Python 的 50+ 練習：資料科學學習手冊

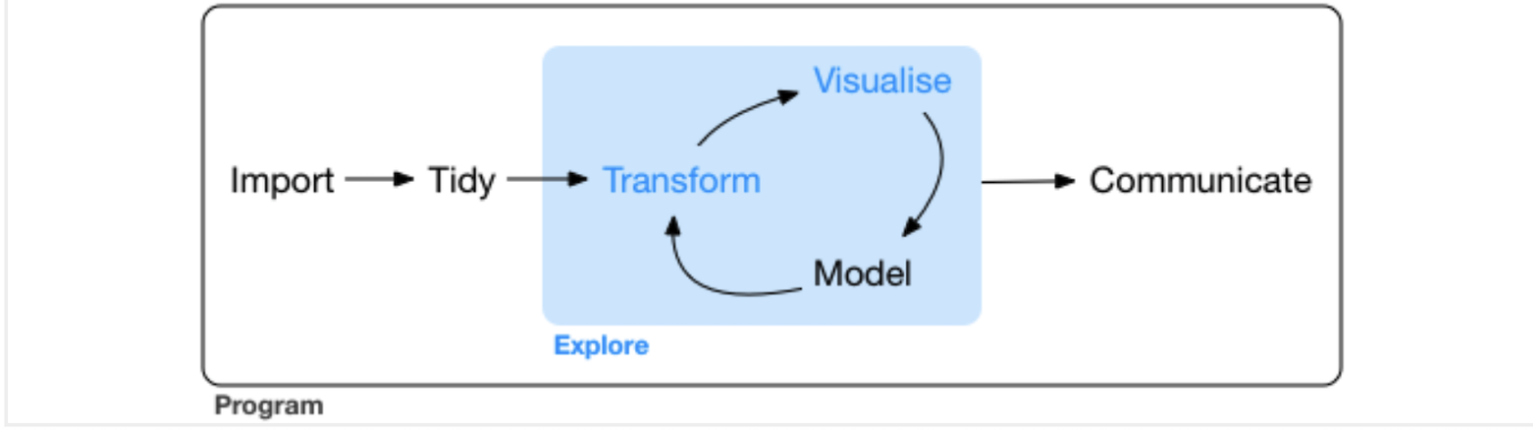
> *探索性資料分析*

數據交點 | 郭耀仁 yaojenkuo@datainpoint.com

這個章節會登場的模組

- `pandas` 模組。
- `matplotlib` 模組。

（複習）現代資料科學：以程式設計做資料科學的應用



來源：R for Data Science

（複習）什麼是資料科學的應用場景

- Import 資料的載入。
- Tidy 資料清理。
- **Transform 資料外型與類別的轉換**。
- **Visualise 探索性分析**。
- Model 分析與預測模型。
- Communicate 溝通分享。

（複習）根據說明文件的範例載入

實際上主要在使用的是 `matplotlib.pyplot`

來源：https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py

In [1]:

```
import matplotlib.pyplot as plt
```

（複習）視覺化的標準五步驟

1. 建立 `ndarray`
2. 建立「畫布物件」與「軸物件」。
3. 使用「軸物件」的作圖方法建立主要圖形。
4. 使用「軸物件」的作圖方法添加圖形元素。
5. 顯示或者儲存圖形。

軸物件不同的方法對應不同的主要圖形

- `AxesSubplot.hist()` 觀察分配的直方圖。
- `AxesSubplot.scatter()` 觀察相關的散佈圖。
- `AxesSubplot.plot()` 觀察趨勢的線圖。
- `AxesSubplot.barh()` 觀察排序的長條圖。
- …等。

不同探索目的主要圖形

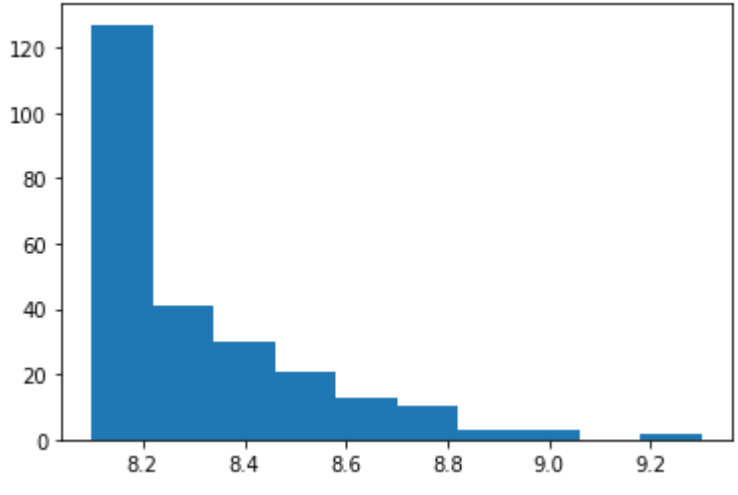
觀察分配的直方圖

`AxesSubplot.hist()` 輸入一個數值 `ndarray`

In [2]:

```
import pandas as pd

movies = pd.read_csv("/home/jovyan/data/internet-movie-database/movies.csv")
ratings = movies["rating"].values
fig, ax = plt.subplots()
ax.hist(ratings)
plt.show()
```

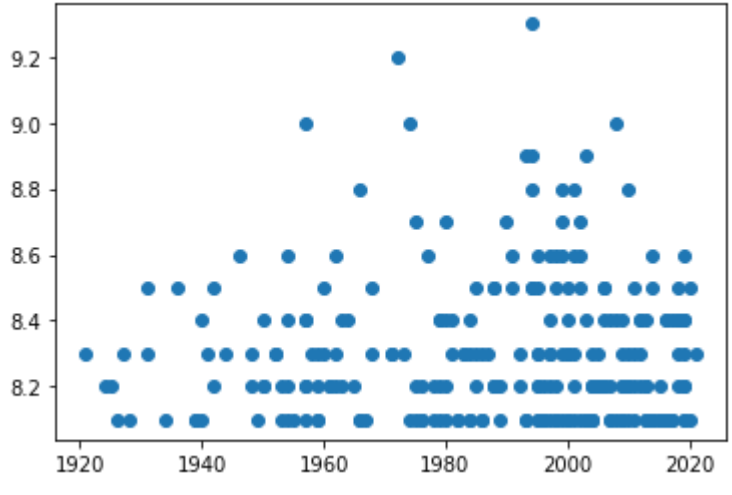


觀察相關的散佈圖

`AxesSubplot.scatter()` 輸入兩個數值 `ndarray`

In [3]:

```
release_years = movies["release_year"].values
ratings = movies["rating"].values
fig, ax = plt.subplots()
ax.scatter(release_years, ratings)
plt.show()
```

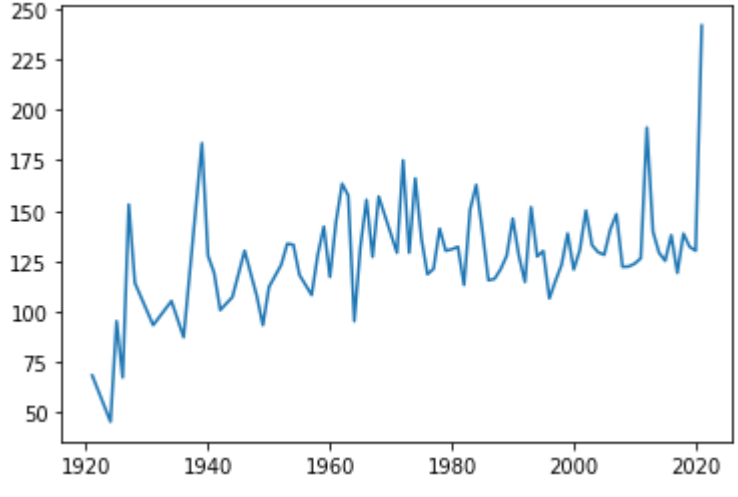


觀察趨勢的線圖

`AxesSubplot.plot()` 輸入一個日期時間、一個數值 `ndarray`

In [4]:

```
runtime_by_years = movies.groupby("release_year")["runtime"].mean()
distinct_years = runtime_by_years.index
mean_runtimes = runtime_by_years.values
fig, ax = plt.subplots()
ax.plot(distinct_years, mean_runtimes)
plt.show()
```

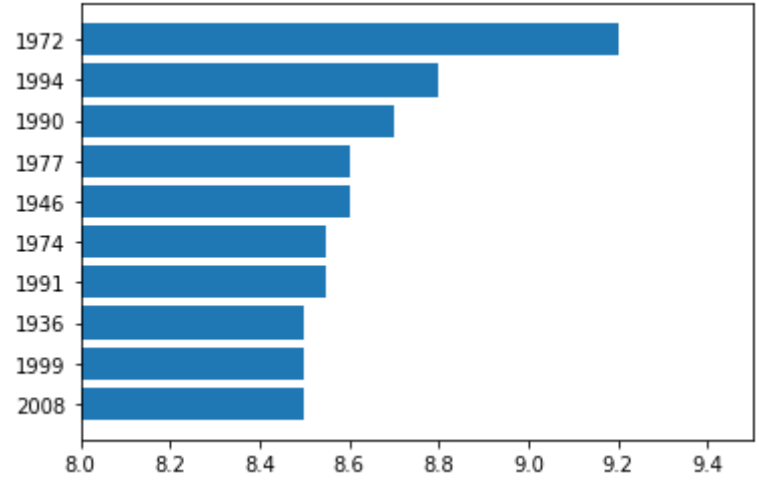


觀察排序的長條圖

`AxesSubplot.barh()` 輸入一個文字序列、一個數值 `ndarray`

In [5]:

```
rating_by_years = movies.groupby("release_year")["rating"].mean().sort_values()
top_ten_rating_by_years = rating_by_years[-10:]
distinct_years = top_ten_rating_by_years.index.astype(str)
mean_ratings = top_ten_rating_by_years.values
fig = plt.figure()
ax = plt.axes()
ax.barh(distinct_years, mean_ratings)
ax.set_xlim(8, 9.5)
plt.show()
```



添加圖形元素

常用的圖形元素

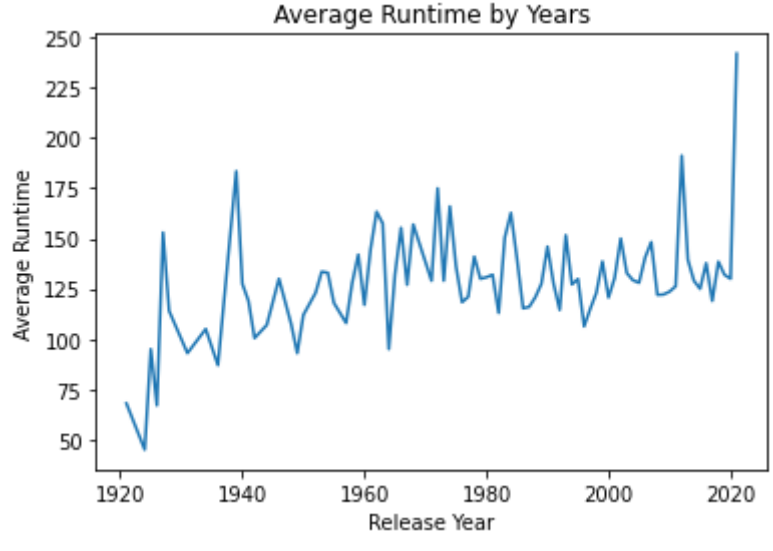
- 添加標題與軸標籤。
- 調整軸刻度。
- 調整軸刻度標籤。
- 調整軸的上下限。
- 添加文字。
- 添加圖例。

如何添加標題與軸標籤

- 使用 `AxesSubplot.set_title()` 添加標題。
- 使用 `AxesSubplot.set_xlabel()` 添加 x 軸標籤。
- 使用 `AxesSubplot.set_ylabel()` 添加 y 軸標籤。

In [6]:

```
runtime_by_years = movies.groupby("release_year")["runtime"].mean()
distinct_years = runtime_by_years.index
mean_runtimes = runtime_by_years.values
fig, ax = plt.subplots()
ax.plot(distinct_years, mean_runtimes)
ax.set_title("Average Runtime by Years") # title
ax.set_xlabel("Release Year")            # x-axis label
ax.set_ylabel("Average Runtime")         # y-axis label
plt.show()
```

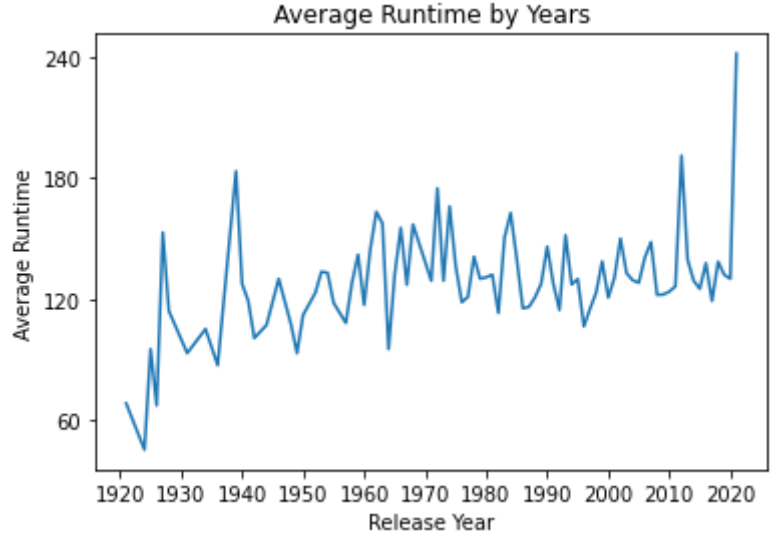


如何調整軸刻度

- 使用 `AxesSubplot.set_xticks()` 調整 x 軸刻度。
- 使用 `AxesSubplot.set_yticks()` 調整 y 軸刻度。

In [7]:

```
fig, ax = plt.subplots()
ax.plot(distinct_years, mean_runtimes)
ax.set_title("Average Runtime by Years")
ax.set_xlabel("Release Year")
ax.set_ylabel("Average Runtime")
ax.set_xticks(range(1920, 2030, 10)) # narrow ticks
ax.set_yticks([60, 120, 180, 240])   # specify ticks
plt.show()
```

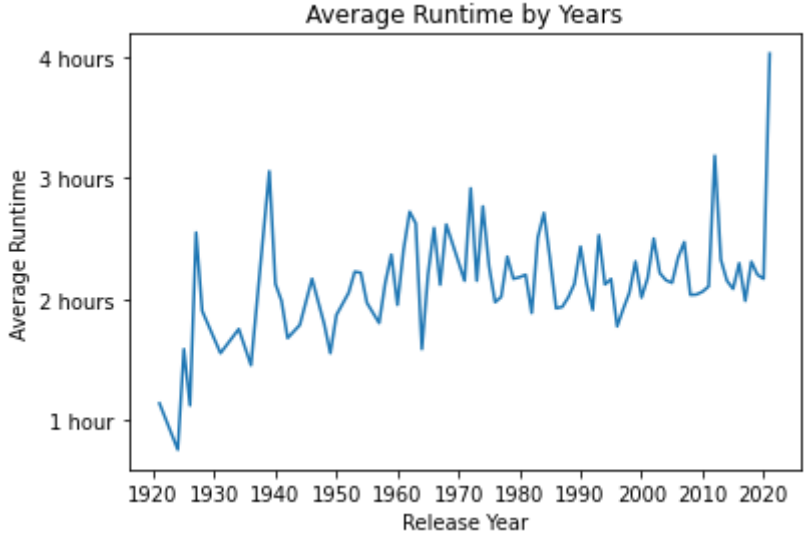


如何調整軸刻度標籤

- 使用 `AxesSubplot.set_xticklabels()` 調整 x 軸刻度標籤。
- 使用 `AxesSubplot.set_yticklabels()` 調整 y 軸刻度標籤。

In [8]:

```
fig, ax = plt.subplots()
ax.plot(distinct_years, mean_runtimes)
ax.set_title("Average Runtime by Years")
ax.set_xlabel("Release Year")
ax.set_ylabel("Average Runtime")
ax.set_xticks(range(1920, 2030, 10))
ax.set_yticks([60, 120, 180, 240])
ax.set_yticklabels(["1 hour", "2 hours", "3 hours", "4 hours"]) # specify ticklabels
plt.show()
```

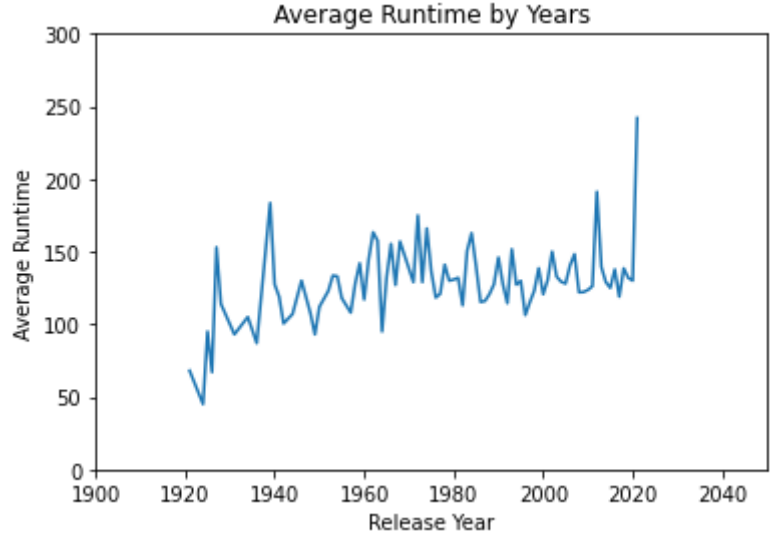


如何調整軸的上下限

- 使用 `AxesSubplot.set_xlim()` 調整 x 軸的上下限。
- 使用 `AxesSubplot.set_ylim()` 調整 y 軸的上下限。

In [9]:

```
fig, ax = plt.subplots()
ax.plot(distinct_years, mean_runtimes)
ax.set_title("Average Runtime by Years")
ax.set_xlabel("Release Year")
ax.set_ylabel("Average Runtime")
ax.set_xlim(1900, 2050) # set x-limits
ax.set_ylim(0, 300)     # set y-limits
plt.show()
```

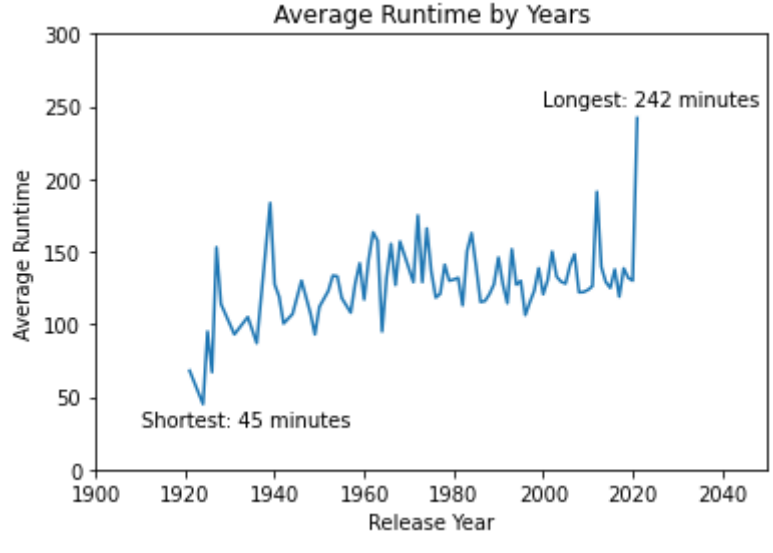


如何添加文字

使用 `AxesSubplot.text(x, y, 'Some Strings')` 在 `(x, y)` 的位置添加 `'Some Strings'`

In [10]:

```
fig, ax = plt.subplots()
ax.plot(distinct_years, mean_runtimes)
ax.set_title("Average Runtime by Years")
ax.set_xlabel("Release Year")
ax.set_ylabel("Average Runtime")
ax.set_xlim(1900, 2050) # set x-limits
ax.set_ylim(0, 300)     # set y-limits
ax.text(1910, 30, f"Shortest: {mean_runtimes.min():.0f} minutes") # add text at (1910, 30)
ax.text(2000, 250, f"Longest: {mean_runtimes.max():.0f} minutes") # add text at (2000, 250)
plt.show()
```

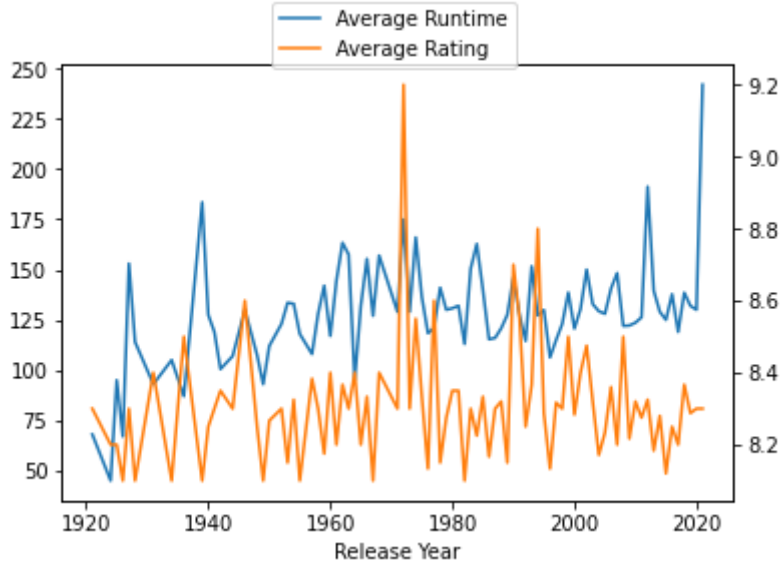


如何添加圖例

- 使用 `Figure.legend()` 並在主要圖形加上 `label` 參數。
- 因為 `runtime` 與 `rating` 的量尺不同，使用 `AxesSubplot.twinx()` 製作雙軸（Dual-axis）

In [11]:

```
runtime_by_years = movies.groupby("release_year")["runtime"].mean()
rating_by_years = movies.groupby("release_year")["rating"].mean()
distinct_years = runtime_by_years.index
mean_runtimes = runtime_by_years.values
mean_ratings = rating_by_years.values
fig, ax1 = plt.subplots()
ax1.plot(distinct_years, mean_runtimes, color="tab:blue", label="Average Runtime")
ax1.set_xlabel("Release Year")
ax2 = ax1.twinx()
ax2.plot(distinct_years, mean_ratings, color="tab:orange", label="Average Rating")
fig.legend(loc="upper center")
plt.show()
```


Average Runtime
Average Rating
250
225
200
175
150
125
100
75
50
9.2
9.0
8.8
8.6
8.4
8.2
1920
1940
1960
1980
2000
2020
Release Year

繪製子圖

什麼是子圖

子圖（Subplots）指的是在一個畫布物件上有多個軸物件的組成結構，多個軸物件就稱為子圖。

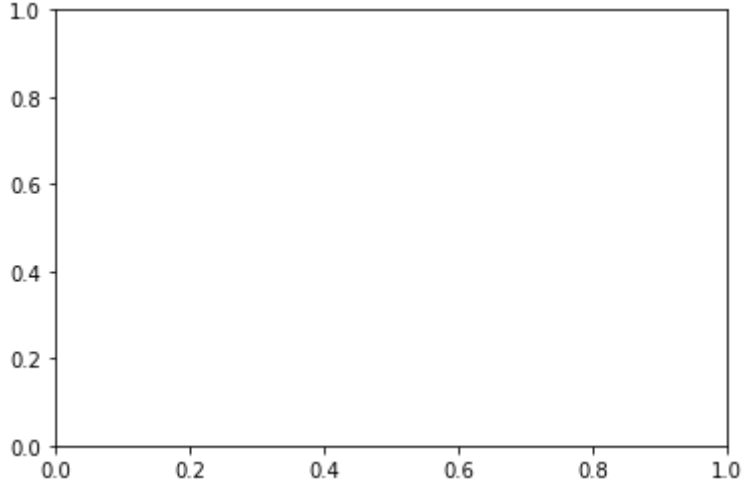
如何在一個畫布上建立子圖

使用 `plt.subplots()` 時採預設，輸出是一個 `Figure` 類別以及一個 `AxesSubplot` 類別。

In [12]:

```
fig, ax = plt.subplots() # no inputs
print(type(fig))
print(type(ax))
```

```
<class 'matplotlib.figure.Figure'>
<class 'matplotlib.axes._subplots.AxesSubplot'>
```



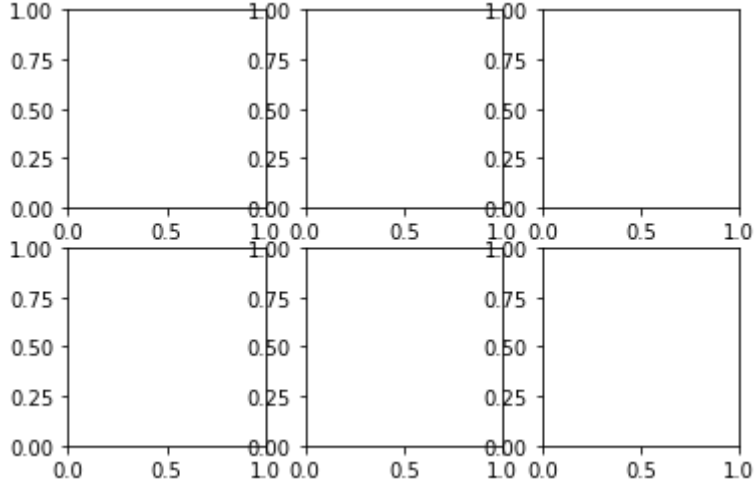
如何在一個畫布上建立子圖（續）

使用 `plt.subplots()` 時輸入 `m, n`，輸出變為一個 `Figure` 類別以及一個外型為 `(m, n)` 的 `ndarray` 類別。

In [13]:

```
fig, axes = plt.subplots(2, 3)
print(type(fig))
print(type(axes))
```

```
<class 'matplotlib.figure.Figure'>
<class 'numpy.ndarray'>
```



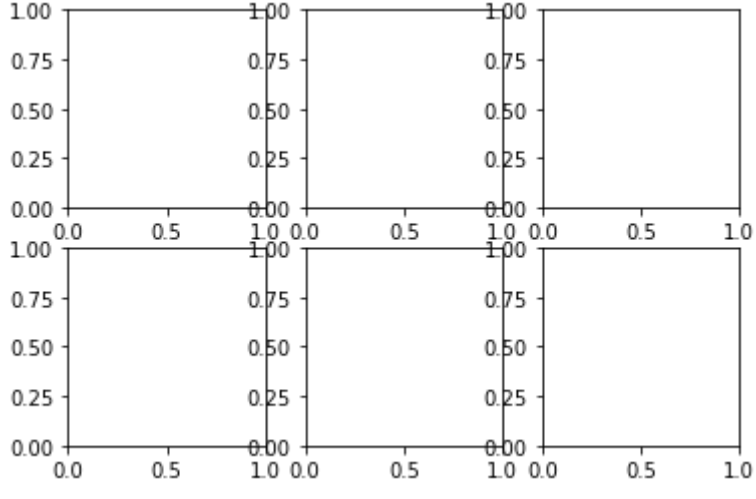
可以使用作圖方法、添加圖形元素的軸物件到哪了呢

- 在 `ndarray` 中有 $m \times n$ 個軸物件。
- 以 `ndarray[row, column]` 就能取得位於 `(row, column)` 的軸物件。

In [14]:

```
m = 2
n = 3
fig, axes = plt.subplots(m, n)
print(axes.shape)
for row in range(m):
    for column in range(n):
        print(type(axes[row, column]))
```

```
(2, 3)
<class 'matplotlib.axes._subplots.AxesSubplot'>
<class 'matplotlib.axes._subplots.AxesSubplot'>
<class 'matplotlib.axes._subplots.AxesSubplot'>
<class 'matplotlib.axes._subplots.AxesSubplot'>
<class 'matplotlib.axes._subplots.AxesSubplot'>
<class 'matplotlib.axes._subplots.AxesSubplot'>
```

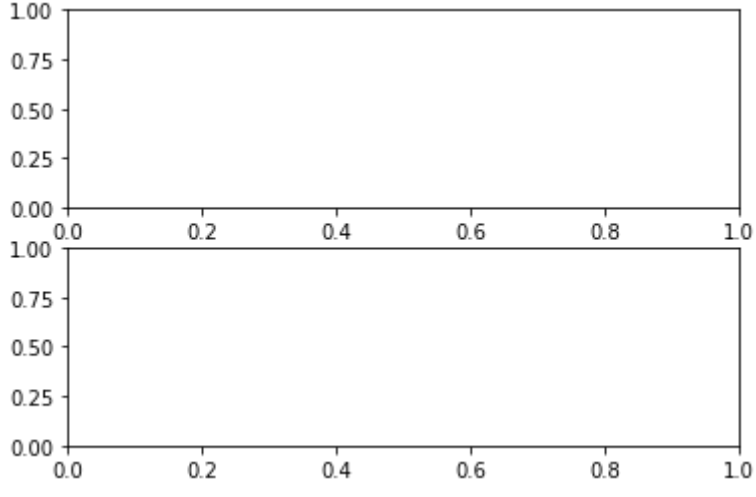


In [15]:

```
fig, axes = plt.subplots(2, 1)
axes.shape
```

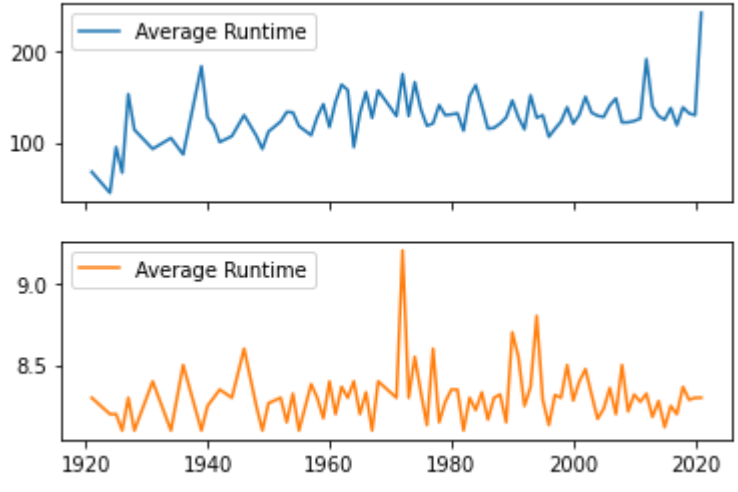
Out[15]:

```
(2,)
```



In [16]:

```
fig, axes = plt.subplots(2, 1, sharex=True)
axes[0].plot(distinct_years, mean_runtimes, color="tab:blue", label="Average Runtime")
axes[0].legend(loc="upper left")
axes[1].plot(distinct_years, mean_ratings, color="tab:orange", label="Average Runtime")
axes[1].legend(loc="upper left")
plt.show()
```



重點統整

- 軸物件不同的方法對應不同的主要圖形
    - `AxesSubplot.hist()` 觀察分配的直方圖。
    - `AxesSubplot.scatter()` 觀察相關的散佈圖。
    - `AxesSubplot.plot()` 觀察趨勢的線圖。
    - `AxesSubplot.bar()` 觀察排序的長條圖。

重點統整（續）

- 常用的圖形元素
    - 添加標題與軸標籤。
    - 調整軸刻度。
    - 調整軸刻度標籤。
    - 調整軸的上下限。
    - 添加文字。
    - 添加圖例。
- 使用 `plt.subplots()` 時輸入 `m, n`，輸出變為一個 `Figure` 類別以及一個外型為 `(m, n)` 的 `ndarray` 類別，在 `ndarray` 中有 $m \times n$ 個軸物件。