CAMBRIDGE

Cambridge University Press
978-1-107-15075-1 — Financial Analytics with R
Mark J. Bennett , Dirk L. Hugen
Excerpt
More Information

1 Analytical Thinking

As an investor, there is no more immediate feeling of excitement than a stock split during a bull market or an acquisition that can bid one's stock up by 20 percent in a day. Maybe it is something like the feeling a soccer forward gets after completing a kick into the goal. Even though semi-random events can risk the desired outcome (specifically, actions by the defenders between the goal and the forward), all of the practice and preparation for the offense is being applied quickly, and the result of success is appreciated by the fans. In a nutshell, when faced with risks, preparation makes success more probable.

This book is all about that preparation. Being one step closer to paying for one's child's college education, or replacing one's employment income as the prospect of retirement looms, makes us feel more financially secure. Financial analytics involves, among other analysis, the creation of forecasted scenarios based upon historical data using simulations. When amateur investors talk about stocks in a qualitative sense – for example, "Hey, Qualcomm is really rocking lately, bud!" or "Hey, I bought some Intuitive Surgical and it's really on a roll!" – this is the way we naturally interact: informal advice. We are human and need to feel our way through many situations. But the question to ask one's self quietly is: "Sure, that sounds like a good investment that you are telling me about, but is there an alternative investment that will do better than that based on historical evidence?" What would a financial analytics approach tell us from a purely objective perspective? As we practice financial analytics, like the soccer forward practicing goal kicking, we are trained, informed, and more prepared for unexpected situations.

Along with robust and accurate data, designing models is a key component of the professional practice of analytics. This book focuses on mastering some of the most important applied models so that they can be adapted, relied upon, and expanded. Models are presented using hand-written code in the R language, using historical market datasets to gain a deeper understanding of the behavior.

Coined from "Big Science," "Big Data" is a term used to describe datasets that are too large to fit into common memory and disk hardware and traditional files and relational databases. Sophisticated algorithms and processing are often required to analyze Big Data. Analytics are applied to Big Data in order to take advantage of the large sample sizes. Insights and discovery are simply more realistically possible with large datasets. This book is intended to foster an individual and classroom software laboratory for performing financial analytics. It serves as a resource for models, program logic, and datasets for stocks and other common securities as we tackle Big Data.

1.1 What Is Financial Analytics?

Since the 2008 financial crisis, market practitioners are realizing that reliance on models which are mathematically pure but fundamentally inaccurate is no longer acceptable. A more practical approach is needed. The markets where the instruments reside have many more tail events than most of the market models of the 2000 decade would acknowledge. These tail events have contributed to the flash crash, tech bubble, and mortgage-based crisis with more to come. Practitioners are in need of tools for quick discovery and simulation to complement and calibrate the mathematics.

Meanwhile, the emerging new field of Analytics, also known as Data Science, is providing computational intelligence to businesses in ways many had never envisioned. Analytical computer programs are recommending everything from medical diagnoses to automobile routes to entertainment contents. Analytics is a practical and pragmatic approach where statistical rules and discrete structures are automated on the datasets as outcomes are observed in the laboratory and in the business world. Corporations are able to mine transactional data and predict future consumer buying patterns. Health professionals can mine health records to help with decision analysis for diagnosing diseases.

In today's world, businesses as well as consumers are affected by fluctuations in consumer prices, industrial production, interest rates, and the price of natural gas. These changes let us know that risk is ever-present. Now that large datasets are widely available, market practitioners are stepping up their efforts to use algorithms to measure econometric patterns and examine their expected trends.

Analytics has become the term used for describing the iterative process of proposing models and finding how well the data fit the models, and how to predict future outcomes from the models. Financial analytics describes our subject: a domain where contributions have been made by scholars and industry professionals for decades, and where the latest technology advancements have made recent discoveries possible. Financial analytics involves applying classic statistical models and computerized algorithms to the financial market data and investment portfolios. Analytics applied in this area address relationships that occur in practice every day in time-critical fashion as investors, speculators, and producers of valued securities and commodities trade across the country and the globe.

Investment firms like PIMCO and Vanguard have helped investors meet retirement goals or send their children to college by carefully delivering positive market exposure. This book will provide the tools for being able to understand better what firms like these and other financial entities do.

