A stochastic rain generator for vegetation modeling

Maik Billing
maik.billing@gmx.de

May 23, 2017

Contents

1 Abstract

A stochastic rain generator was developed to generate daily precipitation time series at one site. The generator combines different approachs of using general linear models (GLM) [1] and the method of monthly fragments [2, 3] for stochastic generation of annual, monthly and daily rainfalls. Major components and functions of the generator will be explained and introduced. A case study demonstrates the proper functioning of the generator.

2 Methods

2.1 Fundamental functionality and principles

The generator is based on a hierarchical top-down three level structure comprising an annual, monthly and daily level. (see figure 1)
Annual level
The data generation starts at the annual level. According to the number of generated years, new annual precipitations are randomly drawn from a Gaussian distribution. The parameters of that Gaussian distribution are the same μ and σ of the annual precipitation of the historical record. These generated annual precipitations are used as input for the monthly level.
Monthly level
The monthly rainfalls are calculated based on the method of monthly fragments by Srikanthan and McMahon [2] and Porter and Pink [3]. For each year of the record, monthly rainfalls were divided by their annual rainfalls resulting in a set of fragments for each year and annual

precipitation. Each drawn annual precipitation is then multiplied by the set of fragments which lies closest to that drawn annual value. This results in 12 new monthly rainfalls for each generated annual rainfall.

Daily level

The daily precipitation is generated by using a rainfall occurrence model and a general linear model (GLM) to predict the precipitation amount on wet days based on the approach of Stern and Coe [1]. The generated monthly precipitations are used as input for the daily level, since the predicted means of both models depend on the monthly precipitation. Rainfall occurrence is modeled by a two-state Markov chain (logistic regression) that predicts the probability of rainfall on a certain day and thus if the day is wet or dry.

Precipitation amounts were drawn from a right tailed distribution. Before the data generation starts, a chi-squared test for 3 different distributions (gamma, lognormal, weibull) is performed to determine which distribution fits best to the historical record. The distribution with the lowest chi-squared value is used to model the rainfall amounts. Parameters of the right tailed distribution may change through the year. For the sake of simplicity, it is assumed that the shape parameter k or σ stays constant through all years, but changes in the mean μ and thus in the scale parameter are allowed. The shape parameter is estimated in the beginning by fitting the distribution to the whole data set. The mean μ is predicted by a GLM that considers the monthly rainfall.

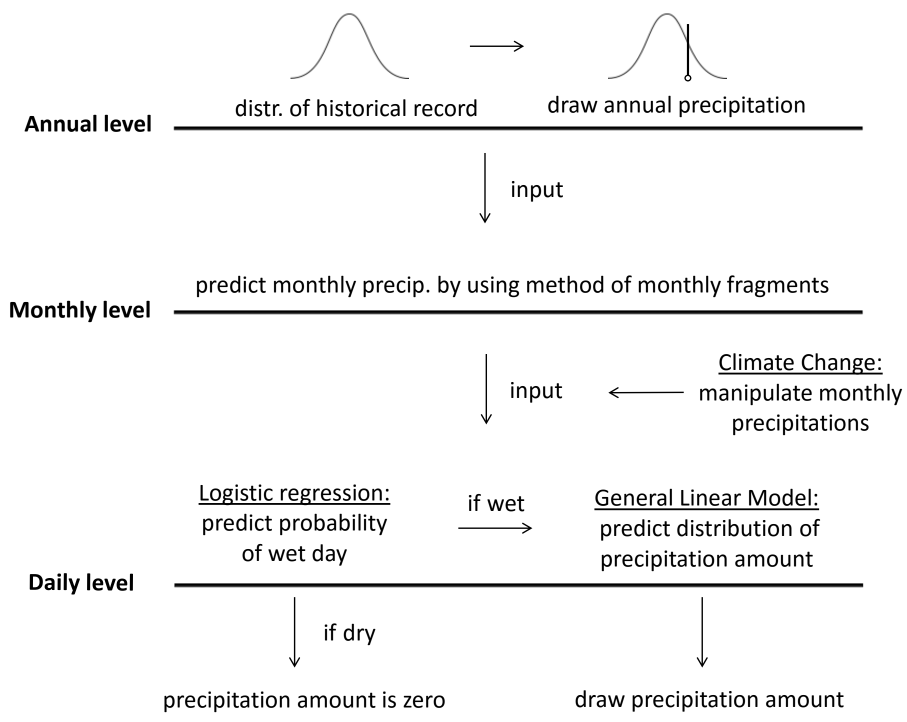

Figure 1: The general 3-level structure of the rain generator.

2.1.1 Climate change scenarios

The generator is able to generate data with mean increased or reduced monthly rainfalls over time. This allows to include long term intra-annual changes in the generated data, that can be used for climate scenario modeling. The effects of climate change are considered in between the monthly and daily level: In case of climate change the calculated monthly precipitation are modified by a certain percentage and passed to the daily level thereafter.

3 Functions of the generator

The code of the rain generator is written in R (version 3.2.2) and is divided into four functions. Each function can be found in a separate R-file. The historical data needs be on a daily resolution and must be provided in a data frame (data) with 4 columns that contain numeric values only. The columns must be named as follows: 'year', 'month', 'day', 'pre'. (pre=daily precipitation amount). Subsequently, the functions and their arguments were explained.

3.1 estimate.models()

In order to get reasonable results from the generator, it is necessary to adjust the model predictors for the logistic regression and the GLM to predict rainfall amounts for a location at once. Therefore the function 'estimate.models(data)' is provided, which determines the best predictors for both models by using a stepwise model selection based on the Bayesian information criterion (BIC).
The function needs the original record as input (data). It delivers the model formulas as a character string in a list (list members: $wet,$amount), which can be used directly as input for 'estimate.bias()' and 'raingen()'. Possible predictors include: monthly precipitation, square root of monthly precipitation, their interaction, the number of wet days in month (only for precipitation amount) and their interaction with the monthly precipitation.

3.2 estimate.bias()

Often stochastic weather generators tend to under- or overestimate interannual variability of the generated data [4, 5, 6]. To face this problem, it is often required to apply bias correction techniques, such as the delta method. In this method a change signal (delta) is added to all observations. Then, this changed observations are used to calibrate the model and the biased output of the model is corrected. In the case study, results of the bias uncorrected rain generator often show an increased variability in the annual rainfall amounts. This can be corrected by reducing the variance of the normal distribution that is used on the annual level for drawing the annual values.
The function 'estimate.bias()' estimates the degree of the biased variance of the generated annual rainfalls iteratively: Data is generated with zero bias in the first step. In each step, the bias adjusted by the difference of the variance of generated data and historical data.
This function needs the record (data) as input and the model equations (see 3.1). The Minimum recommended number of iterations is 10 (default: 15) and the number of simulated years in each step should not be below 500 (default: 1000).

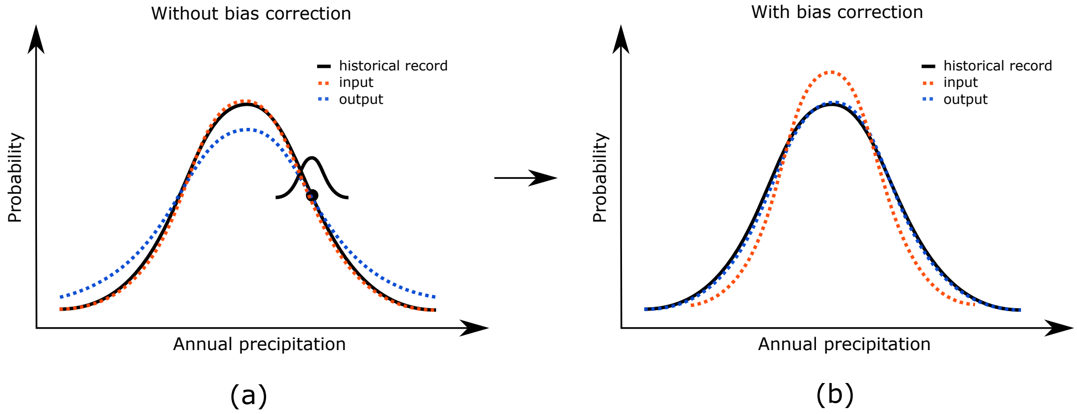

Figure 2: Illustration of bias correction: (a) Let us consider one single drawn annual precipitation based on the historical record (black dot). For that precipitation the generator generates a distribution of annual values due to its stochastical intrinsic characteristics (e.g. a normal distribution around the drawn value). For this case, the probability to generate values that are higher or lower than the drawn value is exactly equal. This results in a broader distribution of the annual precipitations in the output (blue dashed line) and the variance of the generated annual precipitation is systematically overestimated. (b) Applying the 'delta method' shrinks the original variance of the input (red dashed line), so that the generated precipitations fit better to the historical record. The mean is not effected.

3.3 raingen()

The core function of the generator produces daily rainfall data based on a three level structure and requires at least the historical record (data) and the number of years to simulate (nyears, numeric).
To run climate change scenarios a numeric vector with 12 entries as input (cc.vec) is necessary. Each entry of the vector represents the expected mean changes in precipitation of each month in percent. The preset value of the vector is zero.
Is possible to generate data in three different operation modi:

1. Normal mode, mode=0 (default):
Generates rainfall data that fits well to the historical record. (no climate change effects)

2. Linear mode, mode="linear":
Increases climate change effects linearly form the first generated year up to the total number of generated years. (e.g. useable between years 2000 and 2100)

3. Full impact mode, mode="fullimp":
Applies climate change effects by their full strength immediately and equally for all generated years. (e.g. useable after year 2100)

The bias correction value and model formulas depend on the historical data set and can be different from site to site. Therefore, it is recommended to run 'estimate.models()' and 'estimate.bias()' for each data set once and hand their outputs over to the bias correction (bias) and the model formulas (models) function arguments in 'raingen.()'. The bias argument is a numeric value and models requires the model formulas for the wetness and precipitation amount model as a character string in a list (list members: $wet,$amount). The default predictor of both models is merely the monthly precipitation amount. The preset value for bias correction is zero.
In addition, the function offers the option to play a notification sound, when the data generation is finished (sound=true/false). The default is no sound.

The function returns a list, that contains the generated data, annual and monthly rainfall amounts, fitted distributions, model summaries and others. (see table 1)

List member	Description
gen.data	generated data in a data frame
annual.output	generated annual precipitations
annual.input	(bias corrected) drawn annual precipitations on the annual level
monthly.output	generated monthly precipitations
monthly.input	calculated monthly precipitations by the method of monthly fragments
cat.input	annual categories (dry, intermediate, wet) of the drawn annual precipitations
cat.output	annual categories (dry, intermediate, wet) of the generated data
cat.th	thresholds (first/third quartile of the record) of the annual categories
distr.fit	table containing fitted distributions, parameters and chi-squared values for the precipitations amounts
best.distr	name of distribution with the lowest chi-squared value
summary.amount	summary (S3 method for class 'glm') of the precipitation amount model
summary.wet	summary (S3 method for class 'glm') of the wetness model
raw.data	original record data
raw.annual	annual precipitations of the record
raw.monthly	monthly precipitations of the record

Table 1: Members of the returned list of the 'raingen()' function.

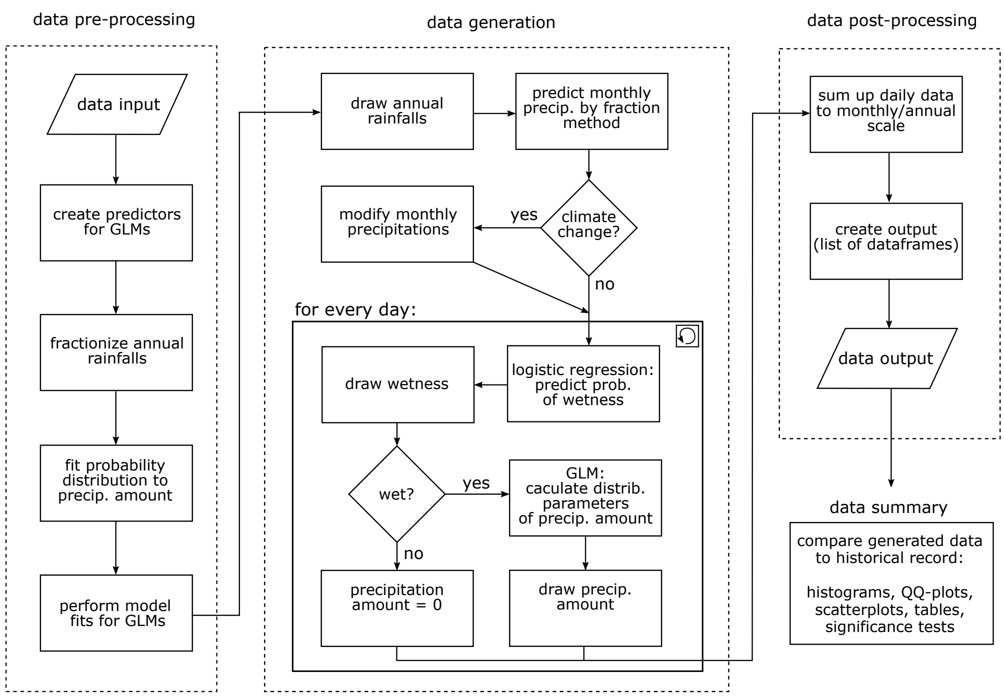

Figure 3: The flow chart of the code: The function 'raingen()' is subdivided in three parts: data pre-processing, data generation and data post-processing. The output of 'raingen()' can be used as input for 'raingen.summary()' directly.

3.4 raingen.summary()

This function allows to plot main the statistics of the generated data compared to the historical record. Therefore, 'raingen.summary()' requires the list, which is returned by the 'raingen()'

function, as input. The function delivers different box plots, QQ-plots, scatter plots, tables and significant tests of annual and monthly variables, as well as the input and output precipitations for each level. The box plots of monthly precipitations, wet days per month and precipitation amounts are additionally separated into dry, intermediate and wet years. Dry (wet) years are defined as years that fall below (exceed) the first (third) quartile of the annual precipitations of the record. All other years are denoted as intermediate years.
The function returns a list containing several statistics of the historical and generated data. (See table 2)

List member	Description
MAP.recdata	mean annual precipitation of the data record
MAP.gendata	mean annual precipitation of the generated data
SD.recdata	annual standard deviation of the data record
SD.gendata	annual standard deviation of the generated data
monthly.recdata	monthly data of the data record (number of wet days, monthly precipiations, etc.)
monthly.gendata	monthly data of the generated data
switched.cat	comparison of years (annual precipitations as input), that switched their annual category after data generation (output) in a table
prop.switched.cat	proportion table of 'switched.cat'

Table 2: Members of the returned list of the 'raingen.summary()' function.

4 Sample code in R

```
source("raingen.R")
source("raingen.summary.R")
source("estimate.models.R")
source("estimate.bias.R")

### number of years to simulate
nyears<-100

### load historical data (load your data here)
load("eneabba.data.RData")

### estimate best models
best.models<-estimate.models(data)

### estimate bias (takes quite long) (alternatively: b<--24)
b<-estimate.bias(data,models=best.models)

### projected climat change vector
cc<-c
    (-2.4,3.8,5.2,1.9,-7.6,-12.9,-11.3,-14.5,-20.3,-17.8,-12.9,-6.4)

#### generate data
raingen.list<-raingen(data,nyears,bias=b,models=best.models,sound=T)
raingen.list.linear<-raingen(data,nyears,cc.vec=cc,mode="linear",
    bias=b,models=best.models,sound=T)
raingen.list.fullimp<-raingen(data,nyears,cc.vec=cc,mode="fullimp",
    bias=b,models=best.models,sound=T)

#### summarize data
summary<-raingen.summary(raingen.list)
```

```
summary.linear<-raingen.summary(raingen.list.linear)
summary.fullimp<-raingen.summary(raingen.list.fullimp)
```

5 Acknowledgments

This generator was developed within the project "TI 824/2-1 Ecosystem resilience towards climate change - the role of interacting buffer mechanisms in Mediterranean-type ecosystems" funded by the German Research Foundation DFG. Furthermore, my thanks go to all members of the AG Tietjen, especially to Susanne Zander, for a previous version of the rain generator and Hanna Weise for general support and discussions. Last but not least, many thanks to Julian Bohländer (FU Berlin, AG Clinical Psychology and Psychotherapy) for fruitful discussions on the statistical problems of the generator.

References

[1] R. Coe R. D. Stern. A model fitting analysis of daily rainfall data. *Journal of the Royal Statistical Society. Series A (General)*, 147(1):1–34, 1984.

[2] R. Srikanthan and T.A. McMahon. Stochastic generation of rainfall and evaporation data. *AWRC Technical Paper No. 84, 301pp*, 1985.

[3] J.W. Porter and B.J. Pink. A method of synthetic fragments for disaggregation in stochastic data generation. *Hydrology and Water Resources Symposium, Institution of Engineers, Australia, 187191*, 1991.

[4] D.S. Wilks. Interannual variability and extreme-value characteristics of several stochastic daily precipitation models. *Agricultural and Forest Meteorlogy, 93, pp. 153169*, 1999.

[5] M.B. Parlange R.W. Katz. Over-dispersion phenomenon in stochastic modeling of precipitation. *J. Climate, 11, pp. 591601*, 1998.

[6] P.D. Jones J.M. Gregory, T.M.L. Wigley. Application of markov models to area-average daily precipitation series and interannual variability in seasonal total. *Climate Dynamics, 8, pp. 299310*, 1993.