RMAWGEN: A software project for a daily Multi-Site Weather Generator with R

RMAWGEN Daily Weather Generator

Emanuele Cordano, Emanuele Eccel

CRI - Fondazione Edmund Mach San Michele all'Adige (TN), Italy

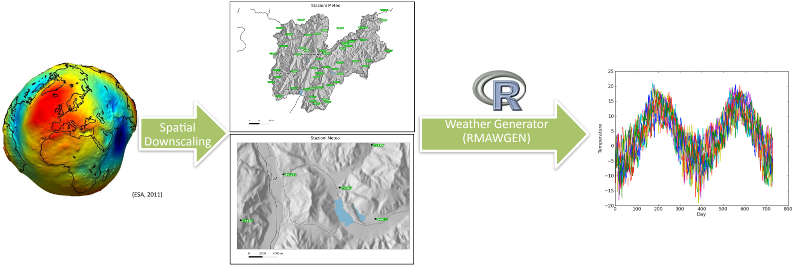
August 13, 2012

Outline

Motivation

- Understanding the effects of climate change on phenological and agricultural processes in Trentino, an alpine region, Italy.
- Need of daily time-series of temperature and precipitation corresponding to climate projections at several sites!
- Development of generation algorithms with R platform

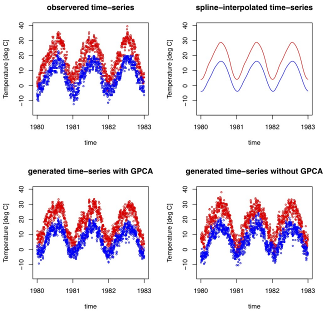
Motivation



Courtesy of Riccardo De Filippi

- Downscaling climate predictions from Global Climate Models (GCM), mean monthly climatology is obtained for several sites in the Trentino region and its neighborhood.
- RMAWGEN aims to generate randomly daily time series from the obtained monthly climatology in each site.

A Daily Weather Generator



- A weather generator (WG) aims to generate weather time series with the same statistical patterns of the observed ones.
- This presentation shows in detail how to calibrate a WG and generate series with RMAWGEN .
- Currently RMAWGEN generates daily maximum and minimum temperature and precipitation.
- Major details are explained going on with the presentation!

Theory: Vector Auto-Regressive Model (VAR)

A set of K random variables can be described by a Vector Auto-Regressive Model (VAR(K,p)) as follows:

$$\mathbf{x_t} = \mathbf{A_1} \cdot \mathbf{x_{t-1}} + ... + \mathbf{A_p} \cdot \mathbf{x_{t-p}} + \mathbf{u_t} \tag{1}$$

where $\mathbf{x_t}$ is a K-dimensional vector representing the set of weather variables generated at day t by the model, called "endogenous" variables, $\mathbf{A_i}$ is a coefficient matrix $K \times K$ for $i = 1, ..., p$ and $\mathbf{u_t}$ is a K-dimensional stochastic process. $\mathbf{x_t}$ and $\mathbf{u_t}$ are usually normalized to have a null mean. u_t is a *Standard White Noise* [Luetkepohl, 2007,def. 3.1], i.e. a continuous random process with zero mean and $\mathbf{u_t}, \mathbf{u_s}$ independent for each $t \neq s$, consequently it has a time-invariant nonsingular covariance matrix.

Theory: VAR with exogenous variable

$$\mathbf{x_t} = \mathbf{A_1} \cdot \mathbf{x_{t-1}} + \ldots + \mathbf{A_p} \cdot \mathbf{x_{t-p}} + \mathbf{u_t}$$
$$\mathbf{u_t} = \mathbf{C} \cdot \mathbf{d_t} + \mathbf{B} \cdot \epsilon_\mathbf{t} \qquad (2)$$

Here, u_t is a process described by (2) where $\mathbf{d_t}$ is a set of known K-dimensional processes, whose components are called "exogenous" variables, and $\epsilon_\mathbf{t}$ is a K-dimensional uncorrelated standard white noise, i.e. $\mathrm{E}[\epsilon_\mathbf{t} \cdot \epsilon_\mathbf{t}^\mathbf{T}] = \mathbf{\Sigma}_{\epsilon_\mathbf{t}} = \mathbf{I_K}$, $\mathbf{C}$ and $\mathbf{B}$ are the respective coefficient matrices and $\mathbf{I_K}$ is the K-dimensional identity matrix. The process $\mathbf{d_t}$ is a generic, previously generated, known meteorological variable that works as a predictor vector for the stochastic generation of $\mathbf{x_t}$.

VAR Coefficient Estimation (VAR Calibaration)

- Method of Moments (based on Yule-Walker Equation) [Luetkepohl, 2007];
- Method of Maximum Likelihood [Luetkepohl, 2007];
- Method of Least Squares (OLS) using command `lm` (see `VAR()` in [Pfaff, 2008]

Currently RMAWGEN uses OLS Method through the command `getVARmodel()` which is wrapper of `VAR()`.

Model Diagnostic

Once built the VAR(p) model and estimated its coefficients, the model residuals $\mathbf{B} \cdot \epsilon_{\mathbf{t}}$, as expressed in (2), can be processed through diagnostic tests, which can be resumed in the following [Pfaff, 2008]:

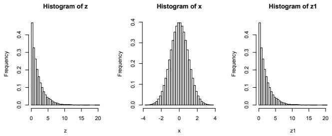
- **Multivariate Portmanteau and Breusch-Godfrey (Lagrange Multiplier) tests**, which verify the absence of time-autocorrelation of the VAR(p) residuals (seriality tests);
- **Jarque-Bera multivariate skewness and kurtosis test**, which validate the multivariate Gaussian probability distribution of the VAR(p) residuals (normality test).

Theory: Gaussianization

VAR models work correctly if the variable $\mathbf{x_t}$ is normally distributed. This requires a normalization procedure of the meteorological value, which can be formally expressed as:

$$\mathbf{x_t} = \mathbf{G_m}\left(\mathbf{z_t}\right) \tag{3}$$

where $\mathbf{z_t}$ is the meteorological time series and $\mathbf{G_p}$ is a suitably defined function so that $\mathbf{x_t}$ is multi-normally distributed and can vary according to time, month, and season. For a weather generator, *m* is an indicator of the month.

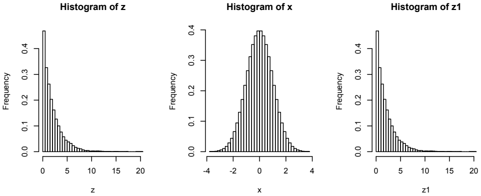


Theory: 1D Marginal Gaussianization

It is based on quantile transformation:

$$x = F_G^{-1}(F_z(z)) \tag{4}$$

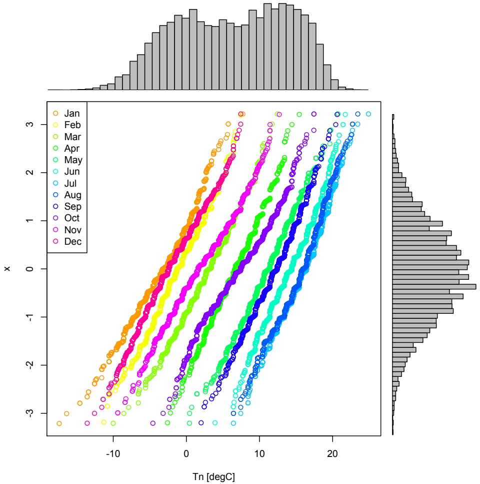
where $F_z(z)$ is cumulate distribution probability of z, $F_G(x)$ is the Gaussian cumulative function with zero mean and standard deviation equal to 1.




```
library(RMAWGEN)
set.seed(123456)
z <- rexp(10000,rate=0.5)
x <- normalizeGaussian(x=z,data=z)
z1 <-
normalizeGaussian(x=x,data=z,inverse=TRUE)
#-- Plot probability histogram of z, x
and z1
zhist <-
hist(z,probability=TRUE,breaks=50)
xhist <-
hist(x,probability=TRUE,breaks=50)
z1hist <-
hist(z1,probability=TRUE,breaks=50)
zhist$counts <- zhist$density
xhist$counts <- xhist$density
z1hist$counts <- z1hist$density
def.par <- par(no.readonly = TRUE) #
save default, for resetting...
par(mfrow=c(1,3),oma=c(15.0,0.0,15.0,0.0))
plot(zhist)
plot(xhist)
plot(z1hist)

par(def.par)
```

Theory: Monthly 1D Marginal Gaussianization



The gaussianization transformation can be different for each month.

```
library(RMAWGEN)
data(trentino)
col <- rainbow(n=12,start=0.1,end=0.9)
col[6:1] <- col[1:6]
col[7:12] <- col[12:7]
plot_sample(x=TEMPERATURE_MIN$T0090,
sample="monthly",
origin="1958-1-1",axes=FALSE,xlab="Tn
[degC]",ylab="x",abline=NULL,col=col)
```

Theory: Multi-Dimensional Gaussianization (1)

Similarly to the 1D case, it is intuitive that an invertible function exists between any multi-dimensional random variable and a Gaussian variable with the same dimensions [Chen and Gopinath, 2000]. Recently, Laparra et al. [2009] proposed an iterative method based on Principal Component Analysis (PCA):

- One-dimensional Gaussianization of each marginal variable (component of $\mathtt{x}$), i.e. *Marginal Gaussianization*;
- Orthonormal transformation of the coordinates based on the eigenvector matrix of the covariance matrix according to PCA analysis.

Details about convergence of the method are given in Laparra et al. [2009].

Theory: Multi-Dimensional Gaussianization (2)

The PCA Gaussianization (G-PCA) is described by the following mathematical formulation (Laparra et al. [2009]).

$$\mathbf{x}^{(\mathbf{k}+\mathbf{1})} = \mathbf{B}_{(\mathbf{k})} \cdot \mathbf{\Psi}_{(\mathbf{k})}(\mathbf{x}^{(\mathbf{k})}) \tag{5}$$

where $\mathbf{x}^{(\mathbf{k})}$ and $\mathbf{x}^{(\mathbf{k}+\mathbf{1})}$ are the vector random variable at the k-th and the $k+1$-th iteration level, $\mathbf{\Psi}_{(\mathbf{k})}(\mathbf{x}^{(\mathbf{k})})$ is the marginal Gaussianization of each component of random variable $\mathbf{x}^{(\mathbf{k})}$ performed by applying (4), $\mathbf{B}_{(\mathbf{k})}$ is the PCA transform matrix, i.e. the orthonormal eigenvector matrix of the covariance matrix of $\mathbf{\Psi}_{(\mathbf{k})}(\mathbf{x}^{(\mathbf{k})})$.

Theory: Multi-Dimensional Gaussianization (3)

The inverse transformation is found by inverting (5) as follows:

$$\left(\mathbf{x}^{(\mathbf{k})}\right) = \mathbf{\Psi}_{(\mathbf{k})}^{-1}\left(\mathbf{B}_{(\mathbf{k})}^{-1} \cdot \mathbf{x}^{(\mathbf{k+1})}\right) = \mathbf{\Psi}_{(\mathbf{k})}^{-1}\left(\mathbf{B}_{(\mathbf{k})}^{\mathbf{T}} \cdot \mathbf{x}^{(\mathbf{k+1})}\right) \qquad (6)$$

where $\mathbf{B}_{(\mathbf{k})}^{-1}$ and $\mathbf{B}_{(\mathbf{k})}^{\mathbf{T}}$ are confused because of the property of rotation matrix ($\mathbf{B}_{(\mathbf{k})}^{-1} = \mathbf{B}_{(\mathbf{k})}^{\mathbf{T}}$).
Laparra et al. [2009] demonstrated the convergence of (5) by showing that the negentropy (i.e. the divergence from the multivariate Gaussian Distribution) of $\mathbf{x}^{(\mathbf{k+1})}$ reduces after a reasonable number of iterations.

