DOUGLAS COLLEGE

FUNDAMENTALS OF MACHINE LEARNING IN DATA SCIENCE

CSIS 3290

PANDAS LIBRARY

FATEMEH AHMADI

Series and Dataframes

- Both **DataFrame** and **series** are the two main data structure of pandas library. Series in pandas contains a single list which can store heterogeneous type of data, because of this, series is also considered as a **1-dimensional data structure**.
- On the other hand, DataFrame is a 2-dimensional data structure which contains multiple lists of heterogeneous type of data. DataFrame can contain multiple series or it can be considered as a collection of series.
- When we analyze a series, each value can be considered as a separate row of a single column, whereas when we analyze a DataFrame, we have multiple columns and multiple rows.

https://www.includehelp.com/python/what-is-the-difference-between-a-pandas-series-and-a-dataframe.aspx

Series and Dataframes

```
a    1
b    A
c    *
dtype: object
```

```
Data Frame:

   NickNames          States          Delicacies  Rating

0   Green City        Gandhi Nagar      Pizza        4.5

1   Golden City        Amritsar           Kulcha       4

2    Yoga City          Rishikesh          Samosa     4.6
```

Series

```
In [1]: import pandas as pd
        series1=pd.Series([1,2,3,4,5], index=['row1','row2','row3','row4','row5'])
        series1
```

```
Out[1]: row1    1
        row2    2
        row3    3
        row4    4
        row5    5
        dtype: int64
```

```
In [ ]:
```

```
In [7]: series1.values
```

```
Out[7]: array([1, 2, 3, 4, 5], dtype=int64)
```

```
In [8]: series1.index
```

```
Out[8]: Index(['row1', 'row2', 'row3', 'row4', 'row5'], dtype='object')
```

```
In [9]: series1.row3
```

```
Out[9]: 3
```

```
In [10]: series1['row2']
```

```
Out[10]: 2
```

Series

```
In [13]: series1.index=['a','b','c','d','e']
```

Rewriting the index

```
In [14]: series1

Out[14]: a    1
         b    2
         c    3
         d    4
         e    5
         dtype: int64

In [ ]:
```

Dataframe

```
In [23]: df1=pd.DataFrame(array1,index=['row1','row2','row3','row4'],columns=['col1','col2','col3','col4'])

In [24]: array1=np.array([[1,5,9,13],[2,6,10,19],[3,7,11,5],[4,8,12,16]])

In [25]: df1=pd.DataFrame(array1,index=['row1','row2','row3','row4'],columns=['col1','col2','col3','col4'])

In [26]: df1
```

Out[26]:

	col1	col2	col3	col4
row1	1	5	9	13
row2	2	6	10	19
row3	3	7	11	5
row4	4	8	12	16

```
In [ ]:
```

Dataframe with Dictionary

```
In [30]: dic1={'col1':[1,2,3,4],'col2':[5,6,7,8],'col3':[9,10,11,12],'col4':[13,14,16,16]}

In [31]: df2=pd.DataFrame(dic1,index=['row1','row2','row3','row4'],columns=['col1','col2','col3','col4'])

In [32]: df2
```

Out[32]:

	col1	col2	col3	col4
row1	1	5	9	13
row2	2	6	10	14
row3	3	7	11	16
row4	4	8	12	16

```
In [33]: df2.index
Out[33]: Index(['row1', 'row2', 'row3', 'row4'], dtype='object')

In [34]: df2.columns
Out[34]: Index(['col1', 'col2', 'col3', 'col4'], dtype='object')

In [35]: df2.values
Out[35]: array([[ 1,  5,  9, 13],
                [ 2,  6, 10, 14],
                [ 3,  7, 11, 16],
                [ 4,  8, 12, 16]], dtype=int64)

In [ ]:
```

Dataframe with Dictionary

```
In [40]: df2.loc['row1'][:]

Out[40]: col1     1
         col2     5
         col3     9
         col4    13
         Name: row1, dtype: int64

In [41]: df2.loc['row1']['col2']

Out[41]: 5

In [42]: df2.iloc[0][:]

Out[42]: col1     1
         col2     5
         col3     9
         col4    13
         Name: row1, dtype: int64

In [43]: df2.iloc[0][1]

Out[43]: 5

In [ ]:
```

Dataframes

In [44]:
```
df2.rename(columns={'col4':'column4'})
```

Out[44]:

	col1	col2	col3	column4
row1	1	5	9	13
row2	2	6	10	14
row3	3	7	11	16
row4	4	8	12	16

In [45]:
```
df2.replace({1:10})
```

Out[45]:

	col1	col2	col3	col4
row1	10	5	9	13
row2	2	6	10	14
row3	3	7	11	16
row4	4	8	12	16

Dataframes – Sorting – Head and Tail

```
In [53]: df2.sort_index(axis=1,ascending=False)
```

Out[53]:

	col4	col3	col2	col1
row1	13	9	5	1.000000
row2	14	10	6	2.000000
row3	16	11	7	3.000000
row4	16	12	8	4.000000

```
In [54]: df2.sort_values(by='col1',ascending=False)
```

Out[54]:

	col1	col2	col3	col4
row4	4.000000	8	12	16
row3	3.000000	7	11	16
row2	2.000000	6	10	14
row1	1.000000	5	9	13

```
In [ ]:
```

```
In [ ]:
```

```
In [55]: df2.head(2)
```

Out[55]:

	col1	col2	col3	col4
row1	1.000000	5	9	13
row2	2.000000	6	10	14

```
In [56]: df2.tail(2)
```

Out[56]:

	col1	col2	col3	col4
row3	3.000000	7	11	16
row4	4.000000	8	12	16

```
In [ ]:
```

Importing data

In [57]:
```
data1=pd.read_csv('F:/00-Douglas College/1- Semester 1/3- Machine Learning in Data Science(3290)/Slides/ozone1.csv')
```

```
C:\Users\Paris\AppData\Local\Temp\ipykernel_1532\1179738982.py:1: DtypeWarning: Columns (17) have mixed types. Specify dtype option on import or set low_memory=False.
  data1=pd.read_csv('F:/00-Douglas College/1- Semester 1/3- Machine Learning in Data Science(3290)/Slides/ozone1.csv')
```

In [58]:
```
data1.head()
```

Out[58]:

	State Code	County Code	Site Num	Parameter Code	POC	Latitude	Longitude	Datum	Parameter Name	Date Local	...	Units of Measure	MDL	Uncertainty	Qualifier	Method Type	Method Code	Meth
0	1	3	10	44201	1	30.497478	-87.880258	NAD83	Ozone	2014-03-01	...	Parts per million	0.005	NaN	NaN	FEM	47	INSTRU - ULTRA
1	1	3	10	44201	1	30.497478	-87.880258	NAD83	Ozone	2014-03-01	...	Parts per million	0.005	NaN	NaN	FEM	47	INSTRU - ULTRA
2	1	3	10	44201	1	30.497478	-87.880258	NAD83	Ozone	2014-03-01	...	Parts per million	0.005	NaN	NaN	FEM	47	INSTRU - ULTRA
3	1	3	10	44201	1	30.497478	-87.880258	NAD83	Ozone	2014-03-01	...	Parts per million	0.005	NaN	NaN	FEM	47	INSTRU - ULTRA
4	1	3	10	44201	1	30.497478	-87.880258	NAD83	Ozone	2014-03-01	...	Parts per million	0.005	NaN	NaN	FEM	47	INSTRU - ULTRA

5 rows × 24 columns

In [ ]: