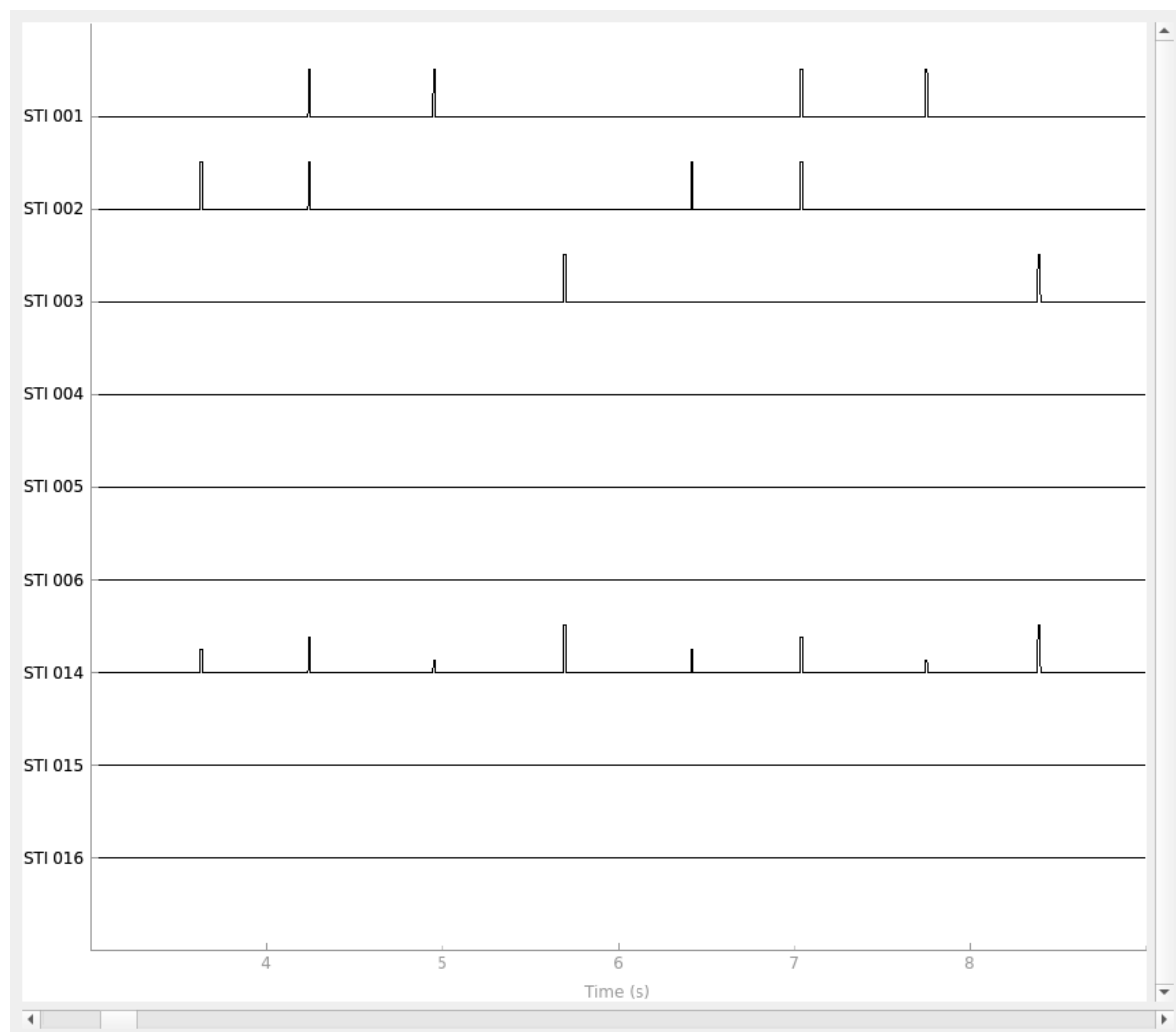


```
raw.copy().pick_types(meg=False, stim=True).plot(start=3, duration=6)
```

