An open software environment for hydrological model assessment and development

F. T. Andrews[d,f,*], B. F. W. Croke[d,e,f], A. J. Jakeman[d,e]

[a] *Integrated Catchment Assessment and Management (iCAM) Centre, Fenner School of Environment and Society, The Australian National University*
[b] *National Centre for Groundwater Research and Training*
[c] *Department of Mathematics, The Australian National University*

Abstract

The **hydromad** (Hydrological Model Assessment and Development) package provides a set of functions which work together to construct, manipulate, analyse and compare hydrological models. The class of hydrological models considered are dynamic (typically at a daily time step), spatially-aggregated conceptual or statistical models. The package functions are designed to fit seamlessly into the R system, and builds on its powerful data manipulation and analysis capabilities. The framework used in the package encourages a separation of model components based on Unit Hydrograph theory; many published models are consistent with this and implementations of several are included. For comparative assessment, model performance can be analysed over time and with respect to covariates to reveal systematic biases. Support has been built in for event-based analysis of data and assessment of model performance. Fit statistics can be defined by choices of (1) temporal scale and aggregation function; (2) weighting and transformation; and (3) reference model. One can define new Soil Moisture Accounting models, routing models, calibration methods, objective functions, and evaluation statistics, while retaining as much of the default framework as is useful. And as the package code is available under a free software licence, one always has the freedom to adapt it as required. Use of the software is demonstrated in a case study of the Queanbeyan River catchment in South-East Australia.

NOTICE: This is the author's version of a work accepted for publication by Elsevier. Changes resulting from the publishing process, including peer review, editing, corrections, structural formatting and other quality control mechanisms, may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *Environmental Modelling and Software* 26 (2011), `doi:10.1016/j.envsoft.2011.04.006`.

Keywords: Model evaluation, Hydrological models, Modelling frameworks, Unit hydrograph, Event separation, R

*Corresponding author
Email address: `felix@nfrac.org` (F. T. Andrews)

An open software environment for hydrological model assessment and development

F. T. Andrews[d,f,*], B. F. W. Croke[d,e,f], A. J. Jakeman[d,e]

[d] Integrated Catchment Assessment and Management (iCAM) Centre, Fenner School of Environment and Society, The Australian National University
[e] National Centre for Groundwater Research and Training
[f] Department of Mathematics, The Australian National University

1. Introduction and motivation

Catchment hydrology is the study of the water cycle at the scale of a drainage basin, focusing for its practical importance on flow at the catchment outlet. Central to this is the use of models to represent processes at the catchment scale and evaluate the implications of hypotheses of different model structures.

In different contexts, the level of detail may vary from simple statistical or conceptual models to complex spatially-distributed or physics-based models (Wheater et al., 1993). In practice all catchment hydrology models need to be calibrated to measured data; model parameters do not have a precise physical analogue when applied at large scales (Wagener et al., 2009). The simplest models of catchment hydrology dynamics are spatially lumped, and conceptual or empirical in their approach. Such models can be used to address questions in terms of aggregate effects, without considering the detail of the processes involved. Often a single dominant mode or process will be identified (Young, 2003). We will focus on this class of models.

Models, although they are usually given names, should not be set in stone. They encode sets of assumptions, which may be more or less valid at different times, places, and scales; and, importantly, for different purposes. Accordingly, models should be tested and evaluated in the unique context of each application (Fenicia et al., 2008).

Given the need for parsimonious models to address a range of management and research problems, many have advocated flexible, iterative model development processes (Fenicia et al., 2008; Wagener et al., 2001). The top-down modelling (Sivapalan et al., 2003; Littlewood et al., 2003) and Data-Based Mechanistic modelling (Young and Ratto, 2008) agendas are particularly prominent, and in fact these are quite similar to each other (Young, 2003). Such processes involve intensive data analysis to drive model development, with detailed comparison and evaluation of model performance. Effective software support for such tasks is crucial. There is a consequent need for flexible software environments for hydrological modelling, tightly linked with data analysis and model analysis methods.

*Corresponding author
Email address: felix@nfrac.org (F. T. Andrews)

Preprint submitted to Environmental Modelling and Software *June 18, 2011*

A major challenge for modelling frameworks is to be flexible enough to support creative problem solving in hydrology; As Savenije (2009) argues, there is an art to hydrology that is often not recognised. However, flexibility ultimately needs to be constrained by rigour. Many authors argue for standardised tests and comparisons of models (e.g. Jakeman et al., 2006; Beven, 2008; Dawson et al., 2007, 2010).

A number of software frameworks for hydrological modelling have been developed and are in active use, such as OMS (Leavesley et al., 2006), PREVAH (Viviroli et al., 2009), FUSE (Clark et al., 2008) and RRMT (Wagener et al., 2001). Most such frameworks are designed for complex problem types, and necessarily restrict the analysis options during model development. An exception is RRMT, the Rainfall Runoff Modelling Toolbox, which is used within the MATLAB environment, and is therefore able to leverage the powerful data manipulation functions it provides. Indeed, the software described in this paper has been influenced by the design of RRMT. However, MATLAB has more of an engineering focus than a statistical focus, and has rather different capabilities compared to primarily statistical software. Furthermore these products are closed-source, which obscures potentially important methodological details, and withholds from users the freedom to adapt the code and share their innovations.

This paper will introduce a software package for top-down modelling of catchment hydrology, **hydromad**. It is based loosely on the unit hydrograph theory of rainfall-runoff modelling, as described in Section 2. **hydromad** is an open-source software package for the R system which is introduced in Section 3. As such it can be used cohesively with workflows based on this increasingly popular software. Section 4 covers the scope of the **hydromad** package and the functions it provides. Two areas of focus for the package, and this paper, are discrete event separation and the design of fit statistics, discussed in Sections 5 and 6 respectively. Section 7 demonstrates how event-based data analysis can be useful in a modelling context. In Section 8 we demonstrate simple conceptual modelling; integral to this is a detailed assessment of model performance, with a view to further model development by discovering systematic biases, in Section 9.

The example we will look at is the Queanbeyan River at Tinderry streamflow gauge, near Canberra in South-Eastern Australia. It has a catchment area of 490 square kilometers and is part of the Upper Murrumbidgee catchment. It has seen much reduced river flow levels in recent years. This catchment is unusual in that it displays marked non-stationary response characteristics, and extended drying periods with intermittent baseflow, making it difficult to model (Kim et al., 2007). Daily streamflow volume records are available from 1966-08-04, and the data used here extend to 2005-12-31. Corresponding estimates of areal rainfall were derived by spatial interpolation from several rain gauges operated by the Australian Bureau of Meteorology and EcoWise. Daily maximum temperature records from Canberra Airport were also used.

2. The hydrological model framework

The class of hydrological models considered here are dynamic, spatially-aggregated conceptual or statistical models. They estimate streamflow at a catchment outlet, given inputs of areal rainfall and potential evaporation (or, more commonly, temperature data as an indicator of this), and potentially other inputs. These inputs and outputs are time series, typically at a daily time step, and extending for many months or years.

As *spatially lumped* models, they do not explicitly represent spatial variation over the catchment area. In particular, the standard formulations do not attempt to model effects of changes in land cover. These models are usually calibrated to a period of observed streamflow, and the parameters defining the modelled relationship between rainfall, evaporation and flow are assumed to be *stationary* in this period.

The model framework used in the **hydromad** package is very general, but encourages a separation of model components based on Unit Hydrograph theory. This implies a two-component structure of a *soil moisture accounting* (SMA) module and a *routing* or *unit hydrograph* module (Figure 1). The SMA module converts rainfall and temperature into *effective rainfall*: the amount of rainfall which eventually reaches the catchment outlet as streamflow (i.e. that which is not lost as evapotranspiration etc). The routing module converts effective rainfall into streamflow, which usually amounts to convolving it with a constant recession curve, the unit hydrograph. This structure is consistent with RRMT (Wagener et al., 2001).

In fact, it is not strictly required to decompose a model this way: a full model could be defined for the SMA component, with the routing component omitted. It is worth noting, also, that the two components are not necessarily simple models but may be composite models, and the whole model may also be arranged into a composite structure.

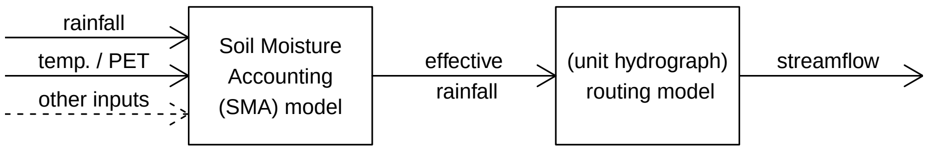

Figure 1: The **hydromad** model framework, based on unit hydrograph theory.

A notable feature of the two-component model structure is that it permits identification of the routing model in a way that is somewhat de-coupled from identification of the soil moisture accounting model. There are a number of different strategies that can be used to calibrate a full hydrological model. The typical approach is a joint optimisation of all parameters. Alternatively the unit hydrograph could be estimated directly from streamflow data (once only, after which it can be fixed), using inverse filtering (Andrews et al., 2010), or average event unit hydrograph estimation (Croke, 2006). Or a simple data-based method could be used in the SMA component to estimate effective rainfall, and this used for a preliminary calibration of the routing model.

A large number of published models are consistent with the unit hydrograph framework. Several of these have already been implemented in the **hydromad** package and are listed in Section 4.

3. R: the synergies of community

R is a language and environment for statistical computing and graphics (R Development Core Team, 2010; Ihaka and Gentleman, 1996). It is based on the high-level S language (Becker, Chambers, and Wilks, 1988), designed for working with data and models. As it aims for maximum power and flexibility, the primary mode of use is typing commands, or writing scripts, rather than pointing and clicking. As free (open source) software, it has become a melting pot for computational statistics, as evidenced by the number of contributed packages available, which has grown exponentially since records began in 2001 (Fox, 2008). This opens up powerful synergies within and between research communities.[1] As Bates (2008) puts it, "I do not think of R as a statistical package or a product. To me, R is a community."

The R community has developed world-class implementations of, for example, generalised additive models (Wood, 2004), time-varying linear models (Petris et al., 2009), forecasting methods (Hyndman, 2010), optimisation algorithms (Ardia and Mullen, 2010), data mining algorithms (Williams, 2009), and statistical graphics (Sarkar, 2008; Wickham, 2009).

The R system thus provides a rich software ecosystem in which a hydrological modelling framework can grow and evolve.

4. The hydromad package

4.1. Core functions

The **hydromad** package provides a set of functions which work together to construct, manipulate, analyse and compare hydrological models. It is intended for:

- defining spatially-lumped hydrological models and fitting them to observed data;
- simulating outputs of these models, including any state variables;
- evaluating and comparing these models: summarising performance by different measures and over time, using graphical displays and statistics;
- straightforward integration with other types of data analysis and model analysis in R, including larger composite modelling studies in which rainfall runoff is just one part.

The main design goals of the package were *flexibility*, to explore research questions and develop improved methods, and *simplicity* of the basic framework. In terms of flexibility, one can define new Soil Moisture Accounting models, new routing models, new calibration methods, new objective functions, and new evaluation statistics, while retaining as much of the default framework as is useful. And as the package code is available under an open source licence, one always has the freedom to adapt it as required.

Particularly strong support has been built in for event-based analysis of data and model performance. This involves isolating relatively discrete events from time series, and analysing statistical properties of the events, rather than the traditional approach of using every time step at which data were recorded.

The package functions are designed to fit seamlessly into the R system. Consistent with other modelling functions in R, many of the functions in **hydromad** are implemented as methods of standard generic functions. A constructor function creates a `hydromad` object, and this can be passed on to methods for calibration, analysis, reporting, etc.

[1]The use of open source code also promotes academic integrity through "reproducible research" (Gentleman and Temple Lang, 2007).

The object encapsulates a model (composed of a SMA model and/or a routing model), with specified parameter values or parameter ranges, along with the data and model outputs.

`hydromad()`	specifies a model, with fixed and/or free parameters
`update()`	modifies the structure, parameters or data of an existing model
`fitBy...()`	calibrates a model to data (several methods)
`predict()`	simulates streamflow (etc.) from a fitted model
`simulate()`	generates a set of models by sampling over parameter ranges
`summary()`	calculates fit statistics and other information
`runlist()`	constructs a named list of models for comparative analysis

Table 1: Core modelling functions provided by the **hydromad** package. See text for more.

The core modelling functions are listed in Table 1. Of course, each of these functions has several arguments. The details are given in help pages, which can be accessed within R, or online at `http://hydromad.catchment.org/`. A tutorial document is also available.

Other standard R methods are provided for convenience to work with `hydromad` objects: for example one can extract parameter values with `coef()`, and model results with `fitted()`, `observed()`, or `residuals()`. An estimate of the parameter variance-covariance matrix can be extracted with `vcov()` where applicable. Several plotting functions are also available; `xyplot()` is the basic function for displaying time series and scatter plots, while `qqmath()` shows empirical cumulative distributions (often called "flow duration curves" in hydrology). Most methods will work either on a single model object or on a `runlist`.

4.2. The models

Currently, the package currently includes implementations of several published SMA models:

- IHACRES Catchment Wetness Index (CWI) model (Jakeman and Hornberger, 1993) including the generalisation of Ye et al. (1997), modified according to Croke et al. (2005). A temperature-dependent drying rate is used to estimate a wetness index, which defines the runoff ratio. This model includes a scale factor which is estimated by mass balance with observed streamflow (or based on the gain of the transfer function used for routing, if applicable).
- IHACRES Catchment Moisture Deficit (CMD) model (Croke and Jakeman, 2004). Accounts for evapo-transpiration and changes in catchment storage. The version use here includes a power law form, and is described in Section 8.1;
- Sacramento Soil Moisture Accounting model (Burnash, 1995) developed by the US National Weather Service. With 13 parameters it is more complex than the other models listed here. Many published studies have used this model, often with good results. This implementation uses code from the University of Arizona.

- the GR4J model (Perrin et al., 2003), modèle du Génie Rural à 4 paramètres Journalier. In fact this is split up into a 1-parameter SMA model and a 3-parameter routing model. The non-linear routing component is based on a groundwater reservoir.
- the AWBM, Australian Water Balance Model (Boughton, 2004). In its simplest form this consists of a 1-parameter SMA model, although the full form has 6 parameters. It is traditionally used with a two-store (3 parameter) routing component.
- the single-bucket models of Bai et al. (2009), including interception, saturation excess runoff and subsurface flow.
- a degree-day factor snowmelt model of Kokkonen et al. (2006). A fraction of rainfall becomes snow, based on temperature thresholds, and a snow reservoir is estimated. Discharge from the snow model is, currently, fed into the CMD model listed above.

A set of simple benchmark models is also available; these are useful for some kinds of calibration methods, and for null models in a comparative analysis. Note that the `dbm` and `runoffratio` models make use of observed streamflow and so can not be used for general simulation. The benchmark models are:

- `scalar`: a constant runoff ratio (i.e. effective rainfall is a constant fraction of rainfall). The fraction is estimated for mass balance with streamflow, or based on the gain of the transfer function used for routing, if applicable.
- `intensity`: runoff ratio estimated by raising rainfall to a power, up to threshold rainfall rate with maximum runoff. With a power of 0 this reduces to the scalar model.
- `runoffratio`: a runoff ratio, estimated by a moving average through the data, is used to scale rainfall.
- `dbm`: observed streamflow raised to a power defines an index of antecedent wetness. This index, possibly lagged, is used to scale the rainfall. As a typical structure used in the early stages of Data-Based Mechanistic modelling (Young, 2003), it is termed `dbm` in **hydromad**.

Routing models currently include

- `armax`: ARMAX-type (auto-regressive, moving average, with exogenous inputs), also known as linear transfer functions (Jakeman et al., 1990; Ljung, 1999);
- `expuh`: exponential component configurations (up to 3 in parallel and/or series). The time constants of each are specified, as well as a choice of configuration. A loss term can be included to represent simple groundwater exchange, similar to the form of Herron and Croke (2009).
- `powuh`: a power law form of the unit hydrograph, parameterised according to Croke (2006).

- `varuh`: a variable partitioning extension of a 2-store model: this is an example of a routing model which is not a constant unit hydrograph, but rather depends on the level of rainfall.

Note that `armax` models up to third order can be converted into `expuh` form within **hydromad**. For these models, efficient estimation methods are available: notably the Simple Refined Instrumental Variable (SRIV) algorithm (Young, 2008), and an inverse filtering algorithm which estimates the parameters directly from streamflow data. In the case of `expuh`, if the solution does not make sense physically — having negative or imaginary recession rates — then these are re-fitted with constraints.

4.3. Optimisation algorithms

Several optimisation functions are available in the **hydromad** package for calibrating models to observed data. The different algorithms may each be preferred for different types of problems, and most have settings to tune their performance. There is generally a trade-off between rapid convergence to a moderately good result, versus a time-consuming search for the best possible solution. The choice in this regard will depend on the task at hand.

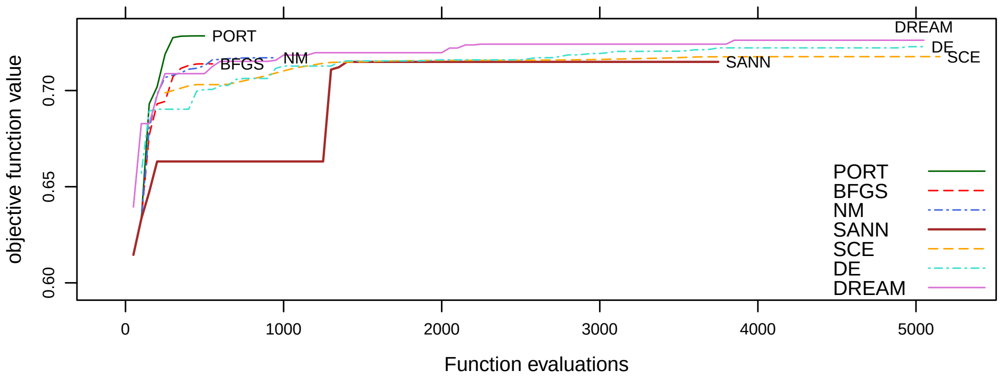

Figure 2: Optimisation traces from 7 algorithms in calibrating an 8-parameter model (described in the text) using an R^2 objective function. Dream was run using a likelihood function analogue to the objective function. Note that each algorithm was run using default settings only, and some of the results could probably be improved by adjusting these settings.

The performance of optimisation algorithms, defined further on, can be visualised with an optimisation trace plot, showing the improvement in the solution found with increasing effort. Effort in this sense means the number of times the model simulation function is run, which is generally proportional to the total running time. One example is given in Figure 2, which shows results from calibrating the IHACRES CMD model with a three-store unit hydrograph, involving 3 free parameters in the SMA component and a further 5 free parameters in the routing component. It was calibrated to a three-year period of streamflow in the Queanbeyan River starting in 1983. The objective function used was the R^2 using untransformed data (Nash and Sutcliffe, 1970). Note that these results are only for illustration of the general point, and that the algorithms were run only once with the somewhat arbitary default settings. In this case the `PORT` algorithm found a good result relatively quickly, but in other cases the evolutionary algorithms such as `DE`, `SCE` or `Dream` may find better solutions. It is often the case that if a fit is needed quickly, such as in a large simulation study or an interactive, exploratory analysis, then algorithms such as `PORT` or `Nelder-Mead` are most appropriate.

The current set of general optimisation functions is:

- `fitBySampling`: allows for Random, Latin Hypercube or regular gridded sampling. The model with best objective function from the sample is selected. Note that these sampling methods can be used for more general analysis with the `simulate()` function; an example is given in Section 8.
- `fitByOptim`: uses R's built-in `optim` and `nlminb` functions to optimise an objective function. The initial parameter values are chosen from a preliminary sampling run, or alternatively, a number of samples can be used as different starting points (multi-start mode). Some of the available methods are:
  - `"PORT"`, using functions from the Bell Labs PORT library[2];
  - `"Nelder-Mead"`, a simplex method;
  - `"BFGS"`, a quasi-Newton method;
  - `"SANN"`, Simulated Annealing, designed to find a reasonable global solution even in ill-conditioned, high-dimensional solution spaces.
- `fitBySCE`: the Shuffled Complex Evolution algorithm developed at the University of Arizona (Duan et al., 1992);
- `fitByDE` Differential Evolution (Price et al., 2005), provided by the **DEoptim** package; and
- `fitByDream` DiffeRential Evolution Adaptive Metropolis (Vrugt et al., 2009). This is a Markov-Chain Monte Carlo method, giving probabilistic results. However, it can also be used simply as a optimisation algorithm.

In addition, some specialised calibration functions are available for specific models. It is straightforward, too, to make use of any other general optimisation function to calibrate `hydromad` model objects.

The important issues of event-based analysis and design of objective functions are discussed in the following two sections.

5. Discrete event separation

We are often interested in hydrological response properties, and modelling these, at the event scale rather than at the level of the raw data. Furthermore, model residuals are

[2] http://www.bell-labs.com/project/PORT/

typically highly autocorrelated, which is problematic when attempting to assess model performance. An attractive approach is to separate the streamflow record into relatively isolated events, and work with attributes of events rather than time steps. Events are most often used in the literature for extreme value analysis (Katz et al., 2002) or for ephemeral flow systems in arid environments (McIntyre and Al-Qurashi, 2009), but can also be applied more generally. For instance Willems (2009) uses event windows to extract local peaks and troughs of a streamflow series. Boyle et al. (2000) separate streamflow series into events, but do model assessment using the raw time-step data rather than event-scale properties.

One might be concerned that, in reducing a series with thousands of observations to perhaps a hundred events, we are throwing away most of the information; but we argue that the latter is often a more reasonable representation of the actual information content of the data for the purposes at hand.

The **hydromad** package provides functions for identifying events based on thresholds and various timing criteria, and applying aggregation functions to events. Events can be characterised by attributes such as total flow or rainfall, antecedent conditions, temperature or season.

Temporal aggregation is more typically undertaken in terms of regular time steps, such as days or months. This is certainly easier, but is arguably less meaningful in hydrological terms, and is susceptible to edge effects. Event-based aggregation requires consideration of how events are to be defined for the purposes at hand. Also, as events may have widely differing durations, in some cases it may be appropriate to weight events by their duration for fitting or assessment.

Events may be defined either from a rainfall series or from a streamflow series (or both). Rainfall-based events are more suitable for assessing SMA model performance, because even when there is no streamflow response to rainfall, this is an important feature of the model. Streamflow-based events are more suitable for investigating streamflow characteristics such as unit hydrograph estimation or flood frequency analysis.

There are several other considerations too: are only the high periods above a threshold of interest, or should such a high period instead initiate an event window which continues until the next such event; or should the high and low periods around a threshold both be considered? Should events be terminated by flow (or rainfall) falling below a threshold, or falling below a threshold for a certain time, or must they be separated from other events by a minimum time? Should events of too short a duration be skipped, or extended? To help one decide these issues, the **hydromad** package provides a graphical user interface for interactively testing different event definitions on rainfall and streamflow time series.

Several event definitions are used in some of the pre-defined statistics listed in Appendix A, and throughout this paper. They are also illustrated in Figure 3.

- `e.rain5`: events are defined by rainfall exceeding 5 mm per day, and continuing until the next such event. Each single event continues at least until rainfall remains below 1 mm for 4 days.

- `e.q90`: events are defined by observed flow exceeding the 90 percentile level for at least 2 time steps, and continuing until the next such event. Each single event continues at least until flow falls below the 90 percentile level for 4 time steps, and must be separated from the next event by a further 5 time steps.

- `e.q90.all`: events are defined by observed flow exceeding the 90 percentile level for at least 2 time steps, but unlike `e.q90`, are not continuing until the next event; rather, these high flow periods and the low flow periods between them are both considered as events. The high events continue until the flow falls below the 90 percentile level for 4 time steps and must be separated from each other by a further 5 time steps.

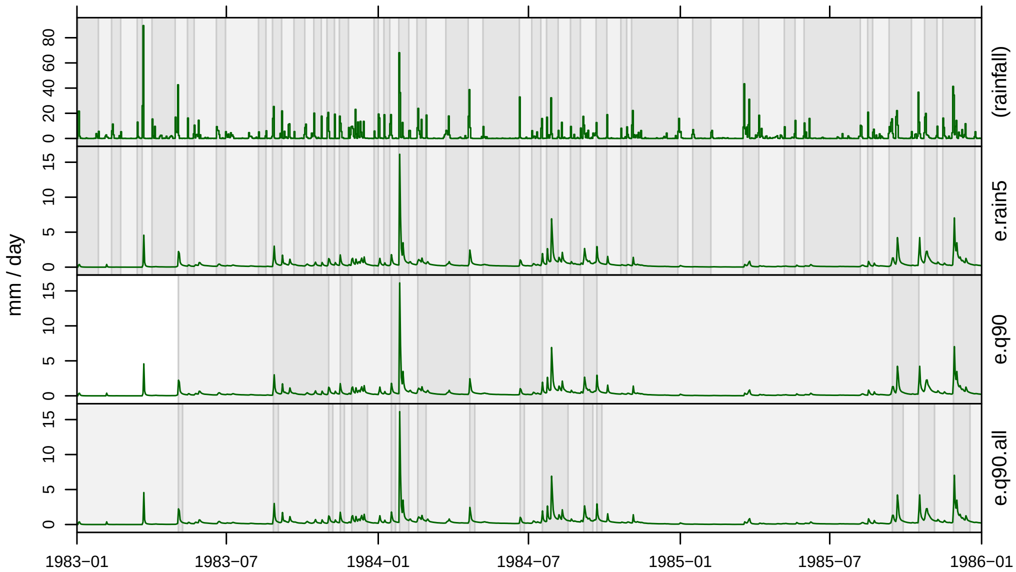

Figure 3: Section of the Queanbeyan River dataset, showing areal rainfall (P) and streamflow (Q). The event windows corresponding to the `e.rain5`, `e.q90` and `e.q90.all` statistics are shown. The alternating shading marks the event windows; there is no difference between the light and dark shaded events.

Finally, the choice of aggregation function is crucial: it is typically the sum or mean, but could also be a quantile or a set of quantiles. Figure 4 shows that taking flow sums in the `e.rain5` event windows does eliminate the auto-correlation evident in the raw time series.

6. Statistics and Objective Functions

At least as important as the choice of calibration algorithm is the design of an objective function. This requires careful consideration of the intended purpose of the model, how it will be applied, what derived values will be ultimately used, and the level of temporal precision required.

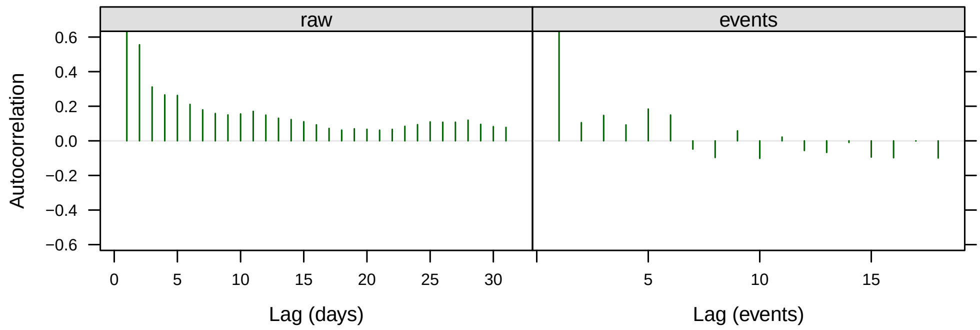

Figure 4: Demonstration of event aggregation of the streamflow data from Queanbeyan River, period 1983-01-01 to 1986-01-01. Events were defined from the rainfall series using the `e.rain5` definition. Note, specifically, that the first-order autocorrelation evident in the raw data is reduced in the event series.

Streamflow data tends to be highly skewed, and this leads implicitly to a large weighting put on a few large observations, a weighting which is often inappropriate for the purposes of the model and the uncertainty in those particular values.

Croke (2009) describes an approach to incorporating uncertainty of individual data values into the formulation of an objective function. This is useful where detailed information on data errors are available. Otherwise, weighting can be managed by a simple transformation of the data: for instance, if we assume that errors in the data are multiplicative, an inverse-variance weighting would correspond to a log transformation. To reduce skewness further, a Box-Cox transform (Box and Cox, 1964) can be chosen such that the observed data approximates a Normal distribution. This is demonstrated in Figure 5.

In terms of the typical assessment of goodness of fit of a simulated time series to the observed time series, there are three main facets to the design of a statistic:

1. temporal precision / aggregation: either regular time steps may be specified, or hydrological events may be separated in some way. The choice of aggregation function is crucial: it is typically the sum or mean, but could also be a quantile or a set of quantiles.
2. data transformation: to adjust the implicit weighting put on different flow levels, and the corresponding sensitivity of fit statistics.
3. reference model: a null model to use as a reference point to compare model performance (after Nash and Sutcliffe, 1970). This is typically as simple as a constant at the mean observed level, but more informative choices are possible. The choice of reference model does not change the ranking of models, but does change the scale of the statistic.

The **hydromad** package allows statistics and objective functions to be specified as arbitrary R functions. A helper function is available to streamline the process of defining a statistic based on the above considerations: `buildTsObjective()` takes optional

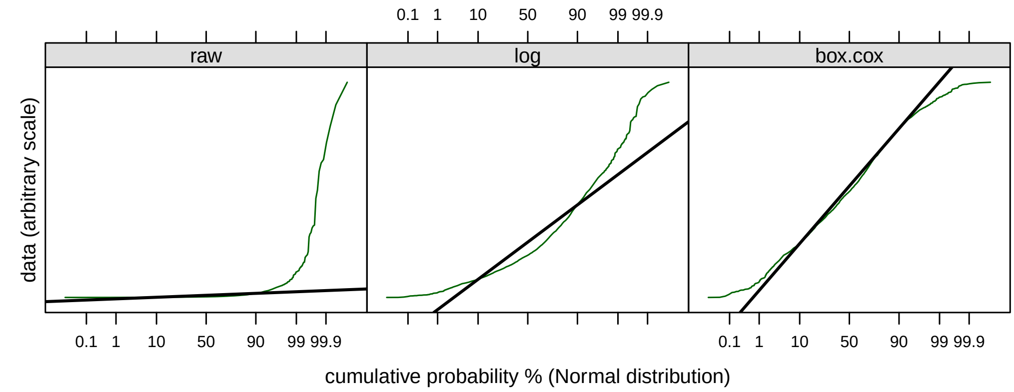

Figure 5: Demonstration of a Box-Cox transformation and log transform of the streamflow data from Queanbeyan River, period 1970-01-01 to 1980-01-01. A reference line is shown through the 10th and 90th percentiles on the normal probability scale, showing deviations from normality.

arguments for grouping, an aggregation function, a data transformation in the Box-Cox family of transformations, and a reference model.

The statistics defined by `buildTsObjective()`, and in most of the pre-defined statistics, are based on a general form of a fit statistic, termed `nseStat()`. It is based on the familiar r^2, a more general form of the Nash-Sutcliffe Efficiency (Nash and Sutcliffe, 1970), defined as

$$\text{nseStat}(Q, X) = r^2 = 1 - \frac{\sum |Q_* - X_*|^2}{\sum |Q_* - Z_*|^2} \tag{1}$$

where Q and X are the observed and modelled values respectively, possibly in aggregated form; Z is the result from a reference model, which is the baseline for comparison. It defaults to the mean of observed data after transformation $\mathrm{E}(Q_*)$, corresponding to the typical r^2 statistic. Subscript $*$ denotes an arbitrary transformation.

The set of pre-defined statistics is listed in Appendix A. These statistics can be used, combined and adapted, either as objective functions or for model evaluation (discussed later).

7. Data analysis methods

As the models considered here are data-driven, there is a core role for data analysis in model development and assessment. Exploratory data analysis is inherently open-ended and should adapt to the unique problems at hand. As a set of possible starting points for analysis, these type of methods are often useful:

- interactive inspection of time series data. The importance of "eyeballing" the data in detail is sometimes forgotten.

- cross-correlation: reveals strength of the linear relationship between pairs of time series, and can be applied over time to identify non-stationarity of the relationship.
- trend estimation by smoothing, where trend can be considered generally as any systematic relationship with covariates, including (but not limited to) time.
- where spatially distributed data is available, such as records from multiple rain gauges, it is often worth checking for spatial effects.

Rainfall patterns are complex, with dynamics from time scales of hours up to decades, and, importantly, with semi-regular seasonal effects. One can easily generate a seasonal and trend decomposition of the rainfall series using the STL algorithm (Cleveland et al., 1990), shown in Figure 6. This reveals a semi-regular seasonal pattern and a long-term non-seasonal climatic pattern: the period from 1992 onwards appears to be much less variable across years. Unexplained, short-term variation remains and the wet period of 1991 for example appears as inconsistent with the general seasonal pattern.

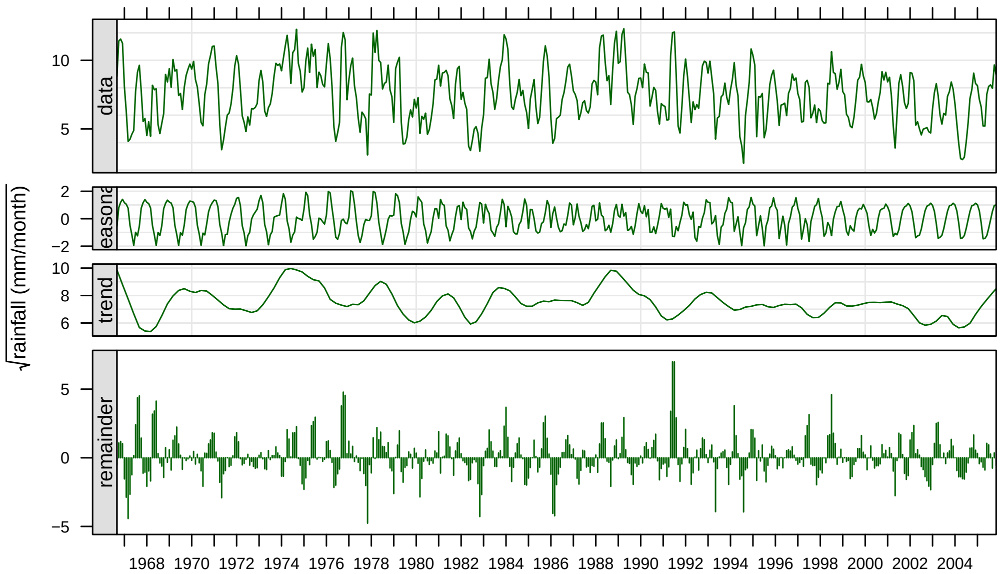

Figure 6: Seasonal and Trend decomposition by Loess (STL) applied to the monthly areal rainfall series (with a moving average over 3 months) for Tinderry catchment. The Loess window for seasonal extraction was set to 10 years.

Of course, many hydrological problems are concerned with the streamflow response to rainfall which occurs at shorter time scales. A good first step in characterising this response is to examine the auto-correlation and cross-correlation functions. These represent average responses. It is also possible to look at distributions of responses in terms of discrete events.

7.1. Regression analysis

Rainfall-runoff dynamics were investigated in the Queanbeyan river dataset. Events were defined by rainfall exceeding 5 mm per day, and continuing until the next such event: the `e.rain5` definition from Section 5. To estimate the effective rainfall, rises in streamflow were extracted, and scaled for mass balance over the whole period of record. This simple approach assumes that the unit hydrograph is constant. A more sophisticated analysis could use a time-varying parameter model (e.g. Norton and Chanat, 2005). Once effective rainfall is estimated, it is straightfoward to calculate the runoff ratio corresponding to each event, and relate this to other variables. The obvious drivers are rainfall amount and antecedent wetness. Streamflow itself can be used as an index of wetness, assuming a direct storage-discharge relationship (Young, 2003). Temperature or season may also capture some residual effects related to dryness or rainfall intensity.

Generalised Additive Models (GAMs) are suitable for this type of analysis. Rather than assuming a functional form, it is possible to allow the data to define the relationships: using splines as a basis, the degree of smoothness is chosen by generalised cross-validation. This is implemented in the R package **mgcv** (Wood, 2004).

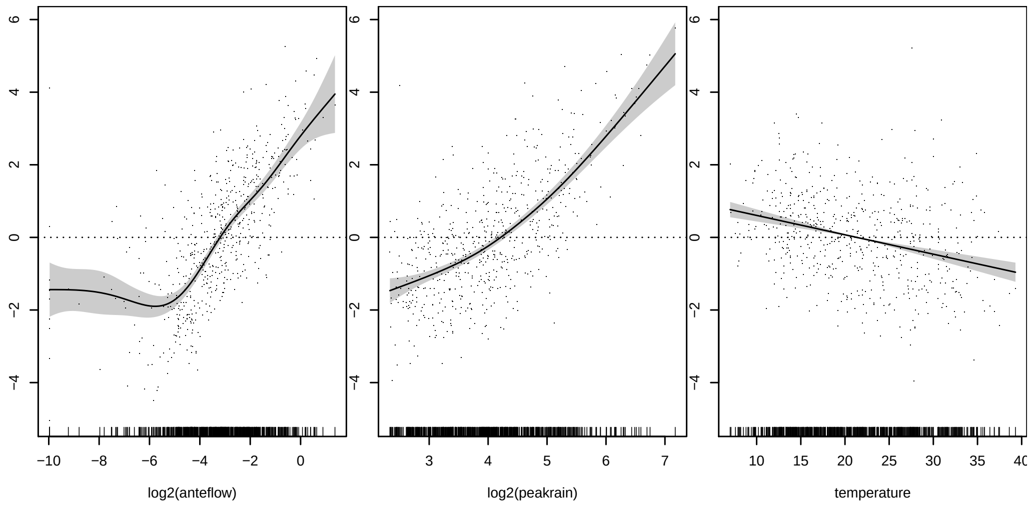

Figure 7: rainfall-runoff proportion, estimated in each event window, related to three potentially relevant variables (antecedent streamflow as an index of wetness, peak rainfall, and antecedent temperature) by fitting a GAM. The panels show additive effect sizes in terms of deviations from the average runoff ratio, in log units. Standard errors (shaded) and partial residuals (points) are also shown.

Figure 7 shows the estimated effects of the three variables mentioned above. The model was formulated as

```
> gam(log2(runoff) ~ s(log2(anteflow)) + s(log2(peakrain)) +
    s(temperature))
```

where `s()` is the smoothing operator. This implicitly includes an intercept term; therefore the estimated terms represent effects as deviations from the average runoff ratio (on a log scale). The strongest relationship is with antecedent wetness, and this appears to be a power law (linear on a log-log scale). It is also obvious that rainfall intensity increases the runoff ratio for rainfall rates above about 20 mm/day. An additional effect of temperature can be seen, and is significant given the standard error bounds shown.

This type of analysis can be used to inform conceptual model development by revealing the form of the effects that need to be accounted for. We will revisit these effects when assessing model performance in Section 9: by assessing the effectiveness of a model in accounting for each effect we can see which of the process representations needs to be improved. A more sophisticated analysis might also estimate interactions between these variables.

7.2. Delay times

Dynamic hydrological models are typically calibrated with an implicit emphasis on streamflow peaks, and are therefore particularly sensitive to timing mismatches between rainfall and streamflow. Most models can only handle a constant delay time. However, it is reasonable to expect that the delay between rainfall and runoff is variable (e.g. due to rainfall intensity, and the timing of rainfall within each time step). Timing mismatches may also in some cases be due to errors in the data.

In some problem contexts, such as estimation of sediment load, it is important to account for high flows but not their timing. In this case all flow peaks could be shifted in line with corresponding rainfall peaks, to match the typical model assumptions.

Using the event-based approach, we can analyse delay times directly. In each event window, the delay time is estimated from the cross correlation function of rainfall with streamflow rises. Event windows with no streamflow response (or a response of less than 0.1 mm/day) were ignored.

	-5	-4	-3	-2	-1	0	1	2	3	4	5	Sum
lowest third	0	0	0	1	0	1	46	45	3	2	0	100
middle third	0	1	0	0	0	4	66	28	0	0	0	100
highest third	0	0	0	0	0	5	85	8	0	0	0	100

Table 2: Distribution of lag times following identified rainfall events (above 5 mm/day), conditioned on the magnitude of the resulting streamflow peak. Each range of magnitudes covers an equal number of events (197), and values shown are percentages in each range. The lowest third of events have peaks below 24 ML/day, and the highest third have peaks above 184 ML/day.

The estimated delay times are shown in Table 2. The table is divided according to the magnitude of peak streamflow: it is clear that small runoff events have a longer and more variable delay time (with a modal time of 2 days), while large runoff events have a more consistent delay time of 1 day. This makes sense as large runoff events would take a more direct flow path. Whether it is worth trying to model this effect would depend on the problem context. Also, some of the outlying delay times may be worth investigating as possible errors.

Having satisfied some basic checks, we can move on to analyse the dynamics in more detail using conceptual models.

8. Model development and calibration

This section introduces simple conceptual modelling of the catchment system. A 3-year calibration period was chosen to focus the model development: 1983-01-01 to 1986-01-01.

8.1. The CMD model

For a Soil Moisture Accounting model, we start with the IHACRES Catchment Moisture Deficit (CMD) model of (Croke and Jakeman, 2004). This accounts for evapo-transpiration and changes in catchment storage in a mass balance equation:

$$M[t] = M[t-1] - P[t] + E_T[t] + U[t] \tag{2}$$

where M represents catchment moisture deficit (CMD) in mm, constrained below by 0 (the nominal fully saturated level). P is catchment areal rainfall, E_T is evapo-transpiration, and U is drainage (effective rainfall), in units of mm per day.

Rainfall effectiveness (i.e. drainage proportion) is a simple *instantaneous* function of the CMD level, dropping to zero at $M = d$. In the default linear form of the model, the rainfall effectiveness $\mathrm{d}U/\mathrm{d}P$ decreases linearly from the full runoff level at $M = 0$. In the power law form it is:

$$\frac{\mathrm{d}U}{\mathrm{d}P} = 1 - \min\left[1, \left(\frac{M}{d}\right)^{\eta}\right] \tag{3}$$

where η is the `shape` parameter. The actual drainage each time step involves the integral of Equation 3:

$$M_f = \begin{cases} d\left(-(1-\eta)\frac{P_t-(M_{t-1}-d)}{d} + 1\right)^{\frac{1}{1-\eta}} & \text{if } M_{t-1} \le d \\ d\left(-(1-\eta)\frac{P_t}{d} + (\frac{M_{t-1}}{d})^{1-\eta}\right)^{\frac{1}{1-\eta}} & \text{if } d < M_{t-1} \le d + P_t \\ M_{t-1} - P_t & \text{if } M_{t-1} > d + P_t \end{cases} \tag{4}$$

Similar solutions for the linear and trigonometric forms are given in Croke and Jakeman (2004).

Evapo-transpiration, as a proportion of the potential rate `E[t]`, is also a simple function of the CMD level, with a threshold at $M = fd$:

$$E_T[t] = eE[t]\min\left[1, \exp\left(2\left(1 - \frac{M_f}{fd}\right)\right)\right] \tag{5}$$

where M_f is the CMD level M after precipitation and drainage have been accounted for.

Following Croke and Jakeman (2004), we fix the evapotranspiration coefficient e at a reasonable value for use with daily maximum temperature data: 0.166. The flow threshold d has been found not to be a very sensitive parameter, and was fixed at a value of 200 mm. This leaves the stress threshold f and shape parameters to be calibrated. Reasonable ranges were selected for these, shown in Table 3.

`f`	stress threshold / wilting point (as fraction of `d`). Range $[0.01, 1]$
`shape`	power in drainage equation. Value less than 1 select the linear form; a value of 1 selects the trigonometric form. Range $[0, 100]$
`d`	flow threshold (mm). Fixed at 200.
`e`	evapotranspiration coefficient. Fixed at 0.166.

Table 3: Parameters of the CMD Soil Moisture Accounting model, and values used in this study.

8.2. Routing component

For the routing component we use an exponential form of the unit hydrograph with two stores in parallel.[3] This form has been found to perform well in a variety of studies (e.g Jakeman et al., 1990). Each component is defined by a time constant τ and fractional volume v, or equivalently a recession rate α and peak response β. They are related as $\alpha = \exp(-1/\tau)$ and $\beta = v(1 - \alpha)$ for each component. When there are two components in parallel, these are conventionally called slow (s) and quick (q) flow components. The total simulated flow X at each time step t is the sum of the two:

$$\begin{aligned} X_s[t] &= \alpha_s X_s[t-1] + \beta_s U[t] \\ X_q[t] &= \alpha_q X_q[t-1] + \beta_q U[t] \\ X[t] &= X_s[t] + X_q[t] \end{aligned} \qquad (6)$$

Only one of the two fractional volume parameters needs to be specified; the other is the remainder, ensuring that the total volume is unchanged. Therefore this routing component has 3 free parameters, and the full model has 5 free parameters.

8.3. Simulation over parameter spaces

The widespread observation of equifinality (Beven, 2006), whereby parameter values can not be uniquely identified on the basis of the available data, should make us cautious when calibrating models. It can be useful to visualise the *objective function surface in parameter space* corresponding to the calibration problem. This is a non-trivial problem and there are various possible approaches (Wagener and Kollat, 2007). Apart from the computational demands of random sampling, there are significant cognitive demands in attempting to visualise multi-dimensional functions.

Of particular interest is comparing different objective functions over the parameter space. We generated 2000 stratified random samples over pre-defined parameter ranges, and calculated fit statistics from each simulation: `r.squared` and `r.sq.log`, defined in Appendix A. For each statistic, the best 3 simulations were identified, and also what might be termed a *90% coverage set*, following Blasone et al. (2008): the smallest set of the best-performing parameter values, such that 90% of the observed streamflow levels fall within the range of the ensemble simulation.

[3] This is using the `expuh` routing function, which currently handles up to three stores in parallel and/or series.

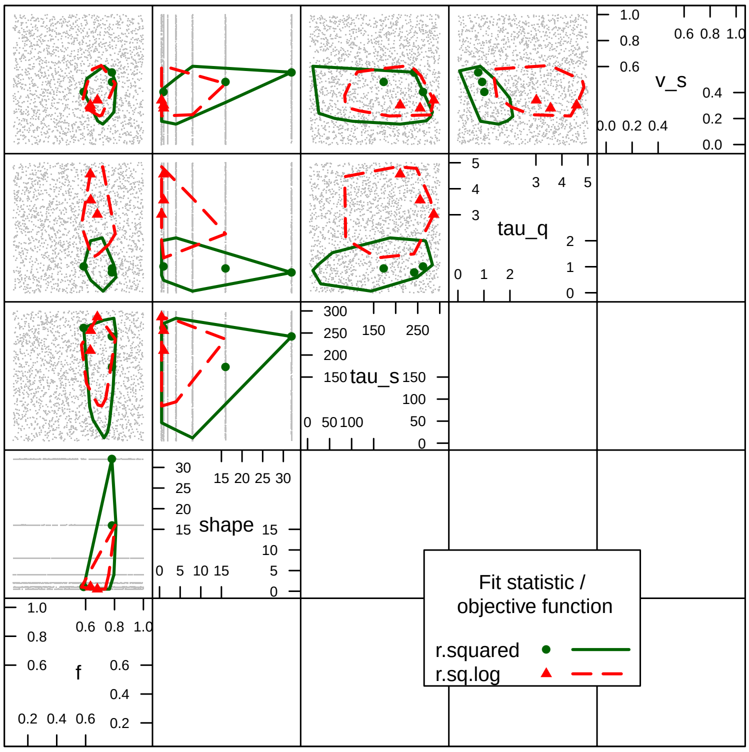

Figure 8: Optimum points and regions of parameter space, projected onto each pair of parameters. Symbols show the best 3 simulations according to each statistic, and lines show convex hulls around the corresponding *90% coverage sets*. Small points represent all 2000 simulations. These simulations used a CMD model with 2-store routing applied to the period 1983-01-01 to 1986-01-01 of the Queanbeyan River data.

The *90% coverage sets* and optimum values are compared between the two objective functions for each pair of parameters in Figure 8. The results show a reasonable level of agreement between the two statistics, as one would hope to be the case. The parameters `shape` and `tau_s` can take on values over their full prior range, although they do constrain the parameter space through their interactions (e.g. high `shape` and low `tau_s` is unacceptable according to the `r.sq.log` statistic). There are differences between the objective functions too: `r.sq.log` statistic favours longer time constants and a higher fractional volume of slow-flow (v_s). Regarding the SMA component, there appears to be an interaction between the two parameters (f and shape), and an ambiguity in where the optimum lies. While most of the optima are around shape $= 0, f = 0.7$, which is a linear form of the drainage equation, one result favoured by `r.squared` is shape $= 32, f = 0.8$, which is a power law form. In fact, with such a high power the drainage equation is almost a step function. This ambiguity is explored in the following sections.

Random sampling is extremely inefficient for exploring high probability density regions of the parameter space. For models with more parameters, a random sampling approach will not be able to find optimum parameter sets in a reasonable amount of time. Adaptive sampling schemes, such as Markov Chain Monte Carlo (MCMC) methods, have been shown to be essential in such contexts (e.g. Blasone et al., 2008). As such, the **hydromad** package has a function to estimate feasible parameter sets from the output of the DREAM MCMC algorithm, as well as from purely random sampling.

8.4. Calibration

In order to assess the model performance in detail it is useful to identify two or three models which capture aspects of our uncertainty about the parameter values and representation of processes.

The CMD model structure as described above was fitted to the calibration period data using two contrasting objective functions: the typical `r.squared` using raw data (Nash and Sutcliffe, 1970); and the same with log transformed data (`r.sq.log`). A log transform would be appropriate when assuming multiplicative data errors, and naturally gives less weight to peak flows than with raw data. As above, the observed 10 percentile flow was used as an offset. In this case the `PORT` method of `fitByOptim` was used for calibration. The two resulting parameter sets capture the ambiguity in the SMA parameters f and shape noted in the previous section, as can be seen from the values given in Table 4.

One more model was selected to act as a reference. The *intensity* SMA model estimates runoff from rainfall on the corresponding time step only, neglecting any consideration of antecedent conditions. In terms of the effects shown in Figure 7, only the peak rainfall effect in the middle panel is included. This provides a useful reference, because any improvement over this can be largely attributed to the dynamic process representation of the CMD model. The *intensity* model is defined as:

$$\frac{U}{P} = c \min\left(1, \left(\frac{P}{P_{\max}}\right)^{\gamma}\right) \tag{7}$$

This model, together with the same two-store routing component used with the CMD model, were calibrated to the `r.squared` objective function. Calibrated parameter values were $\gamma = 0.8$, $P_{\max} = 160$, and $c = 1$. The unit hydrograph routing parameters are given in Table 4.

	f	shape	tau_s	tau_q	v_s	delay
cmd_raw	0.79	32	300	0.88	0.58	1
cmd_log	0.70	1	203	2.99	0.39	1
intensity			12	0.86	0.61	1

Table 4: Calibrated parameters for models calibrated to the period 1983-01-01 to 1986-01-01 for the Queanbeyan River catchment.

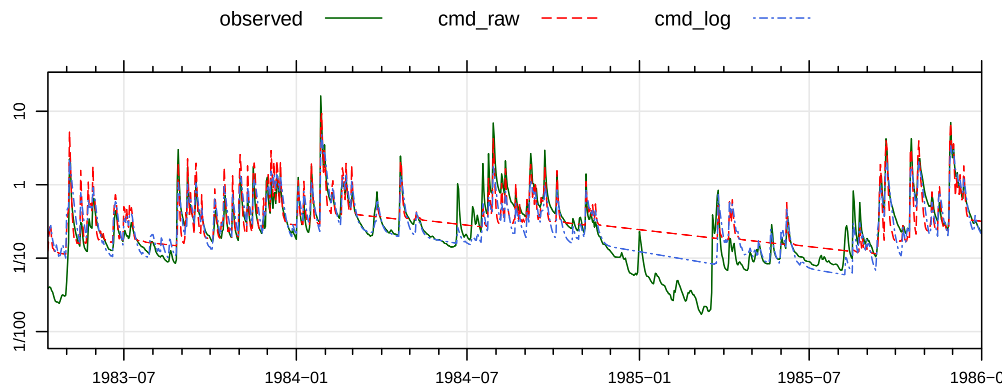

Figure 9: Fitted streamflow time series from IHACRES CMD models, compared to the observed time series.

A sample of the simulated time series from the two calibrated CMD models, compared to the observed streamflow time series, is shown in Figure 9. As the plot is on a log scale, it highlights the low flow dynamics; other aspects of the model fit are considered in Section 9. The most obvious feature of the plot is that the `r.squared` calibration is over-estimating most of the low flows and lacking any response to the smaller peaks. This reflects the skewness in the raw data and the consequent overpowering influence of peak values on the objective function. Somewhat surprisingly, many of the large peaks in the period shown are also over-estimated by the `r.squared` case, though this is not generally the case, as we will see. Both models fail to reproduce the switching off of baseflow in 2003 and 2004, as there is no such mechanism in the model formulation.

The next section will investigate the performance of the three candidate models in detail, with a view to further model development.

