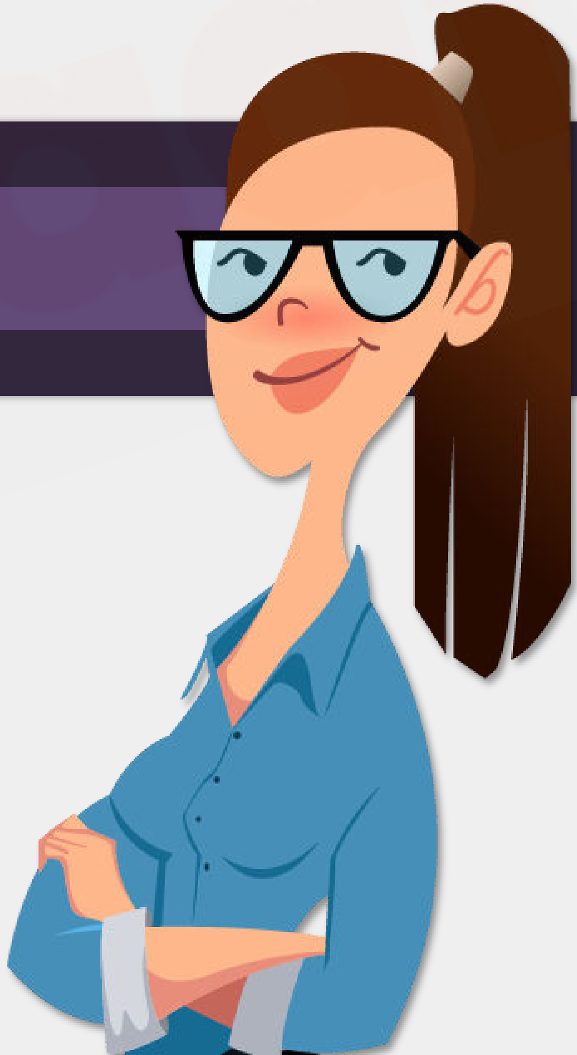
Python Tutorial for Beginners

Introduction to Pandas



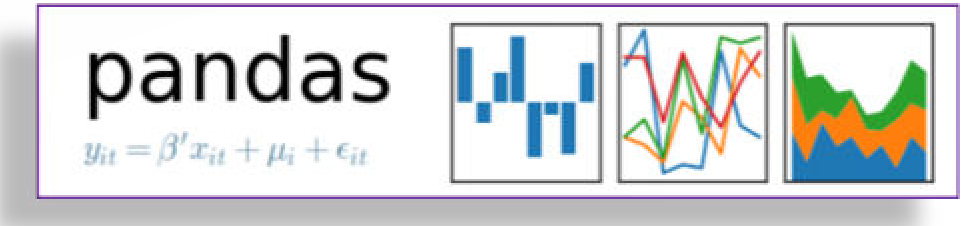
Introduction to Pandas

What is Pandas?

01 Open-source Python library

02 Simple yet powerful and expressive tool

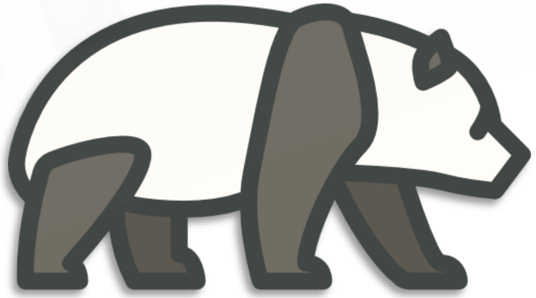
03 Data Manipulation & Analysis

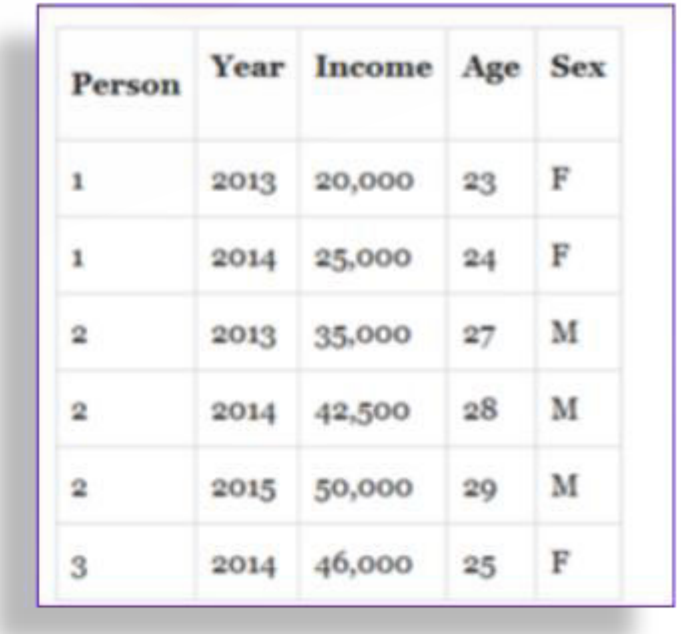


Introduction to Pandas

Where did the name Pandas come from?

- The name Pandas is derived from the word Panel Data
- Panel Data is multi-dimensional data involving measurements over time





Person	Year	Income	Age	Sex
1	2013	20,000	23	F
1	2014	25,000	24	F
2	2013	35,000	27	M
2	2014	42,500	28	M
2	2015	50,000	29	M
3	2014	46,000	25	F

Panel Data

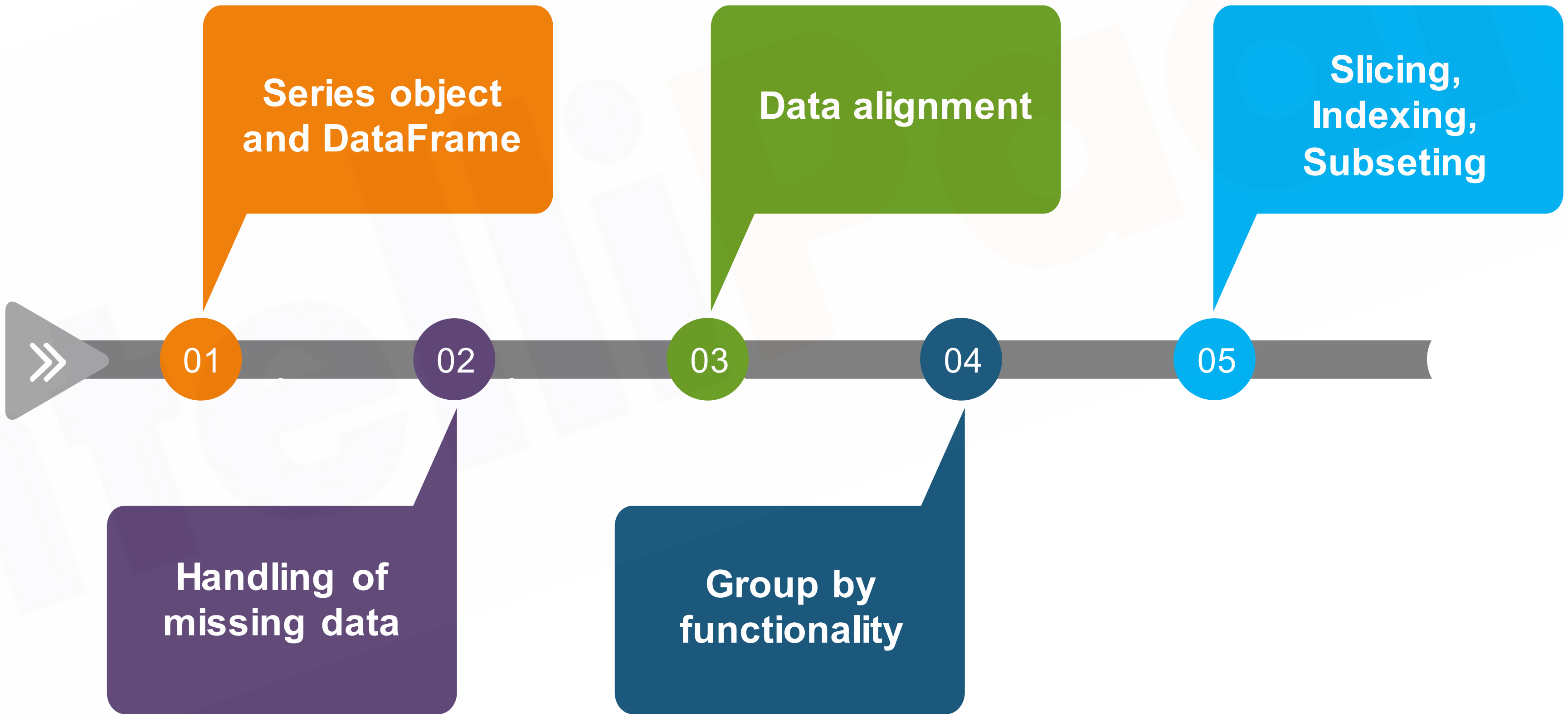
Introduction to Pandas

Who created Pandas?

Created in 2015 by Wes McKinney

Introduction to Pandas

Features of Pandas:



01 Series object and DataFrame

02 Handling of missing data

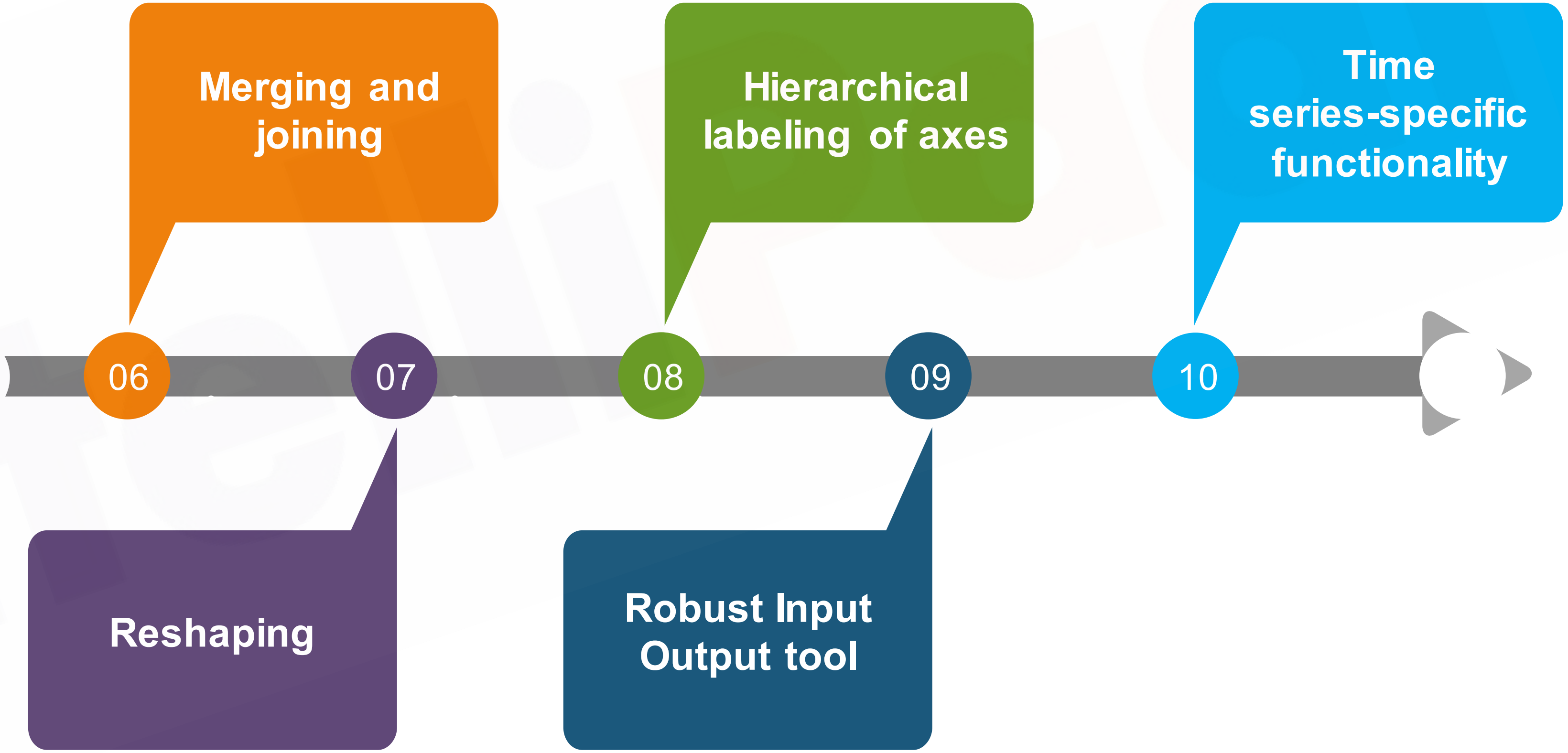
03 Data alignment

04 Group by functionality

05 Slicing, Indexing, Subseting

Introduction to Pandas

Features of Pandas:



06 Merging and joining

07 Reshaping

08 Hierarchical labeling of axes

09 Robust Input Output tool

10 Time series-specific functionality

Pandas vs Numpy

pandas

Pandas performs better than numpy for 500k rows or more.

NumPy

Numpy performs better for 50k rows or less.

Pandas vs Numpy

pandas

Pandas Series Object is more flexible as you can define your own labeled index to index and access elements of an array

NumPy

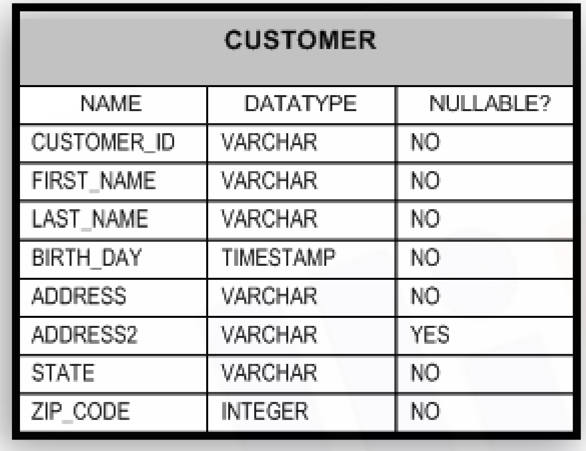
Elements in NumPy arrays are accessed by their default integer position

Introduction to Pandas

How to import Pandas in Python?

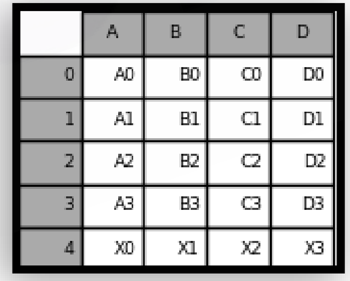
```
import pandas as pd
```

What kind of data does suit Pandas the most?



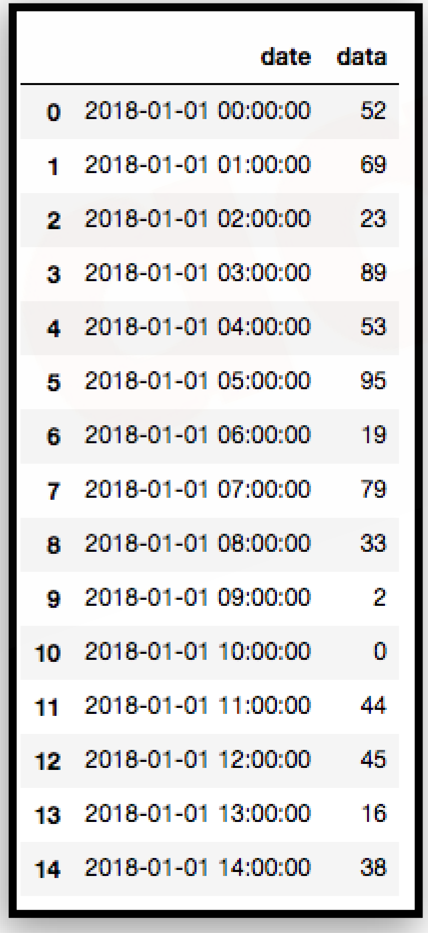
CUSTOMER		
NAME	DATATYPE	NULLABLE?
CUSTOMER_ID	VARCHAR	NO
FIRST_NAME	VARCHAR	NO
LAST_NAME	VARCHAR	NO
BIRTH_DAY	TIMESTAMP	NO
ADDRESS	VARCHAR	NO
ADDRESS2	VARCHAR	YES
STATE	VARCHAR	NO
ZIP_CODE	INTEGER	NO

Tabular data



	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	X0	X1	X2	X3

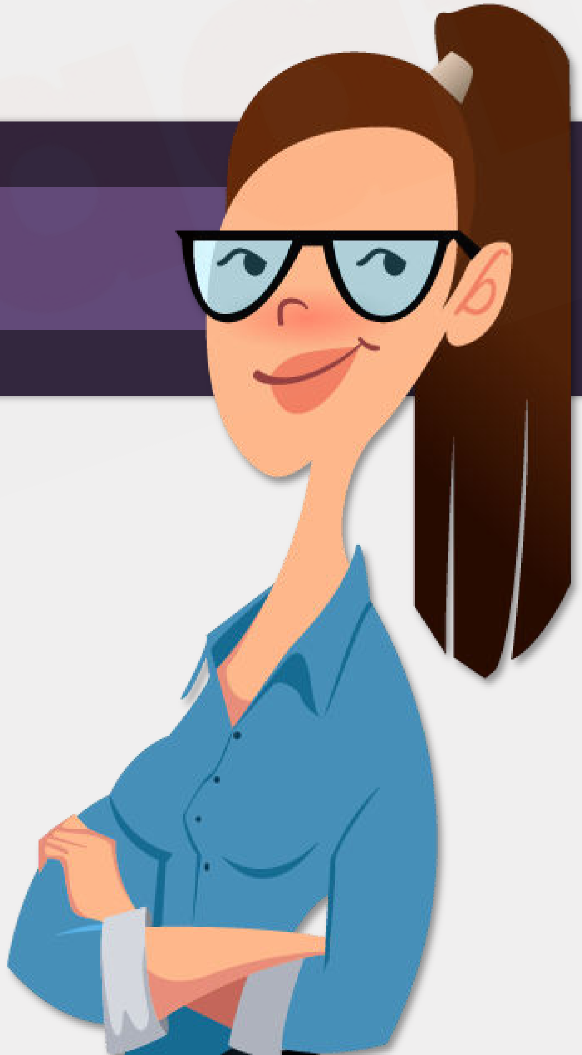
Arbitrary Matrix



	date	data
0	2018-01-01 00:00:00	52
1	2018-01-01 01:00:00	69
2	2018-01-01 02:00:00	23
3	2018-01-01 03:00:00	89
4	2018-01-01 04:00:00	53
5	2018-01-01 05:00:00	95
6	2018-01-01 06:00:00	19
7	2018-01-01 07:00:00	79
8	2018-01-01 08:00:00	33
9	2018-01-01 09:00:00	2
10	2018-01-01 10:00:00	0
11	2018-01-01 11:00:00	44
12	2018-01-01 12:00:00	45
13	2018-01-01 13:00:00	16
14	2018-01-01 14:00:00	38

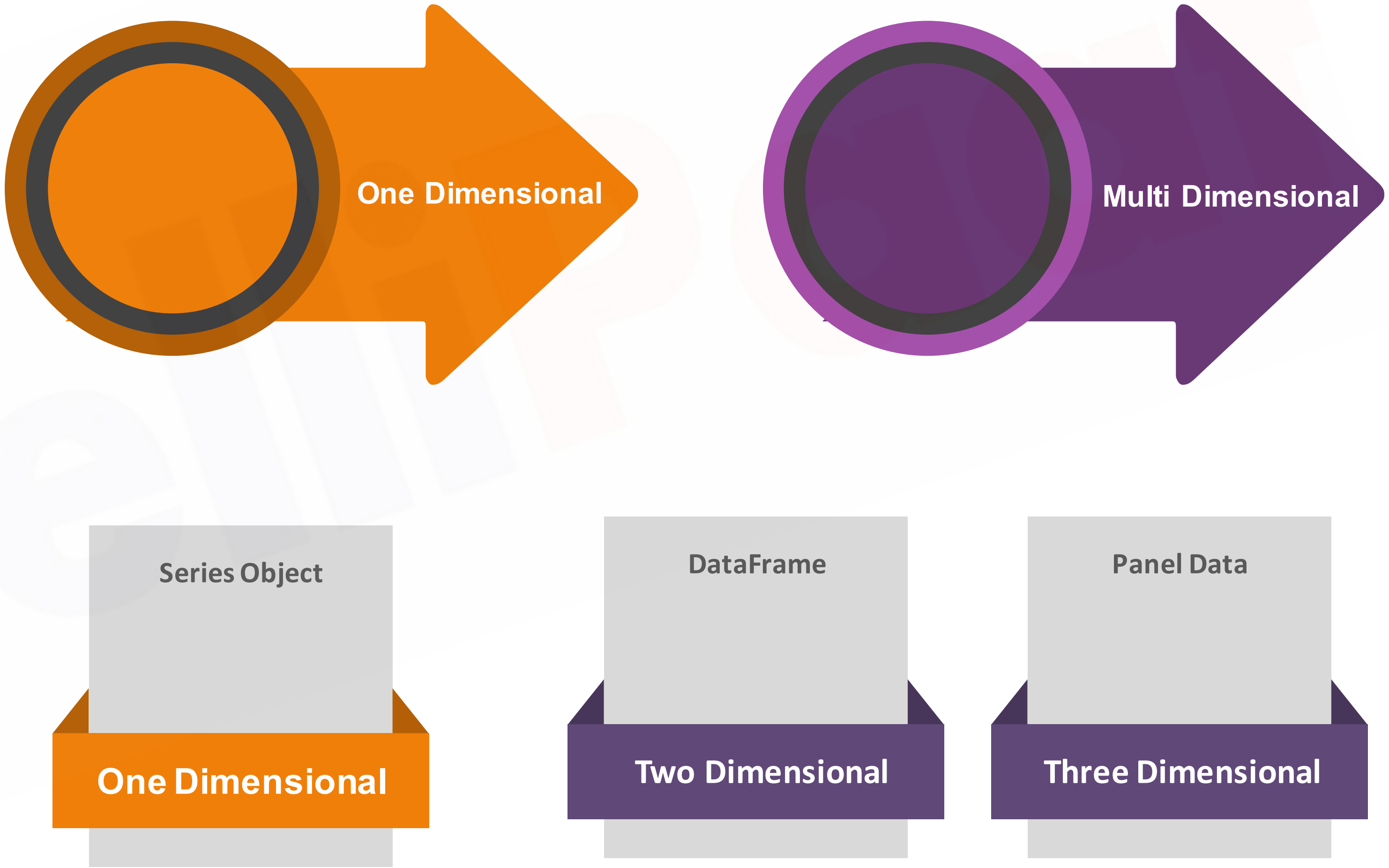
Time Series Data

Data Structure in Pandas



Introduction to Pandas

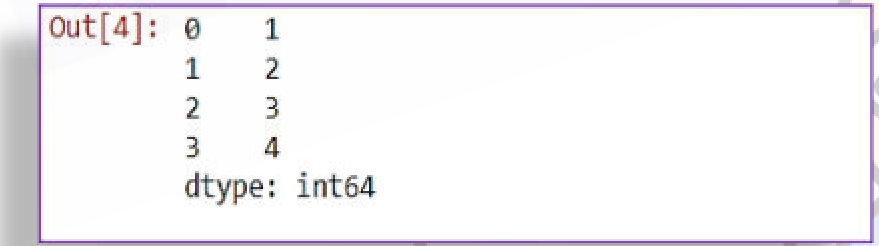
Data-set in Pandas



What is a series object?

- One-dimensional labeled array
- Contains data of similar or mixed types
- Example:

```
data= [1, 2, 3, 4]
series1 = pd.Series(data)
series1
```



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

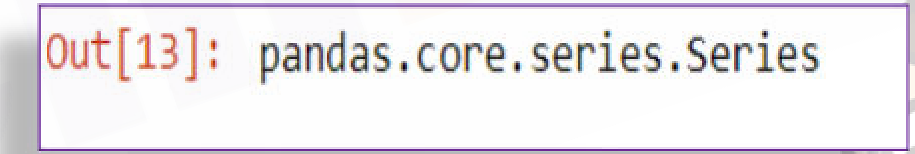
Introduction to Pandas

How to check the type?

Series Object

One Dimensional

```
type(series1)
```



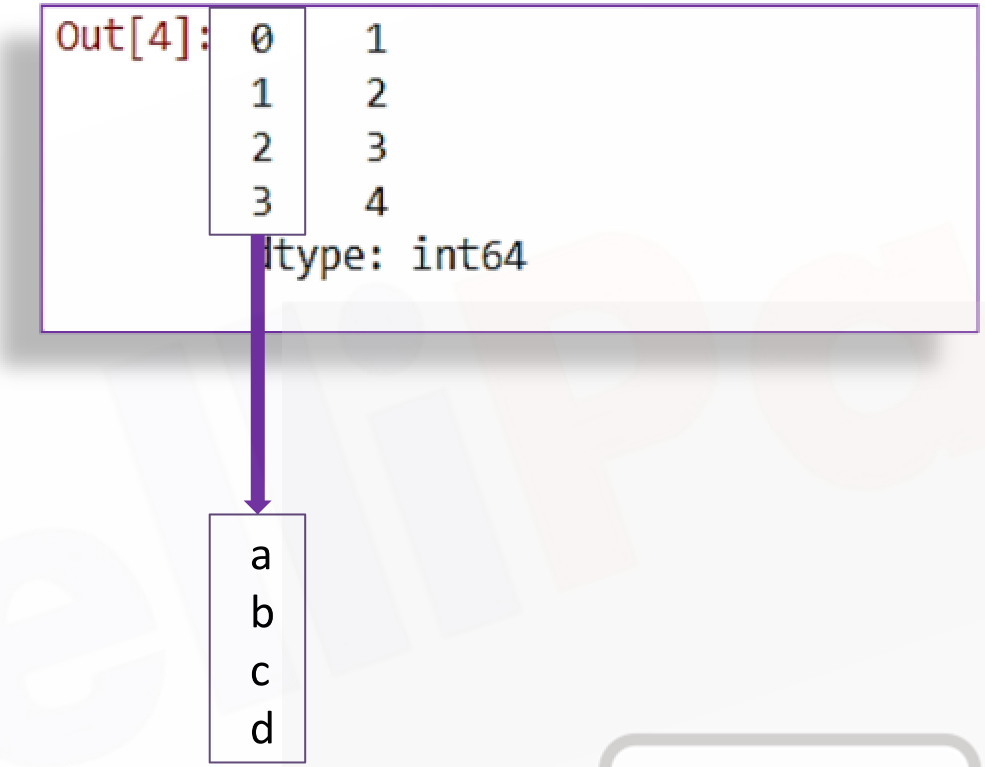
```
Out[13]:  pandas.core.series.Series
```

Introduction to Pandas

How to change the index name?

Series Object

One Dimensional



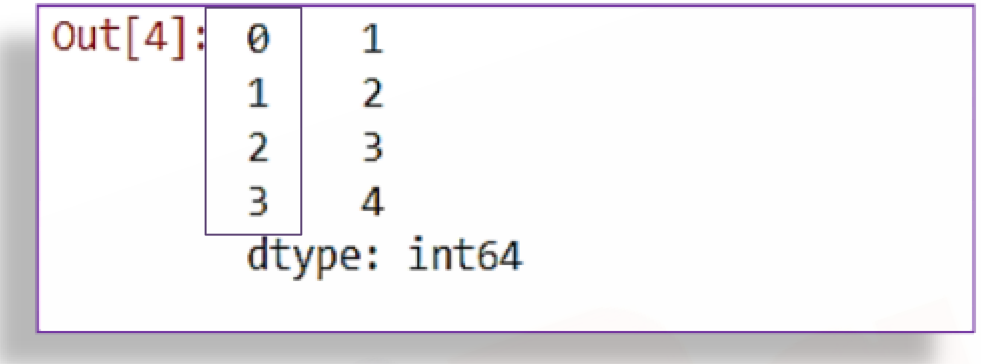
```
Out[4]: 0    1
        1    2
        2    3
        3    4
        dtype: int64
```

One Dimensional

Multi Dimensional

Introduction to Pandas

How to change the index name?

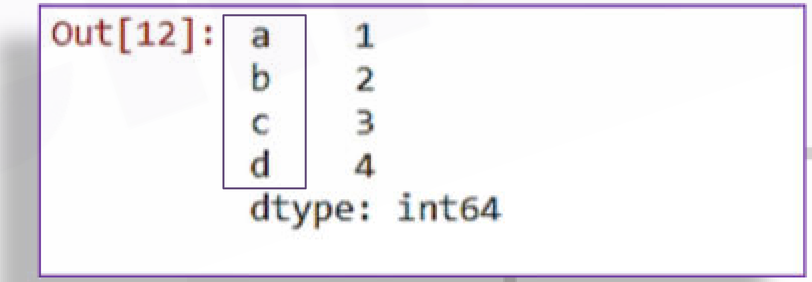


```
Out[4]:  0    1
         1    2
         2    3
         3    4
         dtype: int64
```

Series Object

One Dimensional

```
series1 = pd.Series([1, 2, 3, 4]index=['a', 'b', 'c', 'd’]))

series1
```



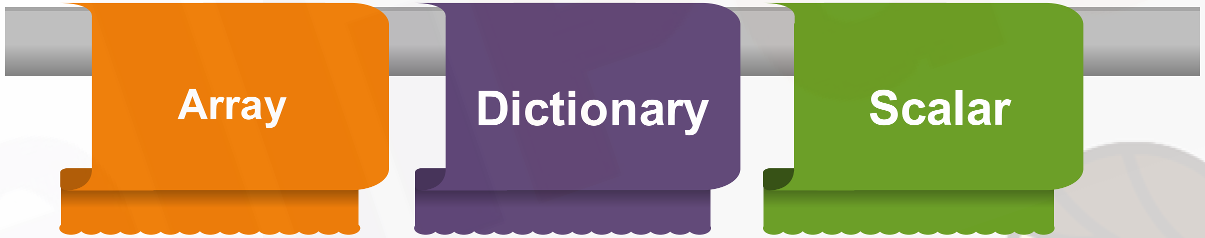
```
Out[12]:  a    1
          b    2
          c    3
          d    4
          dtype: int64
```

Introduction to Pandas

Create different Series Object datatypes

Series Object

One Dimensional



One Dimensional
Multi Dimensional

IntelliPaat

Introduction to Pandas

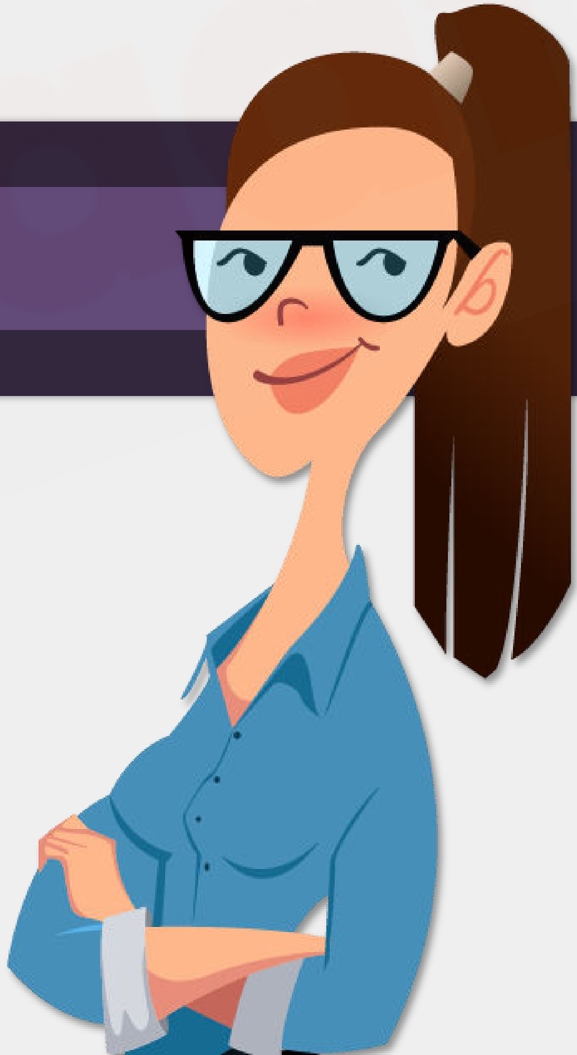
How to create a series object?

Series Object

One Dimensional

```
pd.Series(data)
```

What is a DataFrame?



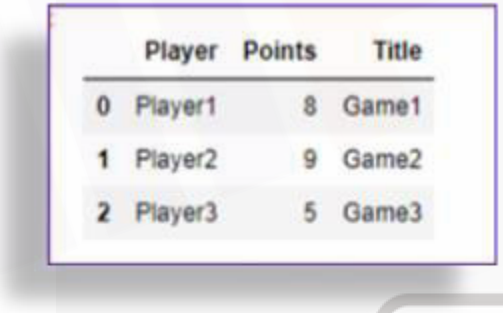
Introduction to Pandas

What is a DataFrame?

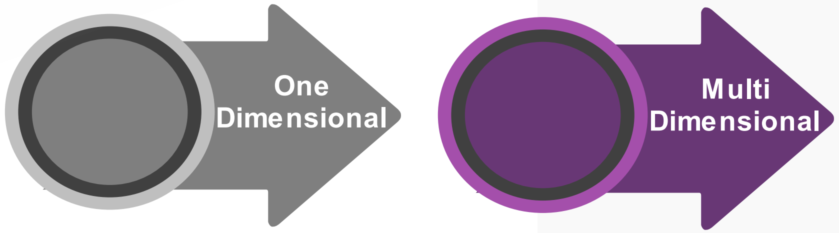
DataFrame

Multi Dimensional

- Two-dimensional labeled data structures with columns of potentially different types
- Example:

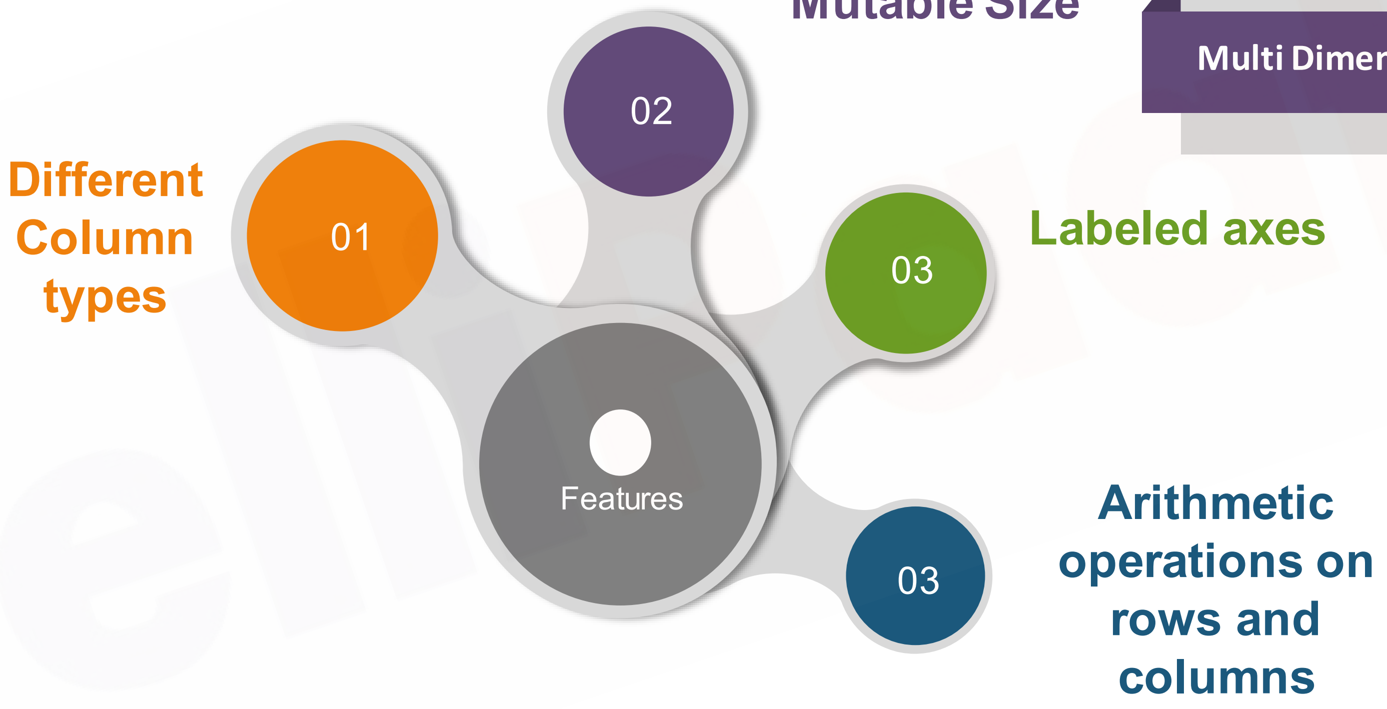


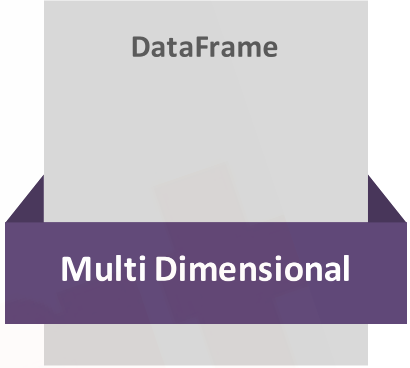
	Player	Points	Title
0	Player1	8	Game1
1	Player2	9	Game2
2	Player3	5	Game3

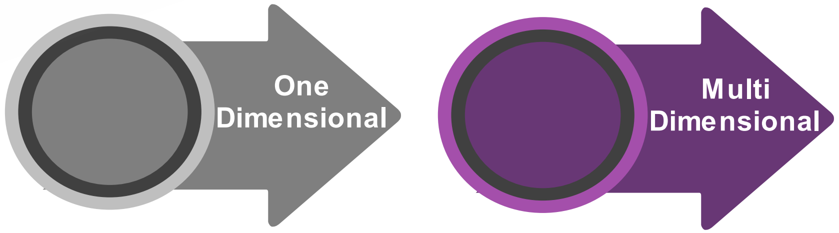


Introduction to Pandas

Features of DataFrame







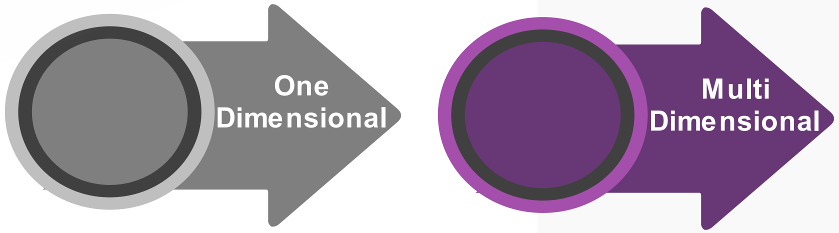
Introduction to Pandas

How to create a DataFrame?

DataFrame

Multi Dimensional

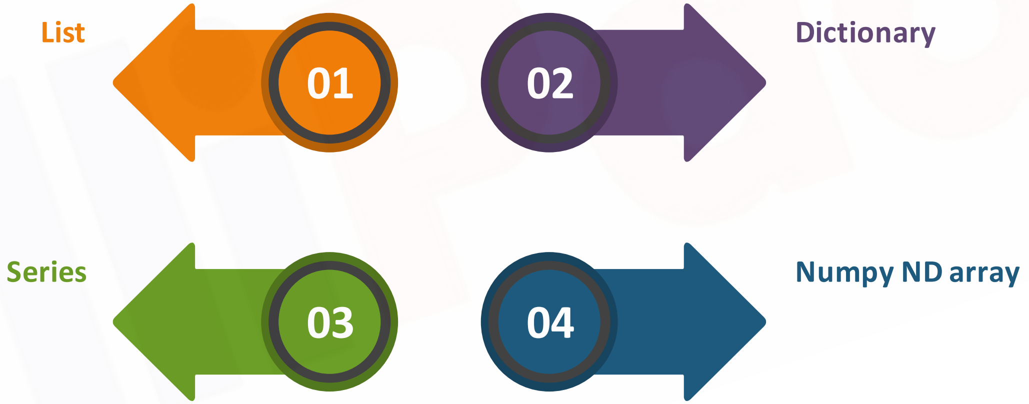
```
pd.DataFrame(data)
```

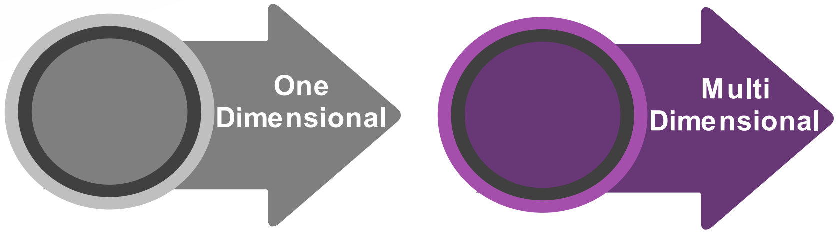


How to create a DataFrame?

DataFrame

Multi Dimensional



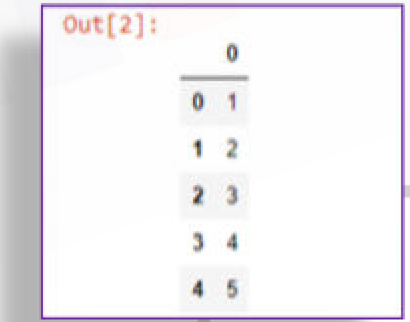


Create a DataFrame from a List

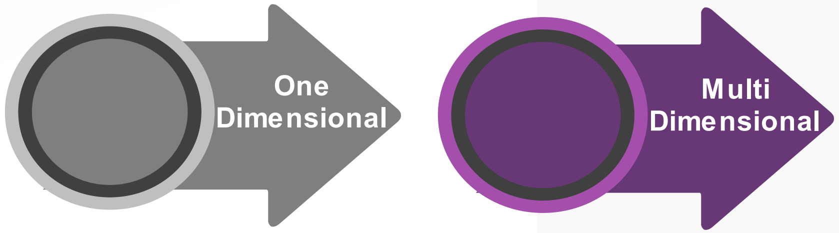
DataFrame

Multi Dimensional

```
data = [1,2,3,4,5]

df = pd.DataFrame(data)

df
```



```
Out[2]:
      0
   0  1
   1  2
   2  3
   3  4
   4  5
```



