valid demi span grouped

```
 1 Valid
 2 Not usable
 3 Refused
 4 Attempted but not obtained
```

SPANVAL (D) Valid mean demi span (cm)

SPANHT (D) Height equivalent of demi span (cm)

```
GET
  FILE='C:\Individual.sav'.

MIS VAL spanint span span2 (-9, -8, -1).

RECODE spanint (1=6)(2=4)(3=5) (-9,-8,-1=COPY) INTO spanok.
COMPUTE xxspan=abs(span-span2).
IF ANY(-8, span, span2) xxspan=-8.
IF ANY(-9, span, span2) xxspan=-9.
IF ANY(-1, span,span2) xxspan=-1.
IF (spanint=1 & xxspan<=3.0 & spanrel=1 & spanrel2=1) spanok=1.
DO IF spanint=1 & xxspan>3.0.
COMPUTE spanok=3.
END IF.
DO IF spanint=1 & spanrel=2 | spanrel2=2.
COMPUTE spanok=2.
END IF.
IF ANY(-8, xxspan)spanok =-8.
IF ANY(-9, xxspan)spanok =-9 .
IF ANY(-1, xxspan)spanok =-1.
VARIABLE LABELS spanok "(D) Whether demi span measurements are valid".
VALUE LABELS spanok
 1 'Usable 1st & 2nd measurements'
 2 'Not useable: unreliable'
 3 'Not useable: difference > 3.0cm'
 4 'Refused'
 5 'Unable to measure ds for other reason than refused'
 6 'only one measurement taken'.
MIS VAL spanok (-9, -8, -1).



RECODE spanok (1=1) (2 thru 3=2) (4=3) (5=4) (6=2) into spanok1.
MIS val spanok (-1).
VAR LAB spanok1 '(D) Valid demi span grouped'.
VAL LAB spanok1
 1'Valid'
 2'Not usable'
 3'Refused'
 4'Attempted but not obtained'.

COMPUTE spanval=-1.
DO IF spanok=1.
COMPUTE spanval=(span+span2)/2.
END IF.
MIS VAL spanval (-1).
VARIABLE LABEL spanval "(D) Valid Mean span (cm)".
EXECUTE.

DO IF sexp=2.
COMPUTE spanht=1.35 * spanval + 60.1 .
ELSE IF sexp=1.
COMPUTE spanht=1.40 * spanval + 57.8.
END IF.
VAR LAB spanht '(D) Height equivalent of demi span'.
EXECUTE.
```

BMICUT (D) UK BMI national classification standards (85th/95th centile)

```
 1 normal-weight
 2 over-weight
 3 obese
```

BMICUT2 (D) UK BMI status (ovrght inc. obese)

```
 1 Neither overweight nor obese
 2 Overweight incl. obese
```

BMICUT3 (D) UK BMI status (non-obese vs obese)

```
 1 Non-obese
 2 Obese
```

NB The derived variable 'INTEXAGE – (D) Exact age at interview' has been added to the data-set without providing the derived variable syntax. This information cannot be documented due to reasons of confidentiality.

```
GET
  FILE='C:\Individual.sav'.

MIS VAL weight height (-1).

COMPUTE htok=2.
IF height>0 htok=1.
VAR LAB htok '(D) valid height'.
VAL LAB htok 1'valid' .


COMPUTE wtok=2.
IF weight>0 wtok=1.
VAR LAB wtok '(D) valid weight'.
VAL LAB  wtok 1'valid'.


compute heightm = (height/100).
execute.
variable label heightm '(I) Height in metres'.

do if wtok=1 and htok=1.
compute bmi2=weight/(heightm*heightm).
end if.
execute.

rename variables (bmi=bmiold).
rename variables (bmi2=bmi).

VARIABLE LABELS BMI '(D) Body mass index - replacement variable'.

COMPUTE bmiok=2.
IF bmi>0 bmiok=1.
VAR LAB bmiok '(D) valid BMI'.
VAL LAB  bmiok 1'valid'.
freq bmiok.

USE ALL.
COMPUTE filter_$=(agep >= 2 & agep < 19 & bmiok=1).
VARIABLE LABEL filter_$ 'agep >= 2 & agep < 19 (FILTER)'.
VALUE LABELS filter_$  0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .

compute bmicut=1.
IF sexp=1 AND RANGE(intexage, 2, 2.49) AND RANGE(bmi,18.12,19.10) bmicut=2.
IF sexp=2 AND RANGE(intexage, 2, 2.49) AND RANGE(bmi,17.83, 18.84) bmicut=2.
IF sexp=1 AND RANGE(intexage, 2.5, 2.99) AND RANGE(bmi,17.80,18.77) bmicut=2.
IF sexp=2 AND RANGE(intexage, 2.5, 2.99) AND RANGE(bmi,17.55,18.56) bmicut=2.


IF sexp=1 AND RANGE(intexage, 3, 3.49) AND RANGE(bmi,17.55,18.51) bmicut=2.
```

```
IF sexp=2 AND RANGE(intexage, 3, 3.49) AND RANGE(bmi,17.39,18.42) bmicut=2.
IF sexp=1 AND RANGE(intexage, 3.5, 3.99) AND RANGE(bmi,17.32,18.27 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 3.5, 3.99) AND RANGE(bmi,17.29,18.35) bmicut=2.


IF sexp=1 AND RANGE(intexage, 4, 4.49) AND RANGE(bmi,17.13,18.08) bmicut=2.
IF sexp=2 AND RANGE(intexage, 4, 4.49) AND RANGE(bmi,17.23,18.32) bmicut=2.
IF sexp=1 AND RANGE(intexage, 4.5, 4.99) AND RANGE(bmi,17.01,17.97) bmicut=2.
IF sexp=2 AND RANGE(intexage, 4.5, 4.99) AND RANGE(bmi,17.17,18.31) bmicut=2.


IF sexp=1 AND RANGE(intexage, 5, 5.49) AND RANGE(bmi,16.96,17.95 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 5, 5.49) AND RANGE(bmi,17.16 ,18.35) bmicut=2.
IF sexp=1 AND RANGE(intexage, 5.5, 5.99) AND RANGE(bmi,16.96,17.99) bmicut=2.
IF sexp=2 AND RANGE(intexage, 5.5, 5.99) AND RANGE(bmi,17.21 ,18.46) bmicut=2.


IF sexp=1 AND RANGE(intexage, 6, 6.49) AND RANGE(bmi,17.01 ,18.10) bmicut=2.
IF sexp=2 AND RANGE(intexage, 6, 6.49) AND RANGE(bmi,17.32 ,18.65) bmicut=2.
IF sexp=1 AND RANGE(intexage, 6.5, 6.99) AND RANGE(bmi,17.10, 18.26) bmicut=2.
IF sexp=2 AND RANGE(intexage, 6.5, 6.99) AND RANGE(bmi,17.49 ,18.91) bmicut=2.


IF sexp=1 AND RANGE(intexage, 7, 7.49) AND RANGE(bmi,17.24 ,18.48) bmicut=2.
IF sexp=2 AND RANGE(intexage, 7, 7.49) AND RANGE(bmi,17.71 ,19.22) bmicut=2.
IF sexp=1 AND RANGE(intexage, 7.5, 7.99) AND RANGE(bmi,17.41 ,18.74) bmicut=2.
IF sexp=2 AND RANGE(intexage, 7.5, 7.99) AND RANGE(bmi,17.96 ,19.56) bmicut=2.


IF sexp=1 AND RANGE(intexage, 8, 8.49) AND RANGE(bmi,17.61 ,19.04 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 8, 8.49) AND RANGE(bmi,18.23 ,19.93) bmicut=2.
IF sexp=1 AND RANGE(intexage, 8.5, 8.99) AND RANGE(bmi,17.83 ,19.36) bmicut=2.
IF sexp=2 AND RANGE(intexage, 8.5, 8.99) AND RANGE(bmi, 18.52,20.30 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 9, 9.49) AND RANGE(bmi,18.08 ,19.70 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 9, 9.49) AND RANGE(bmi,18.82 ,20.70) bmicut=2.
IF sexp=1 AND RANGE(intexage, 9.5, 9.99) AND RANGE(bmi, 18.35, 20.05) bmicut=2.
IF sexp=2 AND RANGE(intexage, 9.5, 9.99) AND RANGE(bmi,19.15 ,21.10) bmicut=2.


IF sexp=1 AND RANGE(intexage, 10,10.49) AND RANGE(bmi,18.64 ,20.42 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 10,10.49) AND RANGE(bmi,19.49 ,21.52) bmicut=2.
IF sexp=1 AND RANGE(intexage, 10.5,10.99) AND RANGE(bmi,18.94 ,20.79) bmicut=2.
IF sexp=2 AND RANGE(intexage, 10.5,10.99) AND RANGE(bmi,19.85, 21.94) bmicut=2.


IF sexp=1 AND RANGE(intexage, 11,11.49) AND RANGE(bmi,19.26 ,21.18 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 11,11.49) AND RANGE(bmi,20.22 ,22.36) bmicut=2.
IF sexp=1 AND RANGE(intexage, 11.5,11.99) AND RANGE(bmi,19.59 ,21.57) bmicut=2.
IF sexp=2 AND RANGE(intexage, 11.5,11.99) AND RANGE(bmi,20.60,22.80 ) bmicut=2.

IF sexp=1 AND RANGE(intexage, 12,12.49) AND RANGE(bmi,19.93,21.96 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 12,12.49) AND RANGE(bmi,20.98,23.22) bmicut=2.
IF sexp=1 AND RANGE(intexage, 12.5,12.99) AND RANGE(bmi,20.29,22.36) bmicut=2.
IF sexp=2 AND RANGE(intexage, 12.5,12.99) AND RANGE(bmi,21.37,23.65 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 13,13.49) AND RANGE(bmi,20.65,22.77 ) bmicut=2.
IF sexp=2 AND RANGE(intexage, 13,13.49) AND RANGE(bmi,21.74 ,24.06) bmicut=2.
IF sexp=1 AND RANGE(intexage, 13.5,13.99) AND RANGE(bmi,21.02 ,23.17) bmicut=2.
IF sexp=2 AND RANGE(intexage, 13.5,13.99) AND RANGE(bmi,22.10,24.45 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 14,14.49) AND RANGE(bmi,21.39,23.58) bmicut=2.
IF sexp=2 AND RANGE(intexage, 14,14.49) AND RANGE(bmi,22.45,24.82) bmicut=2.
IF sexp=1 AND RANGE(intexage, 14.5,14.99) AND RANGE(bmi,21.76 ,23.97) bmicut=2.
IF sexp=2 AND RANGE(intexage, 14.5,14.99) AND RANGE(bmi,22.77, 25.16) bmicut=2.


IF sexp=1 AND RANGE(intexage, 15,15.49) AND RANGE(bmi,22.12,24.36) bmicut=2.
IF sexp=2 AND RANGE(intexage, 15,15.49) AND RANGE(bmi,23.08,25.49) bmicut=2.
IF sexp=1 AND RANGE(intexage, 15.5,15.99) AND RANGE(bmi,22.48,24.74) bmicut=2.
IF sexp=2 AND RANGE(intexage, 15.5,15.99) AND RANGE(bmi,23.35,25.78 ) bmicut=2.
```

```
IF sexp=1 AND RANGE(intexage, 16,16.49) AND RANGE(bmi,22.82,25.09) bmicut=2.
IF sexp=2 AND RANGE(intexage, 16,16.49) AND RANGE(bmi,23.61,26.05) bmicut=2.
IF sexp=1 AND RANGE(intexage, 16.5,16.99) AND RANGE(bmi,23.15,25.44) bmicut=2.
IF sexp=2 AND RANGE(intexage, 16.5,16.99) AND RANGE(bmi,23.84,26.29 ) bmicut=2.


IF sexp=1 AND RANGE(intexage, 17,17.49) AND RANGE(bmi,23.46,25.77) bmicut=2.
IF sexp=2 AND RANGE(intexage, 17,17.49) AND RANGE(bmi,24.06,26.52) bmicut=2.
IF sexp=1 AND RANGE(intexage, 17.5,17.99) AND RANGE(bmi,23.76,26.08) bmicut=2.
IF sexp=2 AND RANGE(intexage, 17.5,17.99) AND RANGE(bmi,24.25,26.72 ) bmicut=2.

IF sexp=1 AND RANGE(intexage, 18,18.49) AND RANGE(bmi,24.05,26.37) bmicut=2.
IF sexp=2 AND RANGE(intexage, 18,18.49) AND RANGE(bmi,24.43,26.91) bmicut=2.
IF sexp=1 AND RANGE(intexage, 18.5,18.99) AND RANGE(bmi,24.32,26.65) bmicut=2.
IF sexp=2 AND RANGE(intexage, 18.5,18.99) AND RANGE(bmi,24.60,27.08 ) bmicut=2.
EXECUTE.


IF sexp=1 AND RANGE(intexage, 2, 2.49) AND (bmi ge 19.1001) bmicut=3.
IF sexp=2 AND RANGE(intexage, 2, 2.49) AND (bmi ge 18.8401) bmicut=3.
IF sexp=1 AND RANGE(intexage, 2.5, 2.99) AND (bmi ge 18.7701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 2.5, 2.99) AND (bmi ge 18.5601) bmicut=3.

IF sexp=1 AND RANGE(intexage, 3, 3.49) AND (bmi ge 18.5101) bmicut=3.
IF sexp=2 AND RANGE(intexage, 3, 3.49) AND (bmi ge 18.4201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 3.5, 3.99) AND (bmi ge 18.2701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 3.5, 3.99) AND (bmi ge 18.3501 ) bmicut=3.

IF sexp=1 AND RANGE(intexage, 4, 4.49) AND (bmi ge 18.0801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 4, 4.49) AND (bmi ge 18.3201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 4.5, 4.99) AND (bmi ge 17.9701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 4.5, 4.99) AND (bmi ge 18.3101) bmicut=3.

IF sexp=1 AND RANGE(intexage, 5, 5.49) AND (bmi ge 17.9501) bmicut=3.
IF sexp=2 AND RANGE(intexage, 5, 5.49) AND (bmi ge 18.3501) bmicut=3.
IF sexp=1 AND RANGE(intexage, 5.5, 5.99) AND (bmi ge 17.9901) bmicut=3.
IF sexp=2 AND RANGE(intexage, 5.5, 5.99) AND (bmi ge 18.4601) bmicut=3.

IF sexp=1 AND RANGE(intexage, 6, 6.49) AND (bmi ge 18.1001) bmicut=3.
IF sexp=2 AND RANGE(intexage, 6, 6.49) AND (bmi ge 18.6501) bmicut=3.
IF sexp=1 AND RANGE(intexage, 6.5, 6.99) AND (bmi ge 18.2601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 6.5, 6.99) AND (bmi ge 18.9101) bmicut=3.

IF sexp=1 AND RANGE(intexage, 7, 7.49) AND (bmi ge 18.4801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 7, 7.49) AND (bmi ge 19.2201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 7.5, 7.99) AND (bmi ge 18.7401) bmicut=3.
IF sexp=2 AND RANGE(intexage, 7.5, 7.99) AND (bmi ge 19.5601) bmicut=3.

IF sexp=1 AND RANGE(intexage, 8, 8.49) AND (bmi ge 19.0401) bmicut=3.
IF sexp=2 AND RANGE(intexage, 8, 8.49) AND (bmi ge 19.9301) bmicut=3.
IF sexp=1 AND RANGE(intexage, 8.5, 8.99) AND (bmi ge 19.3601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 8.5, 8.99) AND (bmi ge 20.3001) bmicut=3.

IF sexp=1 AND RANGE(intexage, 9, 9.49) AND (bmi ge 19.7001) bmicut=3.
IF sexp=2 AND RANGE(intexage, 9, 9.49) AND (bmi ge 20.7001) bmicut=3.
IF sexp=1 AND RANGE(intexage, 9.5, 9.99) AND (bmi ge 20.0501) bmicut=3.
IF sexp=2 AND RANGE(intexage, 9.5, 9.99) AND (bmi ge 21.1001) bmicut=3.

IF sexp=1 AND RANGE(intexage, 10,10.49) AND (bmi ge 20.4201) bmicut=3.
IF sexp=2 AND RANGE(intexage, 10,10.49) AND (bmi ge 21.5201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 10.5,10.99) AND (bmi ge 20.7901) bmicut=3.
IF sexp=2 AND RANGE(intexage, 10.5,10.99) AND (bmi ge 21.9401) bmicut=3.

IF sexp=1 AND RANGE(intexage, 11,11.49) AND (bmi ge 20.1801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 11,11.49) AND (bmi ge 22.3601) bmicut=3.
IF sexp=1 AND RANGE(intexage, 11.5,11.99) AND (bmi ge 21.5701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 11.5,11.99) AND (bmi ge 22.8001) bmicut=3.

IF sexp=1 AND RANGE(intexage, 12,12.49) AND (bmi ge 21.9601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 12,12.49) AND (bmi ge 23.2201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 12.5,12.99) AND (bmi ge 22.3601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 12.5,12.99) AND (bmi ge 23.6501) bmicut=3.

IF sexp=1 AND RANGE(intexage, 13,13.49) AND (bmi ge 22.7701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 13,13.49) AND (bmi ge 24.0601) bmicut=3.
IF sexp=1 AND RANGE(intexage, 13.5,13.99) AND (bmi ge 23.1701) bmicut=3.
```

```
IF sexp=2 AND RANGE(intexage, 13.5,13.99) AND (bmi ge 24.4501) bmicut=3.

IF sexp=1 AND RANGE(intexage, 14,14.49) AND (bmi ge 23.5801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 14,14.49) AND (bmi ge 24.8201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 14.5,14.99) AND (bmi ge 23.9701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 14.5,14.99) AND (bmi ge 25.1601) bmicut=3.

IF sexp=1 AND RANGE(intexage, 15,15.49) AND (bmi ge 24.3601) bmicut=3.
IF sexp=2 AND RANGE(intexage, 15,15.49) AND (bmi ge 25.4901) bmicut=3.
IF sexp=1 AND RANGE(intexage, 15.5,15.99) AND (bmi ge 24.7401) bmicut=3.
IF sexp=2 AND RANGE(intexage, 15.5,15.99) AND (bmi ge 25.7801) bmicut=3.


IF sexp=1 AND RANGE(intexage, 16,16.49) AND (bmi ge 25.0901) bmicut=3.
IF sexp=2 AND RANGE(intexage, 16,16.49) AND (bmi ge 26.0501) bmicut=3.
IF sexp=1 AND RANGE(intexage, 16.5,16.99) AND (bmi ge 25.4401) bmicut=3.
IF sexp=2 AND RANGE(intexage, 16.5,16.99) AND (bmi ge 26.2901) bmicut=3.

IF sexp=1 AND RANGE(intexage, 17,17.49) AND (bmi ge 25.7701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 17,17.49) AND (bmi ge 26.5201) bmicut=3.
IF sexp=1 AND RANGE(intexage, 17.5,17.99) AND (bmi ge 26.0801) bmicut=3.
IF sexp=2 AND RANGE(intexage, 17.5,17.99) AND (bmi ge 26.7201) bmicut=3.

IF sexp=1 AND RANGE(intexage, 18,18.49) AND (bmi ge 26.3701) bmicut=3.
IF sexp=2 AND RANGE(intexage, 18,18.49) AND (bmi ge 26.9101) bmicut=3.
IF sexp=1 AND RANGE(intexage, 18.5,18.99) AND (bmi ge 26.6501) bmicut=3.
IF sexp=2 AND RANGE(intexage, 18.5,18.99) AND (bmi ge 27.0801) bmicut=3.
EXECUTE.

VAR LAB bmicut '(D) UK BMI national classification standards (85th/95th centile)'.
value labels bmicut
1 'normal-weight'
2 'over-weight'
3 'obese'.

RECODE bmicut (1=1) (2 thru 3=2) INTO bmicut2.
VAR LAB bmicut2 '(D) UK BMI status (ovrght inc. obese)'.
VAL LAB bmicut2 1 'Neither overweight nor obese'  2 'Overweight incl. obese'.

RECODE bmicut (1 thru 2=1) (3=2) INTO bmicut3.
VAR LAB bmicut3 '(D) UK BMI status (non-obese vs obese)'.
VAL LAB bmicut3 1 'Non-obese'  2 'Obese'.
```

Chapter 13: Blood Pressure

Note: Frequency counts of BPRESPC and HY140om contain 1 missing value. This is due to a post-derivation edit to the data.

BPRESPC (D) Whether BP readings are valid

```
1 Three valid BP measurements
2 Ate, drank, smoked &/or exercised in previous half hour
3 Not known if ate, drank, smoked or exercised
4 Three valid readings not obtained
5 Pregnant
6 Refused, attempted but not obtained, not attempted
```

```
SPSS Syntax

RECODE respbps (1=1) (2, 3=4) (4, 5, 6=6) (-9 thru -1 =COPY) into bprespc.
IF ANY (999, sys, sys2, sys3, dias, dias2, dias3) bprespc = 4.
IF ANY (-9, sys, sys2, sys3, dias, dias2, dias3) bprespc = 4.
IF ANY (-8, sys, sys2, sys3, dias, dias2, dias3) bprespc = 4.
IF (respbps=1 & ANY(consubx1, 1, 2, 3, 4)) bprespc=2.
IF (respbps=1 & ANY(consubx1, -9,-8)) bprespc= 3.
IF (respbps=1 & ANY(consubx5, -9,-8)) bprespc= 3.
IF (respbps=1 & ANY(consubx5, 1, 2)) bprespc=2.
IF ANY (1, pregnt, upreg) bprespc= 5.
VAR LAB bprespc "(D) Whether BP readings are valid" .
VALUE LABELS bprespc 1'Three valid BP measurements' 2 'Ate, drank, smoked &/or
exercised in previous half hour' 3 'Not known if ate, drank, smoked or exercised' 4
'Three valid readings not obtained' 5 'Pregnant' 6 'Refused, attempted but not
obtained, not attempted'.
```

BPMEDC (D) Whether taking drugs that affect BP
BPMEDD (D) Whether taking drugs prescribed for BP
DIUR (D) Whether taking a diuretic
BETA (D) Whether taking a beta-blocker
ACEINH (D) Whether taking an ACE inhibitor
CALCIUMB (D) Whether taking a calcium blocker
OBPDRUG (D) Whether taking another drug for hypertension
LIPID (D) Whether taking a lipid-lowering drug
IRON (D) Whether taking iron

```
0 Not taking drug
1 Taking drug
```

```
SPSS Syntax

RECODE
  drc1 drc2 drc3 drc4 drc5 drc6 drc7 drc8 drc9 drc10 drc11 drc12 drc13 drc14 drc15
drc16 drc18 (CONVERT)
  (MISSING=SYSMIS)  INTO  drc1J  drc2J  drc3J  drc4J  drc5J  drc6J  drc7J  drc8J
drc9J  drc10J  drc11J  drc12J  drc13J
  drc14J  drc15J  drc16J  drc18J .
VARIABLE LABELS drc1J 'BNF code numeric' /drc2J 'BNF code numeric' /drc3J 'BNF code
numeric' /drc4J 'BNF code numeric' /drc5J
  'BNF code numeric' /drc6J 'BNF code numeric' /drc7J 'BNF code numeric' /drc8J 'BNF
code numeric' /drc9J 'BNF code numeric'
 /drc10J 'BNF code numeric' /drc11J 'BNF code numeric' /drc12J 'BNF code numeric'
/drc13J 'BNF code numeric' /drc14J 'BNF code'+
 ' numeric' /drc15J 'BNF code numeric' /drc16J 'BNF code numeric' /drc18J 'BNF code
numeric'.
EXECUTE .
```

```
DO REPEAT xxdrug=diur beta aceinh calciumb obpdrug lipid iron bpmedc bpmedd.
COMPUTE xxdrug=0.
RECODE drc1J (-9 thru -1=COPY) INTO xxdrug.
END REPEAT.
DO REPEAT xxcode=drc1J to drc18J.
IF xxcode=0 beta =-9.
IF xxcode=0 aceinh =-9.
IF xxcode=0 calciumb =-9.
IF xxcode=0 iron =-9.
IF xxcode=0 lipid =-9.
IF xxcode=0 obpdrug =-9.
IF xxcode=0 bpmedc =-9.
IF xxcode=0 bpmedd =-9.
END REPEAT.
DO REPEAT xxcode=drc1J to drc18J.
IF RANGE (xxcode, 20201, 20208) diur=1.
IF xxcode=20400 beta =1.
IF xxcode=20505 aceinh =1.
IF xxcode=20602 calciumb =1.
IF ANY(xxcode,20501, 20502, 20503, 20504, 200506) obpdrug =1.
IF xxcode=21200 lipid =1.
IF xxcode=90101 iron =1.
END REPEAT.
IF ANY(1, diur, beta, aceinh, calciumb, obpdrug) bpmedc=1.

IF bpmedc=1 & ANY (2, ytake11, ytake12, ytake14, ytake15, ytake17, ytake18, ytake20,
ytake21, ytake23, ytake24, ytake26, ytake27, ytake29, ytake30, ytake32, ytake33,
ytake35, ytake36, ytake38, ytake39, ytake40, ytake41, ytake44, ytake47, ytake50,
ytake53, ytake56, ytake62) bpmedd=1.
VARIABLE LABELS diur "(D) Diuretics".
VARIABLE LABELS bpmedc "(D) Whether taking drugs that affect BP".
VARIABLE LABELS bpmedd "(D) Whether taking drugs prescribed for BP".
VARIABLE LABELS beta "(D) Beta".
VARIABLE LABELS aceinh "(D) Aceinh".
VARIABLE LABELS calciumb "(D) calciumb".
VARIABLE LABELS obpdrug "(D) obpdrug".
VARIABLE LABELS lipid "(D) lipid".
VARIABLE LABELS iron "(D) iron".

VALUE LABELS diur beta aceinh calciumb obpdrug lipid iron bpmedc bpmedd
0 'Not taking drug'
1 'Taking drug'.
```

OMSYSVAL (D) Omron valid mean systolic BP

OMDIAVAL (D) Omron valid mean diastolic BP

HY140OM (D) Hypertensive categories: 140/90: all prescribed drugs FOR BP (Omron readings)

1 Normotensive untreated
2 Hypertensive controlled
3 Hypertensive uncontrolled
4 Hypertensive untreated
-7 Refused, attempted but not obtained, not attempted

```
SPSS Syntax

DO REPEAT omval = omsysval omdiaval.
RECODE bprespc (lo thru 0 =COPY) (2,5=-1) (3,4=-8) (6=-7) INTO omval.
END REPEAT.
DO IF bprespc=1.
COMPUTE omsysval = (sys2+sys3)/2.
COMPUTE omdiaval = (dias2 + dias3)/2.
END IF.
VAR LAB omsysval "(D) Omron valid mean systolic BP".
VAR LAB omdiaval "(D) Omron valid mean diastolic BP".

RECODE bprespc (2 thru 5, -1=-1) (-6, -2=COPY) (6=-7) INTO hy140om.
DO IF bprespc=1.
IF ANY (bpmedd, 0, -1) & RANGE (omsysval, 0, 139.999) & RANGE (omdiaval, 0, 89.999)
hy140om=1.
IF bpmedd=1 & RANGE (omsysval, 0, 139.999) & RANGE (omdiaval, 0, 89.999) hy140om=2.
```

```
IF bpmedd=1 & (omsysval>=140 | omdiaval>=90) hy140om=3.
IF ANY (bpmedd, 0, -1) & (omsysval>=140 | omdiaval>=90) hy140om=4.
END IF.
IF (bpmedd = -9) hy140om = -9.
VARIABLE LABELS hy140om "(D) Hypertensive categories: 140/90: all prescribed drugs FOR
BP (Omron readings)".
VALUE LABELS hy140om 1 'Normotensive untreated' 2 'Hypertensive controlled' 3
'Hypertensive uncontrolled' 4 'Hypertensive untreated' -7 'Refused, attempted but not
obtained, not attempted'.
```

Chapter 14: Analytes

HGB_CUT (D) Haemoglobin under clinical reference points (gender specific)

1 Under lower limit (13g/dl men, 12 g/dl women)
2 Above lower limit

```
SPSS Syntax

DO IF (sexp = 1) .
RECODE hgb
(MISSING=SYSMIS) (Lowest thru 12.99=1) (13 thru Highest=2) INTO hgb_cut .
END IF .
VARIABLE LABELS hgb_cut '(D) Haemoglobin under clinical reference points (gender
specific)'.
EXECUTE .
DO IF (sexp = 2) .
RECODE
hgb
(MISSING=SYSMIS) (Lowest thru 11.99=1) (12 thru Highest=2) INTO hgb_cut .
END IF .
VARIABLE LABELS hgb_cut '(D) Haemoglobin under clinical reference points (gender
specific)'.
EXECUTE .
VALUE LABELS hgb_cut
1 'Under lower limit (13g/dl men, 12 g/dl women)' 2 'Above lower limit'.
EXECUTE .
```

HCT_CUT (D) Haematocrit under clinical reference points (gender specific)

1 Under lower limit (0.4 l/l men, 0.36 l/l women)
2 Above lower limit

```
SPSS Syntax

DO IF (sexp = 1) .
RECODE hct
(MISSING=SYSMIS) (Lowest thru 0.3999=1) (0.4 thru Highest=2) INTO hct_cut .
END IF .
VARIABLE LABELS hct_cut '(D) Haematocrit under clinical reference points (gender
specific)'.
EXECUTE .
DO IF (sexp = 2) .
RECODE hct
(MISSING=SYSMIS) (Lowest thru 0.3599=1) (0.36 thru Highest=2) INTO hct_cut .
END IF .
VARIABLE LABELS hct_cut '(D) Haematocrit under clinical reference points (gender
specific)'.
EXECUTE .
VALUE LABELS hct_cut
1 'Under lower limit (0.4 l/l men, 0.36 l/l women)' 2 'Above lower limit'.
EXECUTE .
```

AGESEX3 (D) Sex and adult child status combined

1 'Adult men'
2 'Adult women'
3 'Boys'
4 'Girls'

```
SPSS Syntax

DO IF (sexp = 1) .
RECODE agegr2 (1=1) (2=3) INTO agesex3 .
END IF .
VARIABLE LABELS agesex3 'Sex and adult child status combined'.
```

```
EXECUTE .


DO IF (sexp = 2) .
RECODE agegr2 (1=2) (2=4) INTO agesex3 .
END IF .
VARIABLE LABELS agesex3 'Sex and adult child status combined'.
EXECUTE .
VALUE LABELS agesex3
1 'Adult men' 2 'Adult women' 3 'Boys'
4 'Girls' .
EXECUTE .
```

RTIN_CU2 (D) Age and sex specific cut-points for ferritin

1 Under 15 mg/l (women and children)/20 mg/l (men)
2 At or above limit

```
SPSS Syntax

RECODE rtin
(MISSING=SYSMIS) (Lowest thru 14.99=1) (15.00 thru Highest=2) INTO
rtin_cu2 .
VARIABLE LABELS rtin_cu2 '(D) Age and sex specific cut-points for ferritin'.
EXECUTE .
DO IF (agesex3 = 1) .
RECODE rtin
(MISSING=SYSMIS) (Lowest thru 19.99=1) (20.0 thru Highest=2) INTO
rtin_cu2 .
END IF .
VARIABLE LABELS rtin_cu2 '(D) Age and sex specific cut-points for ferritin'.
EXECUTE .
VALUE LABELS rtin_cu2
1 'Under 15 mg/l (women and children)/20 mg/l (men)' 2 'At or above limit'.
EXECUTE .
```

TRF_CUT (D) Transferrin receptor above 3.05 mg/l

1 At or under 3.05 mg/l
2 Above 3.05 mg/l

```
SPSS Syntax

RECODE strfr
(MISSING=SYSMIS) (Lowest thru 3.05=1) (3.050001 thru Highest=2) INTO trf_cut .
VARIABLE LABELS trf_cut '(D) Transferrin receptor above 3.05 mg/l'.
EXECUTE .
VALUE LABELS trf_cut
1 'At or under 3.05 mg/l' 2 'Above 3.05 mg/l'.
EXECUTE .
```

RDW_CUT (D) Upper cut-point for red cell distribution width

1 At or below 14%
2 Above 14%

```
SPSS Syntax

RECODE rdw
(MISSING=SYSMIS) (Lowest thru 14.000=1) (14.0001 thru Highest=2) INTO
rdw_cut .
VARIABLE LABELS rdw_cut '(D) Upper cut-point for red cell distribution width'.
EXECUTE .
VALUE LABELS rdw_cut
1 'At or below 14%' 2 'Above 14%'.
EXECUTE .
```

PFOL_CUT (D) Plasma folate under 7 nmol/l

1 Under 7 nmol/liter
2 At or above 7 nmol/l

```
SPSS Syntax

RECODE
plasmfol
(MISSING=SYSMIS) (Lowest thru 6.99=1) (7 thru Highest=2) INTO pfol_cut .
VARIABLE LABELS pfol_cut '(D) Plasma folate under 7 nmol/l'.
EXECUTE .
VALUE LABELS pfol_cut
1 'Under 7 nmol/liter' 2 'At or above 7 nmol/l'.
EXECUTE .
```

EFOL_CUT (D) Plasma corrected red cell folate under 350 nmol/l

1 Under 350 nmol/liter
2 At or above 350 nmol/l.

```
SPSS Syntax

RECODE plasmcor
(MISSING=SYSMIS) (Lowest thru 349.9=1) (350 thru Highest=2) INTO efol_cut .
VARIABLE LABELS efol_cut '(D) Plasma corrected red cell folate under 350 nmol/l'.
EXECUTE .
VALUE LABELS efol_cut
1 'Under 350 nmol/liter' 2 'At or above 350 nmol/l'.
EXECUTE .
```

B12_CUT (D) Vitamin B12 under 118 pmol/l

1 Under 118 pmol/l
2 At or above 118 pmol/l

```
SPSS Syntax

RECODE b12
(MISSING=SYSMIS) (Lowest thru 117.99=1) (118 thru Highest=2) INTO b12_cut .
VARIABLE LABELS b12_cut '(D) Vitamin B12 under 118 pmol/l'.
EXECUTE .
VALUE LABELS b12_cut
1 'Under 118 pmol/l' 2 'At or above 118 pmol/l'.
EXECUTE .
```

VITC_CUT (D) Vitamin C under clinical reference points (11 and 28 umol/l)

1 <11umol/l: indicative of deficiency
2 >=11-28.99umol/l: Indicative of depletion
3 >28umol/l: normal

```
SPSS Syntax

RECODE vitc
(MISSING=SYSMIS) (Lowest thru 10.99=1) (11 thru 28.999=2) (29.000 thru
Highest=3) INTO vitc_cut .
VARIABLE LABELS vitc_cut '(D) Vitamin C under clinical reference points (11 and 28
umol/l)'.
EXECUTE .
VALUE LABELS vitc_cut
1 '<11umol/l: indicative of deficiency' 2 '>=11-28.99umol/l: Indicative of depletion'
3 '>28umol/l: normal'.
EXECUTE .
```

HCY_CUT (D) Plasma tHcy over clinical reference points (10 and12 umol/l)

1 <=10umol/l: Optimal
2 10-12.00umol/l: Normal
3 >12umol/l: Above normal

```
SPSS Syntax

RECODE plasmthc
(MISSING=SYSMIS) (Lowest thru 10.00=1) (10.001 thru 12.00=2) (12.0001 thru
Highest=3) INTO hcy_cut .
VARIABLE LABELS hcy_cut '(D) Plasma tHcy over clinical reference points (10 and 12
umol/l)'.
EXECUTE .
VALUE LABELS hcy_cut
1 '<=10umol/l: Optimal' 2 '10-12.00umol/l: Normal' 3 '>12umol/l: Above normal'.
EXECUTE .
```

RETI_CUT (D) Retinol under 0.70 umol/l

1 Under 0.70 umol/l
2 At or above 0.70 umol/l

```
SPSS Syntax

RECODE retinol
(MISSING=SYSMIS) (Lowest thru 0.699=1) (0.700 thru Highest=2) INTO reti_cut .
VARIABLE LABELS reti_cut '(D) Retinol under 0.70 umol/l'.
EXECUTE .
VALUE LABELS reti_cut
1 'Under 0.70 umol/l' 2 'At or above 0.70 umol/l'.
EXECUTE .
```

ATOC_CUT (D) a-tocopherol under 11.6 umol/l

1 Under 11.6 umol/l
2 At or above 11.6 umol/l

```
SPSS Syntax

RECODE atocop
(MISSING=SYSMIS) (Lowest thru 11.599=1) (11.600 thru Highest=2) INTO atoc_cut.
VARIABLE LABELS atoc_cut '(D) a-tocopherol under 11.6 umol/l'.
EXECUTE .
VALUE LABELS atoc_cut
1 'Under 11.6 umol/l' 2 'At or above 11.6 umol/l'.
EXECUTE .
```

CTOC_CUT (D) Corrected a-tocopherol under 2.25 umol/l

1 Under 2.25 umol/l
2 At or above 2.25 umol/l

```
SPSS Syntax

RECODE corrtoco
(MISSING=SYSMIS) (Lowest thru 2.2499=1) (2.2500 thru Highest=2) INTO ctoc_cut.
VARIABLE LABELS ctoc_cut '(D) Corrected a-tocopherol under 2.25 umol/l'.
EXECUTE .
VALUE LABELS ctoc_cut
1 'Under 2.25 umol/l' 2 'At or above 2.25 umol/l'.
EXECUTE .
```

VITD_CUT (D) Vitamin D under 25 nmol/l

1 Under 25 nmol/l
2 At or above 25 nmol/l

```
SPSS Syntax

RECODE totohd
(MISSING=SYSMIS) (Lowest thru 24.999=1) (25 thru Highest=2) INTO vitd_cut .
VARIABLE LABELS vitd_cut '(D) Vitamin D under 25 nmol/l'.
EXECUTE .
VALUE LABELS vitd_cut
1 'Under 25 nmol/l' 2 'At or above 25 nmol/l'.
EXECUTE .
```

CHOL_CUT (D) Total cholesterol over 5 mmol/l

1 Under 5 mmol/liter
2 At or above 5 mmol/l

```
SPSS Syntax

RECODE chol
(MISSING=SYSMIS) (Lowest thru 4.99=1) (5 thru Highest=2) INTO chol_cut .
VARIABLE LABELS chol_cut '(D) Total cholesterol over 5 mmol/l'.
EXECUTE .
VALUE LABELS chol_cut
1 'Under 5 mmol/liter' 2 'At or above 5 mmol/l'.
EXECUTE .
```

HDL_CUT (D) HDL under 1 mmol/l

1 Under 1 mmol/liter
2 At or above 1 mmol/l

```
SPSS Syntax

RECODE hdl
(MISSING=SYSMIS) (Lowest thru 0.99=1) (1 thru Highest=2) INTO hdl_cut .
VARIABLE LABELS hdl_cut '(D) HDL under 1 mmol/l'.
EXECUTE .
VALUE LABELS hdl_cut
1 'Under 1 mmol/liter' 2 'At or above 1 mmol/l'.
EXECUTE .
```

NON_HDL (D) Non-HDL cholesterol, proxy for LDL

```
SPSS Syntax

*Calculate variable for proxy LDL, see method description for more explainations.
USE ALL.
COMPUTE non_HDL = chol - hdl .
EXECUTE .
VARIABLE LABELS non_HDL '(D) Non-HDL cholesterol, proxy for LDL' .
EXECUTE .
```

NHDL_CUT (D) Non-HDL cholesterol over 3 mmol/l

1 Under 3 mmol/l
2 At or above 3 mmol/l

```
SPSS Syntax

RECODE non_HDL
```

```
(MISSING=SYSMIS) (Lowest thru 2.9999999=1) (3.0000000 thru Highest=2) INTO nhdl_cut .
VARIABLE LABELS nhdl_cut '(D) Non-HDL cholesterol over 3 mmol/l'.
EXECUTE .
VALUE LABELS nhdl_cut
1 'Under 3 mmol/l' 2 'At or above 3 mmol/l'.
EXECUTE .
```

TRIG_CUT (D) Triglycerids over 1.6 mmol/l

1 Under1.6 mmol/l
2 At or above 1.6 mmol/l

```
SPSS Syntax

RECODE trig
(MISSING=SYSMIS) (Lowest thru 1.599=1) (1.6 thru Highest=2) INTO trig_cut .
VARIABLE LABELS trig_cut '(D) Triglycerids over 1.6 mmol/l'.
EXECUTE .
VALUE LABELS trig_cut
1 'Under1.6 mmol/l' 2 'At or above 1.6 mmol/l'.
EXECUTE .
```

CRP_CUT (D) CRP over gender specific upper quintile from HSE

1 Under upper quintile
2 Above upper quintile (3.7 for men, 4.9 for women)

```
SPSS Syntax

DO IF (sexp = 1) .
RECODE hscrp
(MISSING=SYSMIS) (Lowest thru 3.699=1) (3.7 thru Highest=2) INTO crp_cut .
END IF .
VARIABLE LABELS crp_cut '(D) CRP over gender specific upper quintile from HSE'.
EXECUTE .
DO IF (sexp = 2) .
RECODE hscrp
(MISSING=SYSMIS) (Lowest thru 4.899=1) (4.9 thru Highest=2) INTO crp_cut .
END IF .
VARIABLE LABELS crp_cut '(D) CRP over gender specific upper quintile from HSE'.
EXECUTE .
VALUE LABELS crp_cut
1 'Under upper quintile' 2 'Above upper quintile (3.7 for men, 4.9 for women)'.
EXECUTE .
```

SEASON (D) Season for nurse visit

1 January-March
2 April-June
3 July-September
4 October-December

```
RECODE nurmonth
(MISSING=SYSMIS) (1 thru 3=1) (4 thru 6=2) (7 thru 9=3) (10 thru 12=4)
INTO season .
VARIABLE LABELS season '(D) Season for nurse visit'.
EXECUTE .
VALUE LABELS season
1 'January-March' 2 'April-June' 3 'July-September' 4 'October-December'.
EXECUTE .
```

Chapter 15: Physical Activity

DIS_FIL1 (D) LLSI Filter Variable

1 Not Limited by Long-standing Illness-Include'
2 Limited by Limiting Illness-Exclude

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

COMPUTE dis_fil1=1.
IF wlk5int=3 dis_fil1=2.
IF (Limit =1  AND wlk5int=2) dis_fil1=2.
VARIABLE LABEL dis_fil1 '(D) LLSI filter variable'.
VALUE LABEL dis_fil1 1'Not Limited by Long-standing Illness-Include' 2'Limited by
Limiting Illness-Exclude'.
```

ADHSE30b (D) Adults:Days/4weeks heavy housework 30 mins +
ADHS30bg (D) Adults:Days/4weeks heavy housework 30 mins + (grouped)

0 None
1 1 to 3 days
2 4 to 11 days
3 12 to 19 days
4 20 days or more

ANYHSE (D) Adults: Any heavy housework 30mins +

0 None
1 Some

```
SPSS Syntax


GET FILE='C:\Individual.sav'.

missing values all ().
compute adhse30b=hvhw30.
recode hevyhwrk(2=0)(-9,-8=copy) into adhse30b.


recode housewrk(2=0)(-9,-8=copy)(else=copy) into adhse30b.


Recode adhse30b(0=0)(1 thru 3 =1)(4 thru 11=2)(12 thru 19=3)(20 thru hi=4)(else=copy)
INTO adhs30bg.
variable label adhse30b '(D) Adults:Days/4weeks heavy housework 30 mins +'.
variable label adhs30bg '(D) Adults:Days/4weeks heavy housework 30 mins+ (grouped)'.
value labels adhs30bg
 0 'None'
 1 '1 to 3 days'
 2 '4 to 11 days'
 3 '12 to 19 days'
 4 '20 days or more'.
Recode adhse30b(0=0)(1 thru hi=1)(else=copy) into anyhse.
Variable label anyhse "(D) Adults: Any heavy housework 30mins+".
Value labels anyhse
 0 'None'
 1 'Some'
```

ADMAN30B (D) Adults:Days/4weeks heavy manual 30 mins +
ADMN30BG (D) Adults:Days/4weeks heavy manual 30 mins +(grouped)

```
 0 None
 1 1 to 3 days
 2 4 to 11 days
 3 12 to 19 days
 4 20 days or more
```

ANYMAN (D) Adults: Any heavy manual 30mins+

```
 0 None
 1 Some
```

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

Compute adman30b=diy30.
recode manwork(2=0)(-9,-8=copy)(else=copy) into adman30b.
recode garden(2=0)(-9,-8=copy) into adman30b.
variable label adman30b '(D) Adults:Days/4weeks heavy manual 30 mins +'.
Recode adman30b(0=0)(1 thru 3 =1)(4 thru 11=2)(12 thru 19=3)(20 thru hi=4)(else=copy)
  INTO admn30bg.
variable label admn30bg '(D) Adults:Days/4weeks heavy manual 30 mins +(grouped)'.
value labels admn30bg
 0 'None'
 1 '1 to 3 days'
 2 '4 to 11 days'
 3 '12 to 19 days'
 4 '20 days or more'.
Recode adman30b(0=0)(1 thru hi=1)(else=copy) into anyman.
Variable label anyman "(D) Adults: Any heavy manual 30mins+".
Value labels anyman
 0 'None'
 1 'Some'.
```

HOMEACTY (D) Adults: Housework/gardening activity level

```
 1 Inactive
 2 Light (some non-heavy gardening no heavy housework)
 3 Moderate (heavy housework and/or gardening)
```

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

compute homeacty=0.
IF housewrk=2 & garden=2  homeacty=1.
IF housewrk=2 & manwork<>1 & gardlist<>1 homeacty=1.
IF hevyhwrk=2 & garden=2  homeacty=1.
IF hevyhwrk=2 & manwork <>1 & gardlist <>1 homeacty=1.
IF gardlist=1 & ((manwork <>1) & (hevyhwrk <>1)) homeacty=2.
IF hevyhwrk=1 OR manwork=1 homeacty=3.
IF any(-9,housewrk, hevyhwrk, garden, gardlist, manwork) homeacty=-9.
IF any(-8,housewrk, hevyhwrk, garden, gardlist, manwork) homeacty=-8.
IF hevyhwrk=1 homeacty=3.
IF  agep <16 homeacty=-1.
variable label homeacty '(D) Adults: Housework/gardening activity level'.
value labels homeacty
 1 'Inactive'
 2 'Light (some non-heavy gardening no heavy hwrk)'
 3 'Moderate (heavy hwrk and/or gardening)'.
```

WLKACTYB (D) Adults: Walking activity level

1 Inactive
2 Light (30 mins + at slow/steady pace)
3 Moderate (30 mins+ at brisk/fast pace)

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

compute wlkactyb=0.
IF wlk30M=1 & (walkpace=1 OR walkpace=2 OR walkpace=5) wlkactyb=2.
IF wlk30M=1 & (walkpace=3 OR walkpace=4) wlkactyb=3.
IF wlk30M=2 OR wlk5int=2 OR wlk5int=3 wlkactyb=1.
IF wlk5int=-8 OR wlk5int=-9 OR wlk30m=-8 OR wlk30m=-9 OR walkpace=-9 wlkactyb=-8.
IF wlk5int=-1 wlkactyb=-1.
IF  agep <16 wlkactyb=-1.
variable label wlkactyb '(D) Adults: Walking activity level'.
value labels wlkactyb
 1 'Inactive'
 2 'Light (30 mins + at slow/steady pace)'
 3 'Moderate (30 mins+ at brisk/fast pace)'.
```

ADWLK30B (D) Adults: Days/4 weeks walking fast or brisk 30 mins +
ADWK30BG (D) Adults: Days/4 weeks walking fast or brisk 30 mins+ (grouped)

0 None
1 1 to 3 days
2 4 to 11 days
3 12 to 19 days
4 20 days or more

ANYWLK (D) Adults: Any heavy walking 30mins+

0 None
1 Some

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

Compute adwlk30b=dwlk30.
recode wlk30m(2=0)(-9,-8=copy) into adwlk30b.
recode wlk5int(2,3=0)(-9,-8=copy) into adwlk30b.
recode walkpace(1,2,5=0)(-9,-8=copy) into adwlk30b.
variable label adwlk30b '(D) Adults:Days/4 weeks walking fast or brisk 30 mins +'.
Recode adwlk30b (0=0)(1 thru 3 =1)(4 thru 11=2)(12 thru 19=3)(20 thru hi=4)(else=copy)
 INTO adwk30bg.
variable label adwk30bg '(D) Adults:Days/4 weeks walking fast or brisk 30 mins+
(grouped)'.
value labels adwk30bg
 0 'None'
 1 '1 to 3 days'
 2 '4 to 11 days'
 3 '12 to 19 days'
 4 '20 days or more'.
Recode adwlk30b(0=0)(1 thru hi=1)(else=copy) into anywlk.
Variable label anywlk "(D) Adults: Any heavy walking 30mins+".
Value labels anywlk
 0 'None'
 1 'Some'.
```

ACT11 (D) Other sports intensity
ACT12 (D) Other sports intensity
ACT13 (D) Other sports intensity
ACT14 (D) Other sports intensity
ACT15 (D) Other sports intensity
ACT16 (D) Other sports intensity

```
value labels act11-16
 1 Light type
 2 Moderate type
 3 Vigorous type
 4 Vigorous type (swim, cycle, weights, aerobic, football, tennis)
 5 Very vigorous type (running, squash)
```

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

VAL LAB cothac11 cothac12 cothac13 cothac14 cothac15 cothac16
  1 'Swimming'
  2 'Cycling'
  3 'Workout/Exercise Bike/Weight training'
  4 'Aerobics/Keep Fit/Gymnastics'
  5 'Dancing'
  6 'Running/Jogging'
  7 'Football/Rugby'
  8 'Tennis/Badminton'
  9 'Squash'
  10 'Exercise'
  11 'Abseiling/ Paraseiling'
  12 'Adventure playground'
  13 'Aquarobics'
  14 'American football'
  15 'Archery'
  16 'Assault course'
  17 'Back packing'
  18 'Baseball/softball'
  19 'Basketball'
  20 'Battle re-enactment'
  21 'Bowls - indoor, outdoor, crown, green, Petanque'
  22 'Boxing'
  23 'Canal cruising (if responsible for working locks)'
  24 'Canoeing'
  25 'Circuit training'
  26 'Climbing'
  27 'Cricket'
  28 'Croquet'
  29 'Curling'
  30 'Darts'
  31 'Diving'
  32 'Dog training'
  33 'Drumming (in a group)'
  34 'Fell walking'
  35 'Fencing'
  36 'Field athletics'
  37 'Fishing/ Fly fishing'
  38 'Fives'
  39 'Golf'
  41 'Hang gliding/parachuting'
  42 'Hiking'
  43 'Hitting punch sack'
  44 'Hockey'
  45 'Horse riding'
  46 'Ice skating/ Ice dancing'
  47 'Juggling'
  48 'Kabadi'
  49 'Kick boxing'
  50 'Lacrosse'
  51 'Marathon running'
  52 'Martial arts (Karate, Tai Chi etc.)'
```

```
  53 'Motor sports (Motor-cross, go-karting, jet-skiing)'
  54 'Netball/handball'
  55 'Orienteering'
  56 'Polo'
  57 'Post natal exercise'
  58 'Power boat'
  59 'Racketball'
  60 'Rambling'
  61 'Riding'
  62 'Roller skating'
  63 'Rounders'
  64 'Rowing (inc machine)'
  65 'Sailing (inc dingy)'
  66 'Scuba/subaqua diving'
  67 'Shooting'
  68 'Skateboarding'
  69 'Skiing/dry slope skiing/snowboarding'
  70 'Skipping'
  71 'Skirmishing (war games)'
  72 'Skittles'
  73 'Snooker'
  74 'Snorkelling'
  75 'Sumo wrestling'
  76 'Surfing'
  77 'Swing ball'
  78 'Table tennis'
  79 'Tenpin bowling'
  80 'Territorial Army'
  81 'Toning table/bed'
  82 'Trampolining'
  83 'Volley ball'
  84 'Walking on a jogging machine/treadmill'
  85 'Water skiing'
  86 'Weight lifting'
  87 'Wind surfing'
  88 'Wrestling'
  89 'Yoga'
  90 'Other light exercise'
  91 'Other moderate exercise'
  92 'Other vigorous exercise'
  98 'Other - Unknown energy level'
  99 'Irrelevant answer'.


* Recode 'other' sports into intensity scales.
Recode cothac11 (11,12,15,18,21,23,27,28,29,30,31,32,34,37,39,47,48,
53,57,58,60,61,63,65,66,67,68,72,73,74,75,77,78,79,81,84,89,90,33,45,5=1)
(13,16,17,19,20,24,25,35,36,38,41,42,43,44,46,
50,51,54,55,56,59,62,64,69,71,76,80,83,85,87,91,98,4,8,3,6=2)
(14,22,40,49,52,70,82,86,88,92,93,26=3)
(10=2) (1,2,7=4) (9=5)  (ELSE=COPY) INTO act11.

Recode cothac12 (11,12,15,18,21,23,27,28,29,30,31,32,34,37,39,47,48,
53,57,58,60,61,63,65,66,67,68,72,73,74,75,77,78,79,81,84,89,90,33,45,5=1)
(13,16,17,19,20,24,25,35,36,38,41,42,43,44,46,
50,51,54,55,56,59,62,64,69,71,76,80,83,85,87,91,98,4,8,3,6=2)
(14,22,40,49,52,70,82,86,88,92,93,26=3)
(10=2) (1,2,7=4) (9=5)  (ELSE=COPY) INTO act12.

Recode cothac13 (11,12,15,18,21,23,27,28,29,30,31,32,34,37,39,47,48,
53,57,58,60,61,63,65,66,67,68,72,73,74,75,77,78,79,81,84,89,90,33,45,5=1)
(13,16,17,19,20,24,25,35,36,38,41,42,43,44,46,
50,51,54,55,56,59,62,64,69,71,76,80,83,85,87,91,98,4,8,3,6=2)
(14,22,40,49,52,70,82,86,88,92,93,26=3)
(10=2) (1,2,7=4) (9=5)  (ELSE=COPY) INTO act13.

Recode cothac14 (11,12,15,18,21,23,27,28,29,30,31,32,34,37,39,47,48,
53,57,58,60,61,63,65,66,67,68,72,73,74,75,77,78,79,81,84,89,90,33,45,5=1)
(13,16,17,19,20,24,25,35,36,38,41,42,43,44,46,
50,51,54,55,56,59,62,64,69,71,76,80,83,85,87,91,98,4,8,3,6=2)
(14,22,40,49,52,70,82,86,88,92,93,26=3)
(10=2) (1,2,7=4) (9=5)  (ELSE=COPY) INTO act14.

Recode cothac15 (11,12,15,18,21,23,27,28,29,30,31,32,34,37,39,47,48,
53,57,58,60,61,63,65,66,67,68,72,73,74,75,77,78,79,81,84,89,90,33,45,5=1)
```

```
(13,16,17,19,20,24,25,35,36,38,41,42,43,44,46,
50,51,54,55,56,59,62,64,69,71,76,80,83,85,87,91,98,4,8,3,6=2)
(14,22,40,49,52,70,82,86,88,92,93,26=3)
(10=2) (1,2,7=4) (9=5)  (ELSE=COPY) INTO act15.

Recode cothac16 (11,12,15,18,21,23,27,28,29,30,31,32,34,37,39,47,48,
53,57,58,60,61,63,65,66,67,68,72,73,74,75,77,78,79,81,84,89,90,33,45,5=1)
(13,16,17,19,20,24,25,35,36,38,41,42,43,44,46,
50,51,54,55,56,59,62,64,69,71,76,80,83,85,87,91,98,4,8,3,6=2)
(14,22,40,49,52,70,82,86,88,92,93,26=3)
(10=2) (1,2,7=4) (9=5)  (ELSE=COPY) INTO act16.

variable label act11 '(D) Other sports intensity'.
variable label act12 '(D) Other sports intensity'.
variable label act13 '(D) Other sports intensity'.
variable label act14 '(D) Other sports intensity'.
variable label act15 '(D) Other sports intensity'.
variable label act16 '(D) Other sports intensity'.
value labels act11 act12 act13 act14 act15 act16
 1 'Light type'
 2 'Moderate type'
 3 'Vigorous type'
 4 'Vigorous type (swim,cycle,weights,aerobic,football,tennis)'
 5 'Very vigorous type (running, squash)'.
```

HRSSPT (D) Adults: Average hours doing sport per week
HRSSPTG (D) Adults: Average hours doing sports per week (grouped)

```
 0 No time
 1 Less than 1 hour
 2 1, less than 3 hours
 3 3, less than 5 hours
 4 5, less than 7 hours
 5 7 hours or more
```

```
SPSS Syntax

GET FILE='C:\Individual.sav'.


compute hrsspt=0.

IF (WhtAct01=1 AND range(dayexc,1,28)) hrsspt=hrsspt + ((dayexc*exctim)/240).
IF (WhtAct02=1 AND range(dayexc2,1,28)) hrsspt=hrsspt + ((dayexc2*exctim2)/240).
IF (WhtAct03=1 AND range(dayexc3,1,28)) hrsspt=hrsspt + ((dayexc3*exctim3)/240).
IF (WhtAct04=1 AND range(dayexc4,1,28)) hrsspt=hrsspt + ((dayexc4*exctim4)/240).
IF (WhtAct05=1 AND range(dayexc5,1,28) AND excswt5=1) hrsspt=hrsspt +
((dayexc5*exctim5)/240).
IF (WhtAct06=1 AND range(dayexc6,1,28)) hrsspt=hrsspt + ((dayexc6*exctim6)/240).
IF (WhtAct07=1 AND range(dayexc7,1,28)) hrsspt=hrsspt + ((dayexc7*exctim7)/240).
IF (WhtAct08=1 AND range(dayexc8,1,28)) hrsspt=hrsspt + ((dayexc8*exctim8)/240).
IF (WhtAct09=1 AND range(dayexc9,1,28)) hrsspt=hrsspt + ((dayexc9*exctim9)/240).
IF (WhtAct10=1 AND range(dayexc10,1,28) AND excswt10=1) hrsspt=hrsspt +
((dayexc10*exctim10)/240).
IF (range(act11,2,3) AND range(dayexc11,1,28) AND excswt11=1) hrsspt=hrsspt +
((dayexc11*exctim11)/240).
IF (range(act12,2,3) AND range(dayexc12,1,28) AND excswt12=1) hrsspt=hrsspt +
((dayexc12*exctim12)/240).
IF (range(act13,2,3) AND range(dayexc13,1,28) AND excswt13=1) hrsspt=hrsspt +
((dayexc13*exctim13)/240).
IF (range(act14,2,3) AND range(dayexc14,1,28) AND excswt14=1) hrsspt=hrsspt +
((dayexc14*exctim14)/240).
IF (range(act15,2,3) AND range(dayexc15,1,28) AND excswt15=1) hrsspt=hrsspt +
((dayexc15*exctim15)/240).
IF (range(act16,2,3) AND range(dayexc16,1,28) AND excswt15=1) hrsspt=hrsspt +
((dayexc16*exctim16)/240).
IF any(act11,4,5) AND range(dayexc11,1,28) hrsspt=hrsspt + ((dayexc11*exctim11)/240).
IF any(act12,4,5) AND range(dayexc12,1,28) hrsspt=hrsspt + ((dayexc12*exctim12)/240).
IF any(act13,4,5) AND range(dayexc13,1,28) hrsspt=hrsspt + ((dayexc13*exctim13)/240).
IF any(act14,4,5) AND range(dayexc14,1,28) hrsspt=hrsspt + ((dayexc14*exctim14)/240).
IF any(act15,4,5) AND range(dayexc15,1,28) hrsspt=hrsspt + ((dayexc15*exctim15)/240).
IF any(act16,4,5) AND range(dayexc16,1,28) hrsspt=hrsspt + ((dayexc16*exctim16)/240).
IF  agep <16 hrsspt=-1.
IF RANGE(actphy, -9, -1)  hrsspt=-1.

recode hrsspt (40 thru hi=40).
recode hrsspt (0=0) (7 thru hi=5) (5 thru 7=4) (3 thru 5=3) (1 thru 3=2) (0 thru 1=1)
       (else=copy) INTO hrssptg.
VARIABLE LABEL hrsspt '(D) Adults: Average  hours doing sport per week'.
VARIABLE LABEL hrssptg '(D) Adults: Average  hours doing sports per week (grouped)'.
value labels hrssptg
 0 'No time'
 1 'Less than 1 hour'
 2 '1, less than 3 hours'
 3 '3, less than 5 hours'
 4 '5, less than 7 hours'
 5 '7 hours or more'.
```

SPRTACTY (D) Sport activity level

1 Inactive
2 Light activity
3 Moderate
4 Vigorous

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

Compute sprtacty=1.
IF (WhtAct05=1 & excswt5 <> 1) OR (WhtAct10=1 & excswt10 <> 1) OR
  any(1,act11,act12,act13,act14, act15, act16) sprtacty=2.
IF (WhtAct01=1 & excswt <> 1) OR (Whtact02=1 & excswt2 <> 1) OR
 (WhtAct03=1 & excswt3 <> 1) OR (WhtAct04=1 & excswt4 <> 1) OR
 (WhtAct05=1 & excswt5=1) OR (WhtAct07=1 & excswt7 <> 1) OR
 (WhtAct08=1 & excswt8 <> 1) OR (WhtAct10=1 & excswt10=1) OR
 (Act11=3 & excswt11 <> 1) OR (Act12=3 & excswt12 <> 1) OR
 (Act13=3 & excswt13 <> 1) OR (Act14=3 & excswt14 <> 1) OR
 (Act15=3 & excswt15 <> 1) OR (Act16=3 & excswt16 <> 1) OR
 any(2,act11,act12,act13,act14, act15, act16) sprtacty=3.
IF (WhtAct01=1 & excswt=1) OR (Whtact02=1 & excswt2=1) OR
 (WhtAct03=1 & excswt3=1) OR (WhtAct04=1 & excswt4=1) OR
 WhtAct06=1 OR
 (WhtAct07=1 & excswt7=1) OR (WhtAct08=1 & excswt8=1) OR
 WhtAct09=1 OR
 (any(act11,3,4) & excswt11=1) OR (any(act12,3,4) & excswt12=1) OR
 (any(act13,3,4) & excswt13=1) OR (any(act14,3,4) & excswt14=1) OR
 (any(act15,3,4) & excswt15=1) OR (any(act16,3,4) & excswt16=1) OR
 any(5,act11,act12,act13,act14, act15, act16)
 sprtacty=4.
IF actphy=-8 OR actphy=-9 sprtacty=-8.
IF actphy=-1 sprtacty=-1.
variable label sprtacty '(D) Sport activity level'.
value labels sprtacty
 1 'Inactive'
 2 'Light activity'
 3 'Moderate'
 4 'Vigorous'.
```

VIG20SP (D) Adults: No of days vigorous sports last 4 weeks

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

compute vig20sp=0.
IF whtact01=1 AND range(dayexc,1,28) AND exctim >= 20 AND excswt=1
       vig20sp=vig20sp+dayexc.
IF whtact02=1 AND range(dayexc2,1,28) AND exctim2 >= 20 and excswt2=1
       vig20sp=vig20sp+dayexc2.
IF whtact03=1 AND range(dayexc3,1,28) AND exctim3 >= 20 and excswt3=1
       vig20sp=vig20sp+dayexc3.
IF whtact04=1 AND range(dayexc4,1,28) and exctim4 >= 20 and excswt4=1
       vig20sp=vig20sp+dayexc4.
IF whtact06=1 AND range(dayexc6,1,28) AND exctim6 >= 20
       vig20sp=vig20sp+dayexc6.
IF whtact07=1 AND range(dayexc7,1,28) AND exctim7 >= 20 and excswt7=1
       vig20sp=vig20sp+dayexc7.
IF whtact08=1 and range(dayexc8,1,28) and exctim8 >= 20 and excswt8=1
       vig20sp=vig20sp+dayexc8.
IF whtact09=1 and range(dayexc9,1,28) and exctim9 >= 20
       vig20sp=vig20sp+dayexc9.
IF any(act11,3,4) and range(dayexc11,1,28) and exctim11 >= 20 and excswt11=1
       vig20sp=vig20sp+dayexc11.
IF any(act12,3,4) and range(dayexc12,1,28) and exctim12 >= 20 and excswt12=1
       vig20sp=vig20sp+dayexc12.
IF any(act13,3,4) and range(dayexc13,1,28) and exctim13 >= 20 and excswt13=1
       vig20sp=vig20sp+dayexc13.
IF any(act14,3,4) and range(dayexc14,1,28) and exctim14 >= 20 and excswt14=1
       vig20sp=vig20sp+dayexc14.
IF any(act15,3,4) and range(dayexc15,1,28) and exctim15 >= 20 and excswt15=1
       vig20sp=vig20sp+dayexc15.
IF any(act16,3,4) and range(dayexc16,1,28) and exctim16 >= 20 and excswt16=1
```

```
       vig20sp=vig20sp+dayexc16.
IF act11=5 and range(dayexc11,1,28) and exctim11 >= 20
       vig20sp=vig20sp+dayexc11.
IF act12=5 and range(dayexc12,1,28) and exctim12 >= 20
       vig20sp=vig20sp+dayexc12.
IF act13=5 and range(dayexc13,1,28) and exctim13 >= 20
       vig20sp=vig20sp+dayexc13.
IF act14=5 and range(dayexc14,1,28) and exctim14 >= 20
       vig20sp=vig20sp+dayexc14.
IF act15=5 and range(dayexc15,1,28) and exctim15 >= 20
       vig20sp=vig20sp+dayexc15.
IF act16=5 and range(dayexc16,1,28) and exctim16 >= 20
       vig20sp=vig20sp+dayexc16.
IF any(-8,act11, dayexc11, exctim11, excswt11, act12, dayexc12, exctim12, excswt12,
act13,
       dayexc13, exctim13, excswt13, act14, dayexc14, exctim14, excswt14, act15,
dayexc15,
            exctim15, excswt15, act16, dayexc16, exctim16, excswt16, dayexc, exctim,
       excswt, dayexc2, exctim2, excswt2, dayexc3, exctim3, excswt3,
       dayexc4, exctim4, excswt4, dayexc6, exctim6, dayexc7, exctim7, excswt7,
       dayexc8, exctim8, excswt8, dayexc9, exctim9) vig20sp=-8.
IF any(-9,act11, dayexc11, exctim11, excswt11, act12, dayexc12, exctim12, excswt12,
act13,
       dayexc13, exctim13, excswt13, act14, dayexc14, exctim14, excswt14, act15,
dayexc15,
            exctim15, excswt15, act16, dayexc16, exctim16, excswt16, dayexc, exctim,
       excswt, dayexc2, exctim2, excswt2, dayexc3, exctim3, excswt3,
       dayexc4, exctim4, excswt4, dayexc6, exctim6, dayexc7, exctim7, excswt7,
       dayexc8, exctim8, excswt8, dayexc9, exctim9) vig20sp=-9.
recode vig20sp (28 thru hi=28).
if  agep <16 vig20sp=-1.
variable labels vig20sp "(D) Adults: No of days vigorous sports last 4 weeks".
```

ANYSPT (D) Adults: Any heavy walking 30mins+

0 None
1 Some

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

COMPUTE Adsp30=0.
IF (Whtact01=1 AND range(dayexc,1,28) AND exctim >= 30) Adsp30=adsp30+dayexc.
IF (whtact02=1 AND range(dayexc2,1,28) AND exctim2 >= 30) adsp30=adsp30+dayexc2.
IF (WhtAct03=1 AND range(dayexc3,1,28) AND exctim3 >= 30) adsp30=adsp30+dayexc3.
IF ( WhtAct04=1 AND range(dayexc4,1,28) AND exctim4 >= 30) adsp30=adsp30+dayexc4.
IF (WhtAct05=1 AND range(dayexc5,1,28) AND excswt5=1 AND exctim5 >= 30)
       adsp30=adsp30+dayexc5.
IF (WhtAct06=1 AND range(dayexc6,1,28) AND exctim6 >= 30) adsp30=adsp30+dayexc6.
IF (WhtAct07=1 AND range(dayexc7,1,28) AND exctim7 >= 30) adsp30=adsp30+dayexc7.
IF (WhtAct08=1 AND range(dayexc8,1,28) AND exctim8 >= 30) adsp30=adsp30+dayexc8.
IF (WhtAct09=1 AND range(dayexc9,1,28) AND exctim9 >= 30) adsp30=adsp30+dayexc9.
IF (WhtAct10=1 AND range(dayexc10,1,28) AND exctim10 >= 30 AND excswt10=1)
adsp30=adsp30+dayexc10.
IF range(act11,2,3) and range(dayexc11,1,28) and exctim11 >= 30 and excswt11=1
       adsp30=adsp30+dayexc11.
IF range(act12,2,3) and range(dayexc12,1,28) and exctim12 >= 30 and excswt12=1
       adsp30=adsp30+dayexc12.
IF range(act13,2,3) and range(dayexc13,1,28) and exctim13 >= 30 and excswt13=1
       adsp30=adsp30+dayexc13.
IF range(act14,2,3) and range(dayexc14,1,28) and exctim14 >= 30 and excswt14=1
       adsp30=adsp30+dayexc14.
IF range(act15,2,3) and range(dayexc15,1,28) and exctim15 >= 30 and excswt15=1
       adsp30=adsp30+dayexc15.
IF range(act16,2,3) and range(dayexc16,1,28) and exctim16 >= 30 and excswt16=1
       adsp30=adsp30+dayexc16.
IF range(act11,4,5) and range(dayexc11,1,28) and exctim11 >= 30
       adsp30=adsp30+dayexc11.
IF range(act12,4,5) and range(dayexc12,1,28) and exctim12 >= 30
       adsp30=adsp30+dayexc12.
IF range(act13,4,5) and range(dayexc13,1,28) and exctim13 >= 30
       adsp30=adsp30+dayexc13.
IF range(act14,4,5) and range(dayexc14,1,28) and exctim14 >= 30
       adsp30=adsp30+dayexc14.
```

```
IF range(act15,4,5) and range(dayexc15,1,28) and exctim15 >= 30
       adsp30=adsp30+dayexc15.
IF range(act16,4,5) and range(dayexc16,1,28) and exctim16 >= 30
       adsp30=adsp30+dayexc16.
recode adsp30(28 thru hi=28).
if  agep <16 adsp30=-1.
variable label adsp30 '(D) Adults: Occasions/4 weeks sports 30 mins + moderate +'.
Recode adsp30 (0=0)(1 thru 3 =1)(4 thru 11=2)(12 thru 19=3)(20 thru hi=4)(else=copy)
 INTO adsp30g.
variable label adsp30g '(D) Adults:Days/4 weeks sport 30 mins+ moderate + (grouped)'.
value labels adsp30g
 0 'None'
 1 '1 to 3 days'
 2 '4 to 11 days'
 3 '12 to 19 days'
 4 '20 days or more'.
Recode adsp30(0=0)(1 thru hi=1)(else=copy) into anyspt.
Variable label anyspt "(D) Adults: Any heavy walking 30mins+".
Value labels anyspt
 0 'None'
 1 'Some'.
```

ACTIVWC (D) Reported levels of physical activity at work or at college

1 Not at all active
2 Not very active
3 Fairly active
4 Very active

```
SPSS Syntax

GET FILE='C:\Individual.sav'.


RECODE edact (1=4) (2=3) (3=2) (4=1) (else = copy) into activwc.
RECODE active (1=4) (2=3) (3=2) (4=1)(else = copy) into activwc.
MISSING VALUES activwc (-9 thru -1).
VAR LAB activwc '(D) Reported levels of physical activity at work or at college'.
VAL LAB activwc
1 'Not at all active'
2 'Not very active'
3 'Fairly active'
4 'Very active'.
```

WORKACT (D) Adults: Job activity level

1 Inactive- Light activity
2 Somewhat active
3 Active

WORKACTG (D) Adults: Job activity level (grouped)

1 Not active, light activity, assumed not moderate plus
2 Active, assumed moderate plus

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

compute workact=0.
IF ANY(-8,work,active)|ANY(-9,work,active) workact=-8.
IF (work=2 & Educ=2) OR RANGE(activwc,1,2) workact=1.
IF (work=1 | Educ=1)  AND activwc=3  workact=2.
IF (work=1 | Educ=1)  AND activwc=4  workact=3.
IF  agep <16 workact=-1.
variable label workact '(D) Adults: Job activity level'.
VAL LAB workact
 1 'Inactive- Light activity'
 2 'Somewhat active'
 3 'Active'.
RECODE workact (1,2=1) (3=2) (else=copy) INTO workactg.
variable label workactg '(D) Adults: Job activity level (grouped)'.
value labels workactg
 1 'Not active, light activity, assumed not moderate plus'
 2 'Active, assumed moderate plus'.
```

WORKD (D) Adults: Occupational activity - days in 4 weeks

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

compute workd=0.
IF workactg=2 and (eactp=1 & (RegHrs > 19 | Cashrs >19)) workd=20.
IF workactg=2 and ((eactp=1 OR eactp=2) & (RegHrs < 20 | Cashrs <20) ) workd=12.
IF workactg=1 workd=0.
IF workactg=-8 workd=-8.
RECODE workd (-2=-8).
if  agep <16 workd=-1.
VAR LAB workd '(D) Adults: Occupational activity - days in 4 weeks'.
MISS VAL workd (-99 thru -1).
```

ADTOT30 (D) Adults: Total days/4 weeks active 30 mins + moderate +
ADTOT30C (D) Adults: Total days per week active 30 mins + moderate +

```
 0 None
 1 Less than 1
 2 1 or 2 a week
 3 3 or 4 a week
 4 5 or more a week
```

ADT30ANY (D) Adults: Any activity 30 mins + moderate+

```
 0 None
 1 Any
```

ADT30GP (D) New summary activity level

```
 1 Group 1 -low
 2 Group 2 - medium
 3 Group 3 - high
```

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

COMPUTE adtot30=0.
IF range(adsp30,1,28) adtot30=adtot30+adsp30.
IF range(adwlk30b,1,28) adtot30=adtot30+adwlk30b.
IF range(adman30b,1,28) adtot30=adtot30+adman30b.
IF range(adhse30b,1,28) adtot30=adtot30+adhse30b.
IF workactg=2 AND (eactp=1 & (RegHrs > 19 | Cashrs >19))  adtot30=adtot30+20.
IF workactg=2 AND ((eactp=1 OR eactp=2) & (RegHrs < 20 | Cashrs <20) )
adtot30=adtot30+12.
recode adtot30 (28 thru hi=28).
IF any(-8, adsp30, adwlk30b, adman30b, adhse30b, workactg) adtot30=-8.
IF any(-9, adsp30, adwlk30b, adman30b, adhse30b, workactg) adtot30=-9.
if  agep <16 adtot30=-1.
variable label adtot30 '(D) Adults: Total days/4 weeks active 30 mins + moderate +'.
recode adtot30  (1 thru 3=1) (4 thru 11=2) (12 thru 19=3) (20 thru hi=4)
 (else=copy) INTO adtot30c.
variable label adtot30c '(D) Adults: Total days per week active 30 mins + moderate +'.
value labels adtot30c
 0 'None'
 1 'Less than 1'
 2 '1 or 2 a week'
 3 '3 or 4 a week'
 4 '5 or more a week'.
recode adtot30c (1,2,3,4=1) (else=copy) INTO adt30any.
variable label adt30any '(D) Adults: Any activity 30 mins + moderate+'.
value labels adt30any
 0 'None'
 1 'Any'.
recode adtot30c (0,1=1) (2,3=2) (4=3) (else=copy) INTO adt30gp.
variable label adt30gp '(D) New summary activity level'.
value labels adt30gp
 1 'Group 1 -low'
 2 'Group 2 - medium'
 3 'Group 3 - high'.
```

**********CHILDREN**********

CH00TIM (D) Children: Time last week total activities - no lower limit

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

COMPUTE dis_fil1=1.
IF wlk5int=3 dis_fil1=2.
IF (Limit =1  AND wlk5int=2) dis_fil1=2.
IF RANGE(LIMIT, -99,  -1) dis_fil2=-1.
VAR LAB dis_fil1 'LLSI filter variable'.
```

```
VAL LAB dis_fil1 1'Not Limited by Long-standing Illness-Include' 2'Limited by Limiting
Illness-Exclude'.

RECODE wespor (1=2.5) (2=10) (3=22.5) (4=45) (5=75) (6=105) (7=135) (8=165) (9=195)
 (10=225) (11=240) (ELSE=0) INTO xxwesp.
RECODE wkspor (1=2.5) (2=10) (3=22.5) (4=45) (5=75) (6=105) (7=135) (8=165) (9=195)
 (10=225) (11=240) (ELSE=0) INTO xxwksp.
RECODE weact (1=2.5) (2=10) (3=22.5) (4=45) (5=75) (6=105) (7=135) (8=165) (9=195)
 (10=225) (11=240) (ELSE=0) INTO xxweac.
RECODE wkact (1=2.5) (2=10) (3=22.5) (4=45) (5=75) (6=105) (7=135) (8=165) (9=195)
 (10=225) (11=240) (ELSE=0) INTO xxwkac.
RECODE daywlkt (2=10) (3=22.5) (4=45) (5=75) (6=105) (7=135) (8=165) (9=195)
 (10=225) (11=240) (ELSE=0) INTO xxwlk.
RECODE thwk (3=22.5) (4=45) (5=75) (6=105) (7=135) (8=165) (9=195)
 (10=225) (11=240) (ELSE=0) INTO xxhwk.


compute ch00tim =0.
IF (range(dayswlk,1,7)) AND daywlkt ge 2 ch00tim=(dayswlk* xxwlk).
IF (range(dhwkch,1,7)) AND thwk ge 3 ch00tim=ch00tim + (dhwkch*xxhwk).

IF (range(dwesp,1,2)) & (range(xxwesp,2.5,240))
   ch00tim=ch00tim + xxwesp.
IF (dwesp=3) & (range(xxwesp,2.5,240))
   ch00tim=ch00tim +  (xxwesp).
IF (range(daysp,1,5)) & (range(xxwksp,2.5,240))
   ch00tim=ch00tim + (daysp* xxwksp).

IF (range(dweact,1,2)) & (range(xxweac,2.5,240))
   ch00tim=ch00tim + xxweac.
IF (dweact=3) & (range(xxweac,2.5,240))
   ch00tim=ch00tim + ( xxweac).
IF (range(wkactdo,1,5)) & (range(xxwkac,2.5,240))
   ch00tim=ch00tim + (wkactdo* xxwkac).

IF agep>=16 ch00tim=-1.

MISS VAL wlk5ch daywlkt hwkch thwk sportdo wespor  wkspor  weactdo weact wkact ().

******************.
IF ANY(-8,wlk5ch,daywlkt,hwkch,thwk, sportdo,wespor, wkspor, weactdo,  weact,
   wkact) |
   ANY(-9,wlk5ch,daywlkt,hwkch,thwk, sportdo,wespor, wkspor, weactdo,  weact,
   wkact) ch00tim=-8.

variable label ch00tim
  '(D) Children: Time last week total activities - no lower limit'.

MISS VAL sportdo wespor wkspor ().

COMPUTE sptime=0.
IF (sportdo=1 & (RANGE(wespor , -9, -1) & RANGE(wkspor, -9, -1))) or range(sportdo,-
9,-1)  sptime=-8.
IF (range(dwesp,1,2)) & (range(xxwesp,2.5,240))  sptime=sptime + (xxwesp).
IF (dwesp=3) & (range(xxwesp,2.5,240)) sptime=sptime +  (xxwesp).
IF range(daysp,1,5) & range(xxwksp,2.5,240) sptime=sptime + (daysp* xxwksp).
IF agep>=16 sptime=-1.

MISS VAL sptime sportdo wespor wkspor (-99 thru -1).

variable label sptime '  Time spent doing sport last week, minutes'.

COMPUTE sptime2=sptime/60.
IF agep>=16 sptime2=-1.

MISS VAL sptime2 (-99 thru -1).

variable label sptime2 '  Time spent doing sport last week, HOURS'.

MISS VAL weactdo weact wkact ().

COMPUTE actime=0.
IF (weactdo=1 & (RANGE(weact, -9, -1) & RANGE(wkact, -9, -1))) or range(weactdo,-9,-1)
actime=-8.
IF (range(dweact,1,2)) & (range(xxweac,2.5,240))
   actime=actime + xxweac.
```

```
IF (dweact=3) & (range(xxweac,2.5,240))
   actime=actime + ( xxweac).
IF (range(wkactdo,1,5)) & (range(xxwkac,2.5,240))
  actime=actime + (wkactdo* xxwkac).
IF agep>=16 actime=-1.

MISS VAL actime weactdo weact wkact (-99 thru -1).

variable label actime '  Time spent doing active playing last week, minutes'.

COMPUTE actime2=actime/60.
IF agep>=16 actime2=-1.

variable label actime2 '  Time spent doing active playing last week, HOURS'.

MISS VAL actime2 (-99 thru -1).

MISS VAL  wlk5ch dayswlk ().

COMPUTE wlktime=0.
IF (wlk5ch=1 & RANGE(dayswlk, -9, -1))  or range(wlk5ch,-9,-1) wlkdays=-8.
IF (range(dayswlk,1,7)) AND daywlkt ge 2 wlktime=(dayswlk* xxwlk).
IF agep>=16 wlktime=-1.

MISS VAL  wlktime (-99 thru -1).

variable label wlktime '  Time spent  walking in last 7 days (minutes)'.


COMPUTE wlktime2=wlktime/60.
IF agep>=16 wlktime2=-1.
variable label wlktime2 '  Time spent  walking in last 7 days (HOURS)'.

MISS VAL  wlktime2 (-99 thru -1).

MISS VAL hwkch dhwkch ().

COMPUTE hwtime=0.
IF (hwkch=1 & RANGE(dhwkch, -9, -1)) or range(hwkch, -9, -1) hwtime=-8.

IF range(dhwkch,1,7) AND thwk ge 3 hwtime=(dhwkch*xxhwk).

IF agep>=16 hwtime=-1.

MISS VAL  hwtime hwkch dhwkch (-99 thru -1).

variable label hwtime '  House work time last 7 days, minutes'.


COMPUTE hwtime2= hwtime/60.
IF agep>=16 hwtime2=-1.
variable label hwtime2 '  House work time last 7 days, HOURS'.

MISS VAL  hwtime2 (-99 thru -1).

RECODE sptime (0=0) (1 thru 59.99=1) (60 thru 179.99=2) (180 thru 299.99=3) (300 thru
419.99=4) (420 thru hi=5) (ELSE=copy) INTO  sptimeg.
VAL LAB sptimeg
0 'No time'
1 'Some, less than 1 hr'
2 '1, less than 3 hrs'
3 '3, less than 5hrs'
4 '5, less than 7hrs'
5 '7 hrs or more'.
VAR LAB sptimeg  '  Time spent on Sports in last 7 days (hours grouped)'.

RECODE actime (0=0) (1 thru 59.99=1) (60 thru 179.99=2) (180 thru 299.99=3) (300 thru
419.99=4) (420 thru hi=5) (ELSE=copy) INTO  actimeg.
VAL LAB  actimeg
0 'No time'
1 'Some, less than 1 hr'
2 '1, less than 3 hrs'
3 '3, less than 5hrs'
4 '5, less than 7hrs'
5 '7 hrs or more'.
VAR LAB actimeg '  Time spent on Active Play in last 7 days (hours grouped)'.
```

```
RECODE wlktime (0=0) (1 thru 59.99=1) (60 thru 179.99=2) (180 thru 299.99=3) (300 thru
419.99=4) (420 thru hi=5) (ELSE=copy) INTO  wlktimeg.
VAL LAB wlktimeg
0 'No time'
1 'Some, less than 1 hr'
2 '1, less than 3 hrs'
3 '3, less than 5hrs'
4 '5, less than 7hrs'
5 '7 hrs or more'.
VAR LAB wlktimeg '  Time spent on Walking in last 7 days (hours grouped)'.

RECODE hwtime (0=0) (1 thru 59.99=1) (60 thru 179.99=2) (180 thru 299.99=3) (300 thru
419.99=4) (420 thru hi=5) (ELSE=copy) INTO  hwtimeg.
VAL LAB hwtimeg
0 'No time'
1 'Some, less than 1 hr'
2 '1, less than 3 hrs'
3 '3, less than 5hrs'
4 '5, less than 7hrs'
5 '7 hrs or more'.
VAR LAB hwtimeg '  Time spent on Housework in last 7 days (hours grouped)'.

RECODE ch00tim (0=0) (1 thru 59.99=1) (60 thru 179.99=2) (180 thru 299.99=3) (300 thru
419.99=4) (420 thru hi=5) (ELSE=copy) INTO ch00timg.
VAL LAB ch00timg
0 'No time'
1 'Some, less than 1 hr'
2 '1, less than 3 hrs'
3 '3, less than 5hrs'
4 '5, less than 7hrs'
5 '7 hrs or more'.
VAR LAB ch00timg '  Time spent on All activties in last 7 days (hours)'.
```

CH00TOT (D) Children: Days last week all activities - no time limits

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

MISS VAL wlk5ch dayswlk hwkch dhwkch sportdo dwesp daysp weact dweact wkact ().

compute ch00tot = 0.
IF (range(dayswlk,1,7)) ch00tot=dayswlk.
IF (range(dhwkch,1,7)) ch00tot = ch00tot + dhwkch.
IF (range(dwesp,1,2)) ch00tot = ch00tot + 1.
IF dwesp=3 ch00tot=ch00tot + 2.
IF (range(daysp,1,5)) ch00tot = ch00tot+daysp.
IF (range(dweact,1,2)) ch00tot=ch00tot+1.
IF dweact=3 ch00tot=ch00tot+2.
IF (range(wkactdo,1,5)) ch00tot=ch00tot+wkactdo.
IF ANY(-8,wlk5ch,dayswlk,hwkch,dhwkch,sportdo,dwesp,daysp,
  weact,dweact,wkact) |
  ANY(-9,wlk5ch,dayswlk,hwkch,dhwkch,sportdo,dwesp,daysp,
  weact,dweact,wkact) ch00tot=-8.

IF agep>=16 ch00tot=-1.
recode ch00tot(7 thru hi=7).
variable label ch00tot
   '(D) Children: Days last week all activities - no time limits'.

MISS VAL ch00tot wlk5ch dayswlk hwkch dhwkch sportdo dwesp daysp weact dweact wkact (-
99 thru -1).

MISS VAL dwesp daysp sportdo ().

COMPUTE sprtdays=0.
IF (sportdo=1 & (range(dwesp,-9 ,-1) & range(daysp,-9, -1))) or range(sportdo,-9,-1)
sprtdays=-8.
IF  (range(dwesp,1,2)) sprtdays= sprtdays + 1.
IF dwesp=3 sprtdays=sprtdays+ 2.
IF (range(daysp,1,5)) sprtdays= sprtdays+daysp.

IF agep>=16 sprtdays=-1.
```

```
VAR LAB sprtdays '  n of days sports/exercise'.

MISS VAL sprtdays sportdo (-99 thru -1).

MISS VAL dweact wkactdo weactdo ().
COMPUTE actdays =0.
IF (weactdo=1 & (range(dweact,-9 ,-1) & range(wkactdo, -9, -1))) or range(weactdo,-9,-
1)  actdays=-8.
IF (range(dweact,1,2)) actdays=actdays+1.
IF dweact=3 actdays=actdays+2.
IF (range(wkactdo,1,5)) actdays=actdays+wkactdo.

IF agep>=16 actdays=-1.
VAR LAB actdays '  n of days active playing'.

MISS VAL actdays weactdo (-99 thru -1).

MISS VAL dayswlk wlk5ch ().

COMPUTE wlkdays =0.
IF (wlk5ch=1 & RANGE(dayswlk, -9, -1))  or range(wlk5ch,-9,-1) wlkdays=-8.
IF (range(dayswlk,1,7)) wlkdays=dayswlk.

IF agep>=16 wlkdays=-1.
VAR LAB wlkdays '  n of days Walking 5mins+'.
MISS VAL wlkdays wlk5ch (-99 thru -1).

MISS VAL dhwkch hwkch ().

COMPUTE gardays=0.
IF (hwkch=1 & RANGE(dhwkch, -9, -1)) or range(hwkch,-9,-1) gardays=-8.
IF (range(dhwkch,1,7)) gardays= gardays + dhwkch.
IF agep>=16 gardays=-1.
VAR LAB gardays '  n of days housework/gardening (15+)'.

MISS VAL gardays hwkch (-99 thru -1).

RECODE sprtdays (0=0) (1 thru 2=1) (3 thru 4=2) (5 thru highest=3) (else=copy) INTO
ch00sptg.
VAR LAB ch00sptg '   Days last week (no lower limit) sports&exercise (grouped)'.
VAL LAB ch00sptg
0 'None'
1 '1-2 days'
2 '3-4 days'
3 '5 or more days'.

RECODE actdays (0=0) (1 thru 2=1) (3 thru 4=2) (5 thru highest=3) (else=copy) INTO
ch00plyg.
VAR LAB ch00plyg '   Days last week (no lower limit) active playing (grouped)'.
VAL LAB ch00plyg
0 'None'
1 '1-2 days'
2 '3-4 days'
3 '5 or more days'.

RECODE wlkdays (0=0) (1 thru 2=1) (3 thru 4=2) (5 thru highest=3) (else=copy) INTO
ch00wlkg.
VAR LAB ch00wlkg '   Days last week (5+) mins walking (grouped)'.
VAL LAB ch00wlkg
0 'None'
1 '1-2 days'
2 '3-4 days'
3 '5 or more days'.

RECODE gardays (0=0) (1 thru 2=1) (3 thru 4=2) (5 thru highest=3) (else=copy) INTO
ch00hswg.
VAR LAB ch00hswg '   Days last week (15+) mins housework/gardening (grouped)'.
VAL LAB ch00hswg
0 'None'
1 '1-2 days'
2 '3-4 days'
3 '5 or more days'.

RECODE ch00tot (1,2=1) (3,4=3) (5 thru 7=5) (else=copy) into ch00totg.
variable label ch00tot
  '(D) Children: Days last week (no lower limit) total activities'.
```

```
variable label ch00totg
  '  Children: Days last week any physical activties (grouped)'.
value labels ch00totg
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.

compute ch15spt=0.
IF agep>=16 | sportdo=-1 ch15spt=-1.
IF RANGE(wespor,3,11) AND ANY(dwesp,1,2) ch15spt=1.
IF RANGE(wespor,4,11) AND dwesp=3 ch15spt=2.
IF RANGE(wkspor,3,11) AND RANGE(daysp,1,5) ch15spt=ch15spt + daysp.
IF ANY(-8,sportdo,wespor,dwesp,wkspor,daysp)
   |ANY(-9,sportdo,wespor,dwesp,wkspor,daysp) ch15spt=-8.
recode ch15spt (1,2=1) (3,4=3) (5,6,7=5) (else=copy) INTO ch15sptg.
variable label ch15spt '  Children: Days last week 15+min sport'.
variable label ch15sptg
  '  Children: Days last week 15+min sport (grouped)'.
value labels ch15sptg
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.

compute ch30spt=0.
IF agep>=16 | sportdo=-1 ch30spt=-1.
IF RANGE(wespor,4,11) AND ANY(dwesp,1,2) ch30spt=1.
IF RANGE(wespor,5,11) AND dwesp=3 ch30spt=2.
IF RANGE(wkspor,4,11) AND RANGE(daysp,1,5) ch30spt=ch30spt + daysp.
IF ANY(-8,sportdo,wespor,dwesp,wkspor,daysp)
   |ANY(-9,sportdo,wespor,dwesp,wkspor,daysp) ch30spt=-8.
recode ch30spt (1,2=1) (3,4=3) (5,6,7=5) (else=copy) INTO ch30sptg.
variable label ch30spt '  Children: Days last week 30+min sport'.
variable label ch30sptg
  '  Children: Days last week 30+min sport (grouped)'.
value labels ch30sptg
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.

compute ch15ply=0.
IF agep>=16 | weact=-1 ch15ply=-1.
IF RANGE(weact,3,11) AND ANY(dweact,1,2) ch15ply=1.
IF RANGE(weact,4,11) AND dweact=3 ch15ply=2.
IF RANGE(wkact,3,11) AND RANGE(wkact,1,5) ch15ply=ch15ply + wkact.
IF ANY(-8,weact,weact,dweact,wkact,wkact)
   |ANY(-9,weact,weact,dweact,wkact,wkact) ch15ply=-8.
recode ch15ply (1,2=1) (3,4=3) (5,6,7=5) (else=copy) INTO ch15plyg.
variable label ch15ply '  Children: Days last week 15+min active play'.
variable label ch15plyg
  '  Children: Days last week 15+min active play (grouped)'.
value labels ch15plyg
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.

compute ch30ply=0.
IF agep>=16 | weact=-1 ch30ply=-1.
IF RANGE(weact,4,11) AND ANY(dweact,1,2) ch30ply=1.
IF RANGE(weact,5,11) AND dweact=3 ch30ply=2.
IF RANGE(wkact,4,11) AND RANGE(wkact,1,5) ch30ply=ch30ply + wkact.
IF ANY(-8,weact,weact,dweact,wkact,wkact)
   |ANY(-9,weact,weact,dweact,wkact,wkact) ch30ply=-8.
recode ch30ply (1,2=1) (3,4=3) (5,6,7=5) (else=copy) INTO ch30plyg.
variable label ch30ply '  Children: Days last week 30+min active play'.
variable label ch30plyg
  '  Children: Days last week 30+min active play (grouped)'.
value labels ch30plyg
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.
```

```
COMPUTE ch15act=0.
IF (RANGE(ch15spt,0,7)) ch15act=ch15spt.
IF (RANGE(ch15ply,0,7)) ch15act=ch15act + ch15ply.
IF ANY(-8,ch15spt,ch15ply) ch15act=-8.
IF ANY(-1,ch15spt,ch15ply) ch15act=-1.
recode ch15act (1,2=1) (3,4=3) (5,6,7,8,9,10,11,12,13,14=5)
  (else=copy) into ch15actg.
variable label ch15act
  '  Children: Days last week 15+min sport+active play'.
variable label ch15actg
  '  Children: Days last week 15+min sport+active play (grouped)'.
value labels ch15actg
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.

COMPUTE ch30act=0.
IF (RANGE(ch30spt,0,7)) ch30act=ch30spt.
IF (RANGE(ch30ply,0,7)) ch30act=ch30act + ch30ply.
IF ANY(-8,ch30spt,ch30ply) ch30act=-8.
IF ANY(-1,ch30spt,ch30ply) ch30act=-1.
recode ch30act (1,2=1) (3,4=3) (5,6,7,8,9,10,11,12,13,14=5)
  (else=copy) into ch30actg.
variable label ch30act
  '  Children: Days last week 30+min sport+active play'.
variable label ch30actg
  '  Children: Days last week 30+min sport+active play (grouped)'.
value labels ch30actg
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.

compute ch15wlk=0.
IF agep>=16 | wlk5ch=-1 ch15wlk=-1.
IF (RANGE(dayswlk,1,7)) & daywlkt>=3 ch15wlk=dayswlk.
IF ANY(-9,wlk5ch, daywlkt, dayswlk) | ANY(-8,wlk5ch, daywlkt, dayswlk)
ch15wlk=-8.
recode ch15wlk (1,2=1) (3,4=3) (5,6,7=5) (else=copy) INTO ch15wlkg.
variable label ch15wlk '  Children: Days last week 15>min walk'.
variable label ch15wlkg
    '  Children: Days last week 15>min walk (grouped)'.
value labels ch15wlkg
0 'None'
1 '1 or 2'
3 '3 or 4'
5 '5 or more'.

compute ch15hwk=0.
IF agep>=16 | hwkch=-1 ch15hwk=-1.
IF (RANGE(dhwkch,1,7)) ch15hwk=dhwkch.
IF ANY(-9,hwkch,dhwkch)|ANY(-8,hwkch,dhwkch) ch15hwk=-8.
recode ch15hwk (1,2=1) (3,4=3) (5,6,7=5) (else=copy) INTO ch15hwkg.
variable label ch15hwk '  Children: Days last week 15+min housewk/gardening'.
variable label ch15hwkg
  '  Children: Days last week 15+min housewk/gardening (grouped)'.
value labels ch15hwkg
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.

compute ch15tot=0.
IF (RANGE(ch15act,0,14)) ch15tot=ch15act.
IF (RANGE(ch15wlk,0,7)) ch15tot=ch15tot + ch15wlk.
IF (RANGE(ch15hwk,0,7)) ch15tot=ch15tot + ch15hwk.
IF ANY(-8,ch15act,ch15wlk,ch15hwk) ch15tot=-8.
IF agep>=16 ch15tot=-1.
recode ch15tot(7 thru hi=7).
recode ch15tot (1,2=1) (3,4=3) (5 thru 7=5) (else=copy) into ch15totg.
variable label ch15tot
  '  Children: Days last week 15+min total activities'.
variable label ch15totg
  '  Children: Days last week 15+min total activities (grouped)'.
value labels ch15totg
```

```
 0 'None'
 1 '1 or 2'
 3 '3 or 4'
 5 '5 or more'.

compute ch15tim=0.
IF (range(dayswlk,1,7)) AND daywlkt ge 3 ch15tim=dayswlk*xxwlk.
IF (range(dhwkch,1,7)) AND thwk ge 3 ch15tim=ch15tim + (dhwkch*xxhwk).
IF (range(dwesp,1,2)) & (range(xxwesp,22.5,240))
   ch15tim=ch15tim + xxwesp.
IF (dwesp=3) & (range(xxwesp,45,240))
   ch15tim=ch15tim + (xxwesp).
IF (range(daysp,1,5)) & (range(xxwksp,22.5,240))
   ch15tim=ch15tim + (daysp* xxwksp).
IF (range(dweact,1,2)) & (range(xxweac,22.5,240))
   ch15tim=ch15tim + xxweac.
IF (dweact=3) & (range(xxweac,45,240))
   ch15tim=ch15tim + (xxweac).
IF (range(wkact,1,5)) & (range(xxwkac,22.5,240))
   ch15tim=ch15tim + (wkact* xxwkac).
IF ANY(-8,wlk5ch,dayswlk,hwkch,dhwkch,sportdo,dwesp,daysp,weact,
   dweact,wkactdo,wkspor,weact,wkact) |
   ANY(-9,wlk5ch,dayswlk,hwkch,dhwkch,sportdo,dwesp,daysp,weact,
   dweact,wkactdo,wkspor,weact,wkact) ch15tim=-8.
IF agep>=16 ch15tim=-1.
variable label ch15tim '  Children: Time last week 15+min total activities'.
```

CH00MPD (D) Children's min/day all activities - no lower limit
CH00MPDG (D) Children's min/day all activities - no lower limit (grouped)

```
 0 No time
 1 1-29 minutes
 2 30 -59 minutes
 3 60 minutes or more
```

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

IF (range(ch15tot,1,7)) ch15mpd = ch15tim/ch15tot.
IF ch15tim=0 ch15mpd=0.
IF agep>=16 ch15mpd=-1.
IF ANY(-8,ch15tim,ch15tot) ch15mpd=-8.
recode ch15mpd (1 thru 29.99=1) (30 thru 59.99=2) (60 thru 119.99=3)
 (120 thru hi=4) (else=copy) INTO ch15mpdg.
variable label ch15mpd '  Children min/day all activities - 15+min'.
variable label ch15mpdg
   '  Children min/day all activities - 15+min (grouped)'.
value labels ch15mpdg
 0 'No time'
 1 '1-29 minutes'
 2 '30-59 minutes'
 3 '60-119 minutes'
 4 '120 minutes+'.


IF ((RANGE(ch15tot,5,7)) & ch15mpdg=4) ch15sum=1.
IF ((RANGE(ch15tot,5,7)) & ch15mpdg=3) ch15sum=2.
IF ((RANGE(ch15tot,5,7)) & ch15mpdg=2) ch15sum=3.
IF ((RANGE(ch15tot,5,7)) & ch15mpdg=1) ch15sum=4.
IF ((RANGE(ch15tot,1,4)) & (RANGE(ch15mpdg,2,4))) ch15sum=5.
IF ((RANGE(ch15tot,0,4)) & (RANGE(ch15mpdg,0,1))) ch15sum=6.
IF agep>=16 ch15sum=-1.
RECODE ch15sum (SYSMIS=-8).
variable label ch15sum
   '  Children: Summary classification 15+min activity levels'.
value labels ch15sum
  1 '120+mins 5+ days/wk'
  2 '60-119 mins 5+ days/wk'
  3 '30-59 mins 5+ days/wk'
  4 '1-29 mins 5+ days/wk'
  5 '30+ mins 1-4 days/wk'
  6 '<30 mins <5 days'.
recode ch15sum (1,2=1) (3=2) (4,5,6=3) (else=copy) INTO ch15sumg.
```

```
variable label ch15sumg
   ' Children: Summary classification 15+min activity levels (grouped)'.
value labels ch15sumg
  1 'Group 1:60+min on at least 5 days'
  2 'Group 2:30-59min on at least 5 days'
  3 'Group 3:Lower level of activity'.


** mpdg.

IF (range(ch00tot,1,7)) ch00mpd = ch00tim/ch00tot.
IF ch00tim=0 ch00mpd=0.
IF agep>=16 ch00mpd=-1.
IF ANY(-8,ch00tim,ch00tot) ch00mpd=-8.
recode ch00mpd (1 thru 29.99=1) (30 thru 59.99=2) (60 thru hi=3)
 (else=copy) INTO ch00mpdg.
variable label ch00mpd '(D) Children's min/day all activities - no lower limit'.
variable label ch00mpdg
   '(D) Children's min/day all activities - no lower limit (grouped)'.
value labels ch00mpdg
 0 'No time'
 1 '1-29 minutes'
 2 '30 -59 minutes'
 3 '60 minutes or more'.
```

CH00SUM7 (D) Children: Summary classification activity levels - All activities, no lower limits (7 days X 60+mins)

```
  1 Group 3:60+min on all 7 days
  2 Group 2:30-59min on all 7 days
  3 Group 1:Lower level of activity
```

```
SPSS Syntax

GET FILE='C:\Individual.sav'.

COMPUTE ch00sum5=-8.
IF ((RANGE(ch00tot,5,7)) & ch00mpdg=3) ch00sum5=1.
IF ((RANGE(ch00tot,5,7)) & ch00mpdg=2) ch00sum5=2.
IF ((RANGE(ch00tot,5,7)) & (RANGE(ch00mpdg, 0,1))) ch00sum5=3.
IF ((RANGE(ch00tot,1,4)) & (RANGE(ch00mpdg, 0,1))) ch00sum5=3.
IF agep>=16 ch00sum5=-1.
MISS VAL ch00sum5 (-8, -1).
variable label ch00sum5
      '  Children: Summary classification activity levels - All activities, no lower
limits (5 days X 60+mins)'.
VAL LAB ch00sumX
       1 'Group 3:60+min on at least 5 days'
       2 'Group 2:30-59min on at least 5 days'
       3 'Group 1:Lower level of activity'.


COMPUTE ch00sum7=-8.
IF (ch00tot=7 & ch00mpdg=3) ch00sum7=1.
IF (ch00tot=7& ch00mpdg=2) ch00sum7=2.
IF ((RANGE(ch00tot,1,7)) & (RANGE(ch00mpdg, 0,1))) ch00sum7=3.
IF ((RANGE(ch00tot,1,6)) & (RANGE(ch00mpdg, 2,3))) ch00sum7=3.
IF agep>=16 ch00sum7=-1.
MISS VAL ch00sum7 (-8, -1).
variable label ch00sum7
   '(D) Children: Summary classification activity levels - All activities, no lower
limits (7 days X 60+mins)'.
VAL LAB ch00sum7
  1 'Group 3:60+min on all 7 days'
  2 'Group 2:30-59min on all 7 days'
  3 'Group 1:Lower level of activity'.

RECODE
  agep
  (Lowest thru 15=1)  (16 thru Highest=2)  INTO  ageactiv .
VARIABLE LABELS ageactiv '  Adult (>=16) / Child (<16) for physical activity'.
EXECUTE .
value labels ageactiv 1'Child' 2'Adult'.


************
ADULTS
************.
*If regular job then number of hours comes from Reghrs.
*If doesn't know number of hours assume 38.5 hours/week.
*If occasional job: If typcas=yes (1) use number of hours from cashrs. If typcas=no
(2) use number of hours from ncashrs.
*If Regcas = 3 (Both regular job AND occasional job) then total work hours is regular
+ occasional.

compute workhr=0.
recode workhr (0=sysmis).

do if work=1 & regcas=1.
recode reghrs (0=0) (3=3) (4=4) (5=5) (6=6) (7=7) (8=8) (9=9) (10=10) (11=11) (12=12)
(14=14) (15=15) (16=16) (17=17) (18=18) (19=19)
(20=20) (21=21) (22=22) (24=24) (25=25) (26=26) (27=27) (28=28) (30=30) (31=31)
(32=32) (35=35) (36=36) (37=37) (38=38) (39=39)
(40=40) (42=42) (43=43) (45=45) (46=46) (48=48) (50=50) (52=52) (56=56) (60=60)
(72=72) (80=80) into workhr.
end if.
```

```
do if work=1 & regcas=2 & typcas=1.
recode cashrs (1=1) (2=2) (4=4) (5=5) (6=6) (8=8) (10=10) (12=12) (14=14) (15=15)
(18=18) (20=20) (30=30) (32=32) (40=40) into workhr.
end if.


do if work=1 & regcas=2 & typcas=2.
recode ncashrs (1=1) (8=8) (10=10) (12=12) (15=15) (16=16) (20=20) (30=30) (32=32)
(35=35) into workhr.
end if.


do if work=1 & regcas=3.
compute workhr =reghrs+cashrs.
end if.


recode workhr (missing = -7).

do if work=1 & workhr=-7.
recode workhr (-7=38.5).
end if.


do if work=1 & regcas=2 & workhr=38.5.
recode workhr (38.5=-7).
end if.

recode workhr (-7=sysmis).

VARIABLE LABEL workhr '(I) Final number of typical hours at work in regular or
occasional job per week'.




compute act1pw = ((dayexc*exctim)/4)/60.
compute act2pw = ((dayexc2*exctim2)/4)/60.
compute act3pw = ((dayexc3*exctim3)/4)/60.
compute act4pw = ((dayexc4*exctim4)/4)/60.
compute act5pw = ((dayexc5*exctim5)/4)/60.
compute act6pw = ((dayexc6*exctim6)/4)/60.
compute act7pw = ((dayexc7*exctim7)/4)/60.
compute act8pw = ((dayexc8*exctim8)/4)/60.
compute act9pw = ((dayexc9*exctim9)/4)/60.
compute act10pw = ((dayexc10*exctim10)/4)/60.
compute act11pw = ((dayexc11*exctim11)/4)/60.
compute act12pw = ((dayexc12*exctim12)/4)/60.
compute act13pw = ((dayexc13*exctim13)/4)/60.

* Exclude those with less than 15mins as should have done this at interview stage.
* Exlcude those with no days specified as can't use these.

DO IF exctim <15 or dayexc =0.
recode act1pw (else=sysmis).
end if.

DO IF exctim2 <15 or dayexc2 =0.
recode act2pw (else=sysmis).
end if.

DO IF exctim3 <15 or dayexc3 =0.
recode act3pw (else=sysmis).
end if.

DO IF exctim4 <15 or dayexc4 =0.
recode act4pw (else=sysmis).
end if.

DO IF exctim5 <15 or dayexc5 =0.
recode act5pw (else=sysmis).
end if.

DO IF exctim6 <15 or dayexc6 =0.
recode act6pw (else=sysmis).
end if.
```

```
DO IF exctim7 <15 or dayexc7 =0.
recode act7pw (else=sysmis).
end if.

DO IF exctim8 <15 or dayexc8 =0.
recode act8pw (else=sysmis).
end if.

DO IF exctim9 <15 or dayexc9 =0.
recode act9pw (else=sysmis).
end if.

DO IF exctim10 <15 or dayexc10 =0.
recode act10pw (else=sysmis).
end if.

DO IF exctim11 <15 or dayexc11 =0.
recode act11pw (else=sysmis).
end if.

DO IF exctim12 <15 or dayexc12 =0.
recode act12pw (else=sysmis).
end if.

DO IF exctim13 <15 or dayexc13 =0.
recode act13pw (else=sysmis).
end if.

VARIABLE LABEL act1pw '(I) Number of hours spent on swimming per week with
exclusions'.
VARIABLE LABEL act2pw '(I) Number of hours spent on cycling per week with exclusions'.
VARIABLE LABEL act3pw '(I) Number of hours spent on gym per week with exclusions'.
VARIABLE LABEL act4pw '(I) Number of hours spent on aerobics per week with
exclusions'.
VARIABLE LABEL act5pw '(I) Number of hours spent on dancing per week with exclusions'.
VARIABLE LABEL act6pw '(I) Number of hours spent on running per week with exclusions'.
VARIABLE LABEL act7pw '(I) Number of hours spent on football per week with
exclusions'.
VARIABLE LABEL act8pw '(I) Number of hours spent on badmington per week with
exclusions'.
VARIABLE LABEL act9pw '(I) Number of hours spent on squash per week with exclusions'.
VARIABLE LABEL act10pw '(I) Number of hours spent on exercises per week with
exclusions'.
VARIABLE LABEL act11pw '(I) Number of hours spent on activity COTHAC11 per week with
exclusions'.
VARIABLE LABEL act12pw '(I) Number of hours spent on activity COTHAC12 per week with
exclusions'.
VARIABLE LABEL act13pw '(I) Number of hours spent on activity COTHAC13 per week with
exclusions'.


COMPUTE acthr = SUM(act1pw,act2pw,act3pw,act4pw,act5pw,act6pw,act7pw,act8pw
 ,act9pw,act10pw,act11pw,act12pw,act13pw) .
EXECUTE .
VARIABLE LABELS acthr '(I) Total number of hours spent doing exercise per week'.

