Workshop - Text Mining

with Python (Part2)

Kanda Tiwatthanont @ TNI

Wed 17 and Mon 22 May 2017

Agenda - Day 1

- Part 1 : Introduction (10.00 - 11.00)
    - What is Data Mining ?
    - Text Mining -- Social Mind Extraction
- Part 2 : Python (11.00 - 12.00 / 13.00 - 14.00)
    - Python Introduction
    - **Anaconda** Installation (Data Science Distribution of Python)
    - **Jupyter** Introduction (Next Generation Engineering Notebook)
        - “Hello World!” in Jupyter, **and so on**.
- Part3 : Pandas / Seaborn (14.00 - 15.00)
    - **Pandas** (Structured Data Analysis Tool)
    - **Seaborn** (Statistical Data Visualization)

Agenda - Day 2

- Part 4 : Data Mining Framework (10.00 - 12.00)
    - Framework Overview
    - Scikit-learn -- Machine Learning Tool for Data Scientist
    - Data Prediction Hands-on

- Part 5 : Sentiment Analysis (13.00 - 15.00)
    - Introduction Text Mining
    - Unstructured to Structured Data
    - Text Classification

Workshop
Part 2 - Python

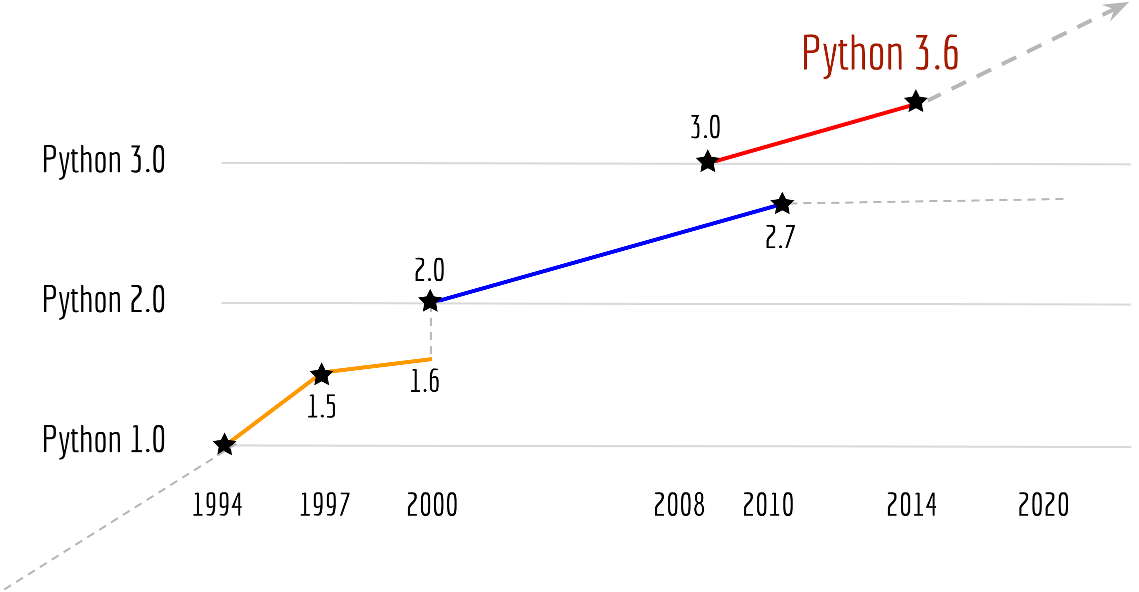
Kanda Tiwatthanont @ TNI

python™

- Python is a widely used high-level programming language
- Python is an interpreted language
- Python has a design philosophy which emphasizes code readability
- Python supports memory management
- A bundle of software to be installed

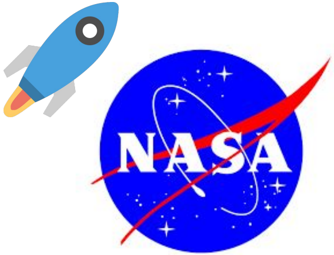
History of Python

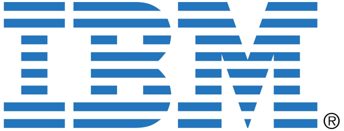
Python Data Analytics



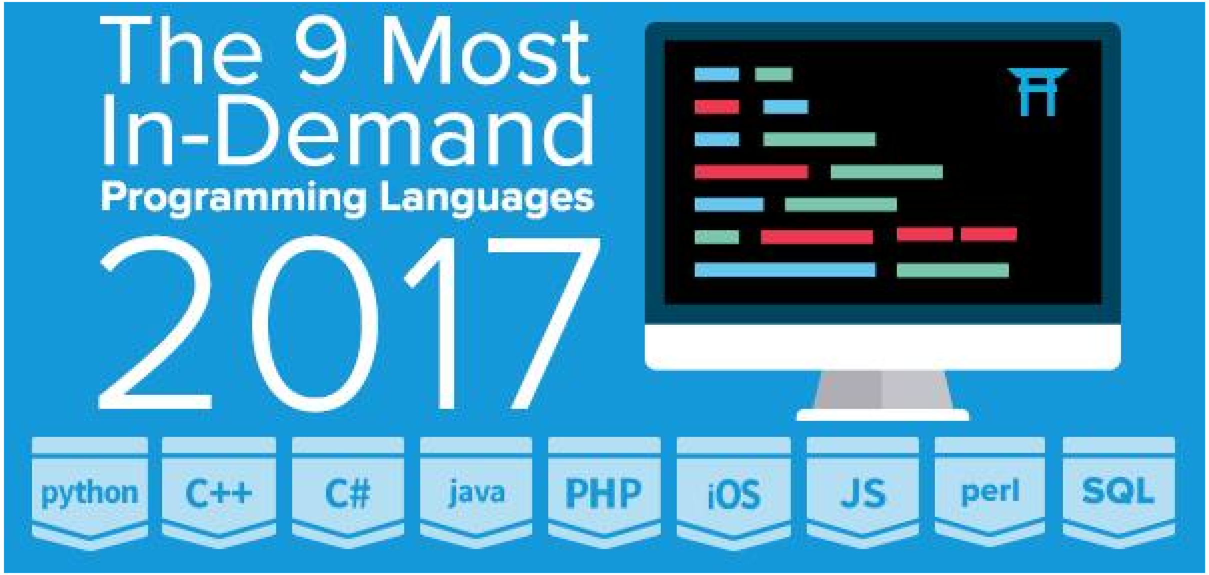
Guido Van Rossum, **1989**

Organizations Using Python





What the most demand programming language ?



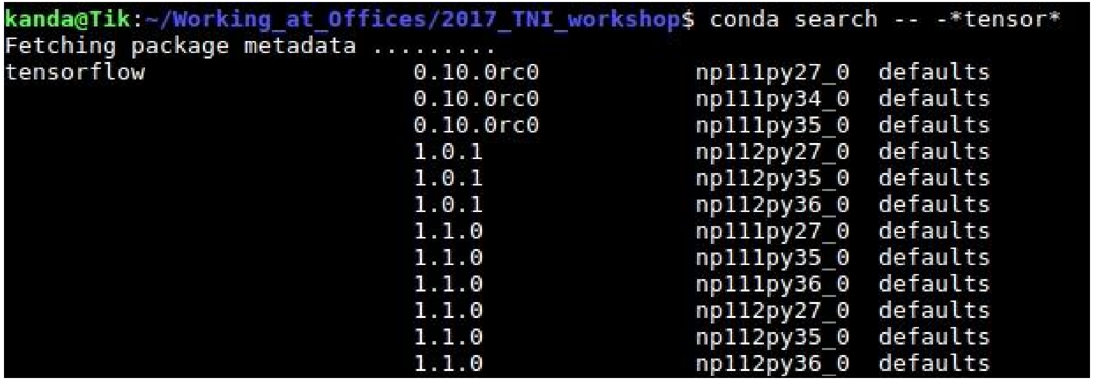
Part 2

Workshop
Python

python™

- Dynamic language with **Interpreter**,
- Numerous contributed additional **packages** (libraries),
- Bundled with **'pip'** package manager.

Highly recommend installing Anaconda. **Anaconda** conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.



```
kanda@Tik:~/Working_at_Offices/2017_TNI_workshop$ conda search -- -*tensor*
Fetching package metadata .........
tensorflow                   0.10.0rc0            np111py27_0  defaults
                             0.10.0rc0            np111py34_0  defaults
                             0.10.0rc0            np111py35_0  defaults
                             1.0.1                np112py27_0  defaults
                             1.0.1                np112py35_0  defaults
                             1.0.1                np112py36_0  defaults
                             1.1.0                np111py27_0  defaults
                             1.1.0                np111py35_0  defaults
                             1.1.0                np111py36_0  defaults
                             1.1.0                np112py27_0  defaults
                             1.1.0                np112py35_0  defaults
                             1.1.0                np112py36_0  defaults
```

jupyter

The Jupyter Notebook is an **interactive computing** environment that enables users to author notebook documents that include: Live code, Interactive widgets, Plots, Narrative text.

- **1980** -- Python was born ..
- **2005** -- First notebook system was found, but NOT succeeded.
- **2011** -- IPython Notebook team got awards.
- **2013** -- Funded by the Alfred P. Sloan Foundations. Renamed to Jupyter

jupyter

Kanda Tiwatthanont @ TNI

Jupyter Notebook

Overview

- System and Kernel
- Cells or Element
- Shortcut Keys
- Hand-on: Let's make the Journal

Workshop

Part 3 - Python Package

Matplotlib & Pandas & Seaborn

Kanda Tiwatthanont @ TNI

Python Packages

Matplotlib Package

- plot sample data
- plot sample data with labelled
- very quick data analysis

Pandas Package

- Series Data vs. DataFrame
- Import csv file
- Statistics with Pandas

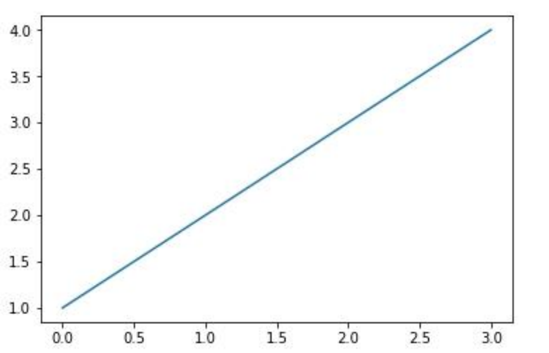
Seaborn Package

- Plot graph with Seaborn

Matplotlib Package

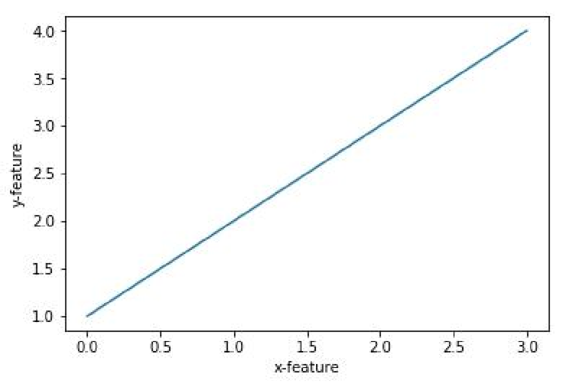
Matplotlib Package

```
…

import matplotlib.pyplot as plt

plt.plot([1,2,3,4])
```

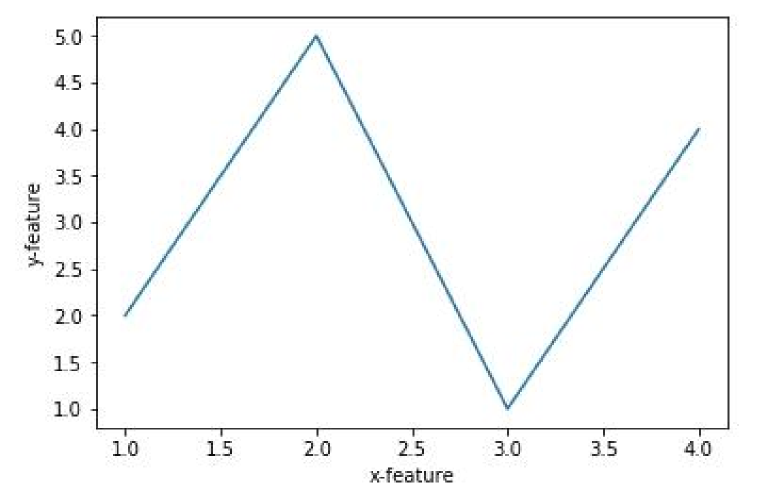


Matplotlib Package

```
…

import matplotlib.pyplot as plt

plt.plot([1,2,3,4])

…
```

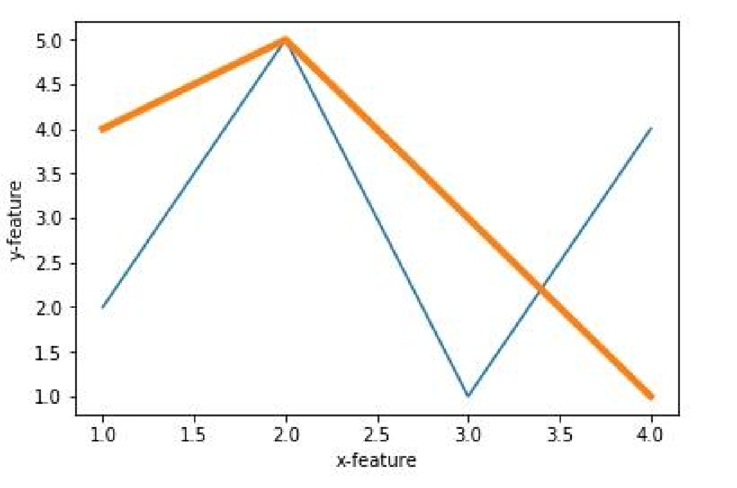


Matplotlib Package



plt.plot(...   )

Matplotlib Package

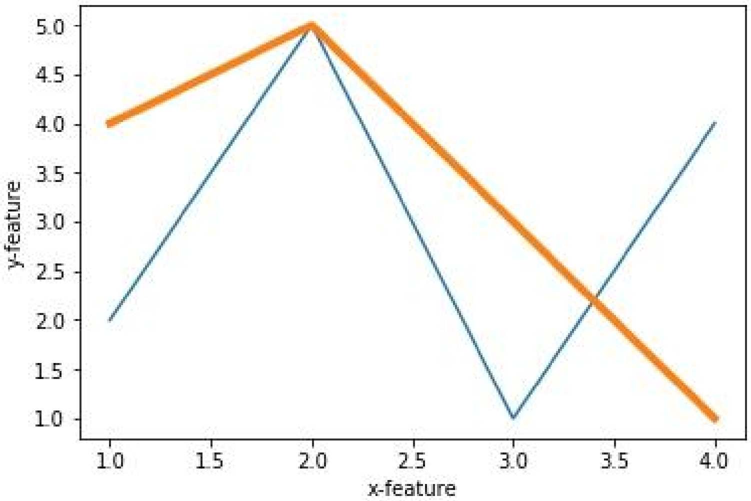


Line # 1

Line # 2

Parameter of pyplot

Matplotlib Package



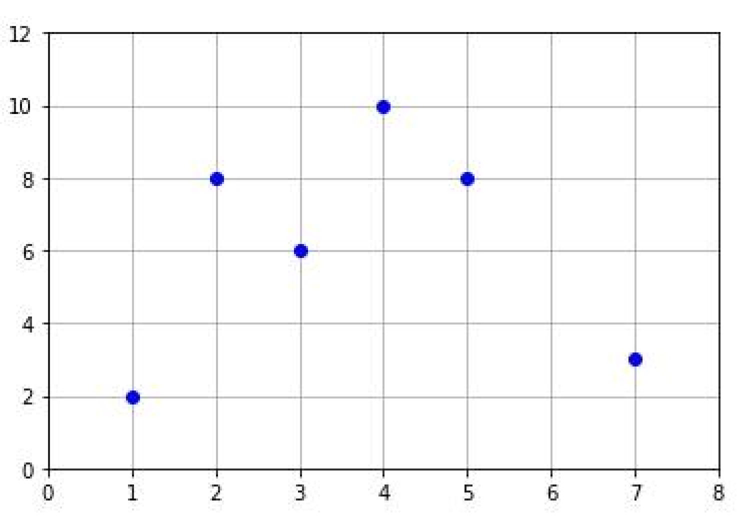
```
plt.plot( [1,2,3,4], [2,5,1,4] )


plt.plot( [1,2,3,4], [4,5,3,1] )
```

```
plt.plot(

            [1,2,3,4], [4,5,3,1],'r-',
            [1,2,3,4], [2,5,1,4], 'g-'

          )
```

Parameter → color & linestyle

help(plt)

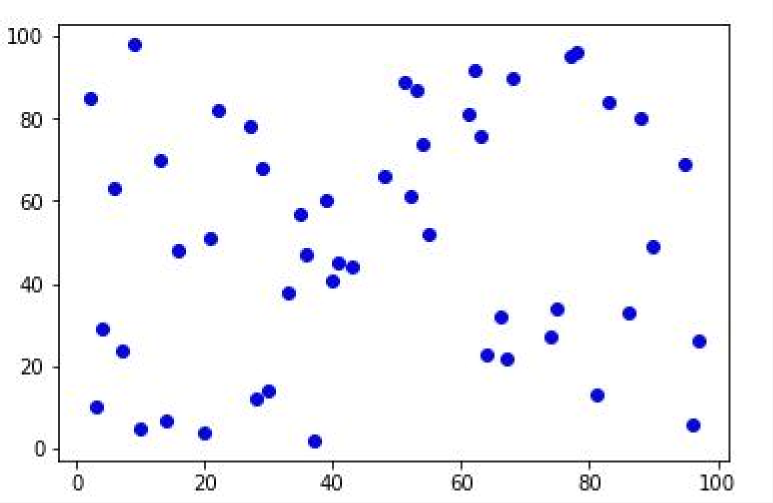
Matplotlib Package



```
plt.plot( [1,2,3,4,5,7], [2,8,6,10,8,3] , ...)
```

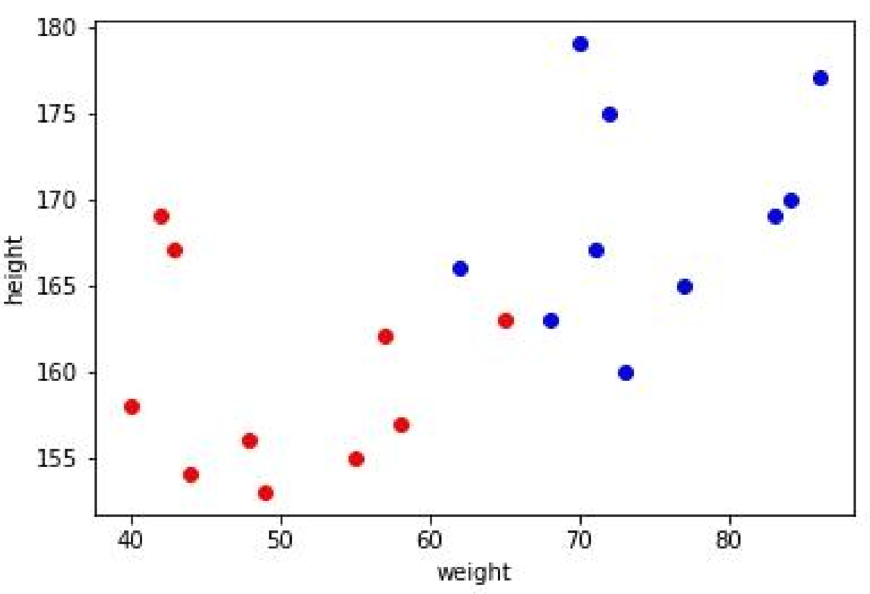
Matplotlib Package

```
import random as rd

x = rd.sample(range(1,100),50)

y = rd.sample(range(1,100),50)
```



Matplotlib Package

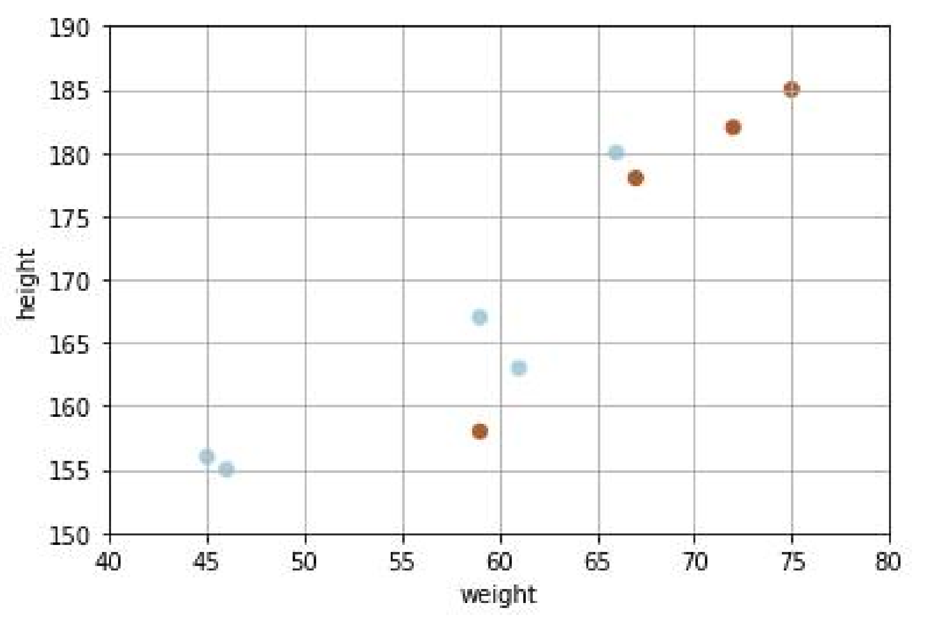
- Red are weight & height of women
- Blue are men
- Random 10 sample of each



Matplotlib Package

```
x1 = [45, 66, 59, 72, 67, 46, 75, 61, 59]

x2 = [156, 180, 167, 182, 178, 154, 183,163, 158]

y = [0,0,0,1,1,0,1,0,1]

plt.scatter(x1,x2,c=y)
```

Matplotlib Package



Matplotlib Package

```
x1 = [45, 66, 59, 72, 67, 46, 75, 61, 59]

x2 = [156, 180, 167, 182, 178, 154, 183,163, 158]


X = [  [45,156], [66,180], [59,167], [72,182], [67,178],
       [46,155], [75,185], [61,163], [59,158]  ]
```

plt.scatter(x1,x2,c=y) → **plt.scatter(X[:,0], X[:,1], c=y)**

Numpy Package

```
x1 = [45, 66, 59, 72, 67, 46, 75, 61, 59]

x2 = [156, 180, 167, 182, 178, 154, 183,163, 158]


X = np.array ( [  [45,156], [66,180], [59,167], [72,182], [67,178],
        [46,155], [75,185], [61,163], [59,158]  ] )
```

`plt.scatter(x1,x2,c=y)` → `plt.scatter( X[:,0], X[:,1], c=y )`

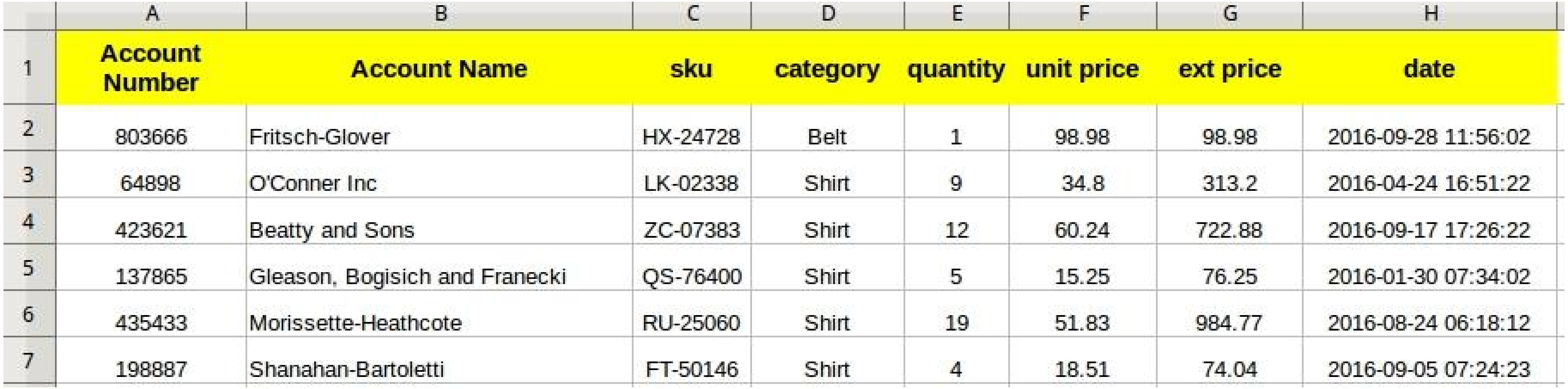
It’s time to predict who is a man or woman ?

Scikit-learn Package

```
X = np.array ( [  [45,156], [66,180], …  ] )
y = [0,0,0,1,1,0,1,0,1]

from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
clf = clf.fit(X, y)
```

```
clf.predict([ [45,156] ])
```

Analysis a huge and more complicated data



	A	B	C	D	E	F	G	H
1	**Account Number**	**Account Name**	**sku**	**category**	**quantity**	**unit price**	**ext price**	**date**
2	803666	Fritsch-Glover	HX-24728	Belt	1	98.98	98.98	2016-09-28 11:56:02
3	64898	O'Conner Inc	LK-02338	Shirt	9	34.8	313.2	2016-04-24 16:51:22
4	423621	Beatty and Sons	ZC-07383	Shirt	12	60.24	722.88	2016-09-17 17:26:22
5	137865	Gleason, Bogisich and Franecki	QS-76400	Shirt	5	15.25	76.25	2016-01-30 07:34:02
6	435433	Morissette-Heathcote	RU-25060	Shirt	19	51.83	984.77	2016-08-24 06:18:12
7	198887	Shanahan-Bartoletti	FT-50146	Shirt	4	18.51	74.04	2016-09-05 07:24:23

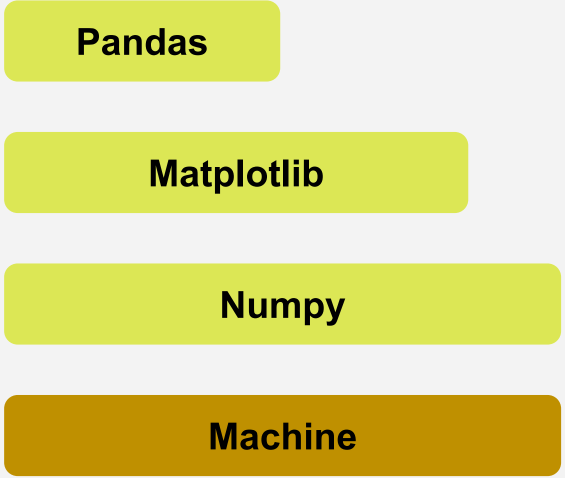
Data Analysis with Pandas Package

Pandas Package

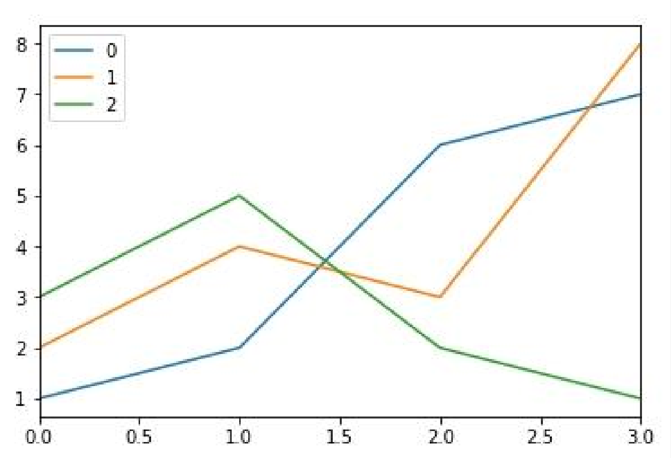
```
import numpy as np

import pandas as pd

X = pd.DataFrame( [ 1,2,6,7 ] )

Y = [ 1,2,6,7 ]
```

```
print(X)

print(Y)
```

```
X.plot()

Y.plot()
```



Pandas Package



```
X = pd.DataFrame( [
                [1,2,3] ,
                [2,4,5] ,
                [6,3,2] ,
                [7,8,1]
            ] )
print(X)
X.plot()
```

Pandas Package

Import data from CSV file

```
sales = pd.read_csv( ‘ datasets/sample-sales.csv ’ )

sales.head()
```

```
sales.head(n=2)
```

Pandas Package

Number of **sales** records ---- print(len(**sales**))

Select ‘**ext price**’ column ---- ext_price = sales['**ext price**']

Minimum of ext price ---- ext_price.**min**()

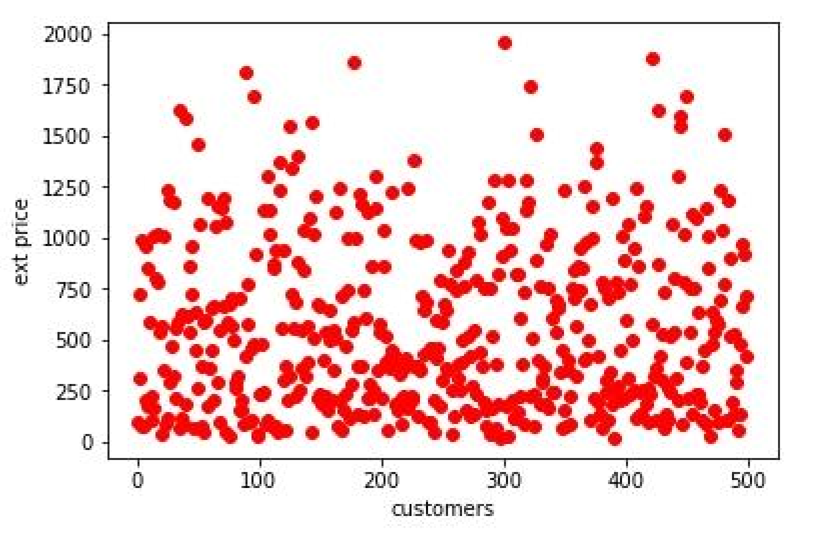
Mode of ext price ---- ext_price.**mode**()

Describe ext price ---- ext_price.**describe**()

Describe sales ---- sales.**describe**()

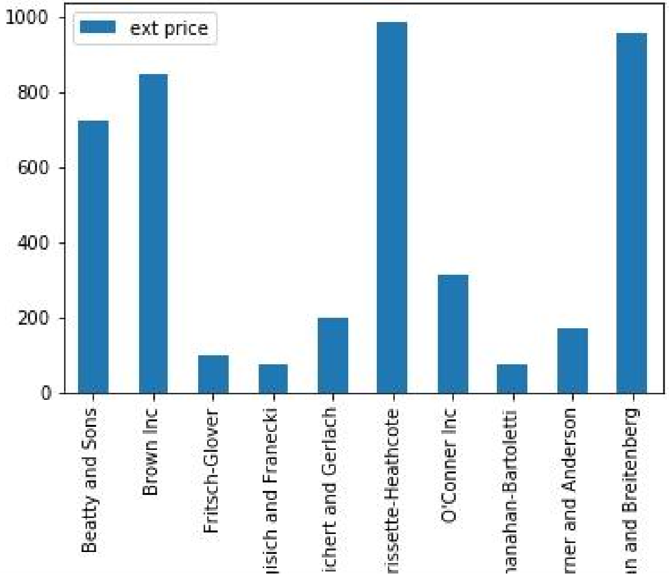
Pandas Package

See ‘ext price’ of 500 customers (hint using matplotlib)



Pandas Package

What does the 'ext price' of each customer (Account Name) **?**

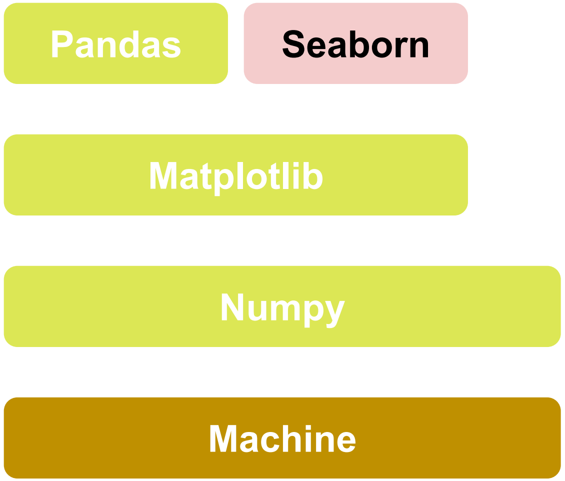


Pandas Package

What does the ‘ext price’ of each customer (Account Name) **?**

```
customer = sales [ [ 'Account Name ' , ' ext price ' ] ] [ : 10 ]

print(customer)

customer_group = customer.groupby('Account Name')

total = customer_group.sum()

total.plot(kind='bar')
```

Seaborn Package



Pandas

Seaborn

Matplotlib

Numpy

Machine

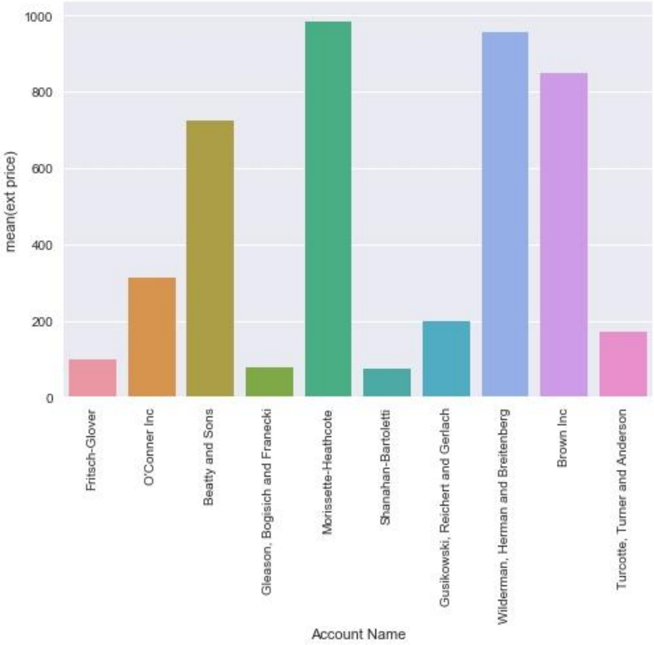
Seaborn Package

What does the ‘ext price’ of each customer (Account Name) **?**

```
import seaborn as sns

sns.set_style('darkgrid') # style must be one of white, dark, whitegrid, darkgrid, ticks

bar_plot = sns.barplot( x=customer['Account Name'],
                        y=customer['ext price'][:10] )

plt.xticks(rotation=90)
```

Seaborn Package

What does the ‘ext price’ of each customer (Account Name) **?**



Seaborn Package

Let’s see

```
import seaborn as sns

sns.set_style('darkgrid') #style must be one of white, dark, whitegrid, darkgrid, ticks

sns.barplot(x='Account Name', y='ext price', data=sales[:15])

plt.xticks(rotation=90)
```

Seaborn Package

Let’s see

```
import seaborn as sns

sns.set_style('darkgrid')

sns.barplot(x='Account Name', y='ext price', hue='category', data=sales[:15])

plt.xticks(rotation=90)
```

Seaborn Package

Let’s see

```
import seaborn as sns

sns.distplot(sales['unit price'])
```

Next
Part 4 - Data Mining

Kanda Tiwatthanont @ TNI