ب- خواندن داده ها از فایل خارجی

اگر اطلاعاتی از قبل در فایل متنی یا بانک اطلاعاتی دیگری ذخیره شده باشد، برای خواندن آن به صورت زیر عمل می‌کنیم.

۱- به وسیله برنامه نویسی

```
Data      ; نام مجموعه داده
Infile ' آدرس \ نام فایل . پسوند ' ;
Input     ; نام متغیرها
run;
```

* اگر تعداد زیادی متغیر عددی در مجموعه داده وجود دارد می‌توان به جای نوشتن همان تعداد متغیر در دستور input، از برای مثال X1 - X10 استفاده کرد. (در مورد ۱۰ متغیر).

* درصورتی که سطر اول داده ها، اسم متغیرها باشد و یا اینکه چند سطر اول توضیحاتی در مورد مجموعه داده باشد از دستور Firstobs = ... در ابتدای دستور Infile می‌توان استفاده کرد.

* درصورتی که داده‌ها با فاصله از هم جدا شده باشد از دستور Infile می‌توان استفاده نمود. در غیر اینصورت باید از دستور Delimiter = ... جهت مشخص کردن وقایع جدا شدن داده‌ها از هم استفاده کرد.

مثال:

```
Data petrol;
Infile ' D:\ petrol_space.txt ;
Input  X1 - X5 ;
run;
```

تمرین: به help نرم افزار مراجعه نمایید و وقایع جدا شدن داده‌ها را مشخص کنید و مثالی برای آن ارائه دهید.

۲- از طریق منو

- ابتدا از منوی File، گزینه Import data را انتخاب می‌کنیم. سپس درصورتی که مجموعه داده‌ها با tab از هم جدا شده باشد از گزینه اول موجود در پنجره استفاده می‌کنیم. Next. سپس از طریق Browse آدرس فایل را مشخص می‌کنیم. با استفاده از گزینه Option می‌توان مشخص نمود که سطر اول مجموعه داده‌ها شامل نام متغیرها باشد.

- درصورتی که داده‌ها با فاصله از هم جدا شده باشد از گزینه دوم موجود در پنجره استفاده می‌کنیم. Next و .... (توضیحات در کلاس داده شد.)

ج- ایجاد مجموعه داده با استفاده از خروجی پروسیجرها

بسیاری از نتایج حاصل از اجرای پروسیجرها، قابل ذخیره شدن به صورت مجموعه داده هستند. این امر با استفاده از گزینه Output صورت می‌گیرد.

```
Output  out = اسم مجموعه داده ;
```

که کلمات کلیدی صفات داده‌های مختلف در پروسیجر مورد نظر

مثال:

```
Data a;
input x y;
Cards;
5  16
8  10
9  12
3  14
;
proc means;
Var x y;
output out=myout mean=m1 m2 stdev=s1 s2;
run;
```

→ ایجاد مجموعه داده جدید با استفاده از دستورهای Set و Do .

با استفاده از دستور Set می‌توان اطلاعات یک مجموعه داده را در مجموعه داده دیگر کپی کرد.

مثال

```
Data b;
Set a;
Z=x*y;
run;
```

با استفاده از دستور Do نیز می‌توان مجموعه داده ایجاد کرد.

```
Do i=1 to N by h;
    عبارت ; output;
End;
```

اختیاری

آرگومان آخرین متغیر ایجاد شده در حلقه do را نیز نگه می‌دارد.

مثال

```
Data d;
Do i=1 to 20;
    x=2*i+100; output;
End;
```

می‌توان از چند حلقه Do نیز استفاده کرد.

مثال

```
Data e;
Do i=1 to 100;
    Do j=1 to 200;
        x=2*i+3*j+5; output;
    End;
End;
```

```
Data f;
Do i=1 to 200;
    Do j=i to i+200;
        x=100*i; output;
    End;
End;
```

ویرایش مجموعه داده‌ها :

در گام داده، در حین یا پس از ایجاد مجموعه داده، می‌توان آن را ویرایش نمود.

- توابع مورد استفاده :

با استفاده از توابع زیر می‌توان تغییرهای جدیدی را از روی متغیرهای موجود، ایجاد کرد.

در صورتی که گام داده، شامل دستور input است، این توابع باید بعد از این دستور آمده باشند و قبل از cards.

برخی از توابع موجود SAS :

ABS, MOD, SIGN, SQRT, MAX, MIN, SUM, EXP, LOG, LOG10

MOD(x,a) باقیمانده تقسیم x بر a

CEIL, FLOOR, ROUND, INT

- CEIL: کوچکترین عدد صحیح بزرگتر از یا مساوی ورودی
- FLOOR: بزرگترین عدد صحیح کوچکتر از یا مساوی ورودی
- ROUND: گرد کننده
- INT: قسمت صحیح یا مساوی ورودی

round(3.678) = 4

round(3.678, .1) = 3.7

* توجه شود که توابع بالا به صورت سطری روی شماره‌های متغیرها عمل می‌کنند. برای مثال در مجموعه داده a، متغیر w ای ایجاد می‌شود که هر مقدار آن ماکزیمم هر ردیف شامل x و y است.

```
Data a;
input x y z;
w = max(x,y);
cards;
1 5 8
9 3 4
6 8 5
9 10 20
;
```

* ماکزیمم شماره‌های موجود در x را باید از طریق پردازش‌ها، محاسبه نمود.

