STSCI 5010

Applied Statistical Computation with SAS

(Fall 2018)

Module 1

Introduction to SAS Programming

Why SAS?

What is SAS?

- Originally **S**tatistical **A**nalysis **S**oftware
- Conceived in 1966 by Anthony Barr
- In 1976, SAS Institute, Inc. was incorporated by Barr, Goodnight, Sall, and Helwig
- Is a collection of software products grouped and offered by SAS
- SAS is used in education, government, healthcare, business, financial companies, …
- SAS can be used with different operating systems (Windows, Unix/Linux and z/OS but not Mac OS)

What is SAS? (cont’d)

- SAS offers an array of tools to organize and analyze data (statistical discovery software)
- Among many functions you can perform:
  - Data management, data entry and retrieval
  - Statistical and mathematical analysis
  - Reporting
  - Business planning, forecasting and decision support
  - Data Mining
  - Graphing
  - Handling big data

Overview of the SAS products

- **BASE** SAS - data management and basic procedures
- SAS/**STAT** - statistical analysis
- SAS/**GRAPH** - presentation quality graphics
- SAS/**OR** - operations research
- SAS/**ETS** - econometrics and time series analysis
- SAS **Enterprise miner** – data mining
- SAS/**QC** - quality control
- SAS/**IML** - a matrix programming language

Many other specialized products are available …

How to learn?

- Read textbook/lecture notes carefully.
- Do quizzes at the end of each chapter.
- Practice, practice and practice…
- Some warnings.

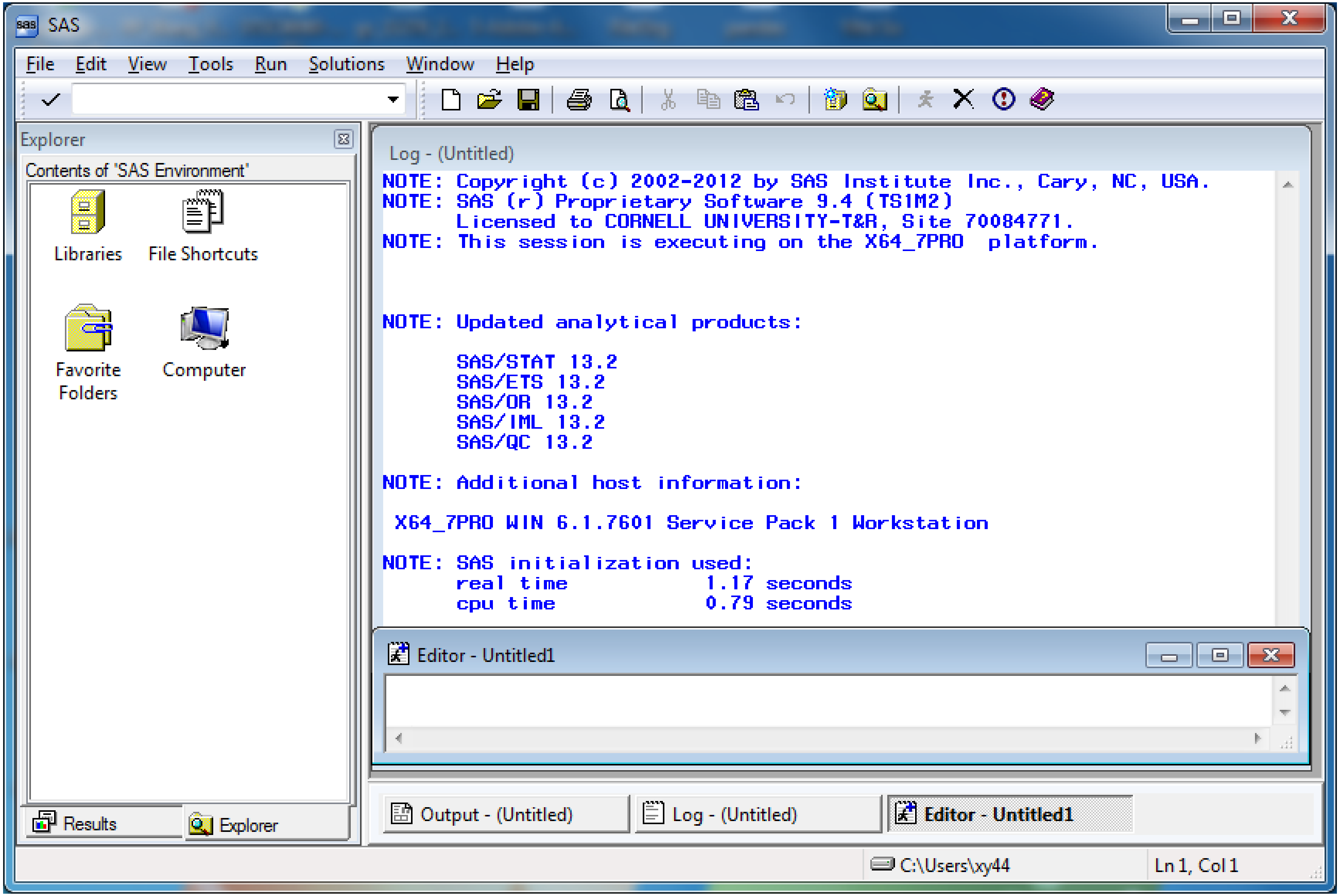
Chapter 1

Basic Concepts

Topics

- Introduction to the SAS Environment
- Overview of a SAS program
- Introduction to SAS data library
- Structure and components of SAS data sets

A SAS 9.4 Screen



```
NOTE: Copyright (c) 2002-2012 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software 9.4 (TS1M2)
      Licensed to CORNELL UNIVERSITY-T&R, Site 70084771.
NOTE: This session is executing on the X64_7PRO  platform.



NOTE: Updated analytical products:

      SAS/STAT 13.2
      SAS/ETS 13.2
      SAS/OR 13.2
      SAS/IML 13.2
      SAS/QC 13.2

NOTE: Additional host information:

 X64_7PRO WIN 6.1.7601 Service Pack 1 Workstation

NOTE: SAS initialization used:
      real time           1.17 seconds
      cpu time            0.79 seconds
```

SAS Windows

There are five main SAS windows:

1. **Explorer**: SAS libraries, file shortcuts
2. **Results**: a tree-like summary of your Output Window. This window is empty until you submit a SAS program that creates some output
3. **Program Editor**: enter, edit, submit SAS program lines (enhanced program editor)
4. **Log**: display notes and error messages
5. **Output**: output from SAS procedures

Other windows: Graph, Help

What is a SAS Program?

A SAS program is a **set of SAS statements** that tells the software what to do (i.e., access, manage, analyze, or present your data) and provides all the needed information to execute it.

The SAS statements are presented in a sequence and are executed in order. For example:

```
data students2;
    set students;
run;
proc print data= students2;
run;
```

SAS Statements

A SAS *statement* has two important characteristics:

- It usually **begins** with a **SAS keyword**.
- It always **ends** with a **semicolon.**

```
data students2;
     set students;
run;
proc print data= students2;
run;
```

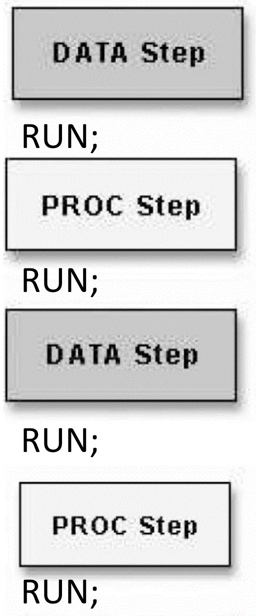
SAS Statements

- Statements **are NOT case sensitive**. Statements can be in upper or lower case. An exception is that text enclosed in quotation marks generally is case sensitive.
- Statements can continue on the next line (cannot split a word however).
- Several statements can be on the same line, separated by semicolon(s).
- Statements can start in any column.
- However, your SAS statements should be properly organized (for human readability).

The Parts of a SAS Program

1. DATA step: **DATA** new;
   weight=35;
2. PROC step: **PROC** print data=new;
3. A RUN statement: **RUN**;

These two types of steps, alone or combined, are often accompanied by a run statement, forming all SAS programs.



DATA Step

RUN;

PROC Step

RUN;

DATA Step

RUN;

PROC Step

RUN;

DATA Step

DATA steps typically **create** or **modify** SAS data sets to produce custom-designed reports or do data analysis.

