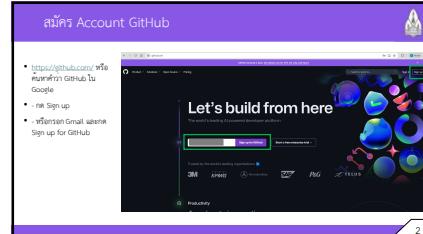
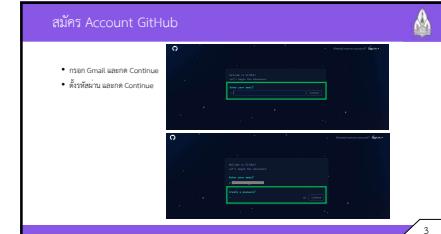




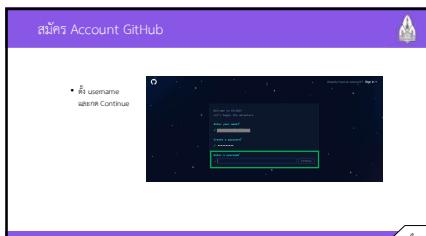
1



2



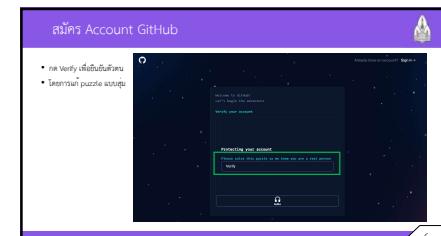
3



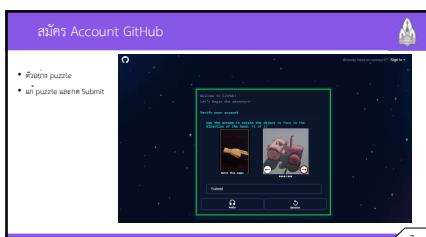
4



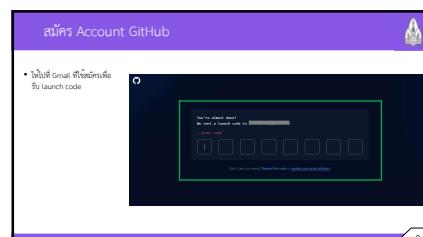
5



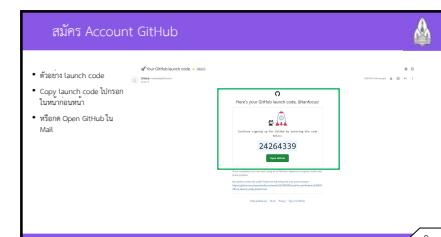
6



7



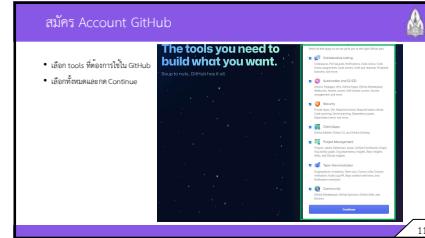
8



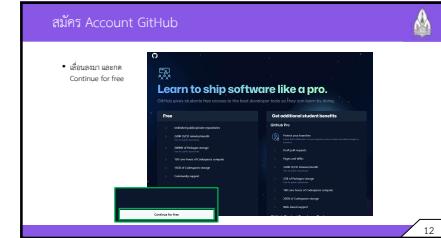
9



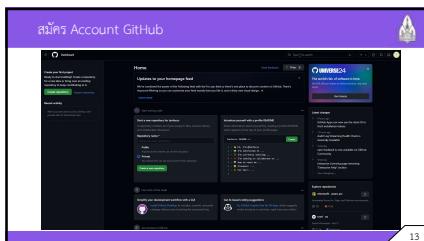
10



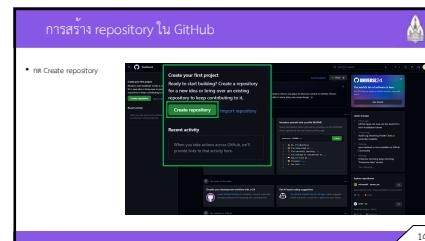
11



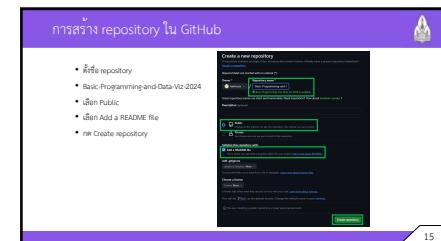
12



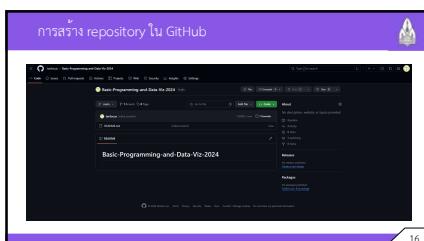
13



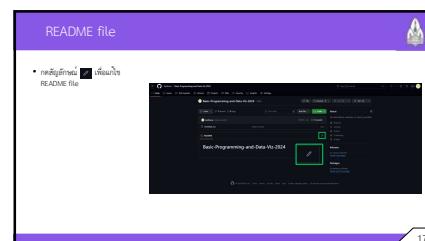
14



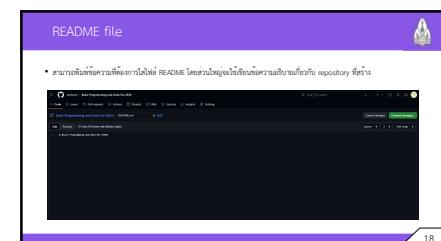
15



16



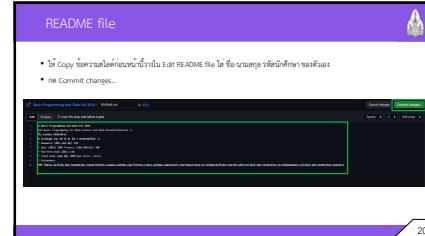
17



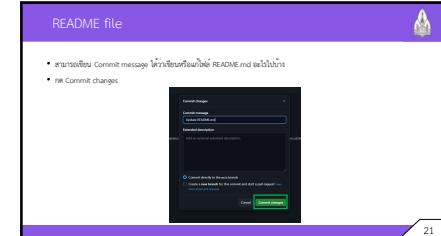
18



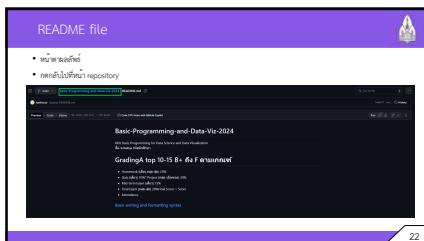
19



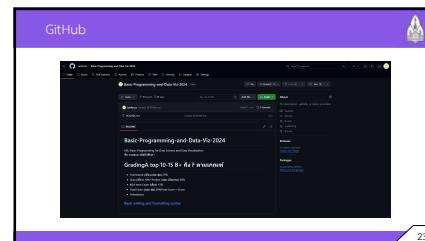
20



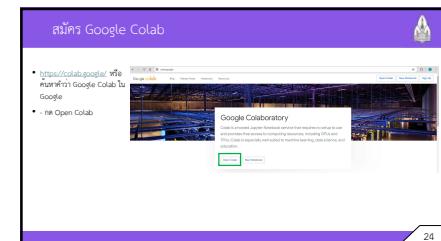
21



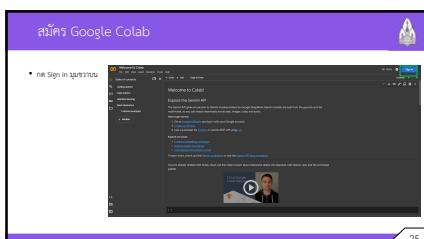
22



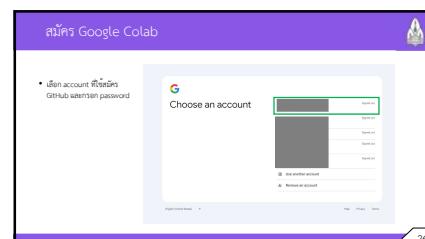
23



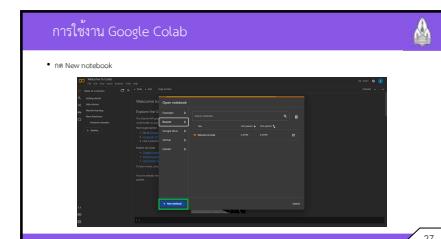
24



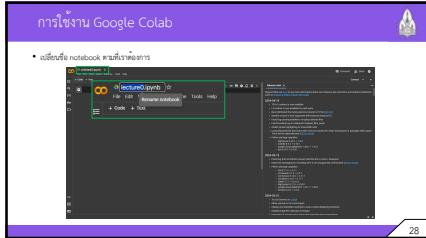
25



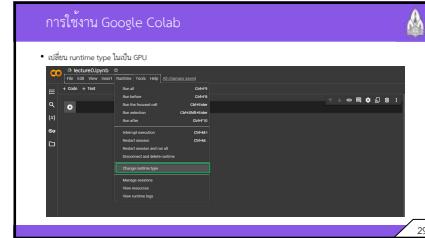
26



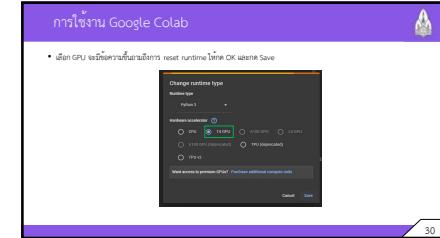
27



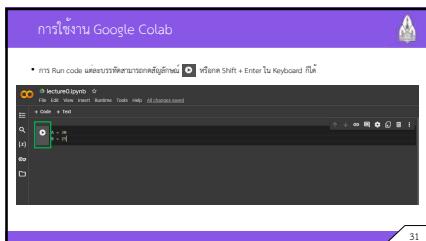
28



29



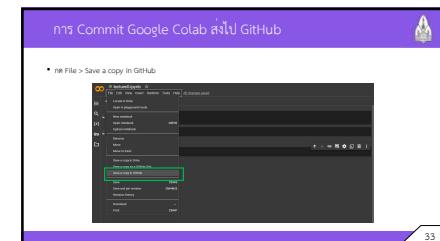
30



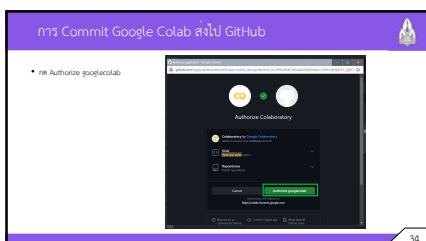
31



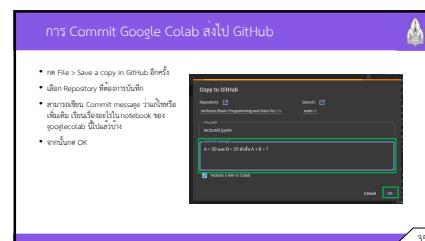
32



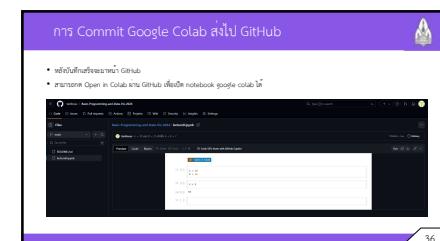
33



34



35



36



37

**Class period 1**

บทที่ 2 ฟังก์ชันและโครงสร้างข้อมูลของภาษา Python

Basic python  
python101

1

**Variables คืออะไร**

- Variables คือตัวแปรใช้เก็บข้อมูล ยุคสมัยที่ก้าวหน้าไป 譬喻 pi = 3.14159265359 ในสิ่งที่เป็นตัวอักษรให้เป็น pi ที่ก้าวหน้าไม่ลงมือทัน
- หลักการคือต้องคำนึงให้ชัดเจน
  - ตั้งชื่อให้ดี
  - ภาษาจัดเรียง
  - ให้ตัวเลขไม่คล้ายกับตัวอักษร
  - ห้ามบันทึก
  - ห้ามตั้งชื่อตัวแปรใช้เพื่อชื่อค่าๆ (def, for, range, etc.)

2

**ชนิดของตัวแปร**

- Int : ตัวเลขจำนวนเต็ม เช่น a = 10
- Float : จำนวนจริง (浮点数) เช่น b = 10.0
- ตัวอักษร (char (character)) ข้อความ (text หรือ string) เช่น c = "python"
- ตัวเลขที่มีจุด przecink ไม่ว่าจะมากน้อย ก็จะ คูณ หาก กับตัวเลข เช่น d = '10.'

3

**variable casting (การเปลี่ยนชนิดของข้อมูล)**

- กรณีต้องใช้ตัวอักษรในการพิมพ์ เช่น เลขมาบรรยายให้นำมือถือของข้อมูลตามที่ได้ต้องการใช้งานให้โดยการ
- กำหนดให้ตัวอักษรที่อยู่ต่อตัวกันเป็นหนึ่งเดียว
- int(d)
- float(d)
- str(d)

4

**Operation (การเข้ากัน 2 ตัวมาทำอะไรกัน)**

(Operators +, -, \*, /, %)

- การบวก
- การลบ
- การคูณ
- การหาร
- การหารลงเศษ % หรือ餘數 modulus คือการหารเหลือเศษ

5

**คำสั่ง print แบบพิเศษ (การ format string)**

- แผนที่ 1 คำสั่ง print ที่สนุก เช่น print('ตัวมันเปร') ลิ๊ฟต์อยู่ชั้นในเมื่อกี้ ตัวมันเปร หรือ string
- แผนที่ 2 การพิมพ์ความที่ต้องการจากตัวมัน เป็นค่าการพิมพ์ ที่มา 'string' และให้ {} ให้ code เช่น print('%%' คือการรวมเลข เช่น 793 = ('%e3'))
- ln คือการเขียนร่องรอยใหม่
- \n ใช้ในการตัด text และแก้ code จะถูกลบไป

6

**DATA STRUCTURE (โครงสร้างข้อมูล)**

- List คือ ภารณาตัวมันเป็นหลายๆ ตัวมาเรียงกัน สามารถเรียกได้ 2 แบบ ดังนี้
- แบบที่ 1 list\_a = []
- แบบที่ 2 list\_a = list()
- list\_b = [1,5,V]
- ตัวต้องเป็นไปได้ มีความสำคัญ ตัวเดียว สอง สาม ฯลฯ 0,1,2,...
- อย่างเดียวมีความสำคัญของ list\_b ตัวที่ 1 ให้เขียน list\_b[1] คือ 5

7

**append() การเพิ่มสมาชิกเข้าใน list**

- คำสั่ง append คือตัวค่าตัวที่ต้องการเพิ่มน () สามารถเพิ่มสมาชิกเข้าไปใน list ที่ต้องการได้ เช่น
- list\_b.append('b') ผลลัพธ์ print(list\_b)
- [1, 5, V, 'b'] จะเห็นว่า 'b' ถูกเพิ่มเข้ามาใน list\_b
- ตัวต้องเป็นไปได้ ก็ต้องมีความสำคัญตัวต่อตัวของ list
- list\_b.pop() คือการลบ print(list\_b)
- [1, 5, V] จะเห็นว่า 'b' ถูกตัดออกจาก list\_b

8

**String > list of characters**

- คำสั่ง len() คือตัวค่าตรวจสอบความยาวของ list (จำนวนสมาชิก)
- String มีใน list เช่น
- t = 'python' is okay'
- len(t) จะหาที่ 14 นับตามจำนวนตัวอักษรและรวมเครื่องหมายรากชี้อยู่ด้วย

9

access a member of a list (list&string)

- សម្រាប់ការបញ្ចូនទីផ្សេងៗចាប់ពី 0 , -1 គឺជាការបញ្ចូនទាំងអស់
- t = "python is easy"
- t[1] មួយការបញ្ចូន យើងមានការបញ្ចូនទី 1 ដែលបានបញ្ជាក់
- t[-1] មួយការបញ្ចូន យើងមានការបញ្ចូនទាំងអស់

10

List slicing

- List slicing សម្រាប់ការបញ្ចូនទីផ្សេងៗ colon (a:b) -> [a:b]
- ការបញ្ចូនទីផ្សេងៗ [] និងវាតាមចំណាំ

11

ពាណិជ្ជកម្ម

- t = "python is easy"
- t[7:9]
- ចាប់បើរាយការណ៍ : អាមេរិកាសារាវាទីនិងអេតិម្យាលី បាន python
- ចាប់បើរាយការណ៍ : អាមេរិកាសារាវាទីនិងអេតិម្យាលី បាន python ឬមិនមែនមួយចំណាំ
- ពីលើក (7:9) មួយការបញ្ចូនទី 7 ទៅ 9 ប៉ុណ្ណោះនឹងត្រូវបានបញ្ចូនទី 8 ហើយ ទី 9 គឺអូអុប គឺ is

12

ពាណិជ្ជកម្ម

- t = 'python is easy'
- [::2] ឬចាប់បើនេះ : តារាងនៃ និងខ្លួន និងការបញ្ចូន step(ការការឡើង) ការបញ្ចូន t[::2] step=2
- សម្រាប់ការបញ្ចូន ដើម្បីពិនិត្យ
- list\_a = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- list\_a[::2]
- សម្រាប់ការបញ្ចូន [0, 2, 4, 6, 8]
- [::2::2] ដែលបានបញ្ចូនទី 2 និងការបញ្ចូន step เป็น 2
- សម្រាប់ការបញ្ចូន to step

13

ការបញ្ចូន list មានតែងៗ

- សារព័ន្ធនៃ list នាមទៅកីនដ៏ក្នុងការបិទ + គារគោរកទៅការកែតែ
- string និង string
- t = 'python is easy'
- t + ???" ទូទាត់ពីរ 'python is easy???"
- list\_b = [1, 5, 'v']
- នៅពេលរាយការណ៍ list បានគិតថាអំពីការបញ្ចូន string នៅក្នុង t + list\_b និងការបញ្ចូន t
- list\_a និង list\_b + list\_a មួយការបញ្ចូន [1, 5, 'v', ]]

14

split string ការបង្របៀន string ធម៌ស្ថិតិការណ៍ពីការបញ្ចូន

- សារព័ន្ធនៃការបិទ .split គារបែកចែកកិច្ចការពិនិត្យការបញ្ចូន()
- t = 'python is easy'
- t.split(' ') អាមេរិកាសារាវាទីនិងអេតិម្យាលី ឬមិនមែនមួយចំណាំ
- ['python', 'is', 'easy']
- time = '12:30:15'
- time.split(':') អាមេរិកាសារាវាទីនិងអេតិម្យាលី ឬមិនមែនមួយចំណាំ
- ['12', '30', '15']

15

វិធីរាយ string ក្នុង

- t = 'python is easy'
- t\_sp = t.split(' ')
- print(t\_sp) = ['python', 'is', 'easy']
- ពួកវារាយការណ៍ដើម្បីការបញ្ចូនទីផ្សេងៗនៃការបញ្ចូន t\_sp[0] ទៅ 'join' ទាំងអស់រាយការណ៍នៅលើលើលើ 0
- t\_join = ' '.join(t\_sp)
- print(t\_join) = python is easy

16

Homework class period 1

ការបញ្ចូនតាមលក្ខណៈបែកចែកនៃការបញ្ចូនទីផ្សេងៗនៃការបញ្ចូន t\_sp[0] ទៅ 'join' ទាំងអស់រាយការណ៍នៅលើលើលើ 0

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

17

**Class period 2**  
บทที่ 2 คุณลักษณะพิเศษของภาษา Python (ต่อ)  
Function\_Loop\_Condition

1

**บททวนภาษาที่แล้ว**

```

• name = "Anon"
• surname = "Afee"
• ID = "603021866-7"

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

```

2

**Function**

- ฟังก์ชันที่รับ input มาประมวลผลออกมาเป็น output
- ยกตัวอย่างเช่นยกตัวอย่างฟังก์ชันที่รับ x = y เป็น input คืนมาเป็น x หรือ f คือ function ที่รับ input ประมวลผลแล้วคืน output ออกมาที่ y

3

**สัญลักษณ์ที่จำเป็นต้องใช้ในการเขียน Program**

- backtick () => nn - กด n, alt + 9+6 (full keyboard with number)
- tild (-)
- curly bracket ({ })
- square bracket ([ ])

4

**Function template**

```

def something():
    do_something_with _Input_ to get _Output_
    return _Output_

```

การนับบรรทัด(indent) จะต้อง  
กด tab ก่อนเดินพิมพ์บรรทัดต่อไป  
จะถูก def เผื่อนย้ายขอบเขตของ  
โปรแกรม

function มีรูปแบบดังนี้ครับ 4 ช่วง

- def python ภาษาจะเรียกชื่อฟังก์ชัน เช่น def function\_name(): (ขาดไม่ได้)
- กำหนดตัวแปรที่ต้องมี input \_Input\_, (ขาดไม่ได้)
- กำหนด操作ที่ต้องมี do\_something\_with \_Input\_ to get \_Output\_, (ขาดไม่ได้)
- ส่วน output return \_Output\_(ขาดไม่ได้)

5

**ตัวอย่างการเขียน normal function**

```

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

```

ฟังก์ชัน print\_name  
รับ input 3 ตัวไปคือ surname, ID, name  
ส่วน return คืนค่า = f'{name} {surname} {ID}' คือ ให้เขียน string โดยใช้ “ ”  
ที่ input แต่ต้องใส่ในตัวแปร  
Output ให้ return st

6

**ตัวอย่างการเขียน normal function**

- การเขียนฟังก์ชันเช่นฟังก์ชันที่รับ input ตามที่ def ไว้
- แบบที่ 1 print\_name('Name', 'xxxxxxxx', 'new')
- ผลลัพธ์จะได้ ชื่อ นามสกุล บิ๊น หรือ ที่ต้อง xxxx0000xx
- แบบที่ 2 print(print\_name(name='new', surname='new', ID='603021855-2'))
- ผลลัพธ์จะได้ ชื่อ นามสกุล บิ๊น นามสกุล บิ๊น รหัส 603021855-2

7

**ฟังก์ชันไม่จำเป็นต้องมี output หรือการ return**

```

def print_name2(surname, ID, name):
    st = f'{name} {surname} {ID}'
    print(st) #จะมี print แล้ว return None

print_name2(name='new', surname='new', ID='603021855-2')

• ผลลัพธ์ =
• ชื่อ นามสกุล บิ๊น บิ๊น รหัส 603021855-2

```

8

**ฟังก์ชันไม่จำเป็นต้องมี input**

```

def PI():
    return 3.14159265359

• PI(7**2) # หาที่เท่ากับวงกลมที่รัศมีเท่ากับ 7 * r**2
• ผลลัพธ์ได้
12.56637061436

• ฟังก์ชันที่ไม่จำเป็นต้องมี input
• 1. def ชื่อฟังก์ชัน();
• 2. นิยามราก (indent) ตามต้องการประมวลผล

```

9

การกำหนดค่า default ให้กับฟังก์ชัน

```

• input ของ function ใน python ถ้า 2 แบบ
  1. input ที่รับค่าโดยอิสระ
  2. input ที่รับค่าโดยมีตัวค่า默定
  * เช่น ค่า默定 ที่เราสามารถตั้งค่าได้

  def print_2lines_default(name, surname, ID, grade='F'):
    s = f'{name} {surname} {ID}\n'
    print(s)
    st2=f'Name Data Viz >>> {grade}'
    print(st2)

  * grade='F' เป็นการกำหนดค่า默定 ให้กับ grade เป็น F

```

10

ตัวอย่างการใช้งานการกำหนดค่า default ให้กับฟังก์ชัน 1

```

• print_2lines_default("Banana", "Mae", "613020551")
  * ผลลัพธ์ =
  * ชื่อ บานานา นามสกุล พรมยลลักษณ์ อัพที่ 613020551-B
  * เนตวิชชานา Data Viz >>> F

  * จะเห็นว่า ไม่มีการตั้งค่า默定 grade ใน input แต่พิมพ์ grade ให้กับวิชชานา Data Viz >>> F
  * เวลาเราไม่พิมพ์ชื่อการกำหนดค่า default ให้กับค่าวิชชานา grade เป็น F

```

11

ตัวอย่างการใช้งานการกำหนดค่า default ให้กับฟังก์ชัน 2

```

• print_2lines_default("Banana", "Mae", "613020551", "A")
  * ผลลัพธ์ =
  * ชื่อ บานานา นามสกุล พรมยลลักษณ์ อัพที่ 613020551-A
  * เนตวิชชานา Data Viz >>> A

  * สำหรับ input ค่าที่ไม่ระบุ grade และ默定ค่า

```

12

งานในห้อง กับไปสร้างฟังก์ชันใน HW python101

- ให้สร้างฟังก์ชันที่รับค่ามาเป็นรูปทรงสามเหลี่ยม และ print ออกมาในรูปสามเหลี่ยม

13

LOOP การวนซ้ำ

```

• for each_member in listA :
    do_something

  * for เป็นคำสั่งของ python ว่าจะก้าลงชื่อใน loop โดย for จะวนศัพท์สมาชิกจาก 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 ให้บูรณาการลูปที่มีใน list [1,2,3] ให้process ค่าตามค่าที่มีในลูป
  * กรณีตัว บูรณาการ อาทิตย์ 2 บูรณาการ ให้บูรณาการ 0 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 กับไป python101

3. ให้สร้าง list ของ เวลา ['12:30:15','13:41:07',... ] แล้วบันทึกเรื่องฟังก์ชันนี้ในคลาส

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 code.
- <https://www.freecodecamp.org/news/how-to-write-clean-code/>
- <https://dev.to/aleksandrmyer/10-most-known-patterns-for-writing-clean-code-with-python-56bf>

17



1

ตัวอย่าง LOOP การวนซ้ำ



- ```
for i in 'Thanapong':  
    print(f'{i}() - /')
```
- หมายความว่า ให้รันฟังก์ชันที่เป็น อะไน "Thanapong" โดยมีค่า input ตามที่เรียกฟังก์ชันมา เช่น 0(), 1(), 2(), 3(), 4(), 5(), 6(), 7(), 8(), 9()
- รันในบัญชีผลลัพธ์ ให้成 f'{i}() - /' ไม่ print string

2

## ตัวอย่าง LOOP การวนซ้ำ

• ผลลัพธ์จะได้ รูปแบบว่า loop ท้าทาง print อย่างไร ก็จะรัน ไปถึง ตามมา คือที่นี่มีการวนซ้ำอยู่ใน อย่าง หรือคำสั่งที่บังคับให้วนไป

• บันทึกไว้

• T => /

• h => /

• a => /

• n => /

• a => /

• p => /

• o => /

• e => /

• g => /

三

- ใช้คำสั่ง `print` ที่มี argument และตัว `default` สำหรับค่าเมื่อไม่ระบุค่า
- `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

ตัวอย่างการสังลูป ให้ print แบบไม่เว้นบรรทัด



- end = '' ใช้
- ในการ python ภาษาเขียนพิเศษ ถึง หมายความว่า การเว้นบรรทัด
- ตัวอย่าง print และเว้นบรรทัด ให้รู้ว่า กรณีที่ต้องใส่ end = '' ต้องเข้าใจว่าต้อง
- for i in 'Thanapong':
  - print(f'{i} / ', end = '')
- กรณี end = '' คือไม่เว้นบรรทัดแต่ใส่
- ? / -> / h -> / a -> / n -> / a -> / p -> / o -> / n -> / q -> /

5

ด้วยการรีส์ loop ให้ print แบบไม่เว้นบรรทัด

```
for i in "Thanapong":  
    print(i) >> / , end = ','  
print end = ',' ไม่ใช่ตัวอักษร แต่เป็นตัวไดร์  
T >> / , h >> / , a >> / , n >> / , p >> / , o >> / , n >> / , g >> / ,
```

6

range() การสร้าง list ด้วยเลขแบบบอตไม่มีติ



- range() คือตัวฟังก์ชันที่เราใช้เพื่อเก็บ เช่น
- `range5_output = range(5)`
- `print(list(range5_output))`
- หมายความว่า ให้เรา ไป เดินทางบน 5 ถ้า เริ่มจาก 0 และที่ก้าวไปในที่สุด range5\_output
- ยังไง print คืนมา range5\_output ในรูปแบบ list
- ผลลัพธ์คือ
- [0, 1, 2, 3, 4]

7

- ตัวอย่างการใช้งาน `range()` สร้าง list ตัวเลขในการวนลูป
  - `for i in range(100):`
  - `print("strand " + str(i))`
- หมายความว่า ให้วน 100 รอบ โดยพิมพ์ชุดตัวอักษรที่เป็น กองหิน(100) ซึ่งเป็น list 0 ถึง 99
  - `print(*range(0,100))`
  - `print(*range(1,100))`
- |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 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 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

8

- `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(0) 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()` ສາມາດກຳນົດຕົວຂອງລາຍເຊີງ `range(start, stop[, step])`

6

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

```

• list(range(1,11))
• วนรอบจาก 1 ถึง 10 สำหรับตัวแปร range ไม่ต้องมี值 1 เป็น 10
• เมื่อพิมพ์ได้
• [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

• list(range(-3,20,4))
• วนรอบจาก -3 ถึง 20 สำหรับตัวแปร range ไม่ต้องมี值 3 เป็น 19 เมื่อพิมพ์ step=4
• เมื่อพิมพ์ได้
• [-3, 1, 5, 9, 13, 17]

```

10

loop in loop

- สามารถเขียน loop ภายใน loop ได้
- โดยต้องบันทึกค่าตามมา กรณีบรรทัด (indent) ให้ต้องบันทึกลงใน loop จะต้องลบ
- หากพิมพ์แล้วลากloop2 ลงมาloop1 ลงloop2 หาบันทึกของloop1 ให้ลบloop1
- คืนloop ลงห้ามบันทึกและลบloop2 ลงloop1 ยกเว้นloop1 ยกเว้นloop

11

ตัวอย่าง loop in loop

```

for loop1 in range(2,5): #(2, 3, 4)
    print("now loop1 = ",loop1)
    for loop2 in range(1,13):
        print("loop1 = ",loop1," , loop2 = ",loop2)
        print("the inner loop is end")

```

จะเห็นว่า ในการเขียนloop ภายในloop ที่ range(2,5) หรือ (2, 3, 4) ให้บันทึกค่าที่ต้องบันทึกลงในloop1 ลงloop2 ให้บันทึก 2 ห้ามบันทึกค่าที่บันทึกลงloop1 13 ซึ่งต้องลบ 1 ถึง 12 ที่บันทึกไว้ 2 ลงloop2 ที่ 2 ให้ print("loop1 = ",loop1," , loop2 = ",loop2) บันทึก 2 จุดที่ 12 เมื่อบันทึก 2 ให้ print("the inner loop is end") จากนั้นจะลบตัวที่บันทึกไว้เมื่อจบการวนloop(2)

12

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 | • the inner loop is end | • the inner loop is end |

13

loop in function

- สามารถบันทึกค่าที่บันทึกได้ ให้อ่าน กรณีบรรทัด (indent) ให้บันทึกค่าที่ก่อน
- def print\_feelings(list\_friends,feeling = "嗨欸"):
 for friend in friends:
 print(f'{feeling} {friend}')
- จะเห็นว่า บันทึกค่าที่บันทึกได้ print\_feelings
- def print\_feelings(list\_friends,feeling = "嗨欸") (defun)
 for friend in friends:
 print(f'{feeling} {friend}')
- จะเห็นว่า บันทึกค่าที่บันทึกได้ print\_feelings

14

ตัวอย่างการใช้งาน loop in function

```

list_friends = ['ปีกุ/พะนး/เหลน/ນุ่น/นิจ/น้องแมว']
print_feelings('สวัสดี',list_friends)

• คำสั่งพิมพ์ list_friends เป็นตัวที่ [ปีกุ/พะนး/เหลน/นุ่น/นิจ/น้องแมว]
• ใช้คำพิมพ์ชื่อ print_feelings([สวัสดี],list_friends)
    ซึ่งพิมพ์ค่าเพื่อต้องการ หาบันทึกค่าที่บันทึกไว้ให้แสดงผลต่อไปนี้ "สวัสดี" = 1, list_friends = friends แล้ว print([feeling] (friend)) ค่าที่บันทึกไว้ 1 คือปีกุ ค่าที่บันทึกไว้ 2 คือพะนุ่น ค่าที่บันทึกไว้ 3 คือเหลน ค่าที่บันทึกไว้ 4 คือนุ่น ค่าที่บันทึกไว้ 5 คือนิจ ค่าที่บันทึกไว้ 6 คือน้องแมว

```

15

ตัวอย่างการใช้งาน loop in function

```

• เมื่อพิมพ์ print_feelings('สวัสดี',list_friends) ไม่มี input ค่าที่บันทึกได้
• คำสั่งพิมพ์ คือ
• คำสั่ง คือ
list_friends = ['ปีกุ/พะนุ่น/เหลน/นุ่น/นิจ/น้องแมว']
• คำสั่ง คือ แบบนี้
def print_feelings(i,friends,feeling = "嗨欸"):
    for friend in friends:
        print(f'{feeling} {friend}')
• แบบ คือ ที่พิมพ์
• แบบ คือ แบบนี้

```

16

ตัวอย่างการใช้งาน loop in function

```

• เมื่อพิมพ์ print_feelings('สวัสดี',list_friends,'嗨欸')
• แบบ คือ ที่พิมพ์
• แบบ คือ แบบนี้
list_friends = ['ปีกุ/พะนุ่น/เหลน/นุ่น/นิจ/น้องแมว']
• แบบ คือ แบบนี้
def print_feelings(i,friends,feeling = "嗨欸"):
    for friend in friends:
        print(f'{feeling} {friend}')
• แบบ คือ ที่พิมพ์
• แบบ คือ แบบนี้

```

17

conditional คือ เงื่อนไข /ประไครเงื่อนไข

```

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

```

สามารถกำหนดเงื่อนไขเพื่อทำการทำบางอย่างบนเงื่อนไขโดยให้บันทึกเงื่อนไขของ if condition1 [ให้ปั๊บเงื่อนไข ก็จะทำงาน]
 ถ้าไม่ใช่เงื่อนไขเดิมๆ ให้ condition2 ตามมาอีก (แต่สามารถมีมากกว่า 1)
 และหากไม่ใช้เงื่อนไขใดๆ ก็จะรับเงื่อนไขของ else จะทำงาน

18

ตัวอย่าง Loop แบบปกติ

```

• list friends = ['ปิยะ', 'พานิช', 'นภัส', 'เพ็ชร์', 'นัฐพงษ์']

• for name1 in list_friends:
    for name2 in list_friends:
        print(name1, '<-->', name2)

• สรุป for บล็อกเพื่อ print(name1, '<-->', name2) เมื่อไหร่จะมีการซ้ำกันกันแน่

```

19

ผลลัพธ์จะได้

|                      |                      |                      |
|----------------------|----------------------|----------------------|
| • ปิยะ กับ พานิช     | • นภัส กับ นภัส      | • เพ็ชร์ กับ นัฐพงษ์ |
| • ปิยะ กับ นภัส      | • นภัส กับ พานิช     | • พานิช กับ นภัส     |
| • ปิยะ กับ เพ็ชร์    | • เพ็ชร์ กับ ปิยะ    | • เพ็ชร์ กับ นัฐพงษ์ |
| • ปิยะ กับ นัฐพงษ์   | • นภัส กับ พานิช     | • นัฐพงษ์ กับ ปิยะ   |
| • พานิช กับ พานิช    | • พานิช กับ เพ็ชร์   | • พานิช กับ นัฐพงษ์  |
| • พานิช กับ นภัส     | • นภัส กับ พานิช     | • เพ็ชร์ กับ พานิช   |
| • พานิช กับ เพ็ชร์   | • เพ็ชร์ กับ พานิช   | • เพ็ชร์ กับ นภัส    |
| • พานิช กับ นัฐพงษ์  | • นัฐพงษ์ กับ พานิช  | • นัฐพงษ์ กับ เพ็ชร์ |
| • นภัส กับ พานิช     | • พานิช กับ นภัส     | • นภัส กับ เพ็ชร์    |
| • นภัส กับ เพ็ชร์    | • เพ็ชร์ กับ พานิช   | • เพ็ชร์ กับ นภัส    |
| • นภัส กับ นัฐพงษ์   | • นัฐพงษ์ กับ พานิช  | • นัฐพงษ์ กับ เพ็ชร์ |
| • เพ็ชร์ กับ พานิช   | • พานิช กับ เพ็ชร์   | • เพ็ชร์ กับ นภัส    |
| • เพ็ชร์ กับ นภัส    | • นภัส กับ เพ็ชร์    | • นภัส กับ นัฐพงษ์   |
| • เพ็ชร์ กับ นัฐพงษ์ | • นัฐพงษ์ กับ เพ็ชร์ | • นัฐพงษ์ กับ นภัส   |

20

ตัวอย่างการใช้งาน conditional

```

• for name1 in list_friends:
    for name2 in list_friends:
        if name1 == name2: # เป็นกรณีซ้ำกัน
            print(name1, '<-->', name2)

• สรุปเมื่อให้มา ที่ name1 > name2 ให้หักห้ามแสดงผลดูบ้าง เมื่อมันเรียกใช้ในฟังก์ชันต่อๆ กันไป
• เช่น ถ้า name1 คือ พานิช ไม่ต้อง name2 ให้แสดงซึ่งจะเป็นไปไม่ได้

```

21

ผลลัพธ์จะได้

|                      |                      |                      |
|----------------------|----------------------|----------------------|
| • ปิยะ กับ พานิช     | • นภัส กับ พานิช     | • เพ็ชร์ กับ พานิช   |
| • ปิยะ กับ นภัส      | • นภัส กับ พานิช     | • เพ็ชร์ กับ นภัส    |
| • ปิยะ กับ เพ็ชร์    | • เพ็ชร์ กับ พานิช   | • เพ็ชร์ กับ นภัส    |
| • ปิยะ กับ นัฐพงษ์   | • นัฐพงษ์ กับ พานิช  | • นัฐพงษ์ กับ นภัส   |
| • พานิช กับ พานิช    | • พานิช กับ นภัส     | • พานิช กับ เพ็ชร์   |
| • พานิช กับ นภัส     | • นภัส กับ พานิช     | • พานิช กับ นัฐพงษ์  |
| • พานิช กับ เพ็ชร์   | • เพ็ชร์ กับ พานิช   | • เพ็ชร์ กับ นภัส    |
| • พานิช กับ นัฐพงษ์  | • นัฐพงษ์ กับ พานิช  | • นัฐพงษ์ กับ เพ็ชร์ |
| • นภัส กับ พานิช     | • พานิช กับ นภัส     | • นภัส กับ เพ็ชร์    |
| • นภัส กับ เพ็ชร์    | • เพ็ชร์ กับ พานิช   | • เพ็ชร์ กับ นภัส    |
| • นภัส กับ นัฐพงษ์   | • นัฐพงษ์ กับ พานิช  | • นัฐพงษ์ กับ เพ็ชร์ |
| • เพ็ชร์ กับ พานิช   | • พานิช กับ เพ็ชร์   | • เพ็ชร์ กับ นภัส    |
| • เพ็ชร์ กับ นภัส    | • นภัส กับ เพ็ชร์    | • นภัส กับ นัฐพงษ์   |
| • เพ็ชร์ กับ นัฐพงษ์ | • นัฐพงษ์ กับ เพ็ชร์ | • นัฐพงษ์ กับ นภัส   |

22

ข้อจำกัดของ การใช้งาน conditional

```

• for name1 in list_friends:
    for name2 in list_friends:
        if name1 == name2:
            "do nothing"
        else:
            print(name1, '<-->', name2)

• ภารกิจสืบทอด ขึ้นบันทึกเมื่อร่วมมูลผลในงานบ้านอย่างไรได้

```

23

สัญลักษณ์ที่ใช้ใน conditional

```

• == คือ ไม่เท่ากัน
• != คือ ไม่เท่ากัน
• >= คือ มากกว่าหรือเท่ากัน ใช้ในการพิจารณาตอบตัวเลข
• <= คือ น้อยกว่าหรือเท่ากัน ใช้ในการพิจารณาตอบตัวเลข
• < คือ น้อยกว่า ใช้ในการพิจารณาตอบตัวเลข
• > คือ มากกว่า ใช้ในการพิจารณาตอบตัวเลข

```

24

Ture False ใน conditional

```

• หากต้องการให้เป็น True จะต้องสอนประยุกต์กรณี If
• If True:
    • print("Yes")
• กรณี False:
    • Yes
• หากต้องการให้เป็น False ให้สอนว่าต้องป้อนอะไรลงมาแทนถ้าไม่ต้องการให้เป็น True
    • If False:
        • print("No")
    • กรณี False:
        • ไม่รู้เรื่องของภาษา Python ให้สอนเข้าไปที่ร้านของบ้านและสอนว่าถ้า If ที่ห้องศูนย์เรียน

```

25

Ture False ใน conditional

```

• ที่อยู่ == ที่เดิม'
• กรณีเช่นเดิม
• False

• ที่อยู่ != ที่เดิม'
• กรณีเช่นเดิม
• True

```

26

Homework class period 3

```

• เรียน function ระบุตัวแปร ให้ input เป็นตัวแปร(0-100), アウト เป็นตัว( F-A)
• ระบุตัวแปร input = (1,50,65,90,101,5,40)
• คำนวณ
• คำนวณตัวแปรทางคณิตศาสตร์ 0 แต่งตัว 50 แต่งตัว 0
• คำนวณตัวแปรทางคณิตศาสตร์ 5 แต่งตัว 55 แต่งตัว 0
• คำนวณตัวแปรทางคณิตศาสตร์ 60 แต่งตัว 60 แต่งตัว 0
• คำนวณตัวแปรทางคณิตศาสตร์ 65 แต่งตัว 70 แต่งตัว 0
• คำนวณตัวแปรทางคณิตศาสตร์ 70 แต่งตัว 75 แต่งตัว 8
• คำนวณตัวแปรทางคณิตศาสตร์ 75 แต่งตัว 80 แต่งตัว 8
• คำนวณตัวแปรทางคณิตศาสตร์ 80 แต่งตัว A
• คำนวณแบบต่อตัว 0 และตัว 100 ไม่รีบยก หาค

```

27

**Class period 4**

บทที่ 3 โน้ตบุ๊กภาษา Python และการใช้โครงสร้างข้อมูล (cont.)  
More advanced data structure 1

1

**ทบทวน List (เรียนไปแล้ว)**

Numbers[index]

index = 0 1 2 3 4

- List คือ Array ยังไง
- 1. เก็บข้อมูลโดยรอบมาใน int float
- 2. มี list slicing

2

**Dictionary**

สามารถกำหนด index ให้กับค่าใน dict และสามารถกำหนด index เป็นอะไรก็ได้ แต่ต้องห้ามไป

```
ex_dict = {'a':10, 'b':11, 'stat':5}
#curly brackets () บรรยาย dict
ex_dict['stat']

ผลลัพธ์ได้
5
```

3

**ข้อก้าวเดินในการใช้งาน Dictionary**

- 2:'two'
- ด้วยว่า - คือ key หรือ index (ตัวซึ้น)
- ตัวซึ่ง - คือ value (ค่า)
- ในภาษาถึง value ไม่ควรที่จะบันทึกเป็น index

```
• ex2_dict['two']
• ผลลัพธ์ได้
• KeyError: 'two'
```

4

**ตัวอย่างการใช้งาน Dictionary**

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

ผลลัพธ์ได้
'two'
```

5

**การเพิ่มสมาชิกใน dict**

- สามารถกำหนด index (ตัวซึ้น) และ value (ค่า) ที่ต้องการได้ เช่น
- ex2\_dict[0] = 'zero'
- ต้องการเพิ่ม index 0 ซึ่งที่ value 'zero' ที่เราให้ใน dict ex2\_dict

```
• ex2_dict
• ผลลัพธ์ได้
{0: 'zero', 1: 'one', 2: 'two', 3: 'three'}
```

6

**คำสั่งที่ใช้บ่อยของ dict**

- .keys() คือคำสั่งที่ใช้ตรวจสอบ keys ใน dict ว่ามี index (ตัวซึ้น) อะไรบ้าง
- ex2\_dict.keys()
- ผลลัพธ์ได้
- dict\_keys([1, 2, 3, 0])

- .values() คือคำสั่งที่ตรวจสอบ values ใน dict ว่ามี values (ค่า) อะไรบ้าง
- ex2\_dict.values()
- ผลลัพธ์ได้
- dict\_values(['one', 'two', 'three', 'zero'])

7

**ตัวอย่างการใช้งานคำสั่ง .keys() และ .values()**

```
• สามารถใช้ร่วมกับ loop for เพื่อให้ keys หรือ values ไปเข้ามาต่อ เช่น
• for index in ex2_dict.keys():
•     print(index)
• ผลลัพธ์ได้
• 1
• 2
• 3
• 0
```

\*\*\*dict อาจไม่ใช้กับ DataFrame และ pandas

8

**Numpy Array (array n มิติ)**

1D array  
shape: (4,) ผลลัพธ์:  
[7 2 8 3]

2D array  
shape: (2, 2) ผลลัพธ์:  
[[3 2 3 4]  
[3 1 3 2]]

3D array  
shape: (4, 3, 2) ผลลัพธ์:  
[[[[3 2]  
[3 1]  
[3 2]  
[3 1]]  
[[3 2]  
[3 1]  
[3 2]  
[3 1]]  
[[3 2]  
[3 1]  
[3 2]  
[3 1]]  
[[3 2]  
[3 1]  
[3 2]  
[3 1]]]

- numpy คือ package ที่ทำมาสำหรับ array ของตัวเลขหลายมิติ ที่มีความเชื่อมโยงกัน
- numpy รองรับภาษา number python

9

วิธีเรียกใช้งาน package

- สามารถใช้ `import` เพื่อเรียก package ของ python ที่ต้องการใช้งาน เช่น
- `import numpy`
- สามารถใช้ `as` เพื่อเรียก package เพื่อตั้งชื่อใหม่ของค่าการใช้งาน เช่น
- `import numpy as np`

10

การแปลง list ให้เป็น numpy array

- เมื่อเขียน list เข้าไป array 2 มิติ ท่านต้องนำ `numpy.array()`
- `ex_2d_array=numpy.array([[5.2,3.0,4.5],[9.1,0.1,0.3]])`
- `print(ex_2d_array)`
- ผลลัพธ์จะได้ ค่าร่าง matrix
- `[[5.2 3. 4.5]  
[9.1 0.1 0.3]]`

11

การตรวจสอบขนาดของ matrix

- ใช้คำสั่ง `.shape` คำนึงถึงข้อมูลที่ตั้งไว้ใน 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)`
- หมายความว่าเป็น matrix 2 แถว 3 หลัก
- `[[5.2 3. 4.5]  
[9.1 0.1 0.3]]`

12

การซั้งใน numpy array

- `ex_2d_array=[[5.2 3. 4.5]  
[9.1 0.1 0.3]]`
- ถ้าต้องการซั้งเพิ่ม 0.3 ใน `ex_2d_array`
- โดยต้องเข้าไปปรับ matrix ที่ 0.3
- จะเห็นใน matrix ที่ 1 พื้นที่ 2 (เน้นน้ำเงิน 0) ต้องบวก
- `ex_2d_array[[1,2]]`
- ผลลัพธ์จะได้
- `[[5.2 3.0 4.5]  
[9.1 0.1 0.3]]`

13

การซั้งใน list ก่อนที่จะเปลี่ยนเป็น numpy array

- `list_x = [[5.2,3.0,4.5],[9.1,0.1,0.3]]`
- ถ้าต้องการซั้งเพิ่ม 0.3 ใน `list_x`
- `print(len(list_x))` แสดงผลว่า 1 2 หมายความว่า `list_x` มีสมาชิก 2 ตัว คือ `[5.2,3.0,4.5]` และ `[9.1,0.1,0.3]`
- `print(list_x[1])` แสดงผลจะได้ `[9.1,0.1,0.3]` ตัวเดียว
- `print(list_x[1][2])` แสดงผลจะได้ 0.3 คือบวกเพิ่มขึ้นที่ 2 ของ `list_x` และจะบวกเพิ่มขึ้นที่ 2 ของ `list_x[1]` ก็จะได้ `list_x[1][2]` คือ 0.3

14

Operations

- `numpy array` หรือ `matrix` สามารถบวก ลบ คูณ หาร คิดได้
- สร้าง matrix ให้มีลักษณะกู่กันไว้แล้วเพื่อ `ex_2d_array`
- `ex2_2d_array = numpy.array([[1,0,0],[0,0,1]])`
- `print(ex_2d_array)` (matrix)
- `[[5.2 3. 4.5]  
[9.1 0.1 0.3]]`
- `print(ex2_2d_array)`
- `[[1 0 0]  
[0 0 1]]`

15

ตัวอย่างใช้งานการบวก matrix

- การบวก matrix ต้องการให้ทั้งสอง matrix มีขนาดเท่ากัน
- `ex_2d_array + ex2_2d_array`
- คุณลองมาดู
- `[[5.2 3. 4.5] + [[1 0 0]  
[9.1 0.1 0.3]] + [0 0 1]]`
- ผลลัพธ์จะได้
- `array([[6.2, 3., 4.5],  
[9.1, 0.1, 1.3]])`

16

การลบค่าใน list ก่อนที่จะเปลี่ยนเป็น matrix

- `list_x = [[5.2,3.0,4.5],[9.1,0.1,0.3]]`
- `list_xd = [[1,0,0],[0,0,1]]`
- `list_x - list_xd`
- ผลลัพธ์จะได้
- `[[5.2, 3.0, 4.5], [9.1, 0.1, 0.3], [1, 0, 0], [0, 0, 1]]`
- จะเห็นว่า list ไม่สามารถลบค่าด้านหน้าด้วยค่าด้านหลัง matrix ใน `numpy array`
- งานเกิดขึ้นไม่สามารถบังคับให้ ลดลงที่ศูนย์ด้วย `numpy array`

17

ตัวอย่างใช้งานการลบ matrix

- `ex_2d_array - ex2_2d_array`
- คุณลองมาดูกัน
- `[[5.2 3. 4.5] - [[1 0 0]  
[9.1 0.1 0.3]] - [0 0 1]]`
- ผลลัพธ์จะได้
- `array([[ 4.2, 3., 4.5],  
[ 9.1, 0.1, -0.7]])`

18

ตัวอย่างใช้งานการคูณ matrix

- การคูณ matrix ใน numpy array จะดูในเม้น array เองที่แทนเพื่อเก็บมาราบกัน
- `ex_2d_array * ex2_2d_array`
- คุณผลลัพธ์ได้
- $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 100 & 108 \\ 142 & 152 \end{bmatrix}$
- `array([[15.2, 0., 0., 1., 0., 0., 0., 0.3]])`

19

matrix multiplication (dot product)

การคูณ matrix ใน numpy array ที่ถูกเรียกว่า dot product คือการหาผลรวมของผลคูณของตัวหารในแต่ละคอลัมน์ของ matrix ที่สองกับตัวเดียวกันในแต่ละแถวของ matrix ที่หนึ่ง

20

ตัวอย่างการคูณ matrix (dot product)

- `numpy.dot(ex_2d_array, ex2_2d_array)`
- คุณผลลัพธ์ได้
- $\begin{bmatrix} 15.2 & 3. & 4.5 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1.2 \end{bmatrix}$
- `ValueError: shapes (2, 3) and (2, 3) not aligned: 3 (dim 1) != 2 (dim 0)`

21

ตัวอย่างการคูณ matrix (dot product)

- หลักการคูณ matrix ก็ หลักของจำนวนตัวเลขน้ำหนักและตัวน้ำหนัก คือบันทึกต่อ transpose matrix (การเปลี่ยนตัวน้ำหนักให้เป็นแมตริกซ์แล้วแล้วเปลี่ยนตัวน้ำหนักกลับ)
- สามารถ transpose matrix ที่จะสามารถให้ได้ออกซีลัง ซึ่งจะเป็นปริญาก้า matrix ตามด้านล่าง
- `print(ex_2d_array)`
- $\begin{bmatrix} 15.2 & 3. & 4.5 \end{bmatrix}$
- $\begin{bmatrix} 9.1 & 0.1 & 0.3 \end{bmatrix}$
- `print(ex_2d_array.T)`
- $\begin{bmatrix} 15.2 & 9.1 \\ 3. & 0.1 \\ 4.5 & 0.3 \end{bmatrix}$

22

ตัวอย่างการคูณ matrix (dot product)

- รันนิ่น การคูณ matrix ใน numpy array ที่ถูกต้อง
- `dot_mat = numpy.dot(ex_2d_array, ex2_2d_array.T)`
- คุณผลลัพธ์ได้
- $\begin{bmatrix} 15.2 & 3. & 4.5 \end{bmatrix} \times \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0.3 \end{bmatrix} = \begin{bmatrix} 15.2 & 4.5 \\ 3. & 0.3 \\ 4.5 & 1.2 \end{bmatrix}$
- `print(dot_mat)`
- `dot_mat`

23

การหา det ของ matrix

- ใช้คำสั่ง `numpy.linalg.det` คำนวณ (ต้องมีการติดต่อภาษา Python)
- `numpy.linalg.det(dot_mat)`
- คุณผลลัพธ์ได้
- $-35.389999999999986$

24

matrix slicing

- `print(ex_2d_array)`
- $\begin{bmatrix} 15.2 & 3. & 4.5 \end{bmatrix}$
- $\begin{bmatrix} 9.1 & 0.1 & 0.3 \end{bmatrix}$
- สามารถตัดได้เหมือนกัน
- `ex_2d_array[:,1:]` หมายความว่า “ตัดมาตั้งแต่คอลัมน์ที่ 1 ไปจนถึงคอลัมน์ที่ 2”
- `ex_2d_array[1:,1:]` หมายความว่า “ตัดมาตั้งแต่คอลัมน์ที่ 1 ไปจนถึงคอลัมน์ที่ 2 และตัดมาตั้งแต่แถวที่ 2 ไปจนถึงแถวที่ 3”
- `dot_mat`
- `array([[15.2, 3., 1., 0.], [9.1, 0.1, 0.3]])`

25

ตัวอย่าง matrix slicing

- `print(ex_2d_array)`
- $\begin{bmatrix} 15.2 & 3. & 4.5 \end{bmatrix}$
- $\begin{bmatrix} 9.1 & 0.1 & 0.3 \end{bmatrix}$
- `ex_2d_array[:,1:]` หมายความว่า “ตัดมาตั้งแต่คอลัมน์ที่ 1 ไปจนถึงคอลัมน์ที่ 2”
- `dot_mat`
- `array([[15.2, 3., 1., 0.], [9.1, 0.1, 0.3]])`

26

Homework class period 4

- ใช้ function คูณ matrix ที่เราเขียนไว้ตอน dot product (ในไฟล์ dot product)
- แล้ว test กับ matrix ขนาด  $(2,3) \times (2,3)$
- $(4,4) \times (4,4)$
- $(2,2) \times (2,2)$

27

**Class period 5**

บทที่ 3 วิบัติพจนานุกรมและการใช้เมธอดในการแก้ไขข้อมูล (cont.)  
More advanced data structure 2

1

**zeros คำสั่ง numpy.zeros()**

numpy.zeros

• เบื้องต้นคำสั่ง “`zeros`” คือการสร้าง matrix ที่มีทุกๆ 值 เป็น 0

```
import numpy as np
np.zeros(2)
array([0., 0.])
```

• `np.zeros(2,3)`

```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

2

**ones คำสั่ง numpy.ones()**

numpy.ones

• เบื้องต้นคำสั่ง “`ones`” คือการสร้าง matrix ที่มีทุกๆ 值 เป็น 1

```
np.ones(2)
array([1., 1.])
```

• `np.ones(2,3)`

```
array([[1., 1., 1.],
       [1., 1., 1.]])
```

3

**Matrix Operation (scalar multiplication)**

scalar multiplication คือ การคูณแต่ละตัว : คำนี้เป็นส่วนหนึ่งของ matrix เช่น

```
M_one = np.ones((2,3))
# แปลง matrix ให้เป็น 2x3
M_one
array([[1., 1., 1.],
       [1., 1., 1.]])
```

• `2*M_one`

• ผลลัพธ์ได้

```
array([[2., 2., 2.],
       [2., 2., 2.]])
```

4

**Random**

คำสั่ง `numpy.random` เป็นคำสั่งที่สร้าง matrix ที่สุ่มค่าภายใน matrix

- คำสั่ง `numpy.random` มีหลายประเภท โดยคำสั่งที่พบบ่อยที่สุด 3 ประการคือ
  1. `numpy.random.rand`
  2. `numpy.random.randn`
  3. `numpy.random.choice`

5

**คำสั่ง numpy.random.rand()**

numpy.random.rand

• ถ้า `random.rand` จะเป็น uniform random นั่นหมายความว่ามีการสุ่มได้มาที่นี่ ซึ่งสุ่มไม่ได้มีผลลัพธ์ 0 ถึง 1

```
np.random.rand(3,2)
array([[0.249091, 0.34341953],
       [0.22536518, 0.86663463]])
```

6

**คำสั่ง numpy.random.randn()**

• ถ้า `random.randn` จะเป็น normal distribution mean=0 std=1 เป็นการสร้างตัวที่เพื่อให้ตัวนั้น 0 จริงๆ ไม่ต้องดูในหน้าจอภาพที่ถูกหัก 0 ซึ่งตัวนี้ต้องดูผลลัพธ์ 0 ถึง 1

```
np.random.randn(3,2)
array([[ 2.06762285,  0.91239845],
       [-2.08011942, -0.46261935],
       [ 0.66804796,  1.19419422]])
```

7

**คำสั่ง numpy.random.choice()**

• ถ้า `random.choice` จะเป็นการสุ่มตัวที่กำกับลง

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

8

**ตัวอย่างการใช้งาน Parameter: size numpy.random.choice()**

• Parameter: size เป็นการกำหนดจำนวนตัวที่ต้องการสุ่ม出來โดย size สามารถ input ค้าบเป็น int (ด้านล่าง) หรือ tuple of ints (ด้านล่างดูบุคลิกจะกันเป็น tuple)

```
np.random.choice([1,2,3,28,11,100],size = 2)
array([28, 31])
```

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

```
array([[ 1,  3, 11],
       [28,  1, 11]])
```

9

ตัวอย่างการใช้งาน Parameter: replace numpy.random.choice()

- Parameter: replace การให้คืน เป็นการก้าวหน้า จะให้คำว่าที่สุ่มออกมามีคืนหรือไม่คืนคือ การสุ่มค่าซึ่งรีปิเม็ท
- โดย default ของ replace = True คือการสุ่มจะไม่ให้มีเม็ดเดียวซ้ำๆ แต่จะเป็นเม็ดเดียวซ้ำๆ บ่อยๆ มากขึ้น ให้กำหนด replace = False

```

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

```

10

ตัวอย่างการใช้งาน Parameter: p numpy.random.choice()

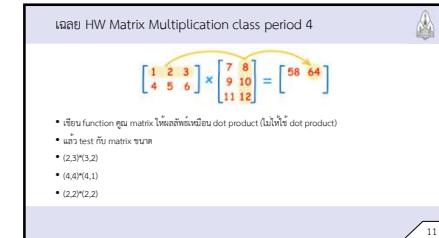
- Parameter: p เป็นการกำหนดความน่าเป็นค่าของแต่ละตัว
- ตัวอย่าง 1 5 แสดงตัวอย่างการให้คำแนะนำตามความน่าเป็นที่คาดหวัง
- กรณีนี้ ให้คำแนะนำเป็นอย่างต่ำ 10%
- กรณีนี้ ให้คำแนะนำเป็นอย่างต่ำ 20%
- กรณีนี้ ให้คำแนะนำเป็นอย่างต่ำ 30%
- กรณีนี้ ให้คำแนะนำเป็นอย่างต่ำ 2%
- กรณีนี้ ให้คำแนะนำเป็นอย่างต่ำ 1%

```

np.random.choice(['A','B','C','D','E'], p=[0.1,0.2,0.2,0.2,0.2])
p = [0.1,0.2,0.2,0.2,0.2]
array(['A', 'B', 'C', 'D', 'E'])

```

11



12

การสร้างฟังก์ชัน คูณ 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

```

13

ขั้นตอนการสร้างฟังก์ชัน คูณ matrix

```

def mat_mul(A,B):
    # คือการรวมผลลัพธ์ของ matrix A กับ matrix B
    C = np.zeros((A.shape[0],B.shape[1]))
    # หมายความว่าให้เรา matrix ที่มีมาเป็น input ที่ห้ามมี matrix A * matrix B

```

14

ขั้นตอนการสร้าง loop1 สำหรับ คูณ matrix

```

for r_a in range(A.shape[0]):
    A.shape[0] คือจำนวนแถวทั้งหมด
    range(A.shape[0]) คือต้องให้จำนวนชั้นในแมทริกซ์ทั้งหมด
    ดัง A.shape[0] = 2
    range(A.shape[0]) = range(2)
    จะทำให้ range(0, 2) ในรูปแบบ list คือ[0, 1]
    ดังนั้น loop1 ใช้กลไกมาแบบ range(A.shape[0]) ซึ่งต้องให้ในรูปแบบ list ด้วย
    range คือต้องให้เป็นรูปแบบ list ด้วย
    shape คือต้องให้เป็นการตรวจสอบขนาด matrix

```

15

ขั้นตอนการสร้าง loop2 สำหรับ คูณ matrix

```

for c_b in range(B.shape[1]):
    B.shape[1] คือจำนวนคอลัมน์
    range(B.shape[1]) คือต้องให้จำนวนคอลัมน์ของ matrix
    ดัง B.shape[1] = 2
    range(B.shape[1]) = range(2)
    จะทำให้ range(0, 2) ในรูปแบบ list คือ[0, 1]
    ดังนั้น loop2 ใช้กลไกมาแบบ range(B.shape[1]) ซึ่งต้องให้ในรูปแบบ c_b

```

16

ขั้นตอนการสร้าง loop3 สำหรับ คูณ matrix

```

for every_element in range(A.shape[1]):
    A.shape[1] คือจำนวนคอลัมน์
    range(A.shape[1]) คือต้องให้จำนวนคอลัมน์ของ matrix
    ดัง A.shape[1] = 3
    range(A.shape[1]) = range(3)
    จะทำให้ range(0, 3) ในรูปแบบ list คือ[0, 1, 2]
    ดังนั้น loop3 ใช้กลไกมาแบบ range(A.shape[1]) ซึ่งต้องให้ในรูปแบบ every_element

```

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]):
• ถ้า
    • 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)')

```

18

สำหรับการที่งานนี้ loop for ในไฟล์ขั้น

- ผลลัพธ์จะเป็น
- |         |            |                   |
|---------|------------|-------------------|
| row : 0 | column : 0 | every_element : 0 |
| row : 0 | column : 1 | every_element : 1 |
| row : 0 | column : 2 | every_element : 2 |
| row : 1 | column : 0 | every_element : 0 |
| row : 1 | column : 1 | every_element : 1 |
| row : 1 | column : 2 | every_element : 2 |
- หากเรามาที่loop3 จะทำงานดูบ่อยกว่าที่loop2 กอนี้ก็ต้องมีอันเป็นวงลูป 1 รอบของ loop2 และ loop2 ของท่านก็ต้องมีวงลูป 1 รอบของ loop1

19

การคำนวณค่าภายใน loop 3

- $C[x_a, c_b] = C[x_a, c_b] + A[x_a, \text{every\_element}] * B[\text{every\_element}, c_b]$
- ในการห้ามการวนซ้ำของเพื่อที่จะเขียน  $C[x_a, c_b]$  จะแทนที่กับ 0
- ตัวอย่างเช่น  $C = np.zeros(A.shape[0], B.shape[1])$
- $A[x_a, \text{every\_element}]$  ก็คือ  $\text{array}(\text{list})$  ทั้งหมดสามตัวที่ความกว้าง  $[x_a, \text{every\_element}]$  เป็น
- ดัง A ก็คือ

  - $\text{array}([12, 20, 3],$
  - $[4, 5, 6]))$
  - $A[0,0]$  จะแทนที่ 1
  - $A[0,1]$  จะแทนที่ 2

20

การคืนค่าวนมกายใน loop3



- B[every\_element, c\_b] คือ รากของ matrix ตัวที่ k ลงมาที่ค่าบัน [every\_element, c\_b] ทั้งหมด
- ดู B ดีๆ
  - array([[ 0, 1, 2],  
... , [ 0, 1, 0],  
... , [111, 122]])
  - **B[0,0]** จะค่าที่กี่?
  - **B[1,0]** จะค่าที่กี่?

21

การคำนวณการคูณ matrix

- ตัวอย่าง
  - $C[r, a, c, b] = C[r, a, c, b] + (A[r, a, \text{every_element}] * B[\text{every_element}, c, b])$
  - ในการท่องผ่านแต่ละ matrix ลูป
  - $C[r, a, c, b]$  จะรับค่าเดิมทุกครั้ง ซึ่งต้องเพิ่มการคำนวณเมื่อที่นั่น ให้ไปบวกใน lopp ระหว่างที่ไม่เข้า
  - $C[0, 0, 0, 0] = C[0, 0, 0] + (A[0, 0, 0] * B[0, 0, 0])$
  - $C[0, 0, 0] = 0.0 + (1 * 7)$
  - $C[0, 0, 0] = 7.0$  และจะวนไปจนกว่าใน lopp หรือที่นั่น
  - $C[0, 0, 0] = C[0, 0, 0] + (A[0, 1, 0] * B[1, 0, 0])$
  - $C[0, 0, 0] = 7.0 + (2 * 9)$
  - $C[0, 0, 0] = 25.0$  และอาจยังวนอยู่ใน lopp ระหว่างที่บวกของ  $C[0, 0]$  และเมื่อันนี้  $C[0, 0]$

22

วิธีใช้งานฟังก์ชัน

- เขียน matrix input ตามที่เรียง
- ```
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]])
```
- $$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 5 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \end{bmatrix}$$

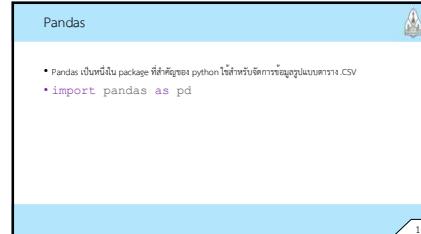
23

- matmul(mati,mat2) ใช้เพื่อคำนวณค่า input(A,B) หรือ  $A \times B$
- ผลลัพธ์ได้
  - array([[ 58., 64.],  
 [139., 154.]])
- ซึ่งสามารถเขียนรูปดังนี้ใน Python ให้คิดเป็นภาษาที่เราเคยหัดใช้กันมาก่อนก็จะง่าย多了
- [[C[0,0], C[0,1]],  
 [C[1,0], C[1,1]]]

24



1



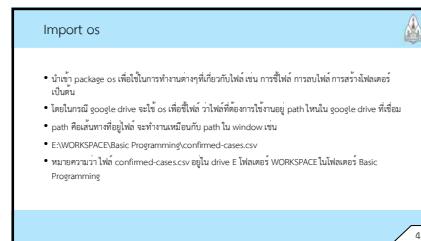
2



3



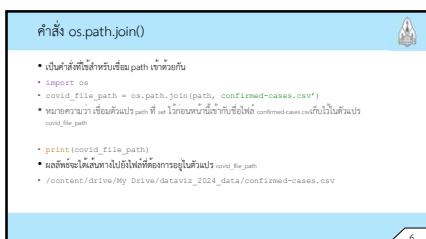
4



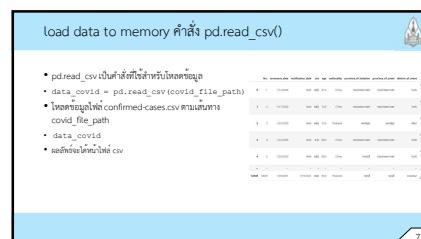
5



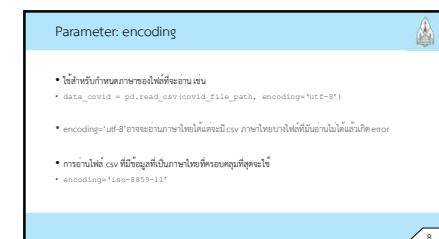
6



7



8



9

ค่าสั้น .head()

- data\_covid.shape ใช้คำสั้นที่เก็บข้อมูลตามตัวช่วง head() ไว้เพื่อให้แสดงเพียงค่าเดียวในตาราง เดียว 5 แถวแรก int, default=5
- สามารถกำหนดจำนวนของค่าที่ต้องการให้แสดงโดยใส่ตัวเลข เช่น
- data\_covid.head(10) จะแสดงค่าเดียวเป็นผลลัพธ์ในตาราง 10 แถว

10

ค่าสั้น .shape

- data\_covid.shape
- ใช้คำสั้นที่เก็บข้อมูลตามตัวช่วง shape ไว้ตรวจสอบขนาดของข้อมูลผลลัพธ์จะได้ (839771, 11)
- หมายความว่า น้ำเงินสูงที่สุด 839,771 และมีค่าบันทึก 11 ค่าบันทึก

11

การซื้อกำหนดในข้อมูลตารางแบบ basic

- ใช้คำสั้นที่เก็บข้อมูลในค่าบันทึกของการ
- data\_covid['province\_of\_onset']

```
data_covid['province_of_onset']
0    นนทบุรี
1    นนทบุรี
2    นนทบุรี
3    นนทบุรี
4    นนทบุรี
5    นนทบุรี
6    นนทบุรี
7    นนทบุรี
8    นนทบุรี
9    นนทบุรี
...
```

12

การซื้อกำหนดในข้อมูลตารางแบบ basic

- การใช้คำสั้นและลักษณะในการซื้อยูนิฟายในแผนกและลักษณะของการ
- data\_covid['province\_of\_onset'][4]
- ผลลัพธ์จะได้ค่าบันทึกที่ 4 นั่น即 0 ในค่าบันทึก province\_of\_onset
- "กรุงเทพมหานคร"

13

การซื้อกำหนดในข้อมูลตารางแบบ .iloc[]

- ใช้คำสั้นและลักษณะการซื้อยูนิฟายในรูปแบบ numpy array หรือ matrix จะใช้คำสั้นเพื่อซื้อยูนิฟายต่อการ เช่น
- data\_covid.iloc[4, 9]
- ผลลัพธ์จะได้ค่าบันทึกที่ 4 ค่าบันทึกที่ 9 (ในชุดของ 11 ค่าบันทึก คือที่ 9) นั่น即 0 คือค่าบันทึก province\_of\_onset
- "กรุงเทพมหานคร"

14

Table slicing การเลือกข้อมูลเฉพาะคอลัมน์ต่อการ

- กรณีต้องดึงเฉพาะค่าที่ต้องการมาเก็บไว้กับแบบที่เรียกว่า "ร่อง"
- smaller\_table = data\_covid[['announce\_date', 'province\_of\_onset', 'risk']]
- หมายความว่า เลือกค่าต่อไปนี้เช่น announce\_date, province\_of\_onset, risk ในข้อมูลที่เก็บอยู่แล้ว
- แบบที่ใช้ชื่อร่อง

15

Table slicing การเลือกข้อมูลเฉพาะค่าว่าที่ต่อการแบบง่าย

- การเลือกข้อมูลเฉพาะค่าว่าที่ต่อการแบบง่าย คือการลงชื่อและต้องการมาในรูปแบบ array แต่การนี้ไม่ใช้ รูปแบบนี้ในตัวเดียว
- data\_covid.iloc[115,:]
- หมายความว่า
- 1:5 คือเลือกค่าที่ต้องบันทึกที่ 1 ไปจนถึงบันทึกที่ 4
- :: คือเลือกค่าทุกค่าที่มีอยู่ ดังนั้น
- data\_covid.iloc[115,:] คือเลือกข้อมูลที่เก็บไว้ในค่าว่าที่ 1 ไปจนถึงบันทึกที่ 4 และ เลือกทุกค่าที่มี

16

Table slicing การเลือกข้อมูลเฉพาะค่าว่าที่ต่อการแบบ advanced

- การเลือกข้อมูลเฉพาะค่าว่าที่ต่อการแบบ advanced คือการใช้ logic query ใน การเลือกข้อมูล
- data\_covid[(data\_covid['province\_of\_onset'] == 'นนทบุรี')]
- หมายความว่า เลือกข้อมูลที่อยู่ในค่าว่าที่ data\_covid ที่มีค่าต่อไปนี้เช่นค่าที่ province\_of\_onset และต้องการค่าทุกค่าที่มีอยู่โดยค่าบันทึก province\_of\_onset เป็นตัวค่าของค่านั้น

17

วิธีการเลือกข้อมูลเฉพาะค่าว่าที่ต่อการแบบ advanced

- การใช้งานของ logic query คือจะมีการแสดงผลลัพธ์การแบบ advanced
- คือ การใช้เกณฑ์ต่อการ และได้ list True/False ที่หากค่าที่ซึ่งบันทึกมาเท่ากับ logic ใน列表 นั้นๆ ค่าจะเข้าใช้ logical expression (True/False) เช่น
- ส่วนการตั้งค่าของค่าต่อการ
- right\_rows\_covid = data\_covid.iloc[18,:]
- right\_rows\_covid
- หมายความว่า เลือกค่าที่อยู่ในตัวค่าว่าที่ data\_covid มากที่ 0 ถึงแม้ที่ 7 ทุกค่าจะเก็บไว้ในตัวค่าว่าที่ right\_rows\_covid

18

- การทํางานของภาษาเรือกข้อมูลและพัฒนาเด็กที่ต้องการแบบ advanced
  - ใช้เมต้าค่าที่สามารถรับค่า True และ False หรือใช้ภาษา Python กำหนดเองได้
  - `df.iltch_rows_cond([True, True, False, True, True, True, False])`
  - ผลลัพธ์จะได้เป็นชุดข้อมูลตามมา True/False ที่มีตัวอักษร [0, 1, 2, 3, 4, 5, 6]
- เขียนเมต้าค่าให้เกินกว่า 1 ตัว อาจ logical expression ที่มีผลลัพธ์เป็น Boolean ให้กับตัวนี้จะไม่สามารถใช้ได้ (เช่น `True/False`) ให้คืนมาเป็นชุดของตัวเลขที่สามารถใช้กับตัวนี้ได้ (เช่น `0, 1, 2, 3, 4, 5, 6`) ไม่ต้องห่วงเรื่องปัญหา

19

กារสร้าง list ของ logical expression

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

20

น้ำ list ของ logical expression ที่สร้างไว้ใช้งาน

- สูงเมื่อนำมาใช้งานแล้วก็จะมีผลลัพธ์ทางคณิตศาสตร์ของการบันทึกการใช้ logic query ในการเลือกข้อมูลนั้นมา
- `weight_rows covid[weight_rows covid['province_of_onset'] == 'เชียงใหม่']`
- ผลลัพธ์จะได้รับข้อมูลที่ถูกเรียงลำดับตามค่าที่ตั้งไว้ `province_of_onset` เช่น เป็น กรุงเทพมหานคร ภูเก็ตฯ 0, 1, 2, 3, 4, 5, 6

21

22

23



- គេងលុយនូវបានការប្រមិជនសំខាន់ខ្ពស់
- គេងលុយនូវបានការប្រមិជនសំខាន់ខ្ពស់
- ការចិត្តរកកុំព្យូទ័រអាជីវកម្ម bias ខ្លោយ
- ការចិត្តរកកុំព្យូទ័រអាជីវកម្ម bias ខ្លោយ
- ការចិត្តរកកុំព្យូទ័រអាជីវកម្ម bias ខ្លោយ
- ការចិត្តរកកុំព្យូទ័រអាជីវកម្ម bias ខ្លោយ

24

The diagram shows a large iceberg floating in dark blue water. The visible portion above the surface is labeled "Human Bias". Below the surface, the submerged part is labeled "Systemic Bias". A small circular icon in the top right corner contains a white elephant.

25

กรุณารีบศึกษาและตัวอย่างจากโลเกดแห่งความเป็นจริง

- นาฬิกาของคุณมีความแม่นยำและถูกต้องจริงๆ แต่ถ้าหันกลับเข้าไปดูแล้วจะพบว่ามันไม่เหมือนที่เราเห็น
- วิเคราะห์ตัวเองให้เข้าใจการคิดเชิงลึกได้เพียงรู้สึกที่มาในใจ นั่นคือด้านจิตใจรวม และวินัยคุณภาพที่อาจขาดทิ้ง
- การสอนให้เด็กเรียนรู้โดยการฝึกซ้อมหรือท่องจำเพื่อให้เข้าใจดี แต่เด็กจะไม่สามารถนำความรู้ที่ได้มาใช้ได้จริง
- <https://www.prolific.com/resource/hacking-a-bias>

26

## Homework class period 6

27

Class period 7

บทที่ 5 การดำเนินการเบื้องต้นกับข้อมูล 1  
Pandas 101 part2

1

เฉลย Homework class period 6

```

• คำนับ ถ้ามีชื่อของสตรี และชาย ของจังหวัดทั้งหมด
• this_data = data_covid[['sex','age','province_of_onset']]
• this_data
• เมื่อพิมพ์ดูจะพบว่ามีชื่อของจังหวัดทั้งหมด
• female = this_data[this_data['sex']=='female']
• เมื่อกดเมาส์ที่ชื่อ female บนหน้าจอ เห็นว่าเป็น female
• female['age']
• เมื่อพิมพ์ชื่อ female บนหน้าจอ female แสดงว่ามีค่าของเพศหญิงทั้งหมด

```

2

เฉลย Homework class period 6

```

• งานนี้บันทึกไปเพื่อภาษาญี่ปุ่น
• sum = 0
• N = 0
• for a in female['age']:
    if a > 0:
        sum += a # sum = sum + a
    N += 1
• print(f'年齢の平均は{sum/N}歳です')
• สำหรับตัวอย่าง sum=0 และ N=0 ให้พิมพ์ในการแก้คู่คิดการงานในกระบวนการและระบบเมื่อรอนอนุญาต
• โดย ผล 0 นี้เป็นภาษาญี่ปุ่น และ N=0 นี้เป็นภาษาญี่ปุ่นด้วยครับ

```

3

เฉลย Homework class period 6

```

• sum = 0
• N = 0
• สำหรับตัวอย่าง sum=0 และ N=0 ที่พิมพ์ในการแก้คู่คิดการงานในกระบวนการและระบบเมื่อรอนอนุญาต
• ให้ sum จะเท่ากับ 0 และ N จะเท่ากับ 0 ด้วยครับ
• for a in female['age']:
    sum += a # sum = sum + a
• สำหรับภาษาญี่ปุ่นที่จะสอนกันในวันเดียว
• sum ไม่ได้เท่ากับ 0 แต่จะเท่ากับ 0 ด้วยครับ
• if a > 0:
    N += 1
• ผลจะได้ 0 นี้เป็นภาษาญี่ปุ่น 0 นี้จะเขียนว่า ญี่ปุ่นค่านอนต่อตัวเดียว เช่น 0 ญี่ปุ่น

```

4

เฉลย Homework class period 6

```

• sum += a # sum = sum + a
• N += 1
• สำหรับภาษาญี่ปุ่น หมายความว่า ญี่ปุ่นที่จะสอน ขบวน 1 รอบก็จะสามารถพิจารณาการรวมที่ “รวมกัน” คือไม่ใช่ “เพิ่ม” คือภาษาญี่ปุ่น
• บันทึก N มากกว่า 1 เพื่อเชื่อมขั้นตอนที่ถูกต้อง
• print(f'年齢の平均は{sum/N}歳です')
• บันทึก sum และ N มากกว่าเพื่อทำตามเช็ค ผลลัพธ์จะได้

```

5

การจัดการ Missing Value

- มีข้อดี 3 แบบ
  1. ลบ record ที่เป็น missing
  2. แทนที่ ค่า missing ด้วย ที่เหมาะสม mean, default, category-unknown
  3. ใช้ ค่าจาก column นึงๆ ซึ่งประมาณค่า ค่านั้น column ที่หายไป (regression, deep learning, etc.)

6

ลบ record (dropna)

```

• missing = None, NaNot authorized, NaN (not a number)
• dropna() เป็นฟังก์ชันในการลบข้อมูลที่เป็น缺失值หรือไม่มีค่า ค่อนข้างน่าจะ
• data_covid.shape ผลลัพธ์จะได้รับขนาดของ data_covid
• (839771, 11)
• data_covid.dropna().shape ผลลัพธ์จะได้รับขนาดของ data_covid ที่ลบ掉ข้อมูลที่มีค่าเป็นNone
• (599988, 11)

```

7

การใช้งาน .dropna()

```

• สามารถเลือกลบข้อมูลที่เป็น None เนื่องจากมีบัญชีของการรักษา แต่ต้องเลือกหากต้องลบขบวน
• this_data = data_covid[['sex','age','province_of_onset']]
• this_data.shape (839771, 3)
• this_data.dropna().shape ผลลัพธ์จะได้
• (599988, 3)
• บันทึกว่าเมื่อพิมพ์ data_covid.dropna().shape ที่เป็นข้อมูลทั้งหมด (599988, 11)
• ข้อมูลที่เลือกจะลบเมื่อต้องการใช้งานจะมีชื่อว่าข้อมูลมากกว่า

```

8

การใส่ “ดูแลเพื่อรักษา”

```

• this_data.dropna()
• print(this_data.shape) ผลลัพธ์
• (839771, 3)
• ดูใน size ของข้อมูลที่มีค่าเป็น None เมื่อจากไปไม่ได้รักษาขบวน เนื่องจาก
• This_data_dn = this_data.dropna()
• print(This_data_dn.shape ผลลัพธ์
• (674906, 3)

```

9

Parameter: inplace ของ .dropna()

```

Pandas.DataFrame.dropna

# inplace จะเป็นการเพี้ยนแปลงในตารางเอง
# ให้เก็บข้อมูลเดิมและเพิ่มข้อมูลใหม่เข้าไปด้วย

• this_data.dropna(inplace=True)
• print(this_data.shape)
• (674906, 3)


```

10

Parameter: subset ของ .dropna()

```

Pandas.DataFrame.dropna

# subset คือเป็นการเลือกเฉพาะคอลัมน์ที่ต้องการลบเมื่อต้องการให้เป็น None เนื่องจากค่าเดิมจะถูกแทนที่
# ให้เก็บข้อมูลเดิมและเพิ่มข้อมูลใหม่เข้าไปด้วย

• this_data = data_covid[['sex','age','province_of_onset']]
• this_data.shape
• (639772, 3)

• this_data.dropna().shape ขนาดของตารางจะเปลี่ยนเป็น None 600000
• (674906, 3)

• this_data.dropna(subset=['age']).shape
• ขนาดของตารางจะเปลี่ยนเป็น subset ของค่าที่ไม่ใช่ None และ subset ที่ค่าเดิม
• (763606, 3)


```

11

แผน missing ด้วยการเพิ่มมาส์ก .fillna()

```


• แผน missing เป็นค่าเดิมที่จะไปแทนค่า missing หรือว่า None ที่ทางผู้สอนให้
• this_data = data_covid[['sex','age','province_of_onset']]
• this_data.fillna('None', inplace=True)
• this_data.shape
• (639772, 3)

• ตรวจสอบว่า ให้เก็บข้อมูลเดิมที่ไม่ใช่ None
• ให้เก็บข้อมูลเดิมที่ไม่ใช่ None
• ให้เก็บข้อมูลเดิมที่ไม่ใช่ None
• ให้เก็บข้อมูลเดิมที่ไม่ใช่ None
• แผน missing ของ province_of_onset ให้เป็น None ที่ใช้ 'None'
• แผน missing ของ province_of_onset ให้เป็น None ที่ใช้ 'None'


```

12

แผน missing ด้วยค่าที่เหมาะสม .fillna()

```


• ตามที่เราตรวจสอบหาก None ถูกแทนที่ด้วยค่าที่เดิมก็ไม่ได้
• this_data_updated[this_data_updated['sex'] == 'None']
• ผลลัพธ์จะเห็นว่ามีค่าในเบื้องต้น sex ที่ถูกแทนที่ด้วย 'None'


```

13

การใช้ logical expression จากข้อมูลตารางอื่น

```


• data_covid[this_data_updated['province_of_onset']=='None']

• จะเห็นว่าในเบื้องต้นที่เก็บข้อมูลใน logical expression มากัดเพิ่ม this_data_updated ซึ่งจะใช้ในการกรอง
  ของตัวเอง data_covid
• ถ้าจะใช้ชุดข้อมูลอื่นๆ ให้ต้องรู้ว่าต้องใช้คอลัมน์ไหน
• บันทึก 2 ค่าที่เราใช้ในบรรดาค่าที่เก็บมาแล้วและมาต่อท้ายกับตัวเดิมที่เราไม่ใช่ None และถ้าต้องการลบ
  เนื้อหาไม่ใช่ None ให้ลบ


```

14

การวนลูป record ในตาราง .iterrows()

```


• แผนmissing เป็นค่าเดิมที่จะมาในแบบนี้มาแทนที่ค่าที่ถูกแทนที่ในกรอบ
• 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'] == 'None'):
•     print(each_row)
• ตรวจสอบว่า
• ให้ลงมาเป็นแบบนี้มาแทนที่ค่าเดิมที่ถูกแทนที่ในกรอบ
• สำหรับคนที่ไม่คุ้นเคยกับ iterrows() ให้ลองกด shift+ctrl+p ก็จะมีคำแนะนำอยู่ในกรอบ age==20 และ
  province_of_onset==None


```

15

1

Quiz สุนเชื่อ (5 ก้าวที่)

- ให้รับ input เซ็ตของข้อมูลในตารางเป็นตัวอักษร .loc ชื่อเป็น numpy array หรือ matrix ให้แสดงค่าของ column เป็นตัว `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'] == 'สมุทร')`
- `print(each_row)`

2

```
ก ารร บก วนช รุ ณ ประบ กษา (pandas dataframe) เป น numpy array หร อ 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] == "swatoh"):
        print(each_row)
    print(this_data.iloc[each_row,:])
```

(三)

การวนซูป

- `for each_row in range(this_data.shape[0]):`
- หาค่าที่ต้องพิมพ์มาเรียกแม่ก็ให้ไปเขียนที่ `each_row` โดยจะวนรอบที่ห้ามไม่สามารถนับได้  
▪ `this_data.shape`  
▪ `(839771, 3)`  
▪ `this_data.shape[0]`  
▪ `839771`
- `range(this_data.shape[0])` คือช่วงตัวที่ต้องมาเรียกแม่ก็ต้องเขียนว่า “`for`” ขึ้นมา  
▪ `[0, 1, 2, 3, ..., 839771]`

4

5

- `print(each_row)`
- `print(this_data.iloc[each_row,:])`
- เมื่อเราได้รับมาแล้ว  
• `print each_row` ก็ คือตัวชี้ที่บอกว่า แถว  
• `print the_data.iloc[each_row,:]` ก็คือ ข้อมูลในบรรทัดที่ คุณต้องการ

6

Quiz ในห้อง (15 นาที)

- ตั้งค่ารายละเอียดพัฒนาปี 2021 announce\_date ในปี 2021
- Hint
- นำสูตร index ของปี 2021
- ตั้งค่ารายละเอียดพัฒนาปี 2021

7

```
    * 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 วนลูปและร้างเงื่อนไข

- `TF = list()`
- วนลูป เอา value ที่เป็นชุด TF เพิ่มเข้าไปใน列表 ตัวอย่าง `value = 'พัฒนาการกรุงเทพ'`
- `for each_row in data_covid.iterrows():`
- นำข้อมูลที่ต้องการมาใส่进去 `data_covid คือตัวแปรที่เก็บในลูป เช่น row`
- `if watch_row[1]['announce_date'].split('/')[2] == '2021':`
- ทำการ append ให้กับตัวแปรที่ต้องการใส่进去 `announce_date, date, เทศกัน 2021 โดยใช้คำสั่ง .split('/')[2]` ==> เผยแพร่ให้กับประเทศไทยเรียบร้อย
- วนลูปนัดเดียวเพื่อต้องดึง `announce_date, date` วนลูปนัดเดียว 3/1/2021 คืนที่เดียว เป็น
- ตัวบันทึกตัวแปรที่ต้องการ / เป็นชุดหนึ่งที่ 2 รายการ

1

พิชณ True False เข้าไปใน List

```

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

• ล้วนๆจะมี True ทั้งหมด
• หากว่า ถ้าข้อมูลเป็นปีที่ปัจจุบัน announce_date เดือน 2021 ให้เก็บ True เข้าไปใน list ให้เก็บ False
• ผลลัพธ์จะได้ list True False ตามลำดับของข้อมูลจาก data_covid ที่เก็บ True เท่านั้น

```

10

ผลลัพธ์

```

data_covid[TF].head()
• ให้ผลลัพธ์เป็น list true false ซึ่งค่ากราบน้ำเงินเดือนพฤษภาคม 2021
• ผลลัพธ์ใช้การตัดสูตรและแยกตามเดือนที่ปัจจุบันของ announce_date มากกว่า 2021

```

11

Function ที่ใช้ใน pandas

- describe() คำนวณหาผลิตภัณฑ์ของข้อมูลที่เป็นตัวเลข
- mean() คำนวณผลเฉลี่ยของส่วนต่อไปไม่มีmissing
- isnull() ตรวจสอบข้อมูลที่ missing (none)

12

.describe()

```

data_covid.describe()
• ใช้คำสั่งคำนวณค่าทางสถิติของข้อมูลที่เป็นตัวเลข
• ในนี้ข้อมูลที่เป็น missing หรือ none น่าจะเป็น
• (sexและageที่มาเป็นnone ให้ตั้งเป็น缺)

```

13

.mean()

```

• ใช้คำสั่งคำนวณค่าเฉลี่ยของข้อมูลที่ไม่มีmissing (ยกเว้นข้อมูลที่มีค่าเป็น none ให้ตั้งเป็น缺)
• data_covid[data_covid['sex']=='female'].mean()
• ผลลัพธ์
• 34.96232702130538

```

14

.isnull()

```

data_covid.isnull()
• ใช้ตรวจสอบข้อมูล missing ในข้อมูลตาราง
• True คือ missing (ค่าว่าง)
• False คือไม่เป็นค่าว่าง

```

15

1

.isnull().any()		
P	Q	P OR Q
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

data_covid.isnull(). any()	
No.	False
country	False
Notified date	True
age	True
gender	True
nationality	True
place of isolation	True
risk	True
source of onset	True
district_of_onset	True
region	True

2

- all() សម្រាប់បង្កើតរបស់បញ្ជីបានត្រូវបាន True False នៅក្នុងខាងក្រោមនេះ តាមរយៈការបញ្ជាផ្ទាល់ទីតាំងនៃអំពីការិយាល័យដែលបានរកឃើញ។ ហើយ

```
data_covid19[{}].all()
```

P	Q	P AND Q
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

三

การซัก'(ในตารางของ .isnull()

- ไม่สามารถตัดง่าย basic ที่สุด อย่างเช่น
- `data_covid['No.'].isnull()`
- `data_covid.iloc[0,0].isnull()`

จุดเด่นที่สำคัญอย่างมาก คือ

- พิมพ์คำว่า `isnull` เท่านั้น
- `data_covid.iloc[1:,0].isnull()`
- เป็นคำสั่งให้เก็บเฉพาะช่วงของข้อมูลที่ 0 ถึง 1
- และจะได้รับ ค่าที่ต้องการ

**pandas.isnull**

```
pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/master/titanic.csv')
```

**参数:** `values (obj)`

Desired missing values for an array-like object.

This function takes a scalar or array-like object and indicates whether values in the array are missing (NaN) or not. In the case of strings, NaN is determined by the empty string.

**Parameters:**

- `obj : scalar or array-like`  
Object to check for null or missing values.

**返回值:**

`bool or array-like object`

For scalar input, returns a boolean. For array inputs, returns a boolean array indicating whether each corresponding element is missing.

4

การ合併ตารางสองใบ X และ y

- คุณภาพ y คือ เรียง records (เรียงจากล่างไปบน)
- คุณภาพ x คือ เรียง columns (เรียงจากซ้ายไปขวาตามที่ต้องการ)

df_name		df_group		df_name_group	
name	value	group	value	name	value
1	redka	1	redka	1	redka
2	blueka	2	blueka	2	blueka
3	greenka	3	greenka	3	greenka
4	yellowka	4	yellowka	4	yellowka

**df\_name**

**df\_group**

**df\_name\_group**

5

- การคัดแยก Y เป็นการนำมาและเก็บข้อมูล 2 ภาระความร่วมหรือเดือน
- เพื่อให้เราสามารถใช้งานต่อไปในหน่วยงานต่อๆ กันได้
- `dataXK = data_covid[(data_covid['province_of_onset'])=="มุกดาหาร"]`
- `dataUUD = data_covid[(data_covid['province_of_onset'])=="มหาสารคาม"]`
- `dataNan = data_covid[(data_covid['province_of_onset'])=="น่าน"]`
- ขั้นตอนนี้จะใช้ `pd.concat` ลดลงให้เป็นตัวอย่างและเก็บการคัดกรองแล้วไปบันทึก `dataXK` `dataUUD` `dataNan` ใช้สำหรับต่อไปในการเรียนรู้ของเรามากขึ้น
- `import pandas as pd`
- `dataYisan = pd.concat([dataXK,dataUUD,dataNan])`
- `dataYisan`

6

- การคัดกรองแทน x สามารถทำได้ 2 แบบ
- 1. ข้อ 2 ค่ารวมต่อกันเป็นสอง (merge)
- 2. เสิร์ฟจากเพิ่มเฉพาะบาง column ก็ได้

7

• การรวมข้อมูลที่ต้องการของ records ทั้งหมด ตามรายละเอียดของแต่ละ record

• เพิ่มค่าการคำนวณที่ต้องการ

```
• data_province = data_covid[["No.", "announce_date", "province_of_onset"]]
data_human = data_covid[["No.", "age", "sex", "nationality"]]
# ใช้เพื่อรวมเพื่อนำมาใช้ในการคำนวณ
merge(data_province, data_human, by="No.", suffixes=("_Province", "_Human"))
full_table1 = data_human.merge(data_province)
full_table1 = full_table1.drop(['id'], axis=1)
full_table1.head()
```

• ใช้ชื่อรูปแบบที่ announce\_date และ province\_of\_onset อยู่ในรูปแบบเดียวกัน

8

- ให้ใช้ command `sort_values()` ของ pandas ในการเรียงลำดับค่าตามต้องการ โดยใช้ชื่อคอลัมน์ที่ต้องการเรียงลำดับเป็น parameter ของ command `sort_values()` เช่น
- `data_human = data_covid[['No.','age','sex','nationality']]`
- `data_human2 = data_human.sort_values('age')`
- `data_human2`
- ผลลัพธ์จะได้ตารางที่เรียงลำดับตามค่าใน column 'age' ของ default จะเรียงจากน้อยไปมาก เช่น

5

จัดองค์การฐานข้อมูล.mergel()

```

• merge() ช่วยในการต่อเนื่องกันให้เป็นหนึ่งเดียว
• data_human2_renamed = data_human2.rename(columns={"No.": "patientNumber"})
• data_human2_renamed.merge(data_province)
• ผลลัพธ์: MergeError: No common columns to perform merge on.
• ต้องระบุตัวแปรที่ต้องการ merge และตัวแปรที่ต้องการใช้เป็น index เพื่อ merge ให้ตรงกับชื่อในตัวแปรที่ต้องเป็น index ของแต่ละชีวิญ หลังจากจัดเรียงเป็น primary key
  
```

10

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

```

• full_table3 = data_human2_renamed.merge(data_province, left_on="patientNumber",
right_on="No.")
• ถ้าเรา merge ระหว่างคอลัมน์ที่ต้องการต่อเนื่องกัน ตามนี้เป็น left_on ก็จะต้องตั้งค่าให้เป็น index ของคอลัมน์ที่ต้องการต่อเนื่องกัน record ของคอลัมน์ที่ต้องการต่อเนื่องกันจะต้องตั้งค่าให้เป็น full_table
• full_table3.head() จะดูว่า 2 ตารางที่ต่อมาเป็นตัวต่อตัวที่ต้องการต่อเนื่องกัน
  
```

patientNumber	sex	age	provinces	patientNumber	sex	age	provinces
1	Male	40	Thailand	1	Male	40	Thailand
2	Male	40	Thailand	2	Male	40	Thailand
3	Male	40	Thailand	3	Male	40	Thailand
4	Male	40	Thailand	4	Male	40	Thailand

11

การสร้างคอลัมน์ใหม่ด้วย pandas

```

• คุณจะต้อง import pandas ในภาษา Python อีกครั้ง คือ ตามอย่างที่เคยสอนให้เราแล้ว
• สำหรับส่วนของคอลัมน์ที่ต้องการเพิ่มต้องตั้งชื่อ record ของ record
• ตัวชี้ column [num] = (list ที่ต้องบันทึกที่ต้องกันกับจำนวน record ของ record)
  
```

```

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

12

ตัวอย่างการสร้างคอลัมน์ใหม่ด้วย pandas

```

• data_human2_renamed['num'] = range(data_human2_renamed.shape[0])
• สร้าง list ของค่าที่ record ของต้องต่อท้ายกับ data_human2_renamed ไฟล์ shape[0] ระหว่างค่าบันทึกและตัวชี้ num ของต้องบันทึกใน data_human2_renamed
• data_human2_renamed แสดงผลลัพธ์
  
```

patientNumber	sex	age	provinces	num
1	Male	40	Thailand	0
2	Male	40	Thailand	1
3	Male	40	Thailand	2
4	Male	40	Thailand	3

13

.map() เดือยมาเพิ่มเฉพาะบาง column

```

• data_human2_renamed['patientNumber'].map(data_covid.set_index('No.')['risk'])
• แปล หมายความว่า index ของตารางคือการตั้งค่าต่อไปนี้จะถูกตั้งค่าเป็นตัวต่อตัว
• ในตัวต่อตัวนี้ ตัวต่อตัวนี้จะต้องมีค่าในตัวต่อตัวนี้เป็น patientNumber ของ ตารางคือ record ของต้องบันทึก ต้องเป็น index และ ตัวต่อตัวนี้จะต้องบันทึก ต้องเป็น index ของต้องบันทึก
  
```

No.	risk
1	Low
2	Medium
3	Medium
4	High

14

การใช้ .map() ต่อตาราง

```

• data_human2_renamed['detail'] = data_human2_renamed['patientNumber'].map(data_covid.set_index('No.')['risk'])
• ถ้า ต้อง แก้ ค่า ที่ต้องการ map ให้กับตัวต่อตัวนี้ต้อง ต้องตั้งค่า data_human2_renamed
• data_human2_renamed แสดงผลลัพธ์
  
```

patientNumber	sex	age	provinces	detail
1	Male	40	Thailand	Low
2	Male	40	Thailand	Medium
3	Male	40	Thailand	Medium
4	Male	40	Thailand	High

15

Homework class period 9

```

• ตรวจสอบว่าเงิน ที่ต้องเป็น sex เป็น missing หรือไม่
• ลบบันทึกที่ record นี้บันทึกเป็น missing (ถ้าไม่ต้องลบ ให้ลองหาว่ามีบัญชีดังนี้)
  
```

16

1

2

คำสั่งที่ใช้สำหรับกลุ่มเพื่อของ .groupby()																									
• <code>groupby</code>																									
• <code>count()</code> ใช้คำสั่งน้ำหนักในการนับจำนวนของแต่ละบุคคลในช่วงเวลา	groupby ที่ต้องห้ามเก็บผลลัพธ์ในตัวแปร																								
• <code>mean()</code> คำสั่ง mean ในตัวแปรจะไม่ถูกห้าม (ยกเว้นค่าบันทึกบันทึก)																									
• <code>max()</code> คำสั่ง max ที่บันทึกและค่าบันทึกอย่างเดียว (ยกเว้นค่าบันทึกบันทึก)																									
<table border="1"> <thead> <tr> <th>Summary operations</th> <th>Agg key operations</th> <th>More complex operations</th> </tr> </thead> <tbody> <tr> <td>mean</td> <td>as_index</td> <td>aggfunc = 'mean'</td> </tr> <tr> <td>median</td> <td>as_index</td> <td>aggfunc = 'median'</td> </tr> <tr> <td>min</td> <td>as_index</td> <td>aggfunc = 'min'</td> </tr> <tr> <td>max</td> <td>as_index</td> <td>aggfunc = 'max'</td> </tr> <tr> <td>prod</td> <td>as_index</td> <td>aggfunc = 'prod'</td> </tr> <tr> <td>describe</td> <td>as_index</td> <td>aggfunc = 'describe'</td> </tr> <tr> <td>corr</td> <td>as_index</td> <td>aggfunc = 'corr'</td> </tr> </tbody> </table>		Summary operations	Agg key operations	More complex operations	mean	as_index	aggfunc = 'mean'	median	as_index	aggfunc = 'median'	min	as_index	aggfunc = 'min'	max	as_index	aggfunc = 'max'	prod	as_index	aggfunc = 'prod'	describe	as_index	aggfunc = 'describe'	corr	as_index	aggfunc = 'corr'
Summary operations	Agg key operations	More complex operations																							
mean	as_index	aggfunc = 'mean'																							
median	as_index	aggfunc = 'median'																							
min	as_index	aggfunc = 'min'																							
max	as_index	aggfunc = 'max'																							
prod	as_index	aggfunc = 'prod'																							
describe	as_index	aggfunc = 'describe'																							
corr	as_index	aggfunc = 'corr'																							

(1)

เฉลย Homework class period 9 ด้วย groupby()

- ร่วมกันใช้คำสั่งที่เก็บใน sex เป็น คอลัมน์ ที่ต้องการ
- `data_covid['sex'].isnull()`
- ตรวจสอบข้อมูลที่ไม่มี sex และ sex ที่เป็น list logical expression True(missing)/False(non missing)
- `missing_sex = data_covid[data_covid['sex'].isnull()]`
- นำ list logical expression ของ missingsex ไปตัดต่อมาในตาราง records ที่มี index เก็บ sex เป็น missing และเก็บตารางที่เหลือ ผู้ที่ไม่เป็น missing sex
- `missing_sex` จะเป็นตารางที่ได้จากการตัดต่อ records ที่ไม่เก็บ index' sex เป็น missing

4

ເສີມ Homework class period 9 ດ້ວຍ groupby()

- ສ່ຽງກຳນົດ record ທີ່ບໍ່ມີຂໍ້ມູນ missing ໃຫ້ groupby ແລະ describe() ອຸກການນີ້ຈະອະນຸມັດການກຳນົດ sex ມີ missing ໂຄງຮ່າງຄວາມດັ່ງນີ້
- 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()

5

6

- ใช้ Dictionary หรือ pd.DataFrame
- ข้อมูลทางการเงิน เช่น หุ้นและผลตอบแทน record ที่สามารถนำมายัง dictionary ให้ Index หนึ่งเป็นชื่อหุ้นและ value เป็นผลตอบแทน record ดังนี้
- records = [{"account": "Alpha Co.", "Jan": 120, "Feb": 200, "Mar": 140},  
 {"account": "Beta Co.", "Jan": 200, "Feb": 120, "Mar": 215},  
 {"account": "Blue Inc.", "Jan": 150, "Feb": 190, "Mar": 195}]
- records\_df = pd.DataFrame(records)
- records\_df[["account", "Feb", "Mar"]]

7

- แบบ List กับ pd.DataFrame.from\_records()
- ใช้ pandas สร้าง DataFrame แบบ 2 ตัว
- ฟังก์ชันนี้ต้องมี input value เป็นรูปแบบ record บิ๊ก เล็กๆ ก็ได้ ไม่ว่าจะเป็น list หรือ dict แต่ record ต้องเป็น array
- ตัวอย่างที่ 1 ใช้ record ของข้อมูลในรูปแบบ list ในการสร้าง DataFrame
- กรณีที่ 2 ใช้ record ของข้อมูลในรูปแบบ dict ในการสร้าง DataFrame
- การตั้งชื่อ
- df = pd.DataFrame.from\_records(awslist, columns=aws2)
- df

	Bucket	File Name	Last Mod
0	awslist	000000000000	
1	Apple	200 200 200	2019-01-01 10:00:00
2	Banana	10 10 10	2019-01-01 10:00:00

List

```
sales = [{"Region": "West", "Sales": 100, "Profit": 200, "Year": 2018}, {"Region": "East", "Sales": 200, "Profit": 300, "Year": 2018}, {"Region": "South", "Sales": 150, "Profit": 250, "Year": 2018}, {"Region": "North", "Sales": 250, "Profit": 350, "Year": 2018}, {"Region": "West", "Sales": 100, "Profit": 200, "Year": 2019}, {"Region": "East", "Sales": 200, "Profit": 300, "Year": 2019}, {"Region": "South", "Sales": 150, "Profit": 250, "Year": 2019}, {"Region": "North", "Sales": 250, "Profit": 350, "Year": 2019}]

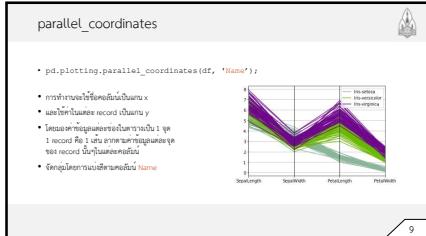
japan = [{"Category": "Electronics", "Sales": 100, "Profit": 200, "Year": 2018}, {"Category": "Clothing", "Sales": 200, "Profit": 300, "Year": 2018}, {"Category": "Food", "Sales": 150, "Profit": 250, "Year": 2018}, {"Category": "Books", "Sales": 250, "Profit": 350, "Year": 2018}, {"Category": "Electronics", "Sales": 100, "Profit": 200, "Year": 2019}, {"Category": "Clothing", "Sales": 200, "Profit": 300, "Year": 2019}, {"Category": "Food", "Sales": 150, "Profit": 250, "Year": 2019}, {"Category": "Books", "Sales": 250, "Profit": 350, "Year": 2019}]
```

8

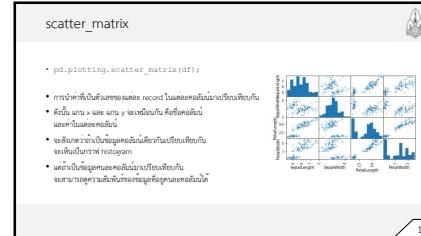
## Simple Visualization

- ให้ดูตัวอย่างการนับจำนวนข้อมูลในชีวิต ชาเขียวคือ
- <https://www.github.com/pandas-dev/pandas/master/pandas/tests/ufc/data/csv/iris.csv>
- ดาวน์โหลดมาแล้ว ลิงค์และเก็บไว้ในเดสก์ท็อป
- df = pd.read\_csv(<https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/ufc/data/csv/iris.csv>)
- df
- ผล df.groupby('Name').count() ดูว่าชื่อของกลุ่ม

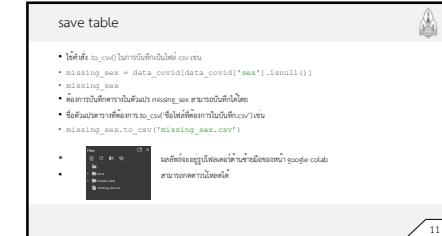
8



10



11



12

Class period 11  
บทที่ 6 การแสดงผลการวิเคราะห์ข้อมูล  
Visualize\_Table\_Data

1

หน้าจอ parallel\_coordinates

- เข้า google code ในชื่อ pandas-parallel-coordinates หรือ
- import pandas as pd
- df = pd.read\_csv("https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/io/data/csv/iris.csv")
- df
- pd.plotting.parallel\_coordinates(df, 'Name')

2

ปรับ parameter: color

- หากต้องการแสดงผลลัพธ์ตามสีของค่าในช่วง ให้แก้ parameter color ด้วย list ของสี html color, คือเช่นนี้ RGB value (rgb) หรือจะกำหนดให้เป็นสี hex color
- pd.plotting.parallel\_coordinates(df, 'Name', color=['#FF0000', '#00FFFF', '#E6CCFF'])
- สีจะถูกแทนที่ด้วยสีที่กำหนด

3

ตัวอย่างสี

- สามารถเลือกสีได้จากสีที่มีอยู่หรือไม่
- R G B W C M Y K
- pd.plotting.parallel\_coordinates(df, 'Name', color=['r', 'g', 'b'])

4

RGB value (r,g,b)

- สามารถเลือกสีได้จากสีที่มีอยู่ในช่อง RGB ที่จะมีตัวเลขอารบิกสามตัวคือสีแดง สีเขียว และสีฟ้า
- เช่น 0.3, 0.5, 0.2 เป็นสีแดง RGB จะเท่ากับ 0.3 0.5 0.2
- pd.plotting.parallel\_coordinates(df, 'Name', color=((1,0,0), (0,0.5,0), (0.5,0,1)))

5

ปรับ parameter: colormap

- เข้าไปใน Choosing Colormaps ใน Matplotlib ใน google ก็จะ
- pd.plotting.parallel\_coordinates(df, 'Name', colormap='cool')

6

matplotlib

- import packet หรือ library ที่สนใจเพื่อ Visualizetion ในการวิเคราะห์ข้อมูล
- ใช้งานได้
- from matplotlib import pyplot as plt
- from matplotlib import image
- import numpy as np
- import pandas as pd
- matplotlib จัดทำอยู่ที่ GitHub คือ https://github.com/matplotlib/matplotlib ที่จะมีรายละเอียดที่แน่นหนาและอธิบายอย่างละเอียด สำหรับผู้ที่สนใจ
- ดูใน ค่าคอมมอนที่มาพร้อมกับ pandas ให้ถูกต้องใน numpy array ก่อนเขียน matplotlib

7

แปลง data จาก pandas dataframe เป็น numpy array

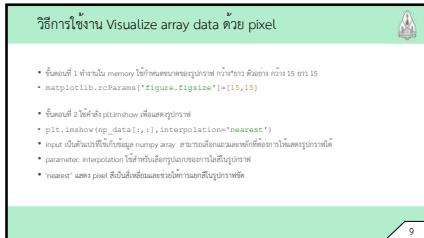
- # ดู np.array() ใน google ให้ความรู้เกี่ยวกับ pandas ที่ต้องการ ให้เก็บค่า คอลัมน์และเรียงตัวอักษร เช่น
- np\_data = df.iloc[1:-1].to\_numpy()
- ให้ลุกค่าอยู่ในชุดของ df เมื่อเราพยายามนำค่ามาเรียงตัวอักษรที่เป็นตัวเลข ค่าจะถูกเรียงตัวอักษรที่ 0 ถึง 9 ตามลำดับต่อๆ กันไป
- np\_data แสดงลักษณะที่เป็น矩陣ของข้อมูลที่เราเรียกว่า matrix of ถ้าแปลงเป็น numpy array

8

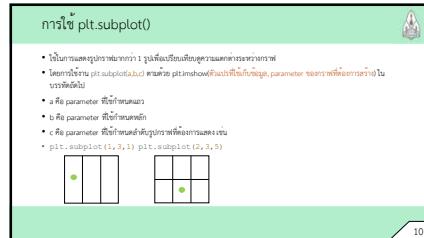
Visualize array data ด้วย pixel

- ปรับภาพมาด้วย matrix ใน numpy array ที่เรียกว่า pixel
- ค่าของแต่ละค่า คือสีของพิกเซล
- สามารถใช้รูปภาพมาแทนค่าได้
- หากต้องใช้ภาพมาแทนค่า ให้แนบไฟล์ภาพมาด้วยด้วยตัวเอง
- หากต้องใช้ภาพ 11 ให้มองหา np\_data

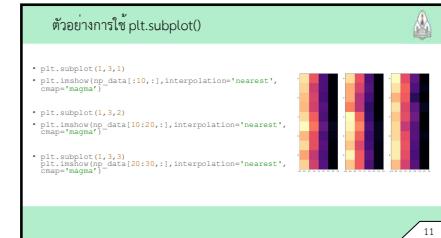
9



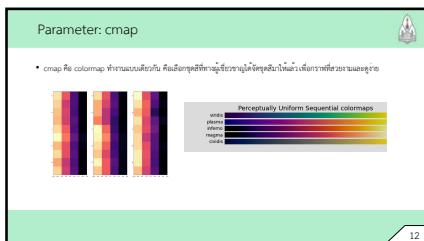
10



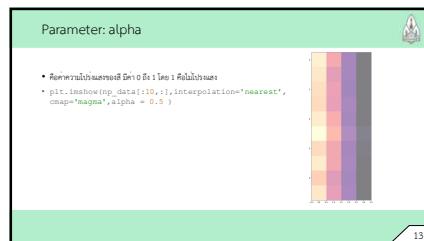
11



12



13



14

Class period 12

บทที่ ๔ การแสดงและการวิเคราะห์ข้อมูล (๒)

Visualize\_Data\_Distribution

1

Quiz2

- ใบหน้าดูดูร้ายแล้วแต่ก็ต้องเดินทางเข้าเรียน จันทร์พุธ曜 และเช้าอาทิตย์
- [https://data.gov/th/dataset/item\\_7049410f-9d68-4c79-9e94-112ca18863e2](https://data.gov/th/dataset/item_7049410f-9d68-4c79-9e94-112ca18863e2)
- ให้นำมากรอกลงในช่องและจัดเรียง ผลจะมีเม็ดเป็น missing ออก

2

การแสดงผลการวิเคราะห์ข้อมูล 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 ในไฟ Visualize\_Data\_Distribution.ipynb
- สามารถดูรายละเอียดในไฟ

3

ดอกไม้ Iris

4

Scatter 2D

- ต้องการจะพิจารณาในความสัมภาระเป็นอุป ๑ ๆ และบ่งชี้น้ำท่วมรากฟ้า เพื่อการวิเคราะห์ข้อมูลต้องใช้ 2 nomination ( nominate )
- โดยใช้ plt.scatter(df['X'], df['Y']) หรือ
- plt.scatter(df['PetalLength'], df['SepalLength'])

SepalLength	PetalLength	PetalWidth	Species
5.1	3.5	1.4	Iris-setosa
4.9	3.0	1.4	Iris-setosa
4.7	3.2	1.3	Iris-setosa
4.6	3.1	1.5	Iris-setosa
5.0	3.6	1.4	Iris-setosa
5.4	3.9	1.7	Iris-setosa
4.5	2.3	1.3	Iris-setosa
4.4	1.7	0.8	Iris-setosa
4.9	3.0	1.4	Iris-setosa
5.0	3.4	1.5	Iris-setosa
5.5	4.5	1.5	Iris-setosa
4.9	3.4	1.4	Iris-setosa
5.4	4.5	1.5	Iris-setosa
5.7	4.4	1.5	Iris-setosa
5.1	4.0	1.3	Iris-setosa
5.7	4.4	1.4	Iris-setosa
5.1	4.3	1.3	Iris-setosa
5.4	4.5	1.5	Iris-setosa
5.1	4.0	1.3	Iris-setosa
5.0	3.4	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.2	3.4	1.4	Iris-setosa
5.5	4.2	1.5	Iris-setosa
5.2	3.4	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.4	4.5	1.5	Iris-setosa
5.1	3.3	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa
5.1	3.7	1.5	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.0	3.4	1.4	Iris-setosa
5.2	3.5	1.5	Iris-setosa

Class period 13

บทที่ ๔ การแสดงผลและการวิเคราะห์ข้อมูล (cont.)  
Visualize\_Data\_Distribution part2

1

quiz3 (5 นาที)

- ให้ผลลัพธ์แสดง ผลลัพธ์ กับ ชื่อชาน west
- ชื่อเบอร์ ซึ่งมีชื่อ west คือ
- ลิงค์ [https://data.gov/th/dataset/item\\_fuck318e-77f34bu2-84d1-a19210a9bd0d](https://data.gov/th/dataset/item_fuck318e-77f34bu2-84d1-a19210a9bd0d)

2

การเพิ่มตัวชี้วัดของ plot

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

3

การเพิ่มตัวชี้วัดใน plot

- สามารถเพิ่มตัวชี้วัดเพื่อแยกบันทึก record ได้โดยการเพิ่ม parameter `marker` หรือ `label`
- `plt.scatter(df2['PetalLength'], df2['SepalLength'], c=df2['Name'])`
- `plt.xlabel('PetalLength')`
- `plt.ylabel('SepalLength')`

4

การแยกข้อมูลเพื่อกำหนดลักษณะ marker แต่ละกลุ่ม

- ในการแยกข้อมูลเพื่อกำหนดลักษณะ marker และกลุ่ม จึงเป็นเพียงตัวชี้วัดและจำนวน record ของแต่ละกลุ่มเท่านั้น
- ในชุดข้อมูลนี้ จะแบ่ง 3 กลุ่มดังนี้
- Iris-setosa และ Iris-versicolor
- Iris-versicolor และ Iris-virginica
- Iris-setosa คือ [0:50] ทั้งหมด record แรก คือ Iris-setosa record ที่ 0 ถึง record ที่ 49
- Iris-versicolor คือ [50:100] ทั้งหมด record ที่ 50 ถึง record ที่ 99
- Iris-virginica คือ [100:] ทั้งหมด record ที่ 99 ถึง record สุดท้าย

5

ตัวอย่างใช้การแยกข้อมูลเพื่อ plot กราฟหัวอนกัน

- `plt.scatter(df2['PetalLength'][0:50], df2['SepalLength'][0:50], c=df2['Name'][0:50])`
- `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:])`

6

Parameter: label และ color ของ `plt.legend()` ใช้กำหนดรูปแบบและสีของ marker และลักษณะ

- สามารถกำหนด marker และลักษณะโดยการนำ Parameter `label` เพื่อบริการต่อสีของความต้องการของข้อมูล record ของ user
- และนำ `color` เพื่อระบุสีของข้อมูลของ user
- `plt.scatter(df2['PetalLength'][0:50], df2['SepalLength'][0:50], c=df2['Name'][0:50], label='setosa')`
- `plt.scatter(df2['PetalLength'][50:100], df2['SepalLength'][50:100], c=df2['Name'][50:100], label='versicolor')`
- `plt.scatter(df2['PetalLength'][100:], df2['SepalLength'][100:], c=df2['Name'][100:], label='virginica')`
- `plt.legend()`

7

ผลลัพธ์ตัวอย่าง กำหนดชื่อและแสดงรูปของ marker และลักษณะ

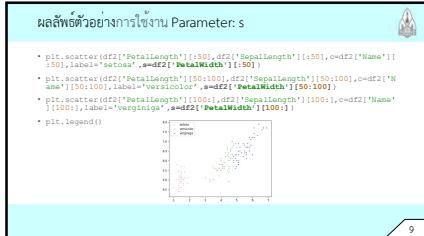
8

Parameter: s ใช้ขนาดของ marker และสีของ feature

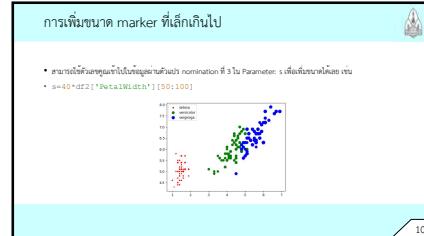
สามารถกำหนดขนาด marker โดยนำ parameter `s` สำหรับการกำหนดขนาดของ marker ที่ต้องการให้เป็น nomination หรือ `i` nomination หรือ `j` nomination ที่ 3 ทางกราฟจะเรียกว่า size ของ marker จะขยายตัว

- การกำหนด Parameter: `i` คือการกำหนดสีของ PetalWidth เป็น nomination ที่ 3
- `plt.scatter(df2['PetalLength'][0:50], df2['SepalLength'][0:50], c=df2['Name'][0:50], label='setosa', s=df2['PetalWidth'][0:50])`

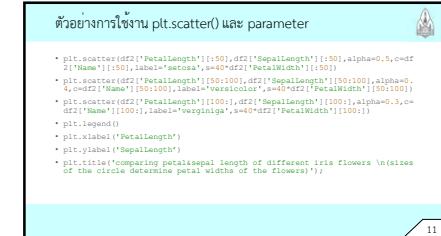
9



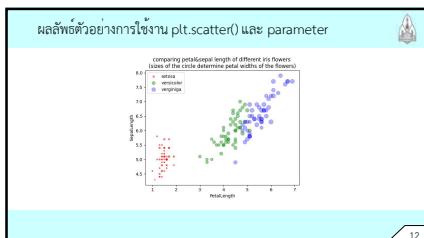
10



11



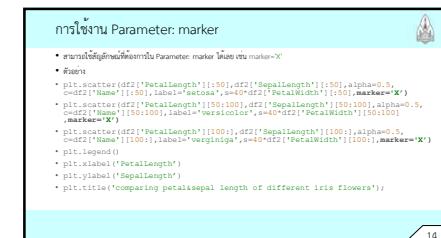
12



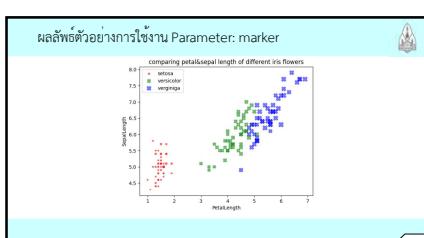
13



14



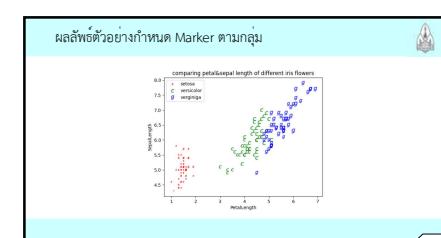
15



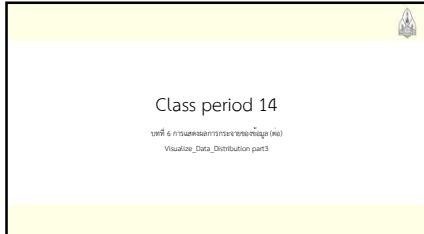
16



17



18



1

### Quiz

- ให้เขียนโค้ดภาษา Python ที่ใช้สำหรับสร้าง scatter plot ของข้อมูล 0\_2563 : 2567 ข้อ
- [https://dataverse.harvard.edu/dataset/916997/0\\_2563-2567/resource/71d552d0-0fa4-4e05-bf2c-420288a44d00](https://dataverse.harvard.edu/dataset/916997/0_2563-2567/resource/71d552d0-0fa4-4e05-bf2c-420288a44d00)
- และต้องใส่ code ดังนี้
- จะแสดงผลลัพธ์ทางภาพดังรูปนี้ ซึ่งจะเป็นแบบที่เราได้ไปฝึกกันไว้ใน

2

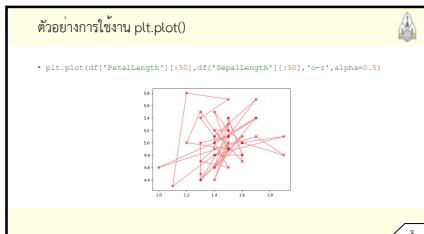
### plt.plot()

- เมื่อเขียนโค้ด scatter ให้ให้เป็นเก็บ และการใช้งานของ plt.scatter สามารถกำหนดให้เป็นสีของสีของ scatter และลักษณะ
- plt.plot() input คือที่ ๓ คือ ตัวที่กำหนดลักษณะของ marker เช่น
- plt.plot(df['PetalLength'][::50], df['SepalLength'][::50], 'o-r')

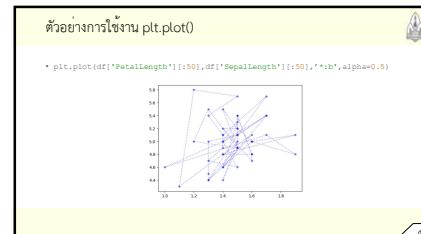
input คือที่ ๓ 'o-r' หมายความว่าตัวกำหนดลักษณะของ scatter เป็นสีแดงและลักษณะเป็นรูปวงกลม

- คือ กำหนดลักษณะ marker เป็นรูปวงกลม
- คือ กำหนดลักษณะของเส้นที่เชื่อมจุดเป็นเส้นสีฟ้า
- / คือ ตัวของลักษณะเส้น

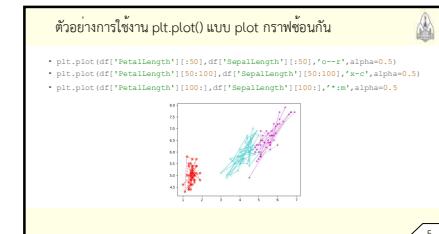
3



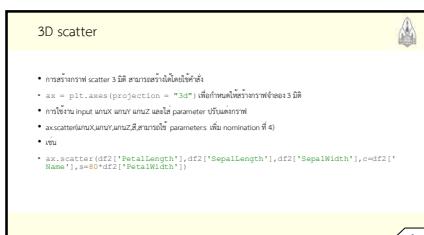
4



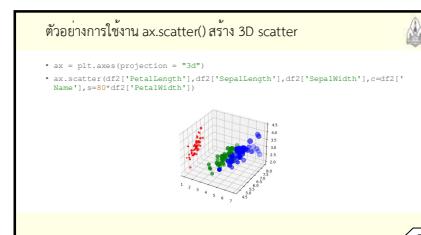
5



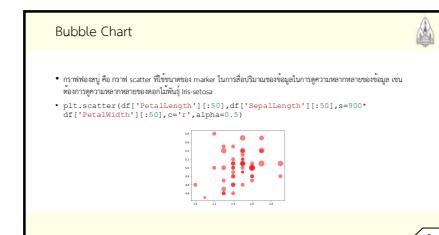
6



7



8



9

ตัวอย่างการวิเคราะห์ข้อมูลตามศาสตร์ Bubble Chart

- ไฟล์ข้อมูลที่ต้องการใช้คือ file:///C:/Users/2562/Downloads/titanic.csv
- <https://storage.googleapis.com/tensorflow/tf-keras-datasets/titanic.csv>
- เขียนภาษา Python ดังนี้

```

import pandas as pd
data = pd.read_excel('content/passenger-19.xlsx')
data

```

10

ตาราง

11

ขั้นตอนการ clean และเลือกข้อมูล

- ใช้ command drop ที่มีชื่อ missing
- data\_drop = data.dropna()
- data\_drop

ผลลัพธ์ที่ได้คือ Chart ที่กรอกข้อมูลโดยอัตโนมัติมาตามเดิม เช่นช่วงวัย "0-10 ปี", "10-20 ปี", "20-30 ปี" และ "30-40 ปี"

```

data.dropna()
data_drop = data.dropna()
print(data_drop)
# ผลลัพธ์ที่ได้คือ Chart ที่กรอกข้อมูลโดยอัตโนมัติมาตามเดิม เช่นช่วงวัย "0-10 ปี", "10-20 ปี", "20-30 ปี" และ "30-40 ปี"
# เรียก command นี้เพื่อให้ข้อมูล "age", "sex", และ "fare" ไม่หายไป: "age[0]", "sex[0]", และ "fare[0]"
0.001 * data_drop.iloc[:,3:7]
0 5236.391
1 2590.464
2 2690.174
Name: fare, dtype: float64

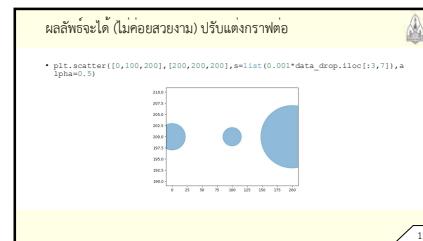
```

12

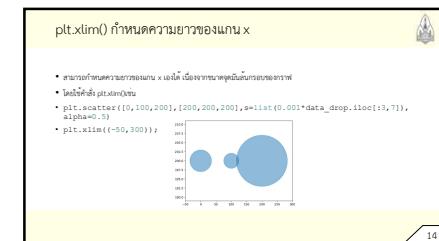
ขั้นตอนสร้าง Bubble Chart

- import pandas as pd
- import matplotlib
- from matplotlib import pyplot as plt
- plt.scatter([0,100,200],[200,200,200],s=[0.001\*data\_drop.iloc[:,3:7]),alpha=0.3)
- plt.scatter([100,200,300],[100,200,300],s=[0.001\*data\_drop.iloc[:,3:7]),alpha=0.3)

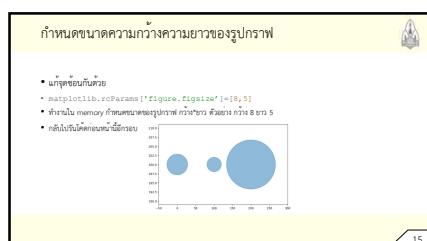
ผลลัพธ์จะได้ (ไม่ครอบคลุม) ปรับแต่งกราฟได้



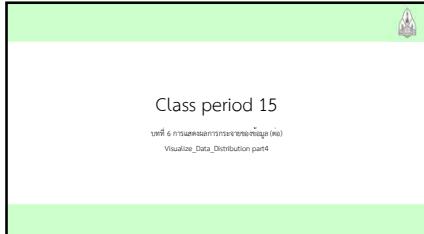
13



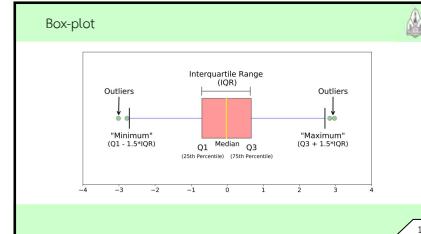
14



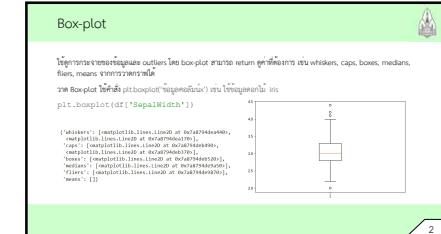
15



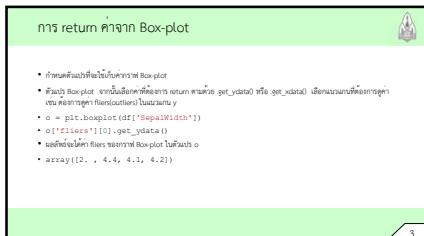
1



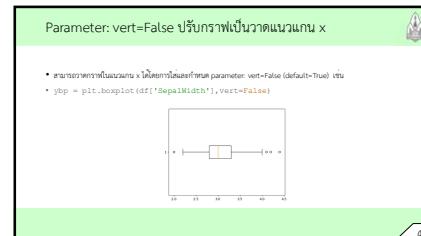
2



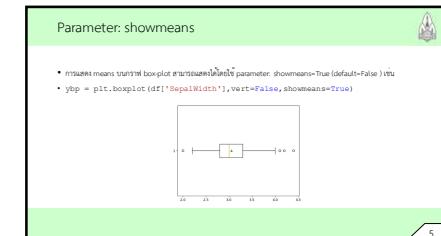
3



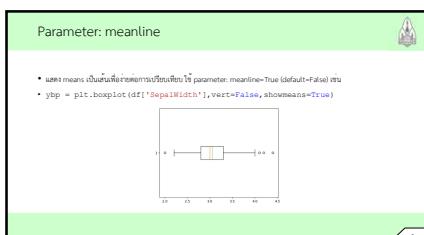
4



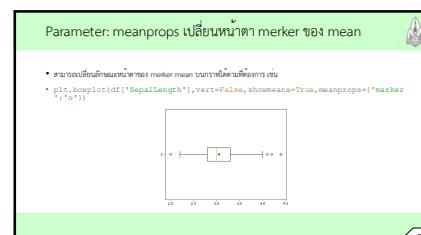
5



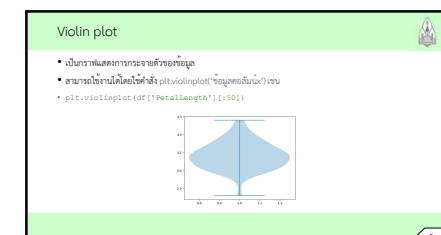
6



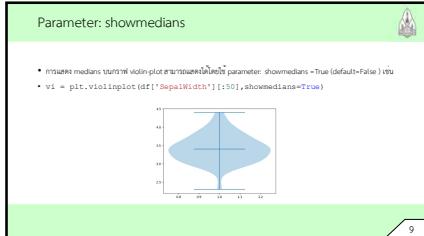
7



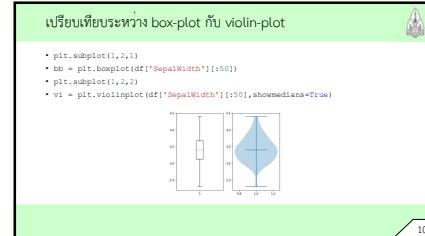
8



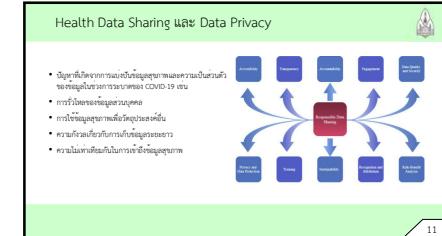
9



10



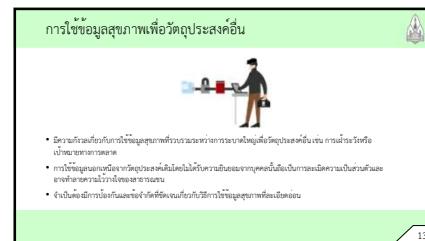
11



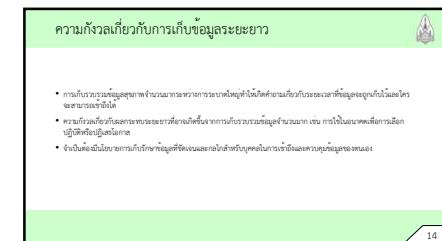
12



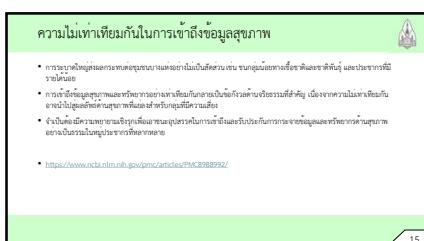
13



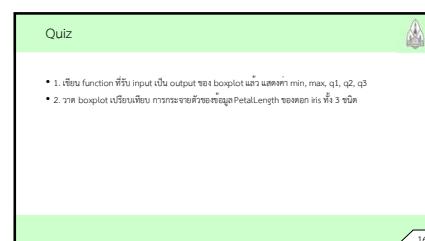
14



15



16



17

**Class period 16**

บทที่ ๔ การเผยแพร่องค์ความรู้ทางคอมพิวเตอร์  
timestamp - datetime

1

**Quiz**

- <https://drive.google.com/drive/folders/1AcMjGQ4ChjQH50751SCIVPyQZT3?usp=sharing>
- ไฟล์ตัวอย่าง example\_timestamp.csv
- เมื่อต้องการแปลง timestamp ของ alpha และ beta ให้เป็นวันที่ 2 มิถุนายน 2020 และวันที่ 2 มิถุนายน 2020

2

**Timestamp - datetime**

timestamp ใช้เวลาบันทึกวันที่และเวลาที่ ให้เวลาบันทึก timestamp  
ต้องแปลง timestamp ให้เป็น datetime ที่มีตั้งแต่วันเดือนปี พ.ศ. หรือ timestamp ต้องแปลง timestamp ให้เป็น datetime ที่มีตั้งแต่วันเดือนปี พ.ศ. หรือ timestamp ให้เป็น datetime ที่มีตั้งแต่วันเดือนปี พ.ศ.  
timestamp ต้องแปลง timestamp ให้เป็น datetime ที่มีตั้งแต่วันเดือนปี พ.ศ. หรือ timestamp ให้เป็น datetime ที่มีตั้งแต่วันเดือนปี พ.ศ.

3

**ยกตัวอย่างการซัพเพนดิคชัน timestamp ธรรมดา**

ต้องนำ timestamp ที่ได้จากการอ่านไฟล์ example\_timestamp.csv 0 ให้เปลี่ยนเป็น datetime ของ alpha และ beta ให้เป็นวันที่และเวลาที่ เช่น

```
be = df[0].to_datetime('1/1/1900 00:00:00')  
be  
be = df[0].to_datetime('1/1/1900 00:00:00')  
Unamed: 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 2020-06-04 2.64 16
```

4

**ยกตัวอย่างการซัพเพนดิคชัน timestamp ธรรมดา**

เมื่อต้องบันทึกวันเดือน และเวลาลงในแบบ string ธรรมดา

```
be = df[df['Unamed: 0'] < '02-04-2020']  
be  
Unamed: 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 01-06-2020 1.94 28  
4 02-06-2020 5.40 20  
5 03-06-2020 5.68 21  
6 03-06-2020 2.64 16
```

จะเห็นว่า เริ่มไปถูก เก็บมาเป็นแบบที่ชูชานและ มอง มอง เป็นชุดๆ  
หนึ่งชุดๆ 0 ถึง 10 หรือ 000 ถึง 999  
บันทึกต่อตัวของชุดๆ As หรือ ก็ ตั้งแต่ตัว 01-06-2020  
ทันทีเวลา 02-06-2020

5

**การใช้ timestamp (.to\_datetime)**

- การใช้งาน
  - `import pandas as pd`
  - `pd.to_datetime('1/1/1900 00:00:00', format='%d-%m-%Y')`
  - ให้ input format จะให้ตั้งแต่วันเดือนปี พ.ศ. ออก
  - %d ตัว วัน
  - %m ตัว เดือน
  - %Y ตัว ปี
  - รูปแบบตัวอักษรในตัวอักษร 01-06-2020 format="%d-%m-%Y"
  - รูปแบบตัวอักษรในตัวอักษร 2020-06-01 format="%Y-%m-%d"
  - รูปแบบตัวอักษรในตัวอักษร 01/06/2020 format="%d/%m/%Y"

6

**ตัวอย่าง timestamp (.to\_datetime) แปลงรูปแบบข้อมูล**

```
* import pandas as pd  
df = pd.read_csv('/content/example_timestamp_03.csv')  
df  
Unamed: 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 2020/06/04 2.64 16
```

7

**ตัวอย่างการใช้ timestamp() แปลงรูปแบบข้อมูล**

```
* df['Unamed: 0'] = df.to_datetime(df['Unamed: 0'], format='%d/%m/%Y')  
* df  
Unamed: 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 2020-06-04 2.64 16
```

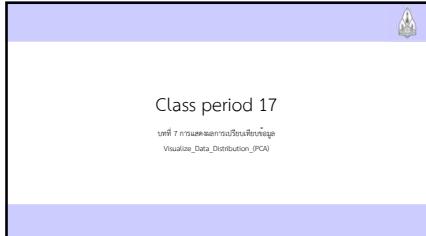
8

**การซัพเพนดิคชัน timestamp ด้วยคำสั่ง datetime()**

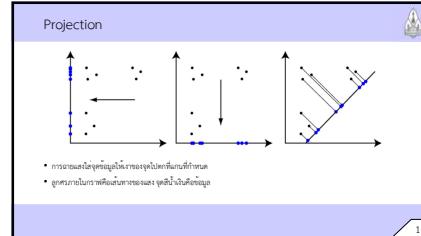
- การซัพเพนดิคชัน timestamp ให้ตั้งแต่ datetime สามารถกำหนดวันเดือนปีที่ต้องการได้
  - `datetime(day=วันที่ต้องการ, month=เดือนที่ต้องการ, year=ปีที่ต้องการ)`
  - \* วัน
  - `from datetime import datetime`
  - `df[df['Unamed: 0'].dt.datetime(day=2,month=6,year=2020)]`

```
Unamed: 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
```

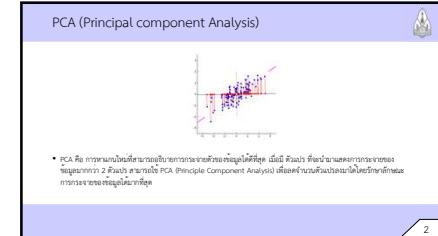
9



1



2



3

```
sklearn -> scikit-learn
```

- <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
- เป็น package ที่ร่วมกับ function ใหม่ๆ Data Science - Machine Learning - Data Mining ทำให้เราสามารถเขียนโค้ดได้เร็วขึ้น
- ทำการ import PCA ด้วย sklearn
- from sklearn.decomposition import PCA

4

การใช้ PCA มี 3 ขั้นตอน

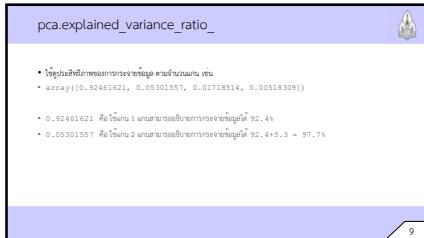
- Import
- from sklearn.decomposition import PCA
- Define
- pca = PCA()
- Fit - Transform คือ เครื่องมือที่ใช้ในการลด 차원ของข้อมูลโดยใช้การคำนวณของเวกเตอร์ที่เรียกว่า "พื้นที่ตัวถอด"
- new\_axis = pca.fit\_transform("ข้อมูลที่จะนำมาใช้ในการ PCA")

5

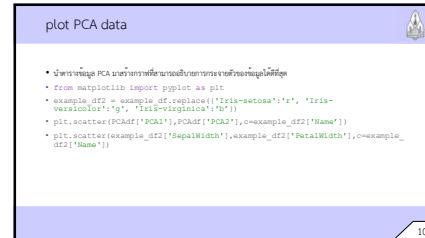
เตรียมข้อมูลดอกไม้ iris

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

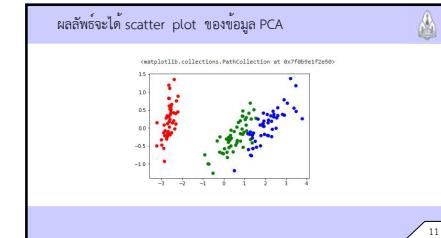
SepalLength	SepalWidth	PetalLength	PetalWidth
5.0	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.4	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.9	3.4	1.4	0.3
5.1	3.7	1.5	0.3
5.0	3.0	1.3	0.3
5.0	3.4	1.5	0.3
5.2	3.5	1.5	0.3
5.2	3.4	1.4	0.3
5.5	3.8	1.7	0.3
5.2	3.4	1.6	0.4
5.1	3.7	1.5	0.3
5.4	3.9	1.7	0.4
5.1	3.3	1.3	0.2
5.0	3.0	1.3	0.2
5.0	3.4	1.5	0.2
5.2	3.5	1.5	0.2
5.4	3.6	1.4	0.2
5.5	4.0	1.5	0.3
5.5	3.5	1.4	0.2
5.6	3.6	1.4	0.2
5.5	3.7	1.5	0.3
5.5	3.8	1.6	0.3
5.4	3.9	1.4	0.2
5.7	4.4	1.5	0.4
5.4	3.9	1.4	0.3
5.5	4.0	1.5	0.3
5.5	4.2	1.5	0.4
5.6	4.3	1.5	0.3
5.5	4.1	1.4	0.3
5.7	4.4	1.5	0.4
5.5	4.2	1.5	0.4
5.5	4.3	1.5	0.3
5.6	4.3	1.5	0.3
5.5	4.4	1.5	0.4
5.5	4.5	1.5	0.4
5.5	4.6	1.5	0.4
5.6	4.6	1.5	0.4
5.5	4.7	1.5	0.4
5.5	4.7	1.5	0.4
5.6	4.7	1.5	0.4
5.5	4.8	1.5	0.4
5.6	4.8	1.5	0.4
5.5	4.9	1.5	0.4
5.6	4.9	1.5	0.4
5.5	5.0	1.5	0.4
5.6	5.0	1.5	0.4
5.5	5.1	1.5	0.4
5.6	5.1	1.5	0.4
5.5	5.2	1.5	0.4
5.6	5.2	1.5	0.4
5.5	5.3	1.5	0.4
5.6	5.3	1.5	0.4
5.5	5.4	1.5	0.4
5.6	5.4	1.5	0.4
5.5	5.5	1.5	0.4
5.6	5.5	1.5	0.4
5.5	5.6	1.5	0.4
5.6	5.6	1.5	0.4
5.5	5.7	1.5	0.4
5.6	5.7	1.5	0.4
5.5	5.8	1.5	0.4
5.6	5.8	1.5	0.4
5.5	5.9	1.5	0.4
5.6	5.9	1.5	0.4
5.5	6.0	1.5	0.4
5.6	6.0	1.5	0.4
5.5	6.1	1.5	0.4
5.6	6.1	1.5	0.4
5.5	6.2	1.5	0.4
5.6	6.2	1.5	0.4
5.5	6.3	1.5	0.4
5.6	6.3	1.5	0.4
5.5	6.4	1.5	0.4
5.6	6.4	1.5	0.4
5.5	6.5	1.5	0.4
5.6	6.5	1.5	0.4
5.5	6.6	1.5	0.4
5.6	6.6	1.5	0.4
5.5	6.7	1.5	0.4
5.6	6.7	1.5	0.4
5.5	6.8	1.5	0.4
5.6	6.8	1.5	0.4
5.5	6.9	1.5	0.4
5.6	6.9	1.5	0.4
5.5	7.0	1.5	0.4
5.6	7.0	1.5	0.4
5.5	7.1	1.5	0.4
5.6	7.1	1.5	0.4
5.5	7.2	1.5	0.4
5.6	7.2	1.5	0.4
5.5	7.3	1.5	0.4
5.6	7.3	1.5	0.4
5.5	7.4	1.5	0.4
5.6	7.4	1.5	0.4
5.5	7.5	1.5	0.4
5.6	7.5	1.5	0.4
5.5	7.6	1.5	0.4
5.6	7.6	1.5	0.4
5.5	7.7	1.5	0.4
5.6	7.7	1.5	0.4
5.5	7.8	1.5	0.4
5.6	7.8	1.5	0.4
5.5	7.9	1.5	0.4
5.6	7.9	1.5	0.4
5.5	8.0	1.5	0.4
5.6	8.0	1.5	0.4
5.5	8.1	1.5	0.4
5.6	8.1	1.5	0.4
5.5	8.2	1.5	0.4
5.6	8.2	1.5	0.4
5.5	8.3	1.5	0.4
5.6	8.3	1.5	0.4
5.5	8.4	1.5	0.4
5.6	8.4	1.5	0.4
5.5	8.5	1.5	0.4
5.6	8.5	1.5	0.4
5.5	8.6	1.5	0.4
5.6	8.6	1.5	0.4
5.5	8.7	1.5	0.4
5.6	8.7	1.5	0.4
5.5	8.8	1.5	0.4
5.6	8.8	1.5	0.4
5.5	8.9	1.5	0.4
5.6	8.9	1.5	0.4
5.5	9.0	1.5	0.4
5.6	9.0	1.5	0.4
5.5	9.1	1.5	0.4
5.6	9.1	1.5	0.4
5.5	9.2	1.5	0.4
5.6	9.2	1.5	0.4
5.5	9.3	1.5	0.4
5.6	9.3	1.5	0.4
5.5	9.4	1.5	0.4
5.6	9.4	1.5	0.4
5.5	9.5	1.5	0.4
5.6	9.5	1.5	0.4
5.5	9.6	1.5	0.4
5.6	9.6	1.5	0.4
5.5	9.7	1.5	0.4
5.6	9.7	1.5	0.4
5.5	9.8	1.5	0.4
5.6	9.8	1.5	0.4
5.5	9.9	1.5	0.4
5.6	9.9	1.5	0.4
5.5	10.0	1.5	0.4
5.6	10.0	1.5	0.4
5.5	10.1	1.5	0.4
5.6	10.1	1.5	0.4
5.5	10.2	1.5	0.4
5.6	10.2	1.5	0.4
5.5	10.3	1.5	0.4
5.6	10.3	1.5	0.4
5.5	10.4	1.5	0.4
5.6	10.4	1.5	0.4
5.5	10.5	1.5	0.4
5.6	10.5	1.5	0.4
5.5	10.6	1.5	0.4
5.6	10.6	1.5	0.4
5.5	10.7	1.5	0.4
5.6	10.7	1.5	0.4
5.5	10.8	1.5	0.4
5.6	10.8	1.5	0.4
5.5	10.9	1.5	0.4
5.6	10.9	1.5	0.4
5.5	11.0	1.5	0.4
5.6	11.0	1.5	0.4
5.5	11.1	1.5	0.4
5.6	11.1	1.5	0.4
5.5	11.2	1.5	0.4
5.6	11.2	1.5	0.4
5.5	11.3	1.5	0.4
5.6	11.3	1.5	0.4
5.5	11.4	1.5	0.4
5.6	11.4	1.5	0.4
5.5	11.5	1.5	0.4
5.6	11.5	1.5	0.4
5.5	11.6	1.5	0.4
5.6	11.6	1.5	0.4
5.5	11.7	1.5	0.4
5.6	11.7	1.5	0.4
5.5	11.8	1.5	0.4
5.6	11.8	1.5	0.4
5.5	11.9	1.5	0.4
5.6	11.9	1.5	0.4
5.5	12.0	1.5	0.4
5.6	12.0	1.5	0.4
5.5	12.1	1.5	0.4
5.6	12.1	1.5	0.4
5.5	12.2	1.5	0.4
5.6	12.2	1.5	0.4
5.5	12.3	1.5	0.4
5.6	12.3	1.5	0.4
5.5	12.4	1.5	0.4
5.6	12.4	1.5	0.4
5.5	12.5	1.5	0.4
5.6	12.5	1.5	0.4
5.5	12.6	1.5	0.4
5.6	12.6	1.5	0.4
5.5	12.7	1.5	0.4
5.6	12.7	1.5	0.4
5.5	12.8	1.5	0.4
5.6	12.8	1.5	0.4
5.5	12.9	1.5	0.4
5.6	12.9	1.5	0.4
5.5	13.0	1.5	0.4
5.6	13.0	1.5	0.4
5.5	13.1	1.5	0.4
5.6	13.1	1.5	0.4
5.5	13.2	1.5	0.4
5.6	13.2	1.5	0.4
5.5	13.3	1.5	0.4
5.6	13.3	1.5	0.4
5.5	13.4	1.5	0.4
5.6	13.4	1.5	0.4
5.5	13.5	1.5	0.4
5.6	13.5	1.5	0.4
5.5	13.6	1.5	0.4
5.6	13.6	1.5	0.4
5.5	13.7	1.5	0.4
5.6	13.7	1.5	0.4
5.5	13.8	1.5	0.4
5.6	13.8	1.5	0.4
5.5	13.9	1.5	0.4
5.6	13.9	1.5	0.4
5.5	14.0	1.5	0.4
5.6	14.0	1.5	0.4
5.5	14.1	1.5	0.4
5.6	14.1	1.5	0.4
5.5	14.2	1.5	0.4
5.6	14.2	1.5	0.4
5.5	14.3	1.5	0.4
5.6	14.3	1.5	0.4
5.5	14.4	1.5	0.4
5.6	14.4	1.5	0.4
5.5	14.5	1.5	0.4
5.6	14.5	1.5	0.4
5.5	14.6	1.5	0.4
5.6	14.6	1.5	0.4
5.5	14.7	1.5	0.4
5.6	14.7	1.5	0.4
5.5	14.8	1.5	0.4
5.6	14.8	1.5	0.4
5.5	14.9	1.5	0.4
5.6	14.9	1.5	0.4
5.5	15.0	1.5	0.4
5.6	15.0	1.5	0.4
5.5	15.1	1.5	0.4
5.6	15.1	1.5	0.4
5.5	15.2	1.5	0.4
5.6	15.2	1.5	0.4
5.5	15.3	1.5	0.4
5.6	15.3	1.5	0.4
5.5	15.4	1.5	0.4
5.6	15.4	1.5	0.4
5.5	15.5	1.5	0.4
5.6	15.5	1.5	0.4
5.5	15.6	1.5	0.4
5.6	15.6	1.5	0.4
5.5	15.7	1.5	0.4
5.6	15.7	1.5	0.4
5.5	15.8	1.5	0.4
5.6	15.8	1.5	0.4
5.5	15.9	1.5	0.4
5.6	15.9	1.5	0.4
5.5	16.0	1.5	0.4
5.6	16.0	1.5	0.4
5.5	16.1	1.5	0.4
5.6	16.1	1.5	0.4
5.5	16.2	1.5	0.4
5.6	16.2	1.5	0.4
5.5	16.3	1.5	0.4
5.6	16.3	1.5	0.4
5.5	16.4	1.5	0.4
5.6	16.4	1.5	0.4
5.5	16.5	1.5	0.4
5.6	16.5	1.5	0.4
5.5	16.6	1.5	0.4
5.6	16.6	1.5	0.4
5.5	16.7	1.5	



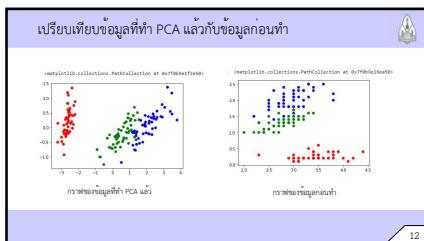
10



11



12



13



1

ให้ผลลัพธ์

- ใช้คำสั่ง `gcloud search query` ค้นหา
- <https://cloud.google.com/search/docs/guides/quickstart> ต้องติดตั้ง `gcloud` แล้ว `gcloud auth login` ก่อน
- บนคอมพิวเตอร์ที่ติดตั้ง `gcloud` แล้ว เรียกใช้คำสั่ง `gcloud drive list` ให้ได้ผลลัพธ์
- บนคอมพิวเตอร์ที่ติดตั้ง `gcloud` แล้ว เรียกใช้คำสั่ง `gcloud drive list` ให้ได้ผลลัพธ์

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

2

```
def search_and_log(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')
```

(一)

## Quiz

- เข้าสู่การซื้อผลิตภัณฑ์ของบริษัท คุณสามารถ ดูรายละเอียดเพิ่มเติมได้ที่ web Wongnai.com เพื่อตัดสินใจซื้ออาหาร ในเมืองนี้

4

### Bar chart (กราฟแท่ง)

Bar chart คือการนำเสนอข้อมูลเป็นรูปแบบของแท่ง

- สามารถอ่านค่าของแท่ง
- แสดงความสูงของค่าที่ต้องการ
- กำหนดความสูง (height)
- กำหนดความกว้าง (width)
- ตั้งชื่อ x (x\_label)
- ตั้งชื่อ y (y\_label)
- ตั้งชื่อ x (ptc\_label)
- ตั้งชื่อ y (ptc\_label)
- ตั้งชื่อ y (ptc\_label)

Chart 5.3.1  
Number of police officers in Chonburi, 2011 to 2019

Year	Number of police officers
2011	~85
2012	~80
2013	~75
2014	~70
2015	~65
2016	~68
2017	~72
2018	~75
2019	~78

5

6

```

from part1 import pyplot as plt
length = plt.bar('เรียนรู้ฟังก์ชัน', 'list ที่มีค่าตัวแปรในตัวฟังก์ชัน') จำนวน
plt.show()
search_request_count จำนวนที่มีค่าตัวแปรในตัวฟังก์ชัน บนเว็บ Mongai.com จำนวนที่มีค่าตัวแปรในตัวฟังก์ชัน

plt.bar([1,2,3,4,5,6], [data[0]*'search_timeavg', data[1]*'search_timeavg', data[2]*'search_timeavg', data[3]*'search_timeavg', data[4]*'search_timeavg', data[5]*'search_timeavg'], width=0.5)
plt.xlabel('จำนวน')
plt.ylabel('ค่าตัวแปร')

```

7

ผลลัพธ์ การสร้าง bar chart

Category	Value
1	1.0
2	0.9
3	0.9
4	1.0
5	1.5
6	2.0
7	1.8

8

Parameter: tick\_label (ชื่อแท่ง)

- ใช้ command กากบาทนี้ได้โดยการใส่ Parameter: tick\_label = ["ชื่อแท่งที่ 1", "ชื่อแท่งที่ 2", ..., "ชื่อแท่งที่ n"]
- เช่น
- plt.bar([1,2,3,4,5,6,7], data["search\_linearamp"], dt.daysweek == 1, shape=10),  
▪ data["search\_linearamp"].dt.daysweek == 1, shape=[0],  
▪ data["search\_linearamp"].dt.daysweek == 2, shape=[0],  
▪ data["search\_linearamp"].dt.daysweek == 3, shape=[0],  
▪ data["search\_linearamp"].dt.daysweek == 4, shape=[0],  
▪ data["search\_linearamp"].dt.daysweek == 5, shape=[0],  
▪ data["search\_linearamp"].dt.daysweek == 6, shape=[0],  
▪ data["search\_linearamp"].dt.daysweek == 7, shape=[0]  
▪ tick\_label=["Mon","Tue","Wed","Thu","Fri","Sat","Sun"]

6

A bar chart titled "ผลลัพธ์ Parameter: tick\_label" showing daily sales data. The x-axis represents the days of the week: Mon, Tue, Wed, Thu, Fri, Sat, Sun. The y-axis represents sales values ranging from 0.0 to 3.0. The bars show sales increasing from Monday (~1.0) to Saturday (~2.8), with Sunday slightly lower (~2.5).

Day	Sales
Mon	1.0
Tue	0.9
Wed	1.0
Thu	1.0
Fri	1.4
Sat	2.8
Sun	2.5

10

11

A bar chart titled "ผลลัพธ์ของการกำหนดชื่อเว็บไซต์และข้อเสนอ" (Results of website naming and promotion) showing the number of visitors (Number of visitors) per day (Day). The Y-axis ranges from 0.0 to 1.0 with increments of 0.5. The X-axis lists the days of the week: Mon, Tue, Wed, Thu, Fri, Sat, Sun. The bars show visitor counts of approximately 0.65 for Monday, 0.65 for Tuesday, 0.65 for Wednesday, 0.65 for Thursday, 0.85 for Friday, 1.0 for Saturday, and 0.85 for Sunday.

Day	Number of visitors
Mon	0.65
Tue	0.65
Wed	0.65
Thu	0.65
Fri	0.85
Sat	1.0
Sun	0.85

12

The chart displays the number of votes each candidate received in each year. Jones and Wilson show a general upward trend in votes over time. Hansen's vote count increased significantly between 2010 and 2012. Peter's votes decreased from 2010 to 2012 before slightly increasing in 2013. Meadham consistently received the highest number of votes, peaking in 2013.

Candidate	2008	2010	2012	2013
Jones	10	25	30	35
Wilson	15	20	25	30
Hansen	10	20	45	40
Peter	10	15	10	20
Meadham	40	50	55	60

13

## ขั้นตอนการสร้าง Grouped bar chart

14

15

กำหนดตัวแปร labels ใช้เก็บชื่อแท่งกราฟ

16

```
เตรียมตัวบ่เป้ที่เข้าร่วมกราฟ
```



```
* import numpy as np
* x = np.arange(len(labels))  
* ravel คือหันหอยหอยของ array ที่ไม่ต้องการให้เป็น二维的 labels
* width = 0.35 เวิร์ดของกราฟ
* fig, ax = plt.subplots () ใช้ subplot ในการกราฟภาพขึ้น
```

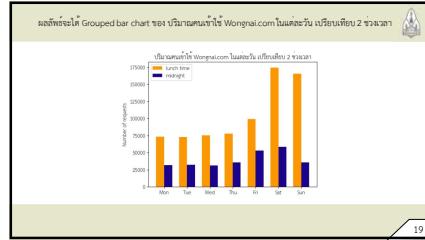
17

- `rects1 = ax.bar(x - width/2, b4lunch, width, label='lunch time',color = '#fc9700')`#mb4lunch
- `rects2 = ax.bar(x + width/2, b4midnight, width, label='midnight',color = '#19908A')`#mb4midnight
- พอกการบาร์ที่มีสีต่างๆเป็น b4lunch จะมีค่า x อยู่ก่อนหน้า แต่พอกการบาร์ที่มีสีต่างๆเป็น b4midnight จะมี rect1, rect2

18



19



20

Class period 19  
บทที่ 7 กรณีและการเรียนรู้แบบกลุ่ม (cont.)  
Stacked bar chart and histogram

1

Stacked Bar Chart

- เป็นกราฟแท่งที่สามารถใช้ในการเรียบเรียงข้อมูลโดยรวมของแต่ละค่าน้ำ และถ้าไม่ได้มากราฟแท่งจะแสดงผลลัพธ์ของข้อมูลของแต่ละกลุ่ม

Category	before lunch	after lunch	Total
Before	~70	~30	~100
During	~20	~10	~30
After	~40	~20	~60

2

ขั้นตอนการสร้าง Stacked Bar Chart

- ต้องทราบจำนวนข้อมูลที่ต้องการสร้าง Grouped bar chart
  - ต้องนำข้อมูลที่ต้องการมาเรียงลำดับตามลำดับที่ต้องการ เช่นเช้า,b4lunch,b4midnight ที่เรียกว่า "เรียงลำดับ"
  - กำหนดชื่อ แสดงว่า นี่คือข้อมูลของ (เช่น ชื่อผู้ใช้ หรือชื่อไฟล์ของข้อมูล)
  - นำข้อมูลที่เรียงลำดับมากราฟ

3

เตรียมตัวประที่ดูสร้างกราฟ

```
* 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 = '#1f933d')
- Parameter bottom ใช้ในการกำหนดฐานของแท่งกราฟให้กับแท่งเดิม
  - กำหนด bottom=b4lunch ในการกราฟที่ต้องมีฐาน b4lunch
  - หากมีการกำหนดฐานต่อๆ ไป b4midnight ก็ต้องมีฐาน b4lunch

5

กำหนดลักษณะกราฟต่างๆ

```
ax.set_ylabel('number of requests')
ax.set_title('จำนวนขอเข้า Wongnai.com ในช่วงเช้า ไนคิดว่า 2 ชั่วโมง')
ax.legend()
plt.show()
```

6

ผลลัพธ์ที่ได้ Stacked Bar Chart ของ บริษัทคุณเข้าไป Wongnai.com ในช่วงเช้า ไนคิดว่า 2 ชั่วโมง

Hour	before lunch	after lunch	Total
6 AM	~15000	~10000	~25000
7 AM	~15000	~10000	~25000
8 AM	~15000	~10000	~25000
9 AM	~15000	~10000	~25000
10 AM	~15000	~10000	~25000
11 AM	~15000	~10000	~25000
12 PM	~15000	~10000	~25000
1 PM	~15000	~10000	~25000
2 PM	~15000	~10000	~25000
3 PM	~15000	~10000	~25000
4 PM	~15000	~10000	~25000
5 PM	~15000	~10000	~25000
6 PM	~15000	~10000	~25000

7

การสร้างกราฟแท่งแนวอนุ

- ให้แน่ใจว่า axbar ในกราฟแท่งแนวอนุจะเป็น
 

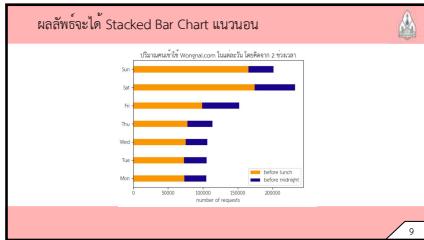
```
ax.barh(labels, b4lunch, width, label='before lunch',color = '#fc9700')
      ax.barh(labels, b4midnight, width, left=b4lunch, label='before midnight',color = '#1f933d')
```

8

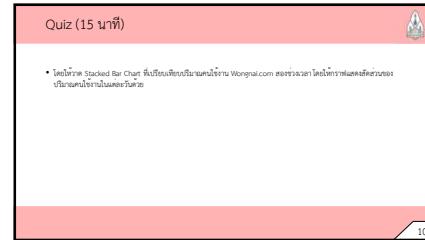
ตัวอย่างการสร้างกราฟแท่งแนวอนุ

```
width = 0.35
fig, ax = plt.subplots()
ax.barh(labels, b4lunch, width, label='before lunch',color = '#fc9700')
ax.barh(labels, b4midnight, width, left=b4lunch, label='before midnight',color = '#1f933d')
ax.set_xlabel('number of requests')
ax.set_title('จำนวนขอเข้า Wongnai.com ในช่วงเช้า ไนคิดว่า 2 ชั่วโมง')
ax.legend()
plt.show()
```

9



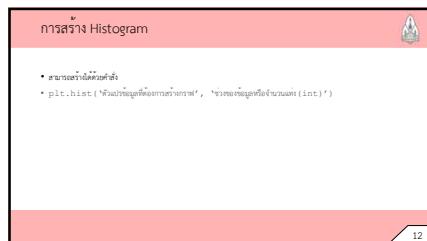
10



11



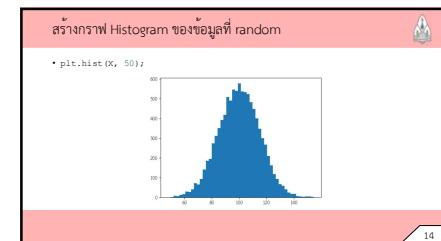
12



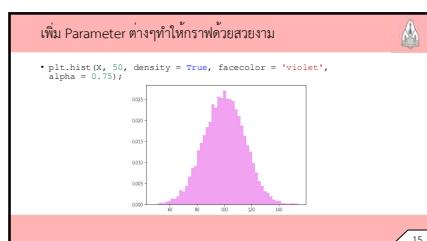
13



14



15



16

Class period 20  
บทที่ 7 การแสดงผลและการเรียกข้อมูล (cont)  
Histogram and render chart

1

ตัวอย่างชี้มูล Wongnai.com

```
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize']=[15,15]
output = plt.hist(list(data['number_of_result']),10,facecolor = 'red',
alpha = 0.75)
```

สร้าง Histogram โดยใช้ข้อมูลค่านั้น 'number\_of\_result' ใน Wongnai.com โดยกำหนดรายละเอียดให้จำนวนผลเป็น 10 ก็จะได้รูปแบบค่าความถี่ตามที่เราได้และเป็น 75%

2



3

แก้ไข แผน x ที่เรียงข้อมูลผิด

```
1. ตรวจสอบ data type ของ column ตัวนึง
data.dtypes
2. เรียกดูเฉพาะ column data type จะพบว่าในคอลัมน์นี้คือ float
type(data['number_of_result'][10])
type(data['number_of_result'][10]) == int
3. แปลงคอลัมน์ type ทุกบันทึกเป็น int
new_type = data['number_of_result'].astype('int32')
```

4

ลบ record

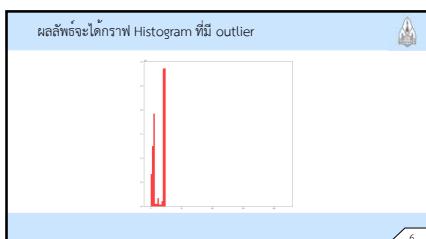
```
1. ตรวจสอบว่า record ใดที่มีค่าใน number_of_result
data[data['number_of_result']==number_of_result].shape[0]
2. ลบ record ที่มีค่าใน number_of_result
data = data.drop(1000016)
```

5

บังคับเปลี่ยน type ข้อมูลเป็น int

```
1. บังคับเปลี่ยน type ข้อมูลเป็น int หลังจากลบ record ที่ 1000016 แล้ว และเก็บข้อมูลที่เปลี่ยนมาไว้ในชื่อ new_type
new_type = data['number_of_result'].astype('int32')
2. สร้าง Histogram ที่สามารถเรียก new_type
output = plt.hist(new_type,100,facecolor = 'red',alpha = 0.75)
```

6



7

ลบ outlier

```
1. หา outlier โดยการนับค่าที่ต่ำกว่าและสูงกว่า 25 ของในส่วนนี้ให้
new_type_nooutlier = new_type[new_type < 25]
2. ตรวจสอบว่ามีค่าที่เป็น outlier หรือไม่
new_type.shape[0] - new_type_nooutlier.shape[0]
3. ลบค่าที่ไม่ใช่ outlier ทั้งหมด 14 ตัว
new_type.shape[0] - new_type_nooutlier.shape[0]
14
```

8



9

**Quiz**

- ເບີນເບີນຂອງການມື່ອພັດທຶນມາກັບ ແລ້ວກົມາກັບ

10

**ເລີຍຂໍ້າວ**

- output ຖື່ນຕົວໄສແລ້ວການປິດໃຫຍ່ ດາວໂຫຼວງໃຫຍ່ ກ່າວມະນີມານີ້ຊຸດ 5717238  
`(array([ 43023., 124726., 1921441., 3720., 5060., 16750.,  
 36883., 35914., 105490., 5717238.]),  
 array([ 0., 2., 4., 6., 8., 11., 13., 15., 17., 19., 21. ]),  
 list of 10 Patch objects)`
- output[0] ເມື່ອການວິວວ່າ ທີ່ທີ່ 0 000 ອາລຸນ ຕົກລາມເນື້ອຂອງການ
`array([ 43023., 124726., 1921441., 5170., 5060., 16750.,  
 36883., 35914., 105490., 5717238.])`

11

**ເລີຍຄວບ**

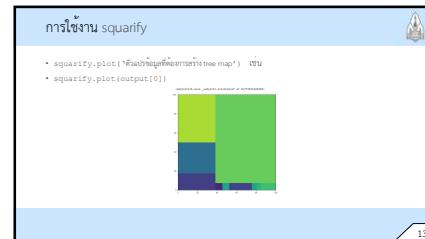
- output[0][-1] ຕົກລາມເນື້ອຂອງການ  
`5717238.0`
- sum(output[0][:-1]) ຕົກລາມເນື້ອການ  
`4287104.0`

12

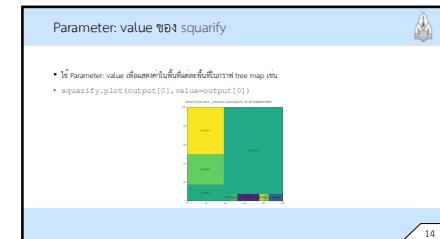
**Tree map**

- ການນິກົນນີ້ແມ່ນຫຼຸດທີ່ການນິກົນທີ່ໃນໄຊ່ປະລິມທີ່ນີ້ ສໍາຜົນການພົບທີ່ຈະກົດໃຫຍ່ການຮູ້
- ສໍາຜົນໃຈການກົດໄວ້ Tree map ທີ່ກຳນົດ packet squarify ໄດ້ຮັບກຳນົດກຳນົດທີ່ຈະກົດ
- `!pip install squarify`
- `import squarify`
- `import numpy as np`
- `import matplotlib.pyplot as plt`

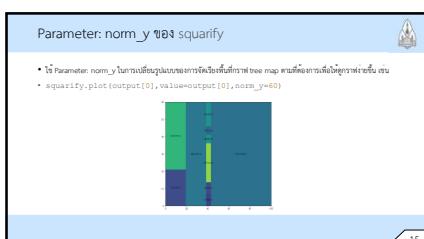
13



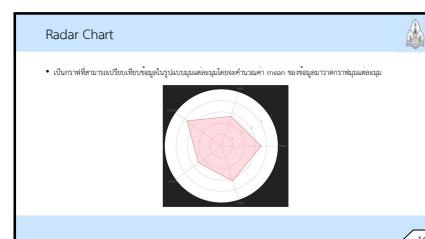
14



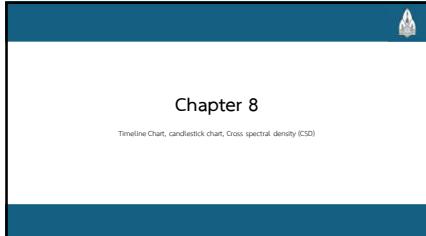
15



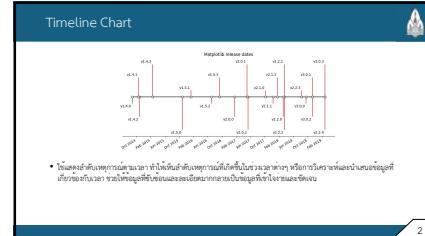
16



17



1



2

```
* from datetime import datetime
* import matplotlib.pyplot as plt
* import numpy as np
* import matplotlib.dates as mdates
```

3

```
* names = ['v2.2.4', 'v3.0.3', '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 = ['2018-02-24', '2018-02-24', '2018-11-10', '2018-11-10',
         '2018-09-07', '2018-08-07', '2018-05-18', '2018-05-18',
         '2018-03-06', '2018-01-18', '2017-12-10', '2017-10-07',
         '2017-05-10', '2017-05-03', '2017-01-17', '2016-09-03',
         '2016-07-03', '2016-01-07', '2015-10-29', '2015-02-18',
         '2014-10-26', '2014-01-18', '2014-08-28']
```

4

```
* dates = [datetime.strptime(d, "%Y-%m-%d") for d in dates]
* np.random.shuffle(dates) ชื่อฟังก์ชัน random.shuffle สำหรับ datetime ของ Python และทำการสุ่ม dates
* levels = np.tile([-5, 5, -3, 3, -1, 1], int(np.ceil(len(dates)/6)))[:len(dates)]
* สำหรับการสุ่มตัวแปรจะมีผลลัพธ์ที่ต้องการในที่สุด 6 ชั้น
* [-5, 5, -3, 3, -1, 1] คือตัวแปรที่ต้องการและจะถูกสุ่มตัวให้ได้ในที่สุด 6 ชั้น
* int(np.ceil(len(dates)/6)) คือจำนวนชั้นที่ต้องการ 6 รอบ แต่จำนวนที่เหลือใน dates
* [:len(dates)] คือตัดเฉพาะในที่สุด
```

5

```
* fig, ax = plt.subplots(figsize=(8, 4), layout="constrained")
* ax.set(title="Matplotlib release dates")
* กับ subplotจะทำภาพเป็นรูป 8 ชิ้น ดู 4 ชิ้น
* layout="constrained" เป็น parameter ที่จะช่วยจัดหน้าจอของ subplot ให้เข้ากับขอบของ
  หน้าจอและไม่เกินขอบเขตของ subplot และ ที่สำคัญ ช่วยลดเวลาในการทำงานของที่ให้กราฟของเรา
* กับชื่อเรียก Matplotlib release dates
```

6

สร้างเส้น timeline ของราย

```
* ax.vlines(dates, 0, levels, color="tab:red")
* เป็นการสร้างเส้น timeline และตั้งค่าบานปลายของเส้น timeline ให้กับร่องค่าว่างกับตัวเส้น timeline
  เส้นตัวเส้น และตัวเส้นที่เป็นเส้นสีแดง

* ax.plot(dates, np.zeros_like(dates), "-o",
          color="k", markerfacecolor="w")
* ใช้คำสั่ง timeline แหล่งที่มาของเส้น timeline และรูป

* np.zeros_like(dates) คือการกำหนดทุกๆวันที่ในdates จะมีค่าเป็น 0 ซึ่งจะแสดงถึงการที่เป็น
  เส้นตัวเส้นที่เราต้องการ
* "-o", color="k", markerfacecolor="w" คือกำหนดรูปทรงเป็นลูกศรบิดเบี้ยว และกำหนด
  สีของเส้นเป็นสีดำ
```

7

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

* for d, l, n in zip(dates, levels, names):
    np.sign(dates) คำการรักษาตัวเลขเดิมของ d ให้เป็น
    ตัวเลขเดิมของ d แต่จะเปลี่ยน sign ให้เป็น 1 หรือ -1
    ax.annotate(n, xy=(d, l),
                xytext=(3, np.sign(l)*3),
                textcoords="offset points",
                horizontalalignment="left",
                verticalalignment="bottom" if l > 0 else "top")
```

8

```
* xytext=(-3, np.sign(l)*3) ใช้กำหนดตัวเลขเดิมของ d เพื่อไม่ให้ไปหลังตัวอักษร
* -3 แทนตัวเลขที่ต้องไม่มากไป คือตัวเลขที่ต้องห่างจากตัวอักษร
* np.sign(l)*3 หมายความว่าต้องไม่มากไป คือตัวเลขที่ต้องห่างจากตัวอักษร levels ให้มาก
* textcoords="offset points" คือการรักษาตัวเลขเดิมของ d แต่จะเปลี่ยน sign ให้เป็น
  ตัวเลขเดิมของ d แต่จะเปลี่ยน sign ให้เป็น 1 หรือ -1
* horizontalalignment="right" หมายความว่าต้องห่างจากตัวอักษรของ d ไปทางขวาของตัวอักษรใน levels
* verticalalignment="bottom" if l > 0 else "top"
* ใช้คำสั่งตัวแปรที่มีชื่อตามตัวอักษรที่กำหนด (เช่น d) ขึ้นมาอย่าง 0 ต้องตั้งค่าให้ตรงกับตัวอักษรนั้น เพื่อ
  ให้รูปตัวเส้นที่เราต้องการมีรูปแบบที่เราต้องการ (เช่น ลูกศรบิดเบี้ยว) ที่สำคัญ
```

9

ก้าวเดียวที่จะทำให้คุณง่ายๆ

```

    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))
    # ให้ตัวเลขหักเศษที่ด้านบน ไม่แสดงตัวเดือน
    ax.xaxis.set_major_formatter(mdates.DateFormatter("%b %Y"))
    # ให้ตัวเลขหักเศษที่ด้านบน x 40 ตัวเพื่อแสดงว่า 4 เดือน
    plt.setp(ax.get_xticklabels(), rotation=30, ha="right")
    # ให้ตัวเลขหักเศษที่ด้านบน x 40 ตัวเพื่อแสดงว่า 30 วัน

```

10

ก้าวเดียวที่จะทำให้คุณง่ายๆ

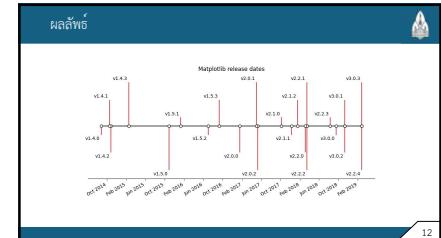
```

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

    ax.yaxis.set_visible(False) # ให้ตัวเลขหักเศษที่ด้านบน
    ax.spines["left", "top", "right"].set_visible(False)
    # ให้ตัวเลขหักเศษที่ด้านบน x 40 ตัวเพื่อแสดงว่า 4 เดือน
    ax.margins(y=0.1)
    # ให้ตัวเลขหักเศษที่ด้านบน x 40 ตัวเพื่อแสดงว่า 30 วัน
    # ให้ตัวเลขหักเศษที่ด้านบน x 40 ตัวเพื่อแสดงว่า 30 วัน

```

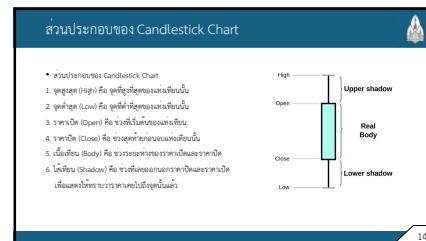
11



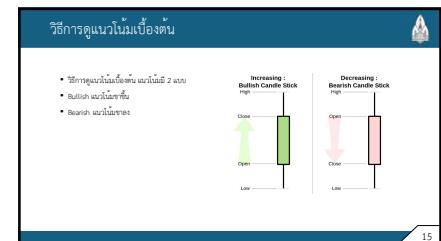
12



13



14



15

Import packet

```

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

```

16

Download data to memory

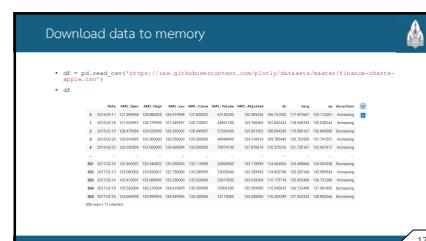
```

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

```

Date	Open	High	Low	Close	Adj Close	Volume	Moving Average
2010-10-01	101.000000	103.000000	99.000000	102.000000	101.000000	101.000000	101.000000
2010-10-04	104.000000	105.000000	102.000000	104.000000	103.000000	103.000000	103.000000
2010-10-05	105.000000	106.000000	103.000000	105.000000	104.000000	104.000000	104.000000
2010-10-06	106.000000	107.000000	104.000000	106.000000	105.000000	105.000000	105.000000
2010-10-07	107.000000	108.000000	105.000000	107.000000	106.000000	106.000000	106.000000
2010-10-08	108.000000	109.000000	106.000000	108.000000	107.000000	107.000000	107.000000
2010-10-09	109.000000	110.000000	107.000000	109.000000	108.000000	108.000000	108.000000
2010-10-12	110.000000	111.000000	108.000000	110.000000	109.000000	109.000000	109.000000
2010-10-13	111.000000	112.000000	109.000000	111.000000	110.000000	110.000000	110.000000
2010-10-14	112.000000	113.000000	110.000000	112.000000	111.000000	111.000000	111.000000
2010-10-15	113.000000	114.000000	111.000000	113.000000	112.000000	112.000000	112.000000
2010-10-16	114.000000	115.000000	112.000000	114.000000	113.000000	113.000000	113.000000
2010-10-19	115.000000	116.000000	113.000000	115.000000	114.000000	114.000000	114.000000
2010-10-20	116.000000	117.000000	114.000000	116.000000	115.000000	115.000000	115.000000
2010-10-21	117.000000	118.000000	115.000000	117.000000	116.000000	116.000000	116.000000
2010-10-22	118.000000	119.000000	116.000000	118.000000	117.000000	117.000000	117.000000
2010-10-23	119.000000	120.000000	117.000000	119.000000	118.000000	118.000000	118.000000
2010-10-26	120.000000	121.000000	118.000000	120.000000	119.000000	119.000000	119.000000
2010-10-27	121.000000	122.000000	119.000000	121.000000	120.000000	120.000000	120.000000
2010-10-28	122.000000	123.000000	120.000000	122.000000	121.000000	121.000000	121.000000
2010-10-29	123.000000	124.000000	121.000000	123.000000	122.000000	122.000000	122.000000
2010-10-30	124.000000	125.000000	122.000000	124.000000	123.000000	123.000000	123.000000
2010-10-31	125.000000	126.000000	123.000000	125.000000	124.000000	124.000000	124.000000
2010-11-01	126.000000	127.000000	124.000000	126.000000	125.000000	125.000000	125.000000
2010-11-02	127.000000	128.000000	125.000000	127.000000	126.000000	126.000000	126.000000
2010-11-03	128.000000	129.000000	126.000000	128.000000	127.000000	127.000000	127.000000
2010-11-04	129.000000	130.000000	127.000000	129.000000	128.000000	128.000000	128.000000
2010-11-05	130.000000	131.000000	128.000000	130.000000	129.000000	129.000000	129.000000
2010-11-06	131.000000	132.000000	129.000000	131.000000	130.000000	130.000000	130.000000
2010-11-09	132.000000	133.000000	130.000000	132.000000	131.000000	131.000000	131.000000
2010-11-10	133.000000	134.000000	131.000000	133.000000	132.000000	132.000000	132.000000
2010-11-11	134.000000	135.000000	132.000000	134.000000	133.000000	133.000000	133.000000
2010-11-12	135.000000	136.000000	133.000000	135.000000	134.000000	134.000000	134.000000
2010-11-13	136.000000	137.000000	134.000000	136.000000	135.000000	135.000000	135.000000
2010-11-16	137.000000	138.000000	135.000000	137.000000	136.000000	136.000000	136.000000
2010-11-17	138.000000	139.000000	136.000000	138.000000	137.000000	137.000000	137.000000
2010-11-18	139.000000	140.000000	137.000000	139.000000	138.000000	138.000000	138.000000
2010-11-19	140.000000	141.000000	138.000000	140.000000	139.000000	139.000000	139.000000
2010-11-22	141.000000	142.000000	139.000000	141.000000	140.000000	140.000000	140.000000
2010-11-23	142.000000	143.000000	140.000000	142.000000	141.000000	141.000000	141.000000
2010-11-24	143.000000	144.000000	141.000000	143.000000	142.000000	142.000000	142.000000
2010-11-25	144.000000	145.000000	142.000000	144.000000	143.000000	143.000000	143.000000
2010-11-26	145.000000	146.000000	143.000000	145.000000	144.000000	144.000000	144.000000
2010-11-29	146.000000	147.000000	144.000000	146.000000	145.000000	145.000000	145.000000
2010-11-30	147.000000	148.000000	145.000000	147.000000	146.000000	146.000000	146.000000
2010-12-01	148.000000	149.000000	146.000000	148.000000	147.000000	147.000000	147.000000
2010-12-02	149.000000	150.000000	147.000000	149.000000	148.000000	148.000000	148.000000
2010-12-03	150.000000	151.000000	148.000000	150.000000	149.000000	149.000000	149.000000
2010-12-06	151.000000	152.000000	149.000000	151.000000	150.000000	150.000000	150.000000
2010-12-07	152.000000	153.000000	150.000000	152.000000	151.000000	151.000000	151.000000
2010-12-08	153.000000	154.000000	151.000000	153.000000	152.000000	152.000000	152.000000
2010-12-09	154.000000	155.000000	152.000000	154.000000	153.000000	153.000000	153.000000
2010-12-10	155.000000	156.000000	153.000000	155.000000	154.000000	154.000000	154.000000
2010-12-13	156.000000	157.000000	154.000000	156.000000	155.000000	155.000000	155.000000
2010-12-14	157.000000	158.000000	155.000000	157.000000	156.000000	156.000000	156.000000
2010-12-15	158.000000	159.000000	156.000000	158.000000	157.000000	157.000000	157.000000
2010-12-16	159.000000	160.000000	157.000000	159.000000	158.000000	158.000000	158.000000
2010-12-17	160.000000	161.000000	158.000000	160.000000	159.000000	159.000000	159.000000
2010-12-20	161.000000	162.000000	159.000000	161.000000	160.000000	160.000000	160.000000
2010-12-21	162.000000	163.000000	160.000000	162.000000	161.000000	161.000000	161.000000
2010-12-22	163.000000	164.000000	161.000000	163.000000	162.000000	162.000000	162.000000
2010-12-23	164.000000	165.000000	162.000000	164.000000	163.000000	163.000000	163.000000
2010-12-24	165.000000	166.000000	163.000000	165.000000	164.000000	164.000000	164.000000
2010-12-27	166.000000	167.000000	164.000000	166.000000	165.000000	165.000000	165.000000
2010-12-28	167.000000	168.000000	165.000000	167.000000	166.000000	166.000000	166.000000
2010-12-29	168.000000	169.000000	166.000000	168.000000	167.000000	167.000000	167.000000
2010-12-30	169.000000	170.000000	167.000000	169.000000	168.000000	168.000000	168.000000
2010-12-31	170.000000	171.000000	168.000000	170.000000	169.000000	169.000000	169.000000

17



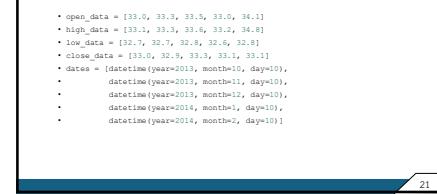
18



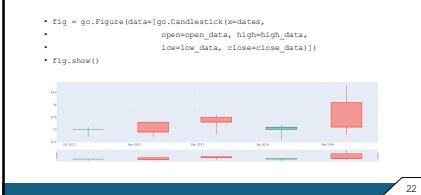
19



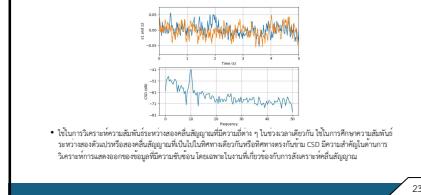
20



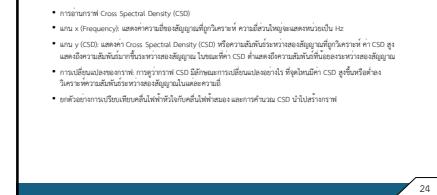
21



22



23



24

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

25

จำลองข้อมูลเวลาในรูปแบบความ

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

- ใช้ส่วน `range` ด้านบนของ `time` ที่มีรูปแบบนี้มาตั้งค่าเป็น 0 ถึง 1 ที่มีจำนวนตัวอย่าง 1000 ตัวเพื่อใช้กับ `np.linspace` จะผลิตตัวอย่าง 0 ถึง 1 ออกบันทึกโดยตรงที่ชื่อ `time` กับตัวอย่าง 1000 ชิ้น

26

จำลองข้อมูลคิ่นไฟฟ้าหัวใจ ECG

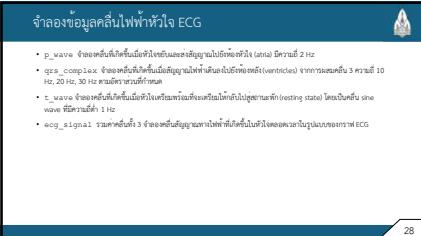
```

    • 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

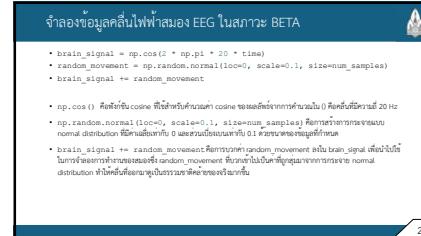
  • np.sin(1) #sin波の振幅を1としたときの周波数を求めるため
  • np.pi #円周率Pi(3.14159)を表示する際のラジアン数を表示する

```

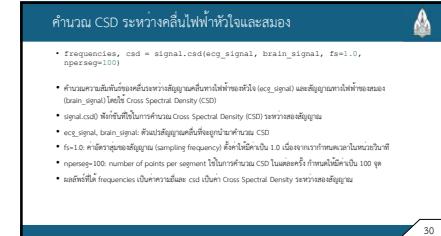
27



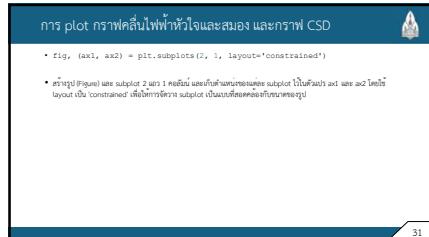
28



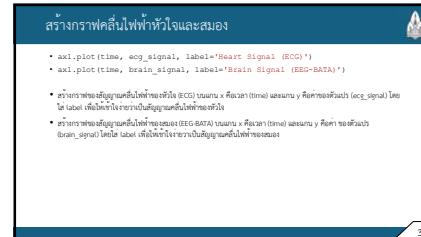
29



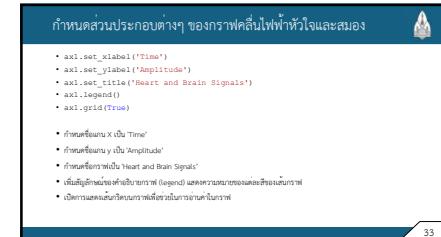
30



31



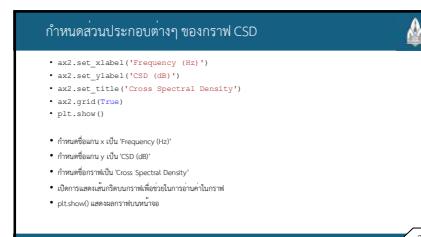
32



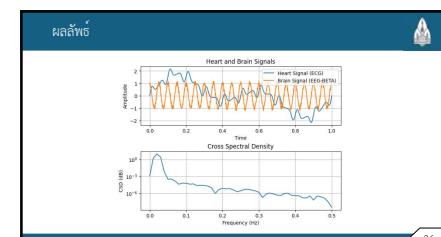
33



34



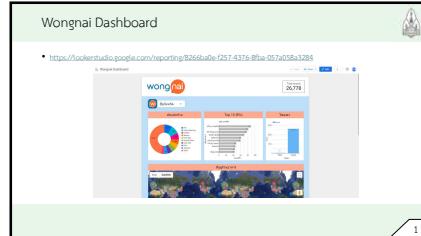
35



36



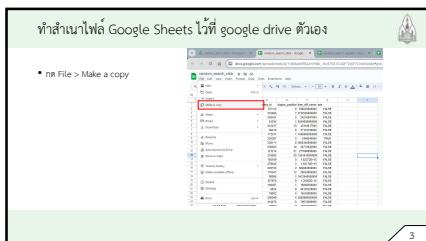
1



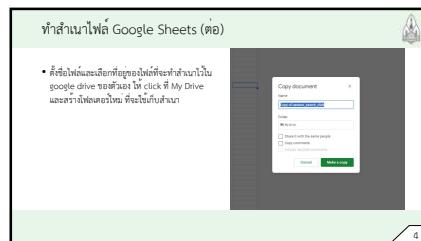
2



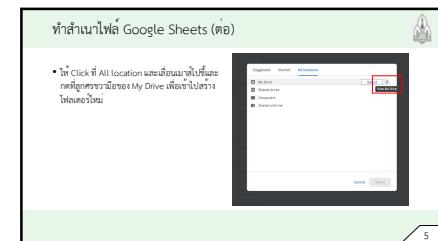
3



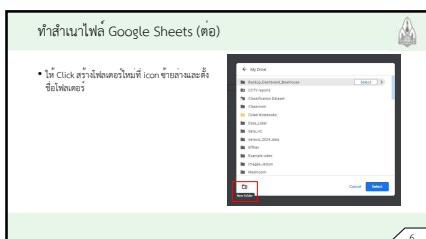
4



5



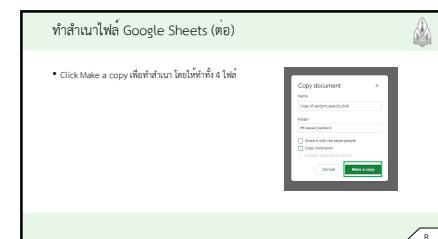
6



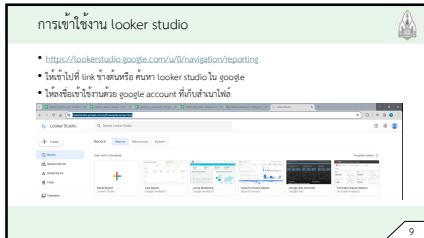
7



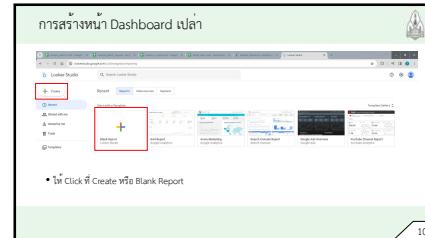
8



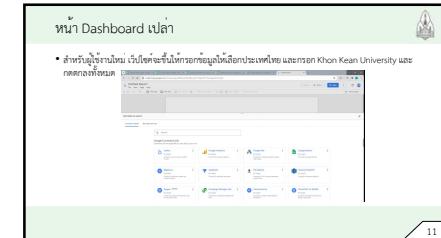
9



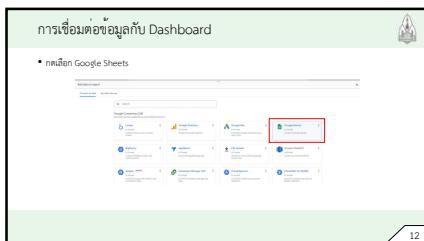
10



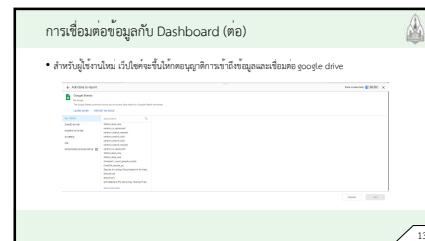
11



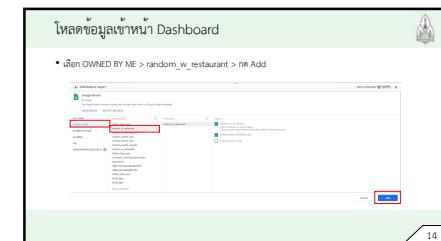
12



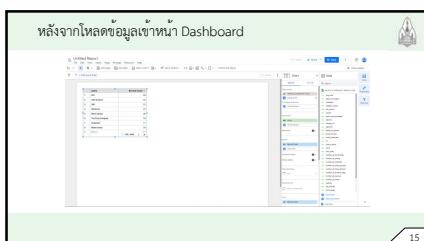
13



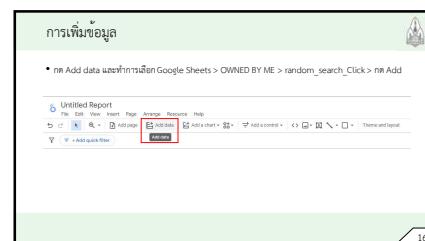
14



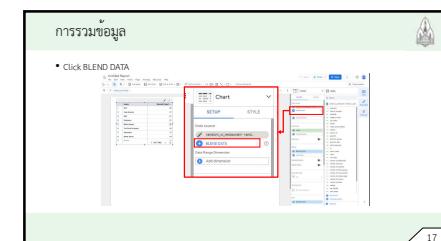
15



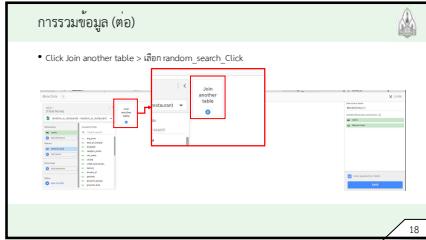
16



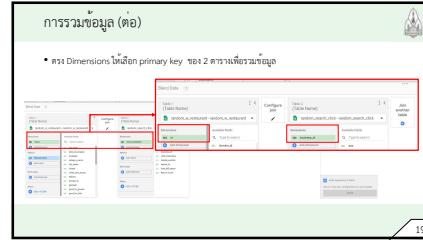
17



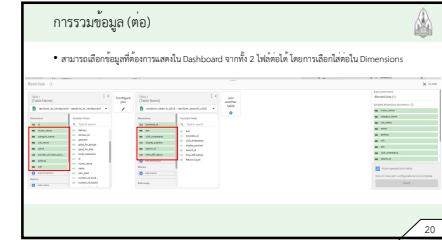
18



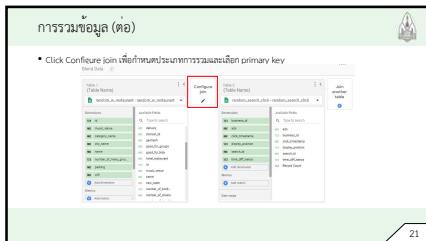
19



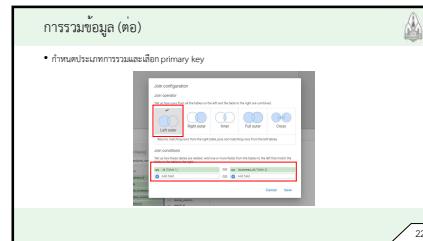
20



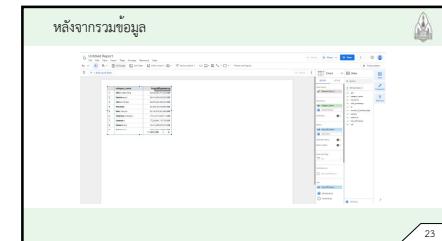
21



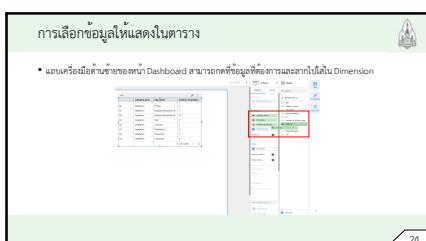
22



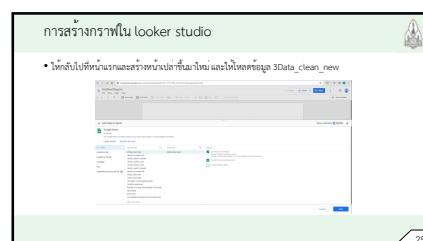
23



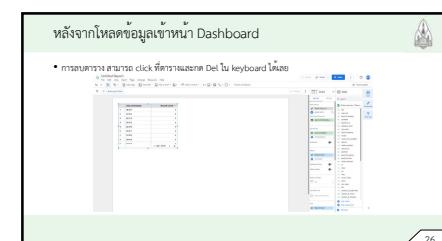
24



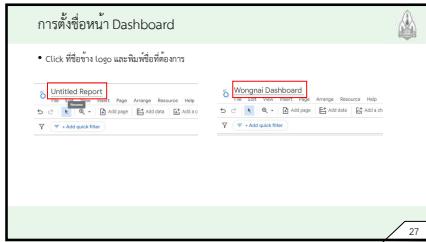
25



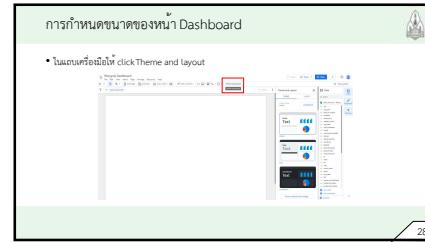
26



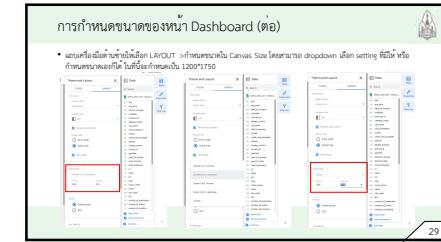
27



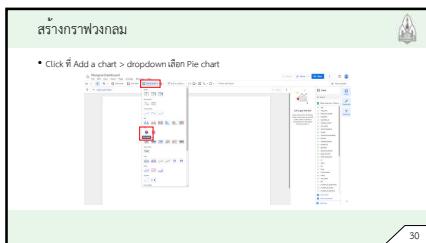
28



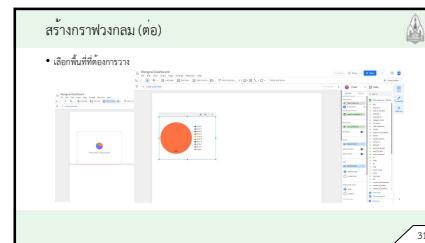
29



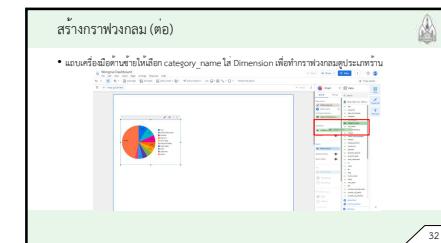
30



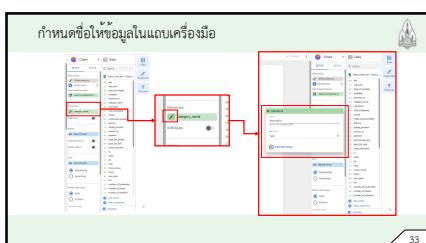
31



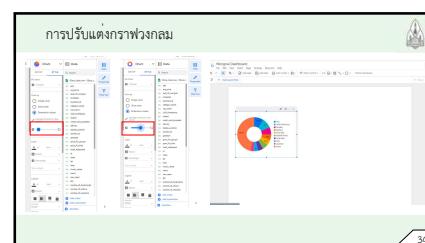
32



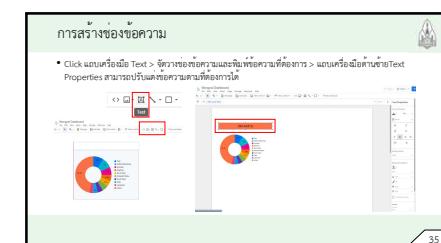
33



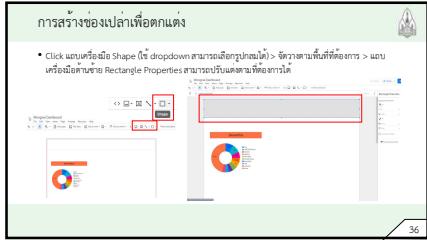
34



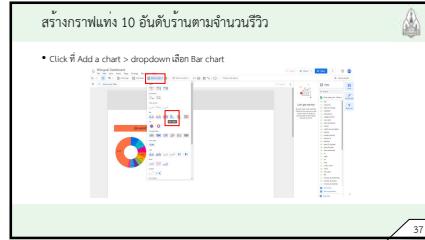
35



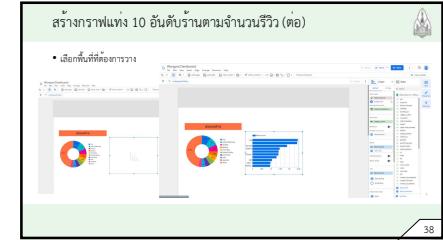
36



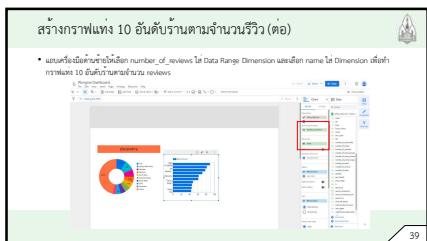
37



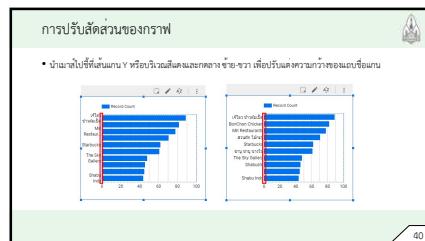
38



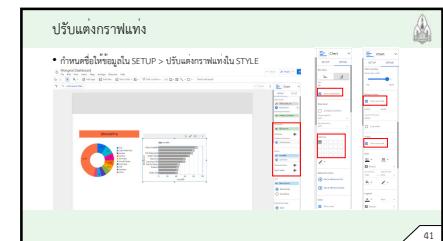
39



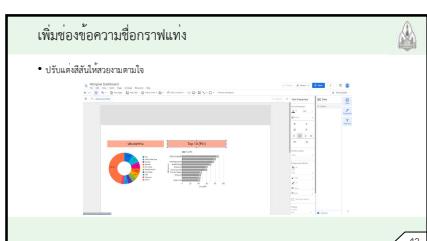
40



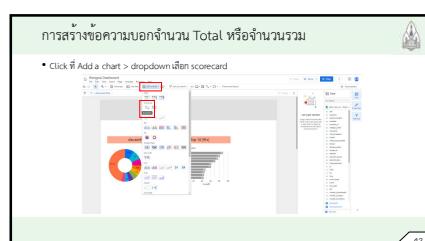
41



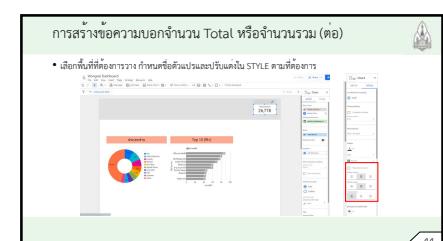
42



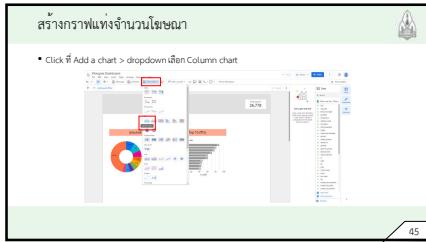
43



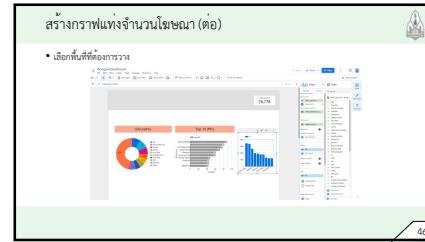
44



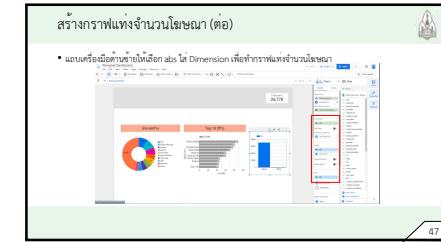
45



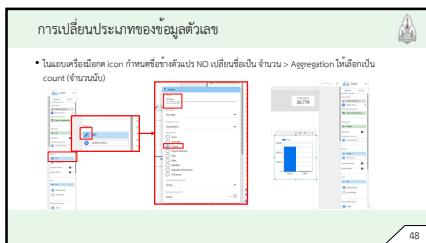
46



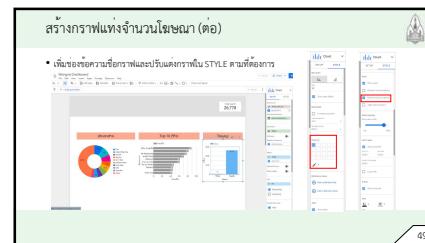
47



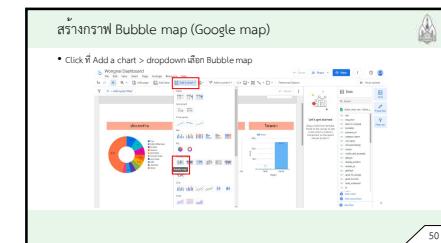
48



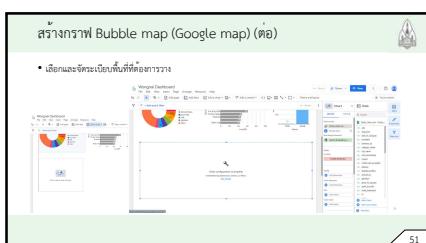
49



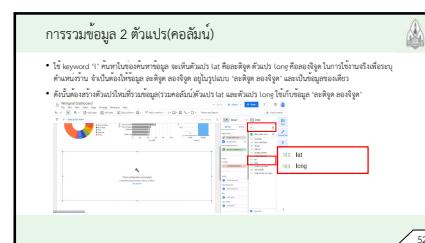
50



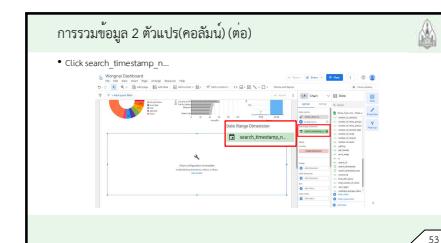
51



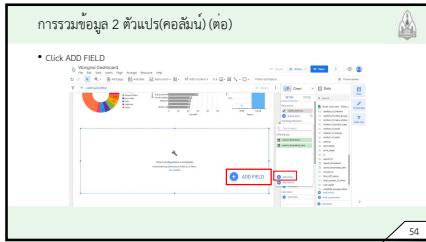
52



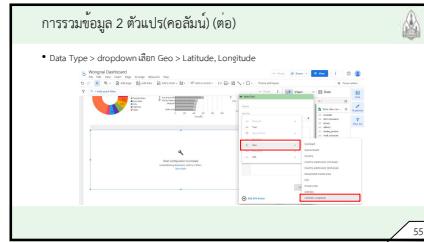
53



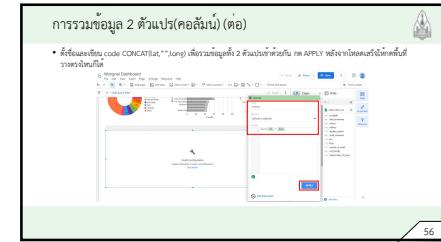
54



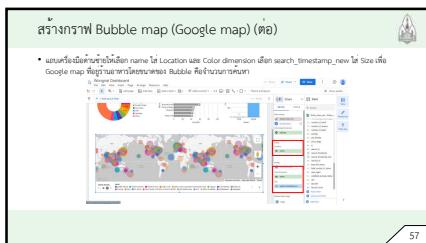
55



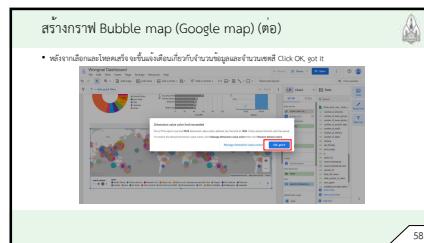
56



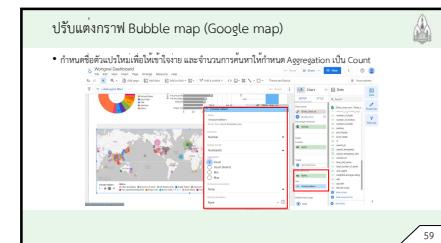
57



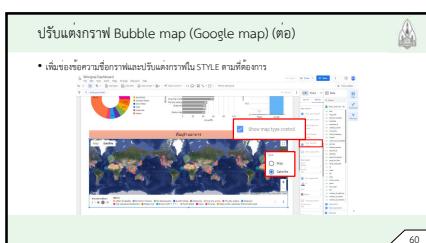
58



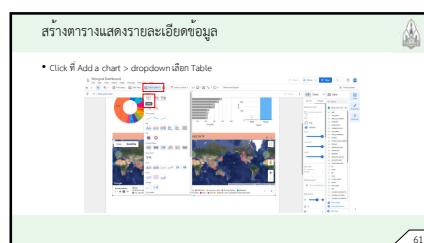
59



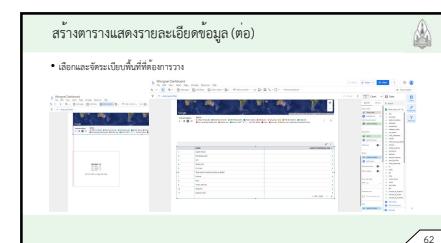
60



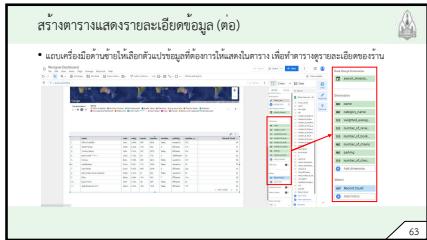
61



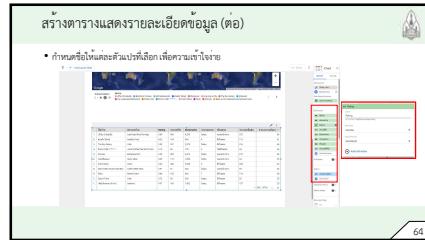
62



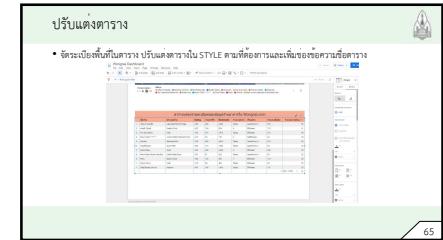
63



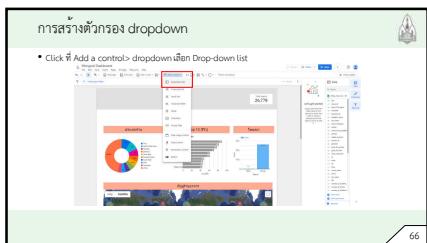
64



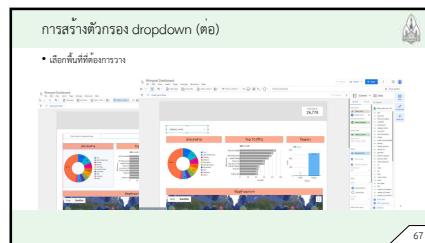
65



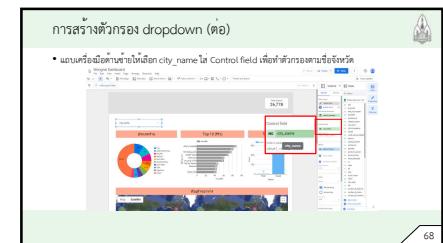
66



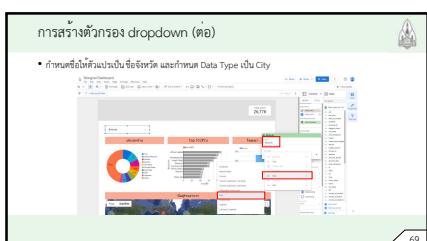
67



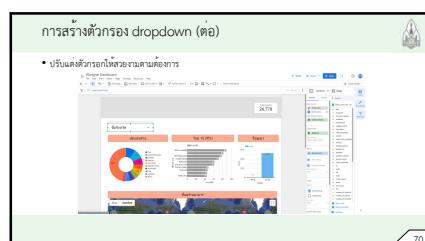
68



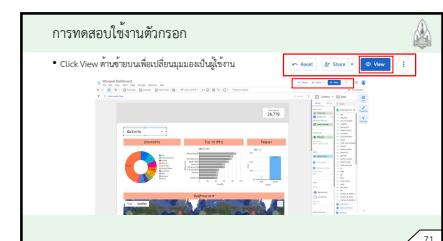
69



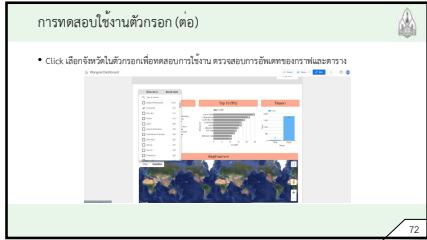
70



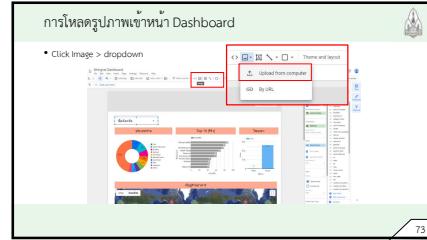
71



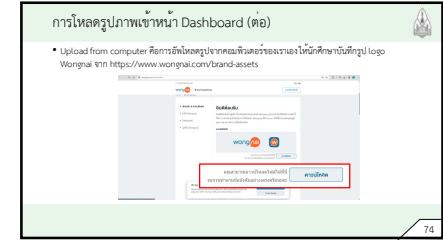
72



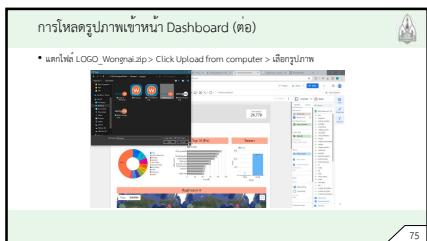
73



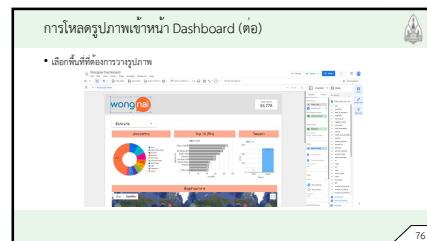
74



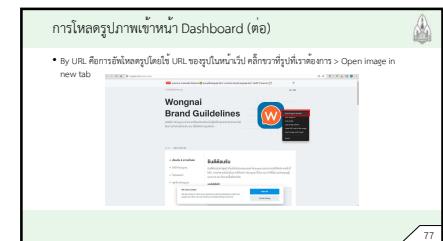
75



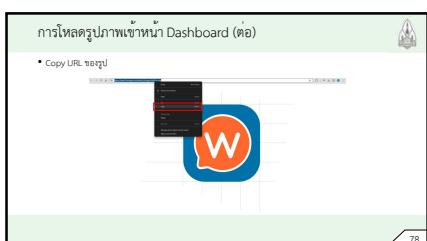
76



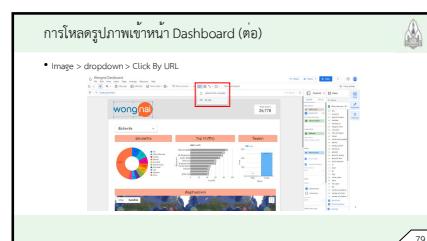
77



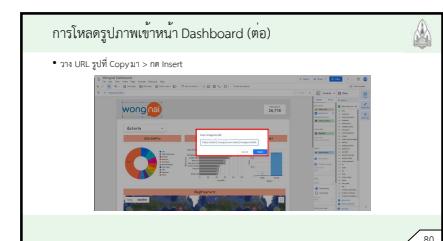
78



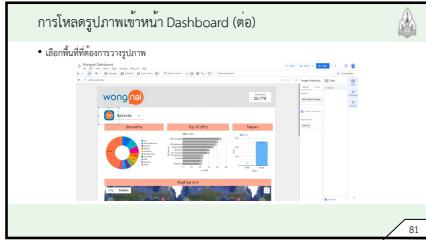
79



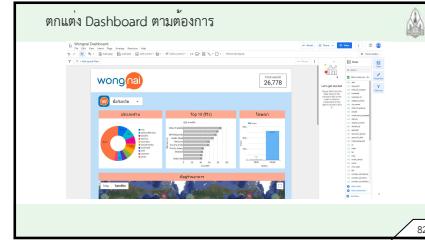
80



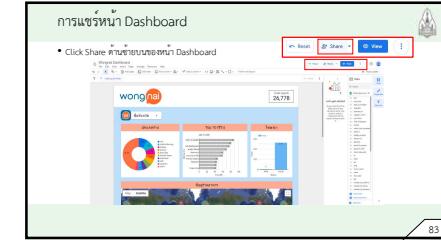
81



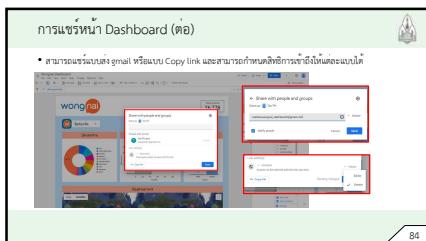
82



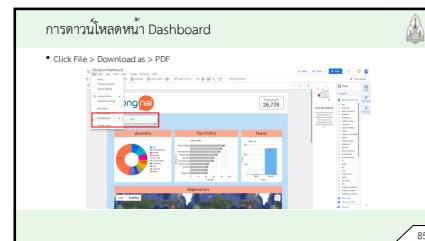
83



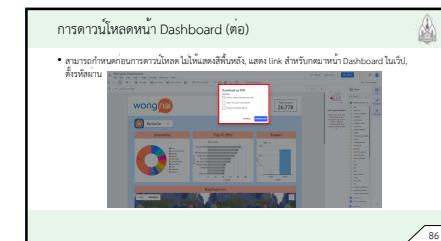
84



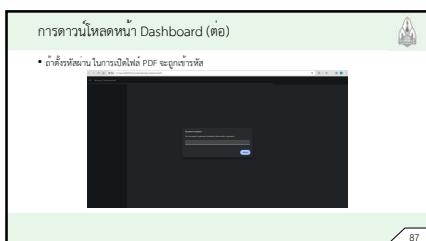
85



86



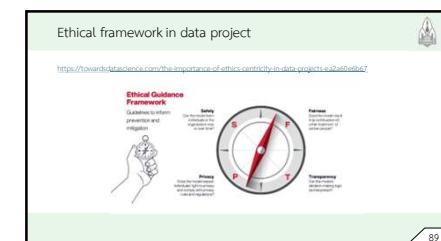
87



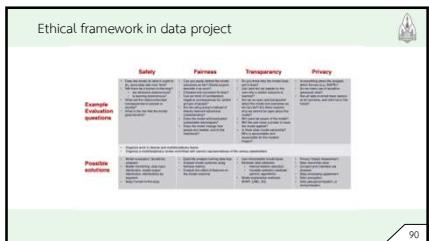
88



89



90



91