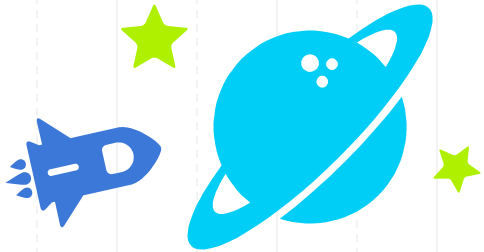


D-Processing

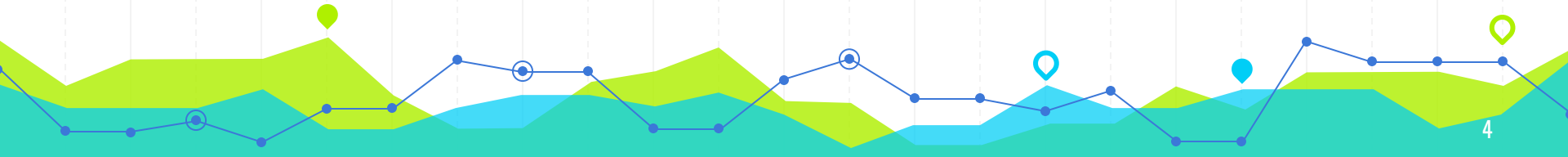
Processing of data includes advance concepts of pandas and allows you to create, clean and manipulate data.

Data Processing OPERATIONS

- Hierarchical Indexing
- Files Operation
- Merge
- Data Transformation
- Groupby and Data aggregator

“

import pandas as pd



Multiple Indexing

```
>>> import pandas as pd
>>> data = pd.Series([10, 20, 30, 40, 15, 25, 35, 25], index = [['a', 'a',
... 'a', 'a', 'b', 'b', 'b', 'b'], ['obj1', 'obj2', 'obj3', 'obj4', 'obj1',
... 'obj2', 'obj3', 'obj4']])
>>> data
a  obj1    10
   obj2    20
   obj3    30
   obj4    40
b  obj1    15
```

Check Data Index

```
>>> data.index
MultiIndex(levels=[['a', 'b'], ['obj1', 'obj2', 'obj3', 'obj4']],
           labels=[[0, 0, 0, 0, 1, 1, 1, 1], [0, 1, 2, 3, 0, 1, 2, 3]])
```

Partial Index

```
>>> data['b']
obj1   15
obj2   25
obj3   35
>>> data[:, 'obj2']
a   20
b   25
```

Unstack Data

```
>>> # unstack based on first level i.e. a, b
>>> # note that data row-labels are a and b
>>> data.unstack(0)
      a   b
obj1  10  15
obj2  20  25
obj3  30  35
obj4  40  25
```

Unstack Data

```
>>> # unstack based on second level i.e. 'obj'
>>> data.unstack(1)
   obj1  obj2  obj3  obj4
a    10    20    30    40
b    15    25    35    25
>>>
```

Unstack Data

```
>>> # by default innermost level is used for unstacking
>>> d = data.unstack()
>>> d
   obj1  obj2  obj3  obj4
a    10    20    30    40
b    15    25    35    25
```

Stack Data

```
>>> d = data.unstack()
>>> d
   obj1  obj2  obj3  obj4
a    10    20    30    40
>>> d.stack()
```

Column Indexing

```
>>> import numpy as np
>>> df = pd.DataFrame(np.arange(12).reshape(4, 3),
...     index = [['a', 'a', 'b', 'b'], ['one', 'two', 'three', 'four']],
...     columns = [['num1', 'num2', 'num3'], ['red', 'green', 'red']]
... )
>>>
```

Display Columns and Index

```
>>> # display row index
>>> df.index
MultiIndex(levels=[['a', 'b'], ['four', 'one', 'three', 'two']],
           labels=[[0, 0, 1, 1], [1, 3, 2, 0]])

>>> # display column index
>>> df.columns
MultiIndex(levels=[['num1', 'num2', 'num3'], ['green', 'red']],
           labels=[[0, 1, 2], [1, 0, 1]])
```

Naming Columns names and Index names

```
>>> df.index.names=['key1', 'key2']
>>> df.columns.names=['n', 'color']
>>> df
```

Accessing Values with rows and columns

```
>>> df[‘num1’]
>>> df.loc['a']
>>> df.iloc[0]
```

Swap and Sort level

```
>>> df.swaplevel('key1', 'key2')
>>> df.sort_index(level='key2')
```

Add rows with similar keys

```
>>> # add all rows with similar key1 name
>>> df.sum(level = 'key1')
n     num1  num2 num3
color  red green  red
key1
a       3    5   7
b      15   17   19
>>>
```

Add columns with similar keys

```
>>> # add all the columns based on similar color
>>> df.sum(level= 'color', axis=1)
color      green  red
key1 key2
a    one       1    2
     two       4    8
b    three     7   14
     four     10   20
```

Load data with pipe separated files

```
data = pd.read_table('url',sep='|')
data
```

Column Concatination

```
data['Unit Power'] = data['Unit'] + data['Power'].apply('str')
data
```