

How can strings be split by a delimiter using the SAS programming language?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can strings be split by a delimiter using the SAS programming language?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164382>

SAS is a programming language used for data analysis and management. It provides various built-in functions to manipulate strings, including splitting a string into smaller parts based on a delimiter. A delimiter is a character or a set of characters used to separate different values within a string. To split a string by a delimiter in SAS, the user can use the "scan" function which takes the string and the delimiter as input parameters and returns the desired portion of the string. This function can be used in various scenarios, such as extracting specific information from a larger string, or separating a list of values into individual elements. Additionally, SAS also provides the "substr" function which allows the user to extract a specific portion of a string based on its starting and ending positions. By using these functions, strings can be efficiently split into smaller chunks, providing the user with more flexibility and control over their data.

SAS: Split Strings by Delimiter

You can use the scan() function in SAS to quickly split a string based on a particular delimiter.

The following example shows how to use this function in practice.

Example: Split Strings by Delimiter in SAS

Suppose we have the following dataset in SAS:

```
/*create dataset*/  
data my_data1;  
input name $25.;  
datalines;  
Andy_Lincoln_Bernard  
Barry_Michael  
Chad_Simpson_Smith
```

```
Derrick_Parson_Henry  
Eric_Miller  
Frank_Giovanni_Goodwill  
;  
run;
```

```
/*print dataset*/  
proc printdata=my_data1;
```

Obs	name
1	Andy_Lincoln_Bernard
2	Barry_Michael
3	Chad_Simpson_Smith
4	Derrick_Parson_Henry
5	Eric_Miller
6	Frank_Giovanni_Goodwill

We can use the following code to quickly split the name string into three separate strings:

```
/*create second dataset with name split into three  
columns*/  
data my_data2;  
set my_data1;  
name1=scan(name, 1, '_');  
name2=scan(name, 2, '_');
```

```
name3=scan(name, 3, '_');  
run;
```

```
/*view second dataset*/  
proc printdata=my_data2;
```

Obs	name	name1	name2	name3
1	Andy_Lincoln_Bernard	Andy	Lincoln	Bernard
2	Barry_Michael	Barry	Michael	
3	Chad_Simpson_Smith	Chad	Simpson	Smith
4	Derrick_Parson_Henry	Derrick	Parson	Henry
5	Eric_Miller	Eric	Miller	
6	Frank_Giovanni_Goodwill	Frank	Giovanni	Goodwill

Notice that the string in the name column has been split into three new columns.

For the names where there was only one delimiter, the value in the name3 column is simply blank.

Note that we could also use the drop function to drop the original name column from the new dataset:

```
/*create second dataset with name split into three  
columns, drop original name*/  
data my_data2;
```

```
set my_data1;  
name1=scan(name, 1, '_');  
name2=scan(name, 2, '_');  
name3=scan(name, 3, '_');  
drop name;  
run;
```

```
/*view second dataset*/  
proc printdata=my_data2;
```

Obs	name1	name2	name3
1	Andy	Lincoln	Bernard
2	Barry	Michael	
3	Chad	Simpson	Smith
4	Derrick	Parson	Henry
5	Eric	Miller	
6	Frank	Giovanni	Goodwill

Additional Resources

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