Homework for the coherence

1. Generate two sets of simulated data (x and y) consisting of a sinusoid oscillating at frequency *f* plus additive Gaussian noise, as follows:
    - For each data set, generate 100 trials of 1 s data sampled at 500 Hz.
    - For the first dataset *x*, for each trial:
        - Simulate a sinusoid at frequency *f* (you choose the frequency) and amplitude 1.
        - Set the **phase** of the sinusoid to a random value between 0 and 2π.
        - Add to the sinusoid Gaussian noise with mean 0 and standard deviation 1.
    - For the second dataset (*y*), for each trial:
        - Simulate a sinusoid with the same frequency *f* (you chose it for the signal *x*) and amplitude 1.
        - **Set the phase of the sinusoid to 0.**
        - Add to the sinusoid Gaussian noise with mean 0 and standard deviation 1.

    For these simulated data, please answer the following questions:

    a. What is the sampling interval (Δ)? What is the total duration of the recording (T)? What is the frequency resolution (df)? What is the Nyquist frequency (f_{NQ})?
    b. **Visualize the data** for x and y. What rhythms do you observe?
    c. **Plot the trial-averaged spectrum** versus frequency for x and y. Are the dominant rhythms in the spectrum consistent with simulation choices and your visual inspection of the data?
    d. **Plot the trial-averaged cross-covariance** between the two datasets. What features do you observe?
    e. **Plot the coherence** between the two datasets. At what frequencies, if any, is the coherence large?
    f. **Summarize (in a few sentences)** the results of your data analysis. Did you expect coherence for these data (i.e., is there a constant phase relationship between x and y at some frequency across trials)? Did your results match your expectation?

2. Consider the dataset **ECoG-1.mat** available in the GitHub repository. Please load these data into Python. Upon doing so, you will find three variables in your workspace. The variables *E1* and *E2* correspond to two simultaneous recordings of brain voltage activity from two electrodes. Both variables are organized so that the rows correspond to trials, and the columns to time. You should find 100 trials, with 500 time points per trial. The variable *t* corresponds to the time axis for these data, in units of seconds. Please use these data to answer the following questions.

    a. What is the sampling interval (Δ)? What is the total duration of the recording (T)? What is the frequency resolution (df)? What is the Nyquist frequency (f_{NQ})?
    b. **Visualize the data** from each electrode. What rhythms do you observe?
    c. **Plot the trial-averaged spectrum** versus frequency for each electrode. Are the dominant rhythms in the spectrum consistent with your visual inspection of the data?
    d. **Plot the trial-averaged cross-covariance** between the two datasets. What features do you observe?
    e. **Plot the coherence** between the two datasets. At what rhythms, if any, is the coherence large?
    f. **Summarize** (in a few sentences) the results of your data analysis. What are the important features of these data you would communicate to a colleague?

3. In this question, we consider a simple example that illustrates the fundamental features of spike-field coherence. Let's consider the case in which the field is a sinusoid plus Gaussian noise, and the spike train is random; the probability of a spike at any time is not related to previous spiking behavior. In this case, we also assume no relation between the field and point process. Therefore, we expect to find no spike-field coherence. Let's simulate some synthetic data, compute the spike-field coherence, and see what we find.

   As a first step, create 100 trials of data, each trial of 1 s duration with a sampling rate of 1000 Hz. We define these parameters as follows:

```
K = 100;      #Define no. of trials.
N = 1000;     #Define no. of samples per trial.
dt = 0.001;   #Define sampling interval.
```

   Now, let's define the synthetic data. We create in each trial a field, which here will be a sinusoid; and a spike train, which here will be drawn from a Bernoulli distribution with a probability *p* of a spike in each sampling interval.

```
y = np.zeros([K,N]); #Matrix to hold field data.
n = np.zeros([K,N]); #Matrix to hold spike data.

for k in np.arange(K):        #For each trial ...
      y[k,:] = np.sin(2.0*np.pi*np.arange(N)*dt * 10)+0.1*np.random.randn(1,N);
      n[k,:] = np.random.binomial(1,0.01,N)
```

   In this code, the frequency of the sinusoid is set to 10 Hz, and the probability of a spike in each sampling interval is set to p=0.01. These choices are reasonable yet arbitrary. With these synthetic multiscale data defined, repeat the analysis you performed on the data above. In particular,

   a. What is the sampling interval (Δ)? What is the total duration of the recording (T)? What is the frequency resolution (df)? What is the Nyquist frequency (f_{NQ})?
   b. **Visualize the data.** What rhythms do you observe? Do you see associations between the LFP and spikes?
   c. **Plot the spectrum** versus frequency for these data. Are the dominant rhythms in the spectrum consistent with your knowledge of the data?
   d. Compute and display the **spike-field coherence**. Do your results match your expectations?
   e. **Describe** (in a few sentences) your results, as you would to a colleague or collaborator?

4. Let's now consider a simulation in which we expect a nonzero spike-field coherence. To produce create the interaction, we must introduce a relation between the spikes and the field. We do so by making the probability of a spike in each time interval a function of the field. Let's fix the number of trials (K=100), the number of samples per trial (N=1000), and the sampling interval (dt=0.001), the same values used in the first simulation example. Then let's define the spike and field data for each trial.

```
f = 0.01; #Parameter for scaling of rate.
b = 1; #Parameter for background spiking.
y = np.zeros([K,N]); #Matrix to hold field data.
n = np.zeros([K,N]); #Matrix to hold spike data.
for k in np.arange(K):       #For each trial ...
        # ...define the LFP as a 10 Hz sinusoid + noise.
       y[k,:] = np.sin(2.0*np.pi*np.arange(N)*dt * 10)+0.1*np.random.randn(1,N);
       # ...draw spikes from a Bernoulli distribution,
       p = f*(b+np.exp(y[k,:]));         #...with probability dependent on LFP
       n[k,:] = np.random.binomial(1,p,N);
```

Note that the probability *p* of a spike in each time interval depends on three factors: (1) an overall scaling term *f*, (2) a baseline level of probability *b*, and (3) the exponentiated field *exp(y)*. We exponentiate the field *y* so that the probability is always positive. In this way, the probability of a spike depends on the field.

With these synthetic data defined, repeat the analysis you performed on the data in the previous example. In particular,

a. What is the sampling interval (Δ)? What is the total duration of the recording (T)? What is the frequency resolution (df)? What is the Nyquist frequency (f_{NQ})?
b. **Visualize the data.** What rhythms do you observe? Do you detect associations between the LFP and spikes?
c. **Plot the spectrum** versus frequency for these data. Are the dominant rhythms in the spectrum consistent with your knowledge of the data?
d. Compute and display the **spike-field coherence**. Do your results match your expectations?
e. **Describe** (in a few sentences) your results, as you would to a colleague or collaborator?

5. Load the file **spikes-LFP-2.mat**, available at the GitHub repository into Python. You will find three variables. The variable *y* corresponds to the LFP data, in units of millivolts. The variable *n* corresponds to simultaneously recorded binary spiking events. The variable *t* corresponds to the time axis, in units of seconds. Both *y* and *n* are matrices, in which each row indicates a separate trial, and each column indicates a point in time. Use these data to answer the following questions.
    a. What is the sampling interval (Δ)? What is the total duration of the recording (T)? What is the frequency resolution (df)? What is the Nyquist frequency (f_{NQ})?
    b. **Visualize the data.** What rhythms do you observe? Do you detect associations between the LFP and spikes?
    c. **Plot the spectrum** versus frequency for these data. Are the dominant rhythms in the spectrum consistent with your visual inspection of the data?
    d. Compute and display the **spike-field coherence**. Do you find evidence for spike-field coherence?
    e. **Describe** (in a few sentences) your results, as you would to a colleague or collaborator?