

# Package ‘reddPrec’

October 14, 2022

**Type** Package

**Title** Reconstruction of Daily Data - Precipitation

**Version** 0.4.0

**Author** Roberto Serrano-Notivoli

**Maintainer** Roberto Serrano-Notivoli <rs@unizar.es>

**Description** Computes quality control for daily precipitation datasets,  
reconstructs the original series by estimating precipitation in missing values,  
creates new series in a specified pair of coordinates and creates grids.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Depends** snowfall, fields

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-10-17 08:34:54 UTC

## R topics documented:

cleaned . . . . .	2
filled . . . . .	2
gapFilling . . . . .	3
gridPcp . . . . .	4
points . . . . .	5
preci . . . . .	6
qcPrec . . . . .	6
sts . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

cleaned

*Daily filtered of suspect precipitation data example*

---

**Description**

Daily filtered of suspect precipitation data example. One month from '2001-01-01' to '2001-01-31' of 48 stations.

**Usage**

```
data("precipDataset")
```

**Format**

A matrix with daily precipitation data without suspect values of five years from '2001-01-01' to '2001-01-31' of 48 stations.

**Examples**

```
data(precipDataset)
```

---

filled

*Daily reconstructed precipitation data example*

---

**Description**

Daily reconstructed precipitation data of one month from '2001-01-01' to '2001-01-31' of 48 stations.

**Usage**

```
data("precipDataset")
```

**Format**

A matrix with daily reconstructed precipitation data of one month from '2001-01-01' to '2001-01-31' of 48 stations.

**Examples**

```
data(precipDataset)
```

---

gapFilling	<i>Daily precipitation reconstruction. Makes predictions to all days and locations.</i>
------------	---

---

### Description

This function uses original data to estimate new predicted values in each location and day.

### Usage

```
gapFilling(prec, sts, inidate, enddate, parallel = TRUE, ncpu = 2, thres = NA)
```

### Arguments

prec	Object of class <code>matrix</code> containing the daily precipitation data. This dataset can be the cleaned <code>RData</code> resulting from <code>qcPrec</code> function. Each column represents one station. The names of columns have to be names of the stations.
sts	Object of class <code>matrix</code> containing the stations info. Must have at least four fields: ID: station identifier; ALT: altitude; X: Longitude in UTM projection (meters); and Y: Latitude in UTM projection (meters). Tabulation separated.
inidate	Object of class <code>Date</code> in format 'YYYY-mm-dd' defining the first day of quality control process
enddate	Object of class <code>Date</code> in format 'YYYY-mm-dd' defining the last day of quality control process
parallel	Logical. If <code>TRUE</code> , parallel computing is activated and the processes will be distributed among the <code>ncpu</code> number of processor cores.
ncpu	Only if <code>parallel = TRUE</code> . Sets the number of processor cores used to parallel computing.
thres	Threshold applied to search nearest stations. If <code>thres=NA</code> the function will search 10 nearest observations without a distance limit. A positive number indicates the threshold in kilometers.

### Value

A new file called `Filled.RData` will be created in working directory. The load of this file (`load('Filled.RData')`) will add a matrix containing the original data with missing values filled with this function. A new days directory will be created in working path with one file per day. Each file contains a `data.frame` with many rows as stations. The columns show the identifier (ID) of each station; the observed value `obs` (if exists); the binomial prediction `predb` (`dry<0.5`; `wet>=0.5`); the raw prediction of precipitation magnitude `pred1` resulting from models; the corrected magnitude prediction `pred2` by binomial prediction (if `dry`, `pred2==0`); the final standardized magnitude prediction `pred3` and the standard error `err` of the model. All variables except `predb` are in the same units as original values (`mm*10`).

