Lab #1: Introduction to SAS

September 7, 2023

**Getting Started**

In THSS 730 you will be using SAS OnDemand for Academics (SAS OnDemand), a cloud-based version of SAS for academic use. The user interface is called SAS Studio.

To get started with SAS OnDemand, go to <https://welcome.oda.sas.com> and sign in using your SAS Profile email address or SAS OnDemand user name. Once signed in, click on SAS Studio to begin.

Once you open the software, you’ll see that SAS Studio has two basic windows. The Navigation Pane houses the files, folders, tasks, snippets, libraries, and file shortcuts. The Work Area contains four tabs: Program, CODE, LOG, and RESULTS.

*Navigation Pane:*

The Navigation Pane contains five menu items. Expand each item to view its contents.

* Server Files and Folders: Displays the content that is visible to SAS OnDemand.
* Tasks and Utilities: Contains menus for data manipulation, graphics, and statistical analysis.
* Snippets: Contains saved segments of SAS syntax or code that can be edited. You can

also add your own snippets.

* Libraries: Contains a collection of SAS data sets. (SAS data sets are stored in libraries)
* File Shortcuts: Provides shortcuts similar to what’s in the Microsoft environment.

*Work Area:*

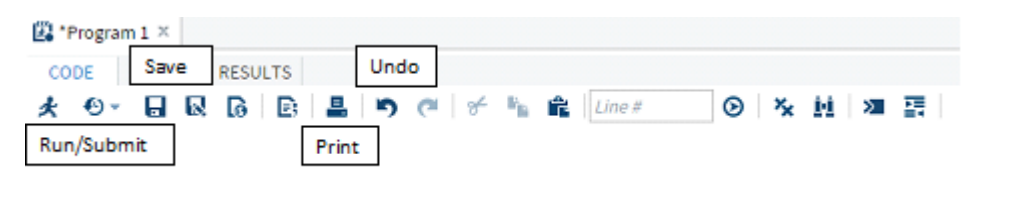
The Work Area has tabs that display different elements of your program or task.

* Program: This tab displays the active program.
* CODE: This tab is where you type, edit, and submit your program.
* LOG: This tab contains notes about your SAS session. After you submit your program, it

shows any notes, errors, or warnings. These items are color-coded.

* RESULTS: This tab shows your output and generates any printable results.

*Tool Bar:*



**The Basics**

To be able to analyze data that you have collected, the data must be in a format that SAS can

read (a SAS data set). The rows in a SAS data set are observations and columns are variables.

When naming a SAS variable, there are a few rules to keep in mind:

* Name must be 32 characters or fewer in length.
* Name must start with a letter or an underscore. It cannot begin with a numeral or

contain special characters.

* Names should be relevant and simple.

In SAS, there are two data types—numeric and character.

* Numeric variables are numbers that can be positive or negative and added or subtracted.
* Character variables contain any value including letters, numbers, special characters, and blanks.
* A missing value is noted with a period (.) for numeric data and with a blank space for character data.

SAS uses statements to write a series of instructions called a program. A typical SAS program is constructed of DATA and PROC steps. DATA steps are used to create and manage SAS data sets. PROC steps (PROCedures) are used to process or analyze SAS data sets.

Important notes about SAS syntax:

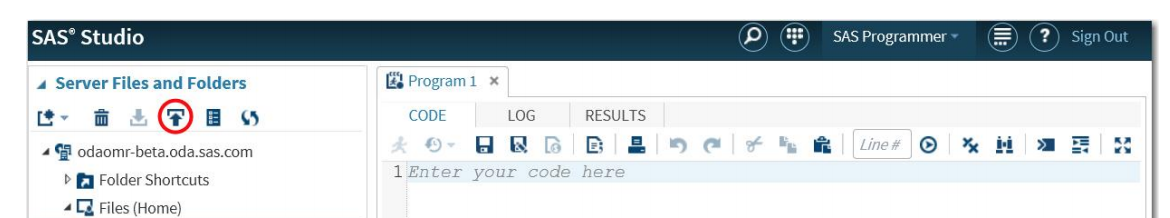
* SAS is not case sensitive.
* Every SAS statement ends with a semicolon.
* You can use /\* and \*/ around text to add comments to your code.

**Importing Data**

In THSS 730, you will be provided with Excel (.xlsx) files via Canvas when needed for labs, problem sets or exams.

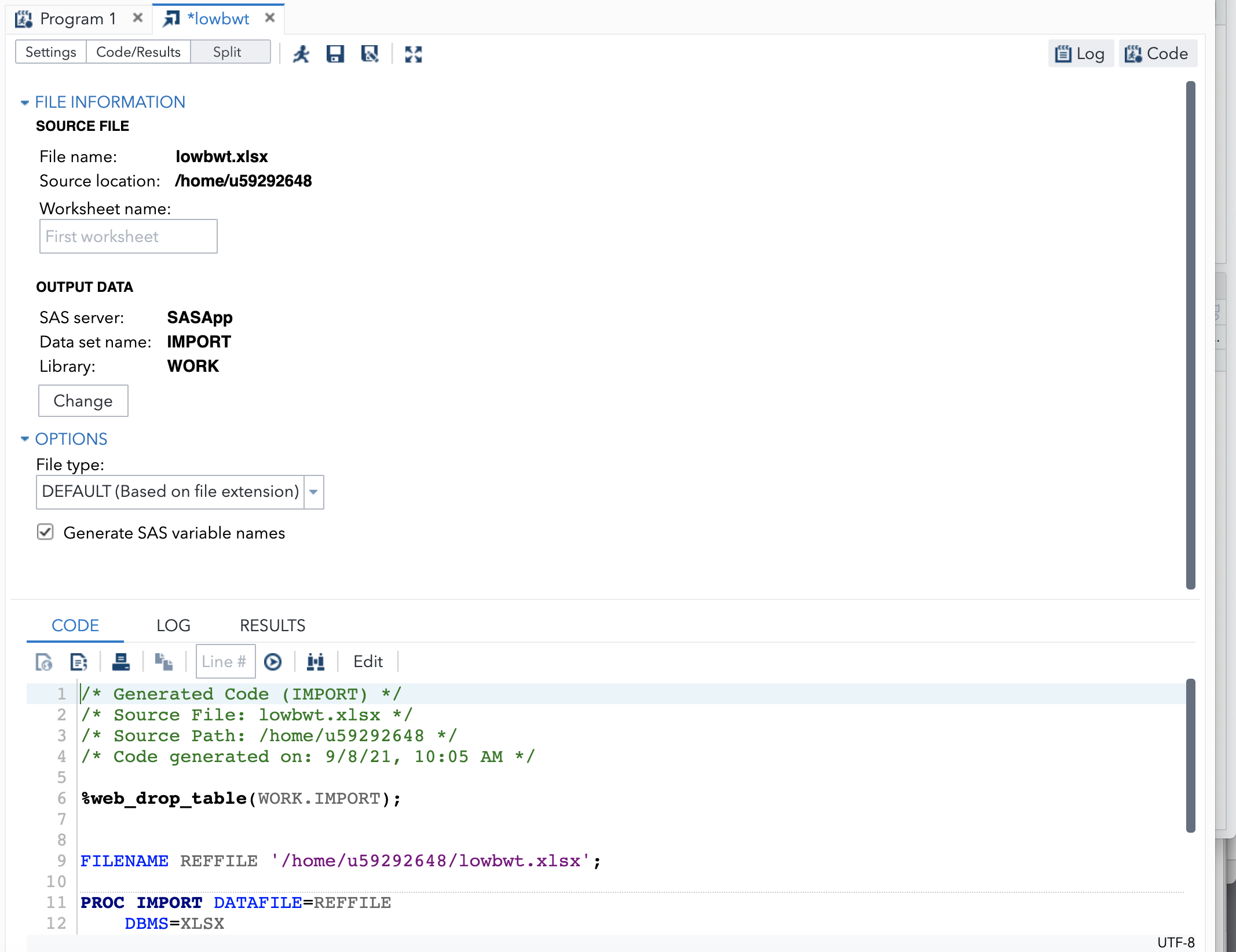
For this lab we will use the “lowbwt.xlsx” data set. The file can be found under Lab #1 under Modules. After you download the file, the following steps will allow you to import the data.

1. Expand Files (Home) in the Navigation pane within Server Files and Folders.
2. You can upload data directly into the Files (Home) folder by clicking on the upload icon (see below) and choosing the file from wherever it is saved on your computer.
3. The file will then show up under Files (Home).



Once the file is visible in the Navigation pane you can then import the data.

1. Right click on the data file in the Navigation pane and select Import Data.
2. This will open up a new tab in the Work Area. In the top of the window of the new tab you will see information about the source file and the output data. For the source data (i.e., your .xlsx file), as a default, SAS will read in the first worksheet. You would only need to edit this if you are trying to read in a worksheet other than the first. For the output data, the default is to call the SAS data set IMPORT. If you want to change this click on Change and then replace IMPORT with whatever you want to call the data set (LOWBWT for this example).
3. Click on the Run icon. This will import the .xlsx file and create a SAS data set.
4. In the bottom window in the new tab under RESULTS you will see some information about the data set. Under OUTPUT DATA you can view the imported data. Under CODE you will see the code used to import the .xlsx file. You could use that code to import the data instead of using the Import Data menu as described above.
5. You now have a SAS data set stored in the WORK library.



**Inputting Data**

If you do not have an external data file to import, you can enter data directly into SAS. Suppose we have the following small dataset:

|  |  |
| --- | --- |
| Gestational Age (weeks) | Birth Weight (grams) |
| 39 | 3544 |
| 39 | 3430 |
| 41 | 3770 |
| 35 | 2522 |
| 39 | 3997 |
| 36 | 3530 |
| 41 | 3402 |
| 44 | 4355 |
| 39 | 2977 |
| 39 | 3474 |

This data set can be entered into SAS by entering the following code in the Work Area under CODE:

DATA baby;

INPUT gestage weight;

CARDS;

39 3544

39 3430

41 3770

35 2522

39 3997

36 3530

41 3402

44 4355

39 2977

39 3474

;

RUN;

**Describe/Print Data Set**

After you import or input a data set into SAS, you may want to review your data. As we have already seen, you can view the data under OUTPUT DATA. You can also go to Libraries in the Navigation Pane and double click on the data set under WORK in My Libraries.

You may also choose to print all or parts of a data set using PROC PRINT or see a summary of your data set using PROC CONTENTS.

PROC print data=baby;

RUN;

PROC contents data=baby;

RUN;

**Descriptive Statistics**

Going back now to the “lowbwt.xlsx” example. The data set has the following variables:

* sbp = systolic blood pressure
* sex = sex (1=male; 0=female)
* tox = toxemia diagnosis for mother (1=yes; 0=no)
* grmhem = germinal matrix hemorrhage (1=yes; 0=no)
* gestage = gestational age
* apgar5 = apgar score at 5 minutes

To describe quantitative data, PROC UNIVARIATE and PROC MEANS are useful.

PROC univariate data=lowbwt;

VAR sbp;

RUN;

PROC means data=lowbwt;

VAR sbp;

RUN;

PROC means data=lowbwt n mean std median p25 p50 min max;

VAR sbp;

RUN;

Suppose we want to describe systolic blood pressure for males and females separately.

PROC sort data=lowbwt;

BY sex;

RUN;

PROC univariate data=lowbwt;

VAR sbp;

BY sex;

RUN;

To describe categorical data, PROC FREQ is useful.

PROC freq data=lowbwt;

TABLE tox;

RUN;

Suppose we want to describe toxemia status for males and females separately.

PROC sort data=lowbwt;

BY sex;

RUN;

PROC freq data=lowbwt;

TABLE tox;

BY sex;

RUN;

Or you could do a cross-tabulation instead:

PROC freq data=lowbwt;

TABLE tox\*sex;

RUN;

Note: You can include more than one variable the VAR or TABLE lines in PROC UNIVARIATE, MEANS and FREQ.

**Graphical Displays**

If you add the plot option to PROC UNIVARIATE, you can get a histogram and box plot side by side.

PROC univariate data=lowbwt plot;

VAR sbp;

RUN;

You can also use PROC UNIVARIATE to get a standard histogram.

PROC univariate data=lowbwt;

VAR sbp;

HISTOGRAM sbp;

RUN;

PROC FREQ can be used to make bar charts. To make a simple bar chart:

PROC freq data=lowbwt;

TABLE tox / plots=FreqPlot;

RUN;

PROC freq data=lowbwt;

TABLE tox / plots=FreqPlot(scale=Percent);

RUN;

To make a clustered bar chart:

PROC freq data=lowbwt;

TABLE tox\*sex / plots=FreqPlot(twoway=cluster);

RUN;

PROC SGPLOT is used to make box plots. To make a simple box plot:

PROC sgplot data=lowbwt;

VBOX sbp;

RUN;

To compare groups using box plots (side-by-side box plots):

PROC sgplot data=lowbwt;

VBOX sbp / category=sex;

RUN;

**Things You Can Do in DATA Steps**

*Create New Datasets:*

Suppose we only interested in male babies.

DATA male;

SET lowbwt;

IF sex = 1;

RUN;

Suppose we only interested in babies with systolic blood pressure greater than 50.

DATA sbp;

SET lowbwt;

IF sbp > 50;

RUN;

*Label Variables:*

Not everyone will understand what your variable names mean, so it can be helpful to label variables when displays will be shared with others.

DATA lowbwt2;

SET lowbwt;

LABEL sbp = "systolic blood pressure"

tox = "toxemia diagnosis for mother";

RUN;

Now re-run the boxplot code with the new labeled data set.

PROC sgplot data=lowbwt2;

VBOX sbp;

RUN;

*Create New Variables:*

Suppose you want to create new variables where you add 10 to the SBP value and multiply the SBP value by 100.

DATA lowbwt2;

SET lowbwt2;

sbpplus10 = sbp + 10;

sbptimes100 = sbp\*100;

RUN;

Suppose you are interested in male infants with mothers who had toxemia. Create a variable where 1 = male infants with mothers who had toxemia and 0 = all other infants.

DATA lowbwt2;

SET lowbwt2;

maletox =.;

if sex=1 and tox=1 then maletox = 1;

else maletox=0;

RUN;

Suppose you are interested in extremely preterm infants (gestational age less than 28 weeks). Create a new variable where 1 = extremely preterm and 0 = not extremely preterm.

DATA lowbwt2;

SET lowbwt2;

expreterm =.;

if gestage < 28 then expreterm = 1;

if gestage >= 28 then expreterm = 0;

RUN;

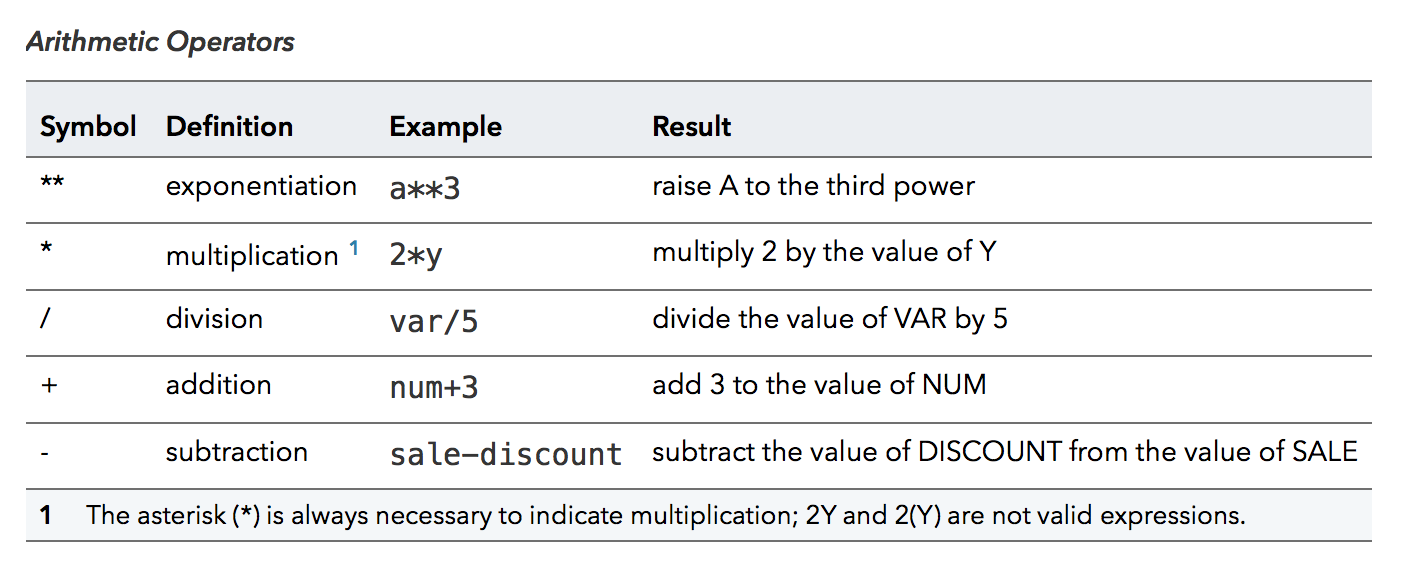
Now check out the frequency of our new variables:

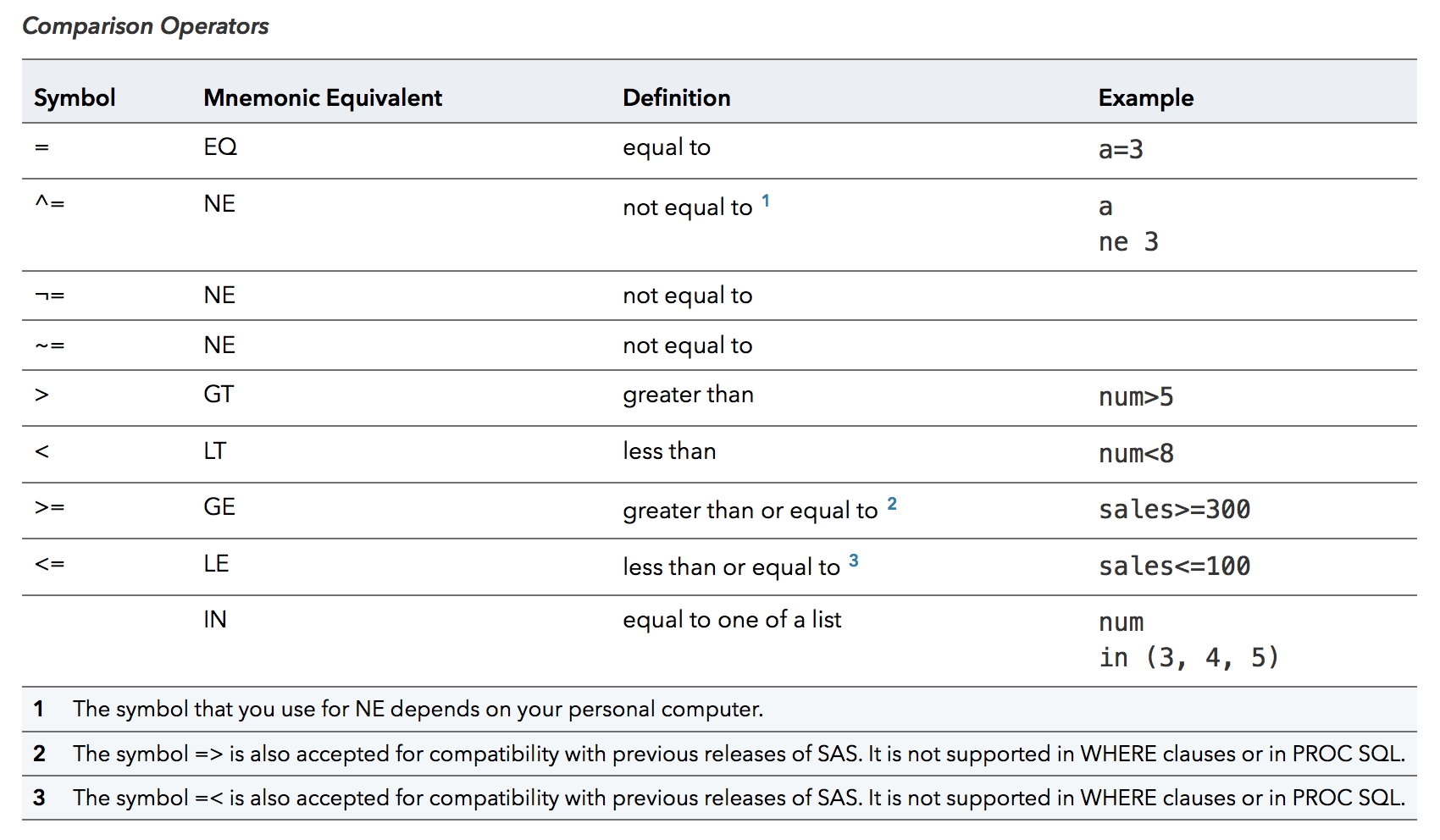
PROC freq data=lowbwt2;

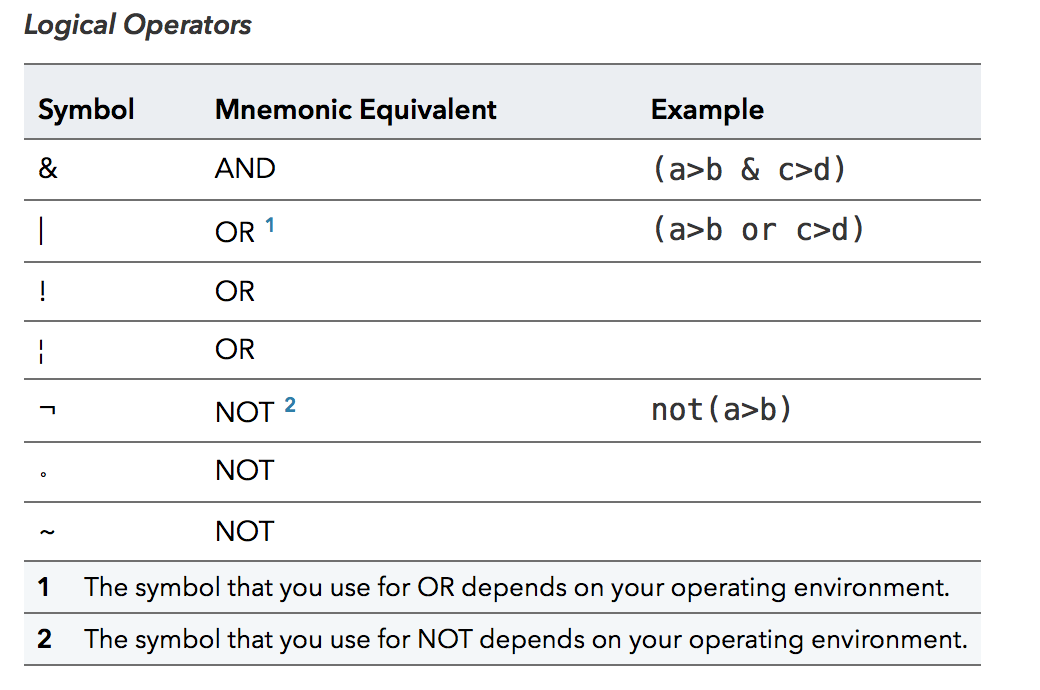
TABLE expreterm maletox;

RUN;

Note: these examples assume there is no missing data.

****

****

****

**Value Labels**

For categorical variables, you may want to label the values that the variables take. For example, the variable sex takes the value 1 for male and 0 for female. We may want tables and graphs to say male and female instead of 1 and 0 to make reading them easier. To do this, we need to create value labels. This is a two-step process. First, PROC FORMAT is used to create value formats.

PROC FORMAT;

VALUE sexf 1 = ‘male’

0 = ‘female’;

RUN;

Now that the format has been created, it can be assigned to the variable in a DATA step using a FORMAT statement.

DATA lowbwt3;

SET lowbwt2;

FORMAT sex sexf.;

RUN;

Now re-run the boxplot code with the new data set with value labels.

