Analytic Guidelines: Creating Disability Identifiers Using the Washington Group Extended Set on Functioning (WG-ES) Stata Syntax

Introduction

As with the WG Short Set on Functioning (WG-SS), analysis of the WG Extended Set on Functioning (WG-ES) can also produce multiple disability identifiers based on the choice of the severity threshold or cut-off. The Stata syntax below, however, provides for the calculation of disability identifiers using different sets of WG-ES domains using the recommended cut-off for international comparisons (described below).

For each of the disability identifiers described, the level of inclusion is at least one domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL – or – for the domains Anxiety, Depression, Pain and Fatigue, the highest level of difficulty on a four-point scale.

Each of the four disability identifiers described in this document is defined based on the choice of domains of functioning included:

WG-SS: *Short Set*: 6 domains, 6 questions.

WG-ES 1: *Extended Set*: 11 domains, 25 questions.

WG-ES 2: *Modified Extended Set* (WG-ES MINUS Pain and Fatigue): 9 domains, 20 questions.

WG-ES 3: *Short Set Enhanced* (WG-SS PLUS Upper body, Anxiety and Depression): 9 domains, 12 questions.

NOTE: For data analysis, use your standard weighting and estimation techniques.

The **Washington Group Implementation Documents** cover the tools developed by the Washington Group on Disability Statistics (WG) to collect internationally comparable disability data on censuses and surveys. The documents address best practices in implementing the Short Set, Extended Set, Short Set – Enhanced, the WG / UNICEF Child Functioning Modules for children 2-4 and 5-17 years of age, and the WG / ILO LFS Disability Module, as well as other WG tools. Topics include translation, question specifications, analytic guidelines, programming code for analyses, the use of the tools for the purposes of disaggregation, and more.

To locate other WG Implementation Documents and more information, visit the Washington Group website: http://www.washingtongroup-disability.com/.

The Stata syntax is based on the *variable labels* indicated in the table below. The complete WG-ES module includes more questions than appear in this table. Disability status is determined through difficulty in the basic, universal activities *without* the use of assistive technology or other assistance. There are several mobility questions, for example, that reference difficulty walking *with* the use of

assistance. Those questions are not included in the analytic plan provided here; however, they can be used in other analyses that look more closely into the effect of assistive technology (environmental facilitators) on functioning.

Only those questions/variables below are used in the determination of disability identifiers.
Ensure that you use the same *variable labels* OR revise the Stata syntax to reflect the *variable labels* in your database.

The WG-SS is administered as part of the U.S. National Health Interview Survey (NHIS). The data used to prepare these guidelines come from the 2013 NHIS.

***Note to users of the NHIS**: the variable names in the NHIS data file and documentation may differ from those used in this document; e.g., the self-care domain variable referenced as SC-SS in this document is referred to as UB_SS in the NHIS data file and documentation.*

WG Extended Set Questions/Domains	Variable Label	Response Pattern
VISION		
1. Do you have difficulty seeing even if wearing glasses?	VIS_SS	1
COMMUNICATION		
2. Using your usual language, do you have difficulty communicating (for example understanding or being understood by others)?	COM_SS	1
HEARING		
3. Do you have difficulty hearing even if using a hearing aid?	HEAR_SS	1
4. Difficulty hearing conversation with one person in quiet room?	HEAR_3	1
5. Difficulty hearing conversation with one person in noisier room?	HEAR_4	1
COGNITION		
6. Do you have difficulty remembering or concentrating?	COG_SS	1
7. Difficulty remembering, concentrating, or both?	COG_1	2
8. How often have difficulty remembering?	COG_2	3
9. Amount of things you have difficulty remembering?	COG_3	4
SELF-CARE / UPPER BODY		
10. Do you have difficulty with (self-care such as) washing all over or dressing?	SC_SS	1
11. Difficulty raising 2 liter bottle of water from waist to eye level?	UB_1	1
12. Degree of difficulty using hands and fingers	UB_2	1

MOBILITY		
13. Do you have difficulty walking or climbing stairs?	MOB_SS	1
14. Difficulty walking 100 yards on level ground without aid or equipment?	MOB_4	1
15. Difficulty walking 1/3rd mile on level ground without aid or equipment	MOB_5	1
16. Difficulty walking up or down 12 steps without aid or equipment?	MOB_6	1
AFFECT (ANXIETY)		
17. How often feel worried, nervous, or anxious?	ANX_1	5
18. Level of feelings when last felt worried, nervous, or anxious?	ANX_3	6
AFFECT (DEPRESSION)		
19. How often do you feel depressed?	DEP_1	5
20. How depressed you felt last time you were depressed?	DEP_3	6
PAIN		
21. Frequency of pain in past 3 months?	PAIN_2	7
22. How much pain you had last time you had pain?	PAIN_4	6
FATIGUE		
23. How often felt very tired or exhausted in past 3 months?	TIRED_1	7
24. How long most recent tired or exhausted feelings lasted?	TIRED_2	8
25. Level of tiredness last time felt very tired or exhausted?	TIRED_3	6

NOTE: **Red** refers to the Washington Group Short Set (**WG-SS**).
All 25 questions are included in **WG-ES 1**.

Red plus **Blue** plus **Green** questions are included in **WG-ES 2**.

Red plus **Green** questions are included in **WG-ES 3**.

Response patterns:

	Pattern 1	Pattern 2	Pattern 3	Pattern 4
1	No difficulty	Difficulty remembering only	Sometimes	A few things
2	Yes, Some difficulty	Difficulty concentrating only	Often	A lot of things
3	Yes, A lot of difficulty	Difficulty both remembering & concentrating	All of the time	Almost everything
4	Cannot do at all			
7	Refused			
8	Not ascertained			
9	Don't know			

	Pattern 5	Pattern 6*	Pattern 7	Pattern 8
1	Daily	A little	Never	Some of the day
2	Weekly	A lot	Some days	Most of the day
3	Monthly	Somewhere in between a little and a lot	Most days	All of the day
4	A few times a year		Every day	
5	Never			
7	Refused			
8	Not ascertained			
9	Don't know			

*** IN THE SYNTAX BELOW, NOTE THAT ITEMS WITH RESPONSE PATTERN 6 (ANX_3, DEP_3, PAIN_4 AND TIRED_3) ARE RECODED TO PLACE "SOMEWHERE BETWEEN" NUMERICALLY IN-BETWEEN "A LITTLE" AND "A LOT".**

The WG-SS is embedded within the WG-ES.
The WG-ES is supplemented with:

- additional questions to those existing 6 domains and
- additional domains (several with multiple questions).

The Stata syntax presented below includes a couple of elements that were particular to the content of the WG-ES.

First, it was important to determine single domain-specific identifiers for those domains of functioning that included multiple questions. For example, upper body functioning includes two questions, each eliciting specific and unique actions: difficulty raising a bottle of water from waist to eye level (arms/shoulders), and difficulty using hands and fingers. Those two questions were analyzed and combined to produce a single upper body indicator with four levels of difficulty ranging from 1 - low

difficulty to 4 - high difficulty – not unlike the categorical responses to the single WG-SS questions: no difficulty, some difficulty, a lot of difficulty and cannot do at all. As with the upper body domain, the WG-ES domains cognition, anxiety, depression, pain and fatigue have different response patterns that do not readily 'translate' into the usual WG response pattern. For these domains of functioning, a similar 4-scale response pattern was produced and annotated as level 1 through 4, where 1 is the lowest level of difficulty and 4 is the highest.

Second, individual domain indicators were assessed together to determine the appropriate cut-off for inclusion into an overall disability identifier – for the purposes of estimating prevalence and disaggregating outcome indicators by disability status.

NOTE:

For all variables, codes (7) *Refused*, (8) *Not Ascertained*, and (9) *Don't know*, are recoded to *Missing*.

Stata WG Extended Set Syntax Annotated with Output Tables

Actual Stata syntax is indented and are in Bold text.

NOTE: For data analysis, use your standard weighting and estimation techniques.

The syntax below produces **frequency distributions** on individual domain questions – **cross-tabulations** on multiple domain questions, and calculates INDICATOR variables for domains with multiple questions – for use in the determination of disability identifiers.

VISION

Step 1. Generate frequency distribution for Vision domain.

VIS_SS is the WG-SS Vision question.

```
gen Vision=VIS_SS if inlist(VIS_SS, 1,2,3,4)
replace Vision=. if inlist(VIS_SS, 7,8,9)
tabulate Vision
```

Vision: Degree of difficulty seeing

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13690	79.0	81.6	81.6
	Some difficulty	2708	15.6	16.2	97.8
	A lot of difficulty	333	1.9	2.0	99.8
	Cannot do at all	36	.2	.2	100.0
	Total	16767	96.8	100.0	
Missing		559	3.2		
Total		17326	100.0		

COMMUNICATION

Step 2. Generate frequency distribution for Communication domain.

COM_SS is the WG-SS Communication question.

```
gen Communication=COM_SS if inlist(COM_SS, 1,2,3,4)
replace Communication =. if inlist(COM_SS, 7,8,9)
tabulate Communication
```

Communication: Degree of difficulty communicating using usual language

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15874	91.6	94.7	94.7
	Some difficulty	745	4.3	4.4	99.2
	A lot of difficulty	94	.5	.6	99.7
	Cannot do at all	43	.2	.3	100.0
	Total	16756	96.7	100.0	
Missing		570	3.3		
Total		17326	100.0		

HEARING

Step 3. Generate frequency distributions and cross-tabulations for Hearing domain questions and determine Hearing Indicator

HEAR_SS is the WG-SS Hearing question.

```
gen Hearing=HEAR_SS if inlist(HEAR_SS, 1,2,3,4)
replace Hearing=. if inlist(HEAR_SS, 7,8,9)
```

HEAR_3 is *Difficulty hearing conversation with one person in quiet room*.

```
gen HEAR_3_R=HEAR_3 if inlist(HEAR_3, 1,2,3,4)
replace HEAR_3_R =. if inlist(HEAR_3, 7,8,9)
```

HEAR_4 is *Difficulty hearing one person in noisier room*.

```
gen HEAR_4_R=HEAR_4 if inlist(HEAR_4, 1,2,3,4)
replace HEAR_4_R=. if inlist(HEAR_4, 7,8,9)
tabulate Hearing
```

Hearing: Degree of difficulty hearing

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13680	79.0	81.6	81.6
	Some difficulty	2753	15.9	16.4	98.0
	A lot of difficulty	310	1.8	1.8	99.9
	Cannot do at all	23	.1	.1	100.0
	Total	16766	96.8	100.0	
Missing		560	3.2		
Total		17326	100.0		

tabulate HEAR_3_R

HEAR_3_R: Difficulty hearing conversation with one person in quiet room

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15249	88.0	91.0	91.0
	Some difficulty	1316	7.6	7.9	98.9
	A lot of difficulty	162	.9	1.0	99.9
	Cannot do at all	10	.1	.1	100.0
	Total	16737	96.6	100.0	
Missing		589	3.4		
Total		17326	100.0		

tabulate HEAR_4_R

HEAR_4_R: Difficulty hearing one person in noisier room

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	11697	67.5	70.0	70.0
	Some difficulty	4191	24.2	25.1	95.1
	A lot of difficulty	779	4.5	4.7	99.7
	Cannot do at all	48	.3	.3	100.0
	Total	16715	96.5	100.0	
Missing		611	3.6		
Total		17326	100.0		

Step 4. For Hearing questions, recode HEAR_3_R and HEAR_4_R to value 4 (cannot do at all) if Hear_SS is 4 (Cannot do at all).
The syntax below recodes HEAR_3_R and HEAR_4_R to 4 (cannot do at all) if Hear_SS is 4 (cannot do at all).

gen HEAR_3_X=HEAR_3_R

```
replace HEAR_3_X=4 if Hearing==4 & HEAR_3_R==.

gen HEAR_4_X=HEAR_4_R
replace HEAR_4_X=4 if Hearing==4 & HEAR_4_R==.
tabulate HEAR_3_X
```

HEAR_3_X: Difficulty hearing conversation with one person in quiet room

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15249	88.0	91.0	91.0
	Some difficulty	1316	7.6	7.9	98.8
	A lot of difficulty	162	.9	1.0	99.8
	Cannot do at all	33	.2	.2	100.0
	Total	16760	96.7	100.0	
Missing		566	3.3		
Total		17326	100.0		

```
tabulate HEAR_4_X
```

HEAR_4_X: Difficulty hearing one person in noisier room

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	11697	67.5	69.9	69.9
	Some difficulty	4191	24.2	25.0	94.9
	A lot of difficulty	779	4.5	4.7	99.6
	Cannot do at all	71	.4	.4	100.0
	Total	16738	96.6	100.0	
Missing		588	3.4		
Total		17326	100.0		

Step 5. Generate a cross-tabulation of the two Hearing Extended Set questions: HEAR_3_X and HEAR_4_X.

The syntax below produces a cross-tabulation of the two Extended Set questions: HEAR_3_X: *Difficulty hearing in a quiet room* and HEAR_4_X: *Difficulty hearing in a noisier room* to determine a single HEARING INDICATOR.

```
tabulate HEAR_4_X HEAR_3_X
```

HEAR_4_X (Difficulty hearing in a Nosier room)		HEAR_3_X : Difficulty hearing conversation with one person in quiet room				
		No difficulty	Some difficulty	A lot of difficulty	Cannot do at all	Total
	No difficulty	11603	94	0	0	11697
	Some difficulty	3373	809	8	0	4190
	A lot of difficulty	253	388	138	0	779
	Cannot do at all	8	24	16	23	71
Total		15237	1315	162	23	16737

Step 6. Create a HEARING INDICATOR (H_INDICATOR) based on the two additional hearing questions HEAR_3_X and HEAR_4_X.

The syntax below creates a HEARING INDICATOR (H_INDICATOR) based on the cross-tabulation of the two additional hearing questions HEAR_3_X and HEAR_4_X.

```
gen H_INDICATOR=.
replace H_INDICATOR=1 if (HEAR_3_X==1 & HEAR_4_X==1) | ///
        (HEAR_3_X==1 & HEAR_4_X==2)
replace H_INDICATOR=2 if (HEAR_3_X==2 & (HEAR_4_X==1 | ///
        HEAR_4_X==2)) | (HEAR_3_X==1 & HEAR_4_X==3)
replace H_INDICATOR=3 if (HEAR_3_X==3 & (HEAR_4_X==1 | ///
        HEAR_4_X==2) | (HEAR_3_X==2 & HEAR_4_X==3) | ///
        (HEAR_3_X==1 & HEAR_4_X==4))
replace H_INDICATOR=4 if ((HEAR_3_X==3 & HEAR_4_X==3) | ///
        HEAR_3_X==4 | (HEAR_4_X==4 & (HEAR_3_X==2 | HEAR_3_X==3)))
tabulate H_INDICATOR
```

H_INDICATOR					
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	14976	86.4	89.4	89.4
	2.00	1156	6.7	6.9	96.3
	3.00	404	2.3	2.4	98.7
	4.00	211	1.2	1.3	100.0
	Total	16747	96.7	100.0	
Missing		579	3.3		
Total		17326	100.0		

