SAS code for “Sample composition alters associations between age and brain structure”

Data are available by request from Katherine Keyes ([kmk2104@columbia.edu](mailto:kmk2104@columbia.edu))

libname pums '/pums';

libname kneuro '/neurok';

\*\*\*\*\*\*\*\*\*\*\*\*\*\* PUMS data - get census level data;

\*household data;

**data** pumsHa;

set pums.psam\_husa;

**run**;

**data** pumsHb;

set pums.psam\_husb;

**run**;

**data** pumsHc;

set pums.psam\_husc;

**run**;

**data** pumsHd;

set pums.psam\_husd;

**run**;

**data** pumsHouse;

set pumsHa pumsHb pumsHc pumsHd;

**run**;

\*individual level data;

**data** pumsa;

set pums.psam\_pusa (keep = RELP RC SFR serialno hisp rac1p SEMP sex sch schg schl AGEP WAGP PWGTP1 PWGTP);

**run**;

**data** pumsb;

set pums.psam\_pusb (keep = RELP RC SFR serialno hisp rac1p SEMP sex sch schg schl AGEP WAGP PWGTP1 PWGTP);

**run**;

**data** pumsc;

set pums.psam\_pusc (keep = RELP RC SFR serialno hisp rac1p SEMP sex sch schg schl AGEP WAGP PWGTP1 PWGTP);

**run**;

**data** pumsd;

set pums.psam\_pusd (keep = RELP RC SFR serialno hisp rac1p SEMP sex sch schg schl AGEP WAGP PWGTP1 PWGTP);

**run**;

\*individual combined file;

**Data** PUMS;

set pumsa pumsb pumsc pumsd;

\*creating hispanic ethnicity;

hisp2 = hisp \***1**;

if HISP2 >**1** then hispeth =**2**; else hispeth =**1**;

\*age group we want to look at. WE have ages 3-18 in PING;

age = agep\***1**;

if **3**<=age <**7** then agecat = **1**;

else if **7**<=age <**11** then agecat = **2**;

else if **11**<=age <**15** then agecat = **3**;

else if **15**<=age <**19** then agecat = **4**;

**run**;

**proc** **sort** data = pumsHouse; by serialno; **run**;

**proc** **sort** data = PUMS; by serialno; **run**;

\*combining individual and household level files;

**data** combPUM(keep = AGEP HINCP HISP HUPAC NOC NP PUMA

RAC1P hispeth RC REGION SCH SCHG SCHL SEMP SEX

SFR WAGP age agecat houseInc relp serialno IDnew kidof refperson kidSerial kidSerial2

refserial parSerial parentof kidrefper PotentialParSer PotentialPar pweight pWeight1 houseweight

income3cat education2);

merge PUMS pumshouse; by serialno;

\*creating new unique ID;

IDNewCh=\_n\_;

IDnew = IDNewCh\***1**;

\*steps taken to clarify family. Doing this to get parental education information for the children;

if relp in (**2**,**3**,**4**) then kidOf = **1**; else kidof = **0**; \*relp = relationship to reference person. 2/3/4 are children in the household;

if relp = **0** then refPerson = **1**; else refPerson = **0**; \*relp =0 is reference person;

if kidof = **1** then kidSerial = serialno;

if refperson = **1** then refSerial = serialno;

\*if reference is between 3-18: ;

if agecat ~=**.** and refperson = **1** then kidRefPer = **1**; else kidrefper = **0**;

if relp =**6** and kidrefper = **1** then ParentOf = **1**; else parentof = **0**;

if parentof = **1** then parSerial = serialno;

if kidof = **1** or kidrefper = **1** then kidSerial2 = serialno; \*giving child the family's ID;

if parentof =**1** or refperson = **1** then PotentialParSer = serialno; \*giving potential parent the family's ID;

if parentof =**1** or refperson = **1** then PotentialPar = **1**; else potentialpar = **0**; \*creating variable for a potential parent;

\*survey weights;

pWeight = PWGTP\***1**;

pWeight1 = PWGTP1\***1**;

houseweight = WGTP\***1**;

\*highest education level each respondent received, 4 groups: High school or less, some college, college degree, more than 4-year college;

if **0**< schl<**18** then education2 = **1**;

else if schl in (**18**,**19**,**20**) then education2 =**2**;

else if schl =**21** then education2 = **3**;

else if **21**<schl<=**24** then education2 = **4**;

\*household income variable split into three groups, <40k, 40k-100k, >=100k;

houseinc = HINCP\***1**;

if **0**<=houseInc<**40000** then Income3cat =**1**;

else if **40000**<=houseInc<**100000** then income3cat =**2**;

else if houseInc>=**100000** then income3cat =**3**;

**run**;

\*use SQL to connect parents to children in same family/household of PUMS;

**proc** **sql**;

create table PumsWithEd AS

select distinct \*

from combPUM join (

select distinct serialno as ParSerialno, education2 AS parentEd, idnew AS parIDNEW

from combPUM

where PotentialPar =**1**)

on (kidSerial2 = parserialno);

**quit**;

\*calculated the number of people in each home;

**proc** **sql**;

create table NumInHouse AS

select count(serialno), idnew

from combPUM

group by serialno;

**quit**;

\*creating variable in dataset that specifies parental/child connection has occured;

**data** PumsWithEd;

set PumsWithEd;

inEd = **1**;

**run**;

**proc** **sort** data = PumsWithEd; by idnew;

**proc** **sort** data = NumInHouse; by idnew;

**data** pumswithed2;

merge PumsWithEd numinhouse; by idnew;

if ined =**.** then delete; \*If there is not a household/family in PUMS with a parent-child then it is deleted;

if agecat = **.** then delete;

if hispeth = **2** then HispCorr = **1**; else if hispeth = **1** then hispCorr = **2**;

if rac1p = **1** then cenRace = **1**; \*white;

else if rac1p =**2** then cenRace = **2**; \*black;

else if hispCorr = **1** then cenRace = **3**; \*hispanic;

else if rac1p = **9** then cenRace = **5**; \*two or more races;

else if rac1p in (**3**,**4**,**5**,**6**,**7**,**8**) then cenRace = **4**; \*other race;

\*create income variable with a category for missing on income response;

incomemiss= income3cat;

if income3cat = **.** then incomemiss = **10**;

**run**;

\*values for creating a census weight;

**proc** **freq** data = pumswithed2;

weight pweight;

table sex cenrace agecat parented income3cat incomemiss; **run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PING DATA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** nuse;

set kneuro.PING\_data\_AHedits;

\*dummy variables;

weight1 = **1**;

nest1 = **1**;

HUH =**1**;

\*round age;

Rage = round(age, **1**);

\*4 age categories;

if **3**<=rage <**7** then ragecat = **1**;

else if **7**<=rage <**11** then ragecat = **2**;

else if **11**<=rage <**15** then ragecat = **3**;

else if **15**<=rage <**19** then ragecat = **4**;

\*race renames;

hispeth = FDH\_1\_Hispanic\_or\_Latino\***1**;

PacificIslander = FDH\_2\_Pacific\_Islander\***1**;

asian = FDH\_2\_Asian\***1**;

black = FDH\_2\_African\_American\***1**;

nativeAmerican = FDH\_2\_American\_Indian\***1**;

white = FDH\_2\_White\***1**;

raceSum = sum(white, black, nativeamerican, asian, PacificIslander); \*2+ races;