### Author(s)

Roberto Serrano-Notivoli

**Examples**

```
#loads example data
data(precipDataset)

#runs function
gapFilling(prec=cleaned,sts=sts,inidate=as.Date('2001-01-01'),
           enddate=as.Date('2001-01-02'),parallel=TRUE,ncpu=2,thres=NA)
```

---

gridPcp

*Gridding daily precipitation.*


---

**Description**

This function uses serially-completed data to estimate new predicted values in each point and day of a given point dataset.

**Usage**

```
gridPcp(filled, points, sts, inidate, enddate, parallel = TRUE, ncpu = 2)
```

**Arguments**

filled	Object of class <code>matrix</code> containing the daily precipitation data. This dataset can be the <code>Filled.RData</code> resulting from <code>gapFilling</code> function. Each column represents one station. The names of columns have to be names of the stations.
points	Object of class <code>matrix</code> containing the points of the new grid info. Must have at least four fields: ID: station identifier; ALT: altitude; X: Longitude in UTM projection (meters); and Y: Latitude in UTM projection (meters). Tabulation separated.
sts	Object of class <code>matrix</code> containing the stations info. Must have at least four fields: ID: station identifier; ALT: altitude; X: Longitude in UTM projection (meters); and Y: Latitude in UTM projection (meters). Tabulation separated.
inidate	Object of class <code>Date</code> in format <code>'YYYY-mm-dd'</code> defining the first day of quality control process
enddate	Object of class <code>Date</code> in format <code>'YYYY-mm-dd'</code> defining the last day of quality control process
parallel	Logical. If <code>TRUE</code> , parallel computing is activated and the processes will be distributed among the <code>ncpu</code> number of processor cores.
ncpu	Only if <code>parallel = TRUE</code> . Sets the number of processor cores used to parallel computing.

**Details**

The precipitation data used to make the grid should be a serially-completed dataset without any missing value. If we use a dataset with gaps, the number of near stations to each point will change along the period and we will introduce inhomogeneities to the final series. We recommend to use the output of `gapFilling` function.

**Value**

A new gridded directory will be created in working path with one file per day. Each file contains a `data.frame` with many rows as points. The columns show the identifier (ID) of each station, the magnitude prediction `pred` and the standard error `err` of the model. All of these values are expressed in the same units as original values ( $\text{mm} \cdot 10$ ).

**Note**

The time computing depends directly on the number of points of the `points` dataset. If you need to fill a large grid we recommend to use a powerful computer with many number of processor cores (to distribute properly the work) and RAM (to process the whole amount of data).

**Author(s)**

Roberto Serrano-Notivoli

**Examples**

```
#loads example data
data(precipDataset)

#runs function (only for two days)
gridPcp(filled=filled, points=points, sts=sts, inidate=as.Date('2001-01-01'),
        enddate=as.Date('2001-01-01'), parallel=TRUE, ncpu=2)
```

---

points

*Information about points of the grid.*

---

**Description**

Information of the all points related to `precip` dataset with an identifier, longitude, latitude and altitude.

**Usage**

```
data("precipDataset")
```

**Format**

A `data.frame` with 676 points on the following 4 variables.

ID identifier of the station.

X longitude of the station.

Y latitude of the station.

ALT altitude of the station.

**Examples**

```
data(precipDataset)
```

---

```
precip          Daily precipitation data example
```

---

**Description**

Daily precipitation data of five years from '2001-01-01' to '2005-12-31' of 48 stations.

**Usage**

```
data("precipDataset")
```

**Format**

A matrix with daily precipitation data of five years from '2001-01-01' to '2005-12-31' of 48 stations.

**Examples**

```
data(precipDataset)
```

---

```
qcPrec          Quality control: Identifies and removes suspect data from the original dataset
```

---

**Description**

This function uses original data to estimate new predicted values and compare them with observations. If exist large differences, removes the original values.