While many business intelligence books have been written to describe *what* is happening in Big Data, this book is specifically focused on *how* to achieve detailed results. The book is multidisciplinary in its combining of statistics, finance, and computer science.

Businesses are looking for profitability and financial risk reduction. Optimization is an important aspect of financial analytics. Any business intelligence approach makes appropriate use of data to attempt to optimize outcomes.

These are the kinds of issues to be tackled by financial analytical simulations. What is the optimized return and what is the level of risk assumed? What kinds of financial metrics can become good random variables and how are they distributed? What datasets are available to sample these random variables analytically? Which financial metrics are highly correlated? Which are relatively independent? Can analytical thinking give an algorithm an edge over a simple holding strategy when generating transactions? These are questions we explore in this book.

1.2 What Is the Laptop Laboratory for Data Science?

Professional data scientists are not purely statisticians. Yes, applied statistical skills are important, but they must also possess practical software engineering skills and be able to build reliable and testable models that run rapidly, repeatably. They must understand data types in order to implement analytical algorithms, so that their employers and clients will gain a competitive advantage from the robust models they produce. Our aim here is not to treat one financial instrument at a time, but rather in mass. In essence, clusters and structures of instruments are needed so that comparisons can be made. Investing is a matter of decision-making, and the more stock candidates, the better the chances of success.

Regarding the subtitle, “Building a Laptop Laboratory for Data Science,” this book guides the reader on how to build a software simulation laboratory on which significant-sized working modules can answer analytics inquiries. Laptops are fast becoming pervasive. When using the R language, any operating system will do. As evidence of how inexpensive powerful computing has become, the laptop on which all of the book’s simulations were run can be found for less than $200. After installing Crouton, a variant of Ubuntu Linux® found for free online, and RStudio, an analyst can soon be downloading datasets and analyzing away with millions of rows from freely available financial datasets.

Our own laptop computer is called AL, short for Analytics Laboratory. (In some families, cars are given names: Betsy, Handsome, Chester, Venom, and Myles are typical. If cars can be named, why not name a device that is at one’s side most days?) AL’s hardware and operating system was purchased from a Groupon coupon for the nominal price of $139. Your version of AL could run on an Apple or Windows PC: any computer that can run RStudio and hold a large set of flat files and a small portion of a database will do.

For the flat files, one of the coded modules downloads and caches six million rows of prices for subsequent analysis. While many books have included code, in this book, when we include the code, the pieces build upon each other, providing an increasing level of sophistication as readers follow along with the option to try running the code on their own computers. The reader can use R on any type of computer operating system for which R is available, which covers most of them. When running the Analytics Library on a higher-performance Apple or Windows laptop with a large internal hard drive, one can literally load the “whole market” and perform queries at will.

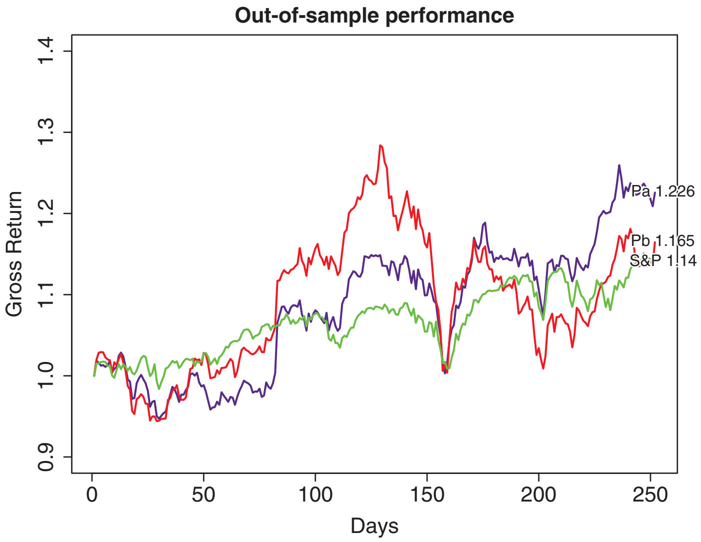


Figure 1.1 An out-of-sample calibration (2014 to 2015) of the S&P benchmark portfolio (green) and two optimized portfolios of NASDAQ and NYSE stocks (purple and red).