Representation of Weather Variable: defining $\mathbf{z_t}$

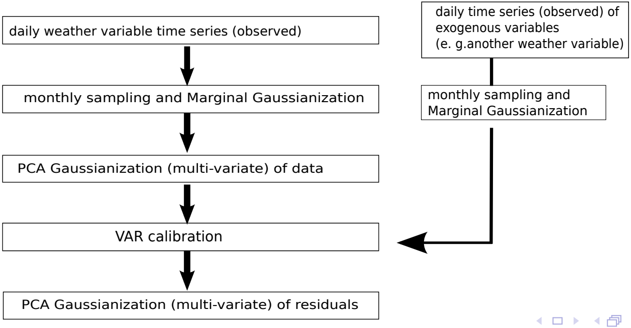
- **Temperature Generation**: $\rightarrow$ preliminary deseasonalization

$$\mathbf{z_t} = \left(\frac{\mathbf{Tx_t} + \mathbf{Tn_t}}{2} - \frac{\mathbf{Tx_{st}} + \mathbf{Tn_{st}}}{2} \middle| \mathbf{Tx_t} - \mathbf{Tn_t} \right) \tag{7}$$

  - $\mathbf{Tx_t}$: daily maximum temperature
  - $\mathbf{Tn_t}$: daily minimum temperature
  - $\mathbf{Tx_{st}}$: daily maximum spline-interpolated temperature (from "monthly climatology")
  - $\mathbf{Tn_{st}}$: daily minimum spline-interpolated temperature (from "monthly climatology")
- **Precipitation Generation**: $\mathbf{z_t}$ is equal to daily precipitation
- **Marginal Gaussianization for both temperature or precipitation case** $\Longrightarrow \mathbf{x_t} = \mathbf{G_m}(\mathbf{z_t})$

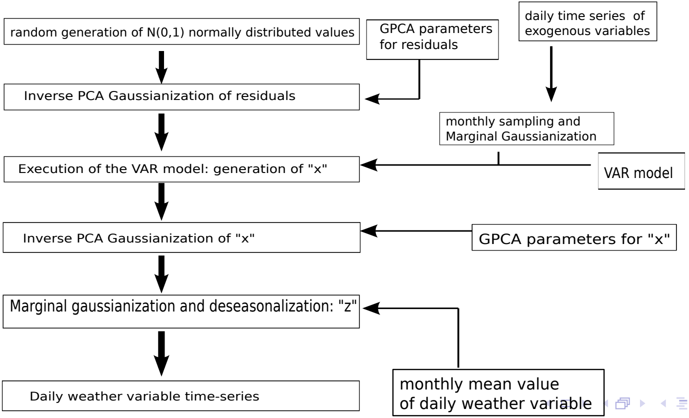
Calibration of a VAR for weather generation from observed data

A VAR model is calibrated from observed weather data according to the following algoritm



Stochastic weather generation by a previously created VAR

Inversely, once calibrated the VAR model, weather data are generated as follows:



Software Implementation

The temperature and precipitation generators are lumped in two main functions:

- `ComprehensivePrecipitationGenerator(...)`
- `ComprehensiveTemperatureGenerator(...)`

Both functions return a stochastic generation of daily temperature or precipitation taking as input the monthly means (e.g., predicted by downscaling Climate Model results) and the Vector Auto-Regressive models (VAR - with Gaussianization) stored as `varest2` or `GPCAvarest2` S4 objects.

Software Implementation: deseasonalization (temperature) and monthly sampling

In particular the monthly sampling and the deseasonalization of daily temperature time series are performed using the function `extractmonths()` and `getMonthlyMean()`. Then, the functions:

- `splineInterpolateMonthlytoDailyforSeveralYears (val=mean_climate_Tn,...)`
- `splineInterpolateMonthlytoDailyforSeveralYears (val=mean_climate_Tx,...)`

make a daily spline interpolation of maximum and minimum temperature and generate the time series $\mathbf{Tx_{st}}$ and $\mathbf{Tn_{st}}$

Software Implementation: monthly sampling

After obtaining $\mathbf{z_t}$, a marginal gaussianization is done by:

- `normalizeGaussian_severalstations( x=data_original,data=data_original, sample="monthly",origin_x=origin,origin_data=origin)`

where:

- `x` is the time series $\mathbf{z_t}$;
- `data` corresponds to the sample representative of the probability distribution (in this case it is equal to $\mathbf{z_t}$);
- `sample` is a character string: if it is "`monthly`" the gaussianization is separately done for each month;
- `origin_x` and `origin_data` are the start day of `x` and `data` time series.

Software Implementation: `GPCAvarest2` object

The obtained time series are then called $\mathbf{x_t^0}$ and are processed as follows:

- `getVARmodel(...)`

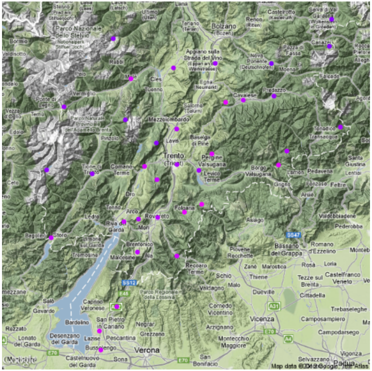
which resumes PCA Gaussianization and VAR calibration and returns the VAR model as a `varest2` or `GPCAvarest2`. The `GPCAvarest2` class extends the `varest2` and contains the following slots:

Software Implementation: GPCAvarest2 object (2)

- GPCA_data: a list containing the parameters of the multi-variate gaussianization of the time series; it is the result of the function GPCA applied to the input data of getVARmodel;
- VAR: the VAR model obtained with OLS estimators of the gaussianized data time series by applying the VAR function [Pfaff, 2008];
- GPCA_residuals: a list containing the parameters of the multi-variate gaussianization of the residuals of the VAR model contained in VAR; it is NULL if no gaussianization of residuals is applied.

varest2 contains only the slot VAR. Major details in function documentation.

Daily Temperature and Precipitation in `trentino` dataset [Eccel et al., 2012]



See file `trentino_map.R` in the doc/example directory of RMAWGEN package source code.

Dataset Overview (1)

- The `trentino` dataset contains recorded and homogenized 50-year long time series of daily precipitation and temperature in 59 sites. Look at `help(trentino)` in RMAWGEN documentation for details.
- The region altitude ranges between 70 m a.s.l. (Lake of Garda) to 3769 m (Mount Cevedale).

Dataset Overview (2)

- The area covered by the weather station network can be ascribed to a Köppen class [Peel et al., 2007] ranging from "Cfb" ("temperate, middle latitudes climate, with no dry season") to "Dfc" classification ("microthermal climate, humid all year round") in the more elevated, mountain areas.
- Precipitation amounts are mostly distributed over two maxima, in the autumn (main) and in the spring (secondary), although in some mountain areas rainfall peaks in summer [Eccel and Saibanti, 2005].

Let's start with a precipitation generation

4 different models with different auto-regression order and with or without GPCA are considered.

```
rm(list=ls())
library(RMAWGEN) # Call RMAWGEN
set.seed(1222) # Set the seed for random generation
data(trentino) # Load the dataset
# Two stations where to work!
station <- c("T0090","T0083")
```

Stations here used for precipitation generation (2)

For simplicity, these two stations are taking into account:

```
R>names (ELEVATION) <- STATION_NAMES
R>names (LOCATION) <- STATION_NAMES
R> ELEVATION[station]
T0083 T0090
665.26 225.24
R> LOCATION[station]
T0083 T0090
"CLES" "MEZZOLOMBARDO (CONVE"
```

Stations used for precipitation generation (2)

The two stations are located at Cles and at Mezzolombardo (Trentino, Italy) respectively. Their elevations are 665 m and 225 m a.s.l. .

The localization of the two stations can be viewed through the script `trentino_map2.R` in the doc/example directory of RMAWGEN package source code.

Precipitation generation parameters (1)

```
 # Monthly climate (given by observed time series)
PREC_CLIMATE <- NULL
# Calibration period
year_max <- 1990
year_min <- 1961
origin <- "1961-1-1"
# Simulation period (Stochastic Generation)
# The same of calibaration period
```

Precipitation generation parameters (2)

Number of GPCA iterations for variable and VAR residuals:

```
n_GPCA_iter <- 5
n_GPCA_iteration_residuals <- 5
```

Two different auto-regression orders *p* and no exogenous variables are considered: `p_test <- 1`

```
p_prec <- 3
exogen <- NULL
exogen_sim <- exogen
```

Launching a Precipitation Generation with auto-regression order $p = 3$ and PCA Gaussianization (P03GPCA)

```
generationP03GPCA_prec <-
ComprehensivePrecipitationGenerator(station=station
,prec_all=PRECIPITATION,year_min=year_min,
year_max=year_max,p=p_prec,
n_GPCA_iteration=n_GPCA_iter,
n_GPCA_iteration_residuals=n_GPCA_iteration_residuals,
exogen=exogen,exogen_sim=exogen_sim,
sample="monthly",
mean_climate_prec=PREC_CLIMATE,no_spline=FALSE)
```

Launching a Precipitation Generation with auto-regression order $p = 1$ and PCA Gaussianization (P01GPCA)

```
generationP01GPCA_prec <-
ComprehensivePrecipitationGenerator(station=station
,prec_all=PRECIPITATION,year_min=year_min,
year_max=year_max,p=p_test,
n_GPCA_iteration=n_GPCA_iter,
n_GPCA_iteration_residuals=n_GPCA_iteration_residuals,
exogen=exogen,exogen_sim=exogen_sim,sample="monthly",
mean_climate_prec=PREC_CLIMATE,no_spline=FALSE)
```

Launching a Precipitation Generation with auto-regression order $p = 3$ and without PCA Gaussianization (P03)

```
generationP03_prec <-
ComprehensivePrecipitationGenerator(station=station
,prec_all=PRECIPITATION
,year_min=year_min,
year_max=year_max,p=p_prec,
n_GPCA_iteration=0,n_GPCA_iteration_residuals=0,
exogen=exogen,exogen_sim=exogen_sim,sample="monthly",
mean_climate_prec=PREC_CLIMATE,no_spline=FALSE)
```

Launching a Precipitation Generation with auto-regression order $p = 1$ and without PCA Gaussianization (P03)

```
generationP01_prec <-
ComprehensivePrecipitationGenerator(station=station
,prec_all=PRECIPITATION
,year_min=year_min,
year_max=year_max,p=p_test,
n_GPCA_iteration=0,n_GPCA_iteration_residuals=0,
exogen=exogen,exogen_sim=exogen_sim,sample="monthly",
mean_climate_prec=PREC_CLIMATE,no_spline=FALSE)
```

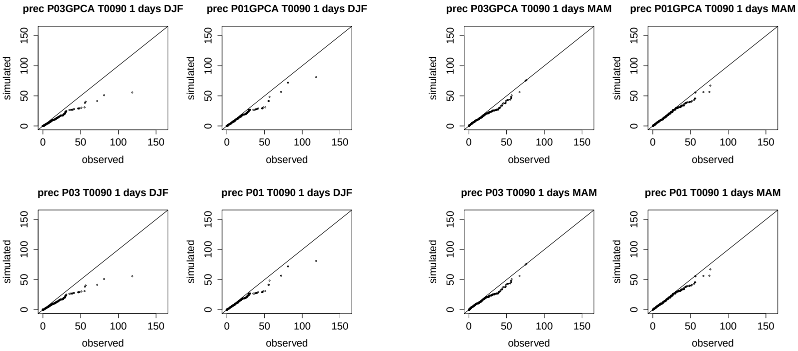
Precipitation VAR Model Diagnostic

```
R> serial_test(generationP03GPCA_prec$var)
R> normality_test(generationP03GPCA_prec$var)
```

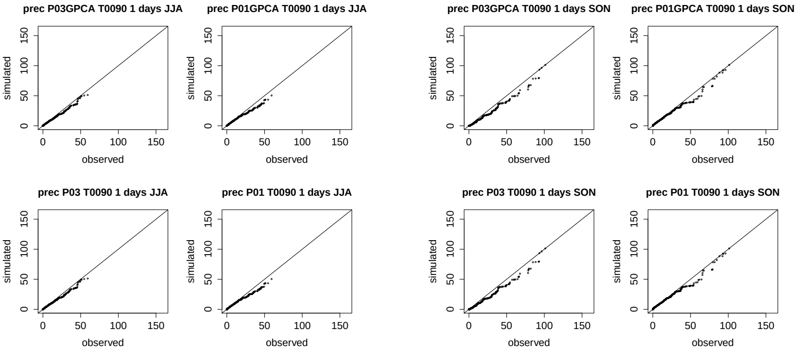
	Normality Test	Serial Test
P01	Unsuccessful	Unsuccessful
P03	Unsuccessful	Successful
P01GPCA	Successful	Successful
P03GPCA	Successful	Successful

Table 1: Test results on VAR residuals (precipitation generation).

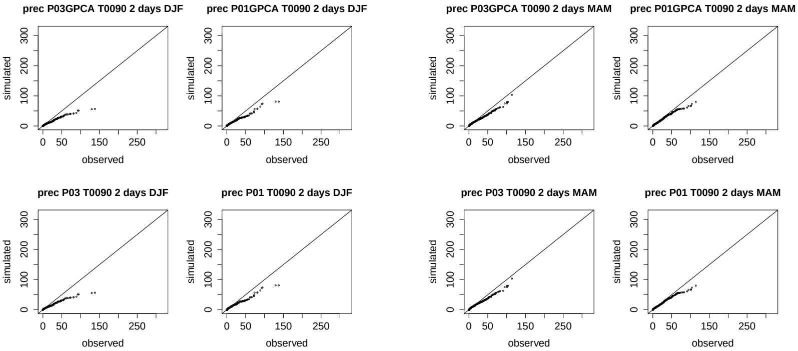
Q-Q plots for daily precipitation (generated vs observed) at T0090 in DJF and MAM)



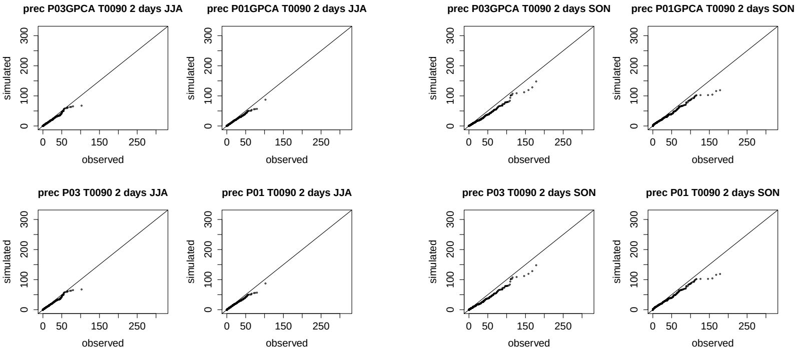
Q-Q plots for daily precipitation (generated vs observed) at T0090 in JJA and SON



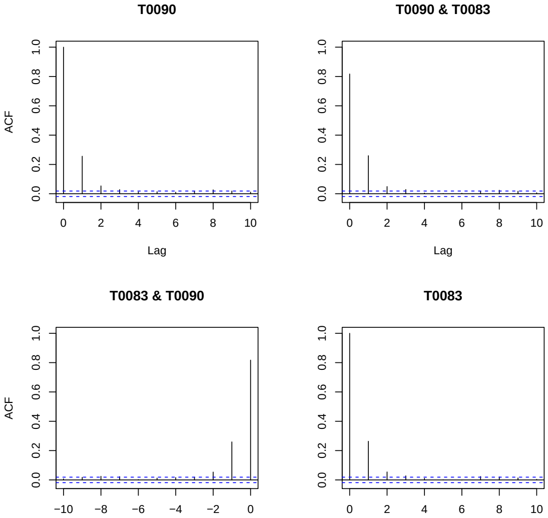
Q-Q plots for LAG 2 daily precipitation (generated vs observed) at T0090 in DJF and MAM



Q-Q plots for LAG 2 daily precipitation (generated vs observed) at T0090 in JJA and SON

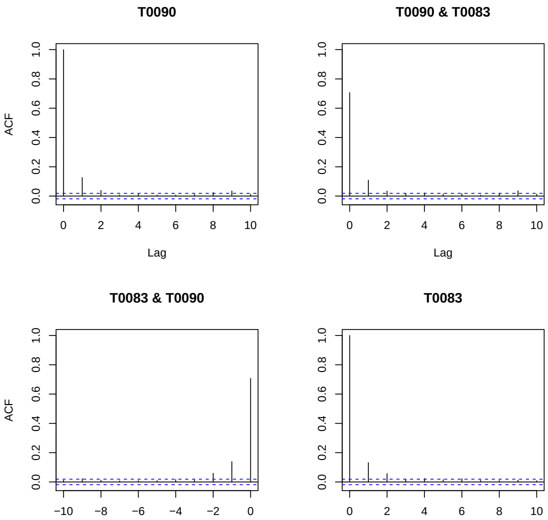


Auto-Correlation Function at T0090 and T0083



Auto-Correlation Function of OBSERVED daily precipitation values.

Auto-Correlation Function at T0090 and T0083



Auto-Correlation Function of P03GPCA GENERATED daily precipitation values.

Discussion

The shown tests and Q-Q plot illustrate the VAR model goodness:

- the Q-Q plots (observed vs generated) shows results, especially for the lower precipitation values (<20 mm) in MAM,JJA and SON seasons and for P03GPCA case;
- probability values of no precipitation occurrence (observed vs generated) are well reproduced:

The P03GPCA VAR model is accepted for stochastic precipitation generation according to diagnostic test results.
Problems to be solved:

- maximum values of precipitation are underestimated, especially in DJF:
- the temporal auto-correlation is underestimated (see LAG 2 Q-Q plots and auto-correlation function plots)

Daily temperature is modeled using precipitation data as exogenous predictors (1)

```
# Monthly climate (given by observed time series)
TN_CLIMATE <- NULL
TX_CLIMATE <- NULL
```

The variables `station`,`year_max`,`year_min`,`origin` (and then the calibration and simulation periods) are the same used for precipitation generation. The same number of GPCA iterations for variable and VAR residuals is considered:

```
n_GPCA_iter <- 5
n_GPCA_iteration_residuals <- 5
```

Two different auto-regression orders p are considered:

```
p_test <- 1
p_temp <- 10
```

Daily temperature is modeled using precipitation data as exogenous predictors (2)

Observed and P03GPCA generated marginally Gaussianized precipitation are taken as an exogenous variable for temperature generation:

```
exogen <-
normalizeGaussian_severalstations(x=prec_mes,
data=prec_mes,sample="monthly",
origin_x=origin,origin_data=origin,step=0)
exogen_sim <-
normalizeGaussian_severalstations(x=prec_gen$P03GPCA,
data=prec_gen$P03GPCA,sample="monthly",
origin_x=origin,origin_data=origin,step=0)
```

Launching a Temperature Generation with auto-regression order $p = 10$ and PCA Gaussianization (P10GPCA)

```
generationP10GPCA_temp <-
ComprehensiveTemperatureGenerator (station=station,
Tx_all=TEMPERATURE_MAX,Tn_all=TEMPERATURE_MIN,
year_min=year_min, year_max=year_max,p=p_temp,
n_GPCA_iteration=n_GPCA_iter,
n_GPCA_iteration_residuals=n_GPCA_iteration_residuals,
exogen=exogen,exogen_sim=exogen_sim,sample="monthly",
mean_climate_Tn=TN_CLIMATE,mean_climate_Tx=TX_CLIMATE)
```

Launching a Temperature Generation with auto-regression order $p = 1$ and PCA Gaussianization (P01GPCA)

```
generationP01GPCA_temp <-
ComprehensiveTemperatureGenerator(station=station,
Tx_all=TEMPERATURE_MAX,Tn_all=TEMPERATURE_MIN,
year_min=year_min, year_max=year_max,p=p_test,
n_GPCA_iteration=n_GPCA_iter,
n_GPCA_iteration_residuals=n_GPCA_iteration_residuals,
exogen=exogen,exogen_sim=exogen_sim,sample="monthly",
mean_climate_Tn=TN_CLIMATE,mean_climate_Tx=TX_CLIMATE)
```

Launching a Temperature Generation with auto-regression order $p = 10$ and without PCA Gaussianization (P10GPCA)

```
generationP10_temp <-
ComprehensiveTemperatureGenerator(station=station,
Tx_all=TEMPERATURE_MAX,Tn_all=TEMPERATURE_MIN,
year_min=year_min,year_max=year_max,p=p_temp,
n_GPCA_iteration=0,n_GPCA_iteration_residuals=0,
exogen=exogen,exogen_sim=exogen_sim,sample="monthly",
mean_climate_Tn=TN_CLIMATE,mean_climate_Tx=TX_CLIMATE)
```

Launching a Temperature Generation with auto-regression order $p = 1$ and without PCA Gaussianization (P01GPCA)

```
 generationP01_temp <-
ComprehensiveTemperatureGenerator (station=station,
Tx_all=TEMPERATURE_MAX,Tn_all=TEMPERATURE_MIN,
year_min=year_min,year_max=year_max,p=p_test,
n_GPCA_iteration=0,n_GPCA_iteration_residuals=0,
exogen=exogen,exogen_sim=exogen_sim,sample="monthly",
mean_climate_Tn=TN_CLIMATE,mean_climate_Tx=TX_CLIMATE)
```

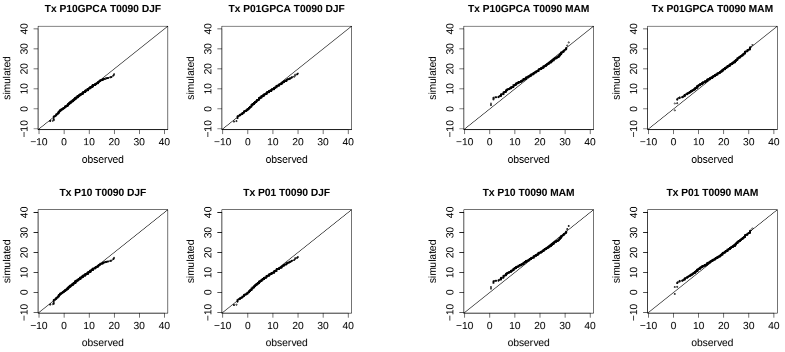
Temperature VAR Model Diagnostic

```
R> serial_test(generationP10GPCA_temp$var)
R> normality_test(generationP10GPCA_temp$var)
```

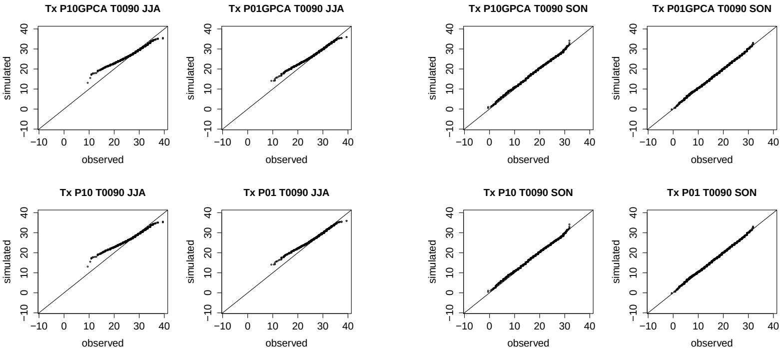
	Normality Test	Serial Test
P01	Unsuccessful	Unsuccessful
P10	Unsuccessful	Unsuccessful
P01GPCA	Successful	Unsuccessful
P10GPCA	Successful	Successful

Table 2: Test results on VAR residuals (temperature generation).

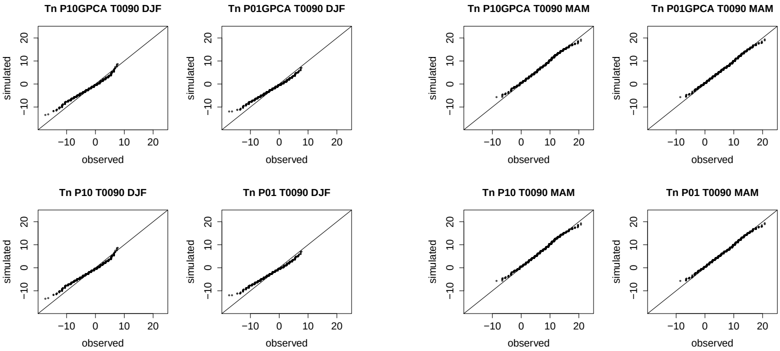
Q-Q plots for daily maximum temperature (generated vs observed) at `T0090` in DJF and MAM.



