機器學習基礎part2

Outline

- Numpy Introduction
- Pandas Introduction
- Matplotlib Introduction
- scikit-learn Introduction
- Different Competition Platform

What’s Numpy

What's Numpy

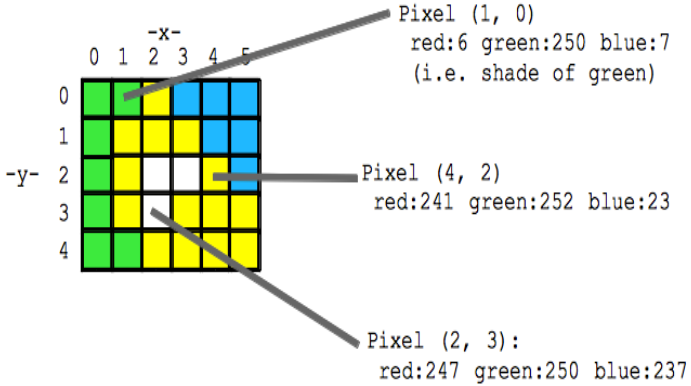
- a library for the Python that support for large, multi-dimensional arrays/matrices
  - http://www.numpy.org/
- core functionality of NumPy is its "ndarray" data structure
  - for *n*-dimensional array
- all elements of a single array must be of the same type

N-dimensional array

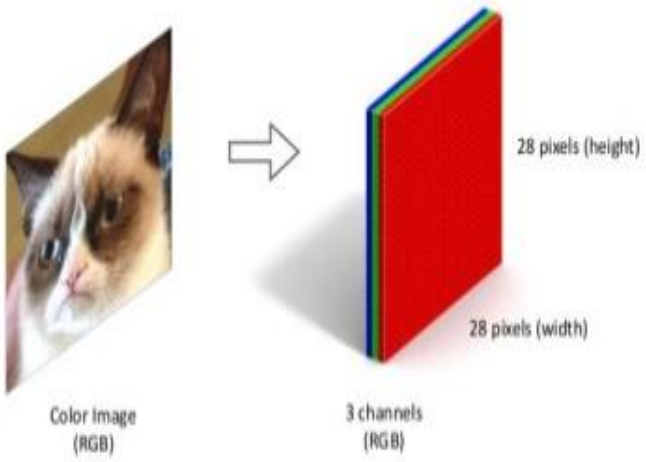
Dimensions	Example	Terminology
1	0 1 2	Vector
2	0 1 2 3 4 5 6 7 8	Matrix
3	0 1 2 3 4 5 6 7 8	3D Array (3rd order Tensor)
N	... ...	ND Array

3D Tensor Example

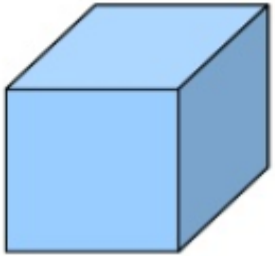
- Each image contain many pixels
  - Each pixels compose red, green, blue(RGB)
- Each channel have brightness levels between 0~255



3D Tensor Example



3D Tensor Example



= **[image width, image height, image channel]**

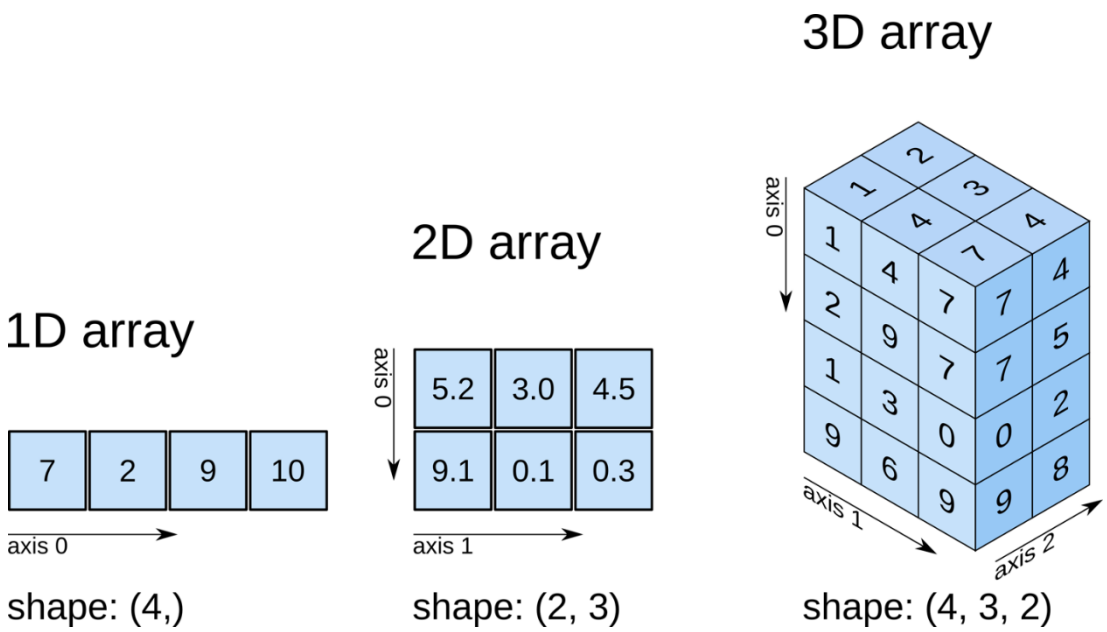
A image is a 3D-tensor

Data type in Numpy

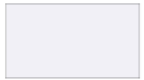
Data type	Description
bool	Boolean (True or False) stored as a byte
int	Platform integer (normally either int32 or int64)
int8	Byte (-128 to 127)
int16	Integer (-32768 to 32767)
int32	Integer (-2147483648 to 2147483647)
int64	Integer (9223372036854775808 to 9223372036854775807)
uint8	Unsigned integer (0 to 255)
uint16	Unsigned integer (0 to 65535)
uint32	Unsigned integer (0 to 4294967295)
uint64	Unsigned integer (0 to 18446744073709551615)

float	Shorthand for float64.
float16	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
float32	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
float64	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
complex	Shorthand for complex128.
complex64	Complex number, represented by two 32-bit floats
complex128	Complex number, represented by two 64-bit floats

What’s axis?


1D array
7
2
9
10
axis 0
shape: (4,)
2D array
5.2
3.0
4.5
9.1
0.1
0.3
axis 0
axis 1
shape: (2, 3)
3D array
axis 0
axis 1
axis 2
shape: (4, 3, 2)

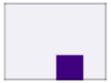
Numpy array creation

Code	Result		Code	Result
`Z = zeros(9)`			`Z = zeros((5,9))`	
`Z = ones(9)`			`Z = ones((5,9))`	

`Z = array(`
`[0,0,0,0,0,0,0,0,0] )`
			`Z = array(`
			`[[0,0,0,0,0,0,0,0,0],`
			`[0,0,0,0,0,0,0,0,0],`
			`[0,0,0,0,0,0,0,0,0],`
			`[0,0,0,0,0,0,0,0,0],`
			`[0,0,0,0,0,0,0,0,0]])`
				

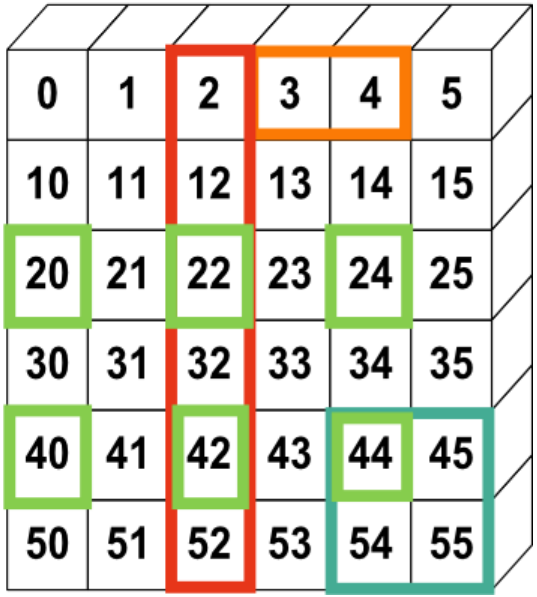
`Z = arange(9)`			`Z = arange(5*9).reshape(5,9)`	
`Z = random.uniform(0,1,9)`			`Z = random.uniform(0,1,(5,9))`	

Numpy array reshape

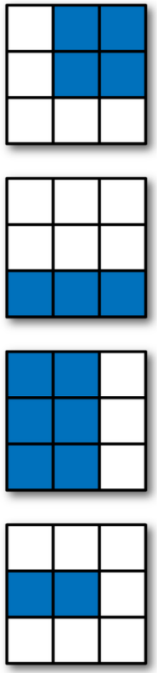
Code	Result	Code	Result
`Z[2,2] = 1`		`Z = Z.reshape(1,12)`	
`Z = Z.reshape(4,3)`		`Z = Z.reshape(12,1)`	
`Z = Z.reshape(6,2)`			
`Z = Z.reshape(2,6)`			

Numpy array indexing/slicing

```
>>> a[0,3:5]
array([3,4])

>>> a[4:,4:]
array([[44, 45],
       [54, 55]])

>>> a[:,2]
array([2,12,22,32,42,52])

>>> a[2::2,::2]
array([[20,22,24]
       [40,42,44]])
```



Numpy array indexing/slicing

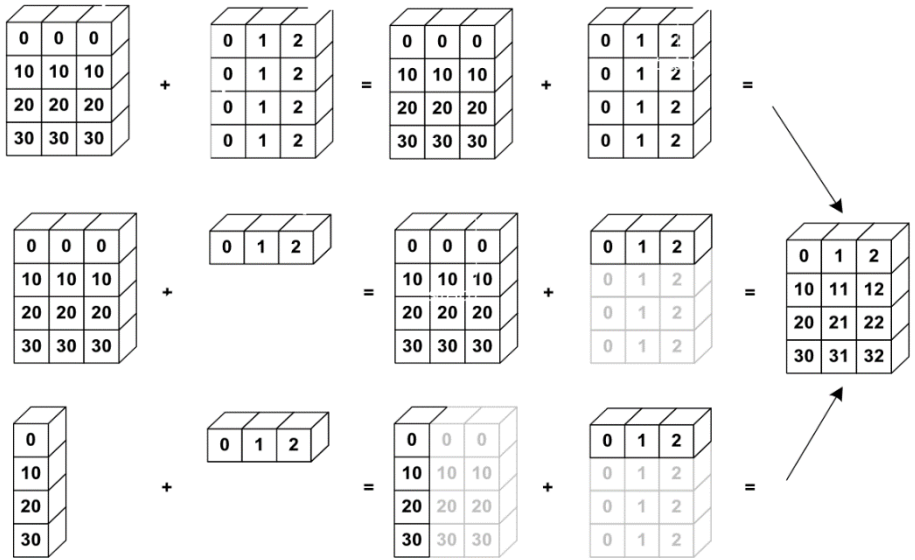


Expression	Shape
`arr[:2, 1:]`	`(2, 2)`
`arr[2]`	`(3,)`
`arr[2, :]`	`(3,)`
`arr[2:, :]`	`(1, 3)`
`arr[:, :2]`	`(3, 2)`
`arr[1, :2]`	`(2,)`
`arr[1:2, :2]`	`(1, 2)`

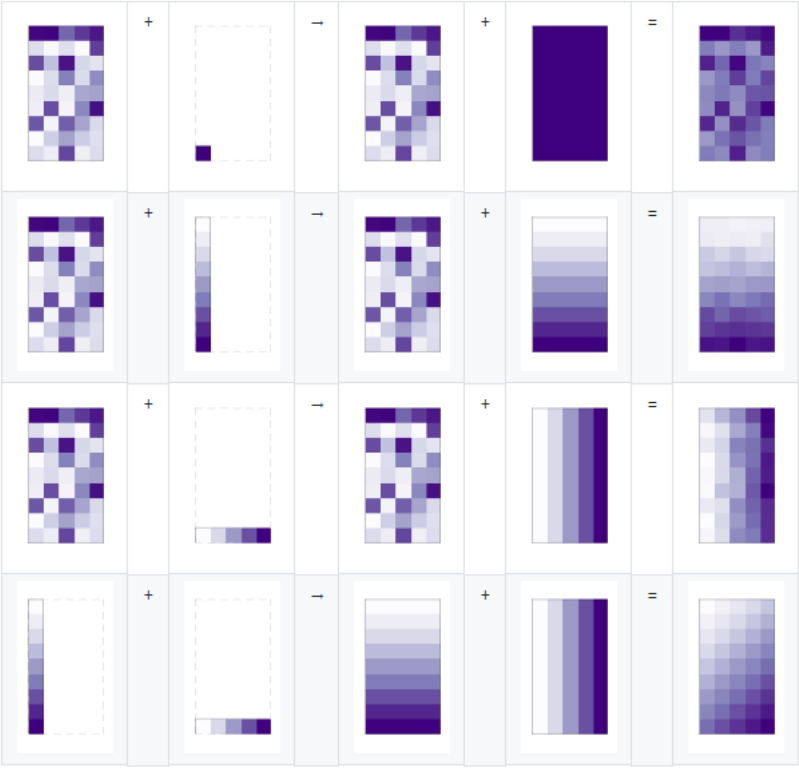
Numpy array indexing/slicing

Code	Result		Code	Result
`Z`			`Z[...] = 1`	
`Z[1,1] = 1`			`Z[:,0] = 1`	
`Z[0,:] = 1`			`Z[2:,2:] = 1`	
`Z[:,::2] = 1`			`Z[::2,:] = 1`	
`Z[:-2,:-2] = 1`			`Z[2:4,2:4] = 1`	
`Z[::2,::2] = 1`			`Z[3::2,3::2] = 1`	

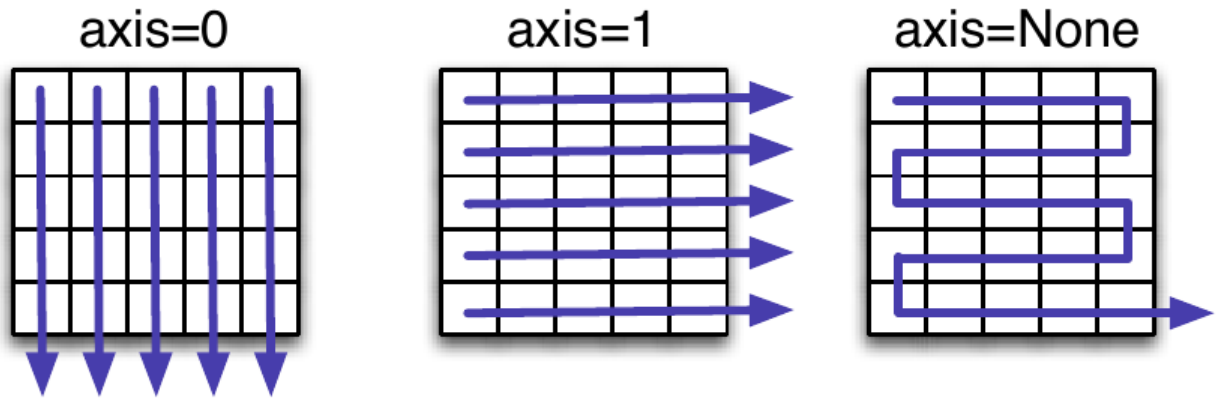
Numpy array broadcasting



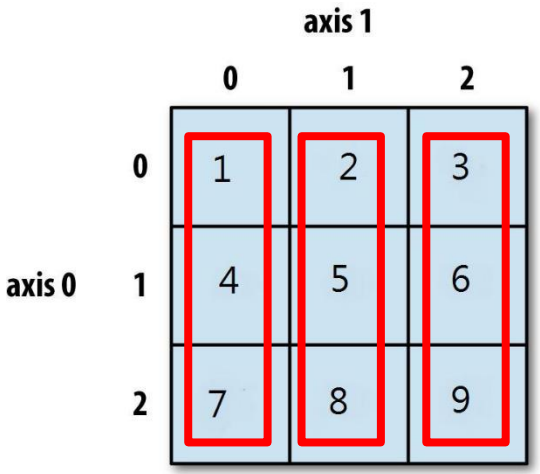
Numpy array broadcasting

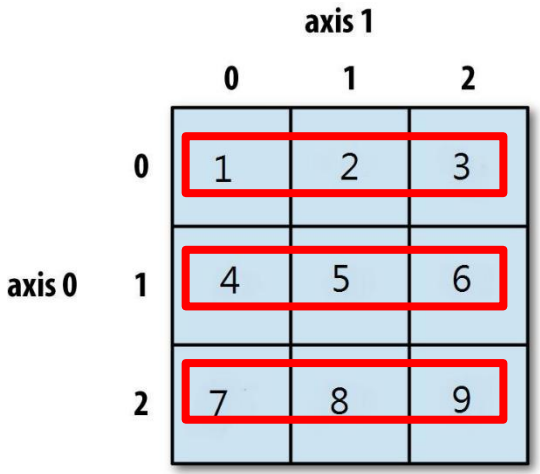


Numpy array operations



Numpy array operations

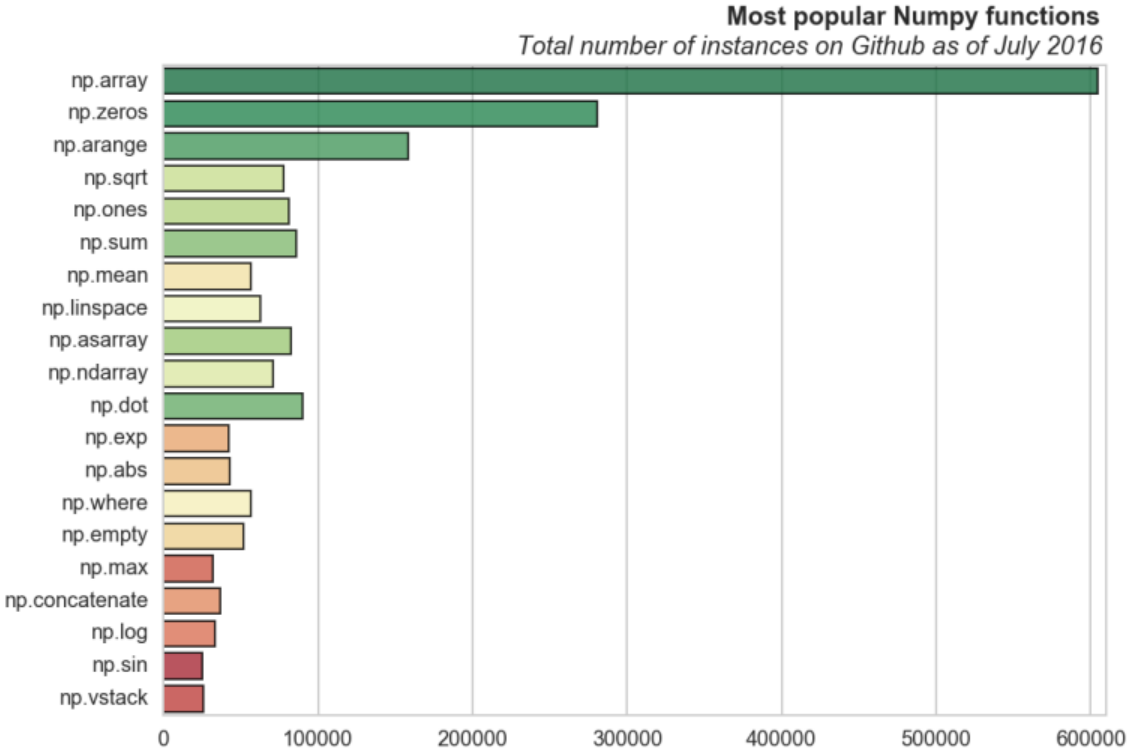




ndarray.sum(axis = 0) -> array([12, 15, 18])

ndarray.sum(axis = 1) -> array([6, 15, 24])

Most popular Numpy functions

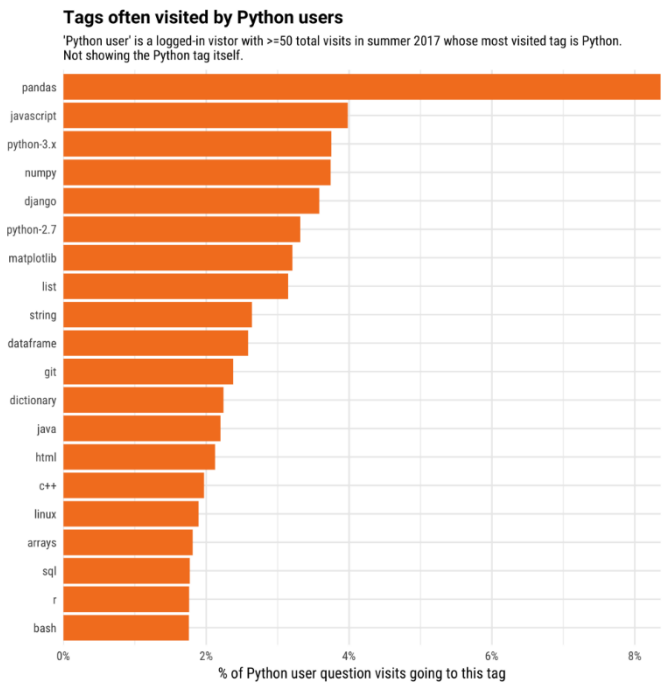


What’s Pandas

What’s Pandas

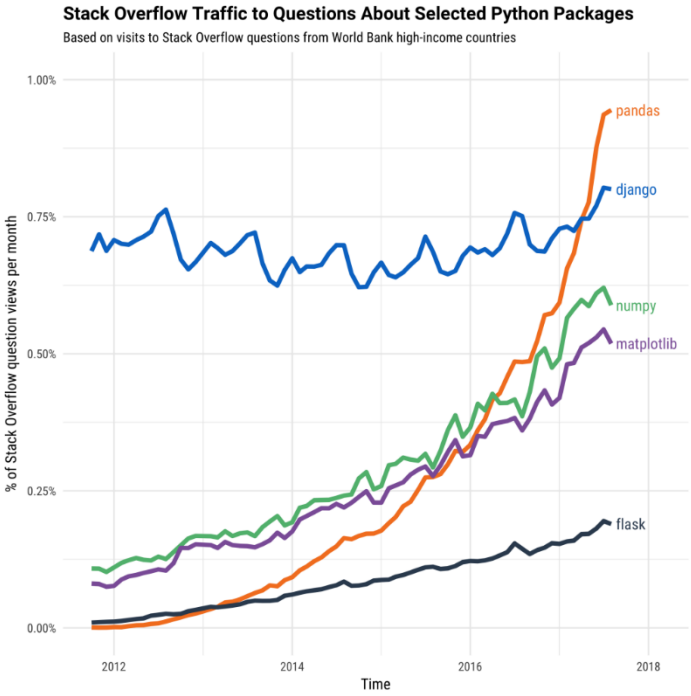
- Pandas is python library that is very useful to manipulate data, especially structure data
- Provide data structures and operations for manipulating numerical tables and time series

Popularity in Pandas



https://stackoverflow.blog/2017/09/14/python-growing-quickly/

Popularity in Pandas



Pandas Data structure

- Pandas series
  - 1-D data
- Pandas dataframe
  - 2-D data

Series

Index	Data
1	‘A’
2	‘B’
3	‘C’
4	‘D’
5	‘E’

Dataframe

	GEOID	State	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	04000US01	Alabama	37150	37952	42212	44476	39980	40933	42590	43464	41381
1	04000US02	Alaska	55891	56418	62993	63989	61604	57848	57431	63648	61137
2	04000US04	Arizona	45245	46657	47215	46914	45739	46896	48621	47044	50602
3	04000US05	Arkansas	36658	37057	40795	39586	36538	38587	41302	39018	39919
4	04000US06	California	51755	55319	55734	57014	56134	54283	53367	57020	57528

Dataframe is consists of rows or columns

Create dataframe

Dictionary

Row Oriented

```
sales = [{'account': 'Jones LLC', 'Jan': 150, 'Feb': 200, 'Mar': 140},
         {'account': 'Alpha Co', 'Jan': 200, 'Feb': 210, 'Mar': 215},
         {'account': 'Blue Inc', 'Jan': 50, 'Feb': 90, 'Mar': 95 }]
df = pd.DataFrame(sales)
```

default

List

```
sales = [('Jones LLC', 150, 200, 50),
         ('Alpha Co', 200, 210, 90),
         ('Blue Inc', 140, 215, 95)]
labels = ['account', 'Jan', 'Feb', 'Mar']
df = pd.DataFrame.from_records(sales, columns=labels)
```

from_records

	account	Jan	Feb	Mar
0	Jones LLC	150	200	140
1	Alpha Co	200	210	215
2	Blue Inc	50	90	95

Column Oriented

```
sales = {'account': ['Jones LLC', 'Alpha Co', 'Blue Inc'],
         'Jan': [150, 200, 50],
         'Feb': [200, 210, 90],
         'Mar': [140, 215, 95]}
df = pd.DataFrame.from_dict(sales)
```

from_dict

```
sales = [('account', ['Jones LLC', 'Alpha Co', 'Blue Inc']),
         ('Jan', [150, 200, 50]),
         ('Feb', [200, 210, 90]),
         ('Mar', [140, 215, 95]) ]
df = pd.DataFrame.from_items(sales)
```

from_items

When using a dictionary, column order is not preserved.
Explicitly order them:
`df = df[['account', 'Jan', 'Feb', 'Mar']]`

Practical Business Python - pbpython.com

DataFrame Basic Functionality

Function	Description
count()	Number of non-null observations
sum()	Sum of values
mean()	Mean of Values
median()	Median of Values
mode()	Mode of values
std()	Standard Deviation of the Values
min()	Minimum Value
max()	Maximum Value
abs()	Absolute Value
prod()	Product of Values
cumsum()	Cumulative Sum
cumprod()	Cumulative Product

What's CSV file

- comma-separated values (CSV) file stores tabular data (numbers and text) in plain text
- Each line of the file is a data record
- Each record consists of one or more fields, separated by commas

What’s CSV file

	A	B	C
1	student name	age	score
2	isaac	18	100
3	kevin	20	70
4	jack	15	90
5			
6			

```
student name, age, score
isaac, 18, 100
kevin, 20, 70
jack, 15, 90
```

Json form

- Json(JavaScript Object Notation) include two symbol
  - {} (object)
  - [] (Array)

key-value pair

{"subject":"Math","score":80}

object

["Tom", "John", "Amy", "Ivy"]

array

Json form

Name	Tom Chen
Math	80
English	90

Name	Amy Lin
Math	86
English	88

[{"name":"Tom Chen","report":[{"subject":"Math","score":80},{"subject":"English","score":90}]},
{"name":"Amy Lin","report":[{"subject":"Math","score":86},{"subject":"English","score":88}]}]

Pandas read/write

File format	Read method	Write method
CSV	read_csv	to_csv
JSON	read_json	to_json
HTML	read_html	to_html
…	…	…
…	…	…

Pandas merge

LEFT	key	A	B
0	K0	A0	B0
1	K1	A1	B1
2	K2	A2	B2
3	K3	A3	B3

RIGHT	key	C	D
0	K0	C0	D0
1	K1	C1	D1
2	K1	C2	D2
3	K4	C3	D3

Left Merge

RESULT	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K1	A1	B1	C2	D2
3	K2	A2	B2	NaN	NaN
4	K3	A3	B3	NaN	NaN

Right Merge

RESULT	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K1	A1	B1	C2	D2
3	K4	NaN	NaN	C3	D3

Inner Merge

RESULT	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K1	A1	B1	C2	D2

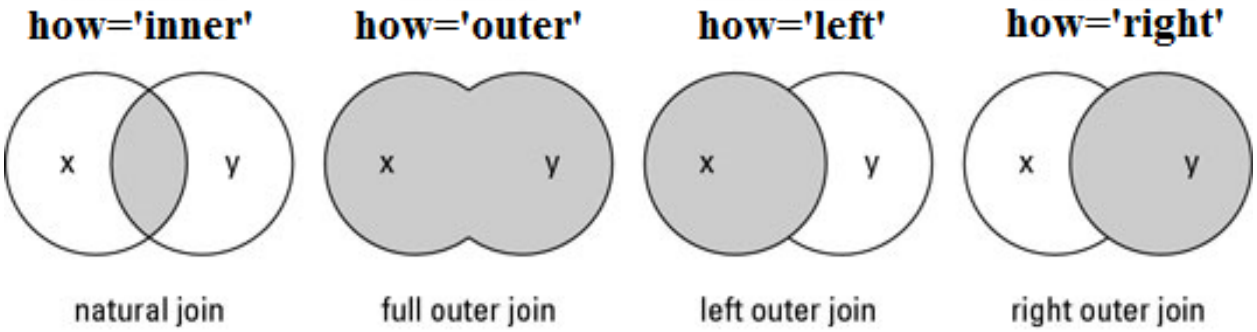
Outer Merge

RESULT	key	A	B	C	D
0	K0	A0	B0	C0	D0
1	K1	A1	B1	C1	D1
2	K1	A1	B1	C2	D2
3	K2	A2	B2	NaN	NaN
4	K3	A3	B3	NaN	NaN
5	K4	NaN	NaN	C3	D3

Pandas merge

Merge method	SQL Join Name	Description
left	LEFT OUTER JOIN	Use keys from left frame only
right	RIGHT OUTER JOIN	Use keys from right frame only
outer	FULL OUTER JOIN	Use union of keys from both frames
inner	INNER JOIN	Use intersection of keys from both frames

Pandas merge



Pandas append

df1

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

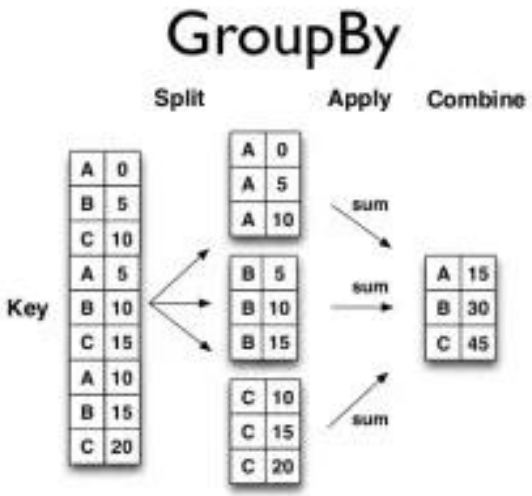
df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

Result

	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	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

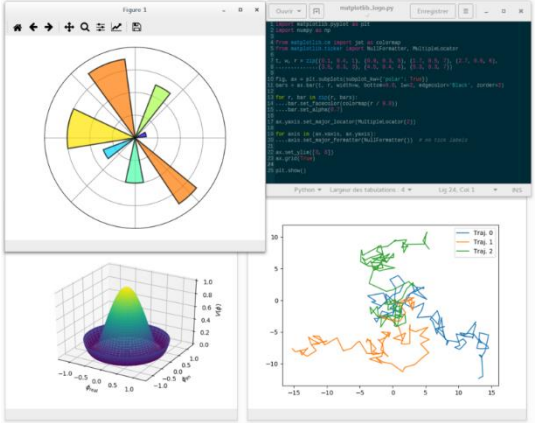
Pandas groupby



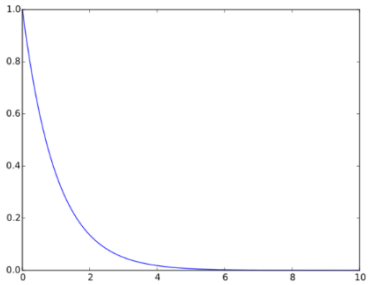
What’s Matplotlib

What’s matplotlib

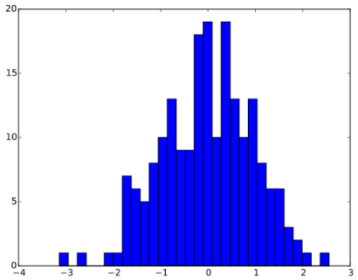
- A plotting library for the Python
  - https://matplotlib.org/
- Numerical mathematics extension NumPy
- Matplotlib 1.2 is the first version of matplotlib to support Python 3.x



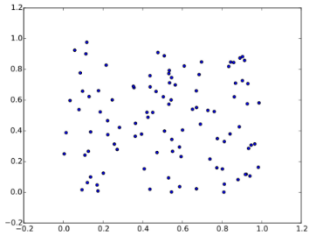
What’s matplotlib



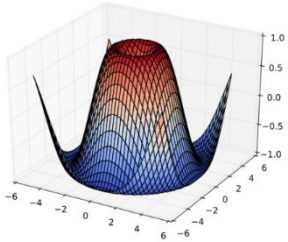
line plot



histogram



scatter plot



3D plot

What's matplotlib

- more example
  - https://matplotlib.org/gallery.html

Figure

- A figure is the windows in the GUI that has “Figure #” as title.
  - figures are numbered starting from 1
  - several parameters that determine what the figure looks like:

Argument	Default	Description
num	1	number of figure
figsize	figure.figsize	figure size in in inches (width, height)
dpi	figure.dpi	resolution in dots per inch
facecolor	figure.facecolor	color of the drawing background
edgecolor	figure.edgecolor	color of edge around the drawing background
frameon	True	draw figure frame or not

Figure

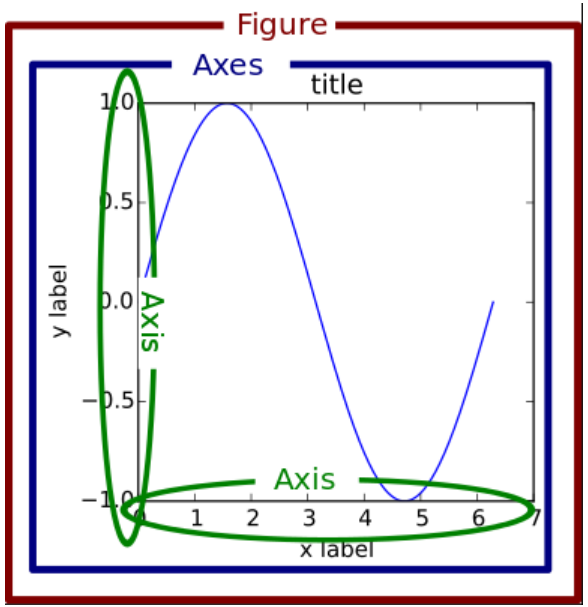
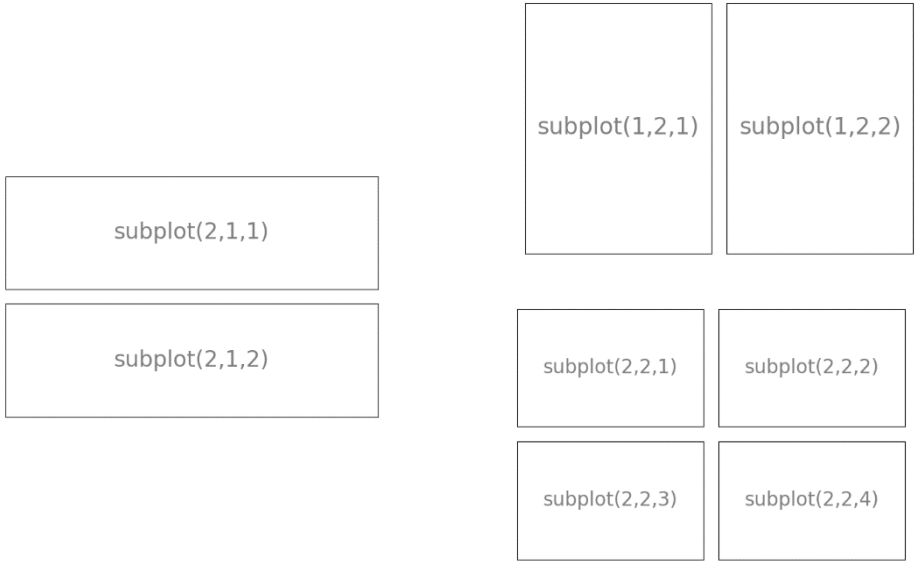


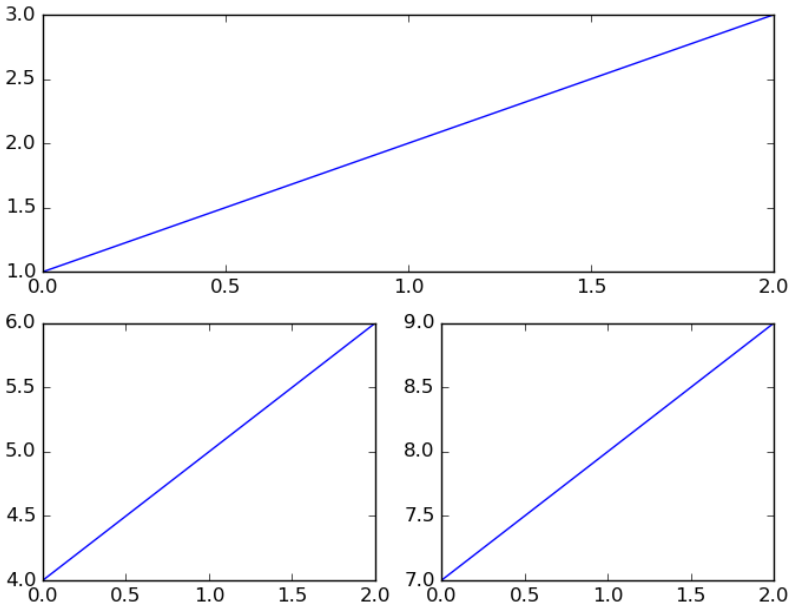
Figure
Axes
title
1.0
0.5
0.0
−0.5
−1.0
y label
Axis
Axis
0
1
2
3
4
5
6
7
x label

Subplot

- Subplot allow user to arrange plots in a regular grid
  - need to specify # of rows/columns and # of the plot

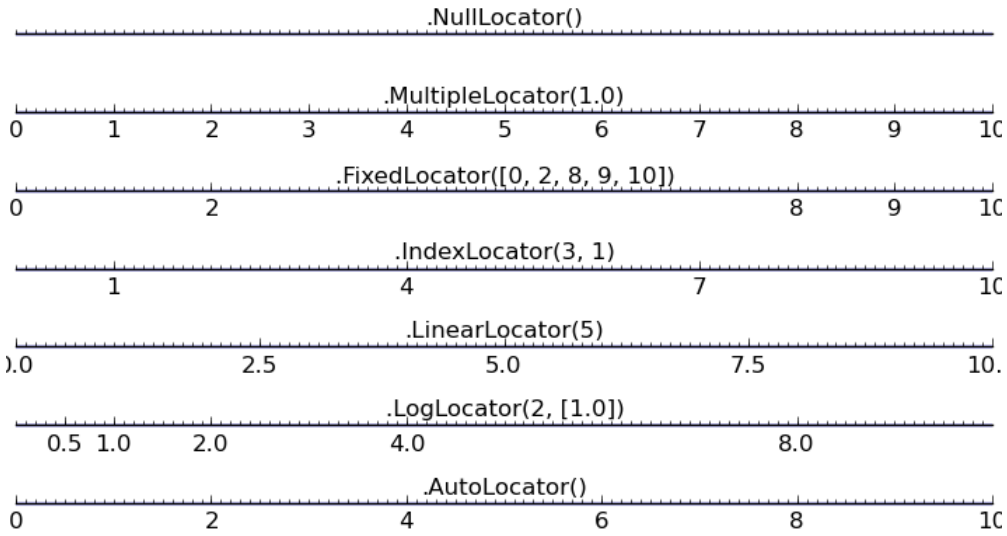


Subplot



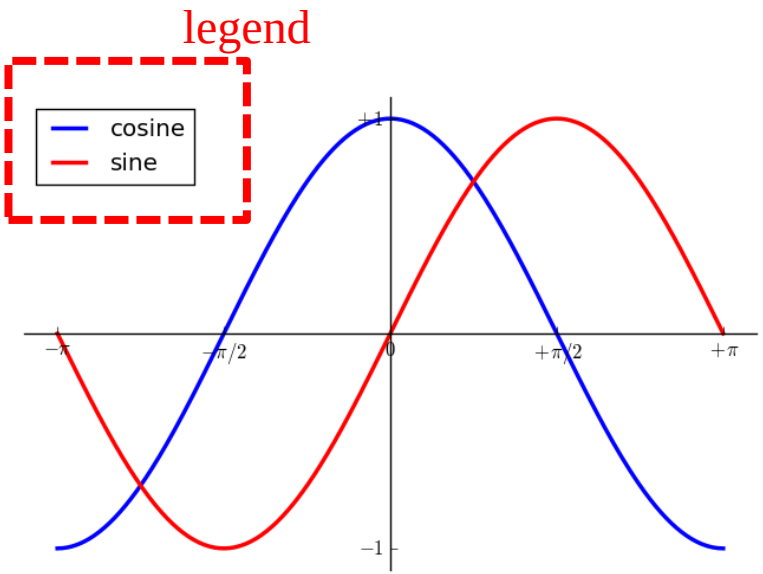
Ticks

- Ticks help specify where ticks should appear and tick formatters to give ticks the appearance you want



Legend

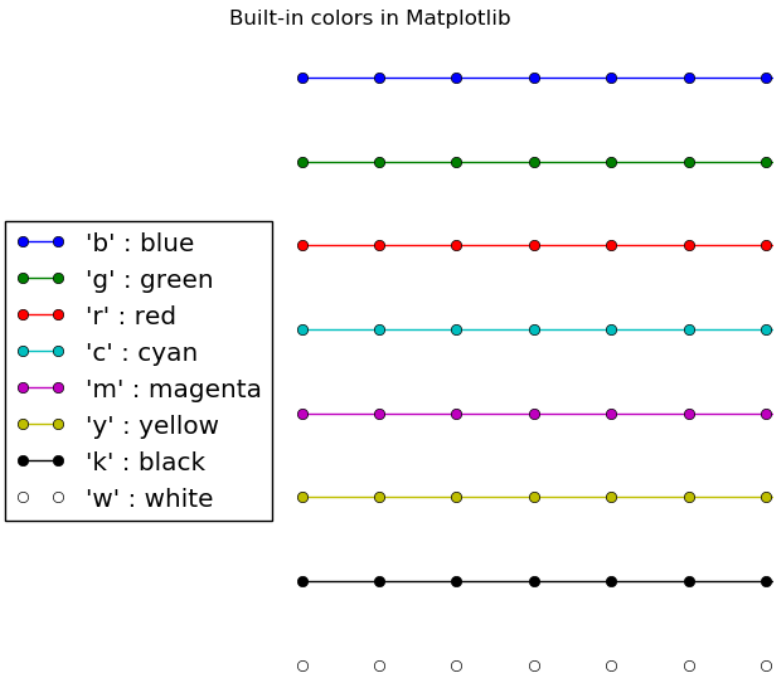
- Lengend help user to describe figure easily



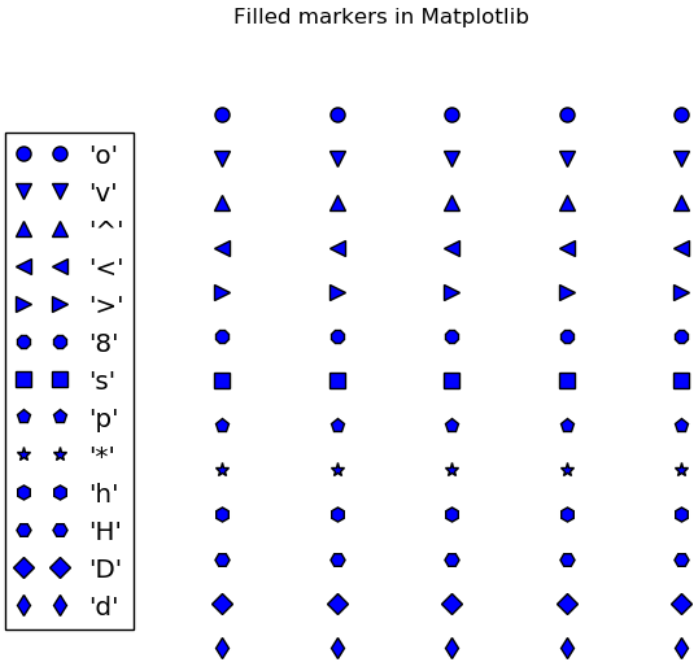
Line property

Property	Description	Appearance
alpha (or a)	alpha transparency on 0-1 scale	
antialiased	True or False - use antialised rendering	Aliased Anti-aliased
color (or c)	matplotlib color arg	
linestyle (or ls)	see Line properties	
linewidth (or lw)	float, the line width in points	
solid_capstyle	Cap style for solid lines	
solid_joinstyle	Join style for solid lines	
dash_capstyle	Cap style for dashes	
dash_joinstyle	Join style for dashes	
marker	see Markers	
markeredgewidth (mew)	line width around the marker symbol	
markeredgecolor (mec)	edge color if a marker is used	
markerfacecolor (mfc)	face color if a marker is used	
markersize (ms)	size of the marker in points	

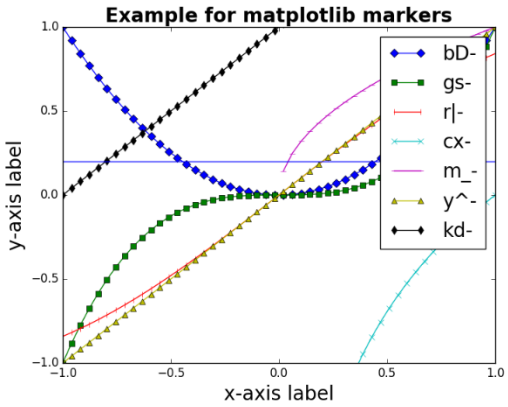
Color

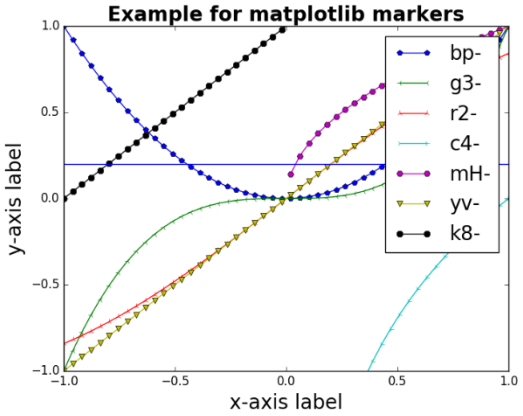

Built-in colors in Matplotlib
'b' : blue
'g' : green
'r' : red
'c' : cyan
'm' : magenta
'y' : yellow
'k' : black
'w' : white

Marker

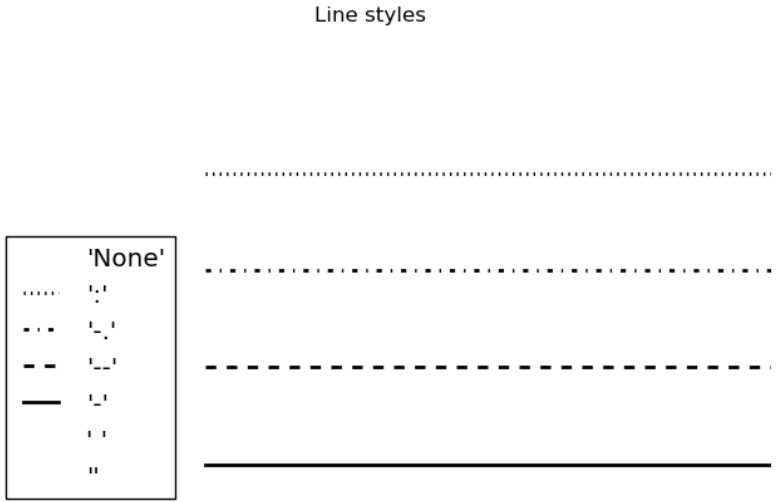


Line style Example





Line style

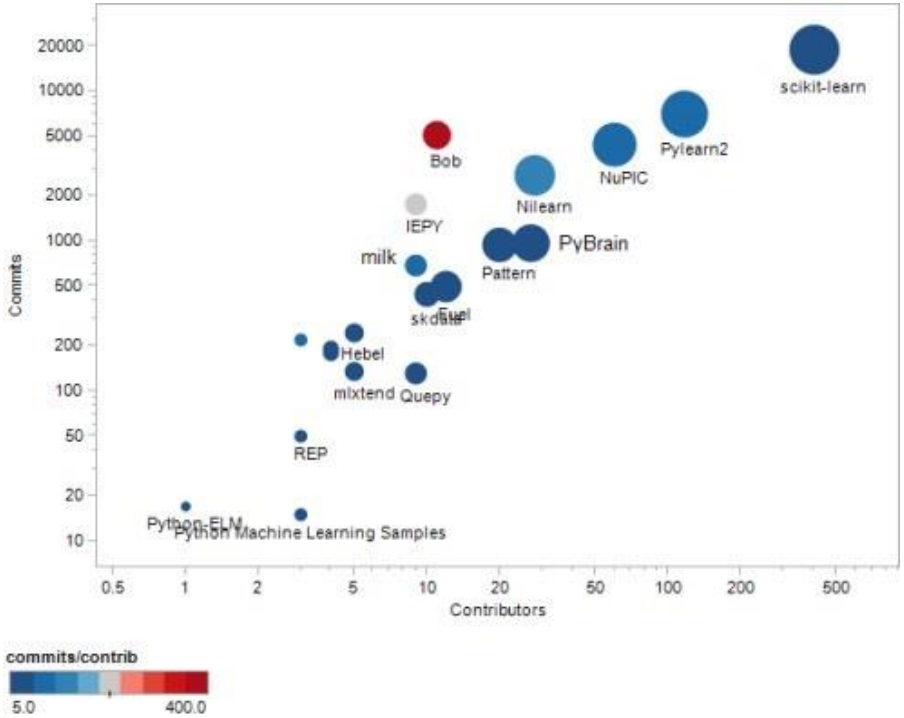


What’s scikit-learn Introduction

What's Scikit-learn

- scikit-learn is a machine learning library for python programming language
  - http://scikit-learn.org/stable/
  - https://github.com/scikit-learn/scikit-learn
- support many famous machine learning algorithm
  - classification, regression, clustering

Machine Learning Open Source



Different Competition Platform

What's Kaggle

- Kaggle is a platform that statisticians and data miners compete to produce the best models for predicting
  - https://www.kaggle.com/
- Datasets are uploaded by companies and users
- Google acquire Kaggle on 8 March 2017

kaggle

What's "天池"

- Tianchi(天池) is chinese version of kaggle
  - https://tianchi.aliyun.com/index.htm?spm=5176.100066.5610778.10.5198d780qaVmpq
- A data platform hosted by Alibaba Cloud

TIANCHI天池
大数据科研平台