Pandas

Pandas stand for Panel Data and it is the core library for data manupulation and data analysis.

It consists of single and multi dimentional data structures for data manupulation.

Pandas Data Structure

1. Single Dimentional is called as Series Object

2. Multi-dimentional is called as Data Frame

Series Object

Series Object is one dimentional labelled object.

In [1]:

```python
import pandas as pd
s1 = pd.Series([1,2,3,4,5])
s1
```

Out[1]:

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

In [2]:

```python
# In the above output 0-1 is teh index number and 1-5 are the value stores at corrsponding indices
```

In [3]:

```python
type(s1)
```

Out[3]:

```
pandas.core.series.Series
```

Changing Index

In [4]:

```python
import pandas as pd #invoking pandas library is not required every time for a new set of codes.
s1 = pd.Series([1,2,3,4,5],index = ['a','b','c','d','e'])
s1
```

Out[4]:

```
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

Creating series object fron 'Dictionary'

In [5]:

```python
pd.Series({'a':10,'b':20,'c':30})
```

Out[5]:

```
a    10
b    20
c    30
dtype: int64
```

Changing index position

In [6]:

```python
pd.Series({'a':10,'b':20,'c':30}, index = ['b','c','d','a'])
```

Out[6]:

```
b    20.0
c    30.0
d     NaN
a    10.0
dtype: float64
```

Extracting Individual Elements

In [7]:

```python
#Extracting Single element
```

```
s1 = pd.Series([1,2,3,4,5,6,7,8,9])
s1[3]
```

Out[7]:
```
4
```

In [8]:
```
#Extracting a Sequence of elements

s1[:4] # This will extract first 4 elements, from index 0 to 3.
```

Out[8]:
```
0    1
1    2
2    3
3    4
dtype: int64
```

In [9]:
```
#Extracting Elements from back

s1[-3:] # This will extract the last 3 elements
```

Out[9]:
```
6    7
7    8
8    9
dtype: int64
```

Basic Maths operation on Series

In [10]:
```
# Adding a scaler value to Series Elements

s1 + 5
```

Out[10]:
```
0     6
1     7
2     8
3     9
4    10
5    11
6    12
7    13
8    14
dtype: int64
```

In [11]:
```
# Adding two Series object

s1 = pd.Series([1,2,3,4,5,6,7,8,9])
s2 = pd.Series([10,20,30,40,50,60,70,80,90])
s1 + s2
```

Out[11]:
```
0    11
1    22
2    33
3    44
4    55
5    66
6    77
7    88
8    99
dtype: int64
```

In [12]:
```
# Multiplication of two Series object

s1 * s2
```

Out[12]:
```
0     10
1     40
2     90
3    160
4    250
5    360
6    490
7    640
8    810
dtype: int64
```

In [13]:
```
# Division of two Series object

s1/s2
```

Out[13]:
```
0    0.1
1    0.1
2    0.1
```

```
3    0.1
4    0.1
5    0.1
6    0.1
7    0.1
8    0.1
dtype: float64
```

Data Frames

Data Frame is a two dimentional labelled data structure. It comprises of row and columns.

In [14]:

```python
import pandas as pd
pd.DataFrame({"Name":['Shila','Munni','Dhanno'],"Marks":[75,67,87]})
```

Out[14]:

	Name	Marks
0	Shila	75
1	Munni	67
2	Dhanno	87

Data Frame in-built functions

In [15]:

```python
# head() - 1st five rows
# tail() - last 5 rows
# shape() - number of rows and colums
# describe() - General information about the data set
iris = pd.read_csv('iris.csv')
```

In [16]:

```python
iris.head()
```

Out[16]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [17]:

```python
iris.head(10) #firts 10 rows
```

Out[17]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

In [18]:

```python
iris.tail()
```

Out[18]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica

149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [19]:

```
iris.tail(10)
```

Out[19]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
140	141	6.7	3.1	5.6	2.4	Iris-virginica
141	142	6.9	3.1	5.1	2.3	Iris-virginica
142	143	5.8	2.7	5.1	1.9	Iris-virginica
143	144	6.8	3.2	5.9	2.3	Iris-virginica
144	145	6.7	3.3	5.7	2.5	Iris-virginica
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [20]:

```
iris.shape
```

Out[20]:

```
(150, 6)
```

In [21]:

```
iris.describe()
```

Out[21]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

.iloc[]

In [22]:

```
iris.iloc[0:3,0:2]
```

Out[22]:

	Id	SepalLengthCm
0	1	5.1
1	2	4.9
2	3	4.7

In [23]:

```
iris.iloc[1:3,1:3]
```

Out[23]:

	SepalLengthCm	SepalWidthCm
1	4.9	3.0
2	4.7	3.2

In [24]:

```
iris.iloc[5:11,2:] #rows 5 to 10 and column 2 to last
```

Out[24]:

	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
5	3.9	1.7	0.4	Iris-setosa
6	3.4	1.4	0.3	Iris-setosa

7	3.4	1.5	0.2	Iris-setosa
8	2.9	1.4	0.2	Iris-setosa
9	3.1	1.5	0.1	Iris-setosa
10	3.7	1.5	0.2	Iris-setosa

.loc[]

In [25]:

```
iris.loc[1:5,("SepalWidthCm","Species")]
```

Out[25]:

	SepalWidthCm	Species
1	3.0	Iris-setosa
2	3.2	Iris-setosa
3	3.1	Iris-setosa
4	3.6	Iris-setosa
5	3.9	Iris-setosa

Dropping Columns

In [26]:

```
iris.head()
```

Out[26]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [27]:

```
iris.drop('PetalLengthCm', axis = 1) #axis = 1 is used for columns
```

Out[27]:

	Id	SepalLengthCm	SepalWidthCm	PetalWidthCm	Species
0	1	5.1	3.5	0.2	Iris-setosa
1	2	4.9	3.0	0.2	Iris-setosa
2	3	4.7	3.2	0.2	Iris-setosa
3	4	4.6	3.1	0.2	Iris-setosa
4	5	5.0	3.6	0.2	Iris-setosa
...	...	...	...	...	...
145	146	6.7	3.0	2.3	Iris-virginica
146	147	6.3	2.5	1.9	Iris-virginica
147	148	6.5	3.0	2.0	Iris-virginica
148	149	6.2	3.4	2.3	Iris-virginica
149	150	5.9	3.0	1.8	Iris-virginica

150 rows × 5 columns

In [28]:

```
iris.drop(1,axis = 0) #axis = 0 is used for rows
```

Out[28]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa

...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

149 rows × 6 columns

In [29]:
```
iris.head()
```

Out[29]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Drop more than one rows

In [30]:
```
iris.drop(["SepalLengthCm", "SepalWidthCm", "PetalLengthCm"], axis = 1)
```

Out[30]:

	Id	PetalWidthCm	Species
0	1	0.2	Iris-setosa
1	2	0.2	Iris-setosa
2	3	0.2	Iris-setosa
3	4	0.2	Iris-setosa
4	5	0.2	Iris-setosa
...	...	...	...
145	146	2.3	Iris-virginica
146	147	1.9	Iris-virginica
147	148	2.0	Iris-virginica
148	149	2.3	Iris-virginica
149	150	1.8	Iris-virginica

150 rows × 3 columns

Drop more than one column

In [31]:
```
iris.head()
```

Out[31]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [32]:
```
iris.drop([2,3,4],axis = 0)
```

Out[32]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa

5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

147 rows × 6 columns

Mean, Median, Maximum & Minimum

In [34]:

```
iris.head()
```

Out[34]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [37]:

```
# Minimum

iris.min()
```

Out[37]:

```
Id                          1
SepalLengthCm             4.3
SepalWidthCm              2.0
PetalLengthCm             1.0
PetalWidthCm              0.1
Species           Iris-setosa
dtype: object
```

In [38]:

```
# Maximum

iris.max()
```

Out[38]:

```
Id                             150
SepalLengthCm                  7.9
SepalWidthCm                   4.4
PetalLengthCm                  6.9
PetalWidthCm                   2.5
Species           Iris-virginica
dtype: object
```

In [39]:

```
# Median

iris.median()
```

Out[39]:

```
Id               75.50
SepalLengthCm     5.80
SepalWidthCm      3.00
PetalLengthCm     4.35
PetalWidthCm      1.30
dtype: float64
```

In [46]:

```
# Mean

iris.mean()
```

Out[46]:

```
Id               75.500000
SepalLengthCm     5.843333
SepalWidthCm      3.054000
PetalLengthCm     3.758667
PetalWidthCm      1.198667
dtype: float64
```

Apply Function

In [47]:

```
iris.head()
```

Out[47]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [48]:

```
def half(s): #'half is the user defined function'
    return s*0.5
iris[['SepalLengthCm','SepalWidthCm']].apply(half)
```

Out[48]:

	SepalLengthCm	SepalWidthCm
0	2.55	1.75
1	2.45	1.50
2	2.35	1.60
3	2.30	1.55
4	2.50	1.80
...	...	...
145	3.35	1.50
146	3.15	1.25
147	3.25	1.50
148	3.10	1.70
149	2.95	1.50

150 rows × 2 columns

value_counts() - value of each category in a coulmn

In [50]:

```
iris['Species'].value_counts()
```

Out[50]:

```
Iris-versicolor    50
Iris-virginica     50
Iris-setosa        50
Name: Species, dtype: int64
```

sort_values() - in increasing order

In [51]:

```
iris.head()
```

Out[51]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [54]:

```
view = iris.sort_values(by = 'SepalLengthCm')
```

In [55]:

```
view.head(10)
```

Out[55]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
13	14	4.3	3.0	1.1	0.1	Iris-setosa
42	43	4.4	3.2	1.3	0.2	Iris-setosa
38	39	4.4	3.0	1.3	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
41	42	4.5	2.3	1.3	0.3	Iris-setosa
22	23	4.6	3.6	1.0	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
47	48	4.6	3.2	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa