SAS Tutorial for Lab 1

Author: Leonore Findsen, Cheng Li, Min Ren

0. Downloading Data

You will have to download all of the data used from the internet before SAS can access the data.

If the file accessed via a link, then right click on the file name and save it to a directory on your hard drive. It can be difficult to find the files, so if you are unfamiliar with the PC, I suggest that you download it directly to the W drive (main campus drive).

SAS can not read a zipped file. Therefore, if any file is zipped, remember to extract it before you try to access it in SAS.

Please remember where your files are stored so that you can access them for the labs.

1. General Information for SAS

a) To help in debugging SAS code, the SAS editor color codes the information. The colors can help you debug the code so I recommend that you memorize them or refer back to this listing. The following are some of the colors used by SAS:

blue or blue: commands	black: responses	green: comments
blue green: numbers	purple: text/titles	black: data

b) In the code that is presented, there are a large number of comments (in green). Please read them to understand what the variables represent. Comments are not required in your submitted code.

c) All command lines in SAS must end in a semicolon ';'.

d) To run a program, be sure that the editor window is active and then Run → Submit or just click on the  . In addition to running all of the code, you may highlight parts of the code and only run that part.

e) If the file does not run as you expect, please look in the log file; specifically at the information in red or green to see if that helps you find the problem.

f) SAS will append the output to the Result Viewer and Log screens which can cause great confusion when you are doing more than one problem in a session. If you are running the programming locally, that is, you are NOT using goremote, I would strongly suggest that you add the following code to the beginning of each program, this will clear both of these screens when you run the code:

```
ods html close;
ods html;
```

g) Even though the "proc print" line is not required, I strongly recommend that you keep this in your code whenever you change your data to be sure that it is changed correctly. Remember to ALWAYS limit the number of observations when you are using large data sets. Please do NOT turn in the output from this code unless explicitly asked.

h) Data set names need to be one word (no spaces) and start with a letter. Capitalization is important. Not all special characters can be used. Please change your data set names and variable names to match each situation. Points will be deducted if this is not done.

i) When you submit the code in your lab report, please include all of the code for the question at the beginning of the question. This makes it easier to replicate your results.

2. Importing Data sets, cleaning, manipulating, and printing them.

Hummingbirds and flowers. (Data Set: ex01-88helicon_m.txt) Different varieties of the tropical flower *Heliconia* are fertilized by different species of hummingbirds. Over time, the lengths of the flowers and the form of the hummingbirds' beaks have evolved to match each other. Here are data on the lengths in millimeters of three varieties of these flowers on the island of Dominica:

H. bihai							
47.12	46.75	46.81	47.12	46.67	47.43	46.44	46.64
48.07	48.34	48.15	50.26	50.12	46.34	46.94	48.36
***H. caribaea* red**							
41.90	42.01	41.93	43.09	41.47	41.69	39.78	40.57
39.63	42.18	40.66	37.87	39.16	37.40	38.20	38.07
38.10	37.97	38.79	38.23	38.87	37.78	38.01	
***H. caribaea* yellow**							
36.78	37.02	36.52	36.11	36.03	35.45	38.13	37.1
35.17	36.82	36.66	35.68	36.03	34.57	34.63	

```
data helicon;
  infile 'W:/ex01-88helicon_m.txt' delimiter = '09'x firstobs = 2 ;
  input variety $ length;
  /*delimiter = '09'x means the file uses tab as delimiter
  firstobs = 2 means we start reading the data from the second line,
     since the first line is the name of variables.*/
  /*Since variety is a categorical variable, a '$' follows it.*/
run;

*For small data sets only;
proc print data = helicon; run ;
```

SAS Tutorial for Lab 1

Author: Leonore Findsen, Cheng Li, Min Ren

```
*For large data sets with a large number of variables;
proc print data=helicon (obs = 10);
  * the obs keyword has to be in parentheses;
  var length; *only the variables listed will be printed;
run;

proc print data=helicon (obs = 10);
  var length variety;
  *The variables will be printed in the other listed.
   The variables should be separated by spaces.;
run ;
```

For small data sets with a small number of variables, you should use the first "proc print" statement. For large data sets with a large number of variables, use the second "proc print" statement. DO NOT USE THE FIRST "PROC PRINT" STATEMENT WITH LARGE DATA SETS!

For small data sets: Since this is a small data set with a small number of variables, the first "proc print' statement is used.

```
*For small data sets only;
proc print data = helicon; run ;
```

The output is below:

The SAS System

Obs	variety	length
1	bihai	47.12
2	bihai	46.75
3	bihai	46.81
4	bihai	47.12
5	bihai	46.67
6	bihai	47.43
7	bihai	46.44

I have highlighted the first five data points.

You will see some error messages because there is some missing data. We will be removing those values later.

SAS Tutorial for Lab 1

Author: Leonore Findsen, Cheng Li, Min Ren

Please name your data set and variables in context. In this case, we are interested in *Heliconia* flowers, so I named the data set helicon. The variables in the text file are Variety and Length so that is what I use those in the input statement. Since capitalization is important, I often only use small letters. If you want to put a space in the name, use an underscore, _. Note that there is a maximum length in SAS for both data set names and variable names so be careful.

If you want to copy tables (or parts of tables) from the SAS output, I suggest that you use the Snipping Tool. You can also use that tool to highlight your answer. This is the procedure that I used in the above table. I used the highlighting function from the Snipping Tool to highlight the data points.

For large data sets: Do NOT use the code for small data sets when you have a large data set like is used in our assignment. Use the following code which is repeated from above:

```
*For large data sets with a large number of variables;
proc print data=helicon (obs = 10);
  * the obs keyword has to be in parentheses;
  var length; *only the variables listed will be printed;
run;
```

The code will print out the first 10 data points for variable Length only. Remember ONLY use this procedure for the assignment. Even though it is not necessary for this data set, to use the code for large data sets, the output from the code is below::

Obs	length
1	47.12
2	46.75
3	46.81
4	47.12
5	46.67
6	47.43
7	46.44
8	46.64
9	48.07
10	48.34

If you want to print out more than one variable, separate them by spaces. The variables will be printed out in the order specified. For example, if I wanted to switch the order that the variables were printed out, I would use the code:

SAS Tutorial for Lab 1

Author: Leonore Findsen, Cheng Li, Min Ren

```
proc print data=helicon (obs = 10);
  var length variety;
  *The variables will be printed in the other listed.
   The variables should be separated by spaces.;
run ;
```

Obs	length	variety
1	47.12	bihai
2	46.75	bihai
3	46.81	bihai
4	47.12	bihai
5	46.67	bihai
6	47.43	bihai

Cleaning and saving datasets

Since there is often missing values in datasets, it is important to be able to remove that data before we start the analysis. The following code will remove data sets that are not complete.

```
data helicon_cleaned;
   set helicon;
   if nmiss(of _NUMERIC_)=0; /* only outputs the data if the numeric
                                data is numeric, not text) */
run;

proc print data=helicon_cleaned; run;
```

Remember for our data set we are using in the assignments, DO NOT PRINT OUT THE WHOLE DATA SET.

In the above code, the new data set is called ‘helicon_cleaned’ which is based on the old data set called ‘helicon’.

Once you have cleaned the data set, you will want to save it so that you don't have to clean the data each time you use the data set. The procedure to save the new data set is as follows:

File → Export Data → In the 'Member', select the appropriate data set, in this case, it would be helicon_cleaned, be sure to select "Write variable labels as column names" (Fig. 1) → Next → Select the Tab Delimited Files (*.txt) (Fig. 2) → Next → Where do you want to save the file? → Browse - Browse on your computer for the appropriate location (I would suggest your W: drive). As I do not have a W: drive on my office computer, I am choosing another location (Fig. 3) → Finish

SAS Tutorial for Lab 1

Author: Leonore Findsen, Cheng Li, Min Ren

Use the following screen shots as guides

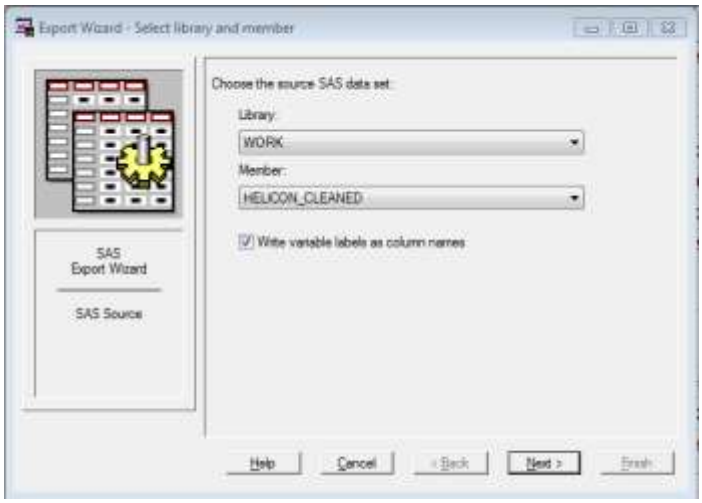

Fig. 1

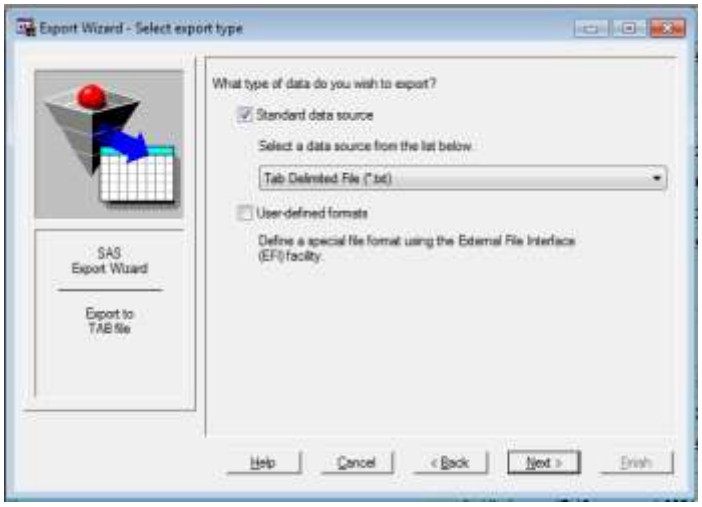

Fig. 2

SAS Tutorial for Lab 1

Author: Leonore Findsen, Cheng Li, Min Ren

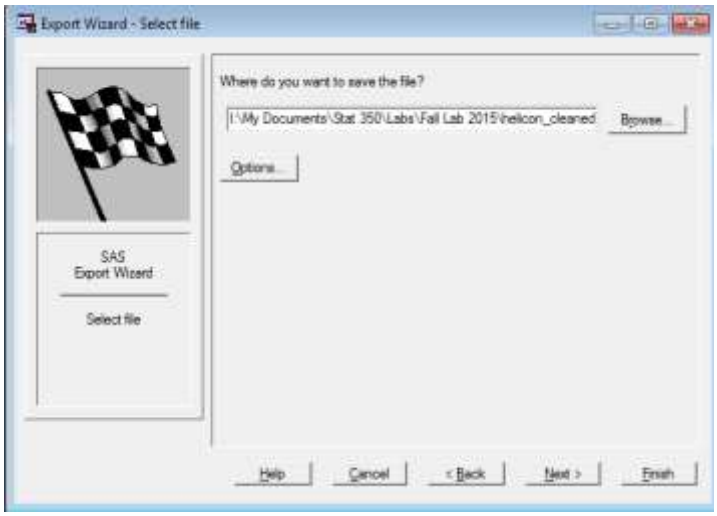

Fig. 3

Manipulating Data

For readability, you might want to change the shortened name or abbreviation to the full version. This is done by the following code:

```
data helicon_new;
   length variety $ 15;
   set helicon_cleaned;
   if variety = 'red' then variety = 'Carabaea_Red';
   if variety = 'yellow' then variety = 'Carabaea_Yellow';

proc print data=helicon_new; run;
```

Again, DO NOT PRINT OUT THE WHOLE DATA SET for the airline data.

Some of the output is shown below:

36	Carabaea_Red	38.23
37	Carabaea_Red	38.87
38	Carabaea_Red	37.78
39	Carabaea_Red	38.01
40	Carabaea_Yellow	36.78
41	Carabaea_Yellow	37.02
42	Carabaea_Yellow	36.52
43	Carabaea_Yellow	36.11

Author: Leonore Findsen, Cheng Li, Min Ren

If the length command is not used, then the name would be truncated. Always use a length command if you are replacing an abbreviation with a full name or increasing the length of a categorical variable. This command needs to be placed right after the data statement.

In addition, often you want to create a new variable based on mathematical operations from old variable(s). You can use the sample code below to convert the lengths of the beaks from millimeters to inches. The conversion factor is 1/25.4.

```
data helicon_new;
   set helicon_new;
   length_inches = length/25.4;

proc print data=helicon_new; run;
```

You will see a new column in the data set called length_inches:

Obs	variety	length	length_inches
1	bihai	47.12	1.85512
2	bihai	46.75	1.84055
3	bihai	46.81	1.84291
4	bihai	47.12	1.85512
5	bihai	46.67	1.83740
6	bihai	47.43	1.86732
7	bihai	46.44	1.82835

I strongly recommend that if you are making changes to a data set that you change the name of the data set. When performing calculations, keep in mind which data set that you are using.