

# Logic and Control

## Statistics 426: SAS Programming

### Module 4

2021

## Logic and Control

SAS has many ways to use variables (also called vectors), and comes with tools for extracting and manipulating (the good kind) information in variables.

## Conditional Statements

SAS will execute a statement or statements conditionally using many different statements, including WHERE, IF, IF/ELSE. The focus at the moment is the WHERE statement within the DATA step

```
data libref.datasetname;  
set libref.inputdata;  
WHERE where-expression;  
:  
run;
```

## where-expressions

```
WHERE Gender='M';
```

or

```
WHERE salary>50000;
```

are both where-expressions; a (logical) comparison condition is to be met to execute the commands

## Logical Comparison Operators I

Arithmetic operators

| Symbol | Definition     |
|--------|----------------|
| **     | Exponentiation |
| *      | Multiplication |
| /      | Division       |
| +      | Addition       |
| -      | Subtraction    |

```
WHERE salary/12 < 6000;
```

```
WHERE Salary/12*1.1 >=7500;
```

## Logical Comparison Operators II

Logical operators; the most commonly known Boolean operations are “and”, “or”, and “not”

| Symbol | Mnemonic | Definition                 |
|--------|----------|----------------------------|
| &      | AND      | Logical and (intersection) |
|        | OR       | Logical or                 |
| ~      | NOT      | Logical not                |

```
WHERE Gender ne 'M' & Salary >=50000;  
WHERE Country='AU' or Country='US';
```

## Logical Comparison Operators III

Comparison operators

| Symbol   | Mnemonic | Definition               |
|----------|----------|--------------------------|
| =        | EQ       | Equal to                 |
| ^= or ~= | NE       | Not equal to             |
| >        | GT       | Greater than             |
| <        | LT       | Less than                |
| >=       | GE       | Greater than or equal to |
| <=       | LE       | Less than or equal to    |
|          | IN       | Equal to one of a list   |

```
WHERE country in ('AU' 'US');
```

## Logical Comparison Operators IV

Special WHERE operators

| Symbol | Mnemonic    | Definition                |
|--------|-------------|---------------------------|
|        | BETWEEN-AND | Inclusive range of values |
|        | IS NULL     | Missing values            |
|        | IS MISSING  | Missing value             |
| ?      | CONTAINS    | A character string        |
|        | LIKE        | A character pattern       |

```
WHERE Salary between 50000 and 100000;  
(is equivalent to)  
WHERE 50000<=Salary<=100000;
```

## More on special WHERE operators

IS NULL and IS MISSING: select observations in which the value of a variable is missing; they can be used for either character or numeric variables

```
WHERE employee_id is null;  
WHERE employee_id is missing;
```

CONTAINS operator: selects observations that include the specified substring  
- the position of the substring within the variable's values is not important  
- the operator **is** case sensitive with comparisons

```
WHERE job_title contains 'Rep';
```

## LIKE operator

LIKE operator: selects observations by comparing character values to specified patterns; two special characters are available for specifying a pattern (1) Percent sign (%) replaces *any* number of characters (2) Underscore (\_) replaces *one* character - Consecutive underscores can be specified, a % and \_ can be used in the same pattern. The operator *is* case sensitive

## LIKE operator examples

Select observations that begin with any number of characters and end with an n  
WHERE name like '%n';

Selects observations that begin with a T, followed by a single character, followed by an M, followed by any number of characters

```
WHERE name like 'T_M%';
```

## Example Hercules where

```
libname hercules 's:\courses\stat-renaes\stat404';
```

```
proc print data=hercules.nonsales;  
run;
```

## Herc print

| Obs | Employee_ID | First    | Last       | Gender | Salary | Job_Title               | Country | Birth_Date | Hire_Date  |
|-----|-------------|----------|------------|--------|--------|-------------------------|---------|------------|------------|
| 1   | 120101      | Patrick  | Lu         | M      | 163040 | Director                | AU      | 18/08/1976 | 01/07/2003 |
| 2   | 120104      | Kareen   | Billington | F      | 46230  | Administration Manager  | au      | 11/05/1954 | 01/01/1981 |
| 3   | 120105      | Liz      | Povey      | F      | 27110  | Secretary I             | AU      | 21/12/1974 | 01/05/1999 |
| 4   | 120106      | John     | Hornsey    | M      | .      | Office Assistant II     | AU      | 23/12/1944 | 01/01/1974 |
| 5   | 120107      | Sherie   | Sheedy     | F      | 30475  | Office Assistant III    | AU      | 01/02/1978 | 21/01/1953 |
| 6   | 120108      | Gladys   | Gromek     | F      | 27660  | Warehouse Assistant II  | AU      | 23/02/1984 | 01/08/2006 |
| 7   | 120108      | Gabriele | Baker      | F      | 26495  | Warehouse Assistant I   | AU      | 15/12/1986 | 01/10/2006 |
| 8   | 120110      | Dennis   | Entwisle   | M      | 28615  | Warehouse Assistant III | AU      | 20/11/1949 | 01/11/1979 |
| 9   | 120111      | Ubaldo   | Spillane   | M      | 26895  | Security Guard II       | AU      | 23/07/1949 | .          |
| 10  | 120112      | Ellis    | Glattback  | F      | 26550  |                         | AU      | 17/02/1969 | 01/07/1990 |

