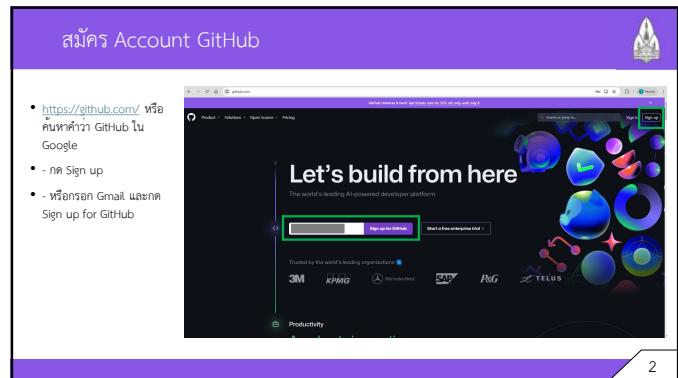
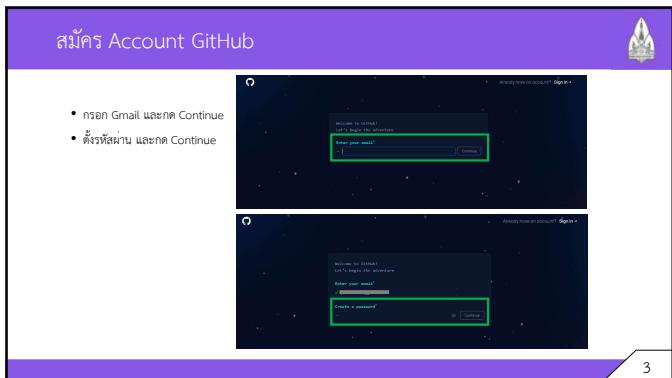




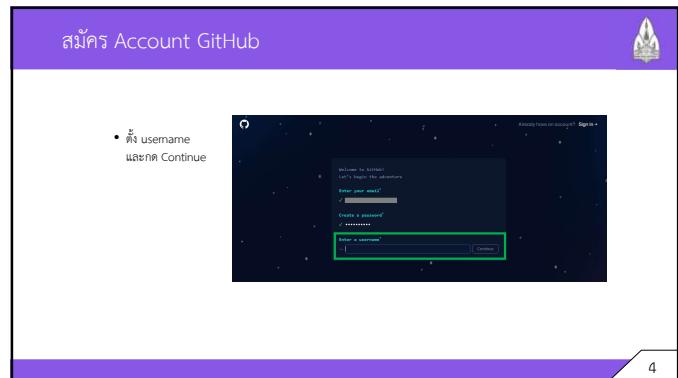
1



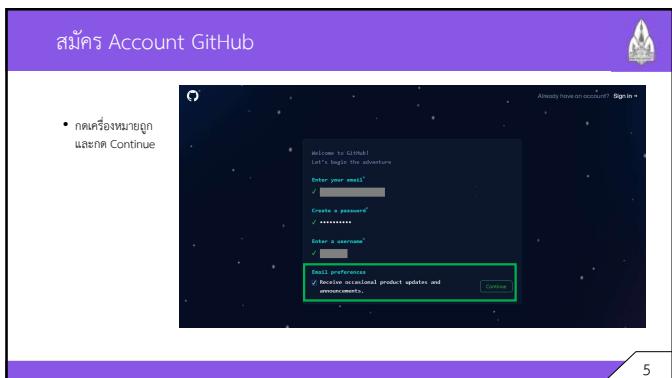
2



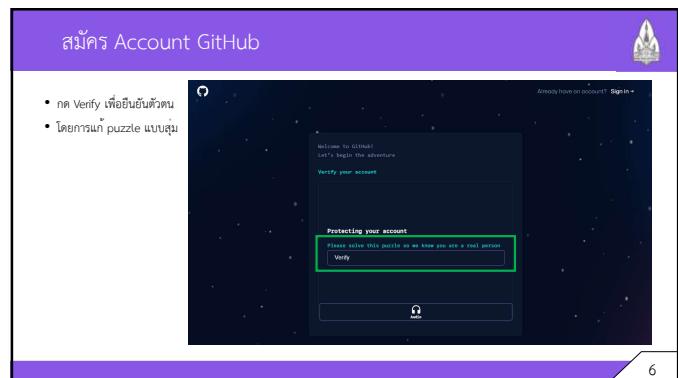
3



4

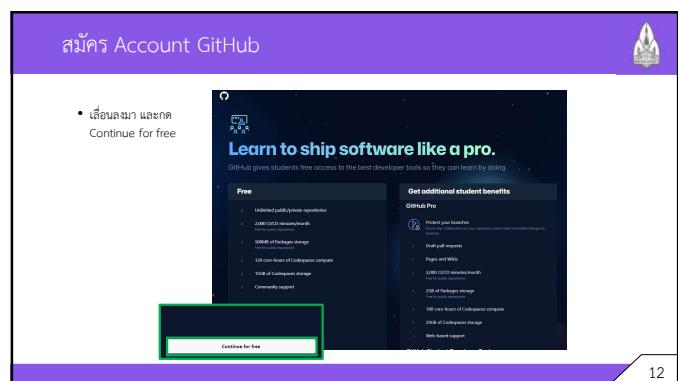
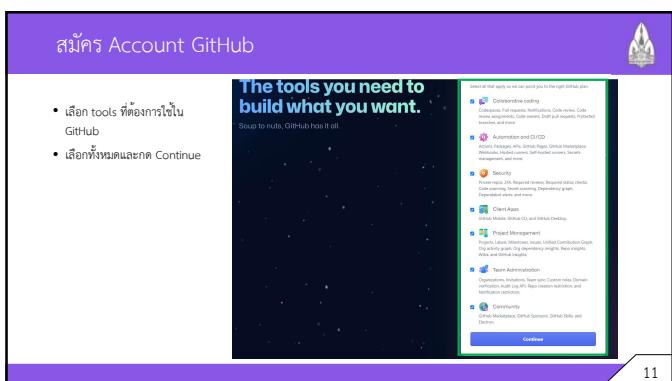
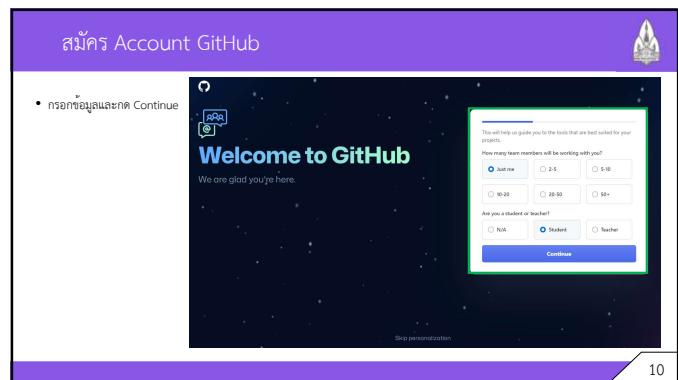
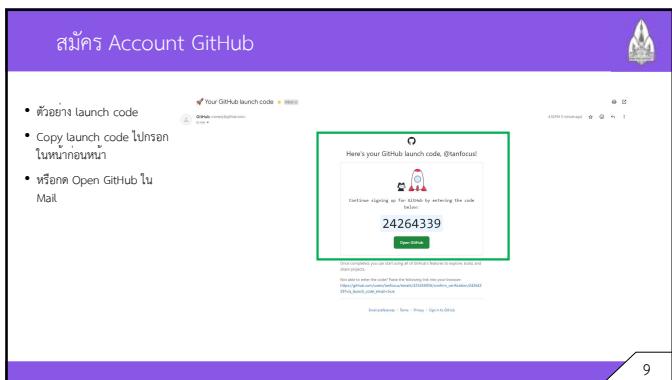
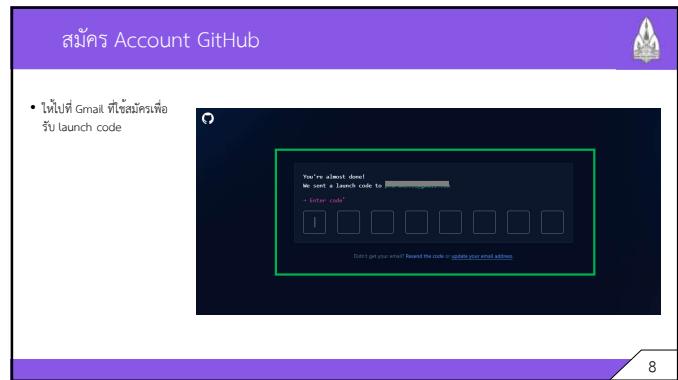
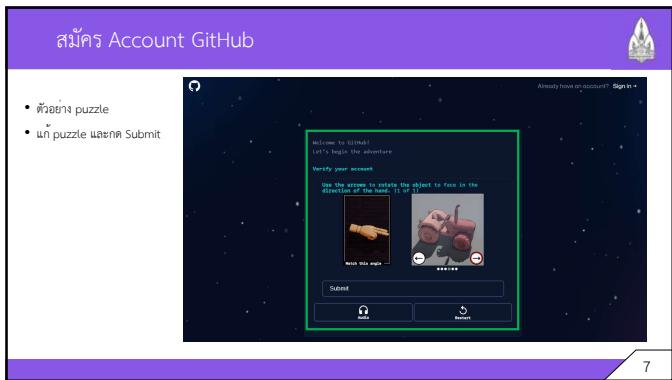


5



6

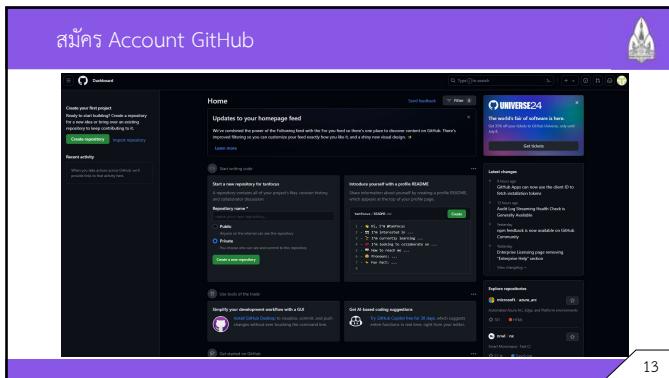
1



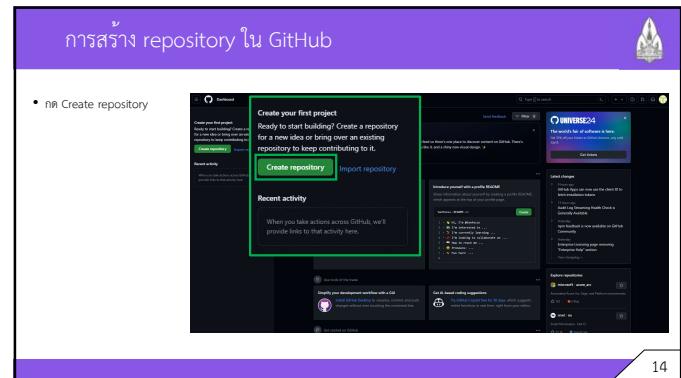
11

12

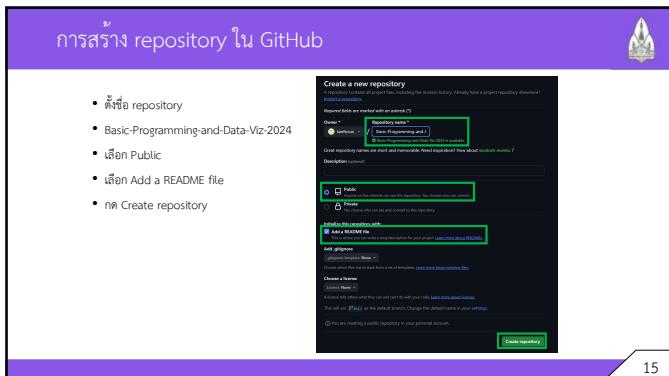
2



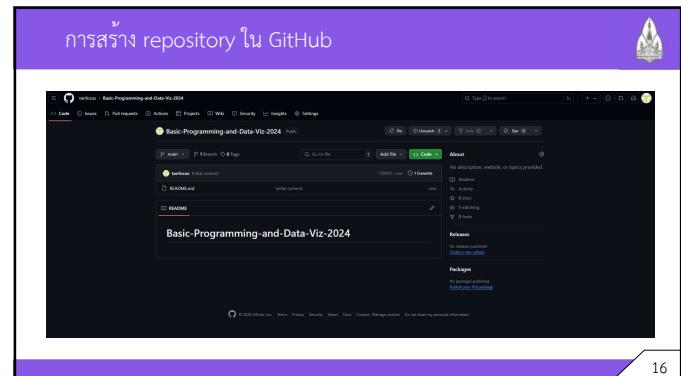
13



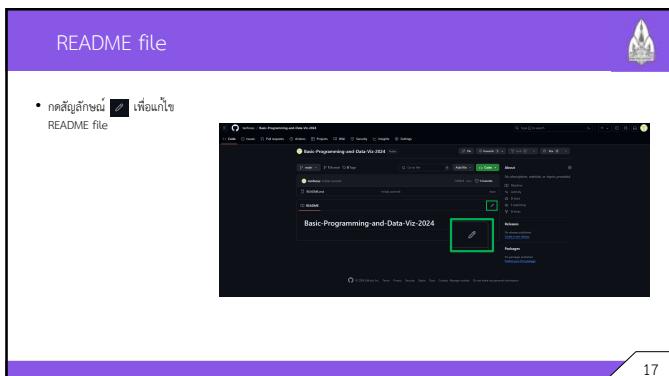
14



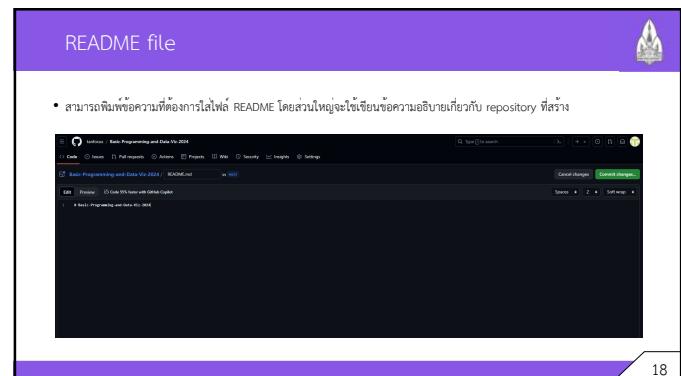
15



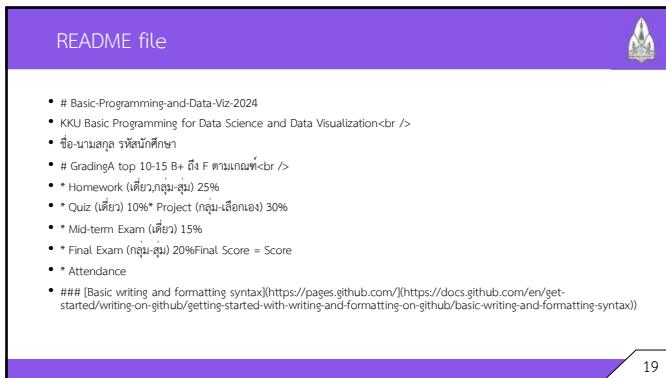
16



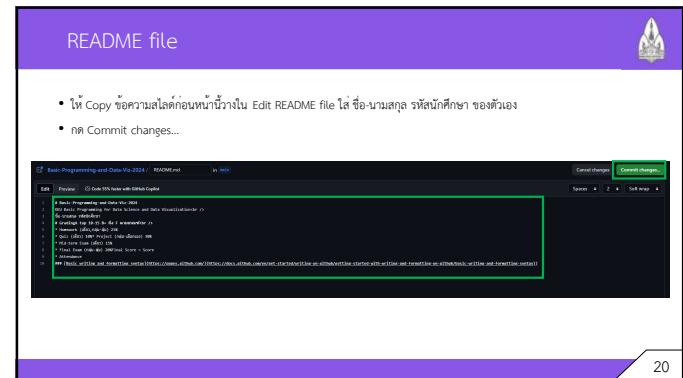
17



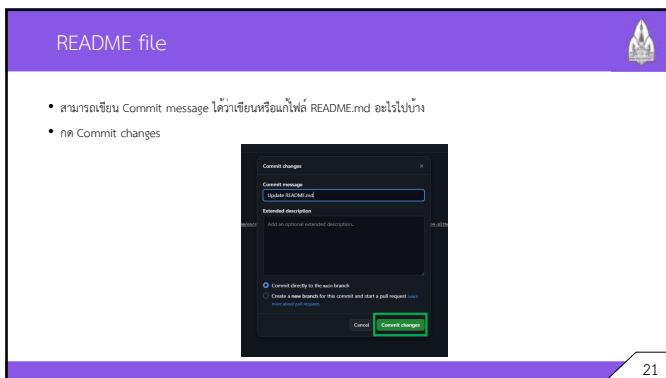
3



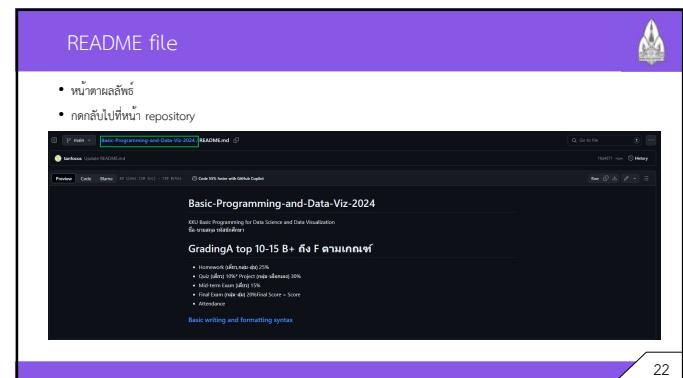
19



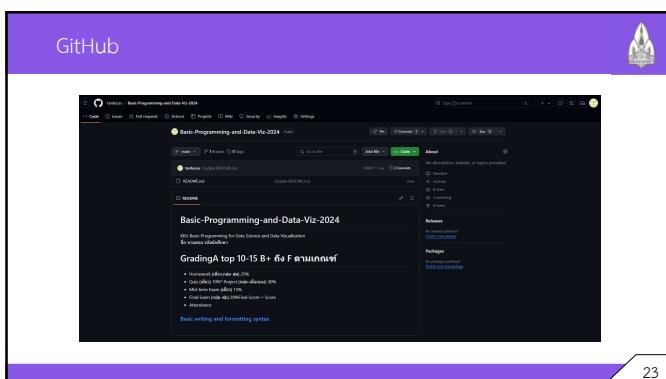
20



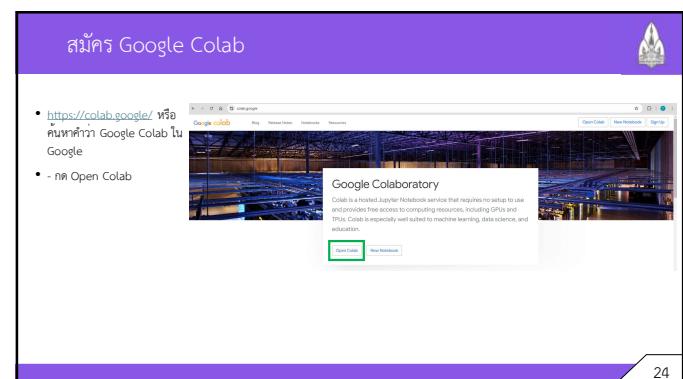
21



22

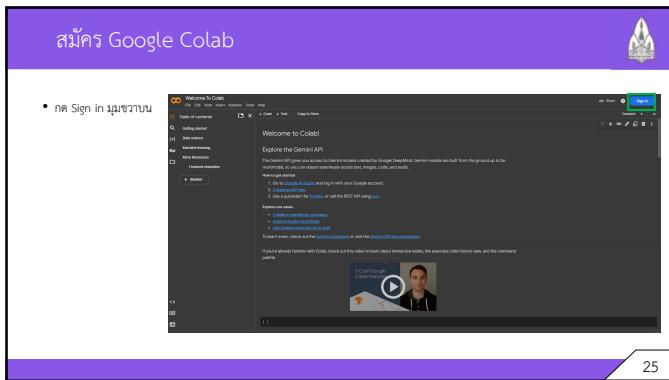


23

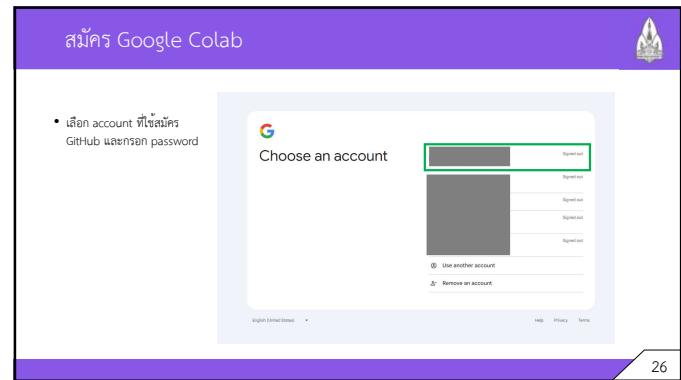


24

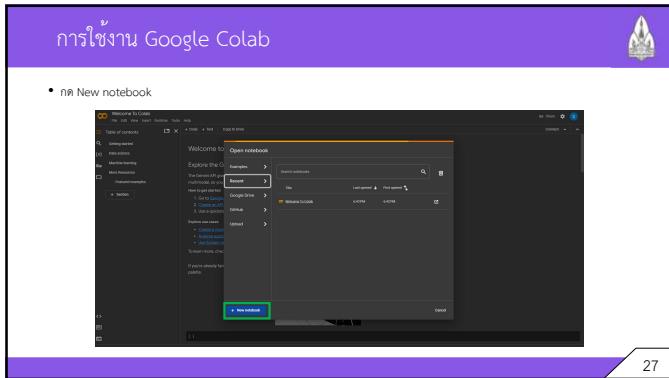
4



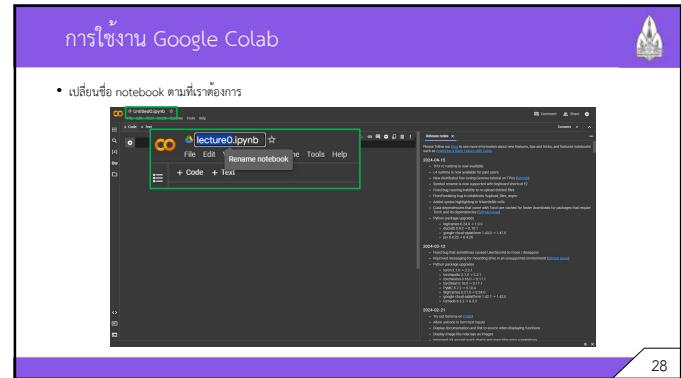
25



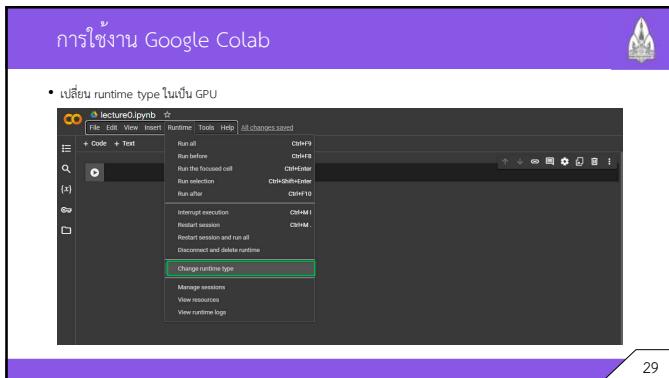
26



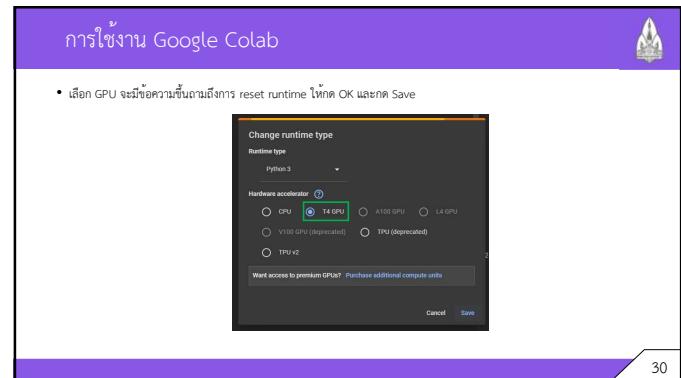
27



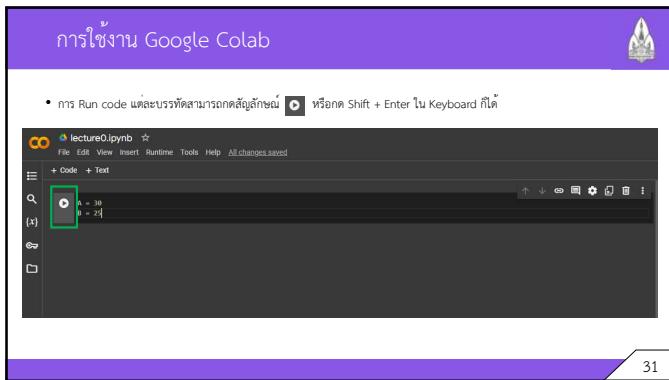
28



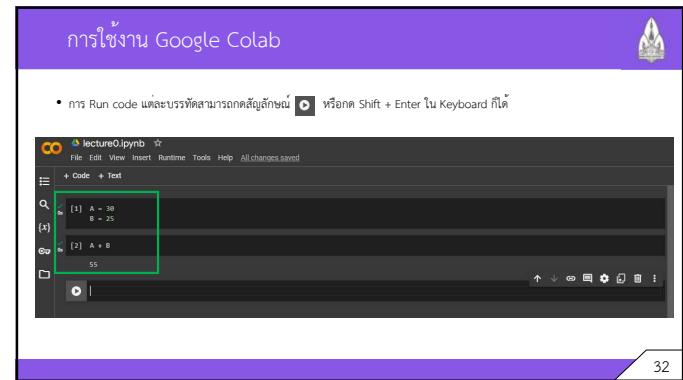
29



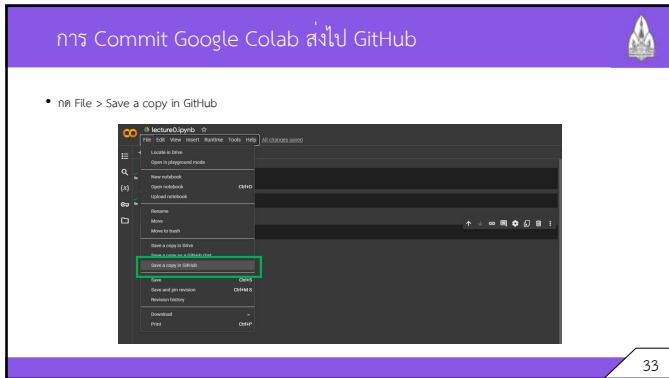
30



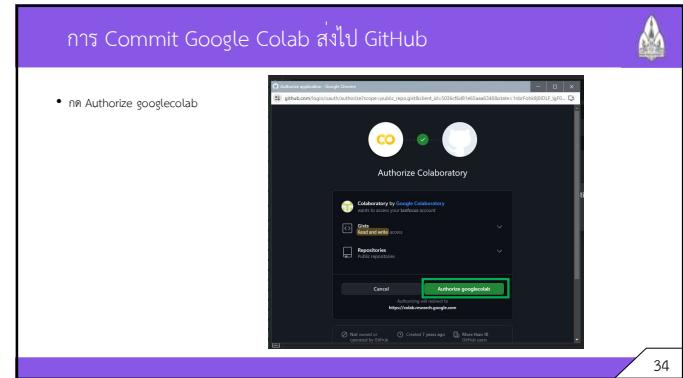
31



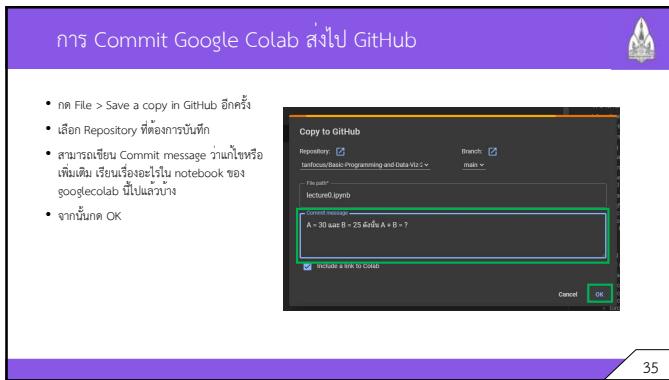
32



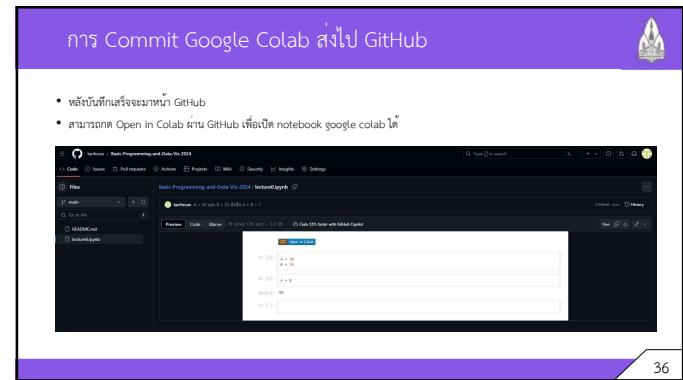
33



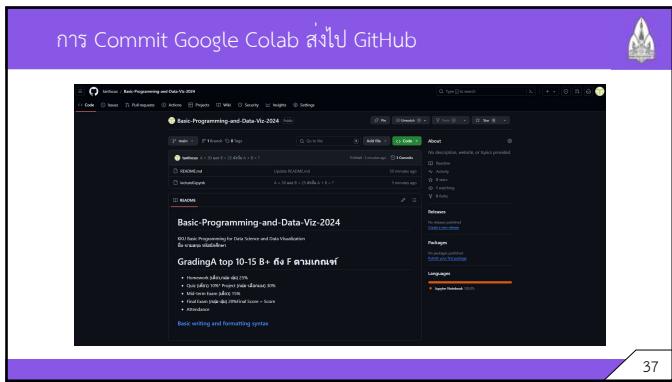
34



35



36



37

Class period 1

บทที่ 2 ตัวแปรและโครงสร้างข้อมูลของภาษาไพธอน

Basic python
python101

1

Variables คืออะไร

- Variables คือตัวแปรที่ใช้เก็บข้อมูล ยกเว้นอย่างกำหนดตัวแปร pi เก็บค่า pi = 3.14159265359 ไม่ต้องพิมพายาให้ตัวแปร pi ที่กากหนาไว้แล้วแทน
- หลักการตั้งชื่อตัวแปรเป็นอย่างไร

 - ตัวอักษร
 - ภาษาอังกฤษ
 - ใช้ตัวเลขได้แต่ห้ามใช้ตัวอักษร
 - ห้ามห้ามระดับ
 - ห้ามตั้งชื่อตัวแปรที่ซ้ำกับชื่อฟังก์ชันต่างๆ (def, for, range, etc.)

2

ชนิดของตัวแปร

- int : ตัวเลขจำนวนเต็ม เช่น a = 10
- float : จำนวนจริง (ทดนิยม) เช่น b = 10.0
- ตัวอักษร (char (character)) ข้อความ (text หรือ string) เช่น c = "ธนาพงษ์"
- ตัวเลขที่เป็น string ไม่สามารถบวก ลบ คูณ หาร กับตัวเลขได้ เช่น d = '10'

3

variable casting (การเปลี่ยนชนิดของข้อมูล)

- กรณีต้องใช้ข้อมูลที่มานามาจากที่อื่น เราสามารถเปลี่ยนชนิดของข้อมูลตามที่เราต้องการใช้งานได้ โดยการ
- กำหนดชนิดของข้อมูลที่ต้องการเปลี่ยนวันนี้ตัวแปร
- int(d)
- float(d)
- str(d)

4

Operation (การเอาตัวแปร 2 ตัวมาทำอะไรกัน)
(Operators + , - , * , / , %)

- การบวก
- การลบ
- การคูณ
- การหาร
- การหารแบบ % เครื่องหมาย modulo คือการหารเอาเศษ

5

คำสั่ง print แบบพิเศษ (การ format string)

- แบบที่ 1 คำสั่ง print พื้นฐาน เช่น print(ตัวแปร) ลิสต์คือถูกตั้งในวงล้อนี้คือ ตัวแปร หรือ string
- แบบที่ 2 การเพิ่มข้อความที่ต้องการนอกจากตัวแปร คือการเพิ่ม f หน้า 'string' และใช้ {} ใส่ code เช่น print(f% คือการหารผลหาร เช่น 7963 = (7%3)')
- หาก คือการเขียนบรรทัดใหม่
- \\" ใน การตัด text แยกใน code และ code จะถูกอ่านเป็น 2 ชิ้น

6

1

DATA STRUCTURE (โครงสร้างข้อมูล)

- List คือ การเก็บตัวแปรหลายตัวมาเรียงกัน สามารถสร้างได้ 2 แบบ ดังนี้
- แบบที่ 1 list_a = []
- แบบที่ 2 list_a = list()
- list_b = [1,5,'V']
- ล่วงตัวบีที่อยู่ใน list มีความสำคัญ สำหรับใน list เริ่มจาก 0,1,2,...
- อย่างเดียวกันกับ list_b ตัวที่ 1 ให้เขียน list_b[1] คือ 5

7

append() การเพิ่มสมาชิกเข้าใน list

- คำสั่ง .append ดาวน์โหลดค่าที่ต้องการเพิ่มใน () สามารถเพิ่มสมาชิกเข้าใน list ที่ต้องการได้ เช่น
- list_b.append('u') ต่อมาดอ print(list_b)
- [1, 5, 'V', 'u'] จะเห็นว่า 'u' ถูกเพิ่มเข้ามาใน list_b
- คำสั่ง .pop() ใช้สำหรับดึงสมาชิกที่สุดท้ายออกจาก list
- list_b.pop() ต่อมาดอ print(list_b)
- [1, 5, 'V'] จะเห็นว่า 'u' ถูกถอดออกจาก list_b

8

String > list of characters

- คำสั่ง len() คือคำสั่งตรวจสอบความยาวของ list (จำนวนสมาชิก)
- String มีค่าเป็น list เช่น
- t = 'python is easy'
- len(t) จะเท่ากับ 14 นับตามจำนวนตัวอักษรและวรรคหรือช่องว่างก็จะถูกนับ

9

access a member of a list (list&string)

- สมาชิกของ list จะเริ่มนับจาก 0 , -1 คือสมาชิกตัวสุดท้าย
- t = 'python is easy'
- t[1] จะเท่ากับ y คือสมาชิกตัวที่ 1 เริ่มนับ 0 เท่ากับ 0
- t[-1] จะเท่ากับ y คือสมาชิกตัวสุดท้าย

10

List slicing

- List slicing สามารถทำได้โดยใช้ colon [a:b] -> [a,b]
- ตัวที่อยู่ท่าใน [] เรียกว่า index(ตัวชี้)

$-2 \leq x < 3$ เช่นนี้คือ $\{x | -2 \leq x < 3\}$ หรือเช่นนี้คือ $(-2, 3)$

$x < -2$ หรือ $x \geq 3$ เช่นนี้คือ $\{x | x < -2 \text{ หรือ } x \geq 3\}$ หรือเช่นนี้คือ $(-\infty, -2) \cup [3, \infty)$

11

ตัวอย่าง

- t = 'python is easy'
- t[7:9]
- ถ้าตัวว่างหน้า : หมายความว่า เริ่มตั้งแต่วันแรก เช่น t[6] คือ python
- ถ้าตัวว่างหลัง : หมายความว่า ไปจนถึงตัวสุดท้าย เช่น t[10:] คือ easy
- ตั้งนั้น t[7:9] จะเท่ากับตัวที่ 7 ใน t ไปจนถึงตัวที่ 8 เพราะ 9 คือจุดจบ คือ is

12

2

ตัวอย่าง

- t = 'python is easy'
- t[-2:] โดยค่าส่ง : ตัวที่ส่อง จะใช้กำหนด step(การกระโดยค) ตั้งนั้น t[-2:] step=2
- ผลลัพธ์จะได้ pto ses
- list_a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- list_a[-2:]
- ผลลัพธ์จะได้ [0, 2, 4, 6, 8]
- t[2:-2] เช่นบันทึกว่า 2 โดยกำหนด step เป็น 2
- ผลลัพธ์จะได้ to ses

13

การนำ list มาต่อกัน

- สามารถนำ list มาต่อกันได้ด้วยการเพิ่ม + ตามด้วยค่าที่ต้องการต่อ
- string ต่อ string
- t = 'python is easy'
- t + ?? จะทำกับ 'python is easy??'
- list_b = [1, 5, 'v']
- ไม่สามารถนำ list ปักดิมต่อ กับ string ได้ เช่น t + list_b ไม่สามารถทำได้
- list ต่อ list
- list_b + list_a จะทำกับ [1, 5, 'v', []]

14

13

14

split string การแบ่ง string ตามสัญลักษณ์ที่กำหนด

- สามารถแบ่งได้โดยการเติม .split ตามด้วยสัญลักษณ์ที่ต้องการใน () เช่น
- t = 'python is easy'
- t.split(' ') หมายความว่า แบ่งชื่อความในตัวแปร t โดยมีสัญลักษณ์ ' ' คือของว่าง ตั้งนั้นจะได้
- ['python', 'is', 'easy']
- time = '12:30:15'
- time.split(':') หมายความว่า แบ่งชื่อความในตัวแปร time โดยมีสัญลักษณ์ ':' ตั้งนั้นจะได้
- ['12', '30', '15']

15

วิธีรวม string กลับ

- t = 'python is easy'
- t_sp = t.split(' ')
- print(t_sp) = ['python', 'is', 'easy']
- สามารถรวมกลับได้โดยการนำทุกตัวอักษรที่ต้องการ ตามด้วย .join ตามด้วยตัวแปรที่ต้องการรวมกลับไปใน()
- t_join = ''.join(t_sp)
- print(t_join) = python is easy

16

15

16

Homework class period 1

คำนวณเวลาเป็นวินาทีของเวลาต่อไปนี้โดยใช้คำสั่ง split() ช่วย และ print ออกมาก่อน

- 12:30:15
- 13:41:07
- 12:53:15
- 00:59:25
- 11:11:11
- 16:06:09

17

17

Class period 2

บทที่ 2 ต้นแบบและโครงสร้างข้อมูลของภาษาโปรแกรม (คศ)

Function_Loop_Condition 1

1

ทบทวนคำที่แล้ว

```

• name = 'กัลยาณ์'
• surname = 'ເນື້ອ'
• ID = '603021866-7'

• b= f'{name} ພະນາກ (surname) ພະ (ID)'
• print(b)

```

1

Function

- ทำหน้าที่รับ input มาประมวลผลออกมาเป็น output
- ยกตัวอย่างเบื้องต้นพื้นที่นี่คือฟังก์ชันคณิตศาสตร์ $f(x) = y$ เช่น input ตัวแปร x เท่าไป f คือ function เพื่อให้ function ประมวลผลแล้ว output ออกมาคือ y

2

3

สัญลักษณ์ที่จำเป็นต้องใช้ในการเขียน Program

- backtick (`) ==> กด ~ ค้าง, alt - 9+6 (full keyboard with number)
- tilde (~)
- curly bracket ({ })
- square bracket ([])

3

4

Function template

```

def ຄົກກຳທັງໝົດ
def function_name(_Input_):
    do_something_with _Input_ to get _Output_
    return _Output_

```

การเน้นบรรทัด (indent) หรือการกด tab ก่อนพิมพ์บรรทัดต่อไปจาก def เพื่อบอกว่าตอนนี้ของโปรแกรม

- function มีส่วนสำคัญทั้งหมด 4 ส่วน
- ของ python ว่าจะเขียนฟังก์ชัน ซึ่งจะใช้ def function_name(): (ขาดไม่ได้)
- ก้าบผลตัวแปรที่จะเป็น input _Input_ (ขาดไม่ได้)
- ส่วนประมวลผล do_something with _Input_ to get _Output_ (ขาดไม่ได้)
- ส่วน output return _Output_ (ขาดไม่ได้)

4

5

ตัวอย่างการเขียน normal function

```

def print_name(surname, ID, name):
    st = f'{name} ພະນາກ (surname) ພະ (ID)'
    return st

```

- ฟังก์ชันชื่อ print_name
- มี input 3 ตัวแปร คือ surname, ID, name
- ส่วนประมวลผล st = f'{name} ພະນາກ (surname) ພະ (ID)' คือให้เขียน string โดยใช้ {} ในตัวแปรที่ input และเก็บไว้ในตัวแปร st
- Output ให้ return st

6

1

ตัวอย่างการเขียน normal function

- การใช้งานให้เขียนชื่อฟังก์ชันและค่า input ตามที่ def ไว้
- แบบที่ 1 `print_name('ลุงหนู', '64xxxxxx', 'มงคล')`
- ผลลัพธ์จะได้ ชื่อ ลุงหนู นามสกุล อินทร์ รหัส 64xxxxxx'
- แบบที่ 2 `print(print_name(name='ลุงหนู', surname='มงคล', ID='603021855-2'))`
- ผลลัพธ์จะได้ ชื่อ กาญจนานา นามสกุล ประสาดุน รหัส 603021855-2

7

ฟังก์ชันไม่จำเป็นต้องมี output หรือการ return

```

def print_name2(surname, ID, name):
    st = f'{name} {surname} พล({ID})'
    print(st) # หมายความว่า print แล้ว return แล้ว

```

- `print_name2(name='ลุงหนู', surname='มงคล', ID='603021855-2')`
- ผลลัพธ์จะได้
- ชื่อ กาญจนานา นามสกุล ประสาดุน รหัส 603021855-2

8

ฟังก์ชันไม่จำเป็นต้องมี input

```

def Pi():
    return 3.14159265359

Pi()**2 # หากต้องการค่าที่มีรีรันให้กับ Pi * r**2

```

- ผลลัพธ์จะได้
- `12.56637061436`
- ฟังก์ชันเป็นค่า 2 อย่าง คือ
- 1. def ชื่อฟังก์ชัน:
- 2. เร้นวรรค (indent) ตามด้วยรูปแบบเดิม

9

การกำหนดค่า default ให้กับฟังก์ชัน

- input ของ function ใน python มี 2 แบบ
- 1. input ที่จัดเป็นค่าได้
- 2. input ที่ไม่จัดเป็นค่าได้ (เช่น default)
- เรื่อง input ที่อ่อนไหวต่อชนิด

```

def print_2lines_default(name, surname, ID, grade='F'):
    st = f'{name} {surname} พล({ID})'
    print(st)
    st2=f'เกรดวิชา Data Viz >>> {grade}'
    print(st2)

grade='F' เป็นการกำหนดค่า default ให้ตัวแปร grade เป็น F

```

10

ตัวอย่างการใช้งานการกำหนดค่า default ให้กับฟังก์ชัน 1

```

print_2lines_default('ลุงหนู', 'มงคล', '613020551-8')

```

- ผลลัพธ์จะได้
- ชื่อ ลุงหนู นามสกุล พล รหัส 613020551-8
- เกรดวิชา Data Viz >>> F
- จะเห็นว่า ไม่มีการใส่ค่าวิ่ง grade ใน input แต่ผลลัพธ์ได้ เกรดวิชา Data Viz >>> F
- เพราะในฟังก์ชันมีการกำหนดค่า default ให้กับตัวแปร grade เป็น F

11

ตัวอย่างการใช้งานการกำหนดค่า default ให้กับฟังก์ชัน 2

```

print_2lines_default('ลุงหนู', 'มงคล', '613020551-8', 'A')

```

- ผลลัพธ์จะได้
- ชื่อ ลุงหนู นามสกุล พล รหัส 613020551-8
- เกรดวิชา Data Viz >>> A
- สามารถ input ค่าตัวแปร grade แบบปกติได้

12

งานในห้อง กลับไปสร้างฟังก์ชันใน HW python101

- ให้สร้างฟังก์ชันคำนวนเวลาเป็นวินาทีของเวลาได้ และ print ออกมาให้สวยงาม

13

LOOP การวนซ้ำ

- for each_member in listA :
 - do_something
- for เป็นคำใช้บอท python ว่าเราเก็บสิ่งซึ่งเป็น loop โดย for จะวนดึงสมาชิกจาก listA มาทำ process do_something

14

ตัวอย่าง LOOP การวนซ้ำ

```
for i in [1,2,3] :
    o = i**2
    print(f'this member = {i} after process = {o}')

•หมายความว่า ให้วนลูปอ่อนค่าสมาชิกใน list [1,2,3] โดยแทนค่าสมาชิกที่อยู่ด้วยตัวแปร i
•ภายในลูป นำตัวแปร i ยกกำลัง 2 เก็บค่าไว้ในตัวแปร o และ print string
•ผลลัพธ์จะได้
•this member = 1 after process = 1
•this member = 2 after process = 4
•this member = 3 after process = 9
```

15

Homework class period 2 กลับไป python101

- ให้ สร้าง list ของ เม็ด ([12:30:15,'13:41:07,...]) และวนลูปเรียกฟังก์ชันคำนวนเวลาเป็น วินาที

16

Patterns for writing clean code in Python

- 1. Use long descriptive names that are easy to read.
- 2. Use descriptive intention revealing names.
- 3. Avoid using ambiguous shorthand.
- 4. Always use the same vocabulary.
- 5. Start tracking codebase issues in your editor.
- 6. Don't use magic numbers.
- 7. Be consistent with your function naming convention.
- 8. Functions should do one thing and do it well.
- 9. Do not use flags or Boolean flags.
- 10. Do not add redundant context.
- <https://www.freecodecamp.org/news/how-to-write-clean-code/>
- <https://dev.to/alexomeyer/10-must-know-patterns-for-writing-clean-code-with-python-56bf>

17



Class period 3

บทที่ 3 โปรแกรมนี้เขียนมา干什么ในภาษา Python
Function_Loop_Condition 2

1



ตัวอย่าง LOOP การวนซ้ำ

```
for i in 'Thanapong':
    print(f'{i} -> /')
```

- หมายความว่า ให้วนลูปอ่านค่าสมาชิกใน string 'Thanapong' โดยแทนค่าสมาชิกที่อ่านทีละตัวด้วยตัวแปร i
- การในลูป นำตัวมันไปให้ใน f(i) -> '/' และ print string

2



ตัวอย่าง LOOP การวนซ้ำ

- ผลลัพธ์จะได้ อยู่ที่นี่ว่า loop ทำการ print string ปกติ ตามค่า i คือตัวอักษรที่อยู่ใน string ที่ส่งตัววนลูปมาจากส่วนที่ 0 ไปจนถึงสิ้นสุดท้าย
- T -> /
h -> /
a -> /
n -> /
a -> /
p -> /
o -> /
o -> /
n -> /
g -> /

3



การสั่ง loop ให้ print แบบไม่เว้นบรรทัด

- โดยคำสั่ง print จะมี key argument และตัว default สำหรับตัวนั้นเองคือ end='\\n'
- print(value, ..., sep=' ', end='\\n', file=sys.stdout, flush=False)

- Prints the values to a stream, or to sys.stdout by **default**.
- Optional keyword arguments:
- file: a file-like object (stream); defaults to the current sys.stdout.
- sep: string inserted between values, default a space.
- end: string appended after the last value, default a newline.
- flush: whether to forcibly flush the stream.

4



ตัวอย่างการสั่ง loop ให้ print แบบไม่เว้นบรรทัด

```
end='\\n'
# ในภาษา python สำหรับการพิมพ์ string หนึ่ง ควรเว้นบรรทัด
# ถ้าต้องการ print แบบไม่เว้นบรรทัด ให้ใส่ input กำหนดค่า end ตัวค่าที่เราต้องการ

for i in 'Thanapong':
    print(f'{i} -> / ', end = '')
# กำหนด end = '' คือไม่ใส่ช่องเว้น ผลลัพธ์จะได้

T -> / ,h -> / ,a -> / ,n -> / ,a -> / ,p -> / ,o -> / ,o -> / ,n -> / ,g -> / ,
```

5



ตัวอย่างการสั่ง loop ให้ print แบบไม่เว้นบรรทัด

```
for i in 'Thanapong':
    print(f'{i} -> / ', end = ',')
# กำหนด end = ',' คือ ใส่สัญลักษณ์ ผลลัพธ์จะได้

T -> / ,h -> / ,a -> / ,n -> / ,a -> / ,p -> / ,o -> / ,o -> / ,n -> / ,g -> / ,
```

6

range() การสร้าง list ตัวเลขแบบอัตโนมัติ

- range() คือคำสั่งที่ใช้สร้าง list ของตัวเลข เช่น
- range5_output = range(5)
- print(list(range5_output))
- หากความว่า ให้สร้าง list ตัวเลขจำนวน 5 ตัว เริ่มจาก 0 และเก็บไว้ในศูนย์ range5_output
- จากนั้น print ทั้งหมด range5_output ในรูปแบบของ list
- ผลลัพธ์จะได้
- [0, 1, 2, 3, 4]

7

ตัวอย่างการใช้งาน range() สร้าง list ตัวเลขในการวนลูป

```

for i in range(100):
    print(' ', end=' ')

```

• หากความว่า ให้วนลูป 100 รอบ โดยอ่านค่าสมาชิกใน range(100) ซึ่งคือ list 0 ถึง 99

• ภายในลูปให้ print(' ', end=' ')

• ผลลัพธ์จะได้

```

  0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99

```

7

8

Key argument ของ range()

- range(stop) -> range object
- range(start, stop[, step]) -> range object
- Return an object that produces a sequence of integers from start (inclusive) to stop (exclusive) by step. range(i, j) produces i, i+1, i+2, ..., j-1. start defaults to 0, and stop is omitted range(4) produces 0, 1, 2, 3. These are exactly the valid indices for a list of 4 elements. When step is given, it specifies the increment (or decrement).
- range() สามารถกำหนดตัวเลขนี้ด้วยการ เริ่ม(start), หยุด(stop) และ step ได้

9

ตัวอย่างการใช้งาน range() ด้วย key argument start, stop, step

```

list(range(1,11))

```

• หากความว่า ใช้ range() กำหนดให้สร้าง list ตัวเลข เริ่มจาก 1 ถึง 10

• ผลลัพธ์จะได้

```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```

```

list(range(-3,20,4))

```

• หากความว่า ใช้ range() กำหนดให้สร้าง list ตัวเลข เริ่มจาก -3 ถึง 19 และให้ step=4

• ผลลัพธ์จะได้

```

[-3, 1, 5, 9, 13, 17]

```

9

10

loop in loop

- สามารถสร้าง loop ภายใน loop ได้
- โดยถ้าหากการหกาน หรืองานหนัก การวนบรรด (indent) โดยสรุปหมายความว่าใน loop1 จะทำงานก่อน
- จากนั้นเมื่อมีการเขียน loop2 ภายใน loop1 loop2 จะถูกบันทึกเป็นรุ่นประมวลผลของ loop1
- ดังนั้น loop2 จะทำงานบุบช่องที่จะวนซ้ำทุกรอบก่อน ถึงจะเป็นรุ่นที่ 1 รอบของ loop1

10

11

ตัวอย่าง loop in loop

```

for loop1 in range(2,5): #(2, 3, 4)
    print(f'now loop1 = {loop1}')
    for loop2 in range(1,13):
        print(loop1, 'x ', loop1, ' = ', loop1 * loop2)
    print('the inner loop is end')

```

• หากความว่า ให้วนลูปอ่านค่าสมาชิกใน range(2,5) ซึ่งคือ [2, 3, 4] ทีจะตัวแทนค่าทั้งหมดของ loop1

• ภายในลูปให้ print('now loop1 = {loop1}') ออกข้อความ ถูกที่ 2

• โดยลูปที่ 2 ให้วนลูปอ่านค่าสมาชิกใน range(1,13) ซึ่งคือ list 1 ถึง 12 ทีจะตัวแทนค่าทั้งหมดของ loop2

• ภายในลูปที่ 2 ให้ print(loop1, 'x ', loop1, ' = ', loop1 * loop2) วนลูปที่ 2 จึงครบ 12

• เมื่อลูปที่ 2 ให้ print('the inner loop is end') จากนั้นจะกลับไปบุนลูปใหญ่ที่จะวนก่อจักรबคณ range(2,5)

11

12

2

ตัวอย่าง loop in loop ผลลัพธ์จะได้

```

• now name1 = 2          • now name1 = 3          • now name1 = 4
• 2 x 1 = 2            • 3 x 1 = 3          • 4 x 1 = 4
• 2 x 2 = 4            • 3 x 2 = 6          • 4 x 2 = 8
• 2 x 3 = 6            • 3 x 3 = 9          • 4 x 3 = 12
• 2 x 4 = 8            • 3 x 4 = 12         • 4 x 4 = 16
• 2 x 5 = 10           • 3 x 5 = 15         • 4 x 5 = 20
• 2 x 6 = 12           • 3 x 6 = 18         • 4 x 6 = 24
• 2 x 7 = 14           • 3 x 7 = 21         • 4 x 7 = 28
• 2 x 8 = 16           • 3 x 8 = 24         • 4 x 8 = 32
• 2 x 9 = 18           • 3 x 9 = 27         • 4 x 9 = 36
• 2 x 10 = 20          • 3 x 10 = 30        • 4 x 10 = 40
• 2 x 11 = 22          • 3 x 11 = 33        • 4 x 11 = 44
• 2 x 12 = 24          • 3 x 12 = 36        • 4 x 12 = 48
• the inner loop is end

```

12

13

loop in function

```

• สามารถอ่านสูบในฟังก์ชันได้ โดยใช้ การบันทุกครั้ง (indent) ก้าวนext ค่าที่การท่าเงิน

def print_feelings(I, friendS, feeling = 'ดีดีง'):
    for friend in friendS:
        print(f'{I} ({feeling}) {friend}')

• หมายความว่า กำหนดฟังก์ชันชื่อ print_feelings
• ถ้า input 3 อย่าง คือ I:friendS:feeling = 'ดีดีง' (default)
• ภายใน def ให้วางสูบค่าตามที่กำหนด friendS (list) ที่จะพัฒนาค่าด้วยตัวแปร friend
• ภายนอก print(f'{I} ({feeling}) {friend}')

```

13

14

ตัวอย่างการใช้งาน loop in function

```

list_friends = ['พี่มู', 'ดายน์', 'แต้ม', 'ฝน', 'พี่เจ็ท', 'น้องแมมป์']
print_feelings('เพื่อน', list_friends)

• กำหนดตัวเมทร์ list_friends เป็นชุด list มีสมาชิก ['พี่มู', 'ดายน์', 'แต้ม', 'ฝน', 'พี่เจ็ท', 'น้องแมมป์']
• ใช้งานฟังก์ชัน print_feelings('เพื่อน', list_friends)
• ซึ่งฟังก์ชันนี้ต้องค่าที่ต้องการ input ตามสืบต้นไปที่กำหนดให้ก่อนสร้างฟังก์ชัน 'เพื่อน' = I, list_friends = friends และตัวไป feeling ไม่ได้ input ค่าต้นไปจะถูกใช้ค่า default คือ 'ดีดีง'


```

14

15

ตัวอย่างการใช้งาน loop in function

```

• ผลลัพธ์ของ print_feelings('เพื่อน',list_friends) ให้ input ค่าพัฒนา feeling จะได้
• เช่น สีคือสี พี่มู
• เจนส์ คือสี ดายน์
list friends = ['พี่มู', 'ดายน์', 'แต้ม', 'ฝน', 'พี่เจ็ท', 'น้องแมมป์']

def print_feelings(I, friendS, feeling = 'ดีดีง'):
    for friend in friends:
        print(f'{I} ({feeling}) {friend}')

• เช่น สีคือสี นั่น
• เช่น สีคือสี พี่เจ็ท
• เช่น สีคือสี น้องแมมป์


```

15

16

ตัวอย่างการใช้งาน loop in function

```

ผลลัพธ์ของ print_feelings('เพื่อน',list_friends, 'รัก') จะได้
• เช่น สี พี่มู
• เช่น สี ดายน์
list friends = ['พี่มู', 'ดายน์', 'แต้ม', 'ฝน', 'พี่เจ็ท', 'น้องแมมป์']

def print_feelings(I, friendS, feeling = 'ดีดีง'):
    for friend in friendS:
        print(f'{I} ({feeling}) {friend}')

• เช่น สี ฝน
• เช่น สี พี่เจ็ท
• เช่น สี น้องแมมป์


```

16

17

conditional คือ เงื่อนไข /ประโยคเงื่อนไข

```

• if condition1 :
    do something
• elif condition2 : #elif #else if
    do another thing
• else:
    do ...

• สามารถกำหนดเงื่อนไขก่อนเข้าห้องทำงานของส่วนประมวลผล โดยหากกันเงื่อนไขของ if condition1 ได้ต่อประมวลผล
  กากไป if จะท่างาน
• ถ้าไม่ผ่านจะไปเงื่อนไขที่ไป elif condition2 ตามลำดับ (elif สามารถมีได้มากกว่า 1)
• แต่ถ้าหากไม่ผ่านเงื่อนไขใดเลย ให้ต่อประมวลผลของ else จะท่างาน


```

17

18

ตัวอย่าง Loop แบบบปกติ

```

list friends = ['พี่บุญ', 'พ่อ', 'แม่', 'พี่น้อง', 'น้องสาว']

for name1 in list_friends :
    for name2 in list_friends :
        print(name1, 'รัก', name2)

• วนลูป for ปกติเพื่อ print(name1, 'รัก', name2) เมื่อไหร่ผลลัพธ์จะออกมานี้ซึ่งขึ้นกับว่าจะ
  
```

18

19

ผลลัพธ์จะได้

| | | |
|----------------------|-----------------------|-----------------------|
| • พี่บุญ รัก พี่บุญ | • แม่ รัก พี่บุญ | • พี่น้อง รัก พี่บุญ |
| • พี่บุญ รัก คายน์ | • แม่ รัก คายน์ | • พี่น้อง รัก คายน์ |
| • พี่บุญ รัก แม่ | • แม่ บุญ รัก แม่ | • พี่น้อง รัก แม่ |
| • พี่บุญ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง | • พี่น้อง รัก พี่น้อง |
| • คายน์ รัก พี่บุญ | • แม่ บุญ รัก คายน์ | • แม่ บุญ รัก คายน์ |
| • คายน์ รัก คายน์ | • แม่ บุญ รัก คายน์ | • แม่ บุญ รัก คายน์ |
| • คายน์ รัก แม่ | • แม่ บุญ รัก แม่ | • แม่ บุญ รัก แม่ |
| • คายน์ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง |
| • คายน์ รัก คายน์ | • แม่ บุญ รัก คายน์ | • แม่ บุญ รัก คายน์ |
| • คายน์ รัก แม่ | • แม่ บุญ รัก แม่ | • แม่ บุญ รัก แม่ |
| • คายน์ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง |
| • คายน์ รัก คายน์ | • แม่ บุญ รัก คายน์ | • แม่ บุญ รัก คายน์ |
| • คายน์ รัก แม่ | • แม่ บุญ รัก แม่ | • แม่ บุญ รัก แม่ |
| • คายน์ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง |

19

20

ตัวอย่างการใช้งาน conditional

```

for name1 in list_friends:
    for name2 in list_friends:
        if name1 != name2: #ไม่เท่ากันถึงจะ印出
            print(name1, 'รัก', name2)

• สามารถเพิ่มเงื่อนไข if name1 != name2: ไปส่วนประมวลผลภายในลูป เมื่อ่านเงื่อนไขนี้จะง่ายในการส่วนประมวลผล
  ก่อนใน if คือ print(name1, 'รัก', name2)
  • ตั้งน้ำ้ดตามเงื่อนไข name1 ไม่เท่ากับ name2 โดยจะปรับแต่งที่ไม่เข้ากันเท่านั้น
  
```

20

21

ผลลัพธ์จะได้

| | | |
|----------------------|-------------------|-----------------------|
| • พี่บุญ รัก คายน์ | • แม่ รัก พี่บุญ | • พี่น้อง รัก พี่บุญ |
| • พี่บุญ รัก แม่ | • แม่ รัก คายน์ | • พี่น้อง รัก แม่ |
| • พี่บุญ รัก พี่น้อง | • แม่ รัก พี่น้อง | • พี่น้อง รัก พี่น้อง |
| • พี่บุญ รัก พี่บุญ | • แม่ รัก พี่บุญ | • พี่น้อง รัก พี่บุญ |
| • คายน์ รัก พี่บุญ | • แม่ รัก คายน์ | • แม่ บุญ รัก คายน์ |
| • คายน์ รัก คายน์ | • แม่ รัก คายน์ | • แม่ บุญ รัก คายน์ |
| • คายน์ รัก แม่ | • แม่ รัก แม่ | • แม่ บุญ รัก แม่ |
| • คายน์ รัก พี่น้อง | • แม่ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง |
| • คายน์ รัก คายน์ | • แม่ รัก คายน์ | • แม่ บุญ รัก คายน์ |
| • คายน์ รัก แม่ | • แม่ รัก แม่ | • แม่ บุญ รัก แม่ |
| • คายน์ รัก พี่น้อง | • แม่ รัก พี่น้อง | • แม่ บุญ รัก พี่น้อง |

21

22

ข้อจำกัดของการใช้งาน conditional

```

for name1 in list_friends:
    for name2 in list_friends:
        if name1 == name2:
            'do nothing'
        else:
            print(name1, 'รัก', name2)

• ภายนอกเงื่อนไข จึงเป็นตัวเมื่อส่วนประมวลผล ไม่สามารถปล่อยไว้ได้
  
```

22

23

สัญลักษณ์ที่ใช้ใน conditional

- == คือ เท่ากับ
- != คือ ไม่เท่ากับ
- >= คือ มากกว่าหรือเท่ากับ ใช้ในการผู้ตรวจสอบตัวเลข
- <= คือ น้อยกว่าหรือเท่ากับ ใช้ในการผู้ตรวจสอบตัวเลข
- < คือ น้อยกว่า ใช้ในการผู้ตรวจสอบตัวเลข
- > คือ มากกว่า ใช้ในการผู้ตรวจสอบตัวเลข

23

24

Ture False ใน conditional

- หากสิ่งเงื่อนไข(รี) เป็น True จะทำงานส่วนประมวลผลภายใน if
- if True:
 - print('Yes')
 - ผลลัพธ์จะได้
 - Yes
- หากสิ่งเงื่อนไข(รี) เป็น False ให้ดูไปที่ส่วนประมวลผลภายใน if และ ถ้ามี
 - if False:
 - print('No')
 - ผลลัพธ์จะได้
 - ไม่ก็จะออกผลลัพธ์ เพราะได้ไม่ถูกไปที่ส่วนประมวลผลภายใน if หรือก็คือปล่อยผ่าน

24

ตัวอย่าง Ture False ใน conditional

- ‘ปั๊ะ’ == ‘พี่อึ๊ะ’
- ผลลัพธ์จะได้
- False
- ‘ปั๊ะ’ != ‘พี่อึ๊ะ’
- ผลลัพธ์จะได้
- True

25

Homework class period 3

- เขียน function วบคูณตัดเกรด โดยที่ input เป็นคะแนน(0-100) , output เป็นเกรด (F-A)
- วบคูณตัดเกรด input = [1,50,65,90,101,-5,49]
- กำหนด
- ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 0 แต่น้อยกว่า 50 จะได้เกรด F
- ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 50 แต่น้อยกว่า 55 จะได้เกรด D
- ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 55 แต่น้อยกว่า 60 จะได้เกรด D+
- ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 60 แต่น้อยกว่า 65 จะได้เกรด C
- ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 65 แต่น้อยกว่า 70 จะได้เกรด C+
- ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 70 แต่น้อยกว่า 75 จะได้เกรด B
- ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 75 แต่น้อยกว่า 80 จะได้เกรด B+
- ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 80 จะได้เกรด A
- ผลลัพธ์จะแสดงผลต้องไม่ต่ำกว่า 0 และมากกว่า 100 ไม่นั้นจะเกิด error

26

Class period 4

บทที่ 3 โปรแกรมนเข้าและออกจากภาษาไพธอน (ต่อ)
More_advanced_data_structure 1

1

ທບທວນ List (ເຮືອນໄປແລ້ວ)

Numbers[index]

| | | | | |
|----|---|---|---|---|
| 10 | 1 | 5 | 4 | 0 |
|----|---|---|---|---|

index = 0 1 2 3 4

- List ຕີກ່າງ Array ຢັ້ງໄຈ
- 1. ເກີນຂໍ້ມູນໄດ້ຫລາຍປະເທດ int str float
- 2. ມີ list slicing

2

Dictionary

ສາມາຮດກໍາທັນດີ index ໃຫ້ຄ່າທີ່ຕ້ອງກາໄດ້ ແລະສາມາຮດກໍາທັນດີ index ເປັນຂອງໄກດີ ແດ້ລຳດັບຈະຫາຍໄປ

```
ex_dict = {'a':10, '1':1, 'stat':5}
#curly brackets {} ໂກມມານວິທີນີ້ dict
ex_dict['stat']
```

ຜລືພ໌ອ່ານໄດ້
5

3

ຂໍ້ກໍາທັນດີໃນການໃຊ້ຈານ Dictionary

- 2:'two'
- ຕັ້ງທຳ : ສຶ່ງ key ທີ່ອື່ນ index (ຕົວຢ້ອງ)
- ຕັ້ງຄ່າ : ສຶ່ງ value (ຄ່າ)
- ໄນສານຮຽນໃຫ້ value ໃນການຈັກສັບໄປໆໃຫ້ index

```
ex2_dict['two']
#ຄລືພ໌ອ່ານໄດ້
KeyError: 'two'
```

4

ຕົວຢ່າງການໃຊ້ຈານ Dictionary

```
ex2_dict = {1:'one', 2:'two', 3:'three'}
ex2_dict[2]
```

ຜລືພ໌ອ່ານໄດ້
'two'

5

ການເພີ່ມສາມາຊີກໃນ dict

- ສາມາຮດກໍາທັນດີ index (ຕົວຢ້ອງ) ແລະ value (ຄ່າ) ທີ່ຕ້ອງກາໄດ້ເລືອຍ ເຊັ່ນ
- ex2_dict[0] = 'zero'
- ດັກການເພີ່ມ index 0 ຊື້ເປັນ value 'zero' ເພີ່ມໃນ dict ex2_dict
- ex2_dict
 - ຜລືພ໌ອ່ານໄດ້
 - {0: 'zero', 1: 'one', 2: 'two', 3: 'three'}

6

1

คำสั่งที่ใช้บ่อยของ dict

- `.keys()` คือคำสั่งที่ใช้ตรวจสอบ keys ใน dict ว่ามี index (ตัวอักษร) อะไรบ้าง
- `ex2_dict.keys()`
- ผลลัพธ์จะได้
- `dict_keys([1, 2, 3, 0])`

- `.values()` คือคำสั่งที่ใช้ตรวจสอบ values ใน dict ว่ามี values (ค่า) อะไรบ้าง
- `ex2_dict.values()`
- ผลลัพธ์จะได้
- `dict_values(['one', 'two', 'three', 'zero'])`

6

ตัวอย่างการใช้งานคำสั่ง `.keys()` และ `.values()`

- สามารถใช้งานร่วมกับ loop for เพื่อนำ keys หรือ values ไปใช้งานต่อ เช่น

```

for index in ex2_dict.keys():
    print(index)


```

- ผลลัพธ์จะได้
- 1
- 2
- 3
- 0

***dict เอาจริงๆ ไม่ใช่รวมกับ Dataframe ของ pandas

7

Numpy Array (array n มิติ)

- numpy คือ package ที่ทำงานกับ array ของตัวเลขเท่านั้น (ที่มีคุณสมบัติมาและนิยมใช้)
- numpy ย่อมาจาก number python

8

วิธีเรียกใช้งาน package

- สามารถใช้ `import` ตามด้วยชื่อ package ของ python ที่ต้องการใช้งาน เช่น
- เรียกใช้งาน numpy
- `import numpy`

- สามารถใช้ `as` ต่อท้ายชื่อ package เพื่อตั้งชื่อใหม่ให้ง่ายต่อการใช้งาน เช่น
- `import numpy as np`

9

การแปลง list ให้เป็น numpy array

- แปลง list ให้เป็น array 2 มิติ ด้วยคำสั่ง `numpy.array()`
- `ex_2d_array = numpy.array([[5.2, 3.0, 4.5], [9.1, 0.1, 0.3]])`
- `print(ex_2d_array)`

- ผลลัพธ์จะได้ การสร้าง matrix
- `[[5.2 3. 4.5]
 [9.1 0.1 0.3]]`

10

การตรวจสอบขนาดของ matrix

- ใช้คำสั่ง `.shape` ตามด้วยชื่อตัวแปรที่ได้เก็บ matrix ในการตรวจสอบขนาด เช่น

```

ex_2d_array = numpy.array([[5.2, 3.0, 4.5], [9.1, 0.1, 0.3]])


```

- `ex_2d_array.shape`
- ผลลัพธ์จะได้
- (2, 3)
- หมายความว่า เป็น matrix 2 แถว 3 หลัก
- `[[5.2 3. 4.5]
 [9.1 0.1 0.3]]`

11

11

12

2

การซักค้าใน numpy array

- ex_2d_array = [[5.2 3. 4.5]
[9.1 0.1 0.3]]

2D array

shape: (2, 3)

12

13

การซักค้าใน list ก่อนที่จะเปลี่ยนเป็น numpy array

- list_x = [[5.2, 3.0, 4.5], [9.1, 0.1, 0.3]]

• ถ้าต้องการซักค้า 0.3 ใน ex_2d_array
• โดยถูกดูจากว่า matrix ค่า 0.3
• จะอยู่ในเมตริกที่ 1 หลักที่ 2 (เริ่มบันจาก 0) ดังนั้น
• ex_2d_array[1,2]
• ผลลัพธ์จะได้
• 0.3

13

14

Operations

- numpy array หรือ matrix สามารถดำเนินการบวก ลบ คูณ หาร กันได้
สร้าง matrix ใหม่อีกอันเพิ่มเป็นตัวแปร ex2_2d_array
- ex2_2d_array = numpy.array([[1,0,0],[0,0,1]])

```
print(ex_2d_array) (matrix แรก)
[[5.2 3. 4.5]
 [9.1 0.1 0.3]]
print(ex2_2d_array)
[[1 0 0]
 [0 0 1]]
```

14

15

ตัวอย่างใช้งานการบวก matrix

- การบวก matrix คือการ加ค่าตำแหน่งเดียวกันมาบวกกัน
- ex_2d_array + ex2_2d_array
- คุณจะได้รับดังนี้

$$\begin{bmatrix} 5.2 & 3. & 4.5 \\ 9.1 & 0.1 & 0.3 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
- ผลลัพธ์จะได้

$$\begin{array}{l} \text{array}([[6.2, 3., 4.5], \\ [9.1, 0.1, 1.3]]) \end{array}$$

15

16

การบวกค่าใน list ก่อนที่จะเปลี่ยนเป็น matrix

```
list_x = [[5.2,3.0,4.5],[9.1,0.1,0.3]]
list_x2 = [[1,0,0],[0,0,1]]
list_x + list_x2

ผลลัพธ์จะได้
[[5.2, 3.0, 4.5], [9.1, 0.1, 0.3], [1, 0, 0], [0, 0, 1]]

จะเห็นว่า list ไม่สามารถบวกเลขเดี่ยวกันแบบบวก matrix ใน numpy array
• งานปกติทั่วไปสามารถใช้ List ได้ แต่งานที่เกี่ยวกับตัวเลขจะใช้ numpy array
```

16

17

ตัวอย่างใช้งานการลบ matrix

```
ex_2d_array - ex2_2d_array
คุณจะได้รับดังนี้
[[5.2 3. 4.5] - [[1 0 0]
 [0 0 1]]]

ผลลัพธ์จะได้
array([[ 4.2, 3., 4.5],
 [ 9.1, 0.1, -0.7]])
```

17

18

ตัวอย่างใช้งานการคูณ matrix

- การคูณ matrix ใน numpy array จะคูณในแบบ array เอ้าตำแหน่งเดียวกันมาคูณกัน
- `ex_2d_array * ex2_2d_array`
- มุมมองคากาอยู่ใน
- $\begin{bmatrix} 5.2 & 3 & 4.5 \\ 9.1 & 0.1 & 0.3 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- ผลลัพธ์จะได้
- `array([[5.2, 0., 0.], [0., 0., 0.3]])`

18

19

matix multiplication (dot product)

- การคูณ matrix ใน numpy array ที่ถูกต้อง จะใช้คำสั่ง `numpy.dot` ตามด้วยตัวแปรที่เก็บค่า matrix ที่ต้องการคูณ(ตัวแรก1, ตัวแรก2)

19

20

ตัวอย่างการคูณ matrix (dot product)

- `numpy.dot(ex_2d_array, ex2_2d_array)`
- มุมมองคากาอยู่ใน
- $\begin{bmatrix} 5.2 & 3 & 4.5 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- ผลลัพธ์จะได้
- `ValueError: shapes (2,3) and (2,3) not aligned: 3 (dim 1) != 2 (dim 0)`

20

21

ตัวอย่างการคูณ matrix (dot product)

- หลังการคูณ matrix ตือ หลักขอตัวหน้าต้องเท่ากับແນວของตัวหลัง ดังนั้นจะต้อง transpose matrix (การกลับหลักเป็นแนวลับและเป็นหลัก)
- สามารถ transpose matrix ที่ต้องการได้โดยใช้คำสั่ง ซึ่งตัวแปรที่เก็บค่า matrix ตามด้วย `.T`
- `print(ex_2d_array)`
- $\begin{bmatrix} 5.2 & 3 & 4.5 \end{bmatrix}$
- `print(ex_2d_array.T)`
- $\begin{bmatrix} 5.2 & 9.1 \\ 3. & 0.1 \\ 4.5 & 0.3 \end{bmatrix}$

21

22

ตัวอย่างการคูณ matrix (dot product)

- ตั้งรื้น การคูณ matrix ใน numpy array ที่ถูกต้อง
- `dot_mat = numpy.dot(ex_2d_array, ex2_2d_array.T)`
- มุมมองคากาอยู่ใน
- $\begin{bmatrix} 5.2 & 3 & 4.5 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$
- `print(dot_mat)`
- ผลลัพธ์จะได้
- $\begin{bmatrix} 5.2 & 4.5 \\ 9.1 & 0.3 \end{bmatrix}$

22

23

การหา det ของ matrix

- ใช้คำสั่ง `numpy.linalg.det` ตามด้วย (ตัวแปรที่ต้องการหา)
- `numpy.linalg.det(dot_mat)`
- มุมมองคากาอยู่ใน
- $\begin{bmatrix} 5.2 & 4.5 \\ 9.1 & 0.3 \end{bmatrix}$
- ผลลัพธ์จะได้
- -39.38999999999986

23

24

matrix slicing

- `print(ex_2d_array)`
- `[[5.2 3. 4.5]`
- `[9.1 0.1 0.3]]`
- สามารถตัดได้ทั้งม้วน list
- `ex_2d_array[1,1:]` หมายความให้อ่านค่าตามอินเด็กส์ที่ 1 หลักที่ 1 ไปจนถึงหลักสุดท้าย
- ผลลัพธ์จะได้
- `array([0.1, 0.3])`



ตัวอย่าง matrix slicing

- `print(ex_2d_array)`
- `[[5.2 3. 4.5]`
- `[9.1 0.1 0.3]]`
- `ex_2d_array[:, :2]` หมายความให้อ่านค่าตามอินเด็กแรกสองหลักแรก ไปจนถึงหลักที่ 1
- ผลลัพธ์จะได้
- `array([[5.2, 3.],`
- `[9.1, 0.1]])`



25

24

Homework class period 4



- เขียน function คูณ matrix ให้ผลลัพธ์เหมือน dot product (ไม่ใช่ dot product)
- และ test กับ matrix ขนาด
- $(2,3) \times (3,2)$
- $(4,4) \times (4,1)$
- $(2,2) \times (2,2)$

27

26

Class period 5

บทที่ 3 โปรแกรมนิเวศและการใช้งานในภาษา Python (ต่อ)

[More_advanced_data_structure 2](#)

1

zeros คำสั่ง numpy.zeros()

`numpy.zeros(shape, dtype='float', order='C', *, like=None)`

Return a new array of given shape and type, filled with zeros.

Parameters:

- `shape` : int or tuple of ints
- `dtype` : data-type, optional
- `order` : {‘C’, ‘F’}, optional, default: ‘C’
- `like` : array-like, optional

This function is used to allow the creation of arrays which are not NumPy arrays; if an array-like passed in as `like` supports the `array_function__` protocol, the result will be defined by it. In this case, it ensures the creation of an array object compatible with that passed in via this argument.

Returns:

`out` : ndarray

Array of zeros with the given shape, `dtype`, and `order`.

1

2

ones คำสั่ง numpy.ones()

`numpy.ones(shape, dtype='float', order='C', *, like=None)`

Return a new array of given shape and type, filled with ones.

Parameters:

- `shape` : int or sequence of ints
- `dtype` : data-type, optional
- `order` : {‘C’, ‘F’}, optional, default: ‘C’
- `like` : array-like, optional

This function is used to allow the creation of arrays which are not NumPy arrays; if an array-like passed in as `like` supports the `array_function__` protocol, the result will be defined by it. In this case, it ensures the creation of an array object compatible with that passed in via this argument.

Returns:

`out` : ndarray

Array of ones with the given shape, `dtype`, and `order`.

3

Matrix Operation (scalar multiplication)

scalar multiplication คือ การคูณค่าคงที่ 1 ค่าที่เป็นเลขเดียวคูณเข้าใน matrix เช่น

- `M_one = np.ones((2, 3))`
- มุมอค์ต้าอยู่ใน

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- `2*M_one`
- ผลพิธีจะได้

$$\begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix}$$

3

4

Random

- คำสั่ง `numpy.random` เป็นคำสั่งที่สร้าง matrix ที่อยู่ค่าภายใน matrix
- คำสั่ง `numpy.random` มีพัฒนาไป โดยคำสั่งที่เบบอยู่มี 3 ประเภท คือ
 - `numpy.random.rand`
 - `numpy.random.rndn`
 - `numpy.random.choice`

5

คำสั่ง numpy.random.rand()

`numpy.random.rand`

• โดย `random.rand` จะเป็น uniform random เป็นการสุ่มค่าที่ทุกค่าไม่ออกมาก่อนถูกให้มากับ จึงควรคำนึงถึงมิติด้านหน้า 0 ถึง 1

`np.random.rand(3, 2)`

`array([[0.12461684, 0.63204405], [0.240901, 0.34341953], [0.22536518, 0.86663463]])`

Parameters:

- `dt0, dt1, ..., dn` : int, optional

Random values in a given shape.

Note

This is a convenience function for users porting code from Matlab, and bears some resemblance to its `rand` function. That function takes a tuple to specify the size of the output, which is consistent with other NumPy functions like `arange`, `sin` and `numpy`.

Returns:

`out` : ndarray, `dtype`, `dt0, ..., dn`

Create an array of the given shape and populate it with random samples from a uniform distribution over (0, 1).

Parameters:

- `dt0, dt1, ..., dn` : int, optional

The dimensions of the returned array must be non-negative. If no argument is given, a single Python float is returned.

Returns:

`out` : ndarray, `dtype`, `dt0, ..., dn`

Random values.

5

6

1

คำสั่ง numpy.random.randn()

• โดย random.randn จะเป็น normal distribution mean=0 std=1 เป็นการสุ่มค่าที่มาทั้งในตัว 0 จะมีโอกาสที่จะสุ่มได้มากกว่าค่าที่อยู่ทาง 0 ซึ่งค่าที่สุ่มได้จะมีค่าตั้งแต่ 0 ถึง 1

```
np.random.randn(3,2)
array([[ 2.0762285,  0.91239845],
       [-2.08011942, -0.46261935],
       [ 0.66804796,  1.19419422]])
```

7

คำสั่ง numpy.random.choice()

• โดย random.choice จะเป็นการสุ่มค่าที่กำหนดเอง

```
np.random.choice([1,2,3,'a','b','c'])
['a']
```

8

ตัวอย่างการใช้งาน Parameter: size numpy.random.choice()

- Parameter: size เป็นการกำหนดจำนวนค่าที่ต้องการสุ่มออกมา โดย size สามารถ input ค่าเป็น int (ตัวเลขเดียว) หรือ tuple of ints (ใช้สำหรับสุ่มค่าอ่อนเป็น matrix) เช่น

```
np.random.choice([1,2,3,28,11,100],size = 2)
array([28,  3])

np.random.choice([1,2,3,28,11,100],size = (2,3))
array([[ 3,   3,  11],
       [28,   1,  11]])
```

9

ตัวอย่างการใช้งาน Parameter: replace numpy.random.choice()

- Parameter: replace การเลือกนี้ เป็นการกำหนดว่า จะให้นำค่าที่สุ่มออกมาได้คืนหรือไม่คืน คือ การสุ่มค่าซ้ำหรือไม่ซ้ำ
- หาก default ของ replace = True คือการสุ่มแบบไม่คืนมีค่าทั้งนี้ได้ ถ้าต้องการสุ่มแบบไม่คืน คือให้มีค่าทั้งนี้ ให้กำหนด replace = False

```
np.random.choice([1,2,3,28,11,100], size = (2,3), replace=False)
array([[ 3,  11,   2],
       [100,  28,   1]])
```

10

ตัวอย่างการใช้งาน Parameter: p numpy.random.choice()

- Parameter: p เป็นการกำหนดความน่าจะเป็นที่ค่าน้อยค่าอยู่กับค่า
- เช่น มีนักศึกษา 33 คนต้องการร่วมต่อให้แล้วแต่ความต้องการน่าจะเป็นที่กำหนด
- ก้ามคนที่ 'A' ให้ความน่าจะเป็นที่จะอยู่กับ 10%
- ก้ามคนที่ 'B' ให้ความน่าจะเป็นที่จะอยู่กับ 20%
- ก้ามคนที่ 'C' ให้ความน่าจะเป็นที่จะอยู่กับ 40%
- ก้ามคนที่ 'D' ให้ความน่าจะเป็นที่จะอยู่กับ 29%
- ก้ามคนที่ 'E' ให้ความน่าจะเป็นที่จะอยู่กับ 1%

```
np.random.choice(['A','B','C','D','F'],size=35,
                 p = [0.1,0.2,0.4,0.29,0.01])
array(['C', 'C', 'C', 'C', 'C', 'B', 'C', 'B', 'C', 'B', 'D', 'D', 'A', 'C', 'C',
       'D', 'A', 'C', 'C', 'D', 'B', 'B', 'C', 'A', 'C', 'C', 'C', 'D',
       'A', 'D', 'C', 'D', 'B', 'D', 'C', 'D', 'C', 'D'], dtype='|<U1')
```

11

เฉลย HW Matrix Multiplication class period 4

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 11 & 12 \end{bmatrix}$$

- เขียน function คูณ matrix ให้ผลลัพธ์เหมือน dot product (ไม่ใช้ dot product)
- เข้า test กับ matrix ขนาด
 - (2,3)*(3,2)
 - (4,4)*(4,1)
 - (2,2)*(2,2)

12

2

การสร้างฟังก์ชัน คุณ matrix

```

• def mat_mul(A,B):
•     C = np.zeros((A.shape[0],B.shape[1]))
•     for r_a in range(A.shape[0]):
•         for c_b in range(B.shape[1]):
•             for every_element in range(A.shape[1]):
•                 C[r_a,c_b] = C[r_a,c_b] + (A[r_a,every_element] *
• B[every_element,c_b])
• 
• return C

```

12

13

ขั้นตอนการสร้างฟังก์ชัน คุณ matrix

```

• def mat_mul(A,B):
•     คือการสร้างและเก็บหน้าที่ของฟังก์ชัน input ที่ต้องการ matrix A * matrix B
• 
• C = np.zeros((A.shape[0],B.shape[1]))
• หมายความว่า ให้สร้าง matrix ที่มีทุกค่าเป็น 0 โดยกำหนดจำนวนแผลและหลัก จากการอ่านคำจำนวนแผล
• ของตัวหน้า(A.shape[0])และจำนวนหลักของตัวหลัง(B.shape[1])จากตัวปรับที่ input เพื่อเตรียมขนาดของ
• matrix ผลลัพธ์

```

13

14

ขั้นตอนการสร้าง loop1 สำหรับ คุณ matrix

```

• for r_a in range(A.shape[0]):
•     A.shape[0] คือจำนวนแถวของตัวหน้า
•     range(A.shape[0]) คือสร้าง list ตัวเลขตามจำนวนแถวของตัวหน้า
•     ด้วย A.shape[0] = 2
•     range(A.shape[0]) = range(2)
•     จะเพิ่งกับ range(0, 2) ในรูปแบบ list คือ[0, 1]
• 
• ผังนั้น loop1 ใช้วนถูป้อนค่าใน range(A.shape[0]) ที่จะตัว เก็บไปในตัวแปร r_a
        range คือคำสั่งที่ใช้สร้าง list ตัวเลข
        shape คือคำสั่งในการตรวจสอบขนาดของ matrix

```

14

15

ขั้นตอนการสร้าง loop2 สำหรับ คุณ matrix

```

• for c_b in range(B.shape[1]):
•     B.shape[1] คือจำนวนหลักของตัวหลัง
•     range(B.shape[1]) คือสร้าง list ตัวเลขตามจำนวนหลักของตัวหลัง
•     ด้วย B.shape[1] = 2
•     range B.shape[1] = range(2)
•     จะเพิ่งกับ range(0, 2) ในรูปแบบ list คือ[0, 1]
• 
• ผังนั้น loop2 ใช้วนถูป้อนค่าใน range(B.shape[1]) ที่จะตัว เก็บไปในตัวแปร c_b

```

15

16

ขั้นตอนการสร้าง loop3 สำหรับ คุณ matrix

```

• for every_element in range(A.shape[1]):
•     A.shape[1] คือจำนวนหลักของตัวหลัง
•     range(A.shape[1]) คือสร้าง list ตัวเลขตามจำนวนหลักของตัวแผล
•     ด้วย A.shape[1] = 3
•     range A.shape[1] = range(3)
•     จะเพิ่งกับ range(0, 3) ในรูปแบบ list คือ[0, 1, 2]
• 
• ผังนั้น loop3 ใช้วนถูป้อนค่าใน range(A.shape[1]) ที่จะตัว เก็บไปในตัวแปร every_element

```

16

17

ลำดับการทำงาน loop for ในฟังก์ชัน

- Loop1 คือ `for r_a in range(A.shape[0]):`
- Loop2 คือ `for c_b in range(B.shape[1]):`
- Loop3 คือ `for every_element in range(A.shape[1]):`
- ด้วย
 - `range(A.shape[0]) = [0, 1]`
 - `range(B.shape[1]) = [0, 1]`
 - `range(A.shape[1]) = [0, 1, 2]`
- Loop1
 - Loop2
 - Loop3
- `print(f'row : {r_a} column : {c_b} every_element : {every_element}')`

17

18

ลำดับการทำงาน loop for ในฟังก์ชัน

ผลลัพธ์จะได้

```

• row : 0 column : 0 every_element : 0
• row : 0 column : 0 every_element : 1
• row : 0 column : 0 every_element : 2
• row : 0 column : 1 every_element : 0
• row : 0 column : 1 every_element : 1
• row : 0 column : 1 every_element : 2

```

หมายความว่า loop3 จะทำงานวนลูปของตัวอิฐจนจบรอบก่อน สิ่งนี้เป็นวนลูป 1 รอบของ loop2 และ loop2 จะทำงานวนลูปของตัวอิฐจนจบรอบก่อน สิ่งนี้เป็นวนลูป 1 รอบของ loop1

19

การคำนวณภายใน loop3

```

• C[r_a,c_b] = C[r_a,c_b] + (A[r_a,every_element] * B[every_element,c_b])

```

- ในการทำงานเร็วๆ ก่อนที่มาของ C[r_a,c_b] จะเท่ากับ 0
- เพรียบเทียบ C = np.zeros((A.shape[0],B.shape[1]))
 • A[r_a,every_element] คือ ค่าของ matrix ตัวหน้าตามตำแหน่ง [r_a,every_element] เช่น
 • ถ้า A คือ


```
array([[1, 2, 3],
                 [4, 5, 6]])
```

 • A[0,0] จะเท่ากับ 1
 • A[0,1] จะเท่ากับ 2

19

การคำนวณภายใน loop3

B[every_element,c_b] คือ ค่าของ matrix ตัวหน้าตามตำแหน่ง [every_element,c_b] เช่น

ถ้า B คือ

```

• array([[7, 8],
       [9, 10],
       [11, 12]])

```

- B[0,0] จะเท่ากับ 7
- B[1,0] จะเท่ากับ 9

20

การคำนวณการคูณ matrix

ตั้งน้ำ

```

• C[r_a,c_b] = C[r_a,c_b] + (A[r_a,every_element] * B[every_element,c_b])

```

- ในการทำงานเร็วๆ ก่อนที่มาของ loop1
 - C[r_a,c_b] จะเป็นตัวแปรที่เข้ามารับค่าผลลัพธ์จากการคำนวณเพื่อนำคืนไปบวกใน loop รอบต่อไป เช่น
 • C[0,0] = C[0,0] + (A[0,0] * B[0,0])
 • C[0,0] = 0.0 + (1 * 7)
 • C[0,0] = 7.0 และเช่นเดียวกันใน loop รอบต่อไป
 • C[0,0] = C[0,0] + (A[0,1] * B[1,0])
 • C[0,0] = 7.0 + (2 * 9)
 • C[0,0] = 25.0 และเช่นเดียวกันใน loop รอบต่อไปจนจบ c[0,0] และไปเริ่ม c[0,1]

21

วิธีใช้งานฟังก์ชัน

สร้าง matrix input ตามตัวอย่าง

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \end{bmatrix}$$

```

mat1 = np.array([[1,2,3],[4,5,6]])
array([[1, 2, 3],
       [4, 5, 6]])
mat2 = np.array([[7,8],[9,10],[11,12]])
array([[ 7,  8],
       [ 9, 10],
       [11, 12]])

```

22

ผลลัพธ์ของฟังก์ชัน

mat_mul(mat1,mat2) ซึ่งฟังก์ชันตามด้วย input(A,B) ที่ตั้งไว้

ผลลัพธ์จะได้

```

• array([[ 58.,  64.],
       [139., 154.]])

```

- ใช้งานรีบเพิ่บตัวแปรในฟังก์ชันคือตัวบปรที่เข้ามารับผลลัพธ์การคำนวณในแมต์ซอร์ของ loop1
 - [[C[0,0], C[0,1]],
 [C[1,0], C[1,1]]]

23



Class period 6

บทที่ 4 การจัดการข้อมูลในรูปแบบตาราง
Pandas 101

1



Pandas

- Pandas เป็นหนึ่งใน package ที่สำคัญของ python ใช้สำหรับจัดการข้อมูลรูปแบบตาราง .CSV
- `import pandas as pd`

1



Download ข้อมูลรายงาน COVID-19 ประจำวัน ข้อมูลประจำประเทศไทย

- <https://data.go.th/dataset/covid-19-daily>

| | |
|---|---|
|  ดาวน์โหลด | รายงานประจำวันติดเชื้อ COVID-19 ประจำวัน  1723 downloads |
|  ดาวน์โหลด | csv = Comma Separated Values ในการจะได้ค่านum จะใช้ comma ในการแยก Meta data = Data that description data ให้อธิบายชุดข้อมูล |

3



การนำข้อมูลเข้า

- สร้าง folder ใน google drive และนำข้อมูล csv ที่ดาวน์โหลดมาไปเก็บใน folder ที่สร้าง
- นำเข้า package pandas และ package ของ google.colab ที่ชื่อ drive เมื่อเข้ามาด้วย google colab

Permit this notebook to access your Google Drive files?
This notebook is requesting access to your Google Drive files. Granting access to Google Drive will permit code executed in the notebook to modify files in your Google Drive. Make sure to review notebook code prior to allowing this access.

No thanks Connect to Google Drive

```
[1] import pandas as pd
from google.colab import drive
drive.mount('/content/drive')
```

3 Connect Google Drive และเลือก Account

4 กด select all และกด continue

```
[2] Mounted at /content/drive
```

4



Import os

- นำเข้า package os เพื่อใช้ในการทำงานต่างๆที่เกี่ยวกับไฟล์ เช่น การรีส์เฟิล์ การลบไฟล์ การสร้างไฟล์เดอร์ เป็นต้น
- โดยในกรณี google drive จะใช้ os เพื่อเข้าไฟล์ ว่าไฟล์ที่ต้องการใช้งานอยู่ path ไหนใน google drive ที่เข้าม
- path คือเส้นทางที่อยู่ไฟล์ จะทำงานเหมือนกับ path ใน window เช่น
- E:\WORKSPACE\Basic Programming\confirmed-cases.csv
- หมายความว่าไฟล์ confirmed-cases.csv อยู่ใน drive E ไฟล์เดอร์ WORKSPACE ในไฟล์เดอร์ Basic Programming

5



การ set path

- path = '/content/drive/My Drive/dataviz_2024_data'
- Set 'path' ที่ต้องบังคับไฟล์เดอร์ที่เก็บไฟล์ .csv ไว้ใน google drive และเป็น string ไว้ในตัวปรับ path
- โดย path หรือเส้นทางที่เข้าไฟล์เดอร์จะต้องต่อๆกันใน os ของ window, mac หรือlinux จะใช้斜線/กับ path แตกต่าง กัน
- Package os จะช่วยให้สามารถเรียก path โดยไม่ต้องจำชื่อลงทุกถูกกันน์ เพราะ os จะนิยมสักถูกกันน์เรื่องไฟล์ตาม platform ที่ใช้งานอยู่ เช่น
- ถ้าใช้ os ของ window ก็จะเรียกคับ \
- ถ้าใช้ os ของ mac หรือ linux จะเรียกคับ /

6

1

คำสั่ง os.path.join()

เป็นคำสั่งที่ใช้สำหรับเชื่อม path เข้าด้วยกัน

```
import os
covid_file_path = os.path.join(path, 'confirmed-cases.csv')
หมายความว่า เชื่อมตัวแปร path ที่ set ไว้ก่อนหน้านี้เข้ากับชื่อไฟล์ 'confirmed-cases.csv' ก็จะได้ไฟล์ covid_file_path
```

```
print(covid_file_path)
ผลลัพธ์จะได้seenท้าไปปังไฟล์ที่ต้องการอยู่ในตัวแปร covid_file_path
/content/drive/My Drive/dataviz_2024_data/confirmed-cases.csv
```

6

7

load data to memory คำสั่ง pd.read_csv()

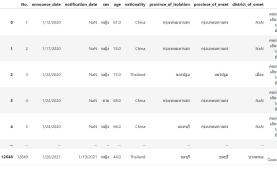
pd.read_csv เป็นคำสั่งที่ใช้สำหรับโหลดข้อมูล

```
data_covid = pd.read_csv(covid_file_path)
```

โหลดข้อมูลไฟล์ confirmed-cases.csv ตามเส้นทาง covid_file_path

```
data_covid
```

ผลลัพธ์จะได้พื้นที่ไฟล์ csv



| No. | onset_date | notification_date | sex | age | industry | province_of_onset | province_of_death | district_of_death |
|------|------------|-------------------|------|------|----------|-------------------|-------------------|-------------------|
| 0 | 2/12/2020 | Not Set | Male | 60.0 | Clean | Residential | Residential | Not Set |
| 1 | 2/12/2020 | Not Set | Male | 70.0 | Clean | Residential | Residential | Not Set |
| 2 | 2/12/2020 | Not Set | Male | 70.0 | Trained | Workplace | Workplace | Not Set |
| 3 | 2/12/2020 | Not Set | Male | 60.0 | Clean | Residential | Residential | Not Set |
| 4 | 2/12/2020 | Not Set | Male | 60.0 | Clean | Residential | Residential | Not Set |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1044 | 1/26/2021 | 5/5/2021 | Male | 40.0 | Trained | Workplace | Workplace | Not Set |

7

8

Parameter: encoding

- ใช้สำหรับกำหนดภาษาของไฟล์ที่จะอ่าน เช่น
- data_covid = pd.read_csv(covid_file_path, encoding='utf-8')
- encoding='utf-8' อาจอ่านภาษาไทยได้แต่จะมี csv ภาษาไทยบางไฟล์ที่มันอ่านไม่ได้แล้วเกิด error
- การอ่านไฟล์ csv ที่มีข้อมูลที่เป็นภาษาไทยที่ครองคุณที่สุดจะใช้
- encoding='iso-8859-1'

8

9

คำสั่ง .head()

- data_covid.shape ซึ่งตัวแปรที่เก็บข้อมูลตามด้วย .head() ใจไปแสดงชื่อคอลัมน์และข้อมูลในตาราง เอกสาร 5 แถวแรก int, default=5
- สามารถกำหนดจำนวนคอลัมน์ที่ต้องการให้แสดงได้ เช่น
- data_covid.head(10) จะแสดงชื่อคอลัมน์และข้อมูลในตาราง 10 แถว

9

10

คำสั่ง .shape

```
data_covid.shape
```

ซึ่งตัวแปรที่เก็บข้อมูลตามด้วย .shape ใช้ตรวจสอบขนาดของข้อมูล ผลลัพธ์จะได้

```
(839771, 11)
```

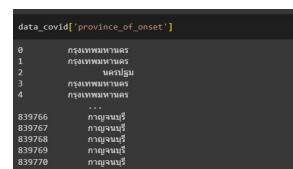
หมายความว่า ข้อมูลทั้งหมด 839,771 แถว มีคอลัมน์ 11 คอลัมน์

10

11

การเข้าในข้อมูลตารางแบบ basic

- ใช้ชื่อคอลัมน์ในการสื้งข้อมูลในคอลัมน์ที่ต้องการ
- data_covid['province_of_onset']



```
data_covid['province_of_onset']
```

| 0 | กรุงเทพมหานคร |
|--------|---------------|
| 1 | กรุงเทพมหานคร |
| 2 | กรุงเทพมหานคร |
| 3 | กรุงเทพมหานคร |
| 4 | กรุงเทพมหานคร |
| ... | ... |
| 839765 | กรุงเทพมหานคร |
| 839767 | กรุงเทพมหานคร |
| 839768 | กรุงเทพมหานคร |
| 839769 | กรุงเทพมหานคร |
| 839770 | กรุงเทพมหานคร |

11

12

2

การซื้อขายในข้อมูลตารางแบบ basic

- การใช้ชี้อีโคล้อตและล้ำดับแล้วในการตีขอ้อมูลในแคนและคอลัมน์ที่ต้องการ
- `data_covid['province_of_onset'][4]`
- ผลลัพธ์จะได้ข้อมูลแนวที่ 4 นับจาก 0 ในคอลัมน์ `province_of_onset`
- 'กรุงเทพมหานคร'

13

การซื้อขายในข้อมูลตารางแบบ .iloc[]

- โดยการมองมุมของข้อมูลตารางในรูปแบบ numpy array หรือ matrix จะใช้ตัวแหน่งเพื่อเข้าขอ้อมูลที่ต้องการ เช่น
- `data_covid.iloc[4, 9]`
- ผลลัพธ์จะได้ข้อมูลแนวที่ 4 คอลัมน์ที่ 9 (เนื่องจาก matrix คือหลักที่ 9) นับจาก 0 คือคอลัมน์ `province_of_onset`
- 'กรุงเทพมหานคร'

14

Table slicing การเลือกข้อมูลเฉพาะคอลัมน์ที่ต้องการ

- การเลือกข้อมูลเฉพาะคอลัมน์ที่ต้องการมาเก็บไว้ในตัวแปรเพื่อนำไปใช้งาน
- `smaller_table = data_covid[['announce_date', 'province_of_onset', 'risk']]`
- หมายความว่า เลือกข้อมูลคอลัมน์ชื่อ `announce_date`, `province_of_onset`, `risk` ในข้อมูลที่เก็บอยู่ในตัวแปร `data_covid` และเก็บข้อมูลเฉพาะคอลัมน์ที่เลือกไว้ในตัวแปร `smaller_table`
- ผลลัพธ์จะได้

15

Table slicing การเลือกข้อมูลเฉพาะแนวที่ต้องการแบบง่าย

- การเลือกข้อมูลเฉพาะแนวที่ต้องการแบบง่าย คือการมองมุมของข้อมูลตารางในรูปแบบ array แต่การนำไปใช้ งาน ใช้บานง่ายไม่ยุ่งยาก
- `data_covid.iloc[1:5, :]`
- หมายความว่า
- : 5 คือเลือกข้อมูลที่อยู่ในแนวที่ 1 ไปจนถึงแนวที่ 4
- , : คือเลือกทุกคอลัมน์ ตั้งนั้น
- `data_covid.iloc[1:5, :]` คือเลือกข้อมูลในตัวแปร `data_covid` ที่อยู่ในแนวที่ 1 ไปจนถึงแนวที่ 4 และ เลือกทุกคอลัมน์

16

Table slicing การเลือกข้อมูลเฉพาะแนวที่ต้องการแบบ advanced

- การเลือกข้อมูลเฉพาะแนวที่ต้องการแบบ advanced คือการใช้ logic query ในการเลือกข้อมูล
- `data_covid[data_covid['province_of_onset'] == 'สมุทรปราการ']`
- หมายความว่า เลือกข้อมูลที่อยู่ในตัวแปร `data_covid` โดยกำหนดค่าคอลัมน์ที่ต้องการคือ `province_of_onset` และต้องการข้อมูลทุกแนวที่ข้อมูลในคอลัมน์ `province_of_onset` เป็นจังหวัดขอนแก่น

17

วิธีการเลือกข้อมูลเฉพาะแนวที่ต้องการแบบ advanced

- การท่านจะสามารถเลือกข้อมูลเฉพาะแนวที่ต้องการแบบ advanced
- คือ การใส่มาตรวัดที่ต้องการ และใส่เป็น list True/False ที่ใช้มาตรวัดที่เก็บจำนวนแนว เพื่อเปรียบเทียบข้อมูลในแนว นั้นๆ ด้วยเงื่อนไข logical expression (True/False) เช่น
- สร้างรายไปใช้รับบกด้วย
- `eight_rows_covid = data_covid.iloc[:8, :]`
- `eight_rows_covid`
- หมายความว่า เลือกข้อมูลในตัวแปร `data_covid` และที่ 0 ถึงแนวที่ 7 ทุกคอลัมน์ที่เก็บไว้ในตัวแปร `eight_rows_covid`

18

การทำงานของการเลือกข้อมูลเฉพาะแผลที่ต้องการแบบ advanced

- ใช้มาตราที่ต้องการตัวยการกำหนดค่า True(มาตราที่ต้องการ)/False(มาตราที่ไม่ต้องการ)
- `eight_rows_covid[[True, True, False, True, True, True, True, False]]`
- ผลลัพธ์จะได้ข้อมูลตารางตามมา True/False ที่เลือกใน list คือมาตราที่ 0, 1, 3, 4, 5, 6

เช่นเดียวกับการสร้าง list ของ logical expression แต่แทนที่จะเลือกเองโดยการใส่ list True/False ให้ กำหนดเงื่อนไขและข้อมูลที่ต้องการ เพื่อเปรียบเทียบและเลือกข้อมูลที่ตรงตามเงื่อนไข โดยถ้าตรงตามเงื่อนไข คือ True ในตรงตามเงื่อนไขคือ False

19

การสร้าง list ของ logical expression

```
eight_rows_covid['province_of_onset'] == 'กรุงเทพมหานคร'
[0, 1, 2, 3, 4, 5, 6, 7]
Name: province_of_onset, dtype: bool
```

19

20

นำ list ของ logical expression ที่สร้างมาใช้งาน

- ซึ่งเมื่อนำมาใช้งานเลือกข้อมูลเฉพาะแผลที่ต้องการแบบ advanced คือการใช้ logic query ในการเลือกข้อมูล นั้นเอง
- `eight_rows_covid[eight_rows_covid['province_of_onset'] == 'กรุงเทพมหานคร']`
- ผลลัพธ์จะได้ข้อมูลทุกมาตราที่มีข้อมูลในคอลัมน์ province_of_onset เป็น กรุงเทพมหานคร คือมาตราที่ 0, 1, 3, 4, 5, 6

21

Bias ในข้อมูล

- bias คือคดิหรือความล้าเอียงที่อาจแฝงอยู่ในข้อมูล ซึ่งอาจเกิดจากวิธีการเก็บข้อมูล การออกแบบแบบสำรวจ หรือความผิดพลาดในการป้อนข้อมูล

21

22

สำรวจ Bias ในชุดข้อมูล

- ใช้ชุดข้อมูลในลักษณะความเป็นจริงที่มี bias หรือข้อจำกัดที่รู้ข้างเป็นตัวอย่างในขั้นเรียน
- ให้นักเรียนนิเคราะห์ชุดข้อมูลเพื่อระบุ bias ที่อาจเกิดขึ้น เช่น การมีตัวแทนของกลุ่มบางกลุ่มหายไปหรือมากเกินไป

23

เทคนิคการประมวลผลข้อมูล

- ครอบคลุมเทคโนโลยีการประมวลผลข้อมูลที่สามารถช่วยลด bias เช่น
- การจัดการกับข้อมูลที่ขาดหายไป
- การจัดการกับค่าผิดปกติ (outliers)
- การสร้างความสมดุลของการกระจายตัวของคลาส

23

24

Algorithmic Bias

- bias สามารถถูกขยายผลหรือเกิดขึ้นซ้ำ ผ่านอัลกอริทึม และไม่ต้องการเรียนรู้ของเครื่อง
- ตัวอย่างของ Algorithmic bias เช่น ระบบการท่านยา อาจพากวนคำศัพท์หรือลักษณะการรับมือรายงานที่เลือกปฏิบัติตามเพศหรือเชื้อชาติ
- นัยยะด้านจริยธรรมของการใช้อัลกอริทึมที่ไม่ดีอย่าง และความสำคัญของความเป็นธรรมและการไม่เลือกปฏิบัติใน การตัดสินใจใช้อัลกอริทึม

24

25

กรณีศึกษาและตัวอย่างจากโลกแห่งความเป็นจริง

- นำเสนอกรณีศึกษาและตัวอย่างจริงของอัลกอริทึมที่ถูกเอียงหรือระบบเก็บข้อมูลที่มีผลกระทบเชิงลบ
- วิเคราะห์กรณีเหล่านี้ในการอภิปรายในชั้นเรียนเพื่อระบุที่มาของ bias นัยยะด้านจริยธรรม และวิธีแก้ปัญหาที่อาบก็ได้
- กระบวนการให้นักเรียนคิดต่อต่อเมื่อพบว่ามีปัญหานี้ แล้วนำมาระบุกติกับแบบฝึกหัดในการเขียนโปรแกรมของตนเอง

<https://www.prolific.com/resources/shocking-ai-bias>

25

26

Homework class period 6

- (ให้เข้าเฉพาะที่อาจารย์สอนไปแล้วในวิชานี้)
- คำนวน สามัญเสี่ยง และสูตร ของอั้มูลังหنمด
- คำนวน สามัญเสี่ยง และสูตร ของอั้มูลังหنمด
- ทำโจทย์ปัญหานี้ที่เป็นคณ “คณต่างๆ คิดเห็นทางมาจากต่างประเทศ”

26

27



Class period 7

บทที่ 5 การเรียนรู้ดูแลรับการทดสอบ 1
Pandas 101 part2

1



เฉลย Homework class period 6

- คำนวณ อายุเฉลี่ย ของผู้หญิง และผู้ชาย ของข้อมูลทั้งหมด
- `this_data = data_covid[['sex','age','province_of_onset']]`
- `this_data`
- เลือกข้อมูลเฉพาะคอลัมน์ที่ต้องการใช้งานและเก็บไว้ในตัวแปร `this`
- `female = this_data[this_data['sex']=='หญิง']`
- เลือกเฉพาะค่าอายุที่ข้อมูลในคอลัมน์ sex ที่เป็น หญิง เก็บไว้ในตัวแปร `female`
- `female['age']`
- เลือกแสดงช่วงอายุในตัวแปร `female` เอกะคอลัมน์ อายุ ก็จะได้ช่วงอายุของเพศหญิงทั้งหมด

1

2



เฉลย Homework class period 6

- กำหนดนิพจน์เพื่อหาอายุเฉลี่ย


```
sum = 0
N = 0
for a in female['age']:
    if a > 0:
        sum += a # sum = sum + a
    N += 1
print(f'อายุเฉลี่ย ของ ผู้หญิง {sum/N}')
```
- กำหนดตัวแปร `sum=0` และ `N=0` เพื่อใช้ในการเก็บค่าจากการบวกในการวนลูปแต่ละรอบจนถึงรอบสุดท้าย โดย `sum` จะใช้เก็บค่าอายุ และ `N` ใช้เก็บจำนวนผู้หญิง

2

3



เฉลย Homework class period 6

- `sum = 0`
- `N = 0`
- กำหนดตัวแปร `sum=0` และ `N=0` เพื่อใช้ในการเก็บค่าจากการบวกในการวนลูปแต่ละรอบจนถึงรอบสุดท้าย โดย `sum` จะใช้เก็บค่าอายุ และ `N` ใช้เก็บจำนวนผู้หญิง
- `for a in female['age']:`
- วนลูปอ่านค่าอายุของผู้หญิงที่เก็บ存ในตัวแปร `a`
- `if a > 0:`
- ถ้าเงื่อนไขนี้ไม่เท่ากับ 0 ถึงจะนำค่าอายุมาบวกค่าวนloopต่อไป หรือหากเกินกว่า 0 ให้ออกจากวนloop
- `print(f'อายุเฉลี่ย ของ ผู้หญิง {sum/N}')`

3

4



เฉลย Homework class period 6

- `sum += a # sum = sum + a`
- `N += 1`
- นำตัวแปร `sum` มาบวกค่าอายุของผู้หญิงที่เล็งคน จนสูง 1 รอบก็จะเอาผลลัพธ์จากการบวกรอบที่แล้วมาบวก ต่อไปเรื่อยๆ เพื่อหาค่าอายุรวม
- นำตัวแปร `N` มาบวก 1 เพื่อใช้บันทึกจำนวนผู้หญิง
- `print(f'อายุเฉลี่ย ของ ผู้หญิง {sum/N}')`
- นำตัวแปร `sum` และ `N` มาหารกันเพื่อหาค่าเฉลี่ย ผลลัพธ์จะได้

4

5



การจัดการ Missing Value

- มีทั้งหมด 3 แบบ
 1. ลบ record ที่เป็น missing
 2. แทนที่ ค่า missing ด้วยค่าที่เหมาะสม mean, default, category-known
 3. ใช้ ค่าจาก columns อื่นๆ ซึ่งประมวลผลค่า ค่าใน column ที่ทางไป (regression, deep learning, etc.)

6

1

ลบ record (dropna)

- missing = None, NA(not authorized), NaN (not a number)
- .dropna() เป็นคำสั่งที่ใช้ในการลบข้อมูลແນວที่ไม่มีค่าหรือไม่มีข้อมูล ตัวอย่างเช่น
- data_covid.shape ผลลัพธ์จะได้ขนาดของข้อมูล data_covid
(839771, 11)
- data_covid.dropna().shape ผลลัพธ์จะได้ขนาดของข้อมูล data_covid ที่ลบແกรข้อมูลที่มีค่าเป็น None
(599988, 11)

7

การใช้งาน .dropna()

- สามารถเลือกลบข้อมูลที่เป็น None เฉพาะในคอลัมน์ที่ต้องการใช้งาน แทนที่จะเลือกลบจากข้อมูลทั้งหมด เช่น
- this_data = data_covid[['sex', 'age', 'province_of_onset']]
this_data.shape ผลลัพธ์จะได้
(839771, 3)
this_data.dropna().shape ผลลัพธ์จะได้
(674906, 3)
- จะเห็นว่าเมื่อเทียบกับ data_covid.dropna().shape ที่เป็นข้อมูลทั้งหมด (599988, 11)
- ข้อมูลที่เลือกเฉพาะคอลัมน์ที่ต้องการใช้งานจะมีจำนวนข้อมูลน้อยกว่า

7

การใช้ตัวแปรเพื่อรับค่า

- this_data.dropna()
print(this_data.shape) ผลลัพธ์จะได้
(839771, 3)
- ซึ่งไม่ใช่ลักษณะที่ได้จากการใช้ .dropna() เพื่อลบข้อมูลແນວที่มีค่าเป็น None เมื่อจากไม่ได้มีตัวแปรเข้ารับค่า เช่น
This_data_dn = this_data.dropna()
print(This_data_dn.shape) ผลลัพธ์จะได้
(674906, 3)

8

Parameter: inplace ของ .dropna()

pandas.DataFrame.dropna

```
Usage: DataFrame.dropna([subset, axis, index, columns, thresh, value, inplace, limit, how, subset, errors, copy, **kwargs], *args, **kwargs)
      or DataFrame.dropna(subset, **kwargs)

Parameters:
    axis : {0 or Index, 1 or Columns}, default 0
        Axis along which to drop rows or columns. When axis=0, drop rows; otherwise drop columns.
    index : Index
        Index values to drop.
    columns : Column labels to drop.
    thresh : int, optional
        Minimum number of non-missing values required along each axis; if fewer than this many, drop the row or column.
    value : scalar, optional
        Value(s) to drop. Can be combined with how.
    subset : list-like, optional
        Subsequence of rows or columns to consider, e.g. if you are dropping rows these would be a list of columns to include.
    inplace : bool, default False
        If True, drop operation performed in-place.
    limit : int, optional
        Maximum number of consecutive null values to consider when creating a new one.
    ignore_index : bool, default False
        If True, the resulting axis will be labeled 0, ..., n - 1.
```

9

Parameter: subset ของ .dropna()

- subset จะเป็นการเลือกเฉพาะคอลัมน์ที่ต้องการลบเฉพาะข้อมูลที่เป็น None เฉพาะคอลัมน์ที่เลือก เช่น
- this_data = data_covid[['sex', 'age', 'province_of_onset']]
this_data.shape ขนาดของข้อมูลตรง
- (839771, 3)
- this_data.dropna().shape ขนาดของข้อมูลตรงที่มีการลบข้อมูลແນວที่เป็น None แบบปกติ
(674906, 3)
- this_data.dropna(subset=['age']).shape
ขนาดของข้อมูลตรงที่มีการลบก้านด้วย subset ลบข้อมูลແນວที่เป็น None เฉพาะ subset ที่กำหนด
(763606, 3)

10

แทน missing ด้วยค่าที่เหมาะสม .fillna()

- .fillna() เป็นคำสั่งที่ใช้ในการแทนที่ค่า missing หรือค่า None ด้วยค่าที่กำหนด เช่น
- this_data = data_covid[['sex', 'age', 'province_of_onset']]
this_data_updated = this_data.fillna(value={'sex': '女', 'age': -1, 'province_of_onset': '不知'})
หมายความว่า ให้แทนที่ข้อมูลที่เป็น None
- ในคอลัมน์ sex แทนด้วย '女'
ในคอลัมน์ age แทนด้วย -1
- และในคอลัมน์ province_of_onset ในแทนที่ค่า None ด้วย '不知'
- ผลลัพธ์จะได้ข้อมูลตรง this_data_updated ที่ไม่มีค่า None

11

11

12

2

แทน missing ด้วยค่าที่เหมาะสม .fillna()

- สามารถตรวจสอบว่าค่า None ถูกแทนที่ด้วยค่าที่กำหนดไว้แล้วหรือไม่
- `this_data_updated[this_data_updated['sex'] == 'null']`
- ผลลัพธ์จะเห็นว่ามีข้อมูลในคอลัมน์ sex ที่ถูกแทนที่ด้วย 'ไม่รู้'

12

การใช้ logical expression จากข้อมูลตารางอื่น

- `data_covid[this_data_updated['province_of_onset'] == '内蒙古']`
- จะเห็นได้ว่าในส่วนที่กำหนดเงื่อนไขของ logical expression มาจากตัวแปร `this_data_updated` ซึ่งนำมามาใช้ในข้อมูลตารางของตัวเอง `data_covid`
- สถานที่ต้นมาระบุไว้ด้วยกันแล้วและผลลัพธ์มาถูกต้อง 2 ตัวแปรที่เป็นข้อมูลตารางคือชื่อจังหวัดมาทำกับมันและในแต่ละตารางมีข้อมูลนี้อยู่บันทึกแน่นหนาด้วยกัน
- ถ้าหาก 2 ตัวแปรมีชื่อเดียวกันแล้วทำกับมันโดยไม่ต้องระบุชื่อช่วงแล้วก็สามารถใช้ร่วมได้ แต่ผลลัพธ์ที่ได้อาจจะไม่ถูกต้อง เพราะไม่ใช่ข้อมูลเดียวกัน

13

การวนลูป record ในตาราง .iterrows()

```

• .iterrows() เป็นคำสั่งที่ช่วยในการวนลูปบันทึกข้อมูลในตาราง
• this_data = data_covid[['sex','age','province_of_onset']]
• for each_row in this_data.iterrows():
•   if (each_row[1]['age'] == 20) and (each_row[1]['province_of_onset']
• == '内蒙古'):
•     print(each_row)
•   #หมายความว่า
•   #ให้วนลูปบันทึกในข้อมูลตารางที่แมปเปิ้นในตัวแปร each_row
•   #ถ้าหัวร้านค้ายืนบันทึกในสูบกานนคือจะเขียนเชิงกากนดใช้เลือก print ข้อมูลเฉพาะแก้วที่มีชื่อสุกี้ในตาราง age=20 และ
•   # province_of_onset='内蒙古'

```

14



Class period 8

บทที่ 5 การเรียกข้อมูลสำหรับการแสดงผล 1 (ตอน)
Pandas 101 part3

1



Quiz สุมชื่อ (5 นาที)

- ให้กูปเข้ารหัสมูลในตารางโดยใช้ .iloc ชี้เป็น numpy array หรือ matrix ให้ผลลัพธ์ออกมาเหมือนนี้ .iterrows()

```

• this_data = data_covid[['sex','age','province_of_onset']]
• for each_row in this_data.iterrows():
•     if (each_row[1]['age'] == 20) and (each_row[1]['province_of_onset']
• == 'เชียงใหม่'):
•         print(each_row)
•

```

1

2



การวนลูปอ่านข้อมูลแบบมองตาราง (pandas dataframe) เป็น numpy array หรือ matrix (.iloc[])

```

• this_data = data_covid[['sex','age','province_of_onset']]

• for each_row in range(this_data.shape[0]):
•     if (this_data.iloc[each_row,1] == 20) and
(this_data.iloc[each_row,2] == 'เชียงใหม่'):
•         print(each_row)
•         print(this_data.iloc[each_row,:])

```

3



การวนลูป

- for each_row in range(this_data.shape[0]):

วนลูปอ่านสักตัวในตารางทีละแถวเก็บไว้ในตัวแปร each_row โดยจำนวนแถวทั้งหมดในตารางสามารถหาได้จาก

- this_data.shape
- (839771, 3)
- this_data.shape[0]
- 839771
- range(this_data.shape[0]) คือ สร้าง list ที่มีเลขตามจำนวนแถวทั้งหมดเพื่อใช้วนลูปเข้าที่จะวน
- [0, 1, 2, 3, ..., 839771]

3

4



การใช้ .iloc[] ในการซื้อข้อมูลในตาราง

- if (this_data.iloc[each_row,1] == 20) and (this_data.iloc[each_row,2] == 'เชียงใหม่'):

สร้างเงื่อนไขสำหรับเลือกเฉพาะข้อมูลที่ต้องการคือ อายุ=20 และ province_of_onset=เชียงใหม่ โดยใช้ .iloc ตามด้วย each_row คือลำดับแถวจากถูกและลำดับคอลัมน์หลักที่ของข้อมูลที่ต้องการ ตัวนี้

- this_data.iloc[each_row,1] == 20 คือ ข้อมูลตัวเดียวในคอลัมน์ อายุ(หลักที่ 1 นับจาก 0) ตรวจสอบว่า เท่ากับ 20 ในตัวแปลง this_data ที่เก็บตารางข้อมูล
- this_data.iloc[each_row,2] == 'เชียงใหม่' คือ ข้อมูลตัวเดียวในคอลัมน์ province_of_onset(หลักที่ 2) ตรวจสอบว่าเท่ากับ 'เชียงใหม่' ในตัวแปลง this_data ที่เก็บตารางข้อมูล

5



print() แสดงข้อมูล

- print(each_row)
- print(this_data.iloc[each_row,:])

เมื่อผ่านเงื่อนไข ให้

- print each_row คือ ตัวเลขลำดับมา และ
- print this_data.iloc[each_row,:] คือ ข้อมูลในแถวที่ n ทุกคอลัมน์

5

6

1

Quiz ในห้อง (15 นาที)

- ตัดตารางอ่อนมาเฉพาะปี 2021 announce_date ในปี 2021
- Hint
- วนลูปหา index ของปี 2021
- ตัดตารางมาเฉพาะ ปี 2021

7

เฉลย

```

• TF=list()
• for each_row in data_covid.iterrows():
•     if each_row[1]['announce_date'].split('/')[2] == '2021':
•         TF.append(True)
•     else:
•         TF.append(False)

• data_covid[TF].head()

```

8

เตรียม list() ว่าง วนลูปและสร้างเงื่อนไข

```

• TF=list()
• สั่ง list ว่างก่อนในตัวแปร TF เพื่อเตรียมรับผลลัพธ์ True False ที่ได้จากการวนลูป

• for each_row in data_covid.iterrows():
• วนลูปนี้มาให้คุณตรวจสอบต่อไปนี้เพิ่มเติม data_covid คือแนวโน้มที่เก็บมาตั้งแต่ปี 2020
• if each_row[1]['announce_date'].split('/')[2] == '2021':

• สร้างเงื่อนไขเพื่อกรอกข้อมูลที่มีมาในคอลัมน์ announce_date เท่ากัน 2021 โดยการใช้ .split('/')[2] เพื่อแยกช่วงความโน้มเหลือง
• เพื่อเข้าใจว่าต้องใช้ในการเรียกอินเด็กซ์
• ข้อมูลนี้ต้องเป็นคอลัมน์ announce_date รูปแบบเป็น 3/11/2021 คือ วัน/เดือน/ปี
• ตั้งนั้นแยกด้วยสัญลักษณ์ / ปัจจุบันต้องมี 2 นับจาก 0

```

9

เขียน True False เข้าไปใน List

```

• if each_row[1]['announce_date'].split('/')[2] == '2021':
•     TF.append(True)
• else:
•     TF.append(False)

• ถ้าผ่านเงื่อนไขให้เขียน True เข้าไปใน list ที่เตรียมไว้ ถ้าไม่ผ่านให้เขียน False
• หมายความว่า ถ้าข้อมูลในคอลัมน์ announce_date เท่ากัน 2021 ให้เขียน True เข้าไปใน list ถ้าไม่ให้เขียน False
• ผลลัพธ์จะได้ list True False ตามจำนวนแนวโน้มของคุณตาราง data_covid ที่เลือก True เฉพาะปี 2021

```

10

ผลลัพธ์

```

• data_covid[TF].head()
• ให้ออกข้อมูลหนึ่ง true false ที่ได้จากการวนลูปเลือกเฉพาะปี 2021
• ผลลัพธ์จะได้ตารางข้อมูลที่มีเพียงข้อมูลในคอลัมน์ announce_date เท่ากัน 2021

data_covid[TF].head()
   No. announce_date notified sex age their_nationality province_of_inisation
124 125 6/6/2020 9/6/2021 wfp 55.0 ♂ Thailand Chonburi Chonburi
6885 6885 1/1/2021 31/12/2020 wfp 40.0 ♂ Thailand Rayong Rayong
6886 6886 1/1/2021 31/12/2020 wfp 21.0 ♂ Thailand Phrae Phrae
6887 6887 1/1/2021 31/12/2020 wfp 20.0 ♂ Thailand Sisaket Sisaket
6888 6888 1/1/2021 31/12/2020 wfp 47.0 ♂ Thailand Ayutthaya Ayutthaya
6889 6889 1/1/2021 31/12/2020 wfp 47.0 ♂ Thailand Chiang Mai Chiang Mai

```

11

Function ตัวช่วยใน pandas

- .describe() คำนวณค่าทางสถิติของข้อมูลที่เป็นตัวเลข
- .mean() คำนวณค่าเฉลี่ยของข้อมูลโดยมีสนใจ missing
- .isnull() ตรวจสอบข้อมูลที่ missing (none)

12

2

.describe()

- ใช้สำหรับคำนวณค่าทางสถิติของข้อมูลที่เป็นตัวเลข
- ไม่คำนึงถึงค่า missing หรือ none มาก่อน
- (ลบข้อมูลเดิมที่มีค่าเป็น none ให้ออกไปแล้ว)

```
data_covid.describe()
   No.      age
count 839771.000000 763605.000000
mean 419886.000000 35.700109
std 242421.150791 16.597799
min 1.000000 0.750000
25% 209943.500000 24.000000
50% 419886.000000 34.000000
75% 629828.500000 46.000000
max 839771.000000 440.000000
```

13

.mean()

- ใช้สำหรับคำนวณค่าเฉลี่ยของข้อมูลโดยไม่สนใจ missing (ลบข้อมูลเดิมที่มีค่าเป็น none ให้ออกไปแล้ว)
- data_covid[data_covid['sex']=='男']['age'].mean()
 - ผลลัพธ์ได้
 - 34.96292702130938

13

.isnull()

- ใช้ตรวจสอบค่า missing ในข้อมูลต่างๆ
- True คือ missing (ค่าว่าง)
- False คือไม่ใช่ค่าว่าง

```
data_covid.isnull()
   No. announce_date Notified_date sex age Unit nationality province_of_isolation
0 False False True False False False False
1 False False True False False False False
2 False False True False False False False
3 False False True False False False False
4 False False True False False False False
...
839766 False False False False False False True False
839767 False False False False False False False False
839768 False False False False False False False False
839769 False False False False False False False False
839770 False False False False False False False False
839771 rows x 11 columns
```

14

Class period 9

บทที่ 5 การเรียนรู้ข้อมูลสำหรับการแสดงผล 2
Pandas 102

1

.isnull().any()

- .any() สามารถใช้สำหรับสรุปค่า True False ในแต่ละ colum ได้อีกความจริงๆในเมื่อจะต้องมีมา OR กัน

| P | Q | P OR Q |
|-------|-------|--------|
| TRUE | TRUE | TRUE |
| TRUE | FALSE | TRUE |
| FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE |

```
data_covid.isnull().any()
```

| No. | announce_date | False |
|-----|-----------------------|-------|
| 1 | Notified date | False |
| 2 | sex | False |
| 3 | age | False |
| 4 | Unit | False |
| 5 | nationality | False |
| 6 | province_of_isolation | False |
| 7 | risk | False |
| 8 | province_of_onset | True |
| 9 | district_of_onset | True |

1

2

.isnull().all()

- .all() สามารถใช้สำหรับสรุปค่า True False ในแต่ละ colum โดยเอาค่าความจริงๆไปในเมื่อจะต้องมีมา and กัน

| P | Q | P AND Q |
|-------|-------|---------|
| TRUE | TRUE | TRUE |
| TRUE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| FALSE | FALSE | FALSE |

```
data_covid.isnull().all()
```

| No. | announce_date | False |
|-----|-----------------------|-------|
| 1 | Notified date | False |
| 2 | sex | False |
| 3 | age | False |
| 4 | Unit | False |
| 5 | nationality | False |
| 6 | province_of_isolation | False |
| 7 | risk | False |
| 8 | province_of_onset | False |
| 9 | district_of_onset | False |

2

3

การเช็คในตารางของ .isnull()

- ไม่สามารถดูว่าจะ base หรือ iloc ปกติ เนื่องจาก
- data_covid['No.'][0].isnull()
- data_covid.iloc[0,0].isnull()
- เขียนเป็นอัลอกิร์ดิฟฟ์เพื่อ iloc numpy array
- ที่มีการตัดตาราง เช่น
- data_covid.iloc[:1,0].isnull()
- เป็นการซื้อไปที่ตารางที่ตัดมาเฉพาะแนวที่ 0 ถึง 1
- และ colum ที่ 0 คือ colum แรก

```
data_covid.iloc[:1,0].isnull()
```

3

4

การต่อตารางแกน X แกน y

- ต่อแกน y คือ เพิ่ม records (เพิ่มจำนวนข้อมูล)
- ต่อแกน x คือ เพิ่ม columns (เพิ่มรายละเอียดของข้อมูล)

4

5

ต่อแกน Y pd.concat()

- การต่อแกน Y เป็นการนำและข้อมูล 2 ตารางรวมหรือมาต่อกัน
- เตรียมตารางที่ต้องการโดยตัดตามเฉพาะช่วงที่ต้องการต่อ

```
dataKK = data_covid[data_covid['province_of_onset']=='กทม']
dataUD = data_covid[data_covid['province_of_onset']=='อุดรธานี']
dataMS = data_covid[data_covid['province_of_onset']=='มหาสารคาม']
# ทำน้ำใจ pd.concat ทางด้วย list ห้ามปรับที่ชื่อเก็บเพื่อต่อตาราง และเก็บตารางที่ต้องกันลงไว้ในตัวแปร dataMYisan ผลลัพธ์จะได้ตารางที่มีเขตข้อมูลของช่วงที่ต้องกัน
```

5

6

1

ตอบแทน X

- การต่อตารางแทน x สามารถทำได้ 2 แบบ
 - จับ 2 ตารางมาต่อกันเลย (merge)
 - เลือกมาเพิ่มเข้ามาบาง column (map)

7

.merge() จับ 2 ตารางมาต่อกันเลย

- การต่อแบบง่าย คือรับส่งตาราง records ตรงกัน สามารถนำตารางมาต่อกันแบบปกติ
- ใช้เมธอด merge ที่จะใช้ต่อกราฟ
- ```
data_province = data_covid[['No.','announce_date','province_of_onset']]
data_human = data_covid[['No.','age','sex','nationality']]
ซึ่งตัวนี้เป็นตัวที่ต้องการจะมาต่อ
full_table1 = data_human.merge(data_province)
full_table1.head()
```
- จะเห็นว่า colum ที่เป็น common คือ announce\_date และ province\_of\_onset ถูกนำมารวบด้วย

7

### .sort\_values()

- ใช้เรียงแมก้าอนุญาตห้ามความคลื่นลุ่มน์ที่ต้องการ โดยใช้ชื่อตัวแปรตารางที่ต้องการตามด้วย .sort\_values(ชื่อคอลัมน์ที่ต้องการ) เช่น
 

```
data_human = data_covid[['No.','age','sex','nationality']]
data_human2 = data_human.sort_values('age')
```
- data\_human2
- ผลลัพธ์จะได้ตารางในลักษณะ pre-data\_human2 ที่แมก้าข้อมูลซึ่งมีค่าเรียงตาม อัจฉริยะ โดย default จะเรียงข้อมูลไปจาก a-z

8

### จำลองการใช้งานจริงของ .merge()

- .merge() จะไม่สามารถต่อตารางที่ไม่ซื้อคอลัมน์ที่เหมือนกันได้ เช่น
 

```
data_human2_renamed = data_human2.rename(columns={'No.': 'patientNumber'})
data_human2_renamed.merge(data_province)
```
- ผลลัพธ์ที่ได้คือ MergeError: No common columns to perform merge on.
- ส่วนใหญ่การใช้งานจริงในการ merge ต้องการจะลบปุ่มแบบนี้
- สิ่งที่ควรทราบก็คือการใช้ parameter: left\_on, right\_on ซึ่งคือคอลัมน์ที่ต้องการให้เป็น index เพื่อ merge ให้ตรงกับชื่อคอลัมน์ที่จะเป็น index ของอีกหนึ่งที่ต้องการจะใช้งานหรือที่เป็น primary key

9

### parameter: left\_on, right\_on ของ .merge()

- ```
full_table3 = data_human2_renamed.merge(data_province, left_on='patientNumber',
right_on='No.')
```
- วิธีเช่น ก่อนหน้า merge ตารางหลักและตารางที่ต้องการต่อแบบปกติ จากนั้นเพิ่ม left_on ก่อนหน้าคือคอลัมน์ที่ต้องการให้เป็น index ของตารางหลัก และเพิ่ม right_on ก่อนหน้าคือคอลัมน์ที่ต้องการให้เป็น index ของตารางที่ต้องการต่อ เก็บตารางที่ต่อแล้วไว้ในตัวแปร full_table3
- full_table3.head() จะเห็นว่า 2 ตารางถูกนำต่อกันโดยใช้ patientNumber ของตารางหลักและ No. ของตารางรองเป็น index

| patientNumber | age | sex | nationality | No. | announce_date | province_of_onset | |
|---------------|--------|------|-------------|----------|---------------|-------------------|------------|
| 0 | 11470 | 0.75 | woman | Burmese | 11470 | 19/5/2021 | suratthani |
| 1 | 114205 | 0.95 | woman | Thailand | 114205 | 19/5/2021 | suratthani |
| 2 | 146574 | 1.00 | man | Thailand | 146574 | 20/5/2021 | suratthani |
| 3 | 523993 | 1.00 | woman | Thailand | 523993 | 20/7/2021 | suratthani |
| 4 | 472541 | 1.00 | man | Thailand | 472541 | 5/8/2021 | suratthani |

10

การสร้างคอลัมน์ใหม่ด้วย pandas

- คุณสมบัติของ pandas ใน การสร้างคอลัมน์ คือ สามารถสร้างคอลัมน์ใหม่ให้ตารางที่ต้องการได้โดย
 - ยกตัวอย่าง df คือตัวแปรตารางที่ต้องการสร้างคอลัมน์ใหม่
 - df['ชื่อ column ใหม่'] = (list ที่มีจำนวนสม圭บกับจำนวน record ของ df) เช่น
- ```
data_human2_renamed.head()
data_human2_renamed['num'] = range(data_human2_renamed.shape[0])
data_human2_renamed
```

11

2

ตัวอย่างการสร้างคอลัมน์ใหม่ด้วย pandas

```
• data_human2_renamed['num'] = range(data_human2_renamed.shape[0])
```

สร้าง list เลขที่ตั้ง record ของข้อมูลตราฟ้าทั้งหมดใน data\_human2\_renamed เช่น shape[0] ตรวจสอบจำนวนและตรวจสอบว่ามี record ไม่ติด range จะได้ไม่สัมภพ ถ้าเก็บน้ำ list สักกี่record ที่ได้ไปสร้างเป็นคอลัมน์ที่ชื่อ num ของพื้นที่ตาราง data\_human2\_renamed

```
• data_human2_renamed ผลลัพธ์ได้
```

13

.map() เลือกมาเพิ่มเฉพาะบาง column

- data\_human2\_renamed['patientNumber'].map(data\_covid.set\_index('No.')['risk'])
- .map จะใช้การกันต์ index ของตารางเดิมและตารางที่ต้องการร่วมกับคอลัมน์ที่ต้องการ
- ในตัวอย่างต่อจาก list ข้อมูลคือชื่อ risk ที่อยู่บันทึกในตาราง data\_covid ให้ไปใช้อันนี้ patientNumber จากตารางเดิมจะ No. จากตาราง data\_covid ที่ต้องการข้อมูลคือชื่อ risk เป็น index และ return list ของค่าในคอลัมน์ risk เพื่อใช้เพิ่มในตารางหลัก

```
data_human2_renamed['patientNumber'].map(data_covid.set_index('No.')['risk'])

111109 ผู้ชายชาวไทย
114048 ผู้ชายชาวไทย
115103 ผู้ชายชาวไทย
522592 ผู้ชายชาวไทย
072509 ผู้ชายชาวไทย
839725 ผู้ชายชาวไทย
839746 ผู้ชายชาวไทย
839747 ผู้ชายชาวไทย
839748 ผู้ชายชาวไทย
839752 ผู้ชายชาวไทย
Name: patientNumber, Length: 83972, dtype: object
```

13

14

การใช้ .map() ต่อตาราง

```
• data_human2_renamed['detail'] = data_human2_renamed['patientNumber'].map(data_covid.set_index('No.')['risk'])
 นำ list ของ risk ที่ได้จากการ map ไปสร้างเป็นคอลัมน์ที่ชื่อ detail ของพื้นที่ตาราง data_human2_renamed
```

```
• data_human2_renamed ผลลัพธ์ได้
```

15

Homework class period 9

- สร้างตารางใหม่ ที่ค่าใน sex เป็น missing ทั้งหมด
- สรุปว่าท่าน record นั้นๆ ที่เป็น missing (ใช้ไม้ได้สอน ให้ลองหารือสรุปด้วยกันเอง)

16

[University Logo]

## Class period 10

บทที่ 5 การเรียนรู้ข้อมูลสำหรับการทดสอบ 2 (ตอน)

Pandas 102 part2

1

[University Logo]

### .groupby()

- <https://www.kaggle.com/crawford/python-groupby-tutorial>
- คือการจัดกลุ่มในคอลัมน์ที่ต้องการ โดยจัดทำข้อมูลที่เหมือนกันในคอลัมน์ที่ต้องการมาไว้ด้วยกัน เช่น
- `data_covid.groupby('nationality')`
- ตัวอย่างที่นักเรียนตรวจสอบ `groupby('ชื่อคอลัมน์ที่ต้องการใช้จัดกลุ่ม')` สามารถได้โดยภาษาของลิ้งค์จัดกลุ่มนี้ที่ต้องการให้มีอยู่ใน list เช่น ([‘ชื่อคอลัมน์1’, ‘ชื่อคอลัมน์2’])
- จากนั้นต้องการจัดกลุ่มตามค่าของลิสต์ที่มีอยู่ในคอลัมน์ `nationality` คำสั่ง `groupby` จะทำการจัดกลุ่มข้อมูลทุกคอลัมน์ โดยจัดกลุ่มตามค่าของลิสต์ที่มีอยู่ในคอลัมน์ `nationality`
- groupby จะทําราบใน memory กรณีลักษณะของการใช้ groupby จะขอใช้คำสั่งเพิ่มเติม สามารถดูได้ทางแบบโดยการเดินค่าสั่งที่ต้องการดูอย่าง เช่น `data_covid.groupby('nationality').count()`

1

[University Logo]

### คำสั่งที่ใช้สำหรับคุณลักษณะ .groupby()

- ยกตัวอย่าง
- `.count()` ใช้สำหรับบัญชีจำนวนสมาชิกในแต่ละคอลัมน์เป็นแต่ละกลุ่มที่เมื่อบอกค่า “ค่าของลิสต์ที่เหมือนกันแต่จะค่าในคอลัมน์ที่ใช้” groupby
- `.mean()`ใช้ค่า mean ในแต่ละคอลัมน์เป็นแต่ละกลุ่ม (ถ้ามีค่าคอลัมน์ที่ไม่มีข้อมูลเป็นค่าว่าง)
- `.max()` ใช้ค่าที่สูงสุดในแต่ละคอลัมน์ (ถ้ามีค่าคอลัมน์ที่ไม่มีข้อมูลเป็นค่าว่าง)

|                                         |                         |                                      |
|-----------------------------------------|-------------------------|--------------------------------------|
| Summary statistics                      | Numpy operations        | More complex operations              |
| <code>mean</code>                       | <code>np.mean</code>    | <code>agg()</code>                   |
| <code>median</code>                     | <code>np.median</code>  | <code>agg(['mean', 'median'])</code> |
| <code>min</code>                        | <code>np.min</code>     | <code>agg(['min', 'max'])</code>     |
| <code>max</code>                        | <code>np.max</code>     | <code>agg(custom_function())</code>  |
| <code>sum</code>                        | <code>np.sum</code>     |                                      |
| <code>describe</code>                   | <code>np.product</code> |                                      |
| <code>count</code> or <code>size</code> |                         |                                      |

2

3

[University Logo]

### เฉลย Homework class period 9 ด้วย groupby()

- สร้างตารางใหม่ ที่ค่าใน sex เป็น missing หั้นนี้
- `data_covid['sex'].isnull()`
- ตรวจสอบค่า wrong (missing) ในคอลัมน์ sex และสร้าง list logical expression True(missing)/False(not missing)
- 
- `missing_sex = data_covid[data_covid['sex'].isnull()]`
- นำ list logical expression มาใช้ก็อปปี้ลงในตารางทุก records ที่มีค่าในคอลัมน์ sex เป็น missing และเก็บตารางที่เลือกแล้วไว้ในตัวแปร `missing_sex`
- `missing_sex` ผลลัพธ์จะได้ตารางที่ทุก records มีค่าในคอลัมน์ sex เป็น missing

3

4

[University Logo]

### เฉลย Homework class period 9 ด้วย groupby()

- สุ่มว่ากันใน record น้ำที่มีเป็น missing ใช้ groupby และ `.describe()` ศูนย์กลางสถิติของข้อมูลเพื่อหาว่าท่าน sex มี missing โดยการตรวจสอบ data หลายบุคคล
- `missing_sex.groupby('nationality').describe()`
- `missing_sex.groupby('province_of_onset').describe()`
- `missing_sex.groupby(['province_of_onset','nationality']).describe()`
- `missing_sex_no_burma = missing_sex[missing_sex['nationality']!='Burma']`
- `missing_sex_no_burma.groupby('risk').describe()`
- `missing_sex.groupby('risk').describe()`

4

5

[University Logo]

### create pandas table

Creating Pandas DataFrames from Python Lists and Dictionaries

**Dictionary**

```
sales = [{"account": "Jones LLC", "Jan": 120, "Feb": 200, "Mar": 140}, {"account": "Alpha Co", "Jan": 200, "Feb": 220, "Mar": 95}, {"account": "Blue Inc", "Jan": 90, "Feb": 80, "Mar": 95}]

df = pd.DataFrame(sales)
```

**List**

```
sales = [{"Account": "Jones LLC", "Jan": 150, "Feb": 200, "Mar": 160}, {"Account": "Alpha Co", "Jan": 200, "Feb": 210, "Mar": 95}, {"Account": "Blue Inc", "Jan": 90, "Feb": 90, "Mar": 95}]

df = pd.DataFrame.from_records(sales, columns=labels)
```

**Row Oriented**

| default | Account   | Jan | Feb | Mar |
|---------|-----------|-----|-----|-----|
| 0       | Jones LLC | 120 | 200 | 140 |
| 1       | Alpha Co  | 200 | 220 | 95  |
| 2       | Blue Inc  | 90  | 80  | 95  |

**Column Oriented**

| Account   | Jan | Feb | Mar |
|-----------|-----|-----|-----|
| Jones LLC | 120 | 200 | 140 |
| Alpha Co  | 200 | 220 | 95  |
| Blue Inc  | 90  | 80  | 95  |

**from\_records**

**from\_dict**

When using a dictionary, column order is not preserved. Explicitly order them.

```
df = pd.DataFrame.from_dict(sales)
```

**from\_items**

sales = [{"account": "Jones LLC", "Alpha Co": 150, "Blue Inc": 200, "Mar": 160}, {"account": "Alpha Co", "Jones LLC": 200, "Blue Inc": 210, "Mar": 95}, {"account": "Blue Inc", "Jones LLC": 90, "Alpha Co": 90, "Mar": 95}]

df = pd.DataFrame.from\_items(sales)

Practical Business Python - pbpython.com

5

6

1

### ตัวอย่างการสร้างตาราง pandas

• แบบ Dictionary ใช้ pd.DataFrame()

- ขั้นตอนการสร้าง สร้าง list ขึ้นมาและเชื่อมต่อกับ record ที่ต้องการในรูปแบบ dictionary โดย index จะเป็นชื่อคอลัมน์และ value จะเป็นค่าของ record นั้นๆ เช่น

```

records = [{'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}]

records_df = pd.DataFrame(records)
records_df

```

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

Dictionary

```

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)

```

7

### ตัวอย่างการสร้างตาราง pandas

• แบบ List ใช้ pd.DataFrame.from\_records()

- ขั้นตอนการสร้าง กำหนดตัวแปร 2 ตัว
- ตัวแรกที่ 1 ให้เก็บ value เป็นค่าของ record นั้นๆ โดยสร้าง list ขึ้นมาและเชื่อมต่อกับ record ที่ต้องการ
- ตัวที่สองที่ 2 ให้เก็บชื่อคอลัมน์ สร้าง list ขึ้นมาและเชื่อมต่อกับลิสต์ที่ต้องการ
- การใช้งาน

```

df = pd.DataFrame.from_records(พารามที่1, columns=พารามที่2)

df

```

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

List

```

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}]

labels = ["account", "Jan", "Feb", "Mar"]

df = pd.DataFrame.from_records(sales, columns=labels)

```

8

### Simple Visualization

- ให้นักศึกษาดาวน์โหลดข้อมูล dataset ชื่อ iris จากลิสต์นี้
- <https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/io/data/csv/iris.csv>
- ดาวน์โหลดข้อมูลจาก link และเก็บข้อมูลไว้ในตัวบ่งชี้
- df = pd.read\_csv(<https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/io/data/csv/iris.csv>)
- df
- ลอง df.groupby('Name').count() ดูพัฒนาของค่าไม่

9

### parallel\_coordinates

• pd.plotting.parallel\_coordinates(df, 'Name');

- การทวน查จะใช้ออกลิ้นเป็นแกน x
- และใช้รากในแกนเดียวกับ record เป็นแกน y
- โดยอัตราของมูลค่าแล้วของในกราฟเป็น 1 จุด
- 1 record คือ 1 เส้น ลากตามค่าของมูลค่าเดี่ยวกัน
- ของ record นั้นๆและลากออกลิ้น
- จัดกลุ่มโดยการแบ่งสีตามค่าใน Name

10

### scatter\_matrix

• pd.plotting.scatter\_matrix(df);

- การบ้าคลั่งที่เป็นตัวเลขของแต่ละ record ในแต่ละคอลัมน์มาเป็นเส้นเทียบกัน
- ตั้งค่า แนว x และ แนว y จะเหมือนกัน คือชื่อคอลัมน์ และค่าในแต่ละคอลัมน์
- จะสังเกตว่าจะเป็นของมูลค่าลิ้นที่ยาวกันเปรียบเทียบกัน จะเห็นเป็นกราฟ histogram
- แต่เป็นข้อมูลแสดงค่าลิ้นมาเปรียบเทียบกัน จะสามารถดูความสัมพันธ์ของมูลค่าอยู่บนค่าลิ้นได้

10

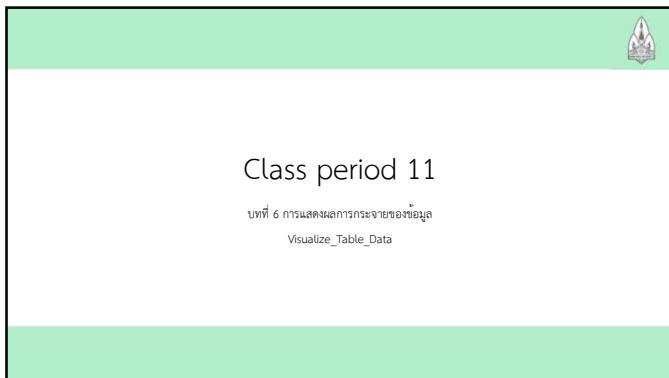
### save table

- ใช้คำสั่ง to\_csv() ในการบันทึกเป็นไฟล์ csv เช่น
- missing\_sex = data\_covid[data\_covid['sex'].isnull()]
- missing\_sex
- ต้องการบันทึกตารางในตัวบ่งชี้ missing\_sex สามารถบันทึกได้โดย
- ใช้ตัวบ่งชี้ของตาราง.to\_csv() หรือไฟล์ที่ต้องการในบันทึก.csv) เช่น
- missing\_sex.to\_csv('missing\_sex.csv')

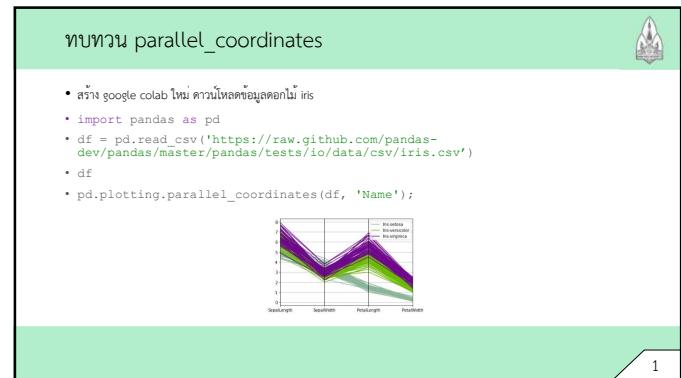
ผลลัพธ์จะอยู่ในไฟล์เอกสารด้านล่างเมื่อลงท้าย google colab  
สามารถดูรายละเอียดได้

11

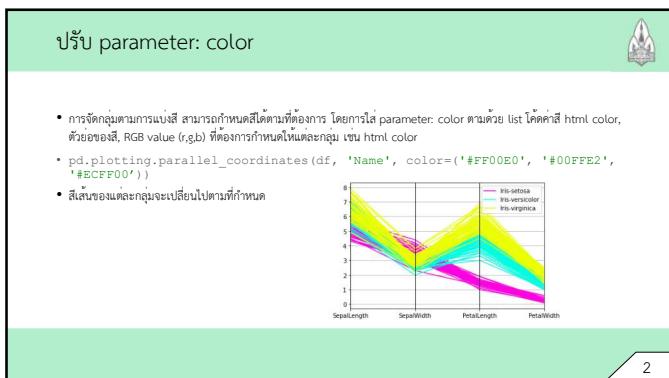
2



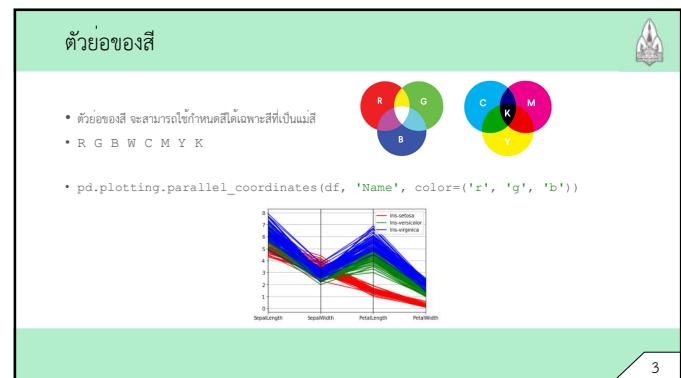
1



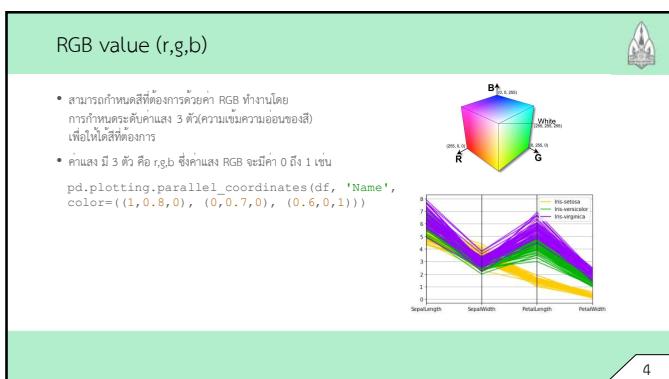
2



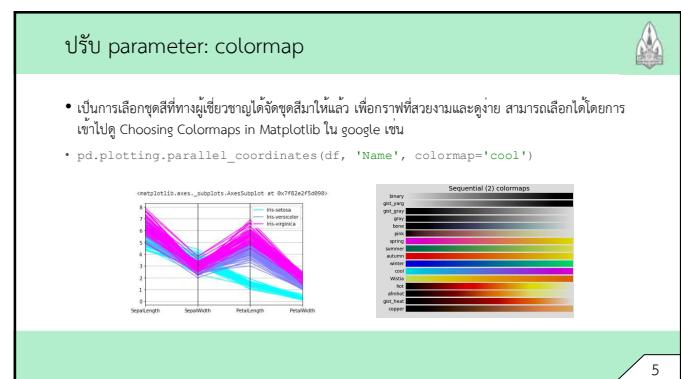
2



3



4



5

5

1

## Matplotlib

เป็น package หรือ library ที่ช่วยในการ visualization ในการวิเคราะห์ข้อมูล

```

import matplotlib
from matplotlib import pyplot as plt

```

matplotlib จะทำงานที่หากข้อมูลที่เป็น numpy array ให้ตัว pandas หรือ matplotlib กับ numpy array ทั้งสองเป็นรูปแบบเดียวกัน ถ้าเป็นรูปแบบ pandas การทำงานของ matplotlib ก็ทำงานได้เพื่อจัดการข้อมูลของ pandas บางประเภท และ matplotlib ทำงานได้กับข้อมูลที่เป็นตัวเลข

ตัวนี้นั้น ควรแปลงข้อมูลจาก dataframe pandas ให้อยู่ในรูปแบบ numpy array ก่อนใช้งาน matplotlib

7

## แปลง data จาก pandas dataframe เป็น numpy array

- ใช้ `to_numpy()` ในการแปลงข้อมูลจาก pandas ที่ต้องการ โดยกำหนด คอลัมน์และแถวที่ต้องการแปลงดังนี้ `iloc` เช่น
- `np_data = df.iloc[:, :-1].to_numpy()`
- ให้แปลงข้อมูลจากในรูปแบบ `df` เสื้อเฉพาะช่วงคอลัมน์ที่เป็นตัวเลข คือตั้งแต่列ที่ 0 ถึงและต่อท้ายและคอลัมน์ที่ 0 ถึงคอลัมน์รองสุดท้าย(คอลัมน์ที่ 3)
- `np_data` จะเห็นข้อมูลที่เป็นตัวเลขของข้อมูลการเรียงตัวใน `df` ถูกแปลงเป็น numpy array

7

## Visualize array data ด้วย pixel

เป็นการแทนค่าตัวเลขใน numpy array ด้วยจุดสี pixel

ค่ามากสีจะรุ่ง ค่าน้อยสีจะเข้ม

สามารถใช้ความแตกต่างของค่าสี

ยกตัวอย่างข้อมูล 11 แล้วมองหา `np_data`

8

## วิธีการใช้งาน Visualize array data ด้วย pixel

- ขั้นตอนที่ 1 ทำงานใน memory ใช้กำหนดขนาดของรูปภาพ กว้าง\*ยาว ตัวอย่าง กว้าง 15 ยาว 15 `matplotlib.rcParams['figure.figsize']=[15,15]`
- ขั้นตอนที่ 2 ใช้คำสั่ง `plt.imshow` เพื่อแสดงรูปภาพ
- `plt.imshow(np_data[:, :], interpolation='nearest')`
- input เป็นตัวเปลี่ยนเป็นรูปแบบ numpy array สามารถเลือกและเลือกที่ต้องการให้แสดงรูปภาพได้
- parameter: `interpolation` ใช้สำหรับเลือกคุณสมบัติของการใช้ในรูปภาพ
- 'nearest' และ pixel ลีบเป็นสีเหลืองและขาวให้การแยกสีในรูปภาพชัด

9

## การใช้ `plt.subplot()`

- ใช้ในการแสดงรูปภาพฟีลมากกว่า 1 รูปเพื่อเรียงต่อกันตามแนวแกนตั้งและแนวนอน
- โดยการใช้งาน `plt.subplot(a,b,c)` ตามด้วย `plt.imshow`(ดูประทิษฐ์เก็บข้อมูล, parameter ของภาพที่ต้องการสร้าง) ในบรรทัดต่อไป
- a คือ parameter ที่ใช้กำหนดแนว
- b คือ parameter ที่ใช้กำหนดคอลัมน์
- c คือ parameter ที่ใช้กำหนดค่าบันทึกภาพที่ต้องการแสดง เช่น
- `plt.subplot(1,3,1)` `plt.subplot(2,3,5)`

10

## ตัวอย่างการใช้ `plt.subplot()`

```

plt.subplot(1,3,1)
plt.imshow(np_data[:10,:], interpolation='nearest',
cmap='magma')

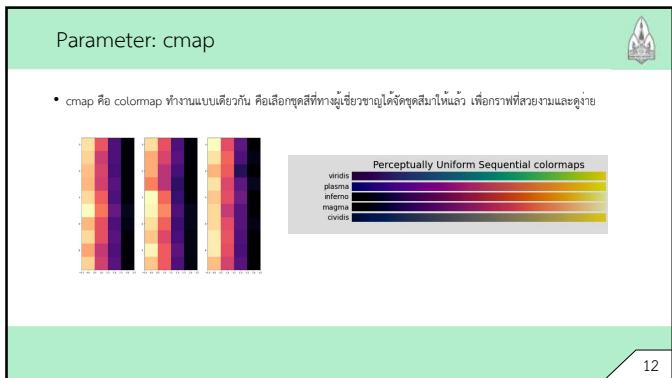
plt.subplot(1,3,2)
plt.imshow(np_data[10:20,:], interpolation='nearest',
cmap='magma') ->

plt.subplot(1,3,3)
plt.imshow(np_data[20:30,:], interpolation='nearest',
cmap='magma') ->

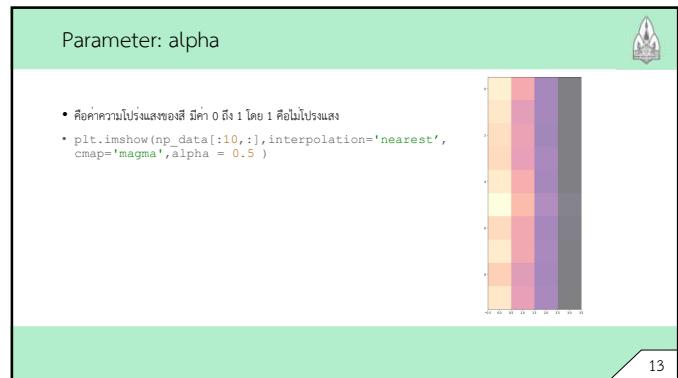
```

11

2



13



14

Class period 12

บทที่ 6 การแสดงผลการกระจายของข้อมูล (ต่อ)

Visualize\_Data\_Distribution

1

Quiz2

- โหลดตัวอย่างรายได้ด้วยตัวเดือนต่อครัวเรือน จำนวนตามมาก และจังหวัดจาก [https://data.go.th/dataset/item\\_7049410f-5bb8-4c75-9e94-112ca18b63e2](https://data.go.th/dataset/item_7049410f-5bb8-4c75-9e94-112ca18b63e2)
- ให้นักศึกษา clean ข้อมูลอีกสิบ 'ภาคและจังหวัด' และคอลัมน์ที่เป็น missing ออก

2

การแสดงผลการกระจายของข้อมูล Scatter Plot

```

import pandas as pd
from matplotlib import pyplot as plt
df = pd.read_csv('https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/io/data/csv/iris.csv')
df.head()

สร้าง google colab ใน Visualize_Data_Distribution_(02)
ดาวน์โหลดชุดข้อมูล iris

```

3

ดอกไม้ iris

*iris setosa*      *iris versicolor*      *iris virginica*

4

Scatter 2D

- ศึกษาระบบการพยากรณ์ในตารางแต่ละค่าเป็นจุด 1 จุด และนำเข้ามาสร้างกราฟ เพื่อถูกการกระจายของข้อมูลระหว่าง 2 nomination (คอลัมน์)
- โดยใช้คำสั่ง plt.scatter(แบบX,แบบY) เช่น

```
plt.scatter(df['PetalLength'], df['SepalLength'])
```

| SepalLength | SepalWidth | PetalLength | PetalWidth | Name            |
|-------------|------------|-------------|------------|-----------------|
| 1           | 3.5        | 1.4         | 0.2        | Iris-setosa     |
| 1           | 3.0        | 1.4         | 0.2        | Iris-setosa     |
| 2           | 3.2        | 1.3         | 0.2        | Iris-setosa     |
| 3           | 1.8        | 5.1         | 1.8        | Iris-versicolor |
| 3           | 1.7        | 4.9         | 1.4        | Iris-versicolor |
| 4           | 4.0        | 4.7         | 1.8        | Iris-versicolor |
| 4           | 3.4        | 4.4         | 1.3        | Iris-versicolor |
| 5           | 4.5        | 4.9         | 1.5        | Iris-versicolor |
| 5           | 4.2        | 5.0         | 1.5        | Iris-versicolor |
| 6           | 5.0        | 5.4         | 2.3        | Iris-virginica  |
| 6           | 4.6        | 5.0         | 1.5        | Iris-virginica  |
| 7           | 5.4        | 5.7         | 2.0        | Iris-virginica  |

5

Parameter: c กำหนดค่าสีทุกจุด

- สามารถกำหนดค่าสีต่อการให้หกจุดได้ แต่หกจุดจะเสียเวลา กับ เช่น
- plt.scatter(df['PetalLength'], df['SepalLength'], c='r')

6

## Parameter: c กำหนดค่าสีตามกลุ่มข้อมูลที่ต้องการ

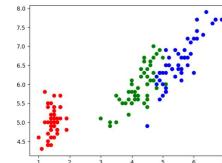
- สามารถกำหนดค่าสีและจุดตามชื่อค่าไม้เล็กๆนั้นได้ โดยการแทนที่ชื่อยูนิตชื่อค่ากับชื่อสี หรือตัวอย่างจะเป็น เช่น
- ใช้คำสั่ง `set(df['Name'])` เพื่อคู่อีคอกไม้ในคอลัมน์ Name ผลลัพธ์จะได้  
('Iris-setosa', 'Iris-versicolor', 'Iris-virginica')
- ใช้คำสั่ง `replace` เพื่อแทนค่าที่ต้องการด้วยตัวเลขหรือตัวอักษร และเก็บการเปลี่ยนที่อยู่ไว้ในตัวแปรใหม่ เช่น  
`df2 = df.replace({'Iris-setosa': 0, 'Iris-versicolor': 1, 'Iris-virginica':2})`
- หรือ
- `df2 = df.replace({'Iris-setosa': 'r', 'Iris-versicolor': 'g', 'Iris-virginica': 'b'})`



6

## Parameter: c กำหนดค่าสีตามกลุ่มข้อมูลที่ต้องการ

- ให้กำหนด Parameter c ด้วยค่าในคอลัมน์ Name ของเพิ่มปริมาณที่เก็บไว้
- `plt.scatter(df2['PetalLength'],df2['SepalLength'],c=df2['Name'])`



7

8

## alpha กำหนดความโปร่งแสง

- ค่าความโปร่งแสงจะมีค่า 0 ถึง 1 โดย 1 คือไม่โปร่งแสง เท่ากับเมืองทึบๆทึบๆ (Bubble Chart)
- `plt.scatter(df2['PetalLength'],df2['SepalLength'],c=df2['Name'],alpha=0.5)`



8

9

Class period 13

บทที่ 6 การแสดงผลการกระจายของข้อมูล (ต่อ)  
Visualize\_Data\_Distribution part2

1

quiz3 (5 นาที)

- โพสต์คราวนง ตค57 ทับ ตาราง พม57
- จากไฟล์ ชั้นเรือ ตค57-ตค58
- จาก [https://data.go.th/dataset/item\\_fadc318e-7743-4bc2-84d1-a19210a9bbd6](https://data.go.th/dataset/item_fadc318e-7743-4bc2-84d1-a19210a9bbd6)

2

การเพิ่มชื่อกราฟ

- สามารถตั้งชื่อกราฟที่ต้องการได้โดยการใช้คำสั่ง `plt.title()` เช่น
- `plt.scatter(df2['PetalLength'],df2['SepalLength'],c=df2['Name'])`
- `plt.title('Comparing petal & sepal length of different iris flowers');`

2

การเพิ่มชื่อแกนในกราฟ

- สามารถตั้งชื่อแกนในกราฟตามที่ต้องการได้โดยการใช้คำสั่ง `plt.xlabel()` และ `plt.ylabel()` เช่น
- `plt.scatter(df2['PetalLength'],df2['SepalLength'],c=df2['Name'])`
- `plt.xlabel('PetalLength')`
- `plt.ylabel('SepalLength')`

3

4

การแยกข้อมูลเพื่อกำหนดลักษณะ marker แต่ละกลุ่ม

- โดยการแยกข้อมูลเพื่อกำหนดลักษณะ marker แพลตฟอร์ม ชื่นเป็นที่จะต้องรู้ว่าคันและจำนวน record ของแต่กลุ่มก่อน เช่น
- ในข้อมูลเดิม Iris มีอยู่ 3 สายพันธุ์ คือ
- Iris-setosa, Iris-versicolor และ Iris-virginica
- เมื่อชี้ช่องลักษณะและจำนวน record แพลตฟอร์มสามารถ record ที่ต้องการแยกได้
- Iris-setosa คือ [ :50 ] ข้อมูล record แรกถึง record ที่ 49
- Iris-versicolor คือ [50:100] ข้อมูล record ที่ 50 ถึง record ที่ 99
- Iris-virginica คือ [100: ] ข้อมูล record ที่ 100 ถึง record สุดท้าย

4

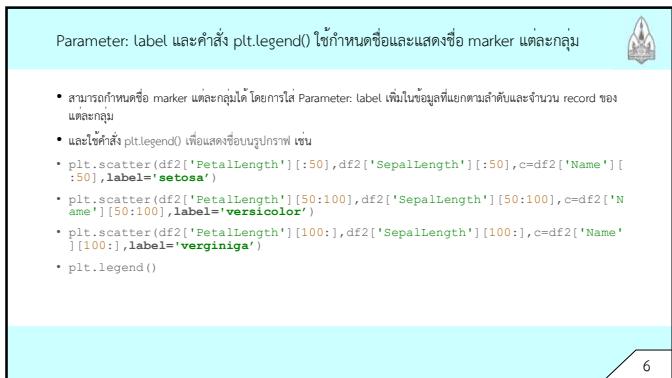
ตัวอย่างใช้การแยกข้อมูลเพื่อ plot กราฟซ้อนกัน

- `plt.scatter(df2['PetalLength'][:50],df2['SepalLength'][:50],  
c=df2['Name'][:50])`
- `plt.scatter(df2['PetalLength'][50:100],df2['SepalLength'][50:100],  
c=df2['Name'][50:100])`
- `plt.scatter(df2['PetalLength'][100:],df2['SepalLength'][100:],  
c=df2['Name'][100:])`

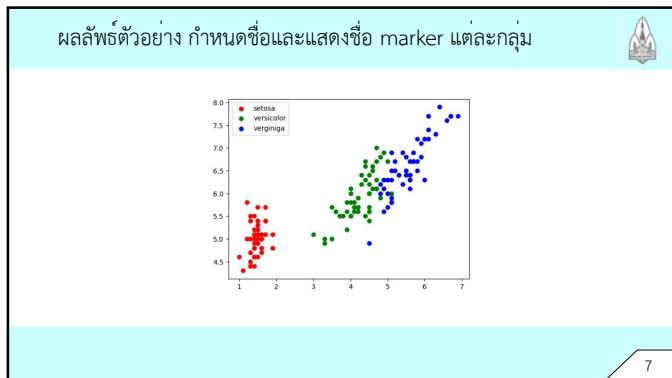
5

6

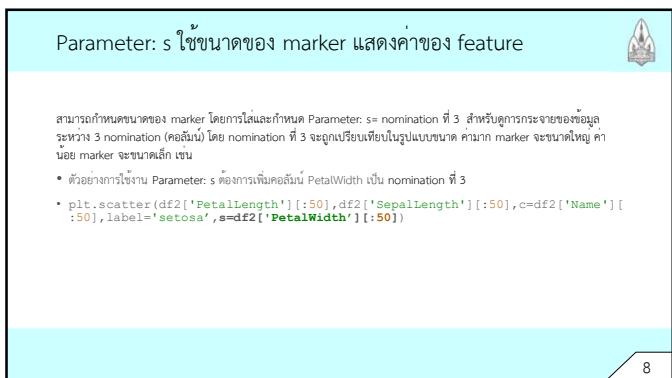
1



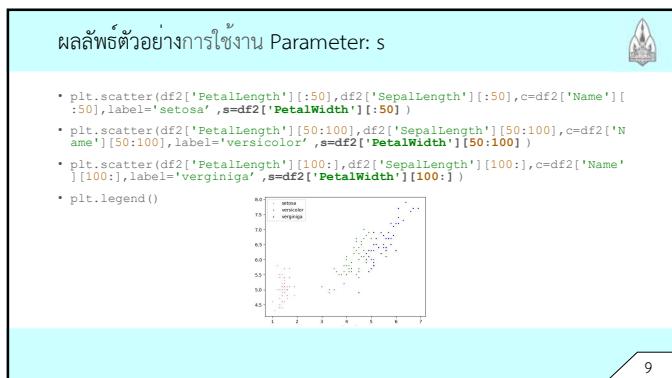
7



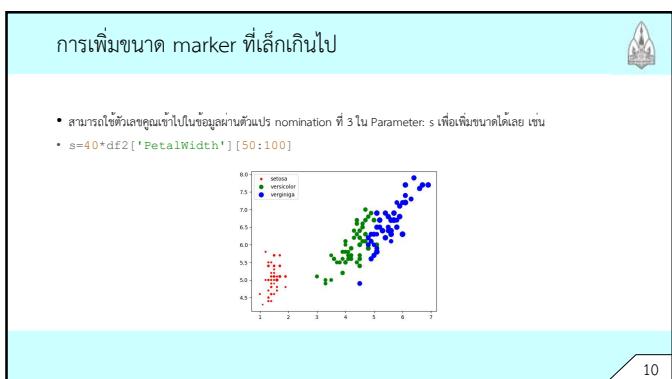
8



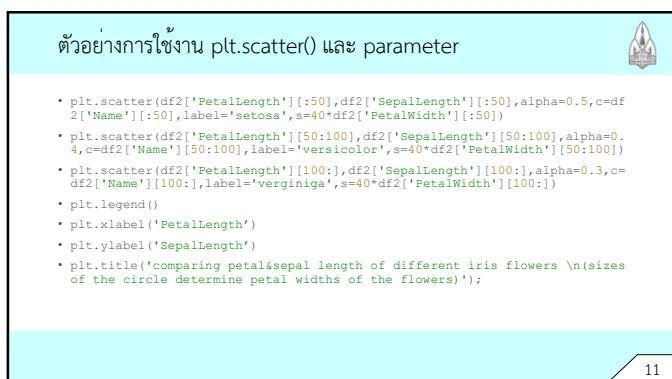
9



10

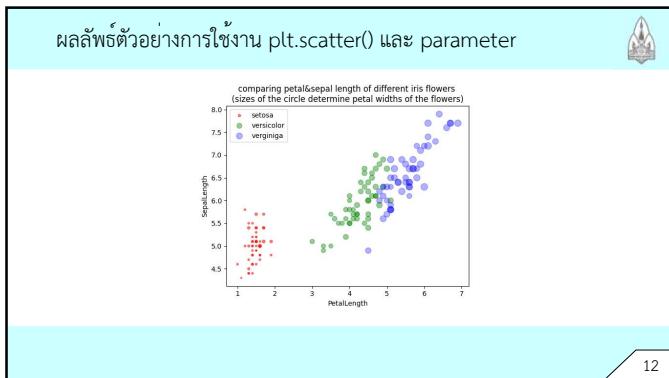


11

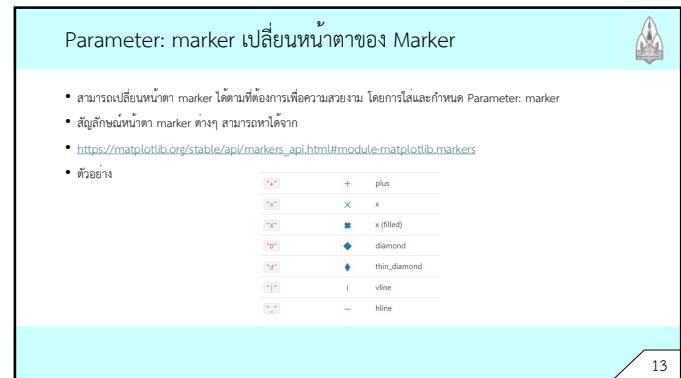


12

2



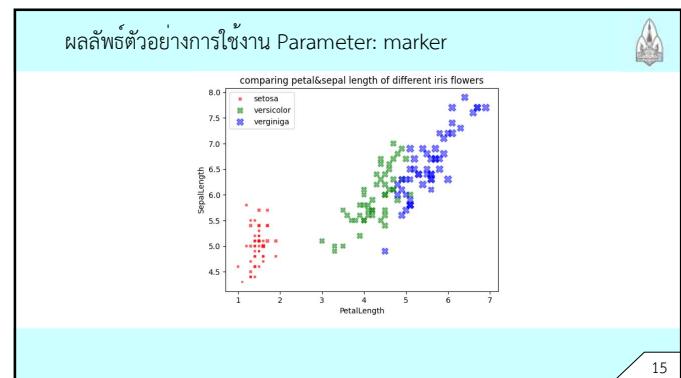
13



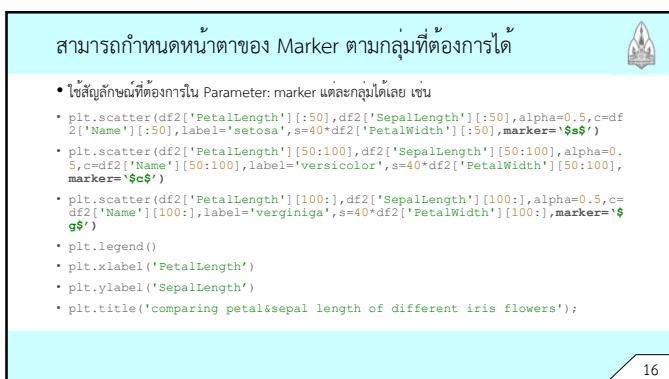
14



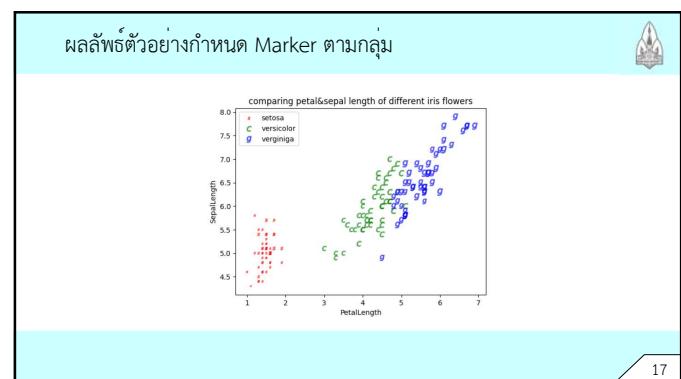
15



16



17



18

## Class period 14

บทที่ 6 การแสดงผลการกระจายของข้อมูล (ต่อ)  
Visualize\_Data\_Distribution part3

1

### Quiz

- ให้บุคคลที่สนใจศึกษาข้อมูลเพื่อการเดินทางบนเครือข่ายถนน ปี 2563 - 2567 จาก <https://data.gov.mot.go.th/dataset/9b9970e9-edd8-4518-99ae-2b81115068c2/resource/71a552d0-0fea-4e05-b78c-42d58aa88db6/download/passengers.xlsx>
- และโหลดลง google colab
- ลองดึงเฉพาะข้อมูล ทางภาคด้านสุวรรณภูมิ ข้าวสารประเวศและสร้างตารางกับใบงานที่มาปรับใหม่

1

### plt.plot()

- พัฒนาชั้น plt สามารถวาดรูป scatter ได้เมื่อebin และการใช้งานจะอธิบายถูกกว่า สามารถกำหนดให้ลักษณะข้อมูล scatter แต่ละจุดได้
- plt.plot() input ตัวที่ 3 คือ ตัวกำหนดหน้าตาของ marker เช่น
- `plt.plot(df['PetalLength'][:50],df['SepalLength'][:50],'o-r')`
- input ตัวที่ 3 'o-r' นำพา marker/ลักษณะของลักษณะข้อมูล/สี
- o คือ กำหนดหน้าตา marker เป็นรูปสี่เหลี่ยมผืน白沙
- - คือ กำหนดลักษณะของเส้นเชื่อมจุดเป็นเส้นทึบ
- r คือ ตัวอย่างของสีแดง

2

### ตัวอย่างการใช้งาน plt.plot()

```
plt.plot(df['PetalLength'][:50],df['SepalLength'][:50],'o-r',alpha=0.5)
```

A scatter plot with two axes. The x-axis is labeled 'PetalLength' and ranges from 1.0 to 1.8. The y-axis is labeled 'SepalLength' and ranges from 4.4 to 5.6. The plot contains 50 data points represented by semi-transparent red circles ('o') connected by thin red lines ('r'). The alpha value is set to 0.5, making the points semi-transparent.

3

### ตัวอย่างการใช้งาน plt.plot()

```
plt.plot(df['PetalLength'][:50],df['SepalLength'][:50],'+:b',alpha=0.5)
```

A scatter plot with two axes. The x-axis is labeled 'PetalLength' and ranges from 1.0 to 1.8. The y-axis is labeled 'SepalLength' and ranges from 4.4 to 5.6. The plot contains 50 data points represented by blue plus signs ('+') connected by dashed blue lines (':'). The alpha value is set to 0.5, making the points semi-transparent.

4

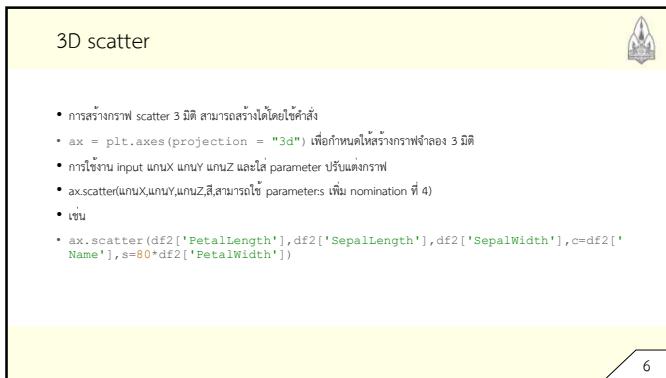
### ตัวอย่างการใช้งาน plt.plot() แบบ plot กราฟซ้อนกัน

```
plt.plot(df['PetalLength'][:50],df['SepalLength'][:50],'o--r',alpha=0.5)
plt.plot(df['PetalLength'][50:100],df['SepalLength'][50:100],'x-c',alpha=0.5)
plt.plot(df['PetalLength'][100:],df['SepalLength'][100:],'*:m',alpha=0.5)
```

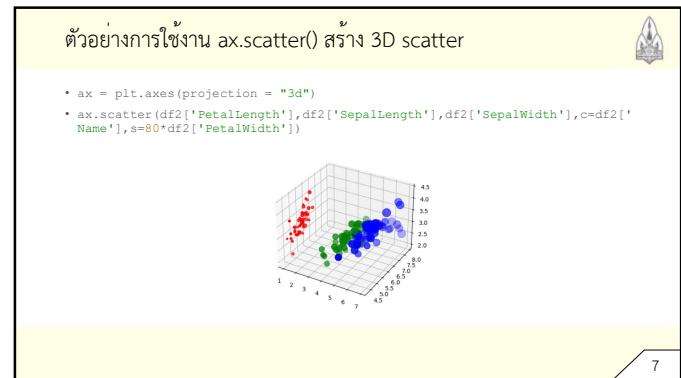
Three stacked scatter plots showing Petal Length vs Sepal Length. The top plot shows the first 50 data points ('o--r', alpha=0.5). The middle plot shows the next 50 data points ('x-c', alpha=0.5). The bottom plot shows the last 100 data points ('\*:m', alpha=0.5). Each segment uses a different color and line style to distinguish between them.

5

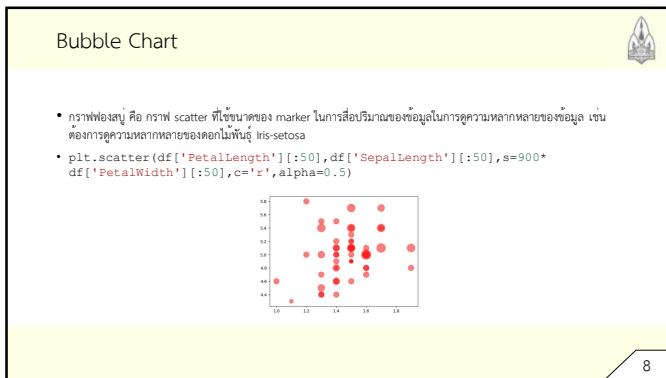
1



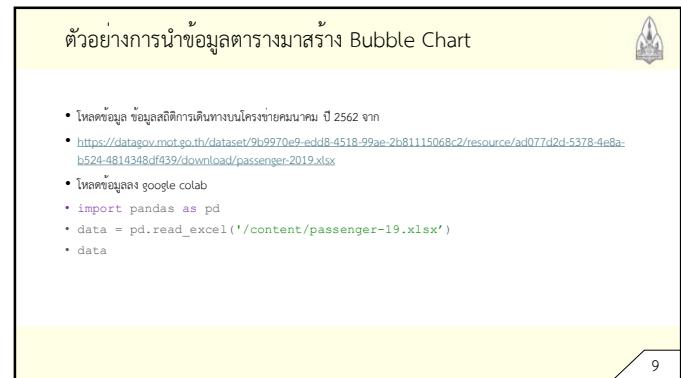
7



8



9

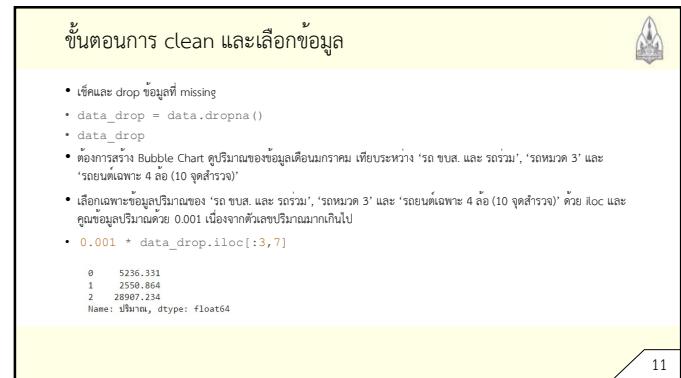


10

### ตาราง

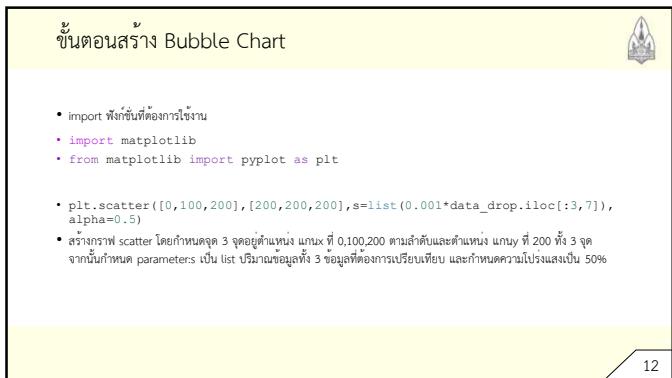
| ผู้โดยสาร | เพศ | ประเทศเดิม | ประเทศท่องเที่ยว | วันเดินทาง | สถานะเดินทาง | จำนวนเดินทาง | จำนวนเงิน |
|-----------|-----|------------|------------------|------------|--------------|--------------|-----------|
| 0         | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-01 | เดินทางไป    | 1            | 522001    |
| 1         | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-01 | เดินทางกลับ  | 1            | 2550004   |
| 2         | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-02 | เดินทางไป    | 1            | 2990204   |
| 3         | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-03 | เดินทางกลับ  | 1            | 16400001  |
| 4         | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-04 | เดินทางกลับ  | 1            | 1050716   |
| ...       | ... | ...        | ...              | ...        | ...          | ...          | ...       |
| 367       | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-05 | เดินทางกลับ  | 1            | 748001    |
| 368       | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-06 | เดินทางกลับ  | 1            | 702000    |
| 369       | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-07 | เดินทางกลับ  | 1            | 885201    |
| 370       | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-08 | เดินทางกลับ  | 1            | 720007    |
| 371       | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-09 | เดินทางกลับ  | 1            | 645004    |
| 372       | ชาย | ประเทศไทย  | ญี่ปุ่น          | 2020-01-10 | เดินทางกลับ  | 1            | 5236.331  |

11

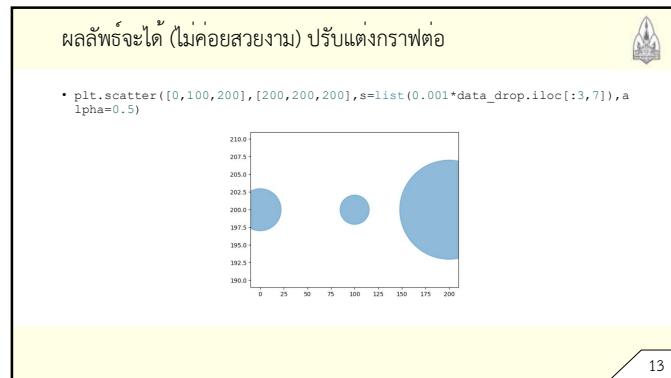


12

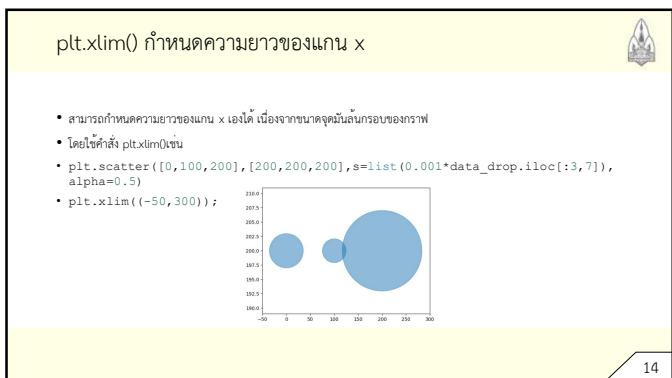
2



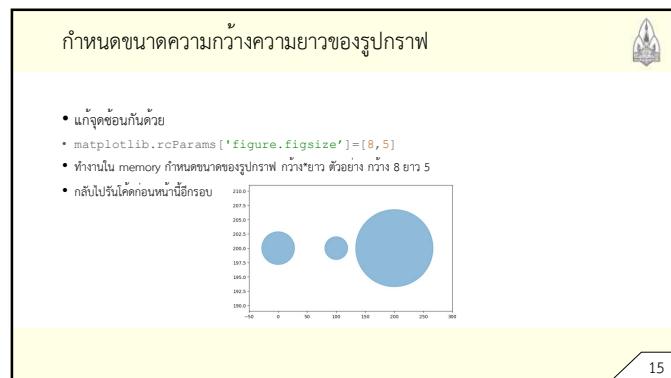
12



13



14

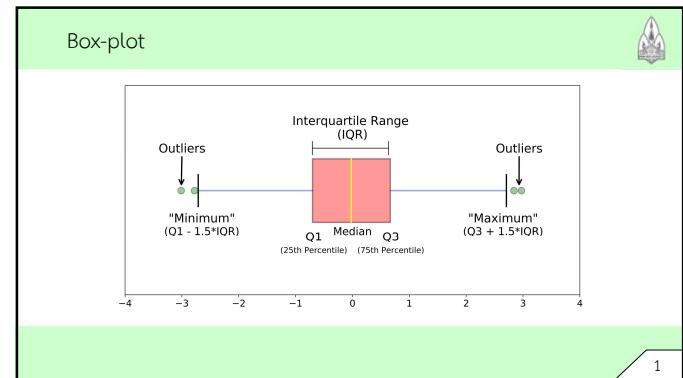


15

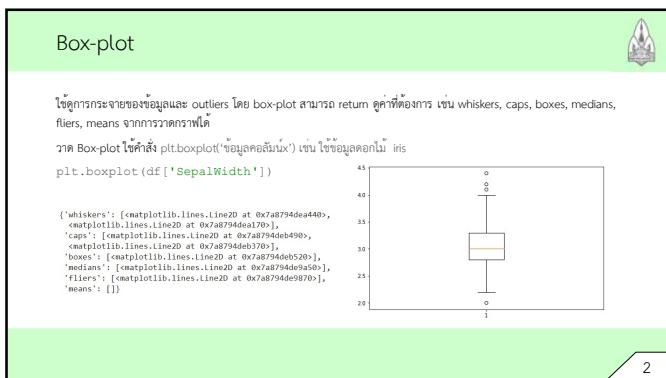
## Class period 15

บทที่ 6 การแสดงผลการกระจายของข้อมูล (ต่อ)  
Visualize\_Data\_Distribution part4

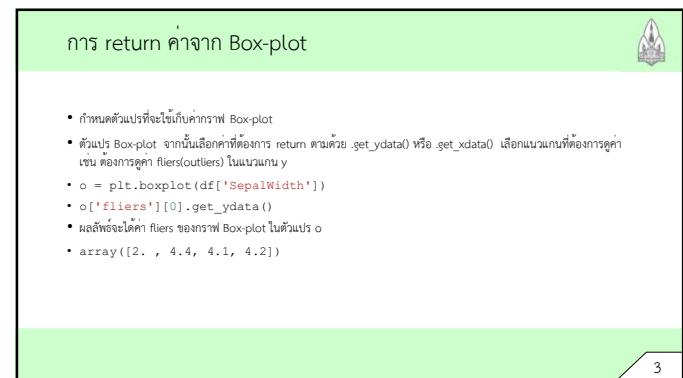
1



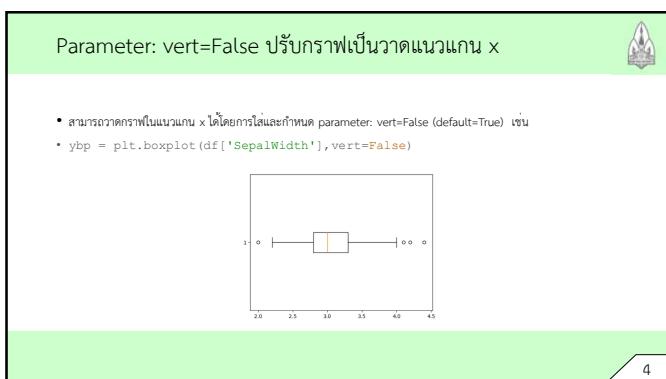
2



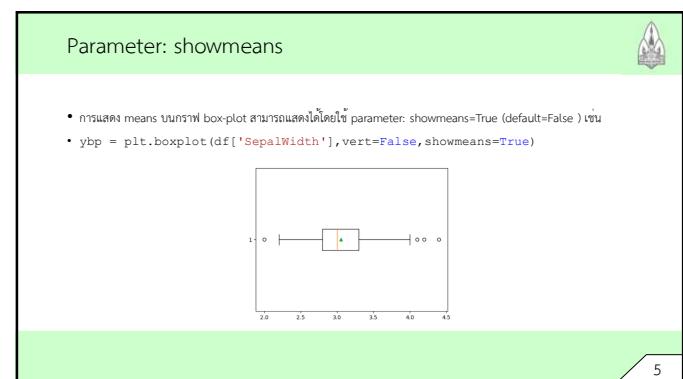
3



4

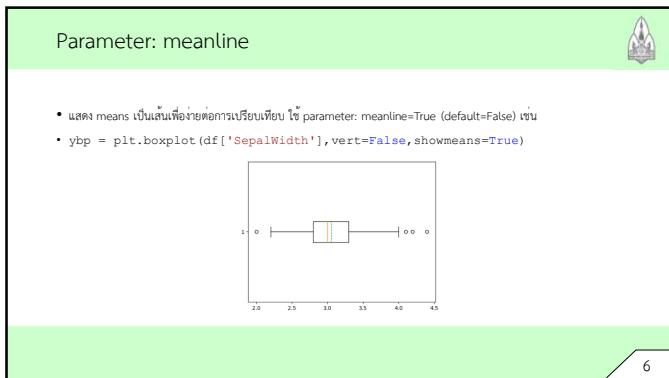


5



6

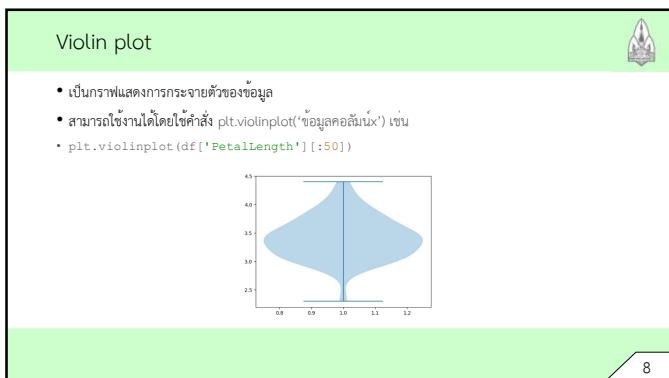
1



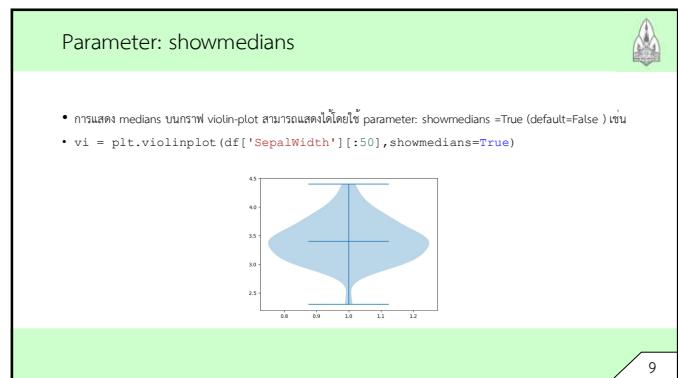
7



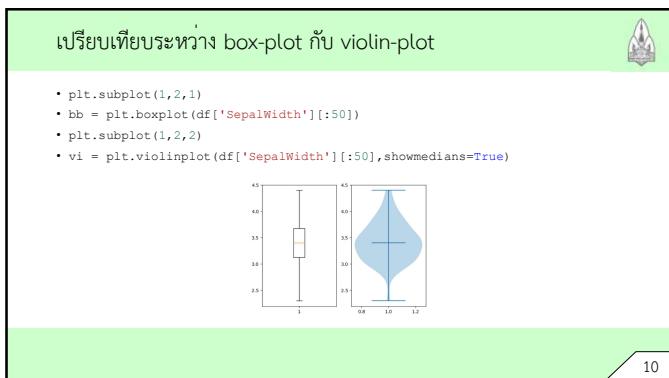
8



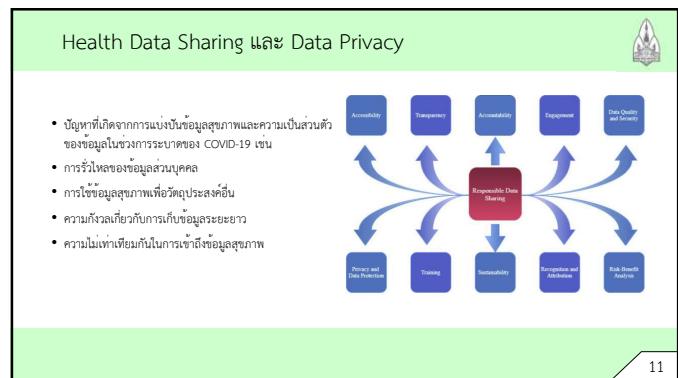
9



10



11



12

### การรับไฟล์ของข้อมูลส่วนบุคคล

• มีกรณีที่อยู่ส่วนบุคคลของผู้ติดเชื้อ COVID-19 รั่วไหลสู่สาธารณะ เช่น โซเชียล และประวัติการเดินทาง

• การรับไฟล์นี้นำไปถูกการพิเคราะห์และภาระเสือปูต่อบุคคลเหล่านั้น รวมถึงครอบครัวและเพื่อนของพากษา

• สิ่งนี้แสดงให้เห็นถึงความสำคัญของการบันทึกความเป็นส่วนตัวของข้อมูลและการมีมาตรการรักษาความปลอดภัยที่เหมาะสม



12

13

### การใช้ข้อมูลสุขภาพเพื่อวัตถุประสงค์อื่น

• เมื่อร่วมกับกลุ่มภัยคุกคามที่ร่วมมระหว่างภาระด้วยกัน เช่น การฟาร์มาหรือ เป้าหมายของการผลิต

• การใช้ข้อมูลของเหล่าจากวัตถุประสงค์เดิมโดยไม่ได้รับความยินยอมจากบุคคลนั้นถือเป็นการละเมิดความเป็นส่วนตัวและ อาชญากรรมทางอาชญากรรม

• จึงเป็นหัวข้อการบังคับและขอซึ่งกันที่จะลงโทษภัยคุกคามที่ละเมิดความเป็นส่วนตัว



13

14

### ความกังวลเกี่ยวกับการเก็บข้อมูลรายยิ่ว

• การเก็บรวบรวมข้อมูลสุขภาพขั้นวนมากจะห่างไกลจากการดำเนินการที่เกี่ยวกับระยะเวลาที่ข้อมูลถูกเก็บไปแล้วและโครงสร้างมีการเปลี่ยนแปลง

• ความกังวลเกี่ยวกับผลกระทบระยะยาวที่อาจเกิดขึ้นจากการเก็บรวบรวมข้อมูลจำนวนมาก เช่น การใช้ในอนาคตเพื่อการเสือปูติหรือบัญสิลโภต

• จึงเป็นหัวข้อในการบังคับและขอซึ่งกันที่จะลงโทษภัยคุกคามที่ละเมิดความเป็นส่วนตัว

14

15

### ความไม่เท่าเทียมกันในการเข้าถึงข้อมูลสุขภาพ

• การระบาดใหญ่ส่งผลกระทบต่อชุมชนบางแห่งอย่างเป็นสัดส่วน เช่น ชนกลุ่มน้อยทางเชื้อชาติและชาติพันธุ์ และประชากรที่มีรายได้ต่ำ

• การเข้าถึงข้อมูลสุขภาพและทรัพยากรบัณฑิตที่ไม่เท่าเทียมกันภายใต้เงื่อนไขของรัฐธรรมนูญที่สำคัญ เมื่อจากความไม่เท่าเทียมกัน อาจนำไปสู่ผลลัพธ์ด้านสุขภาพที่ไม่ดีหรือร้ายแรงที่มีความเสี่ยง

• จึงเป็นหัวข้อความพยายามเชิงรุกที่ออกตามอุปสรรคในการเข้าถึงและรับประทานอาหารจากชุมชนและหัวใจการด้านสุขภาพ อย่างเป็นธรรมในทุกประชารัฐที่หลากหลาย

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8988992/>

15

16

### Quiz

• 1. เขียน function ที่รับ input เป็น output ของ boxplot แล้ว แสดงค่า min, max, q1, q2, q3

• 2. วัด boxplot เปรียบเทียบ การกระจายตัวของข้อมูล PetalLength ของตอ iris ทั้ง 3 ชนิด

16

17

## Class period 16

บทที่ 6 การแสดงผลการกราฟ化ของข้อมูล (ต่อ)

timestamp - datetime

1

## Quiz

- [https://drive.google.com/drive/folders/1AztyMSDdZiwjDHfSl0T51VSCYVRyQ2Z3?fbclid=IwAR1Ta6x5C4Yck3-SPxK4jY2EARAYFW5HTBwt\\_xFDH5IP1N4e4kpunh8](https://drive.google.com/drive/folders/1AztyMSDdZiwjDHfSl0T51VSCYVRyQ2Z3?fbclid=IwAR1Ta6x5C4Yck3-SPxK4jY2EARAYFW5HTBwt_xFDH5IP1N4e4kpunh8)
- ไฟล์ไฟล์ example\_timestamp.csv
- เบรียบด้วยผลรวมของข้อมูล alpha และ beta ก่อนวันที่ 2 มิถุนายน 2020 และตั้งแต่วันที่ 2 มิถุนายน 2020

2

## Timestamp - datetime

timestamp ให้แปลงรูปแบบข้อมูลวันเดือนปีเป็นตัวจีน ให้เป็นรูปแบบ timestamp  
การใช้งาน timestamp จะช่วยให้สามารถเข้ารหัสหรือเรียกข้อมูลวันเดือนปีโดยจะต้องจากการเข้ารหัสหรือเรียกข้อมูลแบบปกติในรูปแบบ string

timestamp จะสามารถกรอกหน่วยรูปแบบข้อมูลวันเดือนปีที่ต้องการและแปลงรูปแบบข้อมูลเป็นรูปแบบ timestamp การใช้งานจะมีดังนี้  
เพรียบเทียบว่า เทrebะบางค่าจะใช้รูปแบบในการเก็บข้อมูลวันเดือนปีทางกัน

datetime ใช้ในการเข้ารหัสรูปแบบ timestamp

3

## ยกตัวอย่างการซัพพอร์ตตัวอักษรในรูปแบบ string ธรรมดากับตัวอักษรในรูปแบบ timestamp

ตัวอย่างข้อมูลวันเดือนปีจากอัลลีน Unnamed: 0 จะเรียกว่าตัวตามหลักการเรียงของข้อมูล string ที่นี่ฐาน การใช้การซึ่งกันและกัน

| Unnamed: 0 | alpha      | beta  |
|------------|------------|-------|
| 0          | 2020-05-29 | 8.78  |
| 1          | 2020-05-30 | 13.00 |
| 2          | 2020-05-31 | 0.44  |
| 3          | 2020-06-01 | 1.94  |
| 4          | 2020-06-02 | 5.40  |
| 5          | 2020-06-03 | 5.68  |
| 6          | 2020-06-04 | 2.64  |

4

## ยกตัวอย่างการซัพพอร์ตตัวอักษรในรูปแบบ string ธรรมดากับตัวอักษรในรูปแบบ timestamp

เปลี่ยนรูปแบบวันเดือนปี และลดลงและเรียงแบบ string

| Unnamed: 0 | alpha      | beta  |
|------------|------------|-------|
| 0          | 29-05-2020 | 8.78  |
| 1          | 30-05-2020 | 13.00 |
| 2          | 31-05-2020 | 0.44  |
| 3          | 01-06-2020 | 1.94  |
| 4          | 02-06-2020 | 5.40  |
| 5          | 03-06-2020 | 5.68  |
| 6          | 03-06-2020 | 2.64  |

จะเห็นว่า เรียงไม่ถูก เพราะการเรียงแบบพื้นฐานของ string จะเรียงจากหน้าไปหลัง, 00 ถึง 10 หรือ 000 ถึง 999  
ตัวบ่งชี้ตัวอักษรจะเริ่มจาก Aa หรือ กก ตั้งนั้นจะมีผล 01-06-2020 ที่น้อยกว่า 02-06-2020

5

## การใช้ timestamp (.to\_datetime)

- การใช้งาน
  - import pandas as pd
  - pd.to\_datetime('ข้อมูลคันนี้คือต้องการ', format='%d-%m-%Y')
  - โดย Input format จะใช้กันครับแบบพื้นเดือนปีของข้อมูล input
  - %d คือ วัน
  - %m คือ เดือน
  - %Y คือ ปี
  - รูปแบบข้อมูลในคอลัมน์คือ 01-06-2020 format='%d-%m-%Y'
  - รูปแบบข้อมูลในคอลัมน์คือ 2020-06-01 format='%Y-%m-%d'
  - รูปแบบข้อมูลในคอลัมน์คือ 01/06/2020 format='%d/%m/%Y'

6

1

### ดึงข้อมูลตัวอย่าง

```

import pandas as pd
df = pd.read_csv('/content/example_timestamp_03.csv')
df

```

|   | Unnamed: 0 | alpha | beta |
|---|------------|-------|------|
| 0 | 29/05/2020 | 8.78  | 24   |
| 1 | 30/05/2020 | 13.00 | 25   |
| 2 | 31/05/2020 | 0.44  | 25   |
| 3 | 1/6/2020   | 1.94  | 28   |
| 4 | 2/6/2020   | 5.40  | 20   |
| 5 | 3/6/2020   | 5.68  | 21   |
| 6 | 3/6/2020   | 2.64  | 16   |

7

### ตัวอย่างการใช้ .to\_datetime แปลงรูปแบบข้อมูล

```

df['Unnamed: 0'] = pd.to_datetime(df['Unnamed: 0'], format='%d/%m/%Y')
df

```

|   | Unnamed: 0 | alpha | beta |
|---|------------|-------|------|
| 0 | 2020-05-29 | 8.78  | 24   |
| 1 | 2020-05-30 | 13.00 | 25   |
| 2 | 2020-05-31 | 0.44  | 25   |
| 3 | 2020-06-01 | 1.94  | 28   |
| 4 | 2020-06-02 | 5.40  | 20   |
| 5 | 2020-06-03 | 5.68  | 21   |
| 6 | 2020-06-03 | 2.64  | 16   |

8

### การซื้อข้อมูล timestamp ด้วย datetime

- การซื้อข้อมูล timestamp จะใช้ค่าสั่ง datetime สามารถกำหนดวันเดือนปีในการซื้อ
- datetime(day=วันที่ต้องการซื้อ, month=เดือนที่ต้องการซื้อ, year=ปีที่ต้องการซื้อ)
- เช่น

```

from datetime import datetime
df[df['Unnamed: 0']<datetime(day=2,month=6,year=2020)]

```

|   | Unnamed: 0 | alpha | beta |
|---|------------|-------|------|
| 0 | 2020-05-29 | 8.78  | 24   |
| 1 | 2020-05-30 | 13.00 | 25   |
| 2 | 2020-05-31 | 0.44  | 25   |
| 3 | 2020-06-01 | 1.94  | 28   |

9

### การใช้ .sum() รวมข้อมูลตาม timestamp

- ซื้อข้อมูล timestamp ที่ต้องการโดยใช้ค่าสั่ง datetime ตามด้วย sum() และกำหนดค่ามเปนเก็บค่า เช่น

```

A = data[data['Unnamed: 0']<datetime(day=2,month=6,year=2020)].sum()
print(A)

```

|  | alpha          | beta   |
|--|----------------|--------|
|  | 24.16          | 102.00 |
|  | dtype: float64 |        |

```

A['alpha']

```

24.160000000000004

10

### การใช้ .sum() รวมข้อมูลตาม timestamp

- ซื้อข้อมูล timestamp ที่ต้องการโดยใช้ค่าสั่ง datetime ตามด้วย sum() และกำหนดค่ามเปนเก็บค่า เช่น

```

BF = data[data['Unnamed: 0']<datetime(day=2,month=6,year=2020)].sum()
AT = data[data['Unnamed: 0']>=datetime(day=2,month=6,year=2020)].sum()
print(BF)

alpha 24.16
beta 102.00
dtype: float64

print(AT)
alpha 13.72
beta 57.00
dtype: float64

```

11

2

Class period 17

บทที่ 7 การแสดงผลการเบรย์บเพื่อข้อมูล  
Visualize\_Data\_Distribution\_(PCA)

1

Projection

- การถ่ายແສໄສຈຸດຂອ້ມູນໄລ້ໄວ້ຮ່ວມຈຸດໃຫຍກທີ່ກໍານົດ
- ຊັກຄຽກຢືນກວດວ່າການຈົບປັດຂອ້ມູນ

2

PCA (Principal component Analysis)

- PCA ຄູ່ການແກນໃໝ່ເພີ່ມຕົວອີເມວຂອງຂອ້ມູນໄດ້ຕີ່ສຸດ ເນື້ນ ດັວນປະ ທີ່ຈະນຳນາມແຄດກາກຮະຈາຍຂອງຂອ້ມູນ
- ການນຳກາວ 2 ຕົ້ນປະ ສາມາດໃຊ້ PCA (Principle Component Analysis) ເພື່ອຈົບປັດວ່າມີແປຣມໄດ້ໂດຍຮັກສິກະນະກາກຮະຈາຍຂອ້ມູນໄລ້ດີນາກ໌ສຸດ

3

sklearn -> scikit-learn

- <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
- ເປັນ package ທີ່ຮັບຮັນ function ການທີ່ Data Science - Machine Learning - Data Mining ເຊົາໄວ້ຈົບປັດນີ້ຕ້ອງເຊີນເຊິ່ງ
- ການຈົບປັດ import PCA ຂອງ sklearn
- from sklearn.decomposition import PCA

4

การໃຊ້ PCA ມີ 3 ຊັ້ນຕອນ

- 1. Import
- from sklearn.decomposition import PCA
- 2. Define
- pca = PCA()
- 3. Fit – Transform ຄູ່ ສໍາສັນເກີດໃຫຍກການແກນໃໝ່ເພີ່ມຕົວອີເມວກາກຮະຈາຍຂອງຂອ້ມູນໄດ້ຕີ່ສຸດ
- new\_axis = pca.fit\_transform('ທີ່ແປຣມທີ່ເປັນກົບຂອ້ມູນທີ່ອອກການທີ່PCA')

5

ເຕີມຂອ້ມູນດອກໄມ້ iris

```

import pandas as pd
example_df = pd.read_csv('https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/io/data/csv/iris.csv')
thisdata = example_df.iloc[:, :-1]
thisdata

```

|     | SepalLength | SepalWidth | PetalLength | PetalWidth |
|-----|-------------|------------|-------------|------------|
| 0   | 5.1         | 3.5        | 1.4         | 0.2        |
| 1   | 4.9         | 3.0        | 1.4         | 0.2        |
| 2   | 4.7         | 3.2        | 1.3         | 0.2        |
| 3   | 4.6         | 3.1        | 1.5         | 0.2        |
| 4   | 5.0         | 3.6        | 1.4         | 0.2        |
| ... | ...         | ...        | ...         | ...        |
| 145 | 6.7         | 3.0        | 5.2         | 1.8        |
| 146 | 6.5         | 2.9        | 5.0         | 1.8        |
| 147 | 6.5         | 3.0        | 5.2         | 2.0        |
| 148 | 6.2         | 3.4        | 5.4         | 2.3        |
| 149 | 5.9         | 3.0        | 5.1         | 1.8        |

150 rows × 4 columns

6

1

### เริ่มทำ PCA

```

from sklearn.decomposition import PCA
pca = PCA()
new_axis = pca.fit_transform(thisdata)
new_axis.shape
new_axis

```

6

7

### ผลลัพธ์การทำ PCA

- จะได้ชุดข้อมูลที่ถูกยูนแมเนลล์ จำนวนข้อมูลเท่ากับข้อมูลที่ input ให้ที่ PCA
- new\_axis.shape จะได้ผลลัพธ์ (150, 4)
- new\_axis จะได้ผลลัพธ์ที่ใช้มาเพื่อถูกยูนแมเนลล์ ในรูปแบบ numpy array

```

array([[-5.65020714e+00, 3.26607315e-01, -2.15118370e-02,
 1.08613724e-03], [-2.75529802e+00, -1.69556848e-01, -2.03521425e-01,
 9.96024240e-02], [-2.88981954e+00, -1.37545610e-01, 2.470922410e-02,
 1.93821510e-02], [-2.74643720e+00, -3.11124316e-01, 3.76719753e-02,
 -7.59552741e-02], [-2.27989736e+00, 3.33924964e-01, 9.62296999e-02,
 -6.21207327e-02], [-2.79897327e+00, 7.47782713e-01, 1.743255619e-01,
 -2.71468037e-02], [-2.82089688e+00, -8.21045110e-02, 2.64251085e-01],
 ...])

```

7

8

### แปลงข้อมูล PCA ให้อยู่ในรูปแบบข้อมูลตาราง

|     | PCA1      | PCA2      | PCA3      | PCA4      |
|-----|-----------|-----------|-----------|-----------|
| 0   | -0.264837 | 0.236947  | -0.220151 | 0.001908  |
| 1   | -2.755298 | -0.169557 | -0.232021 | 0.096602  |
| 2   | -0.889819 | -0.137546 | 0.024719  | 0.079399  |
| 3   | -2.889819 | -0.111243 | 0.027023  | -0.073955 |
| 4   | -2.746437 | 0.339249  | 0.049263  | -0.081722 |
| ... | ...       | ...       | ...       | ...       |
| 145 | 1.344617  | 0.178470  | 0.170203  | 0.420202  |
| 146 | 1.325946  | -0.170752 | -0.120904 | 0.237722  |
| 147 | 1.744646  | 0.170719  | 0.130784  | 0.163265  |
| 148 | 1.919129  | 0.115897  | 0.022074  | 0.049393  |
| 149 | 1.086469  | -0.183087 | 0.160318  | -0.165610 |

150 rows × 4 columns

8

9

### pca.explained\_variance\_ratio\_

- ใช้ประสาทสัมภาระของการกระจายข้อมูล ตามจำนวนเงิน เช่น
- array([0.92461621, 0.05301557, 0.01718514, 0.00518309])
- 0.92461621 คือ ใช้เงิน 1 แสนบาทขอรับการกระจายข้อมูลได้ 92.4%
- 0.05301557 คือ ใช้เงิน 2 แสนบาทขอรับการกระจายข้อมูลได้ 92.4+5.3 = 97.7%

9

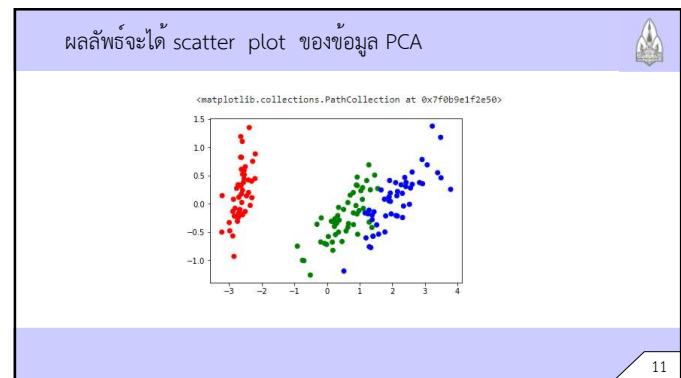
10

### plot PCA data

- นำตารางข้อมูล PCA มาสร้างกราฟเพื่อดูสามารถอธิบายการกระจายตัวของข้อมูลได้ที่ดีสุด
- from matplotlib import pyplot as plt
- example\_df2 = example\_df.replace({'Iris-setosa':'r', 'Iris-versicolor':'g', 'Iris-virginica':'b'})
- plt.scatter(PCAdf['PCA1'],PCAdf['PCA2'],c=example\_df2['Name'])
- plt.scatter(example\_df2['SepalWidth'],example\_df2['PetalWidth'],c=example\_df2['Name'])

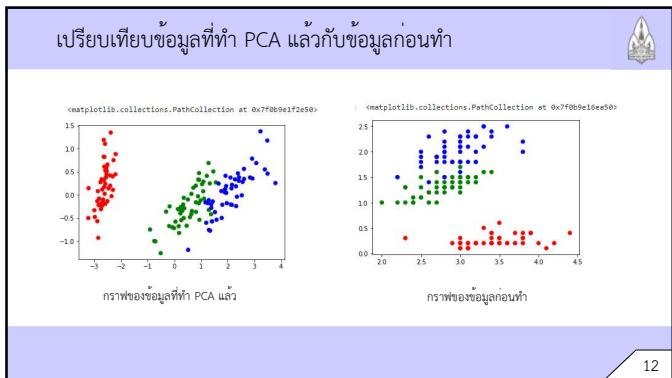
10

11



12

2



13



## Class period 18

บทที่ 7 การแสดงผลการเรียนเพิ่มข้อมูล (ต่อ)

Visualize\_Data\_Comparison  
Bar chart - Grouped Bar chart

1



### โหลดข้อมูล

- ให้เก็บค่าลงในไฟล์ชื่อ `search_request.csv` จาก
 [https://drive.google.com/drive/folders/1Az1YMSDdZiwjDHfsl0T51VSCYVRyQ2Z?fbclid=IwAR1Ta6xSC4Yck3-Spxk4jY2EA8AYFW5HTBwt\\_xFhD15IP1N4e4kpuuh8](https://drive.google.com/drive/folders/1Az1YMSDdZiwjDHfsl0T51VSCYVRyQ2Z?fbclid=IwAR1Ta6xSC4Yck3-Spxk4jY2EA8AYFW5HTBwt_xFhD15IP1N4e4kpuuh8)
- อัปโหลดไฟล์นี้ลง google drive แล้วอ่านและเขียน google drive ในสคริปต์ใน colab

```

from google.colab import drive
import pandas as pd
import os
from datetime import datetime as dt
from datetime import time

```

1

2



### เตรียมและแปลงข้อมูลของเวลาให้เป็นตัวแปรชนิด timestamp

```

drive.mount('/content/drive')
path = '/content/drive/My Drive/dataviz_2024_data'
data = pd.read_csv(os.path.join(path, 'search_request.csv'))
data.head()
data['search_timestamp'] =
pd.to_datetime(data['search_timestamp'], format='%Y-%m-%d %H:%M:%S', errors='coerce')

```

3



### Quiz

- สร้างกราฟแท่งบาร์เรียงตัวยิงบริษัท คนเข้าใช้ web Wongnai.com เพื่อศึกษาเรื่องอาชญากรรมในแต่ละวัน

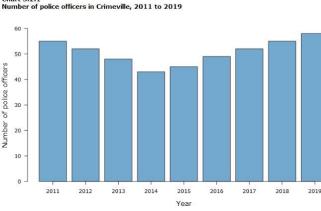
4



### Bar chart (กราฟแท่ง)

- กราฟแท่งในกราฟเรียบเพิ่มขึ้นตามอัตราบัญชี
- จำนวนของกราฟแท่ง
- ตัวกราฟแท่ง (height)
- ตัวแกน x กราฟแท่ง (x)
- ชื่อแท่ง (tick\_label)
- ชื่อกราฟ (plt.title)
- ชื่อแกน x (plt.xlabel)
- ชื่อแกน y (plt.ylabel)

**Chart 5.2.1**  
Number of police officers in Crimeville, 2011 to 2019



| Year | Number of Police Officers |
|------|---------------------------|
| 2011 | 55                        |
| 2012 | 50                        |
| 2013 | 48                        |
| 2014 | 45                        |
| 2015 | 48                        |
| 2016 | 50                        |
| 2017 | 52                        |
| 2018 | 55                        |
| 2019 | 58                        |

5



### การแสดงผลตัวอักษรภาษาไทยในกราฟ matplotlib

- ใช้ชื่อกำกับ

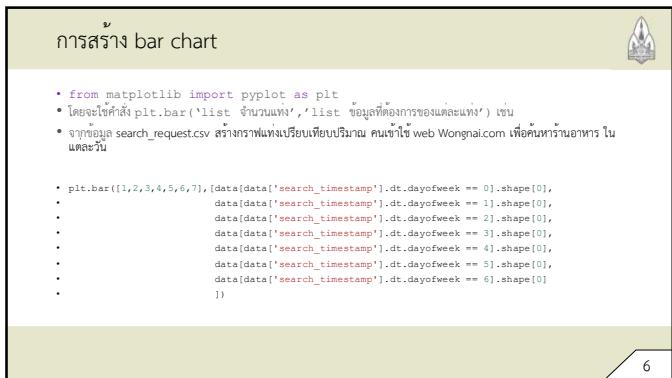
```

https://medium.com/@kanyawee.work/%E0%B9%81%E0%B8%AA%E0%B8%94%E0%B8%87%E0%B8%9C%E0%B8%A5%E0%B8%A0%E0%B8%82%E0%B8%A9%E0%B8%82%E0%B8%84%E0%B8%97%E0%B8%A2%E0%B8%9A%E0%B8%88%E0%B8%99%E0%B8%81%E0%B8%93%E0%B8%82%E0%B8%87%E0%B8%9F%E0%B8%82%E0%B8%AD%E0%B8%87matplotlib-%E0%B8%9A%E0%B8%90-google-colab-37210d9a9f31
https://colab.research.google.com/drive/1stTdITz_Cm51mc8Q_L_0htehWyo4725sGI#scrollTo=Ak9SbVd5L7i8
!wget -q https://github.com/Phonbopit/sarabun-webfont/raw/master/fonts/thsarabunnew-webfont.ttf
import matplotlib as plt
mpl.font_manager.fontManager.addfont('thsarabunnew-webfont.ttf')
mpl.rc('font', family='TH Sarabun New')

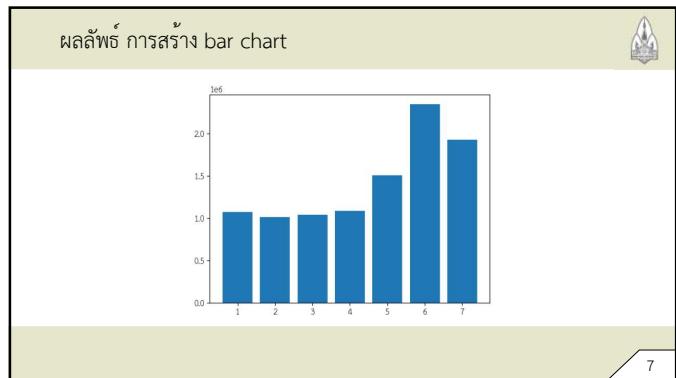
```

6

1



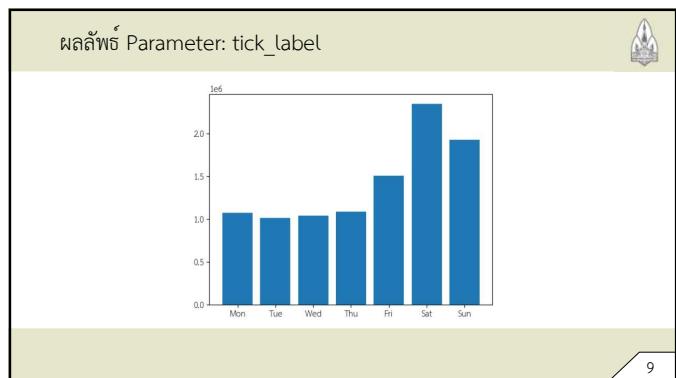
7



8



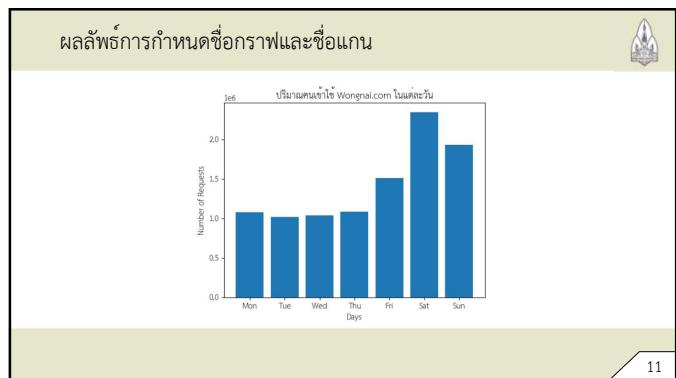
9



10

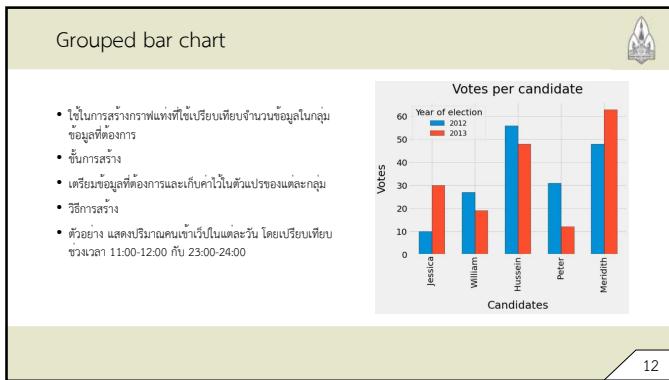


11



12

2



13



13

เตรียมข้อมูลและกลุ่ม

```
b4lunch = [data[(data['search_timestamp'].dt.dayofweek == 0)&(data['search_timestamp'].dt.time < time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 1)&(data['search_timestamp'].dt.time < time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 2)&(data['search_timestamp'].dt.time < time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 3)&(data['search_timestamp'].dt.time < time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 4)&(data['search_timestamp'].dt.time < time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 5)&(data['search_timestamp'].dt.time < time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 6)&(data['search_timestamp'].dt.time < time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 7)&(data['search_timestamp'].dt.time < time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 0)&(data['search_timestamp'].dt.time > time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 1)&(data['search_timestamp'].dt.time > time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 2)&(data['search_timestamp'].dt.time > time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 3)&(data['search_timestamp'].dt.time > time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 4)&(data['search_timestamp'].dt.time > time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 5)&(data['search_timestamp'].dt.time > time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 6)&(data['search_timestamp'].dt.time > time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 7)&(data['search_timestamp'].dt.time > time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 0)&(data['search_timestamp'].dt.time > time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 1)&(data['search_timestamp'].dt.time > time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 2)&(data['search_timestamp'].dt.time > time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 3)&(data['search_timestamp'].dt.time > time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 4)&(data['search_timestamp'].dt.time > time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 5)&(data['search_timestamp'].dt.time > time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 6)&(data['search_timestamp'].dt.time > time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 7)&(data['search_timestamp'].dt.time > time(hour=23))].shape[0]
]
b4lunch
[73249, 77083, 75429, 78024, 90007, 374215, 165448]

b4midnight = [data[(data['search_timestamp'].dt.dayofweek == 0)&(data['search_timestamp'].dt.time >= time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 1)&(data['search_timestamp'].dt.time >= time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 2)&(data['search_timestamp'].dt.time >= time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 3)&(data['search_timestamp'].dt.time >= time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 4)&(data['search_timestamp'].dt.time >= time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 5)&(data['search_timestamp'].dt.time >= time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 6)&(data['search_timestamp'].dt.time >= time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 7)&(data['search_timestamp'].dt.time >= time(hour=23))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 0)&(data['search_timestamp'].dt.time <= time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 1)&(data['search_timestamp'].dt.time <= time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 2)&(data['search_timestamp'].dt.time <= time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 3)&(data['search_timestamp'].dt.time <= time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 4)&(data['search_timestamp'].dt.time <= time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 5)&(data['search_timestamp'].dt.time <= time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 6)&(data['search_timestamp'].dt.time <= time(hour=11))].shape[0],
 data[(data['search_timestamp'].dt.dayofweek == 7)&(data['search_timestamp'].dt.time <= time(hour=11))].shape[0]
]
b4midnight
[31874, 32258, 31153, 35944, 53174, 58306, 35801]
```

14

กำหนดตัวแปร labels ใช้เก็บชื่อแท่งกราฟ

```
labels = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
```

15

เตรียมตัวแปรที่ใช้สร้างกราฟ

```
import numpy as np
x = np.arange(len(labels)) # กำหนดตัวแหนงเพื่อหามุมของกราฟในตัวแปร labels
width = 0.35 # ความกว้างของกราฟ

fig, ax = plt.subplots() # ใช้ subplots ในการสร้างกราฟต่อไป
```

16

สร้างกราฟ

```
rects1 = ax.bar(x - width/2, b4lunch, width, label='lunch time', color = '#fc9700') # รูบบ4lunch
rects2 = ax.bar(x + width/2, b4midnight, width, label='midnight', color = '#19038a') # รูบบ4midnight

แท่งกราฟที่อยู่ด้านล่างของกราฟจะถูกสร้างก่อนต่อไป แท่งกราฟที่อยู่ด้านบนจะถูกสร้างหลังจากตัวแปร rects1, rects2
```

17

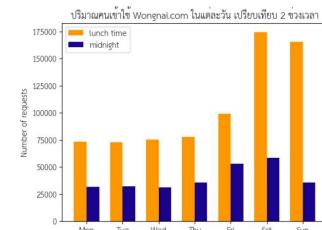
## กำหนดส่วนประกอบกราฟต่างๆ

```
• ax.set_ylabel('Number of requests')
• ax.set_title('ปริมาณคนเข้าใช้ Wongnai.com ในแต่ละวัน เปรียบเทียบ 2 ช่วงเวลา')
• ax.set_xticks(x)
• ax.set_xticklabels(labels)
• ax.legend();
```



18

ผลลัพธ์จะได้ Grouped bar chart ของ ปริมาณคนเข้าใช้ Wongnai.com ในแต่ละวัน เปรียบเทียบ 2 ช่วงเวลา



19

20

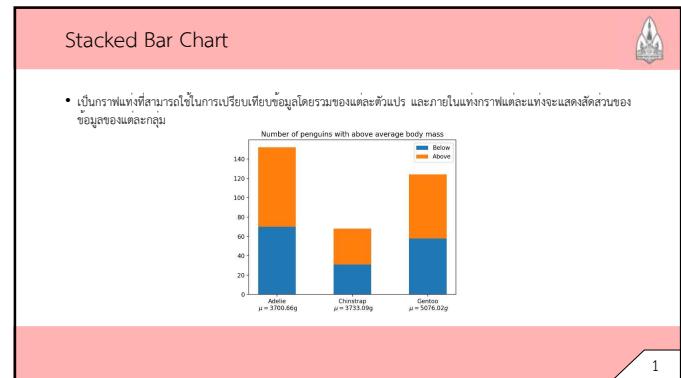


## Class period 19

บทที่ 7 การแสดงผลการเรียนที่ใช้บล็อก (ต่อ)

Stacked bar chart and histogram

1



2



### ขั้นตอนการสร้าง Stacked Bar Chart

- ขั้นตอนการสร้างเหมือนกับ Grouped bar chart
- 1. เริ่มจากนำข้อมูลลงในรูปที่ต้องการและเก็บค่าวันเปรเพื่อเป็นกราฟแท่ง (จะใช้ชื่อ b4lunch, b4midnight ที่เตรียมไว้ก่อนหน้านี้)
- 2. ก้าวหนึ่งนั่นเปร labels ให้กับข้อมูลกราฟ (จะใช้ ดัชนีเปร labels ที่เตรียมไว้ก่อนหน้านี้)
- 3. นำดัชนีเปรที่เตรียมไว้มาสร้างกราฟ

3



### เตรียมตัวแปรที่ใช้สร้างกราฟ

```

import matplotlib.pyplot as plt
width = 0.35
fig, ax = plt.subplots()

```

4



### สร้างกราฟ

```

ax.bar(labels, b4lunch, width, label='before lunch', color = '#fc9700')
ax.bar(labels, b4midnight, width, bottom=b4lunch, label='before midnight', color = '#19038a')

```

Parameter: bottom ใช้ในการกำหนดฐานเดิมต่อกราฟให้เราสามารถต่อกราฟต่อ เช่น

- กำหนด **bottom=b4lunch** ในกราฟกราฟที่อยู่ข้อมูล b4midnight
- หมายถึงหัวกราฟของข้อมูล b4midnight ต้อนกราฟของข้อมูล b4lunch

5



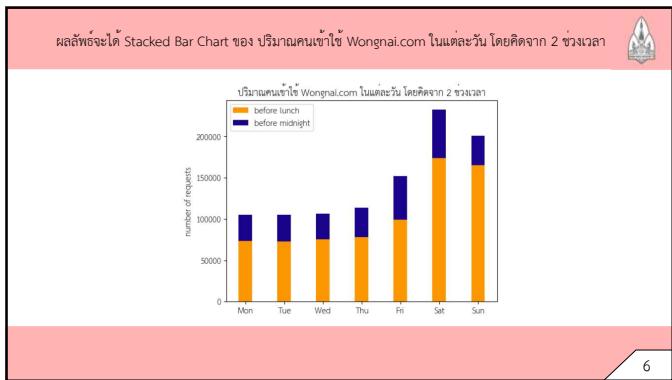
### กำหนดส่วนประกอบกราฟต่างๆ

```

ax.set_ylabel('number of requests')
ax.set_title('จำนวนคำขอใช้ Wongnai.com ในแต่ละวัน โดยคิดจาก 2 ช่วงเวลา')
ax.legend()
plt.show() คำสั่งใช้แสดงกราฟ

```

6



7

### การสร้างกราฟแท่งแนวนอน

- โดยใช้คำสั่ง ax.bart ใน การสร้างกราฟแท่งแนวนอน เช่น

```

• ax.bart(labels, b4lunch, width, label='before lunch',color = '#fc9700')

• ax.bart(labels, b4midnight, width, left=b4lunch, label='before
midnight',color = '#19038a')

```

8

### ตัวอย่างการสร้างกราฟแท่งแนวนอน

```

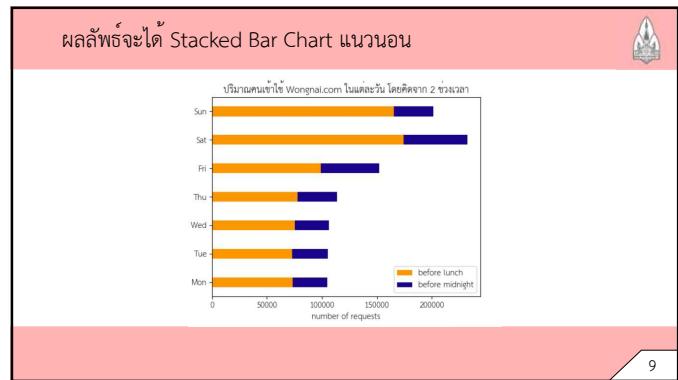
width = 0.35
fig, ax = plt.subplots()

ax.bart(labels, b4lunch, width, label='before lunch',color = '#fc9700')
ax.bart(labels, b4midnight, width, left=b4lunch, label='before
midnight',color = '#19038a')

ax.set_xlabel('number of requests')
ax.set_title('บริษัท Wongnai.com ในแต่ละวัน โดยคิดจาก 2 ช่วงเวลา')
ax.legend()
plt.show()

```

9



10

### Quiz (15 นาที)

- ให้ให้ว่า Stacked Bar Chart ที่ปรับเปลี่ยนรูปแบบคุณใช้งาน Wongnai.com ส่องช่วงเวลา โดยให้กราฟแสดงสัดส่วนของ บริษัท Wongnai.com ในแต่ละวัน

11



12

2

## การสร้าง Histogram

- สามารถสร้างได้ด้วยคำสั่ง
- `plt.hist('ตัวแปรของข้อมูลที่ต้องการสร้างกราฟ', 'จำนวนของข้อมูลหรือจำนวนแท่ง (int)')`



13

## ตัวอย่างข้อมูลที่ random

```
* import numpy as np
* from matplotlib import pyplot as plt
*
* np.random.seed(2021)
*
* mu, sigma = 100, 15
* X = mu + sigma * np.random.randn(10000)
```

• random มาจาก normal distribution ที่มี mean = 100 และ stddev = 15 เก็บไว้ในตัวแปร X

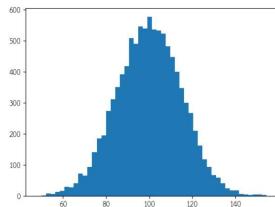


13

14

## สร้างกราฟ Histogram ของข้อมูลที่ random

```
* plt.hist(X, 50);
```

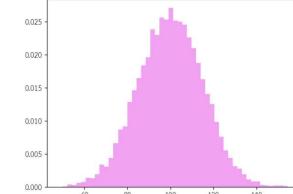


14

15

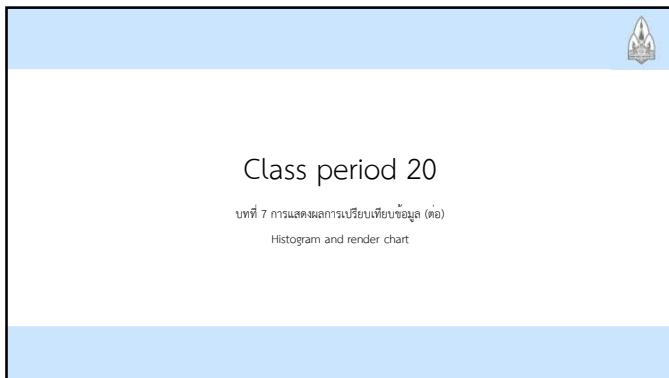
## เพิ่ม Parameter ต่างๆให้กับกราฟด้วยสวยงาม

```
* plt.hist(X, 50, density = True, facecolor = 'violet',
alpha = 0.75);
```



15

16



1

ตัวอย่างข้อมูล wongnai.com

```

• import matplotlib.pyplot as plt
• output = plt.hist(list(data['number_of_result']), 10, facecolor = 'red',
 alpha = 0.75)

• สร้างกราฟ Histogram โดยใช้ข้อมูลคอลัมน์ 'number_of_result' ใน ข้อมูล wongnai.com โดยกำหนดช่วงของข้อมูลเรื่อยๆ จำนวนห้าเบิน 10 กำกับด้วยสีแดง กำกับค่าความโปรดังเป็น 75%

```

1

2



3

แก้ไข แกน x ที่เรียงข้อมูลผิด

- 1. ตรวจสอบ data type ของ ตัวแปลง ด้วย
- data.dtypes
- 2. เรียกดูและตรวจสอบ data type ของที่เปลี่ยนไปคือลิมบ์ที่จะตัวด้วย
- type(data['number\_of\_result'][0])
- type(data['number\_of\_result'][0]) == int
- 3. ลองนั่งต้นเปลี่ยน type ข้อมูลเป็น int
- new\_type = data['number\_of\_result'].astype('int32')

3

4

ลบ record

- จากการตรวจสอบถ้า type ข้อมูลเป็น int จะเจอ Error ว่ามีค่าในคอลัมน์ 'number\_of\_result' ที่เป็น number\_of\_result ทำให้ไม่สามารถเปลี่ยน type ข้อมูลเป็น int ได้ ดังนั้นทำการใช้ drop ลบข้อมูลแรกที่มีค่าเป็น number\_of\_result ที่นี่
- 1. ตรวจสอบว่า record ใดก้างที่มีค่าเป็น number\_of\_result
- data[data['number\_of\_result']=='number\_of\_result']
- ผลลัพธ์จะได้ record ที่ 1000016
- 2. ลบ record ที่มีค่าเป็น number\_of\_result
- data = data.drop(1000016)

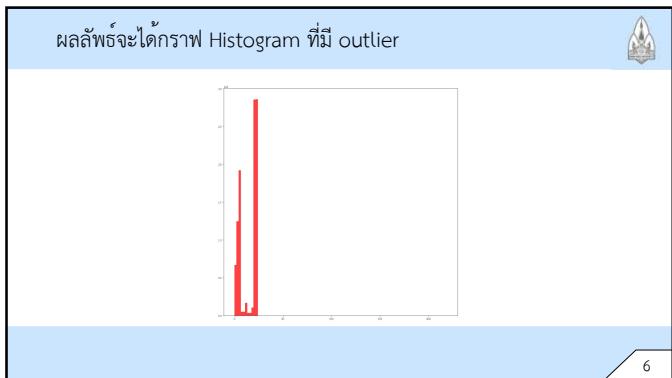
5

บังคับเปลี่ยน type ข้อมูลเป็น int

- บังคับเปลี่ยน type ข้อมูลเป็น int หลังจากลบข้อมูล record ที่ 1000016 และเปลี่ยนข้อมูลที่แปลงแล้วไว้ในตัวแปลง new\_type
- new\_type = data['number\_of\_result'].astype('int32')
- สร้างกราฟ Histogram ด้วยข้อมูลในตัวแปลง new\_type
- output = plt.hist(new\_type, 100, facecolor = 'red', alpha = 0.75)

6

1



7

ลบ outlier

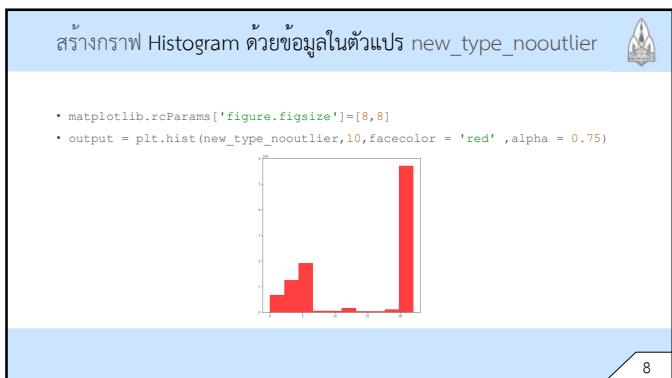
- ลบ outlier โดยการเลือกไข่ให้เก็บเฉพาะช่วงข้อมูลที่ค่าต่ำกว่า 25 ลงไปไว้ในตัวแปรใหม่
- `new_type_nooutlier = new_type[new_type < 25]`
- ตรวจสอบว่าข้อมูลที่เป็น outlier มีกี่ตัว
- `new_type.shape[0] - new_type_nooutlier.shape[0]`
- ผลลัพธ์จะได้รับ outlier ทั้งหมด 14 ตัว

`new_type.shape[0] - new_type_nooutlier.shape[0]`

14

7

8



9

Quiz

- เปรียบเทียบความถี่ของแท่งที่มีค่าน้ำหนักที่สูด กับ แท่งอื่นๆรวมกัน

9

10

เฉลย

- output คือคำสั่งที่ใช้ในการสร้างกราฟเป็นแท่งเที่ยง จากผลลัพธ์ที่ได้แต่กราฟหายคือ คำความถี่ที่มีค่าน้ำหนักที่สูด 5717238

```
(array([[670293., 1247269., 1921441., 51703., 50609., 167502.,
 36883., 35914., 105490., 5717238.]),
 array([0., 2.2, 4.4, 6.6, 8.8, 11., 13.2, 15.4, 17.6, 19.8, 22.]),
 <a list of 10 Patch objects>)
```

- output[0] เอาค่ามาแสดง array ตัวที่ 0 ของ output คือคำความถี่ของแท่งที่กราฟ

```
array([670293., 1247269., 1921441., 51703., 50609., 167502.,
 36883., 35914., 105490., 5717238.])
```

10

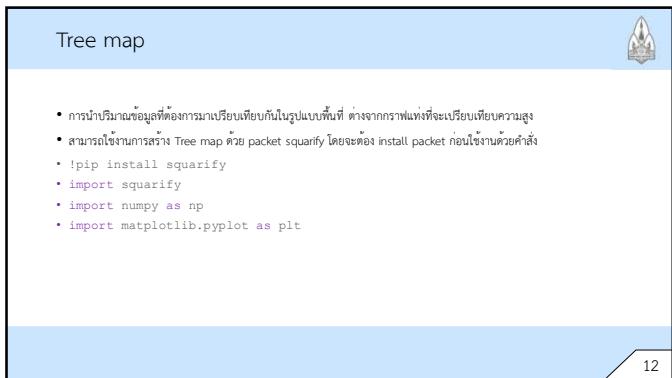
11

เฉลยต่อ

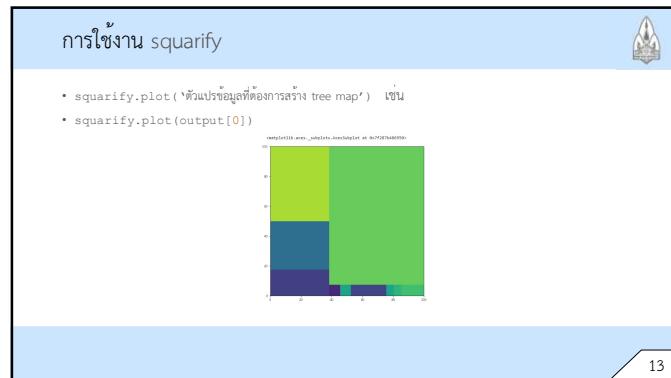
- output[0][-1] คำความถี่ของแท่งที่มีค่าน้ำหนักที่สูด 5717238.0
- `sum(output[0][:-1])` คำความถี่ของแท่งอื่นๆรวมกัน 4287104.0

11

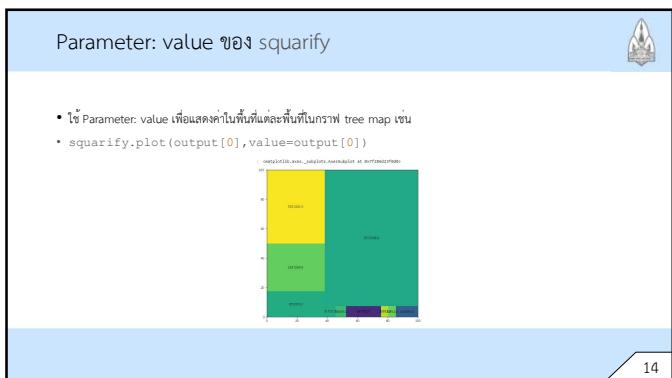
12



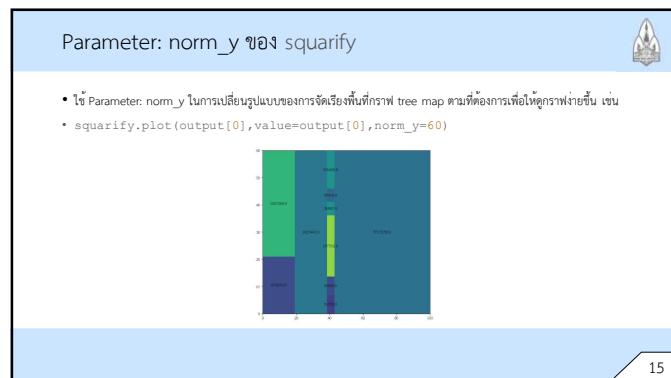
13



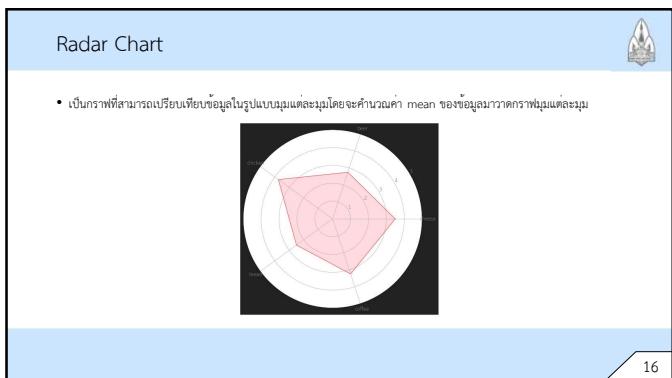
13



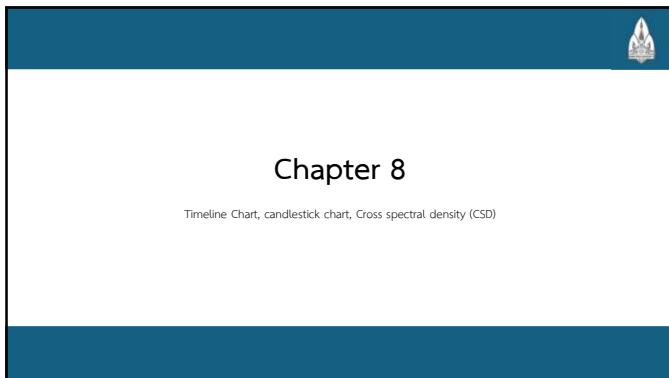
14



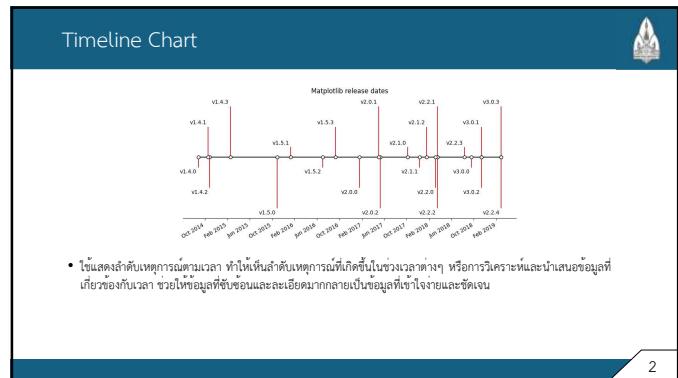
15



16



1



2

```
Import packet
```

3

### กำหนดข้อมูลใน list เก็บไว้ในตัวแปร

```
names = ['v2.2.4', 'v3.0.1', 'v3.0.2', 'v3.0.1', 'v3.0.0', 'v2.2.3',
 'v2.2.2', 'v2.2.1', 'v2.2.0', 'v2.1.2', 'v2.1.1', 'v2.1.0',
 'v2.0.2', 'v2.0.1', 'v2.0.0', 'v1.5.3', 'v1.5.2', 'v1.5.1',
 'v1.5.0', 'v1.4.3', 'v1.4.2', 'v1.4.1', 'v1.4.0']

dates = ['2019-02-26', '2019-02-26', '2018-11-10', '2018-11-10',
 '2018-09-18', '2018-08-10', '2018-03-17', '2018-03-16',
 '2018-03-06', '2018-01-18', '2017-12-10', '2017-10-07',
 '2017-05-10', '2017-05-02', '2017-01-17', '2016-09-09',
 '2016-07-03', '2016-01-10', '2015-10-29', '2015-02-16',
 '2014-10-26', '2014-10-18', '2014-08-26']
```

4

### แปลงข้อมูล datetime และกำหนดข้อมูล levels

```
dates = [datetime.strptime(d, "%Y-%m-%d") for d in dates]
วนลูปไปทุกอย่างในตัวปรับ dates ที่จะต้องรูปแบบ string เป็น datetime หน่วยทุกตัว และเป็นในตัวปรับ dates

levels = np.tile([-5, 5, -3, 3, -1, 1,
 int(np.ceil(len(dates)/6))], [len(dates)])
การกำหนดระดับความกว้างของเส้นที่ถูกจัดเรียง timeline
[-5, 5, -3, 3, -1, 1] คือกำหนดระดับและความกว้างสั้นต่อๆ กัน ในที่นี้คือ 6 เส้น
int(np.ceil(len(dates)/6)) คือที่กำหนดความกว้างสั้นที่ 6 จะตัด ตามจำนวนวันทั้งหมดใน dates
[len(dates)] คือกำหนดจำนวนวันทั้งหมด
```

5

### กำหนดขนาดและชื่อของกราฟ

```
fig, ax = plt.subplots(figsize=(8., 4), layout="constrained")
ax.set(title="Matplotlib release dates")

กำหนดขนาดของกราฟเป็นกว้าง 8.8 นิ้ว สูง 4 นิ้ว
layout="constrained" เป็น parameter ที่จะขยายตัวหน้าและปรับขนาดของมิติ subplot ให้มีขนาดที่เหมาะสม
ไม่ขอทึบกันและไม่เกินขอบเขตของ figure size ที่กำหนดไว้ ช่วยลดความซับซ้อนในการอ่านข้อมูลและทำให้กราฟสวยงาม
กำหนดชื่อกราฟ Matplotlib release dates
```

6

5

6

1

สร้างเส้น timeline ของกราฟ

- `ax.vlines(dates, 0, levels, color="tab:red")`
- เป็นการplotเส้นตัวเลข(dates) แต่จะเป็นเส้นหัวเริมจาก 0 ของเส้น timeline ไปยังระดับความยาวในค่าวีร์บแก้ levels ที่กำหนดไว้ และก้านคนสีเป็นสีแดง
- `ax.plot(dates, np.zeros_like(dates), "-o", color="k", markerfacecolor="w")`
- ใช้วิธีสร้างเส้น timeline และเครื่องหมายมาแทนค่าเวลา(dates) และชุด
- `np.zeros_like(dates)` คือการกันผลตุณเวลาทุกๆค่าใน dates จะมีค่าเท่ากับ 0 ทั้งหมดและลากเส้นราฟเป็นเส้นตรงเชื่อมระหว่างจุดๆกัน
- `"-o", color="k", markerfacecolor="w"` คือก้านคนสีครีมของมายเป็นสีดำ ก้านคนสีสีเหลืองเป็นสีขาว

7

#### กำหนดตำแหน่งข้อความในกราฟ

```

• for d, l, r in zip(dates, levels, names):
• ax.annotate(r, xy=(d, 1),
• xytext=(-3, np.sign(l)*3), textcoords="offset points",
• horizontalalignment="right",
• verticalalignment="bottom" if l > 0 else "top")

• for d, l, r in zip(dates, levels, names):
• plt.text(d+0.05, l-0.05, r, rotation=90)
• ax.annotate(r, xy=(d, 1),
• xytext=(0, -3), textcoords="offset points",
• horizontalalignment="center",
• verticalalignment="top")

```

8

#### กำหนดตำแหน่งข้อความในกราฟ

- `xytext=(-3, np.sign(1)*3)` ใช้กำหนดพื้นที่ของคำว่าที่จะแสดงในรูปแบบพิกเซล
- -3 แทนพื้นที่ของความในแนวนอน x คือด้านซ้ายของอุปกรณ์ที่กำกับดู
- `np.sign(1)*3` แทนพื้นที่ของความในแนวแกน y คือด้านบนหรือด้านล่างตามคำเรียบ `levels` ที่กำหนด

- `textcoords="offset points"` ต้องระบุว่ากำหนดพื้นที่ของข้อความที่แสดงจะใช้ระบบพิกเซล
- `horizontalalignment="right"` ต้องการจัดศักดิ์แท้เมืองที่แสดงของความไวให้อยู่ขวาของข้อความในแนวนอน x
- `verticalalignment="bottom"` if `l > 0 else "top"`
- ใช้จัดแนวนี้ของคำว่าให้ดูสวยงามของอุปกรณ์ที่กำกับดู หมายความว่า 0 ถ้าอยู่กว่า 0 ให้จัดแนวนี้ของความไวให้อยู่ด้านบน เพื่อปรับให้ดูสวยงามของความไวในอุปกรณ์ที่ไม่สามารถตั้งค่า levels ที่กำหนด

9

กำหนดเครื่องหมายและข้อความบนแกน x

- ax.xaxis.set\_major\_locator(mdates.MonthLocator(interval=4))
- ax.xaxis.set\_major\_formatter(mdates.DateFormatter("%b %Y"))
- plt.setp(ax.get\_xticklabels(), rotation=30, ha="right")
  
- ax.xaxis.set\_major\_locator(mdates.MonthLocator(interval=4))
- ໃຫ້ການດັບທີ່ແມ່ນເຊື່ອຮັບມານັ້ນ x ໂດຍສະຫງຸດ 4 ເຕັມ
  
- ax.xaxis.set\_major\_formatter(mdates.DateFormatter("%b %Y"))
- ໃຫ້ການດູແນວມານັ້ນທີ່ກວມທີ່ຈະໄຟລ່ອງສະໜັນ x %b ອີ່ຂອງເຄືອແບບໝັ້ນ %y ອີ່ປີ
  
- plt.setp(ax.get\_xticklabels(), rotation=30, ha="right")
- ໃຫ້ການດູແນວມານັ້ນທີ່ກວມທີ່ຈະໄຟລ່ອງສະໜັນ x ແລ້ວ 30 ລວມ (ເພື່ອຫຼັກສຳເນົາໃຫ້ອັງກິດ)

10

## กำหนดส่วนของกลุ่มต่างๆ องกราฟ

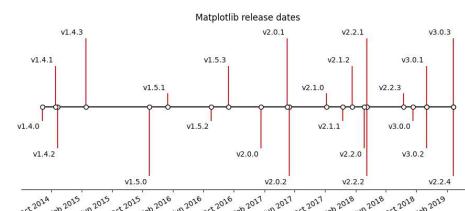
- `ax.yaxis.set_visible(False)`
- `ax.spines[["left", "top", "right"]].set_visible(False)`
- `ax.margins(y=0.1)`
- `plt.show()`

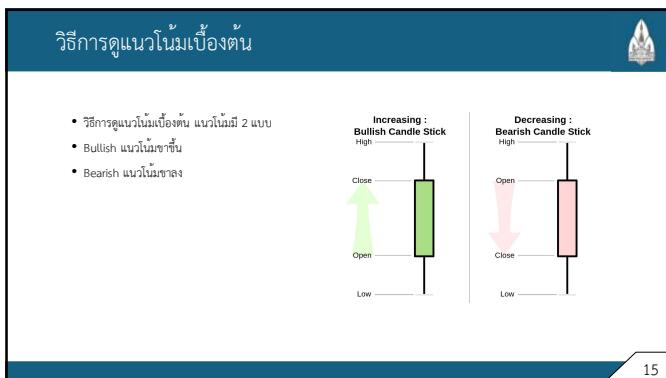
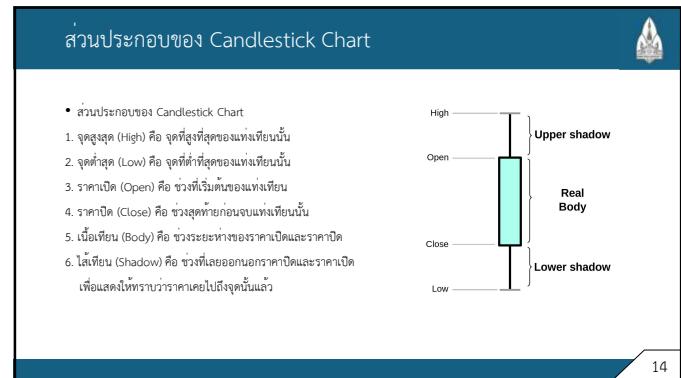
- `ax.xaxis.set_visible(False)` ปิดการแสดงผล y บนกราฟ
- `ax.spines[["left", "top", "right"]].set_visible(False)`
- ปิดกรอบและเส้นขอบของกราฟทั้งสามด้าน ด้านขวา
- `ax.margins(y=0.1)`
- เพิ่มพื้นที่ของ y บนกราฟและดัดแปลงของกราฟ 10% ของความสูงทั้งหมดของกราฟเพื่อให้กราฟอยู่ต่ำกว่ากรอบและทำให้กราฟมองเห็นง่ายขึ้น
- `plt.show()` แสดงกราฟที่สร้างขึ้นตามการกำหนดค่าดังที่ระบุ Matplotlib

11

សេចក្តី



12



### Import packet

```

• import plotly.graph_objects as go
• import pandas as pd
• from datetime import datetime

```

16

### Download data to memory

```

df = pd.read_csv("https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-apple.csv")
df

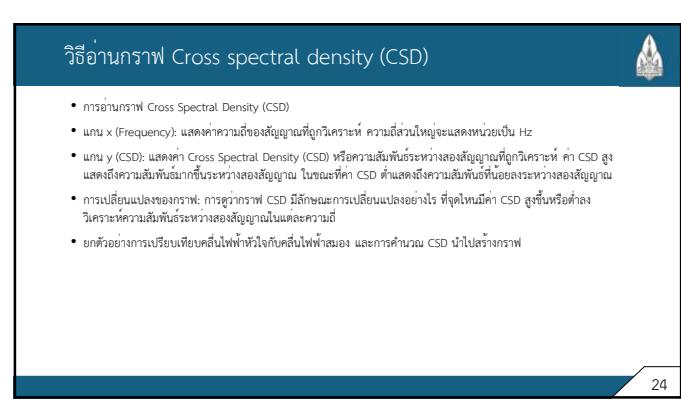
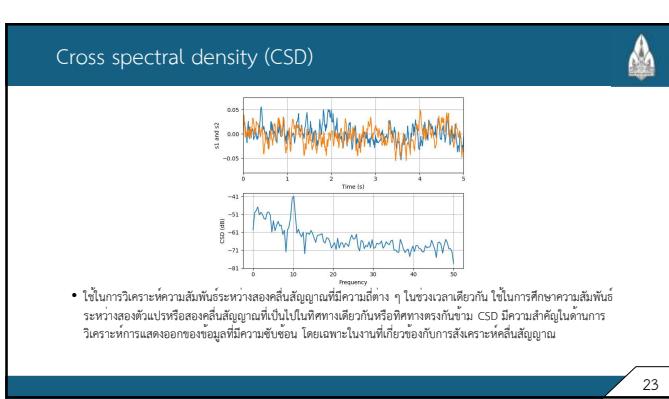
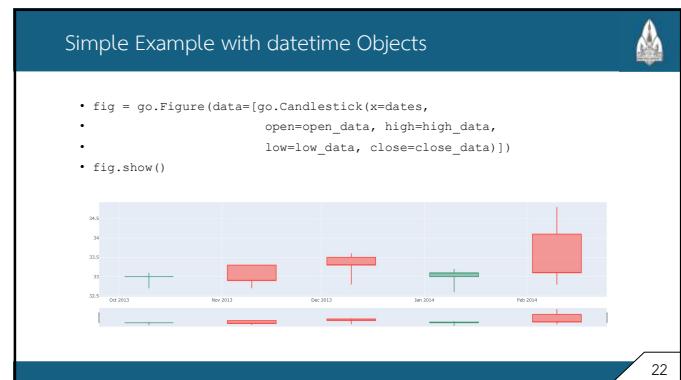
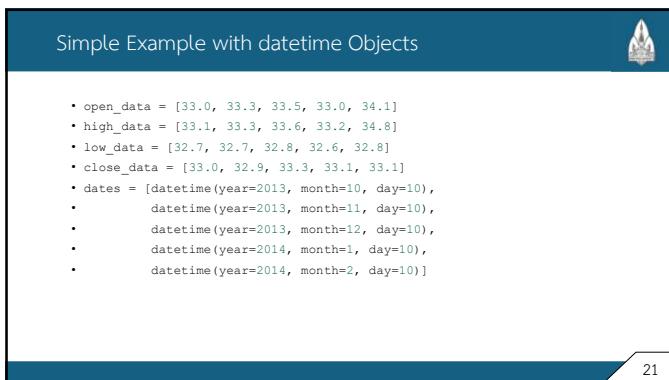
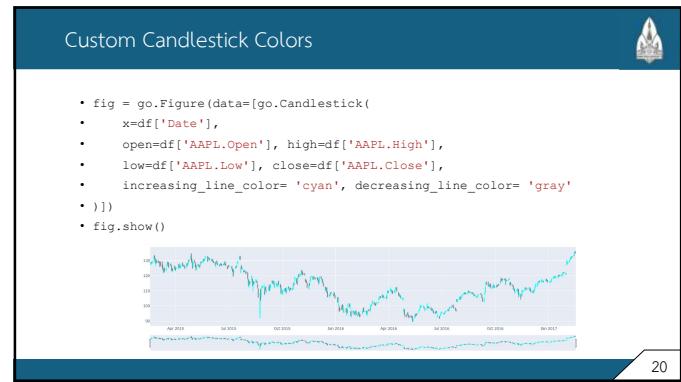
```

| Date | AAPL_Open  | AAPL_High  | AAPL_Low   | AAPL_Close | AAPL_Volume | AAPL_Adjusted | ds         | msg        | up         | direction  |            |
|------|------------|------------|------------|------------|-------------|---------------|------------|------------|------------|------------|------------|
| 0    | 2015-02-17 | 127.499999 | 128.880005 | 126.999999 | 127.850002  | 6315400       | 122.950524 | 108.741052 | 117.276767 | 129.116281 | Increasing |
| 1    | 2015-02-18 | 127.499999 | 128.779999 | 127.499997 | 128.720001  | 44891700      | 123.309690 | 107.842423 | 118.044033 | 130.082024 | Increasing |
| 2    | 2015-02-19 | 128.479999 | 129.029999 | 128.300002 | 128.49997   | 3736400       | 123.501961 | 108.894245 | 119.889917 | 130.884039 | Decreasing |
| 3    | 2015-02-20 | 128.479999 | 129.500000 | 128.000002 | 129.000001  | 4094400       | 124.510194 | 109.785649 | 120.730502 | 131.741551 | Increasing |
| 4    | 2015-02-23 | 130.200004 | 130.000000 | 129.600001 | 130.000000  | 7074100       | 127.897074 | 119.372516 | 131.707014 | 133.061037 | Increasing |
| 501  | 2017-02-10 | 132.400007 | 132.840002 | 132.000000 | 132.119995  | 2086500       | 132.119995 | 114.493054 | 124.493054 | 134.503238 | Decreasing |
| 502  | 2017-02-13 | 133.000001 | 133.830007 | 132.700005 | 132.380001  | 2303400       | 132.380001 | 114.632798 | 125.10514  | 133.580334 | Increasing |
| 503  | 2017-02-14 | 133.470001 | 135.099996 | 133.200002 | 135.000004  | 2381100       | 135.000004 | 115.175718 | 125.555498 | 133.751280 | Increasing |
| 504  | 2017-02-15 | 135.500001 | 136.270004 | 134.499995 | 135.500005  | 3550100       | 135.500005 | 115.545035 | 126.723498 | 137.401963 | Decreasing |
| 505  | 2017-02-16 | 135.469991 | 135.899994 | 134.699996 | 135.350000  | 2218000       | 135.350000 | 114.203299 | 127.004333 | 138.805386 | Decreasing |

505 rows × 11 columns

17





### Import

```

• import numpy as np
• from scipy import signal
• import matplotlib.pyplot as plt

```

25

### จำลองข้อมูลเวลาในรูปแบบความถี่

```

• num_samples = 1000
• time = np.linspace(0, 1, num_samples)

• ใช้ array ที่ชื่อเป็น time ที่ประกอบด้วยค่าเวลาตั้งแต่ 0 ถึง 1 ที่มีขนาดตั้งแต่ 1000 ตัวอย่างโดย np.linspace() จะแบ่งค่าเวลาทั้งหมด 0 ถึง 1 ออกเป็นช่วงๆๆ ที่มีจำนวนมากๆ กัน ทั้งหมด 1000 ช่วง

```

26

25

### จำลองข้อมูลคลื่นไฟฟ้าหัวใจ ECG

```

• p_wave = np.sin(2 * np.pi * 2 * time)
qrs_complex =
 + 0.2 * np.sin(2 * np.pi * 10 * time)
 + 0.3 * np.sin(2 * np.pi * 20 * time)
 + 0.1 * np.sin(2 * np.pi * 30 * time)
)
t_wave = np.sin(2 * np.pi * 1 * time)
ecg_signal = p_wave + qrs_complex + t_wave

• np.sin() คือฟังก์ชัน sine ที่ใช้สำหรับคำนวณค่า sine ของผลพิธีจากการคำนวณ 0
• np.pi คือค่าของ π (พาย) มีค่าประมาณเท่ากับ 3.14159 หรือ 22/7

```

27

27

### จำลองข้อมูลคลื่นไฟฟ้าหัวใจ ECG

```

• p_wave จัดอันดับที่เกิดขึ้นเมื่อหัวใจยังไม่หัวใจต้องหัวใจ (atria) มีความถี่ 2 Hz
• qrs_complex จัดอันดับที่เกิดขึ้นเมื่อสัญญาณไฟฟ้าเดินไปยังห้องหัวใจ (ventricles) จากการสอนคลื่น 3 ความถี่ 10 Hz, 20 Hz, 30 Hz ตามตัวเรานั้นก็หาดู
• t_wave จัดอันดับที่เกิดขึ้นเมื่อหัวใจเตรียมให้กลับไปสู่สถานะพัก (resting state) โดยเป็นคลื่น sine wave ที่มีความถี่ 1 Hz
• ecg_signal รวมค่าคลื่นทั้ง 3 จัดอันดับเมื่อสัญญาณทางไฟฟ้าที่เกิดขึ้นในหัวใจตลอดเวลาในรูปแบบของกราฟ ECG

```

28

28

### จำลองข้อมูลคลื่นไฟฟ้าสมอง EEG ในสภาวะ BETA

```

• brain_signal = np.cos(2 * np.pi * 20 * time)
random_movement = np.random.normal(loc=0, scale=0.1, size=num_samples)
brain_signal += random_movement

• np.cos() คือฟังก์ชัน cosine ที่ใช้สำหรับคำนวณค่า cosine ของผลลัพธ์จากการคำนวณใน () ดีอีคืนที่มีความถี่ 20 Hz
• np.random.normal(loc=0, scale=0.1, size=num_samples) คือการสร้างการกระจายแบบ normal distribution ที่มีค่าเฉลี่ยเท่ากับ 0 และส่วนเบี่ยงเบนมาตรฐานเท่ากับ 0.1 คือขนาดของข้อมูลที่คำนวณ
• brain_signal += random_movement คือการนำค่า random_movement ล้วนๆ brain_signal เพิ่มเข้าไปใช้ในการจำลองการพัฒนาของสมองของซึ่ง random_movement ที่บันทึกไว้ไปเป็นค่าที่ถูกนำมาจากการกระจาย normal distribution ที่ได้คืนที่ออกแบบมาเพื่อรวมชาติด้วยของจริงมากที่สุด

```

29

29

### คำนวณ CSD ระหว่างคลื่นไฟฟ้าหัวใจและสมอง

```

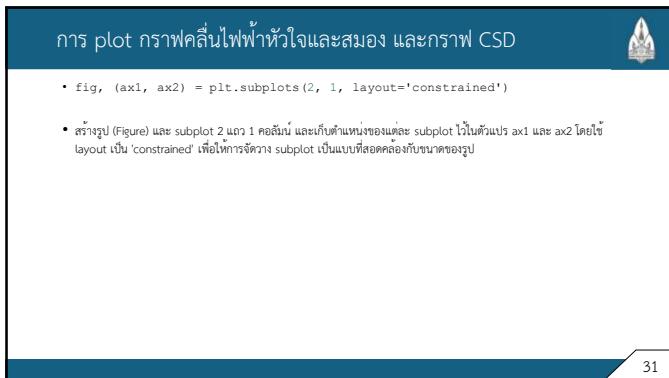
• frequencies, csd = signal.csd(ecg_signal, brain_signal, fs=1.0, nperseg=100)

• คำนวณความสัมพันธ์ของสเปกตรัมระหว่างสัญญาณคลื่นไฟฟ้าหัวใจ (ecg_signal) และสัญญาณทางไฟฟ้าสมอง (brain_signal) โดยใช้ Cross Spectral Density (CSD)
• signal.csd() พัฒนาขึ้นมาในการคำนวณ Cross Spectral Density (CSD) ระหว่างสองสัญญาณ
• ecg_signal, brain_signal: ตัวบ่งชี้สัญญาณที่ต้องนำมาคำนวณ CSD
• fs=1.0: ค่าอัตราของสัญญาณ (sampling frequency) ต้องให้มีค่าเป็น 1.0 เมื่อจะนำรากที่สองมาใช้ในการคำนวณ
• nperseg=100: number of points per segment ให้ในการคำนวณ CSD ในแม่สอดครั้ง ก้าวเดียวให้มีค่าเป็น 100 จุด
• ผลลัพธ์ที่ได้ frequencies เป็นค่าความถี่และ csd เป็นค่า Cross Spectral Density ระหว่างสองสัญญาณ

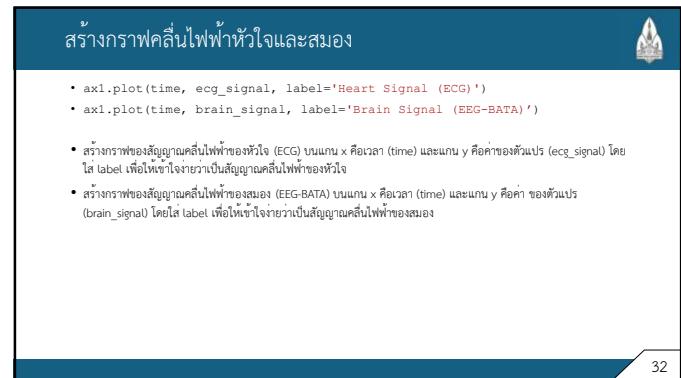
```

30

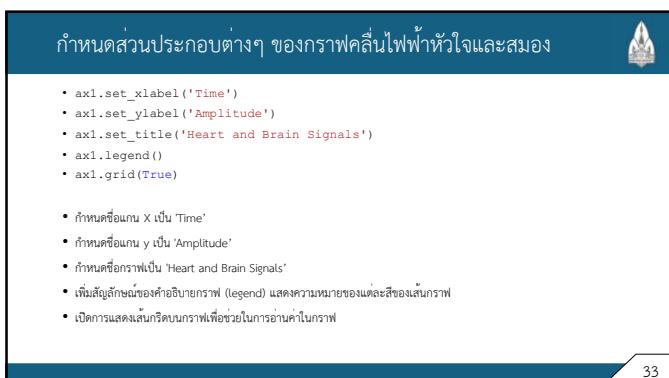
30



31



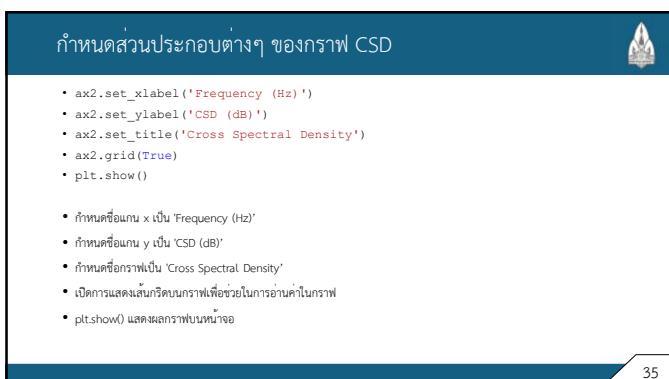
32



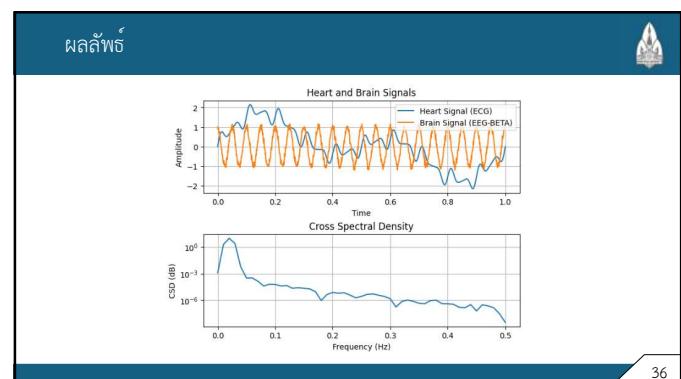
33



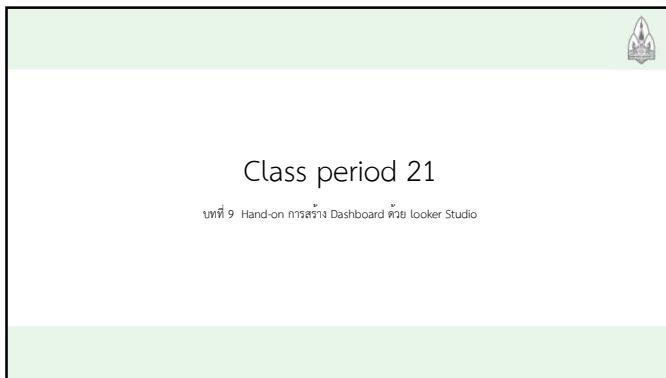
34



35



36



1

Wongnai Dashboard

https://lookerstudio.google.com/reporting/8266ba0e-f257-4376-8fba-057a058a3284

2

ໂທລາດຂອ່ມງວດ

- <https://drive.google.com/drive/folders/1AztYMSDzIwvDHfSI0T51VSCYVRyQ2Z3>
- ให้ “เปิด link google drive ข้างบนและเปิดไฟล์ 4 ไฟล์ดัง Open with > Google Sheets

| File                      | Date         | Actions |
|---------------------------|--------------|---------|
| random_w_restaurant.csv   | Apr 12, 2021 | ⋮       |
| random_search_request.csv | Apr 12, 2021 | ⋮       |
| random_search_click.csv   | Apr 12, 2021 | ⋮       |
| 3Data_clean_new.csv       | Apr 10, 2021 | ⋮       |

3

ทำสำเนาไฟล์ Google Sheets ໄວ້ໃຈ google drive ຕ້ອງ

กด File > Make a copy

random\_search\_click

4

ทำสำเนาไฟล์ Google Sheets (ต่อ)

ต้องซื้อไฟล์และเลือกที่อยู่ของไฟล์ที่จะทำสำเนาไว้ใน google drive ขอทั่วไป ให้ click ที่ My Drive และสร้างไฟล์ครั้งใหม่ ที่จะใช้เป็นสำเนา

Copy document

Name: copy\_of\_random\_search\_click

Folder:

My Drive

Share with the same people

Copy comments

Version history

Comments

Review

Cancel Make a copy

4

ทำสำเนาไฟล์ Google Sheets (ต่อ)

ให้ Click ที่ All location และเลื่อนมาสักปีชั่วโมง กดที่ลูกศรขวาของ My Drive เพื่อเข้าไปสร้างไฟล์ครั้งใหม่

Suggested Started All locations

My drive

shared drives

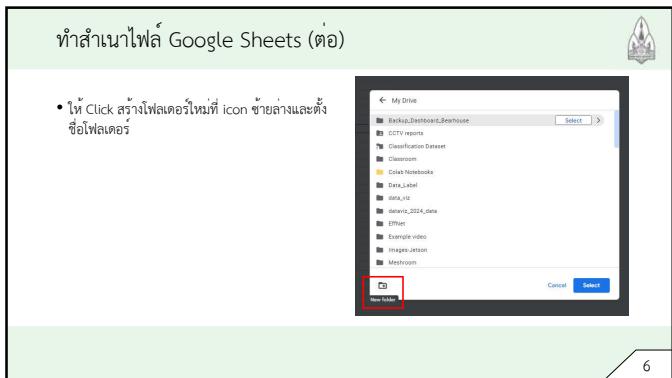
Computers

shared with me

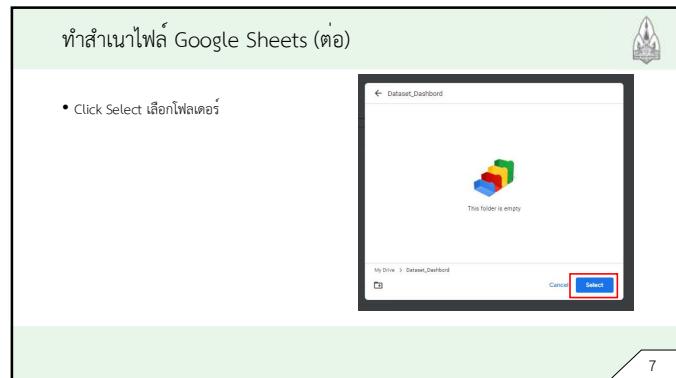
View My Drive

5

1



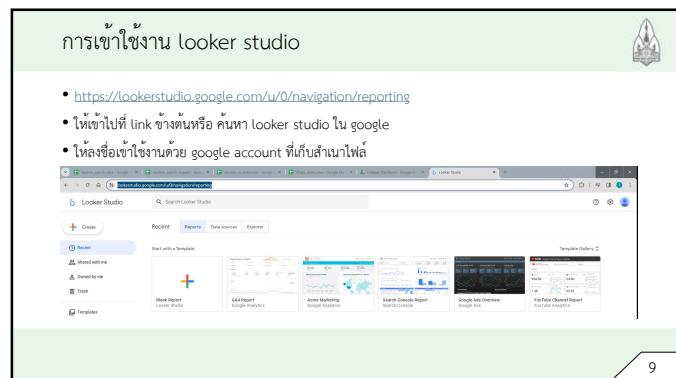
7



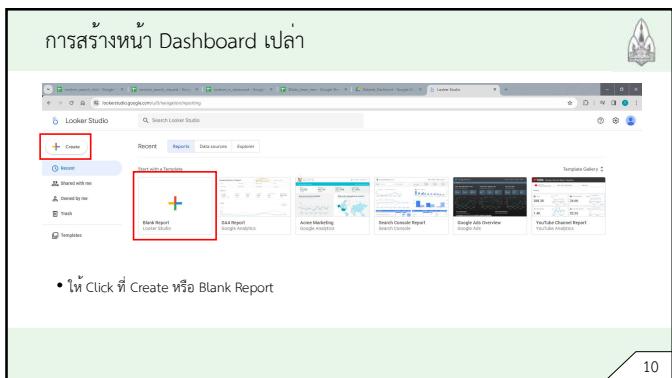
8



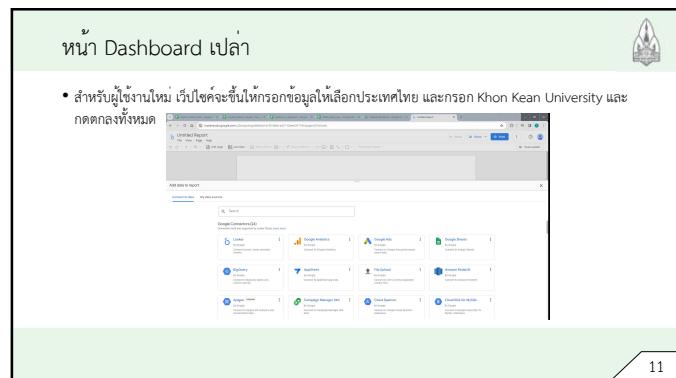
9



10



11



12

2

### การเชื่อมต่อข้อมูลกับ Dashboard

- กดเลือก Google Sheets

12

13

### การเชื่อมต่อข้อมูลกับ Dashboard (ต่อ)

- สำหรับผู้ใช้งานใหม่ เนื่องจากจะต้องให้คุณญาติการเข้าถึงข้อมูลและเชื่อมต่อ google drive

13

14

### โหลดข้อมูลเข้าหน้า Dashboard

- เลือก OWNED BY ME > random\_w\_restaurant > กด Add

14

15

### หลังจากโหลดข้อมูลเข้าหน้า Dashboard

15

16

### การเพิ่มข้อมูล

- กด Add data และทำการเลือก Google Sheets > OWNED BY ME > random\_search\_Click > กด Add

16

17

### การรวมข้อมูล

- คลิก BLEND DATA

17

18

การรวมข้อมูล (ต่อ)

- Click Join another table > เลือก random\_search\_Click

18

19

การรวมข้อมูล (ต่อ)

- ctrl Dimensions ให้เลือก primary key ของ 2 ตารางที่รวมข้อมูล

19

20

การรวมข้อมูล (ต่อ)

- สามารถเลือกข้อมูลที่ต้องการแสดงใน Dashboard จากทั้ง 2 ไฟล์ต่อได้ โดยการเลือกได้ต่อใน Dimensions

20

21

การรวมข้อมูล (ต่อ)

- Click Configure join เพื่อกำหนดประเภทการรวมและเลือก primary key

21

22

การรวมข้อมูล (ต่อ)

- กำหนดประเภทการรวมและเลือก primary key

22

23

หลังจากการรวมข้อมูล

23

24

การเลือกข้อมูลให้แสดงในตาราง

- ແລບຕົວອ່ນດ້ານຂ້າຍຂອງหน້າ Dashboard ສາມາຮອກຄູ່ຂອງສູດທີ່ຕ້ອງຈາກແລກໄປໃນ Dimension

24

25

การสร้างกราฟໃນ looker studio

- ໃຫ້ລັບປັບທີ່ຫຼັງຈາກແລກສ້າງหน້າປໍາກໍ່ນຳໃໝ່ ແລະ ໃຫ້ໄລດ້ຂອຸມືດ 3Data\_clean\_new

25

26

หลັງຈາກໂລດຂໍຂອງມູນເຂົ້າໜ້າ Dashboard

- ການລັບດາຮາງ ສາມາຮອກ click ທີ່ຕ້າງໆແລກ Del ໃນ keyboard ໄດ້ເລີຍ

26

27

การຕັ້ງຊື່ໜ້າ Dashboard

- Click ຫ້ອງຈຳ logo ແລະ ອິນພໍ່ຫ້ອງທີ່ຕ່ອງການ

27

28

ການກຳໜາດຂໍາດຂອງໜ້າ Dashboard

- ໃນແລບຕົວນີ້ໃຫ້ click Theme and layout

28

29

ການກຳໜາດຂໍາດຂອງໜ້າ Dashboard (ຕອ)

- ແລບຕົວອ່ນດ້ານຂ້າຍໃຫ້ LAYOUT >ການອໜາດໃນ Canvas Size ໄດ້ສາມາຮອດ dropdown ເລືອ setting ທີ່ມີໃຫ້ ຂໍສ້າງ

29

30

สร้างกราฟวงกลม

- Click ที่ Add a chart > dropdown เลือก Pie chart

30

31

สร้างกราฟวงกลม (ต่อ)

- เลือกพื้นที่ที่ต้องการวาง

31

32

สร้างกราฟวงกลม (ต่อ)

- ແບ່ນເຄື່ອນໄຫວຢ່າຍເລືອກ category\_name ໃສ່ Dimension ເພື່ອການການຄູປະກທ້ານ

32

33

กำหนดชื่อໃຫຍ່ມູນຄົນແນບຕົວຢ່າງ

33

34

การปรับแต่งกราฟวงกลม

34

35

การสร้างขอความ

- Click ແບ່ນເຄື່ອນໄຫວ Text > ຈັດວາງຂອງขอความແລະຮີມທີ່ຕ້ອງການ > ແບ່ນເຄື່ອນໄຫວດ້ານຢ່າງ Text Properties ສາມກຳປັບແທຂອງขอความທີ່ຕ້ອງການໄດ້

35

36

**การสร้างของเปล่าเพื่อตกแต่ง**

- Click เมนูริชชั่น Shape (ใน dropdown สามารถเลือกรูปแบบได้) > จัดวางตามพื้นที่ที่ต้องการ > และเรื่อยมีอีกขั้น Rectangle Properties สามารถปรับแต่งตามที่ต้องการได้

36

37

**สร้างกราฟแท่ง 10 อันดับร้านตามจำนวนรีวิว**

- Click ที่ Add a chart > dropdown เลือก Bar chart

37

38

**สร้างกราฟแท่ง 10 อันดับร้านตามจำนวนรีวิว (ต่อ)**

- เลือกพื้นที่ที่ต้องการวาง

38

39

**สร้างกราฟแท่ง 10 อันดับร้านตามจำนวนรีวิว (ต่อ)**

- แผนเครื่องมือค้นหาชี้ให้เลือก number\_of\_reviews ให้ Data Range Dimension และเลือก name ให้ Dimension เพื่อทำกราฟแท่ง 10 อันดับร้านตามจำนวนรีวิว

39

40

**การปรับสัดส่วนของกราฟ**

- นำมาส์กไปที่สันแนน Y หรือบริเวณสีแดงและกดคลาย ข้าย-ขวา เพื่อปรับแต่งความกว้างของแมบที่แกน

40

41

**ปรับแต่งกราฟแท่ง**

- กำหนดค่าชื่อให้บุลใน SETUP > ปรับแต่งกราฟแท่งใน STYLE

41

42

เพิ่มช่องข้อความชื่อกราฟแท่ง

- ปรับแต่งสีสันให้สวยงามตามใจ

42

43

การสร้างข้อความบอกจำนวน Total หรือจำนวนรวม

- Click ที่ Add a chart > dropdown เลือก scorecard

43

44

การสร้างข้อความบอกจำนวน Total หรือจำนวนรวม (ต่อ)

- เลือกพื้นที่ที่ต้องการวาง กากบาทซึ่ด้วยเมาส์แลบและปั๊บลงใน STYLE ตามที่ต้องการ

44

45

สร้างกราฟแท่งจำนวนโฆษณา

- Click ที่ Add a chart > dropdown เลือก Column chart

45

46

สร้างกราฟแท่งจำนวนโฆษณา (ต่อ)

- เลือกพื้นที่ที่ต้องการวาง

46

47

สร้างกราฟแท่งจำนวนโฆษณา (ต่อ)

- แบบเครื่องมือค้นหาให้เลือก abs ให้ Dimension เพื่อทำกราฟแท่งจำนวนโฆษณา

47

48

**การเปลี่ยนประเภทของข้อมูลตัวเลข**

- ในແນບຕົວໜີມອັກ icon ກຳນົດຂໍ້ຕົວແປຣ NO ເພີ່ເລີນເປັນຈຳນວນ > Aggregation ໃຫ້ເລືອເປັນ count (ຈຳນວນນັ້ນ)

48

49

**สร้างกราฟแท่งຈຳນວນໂຄສະນາ (ຕອ)**

- ເລືອງຈຳນວນຂໍ້ຕົວແປຣແລ້ວປັບແທກຮາບໃນ STYLE ຕາມທີ່ຕ້ອງການ

49

50

**สร้างกราฟ Bubble map (Google map)**

- Click ທີ່ Add a chart > dropdown ເລືອກ Bubble map

50

51

**สร้างกราฟ Bubble map (Google map) (ຕອ)**

- ເລືອກແລ້ວຈັດຮັບເປັນຫຼື່ງທີ່ຕ້ອງການ

51

52

**การรวมข้อมูล 2 ຕັ້ງແປຣ(ຄອລັມນ໌)**

- ໃຊ້ keyword 'lat' ສັນກາໃນອັນຫັກຂໍ້ມູນ ຮະເຫັນຕັ້ງແປຣ lat ອີເລີດຕິຈຸດ ຕັ້ງແປຣ long ອີເລີດຕິຈຸດ ໃນການໃຊ້ຮັນຈີເພື່ອຮັບຕົ້ນທີ່ຈະໄວ້ ຈຳເປັນໂຄລິຍົງ ລະດົບລົງ ລອຈິງຈຸດ ແລ້ວປັບຂໍ້ມູນຂອງເດືອຍ
- ດັ່ງນັ້ນຕ້ອງຮັບຕັ້ງແປຣໃນໆກ່ຽວຂ້ອງມູນຮົມຄອລັມນ໌ທີ່ມີແປຣ lat ແລ້ວຕັ້ມນປຣ long ໃໃນກໍບໍ່ຂໍ້ມູນ ‘ລະດົບລົງ ລອຈິງຈຸດ’

52

53

**การรวมข้อมูล 2 ຕັ້ງແປຣ(ຄອລັມນ໌) (ຕອ)**

- Click search\_timestamp\_n...

53

54

การรวมข้อมูล 2 ตัวแปร(คอลัมน์) (ต่อ)

- Click ADD FIELD

54

55

การรวมข้อมูล 2 ตัวแปร(คอลัมน์) (ต่อ)

- Data Type > dropdown เลือก Geo > Latitude, Longitude

55

56

การรวมข้อมูล 2 ตัวแปร(คอลัมน์) (ต่อ)

- ต้องขอและเขียน code CONCAT(lat, "", long) เพื่อรวมข้อมูลทั้ง 2 ตัวแปรเข้าด้วยกัน กด APPLY หลังจากโหลดเสร็จให้กดที่นี่ที่ ว่างตรงหน้าไปต่อ

56

57

สร้างกราฟ Bubble map (Google map) (ต่อ)

- แผนเครื่องมือคำนับข้อมูลเลือก name ใส่ Location และ Color dimension เลือก search\_timestamp\_new ใส่ Size เพื่อ Google map ที่อยู่รำลอกหาโดยขนาดของ Bubble ศึกษาแนบการค้นหา

57

58

สร้างกราฟ Bubble map (Google map) (ต่อ)

- หลังจากเลือกและโหลดเสร็จ จะเข้ามาเจ้งเตือนเรื่องเก็บจำนวนข้อมูลและจำนวนเขตสี Click OK, got it

58

59

ปรับแต่งกราฟ Bubble map (Google map)

- กำหนดชื่อตัวแปรใหม่เพื่อใช้เข้าใจง่าย และจำนวนการค้นหาให้กำหนด Aggregation เป็น Count

59

60

10

**ปรับแต่งกราฟ Bubble map (Google map) (ต่อ)**

- เพิ่มขอบเขตความเรื้อรังและปรับแต่งกราฟใน STYLE ตามที่ต้องการ

61

**สร้างตารางแสดงรายละเอียดข้อมูล**

- คลิกที่ Add a chart > dropdown เลือก Table

62

**สร้างตารางแสดงรายละเอียดข้อมูล (ต่อ)**

- เลือกและจัดระเบียบทันทีที่ต้องการว่าจะ

63

**สร้างตารางแสดงรายละเอียดข้อมูล (ต่อ)**

- แตะเครื่องมือค้นหาให้เลือกแบบป๊อปอัปท์ที่ต้องการให้แสดงในตาราง เพื่อทำตารางดูรายละเอียดของร้าน

64

**สร้างตารางแสดงรายละเอียดข้อมูล (ต่อ)**

- กำหนดค่าอย่างเดลล์แล้วปรับเปลี่ยนเพื่อความเข้าใจง่าย

65

**ปรับแต่งตาราง**

- จัดเรียงพื้นที่ในตาราง ปรับแต่งตารางใน STYLE ตามที่ต้องการและเพิ่มช่องข้อมูลเชื่อมตาราง

66

การสร้างตัวกรอง dropdown

- Click ที่ Add a control > dropdown เลือก Drop-down list

66

67

การสร้างตัวกรอง dropdown (ต่อ)

- เลือกฟันธงที่ต้องการวิเคราะห์

67

68

การสร้างตัวกรอง dropdown (ต่อ)

- ແນບเครื่องมือมาช่วยให้เลือก city\_name ให้ Control field เพื่อทำตัวกรองตามชื่อจังหวัด

68

69

การสร้างตัวกรอง dropdown (ต่อ)

- กำหนดชื่อให้ด้วยเป็น ชื่อจังหวัด และกำหนด Data Type เป็น City

69

70

การสร้างตัวกรอง dropdown (ต่อ)

- ปรับแต่งตัวกรอกให้สอดคล้องตามต้องการ

70

71

การทดสอบใบงานตัวกรอก

- Click View ด้านข้างบนเพื่อเปลี่ยนหน้าจอเป็นรูปใบงาน

71

72

12

**การทดสอบใช้งานตัวกรอก (ต่อ)**

- Click เลือกจังหวัดในตัวกรอกเพื่อทดสอบการใช้งาน ตรวจสอบการอัปเดตของกราฟและตาราง

72

73

**การโหลดรูปภาพเข้าหน้า Dashboard**

- Click Image > dropdown

73

74

**การโหลดรูปภาพเข้าหน้า Dashboard (ต่อ)**

- Upload from computer คือการอัปโหลดรูปภาพเพื่อตั้งเป็นรูปของเรารอง ให้นักศึกษาบันทึกรูป Logo Wongnai จาก <https://www.wongnai.com/brand-assets>

74

75

**การโหลดรูปภาพเข้าหน้า Dashboard (ต่อ)**

- แตกไฟล์ LOGO\_Wongnai.zip > Click Upload from computer > เลือกรูปภาพ

75

76

**การโหลดรูปภาพเข้าหน้า Dashboard (ต่อ)**

- เลือกพื้นที่ที่ต้องการวางรูปภาพ

76

77

**การโหลดรูปภาพเข้าหน้า Dashboard (ต่อ)**

- By URL คือการอัปโหลดรูปโดยใช้ URL ของรูปในหน้าเว็บ คลิกขวาที่รูปที่เราต้องการ > Open image in new tab

77

78

13

การໂຄດຮຽນປາກເຂົ້າຫ້າ Dashboard (ຕອ)

- Copy URL ຂອງຮູບ

78

79

การໂຄດຮຽນປາກເຂົ້າຫ້າ Dashboard (ຕອ)

- Image > dropdown > Click By URL

79

80

การໂຄດຮຽນປາກເຂົ້າຫ້າ Dashboard (ຕອ)

- ວາງ URL ຈີ່ Copy ມາ > กົດ Insert

80

81

การໂຄດຮຽນປາກເຂົ້າຫ້າ Dashboard (ຕອ)

- ເລືອກຫັ້ນທີ່ຕ້ອງການຈຸດປະກິດ

81

82

ຕັດແຕງ Dashboard ຕາມຕ່ອງການ

82

83

ການແຊ່ງຫາ Dashboard

- Click Share ດໍາລົງບັນຂອງຫ້າ Dashboard

83

84

**การแชร์หน้า Dashboard (ต่อ)**

- สามารถแชร์แบบลิงก์ Gmail หรือแบบ Copy link และสามารถกรอกกำหนดการสิทธิ์การเข้าถึงให้เหมาะสมได้

84

**การดาวน์โหลดหน้า Dashboard**

- Click File > Download as > PDF

85

**การดาวน์โหลดหน้า Dashboard (ต่อ)**

- สามารถกรอกกำหนดการดาวน์โหลด ไม่ใช้แสดงลิสต์ผู้รับ, และลิงก์สำหรับดาวน์โหลด Dashboard ในเว็บ ตั้งแต่ที่ลืม

86

**การดาวน์โหลดหน้า Dashboard (ต่อ)**

- ต้องตั้งรหัสผ่านในการเปิดไฟล์ PDF จะถูกห้ามรหัส

87

**Locker studio**

88

**Ethical framework in data project**

<https://towardsdatascience.com/the-importance-of-ethics-centricity-in-data-projects-ea2a60e6b67>

**Ethical Guidance Framework**

Guidelines to inform prevention and mitigation

**Safety**  
Can the model harm individuals or the organization over time?

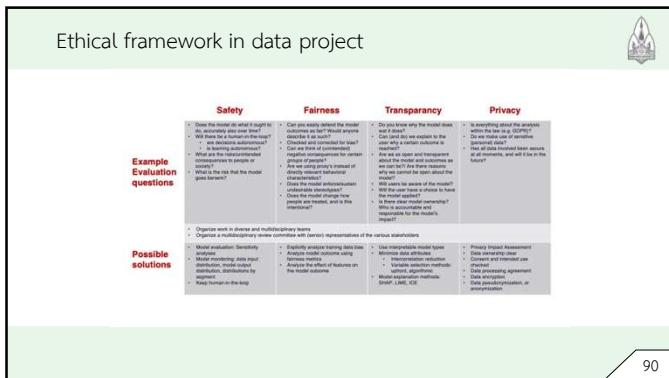
**Privacy**  
Does the model respect individual rights and comply with privacy laws and regulations?

**Fairness**  
Does the model result in a continuation of existing biases of certain people?

**Transparency**  
Can the model's decision-making logic be interpreted?

89

15



91

90