

How can the SCAN function be used in SAS, and what are some examples of its usage?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can the SCAN function be used in SAS, and what are some examples of its usage?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154185>

The SCAN function in SAS is a powerful tool that allows users to extract specific words or phrases from a string of text. It is primarily used to search for a particular pattern or delimiter within a string and return the desired substring. This function can be extremely useful in data processing and analysis, as it enables users to manipulate and extract information from large datasets.

One example of using the SCAN function is in cleaning and organizing messy data. Often, data may contain extra characters or words that need to be removed in order to properly analyze it. With the SCAN function, users can easily identify and remove these unwanted elements.

Another example is in extracting specific information from a string. For instance, if a dataset contains a column with full names, the SCAN function can be used to extract just the first or last name. This can be particularly useful in creating new variables or merging datasets.

In addition, the SCAN function can be used to split a string into multiple variables. This is commonly used when dealing with data that is separated by delimiters, such as commas or spaces. The function can be used to split the string at the designated delimiter and create new variables for each element.

Overall, the SCAN function in SAS offers a wide range of capabilities and can be applied in various scenarios, making it a valuable tool for data manipulation and analysis.

Use the SCAN Function in SAS (With Examples)

You can use the SCAN function in SAS to extract the *n*th word from a string.

This function uses the following basic syntax:

SCAN(string, count)

where:

string: The string to analyze
count: The *n*th word to extract

Here are the three most common ways to use this function:

Method 1: Extract *n*th Word from String

```
data new_data;  
set original_data;  
second_word = scan(string_variable, 2);  
run;
```

Method 2: Extract Last Word from String

```
data new_data;  
set original_data;  
last_word = scan(string_variable, -1);  
run;
```

Method 3: Extract Multiple Words from String

```
data new_data;  
set original_data;  
first_word = scan(string_variable, 1);  
second_word = scan(string_variable, 2);  
third_word = scan(string_variable, 3);  
run;
```

The following examples show how to use each method with the following dataset in SAS:

```
/*create dataset*/  
data original_data;  
input name $20. sales;  
datalines;  
Andy Lincoln Bernard 55  
Barren Michael Smith 41  
Chad Simpson Arnolds 13  
Derrick Parson Henry 29  
Eric Miller Johansen 47  
Frank Giovanni Goode 61  
;  
run;  
  
/*view dataset*/  
proc printdata=original_data;
```

| Obs | name | sales |
|-----|----------------------|-------|
| 1 | Andy Lincoln Bernard | 55 |
| 2 | Barren Michael Smith | 41 |
| 3 | Chad Simpson Arnolds | 13 |
| 4 | Derrick Parson Henry | 29 |
| 5 | Eric Miller Johansen | 47 |
| 6 | Frank Giovanni Goode | 61 |

Example 1: Extract *n*th Word from String

The following code shows how to extract the second word from each string in the name column:

```
/*extract second word in each row of name column*/  
data new_data;  
set original_data;  
second_word = scan(name, 2);  
run;  
  
/*view results*/  
proc printdata=new_data;
```

| Obs | name | sales | second_word |
|-----|----------------------|-------|-------------|
| 1 | Andy Lincoln Bernard | 55 | Lincoln |
| 2 | Barren Michael Smith | 41 | Michael |
| 3 | Chad Simpson Arnolds | 13 | Simpson |
| 4 | Derrick Parson Henry | 29 | Parson |
| 5 | Eric Miller Johansen | 47 | Miller |
| 6 | Frank Giovanni Goode | 61 | Giovanni |

Notice that the new column called `second_word` contains the second word from each string in the name column.

Example 2: Extract Last Word from String

The following code shows how to use the value `-1` in the scan function to extract the last word from each string in the name column:

```
/*extract last word in each row of name column*/
```

```
data new_data;
```

```
set original_data;
```

```
last_word = scan(name, -1);
```

```
run;
```

```
/*view results*/
```

```
proc printdata=new_data;
```

| Obs | name | sales | last_word |
|-----|----------------------|-------|-----------|
| 1 | Andy Lincoln Bernard | 55 | Bernard |
| 2 | Barren Michael Smith | 41 | Smith |
| 3 | Chad Simpson Arnolds | 13 | Arnolds |
| 4 | Derrick Parson Henry | 29 | Henry |
| 5 | Eric Miller Johansen | 47 | Johansen |
| 6 | Frank Giovanni Goode | 61 | Goode |

Notice that the new column called last_word contains the last word from each string in the name column.

Example 3: Extract Multiple Words from String

The following code shows how to use the scan function to extract every word from each string in the name column:

```
/*extract each word in each row of name column*/  
data new_data;  
set original_data;  
first_word = scan(name, 1);  
second_word = scan(name, 2);  
third_word = scan(name, 3);  
run;  
  
/*view results*/
```

proc printdata=new_data;

| Obs | name | sales | first_word | second_word | third_word |
|-----|----------------------|-------|------------|-------------|------------|
| 1 | Andy Lincoln Bernard | 55 | Andy | Lincoln | Bernard |
| 2 | Barren Michael Smith | 41 | Barren | Michael | Smith |
| 3 | Chad Simpson Arnolds | 13 | Chad | Simpson | Arnolds |
| 4 | Derrick Parson Henry | 29 | Derrick | Parson | Henry |
| 5 | Eric Miller Johansen | 47 | Eric | Miller | Johansen |
| 6 | Frank Giovanni Goode | 61 | Frank | Giovanni | Goode |

Notice that three new columns have been created that contain the first, second, and third word from each string in the name column.

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