## Herc print log

```
Log - (Untitled)
14 libname hercules 's:\courses\stat-renaes\stat404';
NOTE: Libref HERCULES was successfully assigned as follows:
      Engine:          V9
      Physical Name:  s:\courses\stat-renaes\stat404
15
16 proc print data=hercules.nonsales;
NOTE: Writing HTML Body file: sashtml1.htm
17 run;

NOTE: There were 235 observations read from the data set HERCULES.NONSALES.
NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.37 seconds
      cpu time           0.24 seconds
```

## Herc where proc print

```
proc print data=hercules.nonsales;
  var Employee_ID Gender Salary Job_Title
      Country Birth_Date Hire_Date;
  where Employee_ID = . or
      Gender not in ('F','M') or
      Salary not between 24000 and 500000 or
      Job_Title = ' ' or
      Country not in ('AU','US') or
      Birth_Date > Hire_Date or
      Hire_Date < '01JAN1974'd;
run;
```

Herc where proc print

### The SAS System

| Obs | Employee_ID | Gender | Salary | Job_Title                 | Country | Birth_Date | Hire_Date  |
|-----|-------------|--------|--------|---------------------------|---------|------------|------------|
| 2   | 120104      | F      | 46230  | Administration Manager    | au      | 11/05/1954 | 01/01/1981 |
| 4   | 120106      | M      | .      | Office Assistant II       | AU      | 23/12/1944 | 01/01/1974 |
| 5   | 120107      | F      | 30475  | Office Assistant III      | AU      | 01/02/1978 | 21/01/1953 |
| 9   | 120111      | M      | 26895  | Security Guard II         | AU      | 23/07/1949 | .          |
| 10  | 120112      | F      | 26550  |                           | AU      | 17/02/1969 | 01/07/1990 |
| 12  | 120114      | G      | 31285  | Security Manager          | AU      | 08/02/1944 | 01/01/1974 |
| 13  | 120115      | M      | 2650   | Service Assistant I       | AU      | 08/05/1984 | 01/08/2005 |
| 14  | .           | M      | 29250  | Service Assistant II      | AU      | 13/06/1959 | 01/02/1980 |
| 20  | 120191      | F      | 2401   | Trainee                   | AU      | 17/01/1959 | 01/01/2003 |
| 84  | 120695      | M      | 28180  | Warehouse Assistant II    | au      | 13/07/1964 | 01/07/1989 |
| 87  | 120698      | M      | 26160  | Warehouse Assistant I     | au      | 17/05/1954 | 01/08/1976 |
| 101 | 120723      |        | 33950  | Corp. Comm. Specialist II | US      | 10/08/1949 | 01/01/1974 |
| 125 | 120747      | F      | 43590  | Financial Controller I    | us      | 20/06/1974 | 01/08/1995 |
| 197 | 120994      | F      | 31645  | Office Administrator I    | us      | 16/06/1974 | 01/11/1994 |
| 200 | 120997      | F      | 27420  | Shipping Administrator I  | us      | 21/11/1974 | 01/09/1996 |
| 214 | 121011      | M      | 25735  | Service Assistant I       | US      | 11/03/1944 | 01/01/1968 |

Herc where log

```
Log - (Untitled)
18 proc print data=hercules.nonsales;
19   var Employee_ID Gender Salary Job_Title
20     Country Birth_Date Hire_Date;
21   where Employee_ID = . or
22     Gender not in ('F','M') or
23     Salary not between 24000 and 500000 or
24     Job_Title = ' ' or
25     Country not in ('AU','US') or
26     Birth_Date > Hire_Date or
27     Hire_Date < '01JAN1974'd;
28 run;

NOTE: There were 16 observations read from the data set HERCULES.NONSALES.
WHERE (Employee_ID=.) or Gender not in ('F', 'M') or not (Salary>=24000 and
Salary<=500000) or (Job_Title=' ') or Country not in ('AU', 'US') or
(Birth_Date>Hire_Date) or (Hire_Date<'01JAN1974'D);
NOTE: PROCEDURE PRINT used (Total process time):
real time          0.06 seconds
cpu time           0.03 seconds
```

## Herc where alt

A DATA step can also be used to create a new dataset that meets the conditions set in the WHERE statement of PROC PRINT (the first methods requires two less lines of code with the same results (although you may not want to create a dataset in some situations)

## Herc where alt

```
data work.emps;
  set hercules.nonsales;
  keep Employee_ID Gender Salary Job_Title
      Country Birth_Date Hire_Date;
  where Employee_ID = . or
      Gender not in ('F','M') or
      Salary not between 24000 and 500000 or
      Job_Title = ' ' or
      Country not in ('AU','US') or
      Birth_Date > Hire_Date or
      Hire_Date < '01JAN1974'd;
run;
```

## Herc where alt proc print

```
proc print data=emps;
run;
```

Here where alt

### The SAS System

| Obs | Employee_ID | Gender | Salary | Job_Title                 | Country | Birth_Date | Hire_Date  |
|-----|-------------|--------|--------|---------------------------|---------|------------|------------|
| 1   | 120104      | F      | 46230  | Administration Manager    | au      | 11/05/1954 | 01/01/1981 |
| 2   | 120106      | M      | .      | Office Assistant II       | AU      | 23/12/1944 | 01/01/1974 |
| 3   | 120107      | F      | 30475  | Office Assistant III      | AU      | 01/02/1978 | 21/01/1953 |
| 4   | 120111      | M      | 26895  | Security Guard II         | AU      | 23/07/1949 | .          |
| 5   | 120112      | F      | 26550  |                           | AU      | 17/02/1969 | 01/07/1990 |
| 6   | 120114      | G      | 31285  | Security Manager          | AU      | 08/02/1944 | 01/01/1974 |
| 7   | 120115      | M      | 2650   | Service Assistant I       | AU      | 08/05/1984 | 01/08/2005 |
| 8   | .           | M      | 29250  | Service Assistant II      | AU      | 13/06/1959 | 01/02/1980 |
| 9   | 120191      | F      | 2401   | Trainee                   | AU      | 17/01/1959 | 01/01/2003 |
| 10  | 120695      | M      | 28180  | Warehouse Assistant II    | au      | 13/07/1964 | 01/07/1989 |
| 11  | 120698      | M      | 26160  | Warehouse Assistant I     | au      | 17/05/1954 | 01/08/1976 |
| 12  | 120723      |        | 33950  | Corp. Comm. Specialist II | US      | 10/08/1949 | 01/01/1974 |
| 13  | 120747      | F      | 43590  | Financial Controller I    | us      | 20/06/1974 | 01/08/1995 |
| 14  | 120994      | F      | 31645  | Office Administrator I    | us      | 16/06/1974 | 01/11/1994 |
| 15  | 120997      | F      | 27420  | Shipping Administrator I  | us      | 21/11/1974 | 01/09/1996 |
| 16  | 121011      | M      | 25735  | Service Assistant I       | US      | 11/03/1944 | 01/01/1968 |

Herc where alt log

```
Log - (Untitled)
29 data work.emps;
30 set hercules.nonsales;
31 keep Employee_ID Gender Salary Job_Title
32 Country Birth_Date Hire_Date;
33 where Employee_ID = . or
34 Gender not in ('F','M') or
35 Salary not between 24000 and 500000 or
36 Job_Title = ' ' or
37 Country not in ('AU','US') or
38 Birth_Date > Hire_Date or
39 Hire_Date < '01JAN1974'd;
40 run;

NOTE: There were 16 observations read from the data set HERCULES.NONSALES.
WHERE (Employee_ID=.) or Gender not in ('F', 'M') or not (Salary>=24000 and
Salary<=500000) or (Job_Title=' ') or Country not in ('AU', 'US') or
(Birth_Date>Hire_Date) or (Hire_Date<'01JAN1974'D);
NOTE: The data set WORK.EMPS has 16 observations and 7 variables.
NOTE: DATA statement used (Total process time):
real time 0.06 seconds
cpu time 0.00 seconds

41
42 proc print data=emps;
43 run;

NOTE: There were 16 observations read from the data set WORK.EMPS.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.03 seconds
cpu time 0.01 seconds
```

## Conditioning with IF

IF statement selects observations for reports by subsetting. The IF statement continues processing only those that meet a specified condition. Without a THEN to go with it, it only subsets by the IF statement.

IF (by itself) *cannot* use the special WHERE operators such as BETWEEN-AND, IS NULL, IS MISSING, CONTAINS and LIKE; all other WHERE operators and operands are used

## General form of IF

General form:

```
IF expression;
```

where **expression** is any valid SAS expression. If the expression is true, the DATA step continues to process that record or observation. If the expression is false, no further statements are processed for that record or observation, and control returns to the top of the DATA step.

```
IF bonus=500;
```

## IF family

IF-THEN statements: execute a SAS statement for observations that meet specific conditions.

