Sample code for estimating ET_0 and forecast series based on multi-ensembles models

Francisco Gomariz-Castillo, Francisco Alonso-Sarría and Marcos Ruiz-Álvarez

26/12/2020

Contents

1. Introduction

This Appendix presents the R code used to carry out the ET_0 ensembles using as example the weather station 7002A of the AEMET network. We have tried to implement a highly reproducible code to be used with one or more weather stations in different study areas and with different data sources. Although the main objective is to perform and evaluate individual regionalized ET_0 models and multi-model ensembles, the process is applicable to any other problem related to ensembles, variables or data sources related to Climate Change. Two functions perform the process:

1. *fun.ETPHargreaves* estimates ET_0 series using the Hargreaves equation calibrated in the study area according to Gomariz-Castillo, Alonso-Sarria, and Cabezas-Calvo-Rubio (2018) (final in Section *(A) Processing data* in Fig. 2 of the paper). Among other options, the function allows a) to estimate ET_0 either if *Rs* is known or if it has to be estimated from *Ra*, b) to modify the default parameters and c) to include calibration parameters based on the methodology proposed by Allen et al. (1994).
2. *fun.ensembles* validates the individual regionalized models, estimates and validates the multi-model ensemble methods, and generates forecasts for the RCP scenarios (Section *(B) Individual models and ensembles* in Fig. 2 of the paper).

The final subsection of this supplementary material describes in detail both functions and the required and optional parameters.

The following files are included as additional material:

- *data/*: Three data files for HISTORICAL, RCP4.5 and RCP8.5 scenarios (weather station 7002A).
- *data/src/functions.R*: *fun.ETPHargreaves* and *fun.ensembles* functions.

- *data/main.R*: Main code to launch the process.

The example version includes the possibility of parallelize the *caret* (machine learning based ensembles) functions, but the rest of the process is not parallelized to ensure compatibility with any platform. The presented process has large computational requirements when applied to daily time series. The performance improves significantly when this example is run on a PC with an Intel i7-9750H processor in a single core, the RAM peak has reached 3.1 GiB and has taken 52 min; this same process using 6 cores has needed a maximum RAM peak of 2.8 GiB per core (reaching 11 GiB overall) and has taken 18.28 min.

2. Data sources and processing data

The datasets used in this study were downloaded from publicly available sources. Daily series of maximum and minimum temperatures for individual regionalized projections (Historical and RCP scenarios) (Amblar Francés et al. 2017; AEMET 2020) are available on the Spanish State Agency of Meteorology (AEMET) data repository (http://www.aemet.es/es/serviciosclimaticos/cambio_climat/datos_diarios). Daily series of maximum and minimum temperature historical observations (reference data, Iberia01) [Herrera2019] are available on the Spanish National Research Council (CSIC) data repository (https://digital.csic.es/handle/10261/183071).

Once the matrices for the projections were obtained, the observed data were extracted from the Iberia01 grids and associated to the weather stations. For this example, the final matrices for the HISTORICAL, RCP4.5 and RCP8.5 scenarios of climate station 7002A are included in the *data/* directory (files *TASdatahist-7002A.csv*, *TASdatarcp45-7002A.csv* and *TASdatarcp85-7002A.csv*).

These tables contain information about:

- *idest* and *Date* columns: Station id and dates (in format %Y-%m-%d).
- *tmax_ref* and *tmin_ref* columns: Daily maximum and minimum reference temperatures.
- *tmax_NAMEMODEL* and *tmin_NAMEMODEL*: 32 (20 in the case of RCP scenarios) columns for daily maximum and minimum temperatures of the individual regionalized models. *_NAMES* represents the name of each individual model.

3. Execution of main.R

3.1. Loading of libraries, functions and data

The process begins by loading the libraries required by *main.R*, the source code (*src/functions.R*) with the functions, and the example data. The libraries required by the implemented functions are automatically loaded when they are executed.

```
# Load libraries and source file

require(Hmisc)
library(readr)

source("src/functions.R")

############### OBTAIN ET0 based on Hargreaves calibrated equation##################

# 1. load data. Example for weather station 7002A-------------------

TASdatahist = data.frame(read_delim("data/TASdatahist-7002A.csv", ";",
    escape_double = FALSE, col_types = cols(Date = col_date(format = "%Y-%m-%d"))))

TASdatarcp45 = data.frame(read_delim("data/TASdatarcp45-7002A.csv", ";",
    escape_double = FALSE, col_types = cols(Date = col_date(format = "%Y-%m-%d"))))
```

```
TASdatarcp85 = data.frame(read_delim("data/TASdatarcp85-7002A.csv", ";",
    escape_double = FALSE, col_types = cols(Date = col_date(format = "%Y-%m-%d"))))

names(TASdatahist)
```

```
##  [1] "idest"              "Date"               "tmin_ref"
##  [4] "tmax_ref"           "tmin_ACCESS1_0"     "tmin_ACCESS1_3"
##  [7] "tmin_bcc_csm1_1"    "tmin_bcc_csm1_1_m" "tmin_BNU_ESM"
## [10] "tmin_CMCC_CESM"     "tmin_CMCC_CM"       "tmin_CMCC_CMS"
## [13] "tmin_CNRM_CM5"      "tmin_inmcm4"        "tmin_IPSL_CM5A_MR"
## [16] "tmin_MIROC_ESM"     "tmin_MIROC5"        "tmin_MPI_ESM_LR"
## [19] "tmin_MPI_ESM_MR"    "tmin_MRI_CGCM3"     "tmax_ACCESS1_0"
## [22] "tmax_ACCESS1_3"     "tmax_bcc_csm1_1"    "tmax_bcc_csm1_1_m"
## [25] "tmax_BNU_ESM"       "tmax_CMCC_CESM"     "tmax_CMCC_CM"
## [28] "tmax_CMCC_CMS"      "tmax_CNRM_CM5"      "tmax_inmcm4"
## [31] "tmax_IPSL_CM5A_MR" "tmax_MIROC_ESM"     "tmax_MIROC5"
## [34] "tmax_MPI_ESM_LR"    "tmax_MPI_ESM_MR"    "tmax_MRI_CGCM3"
```

3.2. ET_0 estimation

The ET_0 estimation begins with the creation of the variables needed:

```
# 2. Parameters to estimate Hargreaves calibrated ET0 according to
# Gomariz-Castillo et al. (2018)

latitude = 37.403
a = 0.08233651
b = 1.1202406

# 3. Define variable names----------------------------------------------
# Individual regionalized models for HISTORICAL and RCP scenarios

modhist <- c("ref", "ACCESS1_0", "ACCESS1_3", "bcc_csm1_1", "bcc_csm1_1_m",
    "BNU_ESM", "CMCC_CESM", "CMCC_CM", "CMCC_CMS", "CNRM_CM5", "inmcm4",
    "IPSL_CM5A_MR", "MIROC_ESM", "MIROC5", "MPI_ESM_LR", "MPI_ESM_MR",
    "MRI_CGCM3")

modrcp <- c("ACCESS1_0", "bcc_csm1_1", "BNU_ESM", "CNRM_CM5", "inmcm4",
    "MIROC_ESM", "MIROC5", "MPI_ESM_LR", "MPI_ESM_MR", "MRI_CGCM3")
```

and then ET_0 is estimated:

```
# 4. Obtain ET0 for all scenarios and reference data----------------

# HISTORICAL scenario:
for (name in modhist) {
    tmax = TASdatahist[, paste0("tmax_", name)]
    tmin = TASdatahist[, paste0("tmin_", name)]
    day = TASdatahist[, "Date"]
    TASdatahist[, paste0("ET0_", name)] = fun.ETPHargreaves(latitude = latitude,
        tmax = tmax, tmin = tmin, day = day, Corr = T, a = a, b = b)$ET0h
}
```

```r
# RCP45 scenario:
for (name in modrcp) {
    tmax = TASdatarcp45[, paste0("tmax_", name)]
    tmin = TASdatarcp45[, paste0("tmin_", name)]
    day = TASdatarcp45[, "Date"]
    TASdatarcp45[, paste0("ET0_", name)] = fun.ETPHargreaves(latitude = latitude,
        tmax = tmax, tmin = tmin, day = day, Corr = T, a = a, b = b)$ET0h
}

# RCP85 scenario:
for (name in modrcp) {
    tmax = TASdatarcp85[, paste0("tmax_", name)]
    tmin = TASdatarcp85[, paste0("tmin_", name)]
    day = TASdatarcp85[, "Date"]
    TASdatarcp85[, paste0("ET0_", name)] = fun.ETPHargreaves(latitude = latitude,
        tmax = tmax, tmin = tmin, day = day, Corr = T, a = a, b = b)$ET0h
}

# Head results for HISTORICAL scenario
head(TASdatahist[, c("idest", "Date", paste0("ET0_", modhist))])
```

```
##   idest       Date   ET0_ref ET0_ACCESS1_0 ET0_ACCESS1_3 ET0_bcc_csm1_1
## 1 7002A 1970-01-01 0.9891439      1.304151      1.547208       1.592106
## 2 7002A 1970-01-02 0.7800518      1.457662      1.694668       1.457720
## 3 7002A 1970-01-03 0.9708450      1.651398      1.312290       1.584194
## 4 7002A 1970-01-04 1.3083929      1.609789      1.734931       1.452008
## 5 7002A 1970-01-05 1.3577167      1.327447      1.544796       1.556691
## 6 7002A 1970-01-06 1.2713864      1.586144      1.819906       1.542366
##   ET0_bcc_csm1_1_m ET0_BNU_ESM ET0_CMCC_CESM ET0_CMCC_CM ET0_CMCC_CMS
## 1         1.555229    1.705774      1.379708    1.848618     1.638145
## 2         1.792117    1.571460      1.436345    1.408960     1.597194
## 3         1.658055    1.883353      1.616259    1.619113     1.608937
## 4         1.751153    1.718842      1.420707    1.472052     1.565651
## 5         1.696524    1.709918      1.579647    1.471325     1.792360
## 6         1.682219    1.645462      1.438202    1.496908     1.677984
##   ET0_CNRM_CM5 ET0_inmcm4 ET0_IPSL_CM5A_MR ET0_MIROC_ESM ET0_MIROC5
## 1     1.359068   1.531080         1.449969     1.2343213   1.645961
## 2     1.383715   1.590046         1.459092     1.4078716   1.547270
## 3     1.514514   1.673734         1.269787     0.9411345   1.518836
## 4     1.551778   1.804108         1.235224     1.2639249   1.576748
## 5     1.490050   1.515664         1.285748     1.6958259   1.645766
## 6     1.101553   1.615962         1.502136     1.8093422   1.703179
##   ET0_MPI_ESM_LR ET0_MPI_ESM_MR ET0_MRI_CGCM3
## 1       1.480321       1.456074      1.558302
## 2       1.504158       1.299603      1.750383
## 3       1.509623       1.675975      1.577835
## 4       1.366665       1.670727      1.662619
## 5       1.693015       1.574855      1.773966
## 6       1.410607       1.365253      1.652226
```

3.3. Evaluation of individual model series and multi-model ensembles

Firstly, the information to be included in the generation and testing process of the series is selected, and the information regarding calibration-validation and testing is prepared.

The function calibrates the parameters of the machine learning (RF and SVM) ensembles. A k-folds CV is prepared for this purpose. In this case each fold corresponds to a decade, but the function can be modified to perform other types of partitions; however, the computation time can increase substantially.

```
############### EVALUATE SERIES AND OBTAIN ENSEMBLE MODELS FROM ETO DATA#############

# 1. Select data for ensemble and rename columns--------------------

modhist <- modhist[-1]  # Delete ref value

datahist = TASdatahist[, c("idest", "Date", "ET0_ref", paste0("ET0_", modhist))]
names(datahist) = c("idest", "Date", "ref", modhist)

datarcp45 = TASdatarcp45[, c("idest", "Date", paste0("ET0_", modrcp))]
names(datarcp45) = c("idest", "Date", modrcp)

datarcp85 = TASdatarcp85[, c("idest", "Date", paste0("ET0_", modrcp))]
names(datarcp85) = c("idest", "Date", modrcp)

# 2. Divide data into training and test data-------------------------
# Case of 2/3 for training and 1/3 test data

timeseries = seq(min(datahist$Date), max(datahist$Date), by = "day")

CalIni = min(timeseries)
CalEnd = timeseries[(length(timeseries) * (2/3))] + 1
TestIni = CalEnd + 1
TestEnd = max(timeseries)

# Column for K-Folds
datahist$year2 = paste0(substr(datahist[, "Date"], 3, 3), "0")
```

The execution of the ensemble function with the above data has a relatively high computational and time cost; for example, in the case of the HISTORICAL scenario, the number of rows (days) is 10957; alternatively, it can be executed for a smaller time window. This example works with an interval of 3 years (1096 rows), although the length is small and not representative, it can serve to check the code execution.

In this case, the first two years will be the calibration data and the last the test data. In the case of machine learning based ensembles, k-folds for CV will include one year each (k=2).

```
############### EVALUATE SERIES AND OBTAIN ENSEMBLE MODELS FROM ETO DATA#############

# 3. Select data for years 1990 to 1993----------------------------

timeseries3 = seq(as.Date("1990-01-01"), as.Date("1992-12-31"), by = "day")
datahist3 = datahist[datahist$Date %in% timeseries3, ]

# 4. divide data into training and test data-------------------------
# Case of 2/3 for training and 1/3 test data

CalIni3 = min(timeseries3)
CalEnd3 = timeseries3[(length(timeseries3) * (2/3))]
TestIni3 = CalEnd3 + 1
TestEnd3 = max(timeseries3)
```

```r
# Column for K-Folds
datahist3$year2 = substr(datahist3[, "Date"], 3, 4)
```

Finally, the function is executed. In the example, 3 cores will be used to parallelize the code. The parameter *numcores* can be modified to use the most appropriate number of cores in each case.

```r
# 5. Generate ensembles----------------------------------------------

# Alternative for three years (for test code)
ensemble3 = fun.ensembles(datahist = datahist3, datarcp1 = datarcp45, datarcp2 = datarcp85,
    ref = "ref", date = "Date", CalIni = CalIni3, CalEnd = CalEnd3, TestIni = TestIni3,
    TestEnd = TestEnd3, modhist = modhist, modensembles = modrcp, groupK = "year2",
    parallel = T, numcores = 3, est.caltest = FALSE)

names(ensemble3)
```

```
## [1] "datahist"        "datarcp1"        "datarcp2"        "validation"
## [5] "validation_cal"  "validation_test" "importance"      "models"
```

```r
ensemble3$validation_test
```

```
##            Model    RMSE PBIAS      NSE      R2     KGE
## 1      ACCESS1_0 2.57936 -41.1 -0.21162 0.77570 0.33970
## 2      ACCESS1_3 2.53478 -39.0 -0.17011 0.68746 0.35364
## 3     bcc_csm1_1 2.55993 -39.7 -0.19343 0.69798 0.34720
## 4   bcc_csm1_1_m 2.42271 -37.7 -0.06892 0.71572 0.40464
## 5        BNU_ESM 2.48134 -37.5 -0.12128 0.62912 0.39056
## 6      CMCC_CESM 2.47028 -38.8 -0.11131 0.71140 0.40060
## 7        CMCC_CM 2.45939 -39.0 -0.10154 0.74712 0.39896
## 8       CMCC_CMS 2.53104 -39.8 -0.16665 0.70569 0.37972
## 9       CNRM_CM5 2.52933 -39.8 -0.16507 0.75902 0.35205
## 10        inmcm4 2.39653 -38.2 -0.04595 0.77972 0.41005
## 11  IPSL_CM5A_MR 2.54947 -40.3 -0.18371 0.76385 0.34519
## 12     MIROC_ESM 2.70318 -41.1 -0.33074 0.55487 0.32890
## 13        MIROC5 2.42524 -38.8 -0.07116 0.77988 0.40470
## 14    MPI_ESM_LR 2.51746 -39.5 -0.15417 0.71661 0.37294
## 15    MPI_ESM_MR 2.50511 -39.8 -0.14287 0.74529 0.38349
## 16     MRI_CGCM3 2.57491 -40.7 -0.20745 0.72572 0.35737
## 17            SA 2.47421 -39.6 -0.11485 0.85884 0.34871
## 18           MED 2.43522 -39.0 -0.07999 0.87168 0.35805
## 19            TA 2.43522 -39.0 -0.07999 0.87168 0.35805
## 20           OLS 0.93889  -7.5  0.83946 0.86791 0.88903
## 21           LAD 0.94109  -7.3  0.83871 0.86550 0.88871
## 22           CLS 2.39862 -38.4 -0.04777 0.80731 0.39966
## 23          EIG1 2.47529 -39.6 -0.11582 0.85776 0.34840
## 24          EIG2 1.41118  -7.5  0.63734 0.85720 0.47713
## 25           BMA 0.93175  -7.0  0.84190 0.86705 0.89265
## 26            RF 0.85766  -4.3  0.86604 0.87599 0.91373
## 27           SVR 0.99561  -6.9  0.81948 0.84623 0.89230
```

The following section describes the help functions and explains the results generated.

