

# How to Map Column Values in a Pandas DataFrame?



Gen. Devin DL. · [Follow](#)

3 min read · Dec 24, 2024



How to Map Column Values  
in a Pandas DataFrame?

Mapping column values refers to replacing specific values in a column with other values, commonly used in data cleaning and transformation.

There are many scenarios where mapping column values is useful. Here are a few common use cases:

1. Mapping certain string values in a column to numbers. For example, mapping “Male” and “Female” to 0 and 1, respectively, to facilitate training and prediction in machine learning algorithms.

2. Replacing abbreviations with full names. For instance, replacing “USA” and “UK” with “United States” and “United Kingdom” to make the data more readable.
3. Correcting mis-spelled words by replacing them with the correct spelling. For example, replacing “Cocacola” with “Coca-Cola” to avoid errors in statistics and analysis.

In this post we will introduce several common mapping techniques.

## Using the `map` function

The `map` function is one of the simplest and most direct methods. For example, in the following case, mapping gender values to 0 and 1.

```
import pandas as pd

df = pd.DataFrame({
    "name": ["Alice", "Bob", "Charlie", "David", "Eva", "Frank"],
    "sex": ["female", "male", "male", "male", "female", "male"],
    "age": [23, 34, 29, 42, 25, 31],
})

df.sex = df.sex.map({
    "female": 0,
    "male": 1,
})

print(df)
```

Before mapping:

	name	sex	age
0	Alice	female	23
1	Bob	male	34
2	Charlie	male	29
3	David	male	42
4	Eva	female	25
5	Frank	male	31

After mapping:

	name	sex	age
0	Alice	0	23
1	Bob	1	34
2	Charlie	1	29
3	David	1	42
4	Eva	0	25
5	Frank	1	31

## Factorize Mapping

Using the `map` function to map column values is the most straightforward method. However, when there are many distinct values in the column, mapping them one by one can be cumbersome. For example, in the case of the `grade` column in the following example, unlike the `sex` column which only has two values, it contains more possible values.

In this case, we can use the `factorize` method to perform the mapping.

```
import pandas as pd

df = pd.DataFrame({
    "name": ["John", "Emily", "Michael", "Sarah", "James", "Olivia"],
    "sex": ["male", "female", "male", "female", "male", "female"],
    "grade": ["A", "B", "C", "A", "B", "C"],
})

df.sex = df.sex.factorize()[0]
df.grade = df.grade.factorize()[0]
print(df)
```

Before mapping:

	name	sex	grade
0	John	male	A
1	Emily	female	B
2	Michael	male	C
3	Sarah	female	A
4	James	male	B
5	Olivia	female	C

After mapping:

	name	sex	grade
0	John	0	0
1	Emily	1	1
2	Michael	0	2
3	Sarah	1	0
4	James	0	1
5	Olivia	1	2

The `factorize` function returns a tuple with two elements. The first element is an array of numbers representing the mapped values, and the second

element is an index type, where the index values correspond to the unique values in the column.

```
df.grade.factorize()
```

```
(array([0, 1, 2, 0, 1, 2], dtype=int64), Index(['A', 'B', 'C'], dtype='object'))
```

So the code uses `factorize()[0]`.

If we want to binarize the values after mapping, for example, in the case of the `grade` column, where there are four different values representing different grade levels. If we only want two categories — fail (F) and pass (non-F), then code will be:

```
df.grade = df.grade.factorize()[0]
df.grade = (df.grade == 2).astype("int")
print(df)
```

After mapping:

	name	sex	grade
0	John	male	0
1	Emily	female	0
2	Michael	male	1
3	Sarah	female	0
4	James	male	0
5	Olivia	female	1

Here A and B are mapped to 0, and C is mapped to 1.

Thanks for your reading.

Pandas Dataframe

Python Data Preprocessing

Pandas



**Written by Gen. Devin DL.**

505 Followers · 4 Following

Follow

AI practitioner & python coder to record what I learned in AI & Data Science & Python project development

**No responses yet**



What are your thoughts?

Respond

**More from Gen. Devin DL.**



M Gen. Devin DL.

## Four Ways to Package a Python Project into an executable EXE...

In Python, packaging a project into an executable EXE file is a common task,...

Sep 14, 2024 🖱 104



M Gen. Devin DL.

## Building Python ETL Data Pipelines with Five Typical Cases

Data pipelines are the backbone of data processing. Simply put, it's about moving da...

Jan 26 🖱 71



M Gen. Devin DL.

## ETL-PIPES: An Efficient Python ETL Data Processing Library

ETL-pipes is a powerful and flexible Python library specifically designed for ETL (Extract...

Nov 30, 2024 🖱 117 💬 6



M Gen. Devin DL.

## Comparison of Flask, Django, and FastAPI: Advantages,...

In Python web development, Flask, Django, and FastAPI are all popular frameworks. This...

