UCD School of Mathematics and Statistics

STAT40840: Data programming with SAS
Laura Kirwan

Lecture 7

www.ucd.ie/mathstat

Lecture 7 – part 1: Manipulating Data

7.1 Using SAS Functions

7.2 Conditional Processing

Objectives – 7.1

- Use SAS functions to create data values.

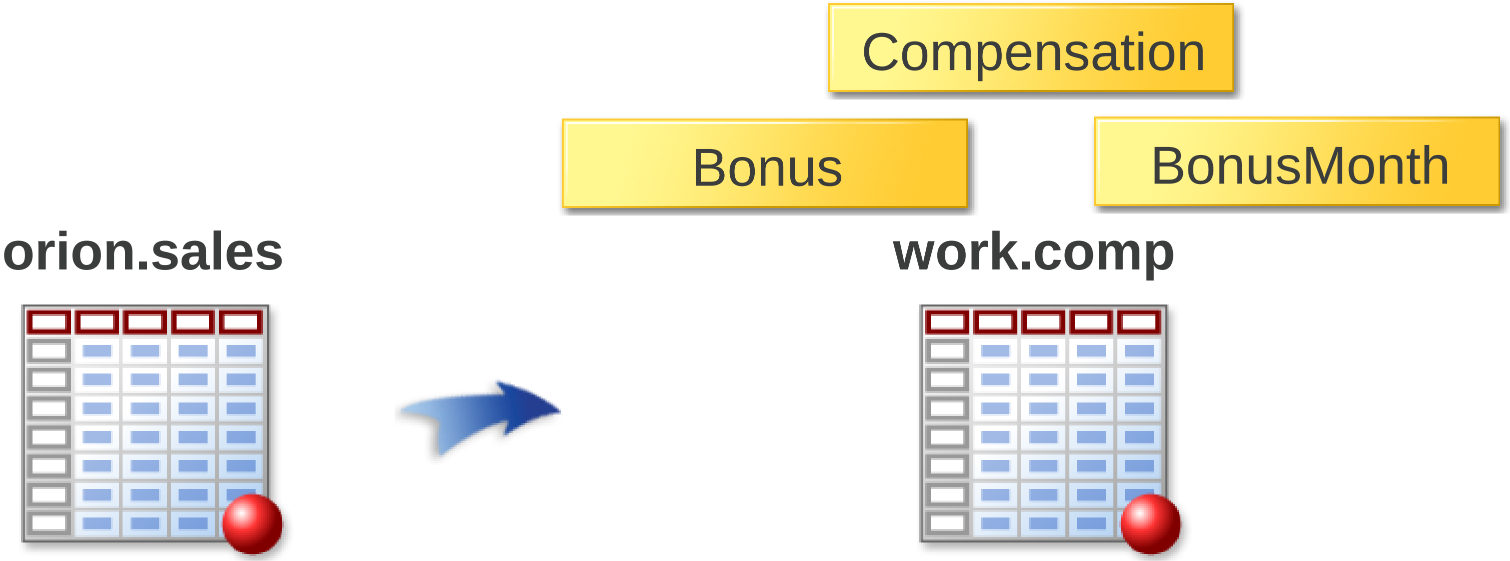
Scenario

Orion Star management plans to give a $500 bonus to each employee in his or her hire month.

Considerations

Create a new data set with three new variables.

- **Bonus**, which is a constant 500
- **Compensation**, which is the sum of **Salary** and **Bonus**
- **BonusMonth**, which is the month in which the employee was hired



Compensation

Bonus

BonusMonth

orion.sales

work.comp

Considerations

Partial **orion.sales**

Employee _ID	First _Name	Last _Name	Gender	Salary	Job_Title	Country	Birth_ Date	Hire_ Date
120102	Tom	Zhou	M	108255	Sales Manager	AU	3510	10744
120103	Wilson	Dawes	M	87975	Sales Manager	AU	-3996	5114
120121	Irenie	Elvish	F	26600	Sales Rep. II	AU	-5630	5114

Partial **work.comp**

Employee _ID	First _Name	Last _Name	Bonus	Compensation	Bonus Month
120102	Tom	Zhou	500	108755	6
120103	Wilson	Dawes	500	88475	1
120121	Irenie	Elvish	500	27100	1

Drop **Gender**, **Salary**, **Job_Title**, **Country**, **Birth_Date**, and **Hire_Date** from **work.comp**.

Exercise 1

Which of the following statements creates a numeric variable, **Bonus**, with a value

a. Bonus=$500;

b. Bonus=500;

c. label Bonus='500';

d. format Bonus 500.;

Partial **work.comp**

Bonus	Compensation	Bonus Month
500	108755	6
500	88475	1
500	27100	1

SAS Functions

SAS functions can be used in an assignment statement. A *function* is a routine that accepts arguments and returns a value.

*variable=function-name***(***argument1***,** *argument2***,** …**);**

Some functions manipulate character values, compute descriptive statistics, or manipulate SAS date values.

- Arguments are enclosed in parentheses and separated by commas.
- A function can return a numeric or character result.

SUM Function

Use the *SUM function* to create **Compensation**. The SUM function is a descriptive statistics function that returns the sum of its arguments.

```
Compensation=sum(Salary,Bonus);
```

SUM(*argument1*,*argument2*, ...**)**

The arguments must be numeric.