COMPUTE househr = SUM(hwrkx,hvhwx,gardx,manwx) .
EXECUTE .
VARIABLE LABELS househr '(I) Total number of hours spent doing any housework per
week'.

COMPUTE totalhr = SUM(workhr, edhrs, househr, wlk5x, acthr) .
EXECUTE .
VARIABLE LABELS totalhr '(I) Total number of hours spent doing work, education,
housework, exercise and walking per week'.

*Assume 8hours of sleep for adults * 7days = 56hours/week.

do if ageactiv=2.
compute sleep = 56.
end if.
execute.
VARIABLE LABELS sleep '(I) Hours of sleep per week'.

do if ageactiv=2 .
```

```
COMPUTE totalhr2 = SUM(totalhr,sleep).
end if.
EXECUTE .
VARIABLE LABELS totalhr2 '(I) Number of hours spent doing any activity per week minus
leisure/leftover time'.


*WORK ACTIVITY LEVEL.
do if active = 1.
COMPUTE workmet = (6.5*workhr).
end if.

do if active = 2.
COMPUTE workmet = (3.5*workhr).
end if.

do if active = 3.
COMPUTE workmet = (2.3*workhr).
end if.

do if active = 4 .
COMPUTE workmet = (1.5*workhr).
end if.
EXECUTE .
variable labels workmet '(I) Activity score at work'.

*EDUCATION /COLLEGE ACTIVITY LEVEL.
do if edact = 1.
COMPUTE edmet = (3.7*edhrs).
end if.
do if edact = 2 .
COMPUTE edmet = (3.1*edhrs).
end if.
do if edact = 3.
COMPUTE edmet = (2.3*edhrs).
end if.
do if edact = 4 .
COMPUTE edmet = (1.5*edhrs).
end if.
EXECUTE .
variable labels edmet '(I) Activity score at college'.

*HOUSEWORK ACTVITY LEVEL.
do if hwrkx >=0.
COMPUTE hwlmet = (2.96*hwrkx).
end if.
EXECUTE .
do if hvhwx >=0.
COMPUTE hwhmet = (3.08*hvhwx).
end if.
EXECUTE .
do if gardx >=0.
COMPUTE gardmet = (3.89*gardx).
end if.
EXECUTE .
do if manwx >=0.
COMPUTE manmet = (5.93*manwx).
end if.
EXECUTE .

variable labels hwlmet '(I) Activity score of light housework'.
variable labels hwhmet '(I) Activity score of heavy housework'.
variable labels gardmet '(I) Activity score of gardening'.
variable labels manmet '(I) Activity score of manual work'.

*WALKING ACTIVITY LEVEL.
do if walkpace=1.
COMPUTE walkmet = (2.5*wlk5x).
end if.
do if walkpace=2.
COMPUTE walkmet = (3.3*wlk5x).
end if.
do if walkpace=3.
COMPUTE walkmet = (3.8*wlk5x).
end if.
do if walkpace=4.
```

```
COMPUTE walkmet = (5*wlk5x).
end if.
do if walkpace=5.
COMPUTE walkmet = (3.3*wlk5x).
end if.
EXECUTE .
variable labels walkmet '(I) Activity score of walking'.


*SLEEPING ACTIVITY LEVEL.
do if ageactiv=2 .
COMPUTE sleepmet = (0.9*56).
end if.
EXECUTE .
variable labels sleepmet '(I) Activity score of sleeping'.

*EXERCISE ACTIVITY LEVEL.
*Activity 1, swimming.
do if excswt=1.
compute exemet1=(8*act1pw).
end if.
do if excswt=2.
compute exemet1=(5*act1pw).
end if.
execute.

*Activity 2, cycling.
do if excswt2 =1.
compute exemet2 = (8*act2pw).
end if.
do if excswt2 =2.
compute exemet2 = (5*act2pw).
end if.
execute.

*Activity 3, gym.
do if excswt3 =1.
compute exemet3 = (6.9*act3pw).
end if.
do if excswt3 =2.
compute exemet3 = (3.33*act3pw).
end if.
execute.

*Activity 4, aerobics, keep fit, gymnastics, dance.
do if excswt4 =1.
compute exemet4 = (5.75*act4pw).
end if.
do if excswt4 =2.
compute exemet4 = (4.38*act4pw).
end if.
execute.

*Activity 5, dancing.
do if excswt5 =1.
compute exemet5 = (4.83*act5pw).
end if.
do if excswt5 =2.
compute exemet5 = (3.0*act5pw).
end if.
execute.

*Activity 6, running/jogging.
do if excswt6 =1.
compute exemet6 = (9*act6pw).
end if.
do if excswt6 =2.
compute exemet6 = (5*act6pw).
end if.
execute.

*Activity 7, football/rugby.
do if excswt7 =1.
compute exemet7 = (9.5*act7pw).
end if.
do if excswt7 =2.
```

```
compute exemet7 = (5*act7pw).
end if.
execute.

*Activity 8, badmington/tennis.
do if excswt8 =1.
compute exemet8 = (5.75*act8pw).
end if.
do if excswt8 =2.
compute exemet8 = (4.5*act8pw).
end if.
execute.

*Activity 9, squash.
do if excswt9 =1.
compute exemet9 = (12*act9pw).
end if.
do if excswt9 =2.
compute exemet9 = (5*act9pw).
end if.
execute.

*Activity 10, exercises.
do if excswt10 =1.
compute exemet10 = (8*act10pw).
end if.
do if excswt10 =2.
compute exemet10 = (4.25*act10pw).
end if.
execute.

variable labels exemet1 ' (I) Activity score exercise 1, swimming'.
variable labels exemet2 '(I) Activity score exercise 2, cycling'.
variable labels exemet3 '(I) Activity score exercise 3, gym'.
variable labels exemet4 '(I) Activity score exercise 4, aerobics'.
variable labels exemet5 '(I) Activity score exercise 5, dancing'.
variable labels exemet6 '(I) Activity score exercise 6, running'.
variable labels exemet7 '(I) Activity score exercise 7, football, rugby'.
variable labels exemet8 '(I) Activity score exercise 8, badmington, tennis'.
variable labels exemet9 '(I) Activity score exercise 9, squash'.
variable labels exemet10 '(I) Activity score exercise 10, exercises'.


*OTHER EXERCISE ACTIVITIES 11-13*.
*OTHER ACTIVITY 11, COTHAC11.
*Code 2, cycling.
do if excswt11 =1 & cothac11=2.
compute exemet11 = (8*act11pw).
end if.
do if excswt11 =2 & cothac11=2.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 7, football.
do if excswt11 =1 & cothac11=7.
compute exemet11 = (9.5*act11pw).
end if.
do if excswt11 =2 & cothac11=7.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 8, tennis/badmington.
do if excswt11 =1 & cothac11=8.
compute exemet11 = (5.75*act11pw).
end if.
do if excswt11 =2 & cothac11=8.
compute exemet11 = (4.5*act11pw).
end if.
execute.

*Code 19, basketball.
do if excswt11 =1 & cothac11=19.
compute exemet11 = (7*act11pw).
end if.
```

```
do if excswt11 =2 & cothac11=19.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 22, boxing.
do if excswt11 =1 & cothac11=22.
compute exemet11 = (9*act11pw).
end if.
do if excswt11 =2 & cothac11=22.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 26, climbing.
do if excswt11 =1 & cothac11=26.
compute exemet11 = (9*act11pw).
end if.
do if excswt11 =2 & cothac11=26.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 27, cricket.
do if excswt11 =1 & cothac11=27.
compute exemet11 = (5*act11pw).
end if.
do if excswt11 =2 & cothac11=27.
compute exemet11 = (3.5*act11pw).
end if.
execute.

*Code 30, darts (same value sweaty & non sweaty).
do if cothac11=30.
compute exemet11 = (2.5*act11pw).
end if.
execute.

*Code 33, drums.
do if excswt11 =1 & cothac11=33.
compute exemet11 = (4*act11pw).
end if.
do if excswt11 =2 & cothac11=33.
compute exemet11 = (2.5*act11pw).
end if.
execute.

*Code 39, golf.
do if excswt11 =1 & cothac11=39.
compute exemet11 = (4.5*act11pw).
end if.
do if excswt11 =2 & cothac11=39.
compute exemet11 = (3*act11pw).
end if.
execute.

*Code 45, horse riding.
do if excswt11 =1 & cothac11=45.
compute exemet11 = (4*act11pw).
end if.
do if excswt11 =2 & cothac11=45.
compute exemet11 = (2.5*act11pw).
end if.
execute.

*Code 46, ice skating.
do if excswt11 =1 & cothac11=46.
compute exemet11 = (7.25*act11pw).
end if.
do if excswt11 =2 & cothac11=46.
compute exemet11 = (4*act11pw).
end if.
execute.

*Code 49, kick boxing.
do if excswt11 =1 & cothac11=49.
```

```
compute exemet11 = (10*act11pw).
end if.
do if excswt11 =2 & cothac11=49.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 52, martial arts.
do if excswt11 =1 & cothac11=52.
compute exemet11 = (10*act11pw).
end if.
do if excswt11 =2 & cothac11=52.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 54, netball.
do if excswt11 =1 & cothac11=54.
compute exemet11 = (8*act11pw).
end if.
do if excswt11 =2 & cothac11=54.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 63, rounders.
do if excswt11 =1 & cothac11=63.
compute exemet11 = (5*act11pw).
end if.
do if excswt11 =2 & cothac11=63.
compute exemet11 = (4*act11pw).
end if.
execute.

*Code 64, rowing.
do if excswt11 =1 & cothac11=64.
compute exemet11 = (7*act11pw).
end if.
do if excswt11 =2 & cothac11=64.
compute exemet11 = (3*act11pw).
end if.
execute.

*Code 68, skateboarding.
do if excswt11 =1 & cothac11=68.
compute exemet11 = (5*act11pw).
end if.
do if excswt11 =2 & cothac11=68.
compute exemet11 = (4*act11pw).
end if.
execute.

*Code 69, skiing.
do if excswt11 =1 & cothac11=69.
compute exemet11 = (7*act11pw).
end if.
do if excswt11 =2 & cothac11=69.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 70, skipping.
do if excswt11 =1 & cothac11=70.
compute exemet11 = (10*act11pw).
end if.
do if excswt11 =2 & cothac11=70.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 72, skittles (same value sweaty and non sweaty).
do if cothac11=72.
compute exemet11 = (3*act11pw).
end if.
execute.
```

```
*Code 73, snooker (same value sweaty and non sweaty).
do if cothac11=73.
compute exemet11 = (2.5*act11pw).
end if.
execute.

*Code 78, table tennis.
do if excswt11 =1 & cothac11=78.
compute exemet11 = (4*act11pw).
end if.
do if excswt11 =2 & cothac11=78.
compute exemet11 = (3*act11pw).
end if.
execute.

*Code 89, yoga (same value sweaty and non sweaty).
do if cothac11=89.
compute exemet11 = (2.5*act11pw).
end if.
execute.

*Code 91, other moderate exercise.
do if excswt11 =1 & cothac11=91.
compute exemet11 = (6.33*act11pw).
end if.
do if excswt11 =2 & cothac11=91.
compute exemet11 = (3.33*act11pw).
end if.
execute.

*Code 92, other vigorous exercise.
do if excswt11 =1 & cothac11=92.
compute exemet11 = (8.63*act11pw).
end if.
do if excswt11 =2 & cothac11=92.
compute exemet11 = (5*act11pw).
end if.
execute.

*Code 97, other don't know (coded met as moderate level).
do if excswt11 =1 & cothac11=97.
compute exemet11 = (6.33*act11pw).
end if.
do if excswt11 =2 & cothac11=97.
compute exemet11 = (3.33*act11pw).
end if.
execute.

*Code 98, other don't know (coded met as moderate level).
do if excswt11 =1 & cothac11=98.
compute exemet11 = (6.33*act11pw).
end if.
do if excswt11 =2 & cothac11=98.
compute exemet11 = (3.33*act11pw).
end if.
execute.

*OTHER ACTIVITY 12, COTHAC12.
*Code 19, basketball.
do if excswt12 =1 & cothac12=19.
compute exemet12 = (7*act12pw).
end if.
do if excswt12 =2 & cothac12=19.
compute exemet12 = (5*act12pw).
end if.
execute.

*Code 22, boxing.
do if excswt12 =1 & cothac12=22.
compute exemet12 = (9*act12pw).
end if.
do if excswt12 =2 & cothac12=22.
compute exemet12 = (5*act12pw).
end if.
execute.
```

```
*Code 30, darts (same value sweaty & non sweaty).
do if cothac12=30.
compute exemet12 = (2.5*act12pw).
end if.
execute.

*Code 73, snooker (same value sweaty and non sweaty).
do if cothac12=73.
compute exemet12 = (2.5*act12pw).
end if.
execute.

*Code 97, other don't know (coded met as moderate level).
do if excswt12 =1 & cothac12=97.
compute exemet12 = (6.33*act12pw).
end if.
do if excswt12 =2 & cothac12=97.
compute exemet12 = (3.33*act12pw).
end if.
execute.

*OTHER ACTIVITY 13, COTHAC13.
*Code 70, skipping.
do if excswt13 =1 & cothac13=70.
compute exemet13 = (10*act13pw).
end if.
do if excswt13 =2 & cothac13=70.
compute exemet13 = (5*act13pw).
end if.
execute.

*Code 97, other don't know (coded met as moderate level).
do if excswt13 =1 & cothac13=97.
compute exemet13 = (6.33*act13pw).
end if.
do if excswt13 =2 & cothac13=97.
compute exemet13 = (3.33*act13pw).
end if.
execute.

variable labels exemet11 '(I) Activity score exercise 11, COTHAC11'.
variable labels exemet12 '(I) Activity score exercise 12, COTHAC12'.
variable labels exemet13 '(I) Activity score exercise 13, COTHAC13'.


*LEFTOVER / LEISURE TIME*.
do if ageactiv=2.
compute leftmet = (168-totalhr2)*1.
end if.
execute.
variable labels leftmet '(I) Leftover time / Activity score leftover time - leisure'.

*Exclude people with values >than 1 week.
if shserial =1156861 & serp=1 leftmet=-9.
if shserial =1086748 & serp=1 leftmet=-9.
if shserial =1030547 & serp=2 leftmet=-9.
if shserial =1093354 & serp=1 leftmet=-9.
if shserial =1053782 & serp=2 leftmet=-9.
if shserial =1155770 & serp=1 leftmet=-9.
if shserial =1125574 & serp=1 leftmet=-9.
if shserial =1148361 & serp=1 leftmet=-9.
if shserial =1115956 & serp=1 leftmet=-9.
if shserial =1117278 & serp=1 leftmet=-9.
value labels leftmet -9 'Excluded as >168hours of activity recorded'.
MIS val leftmet (-9).

*do double check of 168 hours per week.
compute totalhr3 = sum(totalhr2,leftmet).
if shserial =1156861 & serp=1 totalhr3=-9.
if shserial =1086748 & serp=1 totalhr3=-9.
if shserial =1030547 & serp=2 totalhr3=-9.
if shserial =1093354 & serp=1 totalhr3=-9.
if shserial =1053782 & serp=2 totalhr3=-9.
if shserial =1155770 & serp=1 totalhr3=-9.
if shserial =1125574 & serp=1 totalhr3=-9.
if shserial =1148361 & serp=1 totalhr3=-9.
```

```
if shserial =1115956 & serp=1 totalhr3=-9.
if shserial =1117278 & serp=1 totalhr3=-9.
variable labels totalhr3 '(I) Check of hours per week = 168'.
value labels totalhr3 -9 'Excluded as >168hours of activity recorded'.
MIS val totalhr3 (-9).

*Adults - Total MET for day.
do if ageactiv=2 .
compute totalmet = sum
(workmet,edmet,hwlmet,hwhmet,gardmet,manmet,walkmet,sleepmet,exemet1,exemet2,exemet3,e
xemet4,exemet5,exemet6,exemet7,exemet8,
exemet9,exemet10,exemet11,exemet12,exemet13,leftmet)/7.
end if.
execute.

if shserial =1156861 & serp=1 totalmet=-9.
if shserial =1086748 & serp=1 totalmet=-9.
if shserial =1030547 & serp=2 totalmet=-9.
if shserial =1093354 & serp=1 totalmet=-9.
if shserial =1053782 & serp=2 totalmet=-9.
if shserial =1155770 & serp=1 totalmet=-9.
if shserial =1125574 & serp=1 totalmet=-9.
if shserial =1148361 & serp=1 totalmet=-9.
if shserial =1115956 & serp=1 totalmet=-9.
if shserial =1117278 & serp=1 totalmet=-9.
variable labels totalmet '(I) Adults total met value per day'.
value labels totalmet -9 'Excluded as >168hours of activity recorded'.
MIS val totalmet (-9).


***************
CHILDREN
***************.

*WALKING.
do if daywlkt = 1.
compute cwlkhr = (9.5*dayswlk)/60.
end if.
do if daywlkt = 2.
compute cwlkhr = (22*dayswlk)/60.
end if.
do if daywlkt = 3.
compute cwlkhr = (44.5*dayswlk)/60.
end if.
do if daywlkt = 4.
compute cwlkhr = (74.5*dayswlk)/60.
end if.
do if daywlkt = 5.
compute cwlkhr = (104.5*dayswlk)/60.
end if.
do if daywlkt = 6.
compute cwlkhr = (134.5*dayswlk)/60.
end if.
do if daywlkt = 7.
compute cwlkhr = (164.5*dayswlk)/60.
end if.
do if daywlkt = 8.
compute cwlkhr = (194.5*dayswlk)/60.
end if.
do if daywlkt = 9.
compute cwlkhr = (224.5*dayswlk)/60.
end if.

do if daywlkt = 10.
compute cwlkhr = (wlktot*dayswlk/60).
end if.
execute.
VARIABLE LABELS cwlkhr '(I) Children - Total number of hours spent walking per week'.

*HOUSEWORK.
do if thwk = 3.
compute cthwkhr = (22*dhwkch)/60.
end if.
do if thwk = 4.
compute cthwkhr = (44.5*dhwkch)/60.
end if.
```

```
do if thwk = 5.
compute cthwkhr = (74.5*dhwkch)/60.
end if.
do if thwk = 6.
compute cthwkhr = (104.5*dhwkch)/60.
end if.
do if thwk = 7.
compute cthwkhr = (134.5*dhwkch)/60.
end if.
do if thwk = 8.
compute cthwkhr = (164.5*dhwkch)/60.
end if.
do if thwk = 9.
compute cthwkhr = (194.5*dhwkch)/60.
end if.
do if thwk = 10.
compute cthwkhr = (224.5*dhwkch)/60.
end if.
VARIABLE LABELS cthwkhr '(I) Children - Total number of hours spent doing housework
per week'.

*SPORTS  / ACTIVITIES.
recode dwesp (1=1) (2=1) (3=2) (else = copy) into wesptday.
VARIABLE LABELS wesptday '(I) Children - Number of weekend days spent doing sport'.
Value labels wesptday 1'One day - Saturday or Sunday' 2'Two days - Saturday and
Sunday'.

do if wespor = 1.
compute wesporhr = (2.5*wesptday)/60.
end if.
do if wespor = 2.
compute wesporhr = (9.5*wesptday)/60.
end if.
do if wespor = 3.
compute wesporhr = (22*wesptday)/60.
end if.
do if wespor = 4.
compute wesporhr = (44.5*wesptday)/60.
end if.
do if wespor = 5.
compute wesporhr = (74.5*wesptday)/60.
end if.
do if wespor = 6.
compute wesporhr = (104.5*wesptday)/60.
end if.
do if wespor = 7.
compute wesporhr = (134.5*wesptday)/60.
end if.
do if wespor = 8.
compute wesporhr = (164.5*wesptday)/60.
end if.
do if wespor = 9.
compute wesporhr = (194.5*wesptday)/60.
end if.
do if wespor = 10.
compute wesporhr = (224.5*wesptday)/60.
end if.

do if wespor = 11.
compute wesporhr = (wespt*wesptday/60).
end if.
execute.

VARIABLE LABELS wesporhr '(I) Children - number of hours spent doing sport from card Z
at the weekend per week'.



recode daysp (1=0) (2=1) (3=2) (4=3) (5=4) (6=5) (else=copy) into wdsptday.
VARIABLE LABELS wdsptday '(I) Children - Number of week days spent doing sport'.
Value labels wdsptday 0'None' 1'One day' 2'Two days' 3'Three days' 4'Four days' 5'Five
days'.

do if wkspor = 1.
compute wdsporhr = (2.5*wdsptday)/60.
end if.
```

```
do if wkspor = 2.
compute wdsporhr = (9.5*wdsptday)/60.
end if.
do if wkspor = 3.
compute wdsporhr = (22*wdsptday)/60.
end if.
do if wkspor = 4.
compute wdsporhr = (44.5*wdsptday)/60.
end if.
do if wkspor = 5.
compute wdsporhr = (74.5*wdsptday)/60.
end if.
do if wkspor = 6.
compute wdsporhr = (104.5*wdsptday)/60.
end if.
do if wkspor = 7.
compute wdsporhr = (134.5*wdsptday)/60.
end if.
do if wkspor = 8.
compute wdsporhr = (164.5*wdsptday)/60.
end if.
do if wkspor = 9.
compute wdsporhr = (194.5*wdsptday)/60.
end if.
do if wkspor = 10.
compute wdsporhr = (224.5*wdsptday)/60.
end if.

do if wkspor = 11.
compute wdsporhr = (wkspt*wdsptday/60).
end if.
execute.

VARIABLE LABELS wdsporhr '(I) Children - number of hours spent doing sport from card Z
on weekdays per week'.


*TOTAL SPORT HOURS     from card Z.
compute chsporhr=SUM(wesporhr,wdsporhr).
VARIABLE LABELS chsporhr '(I) Children - TOTAL number of hours spent doing sport from
card Z per week'.


*OTHER ACTIVITIES on card ZA at WEEKEND.
recode dweact (1=1) (2=1) (3=2) (else = copy) into weactday.
VARIABLE LABELS weactday '(I) Children - Number of weekend days spent doing other
activities'.
Value labels weactday 1'One day - Saturday or Sunday' 2'Two days - Saturday and
Sunday'.

do if weact = 1.
compute weacthr = (2.5*weactday)/60.
end if.
do if weact = 2.
compute weacthr = (9.5*weactday)/60.
end if.
do if weact = 3.
compute weacthr = (22*weactday)/60.
end if.
do if weact = 4.
compute weacthr = (44.5*weactday)/60.
end if.
do if weact = 5.
compute weacthr = (74.5*weactday)/60.
end if.
do if weact = 6.
compute weacthr = (104.5*weactday)/60.
end if.
do if weact = 7.
compute weacthr = (134.5*weactday)/60.
end if.
do if weact = 8.
compute weacthr = (164.5*weactday)/60.
end if.
do if weact = 9.
compute weacthr = (194.5*weactday)/60.
```

```
end if.
do if weact = 10.
compute weacthr = (224.5*weactday)/60.
end if.

do if weact =11.
recode weactt (-1 = -8).
end if.

do if weact = 11.
compute weacthr = (weactt*weactday/60).
end if.
execute.

VARIABLE LABELS weacthr '(I) Children - number of hours spent doing activities from
card ZA at the weekend per week'.



*ACTIVITIES on card ZA on WEEKDAYS.
recode wkactdo (1=0) (2=1) (3=2) (4=3) (5=4) (6=5) (else=copy) into wkactday.
VARIABLE LABELS wkactday '(I) Children - Number of week days spent doing activities'.
Value labels wkactday 0'None' 1'One day' 2'Two days' 3'Three days' 4'Four days' 5'Five
days'.

do if wkact = 1.
compute wkacthr = (2.5*wkactday)/60.
end if.
do if wkact = 2.
compute wkacthr = (9.5*wkactday)/60.
end if.
do if wkact = 3.
compute wkacthr = (22*wkactday)/60.
end if.
do if wkact = 4.
compute wkacthr = (44.5*wkactday)/60.
end if.
do if wkact = 5.
compute wkacthr = (74.5*wkactday)/60.
end if.
do if wkact = 6.
compute wkacthr = (104.5*wkactday)/60.
end if.
do if wkact = 7.
compute wkacthr = (134.5*wkactday)/60.
end if.
do if wkact = 8.
compute wkacthr = (164.5*wkactday)/60.
end if.
do if wkact = 9.
compute wkacthr = (194.5*wkactday)/60.
end if.
do if wkact = 10.
compute wkacthr = (224.5*wkactday)/60.
end if.

do if wkact =11.
recode wkactt (-1 = -8).
end if.

do if wkact = 11.
compute wkacthr = (wkactt*wkactday/60).
end if.
execute.

VARIABLE LABELS wkacthr '(I) Children - number of hours spent doing activities from
card ZA on weekdays per week'.


*TOTAL ACTIVITY HOURS from card ZA.
compute chacthr = sum (weacthr,wkacthr).
VARIABLE LABELS chacthr '(I) Children - TOTAL number of hours spent doing activity
from card ZA per week'.
```

```
*Assume 6.25hrs/day = 31.25hrs/week spent in school/nursery/similar level of activity.
do if ageactiv=1.
compute schoolhr = 31.25.
end if.
execute.
VARIABLE LABELS schoolhr '(I) Children - number of hours spent at school per week'.



COMPUTE chtothr = SUM(cthwkhr, cwlkhr, chsporhr, schoolhr, chacthr) .
EXECUTE .
VARIABLE LABELS chtothr '(I) Children - number of hours spent doing housework,
walking, sports, school, activity per week'.



*Assume following hours of sleep for children:
*2-4yrs = 12hrs/day = 84hrs/week
*5-7yrs = 11hrs/day = 77hrs/week
*8-11yrs = 10hrs/day = 70hrs/week
*12-15yrs = 9hrs/day = 63hrs/week.

do if agep <5.
compute chsleep = 84.
end if.
execute.
do if agep =5.
compute chsleep = 77.
end if.
execute.
do if agep =6.
compute chsleep = 77.
end if.
execute.
do if agep =7.
compute chsleep = 77.
end if.
execute.
do if agep =8.
compute chsleep = 70.
end if.
execute.
do if agep =9.
compute chsleep = 70.
end if.
execute.
do if agep =10.
compute chsleep = 70.
end if.
execute.
do if agep =11.
compute chsleep = 70.
end if.
execute.
do if agep =12 .
compute chsleep = 63.
end if.
execute.
do if agep =13.
compute chsleep = 63.
end if.
execute.
do if agep =14.
compute chsleep = 63.
end if.
execute.
do if agep =15.
compute chsleep = 63.
end if.
execute.

VARIABLE LABELS chsleep '(I) Children - hours of sleep per week'.



COMPUTE chtothr2 = SUM(chtothr,chsleep).
```

```
EXECUTE .
VARIABLE LABELS chtothr2 '(I) Children - total number of hours spent doing any
activity including sleep per week exc leftover or leisure time'.


*WALKING ACTIVITY LEVEL.
do if chpace=1.
COMPUTE chwlkmet = (2.5*cwlkhr).
end if.
do if chpace=2.
COMPUTE chwlkmet = (3.3*cwlkhr).
end if.
do if chpace=3.
COMPUTE chwlkmet = (3.8*cwlkhr).
end if.
do if chpace=4.
COMPUTE chwlkmet = (5*cwlkhr).
end if.
do if chpace=5.
COMPUTE chwlkmet = (3.3*cwlkhr).
end if.
EXECUTE .

variable labels chwlkmet '(I) Children - Activity score of walking'.

* HOUSEWORK ACTIVITY LEVEL.
COMPUTE chhwkmet = (4*cthwkhr).
variable labels chhwkmet '(I) Children - Activity score of housework/ gardening'.

*SLEEPING ACTIVITY LEVEL.
do if ageactiv=1.
COMPUTE chslpmet = (0.9*chsleep).
end if.
EXECUTE .
variable labels chslpmet '(I) Children - Activity score of sleeping'.

*SCHOOL ACTIVITY LEVEL.
do if ageactiv=1.
COMPUTE schoomet = (2.3*schoolhr).
end if.
EXECUTE .
variable labels schoomet '(I) Children - Activity score of school'.

*EXERCISE ACTIVITY LEVEL.
compute chspomet=(6.25*chsporhr).
execute.
variable labels chspomet '(I) Children - Activity score of sport'.

*OTHER ACTIVITIES LEVEL.
compute chactmet=(4.97*chacthr).
execute.
variable labels chactmet '(I) Children - Activity score of other activities'.


*LEFTOVER / LEISURE TIME.
do if ageactiv=1.
compute chlefmet = (168-chtothr2)*1.
end if.
execute.
variable labels chlefmet '(I) Children - Leftover time / Activity score of leftover
time - leisure'.

*MET value of leftover time will be as leisure (1.0) so therefore same as number of
hours leftover.
*Exclude 5 respondents with >168 hours.
if shserial =1122254 & serp=2 chlefmet=-9.
if shserial =1091184 & serp=2 chlefmet=-9.
if shserial =1131182 & serp=2 chlefmet=-9.
if shserial =1057133 & serp=2 chlefmet=-9.
if shserial =1085799 & serp=2 chlefmet=-9.
value labels chlefmet -9 'Excluded as >168hours of activity recorded'.
MIS val chlefmet (-9).

*Double check of 168 hours per week.
compute chtothr3 = sum(chtothr2,chlefmet).
if shserial =1122254 & serp=2 chtothr3=-9.
```

```
if shserial =1091184 & serp=2 chtothr3=-9.
if shserial =1131182 & serp=2 chtothr3=-9.
if shserial =1057133 & serp=2 chtothr3=-9.
if shserial =1085799 & serp=2 chtothr3=-9.
variable labels chtothr3 '(I) Children - Check of hours per week = 168'.
value labels chtothr3 -9 'Excluded as >168hours of activity recorded'.
MIS val chtothr3 (-9).


*Children - Total MET for day.
do if ageactiv=1.
compute chtotmet = sum
(chwlkmet,chhwkmet,chspomet,chactmet,chslpmet,schoomet,chlefmet)/7.
end if.
execute.

if shserial =1122254 & serp=2 chtotmet=-9.
if shserial =1091184 & serp=2 chtotmet=-9.
if shserial =1131182 & serp=2 chtotmet=-9.
if shserial =1057133 & serp=2 chtotmet=-9.
if shserial =1085799 & serp=2 chtotmet=-9.
variable labels chtotmet '(I) Children - Total met value per day'.
value labels chtotmet -9 'Excluded as >168hours of activity recorded'.
MIS val chtotmet (-9).


do if ageactiv=1.
recode chtotmet (else=copy) into Metac.
end if.
do if ageactiv=2.
recode totalmet (else=copy) into Metac.
end if.


Variable labels metac '  Adults and children final MET value'.
value labels metac -9 'Excluded as >168hours of activity recorded'.
MIS val metac (-9).
```

Chapter 16: Smoking and Drinking behaviour

CIGDYAL (D) Number of cigarettes smoke a day - inc. non-smokers

```
SPSS Syntax

missing values all ().
IF cigwdy>=0 & cigwdy>=0 cigdyal=((5*cigwdy)+(2*cigwed))/7.
IF ANY(-9,cigwdy,cigwed) cigdyal=-9.
IF ANY(-8,cigwdy,cigwed) cigdyal=-8.
IF agep<16 cigdyal=-1.
RECODE cignow(-9,-8,-1=COPY)(2=0) INTO cigdyal.
RECODE smkevr(-9,-8,-1=COPY)(2=0) INTO cigdyal.
RECODE cigever(-9,-8=COPY)(2=0) INTO cigdyal.
```

CIGSTA (D) Cigarette Smoking Status: Current/Ex-Reg/Never-Reg

```
1 Current cigarette smoker
2 Ex-regular cigarette smoker
3 Never regular cigarette smoker
```

```
SPSS Syntax

** overall cig smoking status.
IF any(2,cigever,smkevr) cigsta=3.
recode cignow(1=1) (2=2) into cigsta.
IF ANY(-9,smkevr,cignow,cigever) cigsta=-9.
IF ANY(-8,smkevr,cignow,cigever) cigsta=-8.
IF smkevr=-1 cigsta=-1.
IF agep<16 cigsta=-1.
VARIABLE LABELS cigsta "(D) Cigarette Smoking Status: Current/Ex-Reg/Never-Reg".
VALUE LABELS cigsta
  1 "Current cigarette smoker"
  2 "Ex-regular cigarette smoker"
  3 "Never regular cigarette smoker".
```

CIGST2 (D) Cigarette Smoking Status - Banded current smokers

```
1 Light smokers, under 10 a day
2 Moderate smokers, 10 to under 20 a day
3 Heavy smokers, 20 or more a day
4 Don't know number smoked a day
5 Non-smoker
```

```
SPSS Syntax

** current cigarette smokers status.
RECODE cigdyal (-9=4)(-8=4)(-1=-1)(20 thru hi=3)(10 thru 20=2)(0 thru 10=1) INTO
cigst2.
RECODE cignow (-9=-9)(-8=-8)(2=5) INTO cigst2.
RECODE smkevr (-9=-9)(-8=-8)(-1=-1)(2=5) INTO cigst2.
IF agep<16 cigst2=-1.
VARIABLE LABEL cigst2 "(D) Cigarette Smoking Status - Banded current smokers".
VALUE LABELS cigst2
  1 "Light smokers, under 10 a day"
  2 "Moderate smokers, 10 to under 20 a day"
  3 "Heavy smokers, 20 or more a day"
  4 "Don't know number smoked a day"
  5 "Non-smoker".
```

KCIGREGG (D) Frequency of cigarette smoking (8-15s)

1 Have never smoked
2 Smoked once or twice
3 Used to smoke sometimes, but not now
4 Smoke sometimes, not every week
5 Smoke every week

```
SPSS Syntax

recode kcigreg (lo thru -1=COPY)(1=1)(2=2)(3=3)(4=4)(5,6=5) INTO kcigregg.
VARIABLE LABELS kcigregg "(D) Frequency of cigarette smoking (8-15s) (grouped)".
VALUE LABELS kcigregg
  1 "Have never smoked"
  2 "Smoked once or twice"
  3 "Used to smoke sometimes, but not now"
  4 "Smoke sometimes, not every week"
  5 "Smoke every week".
```

Missing values for drinking variables

```
SPSS Syntax

missing values smkevr to kcigreg (-99 thru -1).
add value labels smkevr to kcigreg
 -9 "No answer/refused"
 -8 "Don't know"
 -7 "Refused/not obtained"
 -6 "Schedule not obtained"
 -2 "Schedule not applicable"
 -1 "Item not applicable".
execute.
```

DNOFT2 (D) Frequency drink alcohol in past 12 months: including non-drinkers

1 Almost every day
2 Five or six days a week
3 Three or four days a week
4 Once or twice a week
5 Once or twice a month
6 Once every couple of months
7 Once or twice a year
8 Not at all in the last 12 months/Non-drinker

```
SPSS Syntax

DO IF agep>=19.
compute dnoft2=dnoft.
recode dnany(2=8)(-9,-8=COPY) into dnoft2.
recode dnnow(-9,-8=COPY) into dnoft2.
variable labels dnoft2 "(D) Frequency drink alcohol in past 12 months: including non-
drinkers".
value labels dnoft2
  1 "Almost every day"
  2 "Five or six days a week"
  3 "Three or four days a week"
  4 "Once or twice a week"
  5 "Once or twice a month"
  6 "Once every couple of months"
  7 "Once or twice a year"
  8 "Not at all in the last 12 months/Non-drinker".
missing values dnoft2 (-9 thru -1).
END IF.
```

D7TOTG (D) Alcohol consumption units in last week

0 None in last week
1 1-7
2 Over 7-10
3 Over 10-14
4 Over 14-21
5 Over 21-28
6 Over 28-35
7 Over 35-50
8 Over 50.

```
SPSS Syntax

***d7unit HAS TO BE DONE FOR EACH OF 7 DAYS FOR LIDNS d7unit1 thru d7unit7.
***this is d7unit1 first day in week.
missing values all ().
COMPUTE d7unit1=0.
IF (NBMeas=1) d7unit1=d7unit1+NBNum.
IF (NBMeas=2) d7unit1=d7unit1+NBNum.
IF (NBMeas=3) d7unit1=d7unit1+(NBNum*2).
IF (NBMeas=4) d7unit1=d7unit1+NBNum.
IF (SBMeas=1) d7unit1=d7unit1+(SBNum*1.5).
IF (SBMeas=2) d7unit1=d7unit1+(SBNum*1.5).
IF (SBMeas=3) d7unit1=d7unit1+(SBNum*1.5*2).
IF (SBMeas=4) d7unit1=d7unit1+(SBNum*1.5).
IF ANY(3, Drnktyp1,Drnktyp2,Drnktyp3) d7unit1=d7unit1+SprLNum.
IF ANY(4, Drnktyp1,Drnktyp2,Drnktyp3) d7unit1=d7unit1+ShryNum.
IF ANY(5, Drnktyp1,Drnktyp2,Drnktyp3) d7unit1=d7unit1+WineNum.
IF ANY(6, Drnktyp1,Drnktyp2,Drnktyp3) d7unit1=d7unit1+PopsNum.
IF ANY(-9,Drnktyp1,Drnktyp2,Drnktyp3,NBNum,SBNum,SprLNum,ShryNum,WineNum, PopsNum)
d7unit1=-9.
IF ANY(-8,Drnktyp1,Drnktyp2,Drnktyp3,NBNum,SBNum,SprLNum,ShryNum,WineNum, PopsNum)
d7unit1=-8.
IF (dofweek>0 and NBNum =-1 and SBNum =-1 and SprLNum =-1 and ShryNum =-1 and
WineNum=-1 and PopsNum=-1)
     d7unit1=-9.
IF any(drinkl7,2,-1) d7unit1=-1.
IF dofweek=-1 d7unit1=-1.
VARIABLE LABEL d7unit1 "(D) Units drunk on day 1 in last week".

***this is d7unit2 second day in week.
missing values all ().
COMPUTE d7unit2=0.
IF (NBMeas2=1) d7unit2=d7unit2+NBNum2.
IF (NBMeas2=2) d7unit2=d7unit2+NBNum2.
IF (NBMeas2=3) d7unit2=d7unit2+(NBNum2*2).
IF (NBMeas2=4) d7unit2=d7unit2+NBNum2.
IF (SBMeas2=1) d7unit2=d7unit2+(SBNum2*1.5).
IF (SBMeas2=2) d7unit2=d7unit2+(SBNum2*1.5).
IF (SBMeas2=3) d7unit2=d7unit2+(SBNum2*1.5*2).
IF (SBMeas2=4) d7unit2=d7unit2+(SBNum2*1.5).
IF ANY(3, Drnktyp9,Drnkty10,Drnkty11) d7unit2=d7unit2+SprLNum2.
IF ANY(4, Drnktyp9,Drnkty10,Drnkty11) d7unit2=d7unit2+ShryNum2.
IF ANY(5, Drnktyp9,Drnkty10,Drnkty11) d7unit2=d7unit2+WineNum2.
IF ANY(6, Drnktyp9,Drnkty10,Drnkty11) d7unit2=d7unit2+PopsNum2.
IF ANY(-9,Drnktyp9,Drnkty10,Drnkty11,NBNum2,SBNum2,SprLNum2,ShryNum2,WineNum2,
PopsNum2) d7unit2=-9.
IF ANY(-8,Drnktyp9,Drnkty10,Drnkty11,NBNum2,SBNum2,SprLNum2,ShryNum2,WineNum2,
PopsNum2) d7unit2=-8.
IF (dofweek2>0 and NBNum2 =-1 and SBNum2 =-1 and SprLNum2 =-1 and ShryNum2 =-1 and
WineNum2=-1 and PopsNum2=-1) d7unit2=-9.
IF any(drinkl7,2,-1) d7unit2=-1.
IF dofweek2=-1 d7unit2=-1.
VARIABLE LABEL d7unit2 "(D) Units drunk on day 2 in last week".

***this is d7unit3 third day in week.
missing values all ().
COMPUTE d7unit3=0.
IF (NBMeas3=1) d7unit3=d7unit3+NBNum3.
IF (NBMeas3=2) d7unit3=d7unit3+NBNum3.
```

```
IF (NBMeas3=3) d7unit3=d7unit3+(NBNum3*2).
IF (NBMeas3=4) d7unit3=d7unit3+NBNum3.
IF (SBMeas3=1) d7unit3=d7unit3+(SBNum3*1.5).
IF (SBMeas3=2) d7unit3=d7unit3+(SBNum3*1.5).
IF (SBMeas3=3) d7unit3=d7unit3+(SBNum3*1.5*2).
IF (SBMeas3=4) d7unit3=d7unit3+(SBNum3*1.5).
IF ANY(3, Drnkty17,Drnkty18,Drnkty19) d7unit3=d7unit3+SprLNum3.
IF ANY(4, Drnkty17,Drnkty18,Drnkty19) d7unit3=d7unit3+ShryNum3.
IF ANY(5, Drnkty17,Drnkty18,Drnkty19) d7unit3=d7unit3+WineNum3.
IF ANY(6, Drnkty17,Drnkty18,Drnkty19) d7unit3=d7unit3+PopsNum3.
IF ANY(-9,Drnkty17,Drnkty18,Drnkty19,NBNum3,SBNum3,SprLNum3,ShryNum3,WineNum3,
PopsNum3) d7unit3=-9.
IF ANY(-8,Drnkty17,Drnkty18,Drnkty19,NBNum3,SBNum3,SprLNum3,ShryNum3,WineNum3,
PopsNum3) d7unit3=-8.
IF (dofweek3>0 and NBNum3 =-1 and SBNum3 =-1 and SprLNum3 =-1 and ShryNum3 =-1 and
WineNum3=-1
     and PopsNum3=-1) d7unit3=-9.
IF any(drinkl7,2,-1) d7unit3=-1.
IF dofweek3=-1 d7unit3=-1.
VARIABLE LABEL d7unit3 "(D) Units drunk on day 3 in last week".


***this is d7unit4 fourth day in week.
missing values all ().
COMPUTE d7unit4=0.
IF (NBMeas4=1) d7unit4=d7unit4+NBNum4.
IF (NBMeas4=2) d7unit4=d7unit4+NBNum4.
IF (NBMeas4=3) d7unit4=d7unit4+(NBNum4*2).
IF (NBMeas4=4) d7unit4=d7unit4+NBNum4.
IF (SBMeas4=1) d7unit4=d7unit4+(SBNum4*1.5).
IF (SBMeas4=2) d7unit4=d7unit4+(SBNum4*1.5).
IF (SBMeas4=3) d7unit4=d7unit4+(SBNum4*1.5*2).
IF (SBMeas4=4) d7unit4=d7unit4+(SBNum4*1.5).
IF ANY(3, Drnkty25,Drnkty26,Drnkty27,Drnkty28) d7unit4=d7unit4+SprLNum4.
IF ANY(4, Drnkty25,Drnkty26,Drnkty27,Drnkty28) d7unit4=d7unit4+ShryNum4.
IF ANY(5, Drnkty25,Drnkty26,Drnkty27,Drnkty28) d7unit4=d7unit4+WineNum4.
IF ANY(6, Drnkty25,Drnkty26,Drnkty27,Drnkty28) d7unit4=d7unit4+PopsNum4.
IF ANY(-9,Drnkty25,Drnkty26,Drnkty27,Drnkty28,
NBNum4,SBNum4,SprLNum4,ShryNum4,WineNum4, PopsNum4) d7unit4=-9.
IF ANY(-
8,Drnkty25,Drnkty26,Drnkty27,Drnkty28,NBNum4,SBNum4,SprLNum4,ShryNum4,WineNum4,
PopsNum4) d7unit4=-8.
IF (dofweek4>0 and NBNum4 =-1 and SBNum4 =-1 and SprLNum4 =-1 and ShryNum4 =-1 and
WineNum4=-1
     and PopsNum4=-1) d7unit4=-9.
IF any(drinkl7,2,-1) d7unit4=-1.
IF dofweek4=-1 d7unit4=-1.
VARIABLE LABEL d7unit4 "(D) Units drunk on day 4 in last week".

***this is d7unit5 fifth day in week.
missing values all ().
COMPUTE d7unit5=0.
IF (NBMeas5=1) d7unit5=d7unit5+NBNum5.
IF (NBMeas5=2) d7unit5=d7unit5+NBNum5.
IF (NBMeas5=3) d7unit5=d7unit5+(NBNum5*2).
IF (NBMeas5=4) d7unit5=d7unit5+NBNum5.
IF (SBMeas5=1) d7unit5=d7unit5+(SBNum5*1.5).
IF (SBMeas5=2) d7unit5=d7unit5+(SBNum5*1.5).
IF (SBMeas5=3) d7unit5=d7unit5+(SBNum5*1.5*2).
IF (SBMeas5=4) d7unit5=d7unit5+(SBNum5*1.5).
IF ANY(3, Drnkty33,Drnkty34,Drnkty35) d7unit5=d7unit5+SprLNum5.
IF ANY(4, Drnkty33,Drnkty34,Drnkty35) d7unit5=d7unit5+ShryNum5.
IF ANY(5, Drnkty33,Drnkty34,Drnkty35) d7unit5=d7unit5+WineNum5.
IF ANY(6, Drnkty33,Drnkty34,Drnkty35) d7unit5=d7unit5+PopsNum5.
IF ANY(-9,Drnkty33,Drnkty34,Drnkty35,NBNum5,SBNum5,SprLNum5,ShryNum5,WineNum5,
PopsNum5) d7unit5=-9.
IF ANY(-8,Drnkty33,Drnkty34,Drnkty35,NBNum5,SBNum5,SprLNum5,ShryNum5,WineNum5,
PopsNum5) d7unit5=-8.
IF (dofweek5>0 and NBNum5 =-1 and SBNum5 =-1 and SprLNum5 =-1 and ShryNum5 =-1 and
WineNum5=-1
     and PopsNum5=-1) d7unit5=-9.
IF any(drinkl7,2,-1) d7unit5=-1.
IF dofweek5=-1 d7unit5=-1.
VARIABLE LABEL d7unit5 "(D) Units drunk on day 5 in last week".
```

```
***this is d7unit6 sixth day in week.
missing values all ().
COMPUTE d7unit6=0.
IF (NBMeas6=1) d7unit6=d7unit6+NBNum6.
IF (NBMeas6=2) d7unit6=d7unit6+NBNum6.
IF (NBMeas6=3) d7unit6=d7unit6+(NBNum6*2).
IF (NBMeas6=4) d7unit6=d7unit6+NBNum6.
IF (SBMeas6=1) d7unit6=d7unit6+(SBNum6*1.5).
IF (SBMeas6=2) d7unit6=d7unit6+(SBNum6*1.5).
IF (SBMeas6=3) d7unit6=d7unit6+(SBNum6*1.5*2).
IF (SBMeas6=4) d7unit6=d7unit6+(SBNum6*1.5).
IF ANY(3, Drnkty41,Drnkty42,Drnkty43) d7unit6=d7unit6+SprLNum6.
IF ANY(4, Drnkty41,Drnkty42,Drnkty43) d7unit6=d7unit6+ShryNum6.
IF ANY(5, Drnkty41,Drnkty42,Drnkty43) d7unit6=d7unit6+WineNum6.
IF ANY(6, Drnkty41,Drnkty42,Drnkty43) d7unit6=d7unit6+PopsNum6.
IF ANY(-9,Drnkty41,Drnkty42,Drnkty43,NBNum6,SBNum6,SprLNum6,ShryNum6,WineNum6,
PopsNum6) d7unit6=-9.
IF ANY(-8,Drnkty41,Drnkty42,Drnkty43,NBNum6,SBNum6,SprLNum6,ShryNum6,WineNum6,
PopsNum6) d7unit6=-8.
IF (dofweek6>0 and NBNum6 =-1 and SBNum6 =-1 and SprLNum6 =-1 and ShryNum6 =-1 and
WineNum6=-1
     and PopsNum6=-1) d7unit6=-9.
IF any(drinkl7,2,-1) d7unit6=-1.
IF dofweek6=-1 d7unit6=-1.
VARIABLE LABEL d7unit6 "(D) Units drunk on day 6 in last week".

***this is d7unit7 seventh day in week.
missing values all ().
COMPUTE d7unit7=0.
IF (NBMeas7=1) d7unit7=d7unit7+NBNum7.
IF (NBMeas7=2) d7unit7=d7unit7+NBNum7.
IF (NBMeas7=3) d7unit7=d7unit7+(NBNum7*2).
IF (NBMeas7=4) d7unit7=d7unit7+NBNum7.
IF (SBMeas7=1) d7unit7=d7unit7+(SBNum7*1.5).
IF (SBMeas7=2) d7unit7=d7unit7+(SBNum7*1.5).
IF (SBMeas7=3) d7unit7=d7unit7+(SBNum7*1.5*2).
IF (SBMeas7=4) d7unit7=d7unit7+(SBNum7*1.5).
IF ANY(3, Drnkty49,Drnkty50,Drnkty51) d7unit7=d7unit7+SprLNum7.
IF ANY(4, Drnkty49,Drnkty50,Drnkty51) d7unit7=d7unit7+ShryNum7.
IF ANY(5, Drnkty49,Drnkty50,Drnkty51) d7unit7=d7unit7+WineNum7.
IF ANY(6, Drnkty49,Drnkty50,Drnkty51) d7unit7=d7unit7+PopsNum7.
IF ANY(-9,Drnkty49,Drnkty50,Drnkty51,NBNum7,SBNum7,SprLNum7,ShryNum7,WineNum7,
PopsNum7) d7unit7=-9.
IF ANY(-8,Drnkty49,Drnkty50,Drnkty51,NBNum7,SBNum7,SprLNum7,ShryNum7,WineNum7,
PopsNum7) d7unit7=-8.
IF (dofweek7>0 and NBNum7 =-1 and SBNum7 =-1 and SprLNum7 =-1 and ShryNum7 =-1 and
WineNum7=-1
     and PopsNum7=-1) d7unit7=-9.
IF any(drinkl7,2,-1) d7unit7=-1.
IF dofweek7=-1 d7unit7=-1.


***Now need to calculate weekly total d7total.
Do repeat xxx=d7unit1 d7unit2 d7unit3 d7unit4 d7unit5 d7unit6 d7unit7.
     Recode xxx (-1=0).
      End repeat.
COMPUTE d7total = d7unit1 + d7unit2 + d7unit3 + d7unit4 + d7unit5+ d7unit6 + d7unit7.
IF ANY(-9,d7unit1, d7unit2, d7unit3, d7unit4,d7unit5,d7unit6,d7unit7) d7total=-9.
IF ANY(-8,d7unit1, d7unit2, d7unit3, d7unit4,d7unit5,d7unit6,d7unit7) d7total=-8.
IF any(drinkl7,2,-1) d7total=0.
Do repeat xxx=d7unit1 d7unit2 d7unit3 d7unit4 d7unit5 d7unit6 d7unit7.
     Recode xxx (0=-1).
      End repeat.



***Derive variable to group weekly units.
***d7totg.
RECODE d7total (-9=-9) (-8=-8) (0=0) (0 thru 7=1) (7 thru 10=2) (10 thru 14 =3) (14
thru 21=4)
   (21 thru 28=5) (28 thru 35=6) (35 thru 50=7) (50 thru hi=8) INTO d7totg.
VARIABLE LABELS d7totg "(D) Alcohol consumption units in last week".
VALUE LABELS d7totg
0 "None in last week"
1 "1-7"
2 "Over 7-10"
```

```
3 "Over 10-14"
4 "Over 14-21"
5 "Over 21-28"
6 "Over 28-35"
7 "Over 35-50"
8 "Over 50".
```

DRNKDAY1 (D) Days drunk alcohol in past week: including non-drinkers

0 None
1 One day
2 Two days
3 Three days
4 Four days
5 Five days
6 Six days
7 Seven days

```
SPSS Syntax

temp.
select if agep>=19.
compute drnkday1=drnkday.
recode dnany(2=0)(-9,-8=COPY) into drnkday1.
recode dnnow(-9,-8=COPY) into drnkday1.
recode drinkl7(2=0)into drnkday1.
variable labels drnkday1 "(D) Days drunk alcohol in past week: including non-
drinkers".
value labels drnkday1
  0 "None"
  1 "One day"
  2 "Two days"
  3 "Three days"
  4 "Four days"
  5 "Five days"
  6 "Six days"
  7 "Seven days".
missing values drnkday1 (-9 thru -1).
```

Chapter 17: Oral Health

FRJGR (D) Fruit juice grams

```
SPSS Syntax

*calculation of total grams of fruit juice, maximum 157 g*.
compute frjgr=wa4fg84*34/100+wa4fg93.
execute.
var labels
frjgr '(D) Fruit juice grams'.
EXECUTE .
```

FRJGR157 (D) Fruit juice grams, max157g

```
SPSS Syntax

RECODE frjgr
(Lowest thru 157=Copy) (157.01 thru Highest=157) INTO frjgr157 .
VARIABLE LABELS frjgr157 '(D) Fruit juice grams, maximum 157'.
EXECUTE .
```

FRUITGR (D) Total grams of fruit consumed inc fruit juice (max 157g of fruit juice)

```
SPSS Syntax

*calculation of total grams of fruit, including fruit juice*.
compute fruitgr=wa4fg81+wa4fg82+wa4fg83+wa4fg84*66/100+wa4fg85*65/100+wa4fg86
+wa4fg13*45/100+wa4fg17*61/100+frjgr157.
execute.
var labels
fruitgr '(D) Total grams of fruit consumed inc fruit juice (max 157 g of fruit
juice)'.
```

FFRUIT (D) Total fresh fruit consumed, grams

```
SPSS Syntax

*calculation of total grams of fresh fruit*.
compute ffruit=wa4fg81+wa4fg82+wa4fg83+wa4fg86.
execute.
var labels
ffruit '(D) Total fresh fruit consumed, grams'.
```

FFRPORT (D) Fresh fruit portions

```
SPSS Syntax

*calcualate portions of fresh fruit, 80g equals one portion*.
compute ffrport=ffruit/80.
execute.
var labels
ffrport '(D) Fresh fruit portions'.
```

BBPUL (D) Baked beans and pulses consumption, grams

```
SPSS Syntax

*calculate total baked beans and pulses consumption, grams*.
compute bbpul=wa4fg68+wa4fg73.
execute.
var labels
bbpul '(D) Baked beans and pulses consumption, grams'.
```

BBPUL1 (D) Baked beans and pulses consumption, grams (max. of 80g)

```
SPSS Syntax

*calculate total baked beans and pulses consumption, grams (max. of 80g)*.
RECODE
bbpul
(Lowest thru 80=Copy) (80.01 thru Highest=80) INTO bbpul1 .
VARIABLE LABELS bbpul1 '(D) Baked beans and pulses consumption, grams (max. of 80g)'.
EXECUTE .
```

VEGTOTGR (D) Total veg consumption in grams, inc veg in dishes and pulses, baked beans and pulses max 80 grams

```
SPSS Syntax

*calculation of total grams of veg eaten, inc veg in composite dishes and pulses*.
compute vegtotgr=vegcomp+vegdis +
wa4fg63+wa4fg64+wa4fg65+wa4fg66+wa4fg67+wa4fg69+wa4fg70+wa4fg71+wa4fg75+bbpul1.
execute.
var labels
vegtotgr '(D) Total veg consumption in grams, inc veg in dishes and pulses, baked
beans and pulses max 80 grams'.
```

RVEGTOT (D) Raw veg consumption, grams

```
SPSS Syntax

*calculation of total grams of raw veg*.
compute rvegtot=wa4fg63+wa4fg64+wa4fg65.
execute.
var labels
rvegtot '(D) Raw veg consumption, grams'.
```

RVEGPOR (D) Raw veg portions

```
SPSS Syntax

*calculate portions of raw veg, portion equals 80g*.
compute rvegpor=rvegtot/80.
execute.
var labels
rvegpor '(D) Raw veg portions'.
```

FRVEGGR (D) Total fruit and veg in grams

```
SPSS Syntax

**calcualate total fruit and vegetable grams.
compute frveggr=fruitgr+vegtotgr.
execute.
var labels
frveggr '(D) Total fruit and veg in grams'.
```

Chapter 18: Environmental and economic factors

MASHGR (D) Main shop groups

```
 1 Large supermarket, home delivery and more than one main
 2 Small supermarket, markets, corner shop etc
```

```
SPSS Syntax

GET
  FILE='C:\Individual.sav'.

compute mshop=shop01.
execute.

if (shop02>0)mshop=mainshp.
execute.

var labels mshop ' Main shop whether use one or more outlets'.
value labels mshop
1 'Large supermarket'
2 'Small supermarket'
3 'Local/corner shop (including newsagent)'
4 'Garage forecourt'
5 'Greengrocer'
6 'Butcher'
7 'Baker'
8 'Fishmonger'
9 'Market (including stalls)'
10 'Farm'
11 'Home delivery'
12 'Other'
13 'More than one of these'.

recode mshop
(1=1)(2 thru 10=2)(11=1)(12=2)(13=2)
into mashgr.


do if (mshop=13 & shop01=1).
recode mashgr (2=1).
end if.
execute.


do if (mshop=13 & shop02=1).
recode mashgr (2=1).
end if.
execute.


do if (mshop=13 & shop07=11).
recode mashgr (2=1).
end if.
execute.


var labels
mashgr '(D) Main shop groups'.

value labels mashgr
1 'Large supermarket, home delivery and more than 1 if l/s used'
2 'Small supermarket, markets, corner shop etc and combination'.
execute.
```

SHTIMGR (D) Shopping time groups

```
 1 less than 15 minutes
 2 15 minutes less than 30
 3 30 minutes less than 1 hour
```

4 Over 1 hour'.

SHTIG2B (D) Shopping time two groups B

1 less than 30 minutes
2 over 30 minutes

SHOFGR (D) Shopping frequency groups

1 Once or more a day
2 2 or 3 times a week
3 weekly
4 Less than twice a month

SHOFG2 (D) Shopping frequency 2 groups

1 up to 3 times a week
2 weekly or less

SHTRGR (D) Transport to shops groups

1 Walking and bicycle
2 By bus and train
3 By car and home delivery
4 By taxi and other

SHTRG2 (D) Transport two groups

1 Not by car
2 By car

```
SPSS Syntax

GET
  FILE='C:\Individual.sav'.

recode shoptime
(1=1)(2=2)(3=3)(4 thru 6=4)
into shtimgr.

var labels
shtimgr '(D) Shopping time groups'.

value labels shtimgr
1 'less than 15 minutes'
2 '15 minutes less than 30'
3 '30 minutes less than 1 hour'
4 'Over 1 hour'.
execute.

recode shoptime
(1=1)(2=1)(3 thru 6=2) into shtig2B

var labels
shtig2B '(D) Shopping time two groups B'.

value labels shtig2B
1 'less than 30 minutes'
2 'over 30 minutes'.

recode shopoft
(1=1)(2 =1)(3=2)(4=3)(5 thru 8=4)
into shofgr.

var labels
shofgr '(D) Shopping frequency groups'.

value labels shofgr
1 'Once or more a day'
2 '2 or 3 times a week'
3 'weekly'
4 'Less than twice a month'.
execute.

recode shopoft
(1 thru 3=1)(4 thru 8=2) into shofg2.

var labels
```

```
shofg2 '(D) Shopping frequency 2 groups'.

value labels shofg2
1 'up to 3 times a week'
2 'weekly or less'.

recode shoptrav
(1=1)(2 =2)(3=3)(4=4)(5=2)(6=1)(7=3)(8=4)
into shtrgr.

var labels
shtrgr '(D) Transport to shops groups'.

value labels shtrgr
1 'Walking and bicycle'
2 'By bus and train'
3 'By car and home delivery'
4 'By taxi and other'.
execute.

recode shoptrav
(1 thru 2 =1) (3=2)(4 thru 8=1) into shtrg2.

var labels
shtrg2 '(D) Transport two groups'.

value labels shtrg2
1 'Not by car'
2 'By car'.
```

FDEXPG2 (D) percent spent on food groups

1 <30%
2 >30%

```
GET
  FILE='C:\Individual.sav’.

SAVE OUTFILE='C:\PersonIncome.sav'
/ KEEP= Shserial serp
shopmuch
adepren2
suburb
allofr
allofr2
school
mshop
mashgr
shtimgr
shtig2B
shofgr
shofg2
shtrgr
shtrg2
friwor2
frewor2
microw
hob
oven
friwor3
frewor3
micwor
ovewor
hobring
kitwork
kiteqgr
hhnums.

GET
  FILE='C:\PersonIncome.sav'.

sort cases by shserial (A) serp (A).
```

```
**save file**.

GET
  FILE='C:\Income.sav'.

sort cases by shserial (A) serp (A).


MATCH FILES /FILE=*
 /TABLE='C:\PersonIncome.sav'
 /BY shserial serp.
EXECUTE.

COMPUTE foodexp = shopmuch / netinc2 * 100.
EXECUTE .

variable labels foodexp '(D) percent spent on food'.

RECODE
  foodexp
  (0 thru 29.9999999999999=1)  (30 thru hi=2) INTO  fdexpg2 .
VARIABLE LABELS fdexpg2 '(D) percent spent on food, groups'.
value labels fdexpg2
1 '<30%'
2 '>30'.
```

ADEPREN2 (D) IMD two groups

1 first four quintiles least deprived
2 lowest quintile, most deprived

```
GET
  FILE='C:\Individual.sav'.

recode adepren
(1=1)(2 =1)(3=1)(4=1)(5=2)
into adepren2.

var labels
adepren2 '(D) IMD two groups'.

value labels adepren2
1 'first four quintiles least deprived'
2 'lowest quintile, most deprived'.
```

SUBURB (D) suburban urban groups

1 suburban and rural
2 urban

```
GET
  FILE='C:\Individual.sav'.

recode urban (1=1)(2=1)(3=2) into suburb.

var labels
suburb '(D) suburban urban groups'.

value labels suburb
1 'suburban and rural'
2 'urban'.
execute.
```

ALLOFR (D) Gathers and grows food

2 Both
3 Either
4 Neither

```
GET
  FILE='C:\Individual.sav'.
```

```
compute allofr=allot+free.

var label allofr '(D) Gathers and grows food'.

value label allofr 2 'Both' 3'Either' 4'Neither'.

recode allofr (2=2)(3=3)(else =sysmis) into allofr2.

var label allofr2 'Gathers and grows food, both or either groups only'.
value label allofr2 2 'Both' 3'Either'.

**Merge final derived variables into Person file**.

GET FILE =’C:/Income.sav’.

SAVE OUTFILE='C:\PersonIncome.sav'
/ KEEP= Shserial serp
mashgr
shtimgr
shtig2b
shofgr
shofg2
shtrgr
shtrg2
fdexpg2
adepren2
suburb
allofr.

GET FILE=’C:/PersonIncome.sav’.

Sort cases by shserial (A) serp (A).

**save file**.

GET FILE=’C:\individual.sav’.

Sort cases by shserial (A) serp (A).

MATCH FILES /FILE=*
 /TABLE='C:\PersonIncome.sav'
 /BY shserial serp.
EXECUTE.
```

LUNFDGR (D) Lunchtime food group

1 Bought school food
2 Packed lunch
3 Free school meal

```
GET
  FILE='C:\Individual.sav’.


RECODE
  schprov
  (-8=1)  (1=1)  (2=1)  INTO  school .
VARIABLE LABELS school 'attends school'.
EXECUTE .

SAVE OUTFILE='C:\Person only school variable.sav'
/KEEP=shserial serp school /COMPRESSED.

GET
  FILE='C:\Person only school variable.sav'.

sort cases by shserial (A) serp (A).

SAVE OUTFILE='C:\Person only school variable.sav'
 /COMPRESSED.

GET
  FILE='C:\Food level data.sav'.
```

```
compute serp=respno.
execute.

sort cases by shserial (A) serp (A).

MATCH FILES /FILE=*
 /TABLE='C:\Person only school variable.sav'
 /BY shserial serp.
EXECUTE.

SAVE OUTFILE='C:\Food level data with sch variable.sav'
 /COMPRESSED.


RECODE
  school  (SYSMIS=2)  (1=1)  .
EXECUTE .
FILTER OFF.
USE ALL.
SELECT IF(school = 1).
EXECUTE .


SAVE OUTFILE='C:\Food level data v1 sch resp only.sav'
 /COMPRESSED.

GET
  FILE='C:\day of week.sav'.

sort cases by shserial serp dayno.


GET
  FILE='C:\Food level data v1 sch resp only.sav'.

sort cases by shserial serp dayno.


MATCH FILES /FILE=*
 /TABLE='C:\day of week.sav'
 /RENAME (agep sexp = d0 d1)
 /BY shserial serp dayno
 /DROP= d0 d1.
EXECUTE.


FILTER OFF.
USE ALL.
SELECT IF(dofwk > 1).
EXECUTE .
FILTER OFF.
USE ALL.
SELECT IF(dofwk  < 7).
EXECUTE .


SAVE OUTFILE='C:\Food level data v1 sch resp only wkdays only.sav'
 /COMPRESSED.

FILTER OFF.
USE ALL.
SELECT IF(fg5 < 83).
EXECUTE .

RECODE
  time
  (41400 thru 50400=1)  (else=0)  INTO  lunchti .
VARIABLE LABELS lunchti 'Ate at lunchtime'.
EXECUTE .

DO IF (lunchti = 1) .
RECODE
  place
  ('G'=1)('H'=2)('I'=3) (else=0) into lunsch.
end if.
variable labels lunsch 'Source of school lunchtime food'.
```

```
execute.

value labels lunsch
1 'Bought school food'
2 'Food from home'
3 'Free school meal'.
SAVE OUTFILE='C:\Food level data v1 sch resp only wkdays only.sav'
 /COMPRESSED.

FILTER OFF.
USE ALL.
SELECT IF(lunsch  >=  1).
EXECUTE .

SAVE OUTFILE='C:\Food level data v1 sch resp only wkdays only lunchtime'+
 ' only.sav'
 /COMPRESSED.


GET
  FILE='C:\Food level data v1 sch resp only wkdays only lunchtime only.sav'.


AGGREGATE
  /OUTFILE='C:\aggr day sch resp wkday lunchtime foods.sav'
  /BREAK=shserial serp sexp agep dayno lunsch fg5
  /wghttot 'wt food group 5' = SUM(wtfood1)
 /np1 = SUM(np1)
/np2 = SUM(np2)
/np3 = SUM(np3)
/np4 = SUM(np4)
/np5 = SUM(np5)
/np6 = SUM(np6)
 /np7 = SUM(np7)
/np8 = SUM(np8)
/np9 = SUM(np9)
 /np10 = SUM(np10)
/np11 = SUM(np11)
/np12 = SUM(np12)
 /np13 = SUM(np13)
/np14  = SUM(np14)
/np15 = SUM(np15)
/np16 = SUM(np16)
/np17 = SUM(np17)
/np18 = SUM(np18)
/np19 = SUM(np19)
/np20 = SUM(np20)
 /np21 = SUM(np21)
/np22 = SUM(np22)
/np23 = SUM(np23)
/np24 = SUM(np24)
/np25 = SUM(np25)
/np26 = SUM(np26)
/np27 = SUM(np27)
  /np28 = SUM(np28)
/np29 = SUM(np29)
/np30 = SUM(np30)
/np31 = SUM(np31)
/np32 = SUM(np32)
/np33 = SUM(np33)
/np34 =  SUM(np34)
/np35 = SUM(np35)
/np36 = SUM(np36)
/np37 = SUM(np37)
/np38 = SUM(np38)
/np39 = SUM(np39)
/np40 = SUM(np40)
/np41  = SUM(np41)
/np42 = SUM(np42)
/np43 = SUM(np43)
/np44 = SUM(np44)
 /np45 = SUM(np45)
/np46 = SUM(np46)
/np47 = SUM(np47)
 /np48 = SUM(np48)
/np49 = SUM(np49)
```

```
/np50 = SUM(np50)
/np51 = SUM(np51)
/np52 = SUM(np52)
/np53 = SUM(np53)
/np54 = SUM(np54)
  /np55 = SUM(np55)
/np56 = SUM(np56)
/np57 = SUM(np57)
 /np58 = SUM(np58)
/np59 = SUM(np59)
 /np60 = SUM(np60)
/school =  MEAN(school)
/lunchti = MEAN(lunchti)
/agegp2 = MEAN(agegp2)
/agegp6 = MEAN(agegp6)
/drvagegp = MEAN(drvagegp).

GET
  FILE='C:\aggr day sch resp wkday lunchtime foods.sav'.

IF(fg5=1) wg5fg1 = wghttot.
IF(fg5=2) wg5fg2 = wghttot.
IF(fg5=3) wg5fg3 = wghttot.
IF(fg5=4) wg5fg4 = wghttot.
IF(fg5=5) wg5fg5 = wghttot.
IF(fg5=6) wg5fg6 = wghttot.
IF(fg5=7) wg5fg7 = wghttot.
IF(fg5=8) wg5fg8 = wghttot.
IF(fg5=9) wg5fg9 = wghttot.
IF(fg5=10) wg5fg10 = wghttot.
IF(fg5=11) wg5fg11 = wghttot.
IF(fg5=12) wg5fg12 = wghttot.
IF(fg5=13) wg5fg13 = wghttot.
IF(fg5=14) wg5fg14 = wghttot.
IF(fg5=15) wg5fg15 = wghttot.
IF(fg5=16) wg5fg16 = wghttot.
IF(fg5=17) wg5fg17 = wghttot.
IF(fg5=18) wg5fg18 = wghttot.
IF(fg5=19) wg5fg19 = wghttot.
IF(fg5=20) wg5fg20 = wghttot.
IF(fg5=21) wg5fg21 = wghttot.
IF(fg5=22) wg5fg22 = wghttot.
IF(fg5=23) wg5fg23 = wghttot.
IF(fg5=24) wg5fg24 = wghttot.
IF(fg5=25) wg5fg25 = wghttot.
IF(fg5=26) wg5fg26 = wghttot.
IF(fg5=27) wg5fg27 = wghttot.
IF(fg5=28) wg5fg28 = wghttot.
IF(fg5=29) wg5fg29 = wghttot.
IF(fg5=30) wg5fg30 = wghttot.
IF(fg5=31) wg5fg31 = wghttot.
IF(fg5=32) wg5fg32 = wghttot.
IF(fg5=33) wg5fg33 = wghttot.
IF(fg5=34) wg5fg34 = wghttot.
IF(fg5=35) wg5fg35 = wghttot.
IF(fg5=36) wg5fg36 = wghttot.
IF(fg5=37) wg5fg37 = wghttot.
IF(fg5=38) wg5fg38 = wghttot.
IF(fg5=39) wg5fg39 = wghttot.
IF(fg5=40) wg5fg40 = wghttot.
IF(fg5=41) wg5fg41 = wghttot.
IF(fg5=42) wg5fg42 = wghttot.
IF(fg5=43) wg5fg43 = wghttot.
IF(fg5=44) wg5fg44 = wghttot.
IF(fg5=45) wg5fg45 = wghttot.
IF(fg5=46) wg5fg46 = wghttot.
IF(fg5=47) wg5fg47 = wghttot.
IF(fg5=48) wg5fg48 = wghttot.
IF(fg5=49) wg5fg49 = wghttot.
IF(fg5=50) wg5fg50 = wghttot.
IF(fg5=51) wg5fg51 = wghttot.
IF(fg5=52) wg5fg52 = wghttot.
IF(fg5=53) wg5fg53 = wghttot.
IF(fg5=54) wg5fg54 = wghttot.
IF(fg5=55) wg5fg55 = wghttot.
IF(fg5=56) wg5fg56 = wghttot.
```

```
IF(fg5=57) wg5fg57 = wghttot.
IF(fg5=58) wg5fg58 = wghttot.
IF(fg5=59) wg5fg59 = wghttot.
IF(fg5=60) wg5fg60 = wghttot.
IF(fg5=61) wg5fg61 = wghttot.
IF(fg5=62) wg5fg62 = wghttot.
IF(fg5=63) wg5fg63 = wghttot.
IF(fg5=64) wg5fg64 = wghttot.
IF(fg5=65) wg5fg65 = wghttot.
IF(fg5=66) wg5fg66 = wghttot.
IF(fg5=67) wg5fg67 = wghttot.
IF(fg5=68) wg5fg68 = wghttot.
IF(fg5=69) wg5fg69 = wghttot.
IF(fg5=70) wg5fg70 = wghttot.
IF(fg5=71) wg5fg71 = wghttot.
IF(fg5=72) wg5fg72 = wghttot.
IF(fg5=73) wg5fg73 = wghttot.
IF(fg5=74) wg5fg74 = wghttot.
IF(fg5=75) wg5fg75 = wghttot.
IF(fg5=76) wg5fg76 = wghttot.
IF(fg5=77) wg5fg77 = wghttot.
IF(fg5=78) wg5fg78 = wghttot.
IF(fg5=79) wg5fg79 = wghttot.
IF(fg5=80) wg5fg80 = wghttot.
IF(fg5=81) wg5fg81 = wghttot.
IF(fg5=82) wg5fg82 = wghttot.
EXECUTE .

format wg5fg1
wg5fg2
wg5fg3
wg5fg4
wg5fg5
wg5fg6
wg5fg7
wg5fg8
wg5fg9
wg5fg10
wg5fg11
wg5fg12
wg5fg13
wg5fg14
wg5fg15
wg5fg16
wg5fg17
wg5fg18
wg5fg19
wg5fg20
wg5fg21
wg5fg22
wg5fg23
wg5fg24
wg5fg25
wg5fg26
wg5fg27
wg5fg28
wg5fg29
wg5fg30
wg5fg31
wg5fg32
wg5fg33
wg5fg34
wg5fg35
wg5fg36
wg5fg37
wg5fg38
wg5fg39
wg5fg40
wg5fg41
wg5fg42
wg5fg43
wg5fg44
wg5fg45
wg5fg46
wg5fg47
wg5fg48
```

```
wg5fg49
wg5fg50
wg5fg51
wg5fg52
wg5fg53
wg5fg54
wg5fg55
wg5fg56
wg5fg57
wg5fg58
wg5fg59
wg5fg60
wg5fg61
wg5fg62
wg5fg63
wg5fg64
wg5fg65
wg5fg66
wg5fg67
wg5fg68
wg5fg69
wg5fg70
wg5fg71
wg5fg72
wg5fg73
wg5fg74
wg5fg75
wg5fg76
wg5fg77
wg5fg78
wg5fg79
wg5fg80
wg5fg81
wg5fg82 (f8.0).

IF(fg5~=1) wg5fg1=0.
IF(fg5~=2) wg5fg2=0.
IF(fg5~=3) wg5fg3=0.
IF(fg5~=4) wg5fg4=0.
IF(fg5~=5) wg5fg5=0.
IF(fg5~=6) wg5fg6=0.
IF(fg5~=7) wg5fg7=0.
IF(fg5~=8) wg5fg8=0.
IF(fg5~=9) wg5fg9=0.
IF(fg5~=10) wg5fg10=0.
IF(fg5~=11) wg5fg11=0.
IF(fg5~=12) wg5fg12=0.
IF(fg5~=13) wg5fg13=0.
IF(fg5~=14) wg5fg14=0.
IF(fg5~=15) wg5fg15=0.
IF(fg5~=16) wg5fg16=0.
IF(fg5~=17) wg5fg17=0.
IF(fg5~=18) wg5fg18=0.
IF(fg5~=19) wg5fg19=0.
IF(fg5~=20) wg5fg20=0.
IF(fg5~=21) wg5fg21=0.
IF(fg5~=22) wg5fg22=0.
IF(fg5~=23) wg5fg23=0.
IF(fg5~=24) wg5fg24=0.
IF(fg5~=25) wg5fg25=0.
IF(fg5~=26) wg5fg26=0.
IF(fg5~=27) wg5fg27=0.
IF(fg5~=28) wg5fg28=0.
IF(fg5~=29) wg5fg29=0.
IF(fg5~=30) wg5fg30=0.
IF(fg5~=31) wg5fg31=0.
IF(fg5~=32) wg5fg32=0.
IF(fg5~=33) wg5fg33=0.
IF(fg5~=34) wg5fg34=0.
IF(fg5~=35) wg5fg35=0.
IF(fg5~=36) wg5fg36=0.
IF(fg5~=37) wg5fg37=0.
IF(fg5~=38) wg5fg38=0.
IF(fg5~=39) wg5fg39=0.
IF(fg5~=40) wg5fg40=0.
IF(fg5~=41) wg5fg41=0.
```

```
IF(fg5~=42) wg5fg42=0.
IF(fg5~=43) wg5fg43=0.
IF(fg5~=44) wg5fg44=0.
IF(fg5~=45) wg5fg45=0.
IF(fg5~=46) wg5fg46=0.
IF(fg5~=47) wg5fg47=0.
IF(fg5~=48) wg5fg48=0.
IF(fg5~=49) wg5fg49=0.
IF(fg5~=50) wg5fg50=0.
IF(fg5~=51) wg5fg51=0.
IF(fg5~=52) wg5fg52=0.
IF(fg5~=53) wg5fg53=0.
IF(fg5~=54) wg5fg54=0.
IF(fg5~=55) wg5fg55=0.
IF(fg5~=56) wg5fg56=0.
IF(fg5~=57) wg5fg57=0.
IF(fg5~=58) wg5fg58=0.
IF(fg5~=59) wg5fg59=0.
IF(fg5~=60) wg5fg60=0.
IF(fg5~=61) wg5fg61=0.
IF(fg5~=62) wg5fg62=0.
IF(fg5~=63) wg5fg63=0.
IF(fg5~=64) wg5fg64=0.
IF(fg5~=65) wg5fg65=0.
IF(fg5~=66) wg5fg66=0.
IF(fg5~=67) wg5fg67=0.
IF(fg5~=68) wg5fg68=0.
IF(fg5~=69) wg5fg69=0.
IF(fg5~=70) wg5fg70=0.
IF(fg5~=71) wg5fg71=0.
IF(fg5~=72) wg5fg72=0.
IF(fg5~=73) wg5fg73=0.
IF(fg5~=74) wg5fg74=0.
IF(fg5~=75) wg5fg75=0.
IF(fg5~=76) wg5fg76=0.
IF(fg5~=77) wg5fg77=0.
IF(fg5~=78) wg5fg78=0.
IF(fg5~=79) wg5fg79=0.
IF(fg5~=80) wg5fg80=0.
IF(fg5~=81) wg5fg81=0.
IF(fg5~=82) wg5fg82=0.
execute.


DO IF (lunsch = 1) .
RECODE
  np7  (else=copy) into kcbou.
end if.
variable labels kcbou 'Energy from bought school food'.
execute.

DO IF (lunsch = 2) .
RECODE
  np7  (else=copy) into kchome.
end if.
variable labels kchome 'Energy from food brought from home'.
execute.

DO IF (lunsch = 3) .
RECODE
  np7  (else=copy) into kcfsm.
end if.
variable labels kcfsm 'Energy from free school meal'.
execute.


AGGREGATE
  /OUTFILE='C:\aggr day sch resp wkday lunchtime foods and nuts.sav'
  /BREAK=shserial serp sexp agep dayno
  /nl1 = SUM(np1)
/nl2 = SUM(np2)
/nl3 = SUM(np3)
/nl4 = SUM(np4)
/nl5 = SUM(np5)
/nl6 = SUM(np6)
 /nl7 = SUM(np7)
```

```
/nl8 = SUM(np8)
/nl9 = SUM(np9)
/nl10 = SUM(np10)
 /nl11 = SUM(np11)
/nl12 = SUM(np12)
  /nl13 = SUM(np13)
/nl14 = SUM(np14)
/nl15 = SUM(np15)
/nl16 = SUM(np16)
/nl17 = SUM(np17)
/nl18 =  SUM(np18)
/nl19 = SUM(np19)
/nl20 = SUM(np20)
/nl21 = SUM(np21)
 /nl22 = SUM(np22)
/nl23 = SUM(np23)
 /nl24 = SUM(np24)
/nl25 = SUM(np25)
/nl26 = SUM(np26)
/nl27 = SUM(np27)
/nl28 = SUM(np28)
/nl29 =  SUM(np29)
 /nl30 = SUM(np30)
/nl31 = SUM(np31)
/nl32 = SUM(np32)
/nl33 = SUM(np33)
/nl34 = SUM(np34)
 /nl35 = SUM(np35)
/nl36 = SUM(np36)
/nl37 = SUM(np37)
/nl38 = SUM(np38)
/nl39 = SUM(np39)
/nl40 =  SUM(np40)
/nl41 = SUM(np41)
 /nl42 = SUM(np42)
/nl43 = SUM(np43)
 /nl44 = SUM(np44)
 /nl45 = SUM(np45)
 /nl46 = SUM(np46)
/nl47 = SUM(np47)
/nl48 = SUM(np48)
/nl49 = SUM(np49)
/nl50 = SUM(np50)
/nl51 =  SUM(np51)
/nl52 = SUM(np52)
/nl53 = SUM(np53)
/nl54 = SUM(np54)
 /nl55 = SUM(np55)
/nl56 = SUM(np56)
 /nl57 = SUM(np57)
/nl58 = SUM(np58)
/nl59 = SUM(np59)
/nl60 = SUM(np60)
/wg5fg1 = SUM(wg5fg1)
/wg5fg2  = SUM(wg5fg2)
/wg5fg3 = SUM(wg5fg3)
/wg5fg4 = SUM(wg5fg4)
/wg5fg5 = SUM(wg5fg5)
/wg5fg6 = SUM(wg5fg6)
 /wg5fg7 = SUM(wg5fg7)
/wg5fg8 = SUM(wg5fg8)
/wg5fg9 = SUM(wg5fg9)
/wg5fg10 = SUM(wg5fg10)
/wg5fg11 = SUM(wg5fg11)
/wg5fg12 = SUM(wg5fg12)
/wg5fg13 = SUM(wg5fg13)
/wg5fg14 = SUM(wg5fg14)
/wg5fg15 = SUM(wg5fg15)
  /wg5fg16 = SUM(wg5fg16)
/wg5fg17 = SUM(wg5fg17)
/wg5fg18 = SUM(wg5fg18)
/wg5fg19 = SUM(wg5fg19)
/wg5fg20  = SUM(wg5fg20)
/wg5fg21 = SUM(wg5fg21)
/wg5fg22 = SUM(wg5fg22)
/wg5fg23 = SUM(wg5fg23)
```

```
/wg5fg24 =  SUM(wg5fg24)
 /wg5fg25 = SUM(wg5fg25)
 /wg5fg26 = SUM(wg5fg26)
 /wg5fg27 = SUM(wg5fg27)
/wg5fg28 = SUM(wg5fg28)
  /wg5fg29 = SUM(wg5fg29)
 /wg5fg30 = SUM(wg5fg30)
 /wg5fg31 = SUM(wg5fg31)
 /wg5fg32 = SUM(wg5fg32)
 /wg5fg33 = SUM(wg5fg33)
 /wg5fg34 = SUM(wg5fg34)
/wg5fg35 = SUM(wg5fg35)
 /wg5fg36 = SUM(wg5fg36)
 /wg5fg37 =  SUM(wg5fg37)
 /wg5fg38 = SUM(wg5fg38)
 /wg5fg39 = SUM(wg5fg39)
 /wg5fg40 = SUM(wg5fg40)
 /wg5fg41 = SUM(wg5fg41)
  /wg5fg42 = SUM(wg5fg42)
 /wg5fg43 = SUM(wg5fg43)
 /wg5fg44 = SUM(wg5fg44)
 /wg5fg45 = SUM(wg5fg45)
 /wg5fg46 = SUM(wg5fg46)
 /wg5fg47 = SUM(wg5fg47)
 /wg5fg48 = SUM(wg5fg48)
/wg5fg49 = SUM(wg5fg49)
 /wg5fg50 = SUM(wg5fg50)
 /wg5fg51 = SUM(wg5fg51)
/wg5fg52 = SUM(wg5fg52)
 /wg5fg53 = SUM(wg5fg53)
/wg5fg54 = SUM(wg5fg54)
  /wg5fg55 = SUM(wg5fg55)
 /wg5fg56 = SUM(wg5fg56)
 /wg5fg57 = SUM(wg5fg57)
/wg5fg58 = SUM(wg5fg58)
/wg5fg59 = SUM(wg5fg59)
 /wg5fg60 = SUM(wg5fg60)
 /wg5fg61 = SUM(wg5fg61)
/wg5fg62 = SUM(wg5fg62)
 /wg5fg63 =  SUM(wg5fg63)
 /wg5fg64 = SUM(wg5fg64)
/wg5fg65 = SUM(wg5fg65)
 /wg5fg66 = SUM(wg5fg66)
 /wg5fg67 = SUM(wg5fg67)
  /wg5fg68 = SUM(wg5fg68)
 /wg5fg69 = SUM(wg5fg69)
 /wg5fg70 = SUM(wg5fg70)
 /wg5fg71 = SUM(wg5fg71)
 /wg5fg72 = SUM(wg5fg72)
 /wg5fg73 = SUM(wg5fg73)
 /wg5fg74 = SUM(wg5fg74)
 /wg5fg75 = SUM(wg5fg75)
/wg5fg76 = SUM(wg5fg76)
/wg5fg77 = SUM(wg5fg77)
/wg5fg78 = SUM(wg5fg78)
/wg5fg79 = SUM(wg5fg79)
/wg5fg80 = SUM(wg5fg80)
  /wg5fg81 = SUM(wg5fg81)
/wg5fg82 = SUM(wg5fg82)
/school_2 = MEAN(school)
/luncht_2 = MEAN(lunchti)
 /agegp2_2 = MEAN(agegp2)
/agegp6_2 = MEAN(agegp6)
/drvage_2 = MEAN(drvagegp)
/kcbou_1 = SUM(kcbou)
 /kchome_1 = SUM(kchome)
/kcfsm_1 = SUM(kcfsm).


