

Class period 9

บทที่ 5 การเตรียมข้อมูลสำหรับการแสดง 2
Pandas 102

1

.isnull().any()

- .any() สามารถใช้สำหรับดูค่า True False ในแต่ละคอลัมน์ โดยหากพบว่ามีค่าในคอลัมน์นั้นค่า OR เป็น

P	Q	# of null
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE


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

no	False
announcement_date	True
initial_data	True
age	True
sex	True
nationality	True
province_of_isolation	True
risk	True
province_of_onset	True
district_of_onset	True
player_team	True

2

.isnull().all()

- .all() สามารถใช้สำหรับดูค่า True False ในแต่ละคอลัมน์ โดยหากพบว่ามีค่าในคอลัมน์นั้นค่า AND เป็น

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


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

no	False
announcement_date	False
initial_data	False
age	False
sex	False
nationality	False
province_of_isolation	False
risk	False
province_of_onset	False
district_of_onset	False
player_team	False

3

การชี้ค่าในตารางของ .isnull()

- โมดูลมาตรฐานของ basic หรือ Aoc ยังใช้ เช่น
- `data_covid["No."][0].isnull()`
- `data_covid.iloc[0,0].isnull()`

- จำเป็นต้องใช้การเขียน `iloc` numpy array
- เขียนการตรวจสอบ เช่น
- `data_covid.iloc[0,0].isnull()`
- เป็นการชี้ไปที่ตารางที่แถวแรกและคอลัมน์ที่ 0 ถึง 1
- และคอลัมน์ที่ 0 คือคอลัมน์แรก

pandas.isnull

pandas.isnull() (method)

(detect missing values for an array-like object)

This function tests a scalar or array-like object and indicates whether values are missing (NaN) or not (not NaN). For array-like objects, the result is a boolean array (1D) of the same shape as the input.

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

Returns: boolean array of bool. For scalar input, returns a scalar boolean. For array input, returns an array of booleans indicating whether each corresponding element is missing.

4

การต่อตารางแนวน X กับ Y

- ต่อแนวน Y คือ เพิ่มแถว (เพิ่มจำนวนข้อมูล)
- ต่อแนว X คือ เพิ่ม คอลัมน์ (เพิ่มขนาดมิติของข้อมูล)



5

ต่อแนวน Y pd.concat()

- การต่อแนวน Y เป็นการนำแถวของ 2 ตารางมาเชื่อมต่อกัน

- เขียนการเชื่อมต่อตารางโดยที่ค่าเฉพาะของข้อมูลการต่อ

- `dataEX = data_covid[data_covid['province_of_onset'] == "sumbu"]`
- `dataUD = data_covid[data_covid['province_of_onset'] == "sumbu"]`
- `dataUD = data_covid[data_covid['province_of_onset'] == "sumbu"]`

- การเชื่อมต่อ pd.concat สามารถใช้ สำหรับเพิ่มข้อมูลเฉพาะที่เพิ่มเข้ามาในตาราง dataEX และ dataUD

- การเพิ่มข้อมูลเฉพาะที่เพิ่มเข้ามาในตาราง dataEX และ dataUD

- Import pandas as pd
- `dataH1can = pd.concat([dataEX, dataUD, dataUD])`
- `dataH1can`

6

ต่อแนวน X

- การต่อตารางแนวน X สามารถทำได้ 2 แบบ
- 1. กับ 2 ตารางมาต่อกัน (merge)
- 2. เลือกมาเพิ่มเฉพาะบาง column (map)

7

.merge() กับ 2 ตารางมาต่อกันเลย

- การต่อแนวน X คือ การเชื่อมตาราง records ตรงกัน สามารถนำตารางมาต่อกันแบบปกติ

- เขียนการเชื่อมต่อตาราง

- `data_province = data_covid[["No.", "announcement_date", "province_of_onset"]]`
- `data_human = data_covid[["No.", "age", "sex", "nationality"]]`
- ใช้ข้อมูลที่ต้องการในตารางมาเชื่อมต่อกัน โดยดูว่าตารางที่ต้องการเชื่อมมีค่าในชื่อ full_table
- `full_table1 = data_human.merge(data_province)`
- `full_table1.head()` จะเห็นค่าของ announcement_date และ province_of_onset ถูกนำค่ามาต่อกัน

8

.sort_values()

- ใช้เรียงลำดับข้อมูลในตารางตามลำดับที่ต้องการ โดยเรียงตามค่าของ sort_values (เรียงตามค่าที่ต้องการ)

- `data_human = data_covid[["No.", "age", "sex", "nationality"]]`
- `data_human2 = data_human.sort_values("age")`
- `data_human2`

- เมื่อเรียงลำดับค่าใน data_human2 เป็นค่าที่เรียงตามค่า age โดย data_human2 จะเรียงตามค่า age

9

จำลองการใช้งานจริงของ .merge()

- `.merge()` ไม่สามารถต่อตารางที่มีชื่อคอลัมน์ซ้ำกันได้เลย เช่น


```
data_human2_renamed = data_human2.rename(columns={'No.': 'patientNumber'})
data_human2_renamed.merge(data_province)
```
- ผลลัพธ์คือ `MergeError: No common columns to perform merge on.`
- สาเหตุจากการใช้งานคือการ merge ตารางของเรามีปัญหานี้
- เพื่อข้ามการแจ้งเตือนการให้ parameter `left_on, right_on` ซึ่งเปลี่ยนให้ชื่อการ join เป็น index เพื่อ merge ได้ตรงกับข้อมูลในคอลัมน์ที่มี index ของแต่ละตาราง เพื่อให้การ join ทำงานเป็น primary key

10

parameter: left_on, right_on ของ .merge()

- `full_table3 = data_human2_renamed.merge(data_province, left_on="patientNumber", right_on="No.")`
- วิธีนี้ใช้ `data_province` ตารางที่มีชื่อคอลัมน์ซ้ำกันอยู่แล้ว เปลี่ยนเป็น `left_on` ค่าของชื่อคอลัมน์ที่ต้องการให้เป็น index ของตารางเดิม และใช้ `right_on` ค่าของชื่อคอลัมน์ที่ต้องการให้เป็น index ของตารางที่ถูกรวมเข้าตารางที่มีอยู่แล้วในชื่อ `full_table3`
- `full_table3.head()` จะเห็นว่า 2 ตารางถูกนำมาเชื่อมกันได้ perfect index ของตารางเดิมและ No. ของตารางของ index

11

การสร้างคอลัมน์ใหม่ด้วย pandas

- คุณสมบัติของ pandas ในการสร้างคอลัมน์ คือ สามารถสร้างคอลัมน์ใหม่ได้หลายครั้งโดยการนำโดย
- ยึดตามการ `set` คือสามารถสร้างคอลัมน์ใหม่ได้
- `set` (ชื่อ คอลัมน์ใหม่) = (ชื่อ ที่ใช้จำนวนซ้ำกันเท่ากับจำนวน record ของ set) เช่น
- `data_human2_renamed.head()`
- `data_human2_renamed['num'] = range(data_human2_renamed.shape[0])`
- `data_human2_renamed`

12

ตัวอย่างการสร้างคอลัมน์ใหม่ด้วย pandas

- `data_human2_renamed['num'] = range(data_human2_renamed.shape[0])`
- สร้าง list ของค่า index record ของข้อมูลตารางเดิมด้วย `data_human2_renamed` ได้ `data_human2_renamed` ตารางถูกนำมาเชื่อมกันเรียบร้อยแล้ว list `data_human2_renamed` จะได้ list ตั้งแต่ 0 ถึงจำนวนของข้อมูลตาราง จากนั้นนำ list ดังกล่าวมาใส่ในตำแหน่งคอลัมน์ชื่อ `num` ของตาราง `data_human2_renamed`
- `data_human2_renamed` ผลลัพธ์ที่ได้

13

.map() เลือกมาเพิ่มเฉพาะบาง column

- `data_human2_renamed['patientNumber'].map(data_covid.set_index("No.")["risk"])`
- `map` จะใช้การกำหนด index ของตารางที่มีและตารางที่ต้องการนำมาเชื่อมกัน
- ในที่นี้คือการนำ list ของคอลัมน์ `risk` ที่อยู่ใน `data_covid` มาใส่ใน `data_human2_renamed` โดยใช้ `map` เพื่อเชื่อมข้อมูลจากตารางที่มีและ `data_covid` ที่ต้องการนำมาเชื่อม `risk` เป็น index แล้วนำมา list ของค่าในคอลัมน์ `risk` แล้วใส่ลงในตาราง

14

การใช้ .map() ต่อตาราง

- `data_human2_renamed['detail'] = data_human2_renamed['patientNumber'].map(data_covid.set_index("No.")["risk"])`
- นำ list ของ `risk` ที่ได้จาก `map` ไปใส่เป็นคอลัมน์ชื่อ `detail` ของตารางเดิมด้วย `data_human2_renamed`
- `data_human2_renamed` ผลลัพธ์ที่ได้

15

Homework class period 9

- สร้างตารางใหม่: ตั้งค่าใน `sex` เป็น missing ซ้ำกัน
- สร้างค่าใน record นั้นๆให้เป็น missing (ถ้าไม่มีเลข 10 ของทาวเวอร์ให้ลบด้วย)

16