

Generating data with DO loops

You can execute SAS statements repeatedly by placing them in a DO loop. DO loops can execute any number of times in a single iteration of the DATA step. Using DO loops lets you write concise DATA steps that are easier to change and debug, and also can greatly reduce the number of statements required for a repetitive calculation.



Constructing DO Loops

To construct a DO loop, you use the DO and END statements along with other SAS statements.

General form, simple iterative DO loop:

```
DO index-variable = start TO stop BY increment;  
    SAS statements
```

```
END;
```

where the start, stop, and increment values

- are set upon entry into the DO loop
- cannot be changed during the processing of the DO loop
- can be numbers, variables, or SAS expressions.

The END statement terminates the loop.

Additional Note: The value of the index variable can be changed within the loop.



```
DO index-variable=start TO stop BY increment;  
  SAS statements  
END;
```

When creating a DO loop with the iterative DO statement, you must specify an index variable. The index variable stores the value of the current iteration of the DO loop. You may use any valid SAS name.

Next, specify the conditions that execute the DO loop. A simple specification contains a start value, a stop value, and an increment value for the DO loop.

The start value specifies the initial value of the index variable.

The TO clause specifies the stop value. The stop value is the last index value that executes the DO loop.

The optional BY clause specifies an increment value for the index variable. Typically, you want the DO loop to increment by 1 for each iteration. If you do not specify a BY clause, the default increment value is 1.



```
DO index-variable=start TO stop BY increment;  
    SAS statements  
END;
```

For example, the specification below increments the index variable by 1, resulting in quiz values of 1, 2, 3, 4, and 5:

```
do quiz=1 to 5;
```

By contrast, the following specification increments the index variable by 2, resulting in rows values of 2, 4, 6, 8, 10, and 12:

```
do rows=2 to 12 by 2;
```

