

एस.ए.एस. मैक्रो की शुरुआत कैसे करे

राजेन्द्र सिंह तोमर एवं प्रमोद कुमार

सारांश

एस.ए.एस. मैक्रो (SAS Macro) प्रोग्रामिंग को साधारणतः एक उन्नत (Advance) विषय समझा जाता है, निःसन्देह एस.ए.एस. मैक्रो तैयार करना एक चुनौती पूर्ण कार्य है परन्तु यह कटुसत्य है कि इसकी आधारभूत अवधारणा (Basic Concept) काफी आसान है । मैक्रो तैयार करने से पूर्व एस.ए.एस. (सैस) कोड विकसित करना आना एक आवश्यकता है क्योंकि सैस (SAS) प्रोग्रामिंग भाषा संगणना के लिए एक उपकरण की भाँति कार्य करता है जो कि प्रयोगकर्ता को अधिक शक्तिशाली बना देती है । इस लेख्य के माध्यम से हम सैस प्रयोगकर्ता का परिचय मैक्रो तैयार करने की कुछ सामान्य आवश्यक तकनीकों से कराना चाहते हैं जिसमें मैक्रो तैयार करने के लिए मैक्रो चरों (Macro Variable) एवं मैक्रो प्रोग्रामिंग के आवश्यक तत्व को उद्घृत किया गया है ।

परिचय:

एस.ए.एस. मैक्रो एक टैक्स्ट (Text) है जिसके अन्दर सैस प्रोग्राम को कहीं पर भी डाला व विस्तृत किया जा सकता है । मैक्रो कोड को तैयार करने में साधारण सैस कोड से कुछ अधिक समय लगता है क्योंकि सैस कोड, मैक्रो का एक हिस्सा है लेकिन मैक्रो तैयार करने के बाद प्रयोगकर्ता को समय बचाने का लाभ मिलता है जिसे वह अन्य कार्य करने के लिए प्रयोग कर सकता है । दूसरा मैक्रो तैयार करने से यह सुविधा मिलती है कि एक समान आँकड़ों जिनसे एक ही तरह के साँख्याकीय परिणाम अथवा विश्लेषण प्राप्त करने है, के लिए मैक्रो प्रोग्राम का प्रयोग बार-बार कर सकते हैं । एक विशेष प्रकार के आँकड़ों के प्रबन्धन एवं विश्लेषण के लिए मैक्रो तैयार कर, उसी मैक्रो का प्रयोग, उचित परिवर्तन के साथ अन्य आँकड़ों के प्रबन्धन एवं विश्लेषण के लिए भी किया जा सकता है ।

परम्परागत सैस (SAS) प्रोग्रामिंग के माध्यम से हम मानक मान (default value) की स्थापना करके, एक सैस कोड पद से अन्य सैस कोड पद के साथ कम्युनिकेशन टैक्स्ट (communication text) को कथन (statement) के रूप में प्रतिस्थापित (substitute), आदि नहीं कर पाते हैं । इसके बावजूद परम्परागत सैस (SAS) प्रोग्रामिंग के अन्तर्गत प्रत्येक पद की गणना व निष्पादन स्वतन्त्र रूप (step independently compute and execute) से होता है । मैक्रो तैयार करने से परम्परागत सैस प्रोग्रामिंग से उत्पन्न बाधाओं को दूर किया जा सकता है क्योंकि मैक्रो कथन को मैक्रो प्रोसेसर के पास संकलक (compiler) से पूर्व पास किया जाता है । द्वितीय मैक्रो प्रोसेसर स्ट्रिंग को जोड़-जोड़ कर (manipulate) कर स्कैनिंग के लिए वापस भेजता है प्रायः मैक्रो स्ट्रिंग, सैस कोड होता है जो कि मैक्रो प्रोग्राम के अन्दर एक वर्ण (characters) की श्रेणी में आता है ।

मैक्रोचर (Macro variable)

मैक्रो चर का मान (value) सदैव एक टैक्स्ट होता है । टैक्स्ट कोई सैस आंकड़ा नहीं होता बल्कि सैस प्रोग्राम का एक हिस्सा होता है । चर को, सन्दर्भ चर (Reference Variable) बनाने के लिए, चर के नाम के आगे & (एम्प्रसेन्ड) का चिह्न लगाकर किया जाता है । यह & (एम्प्रसेन्ड) का चिह्न मैक्रो चर के मान को प्रदर्शित करता है । % (Percentage) चिह्न का प्रयोग मैक्रो व मैक्रो फलन (functions) को प्रारम्भ करने के लिए किया जाता है ।

मैक्रोचर दो प्रकार के होते हैं (i) स्थानीय (local) (ii) सार्वभौम (global), जब कोई चर, मैक्रो के अन्दर परिभाषित किया जाये उसे स्थानीय चर तथा जिस चर को मैक्रो के बाहर खुले में परिभाषित किया जाता है वह सार्वभौम चर कहलाता है । इसके बावजूद बहुत सारे सिस्टम मैक्रो चर हैं । जिनका प्रयोग मैक्रो तैयार करते समय आवश्यकता अनुसार किया जा सकता है ।

मैक्रो बनाम मैक्रो चर

मैक्रो प्रोग्राम तैयार करने के लिए मैक्रो एवं मैक्रो चर दो मुख्य आधार हैं । यह दोनों आपस में भिन्न हैं क्योंकि मैक्रोचर एम्प्रसेन्ड (&) से प्रारम्भ होता है जबकि मैक्रो प्रतिषत (%) चिहन् से शुरु होता है । मैक्रो चर एक मानक डाटा चर (Standard data variable) g`rk gS जो कि सदैव वर्ण की तरह कार्य करता है और इसका डेटा सेट से सम्बन्ध नहीं होता लेकिन जिसका यह मैक्रो प्रोग्राम में मान होता है उसकी स्थापना का कार्य प्रोग्राम निष्पादन के समय अवश्य करता है । मैक्रोचर का मान, कोई चर का नाम, कोई संख्या या फिर टैक्स्ट हो सकती है जिसको हम किसी अन्य मान के स्थान पर स्थापन के तौर पर प्रयोग करते हैं ।

मैक्रो, प्रोग्राम का बड़ा हिस्सा होता है जिसमें जटिल तर्क (logic) होता है जिसमें सम्पूर्ण ऑकड़े और PROC के पग (step) सम्मिलित करने के साथ हम % IF, % THEN, % ELSE, % DO एवं % END प्रयोग भी करते हैं ।

स्वचालित (Automatic) ;k ra= (System) मैक्रो कैसे प्रयोग करें

सामान्यतः इन स्वचालित मैक्रो का प्रयोग हम तब करते जब हम अपनी रिपोर्ट में तंत्र की जानकारी प्रस्तुत करना चाहते हैं या तत्काल ही तंत्र के बारे में जानना चाहते हैं । जैसे की हम अपनी परियोजना के अन्तर्गत एकत्र किये गये आंकड़ों की तिथि और समय प्रदर्शित करें या फिर हम वेतन बिल में यह दर्शाए कि बिल कब और किस समय तैयार हुआ तब हम SYSDATE O SYSTIME को कोड में डाल देंगे ।।

स्वचालित (Automatic) मैक्रोचर की आंशिक सूची

SYSDATE	→	वर्तमान तिथि प्रदर्शित करने हेतु
SYSTIME	→	वर्तमान समय प्रदर्शित करने हेतु
SYSPROD	→	SAS उत्पाद का लाइसेन्स प्रदर्शित करता है
SYSVER	→	SAS Version देखने हेतु
SYSSCP	→	Operating system जिसचर
SYSDEVICE	→	वर्तमान ग्राफिक Deviceआदि

सभी वर्तमान स्वचालित चर को देखने के लिए % Put _All_ का प्रयोग कर सकते हैं । प्रोग्राम एडिटर में इसे टाईप कर जैसे ही हम submit करते हैं लॉग (Log) विन्डो में ग्लोबल एवं आटोमेटिक सभी चर प्रदर्शित हो जाते हैं जिनका प्रयोग हम आपकी आवश्यकता अनुसार कर सकते हैं ।

मैक्रो चर कैसे परिभाषित करें

मैक्रो चर को परिभाषित करने के लिए % LET प्रयोग किया जाता है परिभाषित करने का सामान्य कथन निम्न प्रकार है :-

