An Introduction to SAS

To open the SAS software using University of Iowa Virtual Desktop...

1. Go to *https://virtualdesktop.uiowa.edu*
2. Log on with your hawkid and password
3. Choose **SAS 9_4 64bit**.

IMPORTANT NOTES:

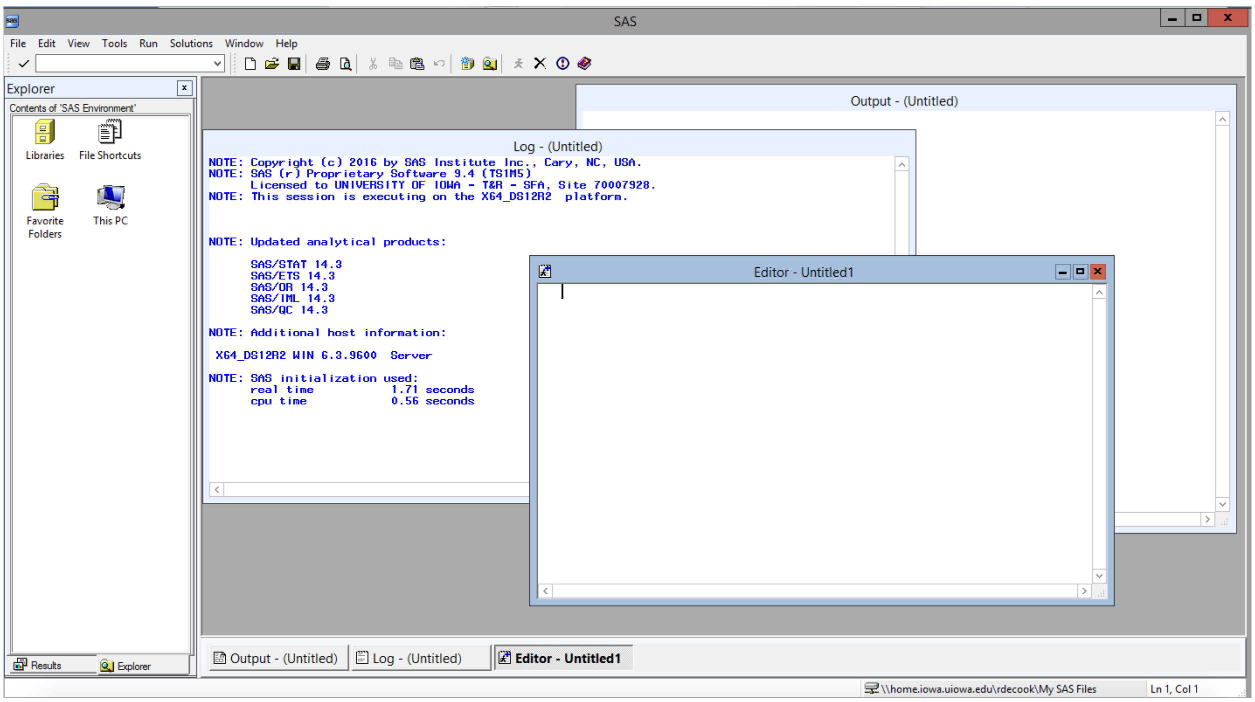
- Personal-laptop access - If you are using a personal laptop, you will need to have the Citrix Workspace App installed to run SAS through virtual desktop.
- Off-campus access - If you want to access SAS from off-campus, you will need to use Virtual Private Network (VPN).
- Files in your university home drive folder (H : drive) are accessible through virtual desktop.

Contents

1 Launching SAS

After launching SAS, four windows will appear:

- The *Editor* window is where syntax and commands are kept (the script). Saved as a .sas file.
- The *Log* box gives you run information after running programs and provides error statements.
- The *Output* window is where listing (i.e. text) results will appear. Saved as a .lst file.
- The *Explorer* gives you you quick folder access to files (you can close this window if you like).



2 Introduction to SAS

SAS is a command-driven statistical package; you enter statements in SAS's language, submit the statements, and then get results output. A fairly friendly user interface is provided to help with this. You enter your SAS statements in an Editor window, then click on a Submit button (a running-man icon at the top of the GUI), and the results appear in an Output window (simple text listing output), and/or HTML output in the Results Viewer window. If your code uses `PROC GPLOT or SGPLOT` you will also generate a GRAPH window. There is also a Log window that displays messages, and an explorer window that allows you to navigate the results and your datasets.

