

Documentation for all Data Sets

INTRODUCTION	2
HEALTH ABC DATA ANALYSIS FILE	6
PARTICIPANT HISTORY FILE (PH)	6
YEAR 15 QUARTERS 1-4 PARTICIPANT TELEPHONE CONTACT DATA	8
YEAR 15 QUARTERS 1-4 PROXY TELEPHONE CONTACT DATA	9
MISSED FOLLOW-UP CONTACT DATA.....	10
SPECIAL MISSING VALUE CODES	11
DROPPED VARIABLES.....	13
LISTINGS	13
Appendix 1 - SAS Programming Code for Calculated (derived) Variables.....	14

INTRODUCTION

Beginning with Year 15 of the Health ABC project, participants are being interviewed by telephone every quarter. If the participant was contacted, s/he was administered a Year 15+ Participant Telephone interview. If the participant could not be reached or was unable to complete the participant interview due to disability, illness, death or another reason, Health ABC clinic staff attempted to contact a relative, friend or other person who could provide information about the participant. If such a person was identified and s/he was willing to serve as a proxy for the participant, that person was given a Year 15+ Proxy Telephone interview.

Interview forms for both the Year 15+ participant and proxy interviews are available on the Health ABC website. You can access them by going to the website and clicking on the “Datasets/Documentation” link in the header, and then the “Year 15 Data Code Book” link and finally the “Annotated Data Collection Forms” link. Once there please select either the “Year 15+ Participant Telephone Interview” or “Year 15+ Proxy Telephone Interview”. The participant interview has 41 pages and the proxy 21. Each question on a page should have a variable name in BLUE next to it and these blue variables will be included in one of the study datasets. (Unreleased variables are crossed out and in gray.) Both the participant and proxy interviews have four separate datasets for Year 15; one for each quarter: 1, 2, 3 or 4.

The questions asked of a participant or proxy depends on what quarter the interview was conducted. Fortunately, if a page is used in a quarter, all the questions on that page are asked. For example, look at page 3 of the annotated forms for the participant interview. In the bottom right-hand corner of page 3 you can see that Q1-Q4 is specified in red. This means that all questions on page 3 are asked every quarter. Find 3 in the page number column of the Year 15+ Participant Interview table below. Looking across to the quarter columns you can see that each cell has a number in it. A 1 in column Year 15 Quarter 1 indicates that page 3 questions are asked in quarter 1. A 2 in column Year 15 Quarter 2 means page 3 is also used in the second quarter. A 3 in the third quarter column means the page is used then. Finally, a 4 in the fourth quarter column means it's included in that quarter. Looking down the rows of the table you can see in which quarters a page is being used. For example, page 4 is only used in quarters 1 and 3.

Each specific quarter's dataset will only contain the questions (variables) on the pages used in that quarter. For example, the Year 15 Quarter 1 dataset will only include questions asked from the pages used for quarter 1. Again, a 1 in column Year 15 Quarter 1 indicates that page is used and all questions on that page are asked. If a cell in the Year 15 Quarter 1 column is blank, those questions are NOT in the Year 15 Quarter 1 dataset.

There is a column in Year 15+ participant interview table labeled Variable Prefix. This reflects the variable naming convention for the Year 15 interviews. Each variable name begins with the quarter in which it's being asked. For example all Year 15 Quarter 1 variable names begin with 'Y15Q1_'. After the underscore comes a two character variable prefix. For example, all variables on page 4 include the prefix 'AD' after the underscore in the quarter portion of the variable name. Note that all variables on page 4 include the prefix 'AD' in their names. You can confirm this by looking at page 4 of the annotated forms. Again, all variable names include that prefix after the quarter portion of the name. Also, the prefix is identified in RED in the lower right hand corner of the annotated form right under the list of quarters when the page is used. The rest of the variable name after the prefix is the unique name of the question being asked.

Lastly, the Page Description column indicates what type of data are included on a page or group of pages. These descriptions come from the label at the top of each annotated form. If there isn't a title on a page one was created.

You can use the Year 15+ tables below to determine what type of data was collected, and in which quarters. Also, you can use these tables with the list of variables for each dataset provided in the “Proc Contents for all Datasets” also under the “Year 15 Data Code Book” section of the Health ABC study website where the annotated forms reside. If you want to know more about the variables on a particular page, look at the group of variables that have that page’s variable prefix in their names. Variable labels are usually a shortened version of the question on the form the variable corresponds with.

The Year 15+ participant telephone interview table is followed by a table for the proxy interview.

Health ABC Participant Telephone Interview

Page Description	Page Number	Variable Prefix	Year 15 Quarter 1	Year 15 Quarter 2	Year 15 Quarter 3	Year 15 Quarter 4
Date of participant interview	1	AA	1	2	3	4
Health status, stay in bed & cut down on things	2	AB	1	2	3	4
Medical status	3	AC	1	2	3	4
Physical function	4	AD	1		3	
	5	AE	1		3	
	6	AF	1	2	3	4
	7	AG	1	2	3	4
Physical activity and exercise	8	AH	1		3	
	9	AJ	1			
	10	AK	1			
	11	AL	1			
Symptoms	12	AM	1	2	3	4
Telephone interview for cognitive status (TICS)	13	AN		2		4
	14	AP		2		4
	15	AQ		2		4
Controlled oral word association	16	AR	1		3	
Work, volunteer & caregiving activities	17	AS	1		3	
Appetite & weight change	18	AT	1	2	3	4
Medical conditions	19	AU	1		3	
	20	AV	1		3	
	21	AW	1		3	
	22	AX	1		3	
	23	AY	1		3	
Medical conditions and eyesight	24	AZ	1		3	
Fatigue	25	BM	1	2	3	4
Informed care	26	BN	1	2	3	4
	27	BP	1	2	3	4
Decision making about health care	28	BR	1	2	3	4
Preferences about health care	29	BS		2		
	30	BT		2		
Health insurance	31	BU	1			
Well being	32	BV	1	2	3	4
Resilience	33	BW		2		4
Depression screen	34	BX	1	2	3	4
Marital status & household occupancy	35	BY	1			
Social network and support	36	CN	1		3	
Current address and phone (not released)	37	CP	1		3	
Contact information (not released)	38	CQ	1		3	
	39	CR	1		3	
	40	CT	1		3	
Hearing & reliability	41	CU	1	2	3	4

Health ABC Proxy Telephone Interview						
Page Description	Page Number	Variable Prefix	Year 15 Quarter 1	Year 15 Quarter 2	Year 15 Quarter 3	Year 15 Quarter 4
Date of last interview, relationship, how often have contact	1	V1	1	2	3	4
Frequency of contact, stay in bed, cut down, overnight care, home care	2	V2	1	2	3	4
Health concerns, seen health care provider & emergency room	3	V3	1	2	3	4
Failing kidneys, life threatening condition, hypertension, diabetes, fallen	4	V4	1		3	
Heart attach, angina, chest pain, congestive heart failure, stroke	5	V5	1		3	
Cancer, pneumonia, fracture	6	V6	1		3	
Other overnight hospitalization, same day outpatient surgery	7	V7	1		3	
Other condition sees health care professional, memory	8	V8	1		3	
Contact with health care professional, new medicines	9	V9	1	2	3	4
New tests, new treatments/procedures, looked on internet for health care information	10	VA	1	2	3	4
Decisions/choices about health/medical care	11	VB	1	2	3	4
Preferences for decisions about health care	12	VC		2		
Preferences for decisions about health care, power of attorney	13	VD		2		
Living will, hospice/palliative care	14	VE		2		
Walking quarter of a mile, climbing 10 steps	15	VF	1		3	
Difficulty: walking across room, in & out of bed/chairs, bathing/showering	16	VG	1	2	3	4
Difficulty dressing, appetite, weight change	17	VH	1	2	3	4
Symptoms	18	VJ	1	2	3	4
Quality of life, friend/family member had accident/illness/died, depression	19	VK	1	2	3	4
Contact information (not released), move in next 6 months	20	VL	1		3	
Reliability, reason contacted proxy	21	VM	1	2	3	4

HEALTH ABC DATA ANALYSIS FILES

To use the data, please contact the PI at your site.

Contents:

SAS Datasets

Y15Q1_ppt	Year 15 Quarter 1 Participant Telephone Interview data
Y15Q2_ppt	Year 15 Quarter 2 Participant Telephone Interview data
Y15Q3_ppt	Year 15 Quarter 3 Participant Telephone Interview data
Y15Q4_ppt	Year 15 Quarter 4 Participant Telephone Interview data
Y15Q1_proxy	Year 15 Quarter 1 Proxy Telephone Interview data
Y15Q2_proxy	Year 15 Quarter 2 Proxy Telephone Interview data
Y15Q3_proxy	Year 15 Quarter 3 Proxy Telephone Interview data
Y15Q4_proxy	Year 15 Quarter 4 Proxy Telephone Interview data
Y15Q1_calc	Year 15 Quarter 1 Calculated (derived) Variables data
Y15Q2_calc	Year 15 Quarter 2 Calculated (derived) Variables data
Y15Q3_calc	Year 15 Quarter 3 Calculated (derived) Variables data
Y15Q4_calc	Year 15 Quarter 4 Calculated (derived) Variables data

In addition, the following files, not specific to any year but updated each time data are released, can be found at the top of the Current Datasets listing on the Health ABC website under the [Datasets/Documentation](#) link:

PH	Participant History File
Formats	SAS Format Library
MissVis	Missed Follow-up Contact data

PARTICIPANT HISTORY FILE (PH)

1. General description

The PH file contains general information about the participants enrolled in the study. Birthdate, race, and gender data come from the edited HCFA data. The Year 1 clinic visit date was taken from the corrected final participant list provided by the each clinic (with later corrections as necessary). All other dates were taken from the corresponding Clinic Visit workbook, Home Visit workbook, Annual Telephone interview, Proxy Interview, Proxy Contact Home Visit Workbook or Semi-Annual Telephone Contact form; participants who missed a visit have no corresponding date (set to .A). Age at each contact is a calculated variable based on birthdate and that contact date.

There are 3075 observations in the PH file. The demographic breakdown of participants in this dataset is as follows:

African-American Female	729
African-American Male	552
White Female	855
White Male	939
Memphis Participants	1548
Pittsburgh Participants	1527

2. Cross reference of dataset names with exact source

A complete list of variable names can be found under the “Proc Contents for All Datasets” link (search under PH).

3. Dataset structure and contents

The PH file contains a single observation per participant.

Key variables:

HABCID	HABC Enrollment ID without the 2-letter prefix
HCFAID	HCFA ID (as assigned by the Coordinating Center)

4. Condition of data

a. Known data errors: None at this time. The data have been edited. Editing will, however, be ongoing (iterative), so use of the most recent dataset is always advised.

b. Strength and weaknesses of dataset items: If a death has been reported on a Missed Visit Form, an Event Form, or the Report of Death, the participant is listed as deceased in the vital status variable (VStatus) variable. Note: VStatus is vital status **as of the release date of the participant history file**, not as of any particular visit. This variable should only be used to determine the most up-to-date snapshot of vital status in Health ABC as of the file date. Vital status variables (VITALxxM) have been created for each Health ABC contact as follows: If a participant had a particular contact, or if they missed a contact but they were determined to still be alive at the time their contact was due (participant refused the contact, was too ill, etc), then VITALxxM is alive. If they missed a contact and were later discovered to have died before the end of their contact window, then VITALxxM is dead. If they missed a contact and were later discovered to have died after the end of their contact window, the VITALxxM is alive for that contact, but dead for the next. Finally, if they missed a contact without a determination of their vital status (participant could not be located, withdrew, no data for a visit for which they are past the visit window, etc) and no further contact with vital status determination has been made since then, then VITALxxM is missing.

The date of death (DOD) variable represents the best available information about the date of death for deceased participants as of the creation date of participant history file (PH).^{*} If a Report of Death

^{*} Run proc contents in SAS to see creation date of the PH file.

form has been entered for the participant, the date of death from that adjudication form is used. If there is no Report of Death form yet, this information is taken from the Event Form dataset, and is therefore an un-confirmed, un-adjudicated date of death.

c. Missing Value Conventions: See Special Missing Value Codes below for special missing value codes applied.

5. Dataset index formulation and key variable mapping

The PH file is sorted by HABCID, which is a unique identifier for each participant.

6. General strategies for manipulating and merging the data

Because the Health ABC datasets are sorted by Health ABC Enrollment ID, the HABCID variable is most useful for merging with other datasets.

YEAR 15 QUARTERS 1-4 PARTICIPANT TELEPHONE CONTACT DATA (Y15Q1-4 ppt)

1. General description

The four quarterly files, Y15Q1_ppt – Y15Q4_ppt, contain information about the participants enrolled in the study gathered from the Year 15+ Quarterly Participant Telephone interview. If a participant did not have a Year 15+ Quarterly Participant Telephone interview or a Year 15+ Proxy Telephone interview, they should have a Missed Follow-up Contact form that explains why.

The participant history file (PH) has a visit type variable for each quarter. For example, the variable, VTYPE15Q1, indicates the type of visit a participant had in Year 15 Quarter 1. A value of 14 in any Year 15 visit type variable indicates the participant had a participant interview. A value of 15 means a proxy interview was conducted. Participant interview data can be found in the quarterly participant datasets (Y15Q1_ppt – Y15Q4_ppt). Proxy data is in one of the proxy files (Y15Q1_proxy – Y15Q4_proxy). If the PH visit type variable equals 3, 4 or 5, the participant did not have a Year 15+ contact and their Missed Follow-up Contact data can be found in the MISSVIS dataset.

2. Cross reference of dataset names with exact source

A complete list of variable names can be found under the “Proc Contents for All Datasets” link (search under the quarter of interest, e.g., for Year 15 Quarter 1 use “Y15Q1_ppt”). Variable names can also be found on the annotated forms. Variables appearing on the forms in blue are included in a dataset. Variables in gray that are crossed out are not released.

3. Dataset structure and contents

The Y15Q1_ppt – Y15Q4_ppt files contains a single observation per participant.

Key variables:

HABCID HABC Enrollment ID without the 2-letter prefix

4. Condition of data

a. Known data errors: None at this time.

b. Strength and weaknesses of dataset items: Calculated (derived) variables have been created to complement those created for Years 1 through 14. To avoid confusion, these variables are listed in the respective quarterly calculated dataset, e.g. Y15Q1_calc for Year 15 Quarter 1.

c. Missing Value Conventions: See Special Missing Value Codes on below for special missing value codes applied.

5. Dataset index formulation and key variable mapping

The Y15Q1_ppt – Y15Q4_ppt files are sorted by HABCID, which is a unique identifier for each participant.

6. General strategies for manipulating and merging the data

Because the Health ABC datasets are sorted by Health ABC Enrollment ID, the HABCID variable is most useful for merging with other datasets.

YEAR 15 QUARTERS 1-4 PROXY TELEPHONE CONTACT DATA (Y15Q1-4 proxy)

1. General description

As the Health ABC cohort has aged, some participants have begun having cognitive difficulties that prevent their being able to answer interview questions, and in some cases, being able to understand the instructions for some measurements. In response to this situation, a new form was created near the end of Year 3, the Proxy Interview.

2. Cross reference of dataset names with exact source

A complete list of variable names can be found under the “Proc Contents for All Datasets” link (search under the quarter of interest, e.g., for Year 15 Quarter 1 use “Y15Q1_proxy”). Variable names can also be found on the annotated forms. Variables appearing on the forms in blue are included in a dataset. Variables in gray that are crossed out are not released.

3. Dataset structure and contents

The Y15Q1_proxy – Y15Q4_proxy files contain a single observation per participant.

Key variables:

HABCID HABC Enrollment ID without the 2-letter prefix

4. Condition of data

a. Known data errors: None at this time.

b. Strength and weaknesses of dataset items: No calculated variables have been created yet for this dataset. The only standard calculated variables that might apply to these data are the self-reported function variables; however, it has been reported that proxies over-report functional limitation (Elam, et al. Am J Public Health. 1991; 81:1127), and therefore more consideration needs to be applied to how these variables should be calculated.

c. Missing Value Conventions: See Special Missing Value Codes below for special missing value codes applied.

5. Dataset index formulation and key variable mapping

The Y15Q1_proxy – Y15Q4_proxy files are sorted by HABCID, which is a unique identifier for each participant.

6. General strategies for manipulating and merging the data

Because the Health ABC datasets are sorted by Health ABC Enrollment ID, the HABCID variable is most useful for merging with other datasets.

MISSED FOLLOW-UP CONTACT DATA (MissVis)

1. General description

The MissVis file contains information about the participants who have missed a follow-up contact (died, refused, lost to follow-up, etc.) at any time during the study, up through the Year 15 Quarter 4 contact. If a participant missed an annual or semi-annual or quarterly contact, they should have a Missed Follow-up Contact form that explains why.

2. Cross reference of dataset names with exact source

A complete list of variable names can be found under the “Proc Contents for All Datasets” link (search under MissVis). Variable names can also be found on the annotated forms.

3. Dataset structure and contents

The MissVis file contains multiple observations per participant.

Key variables:

HABCID	HABC Enrollment ID without the 2-letter prefix
SITE	HABC Clinic site: 1=Memphis; 2=Pittsburgh *
BJID2	Contact missed (9=6-month, 2=Year 2, 10=18-month, 3=Year 3, 11=30-month, 4=Year 4, 12=42-month, 5=Year 5, 13=54-month, 6=Year 6, 14=66-month, 7=Year 7, 15=78-month, 16=Year 8, 17=90-month, 18=96-month, 19=102-month, 20=Year 10, 21=114-month, 22=Year 11, 23=126-month, 24=132-month, 25=138-month, 26=144-month, 27=150-month, 28=156-month, (29, the 162-month contact was canceled, so

* Must link to Health ABC participant history file (PH) to add this variable.

data for that contact was not collected), 30=Year 15 Quarter 1, 31=Year 15 Quarter 2, 32=Year 15 Quarter 3, 33=Year 15 Quarter 4)

4. Condition of data

a. Known data errors: None at this time..

b. Strength and weaknesses of dataset items: If a participant missed a visit due to death or withdrawal from the study, the Missed Follow-up Contact corresponding to the first contact missed for this reason is usually the last Missed Follow-up Contact for that participant. That is, field centers were instructed not to continue completing Missed Follow-up Contacts for each subsequent contact missed after the death of a participant or their withdrawal from the study. If a participant could not be located at one contact and therefore had a Missed Follow-up Contact completed for that contact, then subsequently was found to have died before that contact, the death was recorded on a Missed Follow-up Contact form for the subsequent contact. That is, the Missed Follow-up Contact information reflects the status of the participant as known to the field center at the time of the scheduled contact. Missed Follow-up Contact data should not be used to determine approximate date of death, nor even numbers of participants who had died as of a particular follow-up contact. The best information available at the time of the data analysis file regarding date of death can be found in the Participant History file (PH, DOD, see page 3). If a participant was found to have both a Missed Follow-up Contact form for a particular contact and the corresponding contact forms (e.g. Annual Telephone Interview, Proxy Interview, Semi-Annual Follow-Up Contact form or Year 15+ quarterly telephone interview), the Missed Follow-up Contact form data for that participant were deleted from the analysis file.

c. Missing Value Conventions: See Special Missing Value Codes below for special missing value codes applied.

5. Dataset index formulation and key variable mapping

The MissVis file is sorted by HABCID, which is a unique identifier for each participant. The combination of HABCID and BJID2 is a unique identifier for a participant/contact record in this dataset.

6. General strategies for manipulating and merging the data

Because the Health ABC datasets are sorted by Health ABC Enrollment ID, the HABCID variable is most useful for merging with other datasets. The MissVis file must first be subsetted by BJID2 to the contact desired before merging with a contact-specific, one-record-per-participant dataset.

SPECIAL MISSING VALUE CODES

SAS allows for stratification of missing values. The following missing values have been assigned:

- . = 'Missing Form'
- .A = 'A:Not Applicable'
- .E = 'E:Recoded to Missing'
- .F = 'F:Variable Missing from Form'
- .L = 'L:Permanently Lost'

.M = 'M:Missing'
.N = 'N:Not Required'
.T = 'T:Missing Due to Technical Problems' (reading center data only)
.U = 'UUnacceptable'

Description

. : Missing Form

Used when a value is missing because the entire form has not been entered.

A: Not Applicable

Used when a value is missing but the value is not required (due to simple skip pattern logic).

E: Recoded to Missing

Used to flag that a value was entered originally but should not have been (due to a skip pattern logic error) and that the value has been recoded to missing.

F: Variable Missing from Form

Used to flag a variable that was not originally on the form (form was revised during the visit year) and therefore there is no value for this participant.

L: Permanently Lost

Used to flag a tracking variable when a measurement involving a Reading Center was done, but the data either never made it to the Reading Center, or was lost at the Reading Center. A list of lost measurements, along with a brief description of what happened, can be found in Appendix I of the Reading Center Dataset documentation. Every effort was made to track down these data before they were declared "permanently lost" and the flag assigned.

M:Missing

Used to flag missing values when the value is required (i.e., true missing values).

N:Not Required

Used when a value is missing but the value is not required (not due to simple skip pattern logic). For example, for checkbox variables which are "Check all that apply" each one, individually, is not required. In these cases, a summary calculated variable (not included on the dataset) was used to edit missing responses. Some variables whose skip pattern logic is non-standard (i.e., the skip pattern involves several variables and forms) also have .N flags when missing, whether or not a response was required due to the skip pattern.

T:Missing Due to Technical Problems

Used when a value is missing from the Reading Center dataset due to technical difficulties. An explanation of when this value has been assigned can be found under Strengths and Weaknesses of (Reading Center) dataset items for each Reading Center (Reading Center data documentation; Substudy documentation).

U:Unacceptable

Used with certain Reading Center data when the data exist but cannot be used, for example, DXA data when the whole scan has been reviewed as unacceptable.

General Strategies for Using Special Missing Values

In SAS, when using special missing values in logical expressions, the missing value is no longer only equal to ‘.’ To express a value equal to missing, the code should be written: `<= .z` or alternately: `le .z`

To express a value not equal to missing, the code should be written `>.z` or alternately: `gt .z`
.Z is the greatest value of missing available in SAS.

DROPPED VARIABLES

A number of variables appear on the annotated forms but will not be found in the datasets. Dropped variables are grayed out with a line through them on the annotated forms. There are several reasons why variables were dropped:

1. Participant confidentiality: identifying information such as participant name, acrostic, address, telephone number, etc. are omitted from the analysis file. All participants are instead identified by the HABC Enrollment ID# (HABCID).
2. Bookkeeping variables: a number of variables were put on the forms merely for bookkeeping purposes and are extremely unlikely to be useful for analysis. If an investigator notices that such a variable has been dropped and believes it should not have been, s/he should feel free to contact the Coordinating Center (HABCHelp@psg.ucsf.edu) and let us know that it should be included in future datasets.
3. Redundant variables: in many cases the same information was collected numerous times. We have made a special effort to clean up one version of each of these; and to avoid analysis using uncleaned data, the uncleaned version is omitted.

LISTINGS

A PDF listing of the SAS proc contents printout for all SAS datasets can be found under the “Proc Contents for All Datasets” link.

A text file, Formats.lst, showing all formats and value descriptions (e.g.: 1=White, 2=Black) contained in the SAS Format Library can also be downloaded. Click on FormatsList under the List of Current Datasets on the Health ABC website.

Appendix 1 - SAS Programming Code for Calculated (derived) Variables

```
*****
*****
*****
*****;

*****
*   Saved as   \\Fu-hsing-c\HABC\HABC_SAS\Calculated Variables\Programs\Year 15\Phact_Y15.sas
*
*   Todd Glasser 06-27-2013 - Modified year 14 program for use in Year 15+
*****;

*--- NOTE: INPUT VARS ONLY COLLECTED IN QUARTER 1 *****;
%let yyqq=Y15Q1_;
%let yq=%substr(&yyqq,1,5);

%include '\\Fu-hsing-c\habc\habc_sas\programs\initV8.sas';
libname calc '\\Fu-hsing-c\habc\habc_sas\calculated variables\datasets\Year 15';
libname y15 "\\Fu-hsing-c\habc\HABC_SAS\Data Analysis File\Programs\Year 15 Data";

data calc.&yyqq.phact(KEEP=HABCID &yyqq.MCKKWK &yyqq.FSKKWK &yyqq.TWKKWK &yyqq.WALKTIME &yyqq.WALKCAT
&yyqq.BKTWTIME
                                &yyqq.BRISK180 &yyqq.BRISK90 &yyqq.WSKKWK );

    set y15.&yyqq.ppt;
    by habcid;

    Label &yyqq.MCKKWK='KCAL/KG/WEEK DOING MAJOR CHORES'
          &yyqq.FSKKWK='KCAL/KG/WEEK CLIMBING STAIRS'
          &yyqq.TWKKWK='KCAL/KG/WEEK TOTAL WALKING'
          &yyqq.WALKTIME='MINUTES WALKING/WEEK'
          &yyqq.WALKCAT='MINUTES WALKING/WEEK CATEGORY'
          &yyqq.BKTWTIME='MINUTES WALKING BRISKLY/WEEK'
          &yyqq.BRISK90='WALKS BRISKLY >= 90 MIN/WEEK'
          &yyqq.BRISK180='WALKS BRISKLY >= 180 MIN/WEEK'
          &yyqq.WSKKWK='KCAL/KG/WEEK - WALKING + STAIRS';

    *****----- MAJOR CHORES (17);
    IF &yyqq.AJHC12MO<=.z AND &yyqq.AJHC7DAY<=.z THEN &yyqq.MCKKWK=. ;
    IF (&yyqq.AJHC12MO=0 OR &yyqq.AJHC12MO IN (7,8)) AND (&yyqq.AJHC7DAY<=.z OR &yyqq.AJHC7DAY=0) THEN
&yyqq.MCKKWK=0 ;
    IF &yyqq.AJHC7DAY=0 THEN &yyqq.MCKKWK=0 ;
    IF &yyqq.AJHC12MO=1 AND &yyqq.AJHC7DAY<=.z THEN &yyqq.MCKKWK=0 ;
    IF &yyqq.AJHC7DAY=1 THEN &yyqq.MCKKWK=3.5*&yyqq.AJHCTIM/60 ;
    *IMPUTED MISSING CODE;
```

```

IF &yyqq.MCKKWK LT 0 THEN DO;
IF &yyqq.AJHC12MO=1 AND &yyqq.AJHC7DAY=1 AND (&yyqq.AJHCTIM<=.z OR &yyqq.AJHCDK=-1) THEN &yyqq.AJHCTIM=120;
IF &yyqq.AJHC7DAY=1 THEN &yyqq.MCKKWK=3.5*&yyqq.AJHCTIM/60;
END;

*****----- WALKING FOR EXERCISE (18);
IF &yyqq.AKEWPACE=1 THEN &yyqq.EWMET=4.0;
IF &yyqq.AKEWPACE=2 THEN &yyqq.EWMET=3.0;
IF &yyqq.AKEWPACE=3 THEN &yyqq.EWMET=2.0;
IF &yyqq.AKEW12MO<=.z AND &yyqq.AKEW7DAY<=.z THEN &yyqq.EWKKWK=.;
else do ;
    IF (&yyqq.AKEW12MO IN (0,7,8)) AND (&yyqq.AKEW7DAY<=.z) THEN &yyqq.EWKKWK=0;
    IF &yyqq.AKEW7DAY=0 THEN &yyqq.EWKKWK=0;
    IF &yyqq.AKEW12MO=1 AND &yyqq.AKEW7DAY<=.z THEN &yyqq.EWKKWK=0;
    IF &yyqq.AKEW7DAY=1 THEN &yyqq.EWKKWK=&yyqq.EWMET*&yyqq.AKEWTIME*&yyqq.AKEWTIM/60;
end ;
*IMPUTED MISSING CODE;
IF &yyqq.EWKKWK LT 0 THEN DO;
    IF &yyqq.AKEWTIME > 0 AND &yyqq.AKEWTIM > 0 AND (&yyqq.AKEWPACE<=.z OR &yyqq.AKEWPACE=8) THEN
&yyqq.EWMET=3.0;
    IF &yyqq.AKEWTIME > 0 AND (&yyqq.AKEWTIM<=.z OR &yyqq.AKEWTDK=-1) THEN &yyqq.AKEWTIM=35;
    IF (&yyqq.AKEWTIME<=.z OR &yyqq.AKEWTMDK=-1) AND &yyqq.AKEWTIM > 0 THEN &yyqq.AKEWTIME=4;
    IF &yyqq.AKEW7DAY=1 THEN &yyqq.EWKKWK=&yyqq.EWMET*&yyqq.AKEWTIME*&yyqq.AKEWTIM/60;
END;

*****----- Fix calculation of BKAKEWTIME EK 9/25/03;
IF &yyqq.AKEWPACE=1 THEN BKAKEWTIME=&yyqq.AKEWTIME*&yyqq.AKEWTIM;
    ELSE IF &yyqq.AKEWPACE>1 THEN BKAKEWTIME=0;
    ELSE IF &yyqq.TWKKWK=0 THEN BKAKEWTIME=0;
&yyqq.BKTWTIME=BKAKEWTIME;
IF &yyqq.BKTWTIME=0 THEN &yyqq.BRISK90=0;
    ELSE IF 0<&yyqq.BKTWTIME<90 THEN &yyqq.BRISK90=0;
    ELSE IF &yyqq.BKTWTIME GE 90 THEN &yyqq.BRISK90=1;
IF &yyqq.BKTWTIME=0 THEN &yyqq.BRISK180=0;
    ELSE IF 0<&yyqq.BKTWTIME<180 THEN &yyqq.BRISK180=0;
    ELSE IF &yyqq.BKTWTIME GE 180 THEN &yyqq.BRISK180=1;

*****----- New variable for total walking;
&yyqq.TWKKWK=&yyqq.EWKKWK;
IF &yyqq.TWKKWK=0 THEN &yyqq.WALKTIME=0;
    ELSE IF &yyqq.TWKKWK>0 THEN &yyqq.WALKTIME=&yyqq.AKEWTIME*&yyqq.AKEWTIM;
IF &yyqq.WALKTIME=0 THEN &yyqq.WALKCAT=0;
    ELSE IF 0<&yyqq.WALKTIME<150 THEN &yyqq.WALKCAT=1;
    ELSE IF &yyqq.WALKTIME GE 150 THEN &yyqq.WALKCAT=2;

```

```

*****----- CLIMBING STAIRS (19);
* ASSUME UP/DOWN 1 FLIGHT TAKES 30 SECONDS AND ONE ADDITIONAL MET FOR CARRYING A LOAD;
* Cutoff for flights of stairs - only for purpose of calculated variables ***;
*** See email from MLE dated 4/6/01 ***;
IF &yyqq.AKFSNUM>210 then &yyqq.AKFSNUM=210;
IF &yyqq.AKFSLOAD>210 THEN &yyqq.AKFSLOAD=210;
IF &yyqq.AKFS12MO<=.z AND &yyqq.AKFS7DAY<=.z THEN &yyqq.FSKKWK=.;
IF (&yyqq.AKFS12MO=0 OR &yyqq.AKFS12MO=8 OR &yyqq.AKFS12MO=7) AND (&yyqq.AKFS7DAY=8 OR &yyqq.AKFS7DAY<=.z)
THEN &yyqq.FSKKWK=0;
IF &yyqq.AKFS7DAY=0 THEN &yyqq.FSKKWK=0;
IF &yyqq.AKFS12MO=1 AND (&yyqq.AKFS7DAY<=.z OR &yyqq.AKFS7DAY=8) THEN &yyqq.FSKKWK=0;
IF &yyqq.AKFS7DAY=1 THEN &yyqq.FSKKWK=(4.0*&yyqq.AKFSNUM/120) + (1.0*&yyqq.AKFSLOAD/120);
*IMPUTED MISSING CODE;
IF &yyqq.FSKKWK LT 0 THEN DO;
IF &yyqq.AKFS7DAY=1 AND &yyqq.AKFSNUM GT 0 AND (&yyqq.AKFSLOAD<=.z OR &yyqq.AKFSLOADK=-1) THEN
&yyqq.AKFSLOAD=0;
IF &yyqq.AKFS7DAY=1 AND (&yyqq.AKFSNUM<=.z OR &yyqq.AKFSNUMD=-1) AND &yyqq.AKFSLOAD GT 0 THEN
&yyqq.AKFSNUM=&yyqq.AKFSLOAD;
IF &yyqq.AKFS7DAY=1 THEN &yyqq.FSKKWK=(4.0*&yyqq.AKFSNUM/120) + (1.0*&yyqq.AKFSLOAD/120); END;

*****----- WALKING AND STAIRS;
&yyqq.WSKKWK=SUM(OF &yyqq.FSKKWK &yyqq.TWKKWK);

format &yyqq.BRISK180 &yyqq.BRISK90 yndk. &yyqq.WALKCAT walk9x.;
run;

%MACRO SKIP ;
proc contents data=CALC.&yyqq.phact varnum;
title4 "CALC.&yyqq.phact - Final Dataset";
run;
proc freq data=CALC.&yyqq.phact;
tables &yyqq.BRISK90 &yyqq.BRISK180 &yyqq.WALKCAT/missing;
run;
options nolabel;
proc means data=CALC.&yyqq.phact n nmiss min max mean maxdec=0;
run;
options label;
%MEND SKIP ;
%skip;
***check for duplicates***;
data dupes&yyqq;
set CALC.&yyqq.phact;
by habcid;

```

```
if not(first.habcid and last.habcid);
run;
```

```
proc print data=dupes&yyqq;
title4 "Duplicates in phact &yyqq.";
run;
```

```
*****
*****
*****
*****;
```

```
***** Saved as:
* \\Fu-hsing-c\HABC\HABC_SAS\Calculated Variables\Programs\Year 1f\srfcn_&yyqq..sas
*
* Todd Glasser 06/27/2013 - Modified year 14 prog for use in Years 15+
*****;
```

```
***** HABC Year 15+ SELF-REPORT PHYSICAL
```

```
FUNCTION CALC VAR CODE
```

```
** Creates the following variables:
** &yyqq.EASEQM: ease of walking 1/4 mile **;
** &yyqq.EASE1M: ease of walking 1 mile **;
** &yyqq.EASE1F: ease of climbing 1 flight **;
** &yyqq.EASE2F: ease of climbing 2 flights **;
** &yyqq.EASE10P: ease of lifting/carrying 10 lbs **;
** &yyqq.EASE20P: ease of lifting/carrying 20 lbs **;
** &yyqq.EASEUP: ease of standing up from chair w/o using arms **;
** &yyqq.CSAINDEX: climbing stairs ability index **;
** &yyqq.WKAINDEX: walking ability index **;
** &yyqq.LCAINDEX: lift/carry ability index **;
** **;
** Adapted from F&yyqq.n Harris Y2 code: srfcn_y2.sas **;
*****;
```

```
*DM LOG 'CLEAR' ; *DM OUTPUT 'CLEAR' ;
```

```
%include '\\Fu-hsing-c\HABC\HABC_SAS\Programs\initV8.sas';
libname calc14 '\\Fu-hsing-c\hbc\hbc_sas\calculated variables\datasets\Year 14';
libname calc_y10 '\\Fu-hsing-c\hbc\hbc_sas\calculated variables\datasets\Year 10';
libname calc '\\Fu-hsing-c\hbc\hbc_sas\calculated variables\datasets\Year 15';
libname y15 "\\Fu-hsing-c\hbc\HABC_SAS\Data Analysis File\Programs\Year 15 Data";
```

```
options orientation=landscape; *ls=90 ps=56 formchar='|----+|----+|-\<>*' nodate nofmterr;
```

```

%macro srfcn(yyqq=);
%let yq=%substr(&yyqq,1,5);

proc contents data=y15.&yyqq.ppt noprint out=c(keep=name);
run;
data pages;
  retain AD AE AF AH;
  set c end=eof;
  if substr(uppercase(name),7,2)='AD' then AD='Y';
  if substr(uppercase(name),7,2)='AE' then AE='Y';
  if substr(uppercase(name),7,2)='AF' then AF='Y';
  if substr(uppercase(name),7,2)='AH' then AH='Y';
  if eof;
  keep AD AE AF AH;
run;
title "Pages &yq";
proc print data=pages;
run;
data _null_;
  set pages;
  call symput('AD',AD);
  call symput('AE',AE);
  call symput('AF',AF);
  call symput('AH',AH);
run;
%put pages YQ=&YQ AD=&AD AE=&AE AF=&AF AH=&AH;

data ppt&yq;
  set y15.&yyqq.ppt;
  keep habcid;
  %if &AD=Y %then %do;
    keep &yyqq.ADDwqmyrn &yyqq.ADDwqmdf /*&yyqq.mmrs*/ &yyqq.ADDwqmez &yyqq.ADDwlmyn &yyqq.ADDwlmez;
  %end;
  %if &AE=Y %then %do;
    keep &yyqq.AEdw10yn &yyqq.AEdif &yyqq.AEdw10ez &yyqq.AEDW20YN &yyqq.AEDW20EZ;
  %end;
  %if &AF=Y %then %do;
    keep &yyqq.AFDIFSTA &yyqq.AFEZSTA &yyqq.AFDSTAMT;
  %end;
  %if &AH=Y %then %do;
    keep &yyqq.AHDif10 &yyqq.AHD10AMT &yyqq.AHEZ10LB &yyqq.AHD20LBS &yyqq.AHEZ20LB;
  %end;
run;
data calc.&yyqq.srfcn;
  set ppt&yq;

```

```
keep HABCID;
```

```
*&yyqq.EASEQM: CODE FOR EASE OF WALKING 1/4 MILE  
IF EASE LEVEL MISSING, EASE LEVEL WALKING 1 MILE WAS ASSIGNED. IF NO EASE LEVEL FOR WALKING  
1M WAS CODED AND &yyqq.ADDwlmyn WAS YES, DONT KNOW, OR MISSING THEN NOT THAT EASY WAS ASSIGNED. IF NO  
EASE LEVEL FOR WALKING 1M WAS CODED AND &yyqq.ADDwlmyn WAS NO AND &yyqq.ADDwlmz WAS MISSING THEN  
SOMEWHAT  
EASY WAS ASSIGNED. THE 4 CASES WHO REPORTED DIFFICULTY WERE RECODED TO NOT THAT EASY FOR QM  
AND DIFFICULTY FOR 1M;
```

```
%if &AD=Y %then %do;
```

```
*IF &yyqq.ADDwqmyn=9 AND (&yyqq.mmrs>0 and &yyqq.mmrs ne 22) THEN &yyqq.ADDwqmyn=1;  
IF &yyqq.ADDwqmez=8 and &yyqq.ADDwqmyn ne 8 THEN &yyqq.ADDwqmez=2;  
IF &yyqq.ADDwqmdf=8 and &yyqq.ADDwqmyn ne 8 THEN &yyqq.ADDwqmdf=2;  
IF &yyqq.ADDwqmyn IN (0,7,8) AND &yyqq.ADDwqmez<0 THEN &yyqq.EASEQM=4;  
ELSE IF &yyqq.ADDwqmyn IN (0,7,8) THEN &yyqq.EASEQM=7-&yyqq.ADDwqmez;  
IF (&yyqq.ADDwqmyn=0 AND &yyqq.ADDwqmez<0) THEN &yyqq.EASEQM=7-&yyqq.ADDwlmz;  
IF (&yyqq.ADDwqmyn=0 AND &yyqq.ADDwqmez<0) & &yyqq.ADDwlmz<0 & (&yyqq.ADDwlmyn in(1,8) OR  
    &yyqq.ADDwlmyn<=.z) THEN &yyqq.EASEQM=4;  
IF (&yyqq.ADDwqmyn=0 & &yyqq.ADDwqmez<0) & (&yyqq.ADDwlmyn=0 & &yyqq.ADDwlmz<=.z) THEN &yyqq.EASEQM=5;  
IF &yyqq.ADDwqmyn=1 THEN &yyqq.EASEQM=4-&yyqq.ADDwqmdf;  
  
IF &yyqq.ADDwqmyn=1 AND &yyqq.ADDwqmdf<0 THEN &yyqq.EASEQM=7-&yyqq.ADDwqmez;  
IF &yyqq.ADDwqmyn=1 AND &yyqq.ADDwqmdf<0 AND &yyqq.ADDwqmez<0 THEN &yyqq.EASEQM=2;  
IF &yyqq.ADDwqmyn<0 AND &yyqq.ADDwqmdf>0 THEN &yyqq.EASEQM=4-&yyqq.ADDwqmdf;  
IF &yyqq.ADDwqmyn<0 AND &yyqq.ADDwqmez>0 THEN &yyqq.EASEQM=7-&yyqq.ADDwqmez;  
  
if &yyqq.ADDwqmyn=8 and &yyqq.ADDwqmez=8 then &yyqq.EASEQM=4;  
IF &yyqq.ADDwqmyn=9 /*AND (&yyqq.mmrs<0 or &yyqq.mmrs=22)*/ THEN &yyqq.EASEQM=.;
```

```
*&yyqq.EASE1M: CODE FOR EASE OF WALKING 1 MILE. AS SELF-REPORTED WALKING ABILITY IS A KEY DEPENDENT  
MEASURE IN HEALTH ABC, MISSING VALUES WERE RECODED TO NON-MISSING WHENEVER A REASON  
GUESS COULD BE MADE BASED ON OTHER AVAILABLE INFORMATION. SEE BELOW FOR CODE AND  
RECODING RULES FOR MISSING VALUES;
```

```
keep &yyqq.EASEQM;  
format &yyqq.EASEQM spmiss. ;  
LABEL &yyqq.EASEQM = 'EASE WALKING 1/4 MILE, 6=VERY EASY';
```

```
IF &yyqq.ADDwlmyn=1 THEN &yyqq.EASE1M=0;  
ELSE IF &yyqq.ADDwlmz=3 THEN &yyqq.EASE1M=1;  
ELSE IF &yyqq.ADDwlmz=2 THEN &yyqq.EASE1M=2;
```

```
ELSE IF &yyqq.ADDwlmez=1 THEN &yyqq.EASE1M=3;
```

```
*MISSING VALUE RECODES;
```

```
IF &yyqq.ADDwlmyn=8 AND (&yyqq.ADDwlmez<0 OR &yyqq.ADDwlmez=8) THEN &yyqq.EASE1M=1;  
IF &yyqq.ADDwlmyn<=.z AND &yyqq.ADDwlmez<=.z AND &yyqq.EASEQM=4 THEN &yyqq.EASE1M=0;  
IF &yyqq.ADDwlmyn<=.z AND &yyqq.ADDwlmez<=.z AND &yyqq.EASEQM=5 THEN &yyqq.EASE1M=1;  
IF &yyqq.ADDwlmyn<=.z AND &yyqq.ADDwlmez<=.z AND &yyqq.EASEQM=6 THEN &yyqq.EASE1M=2;  
IF &yyqq.ADDwlmyn=0 AND &yyqq.ADDwlmez<=.z AND &yyqq.EASEQM=4 THEN &yyqq.EASE1M=0;  
IF &yyqq.ADDwlmyn=0 AND &yyqq.ADDwlmez<=.z AND &yyqq.EASEQM=5 THEN &yyqq.EASE1M=1;  
IF &yyqq.ADDwlmyn=0 AND &yyqq.ADDwlmez<=.z AND &yyqq.EASEQM=6 THEN &yyqq.EASE1M=2;  
IF &yyqq.ADDwqmyn=1 THEN &yyqq.EASE1M=0;
```

```
keep &yyqq.EASE1M;  
format &yyqq.EASE1M spmiss. ;  
LABEL &yyqq.EASE1M = 'EASE WALKING 1 MILE, 3=VERY EASY' ;
```

```
*IN SOME CASES THE AMOUNT OF EASE REPORTED FOR WALKING 1 MILE WAS GREATER THAN  
THE AMOUNT OF EASE REPORTED FOR 1/4 MILE. THE CODE BELOW DOWNG&yyqq.DES THE EASE  
LEVEL REPORTED FOR 1M TO THAT REPORTED FOR 1/4M;
```

```
IF &yyqq.EASEQM GT 0 AND ((&yyqq.EASEQM-3) LT &yyqq.EASE1M) THEN DO ;  
IF &yyqq.EASEQM LE 3 THEN &yyqq.EASE1M=0; ELSE  
&yyqq.EASE1M=&yyqq.EASEQM-3;  
END;
```

```
if &yyqq.ADDwlmyn=8 and &yyqq.ADDwlmez in (1,2,3) then &yyqq.EASE1M=4-&yyqq.ADDwlmez;  
if &yyqq.ADDwlmyn=0 and &yyqq.ADDwlmez=8 then &yyqq.EASE1M=&yyqq.EASEQM-4;
```

```
*CODE TO CREATE A SUMMARY MEASURE OF SELF-REPORTED WALKING ABILITY. THIS  
SCALE WILL ULTIMATELY &yyqq.NGE FROM 0 TO 9, BUT AT BASELINE IT &yyqq.NGES FROM  
4-9, AS NO ONE HAS DIFFICULTY. THIS SCALE CAN BE CREATED IN TWO WAYS: 1) KEYING  
OFF THE RESPONSE TO &yyqq.EASE1M AND ONLY LOOKING AT &yyqq.EASEQM WHEN DIFFICULTY IS  
REPORTED FOR 1M OR 2) SUMMING THE RESPONSES TO THE TWO QUESTIONS. I  
DID IT BOTH WAYS AND FOUND THE LATTER APPROACH SUPERIOR;
```

```
&yyqq.WKAINDEX=&yyqq.EASEQM + &yyqq.EASE1M;
```

```
keep &yyqq.WKAINDEX;  
  
format &yyqq.WKAINDEX spmiss. ;  
  
LABEL &yyqq.WKAINDEX = 'WALKING ABILITY INDEX, 9=BEST' ;
```

```

    %end;

%if &AE=Y %then %do;

*CODE FOR EASE OF CLIMBING STAIRS;

*&yyqq.EASE1F;

IF &yyqq.AEdw10yn=9 THEN DO; &yyqq.EASE1F=.; &yyqq.EASE2F=.; END;
IF &yyqq.AEdw10ez=8 THEN &yyqq.AEdw10ez=2;
IF &yyqq.AEdif=8 THEN &yyqq.AEdif=2;
&yyqq.EASE1F=7-&yyqq.AEdw10ez;
IF &yyqq.AEdw10ez<0 THEN &yyqq.EASE1F=7-&yyqq.AEDW20EZ;
IF &yyqq.AEdw10yn=1 THEN &yyqq.EASE1F=4-&yyqq.AEdif;
IF &yyqq.AEdw10yn=1 AND (&yyqq.AEdif=8 OR &yyqq.AEdif<0) THEN &yyqq.EASE1F=2;
IF &yyqq.AEdw10yn=0 AND &yyqq.AEdw10ez<0 AND &yyqq.AEdw10ez<0 THEN &yyqq.EASE1F=4;

IF &yyqq.EASE1F=. & &yyqq.AEdw10yn NE 9 & &yyqq.AEdw10yn>0 & (&yyqq.AEDW20YN=1 OR
    &yyqq.AEDW20YN=8 OR &yyqq.AEDW20YN<=.z ) THEN &yyqq.EASE1F=4;
IF &yyqq.EASE1F=. AND &yyqq.AEDW20YN=0 AND &yyqq.AEDW20EZ<=.z THEN &yyqq.EASE1F=5;

*&yyqq.EASE2F;

IF &yyqq.AEDW20YN=1 THEN &yyqq.EASE2F=0;
ELSE IF &yyqq.AEDW20EZ=3 THEN &yyqq.EASE2F=1;
ELSE IF &yyqq.AEDW20EZ=2 THEN &yyqq.EASE2F=2;
ELSE IF &yyqq.AEDW20EZ=1 THEN &yyqq.EASE2F=3;
IF (&yyqq.AEDW20YN=8 OR &yyqq.AEDW20YN<0)AND (&yyqq.AEDW20EZ<0 OR &yyqq.AEDW20EZ=8) AND &yyqq.EASE1F>0
THEN &yyqq.EASE2F=1;
IF &yyqq.AEDW20YN<=.z AND &yyqq.AEDW20EZ<=.z AND &yyqq.EASE1F=4 THEN &yyqq.EASE2F=0;
IF &yyqq.AEDW20YN<=.z AND &yyqq.AEDW20EZ<=.z AND &yyqq.EASE1F=5 THEN &yyqq.EASE2F=1;
IF &yyqq.AEDW20YN<=.z AND &yyqq.AEDW20EZ<=.z AND &yyqq.EASE1F=6 THEN &yyqq.EASE2F=2;
IF &yyqq.AEDW20YN=0 AND (&yyqq.AEDW20EZ<=.z OR &yyqq.AEDW20EZ=8) AND &yyqq.EASE1F>0 THEN
&yyqq.EASE2F=&yyqq.EASE1F-4;
IF &yyqq.AEdw10yn=1 THEN &yyqq.EASE2F=0;

IF .z<&yyqq.EASE1F<=3 THEN &yyqq.EASE2F=0; ELSE
IF (&yyqq.EASE1F-3) LT &yyqq.EASE2F THEN &yyqq.EASE2F=&yyqq.EASE1F-3;

*CODE TO CREATE A SUMMARY MEASURE OF SELF-REPORTED STAIR CLIMBING ABILITY. THIS SCALE WILL
&yyqq.NGE FROM 0 TO 9, BUT AT BASELINE IT &yyqq.NGES FROM 4-9;

```

```

&yyqq.CSAINDEX=&yyqq.EASE1F + &yyqq.EASE2F;

keep &yyqq.EASE1F &yyqq.EASE2F &yyqq.CSAINDEX;

format &yyqq.EASE1F &yyqq.EASE2F &yyqq.CSAINDEX spmiss.;
label &yyqq.EASE1F='EASE CLIMBING 1 FLIGHT, 6=VERY EASY'
      &yyqq.EASE2F='EASE CLIMBING 2 FLIGHTS, 3=VERY EASY'
      &yyqq.CSAINDEX='CLIMBING STAIRS ABILITY INDEX, 9=BEST';

%end;

%if &AH=Y %then %do;

*CODE FOR LIFTING/CARRYING;

IF &yyqq.AHdif10=0 AND (&yyqq.AHEZ10LB=8 OR &yyqq.AHEZ10LB<0) THEN &yyqq.EASE10P=5;
ELSE IF &yyqq.AHdif10 IN (0,7,8) and &yyqq.AHEZ10LB ne 8 THEN &yyqq.EASE10P=7-&yyqq.AHEZ10LB;
else if &yyqq.AHdif10 in (0,7,8) and &yyqq.AHEZ10LB=8 then &yyqq.EASE10P=4;
IF &yyqq.AHdif10=1 AND (&yyqq.AHD10AMT=8 OR &yyqq.AHD10AMT<0) THEN &yyqq.EASE10P=.;
ELSE IF &yyqq.AHdif10=1 THEN &yyqq.EASE10P=4 - &yyqq.AHD10AMT;
IF &yyqq.AHdif10<=.z & &yyqq.AHD10AMT>0 & &yyqq.AHEZ10LB<0 THEN &yyqq.EASE10P=4 - &yyqq.AHD10AMT;
IF &yyqq.AHdif10<=.z & &yyqq.AHD10AMT<0 & &yyqq.AHEZ10LB>0 THEN &yyqq.EASE10P=7 - &yyqq.AHEZ10LB;
IF &yyqq.EASE10P=. & &yyqq.AHD20LBS NE 1 & &yyqq.AHD20LBS NE 8 THEN &yyqq.EASE10P=7-&yyqq.AHEZ20LB;
IF &yyqq.EASE10P=. & &yyqq.AHD20LBS=0 & (&yyqq.AHEZ20LB LT 0 OR &yyqq.AHD20LBS=8) THEN &yyqq.EASE10P=6;
IF &yyqq.EASE10P=. AND &yyqq.AHdif10=0 AND &yyqq.AHD20LBS=1 THEN &yyqq.EASE10P=4;
IF &yyqq.EASE10P=. AND &yyqq.AHdif10=8 AND &yyqq.AHD20LBS=1 THEN &yyqq.EASE10P=4;
IF &yyqq.EASE10P=. & &yyqq.AHdif10 IN (0,8) & (&yyqq.AHD20LBS<0 OR &yyqq.AHD20LBS=8) &
    (&yyqq.AHEZ20LB<0 OR &yyqq.AHEZ20LB=8) THEN &yyqq.EASE10P=4;
IF &yyqq.EASE10P=. AND &yyqq.AHdif10=0 AND &yyqq.AHEZ10LB<=.z AND &yyqq.AHD10AMT>0 THEN
    &yyqq.EASE10P=4 - &yyqq.AHD10AMT;

*IF &yyqq.AHEZ20LB=8 and &yyqq.AHD20LBS ne 8 THEN &yyqq.AHEZ20LB=2;
&yyqq.EASE20P=4-&yyqq.AHEZ20LB;
if &yyqq.AHD20LBS=8 and (&yyqq.AHEZ20LB=8 or &yyqq.AHEZ20LB<0) then &yyqq.EASE20P=1;
if &yyqq.AHD20LBS=0 and &yyqq.AHEZ20LB=8 then &yyqq.EASE20P=&yyqq.EASE10P-4;

IF &yyqq.AHD20LBS=1 THEN &yyqq.EASE20P=0;
IF (0 LE &yyqq.EASE10P LE 3) THEN &yyqq.EASE20P=0;

IF (&yyqq.AHD20LBS<=.z OR &yyqq.AHD20LBS=0) and (&yyqq.AHEZ20LB<=.z OR &yyqq.AHEZ20LB=8)
    AND &yyqq.EASE10P=4 THEN &yyqq.EASE20P=0;
IF (&yyqq.AHD20LBS=0 OR &yyqq.AHD20LBS<=.z) AND (&yyqq.AHEZ20LB<=.z OR &yyqq.AHEZ20LB=8)

```

```

        AND &yyqq.EASE10P=6 THEN &yyqq.EASE20P=2;
IF (&yyqq.AHD20LBS=0 OR &yyqq.AHD20LBS<=.z) AND (&yyqq.AHEZ20LB<=.z OR &yyqq.AHEZ20LB=8)
    AND &yyqq.EASE10P=5 THEN &yyqq.EASE20P=1;

IF (&yyqq.EASE10P-3) LT &yyqq.EASE20P THEN DO;
    &yyqq.EASE20P=&yyqq.EASE10P-3;
    IF &yyqq.EASE20P<0 THEN &yyqq.EASE20P=0;
END;

*CODE FOR SUMMARY INDEX;

&yyqq.LCAINDEX=&yyqq.EASE10P + &yyqq.EASE20P;

    keep &yyqq.EASE10P &yyqq.EASE20P &yyqq.LCAINDEX;

    format &yyqq.EASE10P &yyqq.EASE20P &yyqq.LCAINDEX spmiss.;
    label &yyqq.EASE10P='EASE LIFT/CARRY 10 LBS, 6=VERY EASY'
        &yyqq.EASE20P='EASE LIFT/CARRY 20 LBS, 3=VERY EASY'
        &yyqq.LCAINDEX='LIFT/CARRY ABILITY INDEX, 9=BEST';
    %end;

    %if &AF=Y %then %do;

*OTHER FUNCTION MEASURES;

IF &yyqq.AFEZSTA=8 and &yyqq.AFDIFSTA ne 8 THEN &yyqq.AFEZSTA=2;
IF &yyqq.AFDSTAMT=8 and &yyqq.AFDIFSTA ne 8 THEN &yyqq.AFDSTAMT=2;
IF &yyqq.AFDIFSTA IN (0,7,8) THEN &yyqq.EASEUP=7 - &yyqq.AFEZSTA;
IF &yyqq.AFDIFSTA=1 THEN &yyqq.EASEUP=4 - &yyqq.AFDSTAMT;
if &yyqq.AFDIFSTA=8 and &yyqq.AFEZSTA=8 then &yyqq.EASEUP=4;
IF &yyqq.AFDIFSTA<=.z AND &yyqq.AFDSTAMT>0 AND &yyqq.AFEZSTA<0 THEN &yyqq.EASEUP=4 - &yyqq.AFDSTAMT;
IF &yyqq.AFDIFSTA=8 AND &yyqq.AFEZSTA<0 AND &yyqq.AFDSTAMT<0 THEN &yyqq.EASEUP=4;
IF &yyqq.AFDIFSTA=0 AND &yyqq.AFEZSTA<=.z THEN &yyqq.EASEUP=5;
IF &yyqq.AFDIFSTA=0 & &yyqq.AFEZSTA<=.z & &yyqq.AFDSTAMT >0 THEN &yyqq.EASEUP=4 - &yyqq.AFDSTAMT;

    keep &yyqq.EASEUP;
    format &yyqq.EASEUP spmiss.;
    LABEL &yyqq.EASEUP='EASE STANDING FROM CHAIR WITHOUT USING ARMS, 6=VERY EASY';
    %end;

run;
title "&yq - Self Reported Physical Functioning" ;
proc contents data=calc.&yyqq.srfcn;
run;

```

```

options nolabel;
proc means data=calc.&yyqq.srfcn n nmiss min max mean maxdec=0;
run;
options label;
***check for duplicates***;
data dupes&yyqq;
  set calc.&yyqq.srfcn;
  by habcid;
  if not(first.habcid and last.habcid);
run;
proc print data=dupes&yyqq;
title4 "Duplicates in SRFCN &yyqq.";
run;
%mend srfcn;
%srfcn(yyqq=Y15Q1_);
%srfcn(yyqq=Y15Q2_);
%srfcn(yyqq=Y15Q3_);
%srfcn(yyqq=Y15Q4_);
%macro skip ;
options orientation=landscape label;
data match;
  merge calc14.y14srfcn(in=y14)
        calc.&yyqq.srfcn(in=y15) current.ph(keep=habcid);
  by habcid;
  FORMAT EASE1F EASE2F EASEQM EASE1M EASE10P EASE20P EASEUP CSAINDEX WKAINDEX
         LCAINDEX SPMISS. /*TIRED1F TIREDQM YNDK.**/;
  FORMAT &yyqq.EASE1F &yyqq.EASE2F &yyqq.EASEQM &yyqq.EASE1M &yyqq.EASE10P &yyqq.EASE20P &yyqq.EASEUP
&yyqq.CSAINDEX &yyqq.WKAINDEX
         &yyqq.LCAINDEX SPMISS. /*TIRED1F TIREDQM YNDK.**/;

run;
title "Year 14 vs &yq - Self Reported Physical Functioning" ;
proc freq data=match ;
  tables EASE1F*&yyqq.EASE1F EASE2F*&yyqq.EASE2F EASE1M*&yyqq.EASE1M EASE10P*&yyqq.EASE10P
        EASE20P*&yyqq.EASE20P EASEUP*&yyqq.EASEUP CSAINDEX*&yyqq.CSAINDEX WKAINDEX*&yyqq.WKAINDEX
LCAINDEX*&yyqq.LCAINDEX / list missing;
run;
data matchy10;
  merge calc14.y14srfcn(in=y14)
        calc_y10.y10srfcn(in=y15 rename=(EASE1F=EASE1Fy10 EASE2F=EASE2Fy10 EASEQM=EASEQMMy10 EASE1M=EASE1My10
        EASE10P=EASE10Py10 EASE20P=EASE20Py10 EASEUP=EASEUPy10 CSAINDEX=CSAINDEXy10 WKAINDEX=WKAINDEXy10))
  current.ph(keep=habcid);
  by habcid;
run;
title "Year 14 vs Year 10 - Self Reported Physical Functioning" ;

```

```

proc freq data=matchy10 ;
  tables EASE1F*EASE1Fy10 EASE2F*EASE2Fy10 EASEQM*EASEQMy10 EASE1M*EASE1My10
         EASE10P*EASE10Py10 EASE20P*EASE20Py10 EASEUP*EASEUPy10 CSAINDEX*CSAINDEXy10 WKAINDEX*WKAINDEXy10 / list
missing;
run;
%mend skip ;
*%skip;

*****
*****
*****
*****;

*DM LOG 'CLEAR' ; *DM OUTPUT 'CLEAR' ;
options symbolgen macrogen mprint orientation=landscape ls=240 pageno=1;
%include '\\Fu-hsing-c\HABC\HABC_SAS\Programs\initV8.sas';
libname habcsql odbc user=lakin dsn=sqlservr database=habc schema=dbo
  DBINDEX=yes DBMAX_TEXT=200 preserve_tab_names=yes;
libname calc '\\Fu-hsing-c\habc\habc_sas\calculated variables\datasets\Year 15';
libname y15 "\\Fu-hsing-c\habc\HABC_SAS\Data Analysis File\Programs\Year 15 Data";
%let test=N; *Y;
%macro calc_TICS(yyqq=);
data pptPagel3&yyqq;
  array p13 {*} &yyqq.anticfn &yyqq.anticln &yyqq.anticmon &yyqq.anticday &yyqq.anticyr &yyqq.anticdyw
    &yyqq.anticdys &yyqq.anticadn &yyqq.anticads &yyqq.anticadc &yyqq.anticadt &yyqq.anticadz;
  set y15.&yyqq.ppt(keep=habcid &yyqq.anticfn &yyqq.anticln &yyqq.anticmon &yyqq.anticday &yyqq.anticyr
&yyqq.anticdyw
    &yyqq.anticdys &yyqq.anticadn &yyqq.anticads &yyqq.anticadc &yyqq.anticadt &yyqq.anticadz);
  do i=1 to dim(p13);
    if i=1 then do;
      p13score=.;
      p13disabled=.;
    end;
    if p13{i}=1 then do;
      p13score+1;
    end;
    else if p13{i}=3 then p13disabled+1;
  end;
  %if %upcase(&test)=N %then %do;
    keep habcid p13Score p13disabled;
  %end;
  drop i;
run;
title "&yyqq PptPagel3 - TICS";
proc freq data=pptPagel3&yyqq;

```

```

tables p13Score p13disabled / missing;
run;
%if %upcase(&test)=Y %then %do;
proc print data=pptPage13&yyqq(where=(p13Score^=.) obs=25);
format _all_;
run;
proc print data=pptPage13&yyqq(where=(p13Score=. ) obs=10);
format _all_;
run;
%end;
data pptPage14&yyqq;
set y15.&yyqq.ppt(keep=habcid &yyqq.apticcb &yyqq.apticwr &yyqq.apticwrr &yyqq.apticnb &yyqq.apticnbr
&yyqq.apticcp &yyqq.aptic12 &yyqq.apticpg &yyqq.apticwl);
p14Score=.;
p14disabled=.;
if &yyqq.apticcb=1 then do;
p14Score+2;
end;
else if &yyqq.apticcb=2 then do;
p14Score+1;
end;
else if &yyqq.apticcb=3 then p14disabled+1;
if &yyqq.apticwr>0 then do;
p14Score+&yyqq.apticwr;
end;
else if &yyqq.apticwrr=3 then p14disabled+1;
if &yyqq.apticnb>0 then do;
p14Score+&yyqq.apticnb;
end;
else if &yyqq.apticnbr=3 then p14disabled+1;
if &yyqq.apticcp=1 then do;
p14Score+1;
end;
else if &yyqq.apticcp=3 then p14disabled+1;
if &yyqq.aptic12=1 then do;
p14Score+1;
end;
else if &yyqq.aptic12=3 then p14disabled+1;
if &yyqq.apticpg=1 then do;
p14Score+1;
end;
else if &yyqq.apticpg=3 then p14disabled+1;
if &yyqq.apticwl=1 then do;
p14Score+1;
end;

```

```

end;
else if &yyqq.apticwl=3 then p14disabled+1;
%if %upcase(&test)=N %then %do;
    keep habcid p14Score p14disabled;
%end;
run;
title "&yyqq PptPage14 - TICS";
proc freq data=pptPage14&yyqq;
    tables p14Score p14disabled / list missing;
    %if %upcase(&test)=Y %then %do;
        tables &yyqq.apticwr*&yyqq.apticwrr &yyqq.apticnb*&yyqq.apticnbr / list missing;
    %end;
run;
%if %upcase(&test)=Y %then %do;
    proc print data=pptPage14&yyqq(where=(p14Score^=.) obs=25);
        format _all_;
    run;
    proc print data=pptPage14&yyqq(where=(p14Score=. ) obs=10);
        format _all_;
    run;
%end;
data pptPage15&yyqq;
    array p15 {*} &yyqq.aqticiab &yyqq.aqticme &yyqq.aqticpre &yyqq.aqticvp &yyqq.aqticwow &yyqq.aqticwog;
    set y15.&yyqq.ppt(keep=habcid &yyqq.aqticiab &yyqq.aqticme &yyqq.aqticpre &yyqq.aqticvp
        &yyqq.aqticftp &yyqq.aqticwow &yyqq.aqticwog);
    do i=1 to dim(p15);
        if i=1 then do;
            p15score=.;
            p15disabled=.;
        end;
        if p15{i}=1 then do;
            p15score+1;
        end;
        else if p15{i}=3 then p15disabled+1;
    end;
    if &yyqq.aqticftp=1 then do;
        p15Score+2;
    end;
    else if &yyqq.aqticftp=2 then do;
        p15Score+1;
    end;
    else if &yyqq.aqticftp=3 then p15disabled+1;
%if %upcase(&test)=N %then %do;
    keep habcid p15Score p15disabled;
%end;

```

```

drop i;
run;
title "&yyqq PptPage15 - TICS";
proc freq data=pptPage15&yyqq;
  tables p15Score p15disabled / missing;
run;
%if %upcase(&test)=Y %then %do;
  proc print data=pptPage15&yyqq(where=(p15Score^=.) obs=25);
    format _all_;
  run;
  proc print data=pptPage15&yyqq(where=(p15Score=.) obs=10);
    format _all_;
  run;
%end;
proc sort data=pptPage13&yyqq;
  by habcid;
run;
proc sort data=pptPage14&yyqq;
  by habcid;
run;
proc sort data=pptPage15&yyqq;
  by habcid;
run;
data tics&yyqq;
  merge pptPage13&yyqq pptPage14&yyqq pptPage15&yyqq;
  by habcid;
  &yyqq.TICSscore=sum(of p13score p14Score p15Score);
  &yyqq.TICSdisabled=sum(of p13disabled p14disabled p15disabled);
  label &yyqq.TICSscore="Total Score on Telephone Interview Cognitive Status Test (TICS)";
  label &yyqq.TICSdisabled="TICS Number of Responses where a Disability prevented ppt from responding";
  %if %upcase(&test)=N %then %do;
    keep habcid &yyqq.TICSscore &yyqq.TICSdisabled;
  %end;
run;
title "&yyqq Combined Pages 13, 14, and 15 - TICS Total Score";
proc freq data=tics&yyqq;
  table &yyqq.TICSscore &yyqq.TICSdisabled / missing;
run;
%if %upcase(&test)=Y %then %do;
  proc print data=tics&yyqq(where=(&yyqq.TICSscore^=.) obs=25);
    var habcid
        &yyqq.TICSscore p13score p14Score p15Score
        &yyqq.TICSdisabled p13disabled p14disabled p15disabled;
  run;
  proc print data=tics&yyqq(where=(&yyqq.TICSscore=.) obs=10);

```

```

var habcid
  &yyqq.TICSscore p13score p14Score p15Score
  &yyqq.TICSdisabled p13disabled p14disabled p15disabled;
run;
%end;
proc sort data=tics&yyqq out=calc.&yyqq.TICS nodupkey;
  by habcid;
run;
title "&yyqq.TICS";
proc contents data=calc.&yyqq.TICS;
run;
options nolabel;
proc means data=calc.&yyqq.TICS n nmiss min max mean sum maxdec=0;
run;
options label;
%mend calc_TICS;
%calc_TICS(yyqq=Y15Q2_);
%calc_TICS(yyqq=Y15Q4_);

```

```

*****
*****
*****
*****;

```