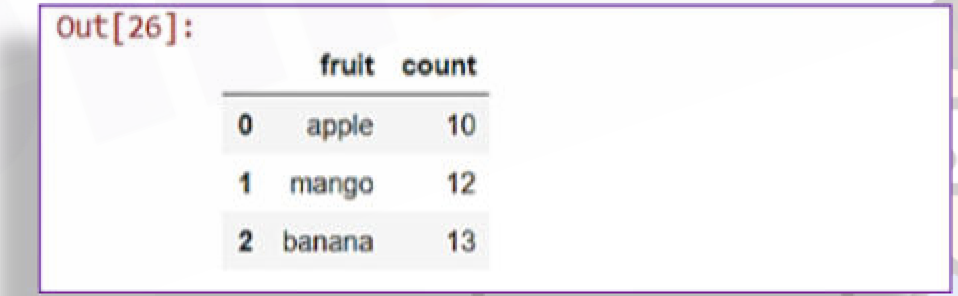
Introduction to Pandas

Create a DataFrame from a Dictionary

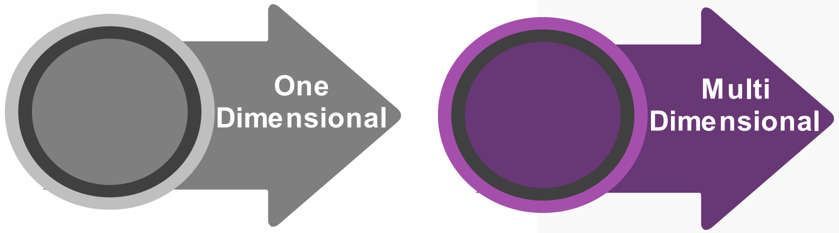
DataFrame

Multi Dimensional

```
dict1 = {'fruit':['apple', 'mango', 'banana'],'count':[10,12,13]}

df = pd.DataFrame(dict1)

df
```



```
Out[26]:
        fruit  count
   0    apple     10
   1    mango     12
   2   banana     13
```

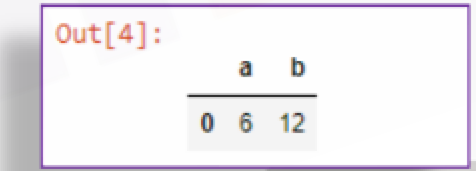


Create a DataFrame from a Series

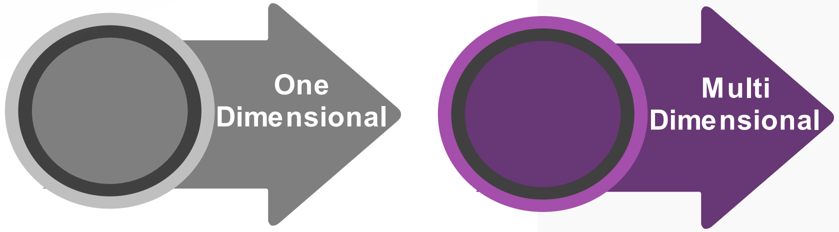
DataFrame

Multi Dimensional

```
data = pd.Series([6,12], index=['a','b'])

df = pd.DataFrame([data])

df
```



```
Out[4]:
      a   b
   0  6  12
```



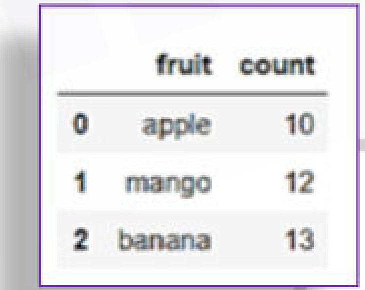
Introduction to Pandas

Create a DataFrame from a numpy ND array

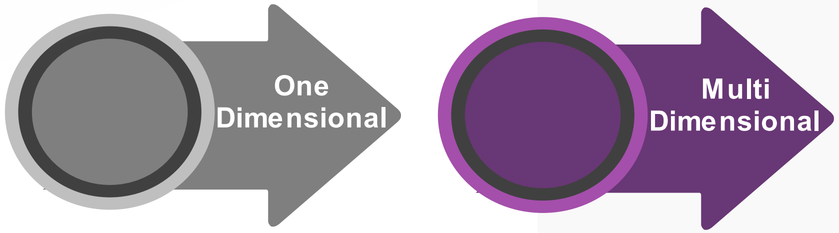
DataFrame

Multi Dimensional

```
import numpy as np

data= np.array([['a','b'], [6,12]])

df = pd.DataFrame({'A':data[:,0],'B':data[:,1]})

df
```



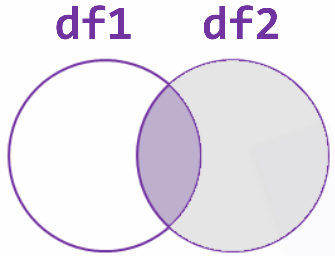
	fruit	count
0	apple	10
1	mango	12
2	banana	13



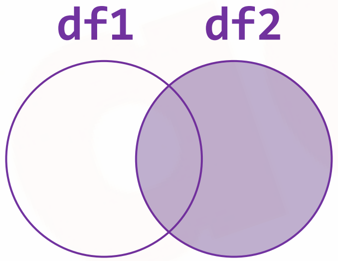
Working with DataFrame

Merge, Join and Concatenate

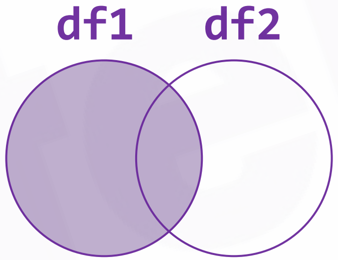
DataFrame for Pandas Merge



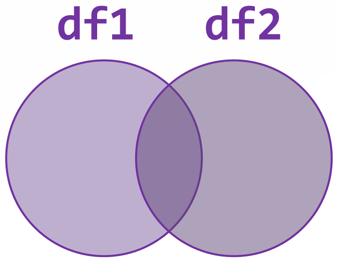
Inner Merge
Inner join



Right Merge
Right Join



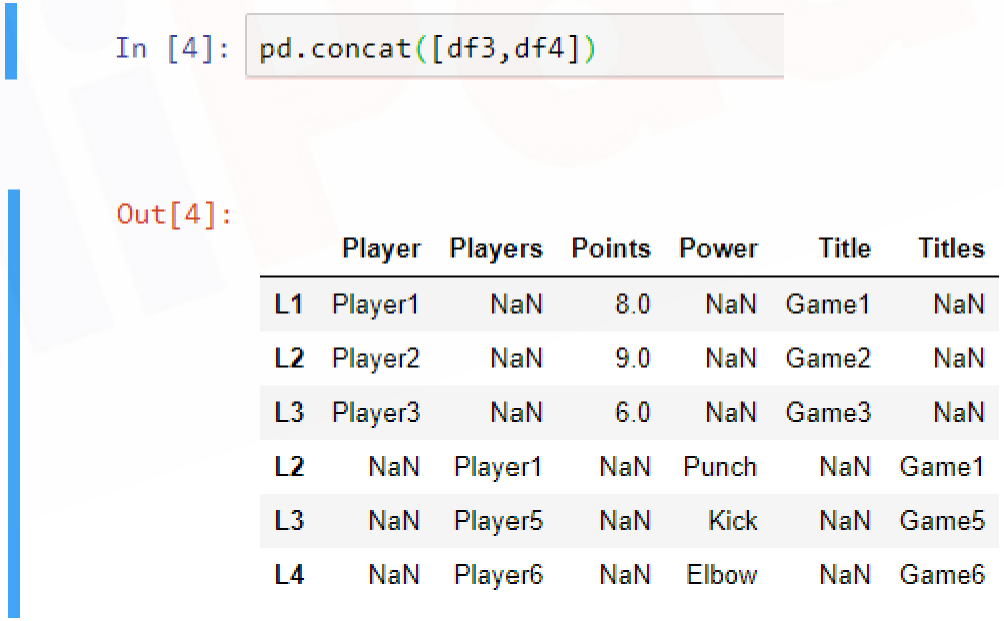
Left Merge
Left Join



Outer Merge
Outer Join

Merge, Join and Concatenate

Concatenate



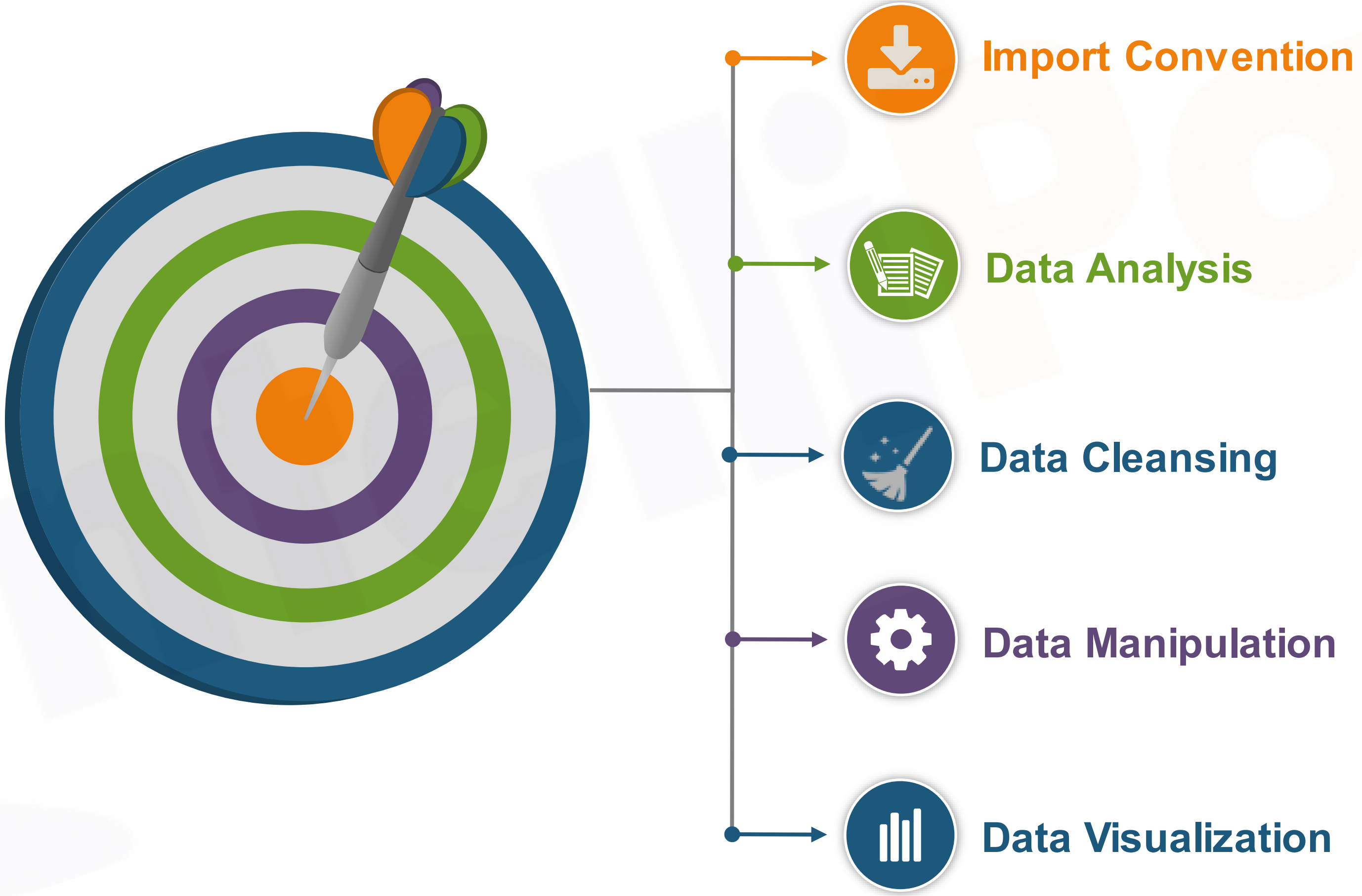
```
In [4]: pd.concat([df3,df4])
```

Out[4]:

	Player	Players	Points	Power	Title	Titles
L1	Player1	NaN	8.0	NaN	Game1	NaN
L2	Player2	NaN	9.0	NaN	Game2	NaN
L3	Player3	NaN	6.0	NaN	Game3	NaN
L2	NaN	Player1	NaN	Punch	NaN	Game1
L3	NaN	Player5	NaN	Kick	NaN	Game5
L4	NaN	Player6	NaN	Elbow	NaN	Game6

Understanding Pandas Operations with example

Hands-on Demonstration



Import Convention

Data Analysis

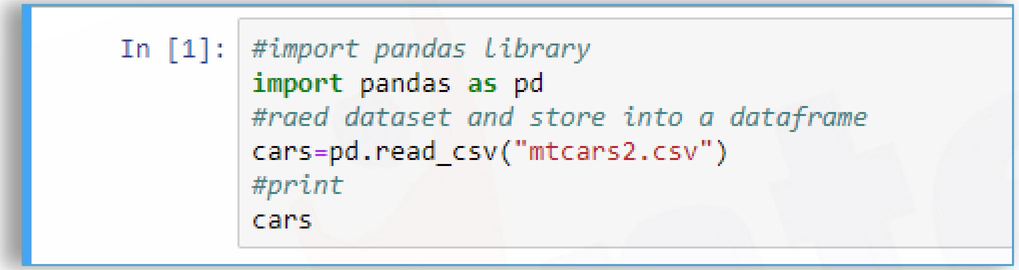
Data Cleansing

Data Manipulation

Data Visualization

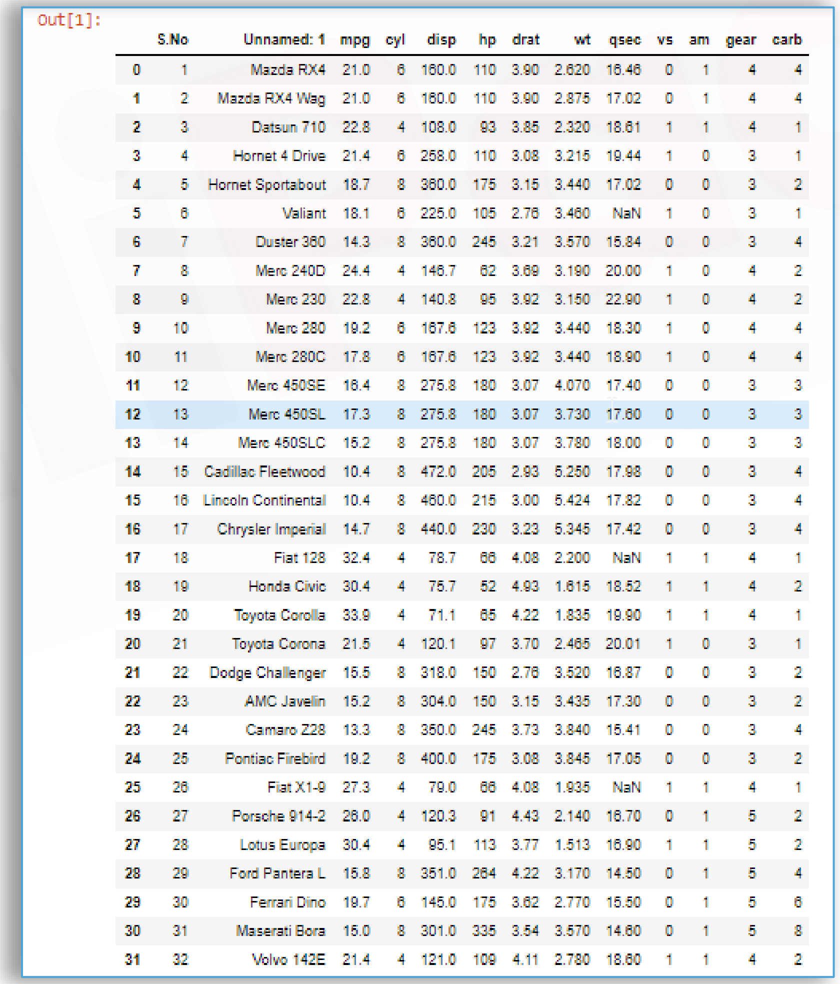
Importing Data set

Input



```
In [1]: #import pandas library
        import pandas as pd
        #raed dataset and store into a dataframe
        cars=pd.read_csv("mtcars2.csv")
        #print
        cars
```

Output



Out[1]:

	S.No	Unnamed: 1	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	NaN	1	0	3	1
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
10	11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
11	12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
12	13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
13	14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
14	15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
15	16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
16	17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
17	18	Fiat 128	32.4	4	78.7	66	4.08	2.200	NaN	1	1	4	1
18	19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
19	20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
20	21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
21	22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
25	26	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	NaN	1	1	4	1
26	27	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
27	28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
28	29	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
29	30	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
30	31	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
31	32	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Analyzing Data set

Functions

df.head()

df.tail()

df.shape

df.info()

df.mean()

df.std()

df.max()

df.min()

df.median()

df.count()

df.describe()

Analyzing Data set

Functions

df.rename()

df.astype()

df.drop()

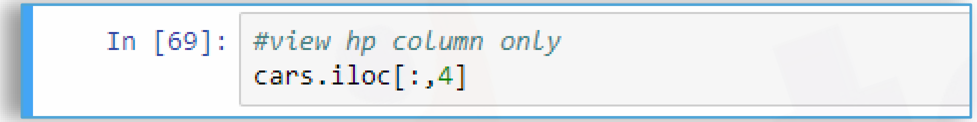
df.fillna()

df.corr()

df.corr()

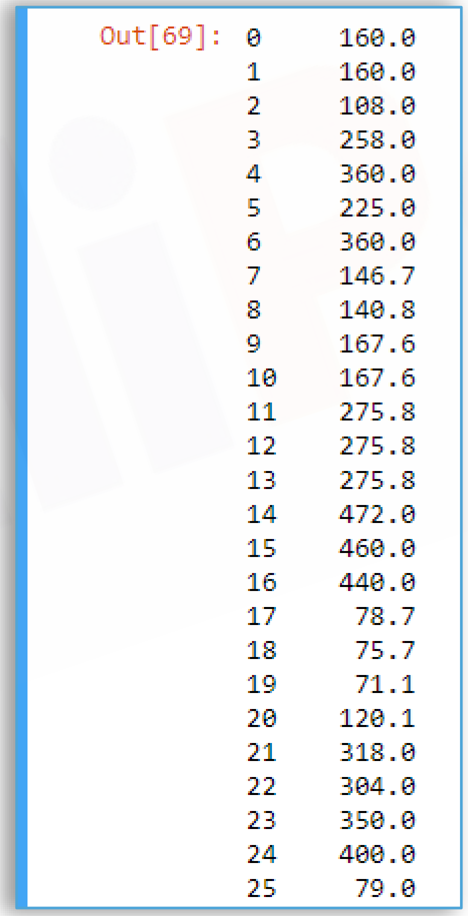
Manipulating Data set: Indexing by position

Input



```
In [69]: #view hp column only
         cars.iloc[:,4]
```

Output

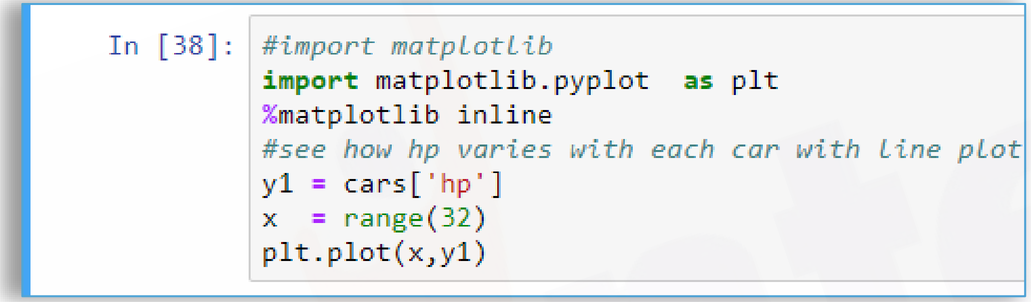


```
Out[69]: 0      160.0
         1      160.0
         2      108.0
         3      258.0
         4      360.0
         5      225.0
         6      360.0
         7      146.7
         8      140.8
         9      167.6
         10     167.6
         11     275.8
         12     275.8
         13     275.8
         14     472.0
         15     460.0
         16     440.0
         17      78.7
         18      75.7
         19      71.1
         20     120.1
         21     318.0
         22     304.0
         23     350.0
         24     400.0
         25      79.0
```

Visualizing Data set

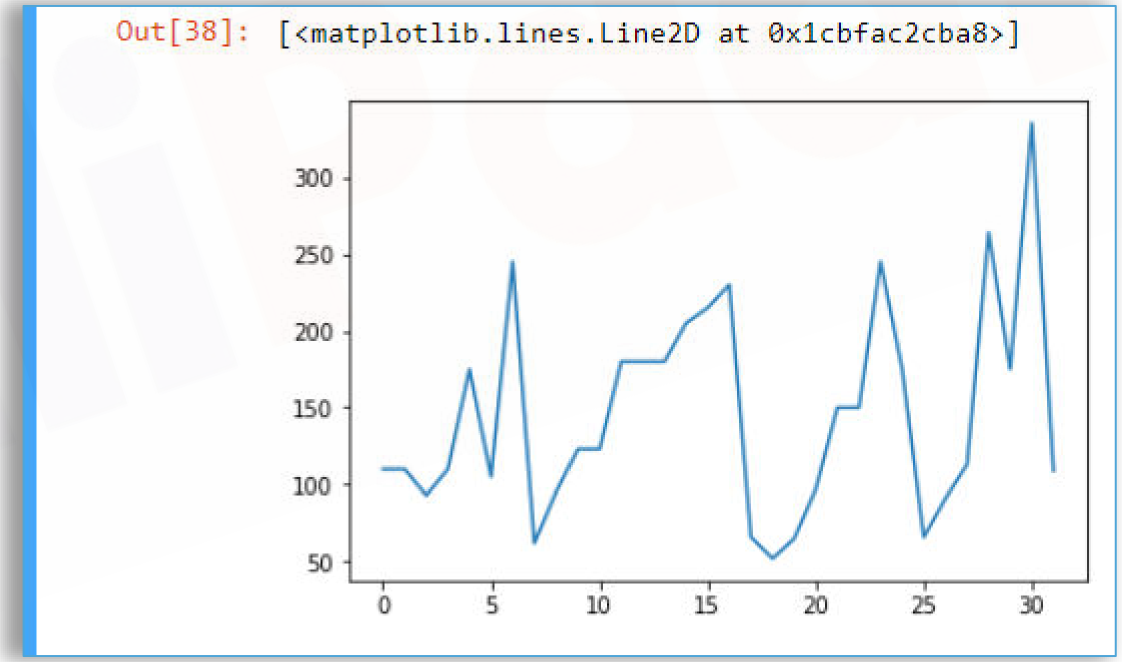
Visualizing Data set: Line Plot

Input



```
In [38]: #import matplotlib
         import matplotlib.pyplot  as plt
         %matplotlib inline
         #see how hp varies with each car with line plot
         y1 = cars['hp']
         x  = range(32)
         plt.plot(x,y1)
```

