Learn Data Science in Python

Matrix and Data Frame

Feb 11, 2017

Welcome!

February 4	1	Setup & Basics
February 11	2	Matrix & Data Frame
February 18	3	Loop & Function
February 25	4	Statistical Methods
March 4	5	Report & Data Viz

Welcome!

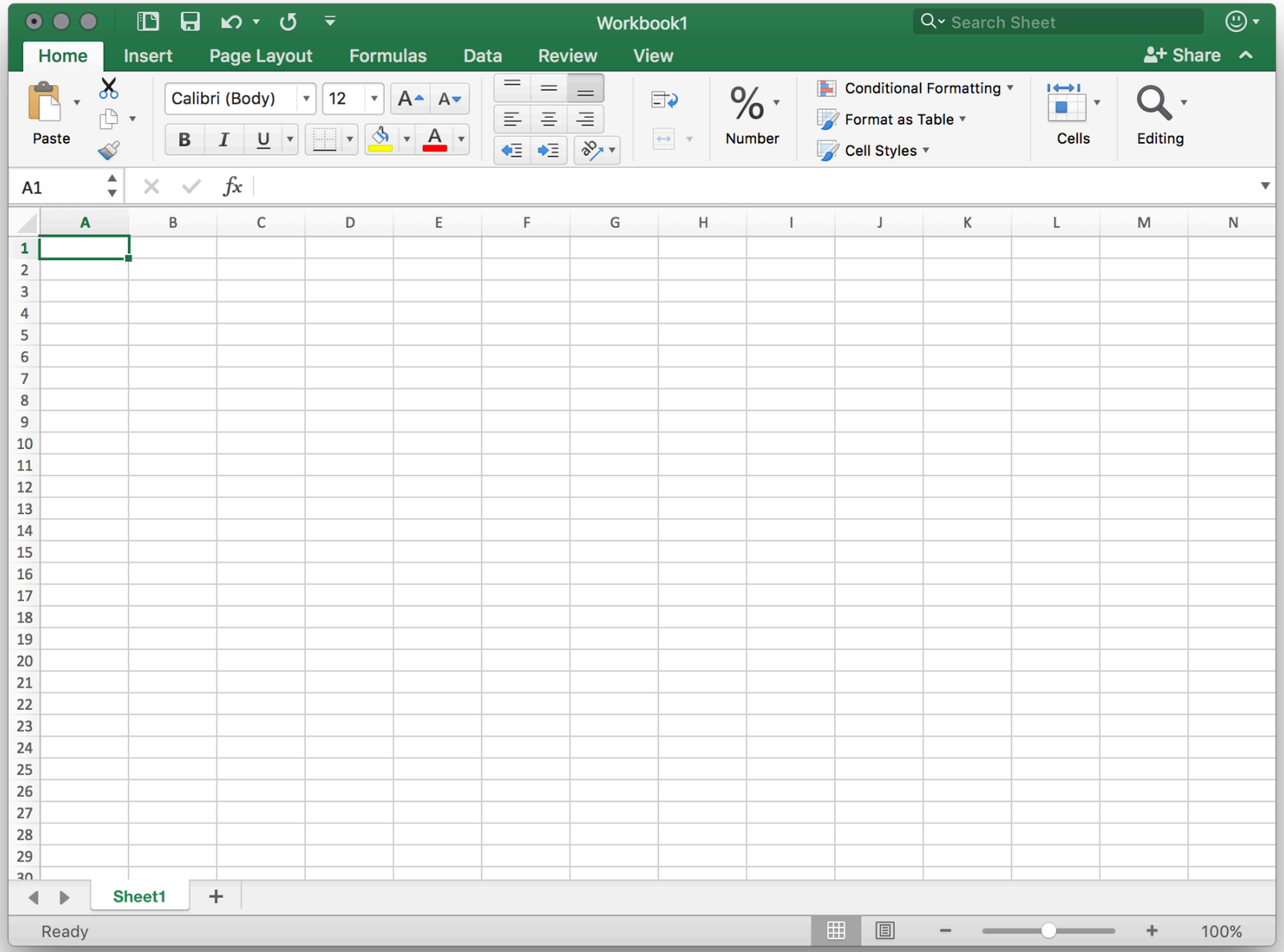
What we've learned previously:

How to assign variables

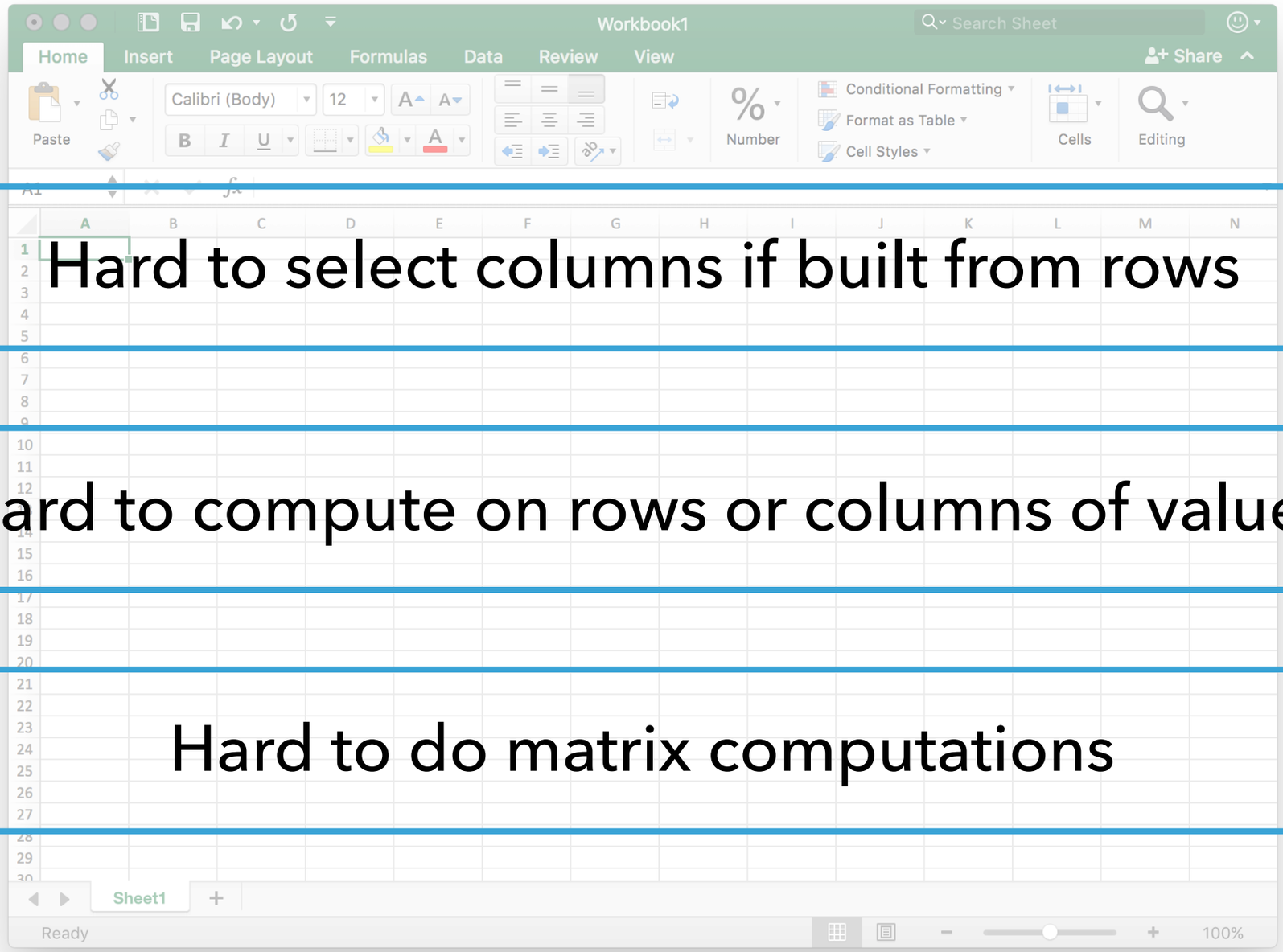
How to calculate numbers

How to manipulate strings and lists

How Far Ahead?



How Far Ahead?

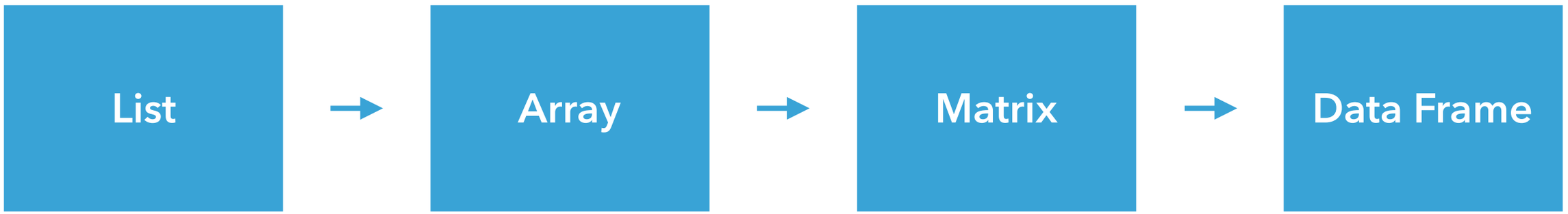


Hard to select columns if built from rows

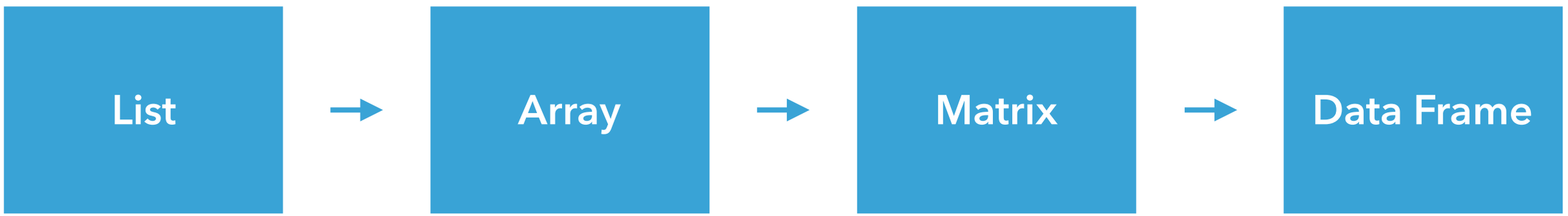
Hard to compute on rows or columns of values

Hard to do matrix computations

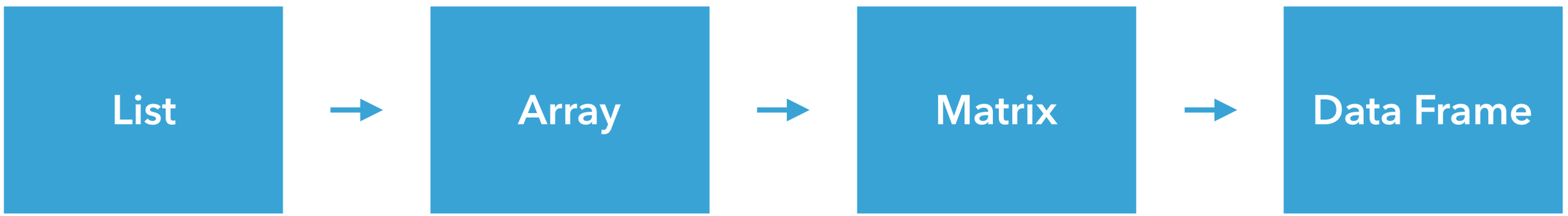
Let's Move On



Let's Move On


List
Array
Matrix
Data Frame
N-dimension
N-dimension
2-dimension
2-dimension

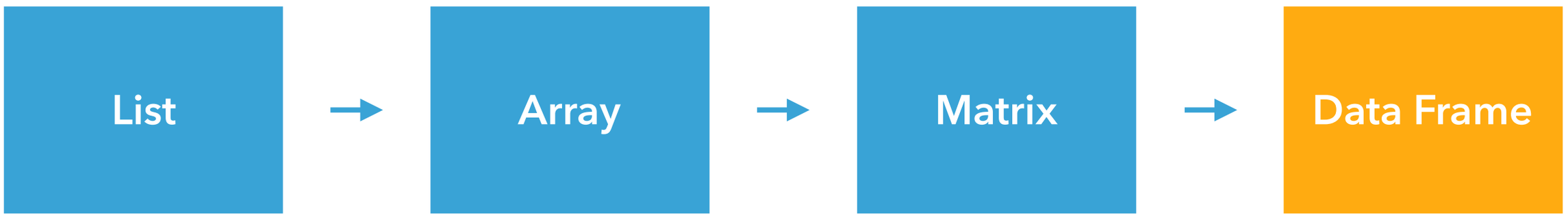
Let's Move On


List
Array
Matrix
Data Frame
N-dimension
N-dimension
2-dimension
2-dimension
Row & Column
Row & Column
Row & Column

Let's Move On

List	Array	Matrix	Data Frame
N-dimension	N-dimension	2-dimension	2-dimension
	Row & Column	Row & Column	Row & Column
Any type	Numerical	Numerical	Any type

Let's Move On


List
Array
Matrix
Data Frame
N-dimension
N-dimension
2-dimension
2-dimension
Row & Column
Row & Column
Row & Column
Any type
Numerical
Numerical
Any type

Array & Matrix

After converting a list with numeric values to a array or matrix, we can do the indexing and calculations in terms of columns or rows.

NumPy

Python + NumPy

NumPy

Python + NumPy

- Array and Matrix
- Useful statistical tools
- A solid foundation for other pkgs

NumPy

Python + NumPy

- Array and Matrix
- Useful statistical tools
- A solid foundation for other pkgs

NumPy

Python + NumPy

- Array and Matrix
- Useful statistical tools
- A solid foundation for other pkgs

`numpy.ndarray`

2D

`numpy.matrix`

NumPy

Python + NumPy

- Array and Matrix
- Useful statistical tools
- A solid foundation for other pkgs

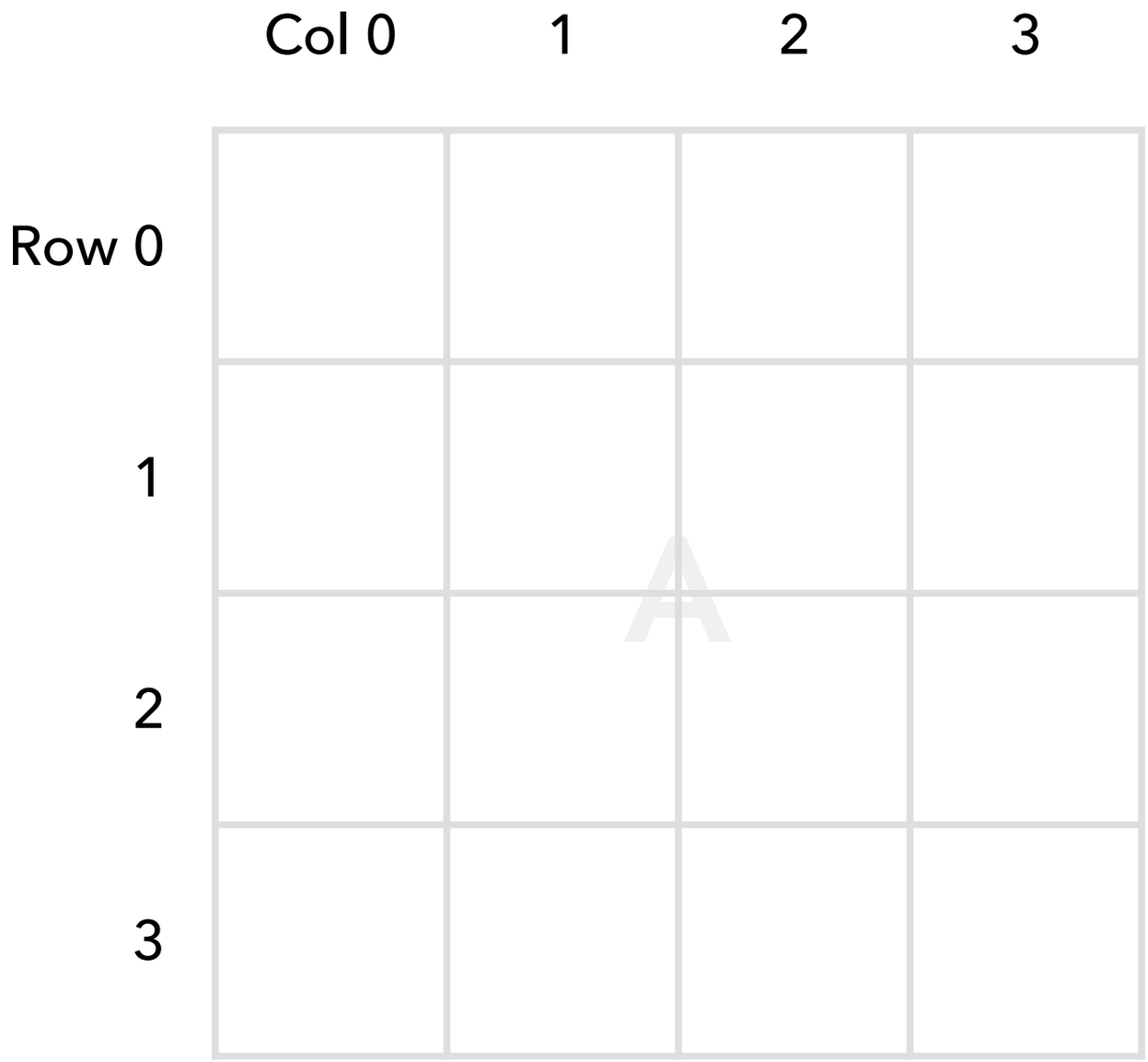
`numpy.ndarray`

2D

`numpy.matrix`

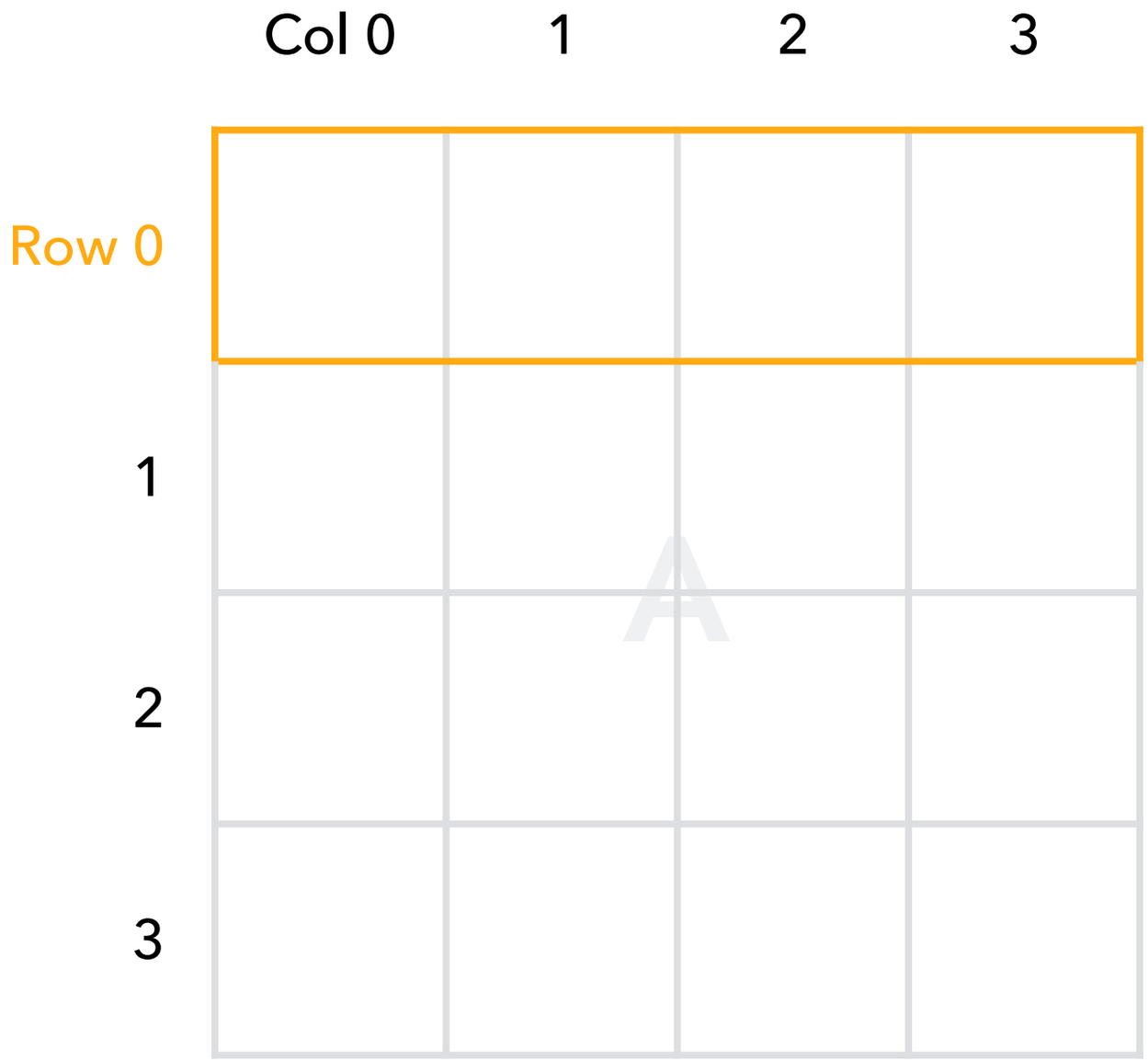
- Matrix calculation
- Subset an array/matrix by using row and column
- Basic manipulation

Index, Slice, Subset




```
A[row,col]
```

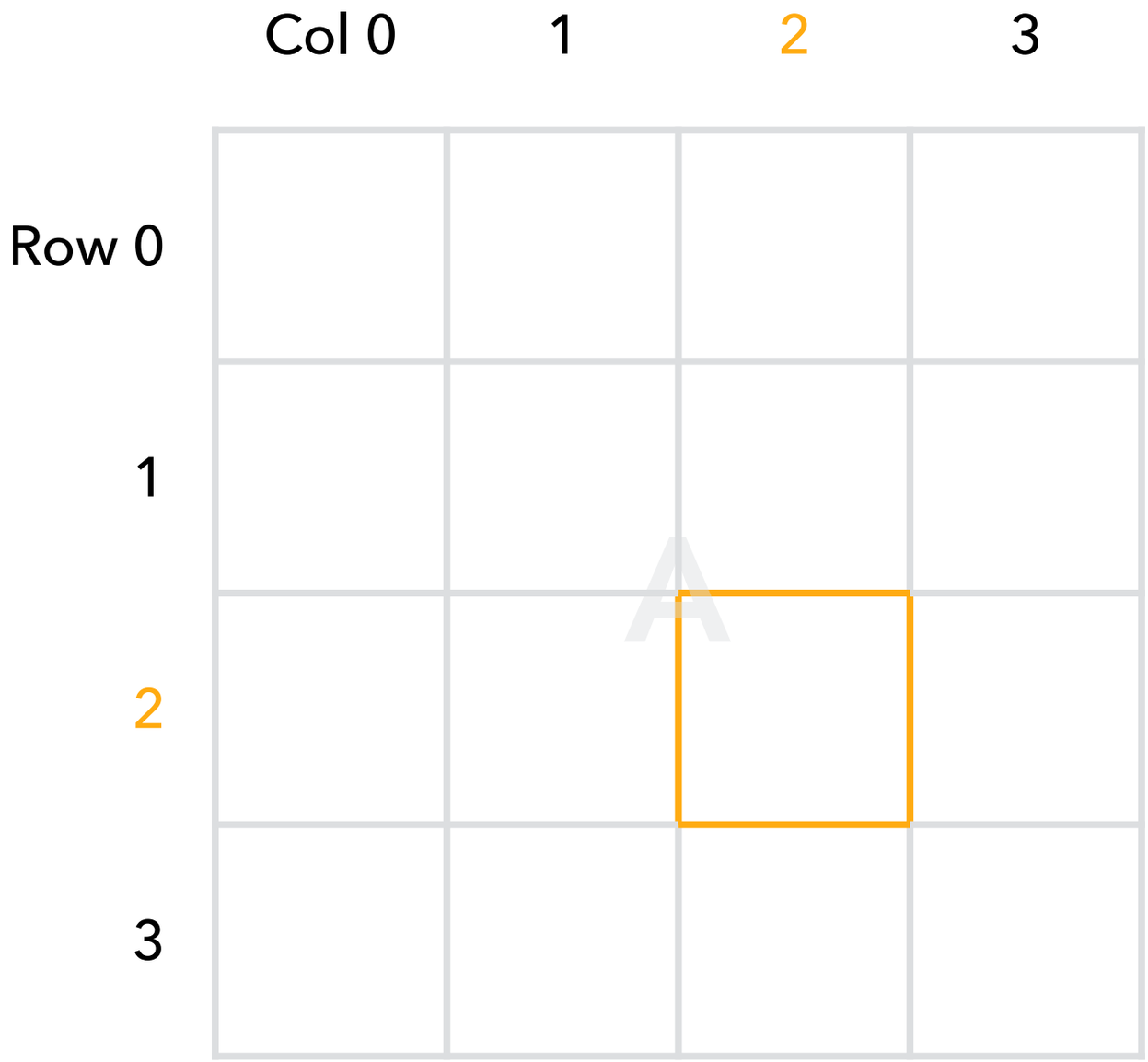
Index, Slice, Subset




```
A[row,col]
```

```
A[0]
```

Index, Slice, Subset

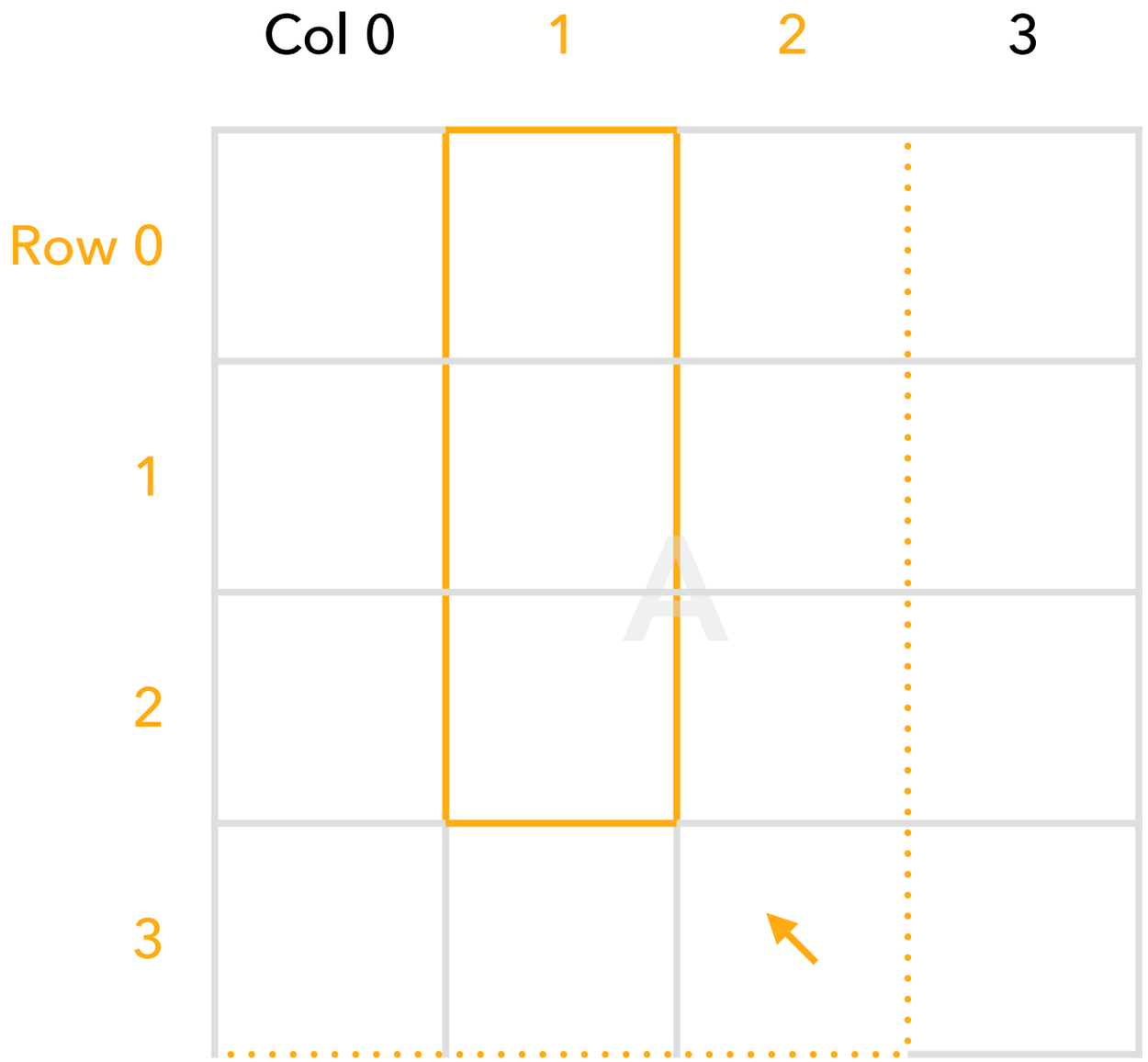



```
A[row,col]
```

```
A[0]
```

```
A[2,2]
```

Index, Slice, Subset



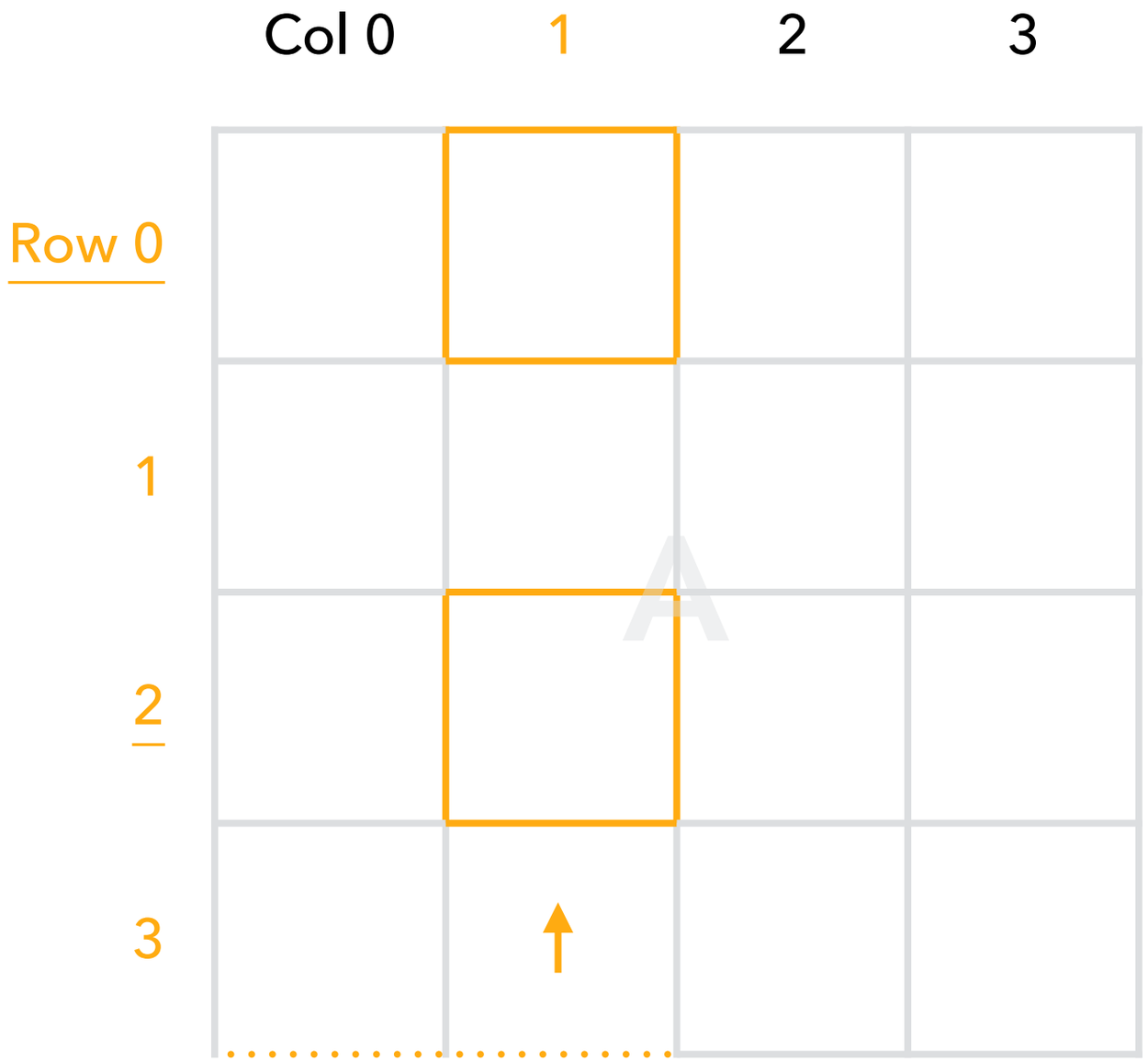

```
A[row,col]
```

```
A[0]
```

```
A[2,2]
```

```
A[0:3,1:2]
```

Index, Slice, Subset




```
A[row,col]
```

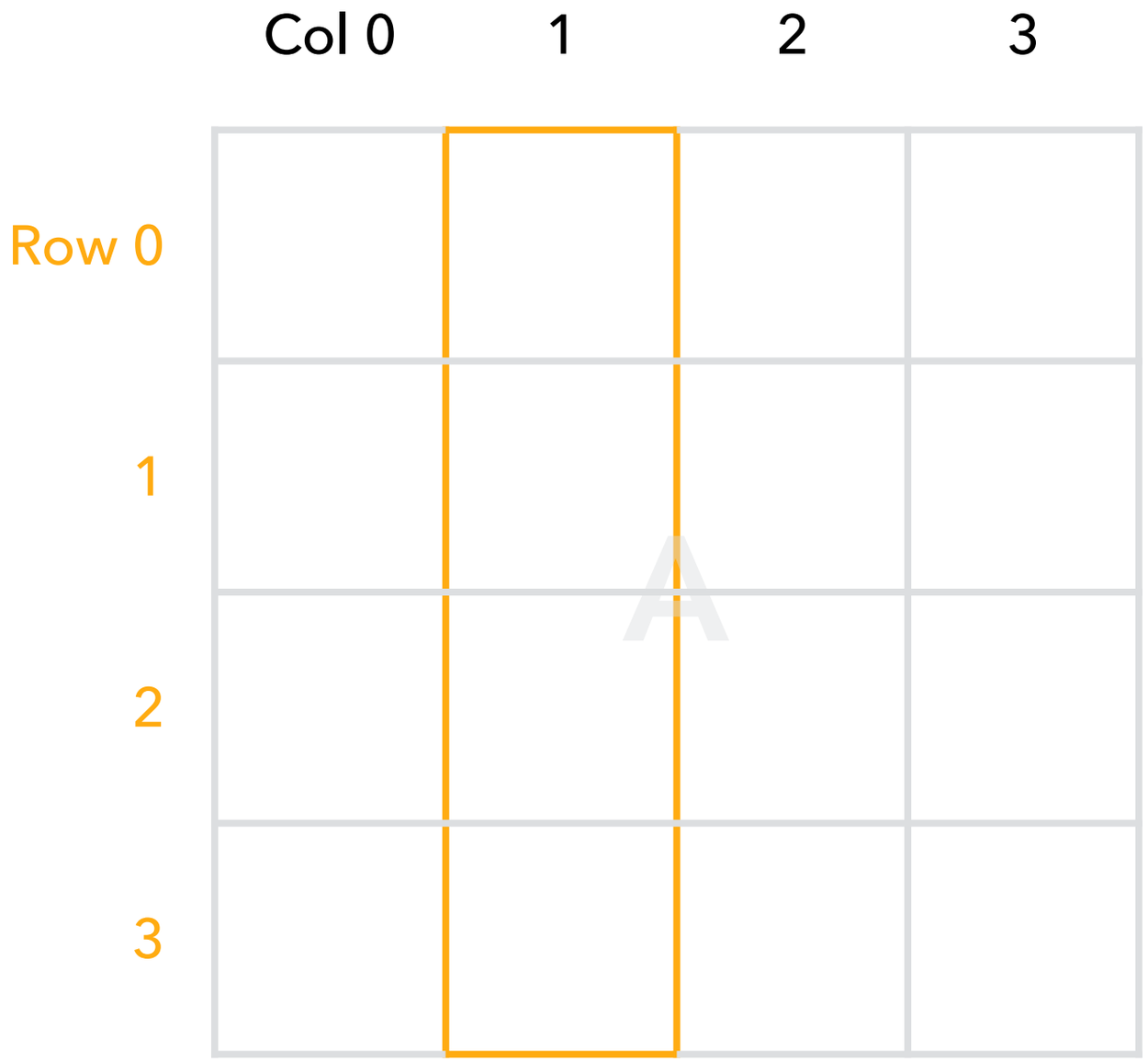
```
A[0]
```

```
A[2,2]
```

```
A[0:3,1:2]
```

```
A[0:3:2,1]
```

Index, Slice, Subset



`A[row,col]`

`A[0]`

`A[2,2]`

`A[0:3,1:2]`

`A[0:3:2,1]`

`A[:,1]`

Before We Move

Python + NumPy

- Array and Matrix
- Useful statistical tools
- A solid foundation for other pkgs

Before We Move

Python + NumPy

- Array and Matrix
- Useful statistical tools
- A solid foundation for other pkgs

Random Sampling

Summarize the Data

Before We Move

Python + NumPy

- Array and Matrix
- Useful statistical tools
- A solid foundation for other pkgs

Random Sampling

Summarize the Data

- SciPy.org
  - Random sampling (numpy.random)
- *Python Data Science Handbook*
  - Computation on NumPy Arrays: Universal Functions
  - Aggregations: Min, Max, and Everything In Between
- Python-Course.eu
  - Python, Random Numbers and Probability

Data Frame

After knowing how to assign values of different data types and index, slice or subset data, we are able to deal with data frame in the most natural way.

Pandas and Data Frame

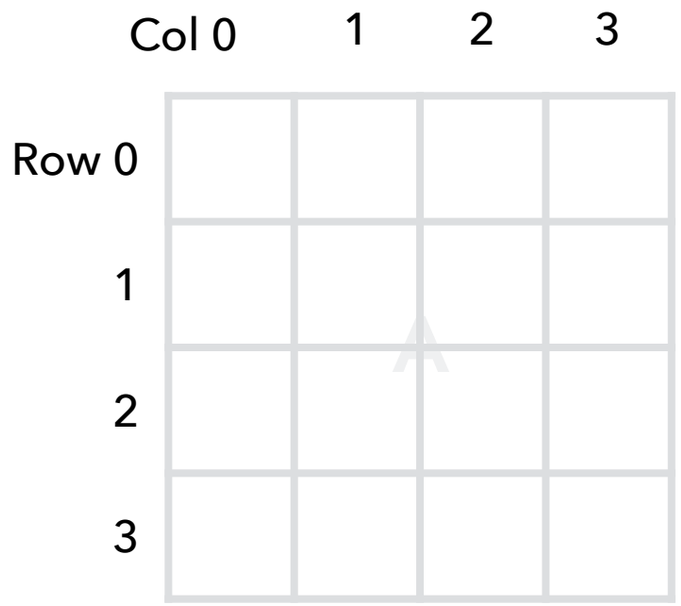
`numpy.ndarray`

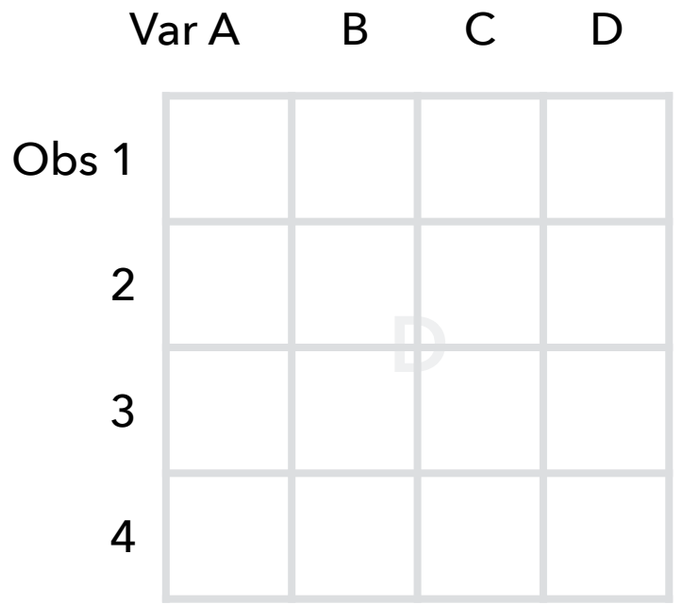
`pandas.DataFrame`

Pandas and Data Frame

`numpy.ndarray`

`pandas.DataFrame`

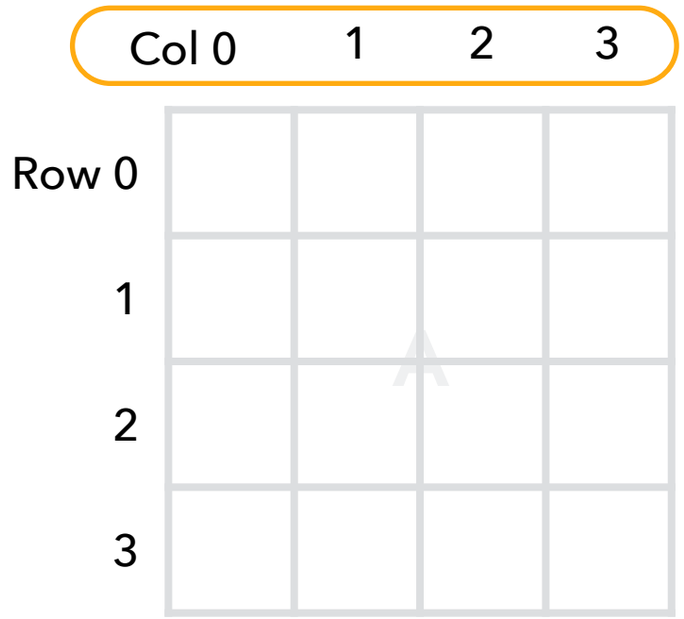


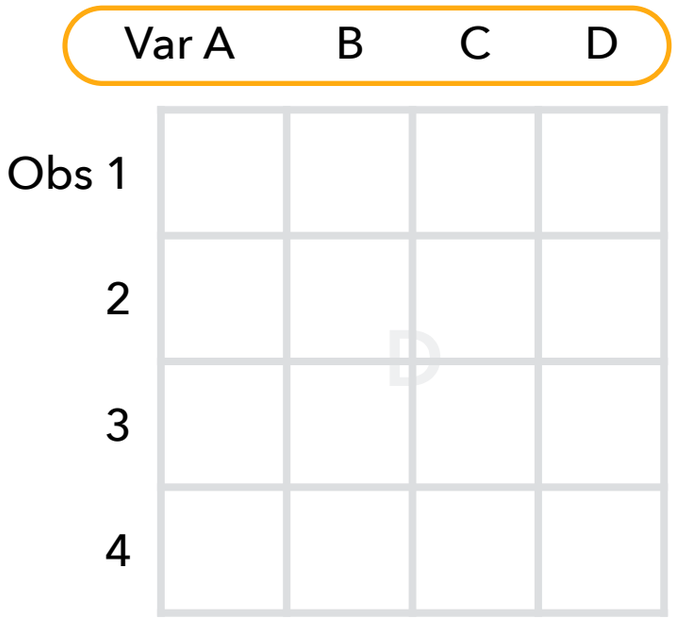


Pandas and Data Frame

`numpy.ndarray`

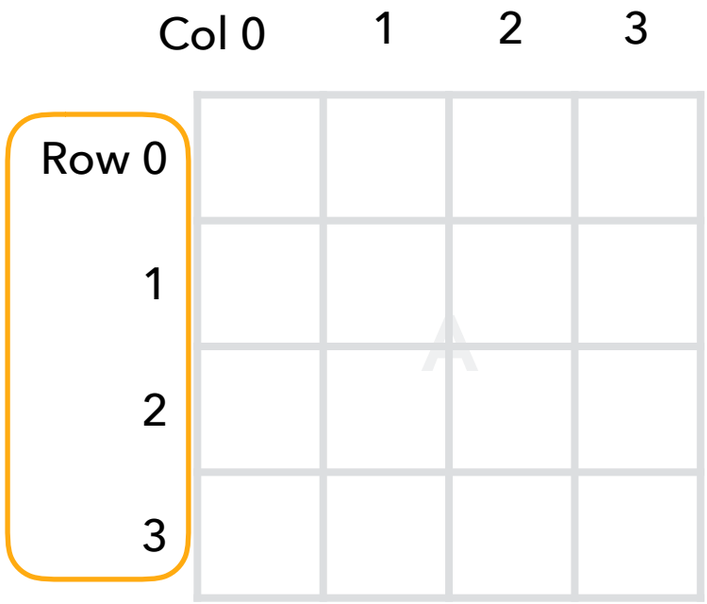
`pandas.DataFrame`



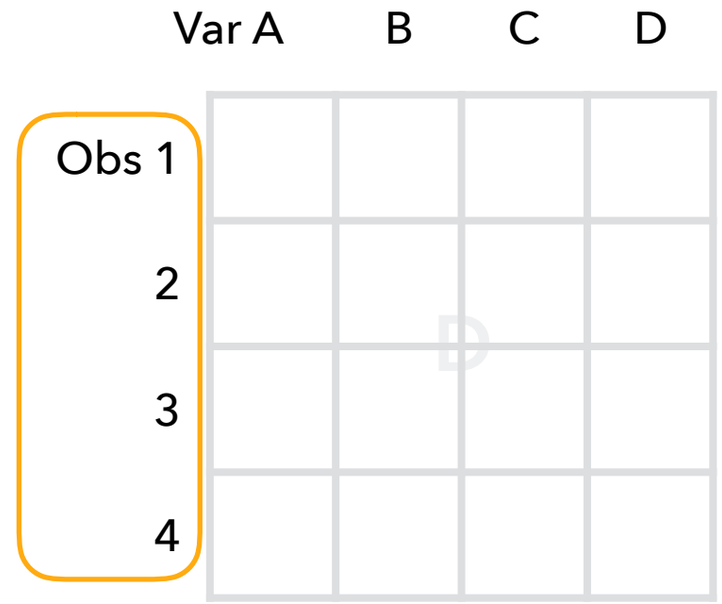


Pandas and Data Frame

`numpy.ndarray`

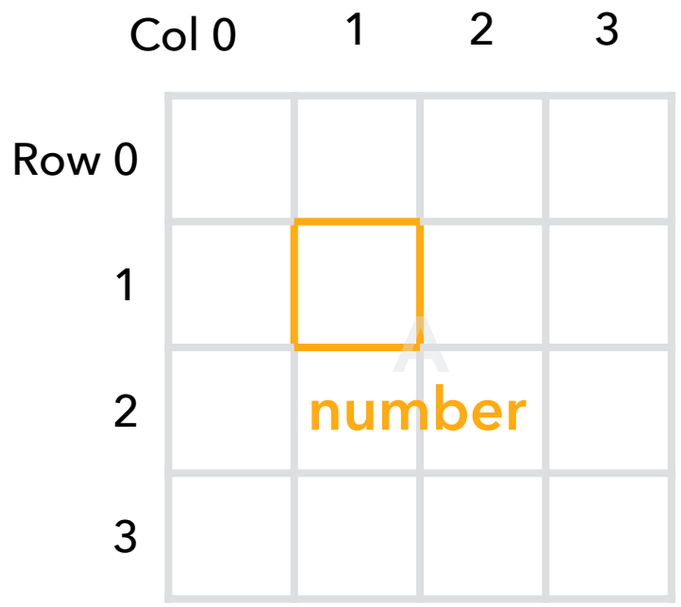


`pandas.DataFrame`

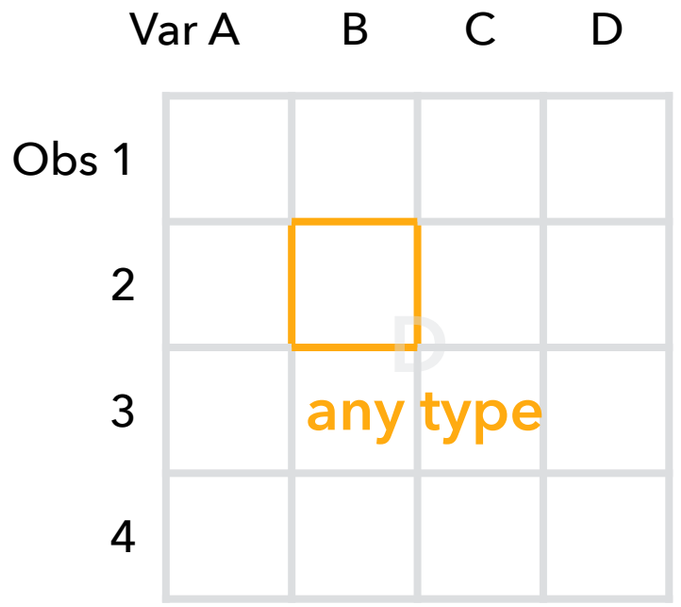


Pandas and Data Frame

`numpy.ndarray`



`pandas.DataFrame`



Pandas and Data Frame

`numpy.ndarray`

Col 0	1	2	3
Row 0			
1	Numerical		
2	Calculation		
3			

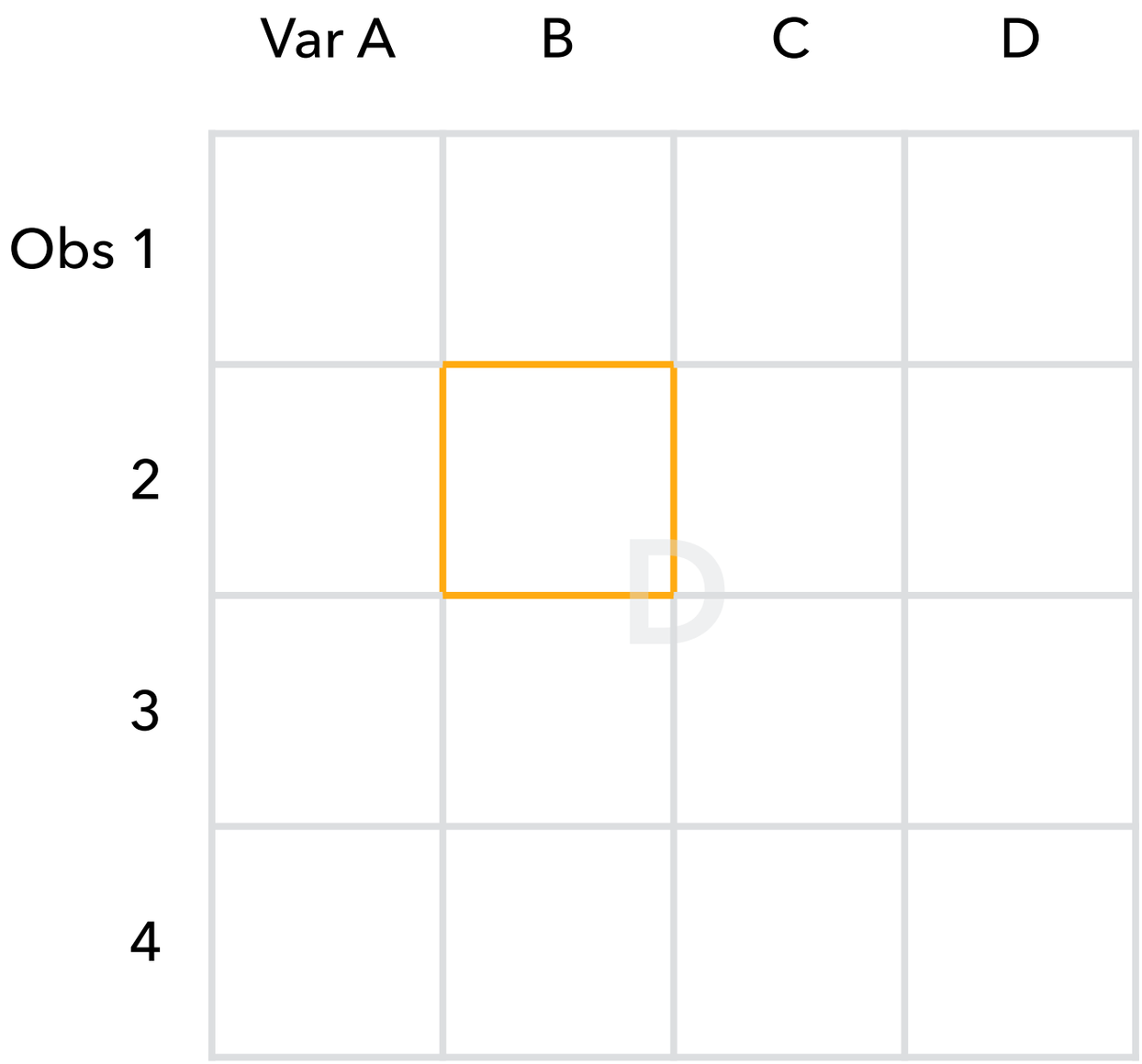
`pandas.DataFrame`

Var A	B	C	D
Obs 1			
2	Data		
3	Analysis		
4			

Index, Slice, Subset

	Var A	B	C	D
Obs 1				
2				
3				
4				

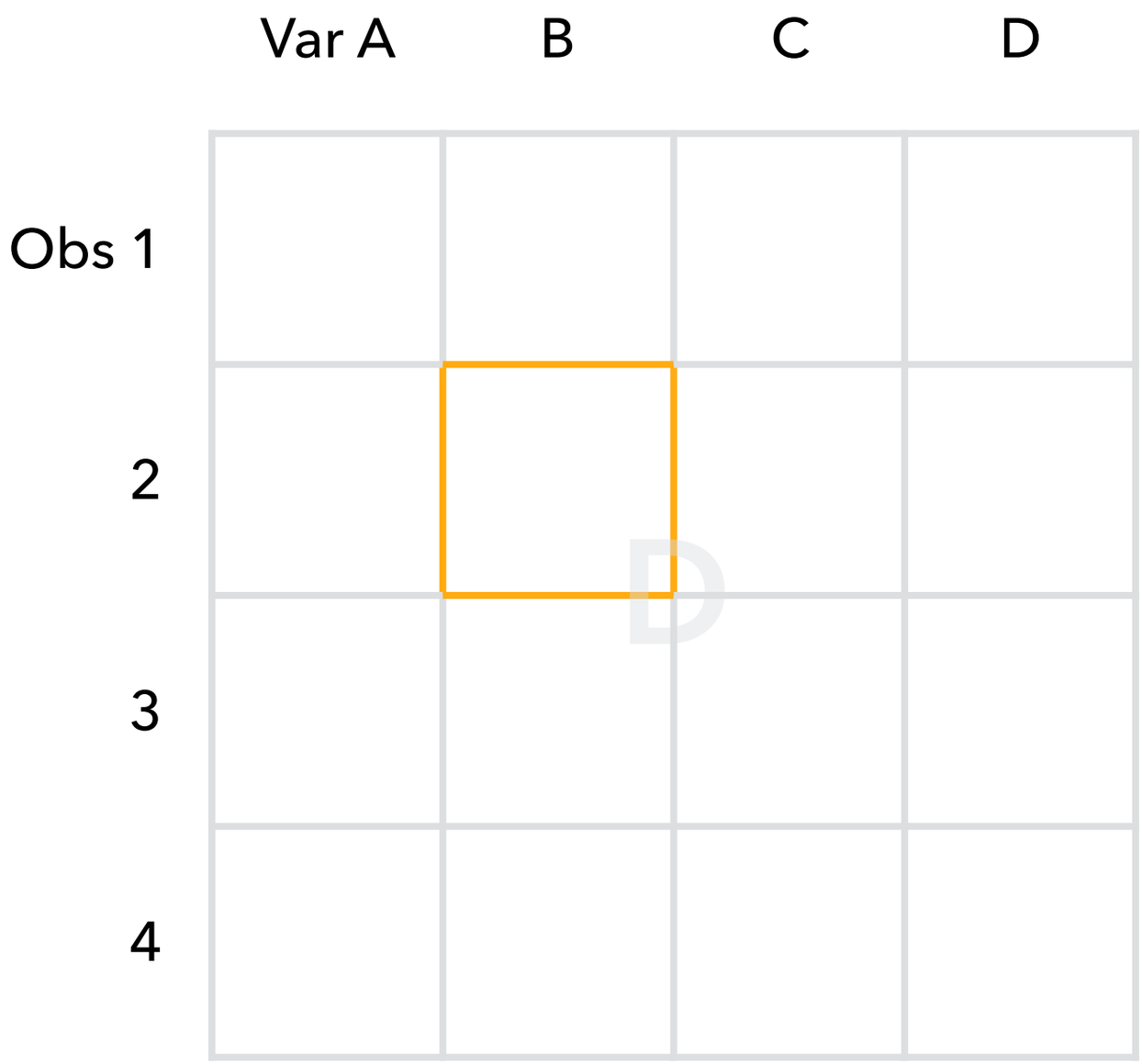
Index, Slice, Subset



Index	[1,1]

Label	[2,B]

Index, Slice, Subset



Index	[1,1]

Label	[2,B]

```
D.loc[2,B]
```

```
D.iloc[1,1]
```

```
D.ix[2,B]
```

Dictionary

	Latitude	Longitude
Los Angeles	34.0207504	-118.6919233
San Luis Obispo	35.2725611	-120.7054056
San Francisco	37.757815	-122.5076402
San Jose	37.2972061	-121.9574961

Dictionary

