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';

\*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**;

\*SETTING ALL HOUSEHOLD DATA TOGETHER IN TEMP FILE;

**data** pumsHouse;

set pumsHa pumsHb pumsHc pumsHd;

inpumsHouse =**1**;

**run**;

\*individual level data;

**data** pumsa;

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

**run**;

**data** pumsb;

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

**run**;

**data** pumsc;

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

**run**;

**data** pumsd;

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

**run**;

\*SETTING ALL INDIVIDUAL DATA TOGETHER IN TEMP FILE, KEEPING 15 OF THE VARS;

\*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**; \*hispeth = 2 - hispanic;

\*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**;

\*if agecat = missing then not in the age population we are looking at;

inpums = **1**;

**run**;

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

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

\*combining household and individual level data. Serialno is the serial number of the household, and each individual in the household has the same serialno;

**data** combAllPUMS1 (drop = ADJHSG ADJINC AGS BATH BDSP BLD BUS CONP DIVISION ELEP FACRP FAGSP FBATHP FBDSP FBLDP

FBUSP FCONP FELEP FES FFSP FFULP FGASP FHFLP relp FINSP FKITP FMHP FMRGIP FMRGP FMRGTP FMRGXP FMVP FPLMP FREFRP FRMSP

FRNTMP FRNTP FRWATP FS FSINKP FSMP FSMXHP FSMXSP FSTOVP FTAXP FTELP FTENP FTOILP FULP FVACSP FVALP FVEHP FWATP FYBLP

GASP GRNTP GRPIP HFL HHL HHT HUGCL HUPAC HUPAOC HUPARC INSP KIT LNGI MHP MRGI MRGP MRGT MRGX MULTG MV

NOC NP NPF NPP NR NRC OCPIP PARTNER PLM PSF PUMA PWGTP1 R18 R60 R65 RC REFR REGION RESMODE RMSP RNTM RNTP RT SINK

SMOCP SMP SMX SRNT ST STOV SVAL TAXP TEL TEN TOIL TYPE VACS VALP VEH WAGP WATP RWAT WGTP2 WGTP3 WGTP4 WGTP5 WGTP6 WGTP7

WGTP8 WGTP9 WGTP10 WGTP11 WGTP12 WGTP13 WGTP14 WGTP15 WGTP16 WGTP17 WGTP18 WGTP19 WGTP20 WGTP21 WGTP22 WGTP23 WGTP24

WGTP25 WGTP26 WGTP27 WGTP28 WGTP29 WGTP30 WGTP31 WGTP32 WGTP33 WGTP34 WGTP35 WGTP36 WGTP37 WGTP38 WGTP39 WGTP40 WGTP41

WGTP42 WGTP43 WGTP44 WGTP45 WGTP46 WGTP47 WGTP48 WGTP49 WGTP50 WGTP51 WGTP52 WGTP53 WGTP54 WGTP55 WGTP56 WGTP57 WGTP58

WGTP59 WGTP60 WGTP61 WGTP62 WGTP63 WGTP64 WGTP65 WGTP66 WGTP67 WGTP68 WGTP69 WGTP70 WGTP71 WGTP72 WGTP73 WGTP74 WGTP75

WGTP76 WGTP77 WGTP78 WGTP79 WGTP80 WIF WKEXREL WORKSTAT YBL);

merge PUMS pumshouse; by serialno;

relpnum= relp\***1**; \*relpnum = relationship status to reference person; \* 0- reference person, 1-husband wife, 2-4 child, 5- sibling, 6- parent;

if relpnum = **.** then delete; \*deleting any observations that are housing units that do not include individuals;

if **6**<relpnum<=**17** then delete; \*delete anyone who is not a child, sibling, parent, or reference person... aka grandparents, cousins, non family members;

if relpnum = **1** then delete; \*deleting husband/wives from data;

if age<**3** then delete; \* deleting any children younger than the age we are looking at;

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

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

\*survey weights;

pWeight = PWGTP\***1**;

houseweight = WGTP\***1**;

**run**;

\* get frequency of MAX number of children possible. Before linking them to their parents

\* we will only look at children who are connected to parents as we are weighting based on parental education

\* MAX weighted N = 59,173,538;

**proc** **freq** data = combAllPUMS1;

weight pWeight;

table agecat; **run**;

**data** pums.combAllPUMS;

set combAllPUMS1;

IDNewCh=\_n\_; \*creating new id variable so everyone has a unique ID;

IDnew = IDNewCh\***1**; \*making IDnew a numeric variable;

**run**;

ods html close; ods html;

\*here checking distribution of id variable;

**proc** **means** data =pums.combAllPUMS n mean p25 p50 p75 min max;

var IDnew;

**run**;

\*This SQL statement is intended to map a variable onto the households who have a kid reference (someone under 18)

onto everyone in that household, so if you look at anyone in that household where kidref is NOT '.' then that household has a kid as the reference person;

**proc** **sql**;

create table kidrefdata AS /\* creates new data set kidrefdata\*/

select \* from pums.combAllPUMS /\* selecting ALL VARIABLES AS THEY ARE FROM pums.combAllPUMS \*/

left join ( /\*LEFT JOIN returns all records from the left table (entire line above), and the matched records from the right table (the smaller dataset below)\*/

select distinct serialno as kidrefserial, refPerson as kidRef /\* selecting the serialno (household identifier), RENAMED kidserial as only want to look at reference kids in this, and the binary variable refPerson for the household \*/

from pums.combAllPUMS /\* variables above taken from the same dataset \*/

where relpnum =**0** and agecat in (**1**,**2**,**3**,**4**)) /\* Where statement selects relpnum = 0 (reference person) between 3-18 years old from second (right) dataset \*/

on (serialno = kidrefserial); /\* this statement merges the two datasets on the serialnumbers in the smaller (right table) dataset \*/

**quit**;

\*checking new variable created, kidref, against variables in the original data;

**proc** **freq** data = pums.combAllPUMS;

table refperson\*relpnum\*age/missing list;

**run**;

**proc** **freq** data = kidrefdata;

table kidref\*refperson\*relpnum\*age/missing list;

**run**;

\*checking observations to make sure;

ods html close; ods html;

options obs=**100**;

**proc** **print** data = kidrefdata;

var idnew serialno kidrefserial kidref refperson relpnum age;

where kidref =**1** and idnew>**1000000**;

**run**;

options obs = max;

\* continue editing data;

**data** combPUM\_0216(keep = AGEP HINCP HISP

RAC1P hispeth SCH SCHG SCHL SEMP SEX kidRef

SFR age agecat houseInc serialno IDnew kidof refperson kidSerial

refserial kidrefper PotentialParSer PotentialPar pweight houseweight

income3cat education2 inpums inpumshouse relpnum haskidserial kidrefserial);

set kidrefdata;

\*\*\*\* variable "kidref" now means that that family's reference is someone between the ages of 15-18;

if kidof = **1** then kidSerial = serialno; \*if someone is a kid of the referenceperson, then kidserial variable is set to the household serial number;

if refperson = **1** then refSerial = serialno;\*if someone is the referenceperson, then refSerial variable is set to the household serial number;

if relpnum = **5** and kidref = **1** and agecat in (**1**,**2**,**3**,**4**) then kidSerial = serialno; \*if someone is a sibling of the kid referenceperson (variable

created in the SQL statement above), then kidserial variable is set to the household serial number;

\*if reference person is between 3-18: ;

\*if agecat is not missing age must be between 3 and 18 so they must be a kid reference person;

if agecat in (**1**,**2**,**3**,**4**) and refperson = **1** then kidRefPer = **1**; else kidrefper = **0**; \*different than kid ref, this only applies to the ACTUAL REFERENCE KID;

if kidrefper = **1** then kidSerial = serialno; \*giving reference kids the variable kidserial;

if kidserial ~=**.** then haskidserial = **1**;

if (relpnum = **6** and kidref = **1**) or (kidrefper = **0** and refperson =**1**) then PotentialPar = **1**; else potentialpar = **0**; \*creating variable for a potential parent;

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

\*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**;

if kidref = **1** and kidrefper = **0** and relpnum in (**2**,**3**,**4**) then delete; \* deleting children of 15-18 year old reference kids;

if kidof = **1** and agecat =**.** then delete; \*deleting children of reference person who are not 3-18;

**run**;

\*testing newly created variables;

