Rotman

PANDAS

February 15, 2022 Prepared by Niti
TDMDAL & FinHUB

Rotman School of Management
UNIVERSITY OF TORONTO

Pandas

pandas

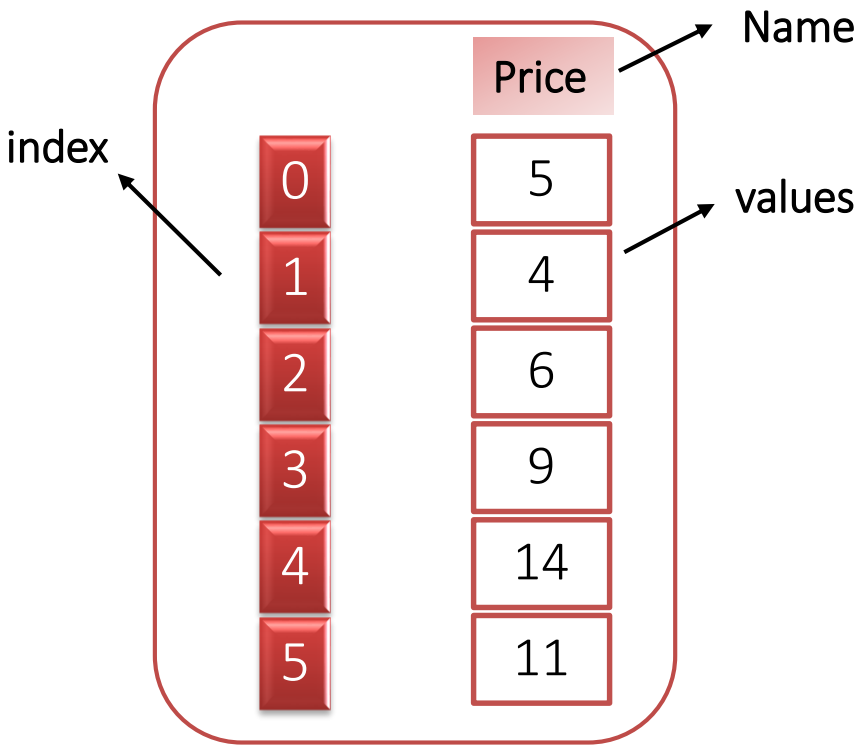
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

- Most widely used python library in data science
- A high-performance tool to extract, clean, transform and analyze data
- Provides an efficient implementation of
  - ➢ multidimensional arrays with row and column labels
  - ➢ allows heterogeneous data types as well as missing data

Series

Pandas : Data Structures

Series

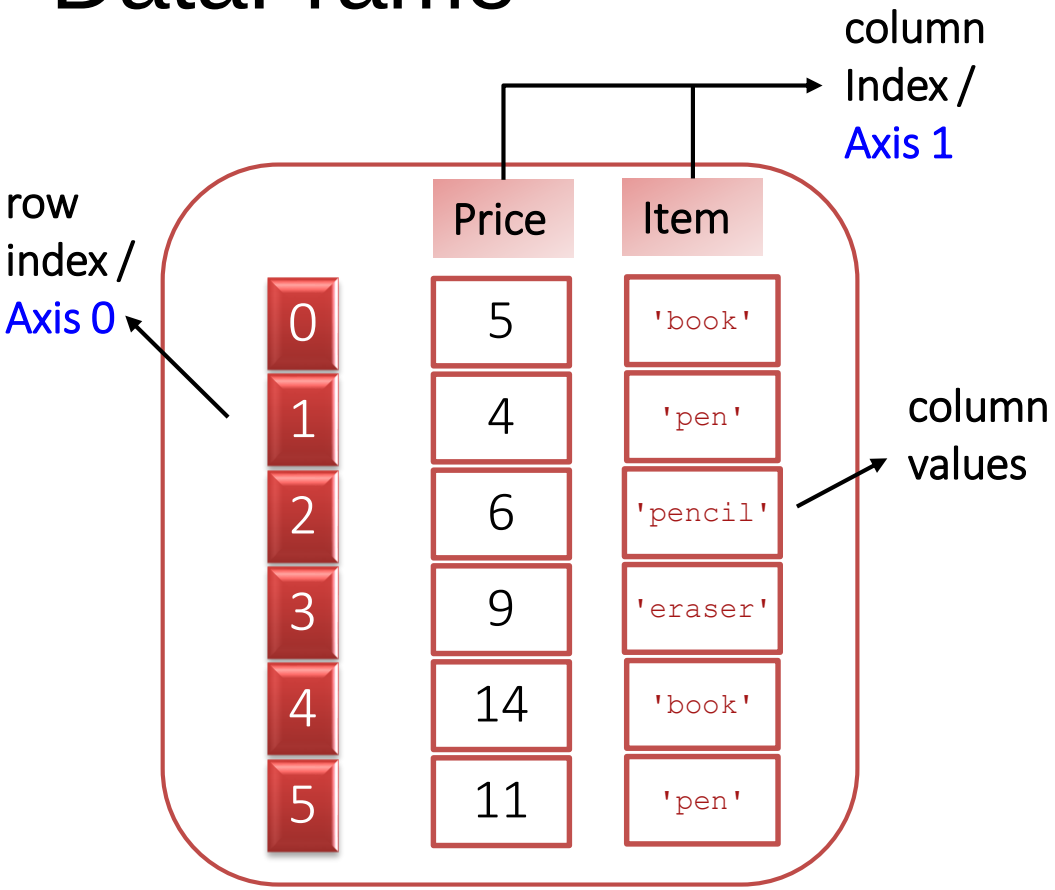


- an array of 1 dimension (similar to column in excel)
- somewhat similar to Python's dictionary in that it provides a mapping from a collection of keys to a collection of values
- each column of a pandas dataframe is a series.

DataFrame

Pandas : Data Structures

DataFrame



- a table of 2 dimension (similar to sheet in excel)
- a collection of series
- also has Python's dictionary like interface, except for the key has multiple values, one for each Series.

Indexing in Pandas

Pandas : Indexing and slicing

- Pandas dataframe allows extractions of elements from both:
  - rows and
  - columns

- Two methods are available to extract elements:
  - .iloc()
  - .loc()

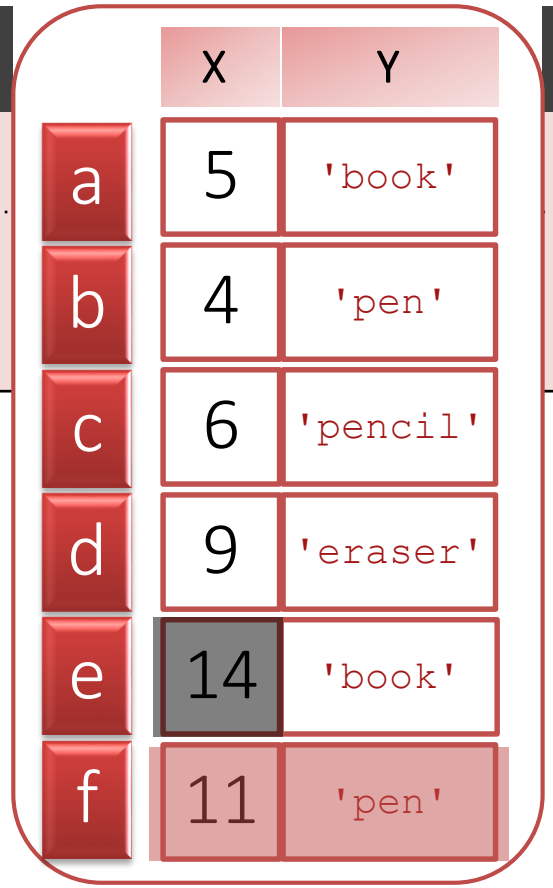
- Extracting elements based on conditions is also possible

Pandas : Indexing

INDEX POSITION	index	ROW LABEL / COLUMN NAME
df[position] *(for series only)*	row	df.loc[row label]
df[start : stop positions]	rows	df.loc[start : stop row labels] df.loc[list of row labels]
Not Possible *(See below)*	column	df[column name]
	columns	df[list of column names]
df.iloc[start : stop positions , start : stop positions]	rows + columns	df.loc [start : stop / list of row labels , start : stop / list of column names]

Pandas : Indexing rows

INDEX POSITION		ROW LABEL / COLUMN NAME
X[4] *(for series only)*		df.loc['f']
df[0 : 4]		df.loc[['a','b','c','d']] df.loc['a' : 'd']

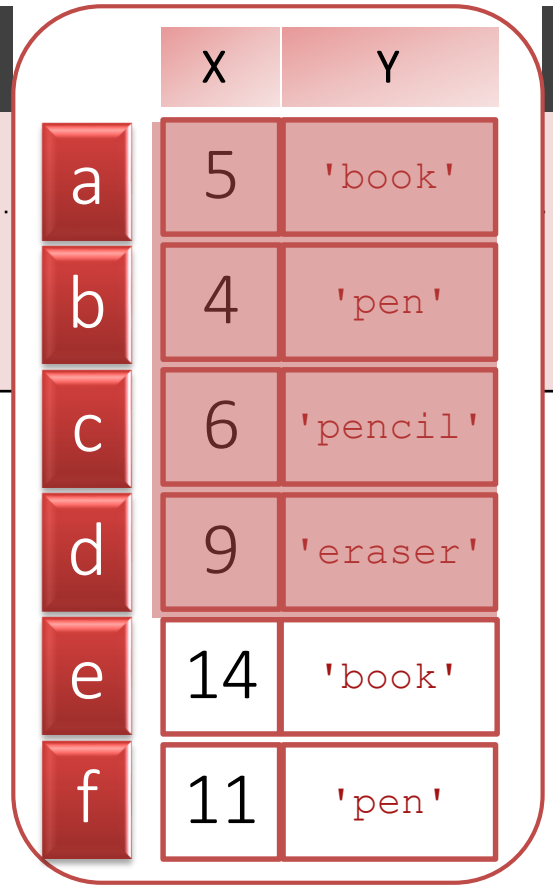


	X	Y
a	5	'book'
b	4	'pen'
c	6	'pencil'
d	9	'eraser'
e	14	'book'
f	11	'pen'

df

Pandas : Indexing rows

INDEX POSITION		ROW LABEL / COLUMN NAME
X[4] *(for series only)*		df.loc['f']
df[0 : 4]		df.loc[['a','b','c','d']] df.loc['a' : 'd']

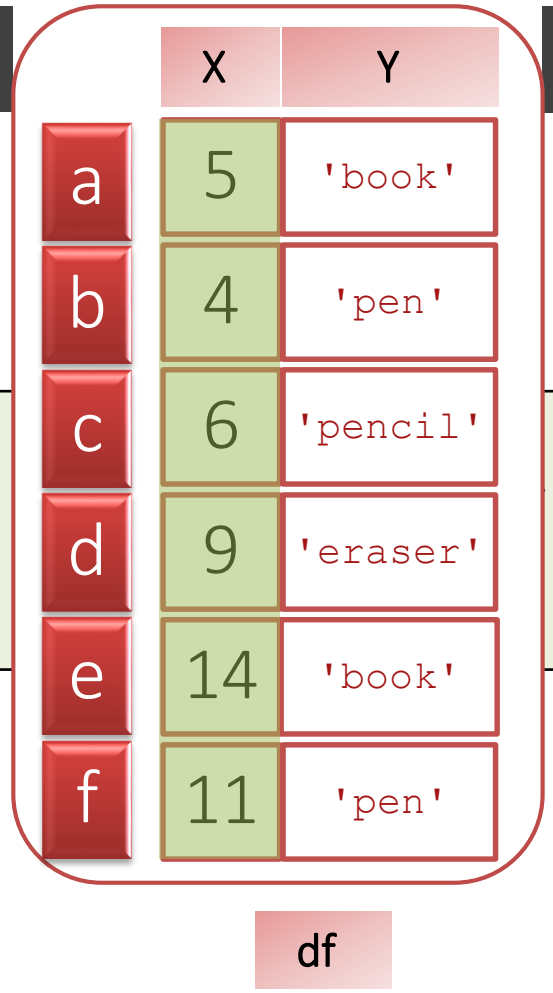


	X	Y
a	5	'book'
b	4	'pen'
c	6	'pencil'
d	9	'eraser'
e	14	'book'
f	11	'pen'

df

Pandas : Indexing columns

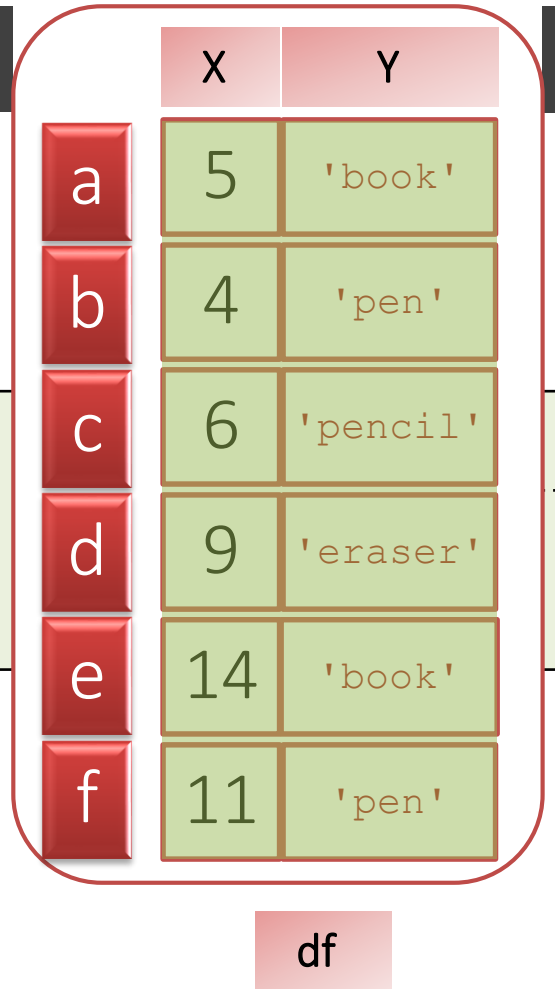
INDEX POSITION	ROW LABEL / COLUMN NAME
Not Possible *(See below)*	`df['X']` `df[['X','Y']]`



Rotman

Pandas : Indexing columns

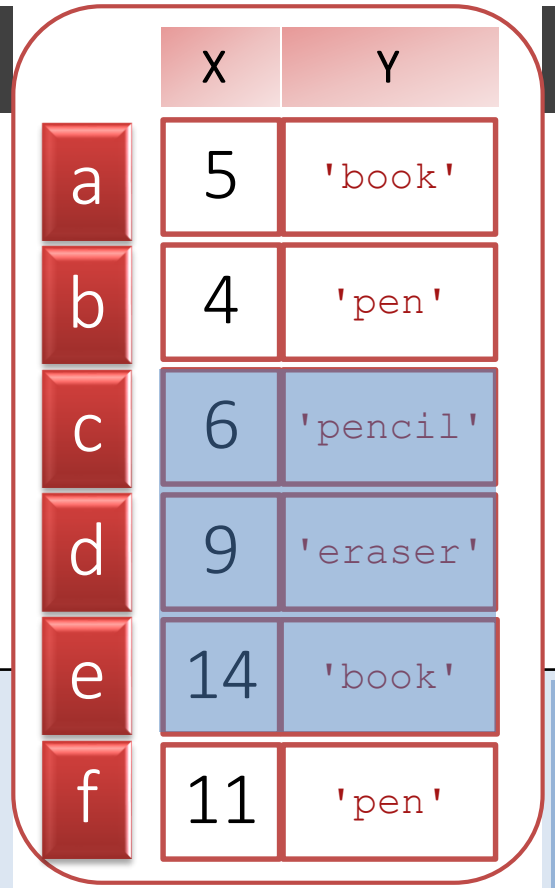
INDEX POSITION		ROW LABEL / COLUMN NAME
Not Possible *(See below)*		`df['X']` `df[['X','Y']]`



Rotman

Pandas : Indexing rows and columns

INDEX POSITION		ROW LABEL / COLUMN NAME
df.iloc[2:5 , 0:2]		df.loc['c':'e' , 'X':'Y'] df.loc[['c','d','e'], ['X','Y']]



	X	Y
a	5	'book'
b	4	'pen'
c	6	'pencil'
d	9	'eraser'
e	14	'book'
f	11	'pen'

df

Pandas : Conditional indexing

CONDITIONAL INDEXING

Index labels can also be used to filter based on conditions.

Multiple conditions can also be specified.

	X	Y
a	5	'book'
b	4	'pen'
c	6	'pencil'
d	9	'eraser'
e	14	'book'
f	11	'pen'

df

CONDITIONAL INDEXING

```
df.loc[ df['Y'] == 'book']
```

```
df.loc[ df['X'] >= 10, 'Y']
```

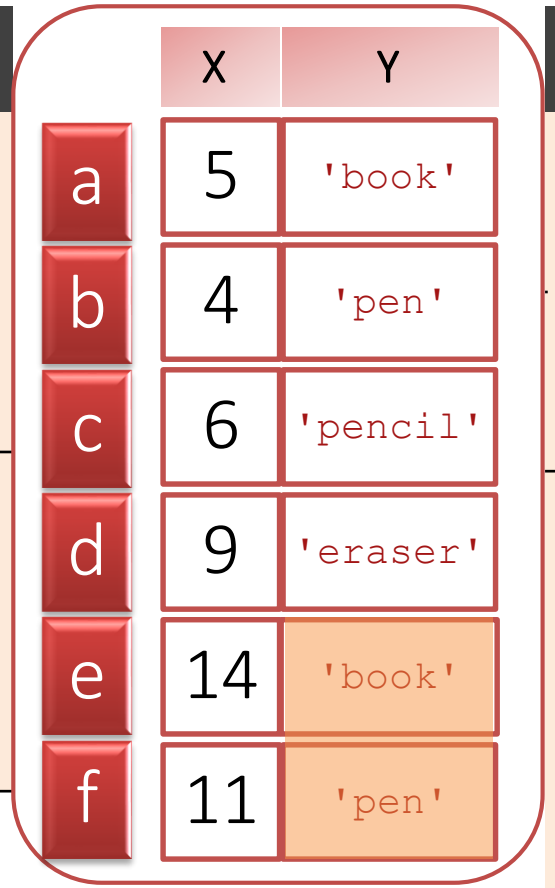
```
df.loc[
        ( df['X'] < 5 ) &
        ( df['Y'] == 'pen' ),
                ['X':'Y']
              ]
```

Pandas : Conditional indexing

CONDITIONAL INDEXING

Index labels can also be used to filter based on conditions.

Multiple conditions can also be specified.



	X	Y
a	5	'book'
b	4	'pen'
c	6	'pencil'
d	9	'eraser'
e	14	'book'
f	11	'pen'

df

CONDITIONAL INDEXING

```
df.loc[ df['Y'] == 'book']
```

```
df.loc[ df['X'] >= 10, 'Y']
```

```
df.loc[
        ( df['X'] < 5 ) &
        ( df['Y'] == 'pen' ),
                ['X':'Y']
            ]
```


Rotman

Pandas : Conditional indexing

CONDITIONAL INDEXING

Index labels can also be used to filter based on conditions.

Multiple conditions can also be specified.

	X	Y
a	5	'book'
b	4	'pen'
c	6	'pencil'
d	9	'eraser'
e	14	'book'
f	11	'pen'

df

CONDITIONAL INDEXING

```
df.loc[ df['Y'] == 'book']
```

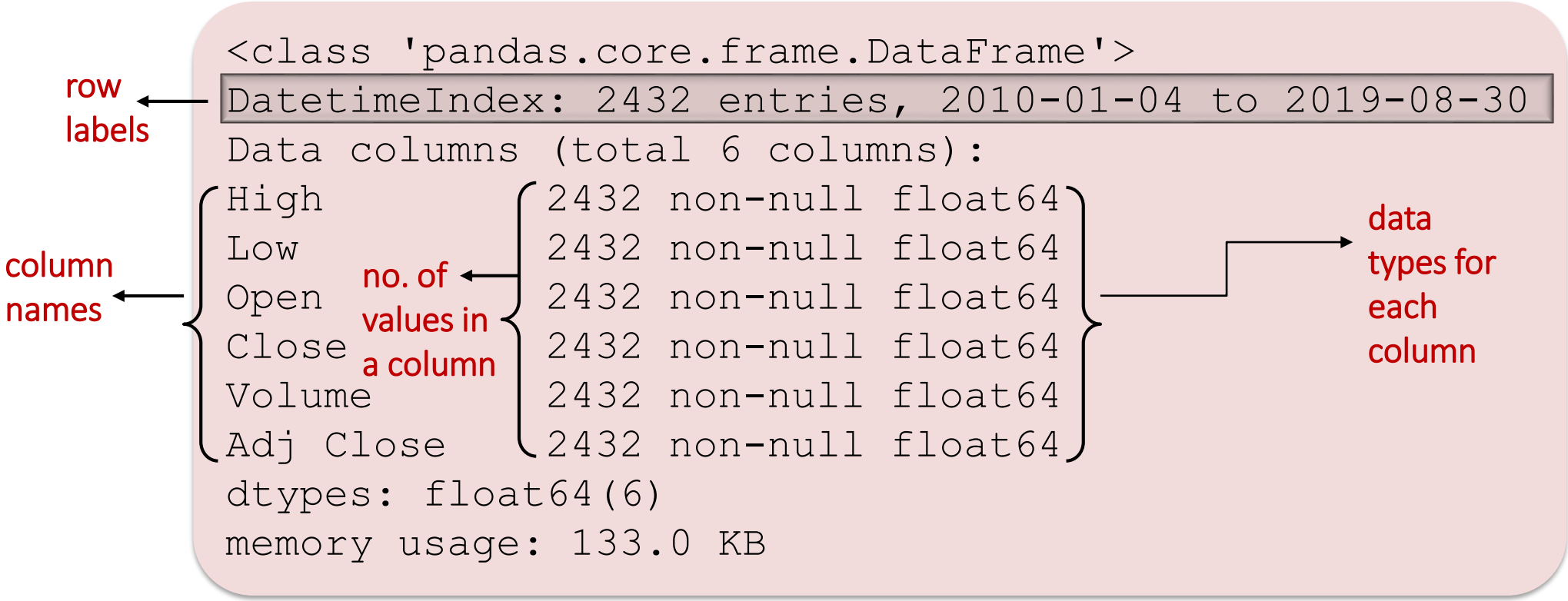
```
df.loc[ df['X'] >= 10, 'Y']
```

```
df.loc[
        ( df['X'] < 5 ) &
        ( df['Y'] == 'pen' ),
                ['X':'Y']
              ]
```

Common Methods

Pandas : Methods

.info()

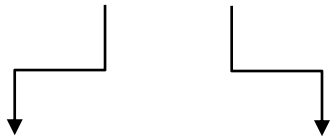


Rotman

Pandas : Methods

.shape

```
(2432,6)
```

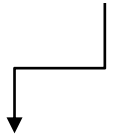


- Axis 0
- Refers to no. of rows

- Axis 1
- Refers to no. of columns

.ndim

```
2
```



- Refers to. of dimensions

.describe ()

- Get statistical summary such as average, standard deviation, minimum, maximum and quantile values of each numeric column

Pandas : Methods

Simple model that predict today's price as the average of the price of the last 2 days

DataFrame (df) for daily price of a stock

5.49
5.65
5.87
6.32
6.59
6.85
7.56
8.56
9.25
9.99

N-days

(5.49 + 5.65) / 2

(5.65 + 5.87) / 2

(5.87 + 6.32) / 2

How do we do this in Pandas?

.rolling()

.shift()

```
df['Price'].rolling(window=2).mean().shift()
```

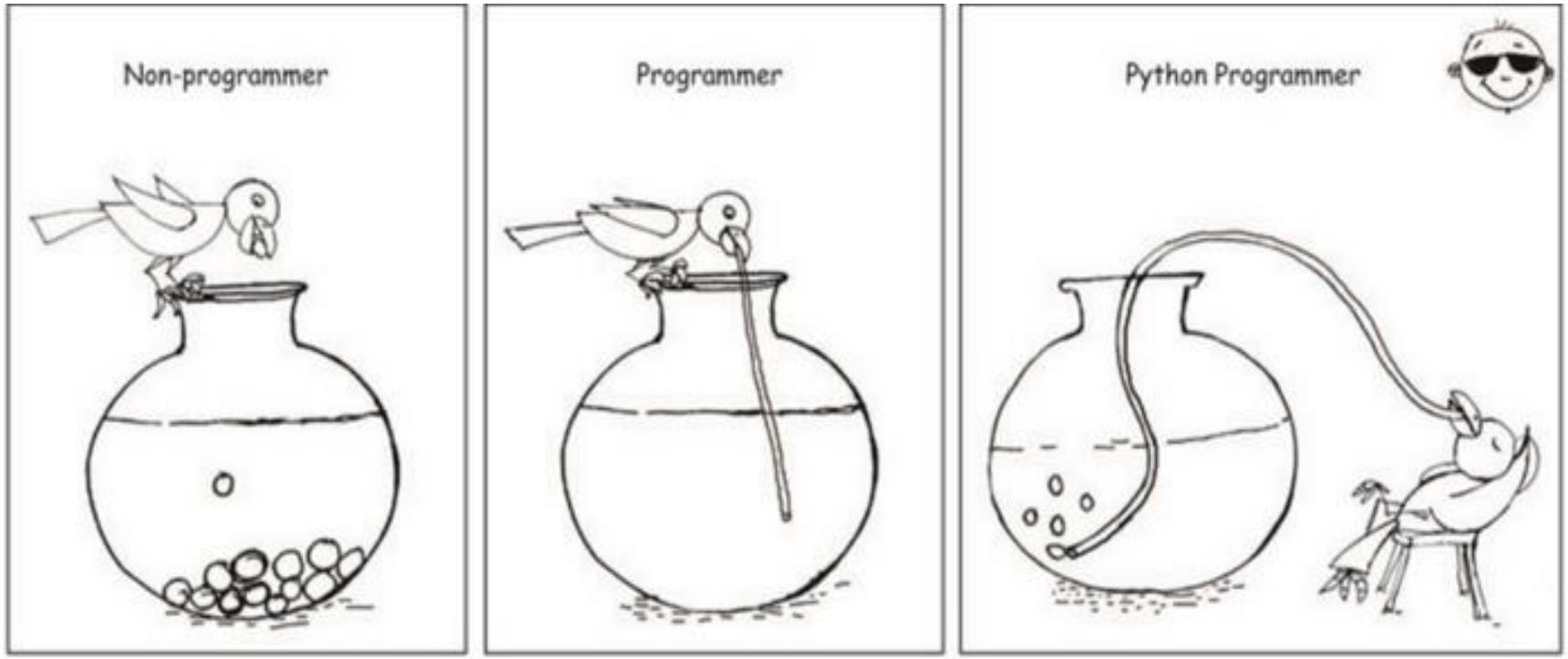
NaN
5.57
5.76
6.10
6.45
6.72
7.20
8.06
8.90
9.62

N-1 days

NaN
NaN
5.57
5.76
6.10
6.45
6.72
7.20
8.06
8.90

N-1 days

Questions?



Who wants to become a Python Programmer?

Thank you