	Latitude	Longitude
Los Angeles	34.0207504	-118.6919233
San Luis Obispo	35.2725611	-120.7054056
San Francisco	37.757815	-122.5076402
San Jose	37.2972061	-121.9574961

List
Index
Column

Dictionary

Dictionary

Key & Value	Latitude	Longitude
Los Angeles	34.0207504	-118.6919233
San Luis Obispo	35.2725611	-120.7054056
San Francisco	37.757815	-122.5076402
San Jose	37.2972061	-121.9574961

- Create dictionaries for each **column**: `{'key': value}`

Dictionary

	Latitude	Longitude
Los Angeles	34.0207504	-118.6919233
San Luis Obispo	35.2725611	-120.7054056
San Francisco	37.757815	-122.5076402
San Jose	37.2972061	-121.9574961

- Create dictionaries for each **column**: `{'key': value}`
- Use `pd.Series()` and combine them into another dictionary

Dictionary

	Latitude	Longitude
Los Angeles	34.0207504	-118.6919233
San Luis Obispo	35.2725611	-120.7054056
San Francisco	37.757815	-122.5076402
San Jose	37.2972061	-121.9574961

- Create dictionaries for each **column**: `{'key': value}`
- Use `pd.Series()` and combine them into another dictionary
- Use `pd.DataFrame()` to create a data frame

Still Some Problems

	Latitude	Longitude
Los Angeles	34.0207504	-118.6919233
San Luis Obispo	35.2725611	-120.7054056
San Francisco	37.757815	-122.5076402
San Jose	37.2972061	-121.9574961
Melbourne	-37.8274812	144.9352466