\*final race variable;

if hispeth = **0** and asian = **0** and black =**0** and nativeAmerican=**0** and PacificIslander = **0** and white = **1** then CenRace = **1**; \*white only;

else if hispeth= **0** and asian = **0** and black =**1** and nativeAmerican=**0** and PacificIslander = **0** and white = **0** then cenRace =**2**; \*black only;

else if hispeth = **1** then cenRace = **3**; \*hispanic;

else if hispeth = **0** and raceSum>=**2** then cenRace = **5**; \*more than one race;

else if hispeth = **0** and asian = **1** or pacificIslander = **1** or nativeamerican=**1** then cenrace = **4**; \*asian/pacific Islander/native american;

\*highest education level parent 1 received, 4 groups: High school or less, some college, college degree, more than 4-year college;

if FDH\_Guardian\_1\_Edu in (**1**,**2**,**3**,**4**) then Par1Ed4cat = **1**;

else if FDH\_Guardian\_1\_Edu in (**5**) then Par1Ed4cat = **2**;

else if FDH\_Guardian\_1\_Edu =**6** then Par1Ed4cat =**3**;

else if FDH\_Guardian\_1\_Edu in (**7**) then Par1Ed4cat =**4**;

\*highest education level parent 2 received, 4 groups: High school or less, some college, college degree, more than 4-year college;

if FDH\_Guardian\_2\_Edu in (**1**,**2**,**3**,**4**) then Par2Ed4cat = **1**;

else if FDH\_Guardian\_2\_Edu in(**5**) then Par2Ed4cat = **2**;

else if FDH\_Guardian\_2\_Edu in(**6**) then Par2Ed4cat = **3**;

else if FDH\_Guardian\_2\_Edu in (**7**) then Par2Ed4cat =**4**;

\*parhighEd3 = highest parental education between the two;

if Par1Ed4cat>=Par2Ed4cat then parHighEd3 = Par1Ed4cat;

else if Par2Ed4cat > Par1Ed4cat then parHighEd3 = Par2Ed4cat;

\*household income for child;

if FDH\_3\_Household\_Income in (**1**,**2**,**3**,**4**,**5**) then income3cat = **1**;

else if FDH\_3\_Household\_Income in (**6**,**7**) then income3cat =**2**;

else if FDH\_3\_Household\_Income>**7** then income3cat =**3**;

\*include missing income as a category;

incomemiss = income3cat;

if income3cat = **.** then incomemiss = **10**;

**run**;

\*if missing on race or not in ages from 3-18, then delete;

**data** NWuse2;

set nuse;

if ragecat = **.** then delete;

if cenrace = **.** then delete;

if parhighed3 = **.** then delete;

**run**;

\*Weighting Code;

**PROC** **WTADJUST** DATA=NWuse2 ADJUST=post NOTSORTED;

WEIGHT weight1;

nest ragecat;

CLASS cenrace gender parhighed3 incomemiss;

MODEL huh=cenrace gender parhighed3 incomemiss;

WTMAX **80000**;

WTMIN **15**;

LOWERBD **15**;

CENTER **42700**;

UPPERBD **80000**;

POSTWGT

/\* total \*/ **58806391**

/\* race \*/ **41100215** **7974668** **4419405** **3225478** **2086625**

/\* gender \*/**30138428** **28667963**

/\* parented\*/ **21889657** **19173950** **11100192** **6642592**

/\* income\*/ **19691520** **23832118** **15260552** **22201**

;

IDVAR subjid ragecat;

OUTPUT IDVAR wtfinal/filename=mod\_weights5

predicted=all replace;

**run**;

**proc** **sort** data = mod\_weights5; by subjid;

**proc** **sort** data = nwuse2; by subjid;

**data** weightedNeuro;

merge mod\_weights5 (keep = wtfinal subjid) nwuse2; by subjid;

ageSq = rAge\*rAge;

\*scanner system/location that was used to scann child;

if Manufacturer = "GE MEDICAL SYSTEMS" and ManufacturersModelName = "DISCOVERY MR750" then scanner5c = **1**;

else if Manufacturer = "GE MEDICAL SYSTEMS" and ManufacturersModelName = "SIGNA HDx" then scanner5c = **2**;

else if Manufacturer = "Philips Medical Systems" and ManufacturersModelName = "Achieva" then scanner5c = **3**;

else if Manufacturer = "SIEMENS" and ManufacturersModelName = "TrioTim" then scanner5c = **4**;

else scanner5c = **5**;

if cenrace =**.** then delete;

if parhighed3 =**.** then delete;

**run**;

\*table 1 percentages;

**proc** **freq** data = weightedNeuro;

table cenrace gender ragecat parhighed3 income3cat;

**run**;

**proc** **freq** data = weightedNeuro;

weight wtfinal;

table cenrace gender ragecat parhighed3 income3cat;

**run**;

**proc** **sort** data = weightedNeuro; by subjid;

**proc** **sort** data = kneuro.CC\_insula\_measures; by subjid;

**proc** **sort** data = kneuro.PING\_AddedValues\_10\_06; by subjid;

\*adding in other brain regions;

**data** wneuro;

merge weightedNeuro kneuro.CC\_insula\_measures kneuro.PING\_AddedValues\_10\_06; by subjid;

if wtfinal = **.** then delete;

**run**;

\*get mean of age to center age variables;

**proc** **means** data = wneuro;

var rage;

**run**;

\*get weighted mean of age to center age variables;

**proc** **means** data = wneuro;

weight wtfinal;

var rage;

**run**;

\*\*\*\*\*\*\*;

**data** wneuro\_safety;

set wneuro;

\*control for scanner in models, so only keep those observations with scanner;

if scanner5c = **5** then scanner5c =**.**;

\*age category for 2-year age groups;

if rage in (**3**,**4**) then ragecat = **1**;

else if rage in (**5**,**6**) then ragecat = **2**;

else if rage in (**7**,**8**) then ragecat = **3**;

else if rage in (**9**,**10**) then ragecat = **4**;

else if rage in (**11**,**12**) then ragecat = **5**;

else if rage in (**13**,**14**) then ragecat = **6**;

else if rage in (**15**,**16**) then ragecat = **7**;

else if rage in (**17**,**18**) then ragecat = **8**;

\*use ragecat for 4-category;

if **3**<=rage <**7** then ragecat2 = **1**;

else if **7**<=rage <**11** then ragecat2 = **2**;

else if **11**<=rage <**15** then ragecat2 = **3**;

else if **15**<=rage <**19** then ragecat2 = **4**;

\*Get age, age squared, and age cubed of mean centered values for unweighted values;

meanAge = **10.4475043**;

Age\_mean = Rage-meanage;

Age\_meansq = age\_mean\*age\_mean;

Age\_meancu = age\_mean\*age\_mean\*age\_mean;

\*Get age, age squared, and age cubed of mean centered values for weighted values;

meanAgeW = **10.8460994**;

AgeW\_mean = RAge-meanAgeW;

AgeW\_meansq = AgeW\_mean\*AgeW\_mean;

AgeW\_meancu = AgeW\_mean\*AgeW\_mean\*AgeW\_mean;

\*Frontal Lobe;

\*Frontal Surface Area, lefthand side;

frontalSA\_LH = sum(MRI\_cort\_area\_ctx\_lh\_superiorfro,

MRI\_cort\_area\_ctx\_lh\_rostralmidd,MRI\_cort\_area\_ctx\_lh\_caudalmiddl,

MRI\_cort\_area\_ctx\_lh\_parsopercul,MRI\_cort\_area\_ctx\_lh\_parstriangu,

MRI\_cort\_area\_ctx\_lh\_parsorbital,MRI\_cort\_area\_ctx\_lh\_lateralorbi,

MRI\_cort\_area\_ctx\_lh\_medialorbit,MRI\_cort\_area\_ctx\_lh\_precentral,

MRI\_cort\_area\_ctx\_lh\_paracentral,MRI\_cort\_area\_ctx\_lh\_frontalpole,

MRI\_cort\_area\_ctx\_lh\_rostralante,MRI\_cort\_area\_ctx\_lh\_caudalanter);

\*Frontal Surface Area, righthand side;

frontalSA\_RH = sum(MRI\_cort\_area\_ctx\_RH\_superiorfro,

MRI\_cort\_area\_ctx\_RH\_rostralmidd,MRI\_cort\_area\_ctx\_RH\_caudalmiddl,

MRI\_cort\_area\_ctx\_RH\_parsopercul,MRI\_cort\_area\_ctx\_RH\_parstriangu,

MRI\_cort\_area\_ctx\_RH\_parsorbital,MRI\_cort\_area\_ctx\_RH\_lateralorbi,

MRI\_cort\_area\_ctx\_RH\_medialorbit,MRI\_cort\_area\_ctx\_RH\_precentral,

MRI\_cort\_area\_ctx\_RH\_paracentral,MRI\_cort\_area\_ctx\_RH\_frontalpole,

MRI\_cort\_area\_ctx\_RH\_rostralante,MRI\_cort\_area\_ctx\_rh\_caudalanter);

\*Frontal Thickness, lefthand side;

frontalTH\_LH = mean(MRI\_cort\_thick\_ctx\_lh\_superiorfr,

MRI\_cort\_thick\_ctx\_lh\_rostralmid,MRI\_cort\_thick\_ctx\_lh\_caudalmidd,

MRI\_cort\_thick\_ctx\_lh\_parsopercu,MRI\_cort\_thick\_ctx\_lh\_parstriang,

MRI\_cort\_thick\_ctx\_lh\_parsorbita,MRI\_cort\_thick\_ctx\_lh\_lateralorb,

MRI\_cort\_thick\_ctx\_lh\_medialorbi,MRI\_cort\_thick\_ctx\_lh\_precentral,

MRI\_cort\_thick\_ctx\_lh\_paracentra,MRI\_cort\_thick\_ctx\_lh\_frontalpol,

MRI\_cort\_thick\_ctx\_lh\_rostralant,MRI\_cort\_thick\_ctx\_lh\_caudalante);

\*Frontal Thickness, righthand side;

frontalTH\_RH = mean(MRI\_cort\_thick\_ctx\_RH\_superiorfr,

MRI\_cort\_thick\_ctx\_RH\_rostralmid,MRI\_cort\_thick\_ctx\_RH\_caudalmidd,

MRI\_cort\_thick\_ctx\_RH\_parsopercu,MRI\_cort\_thick\_ctx\_RH\_parstriang,

MRI\_cort\_thick\_ctx\_RH\_parsorbita,MRI\_cort\_thick\_ctx\_RH\_lateralorb,

MRI\_cort\_thick\_ctx\_RH\_medialorbi,MRI\_cort\_thick\_ctx\_RH\_precentral,

MRI\_cort\_thick\_ctx\_RH\_paracentra,MRI\_cort\_thick\_ctx\_RH\_frontalpol,

MRI\_cort\_thick\_ctx\_RH\_rostralant,MRI\_cort\_thick\_ctx\_rh\_caudalante);

\*Parietal Lobe;

\*Parietal Surface Area, lefthand side;

parietalSA\_LH = sum(MRI\_cort\_area\_ctx\_lh\_superiorpar,

MRI\_cort\_area\_ctx\_lh\_inferiorpar,MRI\_cort\_area\_ctx\_lh\_supramargin,

MRI\_cort\_area\_ctx\_lh\_postcentral,MRI\_cort\_area\_ctx\_lh\_precuneus,

MRI\_cort\_area\_ctx\_lh\_posteriorci,MRI\_cort\_area\_ctx\_lh\_isthmuscing);

\*Parietal Surface Area, righthand side;

parietalSA\_RH = sum(MRI\_cort\_area\_ctx\_RH\_superiorpar,

MRI\_cort\_area\_ctx\_RH\_inferiorpar,MRI\_cort\_area\_ctx\_RH\_supramargin,

MRI\_cort\_area\_ctx\_RH\_postcentral,MRI\_cort\_area\_ctx\_RH\_precuneus,

MRI\_cort\_area\_ctx\_RH\_posteriorci,MRI\_cort\_area\_ctx\_RH\_isthmuscing);

\*Parietal Thickness, lefthand side;

parietalTH\_LH = mean(MRI\_cort\_thick\_ctx\_lh\_superiorpa,

MRI\_cort\_thick\_ctx\_lh\_inferiorpa,MRI\_cort\_thick\_ctx\_lh\_supramargi,

MRI\_cort\_thick\_ctx\_lh\_postcentra,MRI\_cort\_thick\_ctx\_lh\_precuneus,

MRI\_cort\_thick\_ctx\_lh\_posteriorc,MRI\_cort\_thick\_ctx\_lh\_isthmuscin);

\*Parietal Thickness, righthand side;

parietalTH\_RH = mean(MRI\_cort\_thick\_ctx\_RH\_superiorpa,

MRI\_cort\_thick\_ctx\_RH\_inferiorpa,MRI\_cort\_thick\_ctx\_RH\_supramargi,

MRI\_cort\_thick\_ctx\_RH\_postcentra,MRI\_cort\_thick\_ctx\_RH\_precuneus,

MRI\_cort\_thick\_ctx\_RH\_posteriorc,MRI\_cort\_thick\_ctx\_RH\_isthmuscin);

\*Temporal Lobe;

\*Temporal Surface Area, lefthand side;

temporalSA\_LH = sum(MRI\_cort\_area\_ctx\_lh\_superiortem,

MRI\_cort\_area\_ctx\_lh\_middletempo,MRI\_cort\_area\_ctx\_lh\_inferiortem,

MRI\_cort\_area\_ctx\_lh\_bankssts,MRI\_cort\_area\_ctx\_lh\_fusiform,

MRI\_cort\_area\_ctx\_lh\_transverset,MRI\_cort\_area\_ctx\_lh\_entorhinal,

MRI\_cort\_area\_ctx\_lh\_temporalpol,MRI\_cort\_area\_ctx\_lh\_parahippoca);

\*Temporal Surface Area, righthand side;

temporalSA\_RH = sum(MRI\_cort\_area\_ctx\_RH\_superiortem,

MRI\_cort\_area\_ctx\_RH\_middletempo,MRI\_cort\_area\_ctx\_RH\_inferiortem,

MRI\_cort\_area\_ctx\_RH\_bankssts,MRI\_cort\_area\_ctx\_RH\_fusiform,

MRI\_cort\_area\_ctx\_RH\_transverset,MRI\_cort\_area\_ctx\_RH\_entorhinal,

MRI\_cort\_area\_ctx\_RH\_temporalpol,MRI\_cort\_area\_ctx\_RH\_parahippoca);

\*Temporal Thickness, lefthand side;

temporalTH\_LH = mean(MRI\_cort\_thick\_ctx\_lh\_superiorte,

MRI\_cort\_thick\_ctx\_lh\_middletemp,MRI\_cort\_thick\_ctx\_lh\_inferiorte,

MRI\_cort\_thick\_ctx\_lh\_bankssts,MRI\_cort\_thick\_ctx\_lh\_fusiform,

MRI\_cort\_thick\_ctx\_lh\_transverse,MRI\_cort\_thick\_ctx\_lh\_entorhinal,

MRI\_cort\_thick\_ctx\_lh\_temporalpo,MRI\_cort\_thick\_ctx\_lh\_parahippoc);

\*Temporal Thickness, righthand side;

temporalTH\_RH = mean(MRI\_cort\_thick\_ctx\_RH\_superiorte,

MRI\_cort\_thick\_ctx\_RH\_middletemp,MRI\_cort\_thick\_ctx\_RH\_inferiorte,

MRI\_cort\_thick\_ctx\_RH\_bankssts,MRI\_cort\_thick\_ctx\_RH\_fusiform,

MRI\_cort\_thick\_ctx\_RH\_transverse,MRI\_cort\_thick\_ctx\_RH\_entorhinal,

MRI\_cort\_thick\_ctx\_RH\_temporalpo,MRI\_cort\_thick\_ctx\_RH\_parahippoc);

\*Occipital Lobe;

\*Occipital Surface Area, lefthand side;

occipitalSA\_LH = sum(MRI\_cort\_area\_ctx\_lh\_lateralocci,MRI\_cort\_area\_ctx\_lh\_lingual,

MRI\_cort\_area\_ctx\_lh\_cuneus,MRI\_cort\_area\_ctx\_lh\_pericalcari);

\*Occipital Surface Area, righthand side;

occipitalSA\_RH = sum(MRI\_cort\_area\_ctx\_RH\_lateralocci,MRI\_cort\_area\_ctx\_RH\_lingual,

MRI\_cort\_area\_ctx\_RH\_cuneus,MRI\_cort\_area\_ctx\_RH\_pericalcari);

\*Occipital Thickness, lefthand side;

occipitalTH\_LH = sum(MRI\_cort\_thick\_ctx\_lh\_lateralocc,MRI\_cort\_thick\_ctx\_lh\_lingual,

MRI\_cort\_thick\_ctx\_lh\_cuneus,MRI\_cort\_thick\_ctx\_lh\_pericalcar);

\*Occipital Thickness, rightthand side;

occipitalTH\_RH = sum(MRI\_cort\_thick\_ctx\_RH\_lateralocc,MRI\_cort\_thick\_ctx\_RH\_lingual,

MRI\_cort\_thick\_ctx\_RH\_cuneus,MRI\_cort\_thick\_ctx\_RH\_pericalcar);

\*\*\*\*\* sum for subcortal total volume;

\*Left Subcortal Volume;

leftSubcort\_vol= sum(MRI\_subcort\_vol\_Left\_Thalamus\_Pr, MRI\_subcort\_vol\_Left\_Caudate, MRI\_subcort\_vol\_Left\_Putamen,

MRI\_subcort\_vol\_Left\_Pallidum, MRI\_subcort\_vol\_Left\_Hippocampus, MRI\_subcort\_vol\_Left\_Amygdala, MRI\_subcort\_vol\_Left\_Accumbens\_a);

\*Right Subcortal Volume;

rightSubcort\_vol= sum(MRI\_subcort\_vol\_Right\_Thalamus\_P, MRI\_subcort\_vol\_Right\_Caudate, MRI\_subcort\_vol\_Right\_Putamen,

MRI\_subcort\_vol\_Right\_Pallidum, MRI\_subcort\_vol\_Right\_Hippocampu, MRI\_subcort\_vol\_Right\_Amygdala, MRI\_subcort\_vol\_Right\_Accumbens\_);

\*Total Subcortal Volume;

totalSubCort\_vol =sum(leftSubcort\_vol,rightSubcort\_vol);

\*Hippocampus (bilateral) Volume;

totalHippocam\_vol = sum(MRI\_subcort\_vol\_Left\_Hippocampus, MRI\_subcort\_vol\_Right\_Hippocampu);

\*Amygdala (bilateral) Volume;

totalAmygdala\_vol = sum(MRI\_subcort\_vol\_Left\_Amygdala, MRI\_subcort\_vol\_Right\_Amygdala);

\*Basal Ganglia (bilateral) Volume;

totalBasalGang\_vol = sum(MRI\_subcort\_vol\_Left\_Caudate, MRI\_subcort\_vol\_Right\_Caudate,

MRI\_subcort\_vol\_Left\_Putamen,MRI\_subcort\_vol\_Right\_Putamen,MRI\_subcort\_vol\_Left\_Pallidum,

MRI\_subcort\_vol\_Right\_Pallidum, MRI\_subcort\_vol\_Left\_Accumbens\_a, MRI\_subcort\_vol\_Right\_Accumbens\_);

**run**;

\*This macro outputs the linear, quadratic, and cubic unweighted and weighted models

for thickness and surface area brain regions;

**%macro** part1run\_Safe(inputb, title);

\*Unweighted linear;

proc genmod data = wneuro\_safety;

model &inputB = Age\_mean ;

title &inputB;

run;

\*weighted linear;

proc genmod data = wneuro\_safety;

weight wtfinal;

model &inputB = Agew\_mean ;

title &inputB;

run;

\*Unweighted quadratic;

proc genmod data = wneuro\_safety;

model &inputB = Age\_mean Age\_meansq ;

title &inputB;

run;

\*Weighted quadratic;

proc genmod data = wneuro\_safety;

weight wtfinal;

model &inputB = Agew\_mean Agew\_meansq ;

title &inputB;

run;

\*Unweighted cubic;

proc genmod data = wneuro\_safety;

model &inputB = Age\_mean Age\_meansq Age\_meancu ;

title &inputB;

run;

\*Weighted cubic;

proc genmod data = wneuro\_safety;

weight wtfinal;

model &inputB = Agew\_mean Agew\_meansq Agew\_meancu ;

title &inputB;

run;

**%mend**;

\*Thickness measures;

%***part1run\_safe***(MRI\_cort\_thick\_ctx\_mean, Thickness: Total Mean);

%***part1run\_safe***(MRI\_cort\_thick\_ctx\_lh\_mean, Thickness: LH Mean);

%***part1run\_safe***(MRI\_cort\_thick\_ctx\_rh\_mean, Thickness: RH Mean);

%***part1run\_safe***(occipitalTH\_LH, TH: occipital LH);

%***part1run\_safe***(occipitalTH\_RH, TH: occipital RH);

%***part1run\_safe***(temporalTH\_LH, TH: temporal LH);

%***part1run\_safe***(temporalTH\_RH, TH: temporal RH);

%***part1run\_safe***(parietalTH\_LH, TH: parietal LH);

%***part1run\_safe***(parietalTH\_RH, TH: parietal RH);

%***part1run\_safe***(frontalTH\_LH, TH: frontal LH);

%***part1run\_safe***(frontalTH\_RH, TH: frontal RH);

\*Surface Area measures;

%***part1run\_safe***(MRI\_cort\_area\_ctx\_total, Area: Total);

%***part1run\_safe***(MRI\_cort\_area\_ctx\_lh\_total, Area: LH Total);

%***part1run\_safe***(MRI\_cort\_area\_ctx\_rh\_total, Area: RH Total);

%***part1run\_safe***(occipitalSA\_LH, SA: occipital LH);

%***part1run\_safe***(occipitalSA\_RH, SA: occipital RH);

%***part1run\_safe***(temporalSA\_LH, SA: temporal LH);

%***part1run\_safe***(temporalSA\_rH, SA: temporal RH);

%***part1run\_safe***(parietalSA\_LH, SA: parietal LH);

%***part1run\_safe***(parietalSA\_rH, SA: parietal RH);

%***part1run\_safe***(frontalSA\_LH, SA: frontal LH);

%***part1run\_safe***(frontalSA\_RH, SA: frontal RH);

\*Intracranial Volume;

%***part1run\_safe***(MRI\_subcort\_vol\_IntracranialVolu, Vol: intracranialVolume);

\*Macro graphs thickness and area weighted and unweighted models. Takes input of

unweighted (inp1) and weighted (inp2) best fitting models from macro above (linear, quadratic, cubic).

