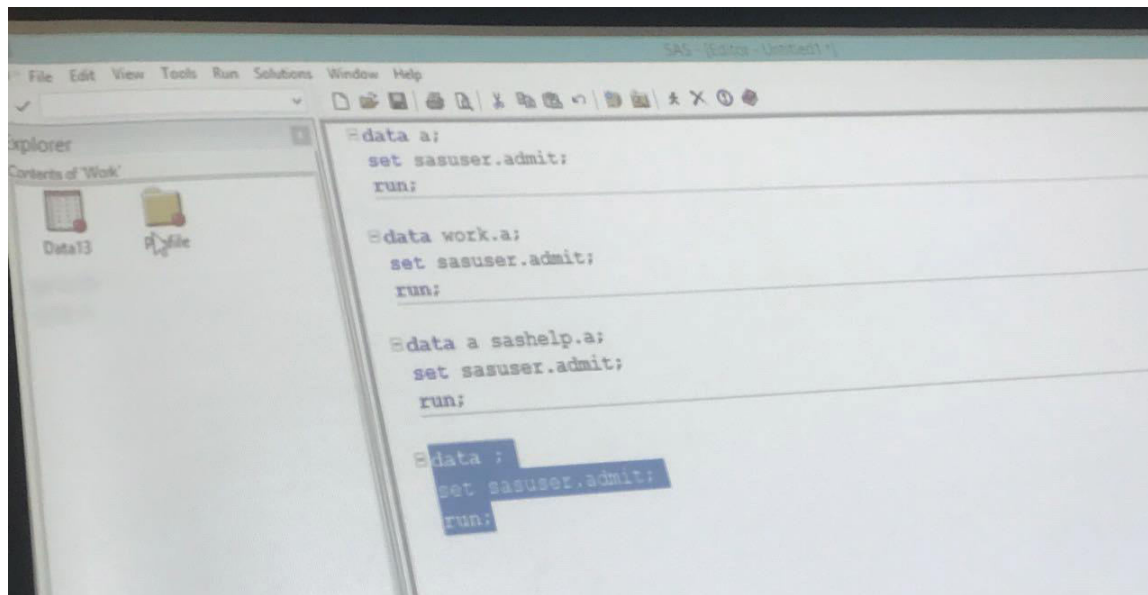


Dated: 17/02/2018

Statistical Analysis System (SAS v9.4): Class 2

Revising on a few codes:



- Data a;
set sasuser.admit;
run;
- Data work.a;
set sasuser.admit;
run;

Result: Both these codes creates a dataset "a" in temporary library "work" from dataset admit of permanent library "sasuser"

- Data a sashelp.a;
set sasuser.admit;
run;

Result: This creates two datasets, one in temporary - "work " library and the other in

permanent "sashelp" library from dataset admit of permanent "sasuser" library.

- Data;

```
set sasuser.admit;
```

```
run;
```

Result: This creates a dataset in temporary library "work" with a default name say "data 1" in the first instance from dataset "admit" of permanent library "sasuser".

Note: It is necessary to define name of data in a datastep otherwise it gets a default name.

