Data Dictionary And Documentation

October 25, 2018
Version: 7.0

LLFS Data Management Coordinating Center
Department of Genetics – Division of Statistical Genomics
Center for Genome Sciences and Systems Biology
Washington University School of Medicine
660 South Euclid Ave, Campus Box 8506-98-601
St. Louis, MO 63110
Phone: (314) 362-3945 Fax: (314) 362-4227
e-mail: l.kniepkamp@wustl.edu

Table of Contents

Introduction - 1 -
Derived Variables - 1 -
Common Variables - 2 -

PHASE I - 2 -

Pre-In Person Visit - 2 -
Family Longevity Collection Score Instrument – Proband (FLoSS) (PTSI) Dataset - 2 -
Proband Relative Contact Information (Relatives) Dataset - 2 -
Sibling Information (SIBINFO) Dataset - 3 -
In Person Visit - 3 -
Blood (Blood) Dataset - 3 -
_FTESTSTRN - Free Testosterone (ng/dL) - 3 -
_ALBBT - Albumin Bound Testosterone (ng/dL) - 3 -
_BIOTESTSTRN - Bioavailable Testosterone (ng/dL) - 3 -
_GLUR_NEW that replaces GLUR (glucose) with some updates. - 3 -
Blood Pressure, Heart Rate, Height, Weight, Waist (BPHR) Dataset - 4 -
_HEIGHT - 4 -
_BMI - 5 -
_WAIST - 5 -
_SIT - 5 -
_HTIN25 & _HTCM25 - 5 -
_KNEE - 5 -
_SBP - 6 -
_DBP - 6 -
_PULSE - 6 -
Carotid Intima-Media Thickness Test Worksheet (CAROTID) Dataset - 6 -
Carotid Intima-Media Thickness Test Function (CAROTIDFUNC) Dataset - 6 -
Mood and Personality Assessment (CES-D and NEO 5-Factor) Dataset - 6 -
NEUROTICISM, SIZE_NEUROTOCOSM, CONSCIENTIOUSNESS and SIZE_CONSCIENTIOUSNESS - 7 -
Coded Medications (CODEDMEDS) Dataset - 8 -
Coded Medications (CODEDMEDS_ATC) Dataset - 8 -
Cognitive Assessments (COGASSESS) Dataset - 8 -
Consent Tracking and Interview Feasibility (CTIF) Dataset - 8 -
Digital Clock Drawing Test (dCDT)Dataset - 8 -
Carotid Intima-Media Thickness Test Plaque Assessment (FINALCQI) Dataset - 9 -
Lung Function (LUNGFUNC) Dataset - 9 -
Medication Inventory (MEDCHK) Dataset - 9 -
Medication Inventory (MEDS) Dataset - 9 -
Medical History (MEDHX) Dataset - 9 -
NEO Five-Factor Inventory (NEO) Dataset - 9 -

OPENNESS, CONSCIENTIOUSNESS, EXTRAVERSION, AGREEABLENESS NEUROTICISM, SIZE_OPENNESS, SIZE_CONSCIENTIOUSNESS, SIZE_EXTRAVERSION, SIZE_AGREEABLENESS and SIZE_NEUROTOCOSM ... - 10 -
Personal History (PERSHX) Dataset ... - 13 -
_SMOKENOW, _PIPENOW, and _PACKYRS ... - 13 -
_SUMPACKYRS ... - 13 -
_SMOKE_CIG, _SMOKE_PIPE, and _SMOKE_CAT ... - 14 -
Physical Function and Activity (PHYSICAL) Dataset ... - 15 -
Performance Measures (PM) Dataset ... - 15 -
_TOTSCORE ... - 15 -
Prevalence of Disease (PREVDISEASE) Dataset ... - 16 -
_HTDIS, _STRK, _LUNGDIS, _HTN, _DIABETES, _PAD, _CANCER: ... - 16 -
_HTN_ATC, and _DIABETES_ATC ... - 18 -
_ADAT2D and _ADAT2D_AGE_REPORTED_DETECTED ... - 19 -
Socio-Demographic Information (SDI) Dataset ... - 22 -
_AGE ... - 22 -
_AGE_REVISED ... - 22 -
Spirometry Safety Questionnaire (SPIRO) Dataset ... - 22 -
Spirometry Safety Questionnaire (SPIROMEDS) Dataset ... - 22 -
Survival Indices (SURVL_INDICES) Dataset ... - 22 -
Healthy Aging Index ... - 22 -
HAI ... - 23 -
HAI_rg ... - 23 -
HAI_m ... - 23 -
HAI_m_rg ... - 24 -
HAI_rl ... - 24 -
HAI_m_rl ... - 24 -
Telephone Interview for Cognitive Status (TICS) Dataset ... - 28 -
Venipuncture (VENIP) Dataset ... - 28 -
_FASTTIME, _FAST, FASTING_LIPID and FASTING_CBC ... - 28 -
What Data Collected per Participant (WHATDATA) Dataset ... - 29 -

PHASE II ... - 29 -

Follow Up (FOLLOWUP) Dataset ... - 29 -

GENOTYPES ... - 30 -

Annotation (Info and Map) Datasets ... - 30 -
Info Datasets Variables ... - 32 -
Anonymous Genotypes (GANON) Datasets ... - 33 -
Gene Frequency (GENEFREQ) Datasets ... - 34 -
GTRIPLET and TRIPLET_visit2 Dataset ... - 34 -

Introduction

This document describes the analysis datasets and derived variables for the Long Life Family Study (Visit 1 and Visit 2). In general, each form/procedure in either Pre-Clinic (PTSI, Relatives), Clinic or Follow-Up is stored as its own SAS dataset (e.g. blood, meds, physical, venip, etc.), with multiple records (observation) per person corresponding to what visit the data was collected during (see below). The few exceptions to this rule are noted in the appropriate places. For the most part, each dataset retains the original, raw form variables as collected on each subject. These are mnemonically named (e.g. SEX instead of Q7), with accompanying SAS labels. A user-defined format library is also included to provide value-labels to codes. Thus, PROC CONTENTS, along with PROC FORMAT with the FMTLIB option can be used in conjunction with the official book of forms and QxQs (also supplied by the Coordinating Center) to provide documentation for the raw form data itself. We concentrate here instead on documenting the dataset organization, and derived analysis variables created at the Coordinating Center.

The data sets for each Panel/Form may contain multiple records/rows per subject with each row corresponding to a different exam/follow-up (depending on how many times the Form was administered on the subject). The variables Visitcode and/or Contactyr in a data set help us to track down which event the data was collected from. The label and values for Visitcode and Contactyr are:

Visitcode (Visit code):
1 – Visit 1 in-person
2 – Visit 1 Follow-up
3 – Visit 2 in-person (returning participants)
4 - Visit 2 in-person (new participants)
5 – Visit 2 follow-up

Contactyr (follow-up contact year):
0 - Visit 1 in-person
1, 2, 3, 4 …, or 9 – follow-up year 1, 2, 3, 4 …, or 9
99 – Visit 2 in-person

Derived Variables

For the most part, all derived variables are named beginning with an underscore, to readily distinguish them from the raw form variables. The derived variables are stored in the "natural" form/procedure dataset from which they were created, e.g. _BMI is in BPHR dataset, total cholesterol is in the BLOOD dataset, etc. In the various sections below, we summarize each dataset, followed by a description of each derived variable in it. In many cases, we give not only the algorithm used in defining the variable, but also the actual SAS code which implements it, for completeness, and to assure that the final coding is as intended. Each of the derived variables is described first, then the SAS code that created all of them is presented.

Common Variables

Most datasets have a few, common identifying variables.

ID

ID is the original key identifying variable, at the time of data collection. ID is usually in each dataset. It is the 8 character LLFS ID. The structure of ID is:

CNNNNNNN
C is the single digit field center number:
1 = Denmark, DK 2 = Boston, MA
3 = Pittsburgh, PA 4 = New York, NY

TOUCHDAT

TOUCHDAT is a SAS datetime variable automatically updated by the Data Entry System at Field Centers to give the date/time when the record was last changed.

DATE

DATE is a SAS date variable giving the date the form/procedure was collected.

FC

FC is the field center.
1 = Denmark, DK 2 = Boston, MA
3 = Pittsburgh, PA 4 = New York, NY

SUBJECT

SUBJECT is a de-identified, unique identifier for each Participant. It is a 5 digit number. Datasets with one obs/subject are uniquely identified by ID and can be merged/linked using this variable.

PHASE I

Pre-In Person Visit

Family Longevity Collection Score Instrument – Proband (FLoSS) (PTSI) Dataset

This dataset contains the answers to eligibility questions, including those that were used to calculate the Family Longevity Selection Score (FLoSS), and some socio-demographic data.
This dataset has no derived variables.

Proband Relative Contact Information (Relatives) Dataset

This questionnaire collected information on the Proband's relatives, such as contact information, and permission to contact them.
This dataset has no derived variables.

Sibling Information (SIBINFO) Dataset

This dataset contains information on the Proband's siblings.
This dataset has no derived variables.

In Person Visit

Blood (Blood) Dataset

This dataset contains the data regarding the blood chemistries, the date the blood was drawn, and the LLFS Subject. Special assays and candidate genes were added as well. This dataset has four derived variables:

_FTESTSTRN - Free Testosterone (ng/dL)

Free T (mol/L)= $\frac{-b+\sqrt{b^2+4a[TT]}}{2a}$,
where a = ka + ks + (ka × ks) ([SHBG (mol/L)] + [Alb (mol/L)] − [TT (mol/L)]),
b = 1 + ks [SHBG] + ka [Alb] − (ka + ks) [TT],
ka = 3.6×10^4 L/mol,
ks = 1×10^9 L/mol.
Total testosterone (TT) (mol/L) = (Reported TT (ng/dl)/288.4) x 10^{-8}
Molecular weight of Testosterone = 288.4
SHBG (mol/L) = Reported SHBG (nmol/L) x 10^{-9}
Alb (mol/L) = (Reported Alb (g/dl) x 10)/69000
Molecular weight of Albumin = 69000

_ALBBT - Albumin Bound Testosterone (ng/dL)

Albumin Bound T (ng/dl) = [ka x ($C_{onc.}$Alb(mol/L)]*Free T
where ka = 3.6 x 10^4 L/mol

_BIOTESTSTRN - Bioavailable Testosterone (ng/dL)

Bioavailable Testosterone = Free Testosterone + Albumin bound Testosterone

_GLUR_NEW that replaces GLUR (glucose) with some updates.

For the subjects whose blood samples were processed after two hours of collection, their lab measured glucose levels were likely underestimated because of glycolysis (sugar breaking down) at room temperature. However, in most of nondiabetic subjects among them, glucose levels may be best estimated through their HbA1c levels (estimated glucose or eAG levels = 28.7*HbA1c - 46.7, Nathan DM et al. Diabetes Care 2008;31:1-6 followed by some necessary

adjustments). In specific, glucose levels = alpha * eAG in nondiabetic subjects whose blood samples were processed within two hours of collection. And thereafter, for each nondiabetic subject whose sample was processed at least two hours after collection of blood, alpha was applied to obtain his/her best estimated glucose value (glucose = alpha * eAG). The estimation of alpha is performed for visit 1 and visit 2 separately. Please note that the visit 2 data subject to change based on final data. The SAS code that creates _GLUR_NEW is as follows:

```
if drawtime>0 and t26time >0 then processtime = INTCK('minute',drawtime,t26time)/60;
   label drawtime = 'Time Venipuncture ended'
        t26time = 'Time SST1 & SST2 tubes 2&6 were spun';
if processtime>=0 and processtime<=2 then gt2hrs=0;
else if processtime>2 then gt2hrs=1;
else if processtime<0 then gt2hrs=-1;

eAG = glyhb * 28.7 - 46.7;
    label eAG='Estimated Average Glucose (mg/dL)';
if (_fast=1 and glur>=126) or glyhb >=6.5 or diabnow=1 or diab=1 then _diabetes=1;
else _diabetes=0;

alpha = glur_mean / eag_mean;
** glur_mean is mean of glur and eag_mean is mean of eAG.  They are
estimated using fasting non-T2D samples that were processed within two hours of
collection.  The estimations are performed for Visit 1 and Visit 2 separately;
if _diabetes=1 then glur_new = .;
if gt2hrs=1 and _diabetes=0 then glur_new = &alpha0 * eag;
if (gt2hrs=0 or gt2hrs=-1) and _diabetes=0 then glur_new = glur;
```

Blood Pressure, Heart Rate, Height, Weight, Waist (BPHR) Dataset

This dataset contains the data regarding the blood pressure, height, knee height and waist circumference.

_HEIGHT

_HEIGHT is the Average Standing Height in cm calculated from stand1, stand2, stand3, and stand4.

```
if stand1>0 and stand2>0 and (not(stand3>0) or not(stand4>0))
      then _height=mean(stand1,stand2);

    else if stand3>0 and stand4>0
      then _height=mean(stand3,stand4);
```

_BMI

_BMI is Body Mass Index calculated from weight and height on BPHR.

```
if (weight >0 and _height>0) then
     _BMI=(weight/((_height/100)**2));
```

_WAIST

_WAIST is the Average Abdominal Circumference in cm calculated from waist1, waist2, waist3, and waist4.

```
if waist1>0 and waist2>0 and (not(waist3>0) or not(waist4>0))
      then _waist=mean(waist1,waist2);
    else if waist3>0 and waist4>0
      then _waist=mean(waist3,waist4);
```

_SIT

_SIT is the Average Height while sitting in cm calculated from sit1, sit2, sit3, and sit4.

```
if sit1>0 and sit2>0 and (not(sit3>0) or not(sit4>0))
      then _sit=mean(sit1,sit2);
else if sit3>0 and sit4>0
      then _sit=mean(sit3,sit4);
```

_HTIN25 & _HTCM25

_HTIN25 is the Height when 25 years old in total inches.
_HTCM25 is the Height when 25 years old in cm.

```
if htft25>0 and htin25>=0 then  _htin25= (htft25 * 12) + htin25;
if htft25>0 and htin25<0 then _htin25= (htft25 * 12);
if htcm25>0 then _htin25= htcm25 / 2.54;
_htcm25 = _htin25 * 2.54;
```

_KNEE

_KNEE is the Average Length of Leg from Heel to Knee in cm.

```
_knee=mean(knee1,knee2);
```

_SBP

_SBP is the Average Sitting Systolic Blood Pressure in mmHg.

```
_sbp=mean(sys1,sys2,sys3);
```

_DBP

_DBP is the Average Sitting Diastolic Blood Pressure in mmHg.

```
_dbp=mean(dia1,dia2,dia3);
```

_PULSE

_PULSE is the Average Sitting Pulse.

```
_pulse=mean(pulse1,pulse2,pulse3);
```

Carotid Intima-Media Thickness Test Worksheet (CAROTID) Dataset

This dataset contains information from URL Carotid IMT Worksheet entered into Redcap by the sites. Its main use has been by the URL lab to check alerts, plaque, and tracking. Carotid plaque variables from this dataset are **NOT** to be used for analyses. Please use the plaque data from the **finalcqi** dataset
This dataset has no derived variables.

Carotid Intima-Media Thickness Test Function (CAROTIDFUNC) Dataset

This dataset contains the carotid artery analysis values that were read at the Ultrasound Reading Lab at Pittsburgh. These variables should be used for carotid function analyses. Carotid assessment, primary variables for analysis, data cleaning, data collection and quality control are in the document "LLFS Carotid Data DictionaryFINAL_ebm.pdf".
This dataset has no derived variables.

Mood and Personality Assessment (CES-D and NEO 5-Factor) Dataset

This dataset contains the data regarding depressive symptomatology, personality dimensions of neuroticism and conscientiousness. The 24 questions NEO is only administered at in person visit 1. Derived variables NEUROTICISM, and CONSCIENTIOUSNESS are sums of different questions on the form. The SIZE_NEUROTOCOSM and SIZE_CONSCIENTIOUSNESS are the number of non-missing responses for NEUROTOCOSM and CONSCIENTIOUSNESS, respectively, which are used to calculate the scores.

NEUROTICISM, SIZE_NEUROTOCOSM, CONSCIENTIOUSNESS and SIZE_CONSCIENTIOUSNESS

```
#R scripts (from Boston field site)
cesdneo.orig.1to5.scale <- read.csv(paste(root.dir,"cesdneoall.csv",sep=""),
header=T, na.strings=c("", "D","R", "N", "U", NA) )
```

```
index.nonreverse <- c("anxious", "sad", "worrier", "blue", "wastetime", "reliable",
"organized", "methodical")
nonreverse <- cesdneo.orig.1to5.scale[, index.nonreverse]
new.nonreverse <- 5 - nonreverse
index.need.reverse <- c("acmplshgoals", "commitment", "neat", "pacing", "productive",
"conscientious", "excellence", "cleargoals", "tense", “angry”,
"worthless", "discouraged", "inferior", "stress", "helpless", "ashamed")
need.reverse <- cesdneo.orig.1to5.scale[, index.need.reverse]
reversed <- need.reverse - 1

other.index <- c("subject", "fc", "version", "date", "form_completed",
"date_completed")
other <- cesdneo.orig.1to5.scale[, other.index]

length(index.nonreverse)+ length(index.need.reverse) # 24

# ========== form new data ============

cesdneo.rescaled <- data.frame(other, reversed, new.nonreverse)

cesdneo.Neuroticism.data <- data.frame(cesdneo.rescaled$tense,
cesdneo.rescaled$worthless, cesdneo.rescaled$anxious, cesdneo.rescaled$angry,
cesdneo.rescaled$discouraged, cesdneo.rescaled$sad, cesdneo.rescaled$worrier,
cesdneo.rescaled$inferior, cesdneo.rescaled$stress, cesdneo.rescaled$helpless,
cesdneo.rescaled$ashamed, cesdneo.rescaled$blue)

   names(cesdneo.Neuroticism.data) <- c("tense", "worthless", "anxious", "angry",
"discouraged", "sad", "worrier", "inferior", "stress", "helpless", "ashamed", "blue")
   length(names(cesdneo.Neuroticism.data))

cesdneo.Conscientiousness.data <- data.frame(cbind(cesdneo.rescaled$wastetime,
cesdneo.rescaled$acmplshgoals, cesdneo.rescaled$commitment,
cesdneo.rescaled$reliable, cesdneo.rescaled$neat, cesdneo.rescaled$pacing,
cesdneo.rescaled$productive, cesdneo.rescaled$organized, cesdneo.rescaled$methodical,
cesdneo.rescaled$conscientious, cesdneo.rescaled$excellence,
cesdneo.rescaled$cleargoals))

   names(cesdneo.Conscientiousness.data) <-
c("wastetime","acmplshgoals","commitment","reliable","neat","pacing", "productive",
"organized", "methodical","conscientious","excellence","cleargoals")

      Neuroticism <- apply(cesdneo.Neuroticism.data, 1, sum, na.rm=T)
Conscientiousness <- apply(cesdneo.Conscientiousness.data, 1, sum, na.rm=T)

      size.Neuroticism <- apply( is.na(cesdneo.Neuroticism.data)==F,1,sum)
size.Conscientiousness <- apply( is.na(cesdneo.Conscientiousness.data)==F,1,sum)

cesdneo.rescaled <- data.frame(cesdneo.rescaled, Neuroticism, size.Neuroticism,
Conscientiousness, size.Conscientiousness)

cesdneo.rescaled <- data.frame(cesdneo.rescaled, Neuroticism, size.Neuroticism,
Conscientiousness, size.Conscientiousness)
```

Coded Medications (CODEDMEDS) Dataset

This dataset contains the four yes/no variables Anne Newman and her staff at the University of Pittsburgh developed from the Medications data. Specifically, they are HTNRX, LIPIDRX, NITRORX, DMRX. These represent whether or not the Participant is currently taking a medication for Hypertension, Lipid Lowering, Angina, or Diabetes Mellitus.
This dataset has no derived variables.
This is for visit 1 data only.

Coded Medications (CODEDMEDS_ATC) Dataset

This dataset contains the four yes/no variables Paola Sebastiani and her staff at the Boston University developed from the Visit 1 Medications data, using the World Health Organization's Anatomical Therapeutic Chemical (ATC) classifications. DMCC recreate 4 of them for Visit 2 Medication data based on the same coding scheme used for Visit 1. Specifically, the four variables are HTN, LIPID, NITRO, DIAB. These represent whether or not the Participant at Visit 1 or Visit 2 is taking a medication for Hypertension, Lipid Lowering, Angina, or Diabetes Mellitus.
This dataset has no derived variables.
This is for Visit 1 data and Visit 2 data.

Cognitive Assessments (COGASSESS) Dataset

There are three forms that compose this battery, the NACC UDS, the Telephone Interview for Cognitive Status (TICS, see below), and the Informant-Based Date of Onset Interview. This dataset has no derived variables.

Consent Tracking and Interview Feasibility (CTIF) Dataset

This dataset contains the answers to the consent questions. There were different questions at the different Field Centers, based upon what their individual Internal Review Boards required. This dataset has no derived variables.

Digital Clock Drawing Test (dCDT)Dataset

This dataset contains the variables obtained from the use of the digital pen in the clock drawing test as part of the cognitive assessment. Scored clocks were sent to the reading lab at MIT and the final data was sent to the DMCC from MIT for release. The long label for each variable is in a separate file called "eclock_variables_01132016.xlsx".
This dataset has no derived variables

Carotid Intima-Media Thickness Test Plaque Assessment (FINALCQI) Dataset

This dataset contains data from the Carotid Duplex Scan Feedback Form that is completed by reading center sonographers after assessing images and video clips for plaque presence and burden as well as image quality scoring. This dataset contains the carotid plaque data to be used for plaque assessment. Details for variables, data collection, and quality control are in the document "LLFS Carotid Data DictionaryFINAL_ebm.pdf".

Lung Function (LUNGFUNC) Dataset

This dataset contains the results of the Spirometry test, sent to the Data Coordinating Center from the Reading Center. This dataset has no derived variables.

Medication Inventory (MEDCHK) Dataset

This dataset contains the response of the first question of the Medication Inventory; if any medication was taken in the past 2 weeks. This dataset has no derived variables.

Medication Inventory (MEDS) Dataset

This dataset contains the responses of the rest of the questions from the Medication Inventory; the medication name, strength, units, formulation code, whether or not the container was seen, and other notes. This data set has 1 record per medication; therefore, multiple records per Participant. It includes the person's ID to link this dataset to the others. This dataset has no derived variables.

Medical History (MEDHX) Dataset

This dataset contains information about the Medical History of the Participants, including all diseases the person has/had. This dataset has no derived variables.

NEO Five-Factor Inventory (NEO) Dataset

This dataset contains the data regarding a personality inventory that examines a person's Big Five personality traits (openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism). This is the full version of the NEO as compared to the short version of the cesdneo data set. This is administered at visit 1 follow up, visit 2 new participants, and anyone who has missing visit 1 NEO data (visitcode is 2, 3, or 4). Derived variables OPENNESS, CONSCIENTIOUSNESS, EXTRAVERSION, AGREEABLENESS and NEUROTICISM are sums of different questions on the form. The SIZE_OPENNESS, SIZE_CONSCIENTIOUSNESS, SIZE_EXTRAVERSION, SIZE_AGREEABLENESS and SIZE_NEUROTICISM are the number of non-missing responses for OPENNESS, CONSCIENTIOUSNESS, EXTRAVERSION, AGREEABLENESS and NEUROTICISM, respectively, which are used to calculate the scores.

OPENNESS, **CONSCIENTIOUSNESS, EXTRAVERSION, AGREEABLENESS NEUROTICISM, SIZE_OPENNESS, SIZE_CONSCIENTIOUSNESS, SIZE_EXTRAVERSION, SIZE_AGREEABLENESS and SIZE_NEUROTOCOSM**

```
*Run this sas code to generate formatted data and the output file
neoall_formatted.csv is used in the R script below;
proc format;
VALUE LIKERT
         1='Strongly Disagree'
```

```
         2='Disagree'
         3='Neutral'
         4='Agree'
         5='Strongly Agree';
run;

data neo;
 set master.neoall;
 format worrier -- excellence LIKERT.;;
run;

proc export data=neo outfile="neoall_formatted.csv" dbms=csv replace; run;
```

```
#R scripts from Boston field site

neo.orig <- read.csv(("neoall_formatted.csv"), header=T, na.strings=c("", "D","R",
"N", "U", NA))
#------- Re-level Scales: -----------
#Strongly Agree -> 0, Agree -> 1, Neutral -> 2, Disagree -> 3, Strongly Disagree -> 4

#levels(neo.orig$worrier)
#"Agree"  "Disagree"  "Neutral"  "Strongly Agree"  "Strongly Disagree"
   levels(neo.orig$worrier)[1] <- 1
   levels(neo.orig$worrier)[2] <- 3
   levels(neo.orig$worrier)[3] <- 2
   levels(neo.orig$worrier)[4] <- 0
   levels(neo.orig$worrier)[5] <- 4

#levels(neo.orig$daydream)
   levels(neo.orig$daydream)[1] <- 1
   levels(neo.orig$daydream)[2] <- 3
   levels(neo.orig$daydream)[3] <- 2
   levels(neo.orig$daydream)[4] <- 0
   levels(neo.orig$daydream)[5] <- 4

#!!PLEASE NOTE!!: We list code here to get new scales for only two variables: worrier
and daydream.  THE SAME CODE TO RE-LEVEL THE SCALE OF THE FOLLOWING 25 VARIABLES
SHOULD BE INCLUDED. We skip those repeated and lengthy codes and do not list them
here.

#25 variables are: stick2, argue, lightheart, selfish, methodical, blue #controversy,
poetry, cynical, alone, takeadv, wastetime, anxious, moods, #moral, cold, optimist,
hardhead, reliable, sad, universe, dislike, #organized, leader, manipulate


#-------- REVERSE Scales --------
#Strongly Agree -> 4, Agree -> 3, Neutral -> 2, Disagree -> 1, Strongly Disagree -> 0

   #levels(neo.orig$ppl_around)
   #"Agree"   "Disagree"  "Neutral"   "Strongly Agree"  "Strongly Disagree"
   levels(neo.orig$ppl_around)[1] <- 3
   levels(neo.orig$ppl_around)[2] <- 1
   levels(neo.orig$ppl_around)[3] <- 2
   levels(neo.orig$ppl_around)[4] <- 4
```

```
   levels(neo.orig$ppl_around)[5] <- 0

   #levels(neo.orig$courteous)
   levels(neo.orig$courteous)[1] <- 3
   levels(neo.orig$courteous)[2] <- 1
   levels(neo.orig$courteous)[3] <- 2
    levels(neo.orig$courteous)[4] <- 4
   levels(neo.orig$courteous)[5] <- 0

#!!PLEASE NOTE!! We list code here to get new scales for only two variables:
ppl_around and courteous.  THE SAME CODE TO RE-LEVEL THE SCALE OF THE FOLLOWING 31
VARIABLES SHOULD BE INCLUDED. We skip those repeated and lengthy codes and do not
list them here.

#neat, inferior, laugh, pacing, stress, patterns, talking, cooperate, tense,
#conscientious, action, cleargoals, worthless, newfoods, energy, liked,
#acmplshgoals, angry, cheerful, commitment, discouraged, excitement, fast,
#considerate, productive, helpless, active, curious, ashamed, abstract, excellence


Neuroticism.data <- data.frame( as.numeric(as.character(neo.orig$tense)),
as.numeric(as.character(neo.orig$worthless)),
as.numeric(as.character(neo.orig$anxious)), as.numeric(as.character(neo.orig$angry)),
as.numeric(as.character(neo.orig$discouraged)),
as.numeric(as.character(neo.orig$sad)),
as.numeric(as.character(neo.orig$worrier)),
as.numeric(as.character(neo.orig$inferior)),
as.numeric(as.character(neo.orig$stress)),
as.numeric(as.character(neo.orig$helpless)),
as.numeric(as.character(neo.orig$ashamed)), as.numeric(as.character(neo.orig$blue)))
    names(Neuroticism.data) <- c("tense","worthless","anxious", "angry",
"discouraged", "sad", “worrier","inferior","stress"," helpless","ashamed", “blue")

   Extraversion.data <- data.frame(as.numeric(as.character(neo.orig$lightheart)),
as.numeric(as.character(neo.orig$talking)),
as.numeric(as.character(neo.orig$leader)), as.numeric(as.character(neo.orig$action)),
as.numeric(as.character(neo.orig$alone)), as.numeric(as.character(neo.orig$energy)),
as.numeric(as.character(neo.orig$cheerful)),
as.numeric(as.character(neo.orig$optimist)),
as.numeric(as.character(neo.orig$fast)),
as.numeric(as.character(neo.orig$ppl_around)),
as.numeric(as.character(neo.orig$laugh)), as.numeric(as.character(neo.orig$active)))
     names(Extraversion.data) <-
c("lightheart","talking","leader","action","alone","energy","cheerful",
"optimist","fast","ppl_around","laugh","active")

   Openness.data <- data.frame(as.numeric(as.character(neo.orig$daydream)),
as.numeric(as.character(neo.orig$stick2)),
as.numeric(as.character(neo.orig$universe)),
as.numeric(as.character(neo.orig$curious)),
as.numeric(as.character(neo.orig$patterns)),
as.numeric(as.character(neo.orig$controversy)),
as.numeric(as.character(neo.orig$abstract)),
as.numeric(as.character(neo.orig$poetry)),
```

```
as.numeric(as.character(neo.orig$newfoods)),
as.numeric(as.character(neo.orig$moods)),
as.numeric(as.character(neo.orig$moral)),
as.numeric(as.character(neo.orig$excitement)))
     names(Openness.data) <-
c("daydream","stick2","universe","curious","patterns","controversy","abstract","poetr
y", "newfoods","moods","moral","excitement")

   Agreeableness.data <- data.frame(as.numeric(as.character(neo.orig$cold)),
as.numeric(as.character(neo.orig$hardhead)),
as.numeric(as.character(neo.orig$courteous)),
as.numeric(as.character(neo.orig$argue)),
as.numeric(as.character(neo.orig$considerate)),
as.numeric(as.character(neo.orig$dislike)),
as.numeric(as.character(neo.orig$selfish)),
as.numeric(as.character(neo.orig$cooperate)),
as.numeric(as.character(neo.orig$manipulate)),
as.numeric(as.character(neo.orig$cynical)),
as.numeric(as.character(neo.orig$takeadv)), as.numeric(as.character(neo.orig$liked)))
     names(Agreeableness.data) <-
c("cold","hardhead","courteous","argue","considerate","dislike","selfish",
"cooperate", "manipulate","cynical","takeadv","liked")

 Conscientiousness.data <- data.frame(as.numeric(as.character(neo.orig$wastetime)),
as.numeric(as.character(neo.orig$acmplshgoals)),
as.numeric(as.character(neo.orig$commitment)),
as.numeric(as.character(neo.orig$reliable)),
as.numeric(as.character(neo.orig$neat)), as.numeric(as.character(neo.orig$pacing)),
as.numeric(as.character(neo.orig$productive)),
as.numeric(as.character(neo.orig$organized)),
as.numeric(as.character(neo.orig$methodical)),
as.numeric(as.character(neo.orig$conscientious)),
as.numeric(as.character(neo.orig$excellence)),
as.numeric(as.character(neo.orig$cleargoals)))

    names(Conscientiousness.data) <-
c("wastetime","acmplshgoals","commitment","reliable"," productive","pacing",
"neat","organized","methodical","conscientious","excellence","cleargoals")

   Neuroticism <- apply(Neuroticism.data,1,sum,na.rm=T)
   Extraversion <- apply(Extraversion.data,1,sum,na.rm=T)
   Openness <- apply(Openness.data,1,sum,na.rm=T)
   Agreeableness <- apply(Agreeableness.data,1,sum,na.rm=T)
   Conscientiousness <- apply(Conscientiousness.data,1,sum,na.rm=T)

   size.Neuroticism <- apply( is.na(Neuroticism.data)==F,1,sum)
   size.Extraversion <- apply( is.na(Extraversion.data)==F,1,sum)
   size.Openness <- apply( is.na(Openness.data)==F,1,sum)
   size.Agreeableness <- apply( is.na(Agreeableness.data)==F,1,sum)
   size.Conscientiousness <- apply( is.na(Conscientiousness.data)==F,1,sum)

  neo.orig <- data.frame(neo.orig, Neuroticism, size.Neuroticism, Extraversion,
size.Extraversion, Openness, size.Openness,
Agreeableness, size.Agreeableness, Conscientiousness, size.Conscientiousness)
```

Personal History (PERSHX) Dataset

These variables cover the smoking and alcohol intake histories of the Participants.

_SMOKENOW, _PIPENOW, and _PACKYRS

_SMOKENOW is current smoker. _PIPENOW is current pipe user. _PACKYRS is the number of packs smoked per day over the number of years smoked. (# Packs/day * yrs smoke(d)).

```
length _Smokenow _Pipenow 3;
  merge age(in=a) smoke(in=b);
  by id; if b=1;
  if smoke100=0 or smokequitage>0 or smokequityr>0 then _Smokenow=0;
    else _Smokenow=smokenow;
  label _Smokenow="Current Smoker?";
  if pipe=0 or pipequitage>0 or pipequityr>0 then _Pipenow=0;
    else _Pipenow=pipenow;
  label _Smokenow="Current Smoker?" _Pipenow="Current Pipe User?";
  format _Smokenow _Pipenow YND12.;
  if smokenow not in (0,1) then _PACKYRS=.;
  if cigday=. then _PACKYRS=.;
  if _age =. then _PACKYRS=.;
  if smokenow=1
    then _PACKYRS=(_age - smoke1stage)*(cigday/20);
    else if (smokenow=0 and smoke1styr ne . and smokequityr ne . and cigday ne .)
        then _PACKYRS=(smokequityr-smoke1styr)*(cigday/20);
    else if (smokenow=0 and smoke1stage ne . and smokequitage ne . and cigday ne .)
        then _PACKYRS=(smokequitage-smoke1stage)*(cigday/20);
  _PACKYRS=round(_PACKYRS,.01);
  label _PACKYRS="# Packs/day * yrs smoke(d)";
```

_SUMPACKYRS

_SUMPACKYRS is similar to _PACKYRS, above; however, non-smokers are now coded as 0 instead of missing, and the number of years smoked is calculated from the start and quit dates or ages smoked.

```
_datayr=year(date);
  if ((_smokenow=0) and (smoke100 ne 1)) then _sumpackyrs=0; /* never smoker */
  if ((_smokenow=1) and (smoke1stage^=.)) then _sumpackyrs=(_age -
smoke1stage)*(cigday/20); /* current smoker, using age */
  else if ((_smokenow=1) and (smoke1styr^=.)) then _sumpackyrs=(_datayr -
smoke1styr)*(cigday/20); /* current smoker, using year */

  if ((_smokenow=0) and (smokenow=0) and (smoke100=1) and (smoke1stage^=.) and
(smokequitage^=.) and (cigday^=.))
```

```
  then _sumpackyrs=(smokequitage - smoke1stage)*(cigday/20); /* former smoker, using age
start and quit */
  else if ((_smokenow=0) and (smokenow=0) and (smoke100=1) and (smoke1styr^=.) and
(smokequityr^=.) and (cigday^=.))
  then _sumpackyrs=(smokequityr - smoke1styr)*(cigday/20); /* former smoker, using year
start and quit */

  _yrstart=dobyr+smoke1stage; /* create year start smoking variable to use for missing former
smokers */
  _yrend=dobyr+smokequitage; /* create year end smoking variable to use for missing former
smokers */

  if ((_smokenow=0) and (smokenow=0) and (smoke100=1) and (_yrstart^=.) and
(smokequityr^=.) and (cigday^=.))
  then _sumpackyrs=(smokequityr - _yrstart)*(cigday/20); /* former smoker, using year start
 and quit with extrapolated year start*/
  else if ((_smokenow=0) and (smokenow=0) and (smoke100=1) and (smoke1styr^=.) and
(_yrend^=.) and (cigday^=.))
  then _sumpackyrs=(_yrend - smoke1styr)*(cigday/20); /* former smoker, using year start and
quit with extrapolated year end*/
LABEL _sumpackyrs="Derived # packs/day * years smoke(d), for current smokers, former
smokers, and non-smokers";
  if _sumpackyrs=-0.2 then _sumpackyrs=.;
  if _sumpackyrs=-0.6 then _sumpackyrs=.;
```

_SMOKE_CIG, _SMOKE_PIPE, and _SMOKE_CAT

_SMOKE_CIG is current, former, or never cigarette smoker. _SMOKE_PIPE is current, former, or never pipe smoker. _SMOKE_CAT is current, former, or never cigarette or pipe smoker.

```
Proc format;
Value smkcat
1 = “Never smoked”
2 = “Former smoker”
3 = “Current smoker”;
Run;
```

****Cigarette smoking only**;**

```
if smoke100=0 then _smoke_cig=1;
   else if smoke100=1 and _smokenow=0 then _smoke_cig=2;
   else if smoke100=1 and _smokenow=1 then _smoke_cig=3;
```

****Pipe smoking only**;**

```
if pipe=0 then _smoke_pipe=1;
   else if pipe=1 and _pipenow=0 then _smoke_pipe=2;
   else if pipe=1 and _pipenow=1 then _smoke_pipe=3;
```

```
**Cigarette + Pipe smoking**;

if _smoke_cig=1 and _smoke_pipe=1 then _smoke_cat=1;
   else if _smoke_cig=1 and _smoke_pipe=2 then _smoke_cat=2;
   else if _smoke_cig=1 and _smoke_pipe=3 then _smoke_cat=3;
   else if _smoke_cig=2 and _smoke_pipe=1 then _smoke_cat=2;
   else if _smoke_cig=2 and _smoke_pipe=2 then _smoke_cat=2;
   else if _smoke_cig=2 and _smoke_pipe=3 then _smoke_cat=3;
   else if _smoke_cig=3 and _smoke_pipe=1 then _smoke_cat=3;
   else if _smoke_cig=3 and _smoke_pipe=2 then _smoke_cat=3;
   else if _smoke_cig=3 and _smoke_pipe=3 then _smoke_cat=3;
```

Physical Function and Activity (PHYSICAL) Dataset

This dataset covers the Physical Exercise form. Variables in the dataset include, activity level, duration, and frequency of exercise. In visit 2, the Pittsburgh Fatigability Scale Test and the Framingham Activity Scale were added to this form. This dataset has no derived variables.

Performance Measures (PM) Dataset

This dataset includes the results of the Short Physical Performance Battery (SPPB), and the Grip Strength Test.

_TOTSCORE

This derived variable provides one score for the entire SPPB.

WALKSCORE: gives ratings for various values of the walking test.

```
if length eq 1 then do;
   if 0 < shorter < 4.82 then walkscore = 4;
   else if 4.82 <= shorter <= 6.20 then walkscore = 3;
   else if 6.21 <= shorter <= 8.70 then walkscore = 2;
   else if 8.70 < shorter < 60.00 then walkscore = 1;
 end;
 if length eq 2 then do;
   if 0 < shorter < 3.62 then _walkscore = 4;
   else if 3.62 <= shorter <= 4.65 then walkscore = 3;
   else if 4.66 <= shorter <= 6.53 then walkscore = 2;
   else if 6.53 < shorter < 45.00 then walkscore = 1;
 end;
```

```
_totscore = sum (sidescore,semiscore,tdmscore,walkscore,chairscore);
```

Prevalence of Disease (PREVDISEASE) Dataset

This dataset was created to derive 11 variables that combine information from the Medical History, Blood Pressure, Blood, and Coded Medications data sets. The 7 derived prevalence disease variables (_htdis, _strk, _lungdis, _htn, _diabetes, _pad and _cancer) from Visit 1 were completed at Pittsburgh site. DMCC recreated 5 of them (_htdis, _strk, _lungdis, _pad, and _cancer) for Visit 2 based on the code used in Visit 1. _pad was recreated for only Visit 2 new participants because ankle-arm blood pressure ratio (aabprl and aabprr) were not measured for the returning participants. The other two derived variables are recreated when the codedmeds_atc dataset for Visit 2 is ready, and they are renamed to _htn_atc and _diabetes_atc. Two more variables for diabetes were derived according to American Diabetes Association (ADA) that used HbA1C as the gold standard for classification of diabetes. They are _adat2d and _adat2d_age_reported_detected.

_HTDIS, _STRK, _LUNGDIS, _HTN, _DIABETES, _PAD, _CANCER:

```
proc sort data=clinic.medhx out=medhx; by id; run;
proc sort data=clinic.bphr out=bphr; by id; run;
proc sort data=blood.blood out=blood; by id; run;

proc sort data=codemeds.codedmeds (keep=id htnrx lipidrx dmrx) out=cmeds;
by id; run;
/* For visit 2:
proc sort data=codemeds.codedmeds_atc (keep=id htn diab rename=(htn= htnrx diab=dmrx)) out=cmeds;
by id; run;
*/
proc sort data=clinic.sdi (keep=id sex) out=sdi; by id; run;

data medhxvars(keep= id  _htdis  _strk _lungdis);
*data medhxvars(keep= id midx cabg _htdis stroke tia _strk asth bronch copd _lungdis);
  set medhx;
*create prevalent disease heart disease*;
if (midx=1 or cabg=1) then _htdis=1;   else _htdis=0;
*create prevalent disease stroke*;

if (stroke=1 or tia=1) then _strk=1;
   else _strk=0;
*create prevalent disease lung diseaase*******;
if (asth=1 or bronch=1 or copd=1) then _lungdis=1;
   else _lungdis=0;
run;
proc sort; by id; run;

data htn(keep=id htdx);  set medhx;   run;
proc sort; by id; run;

data htn1 (keep=id htdx htnrx);
  merge htn(in=a) cmeds(in=b);
  by id;   if a;
run;
```

```
proc sort; by id; run;
data bp(keep=id _htn);
*data bp(keep=id htdx htnrx sys1-sys3 avgsys dia1-dia3 avgdia _htn);
  merge htn1(in=a) bphr(in=b);
  by id;  if a;
  avgsys=mean(sys1,sys2,sys3);
  avgdia=mean(dia1,dia2,dia3);
  if  ((htdx=1 and htnrx=1)
      or (avgsys >=140)
      or (avgdia >=90))
      then _htn=1;
      else _htn=0;
run;
proc sort; by id; run;
data diab(keep=id diab);   set medhx;   run;
proc sort; by id; run;
data diab1(keep=id _diabetes);
*data diab1(keep=id diab dmrx glur _diabetes);
  merge diab(in=a)  cmeds(in=b) blood(in=c);
  by id;   if a;
  if (diab=1 and dmrx=1 or glur >=126)
     then _diabetes=1;
     else _diabetes=0;
run;
proc sort; by id; run;

data pad(keep=id _pad);
*data pad(keep=id aai aabprl aabprr _pad);
  set bphr;
  aai = min(aabprl,aabprr);
  if (aai <= 0.9 and aai > 0) then _pad = 1;
  else if aai > 0.9 then _pad=0;
  run;
proc sort; by id; run;

data cancer(keep=id _cancer);
*data cancer(keep=id sex breast leuk colon lung melan skin esophgl pancr ocancer prost _cancer);
  merge medhx(in=a) sdi(in=b);
  by id;   if a;
  if (breast=1 or leuk=1 or colon=1 or lung=1 or melan=1 or skin=1 or esophgl=1
    or ocancer=1 or prost=1) and sex=1 then _cancer=1;
  if (breast=1 or leuk=1 or colon=1 or lung=1 or melan=1 or skin=1 or esophgl=1
    or ocancer=1) and sex=2 then _cancer=1;
  if _cancer ne 1 then _cancer=0;
  run;
proc sort; by id; run;

data codemeds.prevdisease;
  merge medhxvars(in=a) bp(in=b) diab1(in=c) pad(in=d) cancer(in=e);
  by id;   if a;
  if id='' then delete;
```

```
 if _htdis=.    then _htdis=0;
 if _strk=.     then _strk=0;
 if _htn=.      then _htn=0;
 if _diabetes=. then _diabetes=0;
 if _cancer=.   then _cancer=0;
 if _lungdis=.  then _lungdis=0;
 if _pad=.      then _pad=0;
 ** At visit 2, aabprl aabprr were measured for new participants, not for returning participants;
 ** Leave _pad missing for returning participants;
 run;
```

_HTN_ATC, and _DIABETES_ATC

```
** codemeds for visit 2 (based on ATC coding scheme);
proc sort data=codemeds_atc
  out=cmeds(keep=subject htn lipid diab rename=(htn=htnrx lipid=lipidrx diab=dmrx));
  by subject;   where visitcode in (3,4);
run;

proc sort data=medhxall out=htn(keep=subject htdx);  by subject;  where visitcode in (3,4);  run;
data htn1 (keep=subject htdx htnrx);
  merge htn(in=a) cmeds(in=b);
  by subject;   if a;
run;
proc sort; by subject; run;

proc sort data=bphrall out=bphr(keep=subject sys1 sys2 sys3 dia1 dia2 dia3);  by subject;  where
visitcode in (3,4); run;

data bp(keep=subject _htn_atc);
  merge htn1(in=a) bphr(in=b);
  by subject;  if a;
  avgsys=mean(sys1,sys2,sys3);
  avgdia=mean(dia1,dia2,dia3);
  if  ((htdx=1 and htnrx=1)
      or (avgsys >=140)
      or (avgdia >=90))
      then _htn_atc=1;
      else _htn_atc=0;
      label _htn_atc = '_htn, ATC coding';
run;

proc sort data=medhxall out=diab(keep=subject diab); by subject; where visitcode in (3,4);  run;
proc sort data=bloodall out=blood nodupkey;  by subject;  where visitcode in (3,4); run;

data diab1(keep=subject _diabetes_atc);
  merge diab(in=a) cmeds(in=b) blood(in=c);
  by subject;   if a;
  if (diab=1 and dmrx=1 or _glur_new >=126)
```

```
    then _diabetes_atc=1;
    else _diabetes_atc=0;
    label _diabetes_atc = '_diabetes, ATC code';
run;
```

_ADAT2D and _ADAT2D_AGE_REPORTED_DETECTED

```
proc sort data= medhxall out=medhx(keep=subject visitcode date diab diabage          diabnow
rename=(date=medhx_date)) nodupkey;
 by subject visitcode;
run;

proc sort data= bloodall out=blood(keep=subject visitcode _glur_new glyhb) dupout=dupblood
nodupkey;
 by subject visitcode;
run;
****The data set WORK.DUP has 126 observations (blood re-draw at visit 2);

**visit 2: fasting_lipid=1 that indicates blood sample was used to run lipid biomarker test;
data venipv1 venipv2;
 set venipall (keep=subject visitcode _fast fasting_lipid date rename=(date=venip_date));
 if visitcode=1 and _fast=1 then output venipv1;
 if visitcode in (3,4) and fasting_lipid=1 then output venipv2;
run;

data venip; set venipv1 venipv2; run;

proc sort data=venip dupout=dupvenip nodupkey; by subject visitcode;  run;

** ATC medication coding scheme;
proc sort data=codedmeds_atc (keep=subject visitcode diab)
        out=cmeds(rename=(diab=dmrx));
by subject visitcode;
run;

proc sort data=sdiall out=sdi(keep=subject visitcode _AGE_REVISED) nodupkey;
 by subject visitcode;
run;

********* (1) derive  _ADAT2D;
data tmp;
 merge medhx(in=a) cmeds(in=b) blood(in=c) venip sdi;
  by subject visitcode;
  if (a or b or c) and visitcode in (1,3,4);
  if (_fast>=1 and _glur_new>=126) or glyhb >=6.5 or diabnow=1 or diab=1 or dmrx=1 then _adat2d=1;
   /*if all data in datasets medhx, biomarkers, venipuncture and codedmeds
used for deriving _adat2d is missing, set _adat2d missing */
```

```
  else if medhx_date in (.,.R,.U,.N) and dmrx=. and glyhb=. and _glur_new=. and venip_date in
(.,.R,.U,.N) then _adat2d=.;
  else _adat2d=0;
 label _adat2d='Diabetes? (ADA classification)';
run;

**********(2) derive  _ADAT2D_AGE_REPORTED_DETECTED;
data v1 v2;
 set tmp(keep=subject visitcode _adat2d diabage _age_revised);
 if visitcode=1 then output v1;          /*visit 1*/
 if visitcode in (3,4) then output v2;    /*visit 2*/
run;

proc sort data=v1 nodupkey; by subject; run;
proc sort data=v2 nodupkey; by subject; run;

data v1v2;
 merge v1(in=in1) v2(in=in2 rename=(visitcode=visitcodev2 _adat2d=_adat2dv2 diabage=diabagev2
_age_revised=_age_revisedv2));
 by subject;
 if in1 or in2;

 /*diabetes: YES for both visit 1 and visit 2*/
 /*if more than one age onset was reported, in general, the first age of onset is used, as it is likely closest
to the event.*/
 if _adat2d=1 and _adat2dv2=1 then do;
   if diabage>0 then do;
    _ADAT2D_AGE_REPORTED_DETECTED1 = diabage;
    _ADAT2D_AGE_REPORTED_DETECTED2 = diabage;
   end;
   else if diabage<=0 and diabagev2>0 then do;
    if diabagev2 < _age_revised then do;
      _ADAT2D_AGE_REPORTED_DETECTED1 = diabagev2;
      _ADAT2D_AGE_REPORTED_DETECTED2 = diabagev2;
    end;
    if diabagev2 >= _age_revised then do;
      _ADAT2D_AGE_REPORTED_DETECTED1  = _age_revised;
      _ADAT2D_AGE_REPORTED_DETECTED2 = _age_revised;
    end;
   end;
   else if diabage<=0 and diabagev2<=0 then do;
    _ADAT2D_AGE_REPORTED_DETECTED1  = _age_revised;     /*if reported age onset missing, use
age at visit 1 in person as age onset*/
    _ADAT2D_AGE_REPORTED_DETECTED2 = _age_revised;
   end;
 end;

 /*diabetes: YES for visit 2, not yes for visit 1 */
 else if _adat2d ^= 1 and _adat2dv2 = 1 then do;
  if diabagev2 ^=. then _ADAT2D_AGE_REPORTED_DETECTED2 = diabagev2;
```

```
 if diabagev2 =. then _ADAT2D_AGE_REPORTED_DETECTED2 = _age_revisedv2; /*if reported age
onset missing, use age at visit 2 in person as age onset*/
 end;

 /*diabetes: YES for visit 1, NO for visit 2*/
 /*According to Ping An, once disbetes was detected, then participant has diabetes*/
 else if _adat2d = 1 and _adat2dv2 = 0 then do;
  _adat2dv2=1;
  if diabage ^=. then do;
    _ADAT2D_AGE_REPORTED_DETECTED1 = diabage;
    _ADAT2D_AGE_REPORTED_DETECTED2 = diabage;
  end;
  if diabage = . and _age_revised >0 then do;
    _ADAT2D_AGE_REPORTED_DETECTED1 = _age_revised;
    _ADAT2D_AGE_REPORTED_DETECTED2 = _age_revised;
  end;
 end;
 else if _adat2d = 1 and _adat2dv2 = . then do;
  if diabage ^=. then _ADAT2D_AGE_REPORTED_DETECTED1 = diabage;
  if diabage = . then _ADAT2D_AGE_REPORTED_DETECTED1 = _age_revised;
 end;
run;

data tmp2;
 set v1v2(keep=subject visitcode _adat2d _ADAT2D_AGE_REPORTED_DETECTED1
where=(visitcode=1)
rename=(_ADAT2D_AGE_REPORTED_DETECTED1=_ADAT2D_AGE_REPORTED_DETECTED))
    v1v2(keep=subject visitcodev2 _adat2dv2 _ADAT2D_AGE_REPORTED_DETECTED2
where=(visitcode in (3,4) and _adat2d>=0)
       rename=(visitcodev2=visitcode _adat2dv2=_adat2d
_ADAT2D_AGE_REPORTED_DETECTED2=_ADAT2D_AGE_REPORTED_DETECTED));
 label _ADAT2D_AGE_REPORTED_DETECTED = 'Age onset of diabetes: self-reported or detected';
run;

proc sort data=tmp2; by subject visitcode; run;
proc sort data=prevdiseaseall; by subject visitcode; run;

data prevdiseaseall;
 merge prevdiseaseall tmp2;
 by subject visitcode;
run;
```

Socio-Demographic Information (SDI) Dataset

This dataset contains the information collected on the Socio-Demographic form.

_AGE

_AGE is the age when the cognitive assessment form was filled out.

_AGE_REVISED

_AGE_REVISED is the age when the consent form was filled out.

```
_AGE=floor((date-dob)/365.25); **where date is from cogassessall dataset;
_AGE_REVISED=floor((date-dob)/365.25); **where date is from ctifall
dataset;
```

Spirometry Safety Questionnaire (SPIRO) Dataset

The Spirometry Safety Questionnaire asks background questions that would preclude taking the pulmonary function test, such as major surgery, heart attack, or stroke in the past three months. There are no derived variables in this dataset.

Spirometry Safety Questionnaire (SPIROMEDS) Dataset

This provides the medication data from question 9b of the Spirometry Safety Questionnaire. This data set has 1 record per medication; therefore, multiple records per Participant. It includes the person's ID to link this dataset to the others. This dataset has no derived variables.

Survival Indices (SURVL_INDICES) Dataset

This dataset contains derived variables that are indicators of survival and healthy aging.

Healthy Aging Index

Variables related to the healthy aging index (A. Newman, R. Minster, J. Sanders, et al, Pittsburgh) are ***HAI, HAI_m, HAI_rg, HAI_m_rg, HAI_rl,*** and ***HAI_m_rl***.

HAI

HAI is the evenly weighted healthy aging index. It is calculated from 5 other variables and thus is only calculated when a LLFS participant has a measurement for all 5 component variables. The 5 component variables are: systolic blood pressure, forced vital capacity, mini-mental state exam, serum creatinine, and serum fasting glucose.
To create the ***HAI***, each component variable receives a score of 0 (healthiest tertile), 1 (middle tertile), or 2 (unhealthiest tertile)—with the exception of fasting glucose, for which clinical cutoffs were applied. For systolic blood pressure, if a participant has a physician diagnosis of hypertension or if a participant was using medication for hypertension, they were coded in the unhealthiest tertile (score=2). Similarly for fasting glucose, if a participant has a physician diagnosis of diabetes or if a participant was using medication for diabetes, they were coded in the unhealthiest tertile (score=2). For forced expiratory volume and serum creatinine, separate tertiles were applied to men and women. The scores of the five component variables were then summed for each participant to create the HAI which has a range of 0 (healthiest) to 10 (unhealthiest).

HAI Tertile Thresholds

		0	1	2
Systolic blood pressure, mmHg*		< 126	≥ 126 and <143	≥ 143
Forced vital capacity, L	(*women*)	≥ 2.61	< 2.61 and ≥ 2.14	< 2.14
	(*men*)	≥ 3.84	< 3.84 and ≥ 3.19	< 3.19
MMSE, points		> 26	> 23 and ≤ 26	≤ 23
Serum creatinine, mg/dL	(*women*)	< 0.8	0.8–1.0	> 1.0
	(*men*)	< 1.1	1.1–1.3	> 1.3
Serum fasting glucose, mg/dL†		< 100	100–125	≥ 126

*Physician diagnosis of hypertension or taking anti-hypertensive medication led to score=2

† Physician diagnosis of diabetes or taking medication for diabetes led to score=2

HAI_rg

HAI_rg is the age, sex, and PC1-10 adjusted residuals of the HAI used for the GWAS.

HAI_m

HAI_m is the mortality weighted healthy aging index, where each component variable of the HAI is given a mortality optimized weight. Each component score was multiplied by the weight for that component, divided by the sum of all 5 component weights, and multiplied by 5 to obtain the weight for that component. Each component weight was then multiplied to the score for that component and all 5 weighted component scores were summed to calculate the HAI_m. When we calculated the cox models in CHS for each one point increase in the index we had the betas for mortality. We used the betas as modifiers for the component heritability weights (which before were all equal at 0.20) in the index. That way the weights were optimized for mortality prediction, with the most strongly associated components receiving a greater weight.

HAI_m Weights

	Weight
Systolic blood pressure, mmHg*	0.17085
Forced vital capacity, L	0.38386
MMSE, points	0.42873
Serum creatinine, mg/dL	0.13397
Serum fasting glucose, mg/dL†	0.23880

HAI_m_rg

HAI_m_rg is the age, sex, and PC1-10 adjusted residuals of the mortality weighted HAI, HAI_m, used for the GWAS.

HAI_rl

HAI_rl is the age and sex adjusted residuals of the HAI used for the linkage analysis.

HAI_m_rl

HAI_m_rl is the age and sex adjusted residuals of the mortality weighted HAI, HAI_m, used for the linkage analysis.

R code (from R. Minster)

```
# Objective: Calculate HAI residuals in LLFS.

# Clean up workspace

rm(list = ls(all = TRUE))

 # Set working directory

setwd("/Users/rminster/Documents/Professional/bz-llfs-hai/")

 # Load libraries.

 # Read in data

a <- read.csv("../03-llfs-data/llfsdata_csv_20120413/lungfunc.csv")
b <- read.csv("../03-llfs-data/llfsdata_csv_20120413/gtriplet_v2.csv")
c <- read.csv("../03-llfs-data/llfsdata_csv_20120413/codedmeds.csv")
d <- read.csv("../03-llfs-data/llfsdata_csv_20120413/cogassess.csv")
e <- read.csv("../03-llfs-data/llfsdata_csv_20120413/blood.csv")
f <- read.csv("../03-llfs-data/llfsdata_csv_20120413/bphr.csv")
g <- read.csv("../03-llfs-data/llfsdata_csv_20120413/pm.csv")
h <- read.csv("../03-llfs-data/llfsdata_csv_20120413/sdi.csv")
i <- read.csv("../03-llfs-data/llfsdata_csv_20120413/venip.csv")
j <- read.csv("../03-llfs-data/llfsdata_csv_20120413/pershx.csv")
k <- read.csv("../03-llfs-data/llfsdata_csv_20120413/medhx.csv")

 # Merge data into a single file

data <- merge(a, b, by = "subject")
data <- merge(data, c, by = "subject", all.x = TRUE)
data <- merge(data, d, by = "subject")
data <- merge(data, e, by = "subject")
data <- merge(data, f, by = "subject")
data <- merge(data, g, by = "subject")
data <- merge(data, h, by = "subject")
data <- merge(data, i, by = "subject")
data <- merge(data, j, by = "subject")
data <- merge(data, k, by = "subject")

rm(a, b, c, d, e, f, g, h, i, j, k)

 # Convert data to correct data type

data$sys1 <- as.integer(as.character(data$sys1))
data$sys2 <- as.integer(as.character(data$sys2))

data$creatr <- as.numeric(as.character(data$creatr))
```

```
data$glur <- as.integer(as.character(data$glur))

n <- nrow(data)

data$sbp <- (data$sys1 + data$sys2) / 2
data$fvcl <- data$fvc / 1000
data$proband <- as.integer(data$gen == 2 & data$control == 0)
data$offspring <- as.integer(data$gen == 3 & data$control == 0)
data$control <- as.integer(data$gen == 3 & data$control == 1)
data$gender <- data$sex.x
levels(data$gender) <- c(NA, "F", "M")

  # Code from healthiest (0) to least healthy (2)

sbp_t <- rep(NA, n)
sbp_t[data$sbp < 126] <- 0
sbp_t[data$sbp >= 126 & data$sbp < 143] <- 1
sbp_t[data$sbp >= 143] <- 2
sbp_t[data$htnrx == 1] <- 2
sbp_t[data$htdx == 1] <- 2
data <- data.frame(data, sbp_t)

creat_t <- rep(NA, n)
creat_t[data$gender == "F" & data$creatr < 0.8] <- 0
creat_t[data$gender == "F" & data$creatr >= 0.8 & data$creatr <= 1.0] <- 1
creat_t[data$gender == "F" & data$creatr > 1.0] <- 2
creat_t[data$gender == "M" & data$creatr < 1.1] <- 0
creat_t[data$gender == "M" & data$creatr >= 1.1 & data$creatr <= 1.3] <- 1
creat_t[data$gender == "M" & data$creatr > 1.3] <- 2
data <- data.frame(data, creat_t)

fvc_t <- rep(NA, n)
fvc_t[data$gender == "F" & data$fvcl >= 2.61] <- 0
fvc_t[data$gender == "F" & data$fvcl < 2.61 & data$fvcl >= 2.14] <- 1
fvc_t[data$gender == "F" & data$fvcl < 2.14] <- 2
fvc_t[data$gender == "M" & data$fvcl >= 3.84] <- 0
fvc_t[data$gender == "M" & data$fvcl < 3.84 & data$fvcl >= 3.19] <- 1
fvc_t[data$gender == "M" & data$fvcl < 3.19] <- 2
data <- data.frame(data, fvc_t)

gluc_t <- rep(NA, n)
gluc_t[data$glur < 100] <- 0
gluc_t[data$glur >= 100 & data$glur < 126] <- 1
gluc_t[data$glur >= 126] <- 2
gluc_t[data$X_FASTTIME <= 6] <- NA
gluc_t[data$dmrx > 0] <- 2
gluc_t[data$diab == 1] <- 2
data <- data.frame(data, gluc_t)

  # From Mikeâ€™s code -- from comparison of old data, Amy Matteiniâ€™s codings and CHS

mmse_t <- rep(NA, n)
mmse_t[data$mmsetot > 26] <- 0
mmse_t[data$mmsetot > 23 & data$mmsetot <= 26] <- 1
mmse_t[data$mmsetot <= 23] <- 2
table(mmse_t)
```

```
data <- data.frame(data, mmse_t)

 # Subset data to what is needed

data <-data[, c("subject", "offspring", "proband", "control", "gen", "fc.x",
       "X_AGE", "mmse_t", "sbp_t", "creat_t", "fvc_t", "gluc_t", "gender")]
names(data) <- c("id", "o", "p", "c", "generation", "center",
              "age", "mmse", "sbp", "creat", "fvc", "gluc", "sex")
data$center <- as.integer(data$center)
data$center2[data$center == 2] <- 1
data$center2[is.na(data$center2)] <- 0
data$center3[data$center == 3] <- 1
data$center3[is.na(data$center3)] <- 0
data$center4[data$center == 4] <- 1
data$center4[is.na(data$center4)] <- 0
#data[data$c == 1, 3:ncol(data)] <- NA
data <- data[!duplicated(data$id), ]

 # Equal weighting

w1 <- 0.2
w2 <- 0.2
w3 <- 0.2
w4 <- 0.2
w5 <- 0.2

data$hai <- 5 * (w1 * data$mmse +
              w2 * data$sbp +
              w3 * data$creat +
              w4 * data$gluc +
              w5 * data$fvc)

 # Mortality-optimized weighting

w1 <- 0.42873
w2 <- 0.17085
w3 <- 0.13397
w4 <- 0.23880
w5 <- 0.38386

w <- sum(w1, w2, w3, w4, w5)

w1 <- w1 / w * 5
w2 <- w2 / w * 5
w3 <- w3 / w * 5
w4 <- w4 / w * 5
w5 <- w5 / w * 5

data$hai_m <- w1 * data$mmse +
            w2 * data$sbp +
            w3 * data$creat +
            w4 * data$gluc +
            w5 * data$fvc

table(data$o, data$p, data$c, exclude = NULL, deparse.level = 2)
```

```
names(data)[1] <- "subject"
data <- data[complete.cases(data$hai), ]

m <- lm(hai ~ age + sex, data = data, na.action = na.exclude)
data$hai_rl <- residuals(m)

m <- lm(hai_m ~ age + sex, data = data,
      na.action = na.exclude)
data$hai_m_rl <- residuals(m)

write.table(data[data$c == 0, c("subject", "hai_rl", "hai_m_rl")], "3d5c-hai-linkage.csv",
  sep = ",", row = FALSE, quote = FALSE)

pcs <- read.csv("../bl-llfs-genetic-data/2vc-ancestry-pc's/llfseignvec.csv")
i <- read.table("2ve2-genetic-data-ids.txt", header = TRUE)

data <- merge(data[data$subject %in% i$subject, ], pcs[pcs$outlier == 0, ])

m <- lm(hai ~ age + sex + pc1 + pc2 + pc3 + pc4 + pc5 + pc6 + pc7 + pc8 + pc9 +
   pc10, data = data, na.action = na.exclude)
m <- step(m, scope=list(upper = ~ age + sex + pc1 + pc2 + pc3 + pc4 + pc5 + pc6 + pc7 + pc8 + pc9 +
      pc10, lower = ~ age + sex))
summary(m)
data$hai_rg <- residuals(m)

m <- lm(hai_m ~ age + sex + pc1 + pc2 + pc3 + pc4 + pc5 + pc6 + pc7 +
      pc8 + pc9 + pc10, data = data, na.action = na.exclude)
m <- step(m, scope=list(upper = ~ age + sex + pc1 + pc2 + pc3 + pc4 + pc5 + pc6 + pc7 + pc8 + pc9 +
      pc10, lower = ~ age + sex))
summary(m)
data$hai_m_rg <- residuals(m)

write.table(data[, c("subject", "hai_rg", "hai_m_rg")], "3d5d-hai-gwas.csv",
  sep = ",", row = FALSE, quote = FALSE)
```

Telephone Interview for Cognitive Status (TICS) Dataset

This data set is part of the Cognitive Assessments. It was used when an in-person visit was not feasible. This dataset has no derived variables.

Venipuncture (VENIP) Dataset

The Venipuncture dataset contains information about the blood collection, any bleeding disorders, and the shipment information about the tubes. Derived variables were added as well.

_FASTTIME, _FAST, FASTING_LIPID and FASTING_CBC

_FASTTIME is the fasting hours. _FAST is the fasting status.

```
if daylast in("1","T") then daylast2=0;
        else if daylast in("2","Y") then daylast2=1;
if daylast in("2","Y") and lasttime=. and drawtime^=. then do;
      lasttime=.; end;
        else lasttime2=lasttime;

     datelast=DHMS(date-daylast2,hour(lasttime2),minute(lasttime),0);
     visittime=DHMS(date,hour(drawtime),minute(drawtime),0);
     _FASTTIME=round(((visittime-datelast)/3600),.01);

     if _FASTTIME < 8 then _FAST=0;
     if 8<=_FASTTIME<24 then _FAST=1;
     if _FASTTIME >=24 then _FAST=2;
     drop daylast2 lasttime2 datelast visittime;
     label _FASTTIME="Fasting hours"
           _FAST="Fasting status 0:<8hrs; 1:8-24hrs; 2:>=24hrs";
```

April 2018 update to venipall data
FASTING_LIPID and FASTING_CBC

In visit 2, some participants completed more than one venip form. Reasons were that the blood did not process correctly, or at one visit only saliva was collected and then they went back and collected blood for assays. Due to the potential for multiple forms, some participants have more than one row of data in venipall for visit 2, and we needed to create two new variables for these multiple blood draws to assess fasting for the assays. Thus the variables ***fasting_cbc*** and ***fasting_lipid*** are created to indicate which row of data for visit 2 in the venipall dataset was used for the final values of lipids and cbc and to assess if these values were fasting more than 8 hours or not.

How are the ***fasting_cbc*** and ***fasting_lipid*** created?
Two participants' data from venipall listed in the table below serve as examples. Each participant at visit 2 completed venip form twice, so there are two records for visit 2 in venipall. Reasons for more than one venip form are listed in the right-most column. LLFS Study Design specifies that, with the collected tube #2, i.e. SST tube, the laboratory tests of creatinine, glucose and lipid panel are performed. Also, with the collected tube #3, i.e. EDTA tube, the tests of CBC/diff/platelet and glycosylated hemoglobin are performed. Thus, regardless of value of _FAST, if tube #2 is not collected, i.e. tube2 = 0, then zero is assigned to fasting_lipid. Similarly if tube #3 is not collected, zero is assigned to fasting_cbc.

(1) Participant 6010 (subject ID): The second venip form for visit 2 was administered on June 6, 2016 for blood samples drawn. Because the fasting hour is 3.33, less than 8 hours, the values of fasting_cbc and fasting_lipid are "0". Although the first venip record for visit 2 shows that fasting hour of 16.5, "0" is assigned to fasting_cbc and fasting_lipid because there is only saliva was collected for the first venip.

(2) Participant 22199 (subject ID): Because tube #2 of first draw was not centrifuged, tube #2 was redrawn on March 8, 2016. Laboratory tests of lipid panel were performed on this sample, also the fasting hour is greater than 8 for this redrawn, thus "1" is assigned to fasting_lipid. As the first tube #2 was not used for tests of lipid panel, therefore fasting_lipid is "0" in the first venip

form. Laboratory tests of CBC were performed on tube #3 of the first venip, thus "1" is assigned to fasting_cbc in the first record for visit 2.

subject	visitcode	date	tube1	tube2	tube3	tube4	tube5	tube6	tube7	orag	_FASTTIME	_FAST	fasting_lipid	fasting_cbc	Reason for Duplicate Records in Visit 2
6010	1	8-Jan-08	1	1	1	1	1	1		0	14.95	1			
6010	3	25-Apr-16	0	0	0	0	0	0	0	1	16.5	1	0	0	Only Saliva was collected at visit 2 in home visit so participant went back for blood draws.
6010	3	6-Jun-16	1	1	1	1	1	0	0		3.33	0	0	0	
22199	1	5-May-09	1	1	1	1	1	1			10.82	1			
22199	3	15-Feb-16	1	1	1	1	1	1	1		13.42	1	0	1	The SST tube (#2) was not centrifuged: an unsatisfactory specimen.
22199	3	8-Mar-16	0	1	0	0	0	1	1		23	1	1	0	Participant went back for re-draw.

Please note the variables fasting_cbc and fasting_lipid are created for visit 2, so for visit 1 the variable _fast is still used to assess for fasting for visit 1. If you have any questions about how to use this data, please contact the coordinating center.

What Data Collected per Participant (WHATDATA) Dataset

This data set is an inventory of all the forms and reading center data that were collected for each Participant. This is for visit 1 and visit 2.
This dataset has no derived variables.

PHASE II

Follow Up (FOLLOWUP) Dataset

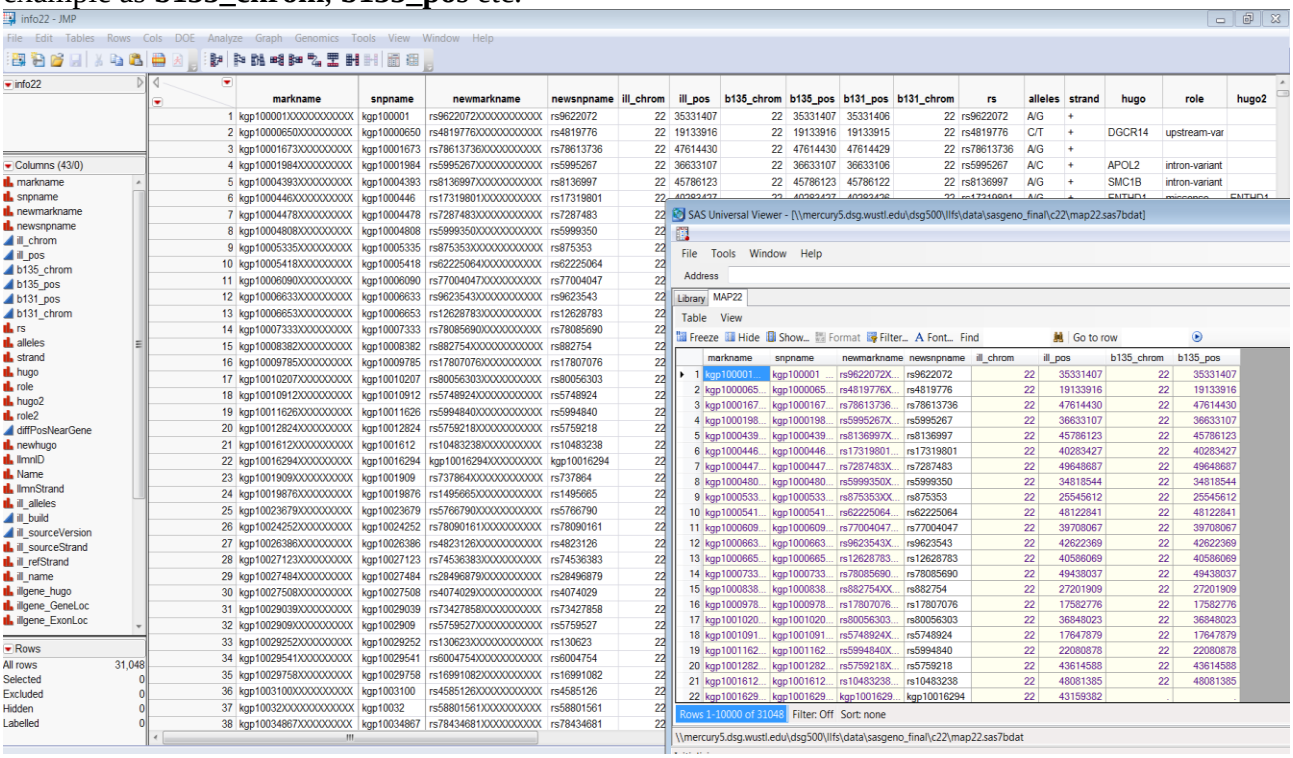
This dataset contains the information from the Follow Up form. It is completed at every follow up occasion. This dataset has no derived variables.

GENOTYPES

Annotation (Info and Map) Datasets

The first version of the annotation file was created based on the gene annotation file Illumina provided. The file corresponding to the chip used in genotyping, namely Human Omni 2_5-8v1 version was used. A large number of markers corresponded with a "kgp" id which was annotating that a particular marker was originated from 1000 Human Genome project. At a later time, an annotation file was provided by CIDR that delivered a correspondence among the "kgp" snps that lacked rs names with now matching rs-names. Therefore, a new annotation is created by merging these new rs names to the existing data. Two file type data are distributed: a) a map file per chromosome; and b) an info file, which have been merged with the NCBI b135 version of dbSNP database to find the most recent version of gene and position annotation. To have a continuation and match with the genotype markers names provided originally by Illumina, the snp names have been kept as the snp identifiers in our database, the improved annotation information by SNP can all be found in the info files for each chromosome.

NOTE: The most important connectors in the data are: **markname** in the map, which is a 20 character long identifier for the SNPs and matched with the genotype, genefreq and info files. The corresponding variable with no padding "XXXXXX" is **snpname**. The next important variable is **newmarkname**, which represent a variable with the most rs-names known in this chip. The corresponding variable with no padding "XXXXXX" is **newsnpname**. The rest of the variables are annotations, such as chromosome, position, gene name, role of the SNP. It will be of interest that one uses the information that comes from the latest NCBI dbSNP annotated for example as **b135_chrom**, **b135_pos** etc.



The above is a picture combination of a view of map22.sas7bdat and info22.sas7bdat.
Following are two tables that summarize the work on markers for the maps and annotation files.

No	ilrs	ilkgp	other	total	new_rs	still_kgp	other	total	b135	b131	illonly	total	diff
1	57,135	126,907	30	**184,072**	168,169	15,873	30	**184,072**	167,408	109	16,162	**183,679**	393
2	55,980	138,146		**194,126**	178,112	16,014		**194,126**	177,299	118	16,249	**193,666**	460
3	45,960	117,712		**163,672**	150,012	13,660		**163,672**	149,541	97	13,872	**163,510**	162
4	39,198	113,647	1	**152,846**	139,835	13,010	1	**152,846**	139,260	86	13,200	**152,546**	300
5	40,847	104,603	3	**145,453**	133,109	12,341	3	**145,453**	132,748	90	12,547	**145,385**	68
6	54,996	99,673	17	**154,686**	142,373	12,296	17	**154,686**	141,543	102	12,725	**154,370**	316
7	36,782	92,253	37	**129,072**	118,479	10,556	37	**129,072**	117,926	72	10,813	**128,811**	261
8	35,929	89,586		**125,515**	115,631	9,884		**125,515**	115,345	90	10,026	**125,461**	54
9	31,739	71,268	4	**103,011**	95,458	7,549	4	**103,011**	95,078	57	7,676	**102,811**	200
10	37,889	81,519		**119,408**	109,694	9,714		**119,408**	109,364	65	9,901	**119,330**	78
11	35,123	80,970	2	**116,095**	106,430	9,663	2	**116,095**	106,099	91	9,816	**116,006**	89
12	34,478	78,219	25	**112,722**	103,177	9,520	25	**112,722**	102,687	92	9,732	**112,511**	211
13	27,034	56,447	2	**83,483**	76,633	6,848	2	**83,483**	76,327	43	7,077	**83,447**	36
14	22,510	54,000		**76,510**	70,360	6,150		**76,510**	70,166	50	6,253	**76,469**	41
15	21,078	51,211	5	**72,294**	66,467	5,822	5	**72,294**	66,291	39	5,916	**72,246**	48
16	21,821	54,787	2	**76,610**	70,874	5,734	2	**76,610**	70,669	35	5,838	**76,542**	68
17	19,260	47,125	2	**66,387**	61,111	5,274	2	**66,387**	60,915	39	5,387	**66,341**	46
18	21,117	47,435		**68,552**	63,465	5,087		**68,552**	63,325	40	5,163	**68,528**	24
19	13,853	33,880		**47,733**	43,767	3,966		**47,733**	43,557	40	4,088	**47,685**	48
20	17,802	38,739	1	**56,542**	52,731	3,810		**56,541**	52,581	25	3,885	**56,491**	51
21	9,940	22,135		**32,075**	29,470	2,605		**32,075**	29,389	16	2,644	**32,049**	26
22	9,837	23,473		**33,310**	31,035	2,275		**33,310**	30,899	13	2,323	**33,235**	75
	690,308	**1,623,735**	**131**	**2,314,174**	**2,126,392**	**187,651**	**130**	**2,314,173**	**2,118,417**	**1,409**	**191,293**	**2,311,119**	**3,055**

Final count of markers included in the data follows:

chrom	LIBNAME	NAME	remobs	NOBS
1	C1	markname	379	**176,754**
2	C2	markname	444	**187,627**
3	C3	markname	154	**158,475**
4	C4	markname	291	**148,058**
5	C5	markname	66	**140,918**
6	C6	markname	288	**148,707**
7	C7	markname	251	**123,975**
8	C8	markname	50	**121,422**
9	C9	markname	188	**98,900**
10	C10	markname	75	**114,826**
11	C11	markname	85	**111,619**
12	C12	markname	198	**108,540**
13	C13	markname	36	**80,896**
14	C14	markname	39	**73,798**
15	C15	markname	44	**69,671**
16	C16	markname	60	**72,736**
17	C17	markname	46	**62,495**
18	C18	markname	24	**66,377**
19	C19	markname	46	**43,783**
20	C20	markname	46	**54,055**
21	C21	markname	23	**30,798**
22	C22	markname	70	**31,048**
Total			**2,903**	**2,225,478**

Info Datasets Variables

markname: Illumina provided locus name (rs number, if available) padded with "XXX" to 20 characters long

snpname: Illumina provided locus name (rs number, if available)

newmarkname: CIDR provided locus name (rs number, if available) padded with "XXX" to 20 characters long

newsnpname: CIDR provided locus name (rs number, if available)

ill_chrom: Illumina provided chromosome number

ill_pos: Illumina provided base pair position

b135_chrom: dbSNP Build 135 chromosome number

b135_pos: dbSNP Build 135 base pair position

b131_pos: dbSNP Build 131 base pair position

b131_chrom: dbSNP Build 131 chromosome number

rs: SNP rs number if found

alleles: dbSNP Build 135 alleles

strand: dbSNP Build 135 strand

hugo: dbSNP Build 135 gene symbol

role: dbSNP Build 135 SNP function class

ABBREV	DESCRIPTION
cds-synon	synonymous change. ex. rs248, GAG->GAA, both produce amino acid: Glu
intron	intron. ex. rs249.
cds-reference	contig reference
synonymy unknown	coding: synonymy unknown
nearGene-3	within 3' 0.5kb to a gene. ex. rs3916027 is at NT_030737.9 pos7669796, within 500 bp of UTR starts 7669698 for NM_000237.2.
nearGene-5	within 5' 2kb to a gene. ex. rs7641128 is at NT_030737.9 pos7641128, with 2K bp of UTR starts 7641510 for NM_000237.2.
STOP-GAIN	changes to STOP codon. ex. rs328, TCA->TGA, Ser to terminator.
missense	alters codon to make an altered amino acid in protein product. ex. rs300, ACT->GCT, Thr->Ala.
STOP-LOSS	changes STOP codon to other non-stop codon
frameshift	indel snp causing frameshift.
cds-indel	indel snp with length of multiple of 3bp, not causing frameshift.
UTR-3	3 prime untranslated region. ex. rs3289.
UTR-5	5 prime untranslated region. ex. rs1800590.
splice-3	3 prime acceptor dinucleotide. The last two bases in the 3 prime end of an intron. Most intron ends with AG.ex.rs193227 is in acceptor site.
splice-5	5 prime donor dinucleotide. 1st two bases in the 5 prime end of the intron. Most intron starts is GU. ex.rs8424 is in donor site.

hugo2: dbSNP Build 135 other strand overlapping gene symbol

role2: dbSNP Build 135 other strand overlapping SNP function class

diffPosNearGene: Distance (bp) to the nearest gene. = 0 if SNP on the gene, < 0, if with lower position (on upstream), > 0 if with higher position (on downstream)

Newhugo: The nearest gene name in ( ) if hugo is missing. If distance to the nearest gene (i.e. diffPosNearGene) > 5 kbp, postfix "_beyond" to gene name.

IlmnStrand: Illumina provided strand

ill_alleles: Illumina provided alleles

ill_build: Illumina provided build version

ill_sourceVersion: Illumina provided source version

ill_sourceStrand: Illumina provided source strand

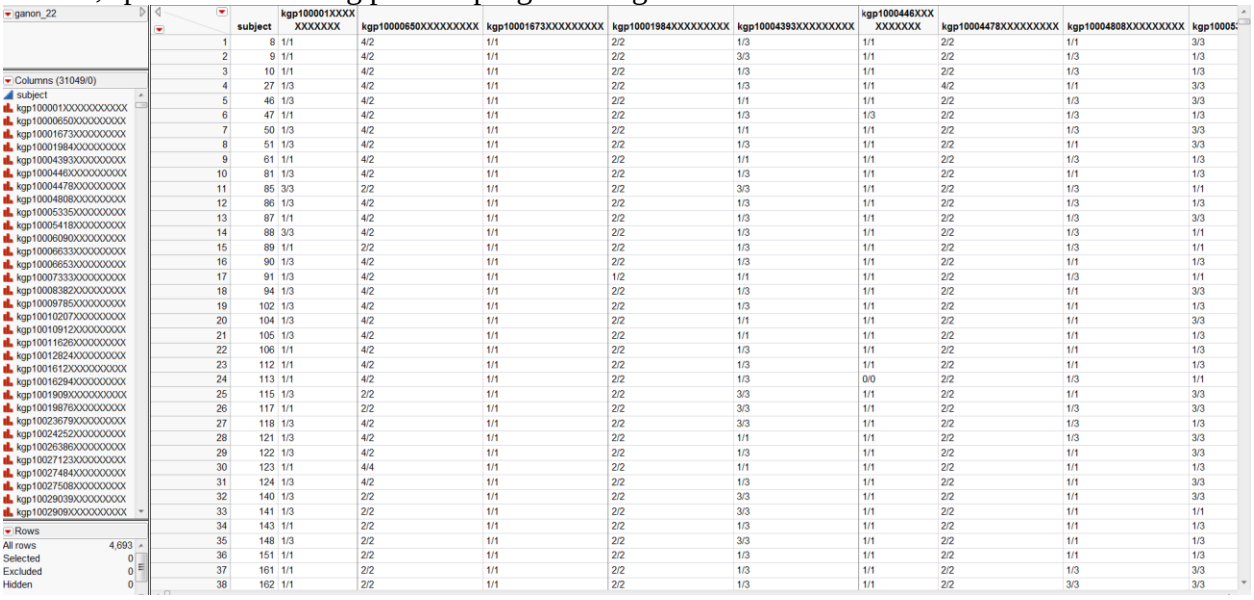
ill_refStrand: Illumina provided reference strand

illgene_hugo: Illumina provided gene symbol

illgene_GeneLoc: Illumina provided gene location
illgene_ExonLoc: Illumina provided exon location
illgene_CodingStatus: Illumina provided coding status
cidr_chrom: CIDR provided chromosome number
cidr_pos: CIDR provided base pair position
P_HWE: SNP Hardy Weinberg Equilibrium P Value
Callrate: SNP genotyping callrate
coded_all: GWAS coded allele
noncoded_all: GWAS other allele
coded_af: Allele frequency of the coded allele

Anonymous Genotypes (GANON) Datasets

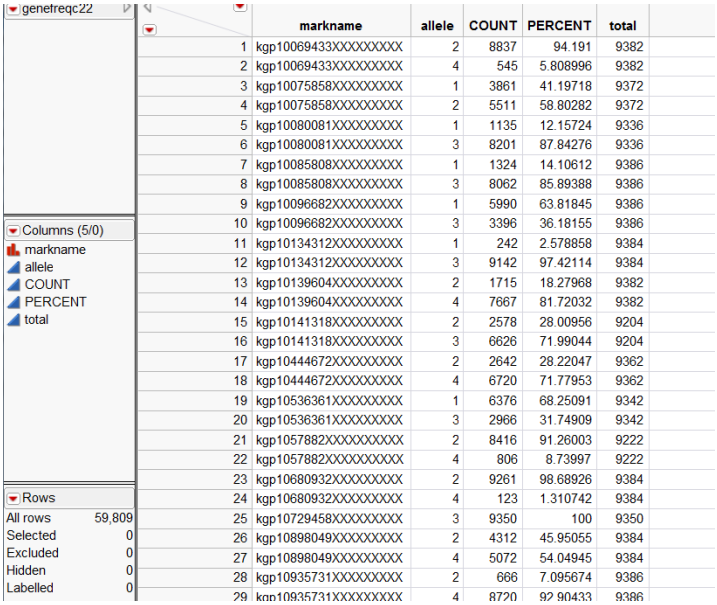
These represent the genotype datasets, organized by chromosome. There is one record per Subject and the columns represent the SNP markers. Each cell contains the genotype coded as allele1/allele2, in numerical representation where 1=A, 2=C, 3=G, and 4=T. These datasets are provided in two different formats, SAS and CSV. In addition, we provide CSV formatted subsets, split for conducting parallel programming.



	subject	kgp100001XXXX XXXXXXX	kgp10000650XXXXXXXX	kgp10001673XXXXXXXX	kgp10001984XXXXXXXX	kgp10004393XXXXXXXX	kgp1000446XXX XXXXXXX	kgp10004478XXXXXXXX	kgp10004808XXXXXXXX	kgp10005
1	8	1/1	4/2	1/1	2/2	1/3	1/1	2/2	1/1	3/3
2	9	1/1	4/2	1/1	2/2	3/3	1/1	2/2	1/3	1/3
3	10	1/1	4/2	1/1	2/2	1/3	1/1	2/2	1/3	1/3
4	27	1/3	4/2	1/1	2/2	1/3	1/1	4/2	1/1	3/3
5	46	1/3	4/2	1/1	2/2	1/1	1/1	2/2	1/3	3/3
6	47	1/1	4/2	1/1	2/2	1/3	1/3	2/2	1/3	1/3
7	50	1/3	4/2	1/1	2/2	1/1	1/1	2/2	1/3	3/3
8	51	1/3	4/2	1/1	2/2	1/3	1/1	2/2	1/1	3/3
9	61	1/1	4/2	1/1	2/2	1/1	1/1	2/2	1/3	1/3
10	81	1/3	4/2	1/1	2/2	1/3	1/1	2/2	1/1	1/3
11	85	3/3	2/2	1/1	2/2	3/3	1/1	2/2	1/3	1/1
12	86	1/3	4/2	1/1	2/2	1/3	1/1	2/2	1/3	1/3
13	87	1/1	4/2	1/1	2/2	1/3	1/1	2/2	1/3	3/3
14	88	3/3	4/2	1/1	2/2	1/3	1/1	2/2	1/3	1/1
15	89	1/1	2/2	1/1	2/2	1/3	1/1	2/2	1/3	1/1
16	90	1/3	4/2	1/1	2/2	1/3	1/1	2/2	1/1	1/3
17	91	1/3	4/2	1/1	1/2	1/1	1/1	2/2	1/3	1/1
18	94	1/3	4/2	1/1	2/2	1/3	1/1	2/2	1/1	3/3
19	102	1/3	4/2	1/1	2/2	1/3	1/1	2/2	1/1	1/3
20	104	1/3	4/2	1/1	2/2	1/1	1/1	2/2	1/1	3/3
21	105	1/3	4/2	1/1	2/2	1/1	1/1	2/2	1/1	1/3
22	106	1/1	4/2	1/1	2/2	1/3	1/1	2/2	1/1	1/3
23	112	1/1	4/2	1/1	2/2	1/3	1/1	2/2	1/1	1/3
24	113	1/1	4/2	1/1	2/2	1/3	0/0	2/2	1/3	1/1
25	115	1/3	2/2	1/1	2/2	3/3	1/1	2/2	1/1	3/3
26	117	1/1	2/2	1/1	2/2	3/3	1/1	2/2	1/3	3/3
27	118	1/3	4/2	1/1	2/2	3/3	1/1	2/2	1/3	1/3
28	121	1/3	4/2	1/1	2/2	1/1	1/1	2/2	1/3	3/3
29	122	1/3	4/2	1/1	2/2	1/3	1/1	2/2	1/1	3/3
30	123	1/1	4/4	1/1	2/2	1/1	1/1	2/2	1/1	1/3
31	124	1/3	4/2	1/1	2/2	1/3	1/1	2/2	1/1	3/3
32	140	1/3	2/2	1/1	2/2	3/3	1/1	2/2	1/1	3/3
33	141	1/3	2/2	1/1	2/2	3/3	1/1	2/2	1/1	1/1
34	143	1/1	2/2	1/1	2/2	1/3	1/1	2/2	1/1	1/3
35	148	1/3	2/2	1/1	2/2	3/3	1/1	2/2	1/1	1/3
36	151	1/1	2/2	1/1	2/2	1/3	1/1	2/2	1/1	1/3
37	161	1/1	2/2	1/1	2/2	1/3	1/1	2/2	1/3	3/3
38	162	1/1	2/2	1/1	2/2	1/3	1/1	2/2	3/3	3/3

Gene Frequency (GENEFREQ) Datasets

These sets of data (one per chromosome) contain the marker names (MARKNAME), numeric representations of the alleles (ALLELES), the frequency of each allele for each marker in the sample (PERCENT), expressed as a percent, and the number of subjects that carried that allele (COUNT) and total of counts. These data sets have 1 record per allele, usually two per marker. In case a marker is nonpolymorphic, then one will see 1 allele with 100 as its percentage.



genefreqc22

Columns (5/0)
markname
allele
COUNT
PERCENT
total

Rows
All rows 59,809
Selected 0
Excluded 0
Hidden 0
Labelled 0

	markname	allele	COUNT	PERCENT	total
1	kgp10069433XXXXXXXXX	2	8837	94.191	9382
2	kgp10069433XXXXXXXXX	4	545	5.808996	9382
3	kgp10075858XXXXXXXXX	1	3861	41.19718	9372
4	kgp10075858XXXXXXXXX	2	5511	58.80282	9372
5	kgp10080081XXXXXXXXX	1	1135	12.15724	9336
6	kgp10080081XXXXXXXXX	3	8201	87.84276	9336
7	kgp10085808XXXXXXXXX	1	1324	14.10612	9386
8	kgp10085808XXXXXXXXX	3	8062	85.89388	9386
9	kgp10096682XXXXXXXXX	1	5990	63.81845	9386
10	kgp10096682XXXXXXXXX	3	3396	36.18155	9386
11	kgp10134312XXXXXXXXX	1	242	2.578858	9384
12	kgp10134312XXXXXXXXX	3	9142	97.42114	9384
13	kgp10139604XXXXXXXXX	2	1715	18.27968	9382
14	kgp10139604XXXXXXXXX	4	7667	81.72032	9382
15	kgp10141318XXXXXXXXX	2	2578	28.00956	9204
16	kgp10141318XXXXXXXXX	3	6626	71.99044	9204
17	kgp10444672XXXXXXXXX	2	2642	28.22047	9362
18	kgp10444672XXXXXXXXX	4	6720	71.77953	9362
19	kgp10536361XXXXXXXXX	1	6376	68.25091	9342
20	kgp10536361XXXXXXXXX	3	2966	31.74909	9342
21	kgp1057882XXXXXXXXXX	2	8416	91.26003	9222
22	kgp1057882XXXXXXXXXX	4	806	8.73997	9222
23	kgp10680932XXXXXXXXX	2	9261	98.68926	9384
24	kgp10680932XXXXXXXXX	4	123	1.310742	9384
25	kgp10729458XXXXXXXXX	3	9350	100	9350
26	kgp10898049XXXXXXXXX	2	4312	45.95055	9384
27	kgp10898049XXXXXXXXX	4	5072	54.04945	9384
28	kgp10935731XXXXXXXXX	2	666	7.095674	9386
29	kgp10935731XXXXXXXXX	4	8720	92.90433	9386

GTRIPLET and TRIPLET_visit2 Dataset

The GTRIPLET dataset reflects the pedigree structures corrected using genetic information and GRR (Graphical Representation of Relationships). Therefore, it is the preferred pedigree structure for analysis. A triplet is the person, his/her mother, and his/her father. This is all the necessary information needed to determine relatedness.

SUBJECT: the de-identified, unique identifier for each Participant. It is a 5 digit number. Datasets with one obs/subject are uniquely identified by ID and can be merged/linked using this variable.
MOMSUBJ: the SUBJECT number of the person's mother
DADSUBJ: the SUBJECT number of the person's father
Proband_status: the index indicates a subject who is the proband in a pedigree.
gpedid: is the preferred indicator of family membership since it is derived using genetic information.
deceased: vital status.
twinrelatn: twin relationship, MZ or DZ.
relative: the Subject is genetically related to the proband.
control: the Subject is married into the Proband's offspring generation. This person married an offspring of the Proband.
gen: Generation Number.

1 = Proband's Parents Generation
2 = Proband's Generation
3 = Proband's Offspring Generation
4 = Proband's Grandchildren Generation

In the pedigree plot (in the figure on the next page), the color code reflects the values of relative and control. The diagonal line indicates a deceased Subject. The black arrow points to the Proband. Circles are females, squares are males, and diamonds are dummy, placeholder children, to indicate the relationship of a spouse pair without biological children between them.