



Class period 13

บทที่ 6 การแสดงผลการกระจายของข้อมูล (ต่อ)

Visualize_Data_Distribution part2

quiz3 (5 นาที)

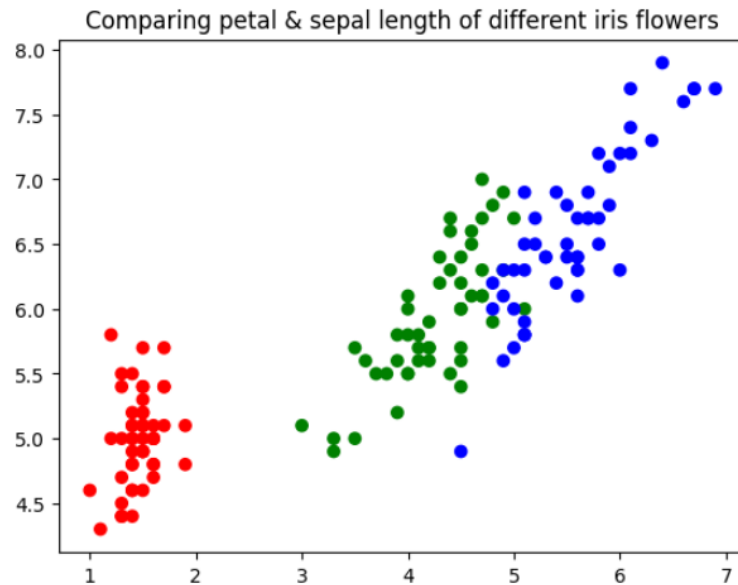


- โหลดตาราง ตค57 กับ ตาราง พย57
- จากไฟล์ จัดซื้อ ตค57-ตค58
- จาก https://data.go.th/dataset/item_fadc318e-7743-4bc2-84d1-a19210a9bbd6

การเพิ่มชื่อกราฟ

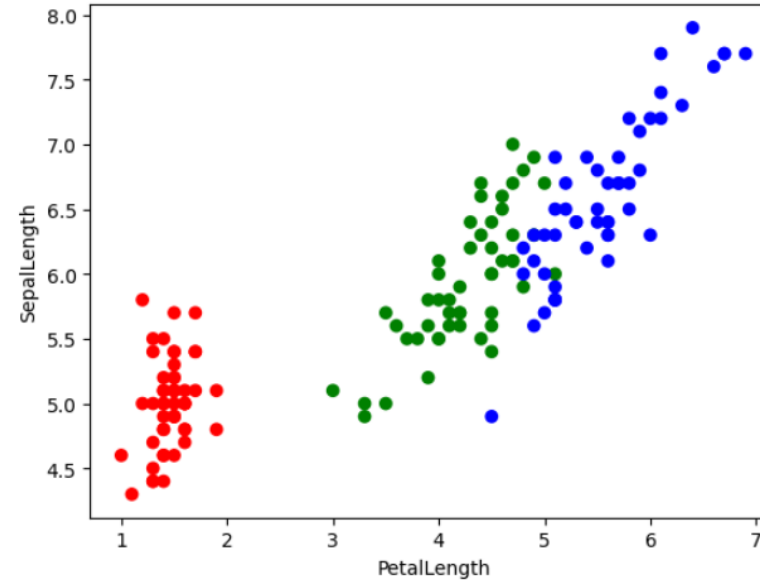


- สามารถตั้งชื่อกราฟตามที่ต้องการได้โดยการใช้คำสั่ง `plt.title()` เช่น
- `plt.scatter(df2['PetalLength'], df2['SepalLength'], c=df2['Name'])`
- `plt.title('Comparing petal & sepal length of different iris flowers');`



การเพิ่มชื่อแกนในกราฟ

- สามารถตั้งชื่อแกนในกราฟตามที่ต้องการได้โดยใช้คำสั่ง `plt.xlabel()` และ `plt.ylabel()` เช่น
- `plt.scatter(df2['PetalLength'], df2['SepalLength'], c=df2['Name'])`
- `plt.xlabel('PetalLength')`
- `plt.ylabel('SepalLength')`





การแยกข้อมูลเพื่อกำหนดลักษณะ marker แต่ละกลุ่ม

- โดยการแยกข้อมูลเพื่อกำหนดลักษณะ marker แต่ละกลุ่ม จำเป็นที่จะต้องรู้ลำดับและจำนวน record ของแต่ละกลุ่มก่อน เช่น
- ในข้อมูลดอกไม้อิริส มีอยู่ 3 สายพันธุ์ คือ
- Iris-setosa, Iris-versicolor และ Iris-verginiga
- เมื่อรู้ข้อมูลลำดับและจำนวน record แต่ละสายพันธุ์ สามารถกำหนด record ที่ต้องการแยกได้
- Iris-setosa คือ [:50] ข้อมูล record แรกถึง record ที่ 49
- Iris-versicolor คือ [50:100] ข้อมูล record ที่ 50 ถึง record ที่ 99
- Iris-verginiga คือ [100:] ข้อมูล record ที่ 99 ถึง record สุดท้าย



ตัวอย่างการใช้การแยกข้อมูลเพื่อ plot กราฟซ้อนกัน

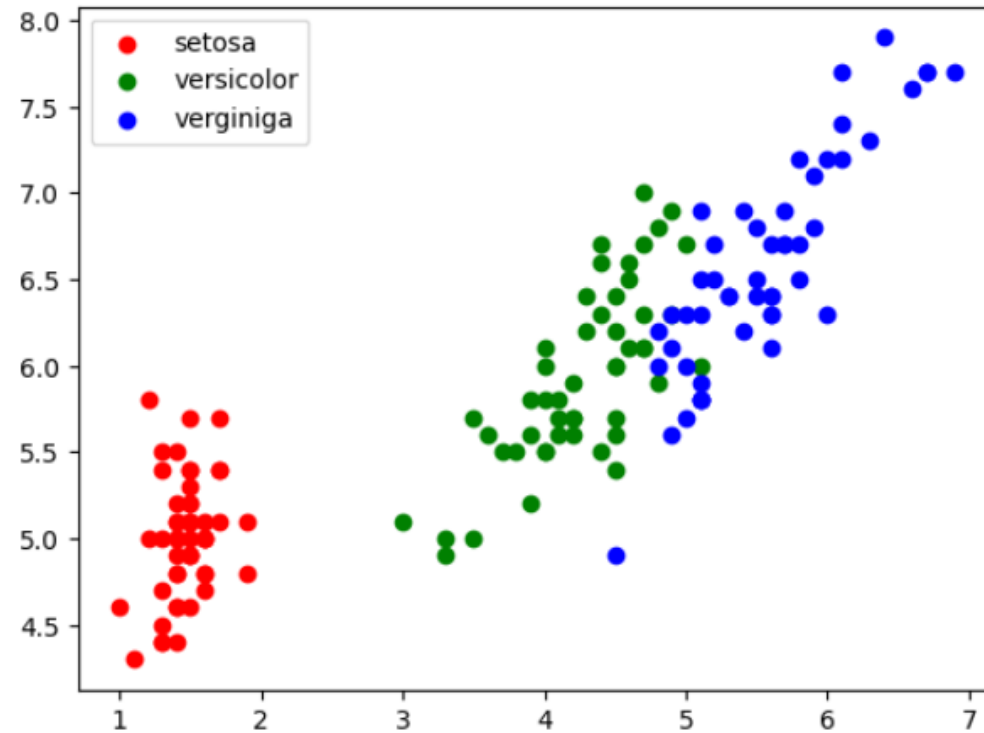
- `plt.scatter(df2['PetalLength'][:50], df2['SepalLength'][:50],
c=df2['Name'][:50])`
- `plt.scatter(df2['PetalLength'][50:100], df2['SepalLength'][50:100],
c=df2['Name'][50:100])`
- `plt.scatter(df2['PetalLength'][100:], df2['SepalLength'][100:],
c=df2['Name'][100:])`



Parameter: label และคำสั่ง plt.legend() ใช้กำหนดชื่อและแสดงชื่อ marker แต่ละกลุ่ม

- สามารถกำหนดชื่อ marker แต่ละกลุ่มได้ โดยการใส่ Parameter: label เพิ่มในข้อมูลที่แยกตามลำดับและจำนวน record ของแต่ละกลุ่ม
- และใช้คำสั่ง plt.legend() เพื่อแสดงชื่อบนรูปภาพ เช่น
- `plt.scatter(df2['PetalLength'][:50], df2['SepalLength'][:50], c=df2['Name'][:50], label='setosa')`
- `plt.scatter(df2['PetalLength'][50:100], df2['SepalLength'][50:100], c=df2['Name'][50:100], label='versicolor')`
- `plt.scatter(df2['PetalLength'][100:], df2['SepalLength'][100:], c=df2['Name'][100:], label='virginica')`
- `plt.legend()`

ผลลัพธ์ตัวอย่าง กำหนดชื่อและแสดงชื่อ marker แต่ละกลุ่ม





Parameter: s ใช้ขนาดของ marker แสดงค่าของ feature

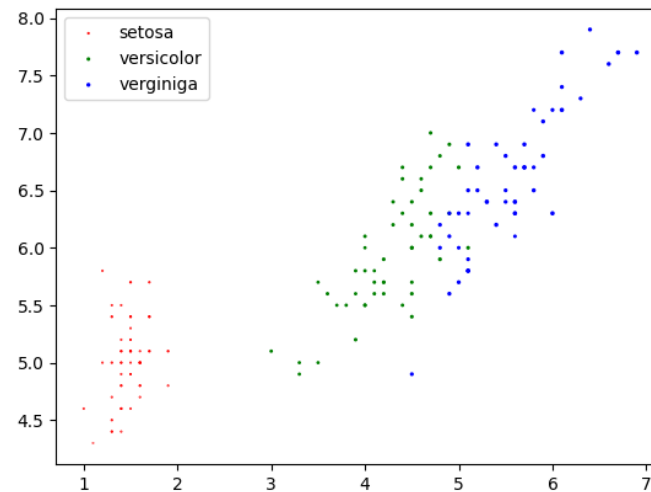
สามารถกำหนดขนาดของ marker โดยการใส่และกำหนด Parameter: s= nomination ที่ 3 สำหรับดูการกระจายของข้อมูลระหว่าง 3 nomination (คอลัมน์) โดย nomination ที่ 3 จะถูกเปรียบเทียบในรูปแบบขนาด ค่ามาก marker จะขนาดใหญ่ ค่าน้อย marker จะขนาดเล็ก เช่น

- ตัวอย่างการใช้งาน Parameter: s ต้องการเพิ่มคอลัมน์ PetalWidth เป็น nomination ที่ 3
- `plt.scatter(df2['PetalLength'][:50], df2['SepalLength'][:50], c=df2['Name'][:50], label='setosa', s=df2['PetalWidth'][:50])`



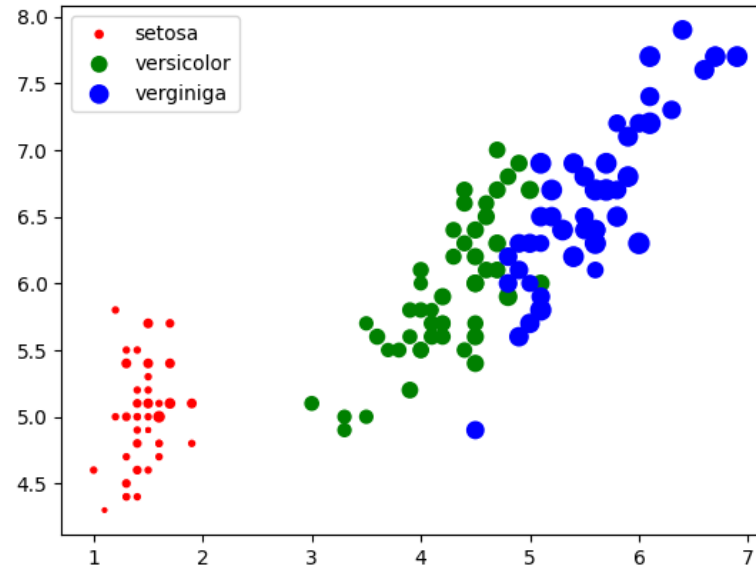
ผลลัพธ์ตัวอย่างการใช้งาน Parameter: s

- `plt.scatter(df2['PetalLength'][:50], df2['SepalLength'][:50], c=df2['Name'][:50], label='setosa', s=df2['PetalWidth'][:50])`
- `plt.scatter(df2['PetalLength'][50:100], df2['SepalLength'][50:100], c=df2['Name'][50:100], label='versicolor', s=df2['PetalWidth'][50:100])`
- `plt.scatter(df2['PetalLength'][100:], df2['SepalLength'][100:], c=df2['Name'][100:], label='virginica', s=df2['PetalWidth'][100:])`
- `plt.legend()`



การเพิ่มขนาด marker ที่เล็กเกินไป

- สามารถใช้ตัวเลขคูณเข้าไปในข้อมูลผ่านตัวแปร nomination ที่ 3 ใน Parameter: s เพื่อเพิ่มขนาดได้เลย เช่น
- `s=40*df2['PetalWidth'][50:100]`

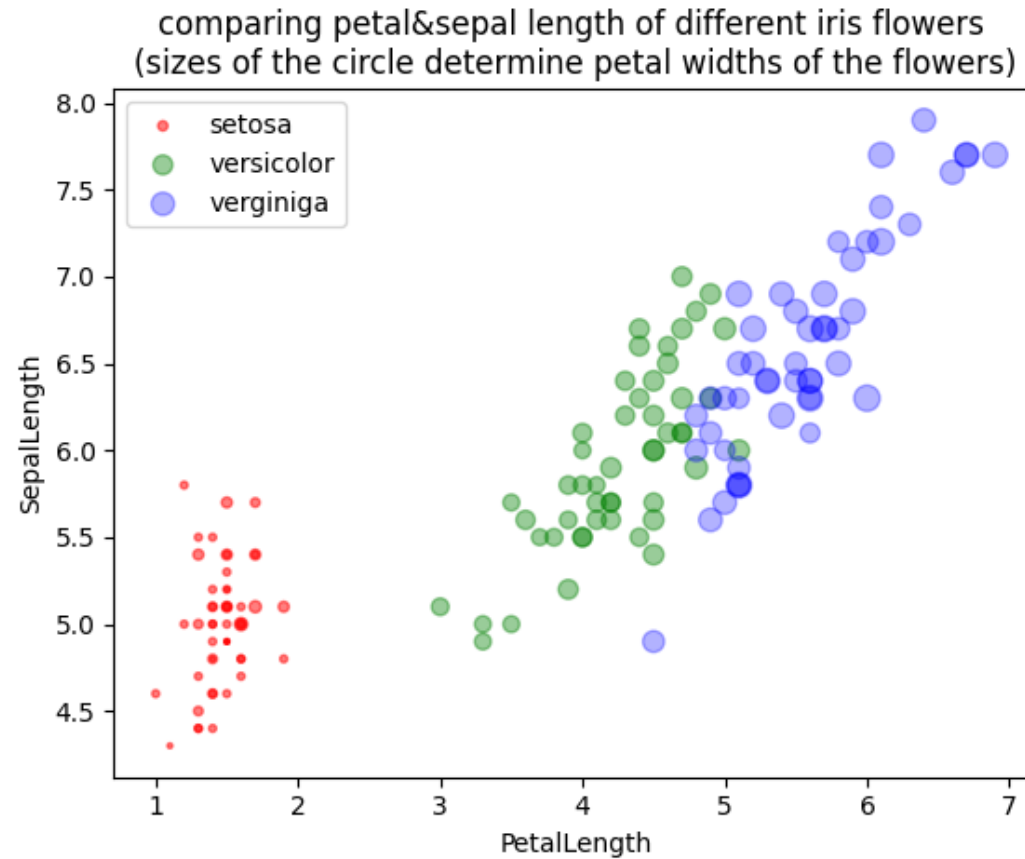




ตัวอย่างการใช้งาน plt.scatter() และ parameter

- `plt.scatter(df2['PetalLength'][:50], df2['SepalLength'][:50], alpha=0.5, c=df2['Name'][:50], label='setosa', s=40*df2['PetalWidth'][:50])`
- `plt.scatter(df2['PetalLength'][50:100], df2['SepalLength'][50:100], alpha=0.4, c=df2['Name'][50:100], label='versicolor', s=40*df2['PetalWidth'][50:100])`
- `plt.scatter(df2['PetalLength'][100:], df2['SepalLength'][100:], alpha=0.3, c=df2['Name'][100:], label='virginica', s=40*df2['PetalWidth'][100:])`
- `plt.legend()`
- `plt.xlabel('PetalLength')`
- `plt.ylabel('SepalLength')`
- `plt.title('comparing petal&sepal length of different iris flowers \n(sizes of the circle determine petal widths of the flowers)');`

ผลลัพธ์ตัวอย่างการใช้งาน plt.scatter() และ parameter



Parameter: marker เปลี่ยนหน้าตาของ Marker

- สามารถเปลี่ยนหน้าตา marker ได้ตามที่ต้องการเพื่อความสวยงาม โดยการใส่และกำหนด Parameter: marker
- สัญลักษณ์หน้าตา marker ต่างๆ สามารถหาได้จาก
- https://matplotlib.org/stable/api/markers_api.html#module-matplotlib.markers
- ตัวอย่าง

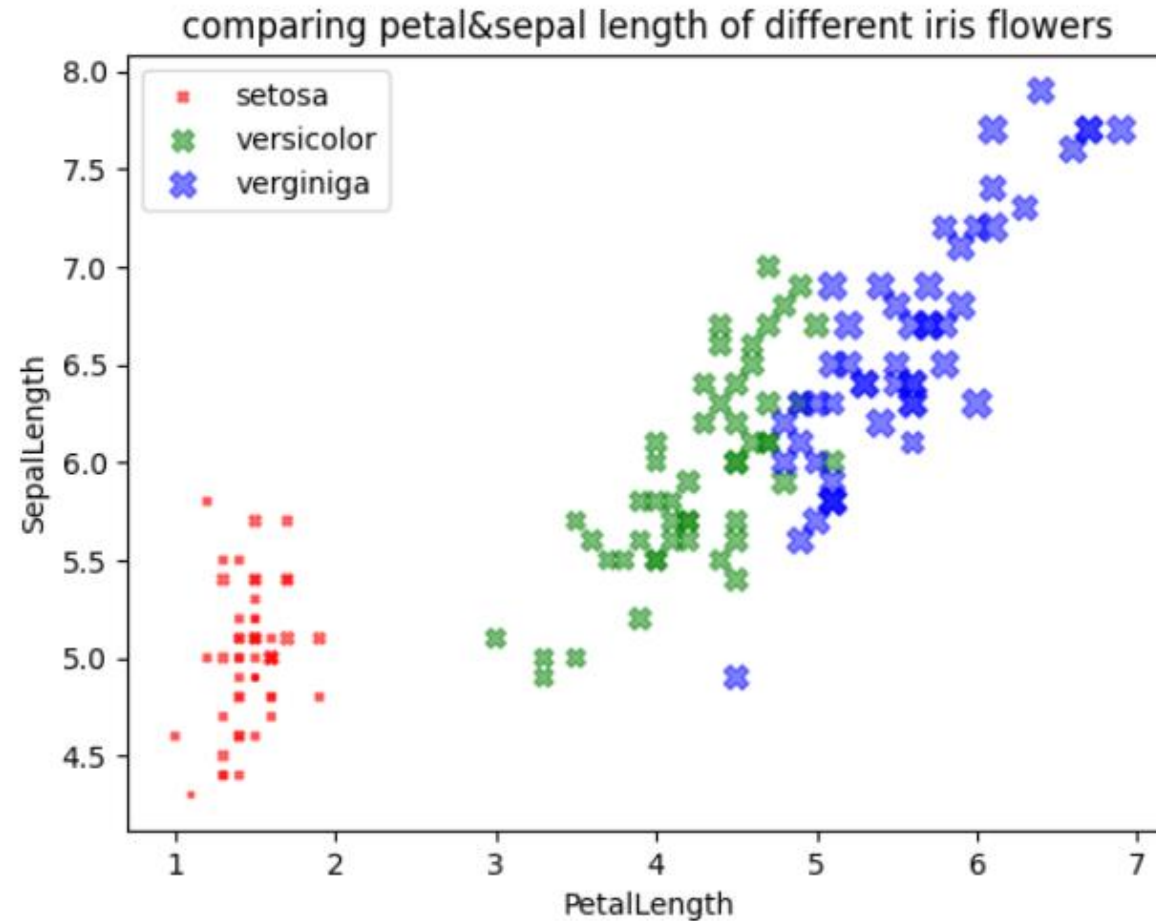
"+"	+	plus
"x"	×	x
"X"	⊗	x (filled)
"D"	◆	diamond
"d"	◊	thin_diamond
" "		vline
"_"	—	hline



การใช้งาน Parameter: marker

- สามารถใช้สัญลักษณ์ที่ต้องการใน Parameter: marker ได้เลย เช่น `marker='X'`
- ตัวอย่าง
- `plt.scatter(df2['PetalLength'][:50], df2['SepalLength'][:50], alpha=0.5, c=df2['Name'][:50], label='setosa', s=40*df2['PetalWidth'][:50], marker='X')`
- `plt.scatter(df2['PetalLength'][50:100], df2['SepalLength'][50:100], alpha=0.5, c=df2['Name'][50:100], label='versicolor', s=40*df2['PetalWidth'][50:100], marker='X')`
- `plt.scatter(df2['PetalLength'][100:], df2['SepalLength'][100:], alpha=0.5, c=df2['Name'][100:], label='virginica', s=40*df2['PetalWidth'][100:], marker='X')`
- `plt.legend()`
- `plt.xlabel('PetalLength')`
- `plt.ylabel('SepalLength')`
- `plt.title('comparing petal&sepal length of different iris flowers');`

ผลลัพธ์ตัวอย่างการใช้งาน Parameter: marker





สามารถกำหนดหน้าต่างของ Marker ตามกลุ่มที่ต้องการได้

- ใช้สัญลักษณ์ที่ต้องการใน Parameter: marker แต่ละกลุ่มได้เลย เช่น
- `plt.scatter(df2['PetalLength'][:50], df2['SepalLength'][:50], alpha=0.5, c=df2['Name'][:50], label='setosa', s=40*df2['PetalWidth'][:50], marker='s')`
- `plt.scatter(df2['PetalLength'][50:100], df2['SepalLength'][50:100], alpha=0.5, c=df2['Name'][50:100], label='versicolor', s=40*df2['PetalWidth'][50:100], marker='c')`
- `plt.scatter(df2['PetalLength'][100:], df2['SepalLength'][100:], alpha=0.5, c=df2['Name'][100:], label='virginica', s=40*df2['PetalWidth'][100:], marker='g')`
- `plt.legend()`
- `plt.xlabel('PetalLength')`
- `plt.ylabel('SepalLength')`
- `plt.title('comparing petal&sepal length of different iris flowers');`

ผลลัพธ์ตัวอย่างกำหนด Marker ตามกลุ่ม

