MEDB 5507 Introduction to SAS
Fall 2016
Course Outline
Notes and Assignments

NOTE –
This document provides important information about each part of the Introduction to SAS course material. Be sure to have this document available and on-hand as you review the lectures and work on your own coding. It will provide you with reminders of coding examples that you can use for your own work.

Intro

Intro – SAS

- Command language
    - We will build command language files that can be used, adapted, and re-used
- Interactive – write commands / run them / see results – Log & output
- Note – "menu driven" version of SAS – Enterprise Guide – NOT covered by this course

- Major types of actions –
    - Data steps – data management / manipulation
    - Proc steps – running procedures to get information out of data

- Hints – based on my preferences / experiences; you will develop your own preferences
    - Ex – variable naming –
        - Requirements -
            - 32 characters or less in length
            - Start with letter or _ (underscore)
            - Contain only letters, numbers, or _ (underscore)
            - No imbedded blanks
            - SAS is not case sensitive
        - Preferences - length of names – clarity vs brevity (typing!!)
    - Always checking the "colors" in your coding – warns you of syntax errors

- Set-up options / Preferences
    - Output vs Results
    - Screen layout

- Menu/Task bar

File	Open program / save as … / import data /
Edit	
View	
Tools	Options / Preferences
Run	
Solutions	
Window	also get to these through the tabs
Help	SAS Help & Documentation

ADDITIONAL NOTES – NOT IN THE RECORDING

- This document will include coding lines from the SAS program code being used in the recorded lectures. This coding is included in this document to serve as annotation for what is being shown.
- The coding files that are provided in the Blackboard course site can be used as the basis for your own coding. Feel free to copy appropriate lines in the program files and edit/revise as appropriate.
- You can have more than one program editor window open at the same time. You will be able to move from one to the other by selecting the appropriate tab at the bottom of the SAS window. This will make it easier to have the example coding file available for reference at the same time you are working in your own program code window.

"On Your Own" assignments –

To show a record of your work

Copy and paste your log file into a text editor

Copy and paste your output into a text editor OR

Save your Results Viewer output as a .mht - include "part 1" and our name/initials in the filename

PART 1 - Initial Steps – Getting Data into SAS and saving SAS dataset (sas_train_part1.sas)

Datasets used – all in the Blackboard folder

ImportDataFile_description.docx
ImportDataFile.xlsx
ImportDataFile.csv
ImportDataFile_tab_txt
ImportDataFile_space.txt
ImportDataFile_fixed_data.txt

Pulmonary dataset – (fixed column)
Pulmonary dataset_description.docx (description) – (see Blackboard folder)
http://www.amstat.org/publications/jse/datasets/fev.dat.txt (data)

Note – Documentation / annotation in a SAS program
Notice colors of the code in the Program editor

File Management

Setting up SAS library

What is a "SAS library"?
How to see libraries that exist / properties

Create using the "library" tool
Creating using command language

Coding –

```
* Command language to set up a SAS library ;
libname "libname" 'path to computer location where files are to be stored' ;
run ;
* example - SAS library on a desktop computer device ;
libname testlib2 'c:\My Documents\MEDB 5507_Intro to SAS\SAS files' ;
run ;
```

Importing data file - .csv / .txt / .sav / .xlsx / .sav

All import data files in the Blackboard folder

(note 64-based systems problem importing .xlsx files)

Through "File / Import Data" tool

Note information in the Log file

Through command language

Coding –

```
* READ-IN fixed-column data file - desktop version - THIS WORKED ;
data pulm ;
       infile 'c:\my documents\MEDB 5507_Intro to SAS\Datasets\SAS
course\pulmonary_data.txt' ;
       input age 1-3 fev 5-10 ht 12-15 sex 19 smoke 25 ;
       run ;
```

Variable name conventions reminder

Work space vs SAS library

Checking data

Categorical variables - Proc Freq
Continuous variables - Proc Means
Continuous variables - Proc Univariate

Coding –

```
* Run FREQUENCY to check data that were imported - does it look right? ;
* Categorical variables ;
proc freq data=csv1 ;
 tables title level number_courses ext_res_fund number_services ;
 run ;

* Continuous variables ;
proc means data=csv1 n mean stddev min max ;
 var years number_courses number_services ;
 run ;

* Alternative way to check distribution of variables - continuous measures ;
proc univariate data=csv1 normal ;
 var years number_courses number_services ;
 run ;
```

Saving SAS dataset - save the "import" dataset - we'll come back to it later
General format for saving a SAS dataset to a SAS library -
data libname.filenameA ; set work.filenameB ;
run ;

libname.filenameA
libname library name pointing to the location where you want to save data; you should have already set up this SAS library
filenameA filename you want to use for the data file you’re saving

work.filenameB
work refers to the working library; this “space” exists during a working session
filenameB filename for the file in the working space you want to save

In the Log window you should see a record of what was saved to the library in terms of # of observations (rows) and # of variables in the file.

Coding –

```
* Save SAS dataset ;
data sastrain.csv1 ; set work.csv1 ;
run ;
```

Document your SAS dataset so you have a record of the file contents

Coding –

```
* Document your SAS datasets ;
proc contents position data=sastrain.csv1 ;
run ;
```

On your own -

Save one of the “import” datasets that you imported into SAS as a SAS dataset.

Import **Gardasil Vaccination data set** –
http://www.amstat.org/publications/jse/v19n1/gardasil.txt (description)
http://www.amstat.org/publications/jse/v19n1/gardasil.xls (data)

Check out that the data in the working file looks correct.
Make a note of anything you have a question about in the dataset

Save the data files as a SAS dataset; document your SAS dataset using Proc Contents

Copy and paste your log file into a text editor

Copy and paste your output into a text editor OR
Save your Results Viewer output as a .mht - include "part 1" and our name/initials in the filename

Import / read in other datasets that will either be used for various activities or will demonstrate issues that can come up with datasets. As you import these files, make a note of anything that comes up when you import the dataset that might have an impact on future work with the data.

Back pain dataset with two matched control groups – (tab-delimited)
http://www.statsci.org/data/oz/backpain.html (description)
http://www.statsci.org/data/oz/backpain.txt (data)

Sleep in Mammals dataset – (tab-delimited) Note – Before I was able to import this dataset, I had to delete a few records in the data file. I don’t know what was wrong with the records, but they created an error on import. See what you get when you import this dataset.
http://www.statsci.org/data/general/sleep.html (description)
http://www.statsci.org/data/general/sleep.txt (data)

Diet / Fiber dataset – (tab-delimited)
http://lib.stat.cmu.edu/DASL/Datafiles/Fiber.html (description & data)
Copy the actual data from this URL; paste it into text editor

Housing dataset – (tab-delimited)
http://lib.stat.cmu.edu/DASL/Datafiles/homedat.html (description & data)

South Africa Project Outcomes dataset – (.xlsx file; see Blackboard folder)
SAProj_Outcomes_dataset_description.docx (description)
SAProj_Outcomes_data.xlsx (data)

South Africa project PSS item-level dataset – (SPSS .sav file; see Blackboard folder)

SAProj_PSS items_description.docx (description)
SAProj_PSSitems.sav (data)

MOTIV8 Adherence Project Beliefs About ART dataset - (SPSS .sav file; see Blackboard folder)
MOTIV8_Beliefs_documentation.docx (description)
MOTIV8_Beliefs_train_data.sav (data; see Blackboard folder)

MOTIV8 Adherence Project Depressive Symptoms dataset - (SPSS .sav file; see Blackboard folder)
MOTIV8_DepressiveSymptoms_documentation.docx (description)
MOTIV8_Depression_train_data.sav (data; see Blackboard folder)

NOTE – Datasets that might be of interest to work on for your own project include:

Cholesterol dataset (Converting from tall and thin to short and fat) – (tab-delimited)
http://www.statsci.org/data/general/cholest.html (description)
http://www.statsci.org/data/general/cholestg.txt (data - general format)

Body Dimensions dataset – (space-delimited)
http://www.amstat.org/publications/jse/datasets/body.txt (description)
http://www.amstat.org/publications/jse/datasets/body.dat.txt (data)

Termite dataset – (tab-delimited)
http://lib.stat.cmu.edu/DASL/Datafiles/Termites.html (description & data)
Copy the actual data from this URL; paste it into text editor

Predicting Board Examination Scores dataset – (SPSS .sav file ; see Blackboard folder)
Predicting Board Exam Scores_description.docx (description)
Predicting_BoardExam_scores.sav (data)

Strength dataset – (SPSS .sav file ; see Blackboard folder)
Strength Data_description.docx (description)
Restructure_Strength_vert.sav (data – vertical format)
Restructure_Strength_horiz.sav (data – horizontal format)

Stroke dataset – (tab-delimited)
http://www.statsci.org/data/oz/stroke.html (description)
http://www.statsci.org/data/oz/stroke.txt (data)

Energy consumption dataset (horizontal format) – (tab-delimited)
http://www.statsci.org/data/general/energy.html (description)
http://www.statsci.org/data/general/energy.txt (data)

Balance dataset - Compare the formats – (tab-delimited; horizontal or vertical format) –
http://www.statsci.org/data/oz/ctsib.html (description)
http://www.statsci.org/data/oz/ctsibrm.txt (horizontal format)
http://www.statsci.org/data/oz/ctsibuni.txt (vertical format)

Depression after an Earthquake dataset (tab-delimited) –
http://www.statsci.org/data/general/lomaprie.html (description)
http://www.statsci.org/data/general/lomaprie.txt (data)

Titanic mortality dataset – (tab-delimited)
http://www.statsci.org/data/general/titanic.html (description)
http://www.statsci.org/data/general/titanic.txt (data)

Diamond prices dataset – (fixed column)
http://www.amstat.org/publications/jse/v9n2/4c.txt (description)
http://www.amstat.org/publications/jse/v9n2/4cdata.txt (data - shorter variable list)
http://www.amstat.org/publications/jse/v9n2/4c1data.txt (data - longer variable list)

PART 2 - Modifying your dataset (sas_train_part2.sas)

Adding information to the SAS dataset – format & labels information

Create new variables

Coding for missing data

Renaming variables

Creating a numeric variable from a character variable

Working with dates

Datasets used –

Gardasil Vaccination data set –

http://www.amstat.org/publications/jse/v19n1/gardasil.txt (description)

http://www.amstat.org/publications/jse/v19n1/gardasil.xls (data)

Back pain dataset with two matched control groups – (tab-delimited)

http://www.statsci.org/data/oz/backpain.html (description)

http://www.statsci.org/data/oz/backpain.txt (data)

Sleep in Mammals dataset – (tab-delimited) Note – Before I was able to import this dataset, I had to delete a few records in the data file. I don't know what was wrong with the records, but they created an error on import. See what you get when you import this dataset.

http://www.statsci.org/data/general/sleep.html (description)

http://www.statsci.org/data/general/sleep.txt (data)

Diet / Fiber dataset – (tab-delimited)

http://lib.stat.cmu.edu/DASL/Datafiles/Fiber.html (description & data)

Copy the actual data from this URL; paste it into text editor

ImportDataFile.csv (or .xlsx … - one of the Import files that you have saved as a SAS dataset.

Using the Gardasil dataset

Format information –

What are formats?

Why should you bother?

Coding –

```
* Creating format information using Gardasil dataset ;
* Create SAS format library - gardform - using library tool ;

* Create format information to be stored in library ;
proc format library = gardform ;

* Value statements ;
value agecateg
 0 = '11-17 yrs'
 1 = '18-26 yrs' ;
run ;

* Command to tell SAS where to find format information ;
* Includes code to prevent error message if there is no format information stored ;
* Include this options statement in all programs in the future ;
options fmtsearch = (gardform) nofmterr ;
```

Variable labels –

What do these do?

Why should you bother?

Coding –

```
* Add information to file - variable labels & format codes ;
data gard2 ; set gard1 ;
 label agegroup = 'Age Group - 0 = 11-17 yrs, 1 = 18-26 yrs' ;
 format agegroup agecateg. ;
 run ;
```

Creating new variables –

Reasons for creating new variables –

New race variable

New insurance variable

Coding –

```
* Create new race-ethnicity variable to represent 3 categories ;
data gard3 ; set gard2 ;
 if race = 0 then new_race = 0 ;
 if race = 1 then new_race = 1 ;
 if race in (2 3) then new_race = 2 ;
 label new_race = 'New Race-Ethnicity variable - white / black / other' ;
 run ;

* Create new insurance variable to represent private vs not private ;
data gard4 ; set gard3 ;
 if insurancetype = 1 then private_ins = 1 ;
  else private_ins = 0 ;
 if insurancetype = . then private_ins = . ;
run ;
```

Using the Back Pain dataset

Code missing data

Note – when importing this dataset, what did you notice about the data for "Years" and "WeekDist"?

