

# Python for Data Science

---

Sirakorn Lamyai

September 19, 2019

Student, Kasetsart U.

Obtain the material



<http://bit.ly/cpe-datascience>



## Sirakorn Lamyai

- Theory Research Group, Kasetsart University
- Research Assistant Intern, 2019, Vidyasirimedhi Institute of Science and Technology
- Research Assistant Intern, 2018, Vidyasirimedhi Institute of Science and Technology
- Love drinking tea
- Knows a little about Python

I know a little about Python

# I know a little about Python

When I say I know *a little* about Python...

# I know a little about Python

When I say I know *a little* about Python...

- I think there's some better methods than I'm using

# I know a little about Python

When I say I know *a little* about Python...

- I think there's some better methods than I'm using
- I think I do sometimes make mistakes

# I know a little about Python

When I say I know *a little* about Python...

- I think there's some better methods than I'm using
- I think I do sometimes make mistakes
- There are tons of people who know things much more than me



# I know a little about Python

When I say I know *a little* about Python...

- I think there's some better methods than I'm using
- I think I do sometimes make mistakes
- There are tons of people who know things much more than me
- I think there's much more for me to learn!

# Prerequisite

A basic Python knowledge will do!

## Your expectations from this talk

# Outline

Data Science

Python

- Python environments

- Jupyter

Python Data Structures

Pandas

QAs

# Data Science

---

# The Data Science Process: OSEMNI

- **Obtain** data from relevant sources

# The Data Science Process: OSEMNI

- **Obtain** data from relevant sources
- **Scrub**, sanitise, and clean the data into machine-understandable formats



# The Data Science Process: OSEMNI

- **Obtain** data from relevant sources
- **Scrub**, sanitise, and clean the data into machine-understandable formats
- **Explore** significant and meaningful patterns with statistical methods

# The Data Science Process: OSEMNI

- **Obtain** data from relevant sources
- **Scrub**, sanitise, and clean the data into machine-understandable formats
- **Explore** significant and meaningful patterns with statistical methods
- **Model** construction for prediction and forecast

# The Data Science Process: OSEMNI

- **O**btain data from relevant sources
- **S**crub, sanitise, and clean the data into machine-understandable formats
- **E**xplore significant and meaningful patterns with statistical methods
- **M**odel construction for prediction and forecast
- **i**n**T**erpret and use the results obtained

# The Data Science Process: OSEMNI

- **Obtain** data from relevant sources
- **Scrub**, sanitise, and clean the data into machine-understandable formats
- **Explore** significant and meaningful patterns with statistical methods
- **Model** construction for prediction and forecast
- **iNterpret** and use the results obtained
- **Iterate** and rethink about your outputs

# Why data?

Google

oxford shoes under \$200


🔍 🗣️ 🔍

🔍 ทั้งหมด 🖼️ ค้นรูป 📄 ช้อป Bing 📺 วิดีโอ 📄 ข่าวสาร ⋮ เพิ่มเติม 🛒 การตั้งค่า 🛠️ เครื่องมือ


ผลการค้นหาประมาณ 31,400,000 รายการ (0.84 วินาที)

ดู oxford shoes under \$200


ผู้สนับสนุน




Ted Baker Murain  
oxford shoes in...  
฿3,850.84  
£100.00  
ASOS




รองเท้าหนังวีวแท้  
oxford style งา...  
฿2,990.00  
Lazada Thailand



Timberland  
Stormbuck Plai...  
฿3,517.00  
Dressinn.com  
★★★★★ (48)




Ted Baker Ollivur  
brogue shoes i...  
฿3,850.84  
£100.00  
ASOS



SOLE Dunstan  
Black Shoes  
฿2,502.66  
£64.99  
Soletrader

12 Best Men's Dress Shoes Under \$200

- Kenneth Cole Reaction Last Laugh. ...
- Nordstrom Cusano Double Monk Shoe. ...
- Cole Haan Briscoe Wingtip. ...



# Why data?

The screenshot shows a Facebook interface with a blue header bar. The left sidebar contains navigation options: News Feed, Messenger, Watch, Marketplace, and Shortcuts. The main content area displays a sponsored post from 'Curated and Co.' for 'New Arrival Berwick Penny Loafer Collection'. The post includes Thai text describing the shoes and a photo of five loafers. The right sidebar shows a list of friends with their names, mutual friend counts, and 'Add Friend' or 'Remove' buttons. The bottom of the page shows a language selector (English, Thai, etc.) and a chat button.

Facebook interface showing a news feed post from Curated and Co. (Sponsored).

The post title is: New Arrival Berwick Penny Loafer Collection !

The post text (in Thai): Berwick Penny Loafer ที่มีรุ่นใหม่เข้ามาให้เลือกกันถึง 6 แบบเลยที่เดียวครับ ไม่ว่าจะเป็น Oiled 173 Suede, Polo Brown Suede ที่ได้รับความนิยมมากในต่างประเทศ รวมถึงหนัง Smooth อีก 4 แบบที่สวยงามตัวมาก ไม่ว่าจะเป็น Vegano Melize, Moka และสุดท้ายเป็นสีดำ Black Box Calf ที่คลาสสิกตลอดกาล

The post includes a photo of five Berwick Penny Loafers (three dark brown and two light brown) displayed on a white surface.

The right sidebar shows a list of friends with their names, mutual friend counts, and 'Add Friend' or 'Remove' buttons.

Friends listed:

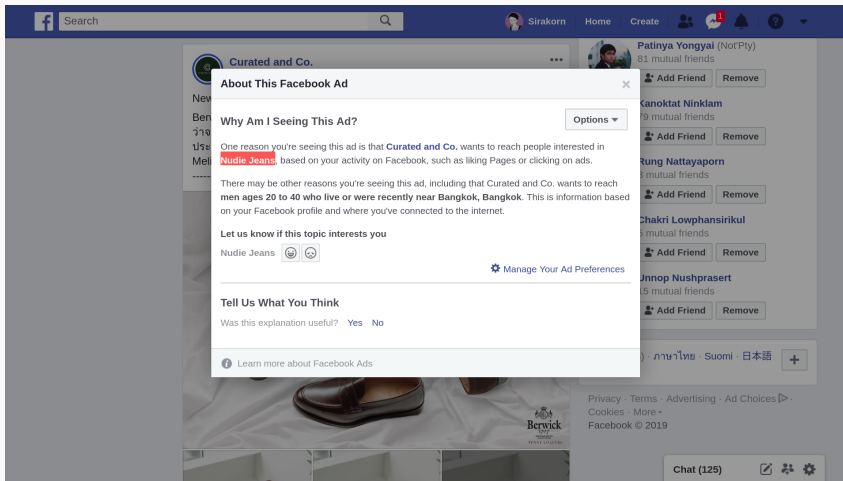
- Patinya Yongyai (NotPty) 81 mutual friends
- Kanoktat Ninklam 79 mutual friends
- Rung Nattayaporn 3 mutual friends
- Chakri Lowphansirikul 5 mutual friends
- Unnop Nushprasert 15 mutual friends

The bottom of the page shows the language selector (English (US) - ภาษาไทย - Suomi - 日本語 - Español) and the Facebook logo.

# Why data?

Facebook interface showing a sponsored advertisement for Berwick Penny Loafers. The ad is titled "Curated and Co. Sponsored" and features a photo of several brown leather loafers. The text of the ad is in Thai, mentioning "New Arrival Berwick Penny Loafer" and listing various styles like "Oiled 173 Suede", "Polo Brown", "Melize", and "Moka". A context menu is open over the ad, showing options: "Hide ad", "Report ad", "Save post", "Why am I seeing this ad?", "Turn on notifications for post", "Embed", and "More options". The "Why am I seeing this ad?" option is highlighted. The right sidebar shows a list of friends with their mutual friend counts. The bottom of the page shows the footer with links to Privacy, Terms, Advertising, Ad Choices, Cookies, and More, along with the Facebook copyright notice for 2019.

# Why data?





Data is the new oil

# Tools for data analysis

## With GUIs

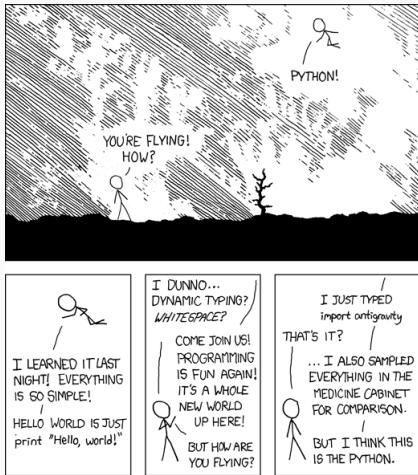
- Spreadsheets
  - Excel
  - Google Spreadsheets
  - Lotus 1-2-3
- Modelling and Visualisation
  - RapidMiner Studio
  - Weka
  - Tableau

## As programming languages

- For data insights
  - R
  - Python
- For data retrieval
  - SQL

# Python

---



Courtesy: xkcd (<https://xkcd.com/353/>)

## *I loved Python...*

- Read it, understand it
- Multiparadigm
- Batteris included
- Lots of great, great libraries!



pip

pip

- PyPA (Python Packaging Authority)'s recommended package installer



pip

- **PyPA** (**P**ython **P**ackaging **A**uthority)'s recommended package installer
- Obtains packages from **PyPI** (**P**ython **P**ackaging **I**ndex)

pip

- **PyPA** (**Py**thon **P**ackaging **A**uthority)'s recommended package installer
- Obtains packages from **PyPI** (**Py**thon **P**ackaging **I**ndex)
- Many useful packages for us to use!





- Cross-platform Python Distribution





- Cross-platform Python Distribution
- Ships with its own package and environment manager



- Cross-platform Python Distribution
- Ships with its own package and environment manager
  - Its environment manager capability is not found in Python vanilla installation



- Cross-platform Python Distribution
- Ships with its own package and environment manager
  - Its environment manager capability is not found in Python vanilla installation
  - Fetches the packages from its own repository, not PyPI





- Cross-platform Python Distribution
- Ships with its own package and environment manager
  - Its environment manager capability is not found in Python vanilla installation
  - Fetches the packages from its own repository, not PyPI
- Aims for Data Science use



- Cross-platform Python Distribution
- Ships with its own package and environment manager
  - Its environment manager capability is not found in Python vanilla installation
  - Fetches the packages from its own repository, not PyPI
- Aims for Data Science use
- **Entirely separated Python**

## Environments 101: \$PATH

```
$ echo $PATH
/home/srakrn/.pyenv/plugins/pyenv-virtualenv/shims:/home/
srakrn/.pyenv/shims:/home/srakrn/.pyenv/bin:/home/srakrn
/.local/bin:/usr/local/bin:/usr/local/sbin:/home/srakrn/.
local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
local/games:/snap/bin
```

## Different machines, different Pythons

On my laptop...

```
srakrn@epsilon-ubuntu:~$ which python  
/home/srakrn/.pyenv/shims/python
```

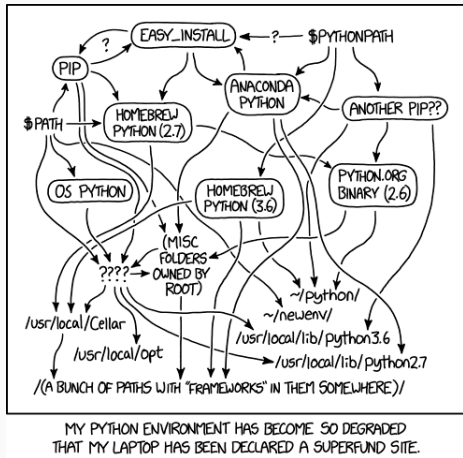
On my <https://charles.srakrn.me/> server...

```
srakrn@charles:~$ which python  
/usr/bin/python  
srakrn@charles:~$ which python3  
/usr/bin/python3
```

## Installed pip

```
$ pip -V
pip 8.1.1 from /usr/lib/python2.7/dist-packages (python 2.7)
$ pip3 -V
pip 8.1.1 from /usr/lib/python2.7/dist-packages (python 3.6)
```

Perhaps now you understand me...



Courtesy: xkcd (<https://xkcd.com/1987/>)









Interactive computing  
environment





- Thinks of a more *dynamic* coding environment.



- Thinks of a more *dynamic* coding environment.
- Inserts snippets of codes alternately with texts, maths, and images.



- Thinks of a more *dynamic* coding environment.
- Inserts snippets of codes alternately with texts, maths, and images.
- A wonderful tool for coding *documented code*.







- Think of an online Jupyter Notebook provided by Google





- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server frame
  - In other words, your code are remotely executed



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server frame
  - In other words, your code are remotely executed
- Could be more powerful for some tasks (like Deep Learning) than your computer



- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server frame
  - In other words, your code are remotely executed
- Could be more powerful for some tasks (like Deep Learning) than your computer
- Free!

`https:  
//colab.research.google.com/`



+ Code + Text

RAM  
Disk

Editing



```
[1] from datetime import datetime
```

```
name = input("Please input your name: ")
```

```
... Please input your name:
```

```
print("Hello, {}".format(name))  
print("It is now {}".format(datetime.now()))
```



## Caveats 1: Execution order

```
[2] a = 10
```

```
[1] a = 5
```

```
[3] print_(a)
```

```
↳ 10
```



|

## Caveats 1: Execution order

```
[2] a = 10
```

```
[1] a = 5
```

```
[3] print_(a)
```

```
↳ 10
```



|

You'll do a lot of out-of-order code execution!



## Caveats 2: Cell edits

```
[1] a = 10
```

```
[2] a *= 2
```

```
[4] print(a)
```

```
↳ 60
```



|

## Caveats 2: Cell edits

```
[1] a = 10
```

```
[2] a *= 2
```

```
[4] print(a)
```

```
↳ 60
```



|

You might sometimes remove a cell, and that shows no visible trace without explicit query.

## Caveats 2: Cell edits

```
[1] a = 5
```

```
[2] a = 20
```

```
[3] print(a)
```

```
10
```



## Caveats 2: Cell edits

```
[1] a = 5
```

```
[2] a = 20
```

```
[3] print(a)
```

```
10
```



Jupyter Notebook offers no cell edited marks, while Colab offers them

## Caveats 2: Cell edits

```
[1] a = 5
```

```
[2] a = 20
```

```
[3] print(a)
```

```
↳ 10
```



Jupyter Notebook offers no cell edited marks, while Colab offers them (note: observe the greyed out cell number)

## Caveats 3: Be neat and tidy

Jupyter Notebook and Colab, unlike IDE and code editors, offers a relatively poor **clean code** tools

- Syntax error highlighting
- Autocomplete
- Linting
- Code formatter

## Sirakorn's Workflow Demo

# Python Data Structures

---



# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.
  - This includes storing lists in a list

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.
  - This includes storing lists in a list
  - So-called a **nested list**

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.
  - This includes storing lists in a list
  - So-called a **nested list**
- Can be resized.

# Lists

```
1 a = [1, 2, 3, 4, 5]
2 b = ["Cats", "Dogs", "Penguins", "Tonkatsu Pieces"]
3 c = [1, "1", True]
```

- **Lists** are a compilation of objects.
- Can store multiple data types.
  - This includes storing lists in a list
  - So-called a **nested list**
- Can be resized.
  - No need to declare its size on the first declaration.

```
1 a = [1, 2, 3, 4, 5]
2 a[0]      # Accessing elements
3 a[1:3]    # Slicing
```

Accessing list



```
1 a = [1, 2, 3, 4, 5]
2 a[0]      # Accessing elements
3 a[1:3]    # Slicing
```

Accessing list

```
1 a = [1, 2, 3, 4, 5]
2 a[0]      # Accessing elements
3 a[1:3]    # Slicing
```

## Accessing list

- **Elementwise:** accessing one elements at a time)

```
1 a = [1, 2, 3, 4, 5]
2 a[0]      # Accessing elements
3 a[1:3]    # Slicing
```

## Accessing list

- **Elementwise**: accessing one elements at a time)
- **Slicing**: accessing a sublist

# List Functions

```
1 vowels = ["a", "e", "o", "u"]
2
3 # Get a's length
4 len(a)
5 # Append the new element to the end of a
6 a.append("y")
7 # Deletes the first occurrence of the element from a
8 a.remove("y")
9 # Inserts the item into a list with a specified index
10 a.insert(2, "i")
```

# List Functions

```
1 vowels = [1, 3, 2, 5, 4]
2 # Get the first index of a specified element
3 a.index(4)
4 # Sort a list and store into a new list
5 sorted_a = sorted(a)
6 # Sort a list, making changes directly to the old one
7 a.sort()
```

```
1 names = {  
2     "Cherprang": "Cher",  
3     "Manipa": "Khamin",  
4     "Jiradapa": "Pupe"  
5 }  
6  
7 kami_nickname = names["Manipa"]  
8 ]
```

```
1 names = {  
2     "Cherprang": "Cher",  
3     "Manipa": "Khamin",  
4     "Jiradapa": "Pupe"  
5 }  
6  
7 kami_nickname = names["Manipa"]
```

- **Dictionaries** store values in a key-pair format.

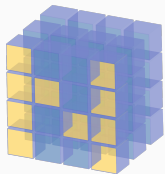
# Dictionary

```
1 names = {  
2     "Cherprang": "Cher",  
3     "Manipa": "Khamin",  
4     "Jiradapa": "Pupe"  
5 }  
6  
7 kami_nickname = names["Manipa"]
```

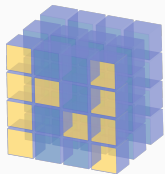
- **Dictionaries** store values in a **key-pair** format.
- From the positional index, dictionary takes the key as an index instead.





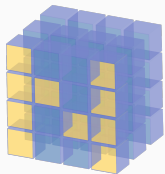


NumPy



NumPy

- NumPy is a powerful library for mathematical computation in Python



NumPy

- NumPy is a powerful library for mathematical computation in Python
- It offers wide range of tools from data structures to advanced functions and operations



- NumPy is a powerful library for mathematical computation in Python
- It offers wide range of tools from data structures to advanced functions and operations
- A very strong ease in mathematical computation, **don't reinvent the wheels!**



*Courtesy: Rebellious Professor,*

<https://www.facebook.com/rebelliousprof/photos/a.301217273588245/599669117076391/>

# Let's go to Notebook!

*I'm too lazy to cover the contents twice...*

# Pandas

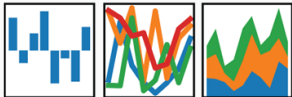
---



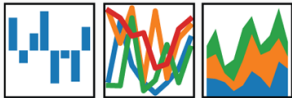


# pandas

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



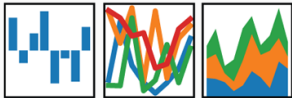
pandas  
 $y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



- Tabular-like structure

# pandas

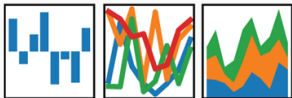
$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Tabular-like structure
- High performance

# pandas

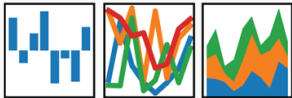
$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Tabular-like structure
- High performance
- Easy to use

# pandas

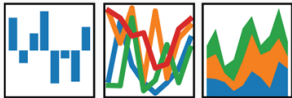
$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Tabular-like structure
- High performance
- Easy to use
- Helps a lot in data preparation

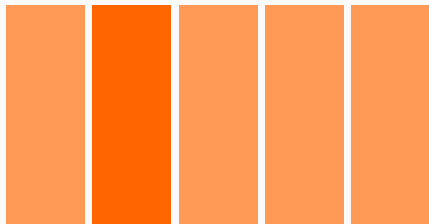
# pandas

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



- Tabular-like structure
- High performance
- Easy to use
- Helps a lot in data preparation
- Considered as a wrapper for Numpy, although there's a lot more

