

Class period 14

Visualize_Data_Distribution part3

Quiz

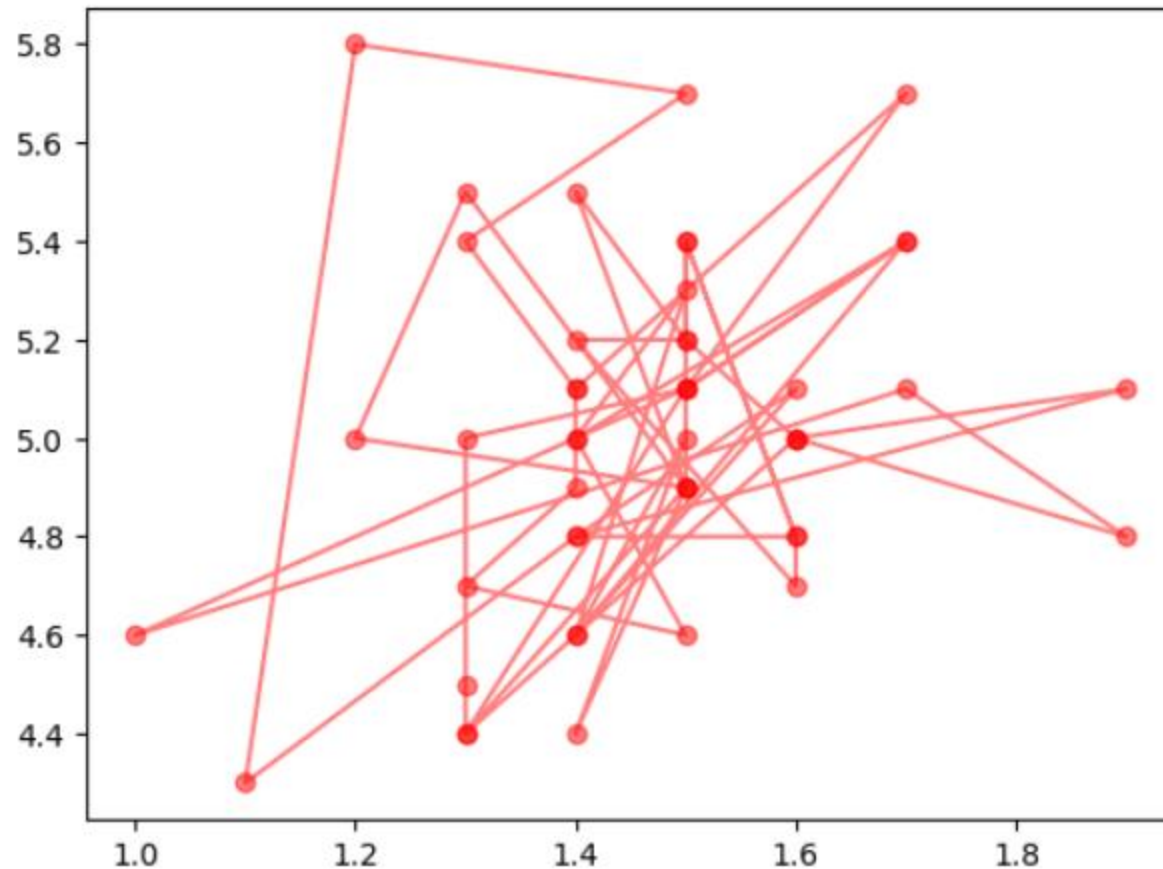
- ให้นักศึกษาโหลดข้อมูลข้อมูลสถิติการเดินทางบนโครงข่ายคมนาคม ปี 2563 - 2567 จาก
- <https://datagov.mot.go.th/dataset/9b9970e9-edd8-4518-99ae-2b81115068c2/resource/71a552d0-0fea-4e05-b78c-42d58aa88db6/download/passengers.xlsx>
- และโหลดลง google colab
- ลองดึงเฉพาะข้อมูล ท่าอากาศยานสุวรรณภูมิ ขาออกประเทศและสร้างตารางเก็บไว้ในตัวแปรใหม่

plt.plot()

- ฟังก์ชัน **plot** สามารถวาด **scatter** ได้เหมือนกัน และการใช้งานจะยืดหยุ่นกว่า สามารถกำหนดให้ลากเส้นเชื่อมจุด **scatter** แต่ละจุดได้
- **plt.plot()** input ตัวที่ 3 คือ ตัวกำหนดหน้าตาของ **marker** เช่น
- `plt.plot(df['PetalLength'][:50], df['SepalLength'][:50], 'o-r')`
- **input** ตัวที่ 3 `'o-r'` หน้าตา **marker/ลักษณะของเส้นเชื่อมจุด/สี**
- `o` คือ กำหนดหน้าตา **marker** เป็นสัญลักษณ์วงกลม
- `-` คือ กำหนดสัญลักษณ์ที่จะใช้ลากเส้นเชื่อมจุดเป็นเส้นทึบ
- `r` คือ ตัวย่อของแม่สีแดง

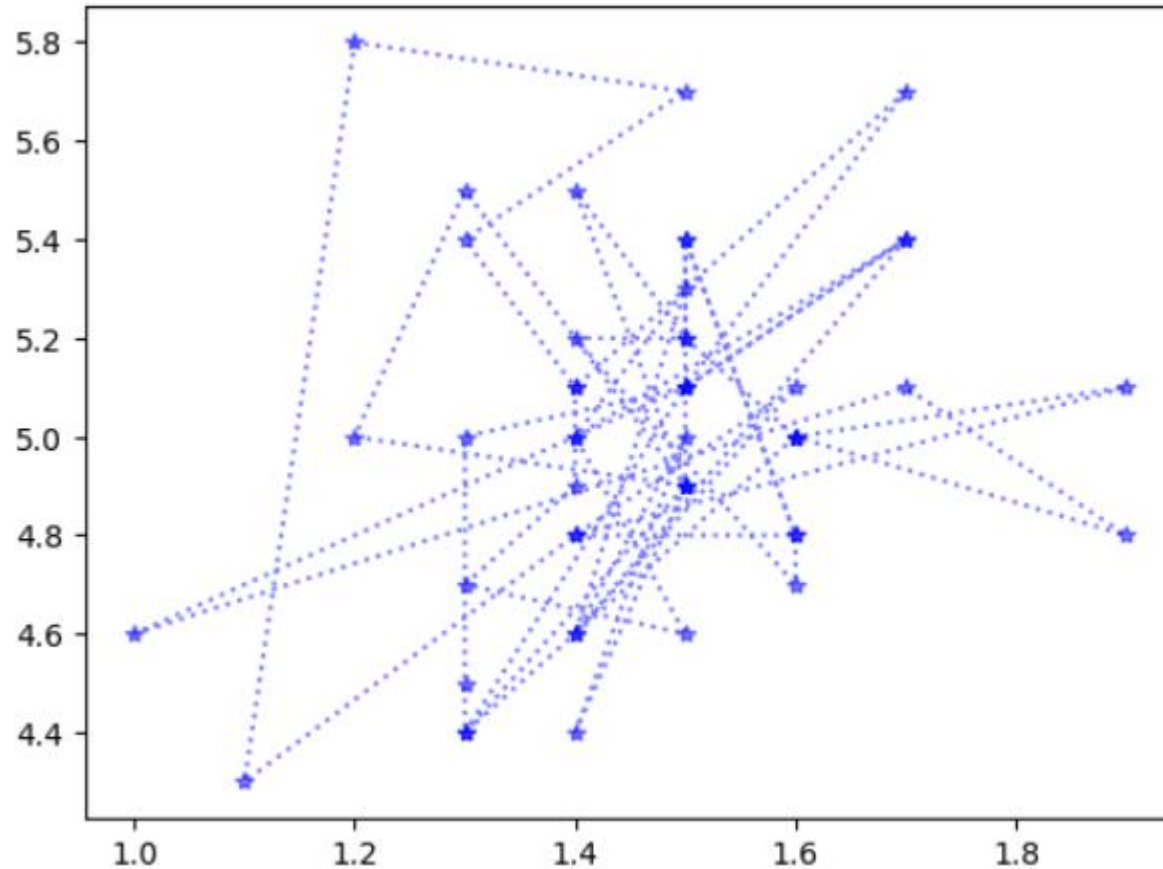
ตัวอย่างการใช้งาน plt.plot()

- `plt.plot(df['PetalLength'][:50], df['SepalLength'][:50], 'o-r', alpha=0.5)`



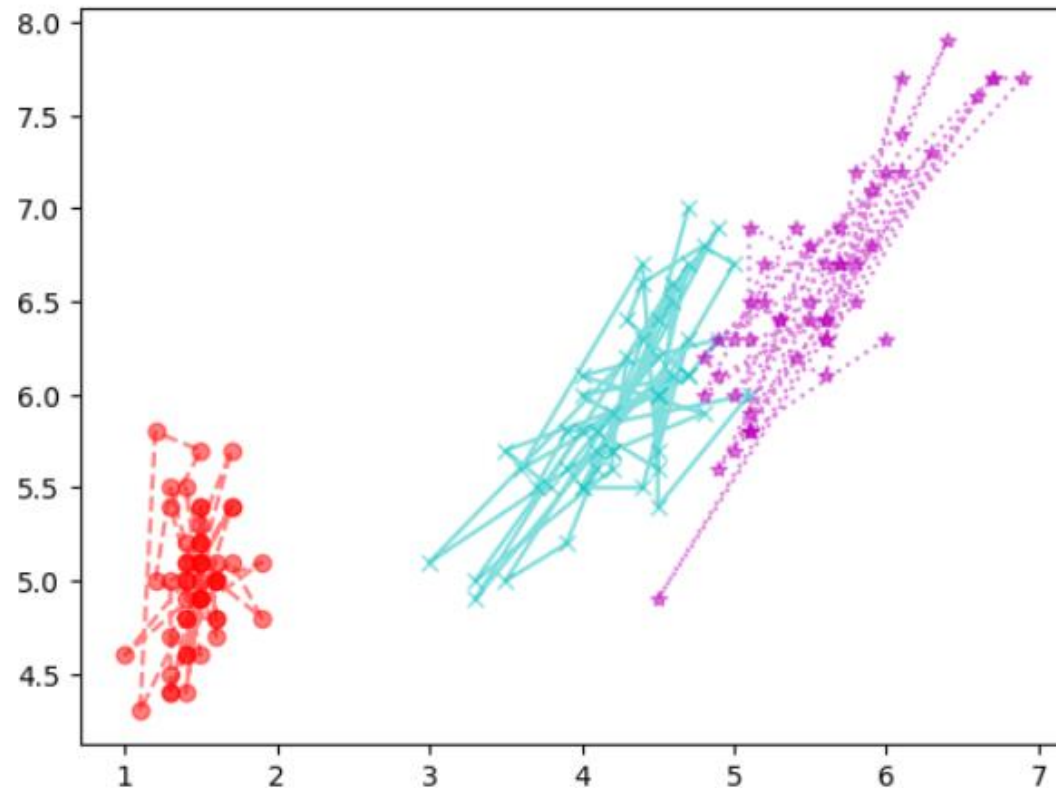
ตัวอย่างการใช้งาน plt.plot()

- `plt.plot(df['PetalLength'][:50], df['SepalLength'][:50], '*:b', alpha=0.5)`



ตัวอย่างการใช้งาน plt.plot() แบบ plot กราฟซ้อนกัน

- `plt.plot(df['PetalLength'][:50], df['SepalLength'][:50], 'o--r', alpha=0.5)`
- `plt.plot(df['PetalLength'][50:100], df['SepalLength'][50:100], 'x-c', alpha=0.5)`
- `plt.plot(df['PetalLength'][100:], df['SepalLength'][100:], '*:m', alpha=0.5)`

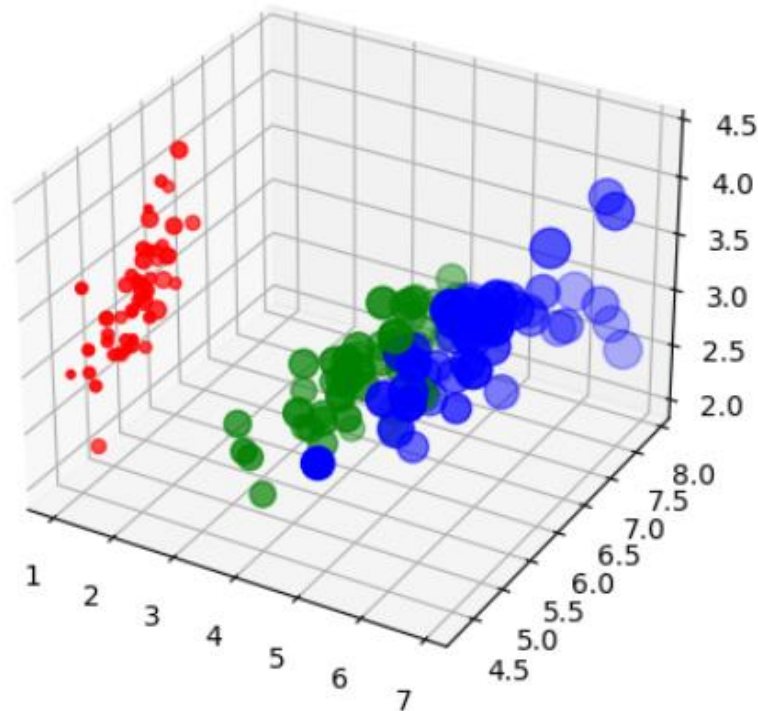


3D scatter

- การสร้างกราฟ **scatter 3 มิติ** สามารถสร้างได้โดยใช้คำสั่ง
- `ax = plt.axes(projection = "3d")` เพื่อกำหนดให้สร้างกราฟจำลอง **3 มิติ**
- การใช้งาน **input** แกน**X** แกน**Y** แกน**Z** และใส่ **parameter** ปรับแต่งกราฟ
- `ax.scatter (แกนX, แกนY, แกนZ, สี, สามารถใช้parameter:s เพิ่ม nomination ที่ 4)`
- เช่น
- `ax.scatter (df2 ['PetalLength'], df2 ['SepalLength'], df2 ['SepalWidth'], c=df2 ['Name'], s=80*df2 ['PetalWidth'])`

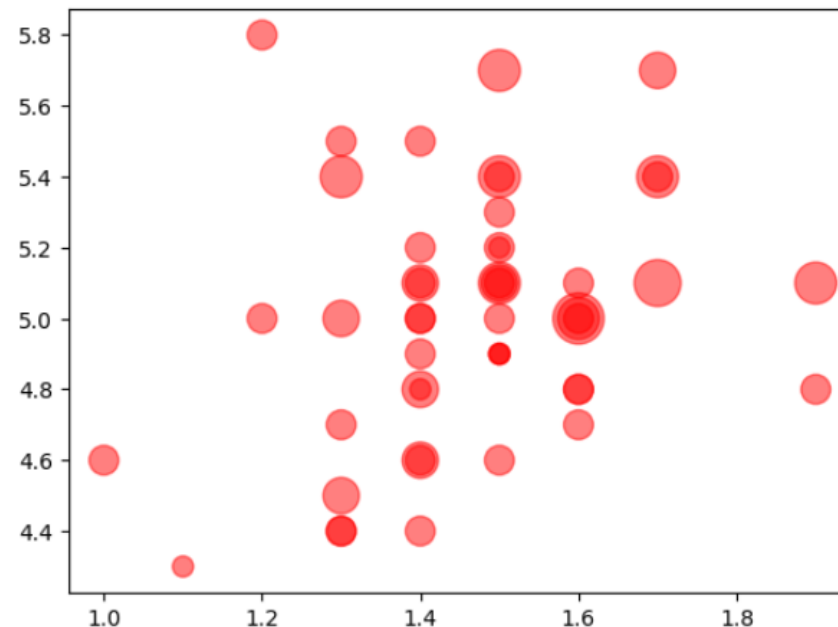
ตัวอย่างการใช้งาน `ax.scatter` สร้าง 3D scatter

- `ax = plt.axes(projection = "3d")`
- `ax.scatter(df2['PetalLength'], df2['SepalLength'], df2['SepalWidth'],
c=df2['Name'], s=80*df2['PetalWidth'])`



Bubble Chart

- กราฟฟองสบู่ คือ กราฟ **scatter** ที่ใช้ขนาดของ **marker** ในการสื่อปริมาณของข้อมูลในการดูความหลากหลายของข้อมูล เช่น ต้องการดูความหลากหลายของดอกไม้พันธุ์ **Iris-setosa**
- `plt.scatter(df['PetalLength'][:50], df['SepalLength'][:50], s=900*df['PetalWidth'][:50], c='r', alpha=0.5)`



ตัวอย่างการนำข้อมูลตารางมาสร้าง Bubble Chart

- โหลดข้อมูล ข้อมูลสถิติการเดินทางบนโครงข่ายคมนาคม ปี 2562 จาก
- <https://datagov.mot.go.th/dataset/9b9970e9-edd8-4518-99ae-2b81115068c2/resource/ad077d2d-5378-4e8a-b524-4814348df439/download/passenger-2019.xlsx>
- โหลดข้อมูลลง google colab
- `import pandas as pd`
- `data = pd.read_excel('/content/passenger-19.xlsx')`
- `data`

ตาราง

	รูปแบบการเดินทาง	วัตถุประสงค์	สาธารณะ/ส่วนบุคคล	หน่วยงาน	ยานพาหนะ/ท่า	เดือน	หน่วย	ปริมาณ
0	ทางถนน	การเดินทางระหว่างจังหวัด	สาธารณะ	บขส.	รถ ขบส. และ รถรวม	มกราคม	คน	5236331
1	ทางถนน	การเดินทางระหว่างจังหวัด	สาธารณะ	ขบ.	รถหมวด 3	มกราคม	คน	2550864
2	ทางถนน	การเดินทางระหว่างจังหวัด	ส่วนบุคคล	ทล.	รถยนต์เฉพาะ 4 ล้อ (10 จุดสำรวจ)	มกราคม	คัน	28907234
3	ทางถนน	การเดินทางระหว่างจังหวัด	ส่วนบุคคล	กทพ.	รถยนต์ทุกประเภท (10 จุดสำรวจ)	มกราคม	คัน	56404661
4	ทางถนน	การเดินทางภายในจังหวัด/กรุงเทพ	สาธารณะ	ขบ.	รถหมวด 4	มกราคม	คน	108716
...
367	ทางอากาศ	การเดินทางระหว่างประเทศ	สาธารณะ	ทอท.	ท่าอากาศยานดอนเมือง ขาออกประเทศ	ธันวาคม	คน	744031
368	ทางอากาศ	การเดินทางระหว่างประเทศ	สาธารณะ	ทอท.	ท่าอากาศยานอื่น ๆ ของ ทอท.ขาเข้าประเทศ	ธันวาคม	คน	755055
369	ทางอากาศ	การเดินทางระหว่างประเทศ	สาธารณะ	ทอท.	ท่าอากาศยานอื่น ๆ ของ ทอท. ขาออกประเทศ	ธันวาคม	คน	682621
370	ทางอากาศ	การเดินทางระหว่างประเทศ	สาธารณะ	ทย.	ท่าอากาศยานภูมิภาค ขาเข้าประเทศ	ธันวาคม	คน	72067
371	ทางอากาศ	การเดินทางระหว่างประเทศ	สาธารณะ	ทย.	ท่าอากาศยานภูมิภาค ขาออกประเทศ	ธันวาคม	คน	59594

372 rows x 8 columns



ขั้นตอนการ clean และเลือกข้อมูล

- เช็คและ drop ข้อมูลที่ missing
- `data_drop = data.dropna()`
- `data_drop`
- ต้องการสร้าง **Bubble Chart** ดูปริมาณของข้อมูลเดือนมกราคม เทียบระหว่าง ‘รถ ขบส. และ รถร่วม’, ‘รถหมวด 3’ และ ‘รถยนต์เฉพาะ 4 ล้อ (10 จุดสำรวจ)’
- เลือกเฉพาะข้อมูลปริมาณของ ‘รถ ขบส. และ รถร่วม’, ‘รถหมวด 3’ และ ‘รถยนต์เฉพาะ 4 ล้อ (10 จุดสำรวจ)’ ด้วย **iloc** และคูณข้อมูลปริมาณด้วย **0.001** เนื่องจากตัวเลขปริมาณมากเกินไป
- `0.001 * data_drop.iloc[:3, 7]`

```
0      5236.331
```

```
1      2550.864
```

```
2     28907.234
```

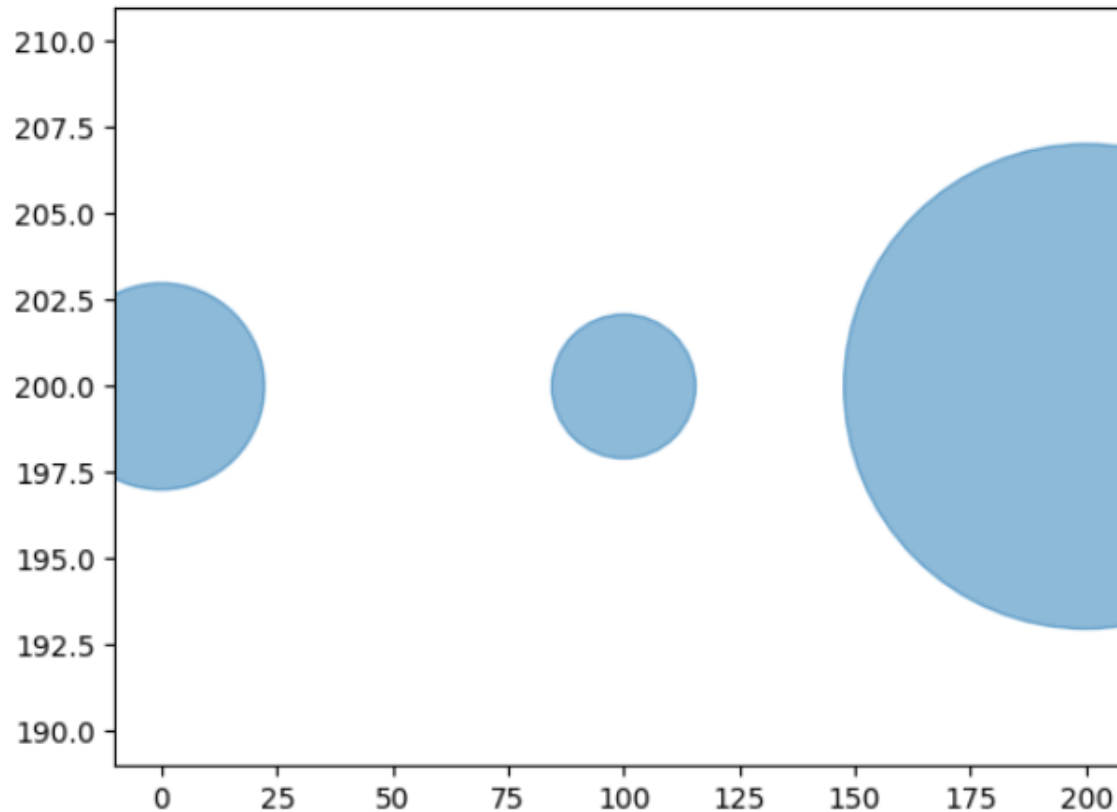
```
Name: ปริมาณ, dtype: float64
```

ขั้นตอนสร้าง Bubble Chart

- import ฟังก์ชันที่ต้องการใช้งาน
- `import matplotlib`
- `from matplotlib import pyplot as plt`
- `plt.scatter([0,100,200],[200,200,200],s=list(0.001*data_drop.iloc[:3,7]),alpha=0.5)`
- สร้างกราฟ **scatter** โดยกำหนดจุด **3** จุดอยู่ตำแหน่ง แกน **x** ที่ **0,100,200** ตามลำดับและตำแหน่ง แกน **y** ที่ **200** ทั้ง **3** จุด จากนั้นกำหนด **parameter:s** เป็น **list** ปริมาณข้อมูลทั้ง **3** ข้อมูลที่ต้องการเปรียบเทียบ และกำหนดความโปร่งแสงเป็น **50%**

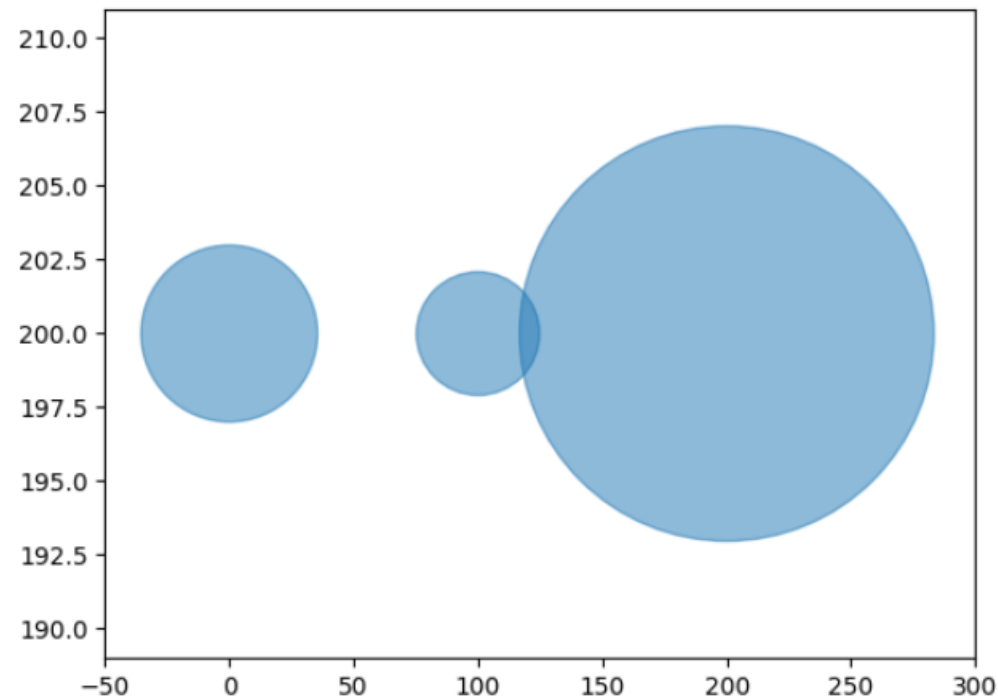
ผลลัพธ์จะได้ (ไม่ค่อยสวยงาม) ปรับแต่งกราฟ

- `plt.scatter([0,100,200],[200,200,200],s=list(0.001*data_drop.iloc[:3,7]),alpha=0.5)`



plt.xlim() กำหนดความยาวของแกน X

- สามารถกำหนดความยาวของแกน X เองได้ เนื่องจากขนาดจุดมันล้อมรอบของกราฟ
- โดยใช้คำสั่ง `plt.xlim()` เช่น
- `plt.scatter([0,100,200],[200,200,200],s=list(0.001*data_drop.iloc[:3,7]),alpha=0.5)`
- `plt.xlim((-50,300));`



กำหนดขนาดความกว้างความยาวของรูปภาพ

- แก้จุดซ้อนกันด้วย
- `matplotlib.rcParams['figure.figsize']=[8,5]`
- ทำงานใน memory กำหนดขนาดของรูปภาพ กว้าง*ยาว ตัวอย่าง กว้าง 8 ยาว 5
- กลับไปรันโค้ดก่อนหน้านี้อีกรอบ

