[A BRIEF ON SAS]

SOMA S DHAVALA

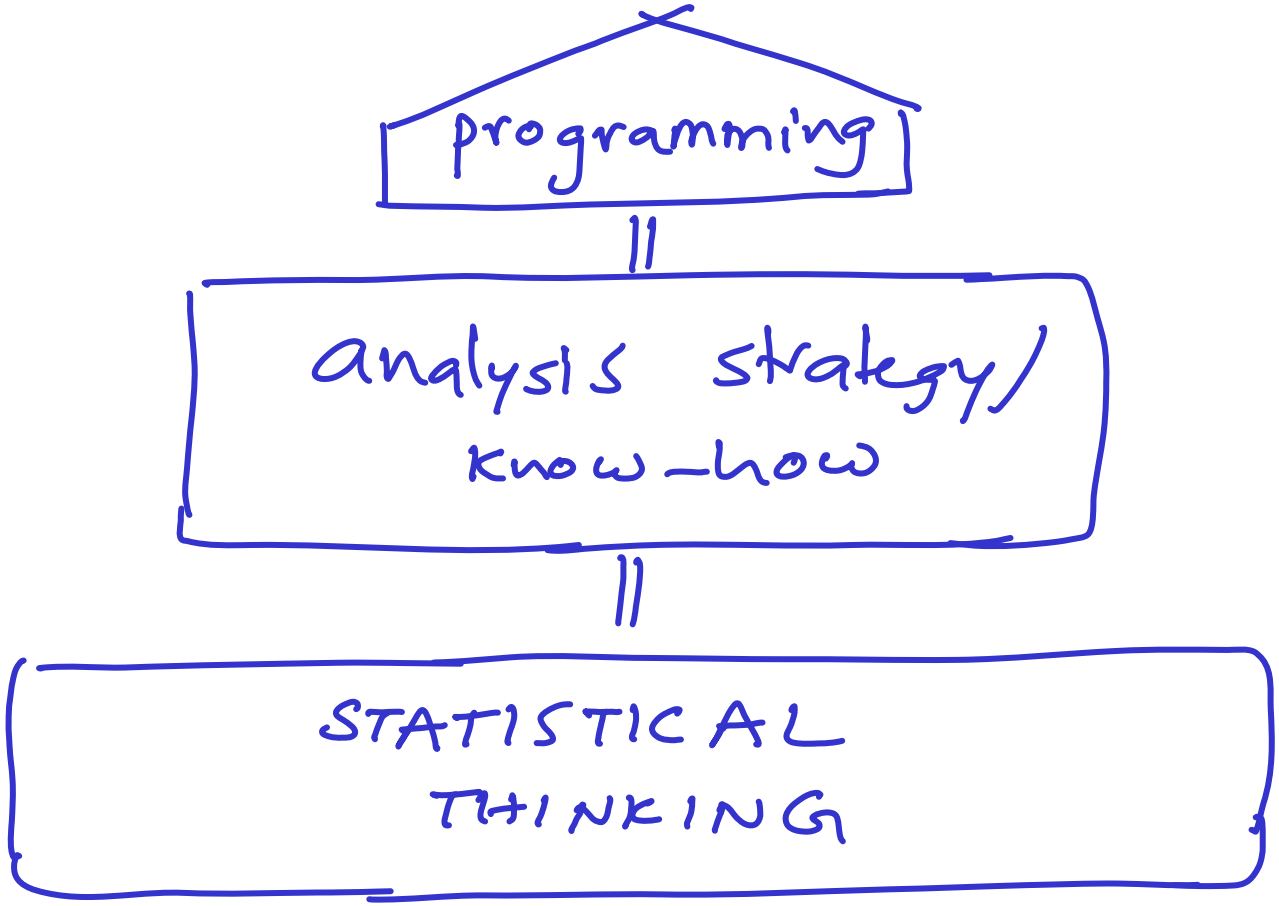
STAT DEPT

TAMU

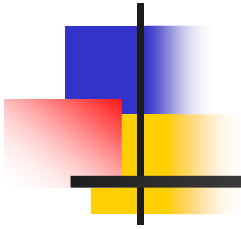
– Soma

Thinking Statistically!

A Brief on SAS



– Soma



Thinking Statistically!

A Brief on SAS

→ Given that today is raining, the traffic might be hit

→ you bough a car for $1700? that's a good deal

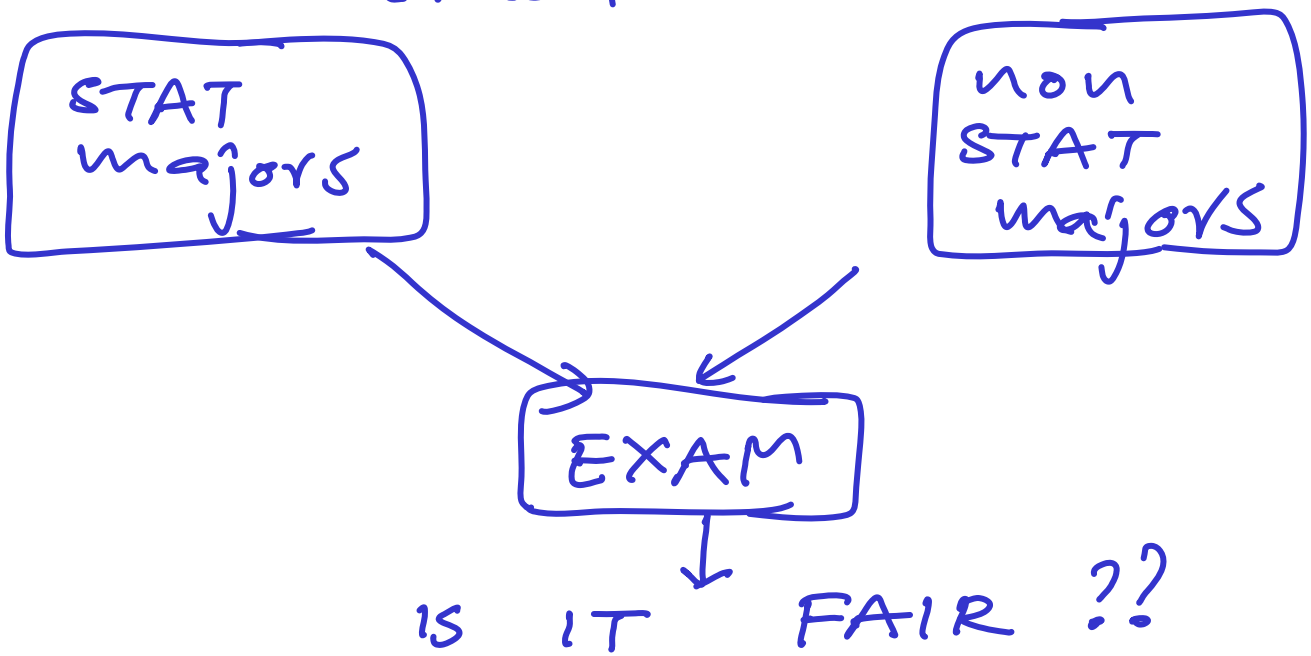
→ I'm not surprised that (S)he was "awarded" the Nobel prize

Thinking Statistically!

A Brief on SAS

→ Don't you think that the president is responsible for this?

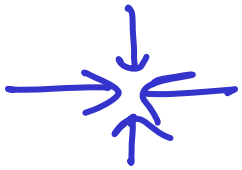
Suppose: evalate the teaching method



Statistically Speaking

A Brief on SAS

* Life is full of uncertainty
* STATS quantifies "



* Inherently, we are doubly stochastic
* STATS brings in formalism thus removing "subjectivity"

My 'learning strategy:

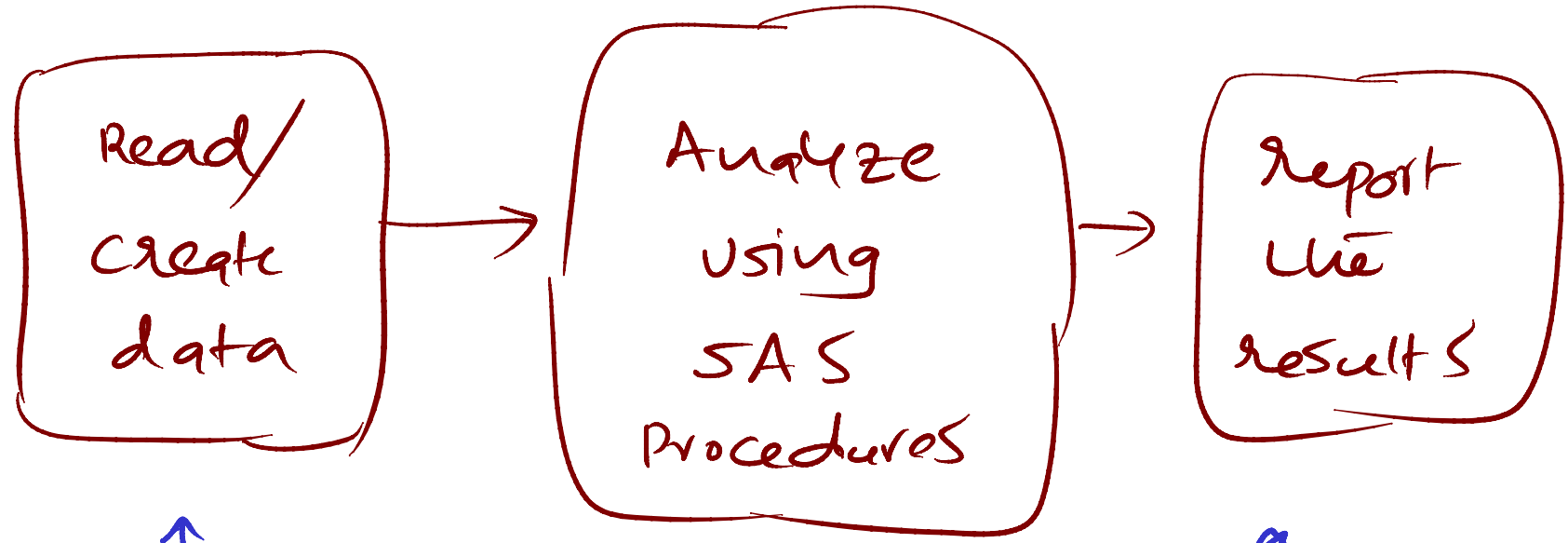
* Define a problem
* formulate the solution
* THEN "program"

* Take a Sample code from "net"
* Tweak the parameters/ play with them
* Test hypothesis

SAS :

