1.
用原資料/使用standard score轉換後的資料/使用scaling轉換後的資料分別算kmeans得到label,然後都去跟原資料跑出within class difference，做了三次後產生下圖：

```
5
pure wicd = 97.20457357401651
5
standard score wicd = 1161.5283586880184
3
scaling wicd = 1051.9768470260174
```

```
6
pure wicd = 97.22486903387325
3
standard score wicd = 1161.7853227902192
2
scaling wicd = 1052.0072844643441
```

```
4
pure wicd = 97.20457357401651
8
standard score wicd = 1161.5992568265601
7
scaling wicd = 1051.9768470260174
```

(每行上面的數字代表跑了幾個iteration後收斂)
透過觀察上面三組數據，可以發現標準化後的數據對於使用Kmeans分群未必有幫助。

2.
用K-nearest neighbors (KNN)演算法對iris資料進行預測，每次拿一筆資料當作test資料，剩下當作train資料，分別印出1-NN到10-NN的10個confusion matrix:

```
1 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 47.  3.]
2 [ 0.  3. 47.]

2 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 47.  3.]
2 [ 0.  3. 47.]

3 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 47.  3.]
2 [ 0.  3. 47.]

4 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 47.  3.]
2 [ 0.  3. 47.]

5 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 47.  3.]
2 [ 0.  2. 48.]
```

```
6 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 47.  3.]
2 [ 0.  3. 47.]

7 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 46.  4.]
2 [ 0.  1. 49.]

8 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 47.  3.]
2 [ 0.  2. 48.]

9 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 47.  3.]
2 [ 0.  2. 48.]

10 NN confusion matrix
    0   1   2
0 [50.  0.  0.]
1 [ 0. 46.  4.]
2 [ 0.  3. 47.]
```

可以觀察到在這組資料集下，使用不同的K對於最後預測的準確度關係不大。