

Documentation for all Data Sets

HEALTH ABC DATA ANALYSIS FILE.....	2
PARTICIPANT HISTORY FILE (PH).....	3
YEAR 9 ANNUAL TELEPHONE CONTACT DATA (Y9Phone).....	6
YEAR 9 PROXY CONTACT DATA (Y9Proxy).....	8
102-MONTH SEMI-ANNUAL TELEPHONE CONTACT DATA (SA102Mo)	9
102-MONTH PROXY CONTACT DATA (SA102Prox)	11
MISSED FOLLOW-UP CONTACT DATA (MissVis).....	12
SPECIAL MISSING VALUE CODES	14
DROPPED VARIABLES.....	15
LISTINGS	16
DATA DICTIONARY	16
Appendix I DROPPED VARIABLES AND ALTERNATES	17
Appendix II 102-MONTH CALCULATED VARIABLES	22

HEALTH ABC DATA ANALYSIS FILE

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

Contents:

SAS Datasets

Y9Phone	Year 9 Annual Telephone Interview data
Y9Proxy	Year 9 Proxy Interview data
SA102Mo	102-month Semi-Annual Telephone Contact Data
SA102Prox	102-month Proxy Interview data
MissVis	Missed Follow-up Contact data
Y9Calc	Year 9 calculated (derived) variables

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

PARTICIPANT HISTORY FILE (PH)

1. General description

The PH file contains general information about the participants enrolled in the study. Variables included are:

HABCID	Health ABC Enrollment ID# without the 2-letter prefix
HCFAID	HCFA Screening ID (as assigned by the Coordinating Center)
DOB	Date of Birth
DOD	Date of Death
GENDER	Gender (1=Male; 2=Female)
RACE	Race (1=White; 2=Black)
RACEGEN	Race-gender group (1=White male, 2=White female, 3=Black male, 4=Black female)
SITE	Clinic Site (1=Memphis, 2=Pittsburgh)
CV1AGE	Age at Year 1 Clinic Visit
CV1DATE	Year 1 Clinic Visit Date
SV06AGE	Age at 6-Month Contact
SV06DATE	6-Month Contact Date
VITAL06M	Vital Status at time of 6-month contact
VTYPE06	Visit Type at 6-month contact
CV2AGE	Age at Year 2 Clinic Visit
CV2DATE	Year 2 Clinic Visit Date
VITAL12M	Vital Status at time of Year 2 (12-month) contact
VTYPE12	Visit Type at Year 2 (12-month) contact
SV18AGE	Age at 18-Month Contact
SV18DATE	18-Month Contact Date
VITAL18M	Vital Status at time of 18-month contact
VTYPE18	Visit Type at 18-month contact
CV3AGE	Age at Year 3 Clinic Visit
CV3DATE	Year 3 Clinic Visit Date
VITAL24M	Vital Status at time of Year 3 (24-month) contact
VTYPE24	Visit Type at Year 3 (24-month) contact
SV30AGE	Age at 30-Month Contact
SV30DATE	30-Month Contact Date
VITAL30M	Vital Status at time of 30-month contact
VTYPE30	Visit Type at 30-month contact
CV4AGE	Age at Year 4 Clinic Visit
CV4DATE	Year 4 Clinic Visit Date
VITAL36M	Vital Status at time of Year 4 (36-month) contact
VTYPE36	Visit Type at Year 4 (36-month) contact
SV42AGE	Age at 42-Month Contact
SV42DATE	42-Month Contact Date
VITAL42M	Vital Status at time of 42-month contact
VTYPE42	Visit Type at 42-month contact
CV5AGE	Age at Year 5 Clinic Visit
CV5DATE	Year 5 Clinic Visit Date

VITAL48M	Vital Status at time of Year 5 (48-month) contact
VTYPE48	Visit Type at Year 5 (48-month) contact
SV54AGE	Age at 54-Month Contact
SV54DATE	54-Month Contact Date
VITAL54M	Vital Status at time of 54-month contact
VTYPE54	Visit Type at 54-month contact
CV6AGE	Age at Year 6 Clinic Visit
CV6DATE	Year 6 Clinic Visit Date
VITAL60M	Vital Status at time of Year 6 (60-month) contact
VTYPE60	Visit Type at 60-month contact
SV66AGE	Age at 66-Month Contact
SV66DATE	66-Month Contact Date
VITAL66M	Vital Status at time of 66-month contact
VTYPE66	Visit Type at 66-month contact
CV7AGE	Age at Year 7 Clinic Visit
CV7DATE	Year 7 Clinic Visit Date
VITAL72M	Vital Status at time of Year 7 (72-month) contact
VTYPE72	Visit Type at 72-month contact
SV78AGE	Age at 78-Month Contact
SV78DATE	78-Month Contact Date
VITAL78M	Vital Status at time of 78-month contact
VTYPE78	Visit Type at 78-month contact
CV8AGE	Age at Year 8 Clinic Visit
CV8DATE	Year 8 Clinic Visit Date
VITAL84M	Vital Status at time of Year 8 (84-month) contact
VTYPE84	Visit Type at 84-month contact
SV90AGE	Age at 90-Month Contact
SV90DATE	90-Month Contact Date
VITAL90M	Vital Status at time of 90-month contact
VTYPE90	Visit Type at 90-month contact
CV9AGE	Age at Year 9 Clinic Visit
CV9DATE	Year 9 Clinic Visit Date
VITAL96M	Vital Status at time of Year 9 (96-month) contact
VTYPE96	Visit Type at 96-month contact
SV102AGE	Age at 102-Month Contact
SV102DATE	102-Month Contact Date
VITAL102M	Vital Status at time of 102-month contact
VTYPE102	Visit Type at 102-month contact
VERSIONPH	Participant History File Release Date
VStatus	Vital Status (1=Alive, 2=Dead) as of PH file release date

The 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: A new variable RACEGEN has been added to PH to ease analyses by race-gender group (1=White male, 2=White female, 3=Black male, 4=Black female)

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 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 9 ANNUAL TELEPHONE CONTACT DATA (Y9Phone)

1. General description

The Y9Phone file contains information about the participants enrolled in the study gathered from the Year 9 Annual Telephone interview. If a participant did not have a Year 9 Annual Telephone Interview or a Year 9 Proxy Visit, they should have a Missed Follow-up Contact form that explains why. In addition, a variable VISITYPE has been appended to allow the analyst to account for all participants, whether or not they had a Year 9 Annual Telephone Contact. If VISITYPE= 13, the participant's Year 9 data can be found in Y9PHONE. If VISITYPE= 8, the participant's Year 9 data can be found in Y9PROXY. If VISITYPE=3, 4 or 5, the participant did not have a Year 9 contact and their Missed Follow-up Contact data can be found in MISSVIS.

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

There are 3075 observations in the Y9Phone file. The types of visits (VISITYPE) are as follows:

Annual Phone Visit	2038
Proxy Phone Visit	174
Missed/Deceased	745
Missed/Withdrew	16
Missed/Other	102

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 Y9Phone). Variable names can also be found on the annotated forms. Please note that not all variables on the forms are contained in the dataset. All variables not found in the dataset are listed in Dropped Variables and Alternates (Appendix I). Alternate variables to use (if applicable) are also listed.

3. Dataset structure and contents

The Y9Phone file contains a single observation per participant.

Key variables:

HABCID	HABC Enrollment ID without the 2-letter prefix
SITE	HABC Clinic site: 1=Memphis; 2=Pittsburgh*
VISITYPE	(3=Missed (other), 4=Deceased, 5=Withdrew, 8=Proxy Phone; 13=Annual Telephone Interview)
MISSREAS	Reason Y9 Annual Telephone Contact Missed

4. Condition of data

a. Known data errors:

Year 9 Annual Telephone Interview: None at this time.

b. Strength and weaknesses of dataset items:

This dataset does not contain the weight variable WTK, which reports participants’ weight in kilograms, since all interviews were done over the phone we were not able to weigh the participants.

Calculated (derived) variables have been created to complement those created for Years 1 through 9. To avoid confusion, these variables are listed in Y9Calc.

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 Y9Phone file is sorted by HABCID, which is a unique identifier for each participant.

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

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 9 PROXY CONTACT DATA (Y9Proxy)

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. In year 9, there are 174 participants who have a proxy phone visit and can be found in Y9Proxy file.

Note that a “proxy phone” contact is empirically defined as having interview data only, but no physical measurements. This definition was used for the creation of the VISITYPE variable, but was not always followed by the clinics in assigning a value to YACONTAC. Thus VISITYPE is a more reliable variable to use to determine how many proxy contacts have only interview information, vs how many have physical measurements as well.

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 Y9Proxy). Variable names can also be found on the annotated forms. Please note that not all variables on the forms are contained in the dataset. All variables not found in the dataset are listed in Dropped Variables and Alternates (Appendix I). Alternate variables to use (if applicable) are also listed.

3. Dataset structure and contents

The Y9Proxy file contains a single observation per participant.

Key variables:

HABCID	HABC Enrollment ID without the 2-letter prefix
SITE	HABC Clinic site: 1=Memphis; 2=Pittsburgh [*]
VISITYPE	(8=Proxy Phone)

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 Y9Proxy 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.

102-MONTH SEMI-ANNUAL TELEPHONE CONTACT DATA (SA102Mo)

1. General description

The 102-month visit file contains information about the participants enrolled in the study gathered from the 102-Month Semi-Annual Telephone Interview. In some cases, a participant was unable to complete the contact by telephone because of deafness, cognitive impairment, or other reasons, and the information was gathered by proxy. Data for these participants can be found in SA102Prox. If a participant did not have a 102-month semi-annual telephone interview or a 102-month proxy, they should have a Missed Follow-up Contact form that explains why. A variable, VISITYPE, has been added to this dataset to allow the analyst to account for all participants, whether or not they had a 102-month follow-up telephone call. If VISITYPE=7, the participant's 102-month data can be found in SA102Mo. If VISITYPE= 8, the participant's 102-month data can be found in SA102PROX. If

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

VISITYPE=3, 4 or 5, the participant did not have a 102-month contact and their Missed Follow-up Contact data can be found in MissVis.

There are 3075 observations in the SA102Mo file.

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 SA102Mo). Variable names can also be found on the annotated forms. Please note that not all variables on the forms are contained in the dataset. All variables not found in the dataset are listed in Dropped Variables and Alternates (Appendix I). Alternate variables to use (if applicable) are also listed.

3. Dataset structure and contents

The SA102Mo file contains a single observation per participant.

Key variables:

HABCID	HABC Enrollment ID without the 2-letter prefix
SITE	HABC Clinic site: 1=Memphis; 2=Pittsburgh*
VISITYPE	(3=Missed (other), 4=Deceased, 5=Withdrew, 7=SATC Phone, 8=Proxy Phone)
MISSREAS	Reason 102MO Visit Missed

4. Condition of data

a. Known data errors:

None at this time.

b. Strength and weaknesses of dataset items:

Calculated (derived) variables have been added to the dataset (see Appendix II).

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

5. Dataset index formulation and key variable mapping

The SA102Mo 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.

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

102-MONTH PROXY CONTACT DATA (SA102Prox)

1. General description

In some cases, a participant was unable to complete the 102-month contact by telephone because of deafness, cognitive impairment or other reasons. In response to this situation, the Proxy Interview was developed to allow another person to answer for them.

At the 102-month visit, there are 177 participants who have a Proxy Interview and can be found in SA102PROX.

A complete list of variable names can be found under the “Proc Contents for All Datasets” link (search under SA102PROX). Variable names can also be found on the annotated forms. Please note that not all variables on the forms are contained in the dataset. All variables not found in the dataset are listed in Dropped Variables and Alternates (Appendix I). Alternate variables to use (if applicable) are also listed.

3. Dataset structure and contents

The SA102Prox file contains a single observation per participant.

Key variables:

HABCID	HABC Enrollment ID without the 2-letter prefix
SITE	HABC Clinic site: 1=Memphis; 2=Pittsburgh [*]
VISITYPE	(8=Proxy Phone)

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.

Not all the proxy interview questions were asked at the semi-annual telephone contact. Only the “starred” questions were asked during this contact.

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

5. Dataset index formulation and key variable mapping

The SA102PROX file is sorted by HABCID, which is a unique identifier for each participant.

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

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 102-month contact. If a participant missed an annual or semi-annual contact, they should have a Missed Follow-up Contact form that explains why.

The number of Missed Follow-up Contact forms related to each of these visits is shown below:

6-month follow-up contact	23
Year 2 visit	77
18-month semi-annual contact	137
Year 3 visit	95
30-month semi-annual contact	199
Year 4 visit	160
42-month semi-annual contact	201
Year 5 visit	125
54-month semi-annual contact	150
Year 6 visit	144
66-month semi-annual contact	142
Year 7 visit	136
78-month semi-annual contact	170
Year 8 visit	204
90-month semi-annual contact	202
Year 9 visit	176
102-month semi-annual contact	171

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)

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, or Semi-Annual Follow-Up Contact form), 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.

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

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 = 'Unacceptable'

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. These variables are listed in Appendix I. 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.

Appendix I lists the omitted variables in alphabetical order. There are two columns; the first, entitled "Variable omitted," lists the name of the variable not included in the dataset. The second column, entitled "Variable to use," lists the cleaned variable, for redundant variables, or the reason the variable was not included.

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.

The following files are zipped together in a self-extracting document also available on the Health ABC website under the Current Datasets listing

Datadict.xls Microsoft Excel 4.0 spreadsheet containing information about all variables included in the SAS datasets

Datadict.dat Tab-delimited text file containing the same information as above.

DATA DICTIONARY (datadict.xls, datadict.txt)

General description

This is a searchable/sortable file that contains all the variables included in the SAS datasets. The following fields are included:

Variable	Variable Name
Label	Description of the variable
Form	Form or reading center origin of the variable
Page	Page number (not applicable for reading center or PH data)
Database	Database location of the variable
Variable Types	Type of variable (text, categorical (numeric), continuous (numeric), date, time, etc.)
Possible Values	Range of possible values associated with the variable (used for range edits)
SAS Format	SAS format assigned to the variable
Page Order	Variable order on the data collection forms (useful for sorting)

General Strategies for Use

The data dictionary is provided in two formats: Excel 2000 and tab-delimited text. They contain exactly the same information. The tab-delimited file was generated to provide easy access to those who do not have Excel version 2000 or higher.

The file is currently sorted by variable name. It can be used to search for details on a particular variable, to group by database or form, or to find a variable location on a form. It is a little more user friendly than the standard SAS proc contents, as it can be sorted, easily searched, and it provides additional details such as page number and form.

Appendix I

DROPPED VARIABLES AND ALTERNATES

Variable omitted	Variable to use
BJID	HABCID
BJACROS	N/A (confidentiality)
BJCONTAC	BJID2
BJSTFID	N/A (bookkeeping)
BJTYPE	BJID2
BJVISIT	BJID2
BLID	HABCID
BLACROS	N/A (confidentiality)
BLINK	N/A (bookkeeping)
BLMOVE	N/A (bookkeeping)
BLREF13A	N/A (bookkeeping)
BLREF13B	N/A (bookkeeping)
BLREF13C	N/A (bookkeeping)
BLREF14A	N/A (bookkeeping)
BLREF14B	N/A (bookkeeping)
BLREF14C	N/A (bookkeeping)
BLREF15A	N/A (bookkeeping)
BLREF15B	N/A (bookkeeping)
BLREF15C	N/A (bookkeeping)
BLREF16A	N/A (bookkeeping)
BLREF16B	N/A (bookkeeping)
BLREF16C	N/A (bookkeeping)
BLREF17A	N/A (bookkeeping)
BLREF17B	N/A (bookkeeping)
BLREF17C	N/A (bookkeeping)
BLREF18A	N/A (bookkeeping)
BLREF18B	N/A (bookkeeping)
BLREF18C	N/A (bookkeeping)
BLREF19A	N/A (bookkeeping)
BLREF19B	N/A (bookkeeping)
BLREF19C	N/A (bookkeeping)
BLREF19D	N/A (bookkeeping)
BLREF19E	N/A (bookkeeping)
BLREF19F	N/A (bookkeeping)
BLREF20A	N/A (bookkeeping)
R2ID	HABCID
R2ACROS	N/A (confidentiality)
R2CONTAC	VISITYPE

Variable omitted	Variable to use
R3ID	HABCID
R3ACROS	N/A (confidentiality)
R3CONTAC	VISITYPE
R4ID	HABCID
R4ACROS	N/A (confidentiality)
R4CONTAC	VISITYPE
R5ID	HABCID
R5ACROS	N/A (confidentiality)
R5CONTAC	VISITYPE
R6ID	HABCID
R6ACROS	N/A (confidentiality)
R6CONTAC	VISITYPE
R7ID	HABCID
R7ACROS	N/A (confidentiality)
R7CONTAC	VISITYPE
R8ID	HABCID
R8ACROS	N/A (confidentiality)
R8CONTAC	VISITYPE
R9ID	HABCID
R9ACROS	N/A (confidentiality)
R9CONTAC	VISITYPE
RAID	HABCID
RAACROS	N/A (confidentiality)
RACONTAC	VISITYPE
RBID	HABCID
RBACROS	N/A (confidentiality)
RBCONTAC	VISITYPE
RBHCHRS	RBHCTIM
RBHCMINS	RBHCTIM
RCEWHR	RCEWTIM
RCEWMINS	RCEWTIM
RCID	HABCID
RCACROS	N/A (confidentiality)
RCCONTAC	VISITYPE
RDID	HABCID
RDACROS	N/A (confidentiality)
RDCONTAC	VISITYPE
RDHIA1HR	RDH1TIME
RDHIA1MN	RDH1TIME
REID	HABCID
REACROS	N/A (confidentiality)
RECONTAC	VISITYPE
REMA1HR	REM1TIME

Variable omitted	Variable to use
REMIA1MN	REM1TIME
RFID	HABCID
RFACROS	N/A (confidentiality)
RFCONTAC	VISITYPE
RGID	HABCID
RGACROS	N/A (confidentiality)
RGCONTAC	VISITYPE
RHID	HABCID
RHACROS	N/A (confidentiality)
RHCONTAC	VISITYPE
RIID	HABCID
RIACROS	N/A (confidentiality)
RICONTAC	VISITYPE
RJID	HABCID
RJACROS	N/A (confidentiality)
RJCONTAC	VISITYPE
RKID	HABCID
RKACROS	N/A (confidentiality)
RKCONTAC	VISITYPE
RKREF39A	N/A (bookkeeping)
RKREF39B	N/A (bookkeeping)
RKREF39C	N/A (bookkeeping)
RKREF40A	N/A (bookkeeping)
RKREF40B	N/A (bookkeeping)
RKREF40C	N/A (bookkeeping)
RLID	HABCID
RLACROS	N/A (confidentiality)
RLCONTAC	VISITYPE
RLREF41A	N/A (bookkeeping)
RLREF41B	N/A (bookkeeping)
RLREF41C	N/A (bookkeeping)
RLREF42A	N/A (bookkeeping)
RLREF42B	N/A (bookkeeping)
RLREF42C	N/A (bookkeeping)
RMID	HABCID
RMACROS	N/A (confidentiality)
RMCONTAC	VISITYPE
RMREF43A	N/A (bookkeeping)
RMREF43B	N/A (bookkeeping)
RMREF43C	N/A (bookkeeping)
RMREF44A	N/A (bookkeeping)
RMREF44B	N/A (bookkeeping)
RMREF44C	N/A (bookkeeping)

Variable omitted	Variable to use
RNID	HABCID
RNACROS	N/A (confidentiality)
RNCONTAC	VISITYPE
RNREF45A	N/A (bookkeeping)
RNREF45B	N/A (bookkeeping)
RNREF45C	N/A (bookkeeping)
RNREF45D	N/A (bookkeeping)
RNREF45E	N/A (bookkeeping)
RNREF45F	N/A (bookkeeping)
RNREF46A	N/A (bookkeeping)
ROID	HABCID
ROACROS	N/A (confidentiality)
ROCONTAC	VISITYPE
RPID	HABCID
RPACROS	N/A (confidentiality)
RPCONTAC	VISITYPE
RUID	HABCID
RUACROS	N/A (confidentiality)
RUCONTAC	VISITYPE
RVID	HABCID
RVACROS	N/A (confidentiality)
RVCONTAC	VISITYPE
RWID	HABCID
RWACROS	N/A (confidentiality)
RWCONTAC	VISITYPE
RXID	HABCID
RXACROS	N/A (confidentiality)
RXCONTAC	VISITYPE
RXADDYN	N/A (bookkeeping)
RXSSESPY	N/A (bookkeeping)
RYID	HABCID
RYACROS	N/A (confidentiality)
RYCONTAC	VISITYPE
RYCIYN	N/A (bookkeeping)
RYKNOK	N/A (bookkeeping)
RYKYN	N/A (bookkeeping)
RZID	HABCID
RZACROS	N/A (confidentiality)
RZCONTAC	VISITYPE
RZPPOA	N/A (bookkeeping)
RZPAYN	N/A (bookkeeping)
RZP2YN	N/A (bookkeeping)
SIID	HABCID

Variable omitted	Variable to use
S1ACROS	N/A (confidentiality)
S1CONTAC	VISITYPE
S1C1YN	N/A (bookkeeping)
T4ID	HABCID
T4ACROS	N/A (confidentiality)
T4CONTAC	VISITYPE
T4FNM	N/A (confidentiality)
T4LNM	N/A (confidentiality)
T4REASON	N/A (bookkeeping)
YAID	HABCID
YAACROS	N/A (confidentiality)
YACONTAC	VISITYPE
YAVISIT	N/A (bookkeeping)
YALINK	N/A (bookkeeping)
YAMOVE	N/A (confidentiality)
YAREF11A	N/A (bookkeeping)
YAREF11B	N/A (bookkeeping)
YAREF11C	N/A (bookkeeping)
YAREF12A	N/A (bookkeeping)
YAREF12B	N/A (bookkeeping)
YAREF12C	N/A (bookkeeping)
YAREF13A	N/A (bookkeeping)
YAREF13B	N/A (bookkeeping)
YAREF13C	N/A (bookkeeping)
YAREF14A	N/A (bookkeeping)
YAREF14B	N/A (bookkeeping)
YAREF14C	N/A (bookkeeping)
YAREF15A	N/A (bookkeeping)
YAREF15B	N/A (bookkeeping)
YAREF15C	N/A (bookkeeping)
YAREF16A	N/A (bookkeeping)
YAREF16B	N/A (bookkeeping)
YAREF16C	N/A (bookkeeping)
YAREF17A	N/A (bookkeeping)
YAREF17B	N/A (bookkeeping)
YAREF17C	N/A (bookkeeping)
YAREF17D	N/A (bookkeeping)
YAREF17E	N/A (bookkeeping)
YAREF17F	N/A (bookkeeping)
YAREF18A	N/A (bookkeeping)

Appendix II

102-MONTH CALCULATED VARIABLES

Variable	Variable Description	Grouping
CSAINDEX	Climbing stairs ability index	Self-Reported Function Calculated Vars.
EASE1F	Ease climbing 1 flight	Self-Reported Function Calculated Vars.
EASE1M	Ease walking 1 mile	Self-Reported Function Calculated Vars.
EASE2F	Ease climbing 2 flights	Self-Reported Function Calculated Vars.
EASEQM	Ease walking 1/4 mile	Self-Reported Function Calculated Vars.
WKAINDEX	Walking ability index	Self-Reported Function Calculated Vars.

Appendix IV
Self-Reported Function Calculated Variables

Using Annual Telephone Interview Variables

Investigator Name: Eleanor Simonsick

Variable	General Description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
EASEQM	Ease walking 1/4 mile	Recode of R5DWQMEZ where 3 is easiest; imputations applied for missing R5DWQMEZ	If R4DWQMYN in (0,7,8) then EASEQM=7-R5DWQMEZ If R4DWQMYN=1 then EASEQM=4-R4DWQMDF	If R4DWQMYN=9 and (R4MNRS>0 or R4MNRS≠22) then R4DWQMYN=1; If R5DWQMEZ=8 and R4DWQMYN≠8 then R5DWQMEZ=2; If R4DWQMDF=8 and R4DWQMYN≠8 then R4DWQMDF=2; If R4DWQMYN in (0,7,8) and R5DWQMEZ<0 then EASEQM=4; If (R4DWQMYN=0 and R5DWQMEZ<0) then EASEQM=7-R5DW1MEZ; if (R4DWQMYN=0 and R5DWQMEZ<0) and R5DW1MEZ<0 and (R5DW1MYN=1 or R5DW1MYN=8 or R5DW1MYN≤Z) then EASEQM=4; if (R4DWQMYN=0 and R5DWQMEZ<0) and (R5DW1MYN=0 and R5DW1MEZ≤Z) then EASEQM=5 if R4DWQMYN=1 and R4DWQMDF<0 then EASEQM=7-R5DWQMEZ; if R4DWQMYN=1 and R4DWQMDF<0 and R5DWQMEZ<0 then EASEQM=2; if R4DWQMYN<0 and R4DWQMDF>0 then EASEQM=4-R4DWQMDF; if R4DWQMYN<0 and R5DWQMEZ>0 then EASEQM=7-R5DWQMEZ; if R4DWQMYN=8 and R5DWQMEZ=8 then EASEQM=4; if R4DWQMYN=9 and R4MNRS<0 and (R4MNRS4<0 or R4MNRS=22) then EASEQM=.	6=very easy 5=somewh at easy 4=not that easy 3=a little difficult 2=somewh at difficult 1=very difficult 0=unable to do

Self-Reported Function Calculated Variables

Variable	General Description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
EASE1M	Ease walking 1 mile	Recode of BLDW1MEZ where 3 is easiest; imputations applied for missing BLDW1MEZ	If BLDW1MYN=1 then EASE1M=0; if BLDW1MEZ=3 then EASE1M=1; if BLDW1MEZ=2 then EASE1M=2; if BLDW1MEZ=1 then EASE1M=3	<p>IF BLDW1MYN=8 AND (BLDW1MEZ<0 OR BLDW1MEZ=8) THEN EASE1M=1; IF BLDW1MYN<=.z AND BLDW1MEZ<=.z AND EASEQM=4 THEN EASE1M=0; IF BLDW1MYN<=.z AND BLDW1MEZ<=.z AND EASEQM=5 THEN EASE1M=1; IF BLDW1MYN<=.z AND BLDW1MEZ<=.z AND EASEQM=6 THEN EASE1M=2; IF BLDW1MYN=0 AND BLDW1MEZ<=.z AND EASEQM=4 THEN EASE1M=0; IF BLDW1MYN=0 AND BLDW1MEZ<=.z AND EASEQM=5 THEN EASE1M=1; IF BLDW1MYN=0 AND BLDW1MEZ<=.z AND EASEQM=6 THEN EASE1M=2; IF BLDWQMYN=1 THEN EASE1M=0;</p> <p>IF EASEQM GT 0 AND ((EASEQM-3) LT EASE1M) THEN DO; IF EASEQM LE 3 THEN EASE1M=0; ELSE EASE1M=EASEQM-3; END; if BLDW1MYN=8 and BLDW1MEZ in (1,2,3) then EASE1M=4- BLDW1MEZ; if BLDW1MYN=0 and BLDW1MEZ=8 then EASE1M=EASEQM- 4;</p>	3=very easy 2=somewhat easy 1=not that easy 0=difficult
WKAINDEX	Walking ability index	Summary measure of self-reported walking ability.	WKAINDEX=EASEQM + EASE1M	Imputed version of component variables used	This scale ranges from 0 (unable) to 9 (very easy)

Self-Reported Function Calculated Variables

Variable	General Description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
EASE1F	Ease Climbing 1 flight	Recode of BLDW10EZ where 3 is easiest; imputations applied for missing BLDW20EZ	EASE1F=7-BLDW10EZ; IF BLDW10YN=1 THEN EASE1F=4-BLDIF;	IF BLDW10YN=9 then EASE1F=.; IF BLDW10EZ=8 THEN BLDW10EZ=2; IF BLDIF=8 THEN BLDIF=2; IF BLDW10EZ<0 THEN EASE1F=7-BLDW20EZ; IF BLDW10YN=1 AND (BLDIF=8 OR BLDIF<0) THEN EASE1F=2; IF BLDW10YN=0 AND BLDW10EZ<0 AND BLDW10EZ<0 THEN EASE1F=4; IF EASE1F=. AND BLDW10YN NE 9 AND BLDW10YN>0 AND (BLDW20YN=1 OR BLDW20YN=8 OR BLDW20YN<=.z) THEN EASE1F=4; IF EASE1F=. AND BLDW20YN=0 AND BLDW20EZ<=.z THEN EASE1F=5;	6=very easy 5=somewhat easy 4=not that easy 3=a little difficult 2=somewhat difficult 1=very difficult 0=unable to do
EASE2F	Ease Climbing 2 flights	Recode of BLDW20EZ where 3 is easiest; imputations applied for missing BLDW20EZ	IF BLDW20YN=1 THEN EASE2F=0; ELSE IF BLDW20EZ=3 THEN EASE2F=1; ELSE IF BLDW20EZ=2 THEN EASE2F=2; ELSE IF BLDW20EZ=1 THEN EASE2F=3;	IF BLDW10YN=9 then EASE1F=.; IF (BLDW20YN=8 OR BLDW20YN<0)AND (BLDW20EZ<0 OR BLDW20EZ=8) AND EASE1F>0 THEN EASE2F=1; IF BLDW20YN<=.z AND BLDW20EZ<=.z AND EASE1F=4 THEN EASE2F=0; IF BLDW20YN<=.z AND BLDW20EZ<=.z AND EASE1F=5 THEN EASE2F=1; IF BLDW20YN<=.z AND BLDW20EZ<=.z AND EASE1F=6 THEN EASE2F=2; IF BLDW20YN=0 AND (BLDW20EZ<=.z OR BLDW20EZ=8) AND EASE1F>0 THEN EASE2F=EASE1F-4; IF BLDW10YN=1 THEN EASE2F=0; IF .z<EASE1F<=3 THEN EASE2F=0; ELSE IF (EASE1F-3) LT EASE2F THEN EASE2F=EASE1F-3;	3=very easy 2=somewhat easy 1=not that easy 0=difficult
CSAINDEX	Climbing stairs ability index	Summary measure of self-reported walking ability.	CSAINDEX=EASE1F + EASE2F	Imputed version of component variables used	This scale ranges from 0 (unable) to 9 (very easy)

* Saved as \\Fu-hsing-c\HABC\HABC_SAS\Calculated
Variables\Programs\SemiAnnual\srfcn_sa102.sas

*

* Lynn Harvey 05/03/07 - Modified year 8 program for use in year 9.5

* Lynn Harvey 10/14/05 - Modified year 6 program for use in year 8

* Lynn Harvey 11/17/06 - Modified to read from Fu-hsing-c instead of ilcavallo

***** ;

** SELF-REPORT FUNCTION PROGRAM CODE;

**

** HABC SEMI ANNUAL TELEPHONE CONTACT SELF-REPORT FUNCTION

** CALC VAR CODE

**

** Creates the following variables for 6mo VISIT:

** EASEQM: ease of walking 1/4 mile

** EASE1M: ease of walking 1 mile

** EASE1F: ease of climbing 1 flight

** EASE2F: ease of climbing 2 flights

** TIREDQM: get tired walking 1/4 mile

** TIRED1F: get tired walking 1 mile

** CSAINDEX: climbing stairs ability index

** WKAINDEX: walking ability index

**

** Adapted from Fran Harris Y2 code: srfcn_y2.sas

** Adapted from Laura Akin Y6 code

*****;

data CALC.SRFCN_102 (keep=HABCID VISIT EASEQM EASE1M EASE1F EASE2F
/*tiredqm tired1f*/ CSAINDEX WKAINDEX);

set habc6.BL2(keep=HABCID BLCONTAC BLdwqmyrn BLdwqmdf BLmnr4
BLdwqmez BLdwqmt2 BLdw1myrn BLdw1mez BLdw10yn BLdif BLmnr2 BLmnr3
BLdw10ez BLdw10wx BLdw20yn BLdw20ez WHERE=(BLCONTAC=19));
rename blcontac=VISIT;

*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 BLDW1MYN WAS YES, DONT KNOW, OR MISSING THEN NOT
THAT EASY WAS ASSIGNED. IF NO

EASE LEVEL FOR WALKING 1M WAS CODED AND BLDW1MYN WAS NO AND
BLDW1MEZ 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 BLDWQMYN=9 AND (BLMNRS>0 OR BLMNRS4 IN(1,2)) THEN BLDWQMYN=1;
 IF BLDWQMEZ=8 and BLdwqmyn ne 8 THEN BLDWQMEZ=2;
 IF BLDWQMDF=8 and BLdwqmyn ne 8 THEN BLDWQMDF=2;
 IF BLDWQMYN IN (0,7,8) THEN EASEQM=7-BLDWQMEZ;
 IF (BLDWQMYN=0 AND BLDWQMEZ<0) THEN EASEQM=7-BLDW1MEZ;
 IF (BLDWQMYN=0 AND BLDWQMEZ<0) AND BLDW1MEZ<0 AND (BLDW1MYN=1
 OR BLDW1MYN=8
 OR BLDW1MYN<=.z) THEN EASEQM=4;
 IF (BLDWQMYN=0 AND BLDWQMEZ<0) AND (BLDW1MYN=0 AND BLDW1MEZ<=.z)
 THEN EASEQM=5;
 IF BLDWQMYN=1 THEN EASEQM=4-BLDWQMDF;

IF BLDWQMYN=1 AND BLDWQMDF<0 THEN EASEQM=7-BLDWQMEZ;
 IF BLDWQMYN=1 AND BLDWQMDF<0 AND BLDWQMEZ<0 THEN EASEQM=2;
 IF BLDWQMYN<0 AND BLDWQMDF>0 THEN EASEQM=4-BLDWQMDF;
 IF BLDWQMYN<0 AND BLDWQMEZ>0 THEN EASEQM=7-BLDWQMEZ;

if BLdwqmyn=8 and BLdwqmez=8 then EASEQM=4;
 IF BLDWQMYN=9 AND BLMNRS<0 AND (BLMNRS4<0 OR BLMNRS4=8) THEN
 EASEQM=.;

*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;

IF BLDW1MYN=1 THEN EASE1M=0;
 ELSE IF BLDW1MEZ=3 THEN EASE1M=1;
 ELSE IF BLDW1MEZ=2 THEN EASE1M=2;
 ELSE IF BLDW1MEZ=1 THEN EASE1M=3;

*MISSING VALUE RECODES;
 IF BLDW1MYN=8 AND (BLDW1MEZ<0 OR BLDW1MEZ=8) THEN EASE1M=1;
 IF BLDW1MYN<=.z AND BLDW1MEZ<=.z AND EASEQM=4 THEN EASE1M=0;
 IF BLDW1MYN<=.z AND BLDW1MEZ<=.z AND EASEQM=5 THEN EASE1M=1;
 IF BLDW1MYN<=.z AND BLDW1MEZ<=.z AND EASEQM=6 THEN EASE1M=2;
 IF BLDW1MYN=0 AND BLDW1MEZ<=.z AND EASEQM=4 THEN EASE1M=0;
 IF BLDW1MYN=0 AND BLDW1MEZ<=.z AND EASEQM=5 THEN EASE1M=1;
 IF BLDW1MYN=0 AND BLDW1MEZ<=.z AND EASEQM=6 THEN EASE1M=2;
 IF BLDWQMYN=1 THEN EASE1M=0;

*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 DOWNGRADES THE EASE LEVEL REPORTED FOR 1M TO THAT REPORTED FOR 1/4M;

IF EASEQM GT 0 AND ((EASEQM-3) LT EASE1M) THEN DO;

IF EASEQM LE 3 THEN EASE1M=0;

ELSE EASE1M=EASEQM-3;

END;

if BLdw1myn=8 and BLdw1mez in (1,2,3) then EASE1M=4-BLdw1mez;

if BLdw1myn=0 and BLdw1mez=8 then EASE1M=EASEQM-4;

*CODE TO CREATE A SUMMARY MEASURE OF SELF-REPORTED WALKING ABILITY. THIS SCALE WILL ULTIMATELY RANGE FROM 0 TO 9, BUT AT BASELINE IT RANGES FROM 4-9, AS NO ONE HAS DIFFICULTY. THIS SCALE CAN BE CREATED IN TWO WAYS: 1) KEYING OFF THE RESPONSE TO EASE1M AND ONLY LOOKING AT 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;

WKAINDEX=EASEQM + EASE1M;

***TIREDQM;

*IF BLDWQMT2<0 THEN TIREDQM=.;

*IF BLDWQMT2=8 OR BLDWQMT2=9 THEN TIREDQM=1;

*IF BLDWQMT2=0 THEN TIREDQM=0;

*IF BLDWQMT2=1 THEN TIREDQM=1;

*CODE FOR EASE OF CLIMBING STAIRS;

*EASE1F;

IF BLDW10YN=9 THEN DO; EASE1F=.; EASE2F=.; END;

IF BLDW10EZ=8 THEN BLDW10EZ=2;

IF BLDIF=8 THEN BLDIF=2;

EASE1F=7-BLDW10EZ;

IF BLDW10EZ<0 THEN EASE1F=7-BLDW20EZ;

IF BLDW10YN=1 THEN EASE1F=4-BLDIF;

IF BLDW10YN=1 AND (BLDIF=8 OR BLDIF<0) THEN EASE1F=2;

IF BLDW10YN=0 AND BLDW10EZ<0 AND BLDW10EZ<0 THEN EASE1F=4;

IF EASE1F=. & BLDW10YN NE 9 & BLDW10YN>0 & (BLDW20YN=1 OR BLDW20YN=8 OR BLDW20YN<=.z) THEN EASE1F=4;

IF EASE1F=. AND BLDW20YN=0 AND BLDW20EZ<=.z THEN EASE1F=5;

```

*EASE2F;
IF BLDW20YN=1 THEN EASE2F=0;
ELSE IF BLDW20EZ=3 THEN EASE2F=1;
ELSE IF BLDW20EZ=2 THEN EASE2F=2;
ELSE IF BLDW20EZ=1 THEN EASE2F=3;
IF (BLDW20YN=8 OR BLDW20YN<0)AND (BLDW20EZ<0 OR BLDW20EZ=8) AND
EASE1F>0 THEN EASE2F=1;
IF BLDW20YN<=.z AND BLDW20EZ<=.z AND EASE1F=4 THEN EASE2F=0;
IF BLDW20YN<=.z AND BLDW20EZ<=.z AND EASE1F=5 THEN EASE2F=1;
IF BLDW20YN<=.z AND BLDW20EZ<=.z AND EASE1F=6 THEN EASE2F=2;
IF BLDW20YN=0 AND (BLDW20EZ<=.z OR BLDW20EZ=8) AND EASE1F>0 THEN
EASE2F=EASE1F-4;
IF BLDW10YN=1 THEN EASE2F=0;

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

```

```

*CODE TO CREATE A SUMMARY MEASURE OF SELF-REPORTED STAIR CLIMBING
ABILITY. THIS SCALE WILL
RANGE FROM 0 TO 9, BUT AT BASELINE IT RANGES FROM 4-9;
CSAINDEX=EASE1F + EASE2F;

```

```

*IF BLDW10WX=8 OR BLDW10WX=9 THEN TIRED1F=1;
*IF BLDW10WX=0 THEN TIRED1F=0;
*IF BLDW10WX=1 THEN TIRED1F=1;

```

LABEL

```

EASEQM='EASE WALKING 1/4 MILE, 6=VERY EASY'
EASE1M='EASE WALKING 1 MILE, 3=VERY EASY'
WKAINDEX='WALKING ABILITY INDEX, 9=BEST'
/*TIREDQM='GETS TIRED WALKING 1/4 MILE, 1=YES'*/
EASE1F='EASE CLIMBING 1 FLIGHT, 6=VERY EASY'
EASE2F='EASE CLIMBING 2 FLIGHTS, 3=VERY EASY'
CSAINDEX='CLIMBING STAIRS ABILITY INDEX, 9=BEST'
/*TIRED1F='GETS TIRED CLIMBING 1 FLIGHT, 1=YES'*/;

```

```

FORMAT EASE1F EASE2F EASEQM EASE1M CSAINDEX WKAINDEX SPMISS.

```

```

/*TIRED1F TIREDQM YNDK.*;/

```

```

RUN;

```

Documentation for Year 9 Calculated Variable Dataset
(Y9Calc)

YEAR 9 CALCULATED VARIABLE DATASET (Y9Calc)..... 31

- Appendix I Calculated Variable List
- Appendix II Lifestyle Calculated Variables
- Appendix III Physical Activity Calculated Variables
- Appendix IV Self-Reported Function Calculated Variables

YEAR 9 CALCULATED VARIABLE DATASET (Y9Calc)

1. General description

The Y9Calc contains 20 variables derived (calculated) as described below. The documentation is grouped by the type of data from which the variables are calculated (physical activity, etc.), and the variables are positioned in the dataset in the same groupings.

There are 3075 observations in the Y9Calc file.

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 Y9Calc) and in Appendix I. NOTE: SOME CALCULATED VARIABLES USE THE SAME VARIABLE NAMES EVERY YEAR. THE ANALYST IS REMINDED TO RENAME VARIABLES BEFORE MERGING WITH OTHER YEARS AS NECESSARY.

3. Dataset structure and contents

The Y9Calc file contains a single observation per participant.

Key variables:

HABCID	Health ABC Enrollment ID# without the 2-letter prefix
GENDER	Gender (1=Male; 2=Female)*
RACE	Race (1=White; 2=Black)*
RACEGEN	Race-gender group (1=White male, 2=White female, 3=Black male, 4=Black female)*
SITE	Clinic Site (1=Memphis, 2=Pittsburgh)*

4. Condition of data

a. Strengths and weaknesses of dataset items: When a calculated variable includes imputations for missing variables, the method of imputation is **bolded** in the documentation.

One person’s physical activity data has been edited since the last release, changing their value of RBMCKKWK.

5. Dataset index formulation and key variable mapping

The Y9Calc 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.

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

Appendix I
CALCULATED VARIABLE LIST
(LINKED)

Variable	Variable Description	Grouping
SMK9	Smoking status at year 9	Lifestyle
BKTWTIME	minutes walking briskly/week	Physical Activity Calculated Vars
BRISK180	walks briskly \geq 180 min/week	Physical Activity Calculated Vars
BRISK90	walks briskly \geq 90 min/week	Physical Activity Calculated Vars
RCFSKKWK	kcal/kg/week climbing stairs	Physical Activity Calculated Vars
RBMCKKWK	Kcal/ kg/ week doing major chores	Physical Activity Calculated Vars
RCTWKKWK	Kcal/ kg/ week total walking	Physical Activity Calculated Vars
WALKCAT	minutes walking/week category	Physical Activity Calculated Vars
WALKTIME	minutes walking/week	Physical Activity Calculated Vars
WSKKWK	kcal/kg/week - walking + stairs	Physical Activity Calculated Vars
CSAINDEX	Climbing stairs ability index	Self-Reported Function Calculated Vars.
EASE10P	Ease lift/carry 10 lbs	Self-Reported Function Calculated Vars.
EASE1F	Ease climbing 1 flight	Self-Reported Function Calculated Vars.
EASE1M	Ease walking 1 mile	Self-Reported Function Calculated Vars.
EASE20P	Ease lift/carry 20 lbs	Self-Reported Function Calculated Vars.
EASE2F	Ease climbing 2 flights	Self-Reported Function Calculated Vars.
EASEQM	Ease walking 1/4 mile	Self-Reported Function Calculated Vars.
EASEUP	Ease rising from chair	Self-Reported Function Calculated Vars.
LCAINDEX	Lift/carry ability index	Self-Reported Function Calculated Vars.
WKAINDEX	Walking ability index	Self-Reported Function Calculated Vars.

Appendix II
Lifestyle Calculated Variables

Using Annual Telephone Interview Variables

Investigator Name: Stephen Kritchevsky

Variable	Descriptive Title	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
SMK9	Smoking Status at Year 9	0 is never smoker; 1 is former smoker; 2 is current smoker	If RGSMOKE=1 then SMK9=1; Else if RGSMOKE = 0 and SMK1=0 and (SMK3=0 or SMK3 le. z) and (SMK5=0 or SMK5 le. z) then SMK9=0; Else if RGSMOKE = 0 and (SMK1 in (1,2) or SMK3 in (1,2) or SMK5 in (1,2)) then SMK9=2; Else SMK9=.M;	If RGSMOKE ≤ .z or RGSMOKE in (7,8) then SMK9=.A	0=Never 1=Current 2=Former

```

*****
* Create calculated variables requested by SKritchevsky for:
* SMK5      Smoking status at baseline
*
* Saved as  \\Ilcavallo\HABC\HABC_SAS\Calculated Variables\Programs\Year
9\smoke_y9.sas
*
* Lynn Harvey  05/02/07 - Modified year 8 program for use in year 9
***** ;

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

%include '\\Fu-hsing-c\habc\habc_sas\programs\initV8.sas';
*libname calc '\\Fu-hsing-c\habc\habc_sas\calculated variables\datasets\year 9';

options ls=159 ps=51 formchar='|----|+|---+=|-\<>*' nodate nofmterr pageno=1
nocenter;

*Pack-Year Calculation;

data y9 ;
    set habcrts.rg (keep=habcid rgsmoke rgcontac  where=(rgcontac=18));
run ;

data y8 ;
    set daf.y8calc (keep=habcid smk8);
run ;

data y5;
    set current.y5calc (keep=habcid smk5);
run;

data y1;
    set current.y1calc (keep=habcid smk1);
run;

data y3;
    set current.y3calc (keep=habcid smk3);
run;

data smoke;
    merge y1 y3 y5 y8 y9 ;
    by habcid;
    if rgsmoke le .z or rgsmoke in (7,8) then SMK9=.A;
    else if rgsmoke=1 then SMK9=1;
*** Use smoking status at baseline and year 3 to determine never/former status
***;
    else if rgsmoke=0 & smk1=0 & (smk3=0 or smk3 le .z) & (smk5=0 or smk5 le .z) &
        (smk8=0 or smk8 le .z)then SMK9=0;
    * LH 11/25/06 modified code below.  Changed from smk3 in (1,3) to smk3 in
(1,2) ;
    else if rgsmoke=0 and (smk1 in (1,2) or smk3 in (1,2) or smk5 in(1,2) or smk8
in(1,2)) then SMK9=2;
    else SMK9=.M;
    ** 0=never, 1=current, 2=former **;
run;

```

```

%macro skip ;
proc freq data=smoke ;
  tables smk9 * rgsmoke * smk1 * smk3 * smk5 * smk8 / list missing nocum ;
run ;

proc print data=smoke (where=(smk9=.a & rgsmoke=.m)) noobs n ;
  var habcid smk9 rgsmoke smk1 smk3 smk5 smk8 ;
run ;
%mend skip ;

data calc.y9smoke;
  set smoke (keep=habcid SMK9);
  label SMK9 = 'Smoking status at Year 9 visit';
  format smk9 ncffmt. ;
run;

%macro skip ;
options ls=108 ps=60 formchar='|----|+|----+=|-\<>*' nodate nofmterr pageno=1
nocenter;
proc contents data=calc.y9smoke;
run;

proc freq data=calc.y9smoke;
  tables smk9;
run;
%mend skip ;

***check for duplicates***;
data dupes;
  set calc.y9smoke;
  by habcid;
  if not(first.habcid and last.habcid);
run;

title4 'Duplicates in Y9SMOKE';
proc print;
run ;
*****no duplicates*****;

```

Appendix III
Physical Activity Calculated Variables
Using Annual Telephone Interview Variables

Investigator Name: Eleanor Simonsick

Step 1: assign met estimates to each activity

Met estimates derive from Ainsworth BE, Haskell WL, Leon AS, Jacobs DR, Montoye HJ, Sallis JF, Paffenbarger RS. Compendium of physical activities: classification of energy costs of human physical activities. Med Sci Sports Exerc 25(1):71-80.

