pandas-workshop

March 10, 2023

1 Introduction to Pandas dataframe

table
excel sheet

Data frame is a main object in pandas. It is used to represent data with rows and columns

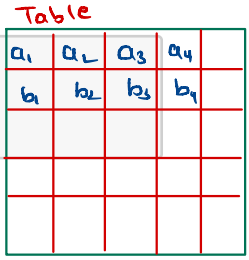
Data frame is a datastructure represent the data in tabular or excel spread sheet like data)

creating dataframe:

Comma Seperated Value

```
[ ]: import pandas as pd
     df = pd.read_csv("weather_data.csv")   #read weather.csv data
     df
```

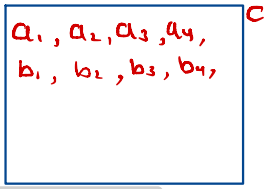
Table




```
[ ]:         day  temperature  windspeed  event
0  1/1/2017           32          6   Rain
1  1/2/2017           35          7  Sunny
2  1/3/2017           28          2   Snow
3  1/4/2017           24          7   Snow
4  1/5/2017           32          4   Rain
5  1/6/2017           31          2  Sunny
```

indexing

Store in Comma separated value



CSV file

There are other ways to construct data.

```
[ ]: #list of tuples

     weather_data = [('1/1/2017', 32, 6, 'Rain'),
                     ('1/2/2017', 35, 7, 'Sunny'),
                     ('1/3/2017', 28, 2, 'Snow'),
                     ('1/4/2017', 24, 7, 'Snow'),
                     ('1/5/2017', 32, 4, 'Rain'),
                     ('1/6/2017', 31, 2, 'Sunny')
                    ]
     df = pd.DataFrame(weather_data, columns=['day', 'temperature', 'windspeed',␣
      ↪'event'])
     df
```

What we are doing here is we are taking 'csv' file & using panda to read this into data_frame.

CSV → Panda read CSV → dataframe

```
[ ]:         day  temp  windspeed  event
0  1/1/2017    32          6   Rain
1  1/2/2017    35          7  Sunny
2  1/3/2017    28          2   Snow
3  1/4/2017    24          7   Snow
4  1/5/2017    32          4   Rain
```

```
5  1/6/2017    31          2  Sunny
```

[ ]:
```
#get dimentions of the table

df.shape   #total number of rows and columns
```

[ ]:
```
(6, 4)
```

[ ]:
```
#if you want to see initial some rows then use head command (default 5 rows)
df.head()
```

[ ]:
```
         day  temperature  windspeed  event
0  1/1/2017           32          6   Rain
1  1/2/2017           35          7  Sunny
2  1/3/2017           28          2   Snow
3  1/4/2017           24          7   Snow
4  1/5/2017           32          4   Rain
```

[ ]:
```
#if you want to see last few rows then use tail command (default last 5 rows will print)
df.tail()
```

[ ]:
```
         day  temperature  windspeed  event
1  1/2/2017           35          7  Sunny
2  1/3/2017           28          2   Snow
3  1/4/2017           24          7   Snow
4  1/5/2017           32          4   Rain
5  1/6/2017           31          2  Sunny
```

[ ]:
```
#slicing
df[2:5]
```

[ ]:
```
         day  temperature  windspeed event
2  1/3/2017           28          2  Snow
3  1/4/2017           24          7  Snow
4  1/5/2017           32          4  Rain
```

[ ]:
```
df.columns   #print columns in a table
```

[ ]:
```
Index(['day', 'temperature', 'windspeed', 'event'], dtype='object')
```

[ ]:
```
df.day      #print particular column data
```

[ ]:
```
0    1/1/2017
1    1/2/2017
2    1/3/2017
3    1/4/2017
```

```
4    1/5/2017
5    1/6/2017
Name: day, dtype: object
```

[ ]:
```
#another way of accessing column
df['day'] #df.day (both are same)
```

[ ]:
```
0    1/1/2017
1    1/2/2017
2    1/3/2017
3    1/4/2017
4    1/5/2017
5    1/6/2017
Name: day, dtype: object
```

[ ]:
```
#get 2 or more columns
df[['day', 'event']]
```

[ ]:
```
        day  event
0  1/1/2017   Rain
1  1/2/2017  Sunny
2  1/3/2017   Snow
3  1/4/2017   Snow
4  1/5/2017   Rain
5  1/6/2017  Sunny
```

[ ]:
```
#get all temperatures
df['temperature']
```

[ ]:
```
0    32
1    35
2    28
3    24
4    32
5    31
Name: temperature, dtype: int64
```

[ ]:
```
#print max temperature
df['temperature'].max()
```

[ ]:
```
35
```

[ ]:
```
#print max temperature
df['temperature'].min()
```

[ ]:
```
24
```

[ ]:
```
#print max temperature
df['temperature'].describe()
```

[ ]:
```
count     6.000000
mean     30.333333
std       3.829708
min      24.000000
25%      28.750000
50%      31.500000
75%      32.000000
max      35.000000
Name: temperature, dtype: float64
```

[ ]:
```
# select rows which has maximum temperature
df[df.temperature == df.temperature.max()]
```

This will give me max. temp.

This will give me the row no. of that max. temp.

[ ]:
```
        day  temperature  windspeed  event
1  1/2/2017           35          7  Sunny
```

[ ]:
```
#select only day column which has maximum temperature
df.day[df.temperature == df.temperature.max()]
```

[ ]:
```
1    1/2/2017
Name: day, dtype: object
```

[ ]: