



Slide 1

SAS® -- A Rapid Introduction



Slide 2

Session Overview

Learning Objectives

At the conclusion of this presentation, participants should be able to:

- state a rationale for the selection of SAS as a research tool
- provide a visual understanding of how the SAS system can be applied to the research process
- demonstrate understanding and knowledge of basic SAS operations and procedures
- develop a step-by-step approach to conducting a SAS analysis

In this session, we will focus on the simplest and most direct methods for reading in data, performing data transformations and conducting simple analyses.



Slide 3

Session Overview

Presentation Outline

- Introduction to SAS
- SAS Role in Research Process
- SAS Basics
- Reading Data into SAS
- Preparing Data for SAS Analysis
- Modifying Data for SAS Analysis
- Using SAS
- SAS Utilities
- SAS Statistics
- SAS Graphics
- Making SAS Output Look Good
- Gaining SAS Proficiency



Slide 4

Introduction

What is SAS?

- SAS is the acronym for Statistical Analysis System.
- Conceived in 1966 by James Goodnight, a North Carolina State graduate student, as a way to implement ANOVA code he had developed
- Has become one of the most popular statistical analysis packages worldwide
- Provides flexible framework for integrating clinical, sample and experimental data



Slide 5

Introduction

What SAS Can Do

- ✓ Data collection and manipulation
- ✓ Data warehousing and management
- ✓ Data exploration and analysis
- ✓ Data presentation and graphics
- ✓ Software compatibility as "add-ons"
- ✓ System management and enterprise solutions

1:49 22 11



Slide 6

Introduction

Why SAS for Researchers

- ❖ Reliable
- ❖ Handles extremely large datasets
- ❖ Performs powerful operations and procedures
- ❖ Provides sample and power testing

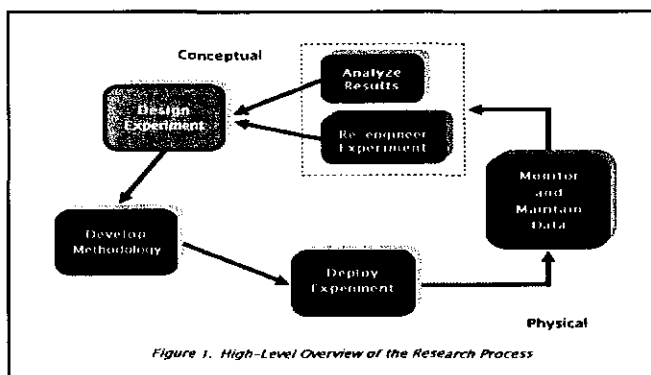
PL-1 - original language
"C" and Java - currently



Slide 7

Introduction

SAS Role in Research Process





Slide 8

Introduction

Availability of SAS Products

SAS Version 9.2 Products at the University of Iowa	
Base SAS	SAS/EIS
SAS Formats Library for Teradata*	SAS/ETS
SAS/ACCESS Interface to DB2	SAS/FSP
SAS/ACCESS Interface to HP Neoview*	↳ SAS/GRAPH
SAS/ACCESS Interface to MySQL	SAS/IML - <i>powerful language: engineering</i>
SAS/ACCESS Interface to Netezza	SAS/INSIGHT
SAS/ACCESS Interface to ODBC	SAS/LAB
SAS/ACCESS Interface to OLE DB	SAS/OR
SAS/ACCESS Interface to Oracle	SAS/QC
SAS/ACCESS Interface to PC Files	SAS/SECURE Windows
SAS/ACCESS Interface to Sybase*	SAS/SHARE
SAS/ACCESS Interface to Teradata*	✓ SAS/STAT
SAS/AF	SAS Formats Library for Teradata*
SAS/ASSIST	↳ SAS/GIS
SAS/CONNECT	○ SAS/Genetics

*Not Available on 64-bit version
**Not Available on Admin versions

genetics module



Slide 9

Introduction

SAS Access at UI

- SAS can be operated in a PC Windows, Macintosh, or Linux environment.
- Academic institutions are eligible to buy SAS at a discounted price.
- SAS is available for departmental or student purchase from the University Book Store (the student version is a locally edited edition limited to base SAS and a few course-related components).
- A complimentary SAS information kit is available from the SAS Institute at http://www.sas.com/ads/google_sasinfo/.
- SAS 9.1 and 9.2 are available on the Virtual Desktop in campus ITCs. (SAS 9.1 is also available on individual computers in the Statistics ITC in Schaeffer Hall and the Hardin Library ITC).



Slide 10

Introduction

Using the Virtual Desktop

- The Virtual Desktop is a web-based system which allows access to a number of software applications from virtually any computer with an Internet connection, on or off campus.
- A Citrix client or plug-in is installed on the local computer that, in effect, remote-controls the software application.

- Instructions for installing the Citrix client or plug-in on your personal computer can be found in the session handout (*see Appendix I*).
- To use SAS on the Virtual Desktop on Windows computers located in campus ITCs, you need only to log into the computer with your HawkID, click on **Start > All Programs > SAS** (the Citrix client or plug-in has already been installed).
- To use SAS on the Virtual Desktop on Windows computers located elsewhere on campus, you may need to install the Citrix client or plug-in. If at any point during the installation or after the client or plug-in has been launched you receive a message that gives you an opportunity to update the client or plug-in, you should select “upgrade later” as your option.



Slide 11

SAS Basics

Starting SAS

- Start SAS by clicking on the Start button, then click on **All Programs**, then select **SAS** and click on **SAS 9.1** or **SAS 9.2**, dependent on which version of SAS you have installed or access rights

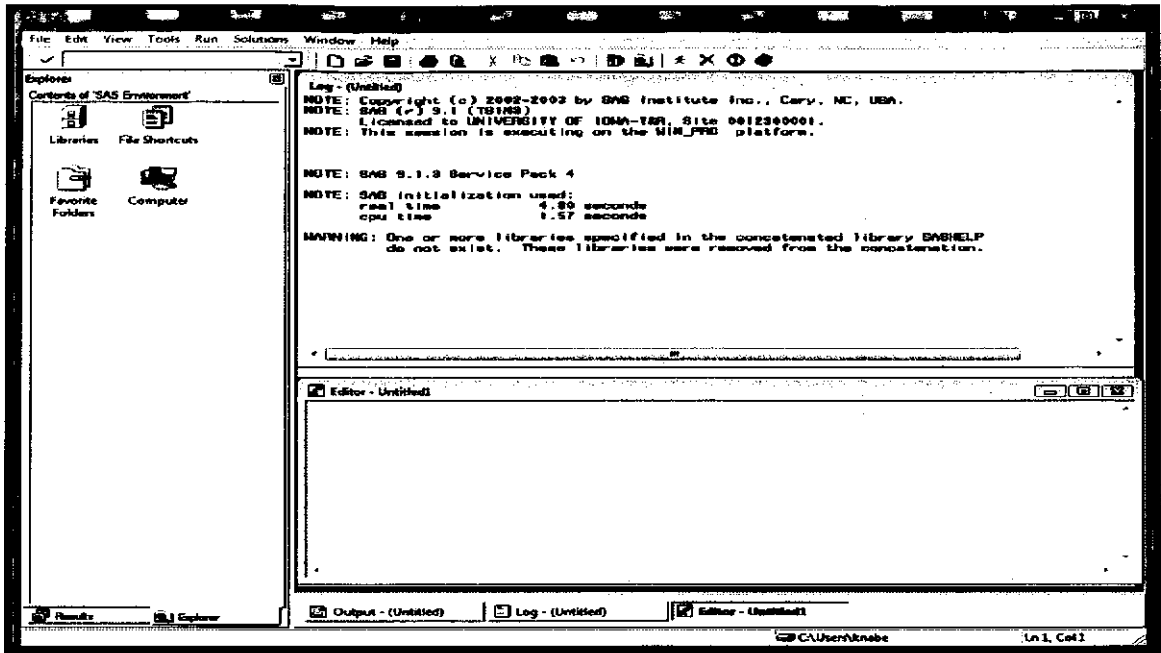


Slide 12

SAS Basics

Understanding the Screens

- SAS will open and automatically make available three windows: the enhanced Editor, the Log, and the Output.
- On the left hand side of the screen there is also a window labeled Explorer; this window can be helpful with file management.



*also
output screen
results screen*

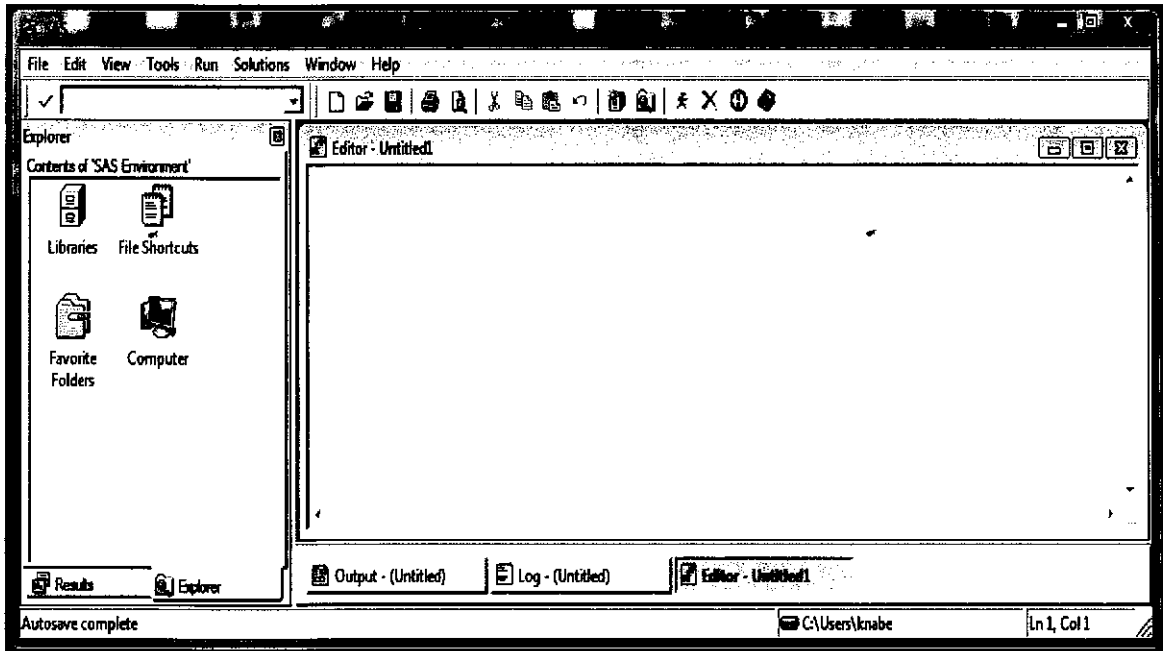


Slide 13

SAS Basics

The Enhanced Editor Window

- The Enhanced Editor Window is where you write the SAS programs you wish to run.
- A SAS program is a series of commands that will be read into SAS, manipulate data, perform the required analysis, and output the results.





Slide 14

SAS Basics

Enhanced Editor Features

- Once a program has been written and saved, the program name will appear at the top of the Editor window.
- The Editor works like a word processor; type the command line then press Enter to get a new line.
- The Enhanced Editor color codes your program for you as you type.

COLOR	COMMAND TYPE	EXAMPLE
BOLD BLUE	Major SAS commands	DATA
ROYAL BLUE	Sub commands, and recognized SAS words	INFILE STUDENT
PURPLE	Words within quotes such as filenames or titles.	'C:\My Documents\DATA.DAT'
BOLD GREEN	Numbers	1-20
GREEN	Commented out commands	*PLOT;
RED	Errors	TALBE
CALORIES	All user defined words such as variable names	CALORIES RESDAT1

SAS Enhanced Editor Color Codes

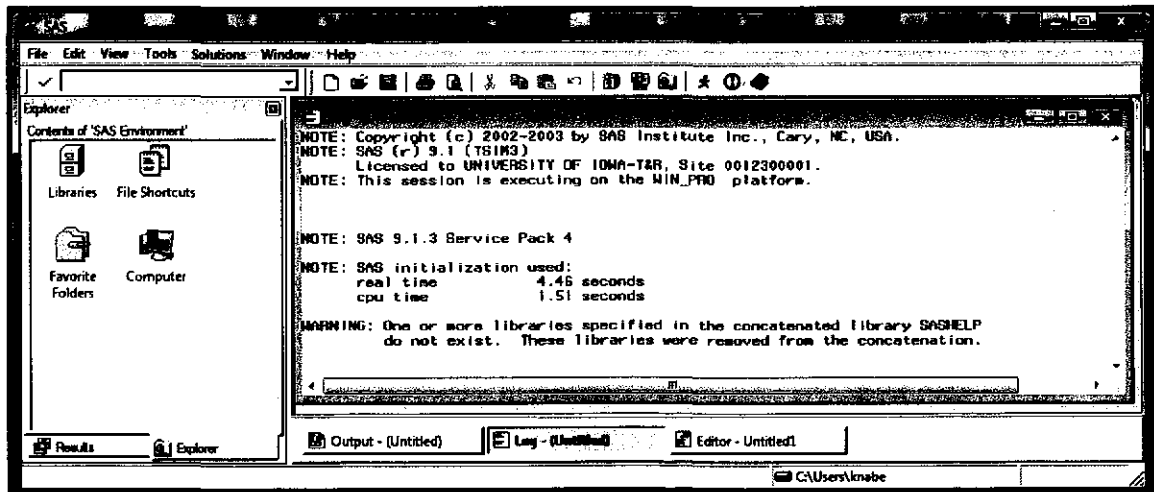


Slide 15

SAS Basics

The Log Window

- When you run a SAS program important information about the run is automatically displayed in the Log Window.
- The SAS log contains a list of program commands and operations, as well as errors that occurred during the program execution.
- It is always a good idea to inspect the log window to confirm that the program ran without errors.



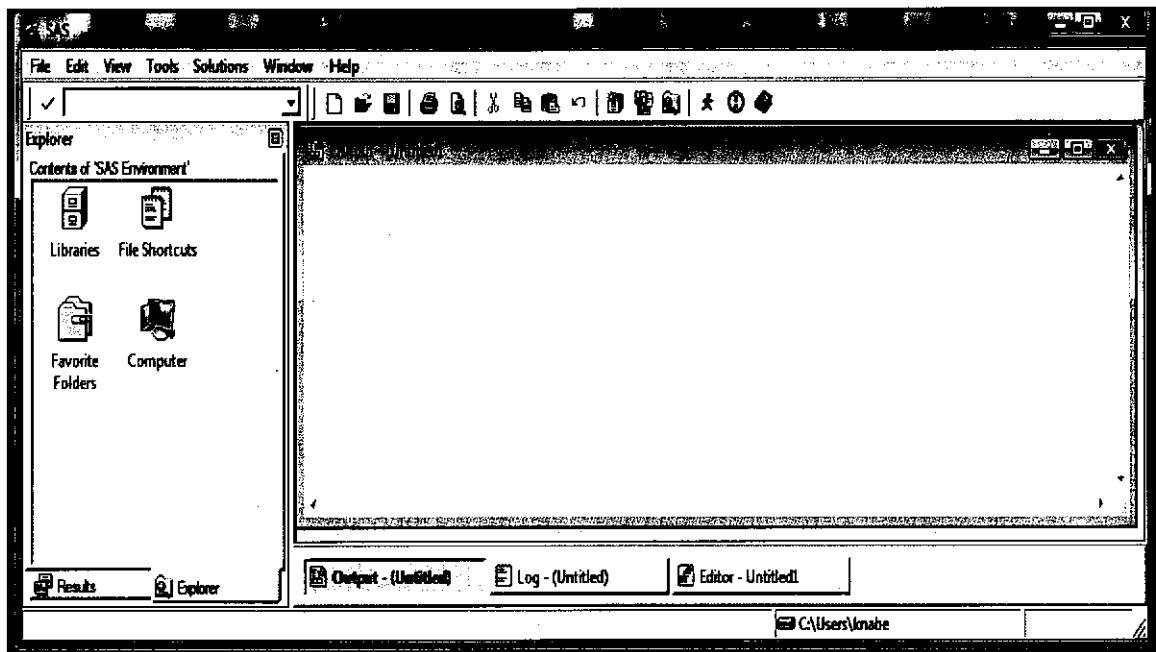


Slide 16

SAS Basics

The Output Window

- When the program executes without error, the results of the analyses are displayed in the Output Window.
- When the contents of the Output Window contains incorrect output, the contents of the Output Window or any other active window can be quickly cleared by placing the cursor in the window and selecting Ctrl-E.

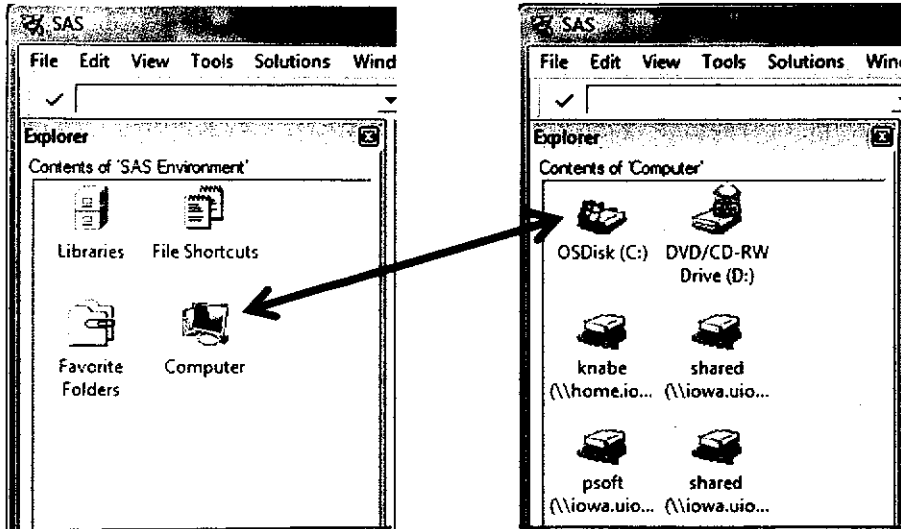


Slide 17

SAS Basics

The Explorer Window

- The Explorer window provides easy access to your saved SAS files and libraries.
- Click on libraries, then the work folder, and this will show any datasets you have read or created in SAS for the current session.
- Double-click on the Libraries icon and SAS will show all the libraries that are currently defined (Sashelp, Sasuser, Work) and any other library for a specific SAS product (e.g., SAS /Graph Maps) or that has been user-defined.
- The Computer icon gives you access to all shared drives /devices affiliated with your computer.

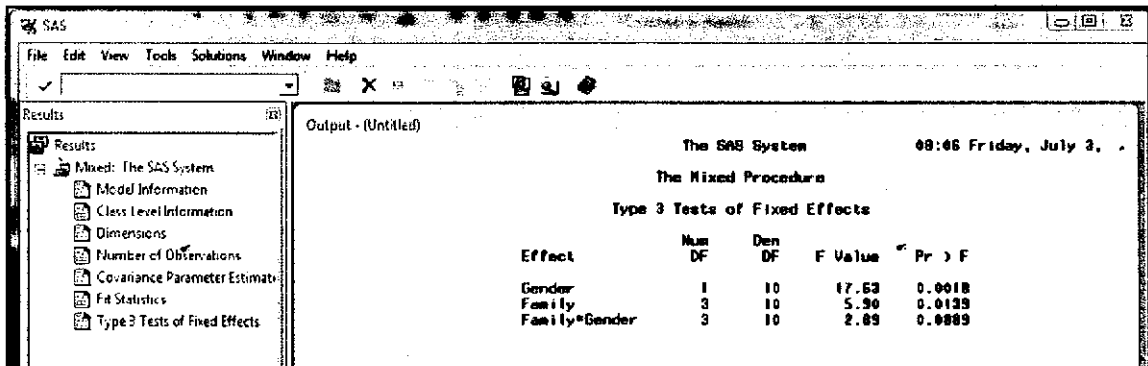


Slide 18

SAS Basics

The Results Window

- Provides a table of contents for your output
- Lists each procedure that produces output in outline form and can be expanded to show each part of the procedure output
- This window can be very helpful when you have a lot of output and you wish to view a particular section of the output.



Slide 19

SAS Basics

Becoming a SAS Programmer

- SAS is best as a "write code-then-run" program; it is not a very good "point-and-click" program.
- To become proficient in the use of SAS you must learn how to write a SAS program.
- Writing a SAS program can be simple if you understand what is required.



Slide 20

SAS Basics

Program Requirements

-- Every SAS program will have at least two parts:

DATA step -- reads the data from the raw data file where it has been stored into the program and carries out any variable manipulation(s) that have been requested.

PROC step -- performs the particular analyses that have been selected.

-- A SAS program may have more than one DATA step and more than one PROC step.

-- SAS does not work directly on the original data file; the PROC step uses data that are stored in a SAS (.sas) data set created by the DATA step.



Slide 21

SAS Basics

Anatomy of a Simple SAS Program

-- Examine the following simple SAS program code based on the infamous *Cyril.dat* file on twin IQ provided by Dr. Susan Graham at Harvard University:

```
data cyril;
  infile 'c:\my documents\SAS_examples\cyril.dat';
  input id fostiq owniq;
  label id      = 'Id number'
        fostiq = 'IQ of twin raised in foster family'
        owniq  = 'IQ of twin raised in own family';
run;

proc univariate plot;
  Title1 'Descriptive statistics for owniq and fostiq';
  var fostiq owniq;
  id id;
run;

proc plot;
  Title1 'Plot of fostiq vs owniq';
  plot fostiq * owniq;
run;
```



Slide 22

Reading Data into SAS

The SAS DATA Step

- The DATA step starts with the keyword DATA followed by the name that you wish to assign to a SAS data set.
- The DATA step has as its sole function to read and modify data.
- The DATA step can include DO loops, IF and IF-THEN/ELSE statements, and an assortment of numeric and character functions.
- DATA steps can also combine data sets by using concatenation and match-merge operations.
- A DATA step ends when SAS encounters a RUN statement or a new step (marked by a DATA or PROC statement).
- A typical SAS program starts with a DATA step to input or modify data and then passes the data to the PROC step which analyzes the data, performs utility functions, or prints reports; but that is certainly not the only pattern for mixing DATA and PROC steps.
- DATA steps execute line by line and observation by observation.

*SAS Data file
YOLR data
≡
Must have at
least one data step*



Slide 23

Reading Data into SAS

Raw Data Files *(start out with)*

- Most common raw data files are flat-files, where each row contains all the data for a particular case.
- Raw data files are usually organized in variables (columns) x cases (rows) format.
- There may be a header line that contains variable names that can be read or omitted.
- Raw data files may be internal or external to the SAS program.



Slide 24

Reading Data into SAS

Writing an INFILE Statement

- The INFILE statement gives SAS the name and physical location of the file that contains the data you wish to use.
- The physical location must begin with the root directory and include specification of each level [folder] required to reach the desired file.
- Current versions of SAS (9.1 and 9.2) can accommodate the long folder names often used to organize files.
- The INFILE statement follows the DATA statement and must precede the INPUT statement.
- In some operating environments, SAS assumes data files have a maximum record [number of characters, including spaces, in a data line] length of 256. If SAS is not reading all of your data, you can add the LRECL= option to the INFILE statement to specify a record length at least as long as the longest record in your data file.

infile single quote mark

```
INFILE 'c:\MyRawData\kunk.dat' LRECL = 2000;
```

-- Check the SAS log window to see if the record length is sufficient to read all your data.



Slide 25

Reading Data into SAS

Writing an INPUT Statement

- The INPUT statement specifies the location of each variable in the raw data set.
- The INPUT statement begins with the word INPUT followed by the names of the variables in the data set in the order in which they appear.
- The INPUT statement allows you to read data in one of four formats:

"free field"
Start here

<u>Format</u>	<u>Example</u>
List	INPUT Item \$ Calories Protein Fat Fat_PC;
Column	INPUT Item \$ 1-20 Protein 26-29 Fat 31-34;
Informat	INPUT Item \$20. +5 Protein. @31 Fat.;
Mixed	INPUT Item \$ 1-20 Calories Protein @31 Fat +1 Fat_PC.;

- The List format requires at least one space between each data value.
- The \$ indicates that the data value is character.
- The Column format does not require spaces between data values, missing values can be left blank, character data can have embedded spaces, and you can skip unwanted variables.
<<Preferred Format>>
- The Informat format name ends with a period (.) but it is not a requirement.



Slide 26

Reading Data into SAS

Using a CARDS or DATALINES Statement

- Another option is to type or paste your data set directly into the SAS editor.
- This works best when you have a small data set (25 lines or less).
- Often used when testing SAS program on a data subset of a large file.
- Use the CARDS or DATALINES statement to inform SAS that your data is internal rather than external.
- The DATALINES statement must be the last statement in the DATA Step.
- It is very important for the last semicolon to appear on the next line after all your data has been listed. If you forget it, your last observation could be deleted because SAS reads all characters as data until it encounters a semicolon (;).

```

/* Read internal data into SAS dataset junk */
DATA junk;
  INPUT Item $ Calories Protein Fat
        Fat_PC Sodium Cost Food;
  DATALINES;
Big Mac          563 26.0 8.25 53 1010 1.43 1
BK Whopper       670 27.0 9.50 51 975  1.60 1
Wendy's Double   560 41.0 8.25 54 575  2.05 1
;
RUN:

```



Slide 27

Reading Data into SAS

Importing Data from External Sources

- In SAS it is possible to import data from other sources such as Excel, Access, Word, etc.
- Data values can be space, comma, or tab delimited.
- SAS provides an *Import Wizard* that takes you step-by-step through the process.
- From any SAS window, you can open the file menu in the main menu bar and select IMPORT DATA.
- The imported data can be saved in a SAS Work library or a SAS permanent library.



Slide 28

Reading Data into SAS

Creating a Permanent SAS Data Set

- A SAS data set can be created as either temporary or permanent. A temporary SAS data set is one that exists only during the current session and is automatically erased by SAS when the session ends. A permanent SAS data set remains after the session ends and can be used again in subsequent sessions.
- If you plan to use a SAS data set more than once it is more efficient to save it as a permanent SAS data set.
- Permanent SAS data sets are saved in a library using a library reference (libref) location defined by a LIBNAME statement.
- The libref can change but the program must point to the same library and member name.

```

LIBNAME dietary 'c:\MySASLIB';
DATA dietary.fastfood;
  INFILE 'c:\MyRawData\junk.dat';
  INPUT Item $ 1-20 Fat 31-34 Fat_PC 36-37;
Run;

LIBNAME example 'c:\MySASLIB';
PROC PRINT DATA = example.fastfood;
  TITLE "Fat Content in Fast Foods";
Run;

```

*Library MySASLIB
Name of file dietary.fastfood*



Slide 29

Preparing Data for SAS Analysis

Handling Missing Data

- Whenever SAS encounters an invalid or blank value in a data field, the value is defined as missing.
- In SAS missing numeric data are represented by a single period (.) and missing character data are represented by blanks in the data field.
- When writing recoding statements in the DATA step, use a period to refer to missing numeric values. For example, if you want to recode missing values in the variable FAT to the value 99 you would write the following IF-THEN statement:

```
IF FAT=. THEN FAT=99;
```

- When you want to assign certain characters to represent special missing values for all numeric variables you can use the MISSING statement. The missing values can be any letter in the alphabet or an underscore. For example, the values 'a' and 'b' will be interpreted as special missing values for every numeric variable in the data set if you include the following statement:

```
MISSING a b;
```



Slide 30

Preparing Data for SAS Analysis

Merging Data Files

- SAS provides the ability to merge files horizontally (add variables) or vertically (add cases).
- Variables can be selected for inclusion (KEEP) or exclusion (DELETE).
- New variables can be constructed from existing variables.
- The most common merge in SAS is a match merge, which is used to match observations from one data set with observations from another data set.
- If the two data sets do not appear in the same exact order, you will need to identify one or more variables common to both and sort both data sets by the common variable(s) prior to using the MERGE statement. Then in a DATA statement, name the new SAS data set to hold the results and then a MERGE statement listing the data steps to be combined. Use a BY statement to indicate the common variable(s):

```
DATA newdataset;  
  MERGE dataset1 dataset2;  
  BY comvar;  
RUN;
```



Slide 31

Modifying Data for SAS Analysis

Transforming Variables

-- Transforming a variable to a different scale

- Identify a name for the transformed variable unique to the data file with which you are working
- Type a statement indicating how an existing variable is to be transformed

-- Consider the following examples taken from a nutrition study:

```

data junkfood;
  infile 'c:\windows\desktop\junkfood.dat';
  input item $ 1-20 calories 22-24 protein 26-29 fat 31-34
        fat_pc 36-37 sodium 39-42 cost 44-47 food 49;
  new_cost = (cost + 100);
  new_pro  = (protein/4);
  new_sod  = (sodium*2);
  new_fat  = (fat-2);
  loge_cal = log(calories);
  jkavg    = (protein+sodium+fat)/3;

run;

```



Slide 32

Modifying Data for SAS Analysis

Mathematical Operations

-- Mathematical Operations available in SAS are listed in the following table:

Function	Operator	Example
Addition	+	height + weight
Subtraction	-	height - weight
Multiplication	*	height * age
Division	/	weight / height
Power	** or ^	weight ** 2
Equal	= or EQ	weight = 120
Unequal	< > or NE	weight < > 120
Less than	< or LT	weight < 120 or weight LT 120
Less than or equal to	<= or LE	weight LE 120
Greater than	> or GT	weight GT 80
Greater than or equal to	>= or GE	weight GE 80



Slide 33

Modifying Data for SAS Analysis

Using SAS Functions

- The SAS system provides an extensive library of "built-in" functions with a catalog of over 400 functions for a variety of programming tasks.
- A function is recognized in a SAS statement by the use of a function name, followed immediately by a function argument(s), separated by commas, and enclosed in parentheses. For a function requiring two arguments, the syntax is as follows:

FUNCTIONNAME (*argument-1*, *argument-2*)

where *arguments* are constants, variables, expressions and/or other functions

- A few of the most commonly used functions are listed in the table below:

Function Name	Syntax	Definition
Numeric		
INT	INT(<i>arg</i>)	Returns the integer portion of argument
LOG	LOG(<i>arg</i>)	Natural logarithm
MAX	MAX(<i>arg, arg, ...</i>)	Largest non-missing value
MEAN	MEAN(<i>arg, arg, ...</i>)	Arithmetic mean of non-missing values
MIN	MIN(<i>arg, arg, ...</i>)	Smallest non-missing value
ROUND	ROUND(<i>arg, round-off unit</i>)	Round to nearest round-off unit
SUM	SUM(<i>arg, arg, ...</i>)	Sum of non-missing values
Character		
SUBSTR	SUBSTR(<i>arg, position, n</i>)	Extracts a substring from an argument
TRIM	TRIM(<i>arg</i>)	Removes trailing blanks from character expression
Date		
DAY	DAY(<i>date</i>)	Returns day of month from a SAS date value*
MONTH	MONTH(<i>date</i>)	Returns the month (1-12) from a SAS date value

* A SAS date value is the number of days since January 1, 1960



Slide 34

Modifying Data for SAS Analysis

Use of IF and IF-THEN/ELSE Statements

- If statements can be used to create data sets which include only specific observations.
- Sometimes you want your SAS program to perform an action only if a certain condition is true.
- An IF statement first evaluates the condition and then executes the statement only if the condition is true.
- IF and IF-THEN/ELSE statements should be placed in the DATA step after the INPUT statement.
- Consider the following example:

```

if age >18 then newvar=var1 - var2;
if gender=2 & earnings <10000 then elibible=1;
if black=1 or hisp=1 then nonwhite=1;
if educ>=12 thne hsdegree=1;
  else hsdegree=0;

```



Slide 35

Using SAS

The SAS Procedure Step

- Each SAS procedure (or PROC) has unique characteristics and elements but many are shared as well. Although statements and options vary from one PROC to another, the basic PROC structure is something like the following:

Data=
Title=
Footnote = red = women, blue = men
by: way to subgroup data
"by gender" - males vs female for
later

```

PROC _____ DATA=_____ ;
TITLE _____ ;
FOOTNOTE _____ ;
BY _____ ;
LABEL _____ ;
FORMAT _____ ;
RUN; <and/or> QUIT;

```

- SAS procedures begin with the keyword PROC which signals to the SAS system that a "canned" program is being launched.
- Once the name of a PROC is specified, you may then specify one or more options available within the PROC, and in any order.
- The DATA= option informs the SAS system what data set is to be used as input to the PROC. If omitted, SAS automatically defaults to the most recently created data set, which may not be the most recent one used.



Slide 36

Using SAS

PROC Syntax Commonalities

- There are syntax rules in effect when learning to write SAS procedure codes.
- All PROCs begin with a procedure name followed by procedure options; statements that define the PROC are followed by statement options.
- Variables are listed in the order in which they are executed.
- Dependent variables are listed first followed by independent variables which are sometimes preceded by a key word like BY or WITH.
- Interaction effects between variables can be requested using a vertical (|) symbol or an asterisk (*).

dependent criteria for selection
Male, female = dependent
request use "by" in "with"

- The CLASS or CLASSES statement always appears before the MODEL statement in the ANOVA or MANOVA procedures.
- If you omit the data = command in the PROC statement, SAS will use the last data set created in the Data step.



Slide 37

Using SAS

PROC Statements and Options

- PROC statements are commands nested within a procedure that tells SAS what operations to perform and in some cases allows you to make your analysis more specific.
- Some PROC statements are necessary while others are non-compulsory.
- Options are commands that further describe a statement and in some cases may also further describe a procedure.
- The SAS language has three basic types of options: system options, statement options, and data set options.
- System options (identified by OPTIONS statements) appear in the DATA step and usually stay in effect for the duration of the session. They are usually placed in the first line of the program so you can quickly see what options are in effect.
- Statement options appear in individual statements and influence how SAS runs a particular DATA or PROC step.
- Data set options affect only how SAS reads or writes an individual data set.



Slide 38

Using SAS

The VAR Statement

- In many SAS procedures a VAR statement is either required or needed, especially if you are working with a large data set with many variables.
- For example, if you are using PROC CORR and you only want SAS to provide correlation statistics on a few of the variables, you can specify those variables using a VAR statement:

```
PROC CORR DATA=Junk;  
  VAR CALORIES PROTEIN FAT;  
  TITLE "Junk Food Correlations";  
RUN;
```



Slide 39

Using SAS

The CLASS Statement

- The CLASS statement tells SAS you have a variable in your data set that is categorical (classifies data into groups or classes).
- The most common usage of the CLASS statement is in the UNIVARIATE, MEANS, ANOVA, and GLM procedures. It is required for the ANOVA and GLM procedures only if you have a categorical variable such as gender. For example, if you had data from a clinical experiment with 20 subjects where five subjects received Treatment 1, five received Treatment 2, etc, treatment would be considered a categorical variable and thus would appear in the class statement of the GLM procedure:

```
PROC GLM DATA=Clinical;
  CLASS treatment;
  MODEL resp=treatment;
RUN;
[where resp is the response for each of the 20 subjects]
```



Slide 40

Using SAS

The MODEL Statement

- The MODEL statement tells SAS which model you would like to use for the analysis of your data.
- In univariate and multivariate analyses the dependent or response variable(s) must appear on the left side of the equals (=) sign and the independent or predictor variables on the right side in the MODEL statement.
- The MODEL statement supports many options in the GLM and REG procedures.



Slide 41

Using SAS

The MEANS and LSMEANS Statements

- Often in an analysis when differences are found among groups, you want to see exactly where those differences occur. MEANS and LSMEANS statements can be used for that purpose.
- If you have no missing values in your data set or you have no covariates, your design is a balanced one and you can use the MEANS statement. If you have missing values or you do have covariates, there is an imbalance in your design and you must use the LSMEANS statement to obtain the proper means and comparisons.
- The MEANS and LSMEANS statements can be used in conjunction with a variety of options in the GLM and REG procedures.

*For model stmt: Left = Right
dependent independent*

*Equal # in ea. cell
use means*

*If sample size in cells is diff
use LS means*



Slide 42

Using SAS

The OUTPUT Statement

- The basic function of the OUTPUT statement is to create a new data set containing both the information in the old data set plus any new diagnostics or statistics produced by the procedure in which the OUTPUT statement appears.
- An OUTPUT statement can be included in a DATA step if you wish to make multiple data sets containing observations that meet different selection criteria. This can be accomplished by using IF-THEN or DO-loop processing. For example, if you create three data sets in your DATA statement:

```
DATA junk1 junk2 junk3;
```

you can select observations to be placed in all three data sets using an IF statement with no OUTPUT data set name :

```
IF Item = 'Big Mac' THEN OUTPUT;
```

or you can specify a data set name and the selected observations will appear in that file only:

```
IF Item = 'Big Mac' THEN OUTPUT junk1;
```



Slide 43

Using SAS

The ALPHA Option

- The ALPHA option allows you to set the confidence limits in some SAS procedures.
- The default confidence level is .05 or 95% confidence.
- The alpha = level tells SAS that you want all confidence intervals, significance testing, etc to be performed at that level. For example, if you want 90% confidence on a PROC MEANS statement, you would specify ALPHA=10 along with the CLM option:

```
PROC MEANS ALPHA= .10 CLM;
```

The CLM option will produce a two-sided confidence limits table for all the variables that are analyzed.



Slide 44

Using SAS

The NORMAL Option

- The NORMAL option produces tests of normality in the UNIVARIATE procedure.

- The normal option gives the Shapiro-Wilks test of normality. This is a useful test to perform if you suspect your data does not meet the requirements of a normal distribution.
- The syntax to produce this option is simply:

```
PROC UNIVARIATE DATA=junk normal;
```



Slide 45

Using SAS

The PLOT Option

- The PLOT option produces three plots of your data (stem-and-leaf, box, and normal probability) in the UNIVARIATE procedure.
- The PLOT option provides a way to visually explore your data and look for outliers or other data anomalies.



Slide 46

SAS UTILITIES

PROC SORT

Application:

Sorts the observations in a data set by a variable(s) in either ascending or descending order.

Syntax:

```
PROC SORT DATA=dataset1 OUT=dataset2 {options};  
  BY variables {options};  
RUN;
```

Discussion:

If sort by more than one variable, list all variables in the order desired on the BY statement. SAS will sort on the first variable in the list, then the next variable, and the next variable until the list is completed. Proc SORT enables one or more columns of data to be physically ordered in a SAS data set. One can even perform a two-way sort where the first variable is ordered in ascending order and the second variable is ordered in descending order.



Slide 47

SAS UTILITIES

Selecting Subgroups with a BY Statement

- The BY statement is required for the SORT procedure.
- After using the BY statement in PROC SORT you can use it in other procedures to select variables for analysis. For example, if you are interested in performing regression analyses of height on weight by gender you would first sort your data set by gender as follows:

```
PROC SORT DATA=yourdata;  
  BY gender;  
RUN;
```

then use the sorted data to obtain two separate regressions – one for males and one for females – as follows:

```
PROC REG DATA=yourdata;  
  MODEL weight=height;  
  BY gender;  
RUN;
```



Slide 48

SAS UTILITIES

PROC PRINT

Application:

View the observations in a SAS data set

Syntax:

```
PROC PRINT {options} DATA=filename{(OBS=100)};  
  {TITLE 'Title of Output';}  
  {BY variables;}  
  {ID variables;}  
  VAR variables;  
  {SUM variables;}  
  {SUMBY byvars;}  
  {PAGEBY byvars;}  
  {FORMAT variable numberfmt./$charfmt.;}  
RUN;
```

Discussion:

*PROC PRINT allows you to print data generated by another SAS procedure with more control over the output variables and layout.



Slide 49

SAS Statistics

PROC UNIVARIATE

Application:

Calculates descriptive statistics, particularly details on data distribution

Syntax:

```
PROC UNIVARIATE {options} DATA=filename;
  {BY variables};
  {WEIGHT variable;}
  VAR variables;
  {OUTPUT OUT=newfile statistics;}
RUN;
```

Discussion:

PROC PRINT produces output similar to PROC MEANS except it provides a larger number of descriptive statistics. As with PROC MEAN, analysis can be performed using a numeric or character categorical variable in a CLASS statement. This produces the descriptive statistics by subgroups.



Slide 50

SAS Statistics

PROC FREQ

Application:

Calculates frequency table for the values of a numeric or character variable and cross tabulation of two or more variables in a data set

Syntax:

```
PROC FREQ DATA=filename;
  {TITLE 'Title of Output' };}
  {BY variables;}
  {WEIGHT variable;}
  TABLES variable{*variable} {/ list NOPRINT OUT=newfile};
  {FORMAT variable numberfmt./$charfmt.;}
RUN;
```

Discussion:

In its simplest form PROC FREQ produces a one-way frequency table. To produce a cross tabulation table for two or more variables you will need to specify the variable names separated by an asterisk (*).



Slide 51

SAS Statistics

PROC MEANS

Application:

Calculates statistics (means, sums, std, min/max, etc.) for numeric variables in a group or class

Syntax:

```

proc means statistics (nway noprint) data=filename;
{title 'Title of Output';}
{class variables;}
{by variables;}
{weight variable;}
var variables;
{format variable numberfmt./Scharfmt.;}
{output out=newfile statistic1=var1 statistic2=var2;}
run;

```

Discussion:

In PROC MEANS the BY and CLASS statements generate separate statistics for the analysis variables. The key differences are the need to sort the data set before using the BY statement and the extra memory required for the CLASS statement. Because of this sort requirement, the CLASS statement is the easiest to use.



Slide 52

SAS Statistics

PROC TTEST

Application:

Performs t-tests for one sample, two sample, and paired observations

Syntax:

```

PROC TTEST {DATA=filename <options>;}
  {CLASS variables;}
  {PAIRED variables;}
  {BY variables;}
  VAR variables {/options} ;
  {FREQ variable;}
  {WEIGHT variable;}
RUN;

```

Discussion:

The one sample t-test compares the mean of the sample to a given number which you supply, the independent two sample t-test compares the differences between two independent groups, and the dependent or paired observations t-test compares the differences within a group or matched groups.



Slide 53

SAS Statistics

PROC CORR

Application:

Calculates correlation coefficients between variables and can also produce covariances

Syntax:

```
PROC CORR options DATA=filename;  
  {TITLE 'Title of Output';}  
  {BY variables;}  
  VAR variables;  
  {WITH variables;}  
  {OUTP/OUTS/OUTK/OUTH=newfile;}  
RUN;
```

Discussion:

PROC CORR computes the Pearson correlation coefficient and simple descriptive statistics for all pairs of variables listed in the VAR statement. It also computes a p-value for testing whether the true correlation is zero. The correlations are reported in matrix form.



Slide 54

SAS Statistics

PROC REG

Application:

Calculates a theoretical model for the relationship between independent and dependent variables

Syntax:

```
PROC REG {options} DATA=filename;  
  MODEL dependent = independents {/options};  
  {SELECTION=options}  
  {WEIGHT variable;}  
  {PLOT dependent*(independents);}  
RUN;
```

Discussion:

PROC REG can be used to perform simple linear regression, nonlinear regression, and multiple linear regression.



Slide 55

SAS Statistics

PROC ANOVA

Application:

Calculates one-way and n-way analysis of variance

Syntax:


```

PROC ANOVA DATA=filename;
  {CLASS variables;}
  MODEL dependent = independent {/options};
  {MEANS variables} {/options};
  {TITLE 'Title of Output';}
RUN;

```

Discussion:

PROC ANOVA requires the dependent comparison groups to be equal. PROC ANOVA is more efficient than GLM when the data is balanced and it will yield the same results.



Slide 56

SAS Statistics

PROC GLM

Application:

Calculates analysis of variance for all types of general linear models

Syntax:

```

PROC GLM DATA=filename;
  {CLASS variables;}
  MODEL dependents= independents {/options};
  /* required */
  ABSORB variable-list;
  BY variable-list;
  FREQ variable;
  ID variable-list;
  WEIGHT variable;
  CONTRAST 'label' effect values... {/options};
  ESTIMATE 'label' effect values... {/options};
  MANOVA H= effects E= effect M= equations...
  MNames= names PREFIX= name {/options};
  MEANS effects / options;
  LSMEANS effects / options;
  OUTPUT OUT= SAS-data-set keywords= names... ;
  RANDOM effects / options;
  REPEATED factorname levels (levelvalues)
    transformation<,...> {/options};
  TEST H= effects E= effect {/options};
RUN;
QUIT;

```

Discussion:

PROC GLM performs univariate and multivariate analysis of variance. PROC GLM must be used if the data are unbalanced (comparison groups have unequal sizes).



Slide 57

SAS Graphics

PROC PLOT

Application:

Plots the values of two variables on a set of coordinate axes (X and Y) and produces single or overlaid plots

Syntax:

```
PROC PLOT DATA=filename;  
  {TITLE 'Title of Output';}  
  {BYvariables};  
PLOT varY*varX{='+'} {varY2*varX3 ...} {/options};  
RUN;
```

Discussion:

PROC PLOT uses a PLOT statement to verify which variable is to be on the x-axis and which is to be on the y-axis. Multiple plots can be requested with one PLOT statement. Plots can appear on separate pages or overlaid on the same page.



Slide 58

SAS Graphics

PROC GPLOT

Application:

Plots the values of two or more variables on a set of coordinate axes (X and Y) and produces a variety of two-dimensional graphs including scatter, overlay, logarithmic, and bubble plots

Syntax:

```
PROC GPLOT {options} DATA=filename;  
  PLOT yvar * xvar {/options};  
RUN;  
QUIT;
```

Discussion:

PROC GPLOT has more options than PROC PLOT and will allow you to produce better looking and more sophisticated graph. GPLOT allows you to use global statements to modify the axes; add titles, footnotes, and notes to the plot; or modify the legend if one is generated by the plot. You can also use an annotate data set to enhance the plot. The basics to know about GPLOT are how to choose symbols and how to draw regression lines.



Slide 59

SAS Graphics

Working with SAS Graphs

- To build a graph you must first choose which SAS/GRAPH procedure meets your needs. After choosing the procedure, you must decide which options are needed. Finally, you must choose the proper global statements.
- The Annotate facility is one of the most powerful building blocks for enhancing your output. This facility enables you to add symbols, colors, labels, images, and other visual enhancements to your graphs.
- How you decide to deliver your graphs output influences its appearance. The SAS Output Delivery System (ODS) provides incredible flexibility in generating SAS Graphs output with a wide range of formatting options including generating HTML output with GIF, JAVA, JAVAIMG, ACTIVEX, ACTXIMG, and JAVAMETA devices.
- The best way to fully understand how to produce good-looking graphs is to examine SAS programs and output that other SAS users have developed. SAS/GRAPH programs are available for download at <http://support.sas.com/md/papers/>.



Slide 60

Making SAS Output Look Good

TITLE and FOOTNOTE Statements

- The TITLE statement adds a title at the top line of each page of the program output.
- FOOTNOTE works the same way as TITLE but prints at bottom of page.
- You can specify up to 10 title lines or footnotes in a SAS program by adding numbers to the keywords TITLE and FOOTNOTE.

```
TITLE3 "This is the third title line";  
FOOTNOTE3 "This is the third footnote";
```

- TITLE statements must be enclosed in single or double quotation marks (if title contains an apostrophe use double).
- TITLE and FOOTNOTE statements may be placed in or outside the PROC step (but since they both apply to procedure output it makes more sense to put them inside the procedure).
- Title and footnotes stay in effect until you replace them or cancel them using a null statement (TITLE; would cancel all current titles).



Slide 61

Making SAS Output Look Good

LABEL Statements

- LABEL statements may be used to identify or describe variables in more detail (each label up to 256 characters).

```
LABEL ITEM      = 'Name of Junk Food'  
      CALORIES = 'Calories per Item'  
      PROTEIN  = 'Protein in Grams';
```

- The LABEL statement is placed after the INPUT statement in the DATA step or may be placed inside a PROC step.
- When a LABEL statement is used in a DATA step the labels become part of the data set; but when used in a PROC step the labels stay in effect only for the duration of that step.
- Labels must be enclosed in single quotes but may be listed on multiple lines.
- Only the last variable label listed is followed by a semicolon (;).



Slide 62

Making SAS Output Look Good

FORMAT Statements

- LABEL statements may be used to label variables but Proc FORMAT must be used to label the values of variables.
- To assign value labels to variables you must use Proc FORMAT.
- In PROC FORMAT you create a format by using the VALUE statement.
- The VALUE statement names the format and label for each value of the variable.
- The format name can be anything but it normally ends in the three letters FMT.
- The format you have defined can be invoked in subsequent Proc statements but you must use a period (.) to identify the format and distinguish it from a variable name.



Slide 63

Gaining SAS Proficiency

Writing SAS Programs: General Guidelines

- Obtain a copy of the data file layout so that you know where the variables you need are located.
- Use the samples in SAS Help and Documentation section as models to work from; you can cut/paste program statements directly into the SAS Editor.
- Remember to check the Log Window to be sure your program ran successfully.



Slide 74

Gaining SAS Proficiency

Exporting SAS Files

- Exporting a SAS data set to Excel, Access, SPSS, or other software program is the opposite procedure of the import process.
- SAS provides an *Export Wizard* that takes you step-by-step through the process.
 - Step 1. Choose the library and member name for the data set that you want to export
 - Step 2. Choose the type of file you want to create
 - Step 3. Choose the location (directory path) where you want to save the exported data
 - Step 4. Choose whether you wish to save the programming statements that are generated by the *Export Wizard*
- If you choose not to use the *Wizard*, you can also create a SAS export file using PROC EXPORT statements.

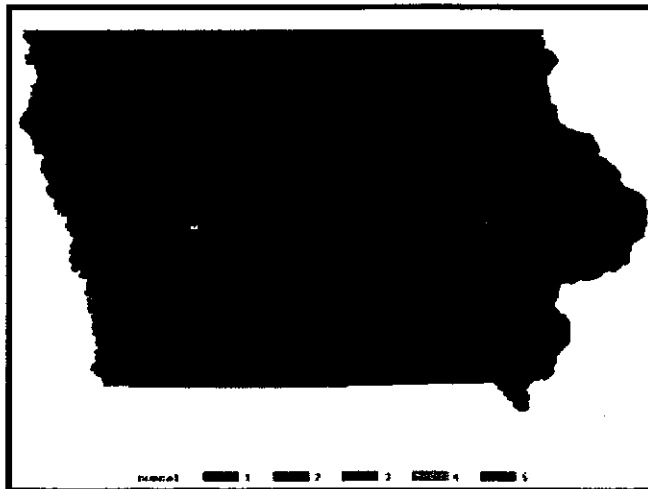


Slide 75

Gaining SAS Proficiency

Exporting SAS Graphs

- Often it is desirable to export SAS graphs to a Word or PowerPoint document.
- Graphs export best from PROC GPLOT, but it is also possible to export graphs from PROC PLOT; they just do not look as nice in the Word or PowerPoint documents.
- There are many different formats for saving graphs and many options for exporting graphs.
- Sometimes it takes a “trial and error” approach to find the best way to export a SAS graph.





Slide 76

Gaining SAS Proficiency

SAS Help/Other Resources

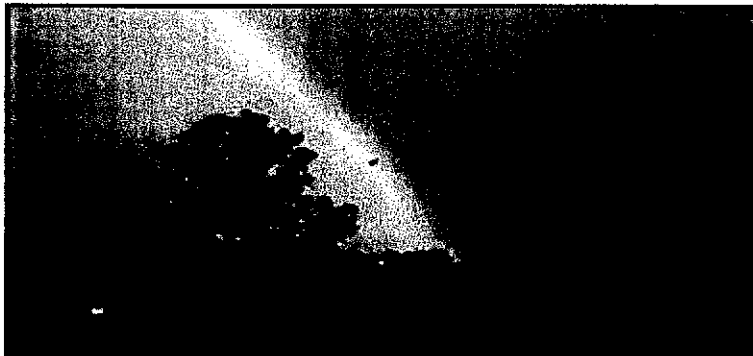
- Local SAS help is available from ITS-Statistical Data Applications at <http://helpdesk.its.uiowa.edu/>. If you are associated with the Colleges of Medicine, Dentistry, Nursing, or Pharmacy, you can also get assistance from the Biostatistics Consulting Center at <http://www.public-health.uiowa.edu/biostat/biocon.html>. Finally, if you are affiliated with the College of Education, you can also get assistance from the Statistical Outreach Center at <http://www.education.uiowa.edu/StatOutreach/>.
- SAS help is also available from the SAS Institute at <http://support.sas.com>.
- A really excellent source of information on SAS software is *The Little SAS Book: A Primer* by Lora D. Delwiche and Susan J. Slaughter (3rd or 4th Edition). Both editions are available online for limited preview at <http://books.google.com>. You can purchase new or used copies from Amazon at <http://www.amazon.com/books-used-books-textbooks/>.
- For a list of useful SAS Web links check out the Michael Davis discussion at [www.bassettconsulting.com/Useful SAS links PhilasUG.ppt](http://www.bassettconsulting.com/Useful_SAS_links_PhilasUG.ppt).
- For SAS coding tips/techniques go to <http://www.sconsig.com.sastip.htm>.
- For a LISTSERV based world-wide SAS discussion group that is always open for discussion check out SAS-L <http://listserv.uga.edu/archives/sas-l.html>.



Slide 77

Gaining SAS Proficiency

Discussion/Questions

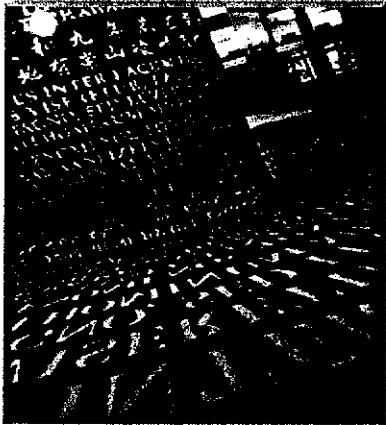




Slide 78

Gaining SAS Proficiency

Discussion/Questions



William E. Knabe
Statistical Data Analysis Support
Information Technology Services
2860-6 University Capitol Centre
The University of Iowa
Iowa City, IA 52242-5500
Telephone: 319-335-5448 (Office)
319-321-5503 (Mobile)
Email: william-knabe@uiowa.edu