tradesys: A framework for encoding and backtesting trading systems in R

Robert Sams

August 11, 2009

1 Introduction

The tradesys package is for modelling trading systems in R. The key functionality of the package is centered around the 'tsys' class. A 'tsys' object collects all of the information needed to completely define a trading system. The key ideas behind the system, like the entry and exit signals, are written by the user in R and stored as unevaluated R expressions in the appropriate 'tsys' slot. The object can subsequently be applied to any timeseries data with the appropriate variables, the logic of the system encoded in the stored expressions are evaluated on this data, and the system's long/short/flat *states* and the resulting *equity curve* are calculated. In short, a trading system consists of certain R-encoded logic and meta-data, defined in an object of class 'tsys', and evaluated on any data of the appropriate structure to calculate the system's states and equity.

As the above description implies, tradesys is about *modelling* a trading system, not analyzing its *results*, the latter being what is properly referred to as "backtesting"....

So, the functionality of the package is modest in its scope. It is, however, ambitious in the implementation. The above model is needed in almost all trading system research and thus represents a problem in need of a common, well-designed solution. This package aims to do this to the highest standard, so that trading system builders who chose R as an important tool of analysis can confidently use this package a key component of their work.

There are three main design goals of the package. First, the package should be *trustworthy*, in the sense elaborated by Chambers

Second, *discoverability*. The logic behind every calculation should not only be scrupulously documented, but also discoverable, in the sense that the key computations done on 'tsys' objects are encapsulated in functions that the user can call and explore. So the package contains a number of functions that are not strictly speaking part of the user interface (the usage of the package can be had without ever calling them), but are documented seperately and exported by the package's namespace to be explored at will.

2 A formal definition of "trading system"

A *trading system* is an algorithm on a timeseries X_t that specifies, for each time t, whether the system's state is long, short or flat. Mathematically, it is a function $f(X_t)$ that calculates each state $s_i \in \{1, 0, -1\}$ on the basis of $X_1, ..., X_i$. X_t may be as simple as a daily series of closing prices but is often a multivariate series with various price and other data. The states vector combined with the timeseries is the raw material for backtesting research from the calculation of period returns onwards. Let's call such a combination $\{X_t, s_t\}$ a *trading system time series*. In this package a trading system time series is represented as class `tsts`.

But what about $f(X_t)$, what form does it take?...

3 Introductory Examples

4 Splicing Timeseries

Computational details

The results in this paper were obtained using R 2.8.0 with the packages `tradesys` 0.1 and `zoo` 1.5–4 R itself and all packages used are available from CRAN at `http://CRAN.R-project.org/`.