2. 헬스케어 데이터

Part3. 헬스케어 데이터의 종류 (Cont'd)
- 실습

sydong@sookmyung.ac.kr

실습 – MNE EEG Analysis

https://mne.tools/stable/index.html

- MNE-Python
  - ✓ Open-source Python package for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, NIRS, and more
- EEG analysis - Event-Related Potentials (ERPs)
  - ✓ Perform standard ERP analyses in MNE-Python using example data
    - sample_audvis_filt-0-40_raw.fif
  - ✓ Also, we will load the events from an external events file
    - sample_audvis_filt-0-40_raw-eve.fif
- You can find this tutorial in the following link
  - ✓ https://mne.tools/stable/auto_tutorials/evoked/30_eeg_erp.html?highlight=sample_audvis_filt+40_raw+fif

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import mne

root = mne.datasets.sample.data_path() / 'MEG' / 'sample'
raw_file = root / 'sample_audvis_filt-0-40_raw.fif'
raw = mne.io.read_raw_fif(raw_file, preload=False, verbose=False)
raw.pick(['eeg', 'eog']).load_data()

events_file = root / 'sample_audvis_filt-0-40_raw-eve.fif'
events = mne.read_events(events_file)

raw.crop(tmax=90)  # in seconds (happens in-place)
events = events[events[:, 0] <= raw.last_samp]
```

```
Removing projector <Projection | PCA-v1, active : False, n_channels : 102>
Removing projector <Projection | PCA-v2, active : False, n_channels : 102>
Removing projector <Projection | PCA-v3, active : False, n_channels : 102>
Reading 0 ... 41699  =      0.000 ...   277.709 secs...
```

실습 – MNE EEG Analysis

- Check data
  - ✓ Since this is a combined EEG/MEG dataset, we restricted the data to just the EEG and EOG channels
  - ✓ By looking at the measurement info we can see that we now have 59 EEG channels and 1 EOG channel
- Channel names
  - ✓ In our data set, all channel types are already correct
  - ✓ Therefore, we'll only remove a space and a leading zero in the channel names and convert to lowercase:
    - EEG 001-> eeg 01

```
print(raw.info)
```

```
<Info | 15 non-empty values
 bads: 1 items (EEG 053)
 ch_names: EEG 001, EEG 002, EEG 003, EEG 004, EEG 005, EEG 006, EEG 007, ...
 chs: 60 EEG, 1 EOG
 custom_ref_applied: False
 dev_head_t: MEG device -> head transform
 dig: 146 items (3 Cardinal, 4 HPI, 61 EEG, 78 Extra)
 file_id: 4 items (dict)
 highpass: 0.1 Hz
 hpi_meas: 1 item (list)
 hpi_results: 1 item (list)
 lowpass: 40.0 Hz
 meas_date: 2002-12-03 19:01:10 UTC
 meas_id: 4 items (dict)
 nchan: 61
 projs: Average EEG reference: off
 sfreq: 150.2 Hz
>
```

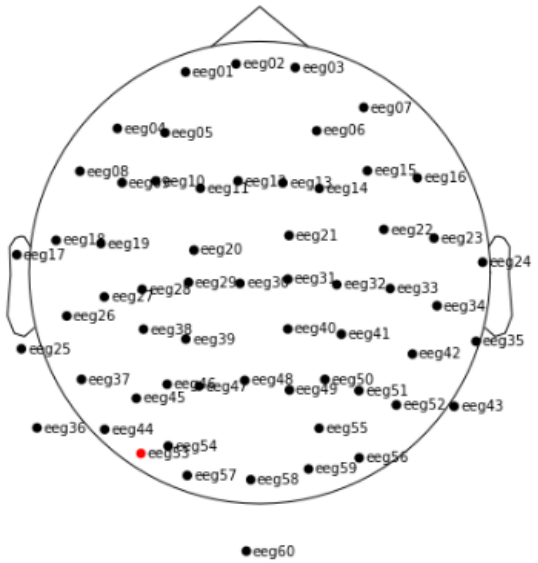
```
channel_renaming_dict = {name: name.replace(' 0', '').lower()
                         for name in raw.ch_names}
_ = raw.rename_channels(channel_renaming_dict)  # happens in-place
```

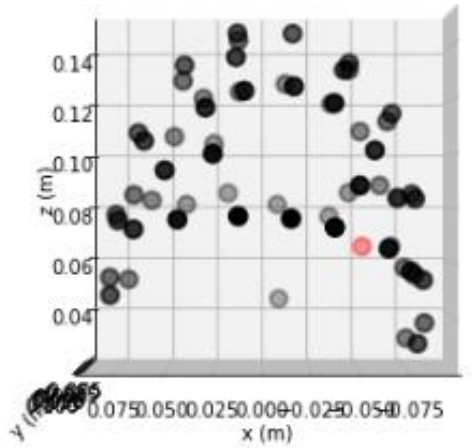
```
raw.plot_sensors(show_names=True)
fig = raw.plot_sensors('3d')
```

실습 – MNE EEG Analysis

- Visualization
  - ✓ plot()
  - ✓ plot_sensors()
    - we can view the sensor locations in 2D or 3D

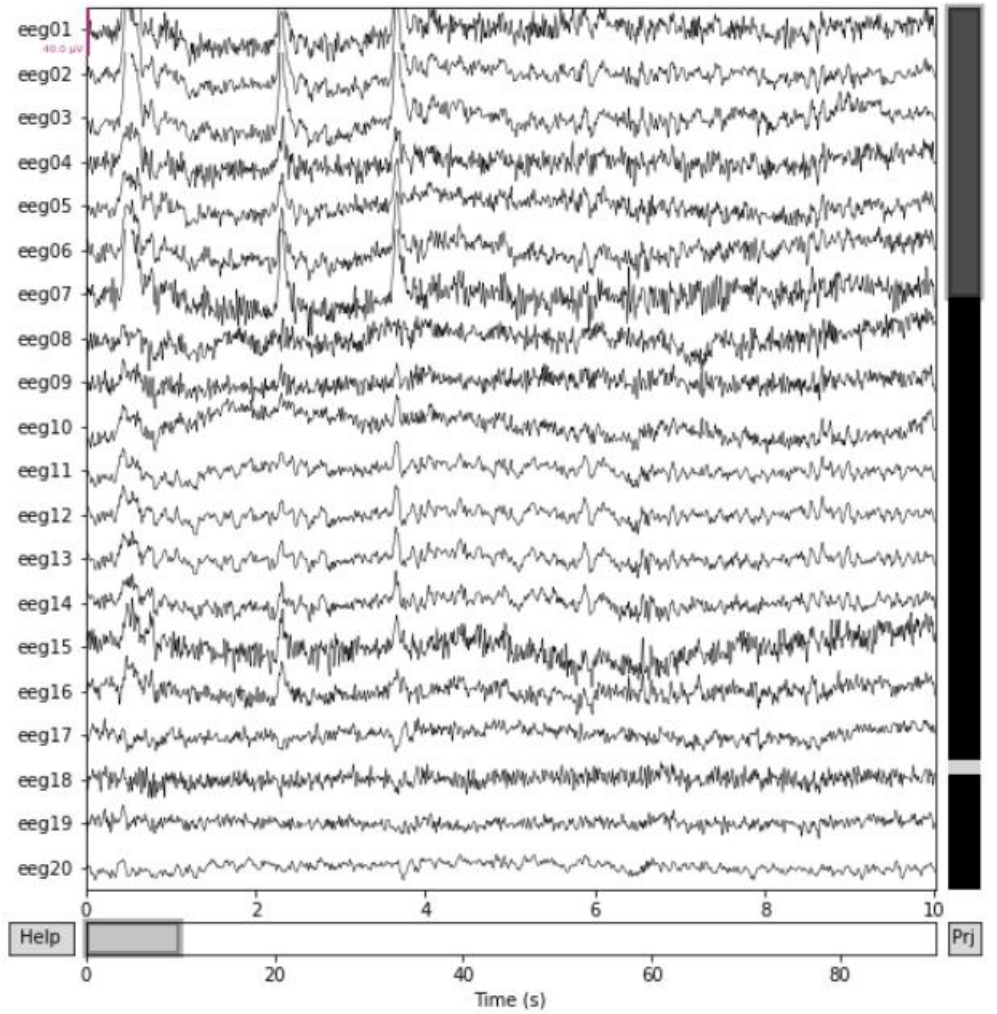
```
raw.plot_sensors(show_names=True)
fig = raw.plot_sensors('3d')
```






```
raw.plot()
```

```
/usr/local/lib/python3.7/dist-packages/matplotlib/colors.py:263: VisibleDeprecati
  c = np.array(c)
```



실습 – MNE EEG Analysis

- Epoching
  - ✓ In MNE-Python, events are represented as a NumPy array containing event latencies (in samples) and integer event codes. The event codes are stored in the last column of the events array:
  - ✓ The Sample data uses the following mapping:
  - ✓ Now we can proceed to epoch the continuous data
  - ✓ An interactive plot allows us to click on epochs to mark them as “bad” and drop them from the analysis
    - only when you run epochs.plot() in a Python console

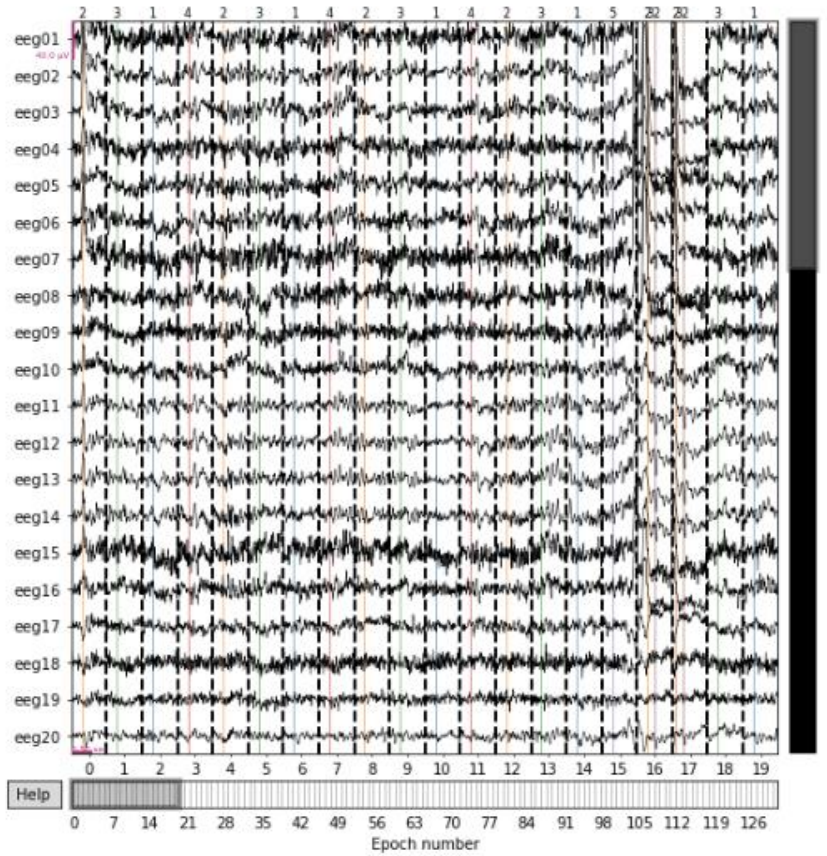
```
np.unique(events[:, -1])
```

```
array([ 1,  2,  3,  4,  5, 32])
```

```
event_dict = {'auditory/left': 1, 'auditory/right': 2, 'visual/left': 3,
              'visual/right': 4, 'face': 5, 'buttonpress': 32}
```

```
epochs = mne.Epochs(raw, events, event_id=event_dict, tmin=-0.3, tmax=0.7,
                    preload=True, verbose=False)
fig = epochs.plot(events=events)
```

```
You seem to have overlapping epochs. Some event lines may be duplicated in the plot.
/usr/local/lib/python3.7/dist-packages/matplotlib/colors.py:263: VisibleDeprecationWarn
  c = np.array(c)
```



Inclass Assignment

- MNE EEG analysis - Event-Related Potentials (ERPs) 를 따라해보고 마지막 시각화 결과를 캡처하여 수업시간 내에 스노우보드에 올립니다
- 사진 파일 하나로 제출합니다

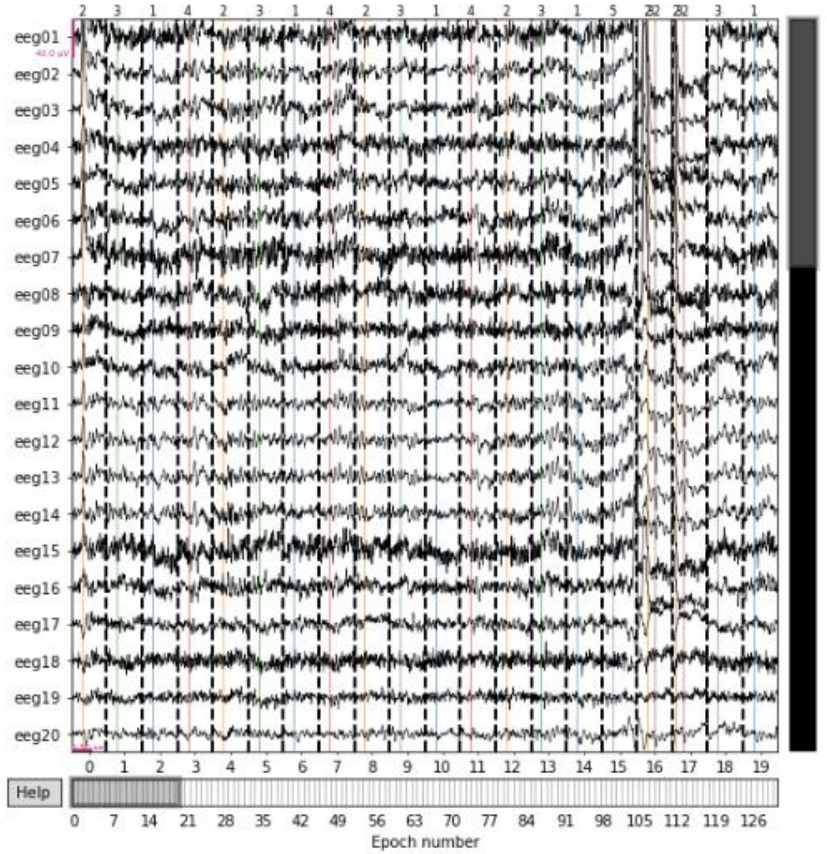
```
np.unique(events[:, -1])
```

```
array([ 1,  2,  3,  4,  5, 32])
```

```
event_dict = {'auditory/left': 1, 'auditory/right': 2, 'visual/left': 3,
              'visual/right': 4, 'face': 5, 'buttonpress': 32}
```

```
epochs = mne.Epochs(raw, events, event_id=event_dict, tmin=-0.3, tmax=0.7,
                    preload=True, verbose=False)
fig = epochs.plot(events=events)
```

```
You seem to have overlapping epochs. Some event lines may be duplicated in the plot.
/usr/local/lib/python3.7/dist-packages/matplotlib/colors.py:263: VisibleDeprecationWarn
  c = np.array(c)
```

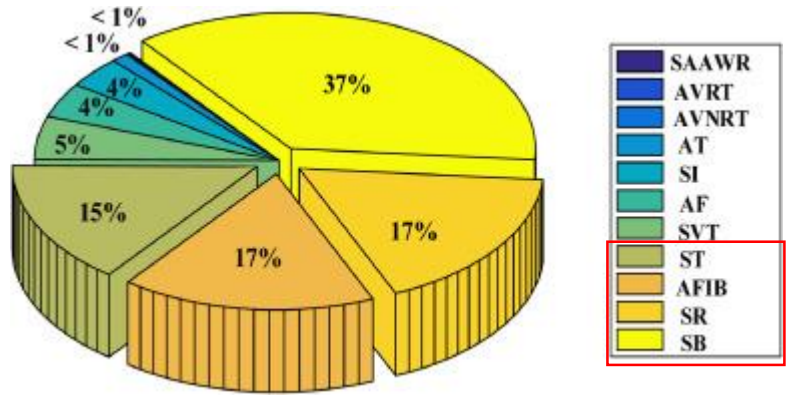


생체 데이터의 종류

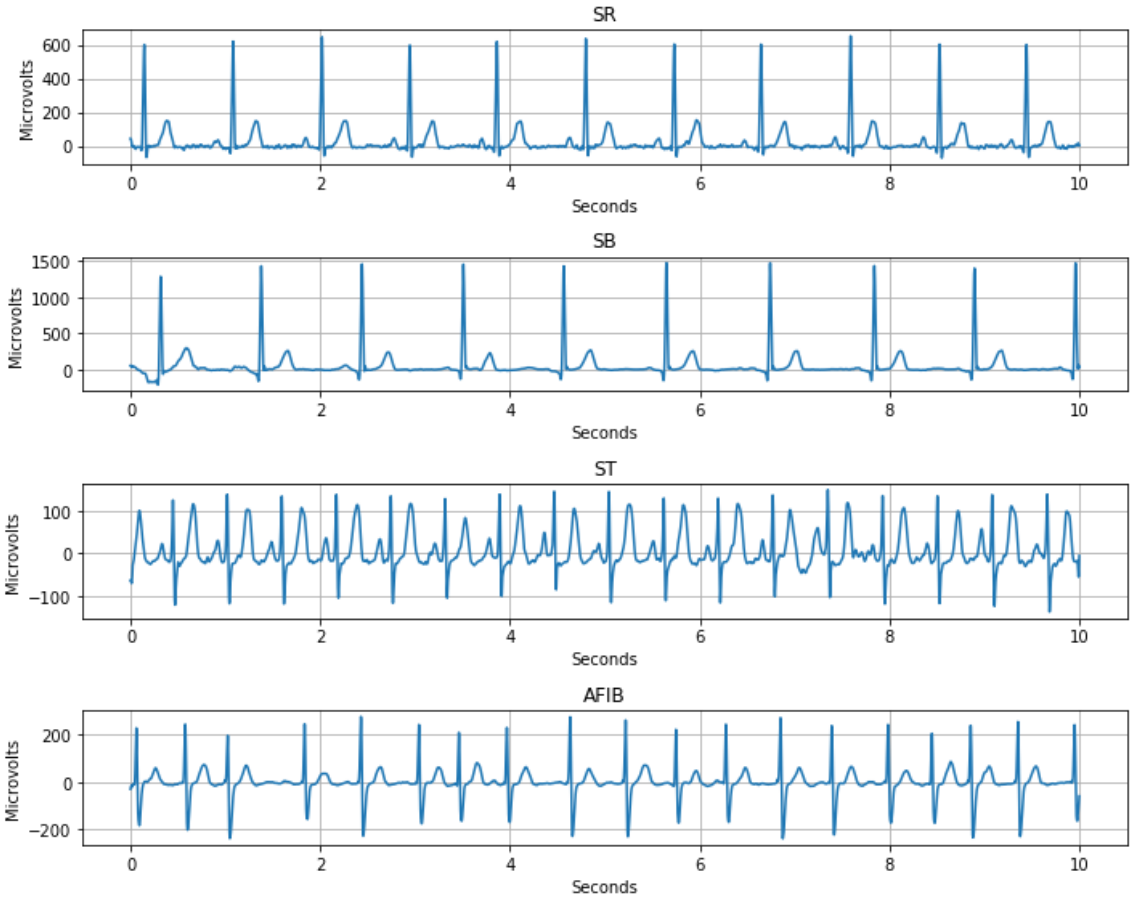
ECG

실습

- *가장 대표적인 4가지 Type에 대한 샘플 데이터를 직접 확인해보자



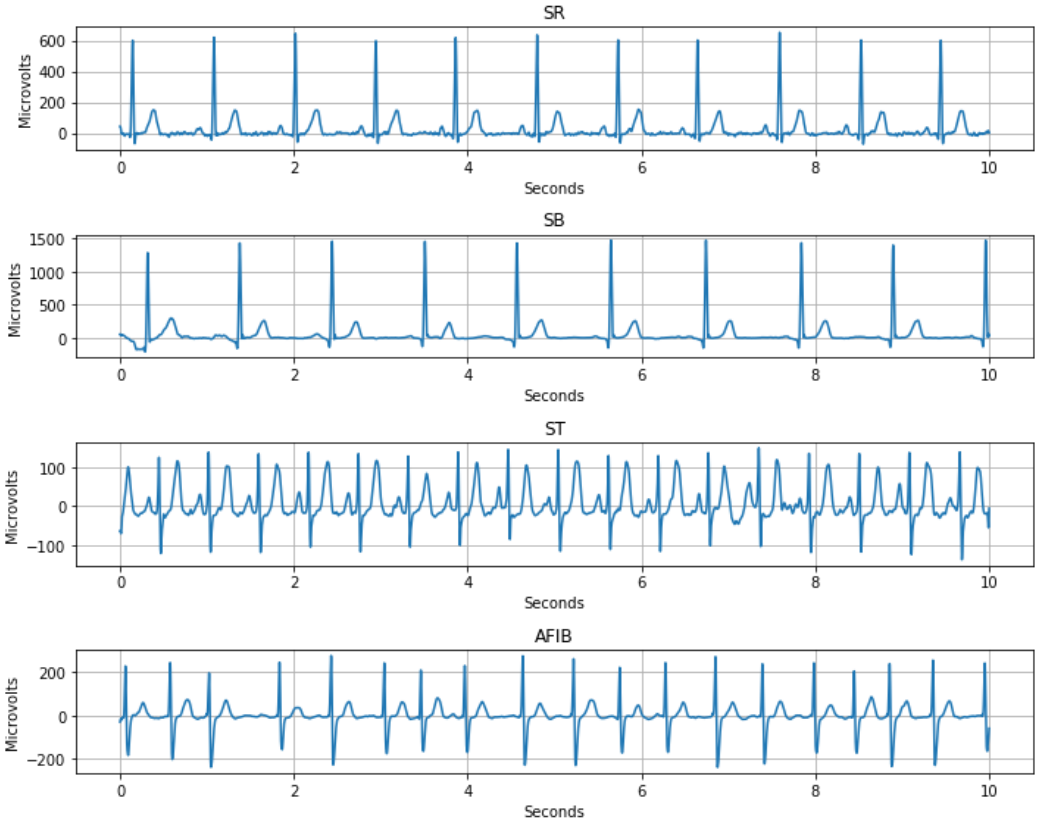
- File name (QRSCount) – All denoised
  - ✓ Sample_ST.csv (17)
  - ✓ Sample_AFIB.csv (18)
  - ✓ Sample_SR.csv (11)
  - ✓ Sample_SB.csv (9)
- Column 이름은 아래 리스트 참고
  - ✓ `col_names = ['I','II','III','aVR','aVL','aVF','V1','V2','V3','V4','V5','V6']`
  - ✓ 첫번째 column만 시각화합니다
- 샘플링레이트는 500Hz



```python
# initial imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

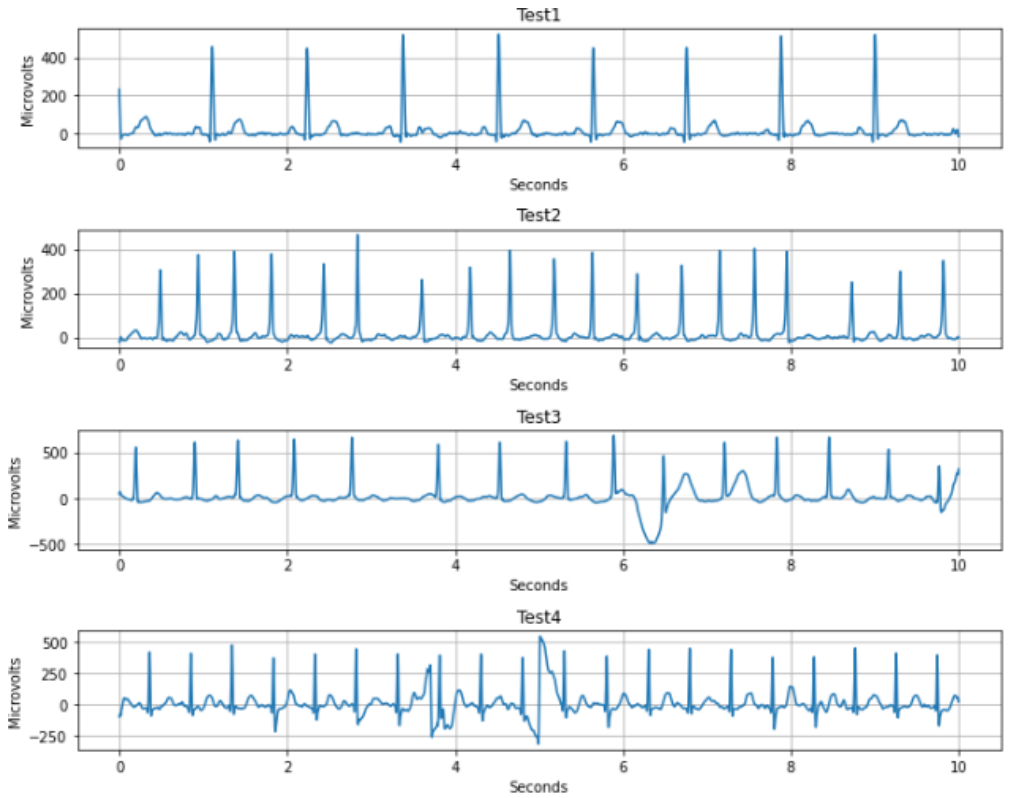
```python
# set column names
col_names = ['I','II','III','aVR','aVL','aVF','V1','V2','V3','V4','V5','V6']
# load data files
df1 = pd.read_csv("/content/drive/MyDrive/data/Sample_SR.csv", names = col_names)
df2 = pd.read_csv("/content/drive/MyDrive/data/Sample_SB.csv", names = col_names)
df3 = pd.read_csv("/content/drive/MyDrive/data/Sample_ST.csv", names = col_names)
df4 = pd.read_csv("/content/drive/MyDrive/data/Sample_AFIB.csv", names = col_names)
srate = 500 # sampling rate
timestamps = np.arange(0, 10, 1/srate) # time in second
x = timestamps # for x axis
```

```python
fig, axs = plt.subplots(4,1, figsize=(10,8))
label_names = ['SR','SB','ST','AFIB']

axs[0].plot(x, df1[col_names[0]])
axs[1].plot(x, df2[col_names[0]])
axs[2].plot(x, df3[col_names[0]])
axs[3].plot(x, df4[col_names[0]])

for i in range(4):
    axs[i].set_xlabel('Seconds')
    axs[i].set_ylabel('Microvolts')
    axs[i].grid(True)
    axs[i].set_title(label_names[i])
    plt.tight_layout()
```



실습 – Inclass Assignment

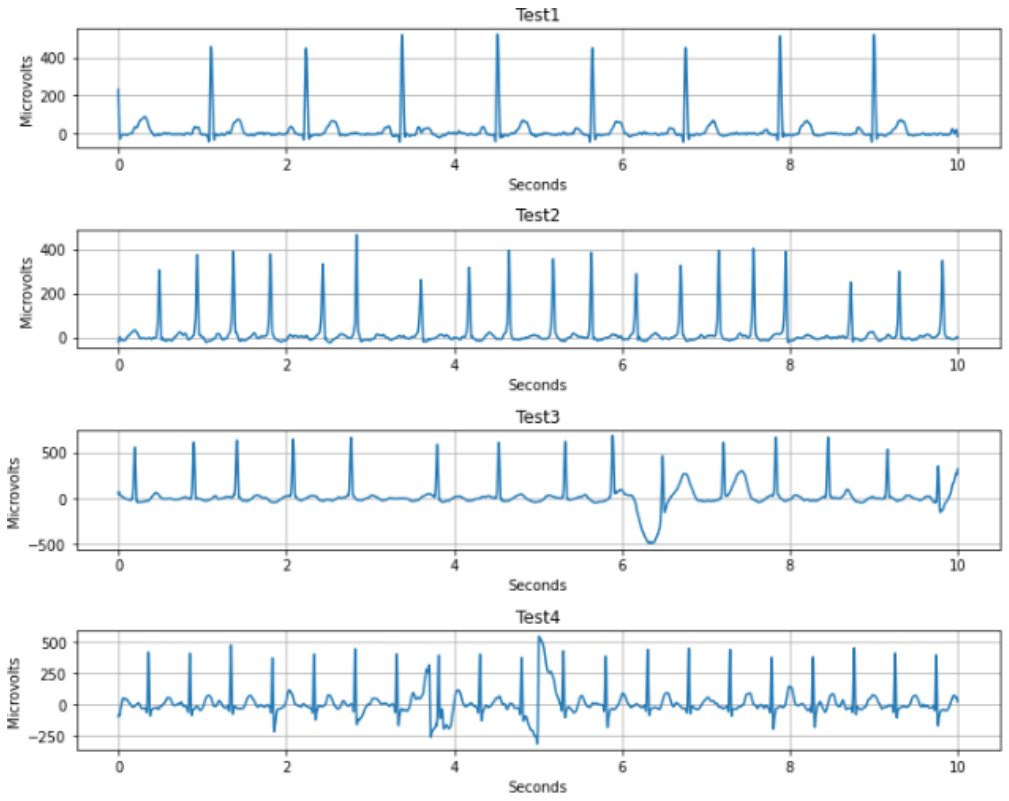
- 함께 올려 둔 테스트 샘플 파일들(Sample_Test1~Sample_Test4)을 그려보고, 각각 어느 label(SR, SB, ST, AFIB)에 해당하는지 추정해보자
- 시각화한 결과를 캡처하여 스노보드에 업로드합니다



실습 – Inclass Assignment

- 함께 올려 둔 테스트 샘플 파일들(Sample_Test1~Sample_Test4)을 그려보고, 각각 어느 label(SR, SB, ST, AFIB)에 해당하는지 추정해보자
- 시각화한 결과를 캡처하여 스노보드에 업로드합니다

	QRS count	Irregularity	Predicted Label
Test1	8	X	SB
Test2	19	O	AFIB
Test3	15	O	AFIB
Test4	20	X	ST



File Information

FileName	Rhythm	Beat	PatientAge	DateofBirth	QRSCount	New FileName
MUSE_20180111_160037_37000	ST	NONE	49	01-01-1969	17	Sample_ST
MUSE_20180111_155154_74000	AFIB	AQW LVHV TWC	83	01-01-1935	18	Sample_AFIB
MUSE_20180209_111606_83000	SR	NONE	69	01-01-1949	11	Sample_SR
MUSE_20180111_155907_78000	SB	1AVB LVHV STTU	68	01-01-1950	9	Sample_SB
MUSE_20180111_160141_83000	SB	1AVB AQW LVHV	77	01-01-1941	8	Sample_Test1
MUSE_20180111_160201_47000	AFIB	LVHV STTC	88	01-01-1930	19	Sample_Test2
MUSE_20180111_160333_03000	AFIB	LVHV	76	01-01-1942	15	Sample_Test3
MUSE_20180111_160352_71000	ST	NONE	79	01-01-1939	20	Sample_Test4

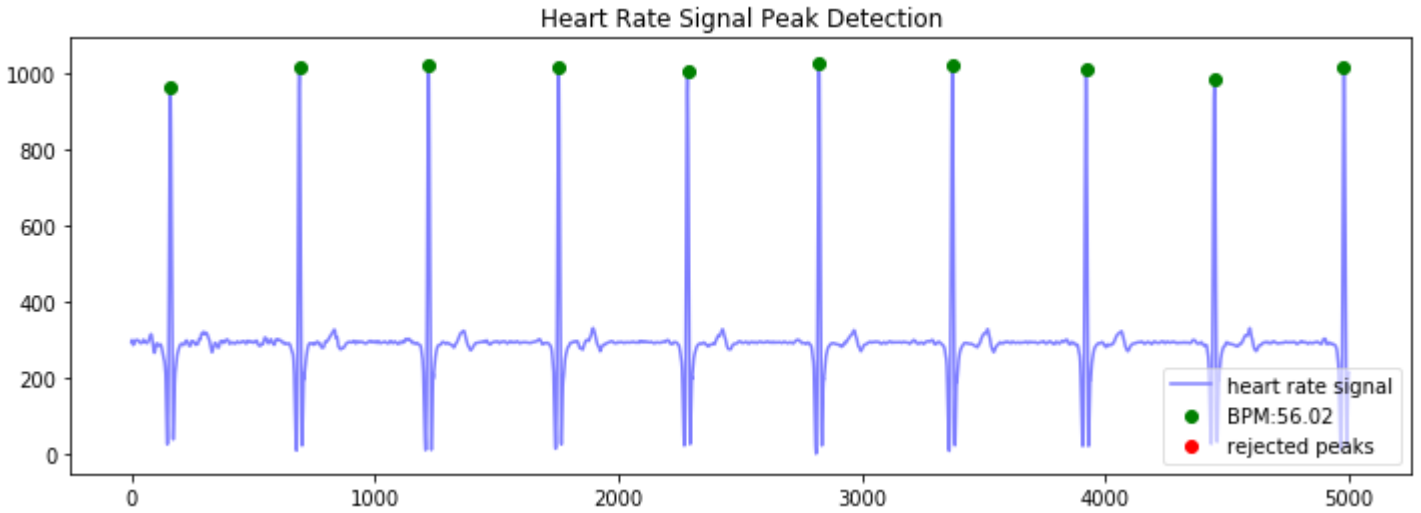
생체 데이터의 종류

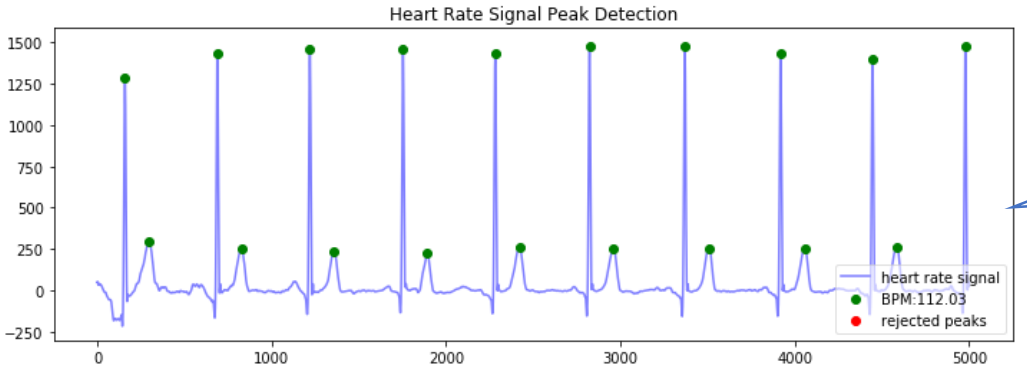
ECG와 심박변이도

실습

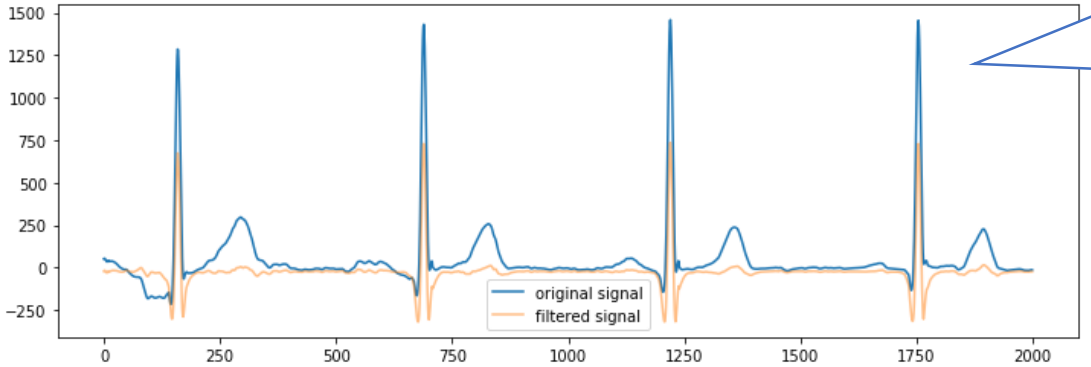
- 지난 실습에서 사용했던 샘플 데이터에 HRV 분석을 제공하는 Python Toolkit을 활용하여 Peak Detection 및 HRV 추출을 해보자
  - ✓ HeartPy (https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/index.html)
  - ✓ 샘플 데이터의 sampling rate은 500Hz이다

```
import heartpy as hp
wd, m = hp.process(data, sample_rate)
hp.plotter(wd, m)
```



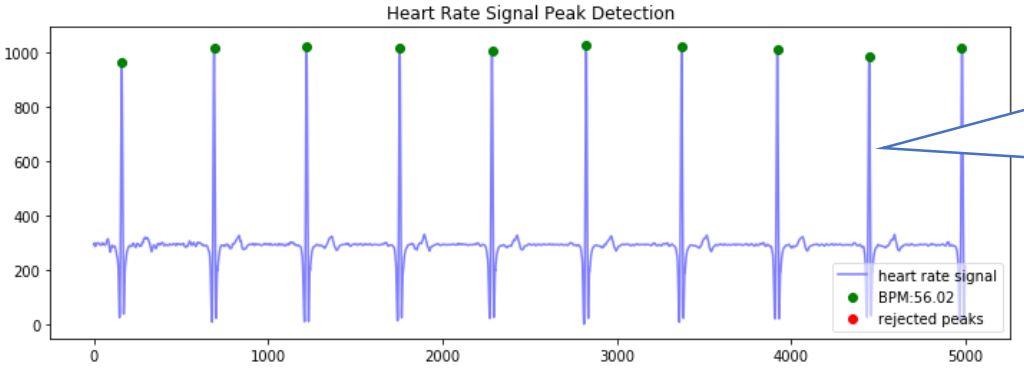


T wave를 R로 오인식하여 bpm이 빠르게 측정되는 문제 발생



T wave를 약화시키기 위한 필터링

filtered = hp.filter_signal(data, cutoff = 0.05, sample_rate = sample_rate, filtertype='notch')



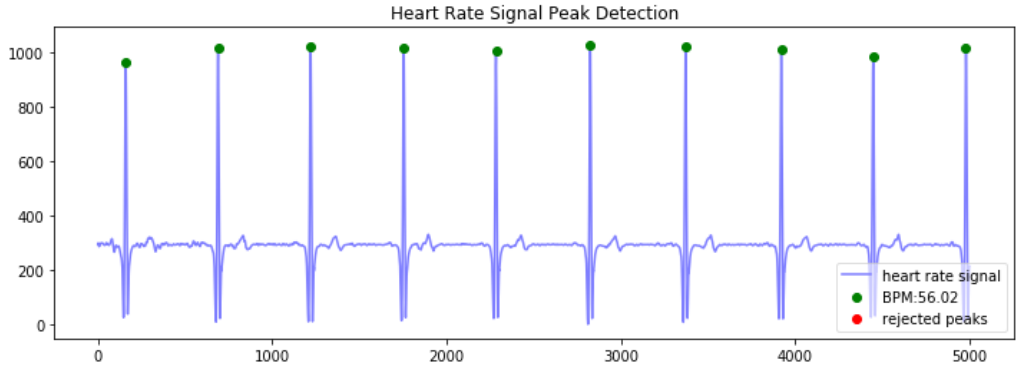
필터링 이후 R이 제대로 인식되는 것을 확인할 수 있음

wd, m = hp.process(hp.scale_data(filtered), sample_rate)

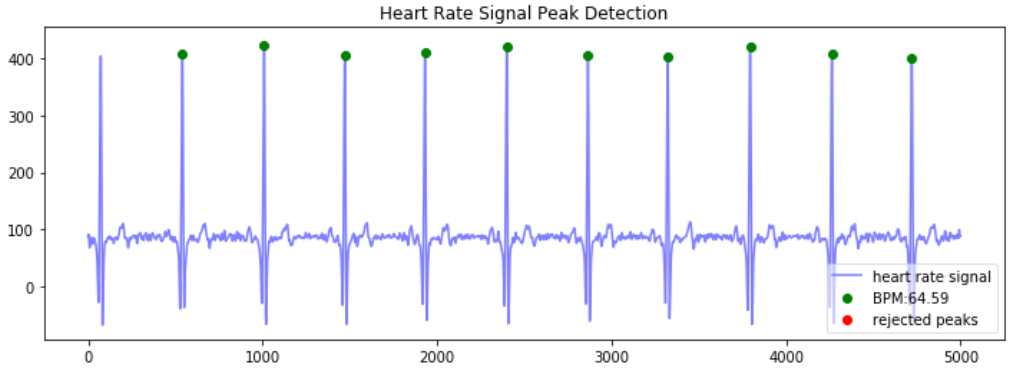
Inclass Assignment

- 시각화한 결과를 캡처하여 스노보드에 업로드합니다

Sinus Bradycardia
BPM 56.02



Sinus Rhythm
BPM 64.59



Sinus Tachycardia
BPM 104.21