GET
  FILE='C:\aggr day sch resp wkday lunchtime foods and nuts.sav'.

compute kcbouP =kcbou_1/nl7*100.
compute kchomP=kchome_1/nl7*100.
compute kcfsmP=kcfsm_1/nl7*100.
```

```
RECODE
  kcboup  (0=SYSMIS)  (SYSMIS=SYSMIS)  (ELSE=copy)  .
EXECUTE .

RECODE
  kchomp  (0=SYSMIS)  (SYSMIS=SYSMIS)  (ELSE=copy)  .
EXECUTE .

RECODE
  kcfsmp  (0=SYSMIS)  (SYSMIS=SYSMIS)  (ELSE=copy)  .
EXECUTE .

RECODE
  kcboup
  (50 thru Highest=1)(ELSE=0)    INTO  kcbouL .
EXECUTE .
var label kcboul 'Bought school food'.

RECODE
  kchomp
  (50 thru Highest=2) (else=0) INTO  kchomL .
EXECUTE .
var label kchoml 'Packed lunch'.

RECODE
  kcfsmp
  (50 thru Highest=3) (else=0) INTO  kcfsmL .
EXECUTE .
var label kcfsml 'Free school meal'.

compute lunfdgr=kcboul+kchoml+kcfsml.
var label lunfdgr '(D) Lunchtime food group'.
value label lunfdgr 1 'Bought school food'  2 'Packed lunch' 3 'Free school meal'.


RECODE
  lunfdgr  (0=2)  .
EXECUTE .

**Merge lunfdgr into daylevel file**.

GET
  FILE='C:\aggr day sch resp wkday lunchtime foods and nuts.sav'.

SAVE OUTFILE='C:\lunfdgr.sav'
/ KEEP = shserial serp dayno lunfdgr.

GET
  FILE='C:\day level data.sav'.

sort cases by shserial (A) serp (A) dayno (A).

MATCH FILES /FILE=*
 /TABLE='C:\lunfdgr.sav'
 /BY shserial serp dayno.
EXECUTE.
```

Chapter 19: Social factors and food choice

QUAL2 (D) Qualifications gained, grouped into two groups

1 GCSE grades A-C or equivalent and above
2 Qualifications at a lower level or no qualifications

QUALP (D) Qualifications of the parent

1 GCSE grades A-C or equivalent and above
2 Qualifications at a lower level or no qualifications

```
GET
  FILE='C:\Individual.sav'.


missing values gno ().
compute gno1=-1.
compute gno2=-1.
if serp=1 gno1=gno.
if serp=2 gno2=gno.
var lab gno1 "Person number of resp. 1 in HH grid".
var lab gno2 "Person number of resp. 2 in HH grid".
execute.
missing values gno gno1 gno2 (-9 thru -1).

sort cases by shserial (A).
AGGREGATE
  /OUTFILE='C:\aggr.sav'
  /BREAK=shserial
  /gno1n = SUM(gno1) /gno2n = SUM(gno2).

GET FILE='C:\aggr.sav'.
sort cases by shserial.

SAVE OUTFILE ='C:\aggr.sav'
 /COMPRESSED.

GET
  FILE='C:\Individual.sav'.
sort cases by shserial.

MATCH FILES /FILE=*
 /TABLE='C:\aggr.sav'.
 /BY shserial.
EXECUTE.
var lab gno1n "Person number of resp. 1 in HH grid".
var lab gno2n "Person number of resp. 2 in HH grid".
execute.
missing values gno1n gno2n (-9 thru -1).

missing values serp gno1 gno2 gno1n gno2n rel1 rel2 rel3 rel4 rel5 rel6 rel7 rel8 rel9
rel10 ().
compute child=-1.
compute parent=-1.
* Who is "child"?.
if serp=2 & gno1n=1 & any (3, 4, 5, 6, rel1) child=1.
if serp=2 & gno1n=2 & any (3, 4, 5, 6, rel2) child=1.
if serp=2 & gno1n=3 & any (3, 4, 5, 6, rel3) child=1.
if serp=2 & gno1n=4 & any (3, 4, 5, 6, rel4) child=1.
if serp=2 & gno1n=5 & any (3, 4, 5, 6, rel5) child=1.
if serp=2 & gno1n=6 & any (3, 4, 5, 6, rel6) child=1.
if serp=2 & gno1n=7 & any (3, 4, 5, 6, rel7) child=1.
if serp=2 & gno1n=8 & any (3, 4, 5, 6, rel8) child=1.
if serp=2 & gno1n=9 & any (3, 4, 5, 6, rel9) child=1.
if serp=2 & gno1n=10 & any (3, 4, 5, 6, rel10) child=1.
if serp=1 & gno2n=1 & any (3, 4, 5, 6, rel1) child=1.
if serp=1 & gno2n=2 & any (3, 4, 5, 6, rel2) child=1.
if serp=1 & gno2n=3 & any (3, 4, 5, 6, rel3) child=1.
if serp=1 & gno2n=4 & any (3, 4, 5, 6, rel4) child=1.
if serp=1 & gno2n=5 & any (3, 4, 5, 6, rel5) child=1.
if serp=1 & gno2n=6 & any (3, 4, 5, 6, rel6) child=1.
if serp=1 & gno2n=7 & any (3, 4, 5, 6, rel7) child=1.
if serp=1 & gno2n=8 & any (3, 4, 5, 6, rel8) child=1.
if serp=1 & gno2n=9 & any (3, 4, 5, 6, rel9) child=1.
```

```
if serp=1 & gno2n=10 & any (3, 4, 5, 6, rel10) child=1.
* Who is parent?.
if serp=1 & gno2n=1 & any (8, 9, 10, 11, rel1) parent=1.
if serp=1 & gno2n=2 & any (8, 9, 10, 11, rel2) parent=1.
if serp=1 & gno2n=3 & any (8, 9, 10, 11, rel3) parent=1.
if serp=1 & gno2n=4 & any (8, 9, 10, 11, rel4) parent=1.
if serp=1 & gno2n=5 & any (8, 9, 10, 11, rel5) parent=1.
if serp=1 & gno2n=6 & any (8, 9, 10, 11, rel6) parent=1.
if serp=1 & gno2n=7 & any (8, 9, 10, 11, rel7) parent=1.
if serp=1 & gno2n=8 & any (8, 9, 10, 11, rel8) parent=1.
if serp=1 & gno2n=9 & any (8, 9, 10, 11, rel9) parent=1.
if serp=1 & gno2n=10 & any (8, 9, 10, 11, rel10) parent=1.
if serp=2 & gno1n=1 & any (8, 9, 10, 11, rel1) parent=1.
if serp=2 & gno1n=2 & any (8, 9, 10, 11, rel2) parent=1.
if serp=2 & gno1n=3 & any (8, 9, 10, 11, rel3) parent=1.
if serp=2 & gno1n=4 & any (8, 9, 10, 11, rel4) parent=1.
if serp=2 & gno1n=5 & any (8, 9, 10, 11, rel5) parent=1.
if serp=2 & gno1n=6 & any (8, 9, 10, 11, rel6) parent=1.
if serp=2 & gno1n=7 & any (8, 9, 10, 11, rel7) parent=1.
if serp=2 & gno1n=8 & any (8, 9, 10, 11, rel8) parent=1.
if serp=2 & gno1n=9 & any (8, 9, 10, 11, rel9) parent=1.
if serp=2 & gno1n=10 & any (8, 9, 10, 11, rel10) parent=1.
var lab child "Whether respondent is a child of the other respondent".
val lab child
1 'Child'.
var lab parent "Whether respondent is a parent of the other respondent".
val lab parent
1 'Parent'.
execute.
missing values serp gno1 gno2 gno1n gno2n rel1 rel2 rel3 rel4 rel5 rel6 rel7 rel8 rel9
rel10 child parent (-9 thru -1).
SAVE OUTFILE='C:\Individual.sav'.
 /COMPRESSED.

missing values qual qualch ().
RECODE qual (-9 thru -1 = COPY) (1 thru 17 = 1) (18 thru 25 =2) (26, 27 = 1) (31=2)
into qual2.
RECODE qualch (2 = 2) into qual2.
VARIABLE LABELS qual2 "(D) Qualifications gained, grouped into two groups".
val lab qual2
1 'GCSE grades A-C or equivalent and above'
2 'Qualifications at a lower level or no qualifications'.
execute.
missing values qual qualch qual2 (-9 thru -1).

*** EDUCATIONAL ATTAINMENT OF CHILDREN-RESPONDENTS' PARENTS.
*** Need to get information about education of the parent as a variable in the data
for the child.

missing values parent qual2 ().
compute qual2p=-1.
if parent=1 qual2p=qual2.
var lab qual2p "(D) Qualifications of the parent".
val lab qual2p
1 'GCSE grades A-C or equivalent and above'
2 'Qualifications at a lower level or no qualifications'.
execute.
missing values parent qual2 qual2p (-9 thru -1).

sort cases by shserial.

SAVE OUTFILE='C:\Individual.sav'
 /COMPRESSED.

AGGREGATE
  /OUTFILE='C:\aggr2.sav'
  /BREAK=shserial
  /qualp = SUM(qual2p).

GET FILE='C:\aggr2.sav'.
sort cases by shserial.

SAVE OUTFILE ='C:\aggr2.sav'
 /COMPRESSED.

GET
```

```
  FILE='C:\Individual.sav'.
sort cases by shserial.

MATCH FILES /FILE=*
 /TABLE='C:\aggr2.sav'
 /BY shserial.
execute.

missing values qualp child ().
if child<>1 qualp=-1.
execute.
missing values qualp child (-9 thru -1).
var lab qualp "(D) Qualifications of the parent".
val lab qualp
1 'GCSE grades A-C or equivalent and above'
2 'Qualifications at a lower level or no qualifications'.
execute.
```

SKILLS1 (D) Cooking skills of MFP: two levels

```
 1 Better developed skills
 2 Less developed skills
 -8 Dont know
 -1 Not Applicable
```

```
GET
  FILE='C:\Individual.sav'.
sort cases by shserial.

missing values hskadv ().
compute skills1=hskadv.
recode skills1 (2 thru 4=2) (else=copy).
var lab skills1 '(D) Cooking skills of MFP: two levels'.
val lab skills1
1 'Better developed skills'
2 'Less developed skills'
-8 'Dont know'
-1 'Not Applicable'.
execute.
missing values hskadv skills1 (-9 thru -1).
```

Chapter 20: Food Security

ENOUGH1 (D) MFP:Whether had enough food in last 12 mo - Coding score

WORRY1 (D) MFP:Worried in last 12 mo food would run out - Coding score

NODURA1 (D) MFP:Whether in last 12 mo food ran out due to no money - Coding score

EXPENS1 (D) MFP:Whether in last 12 mo could not afford balanced meals - Coding score

SKIP1 (D) MFP:Whether reduced portion/skipped meals in last 12mo - Coding score

XSKIP1 (D) MFP:How often reduced portion/skipped meals - Coding score

EATLES1 (D) MFP:Ate less than should in last 12mo due to lack of money - Coding score

HUNGER1 (D) MFP:Ever hungry in last 12mo due to lack of money - Coding score

WTLOS1 (D) MFP:Whether ever lost weight in last 12mo due to lack of money - Coding score

NOTEAT1 (D) MFP:Ever not eaten for whole day in last 12mo due to lack of money - Coding score

XNOTEAT1 (D) MFP:How often did not eat for whole day in last 12mo due to lack of money - Coding score

```
GET
  FILE='C:\Individual.sav'.

RECODE
  enough
  (1=0)  (2=0)  (3=1) (4=1) (ELSE=SYSMIS)  INTO  Enough1 .
VARIABLE LABELS Enough1 '(D) MFP:Whether had enough food in last 12 mo - Coding
score'.
value label Enough1 1'Positive' 0'Negative' .
EXECUTE .

RECODE
  worry
  (1=1)  (2=1)  (3=0)  (ELSE=SYSMIS)  INTO  Worry1 .
VARIABLE LABELS Worry1 '(D) MFP:Worried in last 12 mo food would run out - Coding
score'.
value label Worry1 1'Positive' 0'Negative' .

EXECUTE .

RECODE
```

```
  nodura
  (1=1)  (2=1)  (3=0)  (ELSE=SYSMIS)  INTO  Nodura1 .
VARIABLE LABELS Nodura1 '(D) MFP:Whether in last 12 mo food ran out due to no money -
Coding score'.
value label Nodura1 1'Positive' 0'Negative' .
EXECUTE .

RECODE
  expens
  (1=1)  (2=1)  (3=0)  (ELSE=SYSMIS)  INTO  expens1 .
VARIABLE LABELS expens1 '(D) MFP:Whether in last 12 mo could not afford balanced meals
- Coding score'.
value label expens1 1'Positive' 0'Negative' .
EXECUTE .


RECODE
  skip
  (1=1)  (2=0)  (else=copy)   INTO  skip1 .
EXECUTE .

RECODE
  skip1
  (-8=sysmis) (-1=0) INTO  skip1 .
VARIABLE LABELS skip1 '(D) MFP:Whether reduced portion/skipped meals in last 12mo -
Coding score'.
value label skip1 1'Positive' 0'Negative' .
EXECUTE .


RECODE
  xskip
  (1=1)  (2=1)  (3=0) (-1=0)   INTO  xskip1 .
VARIABLE LABELS xskip1 '(D) MFP:How often reduced portion/skipped meals - Coding
score'.
value label xskip1 1'Positive' 0'Negative' .
EXECUTE .

RECODE
  eatles
  (1=1)  (2=0)  (-1=0) (else=sysmis)   INTO  eatles1 .
VARIABLE LABELS eatles1 '(D) MFP:Ate less than should in last 12mo due to lack of
money - Coding score'.
value label eatles1 1'Positive' 0'Negative' .
EXECUTE .

RECODE
  hunger
  (1=1)  (2=0)  (-1=0) (else=sysmis)   INTO  hunger1 .
VARIABLE LABELS hunger1 '(D) MFP:Ever hungry in last 12mo due to lack of money -
Coding score'.
value label hunger1 1'Positive' 0'Negative' .
EXECUTE .

RECODE
  wtlos
  (1=1)  (2=0)  (-1=0) (else=sysmis)   INTO  wtlos1 .
VARIABLE LABELS wtlos1 '(D) MFP:Whether ever lost weight in last 12mo due to lack of
money - Coding score'.
value label wtlos1 1'Positive' 0'Negative' .
EXECUTE .

RECODE
  noteat
  (1=1)  (2=0)  (-1=0)  INTO  noteat1 .
VARIABLE LABELS noteat1 '(D) MFP:Ever not eaten for whole day in last 12mo due to lack
of money - Coding score'.
value label noteat1 1'Positive' 0'Negative' .
EXECUTE .

RECODE
  xnoteat
  (1=1)  (2=1) (3=0) (-1=0)  INTO  xnoteat1 .
VARIABLE LABELS xnoteat1 '(D) MFP:How often did not eat for whole day in last 12mo due
to lack of money - Coding score'.
value label xnoteat1 1'Positive' 0'Negative' .
```

```
EXECUTE .

recode skip1 (missing=9) into skip1.

do if (skip1=9).
recode xskip1 (0.00=sysmis) into xskip1.
end if.
execute.

recode skip1 (9=sysmis) into skip1.
```

SCREEN1 (D) Food security screener 1

1 Positive response, continue
0 Negative response, do not continue

SCREEN2 (D) Food security screener 2

1 Positive response, continue
0 Negative response, do not continue

```
GET
  FILE='C:\Individual.sav'.

compute Screen1=0.
recode screen1 (0=sysmis) into screen1.
If (enough1=1 or worry1=1 or nodura1=1 or expens1=1) Screen1=1.
If (enough1=0 & worry1=0 & nodura1=0 & expens1=0) Screen1=0.
execute.

Var label Screen1 '(D) Food security screener 1'.
value label Screen1 1'Positive response, continue' 0'Negative response, do not
continue'.



recode screen1 (missing=9) into screen1.

do if  (screen1=9).
recode skip1 xskip1 eatles1 hunger1 wtlos1 noteat1 xnoteat1 (0.00=sysmis) into skip1
xskip1 eatles1 hunger1 wtlos1 noteat1 xnoteat1.
end if.
execute.

recode screen1 (9=sysmis) into screen1.

compute Screen2=0.
recode screen2 (0=sysmis) into screen2.
If (skip1=1 or hunger1=1 or eatles1=1) Screen2=1.
If (skip1=0 & hunger1=0 & eatles1=0) Screen2=0.
execute.
Var label Screen2 '(D) Food security screener 2'.
value label Screen2 1'Positive response, continue' 0'Negative response, do not
continue'.



recode screen2 (missing=9) into screen2.

do if  (screen2=9).
recode noteat1 xnoteat1 (0.00=sysmis) into noteat1 xnoteat1.
end if.
execute.

recode screen2 (9=sysmis) into screen2.
```

SCORE1 (D) Food security score 1 Q1-10, all households

SCOREYN (D) Is there a score1 assigned

0 no score assigned
1 score assigned

FSGROUP1 (D) Food security grouped, 3groups

1 Food secure
2 Food insecure (mild)
3 Food insecure (moderate/severe)

FSGROUP2 (D) Food security grouped, 2groups

1 Food secure
2 Food insecure

```
GET
  FILE='C:\Individual.sav'.

compute score1=(worry1+nodura1+ expens1+skip1+xskip1+eatles1+hunger1+wtlos1+noteat1+
xnoteat1).
execute.
Var label score1 '(D) Food security score 1 Q1-10, all households'.

compute scoreyn=0.
recode score1 (0=1) (1=1) (2=1) (3=1) (4=1) (5=1) (6=1) (7=1) (8=1) (9=1) (10=1) into
scoreyn.
execute.
var label scoreyn '(D) Is there a score1 assigned'.
value label scoreyn 0'no score assigned' 1'score assigned'.

recode score1 (0=1) (1=1) (2=1) (3=2) (4=2) (5=2) (6=3) (7=3) (8=3) (9=3) (10=3)
(else=copy) into FSgroup1.
execute.
Var label FSgroup1 '(D) Food security grouped, 3groups'.
Value label FSgroup1 1'Food secure' 2'Food insecure (mild)' 3'Food insecure
(moderate/severe)'.

recode fsgroup1 (1=1) (2=2) (3=2) (else=copy) into fsgroup2.
variable label fsgroup2 '(D) Food security grouped, 2groups'.
value labels fsgroup2 1'Food secure' 2 'Food insecure'.
```

HHCOMPAC (D) Household with or without children

1 Household without children
2 Household with children

```
GET
  FILE='C:\Individual.sav'.

RECODE
  hhcomp
 (0=1) (1=1)  (2=1)  (3=1)  (4=1) (5=2)  (6=2)  INTO
  hhcompac .
VARIABLE LABELS hhcompac '(D) Household with or without children'.
EXECUTE .
value labels hhcompac 1'Household without children' 2'Household with children'.

do if (shserial=1038518) or (shserial=1055720) or (shserial=1111132) or
(shserial=1124782) or (shserial=1129105).
recode hhcompac (1=2) into hhcompac.
end if.
execute.
```

LO4MON1 (D) MFP:Whether ever cut size of childrens meals in last 12mo because no money for food - Coding score

1 Positive
0 Negative

```
GET
  FILE='C:\Individual.sav'.

RECODE
  lo4mon
  (1=1)  (2=0)  (-9=sysmis) (-8=sysmis) (else=copy)   INTO  lo4mon1 .
```

```
VARIABLE LABELS lo4mon1 '(D) MFP:Whether ever cut size of childrens meals in last 12mo
because no money for food - Coding score'.
value label lo4mon1 1'Positive' 0'Negative' .
EXECUTE .

DO IF (hhcompac = 1) .
RECODE
  lo4mon1  (-1=SYSMIS)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen1 = 0) .
RECODE
  lo4mon1  (-1=0)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen2 = 0) .
RECODE
  lo4mon1  (-1=0)  .
END IF .
EXECUTE .

RECODE
  lo4mon1  (-1=SYSMIS)  .
EXECUTE .
```

CSKIP1 (D) MFP:Children ever skipped meals in last 12mo because no money for food - Coding score

1 Positive
0 Negative

XCSKIP1 (D) MFP:How often have children ever skipped meals in last 12mo because no money for food - Coding score

1 Positive
0 Negative

```
GET
  FILE='C:\Individual.sav’.

RECODE
  cskip
  (1=1)  (2=0)  (-9=sysmis) (-8=sysmis) (else=copy)   INTO  cskip1 .
VARIABLE LABELS cskip1 '(D) MFP:Children ever skipped meals in last 12mo because no
money for food - Coding score'.
value label cskip1  1'Positive' 0'Negative' .
EXECUTE .

DO IF (hhcompac = 1) .
RECODE
  cskip1  (-1=SYSMIS)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen1 = 0) .
RECODE
 cskip1 (-1=0)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen2 = 0) .
RECODE
 cskip1 (-1=0)  .
END IF .
EXECUTE .

RECODE
  cskip1  (-1=SYSMIS)  .
EXECUTE .

RECODE
  xcskip
```

```
  (1=1)  (2=1)  (3=0) (else=copy)   INTO  xcskip1 .
VARIABLE LABELS xcskip1 '(D) MFP:How often have children ever skipped meals in last
12mo because no money for food - Coding score'.
value label xcskip1  1'Positive' 0'Negative' .
EXECUTE .

DO IF (hhcompac = 1) .
RECODE
  xcskip1  (-1=SYSMIS)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen1 = 0) .
RECODE
 xcskip1 (-1=0)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen2 = 0) .
RECODE
 xcskip1 (-1=0)  .
END IF .
EXECUTE .

do if (cskip1=0).
RECODE
  xcskip1  (-1=0)  .
END IF .
EXECUTE .

RECODE
 xcskip1 (-1=sysmis).
EXECUTE.
```

CHUNGER1 (D) MFP:Children were hungry in last 12mo but could not afford food - Coding score

1 Positive
0 Negative

```
GET
  FILE='C:\Individual.sav'.

RECODE
  chunger
  (1=1)  (2=0)  (-9=sysmis) (-8=sysmis) (else=copy)   INTO  chunger1 .
VARIABLE LABELS chunger1 '(D) MFP:Children were hungry in last 12mo but could not
afford food - Coding score'.
value label chunger1 1'Positive' 0'Negative' .
EXECUTE .

DO IF (hhcompac = 1) .
RECODE
  chunger1  (-1=SYSMIS)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen1 = 0) .
RECODE
 chunger1 (-1=0)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen2 = 0) .
RECODE
 chunger1 (-1=0)  .
END IF .
EXECUTE .

RECODE
  chunger1  (-1=SYSMIS)  .
EXECUTE .
```

CNOTEAT1 (D) MFP:Children did not eat for whole day in last 12mo because no money for food - Coding score

1 Positive
0 Negative

```
GET
  FILE='C:\Individual.sav'.

RECODE
  cnoteat
  (1=1)  (2=0)  (-9=sysmis) (-8=sysmis) (else=copy)   INTO  cnoteat1 .
VARIABLE LABELS cnoteat1 '(D) MFP:Children did not eat for whole day in last 12mo
because no money for food - Coding score'.
value label cnoteat1  1'Positive' 0'Negative' .
EXECUTE .

DO IF (hhcompac = 1) .
RECODE
  cnoteat1  (-1=SYSMIS)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen1 = 0) .
RECODE
 cnoteat1 (-1=0)  .
END IF .
EXECUTE .

DO IF (hhcompac = 2 & screen2 = 0) .
RECODE
 cnoteat1 (-1=0)  .
END IF .
EXECUTE .

RECODE
  cnoteat1  (-1=SYSMIS)  .
EXECUTE .
```

SCORE2 (D) Food security score 2 Q11-15, childrens questions only
SCORE3 (D) Food security score 3 Q1-15, all questions

```
GET
  FILE='C:\Individual.sav'.

compute score2=(lo4mon1+cskip1+ xcskip1+chunger1+cnoteat1).
execute.
Var label score2 '(D) Food security score 2 Q11-15, childrens questions only'.


do if (hhcompac=1).
compute score3=(score1).
end if.
execute.

do if (hhcompac=2).
compute score3=(score1+score2).
end if.
execute.
Var label score3 '(D) Food security score 3 Q1-15, all questions'.
```