% LET चर का नाम = मान , उदाहरणार्थ %LET city=Delhi

यहाँ city मैक्रो चर है तथा Delhi मैक्रो चर का मान है ।

जब हमें कोई प्रोग्राम सप्ताह अथवा माह में आंकड़ों के प्रबन्धन या विप्लेषण के लिए, कई बार प्रयोग करना होता है तब बार-2 परिवर्तन से बचने के लिए हम मैक्रोचर को परिभाषित कर लेते हैं । % LET सीधे-2 एक मैक्रो है जो की मैक्रोचर का मान निर्दिष्ट (value assign) करता है । % LET एक ऐसा कथन है जो प्रोग्राम के लचीलेपन में वृद्धि करता है क्योंकि जब हमें आंकड़ों के प्रबन्ध और विप्लेषण हेतू किसी प्रोग्राम का प्रयोग बार-2 करना पड़ता है तो विशेष रूप से विकसित किये गये इस मैक्रो के % LET कथन में आवश्यक परिवर्तन कर, कार्य को आसानी से कर सकते हैं । % LET में परिवर्तन करते ही उस परिवर्तन की पुनरावृत्ति पूरे प्रोग्राम में स्वतः हो जाती है । मैक्रो चर का नाम 32 अक्षर (character) तक हो सकता है चर के नाम का प्रथम अक्षर, वर्णमाला से हो या फिर अन्डरस्कोर (underscane) होना चाहिए उसके बाद अक्षर, संख्या या फिर अन्डरस्कोर आदि कुछ भी हो सकते हैं ।

उदाहरण 1 : माना की अन्वेषक ने अलग-अलग क्षेत्र मे फसल प्रयोग आयोजित किये जिसके आंशिक आँकडे निम्न प्रकार है ।

Zone	Crop	Season	Variety	Yield	Income
A	Paddy	Kh	Radha	1815	34485
A	Wheat	Rb	Sonalika	1350	20250
A	Groundnut	Kh	HR-1	780	19500
B	Wheat	Rb	Kranti	1420	21300
B	Groundnut	Kh	HR-10	795	19875
C	Paddy	Kh	Radhaq	1750	33250
C	Groundnut	Kh	HR-1	801	20025
C	Wheat	Rb	Kranti	1400	21000

हम इन ऑकडो से विभिन्न सूचनाए प्राप्त कर विश्लेषण करना चाहता है । ऑकडो को सैस डेटा-सेट मे परिवर्तित करने के लिए हम इन ऑकडो को सीधे सैस एडिटर विन्डो अथवा Infile statement अथवा Proc Import Statement का प्रयोग कर सकते है । माना हम उपरोक्त ऑकडो का एक डेटा से सीधे सैस एडिटर विन्डो मे zone नाम से बनाते है जो निम्न प्रकार है:-

Data zone;

Input zone crop season variety yield Income;

datalines;

(यहा पर ऑकडो की प्रविष्टि होगी)

;

अब हम मैक्रो चर season नाम से परिभाषित कर ऑकडो को अवलोकन (view) करने के सैस कोड मे परिवर्तन करके, निम्न प्रकार मैक्रो प्रोग्राम का एक हिस्सा तैयार करते हैं

% LET season = kh ;

Proc Print = Zone;

Where Sn = "& season";

Title "Zone wise Production & Income"

Run;

उपरोक्त प्रोग्राम में मैक्रोचर का प्रयोग सिर्फ एक बार हुआ लेकिन बड़े आंकडों के लिए बड़ा प्रोग्राम तैयार करने पर एक ही मैक्रोचर का प्रयोग कई बार हो सकता है । जैसे ही हम प्रोग्राम को रन करेंगे मैक्रो प्रोसेसर, मैक्रो चर के साथ निम्न सैस कोड स्वतः तैयार कर लेगा ।

Proc Print data=zone;

Title "Zone wise Production & Income"

Run;

मैक्रो माडुलर कोड

मैक्रो तैयार करने के लिये दो बातें आवश्यक हैं प्रथम हम मैक्रो को प्रारम्भ कैसे करें और मैक्रो को समाप्त कैसे करें । सैस प्रोग्राम एडिटर में % MACRO कथन लिखने से प्रोग्राम प्रस्तुत (submit) करते ही मैक्रो प्रोसेसर को पता पड़ जाता है कि मैक्रो प्रारम्भ हो रहा है अन्तिम पंक्ति में % MEND कथन संकेत देता है कि मैक्रो समाप्त हो रहा है । इन दोनों कथन के बीच सैस के कई कथन आवश्यकता के अनुसार लिखे जा सकते हैं तथा कई मैक्रोचरों को उत्पन्न किया जा सकता है । मैक्रो माडुलर का सामान्य रूप इस प्रकार होता है ।

% MACRO मैक्रो का नाम ;
(सैस कथन)

% MEND मैक्रो का नाम ;
% मैक्रो का नाम

उदाहरण 2 : उपरोक्त zone आंकड़ों से अन्वेषक सीजन के आधार पर विभिन्न समय पर फसलों की सूची प्राप्त करना चाहता है जिसमें उस फसल पैदावार के आंकड़े भी प्रदर्शित हो । निम्न प्रोग्राम में हम एक मैक्रो LIST नाम से तैयार करते हैं जिसमें बिना छंटनी के रिपोर्ट तैयार करने के लिए PROC PRINT कथन का प्रयोग करेंगे फिर छंटनी के लिए PROC SORT का प्रयोग करते हैं अतः हमें मैक्रोचर LIST का प्रयोग दो बार करना होगा । मैक्रो के अन्दर शीर्षक (Title) सदैव दोहरे उद्धरण (Double Quote) के अन्दर होना आवश्यक है ।

```
% MACRO list;  
Proc Print data = zone NOOBS;  
Title, season by Production of the crops;  
Var crop season yield;  
Run;  
% Mend list;  
Proc Sort Dat a = zone'  
By Season;  
% list
```

% list कथन मैक्रो निष्पादन के लिखना आवश्यक है अन्यथा परिणाम आउट-पुट विन्डो में प्रदर्शित नहीं हो पाएंगे । इस मैक्रो को रन करने पर मैक्रो प्रोसेसर निम्न सैस कोड आन्तरिक रूप से तैयार कर आँकड़ों का परिणाम प्रस्तुत करगा ।

```
PROC PRINT DATA = Zone ;
```

```

Title "Seasonwise production of crops';
Var Crop in yield';
Run'
Proc Sort Data = Zone;
By Season;
PROC PRINT DATA = Zone noobs;
Title
Var, crop in yield;
Run;

```

उपरोक्त की भाँति सभी मैक्रो प्रोग्राम को सैस प्रोसेसर स्वतः सैस कोड में परिवर्तित कर परिणाम आउट-पुट विन्डो में प्रदर्शित करता है।

हम आवश्यकता अनुसार असंख्य मैक्रो चरों को मैक्रो के बाहर या अन्दर परिभाषित कर सकते हैं तथा इन चरों का उपयोग ऑकड़ों के प्रबन्धन एवं विश्लेषण के लिए, मैक्रो अन्दर लिखे जा रहे सैस कोड में कई बार कर सकते हैं। जिसे निम्न स्प्लिट प्लॉट विश्लेषण मैक्रो के द्वारा स्पष्ट किया गया है

उदाहरण 3 : स्प्लिट प्लॉट मैक्रो में मैक्रोचर % let dataset का प्रयोग डाटा सेट splitplot का नाम पढ़ाने के लिये किया गया है जो कि एक ग्लोबल मैक्रोचर है। तत्पश्चात् %let dependent का प्रयोग दो आश्रित चरों yield व s_yield को पढ़ाने के लिए किया गया है चूँकि यहाँ पर एक से अधिक आश्रित चर को लिखे गए हैं इन्हें अलग-अलग करने के लिए अंतराल (space) का प्रयोग किया गया है। अतः स्पष्ट है कि एक से अधिक चरों का विश्लेषण किया जा सकता है। इसी प्रकार % let block, % let main yield S_yield एवं % let Sub, % let comparision एवं % let alpha का प्रयोग क्रमशः पुनरावृत्ति (Replication), मुख्य प्लॉट mplot, तथा सबप्लॉट splot, गुणज तुलना (tukey) तथा सिग्नीफिकेन्स लेवल (0.05) को पढ़ने के लिए किया गया है। यह सभी स्थानीय चर हैं।

```

%let dataset = splitplot;
%macro splitplot ;
%let dependent = yield s_yield;
%let block = replication;
%let main = mplot;
%let sub = splot;
%let comparision=tukey;
%let alpha = 0.01;
Proc GLM Data=&dataset;
    class &block &main &sub ;
    model &dependent = &block &main &block*&main
    &sub &main*&sub ;

```

```

test h=&block &main e=&block*&main;
means &main &sub &main*&sub/&comparosion alpha=&alpha;
Run;
Quit;
%mend splitplot;
%splitplot ;

```

उपरोक्त मैक्रो से स्पष्ट है कि हम एक मैक्रो चर को एक से अधिक बार प्रयोग कर सकते हैं । अब यदि हमारे पास एक से अधिक, एक ही तरह के डाटा सेट हैं तो हमें नया मैक्रो प्रोग्राम बनाने की आवश्यकता नहीं है । हमें केवल नये डाटा सेट का नाम मैक्रोचर % data set = नया नाम, लिखकर मैक्रो को पुनः चला देते हैं ।

मैक्रो में मापदण्डों (Parameters) का प्रयोग

मैक्रो पैरामीटर अथवा प्राचल एक स्थानीय चर को परिभाषित करने का दूसरा तरीका है जो प्रोग्राम निष्पादन के दौरान तभी मौजूद होता है जब प्रयोगकर्ता उसे प्रोग्राम निष्पादन कथन पंक्ति में परिभाषित करता है । प्राचलों (Parameters) का प्रयोग करने के अनेक लाभ हैं । पहला यह कि कम से कम % LET कथन का प्रयोग करना पड़ता है, दूसरा प्राचलों का उपयोग यह सुनिश्चित करता है कि प्रोग्राम का जो हिस्सा मैक्रो से बाहर लिखा है उसका चरों के साथ कभी कोई हस्तक्षेप (Interfere) नहीं होता है। तीसरा लाभ यह कि मैक्रो प्राचल हमें मैक्रो में सूचना पारित करने के लिए समर्थ (enable) बना देते हैं ।

मैक्रो प्रोग्राम में हम % Macro कथन के तुरन्त बाद कोष्ठक में जिन मैक्रो चरों को परिभाषित करते हैं वह चर मैक्रो पैरामीटर कहलाते हैं।

```
%macro मैक्रोनाम (पैरामीटर1=मान1, पैरामीटर2=मान2, पैरामीटर3=मान3, ) ;
```

```

-----
(आवश्यक सैस कोड)
-----

```

```
%mend मैक्रोनाम ;
```

```
% मैक्रोनाम (पैरामीटर1=मान1, पैरामीटर2=मान2, पैरामीटर3=मान3, )
```

मैक्रो का आह्वान (invoke) करने से पूर्व हम अन्तिम पंक्ति में, कोष्ठक में, ऊपर परिभाषित उन प्राचलों के मान को लिखते हैं जिनका विश्लेषण प्राप्त करना है। जब मैक्रो निष्पादित होता है, मैक्रो प्रोसेसर, प्राचलों को निर्दिष्ट मूल्य का मेल करवाता है और आन्तरिक रूप से सैस कोड तैयार कर परिणाम देता है।

उदाहरण 4: मैक्रो पैरामीटर के उपयोग को स्पष्ट करने के लिए हमने डिजाइन रिसोर्स सर्वर ऑन लाइन एनेलेसिस से ऑकडो का एक उदाहरण प्राप्त किया है जिसमें लौकी के संकर बीज उत्पादन (Hybrid Seed Production of Bottle Guard) के आंकड़े प्रस्तुत किये गये हैं । यह ध्यान में रखते हुए कि इस प्रकार के प्रयोग प्रायः संचालित (Conduct) किये जाते हैं और उनके वर्णनात्मक सांख्यिकीय विप्लेषण भी प्रायः किये जाते हैं निम्न मैक्रो प्रोग्राम तैयार किया है :

```
options nodate ;
data des;
input group fs45 fw syp sl;
label fs45 = "No. of fruit Set (45days)";
label fw = "Fruit weight (kg)";
label syp = "Seed yield/plant (g)";
label sl = "Seedling length (cm)";
cards;
1 7.0 1.85 147.70 16.86
1 7.0 1.86 136.86 16.77
1 6.0 1.83 149.97 16.35
1 7.0 1.89 172.33 18.26
1 7.0 1.80 144.46 17.90
1 6.0 1.88 138.30 16.95
1 7.0 1.89 150.58 18.15
1 7.0 1.79 140.99 18.86
1 6.0 1.85 140.57 18.39
1 7.0 1.84 138.33 18.58
2 6.3 2.58 224.26 18.18
2 6.7 2.74 197.50 18.07
2 7.3 2.58 230.34 19.07
2 8.0 2.62 217.05 19.00
2 8.0 2.68 233.84 18.00
2 8.0 2.56 216.52 18.49
2 7.7 2.34 211.93 17.45
2 7.7 2.67 210.37 18.97
2 7.0 2.45 199.87 19.31
2 7.3 2.44 214.30 19.36
;
```

```
%global class var;
options symbolgen mprint mcomple;

%macro testing (data=des, var=fs45 fw syp sl, class=group) ;
proc summary data=&data mean median skewness kurt maxdec=4 print;
title " Descriptive Statistics :mean, median, coefficient of skewness, coefficient of kurtosis of all
characters for each of the group";
class &class;
```



```
var &var;  
run;
```

```
proc univariate data=&data normal;  
title "Testing the normality of data for all the characters and for each of the two groups";  
class &class;  
var &var;  
run; quit;
```

```
proc ttest data=&data;  
class &class;  
var &var;  
run; quit;  
%mend testing ;  
%testing (data=des , var=syp fw, class=group);
```

यहां पर मैक्रो testing के नाम से तैयार किया गया है इसके तुरन्त बाद कोष्ठक में data, var और class नाम के तीन मैक्रो चर परिभाषित किये गये हैं जिनका मान बराबर के चिह्न के बाद प्रदर्शित किया गया है । प्रारम्भ में सभी उपलब्ध चरों fs45, fw, syp व sl को परिभाषा में डाल दिया गया है ताकि हम सभी अथवा आवश्यकता अनुसार कुछ ही चरों के परिणाम प्राप्त कर सकें। मैक्रो की निष्पादन पंक्ति %testing में हम उन सभी चरों को परिभाषित करते हैं जिनके हमें परिणाम प्राप्त करने हैं। यहाँ var चर में syp व fw को परिभाषित किया है।

उपरोक्त में %global कथन का प्रयोग कर दो सार्वभौम चर class व var परिभाषित किये हैं ताकि यदि आवश्यक हुआ तो इनका उपयोग मैक्रो के बाहर भी किया जा सके। options कथन में symbolgen mprint mcompile का प्रयोग इसलिए किया गया है कि यह हमें मैक्रो प्रोग्राम को डिबग करने में सहायता करता है। यदि हमारा मैक्रो सही परिणाम नहीं दे रहा है तब हम लागू विन्डो में गलती देख सकते हैं। symbolgen यह बताता है कि मैक्रो चरों का मान संकल्पित (Resolve) हुए या नहीं यदि किसी चर का मान संकल्पित नहीं हो रहा तो परिणाम प्राप्त नहीं हो पाता।

उपरोक्त सभी मैक्रो उदाहरण अत्यन्त सरल व साधारण श्रेणी के हैं। उन्नत प्रकार के मैक्रो तैयार अथवा विकसित करने के लिए इन मैक्रो उदाहरण से अनुभव प्राप्त कर इसमें उन्नत मैक्रो नियन्त्रकों (Commands) का प्रयोग करके विकसित किये जा सकते हैं।

सन्दर्भ

1. <http://support.sas.com/documentation/onlinedoc>.
2. <http://www2.sas.com/proceedings/sugi29>.
3. <http://iasri.res.in/design>
4. <http://www.iasri.res.in/sscnars>