STAT 8678 - SAS Programming & Data Analysis

Chi-Kuang Yeh

2026-01-01

Table of contents

Preface

Description

This course covers programming using the SAS statistical software package, and it provides an introduction to data analysis stressing the implementation using SAS.

Topics include two main parts:

1) **SAS Programming**: data management and manipulation, basic procedures, macro programming;
2) **Data Analysis**: descriptive statistical analysis, one- and two-sample inference, basic categorical data analysis, regression analysis, and other selected topics.

Prerequisites

MATH 4544/6544, or equivalent.

Instructor

Chi-Kuang Yeh, Assistant Professor in the Department of Mathematics and Statistics, Georgia State University.

- Office: Suite 1407, 25 Park Place.
- Email: cyeh@gsu.edu.

Office Hour

TBA and By appointment

Grade Distribution

- TBA

Assignment

☐ TBA

Midterm

☐ TBA

Topics and Corresponding Lectures

Those chapters are based on the lecture notes. This part will be updated frequently.

Topic	Lecture
Introduction to SAS and modules	1–

Recommended Textbooks

- Statistics 480: Introduction to SAS, The Pennsylvania State University.
- SAS Training, SAS Institute.
- SAS Resources, University of California, Los Angeles.

Acknowledgments

Special thanks to Li-Hsiang Lin for providing the base materials given on this website.

Part I

Introduction

1 Introduction to Basic SAS Operation

Learning objective:

1. Familiarize ourselves with SAS windows (editor, log, output)
2. Create a dataset
3. Sorting Data (by 1 or more variables)
4. Obtain summary statistics of variables

1.1 Introduction to SAS

Q: Wha is SAS?

SAS (Statistical Analysis Software) is a prominent tool in the field of Data Analytics, offering a comprehensive suite for data manipulation, mining, management, and retrieval across various sources, coupled with robust statistical analysis capabilities. It excels in a range of functions including data management, statistical analysis, report generation, business modelling, application development, and data warehousing. SAS is user-friendly, featuring a point-and-click interface for those without technical expertise, while also providing deeper functionality through the SAS programming language. This software is instrumental in employing qualitative methods and processes that enhance employee productivity and business profitability.

Within SAS, data extraction and categorization into tables are pivotal for identifying and understanding data trends. This versatile suite supports advanced analytics, business intelligence, predictive analysis, and data management, facilitating effective operation in dynamic and compet- itive business environments. Additionally, SAS's platform-independent nature allows it to operate seamlessly across various operating systems, including Linux, Windows, Mac, and Ubuntu. SAS provides extensive support to programmatically transform and analyze data in the comparison of drag and drop interface of other Business Intelligence tools. It provides very fine control over data manipulation and analysis.

1.1.1 SAS Installzation

Georgia State University (GSU) has purchased license, so we can access SAS University Edition for free!

To install SAS University Edition, choose from the following options:

- **Option 1**:

  Download on your personal PC: Free SAS license available to GSU students, faculty, and staff via Technology Services (download required; check system requirements): Download from https://technology.gsu.edu/technology-services/software-equipment/university-licensed-software/ (Need to log-in from your GSU Account)

  Get Help for the Installation from https://gsutech.service-now.com/sp

- **Option 2**:

  - On Campus Access: SAS can be found on all GSU Library PCs: Floors 1-4 (not available on Library Macs, because there is no Mac version of SAS)
  - Graduate Biostatistics Computer Lab (SPH): 6th floor of the Urban Life building (swipe card access required)
  - Common MILE Lab whose opening time is
    * Monday & Wednesday: 9 – 18
    * Tuesday & Thursday: 9 – 17
    * Friday: 9 – 15

- **Option 3**:

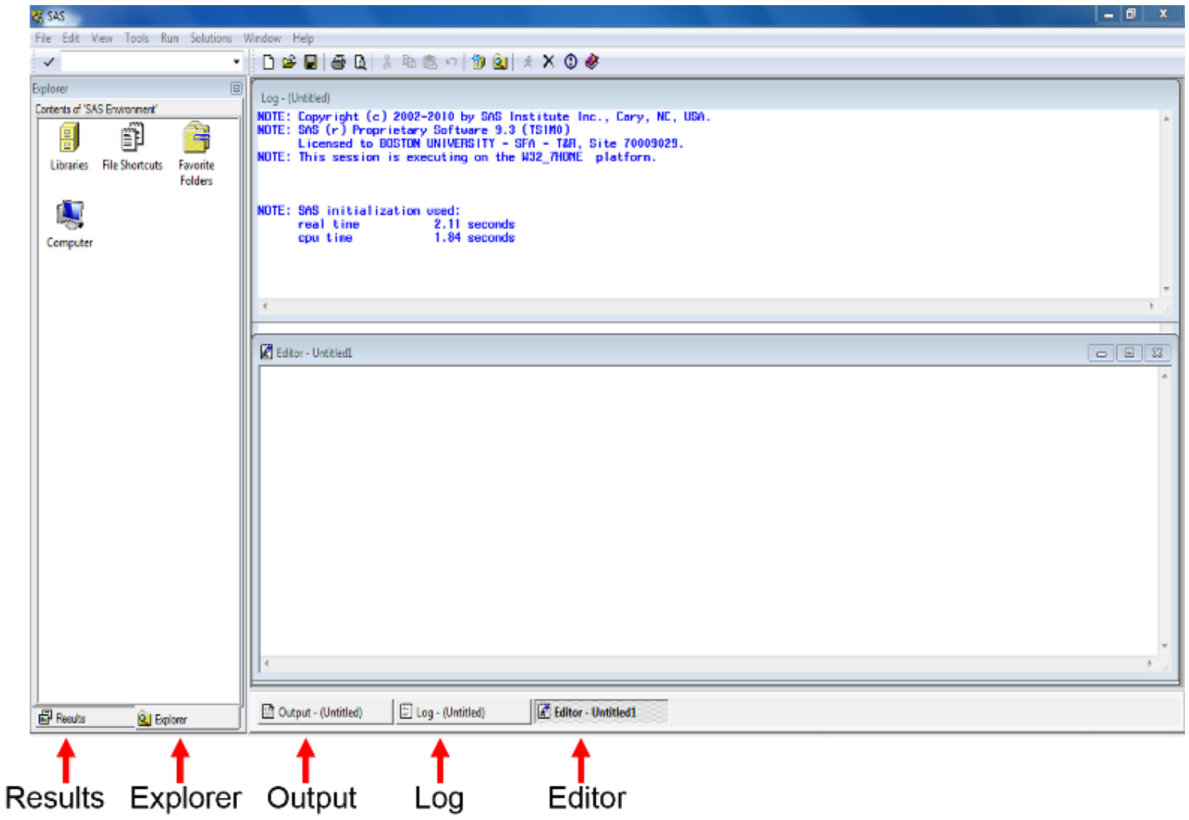
  Access via VLab, GSU's Remote Desktop Environment. Download and Connect to Cisco AnyConnect Client to connect to GSU's VPN (secureaccess.gsu.edu). Once connected to the VPN, login to VLab at: https://vlab.gsu.edu/ to access SAS.

- **Option 4**:

  Access via SAS OnDemand for Academics/SAS Studio. If you do not already have one, create a SAS profile at https://welcome.oda.sas.com/ Then, sign in with credentials and click SAS®Studio to access the web-based SAS environment.

1.1.2 SAS Windows

Once SAS has started, the screen will look similar to the following: The main SAS window is divided into several sub-windows:



- The menu and toolbar along the top of the window
- The explorer/results browser along the left hand side, where you can a listing of the results of successful SAS program.
- The program editor below the log on the bottom right, where you create your SAS program
- The windows bar along the bottom for you to switch all windows.

The Editor (Program Editor) window is a text editor that facilitates writing SAS programs (code). The Log window displays system messages, errors, and resource usage and is thus used to review program statements. The Output window displays output from statistical procedures run within the SAS program; however this is no longer the default. In SAS 9.3 output is sent to the Results Viewer which opens automatically when you run a procedure that generates output. The Results window displays a map of the Output window, and is useful for navigating the results of complicated analyses. Finally, the Explorer window contains all of the data sets in the current SAS session.

These windows can be moved or resized as desired. Only one SAS window is active at a time. The active window will have a shaded title bar at the top of the window, and a highlighted windows bar at the bottom of the screen. In the above example, the Program Editor is the

active window, with an "Untitled" program name. Note that the menu options for the SAS toolbar along the top of the screen depend on which window is currently active. (The active window can be changed by clicking on that window with the mouse, or by selecting the desired window from the Window menu.)