COGNITION: Degree of difficulty remembering or concentrating

Step 7. Generate frequency distributions and cross-tabulations for Cognition domain questions and determine a Cognition Indicator.

```
gen Cognition=COG_SS if inlist(COG_SS, 1,2,3,4)
replace Cognition=. if inlist(COG_SS, 7,8,9)
tabulate Cognition
```

Cognition: Degree of difficulty remembering or concentrating

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13719	79.2	81.9	81.9
	Some difficulty	2632	15.2	15.7	97.6
	A lot of difficulty	382	2.2	2.3	99.9
	Cannot do at all	20	.1	.1	100.0
	Total	16753	96.7	100.0	
Missing		573	3.3		
Total		17326	100.0		

If response to COG-SS is *some difficulty, a lot of difficulty* or *cannot do at all*, then the respondent is asked COG_1: whether they have difficulty remembering, concentrating or both.

```
gen COG_1_R=COG_1 if inlist(COG_1, 1,2,3)
replace COG_1_R=. if inlist(COG_1, 7,8,9)
tabulate COG_1_R
```

COG_1_R: Difficulty remembering, concentrating, or both?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Difficulty remembering only	983	5.7	32.4	32.4
	Difficulty concentrating only	388	2.2	12.8	45.2
	Difficulty with both remembering and concentrating	1659	9.6	54.8	100.0
	Total	3030	17.5	100.0	
Missing		14296	82.5		
Total		17326	100.0		

Step 8. Account for those who did not answer COG_1 (COG_SS is 1 – no difficulty and they were skipped) by recoding COG_1 to 0 (No difficulty).

If response to COG-SS is 1: *no difficulty*, then the variable COG_1_R is recoded into COG_1A, and the value assigned is 0: *no difficulty*.

```
gen COG_1A=COG_1_R
replace COG_1A=0 if COG_SS==1
tabulate COG_1A
```

COG_1A: Difficulty remembering, concentrating, or both? (including no difficulty)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13719	79.2	81.9	81.9
	Difficulty remembering only	983	5.7	5.9	87.8
	Difficulty concentrating only	388	2.2	2.3	90.1
	Difficulty with both remembering and concentrating	1659	9.6	9.9	100.0
	Total	16749	96.7	100.0	
Missing		577	3.3		
Total		17326	100.0		

Step 9. Generate frequency distribution for remaining cognition questions.

Frequency distribution of the Cognition extended REMEMBERING questions: COG_2 *How often have difficulty remembering*, and COG_3 *Amount of things you have difficulty remembering*.

```
gen COG_2_R=COG_2 if inlist(COG_2, 1,2,3)
replace COG_2_R=. if inlist(COG_2, 7,8,9)

gen COG_3_R=COG_3 if inlist(COG_3, 1,2,3)
replace COG_3_R=. if inlist(COG_3, 7,8,9)
tabulate COG_2_R
```

COG_2_R: How often have difficulty remembering?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sometimes	1916	11.1	72.4	72.4
	Often	513	3.0	19.4	91.8
	All of the time	216	1.2	8.2	100.0
	Total	2645	15.3	100.0	
Missing		14681	84.7		
Total		17326	100.0		

```
tabulate COG_3_R
```

COG_3_R: Amount of things you have difficulty remembering?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	A few things	2119	12.2	80.3	80.3
	A lot of things	386	2.2	14.6	94.9
	Almost everything	134	.8	5.1	100.0
	Total	2639	15.2	100.0	
Missing		14687	84.8		
Total		17326	100.0		

Step 10. Generate cross-tabulation of the two Cognition extended set questions COG_2R by COG_3_R.

The syntax below produces a cross-tabulation of the two Extended Set REMEMBERING questions: COG_2_R: *How often you have difficulty remembering* and COG_3_R: *The amount of things you have difficulty remembering* to determine a single REMEMBERING INDICATOR.

```
tabulate COG_2_R COG_3_R
```

		COG_3_R: Amount of things you have difficulty remembering?			
COG_2_R: How often do you have difficulty remembering?		A few things	A lot of things	Almost everything	Total
How often have difficulty remembering?	Sometimes	**1788**	**105**	**20**	1913
	Often	**279**	**197**	**34**	510
	All of the time	**51**	**84**	**80**	215
Total		2118	386	134	2638

Step 11. Create a Remembering Indicator based on distribution of COG_2_R and COG_3_R.

The syntax below creates a REMEMBERING INDICATOR (R_INDICATOR) based on the two additional remembering questions (COG_2_*R* and COG_3_*R*).
If Cognition is 1: no difficulty, then the Remembering Indicator is coded as 1: the lowest level of difficulty.

```
gen R_INDICATOR=0
replace R_INDICATOR=1 if Cognition==1
replace R_INDICATOR=2 if ((COG_2_R==1 & COG_3_R==1) | ///
        (COG_3_R==1 & COG_2_R==2) | (COG_3_R==2 & COG_2_R==1))
replace R_INDICATOR=3 if (COG_3_R==2 & COG_2_R==2)
replace R_INDICATOR=4 if (COG_3_R==3 | COG_2_R==3)
```

Step 12. If COG_1A is coded as 2 (concentrating only), then the Remembering Indicator is coded as 5.

These 388 individuals are respondents who were not included in the Remembering Indicator since they had only difficulty concentrating.

```
replace R_INDICATOR=5 if (COG_1A==2)
```

Step 13. Generate frequency distribution of the Remembering Indicator.

```
tabulate R_INDICATOR
```

R_INDICATOR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	.00	580	3.3	3.3	3.3
	1.00	13719	79.2	79.2	82.5
	2.00	2172	12.5	12.5	95.1
	3.00	197	1.1	1.1	96.2
	4.00	270	1.6	1.6	97.8
	5.00	388	2.2	2.2	100.0
	Total	17326	100.0	100.0	

Step 14. Supplement Remembering Indicator with information on difficulty concentrating.

The syntax below adds information on whether cognitive difficulties are compounded by difficulty concentrating in addition to difficulty remembering.
Create a COGNITION INDICATOR (COG_INDICATOR) based on R_INDICATOR (above) and the cognition question (COG_1_R).

The 388 individuals with 'concentrating only' were allocated as follows:

1. 357 with a little difficulty on Cognition question were classified as **2**
2. 30 with a lot of difficulty on Cognition question were classified as **3**
3. 1 with cannot do on Cognition question was classified as **4**

Those with both remembering and concentrating difficulty were upgraded 36 individuals from **2** to **3**, and 125 individuals from **3** to **4.**

```
gen COG_INDICATOR=R_INDICATOR
replace COG_INDICATOR=2 if (R_INDICATOR==5 & Cognition==2)
replace COG_INDICATOR=3 if (R_INDICATOR==5 & Cognition==3)
replace COG_INDICATOR=4 if (R_INDICATOR==5 & Cognition==4)

replace COG_INDICATOR=3 if (R_INDICATOR==2 & COG_1_R==3 & Cognition==3)
replace COG_INDICATOR=4 if (R_INDICATOR==3 & COG_1_R==3 & Cognition==3)
```

Step 15. Generate frequency distribution of the Cognition Indicator.

```
tabulate COG_INDICATOR
```

COG_INDICATOR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	.00	580	3.3	3.3	3.3
	1.00	13719	79.2	79.2	82.5
	2.00	2449	14.1	14.1	96.7
	3.00	226	1.3	1.3	98.0
	4.00	352	2.0	2.0	100.0
	Total	17326	100.0	100.0	

UPPER BODY

Step 16. Generate frequency distributions and cross-tabulations for Self-care and Upper body domain questions and determine the Upper Body Indicator.

