MAT7500
Statistical Programming
Spring 2017

Michael A. Posner
SAS Lecture 2

Goals for Today

- **Review from last class**
- **Data manipulation**
- **Generating data (for simulations)**
- **Checking data**
- **Example using SAS – a simulation!**

Review from Last Class

- Reading in data – libname, infile, etc.
- Attributes of data – format, label
- Combining data – sort, set, merge

Practice Problems #1 (002)

- Review problem 1
- Do problems 2 and 3

Assignment (creating variables)

- Must be done within a DATA step
- Numeric or Character

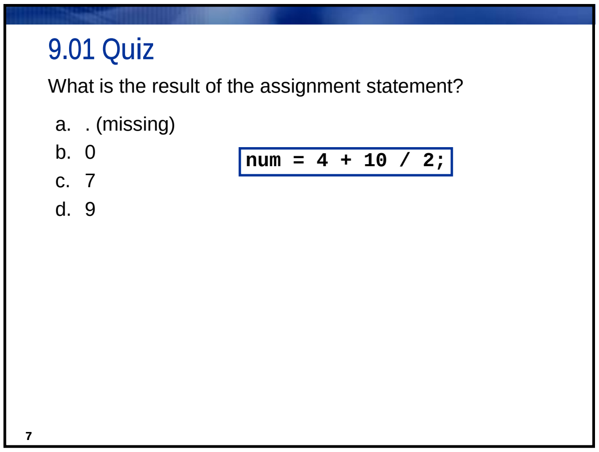
  filenumber = 1; file = ‘file 1’;
- Arithmetic operators (+,-,*,/,**)

  increase = posttest – pretest;

  missing data outputs missing result
- SUM(Var1,Var2,etc.), among others
  - Adds variables, but excludes missing

Practice Problem

- From problem 3, instead of reading in a dataset called *school*, create in *schoolA* which reads in the five names of the kids in school A and *schoolB* which reads in the three names of the kids in school B. For each one, add a variable that identifies the school, then combine these datasets together to recreate the original *school* dataset.



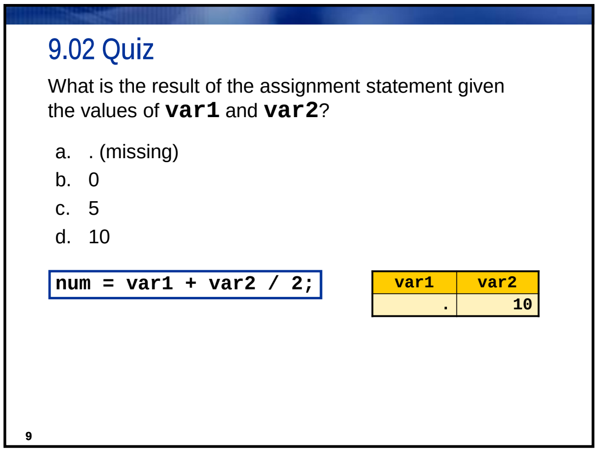
9.01 Quiz

What is the result of the assignment statement?

a. . (missing)
b. 0
c. 7
d. 9

```
num = 4 + 10 / 2;
```

7



9.02 Quiz

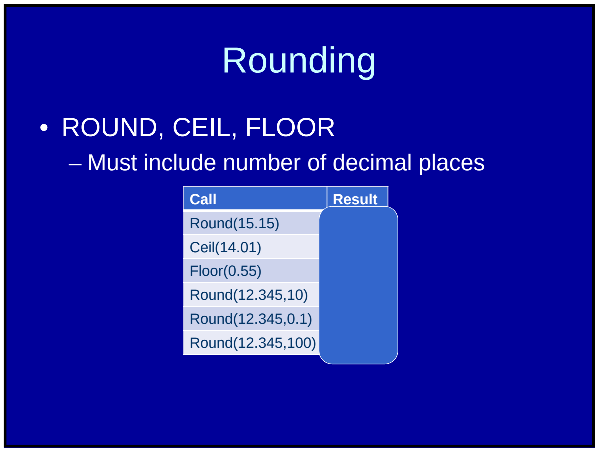
What is the result of the assignment statement given the values of `var1` and `var2`?

a. . (missing)
b. 0
c. 5
d. 10

```
num = var1 + var2 / 2;
```

var1	var2
.	10

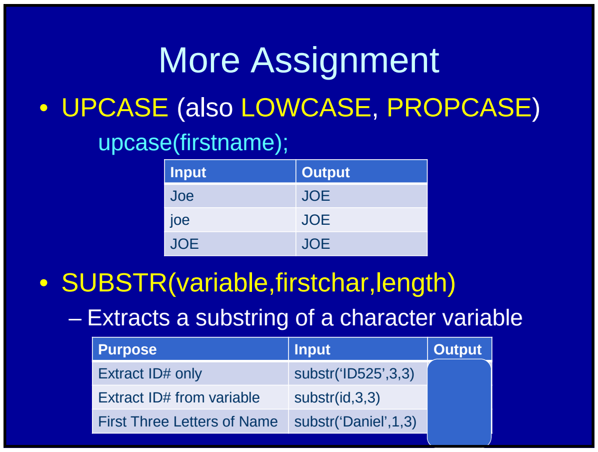
9



Rounding

- ROUND, CEIL, FLOOR
  - Must include number of decimal places

Call	Result
Round(15.15)	
Ceil(14.01)	
Floor(0.55)	
Round(12.345,10)	
Round(12.345,0.1)	
Round(12.345,100)	



More Assignment

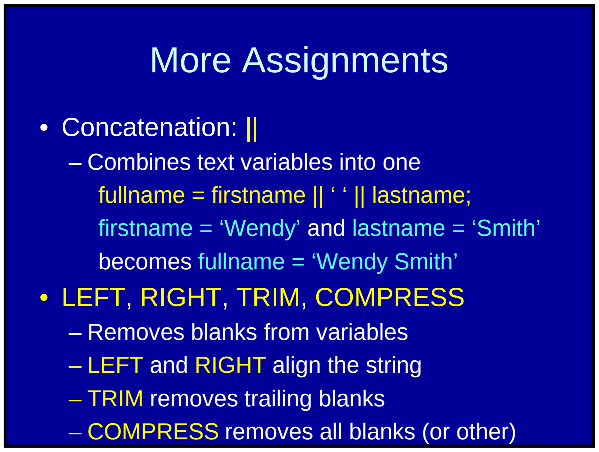
- UPCASE (also LOWCASE, PROPCASE)

  upcase(firstname);

Input	Output
Joe	JOE
joe	JOE
JOE	JOE

- SUBSTR(variable,firstchar,length)
  - Extracts a substring of a character variable

Purpose	Input	Output
Extract ID# only	substr('ID525',3,3)	
Extract ID# from variable	substr(id,3,3)	
First Three Letters of Name	substr('Daniel',1,3)	



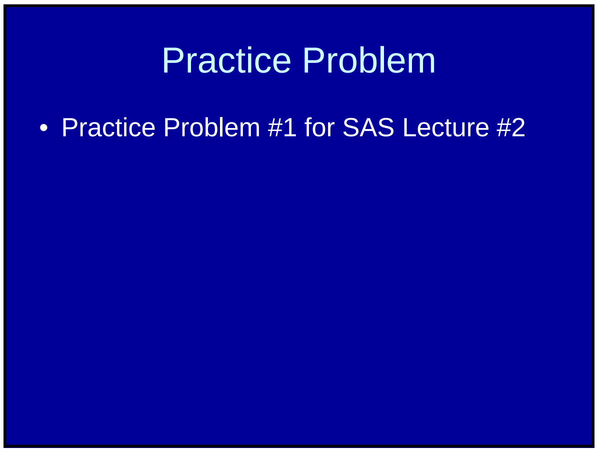
More Assignments

- Concatenation: ||
  - Combines text variables into one

    fullname = firstname || ' ' || lastname;

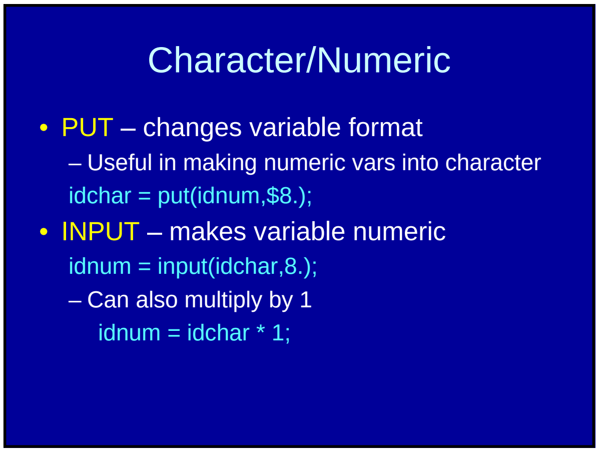
    firstname = 'Wendy' and lastname = 'Smith'

    becomes fullname = 'Wendy Smith'
- LEFT, RIGHT, TRIM, COMPRESS
  - Removes blanks from variables
  - LEFT and RIGHT align the string
  - TRIM removes trailing blanks
  - COMPRESS removes all blanks (or other)



Practice Problem

- Practice Problem #1 for SAS Lecture #2



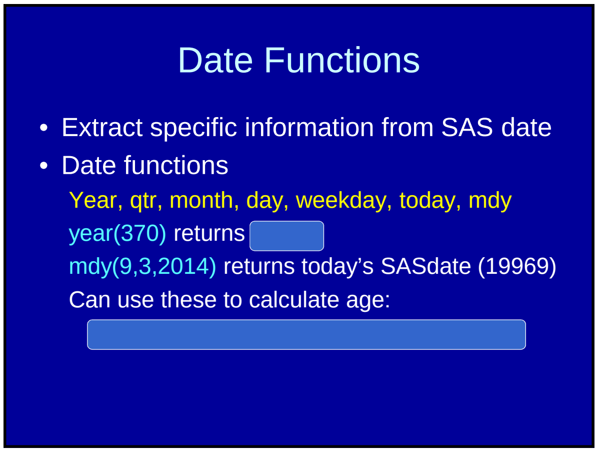
Character/Numeric

- PUT – changes variable format
  - Useful in making numeric vars into character

    idchar = put(idnum,$8.);
- INPUT – makes variable numeric

  idnum = input(idchar,8.);
  - Can also multiply by 1

    idnum = idchar * 1;



Date Functions

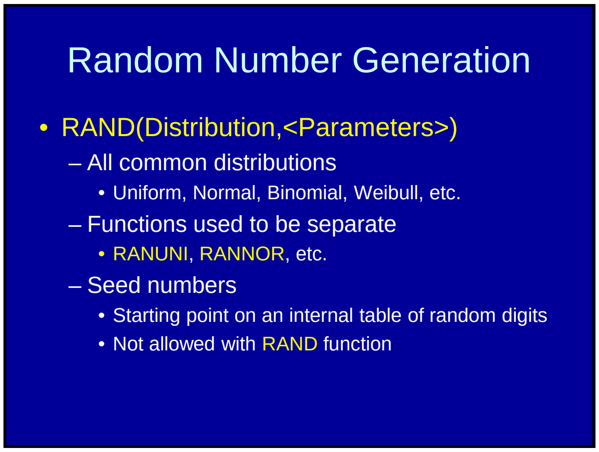
- Extract specific information from SAS date
- Date functions

  Year, qtr, month, day, weekday, today, mdy

  year(370) returns

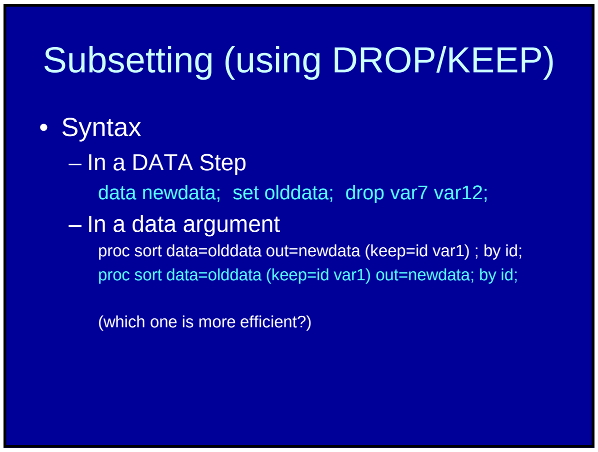
  mdy(9,3,2014) returns today's SASdate (19969)

  Can use these to calculate age:



Random Number Generation

- RAND(Distribution,<Parameters>)
  - All common distributions
    - Uniform, Normal, Binomial, Weibull, etc.
  - Functions used to be separate
    - RANUNI, RANNOR, etc.
  - Seed numbers
    - Starting point on an internal table of random digits
    - Not allowed with RAND function



Subsetting (using DROP/KEEP)

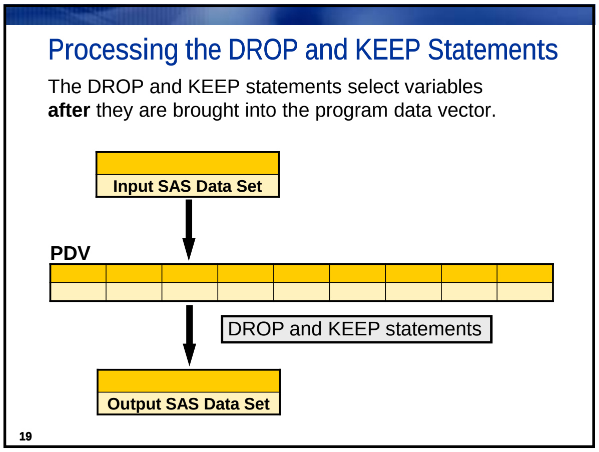
- Syntax
  - In a DATA Step

    data newdata; set olddata; drop var7 var12;
  - In a data argument

    proc sort data=olddata out=newdata (keep=id var1) ; by id;

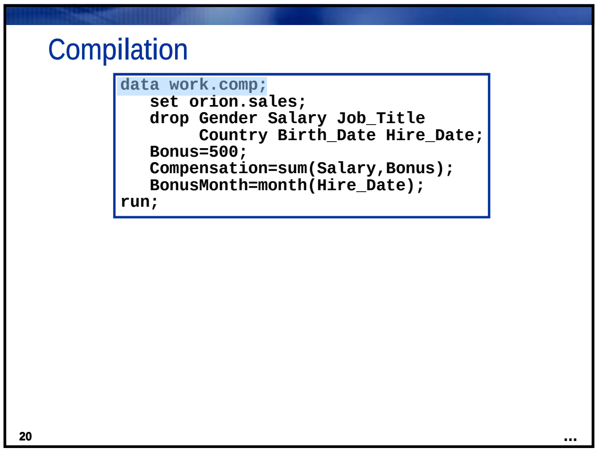
    proc sort data=olddata (keep=id var1) out=newdata; by id;

    (which one is more efficient?)



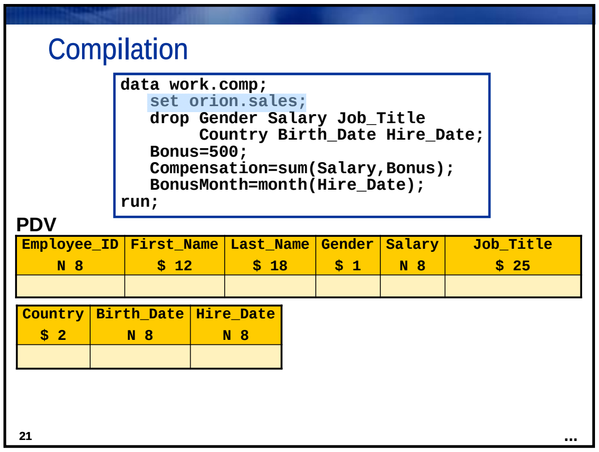
Processing the DROP and KEEP Statements

The DROP and KEEP statements select variables **after** they are brought into the program data vector.



Compilation

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```



Compilation

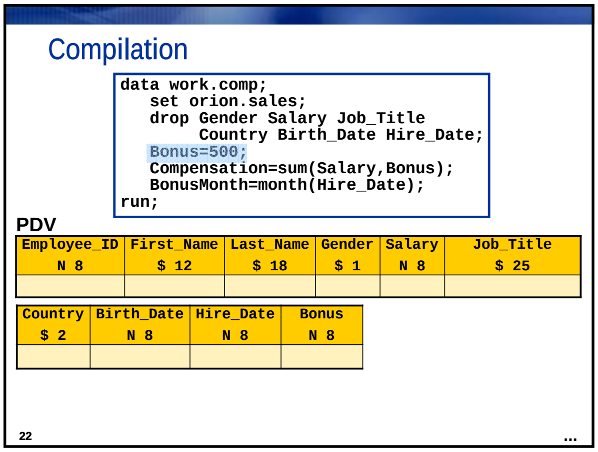
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title
N 8	$ 12	$ 18	$ 1	N 8	$ 25

Country	Birth_Date	Hire_Date
$ 2	N 8	N 8

21 ...



Compilation

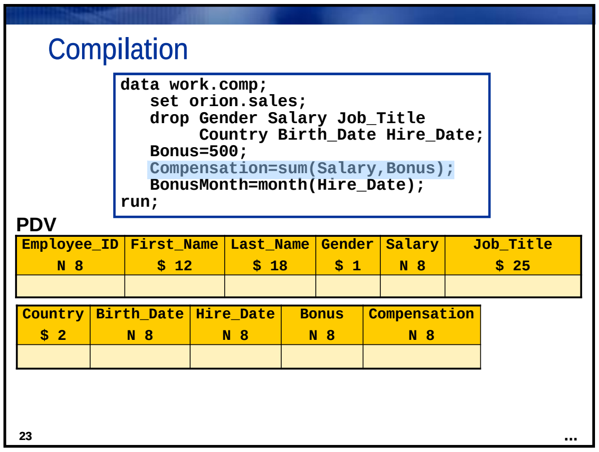
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title
N 8	$ 12	$ 18	$ 1	N 8	$ 25

Country	Birth_Date	Hire_Date	Bonus
$ 2	N 8	N 8	N 8

22 ...



Compilation

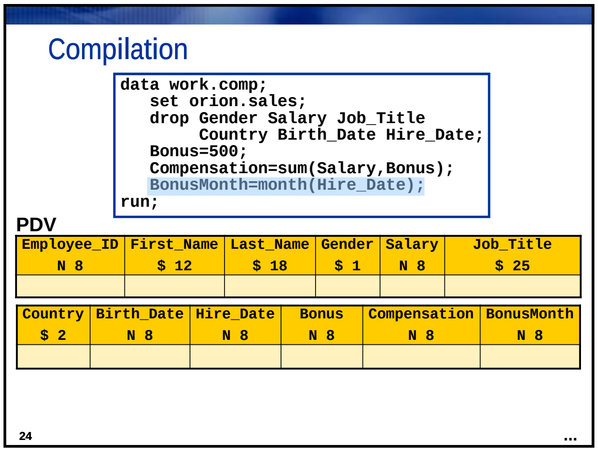
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title
N 8	$ 12	$ 18	$ 1	N 8	$ 25

Country	Birth_Date	Hire_Date	Bonus	Compensation
$ 2	N 8	N 8	N 8	N 8

23 ...



Compilation

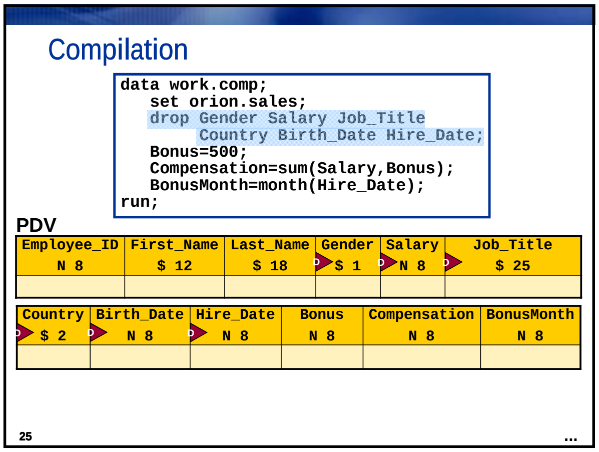
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title
N 8	$ 12	$ 18	$ 1	N 8	$ 25

Country	Birth_Date	Hire_Date	Bonus	Compensation	BonusMonth
$ 2	N 8	N 8	N 8	N 8	N 8

24 ...



