Session 1: General Overview of SAS
What is SAS?

• SAS is a programming language
  • Data management (Base SAS)
  • Statistical analysis (SAS/STAT)
  • Graphical procedures (SAS/Graph)
• Also includes many other elements that you can access at Duke

<table>
<thead>
<tr>
<th>SAS/Access</th>
<th>SAS/ETS</th>
<th>SAS/GIS</th>
<th>SAS OLAP</th>
<th>SAS Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS/AF</td>
<td>SAS/FSP</td>
<td>SAS/IML</td>
<td>SAS/OR</td>
<td></td>
</tr>
<tr>
<td>SAS/Connect</td>
<td>SAS/Genetics</td>
<td>SAS Integration</td>
<td>SAS/QC</td>
<td></td>
</tr>
</tbody>
</table>
How to get SAS at Duke

• If you are part of Duke, but not SSRI, contact OIT:
  https://oit.duke.edu/comp-print/software/

• If you are part of SSRI, request SAS using the IT ticketing system:
  https://ssri.duke.edu/ssri-computer-help
Opening SAS for the first time

• My first time…

• Hopefully, your first time…
Key Components of Session 1

• The SAS system
• Importing / Exporting data
• Creating and reading data
• Data components
• Variable Names
• SAS statement types
• Debugging Programs
SAS Programming System Components

• Program Editor

• Run

• Log

• Output

• Explorer
Program Editor

• Programs are written in this window.
• Create and write programs
• Edit programs already written
• Save programs
• Run programs
Running a Program

- Click on the running man icon.
- F8 key.
- Selecting submit command from the run menu.
- Output found in output window.
Log Window

• SAS messages are displayed in the log.
• The log is a tool for diagnosing problems with the program.
• Comments appear in blue
• Warnings appear in green
• Problems (typically) appear in red – and usually cause the program to quit running
Output Window

- Results from statistical analyses will be shown in the Output window.
- Save as a file or print.
- Results window.
Explorer Window

• Create and find SAS libraries
• Create and view SAS data files
• Open SAS data files and view contents
• Edit and delete programs, logs and data files
• View and delete output
2-minute statement on SAS Statements

- SAS statements include a string of SAS keywords, SAS names, and operators that give information or perform a task.

- Key statement types:
  - Global (e.g., LIBNAME, RUN)
  - DATA
  - PROC

- Each statement must end with a **semicolon**!
Importing / Exporting Data

• Two ways to import and export data to be discussed.
  – Using the SAS Import and Export Wizard.
  – Using StatTransfer to translate from one datafile program to another.

https://www.stattransfer.com/
Importing Example
IMPORTING Syntax

PROC IMPORT OUT= WORK.test
   DATAFILE= "C:\Users\schmid\Desktop\SASexample.xls"
   DBMS=EXCEL REPLACE;
   RANGE="example";
   GETNAMES=YES;
   MIXED=NO;
   USEDATE=YES;
   SCANTIME=YES;
RUN;
Importing / Exporting SAS Data

• Many different ways to import or export SAS data.
  – Using a DATA step to create a data file from raw data.
  – Using Dynamic Data Exchange (DDE) with Excel and Word.
Concerns about Importing / Exporting

• Make sure that your SAS version is compatible with Microsoft
  – 32-bit versus 64-bit processing
• Don’t assume that the data imports correct – check and test!
  – This includes variable names, labels and formats
Reading SAS data sets

• After the import is completed, SAS data sets are available to view, manipulate and analyze.

• Depending on the process used in the OUT= statement, data sets may be temporary – only to be used during the current session or permanent.
Types of SAS Data Sets

- **Temporary**
  - Only exist during the SAS session
  - One-part data set name (i.e. survey) or two-part name starting with WORK.

