

Quick References

Purpose: This document provides some useful information and supports for using the NLTS2 database, including some reference lists and coding tips.

Contents

[Instruments and Files in the NLTS2 Database](#)

[Demographics by Data Source and Wave](#)

[Questionnaire Topic Areas by Section Letters](#)

[Parent Interview Wave 1](#)

[Parent Interview \(Part 1 and Part 2\) Waves 2-4](#)

[Parent Interview \(Part 1 and Part 2\) Wave 5](#)

[Youth Interview \(Part 2\) Waves 2-4](#)

[Youth Interview \(Part 2\) Waves 5](#)

[Teacher Survey, Waves 1 and 2](#)

[School Program Survey Waves 1 and 2](#)

[School Characteristics Survey Wave 1](#)

[Parent/Youth Questionnaire Topic Areas Section Letters by Wave](#)

[Missing Values](#)

[Basic Programming Examples: SPSS Menu Driven](#)

[Basic Programming Examples: SPSS Syntax](#)

[SPSS Tips](#)

[Basic Programming Examples: SAS Syntax](#)

[SAS Tips](#)

Instruments and Files in the NLTS2 Database

Instrument	Wave	File Name*	Variable Prefix	Main Weight**
Parent Survey	1	n2w1parent	np1	np1Wt
Parent/Youth Survey	2	N2W2ParYouth	np2	np2Wt † np2YouthWt
	3	N2W3ParYouth	np3	np3Wt† np3YouthWt
	4	N2W4ParYouth	np4	np4Wt† np4YouthWt
	5	N2W5ParYouth	np5	np5Wt† np5YouthWt
Direct Assessment	2	N2W2DirAssess	nda2	wt_na
Alternate Assessment	2	N2W2AltAssess	naa2	wt_na
School Program Questionnaire	1	N2W1Prog	npr1	wt_NPR1
	2	N2W2Prog	npr2	wt_NPR2
Teacher Questionnaire	1	N2W1Tchr	nts1	wt_NTS1
	2	N2W2Tchr	nts2	wt_NTS2
School Characteristics Questionnaire	1	N2W1SchChar	nsc1	wt_NSC1
Secondary School Transcripts	[n/a]	N2Trans_Overall	nta	tr_Wt
		N2Trans_ByYear	nty	tr_Wt
		N2Trans_ByCourse	ntc	tr_Wt
		N2Trans_ByGrade	ntg	tr_Wt
		N2Trans_Summary	nts	tr_Wt
Cross-instrument Data	1	N2W1MultInstr	nxm1	***
	3	N2W3MultInstr	w3rp	
Sample file	[n/a]	n2sample	[n/a]	(see note)
<p>* Extension is ".sav" for SPSS and ".sas7bdat" for SAS files. For example, N2W2Prog would be "N2W2Prog.sav" in SPSS and "N2W2Prog.sas7bdat" in SAS</p> <p>** Replicate weights with similar names are also included in these files. Replicate weights are often used in software packages such as SUDAAN or Stata. The main weight will be used for all our examples.</p> <p>*** Weight to be selected from smallest sample</p> <p>† Use this weight for all parent/guardian only items and combined youth/parent/guardian items; the second weight listed is a youth weight to be used with youth only items.</p> <p>Note: the n2sample.[sav/sas7bdat] files contains the sample information (Stratum and Cluster) used in some procedures such as SPSS Complex Samples and SAS Proc Survey procedures</p>				

Demographics by Data Source and Wave

The following are the demographic variables commonly used for NLTS2 comparative analysis. Every file for a given source and wave has the appropriate set of “by-variables” included for that data source and wave.

Description	Source	Wave	Variable Name	Values	
District designated disability classification	Parent/ youth	1	w1_DisHdr2001	(1) Learning Disability (2) Speech Impairment (3) Mental Retardation (4) Emotional Disturbance (5) Hearing Impairment (6) Visual Impairment (7) Orthopedic Impairment (8) Other Health Impairment (9) Autism (10) Traumatic Brain Injury (11) Multiple Disabilities (12) Deaf/Blindness	
		2	W2_DisHdr2003		
		3	W3_DisHdr2005		
		4	W4_DisHdr2007		
		5	W5_DisHdr2009		
	School*	1	w1_dis12		
		2	w2_Dis12		
	Transcript **	—	tr_DisHdr		
Student's race/ethnicity	Parent/ youth	1	w1_EthHdr2001		(1) White (2) African American (3) Hispanic (4) Asian/Pacific Islander (5) American Indian/Alaska Native (6) Multiple/Other
		2	W2_EthHdr2003		
		3	W3_EthHdr2005		
		4	W4_EthHdr2007		
		5	W5_EthHdr2009		
	School*	1	w1_eth6		
		2	w2_Eth6		
	Transcript	[n/a]	tr_EthHdr		
Student's gender	Parent/ youth	1	w1_GendHdr2001	(1) Male (2) Female	
	Parent/ youth	2	W2_GendHdr2003		
		3	W3_GendHdr2005		
		4	W4_GendHdr2007		
		5	W5_GendHdr2009		
		School*	1		w1_gend2
	2		w2_Gend2		
	Transcript	[n/a]	tr_GenHdr		

Description	Source	Wave	Variable Name	Values
Household income	Parent/ youth	1	w1_IncomeHdr2001	(1) \$25,000 and Under (2) \$25,001 to \$50,000 (3) Over \$50,000
		2	W2_IncomeHdr2003	
		3	W3_IncomeHdr2005	
		4	W4_IncomeHdr2007	
		5	W5_IncomeHdr2009	
	School*	1	w1_incm3	
		2	w2_Incm3	
	Transcript		tr_IncomeHdr	
Transcript by grade	[n/a]	Tr_IncomeByGrade		
Urbanicity of school student attends†	School*	1	w1_urb3	(1) Rural (2) Suburban (3) Urban
		2	w2_Urb3	
Student's grade level†	School*	1	w1_grade4	(1) 7 th – 8 th grade (2) 9 th grade (3) 10 th grade (4) 11 th – 12 th grade [(.u) ungraded]
		2	w2_Grade4	(1) 10 th or below (2) 11 th grade (3) 12 th grade (4) Ungraded
	Assessment	2***	na_Grade4	(4) Ungraded
	Transcript course level	[n/a]	ntcGrLevel	(9) 9 th grade (10) 10 th grade (11) 11 th grade (12) 12 th grade (13) Grade 13 (14) Ungraded
	Transcript Grade level		ntgGrLevel	
Youth's age at time of data collection	Parent Survey	1	W1_Age2001	Age of youth at time of data collection
	Parent/Youth Survey	2	W2_Age2003	
		3	W3_Age2005	
		4	W4_Age2007	
		5	W5_Age2009	
	School	1	W1_Age	
		2	W2_Age2004	
	Assessment	2***	nda_Age (direct) naa_Age (alternate)	
Transcript	[n/a]	tr_Age[1996-2007]_1201		
Student's secondary school leaving status	Transcript	[n/a]	tr_LvStat	(1) Completer (graduated, aged out, or eventually completed) (2) Dropped out

