Function Categories

Depending on their usage, the functions in SAS are categorised as below.

* **Mathematical**
* **Date and Time**
* **Character**
* **Truncation**
* **Miscellaneous**

Syntax

The general syntax for using a function in SAS is as below.

FUNCTIONNAME(argument1, argument2...argumentn)

Here the argument can be a constant, variable, expression or another function.

Mathematical Functions

These are the functions used to apply some mathematical calculations on the variable values.

Examples

The below SAS program shows the use of some important mathematical functions.

data Math\_functions;

v1=21; v2=42; v3=13; v4=10; v5=29;

/\* Get Maximum value \*/

max\_val = MAX(v1,v2,v3,v4,v5);

/\* Get Minimum value \*/

min\_val = MIN (v1,v2,v3,v4,v5);

/\* Get Median value \*/

med\_val = MEDIAN (v1,v2,v3,v4,v5);

/\* Get a random number \*/

rand\_val = RANUNI(0);

/\* Get Square root of sum of the values \*/

SR\_val= SQRT(sum(v1,v2,v3,v4,v5));

A = SUM(V1,V2,V3);

proc print data = Math\_functions noobs;

run;

proc print data = Math\_functions (OBS=5);

run;

Date and Time Functions

These are the functions used to process date and time values.

Examples

The below SAS program shows the use of date and time functions.

data date\_functions;

INPUT @1 date1 date9. @11 date2 date9. @@;

format date1 date9. date2 date9.;

/\* Get the interval between the dates in years\*/

Years\_ = INTCK('YEAR',date1,date2);

/\* Get the interval between the dates in months\*/

months\_ = INTCK('MONTH',date1,date2);

/\* Get the week day from the date\*/

weekday\_ = WEEKDAY(date1);

/\* Get Today's date in SAS date format \*/

today\_ = TODAY();

/\* Get current time in SAS time format \*/

time\_ = time();

DATALINES;

21OCT2000 16AUG1998

01MAR2009 11JUL2012

;

proc print data = date\_functions noobs;

run;

Character Functions

These are the functions used to process character or text values.

Examples

The below SAS program shows the use of character functions.

data character\_functions;

/\* Convert the string into lower case \*/

lowcse\_ = LOWCASE('HELLO');

/\* Convert the string into upper case \*/

upcase\_ = UPCASE('hello');

/\* Reverse the string \*/

reverse\_ = REVERSE('Hello');

/\* Return the nth word \*/

nth\_letter\_ = SCAN('Learn SAS Now',2);

run;

proc print data = character\_functions noobs;

run;

Truncation Functions

These are the functions used to truncate numeric values.

Examples

The below SAS program shows the use of truncation functions.

data trunc\_functions;

/\* Nearest greatest integer \*/

ceil\_ = CEIL(11.85);

/\* Nearest greatest integer \*/

floor\_ = FLOOR(11.85);

/\* Integer portion of a number \*/

int\_ = INT(32.41);

/\* Round off to nearest value \*/

round\_ = ROUND(5621.78);

run;

proc print data = trunc\_functions noobs;

run;

Miscellaneous Functions

Examples

The below SAS program shows the use of Miscellaneous functions.

data misc\_functions;

/\* Nearest greatest integer \*/

state2=zipstate('01040');

/\* Amortization calculation \*/

payment=mort(50000, . , .10/12,30\*12);

proc print data = misc\_functions noobs;

run;

SAS - Input Methods

Below are different input methods available in SAS.

* **List Input Method**
* **Named Input Method**
* **Column Input Method**
* **Formatted Input Method**

The details of each input method is described as below.

**List Input Method**

In this method the variables are listed with the data types. The raw data is carefully analysed so that the order of the variables declared matches the data. The delimiter (usually space) should be uniform between any pair of adjacent columns. Any missing data will cause problem in the output as the result will be wrong.

**Example**

The following code and the output shows the use of list input method.

DATA TEMP;

INPUT EMPID ENAME $ DEPT $ ;

DATALINES;

1 Rick IT

2 Dan OPS

3 Tusar IT

4 Pranab OPS

5 Rasmi FIN

;

PROC PRINT DATA=TEMP;

RUN;

## Named Input Method

In this method the variables are listed with the data types. The raw data is modified to have variable names declared in front of the matching data. The delimiter (usually space) should be uniform between any pair of adjacent columns.

### Example

The following code and the output show the use of Named Input Method.