SC_SS is the WG-SS Self-care question.
UB_1 is *Difficulty raising 2 liter bottle of water from waist to eye level.*
UB_2 is *Difficulty using hands and fingers*

```
gen SELF_CARE=SC_SS if inlist(SC_SS, 1,2,3,4)
replace SELF_CARE=. if inlist(SC_SS, 7,8,9)

gen UB_1_R=UB_1 if inlist(UB_1, 1,2,3,4)
replace UB_1_R=. if inlist(UB_1, 7,8,9)

gen UB_2_R=UB_2 if inlist(UB_2, 1,2,3,4)
replace UB_2_R=. if inlist(UB_2, 7,8,9)
```

First, calculate frequency distributions on the short set and two extended set questions.

```
tabulate SELF_CARE
```

SELF_CARE: Degree of difficulty with self-care

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	16029	92.5	95.7	95.7
	Some difficulty	544	3.1	3.2	98.9
	A lot of difficulty	114	.7	.7	99.6
	Cannot do at all	68	.4	.4	100.0
	Total	16755	96.7	100.0	
Missing		571	3.3		
Total		17326	100.0		

tabulate UB_1_R

UB_1_R: Diff raising 2 liter bottle of water from waist to eye level

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15677	90.5	93.6	93.6
	Some difficulty	743	4.3	4.4	98.0
	A lot of difficulty	167	1.0	1.0	99.0
	Cannot do at all	166	1.0	1.0	100.0
	Total	16753	96.7	100.0	
Missing		573	3.3		
Total		17326	100.0		

tabulate UB_2_R

UB_2_R: Degree of difficulty using hands and fingers

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15199	87.7	90.7	90.7
	Some difficulty	1229	7.1	7.3	98.1
	A lot of difficulty	255	1.5	1.5	99.6
	Cannot do at all	70	.4	.4	100.0
	Total	16753	96.7	100.0	
Missing		573	3.3		
Total		17326	100.0		

Step 17. Generate a cross-tabulation of the two Upper body Extended Set questions: UB_2_R and UB_1_R.

The syntax below produces a cross-tabulation of the two Extended Set questions: UB_1_R: *Difficulty raising a 2 liter bottle of water from waste to eye level* UB_2_R: *Difficulty using hands and fingers* to determine a single UPPER BODY INDICATOR (UB_INDICATOR).

tabulate UB_2_R UB_1_R

		UB_1_R: Diff raising 2 liter bottle of water from waist to eye level				
UB_2_R: Difficulty using hands and fingers		No difficulty	Some difficulty	A lot of difficulty	Cannot do at all	Total
Degree of difficulty using hands and fingers	No difficulty	**14786**	**309**	**58**	**44**	15197
	Some difficulty	**782**	**355**	**51**	**40**	1228
	A lot of difficulty	**98**	**73**	**51**	**33**	255
	Cannot do at all	**9**	**5**	**7**	**49**	70
Total		15675	742	167	166	16750

Step 18. Create an UPPER BODY INDICATOR (UB_INDICATOR) based on the two additional upper body questions UB_2_R and UB_3_R.

Syntax below creates UB_INDICATOR based on the distribution in the cross-tabulation above.

```
gen UB_INDICATOR=4 if (UB_1_R==4 | UB_2_R==4)
replace UB_INDICATOR=3 if UB_INDICATOR ~=4 & (UB_1_R==3 | UB_2_R==3)
replace UB_INDICATOR=2 if UB_INDICATOR ~=4 & UB_INDICATOR~=3 & ///
        (UB_1_R==2 | UB_2_R==2)
replace UB_INDICATOR=1 if UB_INDICATOR~=4 & UB_INDICATOR~=3 & ///
        UB_INDICATOR~=2 & (UB_1_R==1 | UB_2_R==1)
tabulate UB_INDICATOR
```

UB_INDICATOR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	14790	85.4	88.3	88.3
	2.00	1448	8.4	8.6	96.9
	3.00	331	1.9	2.0	98.9
	4.00	187	1.1	1.1	100.0
	Total	16756	96.7	100.0	
Missing		570	3.3		
Total		17326	100.0		

MOBILITY

Step 19. Generate frequency distributions and cross-tabulations for Mobility domain questions and determine Mobility Indicator.

MOB_SS is the WG-SS Mobility question.
MOB_4 is Difficulty walking 100 yards on level ground without aid or equipment.
MOB_5 is Difficulty walking 1/3rd mile on level ground without aid or equipment.

```
gen Mobility=MOB_SS if inlist(MOB_SS, 1,2,3,4)
replace Mobility=. if inlist(MOB_SS, 7,8,9)

gen MOB_4_R=MOB_4 if inlist(MOB_4, 1,2,3,4)
replace MOB_4_R=. if inlist(MOB_4, 7,8,9)

gen MOB_5_R=MOB_5 if inlist(MOB_5, 1,2,3,4)
replace MOB_5_R=. if inlist(MOB_5, 7,8,9)
```

First, calculate frequency distributions on the short set and two extended set WALKING questions.

tabulate Mobility

Mobility: Degree of difficulty walking or climbing steps

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13424	77.5	80.1	80.1
	Some difficulty	2165	12.5	12.9	93.0
	A lot of difficulty	792	4.6	4.7	97.7
	Cannot do at all	380	2.2	2.3	100.0
	Total	16761	96.7	100.0	
Missing		565	3.3		
Total		17326	100.0		

tabulate MOB_4_R

MOB_4_R: Diff walking 100 yards on level ground w/o aid or equipment

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13892	80.2	84.8	84.8
	Some difficulty	1369	7.9	8.4	93.2
	A lot of difficulty	491	2.8	3.0	96.2
	Cannot do at all	**623**	3.6	3.8	100.0
	Total	16375	94.5	100.0	
Missing		951	5.5		
Total		17326	100.0		

tabulate MOB_5_R

MOB_5_R: Diff walking 1/3rd mile on level ground w/o aid or equipment

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13025	75.2	82.8	82.8
	Some difficulty	1650	9.5	10.5	93.3
	A lot of difficulty	708	4.1	4.5	97.8
	Cannot do at all	347	2.0	2.2	100.0
	Total	15730	90.8	100.0	
Missing		1596	9.2		
Total		17326	100.0		

Step 20. Generate a cross-tabulation of the walking distance questions: MOB_4_R and MOB_5_R.

The syntax below produces a cross-tabulation of the two Extended Set WALKING questions: MOB_4_R: *Difficulty walking 100 yards without equipment* and MOB_5_R: *Difficulty walking 1/3 mile without equipment* to determine a single WALKING INDICATOR.

NOTE: **623** individuals who responded cannot do at all to MOB_4_R were not asked MOB_5_R and they do not appear in the table below. They are, however, accounted for in the WALKING indicator calculation.

```
tabulate MOB_4_R MOB_5_R
```

MOB_4_R: Diff walking 100 yards on level ground w/o aid or equipment	**MOB_5_R: Diff walking 1/3rd mile on level ground w/o aid or equipment**				
	No difficulty	Some difficulty	A lot of difficulty	Cannot do at all	Total
No difficulty	**12950**	**819**	**63**	**39**	13871
Some difficulty	**72**	**810**	**343**	**142**	1367
A lot of difficulty	**3**	**21**	**301**	**166**	491
Cannot do at all **(623)**	**0**	**0**	**0**	**0**	0
Total	13025	1650	707	347	15729

Step 21. Create a WALKING INDICATOR (WALK_INDICATOR) based on the two additional walking questions MOB_4_R and MOB_5_R.

Syntax below creates WALKING_INDICATOR based on the distribution in the cross-tabulation above.

```
gen WALK_INDICATOR=0
replace WALK_INDICATOR=1 if (MOB_4_R==1 & (MOB_5_R==1 | MOB_5_R==2))
replace WALK_INDICATOR=2 if (MOB_4_R==1 & MOB_5_R==3) | (MOB_4_R==2 & ///
        (MOB_5_R==1 | MOB_5_R== 2 | MOB_5_R==3))
replace WALK_INDICATOR=3 if (MOB_4_R==1 & MOB_5_R==4) | (MOB_4_R==3 & ///
        (MOB_5_R==1 | MOB_5_R==2 | MOB_5_R==3))
replace WALK_INDICATOR=4 if (MOB_4_R==2 & MOB_5_R==4) | (MOB_4_R==3 & ///
        MOB_5_R==4)
```

Syntax below includes the **623** who responded cannot do at all to MOB_4_R into the WALKING INDICATOR.

```
replace WALK_INDICATOR=4 if (WALK_INDICATOR==0 & MOB_4_R==4)
replace WALK_INDICATOR=. if WALK_INDICATOR==0
tabulate WALK_INDICATOR
```

WALK_INDICATOR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	13769	79.5	84.2	84.2
	2.00	1288	7.4	7.9	92.1
	3.00	364	2.1	2.2	94.3
	4.00	931	5.4	5.7	100.0
	Total	16352	94.4	100.0	
Missing		974	5.6		
Total		17326	100.0		

Step 22. Supplement Walking Indicator with information on difficulty Climbing steps (MOB_6).

Syntax below adds information from MOB_6 on *difficulty climbing up or down 12 steps* to create a combined Mobility Indicator (MOB_INDICATOR).

```
gen MOB_6_R=MOB_6 if inlist(MOB_6, 1,2,3,4)
replace MOB_6_R=. if inlist(MOB_6, 7,8,9)
tabulate WALK_INDICATOR MOB_6_R
```

		MOB_6_R: Difficulty climbing up or down 12 steps				
WALK_INDICATOR		No difficulty	Some difficulty	A lot of difficulty	Cannot do at all	Total
walk_INDICATOR2	1.00	13048	645	55	17	13765
	2.00	370	767	135	16	1288
	3.00	43	106	200	14	363
	4.00	79	242	226	384	931
Total		13540	1760	616	431	16347

According to the table above, the syntax below reclassifies:

1. 55 individuals with level 1 on the WALKING INDICATOR as level 2 on the MOBILITY INDICATOR
2. 17 individuals with level 1 on the WALKING INDICATOR and 135 individuals with level 2 on the WALKING INDICATOR as level 3 on the MOBILITY INDICATOR, and
3. 16 individuals with level 2 on the WALKING INDICATOR as level 4 on the MOBILITY INDICATOR.

Step 23. Create a mobility indication (MOB_INDICATOR) with information garnered from cross-tabulation above.

```
gen MOB_INDICATOR= WALK_INDICATOR
replace MOB_INDICATOR=2 if (WALK_INDICATOR==1 & MOB_6_R==3)
```

```
replace MOB_INDICATOR=3 if (WALK_INDICATOR==2 & MOB_6_R==3) | ///
        (WALK_INDICATOR==1 & MOB_6_R==4)
replace MOB_INDICATOR=4 if (WALK_INDICATOR==2 & MOB_6_R==4)
tabulate MOB_INDICATOR
```

MOB_IDICATOR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	13697	79.1	83.8	83.8
	2.00	1192	6.9	7.3	91.1
	3.00	516	3.0	3.2	94.2
	4.00	947	5.5	5.8	100.0
	Total	16352	94.4	100.0	
Missing		974	5.6		
Total		17326	100.0		

ANXIETY

Step 24.Generate frequency distribution on ANX_1.

First, calculate frequency distributions on ANX_1: *How often do you feel worried, nervous or anxious*?

```
gen ANX_1_R=ANX_1 if inlist(ANX_1, 1,2,3,4,5)
replace ANX_1_R=. if inlist(ANX_1, 7,8,9)
tabulate ANX_1_R
```

ANX_1_R: How often feel worried, nervous, or anxious?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Daily	1632	9.4	9.8	9.8
	Weekly	1872	10.8	11.2	21.0
	Monthly	1558	9.0	9.3	30.4
	A few times a year	4898	28.3	29.4	59.7
	Never	6714	38.8	40.3	100.0
	Total	16674	96.2	100.0	
Missing		652	3.8		
Total		17326	100.0		

Step 25. The syntax below recodes ANX_3R into ANX_3Y
1) to create a NOT ASKED category based on those who responded NEVER to ANX_1_R and
2) to place "SOMEWHERE BETWEEN" numerically in-between "A LITTLE" and "A LOT".

```
gen ANX_3Y=1 if ANX_3R==1
```

```
replace ANX_3Y=3 if ANX_3R==2
replace ANX_3Y=2 if ANX_3R==3
replace ANX_3Y=. if inlist(ANX_3R, 7, 8, 9)
```

Recode ANX_3Y to 0 (not asked) if ANX_1 is 5 (Never)

```
replace ANX_3Y=0 if ANX_1==5
tabulate ANX_3Y
```

ANX_3Y: Level of feelings last time felt worried/nervous/anxious

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	6714	38.8	40.3	40.3
	A little	5700	32.9	34.2	74.5
	In between a little and a lot	3076	17.8	18.5	92.9
	A lot	1176	6.8	7.1	100.0
	Total	16666	96.2	100.0	
Missing		660	3.8		
Total		17326	100.0		

Step 26. Generate a cross-tabulation of the anxiety Extended Set questions: ANX_1_R and ANX_3Y.

The syntax below produces a cross-tabulation of ANX_1_R: *How often you felt worried, nervous or anxious* (a measure of frequency) and ANX_3Y: *The level of those feeling the last time you felt worried, nervous or anxious* (a measure of intensity) – used to determine a single ANXIETY INDICATOR (ANX_INDICATOR).

```
tabulate ANX_3Y ANX_1_R
```

ANX_3Y: Level of feelings last time felt worried, nervous or anxious	**ANX_1_R: How often feel worried, nervous or anxious?**					
	Daily	Weekly	Monthly	A Few Times A Year	Never	Total
Not asked	**0**	**0**	**0**	**0**	**6714**	6714
A little	**489**	**887**	**897**	**3417**	**0**	5690
In between a little and a lot	**589**	**725**	**535**	**1221**	**0**	3070
A lot	**548**	**256**	**123**	**248**	**0**	1175
Total	1626	1868	1555	4886	6714	16649

Step 27. Create an ANXIETY INDICATOR (ANX_INDICATOR) based on the two anxiety questions ANX_1_R and ANX_3Y.

Syntax below creates ANX_INDICATOR based on the distribution in the cross-tabulation above.

```
gen ANX_INDICATOR=1 if (ANX_3Y <= 4 & (ANX_1_R==4 | ANX_1_R==5))
replace ANX_INDICATOR=2 if ((ANX_1_R==3) | (ANX_1_R < 3 & ANX_3Y==1) | ///
        (ANX_1_R==2 & ANX_3Y==2))
replace ANX_INDICATOR=3 if ((ANX_1_R==1 & ANX_3Y==2) | ///
        (ANX_1_R==2 & ANX_3Y==3))
replace ANX_INDICATOR=4 if (ANX_1_R==1 & ANX_3Y==3)
replace ANX_INDICATOR=. if (missing(ANX_1_R) | missing(ANX_3Y))
tabulate ANX_INDICATOR
```

ANX_INDICATOR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	11600	67.0	69.7	69.7
	2.00	3656	21.1	22.0	91.6
	3.00	845	4.9	5.1	96.7
	4.00	548	3.2	3.3	100.0
	Total	16649	96.1	100.0	
Missing		677	3.9		
Total		17326	100.0		

DEPRESSION

Step 28.Generate frequency distribution on DEP_1.
First, calculate frequency distributions on DEP_1: *How often do you feel depressed*?

```
gen DEP_1_R=DEP_1 if inlist(DEP_1, 1,2,3,4,5)
replace DEP_1_R=. if inlist(DEP_1, 7,8,9)
tabulate DEP_1_R
```

DEP_1_R: How often do you feel depressed?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Daily	756	4.4	4.5	4.5
	Weekly	926	5.3	5.6	10.1
	Monthly	1038	6.0	6.2	16.3
	A few times a year	4012	23.2	24.1	40.4
	Never	9929	57.3	59.6	100.0
	Total	16661	96.2	100.0	
Missing		665	3.8		
Total		17326	100.0		

Step 29. The syntax below recodes DEP_3R into DEP_3Y to place "SOMEWHERE BETWEEN" numerically in-between "A LITTLE" and "A LOT". It also creates the category NOT ASKED, if DEP_1 is NEVER (1)

```
gen DEP_3Y=1 if DEP_3R==1
replace DEP_3Y=3 if DEP_3R==2
replace DEP_3Y=2 if DEP_3R==3
replace DEP_3Y=. if inlist(DEP_3R, 7,8,9)

replace DEP_3Y=0 if DEP_1==5
tabulate DEP_3Y
```

DEP_3Y: Level of feelings last time felt depressed

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	9929	57.3	59.6	59.6
	A little	3775	21.8	22.7	82.3
	In between a little and a lot	2016	11.6	12.1	94.4
	A lot	935	5.4	5.6	100.0
	Total	16655	96.1	100.0	
Missing		671	3.9		
Total		17326	100.0		

Step 30. Generate a cross-tabulation of the depression Extended Set questions: DEP_1_R and DEP_3Y.

The syntax below produces a cross-tabulation of DEP_1_R: *How often do you feel depressed* (a measure of frequency) and DEP_3Y: *The level of those feeling the last time you felt depressed* (a measure of intensity) – used to determine a single DEPRESSION INDICATOR (DEP_INDICATOR).

```
tabulate DEP_3Y DEP_1_R
```

	DEP_1_R: How often do you feel depressed?					
DEP_3Y: Level of feelings last time felt depressed	Daily	Weekly	Monthly	A Few Times A Year	Never	Total
Not asked	**0**	**0**	**0**	**0**	**9929**	9929
A little	**161**	**346**	**548**	**2708**	**0**	3763
In between a little and a lot	**209**	**384**	**378**	**1042**	**0**	2013
A lot	**381**	**191**	**112**	**248**	**0**	932
Total	751	921	1038	3998	9929	16637

Step 31. Create a DEPRESSION INDICATOR (DEP_INDICATOR) based on the two depression questions DEP_1_R and DEP_3Y.

Syntax below creates DEP_INDICATOR based on the distribution in the cross-tabulation above.

```
gen DEP_INDICATOR=. if (missing(DEP_1_R) | missing(DEP_3Y))
replace DEP_INDICATOR=1 if (DEP_3Y <= 4 & (DEP_1_R==4 | DEP_1_R==5))
replace DEP_INDICATOR=2 if ((DEP_1_R==3) | (DEP_1_R < 3 & DEP_3Y==1) | ///
        (DEP_1==2 & DEP_3Y==2))
replace DEP_INDICATOR=3 if ((DEP_1_R==1 & DEP_3Y==2) | ///
        (DEP_1_R==2 & DEP_3Y==3))
replace DEP_INDICATOR=4 if (DEP_1_R==1 & DEP_3Y==3)
tabulate DEP_INDICATOR
```

DEP_INDICATOR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	13927	80.4	83.7	83.7
	2.00	1929	11.1	11.6	95.3
	3.00	400	2.3	2.4	97.7
	4.00	381	2.2	2.3	100.0
	Total	16637	96.0	100.0	
Missing		689	4.0		
Total		17326	100.0		

PAIN

Step 32.Generate frequency distribution on PAIN_2.
First, calculate frequency distributions on PAIN_2: *Frequency of pain in the past 3 months*.

```
gen PAIN_2_R=PAIN_2 if inlist(PAIN_2, 1,2,3,4)
replace PAIN_2_R=. if inlist(PAIN_2, 7,8,9)
tabulate PAIN_2_R
```

PAIN_2_R: Frequency of pain in past 3 months

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Never	6636	38.3	39.8	39.8
	Some days	6556	37.8	39.3	79.2
	Most days	1227	7.1	7.4	86.5
	Every day	2245	13.0	13.5	100.0
	Total	16664	96.2	100.0	
Missing		662	3.8		
Total		17326	100.0		

Step 33. The syntax below recodes PAIN_4 into PAIN_4Y to place "SOMEWHERE BETWEEN" numerically in-between "A LITTLE" and "A LOT". It also creates the category NOT ASKED, if PAIN_2 is NEVER (1).

```
gen PAIN_4Y=1 if PAIN_4==1
replace PAIN_4Y=3 if PAIN_4==2
replace PAIN_4Y=2 if PAIN_4==3
replace PAIN_4Y=. if inlist(PAIN_4, 7,8,9)

replace PAIN_4Y=0 if PAIN_2==1
tabulate PAIN_4Y
```

PAIN_4Y: How much pain you had last time you had pain?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	6636	38.3	39.8	39.8
	A little	4865	28.1	29.2	69.0
	In between a little and a lot	3296	19.0	19.8	88.8
	A lot	1869	10.8	11.2	100.0
	Total	16666	96.2	100.0	
Missing		660	3.8		
Total		17326	100.0		

Step 34. Generate a cross-tabulation of the PAIN Extended Set questions: PAIN_2_R and PAIN_4Y.

The syntax below produces a cross-tabulation of PAIN_2_R: *Frequency of pain in the past 3 months* and PAIN_4Y: *How much pain you has the last time you had pain* (a measure of intensity) – used to determine a single PAIN INDICATOR (P_INDICATOR).

```
tabulate PAIN_4Y PAIN_2_R
```

PAIN_2_R: Frequency of pain in past 3 months

PAIN_4Y: How much pain you had last time you had pain		Never	Some Days	Most days	Every day	Total
	Not asked	**6636**	0	0	0	6636
	A little	0	**4136**	**323**	**401**	4860
	In between a little and a lot	0	**1772**	**624**	**896**	3296
	A lot	0	**645**	**278**	**944**	1867
Total		6636	6553	1225	2241	16655

Step 35. Create a PAIN INDICATOR (P_INDICATOR) based on the two PAIN questions PAIN_2_R and PAIN_4Y.

Syntax below creates P_INDICATOR based on the distribution in the cross-tabulation above.

```
gen P_INDICATOR=1 if (PAIN_2_R==1) | (PAIN_4Y==1 & ///
        (PAIN_2_R==2 | PAIN_2_R==3))
replace P_INDICATOR=2 if ((PAIN_2_R==2 & (PAIN_4Y==2 | PAIN_4Y==3)) | ///
        (PAIN_2_R==3 & PAIN_4Y==2) | (PAIN_2_R==4 & PAIN_4Y==1))
replace P_INDICATOR=3 if (PAIN_2_R==3 & PAIN_4Y==3) | ///
        (PAIN_2_R==4 & PAIN_4Y==2)
replace P_INDICATOR=4 if (PAIN_2_R==4 & PAIN_4Y==3)
tabulate P_INDICATOR
```

P_INDICATOR

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	11095	64.0	66.6	66.6
	2.00	3442	19.9	20.7	87.3
	3.00	1174	6.8	7.0	94.3
	4.00	944	5.4	5.7	100.0
	Total	16655	96.1	100.0	
Missing		671	3.9		
Total		17326	100.0		

FATIGUE (Tired)

Step 36.Generate frequency distribution on FATIGUE Extended Set questions Tired_1, Tired_2 and Tired_3.

First, calculate frequency distributions on TIRED_1: *How often you felt tired in the past 3 months*.

```
gen TIRED_1_R=TIRED_1 if inlist(TIRED_1, 1,2,3,4)
replace TIRED_1_R=. if inlist(TIRED_1, 7,8,9)
tabulate TIRED_1_R
```

TIRED_1_R: How often felt very tired or exhausted in past 3 months

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Never	5619	32.4	33.7	33.7
	Some days	8391	48.4	50.4	84.1
	Most days	1632	9.4	9.8	93.9
	Every day	1019	5.9	6.1	100.0
	Total	16661	96.2	100.0	
Missing		665	3.8		
Total		17326	100.0		

Step 37. Recode Tired_2 to 0 (not asked) if Tired_1 is 1 (Never).

If response to TIRED_1 is 1: Never, then TIRED_2 (*How long most recent tired or exhausted feelings lasted*) is not asked. This variable is recoded so these individuals are included in the syntax below.

```
gen TIRED_2_R=.
replace TIRED_2_R=TIRED_2 if inlist(TIRED_2, 1,2,3)
replace TIRED_2_R=. if inlist(TIRED_2, 7,8,9)
replace TTIRED_2_R=0 if TIRED_1==1
tabulate TIRED_2_R
```

TIRED_2_R: How long most recent tired or exhausted feelings lasted?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	5619	32.4	33.8	33.8
	Some of the day	8036	46.4	48.3	82.0
	Most of the day	1955	11.3	11.7	93.8
	All of the day	1036	6.0	6.2	100.0
	Total	16646	96.1	100.0	
Missing		680	3.9		
Total		17326	100.0		

Step 38. The syntax below recodes TIRED_3 into TIRED_3Y to place "SOMEWHERE BETWEEN" numerically in-between "A LITTLE" and "A LOT".

Also, if response to TIRED_1 is 1: Never, then TIRED_3 (*Level of tiredness*) is not asked. This variable is recoded so these individuals are included in the syntax below.

```
gen TIRED_3Y=1 if TIRED_3==1
replace TIRED_3Y=3 if TIRED_3==2
replace TIRED_3Y=2 if TIRED_3==3
replace TIRED_3Y=. if inlist(TIRED_3, 7, 8, 9)
replace TIRED_3Y=0 if TIRED_1==1
tabulate TIRED_3Y
```

TIRED_3Y: Level of tiredness

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	5619	32.4	33.8	33.8
	A little	4912	28.4	29.5	63.3
	In between a little and a lot	4030	23.3	24.2	87.5
	A lot	2087	12.0	12.5	100.0
	Total	16648	96.1	100.0	
Missing		678	3.9		
Total		17326	100.0		

Step 39. Generate a cross-tabulation of the FATIGUE Extended Set questions: TIRED_1_R, TIRED_2_R and TIRED_3Y.

The syntax below produces a cross-tabulation of TIRED_1_R: *How often you felt tired or exhausted in the past 3 months* (a measure of frequency) and TIRED_2_R: *How long those feelings lasted* (a measure of duration) and TIRED_3Y: *The level of tiredness* (a measure of intensity) – used to determine a single TIRED INDICATOR (T_INDICATOR).

```
table TIRED_2_R TIRED_1_R , by(TIRED_3Y ) contents(freq)
```

TIRED_3Y: Level of tiredness: *Intensity*	**TIRED_2: How long feelings lasted: *Duration***	**TIRED_1: How often felt very tired or exhausted in past 3 months: *Frequency***				
		Never	Some days	Most days	Every day	Total
Not asked	Not asked	**5619**				5619
A little	Some of the day		**4066**	**264**	**124**	4454
	Most of the day		**252**	**73**	**27**	352
	All of the day		**68**	**15**	**18**	101
In between	Some of the day		**2224**	**400**	**166**	2791
	Most of the day		**497**	**266**	**123**	887
	All of the day		**194**	**71**	**84**	349
A lot	Some of the day		**536**	**165**	**84**	785
	Most of the day		**297**	**255**	**160**	713
	All of the day		**237**	**118**	**230**	585
TOTAL		5619	8371	1627	1016	16633

Step 40. Create a FATIGUE INDICATOR (T_INDICATOR) based on the three FATIGUE questions TIRED_1_R, TIRED_2_R and TIRED_3Y.

Syntax below creates T_INDICATOR based on the distribution in the cross-tabulation above.

```
gen T_INDICATOR = .
replace T_INDICATOR=1 if (TIRED_1_R==1)
replace T_INDICATOR=1 if (TIRED_1_R==2 & TIRED_2_R==1 & TIRED_3Y==1)
replace T_INDICATOR=1 if (TIRED_1_R==3 & TTIRED_2_R==1 & TIRED_3Y==1)

replace T_INDICATOR=2 if (inlist(TIRED_1_R, 2,3,4) & inlist(TIRED_2_R, 2,3) & ///
        TIRED_3Y==1)
replace T_INDICATOR=2 if (inlist(TIRED_1_R ,2,3,4) & TIRED_2_R==1 & TIRED_3Y==2)
replace T_INDICATOR=2 if (TIRED_1_R==2 & TIRED_2_R==2 & TIRED_3Y==2)
replace T_INDICATOR=2 if (TIRED_1_R==4 & TIRED_2_R==1 & TIRED_3Y==1)
```

```
replace T_INDICATOR=3 if (inlist(TIRED_1_R, 3,4) & TIRED_2_R==2 & TIRED_3Y==2)
replace T_INDICATOR=3 if (inlist(TIRED_1_R, 2,3,4) & TIRED_2_R==3 & TIRED_3Y==2)
replace T_INDICATOR=3 if (inlist(TIRED_1_R, 2,3,4) & TIRED_2_R==1 & TIRED_3Y==3)
replace T_INDICATOR=3 if (inlist(TIRED_1_R, 2,3) & TIRED_2_R==2 & TIRED_3Y==3)
replace T_INDICATOR=3 if (TIRED_1_R==2 & TIRED_2_R==3 & TIRED_3Y==3)

replace T_INDICATOR=4 if (TIRED_1_R==4 & TIRED_2_R==2 & TIRED_3Y==3)
replace T_INDICATOR=4 if (inlist(TIRED_1_R, 3,4) & TIRED_2_R==3 & TIRED_3Y==3)
tabulate T_INDICATOR
```

T_Indicator

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	9949	57.4	59.8	59.8
	2.00	3864	22.3	23.2	83.0
	3.00	2312	13.3	13.9	96.9
	4.00	508	2.9	3.1	100.0
	Total	16633	96.0	100.0	
Missing		693	4.0		
Total		17326	100.0		

Creating Disability Status Indicators

Type of Disability Indicator		**Number of Questions**
SS_1	**Short Set (SS)**	**6**
Extended Set		
ES_1	**SS** + Hearing-indicator, Mobility-indicator, Cognition-indicator, Upper Body-indicator + PFAD (4)*	**25**
ES_2	**SS** + Hearing-indicator, Mobility-indicator, Cognition-indicator, Upper Body-indicator + AD (4)†	**20**
Short Set Enhanced		
ES_3	**SS** + Upper Body-indicator + AD (4)†	**12**

* PFAD (4): Pain, Fatigue, Anxiety and Depression Indicators at level 4
† AD (4): Anxiety and Depression Indicators at level 4

SS_1: WG Short Set Disability Indicator based on the 6 short set questions

The syntax below calculates the WG Short Set Disability Indicator based on the six short set questions **SS_1** at the recommended cut-off. The level of inclusion is: at least 1 domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL.

```
gen SS_1=2
replace SS_1=. if (missing(Vision) & missing(Hearing) & missing(Mobility) & ///
        missing(Communication) & missing(SELF_CARE) & missing(Cognition))
replace SS_1=1 if ((Vision==3 | Vision==4) | (Hearing==3 | Hearing==4) | ///
        (Mobility==3 | Mobility== 4) | (Communication==3 | Communication==4) | ///
        (SELF_CARE==3 | SELF_CARE==4) | (Cognition==3 | Cognition==4))
tabulate SS_1
```

SS_1: WG Short Set Disability Identifier

		Frequency	Percent	Valid Percent	Weighted Estimate
Valid	WITHOUT DISABILITY	14905	86.0	88.8	90.5
	WITH DISABILITY	1872	10.8	11.2	9.5
	Total	16777	96.8	100.0	
Missing		549	3.2		
Total		17326	100.0		

ES_1: SS_1 + Hearing-indicator, Mobility-indicator, Cognition-indicator + Upper Body-indicator + PFAD (4)

The syntax below calculates an Extended Set Disability Indicator (**ES_1**) based on 25 questions at the recommended cut-off. The level of inclusion is: at least 1 domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL for the six short set question; severity levels 3 or 4 for the Hearing-, Mobility-, Cognition- and Upper body-Indicators; and severity level 4 for Pain-, Fatigue-, Anxiety- and Depression-Indicators.

```
gen ES_1=2
replace ES_1=. if ///
        (missing(SS_1) & ///
        (H_INDICATOR < 1 | H_INDICATOR > 4) & ///
        (MOB_INDICATOR < 1 | MOB_INDICATOR > 4) & ///
         missing(COM_SS) & ///
         missing(SC_SS) & ///
        (COG_INDICATOR < 1 | COG_INDICATOR > 4) & ///
        (UB_INDICATOR < 1 | UB_INDICATOR > 4) & ///
         missing(P_INDICATOR) & ///
        (T_INDICATOR < 1 | T_INDICATOR > 4) & ///
        (ANX_INDICATOR < 1 | ANX_INDICATOR > 4) & ///
        (DEP_INDICATOR < 1 |DEP_INDICATOR > 4) )
replace ES_1=1 if (SS_1==1 | ///
```

```
        (H_INDICATOR==3 | H_INDICATOR==4) | ///
        (MOB_INDICATOR==3 | MOB_INDICATOR==4) | ///
        (COG_INDICATOR==3 | COG_INDICATOR==4) | ///
        (UB_INDICATOR==3 | UB_INDICATOR==4) | ///
         P_INDICATOR==4 | T_INDICATOR==4 | ///
        ANX_INDICATOR==4 | DEP_INDICATOR==4)
tabulate ES_1
```

ES_1: WG-ES Disability Indicator based on 11 domains and 25 questions

		Frequency	Percent	Valid Percent	Weighted Estimate
Valid	WITHOUT DISABILITY	13823	79.8	79.8	82.3
	WITH DISABILITY	3503	20.2	20.2	17.7
	Total	17326	100.0	100.0	

ES_2: SS_1 + Hearing-indicator, Mobility-indicator, Cognition-indicator + Upper Body-indicator + AD (4)

The syntax below calculates an Extended Set Disability Indicator (**ES_2**) based on 20 questions at the recommended cut-off. The level of inclusion is: at least 1 domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL for the six short set question; severity levels 3 or 4 for the Hearing-, Mobility-, Cognition- and Upper body-Indicators; and severity level 4 for Anxiety- and Depression-Indicators.

```
gen ES_2=2
replace ES_2=. if (missing(SS_1) & ///
        (H_INDICATOR < 1 | H_INDICATOR > 4) & ///
        (MOB_INDICATOR < 1 | MOB_INDICATOR > 4) & ///
         missing(COM_SS) & missing(SC_SS) & ///
        (COG_INDICATOR < 1 | COG_INDICATOR > 4) & ///
        (UB_INDICATOR < 1 | UB_INDICATOR > 4) & ///
        (ANX_INDICATOR < 1 | ANX_INDICATOR > 4) & ///
        (DEP_INDICATOR < 1 | DEP_INDICATOR > 4))
replace ES_2=1 if (SS_1==1 | (H_INDICATOR==3 | H_INDICATOR==4) | ///
        (MOB_INDICATOR==3 | MOB_INDICATOR==4) | ///
        (COG_INDICATOR==3 | COG_INDICATOR==4) | ///
        (UB_INDICATOR==3 | UB_INDICATOR==4) | ///
        ANX_INDICATOR==4 | DEP_INDICATOR==4)
tabulate ES_2
```

ES_2: WG-ES Disability Indicator based on 9 domains and 20 questions

		Frequency	Percent	Valid Percent	Weighted Estimate
Valid	WITHOUT DISABILITY	14222	82.1	82.1	84.6
	WITH DISABILITY	3104	17.9	17.9	15.4
	Total	17326	100.0	100.0	

ES_3: SS_1 + Upper Body-indicator + AD (4)

The syntax below calculates the WG Short Set ENHANCED Disability Indicator (**ES_3**) based on the 12 questions at the recommended cut-off. The level of inclusion is: at least 1 domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL for the six short set question; severity levels 3 or 4 for the Upper body-Indicators; and severity level 4 for Anxiety- and Depression-Indicators.

```
gen ES_3=2
replace es_3=. if (missing(Vision) & ///
        missing(Vearing) & ///
        missing(Mobility) & ///
        missing(Communication) & ///
        missing(SELF_CARE) & ///
        missing(Cognition) & ///
        missing(UB_INDICATOR) & ///
        missing(ANX_INDICATOR) & ///
        missing(DEP_INDICATOR) )
 replace es_3=1 if ((Vision==3 | Vision==4) | ///
        (Hearing==3 | Hearing==4) | ///
        (Mobility==3 | Mobility==4) | ///
        (Communication==3 | Communication==4) | ///
        (SELF_CARE==3 | SELF_CARE==4) | ///
        (Cognition==3 | Cognition==4) | ///
        (UB_INDICATOR==3 | UB_INDICATOR==4) | ///
         ANX_INDICATOR==4 | DEP_INDICATOR==4)
tabulate ES_3
```

ES_3: WG-SS Enhanced Disability Indicator based on 9 domains and 12 questions

		Frequency	Percent	Valid Percent	Weighted Estimate
Valid	WITHOUT DISABILITY	14393	83.1	85.8	87.7
	WITH DISABILITY	2384	13.8	14.2	12.3
	Total	16777	96.9	100.0	
Missing		549	3.2		
Total		17326	100.1		

POSTSCRIPT: Why exclude Pain and fatigue?

Of note is the exclusion of the pain and fatigue domains from several of the Disability Identifiers above. There has been considerable discussion within the WG on these domains. They are not, strictly speaking, domains of functioning – and our analyses indicated that they are both highly correlated with other domains – and that the rates of disability with the inclusion of these domains can be very high. Finally, in terms of international comparability, these domains are less universal; that is, they are more susceptible to local, socio-cultural influences than other domains of functioning.

For these reasons, we chose to exclude them from several of these analyses, though they can be included in supplemental analyses carried out by NSOs on a national basis.