

How to Count Distinct Values in PySpark: 3 Easy Methods

Authored by
stats writer

February 8, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Count Distinct Values in PySpark: 3 Easy Methods*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129804>

Counting the distinct values in PySpark can be done using three different methods: the ".distinct()" function, the ".agg()" function, and the "pivot" function.

The ".distinct()" function returns a new DataFrame with unique rows, making it a simple and efficient way to count distinct values. The ".agg()" function allows for more customization by allowing the use of aggregate functions such as "countDistinct()" to specifically count distinct values. Lastly, the "pivot" function can be used to pivot the data and count distinct values across different columns.

Each of these methods offers a unique approach to counting distinct values in PySpark and can be chosen based on the specific needs of the user. By using these methods, accurate and efficient counting of distinct values can be achieved in PySpark.

Count Distinct Values in PySpark (3 Methods)

You can use the following methods to count distinct values in a PySpark DataFrame:

Method 1: Count Distinct Values in One Column

```
from pyspark.sql.functions import col, countDistinct
df.agg(countDistinct(col('my_column')).alias('my_column')).show()
```

Method 2: Count Distinct Values in Each Column

```
from pyspark.sql.functions import col, countDistinct

df.agg(*(countDistinct(col(c)).alias(c) for c in df.columns)).show()
```

Method 3: Count Number of Distinct Rows in DataFrame

df.distinct().count()

The following examples show how to use each method in practice with the following PySpark DataFrame that contains information about various basketball players:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
```

#define data

data = ,

,

,

,

,

,

,

]

#define column names

columns =

#create dataframe using data and column names

```
df = spark.createDataFrame(data, columns)
```

```
#view dataframe
```

```
df.show()
```

```
+----+-----+-----+
```

```
|team|position|points|
```

```
+----+-----+-----+
```

```
| A| Guard| 11|
```

```
| A| Guard| 8|
```

```
| A| Forward| 22|
```

```
| A| Forward| 22|
```

```
| B| Guard| 14|
```

```
| B| Guard| 14|
```

```
| B| Forward| 13|
```

```
| B| Forward| 7|
```

```
+----+-----+-----+
```

Example 1: Count Distinct Values in One Column

We can use the following syntax to count the number of distinct values in just the team column of the DataFrame:

```
from pyspark.sql.functions import col, countDistinct
```

```
#count number of distinct values in 'team' column  
df.agg(countDistinct(col('team')).alias('team')).show()
```

```
+----+  
|team|  
+----+  
| 2|  
+----+
```

From the output we can see that there are 2 distinct values in the team column.

Example 2: Count Distinct Values in Each Column

We can use the following syntax to count the number of distinct values in each column of the DataFrame:

```
from pyspark.sql.functions import col, countDistinct  
  
#count number of distinct values in each column  
df.agg(*(countDistinct(col(c)).alias(c) for c in  
df.columns)).show()
```

```
+----+-----+-----+  
|team|position|points|  
+----+-----+-----+
```

```
| 2| 2| 6|
```

```
+----+-----+-----+
```

From the output we can see:

There are 2 unique values in the team column. There are 2 unique values in the position column. There are 6 unique values in the points column.

Example 3: Count Distinct Values in Each Column

We can use the following syntax to count the number of distinct rows in the DataFrame:

```
#count number of distinct rows in DataFrame  
df.distinct().count()
```

```
6
```

From the output we can see that there are 6 distinct rows in the DataFrame.

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