**Appendix S1: R Script to Clean the Project TALENT Data File**

# R script to process raw Project TALENT data from AIR

# and prepare a cleaned data file for analysis in Stata

#load libraries and set working directory

library(data.table)

library(plyr)

library(dplyr)

setwd("~/Desktop")

#read the sas7bdat file, create a CSV file from it, and clear memory

#reading and writing the data will take a few minutes (be patient)

raw = haven::read\_sas("all\_master0e\_rel.sas7bdat")

write.csv(raw, "allVars.csv", row.names = FALSE)

rm(list=ls())

#read data from the CSV file (skip lines 12-14 once it's created)

raw = data.frame(fread("allVars.csv"))

#name the relevant variables to extract

cogVars = c("BY\_R106", "BY\_R311", "BY\_R312", "BY\_R333",

 "BY\_R172", "BY\_R230", "BY\_R250",

 "BY\_R282", "BY\_R281", "BY\_R270", "BY\_R290")

otherVars = c("BY\_ID\_Rel", "BY\_TESTGRD", "BY\_SEX", "BY\_WTA",

 "Y11\_WTB", "Y11\_E002", "Y11\_O201")

#description of variables

# BY\_R106 = Mathematics Information

# BY\_R311 = Math I. Arithmetic reasoning

# BY\_R312 = Math II. Introductory high school mathematics

# BY\_R333 = Math III. Advanced high school mathematics

# BY\_R172 = Vocabulary Total

# BY\_R230 = English Total

# BY\_R250 = Reading comprehension

# BY\_R282 = Visualization in 3 dimensions

# BY\_R281 = Visualization in 2 dimensions

# BY\_R270 = Mechanical reasoning

# BY\_R290 = Abstract reasoning

# BY\_ID\_Rel = Student identification code

# BY\_TESTGRD = Grade in which student was enrolled at time of testing

# BY\_SEX = Sex (1 = male, 2 = female)

# BY\_WTA = 1960 student weight

# Y11\_WTB = Eleven-year follow-up weight B

# Y11\_E002 = Highest degree received

# Y11\_O201 = Current job-Project Talent 4-digit career code (see Appendix D)

#create a CSV file based on just these selected variables

d = raw[c(cogVars, otherVars)]

write.csv(d, "selectedVars.csv", row.names = FALSE)

#load data of the selected variables

d = read.csv("selectedVars.csv")

#helper function for reformatting cognitive variables

reformatCog = function(var) {

 var[ var == -9] = NA

 var[ var == -99] = NA

 var

}

#iterate through the cognitive variables and reformat

for (var in cogVars) d[var] = reformatCog(d[var])

#impute cognitive missing values using multiple imputation

set.seed(743)

impFormula = as.formula(paste("~", paste(cogVars, collapse = "+")))

imp = Hmisc::aregImpute(impFormula, data=d[cogVars],

 n.impute = 5, type = "regression")

#create list of five imputed datasets

imps = list()

for (i in 1:5) {

 #initialize using non-imputed datasets

 imps[[i]] = d[cogVars]

 #iterate through each variable

 for (var in cogVars) {

 #figure where to impute with what value

 missRows = imp$na[[var]]

 missVals = imp$imputed[[var]][,i]

 #impute values

 imps[[i]][missRows, var] = missVals

 }

}

#helper function to compute cognitive test composites (see Wai et al. 2009)

addCogComposites = function(data) {

 #compute sums

 data = mutate(data, mRaw = 0.55\*BY\_R106 + 1\*BY\_R311 + 0.55\*BY\_R312 + 1\*BY\_R333,

 sRaw = 3\*BY\_R282 + 1\*BY\_R281 + 1.5\*BY\_R270 + 2\*BY\_R290,

 vRaw = 2.5\*BY\_R172 + 1\*BY\_R230 + 1.25\*BY\_R250)

 #set sums to missing if there isn't complete cognitive data

 miss = !complete.cases(data[c("mRaw", "sRaw", "vRaw")])

 data$mRaw[miss] = NA

 data$sRaw[miss] = NA

 data$vRaw[miss] = NA

 #return the data frame

 return(data)

}

#compute test composites for original data file and imputed datasets

d = addCogComposites(d)

for (i in 1:5) imps[[i]] = addCogComposites(imps[[i]])

#create dummy codes for male, teacher, and college gradaute

d = mutate(d, male = 1\*(BY\_SEX==1),

 teacher = 1\*(Y11\_O201 %in% c(2411, 2412, 2421:2429)),

 college = 1\*(Y11\_E002 >= 3))

#create three categories of teachers

young = c(2411, 2412)

STEM = c(2421, 2422, 2428)

nonSTEM = c(2423:2427, 2429)

d$type <- ifelse(d$Y11\_O201 %in% young, 1,

 ifelse(d$Y11\_O201 %in% nonSTEM, 2,

 ifelse(d$Y11\_O201 %in% STEM, 3, 0)))

#description of selected occupation codes (see Appendix D)

# 2411: Teaching of young children - Nursery school or kindergarten teacher

# 2412: Teaching of young children - Elementary school teacher

# 2421: Secondary school teacher (including JHS) - Math

# 2422: Secondary school teacher (including JHS) - Science

# 2428: Secondary school teacher (including JHS) - Trade/industrial/vocational education

# 2423: Secondary school teacher (including JHS) - Social studies

# 2424: Secondary school teacher (including JHS) - English

# 2425: Secondary school teacher (including JHS) - Foreign language

# 2426: Secondary school teacher (including JHS) - Commercial education

# 2427: Secondary school teacher (including JHS) - Home economics

# 2429: Secondary school teacher (including JHS) - Physical education

#write helper functions for standarization using probabiltiy survey weights

wgtScale <- function(var, w) (var - weighted.mean(var, w))/sqrt(Hmisc::wtd.var(var, w))

#helper function to standardize cognitive composites

stdCogVars = function(data) {

 #standardize within grade cohort

 for (i in 9:12) {

 #determine ids to standarize in this iteration

 ids = data$BY\_TESTGRD == i

 ids\_c = (data$BY\_TESTGRD == i) & (data$college == 1)

 #make sure there are no missing elements

 ids[ is.na(ids) ] = FALSE

 ids\_c[ is.na(ids\_c) ] = FALSE

 #standarize relative to high school population

 data$math[ ids ] = scale(data$mRaw[ids])

 data$spatial[ ids ] = scale(data$sRaw[ids])

 data$verbal[ ids ] = scale(data$vRaw[ids])

 #standarize relative to college graduates (weighted means & SDs)

 data$math\_c[ ids\_c ] = wgtScale( data$mRaw[ids\_c], data$Y11\_WTB[ids\_c] )

 data$spatial\_c[ ids\_c ] = wgtScale( data$sRaw[ids\_c], data$Y11\_WTB[ids\_c] )

 data$verbal\_c[ ids\_c ] = wgtScale( data$vRaw[ids\_c], data$Y11\_WTB[ids\_c] )

 }

 #return modified data frame

 return(data)

}

#standardize the unimputed dataset

d = stdCogVars(d)

#standardize the imputed datasets

addVars = c("BY\_TESTGRD", "college", "Y11\_WTB")

for (i in 1:5) {

 #add needed variables and standarize

 imps[[i]][addVars] = d[addVars]

 imps[[i]] = stdCogVars(imps[[i]])

 #add cognitive composites to main data frame

 sumVars = c("math", "spatial", "verbal", "math\_c", "spatial\_c", "verbal\_c")

 d[paste0(sumVars, i)] = imps[[i]][sumVars]

}

#create an indicator variable whether cognitive variables was imputed

d = mutate(d, mImp = is.na(BY\_R106) + is.na(BY\_R311) + is.na(BY\_R312) + is.na(BY\_R333),

 sImp = is.na(BY\_R282) + is.na(BY\_R281) + is.na(BY\_R270) + is.na(BY\_R290),

 vImp = is.na(BY\_R172) + is.na(BY\_R230) + is.na(BY\_R250),

 anyImp = ((mImp>0) | (sImp>0) | (vImp>0)) \* 1)

#write the merged data frame as a CSV

#set missing data string to "" for exporting to Stata

write.csv(d, "cleaned.csv", row.names=FALSE, na="")

**Appendix S2: Stata Script to Analyze the Cleaned Data File**

\*Stata script to analyze data using the cleaned

\*data file produced by the clean.R script

\*load data

cd "~/Desktop"

insheet using cleaned.csv

\*cd "~/Downloads"

\*insheet using cleaned.csv

\*include only teachers with some spatial test data

keep if (teacher==1) & (simp<4)

\*use y11\_wtb as the probability survey weight

svyset by\_id\_rel [pweight=y11\_wtb]

\*\*get counts by type and sex (Table 1)

tab y11\_o201 type

svy: mean male, over(type)

svy: mean male if type==1, over(y11\_o201)

svy: mean male if type==2, over(y11\_o201)

svy: mean male if type==3, over(y11\_o201)

\*\*get counts by type and highest degree (Table 3)

tab y11\_e002 type

\*get means by teacher type (general population)

svy: mean spatial1 spatial2 spatial3 spatial4 spatial5, over(type)

svy: mean math1 math2 math3 math4 math5, over(type)

svy: mean verbal1 verbal2 verbal3 verbal4 verbal5, over(type)

\*get means by teacher type (college graduates)

svy: mean spatial\_c1 spatial\_c2 spatial\_c3 spatial\_c4 spatial\_c5, over(type)

svy: mean math\_c1 math\_c2 math\_c3 math\_c4 math\_c5, over(type)

svy: mean verbal\_c1 verbal\_c2 verbal\_c3 verbal\_c4 verbal\_c5, over(type)

\*get percent above average

gen spatialAbove1 = spatial1>0

gen spatialAbove2 = spatial2>0

gen spatialAbove3 = spatial3>0

gen spatialAbove4 = spatial4>0

gen spatialAbove5 = spatial5>0

gen spatialAbove\_c1 = spatial\_c1>0

gen spatialAbove\_c2 = spatial\_c2>0

gen spatialAbove\_c3 = spatial\_c3>0

gen spatialAbove\_c4 = spatial\_c4>0

gen spatialAbove\_c5 = spatial\_c5>0

svy: mean spatialAbove1 spatialAbove2 spatialAbove3 spatialAbove4 spatialAbove5, over(type)

svy: mean spatialAbove\_c1 spatialAbove\_c2 spatialAbove\_c3 spatialAbove\_c4 spatialAbove\_c5 if college==1, over(type)

\*run regression models

svy: regress spatial1 i.type

estimates store model1

lincom i3.type - i2.type

svy: regress spatial1 i.type math1 verbal1

estimates store model2

lincom i3.type - i2.type

svy: regress spatial\_c5 i.type math\_c5 verbal\_c5 male

estimates store model3

lincom i3.type - i2.type

\*view results

estimates table model1 model2 model3, b(%8.7f) se stats(N r2) drop(\_cons)

\*test for differences within aggregate groups

\*F ratios can't be combined using Rubin's rules

\*Hence, analyze the average of the five imputed outcomes

gen spatialAvg = (spatial1 + spatial2 + spatial3 + spatial4 + spatial5) / 5

svy: regress spatialAvg i.y11\_o201 if type==1

svy: regress spatialAvg i.y11\_o201 if type==2

svy: regress spatialAvg i.y11\_o201 if type==3

\*look at differences within secondary non-STEM teachers

tab y11\_o201 if type==2

svy: mean spatial1 spatial2 spatial3 spatial4 spatial5 if type==2, over(y11\_o201)

\*test for significant pairwise differences (w Bonferroni corrections)

svy: regress spatialAvg i.y11\_o201 if type==2

pwcompare y11\_o201, mcompare(bon)