CLASS-21 (12/05/2018)
****

MACROS

Macro : These are the tools for doing automation.It makes the work faster by automating the task that requires writing same lines of code every time.

Macro variable: The variables of type macros that stores the data as **TEXT** only. Macro variables are referenced by using ampersand (&) followed by macro variable name.

Different ways to create macro variable:

1.%LET : The syntax of the %LET statement

```
%let macro-variable-name=value;
%let  a=amit;
```

***;

Code:

```
%let a=amit;
```

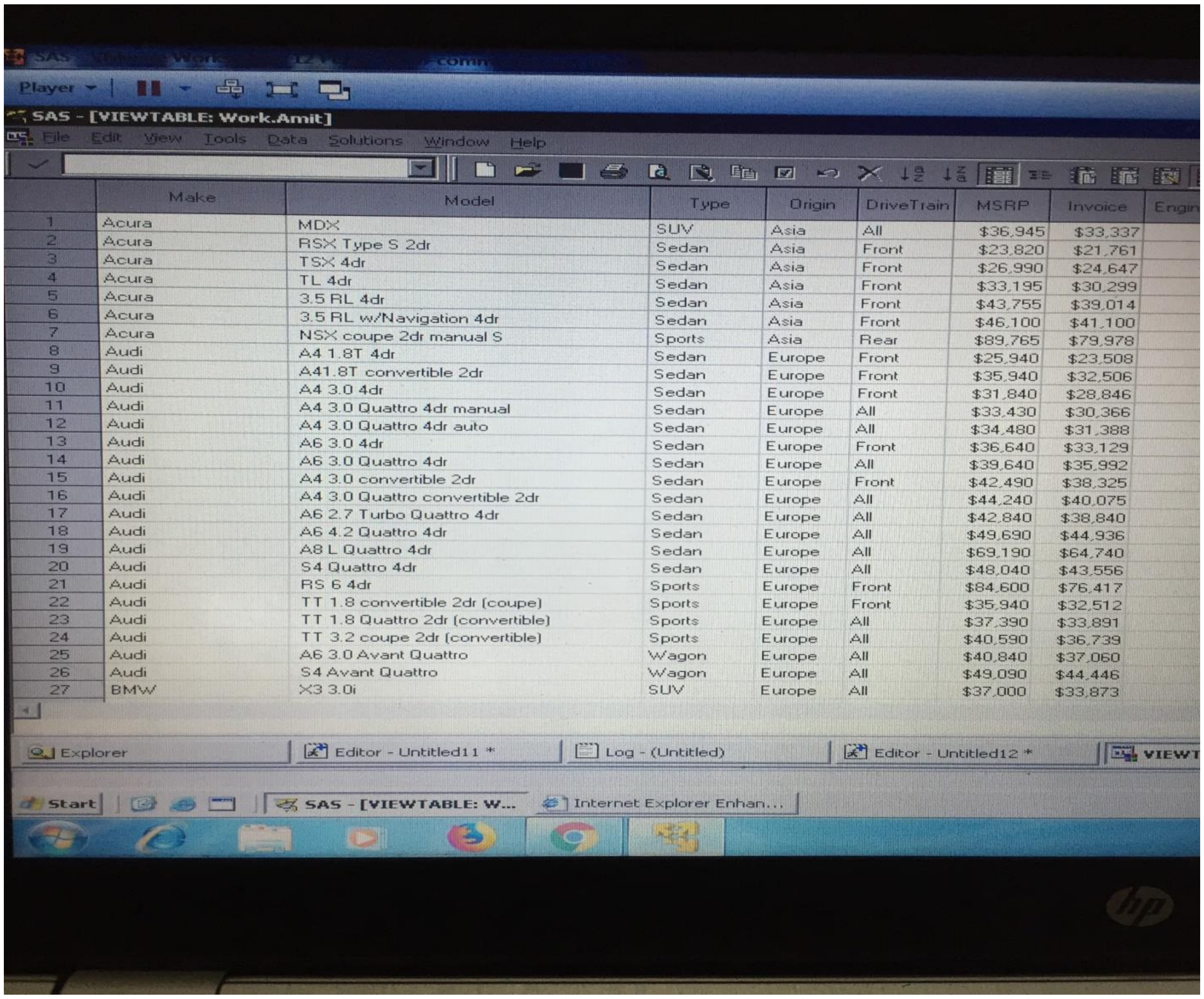
```
Data &a;
Set sashelp.cars;
Run;
```

```
Proc print data=&a;
Run;
```

%let a=amit;	// a= macro variable , value = amit //

Data &a;
Set sashelp.cars;
Run;
	// Suppose amit is to be used many times in the code, so instead of amit you can write "&a"
	
	& is a trigger(call) to the macro variable a . It is text substitution (find and replace). Dataset amit will be created. Same way we can change the value amit by any other name and can create different data sets.//

Proc print data=&a;
Run;	


Player
SAS - [VIEWTABLE: Work.Amit]
File Edit View Tools Data Solutions Window Help
Explorer
Editor - Untitled11 *
Log - (Untitled)
Editor - Untitled12 *
Start
SAS - [VIEWTABLE: W...
Internet Explorer Enhan...

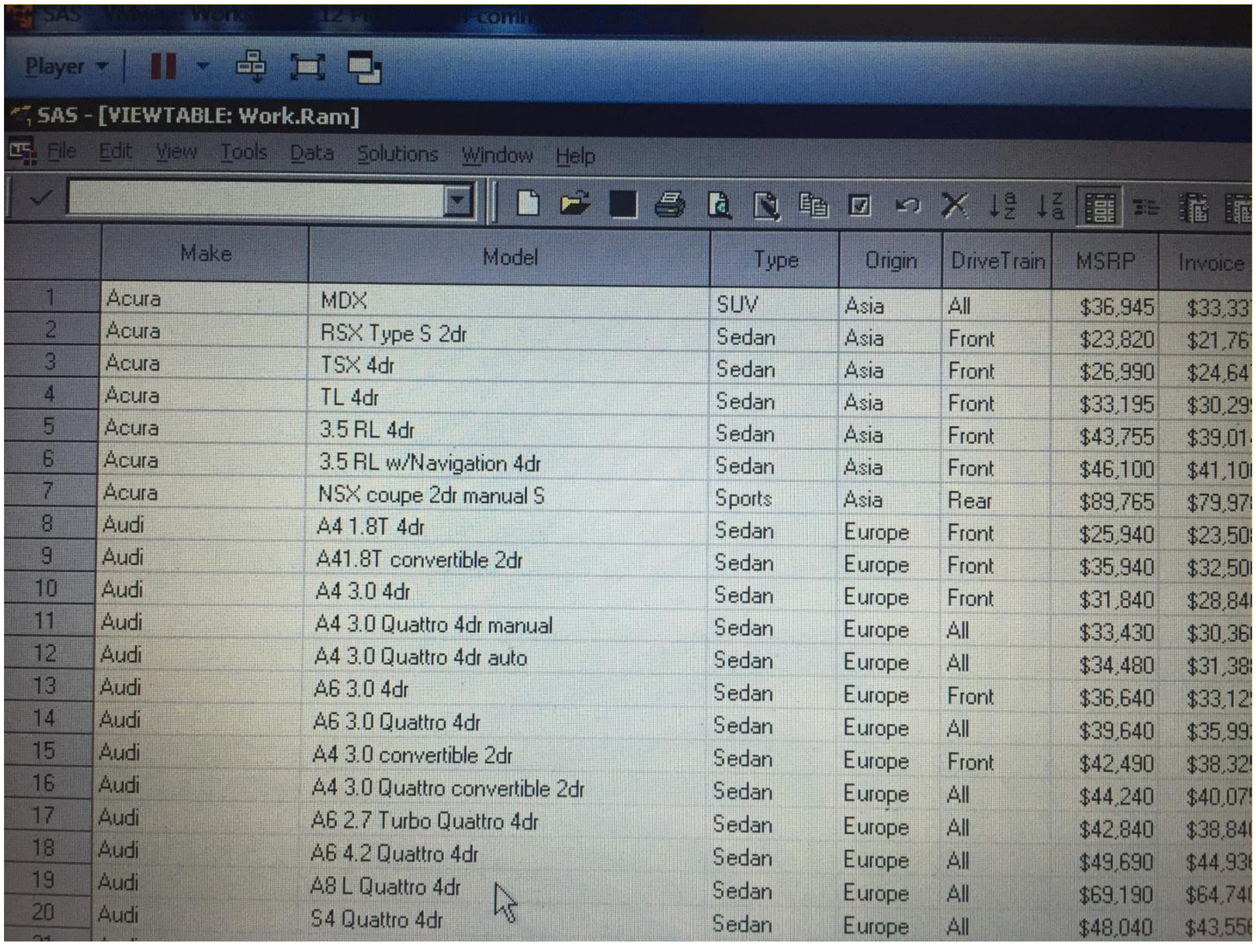
	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice
1	Acura	MDX	SUV	Asia	All	$36,945	$33,337
2	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761
3	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647
4	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,299
5	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,014
6	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,100
7	Acura	NSX coupe 2dr manual S	Sports	Asia	Rear	$89,765	$79,978
8	Audi	A4 1.8T 4dr	Sedan	Europe	Front	$25,940	$23,508
9	Audi	A41.8T convertible 2dr	Sedan	Europe	Front	$35,940	$32,506
10	Audi	A4 3.0 4dr	Sedan	Europe	Front	$31,840	$28,846
11	Audi	A4 3.0 Quattro 4dr manual	Sedan	Europe	All	$33,430	$30,366
12	Audi	A4 3.0 Quattro 4dr auto	Sedan	Europe	All	$34,480	$31,388
13	Audi	A6 3.0 4dr	Sedan	Europe	Front	$36,640	$33,129
14	Audi	A6 3.0 Quattro 4dr	Sedan	Europe	All	$39,640	$35,992
15	Audi	A4 3.0 convertible 2dr	Sedan	Europe	Front	$42,490	$38,325
16	Audi	A4 3.0 Quattro convertible 2dr	Sedan	Europe	All	$44,240	$40,075
17	Audi	A6 2.7 Turbo Quattro 4dr	Sedan	Europe	All	$42,840	$38,840
18	Audi	A6 4.2 Quattro 4dr	Sedan	Europe	All	$49,690	$44,936
19	Audi	A8 L Quattro 4dr	Sedan	Europe	All	$69,190	$64,740
20	Audi	S4 Quattro 4dr	Sedan	Europe	All	$48,040	$43,556
21	Audi	RS 6 4dr	Sports	Europe	Front	$84,600	$76,417
22	Audi	TT 1.8 convertible 2dr (coupe)	Sports	Europe	Front	$35,940	$32,512
23	Audi	TT 1.8 Quattro 2dr (convertible)	Sports	Europe	All	$37,390	$33,891
24	Audi	TT 3.2 coupe 2dr (convertible)	Sports	Europe	All	$40,590	$36,739
25	Audi	A6 3.0 Avant Quattro	Wagon	Europe	All	$40,840	$37,060
26	Audi	S4 Avant Quattro	Wagon	Europe	All	$49,090	$44,446
27	BMW	X3 3.0i	SUV	Europe	All	$37,000	$33,873

