

How can the `ntile()` function in `dplyr` be used? Can you provide examples?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can the `ntile()` function in `dplyr` be used? Can you provide examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155516>

The ntile() function in dplyr is used to divide a numerical variable into equal sized groups or bins. This function is especially useful when working with large datasets, as it allows for efficient and accurate grouping of data points.

To use the ntile() function, one must first specify the variable to be grouped and the number of bins to be created. The function then assigns a number to each data point, indicating which bin it belongs to. This allows for further analysis and comparison within each group.

For example, if we have a dataset containing the test scores of 100 students, we can use the ntile() function to divide them into 5 equal groups based on their scores. This would allow us to compare the performance of each group and identify any patterns or trends.

Another example could be using the ntile() function to categorize income levels of individuals in a dataset. By specifying the number of bins, we can divide the income variable into groups such as low, medium, and high income, allowing for further analysis on the different income groups.

In summary, the ntile() function in dplyr is a helpful tool for dividing numerical data into equal groups, providing a structured and organized approach to data analysis.

Use the ntile() Function in dplyr (With Examples)

You can use the ntile() function from the package in R to break up an input vector into n buckets.

This function uses the following basic syntax:

ntile(x, n)

where:

x: Input vector: Number of buckets

Note: The size of the buckets can differ by up to one.

The following examples show how to use this function in practice.

Example 1: Use ntile() with a Vector

The following code shows how to use the ntile() function to break up a vector with 11 elements into 5 different buckets:

```
library(dplyr)  
  
#create vector  
x <- c(1, 3, 4, 6, 7, 8, 10, 13, 19, 22, 23)  
  
#break up vector into 5 buckets  
ntile(x, 5)  
  
1 1 1 2 2 3 3 4 4 5 5
```

From the output we can see that each element from the original vector has been placed into one of five buckets.

The smallest values are assigned to bucket 1 while the largest values are assigned to bucket 5.

For example:

The smallest values of 1, 3, and 4 are assigned to bucket 1. The largest values of 22 and 23 are assigned to bucket 5.

Example 2: Use ntile() with a Data Frame

Suppose we have the following data frame in R that shows the points scored by various basketball players:

```
#create data frame
```

```
df <- data.frame(player=LETTERS,  
points=c(12, 19, 7, 22, 24, 28, 30, 19, 15))
```

```
#view data frame
```

```
df
```

```
player points
```

```
1 A 12
```

```
2 B 19
```

```
3 C 7
```

```
4 D 22
```

```
5 E 24
```

```
6 F 28
```

```
7 G 30
```

```
8 H 19
```

```
9 I 15
```

```
library(dplyr)
```

```
#create new column that assigns players into buckets  
based on points
```

```
df$bucket <- ntile(df$points, 3)
```

```
#view updated data frame
```

```
df
```

```
player points bucket
```

```
1 A 12 1
```

```
2 B 19 2
```

```
3 C 7 1
```

```
4 D 22 2
```

```
5 E 24 3
```

```
6 F 28 3
```

```
7 G 30 3
```

```
8 H 19 2
```

```
9 I 15 1
```

The new bucket column assigns a value between 1 and 3 to each player.

The players with the lowest points receive a value of 1 and the players with the highest points receive a value

of 3.

ARABPSYCHOLOGY.COM