DATA TEMP;

INPUT

EMPID= ENAME= $ DEPT= $ ;

DATALINES;

EMPID=1 ENAME= Rick DEPT= IT

EMPID=2 ENAME= Dan DEPT= OPS

EMPID=3 ENAME= Tusar DEPT= IT

EMPID=4 ENAME= Pranab DEPT= OPS

EMPID=5 ENAME= Rasmi DEPT= FIN

;

PROC PRINT DATA=TEMP;

RUN;

## Column Input Method

In this method the variables are listed with the data types and width of the columns which specify the value of the single column of data. For example if an employee name contains maximum 9 characters and each employee name starts at 10th column, then the column width for employee name variable will be 10-19.

### Example

Following code shows the use of Column Input Method.

DATA TEMP;

INPUT EMPID 1-3 ENAME $ 5-12 DEPT $ 13-16;

DATALINES;

14 Rick IT

241Dan OPS

30 Sanvi IT

410Chanchal OPS

52 Piyu FIN

;

PROC PRINT DATA=TEMP;

RUN;

## Formatted Input Method

In this method the variables are read from a fixed starting point until a space is encountered. As every variable has a fixed starting point, the number of columns between any pair of variables becomes the width of the first variable. The character '@n' is used to specify the starting column position of a variable as the nth column.

### Example

The following code shows the use of Formatted Input Method

DATA TEMP;

INPUT @1 EMPID $ @4 ENAME $ @13 DEPT $ ;

DATALINES;

14 Rick IT

241 Dan OPS

30 Sanvi IT

410 Chanchal OPS

52 Piyu FIN

;

PROC PRINT DATA=TEMP;

RUN;

SAS - Date Times

SAS Date Informat

The source data can be read properly by using specific date informats as shown below. The digit at the end of the informat indicates the minimum width of the date string to be read completely using the informat. A smaller width will give incorrect result. with SAS V9, there is a generic date format **anydtdte15.** which can process any date input.

|  |  |  |
| --- | --- | --- |
| **Input Date** | **Date width** | **Informat** |
| 03/11/2014 | 10 | mmddyy10. |
| 03/11/14 | 8 | mmddyy8. |
| December 11, 2012 | 20 | worddate20. |
| 14mar2011 | 9 | date9. |
| 14-mar-2011 | 11 | date11. |
| 14-mar-2011 | 15 | anydtdte15. |

DATA NEWDATA;

INPUT dob mmddyy10. Name$;

Datlines;

/…

;

//output

Proc print…;

Format dob…..;

Example

The below code shows the reading of different date formats. Please note the all the output values are just numbers as we have not applied any format statement to the output values.

DATA TEMP;

INPUT @1 Date1 date11. @12 Date2 anydtdte15. @23 Date3 mmddyy10. ;

DATALINES;

02-mar-2012 3/02/2012 3/02/2012

;

PROC PRINT DATA=TEMP;

RUN;

SAS Date output format

The dates after being read , can be converted to another format as required by the display. This is achieved using the format statement for the date types. They take the same formats as informats.

Example

In the below exampel the date is read in one format but displayed in another format.

DATA TEMP;

INPUT @1 DOJ1 mmddyy10. @12 DOJ2 mmddyy10. ;

format DOJ1 date11. DOJ2 worddate20. ;

DATALINES;

01/12/2012 02/11/1998

;

PROC PRINT DATA=TEMP;

RUN;

SAS - Chi Square

A chi-square test is used to examine the association between two categorical variables. It can be used to test both extent of dependence and extent of independence between Variables. SAS uses **PROC FREQ** along with the option **chisq** to determine the result of Chi-Square test.

Syntax

The basic syntax for applying PROC FREQ for Chi-Square test in SAS is:

PROC FREQ DATA = dataset;

TABLES variables

/CHISQ TESTP=(percentage values);

Following is the description of the parameters used:

* **Dataset** is the name of the dataset.
* **Variables** are the variable names of the dataset use in chi-square test.
* **Percentage Values** in the TESTP statement represent the percentage of levels of the variable.

Example

In the below example we consider a chi-square test on the variable named type in the dataset SASHELP.CARS. This variable has six levels and we assign percentage to each level as per the design of the test.

proc freq data = sashelp.cars;

tables type

/chisq

testp=(0.20 0.12 0.18 0.10 0.25 0.15);

run;

Two Way chi-square

Two way Chi-Square test is used when we apply the tests to two variables of the dataset.

Example

In the below example we apply chi-square test on two variables named type and origin. The result shows the tabular form of all combinations of these two variables.

proc freq data = sashelp.cars;

tables type\*origin

/chisq

;

run;

SAS - One Way Anova

ANOVA stands for Analysis of Variance. In SAS it is done using **PROC ANOVA**. It performs analysis of data from a wide variety of experimental designs. In this process, a continuous response variable, known as a dependent variable, is measured under experimental conditions identified by classification variables, known as independent variables. The variation in the response is assumed to be due to effects in the classification, with random error accounting for the remaining variation.

Syntax

The basic syntax for applying PROC ANOVA in SAS is:

PROC ANOVA dataset ;

CLASS Variable;

MODEL Variable1=variable2 ;

MEANS ;

Following is the description of the parameters used:

* **dataset** is the name of the dataset.
* **CLASS** gives the variables the variable used as classification variable.
* **MODEL** defines the model to be fit using certain variables from the dataset.
* **Variable\_1 and Variable\_2** are the variable names of the dataset used in analysis.
* **MEANS** defines the type of computation and comparison of means.

Applying ANOVA

Example

Lets consider the dataset SASHELP.CARS. Here we study the dependence between the variables car type and their horsepower. As the car type is a variable with categorical values, we take it as class variable and use both these variables in the MODEL.

PROC ANOVA DATA = SASHELPS.CARS;

CLASS type;

MODEL horsepower = type;

RUN;

Applying ANOVA with MEANS

Example

We can also extend the model by applying the MEANS statement in which we use Turkey's Studentized method to compare the mean values of various car types. The category of car types are listed with the mean value of horsepower in each category along with some additional values like error mean square etc.

PROC ANOVA DATA = SASHELPS.CARS;

CLASS type;

MODEL horsepower = type;

MEANS type / tukey lines;

RUN;

SAS - Hypothesis Testing

Hypothesis testing is the use of statistics to determine the probability that a given hypothesis is true. The usual process of hypothesis testing consists of four steps as shown below.

Step-1

Formulate the null hypothesis H0 (commonly, that the observations are the result of pure chance) and the alternative hypothesis H1 (commonly, that the observations show a real effect combined with a component of chance variation).

Step-2

Identify a test statistic that can be used to assess the truth of the null hypothesis.

Step-3

Compute the P-value, which is the probability that a test statistic at least as significant as the one observed would be obtained assuming that the null hypothesis were true. The smaller the P-value, the stronger the evidence against the null hypothesis.

Step-4

Compare the p-value to an acceptable significance value alpha (sometimes called an alpha value). If p <=alpha, that the observed effect is statistically significant, the null hypothesis is ruled out, and the alternative hypothesis is valid.

SAS programming language has features to carry out various types of hypothesis testing as shown below.

|  |  |  |
| --- | --- | --- |
| **Test** | **Description** | **SAS PROC** |
| **T-Test** | A t-tests is used to test whether the mean of one variable is significantly different than a hypothesized value. We also determine whether means for two independent groups are significantly different and whether means for dependent or paired groups are significantly different. | **PROC TTEST** |
| **ANOVA** | It is also used to compare means when there is one independent categorical variable. We want to use one-way ANOVA when testing to see if the means of the interval dependent variable are different according to the independent categorical variable. | **PROC ANOVA** |
| **Chi-Square** | We use chi square goodness of fit to assess if frequencies of a categorical variable were likely to happen due to chance. Use of a chi square test is necessary whether proportions of a categorical variable are a hypothesized value. | **PROC FREQ** |
| **Linear Regression** | Simple linear regression is used when one wants to test how well a variable predicts another variable. Multiple linear regression allows one to test how well multiple variables predict a variable of interest. When using multiple linear regression, we additionally assume the predictor variables are independent. | **PROC REG** |

What is the difference between Missover and Truncover?

Use notepad to save the following lines as a text file labeled emplist.txt:

LANGKAMM SARAH E0045 Mechanic

TORRES JAN E0029 Pilot

SMITH MICHAEL E0065

LEISTNER COLIN E0116 Mechanic

TOMAS HARALD

WADE KIRSTEN E0126 Pilot

WAUGH TIM E0204 Pilot

then run the following:

DATA test1;

INFILE "/folders/myfolders/emplist.txt" missover;

INPUT lastn $1-21 Firstn $ 22-31

Empid $32-36 Jobcode $37-45;

RUN;

DATA test2;

INFILE "/folders/myfolders/emplist.txt" truncover;

INPUT lastn $1-21 Firstn $ 22-31

Empid $32-36 Jobcode $37-45;

RUN;