/\* Impact of Disabilities on Mortality \*/

Use Disabilities

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 \* Cleaning Data \*

 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

-- Identify who has dead at last assessment

DROP TABLE #EverDead

SELECT ID, MAX(Alive) as EverDead

INTO #EverDead

FROM [dbo].[Data]

GROUP BY ID

-- (265530 unique individuals)

-- Concatinate comorbidities

-- All disabled when dead

-- Calcualte if dead in 6 months from assessment

-- Drop last assessment

-- Age assessment= age first assessment + days to assessment/365

DROP TABLE [dbo].[Concatenate], #Concatenate

SELECT [dbo].[data].ID, AssessmentID, DayLast AS [Days], Cast(Age as float) as Age

-- Alive in 6 months if never dead or dead after 180 days

 , CASE WHEN EverDead=0 THEN 0.

 WHEN EverDead=1 and Daylast > 0 and cast(DayLast as Float)<=180 THEN 1.

 WHEN EverDead=1 and DayLast> 0 and cast(DayLast as Float)>180 THEN 0.

 ELSE Null -- no 6 months outcomes are available on last assessment

 END AS Dead

 , CASE WHEN Sex='M' THEN '1' ELSE '0' END

 + CASE WHEN uEat=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN uSit=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN uGroom=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN uToilet=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN uBathe=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN uUrine=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN uWalk=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN uDress=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN uBowel=1 THEN '1' WHEN Alive='1' THEN '1' ELSE '0' END

 + CASE WHEN (Cast(Age as Float)+ CAST(DayFirst as float)/365.) >74.

 THEN '1' ELSE '0' END AS AllVariables

INTO #Concatenate

FROM [dbo].[Data] inner join #EverDead ON [dbo].[Data].ID=#EverDead.ID

WHERE Age>19 and DayLast>0 -- drop negative age and last assessment

Go

SELECT \* INTO dbo.[Concatenate] FROM #Concatenate

-- 1,047,117 disability assessments not counting the last one

-- Prepare tables for output of the analysis results

DROP TABLE #Intercept, #MatchedSaved

Create Table #Intercept ([Variable] int, [Intercept] float, [Overlap] float, [Cases Matched] Float)

Create Table #MatchedSaved (Variable int, [Prob Control] float, [Prob Cases] float

 , [Number Controls] int, [Number Cases] int, [Strata] text)

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 \* Balance & Analyze Data \*

 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

-- Start an index & repeat for each variable

DECLARE @Index INT

SET @index = 1

WHILE (@Index <=11)

BEGIN

-- Calculate mortality for cases

Drop Table #Cases

Select SUM(Dead) as cDead -- number dead in cases

, SUM(dead)+SUM(1-dead) AS cCases -- number of cases

-- Set strata to all variables except case/control variable

, STUFF(AllVariables,@Index,1, '\_') AS cStrata

INTO #Cases

FROM [dbo].[Concatenate]

-- Set cases to variable number @index being 1

WHERE SUBSTRING(AllVariables, @Index, 1)='1'

-- Group by all variables except the case/control variable

GROUP BY STUFF(AllVariables,@Index,1, '\_')

-- Calculate mortality for controls

Drop Table #Controls

Select SUM(dead) as mDead -- number dead in controls

, SUM(Dead)+SUM(1-Dead) AS mCases -- number of controls

-- set strata to all variables except case/control variable

, STUFF(Allvariables,@Index,1, '\_') AS mStrata

INTO #Controls

FROM [dbo].[Concatenate]

-- Set control to variable number @index being not present

WHERE SUBSTRING(AllVariables, @Index, 1) <> '1'

-- Group by all variables except the case/control variable

GROUP BY STUFF(AllVariables,@Index,1, '\_')

-- Match cases and controls

Drop Table #matched

Select @Index as Variable

, Round(CAST(mDead as float)/CAST(mCases as Float),2) AS mProb

, Round(CAST(cDead as float)/CAST(cCases as Float),2) AS cProb

, cCases, mCases

, cStrata AS Strata

INTO #matched

FROM #cases inner join #controls on cStrata = mStrata

WHERE cCases>9 and mCases>9

-- Save matched strata, if you wish to have access to the data

 --INSERT INTO #MatchedSaved

SELECT \* FROM #Matched -- print data

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 \* Calculate Intercept & Overlap \*

 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

Declare @Cases as Float

SET @Cases = (SELECT SUM(CAST(cCases as FLOAT)) FROM #Cases)

Insert INTO #Intercept -- Save into temporary file

Select Max(@index) As [Variable Number]

, (SUM(cProb)\*SUM(mProb\*mProb)-SUM(mProb)\*SUM(cProb\*mProb))

 / (COUNT(mProb)\*SUM(mProb\*mProb)-SUM(mProb)\*SUM(mProb)) AS Intercept

, SUM(CAST(cCases AS Float))/@Cases AS Overlap

, SUM(cCases) AS [Cases Matched]

FROM #matched

SET @Index = @Index + 1

END

GO

SELECT \* FROM #Intercept

--SELECT \* FROM #MatchedSaved