To make the code shorter

%let a=ram;
%let l=sashelp.cars;
	l = library

Data &a;
Set &l;
Run;
	If we are using Sashelp.cars many times in the code, so instead of that we can write &l.
	“&l” will find and replace sashelp.cars everywhere used in the code.

Proc print data=&a;
Run;	


Player
SAS - [VIEWTABLE: Work.Ram]
File Edit View Tools Data Solutions Window Help

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice
1	Acura	MDX	SUV	Asia	All	$36,945	$33,33
2	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,76
3	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,64
4	Acura	TL 4dr	Sedan	Asia	Front	$33,195	$30,29
5	Acura	3.5 RL 4dr	Sedan	Asia	Front	$43,755	$39,01
6	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	$46,100	$41,10
7	Acura	NSX coupe 2dr manual S	Sports	Asia	Rear	$89,765	$79,97
8	Audi	A4 1.8T 4dr	Sedan	Europe	Front	$25,940	$23,50
9	Audi	A41.8T convertible 2dr	Sedan	Europe	Front	$35,940	$32,50
10	Audi	A4 3.0 4dr	Sedan	Europe	Front	$31,840	$28,84
11	Audi	A4 3.0 Quattro 4dr manual	Sedan	Europe	All	$33,430	$30,36
12	Audi	A4 3.0 Quattro 4dr auto	Sedan	Europe	All	$34,480	$31,38
13	Audi	A6 3.0 4dr	Sedan	Europe	Front	$36,640	$33,12
14	Audi	A6 3.0 Quattro 4dr	Sedan	Europe	All	$39,640	$35,99
15	Audi	A4 3.0 convertible 2dr	Sedan	Europe	Front	$42,490	$38,32
16	Audi	A4 3.0 Quattro convertible 2dr	Sedan	Europe	All	$44,240	$40,07
17	Audi	A6 2.7 Turbo Quattro 4dr	Sedan	Europe	All	$42,840	$38,84
18	Audi	A6 4.2 Quattro 4dr	Sedan	Europe	All	$49,690	$44,93
19	Audi	A8 L Quattro 4dr	Sedan	Europe	All	$69,190	$64,74
20	Audi	S4 Quattro 4dr	Sedan	Europe	All	$48,040	$43,55

***;

2.Macro Parameters: Second way of creating macro variables is passing parameters to macro.

%macro yo; **//** Base SAS Proc SQL SQL // **% mend yo;** **%yo;** (call to macro)

%macro a(d=);
 	
data &d;
set sashelp.cars;
run;
 	
%mend a;
	%macro a = is the start of macro where “a” is the name of the macro.
	
	(d=) : d is the parameter will value null.
	(macro will take the input “parameter d “)
	
	Writing &d will make “d” a macro variable.
	
	%mend a = end of macro a
	.
	Till this point macro “a” will compile and a catalogue will form(sasmacr) where definition of macro is stored.

%a(d=kaka);
%a(d=nana);
	%a is the call(trigger) to the macro and the value given to the macro is kaka.
	Here, macro “a” is called and the parameter passed is” kaka”.
	It will go to the catalogue, will pick the macro “a” and in place of “&d” will put the value “kaka”.
	Now the code will be:
	data &kaka;
	set sashelp.cars;
	Run;

	The same way we can change the value from “kaka” to “nana” and can make another dataset.

Note:” & “ is the trigger to the macro variable and “ % “ is the trigger to a macro.

**;

In case we want to make a permament dataset.

```
%macro a(d=);

data &d;
set sashelp.cars;
run;

%mend a;
```

```
%a(d=sasuser.admit); // this will replace &d by sasuser.admit //
```

***;

Passing value while creating a macro:

```
%macro a(d=ram); // this will become hard-cored. Everytime dataset ram will form and the
purpose of creating dynamic shell will not be fulfilled. //
```

```
data &d;
set sashelp.cars;
Run;

%mend a;
```

```
%a(d=kaka);
```

**;

Creating macro with SQL.

```
%macro a(d=);

Proc sql;
Create table &d as select * from sashelp.cars;
quit;

%mend a;
```

```
%a(d=kaka);
%a(d=baba);
```

Note: It is necessary to give the table name in SQL, otherwise will show error whereas in Base SAS if no dataset name is given, it will by default take the name as “Data1”.

Passing two parameters (what to make and from where to make)

```
%macro a(d=,l=);  // Two parameters are passed “d” and “l” //

Data &d;    // Two macro variables are formed “&d” and “l” //
Set &l;
run;

%mend a;
```

```
%a(d=kaka , l=sashelp.cars); // Running this will make the dataset kaka which will be the
```

```
exact copy of sashelp.cars.//

%a(d=route , l=sasuser.cargorev);  // Similarly this will make the dataset route which will be
the exact copy of sasuser.cargorev.//

(No need to write the code from scratch)
```

