Best Practices in Logistic Regression: A guide to online resources

Jason W. Osborne, Ph.D.

Thank you for your interest in my book! It was tremendous fun to write, and I hope you find it useful! Please do not hesitate to contact me at jasonwosborne@gmail.com if you have comments on the book or any of my other work, which can also be found on my web site http://jwosborne.com. *If you are an instructor and would like access to some powerpoint collections and/or mid-term assignment please do not hesitate to contact me.*[1]

Data sets I used for this book:

1. NELS88:
    a. Source and documentation: http://nces.ed.gov/surveys/nels88/
    b. Reduced data set used in much of the book (SPSS format): https://dl.dropboxusercontent.com/u/18489687/logistic/NELS_data_pulll_2.SAV
2. NHIS 2010:
    a. Source and documentation: http://www.cdc.gov/nchs/nhis.htm
    b. Reduced data set used in much of the book (SPSS format): https://dl.dropboxusercontent.com/u/18489687/logistic/NHIS2010-master.sav
3. ELS2002:
    a. Source and documentation: http://nces.ed.gov/surveys/els2002
    b. Reduced data set used in some of the book (SAS format): https://dl.dropboxusercontent.com/u/18489687/logistic/els2002.sas7bdat
4. HS&B:
    a. Source and documentation: http://nces.ed.gov/pubsearch/getpubcats.asp?sid=022
5. AAUP data
    a. Original source and documentation: http://lib.stat.cmu.edu/datasets/
    b. Description of data source and set: https://dl.dropboxusercontent.com/u/18489687/logistic/aaup%20document.txt
    c. SPSS version of the data set: https://dl.dropboxusercontent.com/u/18489687/logistic/AAUP.sav

ERRATA

1. P. 28, Table 2.3: bottom row, easiest calculation (of course). 100 in base 10 should be 2, not 1, and 0.01 should be -2, not -1. I believe all other calculations in the table are accurate… (thanks to Jennifer Sloan for this one!)
2. p. 29, top, the book says: "*Thus, the interesting property of logs is that they "pivot" at 1.0, are essentially symmetrical around 1.0, and the log of 100 and 1/100 being identical except for the sign.*". This is a bit obtuse and obviously I got excited about some esoteric mathematical point. What I meant is that the log of X (e.g, 100), and the log of 1/X (e.g., 1/100 or 0.01) are symmetrical in that they are identical except for a negative sign. Sorry for any confusion!
3. p. 35-36: Relative risk calculation at the top of p. 35 should be 0.145/0.029 = 5.00 rather than being rounded to 0.145/0.03 to be consistent with the results on page 36 (Thanks to Michael Kelly for catching this!)

[1] No guarantee they are any good, mind you…

4. p. 161, top: the comparison between 88% (0.8814) and 97% (0.9726) should be a 9% difference not 11%. Stupid brain, I'll get you for that! [2]
5. Page 315- 316: null hypothesis should be "H_o : log (p/1-p) = 0 *or* OR = 1.0. On the top of page 316, those should be alternative hypotheses (H_a) and should be log (p/1-p) ≠ 0 *or* OR ≠ 1.0.
6. Table 8.17 title should read "greater than -5"
7. Thanks to Dr. Marty Levin from the University of Memphis for catching this bonehead issue: "**On page 81 in footnote 2 you indicate that you "advise your students to add a 1 at the end of that (0,000) number" when SPSS reports the significance level as 0.000. Then in the text, you report that the significance level is p< .0001. I had always advised my students to replace the final 0 by 1, so that 0.000 is reported as p< 0.001. I can verify that if, for example, the actual probability is .000212 SPSS will report it as .000. Thus, it is not less than .0001 but it is less than .001."** Thus, I recommend substituting $p < .001$ when the number is printed in SPSS as .000.

[2] Any Homer Simpson fans out there?

