Spike Train Analysis and Modeling

Christophe Pouzat
# Contents

1. **Introduction** 4
   1.1. What are spike trains? 4
      1.1.1. Example of data origin 4
      1.1.2. Examples of spike trains 5
   1.2. Why should we bother with spike trains? 7
   1.3. The 'messy' aspect of spike trains 8

2. **Why a modeling approach with a strong stochastic element?** 9
   2.1. Membrane noise can lead to output fluctuations 9
   2.2. Synaptic noise is ubiquitous 10
      2.2.1. Fluctuations at the neuromuscular junction 10
      2.2.2. Quantal neurotransmitter release 11
      2.2.3. What about the central nervous system? 13
      2.2.4. Sources of quantal size fluctuations 15
   2.3. Other sources of variability 15
   2.4. Conclusions 16
   2.5. Exercises 16
      2.5.1. The inter mepp data of Fatt and Katz (1952) 16

3. **Let Us Now Praise Famous Men** 19
   3.1. Hagiwara (1954) 19
      3.1.1. Overview 19
      3.1.2. The model 20
      3.1.3. Getting the inter spike interval density 21
      3.1.4. Parameter estimation 22
      3.1.5. Comments 22
   3.2. Fitzhugh (1958) 23
      3.2.1. Background 23
      3.2.2. Building a discharge model for the transient regime 25
      3.2.3. Inference 26
      3.2.4. Comments 26
   3.3. Gerstein & Mandelbrot (1964) 27
      3.3.1. Background 27
      3.3.2. Model 27
      3.3.3. Time dependent drift 29
      3.3.4. Comments 30
1. Introduction

1.1. What are spike trains?

1.1.1. Example of data origin

Although it’s far from being the most common data source, insects like the locust, \textit{Schistocerca americana}, constitute a very attractive preparation. When we ‘carefully’ insert electrodes into a neuronal tissue containing not too densely packed neurons that fire fast–sodium dependent–action potentials or \textit{spikes}, our electrodes can catch the neuronal output as illustrated here with data collected from the first olfactory relay, the \textit{antennal lobe} of a locust:

![Image of locust and recording setup]

Figure 1.1.: Left: opened head and brain; center: \textit{antennal lobe} and recording probe (the two shanks with 8 bright spots on each); right: 1 sec recorded from a single \textit{tetrode}, a group of 4 closely spaced recording sites; the data were filtered between 300 Hz and 15 kHz (see [46] for details).
In the locust the antennal lobe, the only neurons that fire fast spikes are the projection neurons (equivalent of the mitral cells of the vertebrate olfactory bulb), the other neuronal type, the local neuron fires only local and 'slow' spikes (probably) calcium mediated [35]. With extracellular recordings we record only from projection neurons. This situation is locust specific, in other insects like the cockroach, Periplaneta americana, some local neurons also fire fast spikes [31].

1.1.2. Examples of spike trains

Spontaneous activity: After a ‘rather heavy’ and not neutral pre-processing stage called spike sorting [44, 36], the raster plot, one way of representing spike trains, can be built:

![Raster plot example](image)

Figure 1.2.: Example of a raster plot representing a portion of the spike trains of three simultaneously recorded neurons in the spontaneous regime (locust, Schistocerca americana, antennal lobe).

This spontaneous activity is driven by a spontaneous activity of the olfactory receptor neurons located on the antenna. If we cut the antennal nerve—the bundle made of the receptors’ axons—the spontaneous activity in the antennal lobe disappears almost entirely.

Odor responses: Since the brain region that is here recorded is the first olfactory relay, we usually look not only at the 'spontaneous' activity but also at stimulus responses (here, odor responses):
Figure 1.3.: Raster plots from one of the neurons showing the responses to 25 presentations (gray background) of cis-3-hexen-1-ol (a typical ‘green’ odor; that’s what you can smell when grass has just been cut).

Spike train representation by raster plots is omnipresent in neurophysiology even though it quickly becomes hard to read—when spikes are too close or when too much time is displayed—and there are better ways to represent such data. In fact, the first figure of the 'classical book' of Cox and Lewis *The statistical analysis of series of events* [16] displays the sample path of the counting process (or the observed counting process) instead. Formally, a stochastic process \( \{ N(t), t \geq 0 \} \) is said to be a counting process if \( N(t) \) represents the total number of 'events' that have occurred up to time \( t \) (see Section 4.4.1 for further discussion).

The sample-path-of-a-counting-process approach gives for the above data:
Figure 1.4.: The sample paths of the counting processes associated with the 25 spike trains just shown. In red, the empirical mean of the 25.

Each sample path (black lines above) is a step function that jumps by 1 every time a spike is observed. The discharge acceleration can be seen on the individual sample paths (the upward slope gets much larger shortly after the beginning of stimulus application–gray region–) and a pause or deceleration is also there on almost all the sample paths after the stimulation (flat region). These features are all clearer on the ‘mean trace’ (red) that is not, strictly speaking, a sample path (it jumps by $1/25$ every time an event is observed).

1.2. Why should we bother with spike trains?

- A key working hypothesis in Neurosciences states that the spikes’ occurrence times [2], as opposed to their waveform [1, 2] are the only information carriers between brain regions.

- This hypothesis legitimates and leads to the study of spike trains per se.

- It also encourages the development of models whose goal is to predict the probability of occurrence of a spike at a given time, without necessarily considering the biophysical spike generation mechanisms.

Remark Everything is not mediated by spikes in the brain! In the retina, the output cells, the ganglion cells apart [20], most of the other neurons: receptors, bipolar cells, horizontal
cells and amacrine cells, do not spike. The function of the action potential is to propagate 'reliably' a signal over 'long' (> 100 – 200 µm) distances [12]. The retina is very thin, the neurites there are 'short' and essentially passive transmission does the job. It also allows an 'analog signal encoding'.

1.3. The 'messy' aspect of spike trains

Figures 1.3 and 1.4 do not lead one to think that spike trains are very regular objects. This suggests, if we want to model them directly, that using stochastic models might be a sound idea. Before presenting actual spike train models, we will briefly review neurophysiological facts that, independently of our interest for spike trains, make clear that neurons (at least most of them) are not deterministic machines.
2. Why a modeling approach with a strong stochastic element?

2.1. Membrane noise can lead to output fluctuations

We now know something that Hodgkin and Huxley did not know when they introduced their famous model [30]: the sodium and potassium conductances responsible for the action potential are due to voltage-gated ion channels that are pore forming transmembrane proteins [28]. As most macromolecules, ion channels can be found in several transient states [34, 18] and will switch spontaneously between those states—a change in membrane voltage will change the proportion of time they spent in the different states—. In the case of voltage-gated sodium channels spontaneous transitions are observed between ‘closed’ and ‘open’ states:

Voltage-gated sodium channels have a small conductance (5-10 pS) and cases where the spontaneous opening of a single channel can give rise to a full action potential are rare, it can only occur in very tiny neurons [39] that have therefore an input resistance large enough to enable a tiny current (through a tiny conductance) to bring a membrane potential change large enough to reach threshold. But the stochastic opening of channels can lead, in some cases at least, to fluctuating / jittering spike times as was demonstrated (before channels were known) by Verveen and Derksen [58].
Fluctuation Phenomena in Nerve Membrane

A. A. VERVEEN AND H. E. DERKSEN

2.2. Synaptic noise is ubiquitous

2.2.1. Fluctuations at the neuromuscular junction

In 1952, Fatt and Katz [21] reported the observation of small spontaneous potentials from the end-plate region of the frog neuromuscular junction (Fig. 2.3 left) that they referred to as miniature end-plate potentials (mepp). They argued that these potentials originate from the spontaneous release of transmitter from the nerve terminal. They then studied the response of the same muscle fiber to presynaptic nerve stimulation in a low extracellular calcium condition—lowering thereby transmitter release—and saw that these reduced evoked responses fluctuate in steps whose amplitudes matched the ones of spontaneous potentials (Fig. 2.3 right).
Figure 2.3.: Montage of Figures 1A (left) and 9C (right) of [21]. Top left: spontaneous activity recorded intracellularly from a frog muscle fiber close to the end-plate. Bottom left: response to a presynaptic nerve stimulation recorded from the same location (lower magnification on the y scale). Top right trace: 50 Hz oscillation giving a time reference. Middle right 3 traces: spontaneous potentials (like on the left side). Bottom right trace: superposed evoked responses recorded in low calcium; the presynaptic stimulation time is indicated by the arrow.

In the same paper, Fatt and Katz studied the distribution of the intervals between successive mepp and showed it to be compatible with an exponential distribution, Figure 2.4. This is a first indication that these miniature events follow a homogeneous Poisson process (Section 4.4). These inter mepp interval data can be found in the Appendix of [16] where they are wrongly described as inter spike interval data; they are also used–with the same wrong description–in the two excellent books of Larry Wasserman [60, 61].

It should be noted that this observation of the mepp and the suggestion that ‘normal’ evoked end-plate potentials are made of a bunch of mepp predates the observation, using electron-microscopy, of what we now call synaptic vesicles in the presynaptic terminal [42, 17]. De Robertis and Bennett [17] in 1955 and Palay [43] in 1956, describing the synaptic vesicles, were the first to suggest that a mepp is due to the fusion of a single vesicle [62].

2.2.2. Quantal neurotransmitter release

In 1954, Del Castillo and Katz [13] investigated systematically the ‘composite nature’ of evoked end-plate potentials. Their conclusion are best summarized by the title they
chose for their paper: *Quantal components of the end-plate potential*. In high magnesium conditions—that reduces release since, as we now know, it blocks calcium channels—, not only could they reproduce Fatt and Katz observation of fluctuating evoked potentials [21] but they could also observe *transmission failures* as shown on Figure 2.5.
Figure 2.5.: Figures 1 and 2 of [13]. Left, Fluctuating evoked end-plate potentials in high magnesium (note the scattered mepp). Right (high magnesium and low calcium conditions): top, mepp; middle, 50 Hz cycles for time reference; bottom, evoked responses with many 'failures'; stimulus artifact and response latency are indicated by a pair of dotted vertical lines.

Comparing the mepp amplitude distribution with the one of the evoked potentials in low calcium conditions, they proposed the following scenario, commonly referred to as the quantal neurotransmitter release:

- the presynaptic terminal contains $n$ vesicles;
- when a presynaptic action potential invades the terminal, each of the $n$ vesicle will independently of the others release the transmitter it contains with a probability $p$ ($p$ depends on the extracellular calcium and magnesium concentrations);
- each vesicle that releases its content gives rise to an elementary end-plate potential whose distribution is the same as the one of the mepp;
- the evoked end-plate potential is the sum of the individual potentials due to each vesicle that released its transmitters.

2.2.3. What about the central nervous system?

These studies carried out at 'the' neuromuscular junction—a synapse designed to be reliable—exhibit marked fluctuations in specific conditions (low calcium or high magnesium) while in normal conditions the fluctuations can be neglected: every time a presynaptic spike
arrives, muscle contraction ensues. But in the central nervous system (CNS), synapses typically have a small number of vesicles \((n)\) and the release probability \((p)\) is rarely close to 1; therefore marked fluctuations are the rule, even in physiological conditions. This is illustrated on Figure 2.6 using data collected from an autapse (a synapse made by the neuron onto itself) of a molecular layer interneuron of the cerebellar cortex \([45]\). The advantage of an autapse is that with a single electrode (or patch-pipette in that case) one has access to both the pre- and post-synaptic sides. A more classical way of studying synaptic transmission in the CNS is to record from two neurons with two electrodes (or patch-pipettes) and be ‘lucky enough’ to find them synaptically connected—this take a lot of time and effort; that’s what I spent most of my PhD doing—.

![Figure 2.6.](image)

Figure 2.6.: Synaptic transmission fluctuation observed at an autapse (a synapse made by the neuron onto itself) of a molecular layer interneuron of the cerebellar cortex \([45]\). Aa top, imposed membrane voltage time course. Aa bottom, mean current responses; the continuous line shows the overall mean, while the dotted line shows the mean of responses classified as ‘failures’. The synaptic response starts slightly more than 2 ms after the end of the voltage pulse. Ab, the overall mean - the failure mean (that’s a way to get rid of the action current appearing during the pulse and due to the unclamped action potential). Ac, standard deviation time courses: continuous line from all responses, dotted line from ‘failures’. B, Individual responses after subtraction of failure mean. C, histograms of the response amplitudes and of their latency (inset).

You should keep in mind that these synaptic transmission studies in the CNS are per-
formed on brain slices and that taking a slice out of a brain is artifact prone [3].

2.2.4. Sources of quantal size fluctuations

The quantal neurotransmitter release we just described exhibits two sources of fluctuations: i) the number of vesicles released varies from one synaptic activation to the next; ii) the size of the elementary response due to a single vesicle release (the quantal size) is not fixed but described by a distribution (formally, it is modeled by a random variable). The reader can find a schematic representation of ‘typical’ CNS synapse on Figure 2.7, that should be helpful to follow the coming discussion. The quantal size distribution can originate from different sources: i) the number of transmitter molecules in a given vesicle is probably not absolutely fixed [59]; ii) the transmitter molecules diffuse in the synaptic cleft upon release and bind to postsynaptic receptors, depending on the numbers of transmitter molecules and receptors, as well as on geometrical factors—location of the fusion site above the receptors, receptors location, etc.—fluctuations of the number of receptors bound to transmitter are expected to occur [59]; iii) like the voltage-gated ion channels we discussed in Sec. 2.1, (most of) the postsynaptic receptors are also ion channels and they also go back and forth between close and open states [4].

![Figure 2.7.: Scheme of a central nervous system synapse. Source Wikimedia Commons.](image)

2.3. Other sources of variability

We have briefly described in the last two sections the most thoroughly studied sources of fluctuations in neurons. They are not the only ones known [65], for instance action potential propagation failures at branch points have been observed and could be at least partly random [55]. More importantly, in a given experimental setting, we can never record
from all the neurons involved—the opposite is the rule, we record from a very tiny fraction
of the neurons—meaning that the neurons we observe receive de facto most of their inputs
from sources we ignore; modeling these unobserved inputs as random inputs, even if they
are fully deterministic, makes sense as soon as there are many of them and that they are
not too strongly correlated.

2.4. Conclusions

We are facing now a modeling problem: where and how shall we incorporate 'stochastic
components' in our neuron models? There are clearly several possibilities; one would be to
stick to the 'detailed biophysical model' tradition and take a 'usual' model—with detailed
morphology, different conductances in different parts of the neuron, etc.—and for instance,
replace the Hodgkin and Huxley conductance model by a Markov process one [15, 63].
The somewhat 'opposite' approach would be to 'lump together' the known and unknown
sources of variability in a single component of the model. This is the way we will pursue
in the sequel and we are going to start by reviewing some of the solutions that have been
proposed along this line.

2.5. Exercises

You are free to use the language you want for your implementation, my solutions are given
in Python and, sometimes, a mixture of C and Python.

2.5.1. The inter mepp data of Fatt and Katz (1952)

This first and long exercise makes you analyze some historical data. If your last task is
to reproduce figure 12 of [21], it makes you implement on the way a full collection of
statistical tests for stationary Poisson processes [16, 41].

1. Getting the data

Larry Wasserman took the data of Fatt and Katz [21] from the Appendix of Cox and
Lewis book [16] and put them on a page of his web site dedicated to the datasets used
in his books. You can therefore download the data from http://www.stat.cmu.
edu/~larry/all-of-nonpar/=data/nerve.dat, they are organized in columns (6
columns) and all the rows are complete except the last one that contains data only
on the first column. The data are inter mepp intervals in seconds and should be read
across columns; that is first row, then second row, etc.

Your first task is to read these data in your Python environment and to keep them
in a list.

2. Basic statistics
Fatt and Katz [21, p. 123] state that they have 800 mepp and that the mean interval is 0.221 s. Compare with what you get. Do a bit more by constructing the five-number summary—namely the minimum, first quartile, median, third quartile and maximum statistics—.

3. Interval correlations

Look for correlations between successive intervals:

- Plot interval $i+1$ vs interval $i$.
- Plot the rank of interval $i+1$ vs the rank of interval $i$ [48]; comment on the difference between the first display and the present one.
- Using the ranks you just computed, get the Spearman’s rank correlation for the intervals for lags between 1 and 5, compare the obtained values with the standard error under the null hypothesis; since the data were truncated they contain many duplicates, making the theoretical confidence intervals not very reliable (see the previous link), you will therefore resample 1000 times from your ranks (use the sample function from the random module of the standard library and do not forget to set your random number generator seed with function seed of this module), compute the maximum (in absolute value) of the five coefficients (for the five different lags) and see were the actual max of five seats in this simulated sampling distribution; conclude.

4. Counting Process Class

You will first create a Class whose main private data will be an iterable (e.g. a list) containing arrival times. In the context of Fatt and Katz data, the arrival times are obtained from the cumulative sum of the intervals. Ideally class instantiation should check that the parameter value provided by the user is a suitable iterable. Your class should have two methods:

- sample_path that returns the counting process sample path corresponding to the arrival times passed as argument for any given time (using the bisect function of the bisect module can save you some work).
- plot that generates a proper step function’s graph, that is, one without vertical lines.

5. Graphical test of arrival times ‘uniformity’

We will show in Section 4.4.3 that the arrival times of a stationary Poisson counting process observed between 0 and $T$, conditioned on the number of events observed are uniformly distributed on $[0, T]$. It is therefore possible, using the improperly called one sample Kolmogorov-Smirnov test (‘Kolmogorov’ test is the proper name for it) and the Kolmogorov distribution for the test statistics to build a confidence band around the ideal counting process sample path that is known once the observation time $T$ and the number of observed events $n$ are known: it should be a straight line going through the origin and through the point $(T, n)$. Strangely enough tables giving
the asymptotic critical values of the Kolmogorov’s statistics $D_n$ ($n$ is the sample size) are rare. [5, p. 86] give the following critical values for $\sqrt{n}D_n$ ($n > 80$):

You can therefore make a quick graphical test by plotting the sample path together with the limits of the 95% and 99% confidence bands.

6. Kolmogorov’s test via C code interfacing (optional)

If you like the ‘fancy stuff’, you can decide to use ‘exact’ values (to 15 places after the decimal point, that is to the precision of a double on a modern computer) of the Kolmogorov’s statistics distribution. Marsaglia, Tsang and Wang [40] give a C code doing this job that can be freely downloaded from the dedicated page of the *Journal of Statistical Software*. If you feel advanturous, you can download this code, compile it as a shared library and interface it using module *ctypes* to your Python session. Once this interface is running, you can compute the p-value of the observed value of the statistics. You can find a very basic *ctypes* tutorial from Jülich and a more advanced in the official documentation.

7. Figure 12 reproduction and more

Reproduce Figure 12 of [21]. Then, remarking that the complementary of the (cumulative) distribution function of an exponential, $F(t) = 1 - e^{-t/\tau}$ is $F^c(t) = 1 - F(t) = e^{-t/\tau}$ make a graph of the empirical version of $F^c$ using a log scale on the ordinate.
3. Let Us Now Praise Famous Men

As announced in the previous section I am going to focus on 'simple' models lumping the 'stochastic component' at a single locus. These models won’t pretend being 'realist' in the way detailed biophysical ones do. By abandoning strict observance of physiological facts, I hope to obtain easier to interpret and to implement (on a computer) models as well as, in some cases, models on which mathematicians can work— that is, prove theorems—.

I have selected few ‘milestone’ papers in this section. This does not imply that the (many) other ones are not good, far from that. Still, the reader will remark that they are relatively old, having been published between 1954 and 1988. This is a deliberate choice that can rightfully be understood as targeting a common pathology in neuroscience: amnesia; a strange pathology to suffer from for people who often work on how the brain memorizes things...

3.1. Hagiwara (1954)

We now discuss a paper of Susumu Hagiwara, ANALYSIS OF INTERVAL FLUCTUATION OF THE SENSORY NERVE IMPULSE [26]. This article is particularly interesting for the method used by Hagiwara to compute the probability density of the intervals between successive spikes.

3.1.1. Overview

Preparation toad or bullfrog tonic muscle spindles fibers (stretch receptors of muscles); by changing the load (weight) attached to one end of the muscle, Hagiwara was able to modulate the discharge rate of the fiber (that’s classic stuff).

Results For a given load the discharge rate was first decreasing— during the first 10 seconds— before stabilizing (his Fig. 1).

1. By selecting restricted parts of the early phase and the whole late part, Hagiwara was able to isolate what he calls ‘stationary states’ (his Fig. 2)—we would rather refer to regions were the counting process exhibits stationary increments (Sec. 4.3.9). Hagiwara writes that in these regions the ‘successive intervals become a stochastic time series’; we would write that they become a sample path from a counting process with stationary increments.

2. Hagiwara looked then at the correlations between successive intervals (in the ‘stationary regions’) and found these correlations to be null (his Fig. 4B); we would say that the counting process has independent increments (Sec. 4.3.9).
A counting process with independent increments whose intervals are not necessarily exponentially distributed is a renewal process, a generalization of the Poisson process, it is fully characterized by the time of the first event and by the interval distribution (the successive intervals are assumed to be drawn independently from the distribution).

3. He found a relation between the mean interval and the standard deviation (his Table 1 and Fig. 3A) implying that the standard deviation is set by the mean value (functional relation).

4. He also found a relation between mean and skewness (his Fig. 3B and 4A) implying that the skewness is also set by the mean value.

5. The data were of insufficient duration to determine reliably the higher order moments.

6. Hagiwara concludes that in the 'stationary regions' the interval distribution depends on a single parameter.

**Discussion** The discussion focuses on the construction of a stochastic model that can account the empirical properties of the interval distributions and be related to physiological parameters. We are going next to give a more detailed account of that part of the paper.

### 3.1.2. The model

The key feature Hagiwara wants to 'explain' is that the inter spike interval distribution depends on a single parameter. To that end he proposes that, at the action potential initiation site, the threshold undergoes a transient modification following a spike, namely, he assumes that the threshold, as a function of time $t$, assuming the last spike occurred at $t = 0$, is described by:

$$ Ae^{c/t}, $$

where $A$ is the rheobase (the stimulus intensity leading to a spike when times goes to infinity or, the minimal stimulus leading to a spike after a 'long time') and $c$ is a parameter specific to the fiber. The 'membrane potential' must exceed the threshold to get a spike.

A given constant stimulus is assumed to lead to a constant membrane potential given by $S$. Lastly, Hagiwara assumes the presence of a 'noise process', that in modern language would be described by a Gaussian process [52, p. 359] with a diagonal covariance matrix having value $\sigma^2$ on the diagonal. Less formally, Hagiwara assumes that at any given time the 'membrane potential' is given by $S + X(t)$ where the $\{X(t), t \geq 0\}$ are independently and identically normally distributed with $E[X(t)] = 0$ and $V[X(t)] = \sigma^2$ (independence implies: $\text{Cov}(X(t), X(s)) = 0$, if $t \neq s$). A spike occurs when:

$$ S + X(t) \geq Ae^{c/t} \quad \text{or} \quad X(t) \geq Ae^{c/t} - S. $$
The probability of spike occurrence between $t$ and $t + \Delta t$ given that no spike occurred since $t = 0$ is then (conditioning, or what 'given' means, is properly defined in Section 4.3.6):

$$P\{\text{one spike in } (t, t + \Delta t) \mid \text{last spike at } t = 0\} = P\{X(t) \geq Ae^{c/t} - S\}$$

$$= \frac{\Delta t}{\sqrt{2\pi}} \int_{\theta(t,S)}^{\infty} e^{-x^2/2} dx,$$

where

$$\theta(t, S) = \frac{1}{\sigma} \left( Ae^{c/t} - S \right).$$

Remember that the distribution function of a standard normal random variable is:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-u^2/2} du.$$

Now if we write $\Phi^c(x) = 1 - \Phi(x)$, we see that our probability of spike occurrence can be written as:

$$P\{\text{one spike in } (t, t + \Delta t) \mid \text{last spike at } t = 0\} = \Delta t \Phi^c(\theta(t, S)).$$

### 3.1.3. Getting the inter spike interval density

To get $f(S, t)\Delta t$, the probability that an interval is between $t$ and $t + \Delta t$, Hagiwara splits $[0, t]$ in $n - 1$ intervals of equal length $\Delta t = t/(n - 1)$ and argues that $f(S, t)dt$ is the product of the probabilities that no spike occurs in the $n - 1$ intervals and the probability that a spike occurs in the interval $(t, t + \Delta t)$. Or, writing

$$\Phi^c_k = \Phi^c(\theta(k\Delta t, S)),$$

we get

$$f(S, t)\Delta t = (1 - \Delta t \Phi^c_0) (1 - \Delta t \Phi^c_1) \cdots (1 - \Delta t \Phi^c_{n-1}) \Delta t \Phi^c_n$$

$$= \left\{ \prod_{i=0}^{n-1} (1 - \Delta t \Phi^c_i) \right\} \Delta t \Phi^c_n. \quad (3.2)$$

Taking the log on both sides yields (remember that the log of a product is the sum of the logs),

$$\log(f(S, t)\Delta t) = \left\{ \sum_{i=0}^{n-1} \log(1 - \Delta t \Phi^c_i) \right\} + \log(\Delta t \Phi^c_n).$$

If $\Delta t \ll 1$, that is if $n \gg 1$, then

$$\log(1 - \Delta t \Phi^c_i) = -\Delta t \Phi^c_i + o(\Delta t),$$

$o(\Delta t)$ is precisely defined in Section 4.4.1 (Definition 2). We therefore have

$$\log(f(S, t)\Delta t) = -\left\{ \sum_{i=0}^{n-1} \Delta t \Phi^c_i \right\} + \log(\Delta t \Phi^c_n) + o(\Delta t).$$

21
But if the Riemann integral of $\Phi^c$ exists then:

$$\lim_{n \to \infty} \sum_{i=0}^{n-1} \frac{t}{n-1} \Phi^c_i = \int_0^t \Phi^c(x) \, dx.$$ 

Since the integral exists, we end up with:

$$\log(f(S, t) \Delta t) = - \left( \int_0^t \Phi^c(\theta(x, S)) \, dx \right) + \log(\Delta t \Phi^c(\theta(t, S)))$$

that is

$$f(S, t) = \Phi^c(\theta(t, S)) \exp \left( - \int_0^t \Phi^c(\theta(x, S)) \, dx \right). \quad (3.3)$$

### 3.1.4. Parameter estimation

If we have a set of $m$ (observed) intervals, \( \{t_1, \ldots, t_m\} \), we can use the maximum likelihood method to estimate parameters $c$ and $\sigma$, that is we construct the function

$$L(c, \sigma) = \prod_{i=1}^{m} f(S, t_i),$$

and we take as estimates $\hat{c}$ and $\hat{\sigma}$ of $c$ and $\sigma$ respectively the pair $(c, \sigma)$ at which the maximum of $L(c, \sigma)$ is located.

In practice we usually work with the log-likelihood—since the location of the maximum of a positive function is the same as the location of the maximum of its log, a strictly increasing function—

$$L(c, \sigma) = \log(L(c, \sigma)) = \sum_{i=1}^{m} \log f(S, t_i).$$

### 3.1.5. Comments

Instead of a 'variable threshold' hypothesis, we could view the model as a constant rate after a smooth recovery from a refractory period. Namely, we can see $-Ae^{c/t}$ as the 'membrane potential' trajectory following a spike at time $t = 0$ and $S - Ae^{c/t}$ as the membrane potential value following a spike at $t = 0$ when a stimulus of intensity $S$ is applied. Hagiwara’s model can then be reformulated by saying that the neuron spikes when the standard random variable $X(t)$ exceeds $(S - Ae^{c/t}) / \sigma$. To see that, draw the graph of standard random variable probability density and, by symmetry, you will immediately see that the area under the right tail: $x \geq (S - Ae^{c/t}) / \sigma$ equals the area under the left tail: $x \leq (S - Ae^{c/t}) / \sigma$. In other words:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(S - Ae^{c/t})/\sigma} e^{-u^2/2} \, du = \frac{1}{\sqrt{2\pi}} \int_{(Ae^{c/t} - S)/\sigma}^{\infty} e^{-u^2/2} \, du.$$
If we stick to the second interpretation of the model, that is the 'membrane potential' of the neuron, $U(t)$, following a spike at time $t = 0$ is

$$U(t) = \frac{1}{\sigma} \left( S - Ae^{c/t} \right)$$

(3.4)

and the probability to have a spike in $(t, t + dt)$ is:

$$P\{\text{one spike in } (t, t + \Delta t) \mid \text{last spike at } t = 0\} = \Delta t \sqrt{\frac{2}{\pi}} \int_{-\infty}^{U(t)} e^{-u^2/2} du,$$

we see that we have in fact a very general model, in particular, one that allows us to include time dependent events: we don't need $S$ to be constant, we could as well work with $S(t)$. Leaving the experimental context of Hagiwara's article, we could account for interactions between neurons; following a spike in a presynaptic neuron, $U(t)$ would be transiently modified, for instance as follows:

$$U(t) = \frac{1}{\sigma} \left( S - Ae^{c/t} + \sum_{t_{p,i} < t} g(t - t_{p,i}) \right),$$

(3.5)

where $\{t_{p,i}\}$ stands for the set of presynaptic spike times and were $g$ could be:

$$g(u) = \begin{cases} 0 & \text{if } u < 0 \\ w e^{-u/\tau} & \text{if } u \geq 0, \text{ with } \tau > 0 \end{cases},$$

where the synaptic weight $w$ would allow us to work with excitatory or inhibitory connections. In such a setting the 'stochastic component' is concentrated in the function converting the 'membrane potential' into the 'spike rate'—in the above case the standard normal distribution function (section 4.3.1)—.

### 3.2. Fitzhugh (1958)

The article of Richard Fitzhugh [22] that we will now discuss contains two parts: in the first a discharge model is proposed; in the second spike train decoding is studied based on the discharge model of the first part. We are going to focus here on the discharge model—this does not imply that the decoding part is not worth reading—.

#### 3.2.1. Background

In the mid-fifties, Kuffler, Fitzhugh—that’s the Fitzhugh of the Fitzhugh-Nagumo model—and Barlow [32] start studying the retina, using the cat as an animal model and concentrating on the output neurons, the ganglion cells. They describe different types of cells, on and off cells and show that when the stimulus intensity is changed, the cells exhibit a transient response before reaching a maintained discharge that is stimulus independent as shown if their figure 5, reproduced here:
They then make a statistical description of the maintained discharge that looks like:

![Figure 3.1](image1.png)

**Figure 3.1.:** Figure 5 of [32]: Frequency of maintained discharge during changes of illumination. Off-center unit recorded with indium electrode. No permanent change of frequency resulted from change of illumination.

They propose a *gamma distribution* model for the inter spike intervals (see Section 4.4.2) for the 'maintained discharge regime', we would say the *stationary regime*:

![Figure 3.2](image2.png)

**Figure 3.2.:** Figure 6 of [32].

The maintained discharge in a single off-center ganglion cell, showing fluctuations in the durations of successive intervals. Retouched photographic records.
Figure 3.3.: Figure 7 of [32]. Comparison of an 'nonparametric' probability density estimator (an histogram) with two theoretical models: an exponential plus a refractory or dead time and a gamma distribution. These are 2 cells among 6 that were analyzed in the same way.

3.2.2. Building a discharge model for the transient regime

In his 1958 paper, Fitzhugh [22] wants first to build a discharge model for the transient parts of the neuron responses; that is, what is seen of Figure 3.1 every time the light intensity is changed. Fitzhugh model boils down to:

- The neuron has its own 'clock' whose rate is influenced by the stimulus that can increase it or decrease it (as long as the rate remains nonnegative).
- The neuron always discharge following an renewal process (Section 4.5) with a fixed gamma distribution with respect to its own time (the time given by its clock).
- Since the clock rate can be modified by a stimulus, the neuron time can be ‘distorted’—the distortion is given by the integral of the rate— with respect to the ‘experimental time’ and the spike times generated by the neuron can be nonstationary and exhibit correlations.

Fitzhugh illustrates the working of his model by assuming the following rate (clock) function \( f \):

![Figure 3.4.](image)

Figure 3.4.: Figure 1a of [22].

The way to generate the observed spike train assuming this rate function is illustrated next:
You can see on the ordinate the realization of a renewal process with a gamma distribution. The curve that starts along the diagonal is the time distortion: \( u = f(t) \) is the neuron's time and \( f' \) is its clock rate. The observed process is obtained by mapping each spike time \( u_i \) of the gamma renewal process (ordinate) to the experimental time axis (abscissa) with \( t_i = f^{-1}(u_i) \) (since \( f' \) is nonnegative function, \( f \) is invertible).

3.2.3. Inference

Fitzhugh proposes to estimate \( f' \) from what we would now call a normalized version of the Peri-Stimulus Time Histogram [23]. The normalization is done by setting the rate in the stationary regime to 1.

3.2.4. Comments

Fitzhugh approach could obviously be generalized to interactions between neurons. Synaptic coupling could translate into acceleration (for excitatory synapses) and deceleration (for inhibitory ones) of the clock rate. It is strange that no one apparently explored these possibilities further. There is moreover a close connection to what statisticians call accelerated failure time model.
3.3. Gerstein & Mandelbrot (1964)

3.3.1. Background

Using data from the cochlear nucleus of cats, Gerstein and Mandelbrot in their 1964 paper [24] try to account for two data features:

1. The spike trains in the stationary regime are well approximated by renewal processes and the distribution of the sums of 2, 3 or more intervals looks sometimes like a scaled version of the distribution of a single interval convolved with itself twice, three times, etc; in other words the interval distribution might be a stable distribution (such distributions were intensely studied by Mandelbrot at that time in connection with economic time series).

2. The interval probability density functions typically exhibit a marked asymmetry (fast rise and sometimes very slow decay), even stronger than the ones fitted with gamma distribution by Kuffler, Fitzhugh and Barlow [32].

3.3.2. Model

In order to get a model with ‘heavy tails’ (a slow decay of the interval probability density function) and with parameters that can be interpreted physiologically, they propose to consider the dynamics of the membrane potential as driven by constant bombardment of not directly observed excitatory and inhibitory inputs: an excitatory input ‘gives an upward kick’ to the membrane potential while an inhibitory one gives a ‘downward kick’. Assuming:

- a large, virtually infinite, membrane time constant;
- a large number of uncorrelated presynaptic neurons leading to individual inputs of small amplitudes;

the membrane potential is expected to follow a random walk converging to a Wiener process/Brownian motion process (as the number of presynaptic neurons goes to infinity and the as the input amplitude goes to 0). If one sets then a threshold such that, every time the membrane potential reaches this threshold, an action potential is emitted and the membrane potential is reset to a standard value, one gets a stochastic model of the discharge. The first ‘problem’ of interest for us is the determination of distribution of the first passage time (FPT) or first hitting time [52, 25, 56, 53] through the threshold, since these FPT are the inter spike intervals.

If one considers situations where the excitatory inputs dominate, the membrane potential exhibits a drift and by playing with excitation/inhibition balance, one can switch from a situation with a very heavy tail and an undefined expectation for the FPT (when excitation and inhibition balance each other) to one where the FPT expectation always exists, is smaller and smaller and where the heavy tail aspect diminishes (as excitation dominates more and more inhibition).
Figure 3.6.: Figure 3 of [24]. Typical random walks in one dimension in a computer simulation of the model. [The ‘threshold’ is named ‘absorber’ on this figure. A spike is generated when the sample path of a random walk reaches the threshold/absorber. The ‘reflector’ set here at -128 does not change significantly the FPT distribution as long that the difference between threshold and reset, here 32, is large compared to the difference between reset and reflector; but for the graphical display it forces all the trajectories to stay within the graph.]

In their article Gerstein and Mandelbrot justify the FPT probability density of their random-walk-plus-drift model. This FPT distribution is an inverse Gaussian distribution also known as a Wald distribution. An elementary derivation of this density (for the continuous version of the random walk, the Wiener process/Brownian motion process, plus drift) can be found in [64]; its expression is:

$$f_{FPT}(t) = \sqrt{\frac{\lambda}{2\pi t^3}} \exp \left( -\frac{\lambda(t - \mu)^2}{2\mu^2 t} \right), \quad T, \lambda, \mu > 0, \quad (3.6)$$

where $\lambda = r^2/\sigma^2$ ($r$ is the distance in [mV] between reset and threshold and $\sigma^2$ is the diffusion constant of the Brownian motion process [mV$^2$/ms]) and $\mu = r/\delta$ ($\delta$ is the membrane potential drift [mV/ms]).

They present comparison between histogram based FPT densities estimation and fitted inverse-Gaussian densities:
3.3.3. Time dependent drift

They go further and study the effect of a time dependent drift. In that case they cannot obtain analytical solution for the FPT density and they use Monte-Carlo simulations of random walks plus drift:

Figure 3.8.: Part of figure 9 of [24]. Left, interval histograms for the time-variant random walk model. Middle, model boundaries. Right, time course of the drift parameter corresponding to 1 stimulus per 1280 steps; values for each portion are given in the form: duration, drift velocity towards absorber.
3.3.4. Comments

This model and some of its variants have enjoyed a big popularity among 'diffusion expert' probabilists [49, 53]. Again, synaptic interactions can be accommodated by allowing specific inputs to modify the drift in a deterministic fashion.

A problem appears when we want to do statistical inference in a general context (the drift depends on synaptic inputs or on stimulation). We then want to get the likelihood implying that we need to compute the probability density of each inter spike interval, many many times (for different parameter values). Simulating diffusion explicitly is not the best solution; it is slow and error prone [53, Sec. 4.3]. It is much more efficient to keep the drift at zero and to make the threshold change (with the opposite magnitude of the original drift). We are then looking for the FPT of a Brownian motion (without drift) across a time dependent boundary and there are very efficient numerical methods–with bounded errors–for that [37].


We now discuss Chornoboy, Schramm and Karr, Maximum likelihood identification of neural point process systems [14]. The point process mentioned in the title is (informally) just the process defined by the times at which the spike train counting process jumps. The raster plot of Fig. 1.2 can be considered as the sample path of a point process. In other words, 'point processes' and 'counting processes' are two sides of the same coin. More formally we could write \( \{X(t), t \geq 0\} \) our point process and define it in terms of the counting process \( \{N(t), t \geq 0\} \) by \( X(t) = \lim_{u \to t^+} N(u) - \lim_{u \to t^-} N(u) \), if the index set is continuous \( (t \in \mathbb{R}^+) \) and by \( X(t) = N(t) - N(t-1) \) if the index set is discrete \( (t \in \mathbb{Z}^+) \).

3.4.1. Overview

This article focuses on the estimation of the synaptic coupling between neurons (from observed spike trains). It is not based on actual data, a model is presented in a very detailed way; inference—the task of going from observations to model parameters—is strongly emphasized and a lot of formal properties of the estimator—the function of the data used to estimate the model parameters—are proven. An estimation algorithm, whose convergence is also proven, is presented. Then simulations are used to check the performances of the method and to compare them with alternative methods available at that time [9, 10].

3.4.2. Model

Like Hagiwara (section 3.1 and [26]), the authors model directly the rate or intensity function, that is, Hagiwara’s function \( \Phi^C (\theta(t, S)) \) of Section 3.1.3—the rate function is properly introduced and discusses in Sections 4.3.7 (Eq. 4.8) and 4.4.1—. Nevertheless, their rate function model differs from Hagiwara’s since they opt for a Hawkes process [27] rate function. More precisely, they consider a network made of \( J \) neurons. The counting
process of \( \{N_j(t), t \geq 0\} \). The counting processes interact and their interaction is modeled by the convolution of a kernel, \( h^1_{j \rightarrow i} \), specific of each neuron pair \((i,j)\) where \( i \) is the postsynaptic neuron index and \( j \) is the presynaptic index (the authors write \( h_{i,j} \) where I write \( h_{j \rightarrow i} \))—and of the differential of the presynaptic neuron \( j \) counting process. This perhaps obscure last phrase means that the effect of \( j \) on the rate function of \( i \) at time \( t \) takes the form:

\[
\int_0^t h^1_{j \rightarrow i}(t-u) dN_j(u)
\]

which means for a given spike train \( \{t^j_1, t^j_2, \ldots\} \) of neuron \( j \):

\[
\sum_{t^j_k < t} h^1_{j \rightarrow i}(t - t^j_k).
\]

The rate function of neuron \( i \) is then given by:

\[
\lambda_i(t) = h^0_i + \sum_{j=1}^J \int_0^t h^1_{j \rightarrow i}(t-u) dN_j(u), \quad (3.7)
\]

where you see that:

- a sum is made on all the neurons of the network, including \( i \) itself;
- the individual effects (of each spike from each neuron) sum linearly;
- the term \( h^0_i \), the ‘spontaneous rate’ would make the process a Poisson process if all the \( h^1_{j \rightarrow i} \) were uniformly 0,
- even if \( h^1_{i \rightarrow i} \) is the only non uniformly null kernel, the resulting process is not a renewal process—since the integral extends all the way back to zero, several of the former spikes of neuron \( i \) could have an influence on (a contribution to) \( \lambda_i(t) \), as long as \( h_{i \rightarrow i} \) does not fall to zero infinitely fast—.

An obvious weakness of this model is that since the rates \( \lambda_j \) must be nonnegative, inhibitory interactions are de facto out of the picture.

### 3.4.3. Inference

Like Hagiwara, but much more explicitly—with a thorough mathematical treatment—they maximize the log-likelihood that takes the following form assuming observation between 0 and \( T \):

\[
\mathcal{L}(\theta) = \sum_{i=1}^J \mathcal{L}_i(\theta_i),
\]

with

\[31\]
\[ \mathcal{L}_i(\theta_i) = \sum_{k=1}^{n_i} \log (\lambda_i(t^i_k; \theta_i)) \exp \left( -\int_0^T \lambda_i(u; \theta_i) du \right), \]

where the \( \{t^i_1, \ldots, t^i_{n_i}\} \) are the spikes of neuron \( i \), \( \theta_i = \{h^0_i, h^1_{i\rightarrow j}, \ldots, h^J_{i\rightarrow j}\} \) are the 'parameters' of neuron \( i \), \( \lambda_i(u; \theta_i) \) is given by equation 3.7 and \( \theta = \{\theta_1, \ldots, \theta_J\} \).

Clearly, the \( J^2 \) kernels \( h^I_{j\rightarrow i} \) cannot be arbitrary functions, otherwise inference would be impossible since the number of parameters to estimate would be infinite; so Chornoboy, Schramm and Karr restrict the kernel to a class of piecewise constant functions (or step functions) with bounded support ([0, 640] in ms) and 64 pieces (each piece 10 ms long), as illustrated below:

Figure 3.9.: Figure 2 of [14]. Part of the kernel used in simulation. Kernels are defined as piecewise functions that can be non null only between 0 and 640 ms. Each piece covers 10 ms; there are therefore 64 parameters (amplitudes) per kernel.

The authors develop an Expectation-maximization (EM) algorithm for estimating the \( J + J^2 \times 64 \) parameters of their model. An advantage of this algorithm is that the nonnegativity of the parameters is automatically satisfied. We will see in the exercises that more general (if less elegant) algorithms can also be used.

3.4.4. Examples

The authors finish the paper with some telling illustrations. They first consider a network made of 2 \( (J = 2) \) neurons, one, say neuron 1, presynaptic to the other one, neuron 2. The effect of 1 on 2, \( h_{1\rightarrow 2} \) is given by the kernel shown on Fig. 3.9. They then generate spike trains of neuron 1 with several schemes, one described as 'over-clustered', the other described as 'under-clustered'—the goal here is to move away from the homogeneous Poisson assumption made by the available methods [9, 10]—and they simulate the postsynaptic train given the presynaptic one and the interaction kernel. They then take the two simulated trains (in each of the two 'conditions') and run their inference method together with the method of choice a that time they name 'method of moments'. For the 'over-clustered' case they get:
Figure 3.10.: Figure 3 a, c and e of [14]. a, autocorrelation function of the presynaptic neuron, displaying 'over-clustering'. c, estimated kernel with the method of moments. e, estimated kernel with the proposed method.

For the 'under-clustered' case they get:

Figure 3.11.: Figure 3 b, d and f of [14]. b, autocorrelation function of the presynaptic neuron, displaying 'under-clustering'. d, estimated kernel with the method of moments. f, estimated kernel with the proposed method.

In both cases, the proposed method is clearly better (but it is not surprising since the data are then fitted with the model that generated them in the model family considered).

They consider next two neurons reciprocally connected with the same kernel and compare again the two methods:
Their method is again clearly superior. The next example involves three neurons: 1 ‘talks’ to 2 that ‘talks’ to 3, but 1 does not ‘talk’ directly to 3. The question is: does the proposed method find that there is no direct connection between 1 and 3?

And the answer is yes! The last example involves three neurons again with 1 talking to
2 and 3 (common input) without direct connection between 2 and 3. Again the question is: does the proposed method find that there is no direct connection between 2 and 3?

As discussed in the paper, inhibition can be dealt with if one ‘rectifies’ the right-hand side of Eq. 3.7, working for instance with:

$$\lambda_i(t) = \exp \left( h_0^i + \sum_{j=1}^{J} \int_{0}^{t} h_{j 	o i}^i (t - u) dN_j(u) \right).$$  \hspace{1cm} (3.8)

The problem is then that the nice EM algorithm developed by the authors does not work anymore (but more general ones can be used). Note also that rectification could in principle be done with other functions like the normal distribution function of Hagiwara or a logistic function.

Another shortcoming of the model is the difficulty to handle a stimulus. Indeed, in the proposed framework, one would also model the stimulus by a piecewise function and would add it on the right-hand side of Eq. 3.7 or within the brackets of Eq. 3.8. But ones runs then into nasty unidentifiability problems because of the interplay between $h_0^i$ and the stimulus baseline (the easy solution is to force that baseline at 0).

All the nice examples shown in the paper are nice because the whole network is observed. In real life, that is never the case and that can lead to estimating wrong connections; exactly as the method of moments does in the last example (but that critic can be made to every method presented so in this document).

This paper was essentially ignored for 15-17 years, but has recently enjoyed a rebirth [47, 29, 38, 33]. The Hawkes process is now also heavily used for the modeling of financial transactions.

3.5. Brillinger (1988)

The last article we discuss, David Brillinger’ Maximum likelihood analysis of spike trains of interacting nerve cells [8], is rarely cited although it had the most profound influence on spike train analysis as it is performed today. On the surface it is:
• the continuation of Hagiwara’s 1954 paper, with two key modifications;
• the follow up and extension of a former work of Brillinger himself together with his long time collaborator José Segundo [11] on Aplysia ganglion (this explains why only pairs of neurons are explicitly considered).

The key modifications of Hagiwara’s approach are:
1. time is discretized into bins thin enough to contain at most a single spike of a given neuron.
2. the $\theta$ function of Hagiwara’s Eq. 3.1—or rather, my reformulation of it in term of Eq. 3.4 and 3.5–become a linear function of the model parameters (it is not linear in Eq. 3.1 since parameter $c$ appears in the exponential).

Modification (1) transforms spike trains into binary vectors (with 0 in bins/vector elements that contain no spike and 1 in bins that contain one spike). For instance if the time unit is the millisecond, if the spike train for the first 10 ms is (2.23, 7.53), then using a 1 ms bin width and starting binning from 0 ms we get:

Index set $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \ldots\}$
Counting process sample path $\{0, 0, 1, 1, 1, 1, 2, 2, 2, \ldots\}$
Point process sample path $\{0, 0, 1, 0, 0, 0, 0, 1, 0, 0, \ldots\}$ the binary vector

From a modeling viewpoint we are then dealing with a sequence of Bernoulli trials, one per bin. That is, in bin $t$ a biased coin is tossed with a probability $p(t)$ to fall on head (give a spike) and $1 - p(t)$ to fall on tail (no spike). From a statistical viewpoint, the problem of estimating the $\{p(t)\}$ is called a binary regression. Note that stating that Brillinger ‘modifies’ Hagiwara’s approach here is bit exaggerated, since this the precisely the first method Hagiwara uses to compute the probability of an inter spike interval (Eq. 3.2). The real difference is that Brillinger absolutely needs discretization while in Hagiwara’s case it can be viewed as a technical devise to reach Eq. 3.3. The ‘absolute necessity’ I just mentioned becomes clear when Brillinger combines (1) and (2) since he end’s up in a generalized linear model [19], the now famous GLM for almost everyone mentions in neuroscience without being able to explain what it is or what it means... To be more explicit, Brillinger writes $\{X_C(t), t \geq 0\}$, respectively $\{X_B(t), t \geq 0\}$, the counting process (associated to the spike train) of $C$, respectively $B$, the probability of the spike train of $C$ can then be written (assuming that observations are available between 0 and $T$):

$$\prod_{t=0}^{T} p_C(t)^{X_C(t)} (1 - p_C(t))^{1-X_C(t)}.$$

The functional form of $p_C(t)$ adopted by Brillinger is:

$$p_C(t) = \Phi(\sum_{k=L_C(t)+1}^{t} h_k X_B(t-k) + \alpha_1 (t - L_C(t) - 1)$$
$$+ \alpha_2 (t - L_C(t) - 1)^2 + \alpha_3 (t - L_C(t) - 1)^3 - \alpha_0),$$
where \( L_C(t) \) stands for the index of the last spike of neuron \( C \), prior to \( t \) \((L_C(t) = \sup\{j < t : X_C(j) = 1\})\); the \( \{h_k\} \) describe the postsynaptic effect in neuron \( C \) at time \( t \) of a spike that occurred at time \( t - k \) in neuron \( B \); the \( \{\alpha_0, \alpha_1, \alpha_2, \alpha_3\} \) describe intrinsic properties of neuron \( C \) (they can be viewed either as a variable threshold or as a self-inhibitory effect), \( \Phi \) is the standard normal distribution function.

We could make this formulation much more similar to the one of Chornoboy, Schramm and Karr [14] by writing for the first term on the right-hand side:

\[
\sum_{k=L_C(t)+1}^{t} h_k X_B(t-k) = \int_{L_C(t)+1}^{t} h_{B \to C}^1(t-u) dN_B(u),
\]

where \( h_{B \to C}^1 \) would also be piecewise constant; for the ‘second’ term:

\[
\alpha_1 (i - L_C(t)) + \alpha_2 (i - L_C(t))^2 + \alpha_3 (i - L_C(t))^3 = \int_{L_C(t)+1}^{t} h_{C \to C}^1(t-u) dN_C(u)
\]

where \( h_{C \to C}^1(x) \) would be \( \alpha_1 + 2\alpha_2 x + 3\alpha_3 x^2 \), and for the last:

\[\alpha_0 = -h_C^0.\]

We would then write our probability of observing a spike from neuron \( C \) at time \( t \) as:

\[
p_C(t) = \Phi \left( \int_{L_C(t)+1}^{t} h_{B \to C}^1(t-u) dN_B(u) + \int_{L_C(t)+1}^{t} h_{C \to C}^1(t-u) dN_C(u) + h_C^0 \right).
\]

We see here a key difference between Brillinger’s model and the ‘rectified’ Hawkes process of Chornoboy, Schramm and Karr [14]: the integration lower limit is set by the time/index of the last spike from the neuron of interest in the former, while it extends all the way back to the recording origin in the latter. Brillinger’s model is a model with reset (a property that is discussed and justified by Brillinger and Segundo [11]).

The ‘insistence’ of Brillinger to put the spike train analysis problem into a GLM framework is justified on two grounds:

- the GLM theory was already fully worked out at the time–estimator convergence, estimator variance, etc.–,

- a generic software, GLIM, was already available (and thoroughly tested) at that time to fit GLM to data and Brillinger did distribute his GLIM script–that makes him not only the most important contributor to spike train analysis from the statisticians community, but also a pioneer of reproducible research.
4. A Probability Reminder (or Primer)

I’m going to use mainly two books by Sheldon M. Ross:


There are a lot of excellent books around. If you’re looking for something more elementary, Brémaud’s An Introduction to Probabilistic Modeling, Springer, 1988 [6], is great (the more recent French version: Initiation aux Probabilités et aux chaînes de Markov, Springer, 2009 [7], is even better). If you’re looking for more advanced (but still “readable” by non-mathematicians) stuff, Ross’ Introduction to Probability Models, Wiley, 2014 [50] and Grimmett and Stirzaker’s Probability and Random Processes, Cambridge University Press, 2001 [25], are cool.

4.1. Random experiment, sample space, event

Probability “starts” with the notion of random experiment: an experiment whose outcome cannot be determined in advance. The set of all possible outcomes of an experiment is called the sample space, and we denote it by $S$.

An event is a subset of a sample space, and is said to occur if the outcome of the experiment is an element of that subset.

4.2. Probability

We shall suppose that for each event $E$ of the sample space $S$ a number $P(E)$ is defined and satisfies the following three axioms:

Axiom (1) $0 \leq P(E) \leq 1$.

Axiom (2) $P(S) = 1$.

Axiom (3) For any sequence of events $E_1, E_2, \ldots$ that are mutually exclusive, that is, events for which $E_i \cap E_j = \emptyset$ when $i \neq j$ (where $\emptyset$ is the null set),

$$P(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} P(E_i).$$

We refer to $P(E)$ as the probability of event $E$. 
4.3. Random variables

Consider a random experiment having sample space $S$. A random variable $X$ is a function that assigns a real value to each outcome in $S$. For any set of real number $A$, the probability that $X$ will assume a value that is contained in the set $A$ is equal to the probability that the outcome of the experiment is contained in $X^{-1}(A)$. That is,

$$\mathbb{P}\{X \in A\} = \mathbb{P}(X^{-1}(A)),$$

where $X^{-1}(A)$ is the event consisting of all points $s \in S$ such that $X(s) \in A$.

4.3.1. Distribution function

The distribution function $F$ of the random variable $X$ is defined for any real number $x$ by

$$F(x) = \mathbb{P}\{X \leq x\} = \mathbb{P}\{X \in (-\infty, x]\}.$$

We shall denote $1 - F(x)$ by $F^c(x)$, and so

$$F^c(x) = \mathbb{P}\{X > x\}.$$

When the random variable $X$ takes only positive or null values, $F^c$ is often called the survival function.

4.3.2. Discrete and continuous random variables

A random variable is set to be discrete if its set of possible values is countable. For discrete random variables,

$$F(x) = \sum_{y \leq x} \mathbb{P}\{X = y\}.$$

A random variable is called continuous if there exists a function $f(x)$, called the probability density function, such that

$$\mathbb{P}\{X \in B\} = \int_B f(x) \, dx$$

for every set $B$. Since $F(x) = \int_{-\infty}^x f(x) \, dx$, it follows that

$$f(x) = \frac{d}{dx} F(x).$$

4.3.3. Joint distribution function, independence

The joint distribution function $F$ of two random variables $X$ and $Y$ is defined by

$$F(x, y) = \mathbb{P}\{X \leq x, Y \leq y\}.$$
The distribution functions of $X$ and $Y$ can then be shown to satisfy:

$$F_X(x) = \mathbb{P}\{X \leq x\} = \lim_{y \to \infty} F(x, y) \quad \text{and} \quad F_Y(y) = \mathbb{P}\{Y \leq y\} = \lim_{x \to \infty} F(x, y).$$

The random variables $X$ and $Y$ are said to be **independent** if

$$F(x, y) = F_X(x) F_Y(y)$$

for all $x$ and $y$. The joint distribution of any collection $X_1, X_2, \ldots, X_n$ of random variables is defined by

$$F(x_1, x_2, \ldots, x_n) = \mathbb{P}\{X_1 \leq x_1, \ldots, X_n \leq x_n\}.$$

Furthermore, the $n$ random variables are said to be independent if

$$F(x_1, x_2, \ldots, x_n) = F_{X_1}(x_1) F_{X_2}(x_2) \cdots F_{X_n}(x_n),$$

where

$$F_{X_i}(x_i) = \lim_{x_j \to \infty, j \neq i} F(x_1, x_2, \ldots, x_n).$$

### 4.3.4. Expected value

The **expectation** or **mean** of the random variable $X$, denoted by $\mathbb{E}[X]$, is defined by

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x dF(x) = \begin{cases} \int_{-\infty}^{\infty} x f(x) dx & \text{if } X \text{ is continuous} \\ \sum_x x \mathbb{P}\{X = x\} & \text{if } X \text{ is discrete} \end{cases}$$

(4.1)

provided that the above integral exists.

Equation 4.1 also defines the expectation of any function of $X$, say $h(X)$. Since $h(X)$ is itself a random variable, it follows from 4.1 that

$$\mathbb{E}[h(X)] = \int_{-\infty}^{\infty} x dF_h(x),$$

where $F_h$ is the distribution function of $h(X)$. However, it can be shown that this is identical to $\int_{-\infty}^{\infty} h(x) dF(x)$. That is,

$$\mathbb{E}[h(X)] = \int_{-\infty}^{\infty} h(x) dF(x).$$

(4.2)

The **variance** of the random variable $X$ is defined by

$$\text{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}^2[X].$$

Two jointly distributed random variables $X$ and $Y$ are said to be uncorrelated if their **covariance**, defined by

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$
is zero. It follows that independent random variables are uncorrelated. However, the converse is not true.

An important property of expectations is that the expectation of a sum of random variables is equal to the sum of the expectations.

\[ E \left[ \sum_{i=1}^{n} X_i \right] = \sum_{i=1}^{n} E[X_i]. \]  \hspace{1cm} (4.3)

The corresponding property for the variance is that

\[ \text{Var} \left[ \sum_{i=1}^{n} X_i \right] = \sum_{i=1}^{n} \text{Var}[X_i] + 2 \sum_{i<j} \text{Cov}(X_i, X_j). \]  \hspace{1cm} (4.4)

### 4.3.5. Moment generating, characteristic functions and Laplace transforms

The moment generating function of \( X \) is defined by:

\[ \psi(t) = E \left[ e^{tX} \right] = \int e^{tx} dF(x). \]

All the moments of \( X \) can be successively obtained by differentiating \( \psi \) and then evaluating at \( t = 0 \). That is,

\[ \psi'(t) = E \left[ X e^{tX} \right], \]
\[ \psi''(t) = E \left[ X^2 e^{tX} \right], \]
\[ \vdots \]
\[ \psi^n(t) = E \left[ X^n e^{tX} \right]. \]

Evaluating at \( t = 0 \) yields

\[ \psi^n(t) = E[X^n], \hspace{1cm} n \geq 1. \]

It should be noted that we have assumed that it is justifiable to interchange the differentiation and integration operations. This is usually the case.

*When a moment generating function exists, it uniquely determines the distribution. This is quite important because it enables us to characterize the probability distribution of a random variable by its generating function.*

Table 4.1 gives the moment generating functions of common discrete distributions, while table 4.2 gives the ones of common continuous functions.

**Example 1.** Let \( X \) and \( Y \) be independent normal random variables with respective means \( \mu_X \) and \( \mu_Y \), and respective variances \( \sigma_X^2 \) and \( \sigma_Y^2 \). The moment generating function of their
Discrete Probability Distribution | Probability Mass Function, \( p(x) \) | Moment Generating Function, \( \psi(t) \) | Mean Variance
--- | --- | --- | ---
**Binomial with** parameters \( n, p \), \( 0 \leq p \leq 1 \) | \( \left( \begin{array}{c} n \\ x \end{array} \right) p^x(1-p)^{n-x} \) | \( (pe^t + (1-p))^n \) | \( np \) \( np(1-p) \)
**Poisson with parameter** \( \lambda > 0 \) | \( \frac{\lambda^x}{x!} e^{-\lambda} \) | \( \exp \{ \lambda(e^t - 1) \} \) | \( \lambda \) \( \lambda \)
**Geometric with parameter** \( 0 \leq p \leq 1 \) | \( p(1-p)^{x-1} \) | \( \frac{pe^t}{1-(1-p)e^t} \) | \( \frac{1}{p} \) \( \frac{1-p}{p^2} \)
**Negative binomial with parameters** \( r, p \) | \( \left( \begin{array}{c} x-1 \\ r-1 \end{array} \right) p^r(1-p)^{x-r} \) | \( \left( \frac{pe^t}{1-(1-p)e^t} \right)^r \) | \( \frac{r}{p} \) \( \frac{r(1-p)}{p^2} \)

Table 4.1.: Common discrete probability distributions and their moment generating functions.

Continuous Probability Distribution | Probability Density Function, \( f(x) \) | Moment Generating Function, \( \psi(t) \) | Mean Variance
--- | --- | --- | ---
**Uniform over** \( (a, b) \) | \( \frac{1}{b-a}, a < x < b \) | \( \frac{e^{bt} - e^{at}}{b-a} \) | \( \frac{a+b}{2} \) \( \frac{(b-a)^2}{12} \)
**Exponential with parameter** \( \lambda > 0 \) | \( \lambda e^{-\lambda x}, x \geq 0 \) | \( \frac{\lambda}{\lambda-t} \) | \( \frac{1}{\lambda} \) \( \frac{1}{\lambda^2} \)
**Gamma with parameters** \( (n, \lambda) \), \( \lambda > 0 \) | \( \frac{\lambda e^{-\lambda x} x^{n-1}}{(n-1)!}, x \geq 0 \) | \( \left( \frac{\lambda}{\lambda-t} \right)^n \) | \( \frac{n}{\lambda} \) \( \frac{n}{\lambda^2} \)
**Normal with parameters** \( (\mu, \sigma^2) \) | \( \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2} \), \( -\infty < x < \infty \) | \( \exp \{ \mu t + \frac{\sigma^2 t^2}{2} \} \) | \( \mu \) \( \sigma^2 \)
**Beta with parameters** \( a > 0, b > 0 \) | \( c x^{a-1}(1-x)^{b-1}, 0 < x < 1 \) | \( \sum_{k=1}^{\infty} \left( \frac{1}{(a^k b)^k \Gamma(k)} \right) \frac{\sigma^2}{\sigma^2} \) \( \frac{\sigma^2}{(a+b)^2(a+b+1)} \)

Table 4.2.: Common continuous probability distributions and their moment generating functions.

The moment generating function of the sum is given by:

\[
\psi_{X+Y}(t) = E \left[ e^{t(X+Y)} \right] = E \left[ e^{tX} \right] E \left[ e^{tY} \right] \quad \text{(by independence)}
\]

\[
= \psi_X(t) \psi_Y(t)
\]

\[
= \exp \left\{ (\mu_X + \mu_Y)t + \frac{(\sigma^2_X + \sigma^2_Y)t^2}{2} \right\}.
\]

where the last equality comes from Table 4.2. Thus the moment generating function of \( X + Y \) is that of a normal random variable with mean \( \mu_X + \mu_Y \) and variance \( \sigma^2_X + \sigma^2_Y \). By
uniqueness, this is the distribution of $X + Y$.

As the moment generating function of a random variable $X$ need not exist, it is theoretically convenient to define the characteristic function of $X$ by

$$
\phi(t) = \mathbb{E}[e^{itX}] , \quad -\infty < t < \infty ,
$$

where $i = \sqrt{-1}$. It can be shown the $\phi$ always exists and, like the moment generating function, uniquely determines the distribution of $X$.

We also define the joint moment generating function of the random variables $X_1, \ldots, X_n$ by

$$
\psi(t_1, \ldots, t_n) = \mathbb{E}\left[ \exp \left( \sum_{i=1}^{n} t_i X_i \right) \right],
$$

or the joint characteristic function by

$$
\phi(t_1, \ldots, t_n) = \mathbb{E}\left[ \exp \left( i \sum_{i=1}^{n} t_i X_i \right) \right].
$$

It may be proven that the joint moment generating function (when it exists) or the joint characteristic function uniquely determines the joint distribution.

**Example 2. The Multivariate Normal Distribution** Let $Z_1, \ldots, Z_n$ be independent standard normal random variables. If for some constants $\mu_i$, and $a_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$,

$$
X_1 = a_{11} Z_1 + \cdots + a_{1n} Z_n + \mu_1 ,
X_2 = a_{21} Z_1 + \cdots + a_{2n} Z_n + \mu_2 ,
\vdots
X_i = a_{i1} Z_1 + \cdots + a_{in} Z_n + \mu_i ,
\vdots
X_m = a_{m1} Z_1 + \cdots + a_{mn} Z_n + \mu_m .
$$

then the random variables $X_1, \ldots, X_m$ are said to have a multivariate normal distribution.

Let us now consider

$$
\psi(t_1, \ldots, t_m) = \mathbb{E}\left[ \exp \left\{ t_1 X_1 + \cdots + t_m X_m \right\} \right] ,
$$

the joint moment generating function of $X_1, \ldots, X_m$. The first thing to note is that since $\sum_{i=1}^{m} t_i X_i$ is itself a linear combination of the independent normal random variables $Z_1, \ldots, Z_n$ it is also normally distributed. Its mean and variance are

$$
\mathbb{E}\left[ \sum_{i=1}^{m} t_i X_i \right] = \sum_{i=1}^{m} t_i \mu_i
$$
and

\[ \forall \left( \sum_{i=1}^{m} t_i X_i \right) = \text{Cov} \left( \sum_{i=1}^{m} t_i X_i, \sum_{i=j}^{m} t_j X_j \right) \]

\[ = \sum_{i=1}^{m} \sum_{j=1}^{m} t_i t_j \text{Cov}(X_i, X_j). \]

Now, if \( Y \) is a normal random variable with mean \( \mu \) and variance \( \sigma^2 \) then

\[ \mathbb{E} \left( e^Y \right) = \psi_Y(t) \bigg|_{t=1} = e^{\mu+\sigma^2/2}. \]

Thus we see that

\[ \psi(t_1, \ldots, t_m) = \exp \left\{ \sum_{i=1}^{m} t_i X_i + 1/2 \sum_{i=1}^{m} \sum_{j=1}^{m} t_i t_j \text{Cov}(X_i, X_j) \right\}, \]

which shows that the joint distribution of \( X_1, \ldots, X_m \) is completely determined from a knowledge of the values of \( \mathbb{E}[X_i] \) and \( \text{Cov}(X_i, X_j) \), \( i, j = 1, \ldots, m \).

When dealing with random variables that only assume nonnegative values, it is sometimes more convenient to use Laplace transforms rather than characteristic functions. The Laplace transform of the distribution \( F \) is defined by

\[ \tilde{F}(s) = \int_{0}^{\infty} e^{sx} dF(x). \]

This integral exists for complex variables \( s = a + bi \), where \( a \geq 0 \). As in the case of characteristic functions, the Laplace transform uniquely determines the distribution.

We may also define Laplace transforms for arbitrary functions in the following manner: The Laplace transform of the function \( g \), denoted \( \tilde{g} \), is defined by

\[ \tilde{g}(s) = \int_{0}^{\infty} e^{sx} dg(x) \]

provided the integral exists. It can be shown that \( \tilde{g} \) determined \( g \) up to an additive constant.

4.3.6. Conditional expectation

If \( X \) and \( Y \) are discrete random variables, the conditional probability mass function of \( X \), given \( Y = y \), is defined, for all \( y \) such that \( \mathbb{P}\{Y = y\} > 0 \) by

\[ \mathbb{P}\{X = x \mid Y = y\} = \frac{\mathbb{P}\{X = x, Y = y\}}{\mathbb{P}\{Y = y\}}. \]

The conditional distribution function of \( X \), given \( Y = y \), is defined by

\[ F(x \mid y) = \mathbb{P}\{X \leq x \mid Y = y\} \]
and the conditional expectation of $X$, given $Y = y$, by

$$
\mathbb{E}[X \mid Y = y] = \int x dF(x \mid y) = \sum_x x \mathbb{P}\{X = x \mid Y = y\}.
$$

If $X$ and $Y$ have a joint probability density function $f(x, y)$, the \textit{conditional probability density function} of $X$, given $Y = y$, is defined for all $y$ such that $f_Y(y) > 0$ by

$$
f(x \mid y) = \frac{f(x, y)}{f_Y(y)},
$$

and the conditional probability distribution function of $X$, given $Y = y$, by

$$
F(x \mid y) = \mathbb{P}\{X \leq x \mid Y = y\} = \int_{-\infty}^{x} f(u \mid y) du.
$$

The conditional expectation of $X$, given $Y = y$, is defined, in this case, by

$$
\mathbb{E}[X \mid Y = y] = \int_{-\infty}^{\infty} f(x \mid y) dx.
$$

Thus all definitions are exactly as in the unconditional case except that all probabilities are now conditional on the event $Y = y$.

Let us denote $\mathbb{E}[X \mid Y]$ that function of the random variable $Y$ whose value at $Y = y$ is $\mathbb{E}[X \mid Y = y]$. An extremely useful property of the conditional expectation is that for all random variables $X$ and $Y$

$$
\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]] = \int \mathbb{E}[X \mid Y = y] dF_Y(y) \quad \text{(4.5)}
$$

when the expectation exists.

If $Y$ is a discrete random variable, then Equation 4.5 states

$$
\mathbb{E}[X] = \sum_y \mathbb{E}[X \mid Y] \mathbb{P}\{Y = y\}.
$$

While if $Y$ is continuous with density $f_Y(y)$, then Equation 4.5 says

$$
\mathbb{E}[X] = \int_{-\infty}^{\infty} \mathbb{E}[X \mid Y = y] f_Y(y) dy.
$$

Thus from Equation 4.5 we see that $\mathbb{E}[X]$ is a weighted average of the conditional expected value of $X$ given that $Y = y$, each of the terms $\mathbb{E}[X \mid Y = y]$ being weighted by the probability of the event on which it is conditioned.

\textbf{Example 3. The Sum of a Random Number of Random Variables} Let $X_1, X_2, \ldots$ denote a sequence of independent and identically distributed random variables; and let $N$ denote a nonnegative integer valued random variable that is independent of the sequence
We shall compute the moment generating function of $Y = \sum_{i=1}^{N} X_i$ by first conditioning on $N$. Now

$$E \left[ \exp \left\{ t \sum_{i=1}^{N} X_i \right\} \mid N = n \right] = E \left[ \exp \left\{ t \sum_{i=1}^{n} X_i \right\} \right] \quad \text{(by independence)}$$

$$= (\psi_X(t))^n,$$

where $\psi_X(t) = E \left[ e^{tx} \right]$ is the moment generating function of $X$. Hence,

$$E \left[ \exp \left\{ t \sum_{i=1}^{N} X_i \right\} \mid N \right] = (\psi_X(t))^N$$

and so

$$\psi_Y(t) = E \left[ \exp \left\{ t \sum_{i=1}^{N} X_i \right\} \right] = E \left[ \left( \psi_X(t) \right)^N \right].$$

To compute the mean and variance of $Y = \sum_{i=1}^{N} X_i$, we differentiate $\psi_Y(t)$ as follows:

$$\psi_Y'(t) = E \left[ N \left( \psi_X(t) \right)^{N-1} \psi_X'(t) \right],$$

$$\psi_Y''(t) = E \left[ N(N-1) \left( \psi_X(t) \right)^{N-2} \left( \psi_X'(t) \right)^2 + N \left( \psi_X(t) \right)^{N-1} \psi_X''(t) \right].$$

Evaluating at $t = 0$ gives

$$E[Y] = E \left[ N \mathbb{E}[X] \right] = E[N] \mathbb{E}[X]$$

and

$$E \left[ Y^2 \right] = E \left[ N(N-1) \mathbb{E}^2[X] + N \mathbb{E} \left[ X^2 \right] \right]$$

$$= E[N] \mathbb{V}(X) + E[N^2] \mathbb{E}^2[X].$$

Hence,

$$\mathbb{V}(Y) = E \left[ Y^2 \right] - \mathbb{E}^2[Y]$$

$$= E[N] \mathbb{V}(X) + E^2[X] \mathbb{V}(N).$$

Not only can we obtain expectations by first conditioning upon an appropriate random variable but we may also use this approach to compute probabilities. To see this, let $E$ denote an arbitrary event and define the indicator random variable $X$ by

$$X = \begin{cases} 1 & \text{if } E \text{ occurs} \\ 0 & \text{if } E \text{ does not occur.} \end{cases}$$
It follows from the definition of $X$ that
\[
E[X] = P\{E\}
\]
\[
E[X \mid Y = y] = P\{E \mid Y = y\} \quad \text{for any random variable } Y.
\]

Therefore from Equation 4.5 we obtain that
\[
P\{E\} = \int P\{E \mid Y = y\} \, dF_Y(y).
\]

### 4.3.7. The exponential distribution, lack of memory and hazard rate function

A continuous random variable $X$ is said to have an exponential distribution with parameter $\lambda$, $\lambda > 0$, if its probability density is given by
\[
f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases}
\]
or, equivalently, if its distribution is
\[
F(x) = \int_{-\infty}^{x} f(y) \, dy = \begin{cases} 1 - e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases}
\]

The moment generating function of the exponential distribution is given by
\[
E[e^{tX}] = \int_{0}^{\infty} e^{tx} \lambda e^{-\lambda x} \, dx = \frac{\lambda}{\lambda - t}.
\]

All the moments of $X$ can now be obtained by differentiating 4.6, and the reader can easily check that
\[
E[X] = 1/\lambda, \quad V(X) = 1/\lambda^2.
\]

The usefulness of the exponential random variables derives from the fact that they possess the memoryless property, where a random variable $X$ is said to be without memory, or memoryless, if
\[
P\{X > s + t \mid X > t\} = P\{X > s\} \quad \text{for } s, t \geq 0.
\]

If we think of $X$ as being the lifetime of some instrument, then 4.7 states that the probability that the instrument lives for at least $s + t$ hours, given that it has survived $t$ hours, is the same as the initial probability that it lives for at least $s$ hours. In other words, if the instrument is alive at time $t$, then the distribution of its remaining life is the original lifetime distribution. The condition 4.7 is equivalent to
\[
F_c(s + t) = F_c(s)F_c(t),
\]

47
and since this is satisfied when $F$ is the exponential, we see that such random variables are memoryless.

It turns out that not only is the exponential distribution "memoryless," but it is the unique distribution possessing this property. To see this, suppose that $X$ is memoryless and let $F^c(x) = \mathbb{P}\{X > x\}$. Then

$$F^c(s + t) = F^c(s)F^c(t).$$

That is $F^c$ satisfies the functional equation

$$g(s + t) = g(s)g(t).$$

However, the only solutions of the above equation that satisfy some reasonable condition (such as monotonicity, continuity, or even measurability) are of the form $g(x) = e^{\lambda x}$ for some suitable value of $\lambda$. A simple proof when $g$ is assumed continuous is as follows:

Since $g(2/3) = g(1/3 + 1/3) = g(1/3)$, it follows that $g(1/3) = g(1/3 + 1/3) = g(1/3)$. Repeating this yields $g(m/n) = g^n(1/n)$. Also $g(1/3) = g(1/3 + \cdots + 1/3) = g^n(1/n)$. Hence $g(m/n) = (g(1))^{m/n}$, which implies since $g$ is continuous, that $g(x) = (g(1))^x$ (a real can be obtained as the limit of a sequence of rational using continued fractions). Since $g(1) = g^2(1/2) \geq 0$, we obtain $g(x) = e^{-\lambda x}$, where $\lambda = -\log(g(1))$. Since the distribution function is always right continuous, we must have:

$$F^c(x) = e^{-\lambda x}.$$

The memoryless property of the exponential is further illustrated by the failure rate function (also called the hazard rate function) of the exponential distribution. Consider a continuous random variable $X$ having distribution function $F$ and density $f$. The failure (or hazard) rate function $\lambda(t)$ is defined by

$$\lambda(t) = \frac{f(t)}{F^c(t)}.$$  \hspace{1cm} (4.8)

To interpret $\lambda(t)$, think of $X$ as being the lifetime of some item, and suppose that $X$ has survived for $t$ hours and we desire the probability that it will survive for an additional time $dt$. That is, consider $\mathbb{P}\{X \in (t, t + dt) \mid X > t\}$. Now

$$\mathbb{P}\{X \in (t, t + dt) \mid X > t\} = \frac{\mathbb{P}\{X \in (t, t + dt), X > t\}}{\mathbb{P}\{X > t\}} = \frac{\mathbb{P}\{X \in (t, t + dt)\}}{\mathbb{P}\{X > t\}} = \frac{f(t)dt}{F^c(t)} = \lambda(t)dt.$$
That is, $\lambda(t)$ represent the probability intensity that a $t$-year-old item will fail.

Suppose now that the lifetime distribution is exponential. Then, by the memoryless property, it follows that the distribution of remaining life for a $t$-year-old item is the same as for a new item. Hence $\lambda(t)$ should be constant. This checks out since

$$\lambda(t) = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda.$$ 

Thus, the failure rate function for the exponential distribution is constant. The parameter $\lambda$ is often referred to as the rate of the distribution. (Note that the rate is the reciprocal of the mean, and vice versa.)

It turns out that the failure rate function $\lambda(t)$ uniquely determines the distribution $F$. To prove this, note that

$$\lambda(t) = -\frac{d}{dt} F^c(t).$$

Integration yields

$$\log F^c(t) = - \int_0^t \lambda(u) du + k$$

or

$$F^c(t) = c \exp \left\{ - \int_0^t \lambda(u) du \right\}.$$ 

Letting $t = 0$ shows that $c = 1$ and so

$$F^c(t) = \exp \left\{ - \int_0^t \lambda(u) du \right\}.$$ 

4.3.8. Limit theorems

Some of the most important results in probability theory are in the form of limit theorems. The two most important ones are:

**Theorem. Strong Law of Large Numbers** If $X_1, X_2, \ldots$ are independent and identically distributed with mean $\mu$, then

$$\Pr \left\{ \lim_{n \to \infty} \frac{X_1 + \cdots + X_n}{n} = \mu \right\} = 1.$$ 

**Theorem. Central Limit Theorem** If $X_1, X_2, \ldots$ are independent and identically distributed with mean $\mu$, and variance $\sigma^2$, then

$$\lim_{n \to \infty} \Pr \left\{ \frac{X_1 + \cdots + X_n - n\mu}{\sigma \sqrt{n}} \leq a \right\} = \int_{-\infty}^a \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx.$$ 

Thus if we let $S_n = \sum_{i=1}^n X_i$, where $X_1, X_2, \ldots$ are independent and identically distributed, then the Strong Law of Large Numbers states that, with probability 1, $S_n/n$ will converge to $\mathbb{E}[X_i]$; whereas the Central Limit Theorem states that $S_n$ will have an asymptotic normal distribution as $n \to \infty$. 

49
4.3.9. Stochastic processes

A stochastic process $X = \{X(t), t \in T\}$ is a collection of random variables. That is, for each $t$ in the index set $T$, $X(t)$ is a random variable. We often interpret $t$ as time and call $X(t)$ the state of the process at time $t$. If the index set $T$ is a countable set, we call $X$ a discrete-time stochastic process, and if $T$ is a continuum, we call it continuous-time process.

Any realization of $X$ is called a sample path. For instance, if events are occurring randomly in time and $X(t)$ represents the number of events that occur in $[0, t]$, then Figure 4.1 gives a sample path of $X$ which corresponds to the initial event occurring at time 1, the next event at time 3 and the third at time 4, and no events anywhere else.

![Sample path of $X(t)$](image)

**Figure 4.1.:** A sample path of $X(t) =$ number of events in $[0, t]$.

A continuous-time stochastic process $\{X(t), t \in T\}$ is said to have independent increments if for all $t_0 < t_1 < t_2 < \cdots < t_n$, the random variables

$$X(t_1) - X(t_0), X(t_2) - X(t_1), \ldots, X(t_n) - X(t_{n-1})$$

are independent. It is said to posses stationary increments if $X(t + s) - X(t)$ has the same distribution for all $t$. That is, it possesses independent increments if the changes in the processes’ value over nonoverlapping time intervals are independent; and it possesses stationary increments if the distribution of the change in value between any two points depends only on the distance between those points.
4.4. The Poisson process

4.4.1. Counting processes

A stochastic process \( \{N(t), t \geq 0\} \) is said to be a counting process if \( N(t) \) represents the total number of 'events' that have occurred up to time \( t \). Hence a counting process \( N(t) \) must satisfy:

1. \( N(t) \geq 0 \).
2. \( N(t) \) is integer valued.
3. if \( s < t \), then \( N(s) \leq N(t) \).
4. For \( s < t \), \( N(t) - N(s) \) equals the number of events that have occurred in the interval \((s, t]\).

A counting process is said to possess independent increments if the number of events that occur in disjoint time intervals are independent. For example, this means that the number of events that have occurred by time \( t \) (that is, \( N(t) \)) must be independent of the number of events occurring between times \( t \) and \( t + s \) (that is, \( N(t + s) - N(t) \)).

A counting process is said to possess stationary increments if the number of events that occur in any interval of time depends only on the length of the time interval. In other words, the process has stationary increments if the number of events in the interval \((t_1 + s, t_2 + s]\) (that is, \( N(t_2 + s) - N(t_1 + s) \)) has the same distribution as the number of events in interval \((t_1, t_2]\) (that is, \( N(t_2) - N(t_1) \)) for all \( t_1 < t_2 \), and \( s > 0 \).

One of the most important types of counting processes is the Poisson process, which is defined as follows.

**Definition 1.** The counting process \( \{N(t), t \geq 0\} \) is said to be a Poisson process having rate \( \lambda \), \( \lambda > 0 \), if:

1. \( N(0) = 0 \).
2. The process has independent increments.
3. The number of events in any interval of length \( t \) is Poisson distributed with mean \( \lambda t \). That is, for all \( s, t \geq 0 \),
   \[
P\{N(t + s) - N(s) = n\} = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, \quad n = 0, 1, \ldots
   \]

Note that it follows from condition (3) that a Poisson process has stationary increments and also that

\[
\mathbb{E}[N(t)] = \lambda t,
\]

which explains why \( \lambda \) is called the rate of the process.

In order to determine if an arbitrary counting process is actually a Poisson process, we must show that conditions (1), (2) and (3) are satisfied. Condition (1), which simply

Poisson process defined through the Poisson distribution

51
states that the counting of events begins at a time \( t = 0 \), and condition (2) can usually be directly verified from our knowledge of the process. However, it is not at all clear how we would determine that condition (3) is satisfied, and for this reason an equivalent definition of a Poisson process would be useful.

As a prelude to giving a second definition of a Poisson process, we shall define the concept of a function \( f \) being \( o(h) \).

**Definition 2.** The function \( f \) is said to be \( o(h) \) if

\[
\lim_{h \to 0} \frac{f(h)}{h} = 0.
\]

We are now in a position to give an alternative definition of a Poisson process.

**Definition 3.** The counting process \( \{N(t), t \geq 0\} \) is said to be a Poisson process having rate \( \lambda, \lambda > 0 \), if:

1. \( N(0) = 0 \).
2. The process has independent increments.
3. \( \mathbb{P}\{N(h) = 1\} = \lambda h + o(h) \).
4. \( \mathbb{P}\{N(h) \geq 2\} = o(h) \).

**Theorem 1.** Definitions 1 and 3 are equivalent.

**Proof.** We first show that Definition 3 implies Definition 1. To do this let \( P_0(t) = \mathbb{P}\{N(t) = 0\} \).

We derive a differential equation for \( P_0(t) \) in the following manner:

\[
P_0(t + h) = \mathbb{P}\{N(t + h) = 0\} = \mathbb{P}\{N(t) = 0, N(t + h) - N(t) = 0\} = \mathbb{P}\{N(t) = 0\} \mathbb{P}\{N(t + h) - N(t) = 0\} = \mathbb{P}\{N(t) = 0\} [1 - \lambda h + o(h)]
\]

where the final two equations follow from Assumption (2) and the fact that (3) and (4) imply that \( \mathbb{P}\{N(t) = 0\} = 1 - \lambda h + o(h) \). Hence,

\[
\frac{P_0(t + h) - P_0(t)}{h} = -\lambda P_0(t) + \frac{o(h)}{h}.
\]

Letting \( h \to 0 \) yields

\[
P_0'(t) = -\lambda P_0(t)
\]

or

\[
\frac{P_0'(t)}{P_0(t)} = -\lambda.
\]
which implies, by integration,
\[
\log P_0(t) = -\lambda t + c
\]
or
\[
P_0(t) = Ke^{-\lambda t}.
\]
Since \(P_0(0) = \Pr\{N(0) = 0\} = 1\), we arrive at
\[
P_0(t) = e^{-\lambda t}.
\] (4.9)

Similarly, for \(n \geq 1\),
\[
P_n(t+h) = \Pr\{N(t+h) = n\} = \Pr\{N(t) = n, N(t+h) - N(t) = 0\}
+ \Pr\{N(t) = n-1, N(t+h) - N(t) = 1\}
+ \sum_{k=2}^{n} \Pr\{N(t) = n-k, N(t+h) - N(t) = k\}.
\]
However, by (4), the last term in the above is \(o(h)\), hence, by using (2), we obtain
\[
P_n(t+h) = P_n(t)P_0(h) + P_{n-1}(t)P_1(h) + o(h)
= (1 - \lambda h)P_n(t) + \lambda hP_{n-1}(t) + o(h).
\]

Thus,
\[
\frac{P_n(t+h) - P_n(t)}{h} = -\lambda P_n(t) + \lambda P_{n-1}(t) + \frac{o(h)}{h}.
\]
Letting \(h \rightarrow 0\),
\[
P'_n(t) = -\lambda P_n(t) + \lambda P_{n-1}(t).
\]
or, equivalently,
\[
e^{\lambda t} [P'_n(t) + \lambda P_n(t)] = \lambda e^{\lambda t} P_{n-1}(t).
\]
Hence,
\[
\frac{d}{dt} (e^{\lambda t} P_n(t)) = \lambda e^{\lambda t} P_{n-1}(t). \tag{4.10}
\]
Now by 4.9 we have when \(n=1\)
\[
\frac{d}{dt} (e^{\lambda t} P_1(t)) = \lambda
\]
or
\[
P_1(t) = (\lambda t + c)e^{-\lambda t},
\]
which since \(P_1(0) = 0\), yields
\[
P_1(t) = \lambda te^{-\lambda t}.
\]
To show that \(P_n(t) = e^{-\lambda t}(\lambda t)^n/n!\), we use mathematical induction and hence first assume if for \(n - 1\). Then by 4.10,
\[
\frac{d}{dt} (e^{\lambda t} P_n(t)) = \frac{\lambda(\lambda t)^{n-1}}{(n-1)!}
\]
implying that
\[ e^{\lambda t} P_n(t) = \frac{(\lambda t)^n}{n!} + c, \]
or, since, \( P_n(0) = \mathbb{P}\{N(0) = n\} = 0, \)
\[ P_n(t) = e^{-\lambda t} \frac{(\lambda t)^n}{n!}. \]
Thus Definition 3 implies Definition 1. The proof of the reverse is left as an exercise.

### 4.4.2. Interarrival and waiting time distribution

Consider a Poisson process, and let \( X_1 \) denote the time of the first event. Further, for \( n \geq 1 \), let \( X_n \) denote the time between the \((n-1)\)st and the \( n \)th event. The sequence \( \{X_n, n \geq 1\} \) is called the sequence of interarrival times.

We shall now determine the distribution of the \( X_n \). To do so we first note that the event \( \{X_1 > t\} \) takes place if, and only if, no events of the Poisson process occur in the interval \([0, t]\), and thus
\[ P\{X_1 > t\} = P\{N(t) = 0\} = e^{-\lambda t}. \]
Hence, \( X_1 \) has an exponential distribution with mean \( 1/\lambda \). To obtain the distribution of \( X_2 \) condition on \( X_1 \). This gives
\[ P\{X_2 > t \mid X_1 = s\} = P\{0 \text{ events in } (s, s+t] \mid X_1 = s\} = e^{-\lambda t} \]
Therefore, from the above we conclude that \( X_2 \) is also an exponential random variable with mean \( 1/\lambda \), and furthermore, that \( X_2 \) is independent of \( X_1 \). repeating the same argument yields the following.

**Proposition 1.** \( X_n, n = 1, 2, \ldots \) are independent identically distributed exponential random variables.

**Remark** The proposition should not surprise us. The assumption of stationary and independent increments is equivalent to asserting that, at any point in time, the process probabilistically restarts itself. That is, the process from any point on is independent of all that previously occurred (by independent increments), and also has the same distribution as the original process (by stationary increments). In other words, the process has no memory, and hence exponential interarrival times are to be expected.

Another quantity of interest is \( S_n \), the arrival time of the \( n \)th event, also called the waiting time until the \( n \)th event. Since
\[ S_n = \sum_{i=1}^{n} X_i, \quad n \geq 1, \]
it is easy to show, using moment generating functions, that Proposition 1 implies that \( S_n \) has a gamma distribution with parameters \( n \) and \( \lambda \). That is, its probability density is

\[
f(t) = \lambda e^{-\lambda t} \frac{(\lambda t)^{n-1}}{(n-1)!}, \quad t \geq 0.
\]

The above could also have been derived by noting that the \( n \)th event occurs prior or at time \( t \) if, and only if, the number of events occurring by time \( t \) is at least \( n \). That is,

\[ N(t) \geq n \Leftrightarrow S_n \leq t. \]

Hence,

\[
P\{S_n \leq t\} = P\{N(t) \geq n\} = \sum_{j=n}^{\infty} e^{-\lambda t} \frac{(\lambda t)^j}{j!},
\]

which upon differentiation yields that the density function of \( S_n \) is

\[
f(t) = -\sum_{j=n}^{\infty} \lambda e^{-\lambda t} + \sum_{j=n}^{\infty} \lambda e^{-\lambda t} \frac{(\lambda t)^{j-1}}{(j-1)!} = \lambda e^{-\lambda t} \frac{(\lambda t)^{n-1}}{(n-1)!}
\]

Remark Another way of obtaining the density of \( S_n \) is to use the independent increment assumption as follows:

\[
P\{t < S_n < t + dt\} = P\{N(t) = n - 1, 1 \text{ event in } (t, t + dt)\} + o(dt) = P\{N(t) = n - 1\} P\{1 \text{ event in } (t, t + dt)\} + o(dt) = e^{-\lambda t} \frac{(\lambda t)^{n-1}}{(n-1)!} \lambda dt + o(dt)
\]

which yields, upon dividing by \( dt \) and then letting it approach 0, that

\[
f_{S_n} = \lambda e^{-\lambda t} \frac{(\lambda t)^{n-1}}{(n-1)!}.
\]

Proposition 1 also gives us another way of defining a Poisson process. For suppose that we start out with a sequence \( \{X_n, n \geq 1\} \) of independent and identically distributed exponential variables each having mean \( 1/\lambda \). Now let us define a counting process by saying that the \( n \)th event of this process occurs at time \( S_n \), where

\[ S_n = X_1 + X_2 + \cdots + X_n. \]

The resultant counting process \( \{N(t), t \geq 0\} \) will be a Poisson process with rate \( \lambda \).
4.4.3. Conditional distribution of the arrival times

Suppose we are told that exactly one event of a Poisson process has taken place by time $t$, and we are asked to determine the distribution of the time at which the event occurred. Since a Poisson process possesses stationary and independent increments, it seems reasonable that each interval in $[0, t]$ of equal length should have the same probability of containing the event. In other words, the time of the event should be uniformly distributed over $[0, t]$. This is easily checked since, for $s \leq t$,

$$\Pr \{ X_1 < s \mid N(t) = 1 \} = \frac{\Pr \{ X_1 < s, N(t) = 1 \}}{\Pr \{ N(t) = 1 \}} = \frac{\Pr \{ 1 \text{ event in } [0, s), 0 \text{ event in } [s, t) \}}{\Pr \{ N(t) = 1 \}} = \frac{\Pr \{ 1 \text{ event in } [0, s) \} \Pr \{ 0 \text{ event in } [s, t) \}}{\Pr \{ N(t) = 1 \}} = \frac{\lambda s e^{-\lambda s} e^{-\lambda (t-s)}}{\lambda t e^{-\lambda t}} = \frac{s}{t}.$$

This result may be generalized, but before doing so we need to introduce the concept of order statistics.

Let $Y_1, Y_2, \ldots, Y_n$ be $n$ random variables. We say that $Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)}$ are the order statistics corresponding to $Y_1, Y_2, \ldots, Y_n$ if $Y_{(k)}$ is the $k$th smallest value among $Y_1, Y_2, \ldots, Y_n$. If the $Y_i$’s are independent and identically distributed random variables with probability density $f$, then the joint density of the order statistics $Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)}$ is given by

$$f(y_1, y_2, \ldots, y_n) = n! \prod_{i=1}^{n} f(y_i), \quad y_1 < y_2 < \cdots < y_n.$$

The above follows since (i) $(Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)})$ will equal $(y_1, y_2, \ldots, y_n)$ if $(Y_1, Y_2, \ldots, Y_n)$ is equal to any of the $n!$ permutations of $(y_1, y_2, \ldots, y_n)$ and (ii) the probability density that $(Y_1, Y_2, \ldots, Y_n)$ is equal to $y_1, y_2, \ldots, y_n$ is $f(y_1) f(y_2) \cdots f(y_n) = \prod_{i=1}^{n} f(y_i)$ when $(y_1, y_2, \ldots, y_n)$ is a permutation of $(y_1, y_2, \ldots, y_n)$.

If the $Y_i$, $i = 1, \ldots, n$, are uniformly distributed over $(0, t)$, then it follows from the above that the joint density function of the order statistics $Y_{(1)}, Y_{(2)}, \ldots, Y_{(n)}$ is

$$f(y_1, y_2, \ldots, y_n) = \frac{n!}{t^n}, \quad 0 < y_1 < y_2 < \cdots < y_n < t.$$

We are now ready for the following useful theorem.

**Theorem 2.** Given that $N(t) = n$, the $n$ arrival times $S_1, \ldots, S_n$ have the same distribution as the order statistics corresponding to $n$ independent random variables uniformly distributed on the interval $(0, t)$. 

56
Proof. We shall compute the conditional density function of \(S_1, \ldots, S_n\) given that \(N(t) = n\). So let \(0 < t_1 < t_2 < \ldots < t_{n+1} = t\) and let \(h_i\) be small enough so that \(t_i + h_i < t_{i+1}, i = 1, \ldots, n\). Now,

\[
\mathbb{P}\{t_i \leq S_i \leq t_i + h_i, i = 1, 2, \ldots, n \mid N(t) = n\} = \frac{\mathbb{P}\{\text{exactly 1 event in } [t_i, t_i + h_i], i = 1, \ldots, n, \text{ no events elsewhere in } [0, t]\}}{\mathbb{P}\{N(t) = n\}} = \frac{\lambda h_1 e^{-\lambda h_1} \cdots \lambda h_n e^{-\lambda h_n} e^{-\lambda (t - h_1 - h_2 - \cdots - h_n)}}{e^{-\lambda t} (\lambda t)^n / n!} = \frac{n!}{h_1 h_2 \cdots h_n}.
\]

Hence,

\[
\mathbb{P}\{t_i \leq S_i \leq t_i + h_i, i = 1, 2, \ldots, n \mid N(t) = n\} = \frac{n!}{t^n} h_1 h_2 \cdots h_n,
\]

and by letting the \(h_i \to 0\), we obtain that the conditional density of \(S_1, \ldots, S_n\) given that \(N(t) = n\) is

\[
f(t_1, \ldots, t_n) = \frac{n!}{t^n}, \quad 0 < t_1 < \cdots < t_n,
\]

which completes the proof. \(\square\)

**Remark** Intuitively, we usually say that under the condition that \(n\) events have occurred in \((0, t)\), the times \(S_1, \ldots, S_n\) at which events occur, considered as unordered random variables, are distributed independently and uniformly in the interval \((0, t)\).

**Example 4.** Suppose that a neuron is subject to synaptic inputs from a large number of other neurons whose collective activity is approximated by a Poisson process having rate \(\lambda\). The \(i\)th presynaptic action potential gives rise to a somatic membrane potential change \(\Delta V_i\) in the neuron of interest. The \(\Delta V_i, i \geq 1\), are assumed to be independent and identically distributed and also to be independent of \(\{N(t), t \geq 0\}\), where \(N(t)\) denotes the number of presynaptic action potentials / spikes in \([0, t]\). The membrane potential change due to a synaptic input is assumed to decrease exponentially in time. That is, if a synaptic input gives an initial membrane potential change \(\Delta V\), then a time \(t\) later the remaining change is \(\Delta V e^{-\alpha t}\).

If we suppose that the membrane potential changes are additive, then \(\Delta V(t)\), the "voltage deviation" at \(t\), can be expressed as

\[
\Delta V(t) = \sum_{i=1}^{N(t)} \Delta V_i e^{-\alpha (t - S_i)},
\]

where \(S_i\) represents the arrival time of the \(i\)th synaptic input. We can now determine

Mean voltage deviation of a Poisson process synaptic input
\[ \mathbb{E} [\Delta V(t)] \text{ as follows:} \]

\[
\mathbb{E} [\Delta V(t) \mid N(t) = n] = \mathbb{E} \left[ \sum_{i=1}^{N(t)} \Delta V_i e^{-\alpha (t-S_i)} \mid N(t) = n \right] \\
= \mathbb{E} \left[ \sum_{i=1}^{n} \Delta V_i e^{-\alpha (t-S_i)} \mid N(t) = n \right] \\
= \sum_{i=1}^{n} \mathbb{E} [\Delta V_i e^{-\alpha (t-S_i)} \mid N(t) = n] \\
= \sum_{i=1}^{n} \mathbb{E} [\Delta V_i \mid N(t) = n] \mathbb{E} \left[ e^{-\alpha (t-S_i)} \mid N(t) = n \right] \\
= \mathbb{E} [\Delta V] \sum_{i=1}^{n} \mathbb{E} \left[ e^{-\alpha (t-S_i)} \mid N(t) = n \right] \\
= \mathbb{E} [\Delta V] \mathbb{E} \left[ \sum_{i=1}^{n} e^{-\alpha (t-S_i)} \mid N(t) = n \right] \\
= \mathbb{E} [\Delta V] e^{-\alpha t} \mathbb{E} \left[ \sum_{i=1}^{n} e^{\alpha S_i} \mid N(t) = n \right].
\]

Now letting \( U_1, \ldots, U_n \) be independent and identically distributed uniform \([0, t]\) random variables, then by Theorem 2,

\[
\mathbb{E} \left[ \sum_{i=1}^{n} e^{\alpha S_i} \mid N(t) = n \right] = \mathbb{E} \left[ \sum_{i=1}^{n} e^{\alpha U_i} \right] \\
= \mathbb{E} \left[ \sum_{i=1}^{n} e^{\alpha U_i} \right] \\
= \frac{n}{t} \int_{0}^{t} e^{\alpha x} \, dx \\
= \frac{n}{\alpha t} (e^{\alpha t} - 1).
\]

Hence,

\[
\mathbb{E} [\Delta V(t) \mid N(t)] = \frac{N(t)}{\alpha t} (1 - e^{-\alpha t}) \mathbb{E} [\Delta V]
\]

and, taking expectations,

\[
\mathbb{E} [\Delta V(t)] = \frac{\lambda \mathbb{E} [\Delta V]}{\alpha} (1 - e^{-\alpha t}).
\]

**Remark** Another approach to obtaining \( \mathbb{E} [\Delta V(t)] \) is to break the interval \((0, t)\) into nonoverlapping intervals of length \( h \) and then add the contributions at time \( t \) of inputs
originating in these intervals. More specifically, let $h$ be given and define $X_i$ as the sum of the potential deviations at time $t$ of all inputs arriving in the interval $I_i \equiv (ih, (i + 1)h)$, $i = 0, 1, \ldots, \lfloor t/h \rfloor$, where $\lfloor a \rfloor$ denotes the largest integer less than or equal to $a$. Then we have the representation

$$\Delta V(t) = \sum_{i=0}^{\lfloor t/h \rfloor} X_i,$$

and so

$$\mathbb{E}[\Delta V(t)] = \sum_{i=0}^{\lfloor t/h \rfloor} \mathbb{E}[X_i].$$

To compute $\mathbb{E}[X_i]$ condition on whether or not an input arrives in the interval $I_i$. This yields

$$\mathbb{E}[\Delta V(t)] = \sum_{i=0}^{\lfloor t/h \rfloor} \left( \lambda h \mathbb{E}[\Delta V e^{-\alpha(t-L_i)}] + o(h) \right),$$

where $L_i$ is the arrival time of the input in interval $I_i$. Hence,

$$\mathbb{E}[\Delta V(t)] = \lambda \mathbb{E}[\Delta V] \sum_{i=0}^{\lfloor t/h \rfloor} he^{-\alpha(t-L_i)} + \left[ \frac{t}{h} \right] o(h).$$

(4.11)

But since $L_i \in I_i$, it follows upon letting $h \to 0$ that

$$\sum_{i=0}^{\lfloor t/h \rfloor} he^{-\alpha(t-L_i)} \to \int_0^t e^{-\alpha(t-y)}dy = \frac{1 - e^{-\alpha t}}{\alpha},$$

and thus from (4.11) upon letting $h \to 0$

$$\mathbb{E}[\Delta V(t)] = \frac{\lambda \mathbb{E}[\Delta V]}{\alpha} (1 - e^{-\alpha t}).$$

It is worth noting that the above is a more rigorous version of the following argument: Since an input occurs in interval $(y, y+dy)$ with probability $\lambda dy$ and since its amplitude at time $t$ will equal $e^{-\alpha(t-y)}$ times its initial amplitude it follows that the expected voltage deviation at $t$ from inputs originating in $(y, y+dy)$ is

$$\lambda dy \mathbb{E}[\Delta V] e^{-\alpha(t-y)},$$

and so

$$\mathbb{E}[\Delta V(t)] = \lambda \mathbb{E}[\Delta V] \int_0^t e^{-\alpha(t-y)}dy$$

$$= \frac{\lambda \mathbb{E}[\Delta V]}{\alpha} (1 - e^{-\alpha t}).$$

59
As an important application of Theorem 2 suppose that each event of a Poisson process with rate $\lambda$ is classified as being either a type-I or type-II event, and suppose that the probability of an event being classified as type-I depends on the time at which it occurs. Specifically, suppose that if an event occurs at time $s$, then, independently of all else, it is classified as being a type-I event with probability $P(s)$ and a type-II event with probability $1 - P(s)$. By using Theorem 2 we can prove the following proposition

**Proposition 2.** If $N_1(t)$ represents the number of type-i events that occur by time $t$, $i = 1, 2$, then $N_1(t)$ and $N_2(t)$ are independent Poisson random variables having respective means $\lambda tp$ and $\lambda t(1 - p)$, where

$$p = \frac{1}{t} \int_0^t P(s) ds.$$

**Proof.** We compute the joint probability of $N_1(t)$ and $N_2(t)$ by conditioning on $N(t)$:

$$\mathbb{P} \{N_1(t) = n, N_2(t) = m\} = \sum_{k=0}^{\infty} \mathbb{P} \{N_1(t) = n, N_2(t) = m \mid N(t) = k\} \mathbb{P} \{N(t) = k\} = \mathbb{P} \{N_1(t) = n, N_2(t) = m \mid N(t) = n + m\} \mathbb{P} \{N(t) = n + m\}.$$

Now consider an arbitrary event that occurred in the interval $[0, t]$. If it had occurred at time $s$, the probability that it would be a type-I event would be $P(s)$. Hence, since by Theorem 2 this event will have occurred at some time uniformly distributed on $(0, t)$, it follows that the probability that it will be a type-I event is

$$p = \frac{1}{t} \int_0^t P(s) ds$$

independently of the other events. Hence, $\mathbb{P} \{N_1(t) = n, N_2(t) = m \mid N(t) = n + m\}$ will just equal the probability of $n$ successes and $m$ failures in $n + m$ independent trials when $p$ is the probability of success on each trial. That is,

$$\mathbb{P} \{N_1(t) = n, N_2(t) = m \mid N(t) = n + m\} = \binom{n + m}{n} p^n (1 - p)^m.$$

Consequently,

$$\mathbb{P} \{N_1(t) = n, N_2(t) = m\} = \frac{(n + m)!}{n!m!} p^n (1 - p)^m e^{-\lambda t (\lambda t)^{n+m}} = e^{-\lambda tp} \frac{\lambda tp^n}{n!} e^{-\lambda t(1 - p)} \frac{\lambda t(1 - p)^m}{m!}.$$

which completes the proof.
4.4.4. Nonhomogeneous Poisson process

In this section we generalize the Poisson process by allowing the arrival rate at time $t$ to be a function of $t$.

**Definition.** The counting process $\{N(t), t \geq 0\}$ is said to be a nonstationary or nonhomogeneous Poisson process with intensity function $\lambda(t), t \geq 0$ if:

1. $N(0) = 0$.
2. $\{N(t), t \geq 0\}$ has independent increments.
3. $P\{N(t + h) - N(t) \geq 2\} = o(h)$.
4. $P\{N(t + h) - N(t) = 1\} = \lambda(t)h + o(h)$.

If we let

$$m(t) = \int_0^t \lambda(t)dt,$$

it can be shown that

$$P\{N(t + s) - N(t) = n\} = \exp\left\{- (m(t + s) - m(t)) \right\} \frac{(m(t + s) - m(t))^n}{n!}, \quad n \geq 0. \quad (4.12)$$

That is $N(t + s) - N(t)$ is Poisson distributed with mean $m(t + s) - m(t)$.

The proof of (4.12) follows along the lines of the proof of Theorem 1 with a slight modification: Fix $t$ and define

$$P_n(s) = P\{N(t + s) - N(t) = n\}.$$

Then,

$$P_0(s + h) = P\{N(t + s + h) - N(t) = 0\} = P\{0 \text{ events in } (t, t + s), 0 \text{ events in } (t + s, t + s + h)\} = P\{0 \text{ events in } (t, t + s)\} P\{0 \text{ events in } (t + s, t + s + h)\} = P_0(s) \left[1 - \lambda(t + s)h + o(h)\right].$$

where the next-to-last equality follows from (2) and the last from (3) and (4). Hence,

$$\frac{P_0(s + h) - P_0(s)}{h} = -\lambda(t + s)P_0(s) + \frac{o(h)}{h}. \quad (4.12)$$

Letting $h \to 0$ yields

$$P_0'(s) = -\lambda(t + s)P_0(s)$$

or

$$\log P_0(s) = -\int_0^s \lambda(t + u)du.$$
or
\[ P_0(s) = e^{-(m(t+s)-m(t))}. \]
The remainder of the verification of (4.12) follows similarly and is left as an exercise.

The importance of the nonhomogeneous Poisson process resides in the fact that we no longer require stationary increments, and so we allow for the possibility that events may be more likely to occur at certain times than at other times.

When the intensity function \( \lambda(t) \) is bounded, we can think of the nonhomogeneous process as being a random sample from a homogeneous Poisson process. Specifically, let \( \lambda \) be such that
\[ \lambda(t) \leq \lambda \quad \text{for all } t \geq 0 \]
and consider a Poisson process with rate \( \lambda \). Now if we suppose that an event of the Poisson process that occurs at time \( t \) is counted with probability \( \lambda(t)/\lambda \), then the process of counted events is a nonhomogeneous Poisson process with intensity function \( \lambda(t) \). This last statement easily follows from Definition 4.4.4. For instance (1), (2), and (3) follow since they are also true for the homogeneous Poisson process. Point (4) follows since \( \Pr\{\text{one counted event in } (t, t+h)\} = \Pr\{\text{one event in } (t, t+h)\} \frac{\lambda(t)}{\lambda} + o(h) \)
\[ = \lambda h \frac{\lambda(t)}{\lambda} + o(h) \]
\[ = h \lambda(t) + o(h). \]

4.5. The renewal process

In the previous section we saw that the interarrival times for the Poisson process are independent and identically distributed exponential random variables. A natural generalization is to consider a counting process for which the interarrival times are independent and identically distributed with an arbitrary distribution. Such a counting process is called a renewal process.

Formally, let \( \{X_n, n = 1, 2, \ldots\} \) be a sequence of nonnegative independent random variables with a common distribution \( F \), and to avoid trivialities suppose that \( F(0) = \Pr\{X_n = 0\} < 1 \). We shall interpret \( X_n \) as the time between the \((n-1)\)st and the \( n \)th event. Let
\[ \mu = \mathbb{E}[X_n] = \int_0^\infty xdF(x) \]
denote the mean time between successive events and note that from the the assumptions that \( X_n \geq 0 \) and \( F(0) < 1 \), it follows that \( 0 < \mu \leq \infty \). Letting
\[ S_0 = 0, \quad S_n = \sum_{i=1}^n X_i, \quad n \geq 1, \]
it follows that \( S_n \) is the time of the \( n \)th event. As the number of events by time \( t \) will equal the largest value of \( n \) for which the \( n \)th event occurs before or at time \( t \), we have
that $N(t)$, the number of events by time $t$, is given by

$$N(t) = \sup\{n : S_n \leq t\}.$$  \hfill (4.13)

**Definition 4.** The counting process $\{N(t), t \geq 0\}$ is called a renewal process.
5. Simulating GL models and friends
Bibliography


Frederick J. Sigworth and Erwin Neher. “Single Na+ channel currents observed in cultured rat muscle cells”. In: Nature 287.5781 (Oct. 1980), pp. 447–449. URL: http://dx.doi.org/10.1038/287447a0.


L. M. Wahl, C. Pouzat, and K. J. Stratford. “Monte Carlo simulation of fast excitatory synaptic transmission at a hippocampal synapse”. In: Journal of Neurophysiology 75.2 (Feb. 1996), pp. 597–608. URL: http://dx.doi.org/10.1152/jn.1996.75.2.597.


A. Solutions to exercises

Most of these exercises use Python 3. If you’re not familiar with this language, you should start by reading the official tutorial (a masterpiece!). If you want to learn more, there are really a lot of good books around, my favorite is Mark Summerfield Programming in Python 3 [57].

Some optional excercices use C, more precisely interface C code with Python. To learn C properly check James Aspnes’ Notes on Data Structures and Programming Techniques. The coding style used in my solution is deliberately elementary and I avoid as much as I can the use of numpy and scipy. The concern here is twofold:

1. By coding things yourself instead of using blackbox functions, you are more likely to understand how things work.

2. Typically numpy codes break within 2 years while scipy ones can break after 6 months (because these huge libraries are evolving fast without a general concern for backward compatibility); by sticking to the standard library as much as I can, I hope that the following codes will run longer.

You are of course free to use more sophisticated libraries if you want to.

A.1. The inter mepp data of Fatt and Katz (1952)

A.1.1. Question 1: Getting the data

1. Command line based solution A first solution to get the data, for people who like the command line, is to use wget:

```
wget http://www.stat.cmu.edu/~larry/all-of-nonpar/~data/nerve.dat
```

I can check the content of the first 5 rows of the file with the head program as follows:

```
head -n 5 nerve.dat
```

```
0.21 0.03 0.05 0.11 0.59 0.06
0.18 0.55 0.37 0.09 0.14 0.19
0.02 0.14 0.09 0.05 0.15 0.23
0.15 0.08 0.24 0.16 0.06 0.11
0.15 0.09 0.03 0.21 0.02 0.14
```
Using `tail`, I can check the content of the last five rows as follows:

```
tail -n 5 nerve.dat
```

0.05 0.49 0.10 0.19 0.44 0.02
0.72 0.09 0.04 0.02 0.06
0.22 0.53 0.18 0.10 0.10 0.03
0.08 0.15 0.05 0.13 0.02 0.10
0.51

We see that the data are on 6 columns and that all the rows seem to be complete (it can be a good idea to check that with a pager like `less`) except the last one that has an element only in the first column (whose index is 0 in Python). The columns are moreover separated by tabs. One way to load the data in Python is:

```python
imepp = []
for line in open("nerve.dat"):
    elt = line.split('t')
    if elt[1] != '':  # the row is complete
        imepp += [float(x) for x in elt]
    else:  # last row with a single element
        imepp += [float(elt[0])]
len(imepp)
```

```
799
```

2. Pure Python solution

Instead of using the command line, we can use the `urllib.request` module of the standard library as follows:

```python
import urllib.request as url  #import the module
fh = url.urlopen("http://www.stat.cmu.edu/~larry/all-of-nonpar/=data/nerve.dat")
content = fh.read().decode("utf8")
imepp = [float(x) for x in content.split()]
len(imepp)
```

```
799
```

Here the `fh.read().decode("utf8")` is used because `fh` behaves as a opened file in binary mode. `content.split()` splits the string at every white space, line break, etc. The penultimate line makes use of a list comprehension.
A.1.2. Question 2: Basic statistics

The quick way to get the sample mean is to use function `mean` from the standard library module `statistics`.

```python
import statistics
statistics.mean(imepp)
```

0.21857321652065081

A slightly crude way to get the five-number summary is:

```python
[sorted(imepp)[i] for i in [round(k*(len(imepp)-1))
for k in [0,0.25,0.5,0.75,1]]]
```

[0.01, 0.07, 0.15, 0.3, 1.38]

This is 'crude' since we use a `round` function to get the index of the sorted data, namely, we get the first quartile by rounding 0.25x798 (=199.5) and we get 200 while, strictly speaking, we should take the sum of the sorted data at indices 199 and 200 and divide it by 2.

A.1.3. Question 3: Interval correlations

To plot interval $i+1$ vs interval $i$, we can simply do (we need to load `matplotlib`):

```python
import matplotlib.pyplot as plt
plt.plot(imepp[:-1],imepp[1:], 'o')
plt.xlabel("Interval i [s]")
plt.ylabel("Interval i+1 [s]")
plt.show()
```
To plot the ranks we first need to obtain them, that is, we need to replace each interval value by its rank in the sorted interval values. An elementary way of doing this job can go as follows:

1. We create tuples with the index and the value of each original interval and store all these tuples in a list (keeping the original order of course).

2. We sort the list of tuples based on the interval value.

3. We fill a rank vector by assigning the value of an increasing index (starting from 0 an increasing by 1) to the position specified by the first element of the sorted tuples.

We can first test the procedure with the example of the Wikipedia page: ‘3.4, 5.1, 2.6, 7.3’ should lead to ‘2, 3, 1, 4’ (with an index starting at 1)

```python
test_data = [3.4, 5.1, 2.6, 7.3]
# Point 1
i_test_data = [(i, test_data[i]) for i in range(len(test_data))]
# Point 2
s_i_test_data = sorted(i_test_data, key=lambda item: item[1])
# Start point 3
idx = 1
test_rank = [0 for i in range(len(test_data))]  # initialization
for elt in s_i_test_data:
    test_rank[elt[0]] = idx
    idx += 1
# End point 3
test_rank
```
Fine, we can now run it on our data (with an index starting at 0):

```python
i_imep = [(i, imepp[i]) for i in range(len(imepp))]  # Point 1
s_i_imep = sorted(i_imep, key=lambda item: item[1])  # Point 2
idx = 0  # Start point 3
rank = [0 for i in range(len(imepp))]  # initialization
for elt in s_i_imep:  # End point 3
    rank[elt[0]] = idx
    idx += 1
```

The plot gives:

```python
plt.plot(rank[:-1], rank[1:], 'o')
plt.xlabel("Rank of interval i [s]")
plt.ylabel("Rank of interval i+1 [s]")
plt.show()
```

Since the interval distribution is skewed to the right (it rises fast and decays slowly, like most duration distributions), plotting intervals against each other make a bad use of the plotting surface and leads to too many points in the lower left corner; structures can’t be seen anymore if some are there. Using the rank eliminates the ‘distortion’ due to the distribution and under the null hypothesis (no correlation), the graph should be filled essentially uniformly.
To compute the Spearman’s rank correlations, we first prepare a version of the ranks from which we subtract the mean rank and that we divide by the rank standard deviation (we can’t use the theoretical value for the latter since many intervals have the same value, due to truncation):

```python
rank_sd = statistics.stdev(rank)
rank_mean = statistics.mean(rank)
rank_normed = [(r-rank_mean)/rank_sd for r in rank]
```

We define a function `spearman` that returns the Spearman’s correlation coefficient for a given iterable containing normed data (mean 0 and SD 1) and a given lag:

```python
def spearman(normed_data, lag=1):
    """Returns Spearman's auto-correlation coefficient at a given lag.

    Parameters
    ----------
    normed_data: an iterable with normed data (mean 0 and SD 1)
    lag: a positive integer, the lag

    Returns
    ------
    The Spearman's auto-correlation coefficient at lag (a float)
    
    n = len(normed_data)-lag
    return sum([normed_data[i]*normed_data[i+lag] for i in range(n)])/n
```

We can then get the auto-correlation at the five different lags:

```python
rank_cor = [spearman(rank_normed, lag=k) for k in [1, 2, 3, 4, 5]]
rank_cor
```

```python
[-0.041293322480167906, 0.0310350375233785, 0.02987780737228069, -0.03737263560582801, 0.049288720472378866]
```

We now go ahead with the 1000 replicates and we approximate the distribution of the max of the five auto-correlation coefficients (in absolute value) under the null hypothesis (no correlation):

```python
import random
random.seed(20061001)  # set the seed
n_rep=1000
actual_max_of_five = max([abs(x) for x in rank_cor])  # observed value
# create a list that will store the statistics under the null
rho_max_of_five = [0 for i in range(n_rep)]
for i in range(n_rep):
    # shuffle the ranks (i.e. make the null true)
    samp = random.sample(rank_normed, len(rank_normed))
    # compute the statistics on shuffled ranks
```

75
\[
\begin{align*}
\rho_{\text{max}} &= \max(\{|x| \text{ for } x \text{ in } \text{spearman(samp, lag=k) for } k \text{ in } [1,2,3,4,5]}\)) \\
\rho_{\text{max}}^{\text{of five}}[i] &= \rho_{\text{max}} \\
\text{# store the result} \\
\text{sum([1 for rho in rho_{\text{max}}^{\text{of five}} if rho \neq \text{actual max of five}])/n_{\text{rep}}} &= 0.408
\end{align*}
\]

That's the p-value and we can conclude that the intervals behave as non-correlated intervals.

### A.1.4. Question 4: Counting Process Class

```python
class CountingProcess:
    """A class dealing with counting process sample paths""
    def __init__(self, times):
        from collections.abc import Iterable
        if not isinstance(times, Iterable):
            raise TypeError('
            times must be an iterable.
            
        n = len(times)
        diff = [times[i+1] - times[i] for i in range(n-1)]
        all_positive = sum([x > 0 for x in diff]) == n-1
        if not all_positive:
            raise ValueError('
            times must be increasing.
            
        self.cp = times
        self.n = n
        self.min = times[0]
        self.max = times[n-1]

    def sample_path(self, t):
        """Returns the number of observations up to time t""
        from bisect import bisect
        return bisect(self.cp, t)

    def plot(self, domain=None, color='black', lw=1):
        """Plots the sample path"
        import math
        import matplotlib.pyplot as plt
        n = self.n
        Y = [i for i in range(1, n+1)]
        if domain is None:
```

76
Here the code is written in a more 'robust' way than the previous ones. Upon creation of the instanciation of a class, that's what the call to the \_init\_ method is doing, checks are performed on the value passed as a parameter. The collections.abc module is used since it contains the definition of the Iterable type allowing us to check against any of the actual iterable types (list, arrays, tuple, etc.) in a single code line. We also check that the elements of the iterable are increasing.

The sample\_path method is just a disguised call to function bisect of module bisect. The call here is to find the index of the leftmost time (element of the original times parameter passed to \_init\_) smaller of equal to the value passed as sample\_path parameter.

The plot method generates a proper graphical representation of a step function, that is one without vertical lines.

Let us see how that works; to that end we must first create a time sequence out of the interval sequence:

```python
n = len(imepp)
cp = [0 for i in range(n)]
cp[0] = imepp[0]
for i in range(1,n): cp[i] = imepp[i]+cp[i-1]
cp1 = CountingProcess(cp)
print("The number of events is: \(n\)\n    "The first event occurs at: \(t_1\)\n    "The last event occurs at: \(t_n\).format(cp1.n,
cp1.min,
    round(cp1.max,2)))
```

The number of events is: 799.
The first event occurs at: 0.21.
The last event occurs at: 174.64.

Let us check that the parameter value passed upon initialization works properly. First with a non-iterable:

```python
cp_test = CountingProcess(10)
```

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>

77
Next with an iterable that is not increasing:

cp_test = CountingProcess([1, 2, 4, 3])

Then with an increasing iterable that is not a list but an array:

```python
import array
cp_test = CountingProcess(array.array('d', [1, 2, 3, 4, 5]))
print("The number of events is: \(0\).\n" "The first event occurs at: \(1\).\n" "The last event occurs at: \(2\).\n".format(cp_test.n, cp_test.min, cp_test.max))
```

The number of events is: 5.
The fist event occurs at: 1.0.
The last event occurs at: 5.0.

We can call the sample_path method as follows:

cp1.sample_path(100.5)

453

So 453 events were observed between 0 and 100.5.
The plot method leads to, guess what, a plot (assuming the import matplotlib.pylab as plt as already been issued):

```python
cp1.plot()
plt.show()
```
A.1.5. Question 5: Graphical test of arrival times 'uniformity'

We proceed as follows using the critical values [5, p. 86] considering one important subtlety: on the sample path representation, the maximum value reached on the right of the graph is not 1 as it is for a proper (cumulative) distribution function but \( n \), the number of events; in other words the \( y \) axis is scales by a factor \( n \) compared to the ‘normal’ Kolmogorov’s test setting, that is why we multiply by \( \sqrt{n} \) instead of dividing by it in the following code.

```python
import math  # to get access to sqrt
cp1.plot(lw=0.2)
delta95 = 1.36*math.sqrt(cp1.n)
delta99 = 1.63*math.sqrt(cp1.n)
plt.plot([cp1.min,cp1.max],
         [1+delta95,cp1.n+delta95],
         lw=0.5,ls='-', color='red')  # upper 95% limit in red and continuous
plt.plot([cp1.min,cp1.max],
         [1-delta95,cp1.n-delta95],
         lw=0.5,ls='-', color='red')
plt.plot([cp1.min,cp1.max],
         [1+delta99,cp1.n+delta99],
         lw=0.5,ls='--', color='red')  # upper 99% limit in red and dashed
plt.plot([cp1.min,cp1.max],
         [1-delta99,cp1.n-delta99],
         lw=0.5,ls='--', color='red')
plt.xlabel("Time [s]")
plt.ylabel("Number of events")
plt.show()
```
A.1.6. Question 6: Kolmogorov’s test via C code interfacing

We start by downloading the code and unzipping it in a subdirectory named code:

```bash
  cd code && wget https://www.jstatsoft.org/index.php/jss/article/downloadSuppFile/v008i18/k.c.zip
  unzip k.c.zip
```

We then compile the code as a share library:

```bash
  cd code && gcc -shared -O2 -o k.so -fPIC k.c
```

In our Python session (assumed to be running 'above' directory code) we import ctypes and 'load' the shared library:

```python
  import ctypes as ct
  _klib = ct.cdll.LoadLibrary("./code/k.so")
```

We now have to look at the signature of the function we want to call (you have to open the C source file k.c for that and you will see: double K(int n, double d). If you read the documentation and you should always do that you see that n is the sample size and d the Kolmogorov’s statistics. But what concerns us now is the type of the parameters and of the return value. If the equivalent Python types are not specified, Python assumes essentially that everything is an integer and that’s not the case for d and for the returned value that are both of type double. We therefore do:

```python
  _klib.K.argtypes = [ct.c_int, ct.c_double]
  _klib.K.restype = ct.c_double
```
Now we check that it works as it should by gettind the probability for the Kolmogorov's statistics to be smaller or equal to 0.274 when the sample size is 10 [40, p. 2]

```
_klib.K(10,0.274)
```

0.6284796154565043

We can now go ahead and get the Kolmogorov's statistics for our sample. If you look at the graph of a counting process sample path, you will see that the maximal difference with the straight line going from the origin to the last observed event necessarily occurs at the arrival times (draw it if necessary to convince yourself that such is the case). So lets us get this maximal difference:

```
K_stat = max((Dplus+Dminus)/cp1.n)
```

0.025477047867323424

And the p-value is:

```
_klib.K(cp1.n,K_stat)
```

0.33228272454713365

We can therefore safely keep the null hypothesis: the data behave as if generated from a stationary Poisson process.

### A.1.7. Question 7: Figure 12 reproduction and more

Reproduce figure 12 is straightforward:

```
X = sorted(imepp)
Y = [i for i in range(1,len(imepp)+1)]
tau_hat = (cp1.max-cp1.min)/cp1.n
xx = [i+1.5/500 for i in range(501)]
yy = [(1-math.exp(-xx[i]/tau_hat))*cp1.n for i in range(len(xx))]
plt.plot(X,Y,"o",color='black')
plt.plot(xx,yy,color='red')
plt.xlabel("Inter mepp interval [s]")
plt.ylabel("Number of events")
plt.show()
```
The graph of the complementary function with a log scale on the ordinate is obtained with:

```
Y_C = [(cp1.n+1)-i)/cp1.n for i in range(1,len(imepp)+1)]
plt.plot(X,Y_C,'o',color='black')
plt.xlabel("Inter mepp interval [s]")
plt.ylabel("Complementary distribution function")
plt.yscale('log')
plt.show()
```
It is a straight line as it should be for data from an exponential distribution.
List of Figures

1.1. Left: opened head and brain; center: antennal lobe and recording probe (the two shanks with 8 bright spots on each); right: 1 sec recorded from a single tetrode, a group of 4 closely spaced recording sites; the data were filtered between 300 Hz and 15 kHz (see [46] for details) .... 4

1.2. Schistocerca americana .......................................................... 5

1.3. Raster plots from one of the neurons showing the responses to 25 presentations (gray background) of cis-3-hexen-1-ol (a typical ‘green’ odor; that’s what you can smell when grass has just been cut) .... 6

1.4. The sample paths of the counting processes associated with the 25 spike trains just shown. In red, the empirical mean of the 25 .... 7

2.1. Figures 1 and 2 of Sigworth and Neher (1980). The first single channel recordings from sodium channels [54]. Left, experimental setting. Right, single channel currents during 10 mV depolarizations: a, imposed patch membrane potential (40 mV pulse amplitude), one pulse was applied every second; b, average of a set of 300 responses; c, nine successive individual records .... 9

2.2. Montage of Figures 1, 2 and 3 of Verveen and Derksen [58] .... 10

2.3. Montage of Figures 1A (left) and 9C (right) of [21]. Top left: spontaneous activity recorded intracellularly from a frog muscle fiber close to the end-plate. Bottom left: response to a presynaptic nerve stimulation recorded from the same location (lower magnification on the y scale). Top right trace: 50 Hz oscillation giving a time reference. Middle right 3 traces: spontaneous potentials (like on the left side). Bottom right trace: superposed evoked responses recorded in low calcium; the presynaptic stimulation time is indicated by the arrow .... 11

2.4. Figure 12 of [21]. Empirical and fitted distribution functions of the inter mepp intervals. The dots are the data and the continuous line the fitted exponential distribution function. A series of 800 mepp was used, the shortest interval that could be resolved was 5 ms, the mean interval was 221 ms .... 12

2.5. mepp ................................................................. 13
2.6. Synaptic transmission fluctuation observed at an autapse (a synapse made by the neuron onto itself) of a molecular layer interneuron of the cerebellar cortex [45]. Aa top, imposed membrane voltage time course. Aa bottom, mean current responses; the continuous line shows the overall mean, while the dotted line shows the mean of responses classified as ‘failures’. The synaptic response starts slightly more than 2 ms after the end of the voltage pulse. Ab, the overall mean - the failure mean (that’s a way to get rid of the action current appearing during the pulse and due to the unclamped action potential). Ac, standard deviation time courses: continuous line from all responses, dotted line from ‘failures’. B, Individual responses after subtraction of failure mean. C, histograms of the response amplitudes and of their latency (inset).

2.7. Scheme of a central nervous system synapse. Source Wikimedia Commons.

3.1. Figure 5 of [32]: Frequency of maintained discharge during changes of illumination. Off-center unit recorded with indium electrode. No permanent change of frequency resulted from change of illumination.

3.2. Figure 6 of [32].

3.3. gamma

3.4. Figure 1a of [22].

3.5. Figure 1b of [22].

3.6. Figure 3 of [24]. Typical random walks in one dimension in a computer simulation of the model. [The ‘threshold’ is named ‘absorber’ on this figure. A spike is generated when the sample path of a random walk reaches the threshold/absorber. The ‘reflector’ set here at -128 does not change significantly the FPT distribution as long that the difference between threshold and reset, here 32, is large compared to the difference between reset and reflector; but for the graphical display it forces all the trajectories to stay within the graph.]

3.7. Figure 5, 6 and 7 of [24]. Histogram based FPT density estimations (dots) and fitted inverse-Gaussian density (continuous lines). With \( a = \lambda/2 \) and \( b = \lambda/(2\mu^2) \) in Eq. 3.6. Note the log ordinate scale.

3.8. Part of figure 9 of [24]. Left, interval histograms for the time-variant random walk model. Middle, model boundaries. Right, time course of the drift parameter corresponding to 1 stimulus per 1280 steps; values for each portion are given in the form: duration, drift velocity towards absorber.

3.9. Figure 2 of [14]. Part of the kernel used in simulation. Kernels are defined as piecewise functions that can be non null only between 0 and 640 ms. Each piece covers 10 ms; there are therefore 64 parameters (amplitudes) per kernel.

3.10. Figure 3 a, c and e of [14]. a, autocorrelation function of the presynaptic neuron, displaying ‘over-clustering’. c, estimated kernel with the method of moments. e, estimated kernel with the proposed method.
3.11. Figure 3 b, d and f of [14]. b, autocorrelation function of the presynaptic neuron, displaying 'under-clustering'. d, estimated kernel with the method of moments. f, estimated kernel with the proposed method. 33

3.12. Figure 4 of [14]. 34

3.13. Figure 5 of [14]. 34

3.14. Figure 6 of [14]. 35

4.1. A sample path of $X(t) = \text{number of events in } [0, t]$. 50
Index

action potential, 3
antennal lobe, 3
Bernoulli trial, 35
binary regression, 35
binary vector, 35
brain slice, 14
Brownian motion process, 26, 27
central nervous system, 13
characteristic function, 42
CNS, 13
continuous random variable, 38
counting process, 5, 29, 50
covariance, 39
discrete random variable, 38
distribution function, 20, 38
estimator, 29
event, 37
expectation, 39
Expectation-maximization, 31
exponential distribution, 46
failure rate function, 47
first hitting time, 26
first passage time, 26
ganglion cell, 22
generalized linear model, 35
hazard rate function, 47
independent increments, 49, 50
independent random variables, 39
index set, 49
inference, 29
inverse Gaussian distribution, 27
joint characteristic function, 42
joint distribution function, 38
joint moment generating function, 42
Laplace transform, 43
local neuron, 4
mean, 39
memoryless, 46
miniature end-plate potentials, 9
mitral cell, 4
moment generating function, 40
neuromuscular junction, 9
nonhomogeneous, 60
nonstationary, 60
olfactory bulb, 4
order statistics, 55
piecewise constant, 31
point process, 29
probability density function, 38
projection neuron, 4
quantal neurotransmitter release, 12
quantal size, 14
random experiment, 37, 38
random variable, 38
random walk, 26
raster plot, 4, 29
release probability, 13
renewal process, 26, 61, 62
sample path, 5, 49
sample space, 37
sequence of interarrival times, 53
spike, 3
spike sorting, 4
spontaneous activity, 4
stationary increments, 49, 50
step function, 31
stimulus response, 4
stochastic process, 49
survival function, 38
synaptic vesicle, 10
tetrode, 3
variance, 39
waiting time, 53
Wald distribution, 27
Wiener Process, 26
Wiener process, 27