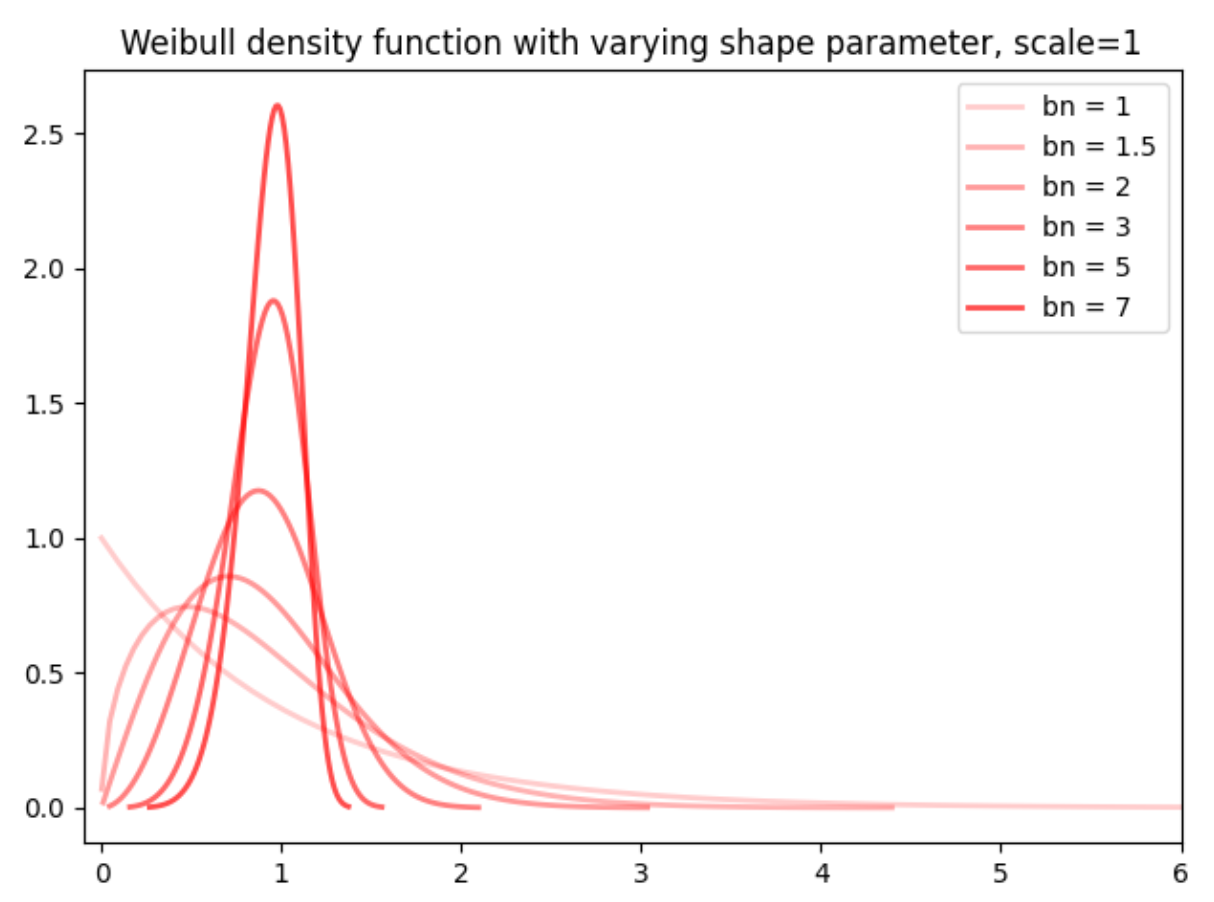
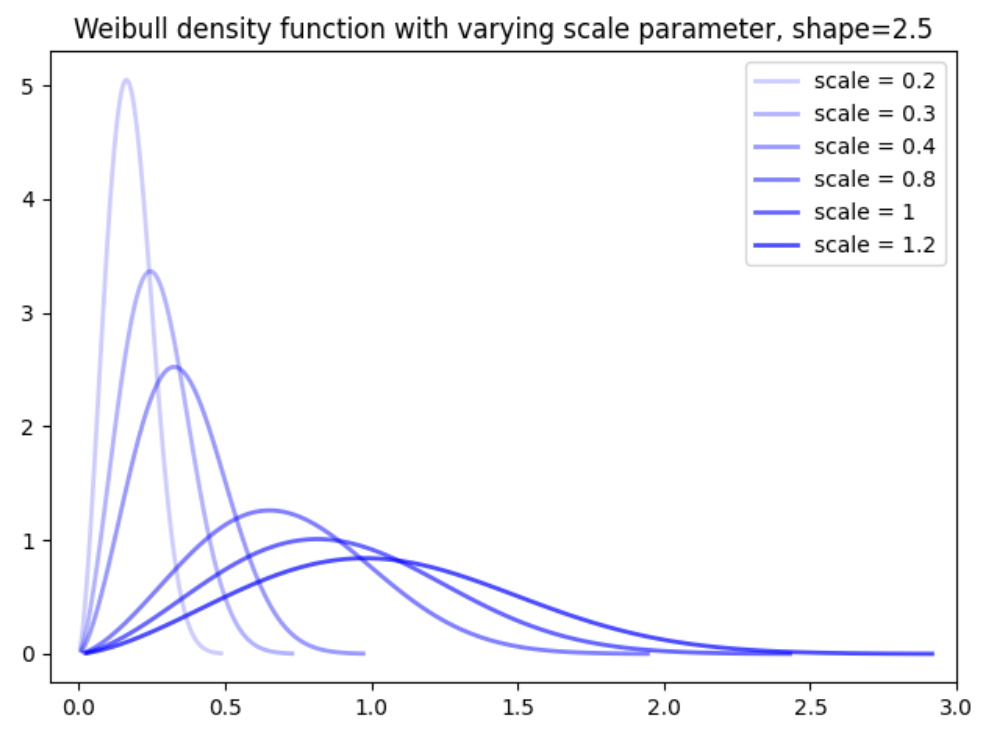
## Bn Parameter Calibration using Bayesian Inference Process:

The bn parameter importance in our modelling:

In the Fatigue estimation, bn in the shape parameter of the variable Nf (number of cycle) that follows a Weibull standard law (standard meaning it comprises a shape parameter, bn, a scale parameter σ0, and no location parameter, or location parameter equal to 0).

Bn has a great importance in the calculation of the final probability: Pf, corresponding to the Cumulative Density Function of the variable Nf.

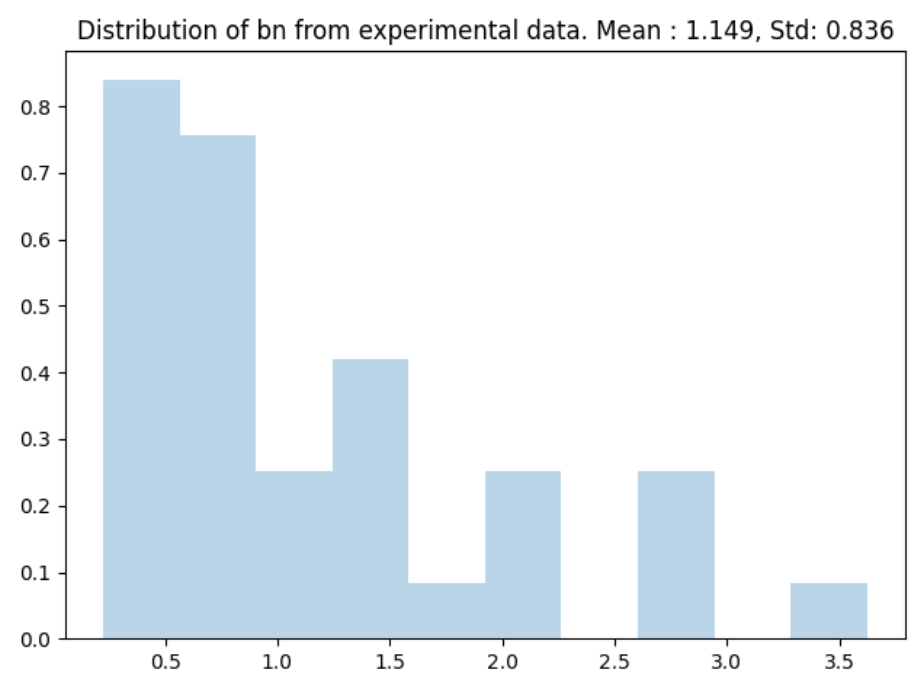




Initial calculation of bn:

We have, from experimental data, a first small sample of bn values, by applying the equation 26, with the constraint of bn being positive:

This range of values give us the following distribution:



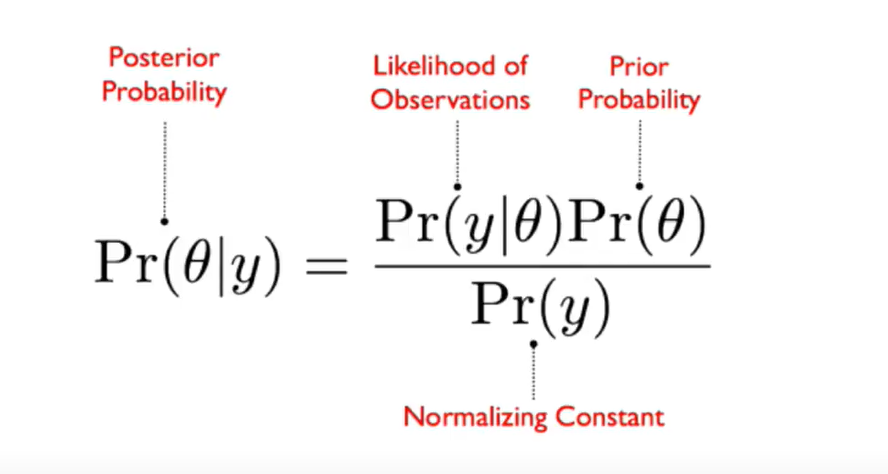
It has to be said though that the number of data points is very small and that the calculation of bn here, from the positive constraint, reflect only a part of the initial combinations possible. In addition, cases where number of cycle for two different volumes was either too close (less than 3% away) or too large (more than 20 times away) have to be excluded cause the calculation produced extreme values.

Bn then cannot be solely evaluated based on this technique. The true distribution of Bn remains unknown, hence the need for a Bayesian inference calibration to fit this unknown distribution (or “target distribution).

Bayesian inference principle:

Bayesian inference relies on conditional probability and inverse probability, this is to say, on the principle of updating prior information with posterior distribution by passing by a “conditional intermediate” called the likelihood distribution. This likelihood distribution is considered to be an easier distribution to sample from.

The update process is done through a sampling mechanism that follows



The Prior distribution quantifies the uncertainty in latent variable(s) (here it would be bn) and sums up the initial beliefs that we have about the true distribution.

The likelihood distribution describes the distribution of a variable linked to the target distribution by conditionality (in our case the variable Nf, describing the number of cycles) and that has observations available which enables the data generating mechanism.

The posterior distribution is not exactly equal to the prior multiplied by the likelihood but equal up to a constant: called Pr(y) here, that would be Pr(Nf). This constant is a marginal probability, meaning the marginal probability of the data Nf which can be obtained by integrating all over the domain of the target variable (below θ and in our case bn)

Une image contenant texte

Description générée automatiquement

The calculation of these integrals, which sometimes can involve multiple parameters in the vector θ,

can be computationally heavy and there is no guarantee to find an analytical solution.

The Bayesian inference procedure, instead, chooses to approach the target distribution by simulating new samples and accepting or rejecting them in a pool of samples based on a specific algorithm. The new value for the θ vector is then updated at each step, and, provided that the algorithm converges, one is certified to arrive to a good enough estimation of the true distribution after a certain number of samples. The first samples, far off from the desired distribution, are not kept (“burn in”).

MCMC methods:

The mathematical logic behind the convergence is given by the application of Monte Carlo Markov Chain methods.

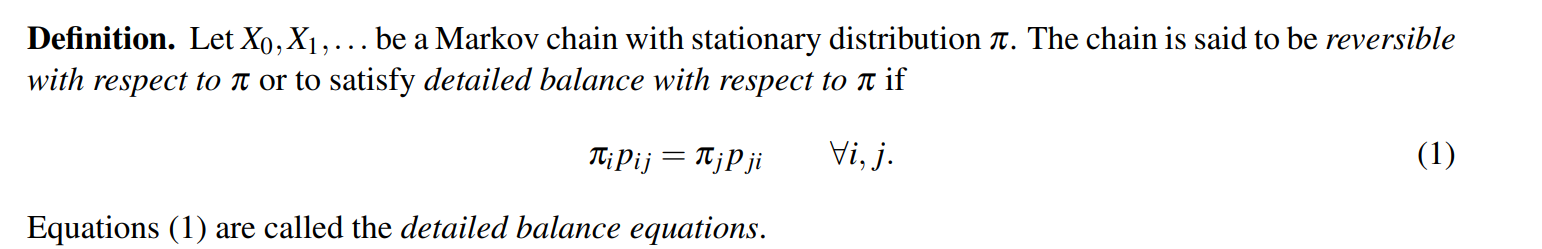
Markov Chain

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event: therefore, the process is considered “memoryless”.

Markov chain Monte Carlo methods create a chain of samples from a continuous random variable, with probability density proportional to a known function. These samples can be used to evaluate an integral over that variable, as its expected value or variance.

By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by recording states from the chain. The more steps that are included, the more closely the distribution of the sample matches the actual desired distribution. The equilibrium distribution condition is ensured by the Detailed Balanced condition.

If we consider X0, X1, … a Markov Chain with stationary distribution π. The chain is said to satisfy the detailed balance condition with respect to π if :



The detailed balance equation says the flow of probability is balanced locally: at each “edge” (each edge between the initial and target distribution), the amount of probability that flows across in one direction in one step, equals the amount that flows in the opposite direction.

If we can show the probability of going to a state Xt of the distribution to a next step Xt+1 of the distribution follows a detailed balanced condition, then we know that P|X is a stationary distribution of the Markov Chain and therefore that the Markov Chain will be sampling from P|X.

Inference algorithm:

Various algorithms exist for constructing chains, including the random walk Metropolis–Hastings algorithm, that has been chosen in the reference paper.

The general Metropolis algorithm proceeds as follows:

1. Start at some random initial point θ(0).
2. Draw a proposed value θ\* from some distribution p(θ\*|θ).
3. Compute the r ratio (see below):

r =

If r>1 then reject θ\* and stay put: θ(t+1) = θ(t)

Otherwise, accept the proposed value with probability

1. If accepted, move to the new spot. Otherwise, stay put. Either way, record your position.
2. Repeat (2), (3), and (4) for a set number of iterations.

This naïve algorithm allows to move increasingly closer to an area of higer probability

Let f be the (possibly unnormalized) target density, x(j) be a current value and be a proposal distribution, then :

* Sample x\* ~
* Calculate the acceptance probability :

With r =

* Set x(j+1) = x\* with probability , otherwise set x(j+1) = x(j) (meaning stand still, do not move from the current point)

With f: density distribution of θ and q: likelihood distribution.

Implementation of the calibration model:

The model has been implemented using the pymc3 python library that has been built specifically for Bayesian and variational inference. Pymc3 add a layer of calculation modules on top of scipy and has built in functionalities for distribution generation as well as built in sampling algorithms.

The paper gives the following indications for the selection of the prior and likelihood distribution:

On the likelihood distribution:

* The likelihood distribution is defined by the following formula:

Une image contenant texte

Description générée automatiquement

Where the expression: ln (No(w)) + (1/bn)\*ln2 -ln(ɣ) corresponds to the log of the parameter: Nf0: the mean value of the number of cycles.

* We should remark, first that Nf has been passed to the log random variable: as the neperian logarithm of Ni and of the mean of Ni is considered instead of a simple Ni.
* However, it is said at the same time that the likelihood distribution result from a maximum likelihood estimation on the Nf variable, meaning, that bn is defined by the following function: meaning the product all values of 1<=i<=t of the density function f of the Weibull random variables N1, N2, ….Nt representing the number of cycles.
* First, the equation 38 does not correspond to a product of Weibull random variable density functions, nor to the product of a log of a series of Weibull random variate. It does approximate the product of log normal density function if we multiply it by : 1/σt and by: σε t

-Density function of a weibull

-Density function of a log Weibull (also called extreme value): where Y = ln(X) and x following a Weibull. α: scale parameter and β shape parameter.

-Density function of a LogNormal distribution:

* One should note that, though, that both the log normal and Weibull are commonly used to model fatigue stress and that they are very close to each other to model skewed events under certain conditions. As what we consider here in the paper formula is not the Nf variable but the logarithm of Nf, one could argue that the lognormal distribution should be preferred over the Weibull.

