How can I import a CSV file into Pandas when each row has a different number of columns?

Authored by

stats writer

June 25, 2024

Pandas is a popular data analysis library in Python that allows users to easily manipulate and analyze large datasets. One of the common tasks in data analysis is importing data from a CSV file into Pandas. However, it can be challenging when each row in the CSV file has a different number of columns. To import such a file into Pandas, users can use the "pd.read_csv()" function and specify the "error_bad_lines=False" parameter. This will skip the rows with a different number of columns and import the remaining rows into a Pandas DataFrame. This approach ensures that the data is imported accurately without any errors, enabling users to efficiently analyze the data.

Pandas: Import CSV with Different Number of Columns per Row

You can use the following basic syntax to import a CSV file into pandas when there are a different number of columns per row:

df = pd.read_csv('uneven_data.csv', header=None, names=range(4))

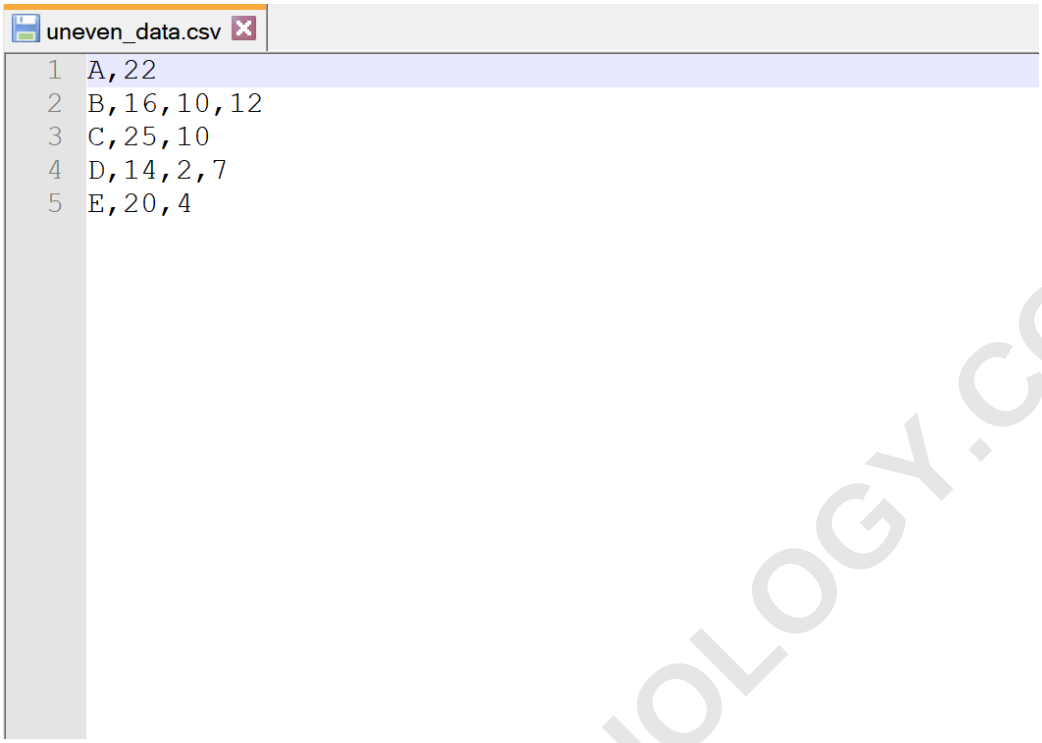
The value inside the range() function should be the number of columns in the row with the max number of columns.

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

Example: Import CSV into Pandas with Different Number of Columns per Row

Suppose we have the following CSV file called

uneven_data.csv:



Notice that each row does not have the same number of columns.

If we attempt to use the read_csv() function to import this CSV file into a pandas DataFrame, we'll receive an error:

```
import pandas as pd

#attempt to import CSV file with differing number of columns per row
```

df = pd.read_csv('uneven_data.csv', header=None)

ParserError: Error tokenizing data. C error: Expected 2 fields in line 2, saw 4

We receive a ParserError that tells us pandas expected 2 fields (since this was the number of columns in the first row) but it saw 4.

This error tells us that the max number of columns in any given row is 4.

Thus, we can import the CSV file and supply a value of range(4) to the names argument:

```
import pandas as pd

#import CSV file with differing number of columns per row
df = pd.read_csv('uneven_data.csv', header=None, names=range(4)))

#view DataFrameprint(df)

0 1 2 3
0 A 22 NaN NaN
```

1 B 16 10.0 12.0

2 C 25 10.0 NaN

3 D 14 2.0 7.0

4 E 20 4.0 NaN

Notice that we're able to successfully import the CSV file into a pandas DataFrame without any errors since we explicitly told pandas to expect 4 columns.

By default, pandas fills in any missing values in each row with NaN.

If you'd like the missing values to instead appear as zero, you can use the fillna() function as follows:

```
#fill NaN values with zeros
df_new = df.fillna(0)

#view new DataFrame
print(df_new)
```

0 1 2 3

0 A 22 0.0 0.0

1 B 16 10.0 12.0

2 C 25 10.0 0.0

3 D 14 2.0 7.0

4 E 20 4.0 0.0

Each NaN value in the DataFrame has now been replaced with a zero.

Note: You can find the complete documentation for the pandas read_csv() function .

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