

Physionet 資料分析

Arousal 覺醒

- 在睡眠領域中，「arousal」（覺醒）是指睡眠期間發生的一種短暫的覺醒狀態或者睡眠中的激動反應。這是指從深度睡眠（非REM睡眠）或者快速動眼期（REM睡眠）中短暫的恢復到較高的覺醒程度。
- arousal 可以是由內外部的刺激所引起，如聲音、光線、夢境、不適感等。當睡眠中的人經歷這些刺激時，他們可能會在一段時間內從深睡眠狀態中醒來，或者從REM睡眠中達到較高程度的覺醒。
- 睡眠中的短暫覺醒是正常且常見的現象，大部分人可能並不會完全意識到自己的覺醒經歷，並能在不久之後重新進入睡眠。然而，一些人可能會更易受到外界刺激的影響，並在夜間多次醒來，這可能會影響他們的睡眠質量和持續時間。
- 睡眠中的覺醒也與一些睡眠障礙相關，如失眠症或睡眠呼吸暫停症。在這些疾病中，人們可能會經歷頻繁的覺醒，導致睡眠品質下降並對健康產生負面影響。
- 了解和評估睡眠中的覺醒是睡眠醫學和研究的重要方面之一。通過使用腦電圖（EEG）和其他生理測量，睡眠專家可以追蹤和記錄睡眠中的覺醒事件(arousal events)，並進一步研究其對睡眠和健康的影響。
- 睡眠中的覺醒是指從深度睡眠或REM睡眠中短暫的恢復到較高程度的覺醒狀態。它可以由內外部刺激引起，是睡眠中的正常現象，但在某些睡眠障礙中可能會變得更加頻繁和困擾。睡眠專家使用不同的測量方法來研究和評估睡眠中的覺醒，以進一步了解其對睡眠和健康的影響。

關於這個資料集

- 競賽網址：<https://physionet.org/content/challenge-2018/1.0.0/>
資料集：<https://reurl.cc/eDz88R>
- Challenge 數據存儲庫包含兩個目錄(training和test)，每個目錄的大小約為 135 GB。每個子目錄為受試者訊號資料(例如training/tr03-0005)，包含訊號資料、標頭檔和標註資料；例如：
 - tr03-0005.mat：包含信號資料的 Matlab V4 文件
 - tr03-0005.hef：記錄標頭檔 - 描述信號數據格式的文本文件
 - tr03-0005.arousal：喚醒和睡眠階段註釋，使用WFDB 註釋格式
 - tr03-0005-arousal.mat：一個 Matlab V7 結構，包含一個具有三個不同值 (+1、0、-1) 的樣本向量，其中：
 - +1：指定喚醒區域
 - 0：指定非喚醒區域
 - -1：指定不計分的區域



讀取信號資料與標註資料

```
# import the WFDB package
```

```
import wfdb, mat73
```

```
# load a record using the 'rdrecord' function
```

```
record = wfdb.rdrecord('tr03-0005')
```

```
arousal = wfdb.rdann('tr03-0005', 'arousal')
```

```
aasmlabel = mat73.loadmat('tr03-0005-arousal.mat')
```

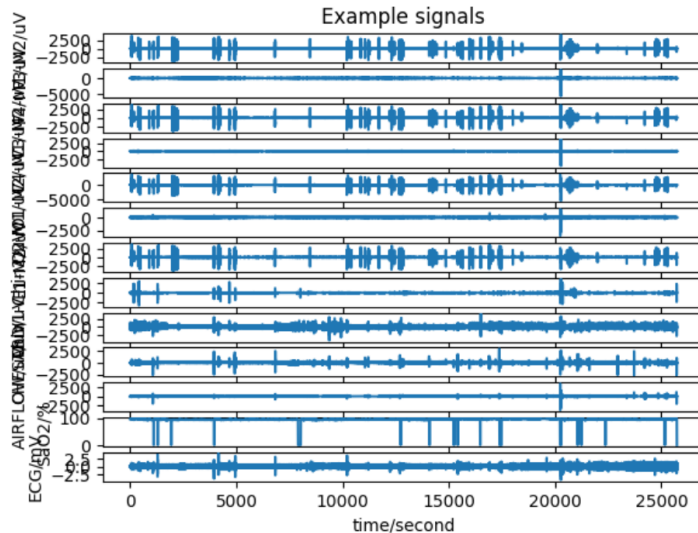
信號資料

Arousal 標註資料，包含睡眠分期，與呼吸中止標註

AASM 標註資料，主要是依據AASM的睡眠分期標註

```
# plot the record to screen
```

```
wfdb.plot_wfdb(record=record, title='Example signals')
```



因為我們現在的分析重點是：

1. 用心率、BBI、心律變異度分析「睡、醒」或「分期」

2. 分析Spindles或K複核波

所以主要讀取信號資料與AASM標註資料即可，忽略Arousal 標註資料

tr03-0005-arousal.mat 資料讀取測試

```
import mat73, os
physionet_folder = 'D:\\challenge-2018-1.0.0.physionet.org\\training'
record_name = 'tr03-0005'
data_dict = mat73.loadmat(os.path.join(physionet_folder, record_name, f'{record_name}-arousal.mat'))
```

↓

```
print(data_dict)
```

↓

```
{'data': {'arousals': array([ 0.,  0.,  0., ..., -1., -1., -1.]), 'sleep_stages': {'nonrem1':
array([False, False, False, ..., False, False, False]), 'nonrem2': array([False, False, False, ..., False,
False, False]), 'nonrem3': array([False, False, False, ..., False, False, False]), 'rem': array([False,
False, False, ..., False, False, False]), 'undefined': array([ True,  True,  True, ..., False, False,
 True]), 'wake': array([False, False, False, ...,  True,  True, False])}}}
```

↓

```
data_dict.get('data').get('arousals').__len__()
```

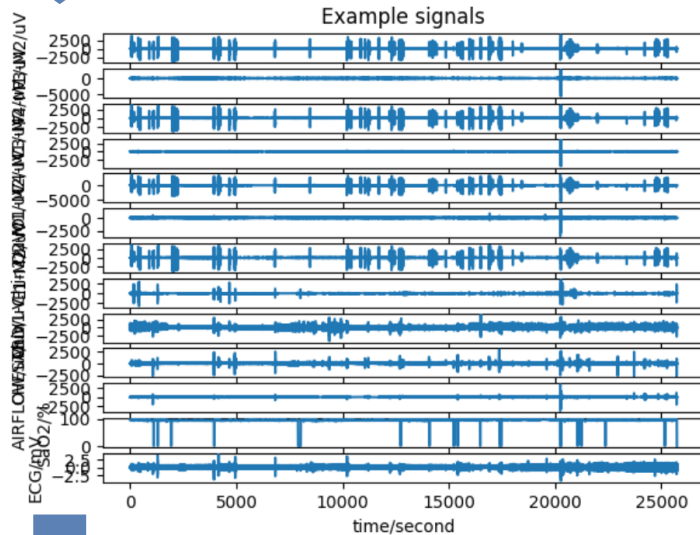
↓

```
5147000
```

tr03-0005.mat 與 tr03-0005.hef 資料讀取測試

```
import wfdb, os
physionet_folder = 'D:\\challenge-2018-1.0.0.physionet.org\\training'
record_name = 'tr03-0005'
record = wfdb.rdrecord(os.path.join(physionet_folder, record_name, record_name))
wfdb.plot_wfdb(record=record, title='Example signals')
```

指定檔名「tr03-0005」，同時讀取 tr03-0005.mat 與 tr03-0005.hef



record.__dict__

取得資料與量測背景

通道數量

取樣率

總資料量

通道名稱

增益

單位

位元深度

```
{'record_name': 'tr03-0005',
 'n_sig': 13,
 'fs': 200,
 'counter_freq': None,
 'base_counter': None,
 'sig_len': 5147000,
 'base_time': None,
 'base_date': None,
 'comments': [],
 'sig_name': ['F3-M2', 'F4-M1', 'C3-M2', 'C4-M1', 'O1-M2', 'O2-M1', 'E1-M2', 'Chin1-Chin2', 'ABD', 'CHEST', 'AIRFLOW', 'SaO2', 'ECG'],
 'p_signal': array([[...]]),
 'd_signal': None,
 'e_p_signal': None,
 'e_d_signal': None,
 'file_name': ['tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat', 'tr03-0005.mat'],
 'fmt': ['16', '16', '16', '16', '16', '16', '16', '16', '16', '16', '16', '16', '16'],
 'samps_per_frame': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
 'skew': [None, None, None, None, None, None, None, None, None, None, None, None, None],
 'byte_offset': [24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24],
 'adc_gain': [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 655.35, 1000.0],
 'baseline': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -32768, 0],
 'units': ['uV', 'uV', 'uV', 'uV', 'uV', 'uV', 'uV', 'uV', 'uV', 'uV', 'uV', '%', 'mV'],
 'adc_res': [16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16],
 'adc_zero': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 'init_value': [-9, 5, -5, 9, 5, 12, -21, 16, -41, -2, 39, 30496, -53],
 'checksum': [139, 1793, 1290, -357, 248, -96, -1068, -741, -31565, -21968, 2879, 9703, -1118],
 'block_size': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]}
```

tr03-0005.arousal 資料讀取測試

```
import wfdb, os
physionet_folder = 'D:\\challenge-2018-1.0.0.physionet.org\\training'
record_name = 'tr03-0005'
arousal = wfdb.rdann(os.path.join(physionet_folder, record_name, record_name), 'arousal')
```

arousal.__dict__

整個睡眠時段的所有標註段落

```
{'record_name': 'tr03-0005',
 'extension': 'arousal',
 'sample': array([ ... ], dtype=int64),
 'symbol': [ ... ],
 'subtype': array([ ... ]),
 'chan': array([ ... ]),
 'num': array([ ... ]),
 'aux_note': [ ... ],
 'fs': 200,
 'label_store': None,
 'description': None,
 'custom_labels': None,
 'contained_labels': None,
 'ann_len': 240}
```

```
1 arousal.__dict__.get('sample').__len__()
```

240

```
1 arousal.__dict__.get('aux_note').__len__()
```

240

```
1 set(arousal.__dict__.get('aux_note'))
```

```
{('arousal_rera',
 '(resp_centralapnea',
 '(resp_hypopnea',
 '(resp_obstructiveapnea',
 'N1',
 'N2',
 'N3',
 'R',
 'W',
 'arousal_rera)',
 'resp_centralapnea)',
 'resp_hypopnea)',
 'resp_obstructiveapnea)')}
```

Arousal 標註 (共9項標註)

■ apnea : 呼吸中止

- resp_centralapnea : 呼吸暫停

- resp_hypopnea : 呼吸不足

- resp_obstructiveapnea : 呼吸阻塞性呼吸暫停

■ arousal : 短暫覺醒

■ Sleep = n1+n2+n3+rem

■ wake

因為我們現在的分析重點是：

1. 用心率、BBI、心律變異度分析「睡、醒」或「分期」
2. 分析Spindles或K複核波

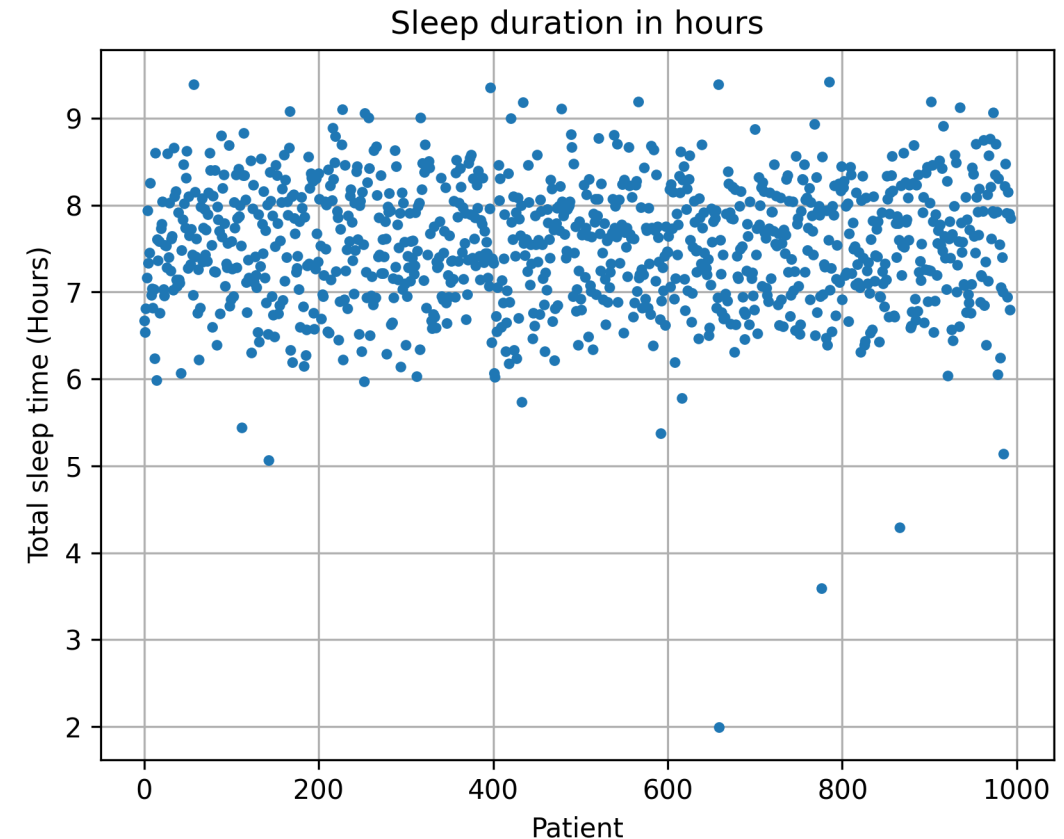
所以主要讀取信號資料與AASM標註資料即可，忽略Arousal 標註資料

Sleep duration in hours (with arousal labeling)

■ Data source: 994 training data

■ 平均睡眠階段時長與所占百分比：

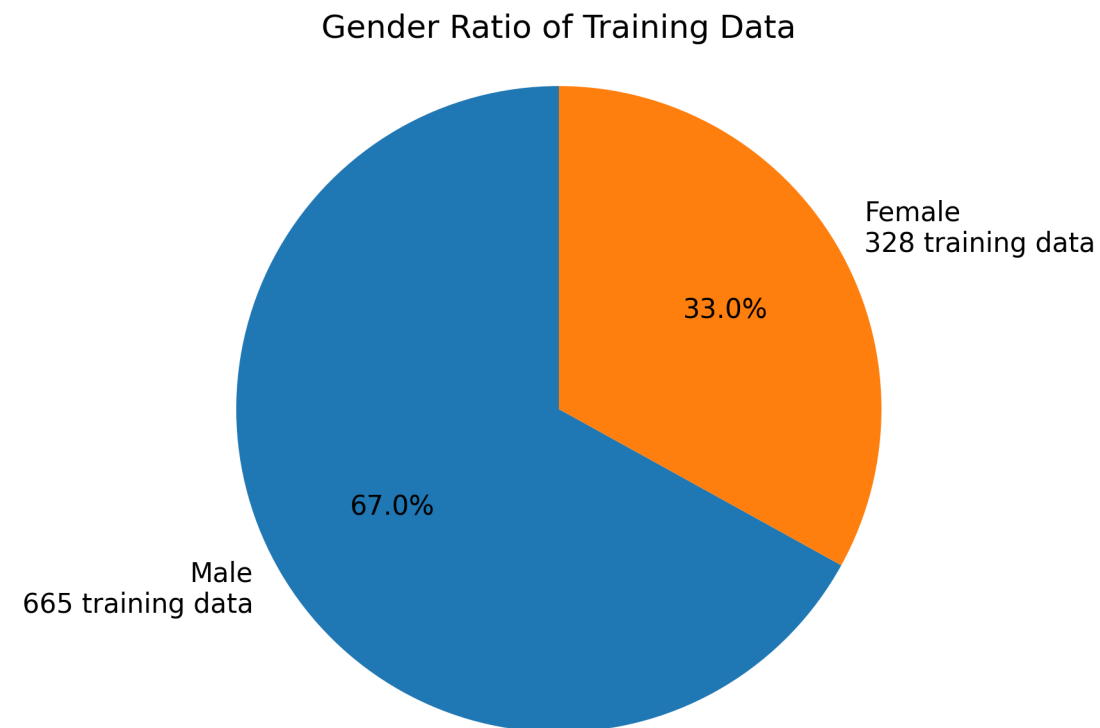
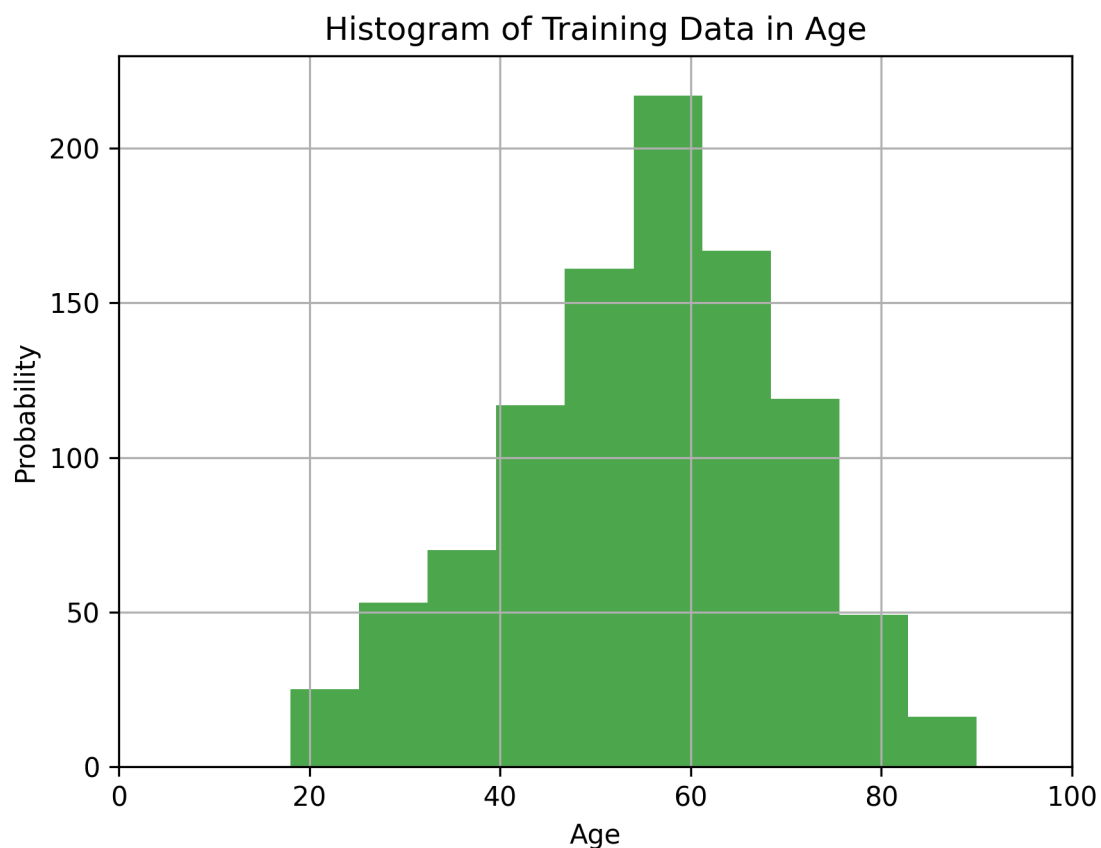
- N1: 1.67 hours, 22%
- N2: 0.87 hours, 12%
- N3: 0.71 hours, 9%
- R: 0.25 hours, 3%
- W: 0.85 hours, 11%
- arousal(喚醒): 1.12 hours, 15%
- hypopnea(呼吸不足): 1.12 hours, 15%
- centralapnea(呼吸暫停): 0.43 hours, 6%
- obstructiveapnea(阻塞性呼吸暫停): 0.48 hours, 6%
- 睡眠總時長: 7.54 hours



Sleep duration in hours (with AASM labeling)

- AASM標註，分成：N1、N2、N3、REM、Wake、Undefined六類
- 睡眠時長 = $N1 + N2 + N3 + REM$
- Data source: 994 training data，AASM標註結果，分析如下：
 - Sleep duration: 6.16 ± 1.16 hours
 - N1: 1.15 ± 0.76 hours (18.66%)
 - N2: 3.17 ± 1.03 hours (51.46%)
 - N3: 0.86 ± 0.62 hours (13.97%)
 - R: 0.98 ± 0.59 hours (18.66%)
 - W: 1.33 ± 0.99 hours
 - undefined: 0.23 ± 0.31 hours

Training Data年齡性別統計

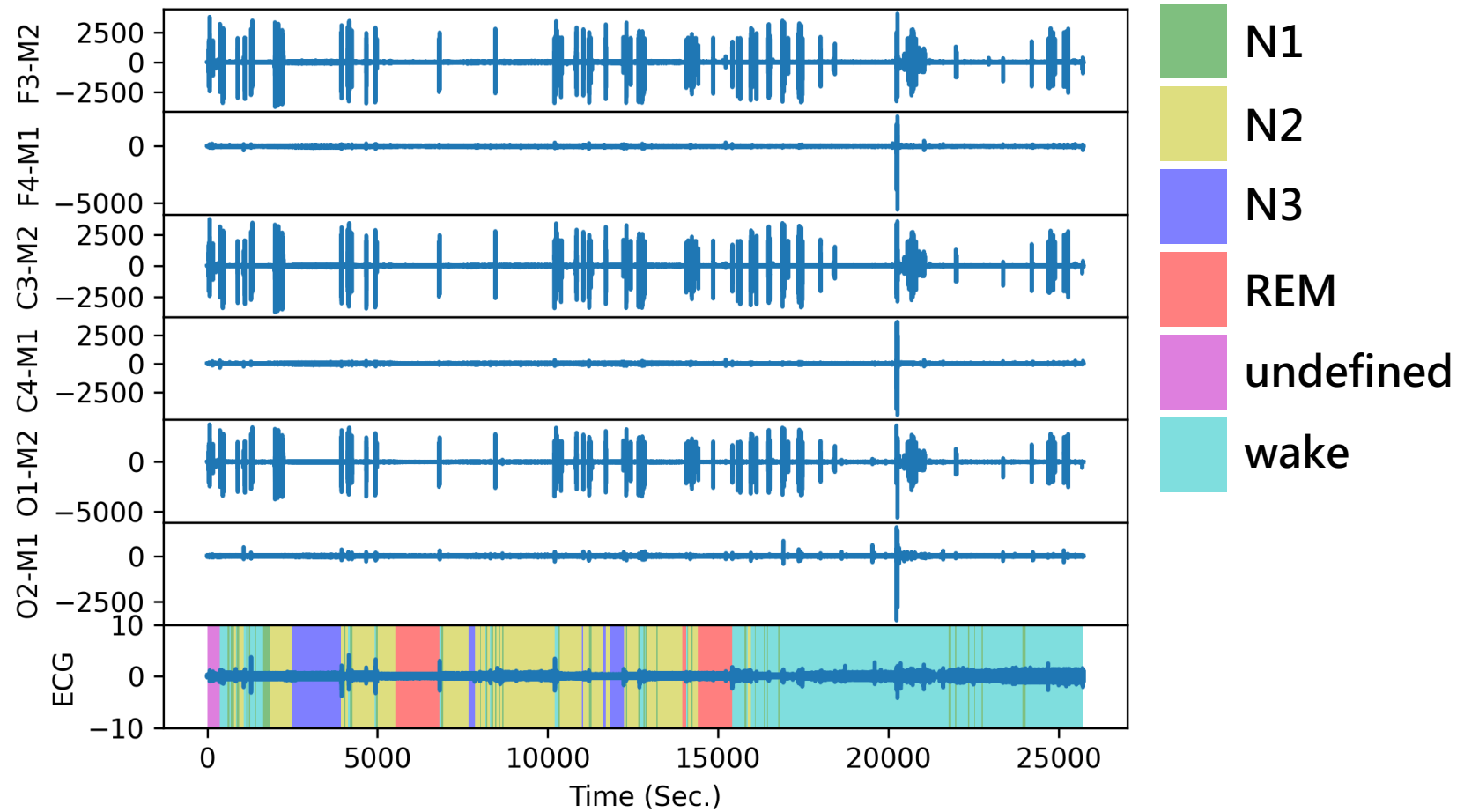
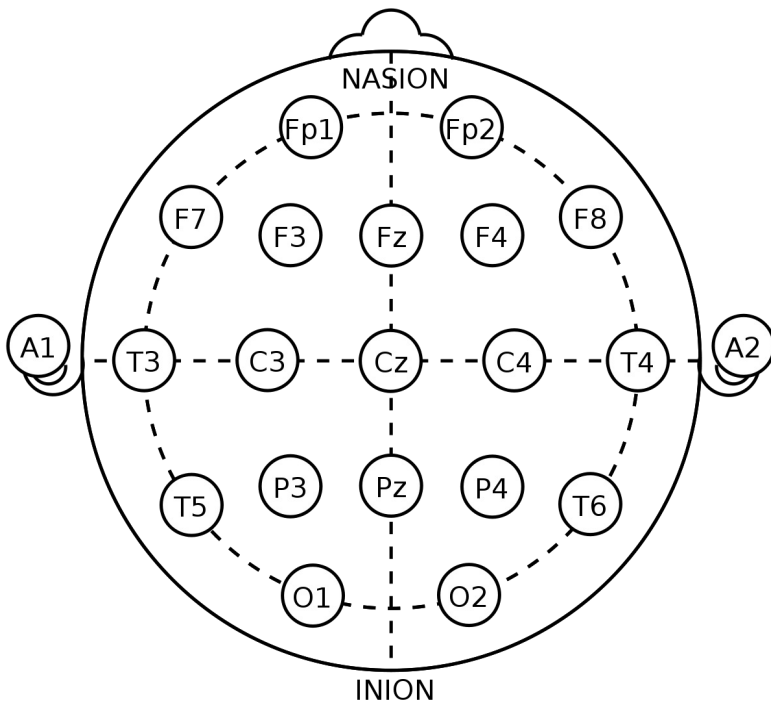


腦波波段

■ δ 波 (0.5~3 Hz)	N3, SWS
■ θ 波 (4~7 Hz)	N2
■ α 波 (8~12 Hz)	N1
■ β 波 (12~30 Hz)	Wake
■ σ 波 (10-15 Hz)	Spindles

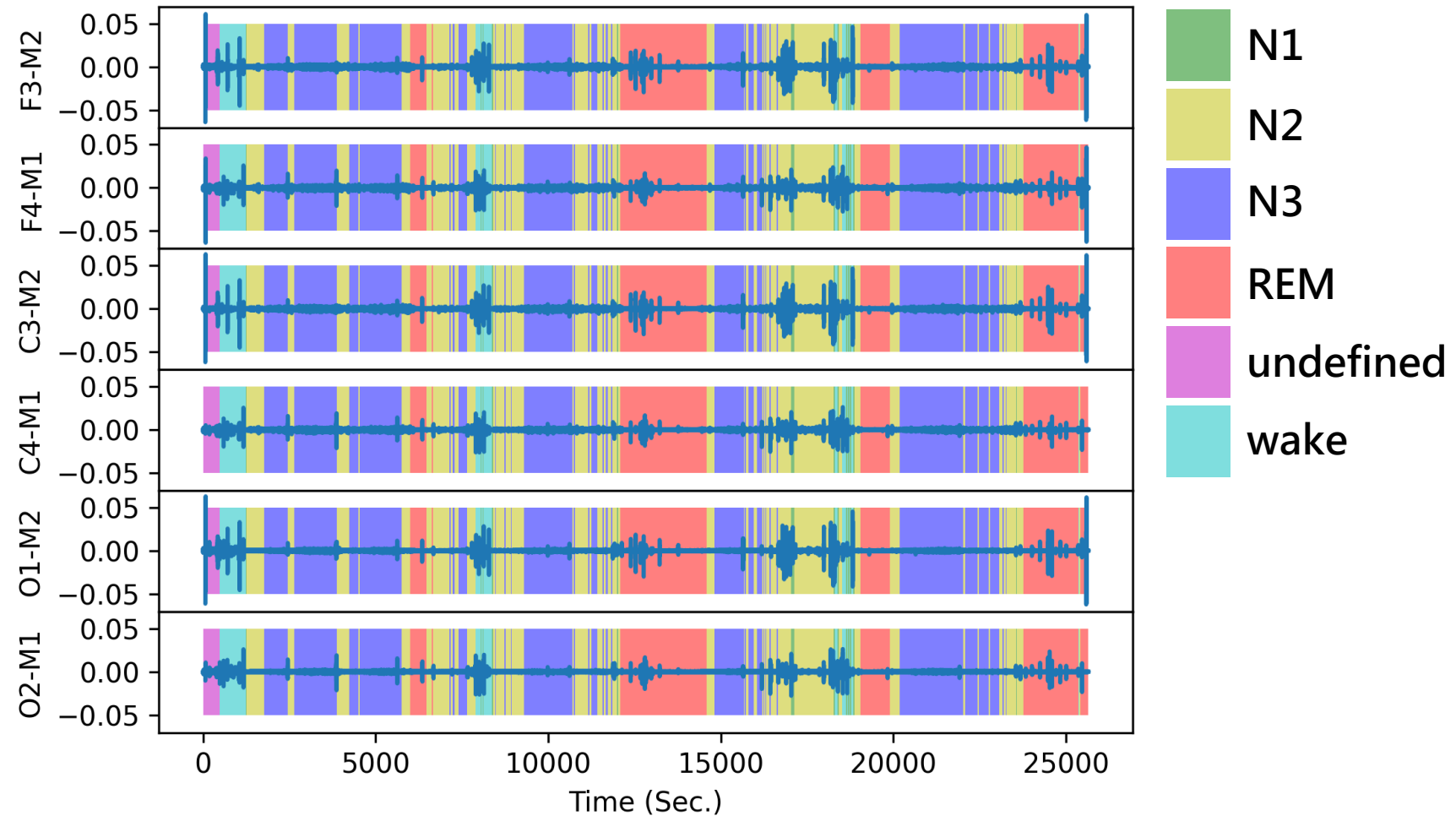
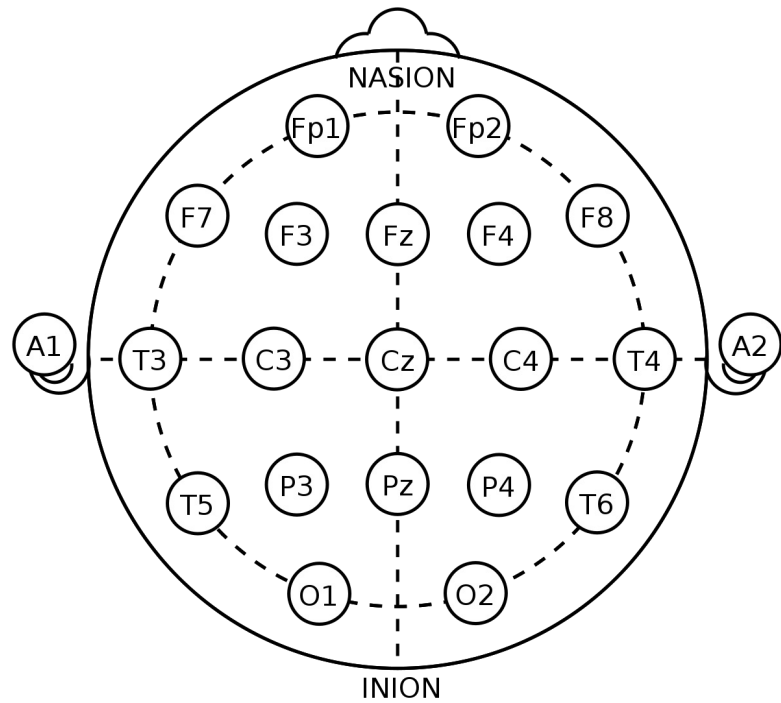
Training data: tr03-0005 (Male, Age 59)

tr03-0005 (M, Age59) EEG Data



Training data: tr03-1333 (Female, Age 21)

tr03-1333 (Fmale, Age21) EEG Data



Case 1

取出某些channels的資料

取出ECG資料

將ECG資料進行後處理

情境說明

- 為了分析睡眠的情況，或為了訓練模型，需要取出某些channels的資料
- 所有channels：F3-M2, F4-M1, C3-M2, C4-M1, O1-M2, O2-M1, E1-M2, Chin1-Chin2, ABD, CHEST, AIRFLOW, SaO2, ECG
- Sampling rate：200 S/sec.

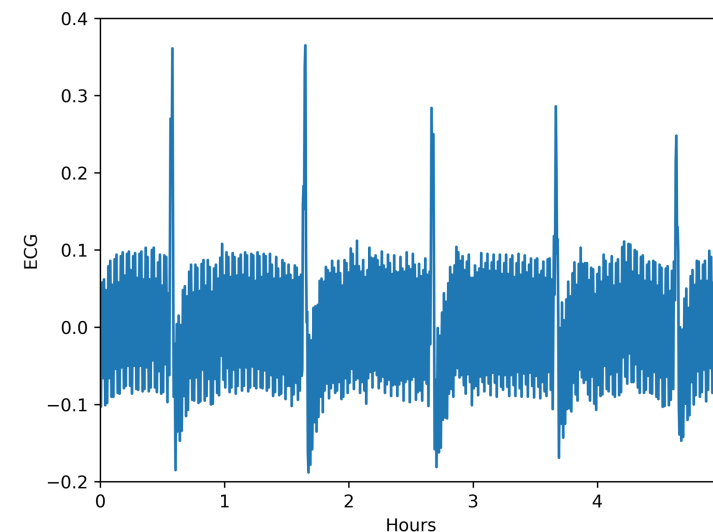
Example

■ 取出前6個channels的data

```
import wfdb, os
import numpy as np
physionet_folder = 'D:\\challenge-2018-1.0.0.physionet.org\\training'
record_name = 'tr03-0005'
record = wfdb.rdrecord(os.path.join(physionet_folder, record_name, record_name))
record_6_channels = record.__dict__.get('p_signal')[:,0:6]
```

■ 取出ECG channel的data

```
import matplotlib.pyplot as plt
x = np.arange(record.__dict__.get('sig_len'))/record.__dict__.get('fs')
y = record_ecg
plt.plot(x, y)
plt.xlim(0, 5)
plt.ylim(-0.5, 0.5)
plt.xlabel('Hours')
plt.ylabel('ECG')
plt.title('ECG')
plt.show()
```



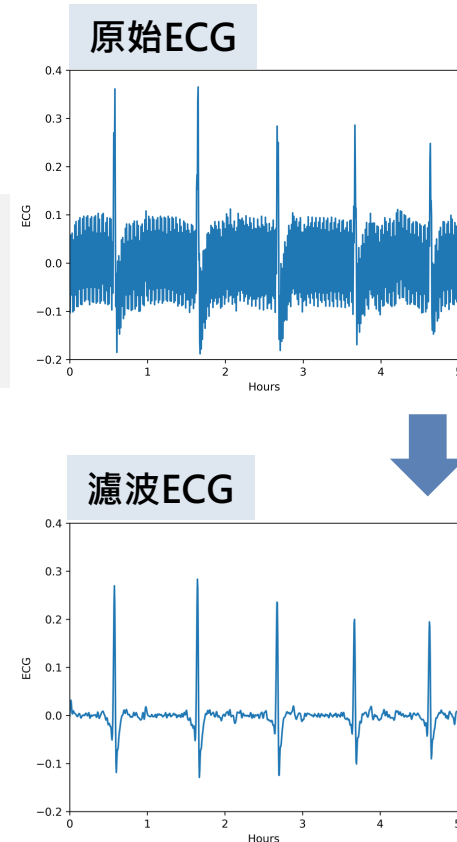
ECG訊號後處理 (post processing) -1

- 原始的ECG訊號包含高頻雜訊，只能以肉眼辨識PQRST的P peak，無法看到其他特徵。另外，如果要分析BBI，須找出所有P-peak，因此需要最原本的ECG訊號做適當後處理

- 可以使用套件「biosppy」做ECG信號的後處理 `pip install biosppy`

```
import biosppy
ecg_biosppy_out = biosppy.signals.ecg.ecg(signal=record_ecg, show=False,
                                          sampling_rate=record.__dict__.get('fs'))
```

```
import matplotlib.pyplot as plt
x = np.arange(record.__dict__.get('sig_len'))/record.__dict__.get('fs')
y = ecg_biosppy_out.__dict__.get('filtered')
plt.plot(x, y)
plt.xlim(0, 5)
plt.ylim(-0.2, 0.4)
plt.show()
```



ECG訊號後處理 (post processing) -2

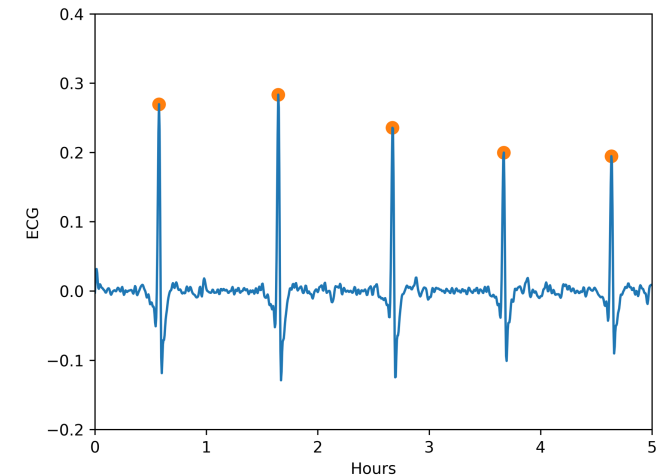
- 另外，經過biosppy套件處理後的ECG資料還包含P peak的資訊，全資料如下：

```
ecg_biosppy_out.__dict__.keys()
```

```
odict_keys(['ts', 'filtered', 'rpeaks', 'templates_ts', 'templates', 'heart_rate_ts', 'heart_rate'])
```

- 將濾波完的ECG與P peak疊在一起：

```
import matplotlib.pyplot as plt
x = np.arange(record.__dict__.get('sig_len'))/record.__dict__.get('fs')
y = ecg_biosppy_out.__dict__.get('filtered')
plt.plot(x, y)
plt.scatter(ecg_biosppy_out.__dict__.get('ts')[ecg_biosppy_out.__dict__.get('rpeaks')],
            ecg_biosppy_out.__dict__.get('filtered')[ecg_biosppy_out.__dict__.get('rpeaks')],
            color='tab:orange')
plt.xlim(0, 5)
plt.ylim(-0.2, 0.4)
plt.xlabel('Hours')
plt.ylabel('ECG')
plt.show()
```

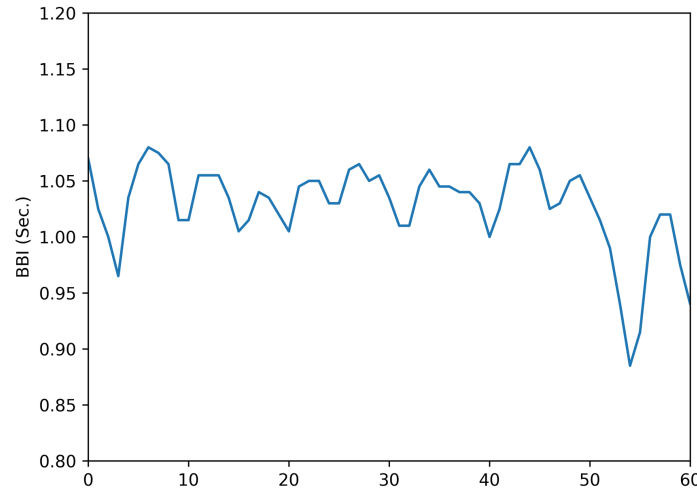


ECG訊號後處理 (post processing) -3

■取得BBI資料

```
ts_rpeaks = ecg_biosppy_out.__dict__.get('ts')[ecg_biosppy_out.__dict__.get('rpeaks')]  
bbi = (ts_rpeaks-np.roll(ts_rpeaks, 1))[1:]
```

```
import matplotlib.pyplot as plt  
x = np.arange(bbi.__len__())  
y = bbi  
plt.plot(x, y)  
plt.xlim(0, 60)  
plt.ylim(0.8, 1.2)  
plt.ylabel('BBI (Sec.)')  
plt.show()
```



■之後使用PyHRV模組分析心律變異度，再與睡眠分期資料訓練ML模型