Package 'hydroState'

August 6, 2024

Title Hidden Markov Modelling of hydrological state change

Version 0.1.0.2

Maintainer Tim Peterson <tim.peterson@monash.edu>

Depends R (>= 3.4.2)

Description HydroState idenifies regime changes in streamflow runoff not explained by variations in precipitation. The package allows a flexible set of Hidden Markov Models of annual, seasonal or monthly streamflow runoff to be built that includes precipitation as a predictor of runoff. Suites of models can be built for a single site, ranging from one to three states and each with differing combinations of error models and autocorrelation terms, allowing the most parsimineous model (by AIC) to easily be idenified. The entire package is written in R S4 object oriented code. See Peterson TJ, Saft M, Peel MC & John A (2021), Watersheds may not recover from drought, Science, DOI: 10.1126/science.abd5085

Imports DEoptim, sn, truncnorm, diagram

BugReports https://github.com/peterson-tim-j/HydroState/issues

URL https://github.com/peterson-tim-j/HydroState

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

ByteCompile true

Collate 'abstracts.R'
'parameters.R'
'Qhat.boxcox.R'
'Qhat.burbidge.R'
'Qhat.log.R'
'Qhat.none.R'
'QhatModel.homo.normal.linear.R'
'QhatModel.homo.normal.linear.AR1.R'
'QhatModel.homo.gamma.linear.R'
'QhatModel.homo.gamma.linear.AR1.R'
'QhatModel.homo.normal.linear.AR2.R'
'QhatModel.homo.gamma.linear.AR2.R'
'QhatModel.homo.normal.linear.AR3.R'
'QhatModel.homo.gamma.linear.AR3.R'

'QhatModel.homo.skewedNormal.linear.R'
'QhatModel.homo.skewedNormal.linear.AR1.R'
'QhatModel.homo.skewedNormal.linear.AR2.R'
'QhatModel.homo.skewedNormal.linear.AR3.R'
'QhatModel.subAnnual.homo.gamma.linear.R'
'QhatModel.subAnnual.homo.gamma.linear.AR1.R'
'QhatModel.subAnnual.homo.gamma.linear.AR2.R'
'QhatModel.subAnnual.homo.gamma.linear.AR3.R'
'markov.annualHomogeneous.R'
'hydroState.R'
'hydroState.allModels.R'
'hydroState.subAnnual.allModels.R'
'markov.annualHomogeneous.flickering.R'
'wrapper.R'

R topics documented:

`hydroState-package` *Overview of methods and procedures*

Description

`hydroState` provides methods to construct and evaluate hidden Markov models (HMM) of annual or subannual streamflow with precipitation as a predictor. The state of the relationship between these observations is evaluated overtime. The package contains a default `hydroState` model to evaluate probable shifts in this rainfall-runoff relationship or the model can be constructed with various items to better define the relationship as discussed below. Once the default or constructed model is built (`buildModel`), the model is fitted (`fitModel`) to determine the most likly rainfall-runoff state at each time-step. To assess the adequacy of the fit, the residuals are plotted (`plot.residuals`), and an adequate fit requires the residuals to be normaly distributed and not have any correlation or trends. The resulting runoff states from the fitted model can then be evaluated overtime (`plot.states`) and even exported (`get.states`) with the state values, confidence intervals, and conditional probabilities of each state.

A `hydroState` model operates in S4, object oriented programming, and constructing a model requires the selection of three objects. The first object is the `data.transform`, or 'Qhat' object, from `select.transform` that offers various transformaitons of the observations including: 'boxcox',

'log', 'burbidge', or 'none'. The second object is the `state.model`, or 'QhatModel' object, from `select.stateModel` that offers a variety of items to better define the rainfall-runoff relationship including: the state number (1, 2, or 3), residual distribution ('normal', 'gamma', or 'truc.normal'), and degree of auto-correlation ('AR1', 'AR2', or 'AR3'). Furthermore `select.stateModel` provides an option to select the parameters in the model expected to shift (`state.shift.parameters`) and apply seasonal variation within model parameters (`seasonal.paraneters`) if monthly data is provided. The third object is the form of the `Markov`, or 'markov' object, from `select.Markov` which currently only provides a homogeneous Markov model. Further details on the selection of options within these objects for constructing a `hydroState` model are explained within each topic.

There is an additional option to construct all possible types of models using the `buildModelAll`, and compare them using the same `fitModel` function. The most likely model is selected based on the AIC where the best model will have the lowest AIC. To begin, it is recommended to evaluate the default model at first with one state and again with two states. This documentation is organized with four sections that define the general workflow for using `hydroState`: Build - Fit - Review - Evaluate.

I. Build

`buildModel`	build hydroState model
`select.transform`	select transformation of observations
`select.stateModel`	select type of state model
`select.Markov`	select form of Markov
`buildModelAll`	build all possible models

II. Fit

`fitModel`	fit built hydroState model(s)

III. Review

`plot.residuals`	plot residuals
`get.residuals`	get residuals

IV. Evaluate

`setInitialYear`	set initial year for assigning state names
`plot.states`	plot states
`get.states`	get states

Authors

Except where indicated otherwise, the methods and functions in this package were written by Tim Peterson.

Acknowledgments

...

`buildModel` *Builds hydroState model*

Description

`buildModel` builds a hydrostate model with either a default `stateModel` or the `stateModel` can be specified with options from `select.stateModel`. After the model is built, the hydroState model is ready to be fitted with `fitModel`

Usage

```
buildModel(
  input.data = data.frame(year = c(), flow = c(), precip = c()),
  data.transform = NULL,
  stateModel = NULL,
  Markov = NULL
)
```

Arguments

`input.data`	dataframe of annual runoff and precipitation observations. For running seasonal models with `seasonal.parameters`, monthly data is required.
`data.transform`	a `Qhat.object` with transformed observations from `select.transform`. If blank, the default uses 'boxcox' to transform observations.
`stateModel`	a `QhatModel.object` from `select.stateModel`. If blank, the default selects a 2-state 'QhatModel.homo.normal.linear' model with a truncated normal error distribution, and allows the intercept 'a0' and standard deviation 'std' to shift as state dependent parameters.
`Markov`	a `markov.model.object` from `select.Markov`. If blank, the default selects a homogeneous Markov model without flickering.

Details

`buildModel`

hydroState operates in S4, object oriented programming, and requires three objects to build a hydroState model. Each object can be selected from `select.transform`, `select.stateModel`, and `select.Markov`; however, if no object is defined a default model is built as discussed in the arguments.

Value

A built hydroState model object ready to be fitted with `fitModel`

buildModelAll *Builds all hydroState models*

Description

`buildModelAll` builds all hydroState models

Usage

```
buildModelAll(ID = "", input.data = streamflow_annual)
```

Arguments

`ID`	character vector of a stream gauge identifier
`input.data`	is a dataframe of annual, monthly, or daily, observations of runoff, precipitation, or concentration. For running seasonal models, monthly values are required.

Details

`buildModelAll`

All hydroState models are build with different QhatModel objects

For an example of how to.. see vignette...

Value

A list of built hydroState models with every combination of QhatModel, ready to be fitted

fitModel *Fit hydroState model*

Description

`fitModel` fits hydrostate model(s) using global optimization by differential evolution DEoptim.

Usage

```
fitModel(model.name = model, pop.size.perParameter = 10, max.generations = 500)
```

Arguments

`model.name`	name of the built hydroState model or name of the list containing all built models
`pop.size.perParameter`	integer that should be greater than or equal to the number of parameters in the model. The default is '10' and is sufficient for all models.
`max.generations`	integer that will stop the optimizer when set number of generations are reached. The default is '500'.

Details

`fitModel`

After a hydroState model object is built, the model is ready to be fitted. The only required input is the given name of the built hydroState model object. `fitModel` works for one built model (`buildModel`) or all (`buildModelAll`). If fitting all models be sure to install and load the parallelly library.

Value

A fitted hydroState model

`get.residuals` *Get residuals*

Description

The normal pseudo residuals are retrieved from the fitted model.

Usage

```
get.residuals(model.name = model)
```

Arguments

`model.name`	name of the fitted hydroState model object.

Details

`get.residuals`

`get.residuals` retrieves residuals from the fitted model and exports them as a data frame.

Value

Data frame of residuals for each time-step

`get.states` *get states*

Description

`get.states` retrieves results from the fitted `hydroState` model.

Usage

```
get.states(model.name = model)
```

Arguments

`model.name`	is the name of the fitted `hydroState` model object.

Details

`get.states`

`plot.states` These results include the time-step (i.e. Year, Month), Viterbi State Number to differentiate states, observations (i.e. stream flow), flow values of the Viterbi state including the 5% and 95% confidence intervals, flow values of the normal state including the 5

Value

data frame of results to evaluate the rainfall-runoff states overtime

---

`plot.residuals` *Plot residuals*

Description

The normal pseudo residuals are plotted for review to check for outliers and validate the fit of the model. It is recommended to ensure the model fit is valid before evaluating results (i.e. `plot.states`). Furthermore, to ensure the multi-state model performs better than the one-state model, it is recommended to visually compare `plot.resdiuals` of both models.

Usage

```
## S3 method for class 'residuals'
plot(model.name = model, do.pdf = FALSE, ID = NULL)
```

Arguments

`model.name`	name of the fitted hydroState model object.
`do.pdf`	option to export residual plots as a pdf. Default is FALSE.
`ID`	character string of catchment identifier (i.e. gauge ID). Default is NULL. Only recommended when do.pdf = TRUE.

Details

`plot.residuals`

`plot.residuals` produces five plots to review and validate the fitted hydroState model. A) Time-series of normal-pseudo residuals to ensure the residuals each year are within the confidence intervals. B) Auto-correlation function (ACF) of normal-pseudo residuals to ensure there is no serial correlation in residuals. Lag spikes should be below confidence interval at each lag (except 0). C) Histogram of uniform-pseudo residuals should show uniform distribution (equal frequency for each residual value) D) Histogram of normal-pseudo residuals should show normal distribution centered on zero and with no skew E) Quantile-Quantile (Q-Q) plot where normal-pseudo residuals vs. theoretical quantities should align on the diagonal line. The last plot contains the Akaike information criterion (AIC) and Shapiro-Wilk p-value. The AIC is an estimator to determine the most parsimonious, best performing model given the number of parameters. When comparing models, the lowest AIC is the best performing model. Shapiro-Wilks test for normality in the residuals and a p-value greater than 0.05 (chosen alpha level) indicates the residuals are normally distributed; the null hypothesis that the residuals are normally distributed is not rejected.

Value

Plots of residuals to evaluate model fit

`plot.states` *plot States*

Description

`plot.states` produces several plots to visualize results of the states overtime. `setInitialYear` is required before `plot.states`.

Usage

```
## S3 method for class 'states'
plot(
  model.name = model,
  ind.variable = TRUE,
  dep.variable = TRUE,
  dep.variable.transformed = TRUE,
  cond.state.prob = TRUE,
  do.pdf = FALSE,
  ID = NULL
)
```

Arguments

`model.name`	is the name of the fitted hydroState model object.
`ind.variable`	option to plot independent variable overtime. Default is TRUE.
`dep.variable`	option to plot dependent variable and states overtime. Default is TRUE.
`dep.variable.transformed`	option to plot transformed dependent variable and states overtime. Default is TRUE.
`cond.state.prob`	option to plot the conditional state probabilities overtime for each state. Default is TRUE.
`do.pdf`	option to export plots as a pdf. Default is FALSE.
`ID`	character string of catchment identifier (i.e. gauge ID). Default is NULL. Only recommended when do.pdf = TRUE.

Details

`plot.states`

`plot.states` produces plots of the results from the fitted hydroState model. The default produces four plots of the A) independent variable, B) dependent variable and states, C) transformed dependent variable and states, and D) conditional state probabilities for each state. These are plotted on the same page, and there is an option to export plots as a pdf to the current working directory. There are also options to only plot one of the four plots.

Value

plots to evaluate rainfall-runoff states overtime along with observations and the conditional probabilities of each state. The data frame includes several items. These include the time-step (i.e. Year, Month), Viterbi State Number to differentiate states, observations (i.e. stream flow), flow values of the Viterbi state including the 5% and 95% confidence intervals, flow values of the normal state including the 5

`select.Markov` `select.Markov` *selects a hidden Markov model*

Description

select.Markov selects a hidden Markov model.

Usage

```
## S3 method for class 'Markov'
select(func = "annualHomogeneous", transition.graph = matrix(TRUE, 2, 2))
```

Arguments

`func`	character sting with name of Markov model defining the transition between states. The default is 'annualHomogeneous'. Other options include 'annualHomogeneous.flickering'.
`transition.graph`	matrix given the number of states. Default is a 2-state matrix (2 by 2): matrix(TRUE,2,2)

Details

There are several hidden Markov models to choose from with different forms. The default markov model is 'annualHomogeneous' where the transition between states only depends on a sequence of observed prior states.

For an example of how to.. see vignette...

Value

A Markov object with a selected Markov model

`select.stateModel` `select.stateModel`

Description

`select.stateModel` provides various options for constructing the rainfall-runoff relationship. Every `stateModel` depends on a linear base model where precipitation is a function of streamflow. The default model is an annual analysis of this base linear model with state shifts expected in the intercept, a_0, and standard deviation, std, of the rainfall-runoff relationship. `select.stateModel` provides additional adjustments to this model with auto-correlations terms, seasonal parameters for a sub-annual analysis, and even other or multiple `state.shift.parameters`. The number of states and assumed error distribution can also be selected.

Usage

```
## S3 method for class 'stateModel'
select(
  input.data = data.frame(year = c(), flow = c(), precip = c()),
  parameters = list("a0", "a1", "std"),
  seasonal.parameters = list(),
  state.shift.parameters = list("a0", "std"),
  error.distribution = "truc.normal",
  transition.graph = matrix(TRUE, 2, 2)
)
```

Arguments

`input.data`	dataframe of annual runoff and precipitation observations. For running seasonal models with seasonal.parameters, monthly data is required.
`parameters`	character list of parameters to construct state model. Required and default: `a0`, `a1`, `std`. Auto-correlation terms optional: `AR1`, `AR2`, or `AR3`.
`seasonal.parameters`	character list of one or all parameters (`a0`, `a1`, `std`) defined as a sinusoidal function to represent seasonal variation. Requires monthly data. Default is empty list.
`state.shift.parameters`	character list of one or all parameters (`a0`, `a1`, `std`, `AR1`, `AR2`, `AR3`) able to shift as dependent on state. Default is `a_0` and `std`.
`error.distribution`	character name of the distribution in the HMM error. Default is "truc.normal". Others include: "normal" or "gauss"
`transition.graph`	matrix given the number of states. Default is a 2-state matrix (2 by 2): matrix(TRUE,2,2)

Details

There are a selection of items to consider when defining the rainfall-runoff relationship and investigating state shifts in this relationship. hydroState simulates runoff, Q, as being in one of finite states, i, at every time-step, t, depending on the distribution of states at prior time steps. This results

in a runoff distribution for each state that can vary overtime ($\widehat{{}_tQ_i}$). The `stateModel` defines the relationship that is susceptible to state shifts with precipitation, P_t, as a predictor. This takes the form as a simple linear model $\widehat{{}_tQ_i} = f(P_t)$:

$\widehat{{}_tQ_i} = P_t a_0 + a_1$

where a_0 and a_1 are constant parameters. These parameters and the model error, std, are required parameters for every `stateModel`. It is possible the relationship contains serial correlation and would be better defined with an auto-regressive term:

$\widehat{{}_tQ_i} = P_t a_0 + a_1 + AR1\widehat{{}_{t-1}Q}$

where $AR1$ is the lag-1 auto-correlation term. Either, lag-1: $AR1$, lag-2: $AR2$, and lag-3: $AR3$ auto-correlation coefficients are an option as additional parameters to better define the rainfall-runoff relationship. For sub-annual analysis, seasonal.parameters provides the option to assume a sinusoidal function better defines either of the constant parameters or error (a_0, a_1, std) throughout the year, i.e:

$a_0 = a_{0.disp} + a_{0.amp} * sin(2\pi(\frac{M_t}{12} + a_{0.phase}))$

where M_t is an integer month at t. Monthly streamflow and precipitation are required as input.data for the sub-annual analysis.

Once the model parameters are chosen, the `state.shift.parameters` can be selected to investigate shifts based on any or all of the previously chosen parameters ($a_0, a_1, std, AR1, AR2, AR3$). The selected `state.shift.parameters` are state dependent where they are subject to shift in order to better explain the state of streamflow. The default `stateModel` evaluates shifts in the rainfall-runoff relationship with a_0 as a state dependent parameter.

The `error.distribution` of the model is chosen as either normal: "normal", truncated normal: "truc.normal", or Gaussian: "gauss" in order to reduce skew. The default is "truc.normal". The number of possible states in the rainfall-runoff relationship and transition between the states is selected with the transition.graph. The default is a 2-state model in a 2 by 2 matrix with a TRUE transition to and from each state.

Value

A `stateModel`, QhatModel object, ready to for `buildModel`.

`select.transform` *Transforms Observations*

Description

Transforms observations to remove heteroscedasticity.

Usage

```
## S3 method for class 'transform'
select(
  func = "boxcox",
  input.data = data.frame(year = c(), flow = c(), precip = c())
)
```

Arguments

func	is the method of transformation. The default is 'boxcox'. Other options: 'log', 'burbidge', 'none'
input.data	dataframe of annual or monthly runoff and precipitation observations.

Details

select.transform

Often there is skew within hydrologic data. When defining relationships between observations, this skew results in an unequal variance in the residuals, heteroscedasticity. Transforming observations is often required with observations of streamflow, precipitation, and concentration. Qhat provides several options to transform observations. Since the degree of transformation is not typically known, 'boxcox' is the default. Other options include: 'log', 'burbidge', and of course, 'none' when no transformation is performed.

For an example of how to transform observations.. see vignette...

Value

A Qhat object with transformed observations ready for buildModel

---

setInitialYear *Sets state names given initial year*

Description

sets the state names for each time-step relative to the initial year given

Usage

```
setInitialYear(model.name = model, initial.year = streamflow_annual$hy_year[1])
```

Arguments

model.name	name of the fitted hydroState model object.
initial.year	year (YYYY). Default is first year in input.data.

Details

setInitialYear

hydroState assigns names to the computed states. This requires choosing an initial year where the state value from that year will be named 'Normal'. Other state values will be given names relative to the state value in the initial year. The choice of the initial year does not affect results. It is a means to more easily interpret the difference in state values relative to each other. It is best to choose a year based on the question being asked. For example, in testing the impact of drought, a year before the beginning of the drought, 1990, was selected as an initial year when conditions were considered 'Normal' (Peterson TJ, Saft M, Peel MC & John A (2021), Watersheds may not recover from drought, Science, DOI: doi:10.1126/science.abd5085)

Value

A fitted hydroState model object with state names for each time-step

Index