

AI Documentation

You said:

In pandas, how do I extract the month as a two-digit string from Year, Month, and Day columns and save it to a new column?

ChatGPT said:

python

CopyEdit

```
nodes['Date'] = pd.to_datetime(nodes[['Year', 'Month', 'Day']]).dt.strftime('%m')
```

You said:

What does the components means, I mean how does this workd

ChatGPT said:

Great question! Let's break down the line of code:

python

CopyEdit

```
nodes['Date'] = pd.to_datetime(nodes[['Year', 'Month', 'Day']]).dt.strftime('%m')
```

Step-by-step explanation:

1. `nodes[['Year', 'Month', 'Day']]`

- This selects the three columns Year, Month, and Day from the DataFrame nodes.
- These are assumed to be numeric or string values representing date components.

2. `pd.to_datetime(...)`

- Converts the selected columns into a single datetime object (i.e., a `datetime64[ns]` column).
- Pandas automatically understands that you want to treat the combination of Year, Month, and Day as a full date.

3. `.dt`

- This accessor allows you to work with datetime properties of the resulting column.

4. `.strftime('%m')`

- `strftime` is used to format the datetime into a string.
- `'%m'` means "month as a zero-padded decimal number" (e.g., 01, 02, ..., 12).

5. `nodes['Date'] = ...`

- Finally, the formatted month string is stored in a new column called `Date`.

Example:

Suppose you have this DataFrame:

python

CopyEdit

```
Year Month Day
```

```
0 2024    6   5
```

```
1 2025   12  25
```

After running the line, you'll get:

python

CopyEdit

```
Year Month Day Date
```

```
0 2024    6   5 06
```

```
1 2025   12  25 12
```

Let me know if you'd like the year or full date as well!

You said:

How do I filter a pandas DataFrame called `edges` to keep only rows where the 'Weight' column is greater than or equal to 0.6?

ChatGPT said:

python

CopyEdit

```
edges = edges[edges['Weight'] >= 0.6]
```

You said:

How does this functions.

2/2

ChatGPT said:

Sure! Here's how that filtering works step-by-step:

python

CopyEdit

```
edges = edges[edges['Weight'] >= 0.6]
```

Explanation:

1. edges['Weight'] >= 0.6

- This creates a **boolean mask** (a Series of True or False values) by checking each row's 'Weight' value.
- For each row, if 'Weight' is **greater than or equal to 0.6**, the result is True, otherwise False.

2. edges[...]

- This uses the boolean mask to **filter the DataFrame**.
- Only rows where the condition is True are kept.
- Rows where the condition is False are dropped.

3. edges =

- The filtered DataFrame replaces the original edges DataFrame, effectively updating it to keep only rows with 'Weight' ≥ 0.6 .

Example:

If your DataFrame looks like this:

Index	Weight	OtherColumn
-------	--------	-------------

0	0.5	A
---	-----	---

1	0.7	B
---	-----	---

2	0.6	C
---	-----	---

3	0.3	D
---	-----	---

After filtering, it will become:

Index	Weight	OtherColumn
-------	--------	-------------

1	0.7	B
---	-----	---

2	0.6	C
---	-----	---

Let me know if you want me to explain how to do this without overwriting the original DataFrame, or how to filter with multiple conditions!