Genmod analyses output predicted values by age. Then unweighted and weighted curves are graphed by age.

Predicted values for unweighted models = opUW

Predicted values for weighted models = opW

;

**%macro** graphItSafe(inputb, side, title, inp1, inp2);

\*Unweighted model;

%if %upcase(&inp1)=LIN %then %do;

proc genmod data = wneuro\_safety ;

model &inputB = Age\_mean ;

output out = opUWdata (keep = wtfinal opUW rage Age\_mean subjID) PREDICTED = opUW;

title &inputB;

run;

%end;

%else %if %upcase(&inp1)=QUAD %then %do;

proc genmod data = wneuro\_safety ;

model &inputB = Age\_mean Age\_meansq ;

output out = opUWdata (keep = wtfinal opUW rage Age\_mean subjID) PREDICTED = opUW;

title &inputB;

run;

%end;

%else %if %upcase(&inp1)=CUB %then %do;

proc genmod data = wneuro\_safety ;

model &inputB = Age\_mean Age\_meansq Age\_meancu ;

output out = opUWdata (keep = wtfinal opUW rage Age\_mean subjID) PREDICTED = opUW;

title &inputB;

run;

%end;

\*Weighted model;

%if %upcase(&inp2)=LIN %then %do;

proc genmod data = wneuro\_safety ;

weight wtfinal;

model &inputB = Agew\_mean ;

output out = opWdata (keep = wtfinal opW rage Age\_mean subjID) PREDICTED = opW;

title &inputB;

run;

%end;

%else %if %upcase(&inp2)=QUAD %then %do;

proc genmod data = wneuro\_safety ;

weight wtfinal;

model &inputB = Agew\_mean Agew\_meansq ;

output out = opWdata (keep = wtfinal opW rage Agew\_mean subjID) PREDICTED = opW;

title &inputB;

run;

%end;

%else %if %upcase(&inp2)=CUB %then %do;

proc genmod data = wneuro\_safety ;

weight wtfinal;

model &inputB = Agew\_mean Agew\_meansq Agew\_meancu ;

output out = opWdata (keep = wtfinal opW rage Agew\_mean subjID) PREDICTED = opW;

title &inputB;

run;

%end;

proc sort data = opUWdata; by subjid;

proc sort data = opWdata; by subjid;

\*File of unweighted and weighted predicted values;

data ranTogether;

merge opUWdata opWdata; by subjid;

run;

\*Get mean value of predicted models by age (3-18);

proc sort data = ranTogether; by rage;

proc means data=ranTogether noprint;

by rage;

var opUW opW ;

output out=testMeanA mean=;

run;

%if %upcase(&inp1)=LIN %then %do;

symbol1 font=marker value=U

color=blue

interpol=join

height=**1.5** width =**2**;

%end;

%else %if %upcase(&inp1)=QUAD %then %do;

symbol1 font=marker value=C

color=blue

interpol=join

height=**1.5** width =**2**;

%end;

%else %if %upcase(&inp1)=CUB %then %do;

symbol1 font=marker value=P

color=blue

interpol=join

height=**1.5** width =**2**;

%end;

%if %upcase(&inp2)=LIN %then %do;

symbol2 font=marker value=U

color=red

interpol=join

height=**1.5** width =**2**;

%end;

%else %if %upcase(&inp2)=QUAD %then %do;

symbol2 font=marker value=C

color=red

line=**2**

interpol=join

height=**1.5** width =**2**;

%end;

%else %if %upcase(&inp2)=CUB %then %do;

symbol2 font=marker value=P

color=red

line=**2**

interpol=join

height=**1.5** width =**2**;

%end;

legend1 value=(

%if %upcase(&inp1)=LIN %then %do;

tick=**1** justify=c "Unweighted Linear"

%end;

%else %if %upcase(&inp1)=QUAD %then %do;

tick=**1** justify=c "Unweighted Quadratic"

%end;

%else %if %upcase(&inp1)=CUB %then %do;

tick=**1** justify=c "Unweighted Cubic"

%end;

%if %upcase(&inp2)=LIN %then %do;

tick=**2** justify=c "Weighted Linear"

%end;

%else %if %upcase(&inp2)=QUAD %then %do;

tick=**2** justify=c "Weighted Quadratic"

%end;

%else %if %upcase(&inp2)=CUB %then %do;

tick=**2** justify=c "Weighted Cubic"

%end;

)

shape=symbol(**8**,**1**)

position=(bottom center inside)

mode=share

label=none frame;

axis3 order=(**3** to **18** by **3**) minor=(n=**2**) label=( 'Age (year)' font='Helvetica/Bold' height=**1.75**);

axis2 major=(n=**5**) minor=(n=**3**) label=(a=**90** j=c "&side" font='Helvetica/Bold' height=**1.75**);

filename outfile "/Area &title..tiff";

goptions device=gif gsfname=outfile gsfmode=replace ftext='Helvetica/Bold' htext=**14**pt;

proc sort data = testMeanA; by rage; run;

proc gplot data=testMeanA;

plot opuW\*rage opW\*rage/overlay legend = legend1 haxis=axis3 vaxis = axis2;

title "&title";

run;

quit;

**%mend**;

%***graphItSafe***(MRI\_cort\_thick\_ctx\_mean, Thickness (mm), Total Mean Thickness, cub, cub);

%***graphItSafe***(MRI\_cort\_thick\_ctx\_lh\_mean, Thickness (mm), LH Mean Thickness, cub, lin);

%***graphItSafe***(MRI\_cort\_thick\_ctx\_rh\_mean, Thickness (mm), RH Mean Thickness, cub, cub);

%***graphItSafe***(frontalTH\_LH, Thickness (mm), LH Frontal Region,cub, cub);

%***graphItSafe***(frontalTH\_RH, Thickness (mm), RH Frontal Region,quad, quad);

%***graphItSafe***(occipitalTH\_LH,Thickness (mm), LH Occipital Region,cub, quad);

%***graphItSafe***(occipitalTH\_RH,Thickness (mm), RH Occipital Region,cub, quad);

%***graphItSafe***(temporalTH\_LH,Thickness (mm), LH Temporal Region,lin,lin);

%***graphItSafe***(temporalTH\_rH,Thickness (mm), RH Temporal Region,lin, lin);

%***graphItSafe***(parietalTH\_LH,Thickness (mm), LH Parietal Region,lin, lin);

%***graphItSafe***(parietalTH\_RH,Thickness (mm), RH Parietal Region,quad, cub);

ods html close; ods html;

%***graphItSafe***(MRI\_cort\_area\_ctx\_total, Surface Area (mm2), Total Cortical Surface Area, quad, cub);