Missing values are ignored by SUM and other descriptive statistics functions.

MONTH Function

Use the *MONTH function* to extract the month of hire from **Hire_Date**.

```
BonusMonth=month(Hire_Date);
```

MONTH(*SAS-date***)**

Other date functions can do the following:

- extract information from SAS date values
- create SAS date values

Date Functions: Extracting Values

Syntax	Description
YEAR(*SAS-date*)	Extracts the year from a SAS date and returns a four-digit year.
QTR(*SAS-date*)	Extracts the calendar quarter from a SAS date and returns a number from 1 to 4.
MONTH(*SAS-date*)	Extracts the month from a SAS date and returns a number from 1 to 12.
DAY(*SAS-date*)	Extracts the day of the month from a SAS date and returns a number from 1 to 31.
WEEKDAY(*SAS-date*)	Extracts the day of the week from a SAS date and returns a number from 1 to 7, where 1 represents Sunday.

Date Functions: Creating SAS Dates

Syntax	Description

TODAY()
DATE()
	Returns the current date as a SAS date value.

MDY(*month*,*day*,*year*)	Returns a SAS date value from numeric month, day, and year values.

Examples
CurrentDate=today();
y2k=mdy(01,1,2000);
NewYear=mdy(Mon,Day,2013);

Using SAS Functions

A function call can be used alone in an assignment statement.

```
BonusMonth=month(Hire_Date);
AnnivBonus=mdy(BonusMonth,15,2008);
```

A function call can be part of any SAS expression.

```
if month(Hire_Date)=12;
```

A function call can be an argument to another function.

```
AnnivBonus=mdy(month(Hire_Date),15,2012);
```

Using SAS Functions

Create **Bonus**, **Compensation**, and **BonusMonth**.

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

L7_D1.sas

```
175  data work.comp;
176     set orion.sales;
177     Bonus=500;
178     Compensation=sum(Salary,Bonus);
179     BonusMonth=month(Hire_Date);
180  run;

NOTE: There were 165 observations read from the data set ORION.SALES.
NOTE: The data set WORK.COMP has 165 observations and 12 variables.
```

orion.sales has nine variables.

Viewing the Output

```
proc print data=work.comp noobs;
   var Employee_ID First_Name Last_Name
       Bonus Compensation BonusMonth;
run;
```

```
                First_                                                  Bonus
Employee_ID     Name          Last_Name     Bonus    Compensation       Month

     120102     Tom           Zhou            500          108755           6
     120103     Wilson        Dawes           500           88475           1
     120121     Irenie        Elvish          500           27100           1
     120122     Christina     Ngan            500           27975           7
     120123     Kimiko        Hotstone        500           26690          10
```

L7_D1.sas

Exercise 2

A DROP statement was added to this DATA step. Can the program calculate **Compensation** and **BonusMonth** correctly?

```
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;
```

