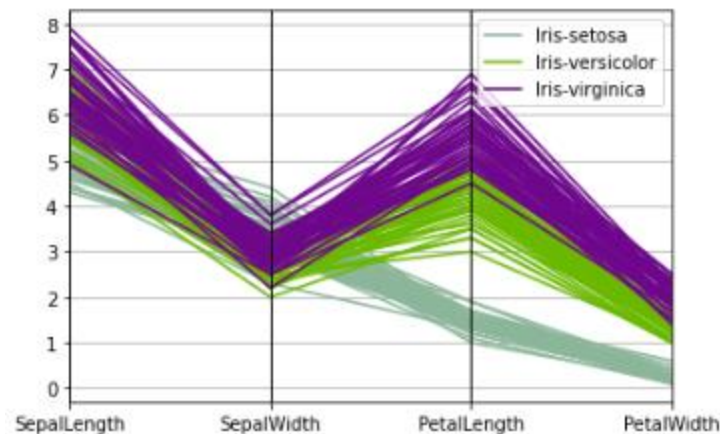


Class period 11

Visualize_Table_Data

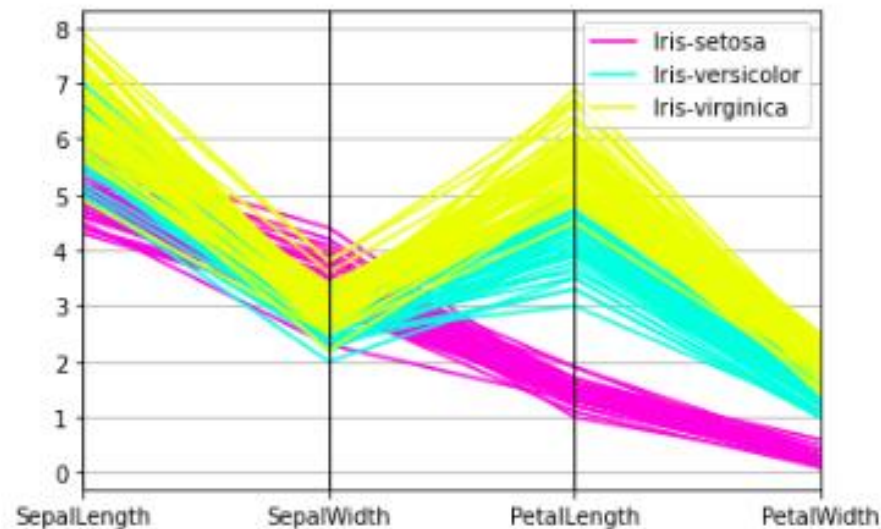
ทบทวน parallel_coordinates

- สร้าง google colab ใหม่ ดาวน์โหลดข้อมูลดอกไม้ iris
- `import pandas as pd`
- `df = pd.read_csv('https://raw.githubusercontent.com/pandas-dev/pandas/master/pandas/tests/io/data/csv/iris.csv')`
- `df`
- `pd.plotting.parallel_coordinates(df, 'Name');`



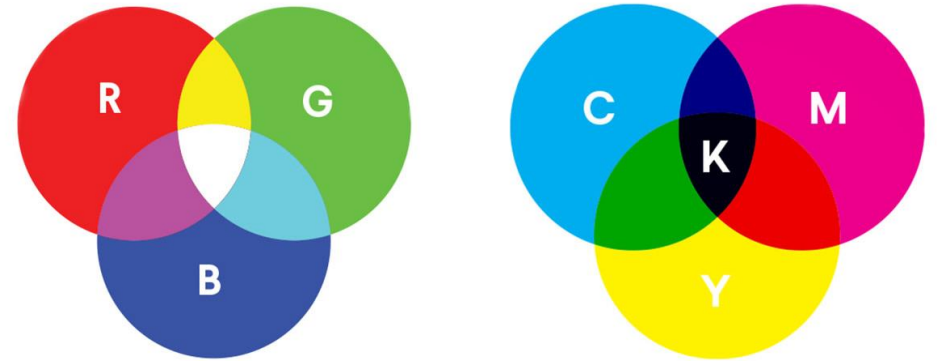
ปรับ parameter: color

- การจัดกลุ่มตามการแบ่งสี สามารถกำหนดสีได้ตามที่ต้องการ โดยการใส่ **parameter: color** ตามด้วย **list** โค้ดค่าสี **html color**, ตัวย่อของสี, **RGB value (r,g,b)** ที่ต้องการกำหนดให้แต่ละกลุ่ม เช่น **html color**
- `pd.plotting.parallel_coordinates(df, 'Name', color=('#FF00E0', '#00FFE2', '#ECFF00'))`
- สีเส้นของแต่ละกลุ่มจะเปลี่ยนไปตามที่กำหนด

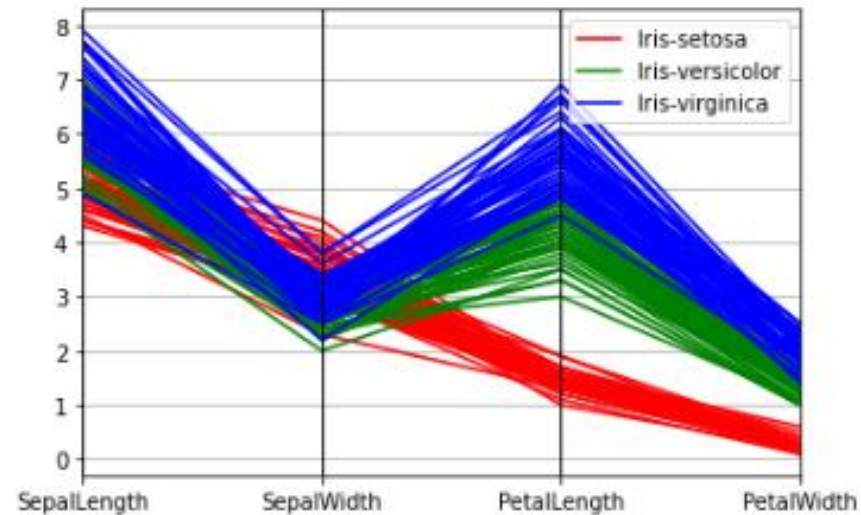


ตัวย่อของสี

- ตัวย่อของสี จะสามารถใช้กำหนดสีได้เฉพาะสีที่เป็นแม่สี
- R G B W C M Y K

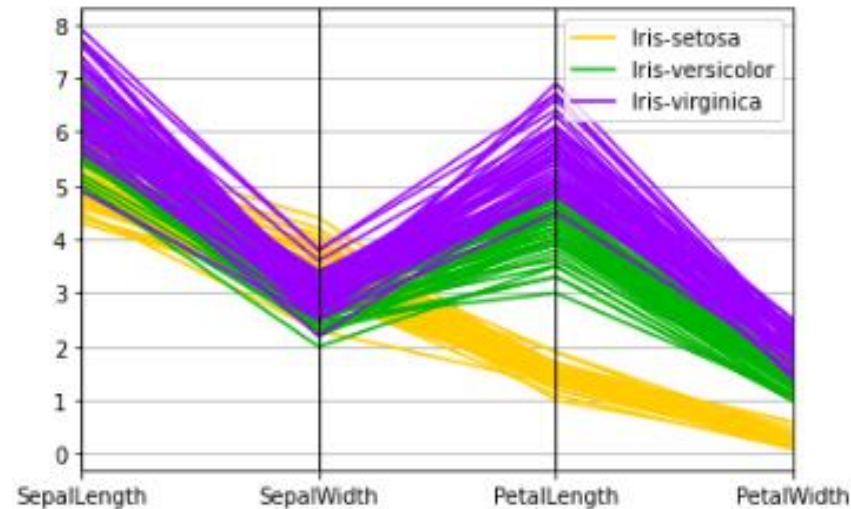
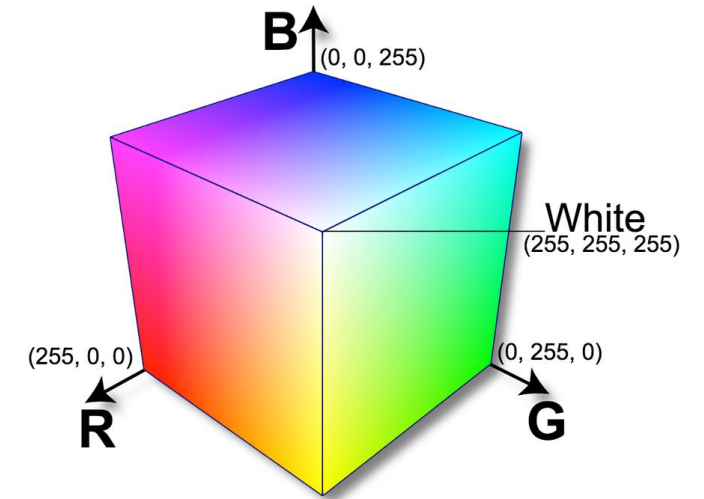


- `pd.plotting.parallel_coordinates(df, 'Name', color=('r', 'g', 'b'))`



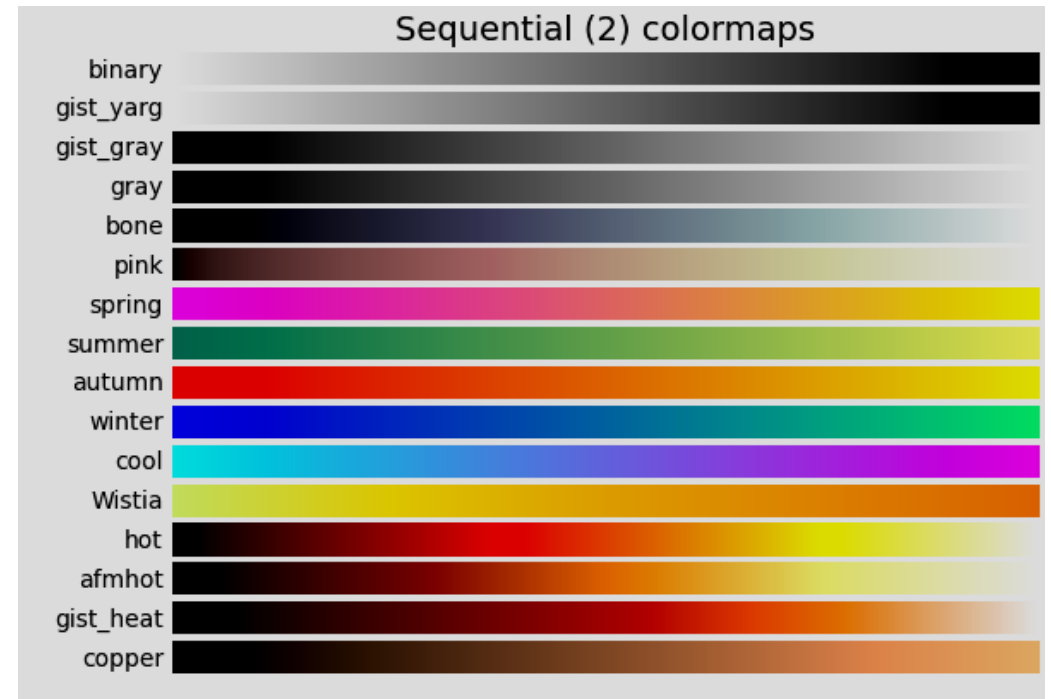
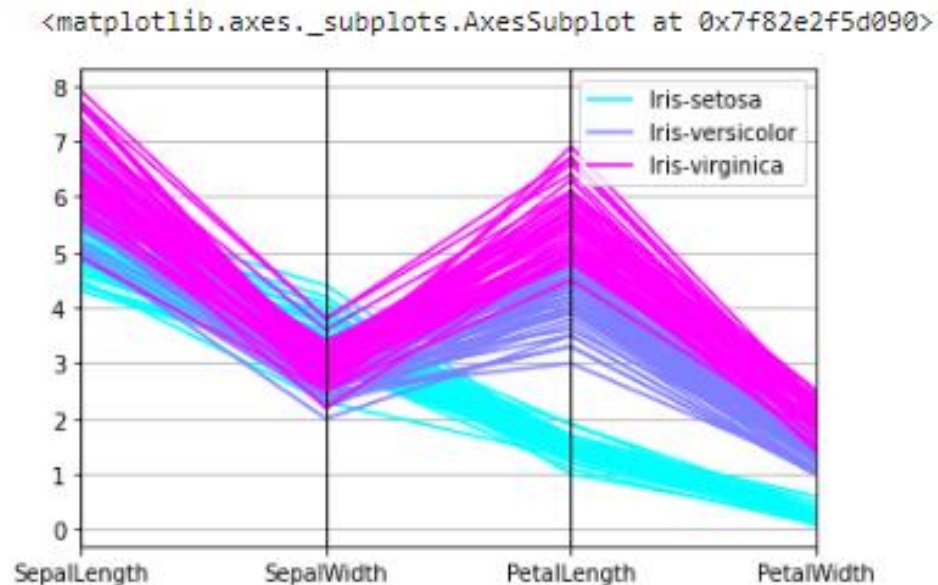
RGB value (r,g,b)

- สามารถกำหนดสีที่ต้องการด้วยค่า RGB ทำงานโดยการกำหนดระดับค่าแสง 3 ตัว (ความเข้มความอ่อนของสี) เพื่อให้ได้สีที่ต้องการ
- ค่าแสง มี 3 ตัว คือ r, g, b ซึ่งค่าแสง RGB จะมีค่า 0 ถึง 1 เช่น
- `pd.plotting.parallel_coordinates(df, 'Name', color=((1, 0.8, 0), (0, 0.7, 0), (0.6, 0, 1)))`



ปรับ parameter: colormap

- เป็นการเลือกชุดสีที่ทางผู้เชี่ยวชาญได้จัดชุดสีมาให้แล้ว เพื่อกราฟที่สวยงามและดูง่าย สามารถเลือกได้โดยการเข้าไปดู **Choosing Colormaps in Matplotlib** ใน google เช่น
- `pd.plotting.parallel_coordinates(df, 'Name', colormap='cool')`



Matplotlib

- เป็น **packet** หรือ **library** พื้นฐานที่ใช้ **Visualization** ในการวาดกราฟต่างๆ
- ใช้งานโดย
 - `import matplotlib`
 - `from matplotlib import pyplot as plt`
- **Matplotlib** จะทำงานเข้ากับข้อมูลที่เป็น **numpy array** ได้ดีกว่า **pandas** เพราะ **matplotlib** กับ **numpy array** ทั้งสองเป็นวิธีพื้นฐานเหมือนกัน ถ้าเป็นข้อมูล **pandas** การทำงานของ **matplotlib** ก็ทำงานได้แต่อาจจะไม่รู้จักรายการข้อมูลของ **pandas** บางประเภท และ **matplotlib** ทำงานได้ดีกับข้อมูลที่เป็นตัวเลข
- ดังนั้น ควรแปลงข้อมูลจากรายการ **pandas** ให้อยู่ในรูปแบบ **numpy array** ก่อนใช้งาน **matplotlib**

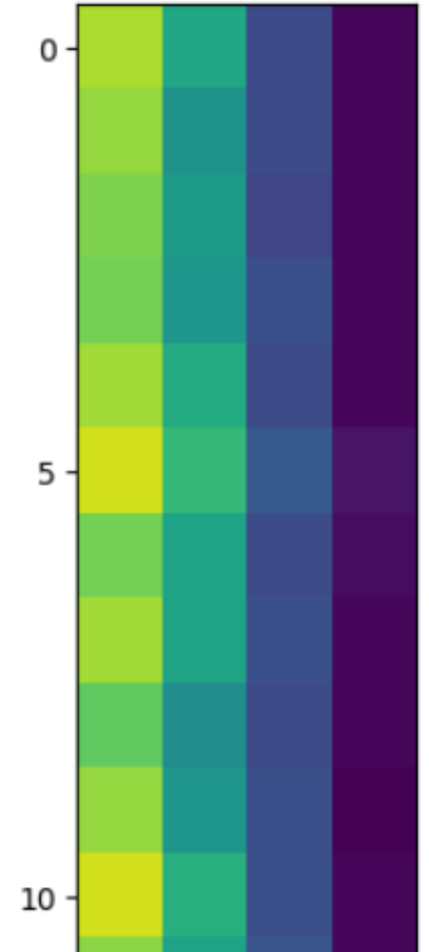
แปลง data จาก pandas dataframe เป็น numpy array

- ใช้ `.to_numpy()` ในการแปลงข้อมูลตาราง pandas ที่ต้องการ โดยกำหนด คอลัมน์และแถวที่ต้องการแปลงด้วย `iloc` เช่น
- `np_data = df.iloc[:, :-1].to_numpy()`
- ให้แปลงข้อมูลตารางในตัวแปร `df` เลือกเฉพาะข้อมูลในคอลัมน์ที่เป็นตัวเลข คือตั้งแต่แถวที่ 0 ถึงแถวสุดท้ายและคอลัมน์ที่ 0 ถึงคอลัมน์รองสุดท้าย (คอลัมน์ที่ 3)
- `np_data` จะเห็นข้อมูลที่เป็นตัวเลขของข้อมูลตารางในตัวแปร `df` ถูกแปลงเป็น `numpy array`

Visualize array data ด้วย pixel

- เป็นการแทนค่าตัวเลขใน **numpy array** ด้วยจุดสี **pixel**
- ค่ามากสีจะสว่าง ค่าน้อยสีจะทึบ
- สามารถใช้ดูความแตกต่างของข้อมูลด้วยสี
- ยกตัวอย่างข้อมูล **11** แถวแรกของ **np_data**

```
array([[5.1, 3.5, 1.4, 0.2],  
       [4.9, 3. , 1.4, 0.2],  
       [4.7, 3.2, 1.3, 0.2],  
       [4.6, 3.1, 1.5, 0.2],  
       [5. , 3.6, 1.4, 0.2],  
       [5.4, 3.9, 1.7, 0.4],  
       [4.6, 3.4, 1.4, 0.3],  
       [5. , 3.4, 1.5, 0.2],  
       [4.4, 2.9, 1.4, 0.2],  
       [4.9, 3.1, 1.5, 0.1],  
       [5.4, 3.7, 1.5, 0.2],
```

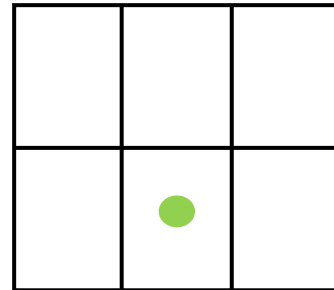
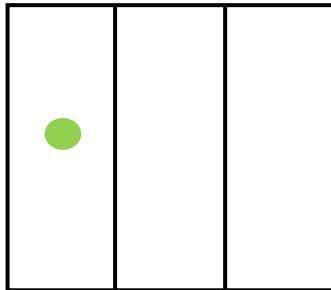


วิธีการใช้งาน Visualize array data ด้วย pixel

- `matplotlib.rcParams['figure.figsize']=[15,15]`
- ขั้นตอนที่ 1 ทำงานใน memory ใช้กำหนดขนาดของรูปภาพ กว้าง * ยาว ตัวอย่าง กว้าง 15 ยาว 15
- `plt.imshow(np_data[:, :], interpolation='nearest')`
- ขั้นตอนที่ 2 ใช้คำสั่ง `plt.imshow` เพื่อแสดงรูปภาพ
- input เป็นตัวแปรที่ใช้เก็บข้อมูล numpy array สามารถเลือกแถวและหลักที่ต้องการให้แสดงรูปภาพได้
- parameter: `interpolation` ใช้สำหรับเลือกรูปแบบของการไล่สีในรูปภาพ
- `'nearest'` แสดง pixel สีเป็นสีเหลี่ยมและช่วยให้การแยกสีในรูปภาพชัด

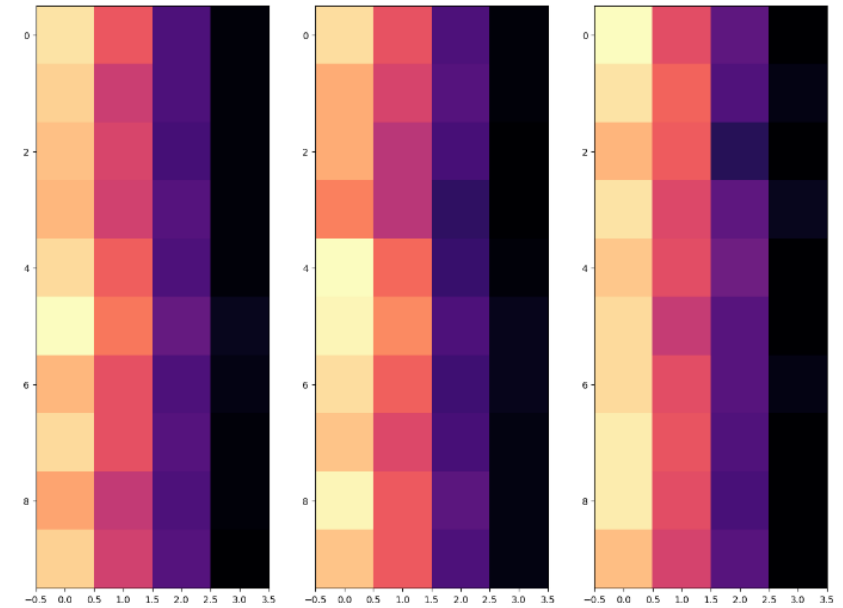
การใช้ subplot

- ใช้ในการแสดงรูปภาพมากกว่า 1 กราฟเพื่อเปรียบเทียบดูความแตกต่างระหว่างกราฟ
- โดยการใช้งาน `plt.subplot(a, b, c)` ตามด้วย `plt.imshow` ตัวที่ 1 ในบรรทัดถัดไป
- a คือ parameter ที่ใช้กำหนดแถว
- b คือ parameter ที่ใช้กำหนดหลัก
- c คือ parameter ที่ใช้กำหนดลำดับรูปภาพที่ต้องการแสดง เช่น
- `plt.subplot(1, 3, 1)` `plt.subplot(2, 3, 5)`



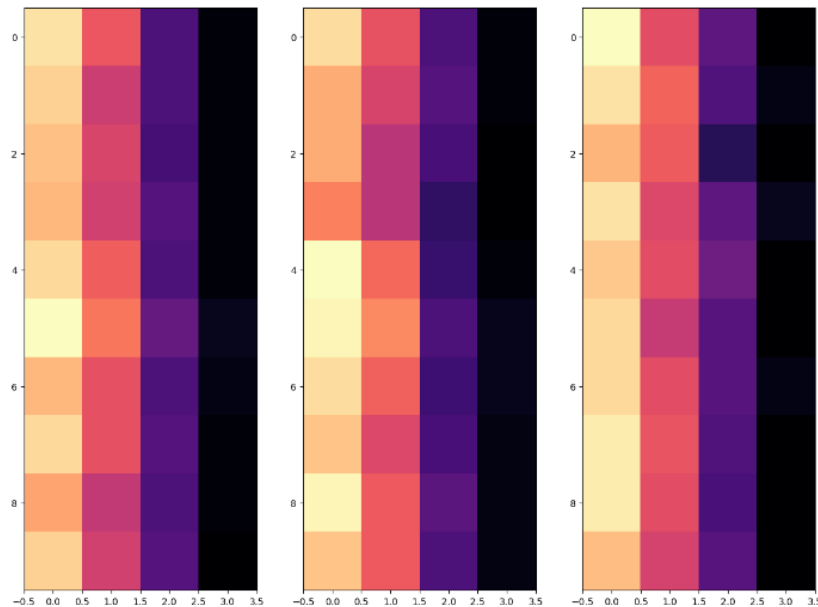
ตัวอย่างการใช้ subplot

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



Parameter: cmap

- **cmap** คือ **colormap** ทำงานแบบเดียวกัน คือเลือกชุดสีที่ทางผู้เชี่ยวชาญได้จัดชุดสีมาให้แล้ว เพื่อกราฟที่สวยงามและดูง่าย



Parameter: alpha

- คือค่าความโปร่งแสงของสี มีค่า **0** ถึง **1** โดย **1** คือไม่โปร่งแสง
- `plt.imshow(np_data[:10, :],
interpolation='nearest',
cmap='magma', alpha = 0.5)`