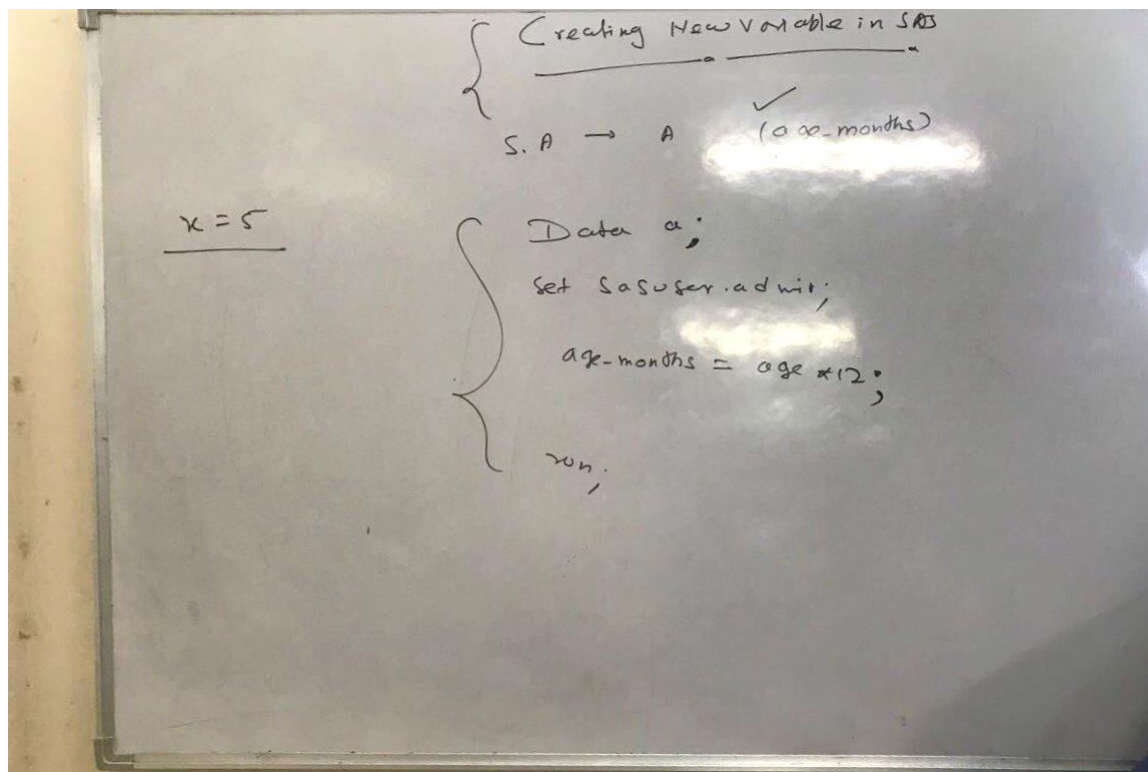
example: given above

Creating new variable in SAS:

- Understand your data, any new variable created should be self explanatory to itself from its name for ease.
- Syntax to create a new variable:

variable name = logic / value calculated / constant or hardcore value
- All the new variables are added to the last in a dataset.

Examples:



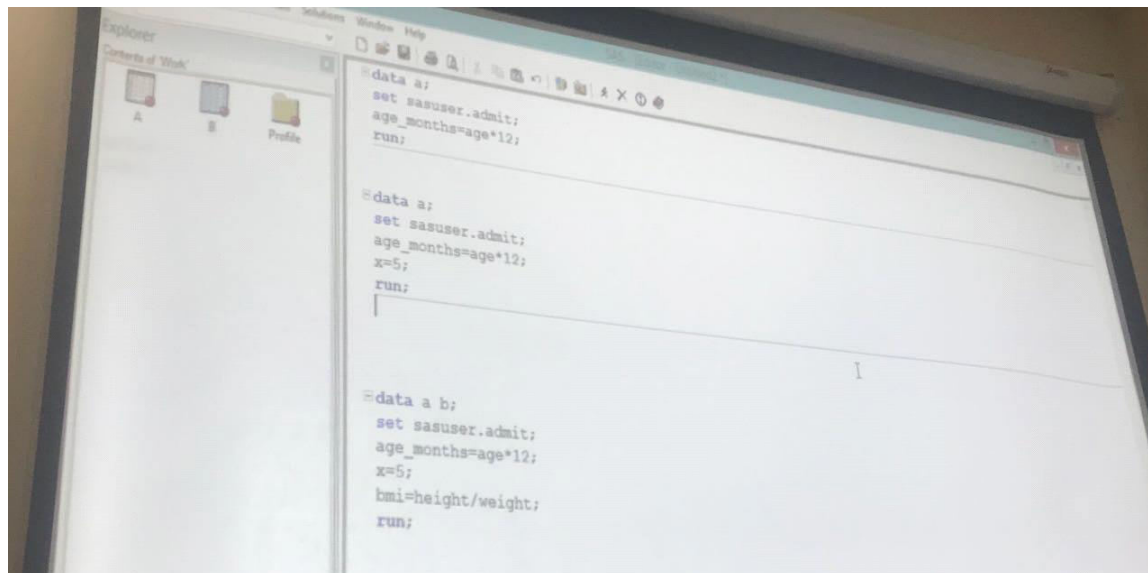
- Data a;

set sasuser.admit;

age_months = age*12;

run;

Result: It creates a dataset "a" in Work library from "admit" dataset of sasuser library alongwith a new variable (column) age_months (displays age in months).