%***graphItSafe***(MRI\_cort\_area\_ctx\_lh\_total, Surface Area (mm2), LH Cortical Surface Area, quad, cub);

%***graphItSafe***(MRI\_cort\_area\_ctx\_rh\_total, Surface Area (mm2), RH Cortical Surface Area, quad, cub);

%***graphItSafe***(frontalSA\_LH, Surface Area (mm2), LH Frontal Region,quad, cub);

%***graphItSafe***(frontalSA\_RH, Surface Area (mm2), RH Frontal Region,quad, cub);

%***graphItSafe***(occipitalSA\_LH,Surface Area (mm2), LH Occipital Region,quad, cub);

%***graphItSafe***(occipitalSA\_RH,Surface Area (mm2), RH Occipital Region,quad, cub);

%***graphItSafe***(temporalSA\_LH,Surface Area (mm2), LH Temporal Region,quad, cub);

%***graphItSafe***(temporalSA\_rH,Surface Area (mm2), RH Temporal Region,quad, cub);

%***graphItSafe***(parietalSA\_LH,Surface Area (mm2), LH Parietal Region,cub, cub);

%***graphItSafe***(parietalSA\_RH,Surface Area (mm2), RH Parietal Region, cub,cub);

%***graphItSafe***(MRI\_subcort\_vol\_IntracranialVolu, Volume (mm3), Subcortical Intracranial Volume, quad, quad);

\*This macro outputs the linear, quadratic, and cubic unweighted and weighted models

for volume brain regions. Cotrolling for intracranial volume;

**%macro** part1runv(inputb, title);

\*Unweighted linear;

proc genmod data = wneuro\_safety;

model &inputB = Age\_mean MRI\_subcort\_vol\_IntracranialVolu;

title &inputB;

run;

\*Weighted linear;

proc genmod data = wneuro\_safety;

weight wtfinal;

model &inputB = Agew\_mean MRI\_subcort\_vol\_IntracranialVolu;

title &inputB;

run;

\*Unweighted quadratic;

proc genmod data = wneuro\_safety;

model &inputB = Age\_mean Age\_meansq MRI\_subcort\_vol\_IntracranialVolu;

title &inputB;

run;

\*Weighted quadratic;

proc genmod data = wneuro\_safety;

weight wtfinal;

model &inputB = Agew\_mean Agew\_meansq MRI\_subcort\_vol\_IntracranialVolu;

title &inputB;

run;

\*Unweighted cubic;

proc genmod data = wneuro\_safety;

model &inputB = Age\_mean Age\_meansq Age\_meancu MRI\_subcort\_vol\_IntracranialVolu;

title &inputB;

run;

\*Weighted cubic;

proc genmod data = wneuro\_safety;

weight wtfinal;

model &inputB = Agew\_mean Agew\_meansq Agew\_meancu MRI\_subcort\_vol\_IntracranialVolu;

title &inputB;

run;

**%mend**;

%***part1runV***(MRI\_cort\_vol\_ctx\_lh\_total, VOL: total LH);

%***part1runV***(MRI\_cort\_vol\_ctx\_rh\_total, VOL: total RH);

%***part1runV***(MRI\_cort\_vol\_ctx\_total, Vol: TOTAL);

%***part1runV***(totalHippocam\_vol, VOL: total Hippocampus);

%***part1runV***(totalAmygdala\_vol, VOL: total Amygdala);

%***part1runV***(totalBasalGang\_vol, VOL: total Basal Ganglia);

%***part1runV***(totalSubCort\_vol, VOL: subcortvol);

\*\*\*\*\*\*\*\*\*\* for graphing Volume measures \*\*\*\*\*\*\*\*\*\*;

\* Create mean of intracranial volume by two-year age groups to graph the volume

controlling for intracranial volume. Create mean measure for unweighted and weighted models;

**proc** **sort** data = wneuro\_safety; by ragecat; **run**;

**proc** **means** data = wneuro\_safety noprint;

by ragecat;

var MRI\_subcort\_vol\_IntracranialVolu;

output out=mIntraVol\_1162 mean=mIntraVol\_1162;

**run**;

**proc** **sort** data = wneuro\_safety; by ragecat; **run**;

**proc** **means** data = wneuro\_safety noprint;

weight wtfinal;

by ragecat;

var MRI\_subcort\_vol\_IntracranialVolu;

output out=mIntraVol\_1162w mean=mIntraVol\_1162w;

**run**;

**proc** **sort** data = wneuro\_safety; by ragecat; **run**;

**proc** **sort** data = mIntraVol\_1162; by ragecat; **run**;

**proc** **sort** data = mIntraVol\_1162w; by ragecat; **run**;

\* combined data with categorical intracranial volume;

**data** wneuro\_safety2 (drop =mIntraVol) ;

merge wneuro\_safety mIntraVol\_1162 mIntraVol\_1162w; by ragecat;

**run**;

**proc** **freq** data = wneuro\_safety2;

tables ragecat\*mIntraVol\_1162\*mIntraVol\_1162w/list;

ods output list=freq\_categ;

**run**;

\*calculate the percentage of age categorical var;

**proc** **sql**;

create table percent\_categ as

select \*, frequency/sum(frequency) as percentNew

from freq\_categ

;

**quit**;

\* Merge with frequency table to get percent of each age category;

**proc** **sort** data = freq\_categ; by ragecat;

**proc** **sort** data = wneuro\_safety2; by ragecat;

**data** wneuro\_safety3;

merge freq\_categ wneuro\_safety2; by ragecat;

\*rename variables;

percentUse = percent;

mIntraVol\_fix = mIntraVol\_1162;

mIntraVolW\_fix = mIntraVol\_1162w;

**run**;

\*rename so can merge into output from genmod equation in graphing macro;

**data** percent\_categ2 (keep = level1 percentuse);

set percent\_categ;

level1 = F\_mIntraVol\_1162 \***1**;

rename percentNew = percentuse;

**run**;

\*rename so can merge with predicted values from genmod equation in graphing macro;

**data** percent\_categ2w (keep = level1 percentuse);

set percent\_categ;

level1 = F\_mIntraVol\_1162w \***1**;

rename percentNew = percentuse;

**run**;

\*create data 'ageuse' with age and categorical intracranial volume measure to later merge into predicted values from genmod equation in graphing macro (Unweighted);

**proc** **freq** data =wneuro\_safety3;

tables mIntraVol\_fix\*mIntraVolW\_fix\*rage\*age\_mean/list;

ods output list=ageuse (keep = rage Age\_mean mIntraVol\_fix);

**run**;

\*create data 'ageuseW' with age and categorical intracranial volume measure to later merge into predicted values from genmod equation in graphing macro (weighted);

**proc** **freq** data =wneuro\_safety3;

tables mIntraVol\_fix\*mIntraVolW\_fix\*rage\*agew\_mean/list;

ods output list=ageuseW (keep = rage Agew\_mean mIntraVolw\_fix);

**run**;

\*Macro to graph volume regions. macro inputs 'inp1' and 'inp2' determine if the

equation for unweighted and weighted respectively should be linear, quadratic, or cubic;

**%macro** graphItvSafe(inputb, side, title, inp1, inp2);

%if %upcase(&inp1)=LIN %then %do;

\*UWests saves the output estimates to graph volume of brain regions controlling for categorical intracranial volume (unweighted);

ods output ParameterEstimates =UWests (keep = parameter estimate level1);

proc genmod data = wneuro\_safety3 ;

class mIntraVol\_fix;

model &inputB = Age\_mean mIntraVol\_fix;\* controling for categorical intracranial volume;

output out = opUWdata (keep = wtfinal opUW rage Age\_mean subjID mIntraVol\_fix percentUse) PREDICTED = opUW;

title &inputB;

run;

\*combine age with parameter estimate output;

proc sql;

create table testx as

select \*

from ageuse, uwests

;

quit;

\*change level1 to a numerical value;

data testx;

set testx;

level1\_2 = level1\***1**;

drop level1;

rename level1\_2 = level1;

run;

proc sort data = testx; by level1;

proc sort data = percent\_categ2; by level1;

\*This dataset sets up the values of the brain region as a function of mean centered age;

data uwests2;

merge testx percent\_categ2; by level1;

\*The variable 'percentUse' is a multiplier for each beta estimate value and

holds the values of the independent variable (mean centered age) to multiply each parameter estimate;

if parameter = "Scale" then delete;

if parameter = "Intercept" then percentuse = **1**;

if parameter = "Age\_mean" then percentuse = age\_mean;

else if parameter = "mIntraVol\_fix" then percentuse =percentuse;

else if percentuse = **.** then percentuse = **1**;

prodCol = percentuse\*estimate; \*value for each piece of the genmod equation;

run;

\*uwestsP creates predV. PredV = total predicted value of volume by each age;

proc sql;

create table uwestsP as

select distinct sum(prodcol) as predV, rage

from uwests2

group by rage

;

quit;

proc sort data = uwests2; by rage; run;

%end;

%else %if %upcase(&inp1)=QUAD %then %do;

\*UWests saves the output estimates to graph volume of brain regions controlling for categorical intracranial volume (unweighted);

ods output ParameterEstimates =UWests (keep = parameter estimate level1 );

proc genmod data = wneuro\_safety3 ;

class mIntraVol\_fix ;

model &inputB = Age\_mean Age\_meansq mIntraVol\_fix;\*cenrace gender scanner5c;

output out = opUWdata (keep = wtfinal opUW rage Age\_mean subjID mIntraVol\_fix percentUse) PREDICTED = opUW;

title &inputB;

\*combine age with parameter estimate output;

proc sql;

create table testx as

select \*

from ageuse, uwests

;

quit;

\*change 'level1' to a numerical variable;

data testx;

set testx;

level1\_2 = level1\***1**;

drop level1;

rename level1\_2 = level1;

run;

proc sort data = testx; by level1;

proc sort data = percent\_categ2; by level1;

\*This dataset sets up the values of the brain region as a function of mean centered age;

data uwests2;

merge testx percent\_categ2; by level1;

\*The variable 'percentUse' is a multiplier for each beta estimate value and

holds the values of the independent variable (mean centered age) to multiply each paramter estimate;

if parameter = "Scale" then delete;

if parameter = "Intercept" then percentuse = **1**;

if parameter = "Age\_mean" then percentuse = age\_mean;

else if parameter = "Age\_meansq" then percentuse = age\_mean\*age\_mean;

else if parameter = "mIntraVol\_fix" then percentuse =percentuse;

else if percentuse = **.** then percentuse = **1**;

prodCol = percentuse\*estimate;

run;

\*uwestsP creates predV. PredV = total predicted value of volume by each age;

proc sql;

create table uwestsP as

select distinct sum(prodcol) as predV, rage, age\_mean

from uwests2

group by rage

;

quit;

proc sort data = uwests2; by rage; run;

%end;

%else %if %upcase(&inp1)=CUB %then %do;

\*UWests saves the output estimates to graph volume of brain regions controlling for categorical intracranial volume (unweighted);

ods output ParameterEstimates =UWests (keep = parameter estimate level1 );

proc genmod data = wneuro\_safety3 ;

class mIntraVol\_fix ;

model &inputB = Age\_mean Age\_meansq Age\_meancu mIntraVol\_fix;\*cenrace gender scanner5c;

output out = opUWdata (keep = wtfinal opUW rage Age\_mean subjID mIntraVol\_fix percentUse) PREDICTED = opUW;

title &inputB;run;

\*combine age with parameter estimate output;

proc sql;

create table testx as

select \*

from ageuse, uwests;

quit;

\*change 'level1' to a numerical variable;

data testx;

set testx;

level1\_2 = level1\***1**;

drop level1;

rename level1\_2 = level1;

run;

proc sort data = testx; by level1;

proc sort data = percent\_categ2; by level1;

\*This dataset sets up the values of the brain region as a function of mean centered age;

data uwests2;

merge testx percent\_categ2; by level1;

\*The variable 'percentUse' is a multiplier for each beta estimate value and

holds the values of the independent variable (mean centered age) to multiply each parameter estimate;

if parameter = "Scale" then delete;

if parameter = "Intercept" then percentuse = **1**;

if parameter = "Age\_mean" then percentuse = age\_mean;

else if parameter = "Age\_meansq" then percentuse = age\_mean\*age\_mean;

else if parameter = "Age\_meancu" then percentuse = age\_mean\*age\_mean\*age\_mean;

else if parameter = "mIntraVol\_fix" then percentuse =percentuse;

else if percentuse = **.** then percentuse = **1**;

prodCol = percentuse\*estimate;

run;

\*uwestsP creates predV. PredV = total predicted value of volume by each age;

proc sql;

create table uwestsP as

select distinct sum(prodcol) as predV, rage

from uwests2

group by rage

;

quit;

proc sort data = uwests2; by rage; run;

%end;

%if %upcase(&inp2)=LIN %then %do;

\*Wests saves the output estimates to graph volume of brain regions controlling for categorical intracranial volume (weighted);

ods output ParameterEstimates =Wests (keep = parameter estimate level1);

proc genmod data = wneuro\_safety3 ;

weight wtfinal;

class mIntraVolw\_fix;

model &inputB = Agew\_mean mIntraVolw\_fix;\*cenrace gender scanner5c;

output out = opWdata (keep = wtfinal opW rage Agew\_mean subjID mIntraVolw\_fix percentUse) PREDICTED = opW;

title &inputB;

run;

\*combine age with parameter estimate output;

proc sql;

create table testx2 as

select \*

from ageuseW, Wests;

quit;

\*change 'level1' to a numerical variable;

data testx2;

set testx2;

level1\_2 = level1\***1**;

drop level1;

rename level1\_2 = level1;

run;

proc sort data = testx2; by level1;

proc sort data = percent\_categ2w; by level1;

\*This dataset sets up the values of the brain region as a function of mean centered age;

data Wests2;

merge testx2 percent\_categ2w; by level1;

\*The variable 'percentUse' is a multiplier for each beta estimate value and

holds the values of the independent variable (mean centered age) to multiply each paramter estimate;

if parameter = "Scale" then delete;

if parameter = "Intercept" then percentuse = **1**;

if parameter = "AgeW\_mean" then percentuse = Agew\_mean;

else if parameter = "mIntraVolW\_fix" then percentuse =percentuse;

else if percentuse = **.** then percentuse = **1**;

prodCol = percentuse\*estimate;

run;

\*westsP creates predV. PredV = total predicted value of volume by each age;

proc sql;

create table westsP as

select distinct sum(prodcol) as predV2, rage

from wests2

group by rage;

quit;

%end;

%else %if %upcase(&inp2)=QUAD %then %do;