- **Permanent**
  - Can be saved to disk or to your hard drive
  - Available for use anytime
  - Two-part data set name (i.e. my.survey)
Best Practices – Data Sets

• Generally, all work should be done as a series of temporary datasets
• Important to save code but not all of the in-between SAS datasets created by the code
• However, significant changes need to be saved as permanent datasets – with versions, etc.
Permanent Data Sets - Libraries

- Permanent SAS data sets are saved in a pre-designated subdirectory or folder

- SAS refers to it as a LIBRARY

- Libraries can be specified through a graphic interface or by using the libname statement.
Permanent Data Sets – LIBNAME Statement

• LIBNAME Statement
  – Tells SAS where the permanent data set is to be stored or is already stored.
  – Syntax:
    ```
    LIBNAME mydata 'C:\temp';
    KEYWORD LIBREF
    DRIVE:\DIRECTORY
    ```
General Assumptions about Data

• Imagine a table or spreadsheet.
• SAS consists of:
  – Variables (columns)
  – Observations (rows)
• Two data types:
  – Character
  – Numeric
SAS data files and elements:
Our Example for this class

<table>
<thead>
<tr>
<th>ID</th>
<th>date</th>
<th>race</th>
<th>gender</th>
<th>age</th>
<th>senseek1</th>
<th>senseek2</th>
<th>senseek3</th>
<th>senseek4</th>
<th>momed</th>
<th>livewmom</th>
<th>livewdad</th>
</tr>
</thead>
<tbody>
<tr>
<td>1101</td>
<td>4/18/2002</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1102</td>
<td>4/23/2002</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1103</td>
<td>5/11/2002</td>
<td>2</td>
<td>0</td>
<td>12</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1104</td>
<td>5/14/2002</td>
<td>2</td>
<td>0</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1105</td>
<td>5/18/2002</td>
<td>2</td>
<td>1</td>
<td>12</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1106</td>
<td>5/22/2002</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1107</td>
<td>6/18/2001</td>
<td>4</td>
<td>0</td>
<td>13</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1108</td>
<td>5/27/2002</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1109</td>
<td>6/1/2002</td>
<td>2</td>
<td>1</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1110</td>
<td>6/6/2002</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1111</td>
<td>6/24/2001</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1112</td>
<td>6/8/2002</td>
<td>2</td>
<td>0</td>
<td>12</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1113</td>
<td>6/30/2001</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1114</td>
<td>6/12/2002</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1115</td>
<td>6/14/2002</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1116</td>
<td>6/18/2002</td>
<td>2</td>
<td>0</td>
<td>12</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>1117</td>
<td>7/8/2001</td>
<td>2</td>
<td>1</td>
<td>13</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1118</td>
<td>6/22/2002</td>
<td>3</td>
<td>1</td>
<td>12</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1119</td>
<td>6/27/2002</td>
<td>2</td>
<td>1</td>
<td>12</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1120</td>
<td>7/1/2002</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1121</td>
<td>7/4/2002</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Numeric Data

- Numeric data are numbers… which can include decimals or negative numbers.
- Dates and times are a special form of numeric data.
- Periods (.) are used to define missing data.
- Primary statistical analysis occurs using numeric data.
Character Data

- SAS defines character data as “a sequence of letters, numbers and/or special characters”
- You use “ “ to define the character variable.
- Manipulation of character data is possible and often necessary
- Some basic analyses can be conducted, but are limited
Converting Character to Numeric (and Numeric to Character)

• Step 1: Create a new variable
• Step 2: Use either PUT (num -> char) or INPUT (char -> num) to identify the correct new variable type.
• Step 3: Delete the original variable
• Step 4: Rename the new variable into the original variable name
Example - Converting

newrace = PUT(race, 1.0);
DROP race;
RENAME newrace = race;

newid = INPUT (id, $4);
DROP id;
RENAME newid = id;
Dates in SAS

• Internal to SAS, dates are stored as the numeric distance away from Jan 1, 1960.
  – Negative numbers are dates before
  – Positive numbers are dates after
• Formats can be used to transform internal dates to usable formats.
  – Date9. = 14 MAR 1978
  – Mmddyy10. = 03/14/1978
Dates in SAS

• Functions can be used to construct or deconstruct a date
  – Mdy (m,d,y) = takes the month, day, year and puts them altogether
  – Month (), day (), year () – takes the date and creates a month, day, and year variable

• Dates can be tricky – make sure that the formats and functions work the way that you want them to.
Missing Data

• Almost all data sets have incomplete data.
• Character data are blanks. (" ")
• Numeric data are periods. (.)
• Importance of missing data to analysis N.
Variable name rules (SAS)

• Names must be 32 characters or less.