9. Model performance assessment

The typical approach to model performance assessment is to calculate fit statistics. Fit statistics for the calibration period are given in Table 5a. The same fit statistics were also calculated over two verification periods — the relatively wet decade of the 1970s and the relatively dry decade of the 1990s — and are listed in Tables 5b and 5c. The statistics are defined in Appendix A.

	r.squared	r.sq.log	e.rain5	e.rain5.log	e.q90	e.q90.all
cmd_raw	**0.72**	0.62	**0.83**	0.75	0.66	0.74
cmd_log	0.54	**0.77**	0.81	**0.82**	**0.70**	**0.81**
intensity	0.67	0.39	0.65	0.58	0.64	0.75

(a) Calibration period 1983–1986

	r.squared	r.sq.log	e.rain5	e.rain5.log	e.q90	e.q90.all
cmd_raw	**0.60**	0.66	**0.61**	0.79	**0.52**	0.55
cmd_log	0.40	**0.83**	0.60	**0.88**	0.50	**0.58**
intensity	0.59	0.31	0.59	0.54	0.33	0.51

(b) Wet period 1970–1979

	r.squared	r.sq.log	e.rain5	e.rain5.log	e.q90	e.q90.all
cmd_raw	**0.73**	0.53	0.79	0.71	0.75	0.76
cmd_log	0.54	**0.75**	**0.80**	**0.83**	**0.79**	**0.80**
intensity	0.60	0.15	0.68	0.34	0.52	0.65

(c) Dry period 1990–1999

Table 5: Fit statistics for three candidate models calibrated to the Queanbeyan River data. The best-performing model according to each statistic is highlighted in each assessment of three periods.

The fit statistics show that, while the `r.squared` model does best in terms of its own objective function, its improvement over the *intensity* reference model appears to be modest in the wet period, but more pronounced in the dry period. In contrast, the `r.sq.log` model appears to do much better than the others in terms of its objective function, and also in terms of all the event-based statistics, even those without a log transformation of the data.

It is easier to see the pattern of the fit statistics when it is displayed graphically, and with finer time resolution. Figure 10 shows the `r.squared` and `r.sq.log` statistics in each calendar year for the three models. An interesting feature that can be seen is that the *cmd_log* model (i.e. that calibrated to `r.sq.log`) actually performs better than the *cmd_raw* model in terms of `r.squared` in many years: specifically in dry years such as 1993, 1994 and 1996. In wet years the *cmd_log* model does less well, whereas the *intensity* model is often one of the best performers on this measure. The story is different on the measure of `r.sq.log`, where the performance is dominated by the corresponding *cmd_log* model.

The fit statistics we have considered so far have been based on comparing observed and simulated values at a daily time step. It often of interest also to consider the model performance at a longer time scale, to reveal the size and direction of any systematic biases. Figure 11 shows this in a plot of model residuals smoothed over a time window of around 1 year. Both raw and log-scale residuals are considered, by analogy with Figure 10. Indeed, the story is similar at this scale: the *cmd_raw* model does best on

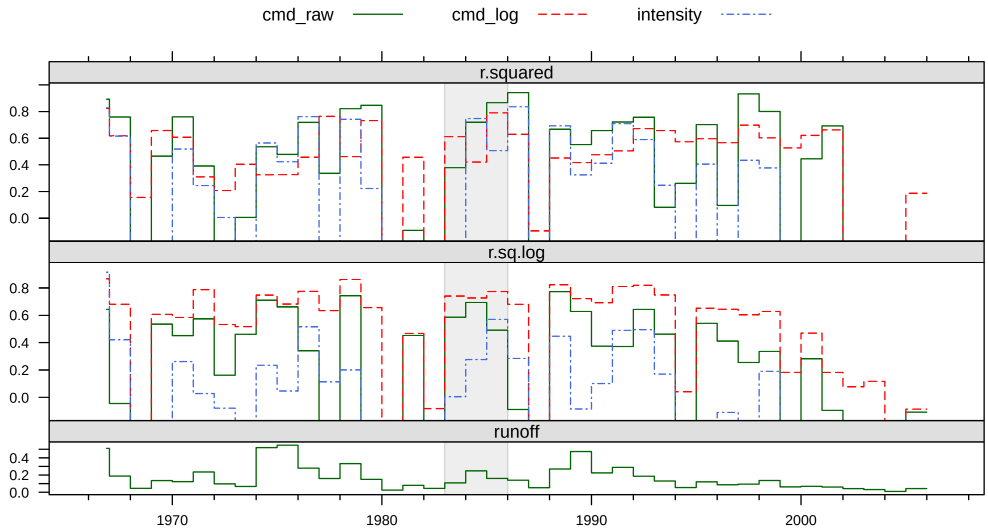

Figure 10: Fit statistics calculated in simulation over each calendar year for three candidate models calibrated to the Queanbeyan River data. The calibration period is shaded.

the raw scale in wet years, but does less well than *cmd_log* in dry years. All models have drastically underestimated the exteme response in years 1974–1975.

This kind of model performance assessment is quite detailed, and is able to show which models are preferred, over time and from different perspectives. However, it is ultimately unsatisfying because it does not provide much guidance in the task of diagnosing model structural problems and developing an improved model. For this, we can invoke an event-based analysis and consider the model residuals in relation to specific features of the data.

Building on the empirical data analysis of Figure 7, we can assess model residuals, aggregated up to total flow in event windows, in relation to the same three covariates: antecedent flow level, peak rainfall, and temperature. If the model were explaining the data well, then there should not be a systematic relationship with these covariates. It is useful to include additional covariates, to provide more information about each event: an index of antecedent rainfall (moving average over preceding 30 days), the peak *effective* rainfall as simulated by each model, and the mean observed flow. Figures 12 and 13 show the estimated relationships between model residuals and these covariates derived from simulation over the whole Queanbeyan River time series. Residuals were calculated as the difference between total observed and total simulated flow in all `rain5` event windows (defined in Section 5).

The results show that the *cmd_raw* model does best in accounting for antecedent conditions on the raw data scale, and even on the log scale under wet conditions but not dry conditions. Both CMD models provide a significant improvement over the *in-*

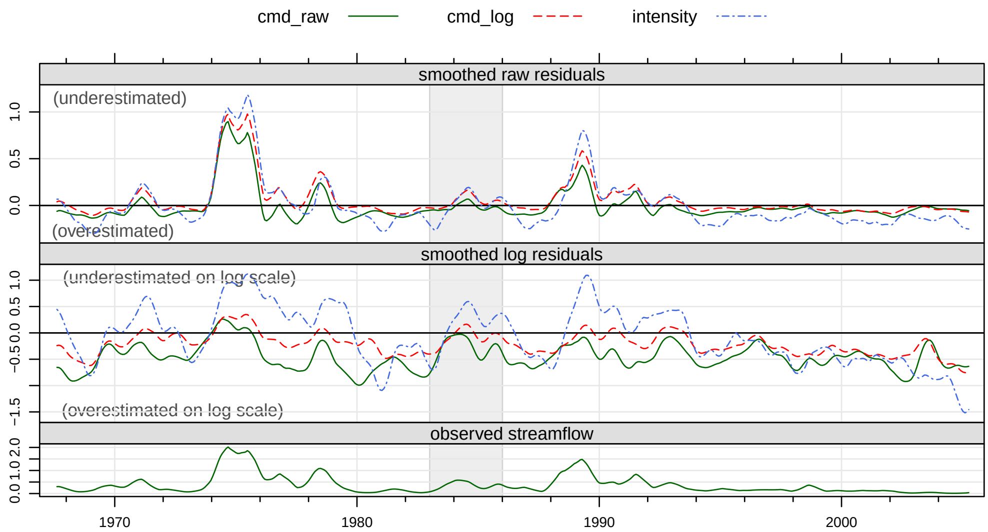

Figure 11: Time series of model residuals, both raw and log-transformed (with an offset). The daily residuals are smoothed over an effective bandwidth of 1 year with a triangular kernel. This reveals longer-term biases while hiding any short-term errors. The calibration period is shaded.

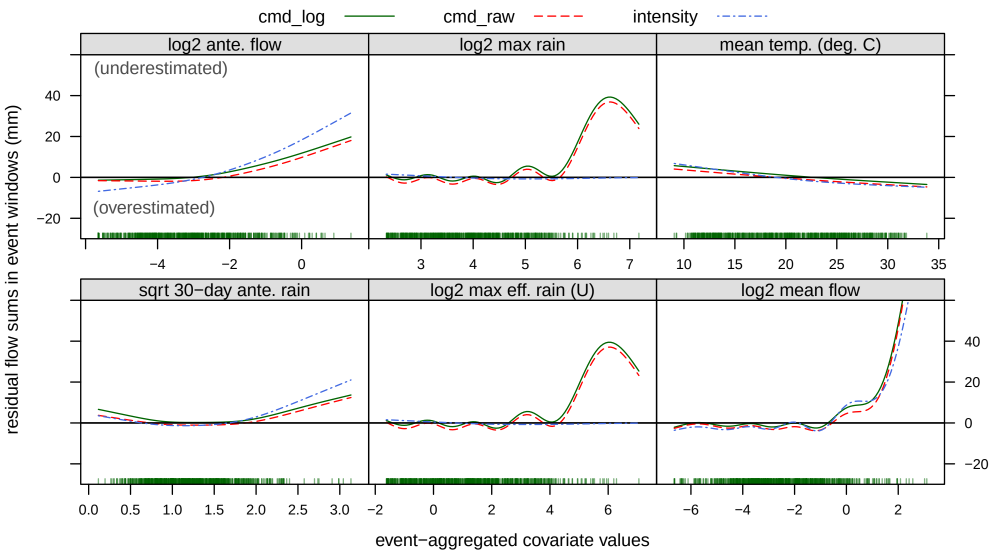

Figure 12: Marginal effects of variables on model residual flow sums in `rain5` event windows. These are GAM smooths fitted independently to the residuals against each covariate. All covariates are in units of mm/day (before transformation) except temperature which is in degrees Celcius.

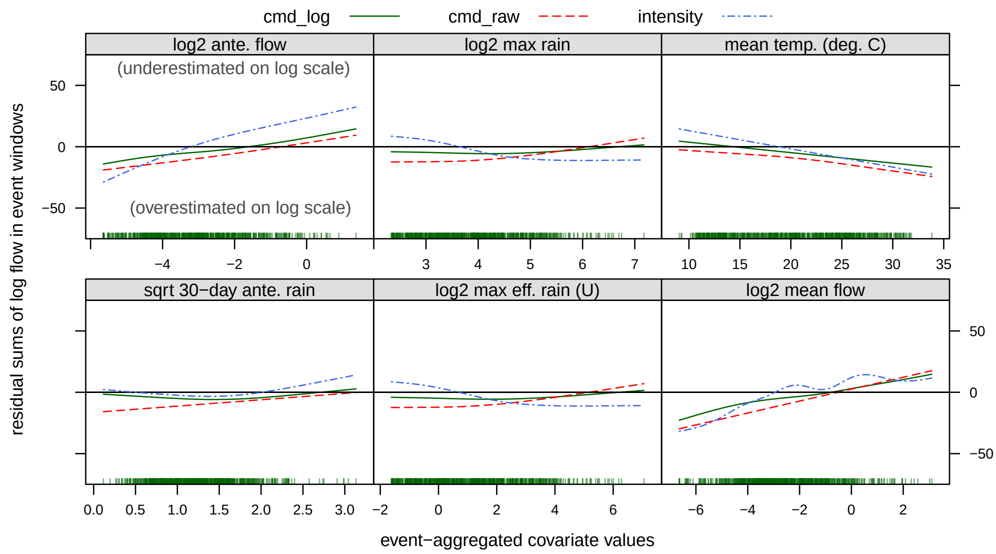

Figure 13: Marginal effects of variables on model log-scale residual flow sums in `rain5` event windows. These are GAM smooths fitted independently to the residuals against each covariate. All covariates are in units of mm/day (before transformation) except temperature which is in degrees Celcius.

tensity reference model, which does not account explicitly for antecedent conditions. On the other hand, as might be expected, the *intensity* model does best in accounting for high intensity rainfall (and effective rainfall), although it is still subject to an obvious systematic effect. The *cmd_log* model is clearly accounting for most effects better than the other models when log-scale residuals are considered. It is still subject to systematic effects with respect to antecedent flow, mean temperature, and mean observed flow; all of these are likely to be correlated.

By examining and comparing systematic errors in the model residuals we can think about how to develop improved models. For instance, there is room for improvement in accounting for antecedent wetness. Also, we have shown that runoff from high rainfall intensity events is being underestimated, and that even a simple intensity model can lead to some improvement in that respect.

10. Conclusions and outlook

The **hydromad** package attempts to offer the modeller a simple, flexible and open software environment, integrated with the R system, for working with hydrological models. For rapid use it includes a set of predefined models, calibration algorithms, and fit statistics. We have shown how an event-based data analysis approach can diagnose model deficiencies and guide model development. It is useful to take a comparative approach in this regard, comparing model performance with one or more reference models.

The package has some support for uncertainty estimation based on parameter sampling. Markov Chain Monte Carlo methods may be used for efficient adaptive sampling, with either a formal or an informal likelihood function. The resulting ensemble of parameter sets may be simulated on new data to produce point-wise confidence bounds. The potential exists to implement the more efficient bootstrap-based methods of Selle and Hannah (2010).

As **hydromad** and R are free software, it is hoped that further methodological innovation and associated software development will come from the wider community.

Acknowledgements

Hak-Soo Kim discovered the interestingness of the Queanbeyan River catchment, and generously provided the estimated areal rainfall data.

Computational details

The results in this paper were obtained using R 2.13.0 with the packages **hydromad** 0.9–8, **zoo** 1.6–5, (Zeileis and Grothendieck, 2005), **mgcv** 1.7–6 (Wood, 2004), **lattice** 0.19–26 (Sarkar, 2010), and **latticeExtra** 0.6–17 (Sarkar and Andrews, 2010). R itself and all packages used are available from CRAN at `http://CRAN.R-project.org/`.

The code used to produce the results and figures in this paper is available from `http://www.nfrac.org/felix/papers/`.

References

Andrews, F. T., Croke, B. F. W., Jeanes, K. W., July 2010. Robust estimation of the total unit hydrograph. In: Swayne, D., Et (Eds.), Proceedings of the iEMSs Fifth Biennial Meeting: "Summit on Environmental Modelling and Software". International Environmental Modelling and Software Society, Ottawa, Ontario, Canada.

Ardia, D., Mullen, K., 2010. The 'DEoptim' Package: Differential Evolution Optimization in 'R'. R package version 2.0-5.

Bai, Y., Wagener, T., Reed, P., August 2009. A top-down framework for watershed model evaluation and selection under uncertainty. Environmental Modelling & Software 24 (8), 901–916. URL http://dx.doi.org/10.1016/j.envsoft.2008.12.012

Bates, D., 2008. Comment on leland wilkinson, "the future of statistical computing". Technometrics 50 (4), 439–440.

Becker, R. A., Chambers, J. M., Wilks, A. R., 1988. The New S Language: A Programming Environment for Data Analysis and Graphics. Wadsworth & Brooks/Cole, Pacific Grove, CA, USA. URL http://www.worldcat.org/isbn/053409192X

Beven, K., March 2006. A manifesto for the equifinality thesis. Journal of Hydrology 320 (1-2), 18–36. URL http://www.sciencedirect.com/science/article/B6V6C-4H16S4M-1/2/571c8821621c803522cc823147bef169

Beven, K., 2008. On doing better hydrological science. Hydrological Processes 22 (17), 3549–3553. URL http://dx.doi.org/10.1002/hyp.7108

Blasone, R., Vrugt, J., Madsen, H., Rosbjerg, D., Robinson, B., Zyvoloski, G., April 2008. Generalized likelihood uncertainty estimation (glue) using adaptive markov chain monte carlo sampling. Advances in Water Resources 31 (4), 630–648. URL http://dx.doi.org/10.1016/j.advwatres.2007.12.003

Boughton, W., October 2004. The australian water balance model. Environmental Modelling & Software 19 (10), 943–956. URL http://dx.doi.org/10.1016/j.envsoft.2003.10.007

Box, G. E. P., Cox, D. R., 1964. An analysis of transformations. Journal of the Royal Statistical Society. Series B (Methodological) 26 (2), 211–252. URL http://www.jstor.org/stable/2984418

Boyle, D. P., Gupta, H. V., Sorooshian, S., 2000. Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods. Water Resources Research 36 (12), 3663–3674. URL http://dx.doi.org/10.1029/2000WR900207

Burnash, R. J. C., 1995. The NWS River Forecast System – Catchment Modeling, revised Edition. Water Resources Publications, Highlands Ranch, Colo.

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., Wagener, T., Hay, L. E., August 2008. Framework for understanding structural errors (fuse): A modular framework to diagnose differences between hydrological models. Water Resources Research 44 (12), W00B02+. URL http://dx.doi.org/10.1029/2007WR006735

Cleveland, R. B., Cleveland, W. S., McRae, J. E., Terpenning, I., 1990. Stl: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics 6, 3–73.

Croke, B. F. W., April 2006. A technique for deriving an average event unit hydrograph from streamflow-only data for ephemeral quick-flow-dominant catchments. Advances in Water Resources 29 (4), 493–502. URL http://dx.doi.org/10.1016/j.advwatres.2005.06.005

Croke, B. F. W., July 2009. Representing uncertainty in objective functions: extension to include the influence of serial correlation. In: Anderssen, R. S., Braddock, R. D., Newham, L. T. H. (Eds.), 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, pp. 3372–3378. URL http://www.mssanz.org.au/modsim09/I7/croke.pdf

Croke, B. F. W., Andrews, F., Jakeman, A. J., Cuddy, S., Luddy, A., February 2005. Redesign of the ihacres rainfall-runoff model. In: 29th Hydrology and Water Resources Symposium.

Croke, B. F. W., Jakeman, A. J., 2004. A catchment moisture deficit module for the ihacres rainfall-runoff model. Environmental Modelling and Software 19, 1–5.

Dawson, C., Abrahart, R., See, L., July 2007. Hydrotest: A web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts. Environmental Modelling & Software 22 (7),

1034–1052.
URL http://dx.doi.org/10.1016/j.envsoft.2006.06.008
Dawson, C. W., Abrahart, R. J., See, L. M., November 2010. Hydrotest: Further development of a web resource for the standardised assessment of hydrological models. Environmental Modelling & Software 25 (11), 1481–1482.
URL http://dx.doi.org/10.1016/j.envsoft.2009.01.001
Duan, Q., Sorooshian, S., Gupta, V., 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resources Research 28 (4), 1015–1031.
URL http://dx.doi.org/10.1029/91WR02985
Fenicia, F., Savenije, H. H. G., Matgen, P., Pfister, L., January 2008. Understanding catchment behavior through stepwise model concept improvement. Water Resources Research 44 (1), W01402+.
URL http://dx.doi.org/10.1029/2006WR005563
Fox, J., 2008. Editorial. R News 8/2, 1–2.
Gentleman, R., Temple Lang, D., 2007. Statistical analyses and reproducible research. Journal of Computational and Graphical Statistics 16 (1), 1–23.
Herron, N., Croke, B., May 2009. Including the influence of groundwater exchanges in a lumped rainfall-runoff model. Mathematics and Computers in Simulation 79 (9), 2689–2700.
URL http://dx.doi.org/10.1016/j.matcom.2008.08.007
Hyndman, R. J., 2010. forecast: Forecasting functions for time series. R package version 2.05.
URL http://CRAN.R-project.org/package=forecast
Ihaka, R., Gentleman, R., 1996. R: A language for data analysis and graphics. Journal of Computational and Graphical Statistics 5, 299–314.
Jakeman, A., Letcher, R., Norton, J., May 2006. Ten iterative steps in development and evaluation of environmental models. Environmental Modelling & Software 21 (5), 602–614.
URL http://dx.doi.org/10.1016/j.envsoft.2006.01.004
Jakeman, A. J., Hornberger, G. M., 1993. How much complexity is warranted in a rainfall-runoff model? Water Resources Research 29, 2637–2649.
Jakeman, A. J., Littlewood, I. G., Whitehead, P. G., 1990. Computation of the instantaneous unit hydrograph and identifiable component flows with application to two small upland catchments. Journal of Hydrology 117, 275–300.
Katz, R. W., Parlange, M. B., Naveau, P., August 2002. Statistics of extremes in hydrology. Advances in Water Resources 25 (8-12), 1287–1304.
URL http://dx.doi.org/10.1016/S0309-1708(02)00056-8
Kim, H. S., Croke, B. F. W., Jakeman, A. J., Chiew, F. H. S., Mueller, N., December 2007. Towards separation of climate and land use effects on hydrology: data analysis of the googong and cotter catchments. In: Oxley, L., Kulasiri, D. (Eds.), MODSIM 2007 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, pp. 74–80.
Kokkonen, T., Koivusalo, H., Jakeman, A., Norton, J., July 2006. Construction of a degree-day snow model in the light of the "ten iterative steps in model development". In: Voinov, A., Jakeman, A. J., Rizzoli, A. E. (Eds.), Proceedings of the iEMSs Third Biennial Meeting: "Summit on Environmental Modelling and Software". International Environmental Modelling and Software Society, Burlington, USA.
Leavesley, G. H., Markstrom, S. L., Viger, R. J., 2006. USGS Modular Modeling System (MMS) âĂŞ-Precipitation-Runoff Modeling System (PRMS). CRC Press, pp. 159–177.
Littlewood, I. G., Croke, B. F. W., Jakeman, A. J., Sivapalan, M., 2003. The role of 'top-down' modelling for prediction in ungauged basins (pub). Hydrol. Process. 17 (8), 1673–1679.
URL http://dx.doi.org/10.1002/hyp.5129
Ljung, L., 1999. System Identification: Theory for the User, 2nd Edition. Prentice Hall.
McIntyre, N., Al-Qurashi, A., June 2009. Performance of ten rainfallâĂŞrunoff models applied to an arid catchment in oman. Environmental Modelling & Software 24 (6), 726–738.
URL http://dx.doi.org/10.1016/j.envsoft.2008.11.001
Nash, J., Sutcliffe, J., April 1970. River flow forecasting through conceptual models part i - a discussion of principles. Journal of Hydrology 10 (3), 282–290.
URL http://dx.doi.org/10.1016/0022-1694(70)90255-6
Norton, J., Chanat, J., June 2005. Linear time-varying models to investigate complex distributed dynamics: A rainfall-runoff example. Mathematics and Computers in Simulation 69 (1-2), 123–134.
URL http://dx.doi.org/10.1016/j.matcom.2005.02.029
Perrin, C., Michel, C., Andreassian, V., August 2003. Improvement of a parsimonious model for streamflow simulation. Journal of Hydrology 279 (1-4), 275–289.

URL http://dx.doi.org/10.1016/S0022-1694(03)00225-7
Petris, G., Petrone, S., Campagnoli, P., 2009. Dynamic Linear Models with R. Use R! Springer.
Price, K., Storn, R. M., Lampinen, J. A., December 2005. Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), 1st Edition. Springer.
URL http://www.worldcat.org/isbn/3540209506
R Development Core Team, 2010. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org/
Sarkar, D., March 2008. Lattice: Multivariate Data Visualization with R (Use R), 1st Edition. Springer.
URL http://www.worldcat.org/isbn/0387759689
Sarkar, D., 2010. lattice: Lattice Graphics. R package version 0.18-5.
URL http://r-forge.r-project.org/projects/lattice/
Sarkar, D., Andrews, F., 2010. latticeExtra: Extra Graphical Utilities Based on Lattice. R package version 0.6-12.
URL http://latticeextra.r-forge.r-project.org/
Savenije, H. H. G., February 2009. Hess opinions: The art of hydrology. Hydrology and Earth System Sciences 13 (2), 157–161.
URL http://www.hydrol-earth-syst-sci.net/13/157/2009/
Selle, B., Hannah, M., August 2010. A bootstrap approach to assess parameter uncertainty in simple catchment models. Environmental Modelling & Software 25 (8), 919–926.
URL http://dx.doi.org/10.1016/j.envsoft.2010.03.005
Sivapalan, M., Blöschl, G., Zhang, L., Vertessy, R., 2003. Downward approach to hydrological prediction. Hydrological Processes 17 (11), 2101–2111.
URL http://dx.doi.org/10.1002/hyp.1425
Viviroli, D., Zappa, M., Gurtz, J., Weingartner, R., October 2009. An introduction to the hydrological modelling system prevah and its pre- and post-processing-tools. Environmental Modelling & Software 24 (10), 1209–1222.
URL http://dx.doi.org/10.1016/j.envsoft.2009.04.001
Vrugt, J. A., ter Braak, C. J. F., Diks, C. G. H., Robinson, B. A., Hyman, J. M., Higdon, D., 2009. Accelerating markov chain monte carlo simulation by differential evolution with self-adaptive randomized subspace sampling. International Journal of Nonlinear Sciences and Numerical Simulation 10 (3), 273–290.
Wagener, T., Boyle, D. P., Lees, M. J., Wheater, H. S., Gupta, H. V., Sorooshian, S., November 2001. A framework for development and application of hydrological models. Hydrology and Earth System Sciences 5 (1), 13–26.
URL http://www.hydrol-earth-syst-sci.net/5/13/1999/
Wagener, T., Kollat, J., July 2007. Numerical and visual evaluation of hydrological and environmental models using the monte carlo analysis toolbox. Environmental Modelling & Software 22 (7), 1021–1033.
URL http://dx.doi.org/10.1016/j.envsoft.2006.06.017
Wagener, T., Reed, P., van Werkhoven, K., Tang, Y., Zhang, Z., 2009. Advances in the identification and evaluation of complex environmental systems models. Journal of Hydroinformatics 11 (3).
URL http://dx.doi.org/10.2166/hydro.2009.040
Wheater, H. S., Jakeman, A. J., Beven, K. J., 1993. Progress and directions in rainfall-runoff modelling. John Wiley, Chichester, pp. 101–132.
Wickham, H., 2009. ggplot2: elegant graphics for data analysis. Springer New York.
URL http://had.co.nz/ggplot2/book
Willems, P., March 2009. A time series tool to support the multi-criteria performance evaluation of rainfall-runoff models. Environmental Modelling & Software 24 (3), 311–321.
URL http://dx.doi.org/10.1016/j.envsoft.2008.09.005
Williams, G. J., December 2009. Rattle: A data mining gui for r. The R Journal 1 (2), 45–55.
URL http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf
Wood, S. N., 2004. Stable and efficient multiple smoothing parameter estimation for generalized additive models. Journal of the American Statistical Association 99 (467), 673–686.
Ye, W., Bates, B. C., Viney, N. R., Sivapalan, M., Jakeman, A. J., 1997. Performance of conceptual rainfall-runoff models in low-yielding ephemeral catchments. Water Resources Research 33, 153–166.
Young, P., August 2003. Top-down and data-based mechanistic modelling of rainfall-flow dynamics at the catchment scale. Hydrological Processes 17 (11), 2195–2217.
URL http://dx.doi.org/10.1002/hyp.1328

Young, P. C., April 2008. The refined instrumental variable method. Journal Européen des Systèmes Automatisés 42 (2-3), 149–179.
URL http://dx.doi.org/10.3166/jesa.42.149-179

Young, P. C., Ratto, M., October 2008. A unified approach to environmental systems modeling. Stochastic Environmental Research and Risk Assessment.
URL http://dx.doi.org/10.1007/s00477-008-0271-1

Zeileis, A., Grothendieck, G., 2005. zoo: S3 infrastructure for regular and irregular time series. Journal of Statistical Software 14 (6), 1–27.
URL http://www.jstatsoft.org/v14/i06/

Appendix A. Statistics

A set of pre-defined statistics is provided in the **hydromad** package and are listed below. These are not intended to be a comprehensive set, but rather for use as quick examples and to demonstrate how to define such functions.

Raw time-step based statistics:

- `bias`: bias in data units, $\sum(X - Q)$
- `rel.bias`: bias as a fraction of the total observed flow, $\sum(X - Q)/\sum Q$
- `abs.err`: the mean absolute error, $\mathsf{E}|X - Q|$
- `RMSE`: Root Mean Squared Error, $\sqrt{\mathsf{E}(X - Q)^2}$
- `r.squared`: R Squared (Nash-Sutcliffe Efficiency), $1 - \sum(Q - X)^2/\sum(Q - \bar{Q})^2$
- `r.sq.sqrt`: R Squared using square-root transformed data: $1 - \frac{\sum|\sqrt{Q}-\sqrt{X}|^2}{\sum|\sqrt{Q}-\sqrt{\bar{Q}}|^2}$
- `r.sq.log`: R Squared using log transformed data, with an offset: $1 - \frac{\sum|\log(Q+\epsilon)-\log(X+\epsilon)|^2}{\sum|\log(Q+\epsilon)-\log(\bar{Q}+\epsilon)|^2}$. Here ϵ is the 10 percentile (i.e. lowest decile) of the non-zero values of Q.
- `r.sq.boxcox`: R Squared using a Box-Cox transform (Box and Cox, 1964). The power λ is chosen to fit Q to a normal distribution. When $\lambda = 0$ it is a log transform; otherwise it is $y_* = \frac{(y+\epsilon)^\lambda - 1}{\lambda}$. Here ϵ is the 10 percentile (i.e. lowest decile) of the non-zero values of Q.
- `r.sq.diff`: R Squared using differences between successive time steps, i.e. rises and falls.
- `r.sq.smooth5`: R Squared using data smoothed with a triangular kernel of width 5 time steps: `c(1,2,3,2,1)/9`.
- `persistence`: R Squared where the reference model predicts each time step as the previous observed value. This statistic therefore represents a model's performance compared to a naive one-time-step forecast.
- `r.sq.seasonal`: R Squared where the reference model is the mean in each calendar month, rather than the default which is the overall mean.

- `r.sq.vs.tf`: R Squared where the reference model is a second-order transfer function (two stores in parallel) fitted directly to the rainfall data, i.e. assuming a constant runoff ratio. The indicates the marginal improvement of including the SMA model component.
- `r.sq.vs.tf.bc`: R Squared using a Box-Cox transform where the reference model is a second-order transfer function (two stores in parallel) fitted directly to the rainfall data, i.e. assuming a constant runoff ratio. The indicates the marginal improvement of including the SMA model component.
- `X0`: correlation of modelled flow with the model residuals.
- `X1`: correlation of modelled flow with the model residuals from the previous time step.
- `ARPE`: Average Relative Parameter Error. Requires that a variance-covariance matrix was estimated during calibration.

Aggregated and event-based statistics:

- `r.sq.monthly`: R Squared with data aggregated into calendar months.
- `e.rain5`: R Squared of flow totals in each event, defined by rainfall exceeding 5mm per time step, and continuing until the next such event. Each single event continues at least until rainfall remains below 1 mm for 4 time steps. An example of this event definition is shown in Figure X.
- `e.rain5.log`: same as `e.rain5` but the event totals are log-transformed (with the non-zero 10 percentile of these totals as an offset).
- `e.rain5.bc`: same as `e.rain5` but the event totals are Box-Cox-transformed (with the non-zero 10 percentile of these totals as an offset).
- `e.q90`: R Squared of flow totals in each event, defined by observed flow exceeding the 90 percentile level for at least 2 time steps, and continuing until the next such event. Each single event continues at least until flow falls below the 90 percentile level for 4 time steps, and must be separated from the next event by a further 5 time steps. An example of this event definition is shown in Figure X.
- `e.q90.log`: same as `e.q90` but the event totals are log-transformed.
- `e.q90.bc`: same as `e.q90` but the event totals are Box-Cox-transformed.
- `e.q90.all`: R Squared of flow totals in *and between* each event, defined by observed flow exceeding the 90 percentile level for at least 2 time steps, and continuing until the flow falls below the 90 percentile level for 4 time steps. Events must be separated by a further 5 time steps. An example of this event definition is shown in Figure X.
- `e.q90.all.log`: same as `e.q90.all` but the event totals are log-transformed.
- `e.q90.all.bc`: same as `e.q90.all` but the event totals are Box-Cox-transformed.

- `e.q90.min`: the same as `e.q90` but instead of flow totals in each event, the minimum flow in each event is extracted.
- `e.q90.min.log`: the same as `e.q90.min` but flow minima are log-transformed (with the non-zero 10 percentile of these minima as an offset).