Python for Data Science

Sirakorn Lamyai

September 19, 2019

Student, Kasetsart U.

Obtain the material

http://bit.ly/cpe-datascience

About me

Sirakorn Lamyai

- DAKDL Laboratory, 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

Blending it all together

Data Science

The Data Science Process: OSEMNI

The Data Science Process: OSEMNI

- **Obtain** data from relevent sources

The Data Science Process: OSEMNI

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

The Data Science Process: OSEMNI

- **Obtain** data from relevent 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 relevent 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

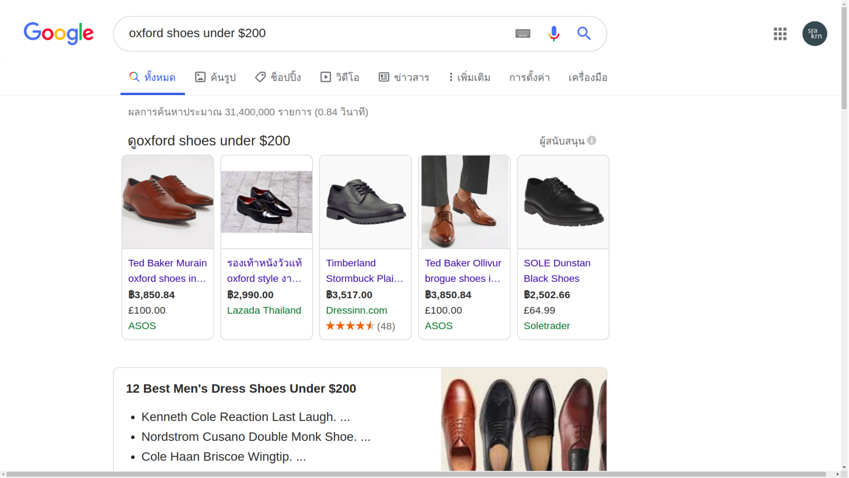
The Data Science Process: OSEMNI

- **Obtain** data from relevent 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

The Data Science Process: OSEMNI

- **Obtain** data from relevent 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
- **Interate** and rethink about your outputs

Why data?



oxford shoes under $200

ผลการค้นหาประมาณ 31,400,000 รายการ (0.84 วินาที)

ดูoxford shoes under $200

ผู้สนับสนุน

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

รองเท้าหนังวัวแท้ oxford style งา…
฿2,990.00
Lazada Thailand

Timberland Stormbuck Plai…
฿3,517.00
Dressinn.com
★★★★½ (48)

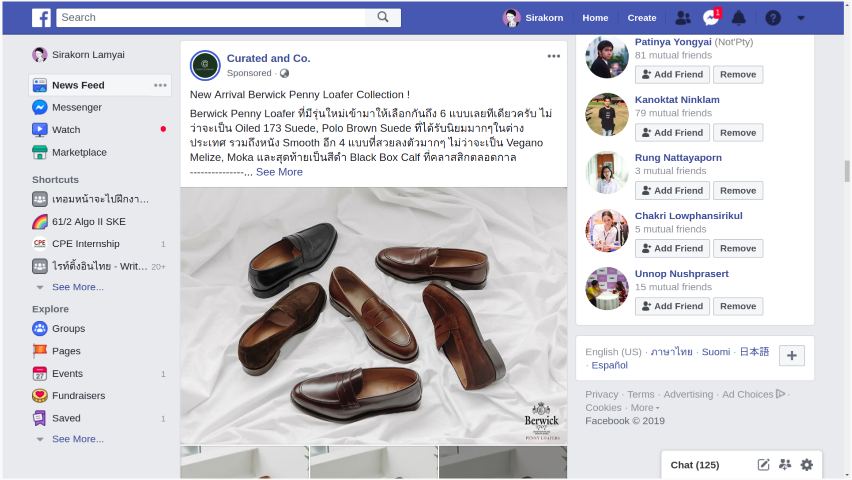
Ted Baker Ollivur brogue shoes i…
฿3,850.84
£100.00
ASOS

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. ...

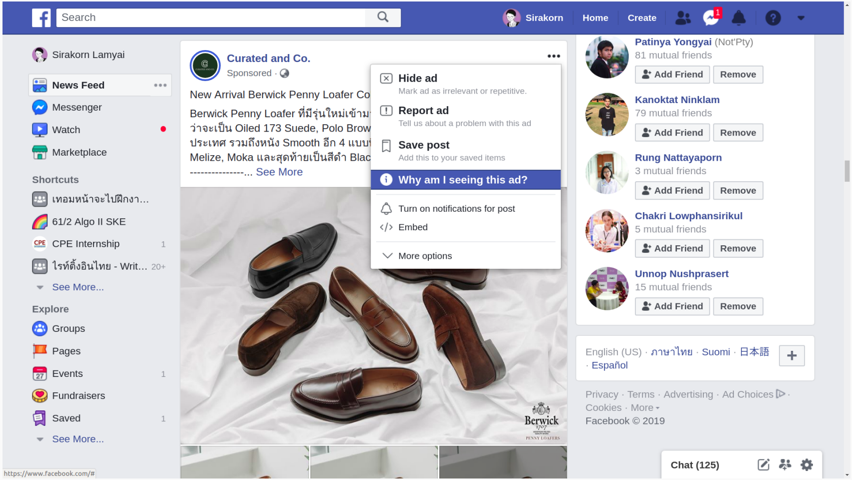
Why data?



New Arrival Berwick Penny Loafer Collection !

Berwick Penny Loafer ที่มีรุ่นใหม่เข้ามาให้เลือกกันถึง 6 แบบเลยทีเดียวครับ ไม่ว่าจะเป็น Oiled 173 Suede, Polo Brown Suede ที่ได้รับนิยมมากๆในต่างประเทศ รวมถึงหนัง Smooth อีก 4 แบบที่สวยลงตัวมากๆ ไม่ว่าจะเป็น Vegano Melize, Moka และสุดท้ายเป็นสีดำ Black Box Calf ที่คลาสสิกตลอดกาล ---------------... See More

Why data?

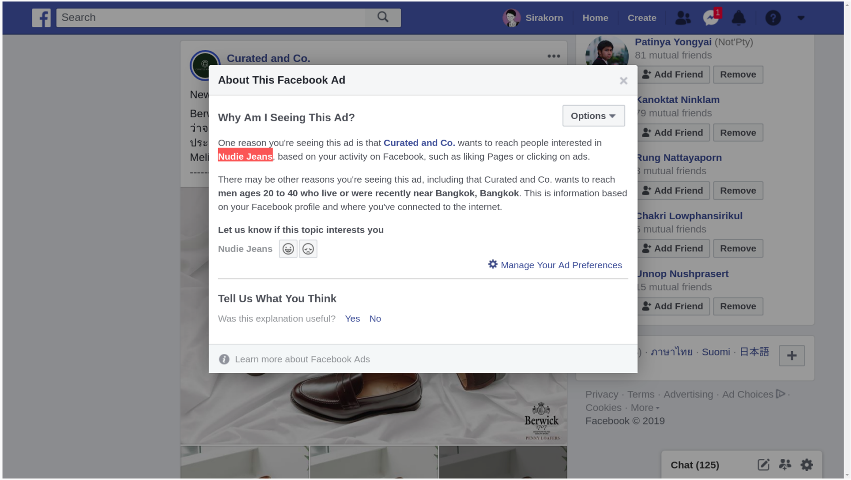


Curated and Co.
Sponsored ·

New Arrival Berwick Penny Loafer Co

Berwick Penny Loafer ที่มีรุ่นใหม่เข้าม
ว่าจะเป็น Oiled 173 Suede, Polo Brow
ประเทศ รวมถึงหนัง Smooth อีก 4 แบบ
Melize, Moka และสุดท้ายเป็นสีดำ Blac
----------------... See More

Why data?



About This Facebook Ad ×

Why Am I Seeing This Ad? Options ▾

One reason you're seeing this ad is that **Curated and Co.** wants to reach people interested in **Nudie Jeans**, based on your activity on Facebook, such as liking Pages or clicking on ads.

There may be other reasons you're seeing this ad, including that Curated and Co. wants to reach **men ages 20 to 40 who live or were recently near Bangkok, Bangkok**. This is information based on your Facebook profile and where you've connected to the internet.

Let us know if this topic interests you

Nudie Jeans

Manage Your Ad Preferences

Tell Us What You Think

Was this explanation useful? Yes No

Learn more about Facebook Ads

Why data?

Data is the new oil

Tools for data analysis

With GUIs

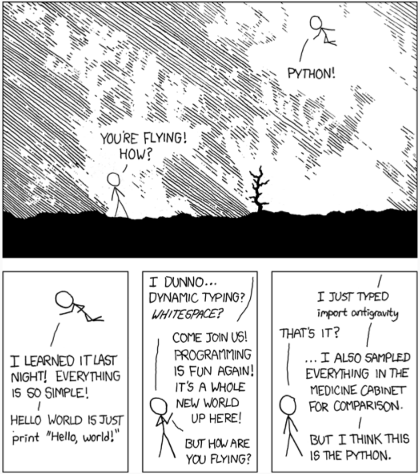
- Spreadsheets
    - Excel
    - Google Spreadsheets
    - Lotus 1-2-3
- Modelling and Visualisation
    - RapidMiner Studio
    - Weka
    - Tableau

As programming languages

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

Python

Python


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Courtesy: xkcd (`https://xkcd.com/353/`)

I *loved* Python...

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

Python Packages

Python Packages

`pip`

Python Packages

`pip`

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

Python Packages

`pip`

- PyPA (Python Packaging Authority)'s recommended package installer
- Obtains packages from PyPI (Python Packaging Index)

Python Packages

`pip`

- PyPA (Python Packaging Authority)'s recommended package installer
- Obtains packages from PyPI (Python Packaging Index)
- Many useful packages for us to use!

Anaconda Python Distribution

Anaconda Python Distribution

Anaconda Python Distribution

- Cross-platform Python Distribution

Anaconda Python Distribution

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

Anaconda Python Distribution

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

Anaconda Python Distribution

- 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

Anaconda Python Distribution

- 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

Anaconda Python Distribution

- 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
```

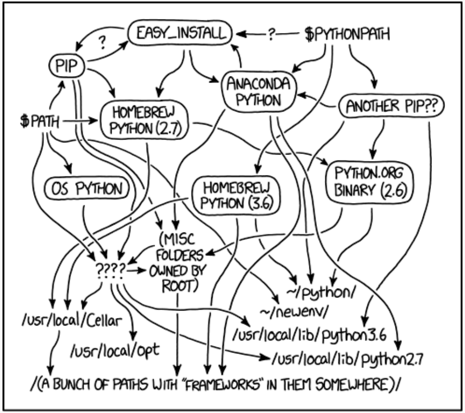
On my https://charles.srakrn.me/ server...

```
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...



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Courtesy: xkcd (`https://xkcd.com/1987/`)

Jupyter

Jupyter

Jupyter

Interactive computing environment

Jupyter Notebook

Jupyter Notebook

- Thinks of a more *dynamic* coding environment.

Jupyter Notebook

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

Jupyter Notebook

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

Google Colaboratory (Colab)

Google Colaboratory (Colab)

Google Colaboratory (Colab)

- Think of an online Jupyter Notebook provided by Google

Google Colaboratory (Colab)

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

Google Colaboratory (Colab)

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

Google Colaboratory (Colab)

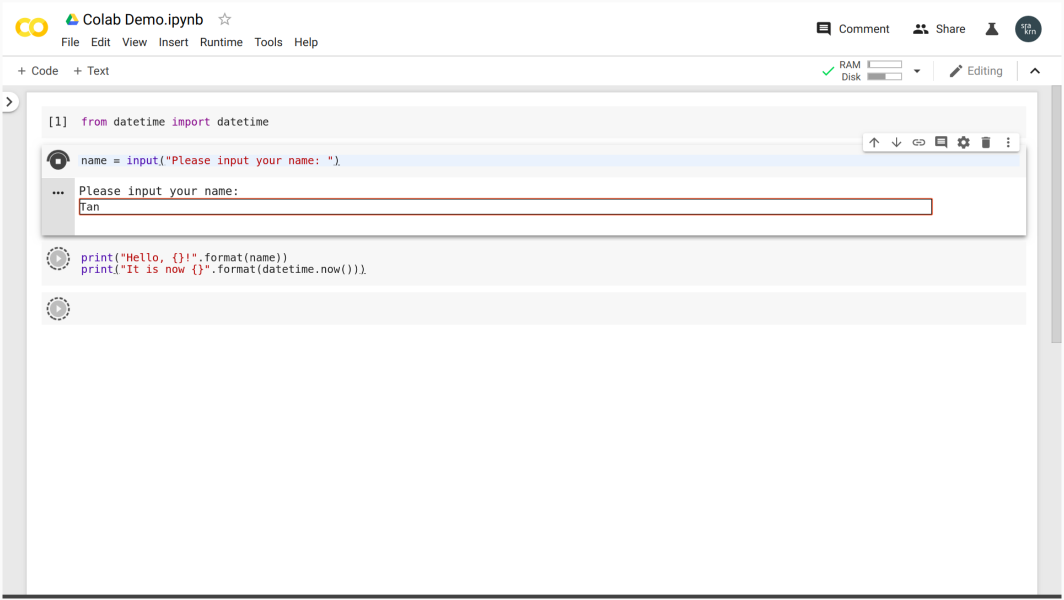
- Think of an online Jupyter Notebook provided by Google
- The runtime relies on Google's server fram
  - In other words, your code are remotely executed
- Could be more powerful for some tasks (like Deep Learning) than your computer

Google Colaboratory (Colab)

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

Google Colaboratory (Colab)

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



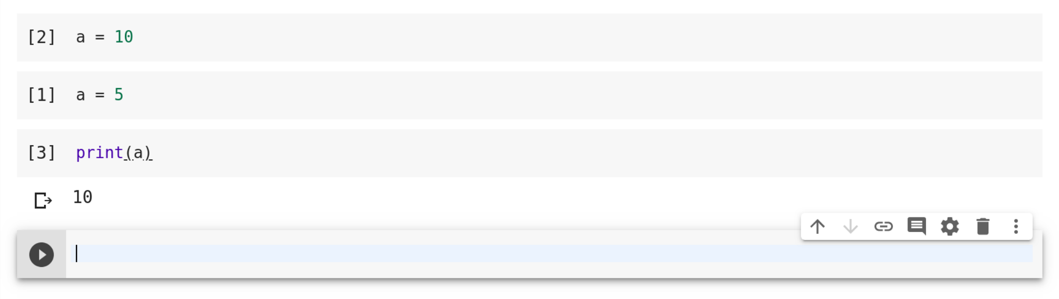
```
[1] from datetime import datetime
```

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

```
Please input your name:
Tan
```

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

Caveats 1: Execution order



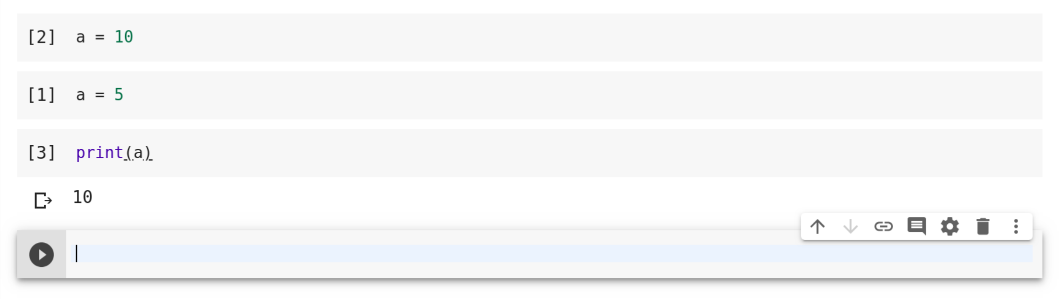
```
[2] a = 10
```

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

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

```
10
```

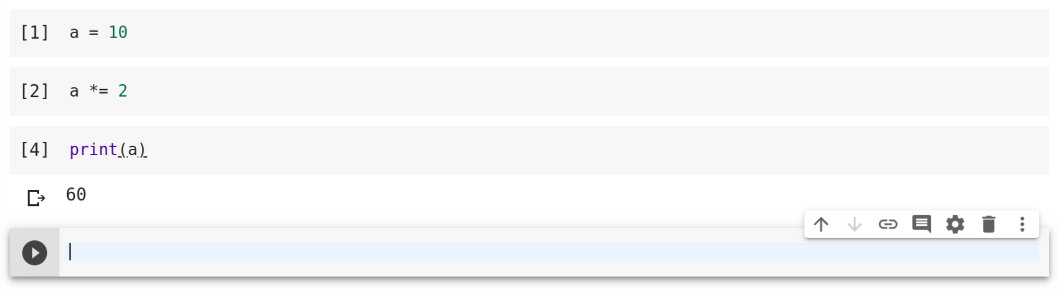
Caveats 1: Execution order



```
[2]  a = 10

[1]  a = 5

[3]  print(a)

     10
```

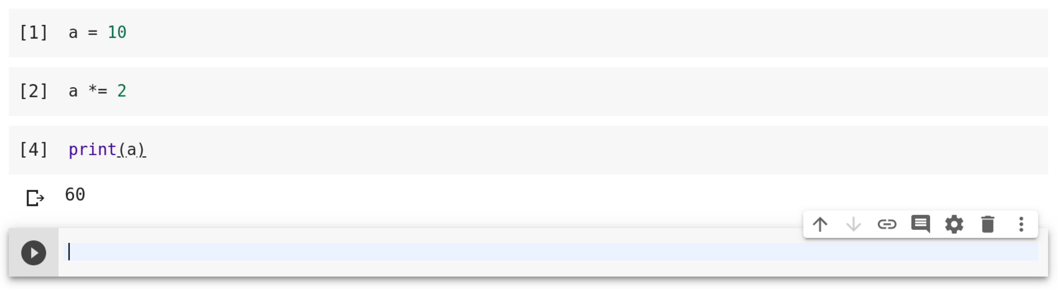
You’ll do a lot of out-of-order code execution!

Caveats 2: Cell edits



```
[1] a = 10

[2] a *= 2

[4] print(a)

    60
```

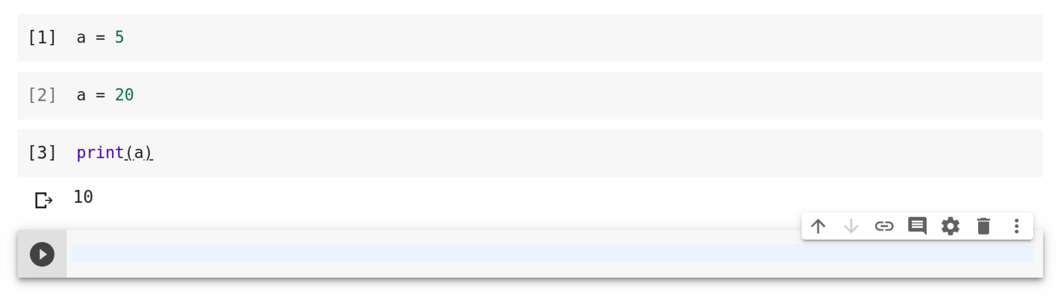
Caveats 2: Cell edits



```
[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



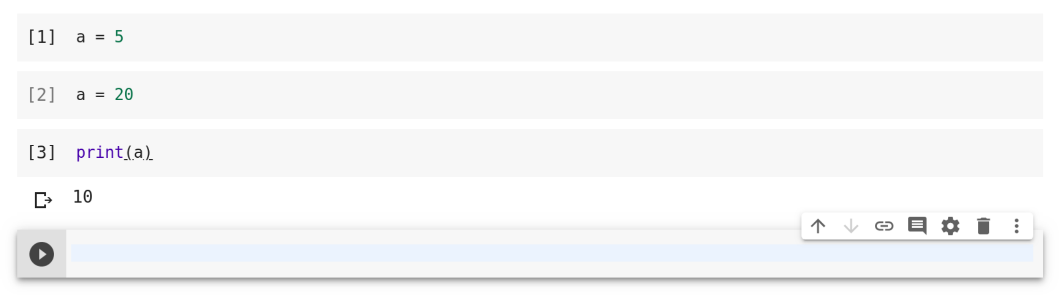
```
[1]  a = 5
```

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

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

```
10
```

Caveats 2: Cell edits



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

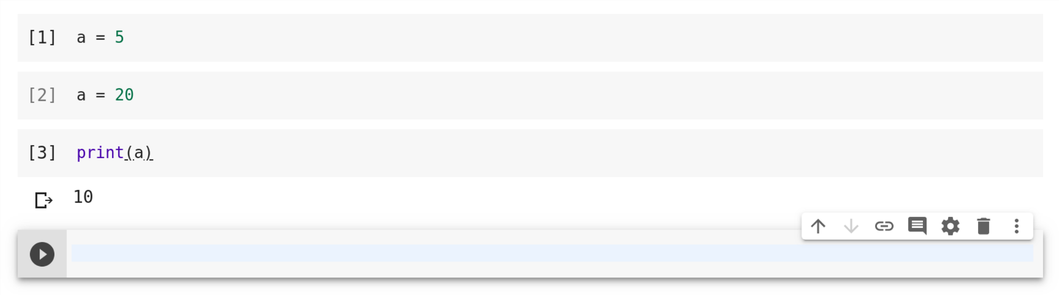
```
[2]  a = 20
```

```
[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

Caveats 3: Be neat and tidy

Sirakorn's Workflow Demo

Python Data Structures

Lists

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

Lists

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

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

Lists

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

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

Lists

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

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

Lists

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

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

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

Lists

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

Accessing list

Lists

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

Accessing list

Lists

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

Accessing list

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

Lists

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

Accessing list

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

List Functions

```python
vowels = ["a", "e", "o", "u"]

# Get a's length
len(a)
# Append the new element to the end of a
a.append("y")
# Deletes the first occurence of the element from a
a.remove("y")
# Inserts the item into a list with a specified index
a.insert(2, "i")
```

List Functions

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

Dictionary

```python
names = {
    "Cherprang": "Cher",
    "Manipa": "Khamin",
    "Jiradapa": "Pupe"
}

kami_nickname = names["Manipa"]
```

Dictionary

```
names = {
    "Cherprang": "Cher",
    "Manipa": "Khamin",
    "Jiradapa": "Pupe"
}

kami_nickname = names["Manipa"
   ]
```

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

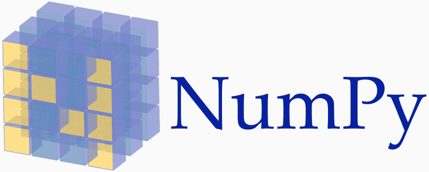
Dictionary

```
names = {
    "Cherprang": "Cher",
    "Manipa": "Khamin",
    "Jiradapa": "Pupe"
}

kami_nickname = names["Manipa"
   ]
```

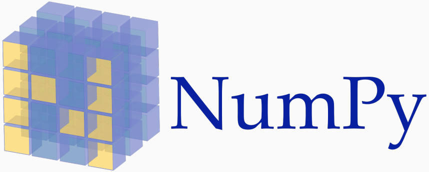
- **Dictionaries** store values in a **key-pair** format.
- From the positional index, dictionary takes the key as an index instead.

NumPy

NumPy

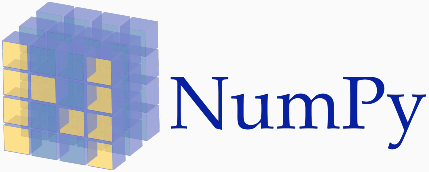


NumPy



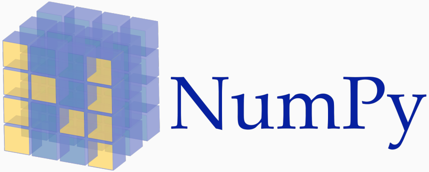
- NumPy is a powerful library for mathematical computation in Python

NumPy



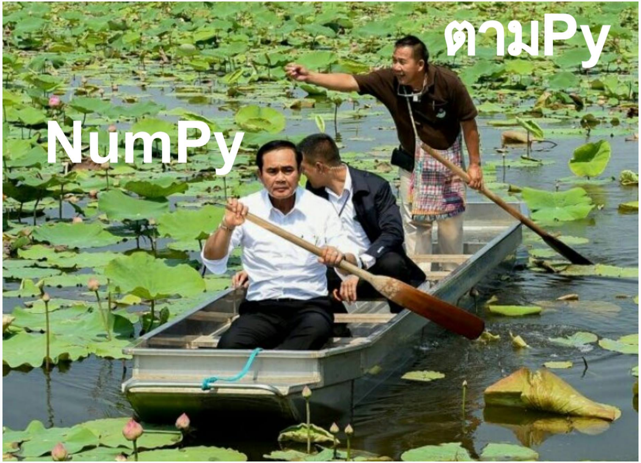
- NumPy is a powerful library for mathematical computation in Python
- It offers wide range of tools from data structures to advanced functions and operations

NumPy



- 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!**

NumPy



Courtesy: Rebellious Professor,

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

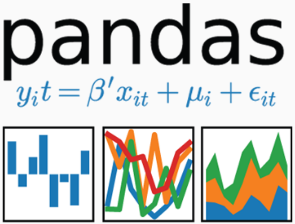
Let’s go to Notebook!

I’m too lazy to cover the contents twice...

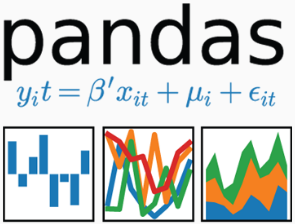
Pandas

Pandas

Pandas

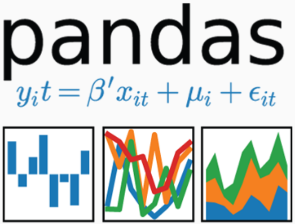


Pandas



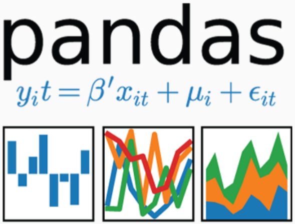
- Tabular-like structure

Pandas



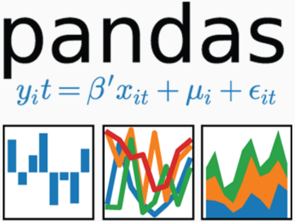
- Tabular-like structure
- High performance

Pandas



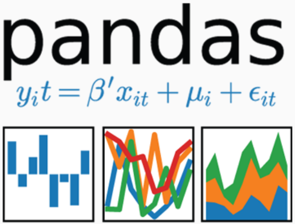
- Tabular-like structure
- High performance
- Easy to use

Pandas



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

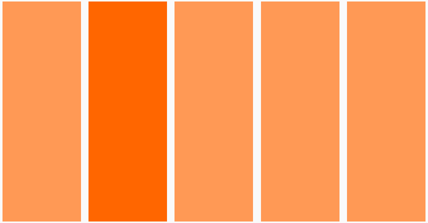
Pandas



- 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

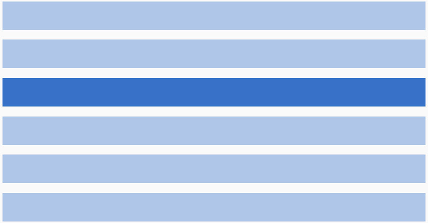
Series and DataFrame

DataFrame



Series

DataFrame



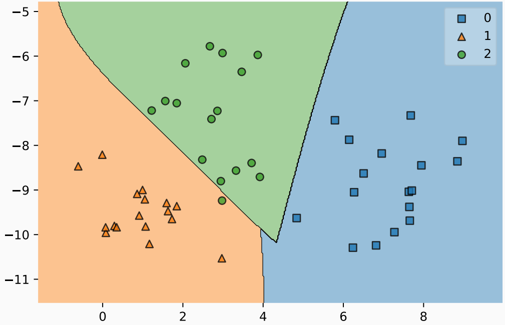
Series

Pandas

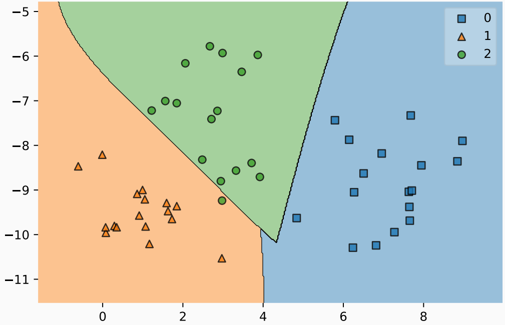
Let’s go to Notebook!

Data Visualisation

Data Visualisation

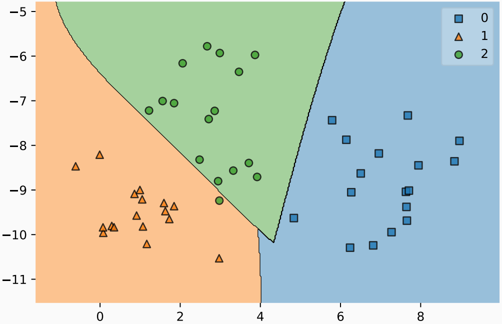


Data Visualisation



Why Visualise?

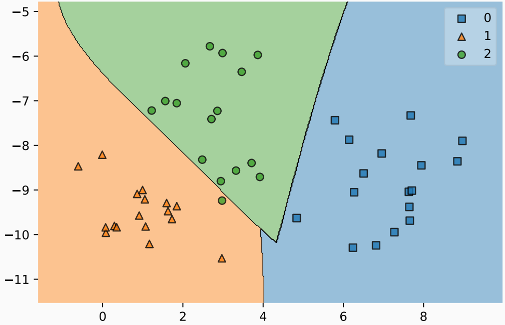
Data Visualisation



Why Visualise?

- Our visual system is so great!

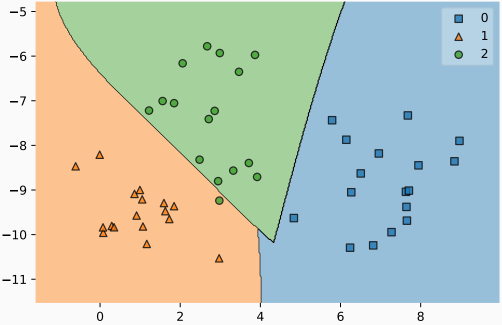
Data Visualisation



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

What if we don't visualise?

The Datasaurus Dozen

`https://www.autodeskresearch.com/publications/samestats`

Matplotlib

Matplotlib

Matplotlib

- feature-rich plotting library in python

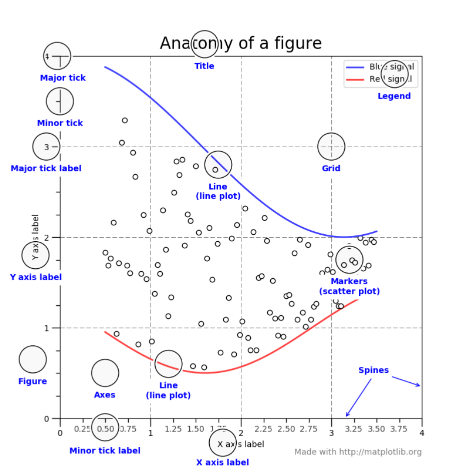
Matplotlib

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

Matplotlib

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

Matplotlib



Source: `https://matplotlib.org/tutorials/introductory/usage.html`

Matplotlib

Let’s go to Notebook!