上面的代码为画出stim的电压幅度，直流脉冲可能全部在一个STIM通道上（在这种情况下，不同的实验事件或试验类型被编码为不同的电压幅度），或者它们可以分布在多个通道上，在这种情况下，脉冲发生的通道可用于编码不同的事件或条件。即使在具有多个STIM通道的系统上，通常也有一个通道记录其他STIM通道的加权和，这样该通道上的电压电平也可以明确地解码为特定的事件类型

求和通道一般都是在STI 014通道上



获取事件

如果数据在 STIM 通道上记录了事件，则可以使用 `find_events` 将其转换为事件数组。每个脉冲的起始（或偏移）的样本数记录为事件时间，脉冲幅度转换为整数，并将这些样本数对加上整数代码存储在 `NumPy arrays`（通常称为“事件数组”或简称为“事件”）

MNE-Python 事件实际上是三个值：在样本编号和整数事件代码之间是一个值，指示事件代码在紧接前一个样本上的内容。实际上，该值几乎总是 0，但它可以用来检测持续时间超过一个样本的事件的终点

所以event中第一个是样本的编号 第二个是指示事件代码在紧接前一个样本上的内容 第三个是事件编号 也就是我们要用的label

其他数据情况

一些EEG / MEG系统生成的文件将事件存储在单独的数据数组中，而不是作为一个或多个STIM通道上的脉冲。例如，EEGLAB 格式将事件存储为 .set 文件中的数组集合。读取这些文件时，MNE-Python 会自动将存储的事件转换为 Annotations对象，并将其存储为Raw对象的 annotations 属性

一、了解基本的数据类型

1.raw的访问格式

通常raw的数据访问方式如下：

```
data, times = raw[picks, time_slice]
```

picks:是根据条件挑选出来的索引，也就是选择通道 channels

time_slice:时间切片 也就是time

想要获取raw中所有数据，以下两种方式均可：

```
data, times = raw[:]
```

```
data, times = raw[:, :]
```

注意区分脑电图和脑磁图的图像区别

raw返回所选信道以及时间段内的数据和时间点，分别赋值给data以及times（即raw对象返回的是两个array）

2.Epochs对象的理解

Epochs对象是一种将连续数据表示为时间段集合的方法，

对象其存储在数组(n_events, n_channels, n_times)

epochs对象类似于mne.io.Raw对象，也具有info属性和event属性。

Epochs中有一个preload参数，将其设置为True进行预加载

可以通过下面两种方式来查看epoch内的event相关信息

把原始数据(raw)切分成若干个epoch(时间片段)的集合，

以每个提前标准好的event为参照，取tmin到tmax这一段时间作为一个epoch来进行数据处理。

取每个event的前0.1秒(tmin = -0.1)和后1秒(tmax = 1)共1.1秒的时间长度作为一个epoch,为后续同类型的event的epochs叠加(average)分析做准备

3.通过打印细节理解一下raw的数据结构

```
Opening raw data file C:\Users\zyb\mne_data\MNE-sample-data\MEG/sample/sample_audvis_raw.fif...
Read a total of 3 projection items:
    PCA-v1 (1 x 102) idle
    PCA-v2 (1 x 102) idle
    PCA-v3 (1 x 102) idle
Range : 25800 ... 192599 =      42.956 ...   320.670 secs
Ready.
Current compensation grade : 0
```

通过打印raw：

```
print(raw)
```

<Raw | sample_audvis_raw.fif, n_channels x n_times : 376 x 166800 (277.7 sec), ~3.6 MB, data not loaded>

可以看出核心数据为n_channels和n_times

二、进行最基本的事件选取

1.events_from_annotations获取事件信息

在该函数中传入参数raw将会返回两个参数，event和event_id分别对应着的是事件与事件的编号

event是一个列表共有三列，包含着三列信息[事件，（官方文档似乎没有找到该列是干什么的，不过该列都是0），事件所对应的id]

event_id是一个字典的数据类型，例如{'T0': 1, 'T1': 2}，是对事件进行的编号，T1，T2等编号意义可以在数据集说明文档中找到

2.pick_events挑选感兴趣的事件进行研究

传入事件和一个列表到该事件中，列表中的数据应该是事件所对应的id，该函数将会返回一个你所需要的Id的事件

3.concatenate_raws进行raw对象的拼接

将多个raw数据对象拼接在一起，合成一个更大的数据集

拼接的对象： `mne.concatenate_raws` 接受一个 `Raw` 对象的列表作为输入。这些 `Raw` 对象应该具有相同的通道名称和通道数，以及相同的采样频率。这是因为在拼接时，函数会假定这些属性是一致的。

时间轴的拼接： 拼接是在时间轴上完成的。对于多个 `Raw` 对象，它们在时间上是相邻的，函数会将它们按照它们的时间顺序拼接在一起，创建一个包含所有数据的新 `Raw` 对象。

`mne.concatenate_raws` 会将 `Raw` 对象列表中所有的事件拼接在一起。在拼接过程中，它会确保事件信息在时间轴上被正确调整，以匹配新的拼接后的 `Raw` 对象

处理信息： 拼接时，`mne.concatenate_raws` 会处理一些关键信息，如文件信息、事件信息等，以确保在拼接后的对象中保留一致性。

三、选取事件后创建一个Epochs对象

1.Epochs对象中参数的传递

传入raw，event，event_id等必须的参数，再去设置tmin和tmax两个时间区间作为一个Epochs对象的时间采样

```
Not setting metadata
15 matching events found
Setting baseline interval to [-1.0, 0.0] s
Applying baseline correction (mode: mean)
0 projection items activated
Using data from preloaded Raw for 15 events and 961 original time points ...
1 bad epochs dropped
```

如图显示有15个事件被检测到

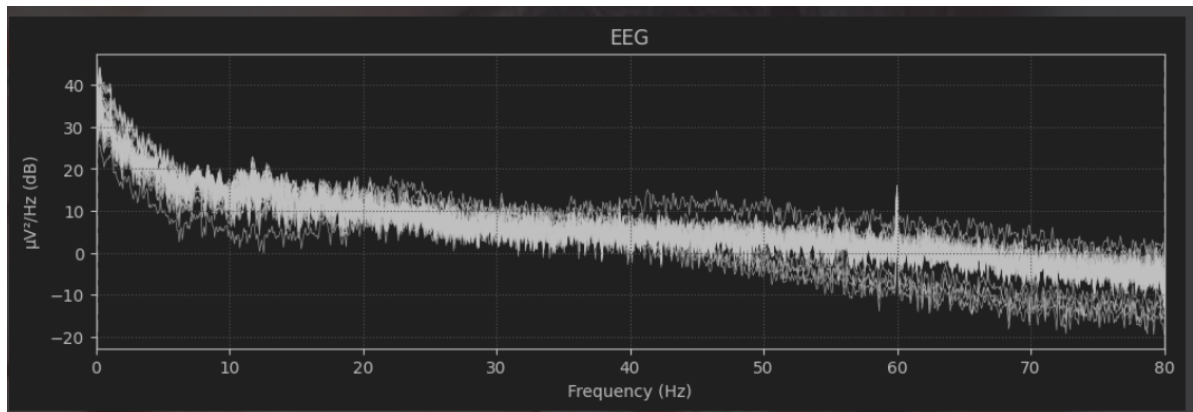
2.观察获取数据，并进行滤波处理

先用get_data()函数获取数据，观察数据。再利用load_data().filter(l_freq,h_freq)传入想要过滤掉的频段进行凹陷滤波mne中的也有特定的方法，raw.notch_filter(freqs=60)

四、去伪迹

1.观察频谱图

利用raw.compute_psd().plot()自动画出频谱图，并观察频谱特征。



应该注意的问题：

在观察运动想象在频谱图中的分布，我们该如何确定其在哪个范围当中呢？

1. **计算功率谱密度 (Power Spectral Density, PSD)：** 使用信号处理工具（如Fast Fourier Transform, FFT）计算运动想象信号的功率谱密度。功率谱密度表示在不同频率下信号的能量分布。
2. **绘制频谱图：** 将计算得到的功率谱密度数据绘制成频谱图。横轴表示频率，纵轴表示功率或能量。这样的图形可以帮助您可视化信号在频率上的分布情况。
3. **观察主要频率成分：** 查看频谱图并观察主要的功率峰值。这些峰值对应于信号的主要频率成分。确定主要峰值的位置，可以帮助您了解运动想象信号集中在哪个频率范围内。
4. **频带分析：** 根据观察到的主要频率峰值，可以进一步进行频带分析，确定一个特定的频率范围作为感兴趣的频带。
5. **考虑个体差异：** 注意不同个体的运动想象信号可能有所不同。因此，频谱分析应该基于您的研究对象和实验设计。

在进行运动想象时，频谱图的特征是什么呢？

1. **主要频率成分：** 运动想象通常会导致频谱图上的主要功率峰值。这些峰值对应于信号的主要频率成分，即在运动想象过程中信号最活跃的频率。观察这些峰值的位置和强度可以帮助确定运动想象信号的主要频率。
2. **频带活动：** 除了主要频率成分外，您还可以观察信号在不同频带上的活动。某些运动想象任务可能导致特定频带的增强或抑制。通过观察频带的活动，您可以更详细地了解信号的频率特性。
3. **频谱形状：** 运动想象信号的频谱形状也是重要的特征之一。某些任务可能导致频谱的扩展或收缩，反映了信号在频率上的分布模式。对于不同类型的运动想象，频谱形状可能会有所不同。
4. **频谱变化随时间：** 如果您的信号是时域信号，即随时间变化的信号，您可以观察频谱图在不同时间点的变化。这可以帮助您了解运动想象信号的时序动态。
5. **与基线比较：** 将运动想象时的频谱图与基线（非运动想象状态）进行比较，以便观察在运动想象期间是否存在显著的频谱变化。这可以帮助确定与运动想象相关的特定频率成分。

请注意，以上特征仅是一般性的观察点，实际情况可能会根据具体的实验设计和研究目标而有所不同。在进行运动想象研究时，通常需要使用专业的信号处理工具和方法来进一步分析和解释频谱图的特征。

上面文档当中解释了利用notch_filter凹陷滤波进行工频伪迹去除

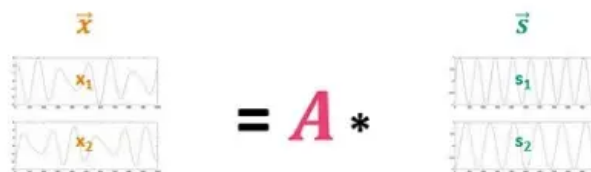
2.ICA

ICA的目的是分离混合数据来恢复原始的未知信号。最终目的是重建数据，使每个维度相互独立

General framework:

$$\vec{x} = A * \vec{s}$$

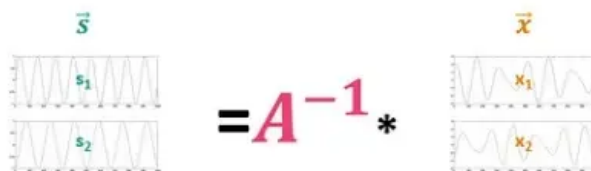
where a_{ij} are mixing coefficients



Goal:

$$\vec{s} = A^{-1} * \vec{x}$$

independent signals



知乎 @冷冻工厂

其中A是混合矩阵，s是矢量的信号，我们的目的是得到每个独立的信号

在mne包中，如果打算使用ICA，那么应该进行高通滤波。**高通滤波器 (High-pass filter)**：只允许某一频率以上的信号无衰减地通过滤波器，去掉了信号中低于该频率的不必要的成分或者说去掉了低于该频率的干扰信号。raw.filter(l_freq=1.0,h_freq=None)进行高通滤波