**proc** **freq** data = combpum\_0216;

table kidof\*haskidserial\*agecat\*kidref\*kidrefper\*relpnum/missing list;

**run**;

**proc** **freq** data = combpum\_0216;

table kidof\*PotentialPar\*agecat\*kidref\*kidrefper\*relpnum/missing list;

**run**;

ods html close; ods html;

options obs=**100**;

**proc** **print** data = combpum\_0216;

var idnew serialno kidserial PotentialParSer kidref refperson relpnum;

where kidref =**1** and idnew>**1000000**;

**run**;

**proc** **print** data = combpum\_0216;

var idnew serialno kidserial PotentialParSer kidref refperson relpnum;

where idnew>**1036160**;

**run**;

\*This SQL statement is intended to map the parent variables onto the kids they may have

so after this, any CHILD in a household with a parent-child relationship, should now have their parents education attached to them;

options obs=max;

**proc** **sql**;

create table PumsWithEd\_0216 AS

select distinct \* from combPUM\_0216 /\* selecting all variables in the dataset created above \*/

left join ( /\*LEFT JOIN returns all records from the left table (entire line above), and the matched records from the right table (the smaller dataset below)\*/

select distinct PotentialParSer as ParSerialno, education2 AS parentEd, relpnum as IsParent, idnew AS parIDNEW/\* selecting the PotentialparSer (potential parent serial number) RENAMED as parserialno because after this merge these will only be actual parents, and their education and relationship variable \*/

from combPUM\_0216

where PotentialPar =**1**) /\* Onlu look at the people who could be potential parents (coded above)\*/

on (kidSerial = parserialno); /\* this statement merges the two datasets on the serialnumbers in the smaller (right table) dataset \*/

\*adding a new variable to the created dataset, counting the number of parents in the data a child has.

This only applies to reference kids and thier siblings;

create table PumsWithEd\_0216\_2 AS

select \*, count(distinct paridnew) as numparents

from PumsWithEd\_0216

group by kidrefserial;

**quit**;

\*Checking output;

ods html close; ods html;

options obs=**100**;

**proc** **print** data = PumsWithEd\_0216\_2;

var idnew serialno kidserial kidrefserial parserialno potentialparser kidref refperson relpnum parented education2 age parIDNEW numparents;

where numparents = **2**;

**run**;

\*The below SQL statement deletes 1 of the two child observations created if there is a kid reference and both parents are in the data. back to original N;

options obs=max;

**proc** **SQL**;

delete from PumsWithEd\_0216\_2

where numparents = **2** and mod(paridnew,**2**) = **0**; \*if a child has two parents, deletes 1 regardless of education status, deletes one that has an even ID. I did it this way as for kids whos parents are the reference person, we do not clarify if this is the parent with higher/lower education;

**quit**;

\*checks that correct parent was deleted and other information stays the same.;

options obs=**100**;

**proc** **print** data = PumsWithEd\_0216\_2;

var idnew serialno kidserial kidrefserial parserialno potentialparser kidref refperson relpnum parented education2 age parIDNEW numparents;

where numparents = **2**;

**run**;

\*varialbe isParent can be 0 or 6. if IsParent is NOT missing, then they are a parent;

\* creating this dataset to specify which 3-18 year olds have a parent in the data. ;

option obs = max;

**data** pums.PumsWithEd\_0216\_2;

set PumsWithEd\_0216\_2;

if kidref = **1** and isparent = **6** and relpnum in (**0**,**5**) then kidParent = **1**; \*the household has a kid reference and the parent of that household is in the data, and they are either THE reference kid or a sibling of the reference kid;

else if kidref = **.** and isparent = **0** and refperson = **0** then kidparent = **1**; \*if kidref = . then the reference person is not a child, isparent =0 means the parent exists and they are the reference person, and refperson =0 means not the reference person;

**run**;

options obs=**100**;

**proc** **print** data = pums.PumsWithEd\_0216\_2;

var idnew serialno kidserial kidrefserial parserialno potentialparser kidref refperson relpnum parented education2 age parIDNEW numparents kidparent;

**run**;

options obs=max;

\*this should be the total N of the data we need for the weight;

**proc** **freq** data = pums.PumsWithEd\_0216\_2;

table kidParent\*refperson\*IsParent\*kidref\*relpnum/list missing;

where kidParent = **1** and agecat in (**1**,**2**,**3**,**4**);