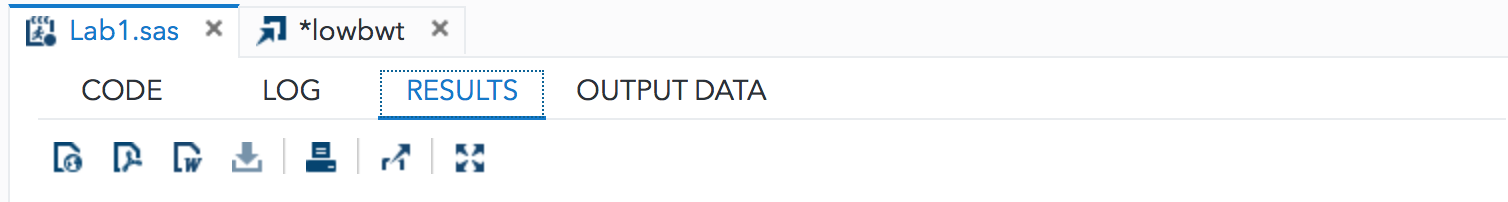
PROC sgplot data=lowbwt3;

VBOX sbp / category=sex;

RUN;

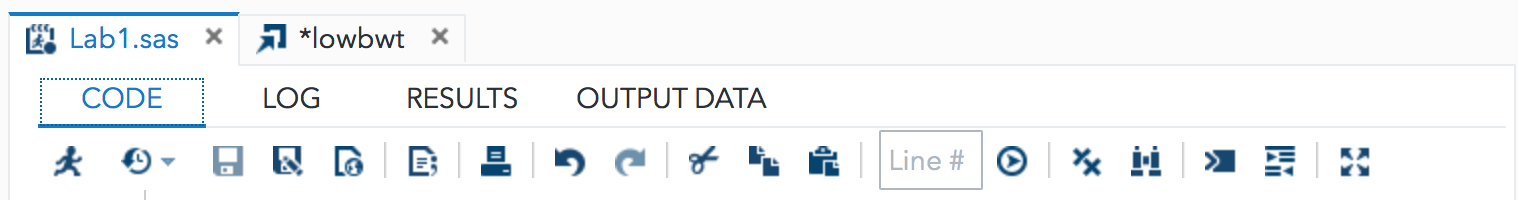
**Saving Results and Code**

You can copy results from the RESULTS tab and paste into a Microsoft Word document. Alternatively, you can download an RTF file. This file will contain all the results that you see in the RESULTS tab. RTF files can be opened in Microsoft Word.





You can save code from the CODE tab by clicking on the disk icon.





**Libraries**

Libraries are where SAS data sets are stored. When you import a data file, the default library is WORK. This library is temporary. Files stored in WORK are available during the current session, but are not saved when you end a SAS OnDemand session. Everything we have done so far is in the WORK library. If you want to save SAS data files to use in a later session, you will need to create a permanent library to store the SAS data file.

To create a library, go to the Navigation pane under Server Files and Folders. Right click on Files (Home) and select Create, then Library. Enter a name (e.g., THSS) and then click OK.

You should now be able to see your new library under Libraries in the Navigation Pane.

You can then use a DATA step to save a SAS data set to the library.

DATA THSS.lowbwt;

SET lowbwt2;

RUN;

The LOWBWT data set now shows up under THSS in Libraries in the Navigation Pane. To be able to access this SAS data set in a new SAS session you will need to use a LIBNAME statement so SAS knows where to find the data set.

LIBNAME THSS “file path”;

You can find the file path by right clicking on the Library in the Navigation Pane and selecting Properties. Copy the path and paste inside the quotation marks.

To refer to the data set in your code you should use data=THSS.lowbwt instead of just lowbwt.

**Extra Practice**

Here is a tutorial on creating new variables:

<https://libguides.library.kent.edu/SAS/ComputeVariables>

Here is another tutorial focused on if-then statements:

<https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/SAS/SAS-Basics1/SAS-Basics16.html>

1. Create a new variable called gestage\_days that reports gestational age in days rather than weeks.
2. Suppose we are interested in comparing characteristics of infants who have an Apgar score lower than 7 to those with an Apgar score of 7 or greater. Create a new variable that will allow us to classify the infants based on Apgar score.
3. Suppose we are particularly interested in female infants with a systolic blood pressure less than 60. Create a variable that equals 1 for this group and 0 for the other infants.