As you read this handout, I suggest that you start up SAS, and copy/paste these statements into SAS's editor and submit them. If you highlight a portion of a SAS program in the Editor window and click Submit (i.e. click on the running-man), only the highlighted portion is submitted to SAS. Otherwise, everything in the Editor window is submitted.

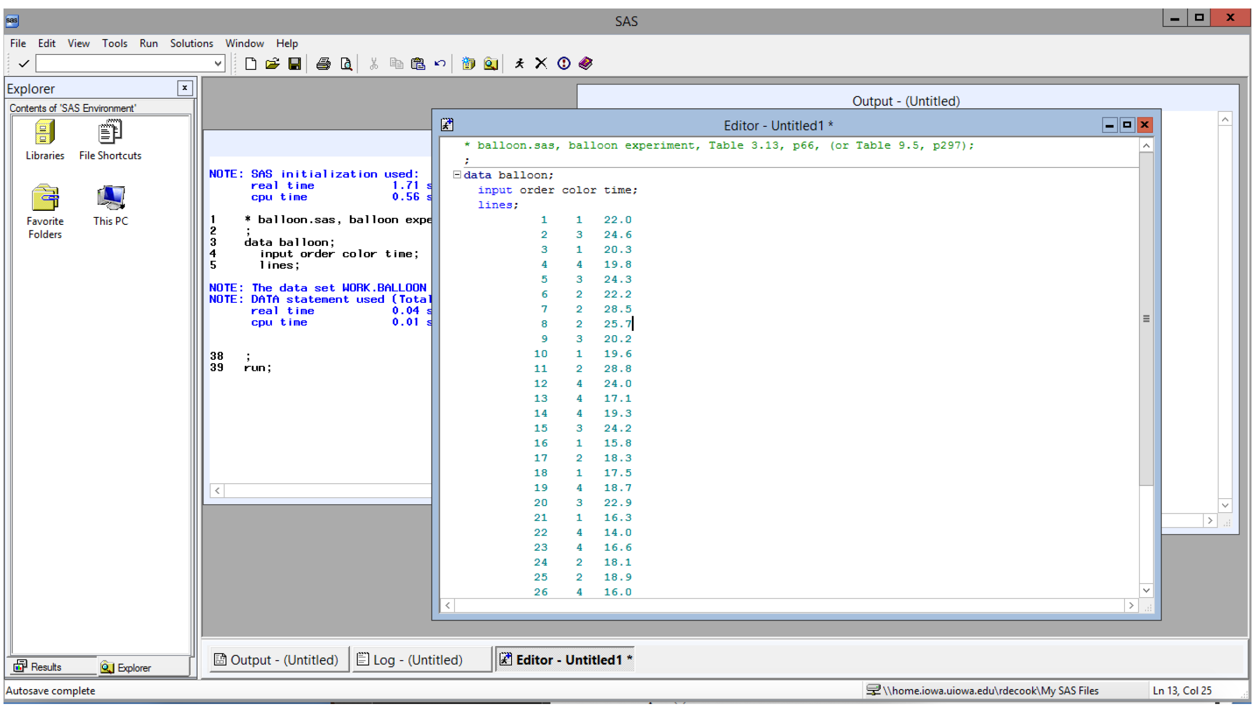
3 Basic SAS Programming Structure

SAS programs consist of three types of building blocks: `options` statements (options are global instructions that affect the entire SAS session and control the way SAS performs operations), `data` steps (for creating data sets, or manipulating datasets), and `proc` steps (for doing analyses). This handout gives you a few basic examples. Notice the usage of the semicolon throughout the SAS code to end each statement.

4 Quick Examples

4.1 Dataset Creation

We will use a SAS dataset supplied by Dean, Voss, and Draguljic for a quick example here. Go to the author's website and select the `SAS_data` link. Click on the `balloon.sas` file. This text represents a .sas file, or a file that is opened in the *editor* window. You could save this file from the web and then open it in SAS, or you can simply copy and paste into the editor window. You should have something similar to what is shown below after copying and pasting:



After you've copied and pasted the code into the *editor*, run the code (first click on the *editor* window to make it the active window, then click the running-man icon) which simply creates a dataset called `balloon` containing three numeric variables called `order`, `color`, and `time`, using the *lines* statement. If you click on the *log* box, you'll see that SAS tells you it created a dataset called `balloon` with 32 observations and 3 variables.

Save this file as *balloon.sas* for later use.

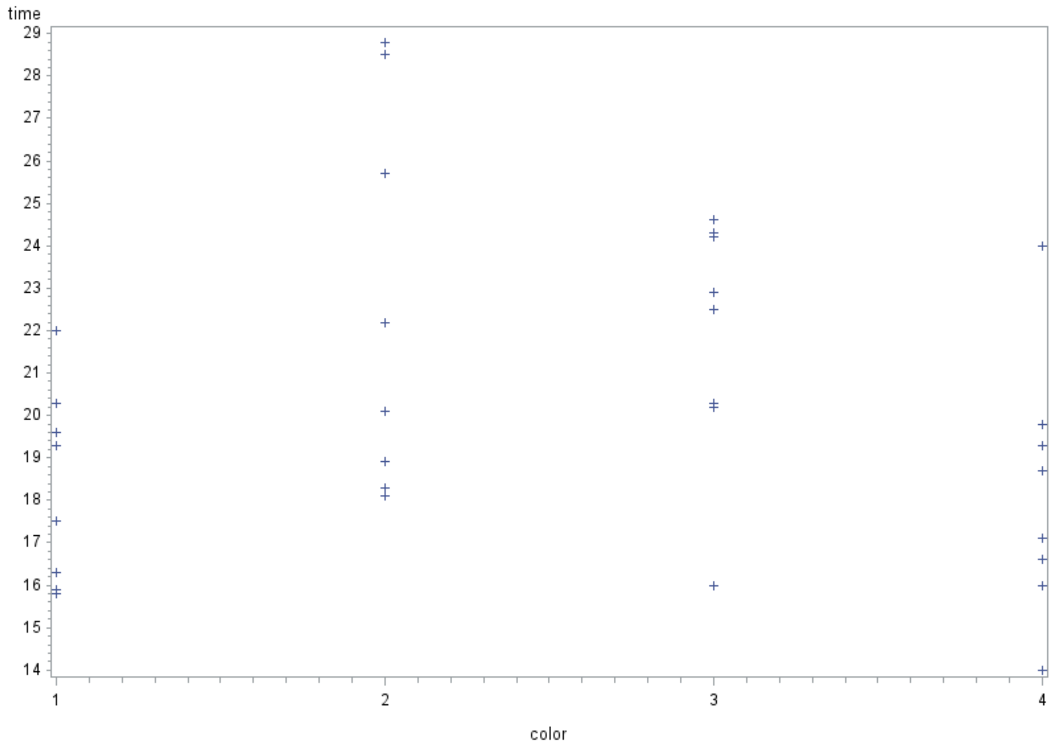
4.2 Plot of Y vs. X

Add the following three lines of code to your *balloon.sas* file (at the bottom). Run the code to get a plot of *time* against *color* for the *balloon* dataset. Here, we are using the *proc gplot* statement to create our plot. We state our X (color) and Y (time) axis variables in the *plot* statement.

SAS statement(s)

```
proc gplot data=balloon;
plot time*color;
run;
```

Graph1 Window



Graphics can be exported and saved in a variety of formats, otherwise, you can simply take a screen shot and save as a .png for later use.

4.3 Descriptive Statistics by a Factor

Get the *time* mean (and some other statistics) for each *color*. The dataset must first be sorted by *color*. Also, save the information in a new dataset called 'mymeans'.

SAS statement(s)

```
proc sort data=balloon;
by color;
proc means data=balloon;
var time;
by color;
output out=mymeans;
run;
```

Results Viewer Window

Results Viewer - sashtml

The MEANS Procedure

color=1

Analysis Variable : time				
N	Mean	Std Dev	Minimum	Maximum
8	18.3375000	2.2996506	15.8000000	22.0000000

color=2

Analysis Variable : time				
N	Mean	Std Dev	Minimum	Maximum
8	22.5750000	4.4991269	18.1000000	28.8000000

color=3

Analysis Variable : time				
N	Mean	Std Dev	Minimum	Maximum
8	21.8750000	2.9266021	16.0000000	24.6000000

color=4

Analysis Variable : time				
N	Mean	Std Dev	Minimum	Maximum
8	18.1875000	3.0215594	14.0000000	24.0000000

Print the new dataset.

SAS statement(s)

```
proc print data=mymeans;
run;
```

Output (Listing) Window

```
The SAS System

Obs    color    _TYPE_    _FREQ_    _STAT_       time

  1      1         0         8       N         8.0000
  2      1         0         8       MIN      15.8000
  3      1         0         8       MAX      22.0000
  4      1         0         8       MEAN     18.3375
  5      1         0         8       STD       2.2997
  6      2         0         8       N         8.0000
  7      2         0         8       MIN      18.1000
  8      2         0         8       MAX      28.8000
  9      2         0         8       MEAN     22.5750
 10      2         0         8       STD       4.4991
 11      3         0         8       N         8.0000
 12      3         0         8       MIN      16.0000
 13      3         0         8       MAX      24.6000
 14      3         0         8       MEAN     21.8750
 15      3         0         8       STD       2.9266
 16      4         0         8       N         8.0000
 17      4         0         8       MIN      14.0000
 18      4         0         8       MAX      24.0000
 19      4         0         8       MEAN     18.1875
 20      4         0         8       STD       3.0216
```

5 Setting Preferences for Output

By default, you will receive output in HTML format. HTML output is nice for graphics, but I've also found the HTML table output sometimes difficult to deal with (like when I'm trying to save pieces of it). Therefore, I also generate all my output as a 'listing' which appears in my Output window. This output is simply text and can easily be copied and pasted as desired. If you are on virtual desktop, you can choose to generate both HTML and listing output by going to...

```
Tools --> Options --> Preferences...
```

Then click the Results tab, and check the box that says 'Create Listing'. Then OK.

This listing output (Output window) is just text and you can easily copy and paste the pieces into Word or LaTeX using the 'verbatim' environment to present it. If you copy and paste into Word, you should **use a monospace font**, such as Andale Mono or SAS monospace to maintain column spacing. If you don't, you may find that you have very ugly results reporting like below (we can do better than this)...

Obs	run	pkg	count	logcount
1	1	Commerci	31000000	7.49136
2	2	CO2	250000	5.39794
3	3	MixedGas	24000000	7.38021
4	4	Vacuum	4100000	6.61278
5	5	MixedGas	26000000	7.41497
6	6	Commerci	67000000	7.82607
7	7	CO2	380000	5.57978
8	8	Vacuum	14000000	7.14613
9	9	Vacuum	19000000	7.27875
10	10	CO2	1100000	6.04139

If you save your listing output it will be as a .lst file.

6 Options

Assuming you want to copy/paste SAS listing output into a report in Word or LaTeX, for example, you should begin your SAS program with a statement like this:

SAS statement(s)

```
options ls=79 nocenter nodate nonumber
   formchar = "|----|+|---+=|-/\<>*" ;
```

This sets up a suitable maximum width, disables centering and page numbers, and specifies the characters to use as rules around tables and such.

7 Creating Datasets and Importing Data into SAS

The following illustrates using a data step to enter data into SAS. We also compute a new variable, logcount. Note that each SAS statement ends with a semicolon. Additional spacing or line breaks within a statement may be used as desired. A semicolon alone is a valid SAS statement, and such blank statements are used to signal the end of the input data. You can include comments as below using the pair of /**comment**/ format. When reading-in character variables with the `input` statement, you need to let SAS know this by following the variable name with a dollar sign \$. Without the dollar sign you'll get an 'invalid data' error in the Log window.

SAS statement(s)

```
data meat;
input run pkg $ count;
logcount = log10(count);
datalines;                      /* ‘cards’ or ‘lines’ will do the same.*/
1  Commercial   31000000
2         CO2     250000
3    MixedGas   24000000
4      Vacuum    4100000
5    MixedGas   26000000
6  Commercial   67000000
7         CO2     380000
8      Vacuum   14000000
9      Vacuum   19000000
10        CO2    1100000
11   MixedGas  100000000
12 Commercial  150000000
13   MixedGas   94000000
14     Vacuum   24000000
15        CO2    2600000
16 Commercial  380000000
17   MixedGas  340000000
18 Commercial  550000000
19     Vacuum   51000000
20        CO2    7200000
;
```

Log Window

```
NOTE: The data set WORK.MEAT has 20 observations and 4 variables.
NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds
```

As below, when we have the data as a separate file, we can use the `PROC IMPORT` statement to read it into SAS. Common `dbms` specifications available are CSV, EXCEL, TAB.