A Brief on SAS

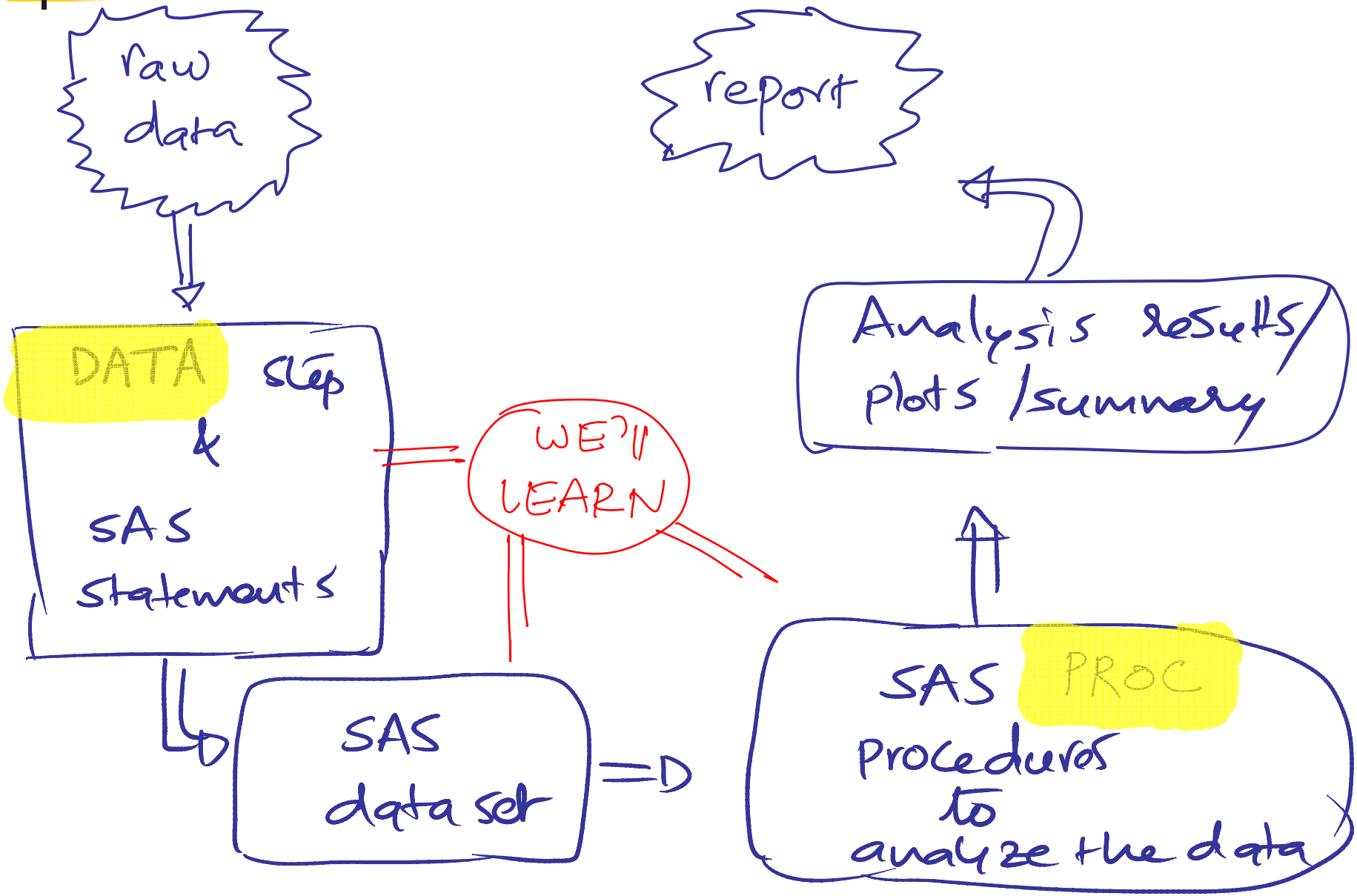
Statistical Analysis Software



Contd..

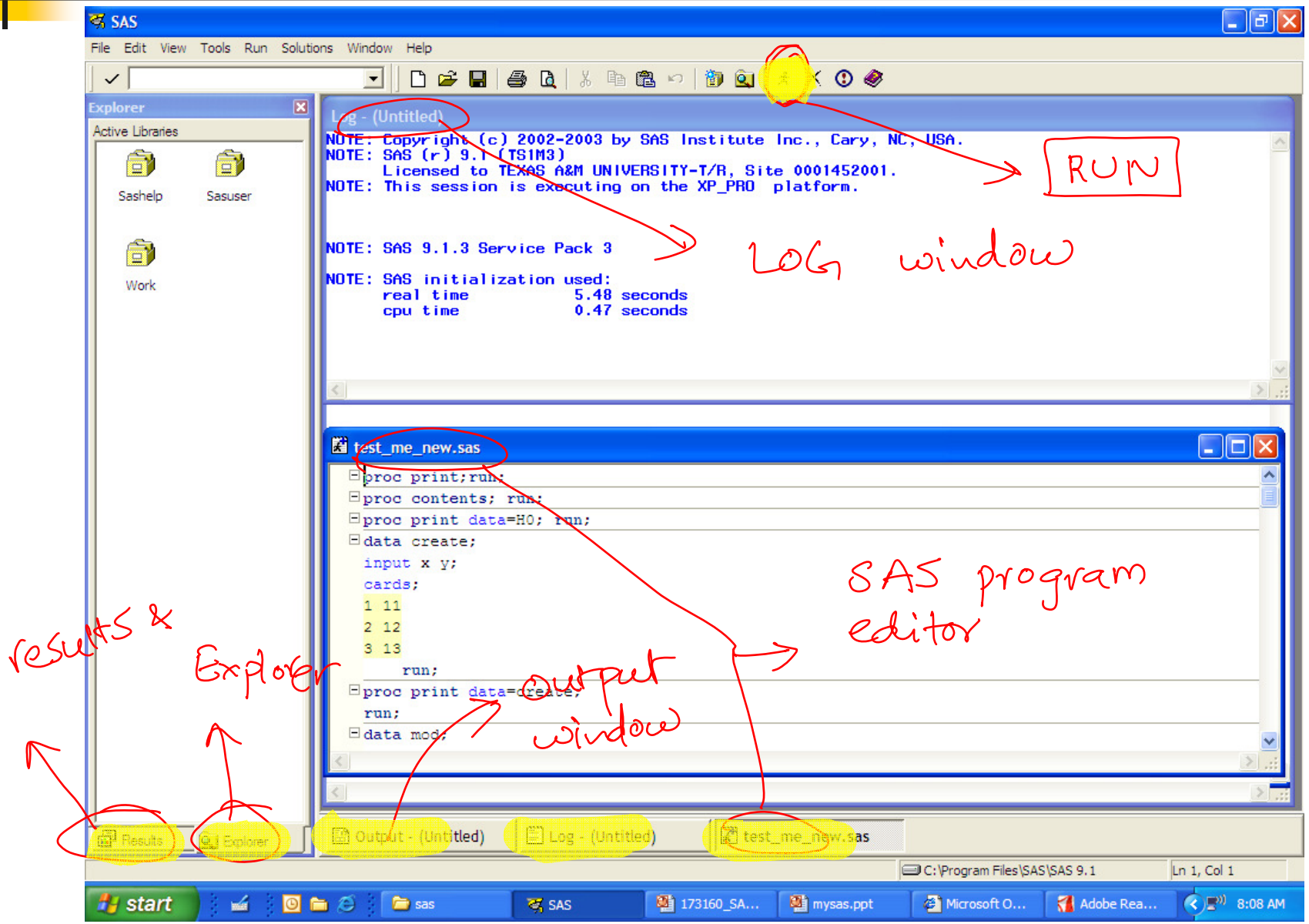
SAS :

A Brief on SAS

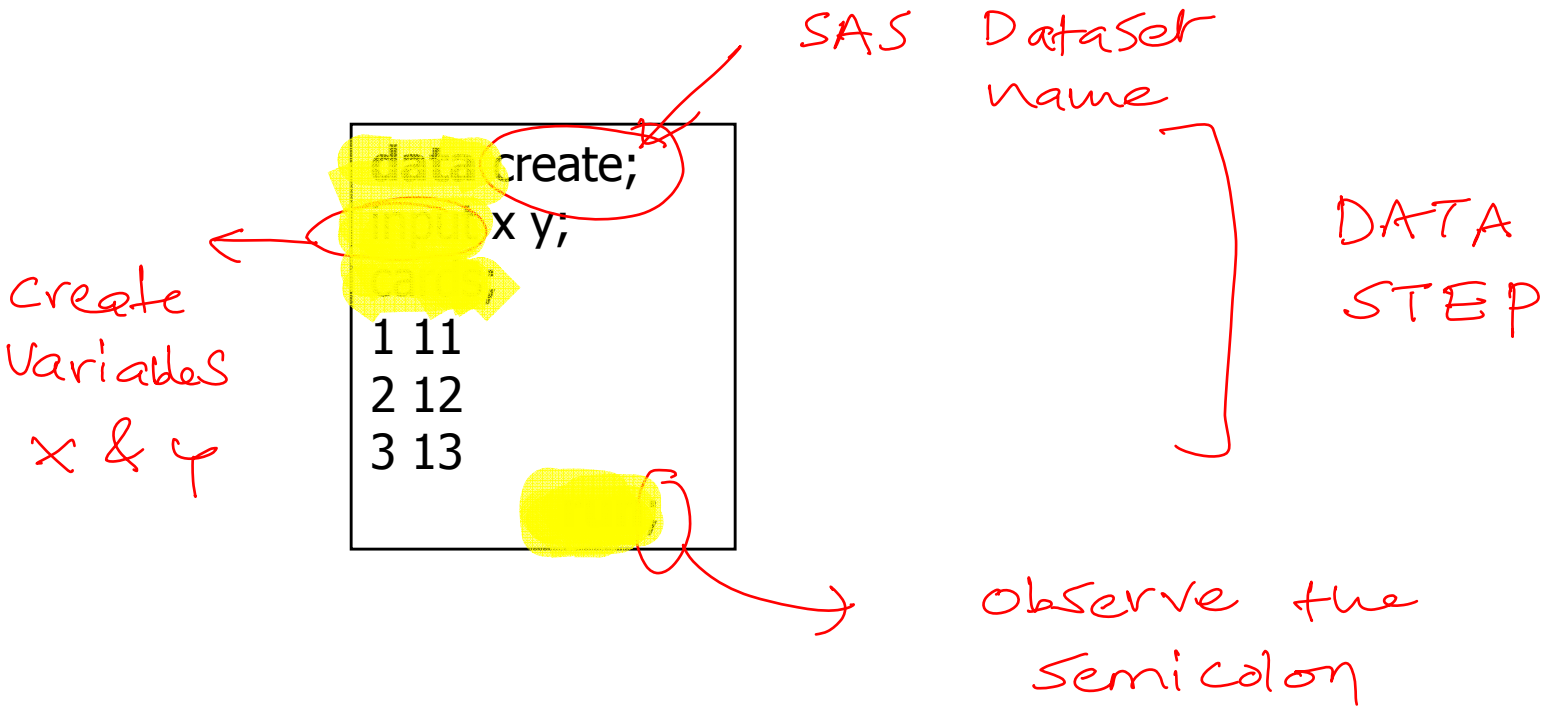


SAS : Environment

A Brief on SAS

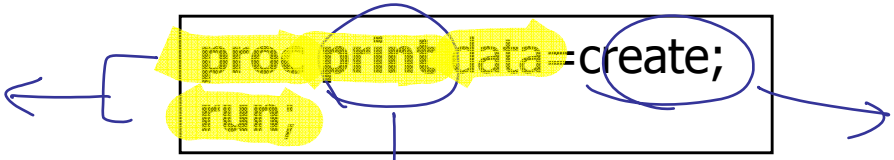


Create Data:



† Shaded in yellow are Keywords

Print Data:



Remember the run at the end

procedure print

data set that we want to view

O/P

obs	x	Y
1	1	11
2	2	12
3	3	13

ADD Data:


```
data mod;
set create;
z = y+2;
run;
proc print data=mod;run;
```

Arithmatic operation

O/p

obs	x	y	z
1	1	11	13
2	2	12	14
3	3	13	15

Set is like a copy command

data mod
set create
} ⇒ mod = create

MANAGE Data:

A Brief on SAS

x	z
2	0
3	0

o/p ⇐

```
data mod;
set create;
z = y+2;
if (x>1) AND (y>11) then z = 0;
keep x z;
label x = 'ind variable' z= 'new variable' z
= 'response';
run;
```

out of x, y, z only retain x & z

x>1, y≥11 when z=0

Label is additional tag that describes the variable

MANAGE Data:

KEEP OK, if what we want to keep are less

Logical alternative ?

proc contents; run; → TRY THIS & SEE WHAT ?

Read Data:

A Brief on SAS

LIST BASED

what we learnt earlier had limitations (limited to 8 columns at a time per variable)

anything inserted between "*" and ";" is a comment

comment

```
data tab;
* column based data reading;
input name $ 1-4 x 6-7 y 9;
cards;
soma 22 3
ram  3  5
run;
```

character variable

column numbers

Manage Data:


```
data tab_new;
set tab;
if name ^= 'bond' then delete;
run;
```

alternative to KEEP

drop

delete

what is the difference ??

i/p

name	x	y
junk	2	3
bond	4	5

o/p ?

name	x	y

??

proc contents; run;

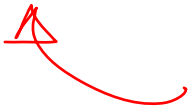
Manage Data:


```
data tab_new;
set tab;
if name ^= 'bond' then delete;
run;
```

⟹ two-pass scheme

holds the records for processing

one-pass scheme



```
data tab_very_new;
input name $ @;
if name ^= 'ram' then delete;
input x y;
cards;
soma 2 3
ram 44 53
run;
```

Collapse Data:

name	X	Y
—	1	4
—	2	5
—	3	6

i/p

```
data pack;
set tab;
z = x;
output;
z = y;
output;
drop x y;
run;
```

o/p's "z"

o/p's "z"

removes x & y

name	Z
—	1
—	2
—	3
—	4
—	5
—	6

o/p

(X) Check-out

→ Set data1, data2

→ merge procedures

Format Data:

w.d

→ w total columns length

→ d # of decimal places (10^{-d}, only if no actual decimal present)

```
data formated;
input x 6.2 y 7.;
cards;
1.23  1.72
12.3  17.2
123.3 172
1230. 1722.4
12334 17233.5
run;
```

⊗ all "y" fields are aligned.

X	y
1.23	1.72
12.3	17.2
123.3	172
1230.	1722.4
12334	17233.5

Sort Data: (procedure)

- useful in merging two data sets

```
data grades;
input name $ 1-5 grade $ 6;
cards;
soma c
josh a
run;
proc print data=grades;run;
proc sort data=grades;
by name;
run;
```

```
data _arith_;
x = 20;
y = sign(x);
put x= y= ;
run;
```

⇒ useful in debugging prints in the LOG window

Permanent Data:


```
data Mylib.reg;
set Mylib.age;
drop sex;
x = age; drop age;
do _n_ = 1 to 5;
y = (3*x) + (0.01*rannor(1));
keep x y;
end;
proc print;
run;
```

do] loop
end

- Two level name saves the data-set permanently,

else everything is stored temporarily in the work space, that you'll loose at the end of the session

Nested PROC within in the Data step

Plot Data:

(procedure)

```
proc gplot data=Mylib.reg;
symbol v = circle;
plot x*y;
title 'scatter plot of x vs y';
run;
```

- Scatter plot for visualization

to observe any trends

Analyze Data:

(Regression: Proc)

```
proc reg data=temp_reg;
model y=x x2;
title 'regressing of y on x';
run;
```

* we just've created data

  $Y = mx + c + \text{noise}$

* a simple linear regression

Analyze Data:

* Creating
- data from an external file

```
data Mylib.age;
infile 'c:\soma\tamu\vol\sas\test_data.txt';
input sex $ age;
run;

proc print data=Mylib.age;run;
proc sort data=Mylib.age; by sex; run;
proc means data=Mylib.age;
by sex;
run;
```

* Sorting the data using "SORT PROC" by SEX
* Using "means" procedure to compute descriptive statistics per category

Lot more

* arrays
* functions
* numerous procedures
* string operations
* SAS Enterprise Guide

Acknowledgements

* WWW
* INFORMS
* Vijaya Kancherla (U-Iowa)
* My Roomate for waking me up from deep-sleep

Thank-you!