Description	Source	Wave	Variable Name	Values
Youth's age category at time of data collection	Parent/ youth	1	np1AgeHdr or w1_AgeHdr2001	(1) 13-14 (2) 15 (3) 16 (4) 17
		2	W2_AgeHdr2003	(1) 16 (2) 17 (3) 18 (4) 19-20
		3	W3_AgeHdr2005	(1) 17 (2) 18 (3) 19 (4) 20-21
		4	W4_AgeHdr2007	(1) 19 (2) 20 (3) 21 (4) 22-23
		5	W5_AgeHdr2009	(1) 21-22 (2) 23 (3) 24 (4) 25
	School*	1	w1_age4	(1) 14 (2) 15 (3) 16 (4) 17-18
		2	w2_Age4	(1) 16
	Assessment	2***	na_Age4	(2) 17
	Transcript	[n/a]	[n/a]††	(3) 18 (4) 19-20
	<p>* School source: Teacher, School Program, and School Characteristics Surveys.</p> <p>** Transcripts: Transcripts have multiple files and can represent multiple years. Transcript data was released in Wave 5 but collected for the years students attended secondary school.</p> <p>*** Assessment age and assessment grade are the age or grade of the student when student was assessed and may differ from values in other school data for Wave 2. Student was assessed in either Wave 1 or Wave 2 based on his or her age in that wave; all assessment data released in Wave 2.</p> <p>† Grade and urbanicity created for school and assessment data only, not applicable for parent/youth interview data or transcripts.</p> <p>†† Age category was not created or used for Transcript data analysis. Age as of December 1 was calculated for each school year found in the NLTS2 transcript data (see tr_Age[1996-2007]_1201).</p>			

Questionnaire Topic Areas by Section Letters

Section Letter	NLTS2 Data Collection Instruments Questionnaire Topic Area
Parent Interview Wave 1	
S	CATI screening
A	Student characteristics
B	Disability characteristics/health
C	Health insurance
D	Secondary and postsecondary school
E	Family interaction/involvement
F	Social and extracurricular activities
G	Youth behaviors
H	Services
I	Employment
I	Screen for continuation/overlap items
J	Expectations: [parent/youth] expectations and youth beliefs
K	Household characteristics (parent/guardian's)
Parent Interview (Part 1 and Part 2) Waves 2-4	
Z	CATI screening
A	Student characteristics
B	Disability characteristics/health
C	Health insurance
D	School experiences
E	Family interaction/involvement
F	Services
G	Parent expectations and youth behaviors
H	Household characteristics (parent/guardian's)
I	Screen for continuation/overlap items
J	Youth behaviors
K	Postsecondary school
K	Secondary school
L	Employment
M	Household characteristics (youth's)
Parent Interview (Part 1 and Part 2) Wave 5	
Z	CATI screening
A	Residential, school, and employment status
B	Change in functioning
C	Services

Section Letter	NLTS2 Data Collection Instruments Questionnaire Topic Area
H	Household income
I	Screen for continuation and closing
J	Social and extracurricular activities
K	Postsecondary education
L	Employment
M	Youth household characteristics
Youth Interview (Part 2) Waves 2-4	
O	CATI screening
P	Social and extracurricular activities
Q	Disability characteristics/health
R	School experiences
S	Postsecondary school
T	Employment
U	Youth behaviors
V	Expectations: [parent/youth] expectations and youth beliefs
W	Household characteristics (youth's)
Youth Interview (Part 2) Waves 5	
O	CATI Screening
P	Social and extracurricular activities and health related items
Q	Disability characteristics/health
S	Postsecondary education
T	Part 1: Employment
	Part 2: Services
U	Risk behaviors
V	Young adult's beliefs
W	Young adult's household characteristics
Teacher Survey, Waves 1 and 2	
A	About This General Education Academic Class
B	About the Instruction This Student Receives in This Class
C	Student Performance and Family Support
D	About You
School Program Survey Waves 1 and 2	
A	Overview of This Student's School Program
B	Student Performance and Family Support
C	Career and Vocational Education and Services
D	Educational Services
E	Transition to Adult Life
F	About the Respondent

Section Letter	NLTS2 Data Collection Instruments Questionnaire Topic Area
School Characteristics Survey Wave 1	
A	School Characteristics
B	Student Characteristics
C	Staff, Programs, and Resources
D	Special Education Policies and Practices
E	Moving On (for Elementary, Middle, or K-8 Schools Only)
F	About Schools Serving 12th Grade Students

Parent/Youth Questionnaire Topic Areas Section Letters by Wave

Parent/Youth Survey Questionnaire Topic Areas	Parent			Youth	
	Wave 1	Waves 2-4	Wave 5	Waves 2-4	Wave 5
CATI screening	S	Z	Z	O	O
Expectations	J	G (W2 and W3)	—	V (W2 and W3)	—
Family interaction/involvement	E	E	—	—	—
Health	B	B	B	Q	Q
Health insurance and benefits	C, K	C, I, M	M	W	W
Overlap items	I	I	—	—	—
Parent/guardian's household characteristics	K	H	H	—	—
Postsecondary school	D	D, K	A, K	S	S
Screen for continuation	I	I	I	—	—
Secondary school	D	D, K	A	R	—
Youth's behaviors	G	G, J	—	U	U
Youth's beliefs	—	—	—	V	V
Youth's change in functioning	—	B (W4)	B	—	—
Youth's characteristics	A	A	—	—	—
Youth's disability characteristics	B	B	—	—	—
Youth's employment	I	L	A, L	T	T (Part 1)
Youth's household characteristics	—	M	M	W	W
Youth's residential status	A	A	A	P	P
Youth's services	H	F	C	—	T (Part 2)
Youth's social and extracurricular activities	F	J	J	P	P

Missing Values

Data can be missing for various reasons in the NLTS2 database. Files associated with each data collection include only those who completed the survey or assessment; the number of respondents varies from file to file. Within a file, respondents may have missing responses to individual items. User defined missing values indicate the reasons items are missing for respondents. In SAS, user defined missing values begin with a period followed by a letter. SPSS does not support the number of discrete missing values that are designated in the NLTS2 database, but SPSS does support using a numeric range of missing values. Missing values in SPSS range from -999 to -900 and in SAS from .a to .z. The following table lists the missing values in the NLTS2 database.

Missing Values		
SAS	SPSS	Missing Value
.a	-999	Not applicable
.b	-978	Not asked this section
.d	-998	Do not know
.g	-990	Sparse data
.i	-993	In secondary school: returned to HS
.m	-997	Mail
.p	-996	Partial
.r	-995	Refused
.s	-994	Skipped
.u	-986	Ungraded
.v	-982	Version
.x	-992	Not in NCES data
.z	-980	Not ascertained

Basic Programming Examples: SPSS Menu Driven

Note: The following examples are for guidance only and not meant to replace software documentation; please refer to your installation's SPSS documentation for instructions specific to your version. The following examples use Version 15 Basic and Complex Samples modules. Be aware that menus and syntax can vary in wording and order from one version to the next and that the examples below are provided for suggested use only.

Operation	Step-by-Step Menu Instruction
Open File	Menu: <i>File</i> Open file [file name].sav from "File" menu
Percentages	Menu: Analyze: Descriptive: Descriptive Statistics From within "Descriptive Statistics," select "Frequencies" Locate [variable(s)] in the variable list and click the right-facing arrow to select it Press "OK" to run, or Optionally: select "Paste" to run code from syntax editor
Means	Menu: Analyze: Descriptive Statistics From within "Descriptive Statistics," select "Descriptives" Locate [<i>variable name</i>] in the variable list and click the right-facing arrow to select it Click on "options" and select "Mean," "Minimum," and "maximum" Press "OK" to run Option: select "Paste" to run code from syntax editor
Crosstabulations: Percentages	Menu: Analyze: Descriptive Statistics From within "Descriptive Statistics," select "Crosstabs" Locate [<i>percentages/row variable name</i>] in the variable list and click the right-facing arrow to the left of the "row(s)" box to select it Locate [<i>comparative/column variable name</i>] in the list and click the right-facing arrow next to the "Column(s)" box Click on "cells" at bottom of menu When the next menu pops up, click on the "Column" box under "Percentages" Press "OK" to run or select "Paste" to run code from syntax editor

Operation	Step-by-Step Menu Instruction
Crosstabulations: Means	<p>Menu: Analyze: Compare Means</p> <p>Select "Means" in subsequent menu</p> <p>Locate [<i>means variable name</i>] in the variable list and click the right-facing arrow to the left of the "Dependent list" box to select it</p> <p>Locate [<i>comparative variable name</i>] in the list and click the right-facing arrow next to the "Independent" box</p> <p>Click on "options" and under "cell statistics" select "Standard Deviation"</p> <p>Click left facing arrow</p> <p>Click continue</p> <p>Press "OK" to run or select "Paste" to run code from syntax editor</p>
Weighting cases	<p>Menu: Data: Weight Cases</p> <p>Click on the "weight cases by" radio button</p> <p>Locate [<i>weight variable</i>] in the list of variables</p> <p>Click on the button with right-facing arrow to select the weight</p> <p>Press "OK" to run or select "Paste" to run code from syntax editor</p>
Remove weight	<p>Menu: Data: Weight Cases</p> <p>Click on "Do not weight cases" radio button</p> <p>Press "OK" to run or select "Paste" to run code from syntax editor</p>
Add all variables from an external file to active file	<p>Menu: Data: Merge Files: Add Variables</p> <p>Select external file from browser window and click "open"</p> <p>Select "ID" in "Excluded Variables" box and click on the following "Match cases on key variables in sorted file"</p> <p>"External file is keyed table" radio button [in some versions the tag may read, "Non-active dataset is keyed table"]</p> <p>Left facing arrow next to "Key Variables"</p> <p>Press "OK" to run or select "Paste" to run code from syntax editor</p> <p>Note: this keeps only cases that match those present in the active file</p>
Add selected variables to the active file [matching to active file cases]	<p>Menu: Data: Merge Files: Add Variables</p> <p>Select external file from browser window and click "open"</p> <p>Select "ID" in "Excluded Variables" box and click on the following "Match cases on key variables in sorted file"</p> <p>"External file is keyed table" radio button [in some versions the tag may read, "Non-active dataset is keyed table"]</p> <p>Left facing arrow next to "Key Variables"</p> <p>Select variables in "New Working Data file" and click left-facing arrow between "Excluded Variables" and "New working..." to move them to the excluded box</p> <p>Use shift-click to select and move several variables at one time</p> <p>Variables in current file are marked with a (*) and those coming from the external file are marked with a (+)</p> <p>Only the variables in the "included" box will be in the new file</p>

Operation	Step-by-Step Menu Instruction
Create new file from existing file	<p>Menu: File: Save as</p> <p>Click on “Variables” box and in next menu click on “drop all”</p> <p>In the box with the variable list, click in little box to select only variables needed from this file</p> <p>Click “Continue,” give the file a new name in the browse window, and click “Save”</p> <p>Starting with this smaller file, merge data as we learned to do before, selecting only what is needed from the other file(s)</p>
Save the new file	<p>Menu: File: Save as</p> <p>Give the file a new name</p> <p>If a new file has already been created with a new name using the steps outlined above, select “Save” under the “File” menu</p>
Create a new variable	<p>Menu: Transform: Compute</p> <p>Enter a variable name under “Target Variable”</p> <p>If applicable, find and select the source variable(s) and click the right-facing arrow to move the variable name into the “Numeric Expression” box</p> <p>Enter functions/operations from the keypad boxes or the select from the list of functions</p> <p>If applicable, click on "if" by "optional case selection condition" and build conditional statements</p> <p>Click “OK” or “Paste”</p> <p>If multiple "if" conditions [if-then-else], then repeat the above steps, ending with the overriding condition</p> <p>Click "Type and Label" to assign a variable label</p>

