APPENDIX 4

AVAILABILITY OF VEGETABLES & FRUIT IN AUSTRALIA AND GREECE

a Department of Trade, Athens, Greece.
b Department of Agriculture, Fresh Food Centre, Melbourne, Australia.
* Fruits and vegetables in peak season are cheaper in Greece and Australia.

	GREECE[a]		**AUSTRALIA**[b]	
S=available in summer W=available in winter	Peak Season No. of months*	Availability Total no. of months	Peak Season No. of months*	Availability Total no. of months
Artichokes	W3	9	W3	9
Green beans	S4	12	S4	12
Beetroot	S3	12	S5	12
Broad beans	S3	5	S4	6
Brocolli	W3	7	W9	12
Cabbage	W4	10	W8	12
Capsicum	S4	12	S7	12
Carrots	W6	12	W8	12
Cauliflower	W3	7	W9	12
Celery	W3	12	12	12
Chikory/endives	W6	12	12	12
Wild greens	W4	4	W4	4
Corn	S3	3	S3	3
Cucumber	S4	12	S4	12
Egg plant	S4	12	S4	12
Leeks	S4	6	S6	12
Lettuce	W4	12	S9	12
Mushrooms	W2	2	W7	12
Okra	W3	5	W3	5
Onions	S4	12	12	12
Peas	S3	6	S6	8
Snow peas	-	-	S5	8
Pumpkin	S9	12	S9	12
Silverbeet	W8	12	12	12
Spinach	W3	12	W6	12
Tomatoes	S4	12	S6	12
Vine leaves	S2	2	S2	2
Zucchini	S5	12	S8	12
Apples	W4	7	12	12
Apricots	S1	3	S1	3
Bananas	S4	6	S9	12
Berries	S2	3	S2	3
Canteloupe	S3	6	S3	6
Cherries	S1	3	S1	3
Figs	S1	3	S1	3
Grapefruit	W2	4	W6	10
Oranges	W2	4	W8	12
Peaches	S2	4	S2	4
Pears	S3	6	9	12
Pineapple	S2	4	S4	9
Plums	S1	3	S1	3
Pommegranate	S2	4	S2	4
Strawberries	S2	4	S4	6

APPENDIX 5

DURNIN & WOMERSLEY TABLES

THE SUM OF 4 SKINFOLDS

(biceps, triceps, subscapular, suprailiac)

% BODY FAT

Source: Durnin & Womersley, Br J Nutr 1974; 32: pp 95.

TRICEP SKINFOLD ONLY

(Table formulated by Body Composition laboratory at Monash Medical Centre
using the Durnin equation)

% BODY FAT

Tricep skinfold	MEN					WOMEN				
	17-19	20-29	30-39	40-49	50+	17-19	20-29	30-39	40-49	50+
5	7.7	10.0	17.8	16.3	18.6	12.4	9.3	12.8	15.3	15.8
7	11.6	13.3	20.1	20.2	22.9	16.5	14.2	17.1	19.8	20.7
9	14.5	15.9	21.9	23.2	26.2	19.6	17.9	20.5	23.2	24.5
11	16.9	17.9	23.3	25.7	28.8	22.1	20.9	23.2	25.9	27.5
13	18.9	19.6	24.5	27.7	31.1	24.3	23.5	25.4	28.2	30.2
15	20.7	21.1	25.5	29.4	33.0	26.1	25.7	27.4	30.2	32.3
17	22.2	22.4	26.4	31.0	34.7	27.7	27.6	29.1	32.0	34.2
19	23.5	23.5	27.2	32.4	36.2	29.2	29.3	30.6	33.5	36.0
21	24.8	24.6	28.0	33.6	37.6	30.5	30.9	32.0	34.9	37.6
23	25.9	25.5	28.6	34.7	38.9	31.7	32.3	33.3	36.2	39.5
25	27.0	26.4	29.2	35.8	40.0	32.8	33.7	34.5	37.4	40.4
27	27.9	27.2	29.8	36.8	41.1	33.8	34.9	35.6	38.6	41.0
29	28.8	28.0	30.3	37.7	42.1	34.8	36.0	36.6	39.6	42.8
31	29.7	28.7	30.8	38.5	43.0	35.6	37.1	37.5	40.6	43.0
33	30.4	29.4	31.2	39.3	43.9	36.5	38.1	38.4	41.5	44.0
35	31.2	30.0	31.7	40.1	44.8	37.3	39.1	39.3	42.3	45.0
37	31.9	30.6	32.1	40.8	45.6	38.0	40.0	40.1	43.2	46.0

APPENDIX 7

NUTRIENT CONVERSION

The following SAS programme written by Dr Bridget Hsu-Hage was used to convert grams/day data into nutrients.

```
DATA grk.NUTS2; SET grk.nut2; KEEP GRKCODE WATER--POLY_FAT; RUN;
PROC TRANSPOSE DATA=grk.nuts2 OUT=grk.nuts2; ID grkcode; RUN;
DATA grk.NUTS2; SET grk.NUTS2; _TYPE_='SCORE'; RUN;
PROC SCORE DATA=grk.ogFDDAY SCORE=grk.NUTS2 OUT=grk.ogNUT2;
 VAR F1--F30 F32--F254; RUN;
data grk.ognut2; set grk.ognut2;
 WATER=WATER/100;
 SUGER=SUGER/100;
 STARCH=STARCH/100;
 FIBRE=FIBRE/100;
 NITROGEN=NITROGEN/100;
 KCAL=KCAL/100;
 KJOULE=KJOULE/100;
 PROTEIN=PROTEIN/100;
 FAT_TOT=FAT_TOT/100;
 CARBO_TO=CARBO_TO/100;
 SODIUM=SODIUM/100;
 POTASS=POTASS/100;
 CALCIUM=CALCIUM/100;
 PHOSPH=PHOSPH/100;
 MAGNES=MAGNES/100;
 IRON=IRON/100;
 ZINC=ZINC/100;
 RETINOL=RETINOL/100;
 CAROTENE=CAROTENE/100;
 THIAMIN=THIAMIN/100;
 RIBOFLAV=RIBOFLAV/100;
 NICOTIN=NICOTIN/100;
 VIT_C=VIT_C/100;
 ALCOHOL=ALCOHOL/100;
 CHOLEST=CHOLEST/100;
 SAT_FAT=SAT_FAT/100;
 MONO_FAT=MONO_FAT/100;
 POLY_FAT=POLY_FAT/100; run;
DATA grk.NUTS2; SET grk.nut2; KEEP GRKCODE WATER--POLY_FAT; RUN;
PROC TRANSPOSE DATA=grk.NUTS2 OUT=grk.NUTS2; ID GRKCODE; RUN;
DATA grk.NUTS2; SET grk.NUTS2; _TYPE_='SCORE'; RUN;
PROC SCORE DATA=grk.spFDDAY SCORE=grk.NUTS2 OUT=grk.spNUT2;
VAR F1--F30 F32--F254; RUN;

PROC MEANS N MEAN DATA=GRK.OGNUT2;
CLASS SEX AGEGRP; VAR KCAL FAT_TOT CARBO_TO PROTEIN ALCOHOL;
RUN;
```

APPENDIX 9

FOOD GROUPS

FG1	EGGS
FG2	MILK-FRESH
FG3	*EVAPORATED MILK (diluted and added to milk-fresh)*
FG4	YOGHURT
FG5	CHEESE (e.g tiropita)
FG6	FISH (e.g tuna, sardines)
FG7	SHELLFISH (e.g crab, prawns)
FG8	LAMB/GOAT
FG9	VEAL&BEEF
FG10	PORK
FG11	CHICKEN/TURKEY
FG12	RABBIT/BIRD (e.g quail)
FG13	ORGAN MEATS (e.g brain, liver)
FG14	ROOT VEGIES (e.g potato,carrot)
FG15	GREEN LEAFY VEGIES (e.g cabbage,spinach pie)
FG16	MARROW VEGIES (e.g zucchini, pumpkin,cucumber)
FG17	FLOWER VEGIES (broccoli, cauliflower)
FG18	OTHER VEGIES (beetroot, celery, tomato, onion)
FG19	VEGIE MIXED DISH (e.g ratatouille, mousaka)
FG20	VEGIE RICE DISH (e.g leek,spinach rice)
FG21	LEGUMES (e.g bean soups)
FG22	BREAD
FG23	PASTA (e.g pastichio, noodles)
FG24	RICE
FG25	CEREALS (breakfast cereals, polenta, trahana)
FG26	CAKES,BISCUITS,PASTRIES (e.g baklava)
FG27	DESSERTS,PUDDINGS (e.g custard, galaktobureko)
FG28	CITRUS FRUITS
FG29	DRIED FRUIT
FG30	FRUIT JUICES
FG31	NUTS
FG32	DIPS (e.g taramosalata)
FG33	SOFTDRINKS
FG34	BEER
FG35	WINE
FG36	SPIRITS
FG37	TEA
FG38	COFFEE
FG39	WATER
FG40	FAT SPREAD (e.g butter)
FG41	OILS
FG42	SUGAR,JAM,HONEY
FG43	PROCESSED MEAT (e.g salami)
FG44	GARLIC
FG45	OLIVES
FG46	OTHER SPREADS (e.g vegemite)
FG47	SOUP (e.g meat,chicken,fish soup)
FG48	CRISPBREAD,DRY BISCUITS
FG49	SAVOURY PASTRY (e.g pizza, quiche)
FG50	REFINED SWEETS(e.glolly,chocolate,jelly,halva)
FG51	STONE FRUITS (e.g plums,apricots)
FG52	APPLE & PEAR FRUIT
FG53	MELONS FRUIT (e.g watermelon, canteloupe)
FG54	TROPICAL FRUIT (e.g mango,avocado)
FG55	OTHER FRUIT (e.g grapes,figs,banana,berries)

```
set grk.ogfdday;                    (programme written in SAS to create food groups)
FG1=F144;
FG2=F131+F132+F133+F136+F137+F187+F127+F211+(F134*2.4)+(F135*2.4);
FG3=F134+F135;
FG4=F138+F139+F140+F141;
FG5=F123+F124+F41+F125+F126+F128+F129+F130+F225+F226+F227;
FG6=F27+F28+F29+F30+F32+F185+F186+F154;
FG7=F33+F34+F35+F36+F37+F38+F39+F155+F221+F222;
FG8=F7+F8+F9+F22;
FG9=F14+F11+F12+F13+F15+F231+F232+F236+F237;
FG10=F23;
FG11=F1+F2+F3+F4+F5;
FG12=F6+F10;
FG13=F24+F25+F26;
FG14=F65+F66+F67+F69;
FG15=F73+F74+F75+F76+F40+F81;
FG16=F68+F84+F88;
FG17=F71+F72+F80;
FG18=F70+F82+F83+F86+F87+F89+F91+F92+F215+F216;
FG19=F17+F48+F49+F51+F52+F53;
FG20=F54+F43+F44+F45+F90+F50;
FG21=F55+F56+F57+F58+F59+F60+F61+F85+F177+F217+F218+F229+F238;
FG22=F145+F146+F147+F148+F149+F190;
FG23=F47+F16+F156;
FG24=F46;
FG25=F191+F192+F193+F62+F79+F194+F195+F196+
     F197+F198+F199+F200+F228;
FG26=F152+F153+F157+F158+F159+F161+F162+F163+F233+F234;
FG27=F142+F143+F160;
FG28=F95+F96+F97;
FG29=F118+F119+F120+F121+F164+F219;
FG30=F209+F210;
FG31=F168+F169+F170+F171+F183+F184+F220;
FG32=F172+F173+F174+F175+F176;
FG33=F207+F208+F239;
FG34=F248+F249;
FG35=F250;
FG36=F251;
FG37=F204+F205;
FG38=F253+F254;
FG39=F201;
FG40=F245+F246+F247;
FG41=F241+F242+F243+F244;
FG42=F206+F178+F179+F180+F223;
FG43=F18+F19+F20+F21+F212+F213;
FG44=F93;
FG45=F94;
FG46=F181+F188+F189+F224;
FG47=F63+F64+F77+F78+F230;
FG48=F150+F151;
FG49=F214+F42;
FG50=F235+F165+F166+F167;
FG51=F106+F107+F111+F102;
FG52=F98+F99;
FG53=F104+F105;
FG54=F110+F112+F114+F115+F116;
FG55=F182+F117+F100+F101+F103+F108+F109+F113+F122;
RUN;
```

BROAD FOOD GROUPS

The 55 food groups were collapsed into 10 broad food groups, as follows:

MEAT	*FG8 + FG9 + FG10+ FG11+ FG12 + FG13 +FG43*
FISH	*FG6 + FG7 + F172*
DAIRY	*FG2 + FG4 + FG5 + FG27 + F175*
LEGUMES	*FG21*
VEGETABLES	*FG14 + FG15 + FG16 + FG17 + FG18 + FG19 + FG20 + FG31+ FG44 + FG45*
CEREALS	*FG22 + FG23 + FG24 + FG25 + FG26 + FG48*
FRUITS	*FG28 + FG29 + FG51 + FG52 + FG53 + FG54 + FG55*
SUGAR PRODUCTS	*FG30 + FG33 + FG42 + FG50*
FAT	*FG40 + FG41 + FG46*
ALCOHOL	*FG34 + FG35 + FG36*

The 10 braod food groups were further collapsed into 2 broad food groups as follows:

ANIMAL	*meat + fish + dairy*
PLANT	*legumes + vegetables + fruit + cereals*

Sugar products, alcohol and fats were kept separate.

APPENDIX 10

TRADITIONAL FOODS & SCORE

F6	Quail
F7	Lamb
F10	Rabbit casserole 'Stifado'
F16	Spaghetti pie 'Pastichio'
F17	Eggplant pie 'Moussaka'
F22	Goat 'Katsikaki'
F26	Tripe 'Patsa'
F29	Cod 'Bakaliaros'
F30	Herring 'Renga'
F32	Whitebait 'Marida'
F35	Squid 'Kalamari'
F36	Octopus 'Htapodaki'
F40	Spinach pie 'Spanakopita'
F41	Cheese pie 'Tiropita'
F43	Spinach and rice dish 'Spanakorizo'
F44	Leeks and rice dish 'Prasorizo'
F45	Cabbage and rice dish 'Lahanorizo'
F47	Pasta
F48	Green bean casserole 'Fasolakia ladera'
F49	Eggplant casserole 'Melitzanes laderes'
F50	Stuffed vegetables 'Gemista'
F51	Okra casserole 'Bamies laderes'
F52	Artichoke casserole 'Aginares avgolemono'
F53	Ratatouille 'Briam'
F54	Cabbage dolmas 'Lahanodolmades avgolemono'
F55	Broad bean salad 'Koukia'
F56	Lima bean casserole 'Gigandes plaki'
F57	Lentil soup 'Fakes'
F58	Chickpea soup 'Revithia'
F59	Haricot bean soup 'Fasolada'
F60	Split pea soup 'Fava'
F62	Semolina and yoghurt soup 'Trahana soupa'
F63	Fish soup 'Psarosoupa'
F64	Chicken soup 'Kotosoupa'
F76	Chickory/Endive boiled 'Radiki/Antidi/Horta'
F77	Meat soup 'Kreatosoupa'
F81	Wild greens 'Agria horta'
F90	Vine leaf dolmas 'Dolmadakia avgolemono'
F91	Capsicum grilled/pickled 'Piperies'
F93	Garlic 'Skordo'
F94	Olives 'Elies'
F108	Grapes 'Stafilia'
F109	Figs 'Sika'
F123	Feta cheese 'Feta'
F126	Kaseri cheese 'Kaseri'
F129	Kefalotiri cheese 'Kefalotiri'
F138	Yoghurt 'Yaourti'
F158	Pastry with honey and nuts 'Baklava'
F160	Pastry and custard 'Galaktobureko'
F164	Fruit in syrup 'Gliko tou koutaliou'
F165	Halva
F168	Almonds 'Amigdala'
F169	Roastedchickpeas 'Stragalia'
F171	Chestnuts 'Kastana'
F172	Fish roe dip 'Taramosalata'
F173	Potato dip 'Skordalia'
F174	Eggplant dip 'Melitzanosalata'
F175	Yoghurt dip 'Tzatziki'
F177	Foulia (a type of broad bean)
F187	Buttermilk 'Xinogalo'
F200	Polenta 'Bobota'
F202	Greek coffee 'Elinikos kafes'
F205	Herb tea 'Tsai tou vounou'
F224	Tahini paste 'Tahini'
F228	Cracked wheat 'Sitari kofto'
F229	Mung bean soup 'Fava'
F237	Souvlaki
F241	Olive oil 'Eleolado'
F250	Wine 'Krasi'

Programme written in SAS to calculate traditional foods score

```
data grk.spfdscr;
set grk.spfdscr; tradscr=0;
if f6 >0 then tradscr=tradscr+1;
if f7 >0 then tradscr=tradscr+1;
if f10 >0 then tradscr=tradscr+1;
if f16 >0 then tradscr=tradscr+1;
if f17 >0 then tradscr=tradscr+1;
if f22 >0 then tradscr=tradscr+1;
if f26 >0 then tradscr=tradscr+1;
if f29 >0 then tradscr=tradscr+1;
if f30 >0 then tradscr=tradscr+1;
if f32 >0 then tradscr=tradscr+1;
if f35 >0 then tradscr=tradscr+1;
if f36 >0 then tradscr=tradscr+1;
if f40 >0 then tradscr=tradscr+1;
if f41 >0 then tradscr=tradscr+1;
if f43 >0 then tradscr=tradscr+1;
if f44 >0 then tradscr=tradscr+1;
if f45 >0 then tradscr=tradscr+1;
if f47 >0 then tradscr=tradscr+1;
if f48 >0 then tradscr=tradscr+1;
if f49 >0 then tradscr=tradscr+1;
if f50 >0 then tradscr=tradscr+1;
if f51 >0 then tradscr=tradscr+1;
if f52 >0 then tradscr=tradscr+1;
if f53 >0 then tradscr=tradscr+1;
if f54 >0 then tradscr=tradscr+1;
if f55 >0 then tradscr=tradscr+1;

if f56 >0 then tradscr=tradscr+1;
if f57 >0 then tradscr=tradscr+1;
if f58 >0 then tradscr=tradscr+1;
if f59 >0 then tradscr=tradscr+1;
if f60 >0 then tradscr=tradscr+1;
if f61 >0 then tradscr=tradscr+1;
if f62 >0 then tradscr=tradscr+1;
if f63 >0 then tradscr=tradscr+1;
if f64 >0 then tradscr=tradscr+1;
if f76 >0 then tradscr=tradscr+1;
if f77 >0 then tradscr=tradscr+1;
if f81 >0 then tradscr=tradscr+1;
if f90 >0 then tradscr=tradscr+1;
if f91 >0 then tradscr=tradscr+1;
if f93 >0 then tradscr=tradscr+1;
if f94 >0 then tradscr=tradscr+1;
if f108 >0 then tradscr=tradscr+1;
if f109 >0 then tradscr=tradscr+1;
if f123 >0 then tradscr=tradscr+1;
if f126 >0 then tradscr=tradscr+1;
if f129 >0 then tradscr=tradscr+1;
if f138 >0 then tradscr=tradscr+1;
if f158 >0 then tradscr=tradscr+1;
if f160 >0 then tradscr=tradscr+1;
        if f164 >0 then tradscr=tradscr+1;
        if f165 >0 then tradscr=tradscr+1;
        if f168 >0 then tradscr=tradscr+1;
        if f169 >0 then tradscr=tradscr+1;
        if f171 >0 then tradscr=tradscr+1;
        if f172 >0 then tradscr=tradscr+1;
        if f173 >0 then tradscr=tradscr+1;
        if f174 >0 then tradscr=tradscr+1;
        if f175 >0 then tradscr=tradscr+1;
        if f177 >0 then tradscr=tradscr+1;
        if f187 >0 then tradscr=tradscr+1;
        if f200 >0 then tradscr=tradscr+1;
        if f201 >0 then tradscr=tradscr+1;
        if f205 >0 then tradscr=tradscr+1;
        if f224 >0 then tradscr=tradscr+1;
        if f228 >0 then tradscr=tradscr+1;
        if f229 >0 then tradscr=tradscr+1;
        if f237 >0 then tradscr=tradscr+1;
        if f241 >0 then tradscr=tradscr+1;
        if f250 >0 then tradscr=tradscr+1;
        run;
```

APPENDIX 11

FOOD GROUP VARIETY SCORES

Programme written in SAS to compute the food group variety scores.

```
data grk.spfdscr; set grk.spfdgrp;
drop fg1--fg55; meatscr=0; fishscr=0; dairyscr=0; cerealsc=0;
legscr=0; vegscr=0; fruitscr=0; sweetscr=0; otherscr=0;
FCHICK=F1+F2+F3+F4;
FLAMB=F7+F8+F9;
FBEEF=F11+F12+F13+F14+F15;
fchick=fchick*30;if fchick>180 then meatscr=meatscr+1;
flamb=flamb*30;if flamb>=180 then meatscr=meatscr+1;
fbeef=fbeef*30;if fbeef>=120 then meatscr=meatscr+1;
f5=f5*30; if f5>=120 then meatscr=meatscr+1;
f6=f6*30; if f6>=120 then meatscr=meatscr+1;
f10=f10*30; if f10>=170 then meatscr=meatscr+1;
f18=f18*30; if f18>=30 then meatscr=meatscr+1;
f19=f19*30; if f19>=30 then meatscr=meatscr+1;
f20=f20*30; if f20>=30 then meatscr=meatscr+1;
f21=f21*30; if f21>=30 then meatscr=meatscr+1;
f22=f22*30; if f22>=120 then meatscr=meatscr+1;
f23=f23*30; if f23>=120 then meatscr=meatscr+1;
f24=f24*30; if f24>=120 then meatscr=meatscr+1;
f25=f25*30; if f25>=120 then meatscr=meatscr+1;
f26=f26*30; if f26>=250 then meatscr=meatscr+1;
f64=f64*30; if f64>=250 then meatscr=meatscr+1;
f77=f77*30; if f77>=250 then meatscr=meatscr+1;
f231=f231*30; if f231>=150 then meatscr=meatscr+1;
f232=f232*30; if f232>=150 then meatscr=meatscr+1;
f236=f236*30; if f236>=150 then meatscr=meatscr+1;
f237=f237*30; if f237>=150 then meatscr=meatscr+1;
f212=f212*30; if f212>=30 then meatscr=meatscr+1;
f213=f213*30; if f213>=20 then meatscr=meatscr+1;
f230=f230*30; if f230>=200 then meatscr=meatscr+1;

FFISH=F27+F28;
ffish=ffish*30;if ffish>=190 then fishscr=fishscr+1;
f29=f29*30; if f29>=120 then fishscr=fishscr+1;
f30=f30*30; if f30>=100 then fishscr=fishscr+1;
f32=f32*30; if f32>=120 then fishscr=fishscr+1;
f33=f33*30; if f33>=60 then fishscr=fishscr+1;
f34=f34*30; if f34>=30 then fishscr=fishscr+1;
f35=f35*30; if f35>=120 then fishscr=fishscr+1;
f36=f36*30; if f36>=60 then fishscr=fishscr+1;
f37=f37*30; if f37>=120 then fishscr=fishscr+1;
f38=f38*30; if f38>=30 then fishscr=fishscr+1;
f39=f39*30; if f39>=30 then fishscr=fishscr+1;
f63=f63*30; if f63>=250 then fishscr=fishscr+1;
f155=f155*30; if f155>=10 then fishscr=fishscr+1;
```

```
f154=f154*30; if f154>=100 then fishscr=fishscr+1;
f172=f172*30; if f172>=20 then fishscr=fishscr+1;
f185=f185*30; if f185>=100 then fishscr=fishscr+1;
f186=f186*30; if f186>=65 then fishscr=fishscr+1;
f221=f221*30; if f221>=60 then fishscr=fishscr+1;
f222=f222*30; if f222>=60 then fishscr=fishscr+1;

f41=f41*30; if f41>=150 then dairyscr=dairyscr+1;
f123=f123*30; if f123>=50 then dairyscr=dairyscr+1;
f124=f124*30; if f124>=50 then dairyscr=dairyscr+1;
f125=f125*30; if f125>=50 then dairyscr=dairyscr+1;
f126=f126*30; if f126>=50 then dairyscr=dairyscr+1;
f128=f128*30; if f128>=50 then dairyscr=dairyscr+1;
f129=f129*30; if f129>=50 then dairyscr=dairyscr+1;
f130=f130*30; if f130>=30 then dairyscr=dairyscr+1;
f131=f131*30; if f131>=200 then dairyscr=dairyscr+1;
f132=f132*30; if f132>=200 then dairyscr=dairyscr+1;
f133=f133*30; if f133>=200 then dairyscr=dairyscr+1;
f134=f134*30; if f134>=50 then dairyscr=dairyscr+1;
f135=f135*30; if f135>=50 then dairyscr=dairyscr+1;
f136=f136*30; if f136>=200 then dairyscr=dairyscr+1;
f137=f137*30; if f137>=200 then dairyscr=dairyscr+1;
f138=f138*30; if f138>=200 then dairyscr=dairyscr+1;
f139=f139*30; if f139>=200 then dairyscr=dairyscr+1;
f140=f140*30; if f140>=200 then dairyscr=dairyscr+1;
f141=f141*30; if f141>=200 then dairyscr=dairyscr+1;
f142=f142*30; if f142>=200 then dairyscr=dairyscr+1;
f143=f143*30; if f143>=30 then dairyscr=dairyscr+1;
f144=f144*30; if f144>=110 then dairyscr=dairyscr+1;
f160=f160*30; if f160>=150 then dairyscr=dairyscr+1;
f175=f175*30; if f175>=20 then dairyscr=dairyscr+1;
f187=f187*30; if f187>=200 then dairyscr=dairyscr+1;
f211=f211*30; if f211>=200 then dairyscr=dairyscr+1;
f214=f214*30; if f214>=200 then dairyscr=dairyscr+1;
f225=f225*30; if f225>=10 then dairyscr=dairyscr+1;
f226=f226*30; if f226>=30 then dairyscr=dairyscr+1;
f227=f227*30; if f227>=30 then dairyscr=dairyscr+1;

f16=f16*30; if 16>=300 then cerealsc=cerealsc+1;
f42=f42*30; if f42>=150 then cerealsc=cerealsc+1;
f46=f46*30; if f46>=200 then cerealsc=cerealsc+1;
f47=f47*30; if f47>=200 then cerealsc=cerealsc+1;
f62=f62*30; if f62>=250 then cerealsc=cerealsc+1;
f79=f79*30; if f79>=200 then cerealsc=cerealsc+1;
f145=f145*30; if f145>=100 then cerealsc=cerealsc+1;
f146=f146*30; if f146>=100 then cerealsc=cerealsc+1;
f147=f147*30; if f147>=100 then cerealsc=cerealsc+1;
f148=f148*30; if f148>=100 then cerealsc=cerealsc+1;
f149=f149*30; if f149>=100 then cerealsc=cerealsc+1;
f150=f150*30; if f150>=30 then cerealsc=cerealsc+1;
f151=f151*30; if f151>=30 then cerealsc=cerealsc+1;
f152=f152*30; if f152>=30 then cerealsc=cerealsc+1;
f153=f153*30; if f153>=10 then cerealsc=cerealsc+1;
f156=f156*30; if f156>=100 then cerealsc=cerealsc+1;
f157=f157*30; if f157>=50 then cerealsc=cerealsc+1;
f158=f158*30; if f158>=100 then cerealsc=cerealsc+1;
f159=f159*30; if f159>=90 then cerealsc=cerealsc+1;
f161=f161*30; if f161>=40 then cerealsc=cerealsc+1;
f162=f162*30; if f162>=50 then cerealsc=cerealsc+1;
```

```
f163=f163*30; if f163>=50 then cerealsc=cerealsc+1;
f190=f190*30; if f190>=55 then cerealsc=cerealsc+1;
f191=f191*30; if f191>=30 then cerealsc=cerealsc+1;
f192=f192*30; if f192>=30 then cerealsc=cerealsc+1;
f193=f193*30; if f193>=30 then cerealsc=cerealsc+1;
f194=f194*30; if f194>=30 then cerealsc=cerealsc+1;
f195=f195*30; if f195>=30 then cerealsc=cerealsc+1;
f196=f196*30; if f196>=30 then cerealsc=cerealsc+1;
f197=f197*30; if f197>=150 then cerealsc=cerealsc+1;
f198=f198*30; if f198>=30 then cerealsc=cerealsc+1;
f199=f199*30; if f199>=10 then cerealsc=cerealsc+1;
f200=f200*30; if f200>=200 then cerealsc=cerealsc+1;
f228=f228*30; if f228>=100 then cerealsc=cerealsc+1;

f55=f55*30; if f55>=200 then legscr=legscr+1;
f56=f56*30; if f56>=200 then legscr=legscr+1;
f57=f57*30; if f57>=300 then legscr=legscr+1;
f58=f58*30; if f58>=280 then legscr=legscr+1;
f59=f59*30; if f59>=300 then legscr=legscr+1;
f60=f60*30; if f60>=200 then legscr=legscr+1;
f61=f61*30; if f61>=200 then legscr=legscr+1;
f85=f85*30; if f85>=80 then legscr=legscr+1;
f177=f177*30; if f177>=200 then legscr=legscr+1;
f217=f217*30; if f217>=200 then legscr=legscr+1;
f218=f218*30; if f218>=200 then legscr=legscr+1;
f229=f229*30; if f229>=200 then legscr=legscr+1;
f238=f238*30; if f238>=100 then legscr=legscr+1;

f17=f17*30; if f17>=300 then vegscr=vegscr+1;
f40=f40*30; if f40>=150 then vegscr=vegscr+1;
f43=f43*30; if f43>=200 then vegscr=vegscr+1;
f44=f44*30; if f44>=200 then vegscr=vegscr+1;
f45=f45*30; if f45>=200 then vegscr=vegscr+1;
f48=f48*30; if f48>=300 then vegscr=vegscr+1;
f49=f49*30; if f49>=250 then vegscr=vegscr+1;
f50=f50*30; if f50>=250 then vegscr=vegscr+1;
f51=f51*30; if f51>=300 then vegscr=vegscr+1;
f52=f52*30; if f52>=250 then vegscr=vegscr+1;
f53=f53*30; if f53>=300 then vegscr=vegscr+1;
f54=f54*30; if f54>=300 then vegscr=vegscr+1;
fpotato=fpotato*30; if fpotato>=200 then vegscr=vegscr+1;
f68=f68*30; if f68>=100 then vegscr=vegscr+1;
f69=f69*30; if f69>=50 then vegscr=vegscr+1;
f70=f70*30; if f70>=70 then vegscr=vegscr+1;
f71=f71*30; if f71>=100 then vegscr=vegscr+1;
f72=f72*30; if f72>=180 then vegscr=vegscr+1;
f73=f73*30; if f73>=100 then vegscr=vegscr+1;
f74=f74*30; if f74>=95 then vegscr=vegscr+1;
f75=f75*30; if f75>=200 then vegscr=vegscr+1;
f76=f76*30; if f76>=200 then vegscr=vegscr+1;
f78=f78*30; if f78>=200 then vegscr=vegscr+1;
f80=f80*30; if f80>=100 then vegscr=vegscr+1;
f81=f81*30; if f81>=200 then vegscr=vegscr+1;
f82=f82*30; if f82>=200 then vegscr=vegscr+1;
f83=f83*30; if f83>=100 then vegscr=vegscr+1;
f84=f84*30; if f84>=100 then vegscr=vegscr+1;
f86=f86*30; if f86>=30 then vegscr=vegscr+1;
f87=f87*30; if f87>=100 then vegscr=vegscr+1;
f88=f88*30; if f88>=100 then vegscr=vegscr+1;
```

```
f89=f89*30; if f89>=60 then vegscr=vegscr+1;
f90=f90*30; if f90>=180 then vegscr=vegscr+1;
f91=f91*30; if f91>=60 then vegscr=vegscr+1;
f92=f92*30; if f92>=60 then vegscr=vegscr+1;
f93=f93*30; if f93>=2 then vegscr=vegscr+1;
f94=f94*30; if f94>=20 then vegscr=vegscr+1;
f168=f168*30; if f168>=15 then vegscr=vegscr+1;
f169=f169*30; if f169>=15 then vegscr=vegscr+1;
f171=f171*30; if f171>=20 then vegscr=vegscr+1;
f173=f173*30; if f173>=20 then vegscr=vegscr+1;
f174=f174*30; if f174>=20 then vegscr=vegscr+1;
f176=f176*30; if f176>=20 then vegscr=vegscr+1;
f183=f183*30; if f183>=15 then vegscr=vegscr+1;
f184=f184*30; if f184>=15 then vegscr=vegscr+1;
f215=f215*30; if f215>=100 then vegscr=vegscr+1;
f216=f216*30; if 216>=50 then vegscr=vegscr+1;
f220=f220*30; if f220>=15 then vegscr=vegscr+1;

f95=f95*30; if f95>=130 then fruitscr=fruitscr+1;
f96=f96*30; if f96>=200 then fruitscr=fruitscr+1;
f97=f97*30; if f97>=100 then fruitscr=fruitscr+1;
f98=f98*30; if f98>=100 then fruitscr=fruitscr+1;
f99=f99*30; if f99>=120 then fruitscr=fruitscr+1;
f100=f100*30; if f100>=100 then fruitscr=fruitscr+1;
f101=f101*30; if f101>=100 then fruitscr=fruitscr+1;
f102=f102*30; if f102>=100 then fruitscr=fruitscr+1;
f103=f103*30; if f103>=100 then fruitscr=fruitscr+1;
f104=f104*30; if f104>=120 then fruitscr=fruitscr+1;
f105=f105*30; if f105>=260 then fruitscr=fruitscr+1;
f106=f106*30; if f106>=120 then fruitscr=fruitscr+1;
f107=f107*30; if f107>=100 then fruitscr=fruitscr+1;
f108=f108*30; if f108>=100 then fruitscr=fruitscr+1;
f109=f109*30; if f109>=100 then fruitscr=fruitscr+1;
f110=f110*30; if f110>=80 then fruitscr=fruitscr+1;
f111=f111*30; if f111>=100 then fruitscr=fruitscr+1;
f112=f112*30; if f112>=150 then fruitscr=fruitscr+1;
f113=f113*30; if f113>=60 then fruitscr=fruitscr+1;
f114=f114*30; if f114>=70 then fruitscr=fruitscr+1;
f115=f115*30; if f115>=100 then fruitscr=fruitscr+1;
f116=f116*30; if f116>=30 then fruitscr=fruitscr+1;
f117=f117*30; if f117>=100 then fruitscr=fruitscr+1;
f118=f118*30; if f118>=50 then fruitscr=fruitscr+1;
f119=f119*30; if f119>=50 then fruitscr=fruitscr+1;
f120=f120*30; if f120>=50 then fruitscr=fruitscr+1;
f121=f121*30; if f121>=35 then fruitscr=fruitscr+1;
f122=f122*30; if f122>=100 then fruitscr=fruitscr+1;
f182=f182*30; if f182>=50 then fruitscr=fruitscr+1;
f209=f209*30; if f209>=200 then fruitscr=fruitscr+1;
f210=f210*30; if f210>=200 then fruitscr=fruitscr+1;
f219=f219*30; if f219>=50 then fruitscr=fruitscr+1;
f233=f233*30; if f233>=100 then fruitscr=fruitscr+1;

f164=f164*30; if f164>=20 then sweetscr=sweetscr+1;
f165=f165*30; if f165>=80 then sweetscr=sweetscr+1;
f166=f166*30; if f166>=30 then sweetscr=sweetscr+1;
f167=f167*30; if f167>=10 then sweetscr=sweetscr+1;
f178=f178*30; if f178>=20 then sweetscr=sweetscr+1;
f179=f179*30; if f179>=20 then sweetscr=sweetscr+1;
f180=f180*30; if f180>=20 then sweetscr=sweetscr+1;
```

```
f181=f181*30; if f181>=20 then sweetscr=sweetscr+1;
f206=f206*30; if f206>=20 then sweetscr=sweetscr+1;
f207=f207*30; if f207>=200 then sweetscr=sweetscr+1;
f208=f208*30; if f208>=200 then sweetscr=sweetscr+1;
f209=f209*30; if f209>=200 then sweetscr=sweetscr+1;
f210=f210*30; if f210>=200 then sweetscr=sweetscr+1;
f223=f223*30; if f223>=20 then sweetscr=sweetscr+1;
f234=f234*30; if f234>=50 then sweetscr=sweetscr+1;
f235=f235*30; if f235>=50 then sweetscr=sweetscr+1;
f239=f239*30; if f239>=200 then sweetscr=sweetscr+1;

f188=f188*30; if f188>=10 then otherscr=otherscr+1;
f189=f189*30; if f189>=10 then otherscr=otherscr+1;
f201=f201*30; if f201>=50 then otherscr=otherscr+1;
f202=f202*30; if f202>=50 then otherscr=otherscr+1;
f203=f203*30; if f203>=200 then otherscr=otherscr+1;
f204=f204*30; if f204>=200 then otherscr=otherscr+1;
f205=f205*30; if f205>=200 then otherscr=otherscr+1;
f224=f224*30; if f224>=10 then otherscr=otherscr+1;
f240=f240*30; if f240>=200 then otherscr=otherscr+1;
f241=f241*30; if f241>=20 then otherscr=otherscr+1;
f242=f242*30; if f242>=20 then otherscr=otherscr+1;
f243=f243*30; if f243>=20 then otherscr=otherscr+1;
f244=f244*30; if f244>=20 then otherscr=otherscr+1;
f245=f245*30; if f245>=10 then otherscr=otherscr+1;
f246=f246*30; if f246>=10 then otherscr=otherscr+1;
f247=f247*30; if f247>=10 then otherscr=otherscr+1;
f248=f248*30; if f248>=200 then otherscr=otherscr+1;
f249=f249*30; if f249>=200 then otherscr=otherscr+1;
f250=f250*30; if f250>=100 then otherscr=otherscr+1;
f251=f251*30; if f251>=30 then otherscr=otherscr+1;
varscr=meatscr+dairyscr+fishscr+cerealsc+vegscr+legscr+
fruitscr+sweetscr+otherscr;
run;
proc print data=grk.spfdscr; var varscr meatscr--otherscr; run;
```

APPENDIX 12

STATISTICAL ANALYSES

Programme written in SAS to perform non-parametric statistics WILCOXON test on continuous variables.

```
proc sort data=grk.splifdis out=grk.splifdi1; by agegrp;
proc npar1way wilcoxon data=grk.splifdi1; class sex; by agegrp;
var ex86sc; run;
proc sort data=grk.splifdis out=grk.splifdi2; by sex;
proc npar1way wilcoxon data=grk.splifdi2; class agegrp; by sex;
var ex86sc; run;

proc sort data=grk.oglifdis out=grk.oglifdi1; by agegrp;
proc npar1way wilcoxon data=grk.oglifdi1; class sex; by agegrp;
var ex86sc; run;
proc sort data=grk.oglifdis out=grk.oglifdi2; by sex;
proc npar1way wilcoxon data=grk.oglifdi2; class agegrp; by sex;
var fgmeats; run;

proc sort data=grk.oglifdis; by id; run;
proc sort data=grk.splifdis; by id; run;
data grk.lifdis; merge grk.splifdis grk.oglifdis; by id; run;
proc sort data=grk.lifdis out=grk.lifdis1; by sex agegrp; run;
proc npar1way wilcoxon data=grk.lifdis1; class centre; by sex agegrp;
var ex86sc; run;
```

Programme written in SAS to perform CHI-SQUARE test on discrete variables.

```
proc sort data=grk.sppast out=grk.sppast1;
by agegrp; run;
proc freq data=grk.sppast1; tables chickrst*sex/all;
by agegrp; run;
proc sort data=grk.sppast out=grk.sppast2;
by sex; run;
proc freq data=grk.sppast2; tables chickrst*agegrp/all;
by sex; run;

proc sort data=grk.ogpast out=grk.ogpast1;
by agegrp; run;
proc freq data=grk.ogpast1; tables chickrst*sex/all;
by agegrp; run;
proc sort data=grk.ogpast out=grk.ogpast2;
by sex; run;
proc freq data=grk.ogpast2; tables chickrst*agegrp/all;
by sex; run;
proc sort data=grk.sppast; by id; run;
proc sort data=grk.ogpast; by id; run;
data grk.past; merge grk.sppast grk.ogpast; by id; run;
proc sort data=grk.past out=grk.past1; by agegrp sex; run;
proc freq data=grk.past1; tables chickrst*centre/all;
by agegrp sex;
run;
```