DATA steps are used to:

- put your data into a SAS data set
- compute values
- check for and correct errors in your data
- produce new SAS data sets by subsetting, merging, and updating existing data sets, etc.

PROC Step

- **PROC (procedure)** steps invoke or call pre-written routines that enable you to analyze and process the data in a SAS data set.
- Starts with the PROC keyword followed by the name of the procedure.

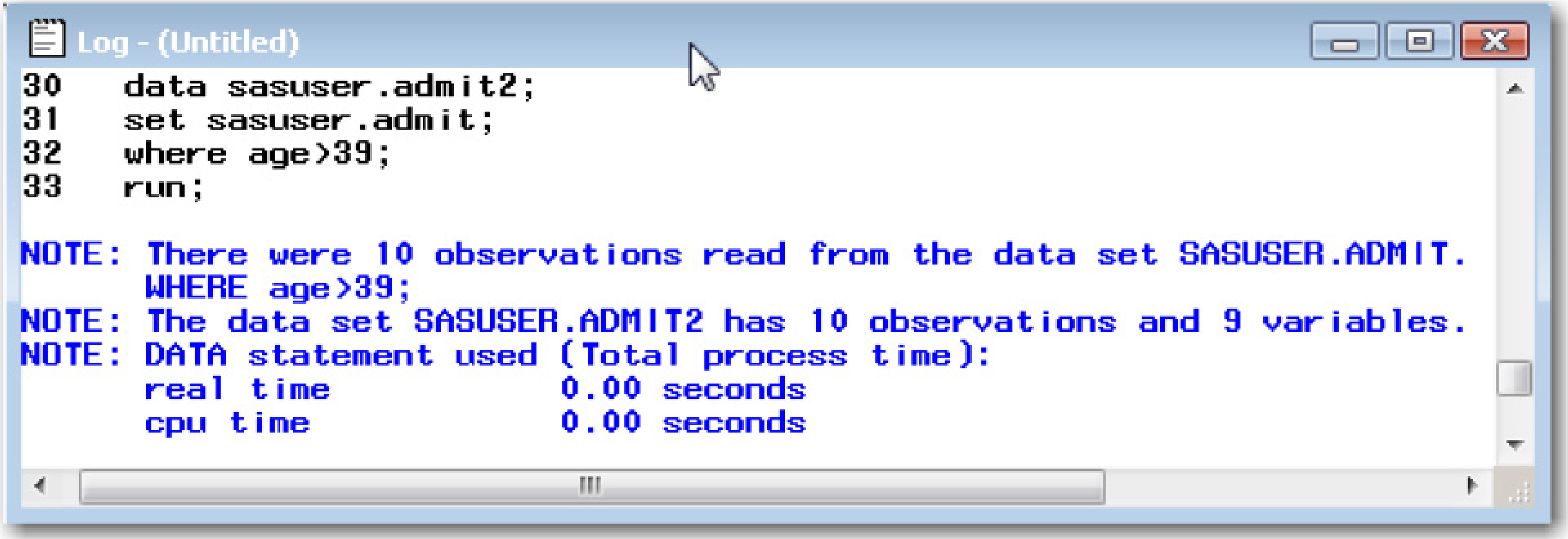
- PROC steps are used, for example, to:
  - print a report
  - produce descriptive statistics
  - create a tabular report
  - produce plots and charts
  - run a statistical analysis

The RUN Statement

- A **DATA** or **PROC** step ends when the program encounters a new **DATA** or **PROC** step, a **run** statement, or reaches the end of the program.
- Run statements are not part of a **DATA** or **PROC** step and indicate to run all the preceding statements.
- Run statements are not required between SAS steps but make the program and SAS logs easier to read.

SAS Logs

Each time a step is executed, SAS generates a log of the processing activities and the results of the processing, as well as any errors if they have occurred.



```
30    data sasuser.admit2;
31    set sasuser.admit;
32    where age>39;
33    run;

NOTE: There were 10 observations read from the data set SASUSER.ADMIT.
      WHERE age>39;
NOTE: The data set SASUSER.ADMIT2 has 10 observations and 9 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

It is important to check the SAS log after you have run a SAS program. Make sure there is no error message.

SAS Library

- In the Windows and UNIX environments, a SAS library is a **location** where SAS files are stored (e.g., a **folder** on your computer drive)

- All SAS data files are stored in **SAS libraries** so that SAS knows the location(s) of your files.

Automatic SAS Libraries

SAS assigns three libraries automatically each time you start SAS:

1. **SASHELP:** a permanent library that contains sample data
2. **SASUSER**: a permanent library that contains SAS files in your personal settings
3. **WORK**: a temporary library for files that do not need to be saved from session to session.

Often you create additional libraries to store your files.

Two types of SAS files

1) Temporary files:

- are deleted when you end your SAS session.
- are stored in the work library and are referred to as **work.filename** or simply by the **filename**.

2) Permanent files:

- stay on your computer after you end your SAS session.
- are always referred to with a 2-part name:

  libref.filename

Assigning a SAS Library

To create a library: assign it a name and tell SAS what location it corresponds to.

There are two ways to assign a SAS library:

1. Use the *LIBNAME* Statement

2. Use the *New Library* option in the Explorer window

How to Assign a SAS Library?

1. Use the **LIBNAME statement,** inside SAS programs, for example:

```
libname MPS "c:\MPS";
```

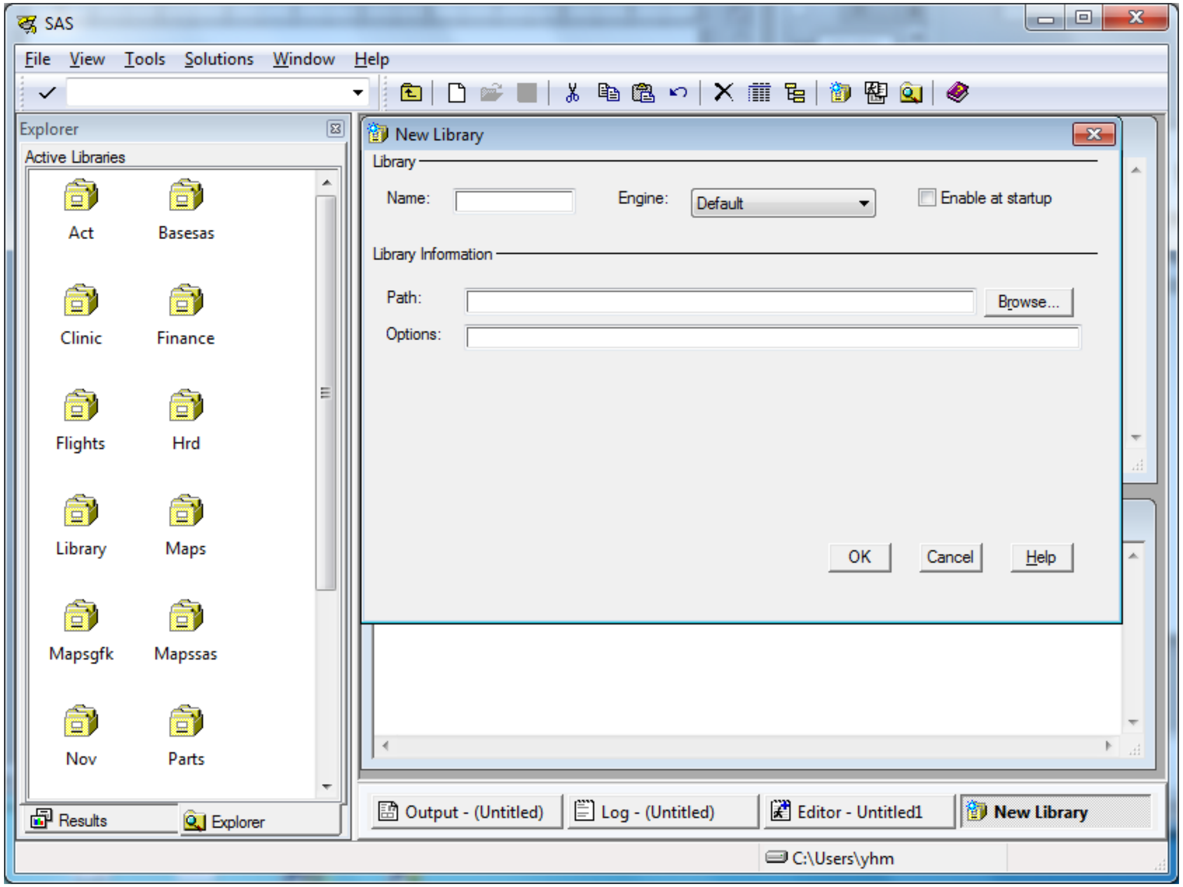
will associate the libref “MPS” with the library **"c:\MPS";**

Naming rule: A **libref** can

- be 1 to 8 character long
- begin with a letter or underscore
- contain only letters, numbers, or underscores.

How to Assign a SAS Library?

2. Use the **New Library** option by **right clicking** anywhere in the Explorer window, and then click on the New library icon. The New Library window opens. Follow steps.



Name:

Engine: Default

□ Enable at startup

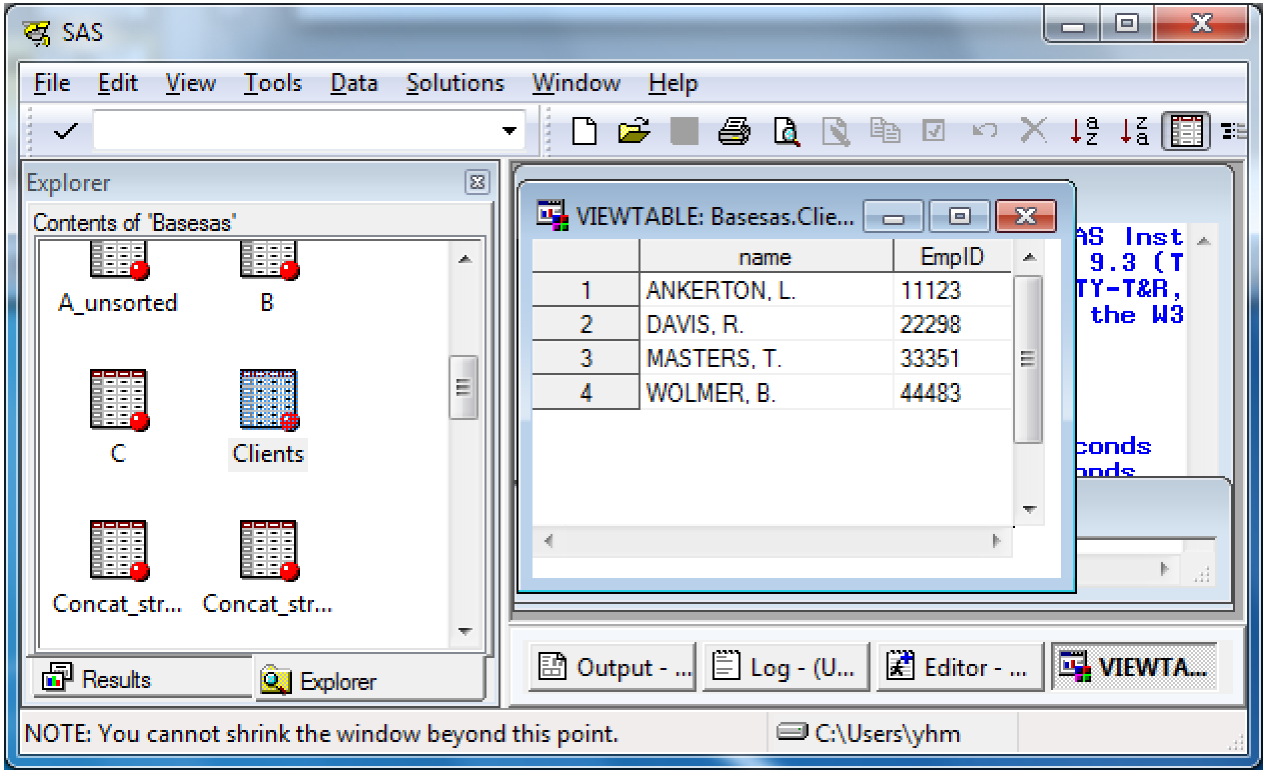
Path:

Options:

C:\Users\yhm

Accessing a SAS dataset stored in a library

Double clicking in the explorer window on ***Libraries*** --> *a specific library* --> *filename* will allow you to open a SAS dataset in the VIEWTABLE window.



VIEWTABLE: Basesas.Clie...

	name	EmpID
1	ANKERTON, L.	11123
2	DAVIS, R.	22298
3	MASTERS, T.	33351
4	WOLMER, B.	44483

NOTE: You cannot shrink the window beyond this point.

Accessing a SAS dataset stored in a library

Accessing a file in a SAS program. For example:

```
libname MPS “c:\MPS”;
data MPS.new;
     weight=35;
run;
```

- “**MPS**” is the name of the SAS library that contains the file
- “**new**” is the name of the file itself.

A **period** separates the library name from the filename.

This program creates the SAS data set “**new**” stored in “**c:\MPS**”. Using Windows explorer you will find a file called **new.sas7bdat**.

Rules for SAS Data Set Names

Rules for making SAS data set names:

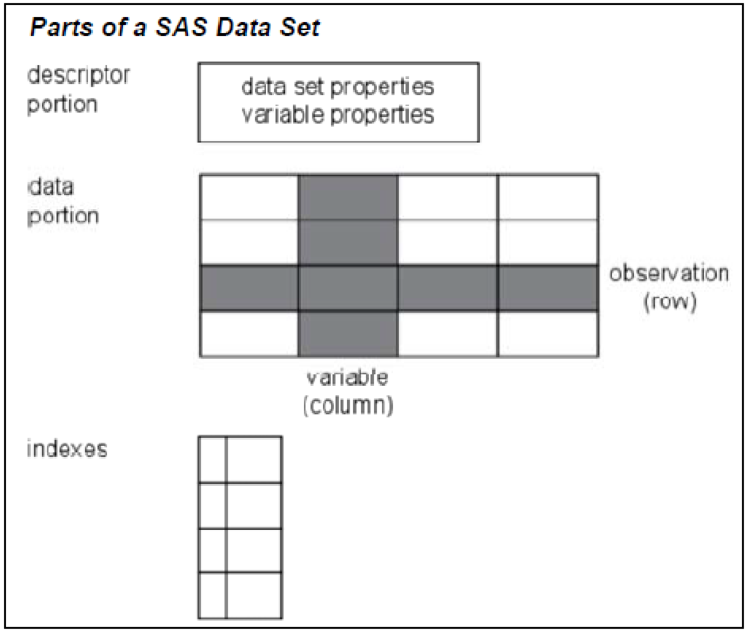
- can be **1** to **32** characters long
- must begin with a letter (A–Z, either uppercase or lowercase) or an underscore (_)
- can continue with any combination of numbers, letters, or underscores.

These are examples of valid data set names:

- Payroll
- LABDATA1995_1997
- EstimatedTaxPayments3

SAS Data Sets

- A SAS data set is a file in a special format that consists of two parts:
    - a descriptor portion
    - a data portion
- Sometimes a SAS data set also has indexes to help locate records efficiently.



SAS Data Sets

- The **descriptor portion** gives general information about the file such as the name, date and time of creation, number of observations and variables.
- The **descriptor portion** also contains information about the attributes of each variable in the data set:
  - **name** – the variable name (the same naming rules as SAS data set names)
  - **type** - character or numeric
  - **length** - number of bytes used to store it (default of 8 bytes for numeric; each character use one byte storage and a character variable can be up to 32,767 bytes long)
  - **format** - affects the way a data value is written/displayed
  - **informat** - affects the way data is read into a SAS data set
  - **label** - can provide more descriptive info than is in the name

More about SAS Variable Attributes

- **Formats** are variable attributes that affect the way data values are written out.

  For example, to display the value *5678* as 5,678.00 in a report, you can use the COMMA8.2 format, which specifies a total width of 8 including 2 decimal places.

- **Informats** determine how data values are read into a SAS data set.

  For example, the value *$1,234.00* contains two special characters, a dollar sign ($) and a comma (,). You can use an informat to read the value while removing the dollar sign and comma, and then store the resulting value as a standard numeric value.

SAS Data Sets

- The **data portion** is a table with **observations** (rows) and **variables** (columns) that SAS can process.

Person	Age	Weight	Height
John	34	183	172
Mary	45	205	194
Ann	60	165	158

Missing Values in SAS Data Sets

- Missing **numeric** data values are represented by a “.” and missing **character** data values by an empty cell.

Person	Age	Weight	Height
John	34	183	172
	45	205	194
Ann	.	165	158