**Usage**

```
qcPrec(prec, sts, inidate, enddate, parallel = TRUE, ncpu = 2,
printmeta = TRUE, thres = NA)
```

### Arguments

<code>prec</code>	Object of class <code>matrix</code> containing the original precipitation data. Each column represents one station. The names of columns have to be names of the stations.
<code>sts</code>	Object of class <code>matrix</code> containing the stations info. Must have at least four fields: ID: station identifier; ALT: altitude; X: Longitude in UTM projection (meters); and Y: Latitude in UTM projection (meters). Tabulation separated.
<code>inidate</code>	Object of class <code>Date</code> in format 'YYYY-mm-dd' defining the first day of quality control process
<code>enddate</code>	Object of class <code>Date</code> in format 'YYYY-mm-dd' defining the last day of quality control process
<code>parallel</code>	Logical. When <code>TRUE</code> , parallel computing is activated and the processes will be distributed among the <code>ncpu</code> number of processor cores.
<code>ncpu</code>	Only when <code>parallel = TRUE</code> . Sets the number of processor cores used to parallel computing.
<code>printmeta</code>	When <code>TRUE</code> , one file per day will be written in subdirectory <code>./meta</code> .
<code>thres</code>	Threshold applied to search nearest stations. If <code>thres=NA</code> the function will search 10 nearest observations without a distance limit. A positive number indicates the threshold in kilometers.

### Details

The process of quality control uses five criteria to flag suspect data. All of them are based on the calculation of reference values (RV) made with the 10 nearest observations (NNS) that day. For this reason, a minimum of 11 available data by day is mandatory. The five criteria are : 1) Suspect data: Observed > 0 & all their 10 NNS == 0; 2) Suspect zero: Observed == 0 & all their 10 NNS > 0; 3) Suspect outlier: Observed is 10 times higher or lower than RV; 4) Suspect wet: Observed == 0, wet probability is over 99%, and predicted magnitude is over 5 litres and 5) Suspect dry: Observed > 5 litres, dry probability is over 99%, and predicted magnitude is under 0.1 litres.

All of these criteria are prepared to work with precipitation in tenths (millimetres\*10).

### Value

A new file called `cleaned.RData` will be created in working directory. The load of this file (`load('cleaned.RData')`) will add a matrix with the original data filtered by quality control. If `printmeta = TRUE`, a new meta directory will be created in working path with one file per day. Each file contains a `data.frame` with many rows as flagged data in that day. The columns show the identifier (ID) of each station; the date; the criteria code through the data was flagged and the removed data. There are five different codes referred to the five criteria: 1 = Suspect data; 2 = Suspect zero; 3 = Suspect outlier; 4 = Suspect wet and 5 = Suspect dry.

### Note

It is possible that many warnings appear at the end of the execution of the function. Probably most of them will be due to a bad fitting in algorithm used. This is because some days in some locations, the nearest stations are so different that they are not enough to explain strongly the precipitation in one location.

**Author(s)**

Roberto Serrano-Notivoli

**Examples**

```
#loads example data
data(precipDataset)

#runs function
qcPrec(prec=preci, sts=sts, inidate=as.Date('2001-01-01'),
       enddate=as.Date('2001-01-02'), parallel=TRUE, ncpu=2, printmeta=TRUE, thres=NA)
```

---

sts

*Information about stations*

---

**Description**

Information of the 48 stations related to precip dataset with an identifier, longitude, latitude and altitude.

**Usage**

```
data("precipDataset")
```

**Format**

A data.frame with 48 observations on the following 4 variables.

ID identifier of the station.

X longitude of the station.

Y latitude of the station.

ALT altitude of the station.

**Examples**

```
data(precipDataset)
```

# Index

- \* **datasets**
  - cleaned, [2](#)
  - filled, [2](#)
  - points, [5](#)
  - preci, [6](#)
  - sts, [8](#)
  
- cleaned, [2](#)
  
- filled, [2](#)
  
- gapFilling, [3](#)
- gridPcp, [4](#)
  
- points, [5](#)
- preci, [6](#)
  
- qcPrec, [6](#)
  
- sts, [8](#)