

## Class period 20

บทที่ 7 การแสดงผลการวัดแบบกระจาย (ค่า)  
Histogram and radar chart

1

## ตัวอย่างข้อมูล wongnai.com

- `import matplotlib.pyplot as plt`
- `output = plt.hist(list(data['number_of_result']),10,facecolor = 'red',alpha = 0.75)`
- สร้างกราฟ Histogram โดยใช้ข้อมูลที่มี 'number\_of\_result' ในข้อมูล wongnai.com โดยกำหนดค่าของ 'number\_of\_result' เป็น 10 ค่าและสีเป็นสีแดง กำหนดความโปร่งแสงเป็น 75%

2



3

## แก้ไข แกน x ที่เรียงข้อมูลผิด

- 1 ตรวจสอบ data type ของ ตัวแปร ตัว
- `data.dtypes`
- 2 เปลี่ยนค่าของ data type ของตัวแปรในกรณีที่ผิดด้วย
- `type(data['number_of_result'])`
- `type(data['number_of_result']) == int`
- 3 ตรวจสอบเปลี่ยน type ข้อมูลเป็น int
- `new_type = data['number_of_result'].astype('int32')`

4

## ลบ record

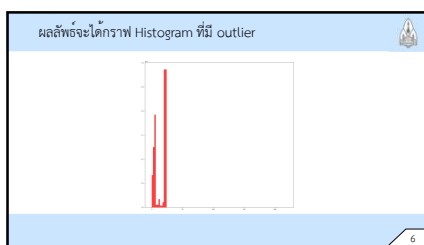
- จากการดูที่ค่าของ type ข้อมูลเป็น int หลังจากดู record ที่ 1000016 แล้ว พบว่ามีข้อมูลที่เป็นค่าที่ไม่ใช่ตัวเลข
- 1 ตรวจสอบว่า record ใดบ้างที่มีค่าเป็น number\_of\_result
- `data[data['number_of_result']!= 'number_of_result']`
- ผลลัพธ์ได้ record ที่ 1000016
- 2 ลบ record ที่ค่าเป็น number\_of\_result
- `data = data.drop(1000016)`

5

## บังคับเปลี่ยน type ข้อมูลเป็น int

- เปลี่ยนเปลี่ยน type ข้อมูลเป็น int หลังจากลบข้อมูล record ที่ 1000016 แล้ว พบว่ามีข้อมูลที่เป็นค่าที่ไม่ใช่ตัวเลข
- `new_type = data['number_of_result'].astype('int32')`
- สร้างกราฟ Histogram ด้วยข้อมูลในตัวแปร new\_type
- `output = plt.hist(new_type,100,facecolor = 'red',alpha = 0.75)`

6

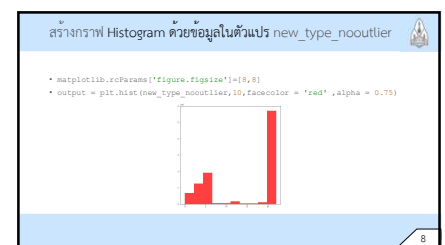


7

## ลบ outlier

- ลบ outlier โดยการลบค่าที่เกินค่าที่กำหนดไว้ที่ค่า 25 และไปลบค่าที่เกิน
- `new_type_nooutlier = new_type[new_type < 25]`
- ตรวจสอบว่าข้อมูลในตัวแปร new\_type\_nooutlier มีค่า
- `new_type_nooutlier.shape[0]`
- ผลลัพธ์ได้มีค่าเป็น 14 ตัว
- `new_type.shape[0] - new_type_nooutlier.shape[0]`
- 14

8



9

### Quiz

- เปรียบเทียบความถี่ของแพคเกจตามกลุ่ม กับ แพคเกจรวมกัน

10

### เฉลย

- output ตัวที่ 1 เป็นแพคเกจกราฟเป็นกราฟ จากผลลัพธ์ที่ได้คือค่าความถี่แพคเกจคือ 5717238  

```
(array([ 676289., 1247289., 192144., 51783., 58669., 167982.,
        36882., 39934., 105490., 5717238.]),
 array([ 0., 12.5, 4.4, 6.4, 8.4, 11., 15.2, 19.4, 17.4, 18.4, 22. ]),
 or list of 10 Patch objects)
```
- output[0] เป็นค่าของ array ตัวที่ 0 ของ output คือค่าความถี่ของแพคเกจ  

```
array([ 676289., 1247289., 192144., 51783., 58669., 167982.,
        36882., 39934., 105490., 5717238.])
```

11

### เฉลยต่อ

- output[0][:-1] ค่าความถี่ของแพคเกจทุกตัว  

```
5717238.0
```
- sum(output[0][:-1]) ค่าความถี่ของแพคเกจรวมกัน  

```
4287104.0
```

12

### Tree map

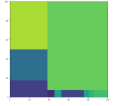
- การนำรูปแบบข้อมูลที่มีลักษณะเป็นลำดับชั้นมาจัดเรียงเป็นกราฟแบบ tree map
- สามารถใช้งานจาก tree map ด้วย package squarify โดยต้อง install package ก่อนใช้งานด้วยคำสั่ง  

```
!pip install squarify
```
- import squarify
- import numpy as np
- import matplotlib.pyplot as plt

13

### การใช้งาน squarify


- squarify.plot('ชื่อไฟล์ข้อมูลที่จะทำการทำ tree map') เช่น
- squarify.plot(output[0])



14

### Parameter: value ของ squarify

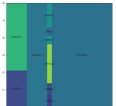
- ไม่ Parameter: value จะใช้ค่าในลำดับชั้นเป็นค่าของ tree map เช่น
- squarify.plot(output[0], value=output[0])



15

### Parameter: norm\_y ของ squarify


- ไม่ Parameter: norm\_y ในการใช้ package นี้จะทำการปรับขนาดของ tree map ตามพื้นที่กราฟที่กำหนด เช่น
- squarify.plot(output[0], value=output[0], norm\_y=60)



16

### Radar Chart

- เป็นกราฟที่แสดงการเปรียบเทียบข้อมูลในหลายๆมิติโดยรอบจุดศูนย์กลาง โดยแต่ละแกนจะแสดงค่าของข้อมูลในแต่ละมิติ



17