General form:

```
IF expression THEN statement;
```



**expression:** a sequence of operands and operators that form a set of instructions

**statement:** any executable statement

```
IF employee_id = 123456 THEN salary=36000;
```

## IF family

If the condition in the IF clause is met, the IF-THEN statement executes a statement for that observation. When an IF expression is true in an IF-THEN statement series, there is no reason to check the remaining IF-THEN statements.

The word ELSE can be placed before the IF, causing SAS to execute conditional statements until it finds the first true statement

## IF-THEN/ELSE

IF-THEN/ELSE:

If the conditions are met in the IF clause, the IF-THEN statement executes a SAS statement. An option ELSE statement gives an alternative action if the THEN clause is not executed. Without the ELSE clause, SAS will evaluate all IF-THEN statements until it encounters the first true statement

```
IF bonus ne 500 THEN DELETE;
```

```
IF country = 'US' THEN DELETE;
```

## WHERE vs IF (family)

| Step and usage               | WHERE | IF  |
|------------------------------|-------|-----|
| PROC step                    | Yes   | No  |
| DATA step                    |       |     |
| INPUT                        | No    | Yes |
| Assignment                   | No    | Yes |
| SET                          | Yes   | Yes |
| SET/MERGE                    |       |     |
| variable in all datasets     | Yes   | Yes |
| variable not in all datasets | No    | Yes |

## Example Hercules IF family

```
data bonus;
  set hercules.sales;
  if country='US' then Bonus=500;
  else if country='AU' then Bonus=300;
run;

proc print data=bonus;
  var First_Name Last_Name Country Bonus;
run;
```

Example Hercules IF family proc print

### The SAS System

| Obs | First_Name | Last_Name  | Country | Bonus |
|-----|------------|------------|---------|-------|
| 1   | Tom        | Zhou       | AU      | 300   |
| 2   | Wilson     | Dawes      | AU      | 300   |
| 3   | Irenie     | Elvish     | AU      | 300   |
| 4   | Christina  | Ngan       | AU      | 300   |
| 5   | Kimiko     | Hotstone   | AU      | 300   |
| 6   | Lucian     | Daymond    | AU      | 300   |
| 7   | Fong       | Hofmeister | AU      | 300   |
| 8   | Satyakam   | Denny      | AU      | 300   |
| 9   | Sharryn    | Clarkson   | AU      | 300   |
| 10  | Monica     | Kletschkus | AU      | 300   |

## Example Hercules IF family log

```
Log - (Untitled)
3  data bonus;
54  set hercules.sales;
55  if country='US' then Bonus=500;
56  else Bonus=300;
57  run;

NOTE: There were 165 observations read from the data set HERCULES.SALES.
NOTE: The data set WORK.BONUS has 165 observations and 10 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.00 seconds

58
59  proc print data=bonus;
60  var First_Name Last_Name Country Bonus;
61  run;

NOTE: There were 165 observations read from the data set WORK.BONUS.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds
```

## Example Hercules IF family alt

```
data bonus;
  set hercules.sales;
  if country='US' then Bonus=500;
  else Bonus=300;
run;

proc print data=bonus;
  var First_Name Last_Name Country Bonus;
run;
```

Example Hercules IF family alt proc print

### The SAS System

| Obs | First_Name | Last_Name  | Country | Bonus |
|-----|------------|------------|---------|-------|
| 1   | Tom        | Zhou       | AU      | 300   |
| 2   | Wilson     | Dawes      | AU      | 300   |
| 3   | Irenie     | Elvish     | AU      | 300   |
| 4   | Christina  | Ngan       | AU      | 300   |
| 5   | Kimiko     | Hotstone   | AU      | 300   |
| 6   | Lucian     | Daymond    | AU      | 300   |
| 7   | Fong       | Hofmeister | AU      | 300   |
| 8   | Satyakam   | Denny      | AU      | 300   |
| 9   | Sharryn    | Clarkson   | AU      | 300   |
| 10  | Monica     | Kletschkus | AU      | 300   |

## Example Hercules IF family alt log

```
Log - (Untitled)
62  data bonus;
63      set hercules.sales;
64      if country='US' then Bonus=500;
65      else Bonus=300;
66  run;

NOTE: There were 165 observations read from the data set HERCULES.SALES.
NOTE: The data set WORK.BONUS has 165 observations and 10 variables.
NOTE: DATA statement used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds

67
68  proc print data=bonus;
69      var First_Name Last_Name Country Bonus;
70  run;

NOTE: There were 165 observations read from the data set WORK.BONUS.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds
```