


Class period 6

บทที่ 4 การจัดการข้อมูลในรูปแบบตาราง
Pandas 101

1




Pandas

- Pandas เป็นหนึ่งใน package ที่สำคัญของ python ใช้สำหรับการจัดการข้อมูลรูปแบบตาราง .CSV
- `import pandas as pd`



1

2




Download ข้อมูลรายงาน COVID-19 ประจำวัน ข้อมูลประจำประเทศไทย

- <https://data.go.th/dataset/covid-19-daily>

 ราชบัณฑิตยสถาน COVID-19 ประจำวัน 1,123 downloads	CSV = Comma Separated Values ในการจะใส่ค่าแต่ละค่า จะใช้ comma ในการแยก
 covid-19-daily_data_dictionary 4,354 downloads	Meta data = Data that description data ข้ออธิบายข้อมูล

2

3



การนำเข้าข้อมูลเข้า

- สร้าง folder ใน google drive และนำข้อมูล .csv ที่ดาวน์โหลดเข้าไปเก็บไว้ใน folder ที่สร้าง
- นำเข้า package pandas และ package ของ google.colab ที่ชื่อ drive เพื่อเชื่อมต่อ google drive กับ google colab


```
import pandas as pd
from google.colab import drive
drive.mount('/content/drive')
```

- กด Connect Google Drive และเลือก Account
- กด select all และกด continue



3

4




Import os

- นำเข้า package os เพื่อใช้ในการทำงานต่างๆที่เกี่ยวข้องกับไฟล์ เช่น การชี้ไฟล์ การลบไฟล์ การสร้างไฟล์เตอร์ เป็นต้น
- โดยในกรณี google drive จะใช้ os เพื่อชี้ไฟล์ ว่าไฟล์ที่ต้องการใช้งานอยู่ path ไหนใน google drive ที่เชื่อม
- path คือเส้นทางที่อยู่ไฟล์ จะทำงานเหมือนกับ path ใน window เช่น
- E:\WORKSPACE\Basic Programming\confirmed-cases.csv
- หมายความว่า ไฟล์ confirmed-cases.csv อยู่ใน drive E ไฟล์เตอร์ WORKSPACE ในไฟล์เตอร์ Basic Programming

4

5



การ set path

- `path = '/content/drive/My Drive/dataviz_2024_data'`
- Set 'path' ที่ชี้ไปยังไฟล์เตอร์ที่เก็บไฟล์ .csv ไว้ใน google drive และเก็บ string ไว้ในตัวแปร path
- โดย path หรือเส้นทางที่ชี้ไปยังไฟล์เตอร์และไฟล์ต่างๆ ใน os ของ window, mac หรือ linux จะใช้สัญลักษณ์ใน path แตกต่างกัน
- Package os จะช่วยให้สามารถเชื่อม path โดยไม่ต้องคำนึงถึงสัญลักษณ์ เพราะ os จะใช้สัญลักษณ์เชื่อมให้เองตาม platform ที่ใช้งานอยู่ เช่น
- ถ้าใช้ os ของ window ก็จะเชื่อมด้วย \
- ถ้าใช้ os ของ mac หรือ linux จะเชื่อมด้วย /

5

6

คำสั่ง os.path.join()

- เป็นคำสั่งที่ใช้สำหรับเชื่อม path เข้าด้วยกัน
- ```
import os
```
- ```
covid_file_path = os.path.join(path, 'confirmed-cases.csv')
```
- หมายความว่า เชื่อมตัวแปร path ที่ set ไว้ก่อนหน้านั้นเข้ากับชื่อไฟล์ confirmed-cases.csv เก็บไว้ในตัวแปร covid_file_path
- ```
print(covid_file_path)
```
- ผลลัพธ์จะได้เส้นทางไปยังไฟล์ที่ต้องการอยู่ในตัวแปร
- ```
covid_file_path
```
- ```
./content/drive/My Drive/dataviz_2024_data/confirmed-cases.csv
```



6

7

## load data to memory คำสั่ง pd.read\_csv()

- pd.read\_csv เป็นคำสั่งที่ใช้สำหรับโหลดข้อมูล
- ```
data_covid = pd.read_csv(covid_file_path)
```
- โหลดข้อมูลไฟล์ confirmed-cases.csv ตามเส้นทาง covid_file_path
- ```
data_covid
```
- ผลลัพธ์จะได้หน้าไฟล์ csv

| no.  | onset_date | onset_datetime | date     | age | sex  | nationality | province_of_residence | province_of_onset | district_of_onset | city     | country  |
|------|------------|----------------|----------|-----|------|-------------|-----------------------|-------------------|-------------------|----------|----------|
| 0    | 1          | 1/1/2020       | 1/1/2020 | 100 | male | Thai        | Chonburi              | Chonburi          | Chonburi          | Chonburi | Thailand |
| 1    | 2          | 1/1/2020       | 1/1/2020 | 100 | male | Thai        | Chonburi              | Chonburi          | Chonburi          | Chonburi | Thailand |
| 2    | 3          | 1/1/2020       | 1/1/2020 | 100 | male | Thai        | Chonburi              | Chonburi          | Chonburi          | Chonburi | Thailand |
| 3    | 4          | 1/1/2020       | 1/1/2020 | 100 | male | Thai        | Chonburi              | Chonburi          | Chonburi          | Chonburi | Thailand |
| 4    | 5          | 1/1/2020       | 1/1/2020 | 100 | male | Thai        | Chonburi              | Chonburi          | Chonburi          | Chonburi | Thailand |
| ...  | ...        | ...            | ...      | ... | ...  | ...         | ...                   | ...               | ...               | ...      | ...      |
| 1000 | 1000       | 1/1/2020       | 1/1/2020 | 100 | male | Thai        | Chonburi              | Chonburi          | Chonburi          | Chonburi | Thailand |

7

8

## Parameter: encoding

- ใช้สำหรับกำหนดภาษาของไฟล์ที่จะอ่าน เช่น
- ```
data_covid = pd.read_csv(covid_file_path, encoding='utf-8')
```
- encoding='utf-8' อาจจะอ่านภาษาไทยได้แต่จะมี ภาษาไทยบางไฟล์ที่มันอ่านไม่ได้แล้วเกิด error
- การอ่านไฟล์ csv ที่มีข้อมูลที่เป็นภาษาไทยที่ครอบคลุมที่สุดจะใช้
- ```
encoding='iso-8859-11'
```



8

9

## คำสั่ง .head()

- ```
data_covid.shape
```

 ชื่อตัวแปรที่เก็บข้อมูลตามด้วย .head() ใช้เพื่อให้เห็นข้อมูลส่วนต้นและข้อมูลในตาราง เฉพาะ 5 แถวแรก int, default=5
- สามารถกำหนดจำนวนคอลัมน์ที่ต้องการให้แสดงได้ เช่น
- ```
data_covid.head(10)
```

 จะแสดงข้อมูลส่วนต้นและข้อมูลในตาราง 10 แถว



9

10

## คำสั่ง .shape

- ```
data_covid.shape
```
- ชื่อตัวแปรที่เก็บข้อมูลตามด้วย .shape ใช้ตรวจสอบขนาดของข้อมูล ผลลัพธ์จะได้
- ```
(839771, 11)
```
- หมายความว่า มีข้อมูลทั้งหมด 839,771 แถว มีคอลัมน์ 11 คอลัมน์



10

11

## การเข้าถึงข้อมูลตารางแบบ basic

- ใช้ชื่อคอลัมน์ในการดึงข้อมูลในคอลัมน์ที่ต้องการ
- ```
data_covid['province_of_onset']
```

```
data_covid['province_of_onset']
0      กรุงเทพมหานคร
1      กรุงเทพมหานคร
2      กรุงเทพมหานคร
3      กรุงเทพมหานคร
4      กรุงเทพมหานคร
...
839766      กรุงเทพมหานคร
839767      กรุงเทพมหานคร
839768      กรุงเทพมหานคร
839769      กรุงเทพมหานคร
839770      กรุงเทพมหานคร
Name: province_of_onset, Length: 839771, dtype: object
```



11

12

การชี้ค่าในข้อมูลตารางแบบ basic



- การใช้ชื่อคอลัมน์และลำดับแถวในการดึงข้อมูลในแถวและคอลัมน์ที่ต้องการ
- `data_covid['province_of_onset'] [4]`
- ผลลัพธ์จะได้ข้อมูลแถวที่ 4 นับจาก 0 ในคอลัมน์ province_of_onset
- 'กรุงเทพมหานคร'

12

13

การชี้ค่าในข้อมูลตารางแบบ .iloc[]



- โดยการมองข้อมูลตารางในรูปแบบ numpy array หรือ matrix จะใช้ตำแหน่งเพื่อชี้ข้อมูลที่ต้องการ เช่น
- `data_covid.iloc[4,9]`
- ผลลัพธ์จะได้ข้อมูลแถวที่ 4 คอลัมน์ที่ 9 (ในมุมมอง matrix คือหลักที่ 9) นับจาก 0 คือคอลัมน์ province_of_onset
- 'กรุงเทพมหานคร'

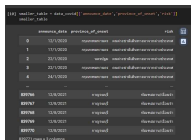
13

14

Table slicing การเลือกข้อมูลเฉพาะคอลัมน์ที่ต้องการ



- การเลือกข้อมูลเฉพาะคอลัมน์ที่ต้องการมาเก็บไว้ในตัวแปรเพื่อนำไปใช้งาน
- `smaller_table = data_covid[['announce_date', 'province_of_onset', 'risk']]`
- หมายความว่า เลือกข้อมูลคอลัมน์ชื่อ announce_date, province_of_onset, risk ในข้อมูลเก็บอยู่ในตัวแปร data_covid และเก็บข้อมูลเฉพาะคอลัมน์ที่เลือกไว้ในตัวแปร smaller_table
- ผลลัพธ์จะได้



	announce_date	province_of_onset	risk
0	2020-02-01	กรุงเทพมหานคร	Low
1	2020-02-01	กรุงเทพมหานคร	Low
2	2020-02-01	กรุงเทพมหานคร	Low
3	2020-02-01	กรุงเทพมหานคร	Low
4	2020-02-01	กรุงเทพมหานคร	Low

14

15

Table slicing การเลือกข้อมูลเฉพาะแถวที่ต้องการแบบง่าย



- การเลือกข้อมูลเฉพาะแถวที่ต้องการแบบง่าย คือการมองข้อมูลตารางในรูปแบบ array แต่การนำไปใช้งาน ใช้งานอะไรไม่ค่อยได้
- `data_covid.iloc[1:5,:]`
- หมายความว่า
- 1:5 คือเลือกข้อมูลที่อยู่ในแถวที่ 1 ไปจนถึงแถวที่ 4
- ,: คือเลือกทุกคอลัมน์ ดังนั้น
- `data_covid.iloc[1:5,:]` คือเลือกข้อมูลในตัวแปร data_covid ที่อยู่ในแถวที่ 1 ไปจนถึงแถวที่ 4 และเลือกทุกคอลัมน์

15

16

Table slicing การเลือกข้อมูลเฉพาะแถวที่ต้องการแบบ advanced



- การเลือกข้อมูลเฉพาะแถวที่ต้องการแบบ advanced คือการใช้ logic query ในการเลือกข้อมูล
- `data_covid[data_covid['province_of_onset'] == 'ขอนแก่น']`
- หมายความว่า เลือกข้อมูลที่อยู่ในตัวแปร data_covid โดยกำหนดชื่อคอลัมน์ที่ต้องการคือ province_of_onset และต้องการข้อมูลทุกแถวที่มีข้อมูลในคอลัมน์ province_of_onset เป็นจังหวัดขอนแก่น

16

17

วิธีการเลือกข้อมูลเฉพาะแถวที่ต้องการแบบ advanced



- การทำงานของการเลือกข้อมูลเฉพาะแถวที่ต้องการแบบ advanced
- คือ การใส่แถวที่ต้องการ และใส่ list True/False ที่มีขนาดเท่ากับจำนวนแถว เพื่อเปรียบเทียบข้อมูลในแถวนั้นๆ ด้วยเงื่อนไข logical expression (True/False) เช่น
- สร้างตารางใช้สำหรับยกตัวอย่าง
- `eight_rows_covid = data_covid.iloc[:8,:]`
- `eight_rows_covid`
- หมายความว่า เลือกข้อมูลในตัวแปร data_covid แถวที่ 0 ถึงแถวที่ 7 ทุกคอลัมน์เก็บไว้ในตัวแปร eight_rows_covid

17

18

การทำงานของเลือกข้อมูลเฉพาะแถวที่ต้องการแบบ advanced

- ใส่แถวที่ต้องการด้วยการกำหนดค่า True(แถวที่ต้องการ)/False(แถวที่ไม่ต้องการ)
- `eight_rows_covid[True, True, False, True, True, True, True, False]`
- ผลลัพธ์จะได้ข้อมูลตามค่า True/False ที่เลือกใน list คือแถวที่ 0, 1, 3, 4, 5, 6
- เช่นเดียวกับการสร้าง list ของ logical expression แต่แทนที่จะเลือกเองโดยการใส่ list True/False ให้กำหนดเงื่อนไขและข้อมูลที่ต้องการ เพื่อเปรียบเทียบและเลือกข้อมูลที่ตรงตามเงื่อนไข โดยถ้าตรงตามเงื่อนไขคือ True ไม่ตรงตามเงื่อนไขคือ False

18

19

การสร้าง list ของ logical expression

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

19

20

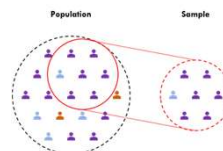
นำ list ของ logical expression ที่สร้างมาใช้งาน

- ซึ่งเมื่อนำมาใช้งานเลือกข้อมูลเฉพาะแถวที่ต้องการแบบ advanced คือการใช้ logic query ในการเลือกข้อมูลนั่นเอง
- `eight_rows_covid[eight_rows_covid['province_of_onset'] == 'กรุงเทพมหานคร']`
- ผลลัพธ์จะได้ข้อมูลทุกแถวที่มีข้อมูลในคอลัมน์ province_of_onset เป็น กรุงเทพมหานคร คือแถวที่ 0, 1, 3, 4, 5, 6

20

21

Bias ในข้อมูล



- bias คืออคติหรือความลำเอียงที่อาจแฝงอยู่ในข้อมูล ซึ่งอาจเกิดจากวิธีการเก็บข้อมูล การออกแบบแบบสำรวจ หรือความผิดพลาดในการป้อนข้อมูล

21

22

สำรวจ Bias ในชุดข้อมูล



- ใช้ชุดข้อมูลในโลกแห่งความเป็นจริงที่มี bias หรือข้อจำกัดที่รู้จักเป็นอย่างดีในชั้นเรียน
- ให้นักเรียนวิเคราะห์ชุดข้อมูลเพื่อระบุ bias ที่อาจเกิดขึ้น เช่น การมีตัวแทนของกลุ่มบางกลุ่มน้อยเกินไปหรือมากเกินไป

22

23

เทคนิคการประมวลผลข้อมูล

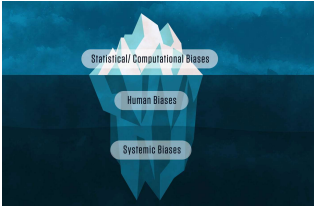
- ครอบคลุมเทคนิคการประมวลผลข้อมูลที่สามารถช่วยลด bias เช่น
- การจัดการกับข้อมูลที่ขาดหายไป
- การจัดการกับค่าผิดปกติ (outliers)
- การสร้างความสมดุลของการกระจายตัวของคลาส

23

24

Algorithmic Bias

- bias สามารถถูกขยายผลหรือเกิดขึ้นซ้ำๆ ผ่านอัลกอริทึม และไม่แสดงการเรียนรู้ของเครื่อง
- ตัวอย่างของ algorithmic bias เช่น ระบบการทำนายอาชญากรรมที่ลำเอียงหรืออัลกอริทึมการรับสมัครงานที่เลือกปฏิบัติตามเพศหรือเชื้อชาติ
- นัยยะด้านจริยธรรมของการใช้อัลกอริทึมที่ลำเอียง และความสำคัญของความโปร่งใสและการไม่เลือกปฏิบัติในการตัดสินใจเชิงอัลกอริทึม



24

25

กรณีศึกษาและตัวอย่างจากโลกแห่งความเป็นจริง

- นำเสนอกรณีศึกษาและตัวอย่างจริงของอัลกอริทึมที่ลำเอียงหรือระบบที่ขับเคลื่อนด้วยข้อมูลที่มีผลกระทบเชิงลบ
- วิเคราะห์กรณีเหล่านี้ในการอภิปรายในชั้นเรียนเพื่อระบุที่มาของ bias นัยยะด้านจริยธรรม และวิธีแก้ปัญหาที่อาจเกิดขึ้น
- กระตุ้นให้นักเรียนคิดไตร่ตรองถึงบทเรียนที่ได้เรียนรู้จากกรณีศึกษาเหล่านี้ และนำมาประยุกต์ใช้กับแนวปฏิบัติในการเขียนโปรแกรมของตนเอง
- <https://www.prolific.com/resources/shocking-ai-bias>

25

26

Homework class period 6

- (ให้ใช้เฉพาะที่อาจารย์สอนไปแล้วในวิชานี้)
- คำนวน อายุเฉลี่ย ของผู้หญิง และผู้ชาย ของข้อมูลทั้งหมด
- คำนวน อายุเฉลี่ย ของผู้หญิง และผู้ชาย ของผู้บ่่วยในจังหวัดขอนแก่น
- หากจำนวนผู้บ่่วยที่เป็นคน "คนต่างชาติดำเนินทางมาจากต่างประเทศ"

26

27