



南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

EEG data analysis Hands on

10/15/2022

EEG分析流程

```
Run Cell | Run Below | Debug Cell | Go to [1]
#%%
import os
import numpy as np
np.__version__
Run Cell | Run Above | Debug Cell | Go to [3]
#%%
import pandas as pd
import matplotlib.pyplot as plt
import mne
import os.path as op

import numpy as np
import matplotlib.pyplot as plt

import mne
from mne.time_frequency import tfr_morlet, psd_multitaper, psd_welch
from mne.datasets import somato

from mne.channels import compute_native_head_t, read_custom_montage
```

- MNE-Python是一个基于Python的用于探索、可视化与分析人脑神经生理数据的开源软件,
- MNE-Python的代码设计基于面向对象的编程思想, 其具有高效(多核的CPU和GPU计算)、易用(代码简单并可进行可视化)且灵活(对各种数据的支持)三大特性。

EEG分析流程

```
'''读取数据，这是刺激下采集的数据，所以也会有事件'''
```

```
# 参考网址: https://mne.tools/stable/auto\_tutorials/io/20\_reading\_eeg\_data.html
```

```
sample_data_folder = mne.datasets.sample.data_path(download=True)
sample_data_raw_file = os.path.join(sample_data_folder, 'MEG', 'sample',
                                     'sample_audvis_raw.fif')
```

自动下载数据集

```
# 读取事件
```

```
events_file = os.path.join(sample_data_folder, 'MEG', 'sample',
                             'sample_audvis_raw-eve.fif')
events = mne.read_events(events_file)
```

```
raw = mne.io.read_raw_fif(sample_data_raw_file, preload=False)
```

原始 FIF 数据的读取器功能

```
raw.pick(['eeg']).load_data() # 挑选EEG通道
raw.load_data()
```

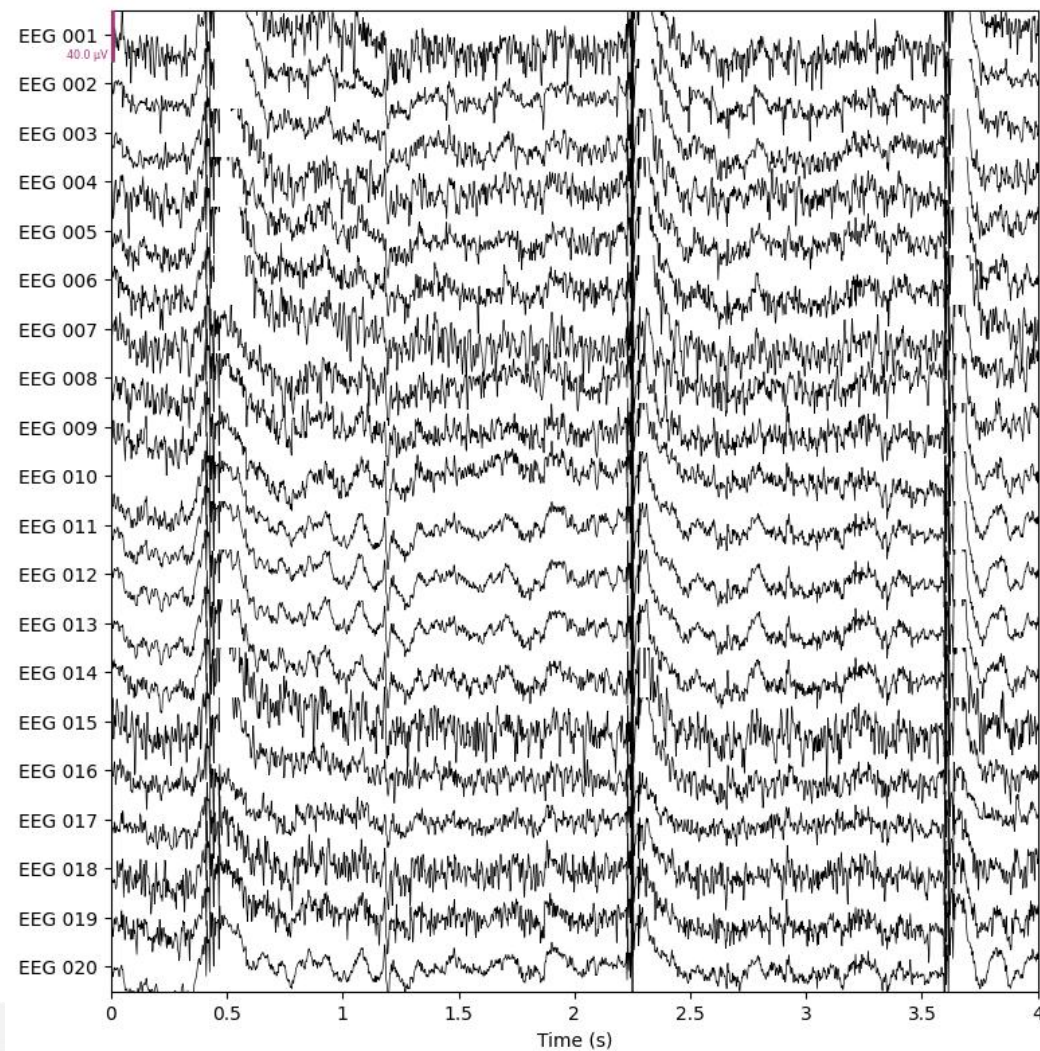
挑选EEG通道

```
raw.plot(duration=4, show_scrollbars=False)
```

展示0~4s间的EEG信号

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

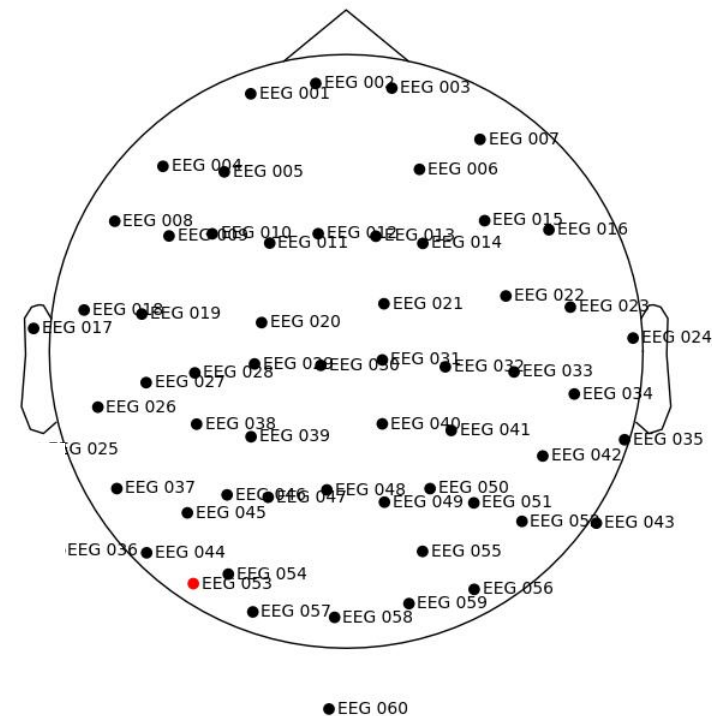
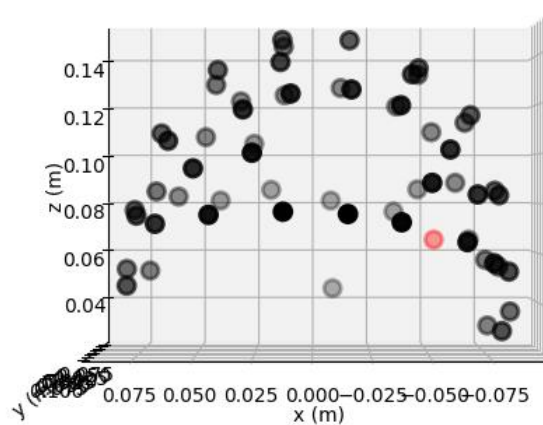
EEG分析流程



展示了20个通道的EEG信号，一共有64个通道

EEG分析流程

```
Run Cell | Run Above | Debug Cell | Go to [2]
# %%
# 可视化电极分布
raw.plot_sensors(show_names=True)
fig = raw.plot_sensors('3d')
```



EEG分析流程

```
''' 滤波与重采样 '''  
# 参考网址: https://mne.tools/stable/auto\_tutorials/preprocessing/30\_filtering\_resampling.html  
  
raw.filter(l_freq=0.1, h_freq=70)  
raw.resample(sfreq=256)
```

滤波器简介

滤波器可以用于去除或衰减部分信号。通常，滤波器作用于信号的特定频率范围——例如，抑制高于或低于某个值的所有频率成分。设计数字滤波器的方法有很多种；有关在 MNE-Python 中过滤生理信号的各种方法的详细讨论，请参阅有关过滤的背景信息。

通过滤波器去除伪迹

有时可以通过过滤数据来修复受限于一定频率范围内的伪迹。频率受限于一定范围内的伪迹的两个示例是缓慢漂移（slow drift）和电源线噪声（power line noise）。

EEG分析流程

Measurement date	December 03, 2002 19:01:10 GMT
Experimenter	MEG
Participant	Unknown
Digitized points	146 points
Good channels	60 EEG
Bad channels	EEG 053
EOG channels	Not available
ECG channels	Not available
Sampling frequency	256.00 Hz
Highpass	0.10 Hz
Lowpass	70.00 Hz
Filenames	sample_audvis_raw.fif
Duration	00:04:38 (HH:MM:SS)

EEG分析流程

```
Run Cell | Run Above | Debug Cell
#%%
''' 利用ICA降噪 '''
# 参考网址: https://mne.tools/stable/auto\_tutorials/preprocessing/40\_artifact\_correction\_ica.html

ica_raw = raw.copy()
ica = mne.preprocessing.ICA(n_components=20, random_state=97, max_iter=800)
ica.fit(ica_raw)

# 可视化ICA分解的成分，进一步挑选要剔除的成分从而实现降噪
ica.plot_components()
ica.plot_sources(raw, show_scrollbars=True)

ica.exclude = [0, 18, 19] # 选择要剔除的独立成分

ica.apply(ica_raw)

ica_raw.plot(duration=4, show_scrollbars=False)
```

在统计学中，独立成分分析或独立分量分析（Independent components analysis, 缩写：ICA）是一种利用统计原理进行计算的方法。它是一个线性变换。这个变换把数据或信号分离成统计独立的非高斯的信号源的线性组合。独立成分分析是盲源分离（Blind source separation）的一种特例。

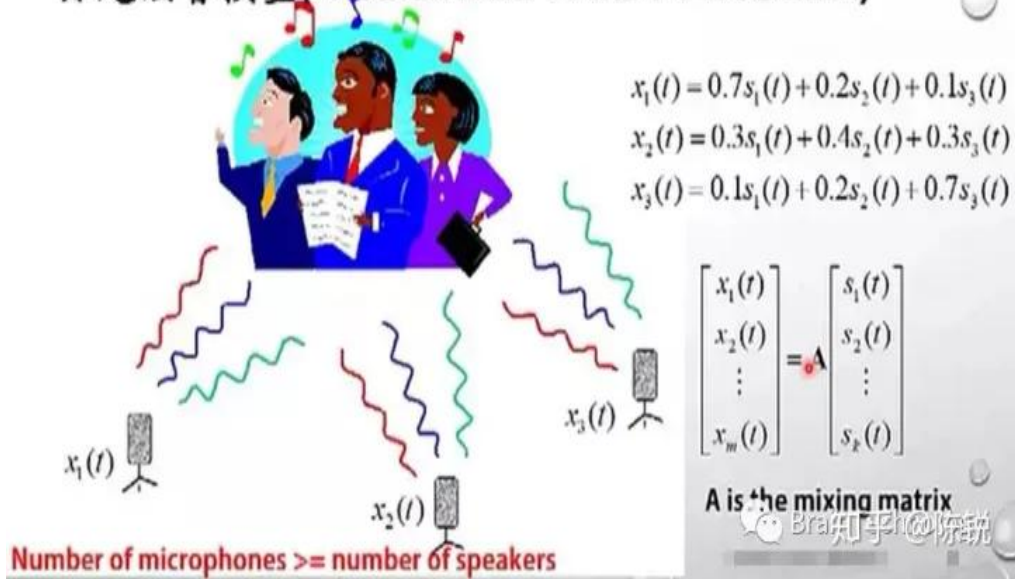
在信号处理中，独立成分分析（ICA）是一种用于将多元信号分离为加性子分量的计算方法。这是通过假设子分量是非高斯信号，并且在统计上彼此独立来完成的。ICA是盲源分离的特例。一个常见的示例应用程序是在嘈杂的房间中聆听一个人的语音的“鸡尾酒会问题”。

EEG分析流程

「鸡尾酒会问题」(Cocktail Party Problem) 诞生于 1953 年，是语音识别领域的经典问题，指的是人们在鸡尾酒会中交谈，语音信号会重叠在一起，机器需要将它们分离成独立的信号。

一个音乐乐团在音乐会上演奏，你想尽可能产生较少的噪音来获取最干净的信号，但这是一项非常困难的事情。明智地，你在音乐厅，舞台和一排排座位上放置并牢固地安装了几个麦克风。此外，你可以安全地假设舞台上的所有演奏者在演唱会期间都是静止的（你不希望他们在演奏时站起来走动），每个乐器组都会播放他们自己的旋律。在一个激动人心的夜晚之后，每个麦克风都会捕获混合原始信号的录音，实际上您将拥有与麦克风一样多的信号混合。由于麦克风放置在音乐厅，混合信号在不同的麦克风上略有不同。现在你的最终目标是将混合信号分离提取或重建“纯信号”。虽然乐器组演奏的不同旋律当然应该保留，但是应该消除来自观众的声音噪音或者其它的一些环境噪声。

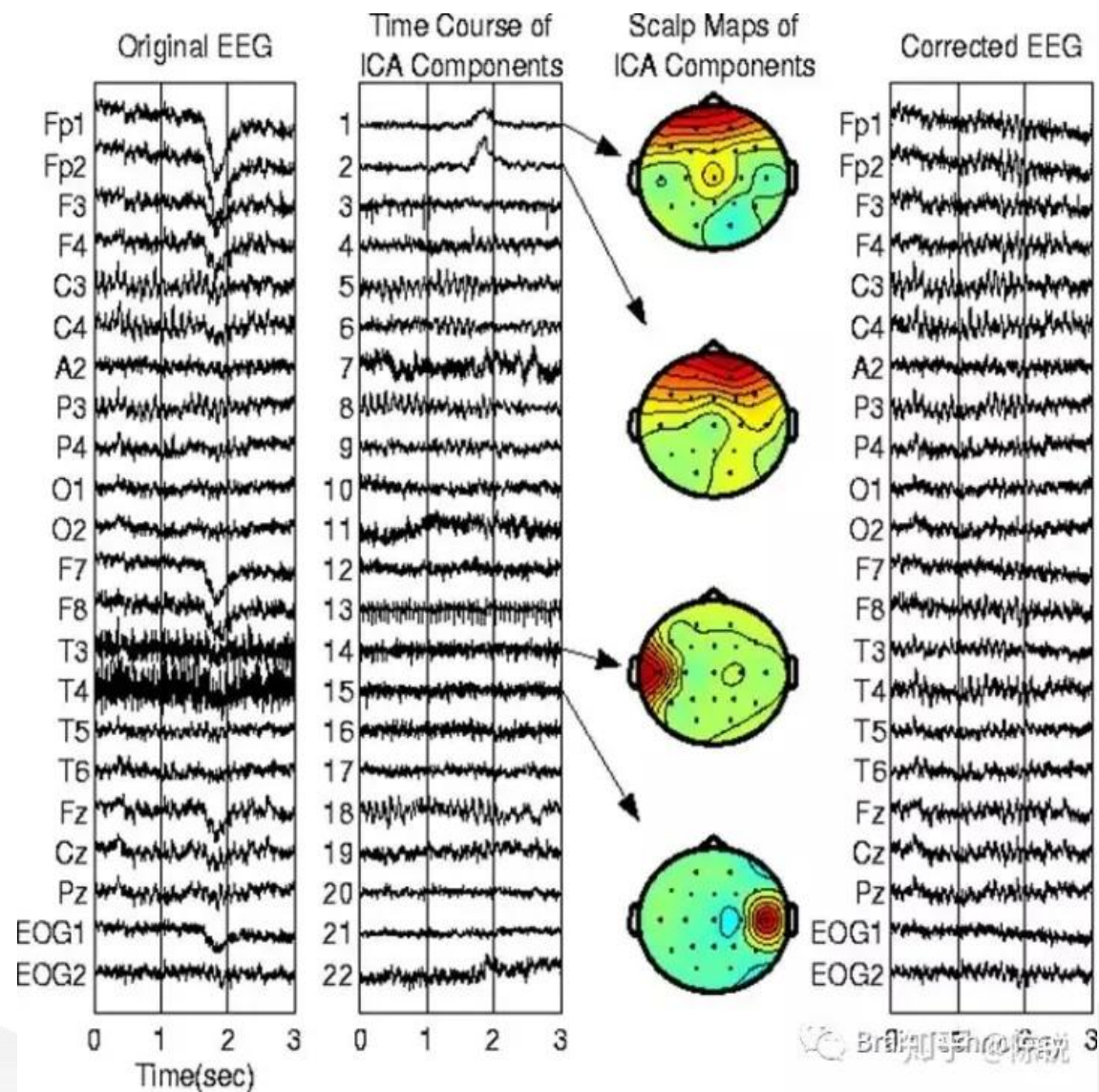
雞尾酒會模型(A COCKTAIL-PARTY PROBLEM)



EEG分析流程

该模型可以以简单的方式转移到EEG记录上，从脑电图电极记录的通道信号（由微伏振幅的时间序列表示）可以被认为是一组脑信号的混合信号，据推测这些混合信号通过皮层和皮质下区域中的神经元簇的同步产生，触发远场电位。虽然神经元本身局部是静止的并且不移动，但是激活模式基于体积传导的原理混合和合并，传播通过皮层，颅骨和组织的所有层，并且最终存在于任何头皮部位。同样，分析的理想结果是混合信号分离开逐步分析。

在示例中，目标是从记录的混合信号中提取统计上的“纯信号”，以便允许选择要保留的信号和丢弃信号。确切地说，这可以通过独立成分分析来完成（ICA; Makeig et al. 1996）。该技术已被公认为减轻伪影和分析头皮和颅内脑电图记录中统计独立皮质过程的有力工具。特别是在EEG数据记录使用有限的电极中，记录诸如眨眼或肌肉活动的伪迹时（例如在患者组，儿童或受试者自由移动的移动EEG实验中），ICA可能优于伪影阈值去伪。



EEG分析流程

ICA使用要求：

满足以下要求，ICA从信号混合信号中提取纯信号（详见 Jung et al. 1998）：

- 1.由ICA提取的纯信号的特征在于它们的时间过程（神经激活模式），其在统计上独立于任何其它信号。事实上，这些激活是独立的成分（IC）。
- 2.纯信号（神经元簇）的发生以及记录位置（电极）在整个记录过程中是静止的。因此，成分朝向记录位置的地形位置是固定的。
- 3.混合是线性的，传播延迟可以忽略不计。
- 4.各个IC激活的概率分布不是精确的高斯分布。

除此之外，ICA没有对数据提出任何进一步的要求。事实上，它完全不知道信号的性质，这就是为什么ICA通常被称为盲源分离算法（Hyvärinen and Oja 2000）。请记住，IC成分纯粹是统计属性，因此它们不会将1:1映射到生理过程。在EEG / MEG数据的背景下，可以基于它们的时间过程（和地形分布）来检查所提取的IC成分，并且可以去除表示噪声，伪迹或其它非大脑过程的IC。然后，校正的一组独立分量反向逆算，这就可以在电极上修改原始信号，将其中的伪迹信号进行校正。

EEG分析流程

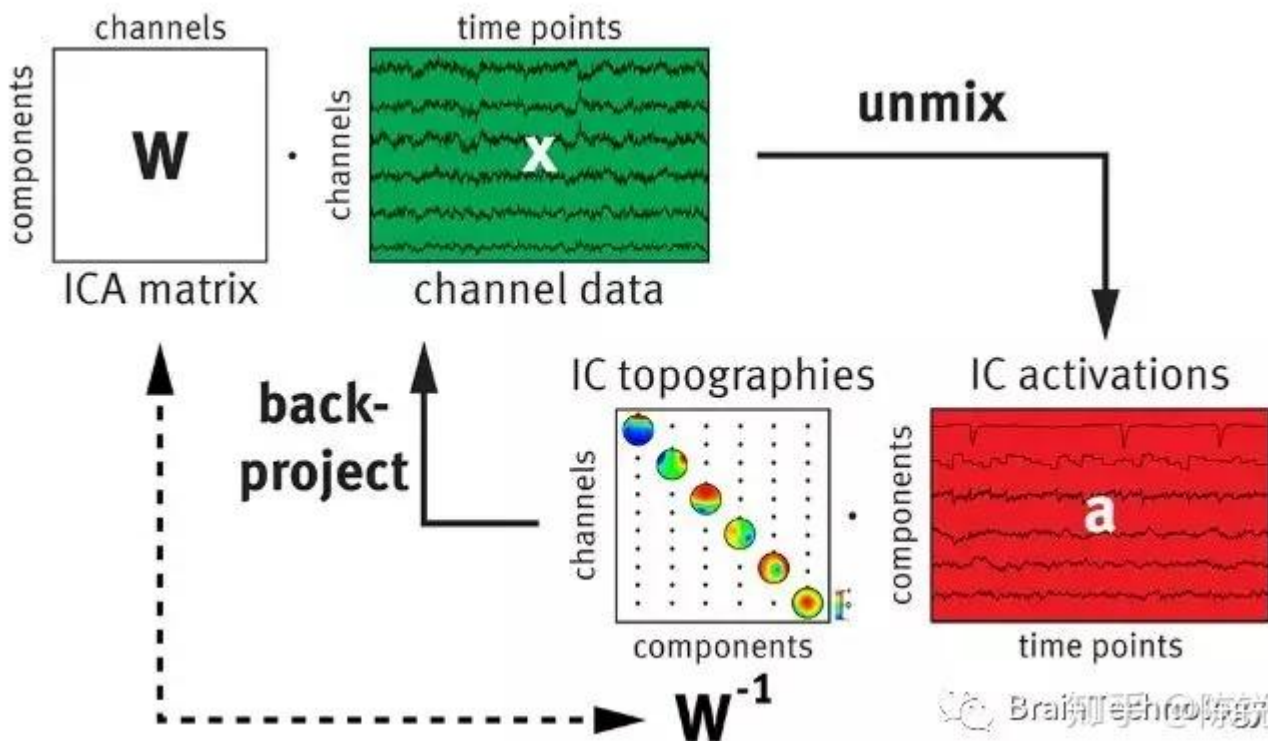
ICA的算法基础

ICA的数学基础，其中包括矩阵运算。通常，我们从电极上记录的时间上的混合信号开始。电极数据可以表示为2-D矩阵，其中行表示通道，列表示采样点。矩阵中的值是每个通道和采样点的记录电压幅值。我们称这个矩阵为 x 。我们现在可以生成解混矩阵 W （ICA矩阵），当与数据矩阵 x 相乘时，将混合数据 x 转换为IC激活 a 。

$$Wx = a$$

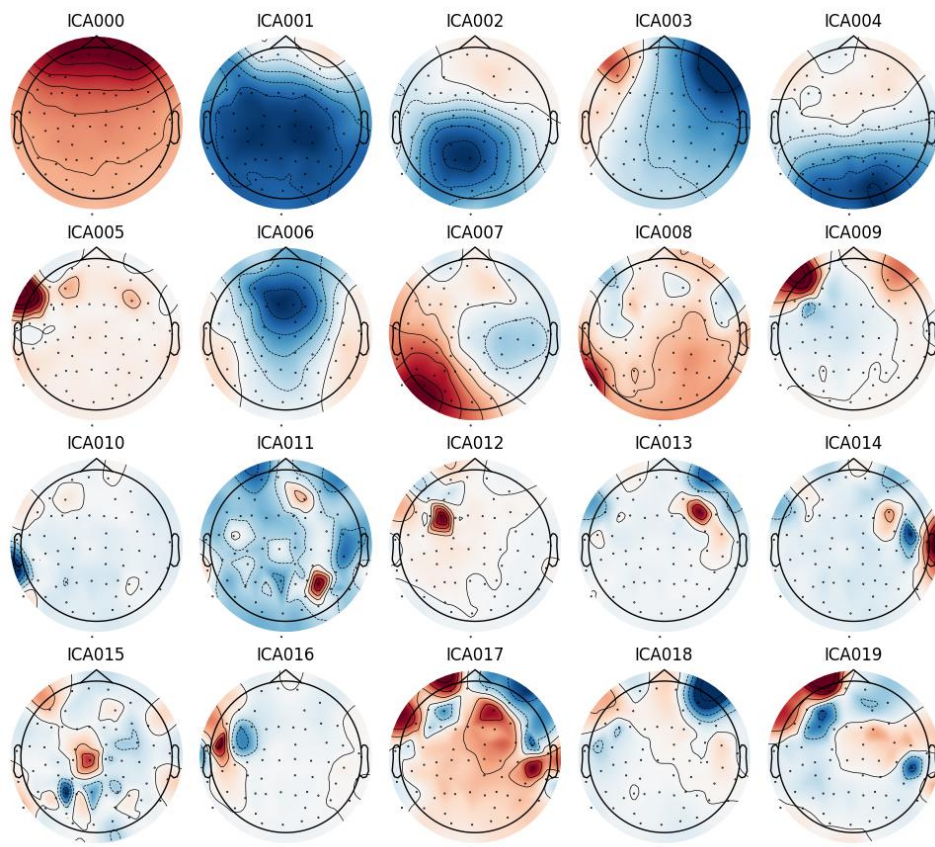
矩阵 a 的每一行代表一个成分，每列代表一个采样点。成分 a 的激活可以表征为加权通道激活的线性和。

从矩阵表示可以看出，你可以随时提取最多与电极一样多的成分（Makeig等，1997），因为来自 N 的混合信号电极被分解成 N 个分量的线性加权和。因此，ICA可以表征为完整的分解技术。这意味着可以通过将IC激活矩阵 a 乘以逆矩阵来反演解混合处理。ICA的计算目标是找到一个解混矩阵 W ，以便实现所有成分的最大时间独立性。事实上，找到 W 是计算中最耗时的部分，在BrainVision Analyzer中提供的ICA组件可进行数据的计算。

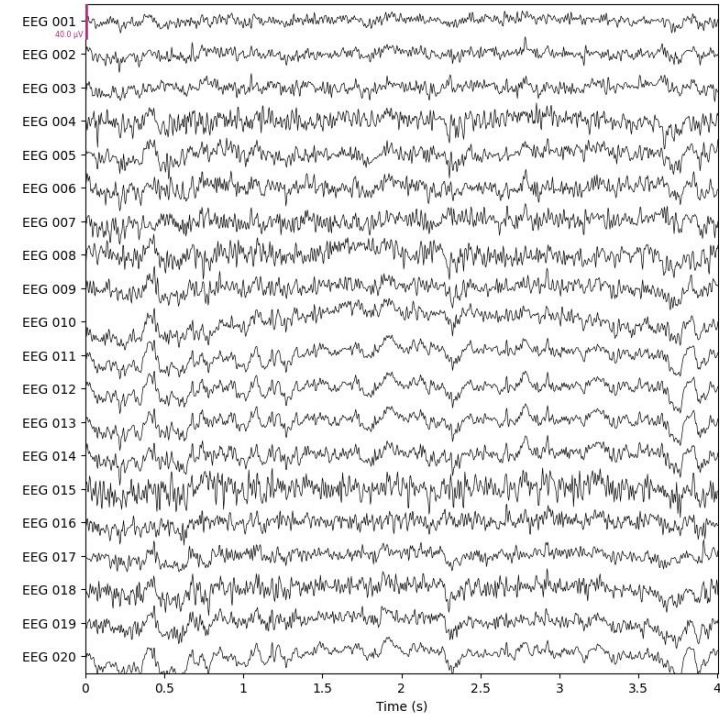
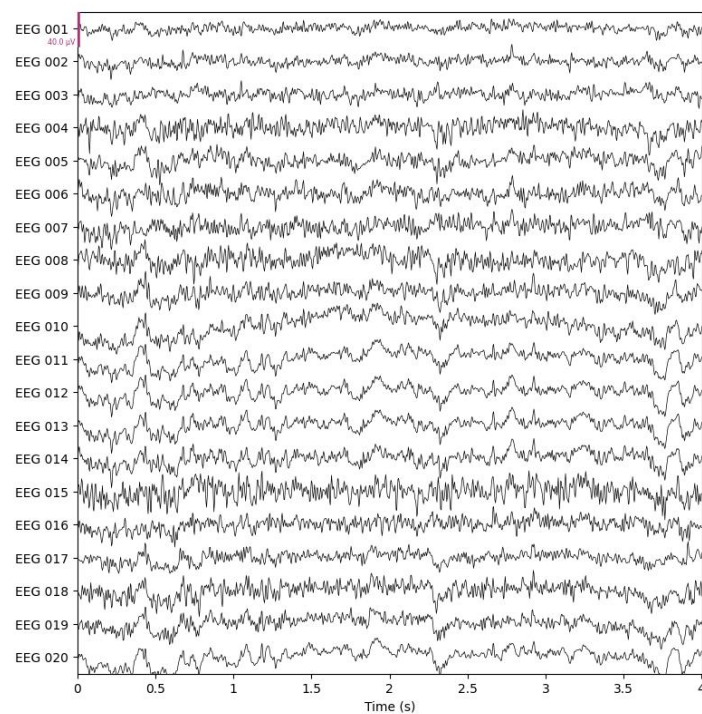
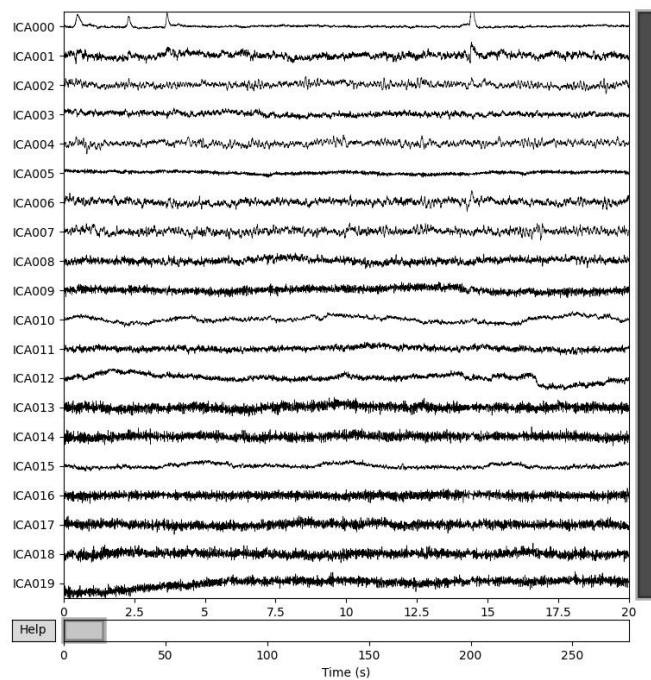


EEG分析流程

ICA components



EEG分析流程



EEG分析流程

```
Run Cell | Run Above | Debug Cell | Go to [4]
#%%
''' power spectral density '''

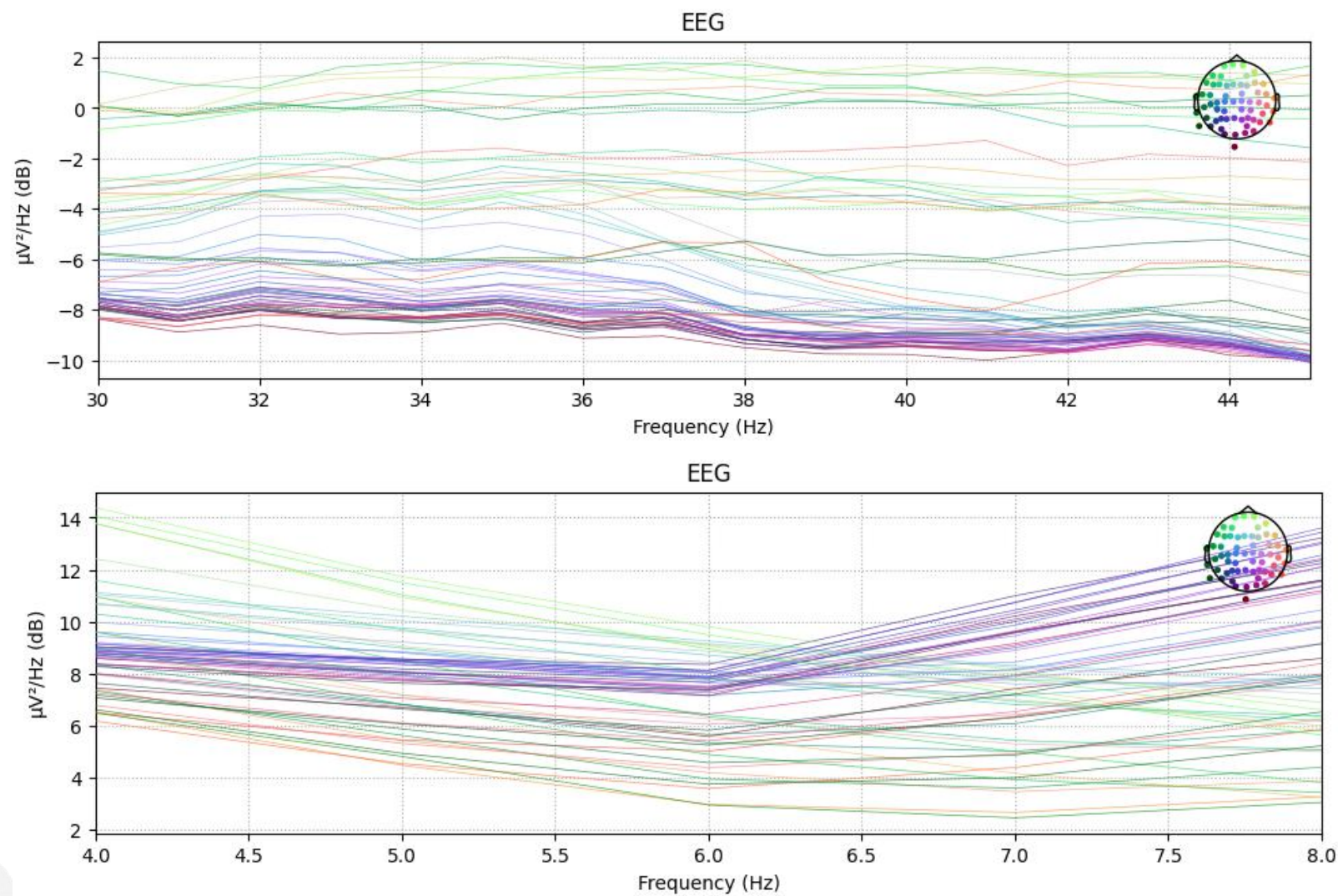
# 可视化完整的PSD
raw.plot_psd(fmax=70)

# 分频段可视化
raw.plot_psd(fmin=0, fmax=4)
raw.plot_psd(fmin=4, fmax=8)
raw.plot_psd(fmin=8, fmax=12)
raw.plot_psd(fmin=12, fmax=30)
raw.plot_psd(fmin=30, fmax=45)
```

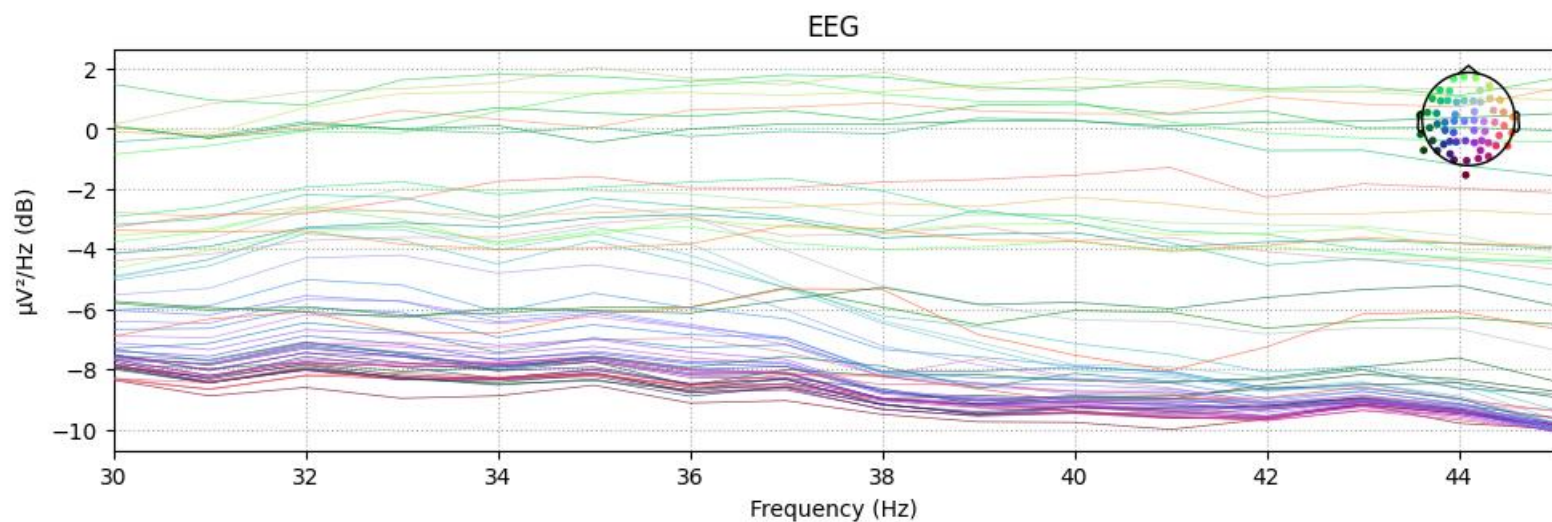
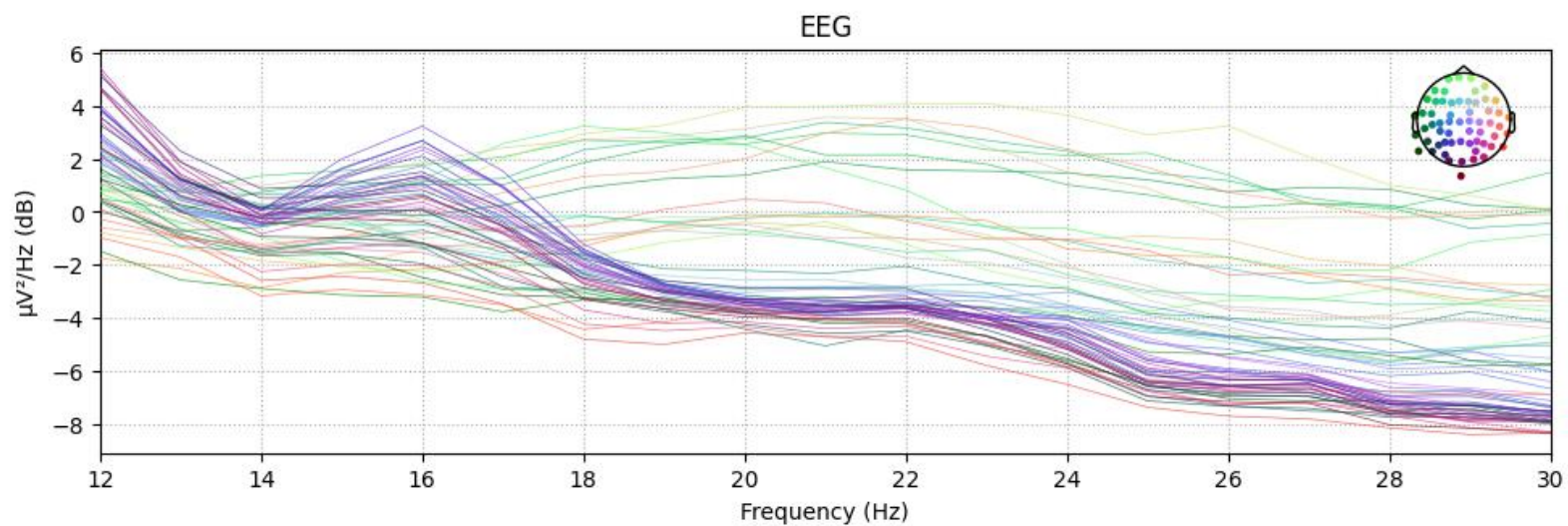
在物理学中，信号通常是波的形式表示，例如电磁波、随机振动或者声波。当波功率频谱密度乘以一个适当的系数后将得到每单位频率波携带的功率，这被称为信号的功率谱密度（power spectral density, PSD）

它定义了信号或者时间序列的功率如何随频率分布。这里功率可能是实际物理上的功率，或者更经常便于表示抽象的信号被定义为信号数值的平方，也就是当信号的负载为1欧姆(ohm)时的实际功率。

EEG分析流程



EEG分析流程



EEG分析流程

```
Run Cell | Run Above | Debug Cell | Go to [6]
#%%
''' 等分数据, 转为epoch, 进行ERP分析 '''
# 参考网址: https://mne.tools/stable/auto\_tutorials/evoked/30\_eeg\_erp.html

# 数据记录的事件的index
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)
fig = epochs.plot(events=events)

# 一种刺激（比如下面的左侧听觉刺激）会做多次，得到多个trials，平均多个trials的结果，可以把噪声平均掉（假设噪声均值为0），从而得到ERP

l_aud = epochs['auditory/left'].average()
fig1 = l_aud.plot(spatial_colors=True)
l_aud.plot_topomap(times=[-0.2, 0.1, 0.4], average=0.05)
```

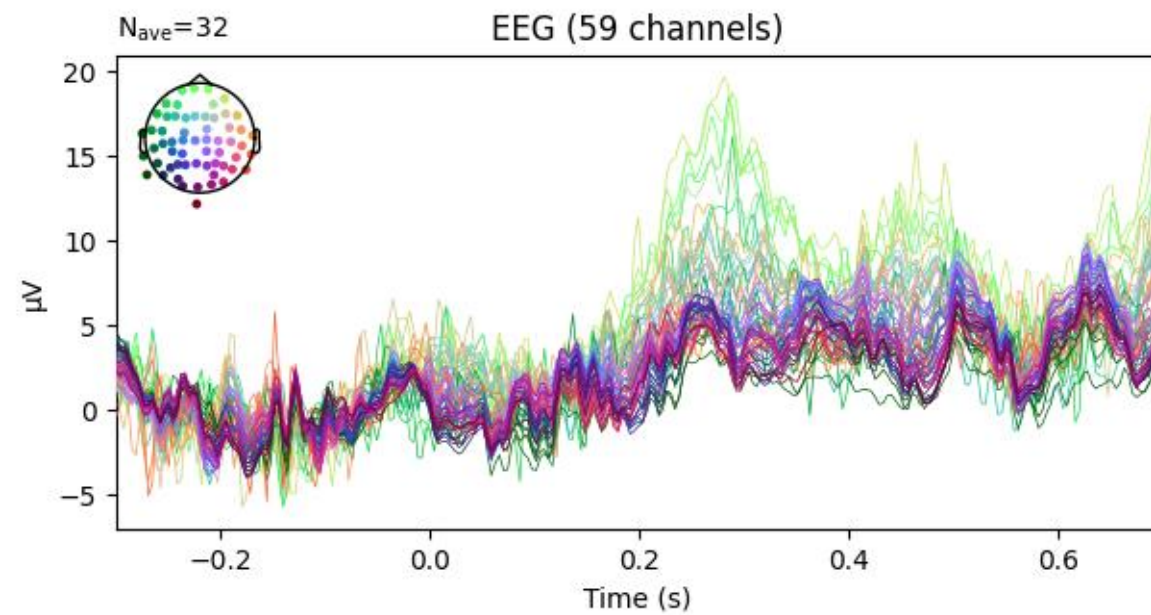
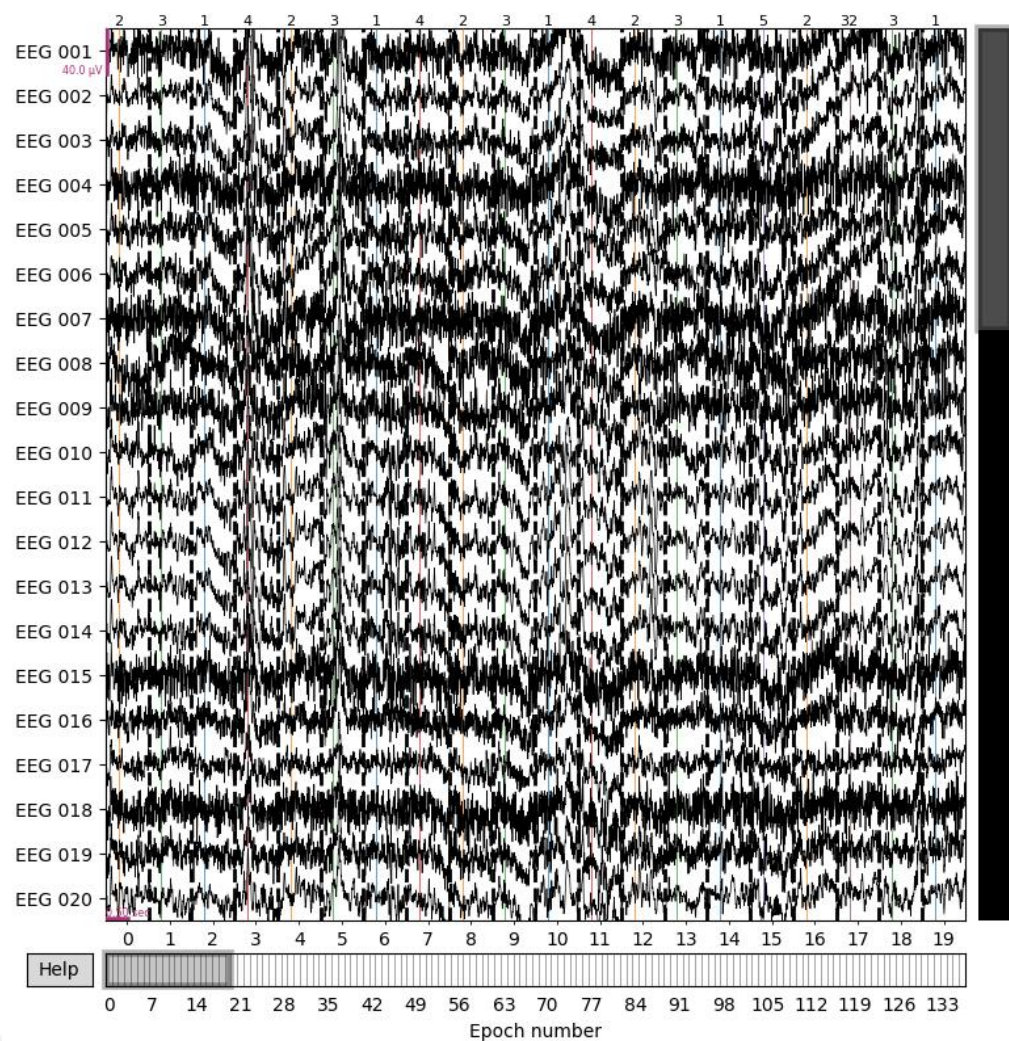
事件相关电位（ERP, event-related potential）是一种特殊的脑诱发电位，通过有意地赋予刺激以特殊的心理意义，利用多个或多样的刺激所引起的脑的电位。它反映了认知过程中大脑的神经电生理的变化，也被称为认知电位，也就是指当人们对某课题进行认知加工时，从头颅表面记录到的脑电位。

EEG分析流程

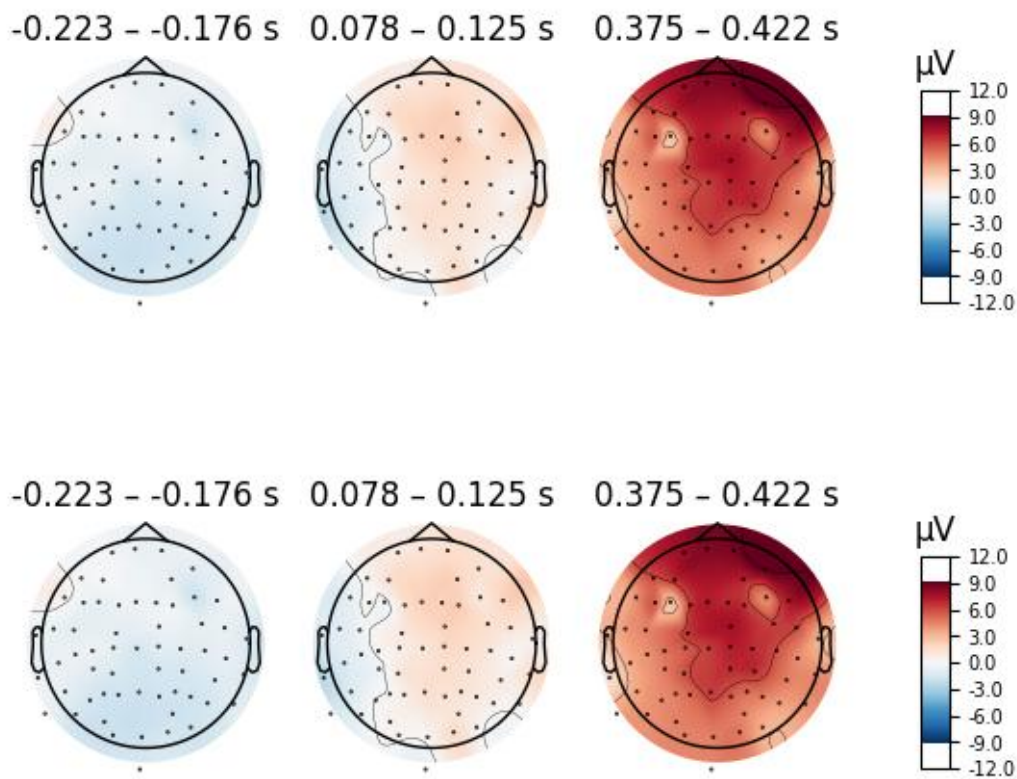
事件相关电位（ERP）是一种特殊的脑诱发电位，诱发电位(Evoked Potentials, EPs)，也称诱发反应(Evoked Response)，是指给予神经系统(从感受器到大脑皮层)特定的刺激，或使大脑对刺激(正性或负性)的信息进行加工，在该系统和脑的相应部位产生的可以检出的、与刺激有相对固定时间间隔(锁时关系)和特定位相的生物电反应。诱发电位应具备如下特征：