L7_E2.sas

Lecture 7: Part 1

7.1 Using SAS Functions

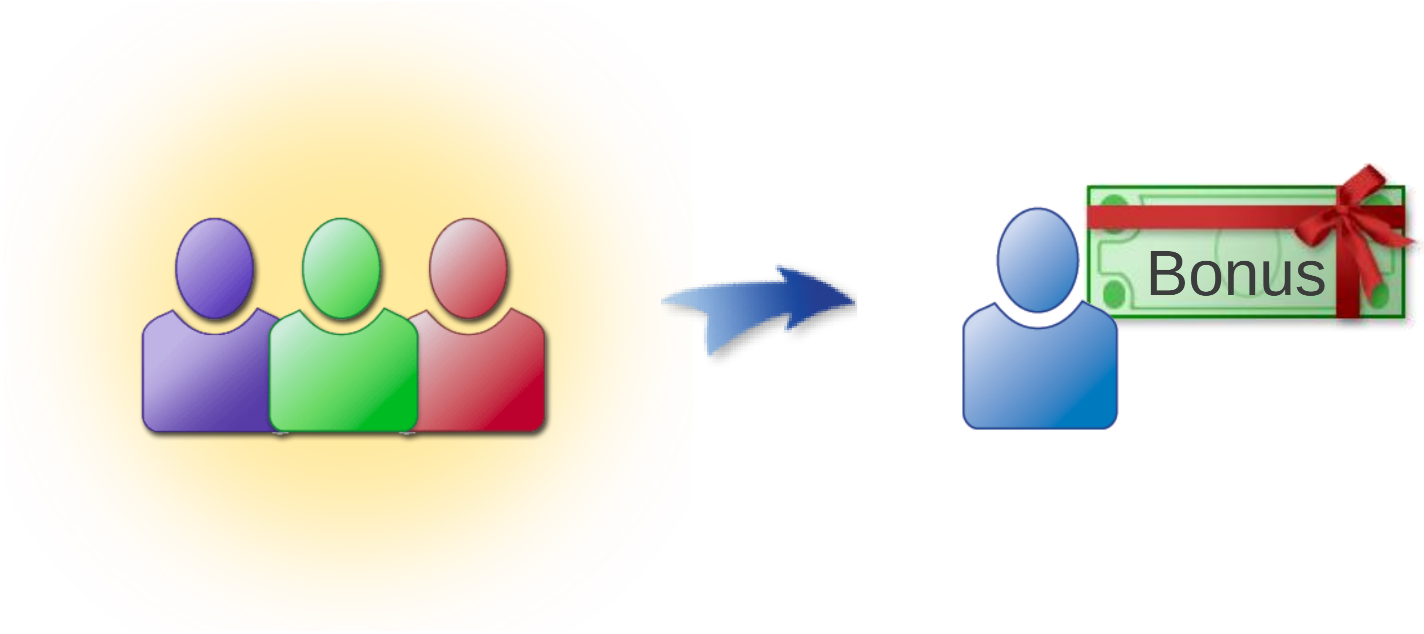
7.2 Conditional Processing

Objectives – Part 2

- Use IF-THEN/ELSE statements to process data conditionally.
- Use DO and END statements to execute multiple statements conditionally.
- Use the LENGTH statement to control the length of character variables.

Scenario

Orion Star management plans to give each sales employee a bonus based on his or her job title.



Considerations

Create a new data set, **work.comp**. Use **orion.sales** as input. Include a new variable, **Bonus**, with a value that is based on **Job_Title**.

Job_Title	Bonus
Sales Rep. IV	1000
Sales Manager	1500
Senior Sales Manager	2000
Chief Sales Officer	2500

IF-THEN Statements

The IF-THEN statement executes a SAS statement for observations that meet a specific condition.

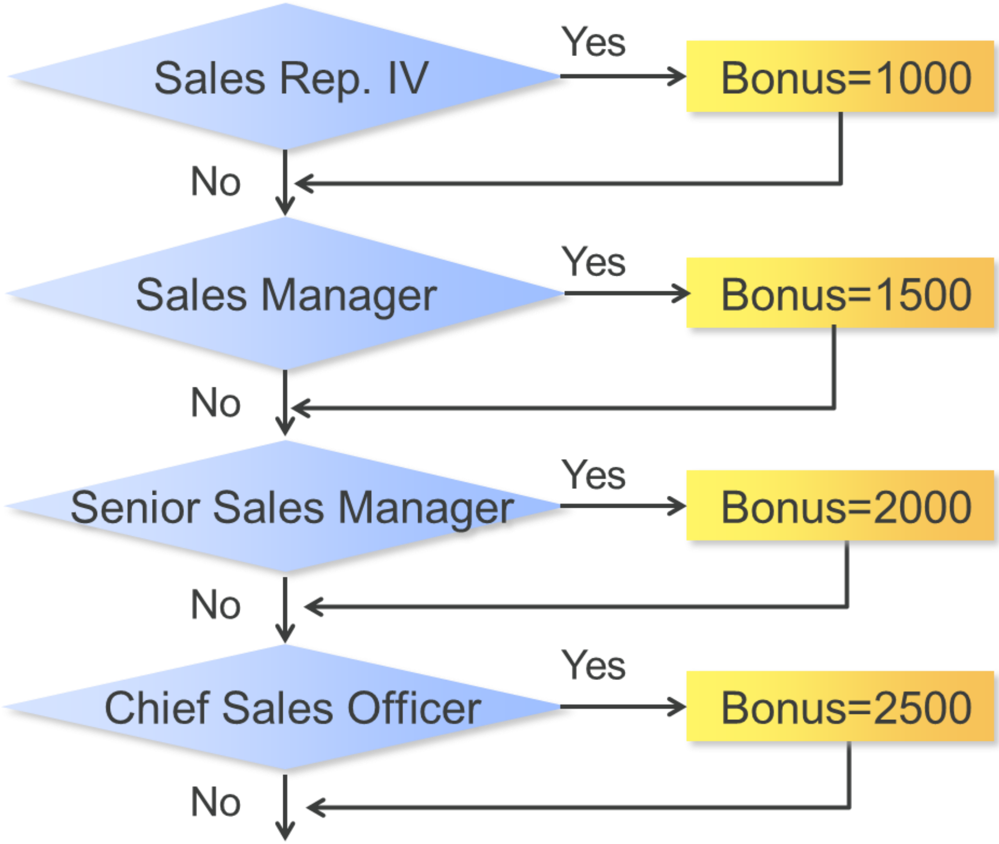
```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   ...
run;
```

IF *expression* **THEN** *statement***;**

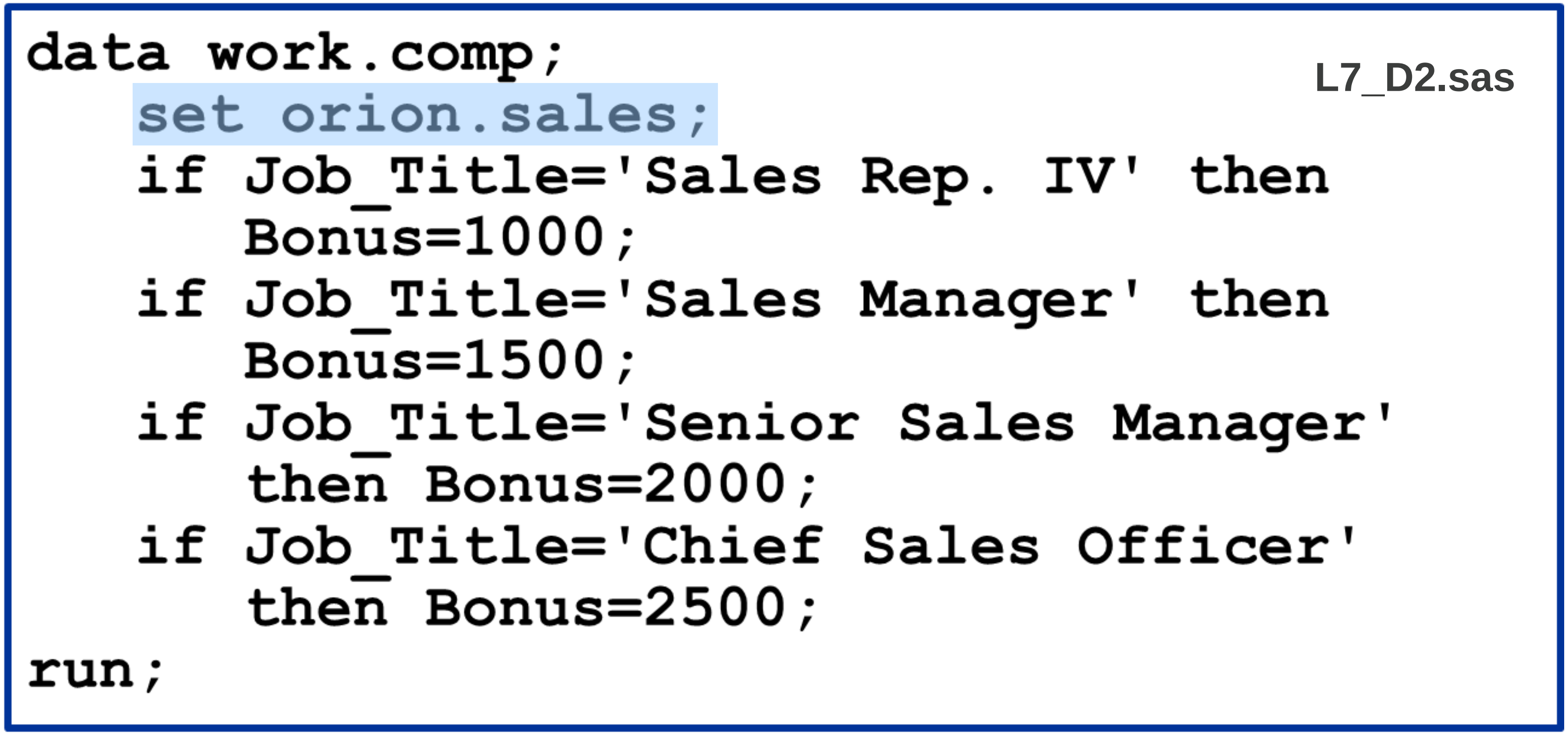
- *expression* defines a condition.
- *statement* can be any executable SAS statement.
- If *expression* is true, then *statement* executes.

Conditional Processing

The value assigned to **Bonus** is determined by testing for various values of **Job_Title**.



Conditional Processing



L7_D2.sas

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   if Job_Title='Sales Manager' then
      Bonus=1500;
   if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

PDV

Employee_ID	Last_Name	...	Job_Title	Bonus
120102	Zhou	...	Sales Manager	.

...

Conditional Processing

false

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   if Job_Title='Sales Manager' then
      Bonus=1500;
   if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

PDV

Employee_ID	Last_Name
120102	Zhou

...

Job_Title	Bonus
Sales Manager	.

...

Conditional Processing

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales R          V' then
      Bonus=1000;
   if Job_Title='Sales Manager' then
      Bonus=1500;
   if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

true

PDV

Employee_ID	Last_Name
120102	Zhou

...

Job_Title	Bonus
Sales Manager	.

Conditional Processing

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   if Job_Title='Sales Manager' then
      Bonus=1500;
   if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

PDV

Employee_ID	Last_Name		Job_Title	Bonus
120102	Zhou	...	Sales Manager	1500

...

Conditional Processing

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   if Job_Title='Sales Manager' then
      Bonus=1500;
   if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

false

PDV

Employee_ID	Last_Name
120102	Zhou

...

Job_Title	Bonus
Sales Manager	1500

Conditional Processing

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   if Job_Title='Sales Manager' then
      Bonus=1500;
   if Job_Title='Senior S     Manager'
      then Bonus=2000;
   if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

false

PDV

Employee_ID	Last_Name
120102	Zhou

...

Job_Title	Bonus
Sales Manager	1500

...

Conditional Processing

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   if Job_Title='Sales Manager' then
      Bonus=1500;
   if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

Employee_ID	Last_Name		Job_Title	Bonus
120102	Zhou	...	Sales Manager	1500

Conditional Processing

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   if Job_Title='Sales Manager' then
      Bonus=1500;
   if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

Continue until EOF

PDV

Employee_ID	Last_Name		Job_Title	Bonus
120102	Zhou	...	Sales Manager	1500

Viewing the Output

```
proc print data=work.comp;
   var Last_Name Job_Title Bonus;
run;
```

```
Obs     Last_Name        Job_Title          Bonus

  1     Zhou             Sales Manager       1500
  2     Dawes            Sales Manager       1500
  3     Elvish           Sales Rep. II          .
  4     Ngan             Sales Rep. II          .
  5     Hotstone         Sales Rep. I           .
  6     Daymond          Sales Rep. I           .
  7     Hofmeister       Sales Rep. IV       1000
  8     Denny            Sales Rep. II          .
  9     Clarkson         Sales Rep. II          .
 10     Kletschkus       Sales Rep. IV       1000
 11     Roebuck          Sales Rep. III         .
 12     Lyon             Sales Rep. I           .
```

L7_D2.sas

Exercise 3

In the program L7_D2.sas, is it possible for more than one condition to be true for a single observation?

a. Yes, more than one condition can be true.

b. No, the conditions are mutually exclusive, so only one condition can be true.

Using the ELSE Statement

Use the *ELSE statement* when you test mutually exclusive conditions.

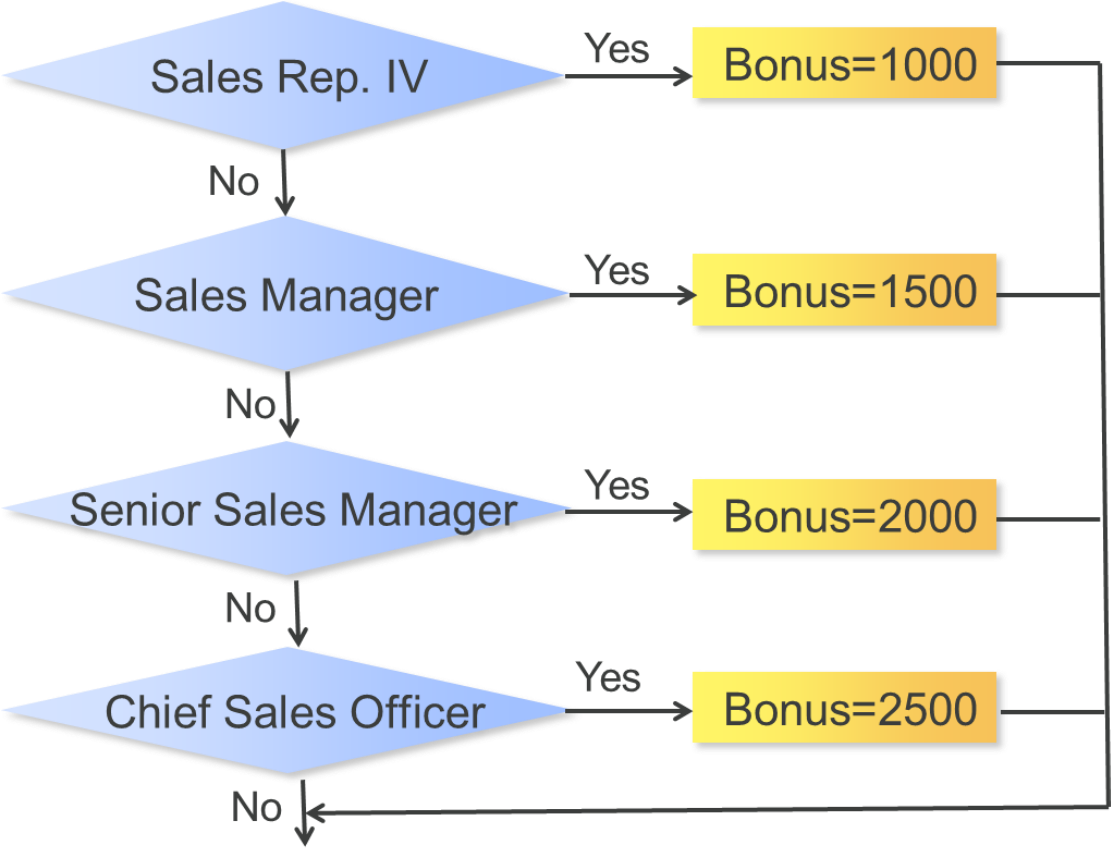
L7_D3.sas

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV`
      then Bonus=1000;
   else if Job_Title='Sales Manager'
      then Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

IF *expression* **THEN** *statement*;
 <ELSE IF *expression* **THEN** *statement*;**>**
 <ELSE IF *expression* **THEN** *statement*;**>**

Conditional Processing

When an expression is true, the associated statement is executed and subsequent ELSE statements are skipped.



IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
      Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

L7_D3.sas

PDV

Employee_ID	Last_Name
120102	Zhou

. . .

Job_Title	Bonus
Sales Manager	.

IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
      Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

false

PDV

Employee_ID	Last_Name
120102	Zhou

...

Job_Title	Bonus
Sales Manager	.

...

IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
      Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

true

PDV

Employee_ID	Last_Name
120102	Zhou

...

Job_Title	Bonus
Sales Manager	.

...

IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
      Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

PDV

Employee_ID	Last_Name
120102	Zhou

...

Job_Title	Bonus
Sales Manager	1500

IF-THEN Statements

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
      Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

Implicit OUTPUT;
Implicit RETURN;

PDV

Employee_ID	Last_Name		Job_Title	Bonus
120102	Zhou	...	Sales Manager	1500

IF-THEN Statements

```
data work.comp;
                 es;
   if Job_Title='Sales Rep. IV' then
      Bonus=1000;
   else if Job_Title='Sales Manager' then
      Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
run;
```

Continue until EOF

PDV

Employee_ID	Last_Name
120102	Zhou

...

Job_Title	Bonus
Sales Manager	1500

Scenario: Part 2

Orion Star management wants to modify the bonus plan as defined below.

