

# How can I use pandas to read only specific rows from a CSV file?

Authored by  
**stats writer**

June 25, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I use pandas to read only specific rows from a CSV file?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152286>

Pandas is a powerful data analysis library in Python that allows for easy manipulation and processing of data. It provides various functionalities for reading and writing data in different formats, including CSV files. To read only specific rows from a CSV file using pandas, one can utilize the "iloc" function, which allows for indexing and slicing of rows and columns. By specifying the desired row numbers, pandas can extract and display only those rows from the CSV file, making it a convenient tool for filtering and selecting data. This feature is particularly useful when dealing with large datasets, as it allows for efficient extraction of relevant information for analysis.

## **Pandas: Only Read Specific Rows from CSV File**

**You can use the following basic syntax to only read in specific rows from a CSV file into a pandas DataFrame:**

```
#specify rows to import  
specific_rows =  
#import specific rows from CSV into DataFrame  
df = pd.read_csv('my_data.csv', skiprows = lambda x: x  
not in specific_rows)
```

**This particular example will read the rows in index positions 0, 2, and 3 from a CSV file called my\_data.csv into a pandas DataFrame.**

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

**Example: Only Read Specific Rows from CSV File into Pandas**

**Suppose we have the following CSV file called `basketball_data.csv`:**

```
1 team,points,rebounds
2 A, 22, 10
3 B, 14, 9
4 C, 29, 6
5 D, 30, 2
```

**If we use the `read_csv()` function, pandas will automatically import each row from the CSV file into a DataFrame:**

```
import pandas as pd
```

```
#import all rows of CSV into DataFrame
```

```
df = pd.read_csv('basketball_data.csv')
```

```
#view DataFrame
```

```
print(df)
```

**team points rebounds**

**0 A 22 10**

**1 B 14 9**

**2 C 29 6**

**3 D 30 2**

However, we can use the following syntax to only import the rows in index positions 0, 2, and 3 from the CSV file into a pandas DataFrame:

```
import pandas as pd
```

```
#specify rows to import
```

```
specific_rows =
```

```
#import specific rows from CSV into DataFrame
```

```
df = pd.read_csv('basketball_data.csv', skiprows =  
lambda x: x not in specific_rows)
```

```
#view DataFrameprint(df)
```

**team points rebounds**

**0 B 14 9**

**1 C 29 6**

**Notice that only the rows in index positions 0, 2, and 3 from the CSV file are imported into the DataFrame.**

**This syntax uses the `skiprows` argument and a lambda function to tell pandas which rows *not to skip* when importing the CSV file.**

**In this example, we tell pandas not to skip the rows in index positions 0, 2, and 3 but to skip all other rows when importing the CSV file.**

**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:**