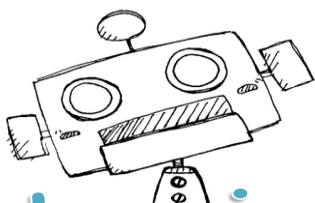




National Taiwan University
Department of Bio-industrial Mechatronics Engineering
Bio-mechatronics Lab

基本資料處理及繪圖工具

Python and R



Biomechatronics

Outline

基本資料處理

- 資料讀取
- 資料排列
- 統計運算
- 繪圖

資料處理應用

- **motor.txt**



程式編輯環境

- 環境變數設定
- **Ipython notebook**
- **RStudio**

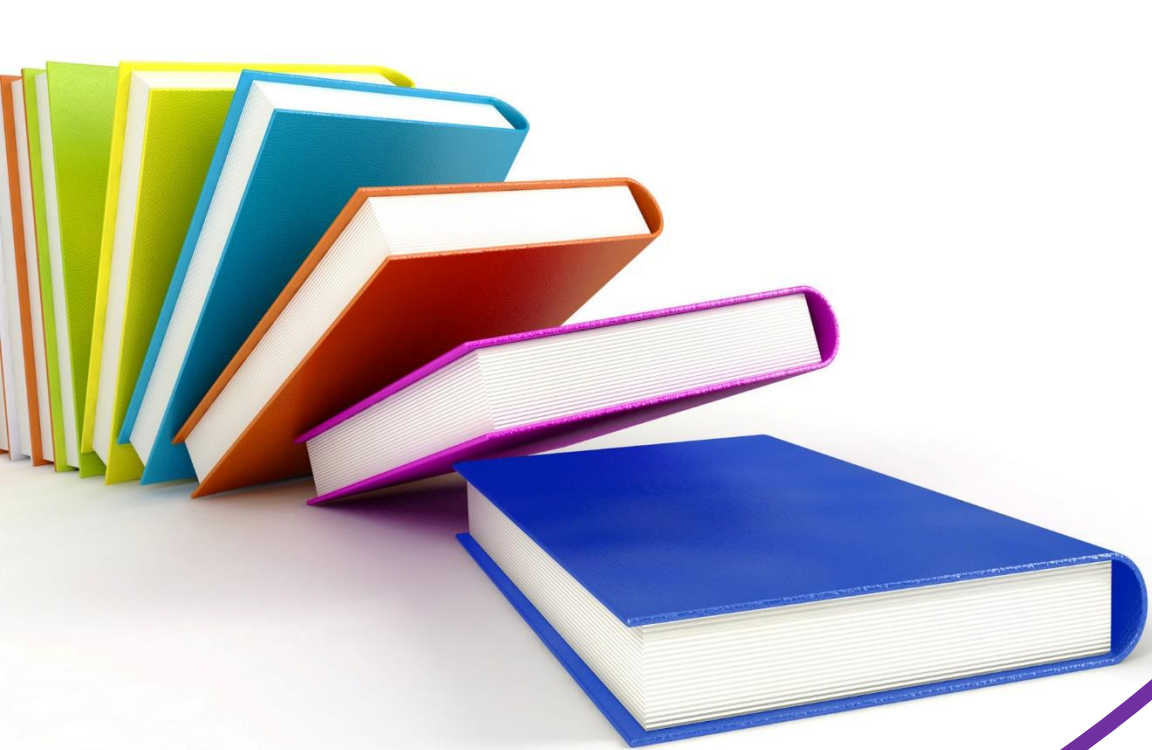


Python

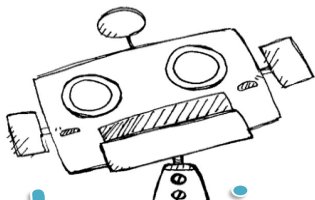


- **Ipython notebook**
- **cmd → pip3 install jupyter notebook**
- **cmd → cd target_folder**
 → jupyter notebook





基本資料處理

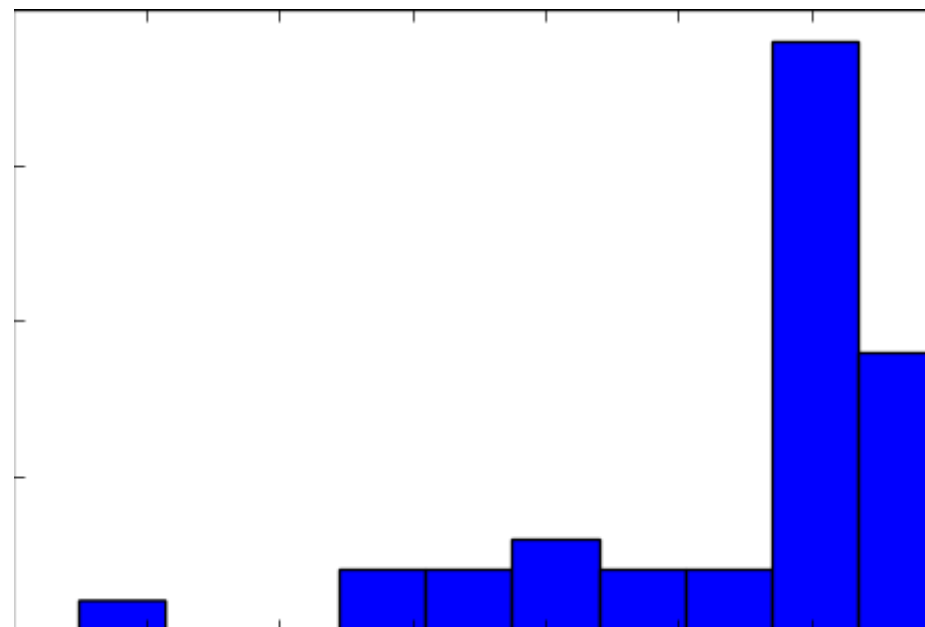


Biomechatronics

成績分析及繪圖



- 計算平均值、標準差
- 畫出分數分佈圖
– Histogram
- **grade.csv**



資料讀取

- 資料儲存於文件中
- 常見儲存資料格式
 - **xlsx**
 - **txt**
 - **csv**
 - **npz**
 - **none**



資料讀取



- **Python**
 - **Numpy and csv**
 - **pip3 install numpy**
 - **Read 'grade.csv'**

```
import numpy as np
import csv
file_name = 'grade.csv'
with open(file_name, 'r') as f:
    data = list(csv.reader(f, delimiter=','))
data = np.array(data)
```

- **觀察data的資料**

```
print(data.shape)
print(data)
```



資料讀取

- 呼叫**1st raw**的資料

```
data[0, :]
```

- 呼叫**2nd column**的資料

```
data[:, 1]
```



資料型態



- 觀察**data**中的資料型態

```
type(data[0, 0])
```

- 資料型態轉換

```
data = data.astype('float')  
type(data[0, 0])
```



資料排列



- 通常會將同類型資料排序成 **raw data**
- 或是依不同演算法的需求排列
- **Numpy: reshape, vstack, hstack, ...**



資料排列



- **reshape**

```
data1 = data.reshape([-1])  
data1.shape
```

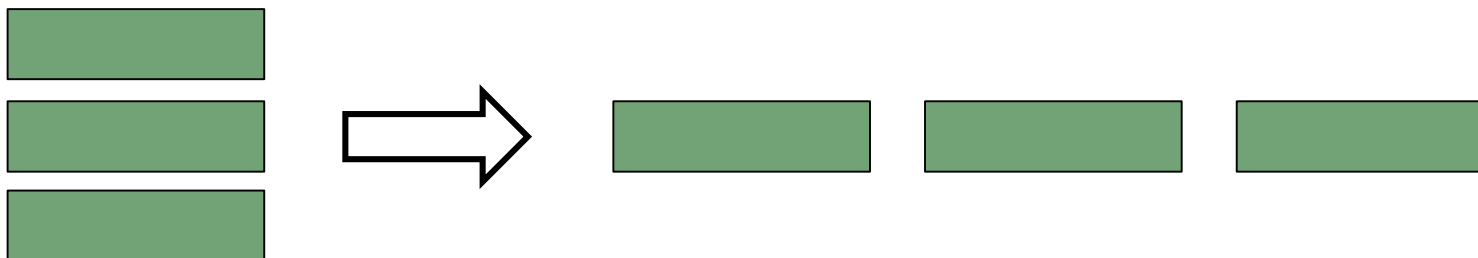
- **hstack**

```
data2 = np.hstack((data[0, :], data[1, :], data[2, :], data[3, :], data[4, :]))  
data2.shape
```

- 比較兩種轉換方式

```
data1 == data2
```

小練習：利用**vstack**，將資料疊成**1**個**column**



統計運算

- 在一般的敘述統計中，最在乎的是平均值及標準差
- 推論統計的統計量在此不予討論
 - **Python**的統計函式庫：**numpy, scipy**



統計運算



- 平均值

```
np.mean(data1)
```

- 標準差

```
np.std(data1)
```

- 進一步的統計運算，可以參考**python**的**scipy module**



統計運算



- **Test:**

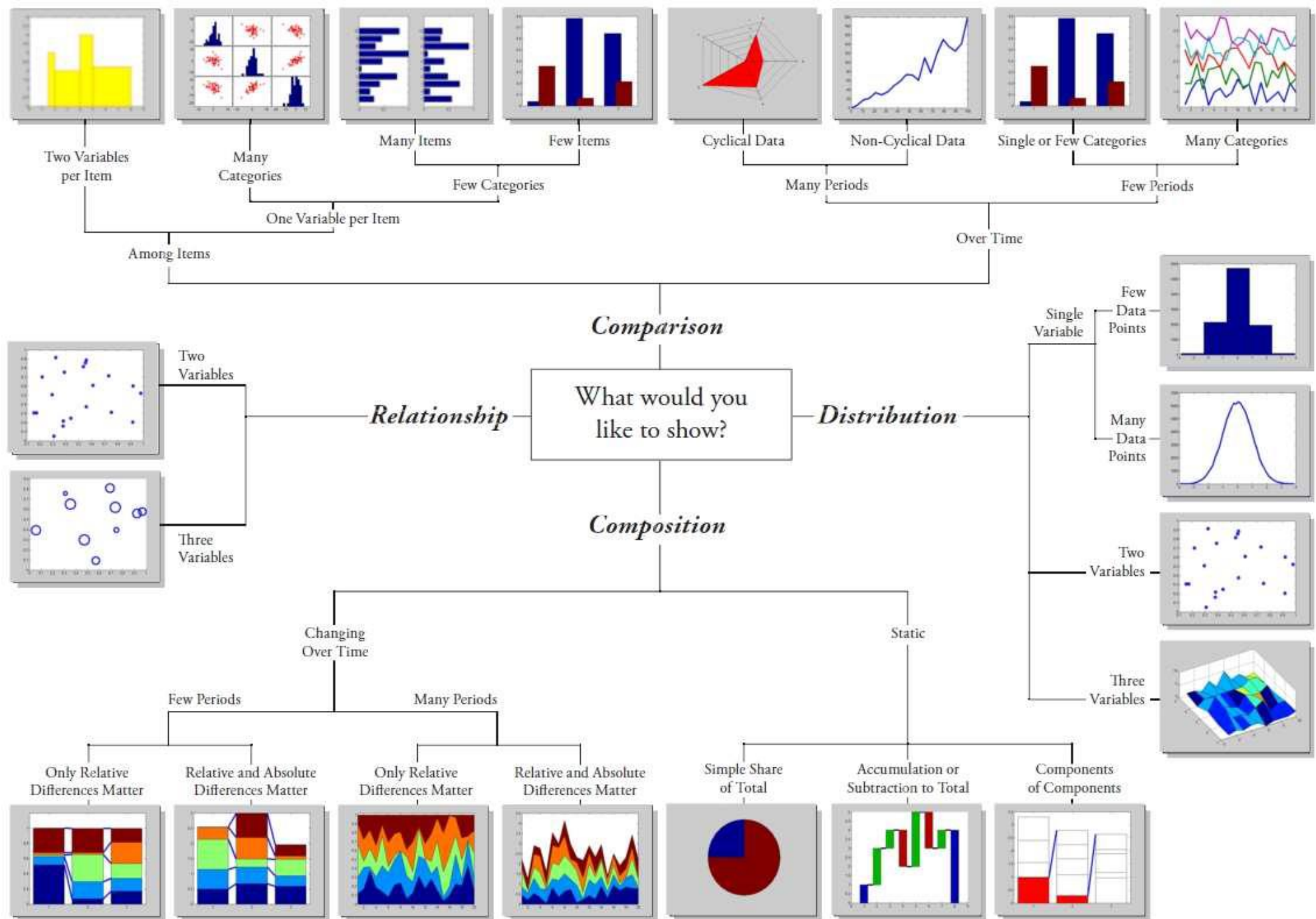
- 不使用上述函數，直接計算平均值及標準差

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}.$$

樣本標準差：分母為**N-1**

母體標準差：分母為**N**





繪圖

- 我們擁有的資料是成績
 - 如何向別人呈現這些資料？
- **Histogram**
 - 呈現考試成績分佈情形



繪圖



- **matplotlib** → **pip3 install matplotlib**
- 小練習：畫一個**sin**函數

```
import matplotlib.pyplot as plt  
x = np.arange(0, 10, 0.1)  
y = np.sin(x)  
plt.plot(x, y, '-r')  
plt.title('sin func')  
plt.show()
```



繪圖

- Histogram

```
plt.hist(data1)  
plt.show()
```



R

- 安裝 RStudio
- 設定開始位置
 - 與**MATLAB**相同，
要有一個工作位置



資料讀取



- Read 'grade.csv'

```
d = read.csv("grade.csv", header=FALSE)
```

- 觀察data的資料

```
d  
dim(d)
```

- 此時資料格式為frame

→ 轉換為較熟悉的matrix

```
d = data.matrix(frame=d)
```

- 觀察第3列第5行的資料

```
d[3, 5]
```

在R語言中，index的起始數與python不同，與MATLAB相同



資料讀取

- 呼叫**1st raw**的資料

`d[1,]`

- 呼叫**2nd column**的資料

`d[, 2]`



資料型態



- 觀察**data**中的資料型態

```
class(d)  
class(d[1, 1])  
typeof(d)  
typeof(d[1, 1])
```

- 資料型態轉換

```
as.double(d)
```



資料排列



- 通常會將同類型資料排序成 **raw data**
- 或是依不同演算法的需求排列
- **as.double**, union, rbind, cbind, c, ...



資料排列

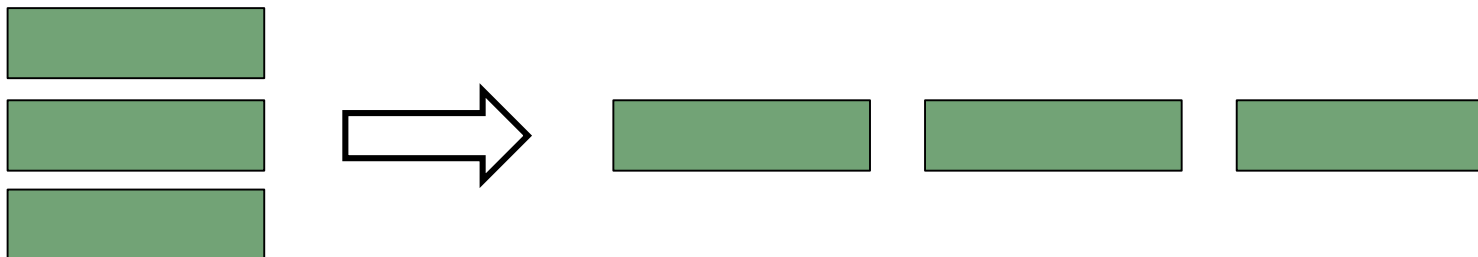


- **as.double**

```
d1 = as.double(d)
```

- **C**

```
d2 = c(d[1,], d[2,], d[3,], d[4,], d[5,])
```



統計運算

- 在一般的敘述統計中，最在乎的是平均值及標準差
- 推論統計的統計量在此不予討論



統計運算



- 平均值

`mean(d2)`

- 標準差

`sd(d2)`
`sqrt(var(d2))`



統計運算



- **Test:**

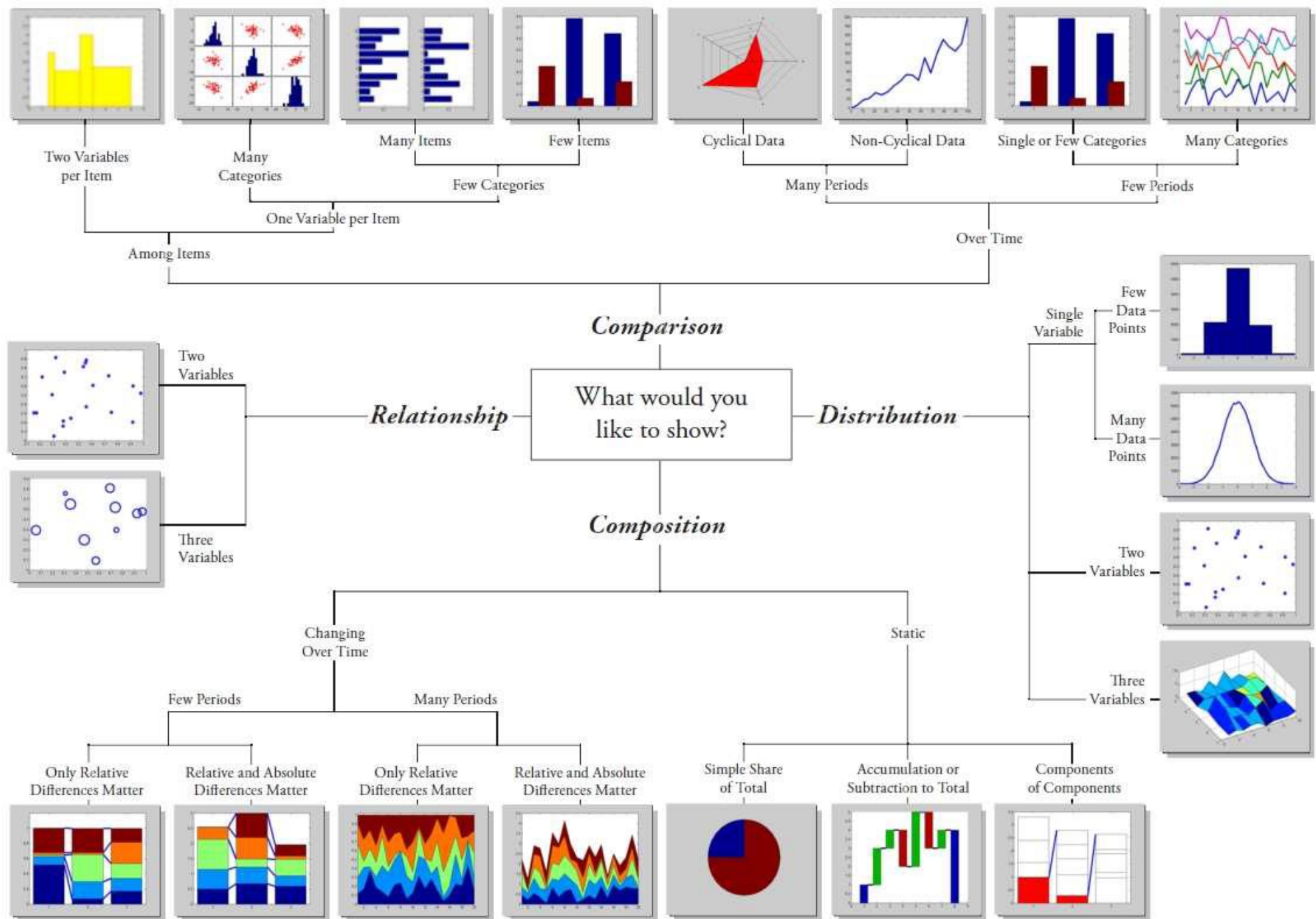
- 不使用上述函數，直接計算平均值及標準差

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}.$$

樣本標準差：分母為**N-1**

母體標準差：分母為**N**





繪圖

- 我們擁有的資料是成績
 - 如何向別人呈現這些資料？
- **Histogram**
 - 呈現考試成績分佈情形



繪圖



- 小練習：畫一個**sin**函數

```
x = seq(0, 10, 0.1)  
y = sin(x)  
plot(x, y, 'l')
```



繪圖

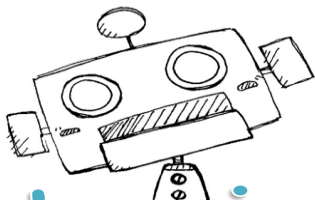
- Histogram

```
hist(d2)  
hist(d)  
hist(d2, breaks=10)  
hist(d2, breaks=5)  
hist(d2, breaks=seq(0, 100, 10))
```





資料處理應用



資料處理及分析



- **motor.txt**
 - 轉速、時間

- **PTT spider**
 - https://github.com/WarrenTseng/ptt_spider





Thanks for your attention