Operation	Step-by-Step Menu Instruction
Recode existing variable	<p>Menu: Transform: Recode Into Different Variable</p> <p>To collapse categories, break a continuous variable into categories, or recode a variable, it is necessary to create a new variable</p> <p>Select variable to be recoded from list and click on right pointing arrow</p> <p>Give the new variable a name in the box under "Output variable"</p> <p>Assign a label to the new variable in the "Label" box under "Output variable"</p> <p>Click "change"</p> <p>Click on the box marked "Old and New Values" and a new box pops up</p> <p>In the new box, under "Old values" Click on radio button for "System or user-missing," click on "System missing" under "New values," and click "Add" next to "Old — >New"</p> <p>For each old to new value(s)</p> <p>Under "old values," click on a radio button by an actual value or range of values box</p> <p>Designate what the old values are, either actual or range of values, in the appropriate box</p> <p>Assign a new code under "New values" and click "Add"</p> <p>When finished with values, click "continue" to return to first box</p> <p>In original box, click "OK" or Paste" to generate code</p>
Adding value labels to new/recoded variables	<p>Menu: [none]</p> <p>In variable view, click on the cell in the "Values" column to bring up a new box</p> <p>Enter a value in the "Value" box, a label for that value in the "Label" box, and click "Add"</p> <p>Do this for every value</p>
Assign value labels	<p>Menu: [none]</p> <p>Click in format column in variable view</p> <p>For each possible value, enter value, label for value, and click add</p>
Assign missing values	<p>Menu: [none]</p> <p>Click in missing values column in variable view</p> <p>Assign range (-999 — -900) or value</p>

Operation	Step-by-Step Menu Instruction
Create a plan file for Complex Samples*	<p>Menu: Analyze: complex samples: prepare for analysis</p> <p>Select "create a plan file" and "browse" to assign a name and location of the plan file in the pop-up window</p> <p>Click next to go to the "Stage 1 Design Variables" window</p> <p>Select "Strata" and click right facing arrow to move the variable to the "Strata" box</p> <p>Select "Cluster" and click right facing arrow to move the variable to the "Clusters" box</p> <p>Select the appropriate weight and click right facing arrow to move the variable to the "Sample Weight" box</p> <p>Click next to go to "Stage 1 Estimation Method" window</p> <p>Select "next" and "WR" for with replacement</p> <p>Click finish</p>
Frequencies in Complex Samples*	<p>Menu: Complex samples: Frequencies</p> <p>Select sample plan file</p> <p>Select "open" and "continue"</p> <p>Select "statistics" and "table percent" in pop-up window</p> <p>Select variable(s) for and click right-facing arrow to move to the "frequency tables" box</p> <p>Click OK or paste to run from syntax editor</p>
Crosstabs in Complex Samples*	<p>Menu: Analyze: Complex samples: Crosstabs</p> <p>Select sample plan file</p> <p>Select "open" and "continue"</p> <p>Select "statistics" and "column percent" in pop-up window</p> <p>Select the comparative "by" variable for "column" and the analysis variables for "row" by selecting variables and clicking the appropriate right-facing arrow</p> <p>Click OK or paste to run from syntax editor</p>
Means in Complex Samples*	<p>Menu: Analyze: complex samples: descriptives</p> <p>Select sample plan file</p> <p>Select "open" and "continue"</p> <p>Select the variable for "measures"</p> <p>Click OK or paste to run syntax</p>
Comparative means in Complex Samples*	<p>Menu: Analyze: complex samples: descriptives</p> <p>Select sample plan file</p> <p>Select "open" and "continue"</p> <p>Select the variable and click right-facing arrow for "measures" and comparative variable for "subpopulations"</p> <p>Click OK or paste to run syntax</p>

Operation	Step-by-Step Menu Instruction
Regression	<p>Menu: Analyze: regression: linear</p> <p>Locate dependent variable in the variable list and click the right-facing arrow to select and move to the "dependent" box</p> <p>Locate independent [variable(s)] in the variable list and click the right-facing arrow to select and move to the "Independent(s)" box</p> <p>Click OK or paste to run syntax</p>
Correlation	<p>Menu: Analyze: correlate: bivariate</p> <p>Locate [variable(s)] in the variable list and click the right-facing arrow to select and move to the "Variables(s)" box</p> <p>Click OK or paste to run syntax</p>
<p>* Weighted standard errors are estimates; SPSS Complex Samples procedures may calculate similar but slightly different standard errors than SAS PROC SURVEY or other similar procedures for estimating weighted standard errors. See "Weighted Standard Errors" in the "NLTS2 Database Structure" document for more discussion about estimated standard errors.</p>	

Basic Programming Examples: SPSS Syntax

Note: The following examples are for guidance only and not meant to replace software documentation; please refer to your installation's SPSS documentation for instructions specific to your version. The following examples use Version 15 Basic and Complex Samples modules. Be aware that menus and syntax can vary in wording and order from one version to the next and that the examples below are provided for suggested use only.

Operation	Syntax/Program Code
Open File	<p>Syntax: Get file '[path]:\[file name].sav.'</p> <p>Example: Get file 'C:\myproj\NLTS2\Data\n2w1tchr.sav'.</p>
Comment	<p>Syntax: * [comment].</p> <p>Example: * Create variable to collapse months into 4 categories. * Subset to exclude those in ungraded program.</p>
Percentages	<p>Syntax: FREQUENCIES VARIABLES=[variable name(s)] /ORDER= ANALYSIS .</p> <p>Example: FREQUENCIES VARIABLES= w2_dis12 w2_gend2 w2_incm3 /ORDER= ANALYSIS .</p>
Means	<p>Syntax: DESCRIPTIVES VARIABLES=[variable name(s)] /STATISTICS=MEAN MIN MAX .</p> <p>Example: DESCRIPTIVES VARIABLES= ndaPC_w /STATISTICS=MEAN MIN MAX .</p>

Operation	Syntax/Program Code
Crosstabulations: Percentages	<p>Syntax:</p> <pre>CROSSTABS /TABLES=[percentage/row variable] BY [comparative/column variable] /FORMAT= AVALUE TABLES /CELLS= COUNT COLUMN /COUNT ROUND CELL .</pre> <p>Example:</p> <pre>CROSSTABS /TABLES= npr1E1 BY w1_incm3 /FORMAT= AVALUE TABLES /CELLS= COUNT COLUMN /COUNT ROUND CELL .</pre>
Crosstabulations: Means	<p>Syntax:</p> <pre>MEANS TABLES=[means variable] BY [comparison variable] /CELLS MEAN COUNT .</pre> <p>Example:</p> <pre>MEANS TABLES= ndaPC_w BY na_grade4 /CELLS MEAN COUNT .</pre>
Weight cases	<p>Syntax:</p> <pre>WEIGHT BY [weight variable] .</pre> <p>Example:</p> <pre>WEIGHT BY wt_nt1 .</pre>
Remove weight	<p>Syntax:</p> <pre>WEIGHT OFF.</pre>
Add selected variables to the active file [matching to active file cases]	<p>Syntax:</p> <pre>MATCH FILES /FILE='[* for current file] or {path}\{file name}.sav' /TABLE='{path}\{external file name}.sav ' /BY [key variable] . EXECUTE.</pre> <p>Example:</p> <pre>MATCH FILES /FILE=* /TABLE='C:\myproj\NLTS2\Data\n2w1tchr.sav' /BY ID . EXECUTE.</pre>

Operation	Syntax/Program Code
<p>Add selected variables to the active file and keep only those variables needed</p>	<p>Syntax:</p> <pre>MATCH FILES /FILE=[* for current file or [path]\[file name].sav] /TABLE='[path]\[file name].sav' /BY [key variable] /KEEP=[key variable] [weight(s)] [variable(s)] . EXECUTE.</pre> <p>Note: Always keep "ID" (so that you can add more variables to the file later) and an analysis weight.</p> <p>Example:</p> <pre>MATCH FILES /FILE=* /TABLE= 'C:\myproj\NLTS2\Data\n2sample.sav' /BY ID /KEEP=ID np4Wt np4E2a np4E2c Cluster Strata . EXECUTE.</pre>
<p>Create new file from existing files</p>	<p>Syntax:</p> <pre>MATCH FILES /FILE='[path]\[file name].sav' /TABLE='[path]\[file name].sav' /TABLE='[path]\[file name].sav' /BY ID /KEEP=[key variable] [weight(s)] [variable(s)]. EXECUTE.</pre> <p>Note:</p> <p>"FILE=" keeps all cases in this file</p> <p>"TABLE=" will keep cases only if they match the cases in the "FILE"</p> <p>"KEEP=" lists only the variables needed in the new file</p> <p>Always keep "ID" so that more data may be added to the new file</p> <p>Example:</p> <pre>MATCH FILES /FILE='c:\myproj\NLTS2\Data\n2w4paryouth.sav' /TABLE='c:\myproj\NLTS2\Data\n2w2dirassess.sav' /TABLE='c:\myproj\NLTS2\Data\n2sample.sav' /BY ID /KEEP=ID np4Wt np4E2a np4E2c ndasci_pr ndass_pr Cluster Strata . EXECUTE.</pre>
<p>Save the new file</p>	<p>Syntax:</p> <pre>SAVE OUTFILE='[path]\[file name].sav'.</pre> <p>Example:</p> <pre>SAVE OUTFILE='c:\myproj\NLTS2\Data\File01.sav'.</pre>

Operation	Syntax/Program Code
Create a new variable	<p>Syntax:</p> <p>[optional condition statement]</p> <p>[<i>new variable</i>] = [existing variable] [function or operation] [old variable] .</p> <p>VARIABLE LABELS [<i>variable name</i>] '[<i>variable label</i>]'.</p> <p>Example:</p> <pre>IF (np1F4 = 0 or np2P6_J2_I4 = 0 or np3P6_J2_I4 = 0 or np4P6_J2 = 0) GrpActiv_ever = 0 . IF (np1F4 = 1 or np2P6_J2_I4 = 1 or np3P6_J2_I4 = 1 or np4P6_J2 = 1) GrpActiv_ever = 1 .</pre> <p>VARIABLE LABELS GrpActiv_ever '(GrpActiv_ever) Youth ever participated in community group or activity'. EXECUTE .</p>
Recode existing variable	<p>Syntax:</p> <p>RECODE [<i>existing variable</i>] (MISSING=SYSMIS) ([<i>old value</i>] = [<i>new value</i>]) ([<i>old</i>] = [<i>new</i>]) ([<i>old</i>] = [<i>new</i>]) INTO [<i>new variable</i>] .</p> <p>VARIABLE LABELS [<i>new variable name</i>] '[<i>new variable label</i>]'.</p> <p>Example:</p> <pre>RECODE npr1B4 (MISSING=SYSMIS) (Lowest thru 1=1) (2 thru 5=2) (6 thru Highest=3) INTO npr1B4_Cat . VARIABLE LABELS npr1B4_Cat '(npr1B4_Cat) Days absent per month categorized'. EXECUTE .</pre>
Adding value labels to new/recoded variables	<p>Syntax:</p> <p>value labels [<i>variable name</i>] [<i>value</i>] "[<i>value label</i>]" [<i>value</i>] "[<i>value label</i>]" [<i>value</i>] "[<i>value label</i>]" [<i>value</i>] "[<i>value label</i>]" .</p> <p>Example:</p> <pre>value labels npr1B4_Cat 1 "(1) None or 1 day absent" 2 "(2) 2 to 5 days absent" 3 "(3) More than 5 days absent".</pre>
Create a plan file for Complex Samples*	[use wizard; see menu driven instructions above]

