Python 的 50+ 練習：資料科學學習手冊

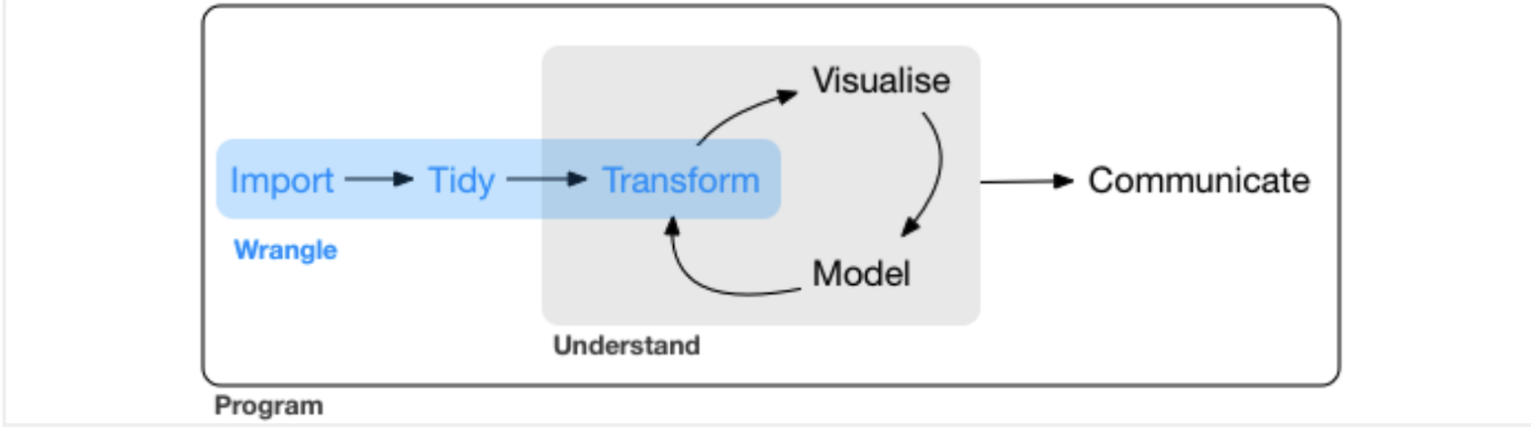
> *進階資料框操作*

數據交點 | 郭耀仁 yaojenkuo@datainpoint.com

這個章節會登場的模組

- `numpy` 模組。
- `pandas` 模組。

（複習）現代資料科學：以程式設計做資料科學的應用

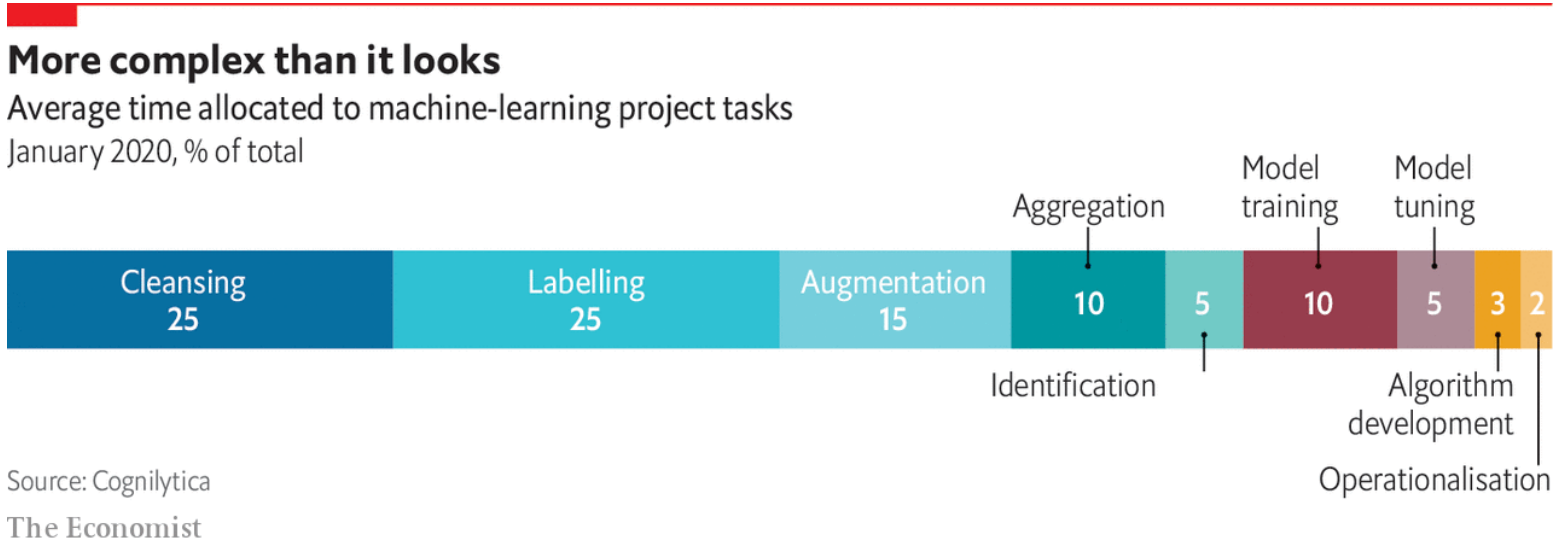


來源：R for Data Science

（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）機器學習專案花費 50% 的時間處理 Data Wrangle 的相關任務



來源：https://www.economist.com/technology-quarterly/2020/06/11/for-ai-data-are-harder-to-come-by-than-you-think

進階的資料框操作包含

- 迭代資料框的列或欄。
- 處理未定義值。
- 轉置資料框。
- 單純合併資料框。
- 關聯資料框。

迭代資料框的列或欄

如何迭代資料框的欄

使用 `DataFrame.iteritems()` 同時取得欄標籤以及變數（ `Series` 形式）

In [1]:

```
import pandas as pd

movies = pd.read_csv("/home/jovyan/data/internet-movie-database/movies.csv")
print(movies.dtypes)
```

```
id                int64
title            object
release_year      int64
rating          float64
director         object
runtime           int64
dtype: object
```

In [2]:

```
for column_label, series in movies.iteritems():
    print(f"{column_label}: {series.dtype}: {type(series)}")
```

```
id: int64: <class 'pandas.core.series.Series'>
title: object: <class 'pandas.core.series.Series'>
release_year: int64: <class 'pandas.core.series.Series'>
```

```
rating: float64: <class 'pandas.core.series.Series'>
director: object: <class 'pandas.core.series.Series'>
runtime: int64: <class 'pandas.core.series.Series'>
```

如何迭代資料框的列

使用 `DataFrame.iterrows()` 同時取得列標籤以及對應的觀測值（ `Series` 形式）。

In [3]:

```
for row_label, series in movies.head().iterrows(): # iterate the first 5 rows of movies
    print(f"{row_label}: {series.values}: {type(series)}")
```

```
0: [1 'The Shawshank Redemption' 1994 9.3 'Frank Darabont' 14
2]: <class 'pandas.core.series.Series'>
1: [2 'The Godfather' 1972 9.2 'Francis Ford Coppola' 175]: <
class 'pandas.core.series.Series'>
2: [3 'The Godfather: Part II' 1974 9.0 'Francis Ford Coppol
a' 202]: <class 'pandas.core.series.Series'>
3: [4 'The Dark Knight' 2008 9.0 'Christopher Nolan' 152]: <c
lass 'pandas.core.series.Series'>
4: [5 '12 Angry Men' 1957 9.0 'Sidney Lumet' 96]: <class 'pan
das.core.series.Series'>
```

處理未定義值

常見未定義值的處理方式

- 檢查未定義值是否存在。
- 刪除未定義值。
- 填補未定義值。

不能使用 `Series == np.nan` 來檢查未定義值是否存在

In [4]:

```
import numpy as np

series_with_npnan = pd.Series([2, 3, np.nan, 7, 11, np.nan, 17, 19])
series_with_npnan == np.nan
```

Out[4]:

```
0    False
1    False
2    False
3    False
4    False
5    False
6    False
7    False
dtype: bool
```

使用 `Series.isnull()` 或者 `Series.notnull()` 檢查未定義值是否存在

In [5]:
```
series_with_npnan.isnull()
```

Out[5]:
```
0    False
1    False
2     True
3    False
4    False
5     True
6    False
7    False
dtype: bool
```

In [6]:
```
series_with_npnan.notnull() # ~(series_with_npnan.isnull())
```

Out[6]:
```
0     True
1     True
```

```
2    False
3     True
4     True
5    False
6     True
7     True
dtype: bool
```

使用 `Series.dropna()` 删除

In [7]:

```
series_with_npnan = pd.Series([2, 3, np.nan, 7, 11, np.nan, 17, 19])
series_with_npnan.dropna()
```

Out[7]:

```
0     2.0
1     3.0
3     7.0
4    11.0
6    17.0
7    19.0
dtype: float64
```

使用 `DataFrame.dropna()` 刪除有未定義值的資料列或欄位

- 因為資料框的外型是 `(m, n)` 因此必須整個資料列或欄位都刪除。
- `axis` 參數預設為 `axis=0` 刪除有未定義值的資料列，指定參數 `axis=1` 刪除有未定義值的欄位。
- 指定參數 `how="any"` 只要資料列或欄位有一個 `np.nan` 就將整個資料列或欄位刪除。
- 指定參數 `how="all"` 必須資料列或欄位全部都是 `np.nan` 才刪除。

In [8]:

```
dataframe_with_npnan = pd.DataFrame()
dataframe_with_npnan["column_0"] = [2, 3, 5, 7, 11, np.nan]
dataframe_with_npnan["column_1"] = [2, np.nan, 5, np.nan, 11, np.nan]
dataframe_with_npnan["column_2"] = [2, 3, np.nan, 7, np.nan, np.nan]
dataframe_with_npnan
```

Out[8]:

	column_0	column_1	column_2
0	2.0	2.0	2.0
1	3.0	NaN	3.0
2	5.0	5.0	NaN
3	7.0	NaN	7.0
4	11.0	11.0	NaN
5	NaN	NaN	NaN

刪除有未定義值的資料列

In [9]:
```
dataframe_with_npnan.dropna()
```

Out[9]:

	column_0	column_1	column_2
0	2.0	2.0	2.0

In [10]:
```
dataframe_with_npnan.dropna(how="all")
```

Out[10]:

	column_0	column_1	column_2
0	2.0	2.0	2.0
1	3.0	NaN	3.0
2	5.0	5.0	NaN
3	7.0	NaN	7.0
4	11.0	11.0	NaN

刪除有未定義值的欄位

In [11]:

```
dataframe_with_npnan = pd.DataFrame()
dataframe_with_npnan["column_0"] = [2, 3, 5, 7, 11, 13]
dataframe_with_npnan["column_1"] = [2, np.nan, 5, np.nan, 11, np.nan]
dataframe_with_npnan["column_2"] = [2, 3, np.nan, 7, np.nan, np.nan]
dataframe_with_npnan["column_3"] = [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]
dataframe_with_npnan
```

Out[11]:

	column_0	column_1	column_2	column_3
0	2	2.0	2.0	NaN
1	3	NaN	3.0	NaN
2	5	5.0	NaN	NaN
3	7	NaN	7.0	NaN
4	11	11.0	NaN	NaN
5	13	NaN	NaN	NaN

In [12]:

```
dataframe_with_npnan.dropna(axis=1)
```

Out[12]:

	column_0
0	2
1	3
2	5
3	7
4	11
5	13

In [13]:

```
dataframe_with_npnan.dropna(axis=1, how="all")
```

Out[13]:

	column_0	column_1	column_2
0	2	2.0	2.0
1	3	NaN	3.0
2	5	5.0	NaN
3	7	NaN	7.0
4	11	11.0	NaN
5	13	NaN	NaN

使用 `Series.fillna()` 填補未定義值為指定資料

In [14]:

```
series_with_npnan = pd.Series([2, 3, np.nan, 7, 11, np.nan, np.nan, 19])
series_with_npnan.fillna(5566)
```

Out[14]:

```
0       2.0
1       3.0
2    5566.0
3       7.0
4      11.0
5    5566.0
6    5566.0
7      19.0
dtype: float64
```

使用 `Series.fillna()` 的參數 `method`

- `method="ffill"` (forward-fill)遇到未定義值就用前一個值來填補。
- `method="bfill"` (backward-fill)遇到未定義值就用後一個值來填補。

In [15]:
```
series_with_npnan.fillna(method="ffill")
```

Out[15]:
```
0     2.0
1     3.0
2     3.0
3     7.0
4    11.0
5    11.0
6    11.0
7    19.0
dtype: float64
```

In [16]:
```
series_with_npnan.fillna(method="bfill")
```

Out[16]:

```
0     2.0
1     3.0
2     7.0
3     7.0
4    11.0
5    19.0
6    19.0
7    19.0
dtype: float64
```

轉置資料框

轉置資料框的函數與方法

- `pd.melt()` 函數可以將資料框由寬格式轉換為長格式。
- `DataFrame.pivot()` 可以將資料框由長格式轉換為寬格式。

什麼是寬格式、長格式

- 寬格式使用一個欄位儲存資料，欄位名稱記錄變數類別、觀測值記錄數值。
- 長格式使用兩個欄位儲存資料，一個欄位記錄變數類別、一個欄位記錄數值。

來源：https://en.wikipedia.org/wiki/Wide_and_narrow_data

寬格式使用欄位名稱記錄變數類別、觀測值記錄其數值

In [17]:

```
persons = ["Bob", "Alice", "Steve"]
ages = [32, 24, 64]
weights = [168, 150, 144]
heights = [180, 175, 165]
wide_format = pd.DataFrame()
wide_format["Person"] = persons
wide_format["Age"] = ages
wide_format["Weight"] = weights
wide_format["Height"] = heights
wide_format
```

Out[17]:

	Person	Age	Weight	Height
0	Bob	32	168	180
1	Alice	24	150	175
2	Steve	64	144	165

長格式使用一個欄位記錄變數類別、一個欄位記錄數值

In [19]:
```
long_format
```

Out[19]:

	Person	Variable	Value
0	Bob	Age	32
1	Alice	Age	24
2	Steve	Age	64
3	Bob	Weight	168
4	Alice	Weight	150
5	Steve	Weight	144
6	Bob	Height	180
7	Alice	Height	175
8	Steve	Height	165

`pd.melt()` 函數可以將資料框由寬格式轉換為長格式

- `id_vars` 參數指定非數值的欄位。
- `value_vars` 參數指定數值的欄位，預設指定 `id_vars` 以外的所有欄位。
- `var_name` 參數指定變數類別的欄標籤。
- `value_name` 參數指定數值的欄標籤。

In [20]:
```
pd.melt(wide_format, id_vars="Person", var_name="Variable", value_name="Value") # value_vars=["Age", "Weight",
```

Out[20]:

	Person	Variable	Value
0	Bob	Age	32
1	Alice	Age	24
2	Steve	Age	64
3	Bob	Weight	168
4	Alice	Weight	150
5	Steve	Weight	144
6	Bob	Height	180
7	Alice	Height	175
8	Steve	Height	165

`DataFrame.pivot()` 可以將資料框由長格式轉換為寬格式

- `index` 參數指定非數值的欄位。
- `columns` 參數指定變數類別，對應 `pd.melt()` 的 `var_name`
- `values` 參數指定數值，對應 `pd.melt()` 的 `value_name`

In [21]:
```
long_format.pivot(index="Person", columns="Variable", values="Value")
```

Out[21]:

Variable	Age	Height	Weight
Person			
Alice	24	175	150
Bob	32	180	168
Steve	64	165	144

為何需要轉置資料框

- 資料框的欄位名稱含有使用者需要的資料值。
- 儲存格式與應用情境不符。

単純合併資料框

單純合併資料框的函數與方法

- 使用 `DataFrame.append()` 單純垂直合併。
- 使用 `pd.concat()` 函數單純垂直與水平合併。

要進行單純垂直合併的資料框

In [22]:

```
upper_df = pd.DataFrame()
upper_df["title"] = ["The Lord of the Rings: The Fellowship of the Rings"]
upper_df["release_year"] = [2001]
upper_df
```

Out[22]:

	title	release_year
0	The Lord of the Rings: The Fellowship of the R...	2001

In [23]:

```
lower_df = pd.DataFrame()
lower_df["title"] = ["The Lord of the Rings: The Two Towers", "The Lord of the Rings: The Return of the King"]
lower_df["release_year"] = [2002, 2003]
lower_df
```

Out[23]:

	title	release_year
0	The Lord of the Rings: The Two Towers	2002
1	The Lord of the Rings: The Return of the King	2003

使用 `DataFrame.append()` 單純垂直合併

- 注意更新機制，將更新的結果回傳（Return），物件本身維持不變。
- 預設不會重設列標籤，可以指定參數 `ignore_index=True` 重設列標籤

In [24]:
```
upper_df.append(lower_df)
```

Out[24]:

	title	release_year
0	The Lord of the Rings: The Fellowship of the R...	2001
0	The Lord of the Rings: The Two Towers	2002
1	The Lord of the Rings: The Return of the King	2003

In [25]:
```
upper_df.append(lower_df, ignore_index=True)
```

Out[25]:

	title	release_year
0	The Lord of the Rings: The Fellowship of the R...	2001
1	The Lord of the Rings: The Two Towers	2002
2	The Lord of the Rings: The Return of the King	2003

使用 `pd.concat()` 函數單純垂直與水平合併

- 將要合併的資料框以結構傳入。
- 參數預設 `axis=0` 單純垂直合併。
- 預設不會重設列標籤，可以指定參數 `ignore_index=True` 重設列標籤

In [26]:
```
pd.concat((upper_df, lower_df))
```

Out[26]:

	title	release_year
0	The Lord of the Rings: The Fellowship of the R...	2001
0	The Lord of the Rings: The Two Towers	2002
1	The Lord of the Rings: The Return of the King	2003

In [27]:
```
pd.concat((upper_df, lower_df), ignore_index=True)
```

Out[27]:

	title	release_year
0	The Lord of the Rings: The Fellowship of the R...	2001
1	The Lord of the Rings: The Two Towers	2002
2	The Lord of the Rings: The Return of the King	2003

要進行單純水平合併的資料框

In [28]:

```
left_df = pd.DataFrame()
left_df["title"] = ["The Lord of the Rings: The Fellowship of the Rings", "The Lord of the Rings: The Two Tower
left_df
```

Out[28]:

	title
0	The Lord of the Rings: The Fellowship of the R...
1	The Lord of the Rings: The Two Towers
2	The Lord of the Rings: The Return of the King

In [29]:

```
right_df = pd.DataFrame()
right_df["release_year"] = [2001, 2002, 2003]
right_df
```

Out[29]:

	release_year
0	2001
1	2002
2	2003

参数指定 `axis=1` 单純水平合併

In [30]:

```
pd.concat((left_df, right_df), axis=1)
```

Out[30]:

	title	release_year
0	The Lord of the Rings: The Fellowship of the R...	2001
1	The Lord of the Rings: The Two Towers	2002
2	The Lord of the Rings: The Return of the King	2003

關聯資料框

什麼是關聯資料框

資料框的列標籤、欄標籤都是 `Index` 類別，由於 `Index` 類別支援集合運算的特性，因此資料框之間能夠進行觀測值與變數的交集、聯集的運算，實踐類似資料庫的關聯式模型，包含與交集對應的內部連接（Inner join）、左外部連接（Left join）、右外部連接（Right join）以及與聯集對應的全外部連接（Full join）。

四種連接

1. 內部連接（Inner join），保留兩個關聯資料框交集的觀測值。
2. 左外部連接（Left join），保留左資料框所有的觀測值以及兩個關聯資料框交集的觀測值。
3. 右外部連接（Right join），保留右資料框所有的觀測值以及兩個關聯資料框交集的觀測值。
4. 全外部連接（Full join），保留兩個關聯資料框聯集的觀測值。

關聯資料框的函數與方法

- 使用 `pd.merge()` 函數利用欄標籤進行四種連接。
- 使用 `DataFrame.join()` 利用列標籤進行四種連接。

欲關聯的兩個資料框

In [31]:

```
left_df = pd.DataFrame()
left_df["title"] = ["The Shawshank Redemption", "The Dark Knight", "The Lord of the Rings: The Return of the Ki
left_df["rating"] = [9.3, 9.0, 8.9, 8.9, 8.8]
left_df
```

Out[31]:

	title	rating
0	The Shawshank Redemption	9.3
1	The Dark Knight	9.0
2	The Lord of the Rings: The Return of the King	8.9
3	Schindler's List	8.9
4	Forrest Gump	8.8

In [32]:

```
right_df = pd.DataFrame()
right_df["title"] = ["The Lord of the Rings: The Fellowship of the Rings", "The Lord of the Rings: The Two Towe
right_df["release_year"] = [2001, 2002, 2003, 2005, 2008, 2012]
right_df
```

Out[32]:

	title	release_year
0	The Lord of the Rings: The Fellowship of the R...	2001
1	The Lord of the Rings: The Two Towers	2002
2	The Lord of the Rings: The Return of the King	2003
3	Batman Begins	2005
4	The Dark Knight	2008
5	The Dark Knight Rises	2012

使用 `pd.merge()` 函數利用欄標籤進行四種連接

In [33]:

```
# default: inner join
pd.merge(left_df, right_df)
```

Out[33]:

	title	rating	release_year
0	The Dark Knight	9.0	2008
1	The Lord of the Rings: The Return of the King	8.9	2003

In [34]:

```
# left join
pd.merge(left_df, right_df, how='left')
```

Out[34]:

	title	rating	release_year
0	The Shawshank Redemption	9.3	NaN
1	The Dark Knight	9.0	2008.0
2	The Lord of the Rings: The Return of the King	8.9	2003.0
3	Schindler's List	8.9	NaN
4	Forrest Gump	8.8	NaN

In [35]:

```
# right join
pd.merge(left_df, right_df, how='right')
```

Out[35]:

	title	rating	release_year
0	The Lord of the Rings: The Fellowship of the R...	NaN	2001
1	The Lord of the Rings: The Two Towers	NaN	2002
2	The Lord of the Rings: The Return of the King	8.9	2003
3	Batman Begins	NaN	2005
4	The Dark Knight	9.0	2008
5	The Dark Knight Rises	NaN	2012

In [36]:

```
# full join
pd.merge(left_df, right_df, how='outer')
```

Out[36]:

	title	rating	release_year
0	The Shawshank Redemption	9.3	NaN
1	The Dark Knight	9.0	2008.0
2	The Lord of the Rings: The Return of the King	8.9	2003.0
3	Schindler's List	8.9	NaN
4	Forrest Gump	8.8	NaN
5	The Lord of the Rings: The Fellowship of the R...	NaN	2001.0
6	The Lord of the Rings: The Two Towers	NaN	2002.0
7	Batman Begins	NaN	2005.0
8	The Dark Knight Rises	NaN	2012.0

使用 `DataFrame.join()` 利用列標籤進行四種連接

In [37]:

```
# default: left join
left_df.set_index('title').join(right_df.set_index('title'))
```

Out[37]:

title	rating	release_year
The Shawshank Redemption	9.3	NaN
The Dark Knight	9.0	2008.0
The Lord of the Rings: The Return of the King	8.9	2003.0
Schindler's List	8.9	NaN
Forrest Gump	8.8	NaN

In [38]:

```
# inner join
left_df.set_index('title').join(right_df.set_index('title'), how="inner")
```

Out[38]:

title	rating	release_year
The Dark Knight	9.0	2008
The Lord of the Rings: The Return of the King	8.9	2003

In [39]:

```
# right join
left_df.set_index('title').join(right_df.set_index('title'), how="right")
```

Out[39]:

title	rating	release_year
The Lord of the Rings: The Fellowship of the Rings	NaN	2001
The Lord of the Rings: The Two Towers	NaN	2002
The Lord of the Rings: The Return of the King	8.9	2003
Batman Begins	NaN	2005
The Dark Knight	9.0	2008
The Dark Knight Rises	NaN	2012

In [40]:

```
# outer join
left_df.set_index('title').join(right_df.set_index('title'), how="outer")
```

Out[40]:

title	rating	release_year
Batman Begins	NaN	2005.0
Forrest Gump	8.8	NaN
Schindler's List	8.9	NaN
The Dark Knight	9.0	2008.0
The Dark Knight Rises	NaN	2012.0
The Lord of the Rings: The Fellowship of the Rings	NaN	2001.0
The Lord of the Rings: The Return of the King	8.9	2003.0
The Lord of the Rings: The Two Towers	NaN	2002.0
The Shawshank Redemption	9.3	NaN

重點統整

- 常見未定義值的處理方式
    - `Series.isnull()` 檢查未定義值是否存在。
    - `Series.dropna()`、`DataFrame.dropna()` 刪除未定義值。
    - `Series.fillna()` 填補未定義值。
- 什麼是寬格式、長格式
    - 寬格式使用一個欄位儲存資料，欄位名稱記錄變數類別、觀測值記錄數值。
    - 長格式使用兩個欄位儲存資料，一個欄位記錄變數類別、一個欄位記錄數值。

重點統整（續）

- `pd.melt()` 函數可以將資料框由寬格式轉換為長格式。
- `DataFrame.pivot()` 可以將資料框由長格式轉換為寬格式
- 單純合併資料框的函數與方法
    - 使用 `DataFrame.append()` 單純垂直合併。
    - 使用 `pd.concat()` 函數單純垂直與水平合併。

重點統整（續）

- 關聯資料框的四種連接
    - 內部連接（Inner join），保留兩個關聯資料框交集的觀測值。
    - 左外部連接（Left join），保留左資料框所有的觀測值以及兩個關聯資料框交集的觀測值。
    - 右外部連接（Right join），保留右資料框所有的觀測值以及兩個關聯資料框交集的觀測值。
    - 全外部連接（Full join），保留兩個關聯資料框聯集的觀測值。
- 使用 `pd.merge()` 函數利用欄標籤進行四種連接。
- 使用 `DataFrame.join()` 利用列標籤進行四種連接。