Values included in the file

How these values were treat when imported to SAS

What kind of variables are they in SAS?

Why did this occur?

Renaming a variable

Coding –

```
data back2 ; set work.back1 ;
 weekdist_rec = weekdist ;
 if weekdist = 0 then weekdist_rec = . ;
 label weekdist_rec = 'weekdist variable with 0 coded as md' ;
 run ;
```

Using the Sleep in Mammals dataset

Code missing data

Coding –

```
* Code character variable NA as missing data ;
```

```
data sleep2 ; set sleep1 ;
 nondreaming_rec = nondreaming ;
 if nondreaming_rec = "NA" then nondreaming_rec = " " ;
 label nondreaming_rec = 'New variable for nondreaming to code NA as missing' ;
 run ;

* Code character variable NA as missing data ;
data sleep3 ; set sleep2 ;
 nondreaming_rec2 = nondreaming ;
 if nondreaming_rec2 = "NA" then nondreaming_rec2 = . ;
 label nondreaming_rec2 = 'Alt - New variable for nondreaming to code NA as missing' ;
 run ;
```

Create a numeric variable from a character variable

Coding –

```
* Create numeric variable from character variable ;
data sleep3 ; set sleep2 ;
 nondreaming_num = nondreaming + 0 ;
 label nondreaming_num = 'Numeric nondreaming variable that includes missing data
coding' ;
 run ;

* Create numeric variable from character variable - alternate method ;
data sleep4 ; set sleep3 ;
 nondreaming_num2 = input(nondreaming,best4.1);
 label nondreaming_num2 = 'Alt - Numeric nondreaming variable that includes missing
data coding' ;
 run ;
```

Using the Diet / Fiber dataset

Create a numeric variable from the character variable

Reason to do this

Create numeric “bloating” variable from character “bloat” variable

Suggested coding structure

bloating	bloat
0	none
1	low
2	med
3	high

Coding –

```
data diet2 ; set diet1 ;
 if bloat = 'none' then bloat_num = 0 ;
 if bloat = 'low' then bloat_num = 1 ;
 if bloat = 'med' then bloat_num = 2 ;
 if bloat = 'high' then bloat_num = 3 ;
run ;
```

Using the original Import dataset

Work with date variables

Coding –

```
* Compute age when hired ;
data import2 ; set import1 ;
 age_hire = int(yrdif(dob, hire_date, 'age')) ;
 age_hire2 = (yrdif(dob, hire_date, 'age')) ;
 run ;

* Compute current age ;
data import3 ; set import2 ;
 current_age = (today() - dob) / 365.25 ;
 current_age_yrs = int(current_age) ;
 run ;

* Create flag variable based on a specific date ;
data import4 ; set import3 ;
 if hire_date > '01jul2010'd then new_hire = 1 ;
  else new_hire = 0 ;
  run ;

* alt - Create flag variable based on a specific date ;
data import4 ; set import3 ;
 if hire_date gt '01jul2010'd then new_hire = 1 ;
  else new_hire = 0 ;
  run ;
```

ADDITIONAL NOTES –

- Before working on formatting on your own, review the “Part 2 Addendum_Intro to SAS … “ recording for additional information regarding format coding. There is a relevant SAS coding file in the Course Content section of the Blackboard course site – “Part 2 addendum – SAS code”
- General information on coding format –

  data work.newfilename ; set work.oldfilename ;
  ….
  run ;

  <u>example statements</u> –

create new variable	newvariable = oldvariable ;
rename variable	rename oldvariable = newvariable ;
if / then statements	if variable = value then newvariable = value ;
if / then / else statements	if variable = value then newvariable = valueA ;
	else newvariable = valueB ;
selecting subset of cases	if variable = value ;
drop selected variables	drop variableA variableB ;
keep selected variables	keep variableC variableD ;
retain selected records	if …. then retain ;
delete selected records	if …. then delete ;
assigning missing data (numeric)	if variable = mdvalue then variable = . ;
assigning missing data (character)	if variable = ‘mdvalue’ then variable = ‘ ‘ ;

- As I recently worked with a student, I was reminded of the upside & downside of the use of format. So, the student sent me a SAS dataset she created that included the use of a format for the age group variable - just as instructed. When I tried to open that dataset, I got an error message because I didn't have the format info in my "SAS system." To prevent these kinds of problems/errors, I routinely include that statement I mentioned in Part 2 - *options fmtsearch - (format library name) nofmterr ;* (Naturally, you have to enter a real library name where I've typed "format library name.") So, this means that I had to create a fake format library - create a library name that points to a location, and use that library name in the options statement. When I do that, I can open the SAS dataset without a problem. I don't have the format info that the student included, but I can get to the data. So, bottom-line - including format info in your saved SAS dataset can be great for your own use of the data. Just keep in mind that if someone else is also going to be using the dataset and they want to use format info, they need to have the format info to include in their program, or to save in the format library that they set up. This complication goes a long way to explaining my love/hate relationship with formats!! :)

On your own -

Gardasil dataset

Add variable labels to all variables

Add format information as appropriate

Create new Insurance variable to represent "private insurance" or not (1 vs 0) if you haven't already done this

Save new Gardasil SAS dataset

Home Prices dataset –

Code for missing data

Change variables that had missing data coded to numeric variables

Document your work using your log record and any relevant output.

PART 3 – Subsetting your dataset and combining datasets (sas_train_part3.sas)

Selecting subset of variables
Selecting subset of cases
Combining (concatenating) datasets together – adding observations
Merging datasets together – adding variables

Datasets used –

Gardasil Vaccination data set –
http://www.amstat.org/publications/jse/v19n1/gardasil.txt (description)
http://www.amstat.org/publications/jse/v19n1/gardasil.xls (data)

South Africa Project Outcomes dataset – (.xlsx file; see Blackboard folder)
SAProj_Outcomes_dataset_description.docx (description)
SAProj_Outcomes_data.xlsx (data)

South Africa project PSS item-level dataset – (SPSS .sav file; see Blackboard folder)
SAProj_PSS items_description.docx (description)
SAProj_PSSitems.sav (data)

Using the Gardasil dataset

Follow the exercises in the recording

Coding –

```
* Creating subsets of your original dataset - variables - 1st method ;
data gard_shortvar1 ; set gard1 ;
 keep agegroup new_race completed private_ins locationtype practicetype ;
 run ;

* Creating subsets of your original dataset - variables - 2nd method ;
data gard_shortvar2 ; set gard1 ;
 drop age race shots insurancetype medassist location ;
 run ;

* Alt - Creating subsets of your original dataset - variables - 1st method ;
data gard_shortvar1x ; set gard1 ;
 keep agegroup completed locationtype--private_ins ;
 run ;

* Creating a subset of your original dataset - observations - 1st method ;
data gard_young1 ; set gard1 ;
 if agegroup = 0 ;
 run ;

data gard_old1 ; set gard1 ;
 if agegroup = 1 ;
 run ;


* Creating subsets of your original dataset - observations - 2nd method ;
data gard_young2 ; set gard1 ;
 if agegroup = 1 then delete ;
 run ;

data gard_old2 ; set gard1 ;
```

```
 if agegroup = 0 then delete ;
 run ;

* Combining files ;
* Adding a 2 separate sets of observations together ;

data new_gard1 ; set gard_young1 gard_old1 ;
run ;
```

Using the South Africa Project datasets

Merging datasets together to add variables from two sources

Identify the variable in common to both datasets

Coding –

```
* Combining files ;
* Adding variables to an existing dataset ;

* Open the two South Africa Project datasets ;
data out1 ; set sastrain.saproj_outcomes ;
run ;

data pss1 ; set sastrain.saproj_pss ;
run ;


* In order to combine / merge variables from two datasets you have to sort each
dataset ;
proc sort data=out1 ;
 by studyno ;
 run ;

proc sort data=pss1 ;
 by studyno ;
 run ;


* When you do the merging you have to tell SAS what variable to use to combine the
records ;
data comb1 ; merge out1 pss1 ;
 by studyno ;
 run ;

* Check LOG - does the # of observations and # of variables make sense? ;
```

On your own -

Using one of the datasets you've saved –

Create a subset dataset that contains only some of the variables

Create a subset dataset that contains only some of the cases

Create a new dataset that re-combines the 2 separate sets of observations

Merge the 2 South Africa Project datasets - we'll be using the merged file in subsequent modules.

Document your work using your log record and any relevant output.

PART 4 – Examining relationships between variables

Examine variables and relationships between variables

Datasets used –

Gardasil Vaccination data set –
http://www.amstat.org/publications/jse/v19n1/gardasil.txt (description)
http://www.amstat.org/publications/jse/v19n1/gardasil.xls (data)

South Africa Project Outcomes dataset – (.xlsx file; see Blackboard folder)
SAProj_Outcomes_dataset_description.docx (description)
SAProj_Outcomes_data.xlsx (data)

Using the Gardasil dataset

Examine relationships between two categorical variables
Distribution
Statistical test - Chi-square
Plots

Coding –

```
* Look at bivariate relationships ;

* Between two categorical variables ;
proc freq data=gard1 ;
 tables race * completed / chisq ;
 run ;

proc freq data=gard1 ;
 tables new_race * completed / chisq ;
 run ;


* Plot relationship between 2 categorical variables ;
proc freq data=gard1 ;
 tables new_race * completed / plots = freqplot (twoway=grouphorizontal) ;
 run ;

proc freq data=gard1 ;
 tables new_race * completed / plots = freqplot (twoway=groupvertical) ;
 run ;

proc freq data=gard1 ;
 tables new_race * completed / plots = freqplot (twoway=stacked) ;
 run ;


* Repeat this with variables listed in reverse order ;
proc freq data=gard1 ;
 tables completed * new_race / plots = freqplot (twoway=grouphorizontal) ;
 run ;

proc freq data=gard1 ;
 tables completed * new_race / plots = freqplot (twoway=groupvertical) ;
 run ;

proc freq data=gard1 ;
 tables completed * new_race / plots = freqplot (twoway=stacked) ;
 title1 'frequency plot of categorical variables' ;
 run ;

proc freq data=gard1 ;
 tables completed * new_race / chisq ;
 tables completed * new_race / plots = freqplot (twoway=stacked) ;
 title1 'frequency plot of categorical variables' ;
 run ;
```

<u>Using the South Africa Project Outcomes dataset</u>

Examining relationships between two continuous variables

Distribution

Statistical measure - correlation

Plot

Proc Corr

Scatter – scatter plot for pairs of variables

Matrix – matrix of scatter plots of all variables

Coding –

```
* Getting a statistical measure of the relationship between two variables ;
proc corr data=sa1 pearson spearman kendall hoeffding ;
 var cd4t4abs hivviralload ;
 run ;


* Plot the relationship between two continuous variables ;
ods graphics on ;
proc corr data=sa1 plots = (scatter matrix) ;
 var cd4t4abs hivviralload ;
 run ;
```

Examining differences between groups on a continuous measure

Coding –

```
* Comparing values of a continuous variable between independent groups ;
* Doing this one variable at a time ;
proc ttest data=sa1 ;
 class treatment_gp ;
 var baseline_depression ;
 title1 ' ' ;
 run ;


* Creating statistical graphics with proc ttest ;
proc ttest data=sa1 plots (only) = (all) ;
 class treatment_gp ;
 var baseline_depression ;
 run ;


* Comparing values of a continuous variable collected at two time points ;
proc ttest data=sa1 ;
 paired baseline_depression * wk24_depression ;
 run ;


* Creating statistical graphics with proc ttest ;
proc ttest data=sa1 plots (only) = (agreementplot profilesplot) ;
 paired baseline_depression * wk24_depression ;
 run ;
```

On your own -
Using one of the datasets you've saved –

Run procedures that will show the relationships between variables

Include graphical representations of these analyses

Document your work using your log record and any relevant output.

PART 5 – More complicated coding in the Data step

Coding subscale scores
Computing mean scores
Using an array and do loop to code

Datasets used –

MOTIV8 Adherence Project Beliefs About ART dataset - (SPSS .sav file; see Blackboard folder)
MOTIV8_Beliefs_documentation.docx (description)
MOTIV8_Beliefs_train_data.sav (data; see Blackboard folder)

MOTIV8 Adherence Project Depressive Symptoms dataset - (SPSS .sav file; see Blackboard folder)
MOTIV8_DepressiveSymptoms_documentation.docx (description)
MOTIV8_Depression_train_data.sav (data; see Blackboard folder)

Using the MOTIV8 Adherence Project Beliefs About ART dataset

Compute subscale scores -

Necessity
Concern

Use 2 methods and compare results

Coding –

```
* Compute subscale scores using one method ;
data beliefs2 ; set beliefs1 ;
 necess1 = baa1 + baa3 + baa4 + baa7 + baa10 ;
 concern1 = baa2 + baa5 + baa6 + baa8 + baa9 ;
 run ;

* Compute subscale scores using another method ;
data beliefs3 ; set beliefs2 ;
 necess2 = sum(baa1, baa3, baa4, baa7, baa10) ;
 concern2 = sum(baa2, baa5, baa6, baa8, baa9) ;
 run ;
```

Compute mean scores -

Necessity items mean response
Concern items mean response

Coding –

```
* Compute mean scores ;
 data beliefs4 ; set beliefs3 ;
  necess_mean = mean(baa1, baa3, baa4, baa7, baa10) ;
  concern_mean = mean(baa2, baa5, baa6, baa8, baa9) ;
  run ;

* Compute mean scores ;
 data beliefs4x ; set beliefs5 ;
  necess_mean = mean(baa1, baa3, baa4, baa7, baa10) ;
  concern_mean = mean(baa2, baa5, baa6, baa8, baa9) ;
  run ;
```

Using the MOTIV8 Adherence Project Depressive Symptoms dataset

Using an array and do loop to

Reverse code the 4 items that are worded in the reverse direction - #4, 8, 12, & 16

Coding –

```
* Use a do loop to reverse code the 4 items - 4, 8, 12, & 16 ;

data depress2 ; set depress1 ;
 array items {4} ds4 ds8 ds12 ds16 ;
 array items_rev {4} ds4_rev ds8_rev ds12_rev ds16_rev ;
  do i = 1 to 4 ;
   items_rev{i} = 3 - items{i} ;
   end ;
run ;
```

Create a sum score

Coding –

```
* Create total score for depressive symptoms ;
data depress4 ; set depress3 ;
 drop i ;
 ds_total = ds1 + ds2 + ds3 + ds4_rev + ds5 + ds6 + ds7 + ds8_rev + ds9 + ds10 + ds11
+ ds12_rev + ds13 + ds14 +
            ds15 + ds16_rev + ds17 + ds18 + ds19 + ds20 ;
 run ;

* Create total score for depressive symptoms - alternate method ;
data depress5 ; set depress4 ;
 ds_total2 = sum(ds1, ds2, ds3, ds4_rev, ds5, ds6, ds7, ds8_rev, ds9, ds10, ds11,
ds12_rev, ds13, ds14,
            ds15, ds16_rev, ds17, ds18, ds19, ds20) ;
 run ;
```

Create a "cut-point" dichotomous variable that reflects if their sum score is considered "high"

Coding –

```
* Create dichotomous high group variable ;
data depress6 ; set depress5 ;
 if ds_total ge 16 then ds_high = 1 ;
 if ds_total lt 16 then ds_high = 0 ;
 if ds_total = . then ds_high = . ;
 label ds_high = 'Depressive Symptoms High Group designation' ;
 run ;
```

On your own -

After computing the "necessity" and "concern" subscale scores, save a new version of this SAS dataset.

After compute the total score and "high" score variables, save a new version of this SAS dataset.

Using one of the datasets you've saved –

Compute a subscale score

Compute a mean score

Use an array and do loop to code variables

Document your work using your log record and any relevant output.

PART 6 – Restructuring your datasets

Creating "horizontal" dataset from a "vertical" dataset
Vice versa

Datasets used –

MOTIV8 Adherence Project Beliefs About ART dataset - (SPSS .sav file; see Blackboard folder)
MOTIV8_Beliefs_documentation.docx (description)
MOTIV8_Beliefs_train_data.sav (data; see Blackboard folder)

MOTIV8 Adherence Project Depressive Symptoms dataset - (SPSS .sav file; see Blackboard folder)
MOTIV8_DepressiveSymptoms_documentation.docx (description)
MOTIV8_Depression_train_data.sav (data; see Blackboard folder)

South Africa Project Outcomes dataset – (.xlsx file; see Blackboard folder)
SAProj_Outcomes_dataset_description.docx (description)
SAProj_Outcomes_data.xlsx (data)

Create a "horizontal" dataset from a "vertical" dataset using the MOTIV8 Adherence Project Beliefs About ART dataset

Create a dataset that only includes evaluation #s 1, 24, and 48

Sort by subject # and evaluation number

Use Proc Transpose to create a new dataset, separately for necessity and concern scores

Rename variables as needed

Merge the necessity and concern "horizontal" datasets

Check to make sure your operations have worked correctly

Coding -

```
* Using your recently saved Adherence Project Beliefs dataset that includes scores ;
* Keep only variables that we need for restructuring ;

data beliefs1 ; set sastrain.beliefs2 ;
 keep subject evalnum necess1 concern1 ;
run ;

* Data on this scale were only collected during evaluations 1, 24, and 48 ;
* Keep observations for those evaluations only ;

data beliefs2 ; set beliefs1 ;
 if evalnum in(1, 24, 48) ;
 run ;

* Sort working dataset by identifier variables ;
proc sort data=beliefs2 ;
 by subject evalnum ;
run ;
```

```
* Restructure the dataset so that 3 sets of scores are in 1 record ;
proc transpose data=beliefs2 out=flip_nec ;
 by subject ;
 id evalnum ;
 var necess1 ;
 run ;

* Rename new variables so that this file can be merged with concern dataset ;
data flip_nec2 ; set flip_nec ;
 rename _1 = necess_1 ;
 rename _24 = necess_24 ;
 rename _48 = necess_48 ;
 run ;

* Do the same thing for the concern scores ;
* Restructure the dataset so that 3 sets of scores are in 1 record ;
proc transpose data=beliefs2 out=flip_con ;
 by subject ;
 id evalnum ;
 var concern1 ;
 run ;

* Rename new variables so that this file can be merged with concern dataset ;
data flip_con2 ; set flip_con ;
 rename _1 = concern_1 ;
 rename _24 = concern_24 ;
 rename _48 = concern_48 ;
 run ;

* Sort these 2 files together using subject value so you can then merge them ;
proc sort data=flip_nec2 ;
 by subject ;
 run ;

proc sort data=flip_con2 ;
 by subject ;
 run ;

* Merge together these 2 sorted files ;
data comb1 ; merge flip_nec2 flip_con2 ;
 by subject ;
 run ;
```

<u>Create a "vertical" dataset from a "horizontal" dataset using the South African Project Outcomes dataset</u>

Open your South African Project Outcomes dataset

Create a new working dataset that contains only the variables needed for this - studyno, baseline_depression, and wk24_depression

Use Proc Transpose to create a new dataset that has 2 records for each person

Check that the operation has worked correctly

Create two new variables that will be easier to work with -

evalnum

depress_score

Coding –

```
* Create a working dataset that has only the variables we will work with ;

data depr2 ; set depr1 ;
 keep studyno baseline_depression wk24_depression ;
 run ;

* To be on the safe side sort by your identifying variable first ;
proc sort data=depr2 ;
 by studyno ;
 run ;

* Transpose the data into the vertical formatted file ;
proc transpose data=depr2 out=depr3 ;
 by studyno ;
 var baseline_depression wk24_depression ;
 run ;

* Create evaluation number variable that is easier to work with ;
data depr4 ; set depr3 ;
 if _name_ = "Baseline_Depression" then evalnum = 1 ;
 if _name_ = "WK24_Depression" then evalnum = 24 ;
 depress_score = col1 ;
 run ;

* Clean-up file to keep only the variables you'll use ;
data depr5 ; set depr4 ;
 keep studyno evalnum depress_score ;
 run ;
```

On your own -

Using the MOTIV8 Depressive Symptoms file that includes the sum score and "high" group variable you created, create a transposed file that contains the data from the 3 evaluations where the data were collected (evalnum 1, 24, & 48) into 1 record for each person.

Use another available dataset and convert it from "horizontal" to "vertical"

Document your work using your log record and any relevant output.

FINAL PROJECT – DUE DATE: FRIDAY, DECEMBER 9, 2016

For the final project, use a dataset of your own choosing from those available through this course or a dataset of your own. Demonstrate your mastery of the coding elements that have been covered in the course, including: import the dataset, manipulate it as needed, and produce a statistical report using at least one graphical display and at least one descriptive statistical method. This final project should include a written explanation of the results. The dataset you work with for this final project should include at least four variables of which two variables will be measured using continuous data and two variables measured using categorical data.

Copy the log file to an external file that can be sent to me for review.
Copy/save the output (or Results Viewer) so I can see the results of our coding.
Send me your program file(s) so I can review the coding you've created.