Numerical Method

National Cheng Kung University **Department of Engineering Science**

Instructor: Chi-Hua Yu

LAB3

注意事項

1. Lab 的繳交期限為**星期二(3/14)17:00**。
2. Lab 的分數分配：Lab 分數 100%，Bonus 20%。
3. 請儘量於 Lab 時段完成練習，完成後請找助教檢查，檢查後即可離開。
4. 檔名規定: 檔名錯誤將記為 0 分
   - i. Lab: 請用 **學號_LabNumber** 為檔名做一個資料夾 **(e.g.,N96091350_Lab3)**，將 ipynb 檔放入資料夾，壓縮後上傳至課程網站**(e.g., N96091350_Lab3.zip)**
   - ii. Bonus: 請用 學號_bonus 為檔名做一個資料夾**(e.g.,N96091350_bonus)**，將 ipynb 檔放入資料夾，壓縮後上傳至課程網站**(e.g., N96091350_bonus.zip)**。
5. **Code 中需有註解。**
6. 未完成者可於**下周一 (3/20) 09:00 a.m.** 前上傳至 Moodle，惟補交的分數將乘以 0.8 計，超過期限後不予補交。
7. Bonus 需於**下周一 (3/20) 09:00 a.m**.前上傳至 Moodle，不予補交。
8. 準時繳交者，請交至「Lab3 準時繳交區」；補交者，請交至「Lab3 補交區」；bonus 請繳交至「bonus 繳交區」

請勿抄襲，抄襲者與被抄襲者本次作業皆 0 分計算

Total: 120%

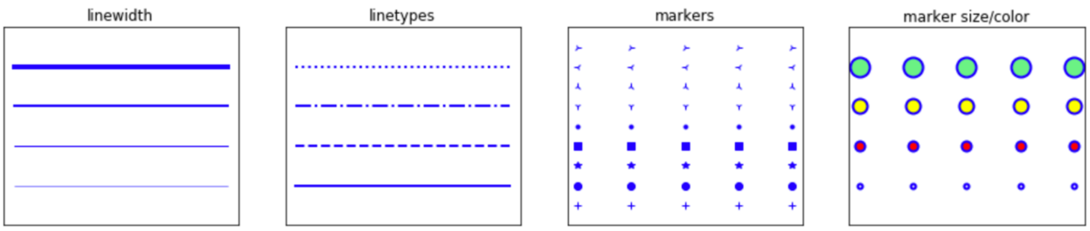
This lab is trying to plot lines or points by using the module **Matplotlib**. So before you start to code, please read the document first.

Matplotlib document link

1.(100%) Please download the template file `LineProp.ipynb`. Plot a figure with four subplots to illustrate the line properties including the following

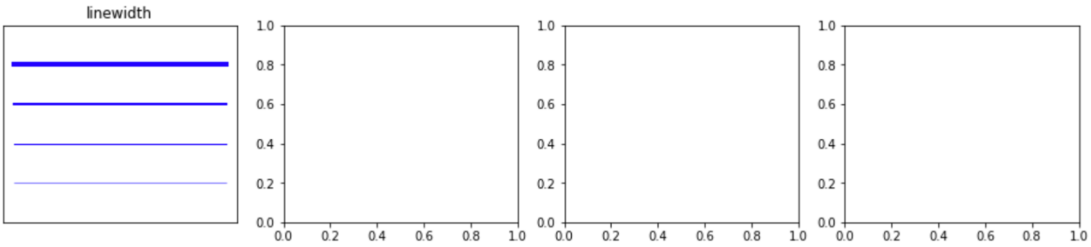
```
linewidths = [0.5, 1.0, 2.0, 4.0]
linestyles = ['-', '--', '-.', ':']
markers = ['+', 'o', '*', 's', '.', '1', '2', '3', '4']
markersizecolors = [(4, "white"), (8, "red"), (12, "yellow"),
(16, "lightgreen")]
```

Below is a sample output:



```
x = np.linspace(-5, 5, 5)

y = np.ones_like(x)
def axes_settings(fig, ax, title, ymax):
   ax.set_xticks([])
   ax.set_yticks([])
   ax.set_ylim(0, ymax+1)
   ax.set_title(title)

fig, axes = plt.subplots(1, 4, figsize=(16,3))

# Line width
linewidths = [0.5, 1.0, 2.0, 4.0]
for n, linewidth in enumerate(linewidths):

axes[0].plot(x, y + n, color="blue", linewidth=linewidth)
axes_settings(fig, axes[0], "linewidth", len(linewidths))
```

(**Big hint**: the following code in the template file will **give you the first subplot**)



bonus(20%). Please download the template file `SimpleRegr.ipynb` For a set of n samples $(x_0, y_1), (x_1, y_1), \ldots\ldots (x_n, y_{n-1})$,
you can easily fit a line

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

for these samples. This fitting is called simple linear regression. Using the least square fitting criterion,
we can show:

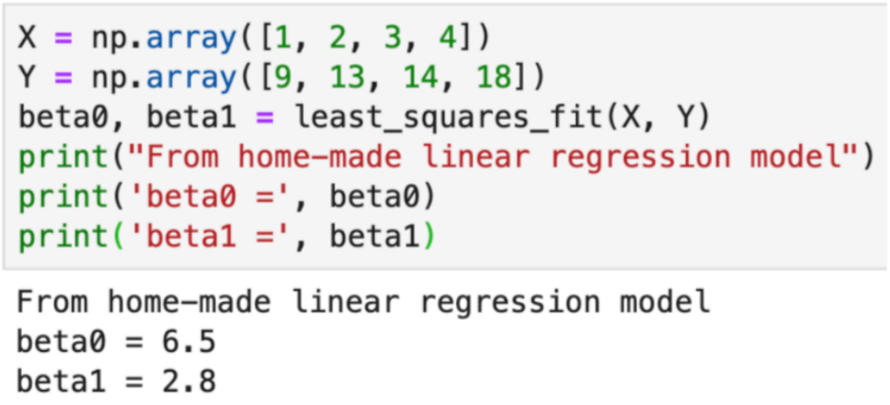
$$\hat{\beta}_1 = \frac{\sum_{i=0}^{n-1}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=0}^{n-1}(x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where x̄ and ȳ are the sample mean. Implement the function `least_squares_fit` in the template file to compute β_0 and β_1 from a set of samples. **Report $\boldsymbol{\beta_0}$ and $\boldsymbol{\beta_1}$ and plot these samples and the fitting line.**

Below is a sample run:

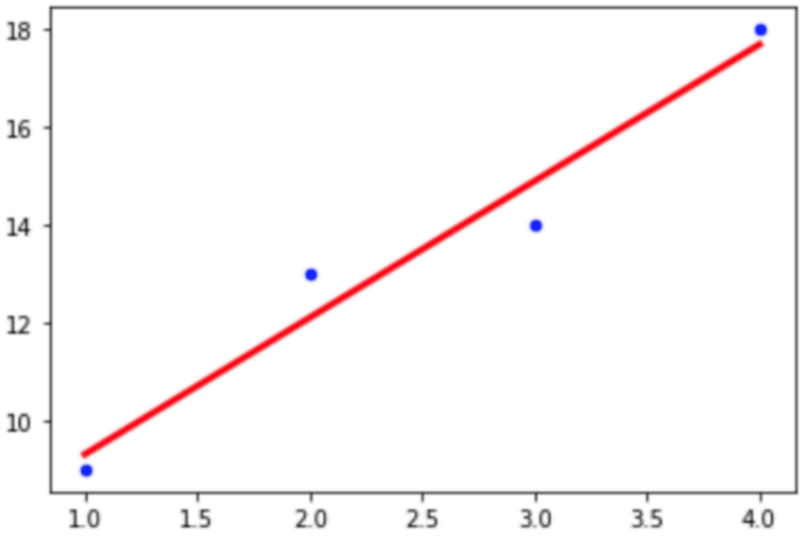
```
def least_squares_fit(x,y):
```



```
X = np.array([1, 2, 3, 4])
Y = np.array([9, 13, 14, 18])
beta0, beta1 = least_squares_fit(X, Y)
print("From home-made linear regression model")
print('beta0 =', beta0)
print('beta1 =', beta1)
```

```
From home-made linear regression model
beta0 = 6.5
beta1 = 2.8
```

And a sample plot (`line width = 3, marker size = 20`):


18
16
14
12
10
1.0
1.5
2.0
2.5
3.0
3.5
4.0