Compilation

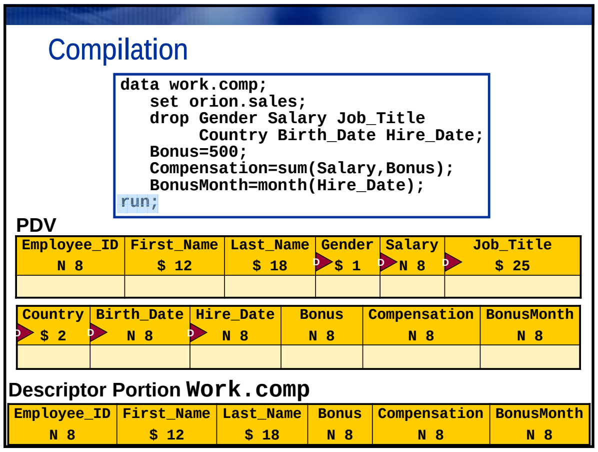
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title
N 8	$ 12	$ 18	D▶ $ 1	D▶ N 8	D▶ $ 25

Country	Birth_Date	Hire_Date	Bonus	Compensation	BonusMonth
D▶ $ 2	D▶ N 8	D▶ N 8	N 8	N 8	N 8

25 ...



Compilation

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

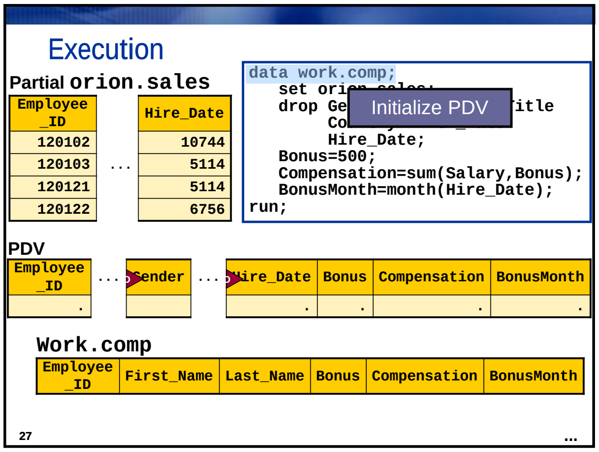
PDV

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title
N 8	$ 12	$ 18	D▶ $ 1	D▶ N 8	D▶ $ 25

Country	Birth_Date	Hire_Date	Bonus	Compensation	BonusMonth
D▶ $ 2	D▶ N 8	D▶ N 8	N 8	N 8	N 8

Descriptor Portion `Work.comp`

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
N 8	$ 12	$ 18	N 8	N 8	N 8



Execution

Partial `orion.sales`

Employee_ID	...	Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

```
data work.comp;
   set orion.sales;
   drop Ge                  Title
        Co
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

Initialize PDV

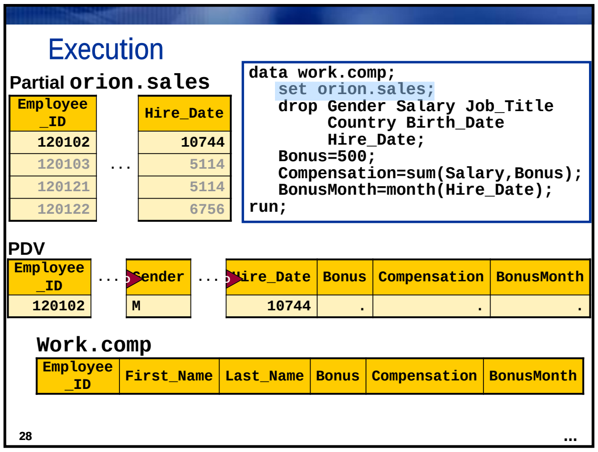
PDV

Employee_ID	...	▶ender	...	▶ire_Date	Bonus	Compensation	BonusMonth
.				.	.	.	.

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth

27 ...



Execution

Partial `orion.sales`

Employee_ID	...	Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

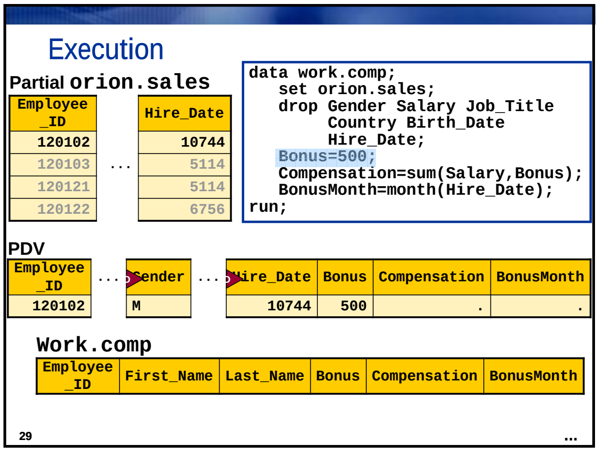
PDV

Employee_ID	...	▶ender	...	▶ire_Date	Bonus	Compensation	BonusMonth
120102		M		10744	.	.	.

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth

28 ...



Execution

Partial `orion.sales`

Employee_ID	...	Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

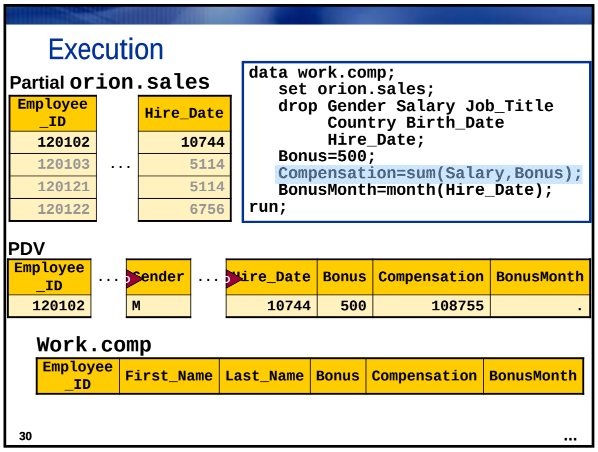
PDV

Employee_ID	...	▶ender	...	▶ire_Date	Bonus	Compensation	BonusMonth
120102		M		10744	500	.	.

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth

29 ...



Execution

Partial `orion.sales`

Employee_ID	...	Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

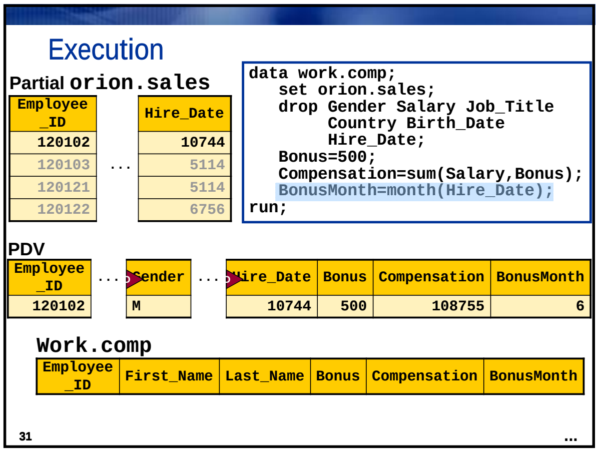
PDV

Employee_ID	...	▶ender	...	▶ire_Date	Bonus	Compensation	BonusMonth
120102		M		10744	500	108755	.

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth

30 ...



Execution

Partial `orion.sales`

Employee_ID	...	Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

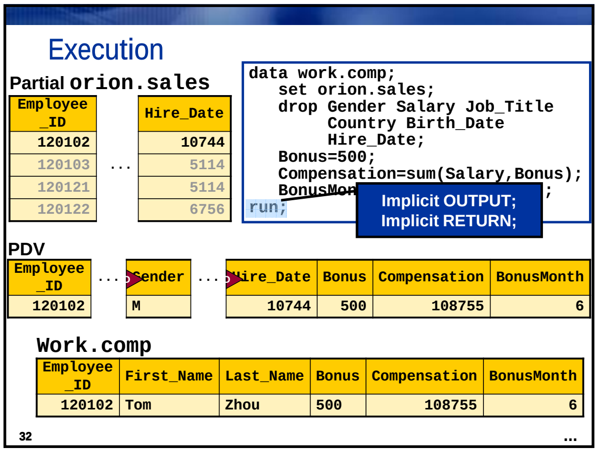
PDV

Employee_ID	...	▶ender	...	▶ire_Date	Bonus	Compensation	BonusMonth
120102		M		10744	500	108755	6

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth

31 ...



Execution

Partial `orion.sales`

Employee_ID	...	Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMon                  ;
run;
```

