題目：集群與視覺化

k-means 是最常見的集群法之一，請讀取`iris.csv` 資料中的數值欄位進行 k-means集群，並將其繪製視覺化圖形

- 相關套件提示
  - Python 套件：pandas, sklearn.cluster, matplotlib.pyplot

15% 小題一：讀取資料

題目說明

- 請讀取 `data`資料夾內 `iris.csv` 資料集

答案示意

- (Python) 以下提供前五筆輸出作為參考：

```
##    Sepal.Length  Sepal.Width  Petal.Length  Petal.Width Species
## 0           5.1          3.5           1.4          0.2  setosa
## 1           4.9          3.0           1.4          0.2  setosa
## 2           4.7          3.2           1.3          0.2  setosa
## 3           4.6          3.1           1.5          0.2  setosa
## 4           5.0          3.6           1.4          0.2  setosa
```

15% 小題二：次數分配表

題目說明

- 請計算 Species 欄位之次數分配表

答案示意

- (Python) 輸出應如下：

```
## setosa        50
## virginica     50
## versicolor    50
## Name: Species, dtype: int64
```

30% 小題三：k-means 集群

題目說明

- 使用 k-means 進行集群，群數為 3，並將各元素之集群結果輸出
- 因集群結果具隨機性質，有集群差異屬正常
  - 若想驗證結果可以透過隨機種子的設定來達成 (此部分非必要執行)
    - Python: `random_state=1`
- 提示：
  - Python 使用 KMeans 函數 (透過 `from sklearn.cluster import KMeans` 載入)

答案示意

- (Python)輸出應如下：

```
## KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
##        n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
##        random_state=1, tol=0.0001, verbose=0)
```

```
## [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  1 1 1 1 1 1 1 1 1 1 1 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 2 2 2 0 2 2 2 2 2 2 0 2
##  2 2 0 0 2 2 2 2 2 0 2 0 2 0 2 2 0 0 2 2 2 2 2 0 2 2 2 2 0 2 2 2 0 2 2 2 2 0 2 2 2 0
##  2 0]
```

20% 小題四：分組計算

題目說明

- 請計算每個 Species 種類的集群結果
  - 若無法取得集群結果，可自行產生模擬結果並作答此題

答案示意

- 因作答方式可能不同，輸出之物件類型不需完全相同
- (Python) 輸出示意如下：
  - 原 setosa 種類之集群結果，皆分至第 1 群
  - 原 versicolor 種類之集群結果，分至第 0 群的有 48 個樣本、分至第 3 群的有 2 個樣本
  - 原 virginica 種類之集群結果，分至第 2 群的有 36 個樣本、分至第 0 群的有 14 個樣本

```
## KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
##        n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
##        random_state=None, tol=0.0001, verbose=0)

## Species     cluster
## setosa      0          50
## versicolor  1          48
##             2           2
## virginica   2          36
##             1          14
## Name: cluster, dtype: int64
```

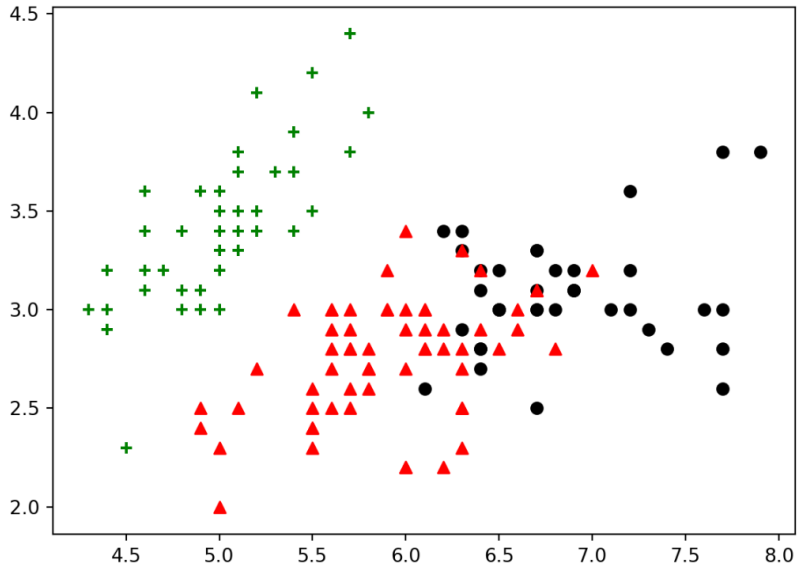
20% 小題五：資料視覺化

題目說明

- 請將集群結果以散點圖(scatter diagram)進行視覺化：
  - (10%) 點的位置：
    - X 軸使用 `Sepal.Length`
    - Y 軸使用 `Sepal.Width`
  - (5%) 點的顏色使用原始分類(Species)
  - (5%) 點的形狀使用集群結果(cluster)
- 若無法取得集群結果，可自行產生測試點畫一散點圖作答此題

答案示意

- (Python) 視覺化結果示意圖：



題目三：隨機森林迴歸預測模型

隨機森林是經典的監督式學習算法之一，請透過該算法進行 cars 資料之迴歸數值預測。

- 相關套件提示
  - R 套件：randomForest
  - Python 套件：pandas, `from sklearn.ensemble import RandomForestRegressor`

10% 小題一：讀取資料

題目說明

- 請讀取 `3_cars.csv` 資料集
  - 相關說明請參考 `3_cars` 資料說明`.pdf`

答案示意

- (R)前五筆資料的輸出應如下：

	speed	dist
0	4	2
1	4	10
2	7	4
3	7	22
4	8	16

- (Python)前五筆資料的輸出應如下：

```
##    speed  dist
## 0      4     2
## 1      4    10
## 2      7     4
## 3      7    22
## 4      8    16
```

20% 小題二：切分訓練集與測試集

題目說明

- 請以資料筆數的 70%和 30%將 cars 資料隨機抽樣分為兩份，訓練集與測試集
  - 因具有隨機性質，抽樣結果可能不同，這部分不硬性規定，但若希望與答案示意結果相同可參考下方設定：
    - R: `set.seed(1)`
    - Python: `random_state=1` (此部分非必要執行)

答案示意

- (R) 輸出應如下：

```
## [1] "訓練集"

##    speed dist
## 4      7   22
## 39    20   32
## 1      4    2
## 34    18   76
## 23    14   80
## 43    20   64
## 14    12   24
## 18    13   34
## 33    18   56
## 21    14   36
## 41    20   52
## 10    11   17
## 7     10   18
## 9     10   34
## 15    12   28
## 40    20   48
## 25    15   26
## 47    24   92
## 12    12   14
## 36    19   36
## 48    24   93
## 20    14   26
## 3      7    4
## 6      9   10
## 49    24  120
## 26    15   54
## 27    16   32
## 31    17   50
## 29    17   32
## 22    14   60
## 32    18   42
## 24    15   20
## 8     10   26
## 35    18   84
## 37    19   46

## [1] "測試集"

##    speed dist
## 2      4   10
## 5      8   16
## 11    11   28
## 13    12   20
## 16    13   26
## 17    13   34
## 19    13   46
## 28    16   40
## 30    17   40
## 38    19   68
## 42    20   56
## 44    22   66
## 45    23   54
## 46    24   70
## 50    25   85
```