4. Funcion documentation (source/functions.R)

4.1. fun.ETPHargreaves

Description

Estimates the reference crop evapotranspiration (ET_0) using the Hargreaves-Samani equation.

Usage

fun.hargreaves(tmax, tmin, latitude, CCoef=0.0135, CT=17.78, Rs=NULL, day, KT=0.17, HE=0.5, Corr=FALSE, a=00, b=1)

Arguments

- **latitude**: Latitude of the weather station (in decimal degrees).
- **tmax** (ºC day^-1): Time series of daily maximum temperatures.
- **tmin**: (ºC day^-1): Time series of daily minimum temperatures.
- **CCoef**: Parameter of Hargreaves.
- **CT**: Empirical temperature coefficient.
- **Rs** (MJ m^-2 day^-1): Time series of mean daily incoming solar radiation. If it is null, Rs is obtained from the the extraterrestrial radiation (Ra), which requires the day of the year and the latitude of the observation point.
- **day**: Vector of dates (required to approximate Rs using Ra).
- **KT**: Empirical temperature coefficient to obtain Rs from Ra.
- **HE**: Hargreaves empirical exponent to obtain Rs from Ra.
- **Corr**: corrects ET_0, using a linear regression between Hargreaves ET_0 and Penman-Monteith FAO ET_0, as proposed in Allen et al. (1994).

Details

This function estimates by default the daily ET_0 from the Hargreaves and Samani (1985) equation using the standard parameters and approximating Rs from the clarity index Ra (Samani 2000).

However, it is possible to include Rs if it is available. It is also possible to change the default equation changing the values of all the default parameters of the equation, and to include the a and b parameters: the intercept and slope obtained from the linear regression analysis between the estimated and reference ET_0 values, as proposed in Allen et al. (1994) and, in the study area, in Gomariz-Castillo, Alonso-Sarria, and Cabezas-Calvo-Rubio (2018).

Value

The function generates a data frame containing the following columns: **ET0h**: Estimation of daily ET_0 (mm m^-2 day^-1). **Rs**: Rs values used as input or estimated from Ra.

4.2. fun.ensembles

Description

This function performs multi-model ensemble methods (MMEs) to combine predictions of a variable (e.g. reference evapotranspiration) from different individual models. Eleven MMEs are incorporated in this version: three simple ensembles, four regression-based ensembles, two geometry-based ensembles and two machine learning based ensembles.

Usage

fun.ensembles(datahist, datarcp1, datarcp2=NULL, ref, date, CalIni, CalEnd, TestIni,TestEnd, modhist, modensembles, groupK, parallel=TRUE, numcores=2, est.caltest = FALSE)

Arguments

- **datahist**: Data frame with the HISTORICAL scenario input data.
- **datarcp1**: Data frame with the climate change scenario to forecast.
- **datarcp2**: A second optional scenario to forecast.
- **ref**: Name of the column with the reference data to calibrate and validate de MMEs.
- **date**: Name of the column with the dates (Date class).
- **CalIni** Start date of the calibration data (Date class).
- **CalEnd**: End date of the calibration data (Date class).
- **TestIni**: Start date of the test data (Date class).
- **TestEnd**: End date of the test data (Date class).
- **modhist**: Character vector with the names of the individual models (column names) included in the HISTORICAL scenario. It can contain more models than those used to build the ensembles (the objective in this case is to evaluate their goodness of fit, but not to use them as ensemble members).
- **modensembles**: Vector with the names of the individual models (column names). It must be a subset of **modhist**.
- **groupK**: The column that indicates which rows will be used to validate and which to calibrate.
- **parallel**: Indicates if the process will be parallelized (TRUE) or executed in a single core (FALSE).
- **numcores**: If **parallel=TRUE**, number of cores to be used.
- **est.caltest**: Boolean to indicate if the results of the series obtained with the model will be exported to validate.

Details

This function obtains multi-model ensemble models based on a HISTORICAL scenario. To do this, it calibrates and tests the models by partitioning the input matrix of the HISTORICAL scenario. Once the models are estimated, it generates a forecast for a hypothetical scenario (and optionally a second scenario) and estimates the goodness-of-fit and error statistics of the complete models, both in calibration and test. In addition, it returns the goodness-of-fit and error statistics for individual members, and the relative importance re-scaled from 0-100 for random forest and support vector regression.

Simple ensembles (simple average, trim average and median forecast combinations), regression based ensembles (ordinary least squares, constrained least squares and least absolute deviation forecast combinations) and geometric based ensembles (standard eigenvector and bias-correct eigenvector forecast combinations) are implemented in the R package *GeomComb* (Weiss and Roetzer 2016).

Bayesian Model Averaging (BMA) analysis is implemented in the R package *BMS* (Zeugner and Feldkircher 2015).

The two machine learning based ensembles included are Random Forest (RF) and Support Vector Regression (SVR). RF is implemented in the R package *randomForest* (Liaw and Wiener 2002). SVR is implemented in R package *kernlab* (Karatzoglou et al. 2004). The parameter calibration of the last two methods was carried out using the R package *caret* (Kuhn and Johnson 2013).

Value

The function generates a list object containing the following components:

- **datahist**: Data frame with this the series generated in the HISTORICAL scenario, including the input individual members and, if **est.caltest=TRUE**, the series generated by the validation models (series estimated in calibration and predicted in validation; the column names will include the suffix *caltest*).

- **datarcp1**: Data frame with the predictions in the analyzed scenario.

- **datarcp2**: Optional second data frame with the predictions in the second analyzed scenario.

- **validation**: Validation table of the complete models, including the model name and the goodness of fit statistics: root-mean-square error (RMSE), percent bias (PBIAS), Nash-Sutcliffe Efficiency index (NSE), coefficient of determination (R^2) and Kling-Gupta Efficiency index (KGE).

- **validation_cal**: As the previous table but estimating the goodness of fit statistics with the validation model and evaluating the calibration fit.
- **validation_val**: As the previous table but estimating the goodness of fit statistics with the validation model and evaluating the validation fit.
- **models**: List with the estimated models. Its structure varies according to the models classes (see details).
- **importance**: Re-scaled relative importance of the individuals models acting as predictors in the RF and SVR ensembles.
- **est.caltest**: It is also possible to include as as result the prediction results of the models estimated to validate, generating series in calibration and test.

5. References

AEMET. 2020. "Proyecciones climáticas para el siglo XXI. Plan Nacional de Adaptación al Cambio Climático (PNACC)." http://www.aemet.es/es/serviciosclimaticos/cambio_climat.

Allen, R G, M Smith, M Perrie, and L S Pereira. 1994. "An update for the calculation of reference evapotranspiration." *ICID Bulletin* 43(2): 35–92.

Amblar Francés, P., M. J. Casado Calle, A. Pastor Saavedra, P. Ramos Calzado, and E. Rodríguez Camino. 2017. *Guía de Escenarios Regionalizados de Cambio Climático Sobre España a Partir de Los Resultados Del Ipcc-Ar5*. Madrid, Spain: AEMET.

Gomariz-Castillo, Francisco, Francisco Alonso-Sarria, and Francisco Cabezas-Calvo-Rubio. 2018. "Calibration and Spatial Modelling of Daily et 0 in Semiarid Areas Using Hargreaves Equation." *Earth Science Informatics* 11 (3): 325–40. https://doi.org/10.1007/s12145-017-0327-1.

Hargreaves, George H, and Zohrab A Samani. 1985. "Reference Crop Evapotranspiration from Temperature." *Applied Engineering in Agriculture* 1 (2): 96–99.

Karatzoglou, Alexandros, Alex Smola, Kurt Hornik, and Achim Zeileis. 2004. "kernlab - An S4 Package for Kernel Methods in R." *Journal of Statistical Software* 11 (9). https://doi.org/10.18637/jss.v011.i09.

Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling*. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-6849-3.

Liaw, A, and M Wiener. 2002. "Classification and Regression by randomForest." *R News* 2 (3): 18–22. https://cran.r-project.org/doc/Rnews/.

Samani, Z A. 2000. "Estimating Solar Radiation and Evapotranspiration Using Minimum Climatological Data." *Journal of Irrigation and Drainage Engineering* 126 (4): 265–67. https://doi.org/10.1061/(ASCE)0733-9437(2000)126:4(265).

Weiss, Christoph E, and Gernot R Roetzer. 2016. "GeomComb: (Geometric) Forecast Combination Methods." https://cran.r-project.org/package=GeomComb.

Zeugner, Stefan, and Martin Feldkircher. 2015. "Bayesian Model Averaging Employing Fixed and Flexible Priors: The BMS Package for R." *Journal of Statistical Software* 68 (4). https://doi.org/10.18637/jss.v068.i04.