

## Class period 9

บทที่ 5 การเตรียมข้อมูลสำหรับการแสดงผล 2  
Pandas 102

1

## .isnull().any()

- .any() สามารถใช้สำหรับสรุปค่า True False ในแต่ละคอลัมน์ โดยเอาค่าความจริงภายในแต่ละคอลัมน์มา OR กัน

P	Q	P OR Q
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

```
data_covid.isnull().any()
No.                False
announce_date      False
Notified date      True
sex                True
age               True
Unit              True
nationality        True
province_of_isolation True
risk              True
province_of_onset  True
district_of_onset  True
dtype: bool
```

1

2

## .isnull().all()

- .all() สามารถใช้สำหรับสรุปค่า True False ในแต่ละคอลัมน์ โดยเอาค่าความจริงภายในแต่ละคอลัมน์มา AND กัน

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

```
data_covid.isnull().all()
No.                False
announce_date      False
Notified date      False
sex                False
age               False
Unit              False
nationality        False
province_of_isolation False
risk              False
province_of_onset  False
district_of_onset  False
dtype: bool
```

2

3

## การชี้ค่าในตารางของ .isnull()

- ไม่สามารถชี้ด้วยแบบ basic หรือ .iloc ปกติ เช่น
  - data\_covid['No.'][0].isnull()
  - data\_covid.iloc[0,0].isnull()
- จำเป็นต้องใช้การชี้แบบ .loc numpy array
- ที่มีการติดตาราง เช่น
  - data\_covid.iloc[:,0].isnull()
- เป็นการชี้ค่าไปที่ตารางที่เฉพาะเฉพาะแถวที่ 0 ถึง 1
- และคอลัมน์ที่ 0 คือคอลัมน์แรก

```
data_covid.iloc[:,0].isnull()
0    False
dtype: bool
```

## pandas.isnull

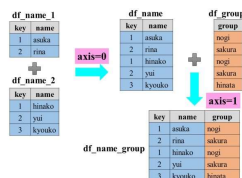
**pandas.isnull(obj)** [\[source\]](#)  
Detect missing values for an array-like object.  
This function takes a scalar or array-like object and indicates whether values are missing (NaN, in numeric arrays, None, or NaN, in object arrays, list, in datetime-like).  
**Parameters:**  
obj : scalar or array-like  
Object to check for null or missing values.  
**Returns:**  
bool or array-like of bool  
For scalar input, returns a scalar boolean. For array input, returns an array of boolean indicating whether each corresponding element is missing.

3

4

## การต่อตารางแกน X แกน y

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



4

5

## ต่อแกน Y pd.concat()

- การต่อแกน Y เป็นการนำแถวข้อมูล 2 ตารางมารวมหรือต่อกัน
- เตรียมตารางที่ต้องการโดยคัดมาเฉพาะข้อมูลที่ต้องการต่อ
  - dataKK = data\_covid[data\_covid['province\_of\_onset'] == 'ขอนแก่น']
  - dataUD = data\_covid[data\_covid['province\_of\_onset'] == 'อุดรธานี']
  - dataMS = data\_covid[data\_covid['province\_of\_onset'] == 'มหาสารคาม']
- จากนั้นใช้ pd.concat ตามด้วย list ตัวแปรที่ใช้เก็บแต่ละตาราง และเก็บตารางที่ต่อกันแล้วไว้ในตัวแปร dataMYisan ผลลัพธ์จะได้ตารางที่มีข้อมูลของจังหวัดขอนแก่น, อุดรธานี, มหาสารคาม
  - import pandas as pd
  - dataMYisan = pd.concat([dataKK, dataUD, dataMS])
  - dataMYisan

5

6

## ต่อแกน X

- การต่อตารางแกน x สามารถทำได้ 2 แบบ
- 1. จับ 2 ตารางมาต่อกันเลย (merge)
- 2. เลือกมาเพิ่มเฉพาะบาง column (map)

6

7

## .merge() จับ 2 ตารางมาต่อกันเลย

- การต่อแบบง่าย คือรู้ว่าจะสองตาราง records ตรงกัน สามารถนำตารางมาต่อกันแบบปกติ
- เตรียมตารางข้อมูลที่จะใช้ต่อตาราง
- `data_province = data_covid[['No.', 'announce_date', 'province_of_onset']]`
- `data_human = data_covid[['No.', 'age', 'sex', 'nationality']]`
- ชื่อตัวแปรที่ต้องการใช้เป็นตารางหลัก ตามด้วย .merge(ตัวแปรตารางที่ต้องการต่อกับตารางที่ต่อแล้วไว้ในตัวแปร full\_table1
- `full_table1 = data_human.merge(data_province)`
- `full_table1.head()` จะเห็นว่าคอลัมน์ announce\_date และ province\_of\_onset ถูกนำมาต่อด้านซ้ายของตาราง

7

8

## .sort\_values()

- ใช้เรียงแถวข้อมูลทั้งหมดตามคอลัมน์ที่ต้องการ โดยใช้ชื่อตัวแปรตารางที่ต้องการตามด้วย .sort\_values(ชื่อคอลัมน์ที่ต้องการ) เช่น
- `data_human = data_covid[['No.', 'age', 'sex', 'nationality']]`
- `data_human2 = data_human.sort_values('age')`
- `data_human2`
- ผลลัพธ์จะได้ตารางในตัวแปร data\_human2 ที่แถวข้อมูลทั้งหมดเรียงตาม age โดย default จะเรียงน้อยไปมาก, a-z

8

9

## จำลองการใช้งานจริงของ .merge()

- .merge() จะไม่สามารถต่อตารางที่ไม่ชื่อคอลัมน์ที่เหมือนกันได้ เช่น
- `data_human2_renamed = data_human2.rename(columns={'No.': 'patientNumber'})`
- `data_human2_renamed.merge(data_province)`
- ผลลัพธ์ที่ได้จะ MergeError: No common columns to perform merge on.
- ส่วนใหญ่การใช้งานจริงในการ merge ต่อตารางจะเจอปัญหานี้
- ซึ่งสามารถแก้ปัญหาได้โดยการ parameter: left\_on, right\_on ชื่อคอลัมน์ที่ต้องการให้เป็น index เพื่อ merge ให้ตรงกัน ข้อมูลในคอลัมน์ที่ใช้เป็น index จะต้องไม่ซ้ำกัน หลักการใช้งานเหมือนกับ primary key

9

10

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

- `full_table3 = data_human2_renamed.merge(data_province, left_on='patientNumber', right_on='No.')`
- วิธีใช้งาน กำหนด merge ตารางหลักและตารางที่ต้องการต่อแบบปกติ จากนั้นเพิ่ม left\_on กำหนดชื่อคอลัมน์ที่ต้องการให้เป็น index ของตารางหลัก และเพิ่ม right\_on กำหนดชื่อคอลัมน์ที่ต้องการให้เป็น index ของตารางที่ต้องการต่อ เก็บตารางที่ต่อแล้วไว้ในตัวแปร full\_table3
- `full_table3.head()` จะเห็นว่า 2 ตารางถูกนำต่อกันโดยใช้ patientNumber ของตารางหลักและ No. ของตารางรองเป็น index

```
full_table3.head()
```

	patientNumber	age	sex	nationality	No.	announce_date	province_of_onset
0	114170	0.75	male	Burmes	114170	19/5/2021	บราซิล
1	114205	0.95	male	Thai	114205	19/5/2021	บราซิล
2	146074	1.00	male	Thai	146074	29/5/2021	บราซิล
3	122692	1.00	male	Thai	122692	21/7/2021	บราซิล
4	92541	1.00	male	Thai	92541	19/5/2021	บราซิล

10

11

## การสร้างคอลัมน์ใหม่ด้วย pandas

- คุณสมบัติของ pandas ในการสร้างคอลัมน์ คือ สามารถสร้างคอลัมน์ใหม่ให้ตารางที่ต้องการได้โดย
- ยกตัวอย่าง df คือตัวแปรตารางที่ต้องการสร้างคอลัมน์ใหม่
- df[column ใหม่] = (list ที่มีจำนวนสมาชิกเท่ากับจำนวน record ของ df) เช่น

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

11

12

## ตัวอย่างการสร้างคอลัมน์ใหม่ด้วย pandas

- `data_human2_renamed['num'] = range(data_human2_renamed.shape[0])`
- สร้าง list เล่าลำดับ record ของข้อมูลตารางตัวแปร `data_human2_renamed` ใช้ `.shape[0]` ตรวจสอบจำนวนแถวทั้งหมด และสร้าง list ด้วย `range` จะได้ list ตั้งแต่ 0 ถึงแถวสุดท้ายของตาราง จากนั้นนำ list ลำดับแถวที่ได้ไปสร้างเป็นคอลัมน์ที่ชื่อ `num` ของตัวแปรตาราง `data_human2_renamed`
- `data_human2_renamed` ผลลัพธ์จะได้

12

13

## .map() เลือกมาเพิ่มเฉพาะบาง column

- `data_human2_renamed['patientNumber'].map(data_covid.set_index('No.')['risk'])`
- `.map` จะใช้การกำหนด index ของตารางหลักและตารางที่ต้องการข้อมูลบางคอลัมน์มาต่อ
- ในตัวอย่างต้องการ list ข้อมูลคอลัมน์ `risk` เพื่อนำมาต่อในตาราง `data_human2_renamed` โดยให้คอลัมน์ `patientNumber` จากตารางหลักและ `No.` จากตาราง `data_covid` ที่ต้องการข้อมูลคอลัมน์ `risk` เป็น index แล้ว return list ของค่าในคอลัมน์ `risk` เพื่อใช้เพิ่มในตารางหลัก

13

14

## การใช้ .map() ต่อตาราง

- `data_human2_renamed['detail'] = data_human2_renamed['patientNumber'].map(data_covid.set_index('No.')['risk'])`
- นำ list ข้อมูล `risk` ที่ได้จากการ `map` ไปสร้างเป็นคอลัมน์ที่ชื่อ `detail` ของตารางตัวแปร `data_human2_renamed`
- `data_human2_renamed` ผลลัพธ์จะได้

14

15

## Homework class period 9

- สร้างตารางใหม่ ที่ค่าใน `sex` เป็น missing ทั้งหมด
- สรุบบ้างทำไม record นั้นๆถึงเป็น missing (ยังไม่ได้สอน ให้องหารวิธีสรุปด้วยตัวเอง)

15

16