Package ‘RSAlgaeR’

April 10, 2018

Type Package

Title Builds Empirical Remote Sensing Models of Water Quality Variables and Analyzes Long-Term Trends

Version 1.0.0

Author Carly Hansen

Maintainer Carly Hansen <carlyahansen@gmail.com>

Description Assists in processing reflectance data, developing empirical models using stepwise regression and a generalized linear modeling approach, cross-validation, and analysis of trends in water quality conditions (specifically chl-a) and climate conditions using the Theil-Sen estimator.

License GPL-2

Encoding UTF-8

URL http://github.com/cahhansen/RSAlgae

BugReports https://github.com/cahhansen/RSAlgae/issues

LazyData TRUE

RoxygenNote 6.0.1

Depends R (>= 2.10)

Imports plyr,lubridate,ggplot2,hydroGOF,stats,cvTools,mblm,graphics,utils

NeedsCompilation no

Repository CRAN

Date/Publication 2018-04-10 13:14:47 UTC

R topics documented:

`annual.summary.climate` *summarize climate conditions on an annual basis*

Description

summarize climate conditions on an annual basis

Usage

```
annual.summary.climate(df, datecol, valuecol, parameter)
```

Arguments

`df`	with estimated values, dates, location identifiers
`datecol`	string, name of column with dates
`valuecol`	string, name of column with climate parameter values
`parameter`	string, name of parameter ("Precipitation","Temperature")

Value

list of annual (seasonal) summaries

Examples

```
data(climatedata)
sumdata <- annual.summary.climate(df=climatedata,valuecol="PRCP",datecol="DATE",
parameter="Precipitation")
```

annual.summary.wq *summarize max and mean water quality conditions on an annual basis*

Description

summarize max and mean water quality conditions on an annual basis

Usage

```
annual.summary.wq(df, valuecol, datecol, locationcol)
```

Arguments

df	data frame with estimated values, dates, location identifiers
valuecol	string, name of column with water quality values
datecol	string, name of column with dates (must be date format)
locationcol	string, name of column with location identifiers

Value

dataframe of annual summaries

Examples

```
data(estimatedrecord)
sumdata <- annual.summary.wq(df=estimatedrecord,valuecol="EstChlValue",datecol="ImageDate",
locationcol="StationID")
```

annualtrend.ts *Explore long term annual trends with Theil-Sen Estimator*

Description

Calculates annual linear trend of average values and significance of with Theil-Sen Estimator (used for robust to non-normal data)

Usage

```
annualtrend.ts(record, valuecol, datecol, var, monthlybias = FALSE)
```

Arguments

`record` dataframe with record of estimated water quality

`valuecol` string, name of column with water quality values

`datecol` string, name of column with dates

`var` string, aggregator (e.g. mean, max)

`monthlybias,` calculates annual average using monthly averages (in case of differing numbers of samples for each month)

Value

summary of the Theil-Sen estimator

Examples

```
data(estimatedrecord)
annualtrend.ts(record=estimatedrecord,valuecol="EstChlValue",
datecol="ImageDate",var="mean",monthlybias="TRUE")
```

`apply.mod.seasonal` *apply.mod.seasonal*

Description

Apply GLM to remotely sensed record

Usage

```
apply.mod.seasonal(df, datecol, model, season, threshold)
```

Arguments

`df` dataframe with reflectance values

`datecol` string, name of column with imagery dates

`model` calibrated GLM

`season` vector of months to include in the season

`threshold` numeric value above which is considered unreasonable/noise

Value

dataframe of predicted values and confidence intervals

Examples

```
data(srdataforapplication)
data(utahsummermod)
estdata <- apply.mod.seasonal(df=srdataforapplication,
datecol="ImageDate",model=utahsummermod,season=c("July","August","September"),threshold=500)
```

climate.factor.effect *climate.factor.effect*

Description

Evaluates difference in values based on climate conditions

Usage

```
climate.factor.effect(wqrecord, imagedatecol, valuecol, climaterecord,
  climatevarcol, climatedatecol, maxlag, noevent, alternative = "two.sided",
  overall = TRUE, months = NULL, locationcol = "",
  ylabel = "Average Value")
```

Arguments

`wqrecord`	dataframe with estimated historical record of water quality parameter
`imagedatecol`	string, name of column with the date of the estimate (date of remotely sensed imagery)
`valuecol`	string, name of column with estimated or field-sampled water quality parameter
`climaterecord`	dataframe with climate variables
`climatevarcol`	character, name of climate variable (column) of interest
`climatedatecol`	string, name of column with the date of the climate observation
`maxlag`	numeric, number of days to lag the climate effect
`noevent`	numeric, threshold for whether an event occurred
`alternative`	character string specifying alternative hypothesis ("two.sided","greater","less")
`overall`	boolean, TRUE: all locations, FALSE: by each location. Default is TRUE
`months`	months an optional character string for if the t-test should be month specific
`locationcol`	string, name of column with unique location identifier, used if overall is FALSE
`ylabel`	string, optional label for plot

Value

results of wilcox test for differences in mean values (and, if overall, boxplots of water quality data)

Examples

```
data(estimatedrecord)
data(climatedata)
effectresults <- climate.factor.effect(wqrecord=estimatedrecord,imagedatecol="ImageDate",
valuecol="EstChlValue",climaterecord=climatedata,climatevarcol="TMAX",climatedatecol="DATE",
maxlag=7,noevent=16,months=c("July"))
```

`climatedata` *Climate data*

Description

A dataset containing precipitation and maximum daily temperature for the Provo BYU NOAA Station

Usage

```
climatedata
```

Format

A data frame with 12238 rows and 3 variables:

DATE date of observation

PRCP precipitation volume (mm/day)

TMAX max temperature (degrees C) ...

Source

NOAA Climate Data Archive

`create.model.vars` *create.model.vars*

Description

Create model variables used in model calibration and application

Usage

```
create.model.vars(filename, rowIndex = TRUE)
```

Arguments

`filename`	CSV file with formatted surface reflectance data
`rowIndex`	True or False, Indicates whether the first column in the formatted data file is a row index

Value

dataframe with variables used in model development or application

`cv.model` *cv.model*

Description

Use k-fold cross validation to evaluate the goodness of fit for a model

Usage

```
cv.model(df, valuecol, k, model, gof)
```

Arguments

`df` dataframe, limited to independent/dependent variables

`valuecol` string, name of column with water quality values

`k` numeric, number of folds (will not be used if there are fewer observations than folds)

`model` formula

`gof` string, measure of the goodness of fit (PBIAS, R2, RMSE)

Value

list of training and testing goodness of fit

`estimatedrecord` *Estimated data*

Description

A dataset containing remotely sensed estimates of chlorophyll for Utah Lake

Usage

```
estimatedrecord
```

Format

A data frame with 2041 rows and 5 variables:

ImageDate date of image acquisition

StationID location (corresponding to sampling locations)

EstChlValue estimated chlorophyll value

Lower lower end of confidence interval

Upper upper end of confidence interval ...

Source

derived from models developed by Carly Hansen and Landsat surface reflectance data

formatSRdata *formatSRdata*

Description

Format surface reflectance data

Usage

```
formatSRdata(data, value, imagerydate, samplingdate = "", location, datatype,
  qaband, qa_accept)
```

Arguments

`data`	dataframe of surface reflectance data. Designed to work with reflectance values for bands from Landsat surface reflectance products (Blue, Red, Green, NIR, SWIR1, SWIR2, QA Band) at specific point locations
`value`	string, name of column with water quality parameter values
`imagerydate`	string, name of column for imagery dates (must be date format)
`samplingdate`	string, name of column for sampling dates (only required if dataset is used for calibration, must be date format)
`location`	string, name of column for location identifier
`datatype`	string, "Calibration" or "Estimated"
`qaband`	string, name of column for QA (such as a cloudmask) rating
`qa_accept`	vector, QA classes which are acceptable

Value

dataframe with formatted data

Examples

```
data(srdata)
formattedsrdata <- formatSRdata(data=srdata,value="FieldValue",imagerydate="ImageDate",
samplingdate="SamplingDate",location="StationID",
datatype="Calibration",qaband="CloudMask",qa_accept=c(0,1))
```

`lagpad` *lagpad*

Description

Lags a variable by a time step

Usage

```
lagpad(x, k)
```

Arguments

`x`	climate data
`k`	time step to lag (positive results in a forward shift, negative results in backwards shift)

`modresults` *modresults*

Description

Evaluate Model Performance (R2 and RMSE)

Usage

```
modresults(model, data, value, title = "")
```

Arguments

`model`	GLM model
`data`	data for calibration/evaluation
`value`	string, name of column with water quality values
`title`	string, optional suffix for the title of the plot

Value

prints summary of model and plot of modeled vs. observed

monthlytrend.ts *Explore long term monthly trends with Theil-Sen Estimator*

Description

Calculates annual linear trend of average monthly values and significance of with Theil-Sen Estimator (used for robust to non-normal data)

Usage

```
monthlytrend.ts(record, valuecol, datecol, months, var)
```

Arguments

`record`	dataframe with record of estimated water quality
`valuecol`	string, name of column with values
`datecol`	string, name of column with dates
`months`	list of months
`var`	string, aggregator (e.g. mean, max)

Value

summary of the Theil-Sen estimator

plotrecord *plotrecord*

Description

Plots estimated and observed data

Usage

```
plotrecord(data, datavalue, date, obsdata, obsdatavalue, obsdate, lake = "",
  labels = TRUE, ylab = expression(paste("Chl-a (", mu, "g/L)")))
```

Arguments

`data`	Dataframe with estimated values
`datavalue`	string, name of column with values in estimated dataframe
`date`	string, name of column with date of imagery used for estimating values (must be date class)
`obsdata`	Dataframe with Observed Data

`obsdatavalue`	string, name of column with values in observed dataframe
`obsdate`	string, name of column with date of observation (must be date class)
`lake`	string, Name of Lake
`labels`	optional for plotting
`ylab`	string, label for y axis

Value

plot of estimated and observed data

plotrecord.cal *plotrecord.cal*

Description

Plots estimated record with calibrated data

Usage

```
plotrecord.cal(data, caldata, value, date, location,
  ylab = expression(paste("Chl-a (", mu, "g/L)")))
```

Arguments

`data`	Dataframe with estimated values (value), dates (ImageDate), location identifier
`caldata`	Dataframe with data used in Calibration (value, ImageDate, and Lake column)
`value`	string, name of column with water quality values
`date`	string, name of column with imagery dates
`location`	string, name of column with location identifiers
`ylab`	string, label for y axis

Value

plot of estimated record with data used for calibration

plotrecord.errors *plotrecord.errors*

Description

Plots estimated record with error bars

Usage

```
plotrecord.errors(data, value, date, location,
  ylab = expression(paste("Chl-a (", mu, "g/L)")))
```

Arguments

data	Dataframe with estimated values,dates, location identifiers, lower and upper bounds (lower and upper)
value	string, name of column with water quality values
date	string, name of column with imagery dates
location	string, name of column with location identifiers
ylab	string, label for y axis

Value

plots the estimated record with error bars

srdata *Surface reflectance data from Landsat 5 and 7*

Description

A dataset containing the surface reflectance for locations in Utah Lake from Landsat 5 and 7 (precollection dataset) in Google Earth Engine.

Usage

```
srdata
```

Format

A data frame with 215 rows and 16 variables:

Blue reflectance in the blue band

CloudMask classes used for masking clouds/haze (0 or 1 are clear/water)

FieldValue sampled or observed chlorophyll a value

AbsDiffInDays calculated difference between imagery date and sampling date

Green reflectance in the green band

ImageDate date of imagery acquisition

ImageName name of Landsat scene

Method method used in sampling

NIR reflectance in the Near infrared band

Organization agency responsible for collecting sample data

Red reflectance in the red band

SWIR1 reflectance in the short wave infrared 1 band

SWIR2 reflectance in the short wave infrared 2 band

SamplingDate date of sample collection

Sensor sensor used to measure surface reflectance

StationID location of sample ...

Source

Utah Division of Water Quality and Landsat

`srdataforapplication` *Surface reflectance data from Landsat 5 and 7*

Description

A dataset containing the surface reflectance for locations in Utah Lake from Landsat 5 and 7 (precollection dataset) in Google Earth Engine, used for applying models.

Usage

```
srdataforapplication
```

Format

A data frame with 2313 rows and 25 variables:

Blue reflectance in the blue band

Green reflectance in the green band

ImageDate date of imagery acquisition

NIR reflectance in the Near infrared band

Red reflectance in the red band

SWIR1 reflectance in the short wave infrared 1 band

SWIR2 reflectance in the short wave infrared 2 band

StationID location of sample

Green_Blue reflectance in the green/blue band

Red_Blue reflectance in the red/blue band

Red_Green reflectance in the red/green band

Red_NIR reflectance in the red/NIR band

Red_SWIR1 reflectance in the red/SWIR1 band

Green_SWIR1 reflectance in the green/SWIR1 band

Blue_SWIR1 reflectance in the blue/SWIR1 band

Red_SWIR2 reflectance in the red/SWIR2 band

Green_SWIR2 reflectance in the green/SWIR2 band

Blue_SWIR2 reflectance in the blue/SWIR2 band

NIR_SWIR1 reflectance in the nir/swir1 band

NIR_SWIR2 reflectance in the nir/swir2 band

NIR_Blue reflectance in the nir/blue band

NIR_Green reflectance in the nir/green band

NDVI NDVI

avgRGB average of reflectance in the visible bands

avgSWIR average of reflectance in the SWIR bands ...

Source

Landsat

step.model *step.model*

Description

Use stepwise regression to parameterize model

Usage

```
step.model(data, imagerydate, value, modelvariables, timewindow, season,
  stepdirection, print.on = TRUE)
```

Arguments

`data`	dataframe, formatted calibration data (model variables: field data and surface reflectance values)
`imagerydate`	string, name of column with dates of imagery
`value`	string, name of column with water quality values
`modelvariables`	vector of strings with the names of columns for bands to consider
`timewindow`	numeric, number of days to allow for near coincidence
`season`	vector, months to include in model
`stepdirection`	string, direction for stepwise regression ("backward","both","forward")
`print.on`	boolean, option to print the results of the model (default is TRUE)

Value

list with the stepwise model and the modeled values

utahsummermod *Example chlorophyll estimation model*

Description

A dataset containing model information for Utah Lake - summer

Usage

```
utahsummermod
```

Format

A list of 30 items:

coefficients

residuals

fitted.values

effects

R

rank

qr

family

linear.predictors

deviance

aic

null.deviance

iter

weights

prior.weights

df.residual

df.null

y

converged

boundary

model

call

formula

terms

data

offset

control

method

contrasts

xlevels ...

Source

developed by Carly Hansen

Index