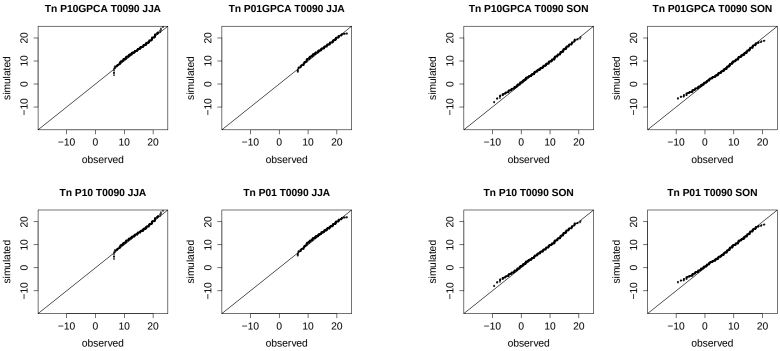
Q-Q plots for daily maximum temperature (generated vs observed) at T0090 in JJA and SON.



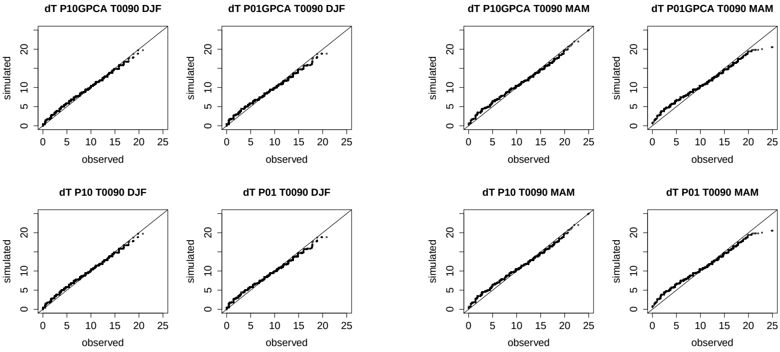
Q-Q plots for daily minimum temperature (generated vs observed) at T0090 in DJF and MAM.



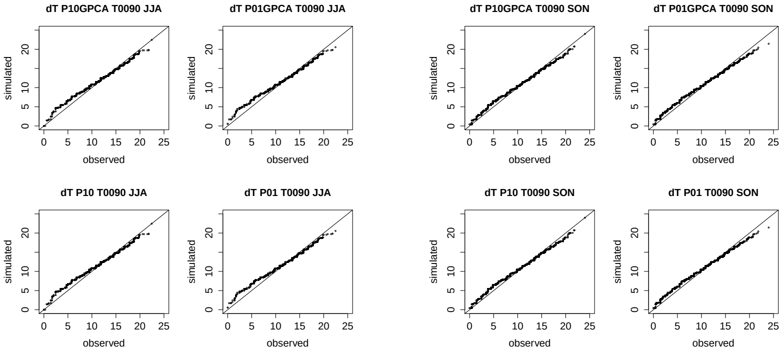
Q-Q plots for daily minimum temperature (generated vs observed) at `T0090` in JJA and SON.



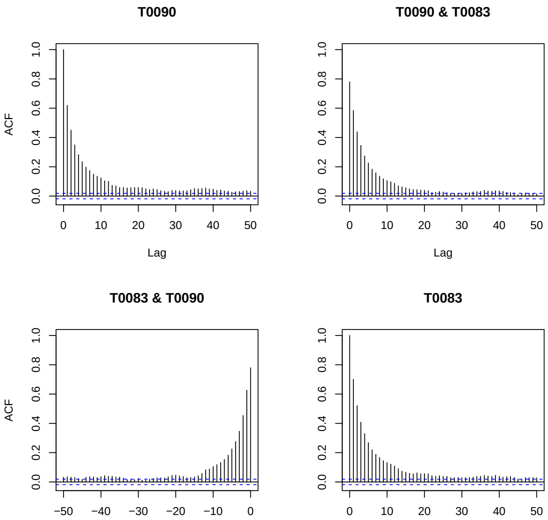
Q-Q plots for daily thermal range (generated vs observed) at T0090 in DJF and MAM.



Q-Q plots for daily thermal range (generated vs observed) at T0090 in JJA and SON.

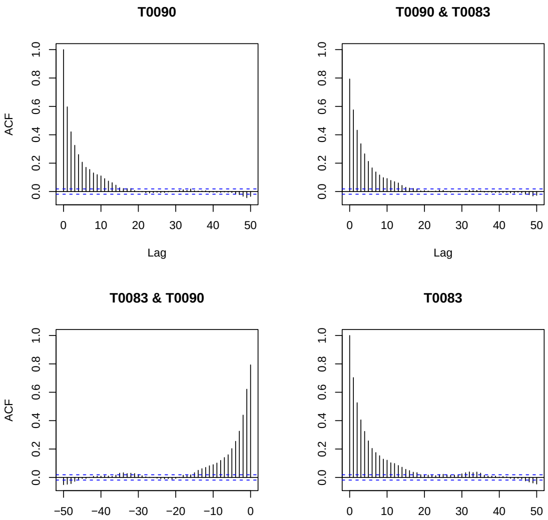


Auto-Correlation Function at T0090 and T0083



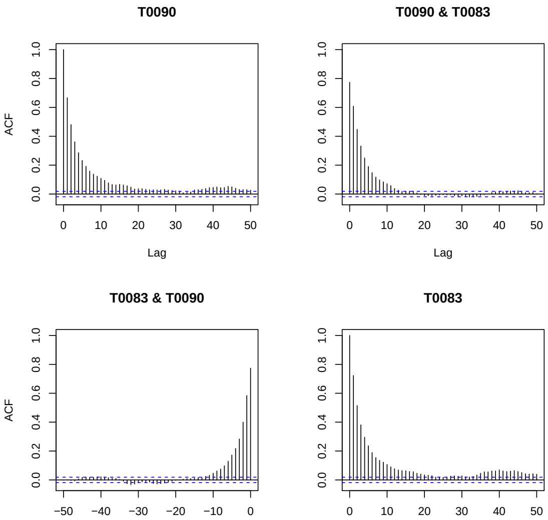
Auto-Correlation Function of OBSERVED daily maximum temperature anomalies.

Auto-Correlation Function at T0090 and T0083



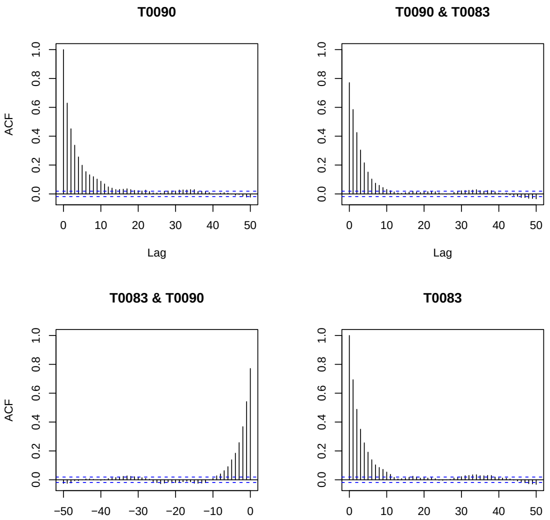
Auto-Correlation Function of P10GPCA GENERATED daily maximum temperature anomalies.

Auto-Correlation Function at T0090 and T0083



Auto-Correlation Function of OBSERVED daily minimum temperature anomalies.

Auto-Correlation Function at T0090 and T0083



Auto-Correlation Function of P10GPCA GENERATED daily minimum temperature anomalies.

Discussion

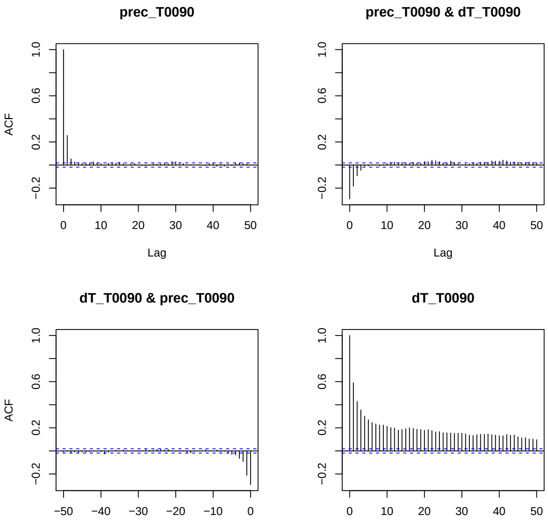
The shown tests and Q-Q plots illustrate the VAR model goodness:

- the Q-Q plots (observed vs generated) show good results for maximum and minimum temperature, especially for P10GPCA case;
- auto-covariance functions of generated data are well reproduced:

The P10GPCA VAR model is accepted for stochastic temperature generation according to diagnostic test results.
Problems to be solved:

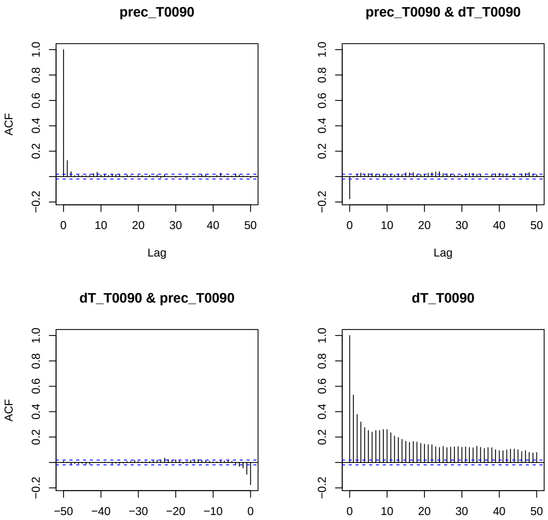
- small discrepancies between observed and generated daily thermal range appear in JJA.

Auto-Correlation Function at T0090 (Prec. vs Temp.)



Auto-Correlation Function of OBSERVED daily precipitation and thermal range.

Auto-Correlation Function at T0090 (Prec. vs Temp.)



Auto-Correlation Function of GENERATED daily precipitation (P03GPCA) and thermal range (P10GPCA).

Discussion

- The relationship between generated daily temperature and precipitation is illustrated through the correlation between daily thermal range and precipitation. These two quantities are significantly dependent, since the presence of rainfall and, hence, of overcast or very cloudy sky strongly reduces the thermal range.
- A negative correlation coefficient between precipitation and daily thermal range is observed (around -0.2), a lower value of correlation coefficient (in terms of absolute value) is instead obtained from RMAWGEN simulations.

Conclusions

- RMAWGEN is a Stochastic Weather Generator for daily temperature and precipitation based on Gaussianized Vector Autor-Regressive Models (VAR).
- RMAWGEN generates simultaneously weather variables at several sites based on a Multi-Dimensional Gaussianization process.
- The presented application shows the good capabilities of RMAWGEN, especially for daily temperature generation.
- The work is under revision for publication on *Journal of Statistical Software*.
- RMAWGEN is available on CRAN: see package manual for major details on functions, classes and methods.

Bibliography

Scott Shaobing Chen and Ramesh A. Gopinath. Gaussianization. In *NIPS*, pages 423–429, 2000.

Emanuele Eccel and Serenella Saibanti. Inquadramento Climatico dell'Altopiano di Lavarone-Vezzena nel contesto generale trentino. *Studi Trentini di Scienze Naturali, Acta Biologica*, 82:111–121, 2005.

Emanuele Eccel, Piero Cau, and Roberto Ranzi. Data reconstruction and homogenization for reducing uncertainties in high-resolution climate analysis in alpine regions. *Theoretical and Applied Climatology*, pages 1–14, 2012. ISSN 0177-798X. URL http://dx.doi.org/10.1007/s00704-012-0624-z. 10.1007/s00704-012-0624-z.

Valero Laparra, Jordi Muñoz-Marí, Gustavo Camps-Valls, and Jesús Malo. Pca gaussianization for one-class remote sensing image classification. In *Proc. SPIE 7477*, volume 7477, page 74770U. SPIE, 2009. doi: DOI:10.1117/12.834011. URL http://dx.doi.org/doi/10.1117/12.834011.

Helmut Luetkepohl. *New Introduction to Multiple Time Series Analysis*. Springer-Verlag, Berlin Hedelberg, Germany, 2nd printing edition, 2007.

M. C. Peel, B. L. Finlayson, and T. A. McMahon. Updated world map of the knöppen-geiger climate classification. *Hydrology and Earth System Sciences*, 11(5):1633–1644, 2007. doi: 10.5194/hess-11-1633-2007. URL http://www.hydrol-earth-syst-sci.net/11/1633/2007/.

Bernhard Pfaff. VAR, SVAR and SVEC Models: Implementation within R package vars. *Journal of Statistical Software*, 27(4), 2008. URL http://www.jstatsoft.org/v27/i04/.