• Must start with a letter or underscore ( _ ).

• Contain only letters, numbers and underscores.
Variable name rules (best practices)

- You want variable names to describe useful information about the data
- Conform not only to SAS but to other statistical software standards
- Can include letters, numbers, and underscores – ONLY!
- Case matters – so typically use all lowercase
- Can be up to 32 characters, but shorter is better
More about SAS programs

• Programs consists of SAS statements
• They are executed in the order they are entered
• Can be UPPER CASE or lower case
• Can continue on the next line or be on the same line as other statements
• Every SAS statements ends with a semicolon
SAS Building Blocks

• DATA step
  – Introduce and prepare data for use in SAS.

• PROC step
  – Issues procedures to actually perform analysis and generate output.
SAS Global Options

• PROC OPTIONS; RUN;

• Types of options
  – Log, output, and procedure options
  – Data set control options
  – Error handling options
  – Reading and writing data options
Run Statement

- Run statement tells SAS that the data step or procedure has ended.
- End each set of statements with a run statement.
- You must submit your programs in order for them to run.
Basic Components of SAS

• Every SAS program is constructed using the DATA step and/or PROC statements.
• Any combination of DATA steps and/or procedures may be used.
• Run statements should be used throughout program.
LIBNAME example "c:\sample";

DATA test;
SET example.sasexample;
newrace = PUT(race, 1.0);
   DROP race;
   RENAME newrace = race;

PROC FREQ;
TABLES race;
TITLE 'Check creation of new race variable';
RUN;

PROC CONTENTS;
TITLE 'Check creation of new race variable'
RUN;
Other Important Statements

• Comments
• Variable labels
• Formats
Comment Syntax

• Two styles of comments
  – One starts with an asterisk ( * ) and ends with a semicolon.
  – The other starts with a slash asterisk ( /* ) and ends with an asterisk slash ( */ ).
Comments

• Explain the program using comments.
• Concepts to include:
  • Program Name
  • Date program written / revised
  • Input and output file names (if applicable)
  • Purpose of the program
Variable Labels

• By default, SAS uses variable names as labels.
• LABEL statement allows you to create your own.
• Syntax: LABEL <var name> = ‘Label description’
Variable Formats

• A format is an instruction that SAS uses to write data values.

• Formats can be used to control the appearance of data values or group values together for analysis.
Syntax for Variable Formats

• <$$> format <w>.<d> where:
  • $ : indicates a character format; its absence indicates a numeric format.
  • Format: names the format.
  • W: specifies the format width.
  • D: specifies an optional decimal
  • Formats ALWAYS contain a period (.)
LIBNAME format "C:\sgsg\sdjljd";

PROC FORMAT;
VALUE ss15
1 = 'Strongly disagree'
2 = 'Disagree'
3 = 'Neither Disagree or Agree'
4 = 'Agree'
5 = 'Strongly Agree';

DATA example;
SET format.example;

LABEL
ID = 'student ID'
Age = 'student age'
race = 'student race/ethnicity'
gender = 'student gender';

FORMAT gender best8. senseek1 senseek2 senseek3 senseek4 ss15.;

RUN;
Debugging Programs

• Reading and understanding the SAS log
• Debug your programs one step at a time
• Common errors:
  – Missing semicolon
  – Misspelled commands
  – Incorrect variable definitions
  – Using the DATA and PROC steps incorrectly or with the wrong options
Example: Problems with Programs

LIBNAME “c:\temp”;

DATA one;
SET schmid.data;
  newrace = PUT(race, 1.0);
  drop race;
  rename newrace = race;
RUN;

PROC FREQ;
  VAR race;
RUN;

PROC MEANS n mean std;
  VAR race;
RUN;
SAS Resources: Helpful Books and Websites
SAS Websites of Note

• SAS Online Documentation
  http://support.sas.com/documentation/

• UCLA SAS Portal
  http://www.ats.ucla.edu/stat/sas/
Recommended Books on SAS


