

How can I select only unique rows in a Pandas DataFrame?

Authored by
stats writer

May 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I select only unique rows in a Pandas DataFrame?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=141565>

Pandas DataFrame is a popular data manipulation tool used in Python. It allows users to store and manipulate tabular data efficiently. One common problem that arises while working with data is the presence of duplicate rows. To address this issue, Pandas provides a function called "drop_duplicates()" that allows users to easily select only unique rows in a DataFrame. This function identifies and removes duplicate rows, leaving behind a DataFrame with only unique rows. This ensures that the data being analyzed is accurate and avoids any bias caused by duplicate entries. The "drop_duplicates()" function in Pandas makes it easy to handle duplicate rows and is a useful tool for data cleaning and analysis.

Select Unique Rows in a Pandas DataFrame

You can use the following syntax to select unique rows in a pandas DataFrame:

```
df = df.drop_duplicates()
```

And you can use the following syntax to select unique rows across specific columns in a pandas DataFrame:

```
df = df.drop_duplicates(subset=)
```

The following examples show how to use this syntax in practice with the following pandas DataFrame:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'a': ,
```

```
'b': ,  
'c': })
```

```
#view DataFrame
```

```
df
```

```
a b c
```

```
0 4 2 2
```

```
1 4 2 2
```

```
2 3 6 9
```

```
3 8 8 9
```

Example 1: Select Unique Rows Across All Columns

The following code shows how to select unique rows across all columns of the pandas DataFrame:

```
#drop duplicates from DataFrame
```

```
df = df.drop_duplicates()
```

```
#view DataFrame
```

```
df
```

```
a b c
```

```
0 4 2 2
```

```
2 3 6 9
```

3 8 8 9

The first and second row were duplicates, so pandas dropped the second row.

By default, the `drop_duplicates()` function will keep the first duplicate. However, you can specify to keep the last duplicate instead:

```
#drop duplicates from DataFrame, keep last duplicate  
df = df.drop_duplicates(keep='last')
```

```
#view DataFrame
```

```
df
```

```
a b c
```

```
1 4 2 2
```

```
2 3 6 9
```

```
3 8 8 9
```

Example 2: Select Unique Rows Across Specific Columns

The following code shows how to select unique rows across just column 'c' in the DataFrame:

```
#drop duplicates from column 'c' in DataFrame
```

```
df = df.drop_duplicates(subset=)
```

```
#view DataFrame
```

```
df
```

```
a b c
```

```
0 4 2 2
```

```
2 3 6 9
```

Two rows were dropped from the DataFrame.

ARABPSYCHOLOGY.COM