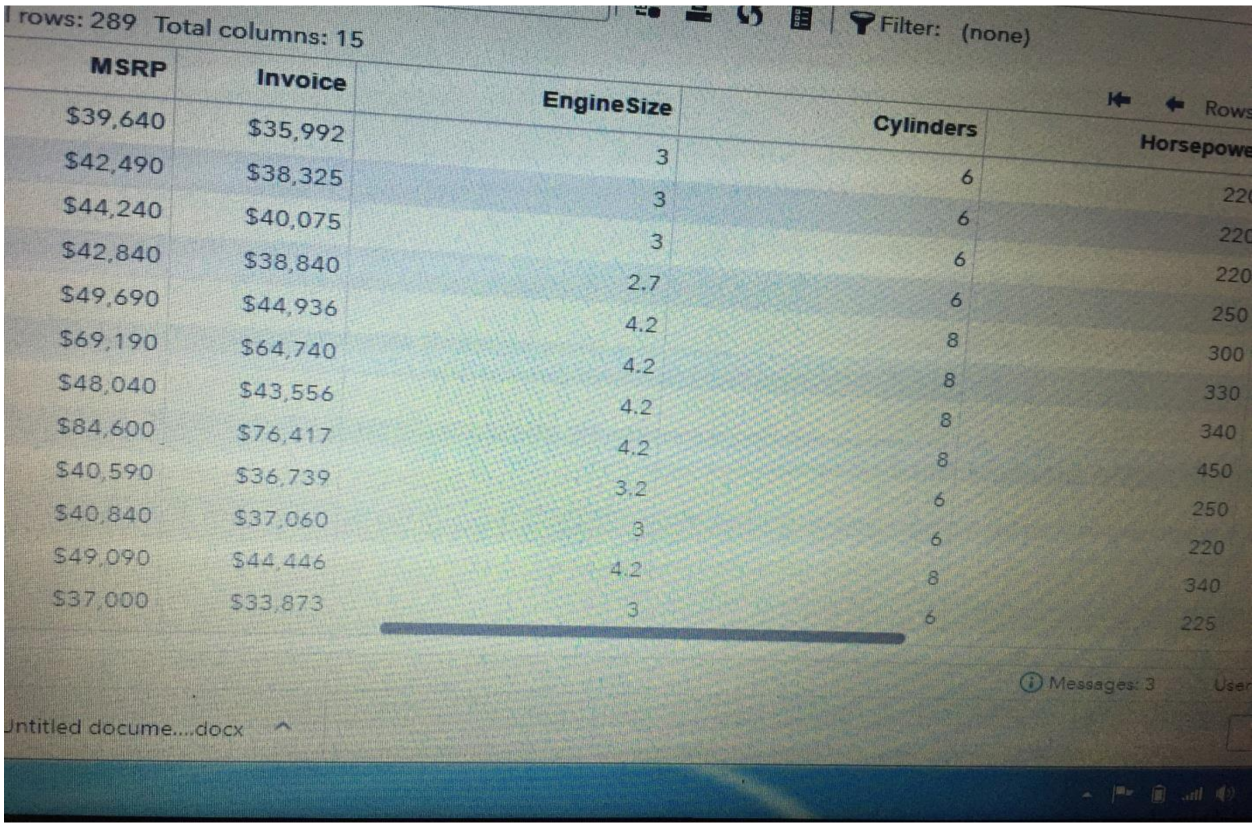
Note: We cannot pass two different values for single parameter like (d=kaka, d=kaka1). This will show error.

Creating above example of macro with SQL.

```
%macro a(d=,l=);

Proc sql;
Create table &d as select * from &l;
quit;

%mend a;
---
%a(d=kaka , l=sashelp.cars);
```

Another example (Text substitution)

```
%let v=Cylinders; // hard-corded value //
---
%macro a(d=,l=,);

Proc sql;
Create table &d as select * from &l where &v gt 4; // Text substitution ( &d=kaka,
&l=sashelp.cars, &v=cylinders) //
quit;

%mend a;
---
%a(d=kaka , l=sashelp.cars);
```


rows: 289 Total columns: 15 Filter: (none)

MSRP	Invoice	EngineSize	Cylinders	Horsepowe
$39,640	$35,992	3	6	22
$42,490	$38,325	3	6	220
$44,240	$40,075	3	6	220
$42,840	$38,840	2.7	6	250
$49,690	$44,936	4.2	8	300
$69,190	$64,740	4.2	8	330
$48,040	$43,556	4.2	8	340
$84,600	$76,417	4.2	8	450
$40,590	$36,739	3.2	6	250
$40,840	$37,060	3	6	220
$49,090	$44,446	4.2	8	340
$37,000	$33,873	3	6	225

***;

Code:

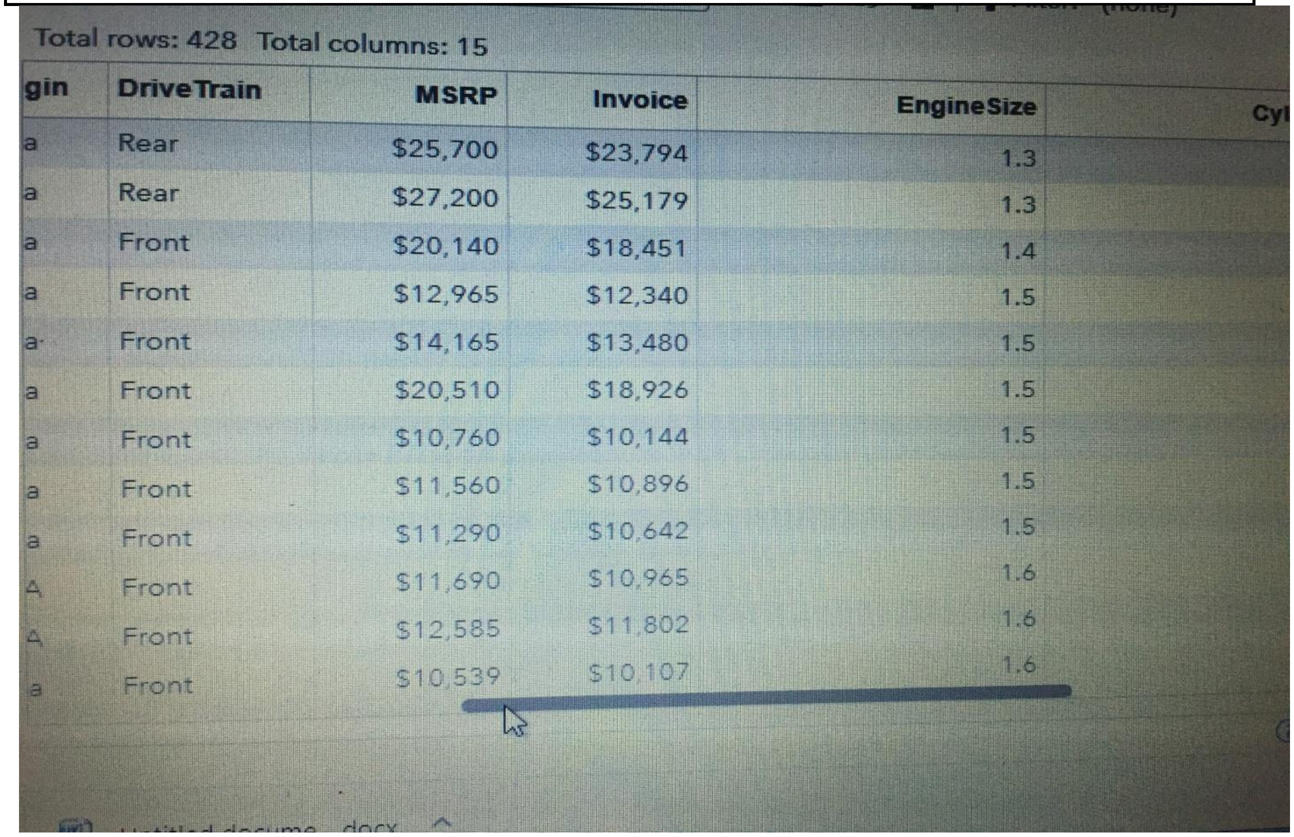
```
%let v=Cylinders;
%let val=4;

%macro a(d=,l=);

Proc sql;
Create table &d as select * from &l where &v gt &val; // val will take the value as 4 , v and val
are created by %let , d and l are created by passing parameters //
quit;

%mend a;

%a(d=kaka , l=sashelp.cars);
```

***;

Create "kaka" from sashelp.cars sorted by EngineSize:

```
%macro sortany(d=, l=, v=); // Three parameters are passed= d, l, v //

Proc sort data = &l  out=d;
By &v;
Run;

%mend sortany;
```

```
%sortany(d=kaka , l=sashelp.cars, v=EngineSize);
// Kaka will be created from sashelp.cars sorted by enginesize //

%sortany(d=kaka , l=sashelp.cars, v=Origin); // The same way we can sort by Origin and
Type) //

%sortany(d=kaka , l=sashelp.cars, v=Type);
```



Total rows: 428 Total columns: 15

gin	DriveTrain	MSRP	Invoice	EngineSize	Cyl
a	Rear	$25,700	$23,794	1.3	
a	Rear	$27,200	$25,179	1.3	
a	Front	$20,140	$18,451	1.4	
a	Front	$12,965	$12,340	1.5	
a	Front	$14,165	$13,480	1.5	
a	Front	$20,510	$18,926	1.5	
a	Front	$10,760	$10,144	1.5	
a	Front	$11,560	$10,896	1.5	
a	Front	$11,290	$10,642	1.5	
A	Front	$11,690	$10,965	1.6	
A	Front	$12,585	$11,802	1.6	
a	Front	$10,539	$10,107	1.6	

**;

Creating above example of macro with SQL.

```
%macro sortany(d=,l=,v=);

Proc sql;
```

```
Create table &d as select * from &l order by &v;
quit;

%mend sortany;
```

```
%a(d=kaka , l=sashelp.cars);
```