Tests performed on the prior and the likelihood for the Bayesian Inference of the calibration of the bn parameter:

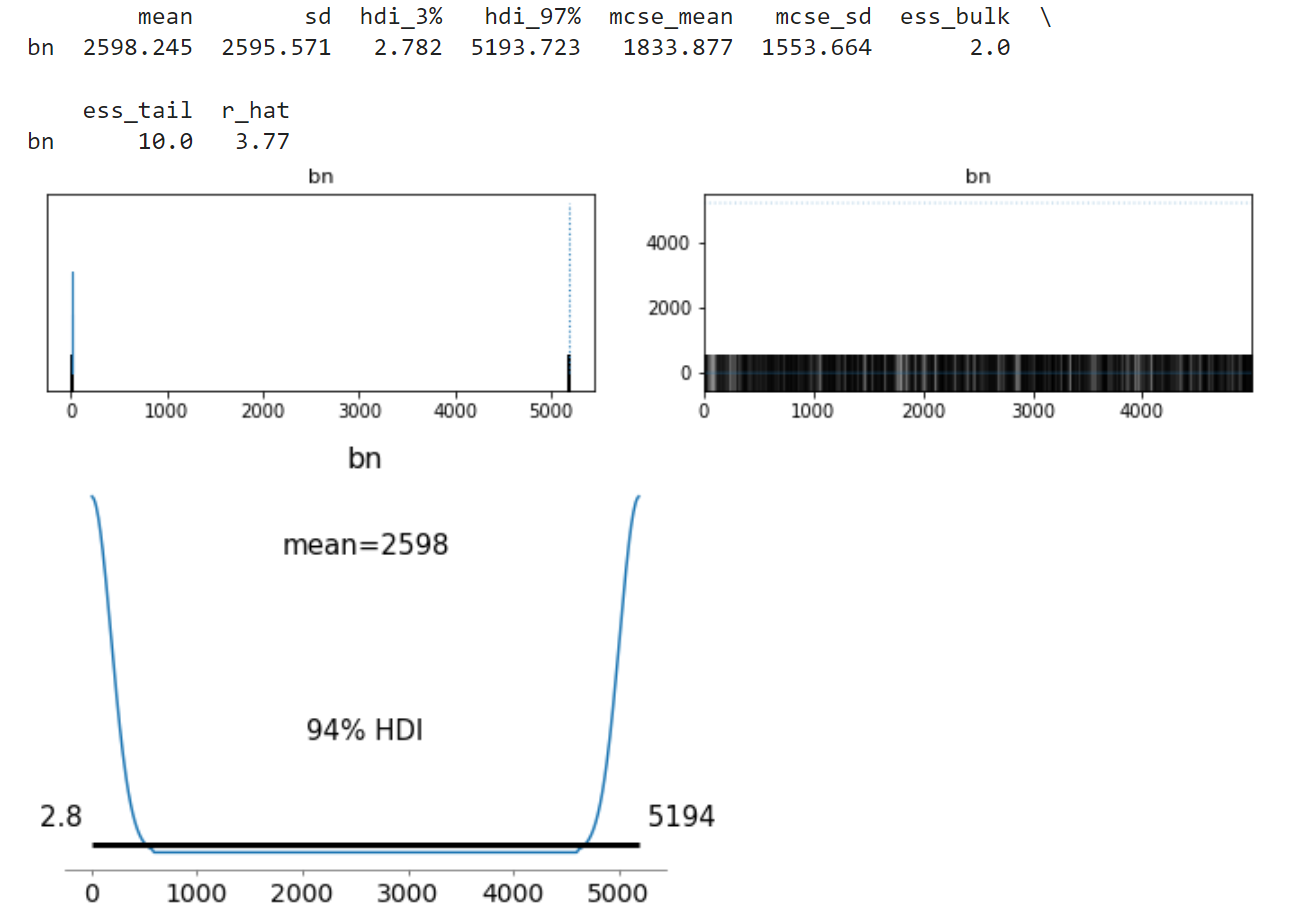
In a Bayesian inference process, the prior is defined as the function that reflects the prior beliefs about the target distribution: meaning in our case, it should be instantiated as a Normal distribution, or, more precisely, a Truncated Normal as the values of bn can only be positive.

However, in the scientific paper, there is no clarity about the definition of the prior distribution:

Test on the prior: prior is defined with the same function as the final custom likelihood without observations:

Output:

* We do not converge: there were more than 500 divergences after tuning. The result are similar with Metropolis Hasting as well as with pymc3 automatic sampler function (meaning automatic search of the best sampler possible).
* Consequently, the number of retained samples after 5000 iterations is very low: less than 200 for some parameters.
* The obtained distribution is not significant.
* Increasing the number of iterations to 10000 does not bring anything in terms of convergence and the mean explodes to levels such as 2500.



Conclusion: The prior distribution most likely does not have a formula similar to the likelihood and the lack of precision of the paper will let us use our own assumption.

As the likelihood distribution depends on bn and arises from bn, make the prior be defined also by the same function had little chance to be successful.

Let’s establish the prior at as a Normal distribution with mean 1.45 and standard deviation 0.32 with only positive values.

We will then use the built in TruncatedNormal from pymc3.

Tests on the likelihood:

Outside of the custom formula from the paper implemented in the main calibration process, the likelihood function is defined as the result of a Maximum Likelihood Estimation on the random variable Nf (number of cycles), that follows a Weibull distribution.

Maximum likelihood with fw, density function of the variable Ni:

Here again, there remains some doubts about what density function should be chosen and what parameters should be chosen, apart from bn.

We tested several options in the following way:

Test 1: Nf following a two parameter Weibull distribution with: location =0, shape =bn and scale λ = ɣ \* N0(w) (effectively transforming the variable from Nf/ ɣ \* N0(w) so that it aligns with the formula of the Cumulative probability given in equation 30 (see below).

Test 2: Nf following a two parameter Weibull distribution with: location=0, shape =bn and but scale, λ = 1.

Test 3: Nf following a of the log of a Weibull distribution, also called an Extreme Value distribution of type 3, with location parameter 0, shape parameter bn and scale parameter, λ = ɣ \* N0(w).

Test4: Nf following a of the log of a Weibull distribution, also called an Extreme Value distribution of type 3, with location parameter 0, with shape parameter bn and scale parameter , λ = 1.

Test 5: Nf following a lognormal distribution with location and standard deviation: sigma. Due to the high similarities between the Weibull and the LogNormal for big samples under certain conditions on the parameters and the resemblance of the paper formula (38) to a lognormal density function.

To sum up:

|  |  |
| --- | --- |
| Test | Distribution |
| Test1 |  |
| Test2 |  |
| Test3 | GEV(0, ɣ \* N0(w), bn) |
| Test4 | GEV (0,1, bn) |
| Test5 | Lognormal (N0(w), σ) |

Test1 and Test2: Weibull distribution

fw will be of the following form:

With bn: shape parameter and λ scale parameter.

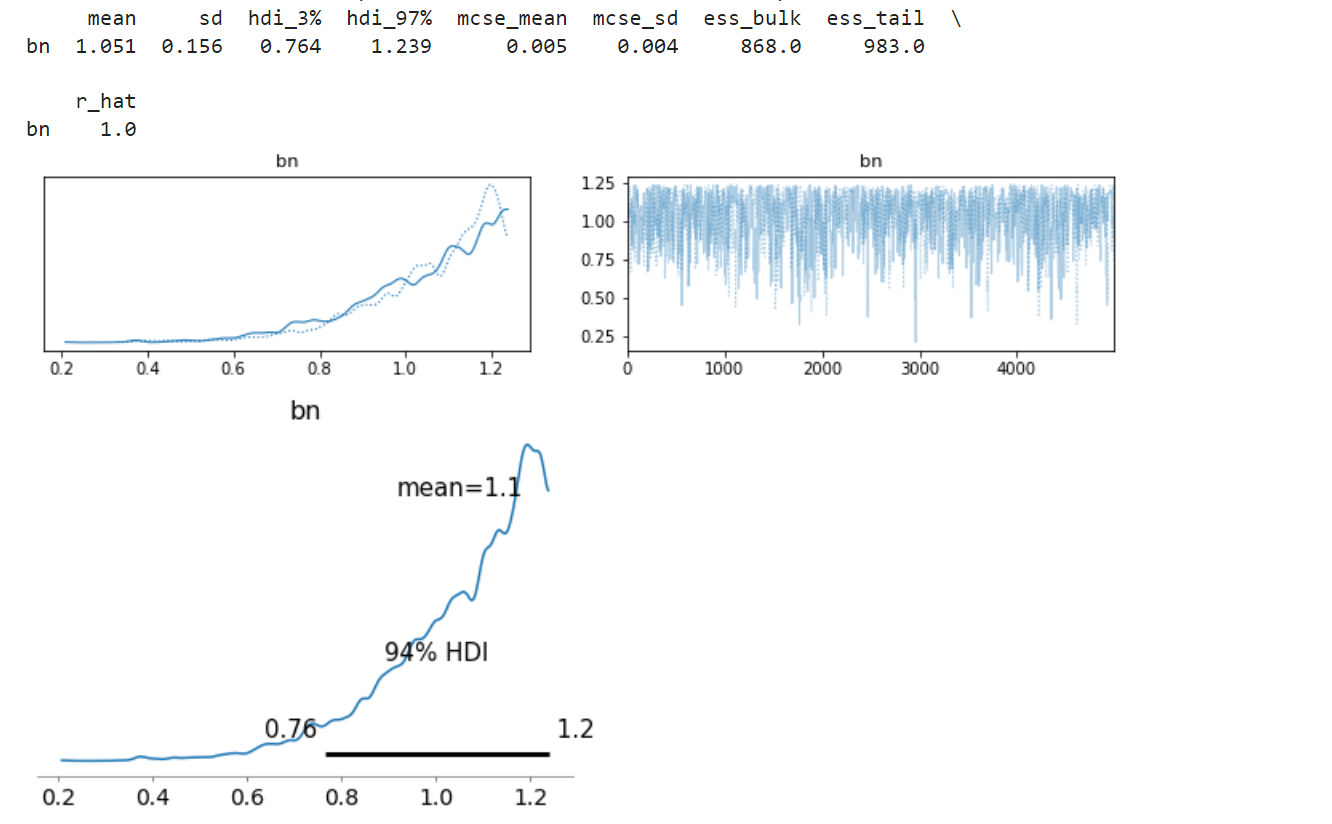
Test 1: **λ = ɣ \* N0(w) to align with the cumulative probability (equation 30)**

Test 2: **λ = 1**

Findings:

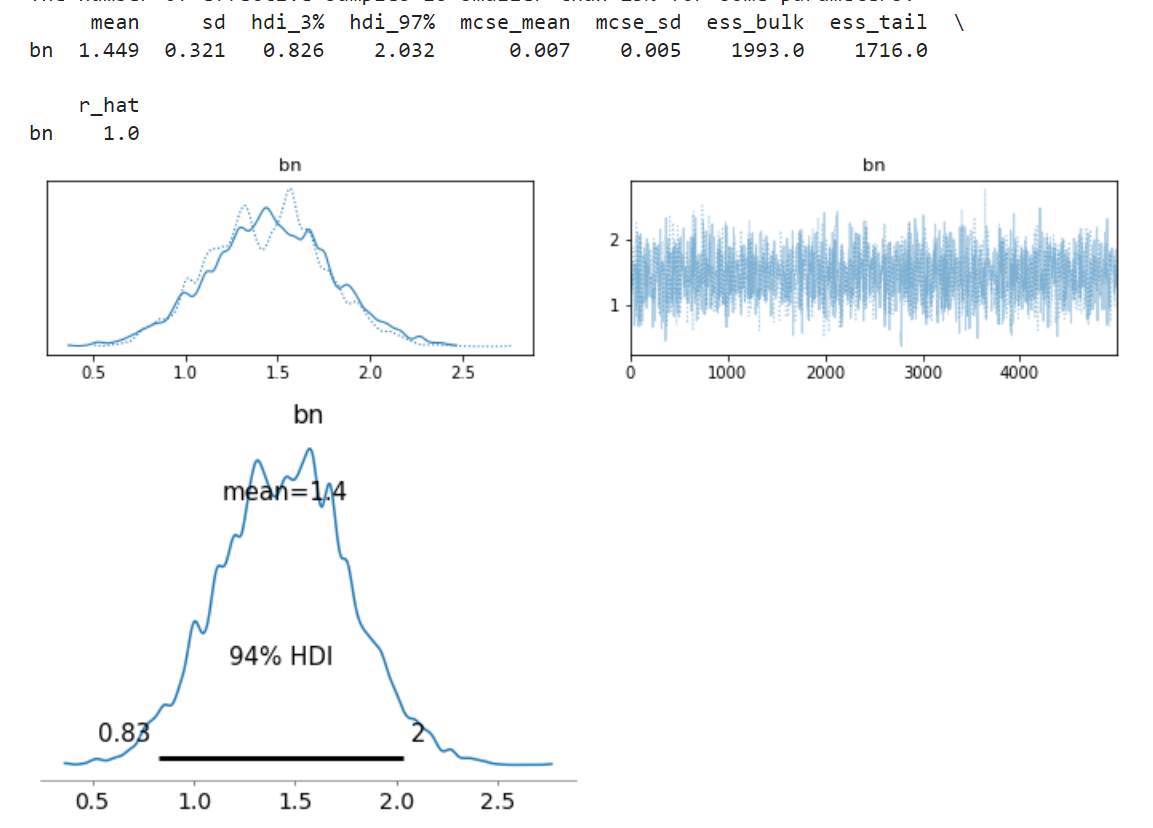
When we consider test 1, or when we transform the Nf variable to Nf / (**ɣ \* N0(w)**), then the algorithm converges but gives out senseless results:

Test with :



However, when we consider the scale parameter as being equal to one and leave the **ɣ \* N0(w) components out of the process, not only do we converge but we find a result similar to the custom likelihood distribution function.**

*Test with :*



This supports the actual link between the Maximum Likelihood process mentioned and the custom distribution even though the “standardization” that should be taking place with the division of the variable by **ɣ \* N0(w)** has to be put aside.

Remark: with this implementation, we obtain a higher number of divergences even if we still reach convergence, implying most likely a higher level of autocorrelation.

Test 3 and 4: Type III Extreme Value distribution

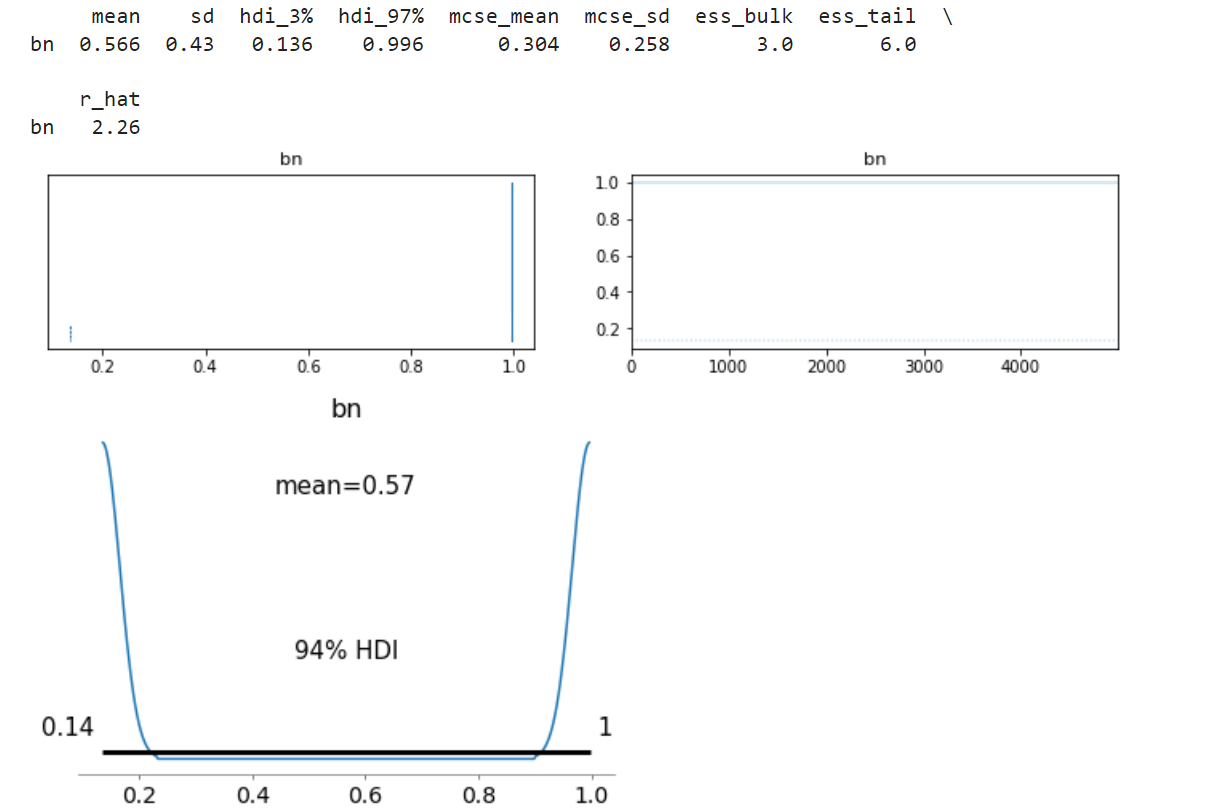
In the paper equation (36), (37) and (38), we see that the variable Nf is passed through a log transformation. We shall test the Extreme Value distribution corresponding to a log of a Weibull with the same two sets of parameters (scale = ɣ \* N0(w) or scale=1).

If a variable X follows a Weibull, distribution with shape parameter bn, then the corresponding Extreme Value density formula will be:

Location is still equal to 0 and the scale parameter is reflected in the variable itself: if we consider Nf only, then scale =1 and if we consider Nf/ ɣ \* N0(w), then scale parameter = ɣ \* N0(w).

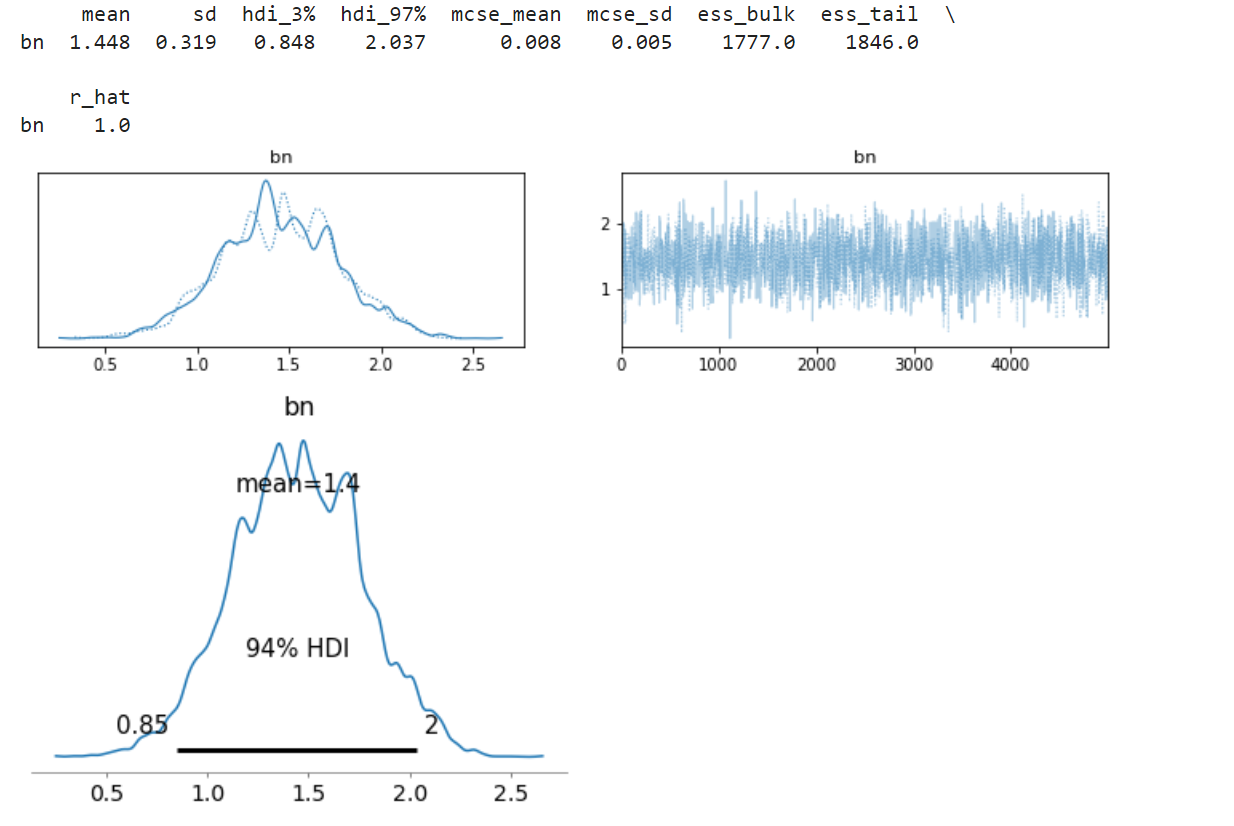
Findings: similarly with the Weibull density: taking the scale =1 (test4) yields a converging algorithm towards a Normal distribution with mean around 1.45 and standard deviation 0.35 for the parameter bn. The inclusion of ɣ \* N0(w) as the scale parameter brings the values of Nf towards values very close to 0 and the impedes the convergence process to take place.

*Test with GEV(0, ɣ \* N0(w), bn):*



We note a higher risk of autocorrelation in this case.

*Test with GEV (0,1, bn):*

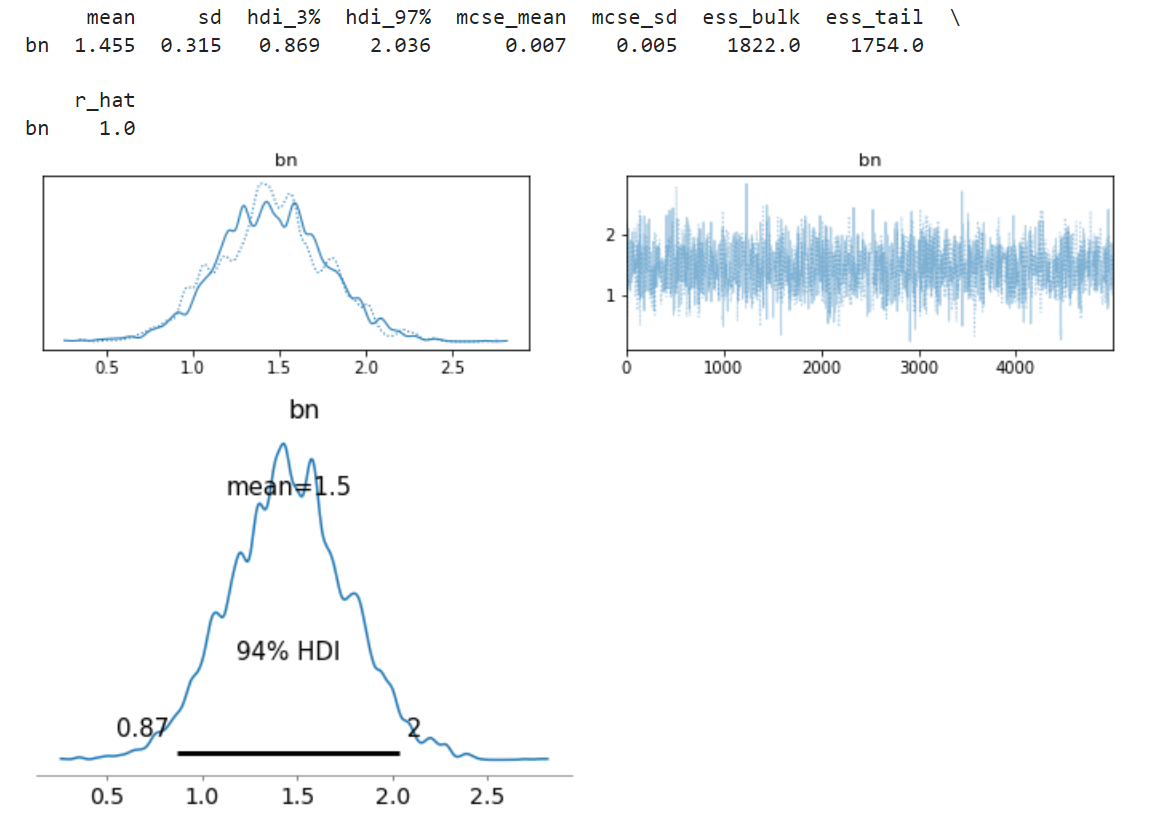


Test5: Lognormal distribution:

Due to the high similarity between the Weibull and the lognormal under certain conditions on the parameters, we also test a lognormal distribution:

Density formula for a Lognormal variable ():

Here we have: = and σ will be equal to the standard deviation of the initial value of bn from experimental data.



We also observe convergence towards the target distribution. Here the “scaled” element is encompassed by through the log transformation, so impacting the Nf variable to a lesser extent.

Sample size effect:

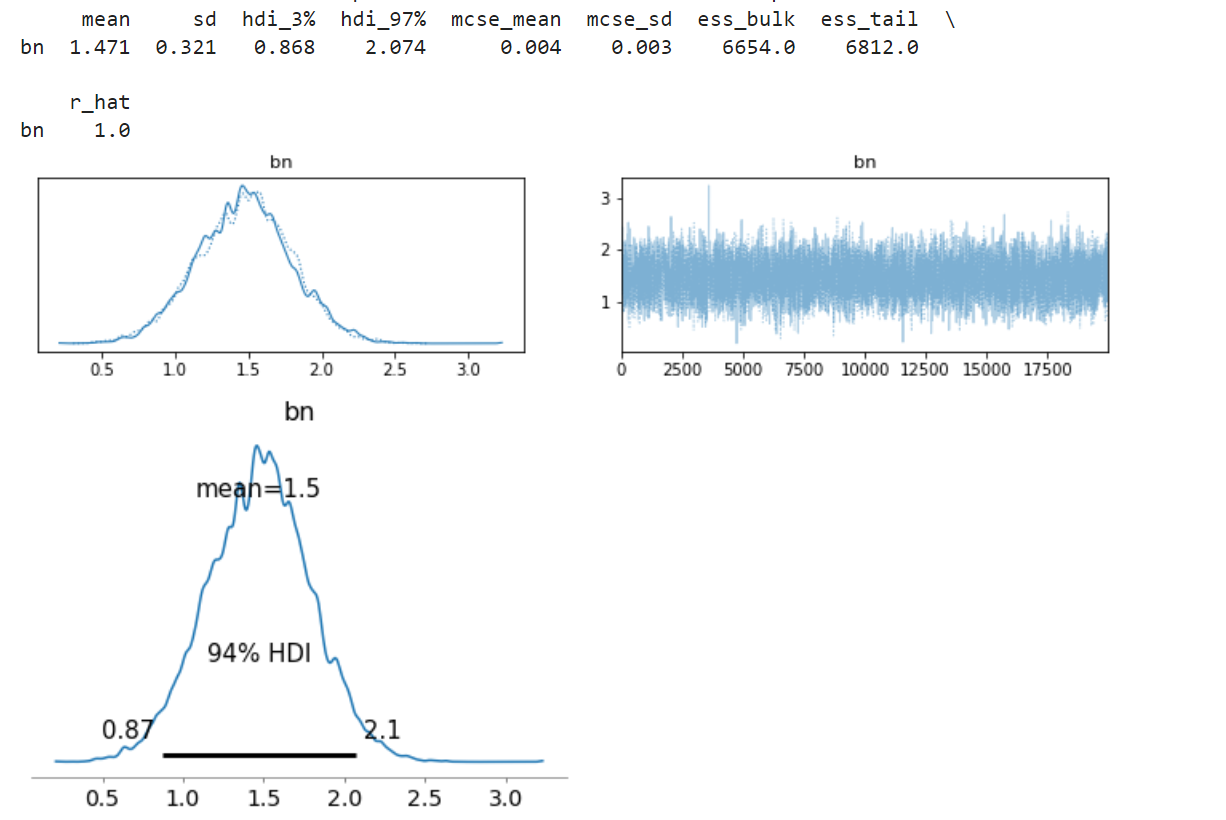
To confirm consistency of the process, we also want to make sure the consistency of the result is not brought by too small a sample size for observations, meaning too big an influence of the priors versus the likelihood.

To test this, we artificially increase the size of the observed sample without changing the distribution, by repeating the same values several times.

With the implantation conditions: 22 experimental Nf values and a 5000 sample simulated in the bayesian process, so a ratio of 0.44%. When we double or triple this ratio, the results stay the same.

Even when we multiply this ratio by 10 (passing to about a 1000 sample for observation and 20 000 simulated samples for example): we keep remarkably close to the same result for the mean : convergence towards a bn with 1.47 mean and 0.32 standard deviation (but as the sample was just repeated, there can be no change on the standard deviation.

*Test with 5% observed values vs sample size ratio:*



Even when we go as high as 10% ratio, the mean value shifts a bit from 1.45 to 1.5, but, in the increasing direction, meaning not in the direction of the observed values not the priors, indicating a real foundation to the convergence and an effective sampling process, not just a repetition of the priors.