

How can I specify dtypes when importing an Excel file in Pandas?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I specify dtypes when importing an Excel file in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152294>

When importing an Excel file in Pandas, it is possible to specify the data types (dtypes) of the columns being imported. This allows for more control over the data being imported and can prevent unexpected data type conversions. By using the "dtype" parameter in the "read_excel" function, users can specify the data types for each column in the Excel file. This helps ensure the data is accurately imported and can be used efficiently for further analysis.

Pandas: Specify dtypes when Importing Excel File

You can use the following basic syntax to specify the dtype of each column in a DataFrame when importing an Excel file into pandas:

```
df = pd.read_excel('my_data.xlsx',  
dtype = {'col1': str, 'col2': float, 'col3': int})
```

The dtype argument specifies the data type that each column should have when importing the Excel file into a pandas DataFrame.

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

Example: Specify dtypes when Importing Excel File into Pandas

Suppose we have the following Excel file called player_data.xlsx:

	A	B	C	D	E	F
1	team	points	rebounds	assists		
2	A	24	8	5		
3	B	20	12	3		
4	C	15	4	7		
5	D	19	4	8		
6	E	32	6	8		
7	F	13	7	9		
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

If we import the Excel file using the `read_excel()` function, pandas will attempt to identify the data type for each column automatically:

```
import pandas as pd
```

```
#import Excel file
```

```
df = pd.read_excel('player_data.xlsx')
```

```
#view resulting DataFrame
```

```
print(df)
```

```
team points rebounds assists
```

```
0 A 24 8 5
```

```
1 B 20 12 3
```

```
2 C 15 4 7
```

```
3 D 19 4 8
```

```
4 E 32 6 8
```

```
5 F 13 7 9
```

```
#view data type of each column
```

```
print(df.dtypes)
```

```
team object
```

```
points int64
```

```
rebounds int64
```

```
assists int64
```

```
dtype: object
```

From the output we can see that the columns in the DataFrame have the following data types:

```
team: objectpoints: int64rebounds: int64assists: int64
```

However, we can use the dtype argument within the

read_excel() function to specify the data types that each column should have:

```
import pandas as pd
```

```
#import Excel file and specify dtypes of columns
```

```
df = pd.read_excel('player_data.xlsx',  
dtype = {'team': str, 'points': float, 'rebounds': int,  
'assists': float})
```

```
#view resulting DataFrame
```

```
print(df)
```

```
team points rebounds assists
```

```
0 A 24.0 8 5.0
```

```
1 B 20.0 12 3.0
```

```
2 C 15.0 4 7.0
```

```
3 D 19.0 4 8.0
```

```
4 E 32.0 6 8.0
```

```
5 F 13.0 7 9.0
```

```
#view data type of each column
```

```
print(df.dtypes)
```

```
team object
```

```
points float64
```

rebounds int32

assists float64

dtype: object

From the output we can see that the columns in the DataFrame have the following data types:

team: object points: float64 rebounds: int32 assists: float64

These data types match the ones that we specified using the dtype argument.

Note that in this example, we specified the dtype for each column in the DataFrame.

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

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