

How can an anti-join be performed in Pandas?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can an anti-join be performed in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155039>

An anti-join in Pandas is a method for combining two datasets while excluding rows that have matching values in a specified column. This can be performed by using the "merge" function in Pandas, with the "how" parameter set to "outer" and the "indicator" parameter set to True. This will create a column indicating which rows are present in the original datasets and which are not. The resulting dataset will only contain non-matching rows from the original datasets. This method is useful for finding differences between datasets or identifying unique values.

Perform an Anti-Join in Pandas

An anti-join allows you to return all rows in one dataset that do not have matching values in another dataset.

You can use the following syntax to perform an anti-join between two pandas DataFrames:

```
outer = df1.merge(df2, how='outer', indicator=True)
```

```
anti_join = outer.drop('_merge', axis=1)
```

The following example shows how to use this syntax in practice.

Example: Perform an Anti-Join in Pandas

Suppose we have the following two pandas DataFrames:

```
import pandas as pd
```

```
#create first DataFrame
```

```
df1 = pd.DataFrame({'team': ,  
'points': })
```

```
print(df1)
```

```
team points
```

```
0 A 18
```

```
1 B 22
```

```
2 C 19
```

```
3 D 14
```

```
4 E 30
```

```
#create second DataFrame
```

```
df2 = pd.DataFrame({'team': ,  
'points': })
```

```
print(df2)
```

```
team points
```

```
0 A 18
```

```
1 B 22
```

```
2 C 19
```

```
3 F 22
```

```
4 G 29
```

We can use the following code to return all rows in the first DataFrame that do not have a matching team in the second DataFrame:

```
#perform outer join
outer = df1.merge(df2, how='outer', indicator=True)

#perform anti-join
anti_join = outer.drop('_merge', axis=1)

#view results
print(anti_join)

team points
3 D 14
4 E 30
```

We can see that there are exactly two teams from the first DataFrame that do not have a matching team name in the second DataFrame.

The anti-join worked as expected.

The end result is one DataFrame that only contains the rows where the team name belongs to the first DataFrame but not the second DataFrame.

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

ARABPSYCHOLOGY.COM