Implicit OUTPUT;
Implicit RETURN;

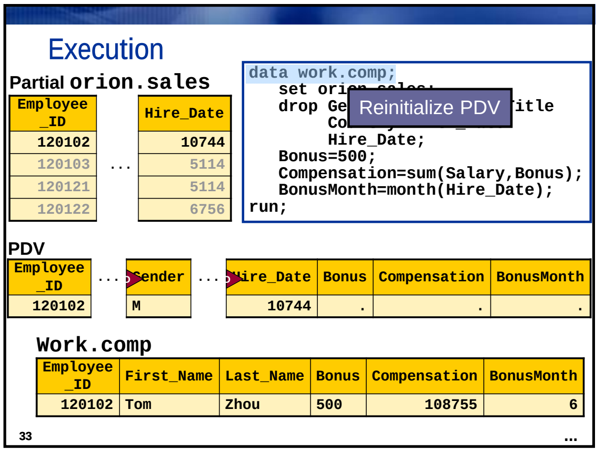
PDV

Employee_ID	...	▶ender	...	▶ire_Date	Bonus	Compensation	BonusMonth
120102		M		10744	500	108755	6

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
120102	Tom	Zhou	500	108755	6

32 ...



Execution

Partial orion.sales

Employee_ID		Hire_Date
120102	...	10744
120103		5114
120121		5114
120122		6756

```
data work.comp;
   set orion.sales;
   drop Ge                  itle
        Co
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

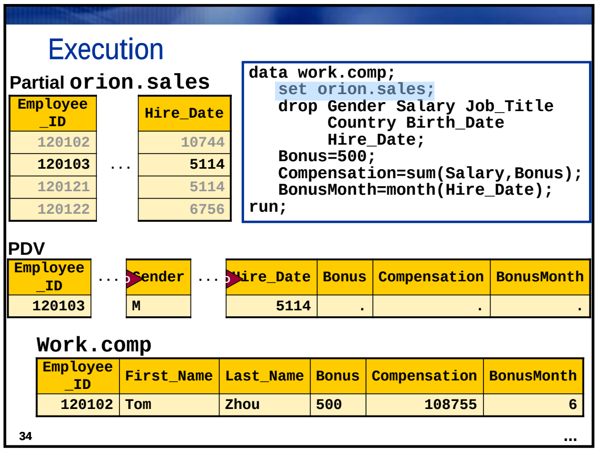
Reinitialize PDV

PDV

Employee_ID		Gender		Hire_Date	Bonus	Compensation	BonusMonth
120102	...	M	...	10744	.	.	.

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
120102	Tom	Zhou	500	108755	6



Execution

Partial orion.sales

Employee_ID		Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

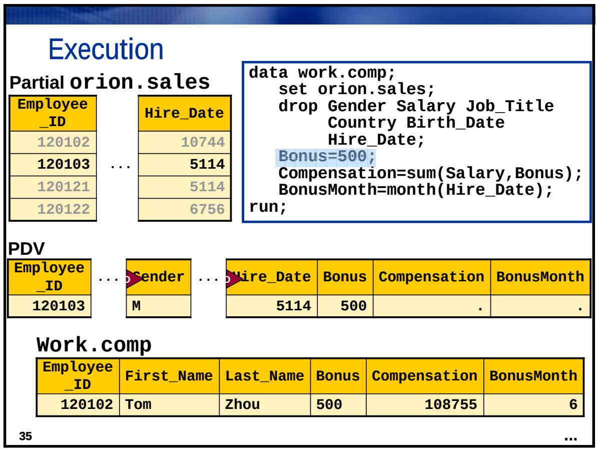
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID		Gender		Hire_Date	Bonus	Compensation	BonusMonth
120103	...	M	...	5114	.	.	.

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
120102	Tom	Zhou	500	108755	6



Execution

Partial orion.sales

Employee_ID		Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

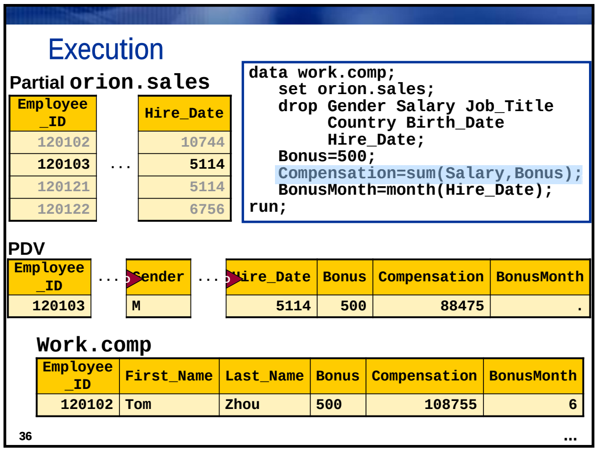
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID		Gender		Hire_Date	Bonus	Compensation	BonusMonth
120103	...	M	...	5114	500	.	.

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
120102	Tom	Zhou	500	108755	6



Execution

Partial orion.sales

Employee_ID		Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

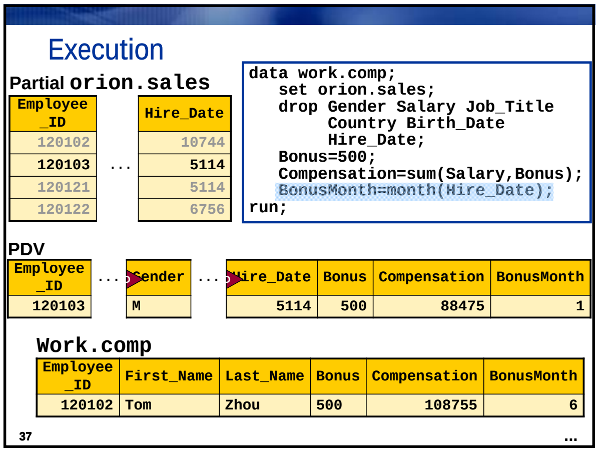
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID		Gender		Hire_Date	Bonus	Compensation	BonusMonth
120103	...	M	...	5114	500	88475	.

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
120102	Tom	Zhou	500	108755	6



Execution

Partial orion.sales

Employee_ID		Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

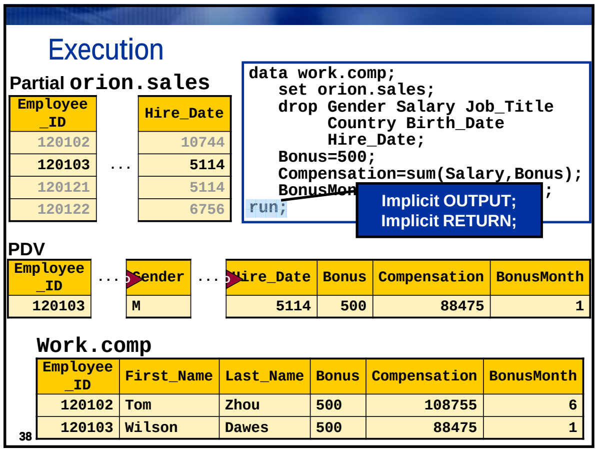
```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