Operation	Syntax/Program Code
Frequencies in Complex Samples*	<p>Syntax:</p> <p>* Complex Samples Frequencies.</p> <pre>CSTABULATE /PLAN FILE = '[path]\[file name].csaplan' /TABLES VARIABLES = [variable] /CELLS POPSIZE TABLEPCT /STATISTICS SE /MISSING SCOPE = TABLE CLASSMISSING = EXCLUDE.</pre> <p>Example:</p> <pre>* Complex Samples Frequencies. CSTABULATE /PLAN FILE = 'C:\Projects\Data\MyPlan.csaplan' /TABLES VARIABLES = w2_incm3 /CELLS POPSIZE TABLEPCT /STATISTICS SE /MISSING SCOPE = TABLE CLASSMISSING = EXCLUDE.</pre>
Crosstabs in Complex Samples*	<p>Syntax:</p> <p>* Complex Samples Crosstabs.</p> <pre>CSTABULATE /PLAN FILE = '[path]\[file name].csaplan' /TABLES VARIABLES = [variable] BY [by variable] /CELLS POPSIZE COLPCT /STATISTICS SE /MISSING SCOPE = TABLE CLASSMISSING = EXCLUDE.</pre> <p>Example:</p> <pre>* Complex Samples Crosstabs. CSTABULATE /PLAN FILE = 'C:\Projects\Data\MyPlan.csaplan' /TABLES VARIABLES = w2_eth6 BY w2_incm3 /CELLS POPSIZE COLPCT /STATISTICS SE /MISSING SCOPE = TABLE CLASSMISSING = EXCLUDE.</pre>
Means in Complex Samples*	<p>Syntax:</p> <p>* Complex Samples Descriptives.</p> <pre>CSDESCRIPTIVES /PLAN FILE = '[path]\[file name].csaplan' /SUMMARY VARIABLES =[variable] /MEAN /STATISTICS SE /MISSING SCOPE = ANALYSIS CLASSMISSING = EXCLUDE.</pre> <p>Example:</p> <pre>* Complex Samples Descriptives. CSDESCRIPTIVES /PLAN FILE = 'C:\Projects\Data\MyPlan.csaplan' /SUMMARY VARIABLES = npr1B4 /MEAN /STATISTICS SE /MISSING SCOPE = ANALYSIS CLASSMISSING = EXCLUDE.</pre>

Operation	Syntax/Program Code
Comparative means in Complex Samples*	<p>Syntax:</p> <p>* Complex Samples Descriptives. CSDESCRIPTIVES /PLAN FILE = '[path]\[file name].csaplan' /SUMMARY VARIABLES =[variable] /SUBPOP TABLE = [by variable] DISPLAY=LAYERED /MEAN /STATISTICS SE /MISSING SCOPE = ANALYSIS CLASSMISSING = EXCLUDE.</p> <p>Example:</p> <pre>* Complex Samples Descriptives. CSDESCRIPTIVES /PLAN FILE = 'C:\Projects\Data\MyPlan.csaplan' /SUMMARY VARIABLES = npr1B4 /SUBPOP TABLE = w1_incm3 DISPLAY=LAYERED /MEAN /STATISTICS SE /MISSING SCOPE = ANALYSIS CLASSMISSING = EXCLUDE.</pre>
Regression	<p>Syntax:</p> <pre>REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT [dependent variable] /METHOD=ENTER [independent variable(s)] .</pre> <p>Example:</p> <pre>REGRESSION /MISSING LISTWISE /STATISTICS COEFF OUTS R ANOVA /CRITERIA=PIN(.05) POUT(.10) /NOORIGIN /DEPENDENT ndaPC_ss /METHOD=ENTER nda_Age W2_Inc_LT25 W2_Inc_LT50 W2_Gen_Male .</pre>
Correlation	<p>Syntax:</p> <pre>CORRELATIONS /VARIABLES=[variables] /PRINT=TWOTAIL NOSIG /MISSING=PAIRWISE.</pre> <p>Example:</p> <pre>CORRELATIONS /VARIABLES=ndaPC_ss nda_Age W2_Inc_LT25 W2_Inc_LT50 W2_Gen_Male /PRINT=TWOTAIL NOSIG /MISSING=PAIRWISE .</pre>

Operation	Syntax/Program Code
<p>* Weighted standard errors are estimates; SPSS Complex Samples procedures may calculate similar but slightly different standard errors than SAS PROC SURVEY or other similar procedures for estimating weighted standard errors. See “Weighted Standard Errors” in the “NLTS2 Database Structure” document for more discussion about estimated standard errors.</p>	

SPSS Tips

Most programming commands operate on the file that is currently active.

Select the programming command in the syntax editor and click the right facing blue arrow to run statements or run statements the “Run” menu.

Programming statements end in a “.” [period] and sometimes require an “Execute.” statement.

If unsure of syntax, use menu driven functions and select the "paste" option to generate the code.

Code generated from the "paste" option can be modified and—this is very important—can be replicated.

Key variable to join files is "ID".

Basic Programming Examples: SAS Syntax

Note: The following examples are for guidance only and not meant to replace software documentation; please refer to your installation's SAS documentation for instructions specific to your version. The following examples use Version 9 Base and Stat packages. Be aware that menus and syntax can vary in wording and order from one version to the next and that the examples below are provided for suggested use only.

Operation	Syntax/Program Code
Library locations	<p>Syntax:</p> <pre>LIBNAME [ddname] "[path]" ; /* location of data files */ LIBNAME library "[path]" ; /* location of format library */</pre> <p>Example:</p> <pre>libname sasdb "C:\myproj\NLTS2\Data" ; libname library "C:\myproj\NLTS2\Data" ;</pre>
View File	<p>Syntax:</p> <pre>LIBNAME [ddname] "[path]" ; PROC CONTENTS DATA = [ddname].[SAS file name] ;</pre> <p>Example:</p> <pre>libname sasdb "C:\myproj\NLTS2\Data" ; Proc contents data = sasdb.n2w1tchr ;</pre>
Comment	<p>Syntax:</p> <pre>* [comment]; /* [comment] */</pre> <p>Example:</p> <pre>* Create variable to collapse months into 4 categories; /* Subset to exclude those in ungraded program */</pre>
Percentages	<p>Syntax:</p> <pre>PROC FREQ DATA = [ddname.file] ; TABLES [variable name(s)] ;</pre> <p>Example:</p> <pre>Proc Freq data = sasdb.n2W2tchr; tables w2 dis12 w2 gend2 w2 incm3;</pre>
Means	<p>Syntax:</p> <pre>PROC MEANS DATA = [ddname.file] MEAN MIN MAX MAXDEC = 2 ; VAR [variable name(s)] ;</pre> <p>Note: options "mean min max maxdec=2" limits the output to mean, minimum value, and maximum value with 2 decimal places</p> <p>Example:</p> <pre>proc means data = sasdb.n2w2dirassess mean min max n maxdec=2; var ndaPC w;</pre>

Operation	Syntax/Program Code
Crosstabulations: Percentages	<p>Syntax:</p> <pre>PROC FREQ DATA = [ddname.file] ; TABLES =[percentage/row variable] * [comparative/column variable] /[table options] ;</pre> <p>Example:</p> <pre>/* prints missing and in list format */ proc freq data = sasdb.n2w1prog ; tables npr1E1 * w1_incm3 /missprint list; /* column percentages and cell counts only */ proc freq data = sasdb.n2w1prog ; tables npr1E1 * w1_incm3 /norow nopercent nocum;</pre>
Crosstabulations: Means	<p>Syntax:</p> <pre>PROC MEANS DATA = [ddname.file] MEAN MIN MAX MAXDEC = 2 ; CLASS [comparison variable] ; var [means variable] ;</pre> <p>Example:</p> <pre>proc means data = sasdb.n2w2dirassess mean min max n maxdec=2; class na_grade4; var ndaPC w ;</pre>
Weight cases	<p>Syntax:</p> <pre>WEIGHT [weight variable] ;</pre> <p>Example:</p> <pre>Proc Freq data = sasdb.n2W2tchr; weight wt_nts2 ; tables w2_dis12 w2_gend2 w2_incm3;</pre>
Remove weight	<p>Syntax:</p> <pre>*WEIGHT[weight variable] ;</pre> <p>Note: this comments out the statement so that it does not execute. The other option is to omit the statement.</p> <p>Example:</p> <pre>Proc Freq data = sasdb.n2W2tchr; * weight wt_nts2 ; /* or delete this line */ tables w2_dis12 w2_gend2 w2_incm3;</pre>

Operation	Syntax/Program Code
Add selected variables to the active file [matching to active file cases]	<p>Syntax:</p> <pre>DATA [NEWfile] ; MERGE [ddname.file] (in=[logical variable name]) [ddname.file] (in=[logical variable name]) ; BY [key variable]; RUN;</pre> <p>Note:</p> <p>Always keep "ID" so that more data may be added to the new file and the appropriate analysis weight(s).</p> <p>The "in=[dummy variable name]" option on the input data set creates a temporary logical value that is "1" if a case is in this file and a "0" if it is not in the file. If a case is in file1 and in file2 then the case will have a value of "1" for both logical variables. If the case is in file1 but there is no match for the case in file2 then the value will be "1" for the file1 variable and "0" for the file2 variable. These logical variables can be used in any statement in the data step and can give control over which cases are kept in the output dataset.</p> <p>Example:</p> <pre>/* keeps cases only if they are in both files */ Data Myfile ; merge sasdb.n2w1tchr (in=intchr) sasdb.n2w2dirassess (in=indir); BY ID; if intchr and indir ; Run;</pre> <pre>/* keeps if in teacher file, can be missing in assessment file */ Data Myfile ; merge sasdb.n2w1tchr (in=intchr) sasdb.n2w2dirassess ; BY ID; if intchr ; Run;</pre> <pre>/* keeps if in teacher file but not in assessment file */ Data Myfile ; merge sasdb.n2w1tchr sasdb.n2w2dirassess (in=indir); BY ID; if not indir; Run;</pre>

Operation	Syntax/Program Code
<p>Add selected variables to the active file and keep only those variables needed</p>	<p>Syntax:</p> <pre>DATA [NEWfile] ; MERGE [ddname.file] (in=[infile1] keep=[variables]) [ddname.file] (in=[infile2] keep=[variables]) ; BY [key variable]; RUN;</pre> <p>Note: Always keep "ID" (so that you can add more variables to the file later) and the appropriate analysis weight(s)</p> <p>Example:</p> <pre>Data Myfile ; merge sasdb.n2w4paryouth (in=inpar keep=ID np4Wt np4E2a np4E2c) sasb.n2sample (keep=id Cluster Strata) ; BY ID; if inpar ; Run;</pre>
<p>Create new file from existing files</p>	<p>Syntax:</p> <pre>DATA [ddname.NEWfile] ; MERGE [ddname.file] (in=[infile1] keep=[variables]) [ddname.file] (in=[infile2] keep=[variables]) [ddname.file] (in=[infile3] keep=[variables]) ; BY [key variable]; RUN;</pre> <p>Note: Always keep "ID" so that more data may be added to the new file and the appropriate analysis weight(s)</p> <p>Example:</p> <pre>Data mysasdb.Myfile ; merge sasdb.n2w4paryouth (in=inpar keep=ID np4Wt np4E2a np4E2c) sasdb.n2w2dirassess (keep=ID ndasci_pr ndass_pr) sasb.n2sample (keep=id Cluster Strata) ; BY ID; if inpar ; Run;</pre>
<p>Save the new file</p>	<p>Syntax:</p> <p>use a 2-level SAS name rather than a work file name, i.e. [ddname].[filename] rather than [filename]</p> <p>Note: if no ddname is designated, the program compiles as "work.[filename]".</p> <p>Example:</p> <pre>data mysasdb.Myfile ; /* saves to library */ set myfile ; /* from this work file */ run ;</pre>

Operation	Syntax/Program Code
Create a new variable	<p>Syntax:</p> <pre>if [condition statement] then [(new) variable] = [value]; LABEL [variable name] = '[variable label]'; Format [variable name] [format name]. ;</pre> <p>Note: examples of [value] are [a numeric value] [an existing variable value] [the result of a function or operation].</p> <p>The above is assigned conditionally, no assignment is made if the logical condition is not true. An assignment statement would not have a condition and would be simply</p> <pre>[(new) variable] = [value];</pre> <p>Example:</p> <pre>/* conditional assignments */ if np1F4 = 0 or np2P6_J2_I4 = 0 or np3P6_J2_I4 = 0 or np4P6_J2 = 0 then GrpActiv_ever = 0 ; else if np1F4 = 1 or np2P6_J2_I4 = 1 or np3P6_J2_I4 = 1 or np4P6_J2 = 1 then GrpActiv_ever = 1 ; /* assignment statement without condition */ /* followed by a conditional assignment */ GrpActiv_ever = np4P6_J2 ; if np1F4=1 or np2P6_J2_I4=1 or np3P6_J2_I4=1 then GrpActiv_ever = 1 ; label GrpActiv_ever = '(GrpActiv_ever) Youth ever participated in community group or activity'; format GrpActiv_ever ynfmt. ;</pre>
Recode existing variable	<p>Syntax:</p> <pre>if [existing variable] = [value] then [recoded variable] = [recoded value] ; else if [existing variable] = [value] then [recoded variable] = [recoded value] ;</pre> <p>LABEL [variable name] = '[variable label]';</p> <p>Format [variable name] [format name]. ;</p> <p>Example:</p> <pre>if missing(npr1B4) then npr1B4_cat = npr1B4 ; else if npr1B4 <=1 then npr1B4_cat = 1; else if 2<=npr1B4<=5 then npr1B4_cat = 2; else if npr1B4 >5 then npr1B4 = 3 ; label npr1B4_Cat = '(npr1B4_Cat) Days absent per month categorized'; format npr1B4_Cat catfmt.</pre>

Operation	Syntax/Program Code
Adding value labels to new/recoded variables	<p>Syntax:</p> <pre>PROC FORMAT ; VALUE [format name] [value] = "[value label]" [value] = "[value label]" [value] = "[value label]"; [in data step for Proc, add] FORMAT [variable name] [format name]. ;</pre> <p>Example:</p> <pre>proc format ; value Catfm 1 ="(1) None or 1 day absent" 2 ="(2) 2 to 5 days absent" 3 ="(3) More than 5 days absent" ; proc freq data = mydata ; tables npr1B4_Cat ; format npr1B4_Cat Catfm. ;</pre>
Frequencies in Proc Survey Procedures	<p>Syntax:</p> <pre>PROC SURVEYFREQ DATA = [ddname.file]; TABLES [variable(s)] ; STRATA [strata variable] ; CLUSTER [cluster variable] ; WEIGHT [weight variable] ;</pre> <p>Example:</p> <pre>proc surveyfreq data = sasdb.mydata; tables w2_incm3 ; strata stratum ; cluster cluster ; weight np3wt ;</pre>

Operation	Syntax/Program Code
Crosstabs in Proc Survey Procedures	<p>Syntax:</p> <pre>PROC SURVEYFREQ DATA = [ddname.file]; TABLES [variable(s)] * [by variable] / [options]; STRATA [strata variable] ; CLUSTER [cluster variable] ; WEIGHT [weight variable] ;</pre> <p>OR</p> <pre>PROC SURVEYFREQ DATA = [ddname.file]; TABLES [by variable] * [variable(s)] / [options]; STRATA [strata variable] ; CLUSTER [cluster variable] ; WEIGHT [weight variable] ;</pre> <p>Note on tables options: the nowt, nocellpercent, and nofreq options suppress printing of weighted frequency counts, cell percentages, and cell counts; col and row options print column percentages and row percentages</p> <p>Example:</p> <pre>proc surveyfreq data = sasdb.mydata; tables w2_eth6 * w2_incm3 /nowt nocellpercent; strata stratum ; cluster cluster ; weight np3wt ;</pre>
Means in Proc Survey Procedures	<p>Syntax:</p> <pre>PROC SURVEYMEANS DATA = [ddname.file]; VAR [variable(s)] ; STRATA [strata variable] ; CLUSTER [cluster variable] ; WEIGHT [weight variable] ;</pre> <p>Example:</p> <pre>proc surveymeans data = sasdb.mydata; var npr1B4 ; strata stratum ; cluster cluster ; weight wt npr1 ;</pre>
Comparative means in Proc Survey Procedures	<p>Syntax:</p> <pre>PROC SURVEYMEANS DATA = [ddname.file]; DOMAIN [by variable] ; VAR [variable(s)] ; STRATA [strata variable] ; CLUSTER [cluster variable] ; WEIGHT [weight variable] ;</pre> <p>Example:</p> <pre>proc surveymeans data = sasdb.mydata; domain w1_incm3 ; var npr1B4 ; strata stratum ; cluster cluster ; weight wt npr1 ;</pre>

Operation	Syntax/Program Code
Regression	<p>Syntax:</p> <pre>PROC REG DATA = [ddname.file]; model [dependent variable] = [independent variable(s)];</pre> <p>Example:</p> <pre>proc reg data = sasdb.n2w2dirassess ; model ndaPC_ss = nda_Age W2_Inc_LT25 W2_Inc_LT50 W2_Gen Male;</pre>
Correlation	<p>Syntax:</p> <pre>PROC CORR DATA = [ddname.file]; VAR [variables];</pre> <p>Example:</p> <pre>proc corr data = sasdb.n2w2dirassess ; var ndaPC_ss nda_Age W2_Inc_LT25 W2_Inc_LT50 W2_Gen Male ;</pre>
<p>* Weighted standard errors are estimates; SAS PROC SURVEY procedures may calculate similar but slightly different standard errors than SPSS Complex Samples or other similar procedures for estimating weighted standard errors. See "Weighted Standard Errors" in the "NLTS2 Database Structure" document for more discussion about estimated standard errors.</p>	

SAS Tips

Programming statements end in a ";" [semicolon]

Data steps and procedures are followed by a "Run;" statement.

Select the programming statements in the syntax editor and click the little running man icon to run statements or select "submit" in the "Run" menu. By default, F8 key also submits highlighted statements. If nothing is highlighted, all statements in the program editor will run when submitted.

SAS function keys that can be useful (defaults listed):

F9 lists all active function keys

F4 recalls last submitted program statements (pressing F4 3 times will bring back the last 3 submissions with the most recent one being first)

F6 toggles to the program log window

F7 toggles to the output window

F5 toggles to the program window(s)

F8 submits highlighted syntax statements or, if nothing is highlighted, all syntax statements in active program window

F1 HELP

Key variable to join files is "ID"

Warning: SAS will happily merge without a "By" statement and cases will be joined with random abandon. Check for that "By" statement when merging.

Suggestion: check the program log for warnings, errors, and number of observations going in and coming out.

SAS help is searchable. Be sure to check documentation for procedure options, tables statement options, printing options, and output options.