# Python 系列 – MIMIC-IV Demo 資料讀取及介紹

鄭哲宇 副統計分析師

近年來 Python 在人工智慧、機器學習與數據分析等領域有非常多的應用，然而在進行上述任一項分析工作前，我們都需要先載入並且預先處理好資料。取得正確並且整齊的資料，是資料分析不可或缺的第一個步驟，因此 Python 中提供許多讀取資料及處理資料的方法。本期 eNews 將使用 MIMIC-IV Demo 資料作為範例，首先示範如何下載 MIMIC-IV Demo 資料，並且簡單說明其資料欄位及資料結構。完成資料下載後，本文將介紹如何使用 Python 常用的套件來讀取檔案，並對讀入資料進行簡單的檢視及資料篩選等工作。

## 第 1 部分 MIMIC-IV demo data 簡介

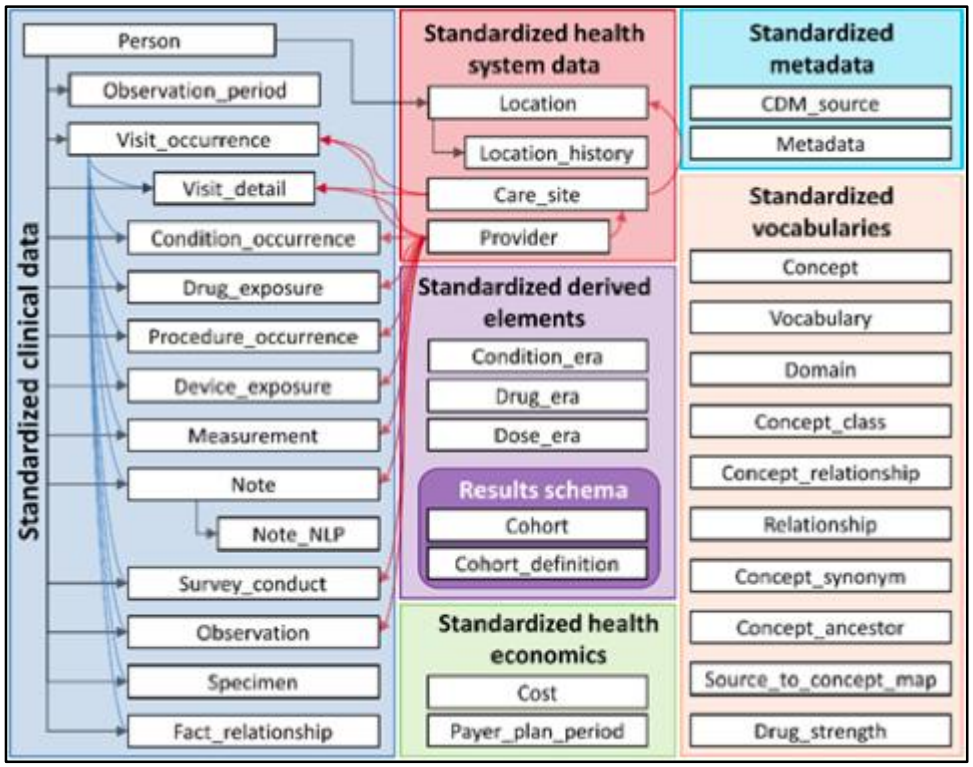
### 1.1 下載 MIMIC-IV demo data

eNews 第 48 期中，對 MIMIC 資料庫有詳細的介紹，並提到目前 MIMIC 官方有釋出 100 筆病患資料做為 demo 檔案。讀者可以前往下列網址：https://physionet.org/content/mimic-iv-demo-omop/0.9/，捲動至網頁下半部後，點選”Download the ZIP file”來下載資料檔。

## 1.2 OMOP-CDM 通用資料模型是什麼?

完成下載資料後，可以發現資料夾中有許多不同的檔案，本文主要會使用到檔案夾 1_omop_data_csv 當中的檔案。資料夾中有各個不同檔案是因為目前我們下載的資料使用 OMOP-CDM 通用資料模型(Common data model, CDM)的規範來儲存。

OMOP-CDM 通用資料模型是國際上統一的資料標準，規範觀察性資料的格式和內容，支援不同來源的觀察性資料。通過資料的擷取、轉換和載入(Extraction-Transformation-Loading，ETL)過程，形成標準化的數據結構，在此基礎上可以進行資料的查詢和分析等應用。簡單來說，OMOP-CMD 通用資料模型將不同訊息的資料，放在不同的資料檔案中，例如表單 Person 存放了有關病人本身相關訊息資料，而 Drug_exposure 表單則儲存了病患使用藥物的相關紀錄。有關 OMOP-CDM 通用資料模型的詳細使用方式，可前往其官方網站 https://www.ohdsi.org/data-standardization/the-common-data-model/，內有更詳盡的說明。



# 第 2 部分 使用 Python 讀取資料

## 2.1 開啟 Jupyter Notebook

本文將使用 eNews 第 47 期所介紹安裝的 Jupyter Notebook 作為示範 Python 程式碼的界面，建議尚未安裝 Jupyter Notebook 的讀者可先參考 eNews 第 47 期介紹方式安裝。順利安裝並開啟 Jupyter Notebook 之後，我們首先要在程式碼中匯入 Pandas 程式庫。

Pandas 是 Python 中擁有高階資料結構及工具的程式庫，適合用來幫助使用者整理資料並進行資料分析。本文將著重介紹應用 Pandas 來讀取資料、進行基礎的資料合併功能，幫助讀者把現有資料準備成適合分析的資料格式，有關 Pandas 程式庫的詳細介紹或是更多學習資源，可以參考下列網站：https://pandas.pydata.org/。

```
1 #匯入pandas程式庫
2 import pandas as pd
```

## 2.2 載入 MIMIC-IV demo data

Pandas 程式庫當中，提供許多工具方便讀取各種資料格式，本文以 MIMIC-IV demo data 作為範例，示範如何讀取 CSV 格式的資料檔。如下圖，在函數 pd.read.csv()中，放入欲讀取檔案之路徑，就可以在 Python 中順利讀取資料夾中的 person.csv 檔案，並將資料儲存為一個名叫 person 的變數。

單純執行變數 person，可以看到程式中列出了我們剛剛讀取的檔案內容，如果沒有正確顯示，則要再回頭檢查 pd.read.csv()中，填寫的檔案之路徑是否有錯誤。我們可以看到 person 資料檔案中，儲存了資料中每一個樣本人數的 ID、出生年等等相關欄位，資料共有 100 列對應的就是 MIMIC-IV demo data 中的 100 個病患。

```
person = pd.read_csv("C:/Users/biostat/Desktop/1_omop_data_csv/person.csv")
person
```

| | person_id | gender_concept_id | year_of_birth | month_of_birth | day_of_birth | birth_datetime | race_concept_id | ethnicity_concept_id | location_id | pr |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 3589912774911670296 | 8507 | 2095 | NaN | NaN | NaN | 0 | 38003563 | NaN | |
| 1 | -3210373572193940939 | 8507 | 2079 | NaN | NaN | NaN | 0 | 38003563 | NaN | |
| 2 | -775517641933593374 | 8507 | 2149 | NaN | NaN | NaN | 8516 | 0 | NaN | |
| 3 | -2575767131279873665 | 8507 | 2050 | NaN | NaN | NaN | 8516 | 0 | NaN | |
| 4 | -8970844422700220177 | 8507 | 2114 | NaN | NaN | NaN | 8527 | 0 | NaN | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 95 | -7671795861352464589 | 8532 | 2052 | NaN | NaN | NaN | 2000001401 | 0 | NaN | |
| 96 | 5734523979606454056 | 8532 | 2069 | NaN | NaN | NaN | 2000001401 | 0 | NaN | |
| 97 | 1532249960797525190 | 8532 | 2106 | NaN | NaN | NaN | 2000001405 | 0 | NaN | |
| 98 | 5894416985828315484 | 8532 | 2055 | NaN | NaN | NaN | 2000001405 | 0 | NaN | |
| 99 | -3780452582396805474 | 8532 | 2058 | NaN | NaN | NaN | 2000001405 | 0 | NaN | |

100 rows × 18 columns

## 2.3 檢視資料框

資料框(Data Frame)是 Pandas 程式庫中相當重要的一種資料結構，通常在做資料分析之前，我們會將原始資料整理成一個或數個資料框以利後續的不同分析方法，因此資料框可以說是資料分析的起點。

資料框將資料分為兩個維度：欄(Column)和列(Row)，每一欄代表著資料的不同屬性，例如先前讀取的 person.csv 檔案中，ID 和出生年都代表著一種觀察值的屬性；而每一列資料，可以被視為一筆觀察值，例如 person.csv 檔案中，每一列都代表著一個不同病患的資料。

我們使用 pd.read.csv()將資料讀取後，儲存的變數 person 就是一個資料框，如果要獲取 person 資料框的更多資訊，則可以用下列程式碼：

1. 資料框的基本資訊

```
#person是一個data.frame
type(person)
```

```
pandas.core.frame.DataFrame
```

```
#資料框的行名
person.columns
```

```
Index(['person_id', 'gender_concept_id', 'year_of_birth', 'month_of_birth',
       'day_of_birth', 'birth_datetime', 'race_concept_id',
       'ethnicity_concept_id', 'location_id', 'provider_id', 'care_site_id',
       'person_source_value', 'gender_source_value',
       'gender_source_concept_id', 'race_source_value',
       'race_source_concept_id', 'ethnicity_source_value',
       'ethnicity_source_concept_id'],
      dtype='object')
```

```
#檢視資料形狀
person.shape
```

```
(100, 18)
```

2. 檢視資料前五列

```
#檢視資料前五列
person.head
```

```
<bound method NDFrame.head of              person_id  gender_concept_id  year_of_birth  month_of_birth  \
0    3589912774911670296               8507           2095             NaN
1   -3210373572193940939               8507           2079             NaN
2    -775517641933593374               8507           2149             NaN
3   -2575767131279873665               8507           2050             NaN
4   -8970844422700220177               8507           2114             NaN
..                   ...                ...            ...             ...
95  -7671795861352464589               8532           2052             NaN
96   5734523979606454056               8532           2069             NaN
97   1532249960797525190               8532           2106             NaN
98   5894416985828315484               8532           2055             NaN
99  -3780452582396805474               8532           2058             NaN

    day_of_birth  birth_datetime  race_concept_id  ethnicity_concept_id  \
0            NaN             NaN                0              38003563
1            NaN             NaN                0              38003563
2            NaN             NaN             8516                     0
3            NaN             NaN             8516                     0
4            NaN             NaN             8527                     0
..           ...             ...              ...                   ...
95           NaN             NaN       2000001401                     0
96           NaN             NaN       2000001401                     0
97           NaN             NaN       2000001405                     0
98           NaN             NaN       2000001405                     0
99           NaN             NaN       2000001405                     0

    location_id  provider_id  care_site_id  person_source_value  \
```

## 2.4 篩選資料

當我們讀取原始資料時，原始資料可能包含了許多我們不需要的資訊，又或者當我們想要從眾多的資料中，找出符合特定條件的某些觀察值時，便可以使用 Pandas 程式庫中資料框的篩選功能。

首先，我們可以篩選出 person 資料框中特定的欄位，例如我們只需要病患的 ID、出生年及性別三欄的資料，則可以使用下列程式碼。

```
#篩選資料框中的特定欄位

person_v2 = person[['person_id','year_of_birth','gender_source_value']]

person_v2

```

| | person_id | year_of_birth | gender_source_value |
|---|---|---|---|
| **0** | 3589912774911670296 | 2095 | M |
| **1** | -3210373572193940939 | 2079 | M |
| **2** | -775517641933593374 | 2149 | M |
| **3** | -2575767131279873665 | 2050 | M |
| **4** | -8970844422700220177 | 2114 | M |
| **...** | ... | ... | ... |
| **95** | -7671795861352464589 | 2052 | F |
| **96** | 5734523979606454056 | 2069 | F |
| **97** | 1532249960797525190 | 2106 | F |
| **98** | 5894416985828315484 | 2055 | F |
| **99** | -3780452582396805474 | 2058 | F |

100 rows × 3 columns

上列程式碼中，我們從資料框 person 取出三個欄位，儲存為一個新的變數，命名為 person_v2。我們可以使用 2.3 章提到的內容來檢視 person_v2 的資訊，可以發現篩選欄位後的 person_v2 同樣是一個資料框。

```
#person是一個data.frame
type(person_v2)
```

```
pandas.core.frame.DataFrame
```

```
#資料框的行名
person_v2.columns
```

```
Index(['person_id', 'year_of_birth', 'gender_source_value'], dtype='object')
```

```
#檢視資料形狀
person_v2.shape
```

```
(100, 3)
```

上述方法是在資料框中篩選欄位的方式，如果我們想要針對資料框的列做篩可以使用下列方法。例如我們想要在資料框 person_v2 中篩選出女性的病患，則可以使用下列程式碼：

```
#利用布林選擇選取列資料
person_v3 = person_v2[person_v2['gender_source_value']=='F']

#檢視資料形狀
person_v3.shape

```

```
(43, 3)
```

```
type(person_v3)
```

```
pandas.core.frame.DataFrame
```

結果顯示，篩選後的資料 person_v3 共有 43 列，因此可以知道 person_v2 中有 43 筆病患為女性。另外可以發現篩選後的資料 person_v3 依然是一個資料框的變數。

當我有多個篩選條件要篩選時，則可以用下列方法。例如我們想要在資料框 person_v2 中篩選出女性的病患，且出生年在 2100 年之後(出生年因為 MIMIC-IV data 釋出前會使用特定方式將資料去識別化，因此並非病患的實際出生年，詳細情形可以至前述 MIMIC-IV data 的網站了解。)，則可以使用下列程式碼：

```
#利用多個布林選擇選取列資料
person_v4 = person_v2[(person_v2['year_of_birth']>=2100)&(person_v2['gender_source_value']=='F')]

#檢視資料
person_v4
```

| | person_id | year_of_birth | gender_source_value |
|---|---|---|---|
| 57 | -2067961723109232727 | 2106 | F |
| 61 | 6128703162302148003 | 2120 | F |
| 64 | 3129727379702505063 | 2145 | F |
| 65 | -7636167699948083600 | 2105 | F |
| 68 | 2341788304019377091 | 2130 | F |
| 70 | 2288881942133868955 | 2102 | F |
| 74 | 421426604671948641 | 2119 | F |
| 76 | 2188642953583197091 | 2102 | F |
| 77 | -6022656226246460545 | 2104 | F |
| 80 | -8205283012979532608 | 2103 | F |
| 83 | 1258460361496302149 | 2108 | F |
| 85 | -6289874722419061830 | 2104 | F |
| 92 | 8527170356523164323 | 2128 | F |
| 93 | -8928428202649726867 | 2119 | F |
| 97 | 1532249960797525190 | 2106 | F |

## 第 3 部分 總結

本文以 MIMIC-IV Demo 資料為範例，示範從資料下載，到使用 python 程式碼讀取、檢視資料的步驟，並且介紹如何使用 Pandas 程式庫中資料框功能進行資料欄或列的篩選，找出分析需要用到的資料，希望能作為初次使用 python 進行資料分析讀者的指引。接下來的 eNews 將會介紹，當我們下載並篩選出所需的資料後，更進一步的資料清理、資料合併等方式，並且如何使用 Pandas 程式庫計算出我們所需要資料的描述性統計等基本資訊。