HAAS data preparation: The following describes how the variables were derived or created for the HAAS data files

haas_single:

- **participant_id** - Study-defined de-identified numeric participant identifier

Renamed the HAAS 'NIA_ID' variable to 'participant_id'

- **sex** - Participant reported sex

all participants in the HAAS study were Male

```
df['sex'] = 'Male'
```

- **age_at_baseline** - age at baseline (years)

derived by subtracting participant's birthdate from the date of exam 4 (when the HAAS study started) divided by 365.25. The calculated value was rounded to the nearest integer.

```
df['age_at_baseline'] = (((df['datex4'] -
df['bdate'])/365.25).round().astype('Int64'))
```

- **age_at_living_status_censored** - age at last observation (right censored, status=0)

If present for exam 6, use exam 6 age

```
((df['datex6'] - df['bdate'])/365.25).round(0).astype('Int64'))
```

If not present for exam 6, use exam 5 age

```
((df['datex5'] - df['bdate'])/365.25).round(0).astype('Int64'))
```

If not present for exam 6 or exam 5, use exam 4 age

```
((df['datex4'] - df['bdate'])/365.25).round(0).astype('Int64'))
```

- **living_status_at_censoring** - Event is death. 0=right censored (not yet) 1= event at the "age"

all participants were right censored (not yet) ('0'). None of the participants were deceased at the beginning of the study ('1')

- **age_at_onset** - Age at onset (any symptom) of the most likely neurodegenerative primary diagnosis?

Use missing value (NaN) if participant's primary_diagnosis value is 'No neurological disorder' and diagnosis_change value is 'No':

```
(df['primary_diagnosis'] == "No neurological disorder") &
(df['diagnosis_change'] == "No")
```

Use value for HAAS 'ageflx4' variable rounded to the nearest integer (`df['ageflx4'].round().astype('Int64')`) if primary_diagnosis is not 'No neurological disorder':

```
(df['primary_diagnosis'] != "No neurological disorder")
```

Calculate and use age at exam 5 rounded to the nearest integer (`(df['datex5'] - df['bdate'])/365.25).round().astype('Int64')`) if diagnosis at exam 5 is not the same as the primary_diagnosis, is not 'No neurological disorder', and is not null:

```
(df['x5_diagnosis'] != df['primary_diagnosis']) & (df['x5_diagnosis'] !=
"No neurological disorder") & (df['x5_diagnosis'].notnull())
```

Calculate and use age at exam 6 rounded to the nearest integer (`(df['datex6'] - df['bdate'])/365.25).round().astype('Int64')`) if diagnosis at exam 6 is not the same as the primary_diagnosis, not the same as diagnosis at exam 5, and is not 'No neurological disorder':

```
(df['x6_diagnosis'] != df['primary_diagnosis']) & (df['x6_diagnosis'] !=
df['x5_diagnosis']) & (df['x6_diagnosis'] != "No neurological disorder")
```

Use missing value (NaN) if participant's primary_diagnosis value is 'No neurological disorder', exam 5 diagnosis is 'No neurological disorder', and exam 6 diagnosis is 'No neurological disorder':

```
(df['primary_diagnosis'] == "No neurological disorder") &
(df['x5_diagnosis'] == "No neurological disorder") & (df['x6_diagnosis']
== "No neurological disorder")
```

Use missing value (NaN) if participant's primary_diagnosis value is 'No neurological disorder', exam 5 diagnosis is null, and exam 6 diagnosis is null:

```
(df['primary_diagnosis'] == "No neurological disorder") &
(df['x5_diagnosis'].isnull()) & (df['x6_diagnosis'].isnull())
```

Use missing value (NaN) if participant's primary_diagnosis value is 'No neurological disorder', exam 5 diagnosis is null, and exam 6 diagnosis is 'No neurological disorder':

```
(df['primary_diagnosis'] == "No neurological disorder") &
(df['x5_diagnosis'].isnull()) & (df['x6_diagnosis'] == "No neurological
disorder")
```

Use missing value (NaN) if participant's primary_diagnosis value is 'No neurological disorder', exam 5 diagnosis is 'No neurological disorder', and exam 6 diagnosis is null:

```
(df['primary_diagnosis'] == "No neurological disorder") &
(df['x5_diagnosis']== "No neurological disorder") &
(df['x6_diagnosis'].isnull())
```

- **education_years** - Number of years of education

Renamed the HAAS 'STEDU' variable to 'education_years'

- **family_hx_1st_stroke** - Did your parents or full-sibling have a stroke(s)?

Values for the HAAS ''PARSTK07' variable are (0:Neither, 1:Father, 2:Mother, 3:Both, 9:Unknown)
Replaced HAAS values:
0 : 'No'
1: 'Yes'
2: 'Yes'
3: 'Yes'
9: 'Don't know'

- **family_hx_1st_dementia** - Did your parents or full-sibling have dementia?

Values for the HAAS ''PARALZ07' variable are (0:Neither, 1:Father, 2:Mother, 3:Both, 9:Unknown)
Replace HAAS values:
0 : 'No'
1: 'Yes'
2: 'Yes'
3: 'Yes'
9: 'Don't know'

- **availability_medication_data** - Is the medication data available? Yes or no for the dataset

If participant has medication information available (if HAAS value for any medication variable is 0, 1, or 9, where 0 = No, 1 = Yes, 9 = Don't Know), the value is 'Yes'.
If participant does not have medication information available (missing values for each variable in medication data), the value is 'No'.

- **primary_diagnosis** - Most likely primary neurodegenerative diagnosis at baseline (baseline-does not include MCI)

HAAS variables and corresponding values:
AD: 0=non-demented, 1= pure AD with no Cerebrovascular disease (CVD), 7= not pure AD but demented at exam 4.
AD1: 0=non-demented, 1=AD1*, 7=not AD1 but demented at exam 4. * xxAD1 category includes AD cases with and without CVD
VD: 0=non-demented, 1=pure VaD, 7=not pure VaD but demented at exam 4.
Other: 0=non-demented, 1=other dementias, 7=not other dementias but demented at exam 4.

Derived values:
'No neurological disorder':

```
(df['AD']==0) & (df['AD1']==0) & (df['VD']==0) & (df['OTHER']==0)
```

'Alzheimer's disease without cerebrovascular disease':

```
(df['AD']==1) & (df['AD1']==1) & (df['VD']==7) & (df['OTHER']==7)
```

'Alzheimer's disease with cerebrovascular disease':

```
(df['AD']==7) & (df['AD1']==1) & (df['VD']==7) & (df['OTHER']==7)
```

'Vascular dementia':

```
(df['AD']==7) & (df['AD1']==7) & (df['VD']==1) & (df['OTHER']==7)
```

'Other neurological disorder'

```
(df['AD']==7) & (df['AD1']==7) & (df['VD']==7) & (df['OTHER']==1)
```

- **latest_diagnosis** - Latest diagnosis or autopsy diagnosis

Derived latest_diagnosis from diagnosis at exam 6. If the participant was not present for exam 6, used the diagnosis at exam 5. If the participant was not present at exam 5, used the diagnosis at exam 4.

<u>HAAS variables for diagnosis at exam 6 and corresponding values:</u>
IIAD: 0=non-demented, 1=pure AD with no CVD, 6=demented at the 4th exam but still in the cohort at the 6th exam,7=not pure AD without CVD but demented at exam 6, 8=demented at the 5th exam but still in the cohort at the 6th exam, 9=seen at the 5th exam but missed at the 6th exam.

IIAD1: 0=non-demented, 1=AD1*, 6=demented at the 4th exam but still in the cohort at the 6th exam, 7=not AD1 but demented at exam 6, 8=demented at the 5th exam but still in the cohort at the 6th exam, 9=seen at the 5th exam but missed at the 6th exam.* xxAD1 category includes AD cases with and without CVD

IIVAD: 0=non-demented, 1=VaD, 6=demented at the 4th exam but still in the cohort at the 6th exam, 7=not VaD but demented at exam 6, 8=demented at the 5th exam but still in the cohort at the 6th exam, 9=seen at the 5th exam but missed at the 6th exam.

