**Question 1: 資料描述**

* 檔案內 key：frames, delta\_t
* delta\_t（每個 frame 的時間步長）：0.0001220703125 s（≈ 122.07 µs）
* 取樣率：8192 Hz（= 1/delta\_t）
* frames 形狀與型別：(81920, ), float64

總長度 10秒（81920 × delta\_t），但根據作業的需求繪製區間：0–5 s 一張含有 文字, 鋼琴, 螢幕擷取畫面, 音樂 的圖片

AI 產生的內容可能不正確。觀察到一開始 0–0.6 s 有明顯觸發很多 spikes；約 2.7–3.8 s 活動稀疏；4.1–4.5 s 又有出現一波 spikes。

一張含有 螢幕擷取畫面, 圖表, 繪圖, 行 的圖片

AI 產生的內容可能不正確。

* 參數：**bin 寬 100 ms (= 0.1 s) （如老師投影片當中的設置）**
* 作法：每個 0.1 s 箱的 spike 數 ÷ 0.1 s → **Hz**
* 0–5 s 統計：
  + 此區間總 spike 數量：**155**
  + 區間平均發生率：**31.0 Hz**（= 155/5）
  + **最高 box** 發率：**160.0 Hz**，發生於 **t ≈ 0.35 s**（主要的中心）

一張含有 文字, 螢幕擷取畫面, 繪圖, 行 的圖片

AI 產生的內容可能不正確。

* 參數：**window寬 100 ms**；離散長度 **L = round(0.1/t) ；總計 819**（樣本）
* 作法：長度 819 的等權windows（和歸一化為 1）與 frames 卷積，結果再除以 t 就等於**Hz**
* 0–5 s 統計：
  + **最大的發出率**：**200.05 Hz**，**t ≈ 0.5228 s**
  + 區間平均發出率：**30.93 Hz**

一張含有 繪圖, 圖表, 行, 文字 的圖片

AI 產生的內容可能不正確。

參數：**σ = 100 ms (= 0.1 s)**；離散化時取 **±4σ**

0–5 s 統計：

* **最大發出率**：**125.06 Hz**，**t ≈ 0.4269 s**
* 區間平均發出率：**30.69 Hz**

最　ｓｍｏｏｔｈ、去噪音效果好，峰值較矩形窗低但保留了主結構。

一張含有 文字, 繪圖, 行, 圖表 的圖片

AI 產生的內容可能不正確。

* 參數：**α = 10 Hz**（時間常數 1/α = **0.1 s**）
* 視窗：（**t≥0**；和歸一化為 1）
* 卷積注意：numpy.convolve 會反轉 kernel；為保持「因果」（只用過去），**實作時對 kernel 再做一次反轉**（kernel\_alpha[::-1]）後再 convolve。  
  依投影片「convolve 會反轉」的說明而再反轉一次
* 0–5 s 統計：
  + **最大發出率**：**126.21 Hz**，**t ≈ 0.4764 s**
  + 區間平均發出率：**30.51 Hz**

符合即時/因果性；曲線在峰值處相較exercise 5略有**延後**（相位滯後）。

＞　但是 exercise 4如果把採樣率設很窄、很密集，最終就會趨近於 Exercise 5的線（就好像音樂採樣變成數位光碟、ｍｐ３）

共同參數均為 **100 ms** 規模（或 **α=10 Hz**），與 Lecture 2 投影片完全對齊。

三種平滑（4–6）都抓到 0–0.6 s 與 ~4.2–4.5 s 的高活動區；差別在平滑強度與和時間的對齊：

* **矩形**最敏感、峰值最高；
* **高斯**最平順；
* **因果**最貼近線上處理但峰值有些延後。

0–5 s 的整體平均發出率約 **31 Hz**（各法的區間平均都趨近這個值）。