Job_Title	Bonus
Sales Rep. III	1000
Sales Rep. IV	1000
Sales Manager	1500
Senior Sales Manager	2000
Chief Sales Officer	2500
All other titles	500

Using Conditional Processing

```
data work.comp;
   set orion.sales;
   if Job_Title='Sales Rep. III' or
      Job_Title='Sales Rep. IV' then
         Bonus=1000;
   else if Job_Title='Sales Manager' then
      Bonus=1500;
   else if Job_Title='Senior Sales Manager'
      then Bonus=2000;
   else if Job_Title='Chief Sales Officer'
      then Bonus=2500;
   else Bonus=500;
run;
```

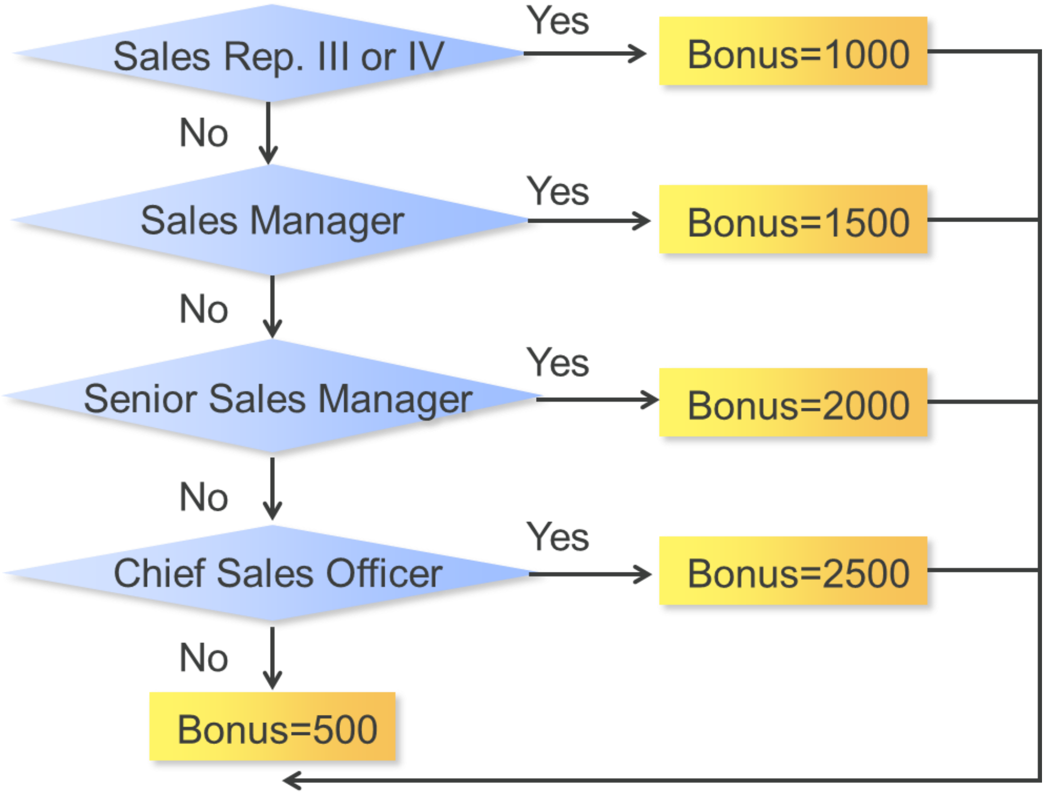
compound condition

IF *expression* **THEN** *statement*;
<**ELSE IF** *expression* **THEN** *statement*;>
<…>
<**ELSE** *statement*;>

L7_D4.sas

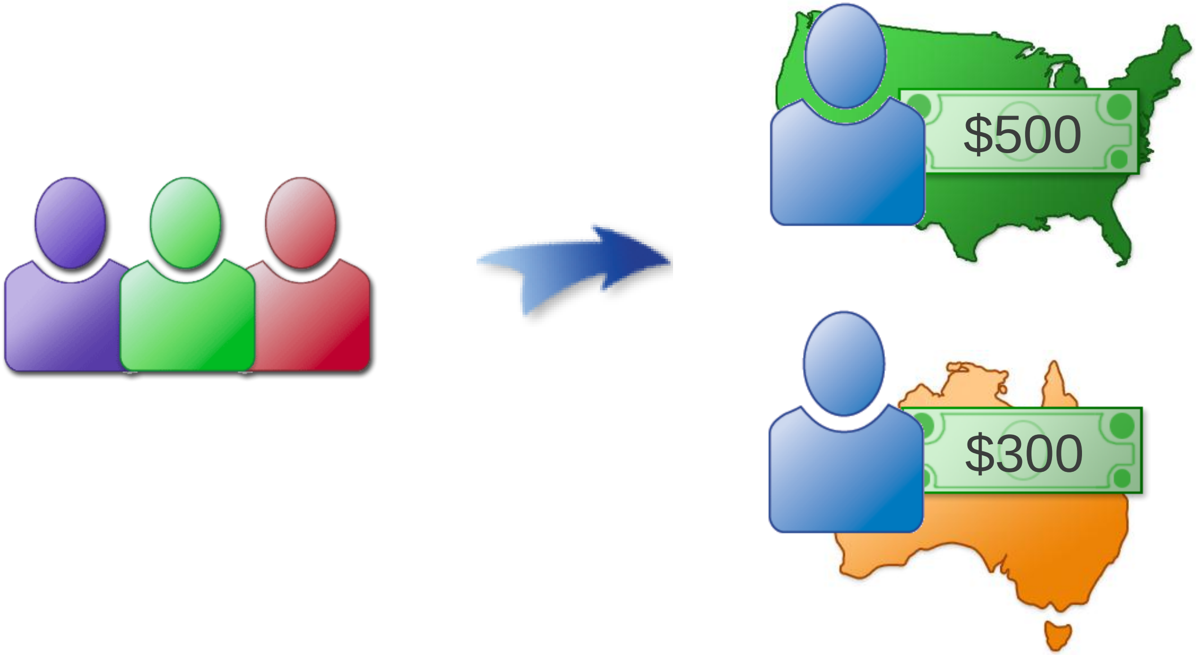
Conditional Processing

An optional final ELSE statement gives an alternative action if none of the conditions are true.



Scenario

Orion Star managers are considering a country-based bonus. Create a new SAS data set named **work.bonus**. Use **orion.sales** as input. The value of the new variable, **Bonus**, is based on **Country**.



IF-THEN/ELSE Statements

If **orion.sales** is validated and includes ***only*** the **Country** values *US* and *AU*, the conditional clause can be omitted from the ELSE statement.

```
data work.bonus;
   set orion.sales;
   if Country='US' then Bonus=500;
   else Bonus=300;
run;
```

IF *expression* **THEN** *statement***;**
ELSE *statement***;**

⚠ All observations not equal to *US* are assigned a bonus of 300.

L7_D5.sas

Exercise 4

Program L7_E4.sas reads **orion.nonsales**, a non-validated data set. Open and submit the program and review the results. Why is **Bonus** set to 300 in observations 125, 197, and 200?

```
data work.bonus;
   set orion.nonsales;
   if Country='US' then Bonus=500;
   else Bonus=300;
run;
```

Cleaning Invalid Data

You can clean the data before checking the value.

```
data work.bonus;
   set orion.nonsales;
   Country=upcase(Country);
   if Country='US'
      then Bonus=500;
   else Bonus=300;
run;
```

L7_D6.sas

It is a best practice to clean the data at the source, but in some cases, that is not possible. With this method, you are creating a clean data set.

Scenario

Orion Star employees receive a bonus once or twice a year. In addition to **Bonus**, add a new variable, **Freq**, that is equal to the following:

- *Once a Year* for United States employees
- *Twice a Year* for Australian employees

Once a Year

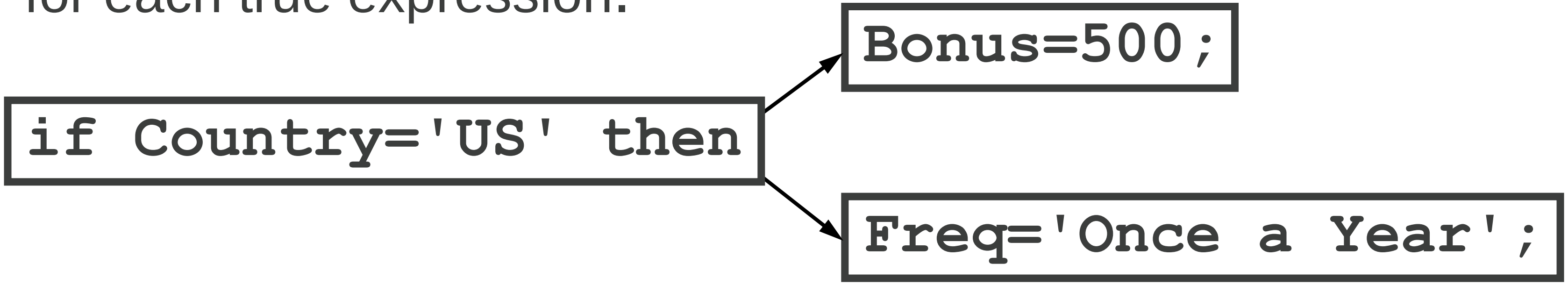
Twice a Year

IF-THEN/ELSE Statements

Only ***one*** executable statement is allowed in IF-THEN and ELSE statements.

IF *expression* **THEN** *statement***;**
ELSE IF *expression* **THEN** *statement***;**
ELSE *statement***;**

For this business scenario, ***two*** statements must be executed for each true expression.



```
if Country='US' then
Bonus=500;
Freq='Once a Year';
```