**run**;

\*total weighted N;

**proc** **freq** data = pums.PumsWithEd\_0216\_2;

weight pweight;

table kidParent\*refperson\*IsParent\*kidref\*relpnum/list missing;

where kidParent = **1** and agecat in (**1**,**2**,**3**,**4**);

**run**;

**data** PumsWithEd\_0216\_3;

set pums.PumsWithEd\_0216\_2;

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

if agecat = **.** then delete; \*delete anyone who is not ages 3-18, including parents;

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

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

else if rac1p = **1** then cenRace = **1**; \*non hisp white;

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

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

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

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

incomemiss= income3cat;

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

**run**;

\*checking that everyone who is hispanic is labeled as hispanic, and no other races include hispanic individuals;

**proc** **freq** data = PumsWithEd\_0216\_3;

table cenrace\*hispeth rac1p\*hispeth cenrace\*rac1p/missing;

**run**;

\*values for creating a census weight;

**proc** **freq** data = PumsWithEd\_0216\_3;

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;

\*2/21/18 corrections;

**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 \*/ **58687792**

/\* race \*/ **32765377** **7680805** **12937798** **3220387** **2083425**

/\* gender \*/**30091602** **28596190**

/\* parented\*/ **21800954** **19142190** **11101112** **6643536**

/\* income\*/ **19585978** **23820095** **15259533** **22186**

;

IDVAR subjid ragecat;

OUTPUT IDVAR wtfinal/filename=mod\_weights5\_c

predicted=all replace;

**run**;

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

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

**data** weightedNeuro;

merge mod\_weights5\_c (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;

\*2/21/18 adjusted weighted mean age for new weight;

meanAgeW = **10.6991121**;

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**;

/\*

data pingSensitivity;

set wneuro\_safety;

if wtfinal >399486.5486 then delete;

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**;

ods html close; ods Html;

\*Thickness measures;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

\*Surface Area measures;

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods html;

%***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 fittign 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;

goptions hsize=**6** vsize=**4.5**;

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

axis2 order=(**9700** to **11600** by **475**) minor=(n=**1**) label=(a=**90** j=c h=**1.75** f=simplex "&side" );

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

filename outfile "/ SA0222 &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***(occipitalSA\_LH,Surface Area (mm2), LH Occipital Region,quad, cub);

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

\*2/22 changes based on updated results;

%***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);

\*\* this one below is different. weighted went from quadratic to cubic;

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

%***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);

\*\* this one below is different. weighted went from cubic to linear;

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

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);

/\*

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

ods html close; ods html;

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

%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, cub);

%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, quad);

%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, cub, cub);

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

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

\*\*\*\*\*\* may 23rd 2.5 fix;

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

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

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

\*/

\*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**;

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

ods html close; ods Html;

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

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

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

controlling for intracranial volume. Create mean measeure 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**;

**proc** **print** data = Percent\_categ2; **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 paramter 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 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 = "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;

goptions hsize=**6** vsize=**4.5**;

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

\*axis2 order=(1.5 to 4.5 by .5) minor=(n=1) label=(a=90 j=c h=1.75 f=simplex "&side" );

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

filename outfile “/Vol0222 &title..tiff";

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

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**;

ods html close; ods html;

%***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);

ods html close; ods html;

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

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

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

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

\*6/29/30 July 7th paper edits;

**proc** **means** data = wneuro\_safety3 range min max;

var MRI\_cort\_vol\_ctx\_total MRI\_cort\_vol\_ctx\_lh\_total MRI\_cort\_vol\_ctx\_rh\_total;

**run**;

**proc** **means** data = wneuro\_safety3 range min max;

var MRI\_cort\_area\_ctx\_total MRI\_cort\_area\_ctx\_lh\_total MRI\_cort\_area\_ctx\_rh\_total;

**run**;

**proc** **means** data = wneuro\_safety3 range min max;

var MRI\_cort\_thick\_ctx\_mean MRI\_cort\_thick\_ctx\_lh\_mean MRI\_cort\_thick\_ctx\_rh\_mean

totalSubCort\_vol;

**run**;

**proc** **freq** data=testMeanA;

table rage\*(opuW opW)/list;

**run**;