How can I read a text file using Pandas, and can you provide some examples?

Authored by

stats writer

April 23, 2024

Pandas is a popular Python library used for data manipulation and analysis. It also provides useful functions for reading and writing data from various file formats, including text files. To read a text file using Pandas, one can use the `read_csv()` function which allows for easy loading of data from a CSV (comma-separated values) file. This function can also be used to read other types of delimited files such as tab-separated or pipe-separated. Additionally, Pandas also has functions for reading fixed-width formatted files and JSON files. Some examples of using Pandas to read text files include loading a CSV file containing sales data into a data frame for further analysis, reading a fixed-width file with customer information, or importing a JSON file containing stock market data. By utilizing Pandas' efficient and flexible methods, users can easily read and manipulate data from text files for their specific needs.

Read a Text File with Pandas (Including Examples)

To read a text file with pandas in Python, you can use the following basic syntax:

df = pd.read_csv("data.txt", sep=" ")

This tutorial provides several examples of how to use this function in practice.

Read a Text File with a Header

Suppose we have the following text file called data.txt with a header:

```
column1 column2
1 4
3 4
2 5
7 9
9 1
6 3
4 4
5 2
4 8
6 8
```


To read this file into a pandas DataFrame, we can use the following syntax:

```
import pandas as pd

#read text file into pandas DataFrame
df = pd.read_csv("data.txt", sep=" ")

#display DataFrameprint(df)
```

```
column1 column2
0 1 4
1 3 4
```

2 2 5

3 7 9

4 9 1

5 6 3

6 4 4

7 5 2

8 4 8

9 6 8

We can print the class of the DataFrame and find the number of rows and columns using the following syntax:

```
#display class of DataFrame
print(type(df))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
#display number of rows and columns in DataFrame
df.shape(10, 2)
```

We can see that df is a pandas DataFrame with 10 rows and 2 columns.

Read a Text File with No Header

Suppose we have the following text file called data.txt with no headers:

```
1 4
3 4
2 5
7 9
9 1
6 3
4 4
5 2
4 8
6 8
```


To read this file into a pandas DataFrame, we can use the following syntax:

```
#read text file into pandas DataFrame
df = pd.read_csv("data.txt", sep=" ", header=None)

#display DataFrameprint(df)
```

0 1

0 1 4

1 3 4

2 2 5

3 7 9

4 9 1

5 6 3

6 4 4

7 5 2

8 4 8

9 6 8

Since the text file had no headers, pandas simply named the columns 0 and 1.

Read a Text File with No Header & Specify Column Names

If we'd like, we can assign column names while importing the text file by using the names argument:

```
#read text file into pandas DataFrame and specify column names
df = pd.read_csv("data.txt", sep=" ", header=None, names=)
```

```
#display DataFrame
print(df)

A B
0 1 4
1 3 4
2 2 5
3 7 9
4 9 1
5 6 3
6 4 4
7 5 2
8 4 8
9 6 8
```

How to Read CSV Files with Pandas

How to Read Excel Files with Pandas

How to Read a JSON File with Pandas