**Chapter XIV – Do Loops**

**DO** loops can execute any number of times in a single iteration of the DATA step, and it makes your **DATA** step easier to debug

1. Constructing **DO** Loops

It can greatly reduce the number of statements required for a repetitive calculation by processing a group of statements repeatedly rather than once

* Basic code:

**DO** *index-variable***=***start* **TO** *stop* **BY** *increment***;**

*SAS statements*

**END;**

* + - The *start*, *stop*, and *increment* values set upon entry into the **DO** loop not be changed during the processing of the **DO** loop be numbers, variables, or SAS expressions.
    - The **END** statement terminates the loop.
    - Index variable must be specified, which stores the value of the current iteration of the **DO** loop
    - Eg:

**DATA** finance.earnings;

Amount=1000;

Rate=.075/12;

**DO** month=1 **TO** 12;

Earned+(amount+earned)\*rate;

**END**;

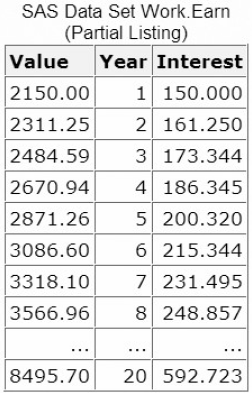
**RUN**;

注意：**DO** loop连续运行12次，然后输出结果，并非输出每次计算的结果。当month累积到第13次时，因为13超过了*stop* specify的12，所以程序自动到下一步**END**。

* Explicit **OUTPUT** Statements

因为**DO** loop是连续运行，并且只显示在运行结束后得出的结果，所以如果想得到每次运算的答案分别是什么，我们需要使用**OUTPUT** statement。此间，**OUTPUT** statement会自动overwrite **DATA** step的输出，因为只有当**OUTPUT**被运行时，**DATA** step会输出一个结果

Eg:

 **DATA** work.earn;

Value=2000;

**DO** Year=1 **TO** 20;

Interest=value\*.075;

value+interest;

**OUTPUT**;

**END**;

**RUN**;

* Decrementing **DO** Loops

Specifying a negative value for the **BY** clause. In decrementing **Do** loop, the start value must always be greater than the stop value in order to decrease the index variable during each iteration

Eg:

**DO** index-variable=5 **TO** 1 by -1;

SAS statements

**END**;

* Specifying a Series of Items

Specify how many times a **DO** loop executes by listing items in a series

* Basic code:

**DO** index-variable=value1, value2, value3... ;

SAS statements

**END;**

* + - *values* can be character or numeric
    - When the **DO** loop executes, it executes once for each item in the series. The index variable equals the value of the current item. You must use commas to separate items in the series
* Eg:
* all numeric values

**DO** index-variable=2,5,9,13,27;

* all character values, with each value enclosed in quotation marks

**DO** index-variable='MON','TUE','WED','THR','FRI';

* all variable names—the index variable takes on the values of the specified variables.

**DO** index-variable='MON','TUE','WED','THR','FRI';

* 注意：Variable names must represent either all numeric or all character values.
* Nesting **DO** Loops

Putting a **DO** loop within a **DO** loop is called nesting

Eg:

**DATA** work.earn;

**DO** year=1 **TO** 20;

Capital+2000;

**DO** month=1 TO 12;

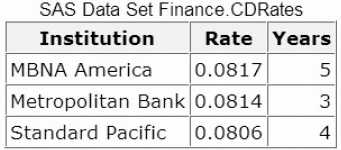
Interest=capital\*(.075/12);

capital+interest;

**END**;

**END**;

**RUN**;



1. Iteratively Processing Observations from a Data Set

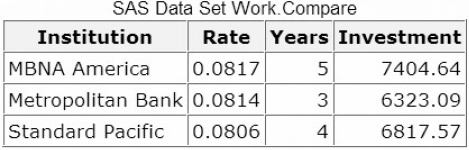
The iterative DO statement specifies a fixed number of iterations for the DO loop

**DATA** step that reads a data set to compute the value of a new variable

Eg：

**DATA** work.compare(**DROP**=i);

**SET** finance.cdrates;

Investment=5000;

**DO** i=1 **TO** years;

investment+rate\*investment;

**END**;

**RUN**;

1. Conditionally Executing **DO** Loops

When you don’t know how many times that the **DO** loop needs to be executed, the **DO WHILE** and **DO UNTIL** statements enable you to execute DO loops based on whether a condition is true or false.

* Using the **DO UNTIL** Statement

The DO UNTIL statement executes a DO loop until the expression becomes true.

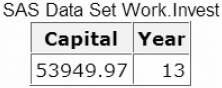
* Basic code:

**DO UNTIL**(*expression*)**;**

*SAS statements*

**END;**

*expression* is a valid SAS expression enclosed in parentheses.

* ****注意：The expression is not evaluated until the bottom of the loop, so a **DO** **UNTIL** loop always executes at least once. When the expression is evaluated as true, the **DO** loop stops.
* Eg:

**DATA** work.invest;

**DO** **UNTIL**(Capital>=50000);

capital+2000;

capital+capital\*.10;

Year+1;

**END**;

**RUN**;

* Using the **DO** **WHILE** Statement
* Basic code:

**DO UNTIL**(*expression*)**;**

*SAS statements*

**END;**

*expression* is a valid SAS expression enclosed in parentheses

* 注意：An important difference between the **DO** **UNTIL** and **DO** **WHILE** statements is that the **DO** **WHILE** expression is evaluated at the top of the **DO** loop. If the expression is false the first time it is evaluated, the **DO** loop never executes
* Eg:

**DATA** work.invest;

**DO** **WHILE**(Capital>=50000);

In this example, the **DO** loop does not execute, because the value of Capital is initially zero, which is less than 50,000

capital+2000;

capital+capital\*.10;

Year+1;

**END**;

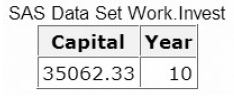
**RUN**;

1. Using Conditional Clauses with the Iterative **DO** Statement

Eg:

* In this **DATA** step, the **DO** **UNTIL** statement determines how many years it takes (13) for an investment to reach $50,000. (上一个**DO UNTIL** example)
* Suppose you also want to limit the number of years you invest your capital to 10 years. You can add the

UNTIL or WHILE expression to an iterative DO statement to further control the number of iterations. This iterative DO statement enables you to execute the DO loop until Capital is greater than or equal to **50000** or until the DO loop executes ten times, whichever occurs first.

**DATA** work.invest;

**DO** year=1 **TO** 10 **UNTIL**(Capital>=50000);

capital+2000;

capital+capital\*.10;

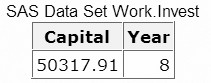
Year+1;

**END**;

**RUN**;

* In this case, the DO loop stops executing after ten iterations, and the value of Capital never reaches **50000**. If you increase the amount added to Capital each year to **4000**, the DO loop stops executing after the eighth iteration when the value of Capital exceeds **50000**.

**DATA** work.invest;

**DO** year=1 **TO** 10 **UNTIL**(Capital>=50000);

capital+4000;

capital+capital\*.10;

Year+1;

**END**;

**RUN**;

练习

1. In the data set Work.Invest, what would be the stored value for Year?

**DATA** work.invest;

**DO** year=1990 **TO** 2004;

Capital+5000;

capital+(capital\*.10);

**END**;

**RUN**;

1. missing
2. 1990
3. 2004
4. 2005
5. Which of the following statements is *false* regarding the program shown below?

**DATA** work.invest;

**DO** year=1990 **TO** 2004;

Capital+5000;

capital+(capital\*.10);

**OUTPUT**;

**END**;

**RUN**;

1. The OUTPUT statement writes current values to the data set immediately.
2. The last value for Year in the new data set is 2005.
3. The OUTPUT statement overrides the automatic output at the end of the DATA step.
4. The DO loop performs 15 iterations.
5. Which of the following would you use to compare the result of investing $4,000 a year for five years in three different banks that compound interest monthly? Assume a fixed rate for the five-year period.
6. DO WHILE statement
7. nested DO loops
8. DO UNTIL statement
9. a DO group