Still Some Problems

	Latitude	Longitude
Los Angeles	34.0207504	-118.6919233
San Luis Obispo	35.2725611	-120.7054056
San Francisco	37.757815	-122.5076402
San Jose	37.2972061	-121.9574961
Melbourne	-37.8274812	144.9352466

Flow Control

Function
(Method)

More About Data Frame

Different Data Types

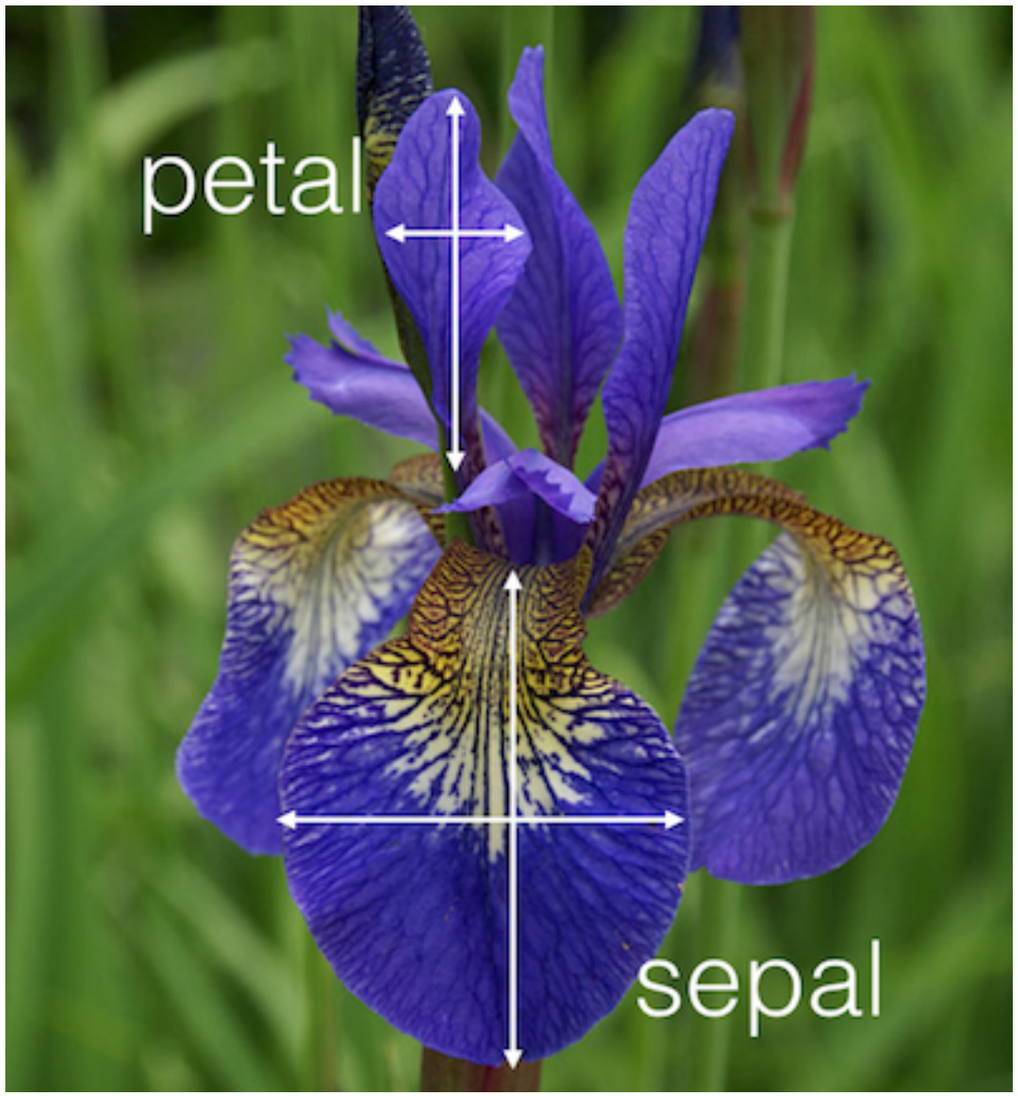
Convenient Methods

- PyData.org
  - 10 Minutes to pandas
  - Merge, join, and concatenate
- *Python Data Science Handbook*
  - Data Indexing and Selection
- Greg Reda
  - Intro to pandas data structures

Lab: Import

Normally we don’t create but import data to our working space, so here’s how you can work like a data scientist.

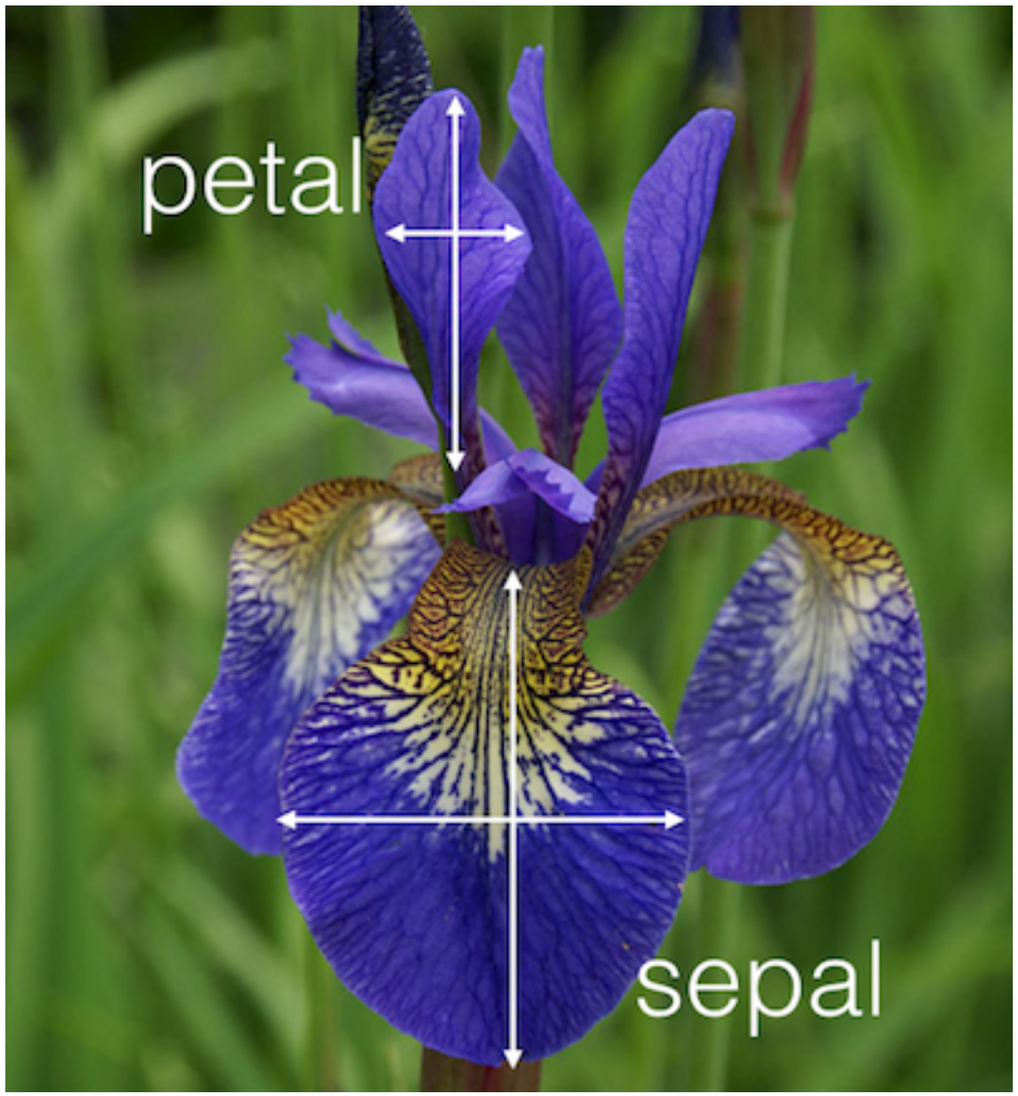
Iris Dataset



- Collected by Edgar Anderson
- Analyzed by Ronald Fisher in his 1936 paper
- Focus on classifying 3 different species of Iris based on their petal and sepal
- Shape: `(150, 5)`

> Photo from Kaggle's *Machine learning first steps with the Iris dataset*

Iris Dataset



- Collected by Edgar Anderson
- Analyzed by Ronald Fisher in his 1936 paper
- Focus on classifying 3 different species of Iris based on their petal and sepal
- Shape: `(150, 5)`
- Use `matplotlib/plotly` for Data Visualization
- Use `scikit-learn` for Data Analysis

> Photo from Kaggle's *Machine learning first steps with the Iris dataset*