Variable	General description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
RCFSKKWK	kcal/kg/week climbing stairs	Assigns 4.0 kcal/kg/hour of stair climbing plus an additional 1.0 kcal/kg/hour carrying a load (Y9 Qaire, page 12, Q19). (Assumes 1 flight up/down takes 30 seconds.)	If RCFS7DAY=1 then RCFSKKWK = 4.0 x RCFSNUM/120 +1.0 x RCFSLOAD/120; If RCFS7DAY=0 then RCFSKKWK=0;	Correction for outliers: If RCFSNUM>210 then RCFSNUM=210; if RCFSLOAD>210 then RCFSLOAD=210; If RCFS12MO≤.z and RCFS7DAY≤.z then RCFSKKWK=.; if (RCFS12MO in (0,7,8)) and (RCFS7DAY≤.z or RCFS7DAY=8) then RCFSKKWK=0; if RCFS12MO=1 and (RCFS7DAY≤.z or RCFS7DAY=8) then RCFSKKWK=0; if RCFSKKWK<0 then do: if RCFS7DAY=1 and RCFSNUM > 0 and (RCFSLOAD≤.z or RCFSLODK=-1) then RCFSLOAD=0; if RCFS7DAY=1 and (RCFSNUM≤.z or RCFSNUMD=-1) and RCFSLOAD > 0 then RCFSNUM=RCFSLOAD	kkcal/kg/we ek

Variable	General description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
RBMCKKWK	kcal/kg/week doing major chores	Assigns 3.5 kcal/kg/hour doing major chores (Y9 Qaire, page 11, Q17). Note: this question combines activities listed separately at baseline. RBMCKKWK is therefore equivalent to the sum of baseline variables FPPAKKWK and FPHCKKWK.	3.5 x RBHCTIM/60 if RBHC7DAY=0 then RBMCKKWK=0;	If RBHC12MO≤z and RBHC7FAY≤z then RBMCKKWK=. if (RBHC12MO=0 or RBHC12MO in (7,8) and (RBHC7DAY≤z or RBHC7DAY=0) then RBMCKKWK=0; if RBHC12MO=1 and RBHC7DAY≤z then RBMCKKWK=0; if RBMCKKWK<0 then do: if RBHC12MO=1 and RBHC7DAY=1 and (RBHCTIM≤z or RBHCDK=-1) then RBHCTIM=120 (median value for non-missings at baseline; used here for consistency)	kkcal/kg/week

Variable	General description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
RCTWKKWK	kcal/kg/week total walking	Assigns 4.0 kcal/kg/hour walking briskly, 3.0 to walking at moderate pace, and 2.0 to strolling (Y9 Qaire, page 12, Q18 - all walking)	If RCEW7DAY=1 then RCTWKKWK= 4.0 x RCEWTIME* RCEWTIM/60 if RCEWPACE=1, 3.0 x RCEWTIME* RCEWTIM/60 if RCEWPACE=2, 2.0 x RCEWTIME* RCEWTIM/60 if RCEWPACE=3 if RCEW7DAY=0 then RCTWKKWK=0;	If RCEW12MO \leq .z and RCEW7RCY \leq .z then RCTWKKWK=.; if (RCEW12MO in (0,7,8) and RCEW7RCY \leq .z then RCTWKKWK=0; if RCEW12MO=1 and RCEW7RCY \leq .z then RCTWKKWK=0; if RCTWKKWK<0 then do: if RCEWTIME > 0 and RCEWTIM > 0 and (RCEWPACE \leq .z or RCEWPACE=8) then RCTWKKWK=3.0 x RCEWTIME*RCEWTIM/60 (median value at baseline; used here for consistency); if RCEWTIME > 0 and (RCEWTIM \leq .z or RCEWTDK=-1) then RCEWTIM=35 (median value at baseline; used here for consistency); if (RCEWTIME \leq .z or RCEWTMDK=-1) and RCEWTIM > 0 then RCEWTIME=4 (median value at baseline; used here for consistency)	kkcal/kg/week

Step 2: create composite variables

Variable	General description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
WSKKWK	kcal/kg/week - walking + stairs	Sum of exercise walking, other walking, and stair climbing variables	RCTWKKWK + RCFSSKKWK	Imputed version of component variables used; SAS SUM function used to sum all non-missing components	kkcal/kg/week

Variable	General description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
WALKTIME	minutes walking/week	Adds minutes exercise walking and other walking in past week	WALKTIME=(RCEWTIME x RCEWTIM)	If RCTWKKWK=0 then WALKTIME=0;	min
BKTWTIME	minutes walking briskly/week	Sum of minutes brisk exercise walking plus minutes brisk other walking	If RCEWPACE=1 then BKTWTIME=RCEWTIME x RCEWTIM; If RCEWPACE>1 then BKTWTIME=0; If RCEWKKWK=0 then BKTWTIME=0;		min

Step 3: create categorical variables

Variable	General description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
WALKCAT	minutes walking/week category	Categories of time spent on any walking per week	If WALKTIME=0 then WALKCAT=0; if $0 < \text{WALKTIME} < 150$ then WALKCAT=1; if $\text{WALKTIME} \geq 150$ then WALKCAT=2	Imputed version of component variables used	0= 0 min/wk 1= between 0 and 150 min/wk 2= more than 150 min/wk

Variable	General description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
BRISK90	walks briskly \geq 90 min/week	Dichotomous variable for walking briskly \geq 90 min/week vs <90 min/week	If BKTWTIME=0 then BRISK90=0; if 0<BKTWTIME<90 then BRISK90=0; if BKTWTIME \geq 90 then BRISK90=1	Imputed version of component variables used	0=No 1=Yes
BRISK180	walks briskly \geq 180 min/week	Dichotomous variable for walking briskly \geq 180 min/week vs <180 min/week	If BKTWTIME=0 then BRISK180=0; if 0<BKTWTIME<180 then BRISK180=0; if BKTWTIME \geq 180 then BRISK180=1	Imputed version of component variables used	0=No 1=Yes

```

*****
* Saved as \\Fu-hsing-c\HABC\HABC_SAS\Calculated Variables\Programs\Year
9\Phact_y9.sas
*
* Lynn Harvey 05/02/07 - Modified year 8 program for use in year 9
***** ;

```

```

data CALC.y9phact (KEEP=HABCID RBMCKKWK RCFSSKKWK RCTWKKWK WALKTIME WALKCAT
BKTWTIME

```

```

                                BRISK180 BRISK90 WSKKWK );
merge habcrts.rb      (where=(rbcontac=18))
      habcrts.rc (where=(rccontac=18)) ;
by habcid;

```

```

Label RBMCKKWK='KCAL/KG/WEEK DOING MAJOR CHORES'
      RCFSSKKWK='KCAL/KG/WEEK CLIMBING STAIRS'
      RCTWKKWK='KCAL/KG/WEEK TOTAL WALKING'
      WALKTIME='MINUTES WALKING/WEEK'
      WALKCAT='MINUTES WALKING/WEEK CATEGORY'
      BKTWTIME='MINUTES WALKING BRISKLY/WEEK'
      BRISK90='WALKS BRISKLY >= 90 MIN/WEEK'
      BRISK180='WALKS BRISKLY >= 180 MIN/WEEK'
      WSKKWK='KCAL/KG/WEEK - WALKING + STAIRS'

```

```
;
```

```
*MAJOR CHORES;
```

```

IF RBHC12MO<=.z AND RBHC7DAY<=.z THEN RBMCKKWK=.;
IF (RBHC12MO=0 OR RBHC12MO IN (7,8)) AND (RBHC7DAY<=.z OR RBHC7DAY=0) THEN
RBMCKKWK=0;

```

```

* IF RBHC7DAY=0 THEN RBMCKKWK=0;          * LH modified code 5/31/07 ;
IF RBHC7DAY IN(0,8) THEN RBMCKKWK=0;    * LH modified code 5/31/07 ;
IF RBHC12MO=1 AND RBHC7DAY<=.z THEN RBMCKKWK=0;
IF RBHC7DAY=1 THEN RBMCKKWK=3.5*rbhctim/60;

```

```
*IMPUTED MISSING CODE;
```

```
IF RBMCKKWK LT 0 THEN DO;
```

```
IF RBHC12MO=1 AND RBHC7DAY=1 AND (rbhctim<=.z OR RBHCDK=-1) THEN
```

```
rbhctim=120;
```

```
IF RBHC7DAY=1 THEN RBMCKKWK=3.5*rbhctim/60;
```

```
END;
```

```
%macro skip ;
```

```
proc freq data=CALC.y9phact ;
```

```
tables rbmckkwk * rbhc12mo * RBHC7DAY *rbhctim / list missprint nocum ;
```

```
run ;
```

```
proc print data=CALC.y9phact (where=(RBMCKKWK=. & rbhc12mo=1)) noobs n ;
```

```
var RBMCKKWK RBHC12MO RBHC7DAY rbhctim RBHCDK ;
```

```
run ;
```

```
%mend skip ;
```

```
*CLIMBING STAIRS;
```

```

*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 RCFSNUM>210 then RCFSNUM=210;
IF RCFSLOAD>210 THEN RCFSLOAD=210;

IF RCFS12MO<=.z AND RCFS7DAY<=.z THEN RCFSKWK=.;
IF (RCFS12MO=0 OR RCFS12MO=8 OR RCFS12MO=7) AND (RCfs7day=8 OR
RCfs7day<=.z) THEN RCFSKWK=0;
IF RCfs7day=0 THEN RCFSKWK=0;
IF RCFS12MO=1 AND (RCfs7day<=.z OR RCfs7day=8) THEN RCFSKWK=0;
IF RCfs7day=1 THEN RCFSKWK=(4.0*RCFSNUM/120) + (1.0*RCFSLOAD/120);
IF RCFS12MO=1 & RCfs7day=1 & rcfsnum<=.z & (rcfsload <=.z or rcfsload=0)
THEN RCFSKWK=0; * LH added code 5/31/07 ;

*IMPUTED MISSING CODE;
IF RCFSKWK LT 0 THEN DO;
IF RCfs7day=1 AND RCFSNUM GT 0 AND (RCFSLOAD<=.z OR RCFSLODK=-1) THEN
RCFSLOAD=0;
IF RCfs7day=1 AND (RCFSNUM<=.z OR RCFSNUMD=-1) AND RCFSLOAD GT 0 THEN
RCFSNUM=RCFSLOAD;
IF RCfs7day=1 THEN RCFSKWK=(4.0*RCFSNUM/120) + (1.0*RCFSLOAD/120); END;

%macro skip ;
proc freq data=CALC.y9phact ;
* tables RCFSNUM RCFSLOAD / list missprint nocum ;
tables RCFSKWK * RCFS12MO * RCFS7DAY *RCFSNUM * RCFSLOAD / list missprint
nocum ;
run ;

proc print data=CALC.y9phact (where=(RCFSKWK=. & rcfs12mo=1)) noobs n ;
var HABCID RCFSKWK RCFS12MO RCFS7DAY RCFSNUM rcfsnumd RCFSLOAD rcfslodk ;
run ;
%mend skip ;

*WALKING FOR EXERCISE;

IF RCEWPACE=1 THEN RCEWMET=4.0;
IF RCEWPACE=2 THEN RCEWMET=3.0;
IF RCEWPACE=3 THEN RCEWMET=2.0;

IF RCEW12MO<=.z AND RCEW7DAY<=.z THEN RCEWKKWK=.;
else do ;
IF (RCEW12MO IN (0,7,8)) AND (RCEW7DAY<=.z) THEN RCEWKKWK=0;
IF RCEW7DAY=0 THEN RCEWKKWK=0;
IF RCEW12MO=1 AND RCEW7DAY<=.z THEN RCEWKKWK=0;
IF RCEW7DAY=1 THEN RCEWKKWK=RCEWMET*RCEWTIME*RCEWTIM/60;
end ;

*IMPUTED MISSING CODE;
IF RCEWKKWK LT 0 THEN DO;

```

```

        IF RCEWTIME > 0 AND RCEWTIM > 0 AND (RCEWPACE<=.z OR RCEWPACE=8)
THEN RCEWMET=3.0;
        IF RCEWTIME > 0 AND (RCEWTIM<=.z OR RCEWTDK=-1) THEN RCEWTIM=35;
        IF (RCEWTIME<=.z OR RCEWTMDK=-1) AND RCEWTIM > 0 THEN RCEWTIME=4;
        IF RCEW7DAY=1 THEN RCEWKKWK=RCEWMET*RCEWTIME*RCEWTIM/60;
    END;
    IF RCEW12MO=1 AND RCEW7DAY=1 & RCEWKKWK=. then RCEWKKWK=0 ; * LH added
5/31/07 ;

%macro skip ;
proc freq data=CALC.y9phact ;
    tables RCEWKKWK * RCEW12MO * RCEW7DAY / list missprint nocum ;
run ;

proc print data=CALC.y9phact (where=(RCEWKKWK=. & RCEW12MO=1)) noobs n ;
    var HABCID RCEWKKWK RCEW12MO RCEW7DAY RCEWTIME RCEWTMDK RCEWTIM RCEWTDK
RCEWMET RCEWPACE ;
run ;
%mend skip ;

    *New variable for total walking*;
    RCTWKKWK=RCEWKKWK;

    IF RCTWKKWK=0 THEN WALKTIME=0;
    ELSE IF RCTWKKWK>0 THEN WALKTIME=RCEWTIME*RCEWTIM;

    IF WALKTIME=0 THEN WALKCAT=0;
    ELSE IF 0<WALKTIME<150 THEN WALKCAT=1;
    ELSE IF WALKTIME GE 150 THEN WALKCAT=2;

    *updated below to fix calculation of BKEWTIME EK 9/25/03;
    IF RCEWPACE=1 THEN BKEWTIME=RCEWTIME*RCEWTIM;
    ELSE IF RCEWPACE>1 THEN BKEWTIME=0;
    ELSE IF RCTWKKWK=0 THEN BKEWTIME=0;

    BKTWTIME=BKEWTIME;

    IF BKTWTIME=0 THEN BRISK90=0;
    ELSE IF 0<BKTWTIME<90 THEN BRISK90=0;
    ELSE IF BKTWTIME GE 90 THEN BRISK90=1;

    IF BKTWTIME=0 THEN BRISK180=0;
    ELSE IF 0<BKTWTIME<180 THEN BRISK180=0;
    ELSE IF BKTWTIME GE 180 THEN BRISK180=1;

    *WALKING AND STAIRS;

    WSKKWK=SUM(OF RCFSKKWK RCTWKKWK);

    format brisk180 brisk90 yndk. walkcat walk9x.;
run ;

%macro skip ;
options ls=108 ps=60 formchar='|----|+|----+|=|-\|<>*' nocenter pageno=1 nofmterr
mprint macrogen;

proc freq data=CALC.y9phact ;

```

```

tables RCTWKKWK*RCEWKKWK / list missprint nocum ;
tables RCTWKKWK * WALKTIME * RCEWTIME*RCEWTIM / list missprint nocum ;
tables WALKTIME * WALKCAT / list missprint nocum ;
tables BKTWTIME*BKEWTIME / list missprint nocum ;
tables BKTWTIME*BRISK90 / list missprint nocum ;
tables BKTWTIME*BRISK180 / list missprint nocum ;
tables WSKKWK * RCFSKKWK * RCTWKKWK / list missprint nocum ;
run ;
%mend skip ;

```

*COMPOSITE MEASURES. SINCE THE SUM FUNCTION ASSIGNS ZERO TO MISSING VALUES, THE IMPUTED VERSION OF THE COMPONENT VARIABLES WERE USED;

*HOUSEHOLD DUTIES;

*HHKKWK=SUM(OF FAPAKKWK FAHCKKWK FALWKKWK FAGSKKWK FALDKKWK);

*EXERCISE AND RECREATION;

*EXKKWK=SUM(OF FAACKKWK FATRKKWK FAHIKKWK FAMIKKWK);

*WORK, VOLUNTEER, CAREGIVING;

*GRAND TOTAL;

*PROGRAM CODE TO CREATE SUMMARY MEASURES OF PHYSICAL ACTIVITY

BASED ON CALORIC EXPENDITURE. MET ESTIMATES DERIVE FROM AINSWORTH BE, HASKELL WL, LEON AS, JACOBS DR, MONTOYE HJ, SALLIS JF, PAFFENBARGER RS. COMPENDIUM OF PHYSICAL ACTIVITIES: CLASSIFICATION OF ENERGY COSTS OF HUMAN PHYSICAL ACTIVITIES. MED SCI SPORTS EXERC 25(1):71-80;

*STEP 1: CALCULATION OF KCALORIES PER KILOGRAM OF BODY WEIGHT PER WEEK EXPENDED IN EACH ACTIVITY. ASSIGNED MET VALUES AND ASSUMPTIONS FOLLOW. FOR EASE OF COMPUTATION WHILE THE DATA IS STILL DIRTY, KCAL/KG/WK FOR EACH ACTIVITY WILL BE CALCULATED DIRECTLY FROM MINUTES ACTIVITY PERFORMED (OR LOADS WASHED, BAGS OF GROCERIES, FLIGHTS OF STAIRS CLIMBED). IT WILL BE ASSUMED THAT A MISSING VALUE IN THESE FIELDS INDICATES NON-PARTICIPATION AND A VALUE OF ZERO WILL BE ASSIGNED;

*MET ASSIGNMENTS AND OTHER DECISION RULES:

Q16 - GARDENING/YARDWORK = 3.5 (THIS HAS BEEN DOWNGRADED FROM 4.5)

Q17 - MAJOR CHORES = 3.5 (THIS HAS BEEN DOWNGRADED FROM 4.5)

SINCE THESE ARE COMBINED IN LATER ROUNDS THE SAME MET VALUE WAS ASSIGNED

Q18 - LIGHT HOUSE WORK = 2.5

Q19 - GROCERY SHOPPING = 3.5 FOR 4 BAGS PLUS 1.25 FOR THOSE CARRIED AND 1.25 FOR THOSE UNPACKED

Q20 - LAUNDRY - ASSUMING 30 MINUTES OF WORK PER LOAD THEN 1.0 PER LOAD WASHED PLUS .5 PER LOAD PUT AWAY

Q21 - STAIRS = 4.0 WITH THE ASSUMPTION 1 FLIGHT UP/DOWN TAKES 30 SECONDS PLUS 1.0 MET FOR EVERY FLIGHT CLIMBED WITH A LOAD

Q22 - WALKING FOR EXERCISE - 4.0 IF BRISK, 3.0 IF MODERATE, AND 2.0 IF STROLL (DOWNGRADED TO 3.0 AND 2.0)

Q23 - WALKING FOR OTHER REASONS - 4.0 IF BRISK, 3.0 IF MODERATE, AND 2.0 IF STROLL (DOWNGRADED TO 3.0 AND 2.0)

Q24 - AEROBICS = 5.0

Q25 - WEIGHT TRAINING = 6.0 (DOWNGRADED TO 6)

Q26 - HIGH INTENSITY EXERCISE - 4.0 IF LIGHT, 6.0 IF MODERATE, 8.0 IF VIGOROUS (DOWNGRADED FROM 7.0 AND 10.0)

Q27 - MODERATE INTENSITY EXERCISE = 3.0 FOR ALL ACTIVITIES AND INTENSITY LEVELS (DOWNGRADED FROM 4.0)

Q28 - PAID WORK = 1.5 FOR LEVEL 1, 2.0 FOR LEVEL 2, 2.5 FOR LEVEL 3, AND 3.0 FOR LEVEL 4

Q29 - VOLUNTEER WORK = 1.5 FOR LEVEL 1, 2.0 FOR LEVEL 2, 2.5 FOR LEVEL 3, AND
3.0 FOR LEVEL 4
Q30 - CAREGIVING = 2.5;

*HERES THE CODE;

*CODE TO EXAMINE KCALS EXPENDED IN WALKING AND VIGOROUS EXERCISE,
SIMILAR TO PAFFENBARGER;

/*****

WSKCAL=P2WTK*WSKKWK;

EXKCAL=P2WTK*EXKKWK;

HAKCAL=SUM(OF WSKCAL EXKCAL);

*CATEGORICAL HARVARD ALUM VARIABLE;

IF HAKCAL=0 THEN HACAT=0;

IF 0 < HAKCAL < 500 THEN HACAT=1;

IF 500 LE HAKCAL < 1000 THEN HACAT=2;

IF 1000 LE HAKCAL < 1500 THEN HACAT=3;

IF 1500 LE HAKCAL < 2000 THEN HACAT=4;

IF HAKCAL GE 2000 THEN HACAT=5;

*****/

*CODE TO CALCULATE AMOUNT OF WALKING

IN MINUTES, AMOUNT OF BRISK WALKING IN MINUTES, AND ENGAGEMENT IN
INTENSE EXERCISE;

*CODE FOR MINUTES OF HIGH INTENSITY EXERCISE. INCLUDES AEROBICS
AND WEIGHT/CIRCUIT TRAINING AND ANY HIGH INTENSITY EXERCISE;

/*****

HIGHXMIN=SUM(OF FAACTIME FATRTIME FAH1TIME FAH2TIME FAH3TIME FAH4TIME);

IF FAACKKWK=0 AND FATRKKWK=0 AND FAH1KKWK=0 THEN HIGHXMIN=0;

IF HIGHXMIN=0 THEN HIGHX90=0;

IF 0<HIGHXMIN<90 THEN HIGHX90=0;

IF HIGHXMIN GE 90 THEN HIGHX90=1;

IF FAHIKKWK=0 THEN HIGHEX=0;

IF FAHIKKWK>0 THEN HIGHEX=1;

*****/

options ls=159 ps=51 formchar='|----|+|----+=|-\<>*' nodate nofmterr;

%MACRO SKIP ;

proc contents data=CALC.y9phact;

title4 'Final Dataset';

run;

%MEND SKIP ;

check for duplicates;

data dupes;

set CALC.y9phact;

by habcid;

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

run;

```
proc print;  
title4 'Duplicates in phact y9';  
run;
```

Appendix IV
Self-Reported Function Calculated Variables

Using Annual Telephone Interview Variables

Investigator Name: Eleanor Simonsick

Variable	General Description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
EASEQM	Ease walking 1/4 mile	Recode of R5DWQMEZ where 3 is easiest; imputations applied for missing R5DWQMEZ	If R4DWQMYN in (0,7,8) then EASEQM=7-R5DWQMEZ If R4DWQMYN=1 then EASEQM=4-R4DWQMDF	If R4DWQMYN=9 and (R4MNRS>0 or R4MNRS≠22) then R4DWQMYN=1; If R5DWQMEZ=8 and R4DWQMYN≠8 then R5DWQMEZ=2; If R4DWQMDF=8 and R4DWQMYN≠8 then R4DWQMDF=2; If R4DWQMYN in (0,7,8) and R5DWQMEZ<0 then EASEQM=4; If (R4DWQMYN=0 and R5DWQMEZ<0) then EASEQM=7-R5DW1MEZ; if (R4DWQMYN=0 and R5DWQMEZ<0) and R5DW1MEZ<0 and (R5DW1MYN=1 or R5DW1MYN=8 or R5DW1MYN≤Z) then EASEQM=4; if (R4DWQMYN=0 and R5DWQMEZ<0) and (R5DW1MYN=0 and R5DW1MEZ≤Z) then EASEQM=5 if R4DWQMYN=1 and R4DWQMDF<0 then EASEQM=7-R5DWQMEZ; if R4DWQMYN=1 and R4DWQMDF<0 and R5DWQMEZ<0 then EASEQM=2; if R4DWQMYN<0 and R4DWQMDF>0 then EASEQM=4-R4DWQMDF; if R4DWQMYN<0 and R5DWQMEZ>0 then EASEQM=7-R5DWQMEZ; if R4DWQMYN=8 and R5DWQMEZ=8 then EASEQM=4; if R4DWQMYN=9 and R4MNRS<0 and (R4MNRS4<0 or R4MNRS=22) then EASEQM=.	6=very easy 5=somewh at easy 4=not that easy 3=a little difficult 2=somewh at difficult 1=very difficult 0=unable to do

Variable	General Description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
EASE1M	Ease walking 1 mile	Recode of R4DW1MEZ where 3 is easiest; imputations applied for missing R4DW1MEZ	If R5DW1MYN=1 then EASE1M=0; if R5DW1MEZ=3 then EASE1M=1; if R5DW1MEZ=2 then EASE1M=2; if R5DW1MEZ=1 then EASE1M=3	If R5DW1MYN=8 and (R5DW1MEZ<0 or R5DW1MEZ=8) then EASE1M=1; if R5DW1MYN<=Z and R5DW1MEZ<=Z then EASE1M=EASEQM-4; if R5DW1MYN=0 and R5DW1MEZ<=Z then EASE1M=EASEQM-4; if R5DWQMYN=1 then EASE1M=0; if EASEQM > 0 and (EASEQM-3 < EASE1M) and EASEQM<=3 then EASE1M=0, else If EASEQM-3<EASE1M then EASE1M=EASEQM-3 If R5DW1MYN=8 and R5DW1MEZ in (1,2,3) then EASE1M=4-R5DW1MEZ; If R5DW1MYN=0 and R4DW1MEZ=8 then EASE1M=EASEQM-4;	3=very easy 2=somewhat easy 1=not that easy 0=difficult
WKAINDEX	Walking ability index	Summary measure of self-reported walking ability.	WKAINDEX=EASEQM + EASE1M	Imputed version of component variables used	This scale ranges from 0 (unable) to 9 (very easy)
EASE1F	Ease Climbing 1 flight	Recode of R7DW10EZ where 3 is easiest; imputations applied for missing R7DW10EZ	EASE1F=7-R7DW10EZ; IF R7DW10YN=1 THEN EASE1F=4-R7DIF;	IF R7DW10YN=9 then EASE1F=.; IF R7DW10EZ=8 THEN R7DW10EZ=2; IF R7DIF=8 THEN R7DIF=2; IF R7DW10YN=1 AND (R7DIF=8 OR R7DIF<0) THEN EASE1F=2; IF R7DW10YN=0 AND R7DW10EZ<0 AND R7DW10EZ<0 THEN EASE1F=4; IF EASE1F=. AND R7DW10YN NE 9 AND R7DW10YN>0 AND (R7DW20YN=1 OR R7DW20YN=8 OR R7DW20YN<=.z) THEN EASE1F=4; IF EASE1F=. AND R7DW20YN=0 AND R7DW20EZ<=.z THEN EASE1F=5;	6=very easy 5=somewhat easy 4=not that easy 3=a little difficult 2=somewhat difficult 1=very difficult 0=unable to do

Variable	General Description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
EASE2F	Ease Climbing 2 flights	Recode of R7DW20EZ where 3 is easiest; imputations applied for missing R7DW20EZ	If R7DW20YN=1 then EASE2F=0; if R7DW20EZ=3 then EASE2F=1; if R7DW20EZ=2 then EASE2F=2; if R7DW20EZ=1 then EASE2F=3	If R6DW10YN=9 then EASE2F=. If (R7DW20YN=8 or R7DW20YN<0) and (R7DW20EZ<0 or R7DW20EZ=8) and EASE1F>0 then EASE2F=1; if R7DW20YN≤.Z and R7DW20EZ≤.Z then EASE2F=EASE1F-4; if R7DW20YN=0 and (R7DW20EZ≤.Z or R7DW20EZ=8) and EASE1F>0 then EASE2F=EASE1F-4; if R6DW10YN=1 then EASE2F=0; if .Z<EASE1F≤3 then EASE2F=0, else If EASE1F-3 < EASE2F then EASE2F=EASE1F-3	3=very easy 2=somewh at easy 1=not that easy 0=difficult
CSAINDEX	Climbing stairs ability index	Summary measure of self-reported walking ability.	CSAINDEX=EASE1F + EASE2F	Imputed version of component variables used	This scale ranges from 0 (unable) to 9 (very easy)

Variable	General Description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
EASE10P	Ease lift/carry 10 lbs	Recode of RADIF10, RAEZ10LB, and RAD10AMT into a single variable, where 6 is easiest; imputations applied for missing	If RADIF10=0 or in (7,8) and RAEZ10LB ≠8 then EASE10P=7-RAEZ10LB; If RADIF10=1 then EASE10P=4-RAD10AMT	If RADIF10=0 and (RAEZ20LB=8 or RAEZ10LB<0) then EASE10P=5; otherwise If RADIF10 in (0,7,8) and RAEZ10LB=8 then EASE10P=4; If RADIF10=1 and (RAD10AMT=8 or RAD10AMT<0) then EASE10P=.; otherwise if RADIF10≤Z and RAD10AMT>0 and RAEZ10LB<0 then EASE10P=4-RAD10AMT; if RADIF10≤Z and RAD10AMT<0 and RAEZ10LB>0 then EASE10P=7-RAEZ10LB; if EASE10P=. and RAD20LBS ≠ 1 and RAD20LB≠8 then EASE10P=7-RAEZ20LB; if EASE10P=. and RAD20LBS=0 and RAEZ20LB=8 or ≤.Z then EASE10P=6; if EASE10P=. and RADIF10=0 and RAD20LBS=1 then EASE10P=4; if EASE10P=. and RADIF10=8 and RAD20LBS=1 then EASE10P=4; if EASE10P=. and RADIF10 in (0,8) and (RAD20LBS<0 or RAD20LBS=8) and (RAEZ20LB<0 or RAEZ20LB=8) then EASE10P=4; if EASE10P=. and RADIF10=0 and RAEZ10LB≤.Z and RAD10AMT>0 then EASE10P=4-RAD10AMT	6=very easy 5=somewh at easy 4=not that easy 3=a little difficult 2=somewh at difficult 1=very difficult 0=unable to do
EASE20P	Ease lift/carry 20 lbs	Recode of RAEZ20LB where 3 is easiest; imputations applied for missing RAEZ20LB	EASE20P=4-RAEZ20LB; if RAD20LBS=1 then EASE20P=0; if 0 le EASE10P le 3 then EASE20P=0	If RAD20LBS=8 and (RAEZ20LB=8 or RAEZ20LB<0) then EASE20p=1; If (RAD20LBS=0 or RAD20LBS≤.Z) and (RAEZ20LB≤.Z or RAEZ20LB=8) then EASE20P=EASE10P-4; if EASE10P-3 < EASE20P then EASE20P=EASE10P-3;	3=very easy 2=somewh at easy 1=not that easy 0=difficult

Variable	General Description	Detailed Description	How variable is calculated	How to handle missing or special values	Value labels
LCAINDEX	Lift/carry ability index	Summary measure of self-reported lifting/carrying ability.	LCAINDEX=EASE10P + EASE20P	Imputed version of component variables used	Ranges from 0 (unable) to 9 (very easy)
EASEUP	Ease rising from chair	Recode of R8DIFSTA, R8EZSTA, and R8DSTAMT into a single variable, where 6 is easiest; imputations applied for missing values	If R8DIFSTA in (0,7,8) then EASEUP=7 - R8EZSTA; if R8DIFSTA=1 then EASEUP=4 - R8DSTAMT	If R8EZSTA=8 and R8DIFSTA ne 8 then R8EZSTA=2; if R8DSTAMT=8 and R8DIFSTA ne 8 then R8DSTAMT=2; if R8DIFSTA=8 and R8EZSTA=8 then EASEUP=4; if R8DIFSTA≤.Z and R8DSTAMT>0 and R8EZSTA<0 then EASEUP=4 - R8DSTAMT; if R8DIFSTA=8 and R8EZSTA<0 and R8DSTAMT<0 then EASEUP=4; if R8DIFSTA=0 and R8EZSTA≤.Z then EASEUP=5; if R8DIFSTA=0 and R8EZSTA≤.Z and R8DSTAMT >0 then EASEUP=4 - R8DSTAMT	Ranges from 0 (least easy) to 6 (easiest)

```

*****
* Saved as \\Fu-hsing-c\HABC\HABC_SAS\Calculated Variables\Programs\Year
9\srfcn_y9.sas
*
* Lynn Harvey 05/03/07 - Modified year 8 program for use in year 9
* Lynn Harvey 10/14/05 - Modified year 6 program for use in year 8
* Lynn Harvey 11/17/06 - Modified to read from Fu-hsing-c instead of ilcavallo
***** ;

*****;
** HABC YEAR 9 SELF-REPORT PHYSICAL FUNCTION CALC VAR CODE **;
** **;
** Creates the following variables for 6mo visit: **;
** EASEQM: ease of walking 1/4 mile **;
** EASE1M: ease of walking 1 mile **;
** EASE1F: ease of climbing 1 flight **;
** EASE2F: ease of climbing 2 flights **;
** EASE10P: ease of lifting/carrying 10 lbs **;
** EASE20P: ease of lifting/carrying 20 lbs **;
** EASEUP: ease of standing up from chair w/o using arms **;
** CSAINDEX: climbing stairs ability index **;
** WKAINDEX: walking ability index **;
** LCAINDEX: lift/carry ability index **;
** **;
** Adapted from Fran Harris Y2 code: srfcn_y2.sas **;
*****;

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

%include '\\Fu-hsing-c\habc\habc_sas\programs\initV8.sas';
*libname calc '\\Fu-hsing-c\habc\habc_sas\calculated variables\datasets\year 9';

*options ls=141 ps=46 formchar='|----|+|---+=|-\<>*' nodate nofmterr;

data calc.y9srfcn (keep=HABCID EASE1F EASE2F EASEQM EASE1M EASE10P EASE20P
EASEUP CSAINDEX WKAINDEX
LCAINDEX); ***y9 data***;
merge habcrts.r2 (keep=habcid r2contac where=(r2contac=18))
habcrts.r4 (keep=habcid r4contac r4dwqmyn r4dwqmdf r4mnrs
where=(r4contac=18))
habcrts.r5 (keep=habcid r5contac r5dwqmez r5dwlmyn r5dwlmez
where=(r5contac=18))
habcrts.r6 (keep=habcid r6contac r6dw10yn r6dif
where=(r6contac=18))
habcrts.r7 (keep=habcid r7contac r7dw10ez r7dw20yn r7dw20ez
where=(r7contac=18))
habcrts.r8 (keep=habcid r8contac r8difsta r8ezsta r8dstamt
where=(r8contac=18))
habcrts.ra (keep=habcid racontac radif10 rad10amt raez10lb
rad20lbs raez20lb where=(racontac=18)) ;
by habcid ;

LABEL EASEQM = 'EASE WALKING 1/4 MILE, 6=VERY EASY'
EASE1M = 'EASE WALKING 1 MILE, 3=VERY EASY'
WKAINDEX = 'WALKING ABILITY INDEX, 9=BEST'
EASE1F = 'EASE CLIMBING 1 FLIGHT, 6=VERY EASY'
EASE2F = 'EASE CLIMBING 2 FLIGHTS, 3=VERY EASY'

```

```

        CSAINDEX      = 'CLIMBING STAIRS ABILITY INDEX, 9=BEST'
        EASE10P       = 'EASE LIFT/CARRY 10 LBS, 6=VERY EASY'
        EASE20P       = 'EASE LIFT/CARRY 20 LBS, 3=VERY EASY'
        LCAINDEX      = 'LIFT/CARRY ABILITY INDEX, 9=BEST'
        EASEUP        = 'EASE STANDING FROM CHAIR WITHOUT USING ARMS,
6=VERY EASY';

        *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 r5dwlmyrn WAS YES, DONT KNOW, OR MISSING THEN NOT THAT
EASY WAS ASSIGNED. IF NO
        EASE LEVEL FOR WALKING 1M WAS CODED AND r5dwlmyrn WAS NO AND r5dwlmez 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 r4dwqmyn=9 AND (r4mnrs>0 and r4mnrs ne 22) THEN r4dwqmyn=1;
        IF r5dwqmez=8 and r4dwqmyn ne 8 THEN r5dwqmez=2;
        IF r4dwqmdf=8 and r4dwqmyn ne 8 THEN r4dwqmdf=2;

        IF r4dwqmyn IN (0,7,8) AND r5dwqmez<0 THEN EASEQM=4;
        ELSE IF r4dwqmyn IN (0,7,8) THEN EASEQM=7-r5dwqmez;
        IF (r4dwqmyn=0 AND r5dwqmez<0) THEN EASEQM=7-r5dwlmez;
        IF (r4dwqmyn=0 AND r5dwqmez<0) & r5dwlmez<0 & (r5dwlmyrn in(1,8) OR
        r5dwlmyrn<=.z) THEN EASEQM=4;
        IF (r4dwqmyn=0 & r5dwqmez<0) & (r5dwlmyrn=0 & r5dwlmez<=.z) THEN EASEQM=5;
        IF r4dwqmyn=1 THEN EASEQM=4-r4dwqmdf;

        IF r4dwqmyn=1 AND r4dwqmdf<0 THEN EASEQM=7-r5dwqmez;
        IF r4dwqmyn=1 AND r4dwqmdf<0 AND r5dwqmez<0 THEN EASEQM=2;
        IF r4dwqmyn<0 AND r4dwqmdf>0 THEN EASEQM=4-r4dwqmdf;
        IF r4dwqmyn<0 AND r5dwqmez>0 THEN EASEQM=7-r5dwqmez;

        if r4dwqmyn=8 and r5dwqmez=8 then EASEQM=4;
        IF r4dwqmyn=9 AND (r4mnrs<0 or r4mnrs=22) THEN EASEQM=.;

%macro skip ;
options ls=141 ps=46 formchar='|----|+|----+=|-\<>*' nodate nofmterr;

proc sort data=y9srfcn ;
    by easeqm ;
run ;

proc freq data=y9srfcn ;
    by easeqm ;
    tables r4dwqmyn*r5dwqmez*r4dwqmdf*r5dwlmyrn*r5dwlmez*r4mnrs / list
        missing nocum ;
run ;
%mend skip ;

        *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;

```
IF r5dwlmyrn=1 THEN EASE1M=0;
ELSE IF r5dwlmez=3 THEN EASE1M=1;
ELSE IF r5dwlmez=2 THEN EASE1M=2;
ELSE IF r5dwlmez=1 THEN EASE1M=3;
```

*MISSING VALUE RECODES;

```
IF r5dwlmyrn=8 AND (r5dwlmez<0 OR r5dwlmez=8) THEN EASE1M=1;
IF r5dwlmyrn<=.z AND r5dwlmez<=.z AND EASEQM=4 THEN EASE1M=0;
IF r5dwlmyrn<=.z AND r5dwlmez<=.z AND EASEQM=5 THEN EASE1M=1;
IF r5dwlmyrn<=.z AND r5dwlmez<=.z AND EASEQM=6 THEN EASE1M=2;
IF r5dwlmyrn=0 AND r5dwlmez<=.z AND EASEQM=4 THEN EASE1M=0;
IF r5dwlmyrn=0 AND r5dwlmez<=.z AND EASEQM=5 THEN EASE1M=1;
IF r5dwlmyrn=0 AND r5dwlmez<=.z AND EASEQM=6 THEN EASE1M=2;
IF r4dwqmyn=1 THEN EASE1M=0;
```

*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 DOWNGRADES THE
EASE

LEVEL REPORTED FOR 1M TO THAT REPORTED FOR 1/4M;

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

```
if r5dwlmyrn=8 and r5dwlmez in (1,2,3) then EASE1M=4-r5dwlmez;
if r5dwlmyrn=0 and r5dwlmez=8 then EASE1M=EASEQM-4;
```

```
%macro skip ;
options ls=141 ps=46 formchar='|----|+|----+=|-/\<>*' nodate nofmterr;
```

```
proc sort data=y9srfcn ;
  by ease1m ;
run ;
```

```
proc freq data=y9srfcn ;
  tables ease1m*r4dwqmyn*r5dwlmyrn*r5dwlmez*easeqm / list
  missing nocum nofreq nopercnt ;
run ;
%mend skip ;
```

*CODE TO CREATE A SUMMARY MEASURE OF SELF-REPORTED WALKING ABILITY. THIS
SCALE WILL ULTIMATELY RANGE FROM 0 TO 9, BUT AT BASELINE IT RANGES FROM
4-9, AS NO ONE HAS DIFFICULTY. THIS SCALE CAN BE CREATED IN TWO WAYS: 1)
KEYING

OFF THE RESPONSE TO EASE1M AND ONLY LOOKING AT 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;

WKAINDEX=EASEQM + EASE1M;

```

%macro skip ;
proc freq data=y9srfcn ;
    tables WKAINDEX * EASEQM * EASE1M / list missing nocum ;
run ;
%mend skip ;

*CODE FOR EASE OF CLIMBING STAIRS;

*EASE1F;

IF r6dw10yn=9 THEN DO; EASE1F=.; EASE2F=.; END;
IF r7dw10ez=8 THEN r7dw10ez=2;
IF r6dif=8 THEN r6dif=2;
EASE1F=7-r7dw10ez;
IF r7dw10ez<0 THEN EASE1F=7-r7dw20ez;
IF r6dw10yn=1 THEN EASE1F=4-r6dif;
IF r6dw10yn=1 AND (r6dif=8 OR r6dif<0) THEN EASE1F=2;
IF r6dw10yn=0 AND r7dw10ez<0 AND r7dw20ez<0 THEN EASE1F=4;

IF EASE1F=. & r6dw10yn NE 9 & r6dw10yn>0 & (r7dw20yn=1 OR
    r7dw20yn=8 OR r7dw20yn<=.z ) THEN EASE1F=4;
IF EASE1F=. AND r7dw20yn=0 AND r7dw20ez<=.z THEN EASE1F=5;

%macro skip ;
options ls=141 ps=46 formchar='|----|+|----+=|-\<>*' nodate nofmterr;

proc sort data=y9srfcn ;
    by EASE1F ;
run ;

proc freq data=y9srfcn ;
    by EASE1F ;
    tables r6dw10yn*r7dw10ez*r7dw20yn*r7dw20ez*r6dif / list missing ;
run ;
%mend skip ;
*EASE2F;

IF r7dw20yn=1 THEN EASE2F=0;
ELSE IF r7dw20ez=3 THEN EASE2F=1;
ELSE IF r7dw20ez=2 THEN EASE2F=2;
ELSE IF r7dw20ez=1 THEN EASE2F=3;
IF (r7dw20yn=8 OR r7dw20yn<0)AND (r7dw20ez<0 OR r7dw20ez=8) AND EASE1F>0
THEN EASE2F=1;
IF r7dw20yn<=.z AND r7dw20ez<=.z AND EASE1F=4 THEN EASE2F=0;
IF r7dw20yn<=.z AND r7dw20ez<=.z AND EASE1F=5 THEN EASE2F=1;
IF r7dw20yn<=.z AND r7dw20ez<=.z AND EASE1F=6 THEN EASE2F=2;
IF r7dw20yn=0 AND (r7dw20ez<=.z OR r7dw20ez=8) AND EASE1F>0 THEN
EASE2F=EASE1F-4;
IF r6dw10yn=1 THEN EASE2F=0;

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

%macro skip ;
options ls=141 ps=46 formchar='|----|+|----+=|-\<>*' nodate nofmterr;

```

```

proc sort data=y9srfcn ;
  by EASE2F ;
run ;

proc freq data=y9srfcn ;
  tables EASE2F*r7dw20yn*r7dw20ez*EASE1F / list missing ;
run ;
%mend skip ;

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

      CSAINDEX=EASE1F + EASE2F;

/*
proc freq data=y9srfcn ;
  tables CSAINDEX * EASE1F * EASE2F / list missing nocum ;
run ;
*/

      *CODE FOR LIFTING/CARRYING;

      IF radif10=0 AND (raez10lb=8 OR raez10lb<0) THEN EASE10P=5;
      ELSE IF radif10 IN (0,7,8) and raez10lb ne 8 THEN EASE10P=7-raez10lb;
      else if radif10 in (0,7,8) and raez10lb=8 then EASE10P=4;
      IF radif10=1 AND (rad10amt=8 OR rad10amt<0) THEN EASE10P=.;
      ELSE IF radif10=1 THEN EASE10P=4 - rad10amt;
      IF radif10<=.z & rad10amt>0 & raez10lb<0 THEN EASE10P=4 - rad10amt;
      IF radif10<=.z & rad10amt<0 & raez10lb>0 THEN EASE10P=7 - raez10lb;
      IF EASE10P=. & rad20lbs NE 1 & rad20lbs NE 8 THEN EASE10P=7-raez20lb;
      IF EASE10P=. & rad20lbs=0 & (raez20lb LT 0 OR rad20lbs=8) THEN EASE10P=6;
      IF EASE10P=. AND radif10=0 AND rad20lbs=1 THEN EASE10P=4;
      IF EASE10P=. AND radif10=8 AND rad20lbs=1 THEN EASE10P=4;
      IF EASE10P=. & radif10 IN (0,8) & (rad20lbs<0 OR rad20lbs=8) &
      (raez20lb<0 OR raez20lb=8) THEN EASE10P=4;
      IF EASE10P=. AND radif10=0 AND raez10lb<=.z AND rad10amt>0 THEN
      EASE10P=4 - rad10amt;

%macro skip ;
options ls=141 ps=46 formchar='|----|+|----+=|-\<>*' nodate nofmterr;

proc sort data=y9srfcn ;
  by EASE10P ;
run ;

proc freq data=y9srfcn ;
  by EASE10P ;
  tables radif10*raez10lb*rad10amt*rad20lbs*raez20lb / list missing ;
run ;
%mend skip ;

      *IF raez20lb=8 and rad20lbs ne 8 THEN raez20lb=2;
      EASE20P=4-raez20lb;
      if rad20lbs=8 and (raez20lb=8 or raez20lb<0) then EASE20P=1;
      if rad20lbs=0 and raez20lb=8 then EASE20P=EASE10P-4;

```

```

IF rad20lbs=1 THEN EASE20P=0;
IF (0 LE EASE10P LE 3) THEN EASE20P=0;

IF (rad20lbs<=.z OR rad20lbs=0) and (raez20lb<=.z OR raez20lb=8)
    AND EASE10P=4 THEN EASE20P=0;
IF (rad20lbs=0 OR rad20lbs<=.z) AND (raez20lb<=.z OR raez20lb=8)
    AND EASE10P=6 THEN EASE20P=2;
IF (rad20lbs=0 OR rad20lbs<=.z) AND (raez20lb<=.z OR raez20lb=8)
    AND EASE10P=5 THEN EASE20P=1;

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

%macro skip ;
options ls=141 ps=46 formchar='|----|+|----+=|-\<>*' nodate nofmterr;

proc sort data=y9srfcn ;
    by EASE20P ;
run ;

proc freq data=y9srfcn ;
    tables EASE20P*raez20lb*rad20lbs*EASE10P / list missing ;
run ;
%mend skip ;

*CODE FOR SUMMARY INDEX;

LCAINDEX=EASE10P + EASE20P;

%macro skip ;
proc freq data=y9srfcn ;
    tables LCAINDEX * EASE10P * EASE20P / list missing nocum ;
run ;
%mend skip ;

*OTHER FUNCTION MEASURES;

IF r8ezsta=8 and r8difsta ne 8 THEN r8ezsta=2;
IF r8dstamt=8 and r8difsta ne 8 THEN r8dstamt=2;
IF r8difsta IN (0,7,8) THEN EASEUP=7 - r8ezsta;
IF r8difsta=1 THEN EASEUP=4 - r8dstamt;
if r8difsta=8 and r8ezsta=8 then EASEUP=4;
IF r8difsta<=.z AND r8dstamt>0 AND r8ezsta<0 THEN EASEUP=4 - r8dstamt;
IF r8difsta=8 AND r8ezsta<0 AND r8dstamt<0 THEN EASEUP=4;
IF r8difsta=0 AND r8ezsta<=.z THEN EASEUP=5;
IF r8difsta=0 & r8ezsta<=.z & r8dstamt >0 THEN EASEUP=4 - r8dstamt;
run ;
options ls=141 ps=46 formchar='|----|+|----+=|-\<>*' nodate nofmterr;

%macro skip ;
proc sort data=y9srfcn ;
    by EASEUP ;
run ;

```

```

proc freq data=y9srfcn ;
    tables EASEUP*r8difsta*r8ezsta*r8dstamt / list missing ;
run ;
%mend skip ;

%macro skip ;
proc freq data=y9srfcn ;
    tables EASE1F EASE2F EASEQM EASE1M EASE10P EASE20P EASEUP CSAINDEX
WKAINDEX
    LCAINDEX ;
*   tables easelm * r5dwlmyrn * r4dwqmyn * r5dwlmez / list missing nocum ;
*   tables r4dwqmyn r4dwqmdf r4mnrs ;
*   tables r4dwqmyn * (r4dwqmdf r4mnrs) / list missing ;
*   tables (r5dwqmez r4dwqmdf) * r4dwqmyn / list missing ;
*   tables r4dwqmyn * r5dwqmez * r5dwlmez / list missing ;
*   tables r4dwqmyn * r4dwqmdf * r5dwqmez / list missing nocum ;
*   tables easeqm ;
*   tables EASEQM*r4dwqmyn*r5dwqmez*r4dwqmdf*r5dwlmyrn*r5dwlmez*r4mnrs / list
missing nocum ;
*   tables r5dwlmyrn*r5dwlmez*EASEQM*r4dwqmyn / list missing nocum ;
*   tables EASE1M*r5dwlmyrn*r5dwlmez*EASEQM*r4dwqmyn / list
missing nocum ;
*   tables WKAINDEX*EASEQM*EASE1M / list missing ;
*   tables r6dw10yn*r7dw10ez*r7dw20yn*r7dw20ez*r6dif / list missing ;
*   tables EASE1F*r6dw10yn*r7dw10ez*r7dw20yn*r7dw20ez*r6dif / list missing ;
*   tables EASE2F*r7dw20yn*r7dw20ez*EASE1F / list missing ;
*   tables CSAINDEX*EASE1F*EASE2F / list missing ;
*   tables TIRED1F*FADW10WX ;
*   tables EASE10P*radif10*raez10lb*rad10amt*rad20lbs*raez20lb / list missing
;
*   tables EASE20P*raez20lb*rad20lbs*EASE10P / list missing ;
*   tables LCAINDEX*EASE10P*EASE20P / list missing ;
*   tables EASEUP*r8difsta*r8ezsta*r8dstamt / list missing ;
*   tables EASEHHW*r6difHW*FAEZHW*FADHWAMT / list missing ;
*   tables r5dwqmez r5dwlmyrn r5dwlmez ;
*   tables r6dw10yn r6dif ;
*   tables r7dw10ez r7dw20yn r7dw20ez ;
*   tables r8difsta ;
*   tables radif10 rad10amt raez10lb rad20lbs raez20lb ;
*   format EASEQM r4dwqmyn r5dwqmez r4dwqmdf r5dwlmyrn r5dwlmez r4mnrs
EASE1M WKAINDEX /*FADWQMT2*/
EASE1F r6dw10yn r7dw10ez r7dw20yn r7dw20ez r6dif
EASE2F CSAINDEX /*TIRED1F FADW10WX */
EASE10P radif10 raez10lb rad10amt rad20lbs raez20lb
EASE20P LCAINDEX
/*EASEHHW r6difHW FAEZHW FADHWAMT*/;
    FORMAT EASE1F EASE2F EASEQM EASE1M EASE10P EASE20P EASEUP CSAINDEX
WKAINDEX
    LCAINDEX SPMISS. /*TIRED1F TIREDQM YNDK.*;/;
run ;

proc contents data=calc.y9srfcn ;
    title 'Year 9 - Self Reported Physical Functioning' ;
run ;
%mend skip ;

***check for duplicates***;

```

```
data dupes;
  set calc.y9srfcn;
  by habcid;
  if not(first.habcid and last.habcid);
run;

proc print;
title4 'Duplicates in SRFCN y9';
run;
```