SAS statement(s)

```
proc import datafile="\\Client\H$\Iowa_classes\examples\sales.csv"
     out=sales
     dbms=CSV
     replace;
run;
```

8 Printing Dataset to Output Window

After the data are read into SAS, it's a good idea to print it on the screen to verify things. SAS won't actually run a proc until it sees another proc or data step, or a run statement. If you are running this as a tutorial, copy and paste the previous statements creating the data set called `meat` and then copy and paste the statement below into SAS's editor window. With the mouse, highlight just the proc print statement. Then click the Submit button. SAS will not complain, but you won't get any output because SAS thinks there may be more statements specifying what to do in proc print. Now highlight just the run; line, click Submit, and youll see the printout.

SAS statement(s)

```
proc print data=meat;
run;
```

Or if you just want to print the first 8 observations...

SAS statement(s)

```
proc print data=meat (obs=8);
run;
```

Output Window

```
Obs    run    pkg            count    logcount

  1      1    Commerci    31000000     7.49136
  2      2    CO2           250000     5.39794
  3      3    MixedGas    24000000     7.38021
  4      4    Vacuum       4100000     6.61278
  5      5    MixedGas    26000000     7.41497
  6      6    Commerci    67000000     7.82607
  7      7    CO2           380000     5.57978
  8      8    Vacuum      14000000     7.14613
```

9 Simple plots

`PROC PLOT` produces text-based graphics; there is also a `PROC GPLOT` that we saw earlier that makes higher-resolution plots. And since SAS 9.1, there's also `PROC SGPLOT`. But text plots work nicely for some simple purposes, especially diagnostic plots. Here is a simple way to get side-by-side dotplots of the data. Text plots always fill up a page, and since we want this one to be short and squat, we set the pages to be only 16 lines high using the options statement. You need a run statement before we set it back to a normal page size; otherwise, the re-sizing goes into effect before the graph is constructed.

SAS statement(s)

```
options ps=16;   /* Set short page size for character-based plot */;
proc plot data=meat;
   plot pkg * logcount;
run;
options ps=60;   /* Go back to normal page size */
```

Results Viewer window

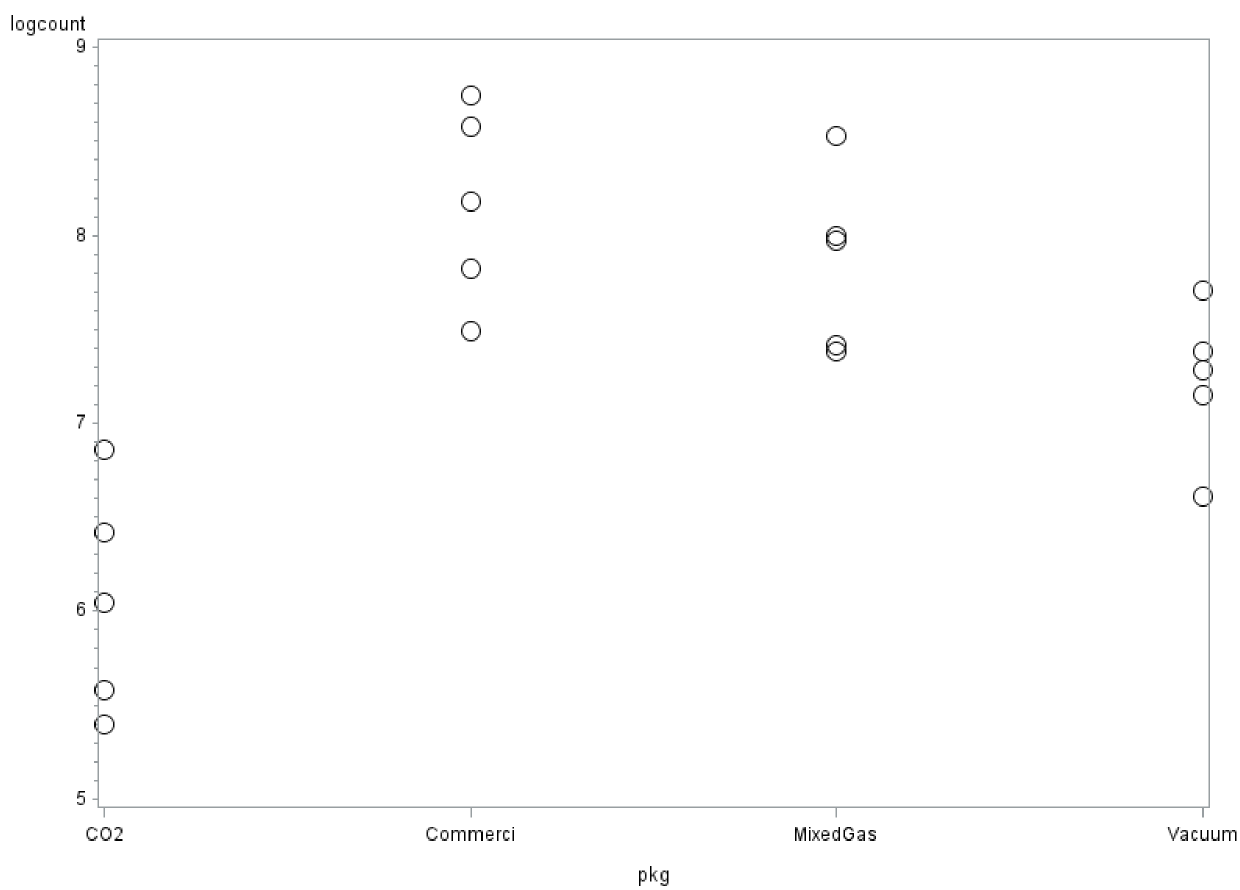
```
       Plot of pkg*logcount.  Legend: A = 1 obs, B = 2 obs, etc.

     pkg |
Vacuum   +                         A      A AA    A
         |
MixedGas +                                   AA       B      A
         |
Commerci +                                     A    A   A     A A
         |
CO2      +        A A      A    A     A
         |
         ---+-------------+-------------+-------------+-------------+--
            5             6             7             8             9
                                    logcount
```