Chapter resources:

Chapter 1: None available currently

Chapter 2:

- Link to Osborne (2011) publication on weighting and complex samples: http://pareonline.net/pdf/v16n12.pdf
- Exercise #2 data: https://dl.dropboxusercontent.com/u/18489687/logistic/CH02-exdata_sm.sav
- Syntax for performing analysis in SPSS:

```
LOGISTIC REGRESSION VARIABLES RICH
/METHOD=ENTER BACHPLAN
/SAVE=ZRESID
/PRINT=CI(95)
/CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).
```

Syntax for performing analysis in SAS:

```
PROC LOGISTIC DATA=book.CH2_ex descending;
MODEL bachplan = rich;
output out=results p=predict;
run;
```

Chapter 3:

- Data set used in Chapter 3 (SPSS): https://dl.dropboxusercontent.com/u/18489687/logistic/Ch03data.sav
- Data set MJ (SPSS): https://dl.dropboxusercontent.com/u/18489687/logistic/mj.sav
- Syntax for performing analyses:

SPSS syntax for producing Hosmer and Lemeshow analyses with continuous variables:

```
LOGISTIC REGRESSION VARIABLES dropout
/METHOD=ENTER Zbyses
/PRINT=GOODFIT CI(95)
/CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).
```

In this case, the /PRINT=GOODFIT command produces that particular output.

SAS syntax for producing logistic regression analyses with Hosmer and Lemeshow analyses:

```
PROC LOGISTIC DATA=book.Ch03_NELS descending;
MODEL dropout = zbyses
/lackfit ctable;
output out=results p=predict ;
run;
```

in this case, the LACKFIT command produces these results. You can customize how many categories are produced but this produces the default ~10 categories.

SAS command for producing the histogram:

```
proc univariate data=BOOK.Ch03_NELS;
```

```
var zbyses;
histogram;
run;
```

Chapter 4:

- Link to ECLS-K document: http://nces.ed.gov/pubs2002/2002135_2.pdf
- Data set for reproducing examples in chapter (SPSS): https://dl.dropboxusercontent.com/u/18489687/logistic/NELS_CH04.SAV
- Data set for exercise #2: https://dl.dropboxusercontent.com/u/18489687/logistic/NHIS2010_CH04.sav
- Data set for exercise #3: https://dl.dropboxusercontent.com/u/18489687/logistic/mj_CH04.sav
- Some example syntax:

  Example SPSS SYNTAX for requesting standardized residuals, Cook's Distance, leverage statistics, etc.

```
LOGISTIC REGRESSION VARIABLES diabetes
/METHOD=ENTER BMI
/SAVE=PRED COOK LEVER DFBETA ZRESID DEV
/PRINT=GOODFIT ITER(1) CI(95)
/CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).
```

  Example SPSS syntax for selecting or filtering cases:

```
temp.
select if (zre_1 gt -4).
LOGISTIC REGRESSION VARIABLES retain
/METHOD=ENTER Zbyses
/SAVE=PRED COOK LEVER DFBETA SRESID ZRESID DEV
/PRINT=GOODFIT CI(95)
/CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).
```

  This is the "classic" way of temporarily selecting certain cases for analysis—without the TEMP command select removes cases permanently from the data set. With the TEMP command, it only removes it for the first analysis. The more recent method is FILTER:

```
USE ALL.
COMPUTE filter_$=(ZBMI < 3 and ZBMI > -3).
FILTER BY filter_$.
EXECUTE.
```

  The FILTER command keeps the cases in the data set but removes them from any analyses until the USE ALL command returns them to analyses. It is a nice, safe compromise to select, but you have to remember to return cases to the data if you want to analyze the entire sample again.

  Example SAS Syntax for requesting extensive plots and leverage statistics

```
PROC LOGISTIC DATA=book.Ch03 descending plots(MAXPOINTS=NONE)=(ROC
dfbetas(unpack) influence(unpack) leverage(unpack) );
MODEL diabetes = BMI
/lackfit ctable ;
output out=book.results p=predict CBAR=cbar DIFchisq=DIFCHI
RESCHI=reschi dfbetas=_ALL_ H=leverage ;
run;
```

Example SAS syntax for deleting particular cases and saving in a new data file called "remove".

```
data remove;
set BOOK.results;
if DIFCHI > 10 then delete;
;
run;
```

