

# How can I drop duplicates across multiple columns in Pandas?

Authored by  
**stats writer**

July 1, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I drop duplicates across multiple columns in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164864>

Pandas is a popular Python library used for data analysis and manipulation. One common task in data analysis is removing duplicate values from a dataset. In Pandas, this can be achieved by using the "drop\_duplicates" function. This function allows for the removal of duplicates across a single column or multiple columns. By specifying the columns to be checked for duplicates, Pandas will identify and remove any rows that have identical values in those columns. This process is useful for cleaning and organizing data, ensuring accuracy and efficiency in data analysis.

## **Pandas: Drop Duplicates Across Multiple Columns**

**You can use the following methods to drop duplicate rows across multiple columns in a pandas DataFrame:**

### **Method 1: Drop Duplicates Across All Columns**

```
df.drop_duplicates()
```

### **Method 2: Drop Duplicates Across Specific Columns**

```
df.drop_duplicates()
```

**The following examples show how to use each method in practice with the following pandas DataFrame:**

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'region': ,  
'store': ,
```

```
'sales': })
```

```
#view DataFrame
```

```
print(df)
```

```
region store sales
```

```
0 East 1 5
```

```
1 East 1 5
```

```
2 East 2 7
```

```
3 West 1 9
```

```
4 West 2 12
```

```
5 West 2 8
```

Example 1: Drop Duplicates Across All Columns

The following code shows how to drop rows that have duplicate values across all columns:

```
#drop rows that have duplicate values across all columns
```

```
df.drop_duplicates()
```

```
region store sales
```

```
0 East 1 5
```

```
2 East 2 7
```

```
3 West 1 9
```

**4 West 2 12**

**5 West 2 8**

The row in index position 1 had the same values across all columns as the row in index position 0, so it was dropped from the DataFrame.

By default, pandas keeps the first duplicate row. However, you can use the `keep` argument to specify to keep the last duplicate row instead:

**#drop rows that have duplicate values across all columns (keep last duplicate)**

```
df.drop_duplicates(keep='last')
```

**region store sales**

**1 East 1 5**

**2 East 2 7**

**3 West 1 9**

**4 West 2 12**

**5 West 2 8**

**Example 2: Drop Duplicates Across Specific Columns**

**You can use the following code to drop rows that have**

**duplicate values across only the region and store columns:**

**#drop rows that have duplicate values across region and store columns**  
**df.drop\_duplicates()**

**region store sales**

**0 East 1 5**

**2 East 2 7**

**3 West 1 9**

**4 West 2 12**

**A total of two rows were dropped from the DataFrame because they had duplicate values in the region and store columns.**

**Additional Resources**

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