quantmod

version 0.3

Quantitative Financial Modelling Framework

Jeffrey A. Ryan www.quantmod.com

Introduction

While many tools currently exist within **R** to manage different stages of the quantitative trading process, no single mechanism exists to manage the entire workflow from data management and alpha exploration to signal generation and post-trade analysis.

After a brief discussion on R-sig-Finance in 2006 regarding the pros and cons of a complete **R** workflow, the author decided that the R community needed an option to manage as much of the process from within **R** as possible.

More details, as well as examples of the **quantmod** package, can be found online at http://www.quantmod.com

What quantmod *IS*:

- A workflow tool to manage the various steps involved in quantitative financial modelling and trading. *100% R workflow.*
- An unobtrusive wrapper to R functionality, offering the analyst full access to both core and contributed tools without having to learn multiple interfaces, all while streamlining the process by removing non-core programming tasks.
- A new way to manage financial data in R - regardless of frequency, source, or class - through an innovative uniform interface
- Easy to extend and integrate into any current modelling workflow -- use what you like and *only* what you like.

What quantmod *IS NOT*:

- It is **NOT** a replacement for *Rmetrics*, *PerformanceAnalytics*, *portfolio*, or any other statistical or financial package in R.
- It is **NOT** a simplification of current tools - its sole purpose is to standardize the interface to certain tools - speeding the process of model development for trading.
- It is **NOT** meant to be used alone. Rather, it is meant to complement the current **R** tools, with a focus on workflow efficiencies and through the addition of functionality not available elsewhere in **R**.
- Most importantly - It is ***NOT*** complete...

The quantmod workflow

Pre-Modelling

Managing data for trading can be a challenge, as an analyst is presented with data of varying frequency and source, which must be processed into one data object for model construction purposes.

```
getSymbols("IBM",src="yahoo",
           return.class="its")

Simple interface to fetch data
from a variety of sources and
coerce to the specified class.

setSymbolsLookup(IBM='MySQL')

Set defaults to streamline
process
```

```
buildData(Next(Op(T))~OpCl(T))

Given a formula, automatically
construct a dataset without
explicitly loading or managing
the underlying data.

The result is a data object that
can be used for external
analysis.
```

```
specifyModel(Next(Op(T)~OpCl(T))

Creates a quantmod object for
use within the quantmod
workflow. Data may also be
extracted from the object with a
call to modelData.
```

Modelling

Once the data has been processed it can be used by any function within **R**, either as an extracted data object, or as a quantmod object. Within the quantmod workflow the next step is *buildModel*, a wrapper function to many of **R**'s most applicable univariate fitting methods.

Currently implemented *buildModel* methods include:

lm [stats], **glm** [stats], **loess** [stats], **ppr** [stats], **rpart** [rpart], **tree** [tree], **randomForest** [randomForest], **mars** [mda], **polymars** [polspline], **lars** [lars], **rq** [quantreg], **lqs** [MASS], **rlm** [MASS], **nnet** [nnet].

```
buildModel(my.quantmod.object, method="ppr", training.per=c('2006-01-01','2006-03-01'))

Simply pass a model specified with specifyModel and specify the method desired. Any further arguments
to the underlying method can be included after the training period argument.

Additional wrappers for new methods may added by the user via buildModel.skeleton.
```

Post-Modelling

After a model has been constructed, it is desirable to analyse the fit, as well as apply new data to examine the model's potential return and risk. With an emphasis on tools external to the quantmod package, a variety of methods are available to *extract the fitted model*, as well as *calculate a return series* given additional trading rules, such as leverage, stop-loss rules, and transaction cost accounting.

"Trade" model and evaluate ...

```
traded<-tradeModel(x,leverage=2,
                   trade.rule=end.of.day)

Simulate trading of model over full
dataset calculating basic return and risk
measures.

Additional "trade.rules" may be user
specified.
```

... using standard R tools

```
fittedModel(my.quantmod)
Extract fitted object. Exam-
ine residuals, coefficients,
 etc.

monthlyReturn(traded)
Get returns suitable for
PerformanceAnalytics
analysis.
```

...or *quantmod* tools.

```
modelReturns(traded):
Summary return measures

 modelRisk(traded):
 Summary risk measures

modelAnalysis(traded):
Summary of risk and return
```

The future of quantmod

As development moves toward a release version of 1.0, some particular areas of interest are:

- Continue to integrate packages of interest to the finance community into the quantmod workflow.
- Continue to add new **buildModel** wrappers to extend the options during the modelling stage of development
- Continue to add new data source methods to **getSymbols**, including ODBC and additional database drivers, internet downloads (public and commercial), and proprietary data feeds (Bloomberg, Interactive Brokers, etc.)
- Improve model analysis routines within the package, to minimize any gaps in typical workflow usage
- Add a **modelSignificance** function, to include Monte-Carlo studies, as well as exploration of random portfolios.
- New graphics for risk, return and significance analysis
- Add mechanisms to manage stored models.
- **Solicit input and contributions from the R community.**

Acknowledgments

- R Development Core Team (2007). R: A language and environment for statistical computing. R Foundation for Statistical Computing,Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.
- John Bollinger, Personal Communication (2006). R-sig-Finance. URL https://stat.ethz.ch/pipermail/r-sig-finance/2006q2/000844.html
- Achim Zeileis and Gabor Grothendieck (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/
- Peter Carl and Brian G. Peterson (2007). PerformanceAnalytics: Econometric tools for performance and risk analysis.. R package version 0.9.5. http://braverock.com/R/
- Patrick Burns (2004). Performance measurement via random portfolios. Working paper, Burns Statistics, URL http://www.burns-stat.com
- Portfolio & Risk Advisory Group and Commerzbank Securities (2006). its: Irregular Time Series. R package version 1.1.5.
- Diethelm Wuertz, many others and see the SOURCE file (2007). fCalendar: Rmetrics - Chronological and Calendarical Objects. R package version 251.70. http://www.rmetrics.org
- A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18--22.
- Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0
- S original by Trevor Hastie & Robert Tibshirani. R port by Friedrich Leisch, Kurt Hornik and Brian D. Ripley. (2006). mda: Mixture and flexible discriminant analysis. R package version 0.3-2.
- Trevor Hastie and Brad Efron (2007). lars: Least Angle Regression,Lasso and Forward Stagewise. R package version 0.9-7. http://www-stat.stanford.edu/~hastie/Papers/#LARS
- Brian Ripley. (2007). tree: Classification and regression trees. R package version 1.0-25.
- Terry M Therneau and Beth Atkinson. R port by Brian Ripley <ripley@stats.ox.ac.uk>. (2007). rpart: Recursive Partitioning. R package version 3.1-36. S-PLUS 6.x original at http://mayoresearch.mayo.edu/mayo/research/biostat/splusfunctions.cfm
- Roger Koenker (2007). quantreg: Quantile Regression. R package version 4.08. http://www.r-project.org
- Charles Kooperberg (2006). polspline: Polynomial spline routines. R package version 1.0.13.
- Jeffrey A. Ryan (2007). Defaults: Create Global Function Defaults. R package version 1.0-5.