1. 必须在特定的部位才能检测出来；
2. 都有其特定的波形和电位分布；
3. 诱发电位的潜伏期与刺激之间有较严格的锁时关系，在给予刺激时几乎立即或在一定时间内瞬时出现。

EEG分析流程



EEG分析流程



脑地形图，脑电波各频段内功率值用不同颜色表示的球面头皮展成的平面图形。

脑地形图是脑功能研究和临床诊断的重要手段。

脑地形图，是一项具先进的，新的检查方法。既能进行病理诊断又可进行功能诊断，具有较高的敏感性，比脑电图带来更多的信息，优于常规脑电图检查。

脑地形图主要用于精神病、痴呆，癫痫、脑肿瘤、脑外伤、脑血管病的辅助诊断。

EEG分析流程

```
Run Cell | Run Above | Debug Cell
# %%
''' 针对epoch进行频域分析 '''

epochs.plot_psd()

''' 针对epoch进行时频分析 '''

frequencies = np.arange(7, 30, 3)
power = mne.time_frequency.tfr_morlet(epochs, n_cycles=2, return_itc=False,
                                     freqs=frequencies, decim=3)

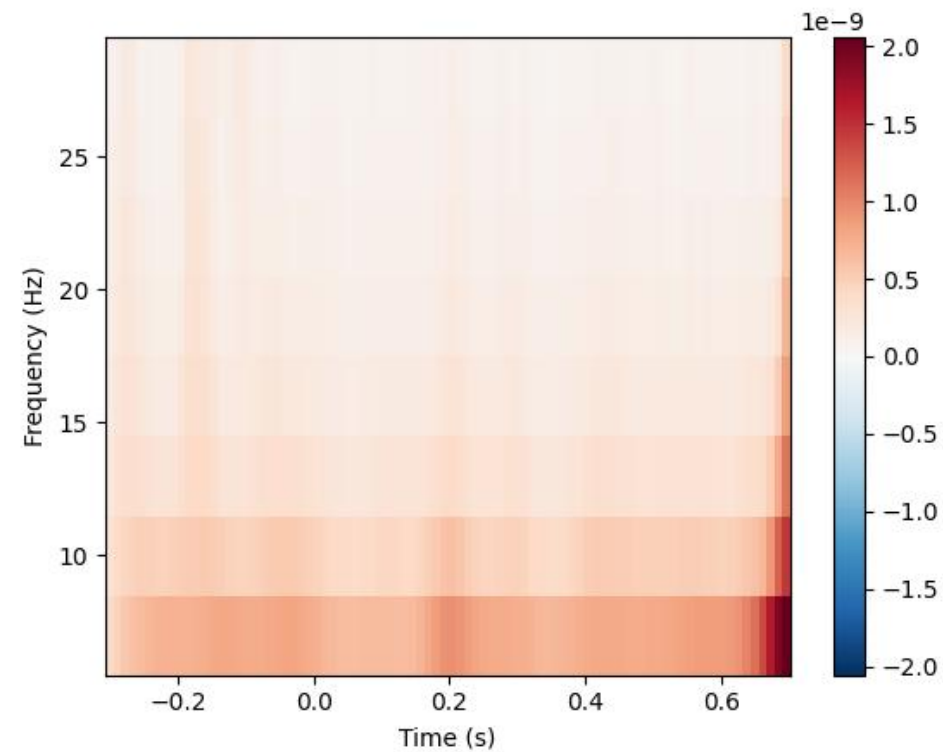
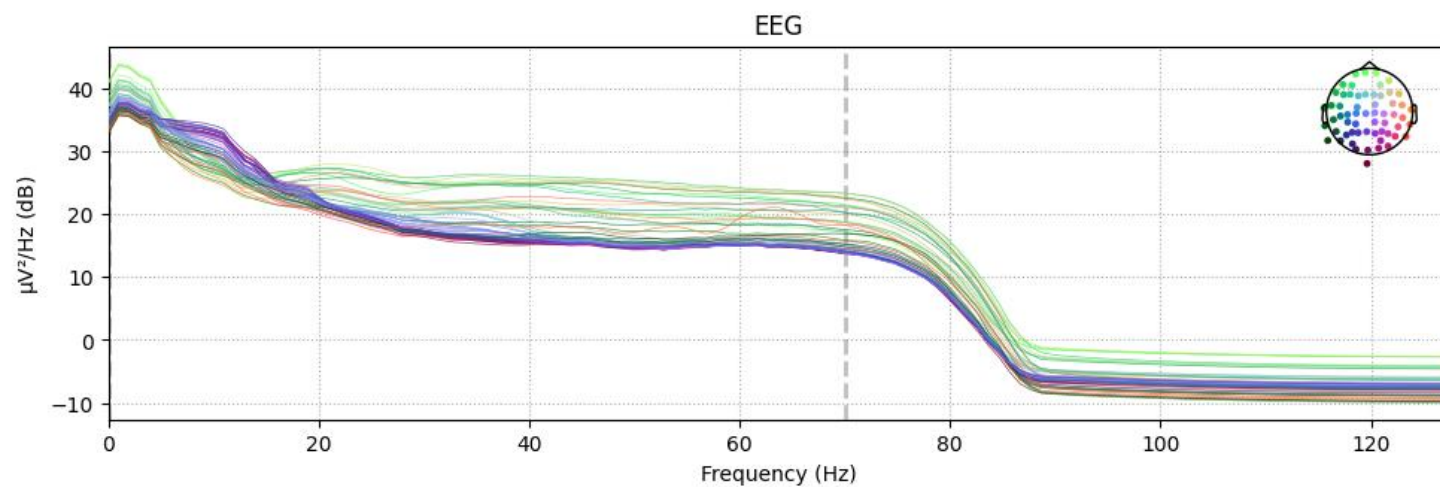
power.plot('EEG 035') # 这里随便选了一个通道可视化
```

时频分布是一项让我们能够同时观察一个讯号的时域和频域资讯的工具，而时频分析就是在分析时频分布。传统上，我们常用傅里叶变换来观察一个讯号的频谱。然而，这样的方法不适合用来分析一个频率会随着时间而改变的讯号，由于傅里叶变换只分析了一维的讯号分布，而时频分析却能分析二维(时域跟频域)的讯号分布，因此在讯号处理中更常被运用。

时频分析也可以说是傅里叶分析的一般化，通常用于频率特性会随时间而变化的讯号上，而在日常生活中符合此特性的讯号非常多，像是演讲、音乐、影像、医学讯号...等，因此能应用的领域相当广泛。

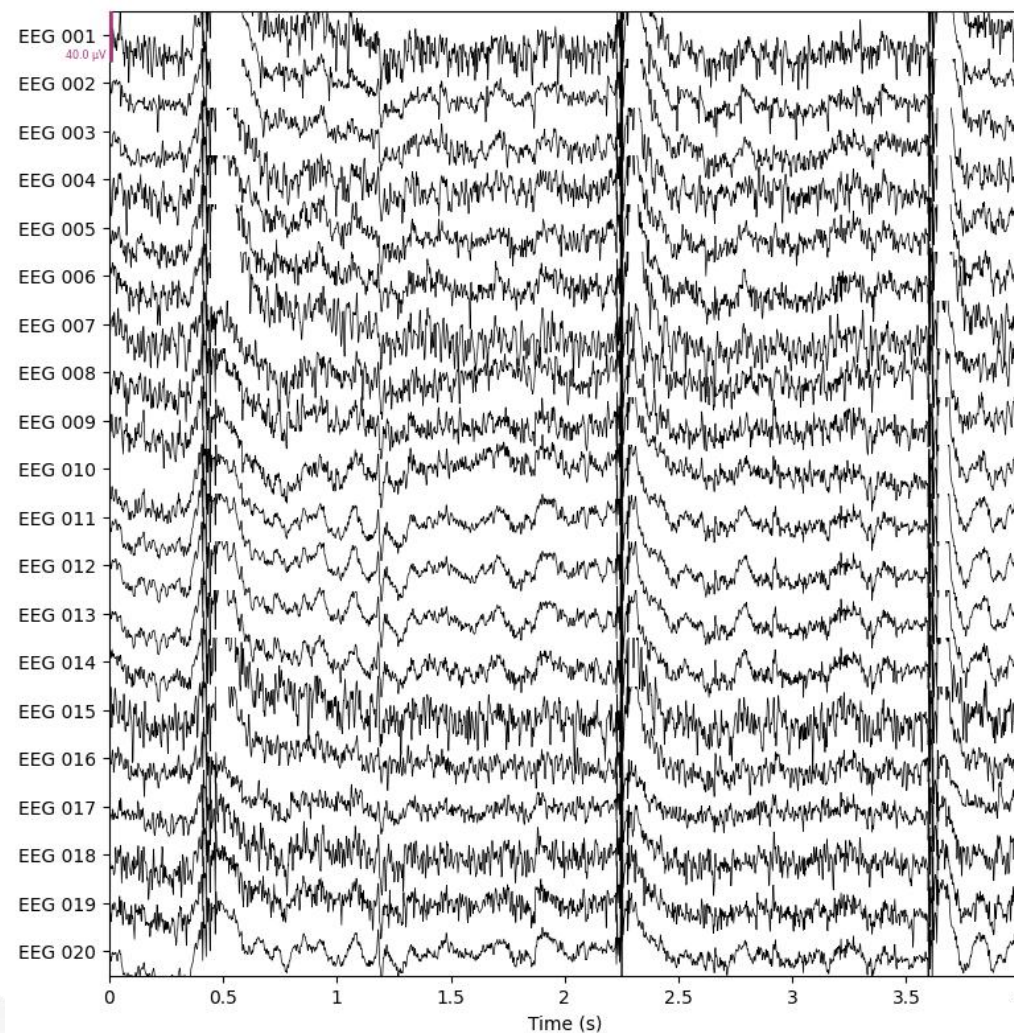
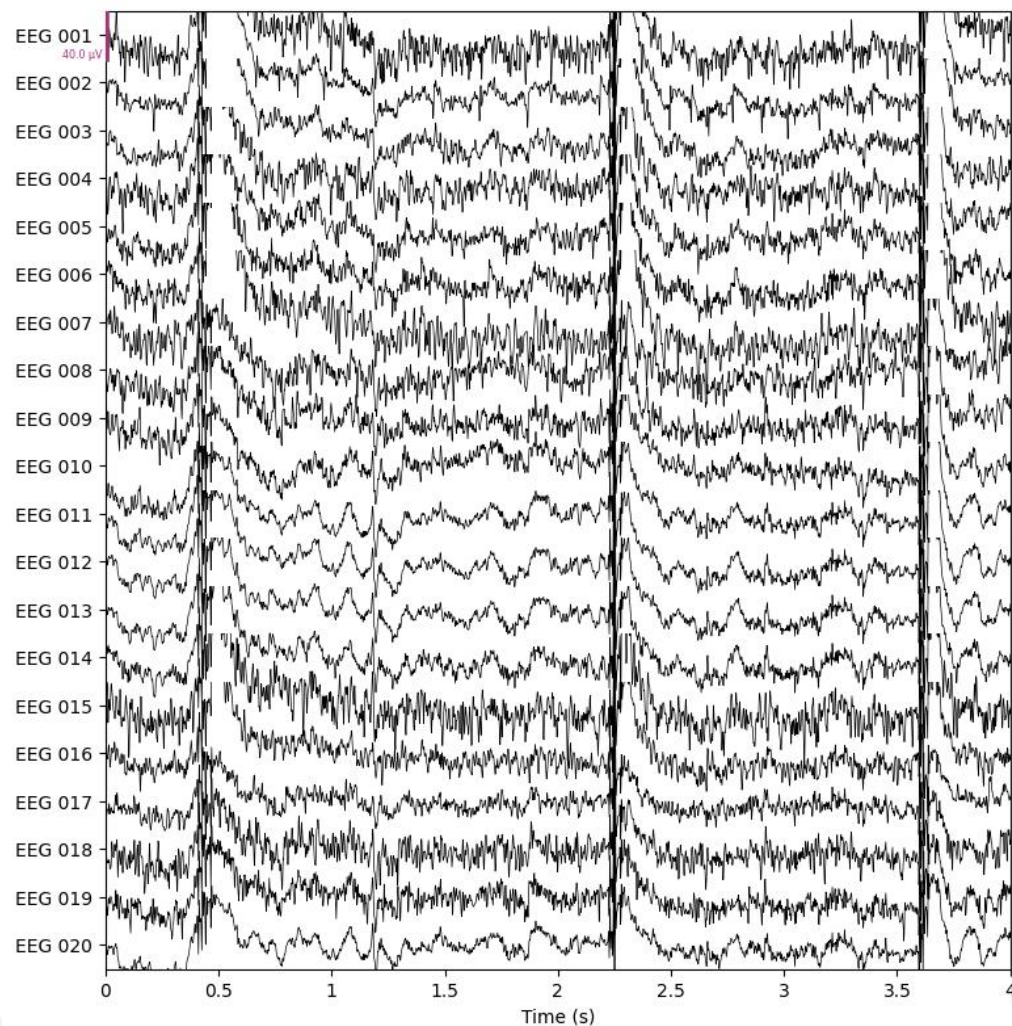
常见的时频分布函数有短时距傅里叶变换（包含加伯转换）、科恩分布函数（包含韦格纳分布）、改进型韦格纳分布，以及加伯-韦格纳分布（Gabor-Wigner distribution function）函数及S转换等。

EEG分析流程



THE END

EEG分析流程



EEG分析流程

Run Cell | Run Above | Debug Cell | Go to [20]

#%%

''' 预处理 '''

''' 滤波 '''

`raw.filter(l_freq=0.1, h_freq=70)`

Run Cell | Run Above | Debug Cell | Go to [22]

#%%

''' 重采样 '''

`raw.resample(sfreq=256)`

EEG分析流程

Measurement date	December 03, 2002 19:01:10 GMT
Experimenter	MEG
Participant	Unknown
Digitized points	146 points
Good channels	60 EEG
Bad channels	EEG 053
EOG channels	Not available
ECG channels	Not available
Sampling frequency	600.61 Hz
Highpass	0.10 Hz
Lowpass	70.00 Hz
Filenames	sample_audvis_raw.fif
Duration	00:04:38 (HH:MM:SS)

EEG分析流程

Run Cell | Run Above | Debug Cell | Go to [23]

```
#%%
```

```
''' ICA '''
```

```
ica_raw = raw.copy()
```

```
ica = mne.preprocessing.ICA(n_components=20, random_state=97, max_iter=800)
```

```
ica.fit(ica_raw)
```

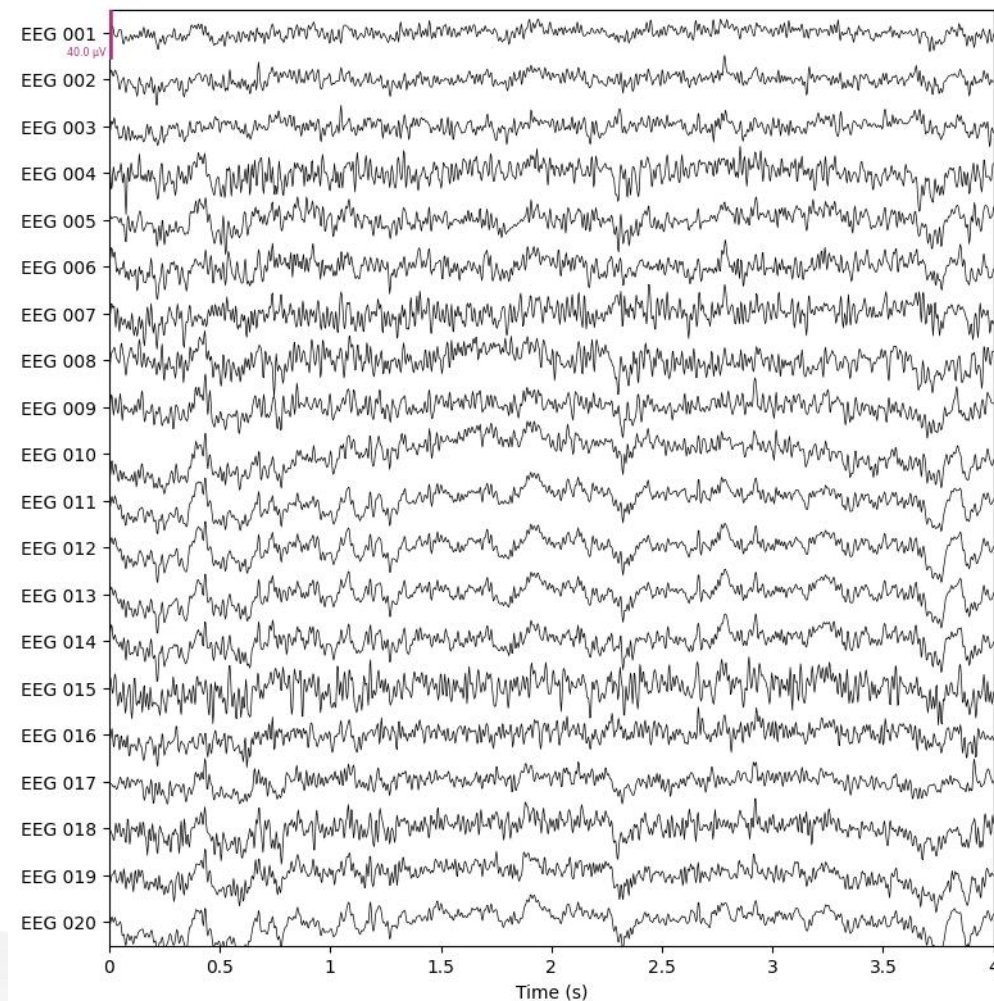
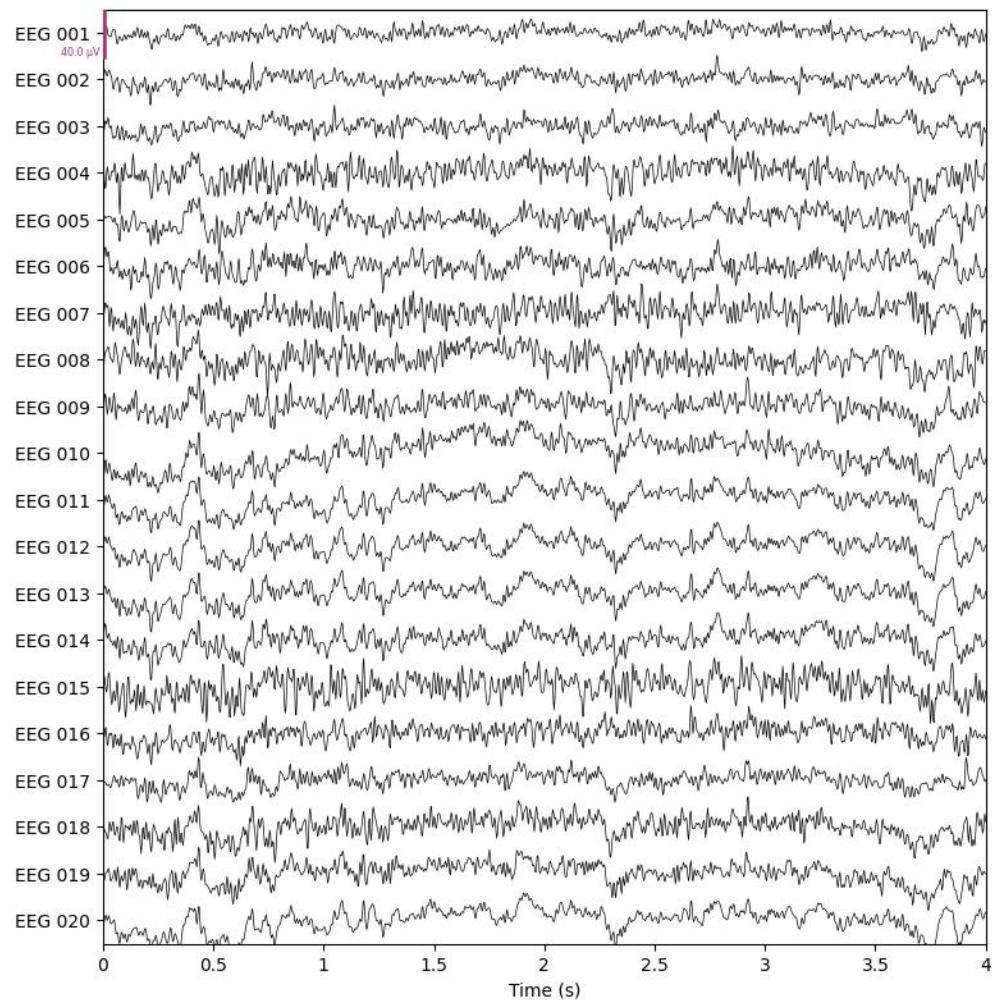
```
ica.exclude = [0, 18, 19]
```

```
# ica.plot_sources(ica_raw, show_scrollbars=False)
```

```
ica.apply(ica_raw)
```

```
ica_raw.plot(duration=4, show_scrollbars=False)
```


EEG分析流程



EEG分析流程

Run Cell | Run Above | Debug Cell | Go to [16]

```
#%%
```

```
''' power spectral density '''
```

```
raw.plot_psd(fmax=70)
```

```
#
```

```
raw.plot_psd(fmin=0, fmax=4)
```

```
raw.plot_psd(fmin=4, fmax=8)
```

```
raw.plot_psd(fmin=8, fmax=12)
```

```
raw.plot_psd(fmin=12, fmax=30)
```

```
raw.plot_psd(fmin=30, fmax=45)
```

EEG分析流程

EEG分析流程

EEG分析流程

Run Cell | Run Above | Debug Cell | Go to [29]

#%%

''' 等分数据, 转为epoch, 也可以直接把整个片段当成一个事件 '''

```
new_events = mne.make_fixed_length_events(raw, start=20, stop=70, duration=50.)
```

```
epochs = mne.Epochs(raw, new_events)
```

```
epochs = epochs.set_eeg_reference(projection=True)
```

```
print(epochs)
```

''' 频率分析 '''

```
epochs.plot_psd_topomap(normalize=False)
```

''' 时频分析 '''

''' 要研究frequencies的作用 '''

```
frequencies = np.arange(7, 30, 3)
```

```
power = mne.time_frequency.tfr_morlet(epochs, n_cycles=2, return_itc=False,
```

```
                                     freqs=frequencies, decim=3)
```

```
power.plot('EEG 035')
```

EEG分析流程

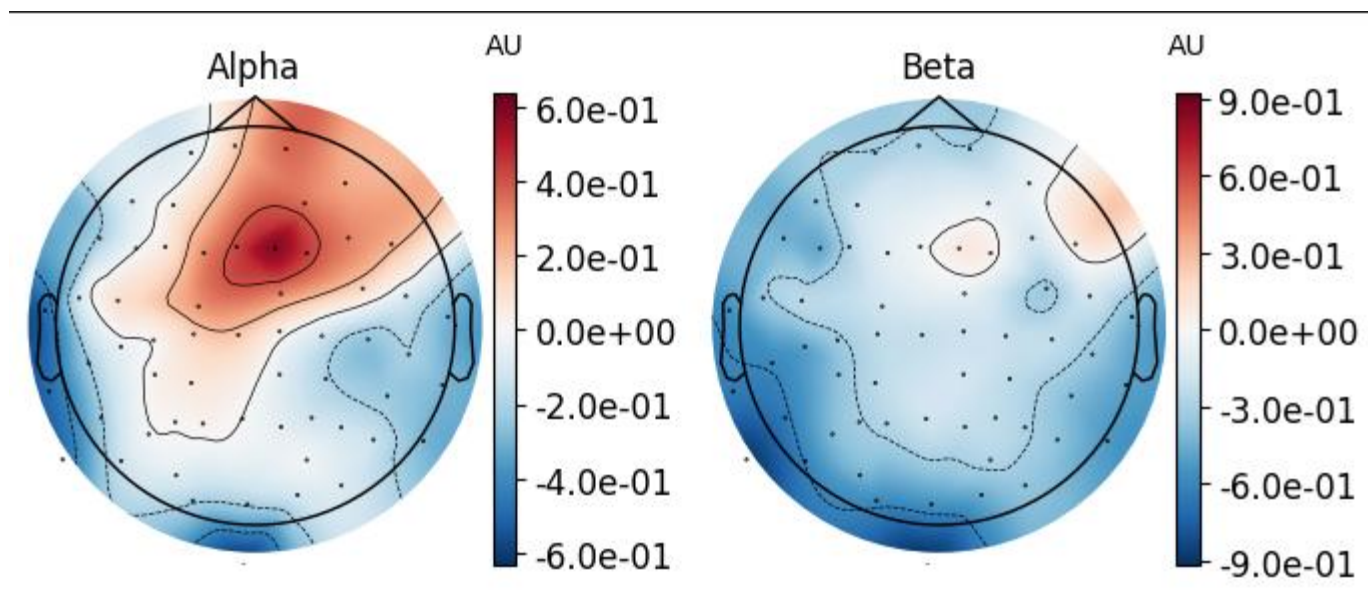
```
Run Cell | Run Above | Debug Cell | Go to [29]
#%%
''' Inspect power '''

power.plot_topo(baseline=(-0.5, 0), mode='logratio', title='Average power')
power.plot([35], baseline=(-0.5, 0), mode='logratio', title=power.ch_names[35])

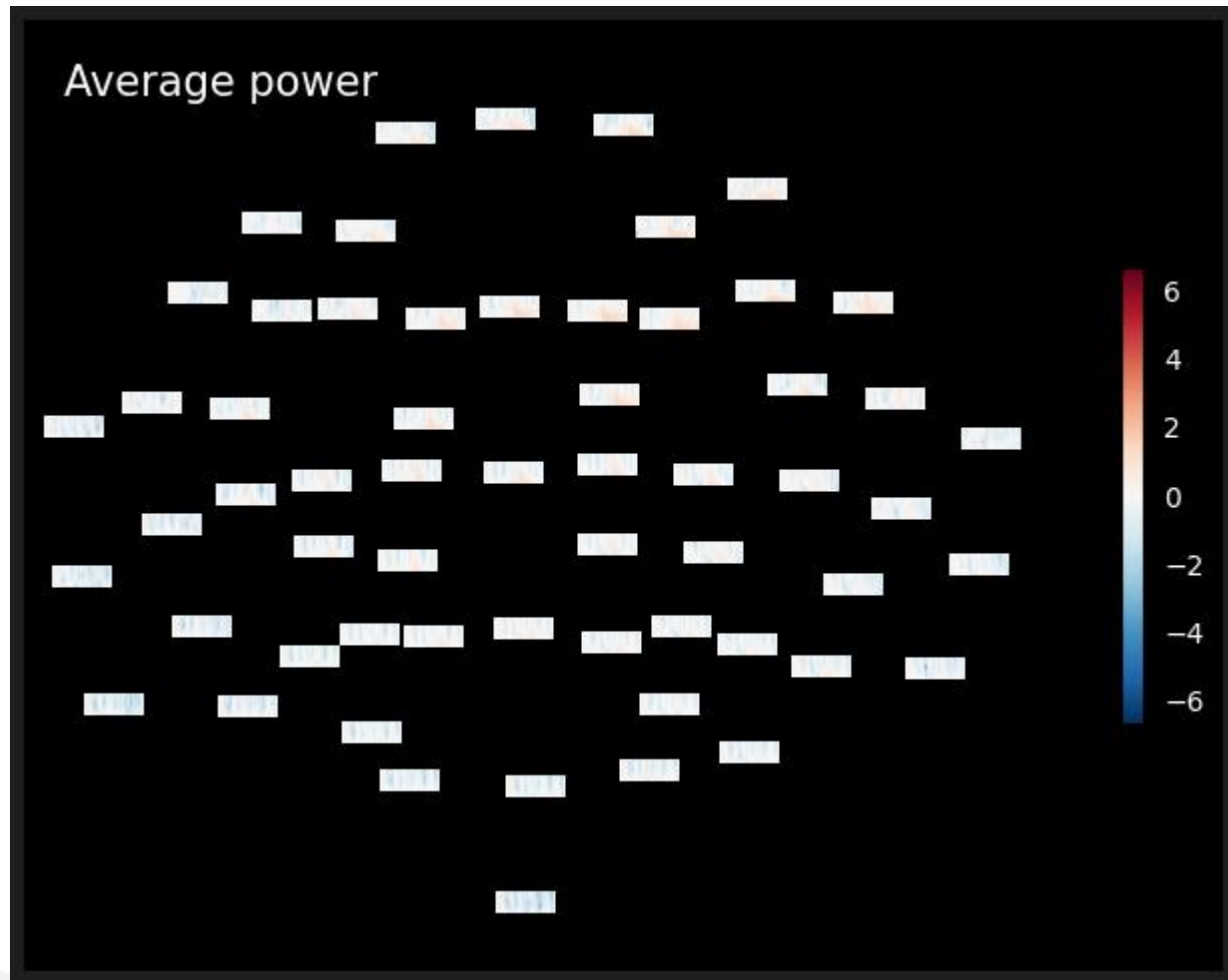
fig, axis = plt.subplots(1, 2, figsize=(7, 4))
power.plot_topomap(ch_type='eeg', fmin=8, fmax=12,
                  baseline=(-0.5, 0), mode='logratio', axes=axis[0],
                  title='Alpha', show=False)
power.plot_topomap(ch_type='eeg', fmin=12, fmax=30,
                  baseline=(-0.5, 0), mode='logratio', axes=axis[1],
                  title='Beta', show=False)

mne.viz.tight_layout()
plt.show()
```

EEG分析流程



EEG分析流程



EEG分析流程

