Project 1

Downscaling: Comparison of Quantile-Quantile Mapping and Lab 06 Methodology

Catherine Jackson (ccj3)

2023-11-13

Table of contents

# 1. Intro

## 1.1 Statistical and Dynamical Downscaling

Global Climate Models, also referred to as GCMs, are incredibly valuable tools in that they can capture large scale climate dynamics (e.g. future greenhouse gas concentration over the entire Earth). Furthermore, many of these phenomena must be studied at a global scale. However, there are limitations to GCMs. Primarily, given the large scale analysis, grid cells and computation points for these models are actually quite large. This leads to problems when trying to study local phenomena, such as the climate of a specific city or region. This is when downscaling must be used. There are two types of downscaling: Dynamical and Statistical.

### 1.1.1 Dynamical Downscaling

This type of downscaling uses a Regional Climate Model (RCM) which takes in the output of a GCM as a boundary condition. Then, RCM use these boundary conditions and physical principles to generate higher resolution local climate. However, these are computationally expensive.

### 1.1.2 Statistical Downscaling

Statistical Downscaling is another common method dependent on the availability of local weather data, such as precipitation. A statistical relationship is developed between the historic observed climate data and the output of the climate model for the same historical period. Then, this statistical relationship is used to downscale the climate model output to the local scale. This can be combined with bias correction and adjustment to improve the accuracy of the downscaled data.

For this project, I will be using statistical downscaling. I will use a Julia package, RainFARM, to downscale coarse precipitation data to a finer scale.

### 1.1.3 Statistical Downscaling Methods

Statistical Downscaling, SD, consists of a identified historical period (calibration) used to analyze the predictor/predictand relationship. This relationship is then applied to predictors from the GCMs to get the local climate projections.

|  |
| --- |
| Note |
| Sources include the Copernicus Climate Change Service (C3S), National Oceanic and Atmospheric Administration (NOAA), Evaluaation of statistical downscaling methods for climate change projections over Spain: Present conditions with perfect predictors (2021). |

# 2. Project

For this project, I will be conducting a slightly different analysis than others in the class. As I proposed to Dr. Doss-Gollin, I want to undertake a deep dive into a Julia based Downscaling Package, RainFARM, analyze the methods and algorithms used, and apply it to the larger scale precipitation data.

# 3. RainFARM

According to Rebora et al. (2006b), RainFARM aims to generate “an ensemble of stochastic realizations of the small-scale precipitation fields that have statistical properties similar to those measured for rainfall in a given area and/or synoptic situation.” In other words, RainFARM takes course data and downscales it to a finer scale while still keeping similar statistical properties (e.g. it presents a faithful representataion of what your data could look like on the finer scale). However, it is important to note that it is a stochastic process, and randomness is involved. Consequently, it is not a deterministic process, and the results will vary each time the code is run. This is why it is important to run the code multiple times to see many different possible downscaled outcomes.

## 3.1 Common Rainfall Downscaling Methods Overview

There are many different methods used for downscaling, but there are three main types used in the literature, according to the Rebora article. These are: 1. Point Processes 2. Autoregressive Processes with Nonlinear Transformation: This involves using patterns from the past (autoregressive) and applying it to a static nonlinear transformation. 3. Fractal Cascades: This creates a model where rain repeats in patterns inspired by fractals. However, there have been examples of people combining these methods to create hybrid models.

## 3.2 Methodology Outline

The RainFARM methodology uses a two step process: 1. Linear Autoregressive Process: The behavior of the rainfall on the course scale can be analyzed for patterns over this large area. A simple forecast is made from this step. 2. Nonlinear Filtering: This is applied to the baseline prediction, and it adds detail and makes adjustments to increase accuracy. This methodology works well with overall weather patterns.

## 3.3 Inputs, Outputs, and Variables

### 3.3.1 Inputs

RainFARM can input a course rainfall field P(X, Y, T). This type of data can come from a few different sources, like GCMs, but it lacks the resolution at a fine scale needed for local analysis and regional studies. X and Y represent the latitude and longitude of the grid on which data is stored, and T indicates time.

The second two equations above define the range of scales for L and T. These ranges refer to the scale of the data. L0 and T0 are defined at the scales on which the data forecast becomes poor. Typically, the spatial forecase becomes poor around 10-20 km, and the temporal forecast becomes poor around 2-4 hours (for the types of data on which RainFARM is used).

### 3.3.2 Outputs

The goal of the RainFARM package is to output a new precipitation field, written as r(x,y,t). The new scale, lambda, will be less than the original scale, L, and typically falls around 1-2 km. The new time scale, tau, will be less than the original time scale, T, and typically falls around 10 minutes.

This output must be thought of as one potential realization of the small scale behavior of this larger system and phenomena.

### 3.3.3 Aggregation and Validation

Interestingly, RainFARM includes its own self validation in which the output, the fine scale precipitation field, r(x,y,t), is aggregated on the original spatial and temporal scales. This aggregated field should produce a coarse scale precipitation field, R(X, Y, T) that must be equal to the original field P.

## 3.4 Detailed Methodology

### 3.4.1 Fourier Analysis & Power Spectrum

The Rebora paper first describes the estimation of the space-time power spectrum of the field P. This space-time power spectrum can be thought of as the power distribution of the field P. It helps describe how precipitation is generated across space and time, and is frequency used in meteorology.

To better understand this, we can go into more detail on both Fourier analysis and Power Spectra.

#### 3.4.1.1 Fourier Analysis

Fourier Analysis was first developed by French mathematician Joseph Fourier. It is a mathematical tool that can break down complex signals into a sum of simpler sinusoidal (wave-like) functions. Consider, for example, a recording of a song. Fourier Analysis could isolate the individual notes that make up the song by analyzing the frequency components.

Fourier analysis itself is complex, but it can be considered a method to break down complex components into simpler ones. Therefore, we can analyze the complex patterns within the precipitation field and break them down into frequency components we can analyze easily.

It is also important to note that these frequency components have an associated power which tells us how impactful they are in the system. This is where the power spectrum comes in.

#### 3.4.1.2 Power Spectra

The power spectrum Sxx(f) of a time series x(t) describes the distribution of power into frequency components composing that signal. The statistical average of any sort of signal as analyzed in terms of its frequency content is its spectrum. This is frequently done as analysis over a frequency-domain representation is easier to analyze than a time-domain representation.

Below, I go through the steps of the RainFARM methodology set up in the Rebora paper. First, the space-time power spectrum is defined, and Ky and Kx represent wave numbers and Omega represents angular frequency. The Nyquist Number is also defined, and it is the maximum frequency that can be represented given the sampling frequency. This is important because it defines the limitations of the analysis.

$$
|\hat{P}(K\_{X}, K\_{Y},\Omega|^2 = space \space time \space power \space spectrum
$$

$$
K\_{X} \space and \space K\_{Y} = wavenumbers \space in \space X \space and \space Y \space directions
$$

$$
\Omega = angular \space frequency
$$

$$
(K\_{X}, K\_{Y}) <= pi/L\_{0} = Nyquist \space Number
$$

$$
\Omega <= pi/T\_{0} = Nyquest \space Frequency
$$

So we have an equation for the space time power spectrum above. In addition, we have representations of our wavenumbers where the wavenumbers are the components of a wavevector along the x and y directions in the case of a two-dimensional setting. Wavenumbers can be thought of as the spatial frequency of a wave. In other words, the wavenumbers allow for a representation of the spatial frequency of these energy waves. K(X) and K(Y) represent how these precipitation patterns vary in the X and Y directions. However, the Nyquist Number defines the limitations of our understanding, but an in-depth discussion of this determination is outside the scope of this work.

The angular frequency, in the general theory, measures how fast the waves oscillate in time. In other words, it represents how fast the pattern changes, so for RainFARM it represents how quickly the precipitation patterns change. Again, the Nyquist Frequency defines the limitations of the analysis.

### 3.4.2 Downscale Power Spectrum

Next, the power spectrum is extrapolated to small scales. In this way, the downscaling occurs in the frequency domain not the spatial domain. This is done through the assumption of a power-law behavior.

#### 3.4.2.1 Power-Law Behavior

According to a helpful Wikipedia, a power-law behavior is a “functional relationship between two quantities where a relative change in one quantity results in a relative change in the other quantity proportional to a power of the change, independent of the initial size of those quantities: one quantity varies as a power of another.” This may appear to be a large assumption, but certain studies (e.g. Ferraris et al. 2003a, b) have shown that a power-law behavior is a reasonable assumption for precipitation fields.

#### 3.4.2.2 Slopes

Once we have a downscaled precipitation field (using the assumption that there is aa functional relationship between the coarse precipitation field and the fine field in the frequency domain), the spatial and temporal logarithmic slopes of the power spectrum, which we call α and β, are defined. These slopes express the rate at which the power of the frequency components changes with respect to space and time. For example, a high spatial logarithmic slope means that the power dynamics can chanage quickly as you move across the field. In addition, isotropy is assumed in the two spatial directions. This is also a valid assumption with precipitation. This uniform behavior in all directions lessens the computational load and simplifies the model.

### 3.4.3 Fourier Spectrum Inversion

We have been conducting analysis in the frequency domain, but another useful product of Fourier Analysis is that this analysis can be inverted to change back into the space time domain. This can be broken up into the definition of the Fourier Spectrum, the Power Spectrum, and the Inversion.

A Fourier spectrum is defined below. This Fourier analysis lends itself to an exploration of the frequency components of the data and a consequent modeling of the distribution of power across frequencies.

A Power spectrum is defined below. This functional equation allows us to describe the power spectrum in a mathematical form.

The next step is inversion. The power spectrum can be inverted to generate a Gaussian field that represents the data in space time. A Gaussian field is a random field involving Gaussian probability density functions of the variables. Gaussian fields are often used to model random processes that vary smoothly in space or time.

### 3.4.4 Generate a Precipitation Field

Finally, a nonlinear transformation, defined below, is applied to all points in the Gaussian field g. This produces a synthetic precipitation field r(x, y, t).

Unfortunately, no documentation online gives the rational for selecting this method of nonlinear transformation. However, it is a relatively simple nonlinear transformation which minimizes computational load.

### 3.4.5 Summary and Validation

At this point in the process, (1) a coarse data field has been taken in, (2) a power spectrum has been generated to represent the energy/precipitation in the atmosphere, (3) the power spectrum is extrapolated to smaller scales with the assumption of power-law behavior, (4) fourier spectrum are used to create a function that represents this downscaled power spectrum and then inverted to create a gaussian field that brings the downscaled data back to a space time field. Finally, a precipitation field is generated by taking “a nonlinear transformation of the Gaussian field g” (Rebora et al., 2006b).

As mentioned earlier, there is an internal validation step in which the generated (or synthetic) precipitation field is compared to the original field. This is done by aggregating the synthetic field to the original scale; in other words, the small scale field is aggregated to L0 and T0.

In each box, the ratio of the original field, P, and the synthetic field, R, is calculated. If the ratio is 1, there is a perfect match. If it is not one, the synthetic field R can be adjusted. This means that the synthetic field closely resembles the original field at the L0 and T0 scale.

## 3.5 Important Notes

As discussed, the RainFARM methodology is stochastic, and this random behavior is associated with the Fourier phases. The phases are randomly generated, and this randomness is what creates the different realizations of the small scale precipitation field. This is why it is important to run the code multiple times to see many different possible downscaled outcomes. The original paper introducing this method writes: “The RainFARM approach is easily used for ensemble predictions; by choosing different sets of random Fourier phases, one can generate a large number of stochastic fields that are all equal to P when aggregated on space and time scales larger than L0 and To, and are different, but with similar statistical properties, on smaller scales.”

# 4. Packages

[ Info: Precompiling AdaptStaticArraysExt [e1699a77-9e31-5da8-bb3e-0a796f95f0a0]  
ERROR: LoadError: ArgumentError: Package AdaptStaticArraysCoreExt does not have Adapt in its dependencies:  
- You may have a partially installed environment. Try `Pkg.instantiate()`  
 to ensure all packages in the environment are installed.  
- Or, if you have AdaptStaticArraysCoreExt checked out for development and have  
 added Adapt as a dependency but haven't updated your primary  
 environment's manifest file, try `Pkg.resolve()`.  
- Otherwise you may need to report an issue with AdaptStaticArraysCoreExt  
Stacktrace:  
 [1] macro expansion  
 @ ./loading.jl:1634 [inlined]  
 [2] macro expansion  
 @ ./lock.jl:267 [inlined]  
 [3] require(into::Module, mod::Symbol)  
 @ Base ./loading.jl:1611  
 [4] include  
 @ ./Base.jl:457 [inlined]  
 [5] include\_package\_for\_output(pkg::Base.PkgId, input::String, depot\_path::Vector{String}, dl\_load\_path::Vector{String}, load\_path::Vector{String}, concrete\_deps::Vector{Pair{Base.PkgId, UInt128}}, source::Nothing)  
 @ Base ./loading.jl:2049  
 [6] top-level scope  
 @ stdin:3  
in expression starting at /Users/catherinejackson/.julia/packages/Adapt/Ur60R/ext/AdaptStaticArraysExt.jl:1  
in expression starting at stdin:3  
┌ Error: Error during loading of extension AdaptStaticArraysExt of Adapt, use `Base.retry\_load\_extensions()` to retry.  
│ exception =  
│ 1-element ExceptionStack:  
│ Failed to precompile AdaptStaticArraysExt [e1699a77-9e31-5da8-bb3e-0a796f95f0a0] to "/Users/catherinejackson/.julia/compiled/v1.9/AdaptStaticArraysExt/jl\_6bIqVS".  
│ Stacktrace:  
│ [1] error(s::String)  
│ @ Base ./error.jl:35  
│ [2] compilecache(pkg::Base.PkgId, path::String, internal\_stderr::IO, internal\_stdout::IO, keep\_loaded\_modules::Bool)  
│ @ Base ./loading.jl:2300  
│ [3] compilecache  
│ @ ./loading.jl:2167 [inlined]  
│ [4] \_require(pkg::Base.PkgId, env::Nothing)  
│ @ Base ./loading.jl:1805  
│ [5] \_require\_prelocked(uuidkey::Base.PkgId, env::Nothing)  
│ @ Base ./loading.jl:1660  
│ [6] \_require\_prelocked(uuidkey::Base.PkgId)  
│ @ Base ./loading.jl:1658  
│ [7] run\_extension\_callbacks(extid::Base.ExtensionId)  
│ @ Base ./loading.jl:1255  
│ [8] run\_extension\_callbacks(pkgid::Base.PkgId)  
│ @ Base ./loading.jl:1290  
│ [9] run\_package\_callbacks(modkey::Base.PkgId)  
│ @ Base ./loading.jl:1124  
│ [10] \_tryrequire\_from\_serialized(modkey::Base.PkgId, path::String, ocachepath::String, sourcepath::String, depmods::Vector{Any})  
│ @ Base ./loading.jl:1398  
│ [11] \_require\_search\_from\_serialized(pkg::Base.PkgId, sourcepath::String, build\_id::UInt128)  
│ @ Base ./loading.jl:1494  
│ [12] \_require(pkg::Base.PkgId, env::String)  
│ @ Base ./loading.jl:1783  
│ [13] \_require\_prelocked(uuidkey::Base.PkgId, env::String)  
│ @ Base ./loading.jl:1660  
│ [14] macro expansion  
│ @ ./loading.jl:1648 [inlined]  
│ [15] macro expansion  
│ @ ./lock.jl:267 [inlined]  
│ [16] require(into::Module, mod::Symbol)  
│ @ Base ./loading.jl:1611  
│ [17] eval  
│ @ ./boot.jl:370 [inlined]  
│ [18] include\_string(mapexpr::typeof(REPL.softscope), mod::Module, code::String, filename::String)  
│ @ Base ./loading.jl:1903  
│ [19] softscope\_include\_string(m::Module, code::String, filename::String)  
│ @ SoftGlobalScope ~/.julia/packages/SoftGlobalScope/u4UzH/src/SoftGlobalScope.jl:65  
│ [20] execute\_request(socket::ZMQ.Socket, msg::IJulia.Msg)  
│ @ IJulia ~/.julia/packages/IJulia/Vo51o/src/execute\_request.jl:67  
│ [21] #invokelatest#2  
│ @ ./essentials.jl:819 [inlined]  
│ [22] invokelatest  
│ @ ./essentials.jl:816 [inlined]  
│ [23] eventloop(socket::ZMQ.Socket)  
│ @ IJulia ~/.julia/packages/IJulia/Vo51o/src/eventloop.jl:8  
│ [24] (::IJulia.var"#15#18")()  
│ @ IJulia ./task.jl:514  
└ @ Base loading.jl:1261

# 5. Downloading Data

24×24×16365 Array{Real, 3}:  
[:, :, 1] =  
 1.37582 1.57975 1.45262 … 0.0 0.0202463 0.0  
 1.20963 1.85082 2.28766 0.0308067 0.0602374 0.114836  
 1.97182 1.45203 2.09293 0.0490085 0.121709 0.237292  
 1.5909 1.33469 1.65327 0.0664271 0.389667 0.678864  
 1.43704 1.75647 2.29071 0.0964478 0.24362 1.0564  
 1.23506 1.71966 1.24682 … 0.29584 0.192309 0.594209  
 1.34688 2.06668 1.14913 0.50683 0.345075 0.153464  
 1.45706 2.27098 0.724487 1.06509 0.576138 0.0801041  
 5.20069 3.71962 0.181735 1.43136 0.706655 0.133022  
 2.88819 0.759342 0.163341 2.08092 1.4474 0.53837  
 1.50673 0.461275 0.207048 … 0 0 0  
 1.43436 0.139224 0.290842 0 0 0  
 3.18961 2.13086 2.59861 0 0 0  
 5.19326 5.16296 4.65167 0 0 0  
 5.3778 4.47193 6.19447 0 0 0  
 2.81836 8.9879 5.9169 … 0 0 0  
 1.80613 8.12168 2.24706 0 0 0  
 6.17247 15.1932 14.4421 0 0 0  
 15.3935 12.0708 20.1535 0 0 0  
 16.41 16.3307 23.8688 0 0 0  
 12.77 20.6869 18.5016 … 0 0 0  
 14.9623 9.67723 7.38247 0 0 0  
 21.7209 10.5213 9.88938 0 0 0  
 20.7196 14.5443 15.6206 0 0 0  
  
[:, :, 2] =  
 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.136311 0.0507107 0.0 0.0  
 0.0 0.0 0.0 0.0 0.353149 0.170228 0.0 0.0  
 0.0344644 0.0828211 0.0 0.0 0.079515 0.0242824 0.0 0.0  
 0.116297 0.232722 0.0 0.0 … 0 0 0 0  
 0.0780643 0.120667 0.0 0.0 0 0 0 0  
 0.790819 0.209601 0.0810387 0.0836833 0 0 0 0  
 0.330909 0.0694368 0.202908 0.172532 0 0 0 0  
 0.0834752 0.366478 0.193584 0.041639 0 0 0 0  
 0.334877 0.392715 0.221516 0.0368422 … 0 0 0 0  
 0.171142 0.847358 0.0837831 0.0 0 0 0 0  
 0.660464 2.01706 0.12625 0.0 0 0 0 0  
 0.309305 0.0865614 0.0 0.0546165 0 0 0 0  
 0.206135 0.0 0.0 0.0464391 0 0 0 0  
 0.397575 0.355726 0.0 0.0436917 … 0 0 0 0  
 1.2978 0.686435 0.064151 0.364207 0 0 0 0  
 0.998617 0.906296 0.373319 2.32323 0 0 0 0  
 0.708568 2.16601 2.14061 2.34233 0 0 0 0  
  
[:, :, 3] =  
 0.0 0.0 0.0 0.245373 … 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.218143 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.174088 0.175847 0.0 0.0 0.0 0.0 0.0 0.0  
 0.16041 0.0296272 0.77125 0.584347 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0822869 0.0 0.0870539 0.0920338 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.153964 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.377731 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.391579 0.0879667 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.178373 0.052071 0.0 0.0 … 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0621035 0.0487938 0 0 0 0 0 0  
 0.0 0.0 0.509918 0.990183 0 0 0 0 0 0  
 0.0 0.0 0.0975961 0.627761 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 … 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 … 0 0 0 0 0 0  
 0.0707157 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
  
;;; …   
  
[:, :, 16363] =  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 … 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0 0 0 0 0 0 0  
 0.0 0.0 0.0 0 0 0 0 0 0 0  
 0.0 0.0259045 0.0 0 0 0 0 0 0 0  
 0.0 0.0 0.207233 … 0 0 0 0 0 0 0  
 0.0 0.0 0.0300441 0 0 0 0 0 0 0  
 0.0 0.0 0.0296209 0 0 0 0 0 0 0  
 0.0761566 0.228387 0.441014 0 0 0 0 0 0 0  
 0.275329 0.189209 0.147844 0 0 0 0 0 0 0  
 0.72749 0.303631 0.0207723 … 0 0 0 0 0 0 0  
 1.02891 0.220527 0.0399159 0 0 0 0 0 0 0  
 0.818845 0.127663 0.0 0 0 0 0 0 0 0  
 0.641579 0.0371985 0.0 0 0 0 0 0 0 0  
  
[:, :, 16364] =  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0 0 0 0 0  
 0.0 2.37197 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.683025 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0 0  
  
[:, :, 16365] =  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0326195 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0 0 0 0 0

# 6. Running the RainFARM Function

Now inputs to the RainFARM function must be determined, and they are informed by information from the methodology.

The first input into the function is the coarse precipitation data, or the coarse data P(X, Y, T). In this case, the coarse data is the precipitation over Texas.

Te second input is the slope. The slope is the spatial spectral slope. This value helps quantify the rate at which the power in the precipitation field decreases as you move to small scales in space.

The final input is nf, which is the ratio of coarse to fine data. For example, if nf=5 the downscaled data will have 5 times the resolution of the original data.

Below, the RainFARM function is called on the precipitation data. Code labels have been added below:

1. RainFARM needs boolean context inputs, and any missing values will come up as “Missing,” which is non-boolean. Consequently, the “Missings” need to be removed. To do so, the missing values were assumped to be no precipitation, so the default rain is set to 0.
2. The default rain value of no precipitation is used to fill in the missing values. This is the field fed into the function.
3. The spatial slope is set for 1. This slope is explored in more detail later.
4. The nf determines the ratio of coarse to fine data. For example, if nf=5 the downscaled data will have 5 times the resolution of the original data. It is also explored later.

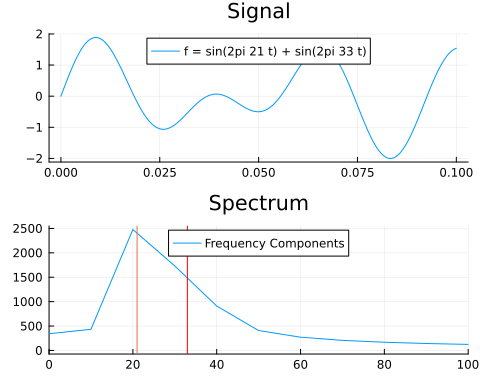
120×120×16365 Array{Float64, 3}:  
[:, :, 1] =  
 1.1258 1.16037 1.38086 … 0.0 0.0 0.0  
 1.44506 1.43336 2.50527 0.0 0.0 0.0  
 1.0311 1.56799 1.09789 0.0 0.0 0.0  
 0.874892 1.23732 0.764098 0.0 0.0 0.0  
 0.525064 0.621539 1.40911 0.0 0.0 0.0  
 1.07537 1.06291 1.06453 … 0.153296 0.0878582 0.0852226  
 1.28809 1.22201 0.806803 0.128118 0.110399 0.120915  
 1.22858 2.70768 2.02264 0.108942 0.0894241 0.0904012  
 0.930021 1.26382 1.56549 0.111893 0.0882434 0.116775  
 1.16465 0.921589 1.38677 0.146783 0.128764 0.222029  
 2.16746 1.30201 2.31049 … 0.117906 0.225888 0.345006  
 1.30349 2.79657 2.46928 0.273279 0.286014 0.164609  
 0.962148 2.29229 2.19201 0.335231 0.328274 0.172111  
 ⋮ ⋱   
 11.949 12.3627 15.8479 0.0 0.0 0.0  
 7.82801 12.069 16.0214 0.0 0.0 0.0  
 11.0665 24.5594 27.1277 … 0.0 0.0 0.0  
 7.59787 16.2302 12.8263 0.0 0.0 0.0  
 9.37665 13.5133 13.988 0.0 0.0 0.0  
 14.5722 26.544 28.9921 0.0 0.0 0.0  
 20.0833 37.4336 45.7416 0.0 0.0 0.0  
 16.0801 29.4971 32.5569 … 0.0 0.0 0.0  
 19.7677 40.2596 21.8404 0.0 0.0 0.0  
 19.9697 22.9062 18.1744 0.0 0.0 0.0  
 14.3727 9.67227 6.91133 0.0 0.0 0.0  
 11.5549 8.95878 11.7744 0.0 0.0 0.0  
  
[:, :, 2] =  
 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 ⋮ ⋱ ⋮   
 0.547373 0.745725 1.69867 2.79066 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.500528 1.18557 1.50172 1.48226 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.516825 0.357264 0.466782 0.475887 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.198897 0.429607 0.765147 0.677536 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.320708 1.2029 1.08233 1.07384 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.629891 1.0439 1.52 1.31932 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 1.34924 0.94964 1.52385 3.21685 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.734174 0.938663 1.39171 2.0493 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.726172 1.63849 1.41912 1.21134 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.357942 0.642676 0.454525 0.490942 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.457695 0.48771 0.333211 0.221675 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.264566 0.180388 0.241581 0.312354 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
  
[:, :, 3] =  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 ⋮ ⋱ ⋮   
 0.0470454 0.0921972 0.0417151 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.122382 0.0468411 0.0402437 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
  
;;; …   
  
[:, :, 16363] =  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 ⋮ ⋱ ⋮   
 0.246877 0.454326 1.16141 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.230942 0.460985 0.713604 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.367266 0.799225 0.531326 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.254736 0.415117 0.50498 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.46642 0.551495 0.263466 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.239817 0.4338 0.522514 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.716099 0.503328 0.570044 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.704518 0.85508 1.03389 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.731543 0.454795 1.02805 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.755952 0.296454 0.542497 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.251868 0.0953299 0.142379 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0855855 0.0640299 0.101344 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
  
[:, :, 16364] =  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 ⋮ ⋮ ⋱ ⋮   
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
  
[:, :, 16365] =  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 ⋮ ⋮ ⋱ ⋮   
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 … 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0  
 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

This created a variable, downscaled\_precip\_tx, with the downscaled precipitation data over the same time period of the original precipitation data.

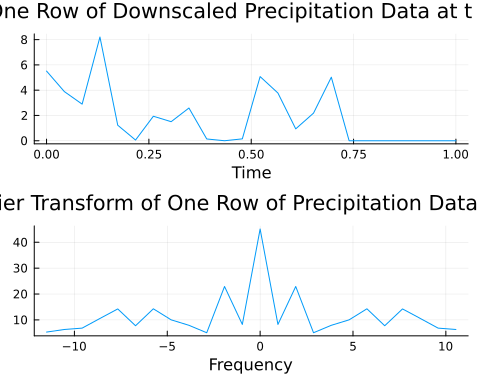
# 7. Fourier Analysis in Code

I have included some examples of how the Fourier Analysis is used to help better understand the methodology used in the rainfarm package.

The first example is an example created off of a randomly generated sin wave with multiple frequency components.



This next example selects a random row and time step in our precipitation field to visualize how the Fourier transform is applied to the data we have.

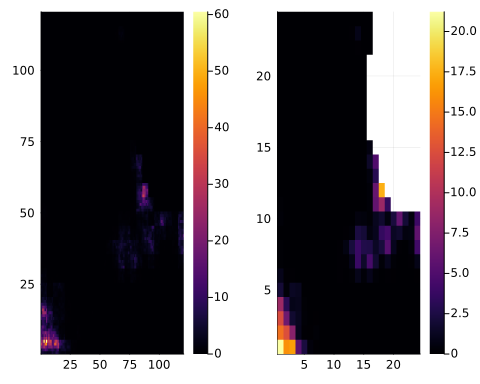


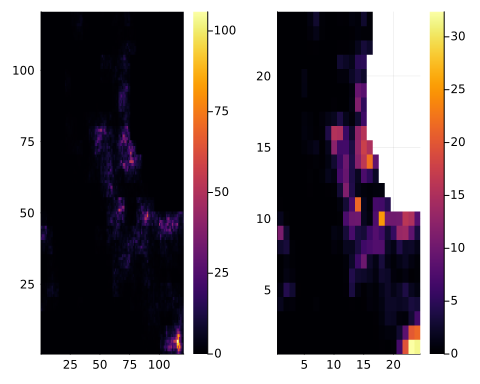
# 8. Results Analysis

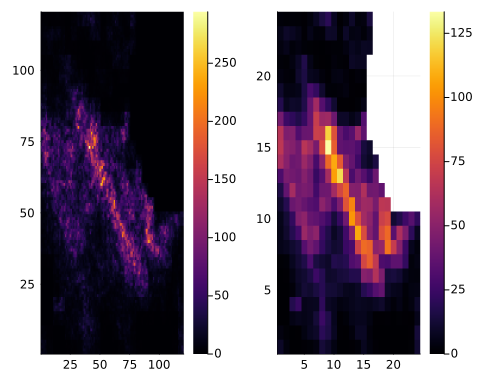
## 8.1 Coarse Input vs. Fine Output

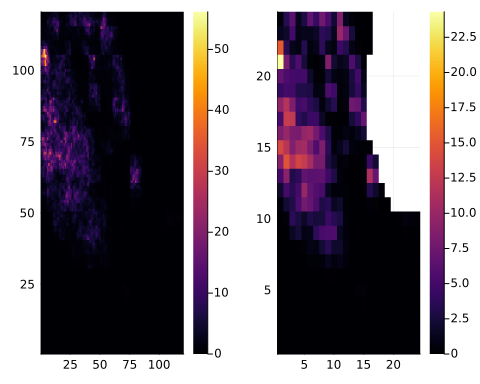
First, it is possible to analyze the original, coarse data and the downscaled data side by side. This can be done by plotting the data at a specific time frame. Consider 10 randomly selected precipitation fields.

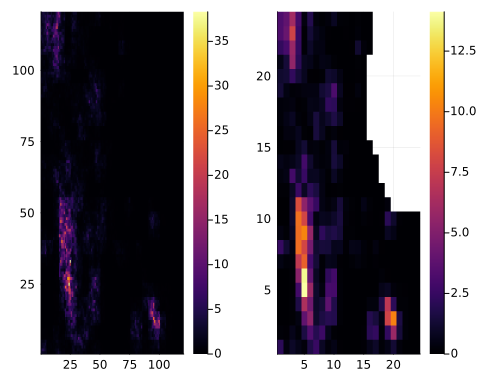
1. Generate 10 random time frames from our downscaled data.
2. Plot the downscaled data
3. Plot the original data



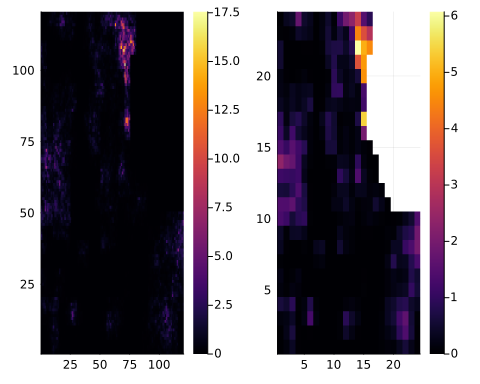


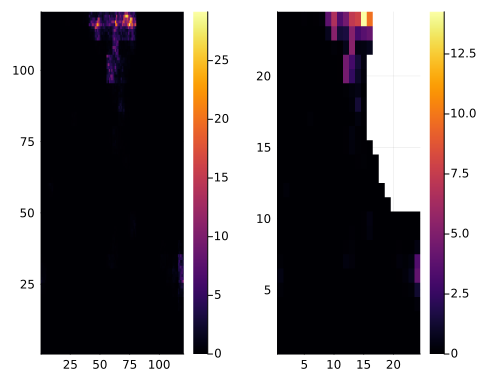


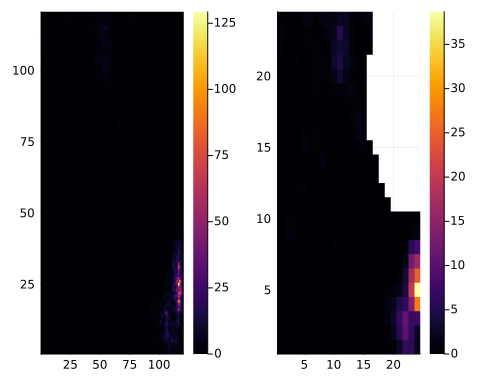


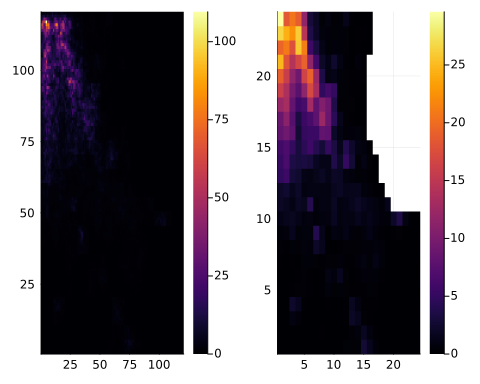






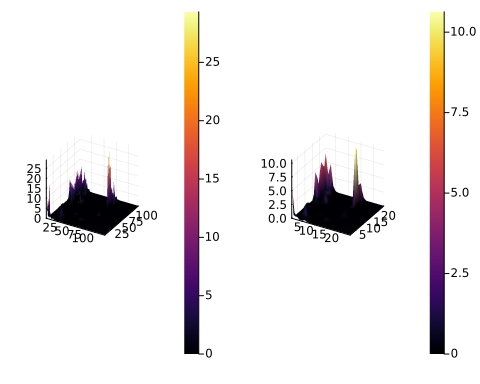


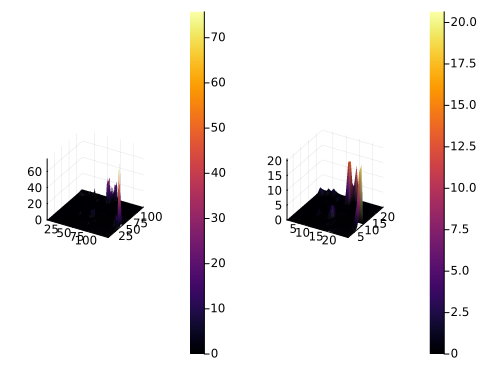


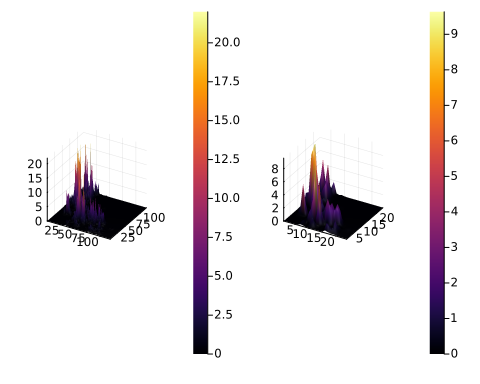


Visually, these side-by-side comparisons show reasonable results for the downscaling. The highest precipitation hot spots are located in the same places across the coarse and fine data, and general spatial patterns are similar.

You can also see this three-dimensionally. Below, the plots show the downscaled data plotted on the grid.





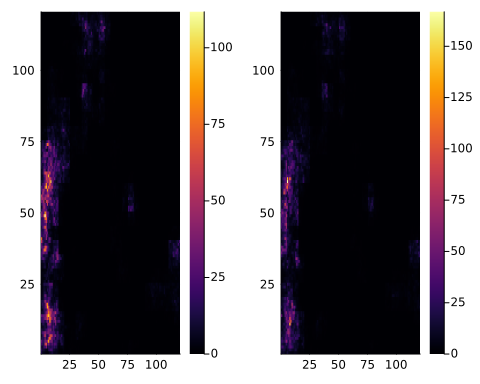


## 8.2 Stochastic Visual Analysis

However, it is also important to remember that this is a stochastic process. Let’s explore the different options that can be generated by this same function.

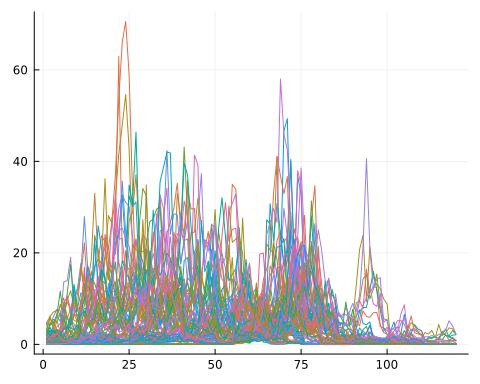
The code below will generate a random time step and then run the downscaling function two times at this time step. The two outputs will be plotted side by side to demonstrate that there are variations within these results. This visually demonstrates that this process is stochastic.

1. Generate a random time step
2. Call the original data on the time step
3. Fix the missing data using a default value
4. Call the downscaling functions multiple times on the same data
5. Plot the results side by side

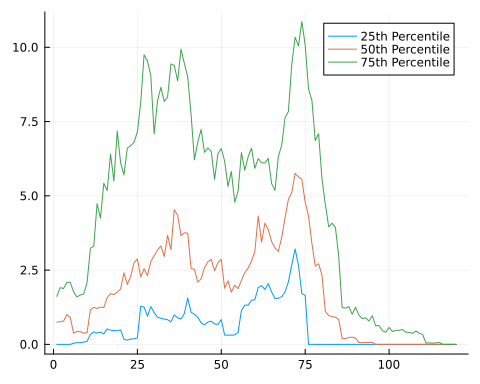


We can explore this further below. The code below plots the downscaled data of a random row at a given time step 100 times to show the variation within these outputs.

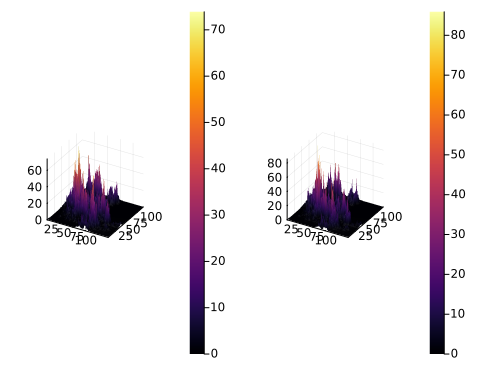
1. Generate a random time step
2. Take a random “slice” or row
3. Plot these realizations



This next code shows the quantiles of these realizations. This is a useful way to visualize the variation in the data. The quantiles are plotted for each time step.



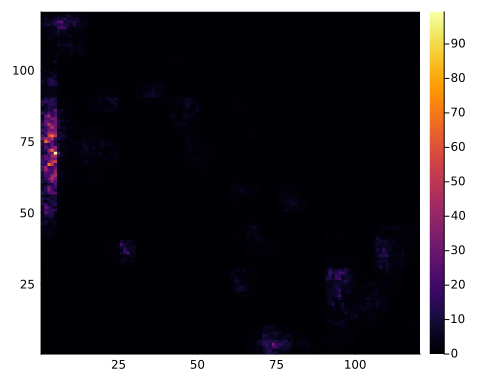
The code below again shows the stochastic nature of these realizations with a surface plot.

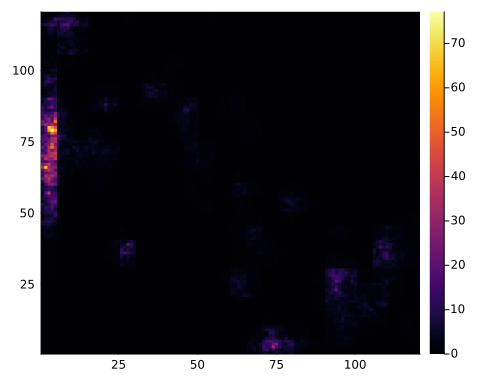


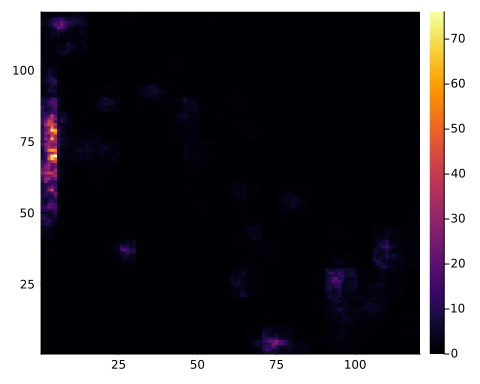
## 8.3 Spatial Slope Analysis

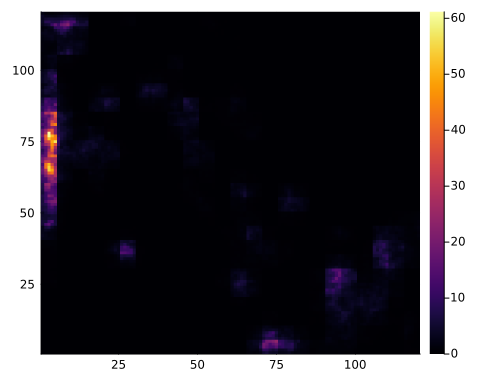
However, another interesting thing to analyze is the determination of the spatial slope factor. This is an important component of our frequency analysis of the energy/precipitation in the system and how large this rate of change is. Let’s compare the graphs with different spatial slopes. Higher spatial slopes correspond to a faster rate of change.

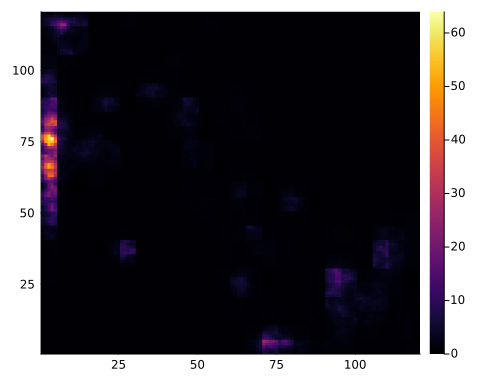
1. Test over a range of slopes between 0.5 and 7, with a step of 0.5.
2. Call the downscaling on each slope value
3. Plot the results

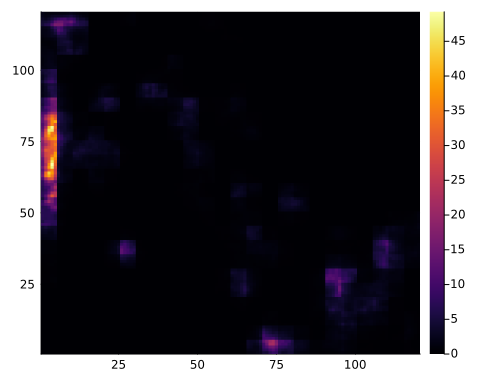


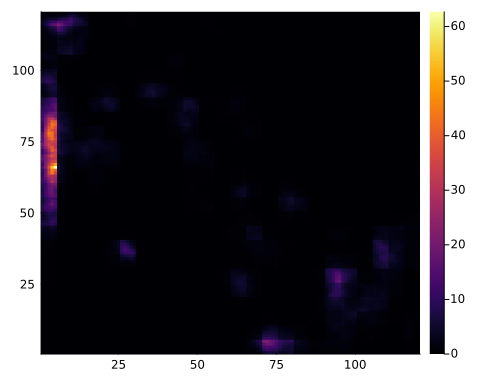


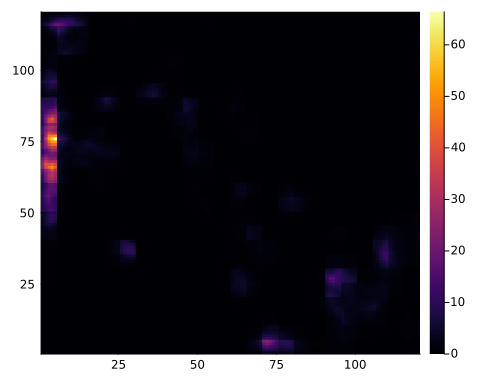


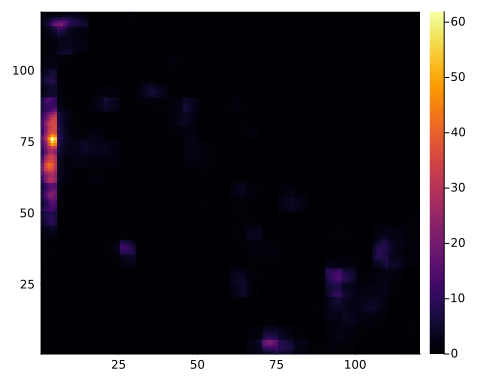


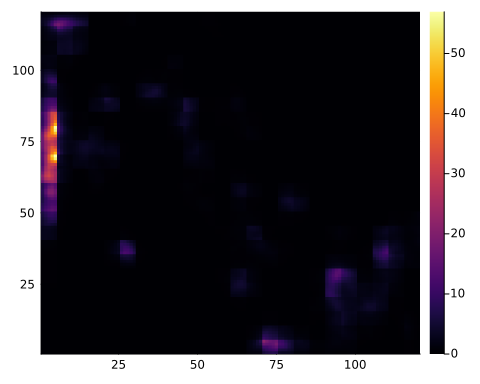


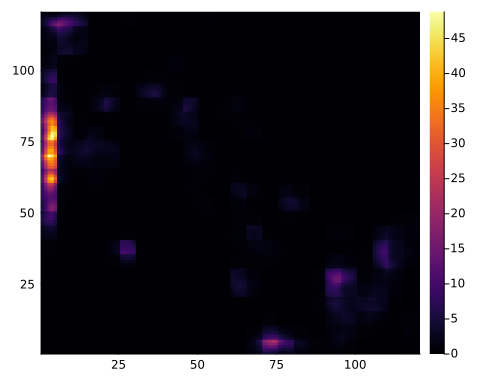


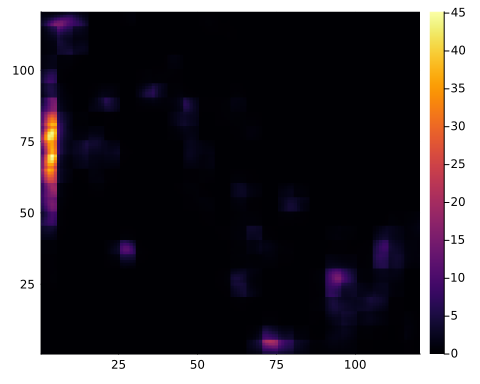


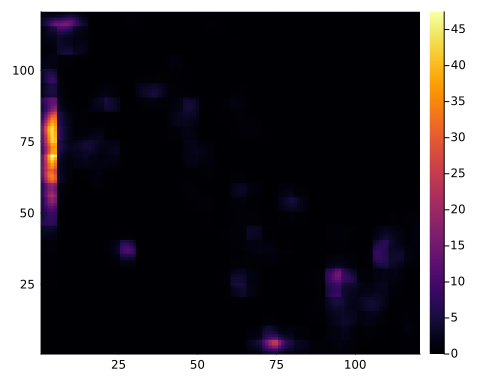


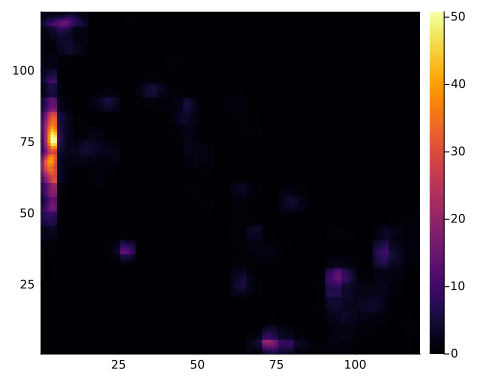


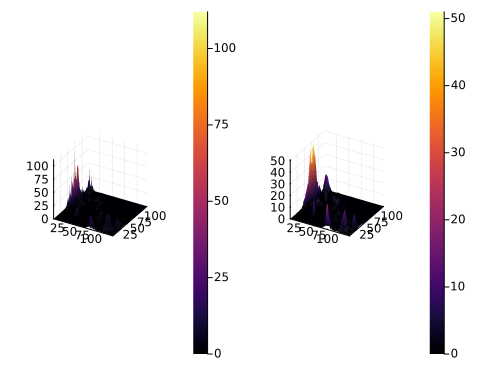










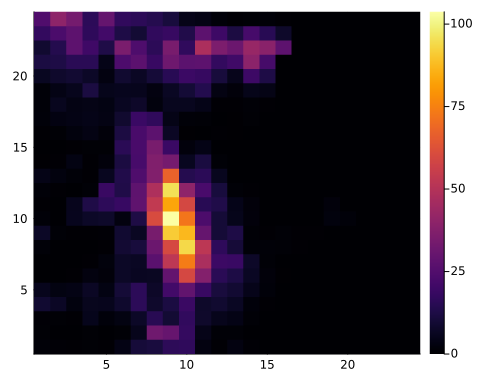


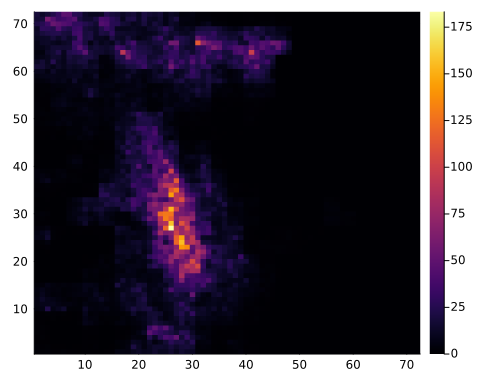
As the slope grows, there is a ‘blurring’ effect on the data. The smallest slope generates the most specific estimate and is the least blurry of our predictions. This generates an interesting question. The available RainFARM documentation does not give many recommendations regarding this spatial slope, and the issue, therefore, becomes a question of variance. With a small slope, there is a small rate of change the predictions are very specific and localized. With a high slope, there is a high rate of change so the predictions become blurry and less specific. Therefore, low slopes have low variance and high slopes have high variance. Low variance is great when the model is accurate, but it can cause problems with generalization. High variance can generalize well, but the accuracy is limited. This is an important question to consnider when setting up this RainFARM model.

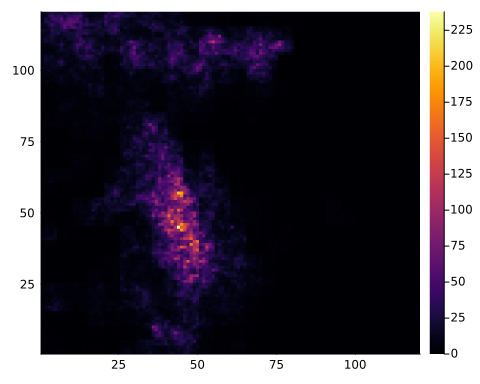
## 8.4 Coarse to Fine Ratio Analysis

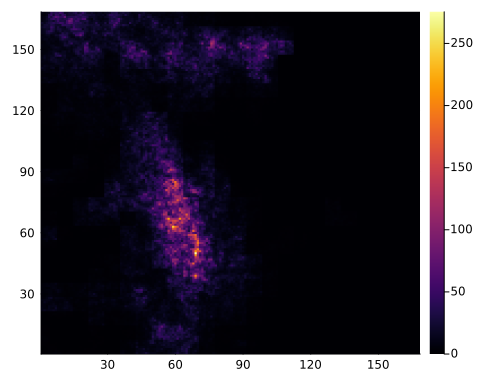
We can also consider the ratio of coarse to fine data, or the nf variable. This variable determines the resolution of the downscaled data. Let’s compare the graphs with different nf values. Higher nf values correspond to a higher resolution.

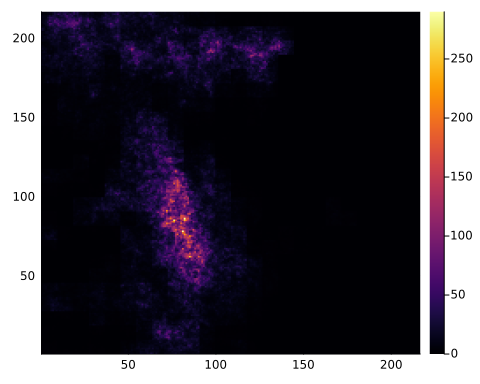
1. Call on your random time step
2. Test a range of nf values between 1 and 30, with a step of 2.

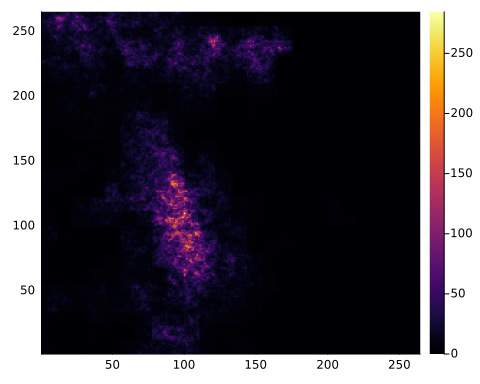


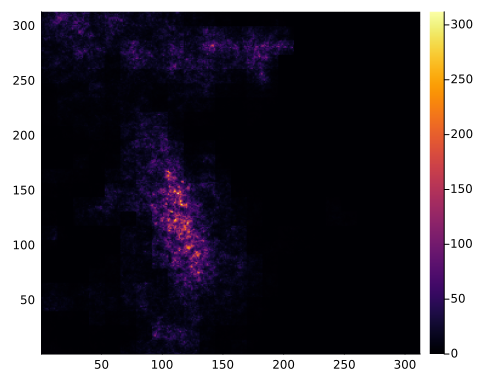


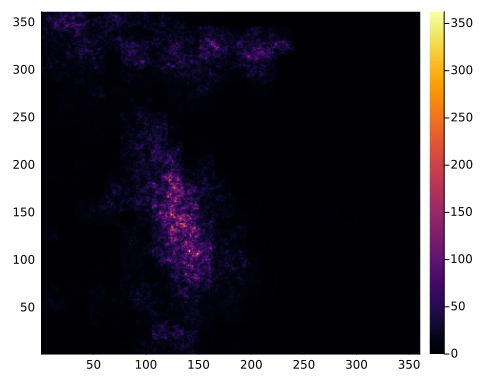


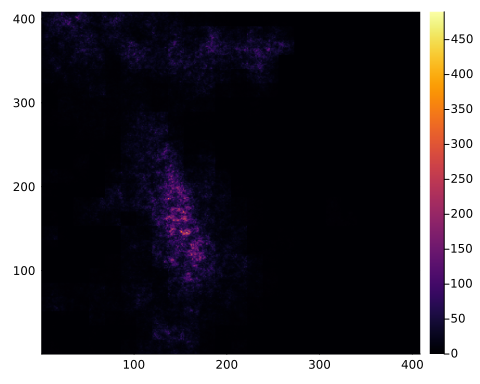


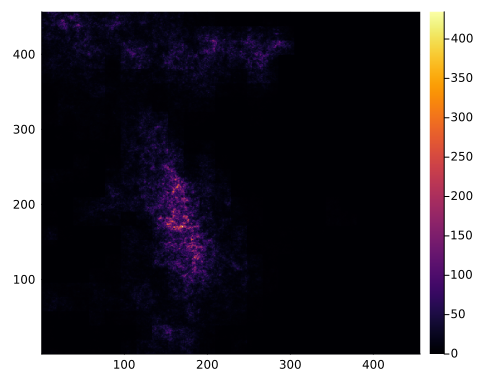


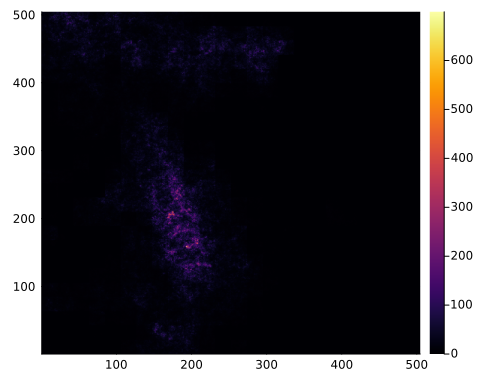


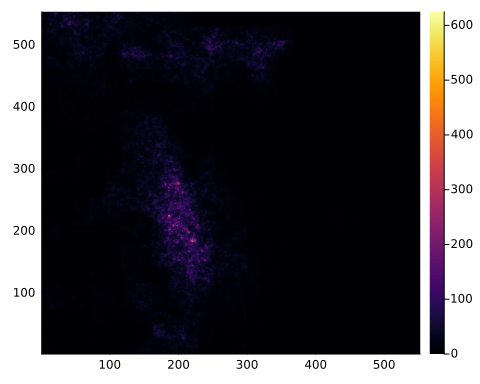


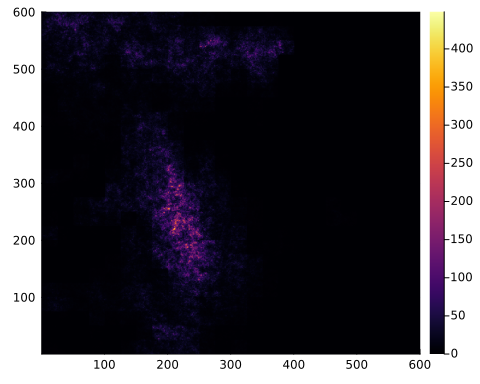


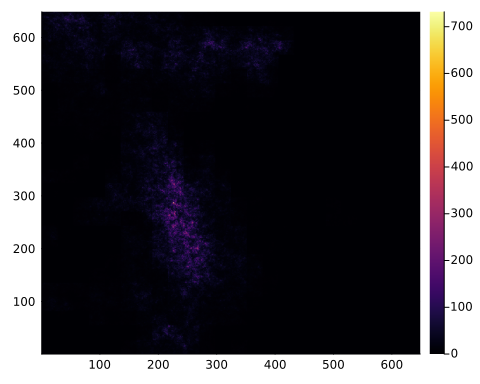


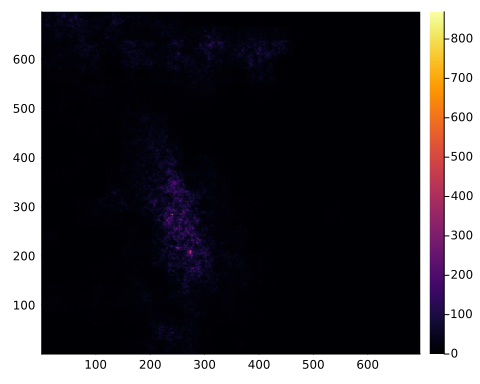












This analysis starts with nf = 1, meaning the output is the original coarse data. As nf grows, the downscaled data becomes increasingly fine. However, there does seem to be a potential point at which the downscaled data is not improving and increasing the ratio only serves to diminish the hotspot peaks of the precipitation. In the heatmap, this can be seen as the downscaled patterns stop changing as much as the ratio increases, but the hotspots dim. This is another important question to consider when setting up this RainFARM model.

# 9. Conclusions

## 9.1 Considerations

Spatial Slope: The choice of spatial slope introduces a trade-off between specificity and generalization. Low slopes provide specific but localized estimates, while high slopes result in more generalized but blurry predictions.

Coarse to Fine Ratio (nf): The analysis indicates a potential point of diminishing returns for increasing the ratio, where fine-scale details may be lost. Recommendations:

## 9.2 Model Performance

The RainFARM Model seemed to make good predictions and acknowledges the sources of error in precipitation data. These stocastic predictions represent the statistical trends of the coarse data well. I believe it is a useful model, and I hope to use it for my thesis later!