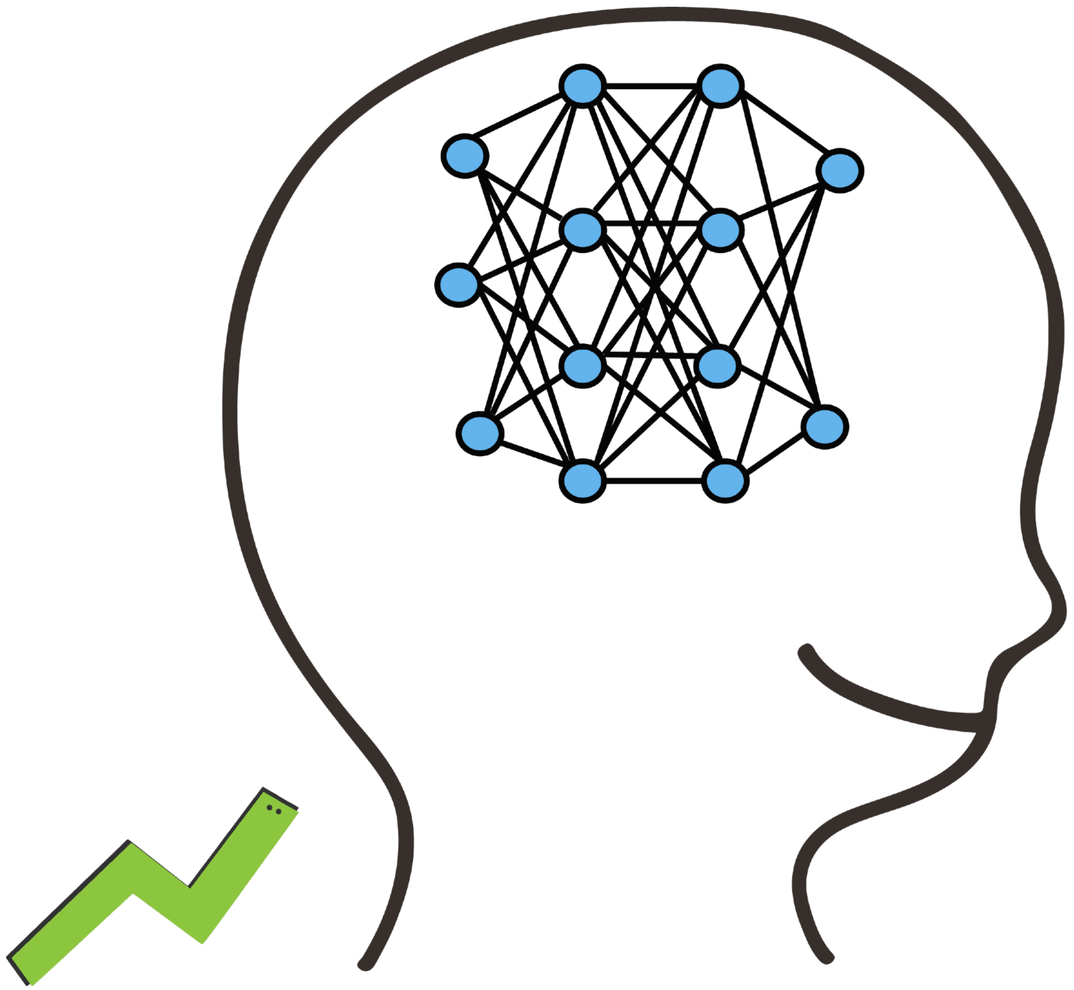


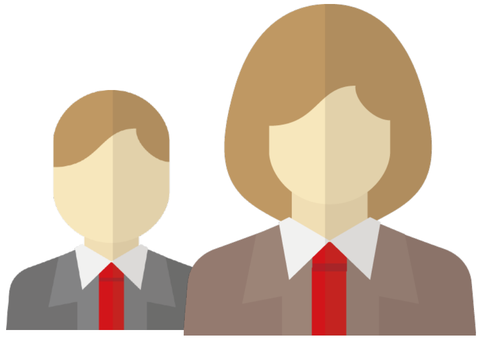
金融數據分析與預測導論

Introduction to Predictive Analytics in Finance

誰在用金融數據？

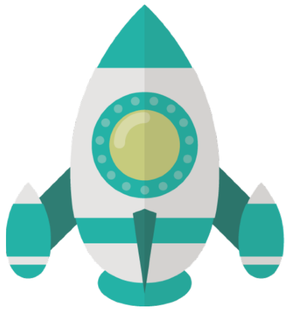
金融機構

(銀行、證券商、保險...)



投資理財者/企業

(資產管理、量化投資...)



FinTech 服務企業

(P2P融資、群眾募資、支付...)

金融數據應用

- 金融機構能利用數據做什麼？
  - 客戶洞察
    - ✓ 金融產品精準行銷
    - ✓ 個人化保險
    - ✓ 借貸風險評估
    - ✓ 理財顧問機器人

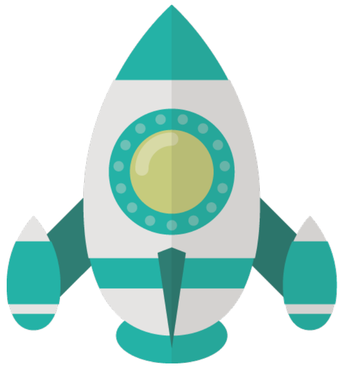
金融數據應用

- Fintech 服務企業能利用數據做什麼？

  ✓ 資產管理服務

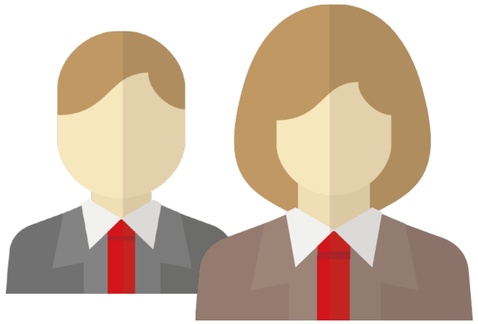
  ✓ 房地產估值

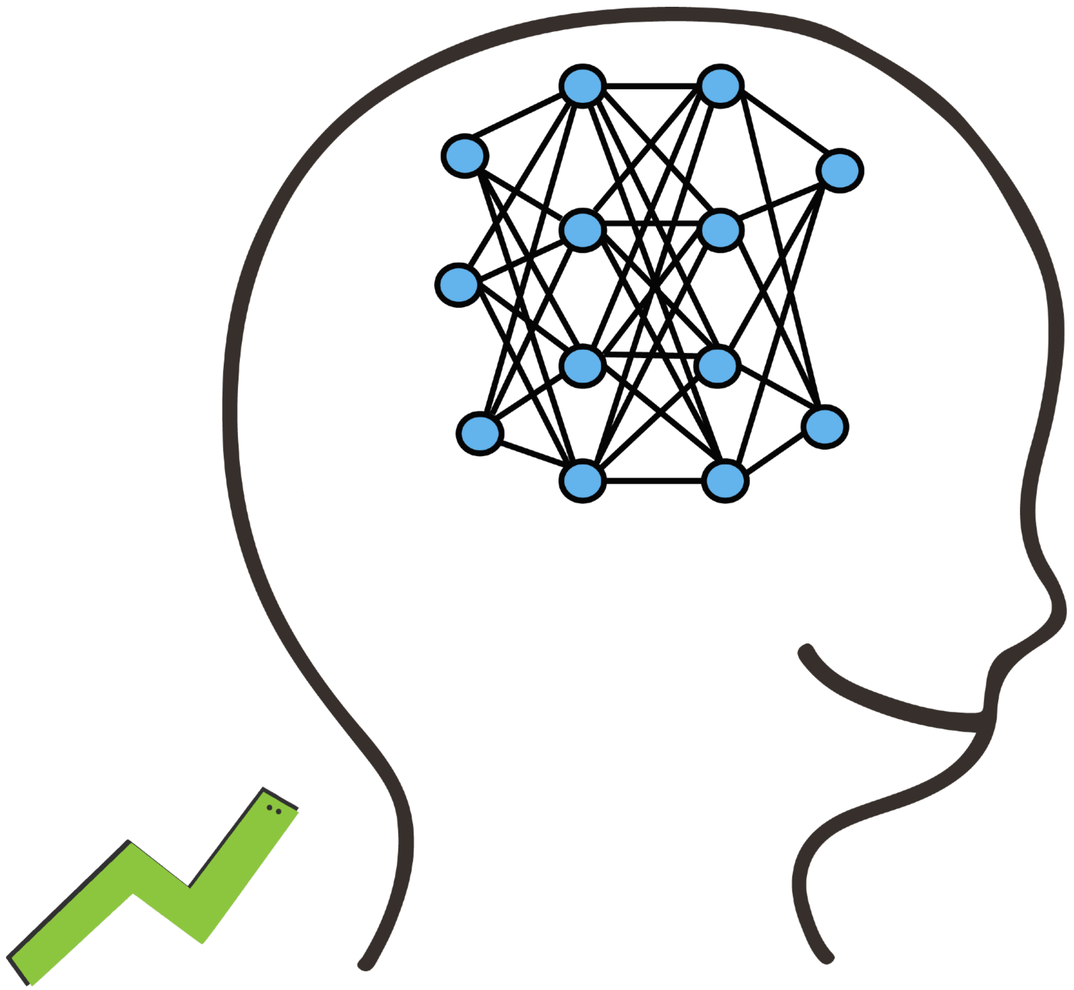
  ✓ 線上融資借貸



金融數據應用

- 投資理財者能利用數據做什麼？
  - 個人資產管理
  - 量化投資管理
    - ✓ 股市投資
    - ✓ 貨幣匯率投資



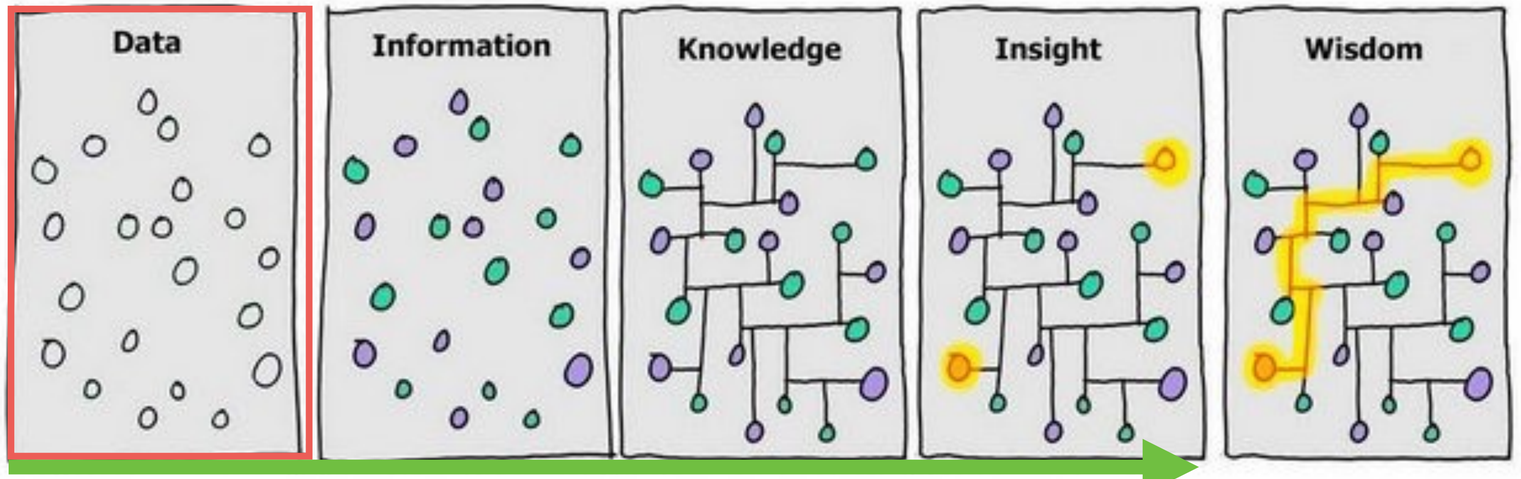


數據的應用層次

The Levels of Data Applications

洞察 (Insights)

- 資料科學是將資料轉化成能解決問題的洞察（Insights），進而產生決策（Decisions）、行動（Actions）、智慧應用（Intelligent Applications）。



轉化成人能解讀的訊息

Original graphic by Hugh MacLeod
extended by David Sommerville (2014)

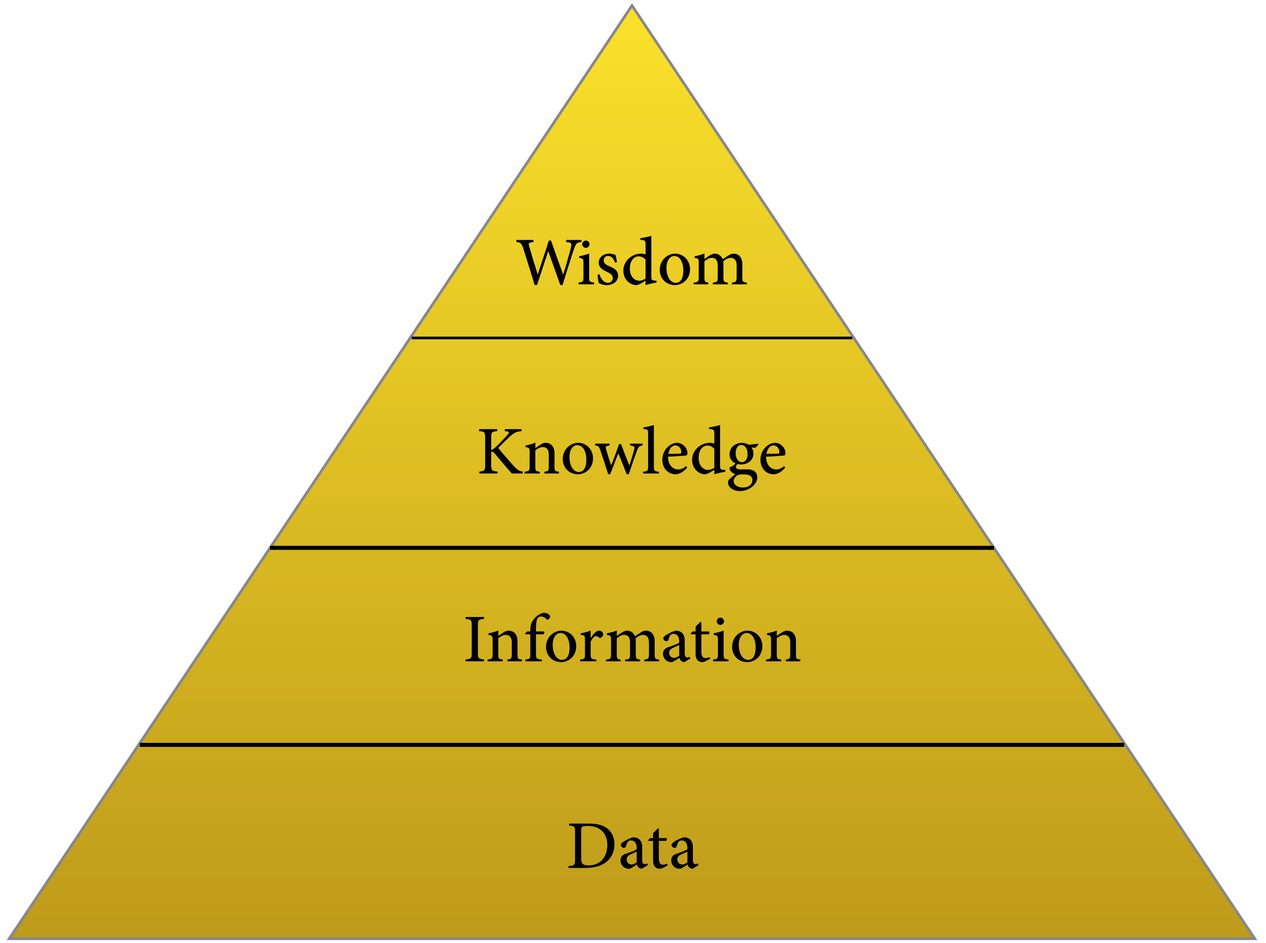
系統化零散的資訊

Wow/Aha！

理解背後的原因

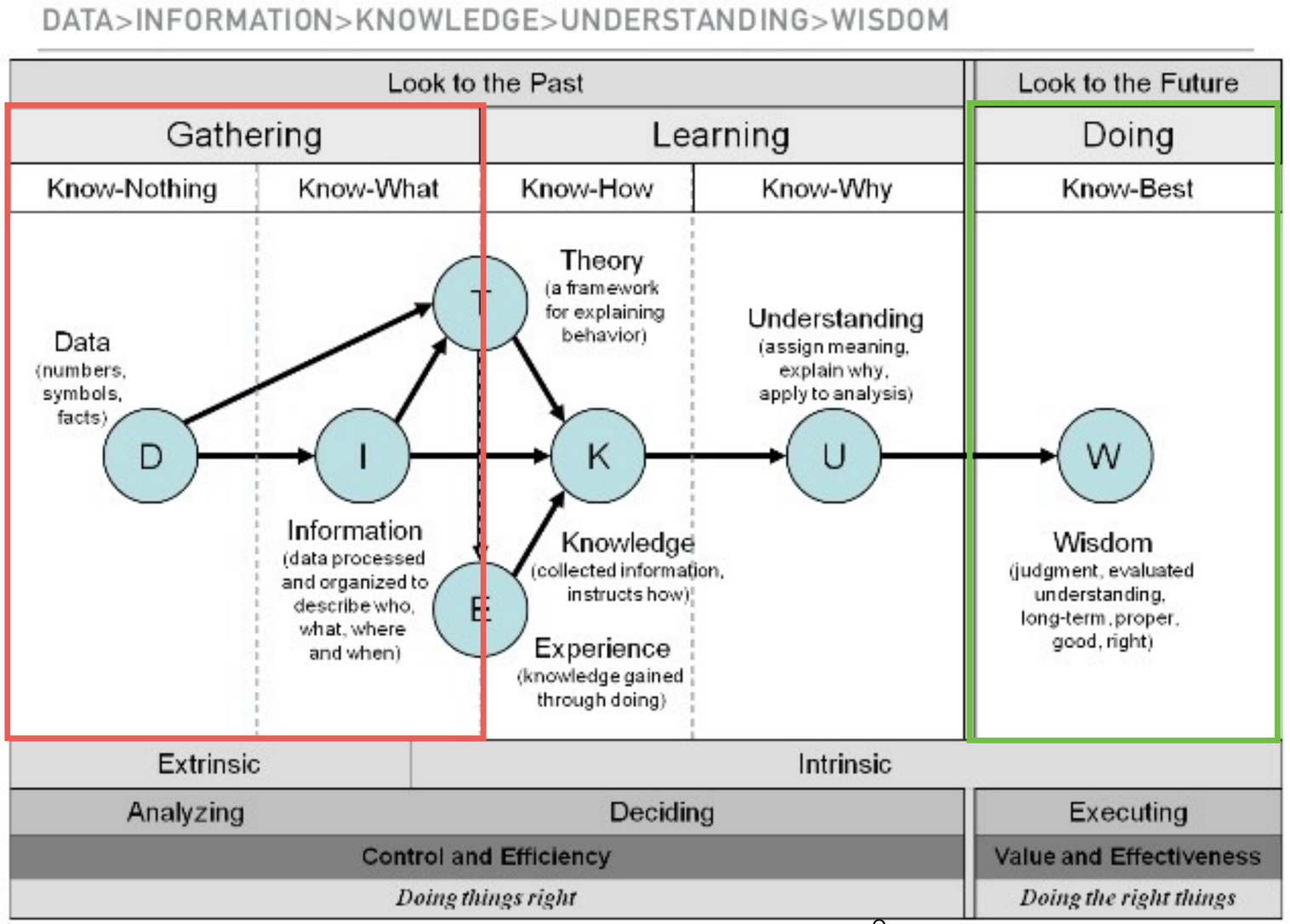
決策、行動、智慧

DIKW 金字塔



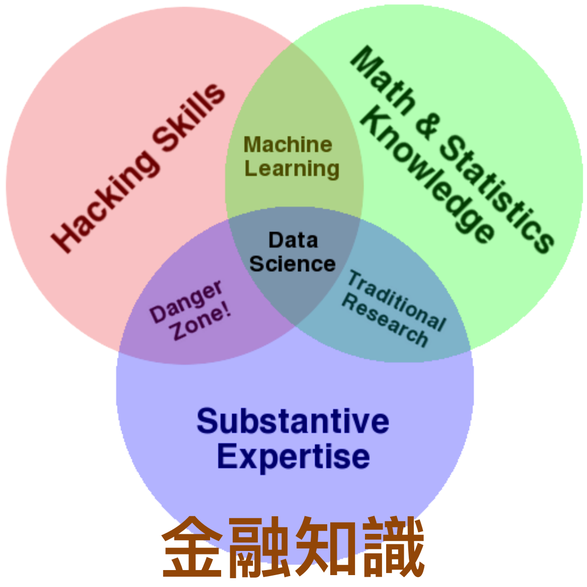
(Ackoff, 1989)

DIKW

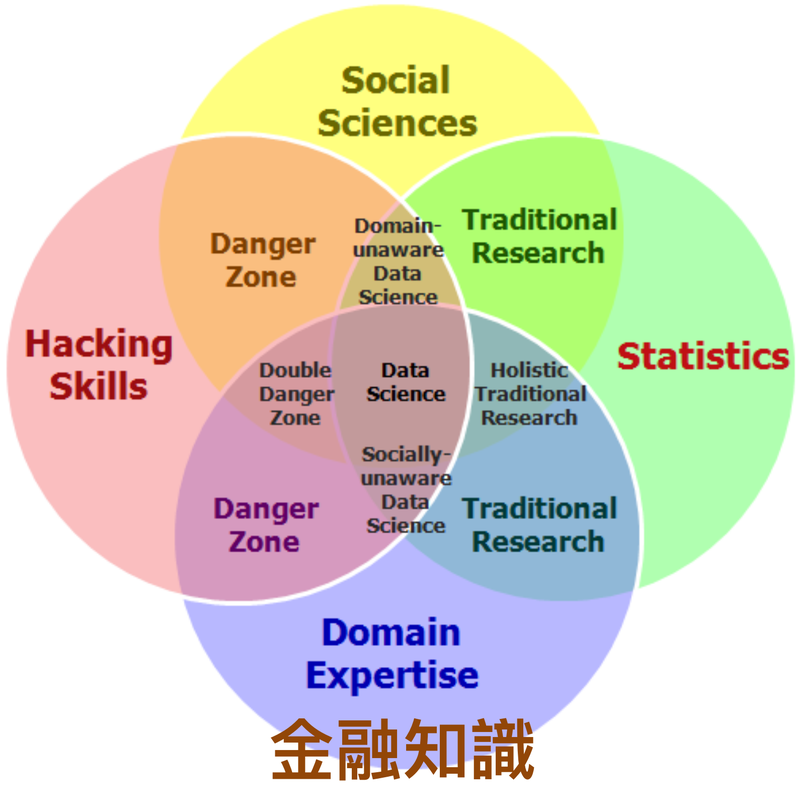


(Ervick & Michael, 2012)

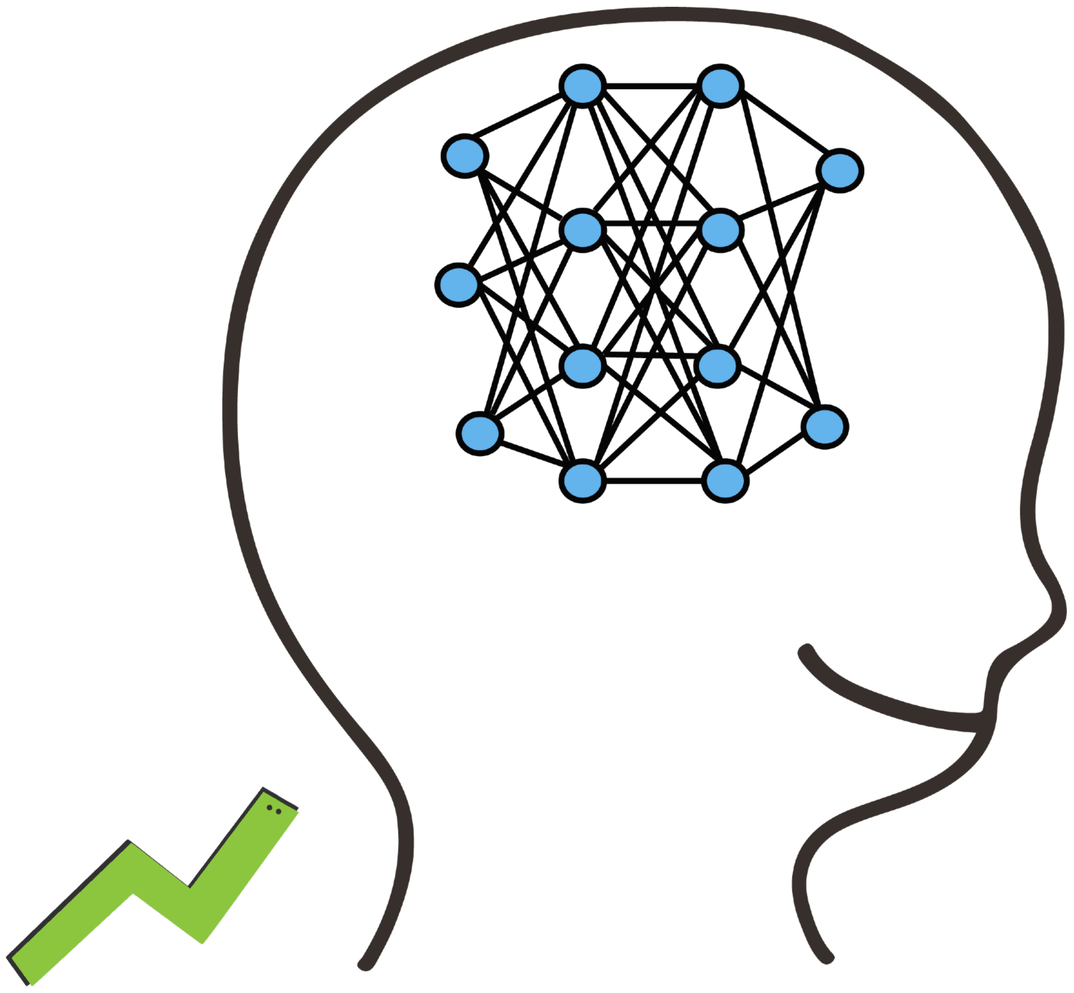
數據分析需要哪些領域知識？



(Drew Conway, 2010)



(Michael Malak, 2014)



Why Python for Finance?

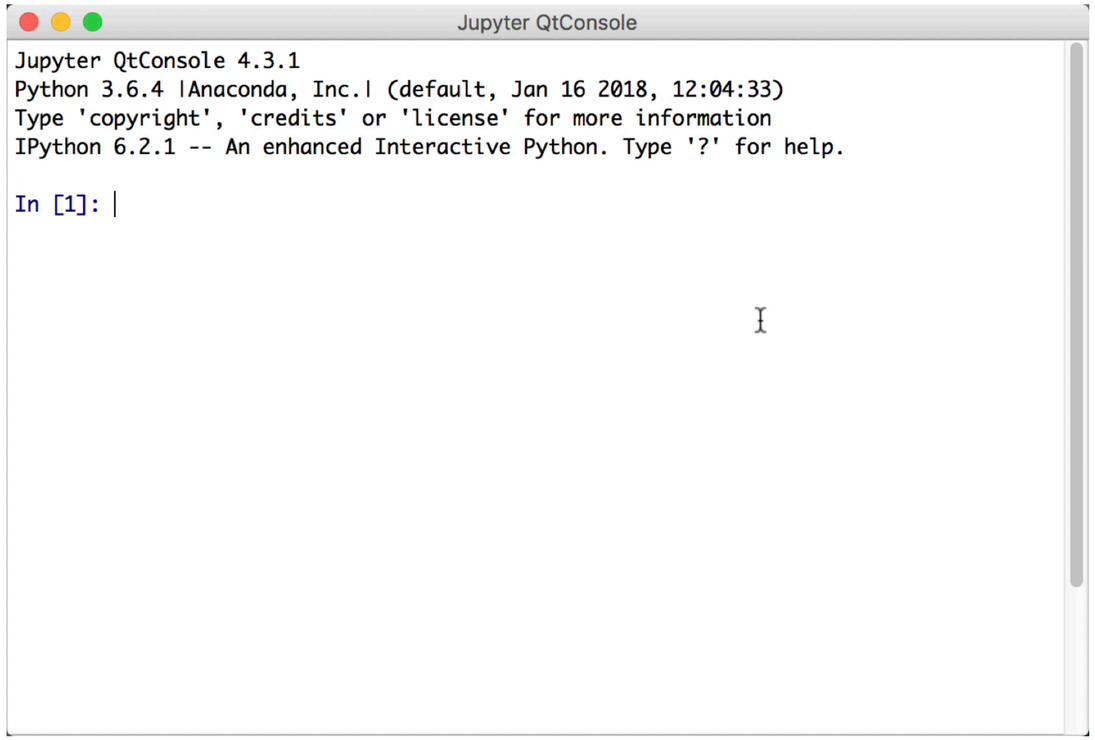
```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Why Python ?

- The Zen of Python（設計哲學）
  - 優美（Beautiful）、明確（Explicit）、簡潔（Simple）、可讀性（Readability）....
  - 溫和學習曲線，適合作為第一個學習的程式語言
- 物件導向、直譯式
- 多用途的程式語言

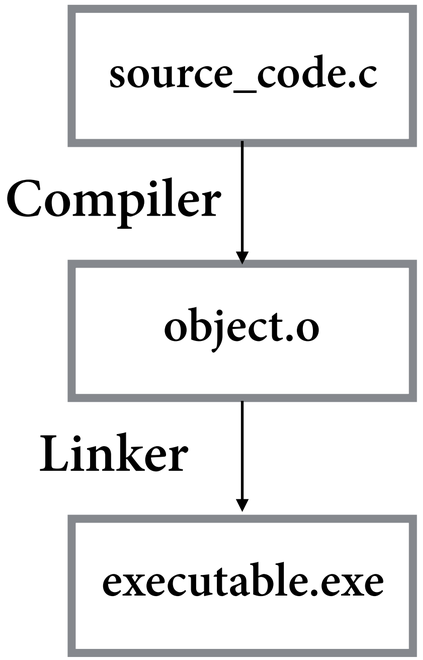
直譯式語言

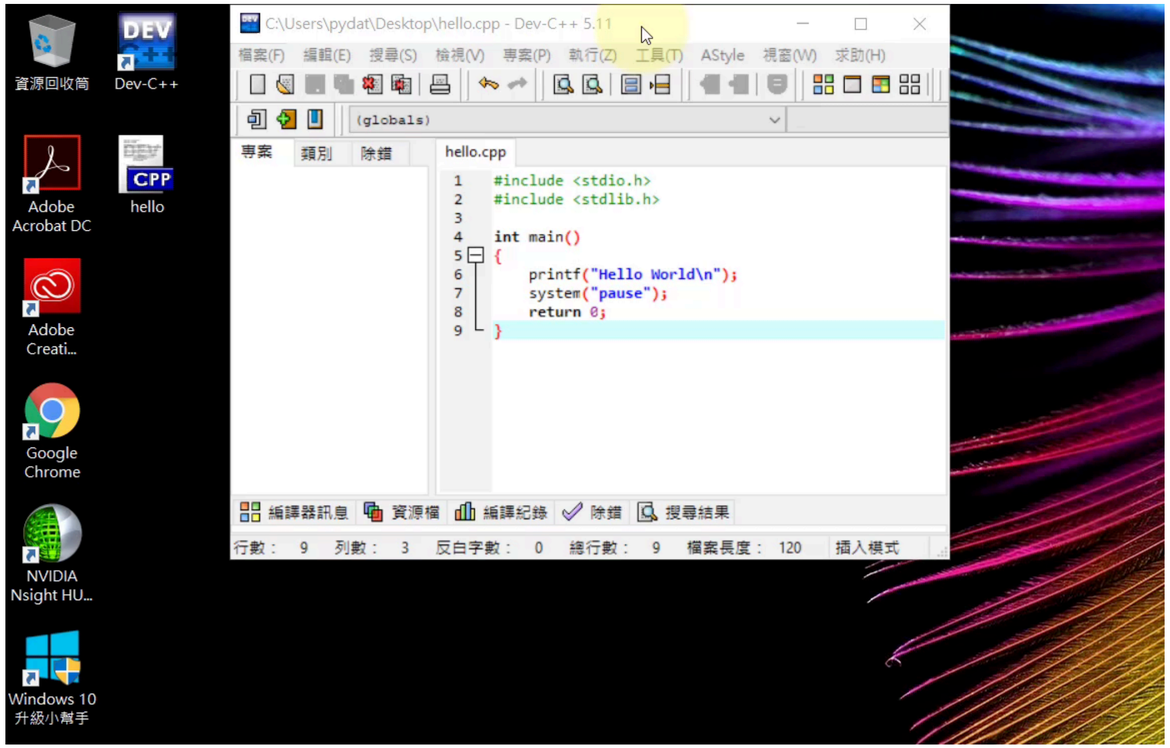
- 直譯式語言（Interpreted Language）：原始碼不需編譯，可一行行直接執行。（如：Python）



編譯式語言

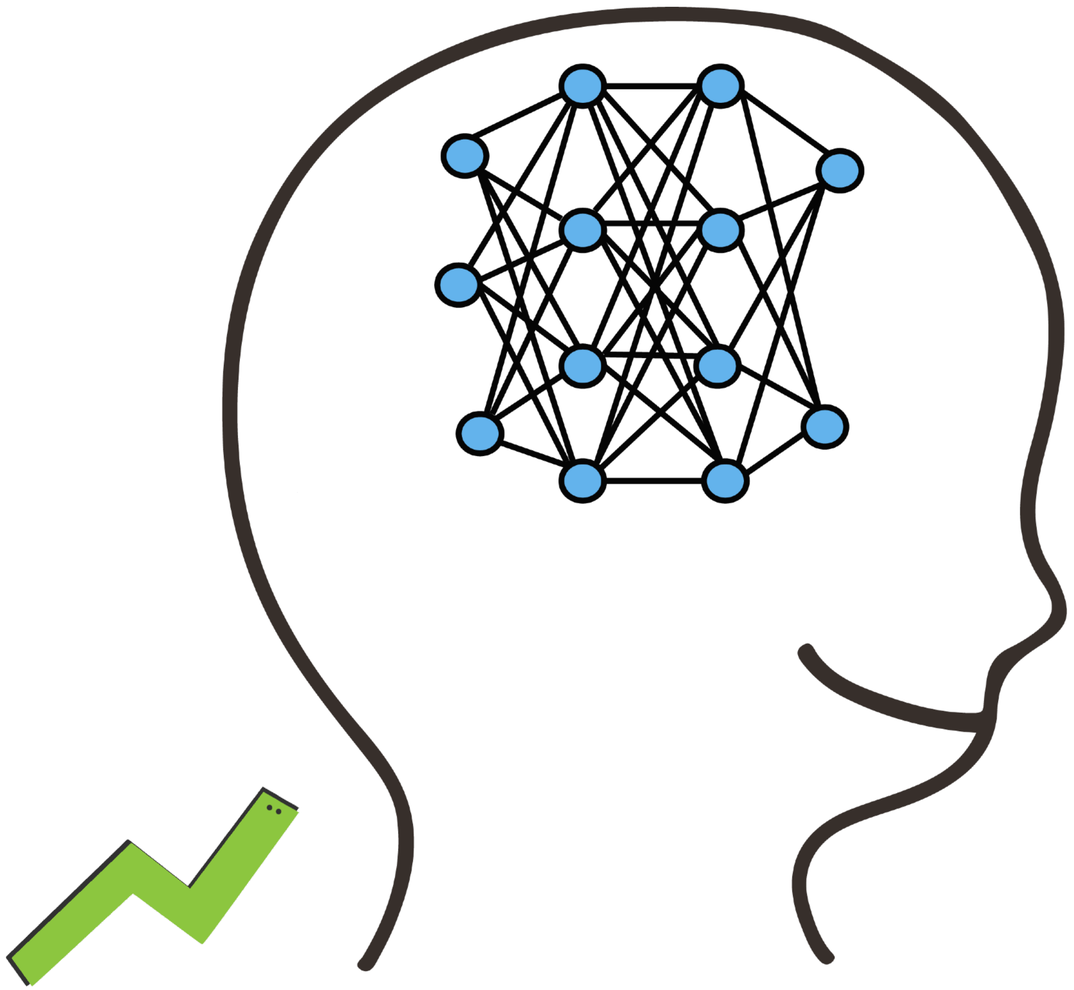
- 編譯式語言（Compiled Language）：原始碼須經過編譯（Compile）後才能執行，如：C語言。





Why Python for Finance?

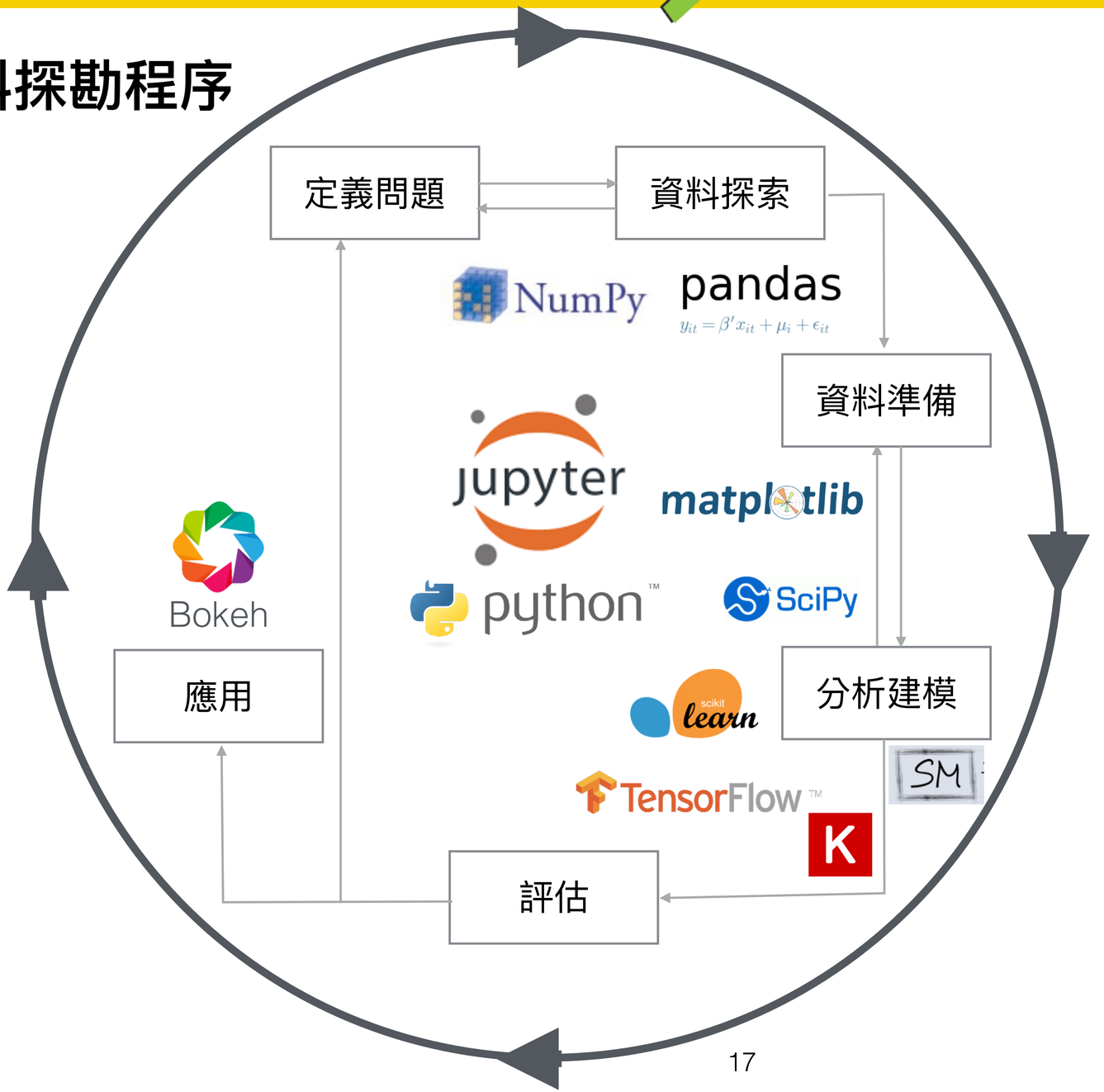
- Python 擁有完善的數據分析、人工智慧、金融應用相關套件，更有多用途的豐富套件：
  - 矩陣運算：Numpy
  - 科學計算、統計分析：Scipy、StatsModels
  - 視覺化：Matplotlib、Seaborn、plotly
  - 資料表分析和視覺化：Pandas
  - 機器學習與深度學習：Scikit-Learn、TensorFlow、Keras…
  - 網路爬蟲：requests、Scrapy...
  - 金融應用：
    - ✓ 自動化交易
    - ✓ 股市數據抓取

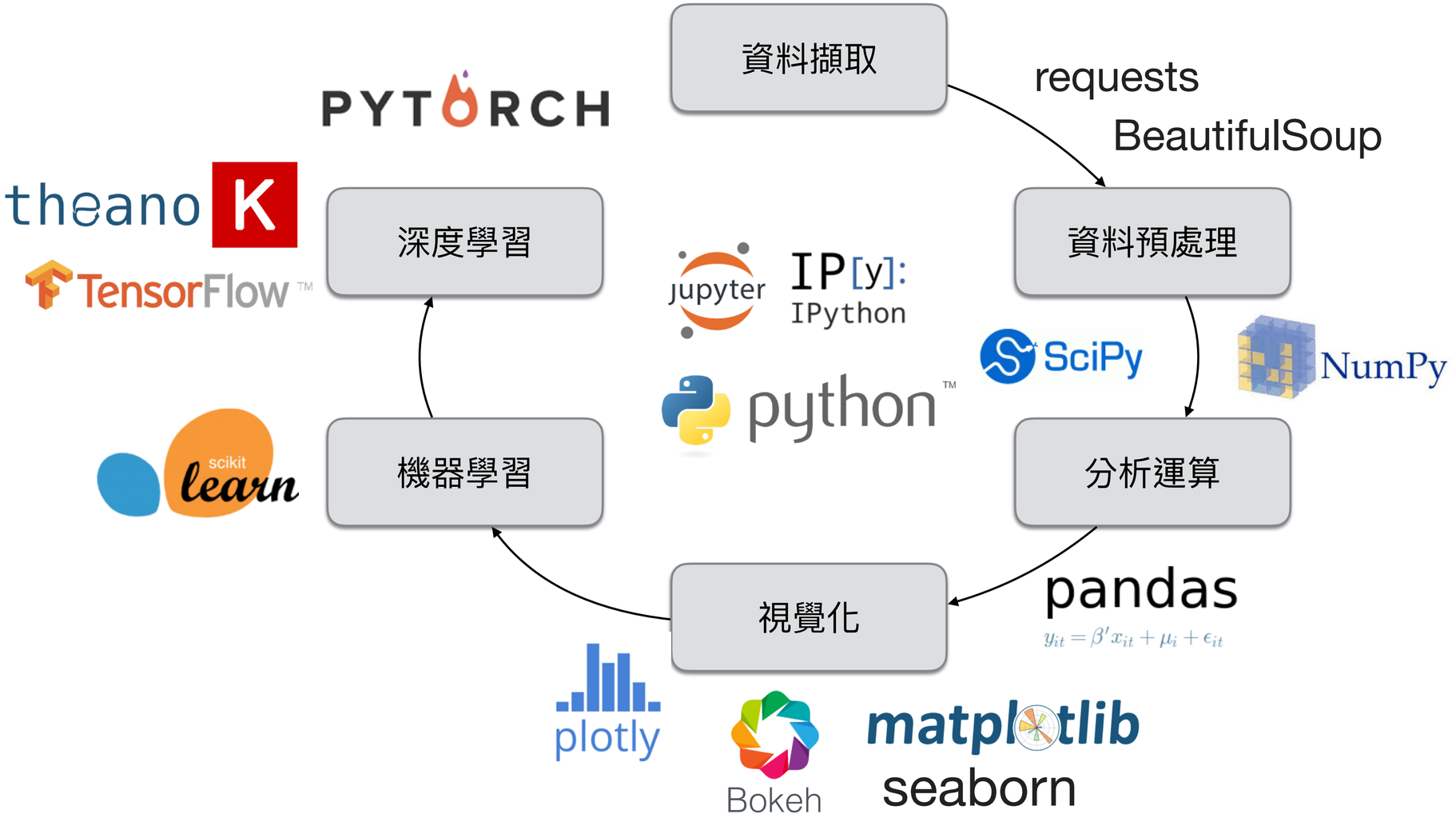


數據分析與預測流程

The Process of Data Analytics and Prediction

CRISP 資料探勘程序




資料擷取
requests
BeautifulSoup
資料預處理
分析運算
pandas
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$
視覺化
matplotlib
seaborn
Bokeh
plotly
機器學習
scikit learn
深度學習
PYTORCH
theano
K
TensorFlow
jupyter
IP[y]:
IPython
python
SciPy
NumPy

數據分析與預測實作流程

問一個好問題
並定義需要的資料

資料擷取 (Extracting)	預處理 (Preprocessing)	分析建模 (Analysis & Modeling)	視覺化 (Visualization)
資料庫 (Database)	清理 (Cleaning)	統計 (Statistic)	統計圖表
資料表 (excel, csv..)	轉換 (Transforming)	資料探勘 (Data Mining)	互動式視覺化
網路爬蟲 (Web Crawler)	標準化 (Normalizing)	機器學習 (Machine Learning)	文字雲
	串接 (Merging)	深度學習 (Deep Learning)	模型評估 (Model Evaluation)
	切割 (Slicing)	社會網路分析 (Social Network Analysis)	
		自然語言處理 (Natural Language Processing)	