1.2 SAS Examples

Our first task in using SAS will be to create a small dataset and "print" that dataset to the output window. As we mentioned in previous paragraph SAS programs usually start with a DATA step where the dataset is created. Once the dataset is available, various procedures can be run on the dataset. The example below is written in the SAS Program window. The program creates a dataset called "People" with 3 variables (columns) which are 'gender', 'height', and 'weight' and 14 observations (rows). Note that the values of the variables on each line are separated by one or more blanks. A few other things that you should note:

- All SAS statements end with a semicolon (**;**)
- More than one SAS statement can be put on a line, or a SAS statement can continue across several lines, if every statement ends with a semicolon.
- Data listed as part of the program is also terminated with a semicolon. Data does not have to be entered in the program; it can also be read from files that are external to the SAS program (more on that next week)
- gender is a character variables as indicated by the $, and height and weight are numeric variables.
- Once the dataset is created, various SAS procedures (called PROCs) can be used to analyze the data and present results. We will start with a listing of the data created with a procedure called PROC PRINT.

```
title1 'STAT 8678 Example 1';
title2 'Your name';
```

```
data people;
input gender $ height weight;

datalines;
m 63 125
m 76 195
f 62 109
m 75 186
f 67 115
f 60 120
m 75 205
```

```
m 71 185
m 63 140
f 59 135
f 65 125
m 68 167
m 72 220
f 66 155
;

proc print data=people;
run;
```

The LOG window gives information on the execution of the program. If your program did not execute properly you should examine the log for error messages that may explain the failure. The program would then be modified if necessary and rerun.

SAS creates a new window called the **Results Viewer** when the program is executed and produces output. This window is in HTML format and a new tab for the window is created below the left-hand windows.

1.2.1 Sorting Data

The data can be sorted (in our case by gender) using PROC SORT by adding the following lines to the program. We can then go ahead and print our new dataset sorted by gender with the proc print step.

Remember: Any procedural step we do must begin with PROC and every line must end with a semicolon and the command run;

```
proc sort data = people;
by gender;
run;

proc print data=people;
title3 "Raw data sorted only by gender";
run;
```

Note: Any time we use PROC SORT our original dataset is sorted. SAS does not create a copy then sort!

1.3 Other Notes

1.4 Data Type

Question:

What is the type of each variable in the following dataset?

- AGE: The respondent's age in years
- GENDER: The respondent's sex coded 1 for male and 2 for female
- HAPPY: The respondent's general happiness, coded:1 for "Not too happy"2 for "Pretty happy"3 for "Very happy"
- TVHOURS: The average number of hours the respondent watched TV during a day

Answer:

Age: Continuous variable; SEX: qualitative; HAPPY: Discrete variable; TVHOURS: Continuous variable

2 Working with SAS Syntax

Objective

1. Identify the characteristics of SAS statements.
2. Explain SAS syntax rules.
3. Insert SAS comments using two methods.
4. Identify SAS syntax errors.
5. Diagnose and correct a program with errors.
6. Save the corrected program

In general, writing a SAS program is to write a sequence of steps, and a step is a sequence of SAS statements:

Exercise 2-1: How many statements are in the following step?

2.1 SAS Statments

SAS statements have these characteristics (Check the highlight context in the following examples):

- Usually begin with an identifying keyword.
- Always end with a semicolon.

```
data work.NewSalesEmps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile 'newemps.csv' dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps;
run;

proc means data=work.NewSalesEmps;
   class Job_Title;
   var Salary;
run;
```

SAS programming statements are easier to read if you begin DATA, PROC, and RUN statements in column one and indent the other statements. This makes it more structured. Also, consistent spacing also makes a SAS program easier to read.

Conventional Formatting

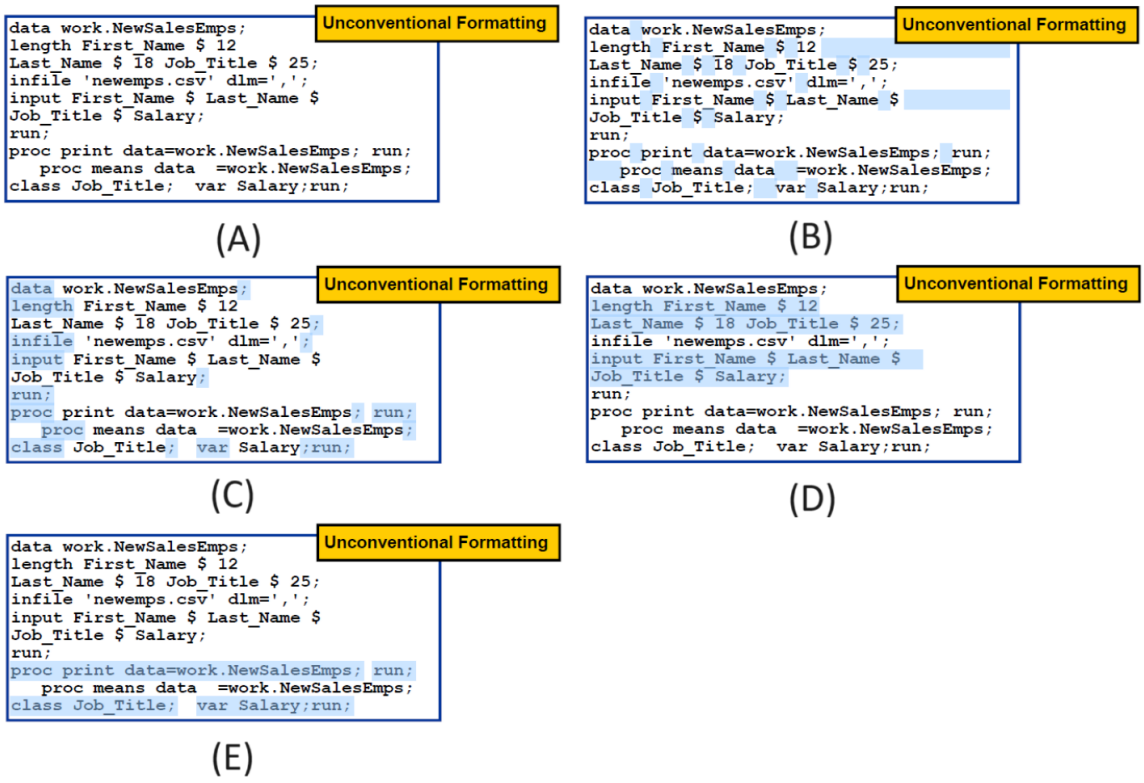
```
data work.NewSalesEmps;
   length First_Name $ 12
          Last_Name $ 18 Job_Title $ 25;
   infile 'newemps.csv' dlm=',';
   input First_Name $ Last_Name $
         Job_Title $ Salary;
run;

proc print data=work.NewSalesEmps;
run;

proc means data=work.NewSalesEmps;
   class Job_Title;
   var Salary;
run;
```

Overall, we can type SAS statements in the following ways:

- One or more blanks can be used to separate words.
- They can begin and end in any column.
- A single statement can span multiple lines.
- Several statements can be on the same line.



Sometimes we may want to make comments/notes to easy remember our SAS statements or to let other people easily understand what our code want to say. These comments are text that SAS ignores during processing. You can use comments anywhere in a SAS program to document the purpose of the program, explain segments of the program, or mark SAS code as non-executing text.

There are two ways to insert comments in a SAS program:

1. Using an asterisk (*) to begin the comment and a semicolon (;) to end the comment.

```
* This is a comment in SAS;
```

2. Using slash-asterisk (*/) to begin the comment and asterisk-slash (*/) to end the comment.

```
/* This is a comment in SAS */
```

Avoid placing the /* comment symbols in columns 1 and 2. On some operating environments, SAS might interpret these symbols as a request to end the SAS job or session. An example is given below:

2.2 Diagnosing and Correcting Syntax Errors

Syntax errors occur when program statements do not conform to the rules of the SAS language.

Examples of syntax errors:

- misspelled keywords
- unmatched quotation marks
- missing semicolons

3 Import and Export Dataset

4 Random Variables

Part II

Statistical Analysis

5 Introduction to Statistical Inference – part I

6 Introduction to Statistical Inference – part II

7 One Sample Nonparametric Test

8 One Sample Proportion Test

9 Writing SAS Macro Program: Using One Sample Variance Test/CI as Example

10 χ^2 Goodness of Fit Test

References