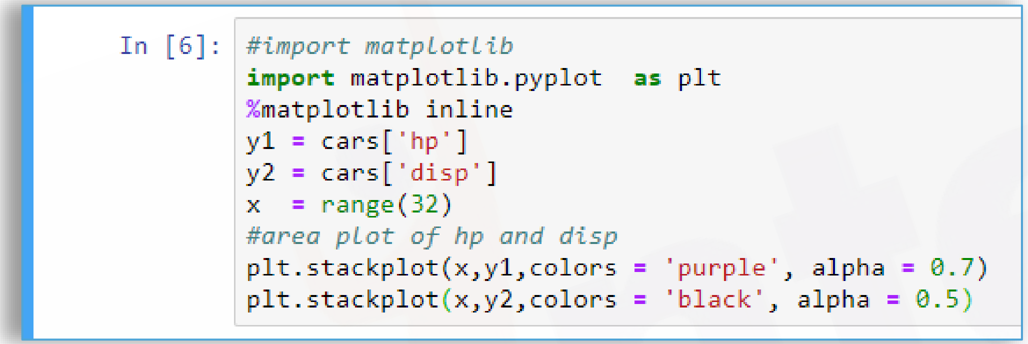
Output



```
Out[38]: [<matplotlib.lines.Line2D at 0x1cbfac2cba8>]
```

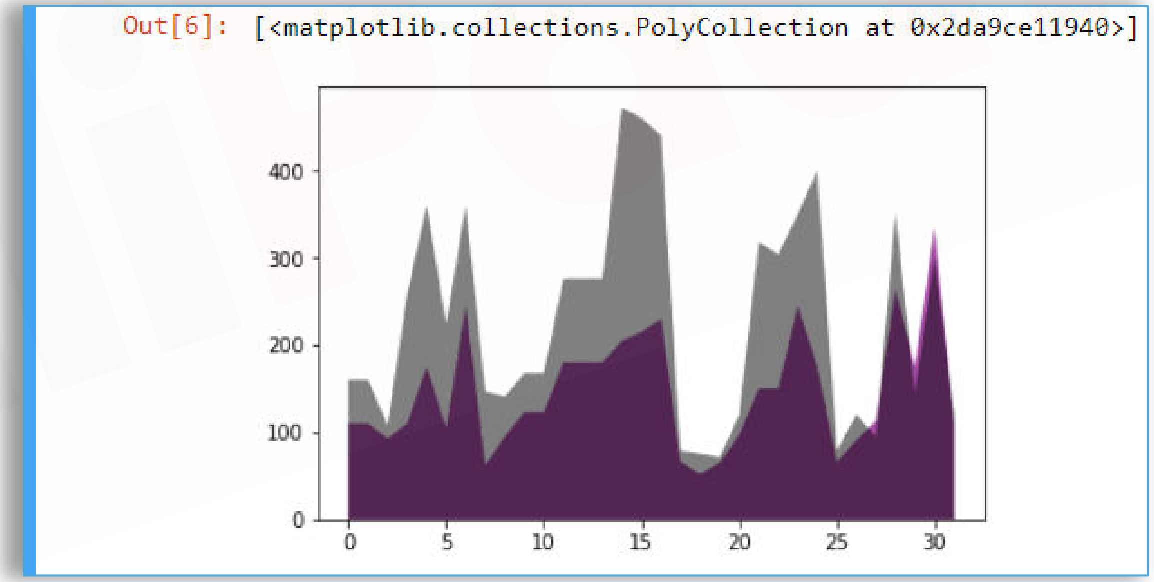
Visualizing Data set: Area plot

Input



```
In [6]: #import matplotlib
        import matplotlib.pyplot  as plt
        %matplotlib inline
        y1 = cars['hp']
        y2 = cars['disp']
        x  = range(32)
        #area plot of hp and disp
        plt.stackplot(x,y1,colors = 'purple', alpha = 0.7)
        plt.stackplot(x,y2,colors = 'black', alpha = 0.5)
```

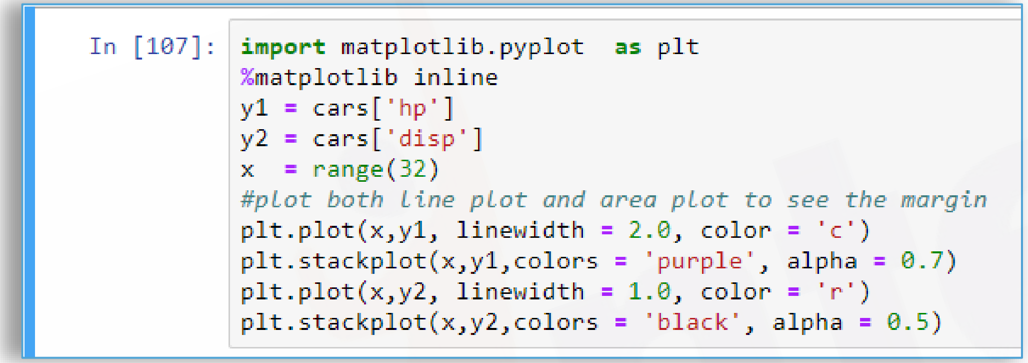
Output



```
Out[6]: [<matplotlib.collections.PolyCollection at 0x2da9ce11940>]
```

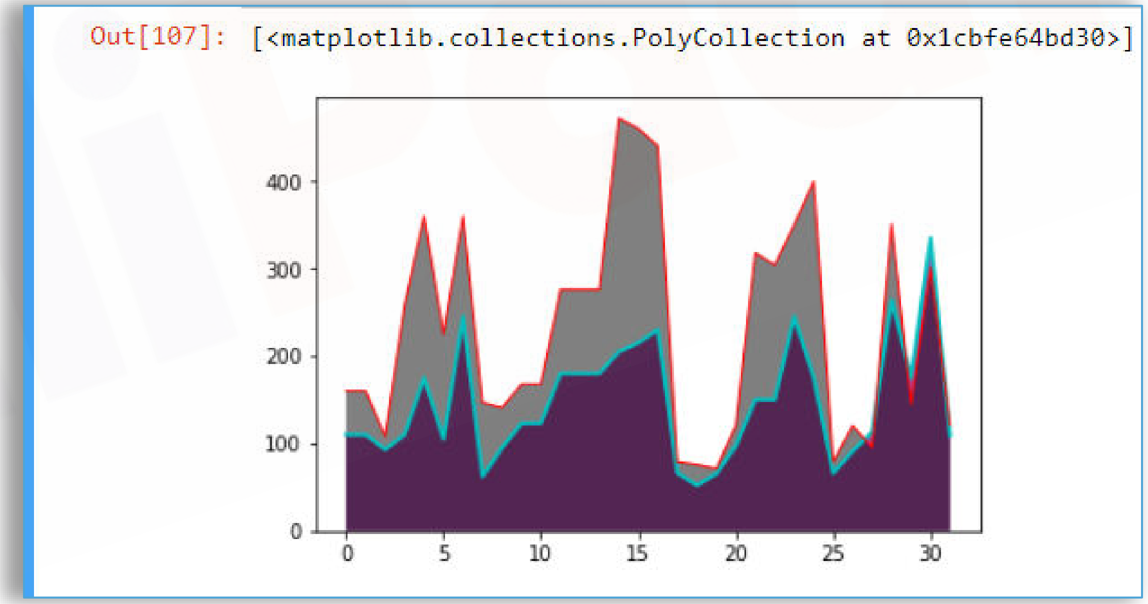
Visualizing Data set: Area and Line Plot

Input



```
In [107]: import matplotlib.pyplot  as plt
          %matplotlib inline
          y1 = cars['hp']
          y2 = cars['disp']
          x  = range(32)
          #plot both line plot and area plot to see the margin
          plt.plot(x,y1, linewidth = 2.0, color = 'c')
          plt.stackplot(x,y1,colors = 'purple', alpha = 0.7)
          plt.plot(x,y2, linewidth = 1.0, color = 'r')
          plt.stackplot(x,y2,colors = 'black', alpha = 0.5)
```

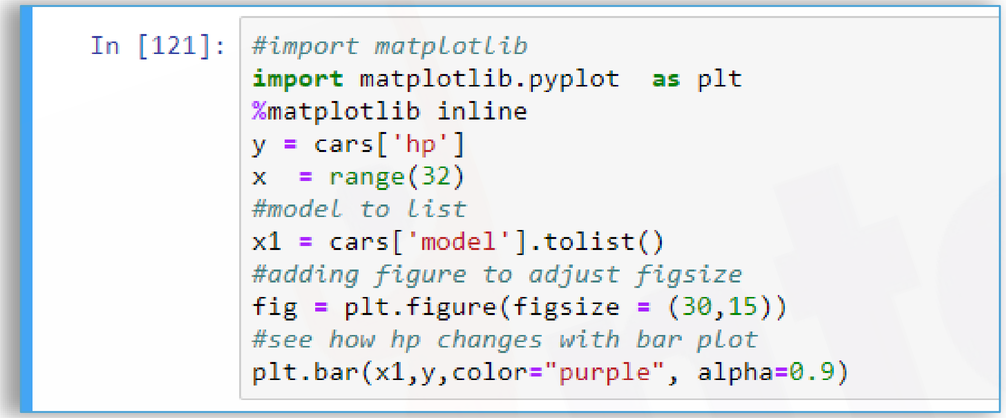
Output



```
Out[107]: [<matplotlib.collections.PolyCollection at 0x1cbfe64bd30>]
```

