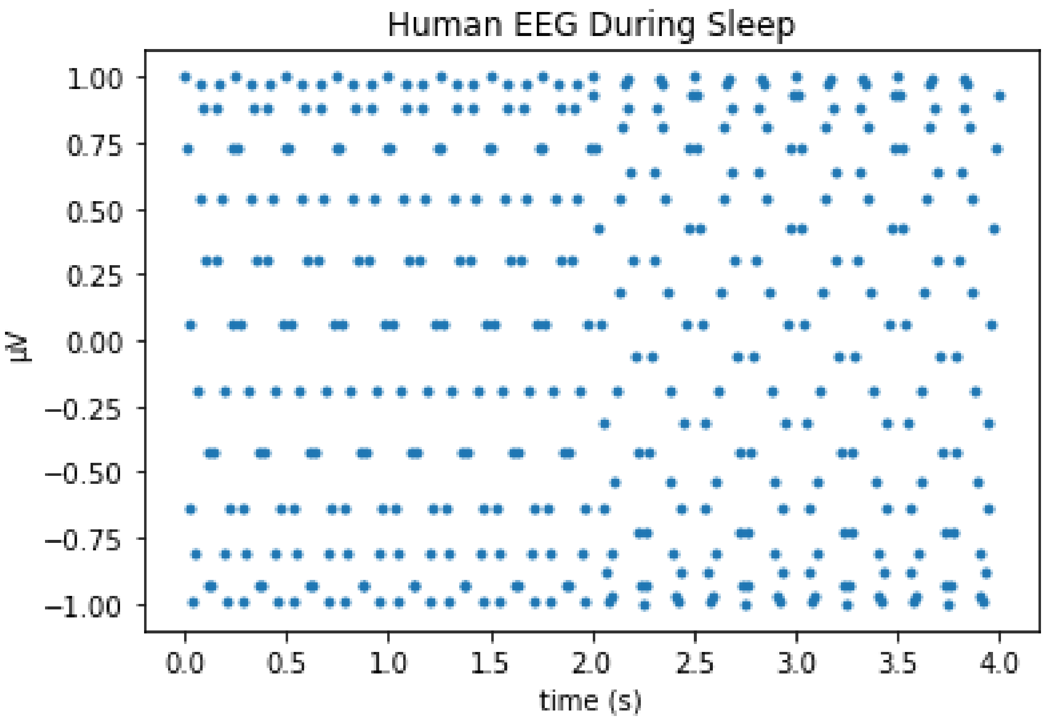
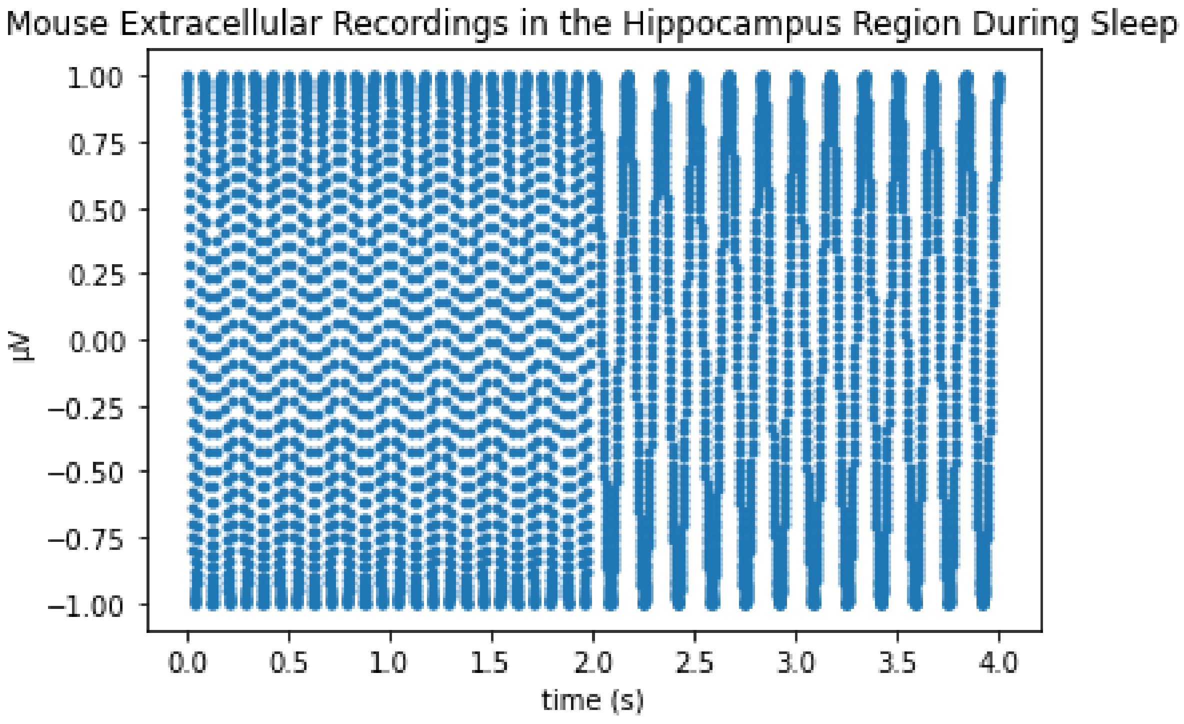
204938203 בנימין מנשה
Signal Processing 2022

Assignment 1

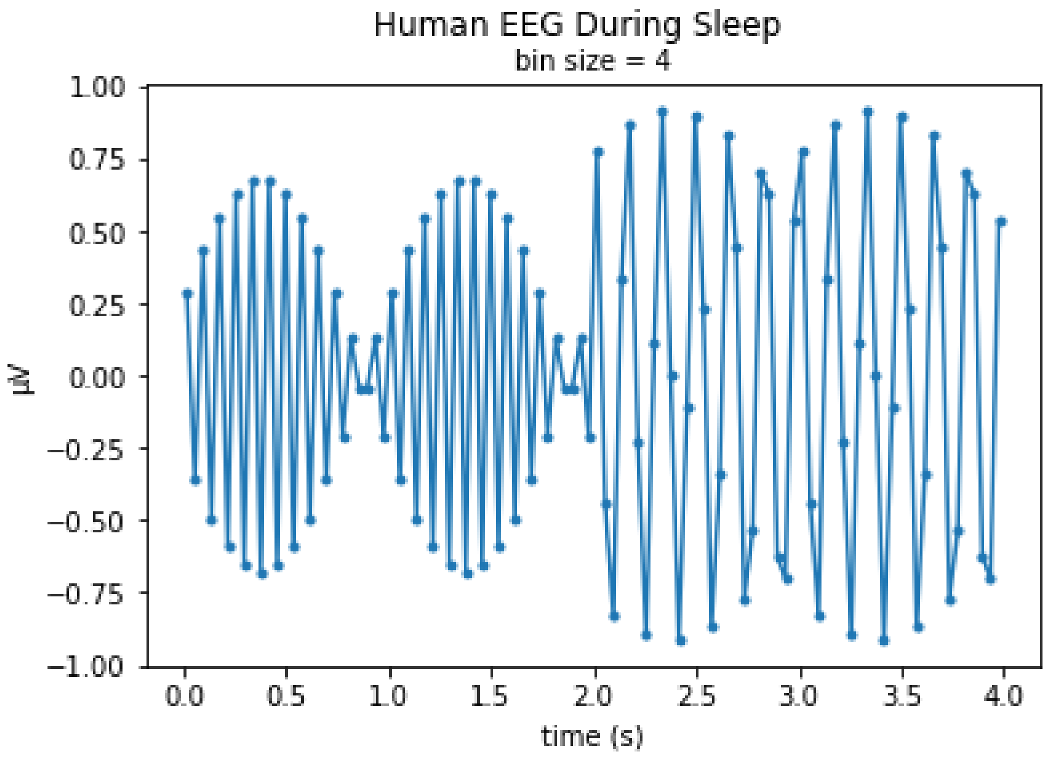
1) **Sampling and quantification**

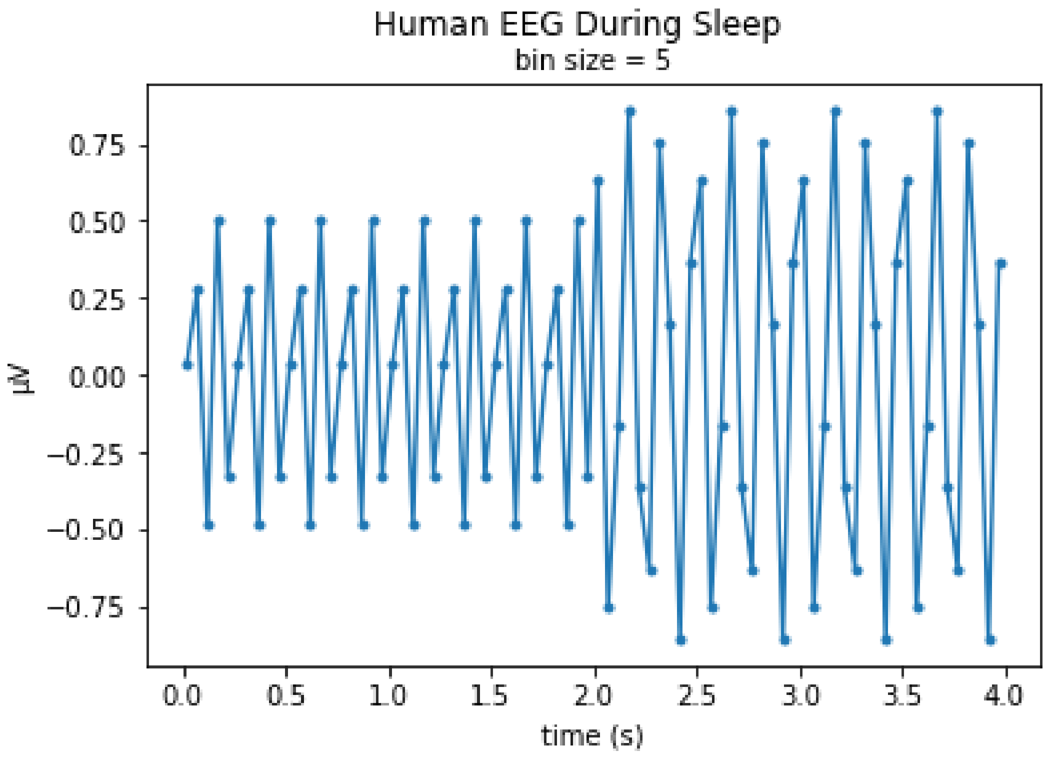
   a. I made separate scatter plots for each method. I assumed the amplitude is between -1 and 1 microvolt (otherwise just scale everything by the appropriate amplitude value, only the y-axis will change).





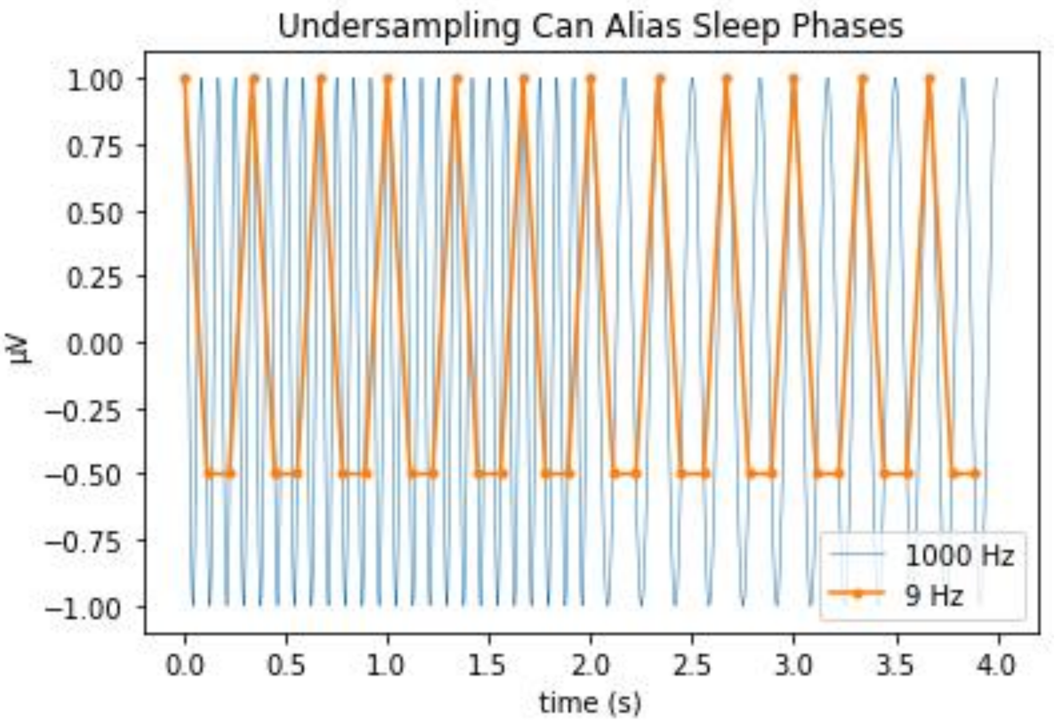
b. I made separate plots for each bin size. I plotted one marker per bin, in the center of the time-window. I then added interpolation lines.





It is easy to see that using a bin size of 4 samples preserves the wakefulness frequency (12 Hz) and phase 1 frequency (6 Hz), just by looking at the figures and counting the peaks in each stage. A more analytic solution is to show that, by Nyquist-Shannon's theorem, a periodic signal must be sampled at more than twice its (highest) frequency to avoid aliasing. For a signal of 12 Hz, a sampling frequency of 24 Hz is necessary. Since the original sampling frequency was 100 Hz, averaging each 4 samples reduces it to 100/4 = 25 Hz (higher than the theorem requires, thus preserving the signal), while averaging each 5 samples reduces it to 100/5 = 20 Hz, which is insufficient and the signal is lost.

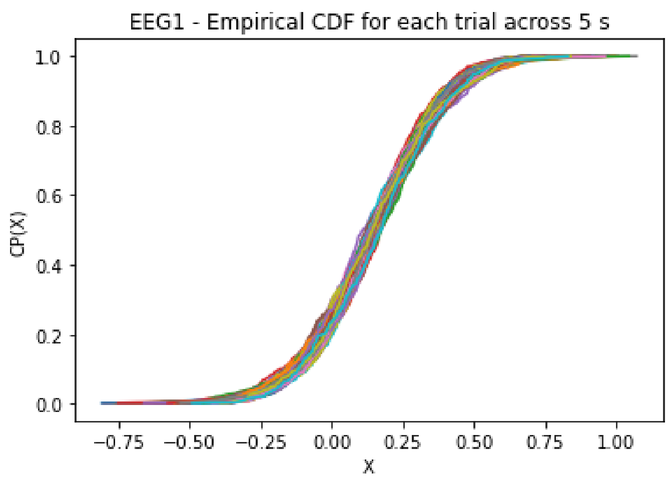
c. Ignoring phase shifts, any sampling frequency Fs with the property of cos(2pi*F1*(n/Fs)) = cos(2pi*F2*(n/Fs)) (for all integers n) will cause aliasing of the F1 and F2 frequencies such that they are indistinguishable. In our case, I used 9 Hz, because cos(2pi*12*(n/9)) = cos(2pi*6*(n/9)) for all integers n. Obviously there are many more possible sampling frequencies which could cause this aliasing, I only illustrate one such example:

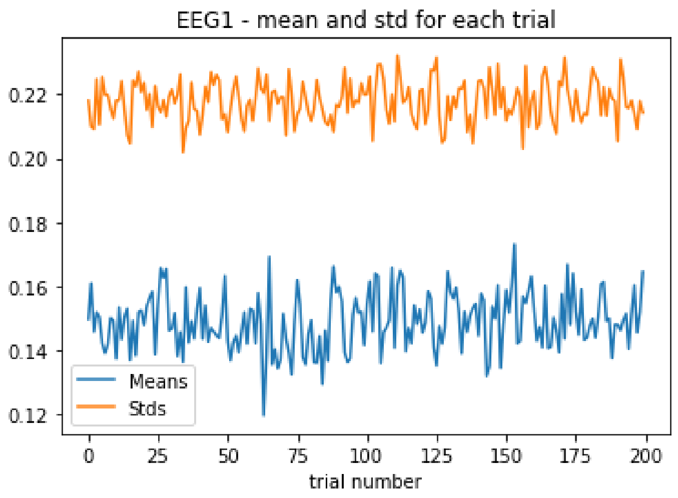


2) Stochastic process

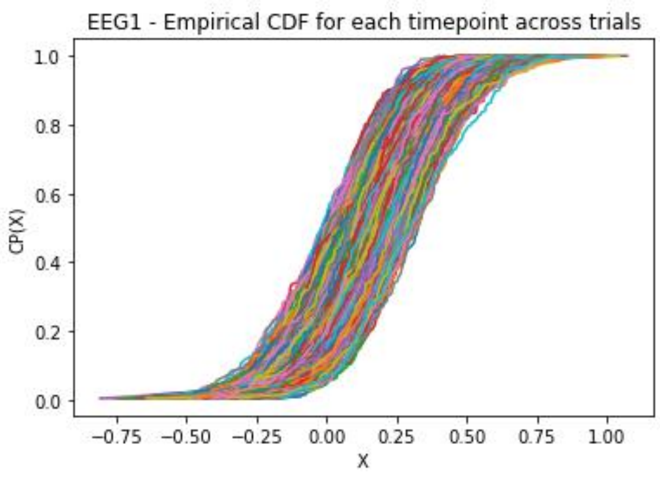
a. Wrote the function in python. To determine stationarity, I checked if the means and standards deviations of each 5 second trial are consistent and have low variance between the trials (their standard deviation must not exceed 0.05). To determine ergodicity, I checked if the means and standard deviations of each timepoint are consistent and have low variance across time (their standard deviation must not exceed 0.05). Obviously, for ergodicity it is most important that the mean and standard deviation of each timepoint across all trials equal the mean and standard deviation for a 5 second trial across all time, but I think that simply checking the variance between timepoints is essentially checking the same thing, since presumably after verifying stationarity the timepoints cannot all have a similar mean and standard deviation that is different than the trial mean and standard deviation across time.

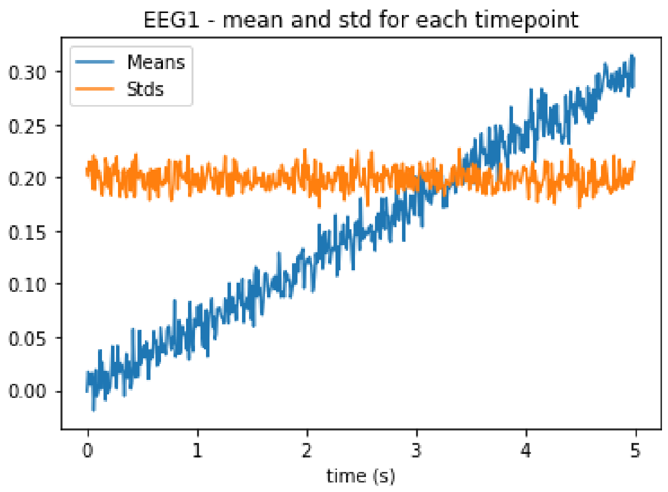
b. <u>EEG1: Stationary and Non-Ergodic</u>
The empirical cumulative distributions of each 5 s trial are similar, and the means and standard deviations seem to be more or less similar for each trial:



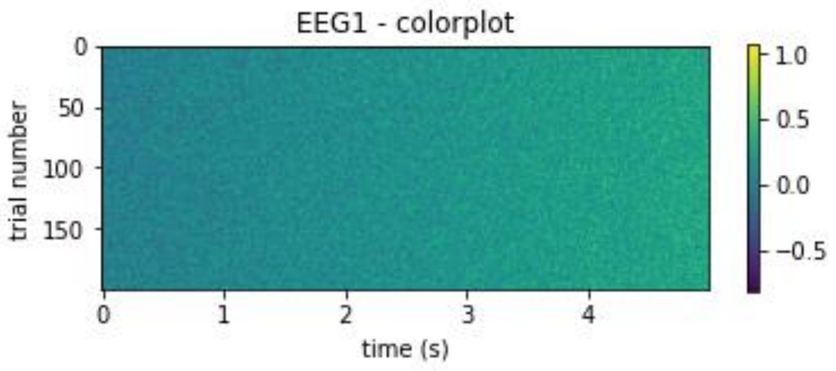


The empirical cumulative distribution of each time point are different, and the mean of each time point seems to be increasing:



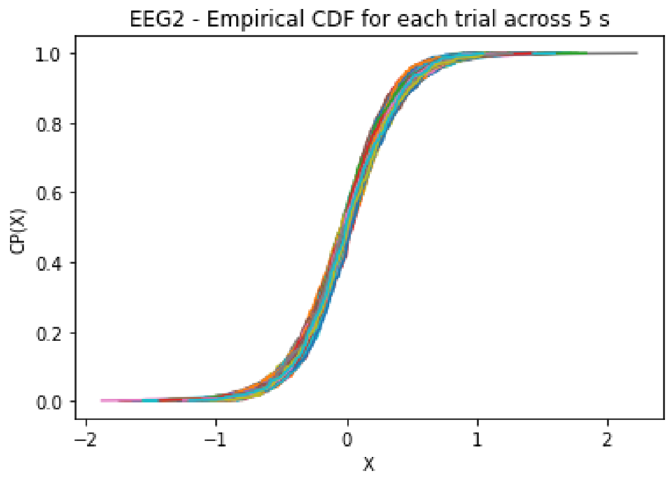


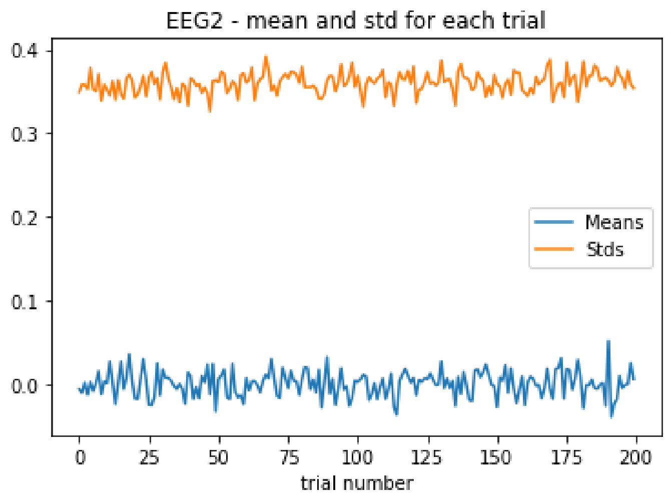
This can also be clearly seen if we color-plot all the trials:



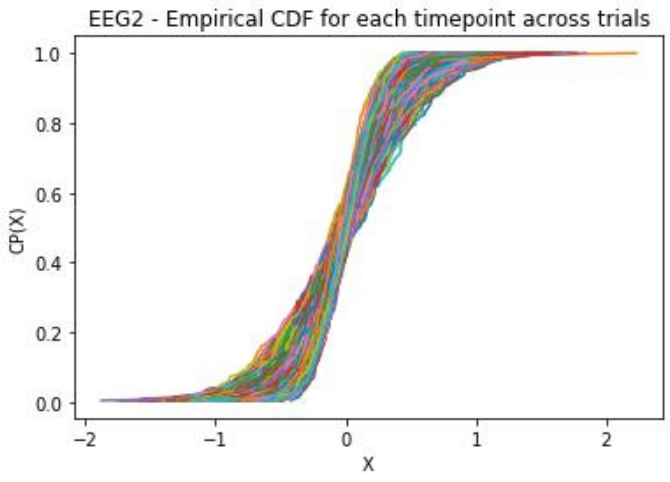
EEG2 – Stationary and Non-Ergodic

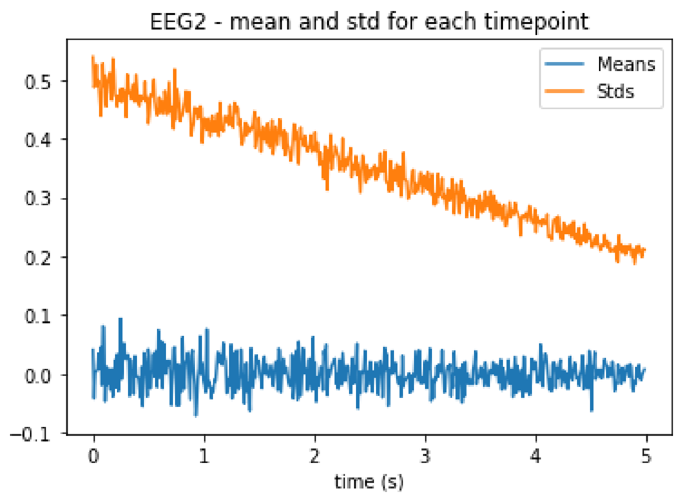
The empirical cumulative distributions of each 5 s trial are similar, and the means and standard deviations seem to be more or less similar for each trial:



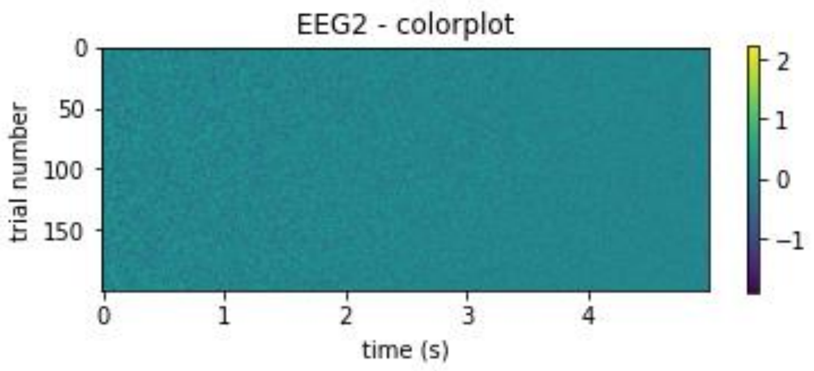


The empirical cumulative distribution of each time point are different, and the standard deviation of each time point seems to be decreasing:



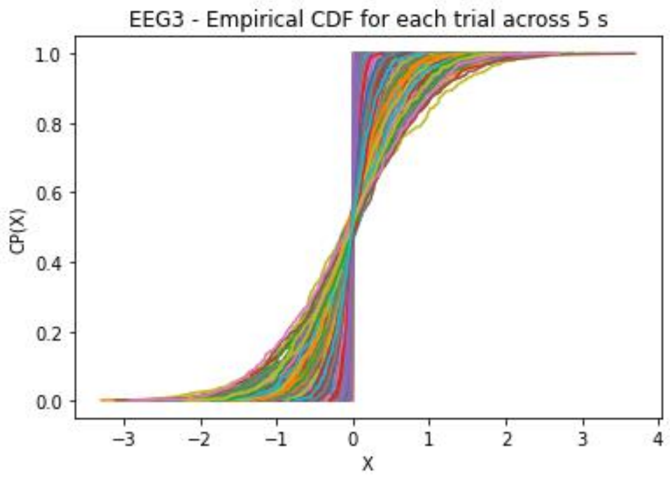


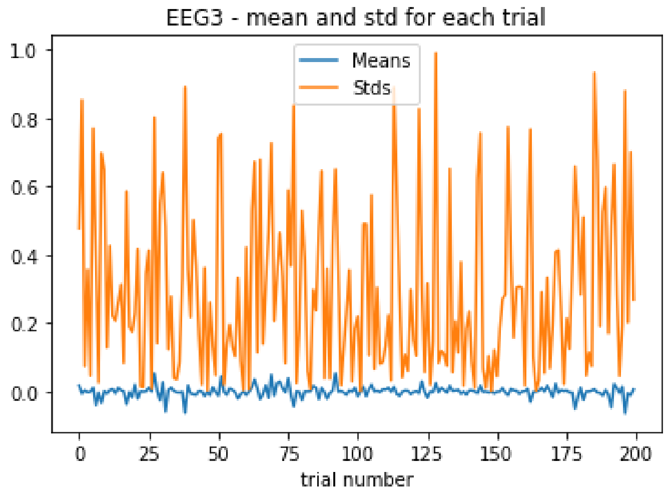
This can also be seen (kind of) if we color-plot all the trials. Notice the later timepoints are much less varied:



<u>EEG3 – Non-Stationary (and Non-Ergodic)</u>

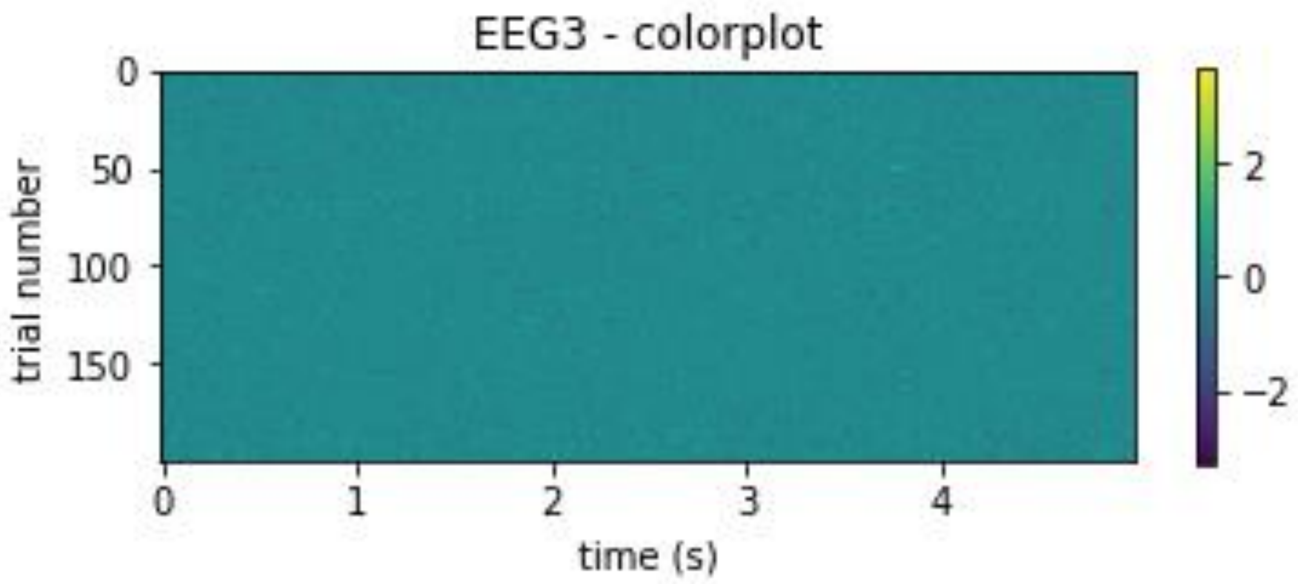
The empirical cumulative distributions of each 5 s trial are different, and while the means of each trial are more or less similar, the standard deviations vary wildly:





Since ergodicity demands stationarity, I did not continue checking.