\*Wests saves the output estimates to graph volume of brain regions controlling for categorical intracranial volume (weighted);

ods output ParameterEstimates =Wests (keep = parameter estimate level1);

proc genmod data = wneuro\_safety3 ;

weight wtfinal;

class mIntraVolw\_fix;

model &inputB = Agew\_mean Agew\_meansq mIntraVolw\_fix;\*cenrace gender scanner5c;

output out = opWdata (keep = wtfinal opW rage Agew\_mean subjID mIntraVolw\_fix percentUse) PREDICTED = opW;

title &inputB;

run;

\*combine age with parameter estimate output;

proc sql;

create table testx2 as

select \*

from ageuseW, Wests;

quit;

\*change 'level1' to a numerical variable;

data testx2;

set testx2;

level1\_2 = level1\***1**;

drop level1;

rename level1\_2 = level1;

run;

proc sort data = testx2; by level1;

proc sort data = percent\_categ2w; by level1;

\*This dataset sets up the values of the brain region as a function of mean centered age;

data Wests2;

merge testx2 percent\_categ2w; by level1;

\*The variable 'percentUse' is a multiplier for each beta estimate value and

holds the values of the independent variable (mean centered age) to multiply each paramter estimate;

if parameter = "Scale" then delete;

if parameter = "Intercept" then percentuse = **1**;

if parameter = "AgeW\_mean" then percentuse = Agew\_mean;

else if parameter = "AgeW\_meansq" then percentuse = Agew\_mean\*Agew\_mean;

else if parameter = "mIntraVolW\_fix" then percentuse =percentuse;

else if percentuse = **.** then percentuse = **1**;

prodCol = percentuse\*estimate;

run;

\*westsP creates predV. PredV = total predicted value of volume by each age;

proc sql;

create table westsP as

select distinct sum(prodcol) as predV2, rage

from wests2

group by rage;

quit;

%end;

%else %if %upcase(&inp2)=CUB %then %do;

\*Wests saves the output estimates to graph volume of brain regions controlling for categorical intracranial volume (weighted);

ods output ParameterEstimates =Wests (keep = parameter estimate level1);

proc genmod data = wneuro\_safety3 ;

weight wtfinal;

class mIntraVolw\_fix;

model &inputB = Agew\_mean Agew\_meansq Agew\_meancu mIntraVolw\_fix;\* cenrace gender scanner5c;

output out = opWdata (keep = wtfinal opW rage Agew\_mean subjID mIntraVolw\_fix percentUse) PREDICTED = opW;

title &inputB;

run;

\*combine age with parameter estimate output;

proc sql;

create table testx2 as

select \*

from ageuseW, Wests;

quit;

\*change 'level1' to a numerical variable;

data testx2;

set testx2;

level1\_2 = level1\***1**;

drop level1;

rename level1\_2 = level1;

run;

proc sort data = testx2; by level1;

proc sort data = percent\_categ2w; by level1;

\*This dataset sets up the values of the brain region as a function of mean centered age;

data Wests2;

merge testx2 percent\_categ2w; by level1;

\*The variable 'percentUse' is a multiplier for each beta estimate value and

holds the values of the independent variable (mean centered age) to multiply each paramter estimate;

if parameter = "Scale" then delete;

if parameter = "Intercept" then percentuse = **1**;

if parameter = "AgeW\_mean" then percentuse = Agew\_mean;

else if parameter = "AgeW\_meansq" then percentuse = Agew\_mean\*Agew\_mean;

else if parameter = "AgeW\_meancu" then percentuse = Agew\_mean\*Agew\_mean\*Agew\_mean;

else if parameter = "mIntraVolW\_fix" then percentuse =percentuse;

else if percentuse = **.** then percentuse = **1**;

prodCol = percentuse\*estimate;

run;

\*westsP creates predV. PredV = total predicted value of volume by each age;

proc sql;

create table westsP as

select distinct sum(prodcol) as predV2, rage, Agew\_mean

from wests2

group by rage

;

quit;

%end;

\*\*\*\*\*\*\*\* Combine unweighted and weighted curves \*\*\*\*\*\*\*\*\*\*\*\*;

proc sort data = westsp; by rage;

proc sort data = Uwestsp; by rage;

data ranTogether;

merge westsp Uwestsp; by rage;

run;

%if %upcase(&inp1)=LIN %then %do;

symbol1 font=marker value=U

color=blue

interpol=join

height=**1.5** width =**2**;

%end;

%else %if %upcase(&inp1)=QUAD %then %do;

symbol1 font=marker value=C

color=blue

interpol=join

height=**1.5** width =**2**;

%end;

%else %if %upcase(&inp1)=CUB %then %do;

symbol1 font=marker value=P

color=blue

interpol=join

height=**1.5** width =**2**;

%end;

%if %upcase(&inp2)=LIN %then %do;

symbol2 font=marker value=U

color=red

interpol=join

height=**1.5** width =**2**;

%end;

%else %if %upcase(&inp2)=QUAD %then %do;

symbol2 font=marker value=C

color=red

line=**2**

interpol=join

height=**1.5** width =**2**;

%end;

%else %if %upcase(&inp2)=CUB %then %do;

symbol2 font=marker value=P

color=red

line=**2**

interpol=join

height=**1.5** width =**2**;

%end;

legend1 value=(

%if %upcase(&inp1)=LIN %then %do;

tick=**1** justify=c "Unweighted Linear"

%end;

%else %if %upcase(&inp1)=QUAD %then %do;

tick=**1** justify=c "Unweighted Quadratic"

%end;

%else %if %upcase(&inp1)=CUB %then %do;

tick=**1** justify=c "Unweighted Cubic"

%end;

%if %upcase(&inp2)=LIN %then %do;

tick=**2** justify=c "Weighted Linear"

%end;

%else %if %upcase(&inp2)=QUAD %then %do;

tick=**2** justify=c "Weighted Quadratic"

%end;

%else %if %upcase(&inp2)=CUB %then %do;

tick=**2** justify=c "Weighted Cubic"

%end;

)

shape=symbol(**8**,**1**)

position=(bottom center inside)

mode=share

label=none frame;

axis3 order=(**3** to **18** by **3**) minor=(n=**2**) label=( 'Age (year)' font='Helvetica/Bold' height=**1.75**);

axis2 major=(n=**5**) minor=(n=**3**) label=(a=**90** j=c "&side" font='Helvetica/Bold' height=**1.75**);

filename outfile "/Vol &title..tiff";

goptions device=gif gsfname=outfile gsfmode=replace;

proc sort data = rantogether; by rage; run;

proc gplot data=rantogether;

plot predv\*rage predv2\*rage/overlay legend = legend1 haxis=axis3 vaxis = axis2;

title "&title";

run;

quit;

**%mend**;

%***graphItvSafe***(totalHippocam\_vol, Volume (mm3), Total Hippocampus, quad, quad);

%***graphItvSafe***(totalAmygdala\_vol, Volume (mm3), Total Amygdala, quad, cub);

%***graphItvSafe***(totalBasalGang\_vol, Volume (mm3), Total Basal Ganglia, quad, cub);

%***graphItvSafe***(totalSubCort\_vol, Volume (mm3), Total Subcortical Volume, quad, cub);

%***graphItvSafe***(MRI\_cort\_vol\_ctx\_lh\_total, Volume (mm3), LH Cortical Volume, quad, cub);

%***graphItvSafe***(MRI\_cort\_vol\_ctx\_rh\_total, Volume (mm3), RH Cortical Volume, quad, cub);

%***graphItvSafe***(MRI\_cort\_vol\_ctx\_total, Volume (mm3), Total Cortical Volume, quad, cub);