

LECTURE 07

21CSE295T

Statistical Data Analytics And Integration

Unit 4 SAS Programming and Automation

4.1	Reading Date and Time with Informat
4.2	Displaying Date and Time Values with Formats
4.3	Basics of SAS Functions
4.4	SAS Function Syntax
4.5	Converting Data with Functions

Virupakshan K

+91-9840847872 | VIRU@KART.LA

SYLLABUS

21CSE295T Statistical Data Analytics And Integration

VirupakshanK@gmail.com

Unit 1 Introduction to Statistical Analysis

9 hours

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:

- 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

9 hours

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:

- 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

9 hours

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:

- 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

9 hours

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:

- 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

9 hours

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:

- 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 07 TOPICS

Unit 4 SAS Programming and Automation

4.1	Reading Date and Time with Informat
4.2	Displaying Date and Time Values with Formats
4.3	Basics of SAS Functions
4.4	SAS Function Syntax
4.5	Converting Data with Functions
4.6	The MEANS Procedure
4.7	Specifying Descriptive Statistics
4.8	Limiting Decimal Places
4.9	PROC FREQ Procedure
4.10	Two-Way and N-Way Tables
4.11	LIST Options
4.12	Output Delivery System (ODS)
4.13	HTML Outputs
4.14	PDF Outputs
4.15	RTF Outputs
4.16	EXCEL Outputs

4.1	Reading Date and Time with Informat
4.2	Displaying Date and Time Values with Formats
4.3	Basics of SAS Functions
4.4	SAS Function Syntax
4.5	Converting Data with Functions

Unit 4 Learning Objectives

- Read date and time values in SAS using informats.
- Display date and time values using SAS formats.
- Understand and use basic SAS functions.
- Convert character and numeric data using SAS functions.
- Use `PROC MEANS` to generate descriptive statistics.
- Control decimal places in statistical output.
- Use `PROC FREQ` for frequency analysis.
- Create one-way, two-way, and n-way frequency tables.
- Use `LIST` options in frequency tables.
- Generate reports using the Output Delivery System, or ODS.
- Produce HTML, PDF, RTF, and Excel outputs from SAS programs.

4.1 Reading Date and Time with Informat

What is an Informat?

An **informat** tells SAS how to read raw data values.

When SAS reads normal numbers, it can understand them easily. But date and time values can appear in many forms, such as:

- 20JUL2026
- 07/20/2026
- 20-07-2026
- 14:30:00
- 20JUL2026:14:30:00

SAS needs informats to correctly convert these values into internal SAS values.

SAS Date Values

SAS stores dates as the **number of days from January 1, 1960.**

- January 1, 1960 = 0
- January 2, 1960 = 1
- December 31, 1959 = -1

Formula

SAS Date Value = Number of days from 01JAN1960

SAS Time Values

SAS stores time values as the **number of seconds since midnight**.

For example:

- 00:00:00 = 0 seconds

- 01:00:00 = 3600 seconds

- 12:00:00 = 43200 seconds

Formula

SAS Time Value = $(\text{hours} \times 3600) + (\text{minutes} \times 60) + \text{seconds}$

SAS DateTime Values

A SAS datetime value stores the **number of seconds from January 1, 1960 at midnight.**

Formula

SAS Date Time Value=Number of seconds from 01JAN1960:00:00:00

Common SAS Date and Time Informat

Informat	Used For	Example Input
DATE9.	Date in day-month-year form	20JUL2026
MMDDYY10.	Month/day/year	07/20/2026
DDMMYY10.	Day/month/year	20/07/2026
YYMMDD10.	Year/month/day	2026-07-20
TIME8.	Time values	14:30:00
DATETIME20.	Date and time	20JUL2026:14:30:00
ANYDTDTE.	Flexible date reading	July 20, 2026
ANYDTTME.	Flexible time reading	2:30 PM
ANYDTDTM.	Flexible datetime reading	July 20, 2026 2:30 PM

Example: Reading Dates

Example: Reading Dates

```
data students;  
  input Name $ AdmissionDate : date9.;  
  datalines;  
Asha 20JUL2026  
Ravi 15AUG2026  
Meena 01SEP2026  
;  
run;  
  
proc print data=students;  
run;
```

```
data students;  
  input Name $ AdmissionDate : date9.;  
  datalines;  
Asha 20JUL2026  
Ravi 15AUG2026  
Meena 01SEP2026  
;  
run;  
  
proc print data=students;  
run;
```

Example: Reading Dates

Example: Reading Dates

```
data students;  
    input Name $ AdmissionDate : date9.;  
    datalines;  
Asha 20JUL2026  
Ravi 15AUG2026  
Meena 01SEP2026  
;  
run;  
  
proc print data=students;  
run;
```

Explanation

- **AdmissionDate : date9.** tells SAS to read the value as a date.
- Internally, SAS stores the date as a number.
- Without a format, SAS may print the date as a numeric value.

Example: Reading Time

```
data attendance;  
  input Name $ LoginTime : time8.;  
  datalines;  
Asha 09:15:00  
Ravi 10:30:00  
Meena 08:45:00  
;  
run;  
  
proc print data=attendance;  
run;
```

```
data attendance;  
  input Name $ LoginTime : time8.;  
  datalines;  
Asha 09:15:00  
Ravi 10:30:00  
Meena 08:45:00  
;  
run;  
  
proc print data=attendance;  
run;
```

Example: Reading DateTime

```
data login;  
  input Name $ LoginDateTime : datetime20.;  
  datalines;  
Asha 20JUL2026:09:15:00  
Ravi 20JUL2026:10:30:00  
Meena 20JUL2026:08:45:00  
;  
run;  
  
proc print data=login;  
run;
```

```
data login;  
  input Name $ LoginDateTime : datetime20.;  
  datalines;  
  
Asha 20JUL2026:09:15:00  
Ravi 20JUL2026:10:30:00  
Meena 20JUL2026:08:45:00  
;  
run;  
  
proc print data=login;  
run;
```

Simple Diagram: How SAS Reads Dates

Raw Date Value		Informat		SAS Internal Value
20JUL2026	--->	DATE9.	--->	Number of days from 01JAN1960
09:15:00	--->	TIME8.	--->	Seconds from midnight
20JUL2026:09:15	--->	DATETIME20.	--->	Seconds from 01JAN1960:00:00

4.2 Displaying Date and Time Values with Formats

What is a Format?

A **format** tells SAS how to display stored values.

Important distinction:

Concept	Purpose
Informat	Reads raw data into SAS
Format	Displays SAS data values

SAS may store a date as a number, but we can display it as a readable date using a format.

Common Date and Time Formats

Format

Display Example

DATE9.

20JUL2026

DDMMYY10.

20/07/2026

MMDDYY10.

07/20/2026

YYMMDD10.

2026-07-20

WORDDATE.

July 20, 2026

WEEKDATE.

Monday, July 20, 2026

TIME8.

09:15:00

TIMEAMPM.

9:15:00 AM

DATETIME20.

20JUL2026:09:15:00

DTMONYY7.

JUL2026

Example: Applying Date Format

```
data students;  
  input Name $ AdmissionDate : date9.;  
  format AdmissionDate date9.;  
  datalines;  
Asha 20JUL2026  
Ravi 15AUG2026  
Meena 01SEP2026  
;  
run;
```

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

```
data students;  
  input Name $ AdmissionDate : date9.;  
  format AdmissionDate date9.;  
  datalines;  
Asha 20JUL2026  
Ravi 15AUG2026  
Meena 01SEP2026  
;  
run;  
  
proc print data=students;  
run;
```

Example: Applying Date Format

```
data students;  
  input Name $ AdmissionDate : date9.;  
  format AdmissionDate date9.;  
  datalines;  
Asha 20JUL2026  
Ravi 15AUG2026  
Meena 01SEP2026  
;  
run;  
  
proc print data=students;  
run;
```

Output Concept

Name	AdmissionDate
Asha	20JUL2026
Ravi	15AUG2026
Meena	01SEP2026

Example: Different Date Formats

```
data date_formats;  
  input DateValue : date9.;  
  format DateValue worddate.;  
  datalines;  
20JUL2026  
;  
  
run;  
  
proc print data=date_formats;  
run;
```

```
data date_formats;  
  input DateValue : date9.;  
  format DateValue worddate.;  
  datalines;  
20JUL2026  
;  
run;  
  
proc print data=date_formats;  
run;
```

Output

DateValue
July 20, 2026

Example: Formatting Time

VirupakshanK@gmail.com

```
data time_example;  
  input LoginTime : time8.;  
  format LoginTime timeampm.;  
  datalines;  
  
14:30:00  
;  
  
run;  
  
proc print data=time_example;  
run;
```

```
data time_example;  
  input LoginTime : time8.;  
  format LoginTime timeampm.;  
  datalines;  
14:30:00  
;  
run;  
  
proc print data=time_example;  
run;
```

Output

```
LoginTime  
2:30:00 PM
```

4.3 Basics of SAS Functions

What is a SAS Function?

A **SAS function** is a built-in tool that performs a specific task and returns a result.

Functions help us:

- Calculate values.
- Convert data types.
- Work with dates and times.
- Clean character data.
- Perform statistical operations.
- Extract parts of strings.
- Handle missing values.

General Idea

sas

```
Result = FunctionName(argument1, argument2, ...);
```

Example:

sas

```
Total = sum(Marks1, Marks2, Marks3);
```

Here:

- `sum` is the function.
- `Marks1` , `Marks2` , and `Marks3` are arguments.
- `Total` stores the result.

Types of SAS Functions

Function Type	Examples	Purpose
Numeric functions	sum , mean , round	Work with numbers
Character functions	upcase , substr , trim	Work with text
Date functions	today , year , month	Work with dates
Conversion functions	input , put	Convert data types
Mathematical functions	sqrt , abs , log	Mathematical calculations
Missing value functions	coalesce , nmiss , cmiss	Handle missing values

4.4 SAS Function Syntax

Basic Syntax

`new_variable = function_name(argument);`

```
new_variable = function_name(argument);
```

`new_variable = function_name(argument1, argument2, argument3);`

```
new_variable = function_name(argument1, argument2, argument3);
```

Example: SUM Function

```
data marks;  
  input Name $ Test1 Test2 Test3;  
  Total = sum(Test1, Test2, Test3);  
  datalines;  
Asha 80 85 90  
Ravi 70 75 80  
Meena 90 95 92  
;  
run;  
  
proc print data=marks;  
run;
```

```
data marks;  
  input Name $ Test1 Test2 Test3;  
  Total = sum(Test1, Test2, Test3);  
  datalines;  
Asha 80 85 90  
Ravi 70 75 80  
Meena 90 95 92  
;  
run;  
  
proc print data=marks;  
run;
```

Example: MEAN Function

```
data marks;  
  input Name $ Test1 Test2 Test3;  
  Average = mean(Test1, Test2, Test3);  
  datalines;  
Asha 80 85 90  
Ravi 70 75 80  
Meena 90 95 92  
;  
run;  
  
proc print data=marks;  
run;
```

```
data marks;  
  input Name $ Test1 Test2 Test3;  
  Average = mean(Test1, Test2, Test3);  
  datalines;  
Asha 80 85 90  
Ravi 70 75 80  
Meena 90 95 92  
;  
run;  
  
proc print data=marks;  
run;
```

Example: ROUND Function

```
data values;  
  input x;  
  RoundedValue = round(x, 0.01);  
  datalines;  
23.4567  
45.9876  
12.3456  
;  
run;  
  
proc print data=values;  
run;
```

```
data values;  
  input x;  
  RoundedValue = round(x, 0.01);  
  datalines;  
23.4567  
45.9876  
12.3456  
;  
run;  
  
proc print data=values;  
run;
```

Important SAS Function Rules

1. Function names are followed by parentheses.
2. Arguments are separated by commas.
3. Functions return one value.
4. Functions can be nested.
5. Missing values may affect the result depending on the function.

Example: Nested Functions

VirupakshanK@gmail.com

```
data example;  
  input Name $ Marks1 Marks2 Marks3;  
  RoundedAverage = round(mean(Marks1, Marks2, Marks3), 0.01);  
datalines;  
Asha 80 85 90  
Ravi 70 75 80  
;  
run;  
  
proc print data=example;  
run;
```

VirupakshanK@gmail.com

Example: Nested Functions

```
data example;
  input Name $ Marks1 Marks2 Marks3;
  RoundedAverage = round(mean(Marks1, Marks2, Marks3), 0.01);
  datalines;
Asha 80 85 90
Ravi 70 75 80
;
run;

proc print data=example;
run;
```

```
data example;
  input Name $ Marks1 Marks2 Marks3;
  RoundedAverage = round(mean(Marks1, Marks2, Marks3), 0.01);
  datalines;
Asha 80 85 90
Ravi 70 75 80
;
run;

proc print data=example;
run;
```

In this example:

1. `mean(Marks1, Marks2, Marks3)` calculates the average.
2. `round(..., 0.01)` rounds the average to two decimal places.

4.5 Converting Data with Functions

Data conversion is common in SAS because raw data may not always be in the correct type.

SAS has two main data types:

Data Type	Meaning
Numeric	Numbers used for calculations
Character	Text values

INPUT Function

The **INPUT** function converts character values to numeric values.

Syntax

```
numeric_variable = input(character_variable, informat.);
```

Example: Character to Numeric

```
data convert1;
  input ScoreChar $;
  ScoreNum = input(ScoreChar, 8.);
  datalines;
85
90
78
;
run;

proc print data=convert1;
run;
```

```
data convert1;
  input ScoreChar $;
  ScoreNum = input(ScoreChar, 8.);
  datalines;
85
90
78
;
run;

proc print data=convert1;
run;
```

Explanation

- `ScoreChar` is character.
- `ScoreNum` is numeric.
- `input(ScoreChar, 8.)` converts character numbers into numeric values.

PUT Function

The **PUT** function converts numeric values to character values.

Syntax

sas

```
character_variable = put(numeric_variable, format.);
```

Example: Numeric to Character

```
data convert2;
    input ScoreNum;
    ScoreChar = put(ScoreNum, 8.);
    datalines;

85
90
78
;
run;

proc print data=convert2;
run;
```

```
data convert2;
    input ScoreNum;
    ScoreChar = put(ScoreNum, 8.);
    datalines;
85
90
78
;
run;

proc print data=convert2;
run;
```

The **PUT** function
converts numeric values
to character values.

Converting Character Date to SAS Date

VirupakshanK@gmail.com

```
data date_convert;
  input DateChar $10.;
  DateNum = input(DateChar, mmddyy10.);
  format DateNum date9.;
  datalines;
07/20/2026
08/15/2026
09/01/2026
;
run;

proc print data=date_convert;
run;
```

```
data date_convert;
  input DateChar $10.;
  DateNum = input(DateChar, mmddyy10.);
  format DateNum date9.;
  datalines;
07/20/2026
08/15/2026
09/01/2026
;
run;

proc print data=date_convert;
run;
```

Converting SAS Date to Character

```
data date_to_char;  
  input DateNum : date9.;  
  DateChar = put(DateNum, worddate.);  
  datalines;  
20JUL2026  
;  
run;  
  
proc print data=date_to_char;  
run;
```

```
data date_to_char;  
  input DateNum : date9.;  
  DateChar = put(DateNum, worddate.);  
  datalines;  
20JUL2026  
;  
run;  
  
proc print data=date_to_char;  
run;
```

input() vs put() : Common Mistake

VirupakshanK@gmail.com

Memory Tip

INPUT = character to numeric

PUT = numeric to character

More generally:

INPUT reads using an informat.

PUT writes using a format.

VirupakshanK@gmail.com

LECTURE 07 TOPICS

Unit 4 SAS Programming and Automation

4.1	Reading Date and Time with Informat
4.2	Displaying Date and Time Values with Formats
4.3	Basics of SAS Functions
4.4	SAS Function Syntax
4.5	Converting Data with Functions
4.6	The MEANS Procedure
4.7	Specifying Descriptive Statistics
4.8	Limiting Decimal Places
4.9	PROC FREQ Procedure
4.10	Two-Way and N-Way Tables
4.11	LIST Options
4.12	Output Delivery System (ODS)
4.13	HTML Outputs
4.14	PDF Outputs
4.15	RTF Outputs
4.16	EXCEL Outputs

4.1	Reading Date and Time with Informat
4.2	Displaying Date and Time Values with Formats
4.3	Basics of SAS Functions
4.4	SAS Function Syntax
4.5	Converting Data with Functions

LECTURE 07

21CSE295T

Statistical Data Analytics And Integration

Doubts?

THANK YOU!

Virupakshan K

+91-9840847872 | VIRU@KART.LA