DO Group

Multiple statements are permitted in a *DO group*.

```
data work.bonus;
   set orion.sales;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

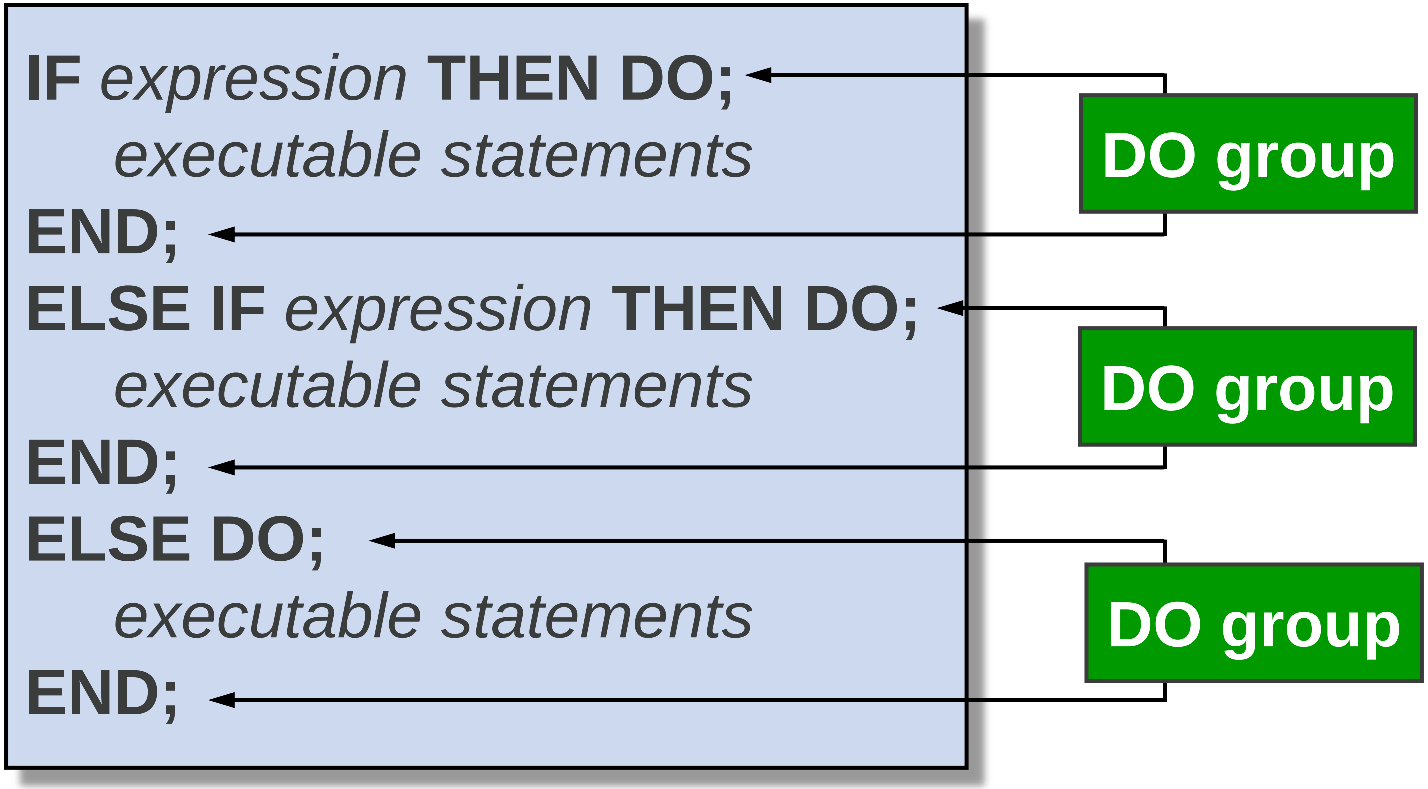
DO group

Each DO group ends with an END statement.

L7_D7.sas

IF-THEN DO/ELSE DO Statements

Multiple statements are also permitted in an ELSE DO group.



```
IF expression THEN DO;
    executable statements
END;
ELSE IF expression THEN DO;
    executable statements
END;
ELSE DO;
    executable statements
END;
```

Compilation

```
data work.bonus;
   set orion.sales;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8

L7_D7.sas

...

Compilation

```
data work.bonus;
   set orion.sales;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

PDV

Employee_ID	First_Name	...	Hire_Date	Bonus
N 8	$ 12		N 8	N 8

...

Compilation

```
data work.bonus;
   set orion.sales;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Co          ' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

11 characters

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Bonus N 8	Freq $ 11

Compilation

```
data work.bonus;
   set orion.sales;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

12 characters

Length does not change.

PDV

Employee_ID	First_Name	...	Hire_Date	Bonus	Freq
N 8	$ 12		N 8	N 8	$ 11

Exercise 5

How would you prevent **Freq** from being truncated?

Defining Character Variables

Set the length of the variable **Freq** to avoid truncation.

L7_D8.sas

```
data work.bonus;
   set orion.sales;
   length Freq $ 12;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

LENGTH *variable(s)* <$> *length*;

It is a good practice to use a LENGTH statement anytime that you create a new character variable.

Compilation

```
data work.bonus;
   set orion.sales;
   length Freq $ 12;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8

L7_D8.sas

...

Compilation

```
data work.bonus;
   set orion.sales;
   length Freq $ 12;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Freq $ 12

Compilation

```
data work.bonus;
   set orion.sales;
   length Freq $ 12;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

PDV

Employee_ID	First_Name	...	Hire_Date	Freq	Bonus
N 8	$ 12		N 8	$ 12	N 8

...

Compilation

```
data work.bonus;
   set orion.sales;
   length Freq $ 12;
   if Country='US' then do;
      Bonus=500;
      Freq='Once a Year';
   end;
   else if Country='AU' then do;
      Bonus=300;
      Freq='Twice a Year';
   end;
run;
```

Length does not change.

PDV

Employee_ID N 8	First_Name $ 12	...	Hire_Date N 8	Freq $ 12	Bonus N 8