Example SAS syntax for examining univariate distributions of variables, such as DIFCHI.

```
proc univariate data=remove;
var DIFCHI;
histogram;
run;
```

Using Pearson Chi Square Deletion Difference in SAS

```
PROC LOGISTIC DATA=book.Ch03 descending plots(MAXPOINTS=NONE)=(ROC
dfbetas(unpack) influence(unpack) leverage(unpack) );
  MODEL diabetes = BMI
  /lackfit ctable ;
  output out=book.results p=predict CBAR=cbar DIFchisq=DIFCHI
RESCHI=reschi dfbetas=_ALL_ H=leverage ;
run;
```

Chapter 5:

- AAUP data: https://dl.dropboxusercontent.com/u/18489687/logistic/AAUP.sav
- ELS 2002 data from this chapter: https://dl.dropboxusercontent.com/u/18489687/logistic/CH05ELS2002.sav
- SPSS syntax that was used to create the dichotomized and extreme groups variables here: https://dl.dropboxusercontent.com/u/18489687/logistic/SyntaxCH05.sps
- Diabetes and BMI data for exercise #3: https://dl.dropboxusercontent.com/u/18489687/logistic/NHIS2010_CH04.sav

Chapter 6:

- Chapter 6 data (SPSS): https://dl.dropboxusercontent.com/u/18489687/logistic/CH06.sav
- Chapter 6 smoking data (SPSS): https://dl.dropboxusercontent.com/u/18489687/logistic/CH06_smoke.sav

Some SPSS Syntax help for recoding variables in the chapter into dummy/effects coded variables:

```
recode  RACERPI2 (1=1) (2=2) (4=3) into race_new.
** race recode:  1= white, 2=black, 3=asian.
recode  RACERPI2 (1=3) (2=2) (4=1) into race_new1.
** race recode:  1= white, 2=black, 3=asian.
recode smkstat2 (4=0) (3=1) (2=2) (1=3) into smoke_cat.
*** smoking recode 0= nonsmoker 1=former, 2=occasional, 3=every day.
execute.
```

```
select if (smoke_cat ge 0).
execute.

***compute dummy coded effects variables***.
compute dum1=0.
compute dum2=0.
compute dum3=0.
if (smoke_cat=1) dum1=1.
if (smoke_cat=2) dum2=1.
if (smoke_cat=3) dum3=1.
execute.

***compute effects coded race variables***.
compute eff1=-1.
compute eff2=-1.
if (race_new=2) eff1=1.
if (race_new=3) eff1=0.
if (race_new=2) eff2=0.
if (race_new=3) eff2=1.
execute.
```

SPSS syntax for recoding the original race variable in the NHIS data set and for receiving contrasts in output (in this example it is asking for REPEATED contrasts):

```
recode RACERPI2 (1=1) (2=2) (4=3) into race_new.
** race recode: 1= white, 2=black, 3=asian.
LOGISTIC REGRESSION VARIABLES diabetes
 /METHOD=ENTER race_new
 /CONTRAST (race_new)=Repeated
 /print=all
 /CRITERIA=PIN(.05) POUT(.10) ITERATE(20) CUT(.5).
```

Another example gives us DEVIATION coding with Caucasian (1) as the reference group:

```
LOGISTIC REGRESSION VARIABLES diabetes
 /METHOD=ENTER race_new
 /CONTRAST (race_new)=Deviation(1)
 /CRITERIA=PIN(.05) POUT(.10) ITERATE(20) CUT(.5).
```

SAS syntax for effects coding

```
proc logistic data=Book.Ch06 descending;
class race_new (ref=first);
model diabetes=race_new /expb;
run;

proc logistic data=Book.Ch06 descending;
class smoke_cat (ref=first);
model diabetes=smoke_cat /expb;
run;
```

To get dummy coding in SAS use the "/param=ref" statement:

```
proc logistic data=Book.Ch06 descending;
class smoke_cat (ref=first) /param=ref;
model diabetes=smoke_cat /expb;
run;
```

to get dummy coding in SAS that uses a different reference group, make sure your categorical variable is a STRING variable, and then simply specify the value. For example, Former smoker is "1" in our data:

```
proc logistic data=Book.Ch06 descending;
class smoke_cat (ref='1') /param=ref;
model diabetes=smoke_cat /expb;
run;
```