**DataFrame**



**Series**

**DataFrame**



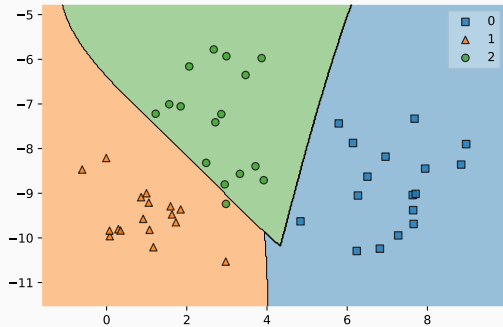
**Series**

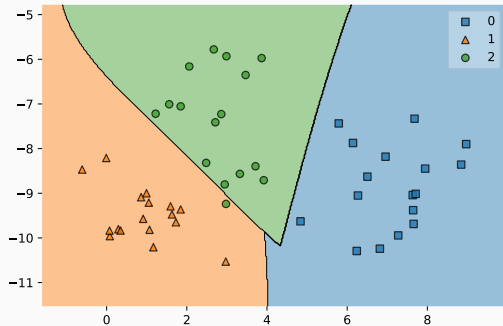


Let's go to Notebook!



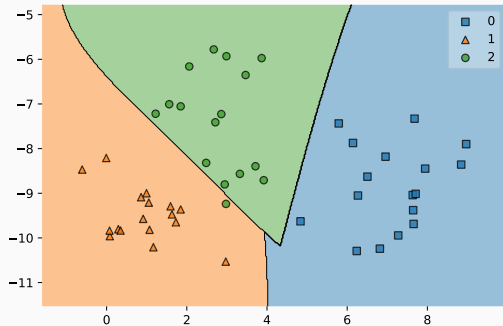
# Data Visualisation





Why Visualise?

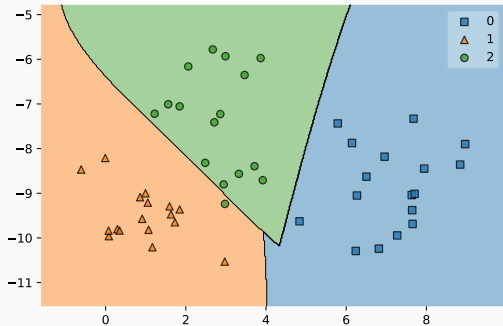
# Data Visualisation



Why Visualise?

- Our visual system is so great!

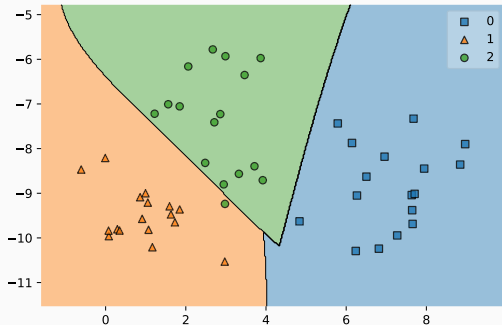
# Data Visualisation



## Why Visualise?

- Our visual system is so great!
- A much better way to represent data than statistical values

# Data Visualisation



## Why Visualise?

- Our visual system is so great!
- A much better way to represent data than statistical values
- Meaningful plots show meaningful insights without needing to do much

# The Datasaurus Dozen

<https://www.autodeskresearch.com/publications/samestats>









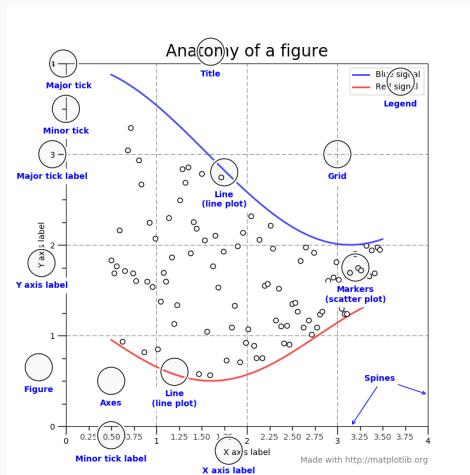
- feature-rich plotting library in python



- feature-rich plotting library in python
- plots multiple types of charts



- feature-rich plotting library in python
- plots multiple types of charts
- extensive support for jupyter notebook



Source: <https://matplotlib.org/tutorials/introductory/usage.html>

Let's go to Notebook!

QAs

---



# *Internships*

# *Applications*

*Today's and tomorrow's importance of Data Science*

*Neural Networks?*