SAS notes – class 3 – Feb 18, 2018

IN | NOT IN – ***works on the values of variables and filters them out.***

-------------------------While-------------------------------

KEEP | DROP ***works only on variables.***

```
Data a;
set sasuser.admit;
if actlevel IN(“HIGH” “LOW”);
run;

data a;
set sasuser.admit;
if age IN(30 40 50);
run;

data a;
set sasuser.admit;
if actlevel NOT IN(“HIGH” “LOW”);
run;
```

First interview question – difference between IF and WHERE

```
Data a;
Set sasuser.admit;
If age gt 40;
Run;
```

```
Data a;
Set sasuser.admit;
Where age gt 40;
Run;
```

They both gives the same output. However, WHERE is more efficient as it applies the filter pre-buffering of data. IF applies post-buffering of data. So, composite filter - Where consumes less time to show the results as compare with composite filter – IF.

```
data a;
set sasuser.admit;
age_m=age*12;
if age_m gt 400;
run;
```

```
data b;
set sasuser,admit;
age_m=age*12;
where age_m gt 400;
run;
```

NOTE :- use KEEP and DROP mainly with set statement.

```
data a(keep id age_m);
set sasuser.admit;
```

```
run;

data a;

set sasuser.admit(keep id age_m);

run;
```

INTERVIEW QUESTION – USE OF MULTIPLE IF | WHERE

```
------------------------------------------first comparison-----------------------------------------------------

data a;

set sasuser.admit;

if age gt 40 AND sex=”M”;

run;


data a;

set sasuser.admit;

where age gt 40 AND sex=”M”;

run;

----------------------------------------------second comparison--------------------------------------------

data b;

set sasuser.admit;

if age gt 40;

if sex=”M”

run;


data c;

set sasuser.admit;

where age gt 40;

where sex=”M”;
```

```
run;
```

----------------------------------third comparison---

```
data c;

set sasuser.admit;

if age gt 40;

where sex=”M”;

run;
```

NOTE:- WHERE always go in loops with only last one, it doesn’t run the previous WHERE loop again.

USE OF “ “ AND ‘ ‘

We can use both of them in SAS statements to cover values of variables. But only in case, if we have any signle ‘ in value and we use ‘ ‘ then code will not run properly and may show error in this condition. So, we are supposed to use only “ “ so that there will be no possibility to get error in this case.

e.g. if we have variable with name **CLIENT** and it has some value equal to **DONALD’S** then if we use single quotes, it will not show us desired results while double quotes will do.’

```
data a;

set sasuser.admit;

if client =”DONALD’S”

run;
```

FEW THINGS TO KEEP REMEMBER ABOUT SAS CODING

- ✓ SAS is not case sensitive.
  →value of character variables:
  If sex=”M”;
  If sex=”m”;

- ✓ SAS is space free.
  ```
  Data          a                    ;
  ```

- ✓ SAS is line free.

```
data

a

;
```

APPENDING TABLES

First of all, appending two or more tables doesn’t mean to merge them. Merging is completely different scenario as compare with appending. Appending two or more tables simply means to connect them with each other via some media i.e. using set statements by calling two or more tables at once.

---------------------------e.g.-1:----------------------------

```
data a;

set sasuser.admit sasuser.admit;

run;
```

---------------------------e.g.-2:----------------------------

```
data b;

set jan feb mar;

run;
```