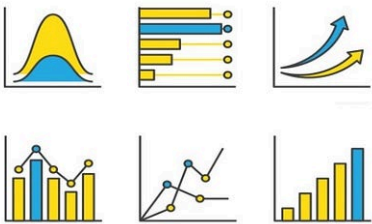
Oct 17, 2023 🖱 92



See all from Gen. Devin DL.

# Recommended from Medium

## ADVANCED STATISTICAL ALGORITHMS FOR DATA SCIENCE

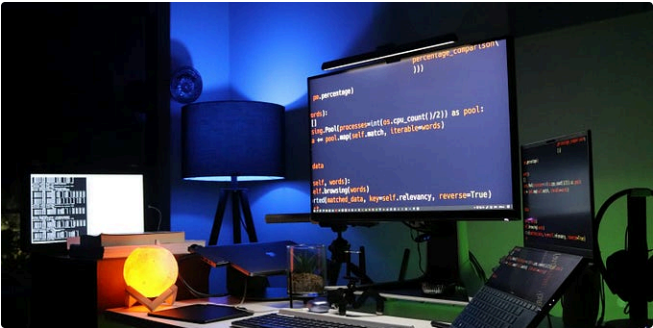


 Sarowar Jahan Saurav

### Beyond the Basics: 11 Complex Statistical Algorithms to Elevate...

Data Science is more than just running standard algorithms or crafting elegant...

Sep 13, 2024  107



 Vijay Gadhave

### When to Use COUNT(\*) vs COUNT(1) in SQL Queries

Note: If you're not a medium member, [CLICK HERE](#)

 Jan 14  399  25

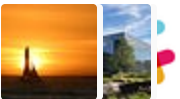


## Lists



### Staff picks

817 stories · 1632 saves



### Stories to Help You Level-Up at Work

19 stories · 943 saves



### Self-Improvement 101

20 stories · 3316 saves




### Productivity 101

20 stories · 2791 saves



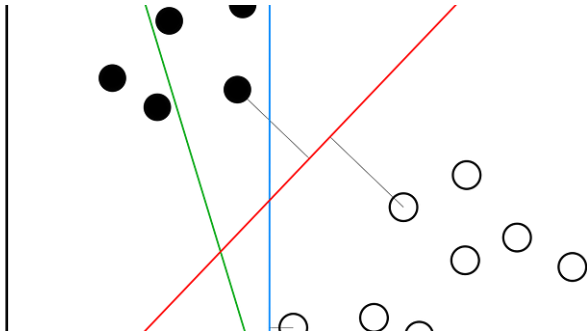



 In Stackademic by Khouloud Haddad Amamou

## Data Analysis with Python Pandas and Matplotlib

1. Introduction

Jan 30  150  8  



 In Artificial Intelligence in Plain En... by Ritesh Gu...

## Data Science All Algorithm Cheatsheet 2025

Stories, strategies, and secrets to choosing the perfect algorithm.

★ Jan 5  1.3K  27  

A	B	C	D	E	F	G	H
Order	Order Date	Product Name	Category	Sales Rep	Units Sold	Sales Amount	
1001	2024-09-10	Laptop X200	Electronics	Li Wei	5	\$1,500.00	
1002	2024-09-12	Smartwatch S5	Wearables	Zhang Ming	10	\$500.00	
1003	2024-09-14	Desktop Z400	Electronics	Wang Yun	3	\$2,000.00	
1004	2024-09-16	Wireless Headset	Accessories	Zhao Lei	15	\$300.00	
1005	2024-09-18	Tablet T10	Electronics	Chen Xia	7	\$800.00	
1006	2024-09-20	Bluetooth Speaker	Accessories	Sun Jie	20	\$250.00	
1007	2024-09-22	Smartphone Z2	Electronics	Li Wei	8	\$900.00	
1008	2024-09-24	Fitness Band F3	Wearables	Zhang Ming	25	\$180.00	
1009	2024-09-26	Monitor M30	Electronics	Wang Yun	4	\$1,200.00	
1010	2024-09-28	Keyboard K20	Accessories	Zhao Lei	50	\$100.00	

 Alice Yang

## How to Create, Update and Remove Tables in Excel with Python

Tables in Excel are a powerful tool that helps us manage, organize, and analyze data...

Oct 10, 2024  11  

```
a = [1, 2, 3]
b = [1, 2, 3]
print(a == b)
print(a is b)

True
False
```

### 2. Mutable Default Arguments

Using mutable objects (like lists or dictionaries) as default arguments in function definitions leads to unexpected behavior.

```
def add_element(element, my_list=[]):
    my_list.append(element)
    return my_list
```

 Python Coding

## 9 Confusing Python Concepts That Frustrate Most Developers

1. The is vs == Confusion

★ Oct 2, 2024  162  2  

See more recommendations