

บทที่ 1

(แนะนำรายวิชา)

บทที่ 1: แนะนำรายวิชา

1. ขี้แจงเกณฑ์การให้คะแนนและการตัดเกรด จำนวนค疤เรียนและชั่วโมงเรียน ข้อตกลงระหว่างการเรียนการสอน การสอบ
2. ความรู้เบื้องต้นเกี่ยวกับการใช้งาน GitHub เพื่อใช้บันทึกและส่งการบ้าน-ข้อสอบ
 - การสร้างบัญชี GitHub: ขั้นตอนและวิธีการสร้างบัญชีใน GitHub
 - การสร้าง repository: ขั้นตอนในการสร้างพื้นที่เก็บโครงการหรือโค้ดบน GitHub
 - การใช้สำหรับส่งงานและเก็บ Python Notebook ผ่าน Google Colab: การใช้ GitHub เป็นพื้นที่เก็บไฟล์ Notebook และวิธีการเชื่อมต่อกับ Google Colab เพื่อทำงานร่วมกัน
 - การเขียน README file: ขั้นตอนในการเขียน README file เพื่ออธิบายเนื้อหาของโครงการหรือโค้ดใน repository
3. แนะนำภาษา Python และการใช้ Python Notebook ผ่าน Google Colab
 - ภาษา Python: ความเป็นมา และคุณสมบัติพื้นฐานของภาษา Python
 - การใช้ Python Notebook ผ่าน Google Colab: ขั้นตอนการใช้งานและการเขียนโปรแกรมภาษา Python ในรูปแบบ Notebook ผ่านแพลตฟอร์ม Google Colab
 - การใช้งานโค้ดภาษา Python ใน Google Colab: วิธีการเขียนและปรับแต่งโค้ดภาษา Python ใน Notebook ของ Google Colab เพื่อทำงานต่างๆ
 - การบันทึก Colab Notebook ลงใน GitHub เพื่อส่งงาน

บทที่ 2

(ตัวแปรและโครงสร้างข้อมูลของภาษาไพธอน)

บทที่ 2: ตัวแปรและโครงสร้างข้อมูลของภาษาไพธอน

1. การเขียนโปรแกรมภาษา Python เป็องตัน	
● นิยามของตัวแปร: ตัวแปรใน Python คืออะไร และความสำคัญของตัวแปร.....	1
● หลักการตั้งชื่อตัวแปรเบ็องตัน: วิธีการตั้งชื่อตัวแปรให้เป็นระเบียบและอ่านง่าย.....	1
● ชนิดของตัวแปรและการเปลี่ยนชนิดของข้อมูล: การระบุชนิดของข้อมูลและการแปลงชนิดของข้อมูลในภาษา Python.....	1
2. Operation (การดำเนินการระหว่างตัวแปร).....	2
● ตัวอย่างการใช้งานเบ็องตัน: การใช้งานเครื่องหมายต่างๆ เพื่อดำเนินการกับตัวแปร	
● การดำเนินการพื้นฐาน: การบวก ลบ คูณ หาร และ มอตูโล	
3. คำสั่ง print (การ format string).....	2
● การใช้งาน print statement: วิธีการใช้คำสั่ง print เพื่อแสดงข้อความและค่าของตัวแปร	
● การ format string ในภาษา Python: การใช้งานรูปแบบสตริงเพื่อแสดงข้อมูลที่ถูกกำหนดรูปแบบที่มีความซับซ้อน	
4. Data structure (โครงสร้างข้อมูล).....	3
● การใช้งาน list: การสร้างและใช้งานลิสต์ในภาษา Python.....	3
- การเพิ่มสมาชิก: วิธีการเพิ่มข้อมูลลงในลิสต์.....	3
- list slicing: การเข้าถึงส่วนหรือสมาชิกของลิสต์โดยใช้สัญลักษณ์ slicing.....	4
- การนำ list มาต่อ กัน: วิธีการรวมลิสต์เข้าด้วยกัน.....	5
● การใช้งาน string: การจัดการ string ในภาษา Python.....	6
- split string: การแบ่ง string เป็นส่วนๆ โดยใช้ตัวแบ่งที่กำหนด	
- วิธีรวม string กลับ: การเชื่อมต่อ string ให้เป็นรูปแบบที่ต้องการ	
5. ความรู้เบ็องตันเกี่ยวกับฟังก์ชัน.....	8
● หน้าที่และความสำคัญของฟังก์ชัน: การใช้ฟังก์ชันเพื่อสร้างโปรแกรมที่มีโครงสร้างและเรียบง่าย ลดการเขียนโค้ดซ้ำซ้อน.....	8
● การเปรียบเทียบระหว่างฟังก์ชันในการเขียนโปรแกรมกับฟังก์ชันในคณิตศาสตร์ การเรียนรู้ การใช้งานฟังก์ชันในบทบาทต่างๆ.....	8

● รูปแบบและส่วนประกอบของการเขียนฟังก์ชัน: การสร้างฟังก์ชันในภาษา Python รวมถึงการใช้งานพารามิเตอร์และการส่งค่ากลับ.....	8
● การเว้นวรรค (indent) และการกำหนดค่า default ให้ฟังก์ชันการแสดงโค้ดให้มีความเรียบง่าย.....	9
6. สัญลักษณ์ที่จำเป็นต้องใช้ในการเขียน Program ด้วยภาษา Python รู้จักการใช้งานสัญลักษณ์พื้นฐานในการเขียนโปรแกรม.....	8
7. เรียนรู้เกี่ยวกับ loop วนซ้ำเบื้องต้น.....	10
● การใช้งาน loop ในภาษา Python: การสร้าง loop เพื่อดำเนินการซ้ำโค้ด	
● ตัวอย่างการเขียน loop: การใช้งาน loop เพื่อการประมวลผลข้อมูลที่มีความซับซ้อน	
● การสั่ง loop ให้ print แบบไม่เว้นบรรทัด: การใช้งาน loop ให้ print ข้อความแบบไม่เว้นบรรทัดโดยใช้ key argument และค่า default สำหรับตัวมันเองคือ end="\n"	
8. (Ethic1) การเขียนโปรแกรมแบบ Clean Code	
● หลักสำคัญ 10 ข้อสำหรับ Clean Code ในภาษา Python	
- Use long descriptive names that are easy to read.	
- Use descriptive intention revealing names.	
- Avoid using ambiguous shorthand.	
- Always use the same vocabulary.	
- Start tracking codebase issues in your editor.	
- Don't use magic numbers.	
- Be consistent with your function naming convention.	
- Functions should do one thing and do it well.	
- Do not use flags or Boolean flags.	
- Do not add redundant context.	

Variables

```
In [ ]: 3.14159265359
```

```
Out[ ]: 3.14159265359
```

หลักการตั้งชื่อตัวแปรเบื้องต้น

1. ตั้งให้สื่อ
2. ภาษาอังกฤษ
3. ใช้ตัวเลขได้แต่ห้ามขีนต้นด้วยตัวเลข
4. ห้ามเว้นวรรค

```
In [ ]: pi = 3.14159265359
```

```
In [ ]: pi
```

```
Out[ ]: 3.14159265359
```

int : จำนวนเต็ม

```
In [ ]: a = 10
```

```
In [ ]: print(a)
```

```
10
```

float : จำนวนจริง (ทศนิยม)

```
In [ ]: b = 10.  
print(b)
```

```
10.0
```

ตัวอักษร (char (character)) ข้อความ (text หรือ string)

.. # hashtag, sharp ใช้สำหรับคอมเม้น

```
In [ ]: c = 'ธนพงศ์' # เราจะบอกว่าตัวแปรเป็นตัวอักษรหรือข้อความโดยการใช้ single quote ' หรือ double
```

```
print(c)
```

```
ธนพงศ์
```

ตัวเลขที่เป็น string ไม่สามารถนำมา + - x / กับตัวเลขได้

```
In [ ]: d = '10'  
print(d)
```

```
10
```

```
In [ ]: d + 1
```

```
-----  
TypeError                                         Traceback (most recent call last)  
<ipython-input-9-2d39be9af043> in <module>()  
      1 d + 1  
  
TypeError: must be str, not int
```

variable casting (การเปลี่ยนชนิดของข้อมูล)

```
In [ ]: int(d) + 1
```

```
Out[ ]: 11
```

```
In [ ]: print(a)  
print(float(a))  
print(str(a))
```

```
10  
10.0  
10
```

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

```
In [ ]: a+b
```

```
Out[ ]: 20.0
```

```
In [ ]: ab = a+b  
print(ab)
```

```
20.0
```

% คือหมาย modulo

```
In [ ]: 5%3
```

```
Out[ ]: 2
```

```
In [ ]: 7%3
```

```
Out[ ]: 1
```

คำสั่ง print แบบพิเศษ (การ format string)

```
In [ ]: print('ตัวแปร') # สิ่งที่อยู่ข้างในวงเล็บคือ ตัวแปร และ string  
ตัวแปร
```

```
In [ ]: print(f'% คือการหารเอาเศษ เช่น 7%3 = {7%3}') # เพิ่ม f หน้า 'string' และใช้ {} ใส่ code  
% คือการหารเอาเศษ เช่น 7%3 = 1
```

```
In [ ]: print(f'% คือการหารเอาเศษ เช่น 7%3 = {7%3} \\  
        แต่\n/ คือการหารปกติ เช่น 7/3 = {7/3} \\  
        และ\n// คือการหารเอาส่วน เช่น 7//3 = {7//3} \\  
        หรือ\nใช้ int() เพื่อหารเอาส่วน เช่น int(7/3) = {int(7/3)}' # \n คือการขึ้นบรรทัดใหม่
```

```
% คือการหารเศษ เช่น 7%3 = 1 \          แต่
/ คือการหารปักติ เช่น 7/3 = 2.333333333333335 \          และ
// คือการหารเอาส่วน เช่น 7//3 = 2 \          หรือ
ใช้ int() เพื่อหารเอาส่วน เช่น int(7/3) = 2
```

DATA STRUCTURE (โครงสร้างข้อมูล)

List คือ การเอาตัวแปรหลายๆตัวมาเรียงกัน

list สามารถสร้างได้ 2 แบบ ดังนี้

แบบที่1 > square brackets

```
In [ ]: list_a = []
print(list_a)

[]
```

```
In [ ]: list_b = [1,5,'v']
print(list_b)

[1, 5, 'v']
```

ลำดับที่อยู่ใน list มีความสำคัญ (ลำดับใน list เริ่มจาก 0,1,2,...)

```
In [ ]: list_b[1]

Out[ ]: 5
```

แบบที่2

```
In [ ]: list_c = list()
print(list_c)

[]
```

append() เพิ่มสมาชิกใน list

```
In [ ]: list_b.append('u')
print(list_b)

[1, 5, 'v', 'u']
```

```
In [ ]: list_b.pop() ### ถึงสมาชิกที่สุดท้ายออกจาก list

Out[ ]: 'u'
```

```
In [ ]: list_b

Out[ ]: [1, 5, 'v']
```

```
In [ ]: list_b.append(list_a)
print(list_b)

[1, 5, 'v', []]
```

String > list of characters

```
In [ ]: t = 'python is easy'  
t
```

```
Out[ ]: 'python is easy'
```

```
In [ ]: len(list_b) # Len คือคำสั่งตรวจสอบความยาวของ List (จำนวนสมาชิก)
```

```
Out[ ]: 4
```

```
In [ ]: len(t)
```

```
Out[ ]: 14
```

access a member of a list (list&string)

โดยการเริ่มนับจะเริ่มนับจาก 0

0 คือสมาชิกตัวแรก , -1 คือสมาชิกตัวสุดท้าย

```
In [ ]: t[1]
```

```
Out[ ]: 'y'
```

```
In [ ]: t[-1]
```

```
Out[ ]: 'y'
```

```
In [ ]: t[-4]
```

```
Out[ ]: 'e'
```

ตัวที่อยู่ข้างใน [] เราเรียกว่า index (ตัวชี้)

List slicing สามารถทำได้โดยใช้ colon :

[a:b] -> [a,b)

$-2 \leq x < 3$ เที่ยนได้ว่า $\{x | -2 \leq x < 3\}$ หรือเที่ยนได้ว่า $[-2, 3)$



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



```
In [ ]: t = 'python is easy'
```

```
In [ ]: print(t)
print(t[7:9])
```

```
python is easy
is
```

ถ้าเว้นว่างหน้า : หมายความว่า เริ่มตั้งแต่ตัวแรก

ถ้าเว้นว่างหลัง : หมายความว่า ไปจนถึงตัวสุดท้าย

```
In [ ]: print(t)
print(t[:6])
print(t[10:])
print(t[-4:])
print(t[:])
```

```
python is easy
python
easy
easy
python is easy
```

```
In [ ]: print(t[::-2]) ## : ตัวที่สอง กำหนด step
pto ses
```

```
In [ ]: print(list(range(10)))
print(list(range(10))[::2])
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[0, 2, 4, 6, 8]
```

```
In [ ]: print(t[2::2])
to ses
```

เราสามารถเอา list มาต่อกันได้ด้วย +

```
In [ ]: t + '??'
```

```
Out[ ]: 'python is easy??'
```

```
In [ ]: t + list_b ## ไม่สามารถ加 List ป กติมาต่อกับ string ได
```

```
-----  
TypeError                                         Traceback (most recent call last)  
<ipython-input-56-8ceeb7529eac> in <module>()  
----> 1 t + list_b ## ไม่สามารถ加 List ป กติมาต่อกับ string ได  
  
TypeError: must be str, not list
```

```
In [ ]: list_b + list_a
```

```
Out[ ]: [1, 5, 'v', []]
```

การแบ่ง string ตามสัญลักษณ์ที่กำหนด -> split string

```
In [ ]: t.split(' ')
```

```
Out[ ]: ['python', 'is', 'easy']
```

```
In [ ]: time = '12:30:15'
```

```
In [ ]: time.split(':')
```

```
Out[ ]: ['12', '30', '15']
```

```
In [ ]: t_sp = t.split(' ')
```

```
print(t_sp)
```

```
## วิธีรวมกลับ
```

```
print(':'.join(t_sp))
```

```
['python', 'is', 'easy']  
python:is:easy
```

HW คำนวณเวลาเป็นวินาทีของเวลาต่อไปนี้โดยใช้คำสั่ง split() ช่วย (print ออกมาให้สวยงาม)

12:30:15

13:41:07

12:53:15

00:59:25

11:11:11

16:06:09

ให้สร้างฟังก์ชันคำนวณเวลาเป็นวินาทีของเวลาใดๆ และ print ออกมาให้สวยงาม

ให้ สร้าง list ของ เวลา
['12:30:15' , '13:41:07' ,....] และวนลูปเรียกฟัง
ก์ชันคำนวณเวลาเป็น วินาที

commit ว่า HW3

In []:

```
In [ ]: name = 'กัதวรรธน์'
surname = 'ใจเที่ยง'
ID = '603021866-7'

b= f'ชื่อ {name} นามสกุล {surname} รหัส {ID}'
print(b)
```

ชื่อ กัதวรรธน์ นามสกุล ใจเที่ยง รหัส 603021866-7

Function

Function template

ทำหน้าที่รับ input มาประมวลผลออกมาระบุ output

$$f(x) = y$$

def คือการกำหนดฟังก์ชัน

backtick (`) ==> กด ~ ค้าง, alt - 9>6 (full keyboard with number)

tilde (~)

curly bracket ({ })

square bracket ([])

```
def function_name(_Input_):
    do_something with _Input_ to get _Output_
    return _Output_
```

function มีส่วนสำคัญทั้งหมด 4 ส่วน

1. บอก python ว่าเราจะเขียนฟังก์ชัน ชื่ออะไร `def function_name():` (ขาดไม่ได้)
2. กำหนดตัวแปรที่จะเป็น input `_Input_` (ขาดได้)
3. ส่วนประมวลผล `do_something with _Input_ to get _Output_` (ขาดไม่ได้)
4. ส่วน output `return _Output_` (ขาดได้)

ตัวอย่างการเขียน normal function

```
In [ ]: def print_name(surname, ID, name):
        st = f'ชื่อ {name} นามสกุล {surname} รหัส {ID}'
        return st
```

```
In [ ]: print_name('อินทร์', '64xxxxxx', 'ธนพงศ์')
```

```
Out[ ]: 'ชื่อ ธนพงศ์ นามสกุล อินทร์ รหัส 64xxxxxx'
```

```
In [ ]: print(print_name(name='กัญจน์', surname='ปราสาคุณ', ID='603021855-2'))
```

ชื่อ กัญจน์ นามสกุล ปราสาคุณ รหัส 603021855-2

เราใช้ เว้นวรรค (indent) เพื่อบอกขอบเขตของโปรแกรม

ฟังก์ชันไม่จำเป็นต้องมี output

```
In [ ]: def print_name2(surname, ID, name):
    st = f'ชื่อ {name} นามสกุล {surname} รหัส {ID}'
    print(st)
```

```
In [ ]: print_name2(name='กาญจนา', surname='ประสาคุณ', ID='603021855-2')
```

ชื่อ กาญจนา นามสกุล ประสาคุณ รหัส 603021855-2

ฟังก์ชันไม่จำเป็นต้องมี input

```
In [ ]: def Pi():
    return 3.14159265359
```

```
In [ ]: Pi()*(2**2) # หาพื้นที่ของวงกลมที่มีรัศมีเท่ากับ Pi * r^2
```

```
Out[ ]: 12.56637061436
```

เราสามารถกำหนดค่า default ให้กับฟังก์ชันได้

input ของ function ใน python มีสองแบบ input ที่จำเป็นต้องใส่ กับ input ที่ไม่จำเป็นต้องใส่ (มีค่า default)

เราต้องเรียง input ที่จำเป็นต้องใส่ขึ้นก่อน

```
In [ ]: def print_2lines_default(name, surname, ID, grade='F'):
    st = f'ชื่อ {name} นามสกุล {surname} รหัส {ID}'
    print(st)
    st2=f'เกรดวิชา Data Viz >>> {grade}'
    print(st2)
```

```
In [ ]: print_2lines_default('ธันยการต์', 'พวงมาลัย', '613020551-8')
```

ชื่อ ธันยการต์ นามสกุล พวงมาลัย รหัส 613020551-8
เกรดวิชา Data Viz >>> F

```
In [ ]: print_2lines_default('ธันยการต์', 'พวงมาลัย', '613020551-8', 'A')
```

ชื่อ ธันยการต์ นามสกุล พวงมาลัย รหัส 613020551-8
เกรดวิชา Data Viz >>> A

LOOP การวนซ้ำ

for เป็นคำที่ใช้บอก python ว่าเรากำลังเขียน loop โดย for จะวนดึงสมาชิกจาก `listA` มาทำ process `do_something`

```
for each_member in listA :  
    do_something
```

คำสั่ง `for` เป็นคำสั่งวนซ้ำที่ใช้ควบคุมการทำงานซ้ำๆ ในจำนวนรอบที่แน่นอน

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

this member = 1 after process = 1
this member = 2 after process = 4
this member = 3 after process = 9

จบวันที่ 7 มค 2564

ตัวอย่างการใช้คำสั่ง `for` ในการวนซ้ำค่าของ `i` เมื่อ `i` คือ 'Thanapong' และคำสั่ง `print(f'{i} -> /')` เพื่อกำหนดให้ผลลัพธ์แสดงสัญลักษณ์ `->` / จากนั้นเพิ่มคำสั่ง `print(' ')` อีกครั้งเพื่อให้ผลลัพธ์มีการเว้นวรรค

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

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

```
In [ ]: for i in 'Thanapong':  
    print(f'{i} -> / ',end = '') #สั่งให้ print แบบ ไม่เว้นวรรคด้วย
```

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

```
In [ ]: for i in 'Thanapong':  
    print(f'{i} -> / ',end = ',') #สั่งให้ print แบบ ไม่เว้นวรรคด้วย
```

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

บทที่ 3

(โปรแกรมวนซ้ำและการใช้เงื่อนไขในภาษาไพรอน)

บทที่ 3: โปรแกรมวนซ้ำและการใช้เงื่อนไขในภาษาไพธอน

1.	เรียนรู้เกี่ยวกับคำสั่ง range() และการใช้งานในการสร้าง list ตัวเลขแบบอัตโนมัติ.....	1
●	คำสั่ง range(): การใช้งานและเข้าใจวิธีการใช้ range() ในการสร้างลิสต์ตัวเลขแบบอัตโนมัติ	
●	Key argument ของ range(): การกำหนดตัวเลขที่ต้องการ เริ่ม(start), หยุด(stop) และ step	
2.	การเขียนวนลูปซ้อนลูป: วิธีการใช้งานวนลูปซ้อนลูป การจัดเรียงลำดับการทำงานของโค้ดแต่ละบรรทัดภายในลูปเพื่อการประมวลผลข้อมูลที่มีโครงสร้างซับซ้อน และยกตัวอย่างการเขียนวนลูปซ้อนลูป.....	1
3.	การเขียนวนลูปในฟังก์ชัน: การใช้งานวนลูปภายในฟังก์ชันเพื่อประมวลผลข้อมูล.....	2
4.	การเขียนวนลูปใน Dictionary: การใช้งานวนลูปภายใน Dictionary เพื่อประมวลผลข้อมูล.....	3
5.	การเขียนเงื่อนไข / โปรแคเงื่อนไขในภาษา Python.....	3
●	การเขียนเงื่อนไข: การใช้งาน if, elif, และ else เพื่อทำเงื่อนไขในโปรแกรม	
-	ข้อจำกัดของการใช้งานการเขียนเงื่อนไข	
-	สัญลักษณ์ที่ใช้ในการเขียนเงื่อนไข	
-	การใช้ logical expression (True/False) ในการเขียนเงื่อนไข	
●	โปรแคเงื่อนไขในภาษา Python: การใช้งานโปรแคเงื่อนไขเพื่อการตัดสินใจและปรับแก้โค้ด	
6.	การใช้งาน Dictionary.....	8
●	การใช้งาน Dictionary: ข้อกำหนดในการใช้งาน Dictionary การสร้างและใช้งานโครงสร้างข้อมูลแบบ Dictionary ใน Python	
●	การเพิ่มสมาชิก: วิธีการเพิ่มข้อมูลใหม่ใน Dictionary.....	8
●	คำสั่งที่ใช้บ่อย: คำสั่งและเทคนิคที่ใช้งานบ่อยในการจัดการ Dictionary การใช้งานคำสั่ง .keys() และ .values().....	8
7.	การใช้งาน Matrix หรือ Numpy Array ในภาษา Python.....	9
●	Numpy Array คืออะไร วิธีเรียกใช้งาน package	
●	การแปลง: การแปลงข้อมูลจาก list ให้เป็น Numpy Array และกลับ ด้วยคำสั่ง numpy.array().....	9
●	การซื้อค่า: วิธีการเข้าถึงและแก้ไขค่าใน Matrix หรือ Numpy Array.....	9

● การดำเนินการ: การทำงานกับ Matrix หรือ Numpy Array เช่น การบวก ลบ คูณ และหาร.....	10
● การตรวจสอบขนาด: วิธีการตรวจสอบขนาดของ Matrix หรือ Numpy Array.....	10
● การหาดีเทอร์มิเนนต์: วิธีการหาดีเทอร์มิเนนต์ของ Matrix หรือ Numpy Array.....	11
● การตัด: การตัดหรือการแบ่งข้อมูล Matrix หรือ Numpy Array เป็นส่วนย่อย.....	11
● การสร้าง: วิธีการสร้าง Matrix หรือ Numpy Array แบบสุ่มหรือแบบกำหนดค่าเองโดยใช้คำสั่ง numpy.zeros, numpy.ones, numpy.random, numpy.random.rand, numpy.random.randn, numpy.random.choice (การใช้งาน Parameter: size, replace, p) อธิบายการใช้งานและยกตัวอย่างการใช้งานของแต่ละคำสั่ง.....	12
● การดำเนินการคูณ matrix แบบ scalar multiplication: การคูณ Matrix หรือ Numpy Array ด้วยค่าคงที่.....	12
● สร้างฟังก์ชันการคูณ matrix กับ matrix: อธิบายเกี่ยวกับขั้นตอนการสร้างแต่ละขั้นตอนลำดับการทำงานของโค้ดแต่ละบรรทัด การคำนวณของโค้ดภายในลูปที่สร้างและวิธีการใช้งานฟังก์ชัน.....	13

สร้าง list ตัวเลขแบบอัตโนมัติ range()

```
In [ ]: range5_output = range(5)
print(list(range5_output))
```

[0, 1, 2, 3, 4]

```
In [ ]: for i in range(100):
          print('v',end=' ')
```

```
In [ ]: list(range(1,11)) #เริ่มต้นเป็นวงทีบ จนเป็นวงกลมไปรัง เหมือน list slicing
```

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

```
In [ ]: list(range(-3,20,4))
```

```
Out[1]: [-3, 1, 5, 9, 13, 17]
```

loop in loop

สามารถใช้ for loop ข้อนักส่องรุนได้

```
In [ ]: for name1 in range(2,5): #(2, 3, 4)
          print(f'now name1 = {name1}')
          for name2 in range(1,13):
              print(name1, ' x ', name2, ' = ', name1*name2)
          print('the inner loop is end')
```

```

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

```

loop in function

การวนลูปในฟังก์ชันฯ

```
In [ ]: def print_feelings(I, friendS, feeling = 'คิดถึง'):
    for friend in friendS:
        print(f'{I} {feeling} {friend}')
```

```
In [ ]: list_friends = ['พี่นุ่ม', 'ดายน์', 'แตงโม', 'ฝน', 'พี่เจ๊ท', 'น้องแคมป์']
```

```
In [ ]: print_feelings('เจมส์', list_friends)
```

เจมส์ คิดถึง พี่นุ่ม
 เจมส์ คิดถึง ดายน์
 เจมส์ คิดถึง แตงโม¹
 เจมส์ คิดถึง ฝน
 เจมส์ คิดถึง พี่เจ๊ท
 เจมส์ คิดถึง น้องแคมป์

```
In [ ]: printFeelings('เจมส์', listFriends, feeling='รัก')
```

เจมส์ รัก พี่บุน
เจมส์ รัก ดายัน
เจมส์ รัก แแตงโน
เจมส์ รัก ฝน
เจมส์ รัก พี่เจ็ท
เจมส์ รัก น้องแคมป์

```
In [ ]: printFeelings('เจมส์', listFriends, 'รัก')
```

เจมส์ รัก พี่บุน
เจมส์ รัก ดายัน
เจมส์ รัก แแตงโน
เจมส์ รัก ฝน
เจมส์ รัก พี่เจ็ท
เจมส์ รัก น้องแคมป์

looping in dict

ทำการวนลูปในค่า dict โดยที่เรากำหนดตัว key และ value ขึ้นมา ได้ดังนี้

```
In [ ]: name_grade = {'พันธิพา' : 'B', 'ศุภนิตา' : 'C', 'ปนัดดา' : 'A', 'ทรงกลด' : 'F'}  
name_grade
```

```
Out[ ]: {'พันธิพา': 'B', 'ศุภนิตา': 'C', 'ปนัดดา': 'A', 'ทรงกลด': 'F'}
```

```
In [ ]: name_grade['ปนัดดา']
```

```
Out[ ]: 'A'
```

```
In [ ]: for i in name_grade.keys():  
    print(i)
```

พันธิพา
ศุภนิตา
ปนัดดา
ทรงกลด

```
In [ ]: name_grade.keys()
```

```
Out[ ]: dict_keys(['พันธิพา', 'ศุภนิตา', 'ปนัดดา', 'ทรงกลด'])
```

```
In [ ]: for i in name_grade.keys():  
    print(f'{i} ได้เกรด {name_grade[i]}')
```

พันธิพา ได้เกรด B
ศุภนิตา ได้เกรด C
ปนัดดา ได้เกรด A
ทรงกลด ได้เกรด F

conditional คือ เงื่อนไข / ประโยคเงื่อนไข

True / False

การวนลูปโดยมีเงื่อนไขต่างๆ มาเกี่ยวข้อง โดยมี pattern ดังนี้

```
if condition1 :  
    do something  
elif condition2 : # elif คือ else if
```

```
    do another thing
else:
    do ...
```

```
In [ ]: print(list_friends)
['พี่บูม', 'ดายน์', 'แตงโน้ม', 'ฝน', 'พี่เจ็ท', 'น้องแคมป์']
```

```
In [ ]: for name1 in list_friends :
        for name2 in list_friends :
            print(name1, 'รัก', name2)
```

```
พี่บูม รัก พี่บูม
พี่บูม รัก ดายน์
พี่บูม รัก แตงโน้ม
พี่บูม รัก ฝน
พี่บูม รัก พี่เจ็ท
พี่บูม รัก น้องแคมป์
ดายน์ รัก พี่บูม
ดายน์ รัก ดายน์
ดายน์ รัก แตงโน้ม
ดายน์ รัก ฝน
ดายน์ รัก พี่เจ็ท
ดายน์ รัก น้องแคมป์
แตงโน้ม รัก พี่บูม
แตงโน้ม รัก ดายน์
แตงโน้ม รัก แตงโน้ม
แตงโน้ม รัก ฝน
แตงโน้ม รัก พี่เจ็ท
แตงโน้ม รัก น้องแคมป์
ฝน รัก พี่บูม
ฝน รัก ดายน์
ฝน รัก แตงโน้ม
ฝน รัก ฝน
ฝน รัก พี่เจ็ท
ฝน รัก น้องแคมป์
พี่เจ็ท รัก พี่บูม
พี่เจ็ท รัก ดายน์
พี่เจ็ท รัก แตงโน้ม
พี่เจ็ท รัก ฝน
พี่เจ็ท รัก พี่เจ็ท
พี่เจ็ท รัก น้องแคมป์
น้องแคมป์ รัก พี่บูม
น้องแคมป์ รัก ดายน์
น้องแคมป์ รัก แตงโน้ม
น้องแคมป์ รัก ฝน
น้องแคมป์ รัก พี่เจ็ท
น้องแคมป์ รัก น้องแคมป์
```

```
In [ ]: for name1 in list_friends:
        for name2 in list_friends:
            if name1 != name2: #ไม่ปรินชื่อคนเดียวกันข้า
                print(name1, 'รัก', name2)
```

พี่บูม รัก ดายน์
พี่บูม รัก แตงโน
พี่บูม รัก ฟุน
พี่บูม รัก พี่เจ็ท
พี่บูม รัก น้องแคมป์
ดายน์ รัก พี่บูม
ดายน์ รัก แตงโน
ดายน์ รัก ฟุน
ดายน์ รัก พี่เจ็ท
ดายน์ รัก น้องแคมป์
แตงโน รัก พี่บูม
แตงโน รัก ดายน์
แตงโน รัก ฟุน
แตงโน รัก พี่เจ็ท
แตงโน รัก น้องแคมป์
ฟุน รัก พี่บูม
ฟุน รัก ดายน์
ฟุน รัก แตงโน
ฟุน รัก พี่เจ็ท
ฟุน รัก น้องแคมป์
พี่เจ็ท รัก พี่บูม
พี่เจ็ท รัก ดายน์
พี่เจ็ท รัก แตงโน
พี่เจ็ท รัก ฟุน
พี่เจ็ท รัก น้องแคมป์
น้องแคมป์ รัก พี่บูม
น้องแคมป์ รัก ดายน์
น้องแคมป์ รัก แตงโน
น้องแคมป์ รัก ฟุน
น้องแคมป์ รัก พี่เจ็ท

```
In [ ]: for name1 in list_friends:  
    for name2 in list_friends:  
        if name1 == name2: #ปริ้นชื่อคนเดียวกันซ้ำ  
            print(name1, 'รัก', name2)
```

พี่บูม รัก พี่บูม
ดายน์ รัก ดายน์
แตงโน รัก แตงโน
ฟุน รัก ฟุน
พี่เจ็ท รัก พี่เจ็ท
น้องแคมป์ รัก น้องแคมป์

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

พี่บูม รัก ดายน์
พี่บูม รัก แตงโนء
พี่บูม รัก ฟุน
พี่บูม รัก พี่เจ็ท
พี่บูม รัก น้องแคมป์
ดายน์ รัก พี่บูม
ดายน์ รัก แตงโนء
ดายน์ รัก ฟุน
ดายน์ รัก พี่เจ็ท
ดายน์ รัก น้องแคมป์
แตงโนء รัก พี่บูม
แตงโนء รัก ดายน์
แตงโนء รัก ฟุน
แตงโนء รัก พี่เจ็ท
แตงโนء รัก น้องแคมป์
ฟุน รัก พี่บูม
ฟุน รัก ดายน์
ฟุน รัก แตงโนء
ฟุน รัก พี่เจ็ท
ฟุน รัก น้องแคมป์
พี่เจ็ท รัก พี่บูม
พี่เจ็ท รัก ดายน์
พี่เจ็ท รัก แตงโนء
พี่เจ็ท รัก ฟุน
พี่เจ็ท รัก น้องแคมป์
น้องแคมป์ รัก ดายน์
น้องแคมป์ รัก แตงโนء
น้องแคมป์ รัก ฟุน
น้องแคมป์ รัก พี่เจ็ท

`==, !=, >=, <=, <, >`

In []: `'พี่บูม' == 'พี่เจ็ท'`

Out[]: `False`

In []: `'พี่บูม' != 'พี่เจ็ท'`

Out[]: `True`

หากหลังเงื่อนไข True ถูกต้องจะทำการ print ข้อความออกมาว่า Yes

In []: `if True:
 print('Yes')`

Yes

In []: `if False:
 print('No')`

หากหลังเงื่อนไข False ไม่ถูกต้องจะไม่ปริ้นข้อความใดๆออกมา

ซึ่งจะต้องใช้เงื่อนไขนี้ จึงจะได้ปริ้นค่าเงื่อนไขนั้นออกมา ดังนี้

In []: `if False:
 print('Yes')
else:
 print('No')`

No

HW

เขียน function ตัดเกรด โดยที่ input เป็นคะแนน(0-100) , output เป็นเกรด (F-A)

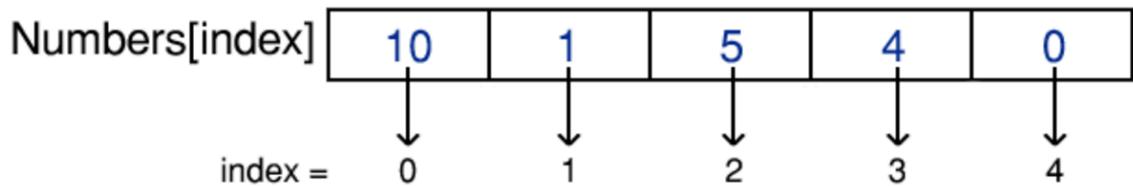
วนลูปตัดเกรด input = [1,50,65,90,101,-5,49]

กำหนด

ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 0 แต่น้อยกว่า 50 จะได้เกรด F
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 50 แต่น้อยกว่า 55 จะได้เกรด D
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 55 แต่น้อยกว่า 60 จะได้เกรด D+
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 60 แต่น้อยกว่า 65 จะได้เกรด C
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 65 แต่น้อยกว่า 70 จะได้เกรด C+
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 70 แต่น้อยกว่า 75 จะได้เกรด B
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 75 แต่น้อยกว่า 80 จะได้เกรด B+
ถ้าคะแนนอยู่ระหว่างมากกว่าหรือเท่ากับ 80 จะได้เกรด A
แต่ว่าค่าคะแนนจะต้องไม่ต่ำกว่า 0 และมากกว่า 100 ไม่รันจะเกิด error

Finish

ที่เรียนไปแล้วมี 1 ตัวคือ list หน้าตาประมาณนี้



Dictionary

สามารถ index ด้วยอะไรมีได้ แต่ลำดับจะหายไป

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

```
In [ ]: ex_dict['stat']
```

```
Out[ ]: 5
```

```
In [ ]: ex_dict[10]
```

```
-----  
KeyError                                                 Traceback (most recent call last)  
<ipython-input-8-008be89af7ca> in <module>()  
      ----> 1 ex_dict[10]  
  
KeyError: 10
```

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

```
In [ ]: ex2_dict[2]
```

```
Out[ ]: 'two'
```

การเพิ่มสมาชิกใน dict

```
In [ ]: ex2_dict[0] = 'zero'
```

```
In [ ]: ex2_dict
```

```
Out[ ]: {0: 'zero', 1: 'one', 2: 'two', 3: 'three'}
```

คำสั่ง

```
In [ ]: ex2_dict.keys()
```

```
Out[ ]: dict_keys([1, 2, 3, 0])
```

```
In [ ]: for index in ex2_dict.keys():  
        print(index)
```

```

1
2
3
0

In [ ]: ex2_dict.values()

Out[ ]: dict_values(['one', 'two', 'three', 'zero'])

In [ ]: index

Out[ ]: 0

```

dict เอาไปใช้ร่วมกับ Dataframe ของ pandas

Numpy Array (array n มิติ)

3D array

1D array

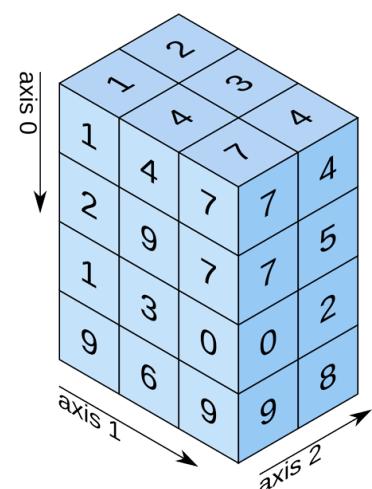
7	2	9	10
---	---	---	----

axis 0 →

2D array

5.2	3.0	4.5
9.1	0.1	0.3

shape: (2, 3)



shape: (4, 3, 2)

numpy คือ package ที่ทำงานเกี่ยวกับ array ของตัวเลข (ที่มีคนเขียนขึ้นมาและนิยมใช้)

numpy -> number python

```
In [ ]: import numpy #วิธีเรียกใช้งาน package
```

สร้าง numpy array

```

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

[[5.2 3.  4.5]
 [9.1 0.1 0.3]]

```

การซื้อขายใน numpy array

```
In [ ]: ex_2d_array[1,2]
```

```
Out[ ]: 0.3
```

```
In [ ]: list_x = [[5.2,3.0,4.5],[9.1,0.1,0.3]] #การสร้าง List  
print(len(list_x))  
print(list_x[1])  
print(list_x[1][2])  
  
2  
[9.1, 0.1, 0.3]  
0.3
```

Operations

```
In [ ]: ex2_2d_array = numpy.array([[1,0,0],[0,0,1]])  
print(ex2_2d_array)  
print(ex_2d_array)  
  
[[1 0 0]  
 [0 0 1]]  
[[5.2 3. 4.5]  
 [9.1 0.1 0.3]]
```

```
In [ ]: ex_2d_array.shape #ตรวจสอบขนาดของ matrix  
Out[ ]: (2, 3)
```

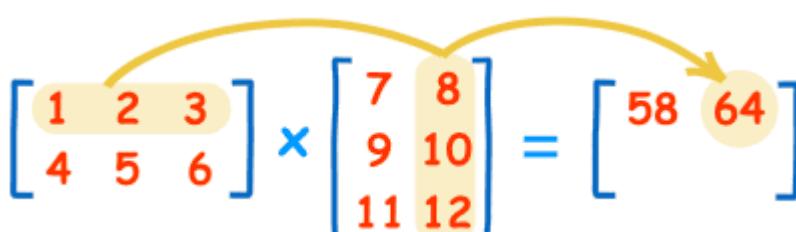
```
In [ ]: ex_2d_array + ex2_2d_array  
Out[ ]: array([[6.2, 3. , 4.5],  
 [9.1, 0.1, 1.3]])
```

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

```
In [ ]: ex_2d_array - ex2_2d_array  
Out[ ]: array([[ 4.2, 3. , 4.5],  
 [ 9.1, 0.1, -0.7]])
```

```
In [ ]: ex_2d_array * ex2_2d_array #การคูณแบบ array เอาร์มันนากัน  
Out[ ]: array([[5.2, 0. , 0. ],  
 [0. , 0. , 0.3]])
```

matrix multiplication (dot product)

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


```
In [ ]: numpy.dot(ex_2d_array,ex2_2d_array)
```

```
-----  
ValueError                                                 Traceback (most recent call last)  
<ipython-input-36-d923e9f0df4f> in <module>()  
----> 1 numpy.dot(ex_2d_array,ex2_2d_array)  
  
<__array_function__ internals> in dot(*args, **kwargs)  
  
ValueError: shapes (2,3) and (2,3) not aligned: 3 (dim 1) != 2 (dim 0)
```

```
In [ ]: print(ex_2d_array)  
print(ex_2d_array.T)
```

```
[[5.2 3. 4.5]  
 [9.1 0.1 0.3]]  
[[5.2 9.1]  
 [3. 0.1]  
 [4.5 0.3]]
```

```
In [ ]: dot_mat = numpy.dot(ex_2d_array,ex2_2d_array.T)  
print(dot_mat)
```

```
[[5.2 4.5]  
 [9.1 0.3]]
```

```
In [ ]: numpy.linalg.det(dot_mat)
```

```
Out[ ]: -39.38999999999986
```

search google -> stackoverflow หรือ web ของ package สำหรับ operations อื่นๆ

matrix slicing

```
In [ ]: print(ex_2d_array)
```

```
[[5.2 3. 4.5]  
 [9.1 0.1 0.3]]
```

```
In [ ]: ex_2d_array[1,1:]
```

```
Out[ ]: array([0.1, 0.3])
```

```
In [ ]: ex_2d_array[:, :2]
```

```
Out[ ]: array([[5.2, 3. ],  
 [9.1, 0.1]])
```

จบวันที่ 14 Jan 2021

HW 5 เขียน function คูณ matrix

แล้ว test กับ matrix ขนาด

- (2,3)*(3,2)
- (4,4)*(4,1)
- (2,2)*(2,2)

zeros, ones, random

```
In [ ]: import numpy as np #import numpy และตั้งชื่อเป็น np
```

zeros

```
In [ ]: np.zeros(2)
```

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

```
In [ ]: np.zeros((2,3))
```

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

ones

```
In [ ]: np.ones((2,3))
```

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

Matrix Operation (scalar multiplication)

```
In [ ]: M_one = np.ones((2,3))
```

```
2*M_one
```

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

Random

```
In [ ]: np.random.rand(3,2) # uniform random [0,1)
```

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

```
In [ ]: np.random.randn(3,2) # sample from normal distribution mean=0 std=1
```

```
Out[ ]: array([[ 2.06762285,  0.91239845],  
               [-2.08011942, -0.46261935],  
               [ 0.66804796,  1.19419422]])
```

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

```
Out[ ]: 'a'
```

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

```
Out[ ]: array([28,  3])
```

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

```

Out[ ]: array([[ 3,   3,  11],
               [28,   1,  11]])

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

Out[ ]: array([[ 3,  11,    2],
               [100,  28,    1]])

In [ ]: np.random.choice(['A','B','C','D','F'],size=35, p = [0.1,0.2,0.4,0.29,0.01])

Out[ ]: array(['D', 'C', 'C', 'C', 'C', 'B', 'C', 'B', 'D', 'A', 'C', 'C',
               'D', 'A', 'C', 'C', 'D', 'D', 'B', 'C', 'A', 'C', 'C', 'C', 'D',
               'A', 'D', 'C', 'D', 'B', 'D', 'C', 'D', 'C'], dtype='<U1')

```

เฉลย HW Matrix Multiplication

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

```

In [ ]: def mat_mul(A,B):
          C = np.zeros((A.shape[0],B.shape[1]))#เดรีຍນขนาดຂອງ matrix ຜລລັບ
          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

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

In [ ]: mat_mul(mat1,mat2)

Out[ ]: array([[ 58.,  64.],
               [139., 154.]])

```

บทที่ 4
(การจัดการข้อมูลในรูปแบบตาราง)

บทที่ 4: การจัดการข้อมูลในรูปแบบตาราง

1. ความรู้เบื้องต้นเกี่ยวกับ Pandas ในภาษา Python

- การใช้ Pandas: การนำเข้าและใช้งานไลบรารี Pandas เพื่อการจัดการข้อมูลแบบตาราง.....1
- การหาแหล่งที่มาของข้อมูล: วิธีการหาแหล่งที่มาของข้อมูล และเรียนรู้เกี่ยวกับประเภทไฟล์ข้อมูล (CSV, Meta data) เพื่อให้รู้จักกับข้อมูลที่จะนำเข้า
- การดาวน์โหลดและนำเข้าข้อมูล csv เข้า Google Colab: ขั้นตอนการดาวน์โหลดและนำเข้าข้อมูลจากไฟล์ csv ลงใน Google Colab เพื่อการวิเคราะห์ข้อมูล.....1
- การใช้ os: การนำเข้าและใช้งานไลบรารี os เพื่อใช้ในการทำงานต่างๆที่เกี่ยวกับไฟล์ เช่น การซีไฟล์ การลบไฟล์ การสร้างโฟลเดอร์ การเชื่อม path เป็นต้น.....1
- การ set path: เพื่อใช้สำหรับซีไฟล์ที่ต้องการ.....1
- การใช้ pd.read_csv: เพื่ออ่านข้อมูลที่ต้องการนำมาใช้งานและเรียนรู้เกี่ยวกับ Parameter: encoding ใช้สำหรับกำหนดภาษาของไฟล์ที่จะอ่าน.....1

2. การตรวจสอบข้อมูล

- การแสดงข้อมูล: วิธีการแสดงข้อมูลในตารางเพื่อตรวจสอบข้อมูล.....2
 - การใช้งานคำสั่ง .head() เพื่อแสดงชื่อคอลัมน์และข้อมูลในตารางตามจำนวนแถวที่ต้องการ
 - การใช้งานคำสั่ง .shape() เพื่อใช้ตรวจสอบขนาดของข้อมูล
- การซื้อข้อมูลตาราง: การเข้าถึงและแสดงข้อมูลในตารางโดย.....3
 - การซื้อค่าในข้อมูลตารางแบบชื่อคอลัมน์เป็นตัวชี้
 - การซื้อค่าในข้อมูลตารางแบบ .iloc
- การตัดตาราง: Table slicing กรองข้อมูลที่ต้องการโดย.....4
 - การใช้ชื่อคอลัมน์ เลือกข้อมูลเฉพาะคอลัมน์ที่ต้องการ
 - การใช้ .iloc[] เลือกข้อมูลเฉพาะแถวที่ต้องการ
 - การใช้ logical expression เพื่อกรองข้อมูลที่ต้องการในตาราง

3. (Ethic2) Bias in data-Job recruitment

- เรียนรู้นิยามของ bias คือความลำเอียงหรือความเอนเอียงที่เกิดขึ้นเมื่อมีการเบี่ยงเบนจากความเป็นจริงหรือความถูกต้อง อาจเกิดขึ้นโดยตั้งใจหรือไม่ตั้งใจก็ได้

- การสำรวจ bias ในชุดข้อมูล: วิเคราะห์ชุดข้อมูลเพื่อระบุ bias ที่อาจเกิดขึ้น
- เทคนิคการประมวลผลข้อมูล: ครอบคลุมเทคนิคการประมวลผลข้อมูลที่สามารถช่วยลด bias
- algorithmic bias: bias สามารถถูกขยายผลหรือเกิดขึ้นซ้ำๆ ผ่านอัลกอริทึมและไม่เดลาการเรียนรู้ของเครื่อง
- กรณีศึกษา bias และตัวอย่าง bias จากโลกแห่งความเป็นจริง: วิเคราะห์กรณีเหล่านี้ในการอภิปรายในชั้นเรียนเพื่อระบุที่มาของ bias นัยยะด้านจริยธรรม และวิธีแก้ปัญหาที่อาจเกิดขึ้น

```
In [ ]: import pandas as pd

In [ ]: from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

In [ ]: import os

path = '/content/drive/My Drive/dataviz_2021_data'

In [ ]: covid_file_path = os.path.join(path, 'pm-20-jan-2021.csv')
print(covid_file_path)

/content/drive/My Drive/dataviz_2021_data/pm-20-jan-2021.csv

In [ ]: a = path + '/pm-18-jan-2021.csv'
print(a)

/content/drive/My Drive/dataviz_2021_data/pm-18-jan-2021.csv
```

load data to memory

```
In [ ]: data_covid = pd.read_csv(covid_file_path)
data_covid
```

Out[]:

No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	pro
0	1	1/12/2020	NaN	หญิง	61.0	China	กรุงเทพมหานคร
1	2	1/17/2020	NaN	หญิง	74.0	China	กรุงเทพมหานคร
2	3	1/22/2020	NaN	หญิง	73.0	Thailand	นครปฐม
3	4	1/22/2020	NaN	ชาย	68.0	China	กรุงเทพมหานคร
4	5	1/24/2020	NaN	หญิง	66.0	China	นนทบุรี
...
12648	12649	1/20/2021	1/19/2021	หญิง	44.0	Thailand	ชลบุรี
12649	12650	1/20/2021	1/19/2021	หญิง	52.0	Thailand	ระยอง
12650	12651	1/20/2021	1/19/2021	หญิง	23.0	Thailand	ระยอง
12651	12652	1/20/2021	1/19/2021	หญิง	29.0	Thailand	ระยอง
12652	12653	1/20/2021	1/19/2021	หญิง	22.0	Burma	ตาก

12653 rows × 10 columns

In []: `data_covid.head()`

```
Out[ ]: No. announce_date notification_date sex age nationality province_of_isolation province_o
```

```
0 1 1/12/2020 NaN หญิง 61.0 China กรุงเทพมหานคร กรุงเทพ:
```

```
1 2 1/17/2020 NaN หญิง 74.0 China กรุงเทพมหานคร กรุงเทพ:
```

```
2 3 1/22/2020 NaN หญิง 73.0 Thailand นครปฐม นนทบุรี กรุงเทพ:
```

```
3 4 1/22/2020 NaN ชาย 68.0 China กรุงเทพมหานคร กรุงเทพ:
```

```
4 5 1/24/2020 NaN หญิง 66.0 China นนทบุรี กรุงเทพ:
```

```
In [ ]: data_covid.shape
```

```
Out[ ]: (12653, 10)
```

ขีดค่าในตาราง

basic

```
In [ ]: data_covid['province_of_onset']
```

```

Out[ ]: 0      กรุงเทพมหานคร
         1      กรุงเทพมหานคร
         2      นครปฐม
         3      กรุงเทพมหานคร
         4      กรุงเทพมหานคร
         ...
         12648     ชลบุรี
         12649     ระยอง
         12650     ระยอง
         12651     ระยอง
         12652     ตาก
Name: province_of_onset, Length: 12653, dtype: object

```

```
In [ ]: data_covid['province_of_onset'][4]
```

```
Out[ ]: 'กรุงเทพมหานคร'
```

.iloc

```
In [ ]: data_covid.iloc[4,7]
```

```
Out[ ]: 'กรุงเทพมหานคร'
```

Table slicing

ตัดมาเฉพาะ column ที่ต้องการ

```
In [ ]: smaller_table = data_covid[['announce_date','province_of_onset','risk']]
```

	announce_date	province_of_onset	risk
0	1/12/2020	กรุงเทพมหานคร	คนต่างชาติเดินทางมาจากต่างประเทศ
1	1/17/2020	กรุงเทพมหานคร	คนต่างชาติเดินทางมาจากต่างประเทศ
2	1/22/2020	นครปฐม	คนต่างชาติเดินทางมาจากต่างประเทศ
3	1/22/2020	กรุงเทพมหานคร	คนต่างชาติเดินทางมาจากต่างประเทศ
4	1/24/2020	กรุงเทพมหานคร	คนต่างชาติเดินทางมาจากต่างประเทศ
...
12648	1/20/2021	ชลบุรี	State Quarantine
12649	1/20/2021	ระยอง	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้า
12650	1/20/2021	ระยอง	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้า
12651	1/20/2021	ระยอง	Cluster ระยอง
12652	1/20/2021	ตาก	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้า

12653 rows × 3 columns

ตัดเฉพาะ rows ที่ต้องการ

แบบง่าย แต่ใช้การอะไรไม่ค่อยจะได้ (มองเป็น array และตัดแบบ array)

```
In [ ]: data_covid.iloc[1:5,:]
```

```
Out[ ]: No. announce_date notification_date sex age nationality province_of_isolation province_o
```

1	2	1/17/2020	NaN	หญิง	74.0	China	กรุงเทพมหานคร	กรุงเทพฯ
---	---	-----------	-----	------	------	-------	---------------	----------

2	3	1/22/2020	NaN	หญิง	73.0	Thailand	นครปฐม	...
---	---	-----------	-----	------	------	----------	--------	-----

3	4	1/22/2020	NaN	ชาย	68.0	China	กรุงเทพมหานคร	กรุงเทพฯ
---	---	-----------	-----	-----	------	-------	---------------	----------

4	5	1/24/2020	NaN	หญิง	66.0	China	นนทบุรี	กรุงเทพฯ
---	---	-----------	-----	------	------	-------	---------	----------

แบบ advanced แต่ใช้ประโยชน์ได้เยอะกว่า (ตัดแบบใช้ logic query)

```
In [ ]: data_covid[data_covid['province_of_onset']=='ขอนแก่น']
```

Out[]:

No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	pro
180	181	3/18/2020	3/15/2020	ชาย	33.0	Thailand	ขอนแก่น
462	463	3/22/2020	3/21/2020	หญิง	36.0	Thailand	ขอนแก่น
1466	1467	3/30/2020	3/26/2020	ชาย	19.0	Thailand	ขอนแก่น
1970	1971	4/3/2020	3/31/2020	หญิง	70.0	Thailand	ขอนแก่น
2637	2638	4/15/2020	4/14/2020	หญิง	63.0	Thailand	ขอนแก่น
2673	2674	4/17/2020	4/16/2020	ชาย	68.0	Thailand	ขอนแก่น
5948	5949	12/26/2020	12/25/2020	หญิง	32.0	Thailand	ขอนแก่น
6082	6083	12/27/2020	12/26/2020	หญิง	36.0	Thailand	ขอนแก่น
9333	9334	1/7/2021	1/6/2021	ชาย	17.0	Thailand	ขอนแก่น
9334	9335	1/7/2021	1/6/2021	ชาย	20.0	Thailand	ขอนแก่น
10610	10611	1/12/2021	1/11/2021	ชาย	17.0	Thailand	ขอนแก่น
11517	11518	1/16/2021	1/15/2021	ชาย	12.0	Thailand	ขอนแก่น
11697	11698	1/17/2021	1/16/2021	หญิง	17.0	Thailand	ขอนแก่น
11698	11699	1/17/2021	1/16/2021	หญิง	37.0	Thailand	ขอนแก่น

```
No. announce_date notification_date sex age nationality province_of_isolation pro
```

การซื้อค่าในตารางด้วย `data_covid[x]` ค่า `x` สามารถมีค่าที่ใส่เข้าไปได้ 2 แบบ

1. ชื่อ column
2. ใส่ rows ที่ต้องการ (ใส่ list ที่มีขนาดเท่ากับจำนวนแถว และบวกกว่าເອົາ ແກ້ວໃຫນບ້າງໂດຍໃຫ້ logical expression (True / False))

```
In [ ]: eight_rows_covid = data_covid.iloc[:8,:] # ตารางຕົວຢ່າງ  
eight_rows_covid
```

Out[]: No. announce_date notification_date sex age nationality province_of_isolation province_o

0 1 1/12/2020 NaN หญิง 61.0 China กรุงเทพมหานคร กรุงเทพ:

1 2 1/17/2020 NaN หญิง 74.0 China กรุงเทพมหานคร กรุงเทพ:

2 3 1/22/2020 NaN หญิง 73.0 Thailand นครปฐม นครปฐม :

3 4 1/22/2020 NaN ชาย 68.0 China กรุงเทพมหานคร กรุงเทพ:

4 5 1/24/2020 NaN หญิง 66.0 China นนทบุรี กรุงเทพ:

5 6 1/25/2020 NaN หญิง 33.0 China กรุงเทพมหานคร กรุงเทพ:

6 7 1/26/2020 NaN หญิง 57.0 China กรุงเทพมหานคร กรุงเทพ:

7 8 1/26/2020 NaN หญิง 73.0 China ประจวบคีรีขันธ์ ประจวบ:

```
In [ ]: eight_rows_covid[[True, False, True, True, False, True, False, False]] #2. ใส่ rows ที่ต้องการ
```

```
Out[ ]: No. announce_date notification_date sex age nationality province_of_isolation province_of_onset
```

```
0 1 1/12/2020 NaN หญิง 61.0 China กรุงเทพมหานคร กรุงเทพฯ
```

```
2 3 1/22/2020 NaN หญิง 73.0 Thailand นครปฐม นครปฐม
```

```
3 4 1/22/2020 NaN ชาย 68.0 China กรุงเทพมหานคร กรุงเทพฯ
```

```
5 6 1/25/2020 NaN หญิง 33.0 China กรุงเทพมหานคร กรุงเทพฯ
```

```
In [ ]: eight_rows_covid['province_of_onset'] == 'กรุงเทพมหานคร' #สร้าง List ของ Logical expression
```

```
Out[ ]: 0    True  
1    True  
2    False  
3    True  
4    True  
5    True  
6    True  
7    False  
Name: province_of_onset, dtype: bool
```

```
In [ ]: eight_rows_covid[eight_rows_covid['province_of_onset'] == 'กรุงเทพมหานคร'] #เราเอา 2 แถวที่เป็น True
```

Out[]: No. announce_date notification_date sex age nationality province_of_isolation province_o

0 1 1/12/2020 NaN หญิง 61.0 China กรุงเทพมหานคร กรุงเทพ:

1 2 1/17/2020 NaN หญิง 74.0 China กรุงเทพมหานคร กรุงเทพ:

3 4 1/22/2020 NaN ชาย 68.0 China กรุงเทพมหานคร กรุงเทพ:

4 5 1/24/2020 NaN หญิง 66.0 China นนทบุรี กรุงเทพ:

5 6 1/25/2020 NaN หญิง 33.0 China กรุงเทพมหานคร กรุงเทพ:

6 7 1/26/2020 NaN หญิง 57.0 China กรุงเทพมหานคร กรุงเทพ:

HW (ให้ใช้เฉพาะที่อาจารย์สอนไปแล้วในวิชานี้)

1. คำนวณ อายุเฉลี่ย ของผู้หญิง และผู้ชาย ของข้อมูลทั้งหมด
2. คำนวณ อายุเฉลี่ย ของผู้หญิง และผู้ชาย ของผู้ป่วยในจังหวัดขอนแก่น

1. หาจำนวนผู้ป่วยที่เป็นคน "คนต่างด้าวเดินทางมาจากต่างประเทศ"

In []: this_data = data_covid[['sex', 'age', 'province_of_onset']]

In []: this_data

```
Out[ ]:      sex  age  province_of_onset
```

0	หญิง	61.0	กรุงเทพมหานคร
1	หญิง	74.0	กรุงเทพมหานคร
2	หญิง	73.0	นครปฐม
3	ชาย	68.0	กรุงเทพมหานคร
4	หญิง	66.0	กรุงเทพมหานคร
...
12648	หญิง	44.0	ชลบุรี
12649	หญิง	52.0	ระยอง
12650	หญิง	23.0	ระยอง
12651	หญิง	29.0	ระยอง
12652	หญิง	22.0	ตาก

12653 rows × 3 columns

```
In [ ]: female = this_data[this_data['sex']=='หญิง']
female['age']
```

```
Out[ ]: 0      61.0
1      74.0
2      73.0
4      66.0
5      33.0
...
12648    44.0
12649    52.0
12650    23.0
12651    29.0
12652    22.0
Name: age, Length: 5332, dtype: float64
```

```
In [ ]: sum = 0
N = 0
for a in female['age']:

    if a > 0:
        sum += a # sum = sum + a
        N += 1

print(f'อายุเฉลี่ย ของ ผู้ป่วยหญิง {sum/N}')
```

อายุเฉลี่ย ของ ผู้ป่วยหญิง 36.63955450022614

```
In [ ]: female_KK = female[female['province_of_onset']=='ขอนแก่น']

sum = 0
N = 0
for a in female_KK['age']:

    if a > 0:
        sum += a # sum = sum + a
        N += 1

print(f'อายุเฉลี่ย ของ ผู้ป่วยหญิง ในจังหวัดขอนแก่น {sum/N}')
```

บทที่ 5
(การเตรียมข้อมูลสำหรับการแสดงผล)

บทที่ 5: การเตรียมข้อมูลสำหรับการแสดงผล

1. การเตรียมข้อมูลด้วยภาษา Python
 - การแปลงข้อมูล: การแปลงข้อมูลเข้าสู่รูปแบบที่เหมาะสมสำหรับการวิเคราะห์
 - การเข้ารหัสข้อมูล: การเข้ารหัสข้อมูลหรือการแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม
 - การทำความสะอาดข้อมูล: การลบข้อมูลซ้ำซ้อน (Duplication) การลบค่าผิดปกติ (Outlier) การจัดการค่าว่าง (Missing Value) การจัดการข้อมูลที่มีความขัดแย้งกัน (Inconsistency)
2. การจัดการ Missing Value.....1
 - การตรวจสอบ Missing Value: วิธีการตรวจสอบข้อมูลที่ขาดหายไปในตาราง
 - วิธีการจัดการ Missing Value: การแก้ไขหรือลบข้อมูลที่ขาดหายไปในตาราง เช่น การเติมค่าพื้นฐาน การใช้ค่าเฉลี่ย หรือการลบแถวหรือคอลัมน์ที่มีข้อมูลที่ขาดหายไป
 - การใช้งาน .dropna(): เพื่อลบ Missing Value และเรียนรู้เกี่ยวกับ Parameter: inplace, subset ของ .dropna().....1
 - การใช้งาน .fillna(): เพื่อแทนค่า Missing Value ด้วยค่าที่กำหนดเอง.....2
3. การวนลูปอ่านข้อมูลในตารางแบบธรรมดาและแบบใช้ตัวเลขในการชี้ (.iloc[]).....5
 - การใช้วนลูป: การใช้ for loop เพื่อวนลูปอ่านข้อมูลในตารางแบบธรรมดา
 - การใช้ .iterrow(): เป็นคำสั่งที่ช่วยในการวนลูปอ่านข้อมูลในตาราง
 - การใช้ .iloc[]: การใช้ .iloc[] เพื่อวนลูปอ่านข้อมูลแบบมองตาราง (Pandas DataFrame) เป็น Numpy Array หรือ matrix (.iloc)
4. พัฒนาตัวช่วยใน Pandas (describe, mean, isnull).....6
 - describe(): การใช้ describe() เพื่อดูภาพรวมของข้อมูลที่นำเข้า ไม่นำข้อมูลที่เป็น Missing หรือ none มาคำนวณ
 - mean(): การใช้ mean() เพื่อคำนวณค่าเฉลี่ยของข้อมูลโดยไม่สนใจ Missing Value
 - isnull(): การใช้ isnull() เพื่อตรวจสอบค่าที่ขาดหายไปในข้อมูล (Missing Value).....7
 - .isnull().any(): ใช้สำหรับสรุปค่า True False ในแต่ละคอลัมน์ โดยเอาค่าความจริงภายในแต่ละคอลัมน์มา OR กัน
 - .isnull().all(): ใช้สำหรับสรุปค่า True False ในแต่ละคอลัมน์ โดยเอาค่าความจริงภายในแต่ละคอลัมน์มา and กัน

- การซื้อค่าในตารางของ .isnull(): ใช้การซื้อแบบ .iloc Numpy Array	
5. การเขื่อมข้อมูล Pandas DataFrame (การต่อตาราง).....	10
● การเพิ่มแถวและคอลัมน์: วิธีการเพิ่มแถวและคอลัมน์ใหม่ในตาราง	
● การเขื่อมข้อมูล: การใช้เมท็อดต่าง ๆ เพื่อเขื่อมข้อมูลจากตารางหรือ DataFrame แต่กันและกัน	
● การต่อตาราง: วิธีการต่อตารางแนวแกนแนวอน และแนวตั้ง.....	10
- การต่อตารางแกน Y หรือการเพิ่มแถว ด้วยคำสั่ง pd.concat().....	10
- การต่อตารางแกน X หรือการเพิ่มคอลัมน์ ด้วยคำสั่ง .merge() จับ 2 ตารางมาต่อกัน parameter: left_on, right_on ของ .merge().....	14
- การต่อตารางแกน X หรือการเพิ่มคอลัมน์ ด้วยคำสั่ง .map() เลือกมาเพิ่มเฉพาะบางคอลัมน์.....	18
6. การเรียงข้อมูล.....	16
● การเรียงข้อมูล: วิธีการเรียงลำดับข้อมูลใน DataFrame ตามค่าข้อมูลในคอลัมน์หรือหลายคอลัมน์ ด้วยคำสั่ง .sort_values()	
7. การสร้างคอลัมน์ใหม่ด้วย Pandas ในภาษา Python.....	19
● การสร้างคอลัมน์ใหม่: วิธีการสร้างคอลัมน์ใหม่โดยใช้ข้อมูลจากคอลัมน์ที่มีอยู่ใน DataFrame	
8. หลักการ Groupby และการ Groupby.....	21
● หลักการ Groupby: การใช้ groupby() เพื่อแบ่งข้อมูลเป็นกลุ่มตามเงื่อนไขที่กำหนด	
● การ Groupby: การสร้างและการใช้งาน groupby() เพื่อการวิเคราะห์ข้อมูลในกลุ่ม	
9. การสร้างตาราง Pandas.....	26
● การสร้างตาราง: วิธีการสร้างตาราง Pandas DataFrame	
- ยกตัวอย่างการสร้างตาราง Pandas แบบ Dictionary โดยใช้คำสั่ง pd.DataFrame()	
- ยกตัวอย่างการสร้างตาราง Pandas แบบ List โดยใช้คำสั่ง pd.DataFrame.from_records()	
10. Built-in Visualization Functions (parallel_coordinates, scatter_matrix).....	27
● parallel_coordinates: การสร้าง parallel_coordinates เพื่อแสดงการกระจายของข้อมูลในตารางในมิติหลาย ๆ พร้อมกัน ด้วยคำสั่ง pd.plotting.parallel_coordinates()	

- scatter_matrix: การสร้าง scatter_matrix เพื่อแสดงการกระจายของข้อมูลแบบ scatter plot ระหว่างคู่คอลัมน์ทั้งหมดในตาราง ด้วยคำสั่ง pd.plotting.scatter_matrix()

11. การบันทึกไฟล์ csv และการดาวน์โหลดใน Google Colab.....29

- การบันทึกไฟล์ csv: ขั้นตอนการบันทึก DataFrame เป็นไฟล์ csv
- การดาวน์โหลดใน Google Colab: วิธีการดาวน์โหลดไฟล์ csv จาก Google Colab

อายุเฉลี่ย ของ ผู้ป่วยหญิง ในจังหวัดขอนแก่น 41.57142857142857

การจัดการ Missing Value

- ลบ record ที่เป็น missing
- แทนที่ ค่า missing ด้วยค่าที่เหมาะสม mean, default, category-unknown
- ใช้ ค่าจาก columns อื่นๆ ช่วยประมาณค่า ค่าใน column ที่หายไป (regression, deep learning, etc.)

ลบ record (dropna)

missing = None NA (not authorized) NaN (not a number)

```
In [ ]: this_data.shape #(12423, 3)
```

```
Out[ ]: (12653, 3)
```

```
In [ ]: print(data_covid.shape)
print(data_covid.dropna().shape)
```

```
(12653, 10)
```

```
(6655, 10)
```

```
In [ ]: this_data.dropna().shape
```

```
Out[ ]: (8478, 3)
```

```
In [ ]: this_data.dropna()
```

```
Out[ ]:
```

	sex	age	province_of_onset
0	หญิง	61.0	กรุงเทพมหานคร
1	หญิง	74.0	กรุงเทพมหานคร
2	หญิง	73.0	นครปฐม
3	ชาย	68.0	กรุงเทพมหานคร
4	หญิง	66.0	กรุงเทพมหานคร
...
12648	หญิง	44.0	ชลบุรี
12649	หญิง	52.0	ระยอง
12650	หญิง	23.0	ระยอง
12651	หญิง	29.0	ระยอง
12652	หญิง	22.0	ตาก

8478 rows × 3 columns

```
In [ ]: this_data.dropna()
print(this_data.shape)
```

```
(12653, 3)
```

```
In [ ]: this_data.dropna(inplace=True)
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
    """Entry point for launching an IPython kernel.
```

```
In [ ]: this_data.dropna(inplace=True)  
print(this_data.shape)
```

```
(8478, 3)
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
    """Entry point for launching an IPython kernel.
```

```
In [ ]: this_data = data_covid[['sex', 'age', 'province_of_onset']]  
this_data.shape
```

```
Out[ ]: (12653, 3)
```

```
In [ ]: this_data.dropna().shape
```

```
Out[ ]: (8478, 3)
```

```
In [ ]: this_data.dropna(subset=['age']).shape
```

```
Out[ ]: (8881, 3)
```

แทน missing ด้วยค่าที่เหมาะสม (fillna)

```
In [ ]: this_data = data_covid[['sex', 'age', 'province_of_onset']]
```

```
In [ ]: this_data.head()
```

```
Out[ ]:   sex  age  province_of_onset  
0  หญิง  61.0      กรุงเทพมหานคร  
1  หญิง  74.0      กรุงเทพมหานคร  
2  หญิง  73.0      นครปฐม  
3  ชาย  68.0      กรุงเทพมหานคร  
4  หญิง  66.0      กรุงเทพมหานคร
```

```
In [ ]: print(f'จำนวน record ก่อน drop missing ใน province {this_data.shape[0]}')  
print(f'จำนวน record หลัง drop missing ใน province {this_data.dropna(subset=["provir
```

```
จำนวน record ก่อน drop missing ใน province 12653
```

```
จำนวน record หลัง drop missing ใน province 10662
```

```
In [ ]: this_data_updated = this_data.fillna(value={'sex': 'ไม้รู้', 'age': -1, 'province_of_onset': 'ไม่ระบุ'})  
this_data_updated.head()
```

```
Out[ ]:    sex  age  province_of_onset
```

0	หญิง	61.0	กรุงเทพมหานคร
1	หญิง	74.0	กรุงเทพมหานคร
2	หญิง	73.0	นครปฐม
3	ชาย	68.0	กรุงเทพมหานคร
4	หญิง	66.0	กรุงเทพมหานคร

```
In [ ]: this_data_updated[this_data_updated['sex']=='เมร์']
```

```
Out[ ]:    sex  age  province_of_onset
```

4391	เมร์	-1.0	สมุทรสาคร
4392	เมร์	-1.0	สมุทรสาคร
4393	เมร์	-1.0	สมุทรสาคร
4394	เมร์	-1.0	สมุทรสาคร
4395	เมร์	-1.0	สมุทรสาคร
...
12635	เมร์	-1.0	สมุทรสาคร
12636	เมร์	-1.0	สมุทรสาคร
12637	เมร์	-1.0	สมุทรสาคร
12638	เมร์	-1.0	สมุทรสาคร
12639	เมร์	-1.0	สมุทรสาคร

2502 rows × 3 columns

```
In [ ]: this_data_updated['province_of_onset']=='เมร์'
```

```
Out[ ]: 0      False
1      False
2      False
3      False
4      False
...
12648  False
12649  False
12650  False
12651  False
12652  False
Name: province_of_onset, Length: 12653, dtype: bool
```

```
In [ ]: data_covid[this_data_updated['province_of_onset']=='เมร์']
```

Out[]:

No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	pro
414	415	3/22/2020	3/22/2020	ชาย	29.0	Thailand	กรุงเทพมหานคร
529	530	3/22/2020	3/20/2020	ชาย	37.0	Thailand	กรุงเทพมหานคร
555	556	3/22/2020	3/21/2020	หญิง	6.0	Thailand	กรุงเทพมหานคร
576	577	3/22/2020	3/20/2020	ชาย	Nan	Thailand	กรุงเทพมหานคร
588	589	3/22/2020	3/21/2020	ชาย	Nan	Thailand	กรุงเทพมหานคร
...
12405	12406	1/18/2021	1/17/2021	Nan	Nan	Nan	สมุทรสาคร
12406	12407	1/18/2021	1/17/2021	Nan	Nan	Nan	สมุทรสาคร
12407	12408	1/18/2021	1/17/2021	Nan	Nan	Nan	สมุทรสาคร
12408	12409	1/18/2021	1/17/2021	Nan	Nan	Nan	สมุทรสาคร
12409	12410	1/18/2021	1/17/2021	Nan	Nan	Nan	สมุทรสาคร

1991 rows × 10 columns

In []: `this_data_updated_2 = this_data.fillna(value={'province_of_onset': 'ไมร์'})`In []: `this_data_updated_2[this_data_updated['sex']=='ไมร์']`

```
Out[ ]:      sex  age  province_of_onset
 4391  NaN  NaN        สุนทรสาคร
 4392  NaN  NaN        สุนทรสาคร
 4393  NaN  NaN        สุนทรสาคร
 4394  NaN  NaN        สุนทรสาคร
 4395  NaN  NaN        สุนทรสาคร
 ...
 12635  NaN  NaN        สุนทรสาคร
 12636  NaN  NaN        สุนทรสาคร
 12637  NaN  NaN        สุนทรสาคร
 12638  NaN  NaN        สุนทรสาคร
 12639  NaN  NaN        สุนทรสาคร
```

2502 rows × 3 columns

การวนลูป record ในตาราง (.iterrows)

```
In [ ]: this_data = data_covid[['sex', 'age', 'province_of_onset']]
```

```
In [ ]: for each_row in this_data.iterrows():
    # print(each_row)
    # print(each_row[1])
    if (each_row[1]['age'] == 20) and (each_row[1]['province_of_onset'] == 'ขอนแก่น'
        print(each_row)
```

(9334, sex ชาย
 age 20
 province_of_onset ขอนแก่น
 Name: 9334, dtype: object)

การวนลูป แบบมองตารางแพนด้าส์(pandas dataframe) เป็น numpy array (.iloc)

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

9334
 sex ชาย
 age 20
 province_of_onset ขอนแก่น
 Name: 9334, dtype: object

Quiz ตัดตารางออกมาเฉพาะปี 2021 announce_date ในปี 2021

Hint

- วนลูปหา index ของปี 2021
- ตัดตารางมาเฉพาะ ปี 2021

```
In [ ]: 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()
```

Out[]:

No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	provin
6884	6885	1/1/2021	12/31/2020	หญิง	40.0	Thailand	กรุงเทพมหานคร
6885	6886	1/1/2021	12/31/2020	หญิง	21.0	Thailand	ปทุมธานี
6886	6887	1/1/2021	12/31/2020	หญิง	20.0	Thailand	นครปฐม
6887	6888	1/1/2021	12/31/2020	หญิง	47.0	Thailand	สมุทรสาคร
6888	6889	1/1/2021	12/31/2020	หญิง	36.0	Cambodia	ปทุมธานี กรุง

Function ตัวช่วยใน pandas

`.describe()` คำนวณค่าทางสถิติของข้อมูลที่เป็นตัวเลข

```
In [ ]: data_covid.describe()
```

Out[]:

	No.	age
count	12653.000000	8881.000000
mean	6327.000000	38.054748
std	3652.750813	15.125178
min	1.000000	0.100000
25%	3164.000000	27.000000
50%	6327.000000	36.000000
75%	9490.000000	48.000000
max	12653.000000	97.000000

`.mean()` คำนวณค่าเฉลี่ยของข้อมูลโดยไม่สนใจ missing

```
In [ ]: data_covid[data_covid['sex']=='ชาย']['age'].mean()
```

```
Out[ ]: 39.46076940331987
```

.isnull()

```
In [ ]: data_covid.isnull()
```

```
Out[ ]:   No. announce_date notification_date sex age nationality province_of_isolation pro
          0 False False True False False False False
          1 False False True False False False False
          2 False False True False False False False
          3 False False True False False False False
          4 False False True False False False False
          ...
12648 False False False False False False False False
12649 False False False False False False False False
12650 False False False False False False False False
12651 False False False False False False False False
12652 False False False False False False False False
```

12653 rows × 10 columns

next

- .describe
- .mean
- HW ວັນລຸ່ມ missing .isnull
- ຕ່ອຕາຮາງແກນ X ແກນ y
- .groupby
- save table

```
In [ ]:
```

```
In [ ]: import pandas as pd
```

```
In [ ]: from google.colab import drive  
drive.mount('/content/drive')
```

Mounted at /content/drive

```
In [ ]: import os
```

```
path = '/content/drive/My Drive/dataviz_2021_data'  
covid_file_path = os.path.join(path, 'pm-20-jan-2021.csv')  
print(covid_file_path)
```

/content/drive/My Drive/dataviz_2021_data/pm-20-jan-2021.csv

```
In [ ]: data_covid = pd.read_csv(covid_file_path)  
data_covid.head()
```

```
Out[ ]: No. announce_date notification_date sex age nationality province_of_isolation province_o
```

0 1 1/12/2020 NaN หญิง 61.0 China กรุงเทพมหานคร กรุงเทพฯ

1 2 1/17/2020 NaN หญิง 74.0 China กรุงเทพมหานคร กรุงเทพฯ

2 3 1/22/2020 NaN หญิง 73.0 Thailand นครปฐม กรุงเทพฯ

3 4 1/22/2020 NaN ชาย 68.0 China กรุงเทพมหานคร กรุงเทพฯ

4 5 1/24/2020 NaN หญิง 66.0 China นนทบุรี กรุงเทพฯ

.isnull()?? None??

In []: `data_covid.isnull()`

Out[]:

	No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	prov
0	False	False	True	False	False	False	False	False
1	False	False	True	False	False	False	False	False
2	False	False	True	False	False	False	False	False
3	False	False	True	False	False	False	False	False
4	False	False	True	False	False	False	False	False
...
12648	False	False	False	False	False	False	False	False
12649	False	False	False	False	False	False	False	False
12650	False	False	False	False	False	False	False	False
12651	False	False	False	False	False	False	False	False
12652	False	False	False	False	False	False	False	False

12653 rows × 10 columns

.any() เอาค่าความจริงภายในแต่ละ column มา OR กัน

In []: `data_covid.isnull().any()`

Out[]:

No.	False
announce_date	False
notification_date	True
sex	True
age	True
nationality	True
province_of_isolation	True
province_of_onset	True
district_of_onset	True
risk	True
dtype: bool	

.all() เอาค่าความจริงภายในแต่ละ column มา AND กัน

In []: `data_covid.isnull().all()`

Out[]:

No.	False
announce_date	False
notification_date	False
sex	False
age	False
nationality	False
province_of_isolation	False
province_of_onset	False
district_of_onset	False
risk	False
dtype: bool	

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

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-8-655f7c695cf3> in <module>()
----> 1 data_covid.iloc[0,0].isnull()

AttributeError: 'numpy.int64' object has no attribute 'isnull'
```

```
In [ ]: data_covid['No.'][0].isnull()
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-9-85eef827ab2a> in <module>()
----> 1 data_covid['No.'][0].isnull()

AttributeError: 'numpy.int64' object has no attribute 'isnull'
```

```
In [ ]: data_covid.iloc[0,0]
```

```
Out[ ]: 1
```

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

```
Out[ ]: 0    False
Name: No., dtype: bool
```

ต่อตารางแกน X แกน y

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

ต่อแกน Y pd.concat()

```
In [ ]: data_covid['province_of_onset']=='ขอนแก่น'
```

```
Out[ ]: 0      False
        1      False
        2      False
        3      False
        4      False
        ...
       12648   False
       12649   False
       12650   False
       12651   False
       12652   False
Name: province_of_onset, Length: 12653, dtype: bool
```

```
In [ ]: dataKK = data_covid[data_covid['province_of_onset']=='ขอนแก่น']
dataKK.head()
```

Out[]:

No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	provin
180	181	3/18/2020	3/15/2020	ชาย	33.0	Thailand	ขอนแก่น
462	463	3/22/2020	3/21/2020	หญิง	36.0	Thailand	ขอนแก่น
1466	1467	3/30/2020	3/26/2020	ชาย	19.0	Thailand	ขอนแก่น
1970	1971	4/3/2020	3/31/2020	หญิง	70.0	Thailand	ขอนแก่น
2637	2638	4/15/2020	4/14/2020	หญิง	63.0	Thailand	ขอนแก่น

In []: `dataUD = data_covid[data_covid['province_of_onset']=='อุดรธานี']
dataUD.head()`

```
Out[ ]: No. announce_date notification_date sex age nationality province_of_isolation province
```

424	425	3/22/2020	3/21/2020	ชาย	33.0	Thailand	อุดรธานี
-----	-----	-----------	-----------	-----	------	----------	----------

434	435	3/22/2020	3/20/2020	หญิง	47.0	Thailand	อุดรธานี
-----	-----	-----------	-----------	------	------	----------	----------

471	472	3/22/2020	3/22/2020	หญิง	26.0	Thailand	อุดรธานี
-----	-----	-----------	-----------	------	------	----------	----------

883	884	3/25/2020	3/24/2020	ชาย	25.0	Thailand	อุดรธานี
-----	-----	-----------	-----------	-----	------	----------	----------

885	886	3/25/2020	3/24/2020	หญิง	20.0	Thailand	อุดรธานี
-----	-----	-----------	-----------	------	------	----------	----------

```
◀ ▶
```

```
In [ ]: dataMS = data_covid[data_covid['province_of_onset']=='มหาสารคาม']
dataMS.head()
```

```
Out[ ]: No. announce_date notification_date sex age nationality province_of_isolation province
```

346	347	3/21/2020	3/20/2020	ชาย	34.0	Thailand	นนทบุรี
-----	-----	-----------	-----------	-----	------	----------	---------

789	790	3/24/2020	3/28/2020	ชาย	48.0	Thailand	มหาสารคาม
-----	-----	-----------	-----------	-----	------	----------	-----------

6690	6691	12/31/2020	12/30/2020	หญิง	42.0	Thailand	มหาสารคาม
------	------	------------	------------	------	------	----------	-----------

10802	10803	1/12/2021	1/11/2021	หญิง	25.0	Thailand	สมุทรสาคร
-------	-------	-----------	-----------	------	------	----------	-----------

```
◀ ▶
```

```
In [ ]: dataMYisan = pd.concat([dataKK,dataUD,dataMS])
dataMYisan
```

Out[]:

No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	pro
180	181	3/18/2020	3/15/2020	ชาย	33.0	Thailand	ขอนแก่น
462	463	3/22/2020	3/21/2020	หญิง	36.0	Thailand	ขอนแก่น
1466	1467	3/30/2020	3/26/2020	ชาย	19.0	Thailand	ขอนแก่น
1970	1971	4/3/2020	3/31/2020	หญิง	70.0	Thailand	ขอนแก่น
2637	2638	4/15/2020	4/14/2020	หญิง	63.0	Thailand	ขอนแก่น
2673	2674	4/17/2020	4/16/2020	ชาย	68.0	Thailand	ขอนแก่น
5948	5949	12/26/2020	12/25/2020	หญิง	32.0	Thailand	ขอนแก่น
6082	6083	12/27/2020	12/26/2020	หญิง	36.0	Thailand	ขอนแก่น
9333	9334	1/7/2021	1/6/2021	ชาย	17.0	Thailand	ขอนแก่น
9334	9335	1/7/2021	1/6/2021	ชาย	20.0	Thailand	ขอนแก่น
10610	10611	1/12/2021	1/11/2021	ชาย	17.0	Thailand	ขอนแก่น
11517	11518	1/16/2021	1/15/2021	ชาย	12.0	Thailand	ขอนแก่น
11697	11698	1/17/2021	1/16/2021	หญิง	17.0	Thailand	ขอนแก่น
11698	11699	1/17/2021	1/16/2021	หญิง	37.0	Thailand	ขอนแก่น

No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	pro
424	425	3/22/2020	3/21/2020	ชาย	33.0	Thailand	อุดรธานี
434	435	3/22/2020	3/20/2020	หญิง	47.0	Thailand	อุดรธานี
471	472	3/22/2020	3/22/2020	หญิง	26.0	Thailand	อุดรธานี
883	884	3/25/2020	3/24/2020	ชาย	25.0	Thailand	อุดรธานี
885	886	3/25/2020	3/24/2020	หญิง	20.0	Thailand	อุดรธานี
941	942	3/26/2020	3/25/2020	ชาย	31.0	Thailand	อุดรธานี
1059	1060	3/27/2020	3/26/2020	ชาย	54.0	United States of America	อุดรธานี
2013	2014	4/4/2020	4/3/2020	หญิง	49.0	Thailand	อุดรธานี
5810	5811	12/24/2020	12/23/2020	ชาย	30.0	Thailand	อุดรธานี
7433	7434	1/3/2021	Nan	หญิง	66.0	Thailand	ระยอง
346	347	3/21/2020	3/20/2020	ชาย	34.0	Thailand	นนทบุรี
789	790	3/24/2020	3/28/2020	ชาย	48.0	Thailand	มหาสารคาม
6690	6691	12/31/2020	12/30/2020	หญิง	42.0	Thailand	มหาสารคาม
10802	10803	1/12/2021	1/11/2021	หญิง	25.0	Thailand	สมุทรสาคร

ต่อแก่น X

- จับ 2 ตารางมาต่อกันเลย (merge)
- เลือกมาเพิ่มเฉพาะบาง column (map)

```
In [ ]: data_province = data_covid[['No.', 'announce_date', 'province_of_onset']]  
data_province
```

Out[]: No. announce_date province_of_onset

0	1	1/12/2020	กรุงเทพมหานคร
1	2	1/17/2020	กรุงเทพมหานคร
2	3	1/22/2020	นครปฐม
3	4	1/22/2020	กรุงเทพมหานคร
4	5	1/24/2020	กรุงเทพมหานคร
...
12648	12649	1/20/2021	ชลบุรี
12649	12650	1/20/2021	ระยอง
12650	12651	1/20/2021	ระยอง
12651	12652	1/20/2021	ระยอง
12652	12653	1/20/2021	ตาก

12653 rows × 3 columns

```
In [ ]: data_human = data_covid[['No.', 'age', 'sex', 'nationality']]  
data_human
```

Out[]: No. age sex nationality

0	1	61.0	หญิง	China
1	2	74.0	หญิง	China
2	3	73.0	หญิง	Thailand
3	4	68.0	ชาย	China
4	5	66.0	หญิง	China
...
12648	12649	44.0	หญิง	Thailand
12649	12650	52.0	หญิง	Thailand
12650	12651	23.0	หญิง	Thailand
12651	12652	29.0	หญิง	Thailand
12652	12653	22.0	หญิง	Burma

12653 rows × 4 columns

แบบง่าย รู้ว่าสองตาราง record ตรงกัน

```
In [ ]: full_table1 = data_human.merge(data_province)  
full_table1.head()
```

Out[]:

	No.	age	sex	nationality	announce_date	province_of_onset
0	1	61.0	หญิง	China	1/12/2020	กรุงเทพมหานคร
1	2	74.0	หญิง	China	1/17/2020	กรุงเทพมหานคร
2	3	73.0	หญิง	Thailand	1/22/2020	นครปฐม
3	4	68.0	ชาย	China	1/22/2020	กรุงเทพมหานคร
4	5	66.0	หญิง	China	1/24/2020	กรุงเทพมหานคร

SORT

In []: `data_human2 = data_human.sort_values('age')`
`data_human2`

Out[]:

	No.	age	sex	nationality
1987	1988	0.10	ชาย	Thailand
11497	11498	0.11	ชาย	Burma
6477	6478	0.11	หญิง	Thailand
1675	1676	0.30	ชาย	Japan
1075	1076	0.40	ชาย	Thailand
...
12635	12636	NaN	NaN	Thailand
12636	12637	NaN	NaN	Thailand
12637	12638	NaN	NaN	Thailand
12638	12639	NaN	NaN	Thailand
12639	12640	NaN	NaN	Thailand

12653 rows × 4 columns

In []: `full_table2 = data_human2.merge(data_province)`
`full_table2.head()`

Out[]:

	No.	age	sex	nationality	announce_date	province_of_onset
0	1988	0.10	ชาย	Thailand	4/4/2020	ระยอง
1	11498	0.11	ชาย	Burma	1/16/2021	สมุทรสาคร
2	6478	0.11	หญิง	Thailand	12/30/2020	เพชรบุรี
3	1676	0.30	ชาย	Japan	4/1/2020	กรุงเทพมหานคร
4	1076	0.40	ชาย	Thailand	3/27/2020	สุราษฎร์ธานี

In []: `data_human2_renamed = data_human2.rename(columns={'No.': 'patientNumber'})`
`data_human2_renamed.head()`

Out[]:

	patientNumber	age	sex	nationality
1987	1988	0.10	ໝາຍ	Thailand
11497	11498	0.11	ໝາຍ	Burma
6477	6478	0.11	ໜີ້ງ	Thailand
1675	1676	0.30	ໝາຍ	Japan
1075	1076	0.40	ໝາຍ	Thailand

In []: `data_human2_renamed.merge(data_province)`

```
MergeError                                     Traceback (most recent call last)
<ipython-input-25-ee0d027e7ef8> in <module>()
----> 1 data_human2_renamed.merge(data_province)

/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py in merge(self, right, how, on, left_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
    7961             copy=copy,
    7962             indicator=indicator,
-> 7963             validate=validate,
    7964         )
    7965

/usr/local/lib/python3.6/dist-packages/pandas/core/reshape/merge.py in merge(left, right, how, on, left_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
    85             copy=copy,
    86             indicator=indicator,
-> 87             validate=validate,
    88         )
    89     return op.get_result()

/usr/local/lib/python3.6/dist-packages/pandas/core/reshape/merge.py in __init__(self, left, right, how, on, left_on, right_on, axis, left_index, right_index, sort, suffixes, copy, indicator, validate)
    643             warnings.warn(msg, UserWarning)
    644
--> 645         self._validate_specification()
    646
    647         # note this function has side effects

/usr/local/lib/python3.6/dist-packages/pandas/core/reshape/merge.py in _validate_specification(self)
    1215             if len(common_cols) == 0:
    1216                 raise MergeError(
-> 1217                     "No common columns to perform merge on. "
    1218                     f"Merge options: left_on={self.left_on}, "
    1219                     f"right_on={self.right_on}, "

MergeError: No common columns to perform merge on. Merge options: left_on=None, right_on=None, left_index=False, right_index=False
```

In []: `full_table3 = data_human2_renamed.merge(data_province, left_on='patientNumber', right`

Out[]:

	patientNumber	age	sex	nationality	No.	announce_date	province_of_onset
0	1988	0.10	ชาย	Thailand	1988	4/4/2020	ระยอง
1	11498	0.11	ชาย	Burma	11498	1/16/2021	สมุทรสาคร
2	6478	0.11	หญิง	Thailand	6478	12/30/2020	เพชรบุรี
3	1676	0.30	ชาย	Japan	1676	4/1/2020	กรุงเทพมหานคร
4	1076	0.40	ชาย	Thailand	1076	3/27/2020	สุราษฎร์ธานี

map() เลือกมาเฉพาะบาง column มาแปะเพิ่มเข้าไป

In []: `data_human2_renamed.head()`

Out[]:

	patientNumber	age	sex	nationality
	1987	1988	0.10	ชาย
	11497	11498	0.11	ชาย
	6477	6478	0.11	หญิง
	1675	1676	0.30	ชาย
	1075	1076	0.40	ชาย

In []: `data_covid.head()`

```
Out[ ]: No. announce_date notification_date sex age nationality province_of_isolation province_o
```

	No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	province_o
0	1	1/12/2020		NaN หญิง	61.0	China	กรุงเทพมหานคร	กรุงเทพฯ
1	2	1/17/2020		NaN หญิง	74.0	China	กรุงเทพมหานคร	กรุงเทพฯ
2	3	1/22/2020		NaN หญิง	73.0	Thailand	นครปฐม	
3	4	1/22/2020		NaN ชาย	68.0	China	กรุงเทพมหานคร	กรุงเทพฯ
4	5	1/24/2020		NaN หญิง	66.0	China	นนทบุรี	กรุงเทพฯ

คุณสมบัติของ pandas ก็คือเราสามารถสร้าง column ใหม่ให้ตาราง df ได้ โดย

```
df['ชื่อ column ใหม่'] = (list ที่มีจำนวนสมาชิกเท่ากับจำนวน record ของ df)
```

```
In [ ]: data_human2_renamed.head()
```

	patientNumber	age	sex	nationality	
	1987	1988	0.10	ชาย	Thailand
	11497	11498	0.11	ชาย	Burma
	6477	6478	0.11	หญิง	Thailand
	1675	1676	0.30	ชาย	Japan
	1075	1076	0.40	ชาย	Thailand

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

Out[]:

	patientNumber	age	sex	nationality	num
1987	1988	0.10	ชาย	Thailand	0
11497	11498	0.11	ชาย	Burma	1
6477	6478	0.11	หญิง	Thailand	2
1675	1676	0.30	ชาย	Japan	3
1075	1076	0.40	ชาย	Thailand	4
...
12635	12636	NaN	NaN	Thailand	12648
12636	12637	NaN	NaN	Thailand	12649
12637	12638	NaN	NaN	Thailand	12650
12638	12639	NaN	NaN	Thailand	12651
12639	12640	NaN	NaN	Thailand	12652

12653 rows × 5 columns

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

Out[]:

1987	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้	Cluster	สมุทรสาคร
11497	...	Cluster	สมุทรสาคร
6477	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้	Cluster	สมุทรสาคร
1675	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้	Cluster	สมุทรสาคร
1075	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้	Cluster	สมุทรสาคร
...	...	Cluster	สมุทรสาคร
12635	...	Cluster	สมุทรสาคร
12636	...	Cluster	สมุทรสาคร
12637	...	Cluster	สมุทรสาคร
12638	...	Cluster	สมุทรสาคร
12639	...	Cluster	สมุทรสาคร

Name: patientNumber, Length: 12653, dtype: object

In []: `data_human2_renamed['detail'] = data_human2_renamed['patientNumber'].map(data_covid
data_human2_renamed`

Out[]:

	patientNumber	age	sex	nationality	num	detail
1987	1988	0.10	ชาย	Thailand	0	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
11497	11498	0.11	ชาย	Burma	1	Cluster สมุทรสาคร
6477	6478	0.11	หญิง	Thailand	2	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
1675	1676	0.30	ชาย	Japan	3	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
1075	1076	0.40	ชาย	Thailand	4	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
...
12635	12636	NaN	NaN	Thailand	12648	Cluster สมุทรสาคร
12636	12637	NaN	NaN	Thailand	12649	Cluster สมุทรสาคร
12637	12638	NaN	NaN	Thailand	12650	Cluster สมุทรสาคร
12638	12639	NaN	NaN	Thailand	12651	Cluster สมุทรสาคร
12639	12640	NaN	NaN	Thailand	12652	Cluster สมุทรสาคร

12653 rows × 6 columns

HW7 สร้างตารางใหม่ ที่ค่าใน sex เป็น missing ทั้งหมด

- สรุปว่าทำไน record นั้นๆ ถึงเป็น missing
- .groupby
- create pandas table
- simple visualization
- save table

.groupby()

<https://www.kaggle.com/crawford/python-groupby-tutorial#>

In []: `data_covid.groupby('nationality') ## จัดกลุ่มค่าที่เหมือนกันไว้ด้วยกัน`

Out[]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x7f0e67c0d9b0>

In []: `data_covid.groupby('nationality').count()`

Out[]:

No. announce_date notification_date sex age province_of_isolation province_of_o

nationality

Afghanistan	1	1	1	1	1	1	1
Albania	3	3	3	3	3	3	3
American	1	1	1	1	1	1	1
American	1	1	1	1	1	1	1
American Samoa	1	1	1	1	1	1	1
...
deutsch	1	1	1	1	1	1	1
thailand	2	2	2	2	2	2	2
ต่างด้าว	21	21	21	21	21	21	21
ไทยในญี่	1	1	1	1	1	1	1
ไฟในญี่	1	1	1	1	1	1	1

100 rows × 9 columns

◀ ▶

In []:

data_covid.groupby('nationality').mean()

Out[]:

No. age

nationality

Afghanistan	4346.000000	31.000000
Albania	2377.333333	40.666667
American	3862.000000	39.000000
American	3905.000000	46.000000
American Samoa	11687.000000	21.000000
...
deutsch	3898.000000	55.000000
thailand	6823.000000	53.500000
ต่างด้าว	2944.952381	20.000000
ไทยในญี่	10379.000000	30.000000
ไฟในญี่	5983.000000	33.000000

100 rows × 2 columns

In []:

data_covid.groupby('nationality').max(' ') ### ทำไม่ได้ !!! และรันได้ !

Out[]:

	No.	age
	nationality	
Afghanistan	4346	31.0
Albania	3522	51.0
American	3862	39.0
American	3905	46.0
American Samoa	11687	21.0
...
deutsch	3898	55.0
thailand	6824	57.0
ต่างด้าว	2998	43.0
ไทยในญี่ปุ่น	10379	30.0
ไทยในญี่ปุ่น	5983	33.0

100 rows × 2 columns

ทำ HW7 ด้วย groupby()

In []: `data_covid['sex'].isnull()`

Out[]:

```
0      False
1      False
2      False
3      False
4      False
...
12648    False
12649    False
12650    False
12651    False
12652    False
Name: sex, Length: 12653, dtype: bool
```

In []: `missing_sex = data_covid[data_covid['sex'].isnull()]`
`missing_sex`

Out[]:

	No.	announce_date	notification_date	sex	age	nationality	province_of_isolation	pro
4391	4392	12/20/2020		NaN	NaN	NaN	Burma	สมุทรสาคร
4392	4393	12/20/2020		NaN	NaN	NaN	Burma	สมุทรสาคร
4393	4394	12/20/2020		NaN	NaN	NaN	Burma	สมุทรสาคร
4394	4395	12/20/2020		NaN	NaN	NaN	Burma	สมุทรสาคร
4395	4396	12/20/2020		NaN	NaN	NaN	Burma	สมุทรสาคร
...
12635	12636	1/20/2021	1/19/2021	NaN	NaN	NaN	Thailand	สมุทรสาคร
12636	12637	1/20/2021	1/19/2021	NaN	NaN	NaN	Thailand	สมุทรสาคร
12637	12638	1/20/2021	1/19/2021	NaN	NaN	NaN	Thailand	สมุทรสาคร
12638	12639	1/20/2021	1/19/2021	NaN	NaN	NaN	Thailand	สมุทรสาคร
12639	12640	1/20/2021	1/19/2021	NaN	NaN	NaN	Thailand	สมุทรสาคร

2502 rows × 10 columns

In []: missing_sex.groupby('nationality').describe()

Out[]:

	No.									
	count	mean	std	min	25%	50%	75%	max	count	me
nationality										
Burma	1366.0	5124.517570	472.244857	4392.0	4733.25	5096.5	5467.75	7075.0	0.0	N
Cambodia	61.0	6861.278689	275.220828	6543.0	6567.00	7035.0	7130.00	7160.0	0.0	N
Thailand	51.0	7819.274510	1649.397527	6272.0	7321.50	7509.0	7540.50	12640.0	1.0	2

In []: missing_sex.groupby('province_of_onset').describe()

```
Out[ ]:
```

		count	mean	std	min	25%	50%	75%	max	No.
province_of_onset										
ชลบุรี		18.0	7506.055556	6.637436	7497.0	7501.25	7505.5	7509.75	7524.0	1.
ระยอง		6.0	6289.000000	49.010203	6267.0	6268.25	6269.5	6270.75	6389.0	0.
สมุทรสาคร		1358.0	5136.421944	639.463677	4392.0	4731.25	5092.5	5461.75	12640.0	0.

```
In [ ]: missing_sex.groupby(['province_of_onset','nationality']).describe() ## groupby หลัก
```

```
Out[ ]:
```

		count	mean	std	min	25%	50%	75%	max	No.
province_of_onset nationality										
ชลบุรี	Thailand	18.0	7506.055556	6.637436	7497.0	7501.25	7505.5	7509.75	7524.0	1.
ระยอง	Thailand	1.0	6389.000000	NaN	6389.0	6389.00	6389.0	6389.00	6389.0	0.
สมุทรสาคร	Thailand	8.0	10251.125000	3294.200724	6272.0	6273.75	12636.5	12638.2	12640.0	0.
	Burma	1349.0	5103.564122	436.284721	4392.0	4729.00	5088.0	5455.00	5455.0	0.

```
In [ ]: misssing_sex_no_burma = missing_sex[missing_sex['nationality']!='Burma']  
misssing_sex_no_burma.groupby('risk').describe()
```

```
Out[ ]:
```

		count	mean	std	min	25%	50%	75%	max	No.
risk										
Cluster ชลบุรี		18.0	7506.055556	6.637436	7497.0	7501.25	7505.5	7509.75	7524.0	1.
Cluster ระยอง		7.0	6363.285714	201.569437	6267.0	6268.50	6270.0	6330.00	6809.0	0.
Cluster สมุทรสาคร		895.0	11423.836872	1231.787200	6124.0	11176.50	11881.0	12191.50	12640.0	0.
อุบลราชธานี		72.0	6854.111111	269.187068	6543.0	6564.75	7027.0	7128.25	7160.0	0.

```
In [ ]: missing_sex.groupby('risk').describe()
```

Out[]:

										No.
	count	mean	std	min	25%	50%	75%	max	count	m
risk										
Cluster ชลบุรี	18.0	7506.055556	6.637436	7497.0	7501.25	7505.5	7509.75	7524.0	1.0	I
Cluster ระยอง	8.0	6452.250000	313.277946	6267.0	6268.75	6270.5	6494.00	7075.0	0.0	I
Cluster สมุทรสาคร	2258.0	7619.011515	3200.279792	4392.0	4978.25	5572.5	11763.75	12640.0	0.0	I
อยู่ระหว่าง การ สืบสวน	74.0	6852.945946	269.058169	6543.0	6564.25	7027.0	7127.75	7160.0	0.0	I

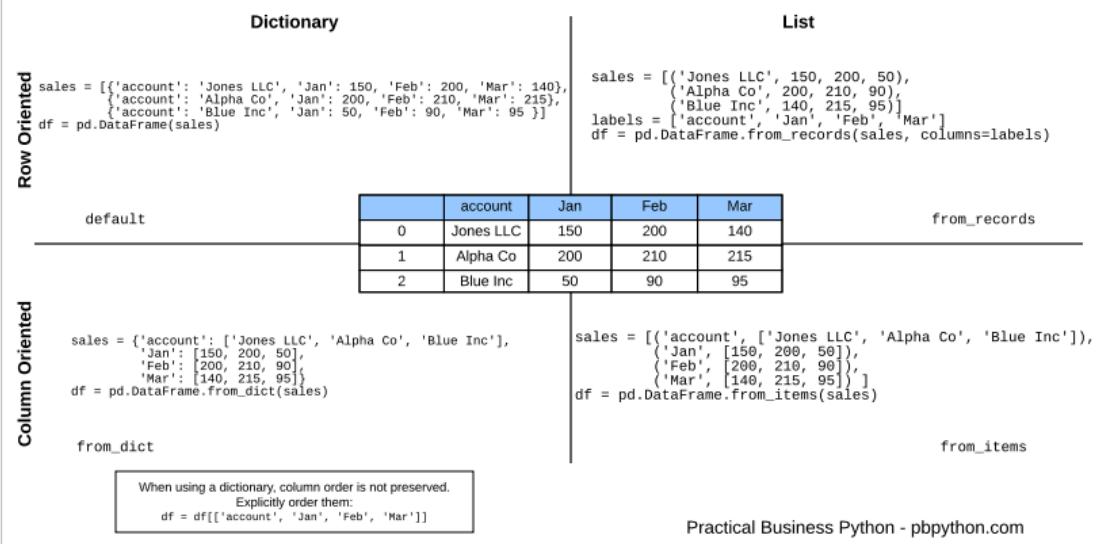
In []: missing_sex[missing_sex['risk']=='Cluster สมุทรสาคร'].groupby('announce_date').count

Out[]:

	No.	notification_date	sex	age	nationality	province_of_isolation	province_of_onset
	announce_date						
announce_date							
1/14/2021	172		172	0	0	0	172
1/17/2021	311		311	0	0	0	311
1/18/2021	269		269	0	0	0	269
1/20/2021	5		5	0	0	5	5
1/5/2021	4		4	0	0	0	4
1/7/2021	109		109	0	0	0	109
12/20/2020	516		0	0	0	516	516
12/21/2020	360		0	0	0	360	360
12/22/2020	397		0	0	0	397	397
12/25/2020	35		0	0	0	35	35
12/26/2020	12		0	0	0	12	12
12/27/2020	36		18	0	0	18	36
12/28/2020	14		0	0	0	14	14
12/30/2020	3		0	0	0	1	3
12/31/2020	15		0	0	0	15	15

create pandas table

Creating Pandas DataFrames from Python Lists and Dictionaries



Practical Business Python - pbpython.com

```
In [ ]: records = [{"account": "Jones LLC", "Jan":150, "Feb":200, "Mar":140},
                  {"account": "Alpha Co", "Jan":200, "Feb":210, "Mar":215},
                  {"account": "Blue Inc", "Jan":50, "Feb":90, "Mar":95}]
records_df = pd.DataFrame(records)
records_df
```

```
Out[ ]:   account  Jan  Feb  Mar
0 Jones LLC  150  200  140
1 Alpha Co   200  210  215
2 Blue Inc   50   90   95
```

Simple Visualization



```
In [ ]: df = pd.read_csv('https://raw.github.com/pandas-dev/pandas/master/pandas/tests/io/c  
df
```

Out[]:

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

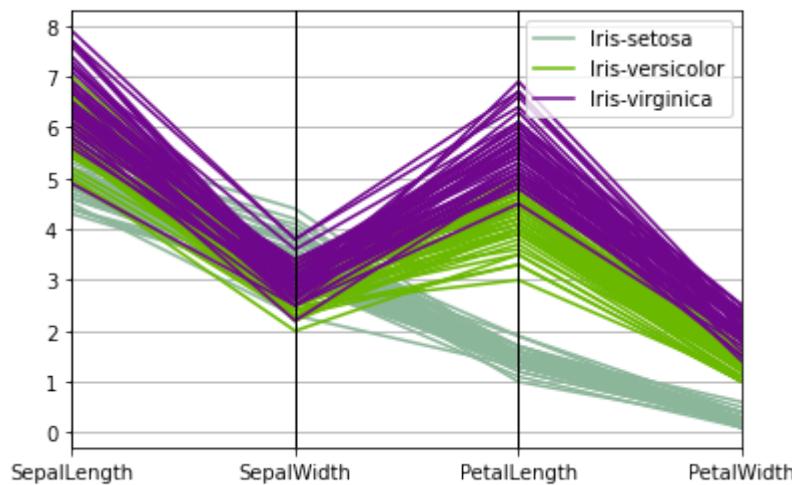
150 rows × 5 columns

```
In [ ]: df.groupby('Name').count()
```

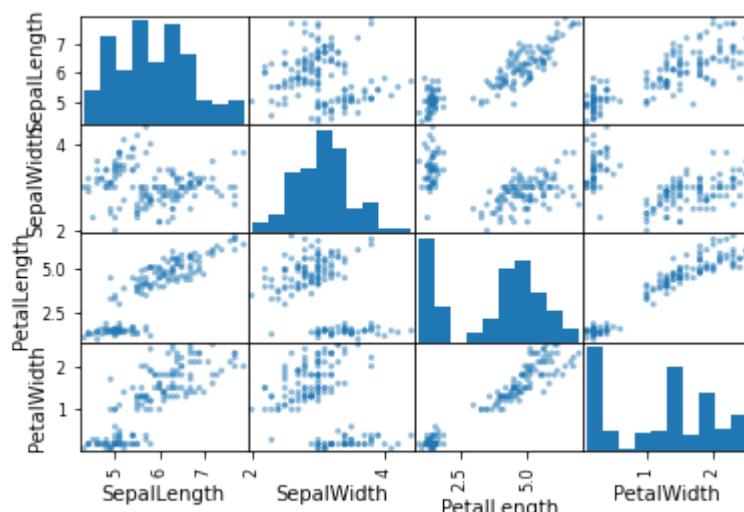
```
Out[ ]: SepalLength SepalWidth PetalLength PetalWidth
```

Name	SepalLength	SepalWidth	PetalLength	PetalWidth
Iris-setosa	50	50	50	50
Iris-versicolor	50	50	50	50
Iris-virginica	50	50	50	50

```
In [ ]: pd.plotting.parallel_coordinates(df, 'Name');
```



```
In [ ]: pd.plotting.scatter_matrix(df);
```



save table

```
In [ ]: announce_date_count = missing_sex[missing_sex['risk']=='Cluster สมุทรสาคร'].groupby(announce_date_count)
```

Out[]:

announce_date	No.	notification_date	sex	age	nationality	province_of_isolation	province_of_onset
1/14/2021	172	172	0	0	0	172	
1/17/2021	311	311	0	0	0	311	
1/18/2021	269	269	0	0	0	269	
1/20/2021	5	5	0	0	5	5	
1/5/2021	4	4	0	0	0	4	
1/7/2021	109	109	0	0	0	109	
12/20/2020	516	0	0	0	516	516	5
12/21/2020	360	0	0	0	360	360	3
12/22/2020	397	0	0	0	397	397	3
12/25/2020	35	0	0	0	35	35	
12/26/2020	12	0	0	0	12	12	
12/27/2020	36	18	0	0	18	36	
12/28/2020	14	0	0	0	14	14	
12/30/2020	3	0	0	0	1	3	
12/31/2020	15	0	0	0	15	15	

In []:

```
announce_date_count.to_csv('announce_date_count.csv')
```

บทที่ 6

(การแสดงผลการกระจายของข้อมูล)

บทที่ 6: การแสดงผลการกระจายของข้อมูล

1. การปรับสีกราฟและประเภทของโค้ดสี: การปรับสีและการใช้โค้ดสีในการสร้างกราฟ.....	1
● การใช้ตัวย่อของสี กำหนดสีได้เฉพาะสีที่เป็นแม่สี	
● การใช้ RGB value (R,G,B) กำหนดสีที่ต้องการ	
● การใช้งาน parameter: color กำหนดสีในกราฟ.....	1
● การใช้งาน parameter: colormap เลือกชุดสีที่ทางผู้เชี่ยวชาญได้จัดชุดสีมาให้แล้ว.....	2
● การใช้งาน Parameter: alpha ปรับค่าความโปร่งแสงของสี.....	9
2. การสร้างแสดงผลข้อมูลแบบ Array ด้วยภาษา Python	
● การวาดกราฟด้วย Package Matplotlib: วิธีการใช้ Matplotlib เพื่อสร้างกราฟใน Python.....	3
● การแปลงข้อมูลจาก Pandas DataFrame เป็น Numpy Array ด้วยคำสั่ง .to_numpy() เพื่อนำข้อมูลที่แปลงแล้วไปใช้งานกับ matplotlib เพราะ matplotlib ทำงานได้กับข้อมูลที่เป็นตัวเลข.....	3
● Visualize array data ด้วย pixel: ช่วยในการทำความเข้าใจข้อมูลในรูปแบบภาพ.....	6
● การเปรียบเทียบกราฟมากกว่า 1 กราฟด้วยคำสั่ง subplot: การใช้ subplot เพื่อแสดงกราฟมากกว่าหนึ่งกราฟในแผ่นผังเดียวกัน.....	8
3. การสร้างกราฟต่างๆ	
● Scatter Plot: การสร้างกราฟ scatter plot เพื่อแสดงการกระจายข้อมูล.....	11
- การใช้งาน Parameter: c เพื่อกำหนดค่าสีให้ทุกจุดและกำหนดค่าสีตามกลุ่มข้อมูลที่ต้องการ	
- การใช้งาน Parameter: alpha กำหนดความโปร่งแสง	
- การวาด Scatter Plot ด้วยฟังก์ชัน plt.plot: การใช้ plt.plot เพื่อสร้าง scatter plot ใน Matplotlib.....	14
- การวาดกราฟ 3D scatter: การสร้าง scatter plot แสดงข้อมูลใน 3 มิติ.....	16
- การใช้งาน ax.scatter สร้าง 3D scatter	
● Bubble Chart: การสร้าง Bubble Chart เพื่อแสดงข้อมูลโดยใช้ขนาดและสีของวงกลม.....	17
- การปรับแต่งกราฟ Bubble Chart: เพื่อความสวยงามและดูเข้าใจง่าย	

- การใช้งาน plt.xlim() กำหนดความกว้างของแกน x ในกราฟ.....	25
- การใช้งาน matplotlib.rcParams['figure.figsize'] กำหนดขนาดความกว้างความยาวของรูปกราฟ.....	25
● Box plot: การสร้าง Box plot เพื่อแสดงการกระจายของข้อมูลและข้อกำหนดสำคัญอื่นๆ.....	25
- การ return ค่าจาก Box-plot	
- การใช้งาน Parameter: vert=False ปรับกราฟ Box-plot เป็นแนวแนวนอน	
- การใช้งาน Parameter: showmeans แสดงค่า means บนกราฟ box-plot	
- การใช้งาน Parameter: meanline แสดง means เป็นเส้นตรงเพื่อจ่ายต่อการเปรียบเทียบ	
- การใช้งาน Parameter: meanprops เปลี่ยนลักษณะหน้าตาของ marker mean บนกราฟ	
● Violin plot: การสร้าง Violin plot เพื่อแสดงการกระจายของข้อมูลและการกระจายความหนาแน่น.....	28
- การใช้งาน Parameter: showmedians แสดง medians บนกราฟ violin-plot	
4. การกำหนดส่วนประกอบของกราฟด้วยภาษา python	
● การใช้คำสั่ง plt.title() กำหนดชื่อกราฟ.....	16
● การใช้คำสั่ง plt.xlabel และ plt.ylabel กำหนดชื่อแกน.....	16
● การใช้คำสั่ง plt.legend() และ Parameter: label ในการกำหนดชื่อตัวแปรแต่ละกลุ่มตัวแปรในกราฟ.....	14
● การกำหนดขนาดและเปลี่ยนหน้าตาของ marker ในกราฟ: การปรับขนาดและรูปร่างของ marker.....	14
- การใช้งาน Parameter: s ในการใช้ขนาดของ marker แสดงค่าของ feature	
- การใช้งาน Parameter: marker เพื่อเปลี่ยนและกำหนดหน้าตาของ Marker ตามกลุ่มที่ต้องการ	
5. timestamp – datetime.....	30
● หน้าที่และความสำคัญของข้อมูลรูปแบบ timestamp	
● การแปลงรูปแบบข้อมูลวันเดือนปีต่างๆ ให้เป็นรูปแบบ timestamp	

- การใช้คำสั่ง datetime() ในการซื้อข้อมูลรูปแบบ timestamp และยกตัวอย่างการซื้อข้อมูล
- ยกตัวอย่างการซื้อวันเดือนปีแบบ string ธรรมดा

6. (Ethic3) Health Data Sharing และ Data Privacy - กรณี covid19

- ปัญหาด้านข้อมูลที่เกิดจากการแบ่งปันข้อมูลสุขภาพและความเป็นส่วนตัวของข้อมูลในช่วงการระบาดของ COVID-19
- การรั่วไหลของข้อมูลส่วนบุคคล: การรั่วไหลของข้อมูลส่วนบุคคลสามารถนำไปสู่การฉุกเฉินเม็ดสิทธิ์ การขโมยข้อมูลประจำตัว การก่ออาชญากรรมทางการเงิน และการสูญเสียความไว้วางใจในระบบสุขภาพ
- การใช้ข้อมูลสุขภาพเพื่อวัตถุประสงค์อื่น: ข้อมูลสุขภาพที่ถูกเก็บรวบรวมในช่วงการระบาดของ COVID-19 อาจถูกนำไปใช้เพื่อวัตถุประสงค์อื่น ๆ ที่ไม่เกี่ยวข้องกับการจัดการการแพร่ระบาด เช่น การตลาด การวิจัยเชิงพาณิชย์ หรือการเฝ้าระวัง
- ความกังวลเกี่ยวกับการเก็บข้อมูลระยะยาว: ความกังวลเกี่ยวกับผลกระทบระยะยาวที่อาจเกิดขึ้นจากการเก็บรวบรวมข้อมูลจำนวนมาก เช่น การใช้ในอนาคตเพื่อการเลือกปฏิบัติหรือปฏิเสธโอกาส
- ความไม่เท่าเทียมกันในการเข้าถึงข้อมูลสุขภาพ: เช่น กลุ่มคนที่อยู่ในพื้นที่ห่างไกล กลุ่มผู้สูงอายุ หรือกลุ่มที่มีสถานะทางเศรษฐกิจต่ำ ความไม่เท่าเทียมกันในการเข้าถึงข้อมูลสุขภาพสามารถนำไปสู่การไม่ได้รับการดูแลที่เหมาะสม การขาดข้อมูลเพื่อการตัดสินใจที่ดี และการเพิ่มความเหลื่อมล้ำทางสุขภาพ

```
In [ ]: import pandas as pd
```

```
In [ ]: df = pd.read_csv('https://raw.github.com/pandas-dev/pandas/master/pandas/tests/io/data/iris.csv')
df
```

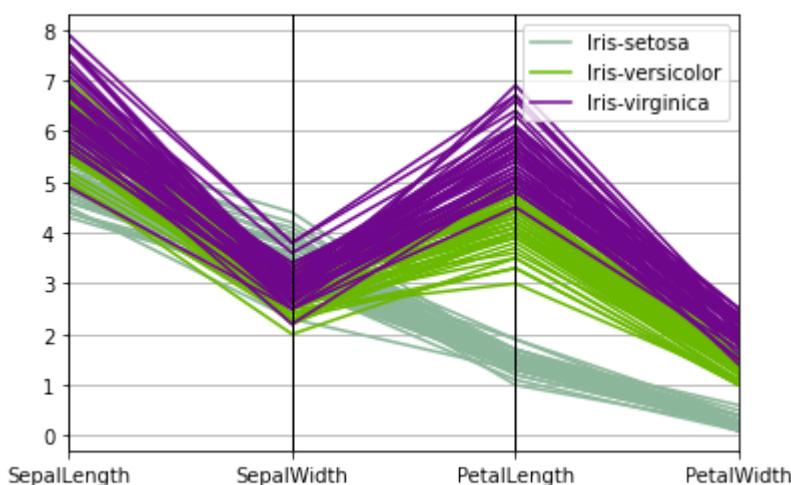
```
Out[ ]:
```

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

Parallel Coordinates

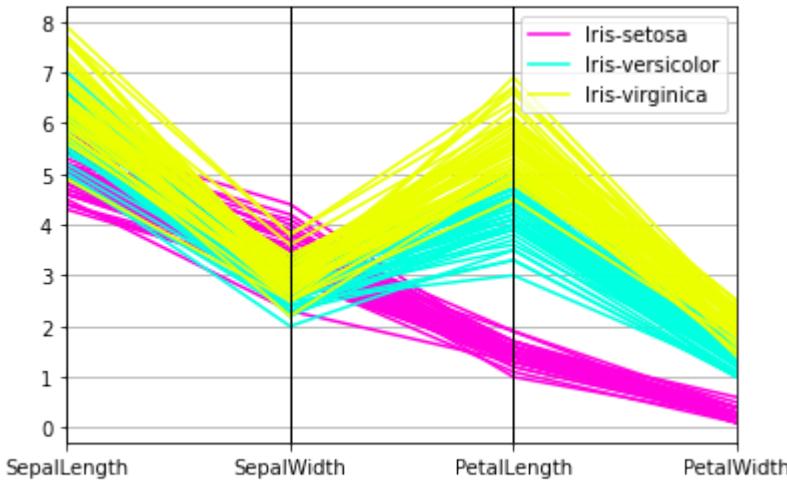
```
In [ ]: pd.plotting.parallel_coordinates(df, 'Name');
```



ปรับ parameter: color

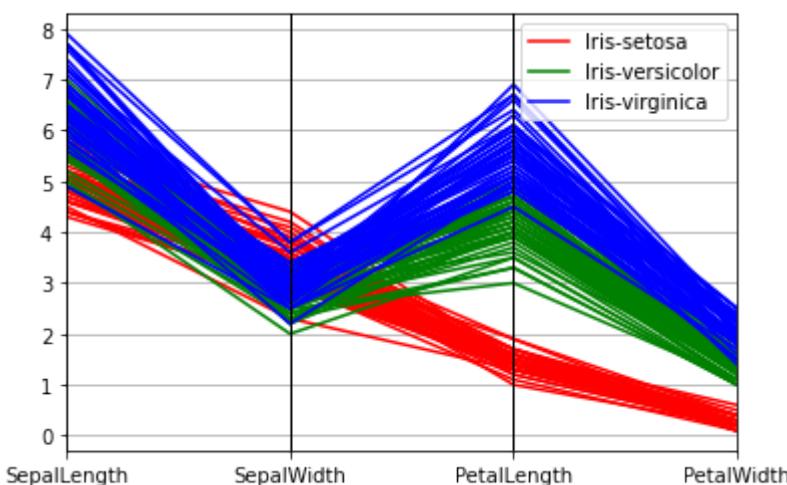
```
In [ ]: pd.plotting.parallel_coordinates(df, 'Name', color=( '#FF00E0', '#00FFE2', '#ECFF00' )
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f82e465bb90>
```



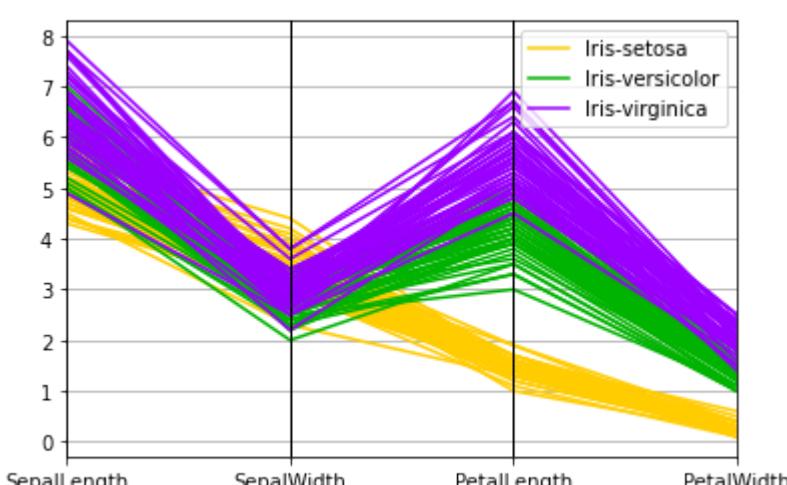
```
In [ ]: pd.plotting.parallel_coordinates(df, 'Name', color=('r', 'g', 'b')) #ตัวอย่างของสี R G B
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f82e3f1e590>
```



```
In [ ]: pd.plotting.parallel_coordinates(df, 'Name', color=((1,0.8,0), (0,0.7,0), (0.6,0,1))
```

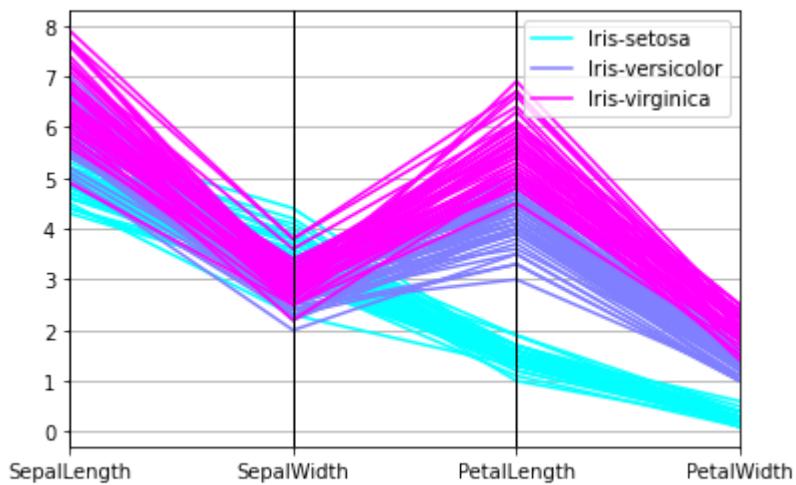
```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f82e3230790>
```



ปรับ parameter: colormap

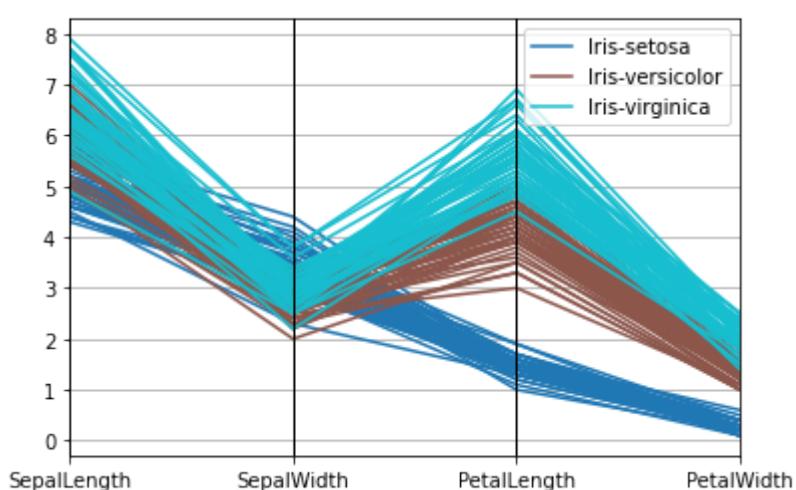
```
In [ ]: pd.plotting.parallel_coordinates(df, 'Name', colormap='cool')
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f82e2f5d090>
```



```
In [ ]: pd.plotting.parallel_coordinates(df, 'Name', colormap='tab10')
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f82e28749d0>
```



เริ่ม Matplotlib

```
In [ ]: from matplotlib import pyplot as plt
```

แปลง data จาก pandas dataframe เป็น numpy array

```
In [ ]: np_data = df.iloc[:, :-1].to_numpy()
np_data
```

```
Out[ ]: array([[5.1, 3.5, 1.4, 0.2],  
   [4.9, 3. , 1.4, 0.2],  
   [4.7, 3.2, 1.3, 0.2],  
   [4.6, 3.1, 1.5, 0.2],  
   [5. , 3.6, 1.4, 0.2],  
   [5.4, 3.9, 1.7, 0.4],  
   [4.6, 3.4, 1.4, 0.3],  
   [5. , 3.4, 1.5, 0.2],  
   [4.4, 2.9, 1.4, 0.2],  
   [4.9, 3.1, 1.5, 0.1],  
   [5.4, 3.7, 1.5, 0.2],  
   [4.8, 3.4, 1.6, 0.2],  
   [4.8, 3. , 1.4, 0.1],  
   [4.3, 3. , 1.1, 0.1],  
   [5.8, 4. , 1.2, 0.2],  
   [5.7, 4.4, 1.5, 0.4],  
   [5.4, 3.9, 1.3, 0.4],  
   [5.1, 3.5, 1.4, 0.3],  
   [5.7, 3.8, 1.7, 0.3],  
   [5.1, 3.8, 1.5, 0.3],  
   [5.4, 3.4, 1.7, 0.2],  
   [5.1, 3.7, 1.5, 0.4],  
   [4.6, 3.6, 1. , 0.2],  
   [5.1, 3.3, 1.7, 0.5],  
   [4.8, 3.4, 1.9, 0.2],  
   [5. , 3. , 1.6, 0.2],  
   [5. , 3.4, 1.6, 0.4],  
   [5.2, 3.5, 1.5, 0.2],  
   [5.2, 3.4, 1.4, 0.2],  
   [4.7, 3.2, 1.6, 0.2],  
   [4.8, 3.1, 1.6, 0.2],  
   [5.4, 3.4, 1.5, 0.4],  
   [5.2, 4.1, 1.5, 0.1],  
   [5.5, 4.2, 1.4, 0.2],  
   [4.9, 3.1, 1.5, 0.1],  
   [5. , 3.2, 1.2, 0.2],  
   [5.5, 3.5, 1.3, 0.2],  
   [4.9, 3.1, 1.5, 0.1],  
   [4.4, 3. , 1.3, 0.2],  
   [5.1, 3.4, 1.5, 0.2],  
   [5. , 3.5, 1.3, 0.3],  
   [4.5, 2.3, 1.3, 0.3],  
   [4.4, 3.2, 1.3, 0.2],  
   [5. , 3.5, 1.6, 0.6],  
   [5.1, 3.8, 1.9, 0.4],  
   [4.8, 3. , 1.4, 0.3],  
   [5.1, 3.8, 1.6, 0.2],  
   [4.6, 3.2, 1.4, 0.2],  
   [5.3, 3.7, 1.5, 0.2],  
   [5. , 3.3, 1.4, 0.2],  
   [7. , 3.2, 4.7, 1.4],  
   [6.4, 3.2, 4.5, 1.5],  
   [6.9, 3.1, 4.9, 1.5],  
   [5.5, 2.3, 4. , 1.3],  
   [6.5, 2.8, 4.6, 1.5],  
   [5.7, 2.8, 4.5, 1.3],  
   [6.3, 3.3, 4.7, 1.6],  
   [4.9, 2.4, 3.3, 1. ],  
   [6.6, 2.9, 4.6, 1.3],  
   [5.2, 2.7, 3.9, 1.4],  
   [5. , 2. , 3.5, 1. ],  
   [5.9, 3. , 4.2, 1.5],  
   [6. , 2.2, 4. , 1. ],  
   [6.1, 2.9, 4.7, 1.4],
```

[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
[5.6, 3. , 4.5, 1.5],
[5.8, 2.7, 4.1, 1.],
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
[6. , 2.9, 4.5, 1.5],
[5.7, 2.6, 3.5, 1.],
[5.5, 2.4, 3.8, 1.1],
[5.5, 2.4, 3.7, 1.],
[5.8, 2.7, 3.9, 1.2],
[6. , 2.7, 5.1, 1.6],
[5.4, 3. , 4.5, 1.5],
[6. , 3.4, 4.5, 1.6],
[6.7, 3.1, 4.7, 1.5],
[6.3, 2.3, 4.4, 1.3],
[5.6, 3. , 4.1, 1.3],
[5.5, 2.5, 4. , 1.3],
[5.5, 2.6, 4.4, 1.2],
[6.1, 3. , 4.6, 1.4],
[5.8, 2.6, 4. , 1.2],
[5. , 2.3, 3.3, 1.],
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
[5.8, 2.7, 5.1, 1.9],
[7.1, 3. , 5.9, 2.1],
[6.3, 2.9, 5.6, 1.8],
[6.5, 3. , 5.8, 2.2],
[7.6, 3. , 6.6, 2.1],
[4.9, 2.5, 4.5, 1.7],
[7.3, 2.9, 6.3, 1.8],
[6.7, 2.5, 5.8, 1.8],
[7.2, 3.6, 6.1, 2.5],
[6.5, 3.2, 5.1, 2.],
[6.4, 2.7, 5.3, 1.9],
[6.8, 3. , 5.5, 2.1],
[5.7, 2.5, 5. , 2.],
[5.8, 2.8, 5.1, 2.4],
[6.4, 3.2, 5.3, 2.3],
[6.5, 3. , 5.5, 1.8],
[7.7, 3.8, 6.7, 2.2],
[7.7, 2.6, 6.9, 2.3],
[6. , 2.2, 5. , 1.5],
[6.9, 3.2, 5.7, 2.3],
[5.6, 2.8, 4.9, 2.],
[7.7, 2.8, 6.7, 2.],
[6.3, 2.7, 4.9, 1.8],
[6.7, 3.3, 5.7, 2.1],
[7.2, 3.2, 6. , 1.8],
[6.2, 2.8, 4.8, 1.8],
[6.1, 3. , 4.9, 1.8],

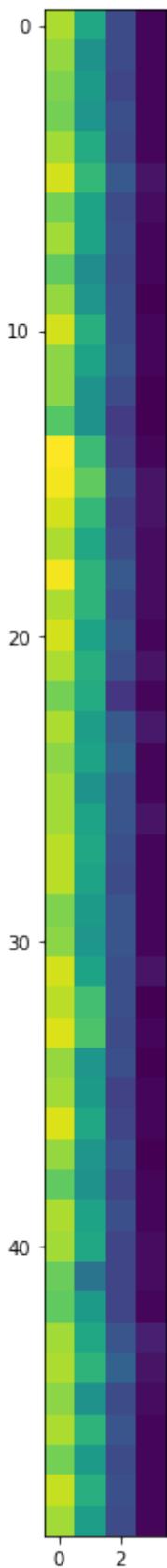
```
[6.4, 2.8, 5.6, 2.1],  
[7.2, 3. , 5.8, 1.6],  
[7.4, 2.8, 6.1, 1.9],  
[7.9, 3.8, 6.4, 2. ],  
[6.4, 2.8, 5.6, 2.2],  
[6.3, 2.8, 5.1, 1.5],  
[6.1, 2.6, 5.6, 1.4],  
[7.7, 3. , 6.1, 2.3],  
[6.3, 3.4, 5.6, 2.4],  
[6.4, 3.1, 5.5, 1.8],  
[6. , 3. , 4.8, 1.8],  
[6.9, 3.1, 5.4, 2.1],  
[6.7, 3.1, 5.6, 2.4],  
[6.9, 3.1, 5.1, 2.3],  
[5.8, 2.7, 5.1, 1.9],  
[6.8, 3.2, 5.9, 2.3],  
[6.7, 3.3, 5.7, 2.5],  
[6.7, 3. , 5.2, 2.3],  
[6.3, 2.5, 5. , 1.9],  
[6.5, 3. , 5.2, 2. ],  
[6.2, 3.4, 5.4, 2.3],  
[5.9, 3. , 5.1, 1.8]])
```

Visualize array data ด้วย pixel

```
In [ ]: import matplotlib
```

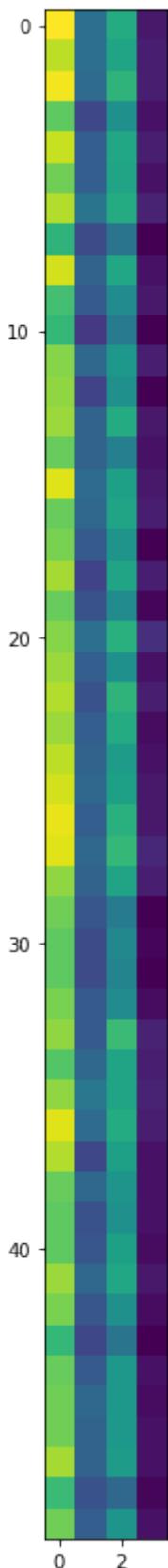
```
In [ ]: matplotlib.rcParams['figure.figsize']=[15,15] #กำหนดขนาดของ figure  
plt.imshow(np_data[:50,:],interpolation='nearest')
```

```
Out[ ]: <matplotlib.image.AxesImage at 0x7f82d9f3de90>
```



```
In [ ]: plt.imshow(np_data[50:100,:],interpolation='nearest')
```

```
Out[ ]: <matplotlib.image.AxesImage at 0x7f82d9fca310>
```



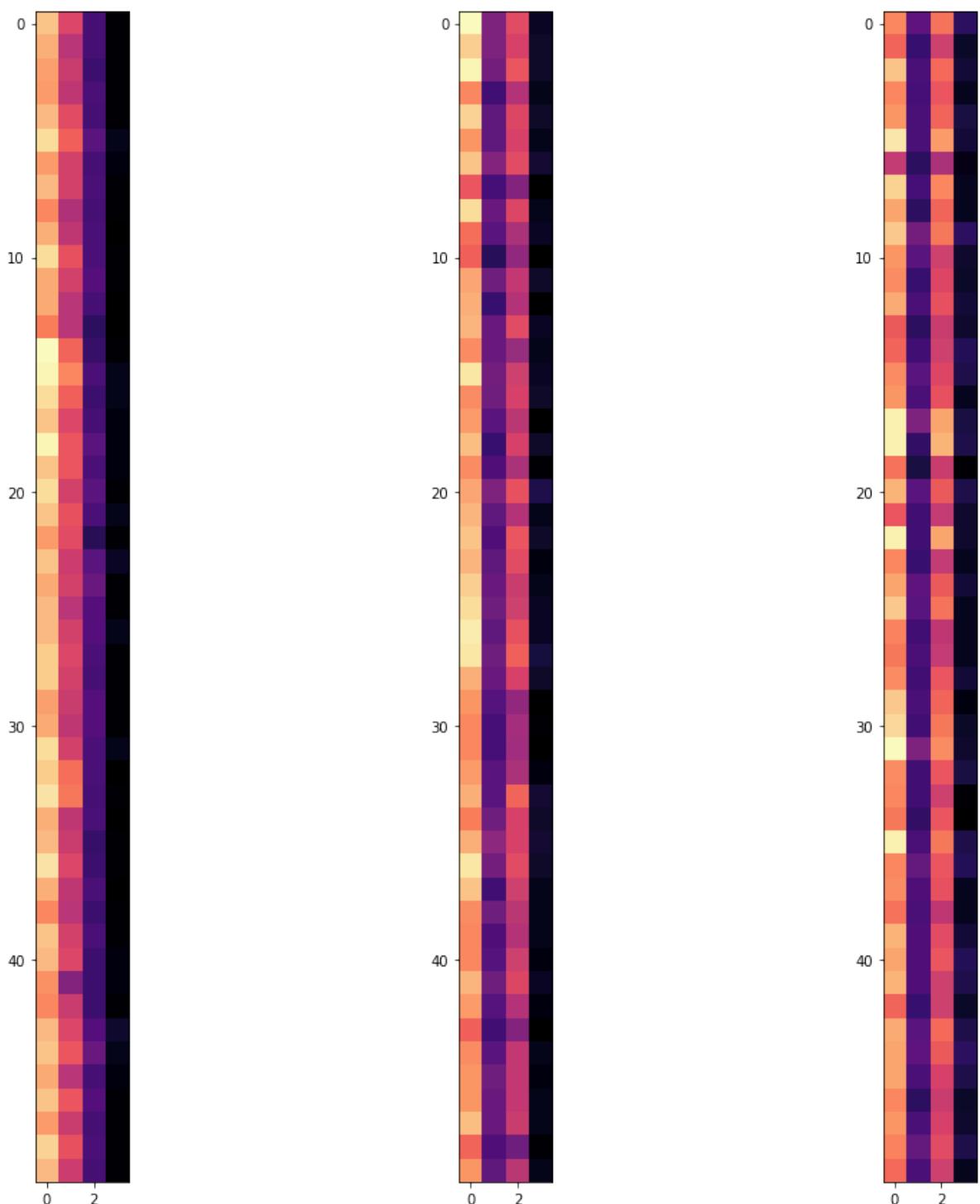
การใช้ subplot

```
In [ ]: plt.subplot(1,3,1) # subplot(มีกี่แถว, มีกี่หลัก, เป็นตัวที่เท่าไหร่)
plt.imshow(np_data[:50,:],interpolation='nearest',cmap='magma')

plt.subplot(1,3,2)
plt.imshow(np_data[50:100,:],interpolation='nearest',cmap='magma')
```

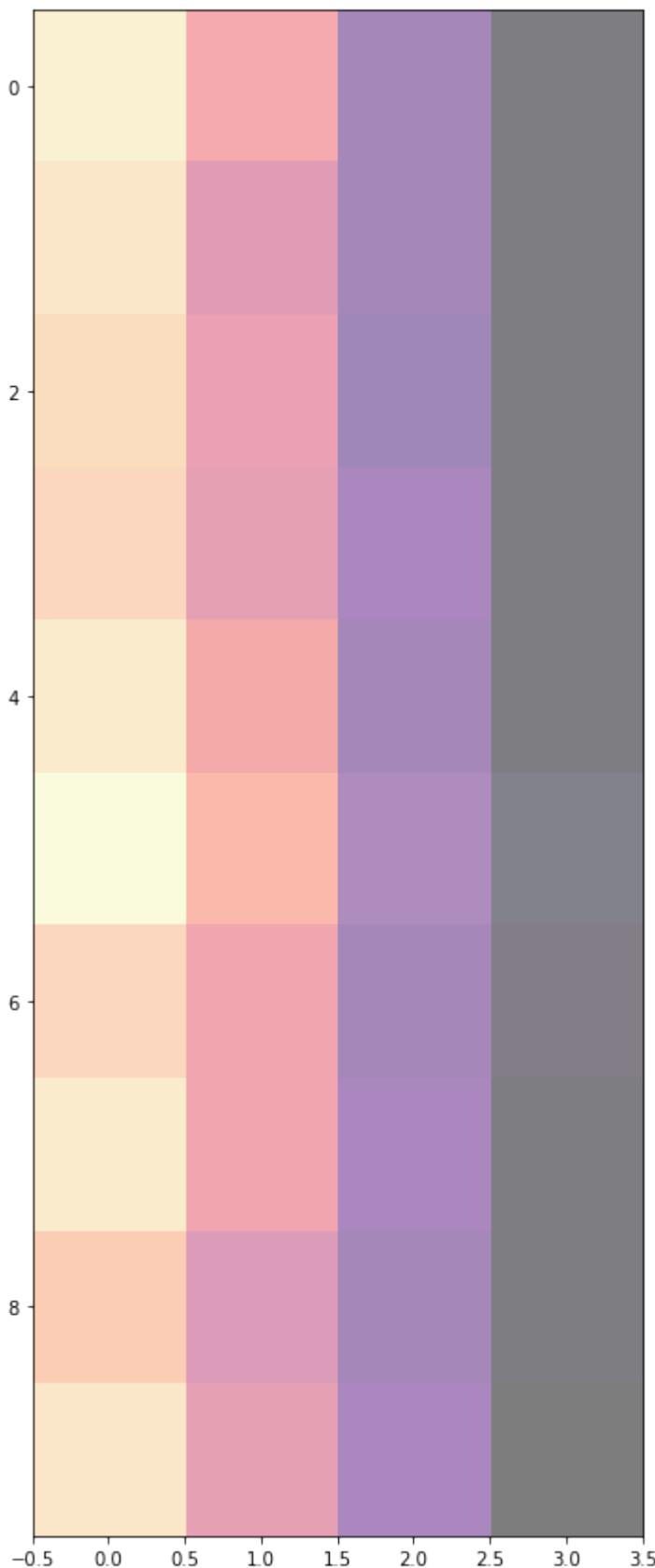
```
plt.subplot(1,3,3)
plt.imshow(np_data[100:150,:],interpolation='nearest',cmap='magma')
```

Out[]: <matplotlib.image.AxesImage at 0x7f82d9a49e50>



In []: plt.imshow(np_data[:10,:],interpolation='nearest',cmap='magma',alpha = 0.5)

Out[]: <matplotlib.image.AxesImage at 0x7f82d9895f10>



In []:

การแสดงผลการกระจายของข้อมูล

```
In [ ]: import pandas as pd  
from matplotlib import pyplot as plt
```

```
In [ ]: df = pd.read_csv('https://raw.github.com/pandas-dev/pandas/master/pandas/tests/io/csv/test.csv')  
df.head()
```

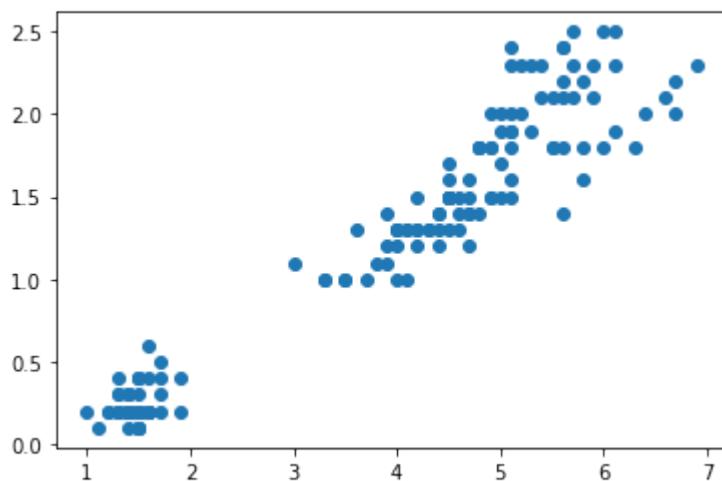
```
Out[ ]:
```

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Scatter 2D

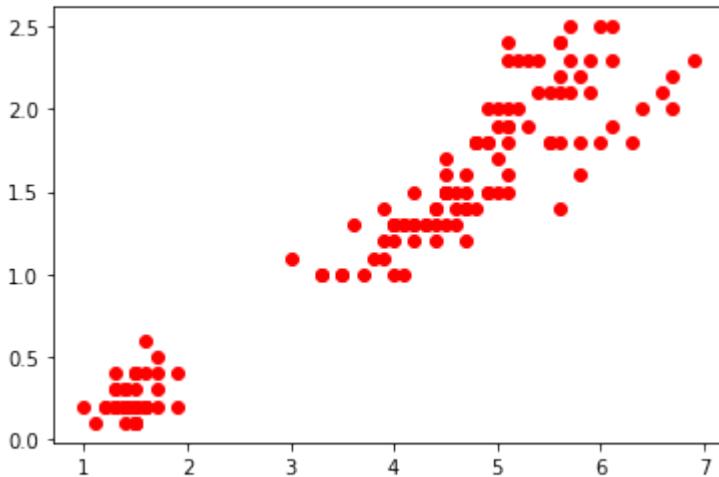
scatter - default

```
In [ ]: plt.scatter(df['PetalLength'], df['PetalWidth'])  
Out[ ]: <matplotlib.collections.PathCollection at 0x7f02775ce250>
```



color

```
In [ ]: plt.scatter(df['PetalLength'], df['PetalWidth'], c='r')  
Out[ ]: <matplotlib.collections.PathCollection at 0x7f02770c3510>
```



ໃສ່ສື່ໃຫ້ແຕ່ລະຈຸດ

In []: `set(df['Name']) #ດູວາມີຄ່າອະໄນນ໌ນັງ`

Out[]: `{'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'}`

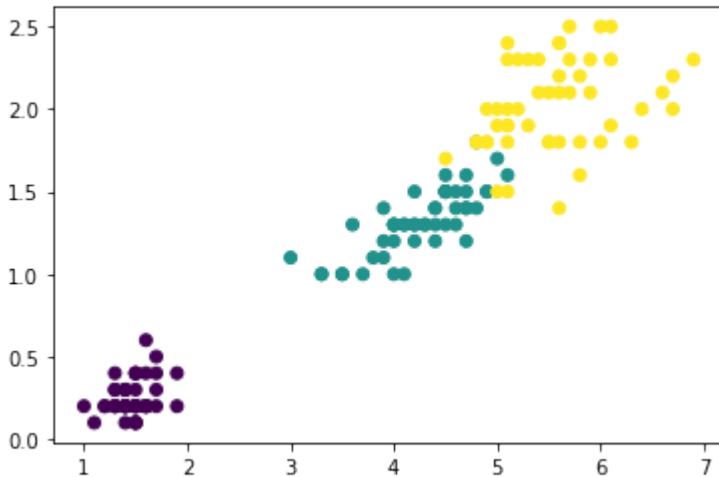
In []: `df2 = df.replace({'Iris-setosa': 0, 'Iris-versicolor': 1, 'Iris-virginica':2})
df2`

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

In []: `plt.scatter(df['PetalLength'], df['PetalWidth'], c=df2['Name'])`

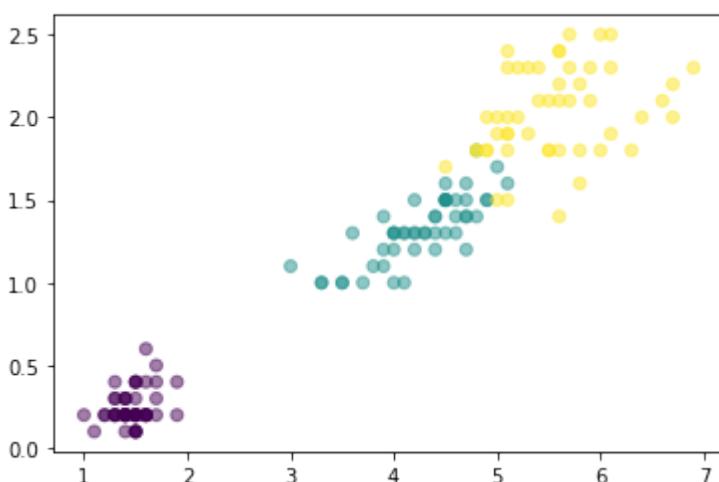
Out[]: `<matplotlib.collections.PathCollection at 0x7f0277052a10>`



alpha กำหนดความโปร่งแสง

```
In [ ]: plt.scatter(df[ 'PetalLength' ],df[ 'PetalWidth' ],c=df2[ 'Name' ],alpha=0.5)
```

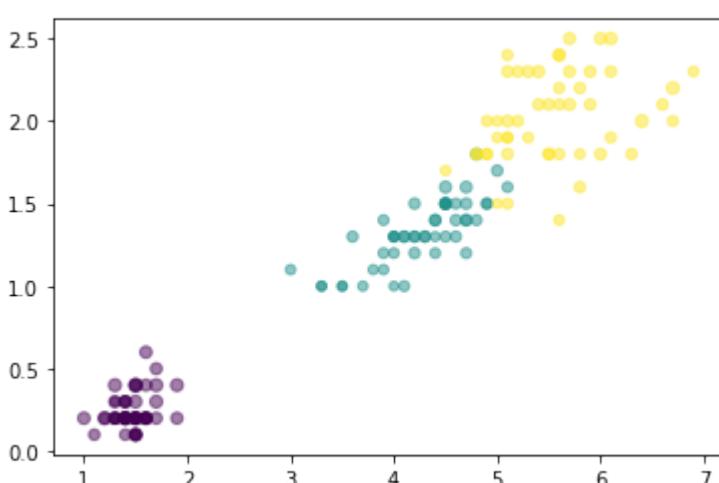
```
Out[ ]: <matplotlib.collections.PathCollection at 0x7f0277006110>
```



ใช้ขนาดของ marker แสดงค่าของ feature

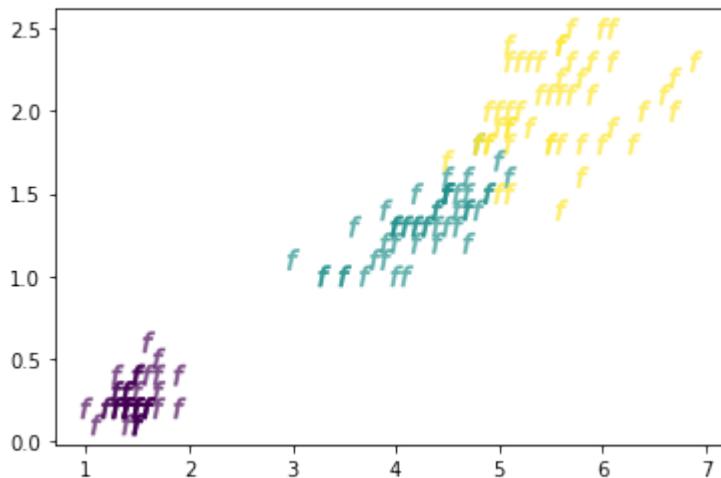
```
In [ ]: plt.scatter(df[ 'PetalLength' ],df[ 'PetalWidth' ],s=10*df[ 'SepalWidth' ],c=df2[ 'Name' ],
```

```
Out[ ]: <matplotlib.collections.PathCollection at 0x7f0275d10250>
```



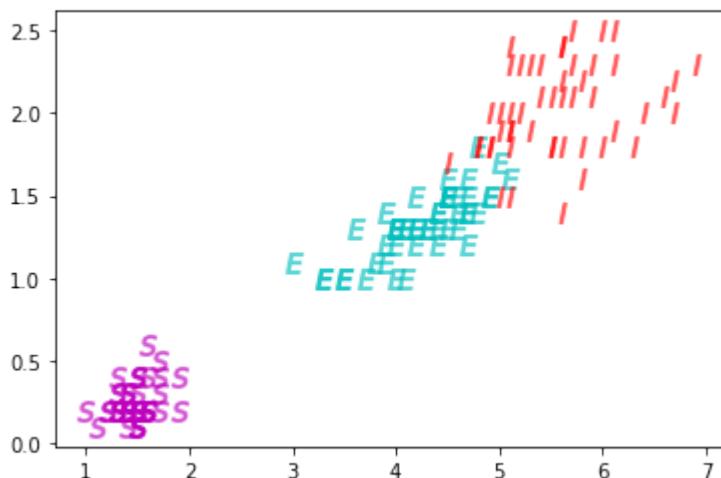
ตัวแปร Marker ใช้ในการเปลี่ยนหน้าตาของ marker

```
In [ ]: plt.scatter(df['PetalLength'], df['PetalWidth'], s=100, c=df2['Name'], alpha=0.5, marker='f')
Out[ ]: <matplotlib.collections.PathCollection at 0x7f027966d890>
```



การ plot กราฟหลายอันซ้อนกัน

```
In [ ]: plt.scatter(df['PetalLength'][:50], df['PetalWidth'][:50], s=100, c='m', alpha=0.5, marker='s')
plt.scatter(df['PetalLength'][50:100], df['PetalWidth'][50:100], s=100, c='c', alpha=0.5, marker='E')
plt.scatter(df['PetalLength'][100:], df['PetalWidth'][100:], s=100, c='r', alpha=0.5, marker='|')
Out[ ]: <matplotlib.collections.PathCollection at 0x7f0275b79c10>
```

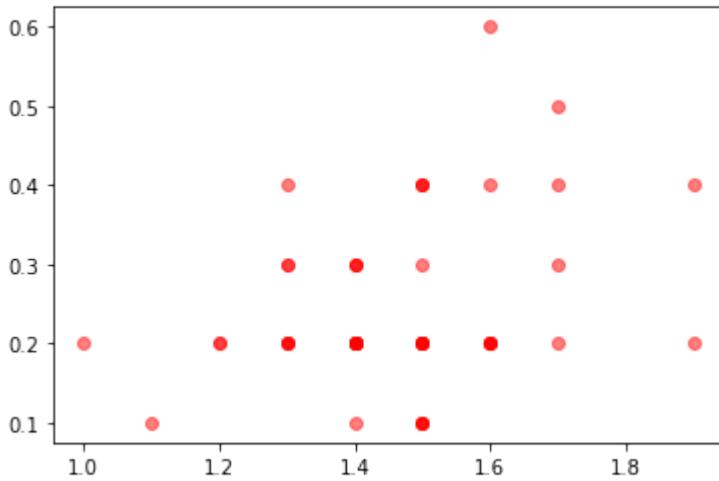


Label-Legend

`plt.plot()` input ตัวที่ 3 คือ ตัวกำหนดหน้าตาของ marker

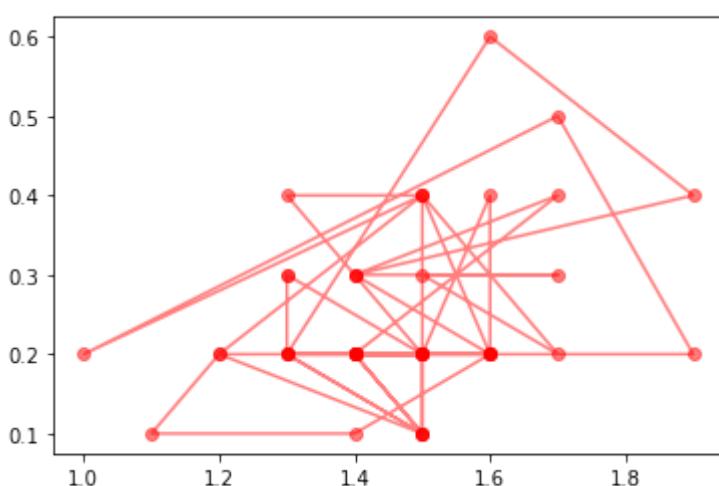
'หน้าตาmarker'-'ลักษณะของเส้นเชื่อมจุด'-'สี'

```
In [ ]: plt.plot(df['PetalLength'][:50], df['PetalWidth'][:50], 'or', alpha=0.5)
Out[ ]: [<matplotlib.lines.Line2D at 0x7f0275789c90>]
```



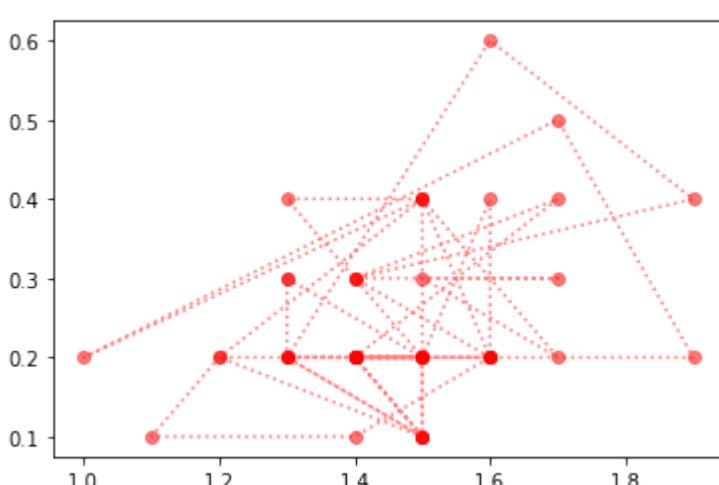
```
In [ ]: plt.plot(df['PetalLength'][:50],df['PetalWidth'][:50],'o-r',alpha=0.5)
```

```
Out[ ]: <matplotlib.lines.Line2D at 0x7f027565fe10>
```



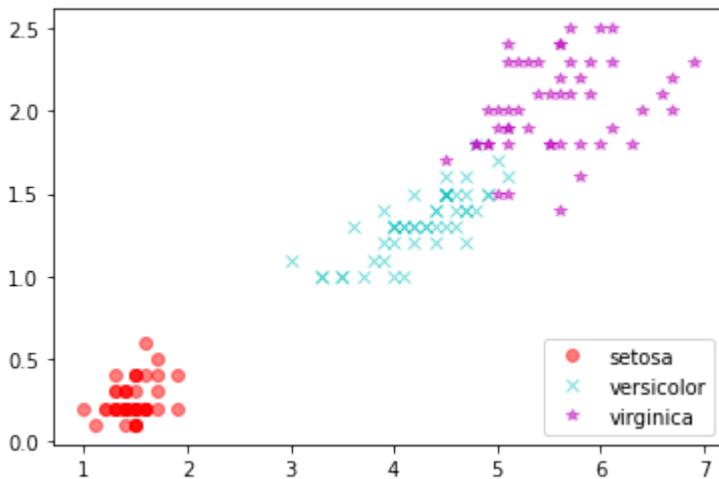
```
In [ ]: plt.plot(df['PetalLength'][:50],df['PetalWidth'][:50],'o:r',alpha=0.5)
```

```
Out[ ]: <matplotlib.lines.Line2D at 0x7f0275483650>
```



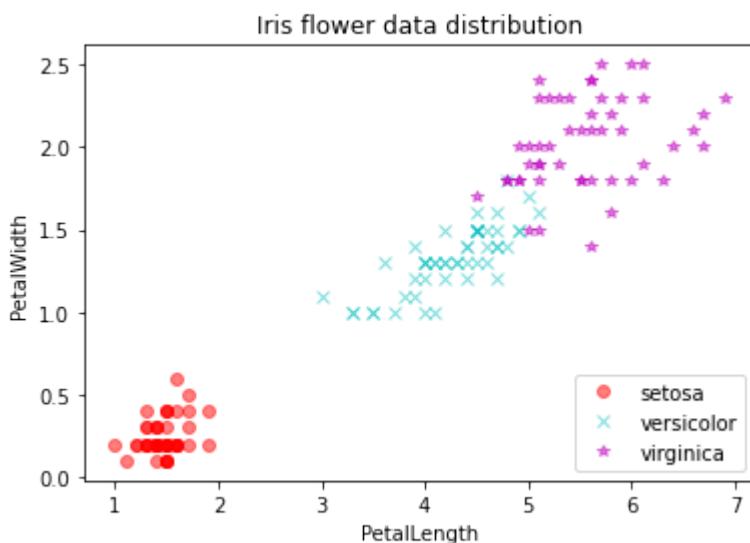
```
In [ ]: plt.plot(df['PetalLength'][:50],df['PetalWidth'][:50],'or',alpha=0.5,label='setosa'  
plt.plot(df['PetalLength'][50:100],df['PetalWidth'][50:100],'xc',alpha=0.5,label='\\'  
plt.plot(df['PetalLength'][100:],df['PetalWidth'][100:],'*m',alpha=0.5,label='virgi'  
plt.legend(loc='lower right')
```

```
Out[ ]: <matplotlib.legend.Legend at 0x7f0274f9f150>
```



การใส่ชื่อกราฟ และชื่อแกน

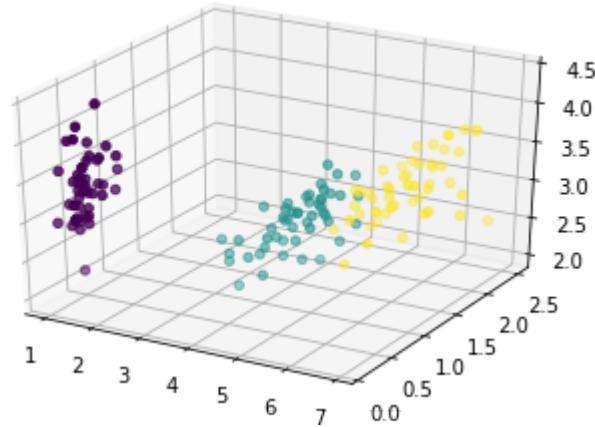
```
In [ ]: plt.plot(df['PetalLength'][:50],df['PetalWidth'][:50],'or',alpha=0.5,label='setosa'
plt.plot(df['PetalLength'][50:100],df['PetalWidth'][50:100],'xc',alpha=0.5,label='versicolor'
plt.plot(df['PetalLength'][100:],df['PetalWidth'][100:],'*m',alpha=0.5,label='virginica')
plt.legend(loc='lower right')
plt.title('Iris flower data distribution') ##ชื่อกราฟ
plt.xlabel('PetalLength') ##ชื่อแกน X
plt.ylabel('PetalWidth'); ##ชื่อแกน y
```



3D scatter

```
In [ ]: ax = plt.axes(projection ="3d")
ax.scatter3D(df['PetalLength'],df['PetalWidth'],df['SepalWidth'],c=df2['Name'])
```

```
Out[ ]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7f15a301d150>
```

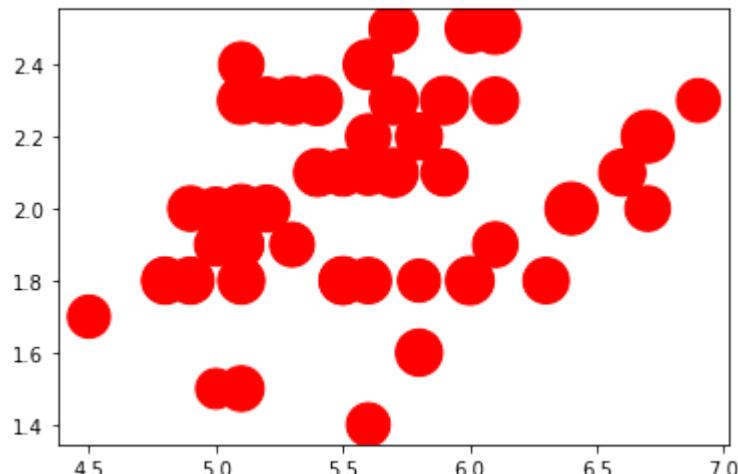


Bubble Chart

Scatter plot ที่ใช้ขนาดของ marker ในการสื่อปริมาณของข้อมูล

```
In [ ]: plt.scatter(df['PetalLength'][100:], df['PetalWidth'][100:], s=180*df['SepalWidth'][100:])

Out[ ]: <matplotlib.collections.PathCollection at 0x7f0274d72450>
```



Example

```
In [ ]: import pandas as pd
from matplotlib import pyplot as plt
```

```
In [ ]: data = pd.read_excel('/content/passenger-covid19.xlsx')
data
```

Out[]:

	ชนคน/ ชน สินค้า	โนมด การ เดิน ทาง	รูปแบบ การ เดิน ทาง	สาธารณ/ ส่วนบุคคล	หน่วย งาน	ยาน พาหนะ/ท่า	ม.ค. 2562	ก.พ. 2562	มี.ค. 2562	เม.ย.
0	ชนคน (คน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	สาธารณ ส่วนบุคคล	บขส.	รถ ขบส. และ รถร่วม	5236331	4628878.0	5234387.0	513
1	ชนคน (คน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	สาธารณ ส่วนบุคคล	ขบ.	รถ Taxi	Nan	Nan	Nan	
2	ชนคน (คน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	สาธารณ ส่วนบุคคล	ขบ.	รถหมวด 3	2550864	2334128.0	2599144.0	260
3	ชนคน (คัน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	ส่วนบุคคล	หล.	รถยนต์ (10 ล้อ สีขาว)	28907234	26699682.0	30210655.0	2854
4	ชนคน (คัน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	ส่วนบุคคล	กทพ.	รถยนต์ (ทางด่วน)	56404661	51536713.0	57376504.0	4683
5	ชนคน (คัน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	ส่วนบุคคล	หล.	รถ จักรยานยนต์	Nan	Nan	Nan	
6	ชนคน (คน)	ทาง บก	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ ส่วนบุคคล	ขบ.	รถหมวด 4	108716	103437.0	102001.0	10
7	ชนคน (คน)	ทาง บก	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ ส่วนบุคคล	ขบ.	รถ จักรยานยนต์ รับจ้าง	Nan	Nan	Nan	
8	ชนคน (คน)	ทาง บก	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ ส่วนบุคคล	ขสมก.	รถเมล์ ขสมก.	32756201	29567092.0	32278496.0	2900
9	ชนคน (คน)	ทาง บก	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ ส่วนบุคคล	ขสมก.	รถร่วม (หมวด 1)	28753451	27871978.0	28249937.0	2387
10	ชนคน (คัน)	ทาง บก	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	ส่วนบุคคล	หล.	รถยนต์	Nan	Nan	Nan	
11	ชนคน (คัน)	ทาง บก	การเดิน ทาง ภายใน	ส่วนบุคคล	หล.	รถ จักรยานยนต์	Nan	Nan	Nan	

ขบวนค์/ ขบวนสินค้า	ที่มุ่งเดินทาง	รูปแบบการเดินทาง	สถานะ%/ส่วนบุคคล	หน่วยงาน	พานหนะ/ท่า	ม.ค. 2562	ก.พ. 2562	มี.ค. 2562	เม.ย.
จังหวัด/ กรุงเทพ									
12	ขบวนค์ (คน)	ทางน้ำ	การเดินทางระหว่างจังหวัด	สาธารณสุข	จท.	เรือภูมิภาค	ไม่มีการจัดเก็บข้อมูล	NaN	NaN
13	ขบวนค์ (คน)	ทางน้ำ	การเดินทางภายในจังหวัด/กรุงเทพ	สาธารณสุข	จท.	เรือเจ้าพระยา	968060	848038.0	819430.0 7
14	ขบวนค์ (คน)	ทางน้ำ	การเดินทางภายในจังหวัด/กรุงเทพ	สาธารณสุข	จท.	เรือคลองแสนแสบ	818195	800744.0	820090.0 6
15	ขบวนค์ (คน)	ทางน้ำ	การเดินทางภายในจังหวัด/กรุงเทพ	สาธารณสุข	จท.	เรือภูมิภาค	6288628	6308324.0	6494292.0 804
16	ขบวนค์ (คน)	ทางน้ำ	การเดินทางภายในจังหวัด/กรุงเทพ	ส่วนบุคคล	จท.	เรือส่วนบุคคล	ไม่มีการจัดเก็บข้อมูล	NaN	NaN
17	ขบวนค์ (คน)	ทางราก	การเดินทางระหว่างจังหวัด	สาธารณสุข	รฟท.	รถไฟฟ้า	2715054	2388220.0	2797101.0 273
18	ขบวนค์ (คน)	ทางราก	การเดินทางภายในจังหวัด/กรุงเทพ	สาธารณสุข	รฟม.	รถไฟฟ้าสายสีน้ำเงิน	9080423	8334110.0	9462844.0 820
19	ขบวนค์ (คน)	ทางราก	การเดินทางภายในจังหวัด/กรุงเทพ	สาธารณสุข	รฟม.	รถไฟฟ้าสายสีม่วง	1172956	1060635.0	1224857.0 107
20	ขบวนค์ (คน)	ทางราก	การเดินทางภายในจังหวัด/กรุงเทพ	สาธารณสุข	รฟฟท.	รถไฟฟ้า ARL	2167528	1986306.0	2294142.0 207
21	ขบวนค์ (คน)	ทางราก	การเดินทางภายในจังหวัด/กรุงเทพ	สาธารณสุข	กทม.	รถไฟฟ้า BTS	20800000	19000000.0	21300000.0 1870
22	ขบวนค์ (คน)	ทางอากาศ	การเดินทาง	สาธารณสุข	ทอท.	ท่าอากาศยาน	458709	425178.0	430965.0 4

ขนดค/ ขน สินค้า	ห้องด การ เดิน ทาง	รูปแบบ การ เดิน ทาง	สารณ/ ส่วนบุคคล	หน่วย งาน	ยาน พาหนะ/ท่า	ม.ค. 2562	ก.พ. 2562	มี.ค. 2562	เม.ย.
ระหว่าง จังหวัด								สุวรรณภูมิ	
23	ขนดค/ คน	ทาง อากาศ	การเดิน ทาง ระหว่าง จังหวัด	สารณ สภาพ	ทอท.	ท่า อากาศ ดอนเมือง	1003125	934616.0	1009713.0
24	ขนดค/ คน	ทาง อากาศ	การเดิน ทาง ระหว่าง จังหวัด	สารณ สภาพ	ทอท.	ท่าอากาศ อื่น ๆ ของ ทอท.	1076360	979634.0	1005178.0
25	ขนดค/ คน	ทาง อากาศ	การเดิน ทาง ระหว่าง จังหวัด	สารณ สภาพ	ทบ.	ท่า อากาศ ภูมิภาค	719670	663355.0	728778.0
26	ขนดค/ คน	ทาง ถนน	การเดิน ทาง ระหว่าง ประเทศ	สารณ สภาพ	บขส.	รถ บขส. ขา เข้าประเทศ	24221	19600.0	19874.0
27	ขนดค/ คน	ทาง ถนน	การเดิน ทาง ระหว่าง ประเทศ	สารณ สภาพ	บขส.	รถ บขส. ขา ออกประเทศ	28670	25574.0	31910.0
28	ขนดค/ คน	ทาง ร้าง	การเดิน ทาง ระหว่าง ประเทศ	สารณ สภาพ	รฟท.	รถไฟ ขา เข้าประเทศ (ป่า ตังเบชาร์)	303	363.0	251.0
29	ขนดค/ คน	ทาง ร้าง	การเดิน ทาง ระหว่าง ประเทศ	สารณ สภาพ	รฟท.	รถไฟ ขา ออกประเทศ (ป่า ตังเบชาร์)	598	693.0	495.0
30	ขนดค/ คน	ทาง น้ำ	การเดิน ทาง ระหว่าง ประเทศ	สารณ สภาพ	จท.	ท่าเรือต่าน ชา เข้าประเทศ	ไม่มีการ จัดเก็บ ข้อมูล	NaN	NaN
31	ขนดค/ คน	ทาง น้ำ	การเดิน ทาง ระหว่าง ประเทศ	สารณ สภาพ	จท.	ท่าเรือต่าน ชา ออกประเทศ	ไม่มีการ จัดเก็บ ข้อมูล	NaN	NaN
32	ขนดค/ คน	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	สารณ สภาพ	ทอท.	ท่า อากาศยาน สุวรรณภูมิ ขาเข้า ประเทศ	2464894	2224587.0	2294408.0
33	ขนดค/ คน	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	สารณ สภาพ	ทอท.	ท่า อากาศยาน สุวรรณภูมิ ขาออก ประเทศ	2028379	1919774.0	2064603.0
34	ขนดค/ คน	ทาง อากาศ	การเดิน ทาง	สารณ สภาพ	ทอท.	ท่า อากาศยาน ดอนเมือง	748089	702251.0	774448.0

ขนคน/ สินค้า	ที่มด การ เดิน ทาง	รูปแบบ การ เดิน ทาง	สารณ/ ส่วนบุคคล	หน่วย งาน	ยาน พาหนะ/ท่า	ม.ค. 2562	ก.พ. 2562	มี.ค. 2562	เม.ย.
ระหว่าง ประเทศ					ขาเข้า ประเทศ				
35	ขนคน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	สารณ ทาง	ทอท.	ท่า อากาศยาน ดอนเมือง ข้าวอก ประเทศ	702441	978370.0	748852.0
36	ขนคน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	สารณ ทาง	ทอท.	ท่าอากาศ อื่น ๆ ของ ทอท. ขาเข้า ประเทศ	753017	710321.0	698460.0
37	ขนคน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	สารณ ทาง	ทอท.	ท่าอากาศ อื่น ๆ ของ ทอท. ขา ออกประเทศ	750052	733723.0	741191.0
38	ขนคน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	สารณ ทาง	ทย.	ท่า อากาศยาน ภูมิภาค ขา เข้าประเทศ	98581	108408.0	71729.0
39	ขนคน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	สารณ ทาง	ทย.	ท่า อากาศยาน ภูมิภาค ขา ออกประเทศ	96843	116047.0	83223.0

```
In [ ]: data_drop = data.dropna()
data_drop
```

Out[]:

	ชนคน/ ชน สินค้า	หมวด การ เดิน ทาง	รูปแบบ การ เดิน ทาง	สาธารณ/ ส่วนบุคคล	หน่วย งาน	ยาน พาหนะ/ ท่า	ม.ค. 2562	ก.พ. 2562	มี.ค. 2562	เม.ย.
0	ชนคน (คน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	สาธารณ ส่วนบุคคล	บขส.	รถ ขบส. และ รถ ร่วม	5236331	4628878.0	5234387.0	5137
2	ชนคน (คน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	สาธารณ	ขบ.	รถหมวด 3	2550864	2334128.0	2599144.0	2602
3	ชนคน (คัน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	ส่วนบุคคล	ทล.	รถยนต์ (10 จด สำรวจ)	28907234	26699682.0	30210655.0	28542
4	ชนคน (คัน)	ทาง บก	การเดิน ทาง ระหว่าง จังหวัด	ส่วนบุคคล	กทพ.	รถยนต์ (ทางด่วน)	56404661	51536713.0	57376504.0	46838
6	ชนคน (คน)	ทาง บก	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ	ขบ.	รถหมวด 4	108716	103437.0	102001.0	102
8	ชนคน (คน)	ทาง บก	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ	ขสมก.	รถเมล์ ขสมก.	32756201	29567092.0	32278496.0	29002
9	ชนคน (คน)	ทาง บก	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ	ขสมก.	รถร่วม (หมวด 1)	28753451	27871978.0	28249937.0	23874
13	ชนคน (คน)	ทาง น้ำ	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ	จท.	เรือ เจ้าพระยา	968060	848038.0	819430.0	712
14	ชนคน (คน)	ทาง น้ำ	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ	จท.	เรือคลอง แสนแสบ	818195	800744.0	820090.0	636
15	ชนคน (คน)	ทาง น้ำ	การเดิน ทาง ภายใน จังหวัด/ กรุงเทพ	สาธารณ	จท.	เรือภูมิภาค	6288628	6308324.0	6494292.0	8040
17	ชนคน (คน)	ทาง ราง	การเดิน ทาง ระหว่าง จังหวัด	สาธารณ	รฟท.	รถไฟ	2715054	2388220.0	2797101.0	2736

ขบวนคุ้มครอง สินค้า	ที่มา การเดินทาง	รูปแบบ การเดินทาง	สารณ์/ ส่วนบุคคล	หน่วยงาน	ยานพาหนะ/ ท่า	ม.c. 2562	ก.พ. 2562	มี.ค. 2562	เม.ย.
18 ขบวน (คน)	ทาง ทาง ทาง	การเดิน ภายใน จังหวัด/ กรุงเทพ	สารณ์ สาย รฟม.	รถไฟฟ้า สีน้ำเงิน	สาย	9080423	8334110.0	9462844.0	8206
19 ขบวน (คน)	ทาง ทาง ทาง	การเดิน ภายใน จังหวัด/ กรุงเทพ	สารณ์ รฟม.	รถไฟฟ้า สายสีม่วง	สาย	1172956	1060635.0	1224857.0	1072
20 ขบวน (คน)	ทาง ทาง ทาง	การเดิน ภายใน จังหวัด/ กรุงเทพ	สารณ์ รฟฟท.	รถไฟฟ้า ARL	สาย	2167528	1986306.0	2294142.0	2075
21 ขบวน (คน)	ทาง ทาง ทาง	การเดิน ภายใน จังหวัด/ กรุงเทพ	สารณ์ กทม.	รถไฟฟ้า BTS	สาย	20800000	19000000.0	21300000.0	18700
22 ขบวน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง จังหวัด	สารณ์ ทอท.	ท่า อากาศยาน สุวรรณภูมิ	ท่า	458709	425178.0	430965.0	411
23 ขบวน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง จังหวัด	สารณ์ ทอท.	ท่า อากาศยาน ดอนเมือง	ท่า	1003125	934616.0	1009713.0	978
24 ขบวน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง จังหวัด	สารณ์ ทอท.	ท่าอากาศ ที่ 1 ของ ทอท.	ท่าอากาศ	1076360	979634.0	1005178.0	931
25 ขบวน (คน)	ทาง อากาศ	การเดิน ทาง ระหว่าง จังหวัด	สารณ์ ทบ.	ท่า อากาศยาน ภูมิภาค	ท่า	719670	663355.0	728778.0	710
26 ขบวน (คน)	ทาง ถนน	การเดิน ทาง ระหว่าง ประเทศ	สารณ์ บขส.	รถ บขส. ขาเข้า ประเทศ	รถ บขส.	24221	19600.0	19874.0	32
27 ขบวน (คน)	ทาง ถนน	การเดิน ทาง ระหว่าง ประเทศ	สารณ์ บขส.	รถ บขส. ขาออก ประเทศ	รถ บขส.	28670	25574.0	31910.0	42
28 ขบวน (คน)	ทาง ทาง ทาง	การเดิน ทาง ระหว่าง ประเทศ	สารณ์ รฟท.	รถไฟฟ้า ขาเข้า ประเทศไทย (ปอดังเบนชาร์)	รถไฟฟ้า ขาเข้า ประเทศไทย	303	363.0	251.0	
29 ขบวน (คน)	ทาง ทาง	การเดินทาง	สารณ์ รฟท.	รถไฟฟ้า ขาออก ประเทศไทย	รถไฟฟ้า ขาออก ประเทศไทย	598	693.0	495.0	

ขนดค/ ขน สินค้า	ที่มติ	รูปแบบ	การ เดิน ทาง	การ เดิน ทาง	สาธารณ/ ส่วนบุคคล	หน่วย	ยาน พาหนะ/ ท่า	ม.ค. 2562	ก.พ. 2562	มี.ค. 2562	เม.ย.
		ระหว่าง ประเทศ				(ป้าดังเบ ชาร์)					
32	ขนดค/ (คณ)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	การเดิน ทาง ระหว่าง ประเทศ	สาธารณ ส่วนบุคคล	ทอท.	อากาศยาน สุวรรณภูมิ ขาเข้า ^{ประเทศ}	2464894	2224587.0	2294408.0	2227
33	ขนดค/ (คณ)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	การเดิน ทาง ระหว่าง ประเทศ	สาธารณ ส่วนบุคคล	ทอท.	อากาศยาน สุวรรณภูมิ ขาออก ^{ประเทศ}	2028379	1919774.0	2064603.0	1891
34	ขนดค/ (คณ)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	การเดิน ทาง ระหว่าง ประเทศ	สาธารณ ส่วนบุคคล	ทอท.	อากาศยาน ดอนเมือง ขาเข้า ^{ประเทศ}	748089	702251.0	774448.0	764
35	ขนดค/ (คณ)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	การเดิน ทาง ระหว่าง ประเทศ	สาธารณ ส่วนบุคคล	ทอท.	อากาศยาน ดอนเมือง ขาออก ^{ประเทศ}	702441	978370.0	748852.0	707
36	ขนดค/ (คณ)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	การเดิน ทาง ระหว่าง ประเทศ	สาธารณ ส่วนบุคคล	ทอท.	ท่าอากาศ ยื่น ๆ ของ ท่าฯ. ขา เข้า ^{ประเทศ}	753017	710321.0	698460.0	591
37	ขนดค/ (คณ)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	การเดิน ทาง ระหว่าง ประเทศ	สาธารณ ส่วนบุคคล	ทอท.	ท่าอากาศ ยื่น ๆ ของ ท่าฯ. ขา ออก ^{ประเทศ}	750052	733723.0	741191.0	600
38	ขนดค/ (คณ)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	การเดิน ทาง ระหว่าง ประเทศ	สาธารณ ส่วนบุคคล	ทบ.	อากาศยาน ภูมิภาค ขา เข้า ^{ประเทศ}	98581	108408.0	71729.0	50
39	ขนดค/ (คณ)	ทาง อากาศ	การเดิน ทาง ระหว่าง ประเทศ	การเดิน ทาง ระหว่าง ประเทศ	สาธารณ ส่วนบุคคล	ทบ.	อากาศยาน ภูมิภาค ขา ออก ^{ประเทศ}	96843	116047.0	83223.0	53

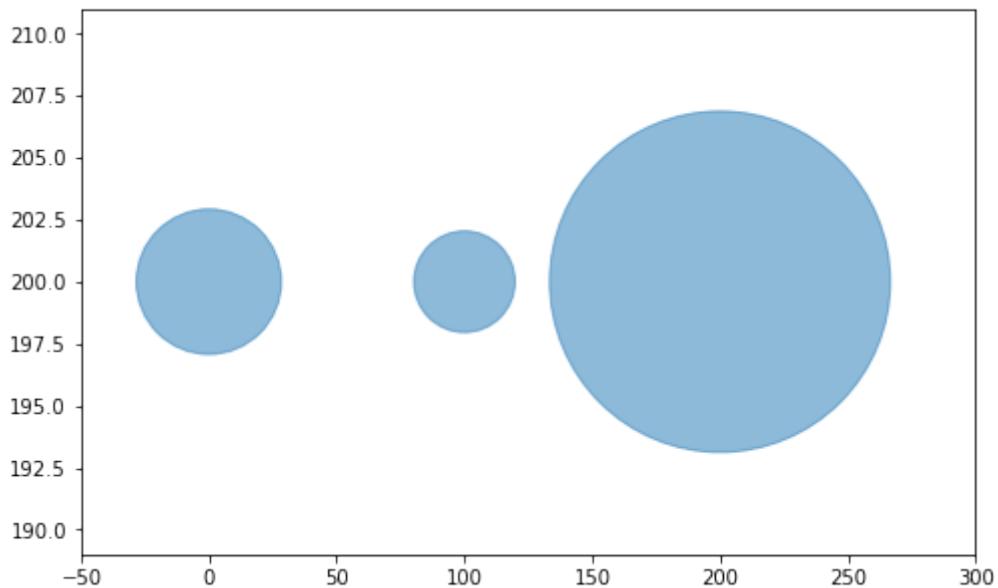
In []: 0.001 * data_drop.iloc[:3,6]

Out[]: 0 5236.33
2 2550.86
3 28907.2
Name: ม.ค. 2562, dtype: object

In []: import numpy as np
import matplotlib

```
In [ ]: matplotlib.rcParams['figure.figsize']=[8,5]
```

```
In [ ]: plt.scatter([0,100,200],[200,200,200],s=list(0.001*data_drop.iloc[:3,6]),alpha=0.5)  
plt.xlim((-50,300));
```



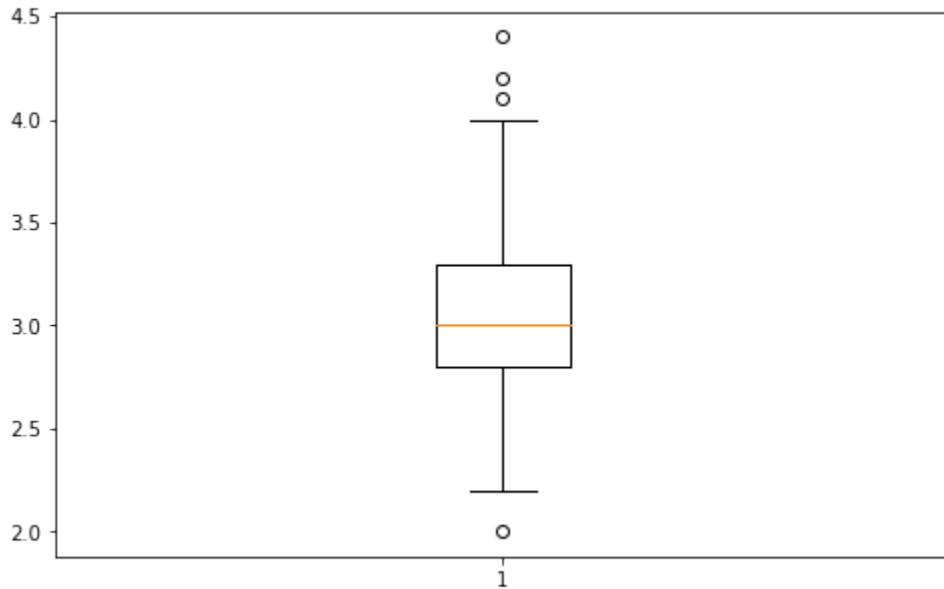
Box-plot

```
In [ ]: df = pd.read_csv('https://raw.github.com/pandas-dev/pandas/master/pandas/tests/io/c  
df.head()
```

```
Out[ ]:
```

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
In [ ]: o = plt.boxplot(df['SepalWidth'])
```



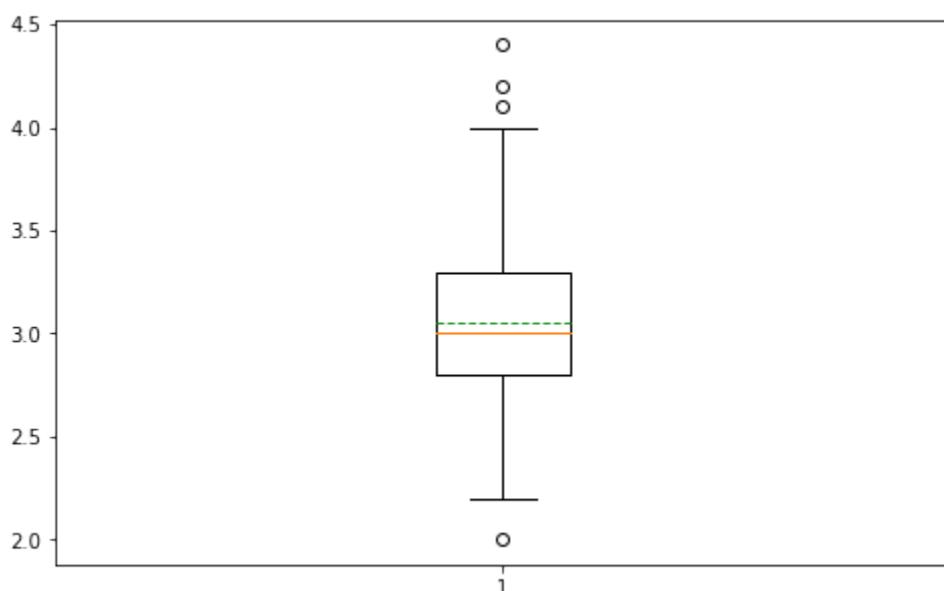
```
In [ ]: 0
```

```
Out[ ]: {'boxes': [ 'caps': [ <matplotlib.lines.Line2D at 0x7f7f44809390>],  
 'fliers': [ 'means': [],  
 'medians': [ 'whiskers': [ <matplotlib.lines.Line2D at 0x7f7f448208d0>]}
```

```
In [ ]: 0['fliers'][0].get_ydata()
```

```
Out[ ]: array([2. , 4.4, 4.1, 4.2])
```

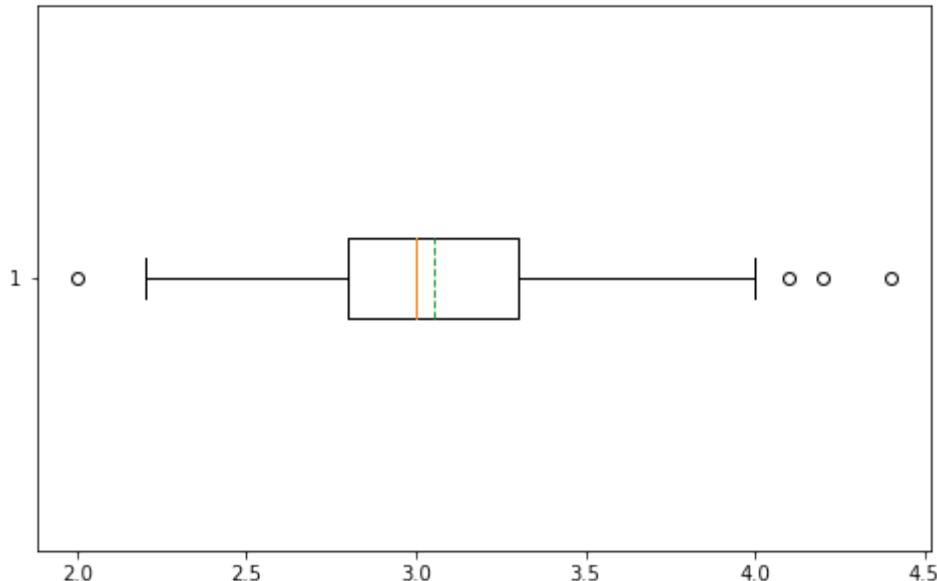
```
In [ ]: 0 = plt.boxplot(df['SepalWidth'], showmeans=True, meanline=True)
```



```
In [ ]: 0
```

```
Out[ ]: {'boxes': [ 'caps': [ <matplotlib.lines.Line2D at 0x7f7f44226490>],  
 'fliers': [ 'means': [ 'medians': [ 'whiskers': [ <matplotlib.lines.Line2D at 0x7f7f442209d0>]}
```

```
In [ ]: o = plt.boxplot(df['SepalWidth'], showmeans=True, meanline=True, vert=False)
```



- เปรียบเทียบ boxplot ของดอกไม้หลายชนิด

```
In [ ]:
```

Violin-plot

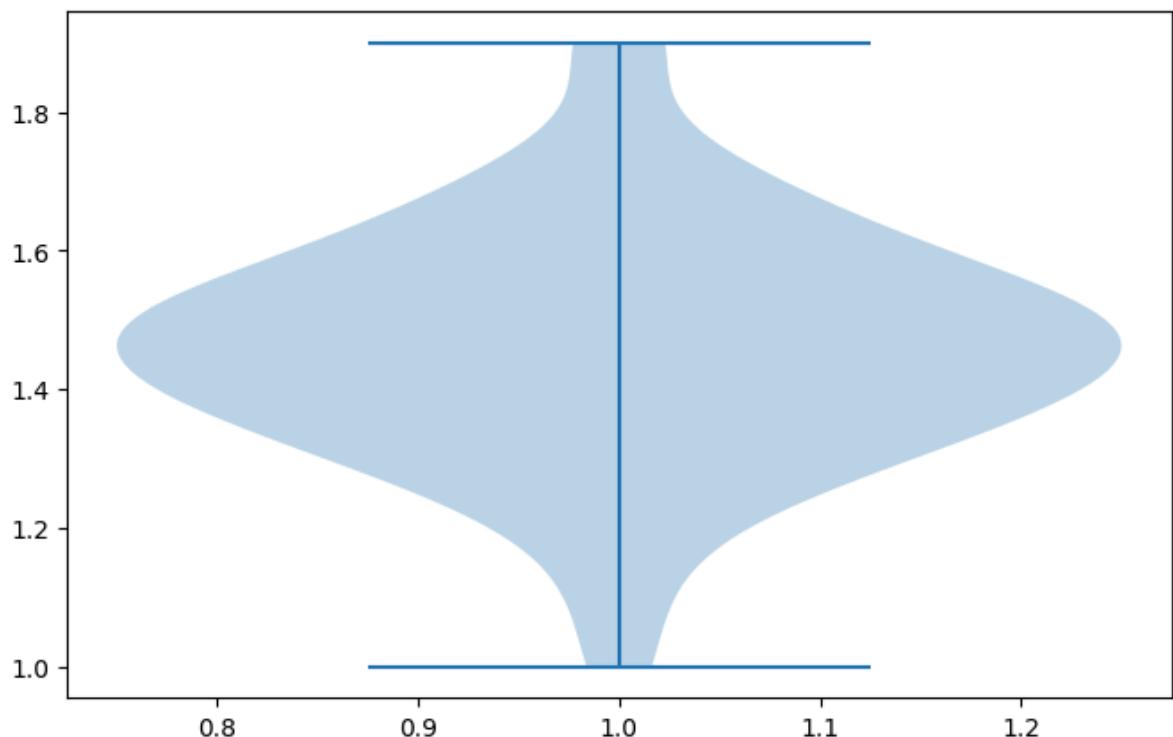
```
In [ ]: df = pd.read_csv('https://raw.github.com/pandas-dev/pandas/master/pandas/tests/io/c  
df.head()
```

```
Out[ ]:
```

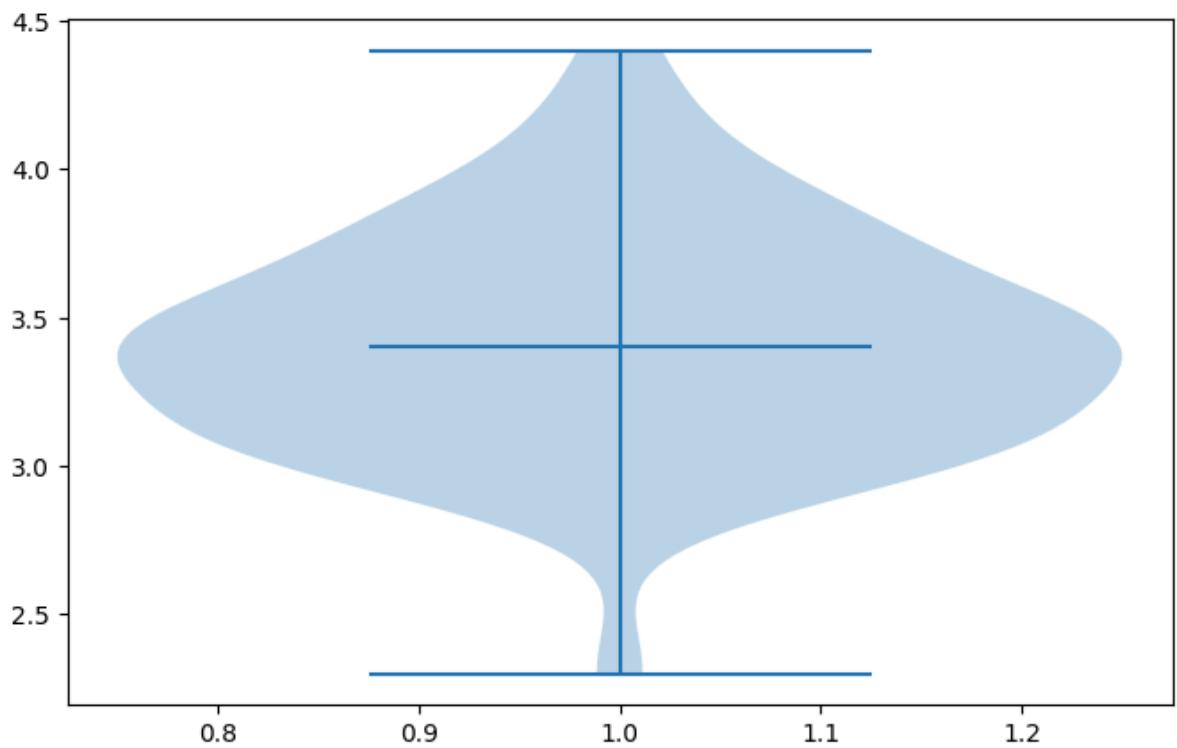
	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
In [ ]: plt.violinplot(df['PetalLength'][:50])
```

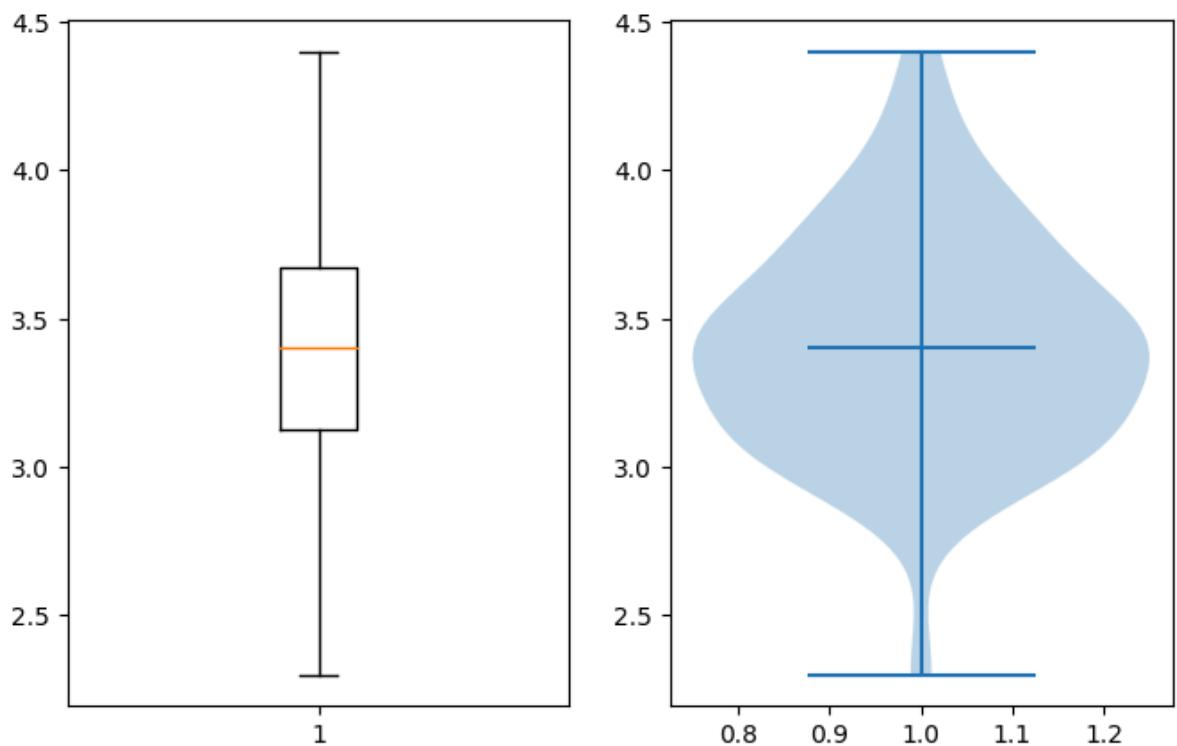
```
Out[ ]: {'bodies': [<matplotlib.collections.PolyCollection at 0x7a369fd94190>],  
'cmaxes': <matplotlib.collections.LineCollection at 0x7a369fd94160>,  
'cmins': <matplotlib.collections.LineCollection at 0x7a369fd947c0>,  
'cbars': <matplotlib.collections.LineCollection at 0x7a369fd94b20>}
```



```
In [ ]: vi = plt.violinplot(df['SepalWidth'][:50], showmedians=True)
```



```
In [ ]: plt.subplot(1,2,1)
bb = plt.boxplot(df['SepalWidth'][:50])
plt.subplot(1,2,2)
vi = plt.violinplot(df['SepalWidth'][:50], showmedians=True)
```



```
In [ ]:
```

จากไฟล์ example_timestamp.csv

เปรียบเทียบผลรวม `alpha` และ `beta` ของข้อมูล ก่อนวันที่ 2 มิถุนายน 2020 และตั้งแต่วันที่ 2 มิถุนายน 2020

```
In [ ]: import pandas as pd
```

```
In [ ]: data = pd.read_csv('/content/example_timestamp.csv')
```

```
In [ ]: data
```

```
Out[ ]: Unnamed: 0 alpha beta
```

0	2020-05-29	8.78	24
1	2020-05-30	13.00	25
2	2020-05-31	0.44	25
3	2020-06-01	1.94	28
4	2020-06-02	5.40	20
5	2020-06-03	5.68	21
6	2020-06-04	2.64	16

```
In [ ]: data.iloc[0,0]
```

```
Out[ ]: '2020-05-29'
```

```
In [ ]: from datetime import datetime
```

<https://docs.python.org/3/library/datetime.html#strftime-and-strptime-behavior>

```
In [ ]: data['Unnamed: 0'] = pd.to_datetime(data['Unnamed: 0'], format='%Y-%m-%d')  
data
```

```
Out[ ]: Unnamed: 0 alpha beta
```

0	2020-05-29	8.78	24
1	2020-05-30	13.00	25
2	2020-05-31	0.44	25
3	2020-06-01	1.94	28
4	2020-06-02	5.40	20
5	2020-06-03	5.68	21
6	2020-06-04	2.64	16

```
In [ ]: data.iloc[0,0]
```

```
Out[ ]: Timestamp('2020-05-29 00:00:00')
```

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

```
alpha      24.16
beta      102.00
dtype: float64
```

```
In [ ]: A['alpha']
```

```
Out[ ]: 24.160000000000004
```

```
In [ ]: data[data['Unnamed: 0']>=datetime(day=2,month=6,year=2020)].sum()
```

```
Out[ ]: alpha    13.72
beta     57.00
dtype: float64
```

```
In [ ]: print(f"ผลรวมของค่า alpha ก่อนวันที่ 2 มิ.ย. 2020 คือ {A['alpha']:.2f}")
```

ผลรวมของค่า alpha ก่อนวันที่ 2 มิ.ย. 2020 คือ 24.16

Query how?

```
In [ ]: data.query('alpha == 1.94')
```

```
Out[ ]: Unnamed: 0  alpha  beta
3    2020-06-01    1.94    28
```

```
In [ ]: data[data['alpha']==1.94]
```

```
Out[ ]: Unnamed: 0  alpha  beta
3    2020-06-01    1.94    28
```

จากไฟล์ search_clicks.csv

เปรียบเทียบเวลาเฉลี่ยที่ใช้ในการกดเลือกร้าน time_diff_nanos ระหว่างช่วงเวลา

10.00-10.59 กับ 23.00-23.59 ว่าช่วงเวลาไหนมี time_diff_nanos น้อยกว่ากัน

```
In [ ]:
```

บทที่ 7

(การแสดงผลการเปรียบเทียบข้อมูล)

บทที่ 7: การแสดงผลการเปรียบเทียบข้อมูล

1. หลักการ Projection.....	1
● การทำ PCA (Principal component Analysis): ขั้นตอนการทำ PCA เพื่อลด Dimensionality ของข้อมูลและความสำคัญของ Principal Components.....	2
● การวาดกราฟ PCA: การแสดงผลข้อมูลที่ถูกปรับมาจาก PCA ในรูปแบบกราฟ.....	2
2. การใช้ package sklearn ของภาษา python.....	2
● การนำเข้าและการใช้งาน: วิธีการนำเข้าและการใช้งาน package sklearn เพื่อการวิเคราะห์ข้อมูลและการประมวลผล	
3. การแสดงตัวอักษรภาษาไทยในกราฟ matplotlib: วิธีการแสดงตัวอักษรภาษาไทยจากข้อมูลที่มีชื่อและข้อมูลเป็นภาษาไทย.....	5
4. Bar Chart.....	4
● การสร้าง Bar Chart: วิธีการสร้าง Bar Chart เพื่อแสดงการเปรียบเทียบข้อมูลระหว่างหมวดหมู่ต่าง ๆ	
● การใช้งาน Parameter: tick_label: กำหนดชื่อแท่งแต่ละแท่งในกราฟตามลำดับ	
● การเพิ่มส่วนประกอบกราฟต่างๆ: กำหนดชื่อกราฟและชื่อแกน	
5. Grouped Bar Chart.....	6
● การเตรียมข้อมูลแต่ละกลุ่มเพื่อสร้าง Grouped Bar Chart	
● การสร้าง Grouped Bar Chart: ขั้นตอนวิธีการสร้าง Grouped Bar Chart เพื่อแสดงการเปรียบเทียบข้อมูลระหว่างหมวดหมู่ต่าง ๆ โดยแบ่งกลุ่มตามหมวดหมู่ย่อย	
6. Stacked Bar Chart.....	8
● การสร้าง Stacked Bar Chart: ขั้นตอนวิธีการสร้าง Stacked Bar Chart เพื่อแสดงการเปรียบเทียบข้อมูลระหว่างหมวดหมู่ต่าง ๆ โดยแสดงในรูปแบบของແຄาของกราฟ	
● การเพิ่มส่วนประกอบกราฟต่างๆ: กำหนดชื่อกราฟและชื่อแกน	
● การสร้างกราฟแท่งแนวนอน: โดยจะใช้คำสั่ง ax.barh() ในการสร้างกราฟแท่งแนวนอน	
7. Histogram Chart.....	9
● การเตรียมข้อมูลเพื่อสร้างกราฟ Histogram	
- การลับ record	

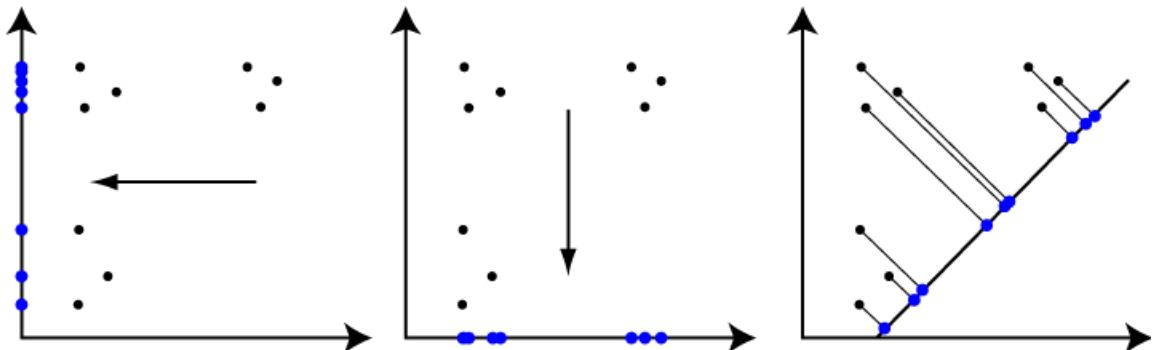
- การบังคับเปลี่ยน type ข้อมูลเป็น int
 - การลบ outliers
- การสร้าง Histogram Chart: ขั้นตอนวิธีการสร้าง Histogram Chart เพื่อแสดงการกระจายของข้อมูลต่อความถี่ของข้อมูลในช่วงที่กำหนด

8. Tree Map.....14

- การสร้าง Tree Map: วิธีการสร้าง Tree Map เพื่อแสดงการเปรียบเทียบข้อมูลในรูปแบบของโครงสร้างต้นไม้ที่มีขนาดและสีแสดงค่า
- การใช้งาน squarify: โดยใช้คำสั่ง squarify.plot()
 - การใช้งาน Parameter: value เพื่อแสดงค่าในพื้นที่แต่ละพื้นที่ในกราฟ tree map
 - การใช้งาน Parameter: norm_y เพื่อเปลี่ยนรูปแบบของการจัดเรียงพื้นที่กราฟ tree map ตามที่ต้องการเพื่อให้ดูกราฟง่ายขึ้น

เมื่อเรามีตัวแปรมากกว่า 2 ตัว เราสามารถใช้ PCA (Principle Component Analysis) ในการลดจำนวนตัวแปรลงได้

Projection



<https://wendynavarrete.com/principal-component-analysis-with-numpy/>

load data

```
In [ ]: import pandas as pd
```

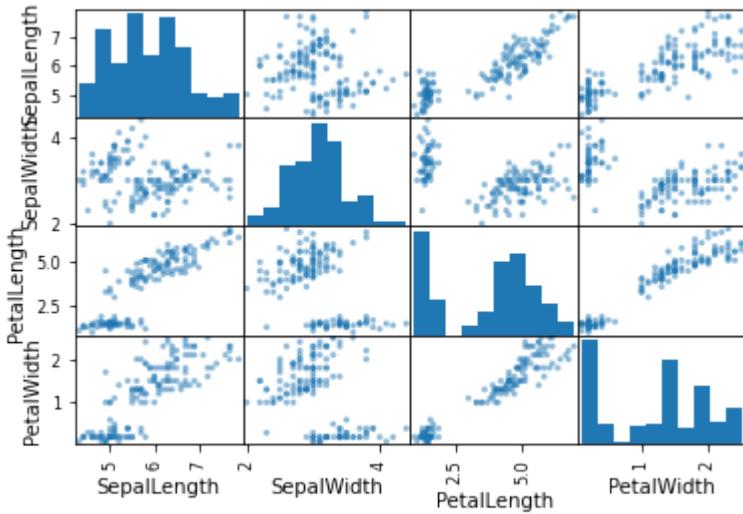
```
In [ ]: example_df = pd.read_csv('https://raw.github.com/pandas-dev/pandas/master/pandas/test/data/iris.csv')
example_df
```

```
Out[ ]:
```

	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

```
In [ ]: pd.plotting.scatter_matrix(example_df);
```



PCA

sklearn -> scikit-learn เป็น package ที่รวมรวม function การทำ Data Science - Machine Learning - Data Mining เอาไว้ให้เราใช้แบบไม่ต้องเขียนเอง

Import

```
In [ ]: from sklearn.decomposition import PCA
```

Define

```
In [ ]: pca = PCA()
```

Fit - Transform

```
In [ ]: example_df.iloc[:, :-1].shape
```

```
Out[ ]: (150, 4)
```

```
In [ ]: new_pca = pca.fit_transform(example_df.iloc[:, :-1]) ## record - แล้ว , dimension -
```

```
In [ ]: new_pca.shape
```

```
Out[ ]: (150, 4)
```

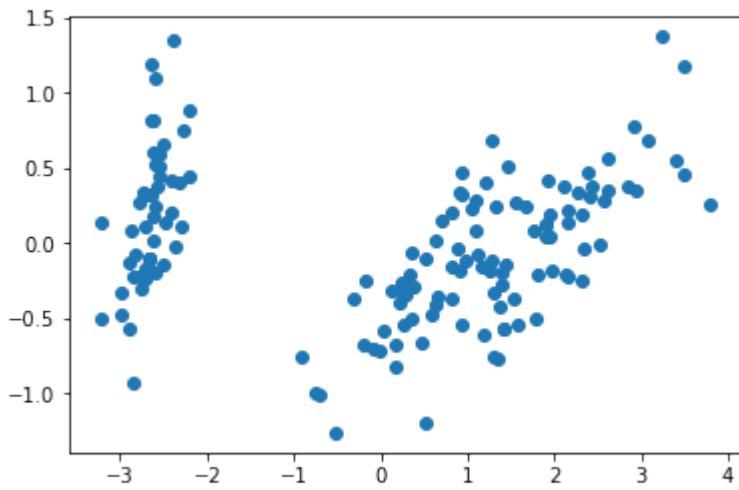
```
In [ ]: pca.explained_variance_ratio_
```

```
Out[ ]: array([0.92461621, 0.05301557, 0.01718514, 0.00518309])
```

```
In [ ]: from matplotlib import pyplot as plt
```

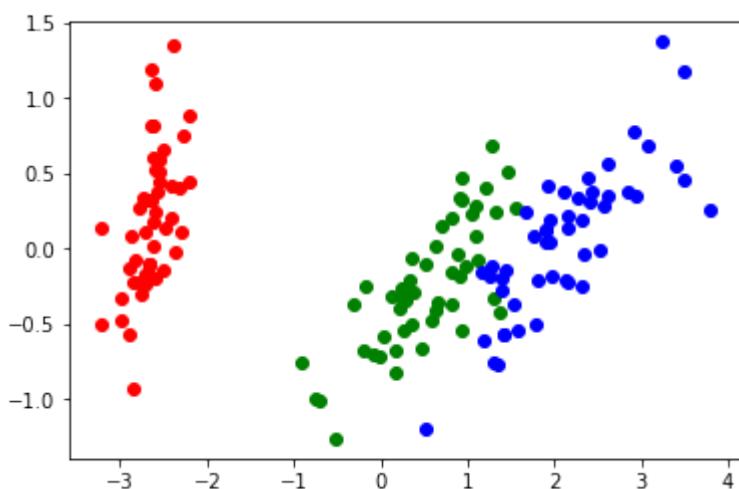
```
In [ ]: plt.scatter(new_pca[:, 0], new_pca[:, 1])
```

```
Out[ ]: <matplotlib.collections.PathCollection at 0x7f671c75fd90>
```



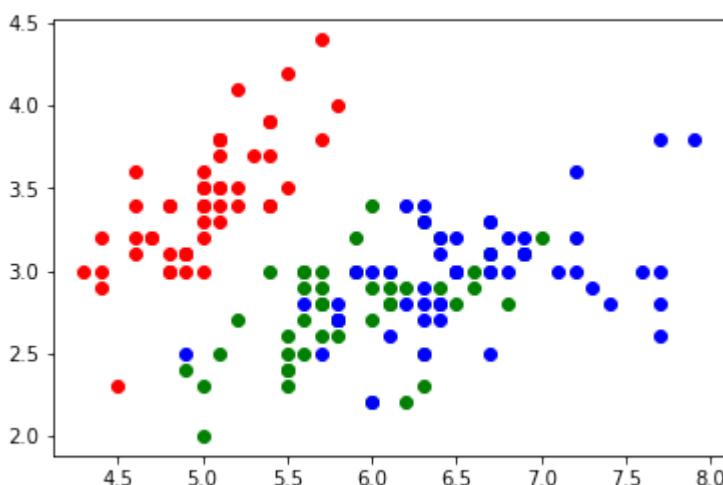
```
In [ ]: plt.plot(new_pca[:50,0],new_pca[:50,1], 'ro')
plt.plot(new_pca[50:100,0],new_pca[50:100,1], 'go')
plt.plot(new_pca[100:,0],new_pca[100:,1], 'bo')
```

```
Out[ ]: <matplotlib.lines.Line2D at 0x7f671c370350>
```



```
In [ ]: plt.plot(example_df.iloc[:50,0],example_df.iloc[:50,1], 'ro')
plt.plot(example_df.iloc[50:100,0],example_df.iloc[50:100,1], 'go')
plt.plot(example_df.iloc[100:,0],example_df.iloc[100:,1], 'bo')
```

```
Out[ ]: <matplotlib.lines.Line2D at 0x7f671c18ab50>
```



```
In [ ]:
```

สอน 1 เมษา 2564

```
In [ ]: import pandas as pd
import os
from datetime import datetime as dt
from datetime import time
```

```
In [ ]: from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
In [ ]: path = '/content/drive/My Drive/dataviz_2021_data'
```

```
In [ ]: data = pd.read_csv(os.path.join(path, 'search_request.csv'))
data.head()
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2718: DtypeWarning: Columns (7,8,9) have mixed types.Specify dtype option on import or set low_memory=False.
    interactivity=interactivity, compiler=compiler, result=result)
```

```
Out[ ]:
```

	Unnamed: 0	search_id	search_timestamp	user_agent	q	user_id	s
0	0	683de889-f923-494e-9d46-44a3d67b7259	2018-06-14 12:34:35.449	Wongnai/8.17.3 rv:8.17.3.3921 (iPhone5,4; iOS; ...)	NaN	NaN	5lqjjikta19d296mo7...
1	1	4a811230-ffa4-4631-a4c8-5d0394137d02	2018-06-14 17:11:19.469	Mozilla/5.0 (iPhone; CPU iPhone OS 11_4 like Mac OS X)	NaN	NaN	1r3iotmp0o9slom9...
2	2	7ad6ee8e-438e-4bea-9183-74dcf9e358e	2018-06-14 13:22:31.736	Mozilla/5.0 (Linux; Android 7.0; SAMSUNG SM-J7...)	NaN	NaN	5ci1eo4v5u9dha4ppgi...
3	3	0c17a5f5-fa89-40f4-ae94-a8659268f827	2018-06-02 12:37:27.331	Mozilla/5.0 (Linux; Android 7.1.1; SM-N950F Bu...)	NaN	NaN	39n535qgje9kpojp0c...
4	4	6870dc3a-5602-44fc-80ed-df0a7783df9d	2018-06-02 11:19:22.404	Mozilla/5.0 (iPhone; CPU iPhone OS 11_3_1 like Mac OS X)	NaN	NaN	5pa03h6lj691to60e...

เตรียมข้อมูล

แปลงข้อมูลของเวลาให้เป็นตัวแปรชนิด datetime

```
In [ ]: data['search_timestamp'] = pd.to_datetime(data['search_timestamp'], format='%Y-%m-%d %H:%M:%S')
```

Bar chart (กราฟแท่ง)

(กราฟผลไม้)

สร้างกราฟแท่งเปรียบเทียบปริมาณ คนเข้าใช้ web Wongnai.com เพื่อค้นหาร้านอาหาร ในแต่ละวัน

quiz 6

```
In [ ]: data[data['search_timestamp'].dt.dayofweek == 0].shape[0]
```

```
Out[ ]: 1076297
```

```
In [ ]: from matplotlib import pyplot as plt
```

ส่วนประกอบของกราฟแท่ง

- ตัวกราฟแท่ง (height)
- ตำแหน่งกราฟแท่ง (x)
- ชื่อแท่ง (tick_label)
- ชื่อกราฟ (plt.title)
- ชื่อแกน x (plt.xlabel)
- ชื่อแกน y (plt.ylabel)

```
In [ ]: import matplotlib  
matplotlib.__version__
```

```
Out[ ]: '3.2.2'
```

การแสดงตัวอักษรภาษาไทยในกราฟ matplotlib

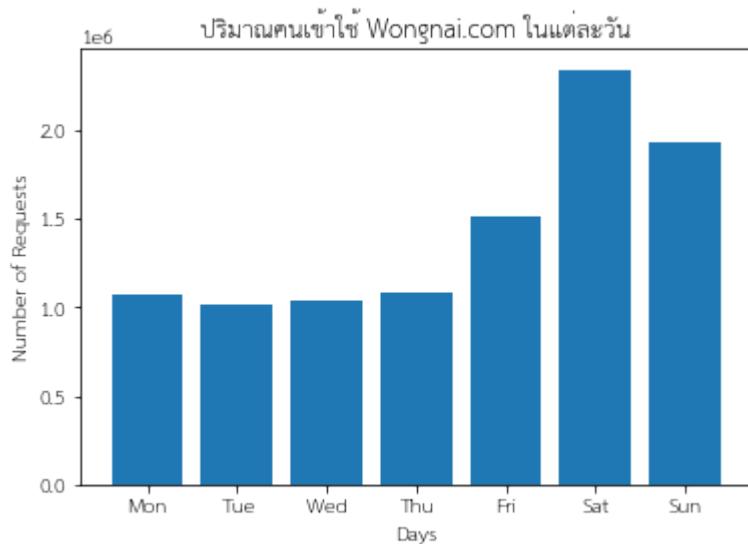
<https://medium.com/@kanyawee.work/%E0%B9%81%E0%B8%AA%E0%B8%94%E0%B8%87%E0%matplotlib-%E0%B8%9A%E0%B8%99-google-colab-37210d9a9f31>

https://colab.research.google.com/drive/1sTdTZx_Cm51mc8OL_QHtehWyO4725sGI#scrollTo=Ak

```
In [ ]: !wget -q https://github.com/Phonbopit/sarabun-webfont/raw/master/fonts/thsarabunnew.ttf
```

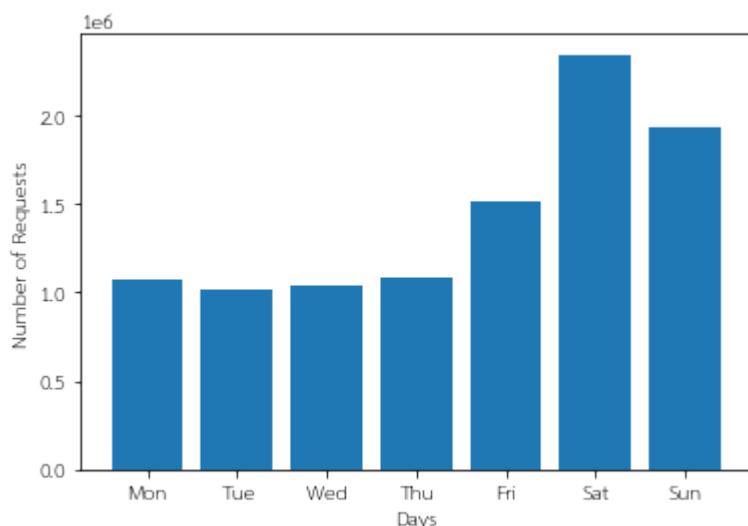
```
In [ ]: import matplotlib as mpl  
mpl.font_manager.FontManager.addfont('thsarabunnew-webfont.ttf')  
mpl.rc('font', family='TH Sarabun New')
```

```
In [ ]: plt.bar([1,2,3,4,5,6,7],[data[data['search_timestamp'].dt.dayofweek == 0].shape[0],  
data[data['search_timestamp'].dt.dayofweek == 1].shape[0],  
data[data['search_timestamp'].dt.dayofweek == 2].shape[0],  
data[data['search_timestamp'].dt.dayofweek == 3].shape[0],  
data[data['search_timestamp'].dt.dayofweek == 4].shape[0],  
data[data['search_timestamp'].dt.dayofweek == 5].shape[0],  
data[data['search_timestamp'].dt.dayofweek == 6].shape[0]  
],tick_label=['Mon','Tue','Wed','Thu','Fri','Sat','Sun'])  
plt.xlabel('Days')  
plt.ylabel('Number of Requests')  
plt.title('ปริมาณคนเข้าใช้ Wongnai.com ในแต่ละวัน');
```



```
In [ ]: plt.bar([1,2,3,4,5,6,7],[data[data['search_timestamp'].dt.dayofweek == 0].shape[0],
                                data[data['search_timestamp'].dt.dayofweek == 1].shape[0],
                                data[data['search_timestamp'].dt.dayofweek == 2].shape[0],
                                data[data['search_timestamp'].dt.dayofweek == 3].shape[0],
                                data[data['search_timestamp'].dt.dayofweek == 4].shape[0],
                                data[data['search_timestamp'].dt.dayofweek == 5].shape[0],
                                data[data['search_timestamp'].dt.dayofweek == 6].shape[0]
                               ],tick_label=['Mon','Tue','Wed','Thu','Fri','Sat','Sun'])
plt.xlabel('Days')
plt.ylabel('Number of Requests')
```

Out[]: Text(0, 0.5, 'Number of Requests')



Grouped bar chart

https://matplotlib.org/stable/gallery/lines_bars_and_markers/barchart.html#sphx-glr-gallery-lines-bars-and-markers-barchart-py

แสดงปริมาณคนเข้าเว็บในแต่ละวัน โดยเปรียบเทียบช่วงเวลา 11:00-12:00 กับ 23:00-24:00

```
In [ ]: data[(data['search_timestamp'].dt.dayofweek == 0) & (data['search_timestamp'].dt.time < time(hour=12)) & (data['search_timestamp'].dt.time >= time(hour=11))].shape[0] # monday 11:00-12:
```

Out[]: 73249

```
In [ ]: b4lunch = [data[(data['search_timestamp'].dt.dayofweek == 0)&(data['search_timestamp'].dt.dayofweek == 1)&(data['search_timestamp'].dt.dayofweek == 2)&(data['search_timestamp'].dt.dayofweek == 3)&(data['search_timestamp'].dt.dayofweek == 4)&(data['search_timestamp'].dt.dayofweek == 5)&(data['search_timestamp'].dt.dayofweek == 6)&(data['search_timestamp'].dt.dayofweek == 7)]
```

```
In [ ]: b4lunch
```

```
Out[ ]: [73249, 73083, 75429, 78024, 99007, 174165, 165440]
```

```
In [ ]: data[(data['search_timestamp'].dt.dayofweek == 0)&(data['search_timestamp'].dt.time
```

```
Out[ ]: 31874
```

```
In [ ]: b4midnight = [data[(data['search_timestamp'].dt.dayofweek == 0)&(data['search_timestamp'].dt.dayofweek == 1)&(data['search_timestamp'].dt.dayofweek == 2)&(data['search_timestamp'].dt.dayofweek == 3)&(data['search_timestamp'].dt.dayofweek == 4)&(data['search_timestamp'].dt.dayofweek == 5)&(data['search_timestamp'].dt.dayofweek == 6)&(data['search_timestamp'].dt.dayofweek == 7)]
```

```
b4midnight
```

```
Out[ ]: [31874, 32258, 31153, 35944, 53174, 58306, 35801]
```

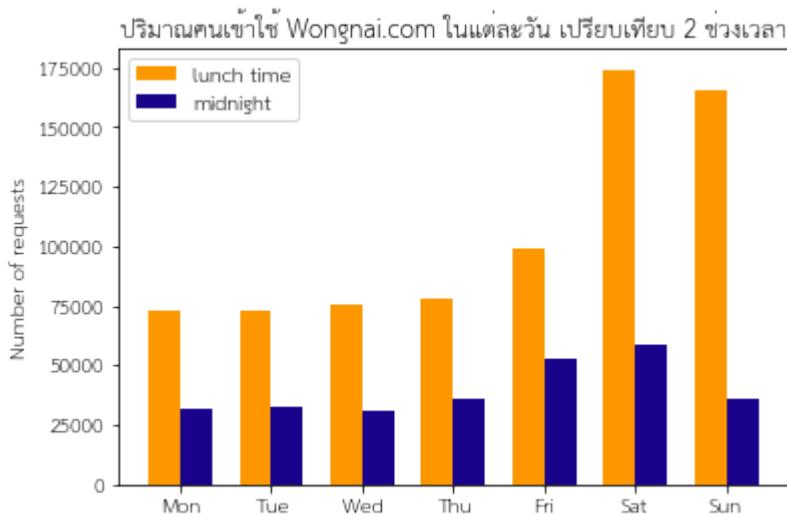
```
In [ ]: labels = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
```

```
In [ ]: import numpy as np
```

```
In [ ]: x = np.arange(len(labels)) # the label locations
width = 0.35 # the width of the bars

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

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Number of requests')
ax.set_title('ปริมาณคนเข้าใช้ Wongnai.com ในแต่ละวัน เปรียบเทียบ 2 ช่วงเวลา')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend();
```



Stacked bar chart

```
In [ ]: import matplotlib.pyplot as plt

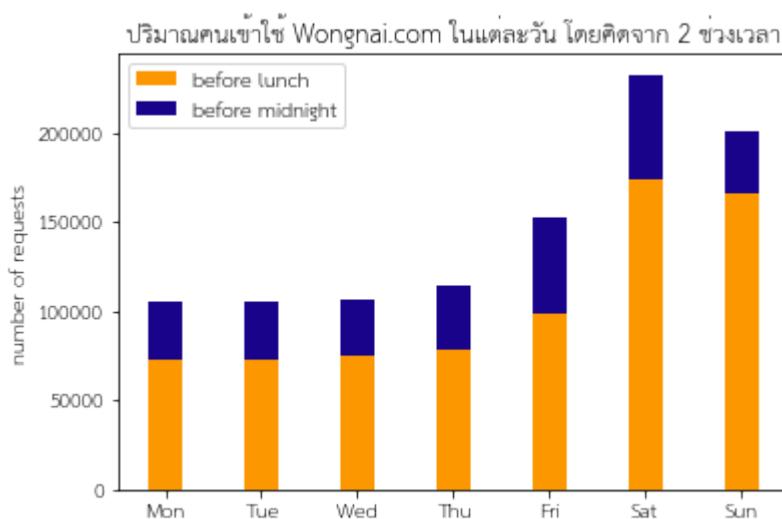
width = 0.35      # the width of the bars: can also be len(x) sequence

fig, ax = plt.subplots()

ax.bar(labels, b4lunch, width, label='before lunch', color = '#fc9700')
ax.bar(labels, b4midnight, width, bottom=b4lunch, label='before midnight', color = '#1f77b4')

ax.set_ylabel('number of requests')
ax.set_title('จำนวนคนเข้าใช้ Wongnai.com ในแต่ละวัน โดยคิดจาก 2 ช่วงเวลา')
ax.legend()

plt.show()
```



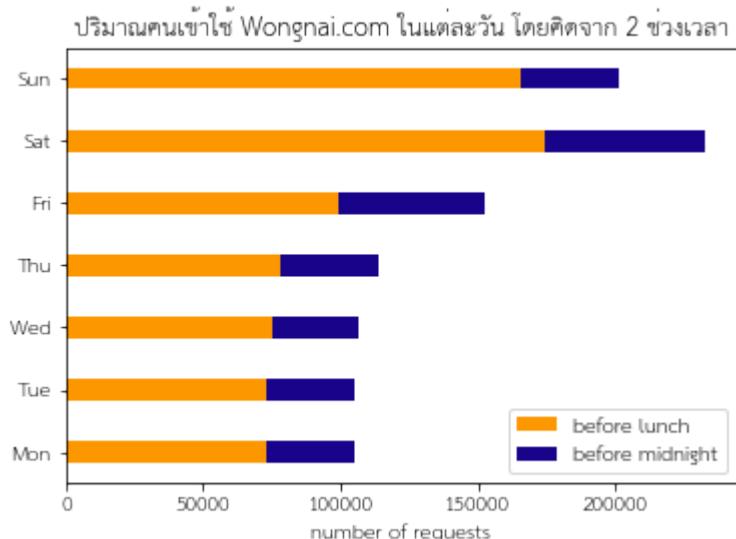
```
In [ ]: width = 0.35      # the width of the bars: can also be len(x) sequence

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 = '#1f77b4')

ax.set_xlabel('number of requests')
ax.set_title('จำนวนคนเข้าใช้ Wongnai.com ในแต่ละวัน โดยคิดจาก 2 ช่วงเวลา')
ax.legend()
```

```
plt.show()
```



[เช็คชื่อ] โดยให้วาด Bar chart ที่เปรียบเทียบปริมาณคนใช้งาน Wongnai.com ส่องช่วงเวลา โดยให้กราฟแสดงสัดส่วนของปริมาณคนใช้งานในแต่ละวันด้วย

In []:

Histogram

กราฟแสดงความถี่ของข้อมูล

ตัวอย่างข้อมูลที่ random มาจาก normal distribution ที่มี mean = 100 และ stdev = 15

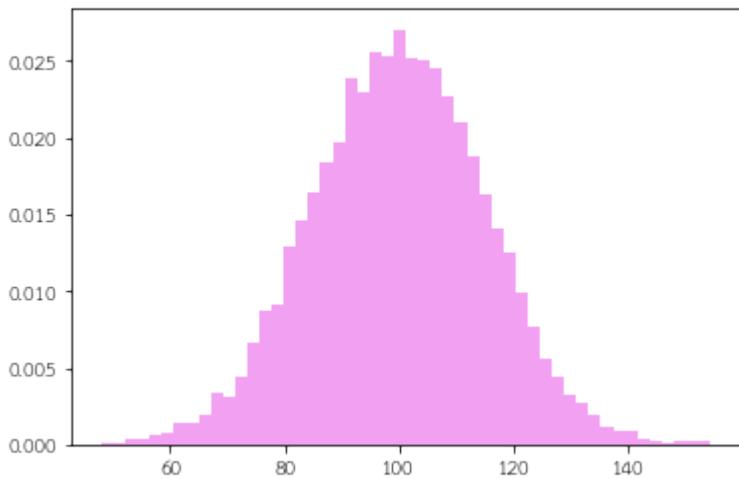
In []:

```
import numpy as np
from matplotlib import pyplot as plt

np.random.seed(2021)

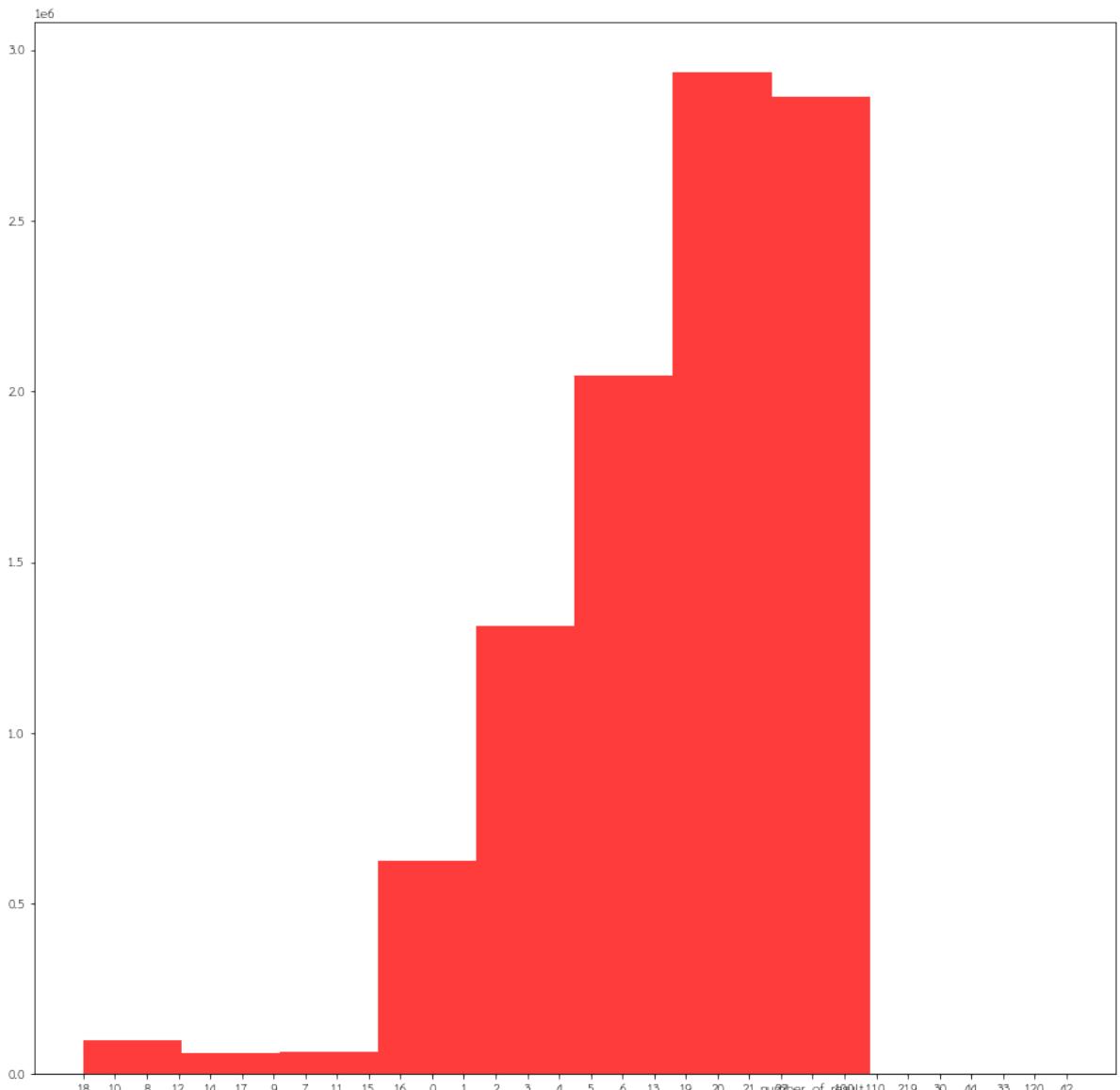
mu, sigma = 100, 15
X = mu + sigma * np.random.randn(10000)

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



ตัวอย่างข้อมูล wongnai.com

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



แก้ไข แกน x ที่เรียงข้อมูลผิด

```
In [ ]: data.dtypes
```

```
Out[ ]: Unnamed: 0           int64
         search_id            object
         search_timestamp    datetime64[ns]
         user_agent           object
         q                     object
         user_id              float64
         session_id           object
         number_of_result     object
         lat                  object
         long                 object
         dtype: object
```

เรียกดู data type ของ ตัวแปร

```
In [ ]: type(data['number_of_result'][0])
```

```
Out[ ]: int
```

ตรวจสอบ data type ของตัวแปร

```
In [ ]: type(data['number_of_result'][0]) == int
```

```
Out[ ]: True
```

ตรวจสอบดูทุกค่าใน column 'number_of_result'

```
In [ ]: # for x in data['number_of_result']:
#       if type(x) != int:
#           print(f'{x} -> {type(x)})
```

```
In [ ]: new_type = data['number_of_result'].astype('int32')
```

```

-----
ValueError                                     Traceback (most recent call last)
<ipython-input-34-29dba17f7bb1> in <module>()
----> 1 new_type = data['number_of_result'].astype('int32')

/usr/local/lib/python3.7/dist-packages/pandas/core/generic.py in astype(self, dtype, copy, errors)
    5546         else:
    5547             # else, only a single dtype is given
-> 5548             new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
    5549             return self._constructor(new_data).__finalize__(self, method
=="astype")
    5550

/usr/local/lib/python3.7/dist-packages/pandas/core/internals/managers.py in astype(self, dtype, copy, errors)
    602             self, dtype, copy: bool = False, errors: str = "raise"
    603         ) -> "BlockManager":
--> 604             return self.apply("astype", dtype=dtype, copy=copy, errors=errors)
    605
    606     def convert(

/usr/local/lib/python3.7/dist-packages/pandas/core/internals/managers.py in apply(self, f, align_keys, **kwargs)
    407             applied = b.apply(f, **kwargs)
    408         else:
--> 409             applied = getattr(b, f)(**kwargs)
    410             result_blocks = _extend_blocks(applied, result_blocks)
    411

/usr/local/lib/python3.7/dist-packages/pandas/core/internals/blocks.py in astype(self, dtype, copy, errors)
    593             vals1d = values.ravel()
    594             try:
--> 595                 values = astype_nansafe(vals1d, dtype, copy=True)
    596             except (ValueError, TypeError):
    597                 # e.g. astype_nansafe can fail on object-dtype of strings

/usr/local/lib/python3.7/dist-packages/pandas/core/dtypes/cast.py in astype_nansafe(arr, dtype, copy, skipna)
    972             # work around NumPy brokenness, #1987
    973             if np.issubdtype(dtype.type, np.integer):
--> 974                 return lib.astype_intsafe(arr.ravel(), dtype).reshape(arr.shape)
    975
    976             # if we have a datetime/timedelta array of objects

pandas/_libs/lib.pyx in pandas._libs.lib.astype_intsafe()

ValueError: invalid literal for int() with base 10: 'number_of_result'

```

ລວມ record ທີ່ມີຄ່າ ໃນ column 'number_of_result' ເປັນ number of result

In []: `data[data['number_of_result']=='number_of_result']`

Out[]:

	Unnamed: 0	search_id	search_timestamp	user_agent	q	user_id	session_id	nur
1000016	1000032	search_id	NaT	user_agent	original_q	228667.0	session_id	nu

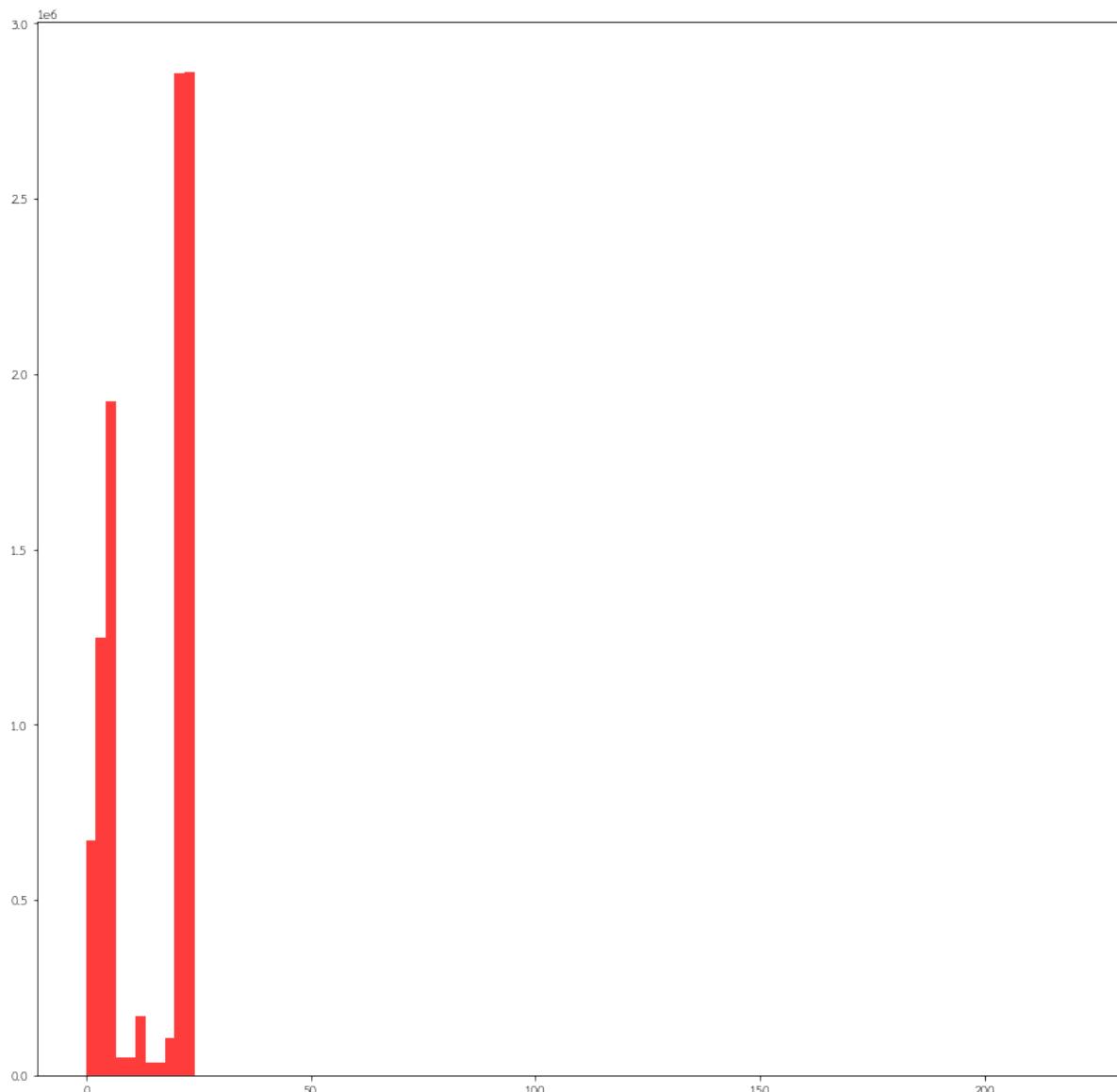
```
In [ ]: data = data.drop(1000016)
```

```
In [ ]: data[data['number_of_result']=='number_of_result']
```

```
Out[ ]: Unnamed:  
0  search_id  search_timestamp  user_agent  q  user_id  session_id  number_of_result  la
```

```
In [ ]: new_type = data['number_of_result'].astype('int32')
```

```
In [ ]: output = plt.hist(new_type,100,facecolor = 'red' ,alpha = 0.75)
```



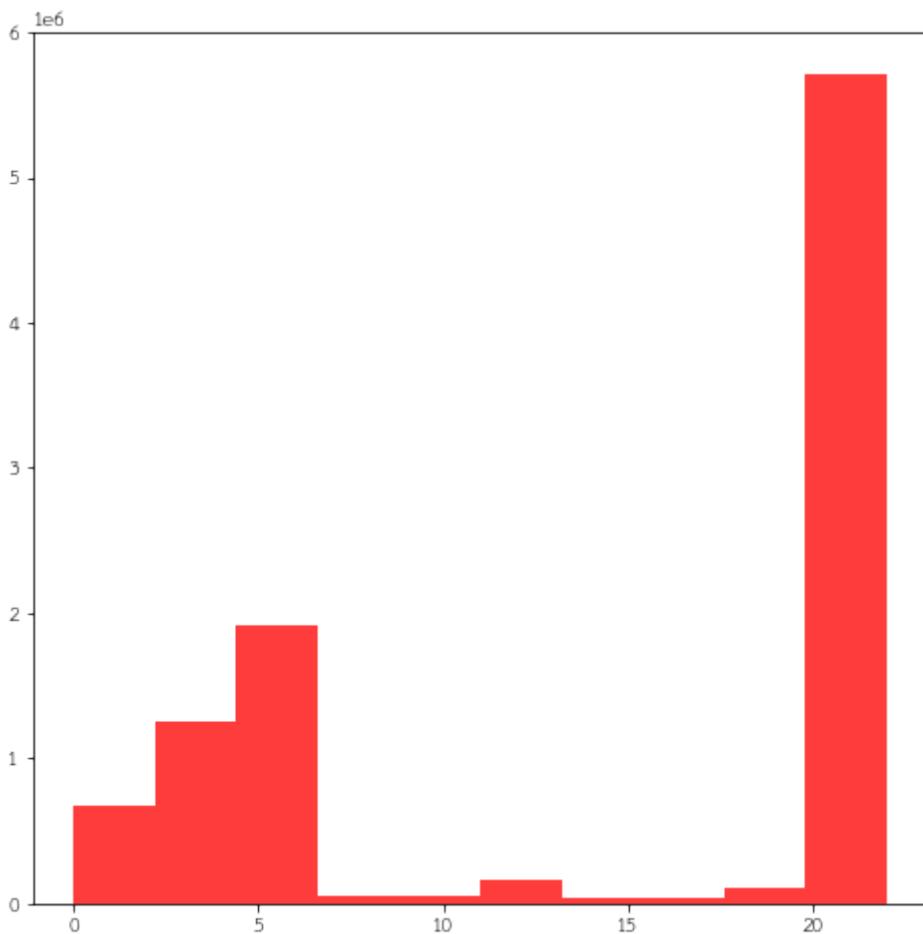
ลบ outlier

```
In [ ]: new_type_nooutlier = new_type[new_type < 25]
```

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

```
Out[ ]: 14
```

```
In [ ]: matplotlib.rcParams['figure.figsize']=[8,8]  
output = plt.hist(new_type_nooutlier,10,facecolor = 'red' ,alpha = 0.75)
```



Quiz 7 เปรียบเทียบความถี่ของแท่งที่มีค่ามากที่สุด กับ แท่งอื่นๆรวมกัน

In []: `output`

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

In []: `output[0]`

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

In []: `output[0][-1]`

Out[]: `5717238.0`

In []: `sum(output[0][:-1])`

Out[]: `4287104.0`

Tree map

In []: `!pip install squarify`

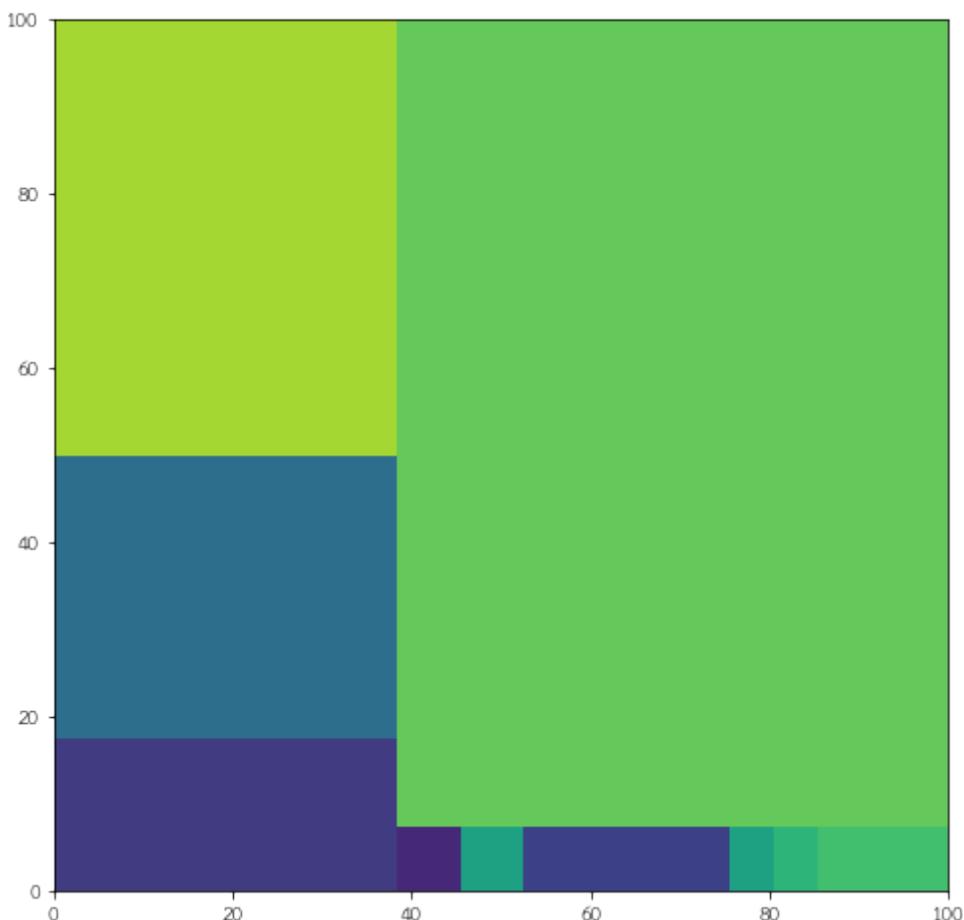
Requirement already satisfied: squarify in /usr/local/lib/python3.7/dist-packages
(0.4.3)

In []: `import numpy as np
import matplotlib.pyplot as plt`

```
import squarify
```

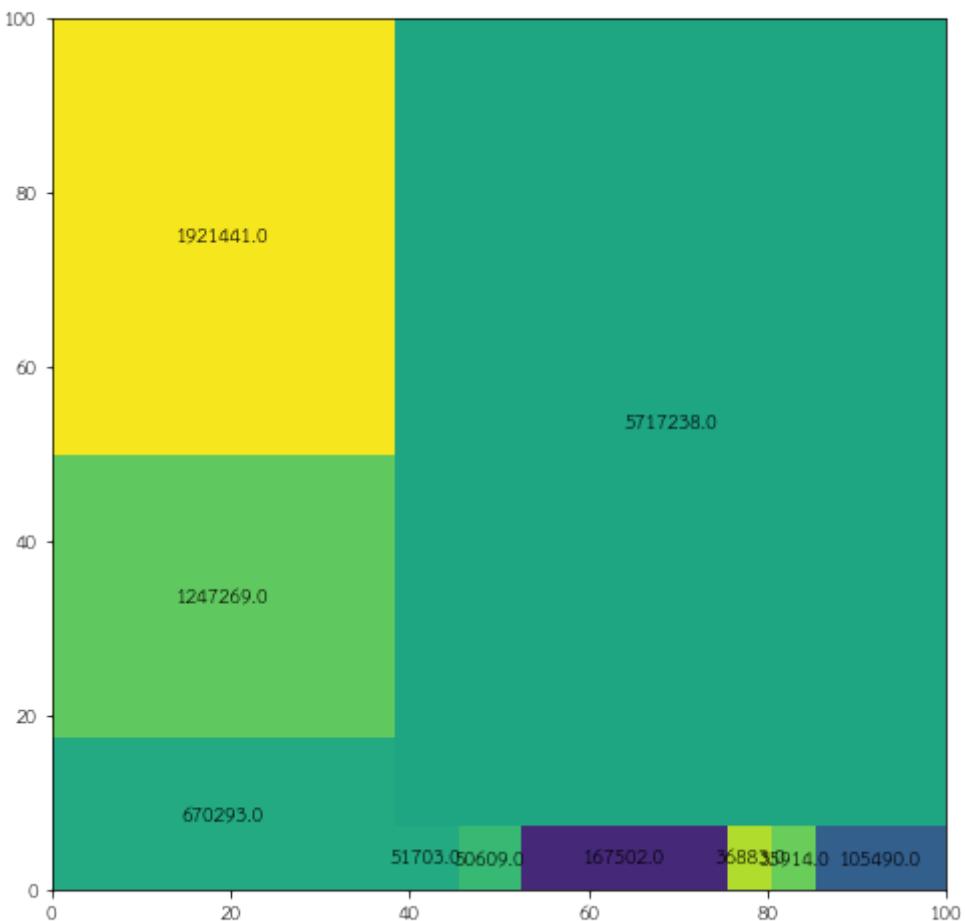
```
In [ ]: squarify.plot(output[0])
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f287b486950>
```



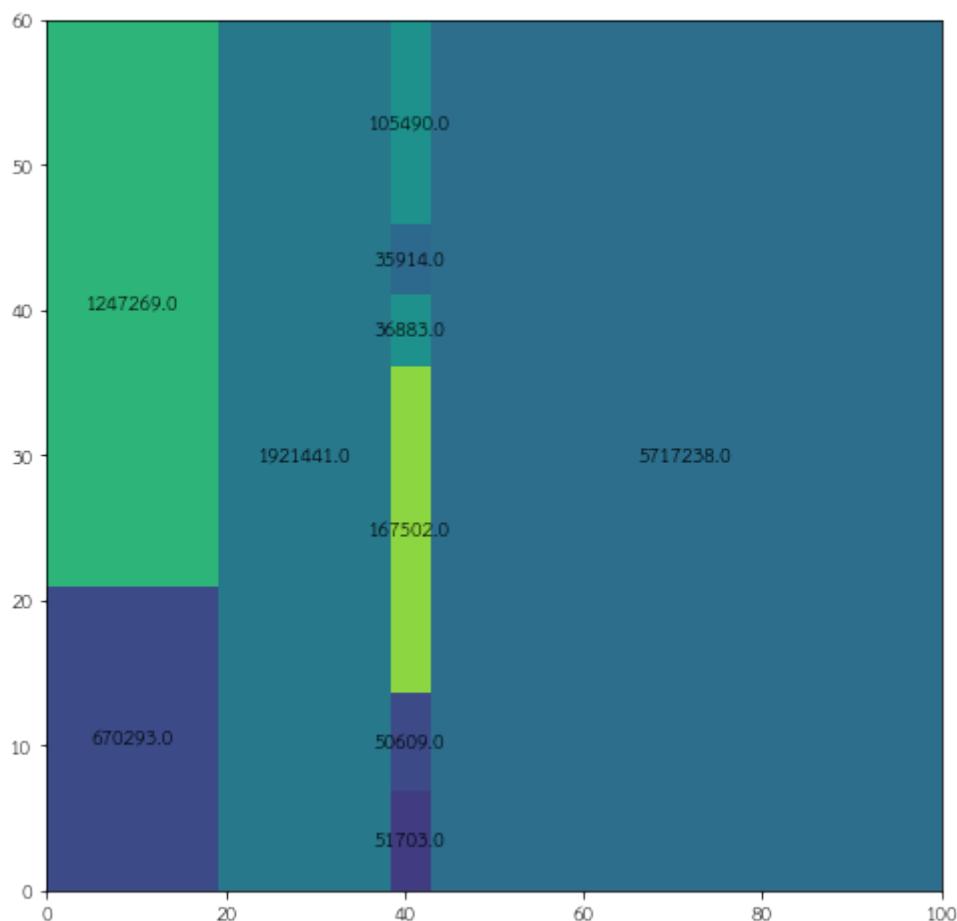
```
In [ ]: squarify.plot(output[0], value=output[0])
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f286d23f0d0>
```



```
In [ ]: squarify.plot(output[0], value=output[0], norm_y=60)
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28680ac210>
```



บทที่ 8
(การแสดงผลข้อมูลเชิงเวลาและอื่นๆ)

บทที่ 8: การแสดงผลข้อมูลเชิงเวลาและอื่นๆ

1. แนวคิดและการสร้างแผนภูมิอิบายข้อมูลเชิงเวลา
 - แนวคิดเกี่ยวกับข้อมูลเชิงเวลา: ข้อมูลเชิงเวลาหมายถึงข้อมูลที่มีการบันทึกตามลำดับเวลาหรือช่วงเวลา เช่น ข้อมูลยอดขายรายวัน อุณหภูมิรายชั่วโมง เป็นต้น
 - ความสำคัญของการวิเคราะห์ข้อมูลเชิงเวลา: การวิเคราะห์ข้อมูลเชิงเวลาเพื่อเข้าใจแนวโน้ม รูปแบบตามฤดูกาล และการเปลี่ยนแปลงของข้อมูลตามช่วงเวลา
 - การเตรียมข้อมูลเชิงเวลา: การรวบรวมและทำความสะอาดข้อมูลเชิงเวลา เพื่อให้ข้อมูลมีความถูกต้องและพร้อมใช้งานในการวิเคราะห์
 - การสร้างแผนภูมิข้อมูลเชิงเวลา: การใช้เครื่องมือและไลบรารีใน Python เช่น Matplotlib, Pandas, และ Seaborn เพื่อสร้างแผนภูมิที่แสดงข้อมูลเชิงเวลา
2. Timeline Chart.....1
 - แนวคิดของ Timeline Chart: การแสดงเหตุการณ์หรือข้อมูลตามลำดับเวลาบนแกนเวลา
 - การสร้าง Timeline Chart ด้วย Matplotlib: ขั้นตอนการสร้าง Timeline Chart ใน Matplotlib
 - การปรับแต่ง Timeline Chart: การปรับแต่งรูปแบบของกราฟ เช่น สี ขนาด และการจัดเรียงข้อมูล
3. Candlestick Chart.....2
 - แนวคิดของ Candlestick Chart: แผนภูมิแท่งเทียน (Candlestick Chart) ใช้สำหรับแสดงข้อมูลราคาหุ้นหรือสินทรัพย์อื่น ๆ ในตลาดการเงิน โดยแสดงราคากลางสุด ต่ำสุด ราคาเปิด และราคาปิดในช่วงเวลาที่กำหนด
 - การสร้าง Candlestick Chart ด้วย Matplotlib: การใช้ไลบรารี plotly.graph_objects เพื่อสร้าง Candlestick Chart
 - การปรับแต่ง Candlestick Chart: การปรับแต่งรูปแบบของกราฟ เช่น สี ขนาด และการจัดเรียงข้อมูล
 - การวิเคราะห์ Candlestick Chart: วิธีการอ่านและวิเคราะห์รูปแบบต่าง ๆ ใน Candlestick Chart
4. Cross Spectral Density (CSD).....5

- แนวคิดของ Cross Spectral Density (CSD): CSD ใช้ในการวิเคราะห์ความสัมพันธ์ระหว่างสองสัญญาณในโดเมนความถี่
- การคำนวณ CSD: การใช้ไลบรารี SciPy เพื่อคำนวณ CSD จากสัญญาณสองชุด
- การปรับแต่ง CSD Chart: การปรับแต่งรูปแบบของกราฟ เช่น สี ขนาด และการจัดเรียงข้อมูล
- การแสดงผล CSD: การสร้างกราฟเพื่อแสดงผล CSD และการวิเคราะห์ความสัมพันธ์ระหว่างสัญญาณในโดเมนความถี่

บทที่ 8 การแสดงผลข้อมูลเชิงเวลาและอื่นๆ

- แนวคิดและการสร้างแผนภูมิอธิบายข้อมูลเชิงเวลา เช่น Timeline Chart, Candlestick Chart, Cross spectral density (CSD) เป็นต้น

Timeline Chart

```
In [ ]: #import
from datetime import datetime

import matplotlib.pyplot as plt
import numpy as np

import matplotlib.dates as mdates
```

```
In [ ]: # กำหนดข้อมูลใน List เก็บไว้ในตัวแปร
names = ['v2.2.4', 'v3.0.3', 'v3.0.2', 'v3.0.1', 'v3.0.0', 'v2.2.3',
         'v2.2.2', 'v2.2.1', 'v2.2.0', 'v2.1.2', 'v2.1.1', 'v2.1.0',
         'v2.0.2', 'v2.0.1', 'v2.0.0', 'v1.5.3', 'v1.5.2', 'v1.5.1',
         'v1.5.0', 'v1.4.3', 'v1.4.2', 'v1.4.1', 'v1.4.0']

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

# Convert date strings (e.g. 2014-10-18) to datetime
dates = [datetime.strptime(d, "%Y-%m-%d") for d in dates]
```

```
In [ ]: # Choose some nice levels
levels = np.tile([-5, 5, -3, 3, -1, 1],
                  int(np.ceil(len(dates)/6)))[len(dates):]

# Create figure and plot a stem plot with the date
fig, ax = plt.subplots(figsize=(8.8, 4), layout="constrained")
ax.set(title="Matplotlib release dates")

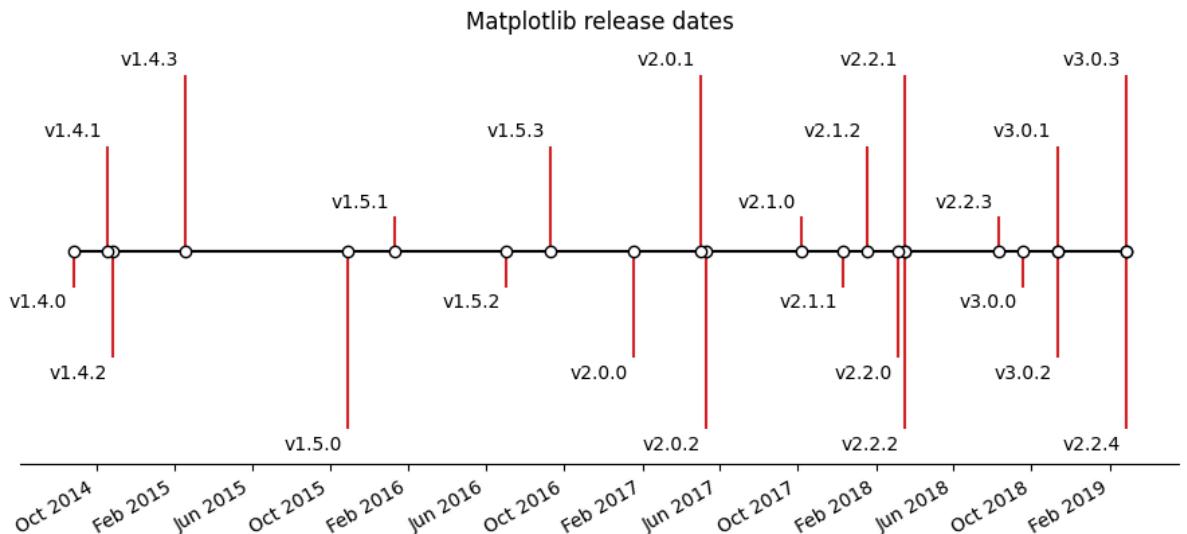
ax.vlines(dates, 0, levels, color="tab:red") # The vertical stems.
ax.plot(dates, np.zeros_like(dates), "-o",
        color="k", markerfacecolor="w") # Baseline and markers on it.

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

# format x-axis with 4-month intervals
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")

# remove y-axis and spines
ax.yaxis.set_visible(False)
```

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



Candlestick Chart

```
In [ ]: #import  
import plotly.graph_objects as go  
  
import pandas as pd  
from datetime import datetime
```

```
In [ ]: df = pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/finance-
```

```
In [ ]: df
```

Out[]:

	Date	AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AAPL.Adjusted
0	2015-02-17	127.489998	128.880005	126.919998	127.830002	63152400	122.905254 106.74
1	2015-02-18	127.629997	128.779999	127.449997	128.720001	44891700	123.760965 107.84
2	2015-02-19	128.479996	129.029999	128.330002	128.449997	37362400	123.501363 108.89
3	2015-02-20	128.619995	129.500000	128.050003	129.500000	48948400	124.510914 109.78
4	2015-02-23	130.020004	133.000000	129.660004	133.000000	70974100	127.876074 110.37
...
501	2017-02-10	132.460007	132.940002	132.050003	132.119995	20065500	132.119995 114.49
502	2017-02-13	133.080002	133.820007	132.750000	133.289993	23035400	133.289993 114.82
503	2017-02-14	133.470001	135.089996	133.250000	135.020004	32815500	135.020004 115.17
504	2017-02-15	135.520004	136.270004	134.619995	135.509995	35501600	135.509995 115.54
505	2017-02-16	135.669998	135.899994	134.839996	135.350006	22118000	135.350006 116.20

506 rows × 11 columns

```
In [ ]: fig = go.Figure(data=[go.Candlestick(x=df['Date'],
                                             open=df['AAPL.Open'],
                                             high=df['AAPL.High'],
                                             low=df['AAPL.Low'],
                                             close=df['AAPL.Close'])])
```

```
In [ ]: fig.show()
```



Candlestick without RangeSlider

```
In [ ]: fig = go.Figure(data=[go.Candlestick(x=df['Date'],
                                             open=df['AAPL.Open'], high=df['AAPL.High'],
                                             low=df['AAPL.Low'], close=df['AAPL.Close'])
                                             ])
```

```
In [ ]: fig.update_layout(xaxis_rangeslider_visible=False)
fig.show()
```



Custom Candlestick Colors

```
In [ ]: fig = go.Figure(data=[go.Candlestick(
    x=df['Date'],
    open=df['AAPL.Open'], high=df['AAPL.High'],
    low=df['AAPL.Low'], close=df['AAPL.Close'],
    increasing_line_color= 'cyan', decreasing_line_color= 'gray'
)])
```

```
In [ ]: fig.show()
```

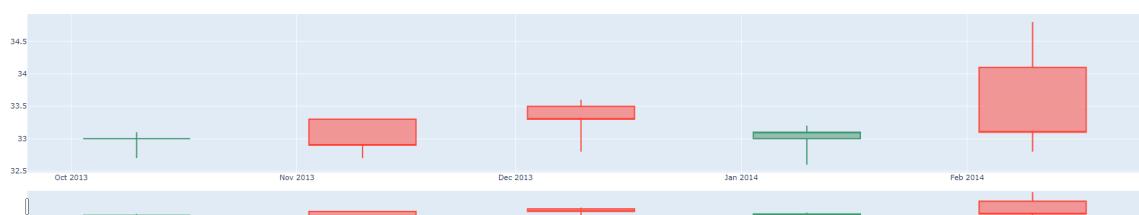


Simple Example with datetime Objects

```
In [ ]: open_data = [33.0, 33.3, 33.5, 33.0, 34.1]
high_data = [33.1, 33.3, 33.6, 33.2, 34.8]
low_data = [32.7, 32.7, 32.8, 32.6, 32.8]
close_data = [33.0, 32.9, 33.3, 33.1, 33.1]
dates = [datetime(year=2013, month=10, day=10),
         datetime(year=2013, month=11, day=10),
         datetime(year=2013, month=12, day=10),
         datetime(year=2014, month=1, day=10),
         datetime(year=2014, month=2, day=10)]
```

```
In [ ]: fig = go.Figure(data=[go.Candlestick(x=dates,
                                             open=open_data, high=high_data,
                                             low=low_data, close=close_data)])
```

```
In [ ]: fig.show()
```



Cross spectral density (CSD)

```
In [ ]: #import
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt
```

```
In [ ]: # Number of samples and time array creation
num_samples = 1000
time = np.linspace(0, 1, num_samples)

# Generating P wave, QRS complex, and T wave for ECG signal
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

# Generating brain signal (EEG-BETA) and adding random movement
brain_signal = np.cos(2 * np.pi * 20 * time)
random_movement = np.random.normal(loc=0, scale=0.1, size=num_samples)
brain_signal += random_movement

# Calculating Cross Spectral Density (CSD) between ECG and EEG signals
frequencies, csd = signal.csd(ecg_signal, brain_signal, fs=1.0, nperseg=100)
```

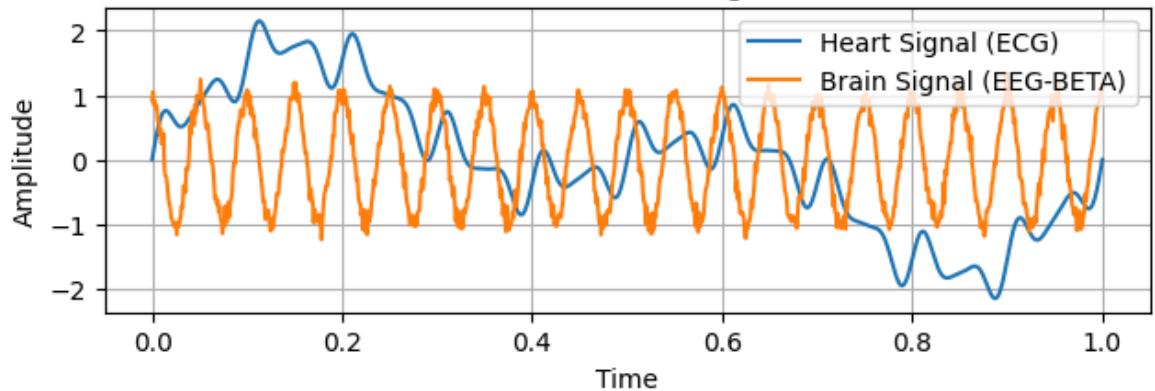
```
In [ ]: # Plotting ECG, EEG-BETA signal and Cross Spectral Density (CSD)
fig, (ax1, ax2) = plt.subplots(2, 1, layout='constrained')

# Plotting ECG, EEG-BETA signal
ax1.plot(time, ecg_signal, label='Heart Signal (ECG)')
ax1.plot(time, brain_signal, label='Brain Signal (EEG-BETA)')
ax1.set_xlabel('Time')
ax1.set_ylabel('Amplitude')
ax1.set_title('Heart and Brain Signals')
ax1.legend()
ax1.grid(True)

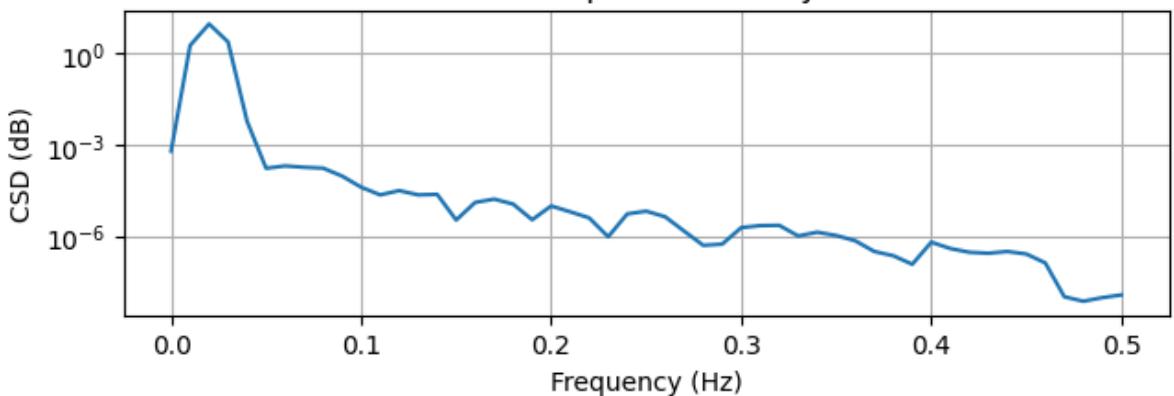
# Plotting Cross Spectral Density (CSD)
ax2.semilogy(frequencies, np.abs(csd))
ax2.set_xlabel('Frequency (Hz)')
ax2.set_ylabel('CSD (dB)')
ax2.set_title('Cross Spectral Density')
ax2.grid(True)

plt.show()
```

Heart and Brain Signals



Cross Spectral Density



In []:

บทที่ 9

(Hand-on การสร้าง Dashboard ด้วย Google Looker Studio)

บทที่ 9: Hand-on การสร้าง Dashboard ด้วย Google Looker Studio

1. การทำความสะอาดข้อมูล example data

- การเตรียมข้อมูล: การเตรียมข้อมูล example data สำหรับการสร้าง Dashboard ใน Looker Studio
- การทำความสะอาดข้อมูล: การทำความสะอาดข้อมูล Example Data เพื่อให้ข้อมูลเป็นไปตามรูปแบบที่ต้องการ
- การทำสำเนาไฟล์ Google Sheets ไว้ที่ google drive ตัวเอง: การเตรียมไฟล์ก่อนนำเข้าไปสร้าง Dashboard ใน Looker Studio

2. การสร้าง Dashboard ด้วย Looker Studio

- การใช้งาน Looker Studio: การเรียนรู้เกี่ยวกับอินเทอร์เฟซและเครื่องมือใน Looker Studio
- การเข้มต่อข้อมูล: การเข้มต่อข้อมูลจาก Google Sheets ใน Google Drive เข้ากับ Looker Studio เพื่อใช้ในการสร้าง Dashboard
- การสร้าง Dashboard: ขั้นตอนการสร้างและปรับแต่ง Dashboard ใน Looker Studio
 - การสร้างและปรับแต่งกราฟวงกลม
 - การสร้างช่องข้อความ
 - การสร้างและปรับแต่งกราฟแท่ง
 - การสร้างช่องข้อความบอกจำนวน Total หรือจำนวนรวม
 - การเปลี่ยนประเภทของข้อมูลตัวเลข
 - การสร้างและปรับแต่งกราฟ Bubble map (Google map)
 - การรวมข้อมูล 2 ตัวแปร(คอลัมน์)
 - การสร้างและปรับแต่งตารางแสดงรายละเอียดข้อมูล
 - การสร้างและปรับแต่งตัวกรอง dropdown
 - การโหลดรูปภาพเข้าหน้า Dashboard

3. การแสดงผลและการแบ่งปัน Dashboard

- การแสดงผล Dashboard: วิธีการแสดงผลและการปรับแต่ง Dashboard ใน Looker Studio
- การแบ่งปัน Dashboard: วิธีการแบ่งปัน Dashboard กับผู้ใช้หรือทีมอื่น ๆ ในองค์กร
 - การแชร์หน้า Dashboard แบบเรียลไทม์

- การดาวน์โหลดหน้า Dashboard เป็นไฟล์ PDF

4. (Ethic4) Ethical Framework of Data Project