

LECTURE 05

21CSE295T

Statistical Data Analytics And Integration

Unit 3 Advanced Statistical Techniques in SAS

3.1	Determining the Structure and Contents of Data Sets
3.2	Testing Program
3.3	Methods for Combining SAS Data Sets
3.4	One-to-One Reading
3.5	Concatenating
3.6	Match-Merging
3.7	Renaming Variables
3.8	Excluding Unmatched Observations
3.9	Selecting Matching Observations

Virupakshan K

+91-9840847872 | VIRU@KART.LA

SYLLABUS

21CSE295T Statistical Data Analytics And Integration

VirupakshanK@gmail.com

Unit 1 Introduction to Statistical Analysis Computational Statistics- Statistical Pattern Recognition, Nonparametric Regression, Exploratory Data Analysis, Monte Carlo Methods for Inferential Statistics-Basic concepts of statistics, Types of data (categorical, numerical), Populations and samples, Measures of central tendency (mean, median, mode), Measures of variability (range, variance, standard deviation), Descriptive vs. inferential statistics. Tutorial:	9 hours
SAS Statements, Structure, Libraries, Rules, Datasets, Representation of missing values, Assigning Libraries, PROC CONTENTS, visualize the descriptor and datasets attributes	
Unit 2 Data Analysis Using SAS Basic Reports – Selecting Variables – Identifying observations – Subsetting the Data – Limiting the observations-SORT Procedure – Generating Column totals – Specifying titles and footnotes in procedure output – Assigning Descriptive labels and permanent label -How SAS Processes Programs – Compilation Phase – PDV – Descriptor Portion-Execution Phase – Data Portion – Debugging a Data Step – Testing Programs-BY-Group Processing Definitions – Preprocessing Data – Determine Whether the Data Requires Preprocessing – Sorting Observations for BY-Group Processing-FIRST. and LAST. DATA Step Variables – How the DATA Step Identifies BY Groups – Assignment Statements – Operators in SAS Expressions Tutorial:	9 hours
Introduction of report and PRINT Procedure-Hands on practice of VAR, ID, WHERE statement, where expression[eq, ne, gt, lt, ge, le, AND, OR, CONTAINS]	
Unit 3 Advanced Statistical Techniques in SAS Determining the Structure and Contents of Data Sets – Testing Program – Methods of Combining SAS Data Sets-One-to-One Reading – Concatenating – Match-Merging – Renaming Variables – Excluding Unmatched Observations -Selecting Matching Observations -Basics of DO Loops – Executing DO Loops -OUTPUT Statements – Nesting DO Loops-Iteratively Processing Observations from a Data Set -DO UNTIL Statement – DO WHILE Statement Defining and Processing Arrays –Arrays -Applying SAS Formats and Informats – Specifying SAS Formats Tutorial:	9 hours
One-to-One Reading to Combine Data Sets-Concatenating to Combine Data Sets-Match-Merging to Combine Data Sets-Renaming Variables-Processing Iterative DO Loops-Indenting and Nesting DO Groups	
Unit 4 SAS Programming and Automation Definitions – Reading Date and Time with Informats – Displaying Date and Time Values with Formats-Basics of SAS Functions - SAS Functions Syntax – Converting Data with Functions- The MEANS Procedure – Specifying Descriptive Statistics – Limiting Decimal Places – Specifying Variable Using VAR Statement – Summarized Data Set Using OUTPUT Statement – The FREQ Procedure-Specifying Variable Using TABLES Statements - Two-Way and N-Way Tables – LIST Options-The Output Delivery System (ODS) – HTML Outputs – PDF Outputs-RTF Outputs – EXCEL Outputs – The EXPORT Procedure Tutorial:	9 hours
Reading Date and Time with Formats and Informats- Execution of Data Values Using Numeric Functions-Creating a Descriptive Statistics by Using PROC MEANS-Creating a Descriptive Statistics by Using PROC FREQ-Creating Outputs with HTML, PDF, RTF, EXCEL Formats-Exporting a External File	
Unit 5 Advanced SAS Topics Creating a Default List Report – Windowing/Non Windowing Mode - Define Statement-Defining Variables – Column Statement – Selecting Observations – Defining the Variables-Group Processing the Variables by Using Different Options – Computing the Variable-Bar Chart – Box Plot – Mean Measurement Over the Time using Line Plot – Spaghetti Plot – Survival Plot – Waterfall Plot – Forest Plot-Introduction to Macro Facility – SAS Programs and Macro Processing – Macro Variables – Macro Processing – Scopes of Macro Variables – Macro Expressions-Scopes of Macro Variables – Macro Expressions-Macro Quoting – Interfaces with Macro Facility -Storing and Reusing Macros. Tutorial:	9 hours
Creating Graphs by Using Bar Chart-Creating Graphs Using Various Types of Plots-Generating SAS Code Using Macros-Defining User-Defined Macro Variables-Creating a Tables Using PROC SQL with different Clauses- Generating a Cartesian Product -Combining Tables Using Multiple Joins	

LECTURE 05 TOPICS

VirupakshanK@gmail.com

Unit 3 Advanced Statistical Techniques in SAS

3.1	Determining the Structure and Contents of Data Sets
3.2	Testing Program
3.3	Methods for Combining SAS Data Sets
3.4	One-to-One Reading
3.5	Concatenating
3.6	Match-Merging
3.7	Renaming Variables
3.8	Excluding Unmatched Observations
3.9	Selecting Matching Observations
3.10	Basics of DO Loops
3.11	Executing DO Loops
3.12	OUTPUT Statements
3.13	Nesting DO Loops
3.14	Iteratively Processing Observations from a Data Set
3.15	DO UNTIL Statement
3.16	DO WHILE Statement
3.17	Defining and Processing Arrays
3.18	Applying SAS Formats and Informats
3.19	Specifying SAS Formats

3.1	Determining the Structure and Contents of Data Sets
3.2	Testing Program
3.3	Methods for Combining SAS Data Sets
3.4	One-to-One Reading
3.5	Concatenating
3.6	Match-Merging
3.7	Renaming Variables
3.8	Excluding Unmatched Observations
3.9	Selecting Matching Observations

3.1.1 Determining the Structure and Contents of Data Sets

Determining the structure and contents of a SAS data set means examining its **metadata** and **data values**. Metadata includes variable names, types, lengths, formats, informats, labels, and number of observations.

Key SAS Procedures

```
PROC CONTENTS DATA=dataset;  
RUN;  
  
PROC PRINT DATA=dataset;  
RUN;  
  
PROC MEANS DATA=dataset;  
RUN;  
  
PROC FREQ DATA=dataset;  
RUN;
```

Explanation

- PROC CONTENTS shows the structure of a data set.
- PROC PRINT displays actual observations.
- PROC MEANS summarizes numeric variables.
- PROC FREQ summarizes categorical variables.

3.1.2 Example

Suppose a data set `students` has:

ID	Name	Marks
101	Ravi	78
102	Meena	85
103	Arun	69

```
PROC CONTENTS DATA=students;  
RUN;
```

```
PROC PRINT DATA=students;  
RUN;
```

Expected structure:

Variable	Type	Length
ID	Numeric	8
Name	Character	5
Marks	Numeric	8

Number of observations = 3

Number of variables = 3

3.1.3 Questions

Likely 2-Mark Questions

1. What is the use of `PROC CONTENTS` in SAS?
2. Differentiate between `PROC PRINT` and `PROC CONTENTS`.
3. What is metadata in a SAS data set?

Likely 5-Mark Questions

1. Explain how to determine the structure and contents of a SAS data set.
2. Discuss the role of `PROC CONTENTS`, `PROC PRINT`, `PROC MEANS`, and `PROC FREQ` in data exploration.

3.2.1 Testing Program

Testing a SAS program means checking whether the program runs correctly, produces expected results, and does not contain syntax or logical errors.

Key SAS Options

```
OPTIONS OBS=10;  
OPTIONS FIRSTOBS=5 OBS=20;  
OPTIONS MPRINT SYMBOLGEN MLOGIC;
```

MPRINT: Translates your macro language into regular SAS code. It prints the actual SAS statements generated by your macro after macro variables are substituted

SYMBOLGEN: Displays exactly what your macro variables resolve to. For example, if your code has `&Year`, it will show that it resolves to 2026

MLOGIC: Traces the execution flow of your macro. It shows you when a macro begins, when parameters are passed, and whether `%IF-%THEN` conditions evaluate to TRUE or FALSE

To turn them off when you're done, run the following statement to avoid cluttering your log:

```
OPTIONS NOMPRINT NOSYMBOLGEN NOMLOGIC;
```

3.2.2 Testing Program

Testing helps identify:

- Syntax errors
- Data input errors
- Missing values
- Incorrect variable names
- Wrong merge or calculation logic

Common testing methods:

- 1.Use a small number of observations.
- 2.Check the SAS log.
- 3.Use `PROC PRINT`.
- 4.Use `PUT` statements for debugging.
- 5.Use `OPTIONS OBS=` to restrict processing.

3.2.3 Testing Program

Suppose a data set has 1,000 records. To test only the first 10:

```
OPTIONS OBS=10;  
  
PROC PRINT DATA=sales;  
RUN;
```

Only 10 observations are printed.

If the first observation starts from 5 and ends at 15:

```
OPTIONS FIRSTOBS=5 OBS=15;  
  
PROC PRINT DATA=sales;  
RUN;
```

Number of observations printed: $15 - 5 + 1 = 11$

3.3.1 Methods for Combining SAS Data Sets

Combining SAS data sets means creating a new data set by bringing together observations or variables from two or more existing data sets.

Main Methods

Method	Purpose
One-to-one reading	Combines observations side by side
Concatenation	Stacks data sets vertically
Match-merging	Combines data sets based on common variable
Interleaving	Combines sorted data sets by order

```
DATA new;  
MERGE data1 data2;  
BY common_variable;  
RUN;
```

```
DATA new;  
SET data1 data2;  
RUN;
```

Explanation

- **SET** is used for concatenation and one-to-one reading.
- **MERGE** is used for combining observations based on a key variable.
- **BY** variable is used for match-merging.

3.3.2 Methods for Combining SAS Data Sets – MERGE example

Data set A:

ID	Name
1	Ravi
2	Meena

Data set B:

ID	Salary
1	30000
2	40000

Output:

ID	Name	Salary
1	Ravi	30000
2	Meena	40000

```
DATA combined;  
MERGE A B;  
BY ID;  
RUN;
```

3.4.1 One-to-One Reading

Definition

One-to-one reading combines observations from two or more SAS data sets **based on their observation number**, not by a common key variable.

Syntax

```
DATA new;  
SET data1;  
SET data2;  
RUN;
```

Explanation

The first observation from **data1** combines with the first observation from **data2**, the second with the second, and so on.

If the data sets have unequal observations, processing stops when the smallest data set ends.

3.4.2 One-to-One Reading - Example

Data set A:

Name
Ravi
Meena
Arun

Data set B:

Marks
80
90

```
DATA result;  
SET A;  
SET B;  
RUN;
```

OUTPUT: Data set result:

Name	Marks
Ravi	80
Meena	90

Only 2 observations are produced because data set B has only 2 observations.

3.5.1 Concatenating

Concatenation means appending observations from one SAS data set below another. It combines data sets vertically.

Syntax

```
DATA new;  
SET data1 data2 data3;  
RUN;
```

Explanation

- Variables with the same name are aligned.
- Variables present in only one data set will have missing values for observations from other data sets.
- Number of observations in output equals the sum of observations in input data sets.

$$\text{Total observations} = n_1 + n_2 + n_3 + \dots + n_k$$

3.5.2 Concatenating - Example

January data:

ID	Sales
1	1000
2	1500

February data:

ID	Sales
3	2000
4	2500
5	3000

```
DATA quarter1;  
SET jan feb;  
RUN;
```

Output has 5 observations.

3.6.1 Match-Merging

Match-merging combines observations from two or more SAS data sets based on common values of one or more **BY** variables.

```
PROC SORT DATA=data1;  
BY ID;  
RUN;  
  
PROC SORT DATA=data2;  
BY ID;  
RUN;  
  
DATA new;  
MERGE data1 data2;  
BY ID;  
RUN;
```

Explanation

Before match-merging, data sets must usually be sorted by the **BY** variable.

3.6.2 Match-Merging - Example

Data set student

ID	Name
1	Ravi
2	Meena

Data set marks

ID	Score
1	85
2	90

Output after merge:

ID	Name	Score
1	Ravi	85
2	Meena	90

```
DATA final;  
MERGE student marks;  
BY ID;  
RUN;
```

3.7.1 Renaming Variables

Renaming variables means changing variable names in a SAS data set.

Using **RENAME** statement:

```
DATA new;  
SET old;  
RENAME oldname = newname;  
RUN;
```

Using data **set** option:

```
DATA new;  
SET old(RENAME=(oldname=newname));  
RUN;
```

Renaming is useful when:

- Variable names conflict during merging.
- Names are unclear.
- Standard naming conventions are required.

3.7.2 Renaming Variables - Example

Original data:

EmpID	Sal
101	30000

```
DATA employee_new;  
SET employee;  
RENAME Sal = Salary;  
RUN;
```

Output data:

EmpID	Salary
101	30000

3.8.1 Excluding Unmatched Observations

Excluding unmatched observations means keeping only those observations that are present in all merged data sets.

```
DATA matched;  
MERGE data1(IN=a) data2(IN=b);  
BY ID;  
IF a AND b;  
RUN;
```

Condition	Meaning
IF a;	Keep observations from data1
IF b;	Keep observations from data2
IF a AND b;	Keep only matched observations
IF a AND NOT b;	Keep only unmatched from data1

The `IN=` data set option creates temporary variables that indicate whether an observation comes from a particular data set.

3.8.2 Excluding Unmatched Observations

Data set A:

ID	Name
1	Ravi
2	Meena
3	Arun

Data set B:

ID	Loan
1	50000
3	70000

Using:

```
DATA matched;  
MERGE A(IN=a) B(IN=b);  
BY ID;  
IF a AND b;  
RUN;
```

Output:

ID	Name	Loan
1	Ravi	50000
3	Arun	70000

ID 2 is excluded.

3.9.1 Selecting Matching Observations

Selecting matching observations means selecting only records that have common **BY** variable values in two or more data sets.

Syntax

```
DATA common;  
MERGE data1(IN=x) data2(IN=y);  
BY ID;  
IF x AND y;  
RUN;
```

Explanation

This is similar to an inner join in SQL.
Only observations that exist in both data sets are selected.

3.9.2 Selecting Matching Observations - Example

Registered customers:

ID	Name
101	Asha
102	Bala
103	Charu

Purchase data:

ID	Amount
101	500
103	800
105	900

Matching IDs are 101 and 103.

```
DATA buyers;  
MERGE registered(IN=r) purchase(IN=p);  
BY ID;  
IF r AND p;  
RUN;
```

Output data:

ID	Name	Amount
101	Asha	500
103	Charu	800

3.9.3 Selecting Matching Observations - Example

```
DATA buyers;  
MERGE registered(IN=r) purchase(IN=p);  
BY ID;  
IF r AND p;  
RUN;
```

1. DATA buyers;

Creates a new dataset named **buyers**.

The merged results will be stored in this dataset.

Output dataset: buyers

4. IF r AND p;

This is the most important statement.

It means:

Keep the observation only if it exists in **both** datasets.

2. MERGE registered(IN=r) purchase(IN=p) ;

- Combines two datasets:

- registered
- purchase

- SAS merges them **observation by observation** based on the BY variable.

What does IN= do?

IN= creates a temporary Boolean (0 or 1) variable.

registered(IN=r)

means

- r = 1 if the current observation comes from registered
- r = 0 otherwise

Similarly,

purchase(IN=p)

means

- p = 1 if the observation exists in purchase
- p = 0 otherwise

These variables exist only while the DATA step executes—they are **not** written to the output dataset.

3.9.3b Selecting Matching Observations - Example

IF r AND p;

Truth table:

r	p	Keep?
1	1	✓ Yes
1	0	✗ No
0	1	✗ No
0	0	Impossible

This makes the merge behave like an **inner join**.

LECTURE 05 TOPICS

VirupakshanK@gmail.com

Unit 3 Advanced Statistical Techniques in SAS

3.1	Determining the Structure and Contents of Data Sets
3.2	Testing Program
3.3	Methods for Combining SAS Data Sets
3.4	One-to-One Reading
3.5	Concatenating
3.6	Match-Merging
3.7	Renaming Variables
3.8	Excluding Unmatched Observations
3.9	Selecting Matching Observations
3.10	Basics of DO Loops
3.11	Executing DO Loops
3.12	OUTPUT Statements
3.13	Nesting DO Loops
3.14	Iteratively Processing Observations from a Data Set
3.15	DO UNTIL Statement
3.16	DO WHILE Statement
3.17	Defining and Processing Arrays
3.18	Applying SAS Formats and Informats
3.19	Specifying SAS Formats

3.1	Determining the Structure and Contents of Data Sets
3.2	Testing Program
3.3	Methods for Combining SAS Data Sets
3.4	One-to-One Reading
3.5	Concatenating
3.6	Match-Merging
3.7	Renaming Variables
3.8	Excluding Unmatched Observations
3.9	Selecting Matching Observations

LECTURE 05

21CSE295T

Statistical Data Analytics And Integration

Doubts?

THANK YOU!

Virupakshan K

+91-9840847872 | VIRU@KART.LA