

## Class period 11

บทที่ 6 การแสดงผลการกระจายข้อมูล  
Visualize\_Table\_Data

1

## ทบทวน parallel\_coordinates

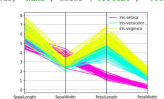
- ดูว่า google colab โหลด สามารถรันโค้ดได้เลย
- import pandas as pd
- df = pd.read\_csv("https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/data/csv/iris.csv")
- df
- pd.plotting.parallel\_coordinates(df, "Name")



2

## ปรับ parameter: color

- การเลือกสีตามการแบ่ง สามารถกำหนดสีตามชื่อของ feature โดยใช้ parameter: color ตามด้วย ชื่อ feature หรือใช้ hex code ของสี
- ตัวอย่างสี: RGB value (r,g,b) ที่ต้องการกำหนดให้แต่ละกลุ่ม เช่น hex code color
- pd.plotting.parallel\_coordinates(df, "Name", color=("#FFD700", "#00FFB2", "#0000FF"))
- สีตามชื่อของ feature เปลี่ยนเป็นตามที่กำหนด

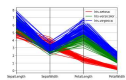


3

## ตัวอักษรสี

- ตัวอย่างสี จะสามารถใช้กับ feature ที่มีค่าเป็นตัวเลขได้
- R G B W C M Y K

pd.plotting.parallel\_coordinates(df, "Name", color=('r', 'g', 'b'))

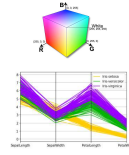


4

## RGB value (r,g,b)

- สามารถกำหนดสีตามค่า RGB ค่าเป็นตัวเลขที่กำหนดได้ 3 สี (ความเข้มของสี) เปลี่ยนเป็นสีที่ต้องการ
- ค่าแต่ละสี 3 สี คือ r,g,b ค่าแต่ละสี RGB จะมีค่า 0 ถึง 1 เช่น

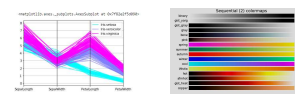
pd.plotting.parallel\_coordinates(df, "Name", color=((1,0.5,0), (0.6,0.7,0), (0.6,0,1)))



5

## ปรับ parameter: colormap

- เป็นการเลือกชุดสีที่กำหนดไว้ให้ดูเป็นภาพแล้ว เพื่อการที่ดูง่ายและดูสวย สามารถเลือกได้โดยการเข้าไปดู Choosing Colormaps in Matplotlib ใน google เช่น
- pd.plotting.parallel\_coordinates(df, "Name", colormap="cool")



6

## Matplotlib

- เป็น package หรือ library ที่ช่วยในการ Visualization ในการทำการต่างๆ
- มีฐานข้อมูล
- import matplotlib
- from matplotlib import pyplot as plt
- matplotlib จะทำงานร่วมกับ numpy array ไม่ใช้ว่า pandas เพราะ matplotlib เป็น numpy array ที่เชื่อมกับ pandas
- การที่ pandas มี matplotlib เป็นส่วนหนึ่งของ pandas ทำให้สามารถนำข้อมูลจาก pandas มาใช้กับ matplotlib ได้โดยไม่ต้องแปลงเป็น numpy array
- ตัวแปรที่ pandas ใช้ในการทำงานกับ numpy array คือ np.array

7

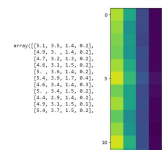
## แปลง data จาก pandas dataframe เป็น numpy array

- ใช้ np.array() ในการแปลงข้อมูลจาก pandas dataframe ไปเป็น numpy array
- np\_data = df.iloc[1:100,1:3].to\_numpy()
- ไม่จำเป็นต้องแปลงข้อมูลจาก pandas dataframe เป็น numpy array เสมอไป ขึ้นอยู่กับความต้องการในการใช้งาน
- np\_data แปลงข้อมูลจาก pandas dataframe ไปเป็น numpy array

8

## Visualize array data ด้วย pixel

- เป็นการแสดงข้อมูลใน numpy array ด้วย pixel
- สามารถแสดงภาพด้วย pixel
- สามารถใช้ความรู้จากความรู้ที่เรียนมา
- หาข้อมูลจาก slide 11 แล้วมาดู np\_data



9

### วิธีการใช้งาน Visualize array data ด้วย pixel

- ขั้นตอนที่ 1 ทำงานใน memory ใช้กำหนดขนาดของรูปภาพ ที่กำหนด สีของภาพ กว้าง 15 พิกเซล สูง 15 พิกเซล
- `matplotlib.rcParams["figure.figsize"]=[15,15]`
- ขั้นตอนที่ 2 ใช้คำสั่ง `plt.imshow` เพื่อแสดงรูปภาพ
- `plt.imshow(np_data[1,:],interpolation="nearest")`
- `npud` เป็นตัวแปรที่ใช้เก็บข้อมูล ขนาดของ array สามารถเป็นได้หลายมิติ การแสดงผลการแปลงรูปภาพได้
- `plt.imshow` จะแสดงออกมาในลักษณะเป็นรูปภาพ
- `imshow` และ `plot` เป็นฟังก์ชันและวิธีการในการแสดงรูปภาพ

10

### การใช้ plt.subplot()

- ใช้ในการแสดงรูปภาพมากกว่า 1 รูป ซึ่งเป็นการแสดงรูปภาพหลายภาพ
- ในการใช้ `plt.subplot(abc)` สามตัว `plt.subplot(abc)` หมายความว่า `abc` เป็นค่าที่ระบุ `parameter` ของกราฟที่ต้องการจะนำในบรรทัดต่อไป
- `a` คือ `parameter` ที่ใช้กำหนดแถว
- `b` คือ `parameter` ที่ใช้กำหนดหลัก
- `c` คือ `parameter` ที่ใช้กำหนดว่าแสดงรูปภาพทั้งหมดกี่ภาพ
- `plt.subplot(1,3,1)` `plt.subplot(2,3,5)`

11

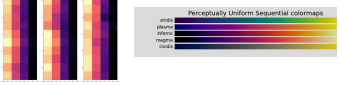
### ตัวอย่างการใช้ plt.subplot()

- `plt.subplot(1,3,1)`
- `plt.imshow(np_data[10,:],interpolation="nearest", cmap="magma")`
- `plt.subplot(1,3,2)`
- `plt.imshow(np_data[10:20,:],interpolation="nearest", cmap="magma")`
- `plt.subplot(2,3,3)`
- `plt.imshow(np_data[20:30,:],interpolation="nearest", cmap="magma")`

12

### Parameter: cmap

- `cmap` เป็น colormap ที่กำหนดให้กับสี ซึ่งสามารถที่จะนำข้อมูลมาใส่แล้วเป็นการที่สีรวมและดูง่าย

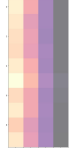


13

### Parameter: alpha

- สีสามารถที่จะแสดงได้ 0 ถึง 1 โดย 1 คือไม่โปร่งแสง

- `plt.imshow(np_data[10,:],interpolation="nearest", cmap="magma", alpha = 0.5)`



14