临床研究SAS高级编程
——SAS概述
What is SAS?

- Used to stand for “Statistical Analysis System”
- One of the largest software companies in the world
- Now the most widely used statistical software
Contents

1. An Overview of SAS
2. SAS Windowing Environment
3. SAS Programs
4. Mastering Fundamental Concepts
Components of the SAS System

- **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/QC**
  - quality control
What to do with SAS

- access data from many sources
- manage your data
- analyze your data and present it as meaningful information in a report that you can deliver to any platform, in any format you need.
General Work Flow

Data → DATA Step → SAS Data Sets → PROC Steps → Information
2. SAS Windowing Environment

When you first start SAS, the five main SAS windows open:

1. In the Explorer window, you can view and manage your SAS files
2. In the Editor window, to enter, edit, and submit SAS programs
3. The Log window displays messages about your SAS session and any SAS programs that you submit.
4. The Output window displays the output from SAS programs that you submit.
5. The Results window helps you navigate and manage output from SAS programs that you submit.
Enhanced Editor

- It provides many helpful features, including color coding and automatically retaining the program after each submit, eliminating the need to recall your program.

Submit SAS program

- Issue the submit command
- Click on
- Select Run Submit on the toolbar
A SAS program is a sequence of steps that the user submits for execution.
3. SAS Programs

SAS Programs

data work.staff;
  infile 'raw-data-file';
  input LastName $ 1-20 FirstName $ 21-30
          JobTitle $ 36-43 Salary 54-59;
run;
proc print data=work.staff;
run;
proc means data=work.staff;
  class JobTitle;
  var Salary;
run;
You can invoke SAS in

- Interactive windowing mode (SAS windowing environment)
- Interactive menu-driven mode (SAS Enterprise Guide, SAS/ASSIST, SAS/EIS, etc)
- Batch mode
- Noninteractive mode
SAS Log

```sas
1  data work.staff;
2      infile 'raw-data-file';
3      input LastName $ 1-20 FirstName $ 21-30
4                  JobTitle $ 36-43 Salary 34-59;
5     run;
NOTE: The infile 'raw-data-file' is:
      File Name= 'raw-data-file',
      RECFM=V,LRECL=156
NOTE: 18 records were read from the infile 'raw-data-file'.
      The minimum record length was 59.
      The maximum record length was 59.
NOTE: The data set WORK.STAFF has 18 observations and 4 variables.
6     proc print data=work.staff;
7     run;
NOTE: There were 18 observations read from the dataset WORK.STAFF.
8     proc means data=work.staff;
9      class JobTitle;
10     var Salary;
11     run;
NOTE: There were 18 observations read from the dataset WORK.STAFF.
```
## PROC PRINT Output

<table>
<thead>
<tr>
<th>Obs</th>
<th>LastName</th>
<th>First Name</th>
<th>JobTitle</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TORRES</td>
<td>IAN</td>
<td>Pilot</td>
<td>50000</td>
</tr>
<tr>
<td>2</td>
<td>LANGKAMM</td>
<td>SARAH</td>
<td>Mechanic</td>
<td>80000</td>
</tr>
<tr>
<td>3</td>
<td>SMITH</td>
<td>MICHAEL</td>
<td>Mechanic</td>
<td>40000</td>
</tr>
<tr>
<td>4</td>
<td>LEISTNER</td>
<td>COLIN</td>
<td>Mechanic</td>
<td>36000</td>
</tr>
<tr>
<td>5</td>
<td>WADE</td>
<td>KIRSTEN</td>
<td>Pilot</td>
<td>85000</td>
</tr>
<tr>
<td>6</td>
<td>TOMAS</td>
<td>HARALD</td>
<td>Pilot</td>
<td>105000</td>
</tr>
<tr>
<td>7</td>
<td>WAUGH</td>
<td>TIM</td>
<td>Pilot</td>
<td>70000</td>
</tr>
<tr>
<td>8</td>
<td>LEHMANN</td>
<td>DAGMAR</td>
<td>Mechanic</td>
<td>64000</td>
</tr>
<tr>
<td>9</td>
<td>TRETTHAHN</td>
<td>MICHAEL</td>
<td>Pilot</td>
<td>100000</td>
</tr>
<tr>
<td>10</td>
<td>TIEZ</td>
<td>OTTO</td>
<td>Pilot</td>
<td>45000</td>
</tr>
<tr>
<td>11</td>
<td>O'DONOGHUE</td>
<td>ART</td>
<td>Mechanic</td>
<td>52000</td>
</tr>
<tr>
<td>12</td>
<td>WALKER</td>
<td>THOMAS</td>
<td>Pilot</td>
<td>95000</td>
</tr>
<tr>
<td>13</td>
<td>NOROVIITA</td>
<td>JOACHIM</td>
<td>Mechanic</td>
<td>78000</td>
</tr>
<tr>
<td>14</td>
<td>OESTERBERG</td>
<td>ANJA</td>
<td>Mechanic</td>
<td>80000</td>
</tr>
<tr>
<td>15</td>
<td>LAUFFER</td>
<td>CRAIG</td>
<td>Mechanic</td>
<td>40000</td>
</tr>
<tr>
<td>16</td>
<td>TERR</td>
<td>JUGDISH</td>
<td>Pilot</td>
<td>45000</td>
</tr>
<tr>
<td>17</td>
<td>WAGSCHAL</td>
<td>NADJA</td>
<td>Pilot</td>
<td>77000</td>
</tr>
<tr>
<td>18</td>
<td>TOERMOEN</td>
<td>JOCHEN</td>
<td>Pilot</td>
<td>65000</td>
</tr>
</tbody>
</table>
4. Mastering Fundamental Concepts

4.1 SAS dataset
4.2 SAS variable and Values
4.3 Syntax
Overview of SAS data sets

A SAS data set is a file that consists of two parts: a descriptor portion and a data portion. Sometimes a SAS data set also points to one or more indexes, which enable SAS to locate records in the data set more efficiently.
4.1 SAS Data Sets - Descriptor portion

Descriptor portion

- The descriptor portion of a SAS data set contains information about the data set, including
  - the name of the data set
  - the date and time that the data set was created
  - the number of observations
  - the number of variables.

- Let's look at another SAS data set. The table below lists part of the descriptor portion of the data set Clinic.Insure.

<table>
<thead>
<tr>
<th>Data Set Name:</th>
<th>CLINIC.INSURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Type:</td>
<td>DATA</td>
</tr>
<tr>
<td>Engine:</td>
<td>V8</td>
</tr>
<tr>
<td>Created:</td>
<td>10:05 Tuesday, March 30, 1999</td>
</tr>
<tr>
<td>Observations:</td>
<td>21</td>
</tr>
<tr>
<td>Variables:</td>
<td>7</td>
</tr>
<tr>
<td>Indexes:</td>
<td>0</td>
</tr>
<tr>
<td>Observation Length:</td>
<td>64</td>
</tr>
</tbody>
</table>
### Example

**Partial PROC CONTENTS Output**

The SAS System

The CONTENTS Procedure

<table>
<thead>
<tr>
<th>Data Set Name:</th>
<th>WORK.STAFF</th>
<th>Observations:</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Type:</td>
<td>DATA</td>
<td>Variables:</td>
<td>4</td>
</tr>
<tr>
<td>Engine:</td>
<td>V9</td>
<td>Indexes:</td>
<td>0</td>
</tr>
<tr>
<td>Created:</td>
<td>Monday, December 01, 2003 10:36:59 AM</td>
<td>Observation Length</td>
<td>45</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Monday, December 01, 2003 10:36:59 AM</td>
<td>Deleted Observations</td>
<td>0</td>
</tr>
<tr>
<td>Protection:</td>
<td></td>
<td>Compressed:</td>
<td>NO</td>
</tr>
<tr>
<td>Data Set Type:</td>
<td></td>
<td>Sorted:</td>
<td>NO</td>
</tr>
</tbody>
</table>

**Alphabetic List of Variables and Attributes**

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>FirstName</td>
<td>Char</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>JobTitle</td>
<td>Char</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>LastName</td>
<td>Char</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Salary</td>
<td>Num</td>
<td>8</td>
</tr>
</tbody>
</table>

PROC CONTENTS DATA=sas-data-set; RUN;

or

PROC DATASETS LIBRARY=sas-library;

CONTENTS sas-data-set; Run;
4.1 SAS Data Sets - Data portion

The data portion of a SAS data set is a collection of data values that are arranged in a rectangular table. In the example below, the name Jones is a data value, the weight 158.3 is a data value, and so on.

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jones</td>
<td>M</td>
<td>48</td>
<td>128.6</td>
</tr>
<tr>
<td>Laverne</td>
<td>M</td>
<td>58</td>
<td>158.3</td>
</tr>
<tr>
<td>Jaffe</td>
<td>F</td>
<td>.</td>
<td>115.5</td>
</tr>
<tr>
<td>Wilson</td>
<td>M</td>
<td>28</td>
<td>170.1</td>
</tr>
</tbody>
</table>
4.1 SAS Data Sets – observations & columns

**Observations (Rows)**

- **Rows** (called **observations**) in the data set are collections of data values that usually relate to a single object. The values *Jones*, *M*, 48, and 128.6 comprise a single observation in the data set shown below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jones</td>
<td>M</td>
<td>48</td>
<td>128.6</td>
</tr>
<tr>
<td>Laverne</td>
<td>M</td>
<td>58</td>
<td>158.3</td>
</tr>
<tr>
<td>Jaffe</td>
<td>F</td>
<td>.</td>
<td>115.5</td>
</tr>
<tr>
<td>Wilson</td>
<td>M</td>
<td>28</td>
<td>170.1</td>
</tr>
</tbody>
</table>

**Variables (Columns)**

- Columns (called **variables**) in the data set are collections of values that describe a particular characteristic. The values *Jones*, *Laverne*, *Jaffe*, and *Wilson* comprise the variable Name in the data set shown below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jones</td>
<td>M</td>
<td>48</td>
<td>128.6</td>
</tr>
<tr>
<td>Laverne</td>
<td>M</td>
<td>58</td>
<td>158.3</td>
</tr>
<tr>
<td>Jaffe</td>
<td>F</td>
<td>.</td>
<td>115.5</td>
</tr>
<tr>
<td>Wilson</td>
<td>M</td>
<td>28</td>
<td>170.1</td>
</tr>
</tbody>
</table>
4. Mastering Fundamental Concepts

4.1 SAS dataset
4.2 SAS variable and Values
4.3 Syntax
4.2.1 SAS Names
- SAS Data Set and Variable Names

- Can be 32 characters long
- Can be uppercase, lowercase, or mixed-case
- Must start with a letter or underscore, subsequent characters can be letters, underscore, or numeric digits

In version 6 and earlier, data set and variable names can only be a maximum of 8 characters long.

Starting in Version 8, special characters can be used in data set and variable names if you put the name in quotes followed immediately by letter N

Example: ‘Flight#’n;

In order to use special characters in variables names, the VALIDVARNAME option must be set to ANY;

Example: options validvarname=any;
A variable's type is either **character** or **numeric**.

- Character variables, such as Name (shown below), can contain **any** values.

- Numeric variables, such as Policy and Total (shown below), can contain **only numeric values** (the digits 0 through 9, +, -, ., and E for scientific notation).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Length</th>
<th>Format</th>
<th>Informat Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy</td>
<td>Num</td>
<td>8</td>
<td></td>
<td>Policy Number</td>
</tr>
<tr>
<td>Total</td>
<td>Num</td>
<td>8</td>
<td>DOLLAR8.2 COMMA10.</td>
<td>Total Balance</td>
</tr>
<tr>
<td>Name</td>
<td>Char</td>
<td>20</td>
<td></td>
<td>Patient Name</td>
</tr>
</tbody>
</table>
A variable's **length** (the number of bytes used to store it) is related to its type.

- Character variables can be up to **32K** long. In the example below, Name has a length of 20 characters and uses 20 bytes of storage.
- All numeric variables have a default length of **8**. Numeric values (no matter how many digits they contain) are stored as floating-point numbers in 8 bytes of storage, unless you specify a different length.

<table>
<thead>
<tr>
<th>Variable Type</th>
<th>Length</th>
<th>Format</th>
<th>Informat Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy</td>
<td>Num</td>
<td>8</td>
<td>Policy Number</td>
</tr>
<tr>
<td>Total</td>
<td>Num</td>
<td>8</td>
<td>DOLLAR8.2 COMMA10. Total Balance</td>
</tr>
<tr>
<td>Name</td>
<td>Char</td>
<td>20</td>
<td>Patient Name</td>
</tr>
</tbody>
</table>
A variable can have a **label**, which consists of descriptive text up to **256** characters long. By default, many reports identify variables by their names. You may want to display more descriptive information about the variable by assigning a label to the variable.

For example, you can label Policy as **Policy Number**, Total as **Total Balance**, and Name as **Patient Name** to display these labels in reports.
4.2.5 SAS Data Value - Date

- SAS stores date values as numeric values.
- A SAS date value is stored as the number of days between Jan 1\textsuperscript{st}, 1960, and a specific date.
4.2.5 SAS Data Value - Missing Data Values

- A value must exist for every variable for each observation.
- Missing values are valid values
  - A numeric missing value is displayed as a period
  - A character missing value is displayed as a blank

<table>
<thead>
<tr>
<th>LastName</th>
<th>FirstName</th>
<th>JobTitle</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>TORRES</td>
<td>JAN</td>
<td>Pilot</td>
<td>50000</td>
</tr>
<tr>
<td>LANGKAMM</td>
<td>SARAH</td>
<td>Mechanic</td>
<td>80000</td>
</tr>
<tr>
<td>SMITH</td>
<td>MICHAEL</td>
<td>Mechanic</td>
<td></td>
</tr>
<tr>
<td>WAGSCHAL</td>
<td>NADJA</td>
<td>Pilot</td>
<td>77500</td>
</tr>
<tr>
<td>TOERMOEN</td>
<td>JOCHEN</td>
<td></td>
<td>65000</td>
</tr>
</tbody>
</table>
4. Mastering Fundamental Concepts

4.1 SAS dataset
4.2 SAS variable and Values
4.3 Syntax
4.3.1 Basic Structure of SAS

**DATA step**
- reads data from external sources,
- manipulates and combines it with other data set and
- prints reports

**PROC step**
- perform analysis on the data, and
- produce output
Step Boundaries

SAS Steps begin with a
- DATA statement
- PROC statement

SAS Detects the end of a step when if encounters
- A RUN statement (for most steps)
- A QUIT statement (for some procedures)
- The beginning of another steps
data work.staff;
  infile 'raw-data-file';
  input LastName $ 1-20 FirstName $ 21-30
    JobTitle $ 36-43 Salary 54-59;
run;

proc print data=work.staff;
proc means data=work.staff;
  class JobTitle;
  var Salary;
run;
4.3.2 SAS Syntax Rules

SAS statements usually begin with an identifying keyword, always end with a semicolon;

SAS statements are free-format:
- One or more blanks or special characters 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

SAS programming statements are easier to read if you begin DATA, PROC, and RUN statements in column one and indent the other statements.
SAS statements...

- usually begin with a SAS keyword.
- can begin anywhere on a line.
- contain blanks or special characters to separate words.
- can be combined on one line.
- can continue over several lines.
- always end with a semicolon.
Definition of word

- A word or token in the SAS programming language is a collection of characters that communicates a meaning to SAS and which cannot be divided into smaller units that can be used independently.
- A word or token ends when SAS encounters one of the following:
  - the beginning of a new token
  - a blank after a name or a number token
  - the ending quotation mark of a literal token.
Types of words or tokens

- **Name** is a series of characters that begin with a letter or an underscore. Later characters can include letters, underscores, and numeric digits.
  - Examples: _new, year_99

- **Literal** consists of 1 to 32,767 characters enclosed in single or double quotation marks.
  - Examples: ‘Chicago’, “1990-91”

- **Number** is composed entirely of numeric digits, with an optional decimal point and a leading plus or minus sign. SAS also recognizes: scientific (E-) notation, hexadecimal notation, missing value symbols, and date and time literals.
  - Examples: 2.35, 5.4E-1, 0b0x, ‘24aug90’d

- **Special character** is usually any single keyboard character other than letters, numbers, the underscore, and the blank.
  - Examples: =, +, @
Placement and spacing of words in SAS statements

Spacing requirements

- You can begin SAS statements in any column of a line and write several statements on the same line.
- You can begin a statement on one line and continue it on another line, but you cannot split a word between two lines.
- A blank is not treated as a character in a SAS statement unless it is enclosed in quotation marks as a literal or part of a literal.
- The rules for recognizing the boundaries of words or tokens determine the use of spacing between them in SAS programs.

Example:

- Blanks are not required: `total=x+y;`
- Blanks are required: `input group 15 room 20;`
There are two forms of comment statements:

**One line:**
- `*text;`
- To use the asterisk at the beginning of the comment.

**Multiple lines:**
- `/*text*/`
- Type `/` to begin a comment
- Type `*/` to end the comment

SAS ignores text between comment symbols during processing.
The following program shows some of the ways comments can be used to describe a SAS program.

```sas
/* Read national sales data for vans */
* from an external raw data file;
data perm.vansales;
  infile vandata;
  input @1 Region $9.
       @13 Quarter 1. /* Values are 1, 2, 3, or 4 */
       @16 TotalSales comma11.;
/* Print the entire data set */
proc print data=perm.vansales;
run;
```
Quiz

Which of the following variable names is valid?

- a. 4BirthDate
- b. $Cost
- c. _Items_
- d. Tax-Rate

Correct answer:  c

Variable names follow the same rules as SAS data set names. They can be 1 to 32 characters long, must begin with a letter (A-Z, either uppercase or lowercase) or an underscore, and can continue with any combination of numbers, letters, or underscores.
Exercises

- SAS statements usually begin with a ________
- Every SAS statement ends with a _________
- Character variable values can be up to ______ characters long, and use ___byte(s) of storage per character
- A SAS variable name has __to __characters and begin with a ___or an ______
- By default, numeric variables are stored in ___bytes of storage
- The internally stored SAS date value for Jan 23\textsuperscript{rd}, 1960, is _____
Exercises

- A missing character value is displayed as a ____
- A missing numeric value is displayed as a ______
- What are the two kinds of Steps?
- What are the two portions of SAS data set?
- What are the two types of variables?
Q&A