


Class period 18

บทที่ 7 การแสดงผลการเปรียบเทียบข้อมูล (ต่อ)

Visualize_Data_Comparison

Bar chart - Grouped Bar chart


1



โหลดข้อมูล

- ให้นักศึกษาโหลดข้อมูล search_request.csv จาก
- https://drive.google.com/drive/folders/1AztvMSDdZjwDHF5l0T51VSCYVRyQZ37fbclid=IwAR1ITa6xSC4Yck3-SPxk4Y2EABAXYFW5HTBwt_xFDtDISIP1N4e4kpunh8
- อัปโหลดไฟล์ขึ้น google drive ตัวเอง และเชื่อม google drive โหลดไฟล์ใน colab
- `from google.colab import drive`
- `import pandas as pd`
- `import os`
- `from datetime import datetime as dt`
- `from datetime import time`


2



เตรียมและแปลงข้อมูลบอกเวลาให้เป็นตัวแปรชนิด timestamp

- `drive.mount('/content/drive')`
- `path = '/content/drive/My Drive/dataviz_2024_data'`
- `data = pd.read_csv(os.path.join(path, 'search_request.csv'))`
- `data.head()`
- `data['search_timestamp'] =`
`pd.to_datetime(data['search_timestamp'], format='%Y-%m-%d`
`%H:%M:%S', errors='coerce')`


3



Quiz

- สร้างกราฟแท่งเปรียบเทียบปริมาณ คนเข้าใช้ web Wongnai.com เชื้อค้นหร้านอาหาร ในแต่ละวัน

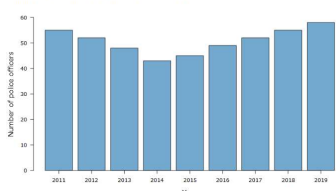
4




Bar chart (กราฟแท่ง)

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

Chart 5.2.1
Number of police officers in Crimeville, 2011 to 2019



5



การแสดงตัวอักษรภาษาไทยในกราฟ matplotlib

- ใช้วิธีจาก
- <https://medium.com/@kanyawee.work/%E0%B8%81%E0%B8%AA%E0%B8%94%E0%B8%87%E0%B8%9C%E0%B8%A5%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2%E0%B8%84%E0%B8%97%E0%B8%A2%E0%B8%83%E0%B8%99%E0%B8%81%E0%B8%A3%E0%B8%B2%E0%B8%9F%E0%B8%82%E0%B8%AD%E0%B8%B7-matplotlib-%E0%B8%9A%E0%B8%99-google-colab-37210d9a9f31>
- https://colab.research.google.com/drive/1sTdtZx_Cm51mc8OL_QhttehWyQ4725sGls#scrollTo=Ak95bVd5L7U8
- `!wget -q https://github.com/Phonbopit/sarabun-webfont/raw/master/fonts/thSarabunnew-webfont.ttf`
- `import matplotlib as mpl`
- `mpl.font_manager.fontManager.addfont('thSarabunnew-webfont.ttf')`
- `mpl.rc('font', family='TH Sarabun New')`

6

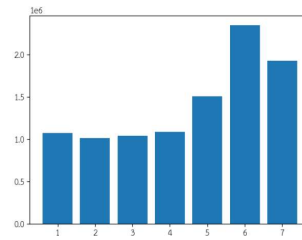
การสร้าง bar chart

- `from matplotlib import pyplot as plt`
 - โดยใช้คำสั่ง `plt.bar('list จำนวนแท่ง', 'list ข้อมูลที่ต้องการของแต่ละแท่ง')` เช่น
 - จากข้อมูล `search_request.csv` สร้างกราฟแท่งเปรียบเทียบปริมาณ คนเข้าใช้ web Wongnai.com เพื่อค้นหาร้านอาหาร ในแต่ละวัน
- ```
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]
])
```

6

7

## ผลลัพธ์ การสร้าง bar chart



7

8

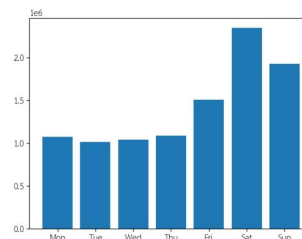
## Parameter: tick\_label (ชื่อแท่ง)

- ชื่อแท่งสามารถกำหนดได้โดยการใส่ Parameter: `tick_label=('list ชื่อแท่งที่ต้องการตั้งตามลำดับแท่ง')`
  - เช่น
- ```
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'])
```

8

9

ผลลัพธ์ Parameter: tick_label



9

10

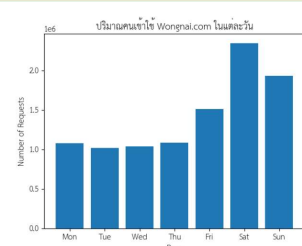
การกำหนดชื่อกราฟและชื่อแกน

- สามารถกำหนดได้โดยใช้คำสั่ง
 - `plt.xlabel('ชื่อแกนX')` และ `plt.ylabel('ชื่อแกนY')`
 - เช่น
- ```
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')
```

10

11

## ผลลัพธ์การกำหนดชื่อกราฟและชื่อแกน



11

12

### Grouped bar chart

- ใช้ในการสร้างกราฟแท่งที่ใช้เปรียบเทียบจำนวนข้อมูลในกลุ่มข้อมูลที่ต้องการ
- ขั้นตอนการสร้าง
- เตรียมข้อมูลที่ต้องการและเก็บค่าไว้ในตัวแปรของแต่ละกลุ่ม
- วิธีการสร้าง
- ตัวอย่าง แสดงปริมาณคนเข้าเว็บในแต่ละวัน โดยเปรียบเทียบช่วงเวลา 11:00-12:00 กับ 23:00-24:00

| Candidates | 2012 | 2013 |
|------------|------|------|
| Jessica    | 10   | 30   |
| William    | 25   | 20   |
| Hussein    | 55   | 45   |
| Peter      | 30   | 15   |
| Meridith   | 45   | 60   |

13

### ขั้นตอนการสร้าง Grouped bar chart

- เตรียมข้อมูลแต่ละกลุ่มที่ต้องการและเก็บค่าไว้ในตัวแปรเพื่อนำมาสร้างกราฟ
- กำหนดตัวแปร labels ใช้เก็บชื่อแท่งกราฟ
- นำตัวแปรที่เตรียมไว้มาสร้างกราฟ

14

### เตรียมข้อมูลแต่ละกลุ่ม

```

b4lunch = [data[data['search_timestamp'].dt.dayofweek == 0 & (data['search_timestamp'].dt.time < time(hour=12))] & data[data['search_timestamp'].dt.dayofweek == 1 & (data['search_timestamp'].dt.time < time(hour=12))] & data[data['search_timestamp'].dt.dayofweek == 2 & (data['search_timestamp'].dt.time < time(hour=12))] & data[data['search_timestamp'].dt.dayofweek == 3 & (data['search_timestamp'].dt.time < time(hour=12))] & data[data['search_timestamp'].dt.dayofweek == 4 & (data['search_timestamp'].dt.time < time(hour=12))] & data[data['search_timestamp'].dt.dayofweek == 5 & (data['search_timestamp'].dt.time < time(hour=12))] & data[data['search_timestamp'].dt.dayofweek == 6 & (data['search_timestamp'].dt.time < time(hour=12))]
b4midnight = [data[(data['search_timestamp'].dt.dayofweek == 0 & (data['search_timestamp'].dt.time > time(hour=23)))] & data[(data['search_timestamp'].dt.dayofweek == 1 & (data['search_timestamp'].dt.time > time(hour=23)))] & data[(data['search_timestamp'].dt.dayofweek == 2 & (data['search_timestamp'].dt.time > time(hour=23)))] & data[(data['search_timestamp'].dt.dayofweek == 3 & (data['search_timestamp'].dt.time > time(hour=23)))] & data[(data['search_timestamp'].dt.dayofweek == 4 & (data['search_timestamp'].dt.time > time(hour=23)))] & data[(data['search_timestamp'].dt.dayofweek == 5 & (data['search_timestamp'].dt.time > time(hour=23)))] & data[(data['search_timestamp'].dt.dayofweek == 6 & (data['search_timestamp'].dt.time > time(hour=23)))]
b4midnight
[31874, 32258, 31153, 35944, 53174, 58306, 35801]

```

15

### กำหนดตัวแปร labels ใช้เก็บชื่อแท่งกราฟ

```

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

```

16

### เตรียมตัวแปรที่ใช้สร้างกราฟ

```

import numpy as np
x = np.arange(len(labels)) กำหนดตำแหน่งแท่งกราฟในตัวแปร labels
width = 0.35 ความกว้างของกราฟ

fig, ax = plt.subplots() ใช้ subplots ในการวาดกราฟต่อกัน

```

17

### สร้างกราฟ

```

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

แท่งกราฟของข้อมูลตัวแปร b4lunch จะถูกสร้างก่อนตัวแปร b4midnight ตาม rects1, rects2

```

18

## กำหนดส่วนประกอบกราฟต่างๆ



```

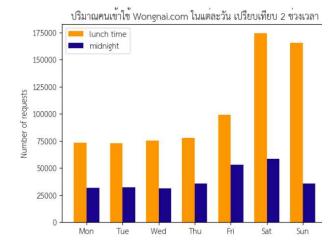
• ax.set_ylabel('Number of requests')
• ax.set_title('ปริมาณคนเข้าใช้ Wongnai.com ในแต่ละวัน เปรียบเทียบ 2 ช่วงเวลา')
• ax.set_xticks(x)
• ax.set_xticklabels(labels)
• ax.legend();

```

18

19

## ผลลัพธ์จะได้ Grouped bar chart ของ ปริมาณคนเข้าใช้ Wongnai.com ในแต่ละวัน เปรียบเทียบ 2 ช่วงเวลา



19

20