IIOTHER: 0=non-demented, 1=other dementias, 6=demented at the 4th exam but still in the cohort at the 6th exam, 7=not other dementia but demented at exam 6, 8=demented at the 5th exam but still in the cohort at the 6th exam, 9=seen at the 5th exam but missed at the 6th exam.

HAAS variables for diagnosis at exam 5 and corresponding values:
IAD: 0 non-demented, 1 pure AD with no CVD, 5 non-demented, refusal at the 5th exam but seen again at the 6th exam, 6 demented at the 4th exam but still in the cohort at the 5th exam, 7 not pure AD but demented at exam 5.

IAD1: 0 non-demented, 1 AD1*, 5 non-demented, refusal at the 5th exam but seen again at the 6th exam, 6 demented at the 4th exam but still in the cohort at the 5th exam, 7 not AD1 but demented at exam 5.

IVD:0 non-demented, 1 pure VaD, 5 non-demented, refusal at the 5th exam but seen again at the 6th exam, 6 demented at the 4th exam but still in the cohort at the 5th exam, 7 not pure VaD but demented at exam 5.

IOTHER: 0 non-demented, 1 other dementias, 5 non-demented, refusal at the 5th exam but seen again at the 6th exam, 6 demented at the 4th exam but still in the cohort at the 5th exam, 7 not other dementia but demented at exam 5.

‘No neurological disorder’:
```
(df['IIAD']==0) & (df['IIAD1']==0) & (df['IIVAD']==0) & (df['IIOTHER']==0)
```
‘Alzheimer's disease without cerebrovascular disease’:
```
(df['IIAD']==1) & (df['IIAD1']==1) & (df['IIVAD']==7) & (df['IIOTHER']==7)
```
‘Alzheimer’s disease with cerebrovascular disease’:
```
(df['IIAD']==7) & (df['IIAD1']==1) & (df['IIVAD']==7) & (df['IIOTHER']==7)
```
‘Vascular dementia’:
```
(df['IIAD']==7) & (df['IIAD1']==7) & (df['IIVAD']==1) & (df['IIOTHER']==7)
```
‘Other neurological disorder’:
```
(df['IIAD']==7) & (df['IIAD1']==7) & (df['IIVAD']==7) & (df['IIOTHER']==1)
```
‘No neurological disorder’:
```
(df['IIAD']==6) & (df['IIAD1']==6) & (df['IIVAD']==6) & (df['IIOTHER']==6)
& (df['AD']==0) & (df['AD1']==0) & (df['VD']==0) & (df['OTHER']==0)
```

‘Alzheimer’s disease without cerebrovascular disease’:

```
(df['IIAD']==6) & (df['IIAD1']==6) & (df['IIVAD']==6) & (df['IIOTHER']==6)
& (df['AD']==1) & (df['AD1']==1) & (df['VD']==7) & (df['OTHER']==7)
```

‘Alzheimer’s disease with cerebrovascular disease’:

```
(df['IIAD']==6) & (df['IIAD1']==6) & (df['IIVAD']==6) & (df['IIOTHER']==6)
& (df['AD']==7) & (df['AD1']==1) & (df['VD']==7) & (df['OTHER']==7)
```

‘Vascular dementia’：

```
(df['IIAD']==6) & (df['IIAD1']==6) & (df['IIVAD']==6) & (df['IIOTHER']==6)
& (df['AD']==7) & (df['AD1']==7) & (df['VD']==1) & (df['OTHER']==7)
```

‘Other neurological disorder’:

```
(df['IIAD']==6) & (df['IIAD1']==6) & (df['IIVAD']==6) & (df['IIOTHER']==6)
& (df['AD']==7) & (df['AD1']==7) & (df['VD']==7) & (df['OTHER']==1)
```

‘No neurological disorder’:

```
(df['IIAD']==8) & (df['IIAD1']==8) & (df['IIVAD']==8) & (df['IIOTHER']==8)
& (df['IAD']==0) & (df['IAD1']==0) & (df['IVD']==0) & (df['IOTHER']==0)
```

‘Alzheimer’s disease without cerebrovascular disease’:

```
(df['IIAD']==8) & (df['IIAD1']==8) & (df['IIVAD']==8) & (df['IIOTHER']==8)
& (df['IAD']==1) & (df['IAD1']==1) & (df['IVD']==7) & (df['IOTHER']==7)
```

‘Alzheimer’s disease with cerebrovascular disease’:

```
(df['IIAD']==8) & (df['IIAD1']==8) & (df['IIVAD']==8) & (df['IIOTHER']==8)
& (df['IAD']==7) & (df['IAD1']==1) & (df['IVD']==7) & (df['IOTHER']==7)
```

‘Vascular dementia’：

```
(df['IIAD']==8) & (df['IIAD1']==8) & (df['IIVAD']==8) & (df['IIOTHER']==8)
& (df['IAD']==7) & (df['IAD1']==7) & (df['IVD']==1) & (df['IOTHER']==7)
```

‘Other neurological disorder’:

```
(df['IIAD']==8) & (df['IIAD1']==8) & (df['IIVAD']==8) & (df['IIOTHER']==8)
& (df['IAD']==7) & (df['IAD1']==7) & (df['IVD']==7) & (df['IOTHER']==1)
```

‘No neurological disorder’:

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==0) & (df['IAD1']==0) & (df['IVD']==0) & (df['IOTHER']==0)
```

‘Alzheimer’s disease without cerebrovascular disease’:

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==1) & (df['IAD1']==1) & (df['IVD']==7) & (df['IOTHER']==7)
```

‘Alzheimer’s disease with cerebrovascular disease’:

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==7) & (df['IAD1']==1) & (df['IVD']==7) & (df['IOTHER']==7)
```

‘Vascular dementia’：

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==7) & (df['IAD1']==7) & (df['IVD']==1) & (df['IOTHER']==7)
```

‘Other neurological disorder’:

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==7) & (df['IAD1']==7) & (df['IVD']==7) & (df['IOTHER']==1)
```

‘No neurological disorder’:

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==6) & (df['IAD1']==6) & (df['IVD']==6) & (df['IOTHER']==6) &
(df['AD']==0) & (df['AD1']==0) & (df['VD']==0) & (df['OTHER']==0)
```

‘Alzheimer’s disease without cerebrovascular disease’:

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==6) & (df['IAD1']==6) & (df['IVD']==6) & (df['IOTHER']==6) &
(df['AD']==1) & (df['AD1']==1) & (df['VD']==7) & (df['OTHER']==7)
```

‘Alzheimer’s disease with cerebrovascular disease’:

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==6) & (df['IAD1']==6) & (df['IVD']==6) & (df['IOTHER']==6) &
(df['AD']==7) & (df['AD1']==1) & (df['VD']==7) & (df['OTHER']==7)
```

‘Vascular dementia’：

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==6) & (df['IAD1']==6) & (df['IVD']==6) & (df['IOTHER']==6) &
(df['AD']==7) & (df['AD1']==7) & (df['VD']==1) & (df['OTHER']==7)
```

‘Other neurological disorder’:

```
(df['IIAD']==9) & (df['IIAD1']==9) & (df['IIVAD']==9) & (df['IIOTHER']==9)
& (df['IAD']==6) & (df['IAD1']==6) & (df['IVD']==6) & (df['IOTHER']==6) &
(df['AD']==7) & (df['AD1']==7) & (df['VD']==7) & (df['OTHER']==1)
```

‘No neurological disorder’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD']==0) &
(df['IAD1']==0) & (df['IVD']==0) & (df['IOTHER']==0)
```

‘Alzheimer’s disease without cerebrovascular disease’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD']==1) &
(df['IAD1']==1) & (df['IVD']==7) & (df['IOTHER']==7)
```

‘Alzheimer’s disease with cerebrovascular disease’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD']==7) &
(df['IAD1']==1) & (df['IVD']==7) & (df['IOTHER']==7)
```

‘Vascular dementia’：

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD']==7) &
(df['IAD1']==7) & (df['IVD']==1) & (df['IOTHER']==7)
```

‘Other neurological disorder’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD']==7) &
(df['IAD1']==7) & (df['IVD']==7) & (df['IOTHER']==1)
```

