Atherosclerosis Risk in Communities Study

Cohort Exam Visit 10 NCS

DERIVE101_NP Derived Variable Dictionary (v1.0)

November 2024

Prepared by the Collaborative Studies Coordinating Center

ARIC DERIVE101_NP Derived Variable Dictionary

Table of Contents

NEW OR CHANGED FROM PREVIOUS DISTRIBUTION

This table describes the changes to the last published DERIVE101 dictionary. As the dataset undergoes modifications, this table will describe the updates made to the previously distributed dataset.

Modification Date	Variable Name	Reason(s) for Change

1. OVERVIEW

The DERIVE101_250219_NP dataset has 1645 records, one for each participant who completed Stage 1 at Visit 10. The purpose of this dataset is to provide ARIC collaborators widely used, verified derived variables, many of which are consistent with variables derived at prior visits.

The dataset naming conventions are as follows: The dataset name retains the retrieval date (ex: DERIVE101_250219_NP) until the dataset is considered final, frozen. After a dataset is frozen, the retrieval date is dropped from the dataset name (ex: DERIVE101_NP). The first digit in the dataset name refers to the visit number. The second digit in the dataset name is incremented in number when the current dataset undergoes significant changes. Datasets with “NP” in the name have date variables removed; instead, date variables are replaced with a variable that calculates the number of follow up days after Visit 1 date and the year of the original date variable. The variable naming convention is similar: Across-visit variables have identical names except for the second to last digit in the variable name, which represents the visit number (ex: GENDER101 at Visit 10 vs. GENDER91 at Visit 9). The last digit in the variable name identifies the definition version of a variable.

Most of the variables are derived directly from the data collected at the visit. However, some variables use ARIC cohort surveillance and ARIC follow-up data in their definitions. DERIVE101 will be final, frozen after the surveillance datasets are complete for events in 2023.

Laboratory variables from Visit 10 are being distributed in V1_V1_LongLab dataset and neurocognitive variables are being distributed in STATUS101 dataset.

2. ADMINISTRATIVE

2.1 SUBJECTID (ARIC Subject ID (CIR))

Type: Character; length: $7.

2.2 ID (ARIC ID - same as SUBJECTID)

Description: The historical participant identifier from visits 1-4 is ID. The value of ID is the same value as SUBJECTID. Use ID when merging Visit 10/NCS stage 1 data with datasets from previous visits necessary for longitudinal analyses.

Type: Character; length: $7.

Algorithm: ID=SUBJECTID

Source variable(s): SUBJECTID

2.3 CENTER (Field Center)

Description: Character variable with four possible values derived from the enrollment site:
F: Forsyth County, North Carolina
J: The city of Jackson, Mississippi
M: Selected northwestern suburbs of Minneapolis, Minnesota
W: Washington County, Maryland

Type: Character; length: $1.

Algorithm: CENTER = First letter of the subject ID

Source variable(s): SUBJECTID

2.4 V10CENTER (Visit 10 Field Center)

Description: Character variable with four possible values derived from the enrollment site:
F: Forsyth County, North Carolina
J: The city of Jackson, Mississippi
M: Selected northwestern suburbs of Minneapolis, Minnesota
W: Washington County, Maryland

Type: Character; length: $1.

Algorithm: The value of V10CENTER is the same as CENTER unless the ARIC study participant has relocated geographically and moved into another field center at Visit 10. In that instance, the value of V10CENTER is the value of the field center where the participant was seen.

2.5 V10DATE101_FollowUpDays (Days of follow up from visit 1 to Visit 10 Date, NCS Stage 1)

Description: The number of days between visit 1 and the date of the participant's Visit 10 exam, stage 1. The Visit 10 exam date is taken from the first present form in V10 / NCS in the following order: SBP, BIO9, ANT.

Type: Numeric

Algorithm: If the eventname from the SBP form is "V10 / NCS" then V10DATE101_FollowUpDays=SBP0a - visit 1 date;

Else if the eventname from the BIO9 form is "V10 / NCS" then V10DATE101_FollowUpDays=BIO0a - visit 1 date;

Else if the eventname from the ANT form is "V10 / NCS" then V10DATE101_FollowUpDays=ANT0a - visit 1 date;

Source variable(s): STAGE_1_COMPLETE, ANT0a, BIO0a, SBP0a, visit 1 date

2.6 V10DATE101_year (Year of Visit 10 date)

Description: Year of the participant's Visit 10 exam, stage 1. The Visit 10 exam date is taken from the first present form in V10 / NCS in the following order: SBP, BIO9, ANT.

Type: Numeric

Algorithm: If the eventname from the SBP form is "V10 / NCS" then V10DATE101_year=year of SBP0a;

Else if the eventname from the BIO9 form is "V10 / NCS" then V10DATE101_year=year of BIO0a;

Else if the eventname from the ANT form is "V10 / NCS" then V10DATE101_year=year of ANT0a;\

Source variable(s): STAGE_1_COMPLETE, ANT0a, BIO0a, SBP0a

2.7 RES_OTH (Restrictions on Other Procedures)

Description: The derived informed consent file, ICTDER05, includes information and dates of final consents for ARIC participants. Change in consent status is recorded in ARIC follow-up and Visit 10 using the ICT form. RES_OTH indicates the types of restriction on other procedures. We request that the investigators exclude appropriate records with partial restrictions prior to data analysis.

Type: Character; length: $130.

Algorithm: if ICT3=1 and ICT4=1 and ICT7=1 then res_OTH='Full Consent';
else if ICT3=1 and ICT4=1 and (ICT7=0 or missing(ICT7)) then res_OTH='Not for Profit';
else if ICT3=1 and ICT4=0 then res_OTH='ARIC Only';
else if ICT3=0 and ICT4=0 then res_OTH='No Consent';

Source variable(s): ICT3, ICT4, ICT7

2.8 RES_DNA (Restrictions on DNA Usage)

Description: The derived informed consent file, ICTDER05, includes information and dates of final consents for ARIC participants. Change in consent status is recorded in ARIC follow-up and Visit 10 using the ICT form. The variable RES_DNA indicates the type of restriction on DNA use. We request that the investigators exclude appropriate records with partial restrictions prior to data analysis.

Type: Character; length: $130.

Algorithm: if ICT5=1 and ICT6=1 and ICT7=1 then RES_DNA='Full Consent';
else if ICT5=1 and ICT6=1 and (ICT7=0 or missing(ICT7)) then RES_DNA='Not for Profit';
else if ICT5=1 and ICT6=0 then RES_DNA='ARIC Only';
else if ICT5=0 and ICT6=0 then RES_DNA='No use/storage DNA';

Source variable(s): ICT5, ICT6, ICT7

2.9 LASTFUINTERVIEW_DAT_FollowUpDays (Days of follow up from visit 1 to Date of last completed follow-up interview)

Description: The number of days between visit 1 and the date of the participant's last completed follow-up interview where an actual contact was made, prior to December 20, 2023.

Type: Numeric

Algorithm: Days between Visit 1 and the max value of AFUcomp1_A in the composite follow-up dataset among the records for a single ID where AFUcomp2_A indicates that the interview was accomplished (AFUcomp2_A in ('A','C','D')) and the date preceded December 20, 2023.

Source variable(s): AFUcomp1_A, AFUcomp2_A, visit 1

2.10 LASTFUINTERVIEW_DAT_year (Year of Date of last completed follow-up interview)

Description: Year of the participant's last completed follow-up interview where an actual contact was made, prior to December 20, 2023.

Type: Numeric

Algorithm: Year of the max value of AFUcomp1_A in the composite follow-up dataset among the records for a single ID where AFUcomp2_A indicates that the interview was accomplished (AFUcomp2_A in ('A','C','D')) and the date preceded December 20, 2023.

Source variable(s): AFUcomp1_A, AFUcomp2_A

2.11 STAGE_1_TYPE (Type of Stage 1 Exam)

Description: Categorical variable that describes the participant's type of Stage 1 exam.

Format: A=Full, B=Abbreviated, C=Home, D=Long Term Care Facility.

Type: Character; length=$1.

Algorithm: STAGE_1_TYPE=RTS17

Source variable(s): RTS17

2.12 V10PROXYATEXAM101 (Proxy present at V10)

Description: Proxy present at V10

Format: Y=Yes, N=No

Type: Character; length=$1.

Algorithm: V10PROXYATEXAM101=RTS18

Source variable(s): RTS18

2.12 V10PROXYDATA101 (Proxy contributed to data collection at V10)

Description: Proxy contributed to data collection at V10

Format: Y=Yes, N=No

Type: Character; length=$1.

Algorithm: If RTS18="Y" and RTS18a="Y" then V10PROXYDATA101="Y";
If RTS18="N" or RTS18a= "N" then V10PROXYDATA101="N";
Else V10PROXYDATA101= missing;

Source variable(s): RTS18, RTS18a

3. SOCIO-DEMOGRAPHIC

3.1 GENDER (Sex)

Description: Categorical variable that describes the participant's gender: M=Male, F=Female.

Type: Character; length=$1.

Algorithm: GENDER = GENDER from DERIVE13

Source variable(s): [DERIVE13] GENDER

3.2 GENDER101 (Corrected Gender (V1CORGE1))

Description: Categorical variable that describes the participant's gender: M=Male, F=Female. Incorrect values for the variable GENDER were identified following the initial data collection on the ARIC cohort. The ARIC Executive Committee has recommended

continuing to use the uncorrected variable (GENDER) for Visit 1 and longitudinal analyses. The corrected version could be used for cross-sectional analyses other than Visit 1 and should be decided by the Investigator.

Type: Character; length=$1.

Algorithm: GENDER101 = V1CORGE1

Source variable(s): V1CORGE1

3.3 RACEGRP (Race)

Description: Categorical variable which describes the participant's race: A=Asian, B=Black, I=Native American, W=White.

Type: Character; length=$1.

Algorithm: RACEGRP = RACEGRP from DERIVE13

Source variable(s): [DERIVE13] RACEGRP

3.4 RACEGRP101 (Corrected Race (V1CORRA1))

Description: Categorical variable which describes the participant's race: A=Asian, B=Black, I=Native American, W=White. Incorrect values for the variable RACEGRP were identified following the initial data collection on the ARIC cohort. The ARIC Executive Committee has recommended continuing to use the uncorrected variable (RACEGRP) for Visit 1 and longitudinal analyses. The corrected version could be used for cross-sectional analyses other than Visit 1 and should be decided by the Investigator.

Type: Character; length=$1.

Algorithm: RACEGRP101 = V1CORRA1

Source variable(s): V1CORRA1

3.5 V10AGE101 (Visit 10 Age)

Description: Participant's age at the time of the Visit 10 exam calculated from the date of birth originally reported for the participant.

Type: Numeric

Algorithm: If Visit 10 exam occurred and date of birth>.z then V10AGE101 = floor((intck('month', date of birth, Visit 10 date)-(day(Visit 10 date) < day(date of birth)))/12);

Source variable(s): Date of birth reported at visit 1, Visit 10 date

3.6 V10AGE102 (Corrected Visit 10 Age)

Description: Participant's age at the time of the Visit 10 exam calculated from the corrected date of birth. The corrected date of birth corrects known errors in the date of birth values reported in the initial data collection on the ARIC cohort. The ARIC Executive Committee has recommended continuing to use the uncorrected variable (V10AGE101) for Visit 1 and longitudinal analyses. The corrected version could be used for cross-sectional analyses other than Visit 1 and should be decided by the Investigator.

Type: Numeric

Algorithm: If Visit 10 exam occurred and corrected date of birth>.z then V10AGE102=floor((intck('month', corrected date of birth, Visit 10 date)-(day(Visit 10 date) < day(corrected date of birth)))/12);

Source variable(s): Corrected date of birth, Visit 10 date

4. ANTHROPOMETRY AND BLOOD PRESSURE

4.1 BMI101 (V10 Body Mass Index in Kg/m^2)

Description: Body mass index [Weight (kg)] / [Height (cm) / 100]^2

Type: Numeric

Algorithm: If missing (V6IN129) or missing (ANT4) then BMI101=missing;
Else BMI101= ANT4/(V6IN129/100)^2

Source variable(s): V6IN129 (Last Measured Height in cm), ANT4

4.2 WSTHPR101 (V10 Waist-to-Hip Ratio)

Description: Ratio of waist girth to hip girth

Type: Numeric

Algorithm: If missing(ANT10a) or missing(ANT10b) or ANT10b=0 then
WSTHPR101=missing
Else WSTHPR101=(ANT10a/ANT10b)

Source variable(s): ANT10a, ANT10b

4.3 SYSTOLIC101 (V10 Mean Systolic BP of 2nd and 3rd Measurements)

Description: Mean of 2nd and 3rd systolic blood pressure measurements, consistent with V1 through V5

Type: Numeric

Algorithm: SYSTOLIC101=mean of SBP8 and SBP11
If SYSTOLIC101=missing then SYSTOLIC101=SBP14

Source variable(s): SBP8, SBP11, SBP14

4.4 DIASTOLIC101 (V10 Mean Diastolic BP of 2nd and 3rd Measurements)

Description: Mean of 2nd and 3rd diastolic blood pressure measurements, consistent with V1 through V5

Type: Numeric

Algorithm: DIASTOLIC101=mean of SBP9 and SBP12
If DIASTOLIC101=missing then DIASTOLIC101=SBP15

Source variable(s): SBP9, SBP12, SBP15

4.5 PULSE101 (V10 Mean Pulse of 2nd and 3rd Measurements)

Description: Mean of 2nd and 3rd pulse measurements, consistent with V1 through V5

Type: Numeric

Algorithm: PULSE101=mean of SBP10 and SBP13
If PULSE101=missing then PULSE101=SBP16

Source variable(s): SBP10, SBP13, SBP16

5. DISEASE PREVALENCE

There are a number of variables in the Disease Prevalence section that utilize cohort surveillance information, through event year 2023, to derive disease prevalence. All derived variables using surveillance data are considered final when the event year is closed. Versions 3 - 5 for diabetes prevalence are not created in DERIVE101.

Diabetes Prevalence Variables

Temporary variables were created for use in the algorithms of the following diabetes-derived variables. They are not found in the DERIVE101 dataset.

Temporary MDDX_DIAB101

Algorithm: MDDX_DIAB101=1 if a participant reported being diagnosed during ARIC Follow-Up interviews with diabetes prior to the Visit 10 date. The composite dataset variables considered are AFUCOMP1_A, AFUCOMP7D_G, and AFUCOMP15_M.

Temporary INCSELFREPDM101

Algorithm: If any of the records for a single ID have a 'Y' value for either AFUcomp7d_G or AFUcomp15_M and
.z<afucomp1_A<="20DEC2023"d
then INCSELFREPDM101 = 1;

Else if AFUcomp7d_G, AFUcomp15_M are (N,'') or ('',N) respectively in all records for a single ID, where
.z<afucomp1_A<="20DEC2023"d
then INCSELFREPDM101 = 0;

Else INCSELFREPDM1 = .T

5.1 DIABTS106 (V10 Diabetes – Hemogoblin A1C, cutpoint 6.5%)

Description: Diabetes variable defined as present if hemoglobin A1C value>=6.5 % or using medication for diabetes or self-report diagnosis of diabetes.

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If (CHEM1>=6.5) or (MSR2 ne 'T' and MSR33c = 'Y') or MDDX_DIAB101=1 then DIABTS106=1;

Else if (.z< CHEM1<6.5) and MDDX_DIAB101 ne 1 and MSR33c ne 'Y' then DIABTS106 =0;

Else DIABTS106=.T ;

Source variable(s): CHEM1, MDDX_DIAB101, MSR2, MSR33c

5.2 DIABTS107 (V10 Diabetes (DM medications or DM reported on AFU))

Description: Diabetes variable defined as present if participant is using medication for diabetes or self-report diagnosis of diabetes.

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If (MSR2 ^= 'T' and MSR33c='Y') or INCSELFREPDM101=1 or DIABMDCODE101=1 then DIABTS107=1

Else if INCSELFREPDM101 ^= 1 and MSR33c ^= 'Y' and DIABMDCODE101 ^= 1 then DIABTS107=0

Else DIABTS107=.T

Source variable(s): MSR2, MSR33c, INCSELFREPDM101, DIABMDCODE101

Myocardial Infarction (MI) Prevalence Variables

5.3 MDDXMI102 (V10 MD Diagnosed Myocardial Infarction)

Description: This variable is derived from ARIC Follow-Up questions that ask if the participant was told by a doctor whether they had a heart attack between visit 9 and visit 10.

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If (AFUCOMP7a_G = 'Y' OR AFUCOMP11a_M = 'Y') AND V9DATE91 < AFUCOMP1_A (AFU DATE) < V10DATE101 then MDDXMI102=1

Else if all values of (AFUCOMP7a_G, AFUCOMP11a_M) for records between visit 9 date and visit 10 date are one of the following combinations ('',U), ('',''), (U,'') then MDDXMI102=.T

Else MDDXMI102=0

Source variable(s): AFUCOMP7a_G, AFUCOMP11a_M, V7DATE71, V10DATE101, AFUCOMP1_A

5.4 HXOFMI102 (V10 History of Myocardial Infarction)

Description: This variable is derived from ARIC Follow-Up questions that ask if the participant was told by a doctor whether they had a heart attack as well as questions asking the participant if they'd been hospitalized for heart attack. The follow-up records from the ARIC Follow-Up composite dataset considered for this variable were collected before the end of V10 data collection (20DEC2023).

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If MDDXMI102=1 or (AFUcomp7_V1= 'Y') or (AFUcomp30_deceased_A='Y') or (AFUcomp30non_deceased_A='Y') then HXOFMI102=1

Else if over all records for a single ID the following value combinations are found for (MDDXMI102, AFUcomp7_V1, AFUcomp30_deceased_A, AFUcomp30non_deceased_A): (.T,'','',''), (.T,'','U',''), (.T,'','','U') then HXOFMI102=.T

Else HXOFMI102=0

Source variable(s): MDDXMI102, AFUCOMP7_V1, AFUCOMP30_deceased_A, AFUcomp30non_deceased_A

Coronary Heart Disease (CHD) Prevalence Variables

5.5 PRVCHD101 (V10 Prevalent CHD before Visit 10)

Description: This variable is derived from the baseline status of CHD (PRVCHD05) and the closed event years of ARIC Cohort Surveillance data through 2023, where the events occurred prior to the participant's Visit 10.

Format: 0=No,
1=Yes,
.=missing.

Type: Numeric

Algorithm: If PRVCHD05=1 or (C7_IN_21SP=1 and .<C7_DATEISP< V10DATE101) or (C7_IN_21SP=1 and V10DATE101=. and C7_DATEISP<="20DEC2023"d) then PRVCHD101=1

Else if PRVCHD05=0 and ((C7_IN_21SP=0 or C7_DATEISP>= V10DATE101>.) or (V10DATE101=. And C7_DATEISP>"20DEC2023"d)) then PRVCHD101=0

Else PRVCHD101=.T

Source variable(s): PRVCHD05, C7_IN_21SP, C7_DATEISP, V10DATE101

5.6 PRVCHD103 (V10 Prevalent CHD by end of Visit 10)

Description: This variable is derived from the baseline status of CHD (PRVCHD05) and the closed event years of ARIC Cohort Surveillance data through 2023, where the events occurred prior to the end of V10 data collection (20DEC2023).

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If PRVCHD05=1 or (C7_IN_21SP=1 and .<C7_DATEISP<="20DEC2023"d) then PRVCHD103=1

Else if PRVCHD05=0 and (C7_IN_21SP=0 or C7_DATEISP>"20DEC2023"d) then PRVCHD103=0

Else PRVCHD103=.T

Source variable(s): PRVCHD05, C7_IN_21SP, C7_DATEISP

5.7 PRVCHD104 (V10 Prevalent CHD - unverified)

Description: This variable is derived from self-reported ARIC Follow-Up data including questions on doctor told participant about heart attack, coronary bypass, and coronary angioplasty on records collected up through the end of V10 data collection (20DEC2023).

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: use AFUcomp: only for those records where .z < AFUcomp1_A <= "20DEC2023"d

If MDDXMI102=1 or (AFUcomp13a_G='Y') or (AFUcomp15a_G='Y') then PRVCHD104=1

Else if over all records for a single ID the following value combinations are found for (MDDXMI102, AFUcomp13a_G, AFUcomp15a_G):((.M or .), ('' or 'U'), ('' or 'U')) then PRVCHD104=.

Else PRVCHD104=0

Source variable(s): AFUcomp1_A, MDDXMI102, AFUcomp13a_G, AFUcomp15a_G

Stroke Prevalence Variables

5.8 PRVSTR101 (V10 Prevalent Stroke by the end of Visit 10)

Description: This variable is derived from the baseline status of stroke (HOM10D) and the closed event years of ARIC Surveillance data on the cohort through 2023, where the events occurred prior to the end of V10 data collection (20DEC2023).

Format: 0=No,

1=Yes,
.T=missing.

Type: Numeric

Algorithm: If HOM10D=1 or (C7_IN21DP=1 and .<C7_ED21DP<=V10DATE101) or (C7_IN21DP=1 and V10DATE101=. and C7_ED21DP<="20DEC2023") then PRVSTR101=1

Else if HOM10D=0 and ((C7_IN21DP=0 or C7_ED21DP>V10DATE101>.) or (V10DATE101=. and C7_ED21DP> "20DEC2023")) then PRVSTR101=0

Else PRVSTR101=.T

Source variable(s): HOM10D, C7_IN21DP, C7_ED21DP, V10DATE101

5.9 PRVSTR102 (V10 Prevalent Stroke-unverified)

Description: This variable is derived from self-reported ARIC Follow-Up data including questions that doctor told participant about stroke or TIA or participant hospitalized for stroke on records collected up through the end of V10 data collection (20DEC2023).

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If any record within a single ID has a "Y" value is found in either AFUcomp29_A or AFUcomp8b_K then PRVSTR102=1
Else if over all records within a single ID the following value combinations are found for (AFUcomp29_A, AFUcomp8b_K): ('',''), ('',U), (U,'') then PRVSTR102=.T
Else PRVSTR102=0

Source variable(s): AFUcomp29_A, AFUcomp8b_K

Hypertension Prevalence Variables

5.10 HYPERT104 (V10 HTN definition 4 (DIASTOLIC101 GE 90 or HTN med))

Description: Hypertension definition defined as diastolic blood pressure (mean of 2nd and 3rd measures) >=90 or medication is being taken for high blood pressure.

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If (DIASTOLIC101>= 90) or (MSR2 ne 'T' and MSR33d='Y') then HYPERT104=1;

Else if (0<DIASTOLIC101<90) and (MSR33d='N' or (MSR33d=missing and MSR2='T')) then HYPERT104=0;

Else HYPERT104=.T ;

Source variable(s): DIASTOLIC101, MSR2, MSR33d

5.11 HYPERT105 (V10 HTN definition 5 (SYSTOLIC101 GE 140 or DIASTOLIC101 GE 90 or HTN medication))

Description: Hypertension is defined as systolic blood pressure (mean of 2nd and 3rd measures) >=140 or diastolic blood pressure (mean of 2nd and 3rd measures) >=90 or medication is being taken for high blood pressure.

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If (DIASTOLIC101>= 90) or (SYSTOLIC101>= 140) or (MSR2 ne 'T' and MSR33d='Y') then HYPERT105=1

Else if (0<DIASTOLIC101<90) and (0<SYSTOLIC101<140) and (MSR33d='N' or (MSR33d=missing and MSR2='T')) then HYPERT105=0

Else HYPERT105=.T

Source variable(s): DIASTOLIC101, SYSTOLIC101, MSR2, MSR33d

5.12 HYPERT106 (V10 HTN definition 6 (SYSTOLIC101 GE 160 or DIASTOLIC101 GE 95 or HTN medication)

Description: Hypertension is defined as systolic blood pressure (mean of 2nd and 3rd measures) >=160 or diastolic blood pressure (mean of 2nd and 3rd measures) >=95 or medication is being taken for high blood pressure.

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If (DIASTOLIC101>= 95) or (SYSTOLIC101>=160) or (MSR2 ne 'T' and MSR33d='Y') then HYPERT106=1

Else if (0<DIASTOLIC101<95) and (0<SYSTOLIC101<160) and (MSR33d='N' or (MSR33d=missing and MSR2='T')) then HYPERT106=0

Else HYPERT106=.T

Source variable(s): DIASTOLIC101, SYSTOLIC101, MSR2, MSR33d

5.13 HYPERT107 (V10 HTN definition 7 (SYSTOLIC101 GE 150 or DIASTOLIC101 GE 90 or HTN medication)

Description: Hypertension is defined as systolic blood pressure (mean of 2nd and 3rd measures) >=150 or diastolic blood pressure (mean of 2nd and 3rd measures) >=90 or medication is being taken for high blood pressure.

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: If (DIASTOLIC101>=90) or (SYSTOLIC101>=150) or (MSR2 ne 'T' and MSR33d='Y') then HYPERT107 = 1

Else if (0<DIASTOLIC101<90) and (0<SYSTOLIC101<150) and (MSR33d='N' or (MSR33d=missing and MSR2='T')) then HYPERT107=0

Else HYPERT107=.T

Source variable(s): DIASTOLIC101, SYSTOLIC101, MSR2, MSR33d

Heart Failure (HF) Prevalence Variables

The heart failure (HF) variables included in this derived variable dictionary define HF prevalence at Visit 10. In general, the Definite HF variable will be most specific for prevalent HF, while the Definite or Possible HF variable will be the most sensitive. The appropriate variable for any given analysis will therefore depend on the balance between the sensitivity and specificity necessary for that analysis.

When conducting analyses of incident HF, it is important to note that the incident HF variable is available in the ARIC Cohort Surveillance dataset. The variable C7_INCHF21 uses V1 as the baseline and considers an event as a hospitalization with an ICD-9 discharge diagnosis code of 428.x in any position or corresponding ICD-10 code.

More detailed analysis recommendations from the ARIC Heart Failure Committee are available in the ARIC Analysis Manual (Manual 30).

The following definitions are temporary variables for use in the algorithms of the heart failure prevalence derived variables and are not included in DERIVE101.

Temporary HospHF_2005onward

Algorithm: For HFC21OCC1 records where
01JAN2005<=HFEVTDATE<V10DATE101
HospHF_2005onward=1, if CHFDIAG in ('A' 'B' 'C') for any record for SubjectID
HospHF_2005onward=0, otherwise

Temporary OnsetDate

Algorithm: Using data from the most recent PHF and PHFphi datasets and PHFA1104
If PHF2c is not missing, OnsetDate=PHF2c
Else if PHF7 is not missing, OnsetDate=PHF7
Else if PHFA2c is not missing, OnsetDate=PHFA2c

Else if PHFA7 is not missing, OnsetDate=PHFA7
Else if PHF0a is not missing, OnsetDate=PHF0a
Else if ENTRY_DA is not missing, OnsetDate=ENRTY_DA

Temporary HF_byPHF

Algorithm: Where missing<OnsetDate<V10DATE101
HF_byPHF=1, if phf1='Y' or phfa1='Y'
HF_byPHF=0, otherwise
Keep last record for each SubjectID only after sorted by SubjectID, HF_byPHF, and OnsetDate

Temporary HF_first428_pre2005

Algorithm: Using C21CELB1 records where missing<CELB04<01JAN2005
HF_first428_pre2005=1, if ICD code in first position (CELB10a) is 428.x for any record per SubjectID
HF_first428_pre2005=0, otherwise

Temporary HF_non1st428_pre2005

Algorithm: Using C21CELB1 records where missing<CELB04<01JAN2005
HF_non1st428_pre2005=1, if ICD code is 428.x anywhere but the first position (CELB10b-CELB10z) for any record for the SubjectID
HF_non1st428_pre2005=0, otherwise

Temporary srHF/srHFdt/Form

Algorithm: For [v3]phxa04, srHF=phxa8j, srHFdt=phxa63, Form=NULL
For [v4]phxb04, srHF=phxb6b, srHFdt=phxb21, Form=NULL
For afu&y1104new (y=g, h, i, j, k), srHF=afu&y7b, srHFdt=afu&y1, Form=NULL

For afu&y1104new (y=l, m),
srHF='Y' if afu&y7b, afu&y8, afu&y9, or afu&y10 is 'Y';
else srHF='N' if afu&y7, afu&y8, afu&y9, or afu&y10 is 'N'.
srHFdt=adu&y1 if not missing; else srHFdt=ENTRY_DA
Form=NULL

For afu_&mrt, &mrt=most recent data pull,
srHF='Y' if afu30 or afu31 is 'Y';
else srHF='N' if afu30 or afu31 is 'N'.
srHFdt=AFU0a if not missing;
else srHFdt=DATESTAMPINITIAL_AFU
Form=NULL

For saf_&mrt, &mrt=most recent data pull,
srHF='Y' if saf4, saf4a, saf5, or saf5a is 'Y';
else srHF='N' if saf4, saf4a, saf5, or saf5a is 'N'
srHFdt=SAF0a if not missing;

else srHFdt=DATESTAMPINITIAL_SAF
Form=NULL

For phfa1104, srHF=phfa1, Form=form
srHFdt is first completed date from the list phfa7, phfa9, ENTRY_DA

For phf_&mrt, &mrt=most recent data pull,
srHF=phf1, Form=form
srHFdt is first completed date from the list phf7, phf9, DATESTAMPINITIAL_PHF

Keep only records where srHF is 'Y' or 'N'

Temporary selfreportHF

Algorithm: From previous temporary dataset, sort by SubjectID, srHFDT, srHF
Keep records where missing<srHFdt<=V10DATE101 OR Form='PHF'

Start with selfreport=0 for first record per SubjectID then
Selfreport=1, if srHF='Y';
Selfreport=0, if srHF='Y' is followed by a record with srHF='N' and form='PHF'

Last record is kept for each SubjectID

Temporary srHFmed/srHFmedDt

Algorithm:
```
%macro selfreportHFmeds(dsn, medDt, med)
If &med='Y'' then do;
srHFmed=1;
srHFmedDt=&medDt;
%mend;
%selfreportHFmeds(v1.msra, MSRA11, msra08d);
%selfreportHFmeds(v2.msrb, MSRB27, msrb24d);
%selfreportHFmeds(v3.msrc04, MSRC29, msrc24e);
%selfreportHFmeds(v4.msrd04, MSRD31, msrd24e);
%selfreportHFmeds(v5.msr, MSRF0A, msrf33h);
%selfreportHFmeds(v6.msr, MSRF0A, msrf33h);
%selfreportHFmeds(v7.msr, MSR0A, MSR33h);
%selfreportHFmeds(v9.msr, MSR0A, MSR33h);
%selfreportHFmeds(v10.msr, MSR0A, MSR33h);
%selfreportHFmeds(postv4.aful1104new, AFUL1, AFUL46D);
%selfreportHFmeds(postv4.afum1104new, AFUM1, AFUM46D);
%selfreportHFmeds(afu.afu_180425, AFU0a, AFU65D);
```

Temporary selfreportHFmeds

Algorithm: Keep records where missing<srHFmedDT<V10DATE101
selfreportHFmeds=1, if srHFmed=1 for any record;
selfreportHFmeds=0, otherwise

5.14 PREVDEFHF101 (V10 Prevalent Definite Heart Failure for Closed Event Years)

Description: A participant is defined to have prevalent definite heart failure if at least one of the following is true: (1) Prior hospitalization (01/01/2005 onward but before V10 visit) classified as Definite (A), Probable (B), or Chronic (C) HF; OR (2) Physician Heart Failure (PHF) Survey with HF onset date prior to V10 (from those with self-reported HF) in which the physician answers YES to "has this patient ever had HF or CM?"; OR (3) Hospitalization with an ICD code 428.x in first position (before 01/01/2005)

Format: 0=No,
1=Yes,
.=missing.

Algorithm: 1. Prior hospitalization (01/01/2005 onward but before V10 visit) classified as Definite (A), Probable (B), or Chronic (C) HF

2. Physician Heart Failure (PHF) Survey with HF onset date prior to V10 (from those with self-reported HF) in which the physician answers YES to "has this patient ever had HF or CM?"

3. Hospitalization with an ICD code 428.x in first position (before 01/01/2005)

Using Temporary Variables:
If V10DATE101 is not missing AND (HospHF_2005onward=1 OR HF_byPHF=1 OR HF_first428_pre2005=1) then PREVDEFHF101=1
Else PREVDEFHF101=0

Source variables: V10DATE101, HospHF_2005onward, HF_byPHF, HF_first428_pre2005

5.15 PREVDEFPOSSHF101 (V10 Prevalent Definite OR Possible Heart Failure for Closed Event Years)

Description: A participant is defined to have prevalent definite OR possible heart failure if at least one of the following is true: (1) Prior hospitalization (01/01/2005 onward but before V10 visit) classified as Definite (A),

Probable (B), or Chronic (C) HF; OR (2) Physician Heart Failure Survey with HF onset date prior to V10 (from those with self-reported HF) in which the physician answers YES to "has this patient ever had HF or CM?"; OR (3) Hospitalization with an ICD code 428.x in first position (before 01/01/2005); OR (4) Hospitalization with an ICD code 428.x in any position other than the first position (any time before 01/01/2005); OR (5) Self-report of HF at AFU prior to V10 or at visits 3-4*, not refuted by the physicians health survey (temporal association will need to be considered); OR (6) Self-report of treatment for HF from any study visit or AFU prior to V10. *Note that self-reported HF was only asked at V3 and V4.

Format:

0=No,
1=Yes,
.=missing.

Algorithm:

1. Prior hospitalization (01/01/2005 onward but before V10 visit) classified as Definite (A), Probable (B), or Chronic (C) HF

2. Physician Heart Failure Survey with HF onset date prior to V10 (from those with self-reported HF) in which the physician answers YES to "has this patient ever had HF or CM?"

3. Hospitalization with an ICD code 428.x in first position (before 01/01/2005)

4. Hospitalization with an ICD code 428.x in any position other than the first position (any time before 01/01/2005)

5. Self-report of HF at AFU prior to V10 or at visits 3-4*, not refuted by the physicians health survey (temporal association will need to be considered)

6. Self-report of treatment for HF from any study visit or AFU prior to V10

*Self-reported HF at V1, V2, V5, V6, V7, V9, V10 not asked

Using Temporary Variables:
If V10DATE101 is not missing AND (PREVDEFHF101=1 OR HF_non1st428_pre2005=1 OR selfreportHF=1 OR selfreportHFmeds=1) then PREVDEFPOSSHF101=1
Else PREVDEFPOSSHF101=0

Source variables: V10DATE101, PREVDEFHF101, HF_non1st428_pre2005, selfreportHF, selfreportHFmeds

6. MEDICATION USE

ARIC uses Generic Product Identifier (GPI) codes to identify use of selected medications based on data collected in the Medication Survey (MSR) form at the ARIC clinic visits. The derived medication variable definitions were reviewed following Visit 7 and resulted in new variable versions for most of the medications of interest. Analyses should only use the most recent versions of the derived medication variables.

Recall for Visits 5 - 7, the old variable versions were retained in the updated derived datasets for reproducibility purposes, however, those variables include "DO NOT USE" in the label.

In Visits 9 - 11, medication variables will include version number in the label. Medication variables with no adjustments will have “ver1” added to the label, and medication variables with new definitions will have “ver2” added to the label. The exception is for CHOLMDCODE: ver1 and ver2 are no longer created, use the updated ver3 and ver4.

For example, CHOLMDCODE51, CHOLMDCODE61, and CHOLMDCODE71 should no longer be used for analyses as “DO NOT USE” is present in the variable label. The updated versions, CHOLMDCODE53, CHOLMDCODE63, and CHOLMDCODE73 should be analyzed. The comparable variable in Visit 10 is CHOLMDCODE103, which now includes “ver3” in the variable label.

In another example, HYPTMD51, HYPTMD61, and HYPTMD71 may be used for analyses as the definitions have not changed. The comparable variable at Visit 10 is HYPTMD101, which includes “ver1” in the variable label as the definition is unchanged.

The following definitions are temporary variables, defined here as reference for the following medication-related, derived variables. They are not found on the DERIVE101 dataset.

Temporary ALLMISS

Algorithm: Do over MSR*a_gpi where * = 5 to 29:
If any MSR*a_gpi >'' then ALLMISS=0, else ALLMISS=1

Temporary MSR*a2_sub10:

Algorithm: Do over MSR*a_gpi where* = 5 to 29:
MSR*a2_sub10=substr(MSR*a_gpi,1,10)

Temporary MSR*a2_sub8:

Algorithm: Do over MSR*a_gpi where * = 5 to 29:
MSR*a2_sub8=substr(MSR*a_gpi,1,8)

Temporary MSR*a2_sub6:

Algorithm: Do over MSR*a_gpi where * = 5 to 29:
MSR*a2_sub6=substr(MSR*a_gpi,1,6)

Temporary MSR*a2_sub4:

Algorithm: Do over MSR*a_gpi where * = 5 to 29:
MSR*a2_sub4=substr(MSR*a_gpi,1,4)

Temporary MSR*a2_sub2:

Algorithm: Do over MSR*a_gpi where * = 5 to 29:
MSR*a2_sub2=substr(MSR*a_gpi,1,2)

Temporary FOUNDCHOL103

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub6 in (279930, 390000-399999, 409925, 771030, 781044) or MSR*a2_sub8 in (40100025, 81250070, 96428049, 96782839) or MSR*a2_sub10 in (9646564700, 9652504244) then FOUNDCHOL93=1
Else FOUNDCHOL93=0.

Temporary FOUNDCHOL104

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub4 in (2599) or (MSR*a2_sub6 in (121030, 121045, 121085, 121099, 363000, 369920, 376000, 379900, 379910, 590700 - 594000) and MSR*a2_sub10 NOT in (1210990250)) or MSR*a2_sub8 in (12105005, 24991002, 24993002, 24995002, 33100010, 33100025, 33100040, 33100045, 33100050, 33200020, 33200021, 33200022, 33200025, 33200030, 37200010) or MSR*a2_sub10 in (2400001500, 2400001600, 2400001700, 2400003000, 2400003500, 2400003504, 2400005500) then FOUNDCHOL104=1, else FOUNDCHOL104=0.

Temporary FOUNDHYPT102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If (MSR*a2_sub6 in (330000-339999 or 340000-349999 or 360000-369999 or 370000-379999) and (MSR*a2_sub8 NOT in (37400010) and MSR*a2_sub6 NOT in (379920))) or MSR*a2_sub4 in (4099) or (MSR33d = 'Y') then FOUNDHYPT102=1,
Else FOUNDHYPT102=0.

Temporary FOUNDSTAT102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If (MSR*a2_sub4 in (3940) and MSR*a2_sub8 NOT in (39409908)) or MSR*a2_sub6 in (279930, 399940, 409925) or MSR*a2_sub8 in (96428049, 96785839) then FOUNDSTAT102=1,
Else FOUNDSTAT102=0.

Temporary FOUNDACOAG101

Algorithm: Do over MSR*a2_sub2 where * = 5 to 29

If MSR*a2_sub2 equal 83 then FOUNDACOAG101=1
Else FOUNDACOAG101=0.

Temporary FOUNDASP102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
if MSR*a2_sub4 IN (6410) or MSR*a2_sub10 in (4399100232, 4399590415, 4399590419, 6030990225, 6499000220, 6499000221, 6499000225, 6499000320, 6499000321, 6499000340, 6499000450, 6499000460, 6499100222, 6499100330, 6599000222, 6599100430, 6599130310,7599000210, 7599000310, 7599000320, 8515001000, 8515990220) then FOUNDASP102=1,
Else FOUNDASP102=0.

Temporary FOUNDANTIANX102

Algorithm: Do over MSR*a2_sub2 where * = 5 to 29
If MSR*a2_sub2 in (57) then FOUNDANTIANX102=1, else FOUNDANTIANX102=0.

Temporary FOUNDANTIPSYCH102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub2 in (59) or MSR*a2_sub10 in (6200003000) then FOUNDANTIPSYCH92=1,
Else FOUNDANTIPSYCH102=0.

Temporary FOUNDHYPNOT102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub2 in (59) or MSR*a2_sub6 in (439935, 439940, 439959, 439966, 439968, 439975, 439988, 600000-609999) or MSR*a2_sub8 in (41100010-41100030, 41200030, 41200040, 41400020, 41500020, 41992002, 43992002, 50200030) or MSR*a2_sub10 in (4399300220, 4399300221, 4399300224, 4399300230, 4399300234, 4399300248, 4399300250, 4399300254, 4399300255, 4399300258, 4399300270, 4399300292, 4399300293, 4399300295, 4399300296, 4399520231-4399520234, 4399520236, 4399530310, 4399530311, 4399530313, 4399530314, 4399530317, 4399530319, 4399530320, 4399530327, 4399530354, 4399530357, 4399530390, 4399570210, 4399570220, 4399570230, 4399570243, 4399580306, 4399580308, 4399580312, 4399580315, 4399580330, 4399580332, 4399580346, 4399580348, 4399580349, 4399580350, 4399580354, 4399580361, 4399580362, 4399580363, 4399580364, 4399580367, 4399580368, 4399580376, 4399580377, 4399800426, 4399800439, 4399800470, 4399890315, 4399890325, 4399890332, 4399890335,

5030990210, 6499000280, 6599300220) then
FOUNDHYPNOT102=1,
Else FOUNDHYPNOT102=0.

Temporary FOUNDANTICONV102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub2 in (72) or MSR*a2_sub8 in (49109904, 59400015, 60100060) or MSR*a2_sub10 in (9672561675) then FOUNDANTICONV102=1, else FOUNDANTICONV102=0.

Temporary FOUNDANTIDEM102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub4 in (6205) or MSR*a2_sub8 in (62000001) then FOUNDANTIDEM102=1,
Else FOUNDANTIDEMT102=0.

Temporary FOUNDCNSALT102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If FOUNDHYPNOT102=1 or FOUNDANTICONV102=1 then FOUNDCNSALT102=1;
Else if MSR*a2_sub2 in (57 - 60) or MSR*a2_sub4 in (6140, 6299) or MSR*a2_sub6 in (613540, 932000) or MSR*a2_sub8 in (62206040, 96426631) or MSR*a2_sub10 in (9652646380) then FOUNDCNSALT102=1,
Else FOUNDCNSALT102=0.

Temporary FOUNDDIAB101

Algorithm: Do over MSR*a2_sub2 where * = 5 to 29
If MSR*a2_sub2=27 then FOUNDDIAB101=1, else FOUNDDIAB101=0.

Temporary FOUNDBETA102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If (MSR*a2_sub2in (33)) or MSR*a2_sub6 in (369920,369927, 369988) then FOUNDBETA102=1,
Else FOUNDBETA102=0.

Temporary FOUNDANGINH102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub6 in (361000, 369918, 369985) or MSR*a2_sub8 in (96645857) then FOUNDANGINH102=1;
Else FOUNDANGINH102=0.

Temporary FOUNDANGIANT102

Algorithm: Do over MSR*a2_sub6 where * = 5 to 29
If MSR*a2_sub6 in (361500, 369930-369945, 369965) then FOUNDANGIANT102=1;
Else FOUNDANGIANT102=0.

Temporary FOUNDALDANT102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub6 IN (362500) or MSR*a2_sub10 IN (3750002000, 3799000220) then FOUNDALDANT102=1;
Else FOUNDALDANT102=0.

Temporary FOUNDLOOPDIU102

Algorithm: Do over MSR*a2_sub# where * = 5 to 29
If MSR*a2_sub6 in (372000) or MSR*a2_sub8 in (96508007) then FOUNDLOOPDIU102=1;
Else FOUNDLOOPDIU102=0.

Temporary FOUNDDIG102

Algorithm: Do over MSR*a2_sub8 where * = 5 to 29
If MSR*a2_sub8 in (31200010, 96485821) then FOUNDDIG102=1;
Else FOUNDDIG102=0.

6.1 CHOLMDCODE103 (V10 Cholesterol Lowering Medication in past 4 weeks – Using Medi-Span GPI Code ver3)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDCHOL103=1 then CHOLMDCODE103=1;

Else if FOUNDCHOL103=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then CHOLMDCODE103=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then CHOLMDCODE103=.T;

Else CHOLMDCODE103=.;

Source variable(s): FOUNDCHOL102, ALLMISS, MSR2

6.2 CHOLMDCODE104 (V10 Medications Which Secondarily Affect Cholesterol in past 4 Weeks – Using Medi-Span GPI Code ver4)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDCHOL104=1 then CHOLMDCODE104=1;

Else if FOUNDCHOL104=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then CHOLMDCODE104=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then CHOLMDCODE104=.T;

Else CHOLMDCODE104=.;

Source variable(s): FOUNDCHOL104, ALLMISS, MSR2

6.3 HYPTMD101 (V10 Hypertension Medications in past 4 Weeks: Self-reported ver1)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If (MSR2 NE 'T') and (MSR33D='Y') then HYPTMD101=1;

Else if (MSR2='T' and MSR33D='') OR MSR33D='N' then HYPTMD101 = 0;

Else If ((MSR2 NE 'T') and (MSR33D='U' or MSR33D = '')) or ((MSR2='T') and (MSR33D = 'Y' or MSR33D='U')) then HYPTMD101=.T;

Source variable(s): MSR2, MSR33D

6.4 HYPTMDCODE102 (V10 Hypertension Lowering Medication in past 4 Weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDHYPT102=1 OR (MSR2 NE ‘T’) and (MSR33D='Y')) then HYPTMDCODE102=1;

Else if FOUNDHYPT102=0 AND ((ALLMISS=1 AND MSR2=’T’) OR (ALLMISS=0) OR (MSR2='T' and MSR33D='') OR (MSR33D='N’)) then HYPTMDCODE102=0;

Else if ALLMISS=1 AND (MSR2=’F’ OR MSR2=’ ‘) then HYPTMDCODE102=.T;

Else HYPTMDCODE102=.;

Source variable(s): FOUNDHYPT102, MSR2, MSR33D, ALLMISS

6.5 STATINCODE102 (V10 Statin Use in past 4 Weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDSTAT102=1 then STATINCODE102=1;

Else if FOUNDSTAT102=0 AND ((ALLMISS=1 AND MSR2=’T’) OR (ALLMISS=0)) then STATINCODE102=0;

Else if ALLMISS=1 AND (MSR2=’F’ OR MSR2=’’) then STATINCODE102=.T;

Else STATINCODE102=.;

Source variable(s): FOUNDSTAT102, ALLMISS, MSR2

6.6 ANTICOAGCODE101 (V10 Anticoagulant Use in past 4 Weeks – Using Medi-Span GPI Code ver1)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDACOAG101=1 then ANTICOAGCODE101=1;

Else if FOUNDACOAG101=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ANTICOAGCODE101=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ANTICOAGCODE101=.T;

Else ANTICOAGCODE101=.;

Source variable(s): FOUNDACOAG101, ALLMISS, MSR2

6.7 ASPIRINCODE102 (V10 Aspirin Use in past 4 Weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDASP102=1 then ASPIRINCODE102=1;

Else if FOUNDASP102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ASPIRINCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ASPIRINCODE102=.T;

Else ASPIRINCODE102=.;

Source variable(s): FOUNDASP102, ALLMISS, MSR2

6.8 ANTIANXMDCODE102 (V10 Antianxiety Medication in past 4 Weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes

.T=Missing

Type: Numeric

Algorithm: If FOUNDANTIANX102=1 then ANTIANXMDCODE102=1;

Else if FOUNDANTIANX102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ANTIANXMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ANTIANXMDCODE102=.T;

Else ANTIANXMDCODE102=.;

Source variable(s): FOUNDANTIANX102, ALLMISS, MSR2

6.9 ANTIPSYCHMDCODE102 (V10 Antipsychotic Medication in past 4 Weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDANTIPSYCH102=1 then ANTIPSYCHMDCODE102=1;

Else if FOUNDANTIPSYCH102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ANTIPSYCHMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ANTIPSYCHMDCODE102=.T;

Else ANTIPSYCHMDCODE102=.;

Source variable(s): FOUNDANTIPSYCH102, ALLMISS, MSR2

6.10 HYPNOTMDCODE102 (V10 Hypnotic/Sedative Medication in past 4 Weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDHYPNOT102=1 then HYPNOTMDCODE102=1;

Else if FOUNDHYPNOT102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then HYPNOTMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then HYPNOTMDCODE102=.T;

Else HYPNOTMDCODE102=.;

Source variable(s): FOUNDHYPNOT102, ALLMISS, MSR2

6.11 ANTICONVMDCODE102 (V10 Anticonvulsant Medication in past 4 weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDANTICONV102=1 then ANTICONVMDCODE102=1;

Else if FOUNDANTICONV102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ANTICONVMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ANTICONVMDCODE102=.T;

Else ANTICONVMDCODE102=.;

Source variable(s): FOUNDANTICONV102, ALLMISS, MSR2

6.12 ANTIDEMMDCODE102 (V10 Antidementia/Nootropic Medication in past 4 weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDANTIDEM102=1 then ANTIDEMMDCODE102=1;

Else if FOUNDANTIDEM102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ANTIDEMMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ANTIDEMMDCODE102=.T;

Else ANTIDEMMDCODE102=.;

Source variable(s): FOUNDANTIDEM102, ALLMISS, MSR2

6.13 CNSALTMDCODE102 (V10 CNS Altering Medication in past 4 weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDCNSALT102=1 then CNSALTMDCODE102=1;

Else if FOUNDCNSALT102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then CNSALTMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then CNSALTMDCODE102=.T;

Else CNSALTMDCODE102=.;

Source variable(s): FOUNDCNSALT102, ALLMISS, MSR2

6.14 DIABMDCODE101 (V10 Diabetic Medications in past 4 weeks – Using Medi-Span GPI Code ver1)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDDIAB101=1 then DIABMDCODE101=1;

Else if FOUNDDIAB101=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then DIABMDCODE101=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then DIABMDCODE101=.T;

Else DIABMDCODE101=.;

Source variable(s): FOUNDDIAB101, ALLMISS, MSR2

6.15 BETAMDCODE102 (V10 Beta-Blocker in past 4 weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDBETA102=1 then BETAMDCODE102=1;

Else if FOUNDBETA102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then BETAMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then BETAMDCODE102=.T;

Else BETAMDCODE102=.;

Source variable(s): FOUNDBETA102, ALLMISS, MSR2

6.16 ANGINHMDCODE102 (V10 Angiotensin converting enzyme inhibitor in past 4 weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDANGINH102=1 then ANGINHMDCODE102=1;

Else if FOUNDANGINH102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ANGINHMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ANGINHMDCODE102=.T;

Else ANGINHMDCODE102=.;

Source variable(s): FOUNDANGINH102, ALLMISS, MSR2

6.17 ANGIANTMDCODE102 (V10 Angiotensin II receptor antagonists in past 4 weeks – Using Medi-Span GPI Code ver2))

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDANGIANT102=1 then ANGIANTMDCODE102=1;

Else if FOUNDANGIANT102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ANGIANTMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ANGIANTMDCODE102=.T;

Else ANGIANTMDCODE102=.;

Source variable(s): FOUNDANGIANT102, ALLMISS, MSR2

6.18 ALDANTMDCODE102 (V10 Aldosterone Antagonist in past 4 weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDALDANT102=1 then ALDANTMDCODE102=1;

Else if FOUNDALDANT102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then ALDANTMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then ALDANTMDCODE102=.T;

Else ALDANTMDCODE102=.;

Source variable(s): FOUNDALDANT102, ALLMISS, MSR2

6.19 LOOPDIUMDCODE102 (V10 Loop Diuretic in past 4 weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDLOOPDIU102=1 then LOOPDIUMDCODE102=1;

Else if FOUNDLOOPDIU102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then LOOPDIUMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then LOOPDIUMDCODE102=.T;

Else LOOPDIUMDCODE102=.;

Source variable(s): FOUNDLOOPDIU102, ALLMISS, MSR2

6.20 DIGMDCODE102 (V10 Digoxin in past 4 weeks – Using Medi-Span GPI Code ver2)

Format: 0=No
1=Yes
.T=Missing

Type: Numeric

Algorithm: If FOUNDDIG102=1 then DIGMDCODE102=1;

Else if FOUNDDIG102=0 AND ((ALLMISS=1 AND MSR2='T') OR (ALLMISS=0)) then DIGMDCODE102=0;

Else if ALLMISS=1 AND (MSR2='F' OR MSR2='') then DIGMDCODE102=.T;

Else DIGMDCODE102=.;

Source variable(s): FOUNDDIG102, ALLMISS, MSR2

7. PHYSICAL FUNCTION

The physical function variables mainly use the data collected on the ARIC PFX form. The short physical performance battery (SPPB) is a group of measures that combines the results of the gait speed, chair stand and balance tests (Guralnik et al., 2000). The scores range from 0 (worst performance) to 12 (best performance).

7.1 SPPBCS101 (V10 Physical Function Chair Stand)

Format: integer variable (0-4 possible points)

Type: Numeric

Algorithm: If (PFX1 in (2,3,4)) OR (PFX2 in (2,3)) OR (PFX2b_DER>=60) then SPPBCS101=0;
Else if (16.70<= PFX2b_DER <60) then SPPBCS101=1;
Else if (13.70<= PFX2b_DER <16.70) then SPPBCS101=2;
Else if (11.20<= PFX2b_DER <13.70) then SPPBCS101=3;
Else if (. <PFX2b_DER <11.20) then SPPBCS101=4;
Else SPPBCS101=.;

Source variable(s): PFX1, PFX2, PFX2b_DER

7.2 SPPBST101 (V10 Physical Function Semi Tandem Stand)

Format: integer variable

Type: Numeric

Algorithm: If (PFX3 in (2,3,4)) then SPPBST101=0;
Else if (PFX3=5) then SPPBST101=1;
Else SPPBST101=.;

Source variable(s): PFX3

7.3 SPPBSBS101 (V10 Physical Function Side-by-Side Stand)

Format: integer variable

Type: Numeric

Algorithm: If (PFX4 in (2,3,4)) then SPPBSBS101=0;
Else if (PFX3=5 OR PFX4=5) then SPPBSBS101=1;
Else SPPBSBS101=.;

Source variable(s): PFX3, PFX4

7.4 SPPBTS101 (V10 Physical Function Tandem Stand)

Format: integer variable

Type: Numeric

Algorithm: If max(PFX5, PFX6) in (2,3,4) then SPPBTS101=0;
else if 3 <= max(PFX5a_DER, PFX6a_DER) <10 then SPPBTS101=1;
else if max(PFX5, PFX6) in (5) then SPPBTS101=2;
else SPPBTS101=.;

Source variable(s): PFX5, PFX5a_DER, PFX6, PFX6a_DER

7.5 SPPBBAL101 (V10 Physical Function Summary Balance Score)

Format: integer variable

Type: Numeric

Algorithm: SPPBBAL101=sum(of SPPBST101,SPPBSBS101,SPPBTS101);

Source variable(s): SPPBSBS101, SPPBST101, SPPBTS101

7.6 SPPB4M101 (V10 Physical Function 4 Meter Walk Score)

Format: integer variable (0-4)

Type: Numeric

Algorithm: If max(PFX7,PFX8)=2 then SPPB4M101=0;
Else if (WALK4M101> 8.70) SPPB4M101=1;
Else if (6.21 <= WALK4M101<= 8.70) then SPPB4M101=2;
Else if (4.82 <= WALK4M101< 6.21) then SPPB4M101=3;
Else if (.z < WALK4M101< 4.82) then SPPB4M101=4;
Else SPPB4M101=.T;

Source variable(s): PFX7, PFX8, WALK4M101

7.7 SPPB101 (V10 Short Physical Performance Summary Battery Score)

Format: integer variable

Type: Numeric

Algorithm: The SPPB total score is only calculated when all three of its subcomponent tasks are not missing. If any of SPPBCS101, SPPBBAL101, or SPPB4M101 are missing, then SPPB101 is set to missing.
If SPPBCS101>NULL and SPPBBAL101>NULL and SPPB4M101>NULL then SPPB101=sum(of SPPBCS101,SPPBBAL101,SPPB4M101);
Else SPPB101 = NULL;

Source variable(s): SPPB4M101, SPPBBAL101, SPPBCS101

7.8 WALK4M101 (V10 Physical Function 4 Meter Walk, Fastest Time of 2 Trials)

Format: continuous variable (f4.2)

Type: Numeric

Algorithm: The better of (1) PFX7a and PFX8a if walking aid is used OR the better of (2) PFX7b and PFX8b if no walking aid is used. There are some instances where a walking aid is used in one trial and not in the other trial. The variable will be the fastest time walked regardless of using the aid or not. All participants at the clinic visits should have a timed walk. Any who did not do the test due to "not attempted/unable" will have a missing value for this variable. If only one trial completed, use the results from that trial.

If .<max(pfx7,pfx8)<=2 then WALK4M101=.T;
Else WALK4M101 = min(PFX7a_der, PFX7b_der, PFX8a_der, PFX8b_der);

Source variable(s): PFX7, PFX7a_der, PFX7b_der, PFX8, PFX8a_der, PFX8b_der

7.9 WALKAID101 (V10 Physical Function 4 Meter Walk: Used Walking Aid)

Format: 0=No,
1=Yes,
.T=missing.

Type: Numeric

Algorithm: This indicator variable will be set to 1 when a patient used a walking aid during the 4 meter walk. The walk is performed twice and in the instance where a patient uses an aid in one trial, but not in the other, this variable will be set according to the presence or absence of a walking aid that goes with the fastest time of the 2 trials.

If WALK4M101>.T and (WALK4M101=PFX7a_der or WALK4M101=PFX8a_der) then WALKAID101 = 1;
Else if WALK4M101>.T then WALKAID101 = 0;
Else WALKAID101=.T;

Source variable(s): PFX7a_der, PFX8a_der, WALK4M101

7.10 GRIPBEST101 (V10 Physical Function Grip, Best of 2 Trials)

Format: continuous variable

Type: Numeric

Algorithm: Max of PFX11b and PFX11c. Only participants who respond "Both" to PFX10a should be excluded (i.e., surgery on both hands). Less than 2% are missing grip strength as of May 2012, likely to have little impact on inferences. Standard approaches to missing data such as sensitivity analyses with multiple imputations can be employed if inappropriate to ignore missingness.

If PFX10 NE missing and PFX10a NE 'B' then GRIPBEST101=max(PFX11b, PFX11c);

Source variable(s): PFX10, PFX10a, PFX11b, PFX11c

7.11 GRIPMEAN101 (V10 Physical Function Grip, Mean of 2 Trials or Result for 1 Trial)

Format: continuous variable

Type: Numeric

Algorithm: Mean of PFX11b and PFX11c. Only participants who respond "Both" to PFX10a should be excluded (i.e., surgery on both hands). The variable will be missing if less than 2 trials are completed. Less

than 2% are missing grip strength as of May 2012, likely to have little impact on inferences. Standard approaches to missing data such as sensitivity analyses with multiple imputations can be employed if inappropriate to ignore missingness.

If PFX10 NE missing and PFX10a NE 'B' then GRIPMEAN101=mean(PFX11b,PFX11c);

Source variable(s): PFX10, PFX10a, PFX11b, PFX11c

7.12 V9V10WTDELTAP101 (% of V9 weight change from V9 to V10 (neg val means loss) – used in frailty weight loss component)

Description: The change in weight from visit 7 to Visit 10 is calculated and presented as the % of visit 7 weight, rounded to 0.1.

Format: continuous variable (0-99, %)

Type: Numeric

Algorithm: ((v10ant4-v9ant4)/v9ant4)*100, where v9ant4 = ant4 from visit 9 and V10ant4=ant4 from Visit 10

Source variable(s): ant4

7.13 WALKSPEED15FT101 (V10 Time in seconds used to walk 15ft – used in frailty slowness component)

Description: The physical function form collects data on times to walk 4 meters. The WALKSPEED15FT101 variable uses the 4m walk rate to calculate the time in seconds needed to walk 15ft.

Format: numeric continuous variable

Type: Numeric

Algorithm: WALKSPEED15FT101=MIN((15/3.28084)* (pfx7a_der/4), (15/3.28084)*(pfx7b_der/4), (15/3.28084)*(pfx8a_der/4), (15/3.28084)*(pfx8b_der/4))

Source variable(s): PFX7A_DER, PFX7B_DER, PFX8A_DER, PFX8B_DER

7.14 TR1WALK4MSP101 (V10 4M Walking Speed for Trial 1 (m/sec))

Format: continuous variable

Type: Numeric

Algorithm: TR1WALK4MSP101=4/(max(PFX7a_der, PFX7b_der)

Source variable(s): PFX7a, PFX7b

7.15 TR2WALK4MSP101 (V10 4M Walking Speed for Trial 2 (m/sec))

Format: continuous variable

Type: Numeric

Algorithm: TR2WALK4MSP101=4/(max(PFX8a_der, PFX8b_der)

Source variable(s): PFX8a, PFX8b

7.16 AVGWALK4MSP101 (V10 Average 4M Walking Speed (m/sec))

Format: continuous variable

Type: Numeric

Algorithm: AVGWALK4MSP101=mean(TR1WALK4MSP101, TR2WALK4MSP101)

Source variable(s): TR1WALK4MSP101, TR2WALK4MSP101

7.17 MINWALK4MSP101 (V10 Fastest 4M Walking Speed for Both Trials (m/sec))

Format: continuous variable

Type: Numeric

Algorithm: MINWALK4MSP101=min(TR1WALK4MSP101, TR2WALK4MSP101) ;

Source variable(s): TR1WALK4MSP101, TR2WALK4MSP101

7.18 UNABLETOWALK101 (V10 Indicator Variable Noting PPT No Attempt to Walk 4M, Not Able)

Format: 0=No,
1=Yes,
.=Missing.

Type: Numeric

Algorithm: If PFX7 and PFX8 are both NULL then
UNABLETOWALK101=NULL
Else if PFX7=2 and PFX8 = missing or 2 then
UNABLETOWALK101=1
Else UNABLETOWALK101=0

Source variable(s): PFX7, PFX8

7.19 UNINTEND_WTLOSS101 (V10 Unintentional weight loss)

Format: 0=No,
1=Yes,
.=missing.

Type: Numeric

Algorithm: If TMW1 is not missing then do;
If TMW3=0 then UNINTEND_WTLOSS101=1;
Else UNINTEND_WTLOSS101=0;

Source variable(s): TMW1, TMW3

7.20 TMW_TOTFT101 (V10 Total Feet Walked in 2-Minute Walk)

Format: continuous variable

Type: Numeric

Algorithm: Calculate if TMW_PRES =1
TMW_TOTFT101=sum((tmw8*50),tmw9);

Source variable(s): TMW_PRES, TMW8, TMW9

7.21 TMW_COMPSPEEDFTPERSEC101 (V10 TMW Speed for Completers (ft/sec))

Format: continuous variable

Type: Numeric

Algorithm: Calculate if TMW_PRES =1 and tmw11=5
TMW_COMPSPEEDFTPERSEC101=tmw_totft101/120;

Source variable(s): TMW11, TMW_TOTFT101

7.22 TMW_COMPSPEEDMPERSEC101 (V10 TMW Speed for Completers (m/sec))

Format: continuous variable

Type: Numeric

Algorithm: Calculate if TMW_PRES =1 and tmw11=5
TMW_COMPSPEEDMPERSEC101=tmw_totft101*0.3048/120

Source variable(s): TMW11, TMW_TOTFT101

7.23 TMW_NONCOMPSECWALK101 (V10 TMW Seconds Walking for Noncompleters)

Format: continuous variable

Type: Numeric

Algorithm: Calculate if TMW_PRES =1 and tmw11=4
TMW_NONCOMPSECWALK101=sum((tmw12a*60),tmw12b)

Source variable(s): TMW11, TMW12a, TMW12b

7.24 TMW_ NONCOMPSPEEDFTPERSEC101 (V10 TMW Speed for Noncompleters (ft/sec))

Format: continuous variable

Type: Numeric

Algorithm: Calculate if tmw_pres=1 and tmw11=4 and
tmw_noncompsecwalk101>0

TMW_NONCOMPSPEEDFTPERSEC101=tmw_totft101/tmw_noncompsecwalk101

Source variable(s): tmw11, tmw_noncompsecwalk101, tmw_totft101

7.25 TMW_ NONCOMPSPEEDMPERSEC101 (V10 TMW Speed for Noncompleters (m/sec))

Format: continuous variable

Type: Numeric

Algorithm: Calculate if tmw_pres=1 and tmw11=4 and tmw_noncompsecwalk101>0
TMW_NONCOMPSPEEDMPERSEC101=tmw_totft101*0.3048/tmw_noncompsecwalk101

Source variable(s): tmw11, tmw_noncompsecwalk101, tmw_totft101

7.26 EXHAUST101 (V10 Responded 2 or 3 on CES3 or CES11 (CESD) – frailty exhaustion component)

Description: Second component in considering frailty of PPT based on exhaustion from depression.

Format: 0=No
1=Yes
.=Missing

Type: Numeric

Algorithm: If (CES3=2 OR CES3=3) OR (CES11=2 OR CES11=3) then EXHAUST101=1;

Else if CES3=. AND CES11=. then EXHAUST101=.;

Else EXHAUST101=0;

Source variable(s): CES2, CES3, CES11

7.27 WTLOSSCOMPA101 (V10 Lost >5% weight or Low BMI (<18.5) – frailty weight loss component for 101a definition)

Description: First component in considering frailty of PPT based on weight loss greater than 5% or low BMI.

Format: 0=No
1=Yes
.=Missing

Type: Numeric

Algorithm: WTLOSSCOMP101=1, if (.<V9V10WTDELTAP101 <=-5) OR (.<BMI101<18.5)

WTLOSSCOMP101=0, if (V9V10WTDELTAP101 >-5) AND (BMI101>=18.5)

WTLOSSCOMP101=NULL, if V9V10WTDELTAP101 and BMI101 are NULL

Source variable(s): V9V10WTDELTAP101 , BMI101

7.28 WTLOSSCOMPB101 (V9 Lost >10% weight or Low BMI (<18.5) – frailty weight loss component for 101b definition)

Description: First component in considering frailty of PPT based on more restrictive weight loss greater than 10% or low BMI.

Format: 0=No
1=Yes
.=Missing

Type: Numeric

Algorithm: WTLOSSCOMP101=1, if (.<V9V10WTDELTAP101 <=-10) OR (.<BMI101 <18.5)

WTLOSSCOMP101=0, if (V9V10WTDELTAP101 >-10) AND (BMI101 >=18.5)

WTLOSSCOMP101=NULL, if V9V10WTDELTAP101 and BMI101 are NULL

Source variable(s): V9V10WTDELTAP101 and, BMI101

7.29 WALKSPEEDCOMP101 (V10 Slowest 20% time to walk 15ft – frailty slowness component)

Description: Fourth component in considering frailty of PPT based on slowness by walking.

Format:
0=No
1=Yes
.=Missing

Type: Numeric

Algorithm: WALKSPEEDCOMP101=1:

Males:
IF (gender = 'M' AND V6IN129 <=173 AND Walkspeed15ft101>= 7)
OR
IF (gender = 'M' AND V6IN129 > 173 AND Walkspeed15ft101>= 6)
OR

Females:
IF (gender = 'F' AND V6IN129 <=159 AND Walkspeed15ft101>= 7)
OR
IF (gender = 'F' AND V6IN129 > 159 AND Walkspeed15ft101>= 6)

WALKSPEEDCOMP101=NULL:
IF gender=NULL or V6IN129=NULL or Walkspeed15ft101= NULL
ELSE WALKSPEEDCOMP101=0

Source variable(s): gender, V6IN129 (participant height last measured at V5), Walkspeed15ft101

7.30 GRIPSTRENGTHCOMP101 (V10 Lowest 20% grip strength – frailty weakness component)

Description: Fifth component in considering frailty of PPT based on grip strength and BMI.

Format:
0=No
1=Yes
.=Missing

Type: Numeric

Algorithm: GRIPSTRENGTHCOMP101=1:

Males:
IF gender = 'M' AND BMI101 <= 24 AND .<MAX(pfx11b, pfx11c) <= 29 OR
IF gender = 'M' AND 24 < BMI101 <= 26 AND .<MAX(pfx11b, pfx11c) <= 30 OR
IF gender = 'M' AND 26 < BMI101 <= 28 AND .<MAX(pfx11b, pfx11c) <= 30 OR
IF gender = 'M' AND BMI101 > 28 AND .<MAX(pfx11b, pfx11c) <= 32

Females:
IF gender = 'F' AND BMI101 <= 23 AND .<MAX(pfx11b, pfx11c) <= 17 OR
IF gender = 'F' AND 23 < BMI101 <= 26 AND .<MAX(pfx11b, pfx11c) <= 17.3 OR
IF gender = 'F' AND 26 < BMI101 <= 29 AND .<MAX(pfx11b, pfx11c) <= 18 OR
IF gender = 'F' AND BMI101 > 29 AND .<MAX(pfx11b, pfx11c) <= 21

GRIPSTRENGTHCOMP101=NULL:
IF gender=NULL or BMI101 =NULL or (pfx11b and pfx11c)=NULL

GRIPSTRENGTHCOMP101=0:
Otherwise

Source variable(s): gender, BMI101, pfx11b, pfx11c

7.31 GAITSPEED101 (V10 Gait Speed (m/sec))

Type: Numeric

Algorithm: If WALK4M101>0 then GAITSPEED101=4/WALK4M101

Source variable(s): WALK4M101

8. HEARING RESULTS

8.1 PTARIGHT101 (V10 Four Frequency Pure-tone Average (PTA) of the Right Ear (Db HL))

Format: numeric continuous variable

Type: Numeric

Algorithm: If AUD4a1 and AUD4a7 are both not missing and the absolute value of (AUD4a1 - AUD4a7)>10 then PTARIGHT101=NULL
Else if AUD0c="H" and AUD3="N" then PTARIGHT101=NULL
Else if AUD4a3 and AUD4a1 and AUD4a9 and AUD4a13 are all not missing then
PTARIGHT101=mean(AUD4a3,AUD4a1,AUD4a9,AUD4a13)

Source variable(s): AUD0c, AUD3, AUD4a1, AUD4a3, AUD4a7, AUD4a9, AUD4a13

8.2 PTALEFT101 (V10 Four Frequency Pure-tone Average (PTA) of the Left Ear (Db HL))

Format: continuous variable

Type: Numeric

Algorithm: If AUD4b1 and AUD4b7 are both not missing and the absolute value of (AUD4b1 - AUD4b7)>10 then PTALEFT101=NULL
Else if AUD0c="H" and AUD3="N" then PTALEFT101=NULL
Else if AUD4b3 and AUD4b1 and AUD4b9 and AUD4b13 are all not missing then
PTALEFT101=mean(AUD4b3,AUD4b1,AUD4b9,AUD4b13)

Source variable(s): AUD0c, AUD3, AUD4b1, AUD4b3, AUD4b7, AUD4b9, AUD4b13

8.3 PTABETTER101 (V10 Four Frequency Pure-tone Average (PTA) of the Better Ear (smaller PTA) (Db HL))

Format: continuous integer

Type: Numeric

Algorithm: If PTARIGHT101 not missing or PTALEFT101 not missing then
PTABETTER101=min(PTARIGHT101, PTALEFT101)

Source variable(s): PTARIGHT101, PTALEFT101

8.4 PTABETTERCAT101 (V10 Hearing Loss Category Based on the Four Frequency Pure-tone Average (PTA) of the Better Ear)

Description: Values of this categorical variable indicate the hearing loss category for hearing in the better ear. This variable is used in the PPT's summary of results.

Format: 1=No hearing loss,
2=Mild hearing loss,
3=Moderate hearing loss,
4=Severe hearing loss

Type: Numeric

Algorithm: If NULL < PTABETTER101<= 25 then PTABETTERCAT101=1
Else if 25 < PTABETTER101 <= 40 then PTABETTERCAT101=2
Else if 40 < PTABETTER101<= 70 then PTABETTERCAT101=3
Else if PTABETTER101 > 70 then PTABETTERCAT101=4

Source variable(s): PTABETTER101