PDV

Employee_ID		Gender		Hire_Date	Bonus	Compensation	BonusMonth
120103	...	M	...	5114	500	88475	1

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
120102	Tom	Zhou	500	108755	6



Execution

Partial orion.sales

Employee_ID		Hire_Date
120102		10744
120103	...	5114
120121		5114
120122		6756

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date
        Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMon                       ;
run;
```

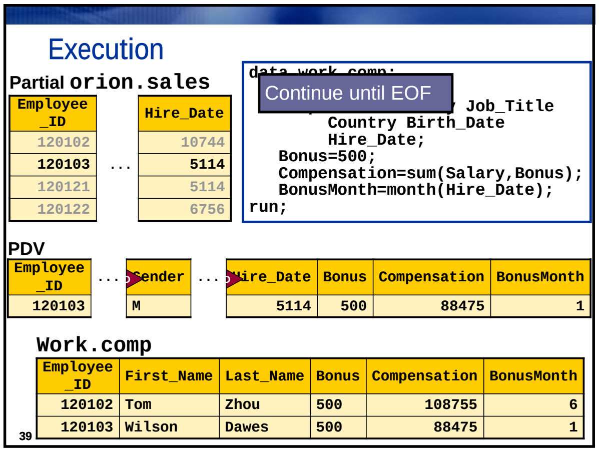
Implicit OUTPUT;
Implicit RETURN;

PDV

Employee_ID		Gender		Hire_Date	Bonus	Compensation	BonusMonth
120103	...	M	...	5114	500	88475	1

Work.comp

Employee_ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
120102	Tom	Zhou	500	108755	6
120103	Wilson	Dawes	500	88475	1



Execution

Partial **orion.sales**

Employee _ID
120102
120103
120121
120122

...

Hire_Date
10744
5114
5114
6756

```
data work.comp;
                  Job_Title
      Country Birth_Date
      Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```

Continue until EOF

PDV

Employee _ID
120103

...

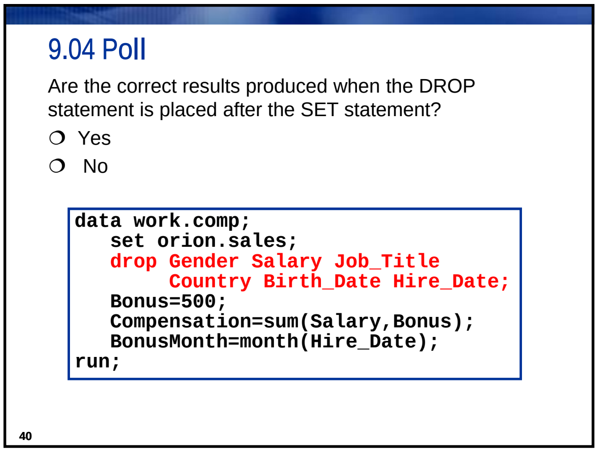
Gender
M

...

Hire_Date	Bonus	Compensation	BonusMonth
5114	500	88475	1

Work.comp

Employee _ID	First_Name	Last_Name	Bonus	Compensation	BonusMonth
120102	Tom	Zhou	500	108755	6
120103	Wilson	Dawes	500	88475	1



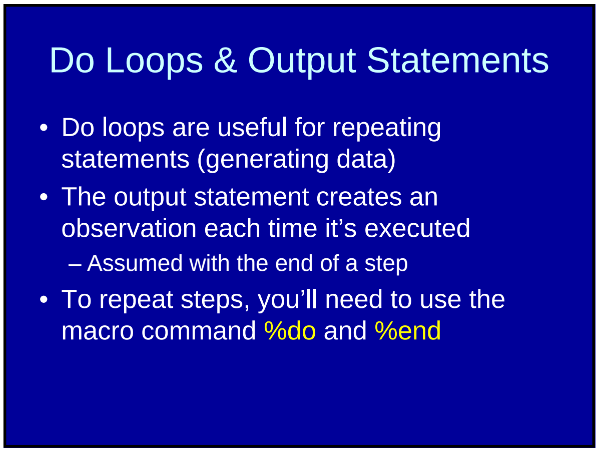
9.04 Poll

Are the correct results produced when the DROP statement is placed after the SET statement?

○ Yes

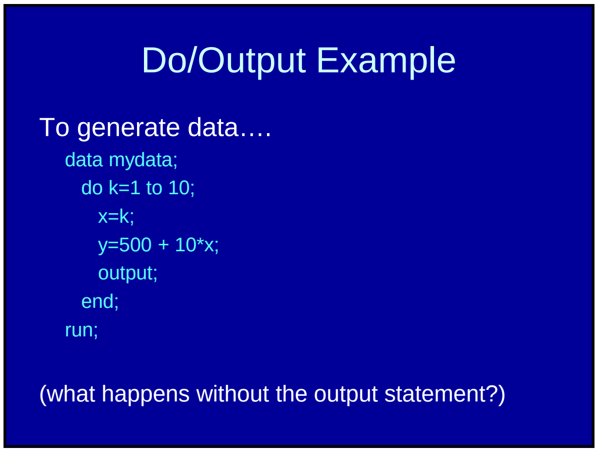
○ No

```
data work.comp;
   set orion.sales;
   drop Gender Salary Job_Title
        Country Birth_Date Hire_Date;
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   BonusMonth=month(Hire_Date);
run;
```



Do Loops & Output Statements

- Do loops are useful for repeating statements (generating data)
- The output statement creates an observation each time it's executed
  - Assumed with the end of a step
- To repeat steps, you'll need to use the macro command %do and %end

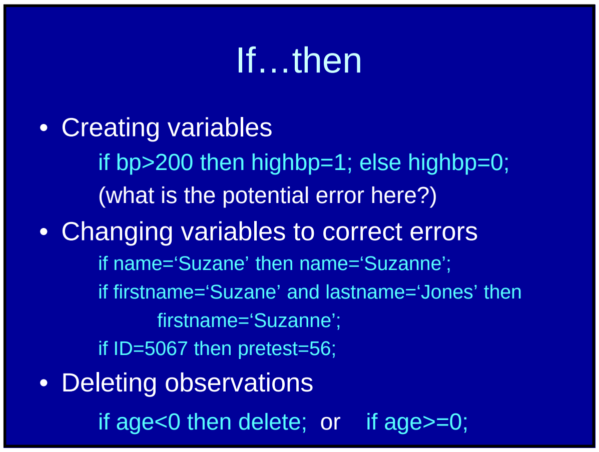


Do/Output Example

To generate data….

```
data mydata;
  do k=1 to 10;
    x=k;
    y=500 + 10*x;
    output;
  end;
run;
```

(what happens without the output statement?)



If…then

- Creating variables

  if bp>200 then highbp=1; else highbp=0;

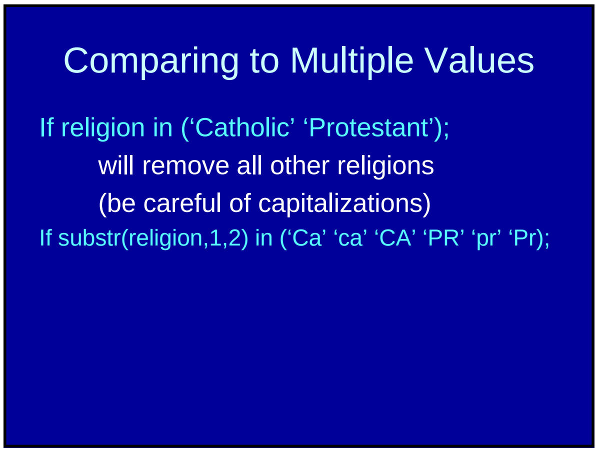
  (what is the potential error here?)
- Changing variables to correct errors

  if name='Suzane' then name='Suzanne';

  if firstname='Suzane' and lastname='Jones' then
  firstname='Suzanne';

  if ID=5067 then pretest=56;
- Deleting observations

  if age<0 then delete; or if age>=0;



Comparing to Multiple Values

If religion in ('Catholic' 'Protestant');

will remove all other religions

(be careful of capitalizations)

If substr(religion,1,2) in ('Ca' 'ca' 'CA' 'PR' 'pr' 'Pr);

If…then…else

- Using else is a more efficient way of processing the data, as it continues from it’s own step.

```
    if age>65 then agecat=‘old;
    if age<=65 then agecat=‘young’;
    vs. if age>65 then agecat=‘old’;
    else if age>. then agecat=‘young’;
```

- In SAS, an else is its own statement

```
  If height > 72 then heightcat=‘tall’;
    else if height < 60 then heightcat=‘short’;
    else heightcat=‘medium’;
```

Using If/Output to Split Datasets

- Splitting men and women from a dataset

```
data males females;
     set alldata;
     if gender=‘M’ then output males;
     if gender=‘F’ then output females;
```

(how can this code be made more efficient?)

Practice Problems

first. and last.

- You can use these to identify the first or last record from sorted data
- When was each Taney Dragon’s first game with one homerun
  - If you have a dataset with the results of each game (with a numeric indicator for each outcome)

```
proc sort data=taney; by name HR gameday;
data taneyfirstHR; set taney; by name HR gameday;
  if first.HR;
Proc print data=taneyfirstHR; where HR=1;
```

Taney Dragons Data

Player	Gameday	HR
Mo’Ne Davis	1	0
Mo’Ne Davis	2	1
Mo’Ne Davis	3	1
Mo’Ne Davis	4	0
Mo’Ne Davis	5	2
Scott Banduras	1	0
Scott Banduras	2	0
Scott Banduras	3	1
Scott Banduras	4	0

Taney Dragons Data - Sorted

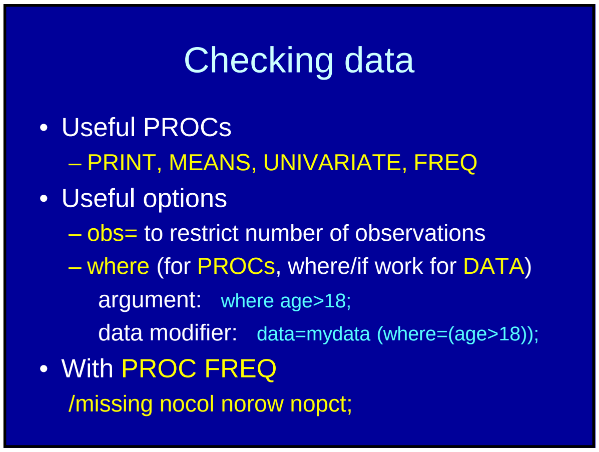
Sorted data

Player	Day	HR	First.HR
Mo’Ne Davis	1	0	1
Mo’Ne Davis	4	0	0
Mo’Ne Davis	2	1	1
Mo’Ne Davis	3	1	0
Mo’Ne Davis	5	2	1
Scott Banduras	1	0	1
Scott Banduras	2	0	0
Scott Banduras	4	0	0
Scott Banduras	3	1	1

If first.HR

Player	Day	HR	First.HR
Mo’Ne Davis	1	0	1
Mo’Ne Davis	2	1	1
Mo’Ne Davis	5	2	1
Scott Banduras	1	0	1
Scott Banduras	3	1	1

where HR=1



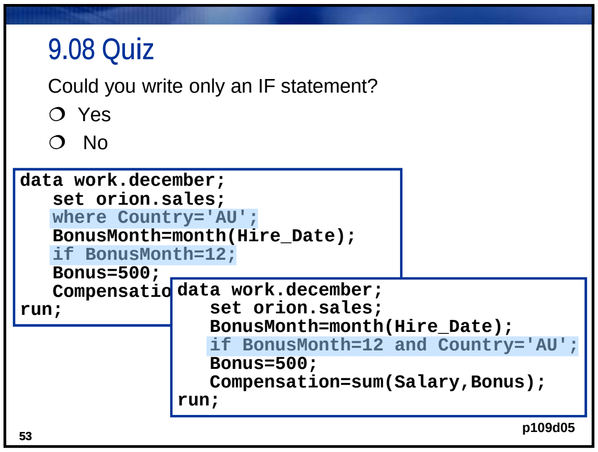
Checking data

- Useful PROCs
  - PRINT, MEANS, UNIVARIATE, FREQ
- Useful options
  - obs= to restrict number of observations
  - where (for PROCs, where/if work for DATA)

    argument: where age>18;

    data modifier: data=mydata (where=(age>18));
- With PROC FREQ

  /missing nocol norow nopct;



9.08 Quiz

Could you write only an IF statement?

❍ Yes

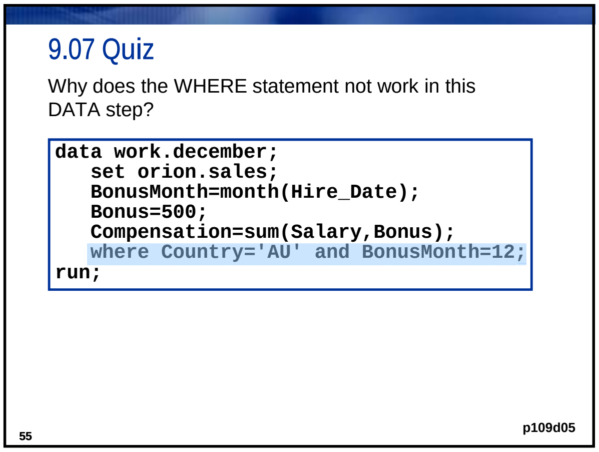
❍ No

```
data work.december;
   set orion.sales;
   where Country='AU';
   BonusMonth=month(Hire_Date);
   if BonusMonth=12;
   Bonus=500;
   Compensatio
run;
```

```
data work.december;
   set orion.sales;
   BonusMonth=month(Hire_Date);
   if BonusMonth=12 and Country='AU';
   Bonus=500;
   Compensation=sum(Salary,Bonus);
run;
```

53

p109d05



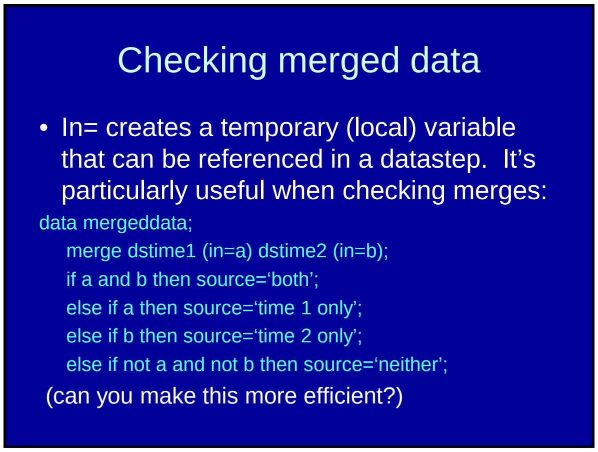
9.07 Quiz

Why does the WHERE statement not work in this DATA step?

```
data work.december;
   set orion.sales;
   BonusMonth=month(Hire_Date);
   Bonus=500;
   Compensation=sum(Salary,Bonus);
   where Country='AU' and BonusMonth=12;
run;
```

55

p109d05



Checking merged data

- In= creates a temporary (local) variable that can be referenced in a datastep. It's particularly useful when checking merges:

data mergeddata;

merge dstime1 (in=a) dstime2 (in=b);

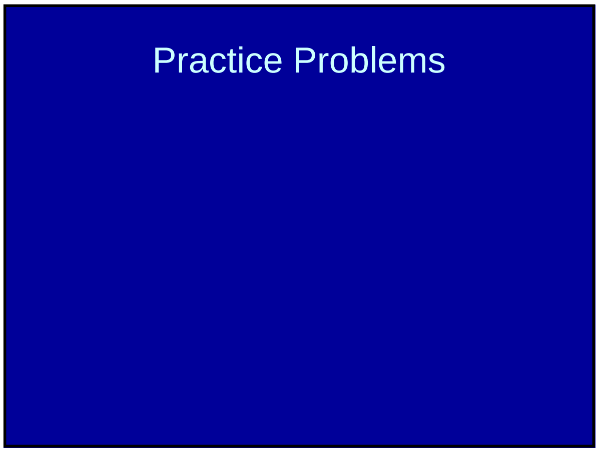
if a and b then source='both';

else if a then source='time 1 only';

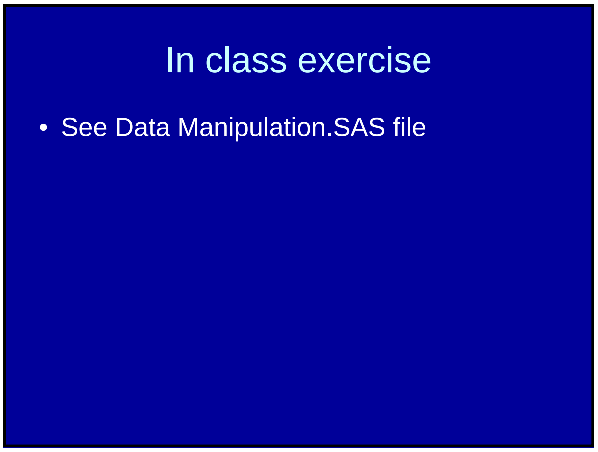
else if b then source='time 2 only';

else if not a and not b then source='neither';

(can you make this more efficient?)



Practice Problems



In class exercise

- See Data Manipulation.SAS file