‘No neurological disorder’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD'].isnull())
& (df['IAD1'].isnull()) & (df['IVD'].isnull()) & (df['IOTHER'].isnull()) &
(df['AD']==0) & (df['AD1']==0) & (df['VD']==0) & (df['OTHER']==0)
```

‘Alzheimer’s disease without cerebrovascular disease’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD'].isnull())
& (df['IAD1'].isnull()) & (df['IVD'].isnull()) & (df['IOTHER'].isnull()) &
(df['AD']==1) & (df['AD1']==1) & (df['VD']==7) & (df['OTHER']==7)
```

‘Alzheimer’s disease with cerebrovascular disease’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD'].isnull())
& (df['IAD1'].isnull()) & (df['IVD'].isnull()) & (df['IOTHER'].isnull()) &
(df['AD']==7) & (df['AD1']==1) & (df['VD']==7) & (df['OTHER']==7)
```

‘Vascular dementia’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD'].isnull())
& (df['IAD1'].isnull()) & (df['IVD'].isnull()) & (df['IOTHER'].isnull()) &
(df['AD']==7) & (df['AD1']==7) & (df['VD']==1) & (df['OTHER']==7)
```

‘Other neurological disorder’:

```
(df['IIAD'].isnull()) & (df['IIAD1'].isnull()) & (df['IIVAD'].isnull()) &
(df['IIOTHER'].isnull()) & (df['IIOTHER'].isnull()) & (df['IAD'].isnull())
& (df['IAD1'].isnull()) & (df['IVD'].isnull()) & (df['IOTHER'].isnull()) &
(df['AD']==7) & (df['AD1']==7) & (df['VD']==7) & (df['OTHER']==1)
```

- **diagnosis_change** - Had the diagnosis changed during the follow-up?

If ‘latest_diagnosis’ is the same as ‘primary_diagnosis, the value is ‘No’
If ‘latest_diagnosis is not the same as ‘primary_diagnosis’, the value is ‘Yes’

```
if df['latest_diagnosis'] == df['primary_diagnosis']:
  return 'No'
elif df['latest_diagnosis'] != df['primary_diagnosis']:
  return 'Yes'
```

```
else:
  return np.nan
```

haas_long:

- **participant_id** - Study-defined de-identified numeric participant identifier

Renamed the HAAS 'NIA_ID' variable to 'participant_id'

- **visit_name** - Study-defined visit name

HAAS began at exam 4 of the Honolulu Heart Program (HHP) study.

Use 'exam 4' for exam 4 of HHP study
Use 'exam 5' for exam 5 of HHP study
Use 'exam 6' for exam 6 of HHP study

- **visit_month** - month from baseline visit (longitudinal study. Negative value is possible for screening visits), rounded integer

Subtracted date of exam 4 from date of visit, and rounded the resulting number of months to nearest integer

```
df['visit_month'] = (((df['visit_date_unix']-
df['datex4'])/365.25)*12).round().astype('Int64')
```

- **mmse_version** - MMSE version used

MMSE total score was calculated from CASI score, so the value enter for each participant is 'CASI derived'

- **mmse_total_score** - MMSE score

The value for exam 4 is the same value as the HAAS MMSEX4 variable.
The value for exam 5 is the same value as the HAAS MMSEX5 variable.
The value for exam 6 is the same value as the HAAS MMSEX6 variable.

```
if (df['visit_name'] == 'exam 4'):
  return df['MMSEX4']
```

```
 elif (df['visit_name'] == 'exam 5'):
   return df['MMSEX5']
 elif (df['visit_name'] == 'exam 6'):
   return df['MMSEX6']
 else:
   return np.nan
```

- **mmmse_total_score** - MMMSE (3MS) score

The value used for exam 4 is the same value as the HAAS MMMSEX4 variable.
The value used for exam 5 is the same value as the HAAS MMMSEX5 variable.
The value used for exam 6 is the same value as the HAAS MMMSEX6 variable.

```
if (df['visit_name'] == 'exam 4'):
   return df['MMMSEX4']
 elif (df['visit_name'] == 'exam 5'):
   return df['MMMSEX5']
 elif (df['visit_name'] == 'exam 6'):
   return df['MMMSEX6']
 else:
   return np.nan
```