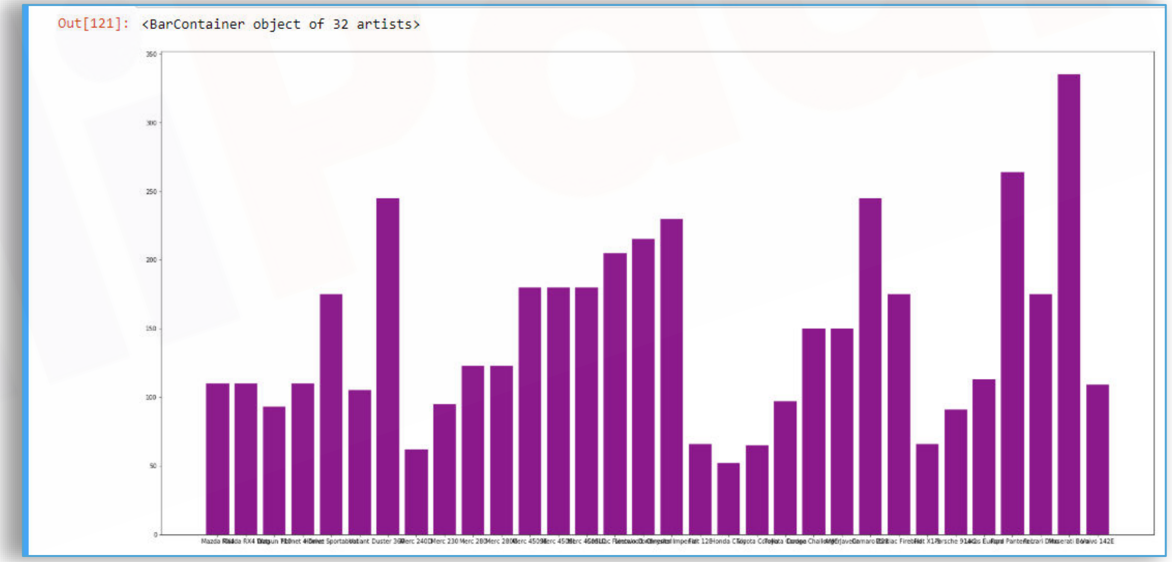
Visualizing Data set: Bar Plot

Input



```
In [121]: #import matplotlib
          import matplotlib.pyplot  as plt
          %matplotlib inline
          y = cars['hp']
          x  = range(32)
          #model to list
          x1 = cars['model'].tolist()
          #adding figure to adjust figsize
          fig = plt.figure(figsize = (30,15))
          #see how hp changes with bar plot
          plt.bar(x1,y,color="purple", alpha=0.9)
```

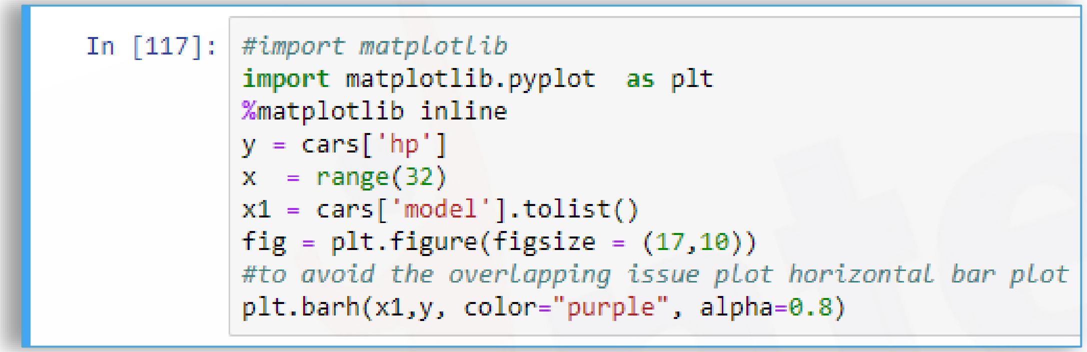
Output



```
Out[121]: <BarContainer object of 32 artists>
```

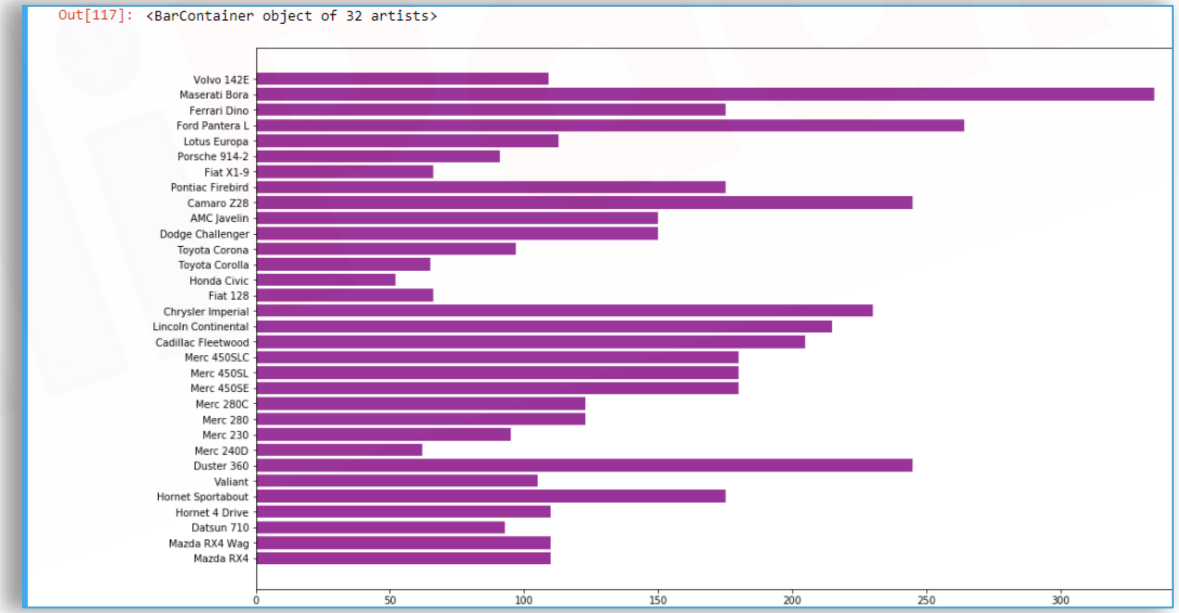
Visualizing Data set: Horizontal Bar Plot

Input



```
In [117]: #import matplotlib
          import matplotlib.pyplot  as plt
          %matplotlib inline
          y = cars['hp']
          x  = range(32)
          x1 = cars['model'].tolist()
          fig = plt.figure(figsize = (17,10))
          #to avoid the overlapping issue plot horizontal bar plot
          plt.barh(x1,y, color="purple", alpha=0.8)
```

Output



```
Out[117]: <BarContainer object of 32 artists>
```

QUIZ

Quiz 1

Which of the following will be used to write to csv file:

A pd.read_csv("Filename.csv")

B pd.read_file("Filename.csv")

C dataframe.read_ csv("Filename.csv")

D dataframe.read_file("Filename.csv")

```
import numpy as np
from scipy import stats
new=np.array([[0.1,1,2],[3,0.2,1],[
1,0.5,4]])
_,p,_,_
=stats.chi2_contingency(new)
```

Answer 1

Which of the following will be used to write to csv file:

A pd.read_csv("Filename.csv")

B pd.read_file("Filename.csv")

C dataframe.read_csv("Filename.csv")

D dataframe.read_file("Filename.csv")

```
import numpy as np
from scipy import stats
new=np.array([[0.1,1,2],[3,0.2,1],[
1,0.5,4]])
_,p,_,_
=stats.chi2_contingency(new)
```

Quiz 2

What will be the output of the following?

```
import pandas as pd
arr=[[1,2],[4,5],[7,8]]
df = pd.DataFrame(arr,
index = ['a','b','c'], columns = ['A',
'B'])
df.loc[:,'A']
```

A

```
a    1
b    4
c    7
```

B

```
0    1
1    4
2    7
```

C

```
1    1
2    2
 3   4
```

D

```
a    1
b    2
c    4
```

Answer 2

What will be the output of the following?

A

```
a  1
b  4
c  7
```

B

```
0  1
1  4
2  7
```

C

```
1  1
2  2
 3  4
```

D

```
a  1
b  2
c  4
```

```
import pandas as pd
arr=[[1,2],[4,5],[7,8]]
df = pd.DataFrame(arr,
index = ['a','b','c'], columns = ['A',
'B'])
df.loc[:,'A']
```

Quiz 3

What is the output of the following:?

```
one = ['one','1','one1']
a = ['A','a','Aa']
df1 = pd.DataFrame({'one':  one,
'a': a})
df1 = df1[['one', 'a']]
a = ['A','a','Aa']
b = ['B','b','Bb']
df2 = pd.DataFrame({'a':  a, 'b': b})
df2 = df2[['a', 'b']]
df1.merge(df2,  on='a', how='inner')
```

A

	one	a	b
0	one	A	B
1	1	a	b
2	one1	Aa	Bb

B

	a	b	one
0	A	B	one
1	a	b	1
2	Aa	Bb	one1

C

	a	one	b
0	A	one	B
1	a	1	b
2	Aa	one1	Bb

D

	0	1	2
one	one	1	one1
a	A	a	Aa
b	B	b	Bb

Answer 3

What is the output of the following:?

```
one = ['one','1','one1']
a = ['A','a','Aa']
df1 = pd.DataFrame({'one':  one,
'a': a})
df1 = df1[['one', 'a']]
a = ['A','a','Aa']
b = ['B','b','Bb']
df2 = pd.DataFrame({'a':  a, 'b': b})
df2 = df2[['a', 'b']]
df1.merge(df2,  on='a', how='inner')
```

A

	one	a	b
0	one	A	B
1	1	a	b
2	one1	Aa	Bb

B

	a	b	one
0	A	B	one
1	a	b	1
2	Aa	Bb	one1

C

	a	one	b
0	A	one	B
1	a	1	b
2	Aa	one1	Bb

D

	0	1	2
one	one	1	one1
a	A	a	Aa
b	B	b	Bb

Quiz 4

To view the first 15 rows of the a given csv dataset, which command is used?

```
import pandas as pd
Dataframe1=pd.read_csv("somedataset.csv")
```

A Dataframe1.row(15)

B Dataframe1.row(0:15)

C Dataframe1.head(0:15)

D Dataframe1.head(15)

Answer 4

To view the first 15 rows of the a given csv dataset, which command is used?

- A Dataframe1.row(15)
- B Dataframe1.row(0:15)
- C Dataframe1.head(0:15)
- D Dataframe1.head(15)

```
import pandas as pd
Dataframe1=pd.read_csv("somedataset.csv")
```

Quiz 5

To rename a column in a given csv dataset, which command is used?

```
import pandas as pd
Dataframe1=pd.read_csv("somedataset.csv")
```

A Dataframe1.rename(columns={'Old-name':'New-name'})

B Dataframe1.title('Old-name':'New-name')

C Dataframe1.rename('Old-name':'New-name')

D Dataframe1.title(columns={'Old-name':'New-name'})

Answer 5

To rename a column in a given csv dataset, which command is used?

[✓ A]	Dataframe1.rename(columns={'Old-name':'New-name'})
B	Dataframe1.title('Old-name':'New-name')
C	Dataframe1.rename('Old-name':'New-name')
D	Dataframe1.title(columns={'Old-name':'New-name'})

```
import pandas as pd
Dataframe1=pd.read_csv("somedataset.csv")
```

Thank You

www.intellipaat.com

CALL US NOW

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com