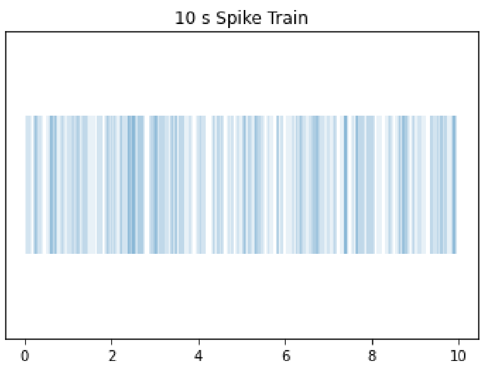
If we squint our eyes, this can also be seen in the color-plot, as some trials (horizontal lines) can be distinguished from others (for example, trial number 50):

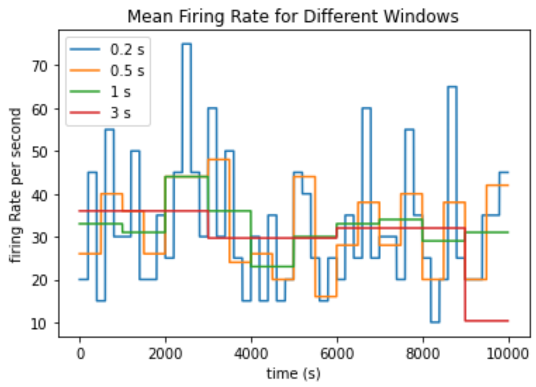


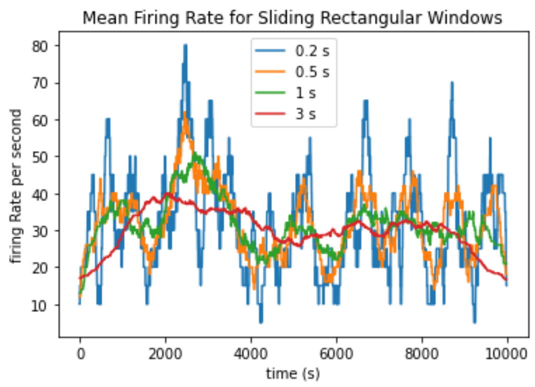
3) Firing rates and convolution

After binning the spikes into 1ms bins, I calculated the average firing rate to be **32.4 spikes per second**. Interestingly, the raw data file contained 327 spikes, which means that we binned more than 1 spike into a single bin exactly 3 times (and thus lost data by binning).

More importantly, I now wanted to see if the firing rate is stable across the whole 10 seconds, so I first used non-overlapping windows, of sizes 0.2s, 0.5s, 1s, and 3s. It is obvious to see that the larger

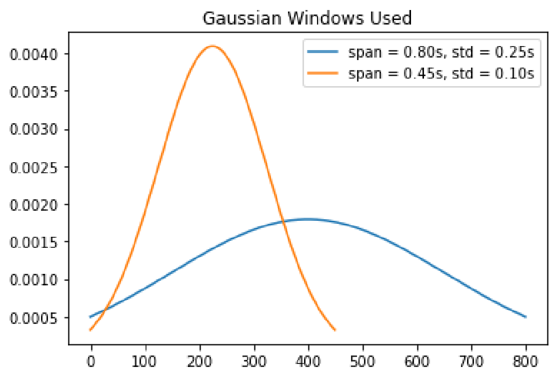


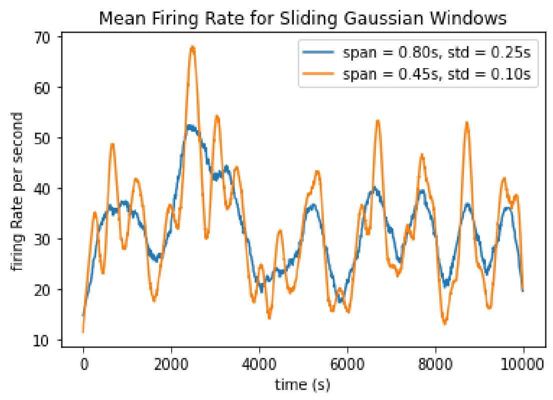




the window the smoother the firing rate, since we are averaging over many more values, and thus losing variance. However, the choice of window boundaries is arbitrary, and so we get unnatural "steps" in the data (middle figure). I decided instead to use sliding rectangular windows (using my convolution function) to get a smoother moving average firing rate. I think it can be seen that once again, larger windows "smooth out" more of the variance and paint a more stable picture. Altogether, the sliding windows appear very similar to the non-overlapping windows, except smoother.

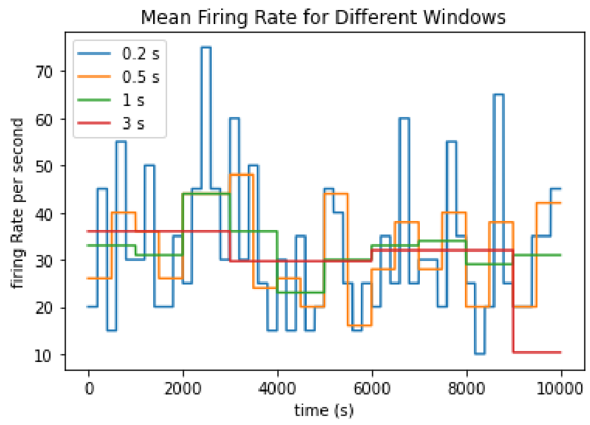
Afterwards, I used 2 Gaussian windows, which also "smooth" the average firing rate for each timepoint, except they also weigh the spikes differently based on their distance from the current timepoint. Once again, the larger window produced a less variant firing rate at each timepoint. Additionally, it seems to me that the firing rate averages are even more "smooth", meaning less jagged. This is probably due to the gradient weighting for each spike, where its contribution to the average decreases gradually (and not abruptly) for further timepoints.



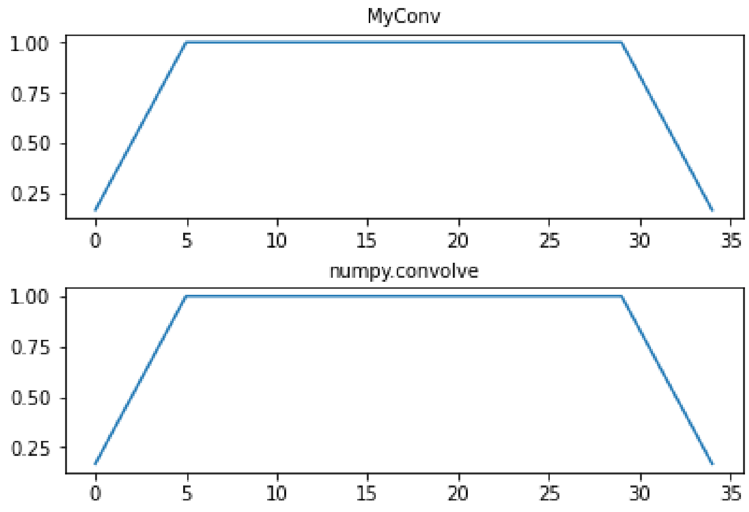


a. Average firing rate = **324 / 10 = 32.4 spikes per s**

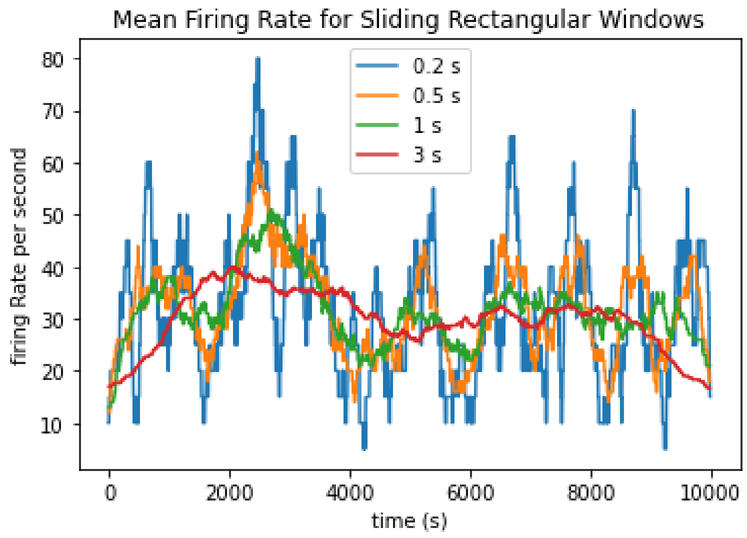
b. Non-overlapping window average:



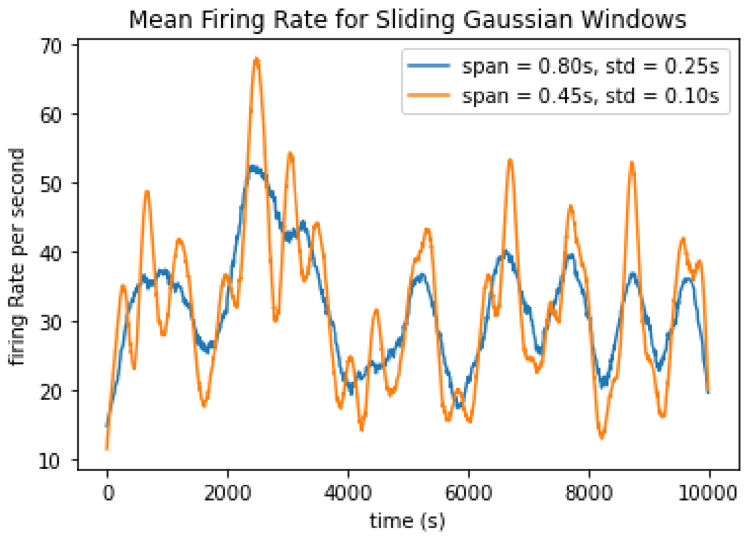
c. My function and numpy function came out identical (even at the edges):



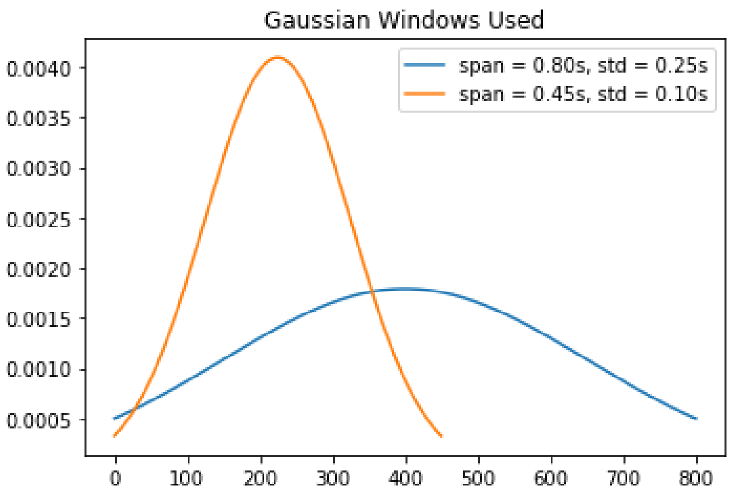
d. Sliding window average:



e. Sliding Gaussian window average:



windows:


Gaussian Windows Used
span = 0.80s, std = 0.25s
span = 0.45s, std = 0.10s