Or using a higher resolution procedure as we saw earlier...

SAS statement(s)

```
SYMBOL1 i=none value=circle c=black height=2;
proc gplot;
   plot logcount*pkg;
run;
```

Results Viewer window



10 Other SAS Resources

- The on-line documentation for SAS is rally vast. If you google a particular procedure or model with 'SAS' in your search, you'll usually find some SAS documentation for it. For example, we'll use `PROG GLM` often in the start of the class. Here is a screen shot of the top of the help page on syntax...

The GLM Procedure

Overview | Getting Started | Syntax | Details | Examples | References

Syntax: GLM Procedure

The following statements are available in the GLM procedure:

```
PROC GLM <options> ;
    CLASS variable <(REF= option)> ...<variable <(REF= option)>> </ global-options> ;
    MODEL dependent-variables = independent-effects </ options> ;
    ABSORB variables ;
    BY variables ;
    CODE <options> ;
    FREQ variable ;
    ID variables ;
    WEIGHT variable ;
    CONTRAST 'label' effect values <...effect values> </ options> ;
    ESTIMATE 'label' effect values <...effect values> </ options> ;
    LSMEANS effects </ options> ;
    MANOVA <test-options> </ detail-options> ;
    MEANS effects </ options> ;
    OUTPUT <OUT=SAS-data-set> keyword=names <...keyword=names> </ option> ;
    RANDOM effects </ options> ;
    REPEATED factor-specification </ options> ;
    STORE <OUT=>item-store-name </ LABEL='label'> ;
    TEST <H=effects> E=effect </ options> ;
```

- The UCLA Institute for Digital Research & Education (IDRE) has some useful SAS tutorial information on-line at 'SAS learning Modules'.

UCLA Institute for Digital Research & Education
Statistical Consulting

Search this website

HOME | SOFTWARE | RESOURCES | SERVICES | ABOUT US

SAS LEARNING MODULES

SAS Learning Modules

- **Fundamentals of Using SAS (part I)**
  - Introduction to SAS
  - Descriptive information and statistics
  - An overview of statistical tests in SAS
  - Exploring data with graphics
- **Fundamentals of Using SAS (part II)**
  - Using where with SAS procedures
  - Missing values in SAS
  - Common SAS options
  - Overview of SAS syntax of SAS procedures
  - Common error messages in SAS
- **Reading Raw Data into SAS**
  - Inputting raw data into SAS
  - Reading dates into SAS and using date variables
- **Basic Data Management in SAS**
  - Creating and recoding variables
  - Using SAS functions for making/recoding variables
  - Subsetting variables and observations
  - Labeling data, variables and values
  - Using Proc Sort and the BY statement
  - Making and using permanent SAS data files (version 8)