

Master thesis : Design and assessment of stochastic rainfall generators for hydrological analyses

Auteur : Champailler, Stephane

Promoteur(s) : Dewals, Benjamin

Faculté : Faculté des Sciences appliquées

Diplôme : Master en science des données, à finalité spécialisée

Année académique : 2021-2022

URI/URL : <http://hdl.handle.net/2268.2/15202>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

Design and assessment of stochastic rainfall generators for hydrological analyses

Stéphane Champailler (s912550)

MSc in Data Science
Faculty of Applied Sciences
Academic year 2021-2022

Je remercie mes professeurs pour avoir accueilli mes questions ainsi que pour m'avoir éveillé à tant de choses nouvelles. Spécialement, Pierre Archambeau qui a eu la patience de répéter plusieurs fois la même chose.

Ce mémoire n'aurait physiquement pu être réalisé sans un soutien dont je ne m'explique toujours pas l'ampleur. Je remercie ma femme et mes enfants pour leur amour et leur patience infinie; mesdames Guisset et Derwa qui ont évité la chute; l'ONEM qui m'a laissé une chance; ma famille qui fut là.

Si jeunesse savait...

Contents

1	Introduction	4
1.1	Problem Statement	4
1.2	Hydrology and Other Definitions	4
2	Data Analysis	6
2.1	A first look at our data	6
2.1.1	Test Data	6
2.2	IDF curves	9
2.3	Stationarity of data	9
3	Review of Literature	12
3.1	STORM	12
3.2	A spatial rainfall generator	14
3.3	STORAGE	17
3.4	SHYPRE	19
3.5	Summary of models/simulators	23
3.6	Conclusion	27
4	IDF Curves	28
4.1	Using pre-made IDF curves	28
4.2	Building the IDF curves	29
5	Modeling	34
5.1	Working at the level of individual stations	34
5.2	Storm separation	34
5.2.1	Simple storm separation	34
5.2.2	SHYPRE	36
5.3	Seasonality	36
5.4	Storms per season	37
5.4.1	Via inter arrival time	37
5.4.2	Via number of storms	37
5.4.3	Conclusion	37
5.5	Storm modeling	38
5.5.1	Notes about distribution fitting	39
5.5.2	Storms durations	40
5.5.2.1	Regular distribution fit	40
5.5.2.2	Gamma mixtures	40
5.5.2.3	Hard mixture model	42
5.5.3	Storms precipitations	42
5.5.3.1	Regular distribution fit	42
5.5.3.2	Sum of Exponential Mixture Model	43
5.5.4	Correlation between duration and intensity	43
5.5.4.1	An uncorrelated simulation	43
5.5.4.2	Copula	45

5.5.4.3	Other approaches	47
5.6	Storm hyetographs	48
5.6.1	Flat hyetographs	48
5.6.2	Triangular hyetographs	48
5.6.3	Peak hyetographs	49
5.6.4	Probable hyetograph with MLE	49
5.6.5	Probable hyetograph with Metropolis Hasting sampling	51
5.6.6	Other approaches	55
5.7	Noise reinjection	55
6	Simulation	57
6.1	Simulation operations	57
6.1.1	General view of operations	57
6.1.2	Performances	58
6.2	Validation	58
6.2.1	Expectations	58
6.3	Global measures	59
6.4	Box storms distribution	59
6.5	Hourly statistics	64
6.6	IDF curves	70
6.7	Conclusion	71
7	Technical information	76
8	Conclusion	77
8.1	Future tasks	78

Chapter 1

Introduction

Following the flood of July 2021 near Liège, we were looking for a master thesis rooted in environmental sciences. I went to meet Pr. Benjamin Dewals about a project to make a rain simulator. Such simulators are interesting in themselves but are also very useful. Indeed their goal is to feed scenarios into hydrological models which are used to design infrastructures and policies to manage rain. In the present case, we would also like to have a simulator that simulates extreme rain (such as the episode of July 2021).

Rain is a challenging phenomenon in the sense that it is quite unpredictable and, at the same time, very familiar. From the point of view of the wide field data sciences, a simulator may seem rather simple as it doesn't usually use complex tools such as neural networks. A simulator is basically a model that produces data according to an example, it is an extrapolator. Although it sounds simple, we will see that it faces a few technicalities which are not easy to tackle, many of which reside in the fact one has to find ways to actually model a complex reality.

The rest of the document is written half as a presentation of a solution to the problem at hand, half as a report of various trials and errors that were done throughout its development.

1.1 Problem Statement

It's always very useful to define the goal of the job we want to accomplish. The “user requirements” phase remains crucial in any software development. So here are the thing we were looking at when we started this work.

We want to make a simulator that:

- is able to create artificial rain that looks like real rain
- this means that we should be able to reproduce the “usual” rain and the extreme rain
- the real rain, used as a reference, is the one that was recorded by DGO
- the simulator must be able to create artificial rain for a very long period (1000 years)

Notice that we don't have any formal or mathematical definition of what a real rain is.

1.2 Hydrology and Other Definitions

Hydrology is a field in itself. Its vernacular is sometimes a bit hard to grasp so we explain some of the most important terms here.

- Return period: the period of time after which an event will occur, on average. This is intimately linked to its probability in the sense that an event with a return period of 100 years has a probability of $1/100$ to occur each year. It is a misleading term because a return period of 100 years might be misinterpreted as an event that occurs once every 100 years. The actual

probability of the event occurring once is given by a binomial distribution: $\binom{n}{k} p^k (1-p)^{n-k}$. In our example, the probability of having *exactly* one ($k = 1$) event in 100 years ($n = 100$): $\binom{100}{1} 0.01^1 (1 - 0.01)^{99} = 0.3697$. That's much more than 0.01. Moreover, the definition allows for the event to occur more than once in the return period. So the probability of having the event to occur *at least once* is 1 minus the probability of it not occurring, that is: $1 - \binom{100}{0} 0.01^0 (1 - 0.01)^{100-0} = 0.634$. Given that return periods are often used to measure the recurrence of dangerous events, that probability is indeed very high.

- Storm: a period of time where it rains without stopping (so a storm might just as well be a shower, but we use the term storm mostly everywhere regardless of the variety of the physical phenomenon).
- Precipitation: rain that falls, usually expressed in millimeters of water.
- Rain intensity: quantity of water that rained (precipitation) per unit of time.
- Hourly rain: the precipitation that fell during one hour.
- Hyetograph: A graphical representation of rainfall over time. It is synonym with hyetograph.
- IDF: Intensity-Duration-Frequency. We usually display Precipitation-Duration-Frequency which are a different view on the data (intensity is precipitation over duration).
- DGO: Direction Générale Opérationnelle - Mobilité et voies hydrauliques (Wallonie)
- IRM: Institut Royal Météorologique (Belgique)

Chapter 2

Data Analysis

2.1 A first look at our data

The data we base ourselves on come from the *Direction générale opérationnelle de la Mobilité et des Voies hydrauliques*. They come in the form of hourly records of precipitation (time series). The unit of time is therefore the hour, expressed in integer numbers. The unit of precipitation is the millimeter of water and its finest resolution is a 10th of a millimeter. Data is therefore discrete. The time series were downloaded once from the official website¹ at the beginning of the project and used throughout.

Each time series corresponds to measurements done at a rain gauge. Many are available on the website and cover most of Wallonia. In the beginning of the project we selected roughly all stations that were close to Liège and inside the Ourthe, Amblève and Vesdre watersheds. But, as we will see later, we will only use a handful of them. Nevertheless, it is interesting to check many stations to get a picture of the overall quality of the data.

The time series don't always cover the same amount of time (because all the gauges were not all built at the same time). The most complete ones go back to 2002 while others go back to 2007 or even sooner. Some of them contain missing values (due to error in measurements, gauge dysfunction, etc.) but that's a minority. The vast majority last up to the end of 2021. The available time series are summarized in table 2.1.

In the end, we selected the time series which were well behaved, that is: going from 2002 to 2021, with less than 100 missing data (represented by NaN's). As can be seen on the table, some stations have a lot of NaN's. That selection was made to ensure we had an homogeneous data set. The 100 threshold was determined to separate obviously problematic stations from others. The 2002-2021 period was chosen because it is the longest one and is quite frequent in the data set. We replaced the rare remaining NaN's by zeros (a single NaN represents an hour, so $\frac{1}{(2021-2002+1) \times 365 \times 24}$ of one station's data set). The stations we kept are depicted on figure 2.1.

We assumed that all measurements were done according to the same procedure. This might not be the case: some instruments may have slight defects, strong wind may impact measurements, etc.

2.1.1 Test Data

In this work, we made most of our computations on three stations: Bastogne, Robertville and Wavre. We reduced our number of stations this way because we worked on each station individually and the computation times were long enough to prevent us from iterating quickly enough in our development over all the stations.

The stations were chosen to have respectively the maximum, the mean and the minimum quantity of rain. Note that the same characteristics (maximum, mean, minimum) hold when we look at the number of storms per year (figure 2.3, note that we will discuss how we count the storms later).

¹See <http://voies-hydrauliques.wallonie.be/opencms/opencms/fr>

Name	First-Last Year	NaN's	Name	First-Last Year	NaN's
Bastogne		-	Soignies		7
Battice		41	Solre		5
Beauraing		30	Steffeshausen		44
Bertrix		73	Sugny		18
Blaregnies		10	Tailles		10
Bouillon		24	Ternell		-
Boussu-En-Fagne		7	Torgny		1
Bousval		8	Uccle		57
Chievres		-	Vielsalm		74
Ciney		16	Waremmes		-
Comines		2	Wasmuel		44
Coo		24	Wavre		-
Crupet		4	Anseremme	2007-2021	75
Cul-Des-Sarts		-	Arlon		2552
Dergneau		-	Athus		12825
Enghien		18	Aubange		133
Erezee		-	Awans	2011-2021	-
Fratin		11	Balmoral		45929
Gedinne		15	Bierset		98528
Gemmenich		29	Butgenbach	2003-2021	70
Jalhay		80	Chatelet	2018-2021	-
Kain		-	Coo		259
Lanaye		4	Croix-Scaille	2018-2021	3
Landenne		26	Daverdisse		152
Libin		65	Flamierges		65732
Louveigne		14	Florennes		411
Marche		13	Gerpennes		149
Meix-Le-Tige		89	Helecine	2010-2021	59
Modave		32	Ligny		3098
Mornimont		97	Lillois	2014-2021	-
Namoussart		-	Louvain	2018-2021	-
Nassogne		88	Mean	2016-2021	-
Ortho		12	Momignies		949
Orval		-	Monceau		110
Ouffet		17	Mouscron	2004-2021	41
Peruwelz		26	Rouveroy	2003-2021	3
Perwez		-	Saint-Hubert	2005-2021	161
Petigny		7	Sart-Tilman		504
Plate		-	Selange		1536
Rachamps-Naville		18	Somme-Leuze		53296
Robertville		7	Spa	2017-2021	-
Rochefort		50	Straimont		128
Roisin		24	Trivieres		128
Saint-Gerard		-	Tubize	2014-2021	-
Sankt-Vith		-	Vedrin	2003-2021	194
Seneffe		-	Vresse	2004-2021	-
Senzeilles		-			

Table 2.1: Data quality report. When not given, the first/last years are 2002-2021.

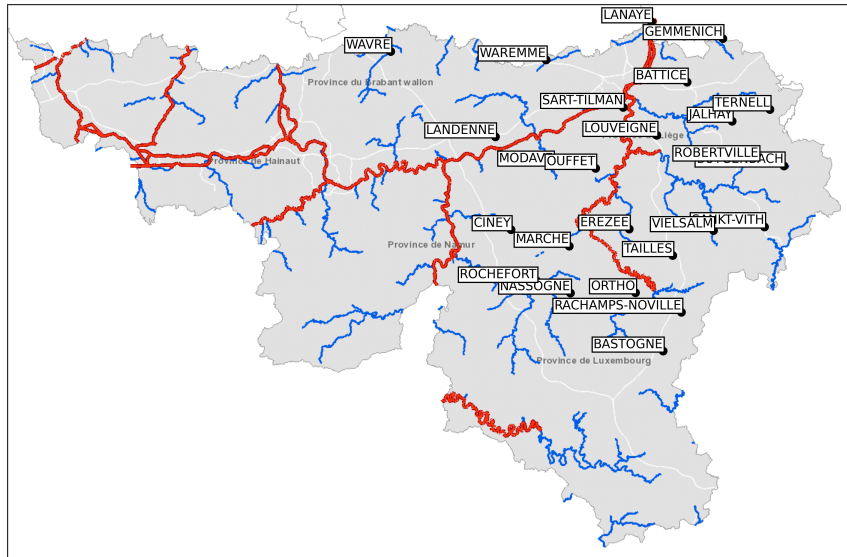


Figure 2.1: Map of the selected measurement stations. The rivers map comes from the WalOnMap website.

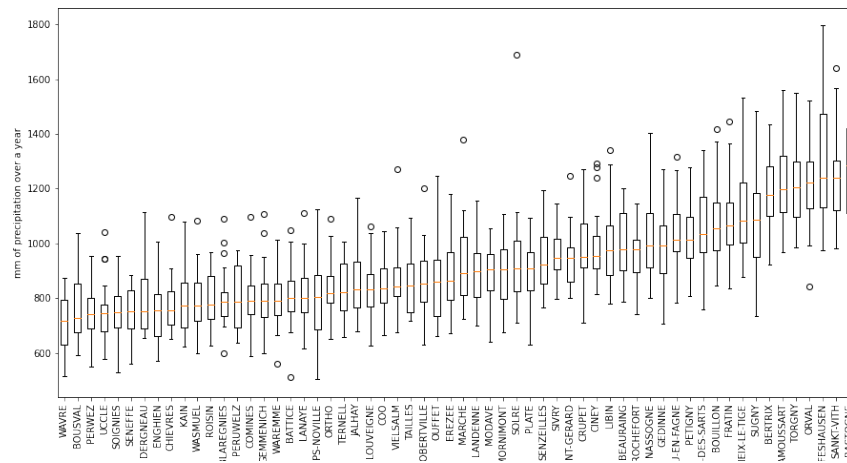


Figure 2.2: Rainfall at different measurement stations measured over twenty years

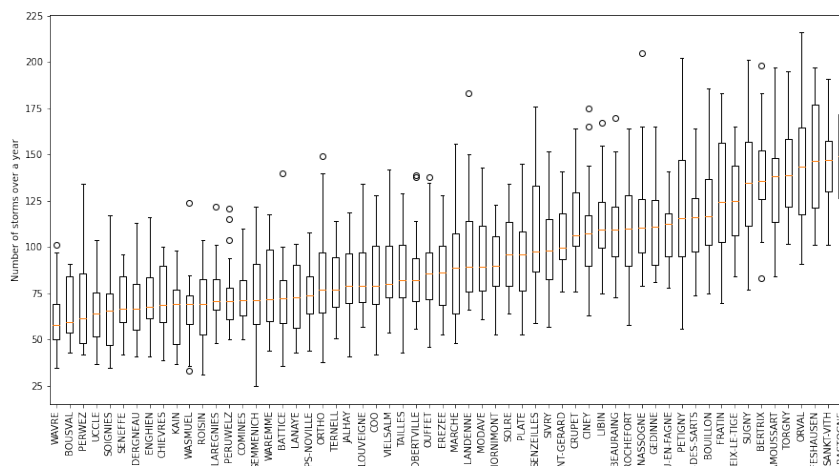


Figure 2.3: Number of storms at different measurement stations measured over twenty years

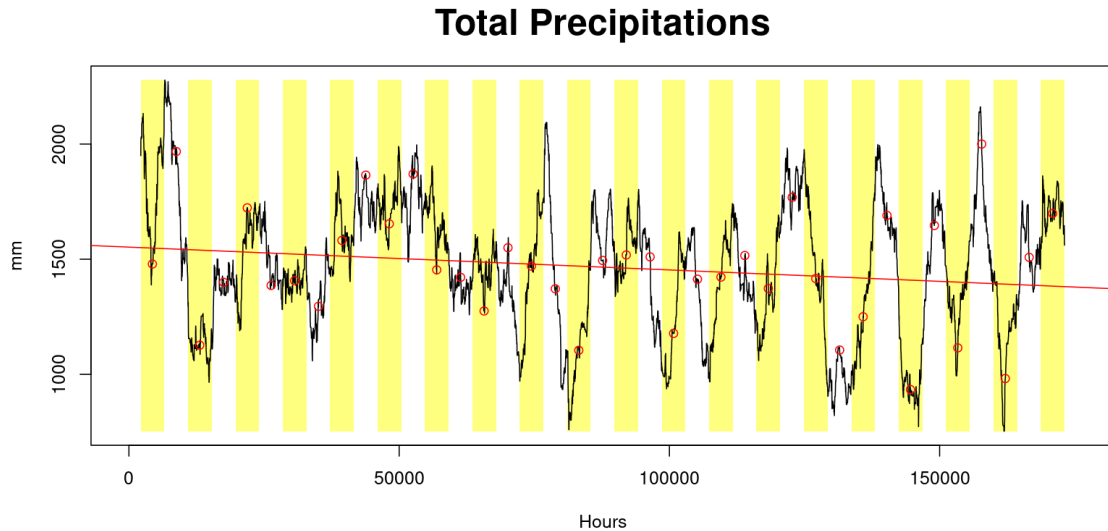


Figure 2.4: Six months rolling sum over hourly measurements (Bastogne data). Yellow and white backgrounds denote summer (April to September) and winter (October to March). Red dots are value at center of season. Red line is a linear regression over rolling values.

2.2 IDF curves

We also have access to IDF curves provided by the *Institut Royal Météorologique*. As we will see later, IDF curves are used to inform us on the recurrence of the most intense rain event over time. IDF curves will be thoroughly discussed in chapter 4.

IDF curves are computed for many places which correspond to the measurement stations of the time series above. The correspondence is not exact though.

2.3 Stationarity of data

Stationarity of the process behind the rain is important to us because we compute:

- storm frequencies over long time ranges (20 years)
- storm characteristics (duration, intensity) over long time ranges
- both of them with models that are, most of the time, not parametrized over time.

Therefore we make the assumption that the underlying process is constant over those ranges. To make sure that assumption holds, we look at some indicators.

First we look at how the mean of hourly precipitations behaves over the whole period. For that we compute a rolling mean over a period of six months. We chose six months because this is related to seasonality. As can be seen on figure 2.5 there is a lot of variability in the mean in relative terms, however, the mean is stable in the long term (cf. linear regression which is mostly horizontal). A closer look at the same figure indicates that there seems to be some cycling over the seasons, albeit not quite clear. To assess that, we analyze two additional values: the rolling mean and standard deviation of the hours where it rains (in the previous chart all hours, even the dry ones, were taken into account). On figure 2.5 it is clearly apparent that seasonality is at play. The charts indicate that the rain intensity (mean) and “structure” (standard deviation) are more dynamic in summer (we don’t say it rains more in summer, the previous chart shows it is not the case).

Now, if we look at linear regressions over all the charts (red line), we can see there is a trend but it is very weak. When we compare our data to those published on I.R.M. web site, there was no obvious difference and no obvious influence from climate change (the latter being suggested by Vyver 2012).

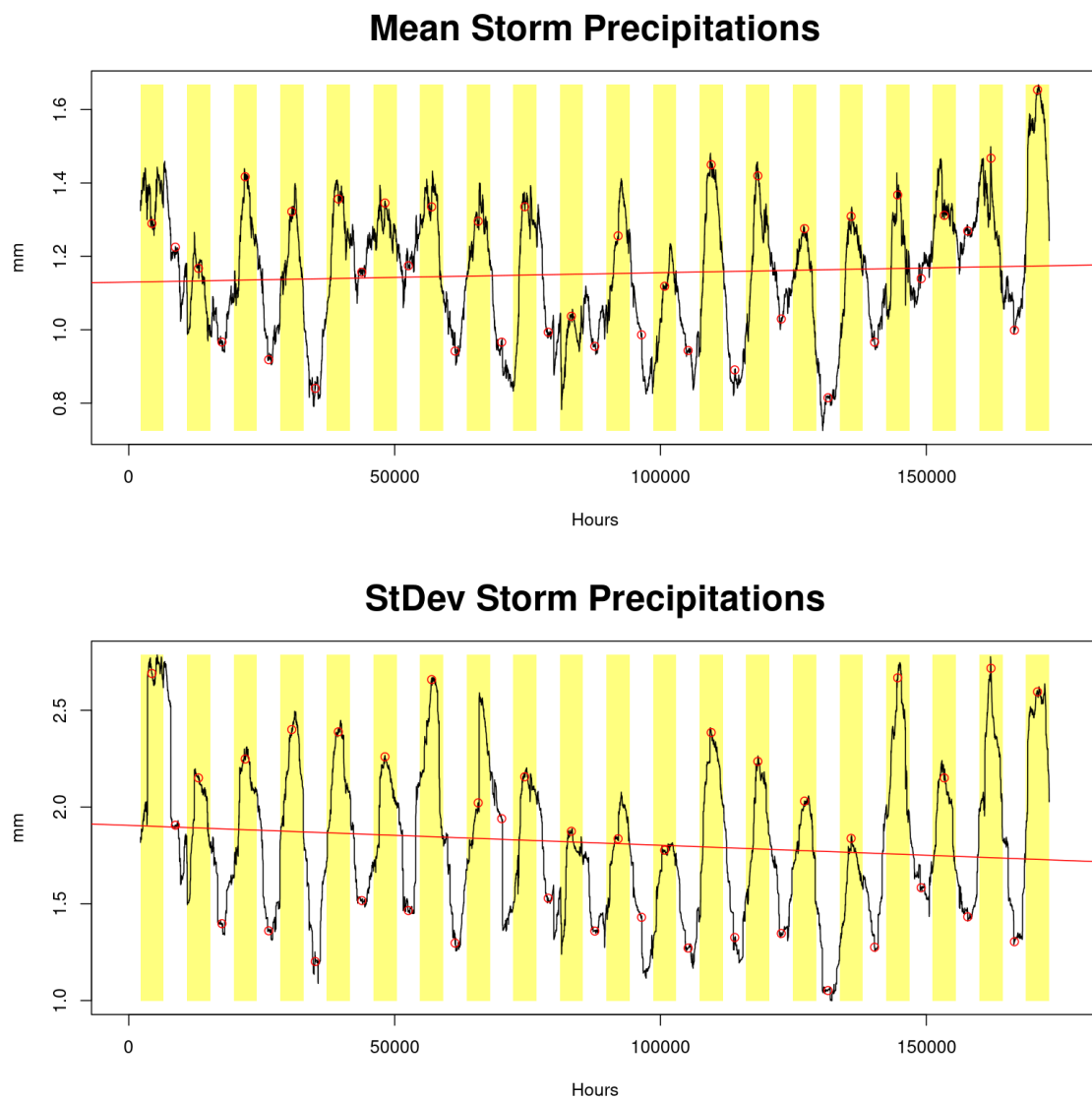


Figure 2.5: Six months rolling mean and standard deviation over precipitations (Bastogne data). Yellow and white backgrounds denote summer (April to September) and winter (October to March). Red dots are value at center of season. Red line is a linear regression over rolling values.

Consequently, we can conclude that the stationarity is sufficient in the long term but that seasonality is strongly at play. It means that the model we will build can be used every year but that inside a year, seasonality must be taken into account.

Chapter 3

Review of Literature

3.1 STORM

The STORM (Singer, Michaelides, and Hobley 2018) simulator is empirical-stochastic, that is, based on distributions computed over data. However, the definition of these distributions is left to the user. We would have expected something that would start from a time series and then build PDF automatically.

The general principle of STORM: find the total precipitation for a year, then compute as many storms as necessary to reach that total. For each storm, its duration is picked randomly (from a PDF) and then its intensity is derived from it via one of the I/D curves. The choice of the I/D curve depends on several parameters. Storms are “sprinkled” on a map at random. So for each point of the map we know the total quantity of water that fell on it. Evapotranspiration is then applied. The time resolution is one minute, so at the end of the simulation one gets a very detailed data set.

The STORM model is composed of several things (see 3.1):

- The total precipitation for a year P_{Total} is given by a PDF. STORM will produce storms until P_{Total} is attained or exceeded.
- PDF of inter-arrival times of storms, modeled as Generalized extreme value distribution (see code line 827 and article figure 3). We wonder how the inter-arrival time is respected provided there is a constraint over P_{Total} .
- A PDF for storm durations (a generalized extreme value distribution is used in the code, there’s no explanation as to why it was chosen like that).
- Intensity P_i - duration P_d curves (it is not clear to us if these are actual IDF curves) are derived like this:
 - For the first I/D curve authors took P_i maximum for each observed P_d .
 - Smoothed (made it continuous) that curve with second order polynomial (locally weighted scatterplot smoothing a.k.a. LOWESS)
 - A distribution is fit over the curve obtained in the previous step. This happens to be a double exponential such as $P_i = \lambda \exp(-0.508 \times P_d) + \kappa \exp(-0.008 \times P_d) + c$
 - Then I/D curves for less intense storms were derived by simply reducing λ, κ, c progressively (following percentiles).

The authors don’t provide the code that generated P_I, P_D . Moreover the authors don’t explain why the double exponential was chosen. The authors acknowledge the fact that this procedure is somewhat manual. Each curve is given a probability which will control how it will be chosen (note that here the authors set these probabilities manually).

- Storm centers locations: places where the storms can occur, these will be picked uniformly on a random grid. The grid is defined over a given region (basin). We guess they chose a grid so

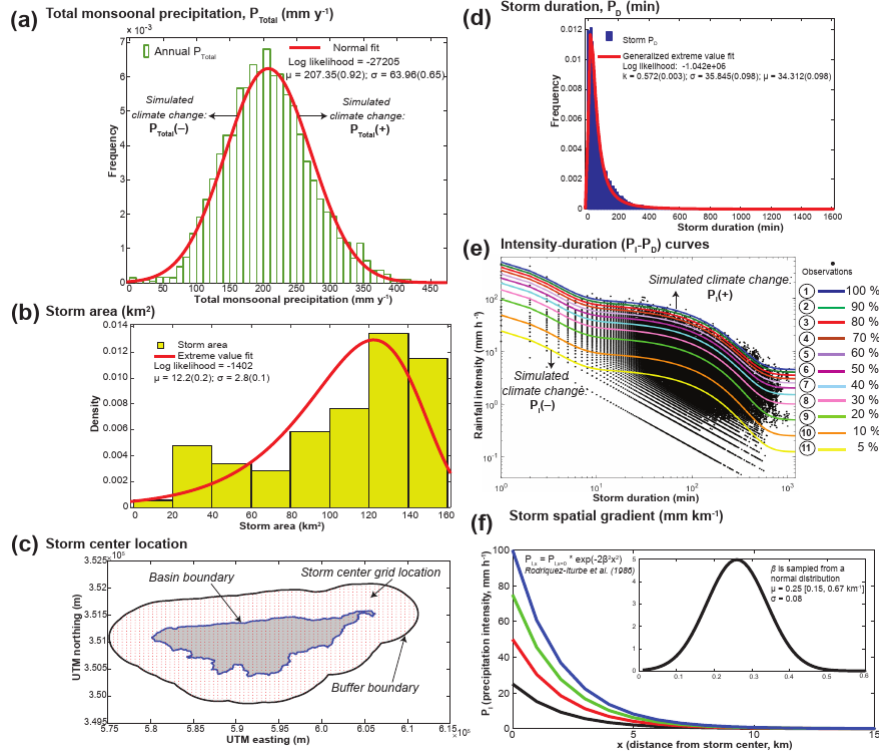


Figure 3.1: STORM main components (copied from the article)

that they can easily enumerate the locations where a storm can occur, but that's crude. Note also there's a boundary that allows us to generate storms that just barely touch the basin area, that is, storms of which the center is not in the basin. The grid resolution is 1km. Therefore, all the storms will be located at a resolution of 1km.

- Storm area (km²): PDF giving the size of area (km²) covered by simulated storms. Modeled as Generalized extreme value distribution in the code (line 846) and on figure 2b.
- Storm spatial gradient: decline of probability P_i with distance from storm center. The curve is given on figure 2f. This allows to model storms in three dimensions: x,y and intensity, with more intensity near the center of the storm. Therefore, the storm shape is a disc (see masks in code at line 868).
- Potential evapotranspiration: assumed to be constant over the whole watershed. Based on observations aggregated over months, giving daytime and nighttime evapotranspiration.
- Orographic groups: will alter the way the I/D curves will be selected in function. Storms occurring in more elevated regions will be less intense than those in lower regions.
- STORM can work on the basis of one year without season or one year composed of two seasons. In that case, most of the parameters are duplicated to account for seasonality.
- Climate change is modeled year on year by changing wetness P_{Total} mean by $\mu \pm \mu \times \Phi$ and by changing storminess by using $P_I \pm \psi P_I$ instead of P_I .

Model evaluation based on these summary statistics, computed at several gauge station:

- Number of storms per year
- Average storm total
- Total annual precipitation

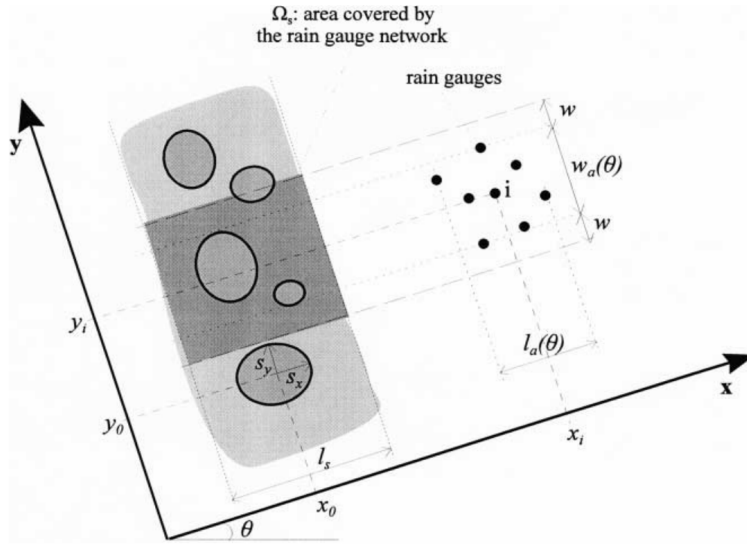


Figure 3.2: Overview of the modeled storms (copied from the article)

These statistics were compared with real data. The main result is that the model over predicts a bit and that it displays a good variety in the storms it predicts. We think that the number of storms per year and total annual precipitation are weak indicators of quality because they are driven by the quality of the fit of the PDF. So they measure how good the fits are, not the full simulation. Moreover, the article (figure 10) doesn't say how many simulations were done for the assessment.

We now quickly review the applicability of STORM to our data:

- We don't have information about the storm areas
- Our resolution is hours, not minutes
- We don't have information about orographic groups
- We don't have information about climate change trends

So STORM is not usable for us.

3.2 A spatial rainfall generator

This simulation (Willems 2001) is the most precise one of this review. It works at a local (city) level but simulates the shape of storms quite precisely. It also models the movement of storms in the sky. The data set resolution is extremely important here. It is composed of time series measured at 12 gauge stations located in the area of Antwerp and spanning about 100 square kilometers. The time series resolution is in *minute* and tenth of millimeters.

In the model, the main object is the rain cell (see ellipsis on figure 3.2). Each of these is tracked and modeled individually. Rain cells are grouped in storms. Rain cells are represented as a Gaussian distribution where the center represents the point of maximum rain intensity. Rain cells travel above an array of gauges.

- The model assumes rain cells all go in the same constant direction θ at the same constant speed u . This simplification makes sense because we're working at a city scale, so these properties of the storm may well stay constant over the duration of the storm.
- The number of rain cells in a storm is studied in relation with the rainstorm area Ω_s . The rainstorm area is given proportional to the width $w_a(\theta)$ (measured orthogonally to θ) of the gauges area times the length l_s (along the direction θ) of the rainstorm. That is, $\Omega_s = l_s(w_a(\theta) + 2w)$ where w is additional space around the gauges area to make sure one includes storms whose

centers are not in the gauges area but which overlap the area. The length of the storm is given by: $l_s = ud_s - l_a(\theta)$ (l_s length of the storm in the direction θ , $l_a(\theta)$ length of the gauges area in the direction θ , d_s duration of the complete passage of the storm over the entire gauge area).

- A spatial Poisson process is used to model the number of points in the Ω_s area, so a parameter λ is determined as $\lambda = \frac{\text{Avg number of cells}}{\Omega_s}$. Once we have λ we generate number of rain cells in the rain storm as a random number coming out of a Poisson process distribution $P(n(S) = k) = \frac{(\lambda\Omega_s)^k \exp(-\lambda\Omega_s)}{k!}$, the coordinates x and y of each rain cell in the storm are then taken out of a uniform distribution.
- Inter-arrival time of rain storm is modeled with a sum of two exponentials probability distribution.
- Each rain cell is then looked at independently of the others:
 - A rain cell intensity r_{\max} is modeled as a normal distribution shape (bell), with maximum rain intensity at the center of the bell.
 - A rain cell is assumed to move in a constant direction θ , with a constant speed vector u . All rain cells inside a storm will move along that direction (see above)
 - A rain cell starts at position (x_0, y_0) , time is measured from t_0 .
 - The space coordinates are set in a reference framework for which the x-axis is oriented in the direction θ .
 - The rain cell overall shape evolution is driven by a parameter γ . Depending on the sign of that parameter, the rain cell either grows or shrinks (decay).
 - The intensity of the rain at a given point (x, y) , at time t is given by: $r(x, y, t) = r_{\max} \exp \left\{ - \left[\frac{[x - (x_0 + \mu(t - t_0))]^2}{2s_x^2} + \frac{(y - y_0)^2}{2s_y^2} + \gamma(t - t_0) \right] \right\}$

Fitting the parameters:

- The goal is to determine the various parameters of the equation above: $r_{\max}, \gamma, s_x, s_y, l_s$. One also needs the speed and direction of the cell movement μ, θ (which is assumed the same for all rain cells) as the frame of reference is built on it. The values of the parameters will be determined by examining all rain cells of a recorded storm. So the first step is to analyze the time series to isolate the rain cells. This is done with an algorithm analyzing the increasing flanks, peaks and decreasing flanks at each gauge to detect peaks:
 - First, minima, peaks and flanks are identified in the time series. This is akin to computing a first derivative. We guess the difficulty is that a proper time scale has to be used. Too small and there are plenty of minima/flanks because of noise, too big and one may miss small storms
 - Second, detected peaks (see previous step) are connected into rain cells. Four steps are necessary (barely reproduced here). The general idea is to compare what happens at one gauge station with what happens at a close station; if they're close enough, their spatiotemporal information may be similar.
 1. A priori numbering of rain cells: The peaks of the time series with the highest number of peaks are numbered. Then peaks in other time series are connected to them based on minimum time distance.
 2. Adaptation of double rain cell numbers: When numbering cells in one time series, it is possible that two rain cells are given the same number (because they're connected to one big encompassing one). So additional tests are done to prevent that
 3. Check for the minimum proximity of rain cells with the same rain cell number: This is to further confirm that a rain cell is connected correctly. Basically, one ensures that the average distance between one rain cell and those with the same number is smaller than the average distance with all the other cells.

4. Further rain cell number identification for rain cells with unknown numbers.

- One computes the μ, θ . By looking at which time peak intensities pass through the different gauges, the authors explain they can rebuild both parameters (we understand this works under the hypothesis that s_x is aligned on θ).
- When this is done s_x, s_y are estimated in turn. For s_x , the idea is to remember the x-axis goes in the direction of the movement. So, by only looking at the distance between a gauge and the peak of the rain cell, the measured intensity and the time, one can fit a normal distribution. Note that since the spatial extent of a rain cell changes over time (because of the parameter γ), a simplification is done here (Taylor approximation for fluid; which we didn't research any further). If we understand correctly, this means assuming the spatial extent remains constant over time (this is confirmed in section 5 of the article). We were not able to reproduce that computation.

Once parameters are estimated for many rain cells, it is possible to fit probability distributions over each of them. With these, one will be able to create simulated rain cells and rain storms over any period of time. The author also computes a correlation matrix between the different parameters values. Interestingly, the s_x parameter (spatial extension of the rain cells) is highly correlated to u (speed of the storm, hence of the rain cell). We have no explanation for that phenomenon.

Once the PDF are known it is easy to generate storms and rain cells:

- Using the inter-arrival formula it is easy to compute the number of storms over a given period of time.
- Using the PDF one can choose random values for s_x, s_y, μ, θ
- One can then compute d_s from the previously shown equation $l_s = ud_s - l_a(\theta)$. (see page 135, the calibration process is different than the process by which the model is derived). Then one can compute Ω_s and from there the number of rain cells in the storm.
- Then for each rain cell, using the PDF one can choose random values for s_x, s_y and from a uniform random distribution, the position of each rain cell in the storm.

The validation of the simulation is done through several means:

- Intensity-Density Functions for various aggregation levels (t=1 year, 1 month, 27 years). Oddly, for the 27 years period, the author compares what the generator produces once calibrated with data for the city of Antwerp to IDF curves computed on basis of data of Ukkel. Ukkel having a longer time series than Antwerp, the idea is that it allows to see more extremes.
- Distributions of rainfall intensities at different aggregation level (e.g. 30 min.), with an exponential quantiles plot.
- Auto-correlations: the author doesn't explain who it makes comparison.

Applicability to our data set:

- Obviously, the time resolution of our data set is way too high to be able to use the proposed model.
- Moreover, the rain cell separation is quite involved and we wonder how one can make sure it works correctly if one only has access to time series.

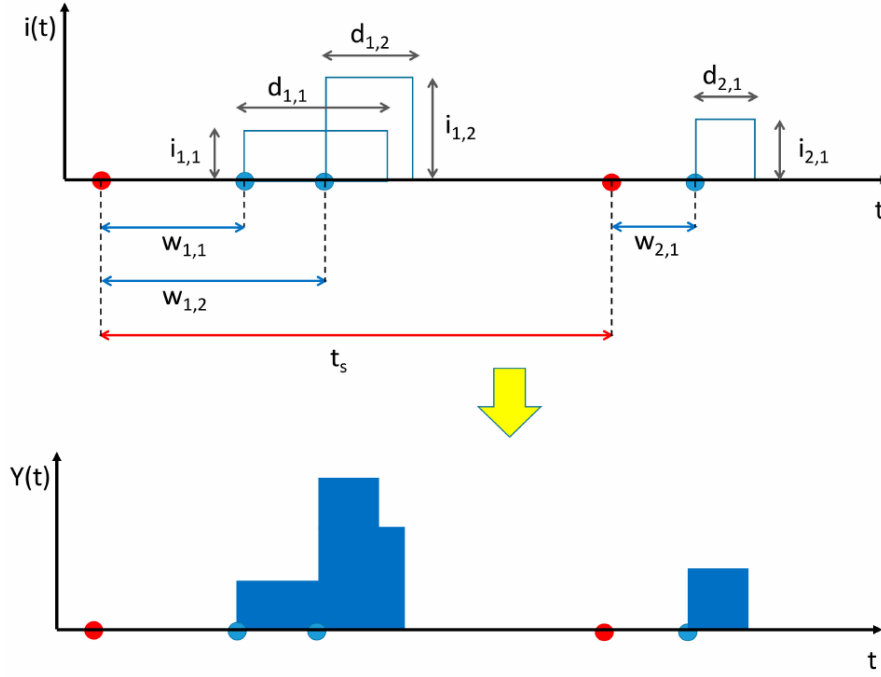


Figure 3.3: Neyman-Scott Rectangular Pulse model

3.3 STORAGE

This STORAGE (De Luca and Petroselli 2021) simulator is quite different in its fitting procedure. Indeed, instead of regular PDF fitting, the simulator first generates thousands of plausible parameters set. Using these, simulations are performed. These simulations are then compared to real data. The parameters corresponding to the best one are stored in a “reservoir” (hence the name of the simulator). When a new simulation is requested, the parameters are picked at random in the reservoir and the simulation is performed.

The model of the simulator is the following:

- The basis of the model is the basic Neyman-Scott Rectangular Pulse model (see figure 3.3). A storm is composed of several rain cells. Each cell is a shower of constant intensity over time. Then cells are added together to form a complete storm. Parameters are:
 - The inter-arrival time of storms T_s with $P[T_s \leq t_s] = 1 - \exp(-\lambda t_s)$
 - Number of rain cells in a rainstorm: geometric r.v. with $P(X = x) = (1 - p)^{x-1}p$ where the number of cells is x , mean $\theta = 1/p$.
 - waiting time between a rain cell and the start of the storm, exponential distribution. $P[W \leq w] = 1 - \exp(-\beta_W w)$ (this comes from Cowpertwait 1991).
 - Intensity and duration of a rain cell, both exponentially distributed. $P[I \leq i] = 1 - \exp(-\beta_I i)$ and $P[D \leq d] = 1 - \exp(-\beta_D d)$
 - Parameters $\beta_i, \beta_d, \beta_w, \theta, \lambda$ are then fit on objective function (made of the various measures of success of the model, see below)
- To include seasonality, one usually duplicates the parameters in 2 or more seasons. According to the authors, this poses some two problems:
 - There are more parameters to fit (since parameters are duplicated by season).
 - There is less data to fit each parameter (since data is split in seasons)
- So the authors introduce another way to represent the five parameters. To that end, they introduce a goniometric function: $p(t) = p_0 + \sum_{n=1}^K A_n \cos(2\pi n/T_y t + \phi_n)$ with p_0 the mean, A_n

the amplitude, ϕ_n the phase shift, K the number of harmonics. With it they give formulas for their basic parameters.

- For the mean value of the inter-arrival time: $\frac{1}{\lambda(t)} = \frac{1}{\lambda_0} + \left(\frac{1}{\lambda_0} - \left(\frac{1}{\lambda} \right)_{\min} \right) \cos(2\pi \frac{t}{T_y} + \phi_{1,\lambda}) + \xi \left(\frac{1}{\lambda_0} - \left(\frac{1}{\lambda} \right)_{\min} \right) \cos(2\pi \frac{t}{T_y} + \phi_{2,\lambda})$ where $\frac{t}{T_y}$ gives how far we are in a year (assuming we cycle every year), $\phi_{k,\lambda}$ are the phase shifts, $\left(\frac{1}{\lambda} \right)_{\min}$ the smallest inter-arrival time between two storms. In this formula, $K = 2$.
- For the mean value of intensity and duration of a rain cell: $\frac{1}{\beta_I(t)} = \frac{1}{\beta_{I,0}} + \left(\frac{1}{\beta_{I,0}} - \left(\frac{1}{\beta_I} \right)_{\min} \right) \cos(2\pi \frac{t}{T_y} + \phi_{1,\beta_I})$ and $\frac{1}{\beta_D(t)} = \frac{1}{\beta_{D,0}} + \left(\frac{1}{\beta_{D,0}} - \left(\frac{1}{\beta_D} \right)_{\min} \right) \cos(2\pi \frac{t}{T_y} + \phi_{1,\beta_D})$. In this formula, $K = 1$.
- etc.

So if we look at the mean $\frac{1}{\beta_D(t)}$ we see we need 3 parameters to express it “seasonally”: $\beta_{D,0}, \phi_{1,\beta_D}, \left(\frac{1}{\beta_D} \right)_{\min}$. If we had four seasons, we would still have 3 “goniometric” parameters but, using parameters duplication, we would have four. The more we would split the year, the more we would have parameters and the better the advantage for the goniometric function provided one doesn’t need to raise K too fast. It is interesting to note that the authors don’t explain how they choose the values of K .

The optimization procedure to figure out all the parameters is brute force:

- The authors generated 50000 parameter sets randomly (parameters as random uniform variables with assigned range of variation, see 3.1.2)
- For each set, a run of simulation of 200 years, with a resolution of 1 minute was generated
- Summary statistics of each run are computed:
 1. mean annual precipitation (MAP)
 2. mean annual number of wet days (rainfall $\geq 1\text{mm}$)
 3. parameters of IDF curves for rainfall duration of 1 to 24 hours. The IDF curves are modelled as $h_T(d) = a_T d^{n_T}$ where d is the duration and $h_T(d)$ is the quantity of precipitation. The parameters are a_T and n_T . There are several curves for $T = 2, 5, 10, 50, 100, 200$ years. The fitting procedure is not explained.
 4. mean values rainfall for each season (december-january-february; march-april-may;...)
- Keep only the parametric sets for which the 200 years runs’ summary statistics have variation ranges comparable to those of *all the rain gauges* (section 4). The comparison procedure is not explained.

Once the “reservoir” of precomputed parameters is filled, new simulations can be run for new sets of values of the four summary statistics above.

- The simulator will find, in its database, the previous run that matches the input summary statistics. The match quality is computed with *objective functions*. Those are:
 1. $\sum \frac{|a_i - a_i^*|}{a_i} + \sum \frac{|n_i - n_i^*|}{n_i}$: in other words, we see that the IDF curves parameters are close to each other. The starred term is the input, the non starred is the one coming from the recorder parameters set. The choice of the formula is not explained.
 2. $\frac{|MAP - MAP^*|}{MAP} + \frac{|nwd - nwd^*|}{nwd}$: where MAP is Mean Annual Precipitation and nwd is the number of wet days (over a year).
 3. The sum of the two previous ones.
 4. $\sum_{i=1}^4 \frac{|s_i - s_i^*|}{s_i}$ where s_i is the mean rainfall over season i . to which we had the previous one.

- Once the objective functions are computed, the simulation can go on by picking the best parameters set according to some strategies:
 - Ranking the best parameters set on the basis of their OF's values (several parameters set may have close OF values). A subset of S parameters set will be used to generate all the years of a simulation. Each of the parameters set will be used more or less often, depending on its OF value. The frequency of use of a given parameters set is $f_i = \frac{1/O_{F_i}}{\sum_{i=1}^S 1/O_{F_i}}$
 - Mixing: taking the param set which performs best on the three first formulas of OF. Then give a frequency based on the fourth formula of OF. Variant: pick the best on all the formulas of OF and, again, base frequency on the fourth formula.

The validation of the simulator was done like this:

- Generated reservoir with the following ranges:
 - The intervals in question are: Mean Annual Precipitation in [450,2500] mm
 - mean annual number of wet days in [50,120]
 - IDF curves: a in [20,65], n in [0.12,0.65]
 - September-October-November: cumulative rainfall must be in a ± 50 mm range of what is predicted by a simple regression model between measured SON and MAP (the figure 15 of the article shows that the linear regression seems indeed applicable).
- Simulated 3 rain gauges, 500 years, 5 min resolution.
- Checked global statistics against a real data set for the region of Calabria:
 - annual monthly rainfall
 - frequency distribution for:
 - * annual rainfall,
 - * annual number of wet days,
 - * seasonal precipitation (3 of the 4 summary stats)

Is this model usable with our data sets ?

- The general approach would certainly be applicable: the model is simple, we have access to IDF curves.
- However, the implementation is in Visual Basic and doesn't work without Microsoft Excel. Code organization is bad (macros) and comments and variables names are in Italian. So we would have to rewrite everything.
- We believe the use of goniometric functions arose from performance issues due to the use of VBA. Therefore, the approach might not be grounded in data but in implementation.
- The brute force approach is weak regarding the optimal fitting of parameters (it may work in practice but the article doesn't question that)

3.4 SHYPRE

The SHYPRE (Cernesson, Lavabre, and Masson 1996, Lang and Lavabre 2007 and Arnaud 2000) simulator is a full hydrological simulation but inside there's a rainfall simulation that is quite close to what we need. The SHYPRE rainfall simulation takes time series as its input and isolates storms in it with the following method.

The first operation is to filter out irrelevant rainy episodes. A rainy episode is a sequence of days of rains such that:

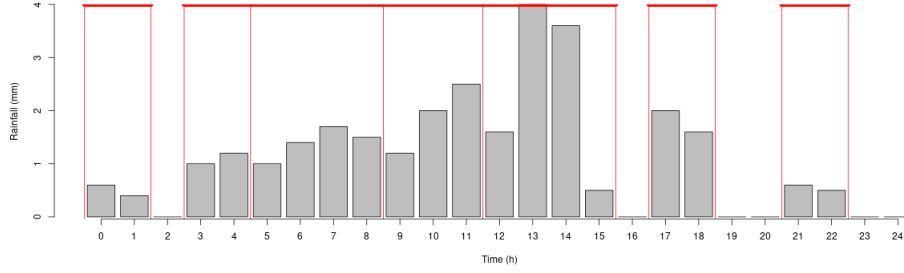


Figure 3.4: SHYPRE storm split example (time is measured relative to the onset of the storm)

- the cumulative precipitation on each day is ≥ 4 mm,
- there is at least one day with cumulative precipitation ≥ 20 mm.

Next, the rainy episodes are split into rainy periods (a day with more than 4 mm of rainfall doesn't imply that each hour has rainfall, so a split is possible), see figure 3.4. A rainy period is a sequence of hours of uninterrupted precipitation. The rainy periods are separated by blocks of no precipitation (see hours 2, 16, 19-21 on the figure).

Each rainy period is cut in elementary hyetographs. An elementary hyetograph contains only one maximum of precipitation (one peak). For example, the storm from hour 3 to 15 is split into 4 elementary hyetographs (see red lines), each of these has exactly one peak. The separation between two elementary hyetographs is the moment where the first derivative changes from a negative sign to a positive sign (that is, a local minimum).

In the SHYPRE method, the elementary hyetograph is used as a basis for estimating several parameters:

- duration of the elementary hyetograph (in hours)
- mean intensity (intensity is total precipitation divided by duration) mm/h
- relative position of the peak inside the storm (from 0 for the onset to 1 the end)
- ratio between peak intensity and mean intensity
- duration of the dry period following the hyetograph and preceding the next one. Note that this can be zero in case two hyetographs are joined or it can be undefined if we're looking at the last hyetograph of the period.

Moreover, one has to estimate:

- The number of rainy episodes in a period of time
- The number of rainy periods in a rainy episode
- The number of elementary hyetographs in each rainy period

(note: The authors don't explain what happens if the number of "things" inside a period of time is too big to fit in that period)

The article then explains that each of these variables can be fitted, one independently of the others, to appropriate classical probability distributions. It is then easy to generate storms.

It is clear that this method could be applied to our data. However, it has two issues:

- The method to separate rainy episodes, rainy periods, etc. is not well motivated. After implementing it, we guess that it was designed to be able to have a collection of parameters which are independent of each other and easy to estimate. Specifically, the elementary hyetographs have a simple triangular shape which is easy to compute/simulate.

- The duration and mean intensity are modeled independently which is intuitively strange.

The second issue has been addressed in Arnaud P. 1999. The idea is to see that the sum of two independent, uniform random variables can be used to measure the dependency between two other random variables. As we were looking for a proper way to model the dependency between a storm duration and its intensity, we investigated this a bit more.

So let X and Y two independent, uniform random variables that have their values in $]0,1[$ and $z=X+Y$ their sum. Looking at the PDF of these random variables f_X, f_Y, f_Z , we can write $f_Z(z) = \int f_X(x)f_Y(z-x)dx$. If $z \leq 1$ then we must have $x \leq z$ else $x+y=z$ won't hold. Recalling that X is uniform over $]0,1[$, we can rewrite $f_Z(z) = \int_0^z f_X(x)f_Y(z-x)dx = \int_0^z 1 \times 1dx = z$. With a similar reasoning, if $z > 1$, then $f_Z(z) = \int_{z-1}^1 f_X(x)f_Y(z-x)dx = \int_{z-1}^1 1 \times 1dx = 2-z$. Therefore,

$$f_Z(z) = \begin{cases} z & \text{if } z \in]0,1] \\ 2-z & \text{if } z \in]1,2] \end{cases}$$

The corresponding CDF can be computed:

$$F_Z(z) = \begin{cases} \frac{z^2}{2} & \text{if } z \in]0,1] \\ 2z + \frac{z^2}{2} - 1 & \text{if } z \in]1,2] \end{cases}$$

Now, if two variables are uniform and independent then, their CDF must be like F_Z . If we look at the values of the CDF of the intensity and duration of storms, these are uniform random variables. If we sum them and compute the CDF and compare that to F_Z then we will see how independent the intensity and duration are. That is what we do on figure 3.5. As one can see, our duration and intensity variables are not independent. In the article, their values also display a dependency but in a more symmetrical way. Therefore, they propose to change the formula of F_Z like this:

$$F_Z(z) = \begin{cases} \frac{z^n}{2} & \text{if } z \in]0,1] \\ 1 - \frac{(2-z)^n}{2} & \text{if } z \in]1,2] \end{cases}$$

Using the additional freedom of n they fit the distribution better (see their figure) and therefore they can move on to propose a procedure to pick duration and intensity randomly:

1. Pick z at random
2. Pick a value for $F_i(i)$ at random (it is easy since it is uniform)
3. Then compute $F_d = z - F_i(i)$

That procedure has an issue, noted by the authors: $z - F_i(i)$ can be outside the domain of F_d in that case, they try another $F_i(i)$. We think this “fix” is not a good idea because it will influence the distribution of F_i . Note also that the fix they propose works because of the symmetry in their data. In our case that symmetry is not present and, although we tried, changing n cannot overcome that.

So, is SHYPRE applicable to our data sets ? For a long time we thought so I implemented most of it. However, we encountered the following issues which ultimately led to discarding SHYPRE (although some of its ideas still permeate around):

- The PDFs proposed in the article to fit the parameters values don't work very well on our data set. We don't have a root cause for that, it just doesn't work very well.
- As we have seen, the SHYPRE way of handling the correlation between intensity and duration is not very satisfactory and we sought another one. As we will see, our choice will lead us to find at the same time the duration and quantity of rainfall of a given storm. Once we have this “box”, we will figure out a hyetograph inside it (in other words, we will sprinkle the set rainfall quantity over a set duration). The SHYPRE model is incompatible with that approach because it needs to have the freedom to select the number and duration of elementary hyetographs in a storm, free from its overall duration.

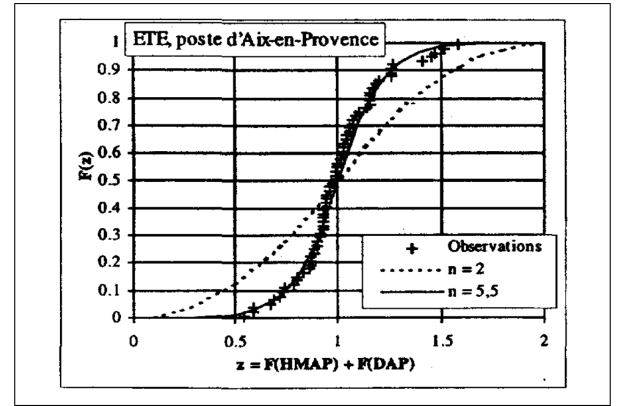
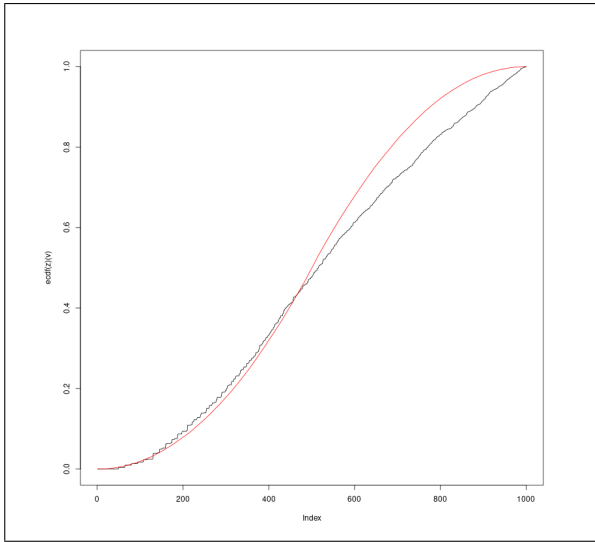


Figure 3.5: Left: Comparing CDFs of the sum of two random uniform variables (red line) and the sum of the CDF of intensity and duration of storms (black; Bastogne data). Right, same comparison as the authors show in their article. HMA is the quantity of rainfall and DAP is the duration of rainy episodes.

3.5 Summary of models/simulators

	STORM	STORAGE	Willems	SHYPRE
Short description	A simulator based on a collection of input PDF's	Stochastically generates plausible parameters sets, filter them if they give good results, then chooses from these for new simulation.	Fit PDF's on data. Then simulate at the rain cell level.	A box based simulator, using elementary hyetograph as the basis for modeling.
Outputs	Gridded precipitations over basin	Rain gauge detail level	Rain gauge detail level	Rain gauge detail level
Time scale	Minutes	Minutes up to 60.	Minutes	Hours
Climate change	yes	no	no	no
Seasonality	2 seasons with 2 sets of parameters/PDFs	Goniometric functions, 4 seasons	no	yes
Code	Python but not easy to reuse (PDF's hardcoded, code duplication for seasons, not modular, mixing simulation/validation, references to non existing data, for example: <code>Intensity_data</code>)	VBA, unusable without Excel, code really hard to read	no code	no code
Limits	STORM doesn't help us to describe our own precipitations. Specifically, the storm area sizes and gradient of precipitations inside a storm are impossible for us to get.	Goniometric functions are not well motivated and may be useless.	Gauge data are measured at one minute resolution, so the gaussian representation of rain cells is possible. Rainstorm speed and direction (and thus, of their rain cells) are assumed constant during the passage of the storm above the gauge area.	Dependency between intensity and duration of storm is not well supported.

	STORM	STORAGE	Willems	SHYPRE
Inputs	<p>PDF of various measurements:</p> <ul style="list-style-type: none"> • storm area size, • yearly precipitations, • storm durations, • intensity-duration curves, • spatial gradient of storm precipitation intensity. • 2D grid of potential storm locations • storm inter arrival time • potential evapotranspiration • orographic groups <p>Moreover, for climate change, additional parameters are used. At each year (Y):</p> <ul style="list-style-type: none"> • $\mu \pm \mu \times Y \times \Phi$: change the mean of the P_{Total} PDF • $P_I \pm \Omega \times Y \times P_I$ 	<ul style="list-style-type: none"> • Time resolution (1-60 min); • Mean Annual Precipitation; • mean annual number of wet days, • cumulative seasonal precipitation (summer, spring, winter, autumn); • ADF power functions $a_T d^{n_T}$ parameters for $T = 2, 5, 10, 50, 100, 200$ years 	Gauges readings, one minute resolution	Gauges readings, one hour resolution

	STORM	STORAGE	Willems	SHYPRE
Fitted Parameters	See input PDF's parameters and climate changes parameters.	<p>The inter-arrival time T_s (exponential, λ); # rain cells in a rain storm: (geometric r.v., mean θ); waiting time between specific rain cell and the start of the storm, (exponential, β); Intensity and duration of a raincell, both exponential (β_I, β_D).</p> <p>These parameters (except W) are then cast to goniometric series, so the "fit" occurs on $\frac{1}{\beta_{I,0}}, \frac{1}{\beta_{D,0}}, \frac{1}{\beta_w}, \theta_0, \theta_{\min}$</p> <p>$\frac{1}{\lambda_0}, \left(\frac{1}{\lambda_0}\right)_{\min}, \phi_{1,\lambda}, \phi_{2,\lambda}, \eta, \chi, \xi$.</p> <p>NB According to the article (page 7) the parameters set choice is linked to Calabria climate.</p>	<p>$\theta, u, r_{\max}, s_x, s_y, l_s$: storm direction, storm speed, rain cell max precipitation, rain cells spatial extent, start position of storm, length of storm, # rain cells per unit area (all are given along a reference frame aligned on θ). We assume all storms start from the same point.</p> <p>Each parameter is associated to a distribution: θ normal (degrees), u Weibull, r_{\max} generalized pareto, s_x, s_y together log-normal, l_s exponential.</p> <p>Finally, storm <i>inter-arrival time</i> is modeled by a double exponential fitted directly on data.</p>	<p>duration of the elementary hyetograph; mean intensity; relative position of the peak inside the storm; ratio between peak intensity and mean intensity; duration of the dry period; number of rainy episodes in a period of time; number of rainy periods in a rainy episode; number of elementary hyetographs in each rainy period.</p>
Calibration	no	Parameters fit on objective function (made of the various measures of success of the model, see below)	<p>Parameters fit on rain gauges data in two phases:</p> <ul style="list-style-type: none"> rain cell detection (+ storm ?) rain cells parameters fitting 	Usual parameters fit on data set

	STORM	STORAGE	Willems	SHYPRE
Validation data	<p>Validation is done at specific points (rain gauges).</p> <ul style="list-style-type: none"> • Number of storms per year; • average storm total; • Total annual precipitation 	<p>Validation over global statistics:</p> <ul style="list-style-type: none"> • annual rainfall • annual number of wet days • mean seasonal precipitation 	<ul style="list-style-type: none"> • Intensity-Density Functions for various aggregation levels (t=1 year, 1 month, 27 years) • extreme value distributions • autocorrelations • scale properties 	<p>Not relevant (Done with hydrological data)</p>

3.6 Conclusion

After having read those articles and while at the same time having analyzed our data, we can say the following:

- All the reviewed simulators use PDF fitting to make simulation
- None of them are usable “as is” (STORM code is a bit tricky and doesn’t help at all in the fitting of PDF; STORAGE code is impenetrable; Willems code models things based on data more accurate than ours)
- The notion of storms is used everywhere: time series are cut into storms.

In light of this, we decide that our simulator will:

- work conceptually at the storm level (not at the hourly rain level, so we won’t model time series as such, for example with Markov chains)
- will model reality with the use of PDF

As an additional comment, we have to say that after having implemented our simulator, we could add another conclusion:

- PDFs choice and fitting is highly dependent on the data set at hand and that may explain the variety of models. This leads us to think that reusing an existing simulator might be difficult.

Chapter 4

IDF Curves

IDF (Intensity-Duration-Frequency) curves relate maximum intensity, duration and frequency of rain events. In general the frequency is given in terms of a return period expressed in years. A return period is the expected number of years before something happens. In our case, the expected number of years before some rain happens. Of course, the event can occur more often than expected and will occur repeatedly if we wait long enough. A point on a curve of a return period P located at duration d means that, each year, we have a probability $1/P$ of meeting or exceeding the corresponding intensity of rain during a period of d unit of time. In our datasets, we work with return periods expressed in years and durations expressed in whole hours.

4.1 Using pre-made IDF curves

Countries usually keep track of their meteorological events and provide IDF curves. Belgium is one of these: the IRM provides IDF curves measured at various sites. Here we give an example of such a curve for Bastogne (data coming directly from the IRM). We give two flavors of it: one where we view the total precipitation (precipitation-density-frequency) and the regular one with the intensity instead of total precipitation.

An important point to note here is that the IRM data upon which their IDF curves were built is richer than the one from the DGO. IRM data spans about 60 years with 10 minutes or 8 hours measurement frequency whereas the DGO data is 20 years hourly. The critical part here is the number of years which directly influences the number of maxima that can be collected.

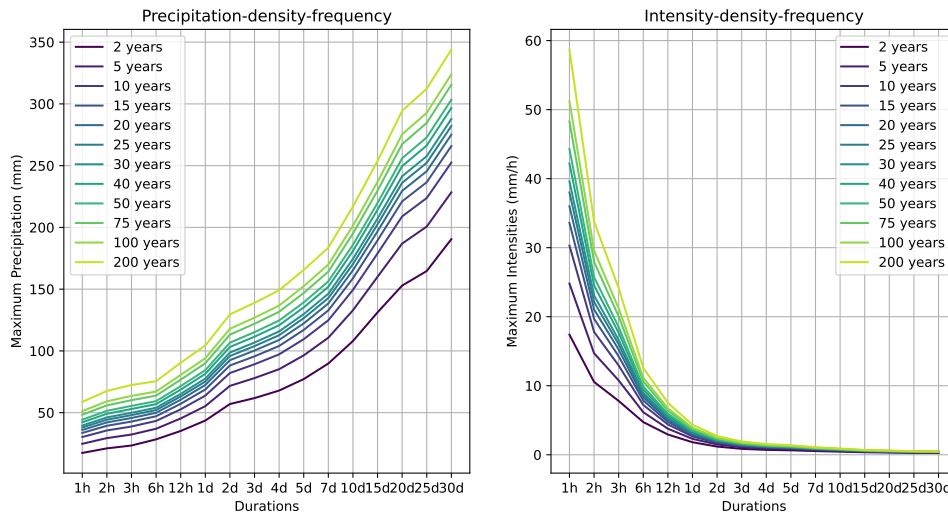


Figure 4.1: Comparing IRM's IDF curves. Left precipitation (mm) view; right intensity view (mm/h)

4.2 Building the IDF curves

To build the IDF curves with a data set spanning Y years for one measurement station we follow the procedure outlined in M. 2017:

1. For each possible duration d .
2. Compute all total precipitation over all possible periods of duration d . This is done with a convolution operation over the full hyetograph. This forms a time series of hourly steps. The value at step i of the time series represents the total precipitation that fell during a period of d hours, starting at the i -th hour.
3. We then extract extreme values from the time series. We follow the Block Maxima approach (of the standard GEV framework) with a block size of half a year.
4. Then we fit (we did it with Maximum Likelihood Estimation and l-Moments without much difference) a generalized extreme values (GEV) distribution over those extremes. So the GEV model gives a probability distribution of (extreme) rainfall over duration d . The GEV has these formulas. For the cumulative distribution function:

$$\mathcal{G}(x) = \begin{cases} e^{-e^{-s}} & \text{for } \xi = 0 \\ e^{-(1+\xi s)^{-1/\xi}} & \text{for } \xi \neq 0, \xi s > -1 \\ 0 & \text{for } \xi > 0, \xi s \leq -1 \\ 1 & \text{for } \xi < 0, \xi s \leq -1 \end{cases}$$

for the probability density function:

$$g(x) = \begin{cases} e^{-s} e^{-e^{-s}} & \text{for } \xi = 0 \\ (1 + \xi s)^{-(1+1/\xi)} e^{-(1+\xi s)^{-1/\xi}} & \text{for } \xi \neq 0, \xi s > -1 \\ 0 & \text{otherwise} \end{cases}$$

both where $s = \frac{x-\mu}{\sigma}$. The parameters have names: μ is the location, σ is the scale and ξ is the shape.

5. Then we compute the precipitations at various return periods (2,5,10,etc. years) by using our GEV formula. This computation is a bit more involved and works like this. Given the GEV cumulative distribution function $\mathcal{G}_\theta(q)$ (q is the total precipitation for which we evaluate the c.d.f., θ are the parameters ξ, μ, σ of the distribution), we have by definition $\mathcal{G}(q) = P(x \leq q)$. Now, a return period T means an event that has a probability of happening $1/T$ each year. So if our event is “it will rain *more than* q mm (in d hours) once every T years”, then the probability that $P(x > q)$ is $1/T$. Conversely, $P(x \leq q) = 1 - 1/T$. Therefore, we can write: $\mathcal{G}(q) = 1 - 1/T$. As we’re interested in finding the total precipitation *given* the return period, we want to invert that equation: $q = \mathcal{G}^{-1}(1 - 1/T)$. That inverse can be computed analytically:

$$\mathcal{G}^{-1}(q) = \begin{cases} \mu - \sigma \log(-\log q) & \text{for } \xi = 0 \\ \mu + \frac{\sigma}{\xi} \left((-\log q)^{-\xi} - 1 \right) & \text{for } \xi > 0, q \in [0, 1[\text{ or } \xi < 0, q \in]0, 1] \end{cases}$$

We then put various values of T in that last equation to compute the associated total precipitation for the duration d .

6. Finally, since IDF curves usually express rain intensities instead of total precipitations, we divide all the computed total precipitation from the previous step by the associated duration d .

This procedure is somewhat involved but has the following advantages:

1. It allows us to compute intensities for any return period. This was not obvious to us in the beginning because we found other, more empirical, ways to compute the IDF, relying on simpler concepts. These approaches were limited to integer return periods. There were also limited to predicting further than the year span of the data.
2. It allows us to compute intensities for return periods greater than the number of years available to the data set, thanks to the GEV distribution fit. Of course, one cannot extend to arbitrarily many years.

Now, in practice, we encountered some difficulties.

- The first one was that the GEV fit relies on the selection of extremes. There are two strategies available: block maxima and peaks over threshold. The threshold method relies on the selection of a threshold above which a value is considered extreme. If it is too high, then one does not collect enough maxima to make a proper fit, if too low, then we collect too many points including some that may not be maxima at all. So there's a balance to find. M. 2017 writes that the threshold is selected so as to find “2 valeurs extrêmes par année en moyenne”. We thought that description was a bit fuzzy: what happens if we select a cluster of maxima close to each other, potentially dependent ? what happens if a year has 4 maxima and the next has zero ? Since the answer to these questions seemed to us a bit subtle, we preferred the block maxima approach. In that approach one selects a period of time over which one looks for a maximum. So for example, we selected a period of 6 months, cut our timeline accordingly and selected one maximum in each 6 months block. This method has the advantage that the timeline is examined uniformly (one maxima for each block). It has the disadvantage that we can still have two maxima close to each other (one at the end of a block close to another one at the beginning of the next block).
- The second difficulty relates to the fitting. We initially selected a block length of one year but noticed the GEV fit was often very bad, it was predicting values which were completely unrealistic. After investigation, we came to the conclusion that the number of extrema selected (about 20) was not enough to fit the data properly. Therefore we changed the block size to half a year, doubling the number of points for the fit. This mostly solved the problem. The consequence was that the return periods were now expressed in 6 months blocks rather than one year blocks. Therefore we had to adapt our computations to convert back to yearly units, as it is customary in IDF curves.
- Finally, computing a block size of “half a year” is not entirely obvious because of leap years. Computing that carelessly leads to missing the last block in our computations of block maxima (quite like an off by one error). In the end, we kept the approximation of 182.625 days for a half year and dropped the last block if incomplete.

On figure 4.2 we present the result of our procedure applied to the Bastogne station, compared to the IRM data, using precipitations instead of intensities. The general shape is respected, so we may think our procedure is good.

As we have said, the IDF curve represents intensity-duration-frequency. But it is also possible to make a curve with total precipitation instead of intensity. Since intensity is total precipitation divided by duration, one can just plot precipitation as intensity times duration. Recall also that we computed the extremes fit over the precipitation instead of the intensities. This leads to figure 4.3.

Although the general shape of the curve (right of the figure) is the same, one cannot be blind to the fact that there are two valleys in the curve we built with our procedure. We tracked the origin of these valleys down to the GEV fitting. On the right part of the chart close to the X axis, we reported the sign of the shape (ξ) parameter of each GEV fit. As one can see, the negative sign is associated with the valleys.

Now a bit more explanation about the GEV distribution is in order. The shape parameter allows the GEV to represent three kinds of distributions which have different behavior for their tails:

- The Gumbel distribution when $\xi = 0$ which has a light tail

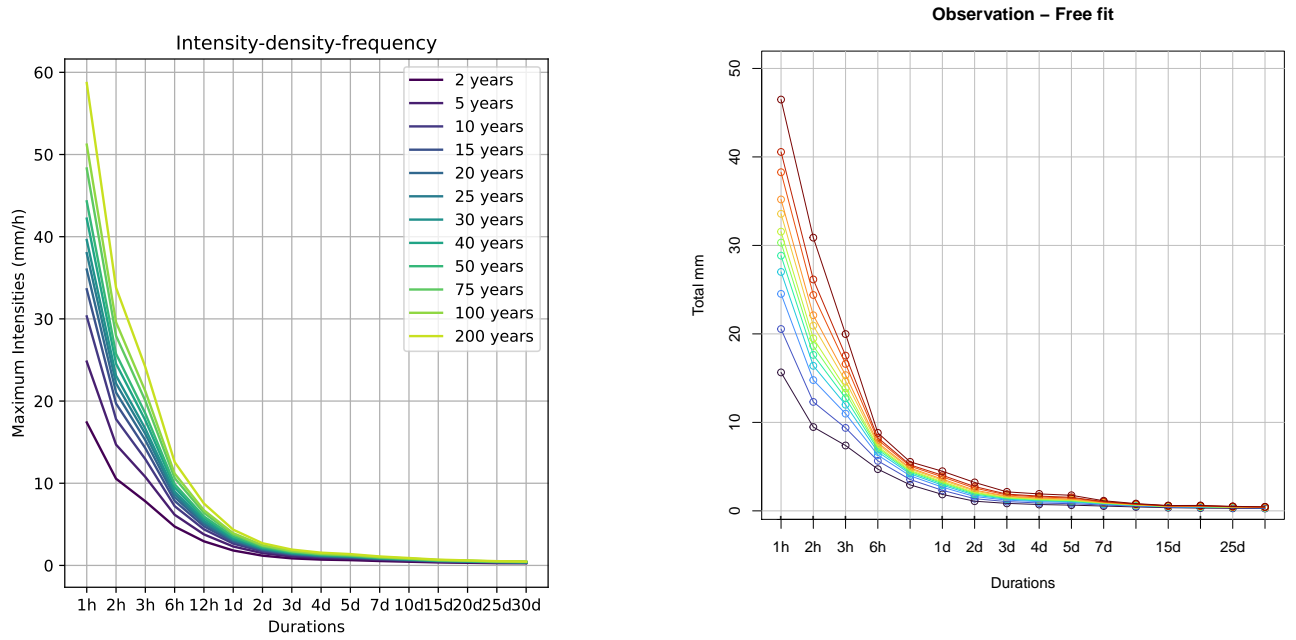


Figure 4.2: Comparing IRM (left) and DGO's data (right)

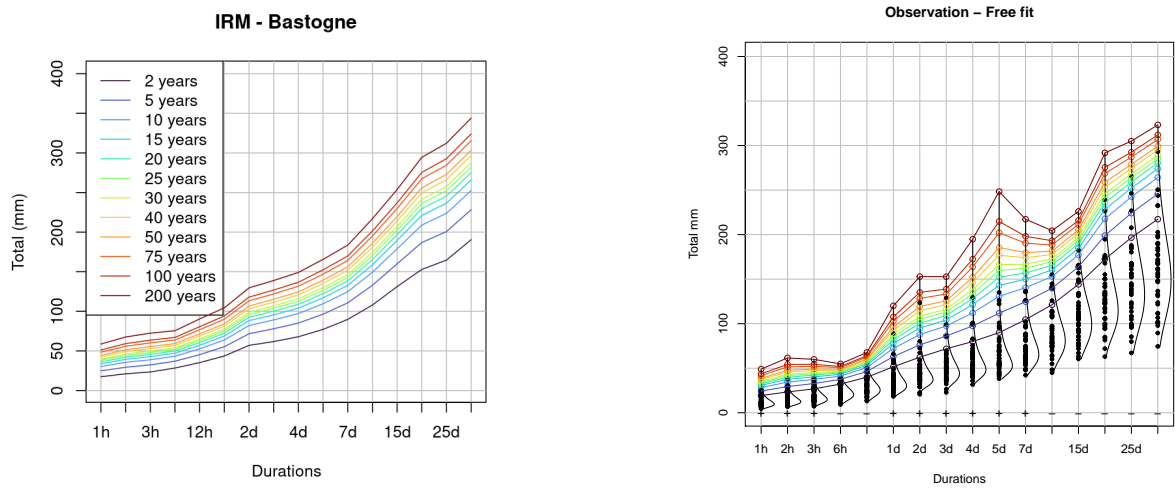


Figure 4.3: Comparing total precipitations “IDF” curve (IRM left, DGO right). The right chart displays the probability density function of the GEV fit to the maxima (black points). Note the bumps/valleys on the right. The black dots are the data points on which the GEV were fit.

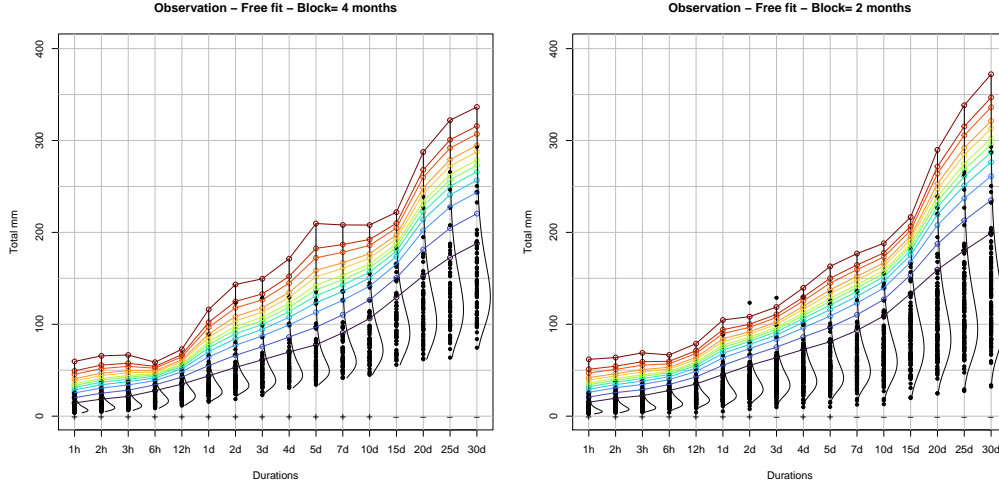


Figure 4.4: Comparing different block sizes for block maxima extraction. The black dots are the data points on which the GEV were fit.

- The Fréchet distribution when $\xi > 0$ which has a heavy tail
- The Weibull distribution when $\xi < 0$ which has a bounded upper tail

The two last distributions are what we encounter in our fits. We have used generally available frameworks to complete the fits (pyextremes and extRemes, both of them display similar issues but with different extent).

So the odd thing is that the fit procedures we used seem to favor the Weibull distribution sometimes although it is bounded on its upper tail. We find that odd because it tells us that the real distribution is bounded. Since we're talking about extremes, we find this surprising as we would expect that one can always find a more extreme point. We understand that there must be a limit to the extremes but we don't see why that limit can be read in the observations. Moreover, it is simply surprising that there are shape swings in the fits over similar data. We tried to improve the situation in two ways:

- We tried to force the ξ parameter to be the same sign everywhere. This smoothed the curve but this made the MLE result questionable so we left that out.
- We once tried to increase the number of selected maxima by decreasing the block size. As can be seen on figure 4.4 a smaller block size leads to a smoother curve, but the changes in the shape parameter were still there. Moreover, we want to compare our IDF curves to those of the IRM which used a six-month block size. So we left that approach out.

As we will use IDF curves during the validation of the simulation, we wonder if the IDF curves coming from the simulation will be like those given to us. To analyze that we made a quick test like this:

1. Randomly choose some maxima (36 out of 40) in our data, using the block maxima approach as before and given a rainfall duration (in our case it will be either 5h, 10h, 20h, 40h).
2. Fit a GEV on these, let's call it \mathcal{D} .
3. Use \mathcal{D} to predict maximum return levels for given period duration
4. Repeat the previous steps 20 times

This process is like building 20 different IDF points for a given duration, each of those being based on slightly different maxima (akin to bootstrapping). The results are plotted on figure 4.5. As one can see the predicted return level for a close future are rather stable across our 20 simulations. However,

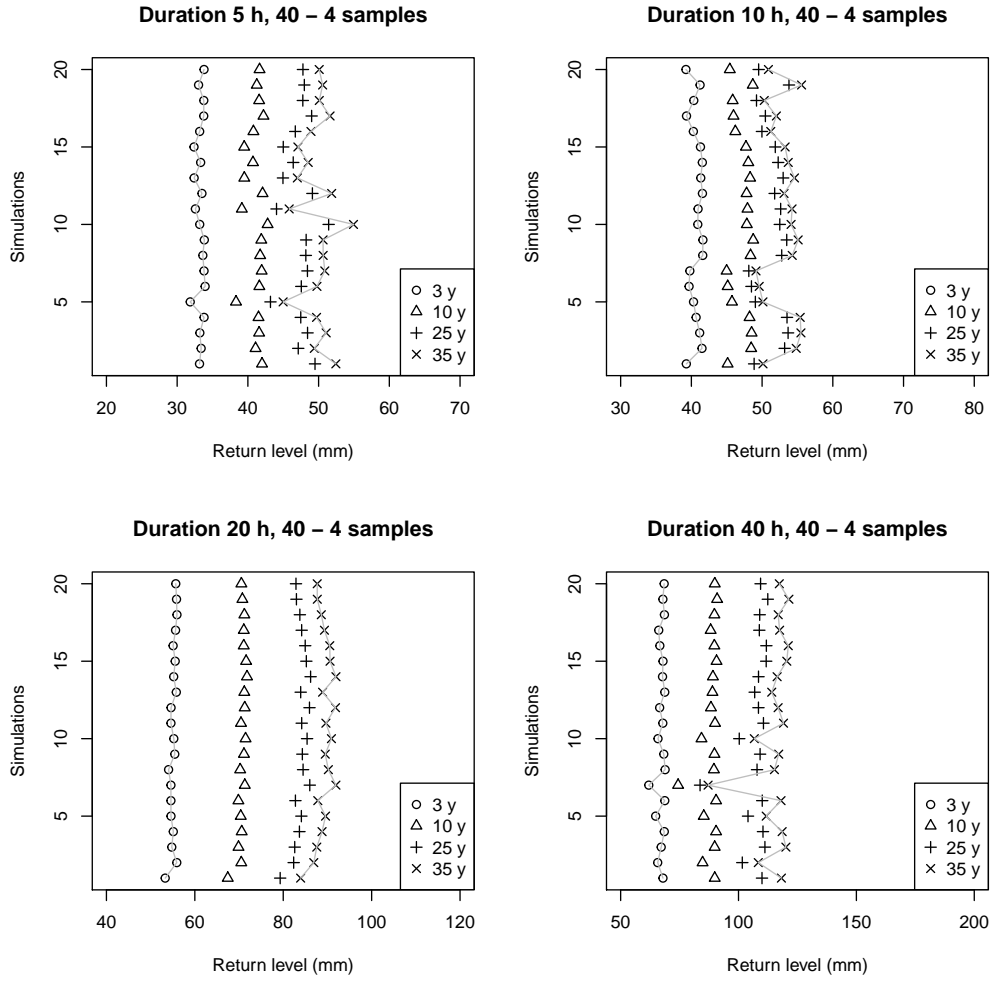


Figure 4.5: GEV variability influence over IDF

the predictions farther in future are less stable. We conduct the experiment with maxima obtained for various rainfall durations in order to cover a broader spectrum of values.

Our conclusion is that IDF curves can be compared to each other but that it is safer to do it over smaller return periods.

Chapter 5

Modeling

In this chapter, we will look at how we build our model for the simulator. We'll also report some of the attempts we made and ultimately rejected. We do that because our experience is that a lot of research was necessary to figure out a working solution.

5.1 Working at the level of individual stations

In the beginning of the project we aimed at working at a global level, taking into account correlations between stations. The main correlation we observed was that when it rains in a station, it is not uncommon that it rains in a close station (see figure 5.1). However, before going to the global level, we thought it'd be safer to first understand what happens at an individual station. As one will see, that task itself was hard enough in itself so we didn't investigate correlation any further and focused on modeling one station at a time, independently of the other.

5.2 Storm separation

As we have seen in the previous chapter, we aim to work at the conceptual level of a “storm”. In this section, we will show how we transform our precipitation time series into storms series. We will see two approaches:

- A simple one where we split storms based on the hours where it doesn't rain.
- The SHYPRE method where periods of rain are first isolated.

5.2.1 Simple storm separation

The simple storms separation is done in three steps: noise removal, storm separation and storm selection.

1. The noise removal consists in removing all hourly precipitation inferior to an arbitrary noise level.
2. The storm separation itself consists in splitting the time series into sequences of strictly positive hourly precipitations (i.e. sequence of uninterrupted rain).
3. The storm selection consists in keeping only those sequences of which the total precipitation amount is greater than m mm.

After consulting Pierre Archambeau, it appeared that removing a bit of noise was making sense from a hydrological perspective.

The question is then how to set both the noise level and the m threshold. For the hourly precipitation, the figure 5.2 indicates that, for example, setting the noise level to 0.2mm discards about 4.4% of the precipitations while 0.3 mm leads to about 8% of loss, rather a lot. Moreover, a look at figure 5.3 shows that the number of storms is not quite impacted by the level of noise. So removing a bit

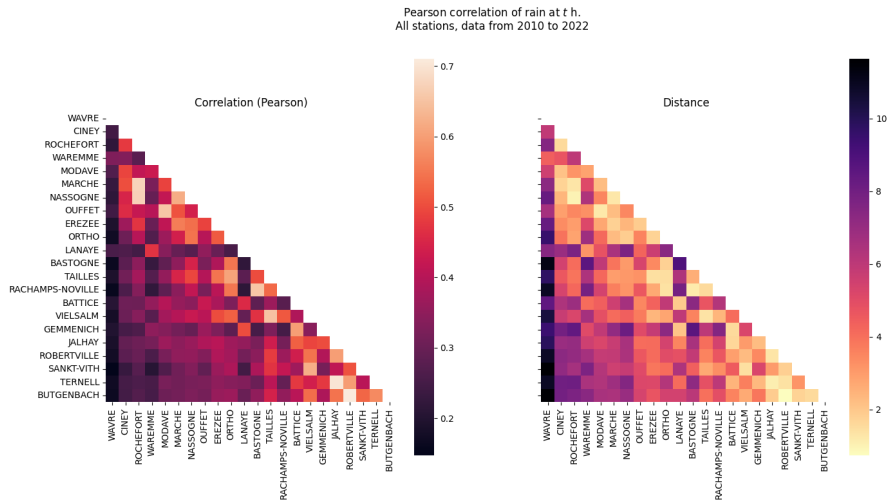


Figure 5.1: Correlation between rain levels at two different stations, at the same time. Close stations are more correlated than stations far away from each other.

of noise is safe in any case. We therefore choose a threshold of 0.2 mm (as the minimum acceptable hourly precipitation).

For the storms selection, we see on figure 5.3 that changing the threshold on the storm total precipitation directly affects the number of found storms. Therefore, we choose a small threshold of 1 mm (as the minimum acceptable total precipitation in any storm). In the end, the noise removal and storm filtering remove about 11.5% of the precipitations.

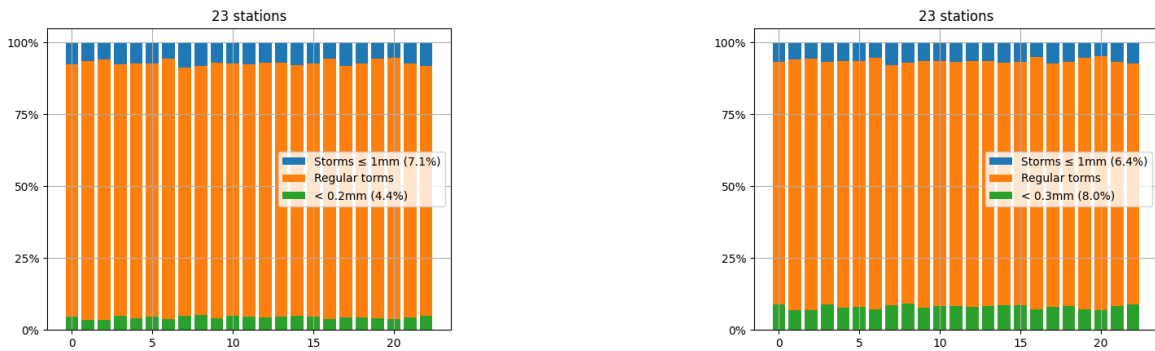


Figure 5.2: Quantity of precipitation in noise, small storms and regular storms; over 22 stations. Left: hourly threshold 0.2mm; right 0.3mm.

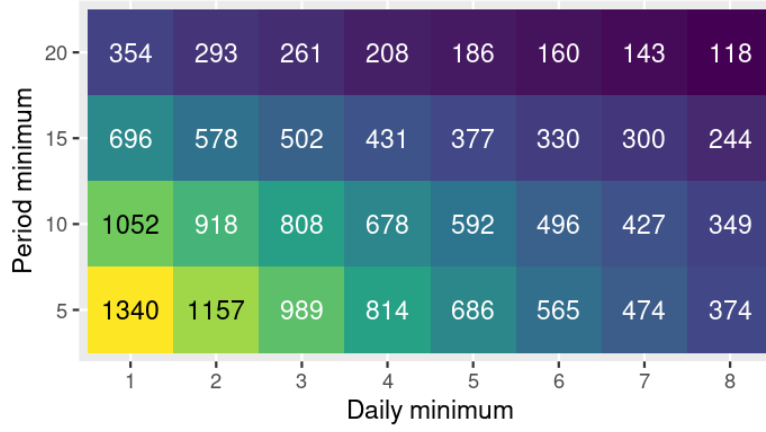


Figure 5.4: Number of storms after SHYPRE separation

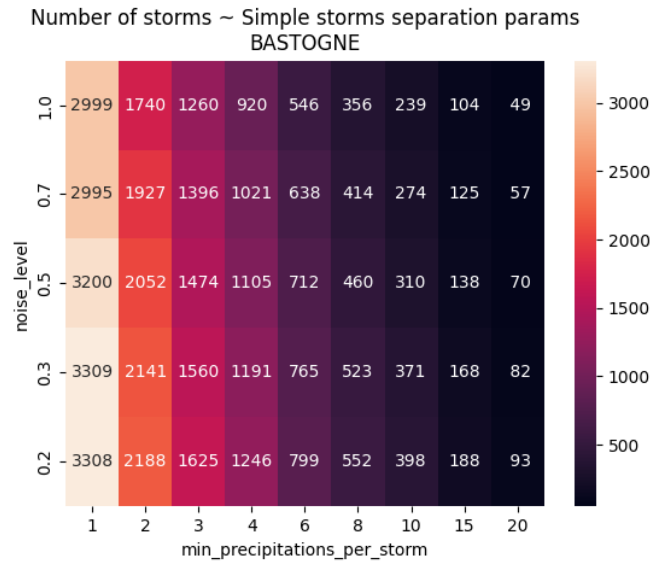


Figure 5.3: Number of storms after simple separation (we only give results for Bastogne, other stations have similar behavior)

5.2.2 SHYPRE

We tried to implement the first filter of SHYPRE on our time series before separating the storms (see above). We show on figure 5.4 how that filter affects the number of storms. Remember that the filter keeps only the rainy periods where there's at least a daily minimum of rainfall and at least one day superior to a threshold). Obviously, the number of storms is reduced and quite a lot. It is our feeling that reducing the number of storms of interest so much is not good because as seen above, we have a lot of storms (and as we will see a lot of small ones). So we will discard that approach.

5.3 Seasonality

As we have seen when analyzing the data set, there is a strong seasonality trend. We will reproduce it by splitting the year in two "seasons". It is customary to split the hydrological year in two parts: from October to march and from April to September. These parts correspond to seasons where one can expect to have different rain profiles.

Of course, splitting the year in two leads to having smaller data sets for the two parts of the year and therefore less data to fit on. We could have split over 12 months but then our data for each month would have become small, making fitting harder. So we settled on half years.

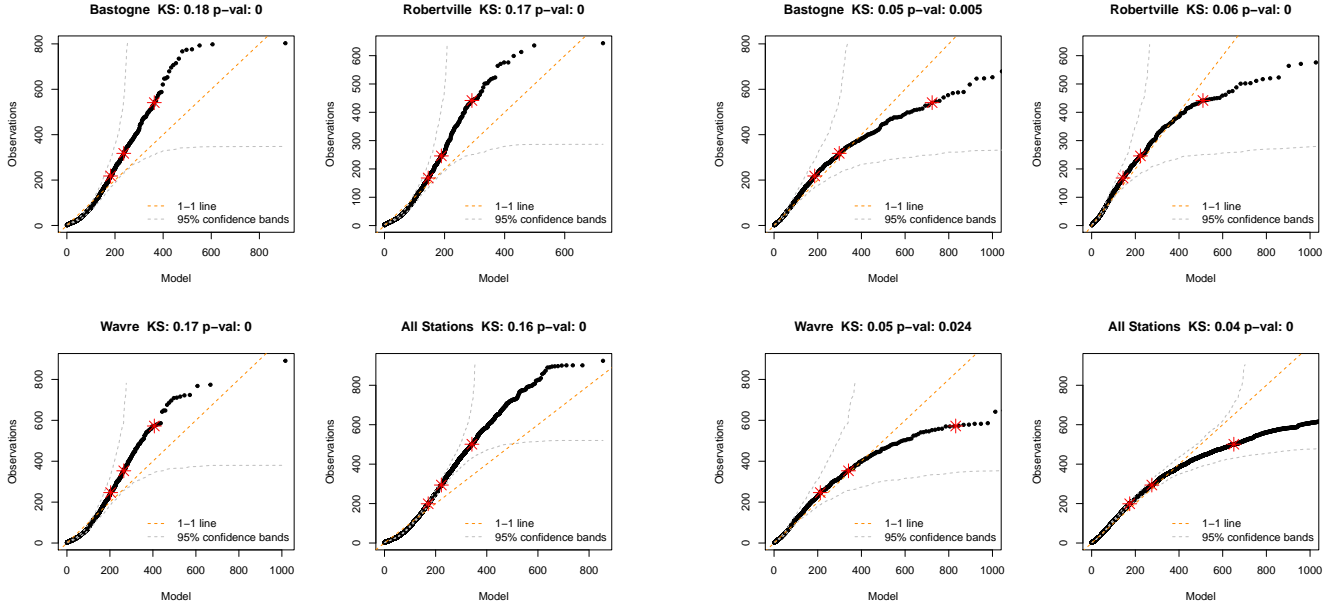


Figure 5.5: Exponential (left) and log-normal (right) fit of inter-arrival time. The red stars represent 0.9, 0.95, 0.99 quantiles.

5.4 Storms per season

To know how many storms occur in each season, we have two possible ways. Either we compute a distribution over the number of storms per season, either we compute a distribution over the storm inter arrival time.

5.4.1 Via inter arrival time

We start with the inter arrival time. The inter-arrival time is the time between the onset of two storms. Putting all storms on a timeline allows us to compute the inter-arrival time between one storm and the next. An histogram of the inter-arrival time leads us to think about some exponential distribution, the hallmark of a Poisson process. Unfortunately, fitting the exponential distribution leads to poor results. We also tried to fit with a log normal (and others) which is a little better according to the Kolomogorov-Smirnov test statistics and p-value (although p-value remains too low to accept the distribution fit hypothesis).

5.4.2 Via number of storms

We now look at modeling the number of storms per season, per year. This has the consequence that with our given data set we will have a limited set of data points.

We tested the Poisson distribution as it is the most common process for things that happen at regular time. The fit was visually acceptable on the individual station but when applied to the whole data set, it was very bad. Although we intend to use the fit for simulation at an individual level, that information casts some doubt over the quality of the fit.

We also tested the normal distribution. We see (figure 5.6) that the fits are better for individual stations as well as for the whole data set. Therefore we keep the normal distribution. Moreover, since we want to validate the data are normally distributed, we can make use of the Shapiro Wilks test. As one can see, its statistic and p values support the normality hypothesis.

5.4.3 Conclusion

Comparing the result of both approaches, we are more confident in the results about modeling the number of storms per season rather than modeling the inter arrival time. To assess one last time the

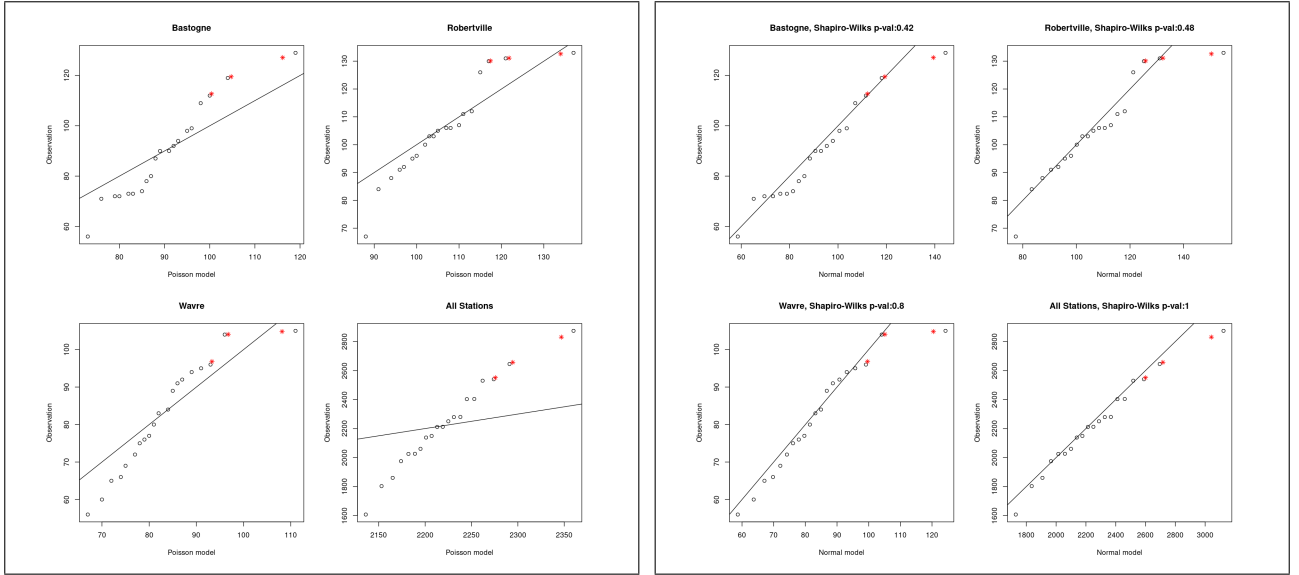


Figure 5.6: Storm per season with Poisson fit (left) and normal fit (right). The red stars represent 0.9,0.95,0.99 quantiles.

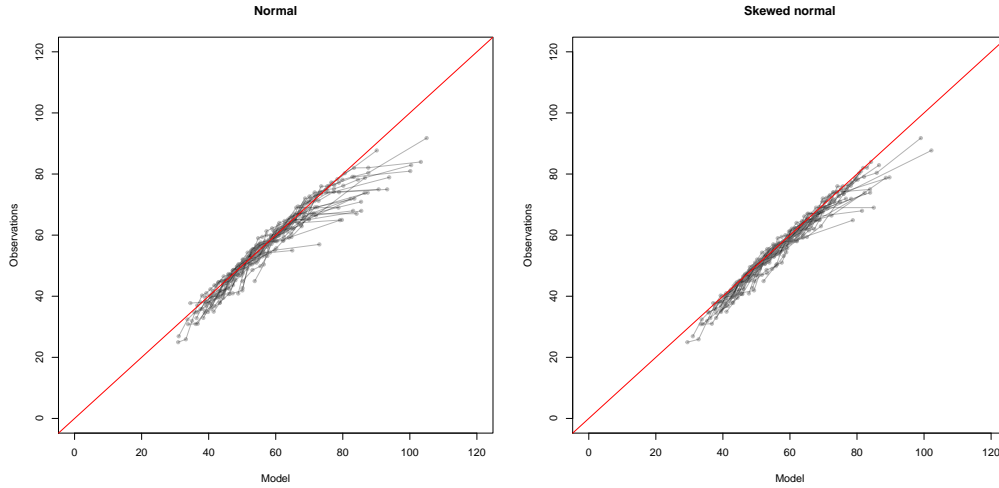


Figure 5.7: QQPlot of normal fits over each stations

quality of our fit, we run it over all individual stations (see the QQ-plot figure 5.7) on the two seasons. As we can see the data points are rather grouped so we're confident our fit is good.

The overall shape, which is bent on its right, indicates that the observation's distribution is less extreme than the fitted one. So we tried a skewed normal distribution (since the data were mostly normal, we thought it would be an easy way to, maybe, improve things). On the QQPlot, the improvement is visible on the summer season; it is not clear on the winter season. To be sure, we also checked that the log-likelihood of the fitted distribution with respect to the observations were better (figure 5.8). The improvement is real in summer and mostly inexistent for the winter. Therefore, we keep the skewed Gaussian.

5.5 Storm modeling

As we have seen, we decided to separate time series into storms. The question left is: how do we model a storm itself? That is, how do we summarize the hyetograph of a separated storm in some parameters? As we have seen in the literature, one can model storms in various way. From the onset of this work, we chose to work with the simplest model possible: a box.

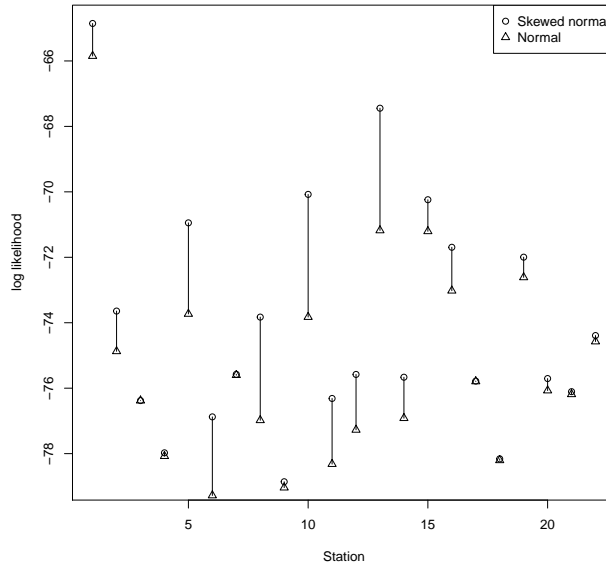


Figure 5.8: Improvement of log likelihood

Basically, a storm is represented as a period of time during which rainfall occurs. For example, a storm lasts 5 hours during which it rains 15mm of water in total. The simplification resides in the fact that we don't represent the "shape" of the storm. The simplest hyetograph possible for such a description is a box: it rains the same amount every hour of the storm. We are then left with two dimensions to model: the storm intensity (or total precipitation) and the storm duration.

Once we will reach a working basis for that simpler model, we will move on to a more complex one where we actually represent a detailed shape for each storm. We stress the fact that we will keep a hierarchical approach: first we simulate storms as boxes, second we improve the simulation of each of these boxes, independently.

Although the box model is quite obvious, we also chose it because it seemed the simplest one. As we had no idea of what it takes to model storms, we thought that growing the model from a simple one to a more complex one was the safer path. Of course, there were other choices, for example a Markov chain. But since we didn't encounter it often, we thought it could be a riskier path to follow.

5.5.1 Notes about distribution fitting

To model the storm intensities and durations, we have tried to fit the data with various univariate probability distributions. This exercise was rather hard to do for several reasons:

- The data are discrete. For the durations, the data are expressed in hours as integers. For precipitations, measured in mm, the rounding is to the first decimal. In both cases, this leads to discrete data (although the physical phenomenon behind is obviously continuous). Although this might seem rather innocuous, this leads to lots of small technicalities which are painful to work with. For example:
 - The detail in the data (especially durations) is very low. This leads to issues when fitting the data because information is concentrated, repeated on several points.
 - Moreover many of the visualization tools at hand expect continuous data. Using discrete data in histograms for example tends to lead to some non uniform grouping in bins when showing histograms, leading to bins with abnormal size. QQ plots end up with many steps which are sometimes misleading.
- The data are left truncated: The durations start at 1h and the precipitations below 2 mm are filtered out. This disturbs the standard tool for fitting. We had two situations:

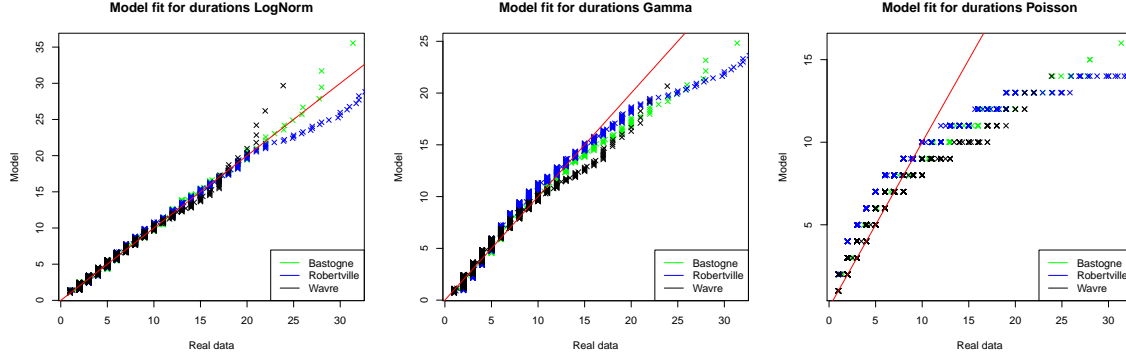


Figure 5.9: Comparing probability distribution fits over all the stations.

- For PDF's which are naturally defined only on the positive domain we often translate the data to the origin. However, this is an issue for some distributions that are not defined at 0. For example, during the fit of a gamma distribution we set its location to the minimum value (2 mm) but obviously, the PDF won't evaluate there. So we ended up moving the location a bit at 1.95.
- We used a truncated distribution which can be built like this: let $d(x)$ a PDF defined over an interval I_d , and $D(x)$ the corresponding cumulative density function. A truncated version of d is $d_t = d(x)/D(b)-D(a)$ where $[a, b] \subseteq I_d$ such that $\int_a^b d_t(x)dx = \int_a^b \frac{d(x)}{D(b)-D(a)}dx = \frac{D(b)-D(a)}{D(b)-D(a)} = 1$ so that it is well a PDF. However, this is not without issue, especially when the mean of the distribution is far from the truncation interval (in that case, the fit may fail because $D(b) - D(a)$ can get close to zero).
- The data has extremes (fat tail): none of the distributions we tried capture that on its own. The extremes pollute the tail of the distribution and they impact the fitting.

Usually, we tried several distributions based on the general shapes of the PDF and histograms of our data. We worked this way because, as far as we know, there's no established way to describe the physical phenomenon of total precipitation or duration of a storm.

We mostly used Maximum of Likelihood Estimation with the L-BFGS-B to fit our distributions. The choice of the algorithm was never an issue to us.

5.5.2 Storms durations

5.5.2.1 Regular distribution fit

We compared several classical distributions: gamma, log-normal and Poisson over the whole data set and all stations. The QQPlot indicates that log-normal is the best fit although not very good in the extremes (figure 5.9).

5.5.2.2 Gamma mixtures

Looking at the log-normal fits above was a bit disappointing in the more extreme values. So we tried to fit a mixture of two gamma distributions. The idea to use that combination comes from the observation that the gamma distribution offers a number of parameters and shapes. So combining two may be flexible enough to cover our needs.

- The PDF of the mixture is $wg(x; \alpha, \beta, \mu) + (1-w)g(x; \alpha, \beta, \mu)$ where $g(x; \alpha, \beta, \mu) = \frac{(x-\mu)^{\alpha-1}e^{-\beta(x-\mu)}\beta^\alpha}{\Gamma(\alpha)}$ with α the shape parameter, β the rate and μ was introduced to help the fit around the value 1 (remember that our durations start at 1, not at zero). Note that μ is inferior to one (minimum duration) to avoid evaluating the PDF or CDF at $x - 1 = 0$ where they are not defined. The PDF is used for the MLE estimation of the parameters with the method L-BFGS-B to allow bounding the parameter space. Usually the optimizer pushes μ as close to one as permitted.

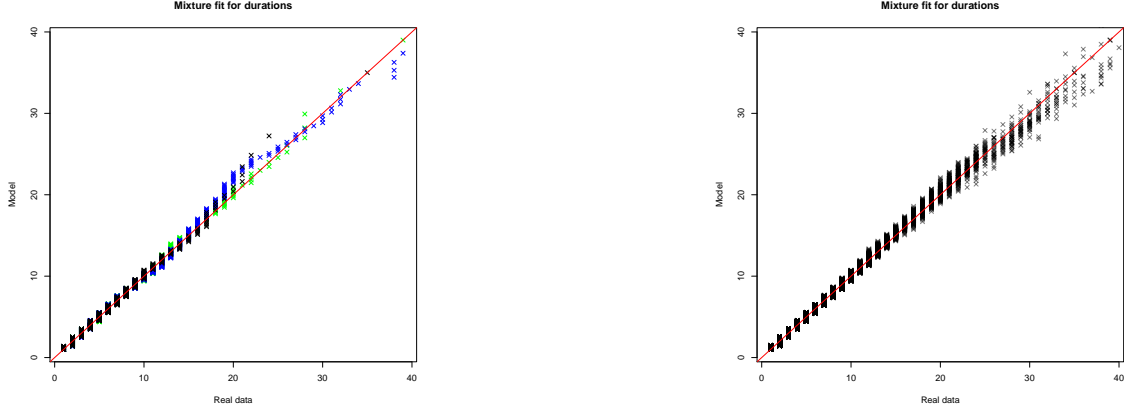


Figure 5.10: Gamma mixture fit for the three test stations (left) and all stations (right)

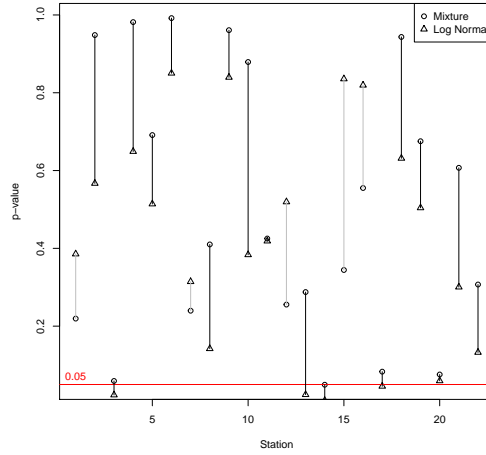


Figure 5.11: Difference between p-values of gamma mixture and log-normal fits over all stations

- The CDF is $wG(x; \alpha, \beta, \mu) + (1 - w)G(x; \alpha, \beta, \mu)$ where $G(x; \alpha, \beta, \mu) = \frac{\gamma(\alpha, \beta(x - \mu))}{\Gamma(\alpha)}$ with γ being the incomplete gamma function. The CDF is used to compute χ^2 goodness of fit statistics.

As we can see on figure 5.10, this fit is better than the previous log-normal one (figure 5.9). If we look at all stations (same figure), we see that the fit is rather good as well. Finally, we compared the p-value of the χ^2 goodness of fit test applied to log normal fit and mixture fit. The figure shows the ratio of their respective p-values for all our stations. We see that the difference is most of the time in favor of the mixture and is significant.

As for the χ^2 goodness of fit test, we would like to mention a few points on how we computed it. Indeed, our procedure for χ^2 goodness of fit is a bit different than textbooks examples because we work on discrete data.

For a χ^2 test, we split our data in bins. Each bin corresponds to one possible value of the observations. Since we have discrete durations the possible values are 1, 2, ... up to the greatest duration. We then count how many observations we have in each bin.

For the χ^2 test to work, the bins must have at least some elements in them, empty bins are not acceptable. Given how our data are distributed, empty bins tend to appear in the tail of the data (e.g. there's no storm of duration 31 hours). Our solution is to group the last bins into one bigger bin. Of course, doing so, we end up with bins of unequal sizes (all bins are of equal width except the last one). We think this is tolerable as the extremes are outliers and this way, we don't drop them.

Once the bins are determined, we can compute the expected probability of each bin. For that we use the CDF of the fitted distribution. For the regular bins (all but the last one), each bin represents a value v . Since v is the result of rounding the values represented by the bin are $[v - 0.5, v + 0.5]$.

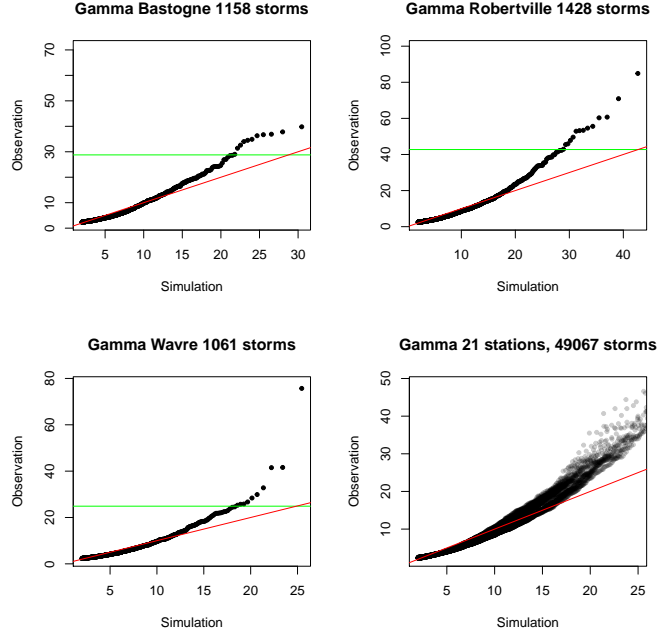


Figure 5.12: Gamma fit of the distribution of precipitations of all storms. The zone inside green lines represents the 99% percentile of the data.

Therefore the probability of the bin is given by $CDF(v+0.5) - CDF(v-0.5)$. The last bin probability will be evaluated as $1 - CDF(v+0.5)$.

Because of the rounding and the fact that our durations are ≥ 1 , the first bin $[0, 0.5]$ is not taken into account. However, doing so, we also drop its associated probabilities so the total probability of all our bins is less than one. To alleviate that we re-normalize the remaining of our probabilities to one, distributing the missing probability over all of them. We can do it because usually there's not much probability in that first bin (about 3% in practice).

Once we have computed the bins and their probabilities, we apply the χ^2 test and read the p-value which must be above 0.05. We choose 0.05 following the common practice.

We also note that our procedure works as long as all the durations have corresponding observations. If it wasn't the case, some bins would be empty and that would make the computation of χ^2 impossible. We could merge some empty bins but then we would have a hard time explaining what the χ^2 test actually represents.

5.5.2.3 Hard mixture model

Thinking again about extreme values, we thought we could build another mixture where one distribution would be fit over the regular precipitations and another one over the extreme precipitations, provided we could split the data set at an appropriate threshold. This approach didn't work well because we ended up with a clearly bimodal PDF, which, when used for simulation, was leading to two clusters of storms: one corresponding to the regular storms and one corresponding to the extremes. The two clusters didn't seem realistic.

5.5.3 Storms precipitations

5.5.3.1 Regular distribution fit

Our best attempt with a single distribution fit was the gamma distribution as shown on figure 5.12. As we can see it doesn't even fit up to the 99% percentile. We needed to look for something better.

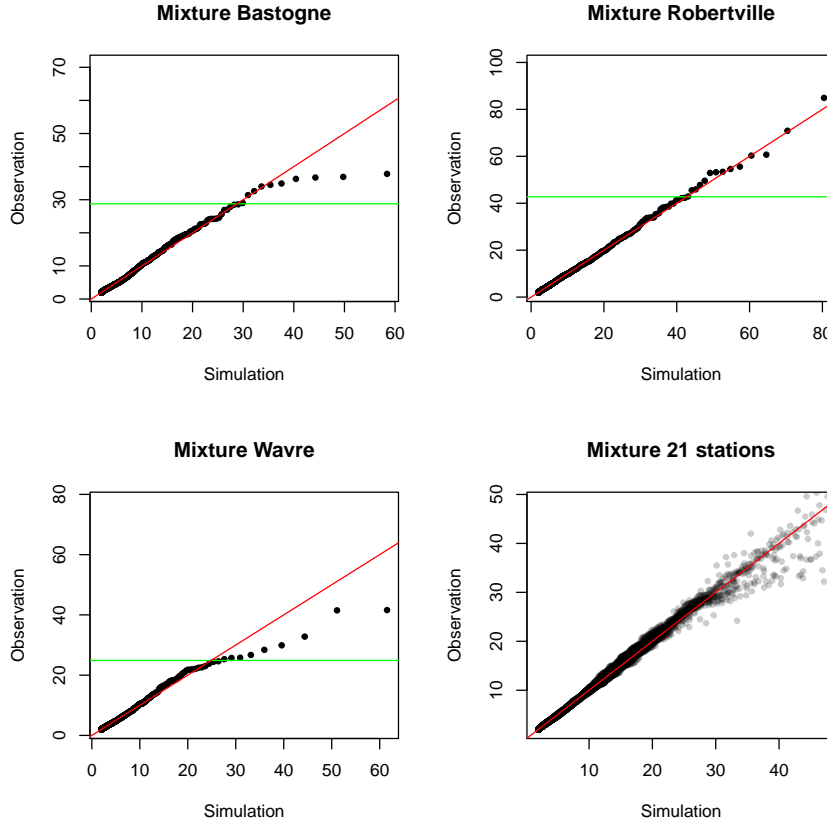


Figure 5.13: QQ plot of mixture distribution for total precipitation versus real data on stations. Green line is the 99% percentile.

5.5.3.2 Sum of Exponential Mixture Model

With the very bad fit we had found above, we looked for something better. We tested a mixture model provided in Furrer and Katz 2008. It is a mixture composed of two distributions. Its CDF is $F(x; \omega, \sigma_1, \sigma_2, c) = 1 - (1 - \omega) \exp(-\frac{x}{\sigma_1}) - \omega \exp(-(\frac{x}{\sigma_2})^c)$ with $x > 0, 0 < \omega < 1, \sigma_1, \sigma_2, c > 0$. We computed its PDF as $\frac{\partial F}{\partial x}$ and then used that to run a MLE optimization to find the parameters.

The resulting figures show a rather big variance between stations when edging towards extremes although if taken as a group, they behave within limited boundaries (see 5.13). The 99% percentile is often met with success. Given that we have about a thousand storms in 20 years, that's about 0.5 “wrong” storm missed every year. We didn't find a better fit.

5.5.4 Correlation between duration and intensity

As we have seen above, we model a storm with two basic dimensions: duration and intensity. Now, it is everybody's experience that intense rain usually doesn't last and that light rain can go for several hours. So intuitively, there is correlation.

5.5.4.1 An uncorrelated simulation

Based on the fitted distribution for precipitations and durations, we build a very basic simulator. It will pick at random precipitations and durations in the distribution fitted above, without any correlation. This might not be as simplistic as it seems as we have seen it done sometimes in the literature. On the figure 5.14 we can see, the overall shape of the distribution in the simulation is quite different than the original data. Although this is entirely expected, this allows us to appreciate the effects of ignoring the correlation. The most obvious one being that we would generate much more long and light rain than there is in reality.

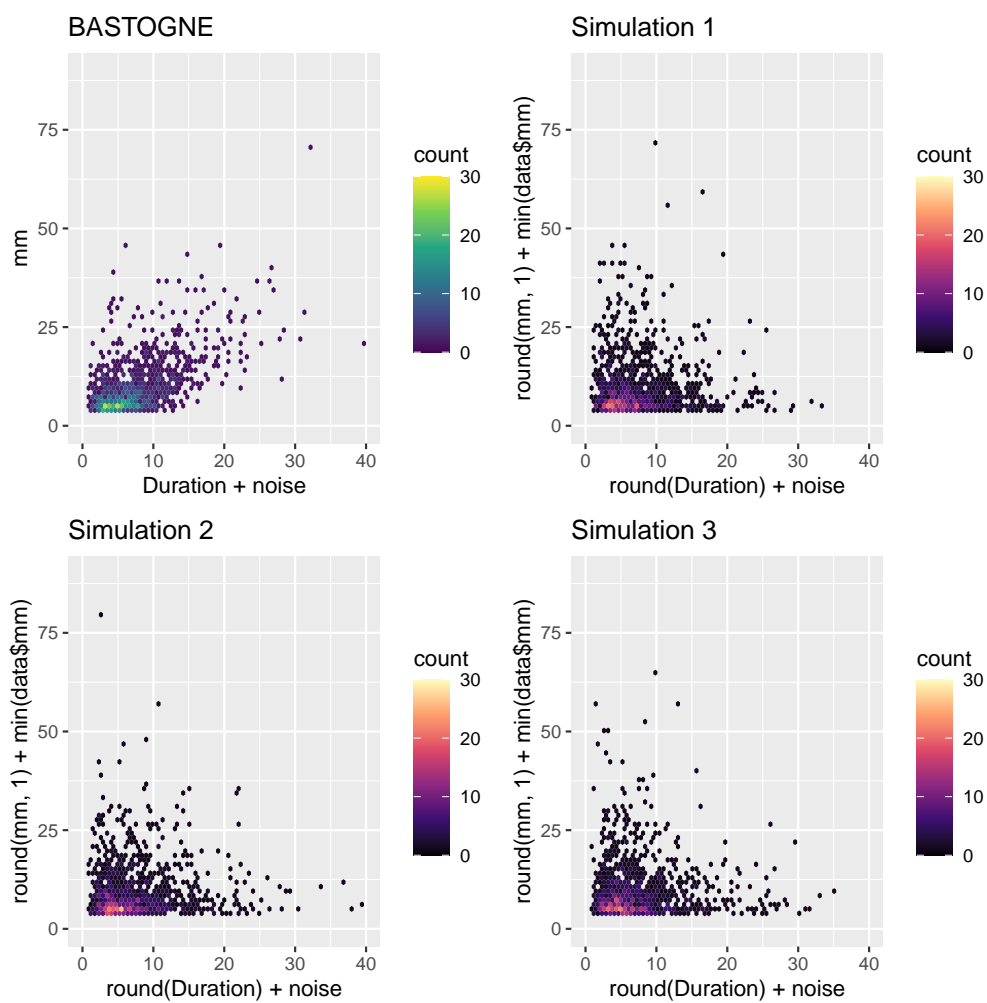


Figure 5.14: An uncorrelated simulator

5.5.4.2 Copula

In an attempt to capture the non-linear correlation between precipitation and duration, we looked at copulas Nelsen 2006.

A copula is a mathematical object that allows to approximate a joint probability distribution given the marginal distributions. In our case, we would like to have an approximation of the joint probability distribution of precipitation and duration of storms given we can express the marginal distributions, one for the precipitation, one for the duration. Remember that we have studied those marginal distributions above. Now we present the copula in more detail. We reproduce here the explanations found in Nelsen 2006.

A copula is a function C with the following properties:

- The domain of C is $I \times I$ where $I = [0, 1]$.
- C is grounded and 2-increasing which implies that C is non decreasing in each of its arguments.
- For $u, v \in I$, $C(u, 1) = u$ and $C(1, v) = v$.

In other words, a copula is a non decreasing function in each of its argument and its maximum value is reached at least in $C(1, 1)$. That's a definition that can be seen as a cumulative distribution function in 2 dimensions. Once the copula is defined, the connection between the joint distribution and the marginal distributions is established by the so called Sklar's theorem:

Theorem 1. *Let H_{XY} be a joint cumulative distribution function with margins F_X and G_Y . Then there exists a copula $C(u, v)$ such that for all $(x, y) \in \bar{R}$, $H_{XY}(x, y) = C(F_X(x), G_Y(y))$. If F_X and G_Y are continuous, then C is unique.*

This theorem tells us that a copula exists, it doesn't tell us how to find it. However, provided we have such a copula, we can generate random elements. There exist several procedures to do so but we give the one we understand. Before doing so we must establish an intermediary result (Schmitz 2003).

$$\begin{aligned}
P(X \leq x | Y = y) &= \lim_{h \rightarrow 0} P(X \leq x | Y \leq y \leq Y + h) \text{ (rewrite in terms of a limit)} \\
&= \lim_{h \rightarrow 0} \frac{P(X \leq x \cap Y \leq y \leq Y + h)}{P(Y \leq y \leq Y + h)} \text{ (conditional P)} \\
&= \lim_{h \rightarrow 0} \frac{H_{XY}(x, y + h) - H_{XY}(x, y)}{G_Y(y + h) - G_Y(y)} \text{ (def. of } H_{XY} \text{ and } G_Y) \\
&= \lim_{h \rightarrow 0} \frac{C(F_X(x), G_Y(y + h)) - C(F_X(x), G_Y(y))}{G_Y(y + h) - G_Y(y)} \text{ (def. of } C) \\
&= \lim_{h \rightarrow 0} \frac{C(F_X(x), \Delta_h + G_Y(y)) - C(F_X(x), G_Y(y))}{\Delta_h} \text{ (posing } \Delta_h = G_Y(y + h) - G_Y(y)) \\
&= \frac{\partial C}{\partial(G_Y(y))}(F_X(x), G_Y(y)) \text{ (def. of partial derivative)}
\end{aligned}$$

Posing $c_v(u) = \frac{\partial C}{\partial v}(u)$ and noting it is a CDF, we can pseudo inverse it, so we get $c_v^{(-1)}(u)$ which allows to go from $P(X \leq x | Y = y)$ back to $F(x)$.

1. Sample two independent values v, t from $\mathcal{U}(0, 1)$. v corresponds to $G_Y(y)$.
2. Evaluate $u = c_v^{(-1)}(t)$. u corresponds to $F_X(x)$.
3. One can then compute $x = F_X^{(-1)}(u)$ and $y = G_Y^{(-1)}(v)$

We still have to choose a copula. For that we started with a plot of empirical CDF of storm intensities and durations (see figure 5.15). Then we compared it to classical copulas (see figure 5.16). Based on the visual likeness, we selected the Gumbel copula.

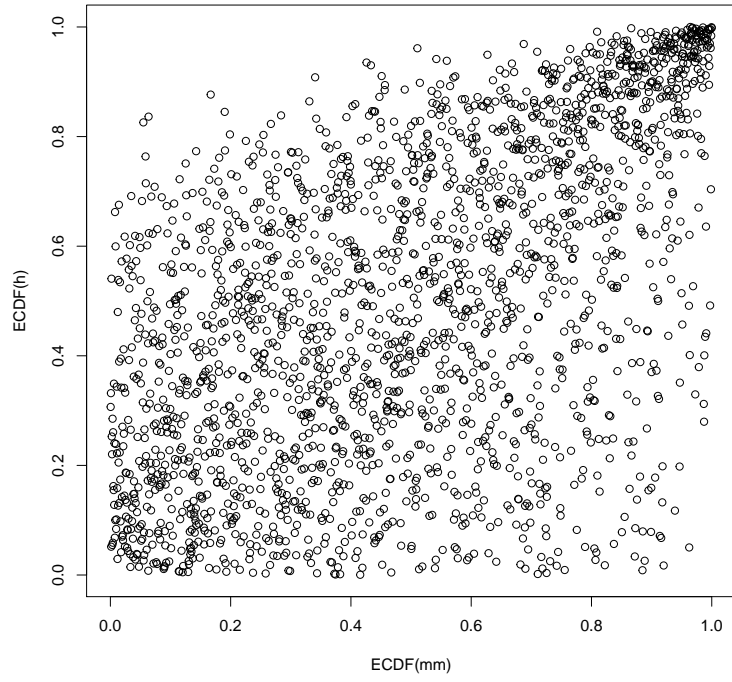


Figure 5.15: A plot of empirical CDF of storm intensities and durations

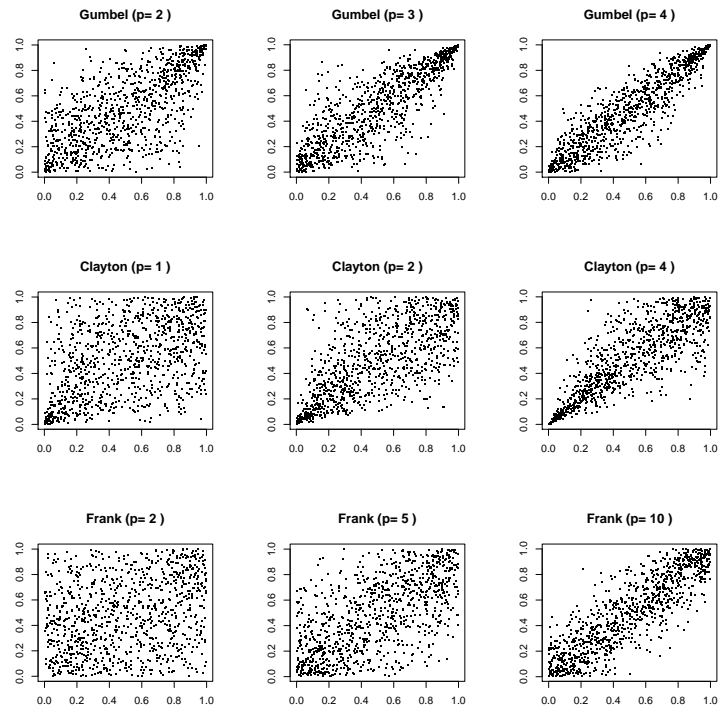


Figure 5.16: Copula gallery

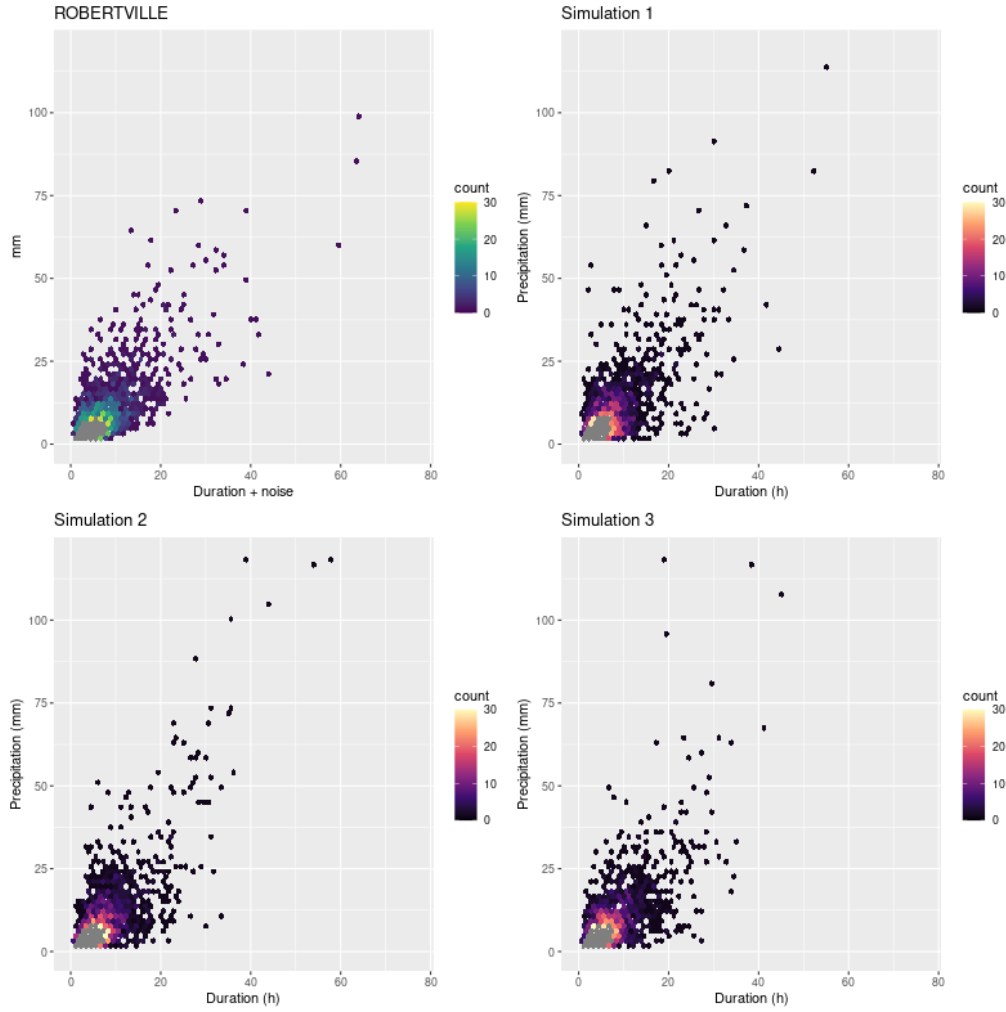


Figure 5.17: Copula for simulation. The upper left picture is the original data. The other charts shows 3 different simulation outputs based on the fitted copula and its fitted margins.

The copula's margins were then fitted with the distributions of the duration and precipitations (as described in previous sections) and a simulation was performed. The results are encouraging as can be seen on figure 5.17.

An important advantage of the copula, although we did not realize it as much as we wanted, is that one can model the extremes in the marginals and then have the copula to reflect those extremes. That allows to handle the extremes more gracefully than, for example, a kernel density estimation.

5.5.4.3 Other approaches

We thought of other approaches to model the correlation.

The first one was to use a kernel density estimation over the cloud of storm points (to be exact, a KDE would model the full distribution, not the correlation only).

With a small bandwidth, the KDE displays a lot of variance which is not good for our simulation as it means that we will reproduce original data (see figure 5.18). With a higher bandwidth, we still don't have much detail in the area of extremes and we introduce a high bias for the most frequent storms (the area at the bottom left of the chart is wider). Moreover, with higher bandwidth, the general shape of the density distribution trends towards a Gaussian one and since the marginal distribution of precipitation and duration are not normal, we thought this would ultimately be a dead end.

We also tried to model the density with a mixture of 2D skewed Gaussian curves (Azzalini and Valle 1996). Our goal was to skew the 2D Gaussian in a way to get a corner close to the origin so as to approximate the shape of our storms distribution. We didn't investigate much because we understood

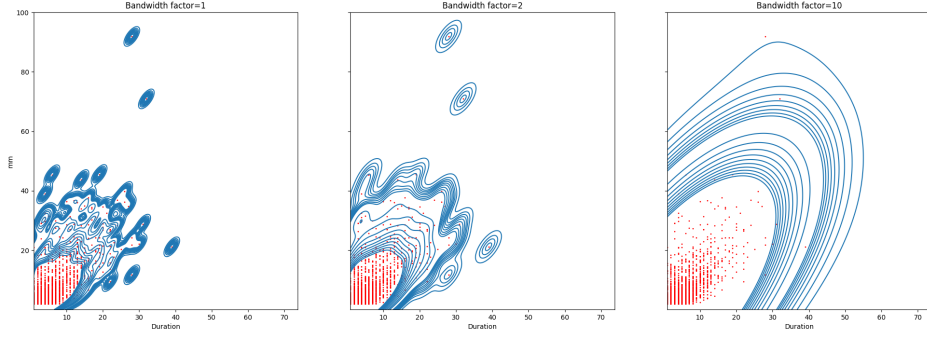


Figure 5.18: KDE experiment. The red dots are actual storms. The white hole shows probability mass below 90%. The lines show probability mass below 90% to 99% (inwards to the graph) and to 99% to 99.9% (outwards), ten lines each. Left: high variance, right: too much smoothing, center: still some islands.

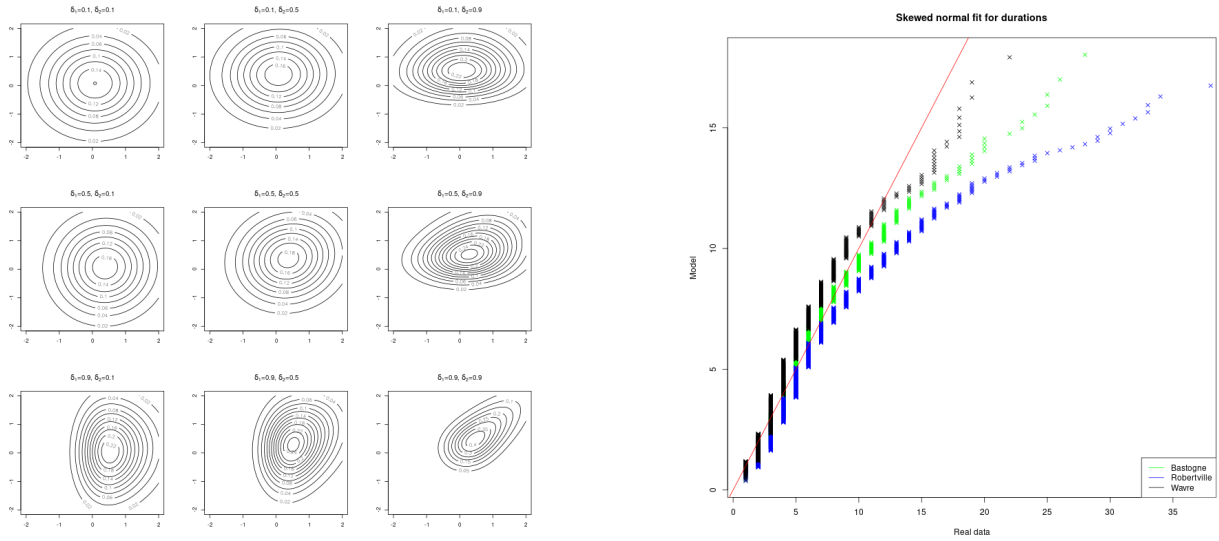


Figure 5.19: Skewed 2D Gaussian experiment

that the skew parameters (δ_1, δ_2) didn't allow that. See figure 5.19 where various values are chosen for the parameter and where the global shape never approximates a corner shape. After reconsideration, we think that it might be possible to truncate the distribution to get the proper shape but we didn't have time to investigate.

5.6 Storm hyetographs

Once we have the storms durations and precipitations, we still need to find their hyetographs. In this chapter we present several ways to do that.

5.6.1 Flat hyetographs

The first test we made was to produce histograms that are just the regular box we modeled so far. That was the simplest way to go. Obviously, it doesn't produce much extreme (see figure 5.20).

5.6.2 Triangular hyetographs

The previous attempt was not very satisfying, so we tried to make triangular histograms in a way akin to the SHYPRE method. Since we couldn't apply the full SHYPRE method (see 3.4), we settled

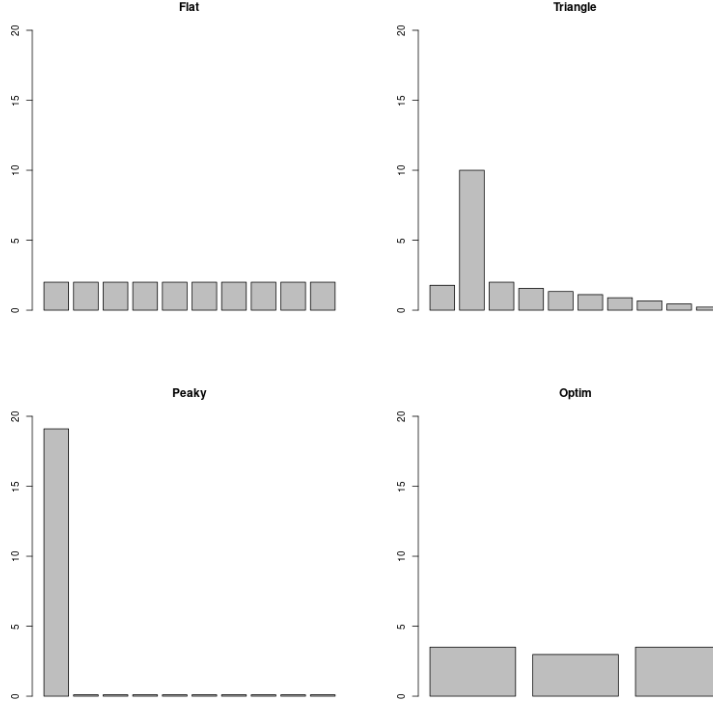


Figure 5.20: Different shapes of storms

on making storms with only one peak, always located in the middle of the storm. This is of course a gross simplification, but since we don't have much information about how to model a storm in this way, we took the simplest approach.

For that we collected all storms and computed for each the ratio between the maximum hourly rainfall and the average hourly rainfall. We then fitted a truncated (because the ration is always greater than one) gamma distribution on these values. As can be seen on figure 5.21, the fit is rather good.

So, to make a triangular storm knowing its duration d and total rainfall p , we first compute its average hourly rainfall as $a = p/d$. Then, using the fitted distribution we pick a corresponding ratio r . From this we compute the hourly maximum of the storm as $m = r \times a$. Now we can put m mm of rain in one hour slot of the hyetograph and then sprinkle the remaining millimeters ($p - m$) into the remaining hours. We start from the smallest possible amount (the noise) and go up with an increment of $\frac{ad-m-hn}{\frac{1}{2}(h-1+1)(h-1)}$ with n the noise level and $h = d - 1$.

This obviously produced better results than the flat storms as far as extremes were concerned.

5.6.3 Peaky hyetographs

As our method was not producing enough extreme, we took a radical approach which is to build storms with only one peak concentrating all the precipitations and leaving the remaining hours at the level of noise. This produced extremes for sure but these were totally unrealistic.

5.6.4 Probable hyetograph with MLE

Here we try another possibility. Since we will validate our simulator with IDF curves, which are built for various durations, then it should be useful to try to model the probability of rainfall in these periods. Once this is done, we could build our storm hyetograph in a way that respects those probabilities. The method we describe below is the fruit of discussions we had with Pierre Archambeau who first proposed to solve the hyetograph as a constraint optimization problem but for which we thought the optimizer would have difficulties. So we changed it to a probability problem.

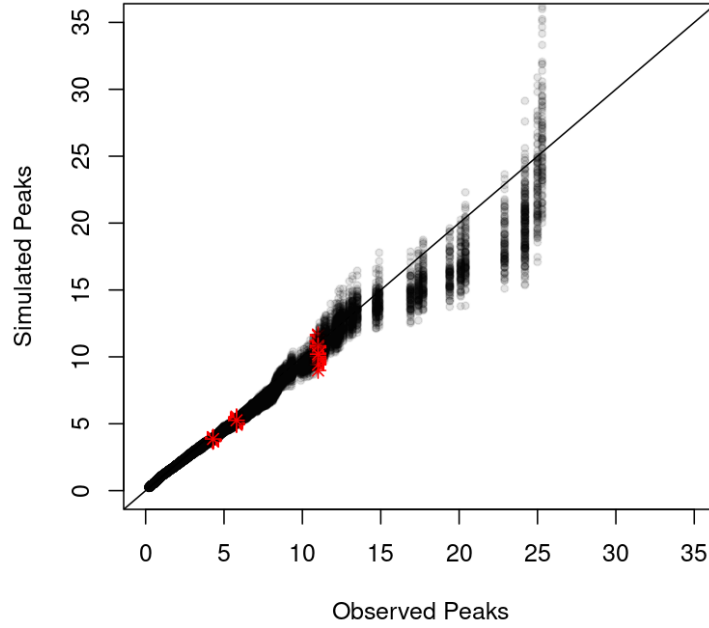


Figure 5.21: Peak versus average distribution fit. QQplot of a hundred simulations versus one series of observations. The red stars represent 0.9,0.95,0.99,0.999 quantiles.

First, we compute the probability distribution of precipitations for periods of one hour $f_1(x)$. We compute that over the whole precipitation data, irrespective of the storm structures we might extract. This corresponds to the precipitation reported on the duration = 1 hour of the IDF curves. Then we move onto periods of 2 hours. For each of them, we compute the total rainfall and again we fit a PDF over these values, $f_2(x)$. Then onto 3 hours, 4 hours, etc.

Now, given a storm duration S_d and its total precipitation S_p , we want to build its hyetograph h (h_i is rainfall during the i -th hour of the storm) in a way that respects the probability distributions above, at best. We define the notion of “at best” as the highest possible product of all the probabilities of each of its parts. Formally, if we denote by $h_{i,p}$ the sum of precipitation in the i -th period of p hours in the hyetograph of a storm, with $p < S_d$, $1 \leq i \leq S_d - p + 1$ and $f_p(x)$ the PDF of precipitation over a period of p hours, then we want to choose the quantity of rainfall in each hours of the hyetograph that maximizes $\prod_{p=1}^{S_d} \prod_{i=1}^{S_d-p+1} f_p(h_{i,p})$ under the constraint $\sum_{k=1}^{S_d} h_{k,1} = S_p$. This general rule accepts two special cases. First, when the storm to create has a duration S_d of one hour, there is nothing to optimize since the precipitation for the whole duration is given by S_p . Second, it is useless to compute the PDF for $p = S_p$ because the total precipitation is a given. The constraint will be represented as a penalization on the product, so our objective function is $|\sum_{k=1}^{S_d} h_k - S_p| \times \prod_{p=1}^{S_d} \prod_{i=1}^{S_d-p+1} f_p(h_{i,p})$, in other words, the further away the total rainfall is from S_p the more we penalize.

Note that the objective function is not a probability of the storm’s hyetograph. But if we divide it by a factor equal to the integral of all possible hyetographs, then we can assimilate it to a probability.

The question is now how we build the probability distribution $f_p(x)$ which, given a duration, provides the probability (it is not exactly a probability, but we do as it was) of the precipitation. After examining the data, it appeared to us that a regular log-normal fit was a good solution for the small durations (see figure 5.22) even more so when we looked at the 0.9,0.95 and 0.99 quantiles which were rather well fitted.

For the longer durations, things degrade sensibly (see figure 5.23). However, one notes that the longer durations represent only a tiny fraction of the data set. So, for the average situation we’re good enough. For the extremes, we don’t have any way to make good predictions. Note also that the durations go even further than those shown in the figure, in which case we usually don’t have

enough data to even make a fit. In both cases, we arbitrarily decide to not fit at all and just don't use any constraint for the more extreme durations. This basically means we will have missing f_p in the objective function. Since the objective function formula remains the same during the search of the best hyetograph, that is, it is only used to compare two hyetographs, these holes mean that we will compare "as best as we can", with the PDFs we have. So we will still be able to find a best hyetograph, but it will not be as good as expected.

Because of the nested products, the computation time of the objective function is $O(S_d(S_d+1)/2)$ which is exponential and this behavior sometimes leads to non practical computation times. So we decided to split long storms in two, optimize them separately and merge them back. A long storm is 25 hours or more. This threshold was chosen on the basis of time needed to compute and the fact that storms of that size are uncommon (about 1 percent). This removed a few very long searches, earning us some hours.

Now that we have set up the problem, we have to look at a strategy to solve it. The first one is to use an optimization procedure. We choose a regular L-BFGS-B to solve our problem.

We had to bring some improvements:

- We noted that using the L1 norm for the penalization was not always enough to enforce the constraint, so we moved to L2 norm (which also has the advantage of not pushing specific hours to zero).
- Unfortunately, the optimization results tend to be a bit "non natural". If one looks at figure 5.24 we see the optimization result for a storm where $S_d = 3h, S_p = 10mm$ has two of its hours close to their minimum while the third one is close to its maximum (9 is close to the max. for the storm, S_p). So the solution is unbalanced. It is not always the case but it happens quite often. To counteract that issue we have brought a final modification to the objective function. We now divide it by the entropy of the proposed hyetograph:

$$\mathcal{O}(h) = \frac{(\sum_{k=1}^{S_d} h_k - S_p)^2}{\mathcal{H}(h)} \prod_{p=1}^{S_d} \prod_{i=1}^{S_d-p+1} f_p(h_{i,p})$$

Doing so we penalize hyetographs with lots of variability. The results are shown on figure 5.25. As one can see, the selected solution (denoted by a red point) gives more precipitation to h_1 and h_2 , balancing the hourly rainfall better.

- The objective function we presented above is not really suitable for actual computations because it is a product of small quantities, so we take its logarithm. As we compute the log of probability densities, it happens that a PDF evaluates to something greater than 1, leading to a positive log. The previous computations all make the assumption that log values are negative (because the penalization, that is making log-likelihood more negative, are done by way of multiplications with positive quantities). So we change our log transformation to $\log(o+p)$ where o is a constant large enough to ensure that the log always gives a negative value and p is the PDF's value (expected, but not always, lower than one).

The objective function becomes:

$$\mathcal{O}(h) = \frac{(\sum h_k - S_p)^2}{\mathcal{H}(h)} \sum_{p=1}^{S_d} \sum_{i=1}^{S_d-p+1} \log(o + f_p(h_{i,p}))$$

5.6.5 Probable hyetograph with Metropolis Hasting sampling

After running the previous algorithm some times we were disappointed by the speed: a regular simulation would take about an hour. Since we were using bare R functions, we had no way to optimize them (and it is not unreasonable to think that R has already received thousands of man hours of optimization). So we decided to explore a trade off. Since L-BFGS-B doesn't give us a guarantee as to a global optimum, we could try another algorithm which could use our objective function provided

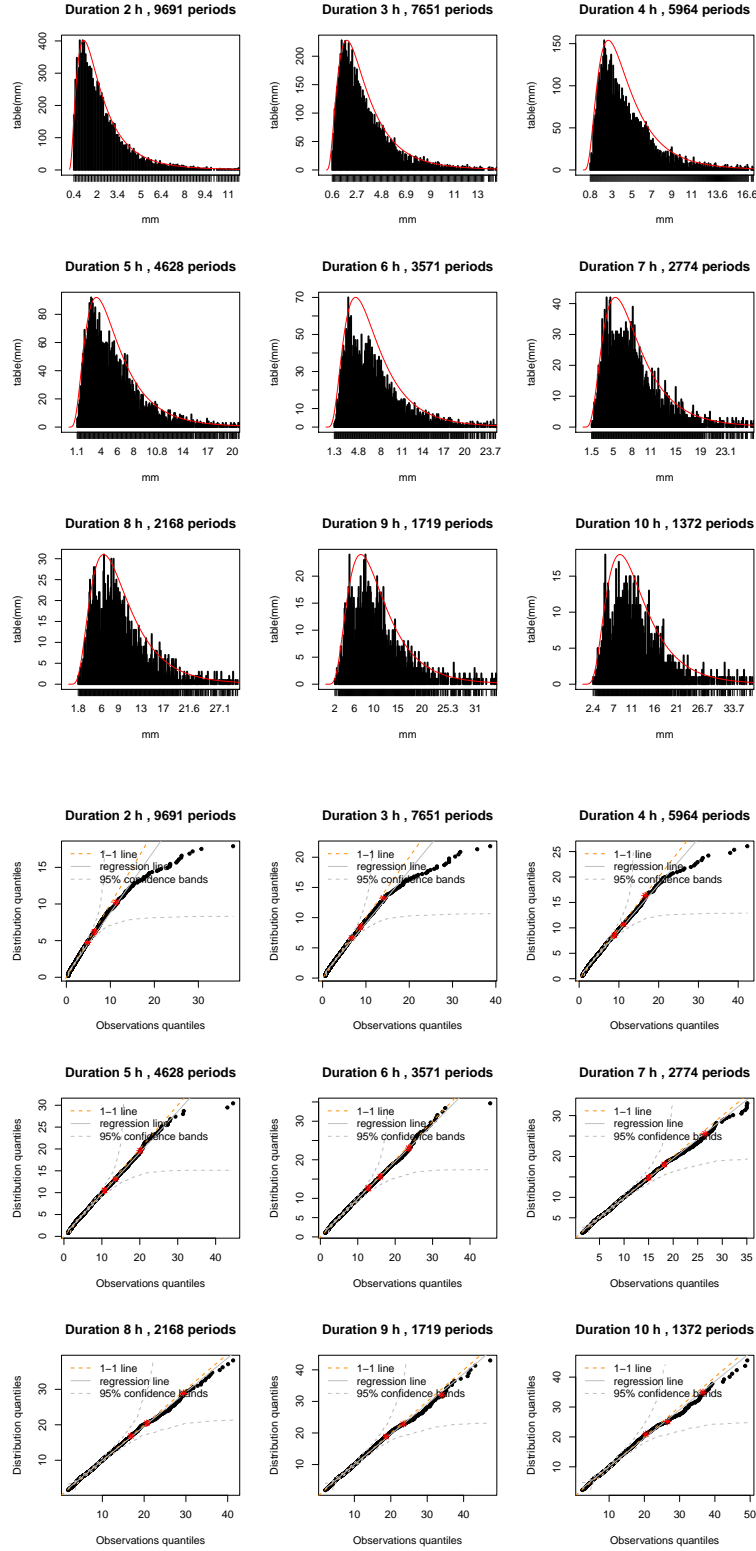


Figure 5.22: Log-normal PDF fit of periods of durations from 2 to 10 hours. We present histograms and QQ-plots. In the qqplots, the red dots represent the 0.9, 0.95 and 0.99 quantiles.

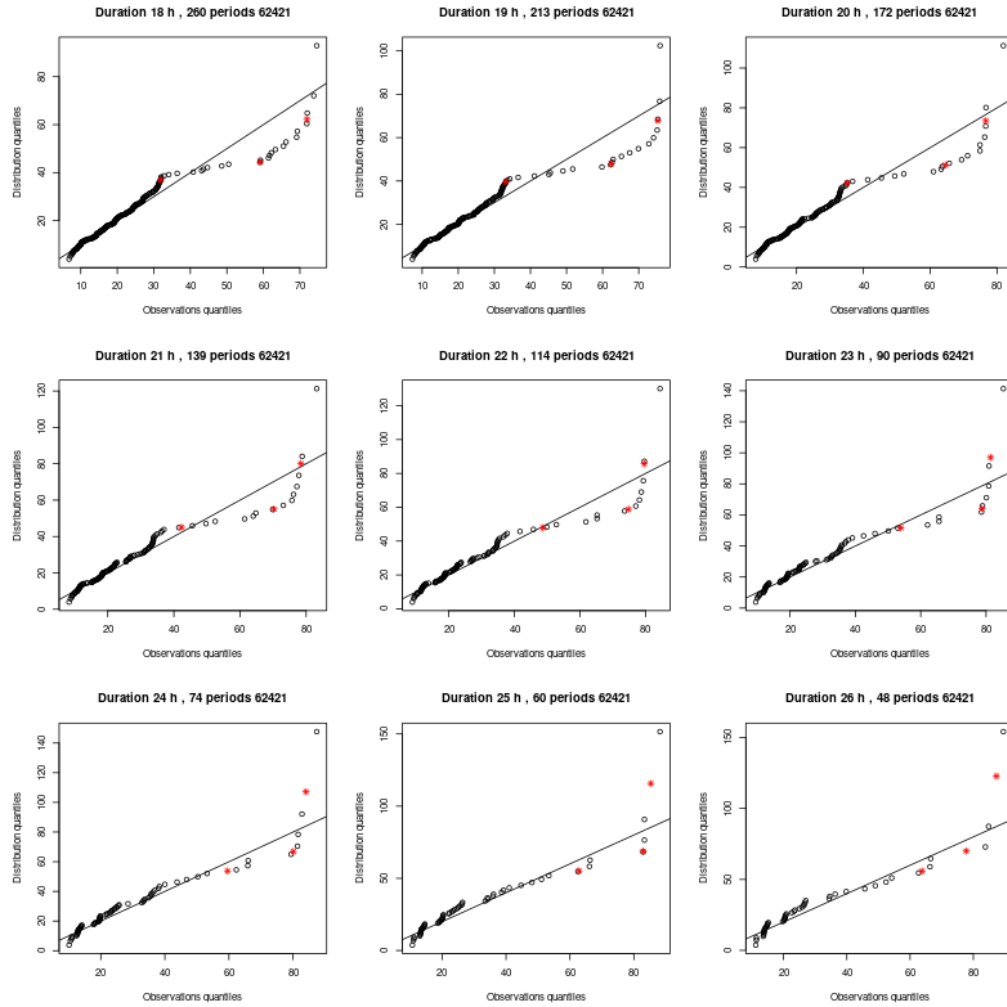


Figure 5.23: Log-normal PDF fit of periods of durations from 18 to 26 hours. We present the QQ-plots, the red dots represent the 0.9,0.95 and 0.99 quantiles.

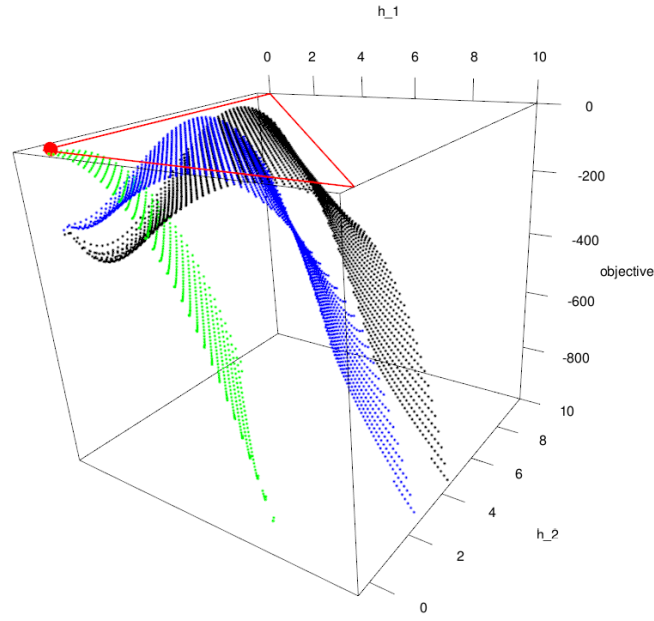


Figure 5.24: The optimization result of a probable hyetograph of 3 hours. The red triangle is a visual clue for the sum of the first two hours of the hyetograph (≤ 10). The red point shows the result of the optimization. The colored curve show different values of the third hour of the hyetograph (black $h_3 = 2$, blue $h_3 = 4$, green $h_3 = 9.5$). Solutions on the green curve will push the other two hours to 0.

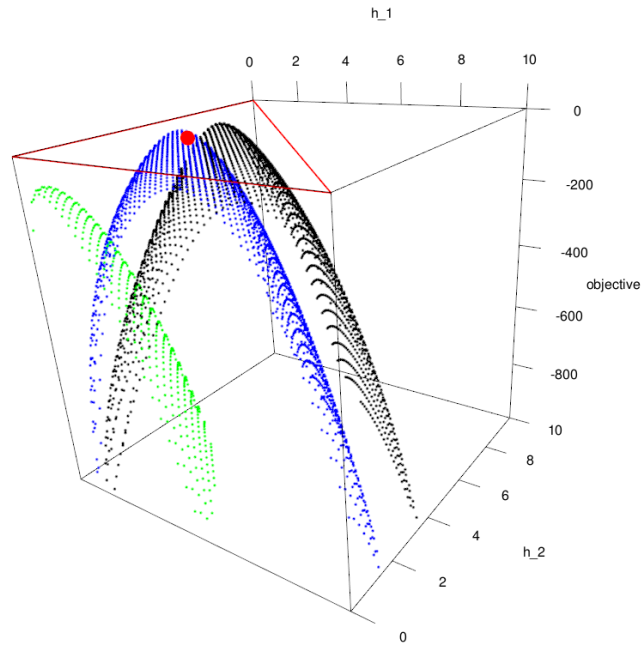


Figure 5.25: Likelihood Optimisation with additional entropy constraint

it offers some reasonable convergence. The algorithm had to be simple enough to be implemented quickly. Moreover, we wanted to investigate some basic MCMC algorithm since we didn't have the opportunity to do it in our courses. So we chose Metropolis Hasting sampling.

Let $h \in \mathbb{R}^d$, d is the duration of the storm:

1. The objective function is the one described above, $\mathcal{O}(x)$.
2. h is an array representing the hourly rainfall for d hours, initialized arbitrarily.
3. $c = []$, a "chain"
4. For a given number of iterations:
 - (a) A proposal is made: $h' \leftarrow \mathcal{N}^d(h, 1)$
 - (b) The acceptance probability: $a = \frac{\mathcal{O}(h')}{\mathcal{O}(h)}$
 - (c) if $\mathcal{U}(0, 1) \leq a$ then we append h' to c , else we append h to c . So if $a \geq 1$, the proposal is automatically accepted, else it will be accepted with a rate proportional to its quality.

We note that by construction, the chain will end up converging to the distribution embodied by $\mathcal{O}(x)$ (see Hauser 2013, Bendel 2020), provided the chain is long enough (Robert 2015).

That algorithm was not much faster in R since obviously, it must evaluate the objective function a lot of times. However, as we wrote it ourselves, we could optimize it. The easiest for us was to rewrite it in a compiled language, we chose rust. This improved performance by a factor of roughly 10 (we don't have much data about this, but rust vastly outperforms R at optimizing all the "glue" between the math intensive tasks such as PDF evaluations and function calls).

The "burn in" was set to 5000, we didn't have enough time to conduct tests to optimize that number.

The proposal function in MCMC was initially written as a variation around the last valid proposal: $x' \leftarrow \mathcal{N}(x, 1)$. But this doesn't work when we are close to the noise level. Indeed, the normal distribution will generate mostly unacceptable proposals (those with components of x smaller than 0.1 or even negative). To avoid that we used the rule $x' \leftarrow \mathcal{U}(0.1, 0.4)$ when $x \leq 0.2$.

5.6.6 Other approaches

We thought of generating the hietograph of size S_d directly from a global probability distribution with a KDE but in S_d dimensions. We don't think this would work because as soon as we reach longer storms (on the order of $S_d > 10$), the number of storms at our disposal to build the distribution decreases drastically. Even at $S_d = 2$, we can see that we already miss a lot of information (see figure 5.26). Although we could have just said that what we miss is extremes and go on, we feared that this issue would grow bigger as the number of dimensions increased.

Another possibility would have been to use copulas again. But there we would have faced the same issue which is the lack of storms of higher (say 10) durations.

Our current approach does a lot of grouping and therefore allows us to work even when we don't have many storms to extract the PDFs from.

5.7 Noise reinjection

As we have noted in the very beginning of this chapter, we remove noise from the data. However, as noise participates in various quantities (e.g. total precipitation per season) we measure when analyzing the simulation results, we have to reinject some artificial noise to compensate.

For the hourly noise, we removed hourly precipitation below a threshold. To reinject the noise, we first compute the frequency of hours with noise throughout the full time series and the average noisy rainfall present in them. Using these two quantities, it is easy to sprinkle the average precipitation on as many hours as necessary, provided these hours are not already covered by storms.

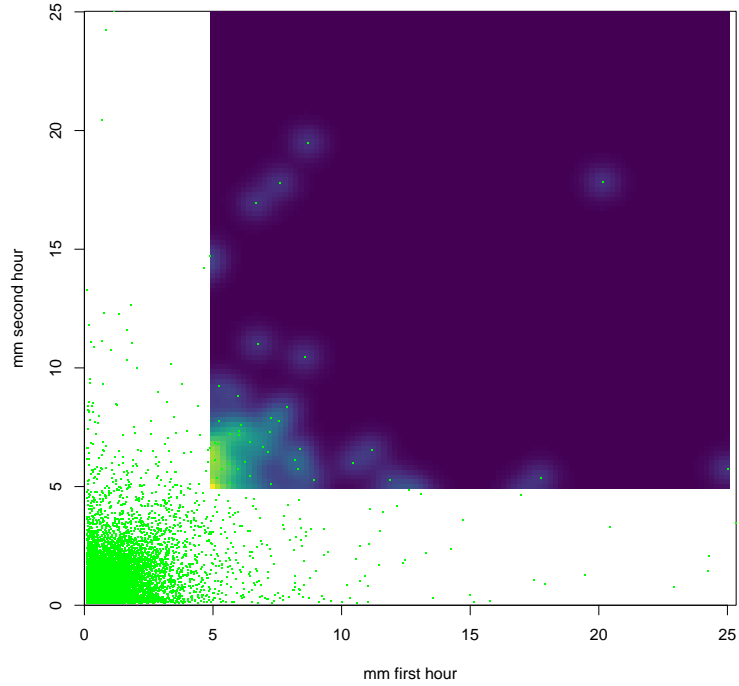


Figure 5.26: 2-hours hyetograph distribution. Each point represents a pair of precipitation for the two hours of the hyetograph that comes out of our data set. On the upper right area, the density is represented (background color) as one can see it is mostly zero.

For the storm noise, we follow the same procedure. However, we didn't model the noisy storm durations and simply synthesized storms of a duration of one hour. This simplification is possible because the precipitation in each of these storms is very small on average ($< 1\text{mm}$).

Chapter 6

Simulation

In this chapter, we present the simulator we have written and we conduct an assessment of its behavior.

6.1 Simulation operations

6.1.1 General view of operations

Before the simulator execution, all the data series for all used measurement stations must be collected and their data must be cleaned. This process is done once and doesn't need to be repeated for each execution.

Once the data collection is complete, the simulator can be executed. The simulator execution is linear and can be decomposed in two phases. First we compute all the parameters of our data model (we could do that in the previous operation but we have left it here because it allowed us to iterate faster on the simulator's design — the computations of the parameters takes a very small fraction of the whole execution):

1. Split the data series by stations and split again in storms
2. Compute various distributions over these storms; one parameters set per season:
 - (a) Distribution of durations, as a mixture of gamma distribution
 - (b) Distribution of precipitations, as mixture of exponentials
 - (c) Copula linking duration and precipitation
 - (d) Number of storms per season per year as a skewed normal distributions
3. Compute a noise model:
 - (a) Noise rate for periods of one hour
 - (b) Noise rate for too small storms.
4. Compute models for hyetograms:
 - (a) Distributions of precipitations over periods of 1,2,... hours (gamma distributions) for MCMC and optimized shapes.
 - (b) Peaks distribution for triangular hyetographs

Second we generate data:

1. For a given number of simulations, each over a given number of years
 - (a) Pick a number of storms
 - (b) Generate (duration, precipitation) pairs for those storms
 - (c) For each storm, depending on the requested storm shape:

- i. triangular: compute a peak value and then build a hyetograph
 - ii. flat or peaky: build a hyetograph
 - iii. “optim”: optimize storm shape using L-BFGS-B and objective function
 - iv. “MCMC”: optimize storm shape using Metropolis Hasting
2. Build a final time series by putting all hyetographs on a full time line.
3. Store the resulting hyetograph on secondary storage for later use, such as reporting.

6.1.2 Performances

Execution times are in the order of minutes for one simulation run covering 20 years (about 2500 storms) on a regular PC for box storms synthesis. Hyetogram synthesis can take less than 5 minutes (flat, triangular, peak, MCMC shapes) or up to an hour (optimized shape). Execution time is roughly linear with the number of storms to generate. Memory usage is well under 1Gb.

6.2 Validation

Once the simulator has produced its storms, we need to validate that the output corresponds to expectations.

6.2.1 Expectations

To build the expectations, we have two concurrent paths:

- Use the principle that the data at hand show what reality is. Therefore a simulator should be able to reproduce the observed data faithfully (but not exactly, as the point of the simulator is to create new situations). In that case we’re looking at the average behavior of the simulator.
- Use the advice of experts to define an operational envelope: the simulation is good if the experts say it is good. This is useful in case, for example, the simulator makes extremes which are more extreme than the observations: are those extremes realistic ? Experts also come in the form of the articles we have read.

During the course of this work, we have mainly followed the first path but the second most probably permeated through various design biases and discussions with the team. Indeed, the way we built the simulator reflects the hypothesis we make about the formation of storms.

Coming back to our expectations, we can say that a good simulator would:

1. Produce average storms that resemble reality. Here we would check a few measures:
 - (a) The average storm duration and intensity
 - (b) The number of storms produced in a season
 - (c) A indication of the probability distribution of the intensities and duration of storms
2. Produce extreme storms that resemble reality. This is a bit more useful for the hydrologist. Indeed it appears that simulators are used to plan the construction of infrastructure. So the infrastructure must be built to last for several years or decades. In that period, an infrastructure, for example a zone to hold water, must be able to work correctly under the stress of an extreme event. So the simulator must be able to produce those extreme events.
 - (a) The tool of choice to validate that is the IDF curve.
3. Produce a meaningful confidence interval.
 - (a) Given the simulator has no analytical form, a set of different runs would help to build a view on the variability of the results.

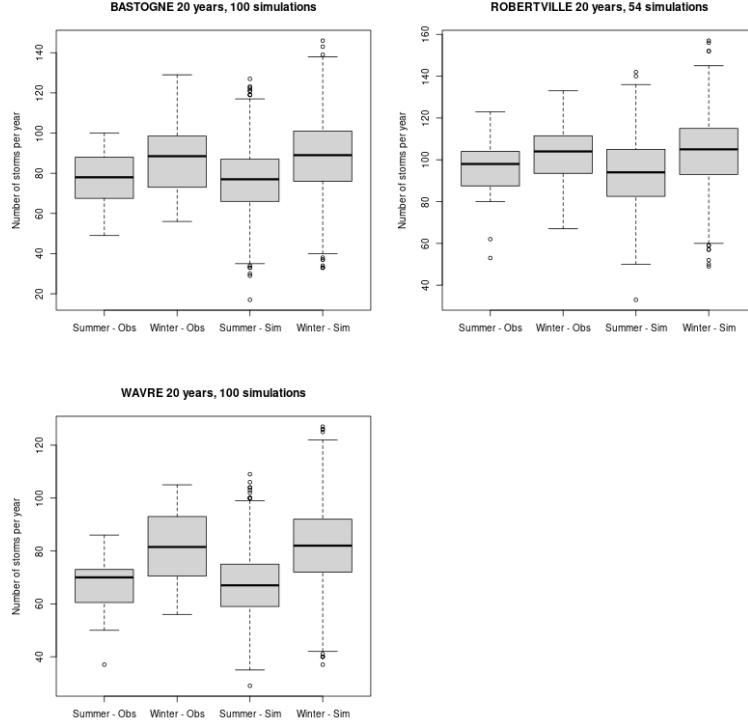


Figure 6.1: Storms per seasons

6.3 Global measures

In this section, we give a look at several “global” measures. They are global in the sense they gather very high level information about the simulator.

Looking at figures, we see:

- the simulated number of storms per season (fig. 6.1)
- the total rainfall per season (fig. 6.2)
- the total number of hours in storms (figure 6.3)

In all of these, the simulation is in line with the observations for the average case. The difference between seasons is present. The simulator tends to make some more extremes but, obviously, we are comparing several simulations series to only one series of the observations so this may be the cause: the simulator has many more opportunities to produce extremes.

6.4 Box storms distribution

As we have seen, the simulation is built on the notion of a box storm. Therefore, we first assess the quality of the generated storms. In the previous section we looked at global indicators and now we go one step down in the detail level.

The most direct visualization of the individual storms is given in figure 6.4. There we can see that both the clouds formed by all the sorts of pairs (represented as points) are of similar shape; their extremes are alike. We observe:

- The overall shape of the simulation charts resemble the observations: the correlation modeling gives good results.
- A tendency for our simulation is to couple the extremes in both dimensions more than the observed data. That is, when an intensity is strong, then the corresponding duration tends to be strong as well. In the observations, the duration extremes dominate more.

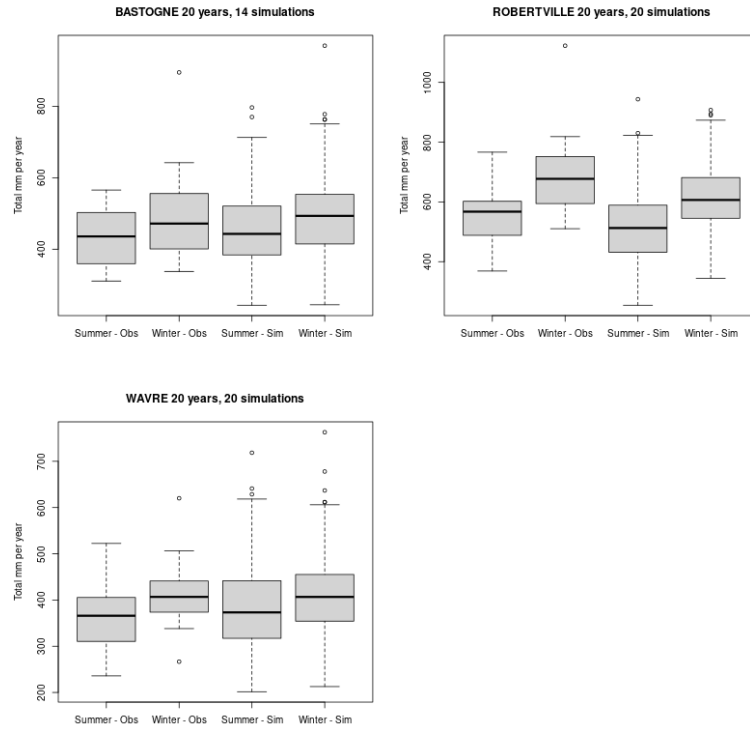


Figure 6.2: Total rainfall per season

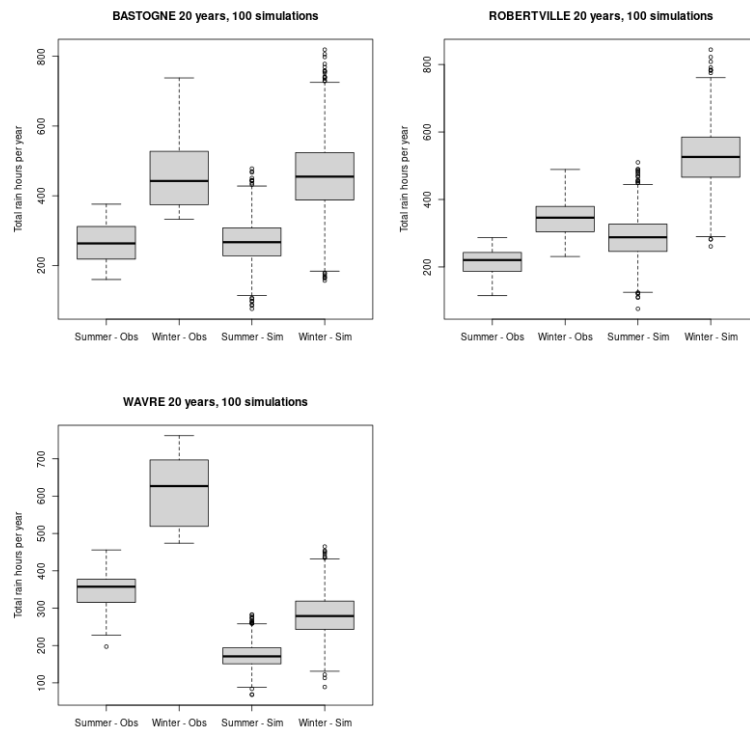


Figure 6.3: Number of hours counted inside storms

- The simulator tends to produce heavier rain (high mm's) more often (there are more storms above the horizontal dotted line).
- The simulated high intensity storms (red triangles) seem to occur according to the observations, although with a bit more extremes. Those storms are of interest for crisis management.

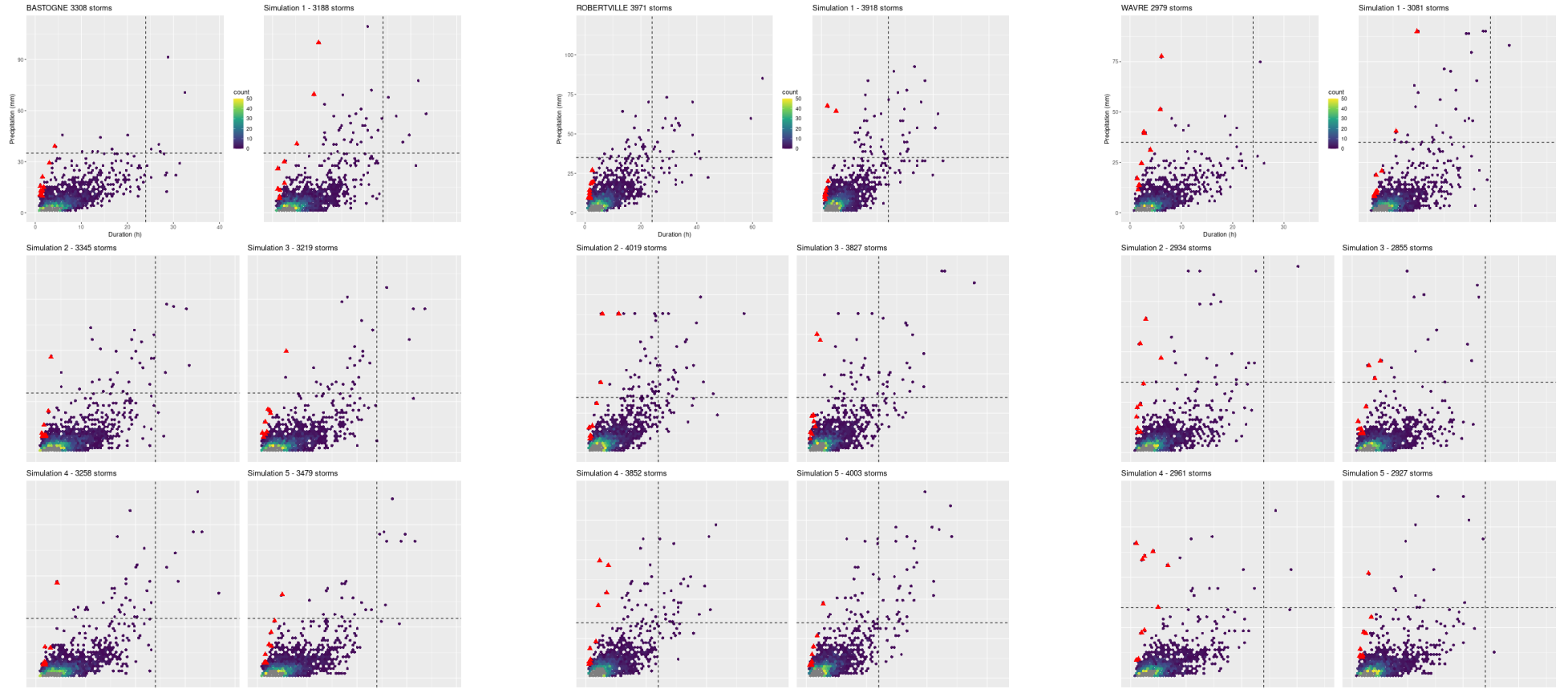


Figure 6.4: Copula generated storms for three stations. The top left chart is the observation, the other charts are one simulation each. The red triangles denote the 10 most intense storms. The gray bins contain more than 50 storms. The dotted lines are visual clues to help compare the simulation charts to the observation one. The axis are equal in all the charts of a given station.

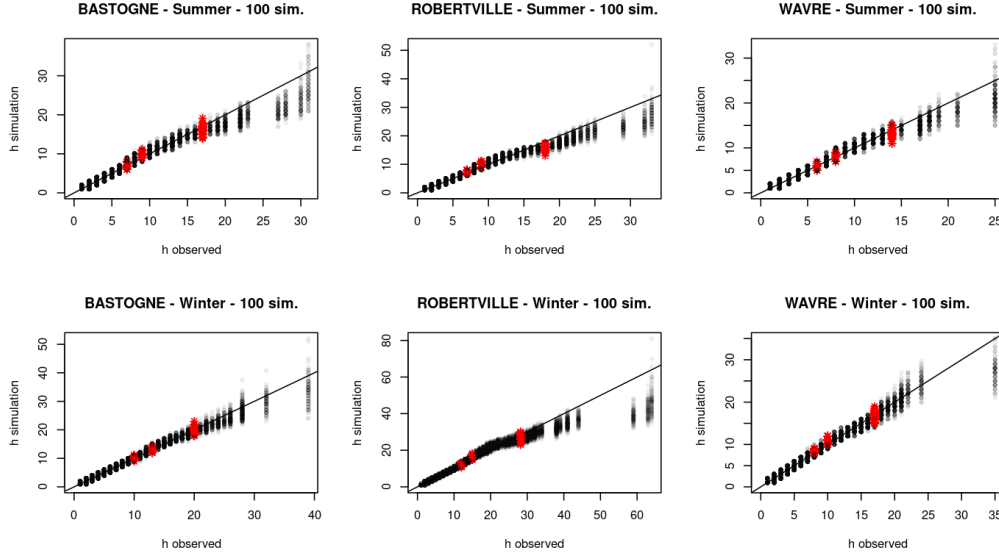


Figure 6.5: QQ plots for storm durations, one point represents one storm. Recall that the number of hours is discrete and so the charts are jagged. The red dots represent the 0.9, 0.95 and 0.99 quantiles.

We now take a closer look at the marginal distributions of the durations and precipitations:

- For the durations, the fit is rather good. But the simulation seems to underestimate in summer.
- For the precipitations (see figure 6.6), the simulation tends to provide stronger extremes (when over the 0.99 quantile). That reflects what we observed above. Moreover, in those extremes, there's a lot of variability which makes the simulation unpredictable.

The last chart (figure 6.7) gives a representation of the error on the number of storms in all simulation. It is an aggregation over the number of storms of a given duration (d) and rainfall (r), $S_{r,d}$. The aggregation is done over all the storm numbers of all cities.

The formula is

$$E_{r,d} = 100\% \times \sum_{\text{city}} \frac{\|\overline{S_{r,d,\text{city}}}^* - S_{r,d,\text{city}}\|}{1 + S_{r,d,\text{city}}}$$
 where $S_{r,d,\text{city}}$ is the observed number of storms of shape (r, d) for one city, \overline{S}^* is the average of the simulated number of storms over 10 simulations. The “1+” term in the denominator is here to avoid division by zero, be aware that might affect the computation in a significant way if $S_{r,d,\text{city}}$ is close to 0. This was computed over 20 cities.

As one can see, the number of small storms is better approximated than the one of the bigger (longer, more intense) storms. In the area of 90% of the storms, error can rise up to 50%. The big errors are located in the bottom of the chart, in 3 or 4 places. In these places, the number of observed storms is quite close to 1 and thus just a few simulated storms mean a pretty big error. Of course, the simulator being random, we expect it to have an “error”, but these numbers seem pretty high.

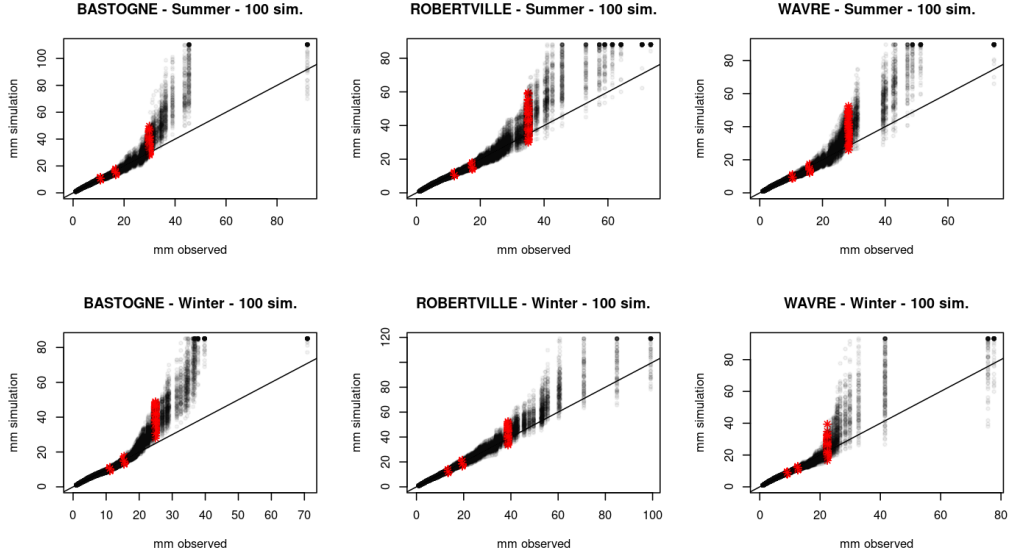


Figure 6.6: QQ plots for storms precipitations. Recall that we compare with observations which do not cover all the mm values so the charts are jagged. The red dots represent the 0.9, 0.95 and 0.99 quantiles.

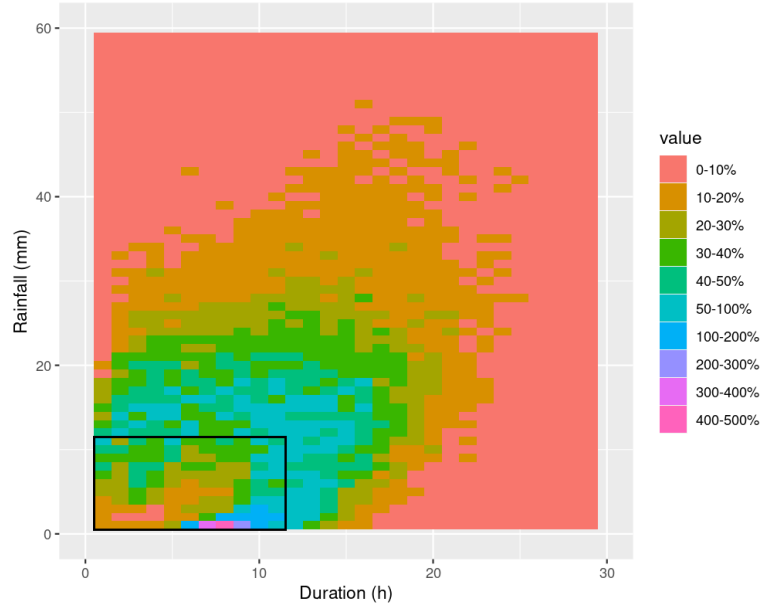


Figure 6.7: Average error on the predicted storms number, computed for each shape of storm, over several stations. The black square denotes the area where roughly 90% of the storms are located.

6.5 Hourly statistics

We first look at the hyetographs produced by the various “shapes” we have used (figure 6.8). We visually evaluate the realism of our storm shapes. The triangle shape gives the best result (or maybe less bad) and, unfortunately, the MCMC approach is quite disappointing considering the effort it took (we will nonetheless analyze its results). All our shapes lack a “multi-peaks” feature.

On the figure 6.9 we show the quantity of water associated with each precipitation level. That is, a point (x, y) on the chart means “it rained a total of y mm in the hours where it rained x mm, during the simulated period (20 years)“. This gives us an indication over how the simulator distributes the hourly rainfalls. This is also a statistic which is measured quite differently from the way synthetic rain is built. So we expect it to reveal more information about the simulator behavior, instead of just

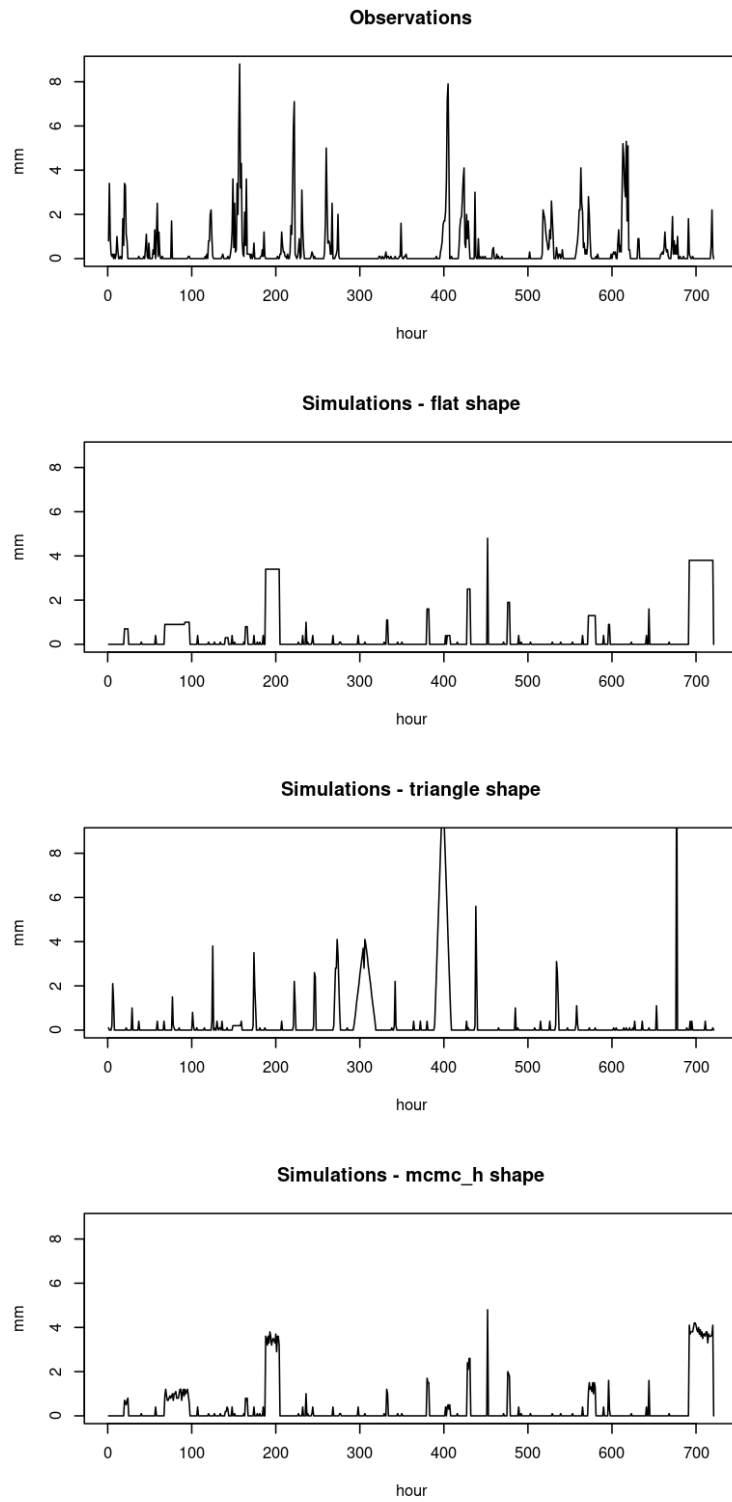


Figure 6.8: Hyetograms of various shape simulations. Example periods picked where there is the most rainfall.

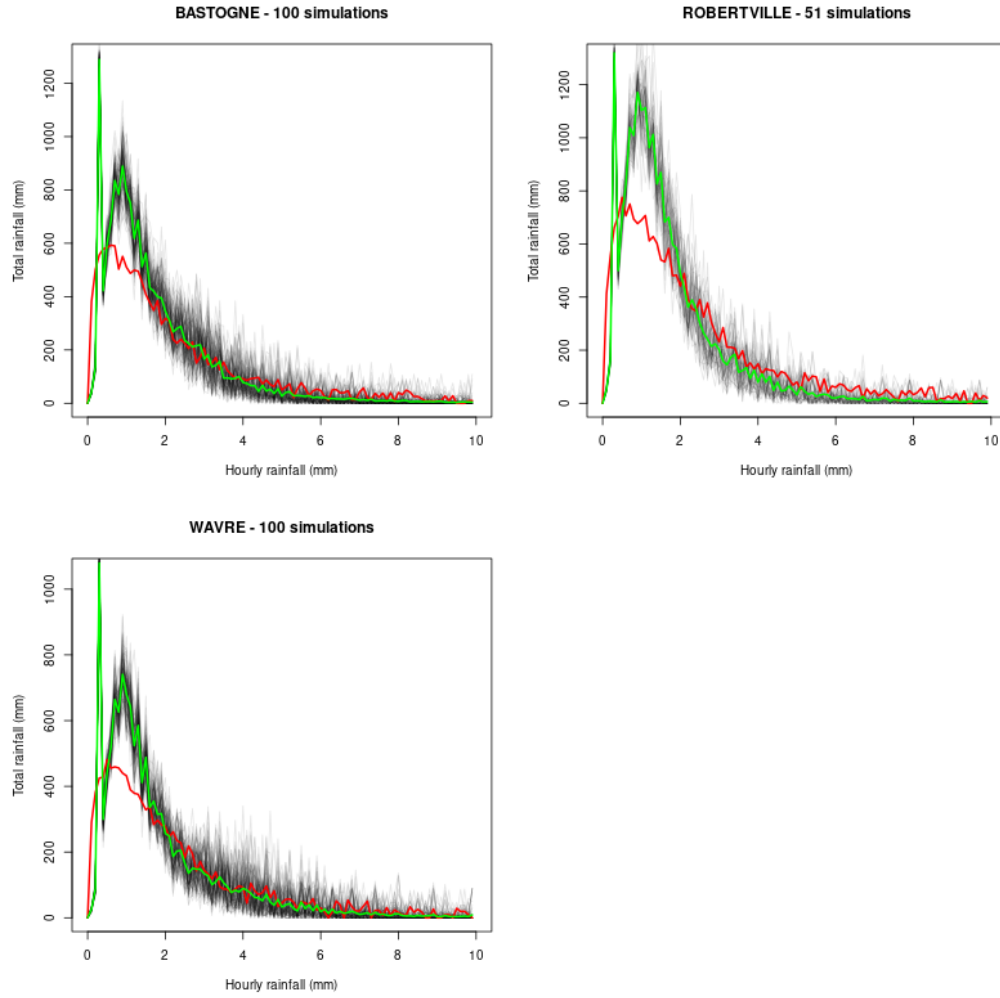


Figure 6.9: Total rainfall per hourly amount of rainfall. The red line is the observed values. The green line is the average over all simulations.

confirming that it follows the distributions built in. We see the following:

- The simulator distributes water roughly as the observations (hunch-like distribution).
- The simulator puts more water in the low hourly rainfalls: the green curve is above the red one there and, consequently, it is below in other places (recall that on average, the simulator produces, on average, the same quantity of rainfall as the observations).
- A clear peak is visible. This corresponds to the small storms that were introduced during the “noise re-injection” phase.
- In the small hourly rainfall, there’s not much variability between different simulations. The same goes for the more extreme amounts of rain.

We now turn our attention to the hourly rainfall distribution inside each storm. The figures 6.10 and 6.11 compare the distribution of rainfall of sub periods inside storms. Results are in line with the observations but as the duration of the sub periods increase, results become very bad. Notice that on that figure, we represent the storm built with the MCMC shapes, which were specifically optimized to match the sub periods distributions. When the duration of the sub storms increase, we are confronted to the lack of data and so the fits degrade quickly. Indeed, if we look at figure 6.12 we can see that storms of duration above 10h are not frequent. However, if we look at those 10 hours storms, there are about 50 of them in the observations, so for that duration, that should be enough to fit a distribution, so the degradation of the quality may not entirely caused by that factor.

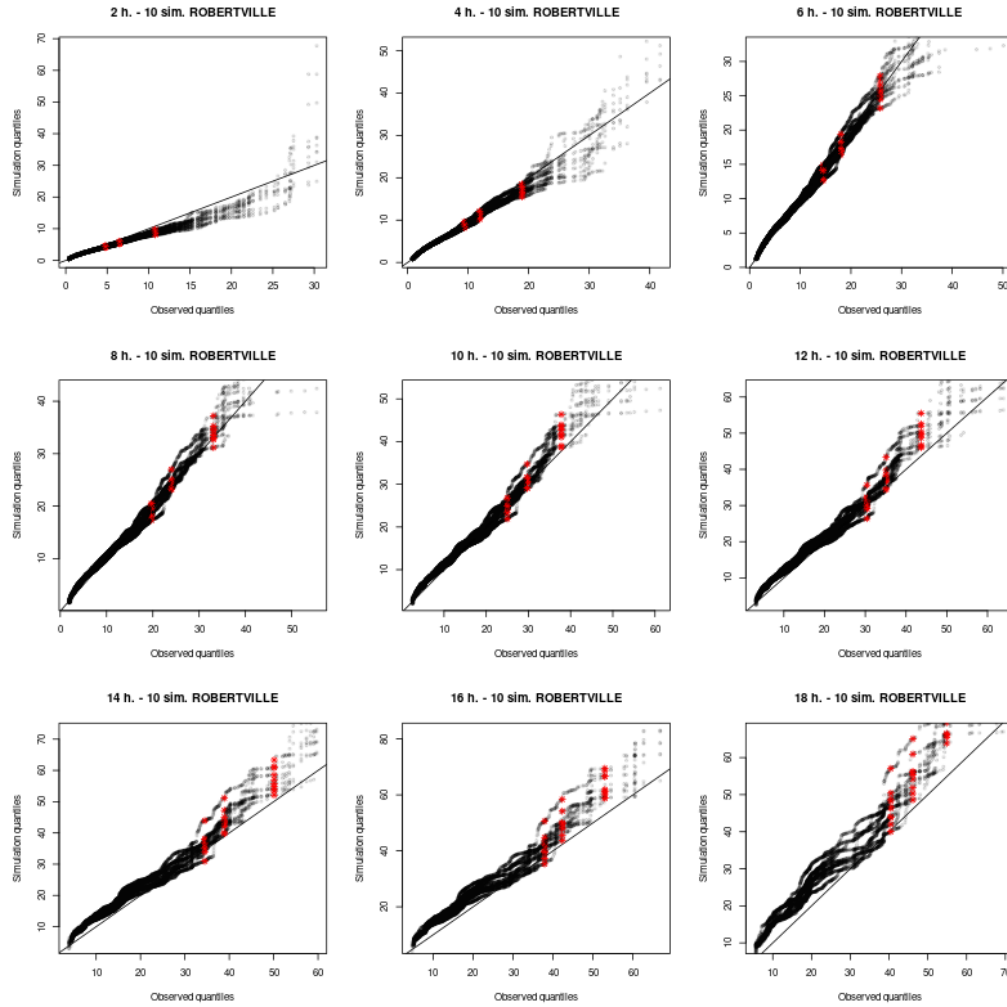


Figure 6.10: QQPlot of rainfall of sub-period (of storms) of various durations. The red dots represent the .90,.95,.99 quantiles. Data for Robertville, MCMC storm shape.

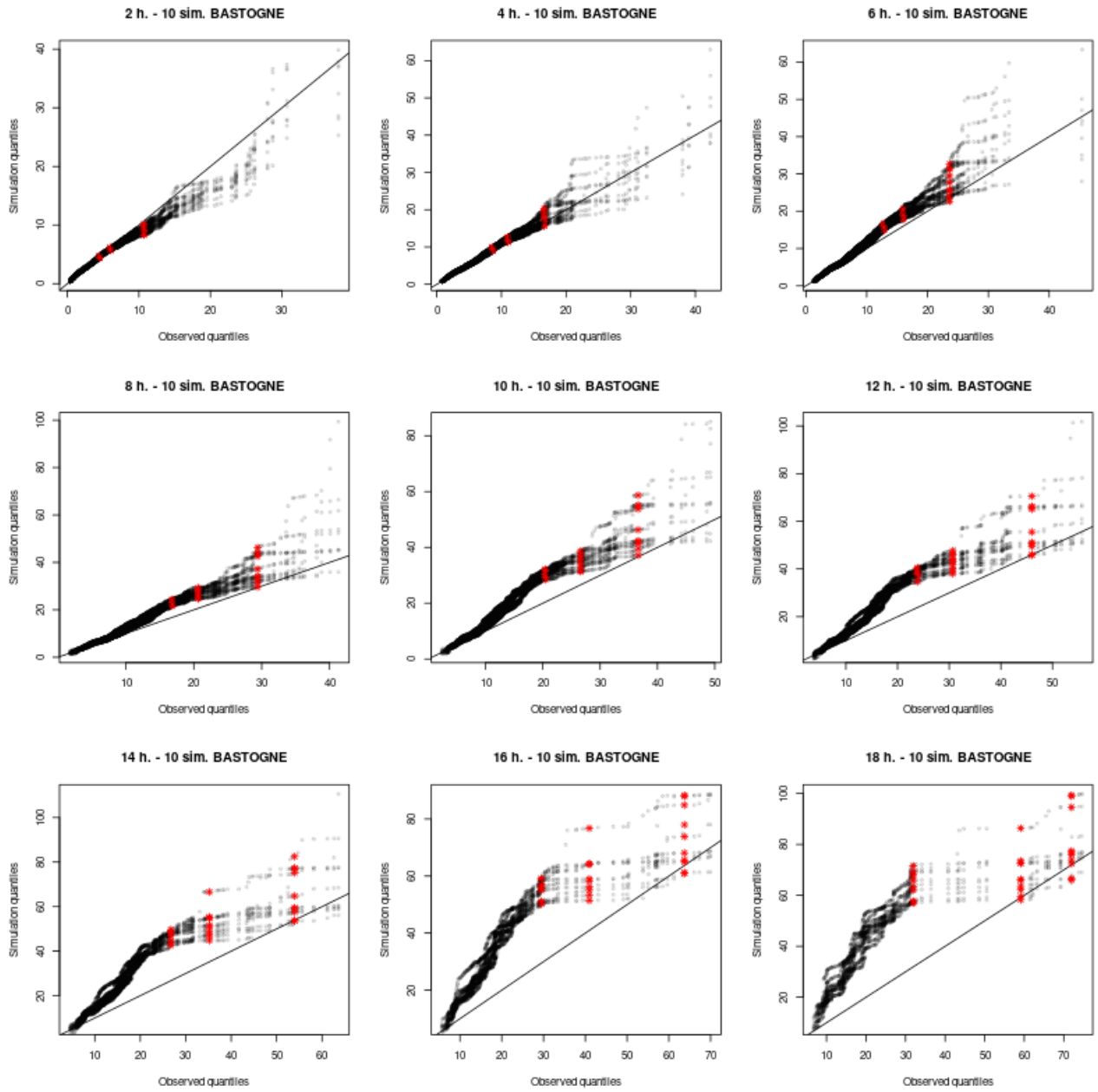


Figure 6.11: QQPlot of rainfall of sub-period of storms of various durations. The red dots represent the .90,.95,.99 quantiles. Data for Bastogne.

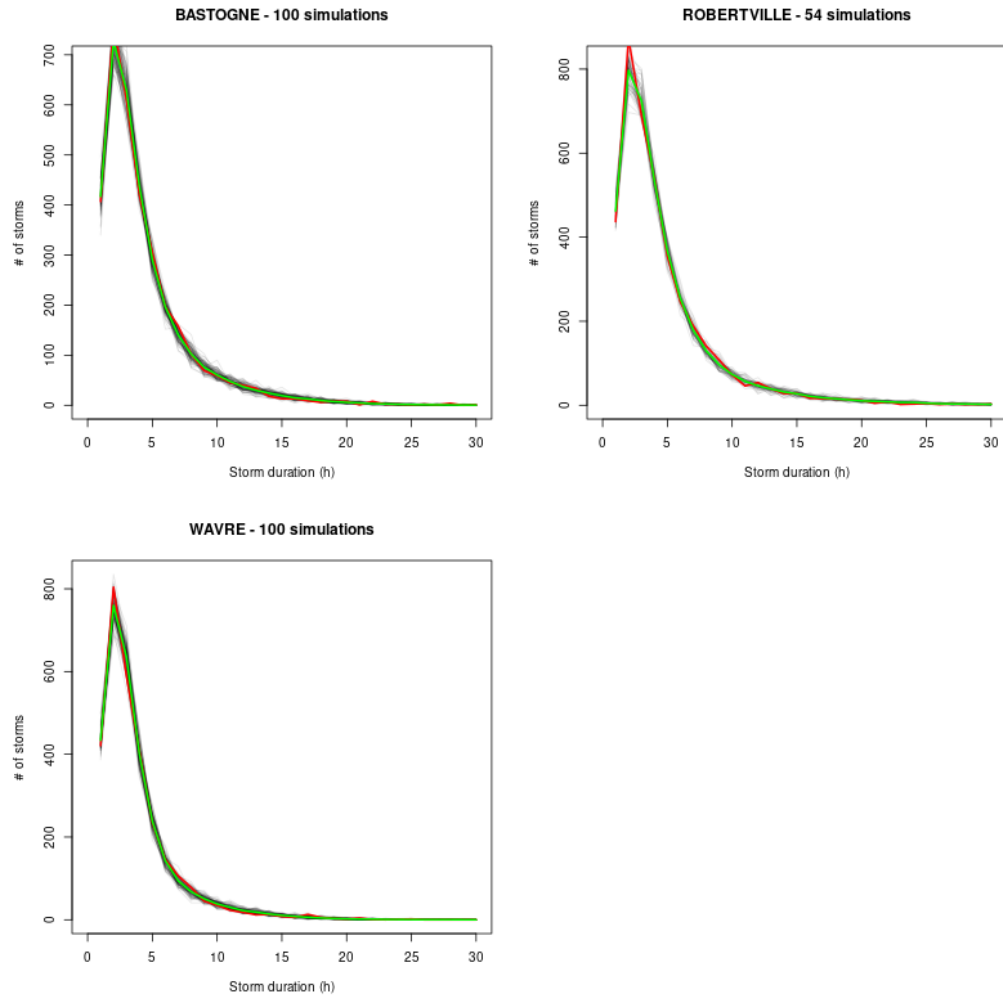


Figure 6.12: Distribution of the storm's durations. The red line is the observed values. The green line is the average over all simulations.

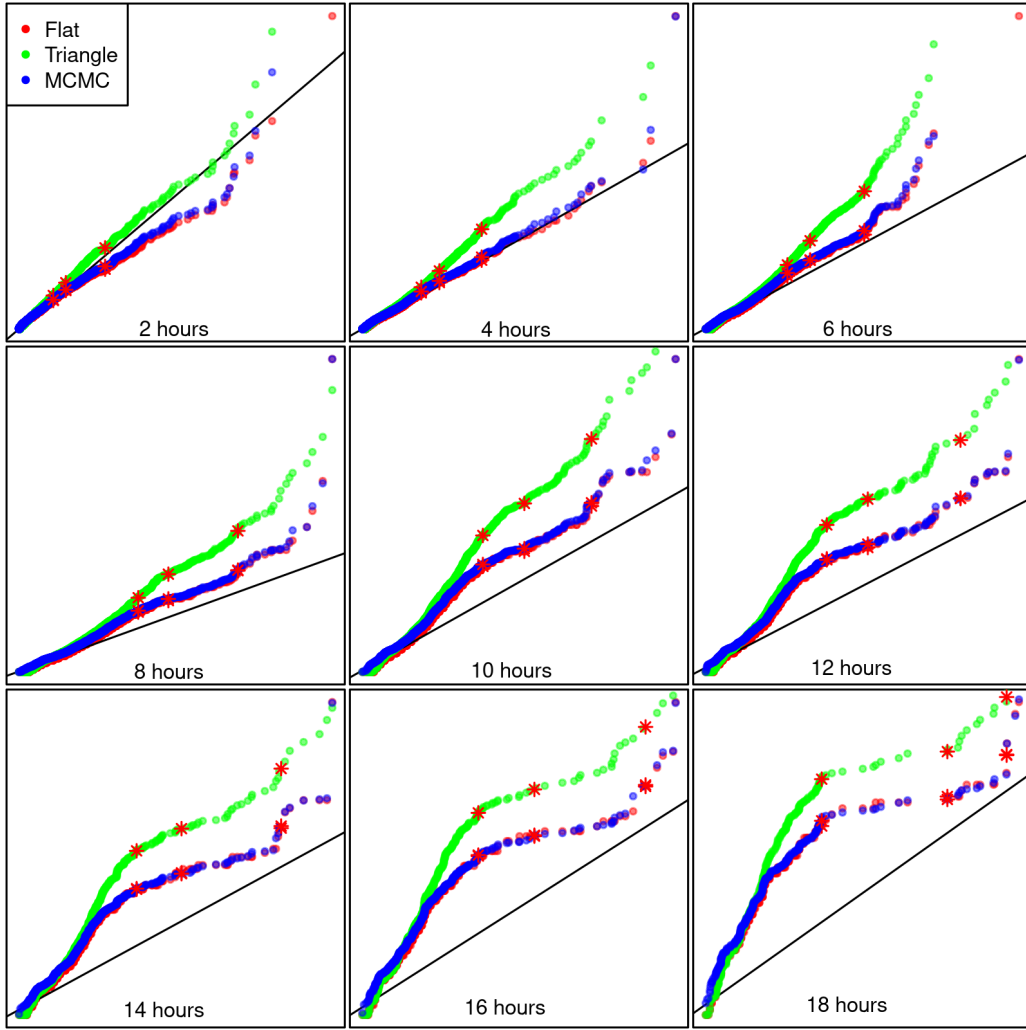


Figure 6.13: Comparing sub-storm rainfall distributions for different durations for three storm shapes, 100 simulations at Bastogne. The axis were hidden to help presentation; they are the same as on figure 6.10.

Nevertheless, prompted by the disappointing results shown on figure 6.8, we conducted another test. We basically replicated the QQPlot above but this time we averaged the simulations result. Then we plotted the results of the various storm shape together to compare them. We present the figure 6.13. As one can see the optimized (MCMC) shape doesn't perform much better than the flat one, confirming that the MCMC approach doesn't give much benefits. The triangle shape performs

6.6 IDF curves

Validation by comparison of IDF curves is useful for looking at extremes. Of course, one has to take into account the fact that the IDF curve is built on block sizes of 6 months. It means we're looking at about a maximum of 40 for a period of 20 years which means, as we have seen in section 4.2, that there can be some variability in the results.

Before comparing simulated IDF curves with reference curves, one has to go back to the storm box model. Remember that once the boxes have been defined, the "interior" hyetograph of each box is built according to some shapes. So the box works as an "envelope" for the storm size. This has some consequence over the extremes the simulation can produce.

If on one IDF curve, we look at a period of for example 6 hours. First, we note that in general, the storms are not quite frequent and so they are far apart (longer than 6 hours). Therefore, if we look at a period of six hours, the extreme can occur in these situations:

- Over a storm of exactly six hours long, we can observe an extreme and its total precipitation will be the one of the overarching box (in our case IDF curves are built with a convolution filter which sums the rainfall over all periods of six hours). So, if the box model doesn't produce an extreme, then the intra box won't provide one either. In other words, if an extreme is produced, it will have to come out of the copula (i.e. storm box model), not from the way we have built the hourly rainfall of the storm.
- Over a storm of less than six hours. We can observe an extreme for the 6 hours period because such a small storm fits entirely in the six hours period. Again, that extreme will be produced by the copula, not by the way we built the hourly rainfall of the storm. We can subsume this case and the previous one by saying: "the storm fits in the period".
- Over a storm of more than six hours. In that case, it is possible that a six hours part of it produces an extreme. This time, changing the hyetograph may or may not produce extremes; the extreme is not entirely determined by the copula (i.e. the storm box model).

So extreme precipitation periods are either produced by copula for storms of the same size or shorter or they are produced by the hourly arrangements of the longer storms. So when we read an IDF curve, we must reckon the fact that the right side of the curve is more determined by the copula itself because more storms fit in the longer periods. Starting at a period of 2 days, the right side of the IDF curve shows periods of a duration greater than most of the storms the copula will ever produce (figure 6.5 shows that 99% of the duration are below 30h). Conversely, the left side of the copula is more determined by the hourly shape of storms. Not also that the left side is where most of the storms are located as longer storms are much less frequent.

We can now examine a few IDF curves. The first three show our best results, using the triangle shape (figure 6.14). Keep in mind that the observed IDF are derived from our data set, not the one from the IRM. One can see that the general shape of the IDF curves is the same which tells us that the behavior of the simulator is acceptable. However, there are visible differences: the extremes in Wavre are off by a wide margin. We didn't have time to investigate these differences in detail.

Finally, the ultimate test consists in comparing our simulated curves with those of the IRM, see figure 6.15. Results are acceptable, except for Bastogne.

As we have said earlier, the right side of the curve is more related to the copula. On IDF curves this side has not much detail so we change the representation to precipitation/duration/frequency curves: the y-axis now gives the rainfall $d \times i$ instead of the intensity i (see figure 6.16). Again, the general shape of the curves is correctly reproduced although there are visible differences.

We have seen the IDF curves can be quite different but nevertheless we wanted to get an idea of how far each simulated IDF is from the observed IDF on a wider scale. To do that we run 10 simulations on 20 cities. We then averaged their results and built a corresponding IDF curve. Then we computed a relative error to the observed IDF. The error is computed like this $\left\| \frac{m(r,d)_{\text{simulated}} - m(r,d)_{\text{observed}}}{m(r,d)_{\text{observed}}} \right\|_1$ where $m(r,d)$ is the intensity computed for the return period r and the duration d (in other words, a point on the curve). In figure 6.17 we present the mean and standard deviation of these error computed over 20 cities.

6.7 Conclusion

We can say the following:

- The simulator produces meaningful results when we look at storm as boxes.
- It tends to produce more extremes than in observations.
- The hourly storm precipitations are not quite like the observations. The best shape is the triangular one and the more sophisticated ones don't work well.
- The "noise" storms are concentrated in some hours by design but this produces unrealistic results.

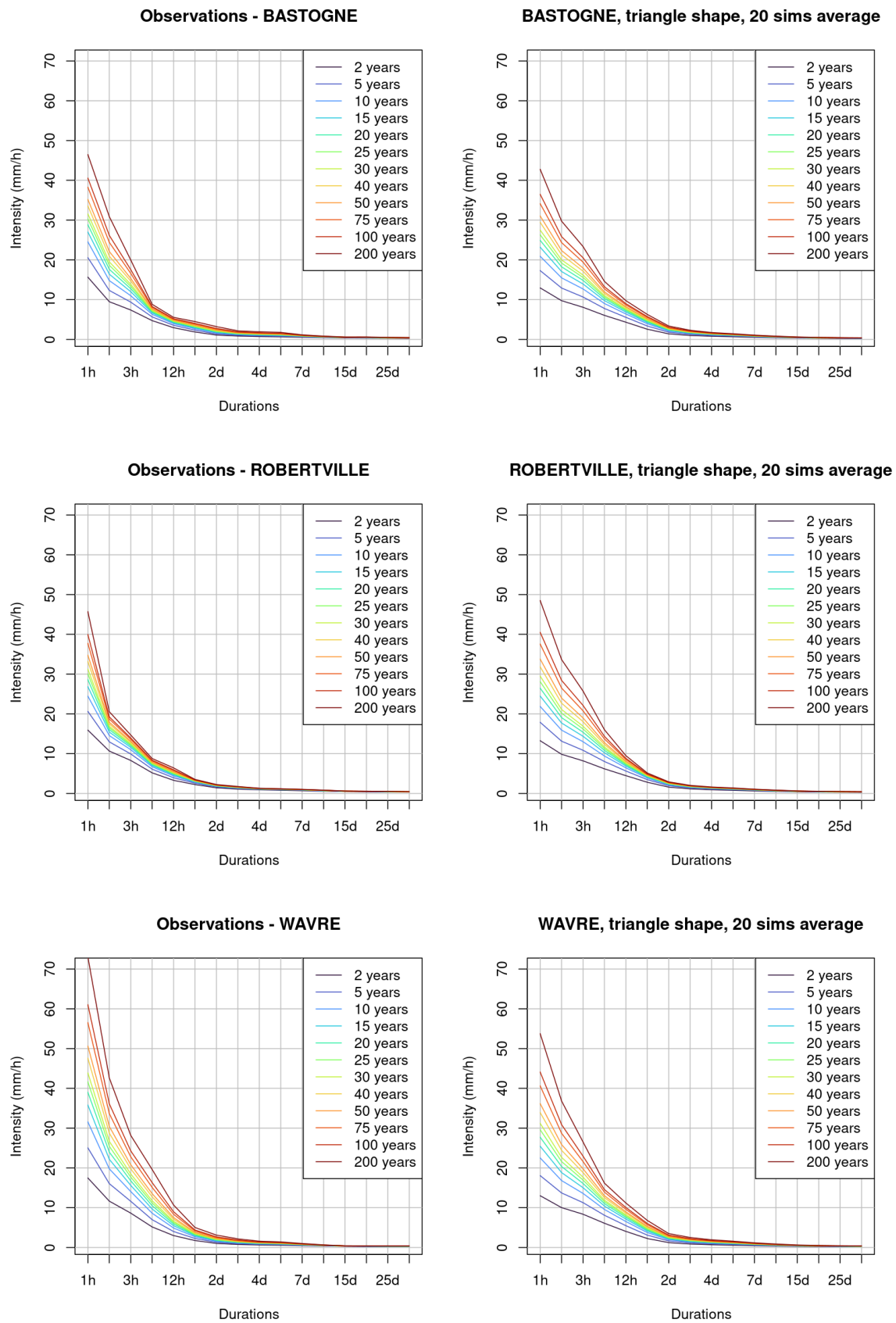


Figure 6.14: IDF curves comparison. Left: curves built on observations; right: curves built on simulated data, triangle storms.

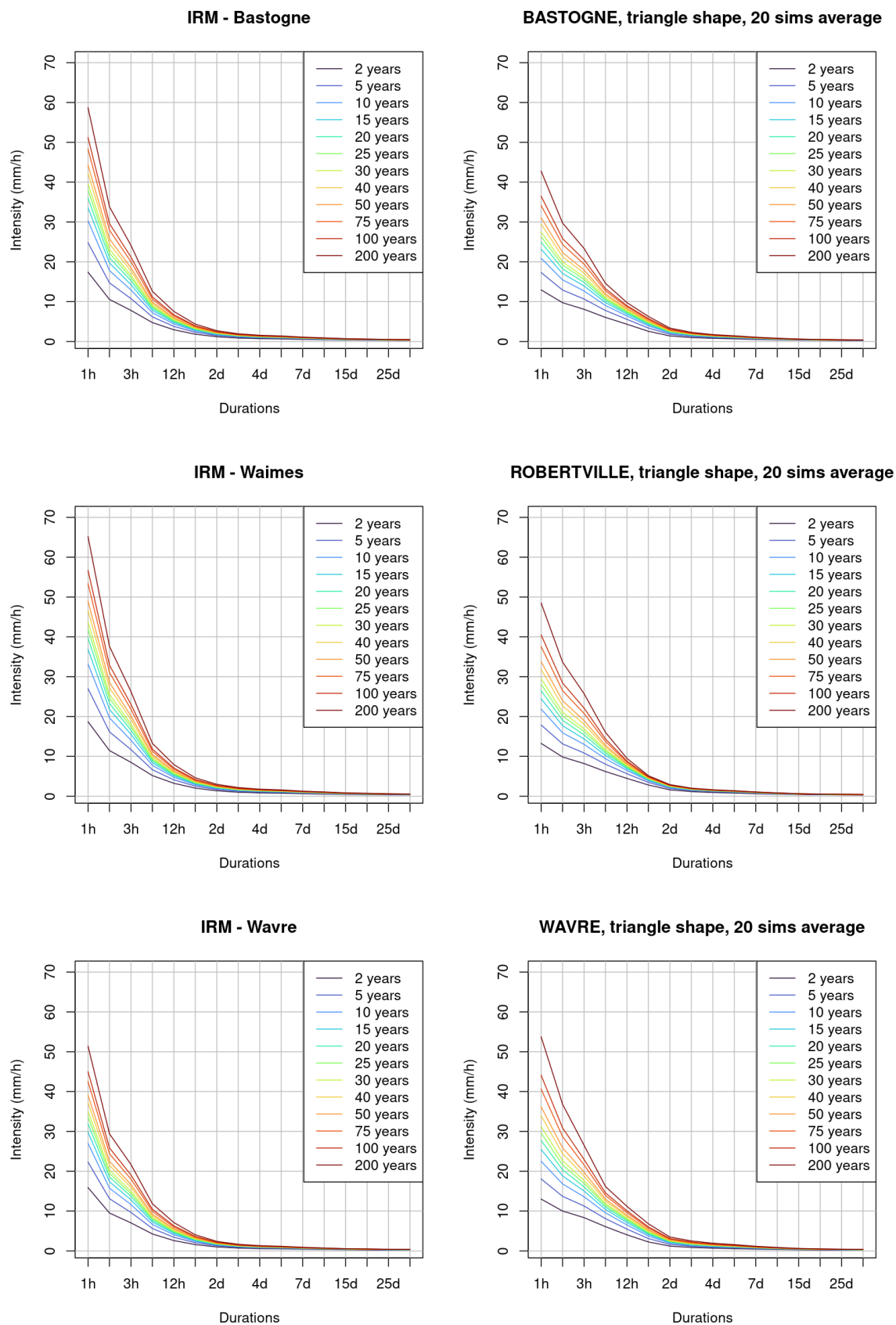


Figure 6.15: Comparing IDF curves with IRM's data. Left: curves from the IRM, right: curves from our simulator. For Robertville, we compare with IRM's data at Waimes because they don't seem to have data for Robertville (Waimes 10km away from Robertville)

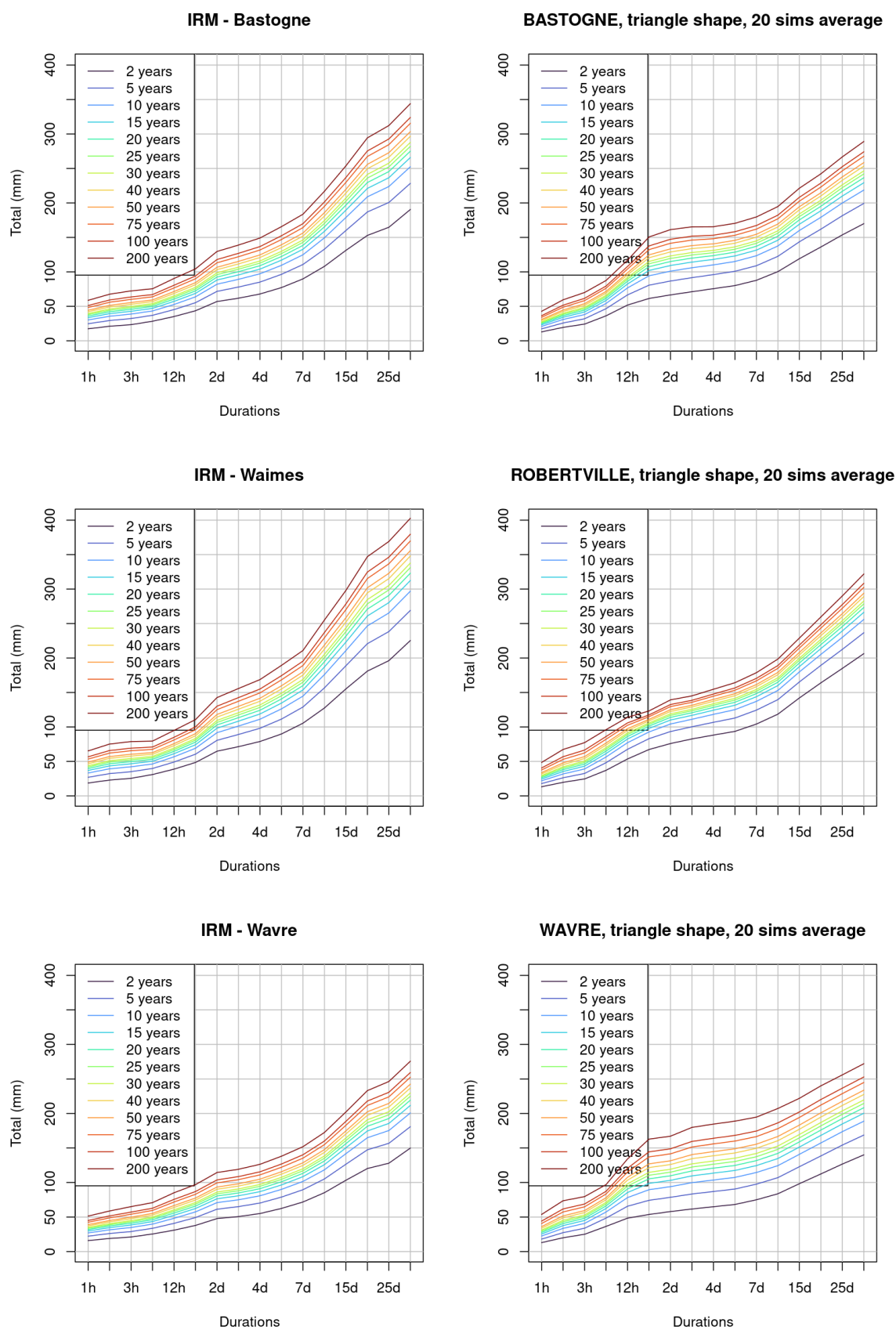


Figure 6.16: Precipitation/duration/frequency curves

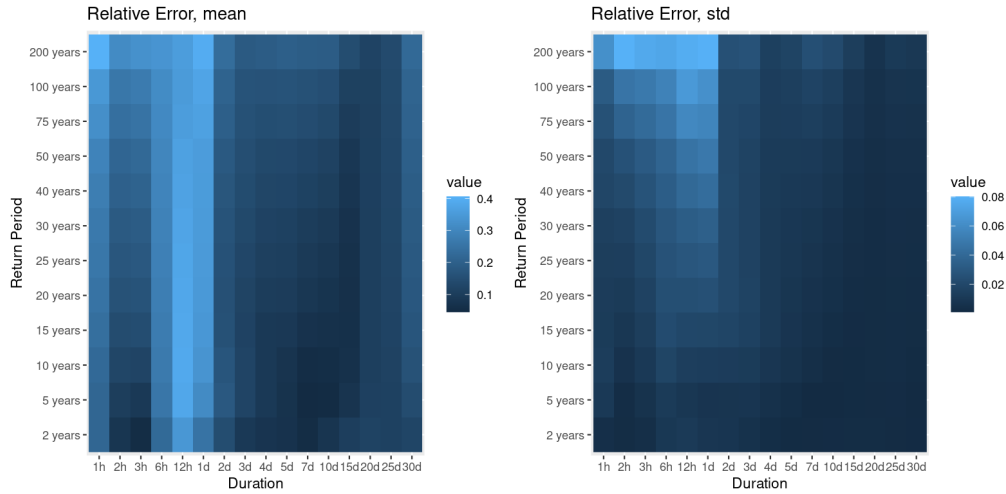


Figure 6.17: Comparing relative error in IDF curves on many stations

- The IDF curves are compatible with those of the IRM although differences exist.

So we can trust our simulator as far as the box model of storm is concerned. If we want to go to the hourly detail, then only the triangular or flat shapes are worth trying as they are simpler to understand.

We think two things could be improved:

- The way we produce extremes is not quite controlled: it is a natural derivative of the distributions at play. We have looked into modeling those more explicitly by using mixtures of standard distributions and GEV distributions, but it didn't work (usually the optimization favors exclusively the regular distribution).
- The hourly details should be investigated more by looking again at SHYPRE. Indeed we said earlier that it lacked motivation for why it builds storms the way it does. But now we have done it ourselves, we understand their choice better: modeling multi-peaks storms is the way to go.
- A wide comparison over more stations would be necessary to assess the general behaviour of the simulator.

Chapter 7

Technical information

With this work, we provide a few programs. Please be aware that although they work they have not been polished for redistribution: comments are scarce and user interface is bare minimum.

- `download.sh` is a bash script used to download all the time series from the DGO website. It pauses between each download in order to prevent submerging the website with too much traffic. So the complete download may take several *days*.
- Once the data have been downloaded, they must be preprocessed in order to be usable. The program `hydro.py` is made for that. IT is used once to parse the downloaded HTML and turn that into numpy array. It is invoked again to export the numpy array to *.csv files to be digested by R. `hydro.py` must know the path to the directory where the downloaded files are located, see `DATAPATH` variable defined inside it.
- The MCMC shape optimizer is written in R. It must be compiled with the regular Cargo tools. Then it must be wired to the simulator (see variable `RUST_OPTIM` in `raingen.R`).
- `simulation.R` is the simulator.
- Finally lots of charts drawing code is in `hydro.py` (see `--help`) and `report.R` (see code).

A full execution would be:

```
# These need to be run only once:
```

```
./download.sh
python hydro.py --import-data
python hydro.py --export-storms
```

```
# Run as many times as needed:
```

```
R --file=simulation.R --no-save --no-echo --quiet \
--args --shapes triangle --storms-only FALSE \
--station-name BASTOGNE --nb-sims 10
```

Should you have any question, please write to schampailler@skynet.be.

Chapter 8

Conclusion

In conclusion of this work, we would like to stress the lessons we have learned throughout its execution:

- As far as rain simulation is concerned, we're confident that:
 - Working with probability distributions is an endless task because the best fitting distributions will be very dependent on the kind of rain one is modeling. So as soon as the data sets change, one will probably need to change the type of PDFs.
 - Modeling rain is hard because there's not much physics behind it. So one ends up modeling data more than an actual phenomenon. We now understand why in the literature about rain simulation we've read, authors usually don't explain how they choose probability distributions nor how they fit them.
 - The consequence of these two points is that we think it would be beneficial to build a meta model that would be trained over several data sets coming from different regions (provided they have some common ground for their units).
- For data sciences:
 - Doing data sciences requires a lot of attention. That's very different from our past experience in software development where bugs manifest themselves usually in under 15 minutes. In data sciences, it is very hard to test for bugs because that would require a model to formalize one's expectations about what one is ... modeling!
 - Fitting probability distributions is a difficult exercise in face of real data (discrete, extremes, exotic). Fitting more complex distributions is even more difficult. We'd say it is an art that requires some practice and domain knowledge. However, it is quite impressive to which extent thinking about a fit gone wrong or about a statistical test's unexpected result helps to understand what the data mean and to clear misconceptions one may have.
 - Although we didn't use many of the tools learned during the courses, we think that the investment we put in the few theoretical concepts at play here lead us to a much deeper understanding of their strengths and limits.
- During the course of our work, it was pretty difficult to choose between R and python. We have a long experience with python but none in data sciences. We started with python but slowly moved to R. Here's why:
 - R repl (with emacs or RStudio) is the most productive way to work we have found compared to python (including Jupyter notebooks).
 - The libraries in R, although hard to choose because they come by dozens, usually propose better defaults than their equivalent in Python.

8.1 Future tasks

When writing the simulator we were mainly driven by one idea: get a working prototype. Now, in hindsight, we think a few things should be done:

- Derive statistics across various rain gauges (we mostly worked on 3 of them)
- Investigate how to represent PDF's with extreme value better.
- Validate that the copula choice is the best one.
- Use more statistics testing to validate our hypothesis on the data and choice of models.
- Test if one can split the year in more than two seasons.
- Use other data sources to include climate change.

Bibliography

- Arnaud P. Lavabre J., Masson J. (1999). “Amélioration des performances d’un modèle stochastique de génération de hyétogrammes horaires: application au pourtour méditerranéen français”. In: *Revue des sciences de l’eau / Journal of Water Science* 12.2, pp. 251–271. DOI: <https://doi.org/10.7202/705351ar>.
- Arnaud P. ; Lavabre, J. (2000). “La modélisation stochastique des pluies horaires et leur transformation en débits pour la prédétermination des crues”. In: *Revue des sciences de l’eau / Journal of Water Science* 13.4, pp. 441–462. DOI: <https://doi.org/10.7202/705402ar>.
- Azzalini, A. and A. Dalla Valle (1996). “The Multivariate Skew-Normal Distribution”. In: *Biometrika* 83.4, pp. 715–726. ISSN: 00063444. URL: <http://www.jstor.org/stable/2337278> (visited on 06/30/2022).
- Bendel, Dan (2020). *Metropolis-Hastings: A Comprehensive Overview and Proof*. <https://similarweb.engineering/mcmc/>. Accessed: 2022-10-5.
- Cernesson, Flavie, Jacques Lavabre, and Jean-Marie Masson (Jan. 1996). “Stochastic model for generating hourly hyetographs”. In: *Atmospheric Research* 42.1, pp. 149–161. DOI: 10.1016/0169-8095(95)00060-7.
- Cowpertwait, Paul S. P. (1991). “Further developments of the neyman-scott clustered point process for modeling rainfall”. In: *Water Resources Research* 27.7, pp. 1431–1438. DOI: <https://doi.org/10.1029/91WR00479>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/91WR00479>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/91WR00479>.
- De Luca, Davide Luciano and Andrea Petroselli (2021). “STORAGE (STOchastic RAINfall GEnerator): A User-Friendly Software for Generating Long and High-Resolution Rainfall Time Series”. In: *Hydrology* 8.2. ISSN: 2306-5338. DOI: 10.3390/hydrology8020076. URL: <https://www.mdpi.com/2306-5338/8/2/76>.
- Furrer, Eva M. and Richard W. Katz (2008). “Improving the simulation of extreme precipitation events by stochastic weather generators”. In: *Water Resources Research* 44.12. DOI: <https://doi.org/10.1029/2008WR007316>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2008WR007316>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008WR007316>.
- Hauser, Kris (2013). *Why does the Metropolis-Hastings procedure satisfy the detailed balance criterion?* <https://people.duke.edu/~kh269/teaching/notes/MetropolisExplanation.pdf>. Accessed: 2022-1-5.
- Lang, M. and J. Lavabre (Jan. 2007). *Estimation de la crue centennale pour les plans de prévention des risques d’inondations*.
- M., Journée (Mar. 2017). “Caractérisation des précipitations extrêmes”.
- Nelsen, Roger B. (2006). *An introduction to copulas*. New York: Springer. ISBN: 0387286594 9780387286594.
- Robert, Christian (Apr. 2015). “The Metropolis—Hastings Algorithm”. In: DOI: 10.1007/978-1-4757-4145-2_7.
- Schmitz, Volker (2003). “Copulas and Stochastic Processes”. PhD thesis. Rheinisch-Westfälischen Technischen Hochschule Aachen.
- Singer, M. B., K. Michaelides, and D. E. J. Hobley (2018). “STORM 1.0: a simple, flexible, and parsimonious stochastic rainfall generator for simulating climate and climate change”. In: *Geoscientific Model Development* 11.9, pp. 3713–3726. DOI: 10.5194/gmd-11-3713-2018. URL: <https://gmd.copernicus.org/articles/11/3713/2018/>.

- Vyver, H. Van de (2012). "Spatial regression models for extreme precipitation in Belgium". In: *Water Resources Research* 48.9. DOI: <https://doi.org/10.1029/2011WR011707>. eprint: <https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2011WR011707>. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011WR011707>.
- Willems, Patrick (2001). "A spatial rainfall generator for small spatial scales". In: *Journal of Hydrology* 252.1, pp. 126–144. ISSN: 0022-1694. DOI: [https://doi.org/10.1016/S0022-1694\(01\)00446-2](https://doi.org/10.1016/S0022-1694(01)00446-2). URL: <https://www.sciencedirect.com/science/article/pii/S0022169401004462>.