

How can I use Pandas to retrieve only unique values from an index column in a dataframe?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use Pandas to retrieve only unique values from an index column in a dataframe?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153613>

Pandas is a powerful data analysis library in Python that allows users to manipulate and analyze data in a tabular form. One useful feature of Pandas is the ability to retrieve only unique values from an index column in a dataframe. This means that the index column, which contains labels for the rows in the dataframe, will only display distinct values without any duplicates. This can be achieved by using the built-in function "unique()" in combination with the dataframe's index column. By using this feature, users can easily filter out redundant or repeated values from their data, making it easier to analyze and draw insights from the dataframe.

Pandas: Get Unique Values from Index Column

You can use the following methods to get the unique values from the index column of a pandas DataFrame:

Method 1: Get Unique Values from Index Column

```
df.index.unique()
```

Method 2: Get Unique Values from Specific Column in MultiIndex

```
df.index.unique('some_column')
```

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

Example 1: Get Unique Values from Index Column

Suppose we have the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'points': ,
'assists': },
index = )
```

```
#view DataFrame
print(df)
```

```
team points assists
```

```
0 A 18 5
```

```
1 B 22 7
```

```
1 C 19 7
```

```
1 D 14 9
```

```
2 E 14 12
```

```
2 F 11 9
```

```
3 G 20 9
```

```
4 H 28 4
```

We can use the following syntax to get the unique values from the index column of the DataFrame:

```
#get unique values from index column
```

```
df.index.unique()
```

```
Int64Index(, dtype='int64')
```

The output displays each of the unique values from the index column.

We can also use the `len()` function to count the number of unique values in the index column:

```
#count number of unique values in index column  
len(df.index.unique())
```

```
5
```

We can see that there are 5 unique values in the index column of the DataFrame.

Example 2: Get Unique Values from Specific Column in MultiIndex

Suppose we have the following pandas DataFrame:

```
import pandas as pd  
#define index values  
index_names = pd.MultiIndex.from_tuples(  
names=)
```

```
#define data values
```

```
data = {'Sales': }
```

```
#create DataFrame
```

```
df = pd.DataFrame(data, index=index_names)
```

```
#view DataFrame
```

```
print(df)
```

```
Sales
```

```
Division Team
```

```
West A 12
```

```
A 44
```

```
B 29
```

```
East C 35
```

```
C 44
```

```
D 19
```

Notice that this DataFrame has a multiIndex.

We can use the following syntax to get the unique values from just the Team column from the multiIndex:

```
#get unique values from Team column in multiIndex
```

```
df.index.unique('Team')
```

```
Index(, dtype='object', name='Team')
```

The output displays the four unique values from the Team column of the multiIndex: A, B, C, and D.

We can use similar syntax to extract the unique values from the Division column of the multiIndex:

```
#get unique values from Division column in multiIndex  
df.index.unique('Division')
```

```
Index(, dtype='object', name='Division')
```

The output displays the two unique values from the Division column of the multiIndex: West and East.

The following tutorials explain how to perform other common functions in pandas: