ماکرونویسی در SAS

جلسه اول

ماکرو ابزاری برای جایگزینی متن است که این متن می‌تواند یک مقدار ساده یا دستورات پیچیده SAS باشد. در واقع ماکرو برنامه‌ای است که برنامه تولید می‌کند. تعبیر زیبایی برای ماکرو هست که می‌گوید فرض کنید یک کار تکراری را نمی‌خواهید خودتان انجام بدهید و یک‌بار می‌نویسید و به دستیار خود می‌گویید بارها تکرار کند. این دستیار که هرگز اشتباه نمی‌کند همان قابلیت ماکرویی SAS است.

دو تا جز کلی در ماکرو وجود دارد: ۱. متغیر ماکرویی ۲.برنامه ماکرویی

۱. **متغیر ماکرویی:** با استفاده از دستور %let می‌توان یک متغیرماکرویی تعریف و مقداری را به آن منتسب کرد. هرجا به مقدار این متغیر ماکرویی نیاز داشتیم کافی است یک علامت & را به ابتدای نام متغیر ماکرویی اضافه کرد.
۲. **برنامه ماکرویی:** مجموعه‌ای از دستورات و عبارت ماکرویی است که می‌تواند شامل یک متن معمولی یا مجموعه‌ای از دستورات پیچیده‌ی SAS باشد.

شکل کلی یک برنامه ماکرویی به صورت زیر است:

```
%macro نام_برنامه_ماکرویی (فهرست پارامترها (اختیاری است));

عبارت‌های ماکرویی

%mend;
```

برای فراخوانی یک برنامه ماکرویی کافی است به صورت زیر نوشته شود:

```
%نام_برنامه_ماکرویی (فهرست پارامترها )
```

مثال: فرض کنید مثلا یک مجموعه داریم (HEIS) که در آن داده‌های ۳۱ اســتان موجود اســت. می‌خواهیم داده‌های هر استان را در شیت جداگانه در یک فایل اکسل ذخیره کنیم. چه کنیم؟

راه حل اول که به ذهنم می‌رسـد این هسـت که ۳۱ بار برنامه‌ای بنویسم که از مجموعه داده‌ی اصلی، یکی یکی استان‌ها را جدا کرده و یکی یکیexport کند. یعنی باید هر برنامه رو کپی کنم و ۴ بار عدد استان رو عوض کنم:

```
/*  00  */
data HEIS_00;
SET HEIS;
WHERE OSTAN='00';
RUN;
PROC EXPORT DATA= WORK.Heis_00
            OUTFILE= "Desktop\OSTAN\OSTAN.xls"
            DBMS=EXCEL REPLACE;
     SHEET="OST_00";
RUN;

/*  01  */
data HEIS_01;
SET HEIS;
WHERE OSTAN='01';
RUN;
PROC EXPORT DATA= WORK.Heis_00
            OUTFILE= "Desktop\OSTAN\O  STAN.xls"
            DBMS=EXCEL REPLACE;
     SHEET="OST_01";
RUN;

…


/*  30  */
data HEIS_30;
SET HEIS;
WHERE OSTAN='30';
RUN;
PROC EXPORT DATA= WORK.Heis_30
            OUTFILE= "Desktop\OSTAN\OSTAN.xls"
            DBMS=EXCEL REPLACE;
     SHEET="OST_30";
RUN;
```

راه حل دوم: با اســـتفاده از یک **متغیر ماکرویی**، زحمت ۴ بار تغییر کد اســـتان را به یکبار تغییر کاهش می‌دهم. طبق تعریف متغیر ماکرویی، می‌نویسیم:

```
%LET ost = 00;

data HEIS_&ost;
      SET HEIS;
      WHERE OSTAN="&ost";
RUN;
PROC EXPORT DATA= WORK.Heis_&ost
            OUTFILE= "Desktop\OSTAN\OSTAN.xls"
            DBMS=EXCEL REPLACE;
     SHEET="OST_&ost";
RUN;
```

برای استان ۰۱ کافی است در خط اول ۰۰ را به ۰۱ تغییر داده و برنامه را اجرا کنیم.

یک نکته مهم:

هر گاه نیاز بود که از مقدار یک متغیر ماکرویی داخل علامت نقل قول اســـتفاده کنید باید از علامت نقل قول دوتایی ("") استفاده کنید:

```
WHERE OSTAN='00'  →  WHERE OSTAN="&ost"
```

راه حل سوم: یک برنامه ماکرویی به صورت یک پارامتر برای استان نوشته و فراخوانی آن‌را برای ۳۱ بار تکرار می‌کنیم:

```
%MACRO export(ost);
      DATA HEIS_&ost;
            SET HEIS;
            WHERE OSTAN="&ost";
      RUN;
      PROC EXPORT DATA= HEIS_&ost
                OUTFILE= "Desktop\OSTAN\OSTAN.xls"
                DBMS=EXCEL REPLACE;
            SHEET="OST_&ost";
      RUN;
%MEND export;

%export(00)

%export(01)

...

%export(30)
```

در حال حاضر، راه حل سوم از بقیه بهتر است اما حتما راه‌حل‌های بهتر دیگری هم هست.

این جلسه، صرفا برای معرفی ماکرو، متغیر و برنامه ماکرویی است. خیلی حالت‌ها و سوال‌های مختلف پیش می‌آید که باید بیشتر در مورد آن‌ها بحث شود که در ادامه یک به یک در مورد آن‌ها کار خواهیم کرد.

در پناه خدای **مهربان**

جلسه دوم

تمرین برای ماکرونویسی در SAS

اگر بخواهیم اصـول کلی ماکرونویسـی رو به زبان خودمانی و سـاده بنویسم باید ابتدا جایی از برنامه که مدام در حال تکرار شدن است را پیدا کنم برای آن متغیر یا برنامه‌ی ماکرویی تعریف کنم. مثلا در مثال جلسه اول، داشتیم

```
data HEIS_00;
SET HEIS;
WHERE OSTAN='00';
RUN;
PROC EXPORT DATA= WORK.Heis_00
            OUTFILE= "Desktop\OSTAN\OSTAN.xls"
            DBMS=EXCEL REPLACE;
     SHEET="OST_00";
RUN;
```

یعنی چیزی که باید عوض شـود کد اسـتان اسـت که در برنامه بالا ۰۰ اسـت و ما می‌خواهیم کدهایی از ۰۰ تا ۳۰ را به ترتیب بگیرد. پس دقیقـا جـاهـایی که پررنگ شـــده که باید تغییر کند را پاک کرده و به جاش یک متغیر ماکرویی می‌نویســـیم که می‌دانیم متغیر ماکرویی باید با & نوشته شود. پس می‌نویسیم:

```
DATA HEIS_&ost;
      SET HEIS;
      WHERE OSTAN="&ost";
RUN;
PROC EXPORT DATA= HEIS_&ost
          OUTFILE= "Desktop\OSTAN\OSTAN.xls"
          DBMS=EXCEL REPLACE;
      SHEET="OST_&ost";
RUN;
```

حـالا بـاید متغیر ماکرویی رو به عنوان ورودی برنامه ماکرویی معرفی کنم. پس برنامه بالا را داخل یک برنامه ماکرویی می‌نویســـم. اسم برنامه ماکرویی را export می‌گذارم. پس می‌نویسیم:

```
%MACRO export(ost);
      DATA HEIS_&ost;
            SET HEIS;
            WHERE OSTAN="&ost";
      RUN;
      PROC EXPORT DATA= HEIS_&ost
```

```
                OUTFILE= "Desktop\OSTAN\OSTAN.xls"
                DBMS=EXCEL REPLACE;
          SHEET="OST_&ost";
     RUN;
%MEND export;
```

برای فراخوانی برنامه هم که باید به شکل زیر عمل کرد:

```
%export(00)
```

یادمان باشد که فراخوانی برنامه‌ی ماکرویی نیاز به ; ندارد.

تمرین

فرض کنید یک فایل با فرمت mdb (Access) داریم که شامل ۴ جدول است. متغیرهای موجود در هر جدول به صورت زیر است:

نام جدول	متغیرهای موجود
جدول ۱ (RP1)	Address,col01,col02,col03
جدول ۲ (RP2)	PK,code,desc,col01
جدول ۳ (RP3)	Adr,dycol01,dycol02,
جدول ۴ (RP4)	Address34,radifno,formno,age,ms,job,edu

فایل معرفی‌شده را در SAS خوانده و برای بندهای زیر برنامه ماکرویی بنویسید.

۱. متغیر اول در همه جداول، متغیر کلیدی است. دو رقم اول آن برابر با کد استان است. برای هر مجموعه داده کد استان را جدا کرده و به عنوان یک متغیر جدا در مجموعه داده‌های موجود ثبت کنید.

۲. متغیر اول همه مجموعه داده‌ها به نام address تغییر دهید.

۳. متغیرهای col01 و dycol01 کاراکتری به طول ۱۲ هستند به متغیرهای عددی تبدیل شوند.

۴. جدول فراوانی برای متغیرهای formno,age,ms,job,edu را به صورت جداگانه برای هر متغیر را به‌دست آورید و در یک مجموعه داده‌ی جدید ذخیره کنید.

کاملا مشخص است که این تمرین‌ها برای یک فرد مبتدی است اما برای این‌که مفهوم ماکرو کاملا مشخص شود لازم است این تمرین‌ها را با ماکرو بنویسیم تا اشکال کارمان معلوم شود.

جلسه سوم

عبارت شرطی در ماکرونویسی

همان‌طور که می‌دانید برای اجرای شـرطی یک دسـتور یا مجموعه‌ای از دسـتورات در محیط SAS از شـکل عمومی زیر استفاده می‌کنیم:

```
IF  عبارت شرطی ۱ THEN دستور ۱;
ELSE IF  عبارت شرطی ۲ THEN دستور ۲;
…
ELSE دستور آخر;
```

اما حال فرض کنید بخواهیم در محیط ماکرو یک عبارت شرطی بنویسم. از شکل کلی زیر استفاده می‌کنیم:

```
%IF شرط۱ %THEN دستور۱;
%ELSE %IF شرط۲ %THEN دستور۲;
...
%ELSE دستورآخر;
```

مثـال ۱: فرض کنید یک متغیر ماکرویی به نام OST داریم که از ۰ تا ۳۰ را شـامل می‌شـود. ما می‌خواهیم برای مقادیر کوچکتر از ۱۰ قبل از مقدار آن یک صفر اضافه کند و حاصل را در متغیر جدیدی به نام OSTAN ذخیره کند.

```
%IF &OST<10 %THEN %LET OSTAN=0&OST;
%ELSE %LET OSTAN=&OST;
```

جلسه چهارم

حلقه‌های تکرار

در محیط SAS هر گـاه بخواهیم کـاری را تکرار کنیم از حلقه‌های تکرار DO، DO WHILE و DO UNTIL اســتفاده می‌کنیم. معادل این دستورها در محیط ماکرونویسی %DO، %DO %WHILE و %DO %UNTIL است.

شکل کلی %DO به صورت زیر است:

%DO متغیر نشانگر=مقدار آغازین %TO مقدار پایانی;

مجموعه‌ی دستورها

%END;

متغیر نشانگر که در اینجا تعریف می‌شود خود یک متغیر ماکرویی است.

مثال: در جلسه اول، برنامه‌ای ماکرویی به صورت زیر را برای ۳۱ استان تکرار کردیم.

```
%MACRO export(ost);
      DATA HEIS_&ost;
            SET HEIS;
            WHERE OSTAN="&ost";
      RUN;
      PROC EXPORT DATA= HEIS_&ost
                OUTFILE= "Desktop\OSTAN\OSTAN.xls"
                DBMS=EXCEL REPLACE;
            SHEET="OST_&ost";
      RUN;
%MEND export;

%export(00)

%export(01)

...

%export(30)
```

حال اگر بخواهیم این تکرار را با استفاده از حلقه انجام دهیم می‌توان نوشت:

```
%MACRO export;
      %DO i=0 %TO 30;
            %IF &i<10 %THEN %LET OST=0&i;
            %ELSE %LET OST=&i;

            DATA HEIS_&ost;
                  SET HEIS;
                  WHERE OSTAN="&ost";
            RUN;
            PROC EXPORT DATA= HEIS_&ost
                        OUTFILE= "Desktop\OSTAN\OSTAN.xls"
                        DBMS=EXCEL REPLACE;
                  SHEET="OST_&ost";
            RUN;
      %END;
%MEND export;
```

توضیح برنامه: نوشتن عبارت شرطی برای این است که کد استان‌ها از ۰۰ شروع می‌شود نه صفر. بنا بر این لازم است در حین نوشتن حلقه، این تبدیل از ۰ به ۰۰، از ۱ به ۰۱ و ... انجام شود. وگرنه برای استان‌های ۰۰ تا ۰۹ یک sheet خالی ایجاد می‌شد.

در پناه خدای **مهربان**

جلسه پنجم

چند نکته باریک‌تر از مو در SAS

۱. نکته‌ی **نقطه**: نقطه یا . در ماکرونویسی برای مشخص کردن پایان نام متغیر استفاده می‌شود. لذا هر جا لازم بود از . استفاده کنیم و قبل از . نام متغیر ماکرویی وجود داشت باید از .. استفاده کنیم. به عنوان مثال فرض کنید نام فایل‌های ورودی ما ost00.xls تا ost30.xls باشد و برای مشخص کردن کد استان متغیر ماکرویی به نام ostan تعریف کرده‌ایم. حال برای مشخص کردن نام فایل باید از الگوی &ostan..xls استفاده کنیم. یعنی اگر می‌نوشتیم &ostan.xls پردازشگر ماکرویی این عبارت را به شکل 00xls تا 30xls تفسیر می‌کرد و چون چنین نام فایلی وجود ندارد خطا می‌گرفت.

برای درک بهتر نکته‌های بعدی به مثال زیر توجه کنید:

مثال خیلی خیلی قشنگ: می‌خواهیم فایل‌هایی را بخوانیم که شامل دو نوع تکرار در نام آن‌ها هستیم. به عنوان مثال نام فایل به صورت data001.dbf، data002.dbf، data011.dbf و ... است. یعنی علاوه بر این‌که در نام فایل کد استان وجود دارد برای هر استان کد شهری/روستایی نیز (۱: شهری و ۲: روستایی) نیز وجود دارد. در این حالت از دو حلقه تو در تو برای خواندن داده‌ها استفاده می‌کنیم:

```
%MACRO import;
      %DO i=0 %TO 30;
            %IF &i<10 %THEN %LET OST=0&i;
            %ELSE %LET OST=&i;
            %DO j=1 %TO 2;
                  PROC IMPORT OUT= data&ost&j
                              DATAFILE= "Desktop\data&ost&j..DBF"
                              DBMS=DBF REPLACE;
                        GETDELETED=NO;
                  RUN;
            %END;
      %END;
%MEND;

%import
```

در این برنامه، متغیر i برای مشخص شدن استان و متغیر j برای مشخص شدن شهری/روستایی به‌کار رفته است. ببینید اگر نمی‌خواستیم از ماکرو استفاده کنیم باید یک برنامه import را ۶۲ بار تکرار می‌کردیم.

نکته‌ی برنامه‌ی تولید شده: برای این‌که برنامه تولید شده توسط ماکرو را ببینید دستورات زیر به ابتدای برنامه‌ی ماکرویی خود اضافه کرده و اجرا کنید:

```
filename mprint "desktop\mymacro.sas";
options mprint mfile;
```

وقتی برنامه اجرا می‌شود یک فایل SAS در دسکتاپ به نام mymacro ساخته می‌شود که تمام اجزای برنامه‌ی ماکرویی را به تفکیک ۶۲ فایل برای شما ذخیره کرده است.

تا وقتی از SAS خارج نشده‌اید، هر چقدر برنامه ماکرویی اجرا کنید تمام مراحل برنامه در این فایل ذخیره می‌شود. مگر این که دستور زیر را اجرا کنید:

```
options nomprint;
```

> شاید سوال این باشد که دیدن این فایل به چه دردی می‌خورد؟
>
> باید گفت گاهی برای رفع اشکال‌های برنامه، دیدن برنامه تولیدشده توسط ماکرو مفید است و در کل به نظر خودم دیدن برنامه‌ای که نوشتن آن زمان زیادی می‌گرفت اما حالا با نوشتن آن در ماکرو سریع‌تر است به تنهایی هم می‌تواند لذت‌بخش باشد☺

ادامه‌ی مثال (روی‌هم ریختن مجموعه‌ی داده‌ها)

همان ۶۲ فایلی که ساختیم را در نظر بگیرید می‌خواهیم این ۶۲ فایل را روی هم ریخته و در یک مجموعه داده به نام ALL ذخیره کنیم. لذا می‌نویسیم:

```
%MACRO ALL;
DATA ALL;
      set
      %DO i=0 %TO 30;
```

```
            %IF &i<10 %THEN %LET OST=0&i;
      %ELSE %LET OST=&i;
            %DO j=1 %TO 2;
            data&ost&j
            %END;
      %END;
      ;
RUN;
%MEND;

%ALL
```

البته اگر داده‌های ما به شـکل data00 تا data30 بودند و شناسه‌ی شهری/روستایی در نام فایل وجود نداشت راه حل غیرماکرویی راحت زیر بسیار مفید بود:

```
DATA all;
      SET data00-data30;
RUN;
```

در پناه خدای **مهربان**

جلسه ششم

ادامه‌ی حلقه‌های تکرار

در ادامه‌ی بحث حلقه‌های تکرار دو شیوه‌ی حلقه‌نویسی به صورت زیر معرفی می‌شوند:

۱. %DO %WHILE

%DO %WHILE(عبارت شرطی);

مجموعه‌ی دستورها

%END;

یعنی تا وقتی عبارت شرطی برقرار است، مجموعه‌ی دستورها اجرا می‌شود.

۲. %DO %UNTIL

این شیوه‌ی حلقه‌نویسی بر خلاف شیوه‌ی بالا، تا وقتی که شرط برقرار نیست مجموعه‌ی دستورها را اجرا می‌کند و با برقرار شدن عبارت شرطی، حلقه خاتمه می‌یابد. شکل کلی آن به صورت زیر است:

%DO %UNTIL(عبارت شرطی);

مجموعه‌ی دستورها

%END;

مثال: مثال گفته شده در جلسه‌ی چهارم را با استفاده از %DO %WHILE مجددا بازنویسی می‌کنیم:

```
%MACRO export;
      %LET i=0;
      %DO %WHILE (&i<=30);
            %IF &i<10 %THEN %LET OST=0&i;
            %ELSE %LET OST=&i;

            DATA HEIS_&ost;
                  SET HEIS;
                  WHERE OSTAN="&ost";
            RUN;
            PROC EXPORT DATA= HEIS_&ost
                        OUTFILE= "Desktop\OSTAN\OSTAN.xls"
                        DBMS=EXCEL REPLACE;
                  SHEET="OST_&ost";
            RUN;
            %LET i=%EVAL(&i+1);
      %END;
%MEND export;
```

برنامه بالا می‌گوید قبل از وارد شدن به حلقه، مقدار متغیر ماکرویی i را برابر ۰ قرار می‌دهیم و در انتهای حلقه نیز به مقدار متغیر یکی اضافه می‌کنیم.

معرفی تابع %EVAL:

را این تابع محاسبات منطقی و ریاضی در ماکرونویسی انجام می‌دهد. به عبارت دیگر اگر می‌نوشتیم

```
%LET i=&i+1;
```

و مقدار i برابر با ۲ باشد، بعد از اجرای این دستور، مقدار متغیر i برابر 2+1 خواهد شد. در صورتی که هدف ما این است که حاصل آن ۳ شود. لذا در ماکرونویسی هرگاه لازم شد محاسبه ریاضی انجام شود، حتما باید از این تابع استفاده کنیم.

(دوستان عزیز این جزوه طبق تدریس آقای حسن رنجی به من (آسیه عباسی) برای استفاده در گروه تلگرامی SAS Users تهیه شده است. استفاده از مطالب این جزوه با ذکر منبع بلا مانع و باعث خوشحالی ماست. چون ماکرو خیلی خوب است ☺)