- Data a;

```
set sasuser.admit;
```

```
age_months = age*12;
```

```
x = 5;
```

```
run;
```

Result: It creates a dataset "a" in Work library from "admit" dataset of sasuser library along with two new variables (column). First is age_months (displays age in months) second is variable "x" whose value is constant 5.

- Data a b;

```
set sasuser.admit;
```

```
age_months = age*12;
```

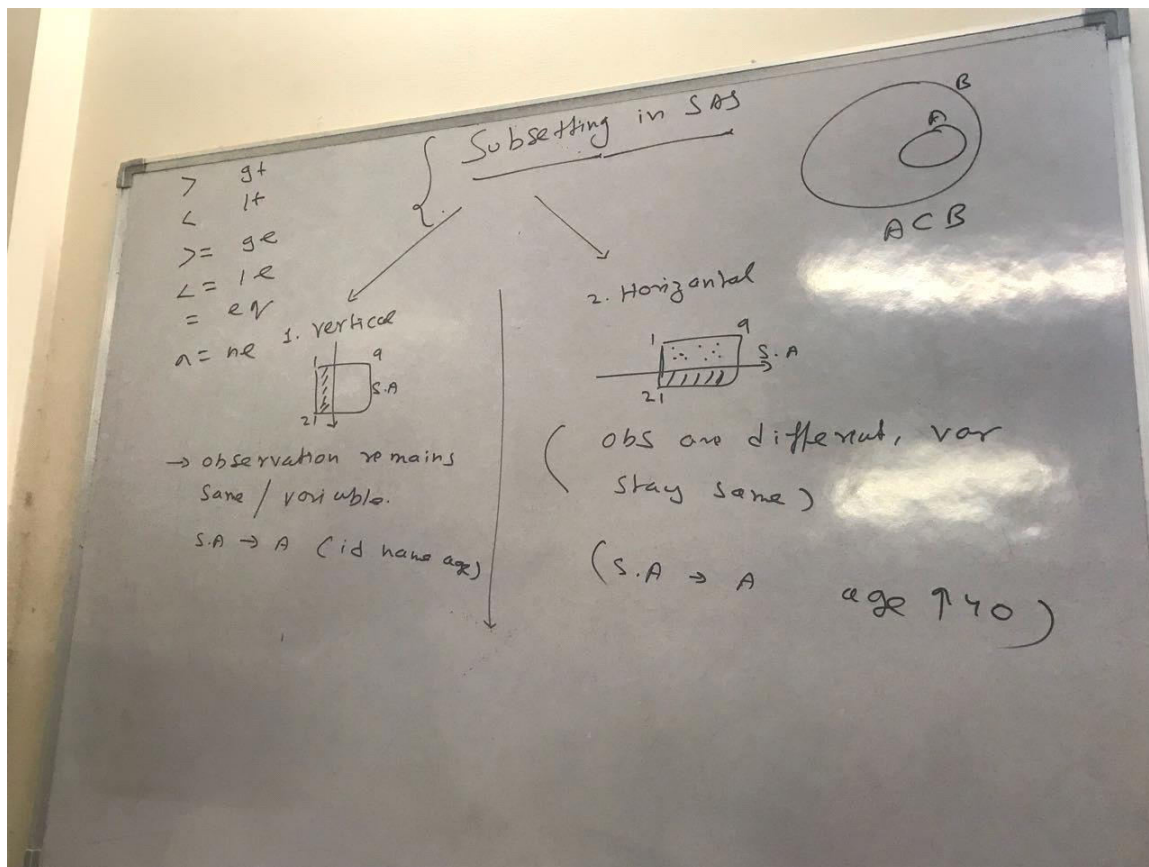
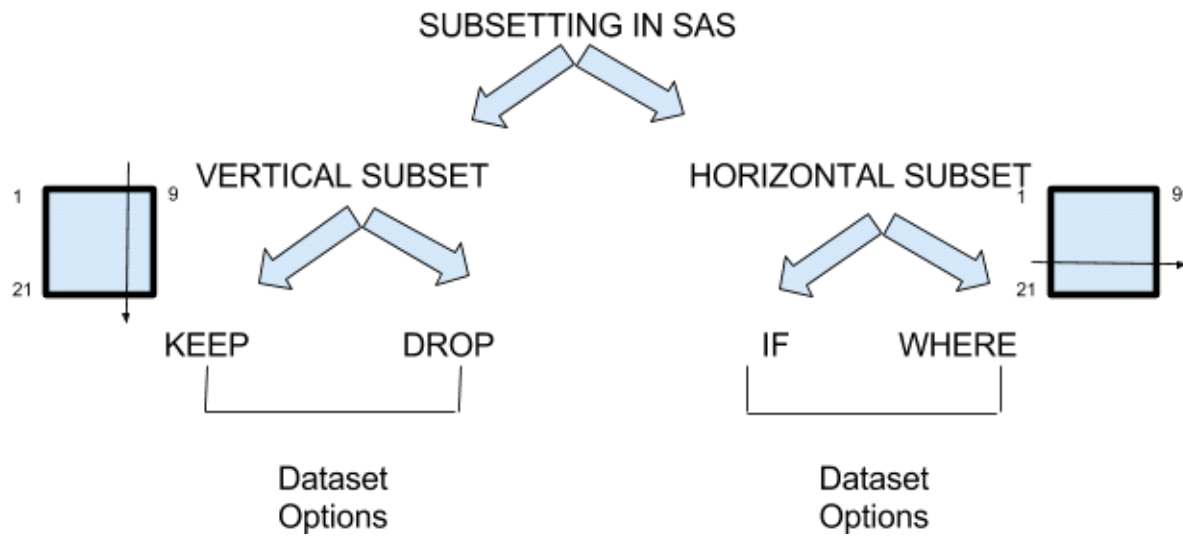
```
x = 5;
```

```
bmi = height / weight;
```

```
run;
```

Result: Likewise here it adds the third variable (column) to the same above example i.e bmi (calculates ratio of height by weight). Also it creates two datasets "a" and "b" in work library from admit of sasuser library.

Subsetting in SAS:

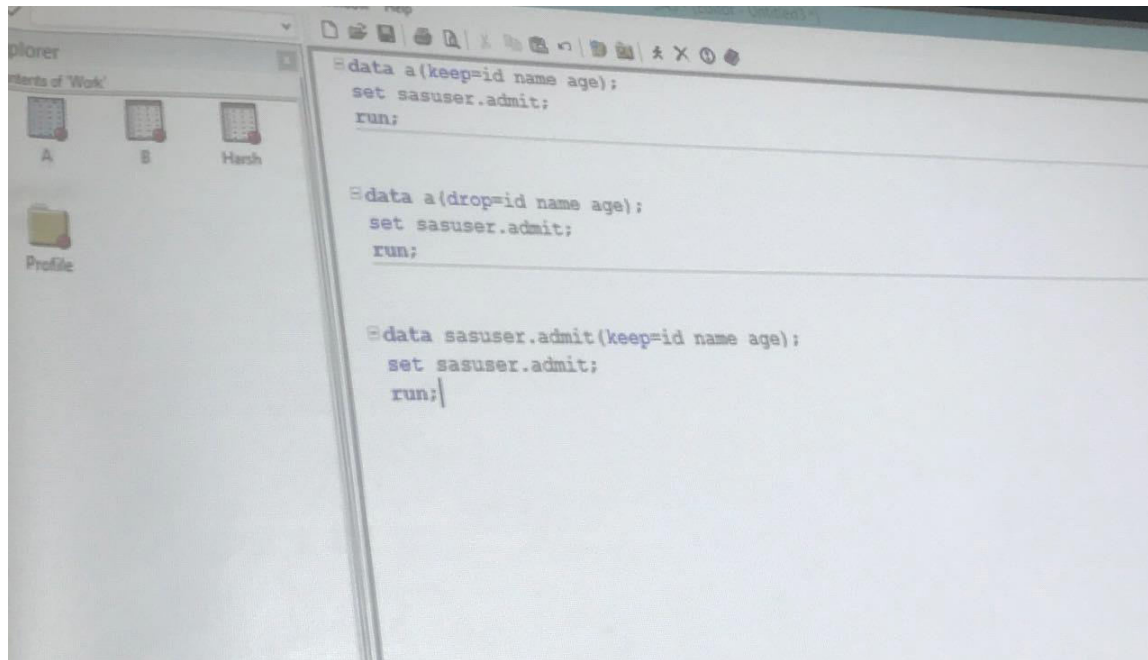


Vertical Subset:

- Observations remains same variables changes.

- Uses KEEP and DROP as the two dataset options for desired actions.

Examples:



- Data a (keep = id name age);

set sasuser.admit;

run;

Result: Creates dataset "a" in work library from "admit" of sasuser library with just variables : ID, name, age but the number of observation does not changes.

- Data a (drop = id name age);

set sasuser.admit;

run;

Result: Creates dataset "a" in work library from "admit" of sasuser library with rest all the same variables as were there initially in "admit" dataset except variables : ID, name, age but again the number of observation does not changes.

- data sasuser.admit (keep = id name age);

set sasuser.admit;

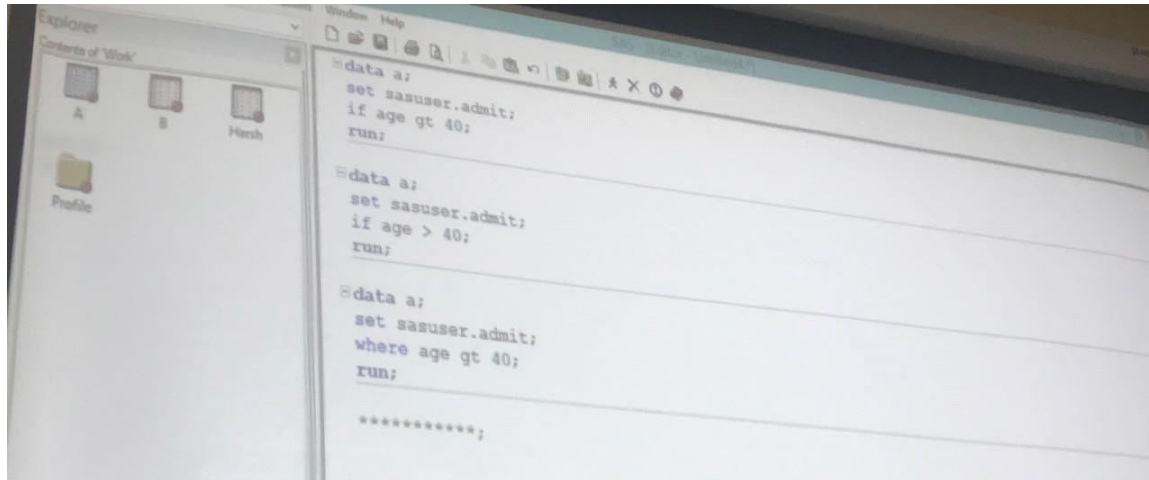
run;

Result: overwrites dataset "admit" in sasuser library keeping just three variables : ID, name, age but the number of observation does not changes.

Horizontal Subset:

- Observations changes variables remains same.
- uses IF and WHERE as the dataset options for the desired actions.

Examples:



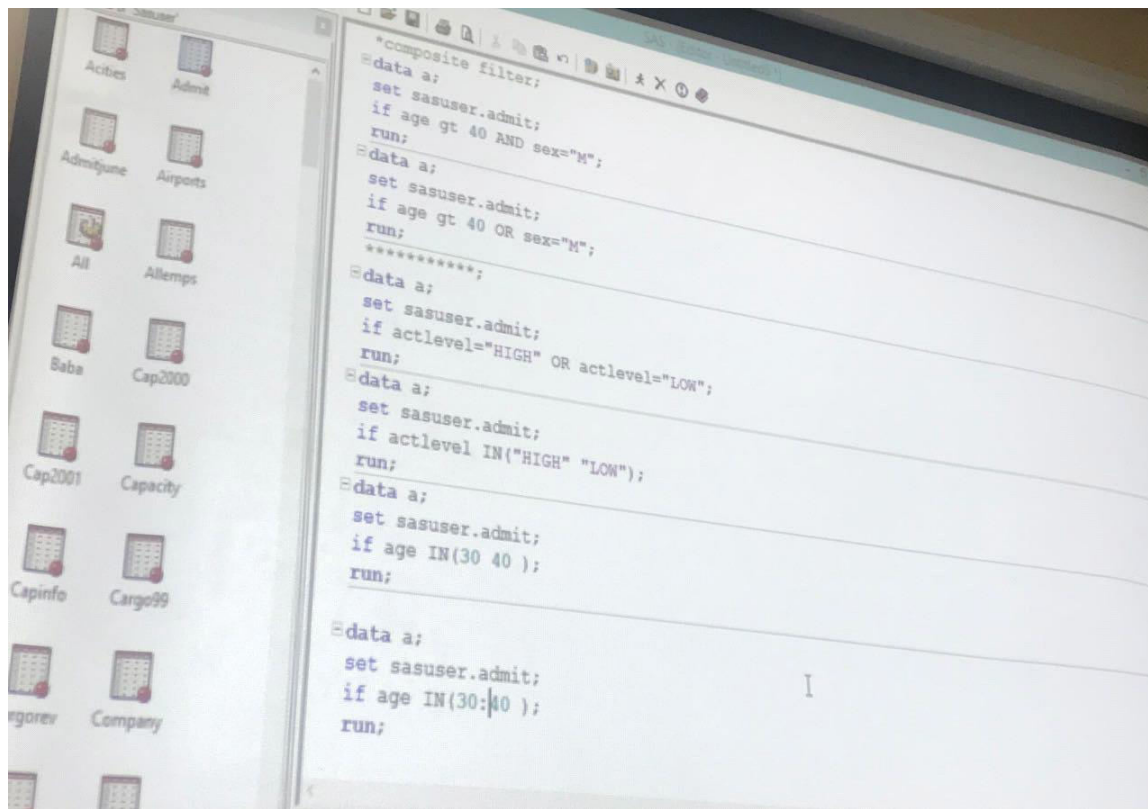
- data a;
set sasuser.admit;
if age gt 40;
run;
- data a;
set sasuser.admit;
if age > 40;
run;
- data a;
set sasuser.admit;
where age gt 40;
run;

Result: In all these 3 examples the respective code creates dataset "a" in work library from "admit" dataset of sasuser library but only for observations (rows) with age greater to 40.

Note: SAS is compatible with both symbols and Mnemonics.

- > gt Greater than
- < lt Lesser than
- >= ge Greater than or equal to
- <= le Lesser than or equal to
- ^= ne not equal to
- = eq equal to

Composite Filter:



- **And:** This to operate both the conditions should be true.

example:


```
data a;  
  
set sasuser.admit;  
  
if age gt 40 AND sex = "M";  
  
run;
```

Result: Creates dataset "a" in work library from "admit" dataset of sasuser library with condition of age greater to 40 and sex = "M", both being true.

- **OR:** This can operate with any one condition being true, also to filter if more than one data with same variable.

example:

```
data a;  
  
set sasuser.admit;  
  
if age gt 40 OR sex = "M";  
  
run;
```

Result: Creates dataset "a" in work library from "admit" of sasuser library if any of the condition age greater to 40 OR sex = "M" satisfies.

example:

```
data a;  
  
set sasuser.admit;  
  
if actlvel = "high" OR actlevel = "low";  
  
run;
```

Result: Creates dataset "a" in work library from "admit" of sasuser library if any of the condition gets true, actlevel = "high" OR actlevel ="low", here actlevel is the same variable from the dataset with different values "high" or "low".

- **IN:** functions similar to OR

Example:

```
data a;
```

```
set sasuser.admit;  
  
if actlevel IN ("high" "low");  
  
run;
```

Result: creates dataset "a" in work library from "admit" dataset of sasuser library if actlevel is either high or low gets satisfied.

- **Not IN:**

```
data a;  
  
set sasuser.admit;  
  
if actlevel Not IN ("high" "low");  
  
run;
```

Result: Creates a dataset "a" in work library from "admit" dataset of sasuser library if variable actlevel does not have value high or low.