Chapter 7:

- NHIS data set used in Chapter 7: https://dl.dropboxusercontent.com/u/18489687/logistic/Ch07_NHIS2010.sav
- NELS88 (MJ) data set used in Chapter 7: https://dl.dropboxusercontent.com/u/18489687/logistic/evermj_curvilinear_sm.sav
- ELS2002 data set used in Chapter 7: https://dl.dropboxusercontent.com/u/18489687/logistic/CH07_ELS2002_sm.sav

Example SPSS Syntax to create squared and cubed terms for exploring curvilinearity:

```
compute zBYACH2=ZBYACH**2.
compute zBYACH3=ZBYACH**3.
execute.
```

Example SPSS Syntax to perform logistic regression entering squared and cubed terms on separate steps:

```
LOGISTIC REGRESSION VARIABLES EVER_MJ
/METHOD=ENTER ZBYACH
/METHOD=ENTER zBYACH2
/METHOD=ENTER zBYACH3
/SAVE=DFBETA ZRESID
/PRINT=CI(95)
/CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).
```

Example SAS syntax to enter variables one at a time on separate steps:

```
PROC LOGISTIC DATA=book.ELS2002 descending ;
MODEL RETAINED = ZBYSES;
;
run;
PROC LOGISTIC DATA=book.ELS2002 descending ;
MODEL RETAINED = ZBYSES ZBYSES2
;
run;
PROC LOGISTIC DATA=book.ELS2002 descending ;
```

```
MODEL RETAINED = ZBYSES ZBYSES2 ZBYSES3
/selection=none sequential;
run;
```

The above syntax might not be the most elegant but it allows for direct comparison of separate models by comparing both change in -2LL and regression equation at each step. There are options if you use stepwise entry methods to accomplish this with one command (such as the "sequential" command) but stepwise entry methods have the potential to produce problematic outcomes under certain circumstances (like a regression model that includes zBYSES and zBYSES3 but not zBYSES2). Thus I prefer the above method that provides absolute control to the analyst.

NOTE: when attempting to reproduce the results of the enrichment exercises involving AGE (e.g., enrichment figure 7.2) double-click on the SPSS output and take the regression coefficients out to several decimal places further than what it gives you by default. It will help you reproduce what I produced.

Chapter 8:

- Data set used in Chapter (NELS): https://dl.dropboxusercontent.com/u/18489687/logistic/NELS_data_pulll_2.SAV
- Data set for race x SES interaction: https://dl.dropboxusercontent.com/u/18489687/logistic/CH08_race_ses.sav
- Exercise #2 data set: https://dl.dropboxusercontent.com/u/18489687/logistic/NHIS2010-master.sav

Chapter 9:

- NHIS data set used for examples in the chapter: https://dl.dropboxusercontent.com/u/18489687/logistic/NHIS2010-master.sav

Technical note for Chapter 9:

For those of you using SPSS, there is a simple way to perform Probit regression via syntax which is not easily available via point-and –click. This is the syntax I used in the first example for this chapter before centering the intercept at locations other than 0.

```
plum diabetes  with age BMI ageBMI
  /link=probit
  /print= PARAMETER SUMMARY.
```

Chapter 10:

- NELS data set used in the chapter: https://dl.dropboxusercontent.com/u/18489687/logistic/NELS_data_pulll_2.SAV
- Resources for power calculations:
    - The most popular website, http://www.dartmouth.edu/~eugened/power-samplesize.php, uses an algorithm created by Demidenko in 2007, which computes power, sample size, and minimal detectable odds ratios using the Wald test.
    - Researchers at UCLA, http://www.ats.ucla.edu/stat/stata/dae/logit_power.htm, detailed a simple method of computing power and sample size using Stata, but these estimates should be considered lower bounds.

- Faul, Erdfelder, Buchner, and Lang (2009) also provide examples and a detailed breakdown of power analysis in logistic regression using the freely-available G*Power software. Users of G*Power have two options for power analysis: the first uses an enumeration procedure which runs simulations to approximate sample size; and the second is based on Demidenko's and Whittmore power formulas that use large-sample sizes to estimate power via the Wald statistic. G*Power 3.1 is available at http://www.psycho.uni-duesseldorf.de/abteilungen/aap/gpower3/.
- SAS software has PROC POWER that is extremely versatile in specifying not only traditional aspects of power (alpha, N, effect size) but also the distributions of predictors, covariates, and the odds ratios of covariates. An overview of this procedure is available online from SAS at: http://support.sas.com/documentation/cdl/en/statug/63347/HTML/default/viewer.htm#statug_power_a0000001016.htm

Syntax for logistic regression bootstrapping in SPSS

This is a macro and syntax modified from the IBM macro for bootstrapping OLS regression found at: http://publib.boulder.ibm.com/infocenter/spsstat/v20r0m0/index.jsp?topic=%2Fcom.ibm.spss.statistics.help%2Foms_sav_examples_bootstrapping3.htm

You can copy and paste this into a SPSS syntax window and alter the last lines to customize the:

- Data set—you must enter the address for the data set here (e.g., c:\data.sav)
- You can change the number of samples you want to analyze (currently set to 1000)
- Change DEPVAR to the name of the dependent variable you want to analyze
- Change INDVARS to the name (or names of the independent variables) you want as predictors in the model. In this case, if you are examining two variables and the interaction, you could put: VAR1 VAR2 INT (but only if those variables and the interaction were already in the data set!)

```
DEFINE logistic_bootstrap (samples=!TOKENS(1)
                           /depvar=!TOKENS(1)
                           /indvars=!CMDEND)
COMPUTE dummyvar=1.
AGGREGATE
  /OUTFILE=* MODE=ADDVARIABLES
  /BREAK=dummyvar
  /filesize=N.
!DO !other=1 !TO !samples
SET SEED RANDOM.
WEIGHT OFF.
FILTER OFF.
DO IF $casenum=1.
- COMPUTE #samplesize=filesize.
- COMPUTE #filesize=filesize.
END IF.
DO IF (#samplesize>0 and #filesize>0).
- COMPUTE sampleWeight=rv.binom(#samplesize, 1/#filesize).
- COMPUTE #samplesize=#samplesize-sampleWeight.
- COMPUTE #filesize=#filesize-1.
ELSE.
```

```
- COMPUTE sampleWeight=0.
END IF.
WEIGHT BY sampleWeight.
FILTER BY sampleWeight.
LOGISTIC REGRESSION VARIABLES !depvar with !indvars.
!DOEND
!ENDDEFINE.

GET FILE='<enter file location here>'.
logistic_bootstrap
   samples=1000
   depvar= graduate
   indvars=zbyses.
```

Chapter 11

- NHIS data used for analyzing missingness in BMI: https://dl.dropboxusercontent.com/u/18489687/logistic/NHIS2010-sm-CH11.sav
- MNAR analysis: https://dl.dropboxusercontent.com/u/18489687/logistic/Ch11_MNAR.sav

Chapter 12

- NELS88 data from chapter: https://dl.dropboxusercontent.com/u/18489687/logistic/Ch12_MJdata.sav
- NHIS data for enrichment: https://dl.dropboxusercontent.com/u/18489687/logistic/CH12_NHIS_sm.sav

SPSS Syntax from the chapter for filtering out influential cases from the results of binary logistic regression

```
DESCRIPTIVES VARIABLES=DFB0_1 DFB0_2 DFB0_3 DFB1_1 DFB1_2 DFB1_3
DFB2_1 DFB2_2 DFB2_3 DFB3_1 DFB3_2 DFB3_3
  /SAVE
  /STATISTICS=MEAN STDDEV MIN MAX.

recode  ZDFB0_1 ZDFB0_2 ZDFB0_3 ZDFB1_1 ZDFB1_2 ZDFB1_3 ZDFB2_1
ZDFB2_2 ZDFB2_3 ZDFB3_1 ZDFB3_2 ZDFB3_3
  (sysmis=0).
execute.


USE ALL.
COMPUTE filter_$=((ZDFB0_1  < 6  and ZDFB0_2 < 6 and ZDFB0_3 < 6 ) and
   (ZDFB1_1  < 6  and ZDFB1_2 < 6 and ZDFB1_3 < 6 and ZDFB1_1  >-6
and ZDFB1_2 >-6 and ZDFB1_3 >-6) and
```

```
   ( ZDFB2_1  < 6  and ZDFB2_2 < 6 and ZDFB2_3 < 6 and ZDFB2_1  >-6
and ZDFB2_2 >-6 and ZDFB2_3 >-6) and
   ( ZDFB3_1  < 6  and ZDFB3_2 < 6 and ZDFB3_3 < 6)).
FILTER BY filter_$.
```

SAS Syntax for ordinal logistic regression[3]

```
proc logistic data = BOOK.ch12b descending;
model MJsas = zACH zSES ;
run;
```

Chapter 13:

- Link to HLM resources: http://www.ssicentral.com/
- Link to HLM data set from chapter (in .MDM format): https://dl.dropboxusercontent.com/u/18489687/logistic/HLM_logistic_example_nol1l2resid

Enrichment example #2: Your job is to help us understand student dropout in high school.
Download the following data files:
Level 1: https://dl.dropboxusercontent.com/u/18489687/logistic/Ch13_l1_N200.sav
Level 2: https://dl.dropboxusercontent.com/u/18489687/logistic/CH13_l2_N200.sav

Variables of interest:

L1:

- Sch_ID: school id
- Dropout: 0=not dropped out, 1=dropped out
- zBY2XCOMP: standardized 8th grade achievement test score
- zBYSES: standardized 8th grade family SES
- RACEBW: race- 0=white, 1=African American
- RACEHW: race: 0=white, 1= Latino/latina

L2:

- zg8enrol: standardized school size
- zg8lunch: standardized % students on free/reduced lunch
- zg8minor: standardized % racial minority students in school

After creating the HLM2 data file (you should tell HLM there is missing data, and tell it to delete the missing data when RUNNING the analysis).

[3] For some reason SAS did not like the original MJ variable coded 0-3 and mis-ordered the variable when imported from SPSS data. Thus I created MJSAS, recoding 0-3 to be 1-4.

You should see something similar to this when creating MDM:

LEVEL-1 DESCRIPTIVE STATISTICS

VARIABLE NAME	N	MEAN	SD	MINIMUM	MAXIMUM
DROPOUT	3511	0.10	0.30	0.00	1.00
ZBY2XCOM	3412	0.00	1.00	-1.95	2.50
ZBYSES	3511	-0.00	1.00	-3.80	2.53
RACEBW	3511	0.12	0.32	0.00	1.00
RACEHW	3511	0.15	0.35	0.00	1.00

LEVEL-2 DESCRIPTIVE STATISTICS

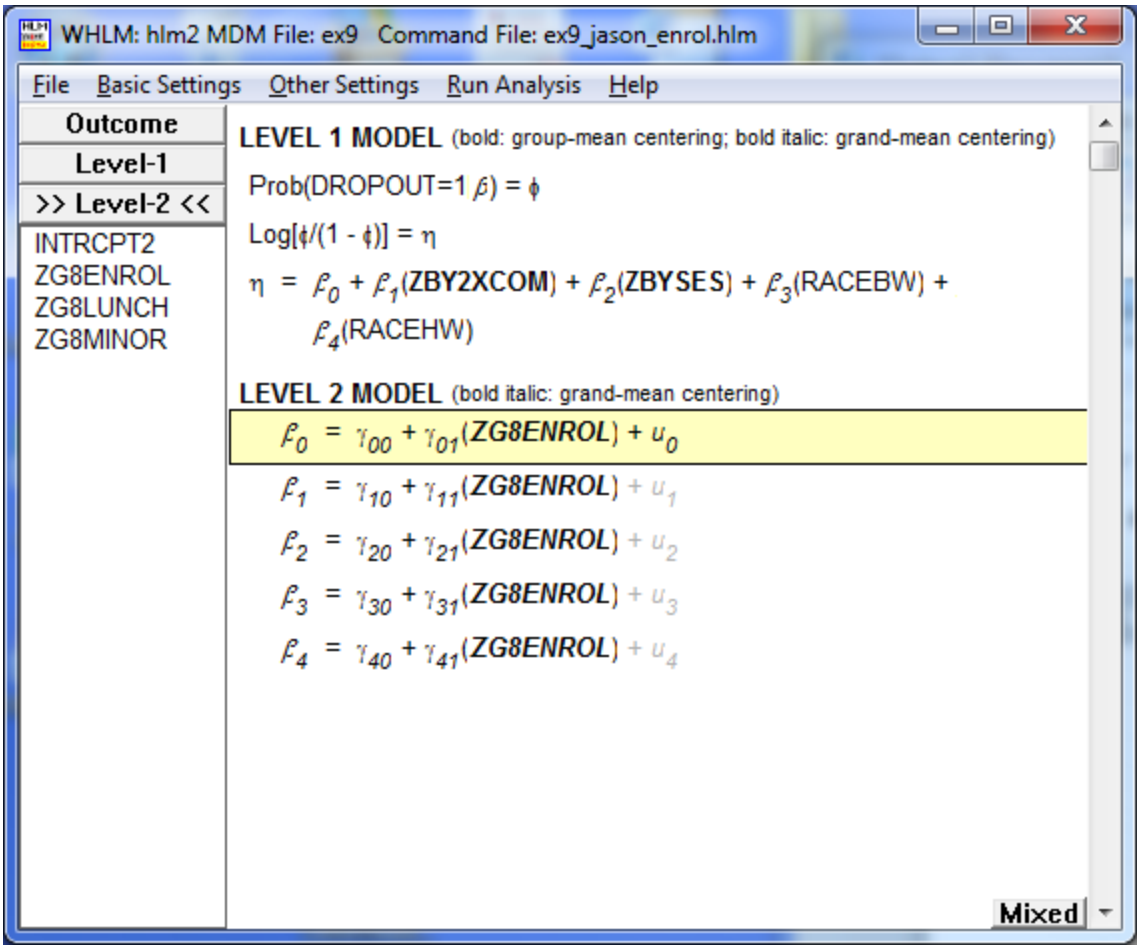
VARIABLE NAME	N	MEAN	SD	MINIMUM	MAXIMUM
ZG8ENROL	200	-0.00	1.00	-2.19	1.63
ZG8LUNCH	200	0.00	1.00	-1.42	2.09
ZG8MINOR	200	0.00	1.00	-1.15	2.13

Under Basic Settings, be sure to tell the computer you have a 0,1 DV as the diagram on the left shows.

Also be sure to set up the two residual files with all relevant variables so you can use them for second round of analyses if need be.

Add ZBY2XCOM and ZBYSES group centered, and two race variables uncentered, as routine.

At level 2, add zg8enrol to all equations, grand centered. This is what your screen should look like before you run the analysis (make sure to save it in a place you can find so you can access the residual files prior to running!)



Reporting:

Focus on the POPULATION with ROBUST STANDARD ERRORS. Make a table with the unstandardized coefficients, significance test information, and odds ratios at a minimum. If you have any cross-level interactions (you should have ONE), graph it out using unstandardized coefficients.

HINTS:

- there are some extreme L1 residuals. Remove them. Once gone, re-run the model using the newly cleaned data. You should see the cross-level interaction become more significant. You will also see that only 2 schools have MDIST >4, and not by much. As my gift to you, you may leave them as is and interpret the second (L1 cleaned) run for your write-up.
- Be sure to interpret the nonsignificant Race effects—they are somewhat unexpected and IMPORTANT conceptually. Note that for these effects, the 95% CIs include OR of 1.00—which means they are not significant.
- Graph the cross-level interaction using the unstandardized coefficients, and be sure to label the Y axis “log likelihood of dropping out”
- When graphing the interaction, you can drop the race effects since they are not significant. You are welcome.