Class period 9

Pandas 102

.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
```

.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
```

การชี้ค่าในตารางของ .isnull()

- .ไม่สามารถชี้ด้วยแบบ basic หรือ .iloc ปกติ เช่น
- `data_covid['No.'][0].isnull()`
- `data_covid.iloc[0,0].isnull()`

- จำเป็นต้องใช้การชี้แบบ .iloc หรือ numpy array
- ที่มีการตัดตาราง เช่น
- `data_covid.iloc[:1,0].isnull()`
- เป็นการชี้ค่าไปที่ตารางที่ตัดมาเฉพาะแถวที่ 0 ถึง 1
- และคอลัมน์ที่ 0 คือคอลัมน์แรก

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

```
0    False
Name: No., 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, `NaT` in datetimelike).

Parameters:

obj : *scalar or array-like*

Object to check for null or missing values.

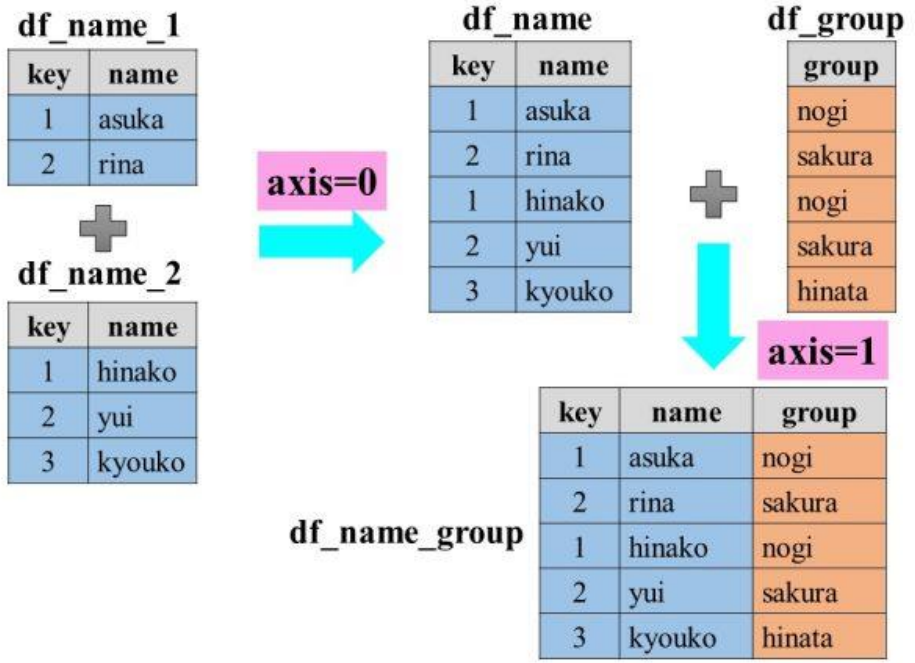
Returns:

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.

การต่อตารางแกน X แกน y

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



ต่อแกน Y pd.concat()

- การต่อแกน Y เป็นการนำแถวข้อมูล 2 ตารางมารวมหรือมาต่อกัน
- เตรียมตารางที่ต้องการโดยตัดมาเฉพาะข้อมูลที่ต้องการต่อ
- `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 ผลลัพธ์จะได้ตารางที่มีแต่ข้อมูลของจังหวัดขอนแก่น, อุดรธานี, มหาสารคาม
- `import pandas as pd`
- `dataMYisan = pd.concat([dataKK,dataUD,dataMS])`
- `dataMYisan`

ต่อแกน X

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

.merge() จับ 2 ตารางมาต่อกันเลย

- การต่อแบบง่าย คือรู้ว่าสองตาราง 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` (ตัวแปรตารางที่ต้องการต่อ) เก็บตารางที่ต่อแล้วไว้ในตัวแปร `full_table1`
- `full_table1 = data_human.merge(data_province)`
- `full_table1.head()` จะเห็นว่าคอลัมน์ `announce_date` และ `province_of_onset` ถูกนำมาต่อด้านซ้ายของตาราง

.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

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` ต่อตารางจะเจอปัญหาแบบนี้
- ซึ่งสามารถแก้ปัญหานี้โดยการใช้ `parameter: left_on, rigth_on` ชี้ชื่อคอลัมน์ที่ต้องการให้เป็น `index` เพื่อ `merge` ให้ตรงกัน ข้อมูลในคอลัมน์ที่ใช้เป็น `index` จะต้องไม่ซ้ำกัน หลักการใช้งานเหมือนกันกับ `primary key`

parameter: left_on, rigth_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	หญิง	Burmese	114170	19/5/2021	ระนอง
1	114205	0.95	หญิง	Thailand	114205	19/5/2021	นครศรีธรรมราช
2	146574	1.00	ชาย	Thailand	146574	29/5/2021	สงขลา
3	523593	1.00	หญิง	Thailand	523593	27/7/2021	ยะลา
4	672541	1.00	ชาย	Thailand	672541	5/8/2021	เชียงราย

การสร้างคอลัมน์ใหม่ด้วย pandas

- คุณสมบัติของ pandas ในการสร้างคอลัมน์ คือ สามารถสร้างคอลัมน์ใหม่ให้ตารางที่ต้องการได้โดย
- ยกตัวอย่าง df คือตัวแปรตารางที่ต้องการสร้างคอลัมน์ใหม่
- df['ชื่อ 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
```

ตัวอย่างการสร้างคอลัมน์ใหม่ด้วย pandas

- `data_human2_renamed['num'] = range(data_human2_renamed.shape[0])`
- สร้าง `list` เลขลำดับ `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

`data_human2_renamed`

	patientNumber	age	sex	nationality	num
114169	114170	0.75	หญิง	Burmese	0
114204	114205	0.95	หญิง	Thailand	1
146573	146574	1.00	ชาย	Thailand	2
523592	523593	1.00	หญิง	Thailand	3
672540	672541	1.00	ชาย	Thailand	4
...	...	...	...	...	...
839745	839746	NaN	ชาย	NaN	839766
839746	839747	NaN	ชาย	Thailand	839767
839747	839748	NaN	ชาย	Thailand	839768
839748	839749	NaN	ชาย	Thailand	839769
839752	839753	NaN	ชาย	Thailand	839770

839771 rows × 5 columns

.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

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

114169                                  อยู่ระหว่างการสอบสวน
114204                                  อยู่ระหว่างการสอบสวน
146573     สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
523592     สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
672540     สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
                                    ...
839745                                      ทัณฑสถาน/เรือนจำ
839746                                      ทัณฑสถาน/เรือนจำ
839747                                      ทัณฑสถาน/เรือนจำ
839748                                      ทัณฑสถาน/เรือนจำ
839752                                      ทัณฑสถาน/เรือนจำ
Name: patientNumber, Length: 839771, dtype: object
```

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

`data_human2_renamed`

	patientNumber	age	sex	nationality	num	detail
114169	114170	0.75	หญิง	Burmese	0	อยู่ระหว่างการสอบสวน
114204	114205	0.95	หญิง	Thailand	1	อยู่ระหว่างการสอบสวน
146573	146574	1.00	ชาย	Thailand	2	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
523592	523593	1.00	หญิง	Thailand	3	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
672540	672541	1.00	ชาย	Thailand	4	สัมผัสใกล้ชิดกับผู้ป่วยยืนยันรายก่อนหน้านี้
...	...	...	...	...	...	...
839745	839746	NaN	ชาย	NaN	839766	ทัณฑสถาน/เรือนจำ
839746	839747	NaN	ชาย	Thailand	839767	ทัณฑสถาน/เรือนจำ
839747	839748	NaN	ชาย	Thailand	839768	ทัณฑสถาน/เรือนจำ
839748	839749	NaN	ชาย	Thailand	839769	ทัณฑสถาน/เรือนจำ
839752	839753	NaN	ชาย	Thailand	839770	ทัณฑสถาน/เรือนจำ

839771 rows × 6 columns

Homework class period 9

- สร้างตารางใหม่ ที่ค่าใน sex เป็น missing ทั้งหมด
- สรุปว่าทำไม record นั้นๆถึงเป็น missing (ยังไม่ได้สอน ให้ลองหาวิธีสรุปด้วยตัวเอง มองด้วยตา)