Supplementary materials to Angourakis et al. (2024)

Andreas Angourakis

01 August, 2024

Table of contents

1 Foreword **3**

2 Daily weather variables (summary statistics per day of year) in example locations around the planet between 01/01/1984 and 31/12/2018. **4**

I Demonstration of parameter effects **6**

3 Demostration of effects of parameter variation in the annual sinusoid curve **7**

4 Demonstration of effects of parameter variation in the annual double logistic curve, annual cumulative precipitation curve and year daily precipitation. **8**

5 Example of simulation outputs of the Weather model for 5 years **10**

II Calibration **13**

6 Calibration walkthrough **14**

6.1 Parameter estimation using `optim()` 14

6.2 Parameter estimation using Genetic Algorithms (`ga` package): 21

7 Calibration targeting weather examples **22**

7.1 Calculate weather example summary statistics 23

7.2 Estimation of year parameters 23

7.2.1 Estimation of hyperparameters based on weather dataset 23

1 Foreword

This file and all other referenced in the code can be found at the repository: https://github.com/Two-Rains/Weather-Angourakis-et-al-2024

(TO DO)

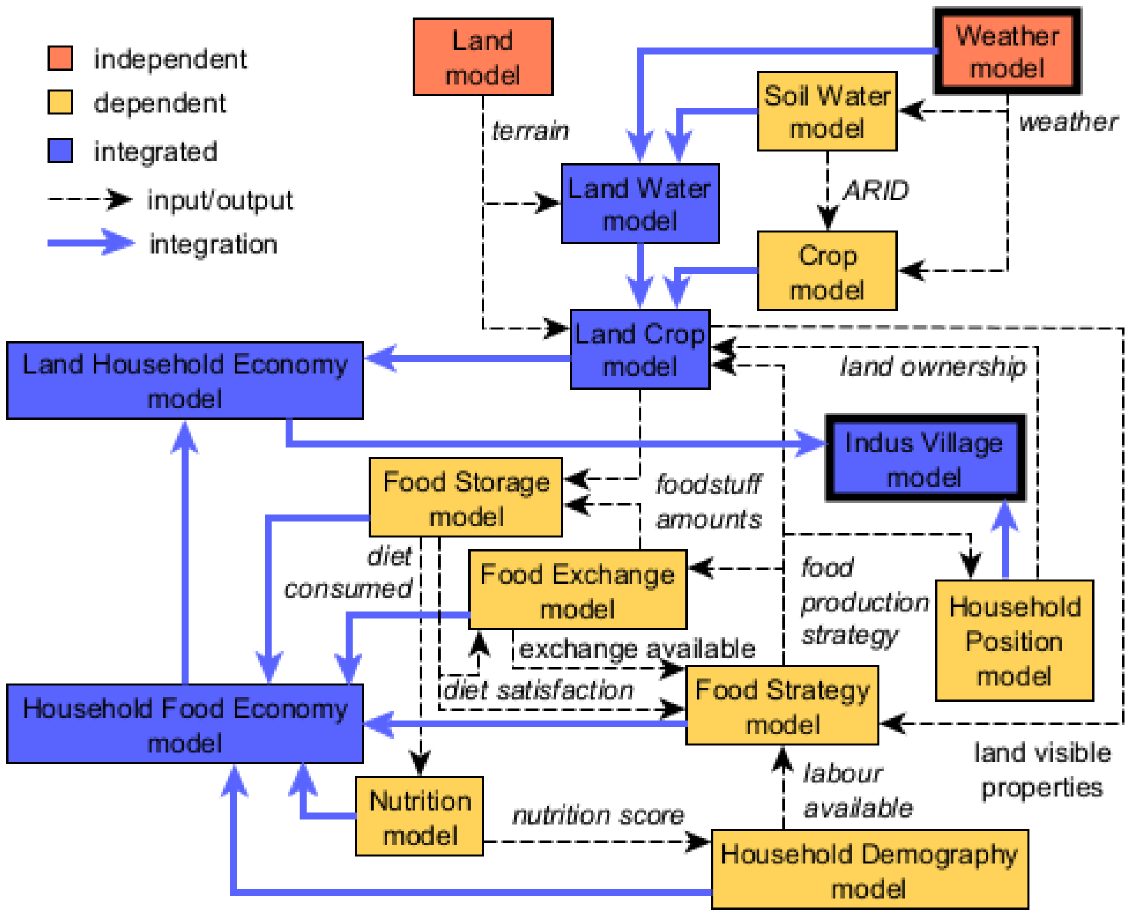


Figure 1.1: Place of the Weather model in the Indus Village

2 Daily weather variables (summary statistics per day of year) in example locations around the planet between 01/01/1984 and 31/12/2018.

Load source file containing the R implementation of the Weather model:

We use the data downloaded at NASA´s POWER access viewer (power.larc.nasa.gov/data-access-viewer/) selecting the user community 'Agroclimatology' and pin pointing the different locations between 01/01/1984 and 31/12/2007. The exact locations are:

- Rakhigarhi, Haryana, India (Latitude: 29.1687, Longitude: 76.0687)

- Irkutsk, Irkutsk Óblast, Russia (Latitude: 52.2891, Longitude: 104.2493)
- Hobart, Tasmania, Australia (Latitude: -42.8649, Longitude: 147.3441)
- Pearl Harbor, Hawaii, United States of America (Latitude: 21.376, Longitude: -157.9708)
- São Paulo, Brazil (Latitude: -23.5513, Longitude: -46.6344)
- Cambridge, United Kingdom (Latitude: 52.2027, Longitude: 0.122)
- Windhoek, Namibia (Latitude: -22.5718, Longitude: 17.0953)

We selected the ICASA Format's parameters:

- Precipitation (PRECTOT)
- Wind speed at 2m (WS2M)
- Relative Humidity at 2m (RH2M)
- Dew/frost point at 2m (T2MDEW)
- Maximum temperature at 2m (T2M_MAX)
- Minimum temperature at 2m (T2M_MIN)
- All sky insolation incident on a horizontal surface (ALLSKY_SFC_SW_DWN)

- Temperature at 2m (T2M)

and from Solar Related Parameters:

- Top-of-atmosphere Insolation (ALLSKY_TOA_SW_DWN)

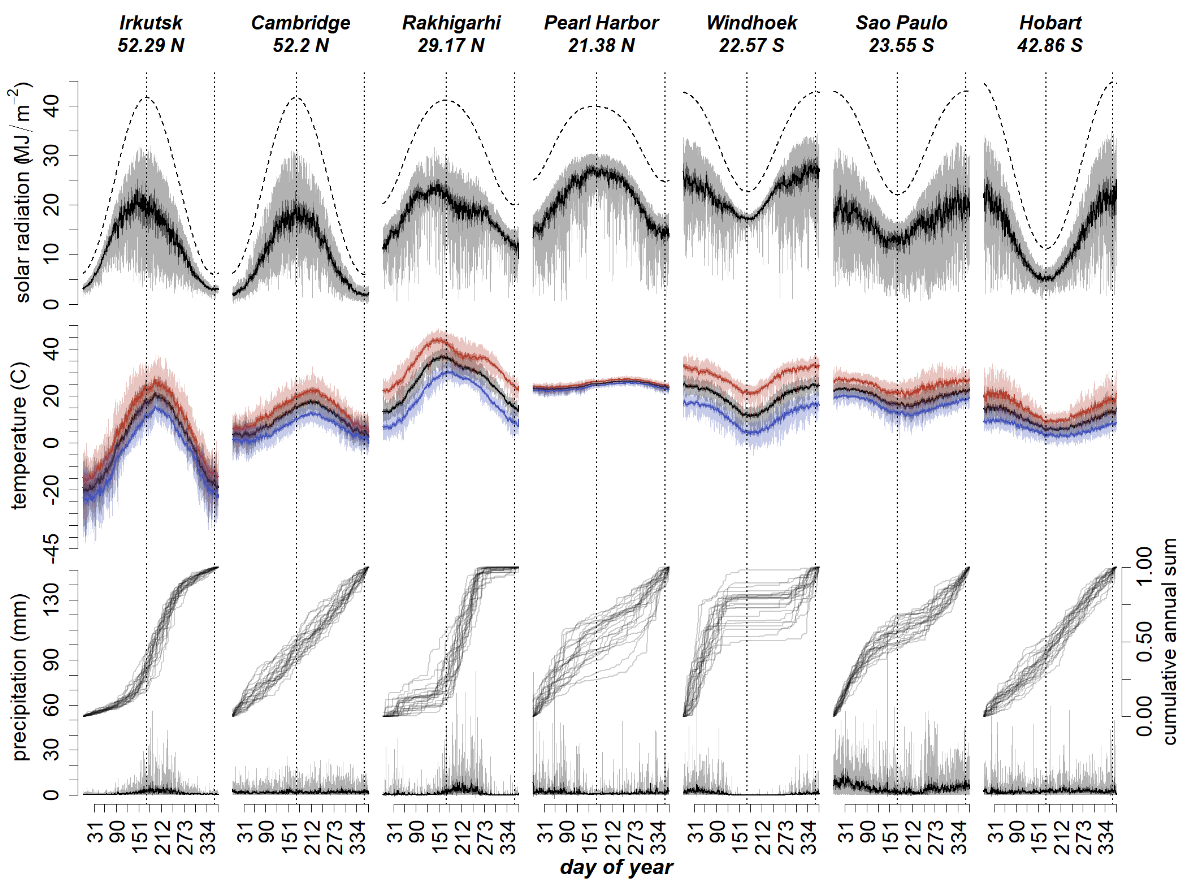
Compute statistics for each site and day of year:

Prepare display order according to latitude:

Set colours for maximum and minimum temperature:

Create figure:

```
pdf
  2
```



Part I

Demonstration of parameter effects

3 Demostration of effects of parameter variation in the annual sinusoid curve

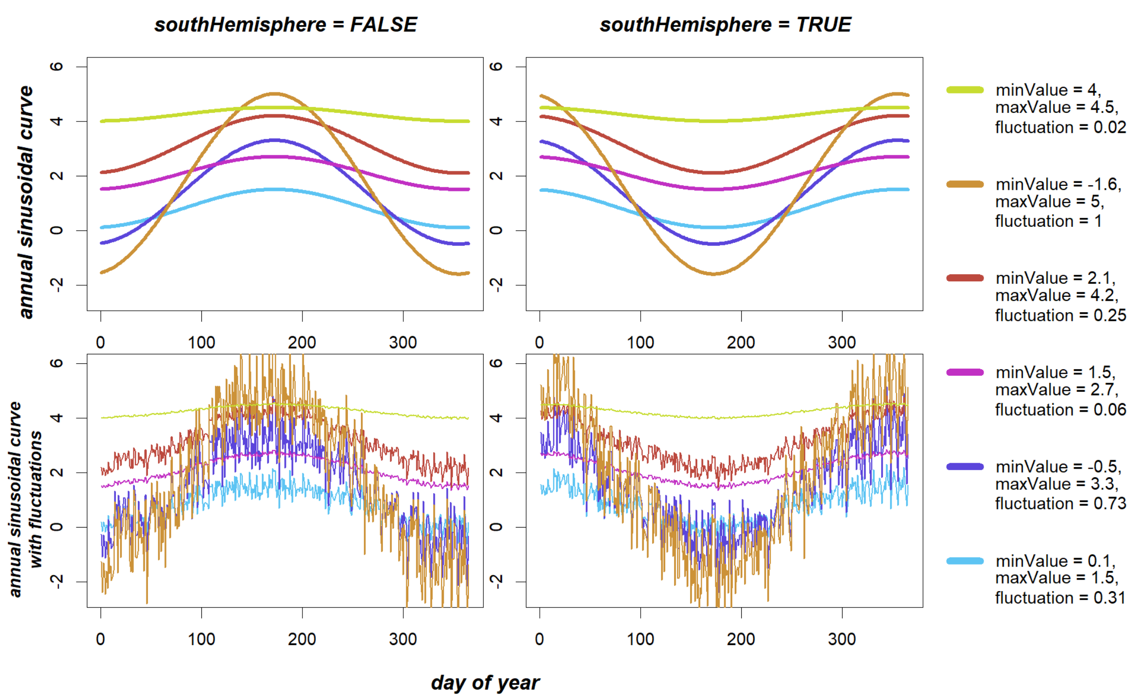
Load source file containing the R implementation of the Weather model:

Set up six variations of parameter settings (i.e. minValue, maxValue, southHemisphere), assuming length of year of 365 days:

Create a colour palette for plotting:

Plot curves:

```
pdf
  2
```



4 Demonstration of effects of parameter variation in the annual double logistic curve, annual cumulative precipitation curve and year daily precipitation.

Load source file containing the R implementation of the Weather model:

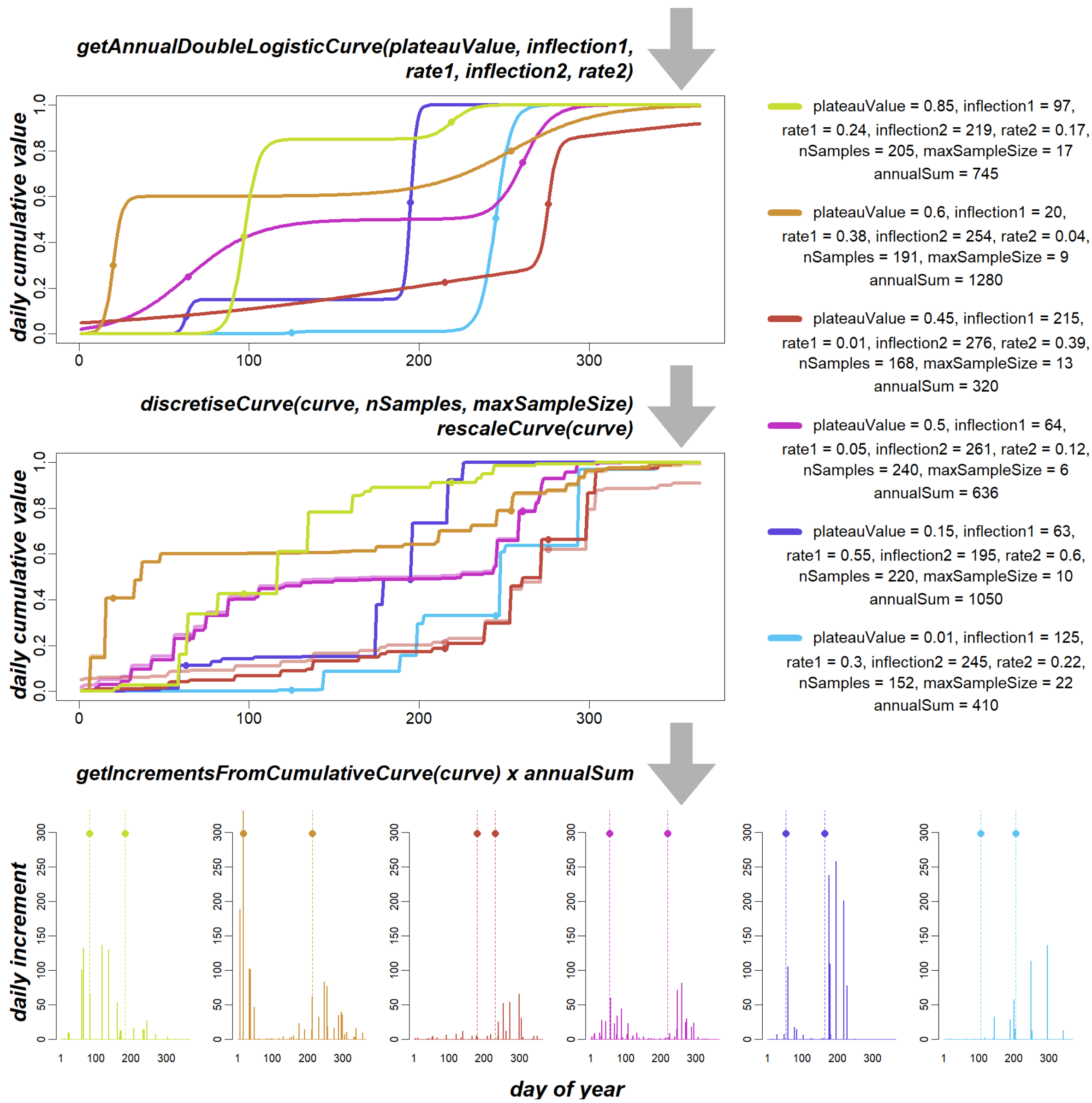
Set up six variations of parameter settings of the annual double logistic curve (i.e. plateauValue, inflection1, rate1, inflection2, rate2), the discretisation producing the annual cumulative precipitation curve (i.e. nSamples, maxSampleSize) and annualPrecipitation, assuming length of year of 365 days. Random generator seed used in discretisation is fixed:

Create a colour palette for plotting:

Initialise data frames for holding curves:

Plot curves:

```
pdf
  2
```



getAnnualDoubleLogisticCurve(plateauValue, inflection1, rate1, inflection2, rate2)
daily cumulative value
discretiseCurve(curve, nSamples, maxSampleSize)
rescaleCurve(curve)
getIncrementsFromCumulativeCurve(curve) x annualSum
daily increment
day of year
plateauValue = 0.85, inflection1 = 97, rate1 = 0.24, inflection2 = 219, rate2 = 0.17, nSamples = 205, maxSampleSize = 17
annualSum = 745
plateauValue = 0.6, inflection1 = 20, rate1 = 0.38, inflection2 = 254, rate2 = 0.04, nSamples = 191, maxSampleSize = 9
annualSum = 1280
plateauValue = 0.45, inflection1 = 215, rate1 = 0.01, inflection2 = 276, rate2 = 0.39, nSamples = 168, maxSampleSize = 13
annualSum = 320
plateauValue = 0.5, inflection1 = 64, rate1 = 0.05, inflection2 = 261, rate2 = 0.12, nSamples = 240, maxSampleSize = 6
annualSum = 636
plateauValue = 0.15, inflection1 = 63, rate1 = 0.55, inflection2 = 195, rate2 = 0.6, nSamples = 220, maxSampleSize = 10
annualSum = 1050
plateauValue = 0.01, inflection1 = 125, rate1 = 0.3, inflection2 = 245, rate2 = 0.22, nSamples = 152, maxSampleSize = 22
annualSum = 410

5 Example of simulation outputs of the Weather model for 5 years

Load source file containing the R implementation of the Weather model:

Initialisation using the default parametrisation, based on data from Rakhigarhi (example location, see Fig. 1):

Show table with parameter values:

parameter	values
seed	0
yearLengthInDays	365
albedo	0.4
southHemisphere	0
temperature - annualMaxAt2m	40
temperature - annualMinAt2m	15
temperature - meanDailyFluctuation	5
temperature - dailyLowerDeviation	5
temperature - dailyUpperDeviation	5
solar - annualMax	7
solar - annualMin	3
solar - meanDailyFluctuation	1
precipitation - yearlyMean	400
precipitation - yearlySd	130
precipitation - nSamples__yearlyMean	200
precipitation - nSamples__yearlySd	5
precipitation - maxSampleSize__yearlyMean	10
precipitation - maxSampleSize__yearlySd	3
precipitation - plateauValue__yearlyMean	0.1
precipitation - plateauValue__yearlySd	0.05
precipitation - inflection1__yearlyMean	40
precipitation - inflection1__yearlySd	20
precipitation - rate1__yearlyMean	0.15
precipitation - rate1__yearlySd	0.02
precipitation - inflection2__yearlyMean	200

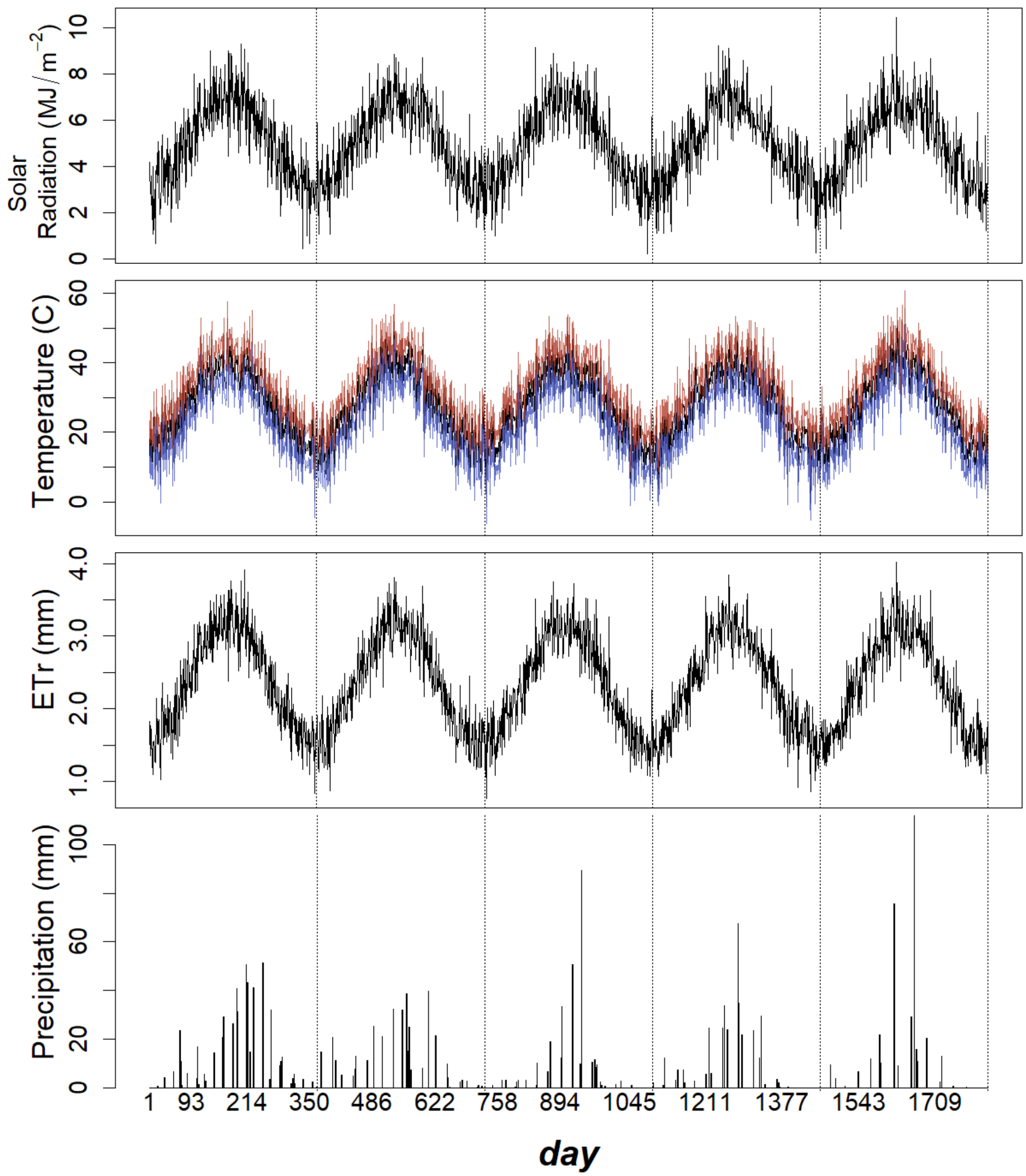
parameter	values
precipitation - inflection2__yearlySd	20
precipitation - rate2__yearlyMean	0.05
precipitation - rate2__yearlySd	0.01

Run for 5 years:

Set colours for maximum and minimum temperature:

Plot time-series:

```
pdf
  2
```


Solar Radiation (MJ/m^{-2})
Temperature (C)
ETr (mm)
Precipitation (mm)
1 93 214 350 486 622 758 894 1045 1211 1377 1543 1709
day

Part II

Calibration

6 Calibration walkthrough

Load source file containing the R implementation of the Weather model:

To generate new data based on a given dataset, we must first be able to estimate the Weather model parameters from said datasets. That is, to find the values of each parameter that can approximate the data of a given year daily series. Once this can be done for each year in the dataset, we can then estimate the hyperparameters as descriptive statistics (i.e., mean and standard deviation, minimum, maximum).

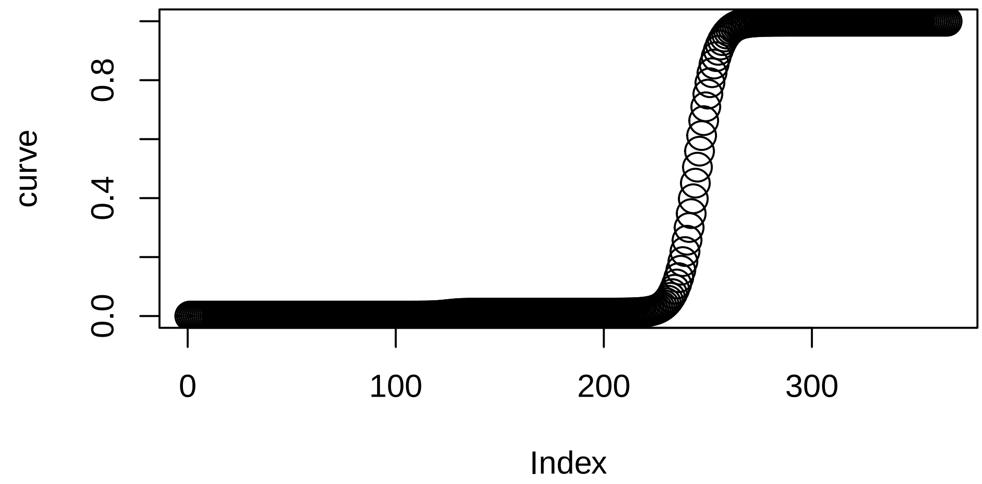
A good estimation of the parameters of the solar radiation and temperature submodels (i.e. sinusoid) can be made directly by measuring the year minimum and maximum.

However, the case of precipitation is far from trivial, given the complexity of the algorithm behind it. The workflow to estimate the parameters of the precipitation submodel deserves a demonstration.

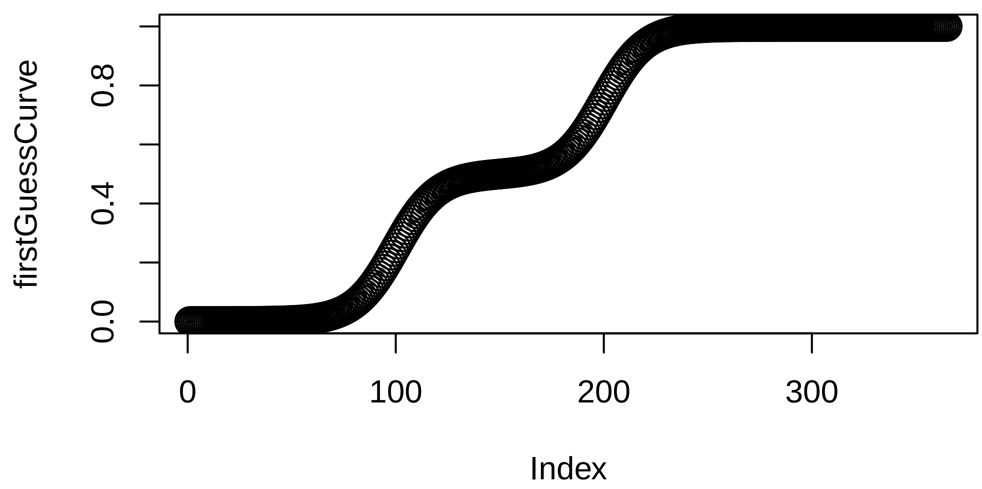
6.1 Parameter estimation using `optim()`

Set up six variations of parameter settings of the annual double logistic curve (i.e. plateauValue, inflection1, rate1, inflection2, rate2), the discretisation producing the annual cumulative precipitation curve (i.e. nSamples, maxSampleSize) and annualPrecipitation, assuming length of year of 365 days. Random generator seed used in discretisation is fixed:

Select the first set of parameter values from the `parValuesDoubleLogistic` dataset and generate the corresponding curve with the `getAnnualDoubleLogisticCurve()` function. These points will represent the original state of the model that we aim to reverse engineer from the outcome curve. Plot it.



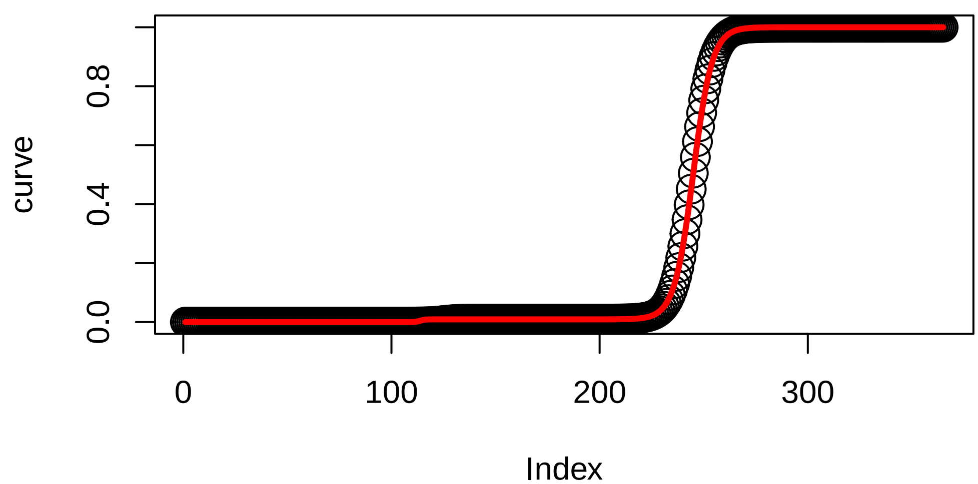
Define the `initialGuess` vector with your initial parameter guess values. Generate the curve using the `getAnnualDoubleLogisticCurve()` function with the initial guess. Plot it. Notice that our initial guess generates a somewhat "average" cumulative curve.



Define the `objectiveFunc()` function that calculates **the sum of squared differences between the observed data and the predicted values**, generated by the `getAnnualDoubleLogisticCurve()` function with a given parameter setting. Then, use the `optim()` function to estimate the best parameter values by minimizing the objective function.

NOTE: `optim()` using method "L-BFGS-B", see `?optim` or: > Byrd, R. H., Lu, P., Nocedal, J. and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing, 16, 1190–1208. doi:10.1137/0916069.

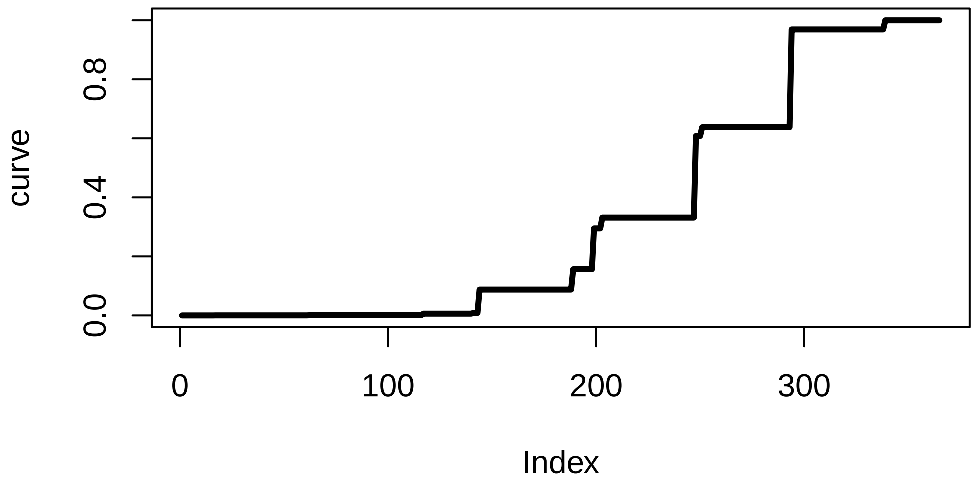
Plot the original curve (curve) and overlay it with the curve generated using the best estimated parameter values (bestEstimationCurve). The best estimated curve is shown in red.

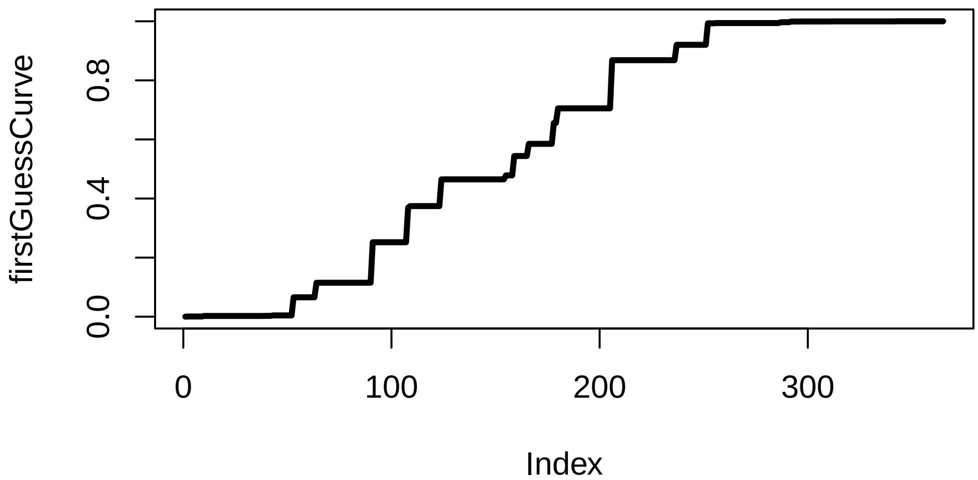


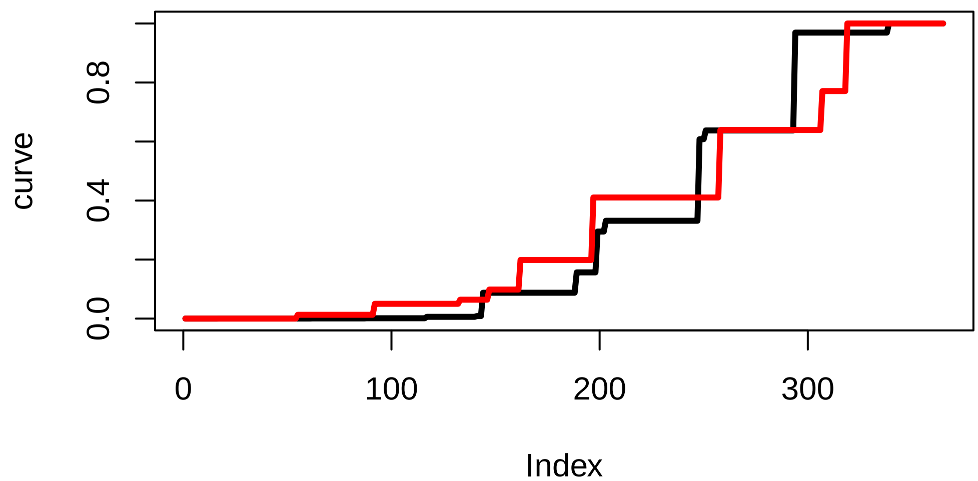
parameter	original	estimated	delta
plateauValue	0.01	0.00909982142182281	9e-04
inflection1	125	113.795551408398	11.204449
rate1	0.3	0.89998427831094	0.599984
inflection2	245	244.987905641175	0.012094
rate2	0.22	0.219538135299713	0.000462

We can see that reverse engineering the parameter values of the double logistic curve is relatively straightforward. However, precipitation in the Weather model presents an additional challenge: the continuous cumulative curve is broken down into "steps" through `discretiseCurve()`, which introduces **stochasticity**. We will also add `rescaleCurve()` to the end of the process, in order to approach the curve that would be created by `getPrecipitationOfYear()`.

Let us extend the workflow used above with `getAnnualDoubleLogisticCurve()` to also cover the two additional parameters of `discretiseCurve()` (for now, fix `seed = 0`):







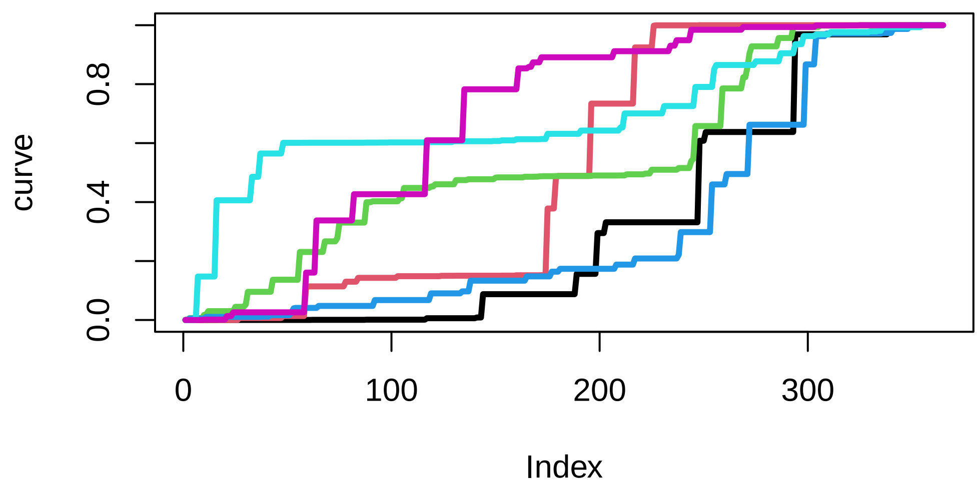
parameter	original	estimated	delta
plateauValue	0.01	0	0.01
inflection1	125	100.289790982745	24.710209

parameter	original	estimated	delta
rate1	0.3	0.01	0.29
inflection2	245	258.301146051514	13.301146
rate2	0.22	0.9	0.68
nSamples	152	270.02228671593	118.022287
maxSampleSize	22	30	8

Close, but a much worse fit than obtained with `getAnnualDoubleLogisticCurve()` only. We should take this performance in consideration going forward.

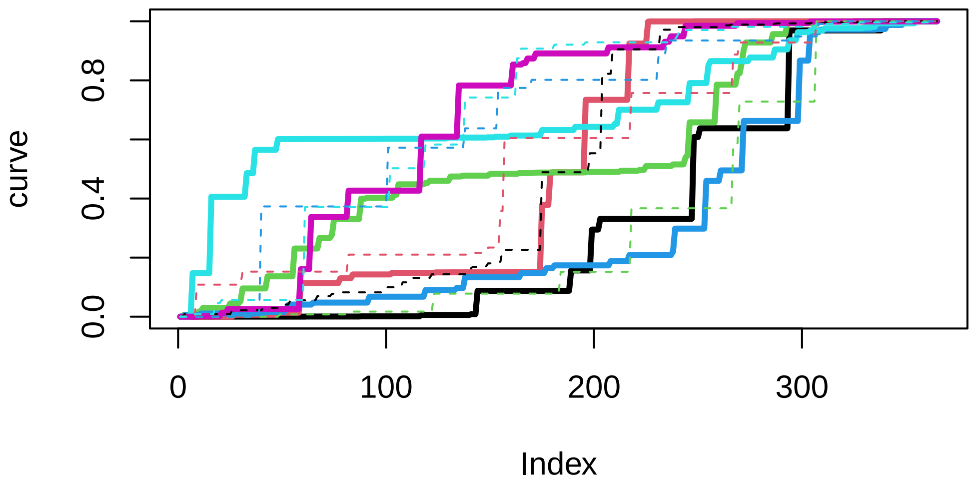
Let us now apply the same workflow for estimating the hyperparameters able to generate an approximation of a sequence of year daily series.

First, generate the original dataset based on the different configurations present in `parValuesDoubleLogistic` and `parValuesDiscretisation`:



Apply `optim`, reusing `initialGuess` and `objectiveFunc`, to each curve and generate a sequence of best estimation curves:

Plot original and estimated curves:



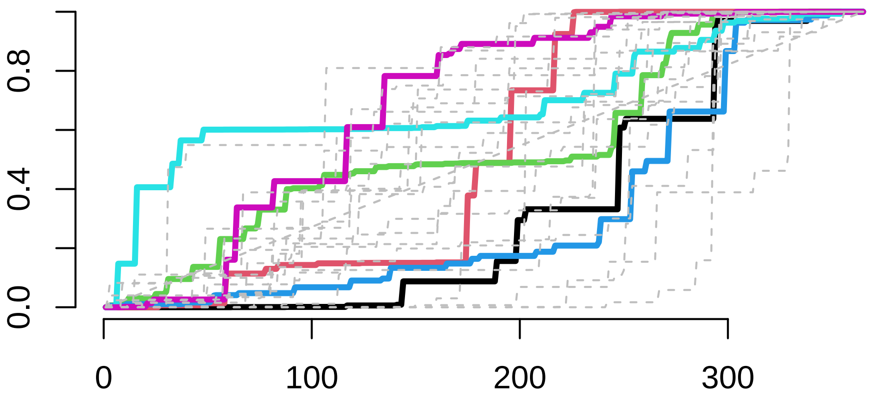
Visualise the aggregate estimation quality:

parameter	original (mean)	original (sd)	estimated (mean)	estimated (sd)	delta (mean)	delta (sd
plateauValue	0.4266667	0.3050683	0.3599647	0.4221796	0.066702	0.11711
inflection1	97.3333333	67.6481091	83.9851037	38.6891002	13.348230	28.95900
rate1	0.2550000	0.2034453	0.3079023	0.4586454	0.052902	0.25520
inflection2	241.6666667	29.6895043	231.6966633	53.9317743	9.970003	24.24227
rate2	0.2566667	0.2050041	0.3511250	0.4356376	0.094458	0.23063
nSamples	196.0000000	32.6741488	211.4276582	42.7020308	15.427658	10.02788
maxSampleSize	12.8333333	5.8452260	25.1674283	8.0057919	12.334095	2.16056

Define the hyperparameters of a weather model instance based on the mean and standard deviation of the best estimation parameter values:

Run the model to generate a number of cumulative curves:

Plot original and generated curves:



6.2 Parameter estimation using Genetic Algorithms (`ga` package):

TO DO

7 Calibration targeting weather examples

Load source file containing the R implementation of the Weather model:

As a final part in this demonstration, we will extend the above process to deal with multiple instances of curves and parameter sets, generated by the same configuration of hyperparameters. We will then want to estimate those original hyperparameter values.

We use the data downloaded at NASA´s POWER access viewer (power.larc.nasa.gov/data-access-viewer/) selecting the user community 'Agroclimatology' and pin pointing the different locations between 01/01/1984 and 31/12/2007. The exact locations are:

- Rakhigarhi, Haryana, India (Latitude: 29.1687, Longitude: 76.0687)

- Irkutsk, Irkutsk Óblast, Russia (Latitude: 52.2891, Longitude: 104.2493)
- Hobart, Tasmania, Australia (Latitude: -42.8649, Longitude: 147.3441)
- Pearl Harbor, Hawaii, United States of America (Latitude: 21.376, Longitude: -157.9708)
- São Paulo, Brazil (Latitude: -23.5513, Longitude: -46.6344)
- Cambridge, United Kingdom (Latitude: 52.2027, Longitude: 0.122)
- Windhoek, Namibia (Latitude: -22.5718, Longitude: 17.0953)

We selected the ICASA Format's parameters:

- Precipitation (PRECTOT)
- Wind speed at 2m (WS2M)
- Relative Humidity at 2m (RH2M)
- Dew/frost point at 2m (T2MDEW)
- Maximum temperature at 2m (T2M_MAX)
- Minimum temperature at 2m (T2M_MIN)
- All sky insolation incident on a horizontal surface (ALLSKY_SFC_SW_DWN)
- Temperature at 2m (T2M)

and from Solar Related Parameters:

- Top-of-atmosphere Insolation (ALLSKY_TOA_SW_DWN)

7.1 Calculate weather example summary statistics

Compute statistics for each site and day of year:

Prepare display order according to latitude:

7.2 Estimation of year parameters

7.2.1 Estimation of hyperparameters based on weather dataset

Declare auxiliary objects for estimating the precipitation cumulative curve with `optim`:

Test an isolated version of the estimation of cumulative precipitation hyperparameters using `optim`:

Run estimation of cumulative precipitation hyperparameters for all sites:

Calculate yearly summary statistics matching parameter inputs for each example location:

Initialise experiments using annual summary statistics and estimated yearly cumulative precipitation parameters of example locations as parameter inputs:

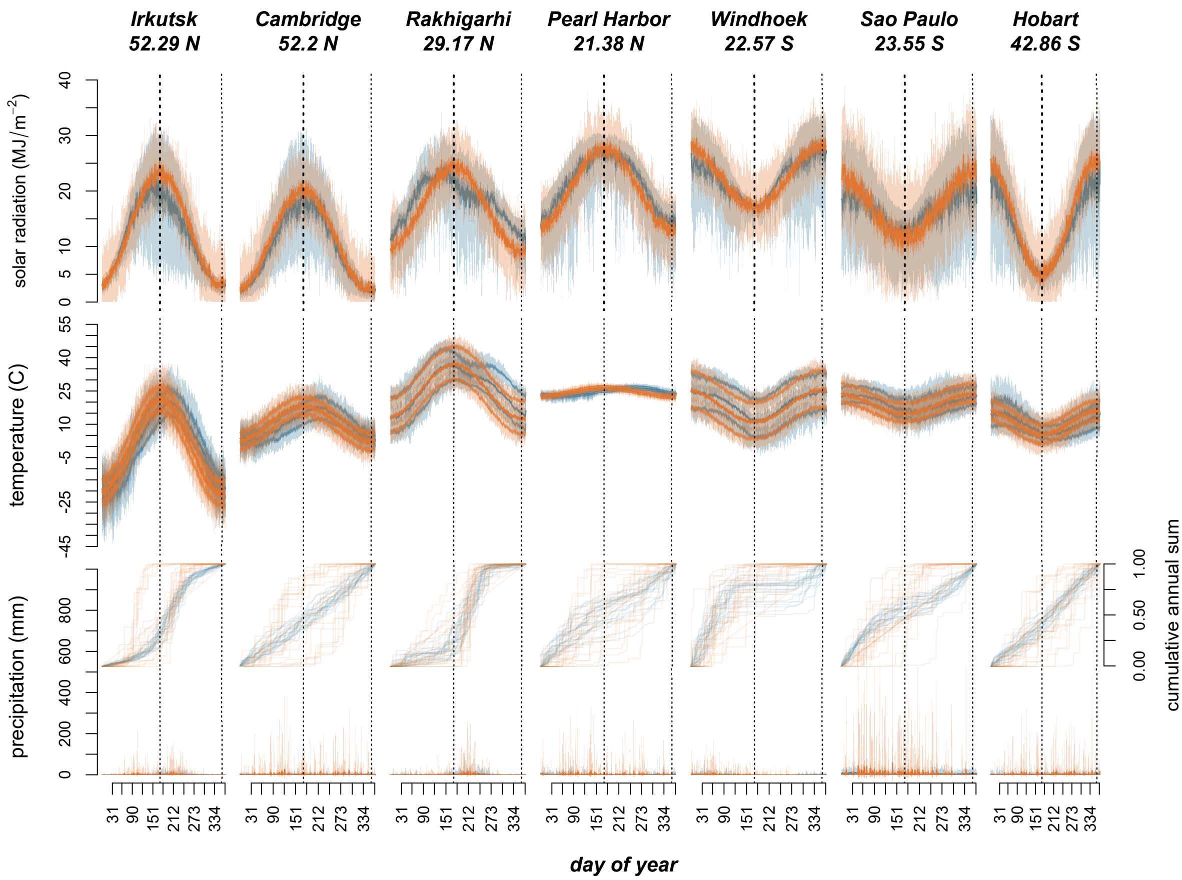
Run experiments:

Create a data frame containing the daily summary statistics of simulations comparable to the one for the real data (created for Fig 1 above):

Set colours for real and simulated data:

Create figure:

```
pdf
  2
```


Irkutsk
52.29 N
Cambridge
52.2 N
Rakhigarhi
29.17 N
Pearl Harbor
21.38 N
Windhoek
22.57 S
Sao Paulo
23.55 S
Hobart
42.86 S
solar radiation (MJ/m^{-2})
temperature (C)
precipitation (mm)
cumulative annual sum
day of year