

What is The Complete Guide to DO Loops in SAS?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *What is The Complete Guide to DO Loops in SAS?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=160170>

The Complete Guide to DO Loops in SAS is a comprehensive resource that provides a detailed explanation of DO loops in the SAS programming language. A DO loop is a control structure that allows repetitive tasks to be performed efficiently and effectively in SAS programs. This guide covers the basics of DO loops, such as syntax and usage, as well as more advanced topics such as nested DO loops and DO WHILE loops. It also includes practical examples and tips for optimizing DO loop performance. This guide is designed to help SAS programmers of all levels fully understand and utilize DO loops in their coding.

The Complete Guide to DO Loops in SAS

A DO loop in SAS can be used to *do* some action a certain number of times.

There are three basic DO loops in SAS:

1. DO Loop

```
data data1;
```

```
x = 0;
```

```
do i = 1 to 10;
```

```
x = i*4;
```

```
output;
```

```
end;
```

```
run;
```

What It Does: This loop performs 10 iterations, from $i = 1$ to 10, where the value in each row is equal to i multiplied by 4.

When It Stops: This loop only stops after 10 iterations have been performed.

2. DO WHILE Loop

```
data data2;  
x = 0;  
do i = 1 to 10 while(x < 20);  
x = i*4;  
output;  
end;  
run;
```

What It Does: This loop will try to perform 10 iterations, from $i = 1$ to 10, where the value in each row is equal to i multiplied by 4.

When It Stops: This loop will stop when the value of x exceeds 20 or when 10 iterations have been performed, whichever comes first.

3. DO UNTIL Loop

```
data data3;  
x = 0;
```

```
do i = 1 to 10 until(x > 30);  
x = i*4;  
output;  
end;  
run;
```

What It Does: This loop will try to perform 10 iterations, from $i = 1$ to 10, where the value in each row is equal to i multiplied by 4.

When It Stops: This loop will stop when the value of x exceeds 30 or when 10 iterations have been performed, whichever comes first.

The following examples show how to use each DO loop in practice.

Example 1: DO Loop

We can use the following DO loop to create a dataset with 10 rows:

```
/*use DO loop to create dataset*/  
data data1;  
x = 0;  
do i = 1 to 10;
```

```
x = i*4;  
output;  
end;  
run;  
  
/*view dataset*/  
proc printdata=data1;
```

Obs	x	i
1	4	1
2	8	2
3	12	3
4	16	4
5	20	5
6	24	6
7	28	7
8	32	8
9	36	9
10	40	10

The result is a dataset that contains 10 rows where the values in column i range from 1 to 10 and the values in column x range from 4 to 40.

Note that you can use drop i to drop the index column from the dataset:

/*use DO loop to create dataset*/

data data1;

x = 0;

do i = 1 to 10;

x = i*4;

output;

end;

drop i;

run;

/*view dataset*/

proc printdata=data1;

Obs	x
1	4
2	8
3	12
4	16
5	20
6	24
7	28
8	32
9	36
10	40

Example 2: DO WHILE Loop

We can use the following DO WHILE loop to create a

dataset with a variable *i* from *i* = 1 to 10, where the value in each row is equal to *i* multiplied by 4 *while* *x* is less than 20:

```
/*use DO WHILE loop to create dataset*/
```

```
data data2;
```

```
x = 0;
```

```
do i = 1 to 10 while(x < 20);
```

```
x = i*4;
```

```
output;
```

```
end;
```

```
run;
```

```
/*view dataset*/
```

```
proc printdata=data2;
```

Obs	x	i
1	4	1
2	8	2
3	12	3
4	16	4
5	20	5

Notice that the loop stopped creating rows once *x* reached 20.

Example 3: DO UNTIL Loop

We can use the following DO UNTIL loop to create a dataset with a variable *i* from *i* = 1 to 10, where the value in each row is equal to *i* multiplied by 4 until *x* is greater than 30:

```
/*use DO UNTIL loop to create dataset*/
```

```
data data3;
```

```
x = 0;
```

```
do i = 1 to 10 until(x > 30);
```

```
x = i*4;
```

```
output;
```

```
end;
```

```
run;
```

```
/*view dataset*/
```

```
proc printdata=data3;
```


Obs	x	i
1	4	1
2	8	2
3	12	3
4	16	4
5	20	5
6	24	6
7	28	7
8	32	8

Notice that the loop stopped creating rows once x exceeded 30.

Additional Resources

The following tutorials explain how to perform other common tasks in SAS: