RainyDay User's Guide

Daniel B. Wright, Ph.D.
danielb.wright@wisc.edu

Updated August, 2019

Table of Contents

1. Overview

RainyDay is a framework for generating large numbers of realistic extreme rainfall scenarios based on relatively short records of remotely-sensed precipitation fields. It is founded on a statistical resampling concept known as stochastic storm transposition (SST). These rainfall scenarios can then be used to examine the extreme rainfall statistics for a user-specified region, or to drive a hazard model (usually a hydrologic model, but the method produces output that would also be useful for landslide models). In contrast with other recent flood hazard modeling approaches, RainyDay is well suited to flood modeling in small to medium-sized watersheds[*]. The framework is made to be simple yet powerful and easily modified to meet specific needs, taking advantage of Python's simple syntax and well-developed libraries. It is still a work in progress. Therefore, the contents of this guide may be out-of-date. I will attempt to keep the documentation in synch with major changes to the code. I would appreciate any feedback on this document and on RainyDay itself, so I can continue to improve both.

Please note that while this framework is relatively simple, it does require some understanding of hydrometeorological principles. In addition, the configuration of Python, as well as any desired alteration to the code, requires some basic familiarity with programming and scientific computing. While I will not provide unconditional technical support, I am interested in seeing people use this software and so I will provide assistance as much as possible.

It is essential that the user understand the basic principles of SST prior to running an analysis. This is because, given the degree of flexibility provided by RainyDay, it is very easy to make conceptual mistakes that will produce unrealistic results. We are working to make this learning process easier. The best way to learn is to read this paper:

- Wright, D.B., R. Mantilla, and C.D. Peters-Lidard. A remote sensing-based tool for assessing rainfall-driven hazards, *Environmental Modelling & Software*, 2017.

The RainyDay source code, available on Github (see Chapter 3 of this guide for a link to the repository), also features two example analyses, one for a watershed in northern Colorado and another for IDF estimation in Madison, Wisconsin. These examples are briefly described in Appendix C and Appendix D of this guide, respectively.

We have also produced a series of three brief Youtube videos about RainyDay. The first of these is a general overview. The second and third describe how to use a web-based version of RainyDay, which is the quickest and easiest way to run the analysis (though you will not have access to all of the software's features.

Additional relevant references include:

---

[*] It is not possible to provide a particular watershed size above which RainyDay is not well-suited, but in general it will work better for basins in which floods are the result of singular rain events (which could last hours to multiple days), rather than combinations of distinct rain events over an extended time period which act in concert to create the flood wave.

- Wright, D.B., J.A. Smith, M.L. Baeck. Flood Frequency Analysis Using Radar Rainfall Fields and Stochastic Storm Transposition, *Water Resources Research*, 50 (1592–1615), 2014.
- Wright, D.B., J.A. Smith, G. Villarini, M.L. Baeck. Estimating the frequency of extreme rainfall using weather radar and stochastic storm transposition, *Journal of Hydrology*, 448(150-165), 2013.
- Yu, G., D.B. Wright, C. Smith, Z. Zhu, K. Holman, Process-Based Flood Frequency Analysis in a Changing Agricultural Watershed, *Hydrologic and Earth Systems Science*, 2019.

RainyDay has two main steps: storm catalog creation, and storm transposition. By tweaking the parameter (".sst") file, the user can perform either one or both of these steps in a single instance. Please note that both steps, but in particular the storm catalog creation, can be time consuming, since the remote sensing datasets input can be very large. Runtimes of an hour or more is certainly possible, particularly on older machines or for very high-resolution datasets. Runtimes for storm transposition should be approximately 1-10 minutes depending on the selected options and resolution of the dataset.

The .sst file is described in more detail below. This document does not provide significant detail on the formatting of the input remote sensing data, but it is important to note that, due to the unfortunate variety of formats of various remote sensing datasets, a preprocessing step is needed to convert the original remote sensing files into a format that RainyDay will recognize. If you would like to know more about this format and preprocessing, please contact me. If feasible, I will share preprocessing scripts that I have created for different datasets, or directly share the datasets that I have prepared. This preprocessing in most cases involves selecting a certain geographic region, such as the contiguous United States, in order to reduce the size (and therefore the time that your computer has to spend loading the files) for the process.

A note about geographic projections: Earlier versions of RainyDay supported different geographic projections for some operations. This functionality was never perfected and for simplicity has been removed. RainyDay assumes that all input GIS data and coordinates are in the WSG84 geographic coordinate system (EPSG:4326, +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs). Furthermore, input rainfall must be projected onto a regular rectangular latitude-longitude grid (though the grid spacing need not be the same north-south as east-west). All coordinates supplied by the user in the .sst file must be in latitude/longitude, with negative latitude indicating southern hemisphere and negative longitude indicating western hemisphere.

There are three main types of output:

- **Storm Catalogs:** A storm catalog is the collection of storms that will be used for subsequent resampling and transposition. It is a NetCDF file. The file structure is detailed in Appendix A. Storm Catalogs can also facilitate all sorts of interesting non-SST analyses.

- **Diagnostic plots:** These plots show different aspects of the rainfall fields and the storm catalog. These diagnostics are explained in more detail in the following sections.
- **.FreqAnalysis file:** This file provides the results from a rainfall frequency analysis for various return periods for an area and a rainfall duration, both of which are specified by the user. The results in the .FreqAnalysis file are roughly equivalent to an Intensity-Duration-Frequency (IDF) curve, which many civil engineers and hydrologists are familiar with. There are important differences, however. The frequency analysis from RainyDay is, by definition, a spatial average, rather than a point analysis. An IDF curve is typically based on rain gage observations, which typically measure rainfall over an area of approximately 0.1 m^2. RainyDay uses remote sensing grid cells, each of which typically covers an area of 10^5 to 10^9 m^2. Therefore, caution is needed when comparing these results against conventional IDFs.
- **Rainfall IDF plot:** This provides a visual representation of the results provided in the .FreqAnalysis file.
- **Spacetime rainfall scenarios:** These are scenarios that can serve as input to a hazard model (typically a hydrologic model). In order to keep the size of these scenarios manageable, they do not consider the entire transposition domain but only the rainfall within a certain user-defined area (usually either a rectangular "box" or a watershed boundary). The file structure for the rainfall scenarios is detailed in Appendix A.

2. Licensing Agreement and Disclaimers

The most recent version of RainyDay is distributed under the MIT Open Source License. Note that earlier some versions were released under the GNU GPL Version 3.0 license, which differs from the MIT license in some important ways.

Disclaimer: I, Daniel Benjamin Wright, as the author of this software, will not be held responsible for the accuracy of the output and results it produces, nor for decisions taken based on said output and results. The input datasets can contain significant uncertainties that will impact the accuracy of output and results. Furthermore, the poor choice of parameters in RainyDay will negatively influence the output and results. Please consult the relevant literature and the RainyDay User's Guide prior to using this software.

3. Installation

RainyDay can be obtained from the Github repository: https://github.com/danielbwright/RainyDay2. I have provided one precipitation datasets for use: the hourly, ~4 km resolution Stage IV dataset. It can be downloaded here. It has fairly high resolution and accuracy for the United States east of the rocky mountains, but is not recommended for the mountainous western United States, and does not extend much beyond the US border. You may contact me if you have interest in collaborating on the use of RainyDay with other rainfall datasets.

1. Install Python: I strongly recommend using the Anaconda distribution from Continuum Analytics. The current version of RainyDay requires Python 3.X (earlier versions used Python 2.7). Installation is easy, it comes with most of the packages that you'll need, and has several options for code development environments (I use Spyder, which comes as part of the Anaconda Launcher, but the other options might be worth examining). If you are new to Python and Anaconda, the easiest way to proceed is to install Anaconda, and then, using either Spyder or the command line, try to run RainyDay (syntax is described below). When errors related to certain packages arise, install the required package (also described below) and re-run RainyDay.

The current version of RainyDay uses the following Python packages (though many of these packages have multiple 'dependencies'—other packages required to run; dependencies are automatically installed by Anaconda):

- os
- sys
- numpy
- scipy
- math
- datetime
- time
- copy
- pandas
- shapefile (pyshp)
- matplotlib
- numba
- subprocess
- glob
- timeit
- netCDF4
- rasterio
- warnings

In addition, you will need to have the GDAL library installed on your machine. GDAL is notorious for compatibility issues with python libraries. To facilitate this, RainyDay accesses GDAL via system commands, rather than through the 'gdal' Python library.

This framework will be installed when you install the 'rasterio' package. I cannot guarantee that this will be painless, however!

2. Configure PYTHONPATH (optional): There might be other ways to do this, but here is the one that I know (on Mac and Linux). In your home directory, create a file called ".bashrc". This is easy to do. At a command line, simply type "vim ~/.bashrc". Regardless of whether this opens a new or existing file, add the following to ".bashrc":

export PYTHONPATH="${PYTHONPATH}:[proper path here]/RainyDay_utilities"

"[Proper path here]" denotes the path to wherever the RainyDay directory resides. This could be in with the rest of your Python libraries, or elsewhere.

Certain linux shells will use other files rather than .bashrc, such as .profile. The syntax differs somewhat, but explanations are relatively easy to find on the internet. I do not know how to configure PYTHONPATH in Windows. It can also be done in the Spyder interactive development environment. In the MacOS version of Spyder, open the 'PYTHONPATH Manager' from the 'Python' dropdown menu.

3. Configure Python packages: Try to run RainyDay using Python/IPython from a command line terminal:

    $ python {RainyDay script name}.py

If you configured PYTHONPATH as explained in step 2, and you get an error related to not being able to find RainyDayUtilities, Type the following command at the terminal:

    $source ~/.bashrc

That should set PYTHONPATH.

The script will not fully execute even if the packages are properly configured, because you haven't supplied a parameter file (.sst file). But if any packages aren't installed, you will get errors that will help you determine what packages haven't been properly installed.

You can look at all packages currently installed under Anaconda using:

    $ conda list

You should be able to install most packages in Anaconda using:

    $ conda install {package name}

If you want to see a list of all packages that can be installed, and the correct names, use:

    $ conda search

4. Preparing an Analysis

1. Prepare the parameter file (.sst file): The .sst file should contain comments which explain the different options. They are explained in more detail below. Though not thoroughly tested, I do not think the order in which these fields appear in the .sst file should matter at all. NOTE: comments can be entered in the .sst file, following a "#" symbol. A comment could be an entire line, or could follow a non-comment on the same line. See below for examples. Most fields in the .sst file are technically optional, since RainyDay will use default values if they are omitted. To produce meaningful results, however, thorough understanding of most of the fields is necessary.

Example .sst file: all file paths in the example shown below would need to be updated to run RainyDay on a different machine. Most comments omitted for clarity:

```
#=====================================================================
# WELCOME TO THE RAINY DAY MODEL!
# USER DEFINED INFORMATION FOR STOCHASTIC STORM TRANSPOSITION
#=====================================================================

# this is an example comment

# BASIC INFO ON THE SCENARIO
MAINPATH	/RainyDayGit/Source/Preprocessing
SCENARIONAME		BigThompson_testing_StageIV
GDALPATH	/Library/Frameworks/GDAL.framework/Versions/Current/Programs

# DEFINE THE RAINFALL INPUT DATASET (NOTE: NOT ALL DATASETS ARE CURRENTLY SUPPORTED, BUT SHOULD BE ABLE TO BE ADDED WITH MINIMAL WORK)
RAINPATH	/DATA/StageIV/NEWREPROJ/ST4.*.newregridded.nc


# SST PARAMETERS
CATALOGNAME			StageIV_24hour_BigThompson_testing.nc
CREATECATALOG		true				# this is an example comment
DURATION		24
NSTORMS				350
NYEARS				1000
NREALIZATIONS		100
TIMESEPARATION		12
DOMAINTYPE		rectangular
DOMAINSHP		/RainyDay2/RainyDay_NewOrleans/NewOrleansSSTdomain.shp
DOMAINFILE	/RainyDayGit/Source/PRISM_24hour_domain_BigThompson_basin.nc
LATITUDE_MIN		42.25
LATITUDE_MAX		44.25
LONGITUDE_MIN		-91.0
LONGITUDE_MAX		-88.0
DIAGNOSTICPLOTS			true
FREQANALYSIS		true
SCENARIOS		false
#EXCLUDESTORMS		40,58,102,187,197	# entire lines can be commented out if needed
EXCLUDEMONTHS		1,2,3,12
INCLUDEYEARS		all
TRANSPOSITION			uniform
```

```
RESAMPLING            poisson
SPREADTYPE            ensemble
RETURNLEVELS          all
DURATIONCORRECTION true
STOCHASTICRESCALING                   false
RAINDISTRIBUTIONFILE /Preprocessing/StageIV_Top100storms_Daily.nc

POINTAREA     basin
WATERSHEDSHP RainyDayGit/Source/BigThompson/Watershed/BigThompson.shp
POINTLAT      40.365
POINTLON      -105.65
BOX_YMIN      43.
BOX_YMAX      43.25
BOX_XMIN      -89.5
BOX_XMAX      -89.25
```

2. Description of Parameters:

MAINPATH: Directory where the RainyDay Python script is located and in which subdirectories will be created or modified.

SCENARIO: Name for the scenario. This will be the name of the subdirectory and the prefix for various output such including the frequency analysis file. If that subdirectory doesn't exist, it will be created.

GDALPATH (not always necessary): The GDAL library sometimes runs into conflicts and version issues, particularly when multiple versions exist on one machine (for example, one for Python and one for QGIS). Sometimes, it is necessary to provide the exact location of GDAL to RainyDay in the .sst file, using the GDALPATH. Finding this location can be tricky. An internet search could probably help point you in the right direction. Try omitting or commenting out this line first and test to see if RainyDay will run without it.

RAINPATH: The location of the rainfall input NetCDF4 files. Only needed if you are creating a new storm catalog. Wildcards ("*") can and should be used. For example, '/DATA/StageIV/NEWREPROJ/ST4.*.newregridded.nc' is valid. Note that, as mentioned before, these files must meet certain formatting conditions-it is not enough that they be netCDF4 files.

CATALOGNAME: The name of the storm catalog. If the user specifies that a new storm catalog should be created (using the CREATECATALOG field, which is explained next), a new file will be created (and any existing one with the same name will be overwritten). If the user does not create a new catalog, the CATALOGNAME should refer to an existing storm catalog. This catalog resides at: MAINPATH/CATALOGNAME}.nc. The storm catalog will be a netCDF4 file.

DURATION: Duration of the rainfall accumulation period in hours. If you are not creating a new storm catalog but rather using one that has already been generated, this duration is compared against the duration of the storm catalog. If the specified duration is shorter than that of the storm catalog, the rainfall duration is adjusted accordingly. If the specified duration is longer than that of the storm catalog, an error message is generated. If you are only intending to estimate rainfall IDF, this selection is straightforward. It is more complicated if you are planning on using RainyDay outputs for hydrologic modeling. In that case, there are no simple guidelines for how to choose the duration. If you are just examining rainfall frequency analysis, then any duration can be used. If spacetime rainfall scenarios are being generated to drive a hydrologic model, then it is best to choose at least 24 hours. The duration should be longer for larger watersheds. It is not recommended to use conventional notions of response time or time of concentration as the criteria for selecting an appropriate duration. This duration should exceed the time of concentration. In principle this duration could be as long as you want to facilitate modeling of very large watersheds. This is not recommended however, and for a number of reasons will not likely produce satisfactory results. A modified version of RainyDay is under development that will work better for very large watersheds. If you have an existing storm catalog that has a duration of, for example, 48 hours, you can specify a shorter duration (for example, 24 hours) and RainyDay will "trim" the longer-duration catalog**.**

Due to some issues intrinsic to how stochastic storm transposition works, and its implementation in RainyDay, if you are attempting to produce short-duration IDF estimates (i.e. a relatively short value given for DURATION), your results will be more accurate if they are generated from a longer-duration storm catalog, with a large number of storms. For example, 6-hour IDF estimates will be more accurate if they are produced using a 48-hour storm catalog. The reasons for this are complicated, just trust me! The DURATIONCORRECTION (see below) option has been added to address this issue.

DURATIONCORRECTION (optional): This new addition to RainyDay is highly recommended if you are interested in generating IDF curves, particularly if they are of relatively short duration (e.g. 24 hours or less). This option will generate a longer-duration catalog (it will be either 72 hours or 3 times the length specified in DURATION-whichever of the two is longer). Then, during transposition and resampling, RainyDay will identify the heaviest rainfall periods within these longer rainfall periods. This, combined with a large number for NSTORMS (such as the default) can help reduce, though not fully eliminate, underestimation biases for more frequent recurrence intervals. It comes at a computational cost, however—the resampling and transposition step can take significantly longer. This is most recommended if you are generating pixel-scale or small-area IDF curves. This option has no effect on the NetCDF scenarios that RainyDay produces for subsequent hydrologic modeling, since conceptually that idea doesn't make a lot of sense. However, it won't cause these scenarios to be "wrong" in any way.

NSTORMS (optional): How many storms to include in the storm catalog. This number will be inherited from the existing storm catalog, if you are not creating a new storm catalog. If you omit this line, a default will be used of 20 storms per year. This number will be reduced accordingly in the following situations: 1.) you are not creating a new catalog and you wish to exclude certain storms (EXCLUDESTORMS); exclude certain months that were not previously excluded in the creation of the storm catalog (EXCLUDEMONTHS), or exclude certain years via the INCLUDEYEARS option. Common return periods (for example, less than 50 years), particularly for short durations, are highly sensitive to this choice. It you care about common return periods, then NSTORMS should be relatively large. There is no simple guidance on what this means, but I recommend something like at least 20 storms per year of available rainfall data (which is the default number if you omit this line).

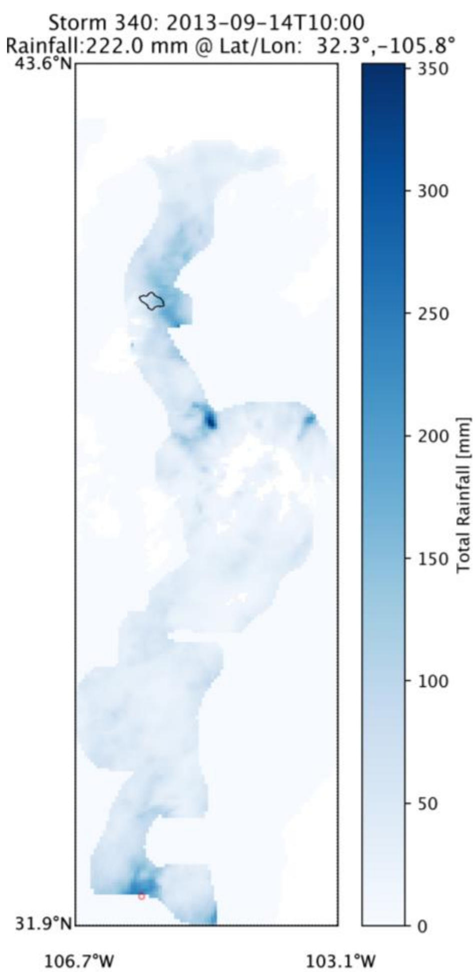


Figure 1: Sample diagnostic plot of storm total rainfall from a 72-hour duration event over eastern Colorado ending at 10:00 on September 14, 2013. Input data is the National Stage IV multisensor QPE Product. An irregular transposition domain was used for this figure. The black outline is the watershed (Big Thompson River upstream of Olympus Dam) used in this analysis.

NYEARS (optional): How many years of annual maxima rainfall to be synthesized. Note that this is not the number of years of rainfall data available, which RainyDay will calculate automatically. If omitted from the .sst file, this will default to 100, which means that the maximum return level that can be estimated is 100 years.

NREALIZATIONS (optional): How many NYEARS-long sequences to be generated. Technically this is optional-the default value is 1. This is not recommended, however, since it will not return uncertainty bounds of any kind.

TIMESEPARATION (optional): The minimum separation time in hours between two storms in the storm catalog. During storm catalog creation, when a new rainfall event is being evaluated for inclusion, its timestamp is compared against those already in the catalog. If the difference in timestamps between the new storm and an existing one is less than TIMESEPARATION, the larger event (in terms of rainfall accumulation) is retained and the smaller event is discarded. If this value is omitted from the .sst file, it will default to zero, meaning storms can be consecutive (but will not temporally overlap).

DOMAINTYPE: There are two options: 'rectangular' (the default) or 'irregular'. 'rectangular' is the simplest, but may only be appropriate in regions with few topographic or coastal features.

The following values are needed only if DOMAINTYPE is set to 'rectangular'. It is recommended that these values not be changed between runs unless a new storm catalog is created. The program is untested and may be unpredictable if these values are changed while an existing storm catalog is used:
LATITUDE_MIN: Southern boundary of transposition domain.
LATITUDE_MAX: Northern boundary of transposition domain.
LONGITUDE_MIN: Western boundary of transposition domain.
LONGITUDE_MAX: Eastern boundary of transposition domain.

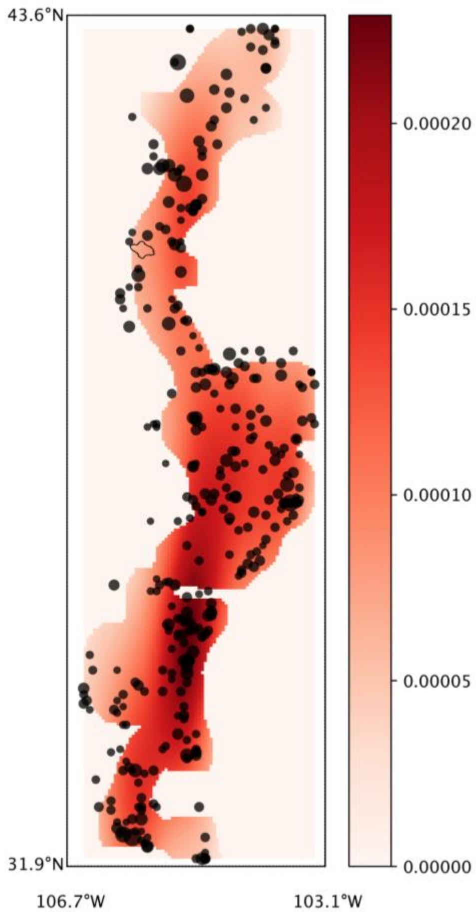


Figure 2: Spatial probability of storm occurrences (red shading) based on locations of storms in storm catalog (black circles). The black outline is the watershed (Big Thompson River upstream of Olympus Dam) used in this analysis.

The following values are needed only if DOMAINTYPE is set to 'irregular'. There are two options-providing a netcdf file that contains a "mask" of the domain, or a shapefile that contains the mask. Both masks must be in the EPSG:4326 geographic projection (i.e. regular latitude/longitude grid). You only need to provide one of the two: DOMAINSHP (if using a shapefile, which is generally easiest) or DOMAINFILE (if using a netcdf file, which may become more streamlined in the future as I develop more streamlined ways of generating and using such files):
DOMAINSHP /Users/daniel/NewOrleansSSTdomain.shp
DOMAINFILE /RainyDayGit/Source/PRISM_24hour_domain_BigThompson.nc

Instructions for "drawing" the shapefile needed to use the DOMAINSHP option, using Google Earth, is shown in Appendix B.

DIAGNOSTICPLOTS (optional): Set to "true" (the default) to produce diagnostic plots. This can take a few minutes, and is only recommended when a new storm catalog is created. There are three types of plots produced. Figure 1 shows an example of a diagnostic plot of storm total rainfall. One such plot will be produced for each storm in the storm catalog. The various components of this figure are explained in the descriptions of other parameters.

Figure 2 shows another diagnostic plot: the probability of storm occurrence. Black circles show the locations of the storm centers that were identified through the storm catalog generation. A 2-dimensional Gaussian smoother is used to estimate the spatial probability of occurrence based on these locations. The red background colors indicate areas higher probability. Do not be alarmed if the background colors do not extend all the way to the boundary of the transposition domain. The technique is still working properly.

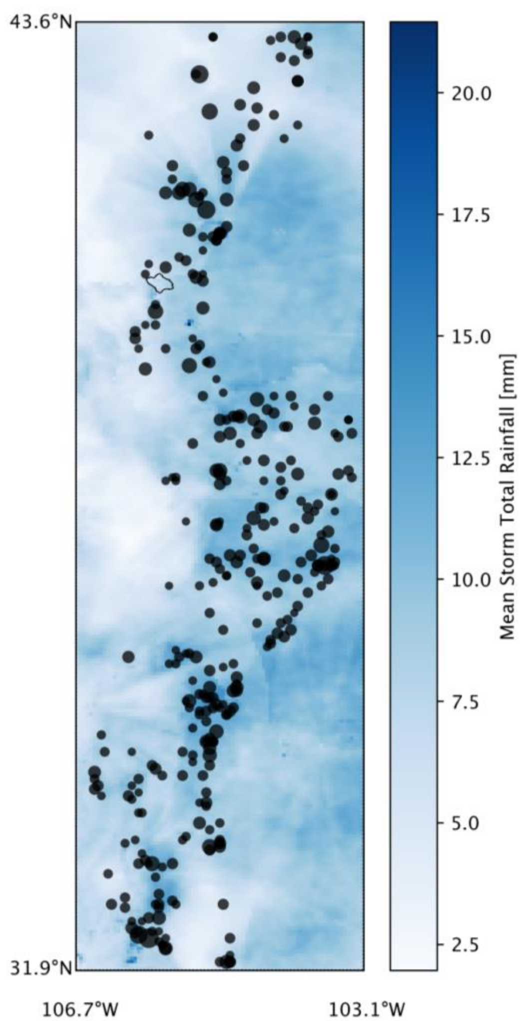


Figure 3: Average storm rainfall and location of storm centers. The black outline is the watershed (Big Thompson River upstream of Olympus Dam) used in this analysis

Figure 3 shows a plot a map of the average storm rainfall calculated from all storms in the storm catalog. It gives a picture of the regional climatology of heavy rainfall. The map also shows the locations of the storm centers from the storm catalog. Figure 4 is similar, except it shows the standard deviation of these storms. Other diagnostic plots may be included in the future.

FREQANALYSIS (optional): Create a ".FreqAnalysis" file based on the rainfall annual maxima generated. The "internal variability" will be calculated for each return period between 1 and NYEARS based on the spread in NREALIZATIONS. The default value is true.

SCENARIOS (optional): Create watershed-specific spacetime rainfall scenarios corresponding to each annual rainfall maxima. These scenarios can be used to drive a hydrologic model or landslide model. If you aren't planning on doing any modeling, there is no need to generate them. The code is currently created such that these scenarios are grouped by realization, so that there are NREALIZATIONS netCDF files created, each of which contains at most NYEARS scenarios. If RETURNTHRESHOLD is greater than 1 (highly recommended) the number of scenarios will be reduced, perhaps dramatically, depending on the specified return period threshold. The default value is false.

SPINPERIOD (optional): This only applies if SCENARIOS is set to "true." It provides the ability to "pre-pend" a rainfall period to the beginning of the output scenarios. Options are either "false" or an integer number of days. So if SPINPERIOD is set to 5, then five days of rainfall will be prepended to each scenario. Currently, the code is written such that the prepended rainfall time period for each scenario is drawn randomly from the rainfall that precedes the storms in the storm catalog from a period of one month earlier to one month later. For example, if SPINPERIOD is set to 5, and if a scenario is being written that is a transposition of a storm that occurred in July, then the preceding rainfall will be randomly selected from the 5-day rainfall periods preceding all storms in the storm catalog that occurred in June, July, and August. This "spin-up period" enabled by SPINPERIOD is potentially very useful, because it allows whatever hydrologic model is being used to achieve a realistic initial moisture condition prior to the onset of the main storm. In the case of floods, it also allows the establishment of realistic baseflow conditions. Caution should be used, however, since for large areas, SPINPERIOD could create very large arrays and there might be memory-related crashes.

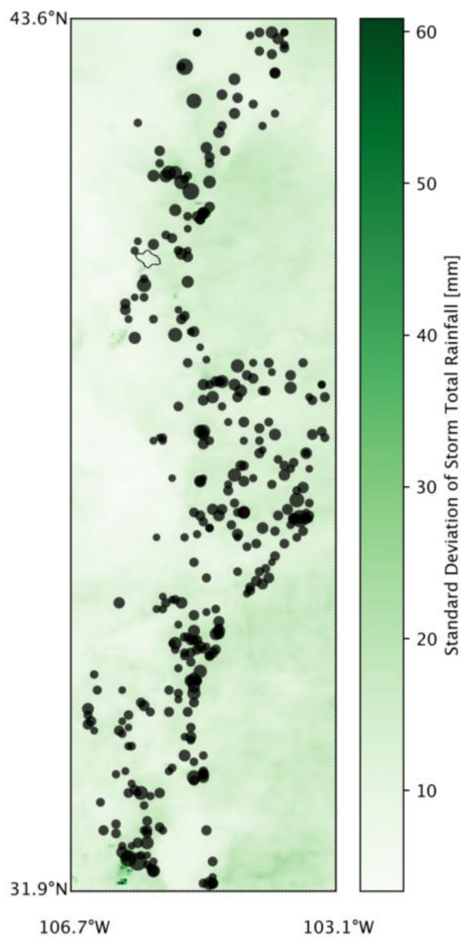


Figure 4: Standard deviation of storm catalog rainfall and location of storm centers. The black outline is the watershed (Big Thompson River upstream of Olympus Dam) used in this analysis.

RETURNTHRESHOLD (optional): Minimum return period for which a spacetime rainfall scenario will be written. This can dramatically reduce the number of (uninteresting) low intensity spacetime rainfall scenarios included in subsequent modeling. The default is to not use a threshold.

EXCLUDESTORMS (optional; highly recommended if using radar rainfall data): To exclude certain storms that are part of the storm catalog but which you do not wish to include in subsequent frequency analysis or spacetime rainfall scenarios. There are two main reasons for doing this: 1.) it allows you to examine the sensitivity of the method to the inclusion of particular storms, and 2.) it allows you to exclude storms that have unrealistic properties (easily diagnosed via the diagnostic plots). This is an important consideration when using a ground-based radar rainfall input dataset, in which radar artifacts can seriously affect the results. Multiple storms must be comma-separated. You can consult the diagnostic plots for the storm numbers.

EXCLUDEMONTHS (optional): Months (1-2 digit numeric) to be excluded from the storm catalog creation and/or subsequent frequency analysis and spacetime rainfall scenario generation. Months must be comma-separated. If you omit or comment out this line, all months will be used.

INCLUDEYEARS (optional): Years (4-digit numeric) to be included in the storm catalog creation and/or subsequent frequency analysis and spacetime rainfall scenario generation. Years can be either specified as a range, separated by "-" or as a comma-separated list. Examples: (2002-2005 is equivalent to 2002,2003,2004,2005). For the vast majority of analyses, you should use 'all', or simply omit/comment out this line.

RESAMPLING (optional): Defines the method used for generating the number of storms to be transposed for each synthetic "year." Options for RESAMPLING are either "poisson" (the default) or "empirical." If "empirical" is selected, then the generator will be a Poisson distribution. The rate parameter for the distribution will be estimated from the number of storms in the storm catalog divided by the number of years in the dataset. If "empirical" is selected, the generator is an empirical distribution determined by the counts per year in the storm catalog. Note that a minor statistical artifact may arise in regions with extremely rare rainfall, because there is a constraint that every synthetic year must have at least one storm. But for most situations this shouldn't be a problem. The option for the empirical distribution is useful in cases where the occurrence of extreme rainfall is over or under-dispersed. To evaluate this formally, you would compute the index of dispersion (the variance of the counts-per-year divided by the mean of the counts-per-year) and determine if it is significantly different from 1.0 via bootstrapping (or possibly some other technique).

TRANSPOSITION (optional): Defines how the random transposition should be done. Currently the options are "uniform" (the default) or "nonuniform." With "uniform," the resampled storms are randomly located according to a 2-D uniform distribution. With "nonuniform" the resampled storms are randomly located according to a 2-dimensional Gaussian kernel density estimate based on the locations of the original storms in the storm catalog. A general recommendation is only to use the kernel density method if you have a strong reason to believe it will provide better results than the uniform method. For example, if there are clear differences in storm locations based on topographic features, then the kernel option is preferable, but the estimated density is sensitive to small sample sizes from the input rainfall dataset. An example of the 2-D Gaussian kernel can be seen Figure 2.

UNCERTAINTY (optional): Defines the type of "uncertainty" you want to calculate for inclusion in the .FreqAnalysis file and the IDF figure. If you want the full ensemble spread for any exceedance probability, put "ensemble" (the default). If you want only the spread between the X/2 and100-X/2 quantiles, put "{X}". "ensemble" is recommended.

ROTATIONANGLE (optional): RainyDay allows for stochastic rotation of storms, though it should be noted that I coded several years ago. Since then, I have made lots of changes and additions to other parts of the code, and I have not thoroughly re-tested the rotation to make sure it is compatible with other changes and additions-so use this option at your own risk. There are a few things to note. First, this can be quite slow for large storm catalogs and for high-resolution rainfall data. Second, it should be used with caution, since in the real world, storm orientation is in many cases tightly tied to terrain such as topographic features, or tied to atmospheric phenomena such as high-pressure ridges. This means that rotation can produce storms that would be unrealistic in the real world. Nonetheless, it is a potentially useful feature when used in a restricted way. Omitting the line entirely or specifying "none" will prevent rotation. If rotation is desired, specify it in the following way "X,Y,Nangles". X must be less than zero and Y must be greater than zero. Nangles specifies the number of discrete "angle bins" between X and Y that will be used. The speed of the rotation scheme will depend greatly on the choice of Nangles, and relatively low values are recommended. Keeping values of X and Y relatively small (for example, X=-20, y=+20) will keep rotation more physically reasonable.

RETURNLEVELS: Specify certain return periods for which you want to produce output scenarios and frequency analyses.

STOCHASTICRESCALING: Omit or comment this out. It is currently experimental and not adequately tested.

RAINDISTRIBUTIONFILE: Omit or comment this out. It is currently experimental and not adequately tested.

POINTAREA: This defines the area that will be used in the calculation of rainfall accumulations. Options include "point" (in reality a single rainfall grid cell, rather than a point estimate), "box," or "basin." Point is not recommended, particularly for ground-based radar rainfall datasets, because it will be very sensitive to radar artifacts. "Basin" in principal provides the most precise estimates, but it is unlikely that it will produce very different results than those using "box" to approximate the size of the watershed. "box" is also useful for comparing results from rainfall datasets that have different input resolutions, as the procedure will aggregate the rainfall estimates to the size of the box. Note that if you wish to perform this aggregation, that it is wisest of the box is at least as large as the input resolution of the coarsest dataset.

WATERSHEDSHP: If "basin" is specified for POINTAREA, then either a ".mask" file or a basin boundary shapefile must be provided. WATERSHEDSHP provides the path and filename to the shapefile. RainyDay will convert this shapefile into a ".mask" array if necessary (see below). The rasterization is accomplished through the GDAL function "gdal_rasterize," which MUST be available through a command line terminal. To check, type "gdal_rasterize" in a terminal. The shapefile should contain only one watershed boundary and no other feature. The method is completely untested for multiple features, will likely crash, and will certainly produce poor results. Shapefiles must be in the geographic projection EPSG:4326 (i.e. a regular latitude/longitude grid).

BASEMAP (optional): I recommend not using this unless totally necessary, since it isn't well-tested. If "false," only a standard and relatively coarse map of land-water boundaries and national borders will be shown on the diagnostic plots and movies. If a path is provided to a shapefile, this shapefile will be loaded and displayed, assuming it is properly configured. A useful source of administrative boundaries for these basemaps is the GADM Database. You can download country-specific or a global shapefile. If you use a large shapefile (such as administrative boundaries for the globe or a large country such as the USA), however, it is strongly recommended that you export only the features you absolutely want, since the shapefile plotting is very slow in the matplotlib package. This exporting can be done using a separate GIS such as QGIS or ArcMap. One could also change the resolution of the basemap provided from matplotlib, but it is highly recommended that you not adjust the basemap resolution options in the source code, due to the very slow plotting speed of higher-resolution options. NOTE: if POINTAREA is set to "basin" and an appropriate basin boundary shapefile exists, it will also be plotted on these diagnostic plots.

BASEFIELD: If BASEMAP is used, then BASEFIELD should be set to the name of the appropriate field in the basemap shapefile.

POINTLAT: The latitude of the analysis point, if POINTAREA is set to "point." The program will find which pixel contains this point.
POINTLON: The longitude of the analysis point, if POINTAREA is set to "point." The program will find which pixel contains this point.

BOX_YMIN: The southernmost boundary of the analysis box, if POINTAREA is set to "box." The program will "snap" this box to the closest rectangular grouping of pixels.
BOX_YMAX: The northernmost boundary of the analysis box, if POINTAREA is set to "box." The program will "snap" this box to the closest rectangular grouping of pixels.
BOX_XMIN: The westernmost boundary of the analysis box, if POINTAREA is set to "box." The program will "snap" this box to the closest rectangular grouping of pixels.
BOX_XMAX: The easternmost boundary of the analysis box, if POINTAREA is set to "box." The program will "snap" this box to the closest rectangular grouping of pixels.
Caution is needed when comparing two different rainfall datasets with differing resolutions, since the snapping won't necessarily produce a box with the same dimensions.

SENS_INTENSITY (optional): This allows easy sensitivity analysis to rainfall intensity. Options are either "false" (no alteration of intensity) or a numeric percentage change. For example, an input of 5.0 means that the intensity of rainfall will be increased by 5%.

SENS_FREQUENCY (optional): This allows easy sensitivity analysis to rainfall frequency. Options are either "false" (no alteration of storm rate of occurrence) or a numeric percentage change. For example, an input of -15.0 means that the rate of storm occurrences will be reduced by 15%.

INTENSDISTR (optional): The user has the option of deriving the rainfall intensities from a GEV distribution (for example, based on a rain gage dataset). This option should be used with caution, particularly due to the spatial sampling differences between rain gages and remote sensing products. (NOTE: this option only applies the intensity sampled from the GEV distribution to the generated annual maxima. This means that it is not suitable for evaluating the space-time substitution effect via simulation. If this note makes no sense to you, don't worry about it!).

5. Running RainDay

The basics: This is the (hopefully) easy part, though, as I describe below, there are some subtle issues. At the command line, type:
python {full path if terminal is not currently in the same directory as the RainyDay script)}.py {full path to parameter file}.sst

The subtle parts: Like most any type of analytical tool, using RainyDay effectively requires an iterative process of refining the analysis and results. It is hard to summarize all the potential issues here, but I provide some recommendations.

Recommendation 1—Always run RainyDay at least twice: This is particularly true when using radar-based rainfall datasets such as NCEP Stage IV, since there can be "artifacts" due to beam blockage, brightband, etc. To start, set the following fields in the .sst file:

CREATECATALOG	true
DIAGNOSTICPLOTS	true
FREQANALYSIS	false
EXCLUDESTORMS	none

This analysis will often take some time, particularly for long datasets, or high-resolution datasets. Once it is finished, quickly examine all the diagnostic plots. Suppose Storms 150 and 324 appear to have unrealistic rainfall for some reason, but the other storms are reasonable. If you are otherwise satisfied with the storm catalog, the selected domain, etc., you can change those fields:

CREATECATALOG	false
DIAGNOSTICPLOTS	false
FREQANALYSIS	true
EXCLUDESTORMS	150,324

If you have done everything else well, the result of this second run will be your desired final outcome.

Recommendation 2—You probably want to use large storm catalogs: SST is prone underestimate more common recurrence intervals, particularly for short durations and for small storm catalogs. The reasons are simple: storms with short durations are spatially small; if storm catalogs are small, in any given stochastically-generated year, it is likely that no storm will impact the watershed/area of interest. This neither of these are realistic, they are the result of the procedure. One way to minimize these issues is to use large storm catalogs. For example, the default number of storms in the catalog is 20 per year, so a 20 year-long rainfall dataset will lead to 400 storms. If, on the other hand, you only care about really extreme storms, smaller catalogs are ok.

Recommendation 3—Use a longer-duration catalog to estimate short-duration IDF curves: There is a lot of interest in short-duration IDF curves. For example, infrastructure design might require a 3-hour duration IDF curve. This poses some problems for SST, mentioned above in Recommendation 2—specifically, short-duration storms are spatially small, which reduces the probability of a storm being transposed over the watershed or area of interest. Using large storm catalogs helps, but so does longer-duration catalogs. If you use a 72-hour catalog to generate a 6-hour IDF curve, it will actually produce more realistic results than using a 6-hour catalog. This is because of a specific duration correction that is coded into RainyDay. One easy way to do this: use the DURATIONCORRECTION option, which takes care of this for you.

Recommendation 4—Begin with a big transposition domain: Remember that you can always adjust your analyses later. Creating a storm catalog and diagnostic plots for a large transposition domain takes some time, but once finished, it can help diagnose regional rainfall heterogeneities and can help perhaps reduce the size of the domain.

Recommendation 5—Begin with a bigger "box" if you are hoping to do small-scale IDF analysis: Again, you can always adjust your analyses later. If you are interested in IDF analysis for small areas, such as a single rainfall grid cell, I nonetheless recommend generating a storm catalog based on a larger area. For example, when I used the Stage IV rainfall dataset, which has a spatial resolution of about 4 km, I always generate my storm catalog using 'POINTAREA box', for example with a box that is 0.25 by 0.25 degrees. Once the storm catalog is created, then I'll to using 'POINTAREA point' to generate the IDF estimates. This is particularly useful if you are using radar data that has unrealistic "speckles" of high-valued single grid cells, but I also think it just does a better job of identifying intense storms, event small scale ones.

Appendix A: Explanation of RainyDay NetCDF files

This section describes the two types of output files from RainyDay: storm catalogs, and "rainfall scenarios."

Storm Catalog	Rainfall Scenario
Purpose: contains all necessary rainfall information needed to conduct SST using RainyDay; basically it is a list of storms that have been identified through the current or a previous run of RainyDay, along with the full set of corresponding rainfall spacetime fields and dates/times	Purpose: contains all necessary rainfall information to drive a hazard model (hydrologic model, landslide model, etc.) with the output from the SST procedure. Note that one "realization" is written to each file, so if 'NREALIZATIONS' is greater than one, multiple rainfall scenario files will be produced. Each one will contain a sequence of
Dimensions: (N * T * Y * X) = NSTORMS * (DURATION ÷ input data temporal resolution) * ([LATITUDE_MAX - LATITUDE_MIN] ÷ input rainfall north-south spatial resolution) * ([LONGITUDE_MAX - LONGITUDE_MIN] ÷ input rainfall east-west spatial resolution)	Dimensions: (n * t * y * x) = NREALIZATIONS[a] * (DURATION[‡] ÷ input data temporal resolution) * (gridmask height ÷ input rainfall north-south spatial resolution) * (gridmask width ÷ input rainfall east-west spatial resolution)
Variables (dimensions): • 'rainrate' (N * T * Y * X): The spacetime rainfall fields for each of N storms that compose the storm catalog, each storm having T fields of size Y*X • 'time' (N*T): The dates and times for each rainfall field. For each of N storms, there are T entries[†] • 'latitude' (Y): Latitude[•], ordered north to south, of the y-coordinate of the rainfall spacetime fields • 'longitude' (X): Longitude[•], ordered east-to-west, of the x-coordinate of the rainfall spacetime fields • 'gridmask' (Y * X): Mask showing the accumulation shape that is used to identify the storms; could be a watershed outline, a rectangle, or a single pixel • 'domainmask' (Y * X): Mask showing the transposition domain; this will only be interesting if an irregular domain is used, rather than a rectangular one. • 'ylocation' (N): the north-south location that corresponds to the upper-left corner of the maximized rainfall of the size/shape of the gridmask for each storm • 'xlocation' (N): the east-west location that corresponds to the upper-left corner of the maximized rainfall of the size/shape of the gridmask for each storm • 'basinrainfall' (N): Average rainfall (in mm)	Variables (dimensions): • 'rainrate' (n * t * y * x): The spacetime rainfall fields for each of n storms that compose the storm catalog, each storm having t fields of size y*x • 'time' (n*t): The dates and times for each rainfall field. For each of n storms, there are t entries[†] • 'latitude' (y): Latitude[•], ordered north to south, of the y-coordinate of the rainfall spacetime fields • 'longitude' (x): Longitude[•], ordered east-to-west, of the x-coordinate of the rainfall spacetime fields • 'ylocation' (n): the randomly selected north-south location of the upper-left corner of the transposition that produces the each t*y*x rainfall spacetime series • 'xlocation' (n): the randomly selected east-west location of the upper-left corner of the transposition that produces the each t*y*x rainfall spacetime series • 'basinrainfall' (n): Average rainfall (in mm) over the size/shape of the gridmask for each transposition • 'returnperiod' (n): estimated rainfall return period for each transposition • 'stormnumber' (n): Which "member" of the storm catalog was used to generate each transposition

over the size/shape of the gridmask for each storm	

‡Note that T and t need not be the same. If RainyDay is run using an existing storm catalog, the user can define any t, as long as it is less than or equal to T.

†All date/time stamps correspond to the end of the accumulation period. For example, if a rainfall dataset has hourly resolution, then the date/time stamp 5/5/2005 12:00Z corresponds to the time period from 5/5/2005 11:00Z to 5/5/2005 12:00Z.

[a]If RETURNTHRESHOLD is greater than 1, then n will be less than NREALIZATIONS. This is a practical way to reduce the number of simulations you will run, and reduce the sizes of these rainfall scenario files.

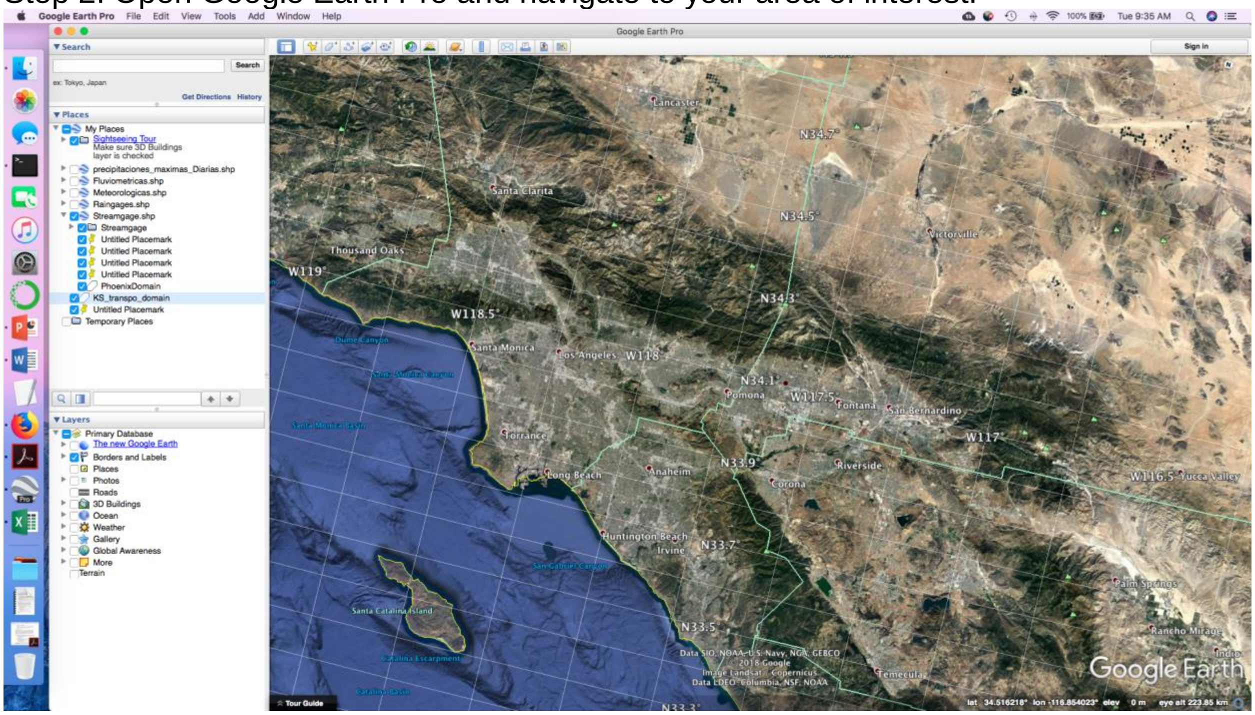
•Note that all latitude and longitude vectors relate the coordinates of the upper-left corner of the grid cell, not the lower-left or center, as some other coordinate systems do.

Appendix B: Creating a shapefile for a transposition domain by hand

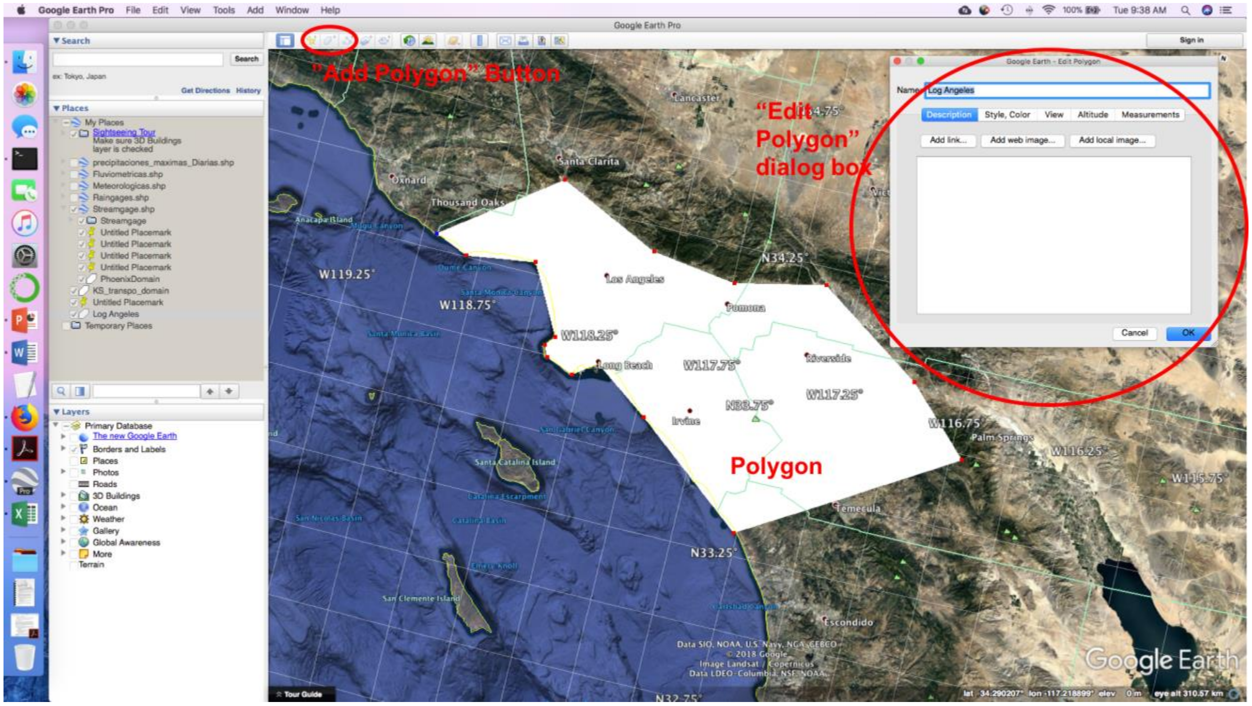
This appendix explains how to create a shapefile for the RainyDay transposition domain. One could also create a shapefile of the watershed outline or other area of interest for IDF computation using this approach (to be provided to the WATERSHEDSHP field in the .sst file), though it may not be the best way to do that. Note that the same result described here could be achieved using QGIS or ArcGIS. However, the following description is much simpler for those who do not have experience or access to such software.

Step 1: Download and install Google Earth Pro for desktop: https://www.google.com/earth/desktop/

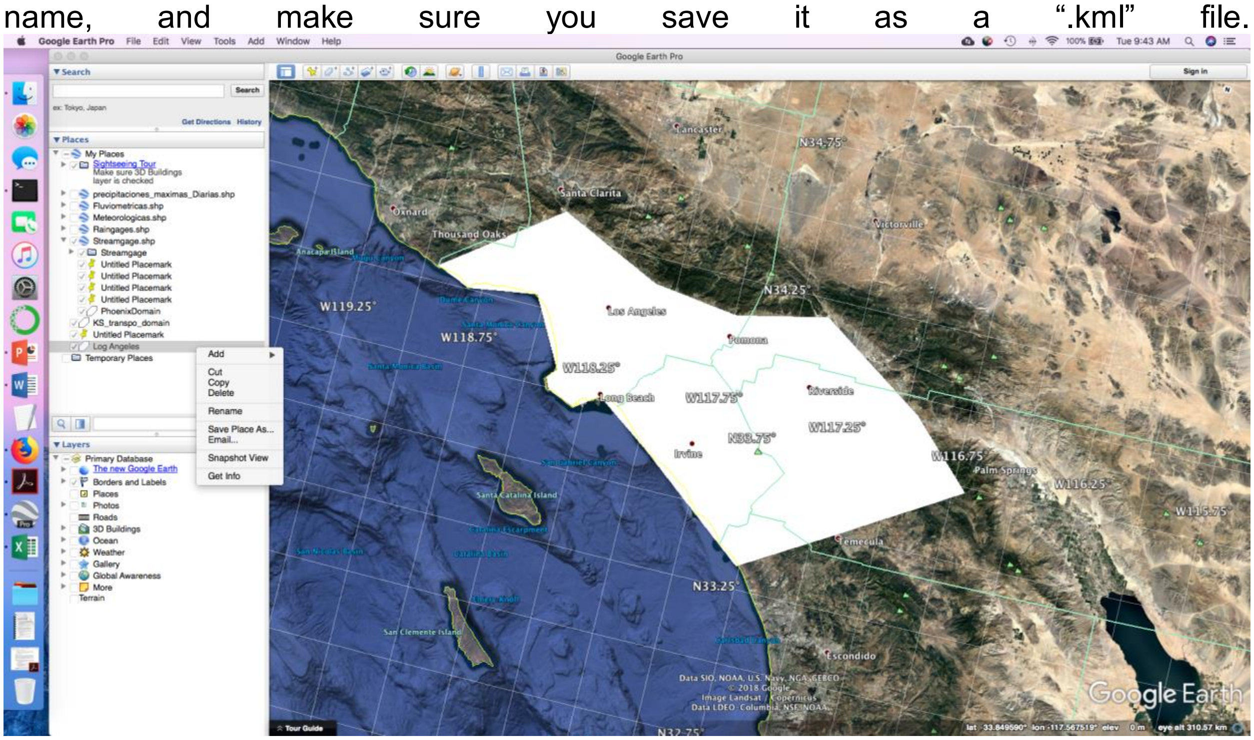
Step 2: Open Google Earth Pro and navigate to your area of interest:



Step 3: Click on the "Add Polygon" button, enter an appropriate name in the "Edit Polygon" dialog box that appears, and draw you're the boundary to define your desired transposition domain, and then click "OK" in the dialog box:

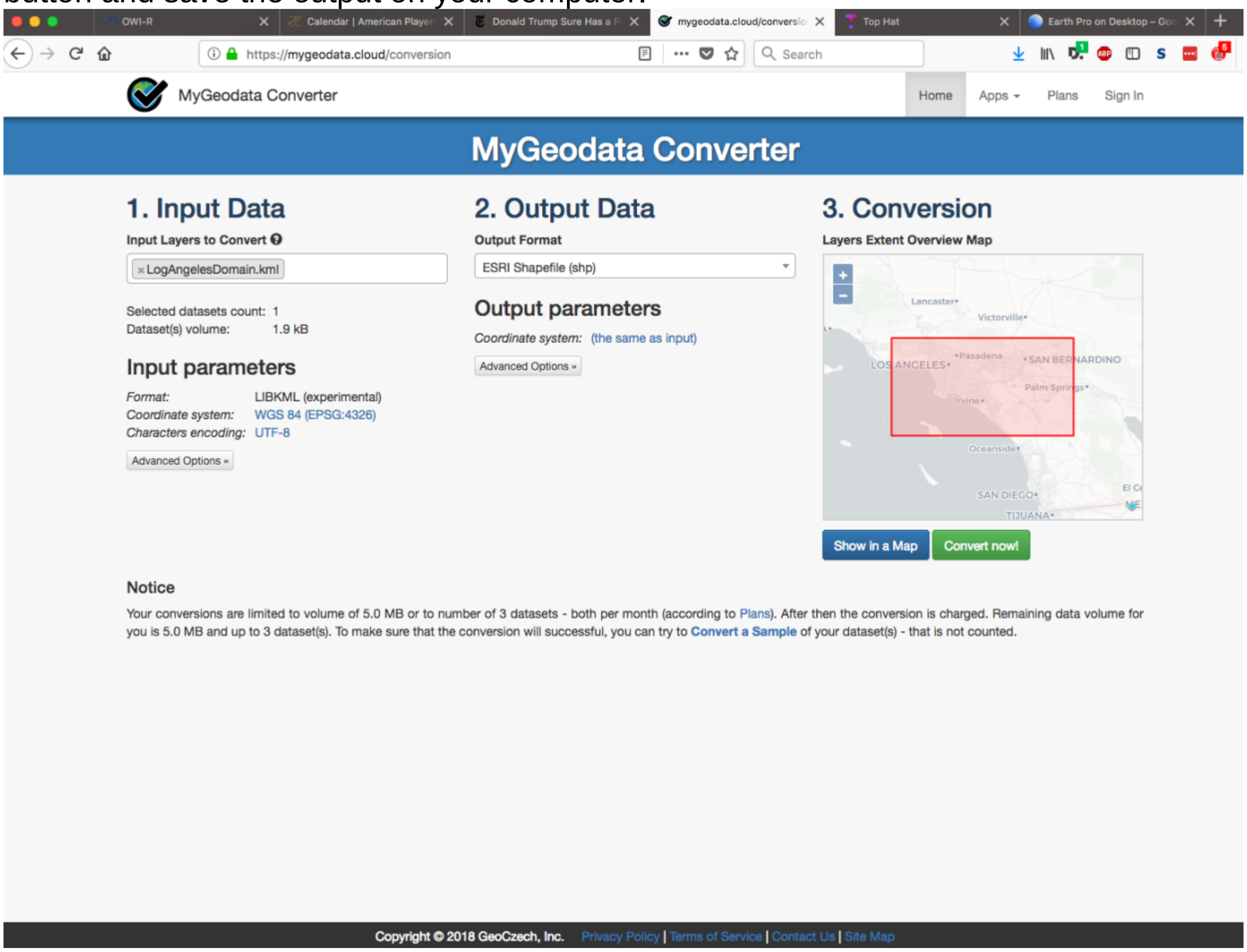


Step 4: You'll now see the polygon appear on the left under "Places". Now right click on the new polygon and select "Save Place As". Select an appropriate location and file name, and make sure you save it as a ".kml" file.



Step 5: In a web browser, navigate to https://mygeodata.cloud/converter/kml-to-shp. Upload the ".kml" file you created. Once you've done this, you will be taken to a new page, which displays information about your ".kml" file (including a map of the "bounding

rectangle") and the conversion to be performed. The most important details to note are that under "Input Parameters", you should see "Coordinate system: WGS 84 (EPSG:4326)". If not, something is wrong. Also, under "Output Parameters", you'll want to leave the default Coordinate System, which will be the same as the input file. Alternatively, if the input coordinate system is not WGS 84 (EPSG:4326), you can set the output to this coordinate system (I haven't tested this). Click the "Convert now!" button and save the output on your computer.



Step 6: You are now ready to use this shapefile in RainyDay. You will find that the website created a number of different files. The one you need to pay attention to is the one with the ".shp" file extension. That can be uploaded to the RainyDay webpage (http://her.cee.wisc.edu/projects/rainyday/) or provided to the RainyDay code via the .sst file using the following lines:

DOMAINTYPE	irregular
DOMAINSHP	{full file path to shapefile}/{shapefile name}.shp

Appendix C: RainyDay Example 1—Big Thompson Watershed

When you retrieve the RainyDay source code from Github (https://github.com/danielbwright/RainyDay2), you will find an example analysis, including a .sst file, storm catalog, diagnostics, and rainfall frequency analysis results. This example is for the watershed of Big Thompson River, upstream of Olympus Dam, in Northern Colorado. This example is provided for illustrative purposes only and the results are not intended to be a demonstration of RainyDay's or SST's accuracy or full capabilities, which are documented in peer-reviewed research papers.

The main purpose of this exam is to show how RainyDay can create watershed-scale IDF curves and rainfall scenario files, and can use transposition domains that have irregular shapes. See Appendix B for one way to create such a domain.

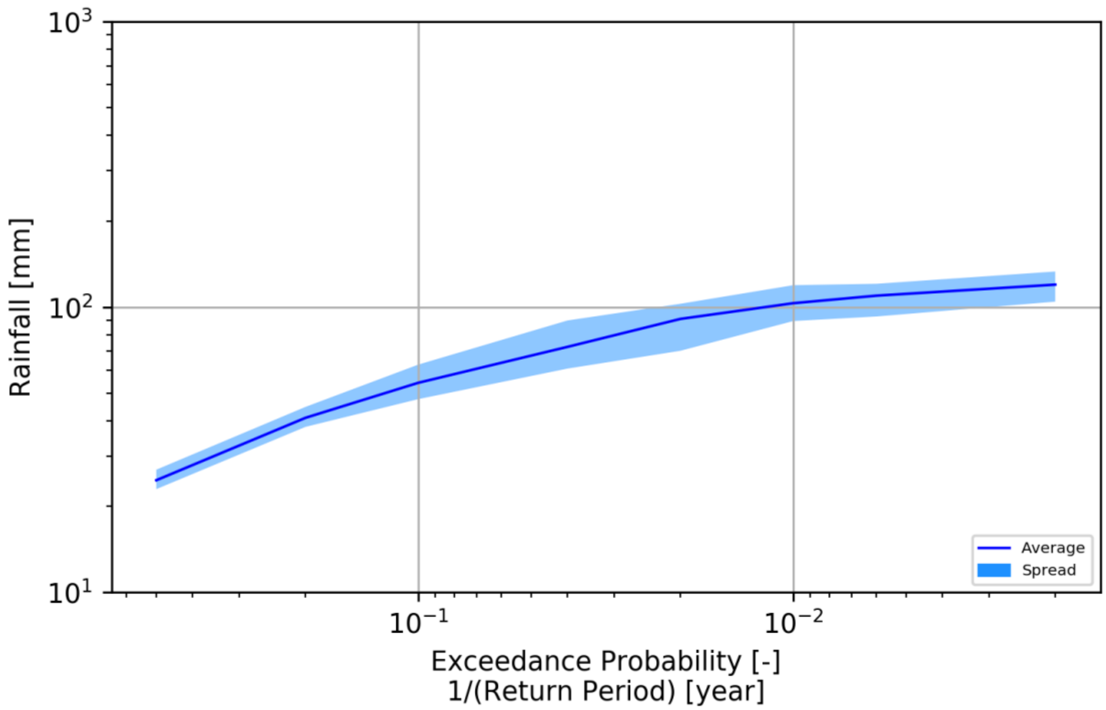


Figure C1: IDF Curve produced by RainyDay for the Big Thompson River watershed upstream of Olympus Dam, in northern Colorado. The curve is generated using Stage IV gage-corrected radar rainfall data, using the example .sst file and transposition domain that comes with the RainyDay source code.

Appendix D: RainyDay Example 2—Madison, Wisconsin IDF Curves

When you retrieve the RainyDay source code from Github (https://github.com/danielbwright/RainyDay2), you will find an example analysis, including a .sst file, storm catalog, diagnostics, and rainfall frequency analysis results. This example creates a pixel-scale IDF curve for Madison, Wisconsin. A key component of this is the DURATIONCORRECTION option.

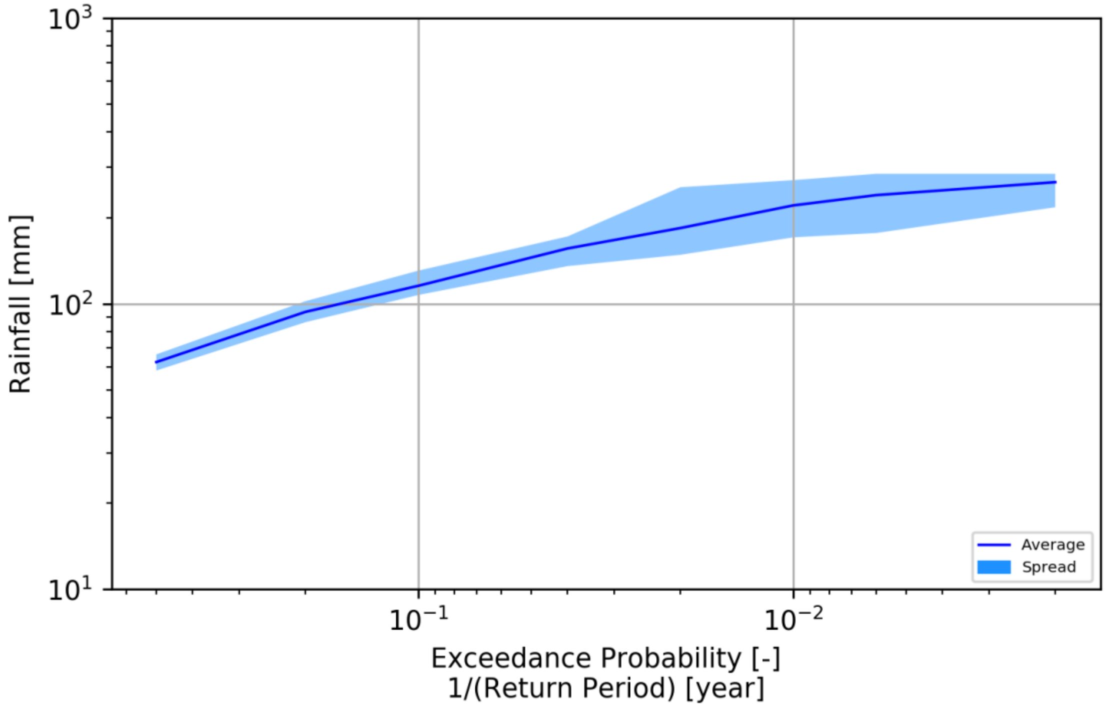


Figure D1: IDF Curve produced by RainyDay for a single radar grid cell in Madison, Wisconsin. The curve is generated using Stage IV gage-corrected radar rainfall data, using the example .sst file and transposition domain that comes with the RainyDay source code.