

# Data X

## Pandas Overview Data-X: A Course on Data, Signals, and Systems

Ikhtlaq Sidhu  
Faculty Director and Founder  
Sutardja Center for Entrepreneurship & Technology  
IEOR Emerging Area Professor Award, UC Berkeley

- Review Ref CS03: [pandas in 10 min](#)
- Notebook versions available at [data-x.blog](#) in module-02



# How Does Pandas Fit In?

- Python is great: easy to understand, compact, flexible – “duct tape of internet”
- Python was not originally built for data analytics
- Sci-Py extends to mathematics, science, and engineering



NumPy  
Base N-dimensional  
array package



SciPy library  
Fundamental  
library for scientific  
computing

Numpy allows arrays  
and matrix math



Matplotlib  
Comprehensive 2D  
Plotting



IPython  
Enhanced  
Interactive Console



Sympy  
Symbolic  
mathematics



pandas  
Data structures &  
analysis

Pandas provides a  
table structure

Data<sup>x</sup>

## Pandas lets us construct tables, called Data Frames

With NumPy, we  
can store and  
manipulate a  
matrix

m =

```
[[-0.09443539 -0.09443531  0.29860729 -0.09761513 -0.09440866]
 [-0.09443526 -0.09443531  0.25596021 -0.10824217 -0.094422  ]
 [-0.09443524 -0.09443531  0.37198598 -0.12371693 -0.09442577]
 [-0.09443568 -0.09443531  0.30667577 -0.10257815 -0.09441752]
 [-0.09443562 -0.09443531  0.41545527 -0.06368836 -0.09441873]
 [-0.09443647 -0.09443531  0.34410876  0.00738793 -0.09440932]
 [-0.0944355  -0.09443531  0.33180906 -0.12472302 -0.09442687]
 [-0.09443587 -0.09443531  0.3643611  -0.16894118 -0.09443041]
 [-0.09443721 -0.09443531  0.43028699  0.0095  -0.09441093]
 [-0.09443846 -0.09443531  0.34737789 -0.07818481 -0.09439922]]
```

With Pandas, we  
can store and  
manipulate a full  
table

df =

	Birth Month	Origin	Age	Gender
<b>Carly</b>	January	UK	27	f
<b>Rachel</b>	September	Spain	28	f
<b>Nicky</b>	September	Jamaica	28	f
<b>Wendy</b>	November	Italy	22	f
<b>Judith</b>	February	France	19	f

DataX

**Pandas** has an object called a *Data Frame* which is like a table

columns		foo	bar	baz	qux
index					
A	→	0	x	2.7	True
B	→	4	y	6	True
C	→	8	z	10	False
D	→	-12	w	NA	False
E	→	16	a	18	False

- NumPy array-like
- Each column can have a different type
- Row and column index
- Size mutable: insert and delete columns

Wes Mckinney

DataX

## Data Structures - High Level

List:

`L = [0, b, "hello"]`

What is `L2 = [0, b, (b,bat)]`?

Numpy Array: (vector)

`arr = np.array([5,4,3,2,1])`

Numpy Array: (matrix)

`mat =`

`np.array([[5,4],[3,2],[1,0]])`

Using the Axis:

`mat.sum(axis=0)`

`mat.min(axis=1)`

`L` →

0  
B  
"hello"

`arr` →

5  
4  
3  
2  
1

`mat` →

5 4  
3 2  
1 0



## Data Structures - High Level

Dictionary:

```
d = { 'dog':20, 'cat':10, 'mouse':1 }
```

What is `d['cat']`?

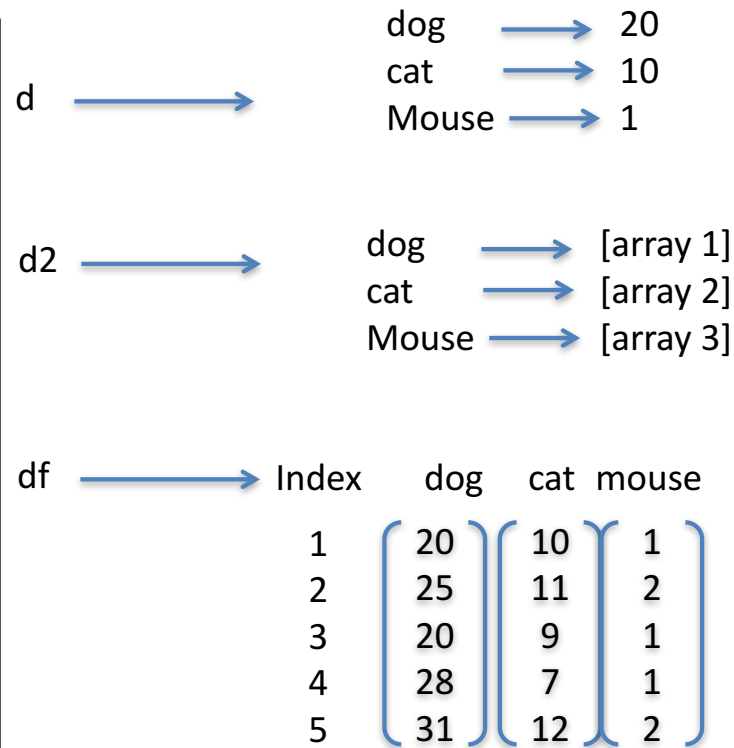
Pandas Data Frame:

Made of Dictionary of  
'labels' and numpy-like arrays  
called Series

```
d2 = { 'dog':ar1, 'cat':ar2,  
       'mouse':ar3 }
```

```
df = pd.DataFrame(d2)
```

What is `df['cat']`?



\* Actually made from the Series object in Pandas

## Code Example in Python Notebook

- Get Stock Data
- Use Pandas to get a CSV format
- Slice the Table
- Convert to Numpy Array Format
- Sample Numpy Operations





# More topics in the 10 Min Guide to Pandas Notebook

## Indexing

DF1

	Quantity	Revenue	Points
Product			
A	523	1103.25	5230
B	200	1525.10	860
C	148	3892.50	0
D	1610	5730.25	0
E	122	580.12	600
F	10	55342.00	100

```
df1.loc['C']  
  
Quantity    148.0  
Revenue     3892.5  
Points        0.0  
Name: C, dtype: float64
```

## Computational Tools

- Covariance  

```
>>> s1 = Series(randn(1000))  
>>> s2 = Series(randn(1000))  
>>> s1.cov(s2)  
0.013973709323221539
```
- Also: pearson, kendall, spearman

Maik Röder

## Descriptive statistics

```
>>> df.mean()  
one      2.263617  
two      -1.316694  
three    -1.975041
```

- Also: count, sum, median, min, max, abs, prod, std, var, skew, kurt, quantile, cumsum, cumprod, cummax, cummin

Maik Röder

## Adding Pandas Tables

	Quantity	Revenue	Points
Product			
A	523	1103.25	5230
B	200	1525.10	860
C	148	3892.50	0
D	1610	5730.25	0
E	122	580.12	600
F	10	55342.00	100

+

	Quantity	Revenue
Product		
D	0	0.00
A	100	22.50
C	200	540.25
B	300	1534.00
E	400	2134.00

=

	Quantity	Revenue	Points
Product			
A	623	1125.75	NaN
B	500	3059.10	NaN
C	348	4432.75	NaN
D	1610	5730.25	NaN
E	522	2714.12	NaN
F	NaN	NaN	NaN

```
df_add = df1.add(df2, fill_value=0)
```

df\_add = df1.add(df2, fill\_value=0)

DataX

End of Section

Data<sup>x</sup>