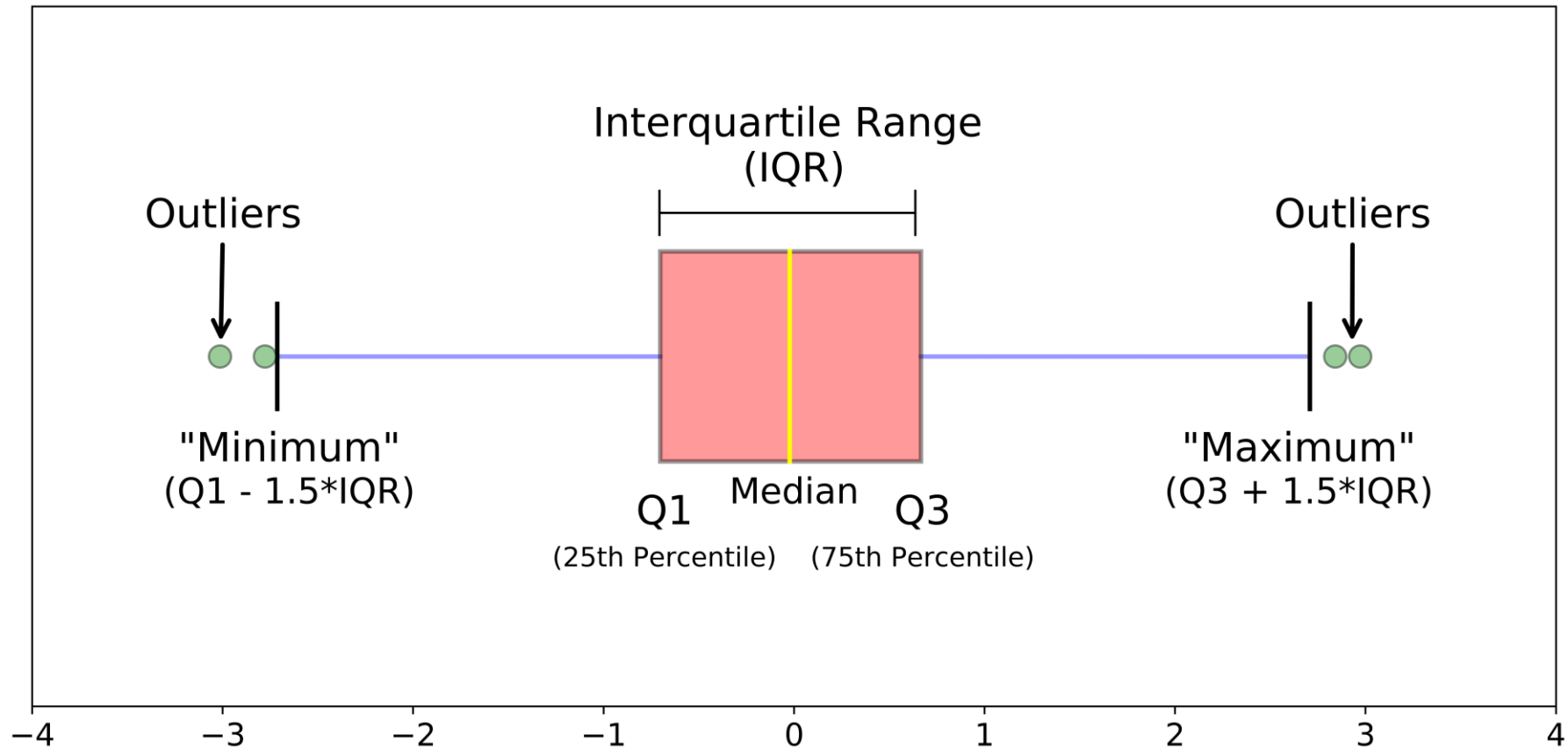


# Class period 15

Visualize\_Data\_Distribution part4

# Box-plot

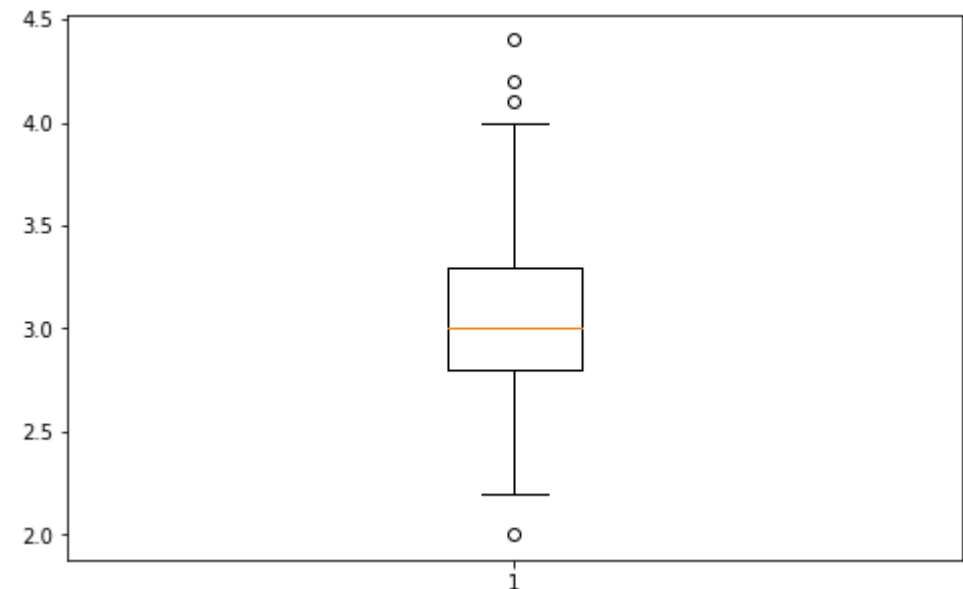


# Box-plot

ใช้ดูการกระจายของข้อมูลและ outliers โดย box-plot สามารถ return ค่าที่ต้องการ เช่น whiskers, caps, boxes, medians, fliers, means จากการวาดกราฟได้

วาด Box-plot ใช้คำสั่ง `plt.boxplot( 'ข้อมูลคอลัมน์x' )` เช่น ใช้ข้อมูลดอกไม้ `iris`  
`plt.boxplot(df[ 'SepalWidth' ])`

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7a8794dea440>,  
<matplotlib.lines.Line2D at 0x7a8794dea170>],  
'caps': [<matplotlib.lines.Line2D at 0x7a8794deb490>,  
<matplotlib.lines.Line2D at 0x7a8794deb370>],  
'boxes': [<matplotlib.lines.Line2D at 0x7a8794deb520>],  
'medians': [<matplotlib.lines.Line2D at 0x7a8794de9a50>],  
'fliers': [<matplotlib.lines.Line2D at 0x7a8794de9870>],  
'means': []}
```

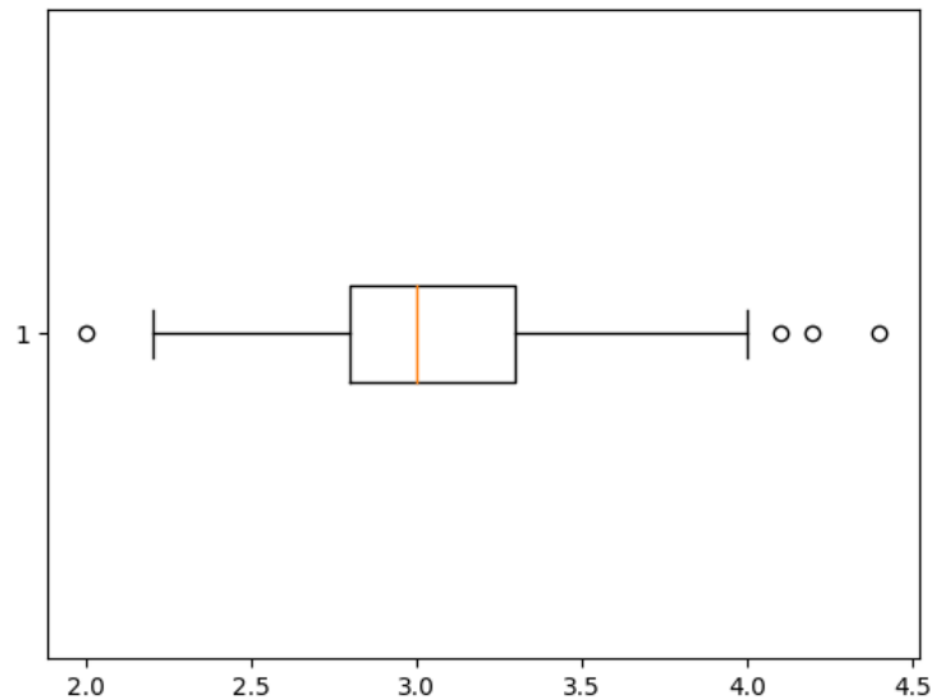


# การ return ค่าจาก Box-plot

- กำหนดตัวแปรที่จะใช้เก็บค่ากราฟ Box-plot
- ตัวแปร Box-plot จากนั้นเลือกค่าที่ต้องการ return ตามด้วย `.get_ydata()` หรือ `.get_xdata()` เลือกแนวแกนที่ต้องการดูค่า เช่น ต้องการดูค่า `fliers(outliers)` ในแนวแกน y
- `o = plt.boxplot(df['SepalWidth'])`
- `o['fliers'][0].get_ydata()`
- ผลลัพธ์จะได้ค่า `fliers` ของกราฟ Box-plot ในตัวแปร `o`
- `array([2. , 4.4, 4.1, 4.2])`

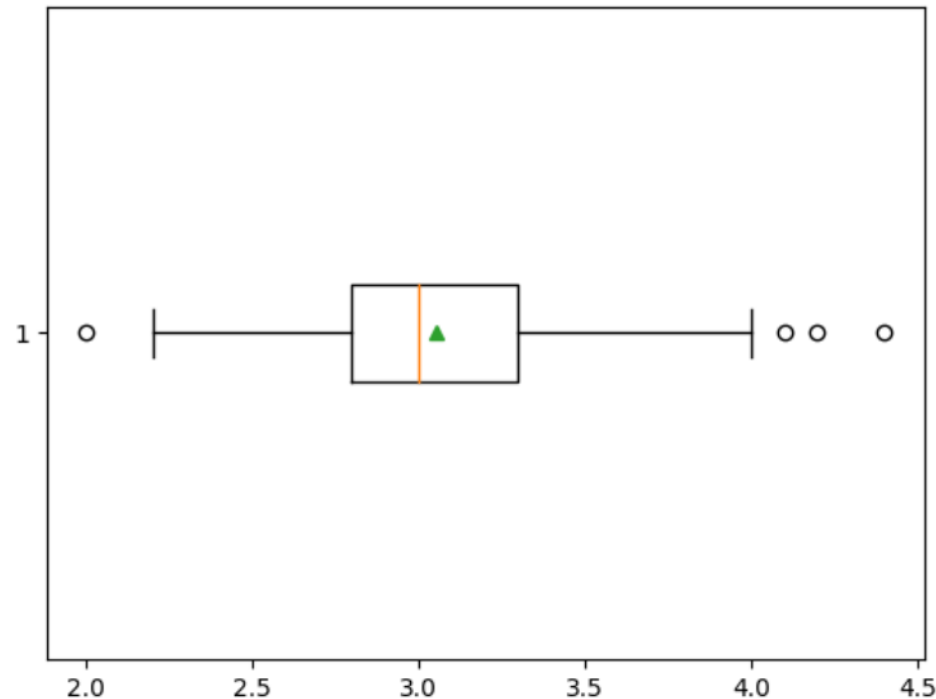
# Parameter: `vert=False` ปรับกราฟเป็นวาดแนวแกน X

- สามารถวาดกราฟในแนวแกน X ได้โดยการใส่และกำหนด parameter: `vert=False` (default=True) เช่น
- `ybp = plt.boxplot(df['SepalWidth'], vert=False)`



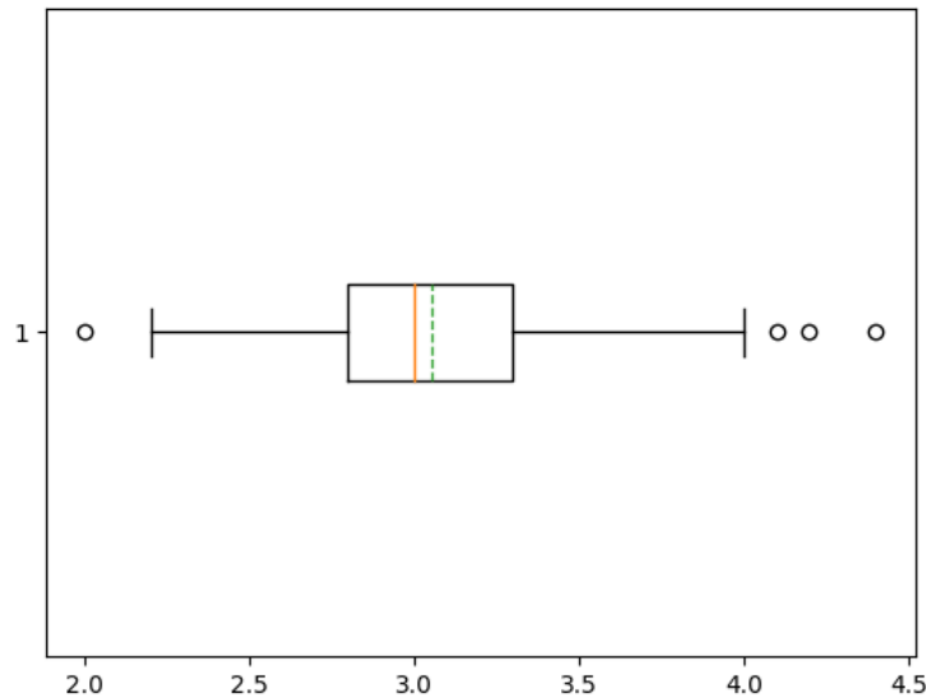
# Parameter: showmeans

- การแสดง **means** บนกราฟ **box-plot** สามารถแสดงได้โดยใช้ parameter: **showmeans=True** (default=False) เช่น
- `ybp = plt.boxplot(df['SepalWidth'], vert=False, showmeans=True)`



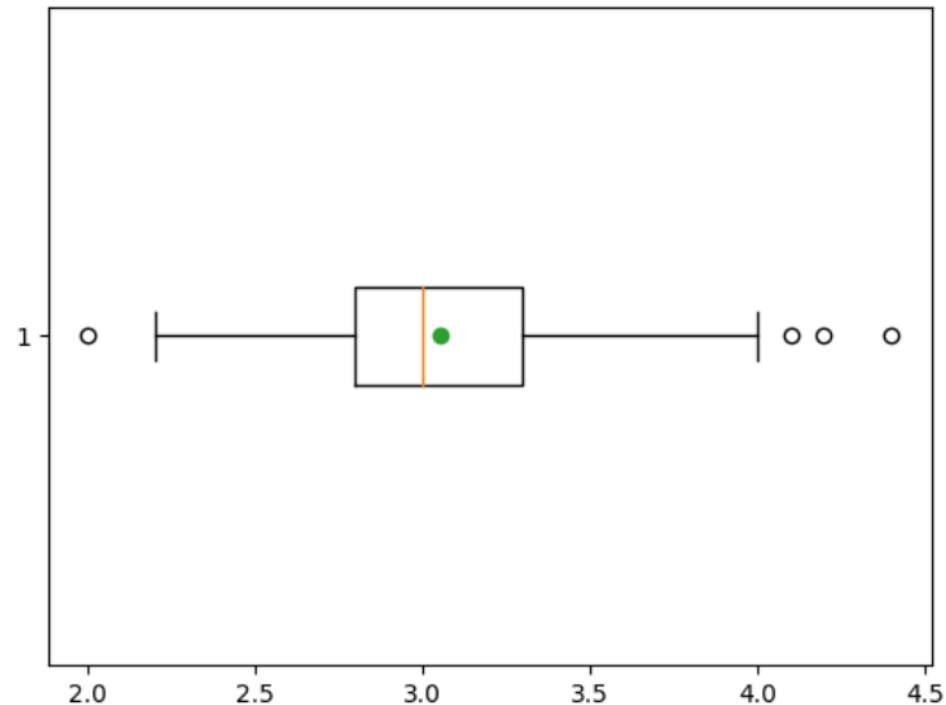
# Parameter: meanline

- แสดง means เป็นเส้นเพื่อง่ายต่อการเปรียบเทียบ ใช้ parameter: meanline=True (default=False) เช่น
- `ybp = plt.boxplot(df['SepalWidth'], vert=False, showmeans=True)`



# Parameter: meanprops เปลี่ยนหน้าตา marker ของ mean

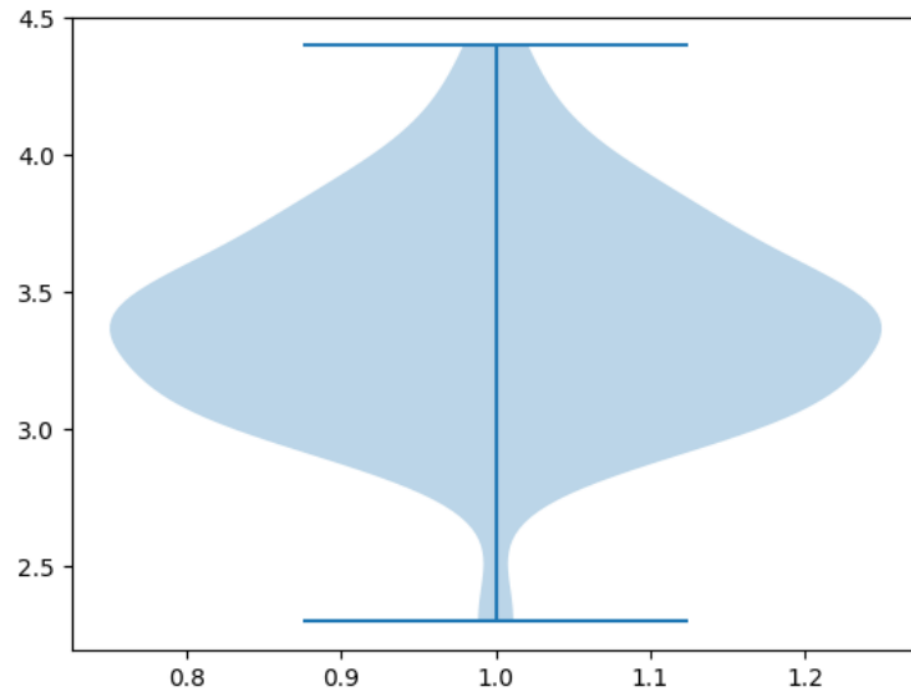
- สามารถเปลี่ยนลักษณะหน้าตาของ **marker mean** บนกราฟได้ตามที่ต้องการ เช่น
- `plt.boxplot(df['SepalLength'], vert=False, showmeans=True, meanprops={'marker': 'o'})`





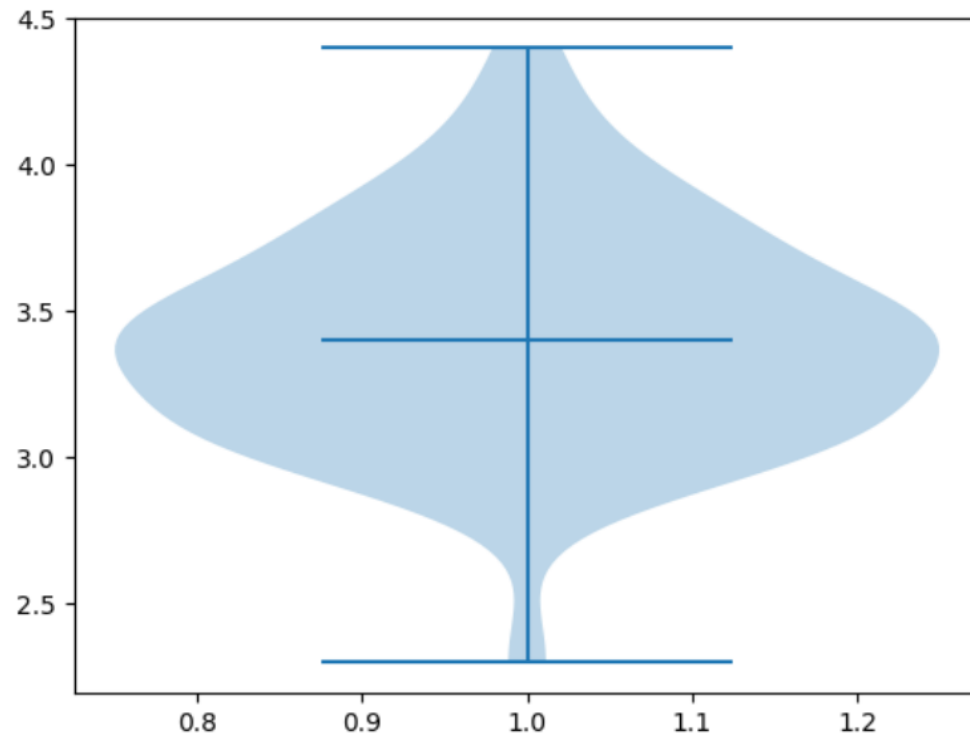
# Violin plot

- เป็นกราฟแสดงการกระจายตัวของข้อมูล
- สามารถใช้งานได้โดยใช้คำสั่ง `plt.violinplot('ข้อมูลคอลัมน์X')` เช่น
- `plt.violinplot(df['PetalLength'][:50])`



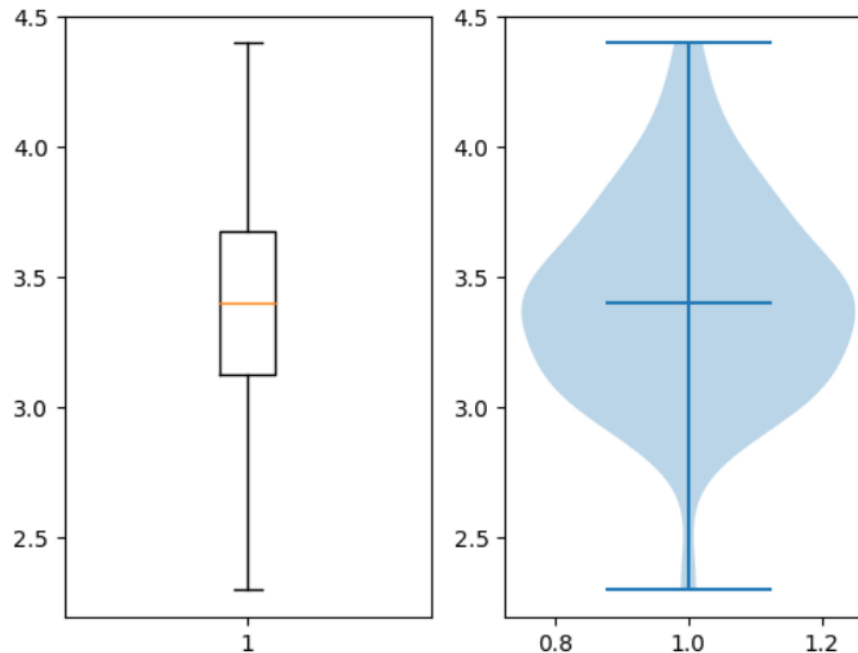
# Parameter: showmedians

- การแสดง **medians** บนกราฟ **violin-plot** สามารถแสดงได้โดยใช้ **parameter: showmedians = True (default=False)** เช่น
- `vi = plt.violinplot(df['SepalWidth'][:50], showmedians=True)`



# เปรียบเทียบระหว่าง box-plot กับ violin-plot

- `plt.subplot(1,2,1)`
- `bb = plt.boxplot(df['SepalWidth'][:50])`
- `plt.subplot(1,2,2)`
- `vi = plt.violinplot(df['SepalWidth'][:50], showmedians=True)`



# Quiz

- 1. เขียน function ที่รับ input เป็น output ของ boxplot แล้ว แสดงค่า min, max, q1, q2, q3
- 2. วาด boxplot เปรียบเทียบ การกระจายตัวของข้อมูล PetalLength ของดอก iris ทั้ง 3 ชนิด