

tutorial_1_useful_tools

Quentin Le Roux

Question 1

Let us consider the model of i.i.d. interspike intervals.

1. Consider the parametric model with density $\theta e^{-\theta(x-\eta)} 1_{x \geq \eta}$, with θ and η positive unknown parameters. Compute the likelihood for n observations.

Given n observations X_1, \dots, X_n independent and identically distributed (assumption of interspike intervals) with $\forall i \in \{1, \dots, n\}, X_i \sim \theta e^{-\theta(x-\eta)} 1_{x \geq \eta}$, we can state the likelihood of such n observations as:

$$\begin{aligned} f_i(\theta, \eta) &= \theta e^{-\theta(x_i-\eta)} 1_{x_i \geq \eta} \quad (\text{with } f \text{ the density of } X_i) \\ \mathcal{L}(\theta, \eta) &= \prod_{i=1}^n f_i(\theta, \eta) \\ &= \prod_{i=1}^n \theta e^{-\theta(x_i-\eta)} 1_{\min(x_i) \geq \eta} \\ &= \theta^n e^{-\theta \sum_{i=1}^n (x_i-\eta)} 1_{\min(x_i) \geq \eta} \end{aligned}$$

We obtain:

$$\mathcal{L}_n(\theta, \eta) = \theta^n e^{-\theta \sum_{i=1}^n (x_i-\eta)} 1_{\min(X) \geq \eta}$$

2. Simulate $n = 10$ i.i.d ISI with this model. For fixed θ , plot the likelihood as a function of η .

We first declare a set of functions to help up answer the question.

```
isi_generate <- function(n, theta, eta) {
  # Generates a number n of ISI observations
  # with parameters theta and eta with eta a time
  # delay starting from 0
  rexp(n, theta) + eta
}

isi_distribution_plot <- function(observations, theta, eta) {
  # Plots the distribution of ISI observations as
  # an histogram
  hist(observations,
       main = paste("Distribution of ISI given Theta=", theta, ", Eta=", eta),
       xlim = c(0, max(simulations)+1))
}

isi_likelihood <- function(observations, theta, eta){
  # Computes the likelihood of a set of ISI observations
  check_min = min(observations) >= eta
  if (check_min) {
    theta^n * exp(-theta * sum(observations - eta))
  } else {0}
}

isi_likelihood_plot_given_theta <- function(observations, theta) {
  # Given a fixed theta and observations provided as function
  # arguments, computes the likelihood of the observations over a
  # range of eta parameters and plot the resulting likelihoods
  etas = seq(0, 10, length.out=50)
  likelihoods = rep(0, 50)
  for (i in 1:50) {
    likelihoods[i] = isi_likelihood(observations, theta, etas[i])
  }
  plot(etas, likelihoods, pch = 1,
       xlab = "Eta", ylab="Likelihood",
       main = paste("Likelihood of observations with fixed Theta=", theta, "\n given Eta ranging from 1 to 10"))
}
```

We decide to randomly generate 10 observations and parameters $\theta = 1$ and $\eta = 2$.

```

n = 10
theta = 1
eta = 2

observations = isi_generate(n, theta, eta)

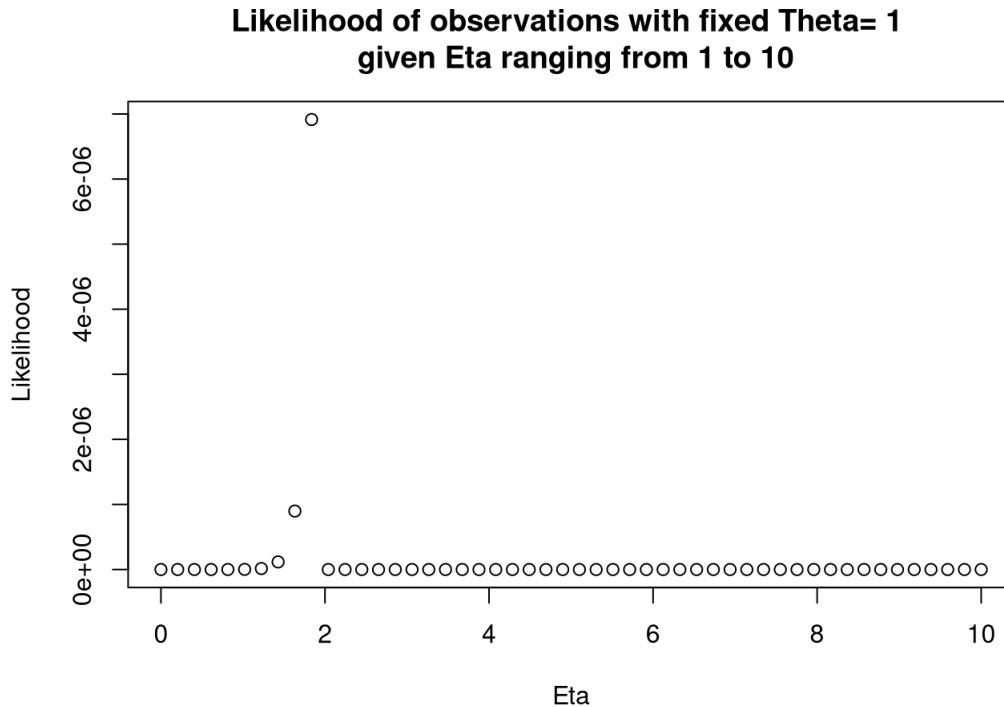
```

Now, we plot the likelihood of those 10 randomly generated observations given a varying η value (we set the range of η from 0 to 10):

```

isi_likelihood_plot_given_theta(observations, theta)

```



3. Find the MLE $\hat{\eta}$ by looking at the formula, after seeing what happened on the plot. Then plug the estimator $\hat{\eta}$ in the formula of the likelihood, take the logarithm and find the MLE $\hat{\theta}$.

By looking at the formula and seeing the plot, and knowing that $\min(X_i) \geq \eta$ in order to have observations fitting an interspike interval situation, we can determine that the maximum likelihood estimator $\hat{\eta}$ of η is:

$$\hat{\eta} = \min(X)$$

i.e. the MLE $\hat{\eta}$ is the minimum value among our observations X_i .

$$\begin{aligned}
\mathcal{L}(\theta) &= \theta^n e^{-\theta \sum_{i=1}^n (x_i - \hat{\eta})} \\
&= \theta^n e^{-\theta \sum_{i=1}^n (x_i - \min(X))} \\
l(\theta) &= \log(\mathcal{L}(\theta)) \\
&= n * \log(\theta) - \theta * \sum_{i=1}^n (x_i - \min(X))
\end{aligned}$$

We now have the formula for the log-likelihood. As such, we are looking for the MLE $\hat{\theta}$ such that:

$$\begin{aligned}
\hat{\theta} &= \underset{\theta \in \Theta}{\operatorname{argmax}} l \\
&\Rightarrow \frac{\partial l}{\partial \theta}(\theta) = 0 \\
&\Rightarrow \frac{n}{\theta} - \sum_{i=1}^n (x_i - \min(X)) = 0 \\
&\Rightarrow \frac{1}{\theta} = \bar{X} - \min(X)
\end{aligned}$$

As such, we can deduce the MLE $\hat{\theta}$ such that:

$$\hat{\theta} = \frac{1}{\bar{X} - \hat{\eta}} = \frac{1}{\bar{X} - \min(X)}$$

with \bar{X} the sample mean.

4. Implement the estimate in practice and show on several choices of η and θ that the MLE converge towards the true value when n tends to infinity.

We declare the functions allowing us to compute the convergence of the estimators over a range of observations from 2 to 5000 in steps of 10.

```
compute_MLE_eta <- function(observations) {
  min(observations)
}

compute_MLE_theta <- function(observations) {
  1/(mean(observations)-min(observations))
}

compute_convergence_estimators <- function(eta, theta){
  # Computes the MLE of parameters eta and theta of an
  # ISI model over the range 2, 5000 by steps of 10
  n = seq(2, 5000, 10)
  estimate_eta = rep(0, length(n))
  estimate_theta = rep(0, length(n))
  for (i in 1:length(n)){
    observations = isi_generate(n[i], theta, eta)
    estimate_eta[i] = compute_MLE_eta(observations)
    estimate_theta[i] = compute_MLE_theta(observations)
  }
  return(list("n" = n, "etas"=estimate_eta, "thetas"=estimate_theta))
}

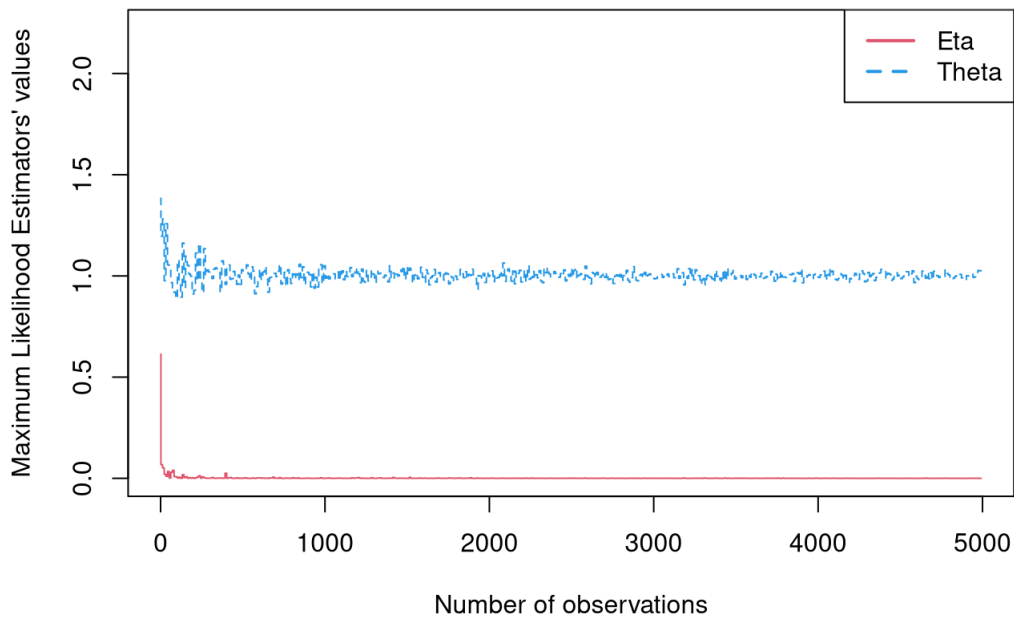
convergence_plot <- function(eta, theta) {
  # Plots the parameter convergence
  estimators = compute_convergence_estimators(eta, theta)
  computed_estimators = c(estimators$etas, estimators$thetas)
  n = length(estimators$n)
  ymin = max(0, min(computed_estimators)-1)
  ymax = max(estimators$etas[n], estimators$thetas[n])+min(10, max(c(eta, theta))*1.2)
  plot(estimators$n, estimators$etas, type="S", lty=1, ylim=c(ymin, ymax),
    main = paste("MLE convergence plot for the ISI model with\n Eta=",
      eta, "and Theta=", theta),
    xlab = "Number of observations",
    ylab = "Maximum Likelihood Estimators' values",
    col = 2)
  lines(estimators$n, estimators$thetas, lty=2, type="S", col=4)
  legend(x = "topright", legend = c("Eta", "Theta"), col=c(2,4),
    lty = c(1, 2), lwd = 2)
}
```

We compute the MLE $\hat{\eta}$ and $\hat{\theta}$ for each time step and plot the results for the following few select choices of η and θ :

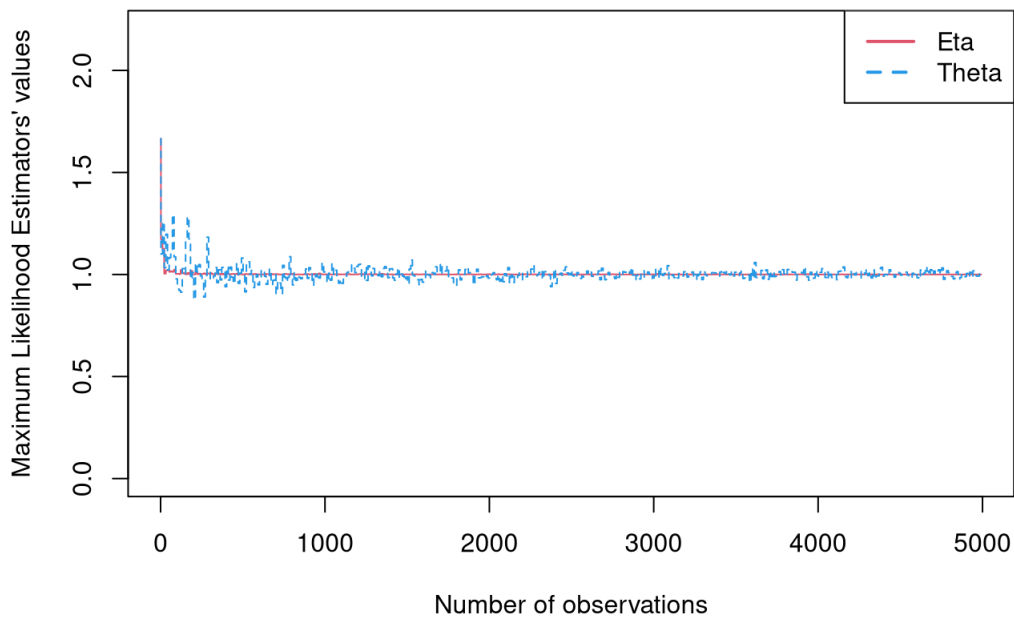
η	θ
0	1
1	1
2	1
1	2
5	5
10	20

```
parameter_mix = matrix(c(0,1,2,1,5,10,1,1,1,2,5,20),nrow=6)
for (i in 1:6){
  eta = parameter_mix[i,1]
  theta = parameter_mix[i,2]
  convergence_plot(eta, theta)
}
```

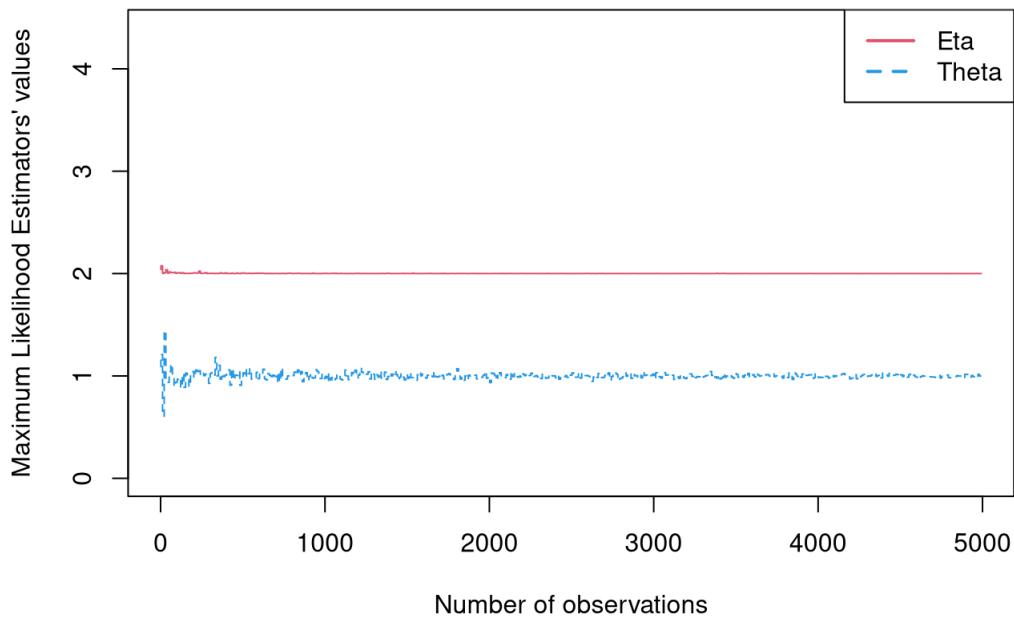
**MLE convergence plot for the ISI model with
Eta= 0 and Theta= 1**



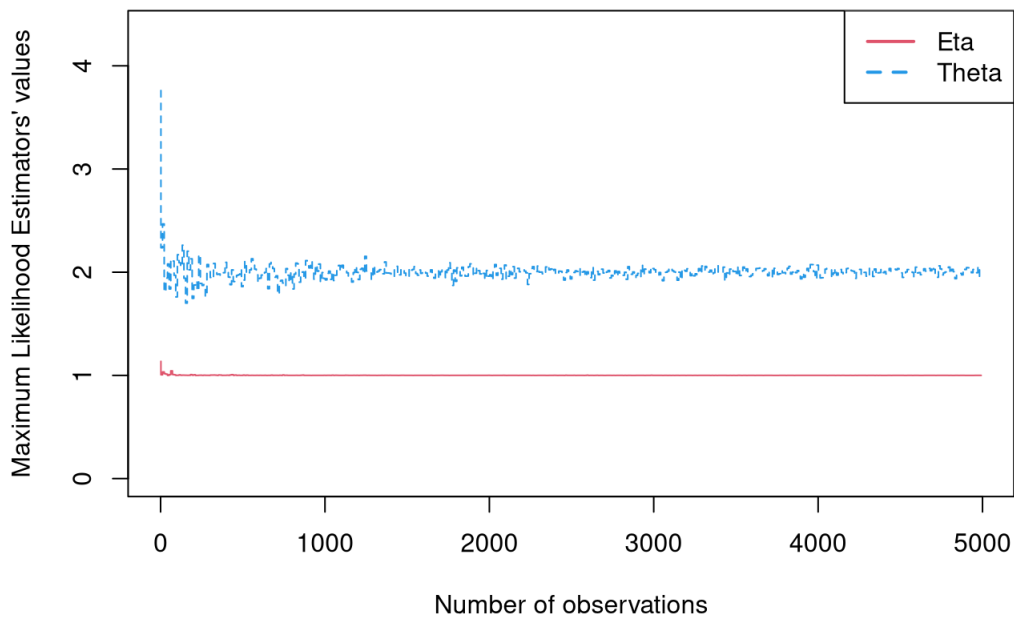
**MLE convergence plot for the ISI model with
Eta= 1 and Theta= 1**



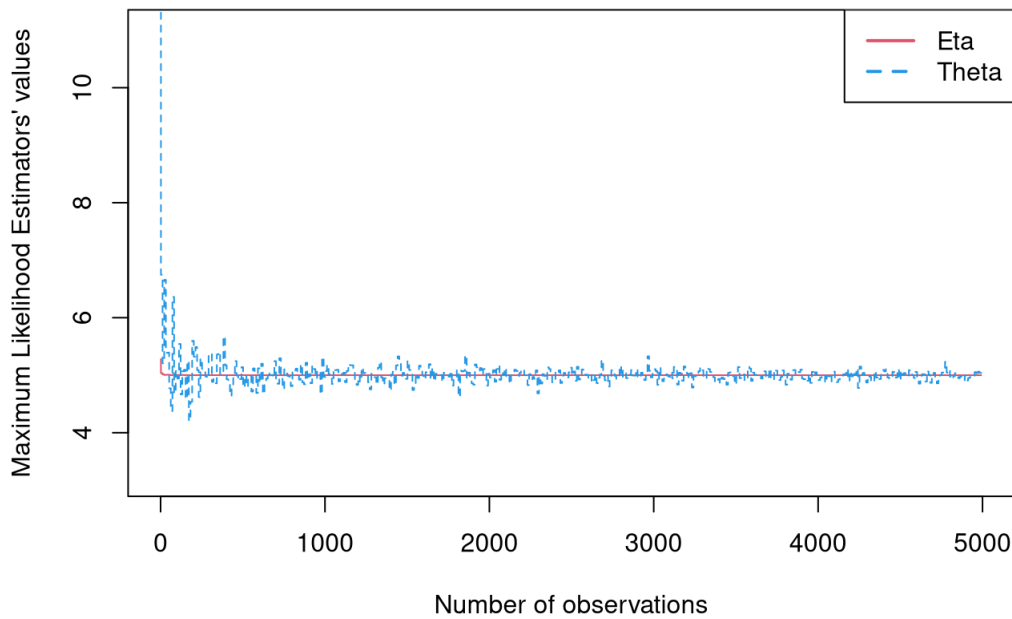
**MLE convergence plot for the ISI model with
Eta= 2 and Theta= 1**



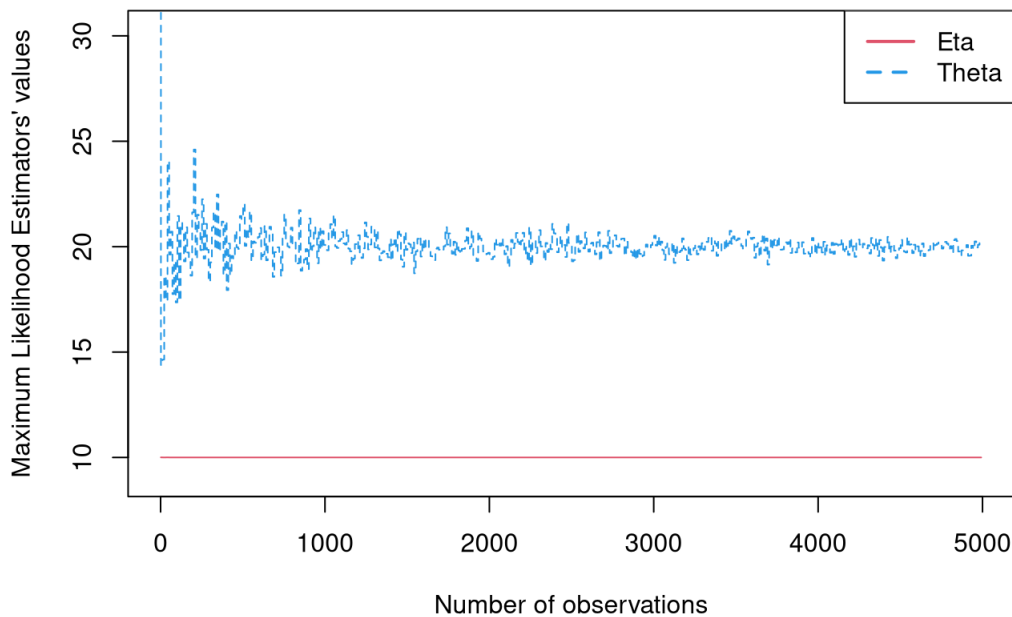
**MLE convergence plot for the ISI model with
Eta= 1 and Theta= 2**



MLE convergence plot for the ISI model with Eta= 5 and Theta= 5



MLE convergence plot for the ISI model with Eta= 10 and Theta= 20



We confirm empirically that we see convergence.

5. Load the R package STAR with the CRAN site and install it. Look at the documentation to see what are the data `cockroachAIData`. Try the MLE on various neurons on various time windows (during stimulation before, after ...) NB : `do data(e070528citronellal)` and `as.vector(e070528citronellal[["neuron 1"]][[1]])` will give you as a vector, the time of the action potentials for the experiment "citronellal", on neuron 1, for the 1st trial

We start by installing the package and loading it. The documentation is available on CRAN (<https://www.rdocumentation.org/packages/STAR/versions/0.3-7>).

```
#install.packages("STAR")
library(STAR)
```

The `cockroachAIData` (<https://www.rdocumentation.org/packages/STAR/versions/0.3-7/topics/cockroachAIData>) contains data records of neuron activity such that:

Four (CAL1S and CAL1V), three (CAL2S and CAL2C), three (e060517spont and e060517ionon), three (e060817spont, e060817terpi, e060817citron and e060817mix), two (e060824spont and e060824citral) and four (e070528spont and e070528citronellal) Cockroach (*Periplaneta americana*) antennal lobe neurons (putative projection neurons) were recorded simultaneously and extracellularly during spontaneous activity and odors (vanilin, citral, citronellal, terpineol, beta-ionon) responses from six different animals. The data sets contain the sorted spike trains of the neurons

We mostly focus on the 4 neurons listed in `e070528citronellal`, i.e., when the odor was *citronellal*. We will also look into the first neuron of CAL1V :

```
data(e070528citronellal)
data(CAL1S)

neuron1_first_trial = as.vector(e070528citronellal$`neuron 1`$`stim. 1`)
neuron2_first_trial = as.vector(e070528citronellal$`neuron 2`$`stim. 1`)
neuron3_first_trial = as.vector(e070528citronellal$`neuron 3`$`stim. 1`)
neuron4_first_trial = as.vector(e070528citronellal$`neuron 4`$`stim. 1`)

neuron1_CAL1S = as.vector(CAL1S$`neuron 1`)
```

We start by declaring our functions: - `compute_spike_count` : used to compute the number of spikes within a time window - `compute_isi` : used to compute the interspike intervals within a time window

```

compute_spike_count <- function(spike_train, time_window){
  # Computes the number of spikes of a neuron for a given period, given a time
  # window length in seconds, e.g., 0.1 for a tenth of a second
  n = length(spike_train)
  max_seconds = spike_train[n]
  steps = seq(time_window, max_seconds+time_window, time_window)
  spike_count = rep(0, length(steps))
  for (s in 1:length(steps)) {
    spike_count[s] = sum(
      (spike_train<=steps[s]) & (spike_train>(steps[s]- time_window))
    )
  }
  return(list("time_steps"=steps, "spike_count"=spike_count))
}

plot_spike_count <- function(spike_count, time_window, title){
  # Plots the spike count given a time window length in seconds
  plot(spike_count$time_steps,
    spike_count$spike_count,
    type="S",
    main=title,
    xlab=paste("Seconds (Steps of ", time_window, "s.)"),
    ylab="Spike count")
}

compute_isi <- function(spike_train, time_window=NULL){
  # Computes the interspike interval given a spike train and a time window
  # if time_window is null, compute over the whole spike_train.
  if (is.null(time_window)) {
    ISI = diff(spike_train)
    return(list("eta"=c(compute_MLE_eta(ISI)),
      "theta"=c(compute_MLE_theta(ISI))))
  } else {
    max_seconds = spike_train[length(spike_train)]
    steps = seq(time_window, max_seconds+time_window, time_window)
    n = length(steps)
    MLE_eta = rep(NA, n)
    MLE_theta = rep(NA, n)
    for (s in 1:n) {
      kept_spikes = Filter(
        function(st) {(st<=steps[s]) & (st>(steps[s]- time_window))},
        spike_train
      )
      if (length(kept_spikes)>2){
        ISIs = diff(kept_spikes)
        MLE_eta[s] = compute_MLE_eta(ISIs)
        MLE_theta[s] = compute_MLE_theta(ISIs)
      } else {
        MLE_eta[s] = NA
        MLE_theta[s] = NA
      }
    }
    return(list("steps"=steps, "eta"=MLE_eta, "theta"=MLE_theta))
  }
}

plot_isi_parameters <- function(isi_parameters, time_window, title) {
  # Plots the interspike interval MLEs given a time window length in seconds
  n = length(isi_parameters$steps)
  computed_estimators = c(isi_parameters$eta, isi_parameters$theta)
  ymin = max(0, min(computed_estimators)-1)
  ymax = max(computed_estimators)+1
  plot(isi_parameters$steps,
    isi_parameters$eta,
    type="S", lty=1,
    ylim=c(ymin, ymax),
    main = title,
    xlab = paste("Seconds (Steps of ", time_window, "s.)"),
    ylab = "Maximum Likelihood Estimators' values",
    col = 2)
  lines(isi_parameters$steps, isi_parameters$theta, lty=2, type="S", col=4)
  legend(x = "topright", legend = c("Eta", "Theta"), col=c(2,4),

```



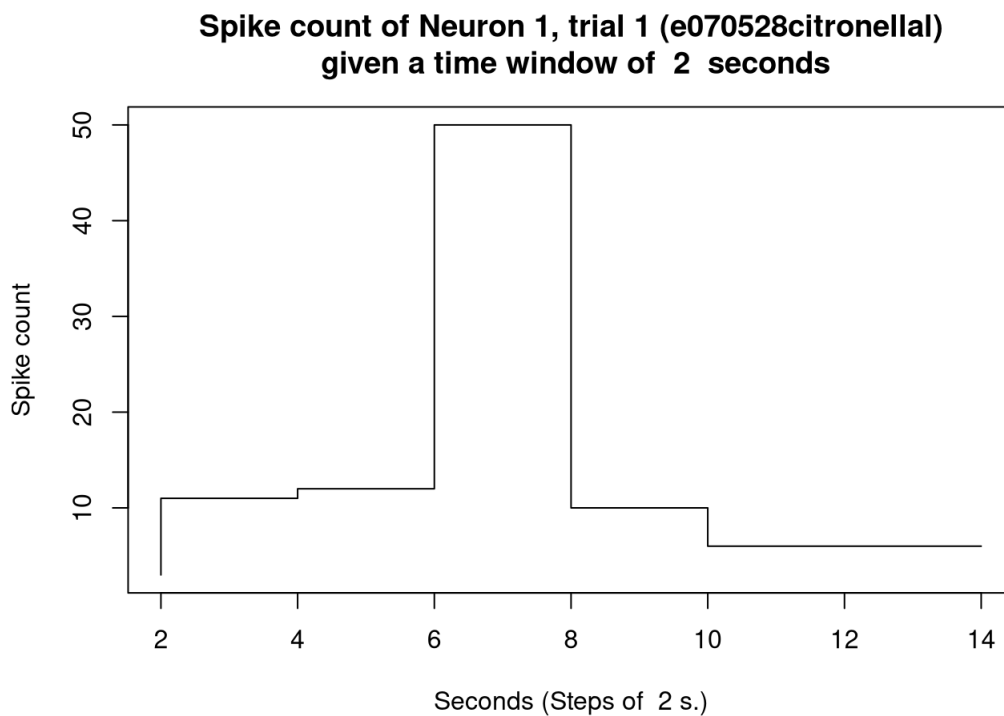
```
lty = c(1, 2), lwd = 2)
}
```

For data visualization purposes, we set a time interval where to compute spike counts and interspike intervals at 2 seconds.

```
time_window = 2
```

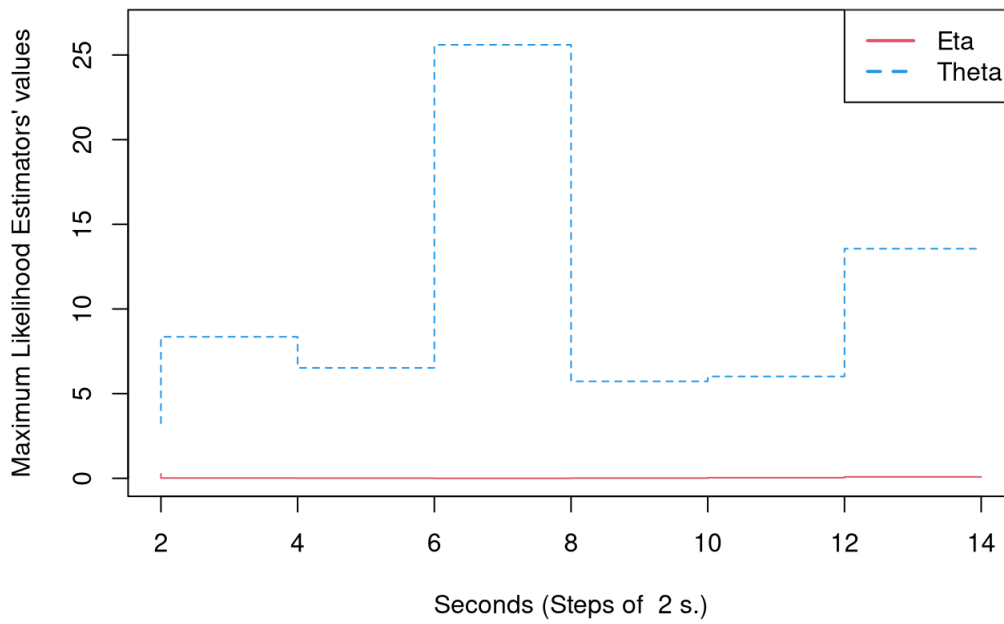
Neuron 1, first trial (e070528citronellal)

```
title = paste("Spike count of Neuron 1, trial 1 (e070528citronellal)\n",
              "given a time window of ", time_window, " seconds")
spike_count = compute_spike_count(neuron1_first_trial, time_window)
plot_spike_count(spike_count, time_window,
                 title=title)
```



```
title = paste("ISI MLE of Neuron 1, trial 1 (e070528citronellal)\n",
              "given a time window of ", time_window, " seconds")
ISIs = compute_isi(neuron1_first_trial, time_window)
plot_isi_parameters(ISIs, time_window, title)
```

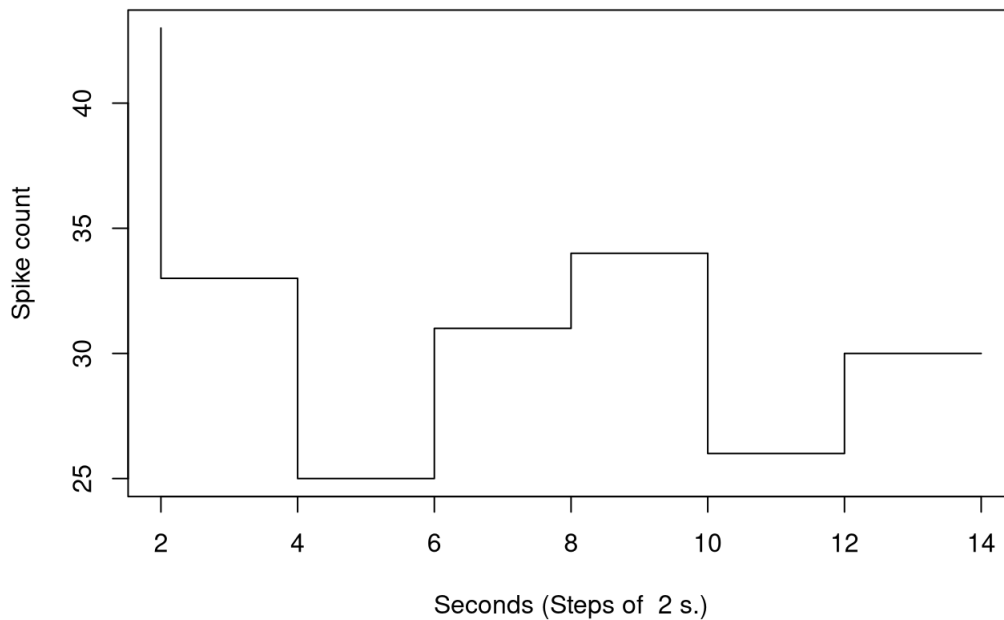
ISI MLE of Neuron 1, trial 1 (e070528citronellal) given a time window of 2 seconds



Neuron 2, first trial (e070528citronellal)

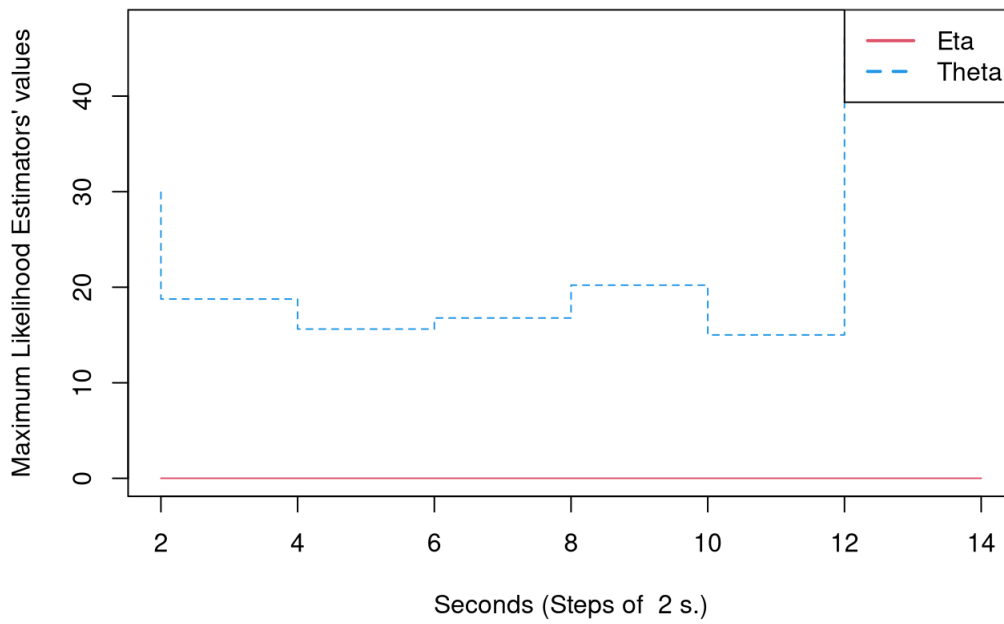
```
title = paste("Spike count of Neuron 2, trial 1 (e070528citronellal)\n",
              "given a time window of ", time_window, " seconds")
spike_count = compute_spike_count(neuron2_first_trial, time_window)
plot_spike_count(spike_count, time_window,
                 title=title)
```

Spike count of Neuron 2, trial 1 (e070528citronellal) given a time window of 2 seconds



```
title = paste("ISI MLE of Neuron 2, trial 1 (e070528citronellal)\n",
              "given a time window of ", time_window, " seconds")
ISIs = compute_isi(neuron2_first_trial, time_window)
plot_isi_parameters(ISIs, time_window, title)
```

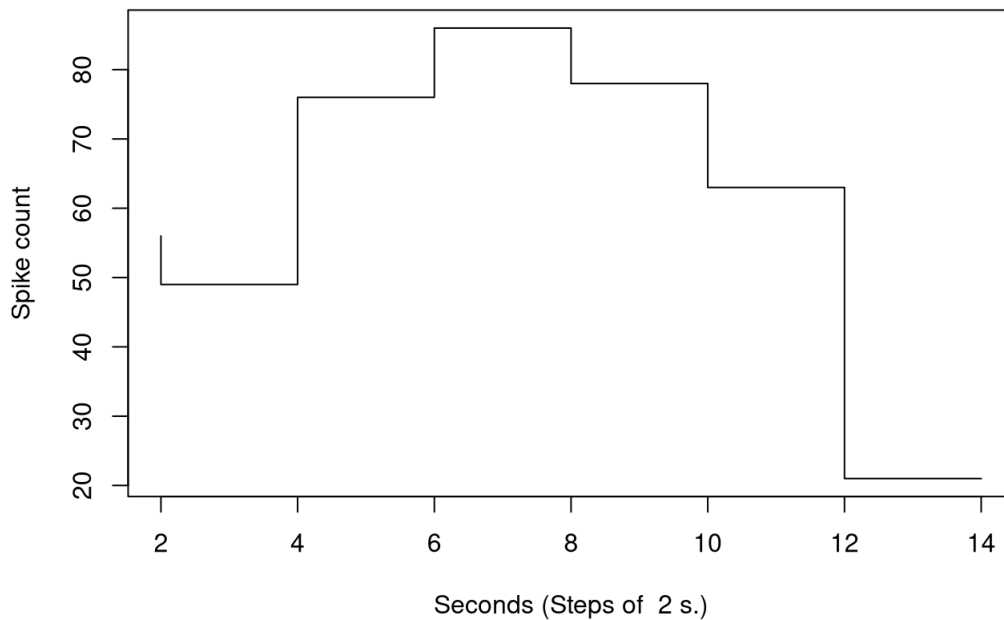
ISI MLE of Neuron 2, trial 1 (e070528citronellal) given a time window of 2 seconds



Neuron 3, first trial (e070528citronellal)

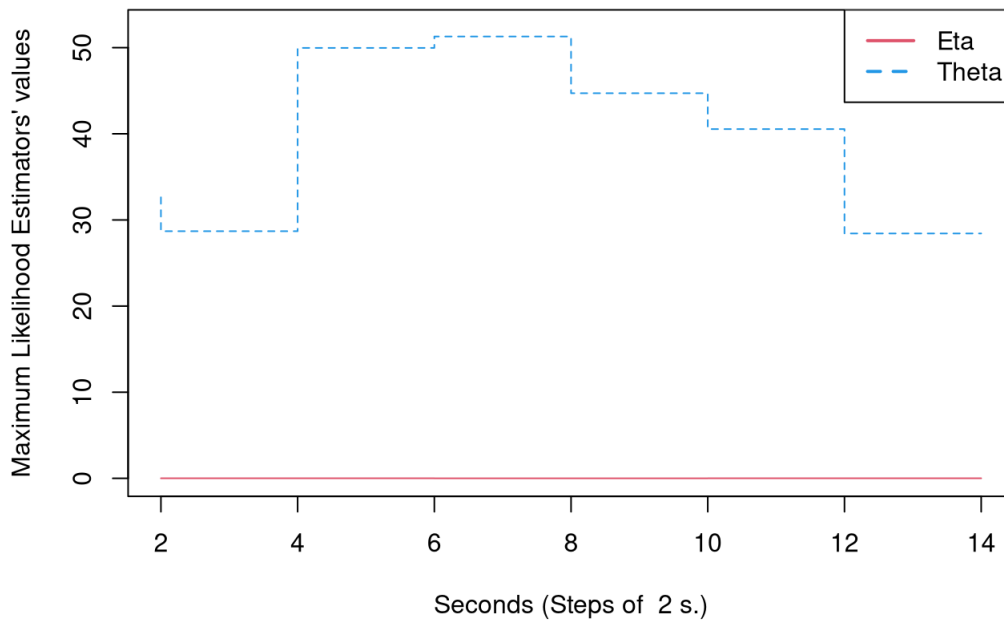
```
title = paste("Spike count of Neuron 3, trial 1 (e070528citronellal)\n",
              "given a time window of ", time_window, " seconds")
spike_count = compute_spike_count(neuron3_first_trial, time_window)
plot_spike_count(spike_count, time_window,
                 title=title)
```

Spike count of Neuron 3, trial 1 (e070528citronellal) given a time window of 2 seconds



```
title = paste("ISI MLE of Neuron 3, trial 1 (e070528citronellal)\n",
              "given a time window of ", time_window, " seconds")
ISIs = compute_isi(neuron3_first_trial, time_window)
plot_isi_parameters(ISIs, time_window, title)
```

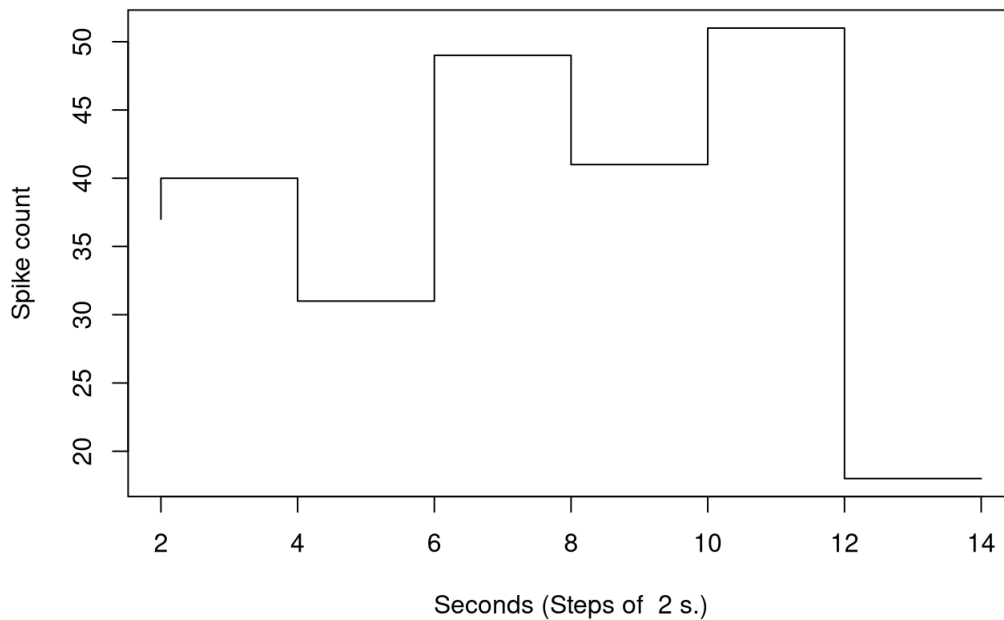
ISI MLE of Neuron 3, trial 1 (e070528citronellal) given a time window of 2 seconds



Neuron 4, first trial (e070528citronellal)

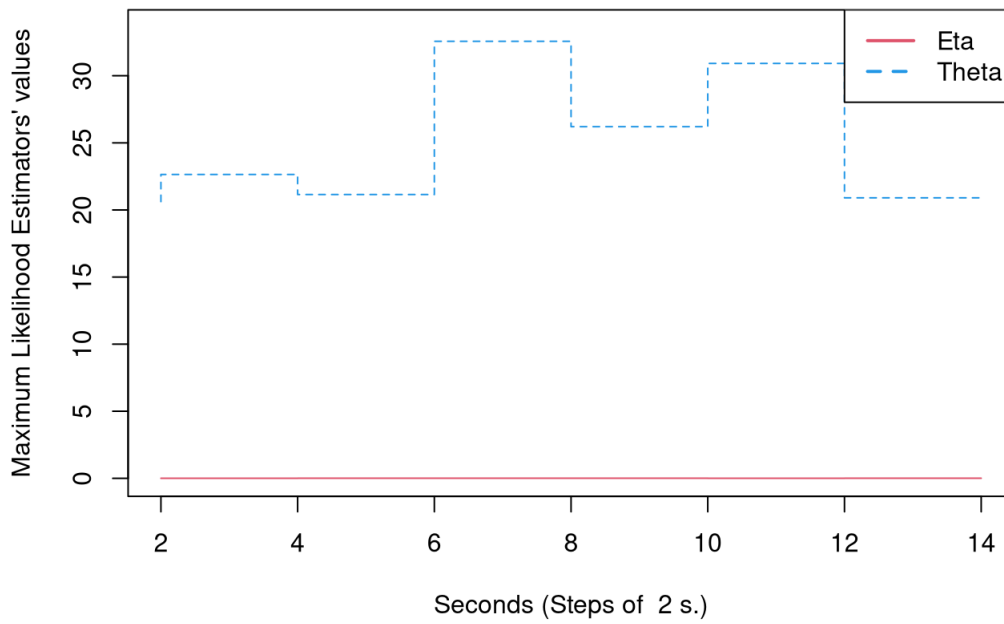
```
title = paste("Spike count of Neuron 4, trial 1 (e070528citronellal)\n",
             "given a time window of ", time_window, " seconds")
spike_count = compute_spike_count(neuron4_first_trial, time_window)
plot_spike_count(spike_count, time_window,
                 title=title)
```

Spike count of Neuron 4, trial 1 (e070528citronellal) given a time window of 2 seconds



```
title = paste("ISI MLE of Neuron 4, trial 1 (e070528citronellal)\n",
             "given a time window of ", time_window, " seconds")
ISIs = compute_isi(neuron4_first_trial, time_window)
plot_isi_parameters(ISIs, time_window, title)
```

ISI MLE of Neuron 4, trial 1 (e070528citronellal) given a time window of 2 seconds



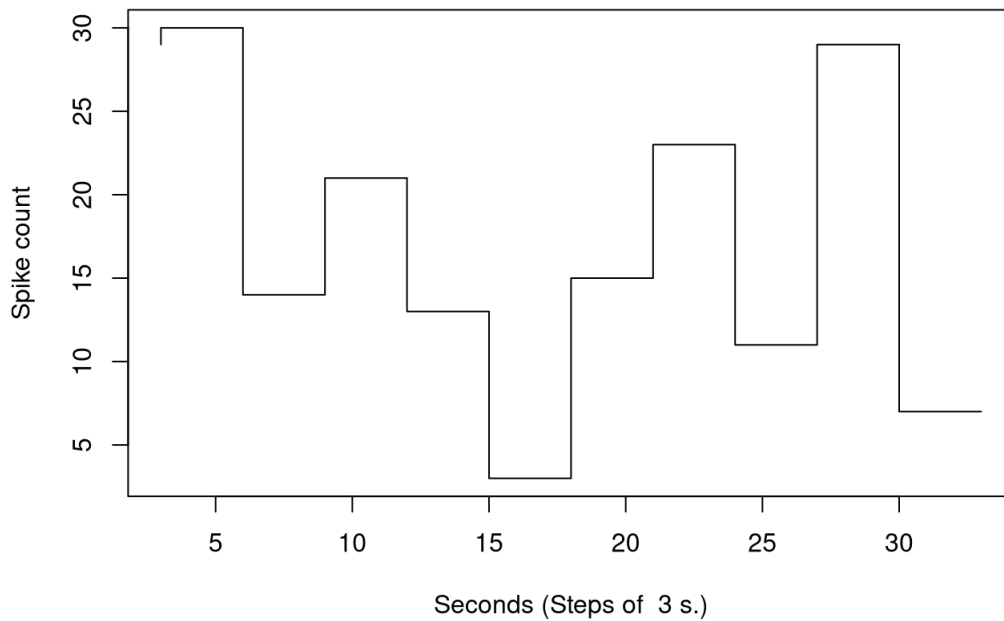
Neuron 1, first trials (CAL1S)

We switch to a time window of 3 seconds.

```
time_window = 3

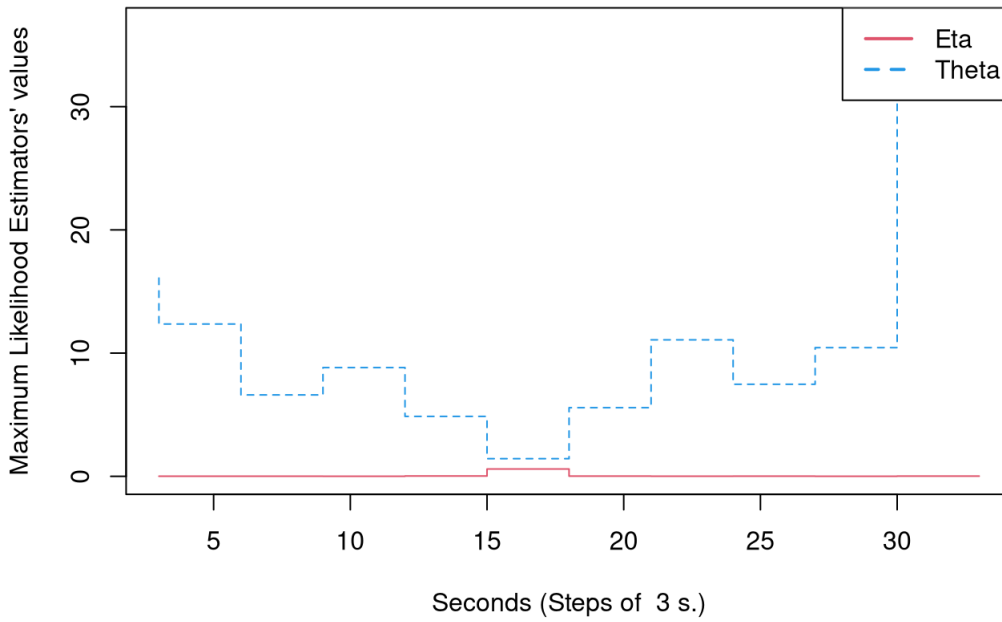
title = paste("Spike count of Neuron 1 (CAL1S)\n",
              "given a time window of ", time_window, " seconds")
spike_count = compute_spike_count(neuron1_CAL1S, time_window)
plot_spike_count(spike_count, time_window,
                 title=title)
```

Spike count of Neuron 1 (CAL1S) given a time window of 3 seconds



```
title = paste("ISI MLE of Neuron 1 (CAL1S)\n",
              "given a time window of ", time_window, " seconds")
ISIs = compute_isi(neuron1_CAL1S, time_window)
plot_isi_parameters(ISIs, time_window, title)
```

ISI MLE of Neuron 1 (CAL1S) given a time window of 3 seconds



6. Superpose the non parametric density estimator and the curve $\theta e^{-\theta(x-\eta)} 1_{x \geq \eta}$ for the different choices of neurons/ time windows experiments... Eventually comment

TBD

Question 2

For the simplified Georgopoulos setting, we consider that the spiking rate of a given neuron of the motor cortex is given by:

$$Y = a + b \cos(\theta) + \sigma \epsilon$$

with $\epsilon \sim \mathcal{N}(0, 1)$, and θ the angle between the actual direction of movement M and the preferred direction of movement of the cell, say C . We assume M and C to be unit vectors of the plan \mathbb{R}^2 .

1. Convince yourself by doing pictures that $\cos(\theta) = \langle M, C \rangle$ the scalar product between M and C .

We recall that, given $M, C \in \mathbb{R}^2$ such that $M = (a, b)^T$ and $C = (c, d)^T$,

$$\langle M, C \rangle = \sum_{i \in \{1, 2\}} M_i * C_i = a * c + b * d$$

Furthermore, if M and C are inscribed on the unit circle, we have:

$$\begin{aligned} M &= (a, b) \\ C &= (c, d) \\ a, b, c, d &\in [0, 1] \\ a &= \cos(M_\theta) \quad (\text{with } M_\theta \text{ the angle of } M \text{ to the x-axis}) \\ b &= \sin(M_\theta) \\ c &= \cos(C_\theta) \quad (\text{with } C_\theta \text{ the angle of } C \text{ to the x-axis}) \\ d &= \sin(C_\theta) \end{aligned}$$

We also note:

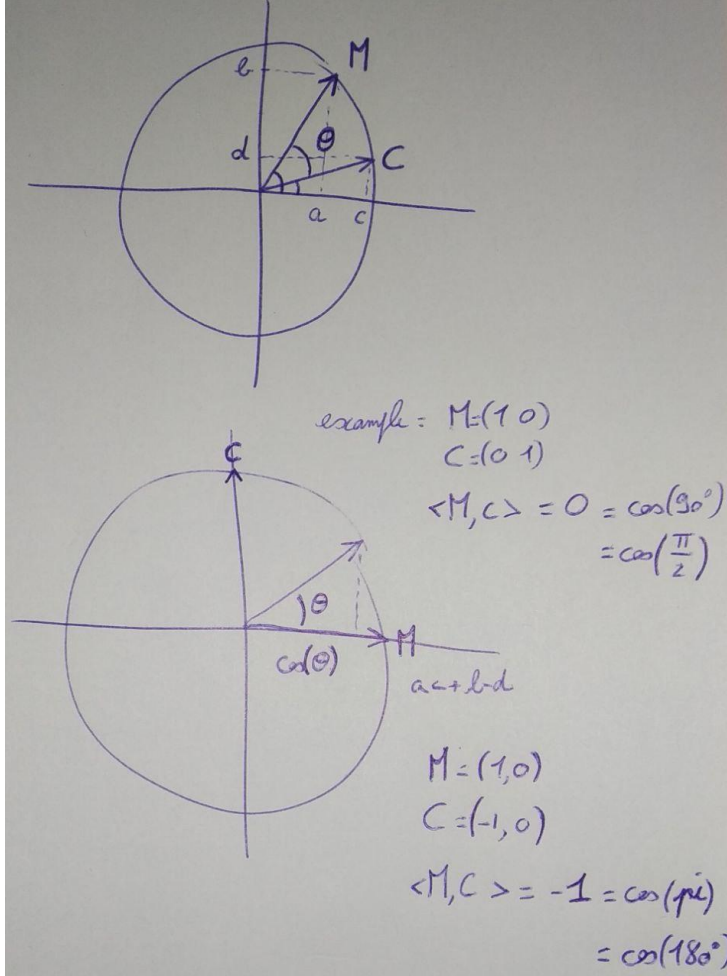
$$\begin{aligned} \sin(x) * \sin(y) &= \frac{1}{2} (\cos(x - y) - \cos(x + y)) \\ \cos(x) * \cos(y) &= \frac{1}{2} (\cos(x - y) + \cos(x + y)) \end{aligned}$$

Given those expressions we find that:

$$\begin{aligned}
\langle M, C \rangle &= ac + bd = \cos(M_\theta) * \cos(C_\theta) + \sin(M_\theta) * \sin(C_\theta) \\
&= \frac{1}{2} (\cos(M_\theta - C_\theta) + \cos(M_\theta + C_\theta) + \cos(M_\theta - C_\theta) - \cos(M_\theta + C_\theta)) \\
&= \frac{1}{2} (\cos(M_\theta - C_\theta) + \cos(M_\theta - C_\theta)) \\
&= \cos(M_\theta - C_\theta)
\end{aligned}$$

As such, given θ the angle between the direction of movement M and C (i.e., inscribed on the unit circle, the difference between the angle of M and the angle of C to the x-axis), we find that:

$$\begin{aligned}
\theta &= M_\theta - C_\theta \\
\langle M, C \rangle &= \cos(\theta)
\end{aligned}$$



t1_1_image

2. Assume you have a very nicely behaved population of neurons that is recorded: n_1 of them have for C , the vector $e_1 = (1, 0)$ and n_2 of them the vector $e_2 = (0, 1)$ and no other preferred direction NB : *this is to make the math easier, it is not true in practice*. Assume that you record all the cells at the same time and that they behave independently of each other given the direction of movement M . Rewrite the model with only one Gaussian vector for which you will give the mean vector and the covariance matrix.

We have established that a neuron can be described as $Y = a + b \cos(\theta) + \sigma \epsilon$, and that $\theta = M_\theta - C_\theta$ and $\langle M, C \rangle = \cos(\theta)$. Furthermore, given the stated vectors e_1 and e_2 , we can state:

$$\begin{aligned}
M &= (x, y) \quad \text{with } x, y \in \mathbb{R} \\
e_1 &= (1, 0) \\
e_2 &= (0, 1) \\
\langle M, e_1 \rangle &= x = \cos(M) \\
\langle M, e_2 \rangle &= y = \sin(M)
\end{aligned}$$

As such, we can respectively describe the neurons with $C = e_1$ and $C = e_2$ as:

$$\begin{aligned}
\text{neuron}_{n_1} &= a_{n_1} + b_{n_1} * \cos(M) + \sigma_{n_1} \epsilon_{n_1} \\
\text{neuron}_{n_2} &= a_{n_2} + b_{n_2} * \sin(M) + \sigma_{n_2} \epsilon_{n_2}
\end{aligned}$$

Given each neuron is independent and identically distributed, we can rewrite the behavior of the ensemble of neurons as a single Gaussian vector such that:

$$\begin{aligned}
& \forall i \in \{1, \dots, n_1 + n_2\} \\
& a_i, b_i, \sigma_i \in \mathbb{R} \\
& \epsilon_i \sim \mathcal{N}(0, 1) \\
& Y = \begin{pmatrix} a_1 \\ \vdots \\ a_{n_1} \\ a_{n_1+1} \\ \vdots \\ a_{n_1+n_2} \end{pmatrix} + \begin{pmatrix} b_1 * \cos(M) \\ \vdots \\ b_{n_1} * \cos(M) \\ b_{n_1+1} * \sin(M) \\ \vdots \\ b_{n_1+n_2} * \sin(M) \end{pmatrix} + \sigma \epsilon \\
& \sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & & \ddots & \\ 0 & & & \sigma_{n_1+n_2} \end{bmatrix}; \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_{n_1+n_2} \end{pmatrix}
\end{aligned}$$

Given that $\forall i \in \{1, \dots, n_1 + n_2\}$, $\mathbb{E}[\epsilon_i] = 0$ we have:

$$\begin{aligned}
\mu = \mathbb{E}[Y] &= \begin{pmatrix} a_1 \\ \vdots \\ a_{n_1} \\ a_{n_1+1} \\ \vdots \\ a_{n_1+n_2} \end{pmatrix} + \begin{pmatrix} b_1 * \cos(M) \\ \vdots \\ b_{n_1} * \cos(M) \\ b_{n_1+1} * \sin(M) \\ \vdots \\ b_{n_1+n_2} * \sin(M) \end{pmatrix} \\
\Sigma = \sigma \sigma' &= \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & & \ddots & \\ 0 & & & \sigma_{n_1+n_2}^2 \end{bmatrix}
\end{aligned}$$

3. The question is now to decode the signal : you just observe the neurons, you know the parameters a_i , b_i and $\sigma_i = \sigma$ (i.e. different intercept and slope but the same variance), for each cell i , you want to guess the actual direction of movement M . Write down the likelihood (as a function of M) of this problem

Given $\forall i \in \{1, \dots, n_1 + n_2\}$, $\sigma_i = \sigma$, we can restate Y such that:

$$\begin{aligned}
Y = Y(M) &= \begin{pmatrix} a_1 \\ \vdots \\ a_{n_1} \\ a_{n_1+1} \\ \vdots \\ a_{n_1+n_2} \end{pmatrix} + \begin{pmatrix} b_1 * \cos(M) \\ \vdots \\ b_{n_1} * \cos(M) \\ b_{n_1+1} * \sin(M) \\ \vdots \\ b_{n_1+n_2} * \sin(M) \end{pmatrix} + \sigma * I_{n_1+n_2} \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_{n_1+n_2} \end{pmatrix} \\
\mu(M) &= \begin{pmatrix} a_1 \\ \vdots \\ a_{n_1} \\ a_{n_1+1} \\ \vdots \\ a_{n_1+n_2} \end{pmatrix} + \begin{pmatrix} b_1 * \cos(M) \\ \vdots \\ b_{n_1} * \cos(M) \\ b_{n_1+1} * \sin(M) \\ \vdots \\ b_{n_1+n_2} * \sin(M) \end{pmatrix} \\
Y(M) &= \mu(M) + \sigma * I_{n_1+n_2} \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_{n_1+n_2} \end{pmatrix}
\end{aligned}$$

Now, given the IID characteristic of each neuron, we can write the likelihood of the system as a function of M such that:

$$\begin{aligned}
& \forall i \in \{1, \dots, n_1 + n_2\} \\
& a_i, b_i, \mu_i, \sigma \in \mathbb{R} \\
& \epsilon_i \sim \mathcal{N}(0, 1) \\
& f_i(M) = a_i + b_i * \cos(M) + \sigma \epsilon_i \\
& g_i(M) = a_i + b_i * \sin(M) + \sigma \epsilon_i \\
& L(M) = \prod_{i=1}^{n_1} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(f_i(M) - \mu_i)^2}{2\sigma^2}\right) * \prod_{i=1}^{n_2} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(g_i(M) - \mu_i)^2}{2\sigma^2}\right) \\
& = \frac{1}{(\sqrt{2\pi\sigma^2})^{n_1+n_2}} * \left(\prod_{i=1}^{n_1} \exp\left(-\frac{(f_i(M) - \mu_i)^2}{2\sigma^2}\right) * \prod_{i=1}^{n_2} \exp\left(-\frac{(g_i(M) - \mu_i)^2}{2\sigma^2}\right) \right) \\
& = \frac{1}{(\sqrt{2\pi\sigma^2})^{n_1+n_2}} * \left(\exp\left(-\frac{\sum_{i=1}^{n_1} (f_i(M) - \mu_i)^2}{2\sigma^2}\right) * \exp\left(-\frac{\sum_{i=1}^{n_2} (g_i(M) - \mu_i)^2}{2\sigma^2}\right) \right) \\
& = \frac{1}{(\sqrt{2\pi\sigma^2})^{n_1+n_2}} * \left(\exp\left(-\frac{\|f(M) - \mu\|^2}{2\sigma^2}\right) * \exp\left(-\frac{\|g(M) - \mu\|^2}{2\sigma^2}\right) \right) \\
& = \frac{1}{(\sqrt{2\pi\sigma^2})^{n_1+n_2}} * \left(\exp\left(-\frac{\|f(M) - \mu_f\|^2}{2\sigma^2}\right) * \exp\left(-\frac{\|g(M) - \mu_g\|^2}{2\sigma^2}\right) \right)
\end{aligned}$$

We find that:

$$\begin{aligned}
Y - \mu &= \|f(M) - \mu_f\|^2 + \|g(M) - \mu_g\|^2 \\
L(M) &= \frac{1}{(\sqrt{2\pi\sigma^2})^{n_1+n_2}} * \exp\left(-\frac{\|Y(M) - \mu(M)\|^2}{2\sigma^2}\right) \\
l(M) &= -\frac{n_1 + n_2}{2} \log(\pi\sigma^2) - \frac{\|Y(M) - \mu(M)\|^2}{2\sigma^2}
\end{aligned}$$

4. Show that the maximization of the likelihood in M is equivalent to a least-square minimization problem and that σ does not play any role in it.

Maximizing $L(M)$ corresponds to maximizing $l(M)$, i.e. minimizing $\|Y(M) - \mu(M)\|^2$.

As part of the least-square minimization problem, we consider the system $X\beta = Y$ where, in our case, we set the following:

$$\begin{aligned}
X &= \begin{pmatrix} 1 & X_1 \\ \vdots & \\ 1 & X_{n_1+n_2+2} \end{pmatrix} \\
\hat{\beta} &= \begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix}
\end{aligned}$$

It implies that we are looking to find $Y - X\hat{\beta} = 0$, i.e.:

$$\begin{pmatrix} Y_1 - \hat{a} + \hat{b}X_1 \\ \vdots \\ Y_{n_1+n_2} - \hat{a} + \hat{b}X_{n_1+n_2} \end{pmatrix} = 0_{n_1+n_2}$$

However, we find that this also corresponds to $\beta = \arg\min \sum_{i=1}^{n_1+n_2} (Y_i - \sum_{j=1}^2 X_{i,j} \beta_j)^2 = \|Y - X\hat{\beta}\|^2$.

Consequently, we find a correspondence with our preliminary minimization problem $\|Y(M) - \mu(M)\|^2$ and the OLS.

5. Solve for each of the two coefficients of M (on e_1 and e_2), the minimization problem. Or solve it directly on θ

TBD