* از عمگرهای +، -، /، * نیز می‌توان استفاده نمود.

- دستور IF :

از دیگر دستوراتی که می‌توان در گام داده به کار برد دستور if است.

```
If شرط then عبارت;
```

یا

```
If شرط then عبارت else عبارت دوم;
```

* در شرط از عمگرهای مقایسه‌ای زیر می‌توان استفاده کرد.

= , ^= , > , < , >= , <=

$\neq$ یعنی

مثال:

```
Data a;
input x y w $;
If x>5 then z=0;
cards;
8 3 A
4 20 B
9 11 A
5 3 A
6 27
8 21 B
;
```

* توجه شود که این دستور نیز در صورت وجود دستور input، بعد از این دستور و قبل از cards می‌آید.

* از عمگرهای and و or نیز می‌توان در قسمت شرط استفاده کرد.

مثال

```
Data b;
set a;
if W='A' then W='Good';
if Y<5 and X>2 then delete;
run;
```

* به منظور حذف مشاهده از دستور delete می‌توان استفاده نمود.

* با استفاده از دستور delete به منظور حذف مشاهده‌ای خاص از یک تغییر، مشاهدات تغییرهای دیگر که شرایط با آن مشاهده بودند نیز حذف می‌کند.

* دستورهای Keep و Drop

- همان طور که در بالا ذکر شد به منظور حذف مشاهده از دستورهای if و delete می‌توان استفاده نمود. ولی همیشه حذف تغییر از مجموعه داده، از دو دستور Keep و Drop می‌توان استفاده کرد.

- این دو دستور نیز در صورت وجود input، بعد از آن قبل از Cards می‌آیند.

```
Data e;
set b;
Drop X Z;
run;
```

- تغییرهایی که جلوی دستور Drop می‌آیند از مجموعه داده حذف می‌شوند و تغییرهایی که جلوی دستور Keep می‌آیند، در مجموعه داده حفظ می‌شوند.

یا به طور مثال

```
Data f;
set b;
Keep Y W;
run;
```

تمرین: مجموعه داده‌ای ایجاد کنید که تنها شامل تغییر Y باشد و مقادیر آن اعشاری عدد 13 که کمتر از 200 هستند، باشد.

* داده‌های گمشده:

- داده‌های گمشده عددی را با . و رشته‌ای را با فاصله خالی مشخص کرد.

- این داده‌ها را با دستورهای if و delete می‌توان حذف کرد.

```
Data g;
input X 1-2 Y$ 4-5;
if X=. then delete;
if Y=' ' then delete;
Cards;
12 A
5 AB
6 A
17
8 B
;
```

→ دستور input مشخص می‌کند، فاصله‌ها در دستور input مشخص می‌کند، فاصله‌ها در داده‌ها، در هنگام ورود اطلاعات اصلاح پیدا می‌کند.

* عبارات مورد استفاده پس از then یا else در دستور if تنها شامل یک جمله می‌تواند باشد. در صورتی که بیشتر از جمله مورد نیاز بود به طریق زیر عمل می‌کنیم

```
if b>0 then
  do;
  عبارت 1;
  ⋮
  n عبارت;
  end;
else;
  do;
  n+1 عبارت;
  ⋮
  n+m عبارت;
  end;
```

مثال:

```
Data h;
set g;
if x<5 then
  do;
  xx=0;
  y='NO';
  end;
```

* در SAS برای نوشتن جملات توضیحی و یا غیرفعال کردن قسمتی از دستورات، بدون نیاز به حذف آنها، از نماد * استفاده می‌شود.

مثال:

```
Data a;
input x y;
cards;
1 3
2 15
3 7
;
* This is a SAS Code.
/* if x<2 then z=5;
   if y>5 then w=3; */
```

مثال:

```
Data a;
input x y;
cards;
1 2
3 4
5 6
;
Data b;
input x y;
cards;
7 8
9 10
;
Data c;
input z;
cards;
7
8
9
10
;
```

```
Data all1;
set a b;
run;
Data all2;
set a;
set c;
run;
Data all3;
merge a c;
run;
```

تلفیق چند مجموعه داده

- اگر مجموعه داده‌ها دارای متغیرهای یکسان و مشاهدات متفاوت باشند، با آوردن نام مجموعه داده‌ها در جلوی دستور set می‌توان مجموعه داده‌ها را تلفیق کرد.

- اگر مجموعه داده‌ها دارای متغیرهای متفاوت باشند با استفاده از چند دستور set و یا استفاده از دستور merge می‌توان مجموعه داده‌ها را تلفیق کرد. در این حالت اگر تعداد مشاهدات متغیرهای مجموعه داده‌ها یکسان نباشد، دستور merge ما را از تعداد مشاهدات و دستورهای set، سیستم تعداد مشاهدات را ملاک می‌گیرند.