It has been a successful year for AL, the laptop laboratory for data science. On the data mining side of things, through the analytical programs presented herein, AL was able to find choice stock candidates for the portfolio optimizer. By using classic mean–variance optimization, the R program was able to deliver a stock portfolio that beat the S&P 500 Index return: not in sample in the laboratory, but in the actual stock market. This was accomplished by putting together a portfolio that had higher volatility than the S&P 500 Index, but not substantially higher. In fact, when measuring the return over risk of the portfolio from AL compared to the Index, it was better. This means more return for the amount of risk we are taking. Figure 1.1 shows the in-sample performance of two optimized portfolios against this benchmark.

```
> logRetPortf = diff(log(indexRecentPrices1))
> mean(logRetPortf)/(sd(logRetPortf)*sqrt(252))
[1] 0.006083248
> logRetBench = diff(log(benchPrices/benchPrices[1]))
> mean(logRetBench)/(sd(logRetBench)*sqrt(252))
[1] 0.00497942
```

You may ask why we think the recommendations of AL were successful. Well, by making use of R's functional and vectored expressive notation and packages, AL gives us an ability to process hundreds or even thousands of possible stocks and select the best one based upon the most consistent past performance. With R this can be done more correctly and with less code than with many other platforms.

Those of us who invest often receive emails from investment advisors, web sites and might pick stocks qualitatively. SeekingAlpha.com is one such web site we can read for information in decision-making. MotleyFool.com is another. These are good sites. They focus on a particular stock in the articles, and they can be quite entertaining when reading about why they think that the Google CFO resigned, or why they think a travel web site company is overvalued. Of course, AL does not need to be entertained; in fact, AL *cannot* be entertained, and it therefore does not get distracted with pieces of qualitative information. Unlike people who brag about certain winning picks they found by conversing with the right folks, AL is a system and can only look at datasets and statistics. Using AL with R enforces a rigorously quantitative decision approach which is worth considering.

1.3 What Is R and How Can It Be Used in the Professional Analytics World?

Since the financial crisis of 2008 there has been a need for professionals in the banking, insurance, fund management, and corporate treasury sectors who are more knowledgeable about statistics and data analysis and can discuss and measure the various risk metrics, especially those involving extreme events. While quantitative finance programs appeared at various universities in the 1990s, these programs are more mathematical in nature and students spend more of their time constructing proofs and deriving formulas and less of their time with datasets from the markets. While deriving the formulas helps the understanding of the models, on an opportunity cost basis, that time could be used building an operating data analysis platform.

People can enter the financial analytics profession through a combination of experience and education. Professionals with an extensive math background can find it easier to make the transition to the field. Those with a less formal math background, and those lacking some financial vocabulary, will find this book quite helpful in making the transition. This book provides the intuition and basic vocabulary as a step toward the financial, statistical, and algorithmic knowledge needed to resolve the industry problems and issues. For more experienced readers, the book presents the latest techniques, leveraging new packages which meld traditional finance metrics with modern data mining and optimization subjects. Professionals who are making the transition to analytics, professional quantitative finance analysts, and students who want to supplement their background with financial analytics, would find this book of interest.

This book presents a systematic way of developing analytical programs for finance in the statistical language R. R has become the language of choice for use in academic analytics circles because of its sophisticated expressibility for statistical algorithms. It is open source and freely available via download for all common computer operating systems. And thousands of previously contributed and available packages eliminate the need for redeveloping common algorithms from scratch.

Since financial analytics is the application of statistical and economic rules with computational logic in ways that can solve problems, the role of the analytical computer program is expanding: to tie together models which would previously have been isolated. These computer programs can be constructed more efficiently using specialized programming languages.

This book presents the reader, both the practitioner and the scholar, with many solutions in financial analytics. For individual investors and investment firm analysts, as shown in this book, the results can be obtained by a reference model and a manageable-sized R program. The book begins with a background in probability and statistics for markets, basic algorithms in R for finding price vector characteristics, including returns, split adjustments for quotes, and also comparing performance of securities, measuring volatility and risk, direction, skew, and market tail weight with examples. Finding optimal portfolios and using unsupervised machine learning techniques using graphs and clustering algorithms to connect related securities within portfolios are ways to gain insight. The acceleration of the speed of the financial markets means that quantitative analysis and financial engineering are no longer exclusively focused on minute details of a single instrument but on the big picture of thousands of prices and transactions happening nearly simultaneously. This book presents a new step in this direction.

1.4 Exercises

1.1. Examine Figure 1.1. What is the return of the S&P 500 Index in percent for the 252-day or one-year period, assuming it is adjusted so that it begins the period at 1.0, as in the figure?

2 The R Language for Statistical Computing

Like so many innovations in computing, including the Unix operating system and the C and C++ languages, the R language has its roots at AT&T Bell Laboratories during the 1970s and 1980s in the S language project (Becker, Chambers, and Wilks, 1988). People think that the S language would not have been designed in the way it was if it had been designed by computer scientists (Morandat, Hill, Osvald, and Vitek, 2012). It was designed by statisticians in order to link together calls to FORTRAN packages, which were well known and trusted, and it flourished in the newly developed Unix and C environment. R is an open source variant of S developed at the University of Auckland by Ross Ihaka and Robert Gentleman, first appearing in 1993 (Ihaka, 1998). The chosen rules for scoping of variables and parameter passing make it hard for interpreter and compiler writers to make R run fast. In order to remedy this, packages such as Rcpp have been developed for R, allowing R programs to call pre-compiled C++ programs to optimize sections of the algorithms which are bottlenecks in terms of speed (Eddelbuettel and Sanderson, 2014). We discuss the Rcpp package toward the end of the book.

Clearly the recent popularity of R, fueled by its open source availability and the need for statistical and analytical computing tools, shows that the benefits of R far outweigh the negatives. Overall, R is based upon the vector as a first class item in the language. R shares this attribute with LISP, Scheme, Python, and Matlab. This and the prevalence of over 4,000 publicly available packages are two of the many strengths of R. In this book, we will focus on R packages that revolve around financial analytics.

It is our intention to introduce R at this point for those readers who need or are interested in a summary. Feel free to skip this chapter if you are experienced in R. For those who are not, many of the examples are worth trying out in an R environment to get a feel for the language. By including this section, this book is self-contained and we make no assumption that the reader arrives at this book having an R background. Covering this chapter as an introduction to R or as an R refresher will position the reader for the upcoming analytical programs which will slice and dice market datasets to uncover what is happening.

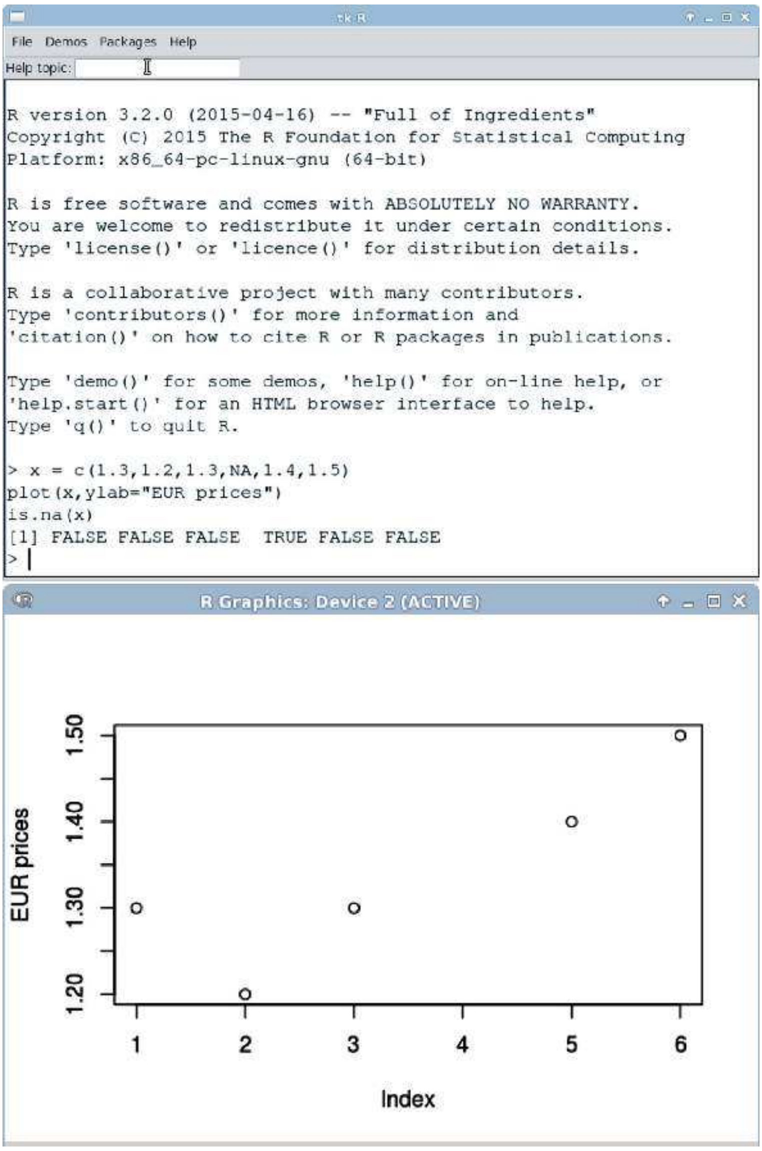
2.1 Getting Started with R

One of the great things about R is how easy it is to install. In your browser, head to the web site for the Comprehensive R Archive Network (CRAN), `http://cran.r-project.org` and, whether running an Apple Mac, a Linux system, or a Windows

PC, the basic R interpreter is available for download. R began as a command line interface (CLI), but, once downloaded and installed, there is a basic graphical user interface (GUI) available via the

```
R --gui=Tk
```

command on an Apple or Linux operating system, and that will display a GUI window as shown in Figure 2.1. For Windows, this same R GUI can be launched from an icon.



```
R version 3.2.0 (2015-04-16) -- "Full of Ingredients"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> x = c(1.3,1.2,1.3,NA,1.4,1.5)
plot(x,ylab="EUR prices")
is.na(x)
[1] FALSE FALSE FALSE  TRUE FALSE FALSE
> 
```

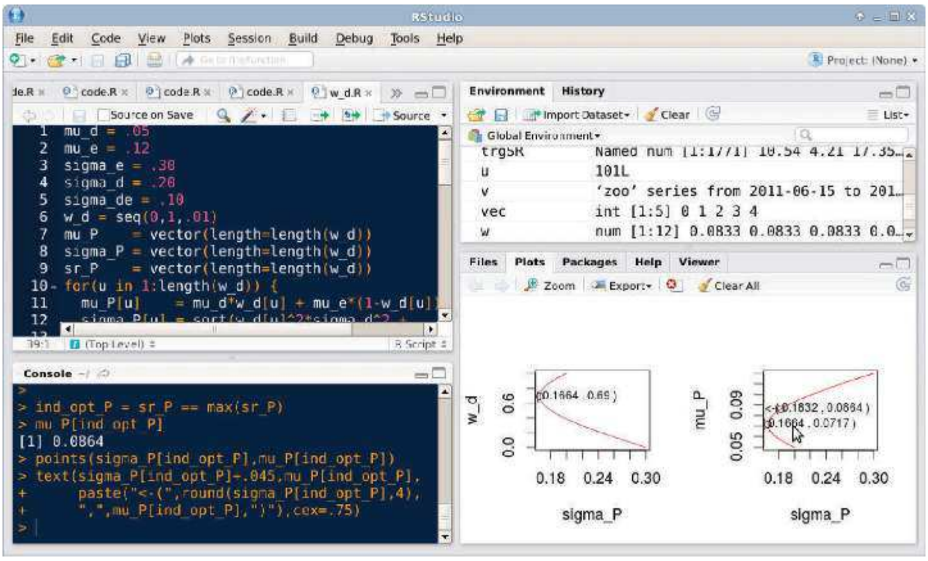
Figure 2.1 The basic R user interface window and a second pop-up window showing the result of the *plot*() command.

Just as a basic test, we can create a vector of prices and plot it with this block of code:

```
> x = c(1.3,1.2,1.3,NA,1.4,1.5)
> plot(x,ylab="EUR prices")
> is.na(x)
[1] FALSE FALSE FALSE  TRUE FALSE FALSE
```

The *c*() operator creates a vector of elements. This is the basic vector operator in R. Note the "not available" (NA) element appearing as the fourth item of the vector. R's ability to handle NAs, infinite values (Inf), and not a number (NaN) is one of its many strengths. Three Boolean-valued functions can be used to interrogate a variable for these respective values: *is.na(), is.infinite()*, and *is.nan()*. In data science, we certainly do encounter these erroneous values as inputs or results from algorithms.

Back on the subject of R interpreters, a later development is the RStudio GUI available from the web site `www.rstudio.com`. RStudio is a commercially developed GUI allowing management of plot windows, variable contents, and better debugging than the basic R interpreter. Figure 2.2 shows how the plotting window, variable contents, and workbench are all integrated into one view. People have spent years being productive in the basic R interpreter from CRAN, but those who have used interactive development environments for C++ or Java will find that the syntax highlighting, options for execution, and multiple source file handling of RStudio are more like what they are used to. Projects like RStudio and the Oracle Big Data Appliance are evidence of the growing popularity and commercialization of R (Ora, 2011).



```
mu_d = .05
mu_e = .12
sigma_e = .30
sigma_d = .20
sigma_de = .10
w_d = seq(0,1,.01)
mu_P     = vector(length=length(w_d))
sigma_P = vector(length=length(w_d))
sr_P     = vector(length=length(w_d))
for(u in 1:length(w_d)) {
  mu_P[u]      = mu_d*w_d[u] + mu_e*(1-w_d[u]
```

```
>
> ind_opt_P = sr_P == max(sr_P)
> mu_P[ind_opt_P]
[1] 0.0864
> points(sigma_P[ind_opt_P],mu_P[ind_opt_P])
> text(sigma_P[ind_opt_P]+.045,mu_P[ind_opt_P],
+      paste("<-(",round(sigma_P[ind_opt_P],4),
+       ",",mu_P[ind_opt_P],")"),cex=.75)
>
```

```
trgSR      Named num [1:1771] 10.54 4.21 17.35…
u          101L
v          'zoo' series from 2011-06-15 to 201…
vec        int [1:5] 0 1 2 3 4
w          num [1:12] 0.0833 0.0833 0.0833 0.0…
```

Figure 2.2 RStudio is a second-generation R user interface with integrated code, execution, variable inspection, and plotting windows, and expression completion.

A very important initial task in order to use the code from this book with an R language tool is to be sure to always have the current directory path defined by setting the *homeuser* variable. We reserve this variable to set the base directory where all the code for the book will reside. If, every time we use R, we set the *homeuser* variable as follows:

```
homeuser="<basedir>"
```

where <basedir> is something such as /home/<myuserid> or c:/Users/<myuserid>, which is specific to your computer system, then <basedir>/FinAnalytics/<dir> is where the input and output will occur from the R code. <dir> is typically ChapII for this chapter or another working directory name and is stored in the R variable *dir*. The publisher's web site for this book, www.cambridge.org/financialanalytics, contains a downloadable archive file with the code and datasets set up in directories so that FinAnalytics/<dir> will be ready once unpacked. The file is called FinAnalytics.zip. Download it and unpack it to obtain the book code, and remember to define the *homeuser* each time you use it.

Any time a *library* statement is encountered, R will check that the package is available. If not, it must be downloaded. As an example, to download the ggplot2 package, use the following command:

```
update.packages()
install.packages("ggplot2",dependencies=TRUE)
library(ggplot2)
```

Packages can be dependent upon other packages: hence the "dependencies=TRUE" setting. This flag is very important in order to avoid chasing down all the dependent packages and loading them one-by-one. Packages do not always succeed in loading. The best way to troubleshoot package installation is using your favorite browser and search engine to locate a helpful page on the World Wide Web by entering the error message into a good search engine.

2.2 Language Features: Functions, Assignment, Arguments, and Types

For many use cases, R provides a computational statistics platform. Mathematical functions are readily available. The basic log() function provides a natural logarithm. Of course, executing log() on a vector, *x*, results in a vector of natural logarithms, *y*. Unlike many imperative languages, no looping is required. The last line computes the simple expression, *y*, and prints its contents, rounded to three digits. Note how the NA value was preserved. The computation of log() on NA is NA as expected.

```
> #Filter prices:
> x[x > 1.3]
[1]  NA 1.4 1.5
> #Keeps the NA intact:
> y <- diff(log(x))
> round(y,3)
[1] -0.080  0.080     NA     NA  0.069
```