How can we access sample datasets in Pandas?

Authored by

stats writer

June 28, 2024

Pandas, a popular Python library for data analysis, provides various methods for accessing sample datasets. These datasets can be accessed either from the internet or from local machine storage. To access datasets from the internet, Pandas offers a built-in function called "read_csv()" which can directly import datasets from a URL. Additionally, Pandas also allows users to load datasets from local storage by specifying the file path using the "read_csv()" function. Furthermore, Pandas provides a collection of pre-loaded datasets, known as "sample datasets", which can be accessed using the "get_sample_data()" function. These sample datasets serve as a useful resource for practicing and learning data analysis techniques in Pandas. In summary, by utilizing the various functions and methods provided by Pandas, users can easily access and work with sample datasets for data analysis.

Access Sample Datasets in Pandas

Often you may want to access sample datasets in pandas to play around with and practice different functions.

Fortunately you can build sample pandas datasets by using the built-in testing feature.

The following examples show how to use this feature.

Example 1: Create Pandas Dataset with All Numeric Columns

The following code shows how to create a pandas dataset with all numeric columns:

import pandas as pd

#create sample dataset

df1 = pd.util.testing.makeDataFrame()

#view dimensions of datasetprint(df1.shape)

(30, 4)

#view first five rows of datasetprint(df1.head())

A B C D
s8tpz0W5mF -0.751223 0.956338 -0.441847 0.695612
CXQ9YhLhk8 -0.210881 -0.231347 -0.227672 -0.616171
KAbcor6sQK 0.727880 0.128638 -0.989993 1.094069
lH3bptMpdb -1.599723 1.570162 -0.221688 2.194936
gaR9ZxBTrH 0.025171 -0.446555 0.169873 -1.583553

By default, the makeDataFrame() function creates a pandas DataFrame with 30 rows and 4 columns in which all of the columns are numeric.

Example 2: Create Pandas Dataset with Mixed Columns

The following code shows how to create a pandas dataset with all numeric columns:

import pandas as pd

#create sample dataset

df2 = pd.util.testing.makeMixedDataFrame()

#view dimensions of datasetprint(df2.shape)

(5, 4)

#view first five rows of datasetprint(df2.head())

A B C D
0 0.0 0.0 foo1 2009-01-01
1 1.0 1.0 foo2 2009-01-02
2 2.0 0.0 foo3 2009-01-05
3 3.0 1.0 foo4 2009-01-06
4 4.0 0.0 foo5 2009-01-07

By default, the makeMixedDataFrame() function creates a pandas DataFrame with 5 rows and 4 columns in which the columns are a variety of data types.

We can use the following code to display the:

#display data type of each column
df2.dtypes

A float64
B float64

C object
D datetime64
dtype: object

From the output we can see:

Column A is numericColumn B is numericColumn C is a stringColumn D is a date

Example 3: Create Pandas Dataset with Missing Values

The following code shows how to create a pandas dataset with some missing values in various columns:

```
import pandas as pd

#create sample dataset
df3 = pd.util.testing.makeMissingDataFrame()

#view dimensions of datasetprint(df3.shape)

(30, 4)

#view first five rows of datasetprint(df3.head())

A B C D
YgAQaNaGfG 0.444376 -2.264920 1.117377 -0.087507
```

JoT4KxJeHd 1.913939 1.287006 -0.331315 -0.392949
tyrA2P6wz3 NaN 2.988521 0.399583 0.095831
1qvPc9DU1t 0.028716 1.311452 -0.237756 -0.150362
3aAXYtXjlO -1.069339 0.332067 0.204074 NaN

By default, the makeMissingDataFrame() function creates a pandas DataFrame with 30 rows and 4 columns in which there are some missing values (NaN) in various columns.

This function is particularly useful because it allows you to work with a dataset that has some missing values, which is common in real-world datasets.

Additional Resources

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